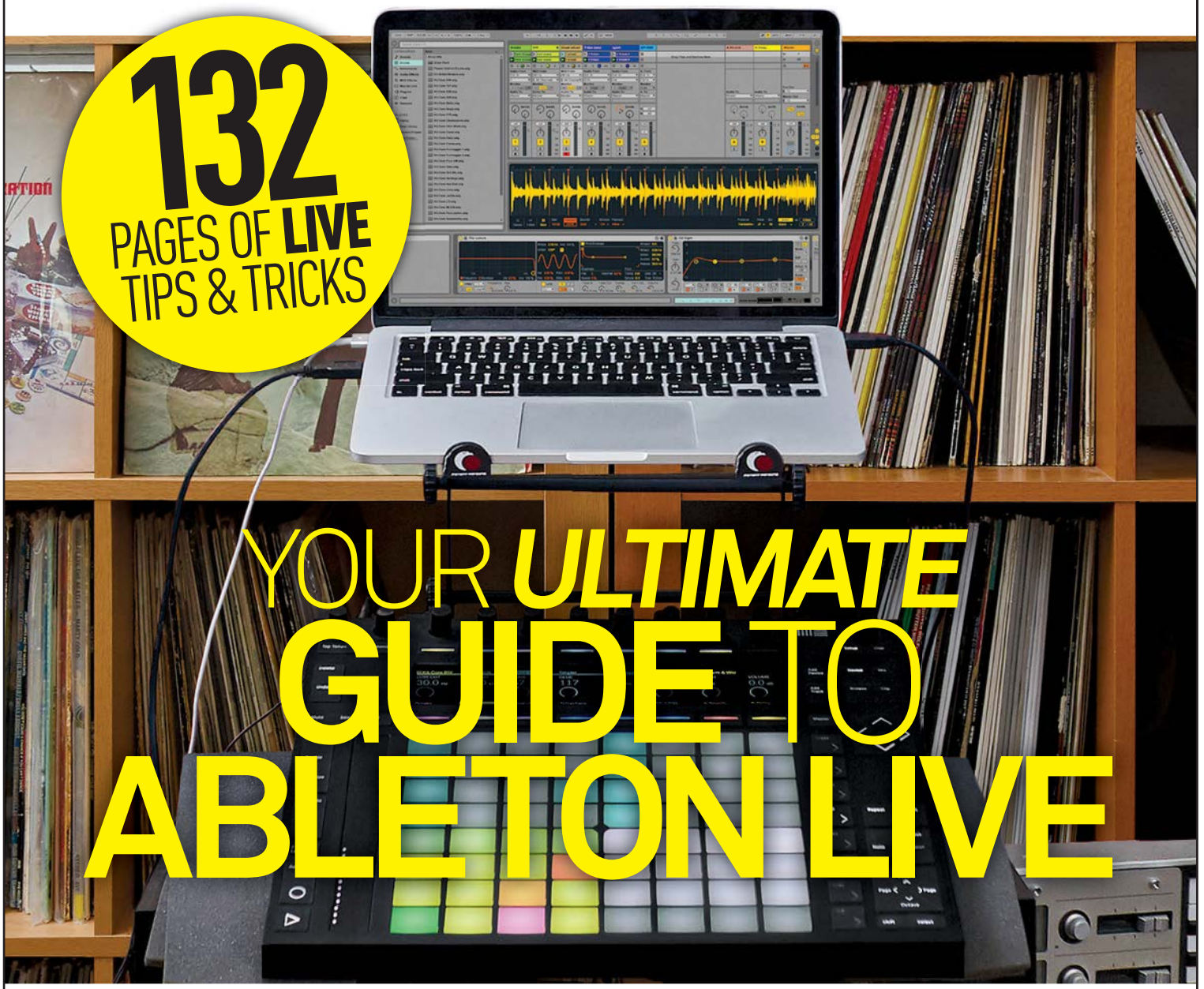
MusicTech FOCUS

DVD ROM FREE 1GB OF ROYALTY FREE SAMPLES

ABLETON LIVE 2017

THE IN-DEPTH GUIDE FOR THE CREATIVE MUSICIAN



- **34 pages of all-new workshops** – ESSENTIAL **Live help!**
- How to **use Link & connect hardware** to your Live set-up
- The **new guide to mastering** your music **for online**
- **Packed DVD! 1GB** of royalty-free samples & **2hrs** of video
- PLUS: **Freeware 50 Guide** • Live interviews • **40 gear tests**

ANTHEM

9 781911 308065
Ableton Live 2017
£8.99

Compiled by the Ableton Live experts from MusicTech

PLUGINBOUTIQUE
STAFF PICKS



FILTERSHAPER CORE
ESSENTIAL FILTER WITH CUSTOM MODULATION

STAFF PICK

"When I was making drum & bass, the biggest struggle I faced was achieving interesting and complex movement in my basslines. Cableguys' intuitive modulation curves make achieving these results quick and effortless. ***FilterShaper Core*** is the plugin I wish I always had, I'm recommending it to everyone I meet!"

Gareth Halsall
General Manager



FilterShaper Core combines the great-sounding filter technology of the full FilterShaper 3 with a streamlined modulation system. Using the same acclaimed Sallen-Key filters as FilterShaper 3, you can expect all the richness, warmth and intricacy of a high-quality analogue filter.

Running inside Cableguys' exclusive new ShaperBox creative effects rack, you can stack up FilterShaper Core with other Cableguys Shapers to create elaborate, multilayered effects for your productions.

cableguys

PLUGIN BOUTIQUE.COM
THE PLACE FOR MUSIC PLUGINS

Welcome





... **to Ableton Live 2017!** This is a special issue of *MusicTech* dedicated to the incredible music making marvel that is Ableton Live. It used to be that Ableton or Live (you decide) was the number one piece of software for music producers who just wanted to make music on a laptop and nothing else. But with the trend of hardware returning to music production and, of course, the increased use of iOS devices, Live has matured into something of a hub for everything – hard or soft. With Link and this issue's all-new Live tutorials (starting on p34) you can, well, link pretty much any type of gear into your Live set-up making it quite possibly the most exciting set-up around for music production. As well as that we have another installment in our beginner's Lazy Guide to Live (although not just for beginner's I hasten to add) so there is more than enough workshop action on these pages. The bottom line is that we want you to make better music with whatever technology is at your disposal and hope that this issue will join the dots of it all to make one seamless studio. I hope you enjoy the issue. Happy music making!

Ableton Live has matured into the hub of everything – hard or soft

Andy Jones Senior Editor, *MusicTech & MusicTech.net*
Email: andy.jones@anthem-publishing.com
(And contact me to Show Off Your Live Studios! See page 18)

MusicTech FOCUS

MUSICTECH FOCUS MAGAZINE
www.musictech.net
Anthem Publishing Ltd
Suite 6, Piccadilly House
London Road, Bath BA1 6PL
Tel +44 (0) 1225 489984
Fax +44 (0) 1225 489980
editorial@anthem-publishing.com

Senior Editor Andy Jones
andy.jones@anthem-publishing.com

Art Editor Debra Barber
debra.barber@anthem-publishing.com

Digital Editor Andy Price
andy.price@anthem-publishing.com

Multimedia Editor Alex Holmes
alex.holmes@anthem-publishing.com

Business Dev. Manager Di Marsh
di.marsh@anthem-publishing.com

Contributors
Owen Bailey, Mark Cousins, Alex Holmes, Matthew Mann, Dave Gale, Mike Hillier

CEO Jon Bickley
jon.bickley@anthem-publishing.com

Managing Director Simon Lewis
simon.lewis@anthem-publishing.com

Art Director Jenny Cook
jenny.cook@anthem-publishing.com

Licensing enquiries Jon Bickley
+44 (0) 1225 489984
www.anthem-publishing.com

Printed by Polestar UK Print Ltd
Tel +44 (0) 1582 678900

Distributed by Marketforce (UK) Ltd
The Blue Fin Building, 110 Southwark St, London SE1 0SU Tel +44 (0) 20 3148 3300
Subscriptions to MusicTech Magazine
Tel +44 (0) 870 444 8468
Price (12 issues) £59.99
UK basic annual rate

All content copyright Anthem Publishing Ltd 2016 and 2017, all rights reserved. While we make every effort to ensure that the factual content of *MusicTech Focus* is correct we cannot take any responsibility nor be held accountable for any factual errors printed.

Please make every effort to check quoted prices and product specifications with manufacturers prior to purchase. No part of this publication may be reproduced, stored in a retrieval system or resold without the prior consent of Anthem Publishing Ltd.

MusicTech Focus recognises all copyrights contained within this issue. Where possible we acknowledge the copyright holder.

PPA independent publisher awards 2014 winner – Independent Publishing Company of the Year

ANTHEM PUBLISHING PPA

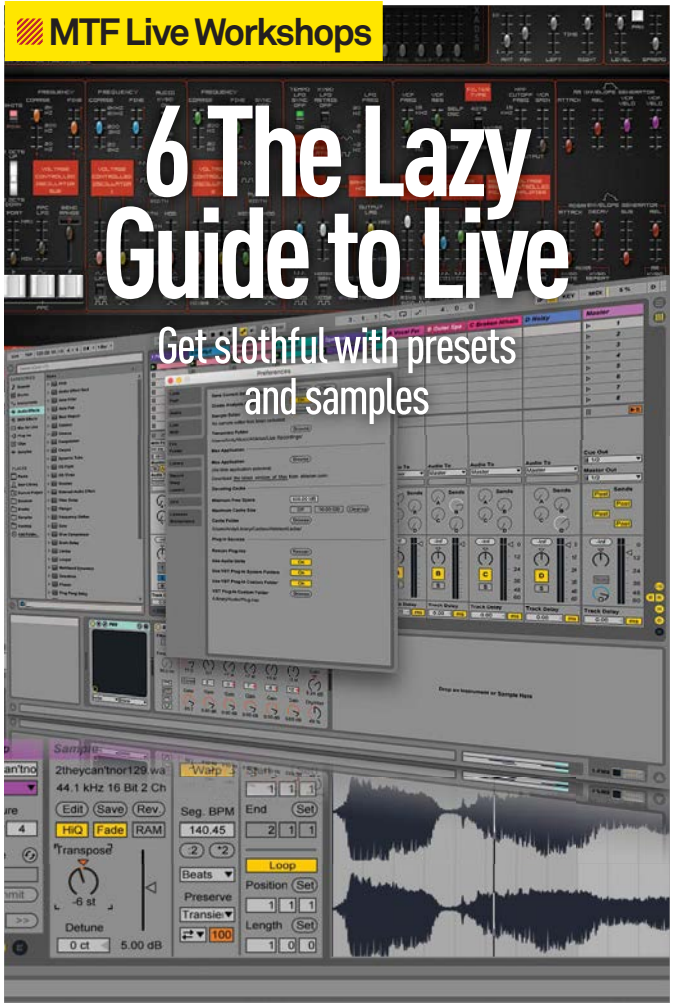
TEXTING MUSIC TECH TERMS AND CONDITIONS
By entering *MusicTech* competitions you are agreeing to receive details of future promotions from Anthem Publishing Limited and related third parties. If you do not want to receive this information please text 'NO INFO' at the end of your message. Texts will be charged at the specified price plus one message at your standard network tariff rate.

MusicTech FOCUS

Issue 46 Ableton Live 2017

THE ULTIMATE GUIDE TO ABLETON LIVE! POWER UP NOW!

Ableton Live can truly become the hub of your music making set-up whatever you use, be it hardware, iOS or software. We have tutorials to help you make the most of it in any situation plus features on freeware, mastering and mobile music making. Then there is all the new gear you need for you Live studio on test. POWER UP NOW!

















MTF46 Live 2017 Full listings

WORKSHOPS

006 | The Lazy Guide to Live How to cheat with presets and samples!

034 | Making the most of the Sine wave Create with this versatile wave

038 | Using Link to sync your computers and OS devices

042 | Put your projects on a diet and reduce your track count in Live

054 | Integrate Live with your iOS devices for control and music

058 | Control your hardware drum machines and synths with Live

062 | Attack, Decay, Sustain, Release what ADSR can do for you

FEATURES

016 | *6 Of The Best:* Software instruments

018 | *Show Off Your Ableton Live-based Studio*

022 | *Go mobile:* with our iOS guide

066 | *The New World Of In-The-Box Mastering* for on-line

080 | *6 Of The Best:* Headphones

082 | *Freeware 50:* The best free software for your Live rig

INTERVIEWS

046 | *Live interview:* Highasakite

076 | *Live interview:* Stoni

092 | *Live interview:* Apothek

SUBSCRIBE!

096 | Get a great deal here

MTF REVIEWS

100 | Spectrasonics Keyscape multi-keyboard instrument

104 | Big Fish Audio Cyborg terror on your hard drive

106 | Output Movement plug-in

108 | Wave DNA harmony helper

109 | Sonic Academy Kick 2 Get your kicks here

110 | ROLI RISE 49 unusual multi-touch keyboard

112 | Dynaudio Lyd monitor range

114 | IKM Miroslav Philharmonik orchestral instrument

116 | Softube Modular synths

118 | AKG K-Series headphones

120 | UA Fender Tweed guitar sim

121-128 | *Mini Reviews*

130 | *On your MTF DVD*

Ableton Live | A Lazy Guide

Level | Beginner

The Lazy Guide to Ableton Live – Preset and sample busting

It's another installment of our Lazy Guide to Live. This time around we detail how to carry on being lazy with Live by disguising those synth presets and samples and how, with just a few simple tricks, you can easily call them your own…

Welcome to our world of Live laziness, but don't feel bad about entering it as all of us technology users are inherently lazy – that's what technology does to us. It takes the slog out of doing 'stuff'; it's just in our case, that 'stuff' is music. DAWS like Ableton Live are sitting at the very cutting edge of music technology and are here to take the strain out of music making so, really, you're on a bit of a lazy tip already, admit it!

However, this tutorial is designed to take even more strain out of using Live. You may think it's easy enough already but we're here to guide you through some tips and techniques that you may be unaware of that will help make your compositions better. Ok easier...

It's not the first time we've embarked on a course of slothful DAW use. In the last Ableton Live Focus (issue 44 available from www.musictech.net) we introduced Templates and the Virtual keyboard; got creative with clips; and then moved onto Follow Actions and Clip Automation. Those are standalone guides and you needn't have read them to follow this one – each of our Lazy Guides has tutorials that work completely on their own.

You might think that Ableton Live is **easy enough to use already** but this tutorial is designed **to make it even easier**…



FOCUS ON EASY TEMPLATES

You know that bit in the main text where we said that you didn't need to have read our first Lazy Guide to do this one? Well we were stretching the truth… a little. In that tutorial we discussed Templates and how useful it is to have a Live Template loaded up with all of your favourite instruments and effects placed across various tracks. As one of our Lazy tutorials concerns them this time around, here's a brief recap on how to set a Template up in Live if you don't already follow this easy practice. So start a project from scratch, fill it up with your most used instruments and effects. You can even have a selection of beats loaded in there too – whatever you want basically. Then select Live's Preferences and go to the File Folder tab. Check Save Current Set as Default and there you have it, whenever you load Live up this set will come with it…

MTF Step-by-Step 1. Easy preset busting



01 Looks it's our new Template File – aptly called Live 2017. Templates are great but after a while those sounds can become a little tedious...



02 We're going to change some of our favourite presets fast so we can still keep using our Template File but have options to quickly change its contents. The first overused sound comes from FM7...



03 It's a pad sound and to quickly make it ours we're going to add a Resonator effect – these will be good to change any pad sound. Try the first one, called Berlin, for size....



04 Upping the Decay dial and third Pitch control really increases the resonance effect but still keeps the essence of the original preset. However it's a preset no longer. It's your sound!



05 All of the Resonator effects are named after different European cities which is a nice touch. We found the Vienna one to be the best for our pad sound as it adds a string essence up top.



06 Vienna is a great one to add to a string sound too, to get that eerie distorted fret sound so beloved by composers like Thomas Newman. We moved the Pitch parameters up to result in it almost taking over the sound and the bulk of what you hear is the effect.

That first Lazy Guide was very much utility and creativity based – how to use the software's easy features and time savers and how to get a little creative with clips and beats. This time around we're very much focussing on the sounds and samples, and a lot of the following workshops utilise Live's built-in effects – those that come with the software so don't worry about shelling out on third party plug-ins.

These effects are sometimes overlooked in Live, possibly as they look a little samey and utilitarian, but actually they are many and varied and, like a lot of Ableton's plug-ins, they do the job incredibly well – they just might not look that flash.

So it's time to get bone idle – come join us for several tutorials to make your Live composing even easier...

1. Easy preset busting

There's always been a voice in our heads when we use any software or hardware instrument that says 'don't use presets!' It's probably the same one that says 'don't use samples' and we do, largely, tend to ignore it. After all someone has taken ages to program these sounds, so it would be churlish of us not to use them.

However after following our own advice from the previous Lazy Guide and having the same Template open whenever we start a new tune – see two pages back for a quick guide – we did find ourselves starting to tire of the same presets, so started exploring a number of different ways to subtly, and not so subtly change them... quickly.

So this tutorial not only shows you how to quickly twist your Template away from the last time you used it, but also helps you turn preset sounds – that the voice is still telling you you shouldn't use – into something you can suddenly call your own. And there's no synth programming needed,

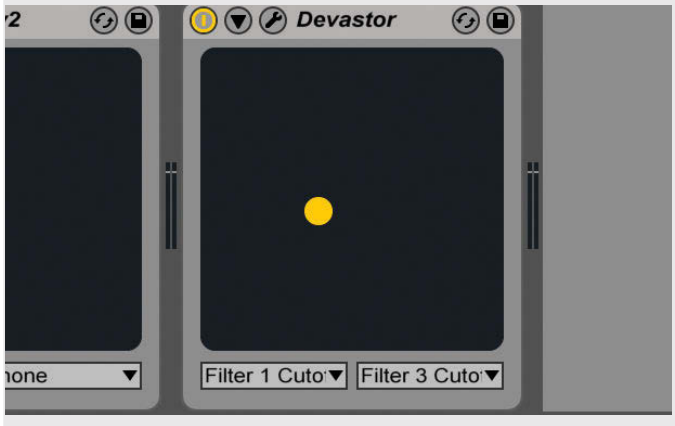
MTF Step-by-Step 1. Easy preset busting (cont'd)



07 We'll stay with strings and pads for another couple of steps as there is another neat – if less subtle – effect that you can use to alter presets, this time from our trusted Oddity.



08 It's another simple-looking effect but Devastor can have a, well, devastating effect on your sound, adding lots of high-end screech. Don't worry, it sounds a lot better than that sounds!



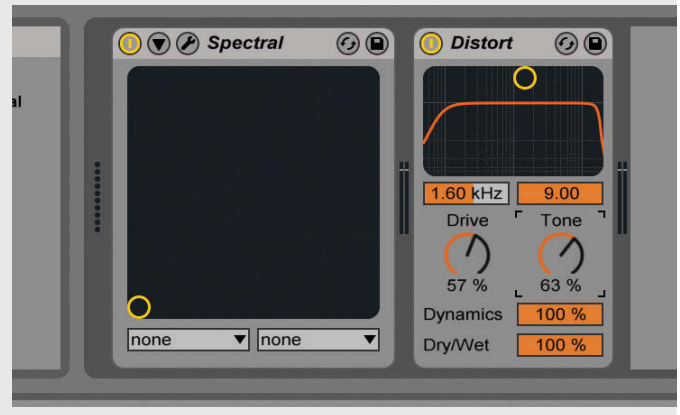
09 And if you think that is far from subtle, employ Devastor's X-Y pads to introduce parameters to adjust in real time. We've chosen the Cutoff from both Filter 1 and 3 for maximum drama. .



10 Now onto bass lines, but even simpler – arpeggiated bass so you don't even have to program the tune. Lazy! We love this Arp bass preset from Spectral called ym Bulky-Arp. a little too much...



11 We could simply adjust Filter Resonance and Cutoff to drastically alter it but to really make a bass sound like this ours, we'll add Distort from the Overdrive plug-ins that instantly gives it a 303 edge.



12 It's a little bit too full on to be honest so we notch the Drive parameter down to 57%. The Tone control emphasises the tones and low and high frequencies and can be quite dramatic too.

just a few of Ableton Live's effects.

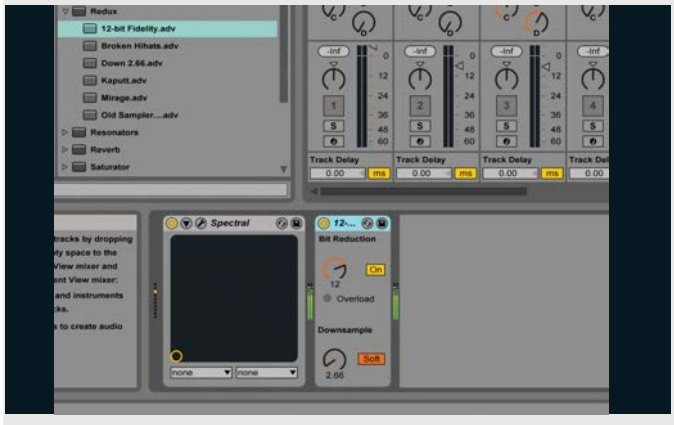
We start with pad and string sounds and how the Resonator effects can lift them and swirl them, and how with just a few slight pitch changes you can take the preset up a notch but still leave its core character intact.

Still staying with pads and strings we demo the Devastor effect which is nowhere near as subtle as the previous one and does rather take over the sound but we love using it on dreamy pads to make them screech.

We then switch to bass lines, or more lazily (of course!) an arpeggiated line and discover which effects can give these some or your own identity. The Overdrive effects here are great as are the bit crushers although the latter ones can be a little too dramatic

Finally we move on to lead sounds and, yes, you can get away with using the same Resonators that you used on the pads and strings – albeit with the Dry/Wet dial lowered – but we think the Vocoder folder offers all you need for everything →

MTF Step-by-Step 1. Easy preset busting (cont'd)



13 We'll stick with arpeggiated bass and move onto a different effect to keep its essence, and that is 12-bit Fidelity from the Redux set. Rather than crushing the bass sound it gives it a pleasant high ring.



14 Not much to alter in 12-bit Fidelity but if you sweep the Bit Reduction down it tends to add more hiss until you crush it to 2 then it goes crazy. Great sound but unsubtle and unstable!



15 So that's pads, strings and bass dealt with and easily altered, so now we move to leads and it's back to the Oddity with the rather shiny lead called Tamtims...



16 You'd be right in thinking that we can use some of the effects we used on the pads and strings and indeed the Resonators work but do reduce the Dry/Wet mix from the amount you had on the pads.



17 But actually the effects that are well suited to adjusting leads like this are Live's Vocoders. Start with the Formant +5 and adjust the Formant rotary as it still sounds a little too like the original.

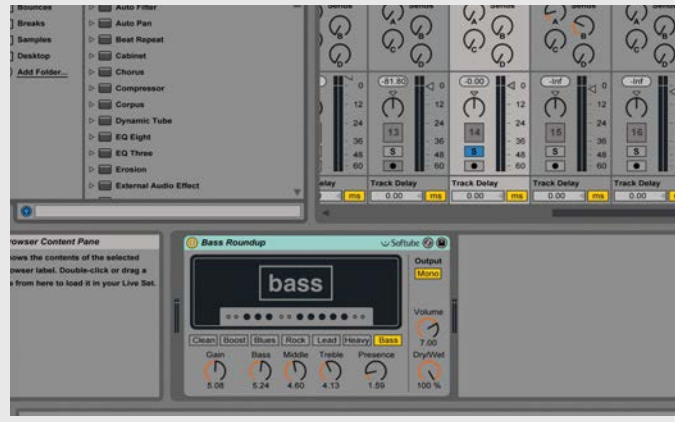


18 Moving further away from the original lead and right back into pad territory try the Chromatic Vocoder. Again the Dry/Wet dial is your friend. The more you put in the 'paddier' it gets!

MTF Step-by-Step 2. East sample busting – beats



01 So we've messed with your presets and now it's time to disguise some samples. First we'll start with beats so load in one of your favourite loops, maybe one you over use but want to retain its groove.



02 We'll start with a fairly subtle effect, and Amp sim. The key is to retain some of the key bass elements to underpin your track and the Bass Roundup setting does this well by adding low end.



03 In fact we're going to use this amp as the basis of all of our effects as it easily allows you to bring more bass, mids or trebles back in – as shown – if more extreme effects dampen them, later on.



04 With the extra lows supplied by the Bass Roundup amp, start messing with some more effects knowing your lows can be made more solid to retain the groove. Now load the Chorus plug-in.



05 If you now mute the Bass Roundup you'll hear the Churus in action and it's reduced the kick of your original loop. Bring the BassRoundup back in and it's back, just what we wanted!



06 So now whatever extreme effects we use – in this case by really dialling up the Feedback rotary on the Chorus – the Bass Roundup retains some low end groove, arguably the most important part of the original sampled loop.

→ from subtle to not so subtle.

2. Easy sample busting – beats

As we've already hinted at, one of the other easy ways to make music – aside from using presets – is to use samples, and there's no other piece of software that makes them as easy to incorporate into you tunes than Live – Live and samples are the lazy person's dream! Andy why not use them we say? However, it's that voice again. 'Using other people sounds again are you?' it asks.

So if you have any guilt whatsoever about using shop bought samples – and you shouldn't (illegal ones on the other hand...) – then this part of the Lazy Guide is for you. This and the previous part on disguising presets are, if you like, The Lazy Guide To Continue Being Lazy with Live, but that title simply wouldn't fit on the first page.

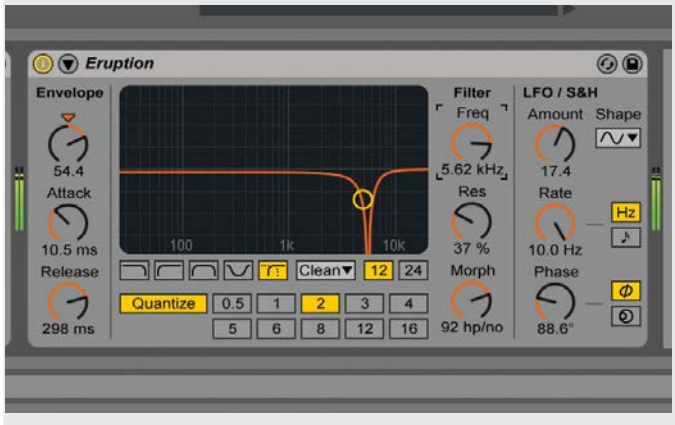
MTF Step-by-Step 2. East sample busting – beats (cont'd)



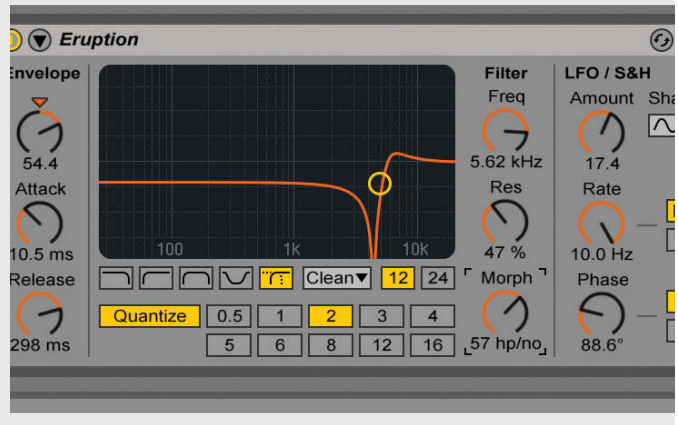
07 Early samplists used filters to disguise breaks that they sampled from records and there's no reason why you can't do this in Live to disguise your samples and make them yours. Live has loads of filters...



08 The Auto Filter can be a little 'Marmite' – it has been very popular but is very unsubtle in many cases. The Eruption preset is more subtle and a good starting point to start some gentle manipulation.



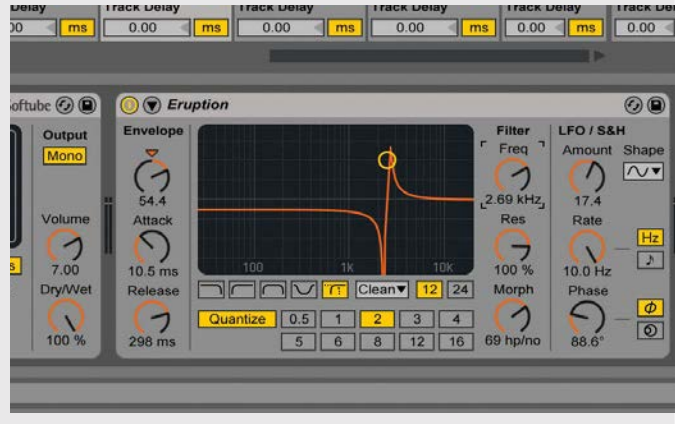
09 Again we want to retain the kick which is being underpinned by the Amp plug-in so adjust the Freq so that the notch moves up to around 5.5kHz as shown. Shouldn't sound too different yet...



10 Now move the Morph dial down. You'll note the bass starts going down so don't move it too far – not even your Amp will be able to save you if you do! Try a value of around 57.



11 You'll see and hear what we're doing now as if you move the Resonance dial up, a specific frequency with stand out and add a great percussive edge to your loop at that part of the frequency spectrum. Effectively we're using it as a precision EQ only it's easier!



12 You can now use this method to home in on specific frequency areas in your drum loop – here we've nudged it to 2.89kHz and take the sound right away from the original but retain the groove underneath (unless you move the Morph dial down further, that is!)

It is, of course, very easy in Live to mess your original sample up, to disguise it by heaping stuff like the Grain Delay and Distortion onto it, or simply Overdriving it to hell and back. However what this tutorial is designed to do is retain the original groove so we need to be a little bit subtle about how we add effects... but we'll still keep it easy.

This is because the reason you want to use that sample in the first place is how its groove fits yours or even supplies it, so while you can heap effects aplenty on it so no-one will know what it originally was, you still want to hear the very essence of what made you choose it and use it – or over-use it – in the first place.

So we'll start by introducing an Amp sim plug-in that we'll introduce at the start of the effects chain that easily lets you →

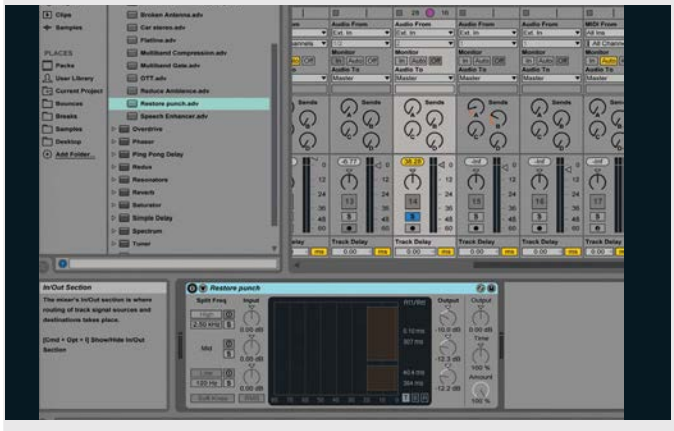
home in on bass, mids and trebles. In fact you could use an EQ here but the BassRoundup makes it even easier to do this (and the is The Lazy Guide).

It also adds some bass frequencies because, as you'll see, as soon as we get into using more extreme effects like choruses and filters, you'll lose certain frequencies and need easy controls to try and wrestle them back in.

We'll then add a chorus, again subtle at first, to show how the groove is retained when you make it unsubtle. Next up we add an Auto Filter and, using a combination of the Morph, Freq and Res dials, show how you can easily home in on specific percussive elements within your sampled drum loop to make them sound more extreme and different while, yet again, retaining the bass and groove.

Next up we discuss how you can retain the groove to work on with the Multiband Dynamics plug-in where we can strip away most of the tops and upper mid frequencies and only retain some of the lower mids and kicks so again you get the

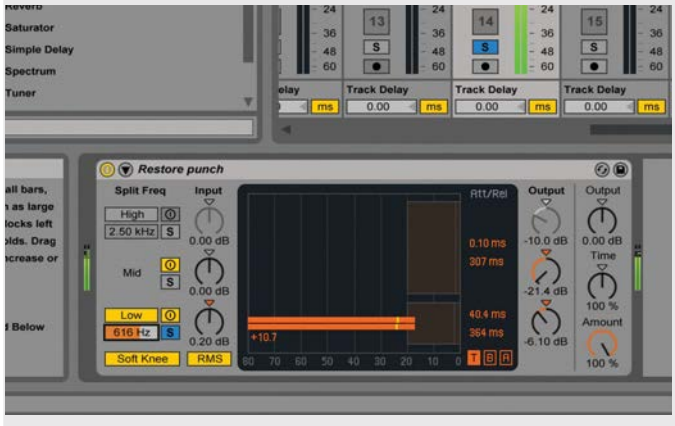
MTF Step-by-Step 2. East sample busting – beats (cont'd)



13 Now we'll show you a very simple way to get to the groove of the original loop you have sampled and show you how you can almost strip it back to its fundamental beats using a Multiband Dynamic plug.



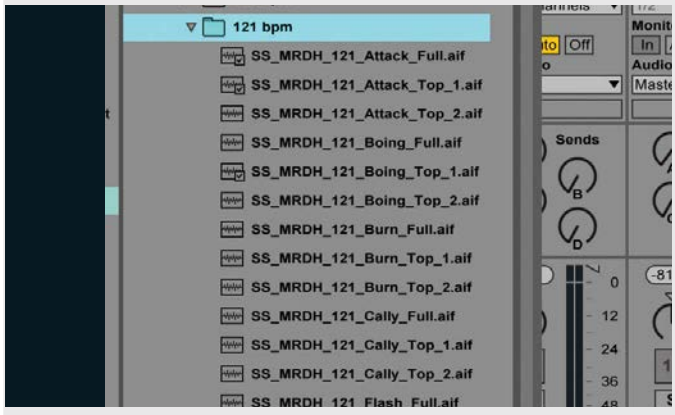
14 Load in Restore Punch – the clue is in the name. If you bypass it you can already hear that it removes a lot of the extraneous percussion, leaving you with the backbone beats.



15 We can even home in on the lower frequencies by muting the mid part and lowering the frequency of the remaining LF to, in our case, just over 600Hz although it now sounds a little muffled.



16 Experiment with these two parts (Low and Mid) to see if you can keep the groove of the beat but lose any distinguishing percussive parts.



17 Loads of sample packs now come with the samples broken down from the full loop into the percussive top end and then the kicks so you can easily drop the kicks in and out of a loop.



18 Now you can add one of these percussion top loops – or indeed any loop – to the groove of low-end beats that we have created here to create a completely new loop, that is (almost) yours!

essence of the loop which you can then use to underpin other loops so rather than becoming the main sample, it becomes a less obvious component part of the track.

3. Easy sample busting – vocals

So we've shown you how to disguise beats to varying degrees all the time retaining the groove. The next subject concerns the vocal sample and this kind of audio is a little harder to disguise. Also, if you like a vocal sample you are not going to want to change it too much beyond its original sound... otherwise you wouldn't have chosen it in the first place. However there are plenty of ways to change a vocal sample drastically but in a way that keeps some essence of the original, and hopefully at least part of why you selected it in the first place. So what we end up with is certainly not as subtle as the beat examples but will certainly disguise what you started with.

We'll open with changing some of the common

MTF Step-by-Step 3. Easy sample busting – vocals



01 So we start with our vocal. We have a short clip which we are looping as we try various effects, but we'll start with the parameters within the Sample edit window.



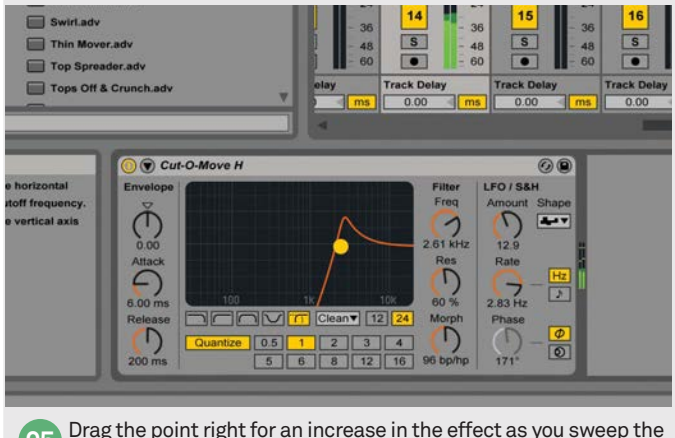
02 The Sample edit window really comes into its own with vocal samples. Ours was originally a female vocal but here we've transposed it down by six semitones to turn it into a male!



03 And if you start with a male vocal, try upping it by 6 semitones to turn it into a female. The sample is all yours now! Don't try taking a female vocal up by six semitones as you'll end up with Mickey Mouse.



04 Load in the Cut-O-Move H Auto Filter. You'll instantly hear a slightly crushed vocal that you can either double with the original for a subtle background vocal effect or use on its own to be less subtle.

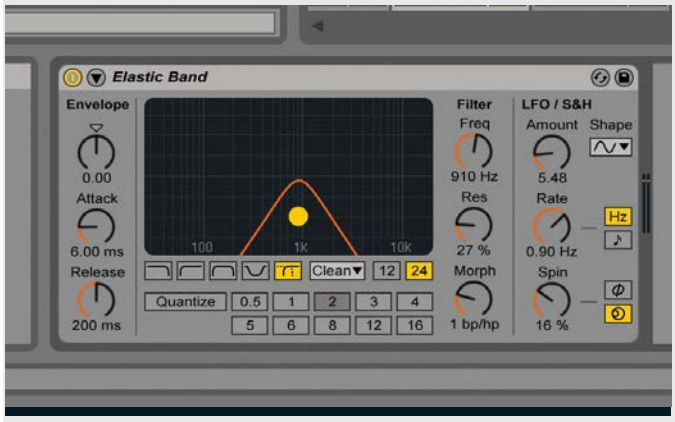


05 Drag the point right for an increase in the effect as you sweep the notch up the frequency range. If you record this as automation you can use this as an effect in a dance breakdown...



06 And then sweep it back to the left as the breakdown builds back up to its impactful return. The further left you drag it, the more like the original vocal it sounds.

MTF Step-by-Step 3. Easy sample busting – vocals (cont'd)



07 While we're on the Auto Filter tip, load in the Elastic Band preset as shown and suddenly your voice, whether male or female, sounds like he or she is on the phone.



08 Nudge the notch upwards here for some extreme vocal effects – less phone, more like they are calling you from some kind of spaceship with bad reception.



09 OK that was a bit extreme so now we'll use a little Chorus to get back to a more subtle vocal effect. Start with the Minus preset for just a little chorus...



10 Move the modulation yellow tab up and right for a more extreme effect – this moves both the Chorus Rate and Amount and can quickly get out of hand.



11 Load in the Gagalon preset one of our favourite Chorus sounds. It treads a fine line between disguising the original but being subtle. Firstly it makes the vocal sound like it's been recorded in your loo!



12 Try increasing the Rate parameter by moving the Modulation ball right. This is far more subtle than just moving it upwards – and increasing the Amount.

→ parameters within the main Sample Edit window. We've ignored these so far and could have easily turned to them for beats too but here, if used sparingly, they can add a little nuance (and a lot of Mickey Mouse!) or even simply change your original female vocal to a male (and vice versa) just by transposing the original sample up or down.

We'll then move on to our friend the Auto Filter to show how you can easily change a vocal for a breakdown and to get that building effect where the vocal returns to its original state as the breakdown comes back for impact. We'll also use the Auto Filter again to change the vocal to a phone-type voice and extreme phone-type voice and then the Chorus effect for some more subtle changes to the original vocal sample while still managing to disguise it.

One of our great finds was one of the effects we used right back at the start of this tutorial: the Resonator, in particular

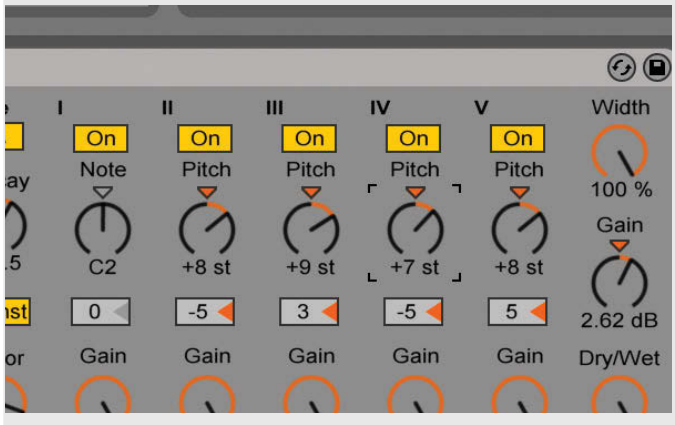
MTF Step-by-Step 3. Easy sample busting – vocals (cont'd)



13 We'll finish up this tutorial with some of the effects you might not realise will help with vocal disguising, and surprisingly we return to the Resonator that was so good at disguising pad sounds.



14 Los Angeles worked brilliantly with our particular female vocal sample adding a beautiful pitch effect that could easily marry it to your main melody while changing it significantly from its source.



15 If you want this to blend in with your exiting melody you'll simply need to adjust the Note dial (here showing C2) and maybe a little trial and error with the various Pitch parameters to get it accurate.



16 Beat Repeat is one plug-in that many a Live user has explored... with beats. Here we put it to work on our vocal sample. The Squishy preset is particularly good.



17 Chopping Block is one for fans of 1970s comics as it cuts various words out of your vocal sample... a bit like when your microphone goes wrong. OK, it's not an 'in-demand' effect...



18 But Robo Voice is an in-demand effect doing what it says on the tin for instant Daft Punk. So we've changed male to female and human to machine, all with some simple effects!

the Los Angeles preset. For our sample – and this will vary from sample to sample – it added a brilliant pitch element, changing the original sound but keeping the vocal itself distinct.

Finally we put Beat Repeat – another plug-in we could have easily used on the beats earlier – to good use on the vocals with the Robot results and more random repeats that you might expect.

Using plug-in presets and third party samples is sometimes frowned upon – wrongly in our books – but is a great way for novice music makers to get started and us lazier producers to get ideas down quickly. We hope that with some of these subtle and not so subtle uses of Ableton Live's effects you will continue to use them – and then abuse them – by easily making them your own. Never fear the preset or sample ever again! Happy laziness! MTF

Six of the best

Hardware ▨ Software ▨ Mobile Technology ▨ Samples

Welcome to the ***MusicTech* Buyer's Guide**, where we round up some of the best products we've recently reviewed. For this first collection we've gone soft with a selection of **software instruments** – from brass, to the evocative gamelan…

BEST Piano (like)

Una Corda



Details
Price £129
Contact via website
Web www.native-instruments.com

Two entries for Native Instruments in our 'six', but the Berliners do like their real instruments. And while we haven't looked at many pianos of late, we still reckon this is one of the best out there. Una Corda is a little different, and offers up a deeply sampled version of a unique piano, hand-built by David Klavins in collaboration with composer Nils Frahm. It only has one string per note, which gives a much purer sound than your average piano, and allows the user to easily slot in different prepared materials to dampen and affect the way the hammers hit the strings. We said "it's been created with love and attention, and the results are often breathtaking. It's an inspiring instrument that oozes class and with very few tweaks, can be morphed from a gentle, upfront piano to a swirling, dissonant and clunking beast. ***"An effortlessly atmospheric and beautiful prepared piano instrument."***

BEST Brass collection

NI Brass Ensemble



Details
Price Ensemble: £339; Solo £249; bundle £429
Web www.native-instruments.com

You get two instruments – Brass Ensemble and Brass Solo, which you can buy separately or as a bundle. Together, it's a hefty 70GB install, but, as we said, they are "a formidable addition to the tool kit of any composer, with Ensemble capable of huge, stirring and epic orchestral soundscapes. They're also good for regular music production, for adding an orchestral layer to rock or pop… but it's probably for scoring that they're most perfectly suited. ***A rich-sounding orchestral brass collection with great articulations and wide stylistic range."***



Details
Price $399
Contact via web
Web www.samplelogic.com

BEST Atmosphere

Sample Logic Gamelan

On the face of it, a 14GB Gamelan instrument might not be your number-one choice to take up space on your hard drive, but Sample Logic has combined sampling and synthesis with some of the most advanced Kontakt instrument design around to produce a real monster of a sound-design tool. And with the Gamelan being one of the most evocative instruments there is, you have a highly useable and inspiring piece of software. "There's depth and movement, endlessly flexible rhythms, textures and atmospheres", we said in our review and concluded: "Don't be fooled by the name, there's much more to Gamelan than meets the eye. ***This is a versatile and powerful cinematic and sound-design instrument, with plenty of controls."***

BEST Retro keys collection



Big Fish Audio Vintage Collection

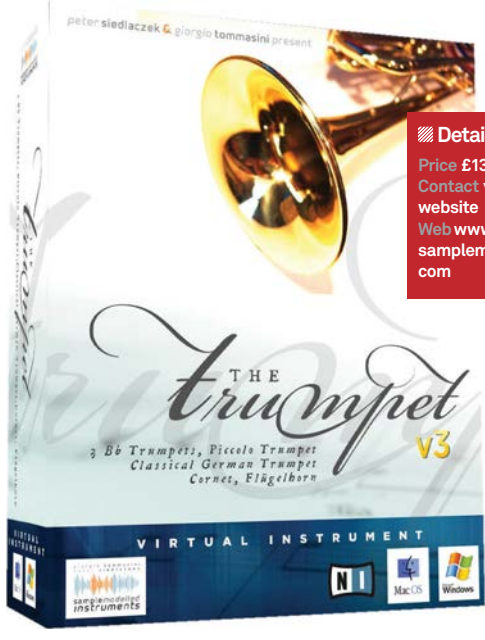
Details
Price £280
Contact Time+Space 01837 55200
Web www.timespace.com

The Vintage Collection combines three of Big Fish Audio's virtual instruments – Vintage Horns, Strings and Rhythms – which, when bought together, represents a saving of over £130. It can work as a standalone or plug-in product and weighs in at around 16GB. We said: "This is a great collection and achieves the aim of bringing warmth and a retro feel to a production, as opposed to the mechanically perfect quality you sometimes get with sampled instruments. The Rhythm section is the most diverse of the three collections, but all three hang together very nicely. ***A very impressive collection of instruments for anyone wanting a more authentic, old-school flavour for their productions."***

> "It's a **very impressive collection** for anyone wanting **an authentic flavour**..."

BEST Trumpet

Samplemodeling The Trumpet 3



Details
Price £139
Contact via website
Web www.samplemodeling.com

For a solo trumpet instrument, it doesn't get any better than this. The Trumpet 3 is more flexible, because it can vary its timbre with real-time timbral shaping. For purity, it was recorded in an anechoic chamber, without ambience. However, by using spatial information collected from the multi-microphones used, an early reflection impulse response has been developed for a richer sound. It's also an incredibly accurate and realistic recreation, over a huge variety of styles. When we reviewed it back in issue 153, we said: ***"With a number of significant improvements, The Trumpet 3 continues to hold the virtual trumpet top spot. It sounds great, is enjoyable to play and has a wealth of controls."***

BEST Classic organ



Details
Price £95
Contact Time+Space 01837 55200
Web www.toontrack.com

Toontrack EZ Keys Pipe Organ

EZ Keys' instrument range is becoming quite the one to plunder for 'real' instruments like this, a rather superb pipe organ. The man behind the sampling, Björn Gäfvert, himself an organist, chose to play and record the organ installed in the Täby Kyrka, a historic church dating back to the 13th century and the results are, as you'd expect, brilliant. We said: "One of the best pipe organs that we have come across, especially at this price point. It's extremely gratifying to play, mainly because of the extremely clear sound. ***A majestic-sounding pipe organ, with a beautifully clear sound and suitable for many types of music."***

Show off your Live studio (and dog)

Where our readers show off their Ableton Live-based set-ups. **Show Off *Your* Studio** via the ***MusicTech*** **Facebook** page and you could take the top slot here (pets in shot certainly help, too!)



Now *that* is what we call komplete kontrol

Driftmoon

Contact: **driftmoonmusic@gmail.com**

MusicTech: Tell us about your studio…

Technically, my company is called Driftmoon Audio Suite, but the studio is called Double Helix Mastering. I use the studio 50 per cent for my own music composition under the Driftmoon alias and 50 per cent for mixing, producing and mastering for my clients. I didn't expect the studio to actually be so time-demanding, but it's proving to be an amazing experience. Working with so many different clients and musical genres has opened up a world of new possibilities to me.

MT: Where, when and how?!

I've totally refurbished and reconstructed my home to be a full-on studio. I know many people prefer to have a studio somewhere else, but I love the fact that when I have an idea for new track or melody, I can write it down immediately. Also, it's good to be available if clients need something to be done as



soon as humanly possible. The music scene is a rapid changing world, so it's good to stay ahead of any potential problems. Also, considering that I live in the heart of Prague right next to the most visited tourist locations, finding some place just for a studio setup would be extremely difficult.

MT: Kit list?

Mac Pro with 12-Core Xeon, 1TB SSD and 128GB RAM; Universal Audio Apollo Quad and UAD-2 Satellite Quad; Waves MaxxBCL; API 5500; Kush Clariphonic; SSL G Compresor; SPL M/S Master and MasterBay; Focusrite ISA One; Mutec MC-3+ Smart Clock; RTW TM3-Primus; 2x Event Opal; 2x Avantone Mixcubes; Roland JU-06 and JP-08; Dave Smith Instruments Sequential Prophet-6; Native Instruments Komplete Kontrol S88; software from Waves, UA.

MT: Which DAW?

For composing and production, I →

Driftmoon (cont'd)

→ use Ableton Live. I somehow love the workflow and even for orchestral composition, it's absolutely amazing. Granted, it doesn't have as many features as, for example, Cubase when it comes to composing, but you can work around them and find your own 'way' of doing things. For mixing, I use Pro Tools and for mastering, Sequoia.

MT: **What is your favourite gear?**
I recently purchased a Dave Smith Prophet-6 and, oh my, it's absolutely phenomenal! I've spent days playing with the sounds and writing them down in an Excel sheet to categorise them and then I immediately started reworking my album tracks, because the sound of the Prophet is amazing. It just stands out.

MT: **How often are you in your studio?** I am here daily for 10 to 16 hours – it depends on how much work there is to be done. I stopped having weekends years ago, now every day is a work day, but every day is amazing and different. I somehow got used to non-linearity and it's making every day interesting.

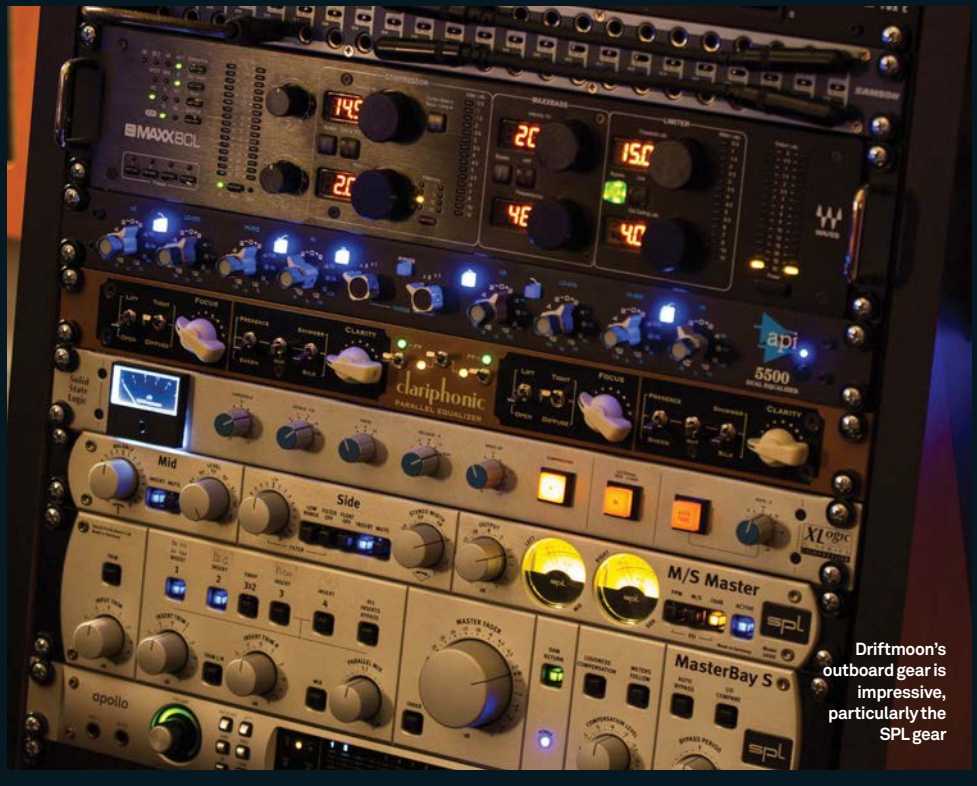
MT: **How do you use it?**
My studio space consists of 'only' one room, so there isn't enough space to record a whole band: nevertheless I can record singers and instrumentalists with ease. I mostly use the studio for producing, composing, mixing and mastering.

MT: **Next on your shopping list?**
Right until now, I've had no need to go into the 500 Series world. But after loads of positive reviews about Meris' Mercury7 Reverb, I think I'll start collecting different 500 Series modules. I've also tested the Moog Ladder Filter module, it's incredible.

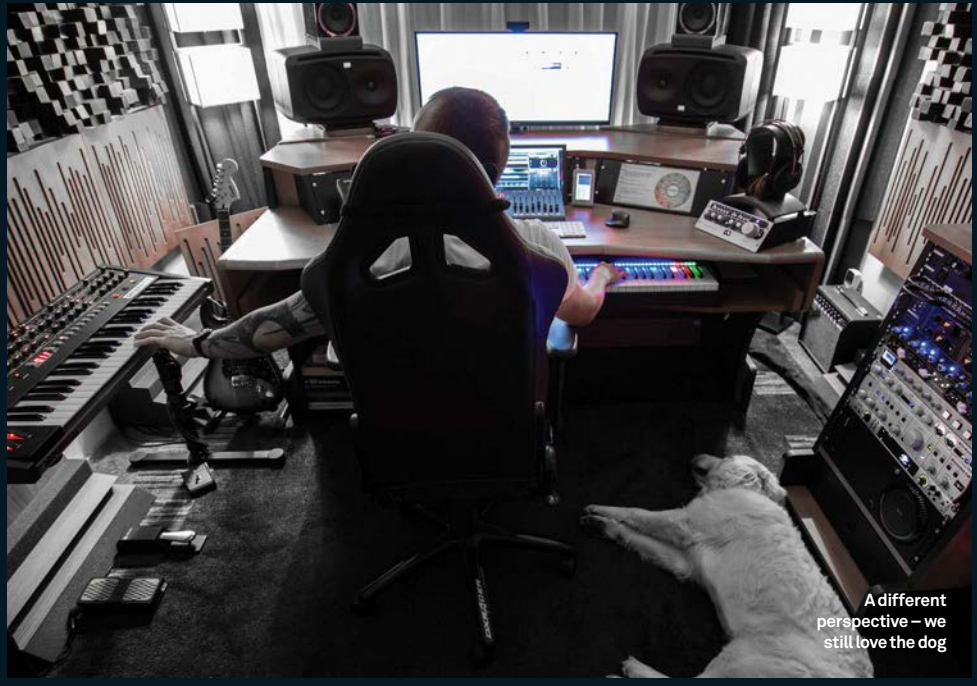
MT: **Does anything annoy you about your studio setup?**
Loads of time and effort (not to mention money) went into this space. I've had room to work with and I made it work. The room itself is treated to industry standards. If I could change one thing, it would be the air conditioning. However, because the studio is located in a very old building which is under UNESCO protection, I can't have an AC unit here.

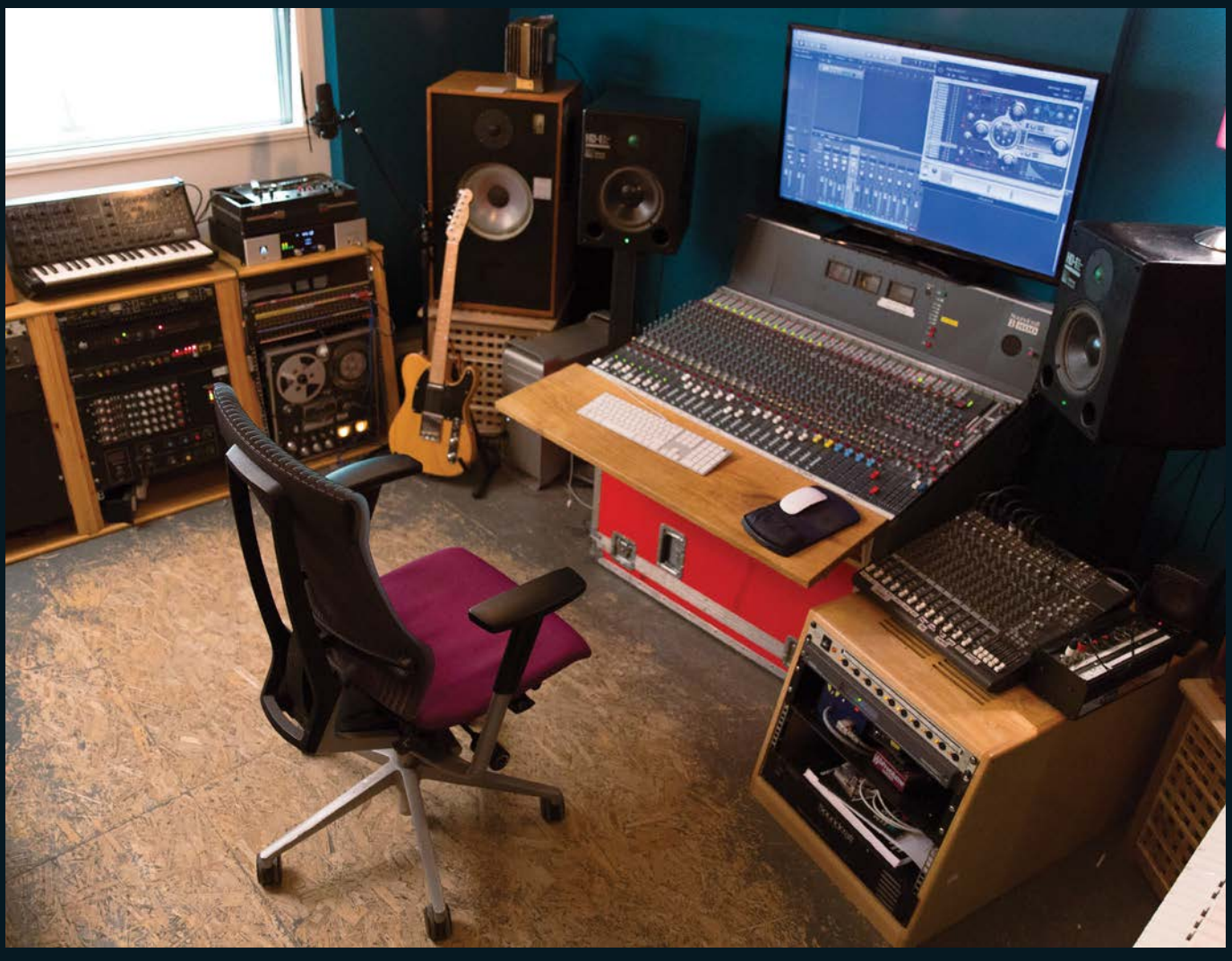
MT: **Any dream gear?**
Barefoot monitors, I really hope one day I will be able to get a proper space which will be treated so I could buy and use these. So far, they are huge for my current space but it never hurts to dream, right?

MT: **Do you have any advice for MT readers about setting up a studio?**
To try and remember that we all started with something little. Whether or not you will make something great out of it depends only on you. Listen to as much music (in different genres) as possible and also to be kind and respectful to other musicians.



Driftmoon's outboard gear is impressive, particularly the SPL gear


A different perspective – we still love the dog



A big desk, big speakers and big screen make a big studio...

Secretsundaze Studios

Interviewee: **Luke Hopper** Contact: **0208 525 2838 (London, UK) www.secretsundaze.net**

MT: Key components? 24-channel B800 broadcast desk with Penny & Giles faders, Studer preamps, AMEK/Neve channel strip, preamp and EQ plus Neve Pure Path; 1968 Drawmer MkII Tube compressor; Audio & Design compressor; Akai Valve tape; WEM Copycat delay.

MT: Which DAW? Logic Pro X and Ableton Live. We use a combination of both, suited to our artists' needs.

MT: Favourite gear? Probably the Akai Valve tape machine – a perfect bit of kit to make things sound full and old, which we love to do. Just a little touch is enough to make a difference. It features valve in and valve out, to provide a warming distortion that sounds amazing.

MT: How do you use your setup? It's a professional working studio, as well as the creative home of our in-house artists, engineers and producers: George Levings (Endian/Commix), Tony Nwachukwu (CDR/Attica Blues) and Sam Jones (Soundthread). We offer dry hire, hire with an engineer, mixing and mastering plus courses and training. We have projects going on all the time with clients, artists and students taking part in classes and workshops. We also have our monthly 'Out The Box' feedback sessions for unsigned producers, run in collaboration with CDR, and we're going to be hosting regular artist masterclasses, too.

MT: Next on your shopping list? An MPC for the studio – great for sampling and the groove is amazing. We'd be keen to get a Eurorack going to create out-there sounds. We also love the new Eventide multi-FX unit.

MT: Anything annoy you? Working with older gear can be a little tricky, but it's all worth it when you hear the character and warmth of the sound.

MT: Any advice? Take your time and be sure to choose the right setup for you, and shop around. MTF



MTF Feature iOS music making

Go Mobile

What iOS apps and interfaces do we need to replace our favourite 'real' music production setups? **Martin Delaney** rounds up the best hardware and apps and says it's all about the mix and match...

iOS music production has reached a certain level of maturity, it's true; but it's more like a teenage level of chaotic development, with hormones and zits all over the place! Some organising and filtering is needed and in this guide we'll highlight the apps and iOS-compatible audio interfaces that we think represent the current state of the art in iOS music. We'll also suggest setups based around combinations of these, putting together apps and interfaces in different combinations to replicate – or attempt to replicate – existing big-platform DAWs. The primary test device used for this was the iPad Pro, the 12.9" version, with the iPhone 6 Plus also coming into play. Both are slightly bigger-than-average iOS devices, and that's not a coincidence; every square cm of touchy glass goodness helps when you're working with music apps, and their sometimes-crowded interfaces. For any serious music use, I'd say an iPad is vital – most of these apps run on iPhone as well, but in terms of performance and interaction you just won't be getting the full benefit of the iOS music experience.


MANAGE SCENES
AnotherThought
FilterAuto
Group
realtime
Current scene: AnotherThought
Universe
iMASCHINE 2
Dopeness
Pattern 3
No Pattern in Scene
Pattern Length
Clear
Double
Step Mode
Erase
Epic Poly
Keyboard
Knobs
Routing
Keyboard Modes
Portamento
Trigger
Master Volume
thor
Sleep

iOS DAWS

For every computer-based studio, there's a **Digital Audio Workstation** – the one application that handles the bulk of recording, composition, and mixing. In iOS Music Land, however, things are more fragmented, and it's not likely that you'll find one DAW that does it all, although a well-chosen combination of two can usually cover everything...



Auria Pro blurs the boundaries between apps and DAWs

AURIA PRO

Price **£39**

Contact **www.wavemachinelabs.com**

Auria Pro is the best-specified iPad DAW around, with a featurelist worthy of any computer DAW – recording up to 24 tracks of audio simultaneously (there's no theoretical project track limit), audio warping, transient slicing, waveform editing, sample rates of 44.1, 48, and 96kHz, 24-bit audio, MIDI recording and editing, quantisation, groove templates, automation, MIDI processing, and plug-ins including the Lyra sampler which supports the EXS format as well as disc streaming, for multi-gigabyte sample playback. Auria Pro also includes a suite of plug-ins, including a rather nice PSP channel strip with EQ, compression, expansion, and gating. There's also a convolution reverb, a handful of other effects including stereo delay, and the FabFilter One and FabFilter Twin 2 synths are included. The mixer includes six aux sends, and PSP have also supplied a master strip with EQ, compression, and limiting. Extra plug-ins are available as in-app purchases including PSP Micro Warmer, FXpansion's DCAM BusComp, Positive Grid's JamUp guitar amp, and even Drumagog! Other purchasable items include IR files, video-import functionality, and various loops and samples. On the control front, Auria Pro is MIDI mappable, and supports remote control via MCU and HUI. This is as full-blown as iOS DAWs get.

KORG GADGET

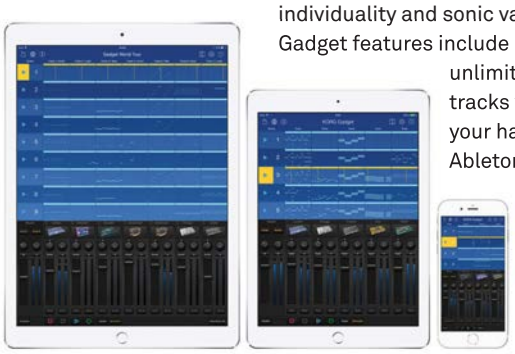
Price **£29.99**

Contact **www.korg.com**

If you're more concerned with electronic music, and you're still wishing Ableton would pull its finger out and release Live iOS, your best bet is Korg Gadget. Gadget doesn't do audio recording at all, but you can import samples, record and program MIDI patterns, and even trigger your arrangement in sections, much like the way Ableton Live uses scenes. There's a reason for this – not only can Gadget projects be rendered as stereo mixes or MIDI files, they can be exported as Ableton Live projects that you can open with Live on your computer, so, naturally, the similarities with Live's organisation of clips and scenes are essential to make this work. This is a gift for Live users who are travelling light. Gadget installs with a selection of instruments, and there are more available as in-app purchases, including the London drum module, Bilbao sample player, Chicago acid-house bass machine, Montreal vintage electric piano, and the Kiev digital synth. Each gadget has presets, and a fairly simple level of control available, but by mixing and matching them, and by employing automation within the clips, a decent level of sound design is possible – the results are good enough that they don't embarrass themselves when you port the project over to Ableton Live and mix it with sounds from other sources. Naturally, importing your own custom samples will impart more individuality and sonic variety to proceedings. Other Gadget features include MIDI sync, MIDI mapping, unlimited tracks (exactly how many tracks you can use will be limited by your hardware), and support for Ableton Link, inter-app audio, and audio bus. Especially since it runs on the iPhone as well, Gadget is a fantastic Ableton-ish sketch pad that you can always have with you.

Gadget can trigger your arrangement in sections, much like Live uses Scenes

Gadget – a portable pal for Ableton Live aficionados





THE SETUPS

The multi-channel recordist

PART 1

To create a DAW-type rig with an iPad and Auria Pro, you'll need a multichannel audio interface – with an app that can record up to 24 tracks simultaneously, a simple two-in/two-out isn't really going to be adequate. There's the **Focusrite Scarlett 18i8**, for example, which can provide up to 16 channels of analogue in; but you'll need to power it from the provided mains power supply as an iPad cannot provide enough power for a device like this. You'll then need a basic USB cable and Apple's Camera Connection Kit. If you're willing to jump through more hoops by adding a mains-powered USB hub, you could add a MIDI control surface as well. If you're more interested in an Ableton Live-style rig, which might be more about launching loops and MIDI control with Gadget, you could choose the **iConnectivity iConnectAUDIO2+**, and pair it with the **Faderfox PC4**, a compact MIDI controller with 24 knobs. The iConnectAUDIO2+ needs mains power, but does connect directly to the iPad without the CCK, and the PC4 connects to the iConnectAUDIO – saving a few cables along the way. GarageBand iOS only supports stereo recording, so hardware selection for that isn't so critical.

GarageBand – possibly the best four quid you'll ever spend, and it's becoming more poweful with each update, too

GARAGEBAND

Price **£3.99**

Finally, we have GarageBand – which might seem a bit of a tame option. It's Apple, and everybody knows about it already, right? However, it's very capable for basic recording and composition, with a great suite of 'instant music maker' tools if you're really in a hurry to sketch things out. The killer GB iOS feature is, of course, that the projects it creates can be dropped over to a Mac and opened immediately in the OS X version of GarageBand, or even Logic Pro X itself – no intermediate conversion is required. The last major update added two important features: Audio Unit Extension support, which allows the hosting of third-party-instrument apps within the GarageBand environment, and – provocatively – the introduction of a suspiciously Ableton-like interface for Live Loops, where you can launch individual audio or MIDI loops, manipulate audio effects (including orientation-based parameters), and trigger rows of loops simultaneously to create scene-like behaviours. GarageBand costs £3.99, which is a huge bargain, considering what it can do. But don't underestimate it because it's cheap!



ISM INCORPORATED SOCIETY OF MUSICIANS

Join our growing membership of professional musicians.
Make us your ISM.

@ISM_Music
facebook.com/ISMusicians
ism.org

Interfaces

Sometimes, plugging in a pair of headphones is enough for iOS production. But for musical applications beyond basic song sketching, you'll need some kind of **input and output**, which allows more flexibility than 'phones and a built-in mic. It's just like desktops, when you think about it...

A surprisingly large number of USB interfaces are iOS compatible – which is mostly thanks to Apple's Camera Connection Kit – and as you'd expect with such a popular platform, there's something for everyone.

Beginning at the higher/more expensive end, there's the **RME UCX** (www.rme-audio.de). This is a typical RME product – perfectly constructed, great-sounding, and stable in performance terms. And, at £899, it's not cheap. This is class-compliant with iOS, and will provide up to 18 channels of input and output – that's 8/8 analogue, 8/8 ADAT, and 2/2 SPDIF. That count includes two mic preamps. MIDI in/out is also available on a breakout cable – this is a half-rack-sized unit, so you can forgive them for not fitting everything onboard, I guess. As well as the iOS functions that concern us here, the UCX also connects to a computer via USB or FireWire. Breakouts notwithstanding, you're going to get full regular performance from the UCX, as well as – according to RME – the same sound quality and timing accuracy as with OS X or Windows. As long as you can afford it, you can't go wrong with RME.





The RME UCX is **perfectly constructed, great-sounding,** and **stable**. And, at £899, **not cheap**

The **Keith McMillen Instruments K-Mix** (www.keithmcmillen.com) is a totally up-to-date take on the audio interface, and this one again is compatible with computers and iOS devices, not to mention featuring a standalone mode, so you can use it in hardware rigs. The K-Mix includes eight balanced inputs (with two mic preamps) and 10 balanced outputs (including the headphone out). Interestingly, those top-panel controls can be employed as a MIDI control surface as well, simply by changing presets, so this is a very versatile unit that also saves desk space... and I haven't even mentioned the onboard digital effects yet. When used with a computer, the K-Mix is fully functioning on USB power only, which is impressive. Unusually, the K-Mix features two USB connections, so it can be powered from the mains at the same time as it's connected to your iPad, without needing the Camera Connection Kit.

The iConnectAUDIO2+ (below) is an iOS-literate MIDI toolkit





The small, yet perfectly formed iRig Pro Duo interface has plenty of connectivity options

THE BEST INTERFACE FOR THE JOB...

The best thing about all of the interfaces mentioned here is that they're not platform-specific. You can use them with your iOS devices and apps, which, realistically, are still more ephemeral than their OS X/Windows cousins, but they also work with those computer cousins as well. That's about as future-proof as it gets with these! RME, in particular, has a stellar record in terms of support, although that's reflected in the price, of course. The RME interfaces are very complex devices, though, and could be reasonably considered overkill for the majority of iOS users.

Each interface mentioned here has a speciality, and hopefully, it'll be clear which suits your needs. It's like MIDI controllers – you'll end up with more than one – it's not unusual to have a more complex interface for studio use, and a simpler stereo model for live use. The UCX has the wealth of inputs and pro features; the K-Mix has the bonus MIDI control surface operation; the iConnectAUDIO2+ serves when you must use two platforms at once in your rig. Plus, the iRig Pro Duo has battery power and MIDI routing, and the AUDIO2 has the ruggedness, compactness, and the focus on DJing.

The mini-est digital DJ rig known to man – iPhone with Traktor DJ, and the AUDIO2

The **iConnectivity iConnectAUDIO2+** (www.iconnectivity.com, approx £179) is a very 'iOS-literate' audio and MIDI interface – it's not just included as an afterthought. The AUDIO2+ features two audio inputs, six outputs, and MIDI in/out on 'proper' connectors as well. It'll let you connect two computers at the same time, or – which is what we want here – an iOS device and a computer simultaneously, and it'll charge the iOS device as well. Add a very comprehensive range of MIDI and audio routing options, and this is a fantastically useful audio/MIDI toolkit for the studio where iOS and MIDI meet. I'm seeing the iConnect range in a lot of live performance rigs.

Going smaller and ever more portable, the **IK Multimedia iRig Pro Duo** (www.ikmultimedia.com, approx £149), is a two-channel audio interface that'll connect to your Mac, PC, iOS, or Android device. Cables for all of these options are included, and there's an optional one for connecting older 30-pin iOS devices. It can be powered by the connection where possible, or mains, or batteries, so lots of options there. I'm happy to see it includes two full-size 1/4-inch balanced outputs and 48V phantom power, alongside the two XLR mic/instrument/line inputs. The package also includes adaptor cables, so you can route MIDI communications from hardware through the iRig Pro Duo to your computer/device/whatever. That, and the battery-power option, will be big selling points to some folks.

As our interface selections continue to shrink (not in a bad way), we encounter **NI's Traktor Audio 2** (www.nativeinstruments.com, approx £69). This connects to your computer via USB, but is also lightning-compatible, so you don't need the Camera Connection Kit to plug it into your iPad. An optional power supply enables iOS device charging, too. This DJ-focused interface has two separate stereo outputs, so you can cue your tunes with it. The mini-est digital DJ rig known to man – iPhone with Traktor DJ, and the AUDIO2. That's pretty mini, I think you'll agree. And in case you're wondering, it's not Traktor-only!



Traktor Audio 2 is DJ-focused, with two stereo outputs

Beats and DJing

You can, of course, **work with beats** in various ways inside iOS DAW apps such as Gadget and GarageBand, but there are some surprisingly functional iOS beatmaking apps. Here, Martin Delaney highlights two of the best, as well as squeezing in a DJ app, and a looper app, just for the sake of completeness





Maschine 2 (£7.99) is an iOS take on Native Instruments' popular Maschine hardware/software combo. As well as the traditional 4x4 grid of pads, there's a keyboard option, with useful chord, scale, and arpeggiator frills. But primarily, it's about making beats – be they recorded in real time, or created in step mode for the less finger-drum-savvy. Use the library content, your iTunes library, or record audio straight into the app. 3D touch is supported on compatible devices, for certain functions like note repeat. Ableton Link is included, as is the ability to transfer projects over to the desktop version of Maschine. Arrangements can be created by chaining patterns, but iMaschine is probably best viewed as a simple environment in which to create loops rather than entire songs.

iMaschine 2 enables you to transfer your projects to the desktop version of Maschine



Akai's contribution to the iOS drum category is the **iMPC Pro** (£9.99), which scales up nicely to the glassy expanses of the larger iPad Pro, with separate tabs for main functions such as Mixer, Timeline, Song, etc. Impressively, up to 64 tracks are available, with gain, EQ, and sends available on each. Other standouts include an expandable sound library through in-app purchases, onboard sampling, time stretching, and sidechaining. Songs are constructed through 'playlist'-style lists of patterns, which is not as visually rewarding as seeing an arrangement timeline, but works well enough. Detailed pattern editing is available, with neat shortcuts for common tasks such as tuning, velocity, and filter frequency and resonance.

If you want to DJ with your iPad (or even your iPhone) **NI's Traktor DJ** (£7.99) supplies two decks, and all the DJ features you'd expect – sync buttons, crossfader, EQ and effects, looping, tempo and key detection, a browser etc, with a cool SuperSlicer tool available for in-app purchase. Touching waveforms to set cues and loops never gets old. Link isn't supported, but inter-app audio is. It's compatible with NI's flagship S2 MKII and S4 MKII controllers/interfaces, which should be awesome, but sadly, I haven't tried it yet.

In the 'we had to cover it somewhere' category, is **Blocs Wave** (£4.99), which focuses on importing, recording, and interacting with sets of loops. Other



The iMPC Pro has an expandable sound library, onboard sampling, time stretching, and sidechaining

than that, it's bare-bones, though Link is supported, as are AudioCopy and Audiobus. It's all about the interface, which is futuristic but still makes sense. Six sets of eight loop pads, known as sections, allow enough combinations to sketch out ideas. There's time-stretching, and a neat slice mode – waveform editing is a treat. Bonus – library content stays in key if you change key. It's a great host for sampled beats when paired with other Link-enabled apps.

BEATS AND DJING

This is a good time to discuss Ableton Link, a sync method that enables music apps on different devices to play in sync. That's all it does, but this is one of the biggest issues when it comes to using more than one application or device together. Link works across WiFi, or even an ethernet cable if you want it to. You can be using Ableton Live on multiple computers, or Link-enabled apps on iOS, or mix and match platforms. You don't need to include a computer at all, it'll work with just iOS devices. It even works with multiple apps on the same device. There's no theoretical limit to how many apps or devices can be on the same Link session at the same time. All you have to do is enable Link on the apps in question, and start them playing. There's no start/stop or transport control, that's still for you to do, all it does is keep them playing in time. Anybody can change tempo, and anybody can join or leave the session at any time. External hardware can be integrated, if one of the participants is sending MIDI sync out. Link is excellent!



Blocs Wave (right) is a sketchpad for beats, which uses eight 'sections' to construct ideas from interacting loops



iOS Synthesisers

Getting a great **iOS synth** has never been easier, with hundreds to choose from. Here are six great and varied synths – and advice on getting the best from them...



IM1

Developer **Korg**
Compatible **iPad iOS 8.0**
Price **£22.99**

Where most iOS synths emulate classic analogue synths – and why not, with a touch interface that lends itself to twiddling virtual knobs? – or create something completely new, there are few that plunder the history of *digital* synths. Korg's iM1 does just that by recreating possibly the greatest digital synth ever made – or at least the one that made the most impact on its release in the late-80s. The M1 scored that period of music, and it is recreated in the iM1 – which, we'd argue, is even better than the the real thing. It's easy to program, you can buy all the extra sound packs (which won't cost you 'loadsamoney' *[like the 80s reference there* – Ed]) and nothing else will give you the brash, go-faster stripes that were digitised synths from this decade. Expensive, but iM1 does a lot for your buck.

Korg's iM1 (above) is an iOS emulation of its 1988 digital workstation, the all-conquering M1



ISEM

Developer **Arturia**
Compatible **iPad iOS 8.0**
Price **£7.99**

Arturia is the king of recreating classic synths in software, so it's no surprise that the company has now become the iKing of creating synth classics on the iOS platform. iSEM is a synth based on the 1974 Oberheim Synthesizer Expander Module, one of the first all-in-one synthesisers but, as usual with Arturia, there are one or two modern additions. You get the ability to edit up to eight voices simultaneously, a bunch of effects, and an arpeggiator. The 500+ presets utilise a similar technology that the company uses in its desktop synths – a modelling rather than sample-based engine – so the results are slick, 'analogue' and as real as you like. They're perhaps more sheeny than gritty, but evoke the original like no other [*apart from maybe Tom's hardware synths* – Ed].



MAGELLAN

Developer **Jonac**
Compatible **iPad iOS 7.0**
Price **£10.99/$14.99**

Two synths in one, anyone? Magellan has a pretty mighty sound, thanks to two independent synth engines and a rack full of colourful effects. Each engine has three oscillators, two filters (with 11 types each), two LFOs and an ADSR envelope.

In addition, you get a sequencer, chord maker, arpeggiator, comprehensive modulation section and over 300 presets, which vary from some inspiring pads to big basses, plus another 10 categories. To be honest, there's not a lot you don't get – which makes some of the controls a little small and fiddly. But it does result in one of the most fully-featured and biggest-sounding synths out there. Quite superb.

MODEL 15

Developer **Moog**
Compatible **iPhone 5+ or iPad iOS 9.3**



Price **$29.99**

At Superbooth16, we saw a number of companies attempting variants of modular synthesis on iPads, but Moog's is an out-and-out app version of a full modular system, comprising the usual (albeit classic Moog-type) oscillator, filter and other modules. You get virtual wires and a very usable pinch-and-swipe system to navigate around. You'll soon be enjoying those deep Moog sounds, although, with a whole stack of presets, you don't really have to know too much about modular to get them. If nothing else, this is one of the cheapest ways to get that full-on modular synthesis experience – yes, it's quite a lot for an app, but cheap for the equivalent hardware, believe us. Model 15 is possibly the most involving synth app in our round-up – you'll need later iMachines to run it – but it's also the most true and, well, real...!



SUNRIZER

Developer **Beepstreet**

Compatible **iPad iOS 5.0**

Price **£7.99/$9.99**

It's another virtual analogue synth, but it's so well adapted to the iOS environment, you'll quickly be creating more different sounds with this than most others. It's based partly around the Roland JP-8000 Supersaw, although the two oscillators feature multiple waveforms, plus noise and ring modulator and there's up to 20 notes of polyphony. There are also two filter units and onboard arpeggiator, plus two LFOs and ADSR envelopes. We were more than impressed with the onboard effects, especially the distortions and reverbs, which take the many sounds you get (over 200 from four banks, plus favourites) in completely new directions. Easy preset categories and a fine overall sound make this a modern classic.

THE PERFECT SYNTH SET-UP

Getting an iOS synth to run with your favourite desktop DAW is not really the point of this article, although it is a relatively easy process just to use something like the Audio Unit Extension app that lets you run iOS apps as plug-ins. On a simpler level, you could connect an iOS MIDI interface and treat your iPad like an external instrument, recording audio in. Better still, you could and should use an app like Secret Base Design's Music IO to bring everything together. Essentially, this lets you run your iOS synths like AU or VST instruments by connecting their audio and MIDI streams with your desktop machine. We had Magellan and Sunrizer running perfectly within our Ableton Live tunes, for example. However, as this is a mobile feature, if you want the perfect mobile synth setup, there's little better, simpler and more stunning than Korg's Gadget (see p24) – a kind of synth studio on acid. All the synths you will ever need, and more, ready to record in one app...

Thor has real **complexity,** with **six oscillator types, four filters, envelopes, and LFOs**

THOR

Developer **Propellerhead Software**

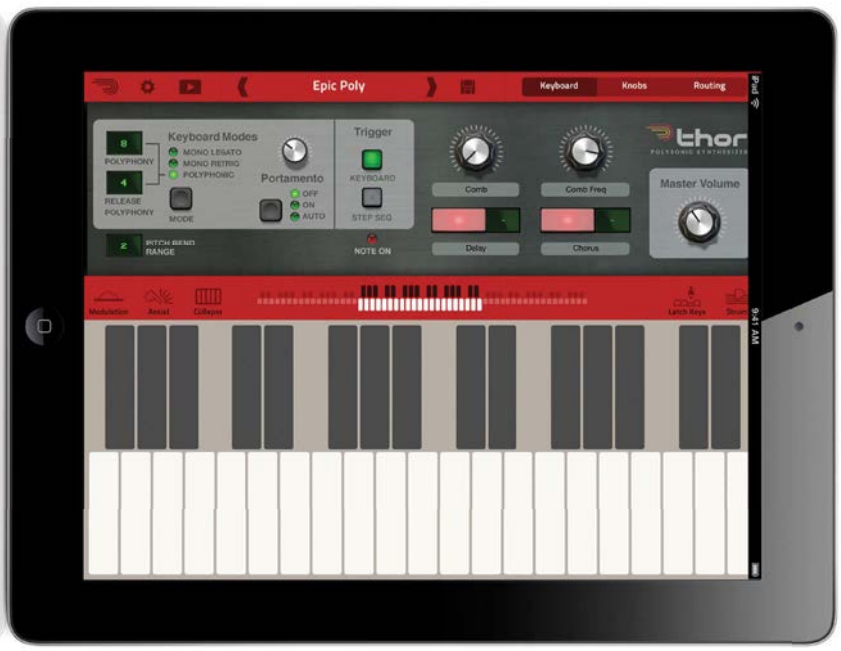
Compatible **iPad iOS 8.0**

Price **£10.99/$14.99**

With a name like Thor and the developers behind Reason, you'd expect this synth to sound amazing and be easy to use. Yet it also has a real level of complexity in terms of the possibilities, with six oscillator types, four filters, envelopes, and LFOs. We particularly like the semi-modular approach to routing from left to right – fairly standard, but very well implemented.

The six types of oscillators will give you everything from analogue to FM to wavetable, resulting in versatile soounds ranging from classic to in-your-face EDM sounds. Thor might have been released over three years ago now, but it still has a great sound and enough flexibility for most tasks. You might be a little overawed by the flexibility and interface at first, but Thor will reward some tinkering beneath its fine-looking and sounding surface.

Sunrizer (left) is a modern classic in the virtual analogue synth realm; whereas Thor (below) is packed with possibilities and rewards time spent getting to know its ways



Sound Design

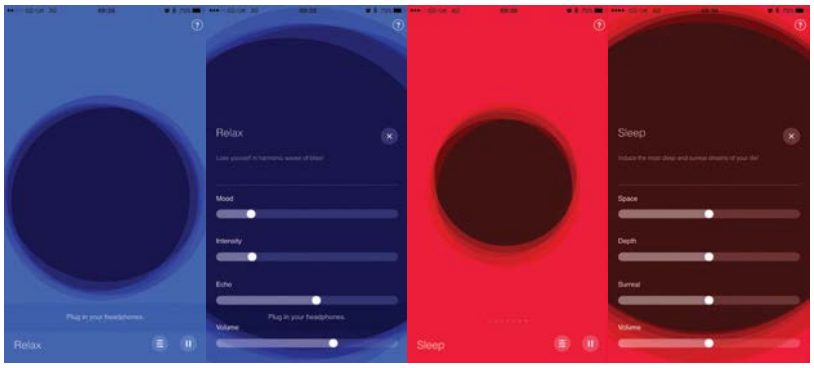
They're neither instruments nor recorders; instead, our final collection of apps covers the broader area of sound design. Martin Delaney gathers the most creative sonic tools for mobile music production...



With our final collection we're rounding up some apps that aren't typical synths or instruments but provide unique ways of creating and interacting with sound. **Samplr** (£7.99) is an iPad-only app which makes full use of the ability to touch and manipulate waveforms directly. It features eight different play modes including Slicer, Looper, and Keyboard, and the ability to play six samples simultaneously, with up to five effects per sample. Use the provided sample library, or your own imported sounds, or record straight into the app. AudioPaste, AudioCopy, and AudioShare are supported, and Samplr can receive MIDI clock input. It's a really sweet glitchy tool that scores bonus points because it doesn't try to emulate any real-world hardware.

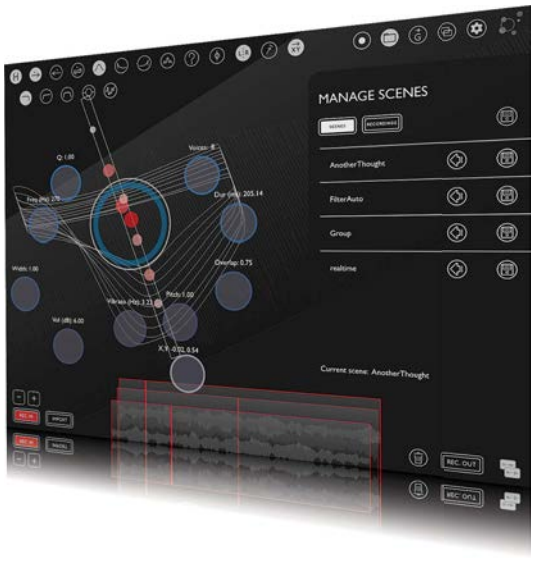
Borderlands Granular (£7.99) shares the Simplr design philosophy – embracing the possibilities of iOS interface design instead of looking back to wooden end panels and big knobs! This does mean a few minutes are required to get familiar with the controls, but this does pay off. It is effectively a multitouch based granular synthesizer, which can use the factory sound library, your own recordings, or real-time input from the iOS device microphone or an audio interface. You get touch control over 'clouds' of parameters, including number of voices, duration, vibrato, pitch, and more. Borderlands Granular can be used equally as a performance instrument or an in-studio sound design tool. Record your jams and export them to via AudioShare, then to Dropbox, Mail, AudioCopy, or other music apps.

Yes, **'The app formerly known as H__r'** (£ free) is really called that, although a competition to rename it was under way. Coming from the creators of the RJDJ app, this requires a headset with mic, so it can process the incoming audio from your surroundings and apply one of several presets designed to create different ambiences - mostly they're kind of electronic and edgy but there are some interesting results, especially after you edit the settings to suit your situation and mood. Works best with headphones that effectively exclude outside sounds, for a more immersive experience. This isn't a music 'creation' app as such, but it's thought provoking and worth a go if you're interested in real-time immersive audio.



Finally, and using the term 'sound design' in a slightly different context, there's **BIAS Amp** (£3.99),

> It's not a **music creation app as such** but it is **thought provoking** and an **immersive experience**



Samplr (top left), the app with no name (top) and Borderlands Granular (right).



Bias Amp simulates both vintage and modern guitar amps.

from Positive Grid. This app includes simulations of many vintage and modern guitar amps, with presets ready to use, but beyond that, the amps and cabinets are customisable, right down to the tubes. However there's more, because you can begin building your own amps from scratch, selecting every component, from power amps to transformers, to tone stacks. After you've created your dream rig with BIAS Amp, it can be opened in Positive Grid's **BIAS FX** and **JamUp** guitar apps (Inter-App Audio and AudioBus are also supported). Even if you don't feel ready to jump into full-blown amp design, the BIAS presets sound great right off – this is one of the best iOS guitar amps around. **MTF**

EASY AUDIO

There are separate iOS apps that'll make your life easier, and get your iPad productions a bit more finished. Throughout this story we've referred to apps such as **AudioBus** (£3.99) – which routes audio between applications on the same device; **AudioShare** (£3.99), an audio-and-MIDI file management tool, and **AudioCopy** (free) which is a clipboard, organiser, and editor, for audio content. These can make all the difference when sharing content between apps and, thanks to them, you have the option to include audio effects in your app chains as well. For example I use **Kymatica**'s AUFX effects such as **Dub** (£3.99), a really nice stereo delay, and the **Space** reverb (£3.99). If I really want to finish a tune within the iPad and share it online immediately I'll move the stereo mix to **Positive Grid's Final Touch** (£13.99, see below), a very well-featured mastering app with maximiser, EQ, compression, stereo imaging, and multiband dynamics – everything you need for that last-minute tweak. Using a bunch of little apps like this may not as elegant as doing it all within a DAW, but it does work!





Ableton Live Workshop

On the disc
DVD ROM
Accompanying project file included on the DVD

Making the most of the Sine wave in Live...

The humble Sine wave is the most vanilla of all raw synthesiser waveforms, but it's incredibly versatile and effective, and appears throughout Live. **Martin Delaney** explores ways to get creative with it...

Waveforms are the fundamental building blocks of synthesis, whether you're using software or hardware, and the Sine wave is arguably the most fundamental of those; it's definitely the most basic. Waveforms are found within oscillators, and that means LFOs – low-frequency oscillators – are also dependent on waveform selection for their character, including the Sine. I'm not a deeply techy person – and neither should you be, you've got music to make, after all – but it's worth getting a handle on some synthesis basics. The Sine is a totally 'vanilla' sound, but it's incredibly useful. We'll talk about some examples of how to use this flat tone to create far more interesting sounds.

The classic synthesiser waveforms are Sine, Saw, Square, and Triangle. Each of these has a distinctive sound, although there are many possible variations. Inside Operator, for example, there are three Sines, eight Saws, eight Squares, and just one Triangle. Hardware synths don't necessarily feature Sine waves: for example, the Korg MS-20 uses Triangle, Sawtooth, Square, and white noise. In FM (frequency modulation) synthesis, the Sine wave is very important, and that's partly how we're using it in our tutorial here. Operator is at heart an FM synth, even the name comes from that – an 'operator' is what an oscillator is called in FM synthesis, and an oscillator is the part of the synth that produces the waveforms. Sine waves are used for drums, bass, keyboards, pads, everything , in fact – you could build a whole composition, a proper viable production, using only Sines as a sound source. That would be a great creative challenge, and a great way to learn more about synthesis, Operator, and Live. Hmm, I'm going to try that project myself!

Inside Live itself, you'll find Sines in the Operator and Analog synths, and any device with an LFO, including audio effects such as Auto Filter and Auto Pan. Thanks to the expansion opportunities provided by Max For Live, there's an audio LFO device and a MIDI LFO device available, and again, both of these include Sine waves (they're in the free Max For Live Essentials pack, downloadable from www.ableton.com).

I've provided a Live set which contains some audio examples of what we're talking about, as well as some beats and other sounds to play along with. You'll need Operator to follow the exact steps in the tutorial – I won't apologise for using this pay-synth for a tutorial, because it's so good, and as I've said previously, it's the only Live add-on that I recommend for every user. If you load the default Operator setting, you're hearing a Sine wave. Operator has four oscillators – four opportunities to create waveforms, and you can turn each on or off – but if you turn them all off, you won't hear anything. By default, they're all activated, but only A has its Level control raised,



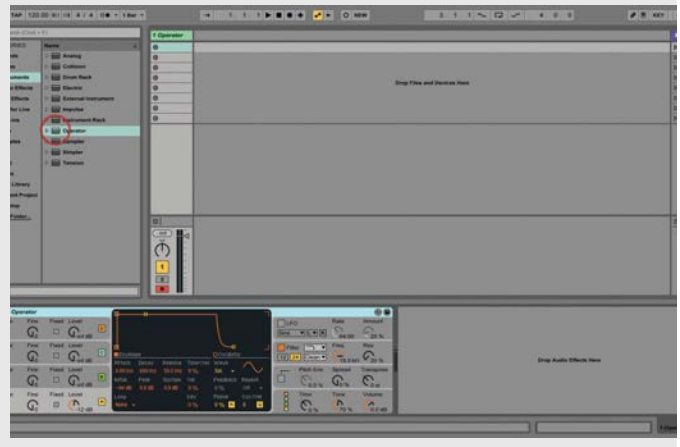
HARDWARE OR SOFTWARE SYNTHS?

You can use hardware or software synthesisers – both have pros and cons, and I guess budget is a major consideration for most of us. You can do it all with software, although it isn't as glamorous as those dust-attracting hardware boxes. I used to divide the tasks; for analogue synth sounds, I'd use real hardware analogue synths; for more experimental and digital sounds, I'd use software – that's why I bought an Analog Keys instead of a Virus. These days, I'm back to software, mostly – hardware selection is more about the user experience, looking for something with character to create a certain vibe.

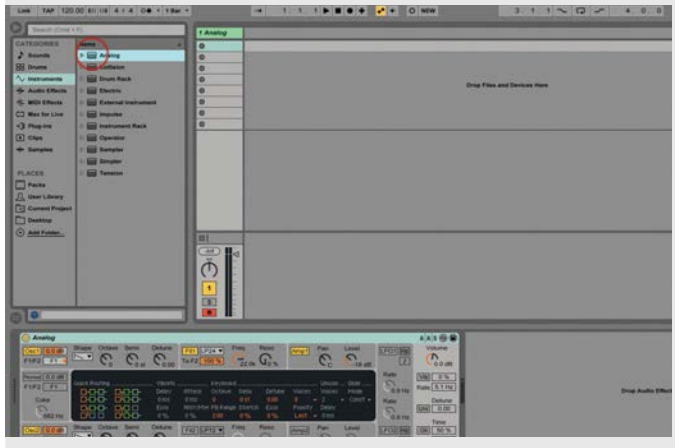
MTF Step-by-Step It's a Sine



01 We're going to explore Sine waves inside Live. Open our example Live set – it has a few clips you can play along with, as well as some 'after' examples of Sine-wave practice. Everything's labelled.



02 This tutorial is nearly entirely based around Ableton's Operator synth, which is included in the Live 9 Suite, and also available as a separate purchase – highly recommended (by me) for all Live users!



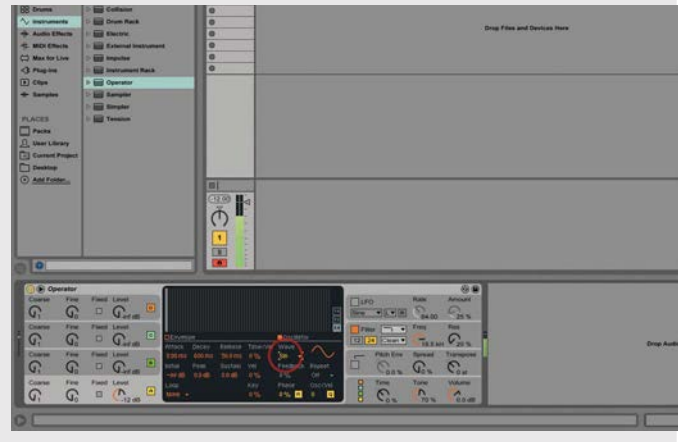
03 If you don't own Operator already, try it inside the Live 9 demo, which includes Operator and all of the other Live 9 Suite instruments and effects, such as Analog, another very cool synth.



04 So, assuming you have Operator one way or the other, let's begin. The default sound's what we want – just a basic Sine wave. Play it from a keyboard, Push, or Live's virtual MIDI keyboard.



05 It's a straightforward sound. Select the Oscillator A tab in Operator and in the darker central area, you can choose other waveforms, including variations on the Sine. Try those, too.



06 When you play the Sine, you might notice a 'glitch' at the start and end of notes... that's how it is with Sines, and there are little tricks to conceal that – see our main text.

the others aren't having any effect. As we raise the Level control for B, the sound changes drastically, even though this is just another Sine wave being brought into play.

You're not hearing the sound of two Sine waves playing alongside each other, rather, Operator is using frequency modulation, traditionally (but not always) based on Sines, where one oscillator has an effect on another – when one waveform is applied to another in this way, changing the level affects the tonal character of the original. That's the very short explanation! Through this process, we get something much more dynamic and interesting, especially if we keep the Level control moving, adjusting it in real time, or automating its movement. That's only with oscillators A and B, so of course, you can introduce C and D as well. If you look near the bottom-right of Operator's central panel, you'll see a little graphic with four coloured →

MTF Step-by-Step It's a Sine (cont'd)



07 A combination of Sines can produce textures far richer than you'd expect. While triggering some notes, raise the Level for Oscillator B – that's just another Sine, but the sound changes dramatically.



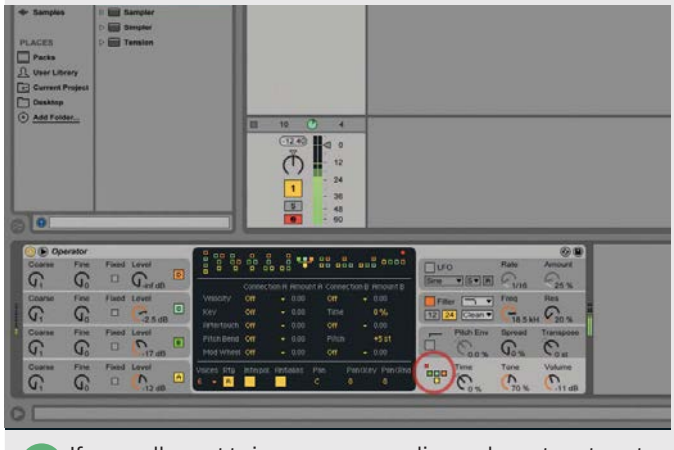
08 We can go further with this by introducing C and D as well. Personally, I rarely use all four oscillators, but that's up to you. This is based on FM synthesis – for more info, see our main text.



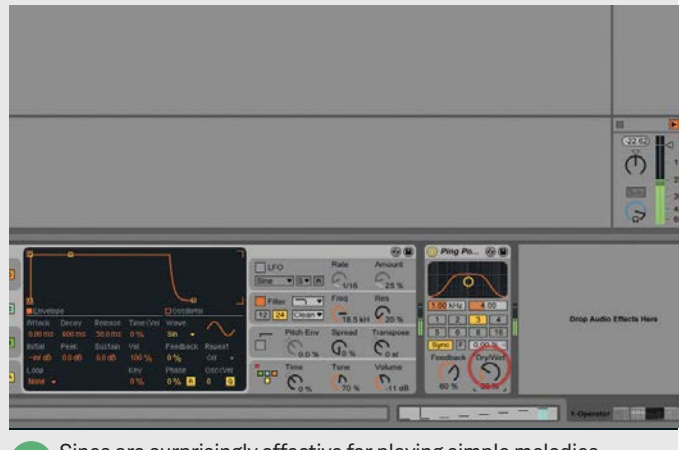
09 Introduce dynamics by using velocity to change the oscillator levels. Select an oscillator, then set a 'Vel' value, and it'll respond. You'll need to set an initial level other that 0, otherwise it won't work.



10 Another thing you can do with two or more active oscillators is to change the default algorithms (the little boxes at the bottom), to recombine the Sines in different orders. This can make big changes.



11 If you really want to impress me, use clip envelopes to automate switching between algorithms as your music plays. This can create some interesting 'stepped' changes in your basic synth sound.



12 Sines are surprisingly effective for playing simple melodies – I (over)use them a lot. Add some Ping Pong Delay, play something in a minor scale, ideally over an 808 beat, and you're laughing.

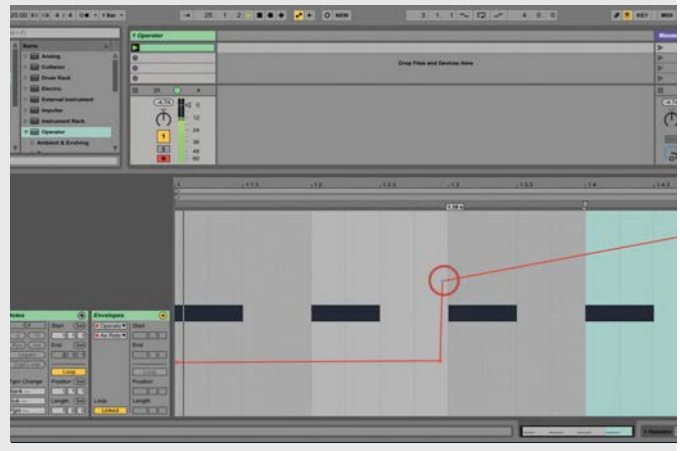
→ boxes, corresponding to the colours of oscillators A to D. Click there and you can choose different arrangements of these different algorithms that affect how the oscillators interact with each other. One of these does, in fact, 'undo' the modulation and play the oscillators side-by-side, so you should try that, just for comparison purposes. The algorithms are MIDI-selectable, which opens up a lot of fun options for automation or manual changes during a performance or recording.

Sines can have an audible 'bump' at the beginning and end of notes. The Sine wave is a smooth shape, and note on/off actions create a glitch, because there's no clear zero point where there's nothing happening. Sometimes in a mix this isn't noticeable, so don't stress too much if you can

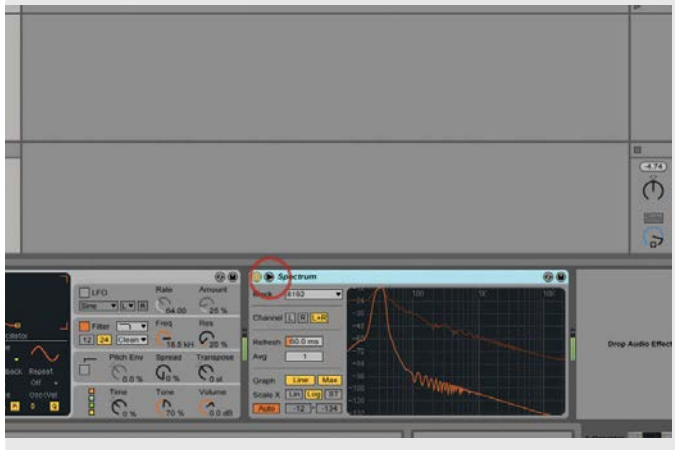
MTF Step-by-Step It's a Sine (cont'd)



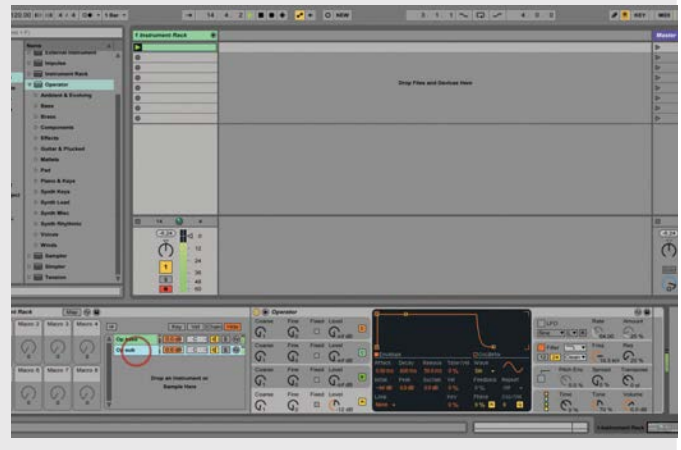
13 Sines are great for kick drums. Experiment with the attack if you need to soften it a bit, and have fun with the release, giving it more or less of a tail. Add a compressor.



14 Use your Sine kick on its own, or layer it under another kick to fatten things up. Automate the release as the song progresses, this can help to clean up your mix a lot.



15 The Sine is, of course also effective as a sub-bass source, precisely because of that 'vanilla' quality – it adds low frequencies without adding much tone to affect your existing palette.



16 Put your Operator Sine sub in an instrument rack with your primary bass sound, so they're both triggered at the same time. Use EQ, and the rack's mixing controls, to balance them out.



17 I also use Sines for electric pianos. You can achieve some really super-clean cheesy piano tones, or add audio and MIDI effects to create darker, more complex custom presets.



18 Finally, every LFO in Live (and there are quite a few) has a Sine option. Load Auto Filter and try it with our example audio clips – choosing different waveforms for the LFO can make a difference.

hear it solo'd, but not with everything else playing. In some situations, you can edit the attack and release of the notes to smooth that off with longer fade-ins and tails.

There have been times when I've flattened the Sine wave MIDI part to an audio clip, and then automated a clip envelope to take the edge off it. More recently, I've been using the Transient Machines Max For Live device from Surreal Machines, which is designed to deal with exactly this type of problem, and I've found it very effective. Hopefully, this has been a good excuse to learn a little about FM synthesis in general, and Operator in particular. As I said, don't be put off if you don't use Operator yet. It's absolutely worth the time (and the very modest asking price), so try the demo. MTF

Ableton Live Workshop

On the disc
DVD ROM
Accompanying project file included on the DVD

Using Ableton Link to synchronise your computers and iOS devices

Ableton Link is the smoothest, easiest, possible way to get your iOS music apps and Ableton Live talking to each other. **Martin Delaney** demonstrates how to do it – and thankfully, it's brilliantly simple…

After years of forcing us to be locked away at home, the world of music tech is finally getting more social. These days, we're more likely to take our precious music gear out of the house/studio, and meet with our fellow musicians; the ability to create music with our phones and tablets only adds to this. However, one issue that arises when electronic-music people get together is synchronisation – getting all of the associated gadgets to play in time together. Traditionally, we've had methods for dealing with this, such as ReWire and MIDI Sync, but on days where we don't want to carry too many interfaces and cables with us, but we still want to sync tightly, we need a clean solution, and that's where Ableton Link comes in. This is a method of synchronisation that's built into Ableton Live 9, and works over a Wi-Fi network or an ethernet cable. Outside the computer (and Link is cross-platform), Link is also available for iOS and Android developers (DAW developers too, it's already made it to Reason and Serato DJ), so suddenly we can include iPhones and iPads into our setups, in a reasonably seamless fashion. The setup doesn't even have to include Live, or any computer at all: Link works perfectly well between iOS devices.

There are many Link-enabled music apps available, although I have two favourites, the first being TweakyBeat, a fantastic bleepy beat-making app that never fails to make great sounds, and has the advantage of being free; if you haven't got that already, get it now! To configure TweakyBeat for Link, tap the little 'i' at the bottom right of the screen, then 'Link and MIDI' at the top on the next page that appears, then tap the red words 'Ableton Link'. You then see the Ableton Link page, where you can turn Link itself on and off, and choose to receive in-app notifications when other apps join or leave the session.

My other favourite is Korg's Gadget, which is the closest thing to Live on iOS – if you see Live as a programming/MIDI/synth environment, with scenes for launching different sections. Gadget is a pay app, with in-app purchases to add extra instruments. It has great synths, drum machines, and samplers. It also has the ability to export songs as either MIDI files, or more relevantly, as Ableton Live projects. Very cool (Novation's Blocs Wave app can also do this). The only shortcoming as such is that it – understandably – has to flatten all of your programmed MIDI parts into audio clips and scenes to ensure Live can



LINKED IN

Link is all around us. In the walkthrough steps here, I'm talking about Live, of course, which I'm running on macOS Sierra, then about iOS devices and apps. Make no mistake, though, Link is cross-platform, just like Live itself, so it'll work on Windows, and between Mac and Windows machines. There are currently a handful of Link apps for Android, but I'm sure that number will increase; I've also seen people experimenting with Link on the Raspberry Pi platform. And failing all else, thanks to good old MIDI, any hardware or software that understands MIDI can also be included.

MTF Step-by-Step Sync with Link



01 To experience Ableton Link, you'll need at least one Mac or PC running Ableton Live 9.6 or above, a Wi-Fi network, and at least one iOS device, with a Link-compatible music app.



02 Link-enabled apps include Korg's Gadget (the king of iOS music apps), TweakyBeat (free), and Moog's Model 15, to name but a few. There's a list of Link-enabled apps on the Ableton website.



03 Let's begin by syncing two computers running Live. Connect both to the same Wi-Fi network. Open Live on each, go to Preferences/Link MIDI, and enable Show Link Toggle. Close Preferences.



04 You'll now have a Link switch in Live's Control Bar, so you can enable Link more easily. Turn it on now, on both computers. A numeral will appear to show how many devices are present.



05 You'll also see a scrolling 'progress bar', under the arrangement position display. Start Live on one computer, and set the tempo. Start the other one, and it'll begin playing at the same tempo.

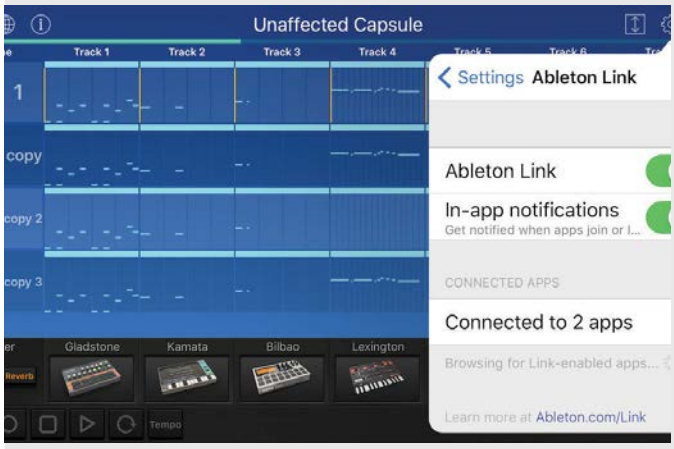


06 They're synced! You have to start and stop each independently, it doesn't include song position or anything like that. There's no sync master or slave, so either computer can exit the session any time.

open them properly. That's good enough for now, though. And of course, back to the point, Gadget does Link as well. Setting it up is quite similar to TweakyBeat – tap on the 'gear' icon at the top right of the main Gadget page to see the Settings menu. Tap to enable Ableton Link. That'll open up the Ableton Link menu, where you turn Link on or off, and once again can opt to receive in-app notifications regarding which apps are joining or leaving the session.

Using Link doesn't mean you can't sync to external equipment with MIDI at the same time – that would be kind of silly. As we all know, Live is a fantastic central hub for all kinds of music setups; anything that's synced via MIDI to Live will follow along if you're changing tempo via Link. It'll also work the other way round. If Live is receiving MIDI clock from a drum machine, for example, telling it to run at a certain number of beats per minute, then any Link →

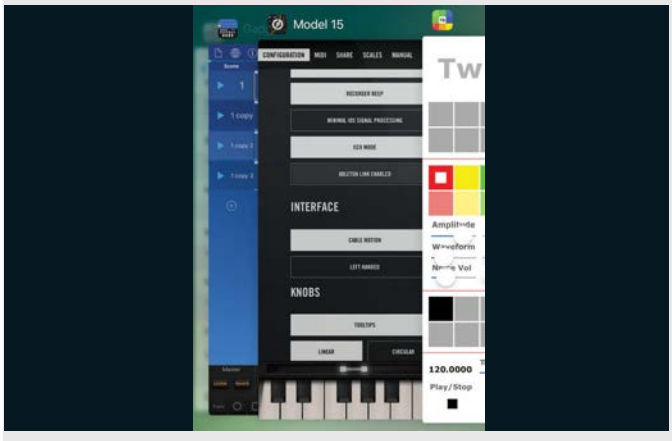
MTF Step-by-Step Sync with Link (cont'd)



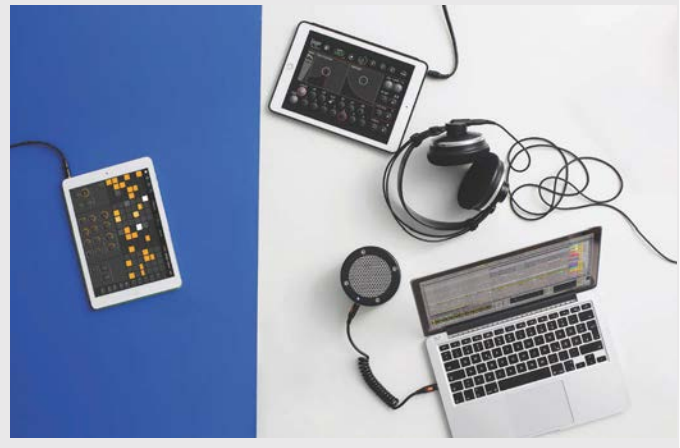
07 It's the same process with iOS. Any Link-enabled music app will have a switch somewhere in its settings or preferences, where you can turn Link on. After that, you'll see the progress bar again.



08 You'll see the display show the updated number of recognised devices. You can sync iOS devices with Live, or multiple iOS devices. Live doesn't have to be present to sync iOS devices.



09 It's also worth remembering that you can Link many iOS apps on the same device, if you need to, and from what I've seen so far, there's no practical limit to how many can be running at once.



10 Link is very, very good for those types of casual jams where any kind of setup time is going to seem like overkill and you just want to experiment or have a little social play.



11 If you're using computers rather than iOS devices, and you don't trust Wi-Fi, you can connect them via an ethernet cable and Link should run happily over that. I haven't tried this myself yet, though.



12 Tempo changes? No problem. Tap in a new tempo, or type in the numbers, or send it from MIDI, and everything that's linked will follow along smoothly. Plus, tempo changes can come from any device.

→ apps connected to Live will also follow that tempo. As we've said, Link handles sync only – it keeps everything playing on the beat at the same tempo. It's very simple and accurate, which is why we like it. However, if you want more detailed control than that, you'll have to look elsewhere.

To sync two music applications running on one computer, your best bet is to use ReWire. This is accurate in timing terms, but also provides start/stop and song-location info, so when you skip across your timeline, both programs stay tight. It also provides quite deep MIDI communication, so you can send MIDI notes and control messages from one application to the other. Equally with audio, ReWire will route audio between DAWs. For example, you could slave Live to Logic Pro X, and route the stereo mix

MTF Step-by-Step Sync with Link (cont'd)



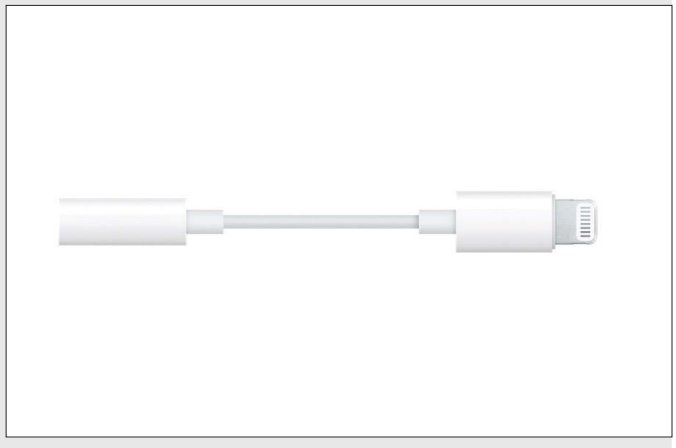
13 Of course, you'd need something beefier to properly hear your devices in a gig situation, but it's really cool just to have a bunch of iOS devices on the table and use their built-in speakers.



14 If you do need something more controllable in audio terms, get a little mixer and some speakers. It doesn't have to be expensive. Some small mixers are battery powered, too, for more mobility.



15 If your regular sound card has a standalone mode, which isn't so unusual, you could use that – give it some mains power, and you're off. The Keith McMillen K-Mix is fantastic for tabletop jams.



16 Maybe this is where we should mention the dreaded iPhone 7 and 7 Plus, which don't have an audio output, unless you use the lightning-to-audio adaptor, which at least is included in the box.



17 If you want to charge an iPhone 7 and get audio output simultaneously, you'll need to purchase a separate lightning adaptor like this one from Belkin, to enable both at the same time.



18 MIDI hardware can play with Link as well – just configure as usual with your MIDI interface and Live's Preferences, and any MIDI hardware that's synced to Live will follow along, or vice versa.

or even individual tracks out from Live to be processed collectively or individually inside Logic. It does create a very powerful setup. ReWire doesn't work between separate computers: to do that, you'll need a MIDI connection. This can be through MIDI cables between interfaces, or over a Wi-Fi network. With a bit of trial and error, this setup can work very well. Mac's OS supports network MIDI natively, and includes the ability to access a MIDI controller such as a keyboard, on any connected computer.

Link is great fun for the casual user – it's a pick-up-and-play, game-like way to sync with other 'device musicians', as well as being another useful tool for more serious setups as well. If you've got an up-to-date version of Live, and an iOS device, grab a Link-friendly app and take it for a spin. **MTF**

Ableton Live Workshop

On the disc
DVD ROM
Accompanying project file included on the DVD

Put your projects on a diet – reduce your track count in Ableton Live...

It's a wonderful thing that with computer production, there are few limitations on track count. But there are practical and artistic implications. **Martin Delaney** shows you how to streamline your projects...

Our aim is simple. We're going to look at some techniques we can use to reduce the number of audio and MIDI tracks in an Ableton Live project. We'll use instrument racks, freezing, resampling, and group tracks to achieve this aim. And why, precisely, are we doing this? 'Because we can', is the smart answer; 'because we want to learn some cool new things', is a better answer...

For most of us, our computers, or even phones, can run projects with more tracks than we need, that's not much of an issue any more. But those track-heavy projects can be hard to handle, hard to keep an overview on, hard to mix... there's an aesthetic beauty about a minimal layout that's clean, but still functioning as a composition, a mix and a performance space. For live shows, it's definitely on my mind: I want to have fun, not feel like I'm spinning plates.

If you want to make this a long-term thing, you can embrace it from the offset by composing in a more minimal way. Sometimes using fewer tracks sounds bigger and more exciting than using a lot of tracks. Instrument racks and clip envelopes are a great way to 'cut down', if you want to keep the full flexibility of MIDI programming and software instruments; using a clip envelope to select the correct chain means that you can launch the clips in any order and they'll still sound as they should. There are more dynamic ways to switch chains, based on manual selection, or note range, or velocity (drum racks behave differently – they don't have the same chain-select tools, but you can put a drum rack inside an instrument rack, and control the chain selection from there). There are also audio effect racks, so once again, you don't need separate tracks just because you want to use a certain compressor or EQ at one point in your set. The freeze and flatten commands, on the other hand, do away with all of that MIDI flexibility entirely, creating simple audio clips, which are much easier to reorganise while Live's running, without the risk of triggering the wrong instrument sounds. There are a couple of things to bear in mind with these commands – firstly, if you try freezing a sidechained track, you'll see this message 'The track [name of current track] cannot be frozen because of the routing [name of sidechain source track].' The other thing is that freeze can't be executed while Live's transport is running, so better do that before you begin your set!

If you create a new audio track, and open Live's In/Out



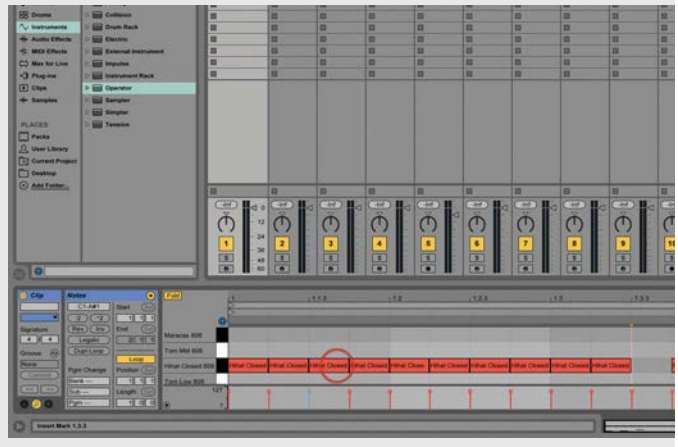
THE HATEFUL EIGHT

Let's talk about tracks. Why does everything have to be in multiples of four? I guess it goes back to the days of tape, where you'd be recording on two, four, eight or 16 tracks. But we're not recording on tape any more, not if we've got any sense. Hardware controllers are still based on that format, though – eight knobs, eight faders, eight rows of pads. Ableton's Push is just as guilty. And then we get the famous red box in Live's Session View, which forces us to work in banks of eight rows, eight scenes... somebody please try something different!

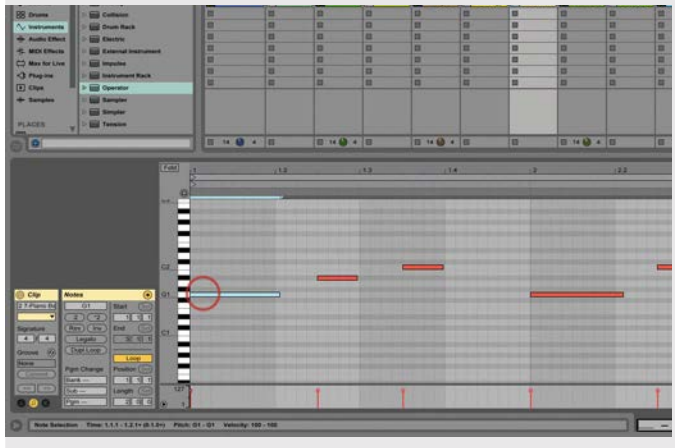
MTF Step-by-Step Reducing tracks in Live



01 We're aiming to reduce the number of audio and MIDI tracks in a Live set, so we're more free to rearrange our clips, and maybe to give the computer CPU a break as well.



02 We have an example Ableton Live set with 10 tracks in it, a combination of MIDI and audio content, and we're going to look at some different ways we can strip it down.



03 Before we begin – if you're focused on keeping a low track count from the inception of your project, you can make a good start by setting some rules for how you'll compose your parts.



04 Set a limit to the overall number of tracks – or how many tracks play simultaneously, which is not the same thing. I've sometimes resolved 'mixing' issues by restricting how many parts play at one time.



05 Anyway, moving on... in our example set, we have three bass tracks. We want to keep the different sounds, we need them all. But there's no need to give up three tracks for this.



06 Go to track 6 – Muugy, and the Simpler instrument at the bottom. Click to select it and then type Cmd-G, putting Simpler into an instrument rack. It'll still play as normal, at this point.

View, you can choose Resampling as an audio input source. This feeds everything from the Master output back in, so you can capture little submixes – combinations of tracks that you want to merge down into a single clip. I do this a lot with percussion tracks, mixing the relative volume of each part first, then crunching them down into a two- or four-bar looping clip. That can save me about four tracks from a set.

If you absolutely can't convert or merge anything else, but you're still desperate to have fewer tracks to look at, then it might be time for group tracks – simply created by selecting the tracks you want to group, then typing Cmd-G. You can combine audio and MIDI tracks in any combination, and apply audio processing to them as a group, which is really helpful for mixing. The good thing about group tracks is that they can contain many clips and devices, while at →

MTF Step-by-Step Reducing tracks in Live (cont'd)



07 You'll probably need to unfold the rack. Click on the Show/Hide Chain List button to, well, do exactly as the name suggests. There'll only be one entry, one 'chain', in there so far.



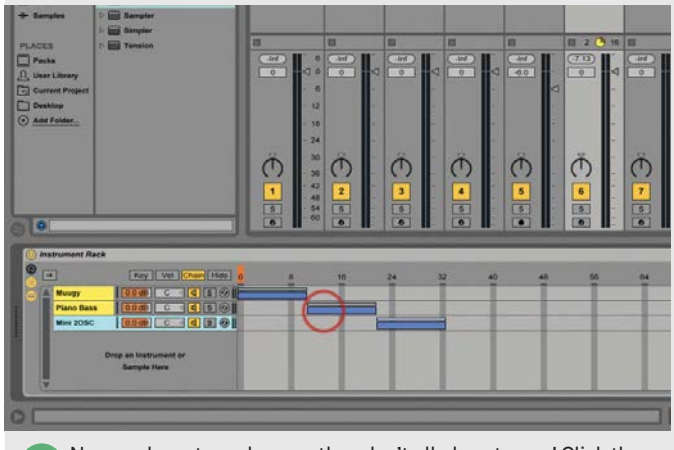
08 Go to track 7 – Piano Bass, and the Simpler instrument at the bottom. Drag that Simpler across into the rack in the first track, adding to the drop area below the first Simpler.



09 If you play anything in that track now, you'll hear both sounds playing at the same time, but we'll fix that in a minute. Make sure the MIDI clips from both tracks are in there now.



10 Repeat the process with Bass 3. You should end up with one track containing all three bass sounds in a rack, and all of the necessary MIDI clips. You can delete those other, empty, tracks.



11 Now we have to make sure they don't all play at once! Click the Chain box, and set a different zone for each, like in the screen shot. Now go to the clip for Bass part 1.

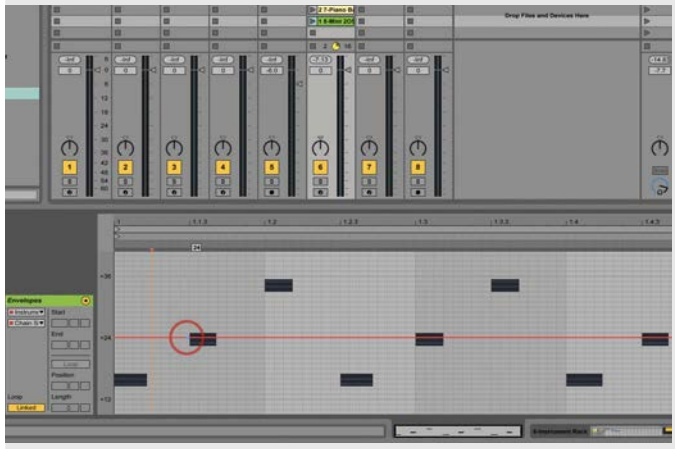


12 Click the Envelopes button. In the top-left chooser, select Instrument Rack, and below that, choose Chain Selector, and draw an envelope to select the range defined by your chain zone.

→ the top level behaving like a single track. The tradeoff is that if you want to change anything, some digging is required; although if you're using Push, you can fold and unfold group tracks and access their individual elements quite smoothly. The absolute last resort if you want to go simple is to mix your songs down to stereo mixes, then load them into a new empty Live set and use them DJ-style!

Then all you need is a couple of tracks to treat like 'decks' and a simple DJ hardware controller. Talking of hardware, I try to keep my live sets down to eight tracks, purely because that's how all controllers are laid out – banks of eight tracks – eight rows of everything. All the popular grid controllers that work with Live, such as Push and the Launchpad, follow this formula and provide fast shortcuts

MTF Step-by-Step Reducing tracks in Live (cont'd)



13 Repeat this process with the other clips, applying an envelope to select each relevant zone. After this, whenever you launch a clip, it'll always select and play the correct sound for the part.



14 The great thing about this is it all takes place in one track – and you have many chains in one rack, all with different instruments (and effects, if needed). Goodbye, one-sound-per-track!



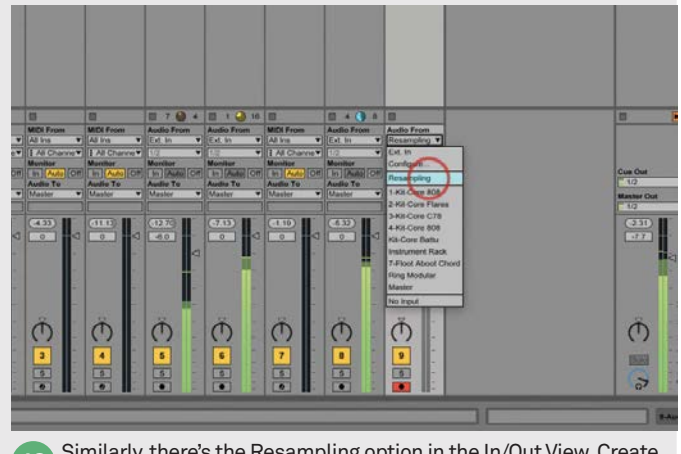
15 Most of this is about having the freedom to reorganise clips in Session View, to put similar instruments into one track in a logical way, and not have to worry about whether the right sound plays.



16 That's very cool, but we can try another way now with that track. Right-click on it and choose Freeze, and the entire track goes blue. Right-click again and choose Flatten.



17 Now every clip in the track has become an audio clip, with any effects or automation embedded, ready to be freely moved around other audio tracks in the set.



18 Similarly, there's the Resampling option in the In/Out View. Create a sample sub-mixing everything currently playing: a really fast way to save tracks, especially with like-minded parts, like percussion.

to move between banks vertically and horizontally. Sometimes I wish there was a Push expansion module, so I could see more tracks and clips at the same time. Using banks just isn't the same for me, but for now, we're kind of stuck with it. I know people who love using a load of tracks, and I do that sometimes, but generally, for me, it's pretty minimal. Visually, I find it much easier to keep a handle on where everything is if I use fewer tracks. It's not always appropriate or necessary to make the effort to reduce your track count. It's more likely you'll want to do it for live shows, in my experience. Even if you don't need to do it, though, play along – it's good practice, and might give you some new insights into how to work with your tracks, your compositions, and your mixes. **MTF**



Highasakite

MTF Interview Highasakite

After huge critical acclaim for 2014 album, *Silent Treatment*, Norwegian indie-rock band Highasakite are back with a third album, *Camp Echo*. **MusicTech** gains unique studio access to the recording via band members Trond Bersu and Kristoffer Lo and the album's co-producer, Kåre Christoffer Vestrheim…

Highasakite – Ingrid Helene Håvik (vocals), Trond Bersu (drums), Øystein Skar (synths), Marte Eberson (synths) and Kristoffer Lo (guitar, percussion and flugabone) – have a unique sound that mixes traditional and technological instruments. It's quickly led to them becoming one of the biggest names on the indie rock and pop scene. The band's second album, *Silent Treatment*, spent no less than 40 weeks in their home charts and won them two Norwegian Grammies (Pop Group Of The Year and Composer Of The Year). They've since spent a couple of years touring around the world headlining and supporting some huge names, including Of Monsters And Men and London Grammar.

MusicTech caught up with Trond and Kristoffer plus co-producer Kåre Christoffer Vestrheim as they put the finishing touches to their third album, *Camp Echo*.

MusicTech: When you go into the studio as a band to record, do you have a plan or philosophy?

Trond: I think individually, we are five people with strong musical voices, but when we mix that together, we end up with the sound of Highasakite. I´m inspired by new production methods and always try to challenge myself, and always move forward. For the new album, I programmed all of the drum parts in Ableton, and used some drum machines and synths. Mostly Elektron Machinedrum, Casio Rapman and samples in Ableton.

Kåre: Mike [Hartung, co-owner of Propeller Music Division studio in which the album was recorded] and I have always been extremely strict about what leaves our studio. We spend thousands of hours tweaking the productions, so it's always as great as we can possibly get it.

Trond: I think our sound has changed a bit since the first album (*All That Floats Will Rain*). Back then, we used a lot of acoustic instruments like a zither harp, lots of percussion, bass and guitars. Now everything has become much more electronic, but I still feel we have kept the sound of Highasakite, somehow. The sound of the electronic instruments still has an organic element to it, and we're spending a lot of time trying to find the right sample for our sound for the song.

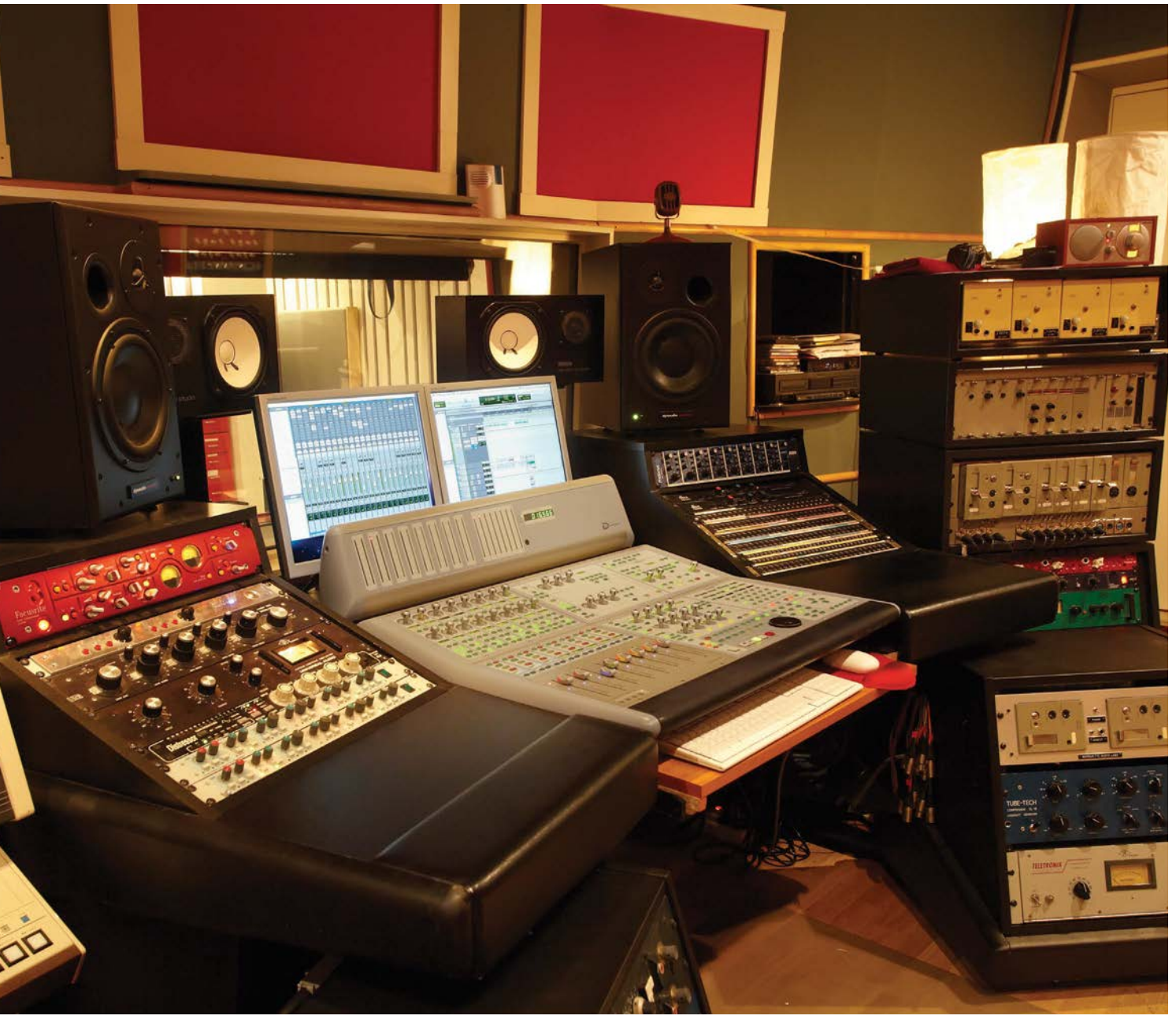
MT: What about specific areas – what do you each bring to the recording?

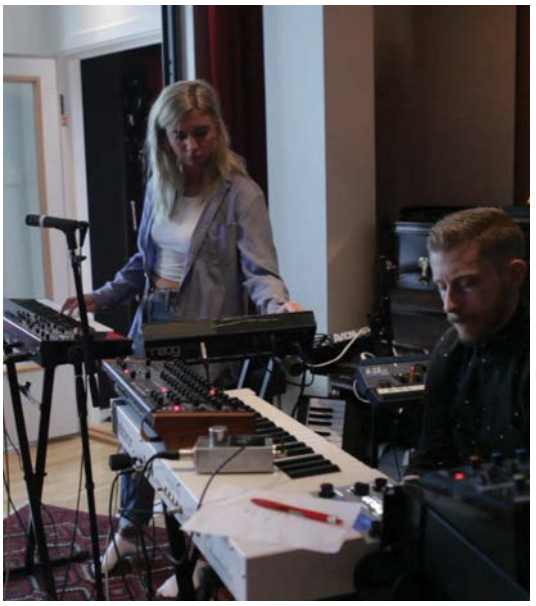
Above: Despite bringing in more electronic gear for the new album, *Camp Echo* is still very much a Highasakite album

Right: Propeller Music Division Studio plus (clockwise) Kare; a DSI/Sequential Prophet-6; Øystein; Ingrid; Kare at the computer; Marte and Øystein

> "I'm inspired by **new production methods**. I always **try to challenge myself**, and **move forward**"

















"You can **emulate all kinds of things** with **samples** and **plug-ins**, but nothing beats the **real deal**"

Above left: Vocalist Ingrid Helene Håvik, pictured with trusty Shure SM7, starts off Highasakite's creative process by bringing demos of song ideas to the rest of the band

Other pictures: Push gets a workout; the band and Kare; and the producer (bottom) makes some final tweaks

Trond: As a drummer, I think it's cool to think of the drums as a melodic instrument. Trying to find drum parts that almost could be a song by themselves. I use lots of toms and samples to fit the harmony of the song.

Kåre: It's important for me that the music has layers and depth, new stuff to discover every time you listen to the track. Even in very sparse, orchestrated pieces, I try to have something that's a bit out of the ordinary. At the same time, I am especially focused on keeping the vocal in the spotlight – both sonically and in the performance. The person fronting the band has to connect with the listener.

MT: Kåre, how do you see your role during the recording process?

Kåre: One of my favourite things about studio work is the creative freedom to experiment. I love making sounds that I've never heard before. I see all the synths, guitars, xylophones, effect boxes and all the other instruments in the studio as one big orchestra – blending them together in new ways often makes a production unique-sounding. I hardly ever use presets. Sonology is very important to me. What function do the different parts, instruments and dynamics have in the bigger picture of a whole song, and in the even bigger picture of a whole album? Sculpting the song, the arrangement, the sound and the performance is a large part of what I do.

MT: Tell us about the gear setup in the studio?

Kåre: Propeller Music Division is a modern multi-room studio complex with a large recording room, a surround mixing room, two smaller mix rooms, a composition room and a world-class mastering studio. That didn't happen overnight. Our first studio was a tiny room located down a dark driveway, in an abandoned building known as The Chocolate Factory. It was from those humble beginnings that we managed to eke out some of Norway's most popular albums of the time. When the building was finally condemned, we were forced to find a new location, which is where we have been for the last 12 years – an old mill from 1857.

We don't have any equipment that is THAT old, but we have been collecting old, vintage and fun stuff since our earliest days. We also build mic-pres,





EQs and compressors ourselves. Sometimes, you need something extravagant, vibe-y and vintage, and sometimes you need something that sounds beautifully crap. We work in Pro Tools, but from time to time, we also do the tracking on our 24-track two-inch machine. It's great fun and a great sound!

Trond: The gear I used for the new Highasakite album was Ableton Live and Ableton Push, Elektron Machinedrum, Casio Rapman, Roland SPD-SX and some live percussion. When I'm producing music for others and myself, I also use the Access Virus TI2 synth, and Sylenth a lot. I think Native Instruments and Vengeance have some nice drum-sample packs for electronic music, but I think you should mix that with some organic drum machines and tweak the sounds a bit, so it becomes more organic.

Kristoffer: Being a guitar and horn player, those instruments are always very close when I'm working in the studio. I haven't yet found my way into working with synths as a creative tool in the studio. I still like acoustic instruments a lot, which is okay, since we have two amazing synth players in Highasakite. One of my favourite creative tools in the studio is a reed organ, or a pump organ. I love the way it can drone away.

On the *Camp Echo* album, my main guitar was a Fender Bass VI. Most of the guitars are actually done directly into Pro Tools, skipping the amp. We just liked the sound of it for this album. And, of course, a bunch of effects like harmonisers, repeaters, distortions… basically, anything that can make the guitar sound a bit different.

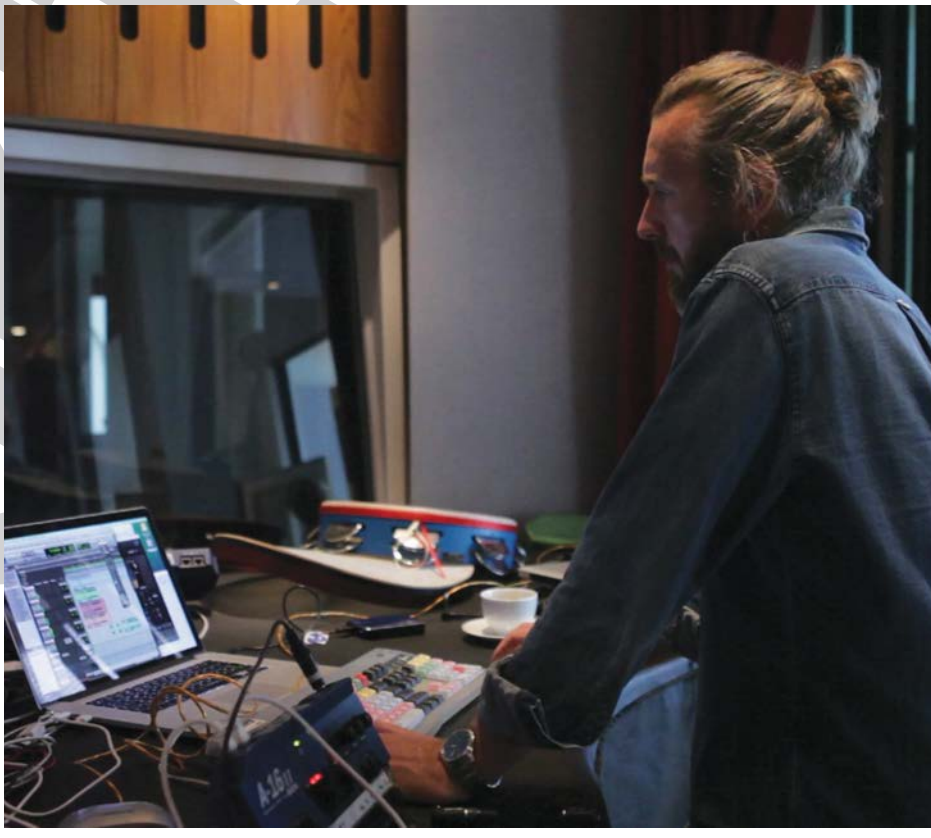
We had some fun in the studio recording a dulcimer for the song *My Own Disease*. Kåre has one lying around at Propeller Studios, so I tuned it and turned out pretty cool. You can emulate all kinds of things using samples and plug-ins, but nothing beats the real deal!

MT: Do you have any specific outboard gear that you use, or want to use?

Trond: I really love the Soundtoys stuff, Valhalla Shimmer as a crazy reverb and I think that Ableton is the most creative software out there. You can easily do whatever you want, and it's so hands-on. I don't have a lot of outboard stuff, but I really want some nice compressors and preamps. On the other hand, it's nice to be able to work on production wherever you are in the world with a laptop.

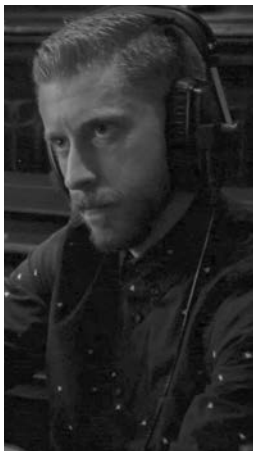
Kristoffer: I'm an outboard sucker. I like to put a bunch of pedals between my guitar (or horn) and amp, instead of doing the effects later in the process. I also like to have outboard preamps and EQs, just to get feeling of tweaking something with your fingers. For the *Camp Echo* album, my Eventide PitchFactor, MASF Repeater and Electro-Harmonix Memory Man was present almost all the time. But, it's always fun to play around with Soundtoys.

Kåre: We tend to spend a lot of time on sound before it goes into the computer. Which means using all sorts of stompboxes and vintage stuff to create the sound we're seeking. I love tracking through a Telefunken V76 or a Neve 1081 or 1073. Tube-Tech CL 1B is usually also in the chain somewhere. On Highaskite singer Ingrid's voice, we use a →













Above: Highasakite have an open-door policy when it comes to instrumentation: vintage synths happily rub shoulders with acoustic instruments and MIDI controllers

Right: Kristoffer used the Fender Pawn Shop Bass VI – a six-string baritone with a vibrato – for much of the guitar playing on the new album

> "After working with **hundreds of tracks**, having **only 16** starts to sound **very attractive...**"

plain-old Shure SM7 through the 1081, with minor EQ and the CL 1B.

We have a great selection of keys in the studio, from Mellotron to Jupiter-8, Prophet and SidStation – all colourful in their own way. In the box, it's common for us to use Echoboy and Altiverb, plus we have great packages from Waves, FabFilter, iZotope, Steven Slate and Brainworks that are all used on a daily basis.

MT: What is on your wishlist, studio gear-wise?
Trond: Some Neve 1073 pre amps, the Roland SH-101 synth and a nice pair of monitors, like the Barefoot MicroMain 27.
Kristoffer: I would love to have some more preamps and maybe a nice 'lunchbox'. We just made some DIY Neve preamps and some DIY UA 1176 compressors that are working great. An API 512c maybe. And some more microphones would be fun. Right now, I'm using a Pearlman TM-2, Coles 4038 and the Sennheiser MD 421 a lot, but it would be nice to have a Neumann U 68, this after all being a wishlist.
Kåre: Right now, I'm looking for a 16-track two-inch tape machine and a smaller vintage console to do smaller tracking sessions. After working on mega productions like the Highaskite album with hundreds of tracks, having only 16 starts to sound very attractive. You have to commit!

MT: What would you like to see developed in terms of studio technology and why?
Kåre: A Pro Tools rig that never crashes...

MT: So how does it work with the five of you (plus producers) in the studio? How does a track come together?

HIGH ON ADVICE

MusicTech: You must have picked up some experience working in the music industry over the last few years. Any advice you can pass on?
Kåre: Follow the vision you believe in, not the one everyone else is talking about or that is the current hype. Having too many references will make you lose focus and end up with something that has nothing genuine to say.

MT: And what about your advice for working in the studio?
Kåre: A big part of being a producer is to get to know the artist. In order for them to deliver something unique, she or he has to relax and be able to focus only on the music. It's all about creating a vibe and a space that inspires artist to perform their best; whether it be candlelight, good tea, Persian rugs or film projection on the studio walls – whatever it takes. Everybody that works here in Propeller is aware of this and contributes in different ways to this. A relaxed vibe is important, but that's not to say that I don't push the artists when necessary. Some artists need to be challenged and some need to be forced out of their comfort zone. That's a tightrope I sometimes have to walk, but almost always, the artist appreciates the end result. It's important to play ball with the client. After all, it's their album and I don't wish to dictate anything to anybody.







Trond: For Highasakite, Ingrid makes the demos with the melodies and chords. This time, I got the demos at an early stage and tried to come up with drum parts, just to have something to build on when we started working in the studio.

Actually, a lot of the demo parts were used on the record. It's strange, but very often, the first idea is the best.

Kristoffer: Yes, Ingrid writes the basics of the songs like melodies and lyrics, and she brings the demos to us as band. For both *Silent Treatment* and *Camp Echo*, most of the arranging has been done together with Kåre, either in the studio or at his farm outside of Oslo. It's been fun working on the *Camp Echo* album, because we've been touring so much and talked so much about music that inspires us, so when we finally got to the studio, we had a lot of the same references.

MT: So tell us more about *Camp Echo*?

Trond: It's an album about war and terror, and I think that's something that we can't ignore now. It's not a political album, but more a view on the world. We had a lot of songs when we started recording, and used a lot of time to choose the right ones for the album.

I think what's cool about the album is that it's electronic, but you'll still feel it's a band and not one DJ guy producing it all, like a lot of stuff these days. We have used a lot of fantastic vintage gear from compressors, preamps, vocoders and instruments, mixed together with more modern stuff. We used some Elektron stuff like the Machinedrum and SIDStation. Also the Roland Jupiter-8, Roland Juno-60 and Super Jupiter. Even some modular synths… I think the sound of the album is timeless, it could have been made in the 80s and now.

> "The album is **electronic**, but **you'll still feel it's a band** and **not one DJ guy** producing it all"

Above left: Øystein is one of two synth players in the band. Right: Ingrid takes a break and (below) Trond and Kare discuss some finer production points…

MT: What is next for Highasakite?

Kristoffer: Hopefully, things will get busy. We're starting off the release tour for *Camp Echo* in the UK, playing Brighton, London and Manchester, and Dublin and Amsterdam. Then there's summer festivals, which are always fun. We're doing Roskilde festival in Denmark, some festivals in Germany and then five nice festivals here in Norway. And then we'll see what happens next.

Trond: We're rehearsing a lot these days, and in May, we'll start touring with the new album. We're playing some really cool festival headline shows in Norway this summer at Øya, Pstereo, Bergenfest and Parkenfestivalen. We're also playing Latitude festival, Roskilde and a bunch of other festivals in Europe!

Highasakite's new album, *Camp Echo*, is out now on Propeller Recordings. MTF


FIRST OUR WORLD WAS A DISC. THEN, A SPHERE. TURNS OUT, IT'S TWO DISCS!
THE POWER OF MUSIC!



th•mann
MUSIC IS OUR PASSION

Ableton Live Workshop

On the disc
DVD ROM
Accompanying project file included on the DVD

How to integrate Ableton Live with your iOS devices for control and music

Ableton Live can enjoy a rich and varied relationship with your iOS devices – by syncing with Link, by using them as hardware controllers or as sound sources. Here, **Martin Delaney** explains how…

Since 2007, iOS has matured into a sophisticated mobile-computing platform with the hardware to back it up, and some highly evolved apps for music recording, programming, and control. Although Ableton itself has never fully embraced the mobile platform and graced it with an iOS version of Live, its influence is felt – in apps that attempt to provide a Live-like experience, apps that support the Ableton Link sync protocol and apps that export their files as Ableton Live projects, ready to open on our computers.

It's true that it's a little inelegant at times to get audio in and out of iOS apps; it sometimes requires a few extra steps, but the workflow is getting smoother all the time, and there are apps which make iOS audio management much easier, like AudioCopy, Audiobus, and AudioShare. Apps like Apple's Voice Memos let you Airdrop recordings, which is fast and convenient. There's also no reason why you can't continue to record audio straight out of your device into Live, courtesy of your audio interface. If you're using MIDI or Link to sync everything as well, there's every chance of getting a very neatly cut, ready-to-use clip straight away. Once you wade through the options and the connection possibilities, you can build a Live/iOS rig that includes MIDI and audio coming from both boxes, in a stable and repeatable setup. The Mac Audio MIDI Setup utility is a fantastic Swiss Army knife for audio and MIDI routing, and if you've never explored it, it's well worth spending some time to poke around in there. I find that using an ad-hoc connection rather than a router invariably works better.

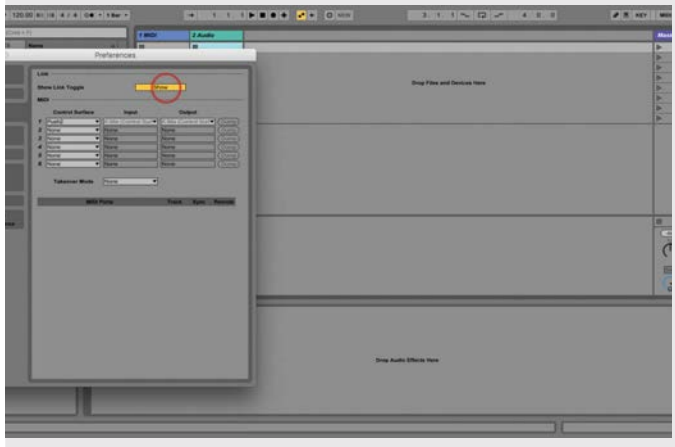
Whatever else you might feel about Apple and iPads and iPhones, the multi-touch functionality works undeniably well and the range of apps available is amazing. A full-sized regular iPad, or ideally the larger iPad Pro, is a great synth and DAW host – but also makes arguably the most powerful and flexible MIDI control surface on the planet. As Live users, we can take advantage of touchAble, which is specific to Live, and provides very deep device control, as well and clip and scene launching, a mixer, a keyboard, pads, as well as a custom configurable area. Plus, touchAble is available as separate iPhone and iPad apps, and the device controls are an in-app purchase that's well worth having. This one's really hard to beat for Live use; put TouchAble and Push together, and you've got it all. There are more 'DIY' controllers, such as Lemur and TouchOSC, where you can use a library of objects to build your own personalised



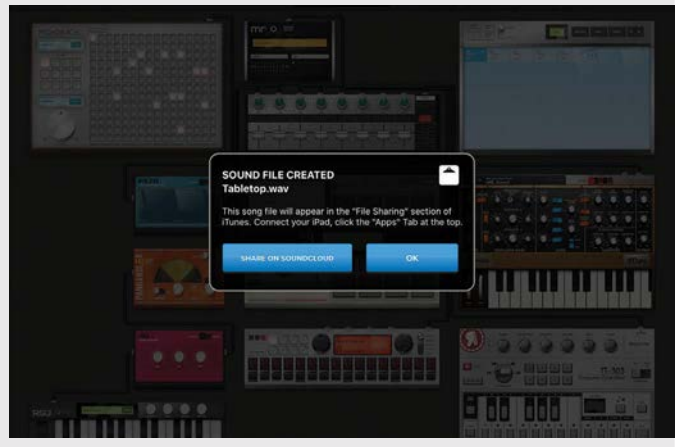
MUSIC ON THE MOVE

Most iOS music apps run on both iPad and iPhone; and for me, that's really important for true mobility, and the feeling that you can use the same tools to make music wherever you go. Even the controller apps such as Lemur, TouchOSC, and touchAble work on iPhone, and Gadget and Blocs Wave will let you create Ableton Live projects from your iPhone or iPad, which still blows my mind a bit. Maybe I'm just easily impressed! Having said that, at the moment, I believe the 12.9-inch iPad Pro is the best possible option for iOS music.

MTF Step-by-Step Linking devices to Live



01 Ableton Link is an awesome tool for syncing iOS apps to each other, and to Ableton Live, over Wi-Fi. I'm not going to talk about it too much here, because we have a separate tutorial covering Link.



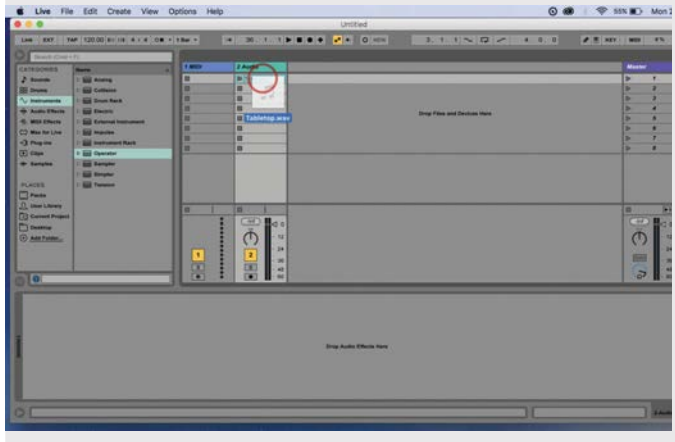
02 You might want to include audio from your iOS app in your Live productions. How this works will depend on the app, but many of them rely on iTunes File Sharing.



03 Let's use Tabletop as an example. Here, I have a little sequence that I want to get into Live. The first step is to save that as an audio file, stored on the iPad itself.



04 When I connect the iPad to a Mac, I can access that audio file from within iTunes. Other apps might have different sharing options – they don't all work the same way.



05 There won't be any format issues: you should be able to drag the resulting audio file straight into Live. If you have a commonly used 'import' folder, make a shortcut to it in Live's Browser.



06 Alternately, I can take stereo out from the iPad headphone jack and connect it to my soundcard, configure the inputs in Live and record the audio in, then crop and warp it.

control surface. Both Lemur and TouchOSC run on either iPhone or iPad with a single purchase. If you're open to something more idiosyncratic, particularly if you're more interested in performance and gesture-based control, you should look at AC Sabre, an iPhone app which combines onscreen buttons with the iPhone's gyroscope and accelerometer to send MIDI notes and CCs, creating a unique 'instrument' approach; if you've spent any time configuring game controllers like the old Wii Remote Control for MIDI, you'll take to this, and also appreciate that it's a little easier to get advanced results. I guess you could use AC Sabre with an iPad, but it'd be like waving a big expensive glass dinner plate around. For day-to-day Live control, I use Push; for mobile purposes, I use touchAble on the iPad Pro; when I need something more boutique, I build it in Lemur (also on the iPad Pro). The other great advantage →

MTF Step-by-Step Linking devices to Live (cont'd)



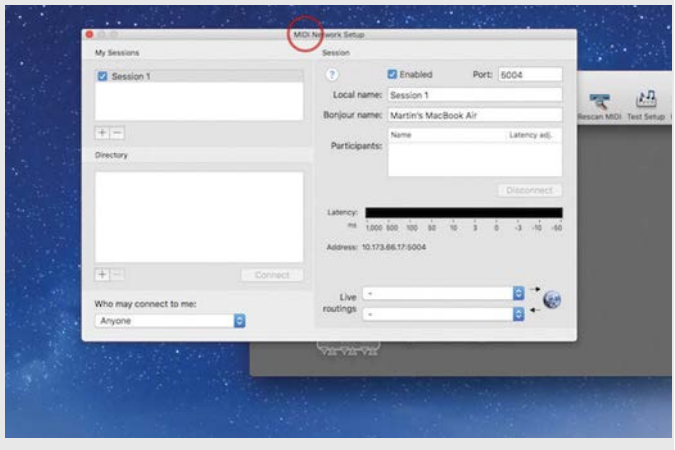
07 iPhone 7 users will need to use their lightning-to-audio adaptor to accomplish this task. It's not such a big deal; the adaptor's in the box with your phone, behind the earbuds.



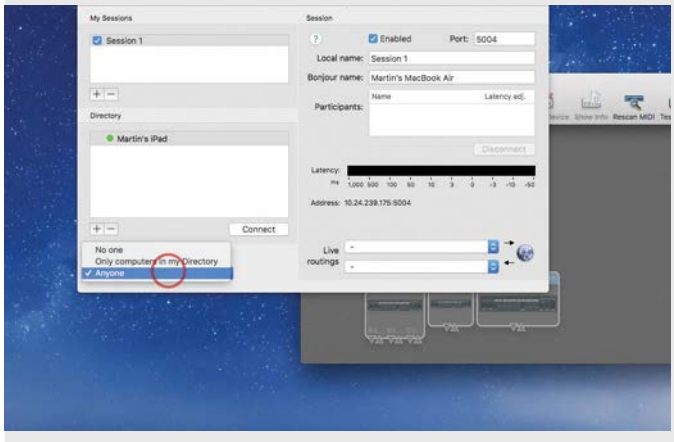
08 There are some wonderful iOS synths, like Moog's Model 15, for example. If you want to send MIDI from Live to that, you'll need some way to get MIDI from your computer to the iPad.



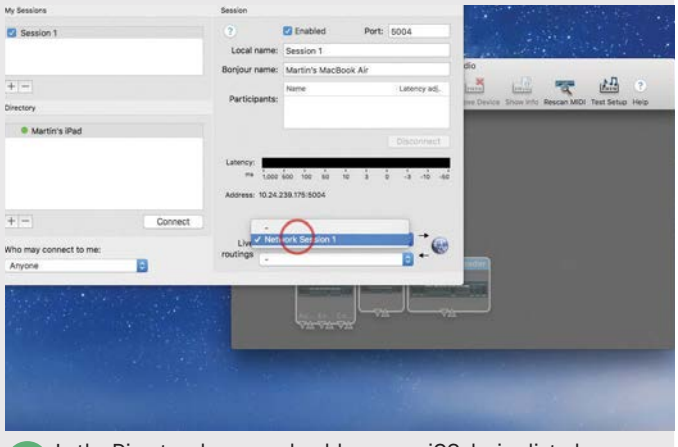
09 You could try physical connections by using a hardware device like the iConnectivity iConnectAUDIO4+. It will connect to and power a computer and iOS device and transmit MIDI between them.



10 We can also send MIDI over Wi-Fi. On macOS, you do it via the Audio MIDI Setup utility. Get both devices on the same network. Open Audio MIDI Setup, then open the MIDI Studio window.



11 Click the + sign under the 'My Sessions' box to add a new session, and switch it on. Rename it if you want to. At the bottom, where it says Who may connect to me?, choose Anyone.

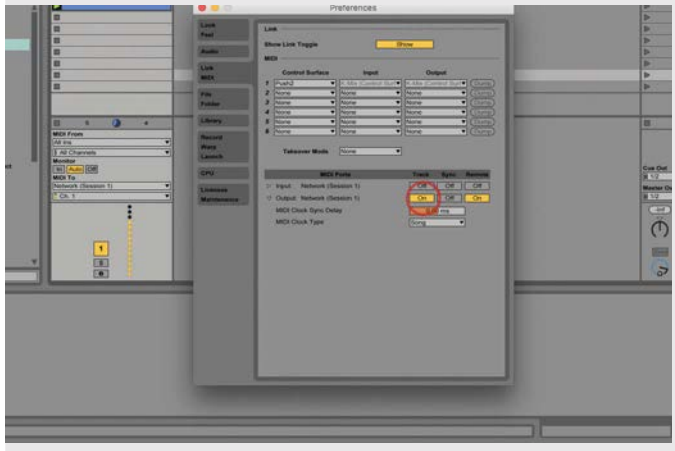


12 In the Directory box, you should see your iOS device listed. Choose it and click Connect. In the Live Routings section, choose Network Session 1 as the output.

→ with the iPad as a control surface over hardware is that if you switch DAWs, it follows right along – Apple's free Logic Remote app is great for Logic Pro X users, for example. In a further instance of Live/iOS integration, there are a few apps with the ability to export files as Live projects – e.g. Korg's Gadget and iKaossilator, and Novation's Blocs Waves – bringing Live and iOS much closer together.

This process will get you a Live set containing audio clips exclusively, with tracks and scenes as applicable, but no MIDI. Interestingly, Korg has announced Gadget for Mac, which is completely compatible with the iOS versions, but with the layout optimised to use larger displays (and obviously no touch screen). I hope this doesn't threaten the included Ableton project support on the iOS versions: that

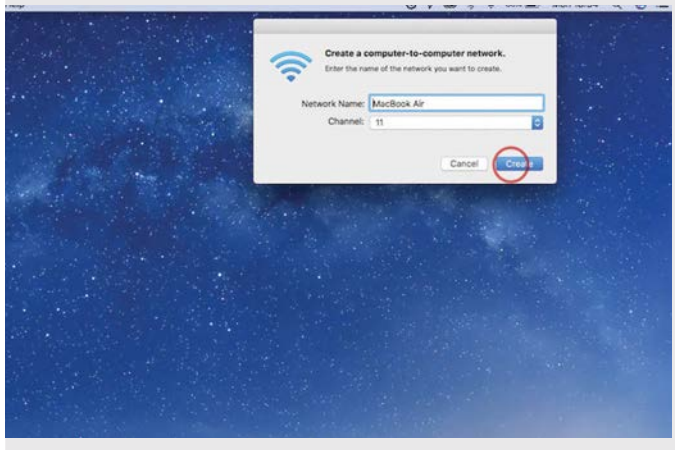
MTF Step-by-Step Linking devices to Live (cont'd)



13 You can choose Network Session 1 as a MIDI output in Live's Preferences, sending track, sync, or remote information, or all three. You'll need to select that in your MIDI track output as well.



14 On your iOS device, launch your selected app and make sure Network MIDI is on. Now when you start Live running, with a MIDI clip and some notes, you should hear the iOS synth play.



15 If it doesn't work, try creating an ad-hoc network from the computer rather than going through a router. Do this in the Wi-Fi menu here, and then join it from your iOS device.



16 Once you've mastered that kind of setup, it should be relatively easy to figure out how to do it the other way round – to send MIDI from your iOS device to Live.



17 Your iPad or iPhone makes a great multi-touch controller, either used alone or alongside something like Ableton's Push. Between them, you can do just about anything.



18 If you prefer to use cables, controller apps such as Touchable and Lemur support MIDI via the Apple Lightning cable straight to the computer, which might feel more secure in some situations.

would be a real shame. According to the announcement, though, the desktop version will support Ableton Link, just as the iOS ones do. Also, it seems the Gadget synth and sampler instruments will be installed as separate AU/VST plug-ins, so you can even run them inside Live.

This is the best possible example of blurring the lines between iOS and Mac OS. I'm not a Windows user, but I'd like to see Gadget make it over to Windows as well, so it becomes a true cross-platform DAW that everybody can use. Just to be extra-thorough, a purchase of Korg Gadget also includes a free Ableton Live Lite download. iOS is a friend and playmate for your Live setup, and can merge perfectly with Push or any other hardware controller – make the most of it! MTF

Ableton Live Workshop

Control your hardware drum machines and synthesizers from Ableton Live

Everybody reading this has, or will eventually have, a hardware synthesiser or drum machine that they want to hook into their Ableton Live rig. **Martin Delaney** explains how to link it with Live…

On the disc
DVD ROM
Accompanying project file included on the DVD

Live works brilliantly well as the hub of a studio or stage rig that features hardware as well as software instruments. And it goes beyond that, into effects, VJ equipment, lighting, even guitar amps. For the walkthrough steps here, I've use the Novation Bass Station, which is a classic 1990s rack synth (and there's still a current v2 of the keyboard Bass Station available). The Bass Station has MIDI in and out, and audio outputs, so it's a good example of a typical keyboard set up. More modern instruments might connect with USB for audio or for MIDI, and include their own dedicated editor software, which will affect how they integrate with your DAW. Our walkthrough focuses on Live's External Instrument device, but don't forget there's the External Effect device as well, which configures in the same way, but is designed to feed audio from Live out to, and back from, hardware audio effect processors.

While you're in Live's MIDI Preferences, make sure you've configured the correct inputs and outputs, and disable any that you don't need – this can solve problems with more elaborate hardware rigs. You can also choose to have Live sync from the hardware, if you think you'll get better results that way. It's also possible to build a rig that includes MIDI clock travelling between hardware and software, as well as iOS devices responding to Ableton Link on a wi-fi network or ethernet. Instrument racks let you build complex setups that use sounds coming in 'live' from hardware instruments, as well as sounds from plug-in synths, even stacked and playing together. I have external instrument device presets that load with limiters and compressors in the chain; nice and easy to recall, and easy to move between different machines if necessary. I love using hardware synths for their sonic character and distinctive workflows, but other than capturing the raw sounds, I prefer to do everything else with Live as working on a computer display is faster, easier and more controllable. You might disagree and have your own level of hardware/software preferences, but that's the point – with Live you can set it up the way you want it. I mentioned arpeggiators for beats in the walkthrough, and that's something I've been playing around with a lot lately, especially for triggering beats in real time from a keyboard. Create an instrument rack with, say, four chains of arpeggiators, and set each one to trigger notes over a certain range – most likely just a single note, so each arpeggiator is triggering only one note on your drum machine. Try it - it's very cool, even with software drum kits.

Sync isn't always about MIDI, if you're using older gear, or some new modular components, you might find CV/gate (control voltage) is required instead of MIDI. Live can't



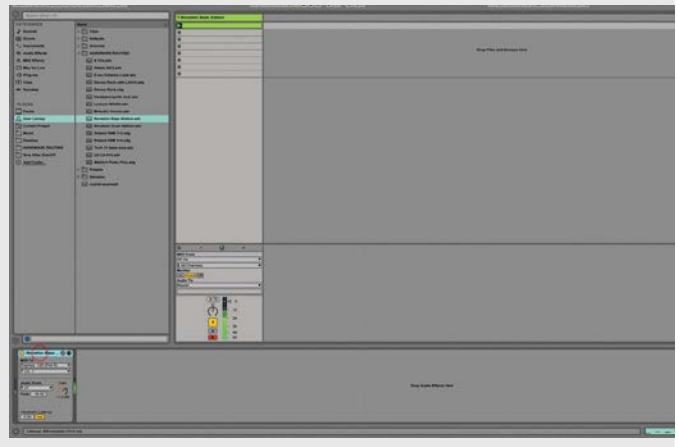
HARDWARE V SOFTWARE

You can adopt extreme positions if you like – only using hardware vs only using software. Both are valid, but the truth is combining the two will always be the best option. You're always going to need a computer at some stage in the production and recording process, and your sound and working methods will benefit from adding some choice bits of hardware to your setup. You can do everything with a computer, sure, and that used to be a really exciting option, but these days there's so much great (and affordable) hardware around, it's a waste not to use it!

MTF Step-by-Step Control your hardware…



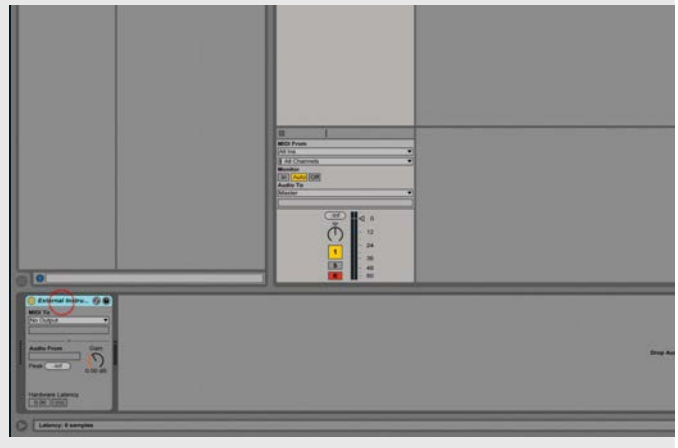
01 For this tutorial you'll need some kind of external hardware synthesiser module or keyboard or a drum machine, an audio/MIDI interface and, of course, Ableton Live running on a Mac or PC.



02 Our aim here is to use Live MIDI clips and effects to trigger notes and control messages to our hardware and to receive, process, and record the resulting audio coming back in.



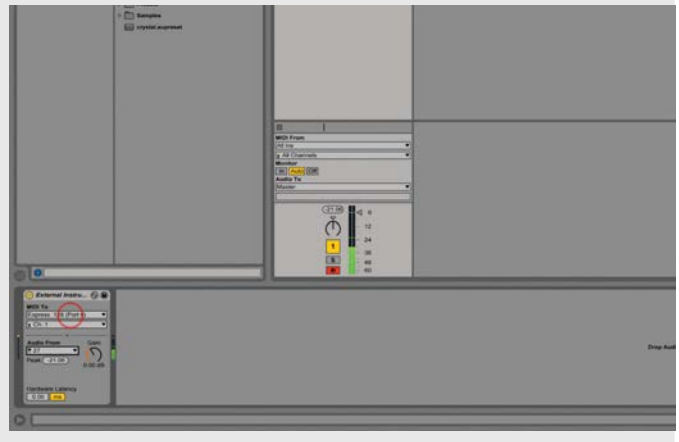
03 Modern synths use USB as well as standard MIDI/audio, but let's work with the more generic/classic options. Connect your hardware's audio outputs and MIDI inputs to your interface.



04 Open a new Live set, go to Session View and add the External Instrument Device MIDI effect into a MIDI track. This device will send MIDI but also receive audio within one track.



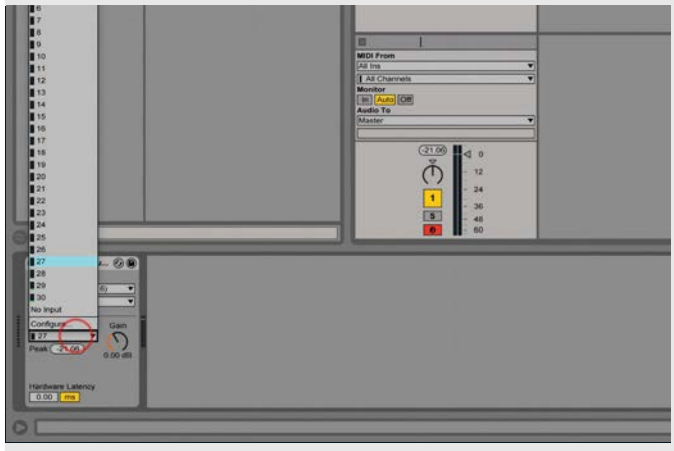
05 Open Live's Link MIDI Preferences and make sure your interface is configured correctly, receiving MIDI Track, Sync, and Remote, messages from Live. Close Preferences after you're done.



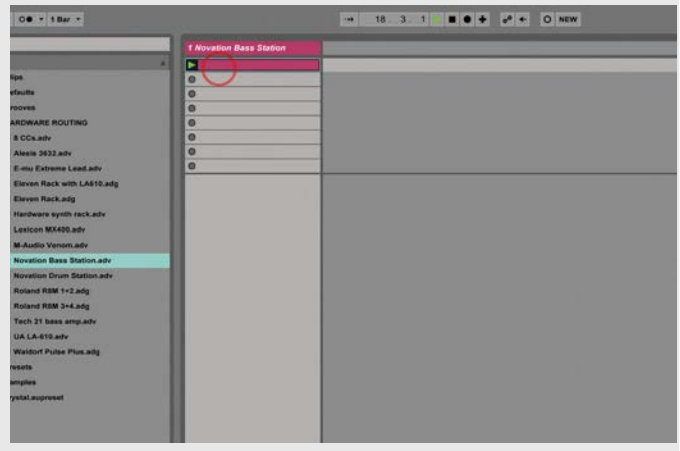
06 As we're using the External Instrument device, set up the routings there instead of in the In/Out View. Choose the MIDI channel required by your hardware (you might have to look in its manual).

handle this directly, but you can buy a dedicated converter device – Kenton produces a range of these, including the Pro Solo MkII, or use something like the Arturia BeatStep, a very well spec'd hardware sequencer, which includes CV as well as MIDI output. There are devices that use other means of communication, like the Teenage Engineering Pocket Operator range which will sync to an audio signal sent from your DAW. It's crude, but it works. When you're sending MIDI from Live's MIDI clips, it's not just about notes and CCs; a MIDI clip can send Program and Bank changes, which are used by hardware to load different presets or different sequencer patterns. You never have to load the wrong sound by mistake – whatever order you play your clips in, the instruments are always going to load the right sound. Some MIDI hardware, the Elektron MachineDrum, for example, includes an additional MIDI Thru connection, →

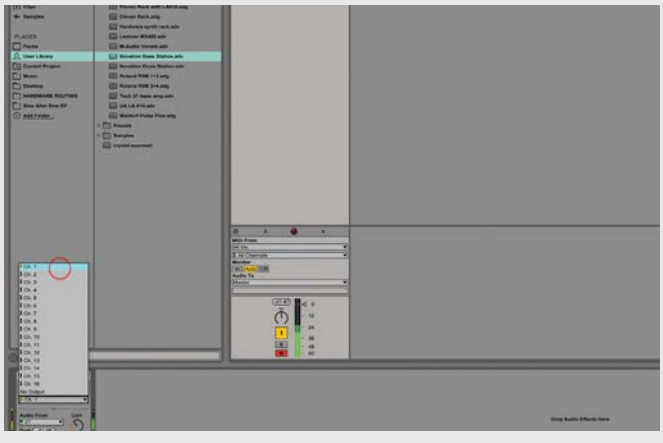
MTF Step-by-Step Control your hardware… (cont'd)



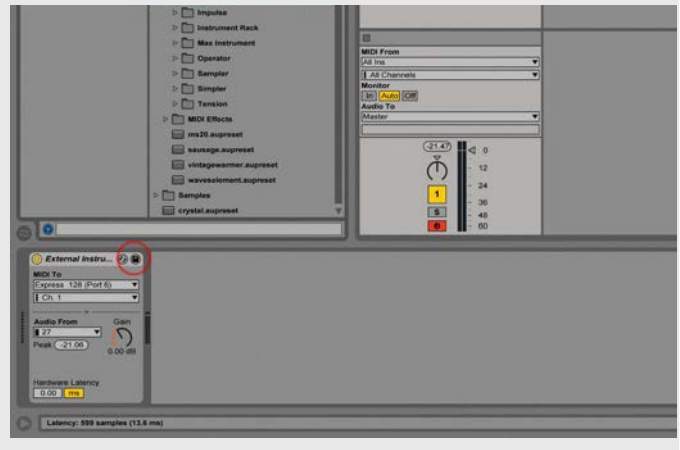
07 Then choose the correct inputs on your audio interface to receive audio back in. You can set input gain levels within the device. Don't let those levels peak red!



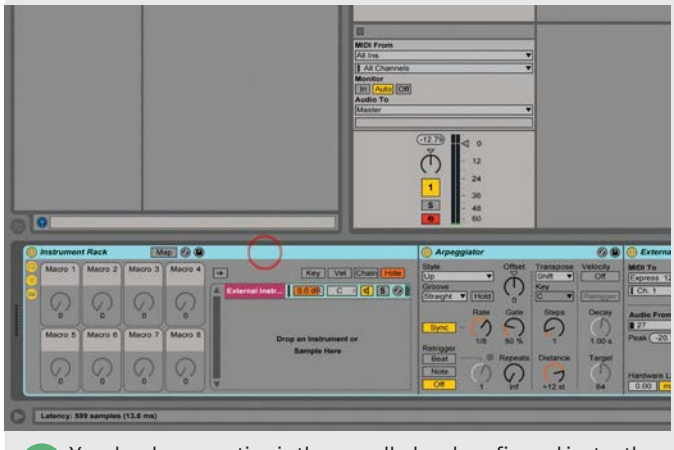
08 Once this is setup you can use notes in a Live MIDI clip, or play a connected keyboard, or even Live's virtual MIDI keyboard, to send notes out and see if you're getting a response.



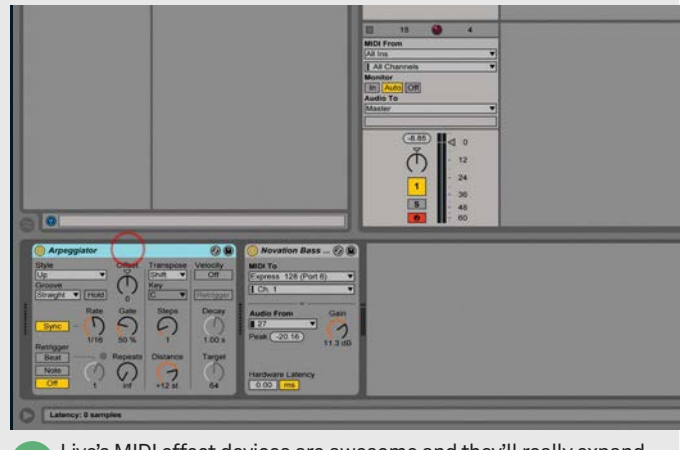
09 A lot of MIDI hardware has an activity indicator, so you should be able to see if it's receiving MIDI, if you can't hear anything, which is useful for trouble shooting.



10 Once this is working smoothly, save the audio effect device as a preset, named after the piece of hardware (that's how I do it, anyway). That means it's easy to recall the setup at any time.



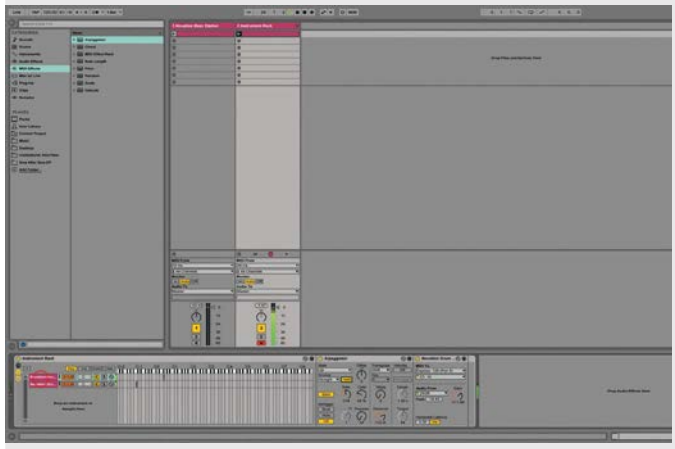
11 Your hardware routing is then recalled and configured instantly just like an instrument plugin. If you put it into an instrument rack with Cmd-G, you can then add MIDI and audio effects too.



12 Live's MIDI effect devices are awesome and they'll really expand what you can do with hardware. The arpeggiator and chord devices can add a lot of movement to keyboard parts.

→ facilitating the daisy chaining of more hardware, effectively using the first device as a MIDI hub. This simplifies the addition of more bits of hardware without kludging around with hubs or MIDI mergers and splitters. By using hardware sound sources in your Live projects, you're compromising the mobility of your project. You can't simply close your laptop, go to another location, pop it open, and have all your sounds in place – the hardware isn't there! There is a workround, though: you can freeze the tracks which will require Live to scan through the project in real time to create an audio file 'preview' version of each external instrument track. Now you can open the project at another location or on another computer, and you'll have the intact audio from your synths or whatever, you just won't be able

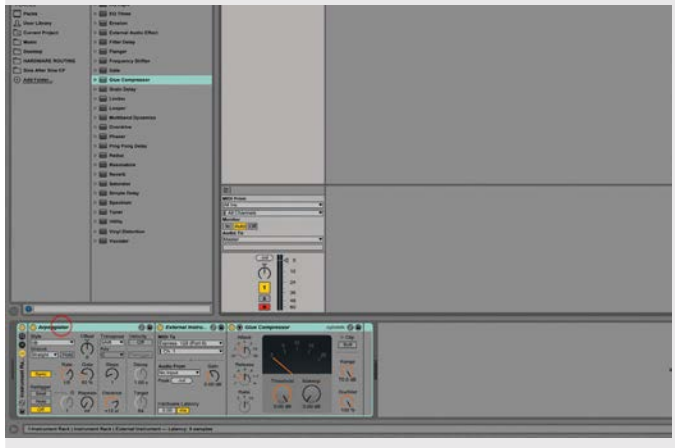
MTF Step-by-Step Control your hardware… (cont'd)



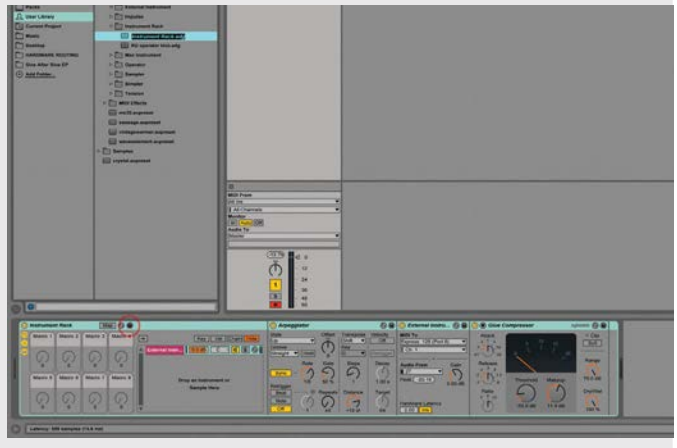
13 The arpeggiator can also do some interesting things with drum machines, giving you a new way of creating beats and breaks. A few arpeggiated drum parts in a rack can sound amazing.



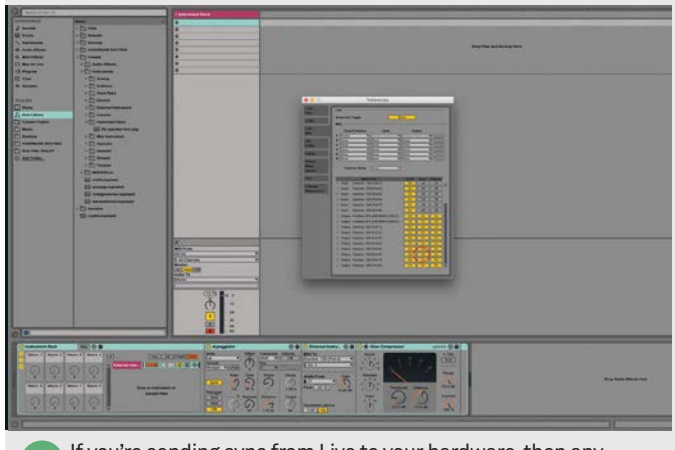
14 Of course you can apply audio effects to the incoming audio. You might use more 'showy' effects such as filters and delays, or else more practical ones like EQ, utility, limiters and compression.



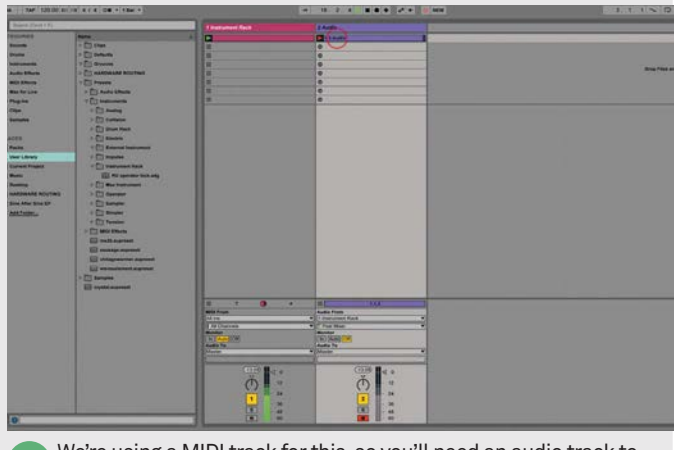
15 You can end up with a recallable rack that contains routings to and from your hardware, MIDI effects, audio effects and any third-party plugins you want to add as well.



16 You can then save all of this again and then you can recall it all later, ready to rock – as long as you remember to switch on your hardware naturally!



17 If you're sending sync from Live to your hardware, then any time-based parameters on it will follow along with Live's timing, whether they be beats, onboard arpeggiators or delay effects.



18 We're using a MIDI track for this, so you'll need an audio track to record it. Create a new audio track and use the In/Out View to select the source track, then arm it.

to edit or interact with it in any way. If you want to create permanent undoable audio versions of the tracks, use the Flatten command after freezing, to embed those parts into an audio track and discard the original MIDI material and hardware routings. Because Live's scanned through the project for the freeze process, flattening is a fast final step. On the occasions when I've done this, I've either created a new version of the project, containing the frozen and flattened material, or otherwise I've duplicated the external MIDI instrument tracks before freezing and flattening them, so I have every option for going back to them later.

There can be bumps along the way, but if you have the will and the gear to make the connections, you can get Live talking to almost anything… MTF

Ableton Live Workshop

On the disc
DVD ROM
Accompanying project file included on the DVD

Attack, Decay, Sustain, Release, in Ableton Live – what ADSR can do for you!

Whether you're familiar with the trusty synthesis principles of ADSR or not, you can get to work immediately using them in Ableton Live, **Martin Delaney** fills the envelope full of music…

ADSR is a commonly used term in synthesis, and sound design – it stands for Attack, Decay, Sustain, and Release. These are held to be the important characteristics to bear in mind when you're creating a sound, whether it's intended to be a realistic sound or some crazy new thing that could never exist in the so-called real world.

As far as we're concerned today, ADSR relates to what's known as the volume envelope. You'll see ADSR, or some elements of it, displayed throughout Live's instruments and effects, sometimes with a graphic representation of the envelope, sometimes not. It's a way to visualise what's happening with a sound, whether it's a kick, a bass or a trumpet – it works across the field.

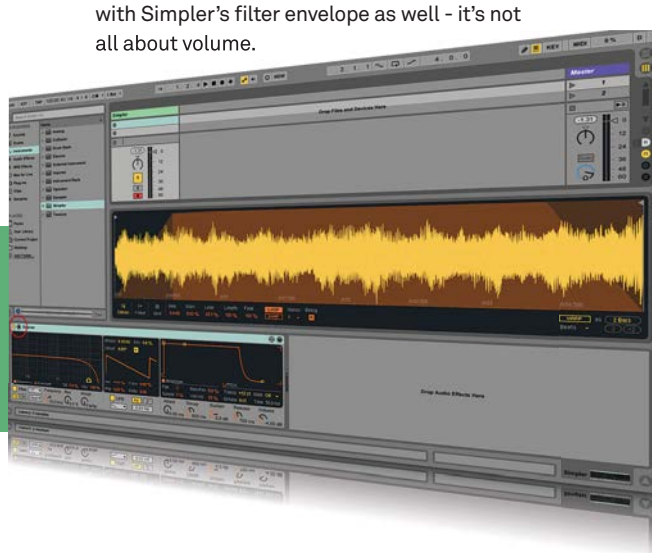
Every sound is like a story, with a beginning, a middle, and an end. Attack is the beginning, Decay and Sustain are the middle, and Release is the end. Yes – I think that makes sense! I'm using Simpler and Operator for this tutorial, but the concept of envelopes is pretty universal, so you'll be able to recycle this information later, whatever instruments or software you're working with. For example, even within Live itself, you'll find full ADSR controls within the Analog synthesiser, or the Max For Live Bass instrument and attack and release controls inside audio effect devices such as Auto Filter and Gate.

I've created an example Live set to go with the tutorial, so make sure you're using that. If you don't have Operator installed, you can at least listen to the audio examples I've provided from each Operator-related tutorial step, so you can hear what I'm describing.

I love Operator, as you'll know if you've read any of my other Live tutorials. Even the sample in the Simpler track was created with Operator. Simpler is one of Live's old reliable instrument devices, but it enjoyed a deep overhaul not long ago, adding a lot of cool features, but also enabling Push users to dig deeper into controlling their samples without having to leave the Push interface – especially true if using Push 2 with the lovely new display.

Our starter sample is pretty raw and doesn't do much except start, play, then loop, but before we progress to programming or editing or effects, we can make it more entertaining by playing around with the ADSR controls.

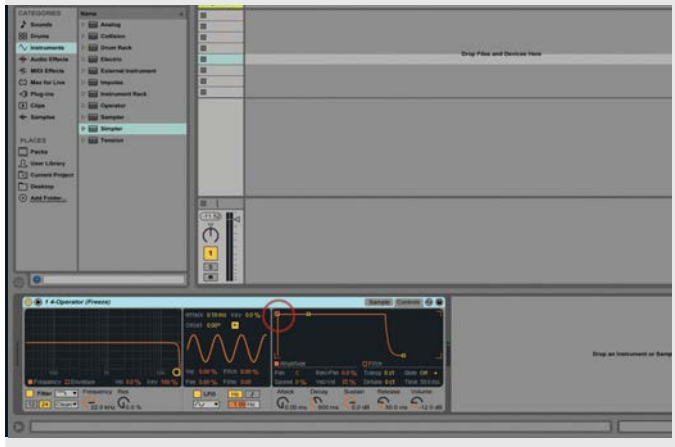
Sometimes you have a goal in mind, a particular sound you're chasing, and sometimes you're just enjoying the journey, happy to go wherever the software and your mood lead. If you want to go further than we do in the tutorial, you can go to that panel at the left and start experimenting with Simpler's filter envelope as well - it's not all about volume.



A SIMPLE START

I start with Live's Simpler instrument device for this tutorial, partly because it's included with every version of Live, while Operator, which we move on to after that, is an additional purchase, so not everybody has it. Over the years, Simpler, despite being the 'poor relation' to the grander Sampler instrument, has matured into a powerful device in its own right; now it's more a matter of appreciating each for its distinct features, rather than holding one to be 'better' than the other. Throw Push 2 into the equation and you have an incredibly powerful sampling setup straight out of the box.

MTF Step-by-Step Back in the ADSR



01 We'll look at how we can use ADSR envelopes to transform a basic sound into something more exciting. Envelopes are everywhere, so you'll be able to recycle these concepts later!



02 We've got an example Live set, which contains a Simpler starter preset, as well as an Operator one, for those who have it installed, and some before and after audio examples for those who don't.



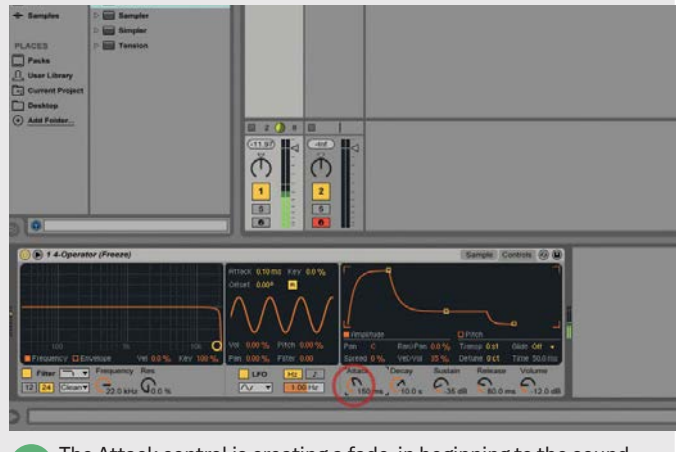
03 We'll begin with Simpler, as that's the one every Live user has. I'm using the current version – it had a major update recently so if yours looks different, it's probably time to think about upgrading.



04 Go to the Simpler 1 track and launch the MIDI clip. Click the Controls tab to view the Volume ADSR envelope (the controls are available at the bottom of either view but let's see it).



05 Set the Attack value to 150ms to take the edge off the beginning, and set Decay to 10.0ms, and Sustain to -35dB, and finally the Release value to 80.0ms to add a longer tail.



06 The Attack control is creating a fade-in beginning to the sound, which is good for more pad-type sounds, while you'd go the other way and use shorter attacks if you wanted something more percussive.

Ableton's Operator synth is a fantastic platform for envelope experiments, it has envelopes for each of the four oscillators, as well as the filter, LFO, and pitch, and these are completely independent of each other. In our tutorial we only use two oscillators, but things can get very deep indeed if you start working with all four of them.

If you're working with a MIDI clip and programmed notes, I'd suggest taking some time to play Operator from your keyboard or Push pads as well, because some of these controls really come to life with playing dynamics. By the time we get to the final step of the walkthrough, with the looping envelope, the LFO, and the filter, it sounds as if there's a lot more happening than is actually the case. Use your imagination – once you add control from Push, clip envelopes, and MIDI and audio effects, you can build very complex patches and instrument racks. None of these →

MTF Step-by-Step Back in the ADSR (cont'd)



07 The Decay control sets the time it takes for the initial attack 'spike' to reach the level at which it'll continue onwards. There's no right or wrong with these, it's about what works with your project.



08 The Sustain control sets the level at which the sound continues, the time having been set by the Decay control before it. It'll continue at this level until the note is released.



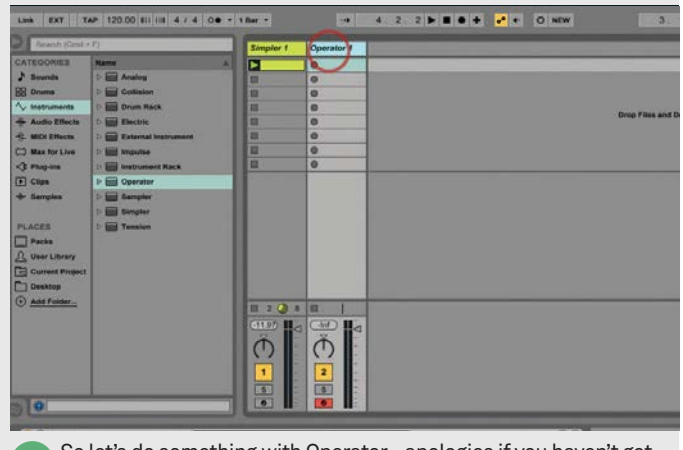
09 Finally, the Release control sets how long it'll take for the note to fade to silence. It's like a story – every sound has a beginning, a middle, and an end!



10 Without doing anything else, we've changed this original sample considerably. Try changing the Vel>Vol value to increase response to velocity changes if you're using a keyboard.



11 Note at the left, that Simpler's Filter also has an envelope with adjustable ADSR values – click the little 'Envelope' button. Related features are also found in some separate audio effect devices.



12 So let's do something with Operator - apologies if you haven't got it, but it's one of the fundamental Ableton instruments, and a good example for us. Go to the Operator 1 track in our set.

→ values should be thought of as being 'fixed' at the start of your song, they can be fluid and continually automated through a mix, or manually tweaked as part of a performance – each element of a volume envelope can be automated separately.

If you use the envelope loops in Operator, as described in the tutorial, you don't use the automation so much, as this is another way of getting a continually changing sound, especially if you apply different loop behaviours and values to each oscillator, so they don't all start and stop at the same time. This is an important Ableton Live tip: avoid moments where everything starts and stops at the same time – it's not very organic! Try using two oscillators, and sending the LFO to just one of them, and looping the

MTF Step-by-Step Back in the ADSR (cont'd)



13 Click the Envelope button at the bottom left of the Envelope display to view the volume envelope for oscillator A. We're using the default sine wave with oscillator A; the others are silent.



14 Set Attack to 130ms, Release to 2.00 , peak to -14. Leave Decay and Sustain alone for this one. You should have a simple sound that fades in gently and out the same way.



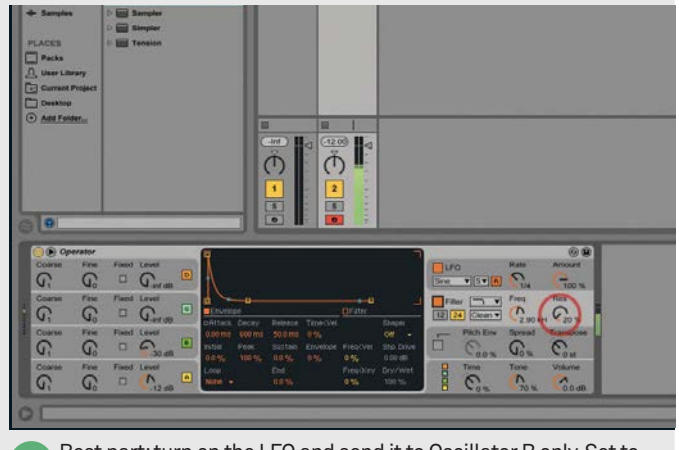
15 Now let's use another volume envelope. Raise the level of oscillator B to -30, and set it to SW32. Bit more buzzy now! Set Attack to 750ms, Decay 500ms, Release 1.00ms.



16 Now when you play, you can hear the two envelopes doing different things at different times. You can really expand on this concept if you start to use all four oscillators.



17 Go to Oscillator A and turn on Loop/Beat in the central panel and set the Repeat value to 1/8. Now you'll hear the envelopes loop rhythmically as you play in notes.



18 Best part: turn on the LFO and send it to Oscillator B only. Set to Sync, rate 1/4, amount 100%. On the Filter, set Cutoff to 2.90kHz, and Res to 20%. Sounds good?

envelopes for both oscillators as well as the LFO, but all set to different rates. Use Operator's Time control to mess globally with all of the envelopes if you really want to mess things up. A hardware controller that easily communicates with Simpler or Operator is a real asset for this, so I'd look at something like Push 2, or if you like to use iOS, try TouchAble, with the Live Device Template in-app purchase, which effectively gives you a multitouch version of Simpler or Operator on your iPad, which is pretty awesome to behold. Better yet, use Push and TouchAble together, which is arguably the best possible control combination for Live.

Within Live's instruments and audio effect devices, you'll find that envelopes are all over the place. You've got them, you might as well learn how to use them. MTF

MTF Feature In The Box Mastering

THE NEW WORLD OF IN-THE-BOX MASTERING

Maximising the impact of your finished mix has changed radically over recent years with the intruction of so many new delivery formats. Here, **Mike Hillier** covers the in-the-box tools and tricks you need to know...

The days of mastering being used to squash the life out of mixes are over – the loudness war has been won, and dynamics are finally coming back.

The process began with the introduction of EBU R128 in 2010, a set of recommendations regarding loudness set down by the European Broadcasting Union. This set of guidelines defined a new way of measuring loudness – the Loudness Unit (LU) – and recommended that broadcast audio be normalised at -23LU referenced to full-scale digital audio (LUFS). This guideline quickly found its way into law in Italy; and broadcasters in many other European countries, including the UK, set personal goals of complying with the standard. A similar law, the CALM Act, was passed in the US, which demanded broadcasters there comply with ATSC A/85, which is based on R128.

These systems, though, only apply to broadcasters – and while mastering engineers began to take note, it wasn't until streaming services such as Apple Music, Spotify and YouTube all began implementing similar 'loudness normalisation' that the war was finally won. The loudness normalisation techniques used by each of these services differ slightly, but the principle is similar in each: the loudness of each track is measured before playback, and each file is automatically turned down so that they play back with the same loudness. Heavily limited tracks are turned down the most, while dynamic tracks are mostly left alone – meaning they now punch louder than the limited tracks.

Of course, this doesn't mean that mastering is no longer necessary. Mastering has always been about far more than simply making everything louder. And artists haven't stopped wanting to hear their music as loud and in-your-face as possible. It's simply that the techniques to achieve loud and aggressive mixes have changed, and the methods we used to use now have the opposite effect, making mixes quieter and less noticeable.

With this in mind, it wouldn't be much of a surprise to see many albums from the loudness wars getting the remaster treatment, but this time with their dynamics left alone, or even enhanced.

In this feature, then, we're going to look at what you can do in your DAW to make your masters sound loud and punchy, without sacrificing the dynamics. We're also going to look closely at the measurements you should be making to ensure you are listening to your tracks as they will play back on each of these streaming services.

THE NEW LOUDNESS PARADIGM

Previously, mastering engineers have used all manner of tools to measure loudness: from PPM and VU meters to RMS measurements, dynamic-range meters, and now the new Loudness Unit. So which should you use in your own mastering sessions?

Before we go on, it's important to understand the difference between each of these types of meter. Peak meters, such as PPMs, measure the loudest points on the waveform, and are useful for informing you of how much headroom you have before the signal will clip, but do not offer any indication of the subjective loudness of the material. A heavily compressed signal may sound loud while not containing any loud peaks, while a less-compressed signal may sound comparatively quiet, while containing some very high peaks. Peak meters can also be calibrated to over-sample the incoming signal in order to determine if the signal contains any inter-sample peaks which may cause problems further along the chain, either when converted to an analogue signal, or when converted to mp3. This type of peak meter is sometimes referred to as a True-Peak meter.

RMS, VU and Loudness Unit meters provide a more useful measure of perceived loudness as they average out the signal over time. This means they are less useful for knowing where clipping may be occurring in your signal, but will let you know how loud one signal will be



perceived compared to another. However, even these are not ideal measurements, as our ears perceive volume differently at different frequencies, and have no means of showing how much of the signal lies within the range of human hearing. Two signals with similar RMS and peak levels may still not sound comparatively similar in level, if one of the signals contains a good portion of its energy outside of the range of human hearing while the other is more heavily band-limited. Even within the limits of human hearing, a 200Hz tone will sound quieter
than a 2kHz tone at the same level, following the equal-loudness contour, or Fletcher-Munson curves.

To counter this, there are various frequency curves which can be applied to weight the signal before measuring. A-weighted curves have been commonly used for measuring loudness and are built into many meters. You will also occasionally see C-weighted measurements, and a new weighting curve, the K-weighting, which is most commonly used alongside Loudness Units to measure loudness in Loudness K-Weighted Full Scale (LKFS).

With all of this in mind, it's clear that there isn't one single measurement you should be using, but a combination of several different meters: one which can show you the true-peak values of your →



Powerful emulations of traditional studio mastering trickery are available to the home producer in the form of dedicated metering, EQ, limiting and compression plug-ins

signal, and another which can show you the perceived loudness. The difference between these two figures is also important, as it is this which gives us an indication of how much dynamic range has been left in the recording.

It is also important to consider the difference between short-term loudness, and long term, or 'integrated' loudness. A song might, for instance, have quiet verses and loud choruses. This will mean we should get a different loudness measurement for each. In fact, even if we look at a single bar of music, it's likely that some beats will have more energy than others. A short-term measurement over 500ms is likely to produce a different result to one over five seconds. However, it is the integrated loudness that's important to us, as this is what will be measured by streaming platforms to determine the playback gain, or how much the track is to be turned down by.

WHAT IS THE NEW LOUD?

Up until now, in the digital era we have only needed to pay close attention to the peak meters to make sure the track wasn't clipping, while VU, RMS and other loudness meters have been used merely as a guide. And while many mastering engineers would argue vociferously about increasing the dynamic range in the final master, the client – be that record label or artist – would nearly always be pushing for louder masters, hoping, perhaps naïvely, that louder masters would make their music stand out. Now that audio-playback software, such as iTunes, and streaming services, such as Spotify and YouTube automatically normalise the loudness of music, adjusting the volume of louder tracks down to make everything the same, we need to start paying closer attention to the loudness and less to the peak level.

Consider the difference between short-term loudness, and long term, or 'integrated' loudness

A NOTE ON PEAKS

While peak meters will provide a sample-accurate measurement of any peaks in your audio signal, when the signal is converted from digital to analogue, inter-sample peaks can still cause distortion problems with some systems. Similarly, these inter-sample peaks can cause problems when converting from a lossless audio format, such as WAV, to a lossy format, such as mp3. Try to use True-Peak limiters and True-Peak meters to avoid any inter-sample clipping, or alternatively, leave an extra dB of headroom available at the end. Peaks at -1dB and a loudness of -11 LUFS should still give you plenty of impact and level.

INTEGRATED LOUDNESS

Without going into the maths, 'integrated loudness' is essentially a measure of the average short-term loudness over the track. However, to prevent long periods of silence from bringing the integrated loudness down, the meters stop integrating the measurements whenever the signal drops below a threshold of -10 LU (relative to the un-gated measurement). This means that you can't trick the system by adding long periods of silence to the end of your tracks to bring down the integrated loudness.

MTF Step-by-Step Mastering with IK Multimedia T-RackS



01 Add your track, or tracks to T-RackS and play through them several times. Pay close attention to the stereo spread, frequency balance and dynamic response of the track. Make notes on anything you feel might need to be altered.



02 Add a Brickwall Limiter in the final slot. Set the Output Ceiling at -1dB and for now, leave the Input at 0dB. We will come back at the end of processing to alter the Input in order to set our final level.



03 Add a Classic EQ in slot 5 and use this to notch out any problem frequencies. We're using slot 5 to avoid dealing with the parallel processing in slots 1 to 4. You can use the A/B/C/D chains to compare different EQs or EQ settings.



04 If the dynamics need reigning in, add a compressor in position 6 in the signal chain. We like to compare a fast, modern-style compressor with a slower vintage-style one as, even with similar settings, each will bring something different to the final sound.



05 After the compressor, we're going to use another EQ for more general voicing. Add a Vintage Program EQ 1A in position 7. Where the Classic EQ was used to notch out problems, this EQ will be used to shape the overall tone of the master.



06 Now go back to the Brickwall Limiter in Slot 8 and set the input gain such that the RMS Meter reads around -14dB. If your peaks are still a long way from the ceiling, consider reducing compression earlier on. If your limiter is working too hard, compress harder earlier.

MTF Step-by-Step Mastering with iZotope Ozone 7



01 Add the track or tracks you want to work on to the Ozone project, and initialise the project by removing or bypassing all of the active modules. Listen to the material closely, paying attention to the stereo spread, frequency balance and dynamic response. Make notes.



02 Don't add loads of modules The fewer you add, the less likely you are to over-process. Start with a simple Maximizer module, with True Peak mode engaged. The Limiter shouldn't be doing anything initially: if it is, turn the input gain down to give yourself headroom.



03 Add an EQ module and drag it into the first position in the chain. Use this to notch out any problem frequencies. The digital mode in this EQ is capable of some very precise EQ notching – very useful for targeting resonances without effecting the rest of the signal.



04 Add any additional processing as you feel it's necessary, whether that's compression, further EQ tweaks, stereo imaging or excitement. Drag each module into position in the chain before the final Maximizer module.



05 You can add third-party VST or AU plug-ins, too, which can be useful for adding additional metering, such as a loudness meters, de-noising tools or a favourite audio processor.



06 Set level with the Maximizer Threshold: add 1dB headroom with the Output Ceiling. You can even hear the difference this makes to future mp3 encoding via the built-in Codec monitoring. As this is a True Peak limiter, you may be able to get away with a higher ceiling.

MASTERED FOR ITUNES

In addition to using a low loudnes-normalisation value of -14LU, Apple has made other steps to try and improve the quality of music available from its store, or streaming services. Apple's AAC codec has been designed to take advantage of higher bit-depth and sample-rate files, and Apple suggest submitting songs as 24-bit/96kHz where available. Additionally, Apple suggests that files should peak at -1dBFS. This little bit of headroom ensures no inter-sample distortion is captured in the AAC file.

More to the point, in order to get our tracks sounding as loud as possible on these platforms, we need to start targeting our loudness to these platforms while maximising the peak-to-loudness ratio (PLR).

While it would make our lives simpler were each of the major platforms (Apple Music, Spotify and YouTube) to use the Loudness Unit to measure the program loudness, and all to converge on a single normalisation value, the reality is that each of the platforms has its own proprietary loudness measurement and each has a slightly different figure for normalising to.

Apple Music, for example, uses SoundCheck to determine the program loudness, normalising each track to approximately –16 LUFS, while YouTube is normalising to around -13 LUFS and Spotify is currently normalising to -11 LUFS, which, while 5LU louder than Apple, is still dynamic compared to many 'loudness war' casualties.

In effect, this means that on Apple Music, any track mastered louder than -16 LUFS will be turned down automatically to playback at -16LUFS, while on Spotify, any track mastered louder than -11 LUFS will be turned down.

On Apple Music and YouTube, more dynamic tracks with a peak-to-loudness ratio greater than the platform target loudness allows will not be adjusted.



Q&A WITH PETE HEWITT-DUTTON



PETER HEWITT-DUTTON is a mastering engineer at Metropolis Studios, London. He has followed the traditional rise, from receptionist, through assistant and finally to mastering engineer in his own right. Peter is now a specialist in cutting vinyl at the studio. *MusicTech* caught up with him to discuss the challenges of working with vinyl, and the more exotic methods of cutting that the studio employs.

Q: What effect has the rise in vinyl sales had on you at Metropolis?
Our cutting lathes are busier than ever! We almost can't cut lacquers quick enough to keep up with demand. It's been fantastic. While a lot of studios (sometimes literally) threw out their lathes in the 90s when CD was killing vinyl sales, ours were carefully maintained, waiting for the time when they would be busy again.

Q: Why is vinyl becoming popular again?
I think the main reason is that in a world of people listening mainly to downloads, the LP seems a better physical product to own than the CD. The ritual of putting the disc on the turntable, the artwork and lyric sheets is all being introduced to a whole new generation now.

It should not be forgotten that the numbers of records being sold are still comparatively small, but that is growing steadily, and showing no signs of slowing. The biggest restriction is becoming the lack of time available at pressing plants.

Q: Are more pressing plants opening?
It's not that easy… the equipment has not been manufactured for over 30 years now, and keeping the old gear running is getting harder by the year. Some companies are starting to manufacture new presses now, so hopefully, this situation will get better in the near future.

Q: Has the process of cutting a record changed much over the years?
Very little has, in essence; it is a very simple process of cutting a groove into a lacquer-coated aluminium disc which has hardly changed since the 1930s. Over the years, there has been refinement of the process – computers designed in the 70s and 80s designed to fit more audio on a disc, and at a higher fidelity for example, but in essence, what we are doing has not changed.

Q: What is half-speed mastering?
It's one of the best ways we have to make a record sound better. The groove on the record is a direct representation of the acoustic sound, so the more detail you can get in that groove, the more audio detail will be reproduced. High frequencies, for example, can be very difficult to cut, as the stylus cutting the groove has to move very quickly, slowing the whole process means it has more time to cut the groove, and therefore can put more detail in. We play back the audio at half the original speed, and spin the turntable at 16 rpm, so when you play it back at 33 rpm, everything is back to normal speed, but sounding better than if we'd cut it at 33 rpm.

Q: Do half-speed cuts improve all kinds of music?
Yes, every area of the audio sounds better when cut at half speed. So for dance music, you will get a tighter, punchier bass; pop benefits from the stability of the centre image and clarity of the vocal, and the open, clear top end sounds great on classical recordings.

Q: Are there any downsides to working at half-speed?
Only for the engineer doing the cut. Several of the processes done when cutting at full speed don't work the same at half speed, because they are frequency specific. For example, sibilance has to be very carefully controlled on vinyl, but that processing has to be done beforehand for a half speed cut, as the top-end frequencies which need taming are now in midrange.

MTF Step-by-Step Mastering with plug-ins



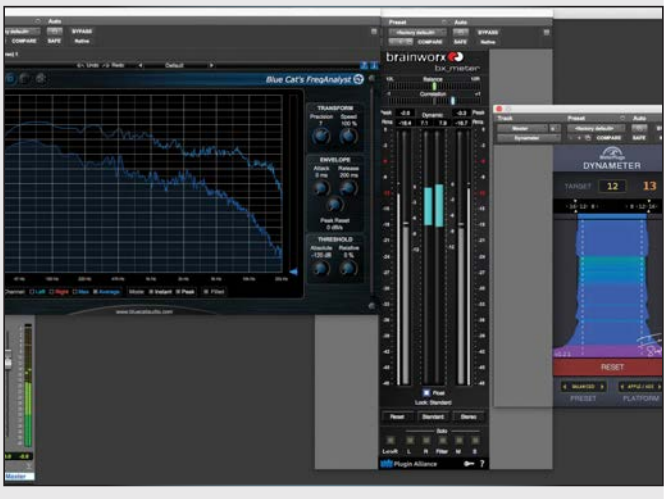
01 Using plug-ins inside a DAW gives you the most flexibility to choose which processors to use at which point. But, as before, first listen to the material and make notes, paying attention to the stereo spread, frequency balance and dynamic response of the track.



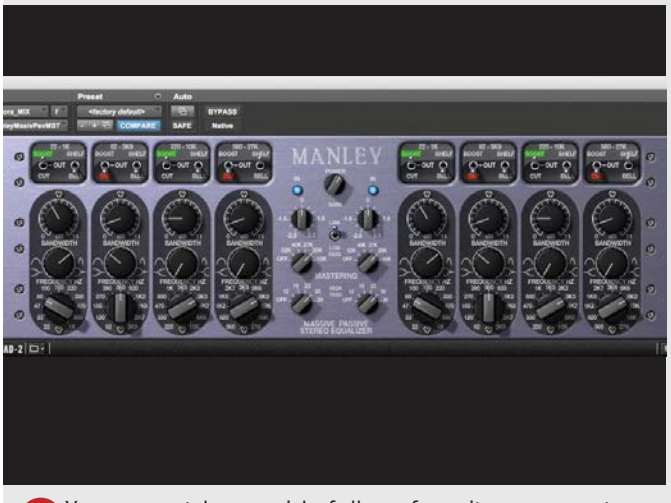
02 We always start by adding the final processor first, so let's add a brickwall limiter in the last plug-in slot on the channel. We're using the UAD Precision Limiter which is not True Peak, so we're setting the output to -1dB.



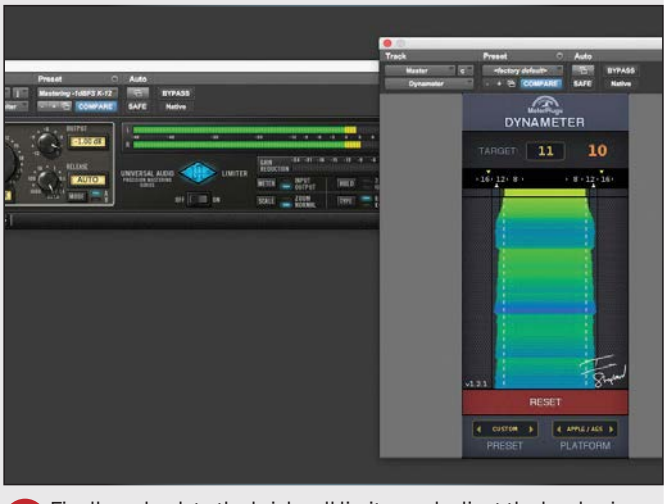
03 In the first slot, we're going to add the Brainworx bx_digital V3 equaliser, and we're going to immediately follow that with a FabFilter Pro-Q 2 equaliser. We'll use the bx_digital as an M/S EQ, and the Pro-Q 2 as a standard stereo EQ.



04 We have a variety of meters we can place on the Master channel. Here, we're adding MeterPlugs' Dynameter, Brainworx bx_meter and the Blue Cat FreqAnalyst. Alongside the DAW meters, this offers loudness measurements and a frequency spectrum.



05 You can now take your pick of all your favourite processors to add. Only use what you need; don't automaticaly add 'favourites'. Here, we've added a further shaping EQ (UAD Manley Massive Passive MST) and we're not going to use any compression or saturation stages.



06 Finally, go back to the brickwall limiter and adjust the level using either the input or threshold, depending on the plug-in you've opted for. You want to try and bring the loudness up to around -11LU with peaks ideally at 0dBFS.

So they will sound quieter, maintaining the integrity of the dynamics in the recording.

However, Spotify will apply peak limiting to tracks with a peak-to-loudness ratio greater than 11, which in effect turns everything up, but at the expense of dynamics.

HOW TO MAKE YOUR MUSIC STAND OUT

Since Spotify is using limiting, and has the highest platform loudness, it makes sense right now to use Spotify as our 'target loudness', in order to avoid having our masters further limited by Spotify – unless you intend to master separately for each platform. Masters will continue to be turned down on other platforms, but the dynamics will be left alone, and our masters will still sound more dynamic on these platforms and have more energy than loudness war tracks, which will have been turned down a far greater amount.

If you don't have a loudness meter with measurements in Loudness Units, a simple RMS meter will provide an adequate approximation of short-term loudness; but measuring integrated loudness is more difficult without a dedicated meter. If the track has roughly equal amounts of loud and quiet sections, you could try and balance the two out, so that the average remains around -11dB RMS, with verses measuring around -13dB RMS and choruses up to around -9dB RMS. But if you are thinking of becoming or already are a professional mastering engineer, you'll want to invest in one of the available meters.

When working on an album, don't think you have to make every song the same loudness. Many platforms are intelligent enough to know when the listener is listening to a full album, and will measure an integrated loudness for the whole album, and play back at that level when the album is played as a whole – only using the track integrated loudness when playing back playlists, or random play. This enables us as mastering engineers to continue to craft a journey through the album, with loud tracks having more impact after a quieter ballad.

If we can't use limiters to make masters sound louder, what tools and tricks should we be using to give our masters impact? The simplest trick is to make use of the extra headroom available. If your average loudness is -11 LUFS (the Spotify loudness target), you have 11 LU of headroom for your peaks. If you can use this full range your mix will instantly carry more dynamic punch than mixes with quieter peaks. In many cases, this will mean finding new ways to add to rather than reduce the dynamic range of masters. Expanders are one option; however, they often don't produce great results on full-programme material, being better suited to specific tasks.

EQ is the best weapon in the mastering engineer's arsenal for giving the mix impact, particularly since the weighted curves of the loudness meters will enable us to boost some frequencies without much change in the measured loudness; while cutting others will help us reduce the measured loudness while keeping the peaks where they are, in effect increasing our dynamic impact.

Since the weighting filters almost all work by focusing the measurement onto the most receptive part of human hearing – between roughly 1 to 5kHz – we can focus our attention closely on this range. Any cuts made to this range will provide us with a more heavily reduced loudness compared to cuts in other ranges, while boosts outside of this frequency range will not increase the measured loudness by as much. There is little point pulling everything out in the 1 to 5kHz region of course, as your master will sound terrible – the goal should still be to produce well-balanced masters – but if there is room to pull some out, you can now do so knowing that your mix will play back louder as a consequence. Also, now that we have headroom available to us, we can safely add a little extra sub into our masters without lifting the loudness too much.

Listen closely, several times, before you consider reaching for a single processing tool

Of course, we're not suggesting that you can't use limiters at all. They are still very useful tools for protecting your final master from clipping, particularly True-Peak limiters. And they still have a useful purpose for deliberately squashing material for creative musical purposes, irrespective of the loudness normalisation.

THE HIPPOCRATIC OATH

Mastering compression, like limiting, will reduce the dynamic range of the mix. And despite using lower ratios it can often have a more dramatic effect on the audio, even with only one or two dB of gain reduction. While it does reduce dynamic range (thus increasing the measured loudness), subtle use of compression can still be a very important tool in mastering, precisely because it can have a large

RMS

DAWs and meters occasionally use different methods for calibrating their RMS measurements, using square waves rather than the standard sine wave. This results in an RMS reading that is approximately 3dB lower than a sine-wave-calibrated RMS reading. As the AES stipulates using the sine-wave reading to calibrate RMS levels and the LU meters use a similar calibration, you should aim for a value of -14dB RMS on square-wave-calibrated meters, or risk mastering your tracks too loud.

COMPRESS OR LIMIT?

In the fight for louder and louder masters, several stages of compression and limiting were often applied to try and squeeze every last drop of level out of a master. In the new loudness-normalised paradigm, this will only result in flat, dull masters which sound lifeless next to more dynamic, uncompressed tracks. But both compression and limiting still have a place. Compression tends to be used at lower thresholds, and so has a bigger impact on the overall tonality and cohesion of a mix (a little can go a long way here), while limiting is almost transparent when used only on peaks. However, it should be used sparingly.

Loudness Meters

New loudness measurements require new measuring tools. While many engineers will be happy approximating for loudness with **RMS meters**, mastering professionals will want dedicated **LU metering tools**, which can show both the short-term and integrated loudness measurements of tracks they are working on...

IZOTOPE INSIGHT

Price **$499**

Web **www.izotope.com**



Insight is one of the most comprehensive metering tools available. In addition to a range of loudness meters, showing short term and integrated loudness over time and a loudness graph, Insight has a suite of tools for metering and visualising, including spectrum analysers, peak meters and an interesting tool for comparing channels in a mix called 'meter tap'. Insight is included with the Advanced versions of iZotope RX and iZotope Ozone.

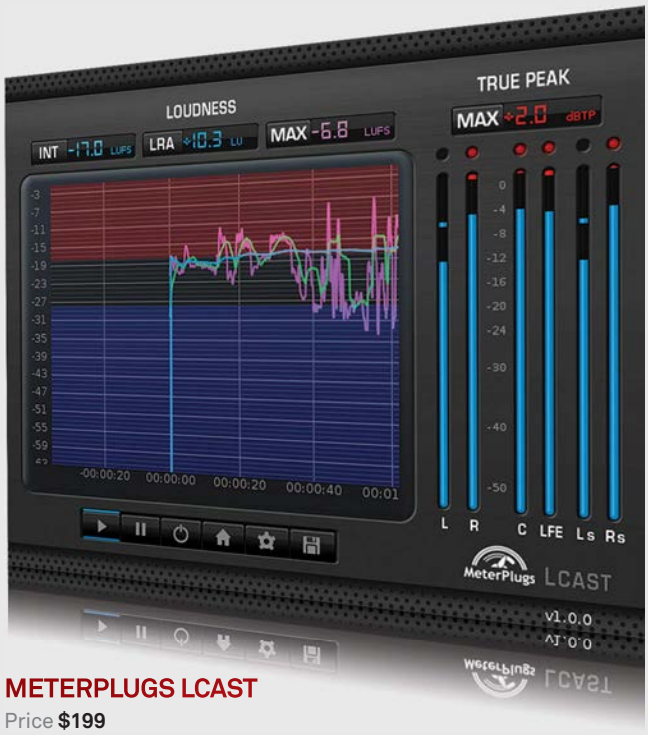


KLANGFREUND LUFS METER

Price **$49**

Web **www.klangfreund.com**

Klangfreund LUFS Meter might look quite basic, but it is capable of all the necessary measurements including short term and integrated-loudness measurements, true-peak measurements and can even be used to automatically adjust the loudness to a predetermined level, which is particularly useful for anyone working in broadcast. A non-commercial, feature-limited version of Klangfruend LUFS Meter is also available for $29.



METERPLUGS LCAST

Price **$199**

Web **www.meterplugs.com**

In addition to short-term, integrated and true-peak levels, LCAST lets you target your audio to the loudness-normalisation levels of Spotify and iTunes. This meter also supports offline bounce, enabling you to get faster-than-real-time results for your integrated loudness and loudness history. A Surround version of LCAST is available, which can export the loudness measurements to a .CSV file, enabling you to generate loudness reports.

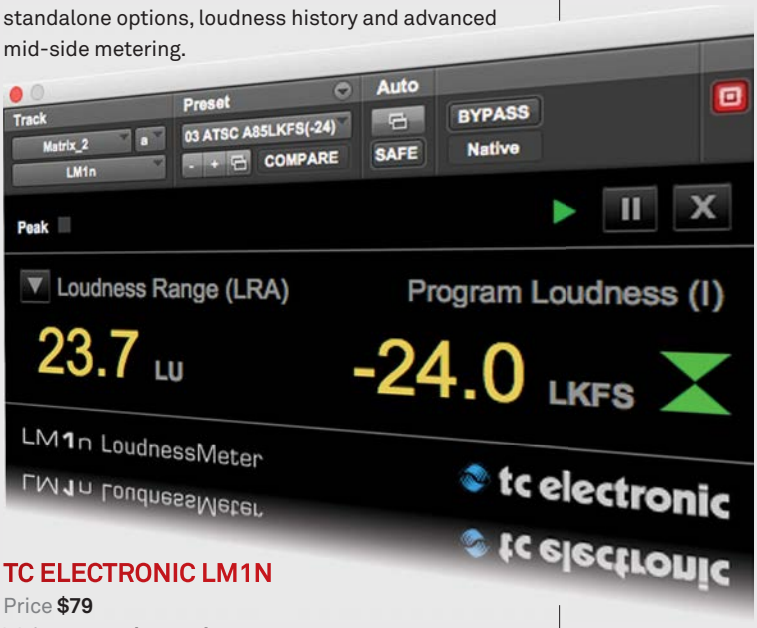


NUGEN VISLM-H2

Price **£342**

Web **www.nugenaudio.com**

VisLM-H2 has true peak, momentary, short term and integrated-loudness meters, with support for up to 7.1-channel Surround Sound. Furthermore, VisLM-H2 is capable of showing mid-side loudness separately from stereo or surround loudness; loudness history can be viewed and exported for offline reporting. A compact version VisLM-C2 is also available, which lacks the standalone options, loudness history and advanced mid-side metering.



TC ELECTRONIC LM1N

Price **$79**

Web **www.tcelectronic.com**

LM1n is the simplest of TC Electronics' range of loudness-measuring tools, providing a quick measure of the integrated loudness, and loudness range. Pro Tools users can quickly measure the loudness of any clips by using the offline tool. And simple up and down arrows provide instant visual feedback if the material needs to be made louder or quieter.



WAVES WLM PLUS

Price **$399**

Web **www.waves.com**

Waves WLM Plus provides momentary, short term and integrated loudness measurements, as well as true-peak readings, all of which can be exported to a .CSV file for offline reporting. WLM Plus also includes a true-peak limiter, which can be used to ensure that there are no inter-sample peaks in your audio which could cause distortion further down the chain.

→ effect on our perception of the mix even at subtle settings. Compression can glue a mix together like no other tool at our disposal. It can also pull all of the life out of a mix – so be careful. Not every mix needs mastering compression, but if a mix isn't sitting together, and EQ alone isn't helping you to place it all in one space, then adding a slow-attack, slow-release compression stage with a low ratio can help to sit everything together.

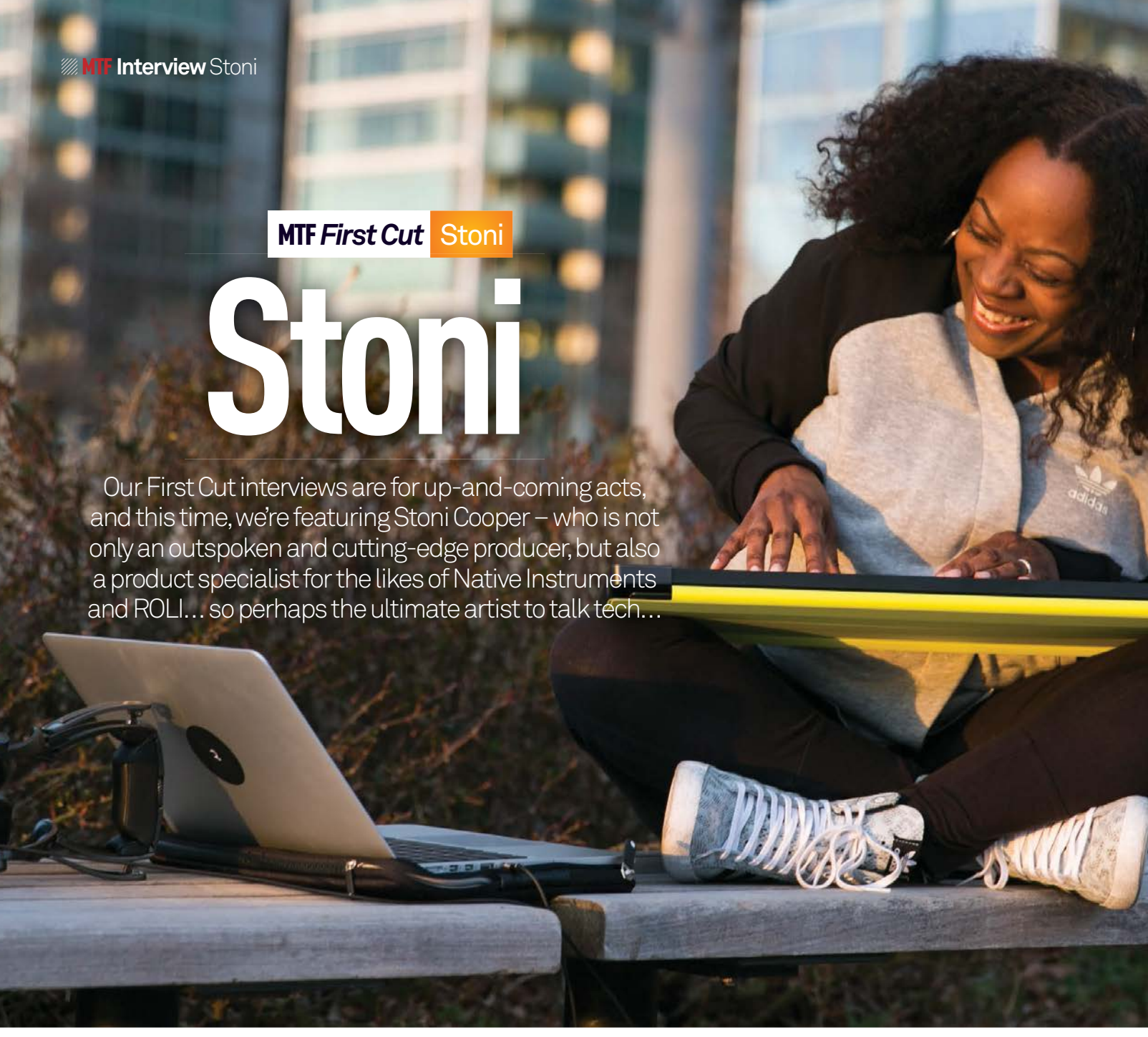
Saturation tools can also help to sit a mix together, especially one that has been done entirely in the box. However, this too should be applied sparingly, and often not at all. If a mix is sounding overly digital, spiky, and harsh, a little soft-clipping through a tape simulator can be the ideal tool, but many mix engineers are now employing analogue emulation tools, such as tape simulators and mix-buss emulations at nearly every stage, adding noise and saturation throughout the mix. In this case, a final stage added at mastering is only going to raise the noise floor and add mud to the mix. As with everything else in mastering, these tools should only be used where appropriate.

In an ideal situation, a great mix will need no mastering processing added at all, and the mastering engineer will simply listen, and then pass on the mix untouched. Even among experienced professionals this is likely to be rare, but should give you an insight into how to approach a master. Listen closely, preferably several times, before reaching for a single processing tool. Describe for yourself what needs to be altered in the mix to make it sound better, and then – only when you are sure you can make it better – should you apply the process. Most times, we find ourselves needing a little gain adjustment first, followed by some EQ – usually to pull out a resonant frequency in the mix. With this done, listen again, and compare the master closely with other tracks in the genre, before deciding if any gentle shaping needs to be done with further EQ processing, or if more drastic attention is needed with compression, M/S tools, multi-band processing or saturation.

There are, of course, still some listeners who listen to physical formats, as well as a couple of streaming services which don't yet use any loudness normalisation. So some artists will still want their music to sound as loud as possible on CD. Sometimes this can't be avoided, but it's worth considering that a more dynamic mix will sound better (if not as loud) on these formats. Plus, many listeners who still use CD are doing so on high-quality systems, and will have plenty of volume available on the amp should they wish to simply turn the CD up. MTF

STEREO WIDTH

One of our favourite mastering tricks is stereo-image manipulation. Advanced frequency-dependent tools exist for shrinking the low end and widening the top end; but often, just a simple full-band expansion of around 10 per cent can be all that is needed to give a track a little extra dimension. It is easy to overuse, however, and a skilful mastering engineer will know when a mix should be shrunk rather than expanded in the stereo field, and will even occasionally automate the stereo width through a song to bring extra attention to choruses.



MTF *First Cut* Stoni

Stoni

Our First Cut interviews are for up-and-coming acts, and this time, we're featuring Stoni Cooper – who is not only an outspoken and cutting-edge producer, but also a product specialist for the likes of Native Instruments and ROLI… so perhaps the ultimate artist to talk tech…

We love featuring artists who are using technology in innovative or perhaps extreme ways. This time, we have someone who has not only been inspired by a rich and diverse background of music, but one who's better versed than most in music technology, having been a product specialist for several high-tech companies including Native Instruments and (currently) ROLI, Stoni Cooper.

MusicTech: So who is Stoni?

Stoni: An outspoken, powerhouse music producer who's unafraid of making waves and unapologetic for changing the musical landscape. Born in Brooklyn, NY, I grew up listening to everything from Sam Cooke and Stevie Wonder to Prince and Michael Jackson, but my influences don't begin and end there. I cut my teeth alongside some of the legends of hip-hop and rock music, and was mentored by engineering legends.

I gained a vast knowledge of the inner workings of the music and recording industry and discovered a love for sound design. It was my passion for sound manipulation that led me into production. I'm very hands-on and I work very hard. I always have. My work ethic is unlike anyone I know and I believe that this is what determines your success. I love what I do, and for me, that is the ultimate success.

MT: Tell us about your sound or philosophy?

Stoni: I've been told that my trademark was the hard crack and thump of my drums, but it's really the emotion that my music evokes. My approach to production comes from the heart, meaning whatever I'm feeling at the moment that I'm creating a body of work. If I'm having a bad day, or feeling generally frustrated about a particular situation, I'll use the aggressive energy from that and pour it into the music and that's what is reflected. If I'm feeling super relaxed or very Zen-like, you'll get a track that will feel much smoother, more melodic and less syncopated.

When I produce other people, I allow the artist to be themself at all times. The goal is to get the most authentic performance from them – anything



less than that just isn't believable. You, the listener, should feel everything that person is sharing with you. If you can't feel all of that, then the music is disingenuous.

MT: Tell us how you put a track together?
Stoni: There's isn't one process that defines my sound. I'm learning new things and adding various processes to my workflow. I'm a consummate student of the technology. Whenever anything new comes out, I learn about it, study it and if it works with my existing workflow, then I'll incorporate it. If it doesn't, then at least I still know more about the hardware and software than the average producer would and I can share that with others.

I will share one trick that I love to do. I have this process where I take my voice and make a sine wave out of it. I then create a melody with that same vocal sample. I then add an acoustic sound (guitar or strings) to give it that organic quality. I continue to layer a VST instrument from either Massive, Kontakt or whatever VST fits what I'm feeling, in that space. The end result is always a crowd pleaser.

MT: Describe the Stoni sound…
Stoni: Eclectic. The beat is hip-hop based, but the music is really different. It's world music. It's global. It comes from the kind of music I listen to everyday, which could be French music, hip-hop, electronic, jazz, music from any and everywhere. My sound at its core is very beats-driven – aggressive and edgy – but it does have many soft elements to it and lyrical ones as well. The music has to flow well for it to make sense and elicit a certain feeling or emotional response. Musicality is very important to me, so I make sure that the beats aren't the only thing you're hearing. My music makes you feel good. It's the kind of music people just want to dance to.

MT: How do you produce your music?
Stoni: When I approach making music, the process usually starts with how I'm feeling. I like to sketch my ideas while sitting on the floor, legs crossed or on a bean bag in a relaxed sitting position with the Rise or my Maschine Mk2 in my lap. Then I will comb through different sound ideas. It could be acapellas to trigger inspiration, or it can be a synth, maybe even a simple hi-hat pattern that might send me all the way into my zone. I have various methods that I use to cut a track together, from building patterns and layering them into a song, or I open up a DAW and have it in a linear form and create melodies and drum patterns across a timeline.

The role of the producer is to be a visionary and complete a body of work (a song or an album) from A to Z. I'm the one who sets the tone in the studio session. A producer is supposed to give the artist, beat-makers and musicians direction. I select which musicians I want to use for a particular song, and then lead them through the creative process. "Sing the riff more soulfully, don't play that line so staccato, make the groove feel lazier," etc. I'm like the leader of the band, so I have to be clear and precise in my instructions and know where I want the music to go so we can all arrive together in the same place, otherwise the song and the sound will fall apart. I'm also an excellent listener, because it's imperative to hear the artists' or musicians' intention and then work together to bring it to fruition. I am the person who orchestrates what the sound will be on the project. Meaning, I determine what sounds are used and what musicians will embellish the track best.

> "The **music has to flow well** for it to **make sense** and elicit a certain **feeling** or **emotional response**"

I'm also the person who sits with artist to discuss their lyrical content, assist the songwriter with the flow of the song, and contribute to vocal production and arrangement. I'm responsible for the format of the album, meaning the order of songs based on content and context. I create the beats and the melodies. →



MT: What gear do you use?
Stoni: The main components in my studio are the MacBook Pro, NI Maschine Studio, Maschine Mk2, Komplete Kontrol Keyboard, Ableton Push 2, Moog Sub 37, Seaboard Rise 25, 49 and Komplete Audio 6. The history behind me using the Maschine is very simple. For starters, I loved the amazing browser that it has. There's so much you can do with this unit. I realised that portability has a big influence on what I use at home, as well. Having the Komplete sound collection, coupled with various synths and plug-ins, allows me to write and make records anywhere and at anytime. The Seaboard Rise allows me a level of creativity that I've never been able to achieve with the standard keyboard. The 5D touch allows me to manipulate sound in a way that shapes the direction of my creative vision.

My go-to tools are Maschine, the Moog, Ableton Live Instruments, Komplete Ultimate 10 Bundle, Arturia, Reason Refills, Serum, and U-he. I have a lot of fun using the Push 2 and it keeps my creativity flowing. I love the way you can play chords and melodies with the pads. Using the Moog Sub 37 creating new patches is awesome. I never know what the end result will be, and lately, I've been diving into using the iPad apps with my Ableton setup. Man, that Link option is so powerful, it turned my iPad into another instrument. That's the best part of using these tools on a day-to-day basis. The creativity and fun never stops.

MT: What's on your wishlist, gear-wise?
Stoni: A UA Apollo Twin and all the UA plug-ins, PMC Passive MB2S, Neumann U87, API Lunchbox, and an aluminium Moog Voyager.

MT: What would you like to see developed on the music-tech side of things?
Stoni: I'd like to see a more cohesive system with all the DAWs, across the board. I would like to see more hybrid platforms for controllers, so that producers and DJs can work with or without the laptop.

MT: What advice do you have for working in the music industry?
Stoni: The best advice that I've picked up in the music industry is: Be as self-contained as much as possible; master your craft and learn all of the ins and outs of whatever it is you choose to do; build solid relationships, because ultimately, those relationships will help you grow; be selective about the company you keep and make sure the producers you are around are as productive as you are.

MT: And, similarly, what advice you have from working in the studio?
Stoni: Studio etiquette is everything. Your behaviour in a studio will either allow you to grow or cause you to not be invited back. Become the consummate student. There are several aspects of the music industry to learn about. Working consistently in the studio will give you the opportunity to meet A&Rs, engineers, artists, songwriters and ultimately, label executives. Watching what these people do in the studio, listening to the things they talk about and the things that are important to them, will give you a plethora of information and arm you for the future. Do more listening and less talking. It's great to share your ideas with like-minded individuals because it gives you a sense of validation, but the best thing you can do sometimes is just watch and listen. Some of the most invaluable lessons I've ever learned came just by listening.

> "Do **more listening and less talking.** Sometimes the best thing you can do **is watch and listen**"

MT: Tell us about your new EP?
Stoni: It has been long-awaited, not only by me, but by my loyal followers. I had a great time putting my experiences into musical form. There's a tremendous satisfaction you feel when you have a vision for yourself and you make it manifest. This EP will reflect the different styles of production I've brought together by using the most unlikely pairings, which make for the most amazing music. I want the people I've met and who've been connected to me through music to have something tangible to walk away with and listen to over and over again.

MT: What do you have planned for the future?
Stoni: I plan on touring and playing the songs on the album on the road. I'm creating a performance set for the people who enjoy the music and I'll get the chance to meet my fans face-to-face. I also plan on signing a few producers and giving them the opportunity to work on projects with me and build a team of creatives. And the next project I release will feature a few new collaborators and artists who I believe are really going to make a big impact on music culture and the industry. Stay tuned. MTF

LOOPMASTERS.COM

GREAT SOUNDS. SERIOUS INSPIRATION.

SAVE 10%
Quote code **MT10**

The number one website dedicated to providing 100% royalty-free sample libraries for producers worldwide from the best sample labels on the planet.

FUNK | DRUM&BASS | HOUSE | DUBSTEP
TECHNO | TRANCE | VOCALS | BREAKS | RETRO
WORLD | URBAN | CINEMATIC





















THE SAMPLE BOUTIQUE

loopmasters.com

Six of the best

Hardware ▨ Software ▨ Mobile Tech ▨ Accessories

Welcome to the second ***MusicTech Focus* Buyer's Guide**, where we round up some of the best products recently reviewed in *MusicTech*. This time we look at some of the **headphones** we've tested for all ears and studio uses…



Details
Price £239
Contact Synthax +44(0)1727 821870
Web www.synthax.co.uk

BEST In-ear

Audio Technica ATH-E70



Details
Price £280
Contact Audio Technica 0113 277 1441
Web eu.audio-technica.com/en

We like pretty much all the models in the ATH IEM series of in-ear 'phones. The cheapest E-40s are particularly good for just £70 but if you want the best in-ear experience, you'll have to go for the top-end E-70s. Yes they're four times the price but, well, here's what reviewer Andy Jones said: "The 70s are the best all round, as you might expect for the money, and the best in-ear 'phones I've mixed on, translating all I'd done on much more expensive headphones and £1,300 monitors. They will take some getting used to, but if you want your music making streamlined and free of excess weight and size distractions, these are as good as it gets. ***It's the old 'you gets what you pays for' message, as these sound the best of the AT IEM range."***

BEST Mixing

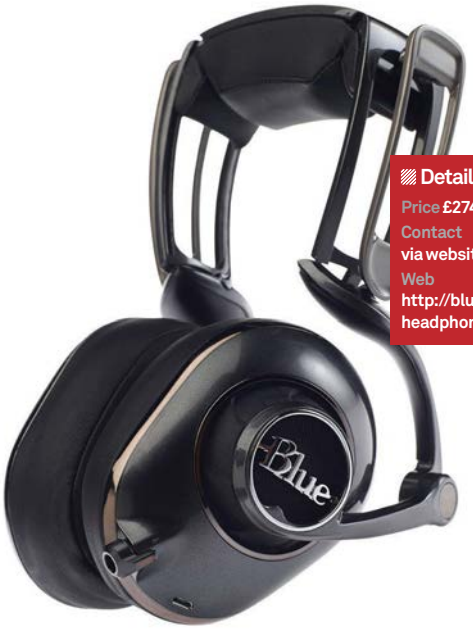
Ultrasone P880

The Holy Grail of headphones is to get comfort, accuracy and something that doesn't fatigue over long sessions, and these might just fit all bills. Andy Jones said: "The 880s are a great fit and feel snug and assured. The sound is flat and accurate, and surprisingly not too fatiguing which – like flat monitors – can happen with such responses over long sessions. So flat, comfortable *and* suitable for long sessions? That is a rare and great combination to achieve. ***Good price, great sound, accurate and comfortable. Win, win, win and win!"***

BEST Looking

Blue Mo-Fi

Okay, we admit that we were drawn to these more through their looks than anything else when we saw them on display at last year's NAMM show. Luckily, they not only sound as good as they look but are also easily as comfortable. We said: "They fit incredibly well around your head. The headband extends from a compact, sprung storage position outwards, and if this wasn't enough, both 'phones extend downwards, too. This is the clever bit; rather than notching them down, you gently pull the 'phones away from you and they drop – very cool! Sound-wise, these are right up there with our reference phones and they translated a mix we were working on very well indeed – all of the detail we expected was there. ***They look and sound great – the amp and unique design set them apart from the rest."***



Details
Price £274
Contact via website
Web http://blue-headphones.com

BEST Budget(ish)

Fostex TR Series



Details
Price £155
Contact SCV Distribution
Web www.scvdistribution.co.uk

These might not be budget for some people, weighing in, as they do, at £155, but as with studio monitors, we recommend spending as much as you can on your cans, so this is cheap believe us! There are actually three models in the TR range: open back, closed back and semi-open and these will very much suit whatever you are used to. We find that open back, for example, might offer a more immersive experience but do need to be driven harder and are not as good at insulating the outside world. Of the range, reviewer Andy Price said: "Overall comfort scores highly and the unique slider bars are easily scalable to get the perfect fit. For the all-important sound quality we tried out a variety genres, finding that the Fostex TR Series offered a magnificent and dynamic listening experience, balanced between the weighty low end, detailed mids and sharp trebles." He concluded: ***"Our new go-to 'phone choice for music listening."***

BEST Accuracy



Details
Price £165
Contact Pioneer 0208 836 3500
Web www.pioneerproaudio.com

Pioneer HRM-7

With a frequency response of 5 to 40kHz you can expect these to deliver accuracy... and they do. They are also comfortable and offer good isolation, but that accuracy can fatigue over longer periods. We said: "As with monitors, you want them to be as accurate as possible and these 'phones are good for critical listening. Use with caution and they can open up parts of your mix that others don't reach. ***You really can hear that response. A fine addition to your studio."***

"Money should be **no object with monitoring** – but try explaining **£1,300 for a pair of 'phones to your wife.** If you can, though, **your mixes will benefit...**"

BEST Ultimate!

AKG K812



Details
Price £1,099 +VAT
Contact Sound Technology on 01462 480000
Web www.soundtech.co.uk

At the best part of £1,300 (retail, you can get them for under a grand street) these are the most expensive 'phones we have tested but they do certainly sound it. You can literally hear everything you need to within an airy environment that exudes quality. We said: "These are the best-sounding 'phones we've used, bar none. Price will be an issue. Even though we've said that money should be no object when monitoring, try explaining £1,300 for some headphones to your wife. If you can, however, your mixes and masters won't regret it. ***Accurate, vibey, spacious, comfortable and perfect for mixing, these are the ultimate 'phones (at a price!)"***

MTF Feature

TOP 50 FREEWARE!



It's time for our annual *MusicTech Focus* freeware roundup! We've gathered the best free software titles in a variety of music-production categories and presented them here. Want to make music for nothing? Look no further…

Last year, we predicted that this would become an annual event here at *MusicTech*, and we were right! So successful was 2015/6's Freeware 50 that we've decided to do it again for 2016/7 – that is, list the best 50 freeware plug-ins in the world, right now!

The freeware scene has been around since the birth of software. It highlights either free versions of paid-for apps (these might act as an advert for a more fully featured version), or complete fully functioning software from a small company which programs for fun. There are a huge number of developers – from big names like Native Instruments right down to one-person companies. We admit that we have some favourites, but this year we've kept a tight rein on things, with firm rules for our list of the best free software out there. These are:

1. It's an-all new list, apart from half a dozen that just had to be included again and, yes, Crystal is one of them.

2. We've included both real and synth instruments; effects (creative, mixing and mastering); plus other studio production options, including utilities.

3. We've also chosen all (bar one!) titles that are available on Mac or PC.

So, sit back and enjoy *MusicTech*'s freeware of the year… and don't spend a penny on any of it. →




Ticky Clav
VB1 Virtual Bass Unit
Sustain Bass
steinberg
Stereo Touch
Narrow Space
Compare
Copy
Paste
View: Editor
Side Channel Delays
DELAY 2
D2 PH INV
HPF ENABLE
LPF ENABLE
Delay 1
D1 Gain
Delay 2
D2 Gain
HP Filter
LP Filter
Out
Out Gain
Cymbalistic
SubBass1 - O2
One Stage
Compare
Editor
Phazor
STAGES
FREQ
FB
SPREAD
RATE
DEPTH
LFO

→ **FREEWARE** Synths

We kick off with freeware synths and a whole host of newbies, plus a comeback king and a couple that will never go away...



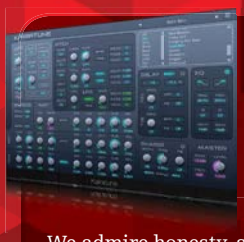
TOGU AUDIO LINE NOISE MAKER
Wavetable synth

We included TAL's Elek7ro last year and its BassLine is a firm SH-101 freeware favourite. Add the fact that this is but one of three TAL inclusions this year, and you could conclude that we like the company a lot. NoiseM4k3r (as it's stylised) is a new and improved version of Elek7ro and includes a couple of oscillators plus sub, LFOs, ADSR and four oversampled filters. The effects really give it a nice pro edge with reverb, delay and bitcrusher on offer among others. But 256 presets also explore the full sonic palette of a virtual analogue synth, with leads, pads, arps and basses aplenty. An essential freebie.
W: tal-software.com/products/tal-noisemaker
Platform: Mac, PC



NUSOFTING SINNAH
Complex waveshape synth

They say: 'Sinnah the sinner, is it a sin to be free, or is it a sin to sound so lush?' We'd say 'no' on both scores, but Sinnah certainly stands out in the freeware-synth world. It only has one oscillator, but delivers a wide sound with complex waveshapes, a noise element and a rather bold and fully featured Delay Matrix, which takes up almost half the space on the GUI. It results in some complex presets and some fairly mediocre ones on first listen. But there's plenty to get these beefed up – and a fantastic, clean GUI to do this – and the bank of 50 free presets has got to be worth downloading, too.
W: nusofting.liqihsynth.com/Sinnah_VA_synth.html
Platform: Mac, PC



FUTUCRAFT KAIRATUNE
Electronic synth

We admire honesty, so we appreciate Futucraft's claim that Kairatune is not an all-in-one synth, but one aimed at electronic-music composers. And for that, it does a great job, delivering a tight, punchy sound that will sit well in stripped-back dance mixes. We really like the preset-management system that allows you to get in there by category and then select your presets with absolutely no fuss whatsoever. There's an almost bewildering number of controls on offer, but happily, the values are shown by great dial-type indicators. The Delay section, including phaser, delay and EQ components, is particularly welcome.
W: futucraft.com/category/kairatune/release
Platform: Mac, PC



DATSOUNDS OBXD
Classic synth emulation

There's no getting away from what OBXd is emulating: a classic Oberheim synth, as is everyone and his dog at the moment (including, thankfully, Tom Oberheim!). OB-Xd does it very well, and claims to take some of the features further than the original. There are a dozen banks of presets to explore including bass, drums, pads, leads, keys and brass and, in the main, it offers a great stab at all of them (even though we all know that brass emulations should really be made illegal in this day and age). The front panel is functional and large, with a Voice Variation section particularly handy. A very solid and useful synth.
W: obxd.wordpress.com
Platform: Mac, PC



U-HE TRIPLE CHEESE
Comb-filter synth

Like Frohmage, this is another cheese-based freeware plug-in that's been around for a while, but now deserves a bit of a comeback. We suppose you could say they have 'matured well' (sorry). Triple Cheese is a comb-filter based synth that first appeared a decade ago. Since then, it's been updated, had presets added and should run in whatever DAW you have. We had it up and running in Logic (although needed to reinstall it) and were soon enjoying its rather more unusual presets. There are lots to get your teeth into across more than a dozen banks: lots of percussive and out-there types, plus some great delayed sounds. Superb.
W: u-he.com/cms/triple-cheese
Platform: Mac, PC



GREEN OAK CRYSTAL
Semi-modular synth

Yes, we included Crystal last year, and yes, we included Crystal the year before and yes, we will probably include it next year. If you're new to freeware, we need to tell you: Crystal is where it's at and probably always will be. It's been around since 2002, and some say it will never stop working. Okay, that's not quite true, but we downloaded version 2.5.4 (released last April) and it slotted in straight away within our sleek and modern DAW setup. It looks old but sounds amazing, with a huge variety of presets from a semi-modular synth with both subtractive and FM synthesis. There are also plenty of control and modulation options, too.
W: greenoak.com
Platform: Mac, PC



MATT TYTEL HELM
Original synth

Despite stating it's free, Helm is one of those downloads that makes you feel guilty for not donating at least a dollar, and maybe you should because while you can download this for free, it's quite an object to behold. The GUI is great and very modern with some neat graphical representations of the various parameters and the sound is anything but boring with swirls, drones and even a gameshow theme to tempt you. It might be a little too out there for some but has masses of potential.
W: tytel.org/helm
Platform: Mac, PC



ARCHETYPE INSTRUMENTS LOCAMOTIV Synth

Another out-and-out dance-type synth, Locamotiv might not be that pretty – it's pretty dark and just black and white – but has a fat sound. It has some fantastic basses and leads, and is ultra clean and perfect for today's electronic music producer. However, the developer's site was down as we wrote this – which could be a shame – but the plug-in is downloadable at other sites.
W: archetype-instruments.com (not presently live)
Platform: Mac/PC



SYNTH1 Virtual Nord Lead

Synth1 is another freeware synth classic that we have to include every year, or people will march on the MT offices with virtual pitchforks. It's apparently modelled on a Nord Lead 2 (sonically, not visually). We managed to get it working in the latest Logic pretty much straight away, although issues are creeping in, like odd mouse movements. Worth persevering with for the sound, but possibly its last inclusion (sound of protest march starts – oh dear).
W: www.geocities.jp/daichi1969/softsynth
Platform: Mac, PC



DIGITAL SUBURBAN DEXED
DX-type FM synth

We've had Dexed in our plug-in folder for some time, but for some reason, have never included it as a recommendation. This could be down to an interface that attempts to expose FM synthesis in all its glory – something that not many have been brave enough to try. Yes, it might put you off wrestling with it, but actually all the controls are there to make some pretty dramatic real-time changes, something that FM is not known for but also something that both Yamaha and Korg have attempted in hardware. So dive in, enjoy the 32 percussive, chimey and dreamy presets and give FM a go. It's never been easier. Okay, we mean it's never been cheaper.
W: le-son666.com/asb2m10
Platform: Mac, PC



BRAIN CONTROL TUNEFISH 4
Virtual Analogue synth

We're quite chuffed that a) we've managed to get all Mac and PC freebies on to this list of synths and b) that they're not all virtual analogue. Not that there's anything particularly wrong with VA, but there are rather a lot of them about.

Tunefish 4 is, thankfully, an example of a good one that not only has a good analogue signal flow, but offers enough extras – the kind of thing you should get with software. So you get stacking effects, loads of modulation routing possibilities and loads of hands on and clean-looking controls to get to the heart of programming quickly. A fine looking and sounding synth.
W: tunefish-synth.com/?action=about
Platform: Mac, PC



NUSOFTING DA HORNET
Virtual Wasp synth

Slight caveat with this one. We included Da Hornet as a fun version of the well regarded UK Wasp synth – itself becoming legendary, as it was one of the first truly affordable synths in the country and had odd touch-sensitive keys. The software version is available as a Mac and PC freeware synth, but only as a 32-bit version (so you'll need to use bridging software for 64-bit installation). There is now a new 64-bit version – another reason it's back in our list –but you are urged to donate at least $9 to keep support going for the plug-in which we'd recommend, because it is a great, if slightly noisy – but that is the point – synth.
W: nusofting.liqihsynth.com/freeplugins.html
Platform: Mac, PC

FREEWARE Drums and drum machines

Free beats with our round-up of six of the best…



MELDA MDRUMMER SMALL
Virtual drummer

Melda Production has loads of freeware and MDrummer Small is, not surprisingly, a free version of the company's MDrummer Large. Nevertheless, it's quite a substantial product and download itself. It comes with 500MB of sounds and data, which includes 30+ drumsets, 400+ drumset components, 200+ samples, 50+ multisamples, 60+ rhythms, 200+ base-rhythms and 2,500+ loops. There's an onboard arranger/sequencer and effects, plus loads of hands-on controls for creating varied machine or human-like beats.

W: meldaproduction.com/MDrummer%20Small
Platform: Mac, PC



MONADE SOUNDS MINI909
TR-909 emulator

We like this because it's simple. Yes, like other plug-ins – particularly drum-based ones – it is trying to sell something else (in this case, the full Monade Sounds Pure909 which has 19 kits). However, Mini909 is easy to get and use and you do get a usable 909 kit, one of our (and many other people's) favourite kits and the backbone to many a dance track. The sounds are there and it certainly does its selling job well, as after listening, we were intrigued enough by the full version's processed 909 kits, which only cost marginally more than a tenner to get. If you're poor this will fill a sizeable 909 gap.

W: www.monadesounds.com/2016/06/pure909-plugin.html
Platform: Mac, PC



VINTAGE DRUM ELEMENTS
Yamaha RX5 emulator

You can download this in return for your email address (and a voluntary donation) and what you get is a drum-pad player loaded with RX5 kits plus instruments, including DX bass, ethnic and synth. Usefully, you get Standard, Vinyl, Sub and Tremolo effects to add which can really vary the sound.

As to the core sounds, we weren't convinced that this was the classic drum kit we needed until the developers SampleScience reminded us that Cocteau Twins, Bomb The Bass, Masters At Work, OMD, Vangelis, Depeche Mode, and The Pet Shop Boys all made rather good use of the original. So an absolute bargain, after all…

W: samplescience.ca/2015/08/vintage-drum-elements.html
Platform: Mac, PC



BEATSKILLZ BEATFACTORY DRUMS
Drum- synth module

It's fair to say that a lot of the free drum plug-ins here could well be player shells that you pay extra to get sound packs for. This one might well follow along those lines, but does come with 10 kits (808, Trap, 80s, Dubstep, Indian and Sample kits, among others) to start with. Each pad can be edited and an envelope shape applied to the list: plus there's a useful reverb effect. One of the big selling points of the plug-in – yes, we know you don't actually buy it – is that it was recorded using an MPC through some pretty high-end outboard gear and the results do sound good: snappy and contemporary and ideal for a slew of polished genres, including hip-hop, trap and EDM.

W: beatskillz.com/drums
Platform: Mac, PC



DISTOCORE BAZZ MURDA
Kick-drum synth

One that could have gone in the synth section, as the results are more than percussion! Kick synths are all the rage, not only in software (with lots of paid-for apps), but hardware, too, like Korg's Volca Kick. Bazz Murda is a cut-down piece of kick-synth freeware, but there's still a lot left in terms of control.

This is a full-featured synth dedicated to hard kicks for the harder side of dance, and it doesn't disappoint. Those after something with less bite might be shocked at the attitude that this plug-in is capable of reaching – pop producers look away now, as this is strictly for dubstep, gabber, drum 'n' bass. For kicks with added extras, and plenty of them.

W: distocore.net/bazz_murda.html
Platform: Mac, PC



DOPEKITZ RUPTURE
16-part drum machine

Finally – something light on your processor and easy to use, a 16-part drum machine and sample player. DopeKitz's Rupture comes with five kits – West Coast, Tribeca, Ever 1da, Crunk'd and Danger Drumz – so enough to get your teeth into. Yes, there are more expansion packs available for your money, but these will show off the flexibility of Rupture. With sliders, pads and Attack rotaries, there's enough here to give your beats an edge.

You can create and mix kits, too, and pitch the whole lot up and down. A useful plug-in that is both simple and effective.

W: dopekitz.com/rupture-virtual-drum-machine.html
Platform: Mac, PC

FREEWARE Effects

Freeware effects that you can use creatively for sound mangling, or for mixing your music …



TOGU AUDIO LINE REVERB-4
Modulated vintage reverb

There's no doubting that freeware often offers you a simple version of a piece of paid-for software, and TAL's Reverb-4 is just that. You were never going to get a convolution reverb with a gazillion room and modelling parameters to adjust now, were you? Instead, you get a high-quality and simple-to-use plug-in, a version of which appears in TAL's own Sampler. There are Wet, Dry, Delay and Size parameters to adjust – with sizeable ranges for each for dramatic effect – but maybe most useful are the Hi- and Lo-cut dials, with a nice dose of variation across the frequencies. It's simple but sonically very useful.

W: tal-software.com/products/tal-reverb-4
Platform: Mac, PC



AUDIO DAMAGE FUZZPLUS 3
Vintage distortion pedal effect

FuzzPlus 3 is "back by popular demand" and a "ground-up rewrite of our popular FuzzPlus series", according to Audio Damage. And whereas its aim is vintage – with claims that it accurately models vintage distortion-pedal circuitry, as well as the filter characteristics from a Korg MS-20 synth – it's a very modern-looking plug-in with neat meters and dials for Frequency, Feedback, Resonance, Distortion and Level. In practice, it offers a hell of a bite and kick, really livening up whatever you throw at it, especially beats. With many presets, you'll be bringing this one into your mix more than most. Dirty or clean, it certainly has an attitude you'll want.

W: audiodamage.com
Platform: Mac, PC



DDMF COLOUREQ
Five-band parametric EQ

With five bands of EQ and Frequency, Gain, Q and Mix dials per band, ColourEQ is already just a point-and-drag away from being as fully featured as many a paid for plug-in. Okay, the front panel might not be the prettiest, but the GUI is colour adjustable, as is the window size. The sound clicked on occasion in Logic, but otherwise swept along nicely, with a useful Gain slider really ramping things up when you need it. Not perhaps as accurate as some – but nor would you expect it to be – ColourEQ is a versatile plug-in that is more creative than you might at first suspect. Lots of potential for sonic tweaking, for zero cash outlay.

W: ddmf.eu/freeware
Platform: Mac, PC



IZOTOPE VINYL
Dusty old vinyl-record simulator

With vinyl back in fashion again – to the point that we've launched a sister magazine about it – this 15-year-old plug-in is now freely available, in return for your registration details at iZotope's website. You too can add noise, dust, hiss and wear to your recordings, so if you're fed up with the ultra clean and pristine, then this is for you. Yes, we can't believe we wrote that, either. Vinyl does seem to go against everything you stood for – as does the resurgence of the format, in some ways (ooh, being a bit controversial there) – but actually, one listen to it and you will find a place in your hearts and mixes for this plug-in.

W: izotope.com/en/products/create-and-design/vinyl.html
Platform: Mac, PC



BLUE CAT AUDIO CHORUS 4
Vintage chorus module

This is another 'vintage' effect, but there's more to it than just creating vocal chorus effects. The front-panel controls allow you to get in and change the original sound beyond recognition. You get the usual Gain, Rate, Depth, Dry and Wet controls, but the Delay dial brings in the most drama, and a useful Stereo Spread dial offers even more presence. It's simple, yes, but as transparent or as in-depth and colourful as you want it to be, thanks to a varied range of onboard presets. Blue Cat Audio does a freeware bundle which we'd highly recommend (with at least two of the plug-ins from it listed here).

W: bluecataudio.com/Products/Product_Chorus
Platform: Mac, PC



VLADG MOLOT
Colourful compressor

We should admit that we were taken in by the Russian military looks of Molot before checking the sound. We'll say it again, we've been overwhelmed how good freeware GUIs have got over the last year and this is one of the best. The look simply screams that it's a compressor that will add colour and character – indeed, the developer says that if you want transparency, you've come to the wrong place! We also love the straight-to-the-point instructions like 'Too noisy? Turn off the dithering. Too complex? This plug-in is not for beginners'. It might not be – but a bombastic compressor it is…

W: vladgsound.wordpress.com/plugins/molot
Platform: Mac, PC

FREEWARE Effects (cont'd)

More free effects for creative mixing and fabulous, free sound design…



CABLEGUYS PANCAKE
Dynamic panner

We were initially unsure as to whether to slot this one into our Utilities list, as it looked like 'just' a stereo spreader at first, but Cableguys PanCake is a lot more than that. Sure, you can pan your audio hard left or right with it, but you'd be missing out on a whole load of functionality. The main draw is the ability to pan according to modulations sync'd with your audio, so put a beat loop through it and it can, for example, pan left and right in time with beats, creating a more musical movement than you might otherwise achieve. Obviously, you can go OTT with this kind of thing, so be careful. What are we, your mothers, now? Having to warn you to be careful with effects? Sheesh.
W: cableguys.com/pan-cake.html
Platform: Mac, PC



TRITIK KRUSH
Modulating bitcrusher

Out of all the effects on offer in this year's roundup, this one perhaps offers the most instant, smile-on-your face fun when booting it up for the first time, not to mention having yet another clean and modern GUI (we've been really impressed with the modernity of a lot of the interfaces on the freeware we've looked at this time around, we have to say). Krush is just that – a bitcrusher with Drive and Krush dials plus Frequency and Resonance filter controls. There's a sync'able Modulation section that is well worth exploring, and which offers some fine rhythmic additions, plus plenty of presets to get your teeth into. If you're going to try any of this year's roundup, give this a go – we dare you try it and not smile.
W: tritik.com/products/krush
Platform: Mac, PC



OHM FORCE FROHMAGE
Resonant filter

Ohmforce was one of the first high-profile plug-in companies, boasting odd-sounding, cheesy (in name) effects and oddities. Frohmage was one of our first experiences with a piece of freeware – yes, it's been around for a while – and we've shied away from its inclusion in the past, simply because we used it so much we thought everyone might be sick of it! But such are musical fashions, we reckon Frohmage is due a revival – and after loading it back into our rack, we're right back on it – as it's one of those effects you can use (and easily over-use) on everything. Now it has a new list of possible applications. Welcome back, Mr Cheese.
W: ohmforce.com/ViewProduct.do?p=Frohmage
Platform: Mac, PC



SMART ELECTRONIX AMBIENCE
Reverb

If ever you'd choose a plug-in on its name, it's probably this one. Who doesn't want a bit of ambience now and again? You certainly wouldn't choose it on looks – even the knob view (as opposed to the slider shown) is not great, but it does it sound great. We still have a tendency to overdo reverbs – it's an 'age' thing, believe us – and it's VERY easy to do with Ambience. You can quickly be inside the biggest cathedral and then put that inside a planet for OTT dreaminess. Yes, it's easy to get a bit lost in dreamland, but why not, once in a while? Ambience genuinely does sound better – on many occasions – than a paid-for reverb, so don't judge a plug-in by its cover, please don't…
W: magnus.smartelectronix.com
Platform: Mac, PC



TAL-FILTER-2
Host-synced filter

It'd be easy for us to fill these pages with TOGU Audio Line (or TAL) products, as they have so many freebies up for grabs and they are all good. We've been restrained this time around and only included three(!) and this is a sync'able filter that's got oodles of potential. Sync'ing, of course, means lots of rhythmic fun, a little like Frohmage above, but this is perhaps a little easier to control. There are four TAL NoiseMaker filters (LP 24dB, LP 18dB, LP 12dB, LP 6dB, HP 12dB, BP 12dB, N12dB for you filter fans), and modulation, saturation, distortion – and any other word with 'tion' at the end – are all easy to conjure up. And while you're downloading it, get the other TAL plug-ins while you're there, eh?
W: tal-software.com/products/tal-filter
Platform: Mac, PC



ADAM SZABO PHAZOR
Access Virus-type phaser

There are specific effects for very specific things, and this is certainly one of them. The Access Virus synth has certainly picked up a lot of fans over the years (the editor of this magazine has spent the last 10 trying to get one) and Adam Szabo has coded – with the help of J. Bradford Hunter – a simple phaser plug-in to emulate a characteristic of the synth that earned it much of its reputation. The Virus is a virtual analogue that has been used across many dance genres, including a lot of trance, and this will give you that hard-edged phasing and movement that a lot of the sounds in that genre use. It's simple and not that flexible, but if you want specifics, then this is it.
W: adamszabo.com/phazor
Platform: Mac, PC

FREEWARE Effects (cont'd)



IGNITE AMPS PTEQ-X
Three-part EQ emulator

Take one look at PTEq-X and you'll recognise the EQ type that it is attempting to emulate – it certainly has an air of authority and a touch of the Pultecs (not an illness). The plug-in actually features three types of EQ and four valve types, so offers your productions more flexibility than you might initially think.

There are plenty of options and controls around the outside of the main GUI, which offer more than the original models but at the same time, don't detract or distract you away from the great GUI – and all-in-all, it's a mighty fine emulation.

W: igniteamps.com/en/audio-plug-ins
Platform: Mac, PC



STAGECRAFT SOFTWARE
Bitcrusher and filter

How many times do we have to say it? 2016 is the year of great-looking freeware. SC Software should certainly be applauded for making this interface so vibrant and very 'now'. There's a Stereo Spread and Noise main window that pulses with your music and large LFO and Filter windows to allow you to tweak and twist with ease (the filter applied to the noise and the LFO to the filter). Bit rates and rate reduction processes are as easy to do, as they are dramatic in impact. With plenty of presets and more being added, this plug-in sounds as good as it looks.

W: stagecraftsoftware.com/products/bitcrusher/
Platform: Mac, PC



ACON DIGITAL MULTIPLY
Chorus effect

Digital Multiply is yet another good-looking freebie with simple-to-use controls and one that's nicely sectioned, so that you instantly get your head around the GUI. Multiply processes each voice with what Acon Digital describes as "a phase-randomising filter, so that unpleasant comb filter effects are avoided".

The result is that you get a more realistic emulation of several players, rather than the often more fake multiplier and pitch-shifting you might end up with in other plug-ins. The results are very good indeed, as is the onboard EQ.

W: acondigital.com/products/multiply
Platform: Mac, PC

FREEWARE Utilities

Add-ons and mastering applications that, once used, you'll wonder how you managed without …



PSP VINTAGEMETER
Analogue-style VU meter

The message this year with freeware, is quite definitely vintage. Play your vintage synth plug-in through your vintage effects, and now monitor the output on your vintage meter. While a free set of meters might not be too exciting, a) these are by *MT* favourites PSP and b) they're a tad more useful than you might imagine. The metering features in your DAW may not be the best or the most accurate, but Vintage Meter can have its VU level set from -20 to +3 or a wider -40 to +6 (or click a PPM pseudo-peak option), plus you can adjust sensitivity of the meters.

W: pspaudioware.com/free_psp_plug-ins
Platform: Mac, PC



VOXENGO STEREO TOUCH
Mono-to-stereo processor

A simple mono-to-stereo plug-in would probably not have made this list, but Voxengo's uses the classic mid-side technique to create a more realistic stereo image using mid/side coding. So whereas you might think just by doubling and panning will get you stereo, this uses a technique used in mastering and precise EQ-ing to get you the stereo results. And the results are pretty darn good. We used it on traditionally mono tracks to see how it would cope, and it was neither too retrained nor too OTT. The presets help you get started and should be used as a platform to explore. Sounds far better than it looks.

W: voxengo.com/product/stereotouch
Platform: Mac, PC



LVC AUDIO LIMITED-Z
Easy-to-use limiter

We've included limiters in our utilities list as they're arguably less creative than in-your-face mix and sound design effects. This is one of the best out there, certainly in terms of ease of use, and is based on LVC's fully fledged Limited-MAX mastering limiter. You get metering, look-ahead brick-wall limiting, and several onboard algorithms to provide transparent limiting. Again, it's the interface that grabbed us, with its almost liquid action and dynamic algorithm graphics, you'll soon learn the ropes of limiting – it's very easy to get your head around.

W: lvcaudio.com/plugins/limited-z/#Download
Platform: Mac, PC

FREEWARE Utilities (cont'd)



AUDIO VITAMINS CONTRA FREE
Workflow

If you have a stack of similar plug-ins that perform a same-ish but subtly different task, then Contra could be the plug-in that helps you stop the faffing that goes with the territory of multi-plug-in auditioning. You're probably used to setting up multiple plug-ins on a track and muting like a demon. Contra allows presets to be created that represent your favourite outboard and how they might be good for a particular sound. Working on a bass sound? Load up your favourite compressors, set levels and easily audition them on the fly. Simple!
W: audiovitamins.com/contra-free
Platform: Mac, PC



BLUE CAT AUDIO FREQ ANALYST 2
Spectrum analyser

While these are adverts for bigger paid-for plugs, they still offer a tremendous range of features for the buck, or lack of it. This spectrum analyser lets you get to the heart of your sound by allowing you to monitor its spectral content in real time and in high definition, thanks to its 'unique smooth interpolation algorithms'. You get various modes, including left and right channels, maximum and average. Controls include Transform and Envelope and you can adjust GUI transparency. Indeed, you wonder what the paid-for version can offer over it.
W: bluecataudio.com/Products/Product_FreqAnalyst
Platform: Mac, PC



TB PRO AUDIO DPMETER II
Multichannel meter

Metering is very important, and if your setup is multi-channel, then this freebie offers you a great way to keep tabs on various signals. The meter includes RMS, EBU R 128 and TruePeak measurements and includes manual loudness matching. Surround and multi-channel processing is tailored for with two to six RMS channels, 2.0 stereo, 4.0, 4.1 and 5.1 surround. The GUI is good – not the best, but a cut above many – and there's a hell of a lot crammed in there including several Modes, six RMS and Peak readings and seven Loudness parameters. Lots to get your teeth into.
W: tb-software.com/TBProAudio/dpmeter2.html
Platform: Mac, PC

FREEWARE Real instruments

The best free plug-ins for emulating real or acoustic instruments…



UVI WORKSTATION
Multitimbral instrument host

There are a couple of players that we simply have to include in every freeware roundup's 'real' section. Essentially, they are adverts for paid-for collections, but they're by big companies who tend to a) be good at making plug-ins and b) be generous with their demo sounds. So UVI's is first. The company makes great sound libraries hosted in this free player, but you also get a few free demo instruments like an electric piano, strings, drums, guitar and bass. With built-in effects and the option to run in standalone mode, there's enough here to warrant the download.
W: uvi.net
Platform: Mac, PC



KONTAKT PLAYER FREE
Multitimbral VI host

Another of the big gun's free players that we simply have to include in this roundup is Native Instruments' Kontakt Player. This is probably used more than any other piece of music software, as it hosts not only NI's wares but those of many other companies.

The free Player version will run Kontakt Factory Selection, which is a mighty 650MB free download and contains 13 instruments from NI's Band, six from World, 12 from Synth, 13 from Vintage and six lots of Urban Beats. That adds up to an impressive 50 instruments. You really can't afford not to have them… **W: native-instruments.com/en/products/komplete**
Platform: Mac, PC



BIGTICK TICKY CLAV
Hohner Clavinet emulator

One of the few single instruments we've brought over from last year, if only to highlight Big Tick's range of freeware plug-ins, which is now available if you register at its site and which includes this and Cheeze Machine, which is a great string instrument – 32-bit only, but well worth the download. Ticky Clav on the other hand, emulates a Hohner Clavinet, using a physical modelling synth engine that emulates the string vibration and the pickups specifically from the Clavinet Model C. The 'click' is the key click that was such a big part of the characterful sound of the original. There's also a built-in wah effect and selectable manual and auto-wah.
W: www.bigtickaudio.com
Platform: Mac, PC

FREEWARE Real instruments (cont'd)



VERSILIAN UPRIGHT PIANO Piano plug-in

Most of our 'real' instrument plug-in developers have lots of freebies available, so while we're focusing on a particular plug-in from said developer, we're also introducing their ranges, too. And so it is with Versilian. The upright piano is a great little freebie – an emulation of the kind of upright piano you might have at home – and very good it is, too. You may also be interested in the free harp or indeed a complete orchestra plug-in that features 14 instruments.

W: vis.versilstudios.net/upright-1.html
Platform: Mac, PC



4FRONT R-PIANO Electric-piano emulator

4Front has a range of simple plug-ins, some so simple that they have no parameters to adjust – the standard electric piano, for example, just plays one sound. The Rhodes-based R-piano, however, has a lot to adjust and is all the better for it. You get Dynamic and Drive sliders to enable you to get to the heart of the Rhodes sound, but there are also controls for Q, Release, Decay, Cutoff and Ambience, and an onboard reverb effect with its own pop-up parameters to adjust. There's a lot of variation to be had from one solitary piano type, along with preset locations to save your results.

W: yohng.com/software/rpiano.html
Platform: Mac, PC



ALAN VISTA CYMBALISTIC Cymbal percussion

This could have been included in the drums section, but it does actually lack drums, concentrating as it does on cymbals, and lots of them. They – a hi-hat, ride, four crashes and two Chinese cymbals – are sampled and mapped across eight channels and four stereo outputs. There are useful volume, pan and release controls per channel and an overall Dynamic dial. If it sounds simple, it is, which is part of its charm (and indeed a lot of Alan's other VSTs – see the website for more). It is a very specific plug-in and at 190MB, quite a large download – but it does encourage you to play and experiment with a little top-end percussion, an oft-undervalued way of lifting a song.

W: alanvista.com/cymbalistic
Platform: Mac, PC

FREEWARE Guitar

The guitar gets the last freebie laugh, with five amp- and instrument-based plug-ins…



AMPLESOUND AGM LITE II Guitar emulator

It's very much the 'lite' version of AmpleSound's Ample Guitar M, which is a Martin Guitar emulator. The full version is a mighty 6GB instrument, but this version lacks over 5GB of that content, some of the articulations and is 16-bit as opposed to 24-bit. However, it does give you a good flavour of the Martin and is a very playable plug-in, one of the best ways to see proper guitar emulation in action without paying for it. There are tab and effects options and a keyboard for playing it (we'll assume if you can play a guitar, you'll opt for the real thing, anyway). While it is free, we think you'll be sorely tempted to upgrade, which will set you back $169.

W: amplesound.net/en/pro-pd.asp?id=7
Platform: Mac, PC



LEPOU AMP SIM SUITE Guitar-amp simulators

This is a suite of five plug-ins – yes, we said this was a Top 50, so now it's a Top 55 – which are all amp simulators. There's no fuss, no bother, just download and boot up five great little plug-ins that you can feed your guitar audio through. Better still, experiment by feeding your synths through them for even wilder results…

W: lepouplugins.blogspot.co.uk
Platform: Mac, PC



IGNITE AMPS SUITE Stompboxes to amp sims

We've already included Ignite's EQ in our effects listing, but the developer's list of guitar effects is impressive enough to include more. We've highlighted the TS-999 overdrive stompbox but The Anvil is equally as impressive, as is the Emissary amp.

W: igniteamps.com
Platform: Mac, PC



NI GUITAR RIG PLAYER FREE Guitar processing suite

This version is expandable with the free Guitar Rig Factory selection. You get an amp and cab, distortion, modulation, reverb, delay, EQ and more, all for free!

W: native-instruments.com
Platform: Mac, PC



VB-1 Virtual bass instrument

VB-1 is one of the older plug-ins that Steinberg once sold and is still available for download. It emulates a proper bass guitar and makes a very good stab at it.

W: steinberg.net/en/support/unsupported_products/vst_classics_vol_1.html
Platform: Mac, PC


MTF Interview Apothek

APOTHEK

Pair up a self-confessed studio perfectionist with a traditional songwriter and the results could be a long time in coming. Fortunately, Apothek's debut album is finally out and it was worth the wait. *MusicTech* catches up with the nerd half to test him on his studio knowledge…

Apothek is a Norwegian duo comprising a self-confessed studio nerd, Nils Martin Larsen, and songwriter Morten Myklebust. The band has already attracted the attention of the likes of *The Guardian* to be touted as last year's big thing. But with such a meticulous attention to studio detail, the tracks that caused a stir at the beginning of that year – *Family* and *Waiting For The Thunder* – could well have been forgotten with a wait for a follow-up album.

As Nils says: "I'm really used to sitting for 12 hours in the studio making just one tiny detail of a song, whereas Morten is from a more traditional background, where the track is the track and the song is the most important thing all of the time." Luckily, the resulting album, the self-titled *Apothek*, is here and is as perfect as it was long in inception. *MusicTech* catches up with Nils to talk technology and whether one sound is worth a whole day…

MusicTech: **So how did Apothek come about?**
NML: We are Nils and Morten and we're based in Oslo. We came into the music scene from two widely different directions. Morten comes from a songwriter background and has his own career doing more acoustic stuff. I come from a more studio-focused and compositional background and am more of a studio nerd than he is. I've run a studio in Oslo with a friend for the past six years, so we both came into the music scene quite gradually and naturally from two different backgrounds.

THE APOTHEK PROCESS

MT: **You say you pay a lot of attention to production – where did that ethos come from?**
NML: I think the key to it all is hard work and having a plan to focus on. 'No shortcuts' is probably our main motto. When it comes to songwriting and production, we really want to feel like we've explored every angle before agreeing the track is finished. And setting a few tripwires for each other along the way seems to help us from going down the same route each time.

MT: **How does it work with such different people in the duo? What does each of you bring to a typical Apothek project?**
NML: I think the main thing we each bring is flexibility and being complementary to each other. Usually, this will mean that if one of us is stuck, the other might see a way out of it. Since we largely work apart, it gives us the ability to have some sense of objectivity that bands that work as a group all the time might lose. Morten brings the lyrical side and has a great sense of what makes a track work, as we want it, whereas I more often will be more focused on details. So in the grand scheme of things, while writing, Morten often has the macro focus and I have the micro focus.

MT: **So it sounds like you work apart a lot, but how did each track start?**
NML: It usually started with one of us having a phrase, a chord progression or simply a sound. →

From there, I'd have a couple of sessions in the studio trying to expand the idea and send it to Morten, who would start working on the lyrics and melody lines, before workshopping it together again.

MT: How did you approach putting the album together in terms of studio use?
NML: Apothek is, without doubt, a band where we use the studio as an integral part of the process. We never really had a long period of time where we were supposed to go somewhere and record the album. It was made over a longer and more spread-out period, working when we had time between other projects.

Morten and I also recorded his solo record, *The Sparks Flew Up*, together in the same period we wrote parts of the Apothek record and I produced and played with other bands. So, until a point where all the tracks had a clear shape and intention, we mostly worked separately and spread out.

All over the record, we had this mantra that felt like an integral part of the band's sound. It was trying to make acoustic sounds sound really mechanical and mechanical sounds sound acoustic. So I've spent a lot of time in my loft reamping synths, running them through different gritty effects and at the same time, slicing and trying to make my old piano sound like a strange synth. That's been a really fun and new way to work. Also, Ableton Live was a very integral part of the whole thing.

The last stage was spending a few months with Lars Horntveth of Jaga Jazzist in his Oslo studio – he produced the record with me. We wanted to make sure all the tracks had been optimised as far as we could take it and then Lars came in with fresh ears and helped with arranging and trimming away excess stuff. He has a great sense of arrangement and form so that was probably the most intense time in making the record where an 'outsider' to the project could come in with little previous knowledge of the tracks or their process, and help us see where we could go even further or back off a bit.



"We made the acoustic sounds sound mechanical and mechanical sounds sound acoustic"



Above: Nils Martin Larsen (left) and Morten Mykleburst of Apothek

MT: Are there any typical processes that you use?
NML: I use the internal microphone in my Mac a lot. It's noisy and grainy and stupid, but it has a character that I appreciate. Every so often, I'd track a piano with proper microphones and dub it with an alternate take, just recording it with the internal mic placed on top of my piano.

Then I'll treat the proper recording to make it sound grittier and I'll try to make the laptop recording sound prettier. Then I'll balance the two and keep them both in the mix. A lot of the time, I will pitch-shift one of them or change them using a formant filter or Melodyne as well. Sometimes, it works; sometimes, it's a total disaster. For example, the main chord parts in [debut single] *Family* is a Rhodes recorded like this, sliced up and triggered with a drum pad.

THE APOTHEK STUDIO

MT: Tell us a little about your studio?
NML: My studio is located in an old industrial



building on the east side of Oslo and has been my base for six years. I run it with my producer friend Sjur Lyseid and we've rebuilt and expanded it over the years until it got to the state it is in now – consisting of a loft studio that is my main space, and a downstairs analogue recording studio. We came across it accidentally and initially rented just the first floor, and expanded a few years later. We've made a lot of records, film music and other stuff between us in this space.

My loft is made more for when I work alone, while the downstairs studio is made more for recording acoustic stuff and bands, as Sjur often produces bands. The main space is based around a Toft ATB 24 console where all my instruments are hardwired to their own channel, making it very easy to work quickly and efficiently. So this setup is heavily based on instruments and synths and being able to get stuff on tape really quickly.

I have everything from old Korgs like the Sigma and Yamaha CS-15 to newer stuff like Vermona Mono Lancet and Pocket Piano and Moog SUB37, trying to cover most bases. I also have an extensive collection of effect pedals and old spring reverbs. Then there's my turn-of-the-century piano and all my guitars, as well as all kinds of strange bits and bobs I've collected over the years. I've been an instrument collector since I was a kid, so I can't have everything out at the same time, but luckily, we have a lot of storage space as well…

The downstairs studio is based around an old Neve mixer and old outboards, including a Gates STA-Level, 1176s and various other great gadgets. We also have a selection of vintage microphones that Sjur has collected over the years, and a ton of amazing old guitars and amps. Both rooms are really great places to work.

MT: You use a lot of synths in your music – do you have any favourites?

NML: On the synth side, Vermona Mono Lancet, Korg MS-20, Critter & Guitari's Pocket Piano, Yamaha CS-15, four DSI Mopho x4 and Waldorf Micro Q have also been a few of many favourites for this record.

I've always really loved sound sources that you have to fight to get the right sound out of. So synths

APOTHEK KIT LIST CONTINUED

OTHER HARDWARE/ SOFTWARE

- **Moog MF-102 Ring Modulator and filter pedals**
- **Old phasers from MXR**
- **Vintage chorus pedals from EHX**
- **Shady, DIY octave pedals from Russia and Germany**
- **A lot of IR reverbs. Both self-made and various odd-sounding spaces made by others**
- **TAL's free synth plug-ins. Both their Elek7ro and their Juno plug-ins**
- **Also ACE and Bazille from u-he**
- **Various sample libraries from Sound Dust and Rhythmic Robot**
- **Simpler, Sampler and Impulse in Ableton**

LIVE

- **Two Macs with Ableton**
- **DSI synthesisers**
- **Vermona synthesisers**
- **Novation keyboards**
- **Critter & Guitari synthesisers**
- **Bastl Instruments sampler**
- **Elektron Octatrack and drum pads**

THE APOTHEK PROCESS

A huge part of the Apothek sound is processing – as they say, trying to make electronic sources sound acoustic and vice versa. Here, Nils details both his hardware and software solutions

MT: We like the sound of all of that hardware processing you use including all of the pedals. Tell us a little about how you use it…

NML: I try to use hardware as much as I can, since I have a lot of it around. I've been in love with old, strange spring reverbs for a long time and have a few of them. My very favourite is an old Premier tube spring reverb. I use it for almost everything and keep plugging it in as an insert on the mixer. Drum machines, synths, vocals, pianos – everything goes in! We also use the Boss CE-3 chorus a lot. Sjur has a pretty crazy collection of old tape delays consisting of a Space Echo, Maestro Echoplex, Korg Stage Echo and Binson Echorec, so I tend to sneak downstairs to borrow them as they beat any plug-in. Amps are also a huge part of my workflow, so I have a few of them in my patchbay making it easy to re-amp stuff – great to give a sound some texture.

MT: What about other processing?

NML: Digitally, I have a few sets of IR reverbs made by people I know and have also fallen in love with some of the sample packs from Sound Dust and Rhythmic Robot. And again, I use Ableton Live – in many ways as one of my main instruments. Drum Rack and Impulse holds all my samples made over the years; Sampler is super easy to create cool textures out of Morten's processed vocals and the warp and slice features are great to make cool stuff with some personality.

We've also had a lot of fun processing Morten's voice on this record. A lot of percussive sounds are made from small sounds between takes, for instance. No one would know what they are, but they bring something organic into the mix instead of just working with normal samples. We've also done a lot of really fun digital harmonising that went way past the normal fixed-interval pitch-shifting thing. We kind of went a bit crazy with Melodyne there for a while. Haha! My piano is also a big part of this. It's been recorded on everything from U47s to the mic inside my laptop, and both options are valued absolutely equally, which is kind of crazy. I have to say, Ableton Live has to get a lot of credit here.

MT: Give us your top piece of advice…

NML: At some point, you get to where you know exactly how to get the sound in your head to end up on tape. My advice would be to keep trusting that voice and not listen to what sounds everyone else is making and comparing yourself to others too much. You'll only go mad. It's really hard sometimes, especially in the periods where you're kind of stuck (and you will be). But building some self-awareness in what you do and trusting your own sense of aesthetics is one of the most important things for me. A concept that gets challenged daily, luckily.

MT: Any advice from the music industry as a whole that you can pass on?

NML: Not really. I try to focus on what I know I can do, and that is not industry-type things. If anything, it'd be to surround yourself with people you trust and who know the rest of the stuff. Luckily, we get to work with Propeller Recordings who know what they are doing, and we feel confident following their advice on matters we're not experts on.











A selection of shots of Apothek's mix-and-match studio gear, including drum machines and some choice outboard units

APOTHEK KIT LIST

DAW
- Ableton Live – used more as an instrument than a DAW
- Some edits in Pro Tools

FOR SYNTHS, MAINLY ANALOGUE STUFF:
- Korg MS-20
- Vermona Mono Lancet
- Korg Sigma
- Sequential Prophet 5
- DSI Mopho x4
- DSI Prophet 12
- Moog Sub 37
- Roland SH-09

VARIOUS PIANOS FEATURED – AT LEAST SIX OR SEVEN, VARYING FROM SMALL PIANOLA TO FULL-SCALE CONCERT GRAND…
- Wurlitzer 100A
- Hohner Pianet T
- Various Rhodes models

DRUM MACHINES/ ELECTRONICS
- Field recordings turned into rhythmic patterns with Elektron Octatrack
- Drum programming done with self-sampled sounds in addition to Elektron
- Elektron Machinedrum
- MFB & Jomox drum machines

CONSOLE/OUTBOARD
- Toft ATB 24 mixing console
- Old Microtech Gefell mic from the 50s
- Gates STA-Level (bought from an American radio station active in the 60s)
- Mixed on a Trident 80B console

MONITORS
- Dynaudios
- Yamaha NS-10s
- Old Klein + Hummel speakers



The main studio is geared towards capturing ideas, with all of the instruments hardwired to their own channel on the Toft console

like the Formanta Polivoks – a Soviet-built monster of a monosynth – is one of my absolute favourites. Elektron's Octatrack and Machinedrum have been huge parts of Apothek ever since we made the first track, *Family*. The sequencers in those things are such amazing tools – go Sweden! Also, drum machines from Jomox are featured heavily.

MT: You have a very esoteric collection of gear. Is there anything you want to add to it?

NML: I've been dreading the day where I start a modular-synth setup for a long time. It's on the wish list, but even higher on the list of things I fear – because I know that within a year, I'll be the pasty white guy with a beard and a friggin' cape, with a wall full of modules. Time will tell… An ARP2600 would also be great fun. Apart from that, I'm actually quite content at the moment! I never thought I'd say that.

MT: What studio tech would you like invented?

NML: I remember when I first saw that they had made Melodyne polyphonic, and I thought that everything had changed. It just felt so new and complex that it melted my brain. I don't think I can even begin to guess at what brilliant tech nerds will be able to do in the future, but I'm hopeful that I can be interested in it and utilise what they make in a



musical way when they get there.

I'm excited to see what people like Audioease, Melodyne, Ableton, Max and stuff like that come up with in the future. And I'll be sitting there with my keyboards trying to understand what they allow us to do and hope I'll figure out cool ways to use it.

MT: Is the album all that you expected it to be?
NML: We just wanted to make the best album we could at the time we made it, and we wanted it to sound natural and not forced in any way, despite it being very detailed. Most of all, we wanted it to be a fun new way of working for the both of us, resulting in a record we could be proud of – both as a band and individually – knowing we'd made something that neither of us could have done alone.

MT: What have you got planned now?
NML: We're extremely excited to show the album off and see how people react to it. We 're also doing a lot of live shows and hoping to keep expanding that list. Lately, we've started thinking about recording new music. We've got some ideas flowing around, but nothing concrete yet.

MT: Finally, what is the future of music production, in under 100 words?
NML: I'm horrible with the Tarot cards, but I hope that we get to see music production in the pop scene turn more performative and less about laptops and headsets in hotel rooms. I like dusty instruments with dents and scratches making new sounds. [Too right – Ed].

Apothek's self-titled debut album is out now via Propeller Recordings. www.apothekmusic.com MTF



The Apothek challenge: make this beautiful old piano sound like a strange synth

CHOOSE YOUR SUBSCRIPTION PACKAGE



THE BUNDLE SUBSCRIPTION...

- Save 37% on the single copy prices*
- Just £34 every 6 issues on Direct Debit*
- Free UK delivery direct to your door
- Access to the app through iTunes, Kindle Fire and Android

CODE - MTP169B



THE PRINT SUBSCRIPTION...

- Save 30% on the shop price*
- Just £25 every 6 issues on Direct Debit*
- Free UK delivery direct to your door

CODE - MTP169

OTHER PAYMENT METHODS



PRINT + DIGITAL...

- **UK** – £74.99 for 12 issues by Credit/Debit card, save 30%
- **Europe** – €72.07 every 6 issues by Continuous Credit Card, saving 30%
- **Australia** – $90.75 every 6 issues by Continuous Credit Card, saving 30%
- **Rest of world** – £65 every 6 issues by Continuous Credit Card
  (Sorry, bundle subscription not available to our USA readers)

CODE - MTP169B



PRINT ONLY...

- **UK** – £53.50 for 12 issues by Credit/Debit card, save 26%
- **Europe** – €50 every 6 issues by Continuous Credit Card, saving 34%
- **Australia** – $76.50 every 6 issues by Continuous Credit Card, saving 15%
- **Rest of world** – £43 every 6 issues by Continuous Credit Card
- **USA** – $52.50 every 6 issues by Continuous Credit Card, saving 45%

CODE - MTP169

MusicTech SUBSCRIPTION OFFER

Save up to 37%

when you subscribe to MusicTech



TWO EASY WAYS TO SUBSCRIBE

anthem.subscribeonline.co.uk/musictech

ENTER **MTP169B** (bundle) OR **MTP169** (print only) in the promo code box

0844 844 0398*

*Calls cost 7 pence per minute plus your phone company's access charge

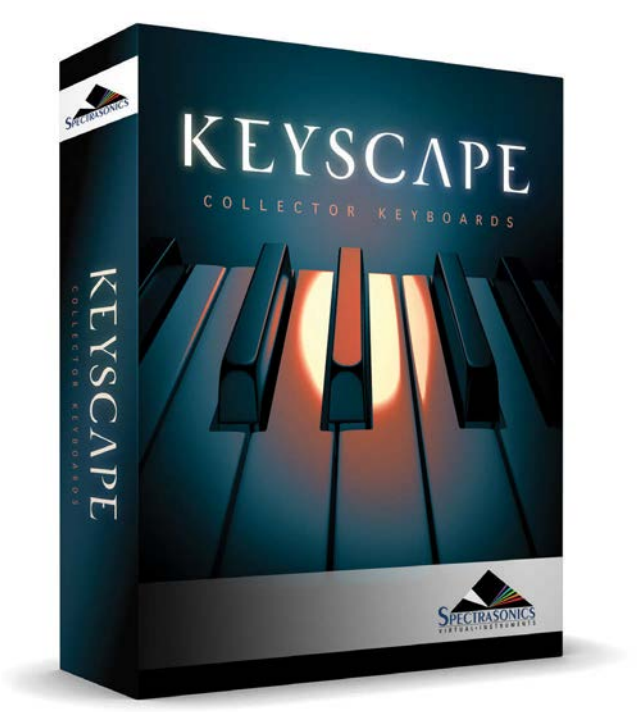
(*overseas readers* +44 1795 592849)

QUOTING CODE **MTP169B** (bundle) OR **MTP169** (print only)

OFFER ENDS **30 APRIL 2017** CODE **MTP169B/MTP169**

Your subscription will start with the next available issue unless you state otherwise. There are 12 issues in a year.

*30% & 37% saving only available to subscribers paying by UK Direct Debit. The print saving is calculated against the cover price x12 issues. The bundle saving is based on buying 12 print copies in a shop plus 12 digital single copies. Details of how to access your digital editions will be sent to you with acknowledgement of your subscription order. Please provide your email address for this.





SPECTRASONICS

Keyscape

With 36 rare instruments to play with, could Spectrasonics' new virtual instrument, Keyscape, be the ultimate software keyboardists' dream? **Mark Cousins** investigates…

Details

Manufacturer
Spectrasonics
Price **£285**
Contact **www.spectrasonics.net**

Key Features

- **500+ 'best-in-class' sounds**
- **36 instrument models and hybrid 'Duo' patches**
- **32-way velocity switching**
- **Round robins**
- **Authentic circuit-modelled amplifiers and effects**
- **Full Omnisphere 2 integration**

Having waited almost four years to move from Omnisphere 1.5 to version 2, it would be fair to say that Spectrasonics doesn't exactly like to rush its creative process. However, the task of crafting a world-class virtual instrument with thousands of presets isn't something you can roll off the production line every few months.

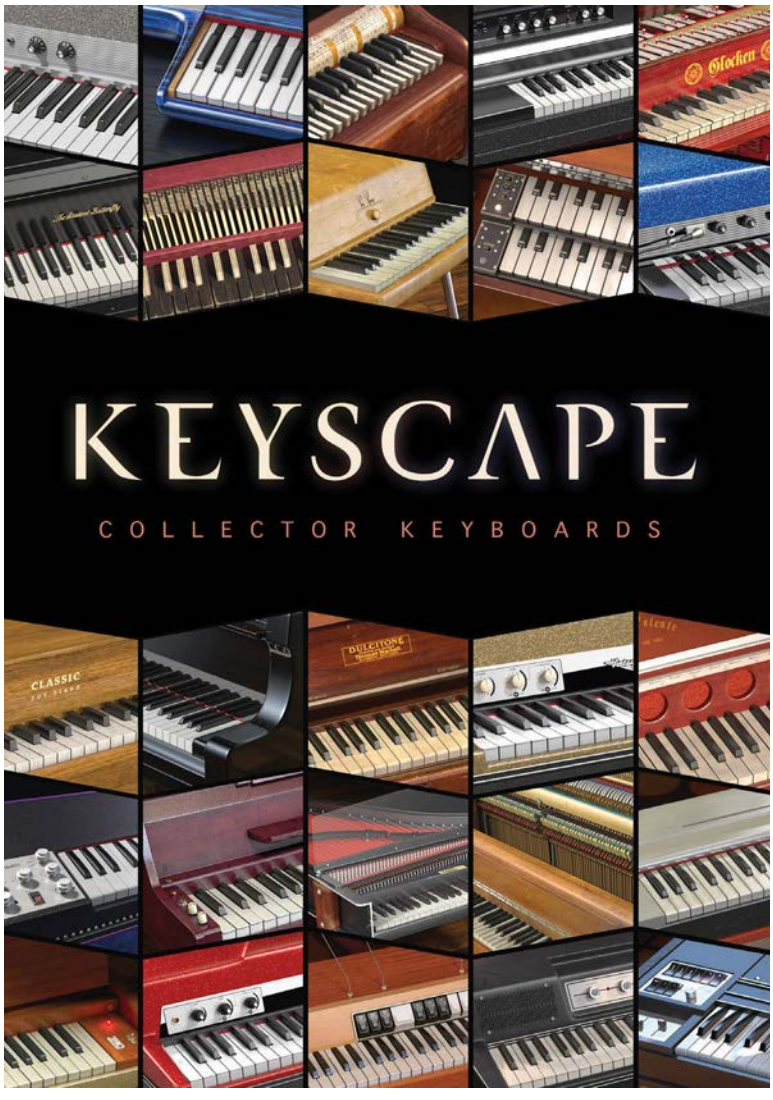
In truth, instruments like Omnisphere, Trillian and Stylus RMX are all products of years of investment. From the task of sampling a range of acoustic and electronic sound sources, through to creating a creatively rewarding virtual instrument – not to mention all the presets required to show all of this off – Spectrasonics has demonstrated perfection is time-intensive task. Given this attention to detail, the fact that Keyscape has been almost 10 years in the making shouldn't really come as a surprise.

The concept is simple – an exhaustive library of 36 sought-after 'collector' keyboards painstakingly multisampled and presented in an Omnisphere-like interface. Scan the list of instruments and you'll be impressed by the comprehensive set of electromechanical and acoustic pianos, or the host of intriguing additions like the Dulcitone and Chimeatron, but does Keyscape really raise the bar over what many players have access to already?

> Instruments like **Omnisphere**, **Trillian** and **Stylus RMX** are the result of **years of investment**

Thanks for the memory

Keyscape is available as a download direct from the Spectrasonics site, or as boxed edition, with the content supplied on two USB drives. At a total of 77GB, Keyscape is certainly a big library, although when you consider than many of the 36 instruments have been sampled with up to 32 velocity levels as well with round robins, you can actually see that the space has been used effectively, rather than just being squandered. As you'd expect, Keyscape benefits from a healthy RAM allocation (8GB is the recommended standard, but we tested it with 64GB) and access to an SSD drive, but there is a Thinning



The 'Collector's Keyboards' in Keyscape – 36 in total – have been tracked down and deeply multisampled

What **immediately grabs you** is how **good Keyscape feels** underneath **your fingers**

button to lighten the load on older computers or on slower HDs, as well as a variety of options to fine-tune how Keyscape streams its sample data.

If you're mainly interested in 'big players' included in Keyscape (like the LA Custom C7 piano, for example) there's the option of a 30GB 'lite' install. Rather than thinning down the velocity layers, the lite installation reduces the number of keyboard models from 36 to eight, which should be sufficient for those playing live, who simply need a good version of each principal instrument group. If you're an existing Spectrasonics user, it's also worth noting that Keyscape patches can be opened directly from Omnisphere (more on this later), assuming you've upgraded to version 2.3 or higher.

Key signature

Keyscape is a software instrument that demands to be played. We could wax lyrical about the quality of the sounds, of course, but what immediately grabs you is how good Keyscape feels underneath your fingers. Some libraries – even the good ones – can feel like a series of recorded snapshots triggered by your fingers' movements, whereas Keyscape feels and plays like a real instrument. Understanding why that is, though, reveals why Keyscape is such sonically effective musical tool.

Firstly, the efforts expended in sourcing and restoring the instruments used in Keyscape has reassuringly nerdy attention to detail. The C7 Grand Piano, for example, belongs to renowned Los Angeles piano tech, Jim Wilson, who's spent years tweaking his piano to perfection with Blue Point hammers and Weickert felt. Likewise, multiple generations of classic Rhodes, ancient clavichords, and vintage tack pianos have all be dusted off, polished up and restored to their former glory.

In short, Spectrasonics has sourced what should be considered the *de facto* standard for each keyboard model, so that the patch you load sounds similar to countless versions you'll have heard on record.

Working in Omnisphere

Opening Keyscape patches in Omnisphere is a real eye opener, seeing how the basic sample material – presented as a series of 179 Soundsources – has been extended and developed using Spectrasonics' STEAM engine. One immediate benefit is the ability to customise the effects as you see fit, either tweaking existing parameters, or adding some of Omnisphere's more experimental options (like the Innerspace module, for example).

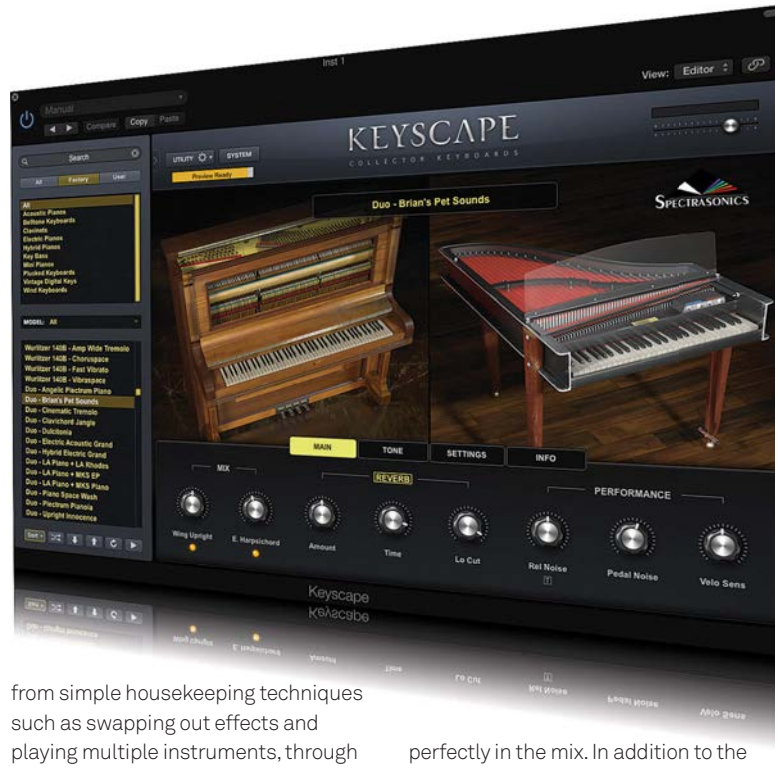
One important addition is the ability to use Omnisphere's functionality in respect to multi patches, including the Stack and Live modes. In this case, though, you'll want to ensure you have plenty of RAM if you're working with some of the larger patches.

Dynamic performance

The next-most-important quality is Keyscape's response to velocity – something that is clearly high in Spectrasonics' mind, given the number of pre-programmed velocity scaling maps included with Keyscape. All the instruments demonstrate an expressive range of tonal and dynamic colours, which respond impeccably to your playing. The C7 Grand Piano, for example, sounds superbly warm and intimate in its lower velocity settings, while the Rhodes have all the nasal-like bite you expect when you wallop the keys. Certainly, my weighted Akai MPK88 keyboard has never felt more responsive than when it's been hooked up to Keyscape.

Another area that many developers often get wrong is noise. Again, this is another factor Spectrasonics seems to have nailed in Keyscape, understanding that those additional creaks, knocks and clunks make an instrument 'characterful', yet still giving you control to remove or tame noise elements as you see fit. Many of the instruments have direct controls for added 'character' – like pedal noise, for example, or, in the case of the Fender Rhodes instruments, the mechanical noise the piano makes as you hit the tines. There's also a handy De-Noise →

→ button that has a direct impact on the hiss that some might find offensive.



Keyscape's Duo Patches are instantly musically inspiring

Graphic delights

Keyscape's interface will be immediately familiar to anyone who's used Omnisphere 2, borrowing both its look and principal means of operation. For example, on the left-hand side of Keyscape, you'll find a collapsible Patch Browser, letting you navigate the 500 or so patches included in Keyscape in an easy and effective way. Once loaded, each instrument has its own collection of macro controls, organised across a series of tabbed windows. The precise set of controls is customised for each instrument, but you can expect to find controls for different mic sets (usually Close and Room) and integral instrument effects, as well and compression and EQ settings.

Interestingly, each keyboard model is presented both in its relatively neutral form, as well as a series of presets that explore the sonic treatments that define the instrument 'on record'. Rather than use a completely different sample set, though, Keyscape instrument variations make use of the effects options introduced in Omnisphere 2, including various modelled amplifier settings, reverbs, chorus units and so on.

For many of the electromechanical instruments (like the Clavinet, to give one example), these effects are indispensible. Thanks to some surprisingly authentic modelling, they play a part in really adding to the musicality and versatility of the basic instrument model.

To access the full 'under the bonnet' editing, you'll need to open the patches in Omnisphere 2, where Keyscape appears as its own discrete library. Once loaded, the patches appear with the familiar macro controls found in the Keyscape editor. Close this window down, though, and you'll see the more familiar STEAM-driven Omnisphere editing pages.

As you'd expect, the journey into Omnisphere brings about an exponential rise in creative possibilities: from simple housekeeping techniques such as swapping out effects and playing multiple instruments, through drilling down to the more focused and creative extremes of techniques like granular synthesis.

In use tip

Although you can't edit the patches with the same degree of detail as you can in Omnisphere, it's still possible to tweak the variety of macro settings offered and then select 'Save Patch As' from the utility menu. In the case of the C7 piano, this would let you create preferred EQ, Compression, Reverb, Tape Saturation and Performance settings that match your playing or style of music.

Key sounds

The cornerstone of Keyscape has to be the C7 Grand Piano, which for many, may well be worth the entry price alone! Having used plenty of alternative virtual pianos over the years, Keyscape's C7 is certainly up there with the best: versatile enough to move from pop to classical, beautifully expressive, and realistic enough to stand on its own in a mix. Compared to other sampled pianos, the C7 Grand Piano favours a more intimate studio setting, rather than the roomier offerings included in instruments like Garritan's CFX Concert Grand. Ultimately, it's a case of horses for courses – if you like you pianos drenched in natural studio acoustics then the C7 might not excite you, but if you like a controllable and adaptable instrument, then the Keyscape solution is a winner.

When it comes to upright pianos, some alternative libraries can disappoint, often sounding dull and clunky rather than capturing the quirky charm of a real upright piano. Keyscape's Wing Upright, though, is a real triumph, with all the rough-and-ready edges of an upright piano, but with a distinctly hi-fi tone that sits perfectly in the mix. In addition to the main upright sound, you'll also find a version of the Wing Upright that feature a Tack modification as well as an intriguing (and very useful) Tremolo option created using small metal beads.

As you'd expect, the electromechanical keyboards in Keyscape are covered in plenty of detail. Just the Rhodes section alone seems pretty exhaustive: with an original MK1; a customised 1970s model; the infamous Piano Bass instruments as used by the Doors and even a modern recreation, in the form of the Vintage Vibes Tine Piano. Away from the Rhodes, you'll also fine a healthy collection of Clavinets, Pianets and Wurlitzers, all captured in a way that mixes both the mechanical noise of the instrument alongside the more familiar direct output. Most importantly, they all sound great, balancing a crisp tone with plenty of low-end punch when needed.

> Keyscape's **cornerstone** is the **C7 Grand Piano** – it may be **worth the entry price** alone

If you're a fan of Tom Waits' eclectic use of instruments, then there's a real treasure chest of unusual acoustic keyboards included in Keyscape. From Celeste to Toy Piano and Dulcitone, there's a range of quirky acoustic delights that clunk, click and chime in just the right way!

As with the piano, the emphasis is on an intimate studio setting, which really helps bring out the detail of each instrument. However, as each model is

generally presented with both Close and Room mics, you can always opt for a roomier sound.

Just when I thought I'd exhausted the fun possible from Keyscape, I turned my attention to the Duo patches. As the name suggests, the Duo patches layer two models as part of the same patch, so, for example, the Tack setting on the Wing Upright is layered with the Fender Rhodes. The Duo patches are instantly inspiring and exciting to the ear, arguably illustrating how some of the more unusual acoustic instruments work nicely as a means of doubling the main instruments. Possibly the only downside of the Duo patches is that you're restricted to the pairings Spectrasonics offers, unless of course, you work with the Keyscape patches inside Omnisphere 2.

Keys to the kingdom

Factoring in both the range of models included in Keyscape, combined with the sheer quality of sampling involved in capturing these unique instruments, there's little doubt that Keyscape is an impressive package. Even so, given the wealth of content included with applications like Logic Pro X or Komplete 11, it's highly likely that many players will have many (though not all) of these instruments in an existing form. Komplete 11, for example, now comes with three workable acoustic pianos – The Gentleman, The Maverick and The Grandeur – as well as a host of other vintage keyboards as part of Kontakt's sound library.

The real appeal of Keyscape, and the reason many musicians will jump on it, is that it offers some of the finest virtual versions of these classic vintage keyboards. Individually, of course, there still some great 'boutique' libraries available for Kontakt, usually focusing on one or more of Keyscape's instrument sets.

Also, given that Spectrasonics has concentrated on just two microphone sets, there may still be a place for solutions like Spitfire Audio's North 7 Vintage Keys that offer four mixable channels per instrument.

However, we'd still prefer one good playable version of any instrument, rather than a host of (occasionally) superfluous variations.

As always, the quality of what's offered in Keyscape has a price tag, both in respect to the cost of the software itself, as well as the cost of the computer to get the best from it. Big patches like the C7 Grand Piano will need a fast drive and plenty of RAM both to ensure fast loading times and plenty of voices – something that is true of any contemporary sound library with an emphasis on quality. As for the cost of Keyscape itself, well, looked at from a per-instrument basis, the models are less than £10 a pop – a bargain, by anyone's standards!

Ultimately though, my real delight in Keyscape was in playing the instrument, getting lost in the same way you would if you had a Grand Piano in the corner of your studio, or a Fender Rhodes. Probably my only real want, as such, was for more: what about other classic keyboards like the Hammond organ, for example (something we've rarely heard a good virtual version of since Native Instruments discontinued the B4 II), or some Mellotron Choirs and Flutes? For now, though, I'm more than happy with Keyscape, as it remains one of the most enjoyable and playable virtual instruments I use. **MTF**

> Ultimately though, my **real delight in Keyscape** was simply **playing the instrument** and getting lost in it...



Keyscape appears as a separate library in Omnisphere 2, where further editing can take place

Alternatives

When you consider the full range of sound on offer, there's little that directly completes with Keyscape. Both MOTU's Electric Keys (£299) and Spitfire Audio's North 7 Vintage Keys (£199) cover the electromechanical angle well, although Electric Keys is starting to show its age somewhat as it was first released back in 2008. North 7 Vintage Keys might be a good alternative for those that enjoying 'mixing and matching' different combination of mics and amplifiers, with at least four alternatives for each instrument. Then of course you could consider NI's Komplete Ultimate which has more, different instruments but costs more.

MTF Verdict

+ Fantastic range of classic keyboard sounds
+ Detailed and expressive instruments
+ Great selection of 'eccentric' instruments
+ Duo patches extend musical variety

- It's a big install
- Some players may have alternative versions

Keyscape is a real labour of love, offering a definitive set of sampled keyboard instruments that's both responsive to your playing and inspiring to use...

10/10



BIG FISH AUDIO

Cyborg

Looking to add a touch of the mechanical, ethereal and otherworldly to your productions? **Matthew Mann** is menaced by the machine…

Details

Price **£139**
Distributor
Time And Space
Contact **via website**
Web
www.timespace.com
System Requirements
Kontakt 5.5 or Kontakt Player

Features

- Drones & atmospheres
- Drums & percussion
- Guitars & basses
- Keyboards
- Cinematic FX
- Multi-layered instruments
- 8.4GB Wav data
- Over 300 instrument patches

In the last few years, there's been a shift in what's expected from a soundtrack composition. Traditionally, a soundtrack was composed with strictly orchestral instruments in mind. Composers like John Carpenter, Hans Zimmer, BT, Giorgio Moroder and others have used electronic instruments in their compositions for film and TV… forever expanding the soundtrack palette.

With this paradigm shift, the composer now has pretty much anything available to use in composing – and several sample-library companies have developed libraries specifically tailored to today's cinematic composer and the ever-expanding palette for movie music.

One such library is Cyborg from Funk/Soul Productions and Big Fish Audio. This library combines organic and electronic elements into a swirling, evolving, mechanical whirlpool of cinematic sound. So, if you're a composer of cinematic soundscapes or just looking for an interesting library of sounds to add to your own productions, this may be a good place to start.

Man… or machine?

Cyborg includes swirling, droning, pulsating patches designed to evoke images of worlds we've only dreamt about and depths that we've had nightmares about. The sounds are, at times, very rigid and unnatural – at other times, they're very organic and loose feeling. This library made me feel uneasy and concerned for my own safety (in a good way).

Cyborg is a Kontakt library that requires Kontakt 5 and can even be used with the free Kontakt Player (again, at least version 5). Opening the library presents the user with a splash-screen image that looks like it was ripped straight from the mind of HR Giger… and it is very representative of the sound and feel of this library. There are five folders in total. They are:

- ***Drones & Atmospheres (40 patches)***
- ***Drums & Percussion (27 patches)***
- ***Guitars & Basses (29 patches)***
- ***Keyboards & Misc (37 patches)***
- ***Cinematic FX (15 patches)***

> This library made me feel **uneasy** and **concerned** for my own safety **(in a good way)**

Each folder leads you down a dark and disturbing path. Drones & Atmospheres is exactly what it sounds like – dark, swelling, bleeping and brooding. The swirling, glitching, resonating madness of this folder would make an excellent backdrop for your latest musical trip through hell.

Although there's no key-switching or multi-layered, velocity-switched anarchy going on behind the scenes, the patches are still very good. You'll only get changes in volume based on playing at different velocities. I would expect a little more in this department with all the possibilities available in Kontakt, but I really can't complain. The sounds really are quite good.

Drums & Percussion contains several kits and percussive loops. From glitch kits to percussion drowned in filth, these percussive elements and loops add fuel to your soundscape.

The loops are not all tightly locked to a metronome, either. Some are in fact very organic and loose – reminding us that inside this mechanised monster is the beating heart of something that was once human.

Moving on, we have Guitars & Basses, a collection of stringed instruments whose sole purpose is to crush your soul. Metal-guitar patches and loops give you chunky, screaming filler. The basses are so deep, chorus-y and gothic that they would make The Cure's Robert Smith extremely proud. There are also heavy guitar loops that consist of chugging metal chords and riffs that can add ballsy, distorted rhythm and movement to your undulating compositions.

The face of darkness

Cyborg's front panel consists of 12 basic controls. These do not change as you move from one patch to the next, but are standard throughout the library. The first eight controls are basic sliders that give you more or less of their effect. These consist of: Distort, Lo-Fi, Limiter, and Reverb on the left side and Bass, Treble, Return, and F/Back on the right. These do as you might expect. Distort adds filth to your patch, Lo-Fi adds noisy bit reduction, Limiter makes it loud, and Reverb... well, you know what that is. There's even a selector below Reverb giving access to a variety of reverb patches from Compact Room and Metal Chamber to TankVerb and Wood Room. These are simple and good. Bass and Treble provide basic EQ for patches. Return determines that amount of delay you can add to your track. F/Back is the number of repetitions of that delay. Below the delay is a selector that lets you add different types of delays including 1/4 note, 1/8 note, 1/16 note, dotted and triplet, a doubler, slapback and more.

Inside this mechanised monster is the beating heart of something that was once human

Keyboards & Misc fills a void that you didn't even know was missing. This folder contains mournful piano loops, buzzy arps, distorted synth riffs, dark vibes, more ethereal pads and voices, a couple of nasty leads, and sound effects like piano-string scrapes and such. I found use for many of the patches in this folder... especially the arps. One of my favourite things to do while checking out this folder was to stick random MIDI drum parts in and let them play some of the SFX patches. Killer results without even trying! (See disc for examples.)



Cyborg's a versatile library, and may even appeal to extreme metal heads

At the bottom of the interface are the two final control sections. On the left is the ADSR and does what you would expect. On the right is a basic filter with Cut and Rez controls. There's also a little button that lets you choose the type of filter from 4-pole LPF, 2-pole LPF, 4-pole HPF, 2-pole HPF, Band Pass and Notch. It's basic, but sounds good.

One thing I couldn't seem to do with Cyborg was automate parameters... like the filter cutoff. For example, I found that Basic Arp 02 was a great, simple patch with lots of character. Had I been able to automate the filter cutoff on the interface, it would have become an instant Dark Wave bass riff a la Depeche Mode or Cause and Effect. I don't know if this was a limitation of the programming in Cyborg, or a limitation in Kontakt, but I have to believe it was the former rather than the latter. Yes, I can add a separate filter on the track and accomplish the same thing, but it would have been nice to add this functionality inside Cyborg.

To be... or not

I really enjoyed digging into the sounds in Cyborg. They're very good. The loops go from tightly rigid sequences to organic and swirling. I discovered uses for most of the sounds that I normally wouldn't have thought twice about.

This library really does lend itself to experimentation. The results did wonders on some 'less than stellar' tracks that I'd been working on – adding grit, grime and cybernetic excitement and making my tracks much more interesting. It's not the cheapest sample library out there, but it's rather unique and just might give you that little something extra you've been missing. MTF

MTF Verdict

+ Great sound library of useful sounds and loops
+ Lends itself to experimentation
+ Simple, but effective interface
+ Can be used with Kontakt Player

- Could stand a little thought on velocity-layering
- No automatable parameters

Overall, this is an excellent library of useful sounds. It initially seems niche, but can be used in lots of different styles. Strongly recommended for those who desire something a little darker in their productions.

8/10



Choice 9/10 MusicTech

OUTPUT Movement

Movement is Output's first audio plug-in, but does it match the calibre of its previous sample-based instruments? **Mark Cousins** finds out…

Details

Manufacturer **Output**
Price **£122**
Contact **Time+ Space 01837 55200**
Web **output.com**

Features

- 300+ presets
- Sidechain, LFO, Step Sequencer rhythm modulation
- Flux and Randomizer
- Filters, EQ, delay, distortion, compression and reverb
- X-Y pad
- Live and studio use

Over the course of its three previous products – REV, SIGNAL and EXHALE – Output has developed a range of Kontakt-based virtual instruments with a consistent level of ingenuity and creative flair. Anticipation for the company's fourth release was high, then, but few expected that Output would make the bold jump between the world of sample-based instruments and the parallel universe of audio plug-ins.

Movement is Output's first audio plug-in, and it's designed to impart a range of rhythmic qualities onto any number of audio sources that you pass through it. However, given the high standards we've come to expect from Output instruments, does Movement make the grade as an audio processor?

To fully understand what Movement has to offer, it's worth taking some time to describe its signal path and the creative possibilities it entails. Movement is powered by two parallel effects engines (A and B), each of which has up to four different effects slots that can be used to instantiate a range of signal-processing options, including filters, delays, reverbs, compression and distortion. What makes Movement really interesting, though, is the introduction of four rhythm generators, spread across the two FX engines. The rhythm generators are presented as LFOs, step sequencers and a sidechain input, all designed to add movement to the aforementioned, and otherwise static, FX engines.

Sonically speaking, the output of Movement can be characterised as a range of animated FX treatments that vary given the material presented to them. A vanilla-sounding synth pad, for example, suddenly comes to life with pulsating filter treatments; or a simple guitar line suddenly becomes a rhythmic feature with a choppy, staccato-like gate, bathed in echoing delays. For new users, it's well worth spending time with Movement's presets just to see the breadth of what the plug-in can achieve, and thanks to a clear preset-tagging system (using descriptions such as choppy, pulses or sidechain), it's easy to find the treatment that you're after.

Into the groove

Part of the appeal of Movement, as with Output's other products, is the way the interface integrates itself into the process. What could be a complicated plug-in, therefore, remains intuitive throughout the entirety of its use – immediate and accessible for the first-time user, yet deep enough to keep ardent knob-twiddlers happy for hours. The accessibility of Movement is perfectly illustrated in respect to the modulation mapping, where the rhythm sources are assigned to parameters within the FX engine simply by a process of drag-and-drop. Once assigned, numbers in the centre of each pot indicate the routing, as well as small bar graphs on the bottom of the pot to see the depth of modulation.

> Accessible for the **first-time user**, yet deep enough to keep ardent **knob-twiddlers** happy

The beating heart of Movement is the four rhythm sources: each set to work as an LFO, step sequencer, or sidechain modulator. Looking first at

the LFO, it's pleasing to see a range of creative waveshapes (like Quad Stepped Sawtooth, or Notch) as well as the conventional sine, triangle and sawtooth shapes. We also enjoyed the addition of a Chaos control, which varies the amplitude of the LFO on each step, making the output more interesting to the ear. As you'd expect, Rate can be locked to various clock divisions – from 1/64th notes, right up to eight bars – as well as including both dotted and triplet settings.

Flux capacity

One of the more intriguing elements has to be the Flux control, which lets you modulate the speed of a selected LFO/step sequencer using the rate of the other. The result of using Flux is a unique form of shuffling or skipping movement – whether it's a subtle Tangerine Dream-like movement between 1/16ths and 1/32nd settings, or a more radical Dubstep-like modulated wobble moving from 1/8ths to 1/24ths. Not surprisingly, Flux plays a big part in many of the presets, and certainly a key component to the unique sound of Movement.

With up to 32 steps, the step sequencer is a great of creating patterns within Movement, with a handy set of pre-programmed sequences to get you started. To vary the effect of the step sequencer, Movement includes a variety of step shapes, alternating between 'gated' square shapes across the steps, for example, to a more typical 'spiked' AD-enveloped shape. While it might have been nice to see shape-editable on a per-step basis, it's still a great feature that tallies with the immediacy of Movement.

One prominent element of the GUI is the large X-Y controller, which is configured to add a level of 'performance control' to all of the presets. Indeed, it's well worth tweaking the X-Y controller as you move through the presets, as its effect can have quite a profound impact on the plug-in's output. Even better still, consider automating the control so that you can add a whole level of animation and progression to the sound that goes above and beyond what the rhythm sources offer.

Although some may be surprised by Output's move into the audio plug-in world, it makes a lot of sense. One consistent theme to all Output's products so far has been the sonic depths of the playback engine. Where some libraries offer a 'fait accompli' with respect to their sound content, Output's sample-based instruments used their content as just the beginning of what you could achieve. Indeed, with a range of filters, effects options, modulators, step sequencers and so on, instruments like REV, SIGNAL and EXHALE felt like creative tools in their own right, rather than pre-assembled collections of sounds and effects.

In keeping with Output's other offerings, therefore, Movement sounds fantastic, with a set of presets that will surprise and delight in equal measure, especially when you start manipulating the aforementioned X-Y pad. Having always wondered what SIGNAL would sound like with our own audio material passed through its audio engine, Movement does, to a large extent, answer that question. As you'd expect, pads and any ambient and textural sources gain an extra sense of dimensionality and animation, which can range from something subtle to more extreme sonic territories. What was a big surprise, however, was the transformative effect Movement had on 'vanilla' drum loops, highlighting the versatility of the plug-in.

Like a number of select software-based tools, Movement is the perfect remedy for those moments when inspiration starts to run dry; or indeed, an existing palette of sounds becomes too familiar. Movement is a tool that works best when a degree of serendipity is thrown into the equation – when you're not quite sure exactly what you want, but you're aware your music needs some form of rhythmic or textural lift. The result might be as simple as a small stutter effect, but it will ultimately provide the all-important 'ear candy' that your track needs.

Alternatives

Movement is a relatively unique product, although there are alternatives that explore similar areas, especially in respect to step sequencer-driven effects. Waves' MetaFilter ($149) is a good combination of analogue-style filtering controlled by a step sequencer, LFO and sidechaining – although, of course, you miss out on the other effects included in Movement. I've achieved similar modulation-style effects using Native Instruments' Guitar Rig 5 (£169), which has a modular-like signal path and all-important step sequencers and LFOs, but lacks the immediacy or focus of a tool such as Movement.

X-Y FX-engine control

You can assign the X-Y pad to any parameter by either right-clicking, or cmd-clicking on its control. The two axes of the pad correspond with the two FX engines, hence the cyan and orange colouring. By clicking on the small X-Y pad icon next to the control, you can set the relative knob range, moving between subtle and more extreme parameter changes.



Increased output

While it's hard not to be impressed by the ingenuity and creativity embedded into Movement, it does have a small number of caveats, which are worth considering. The modulation-routing system, though intuitive, doesn't necessarily facilitate the summing of two rhythms to a single destination – so that, for example, a fast-moving step-sequencer line is slowly modulated up and down using a slow-moving LFO. Equally, the FX options, though competent for the task in hand, could have greater scope – whether it's the option to change bitcrushing as well as distortion, or a greater amount of filter-pole settings, or some of the granular options included in SIGNAL.

Overall, Movement is an exciting development for Output, demonstrating that its unique approach isn't just restricted to sample-based instruments. Output seems to have a knack of understanding what piques a producer's interest, which it then manages to develop into a coherent musical product: whether it's an instrument focused on reversed sound, like REV, or in Movement's case, a plug-in that adds motion and interest to anything that passes through it. **MTF**

MTF Verdict

+ Transformative and imaginative effects
+ Intuitive GUI
+ Good range of presets
+ Instant inspiration

- Modulation routing could be more flexible
- Range of effects could be expanded

Demonstrating the same creative spark as its innovative instruments, Output's Movement is an inspiring and intriguing audio plug-in.

9/10



WAVE DNA
Liquid Music For Live

Choice 9/10 MusicTech

An amazing tool to help create harmonies, melodies, and chords, inside the Live Suite. **Martin Delaney** investigates

Details

Price **$149.99**
Contact **www.wavedna.com**
Minimum system requirements
MAC:
OS X 10.7.5
4GB RAM
Java version 1.6
750MB disk space
WINDOWS:
Windows 10, 8, 7, or Vista
4GB RAM
Java 1.8 (recommended Java SE8 by Oracle)
750MB disk space

Features

- Max For Live add on, AU and VST
- MIDI effect plug-in integrates with Live clips
- Sketch melodies, harmonies and chords in real time
- Create rhythmic ideas from beat-building blocks
- Use presets to create then manipulate fresh ideas
- Chord transposition and substition

Flashback to the NAMM Show in January 2016: during the last hour of the show, I discovered the WaveDNA booth, and Liquid Music For Live, which instantly became a personal show highlight. This is a Max For Live-based Ableton Live 'add on', which aims to provide a new take on MIDI composition – whether that involves beats, melodies, harmonies, or chords. The idea is as you draw in a succession of single notes, the software will generate chords or harmonies around them, based on either presets, or on your own customised settings.

This review covers the Max For Live version, although AU and VST versions are included in the purchase, for the Ableton-challenged person. After installation, Liquid Music appears as a Max For Live MIDI effect in Live's Browser. Drag it into a MIDI track, then load an instrument after it – it's best to use a straightforward polyphonic instrument like a piano at first. Create an empty MIDI clip in the track, then click the button in the device to launch and connect to the separate Liquid Music application. After that, all that's required is to select a preset and start drawing notes. As you do so, you'll see the notes appearing in the Live MIDI clip – complete with extra notes generated by whatever chords or harmonies were created by the preset chosen earlier. This is very impressive, and blew my mind the first time I saw it.

Traditional MIDI 'editing' functions aren't available within the plug-in, but you've already got Live if you want to drag MIDI notes around. in Liquid Music, it's about editing the preset settings, and applying those to the current notes, or drawing in some fresh ones... or clicking the 'Surprise Me!' button for a bit of randomisation. It's a mixture of game-like exploration, and as much music theory as you want to add.

> The **musically ignorant** can use it to **create compositions** they **never would've** otherwise

There's a great chord-sequencing feature, with a tool called the Suggester, which will suggest chords that naturally follow on from your initial selection, and you can also base chord selection on selected moods. Interestingly, a single instance of Liquid Music, loaded into any MIDI track, can access any MIDI clip, in any track, in your set. Another cool Ableton-specific feature is the ability to import a melody from an existing Live MIDI clip into Liquid Music. Full MIDI-control mapping is available, so you could use Liquid Music as an exciting performance tool as well. The musically ignorant (including me) can use the plug-in to create compositions they never would've managed otherwise – it's not the same as simply using Live's MIDI effect devices, for example – while the more experienced musician looking for a change will also be rewarded. I should mention this also includes the full version of Liquid Rhythm, which is of course the beat-oriented equivalent.

Liquid Music For Live is a unique product, and a must-have for anybody either struggling to compose their own MIDI parts, or seeking new composition and programming techniques. There were points where I was getting over-excited and thinking, "Wow, I'm gonna use this for all my music from now on!" – realistically, that probably isn't going to happen, but I am still very enthusiastic about Liquid Music as a music creation tool. MTF

MTF Verdict

+ Integrates with Ableton Live for immediate visual feedback
+ Max For Live, AU and VST versions
+ Possibly the ultimate cure for writers' block

- Not as clearly explained as it could be – documentation lacking
- For Live needs Max For Live

An authentic creation tool and at no point does it feel like you're 'cheating'. The 'For Live' version is particularly impressive.

9/10



Choice
9/10
MusicTech

SONIC ACADEMY
Kick 2

Sonic Academy's kick-drum instrument gets a makeover, but is it a stairway to drum heaven, or a one-way ticket to drum hell? **Alex Holmes** finds out...

Details

Price £49.95 (£24.95 upgrade from Kick 1)
Contact via website
Web www.sonicacademy.com
Minimum system requirements
VST, AU, AAX 32-/64-bit host
Windows 10, 8, 7, Vista (32 Bit), Intel Core Duo or AMD Athlon 64 X2, 512 MB Ram
Mac OS X 10.7 or later (32/64 Bit), Intel Core Duo, 512 MB RAM (no PPC)

Features

- Kick-drum instrument with Sub oscillator and three Click sample layers
- 230-plus genre presets
- 180 click samples
- Modelled Tube, Clip and Wave distortion
- EQ, Compressor, Limiter and Harmonics controls

It's fair to say that the kick drum is king in most modern dance music. As genres come and go, it tussles with the bassline for superiority, but in most club music, getting the right kick sound is so fundamental, it can make or break a track. We've been big fans of Sonic Academy's Kick plug-in over the last couple of years, and its punchy sounds have been used on a multitude of tracks as both kick drums and basslines. We knew Kick 2 was in development, but we're excited to finally get our hands on this new and improved drum synth/sampler instrument.

Click track

Kick 2 adds a whole slew of new features and changes, the first of which is the dropping of the Nicky Romero tag that came with the original plug-in. The interface is now larger and more polished, with splashes of colour and a waveform view that updates as you edit the various parameters. Most fundamentally, the Click area has been expanded from one to three sample layers, meaning you can create more complex textures and do more effective drum layering, choosing from either the extensive list of drum clicks, hats and noises, or loading your own with a new drag-and-drop functionality. Each Click section still has pitch and volume controls, but now adds dials for sample length, start point, pan, and phase alongside a LP/HP filter. This makes it much more flexible, as you can now load in something like a live kick sample and easily scoop out all the low end to allow space for the Sub part of the instrument. Our only gripe here is that the Pan control is so tiny that we almost missed it. All the Clicks, along with the Sub section, can be muted or solo'd, and have a button to toggle whether they respond to the keyboard, so you could have a pitched bass kick, with static transient click.

Sub focus

The main part of the plug-in is still the Sub oscillator, with a large window for drawing break points and bezier curves for the pitch and amplitude. This has been enhanced with an excellent harmonics section, that lets you add character to the waveform by dragging up eight sliders to bring out different parts of the spectrum. Other improvements include the addition of analogue-modelled Tube and Wave distortions, which are a big improvement on the (also included) Clip distortion from Kick 1. There's also a Compressor, a final output Limiter, and a Drive dial that controls how hard the signal hits the output. It's in the combination of these elements, the distortion, and the Harmonics section where the power of Kick 2 as a tone shaping tool really starts to shine. Different combinations of each now give different types of saturation and overtones, meaning you have a broader palette to play with. Other notable features include a drag-and-drop export function, and a new EQ section with a low shelf, high shelf and two notch filters for extra shaping of your overall tone. This is a nice addition, although a spectrum analyser would have been the icing on the cake to help with your tweaking. Last but not least, (as it was high on our wish list for Kick 1), the instrument is now monophonic with a new dial for Portamento. This makes it easier to have long tails without the worry of them overlapping and muddying up your low end. That said, without wanting to sound fussy, it would have been nice to be able to switch between voice modes.

Alternatives

The most obvious alternative is Big Kick from Plugin Boutique (£39.95), which doesn't quite have the depth of control of Kick 2, but has some excellent artist preset packs. If you want even more control, built-in FX and a mod matrix, then there's Vengeance Sounds' Metrum (£89), although it might be overkill for some.

New and improved

Kick 2 is a serious upgrade on the original plug-in, and is now more flexible than ever. This can be seen in the 230-plus genre presets that also include layered snares, claps and toms; a suggestion perhaps that this is much more than just a kick-drum instrument. There are also a few bass presets, although we would have liked to have seen more of them, given how surprisingly good Kick 2 is as a bass synth. Whatever the weather, this will no doubt be getting some serious use in our future tracks. MTF

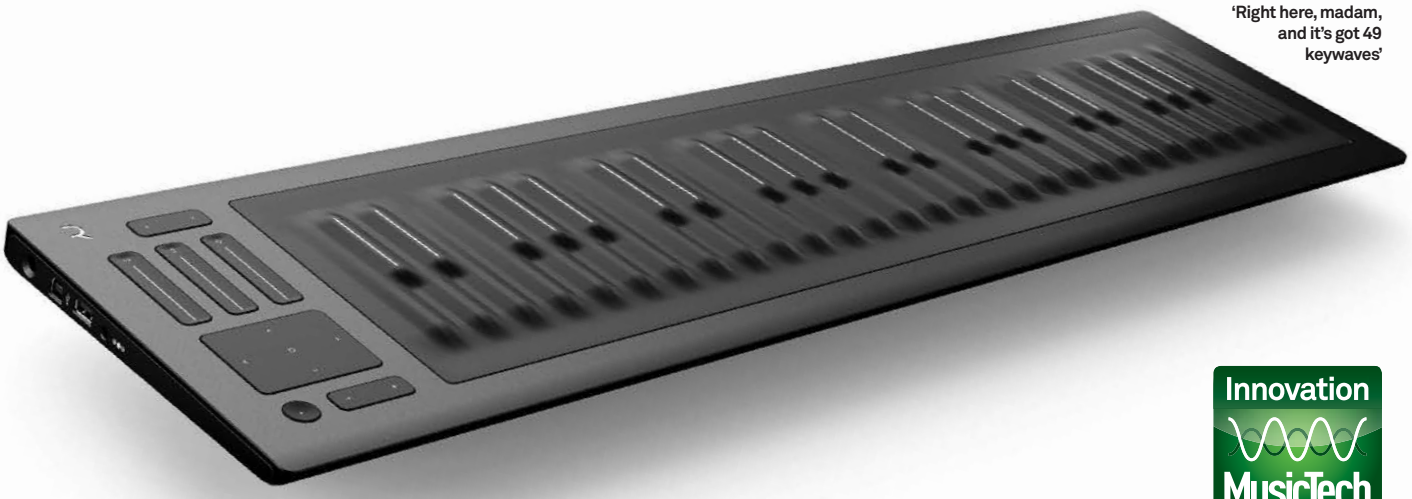
MTF Verdict

+ Highly flexible tool for sculpting kicks and hits
+ Solid collection of presets
+ Excellent distortion and harmonics section
+ Crisp, clear GUI

- Kick 1 user presets not compatible with Kick 2
- Pan controls incredibly small!
- No spectrum analyser for EQ

Kick 2 is a more nuanced instrument with a more refined palette of tools for fine-tuning your kicks and sounds, and it's an essential purchase for anyone writing music destined for the club.

9/10



'Ooh, where's me Seaboard?'
'Right here, madam, and it's got 49 keywaves'

Innovation MusicTech

Choice 9/10 MusicTech

ROLI

SEABOARD RISE 49

ROLI's quest for world domination continues with RISE 49. Bigger *and* better? ponders **Andy Jones**, as he re-enters the fifth dimension of sound…

Details

Price £949.99
T +44 (0)20 7254 2155
W www.roli.com
System requirements
Mac: OS X 10.8+
PC: Windows 7+ / iOS 7+, Intel Core i5 2.5GHz or faster, 4GB RAM (8GB RAM rec), 2GB available disk space for Equator, USB 2.0+ port for USB compatibility

Key Features

- Hardware-software integrated controller
- 49 keywaves
- Bundled with Equator, multi-dimensional soft synth and Bitwig Studio
- Completely wireless with MIDI over Bluetooth
- Connections: USB B port (MIDI out and power); USB A port (for charging peripherals); continuous pedal input (1/4" jack)
- Weight: 5.5kg
- Size (mm): 834 x 210 x 22.9

Here at *MusicTech* we love companies like ROLI, although there's only one company *really like* ROLI. It came from nowhere a few years ago, armed with a few million dollars in funding and a silicone keyboard that played differently from any other – but also the same, if you wanted it to – and could open up your sounds by exploring new dimensions of performance.

'Fly by nights' was the muttering from a few industry 'pundits', but ROLI has done its best to prove these neysayers wrong. Since the opening Seaboard GRAND keyboard, it's gone on to release a more consumer-level (but equally innovative) version, the RISE 25 (which we reviewed in the magazine last December, and which was the worthy recipient of both our *MT* Choice and Innovation awards).

It's backed this up with its multidimensional (and a lot better than it looks) Equator software; released an incredible free version of this called NOISE for the iPhone (which utilises the iPhone 6's 3D touch to great effect); and now, ROLI has released this – a 49-key (or, we should say, keywave) version of RISE. Oh, and also announced its consumer level music Blocks music makers. Fly-by-nights? Not really, it would seem…

A quick recap

With RISE, we have a keyboard that is silicone and a little squidgy, but one that otherwise follows a traditional 12-note-per-octave layout, so can be played similarly to a traditional 'board. Having said that, I'd recommend that full-on concert pianists try out RISE for size before committing. Even as a not-particularly-great player, I found that my piano style had to alter a little.

> For some **delicate touches**, RISE is **actually better** to play than **a traditional keyboard**

Sometimes, for delicate touches, RISE is actually better than a real piano and obviously the extra dimensions that I will come to are there to be explored, but equally obviously, it will be different for traditional players; so the best thing I can say to all players is, 'give it a go'.

So, to those extra dimensions and firstly, we get MPE (Multi-Dimensional Polyphonic Expression), the system that RISE uses which enables you to control sounds in extra ways, but still via MIDI. So you get velocity and aftertouch, yes, but you get them wrapped up in what ROLI describes as 'The Five Dimensions Of Touch' – which are Strike (the force of a finger hitting a key); Press (continuous pressure applied after a Strike); Glide (horizontal movements on a key or keywave); Slide (vertical movements); and, lastly, Lift (the speed of lift-off after a Press).

Within these five dimensions, the main ones that are perhaps different are Glide and Slide, which alter notes as you play up, down, left and right (with Slide, it typically mixes between sounds when used with Equator).

RISE has two modes: Expression Mode for playing the keyboard and controlling Glide, Slide and Press using the three Touch Faders to the left of the main keyboard. You also get the more standard MIDI Mode, which enables you to assign the three Touch Faders and the X-Y Touchpad that's also on the left of the keyboard to any MIDI CC.

Bigger, better, more?

So with the 49, we have, on the face of it, just a bigger version of RISE 25, but there's a bit more to it than that.

The first thing is immediately obvious as I unpack the 49 – a case! RISE 25 had a plastic-type case, which wasn't great and one of my criticisms of the product. Happily, there's an optional flip case available for the 25 now, and RISE 49 comes with a soft and well-padded case with a shoulder strap *as standard*. The keyboard is reassuringly heavy (in that you can tell it's well made), so sits well and solidly on your studio work surface, but is not exactly portable on its own – which is one of the reasons I love the 25-note version – but the addition of the case makes it far more luggable and is a very welcome inclusion.

One thing ROLI wasn't ever going to do anything about was another small criticism of mine: the dark design. It's cool, yes, it lights up just enough to see what is going on... but RISE is no Komplete Kontrol. It's elegant rather than garish, but as a live-performance keyboard for dingy club nights, it's still yet to prove itself as practical – black keys and low lights may not mix particularly well...

On a more positive note, setting up the 49 was a doddle. Logic picked it up straight away: no old-school driver updates or anything else needed. I plugged in and was playing the Equator synth in no time, and was quickly back to enjoying the extra dimensions that it has, more so in fact.

Of course, a 49-note keyboard will offer a more fullsome playing experience than any smaller version, but with the RISE 49 this is enhanced somehow. It may be the extra-long pitch-bends strip that runs along the bottom; but I suspect it's simply the extra sonic flexibility that the five dimensions of touch offer. Because you have the ability to change sounds in many more ways, these changes appear more dramatic with the wider range of octaves you have at your disposal. That string emerging from a pad sound at the top end of the scale is that much more dramatic at the bottom. Maybe it's just me, but MPE does seem to lend itself to a bigger, more playable keyboard.



EQUATOR ain't the prettiest – but it certainly shows RISE off in all of its sonic and expressive glory. More RISE features in NOISE, please!

A great engine, too

Credit for this success must, in part, also be given to the Equator software. It is the engine that shows off the expressive capability of the hardware. The software is a little unapproachable at first, but becomes more obvious and accomplished as you go through the presets and realise what does what. Simply play the keyboard – press, stroke and nudge – and you'll soon see which parameters alter on screen and get to grips with the features. And with three oscillators, sample based-synths, a noise engine and FM, there's plenty of flexibility to create different sound types.

All of these new dimensions of expression are recordable via your DAW with Dashboard software, which also lets you assign many of your MIDI controllers. Of course, in order to get the most from both hardware and software, you should take some time to master the extra performance areas to get the best from both. But you knew that.

Conclusion

RISE 49 might just be a bigger version but the extra length does emphasise the added expression that the RISE concept affords. The bottom line, then, is that bigger is definitely better. With a nicer case and a bigger range of keys, it may not be quite as compact and *bijou* as the 25, but RISE 49 is a fabulous concept which is made real by some great-sounding software. **MTF**

MTF Verdict

+ Great, innovative design
+ Good build, chunky and sturdy
+ Comes with a good case that makes it more portable
+ Equator is easily as good as we remember and is worth trying whether you have the RISE or not. Do try the free NOISE version!
+ Wireless control via Bluetooth
+ Playing takes you on some incredible sonic journeys
+ Play as a 'normal' keyboard if you like (but do try it pianists)
+ Not everyone will get it (physically or mentally)

- Still dark in the dark
- Proper players should try before buying (they may be surprised that some aspects are actually easier to play, though)

RISE 25 already had bags of extra expression and cool over a million standard controller keyboards, and the bigger version serves to drives these extras home. It is more playable and could be the RISE to go for if you're more of a performer than a sonic experimenter.

9/10

Alternatives

You now have three Seaboards to choose from: the RISE 49 on test here; the bigger 61-note Grand (which now costs £2,499, and the one we'd recommend for serious players); or the RISE 25 (compact, portable and a great entrance to the multi-dimensional sonic world that RISE offers). Of the traditional keyboards we've tested, Komplete Kontrol is still the one to go for, especially with the software bundle that now ships with it. Of the three, we'd probably opt for the 49 or 61-note versions.





Choice 9/10 MusicTech

Details

Kit **LYD 5 & LYD 8**
Manufacturer
Dynaudio
Price **LYD 5, £395 each (street); LYD 8 £555 each (street)**
Contact **Audio Distribution Group sales@audiodistributiongroup.com**
Web
www.dynaudio.com www.audiodistributiongroup.com

Features

Shared

- Bi-amp two-way active speakers
- Bass extension
- Position switch for two tunings
- Tweeter: 28mm soft dome
- Connections: XLR and RCA analogue
- Sensitivity: +6 dB: 50mV 0dB: 100mV -6dB: 200mV

LYD 5

- Woofer: 5" MSP
- Max SPL: 106dB
- Crossover freq: 5.2kHz
- Freq range: 50Hz to 22kHz
- Output power LF: 50w, HF 50W
- Power cons: 100W (<0.5W on standby)
- Dimensions: 170x260x211mm
- Weight(kg): 5.7

LYD 8

- Woofer: 8" MSP
- Max SPL: 112dB
- Crossover freq: 3.9kHz
- Freq range: 45Hz to 22kHz
- Output power LF: 80w, HF 50W
- Power cons: 100W (<0.5W on standby)
- Dimensions: 234 x369x328mm
- Weight (kg): 10.2

DYNAUDIO LYD 5 & 8

Dynaudio has an incredible rep for speaker design. **Jon Andrews** looks at two sets from the new LYD range. And yes, they are white…

We all know that music production hardware has become more exciting in terms of looks and popularity – Eurorack, colourful controllers, etc – but speakers? They have always been mostly black – the most outrageous departure being KRK yellow and the NS-10 white cone.

Dynaudio wanted to make a splash with the LYD range, so made it… white. Okay, it's not that radical and the black cones take so much of the front panel up that, along with black sides, you get used to it pretty quickly.

Whether you like white is, of course, a matter of personal choice. I rather like it. It accentuates the front design a little more than black would, with an angled curve around the woofer that lends the speakers a look of authority and perhaps a little futuristic appeal.

Small, medium, large

I've had two of the range on test for the last month: the LYD 5 and LYD 8. As has become normal, speakers get released in threes, with a clue to the woofer size in the name. So while each shares a 28mm soft-dome tweeter, we switch to imperial measurements with the 8- and 5-inch woofer'd versions here.

The LYD range is designed for small and medium studios and setting the speakers up is very straightforward. There's an excellent manual with a great design that gets you to the heart of speaker positioning very quickly. There are extras around the back of each that make initial tweaks and setups easy too, depending on your room size and audio setup. Sensitivity lets you match to your pro or consumer interfacing with -6dB and +6dB options for high and domestic output levels.

I'm a big fan of Standby mode. Yes, I do get irritated if it kicks in too early, but it's probably saved me money over the years and you get the choice to use it. Then you get to tweak the low ends by +/- 10Hz with handy Bass Extension options. Again, this is related to volume, so if you mix at low levels or have a smaller room, you might want to have this switched to -10Hz. But do experiment with these in your setups and stick with the setting you are happy with. Two final settings are perhaps less common: Position and Sound Balance. The former has two settings – Wall and

Free – to "help with anomalies created by reflections coming off the back wall". Basically, set it to Wall if your speakers are within 50cm of your back wall; otherwise, it's Free. How this works is not really explained, but I did try it, with hit-and-miss results.

The Sound Balance is a little harder to explain, and is dependent on your acoustic treatment. Dead rooms might require the Bright setting to give the speakers more zing, while one with lots of reflections might require Dark to help deaden the response. It's a useful setting to have, but again, will add to your setup time as you experiment.

It sounds like a lot of options to fiddle with, but you should always invest time at this stage of a monitor setup to get better results for your mixes down the line. Indeed, the manual lists quite a few suggestions on setting up and tweaking here – it's great advice, actually (we'll probably cut-and-paste it into an article soon!) Finally, the manual advises running the speakers in, something you'd expect to have to do with a lot of monitors, especially as you get above a certain price point.

Lifting the…

With so many combinations of setup options available, I decided to go down the middle as much as I could and experiment along the way; in effect, taking the advice from earlier. As the speakers were within 50cm of the back wall, I also engaged the Wall option with Standby mode on for good measure.

I set both the LYD 5s up; then the 8s for comparison; and then – and why not – one of each, for a direct comparison across a mix in stereo, which turned out to be the most revealing (FYI, I used an Allen & Heath Qu-24 mixer to feed from both Live and Logic).

Often when testing speakers you start with preconceptions based on reputation, price, specs etc, so you can then be surprised by units not living up to, or exceeding these expectations. This time, though, what I expected was pretty much the result, up to a point.

Like I said, the A/B comparison between the two monitors was the most revealing. As you might expect, the slightly larger woofer of the LYD 8 resulted in a better response further down the range. You might also expect the upper mids and top to be more or less the same, because the tweeters are identical. It's a simplistic way of looking at it, but direct comparison revealed just that. Both have an exceptional degree of clarity across these ranges with pinpoint detail on the soundstage – in fact, I couldn't fault them – with the 8s providing just that extra bit of boom down under. So the 8s delivered this bottom but, surprisingly, at lower levels: and on certain tracks I was mixing on I was getting a more rounded and tighter picture from the 5s – it just held together that much better.

I then decided to notch the bass response back on the 8s and bring it up on the 5s using the Bass Extension options, thinking I might be able to match them (to see if I could get an equal sound from the 5s and the 8s, but pay less for the 5s!). This didn't quite work – the 8s always 'won', bass-wise. Then I tried the Wall and Free options on both. It didn't make much difference on the 5s, but I did notice a slight muddying of the bass with the 8s when in the wrong mode. The manual explains little of the physics, bar 'anomalies at low frequencies', but it did alter the LF.

The 5s are suited to smaller rooms but both can be 'edited to fit' if you spend time with them



Around the back, you get a number of options not always available on other monitors, including Position and Sound Balance

Alternatives

There are literally hundreds of monitors available in this price range, so I'll just list some of the ones we've looked at recently, including Dynaudio's own BM Compact mkIII set-up and the MunroSonic Egg100 system. More recently, I also reviewed the RM series from Pioneer, very much the new kids on the block but great monitors, especially the RM-07.

Conclusion

So what does all of this prove? These are great-sounding speakers – I was very impressed with what I heard on mixes from similar priced and more expensive monitors and could make all sorts of improvements, and at lower levels. Yet the flexibility from all these extras makes the LYDs suitable for all sorts of studio situations, even though it takes time to set up. There's little to call between them. The 5s are obviously suited to smaller environments but both can be 'edited to fit'. Spend time sorting this and you'll be very happy with either. **MTF**

MTF Verdict LYD 5

+ Great and accurate response across mids and highs
+ Lows are tight and not as lacking as you might expect
+ Bass Extension and other controls add setup flexibility
+ Solid build
+ An overall tight sound

- Manual a little vague on how some of the 'extras' actually work
- Larger rooms will require the optional BM sub (or 8s)

Tight sound for small rooms and enough control for accurate results.

8/10

MTF Verdict LYD 8

+ Great response across the range
+ Good energy for mid size rooms
+ Again, lots of flexibility in set up
+ Very solid build
+ Good set-up info in the manual
+ Good detail for editing

- The 5s will be better at low levels
- Again, some extras a bit vague

It's hard to split these two, but the 8s deliver for larger rooms and have a great balance and accuracy.

9/10



IK MULTIMEDIA

Miroslav Philharmonik 2 CE

Miroslav Philharmonik 2 CE puts a complete orchestral instrument under the fingers of musicians on a budget. **Keith Gemmell** conducts his investigation…

Value €£$ MusicTech

Details

Price €119
Contact +44 (0)1223 234414
Web www.ikmultimedia.com

Features

- Over 10GB of 16-bit samples
- Over 700 instruments
- Multiple articulations per instrument
- Convolution reverb
- 34 high-quality studio effects
- Four mastering processors from T-RackS
- Standalone workstation, DAW plug-in, or add-on for SampleTank 3

Orchestral sample libraries, complete with all the sections and important solo instruments, cost a great deal of money to record and develop. That's one of the reasons why, once established, they tend to remain pretty much unchanged for many years. Every so often, they are updated in some way, usually to keep abreast with technical changes. Garritan Personal Orchestra 4, for example, was recently upgraded to GPO 5 with a large helping of extra samples from the defunct Garritan Orchestral Strings. And we've just heard that Steinberg's forthcoming notation program, Dorico, will feature the company's full HALion Symphonic Orchestra (another golden oldie) for its orchestral playback sounds.

Both the aforementioned libraries are mere youngsters, though, compared to the original Miroslav Philharmonik Orchestra, which first appeared over 20 years ago. The full, newly enhanced version with extra content recorded at CNSO Studios in Prague is now under the auspices of IK Multimedia. For those on a budget, IK has recently released a slimmed-down version, Miroslav Philharmonik 2 CE, based on SampleTank 3's 64-bit engine. With a 10GB library and over 700 instruments spread across the same categories as the full version, the word 'budget' doesn't really apply, content-wise.

That said, the 16-bit sample rate here is lower than the full version. The upside to this, of course, is that it is more CPU-friendly, making it suitable for all types of computer. It's supplied as a standalone workstation or a DAW plug-in. It can also be used as an add-on instrument collection for SampleTank 3.

> The original **Miroslav Philharmonik Orchestra** first appeared over **20 years ago**

Altogether now

All four sections of the orchestra are represented – strings, brass, woodwinds and percussion. There's also a range of choirs and four grand pianos.

The Play window

The Play window is where you select instruments and organise their articulations. For example, select a 14 Violin Multi and it loads with 20 articulations, which can then be played and key-switched chromatically from C0 to G1 on a controller keyboard. Selected articulations and their switches are displayed in a separate panel. For more in-depth work, articulations can be loaded separately. Legato Violins, for example, contains 11 legato choices. A set of instrument-specific macros is displayed just above the virtual keyboard.



Articulations are numerous throughout, which is ideal for experienced composers. Fortunately, for orchestral newbies, who are more likely to be attracted to this library, a range of playable ensembles is also provided for simple arranging.

As well as the sections, there are also plenty of solo instruments – again, with many articulations that can either be keyswitched on-the-fly in multis, or loaded as single samples.

The Mix window

The Mix window has 16 channels and four insert effects per channel, four stereo effect returns (with five insert effects each) and a master channel with five insert effects. Here, we have inserted a Vintage EQ to our 14 Violins Multi channel strip and a convolution reverb (ConvoRoom) to the Master channel. FX controls for both can be found just above the virtual keyboard.



percussion section, complete with maps. As if all this were not enough, to round it all off there's a huge selection of mixed ensembles spread across the keyboard for a fast workflow and useful sketching. Titles like Action Sequence and Brass Drama provide a good idea of what to expect here.

To top it off, 34 plug-ins from IK Multimedia's SampleTank 3 workstation can be inserted in the mixer, plus a convolution reverb and four mastering processors from their T-RackS. These can be controlled individually, or by using each instrument's eight macros which enable quick access to the most commonly used parameters.

Although articulations are listed on IK Multimedia's website, there's no in-depth information about editing. The Edit page itself contains a ton of controls and a manual would help enormously. Unfortunately, there doesn't appear to be one.

Has all you need for adding **orchestral music to tracks**, be they **pop**, **jazz** or **cinematic**

The sound quality throughout is, on the whole, excellent and pretty much what you might expect from a library of this price – good, but not spectacular. They're certainly on a par with the competition, though, such as the Garritan Personal Orchestra or the HALion Symphonic Orchestra. The strings, in particular, have a lighter sound than many other orchestral libraries with a nice woody timbre, which we found very refreshing.

Sing out

A very welcome addition are the choirs – female, male and mixed – that are as good, if not better than, other budget choirs we've encountered. So, too, are the tuned chromatic percussion instruments, marimba, glockenspiel and chimes. Other instruments of note include organ, guitars, harps and harpsichord and a full orchestral

Alternatives

Garritan Personal Orchestra 5 (GPO5) is a good choice and contains a complete orchestra with over 500 instrument choices, along with comprehensive control and a very easy-to-use ARIA player. The HALion Symphonic Orchestra has a wide range of orchestral instruments, including solo, tutti and ensemble sounds in various playing styles and articulations. HALion Sonic SE 2 is included in Cubase (version 7.5 or higher) and Cubase Artist (version 7.5 or higher) and Nuendo NEK (version 6.5 or higher).

The Edit window

The Edit window contains resampling, pitch and time-stretch, controls for fine-tuning selected samples along with filter and velocity controls. Below these are sections for two LFOs and two ADSR envelopes. To their right, we find a choice of polyphonic, mono and legato one- and two-key modes, with portamento and curve controls.



It's difficult to fault what is, after all, a remarkably comprehensive orchestral workstation that contains everything needed for adding orchestral music to existing tracks, be they pop, jazz, cinematic or whatever.

For pure classical-music projects and highly professional mock-ups, though, things have moved on technically in this area, with a variety of individual-instrument sample libraries on the market. They, of course, are far more expensive and the price paid in many cases for a single instrument will be higher than the complete Miroslav Philharmonik 2 CE, which also contains a very decent set of solo instruments, anyway. And if you'll pardon the pun, this is where IK Multimedia's offering scores very highly indeed – it's fantastic value for money. **MTF**

Do you really need this?

This library was the first of its kind back in the day and was developed by the bassist with Weather Report, Miroslav Vitouš, primarily for his own needs. It's been used by many media composers since. This version retains some of the original material, along with much that is new and is very computer-friendly. So if you are on the look out for a complete orchestra at a bargain price, the Miroslav Philharmonik 2 CE could be just what you need.

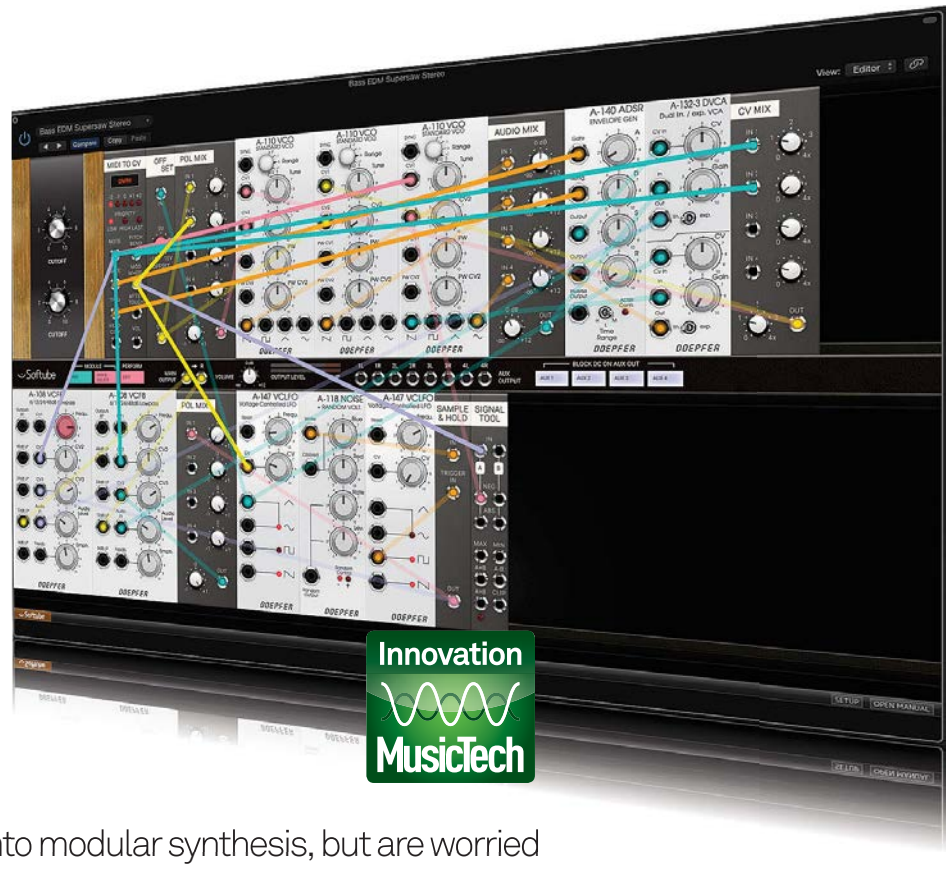
MTF Verdict

+ Long pedigree
+ Fully featured orchestrally
+ Solo strings
+ Fine sounds

- No manual

Value-wise, orchestral sample libraries don't come better than this – a full orchestra, plus solo strings, lots of ensembles and comprehensive mixing and editing features.

8/10



The multicoloured cable shop! Cables switch colour automatically, so it's easy to follow them from source to destination

SOFTUBE Modular

If you've ever wanted to get into modular synthesis, but are worried about spending the cash, this could be the solution for you. **Dave Gale** boots up the latest incarnation of modular in software format…

Details

Price **£69/$99**

Contact
MSL Professional
0207 118 0133

Web
www.softube.com

It has to be said that there is an element of commitment involved in getting into Eurorack and modular synthesis. There's the inevitable inherent price tag and then there's the bewildering array of modules available, so if your path towards modular glory is laced with trepidation, the new Modular virtual synth from Softube could well be the answer – at least in the short term.

Modular concept

Once installed, Modular appears seamlessly in your DAW, and springs open to reveal a blank case. "Where are all the modules?" I cry, but then, clicking the intuitive Add button reveals the available units with six basic, but good, modules from Doepfer, and a host of utility modules.

The ethos of Modular is to offer the user modules from the real world, and while there's a large collection of generic utility modules, many of which offer the sort of functionality expected in a Eurorack, there's also a helpful smattering of modules from existing companies, most notably at this stage Doepfer and Intellijel, with the latter requiring an additional small purchase.

Diving straight in with some of the presets purely to get a taste of what is on offer, I was immediately hit by the depth of the sonic palette. The bottom end is thunderous, with wonderful warmth in the lower mids, while the top end offers a wispy quality. It could be argued what I was hearing in the presets is akin to what I refer to as the 'Bose effect' – where the frequency palette is coloured by an overarching top and bottom end, that feels pleasing to the ear in its initial form. So now was the time to break open a pack of virtual patch cables and build up a timbre.

> I had to take off my '**soft synth**' **hat**, and very firmly put on my '**Eurorack boffin cloak**'

At this point, I discovered any illusions I may have had that this wasn't a traditional modular were about to come into sharp focus. Creating a sound, I had to take off my 'DAW-based soft synth' hat, and very firmly put on my 'Eurorack boffin cloak'. This synth will not make a sound unless you look after the basics; you'll need a MIDI-to-CV module to be in place if you want to trigger a sound from your keyboard, so having attended to this, I moved to an oscillator. The Doepfer A-110 VCO is one of the first modules on view, so this was my starting point. Once selected, the module appears in the modular case, and is ready for use, which essentially means patching it in – it's a modular synth, remember – so rather than waste time, in goes a filter, VCA and envelope. Patching together is a cinch, thanks to the elegant design of the interface. As you select an output, all the available inputs for connection glow green, as does the cable you are

Features

- A fully fledged modular synth in software form
- Basic selection of modules to work with
- Virtual 'Models' of existing units
- Expandable, with more modules arriving soon
- Supported plug-in formats: VST, VST3, AU and AAX Native

iLok, uLok

It's worth mentioning that authorisation of the Modular is made through the iLok system which, if you are not familiar, is a pretty well established anti-piracy and authorisation system. The ideal setup is to have an iLok USB key, which will hold iLok licences; but the said licence can also be written to your computer's hard drive instead, if a USB key is not present. This can be slightly more complicated if transferring a licence from one machine to another, which is why many prefer the iLok USB-key method, as the key can be easily moved. iLok is also used by many other companies (Eventide, Synthogy, Celemony, iZotope, Avid) as an anti-piracy system, so the likelihood is that you will use it for other software over time.





connecting. However, just like a modular, it is still possible to make CV connections that might not illicit a response at first, so unlike many soft synths, if you're creating a patch from scratch, you will need to engage the grey matter to ensure that connections make sense. Helpfully, cable colours alter with each one patched, so it's easy to trace the route of any cable.

And so the fun can begin, as it is even possible to build step sequencers into the frame as well, which will then, in turn, clock to your DAW, via the helpful DAW Sync module. However, special mention must go to the additional modules that are available at extra cost. There's a great choice of basic Doepfer modules available straight out of the box, which will more than allow you to get going, and Softube has implemented its own Heartbeat drum synth into the system – also at additional cost – which means you can turn your modular into a very beefy drum machine; but it has also teamed up with Intellijel to model the Korgasmatron II filter, Rubicon VCO and µFold modules.

Further models from other companies are in the pipeline, which ultimately means that the Modular is pretty future proof. In the short term, you might want to consider the Intellijel extras, because even though it's perfectly solid, the tonal colour of the Doepfer module set will inevitably become tired in use.

Above: The Module Selection menu – it's the modular equivalent of pick 'n' mix, and just as sweet to boot

Below: 16 notes of fun… the step sequencer, ready to sync to your DAW

Other companies have models in the pipeline, so the Modular is pretty future proof

Alternatives

If you're really into the concept of a modular synth on your computer, then there are a number of very fine options available in this arena. Arturia has modelled the Moog Modular and the ARP 2600, both of which are excellent clones and cost a fraction of the price of the vintage examples. If you own a Roland AIRA FX pedal, you can use Roland's new Modular Customizer, which will allow you to build a modular on a Mac/PC or iPad, and load it into the pedal for use. Similarly, if you have an iPad, the new Moog 15 App, from Moog, is outstanding – with all the touchy-feely elements that can be exploited on a touchscreen device.

Conclusion

I have always been a little sceptical of modular synths in software format, and this is quite possibly because I am so spoiled by using the real thing, but I have to say that I really enjoyed my time with the Modular. The overarching fact is, it sounds huge and certainly to my ears, there's some very realistic modelling of third-party modules.

So, are there any circumstances where it could actually take the place of my Eurorack? Only time will tell, and this is most likely to be dictated by the situation and how practical it is that I will have my Eurorack with me. The basic, slim collection of tonal modules that are currently available at this time is a minus, but with more modules on the way, Modular is a great investment if you want an inexpensive version of modular on the move. **MTF**

Do I really need this?

There is no doubt that if you want a modular on a budget, this is an excellent solution, and I cannot overstate how good the Modular sounds, but there are lots of great soft synths out there, and I think the question is more about whether you're the sort of electronic musician who likes to experiment. If you are, this will most likely open new horizons, but if you're after presets aplenty, this is probably not the best choice for you.

MTF Verdict

+ Sounds wonderful
+ Very flexible – like a modular!
+ Ergonomic interface
+ Basic, but good, module choice, right out of the box
+ Further modules can be purchased
+ Pretty future proof
+ Excellent value for money

- Will require some time and effort, unlike some other soft synths
- Slim collection of tonal modules at present
- Software did crash on me a few times, so further stability updates are likely
- No standalone version – plug-in only

The overall sound of this soft synth will excite many, but you'll need to put in some time to realise its full potential…

8/10



Details

Kit **K52, 72 and 92 headphones**
Manufacturer
AKG
Price
£39, £49, £59 +VAT
Contact
Sound Technology 01462 480000
Web **www.soundtech.co.uk**

Features

- Closed-back headphones
- 40mm driver
- 3.5mm jack with 6.3mm adaptor
- Weight: 200g

K52

- Frequency range: 18 Hz to 20 kHz
- i/p impedance: 32 ohms
- i/p power: 200mW
- Cable 2.5m

K72

- Frequency range: 16 Hz to 20 kHz
- i/p impedance: 32 ohms
- i/p power: 200mW
- Cable 3m

K92

- Frequency range: 16 Hz to 22 kHz
- i/p impedance: 32 ohms
- i/p power: 200mW
- Cable 3m

AKG K Series

Value €£$ MusicTech

Choice 9/10 MusicTech

AKG has released possibly its most affordable series of headphones ever. **Andy Jones** asks if they are too good to be true…

When it comes to a lot of music-production gear, we don't mind admitting that we can act a bit highbrow and snobby. Generally speaking, we tend to recommend expensive studio monitors, for example, simply because they tend to be the *best* ones. You pay a lot more money for an extra two-per-cent accuracy, but that accuracy is the most important two per cent you absolutely need when you are mixing. The same applies to headphones used for monitoring – of course it does. After all, it's no good spending a fortune on synths, software, outboard and mics, if you're mixing the end results on a cheap set of 'phones.

Which is a fine, and we think admirable, stance to take on the matter. But then, along comes AKG – a company that produces at least one set of gobsmackingly accurate headphones that cost well over four figures – and releases a range of headphones that cost, at most, 70 quid. This is a company that knows its stuff sonics-wise, so how dare it try and rock our snobby, highbrow 'throw money at your speakers' boat? How dare it indeed.

How dare they try and rock our 'throw a load of money at your speakers' boat?

0… K…

You can afford any model in the K Series. The cheapest, the K52, weighs in at just £39 plus VAT; the most expensive at just £59 plus VAT. For headphones aimed at the mixing market, that is extraordinary. I've been recommending headphones costing at least three times this amount and will be putting them up against such reference 'phones – so we could be in for some surprises.

The K range relies on two main design elements for its price-versus-quality selling point. The first is the 40mm drivers present in all three, which offer a wider frequency range and a higher sensivity, for a bigger output. The second feature is the closed-back design, which leans heavily towards comfort, as we shall see.

We'll start with the cheapest 52s and spec-wise, these have the shortest cable of the three (2.5m) and narrowest frequency range. All three 'phones are lightweight, which could be seen as 'cheap' – but actually, I'd be pointing out this as a positive on more expensive 'phones, so let's look at this as a bonus – you don't want something heavy on your head during long mix sessions.

The 52s are very comfortable indeed. The band keeps them tight – but not too clammy – on your head and the isolation is decent, though not spectacular. They don't pass my typing test – i.e. can I hear myself typing while wearing the 'phones? – but they do feel encompassing and just right.

Sound-wise, they are found wanting in the bass, but not as much as you'd expect. This is an area that cheaper headphones often struggle with and one thing manufacturers will do – as with speakers – is colour the sound here. Thankfully, AKG hasn't, but that does mean it's lacking. Having said that, I've heard far worse than these at exactly this price point and the sound does punch through... so they'd be a good set of cans for monitoring in a live scenario. Upper mids are there in abundance, lower mids not so much. However, the spatial separation *is* there and almost a match against our reference ATs; but the sound sadly isn't – it's not rounded, and a little too aggressive – but let's keep reminding ourselves that these are one sixth of the price. Yes, you read that right: *one sixth*!

Stepping up a level

So now we move on to the K72s and the difference is not immediately obvious. You get 'phones that are just as light in weight, and just as comfortable – identical, in fact. The isolation is also the same – decent if not perfect – and the wearing experience is good. They are comfortable, not so noticeable and good for longer sessions.

Sound-wise though, these are a step up. When A/B-ing with the 52s, you really can hear the presence of the extra couple of bass hertz, which helps make the sound less in-your-face. It's still up there and upper middy compared to our reference set, but you can start to pinpoint more things in your mix compared to the 52s and that's important in mixing. If you can identify a problem area – which you can just do with these – then you can start mixing.

Onto the final set, the K92s, and we're talking nearly double the price of the cheapest K52s – but still only a shade over 70 quid in total. While the only significant immediate difference with the 72s over the 52s was a silver circle on each earpiece, the K92s offer gold trim and a headband – not real gold, you understand, although the adaptor is gold plated. This obviously helps them stand out against the other models in the range, as do the specs.

The 92s have the widest quoted frequency range and this is apparent in my tests. While they are still not up there with my reference 'phones (now only three times the cost!), they're not bad at all. I could pick out more bass, more lower mids and separation is again good, as is comfort. Isolation is again reasonable – we're talking a very similar wearing experience to the others, really (same design and weight) – and as for the sound, you can certainly pick out what you need to in order to make a good stab at mixing. Again, this mix is a bit shouty compared to the more rounded performance from my reference set, but it is certainly a lot calmer and more consolidated than both the 52s and 72s and really pretty good for ID-ing mix issues.

For a laugh – as you do – I put the 92s up **against headphones that cost £1,300**

For a bit of a laugh – as you do – I put the 92s up against a set of AKG K812s that I have finally got in for review (long story) just to see how they'd cope. These AKG s are £1,300-plus, so, yes, they sound incredible. A beautiful, open mix. However... the K92s didn't sound like just 1/18 of the price – you could hear much of the detail, albeit in a more compressed environment, like the aural equivalent of the roof going on at Wimbledon (okay, maybe not).

Conclusion

Of the three, we definitely recommend the K92s. They have the best specs and are a step up sound-wise, with the wearing experience of all three being otherwise similar. Indeed, they do challenge, but not quite beat, our 'spend more money' on speakers philosophy, and could be the best 'phones you can get for two figures. **MTF**





Alternatives

We haven't looked at many sets of cheaper headphones of late, so these stand alone in the low end of the market. We still – in the main, anyway – stand by our 'spend more' philosophy and on that score, some of our standards include recent models by Ultrasone (P880); Audio Technica (ATH R or M70), KRK (KNS8400) and Blue (Mo-Fi).

MTF Verdict K52

+ Lightweight phones
+ Comfortable
+ Dynamic sound with good separation
+ Reasonable isolation for the money
+ Stupidly cheap

- Bass is lacking
- Mix is a little aggressive

With a big, brash sound, these are more suited for live monitoring than studio mixing. Okay if you lack funds, but we'd recommend spending more for serious work.

7/10

MTF Verdict K72

+ Lightweight phones
+ Very comfortable and good for longer sessions
+ Vibe-y sound with good separation
+ Reasonable isolation and, again, very cheap

- In-your-face sound can be tiring
- Not quite the level of mix you are looking for

While these are a step up from the 52s sound-wise, we can't help thinking that an extra tenner will buy you the top-end model!

8/10

MTF Verdict K92

+ Most accurate sound of the range
+ Good imaging
+ Good comfort
+ Most expensive, but still cheap
+ Good detail for mixing
+ Better bass response

- Mix still a little more enclosed
- As with the others, the upper mids are pronounced

The best in the K range and well worth the extra £ for proper mixing.

9/10



Choice 9/10 MusicTech

Value €£$ MusicTech

UNIVERSAL AUDIO

Fender '55 Tweed Deluxe

Universal Audio finally applies its studio-grade modelling expertise to a classic amplifier. **Mark Cousins** investigates the Fender '55 Tweed Deluxe emulation …

Details

Manufacturer
Universal Audio
Price **£149**
Web **www.uaudio.com**

For several years now, the UAD platform has included a range of excellent amplifier simulators from the likes of Brainworx and Softube, but as yet, none have been developed by Universal Audio itself. With the release of the Fender '55 Tweed Deluxe, though, we finally get to see Universal Audio's expertise applied to a guitar amp.

Released in 1955, the classic 5E3 Fender Deluxe amplifier is an intriguing device, known for its unique and somewhat colourful response to the dynamics of the input. Compared to the significantly cleaner Fender Twin that followed a few years later, the Deluxe amplifier has a unique range of tones - from a soft and subtle tube saturation when the amp is driven lightly, to a raucous, overdriven tone when the amp was driven hard. In short, it was an amplifier that responded to your playing, and despite a relatively simple set of controls, could produce a range of different timbres.

> Some **amp simulators** offer a **'bells and whistles' approach**, but this one is **all about tone**

Key Features

- End-to-end emulation of the entire 5E3 tweed Deluxe circuit
- Unison technology allows for authentic guitar interaction with Apollo Interfaces
- Best-in-class speaker-cabinet emulation
- UA's first complete, stem-to-stern guitar amplifier model

Full tweed jacket

As you'd expect, the Fender '55 Tweed Deluxe is a true end-to-end example of modelling: including the amplifier head, a number of different speaker cabinets, and a collection of studio-grade microphones used to record it. While some amplifier simulators offer a 'bells and whistles' approach, the Fender '55 Tweed Deluxe is all about tone – whether it's switching between the Inst and Mic inputs, swapping speaker cabinets, or moving microphone positions. Ultimately, it's these key parameters – rather than a host of FX pedals – that deliver a truly great guitar sound.

Whether you're a guitar player, or simply a producer using the plug-in as a re-amping tool, what becomes immediately apparent is just how dynamic and responsive the '55 Tweed Deluxe feels. Take the Tone control, for example. Being more than just a rudimentary filter, the Tone actually doubles as another form of gain control (in addition to the Instrument Volume and Mic Volume controls), making the amplifier increasingly overdriven as you push the control harder. Like plenty of examples of classic vintage gear from yesteryear, therefore, an apparently simple control setup conceals a complex set of sonic interactions.

In unison

Another pleasingly studio-friendly feature is the Mic Setup section, which includes various classic Dynamic, Ribbon and Condenser microphones. What's particularly useful is the ability to use two mics simultaneously, either to mix and match different timbres, or to create a pseudo stereo effect by panning two mics left and right respectively. With the addition of an off-axis positioning control, perfect phase alignment, and a bass roll-off control, if felt as close as possible to walking through to the studio area and mic'ing the amp yourself.

One last modelling detail worth picking up on would be the pairing of the Fender '55 Tweed Deluxe plug-in and a Unison-powered Apollo interface. In this case, the plug-in actively changes both the impedance switching and gain staging, based on which of the virtual inputs you select, injecting another further layer of realism into its sonic behaviour.



Alternatives

Other amplifier simulators include versions of the tweed amplifier, including Native Instruments' Guitar Rig 5 (£169), and IK Multimedia's AmpliTube Fender (€163). However, both solutions provide a wide range of amps beyond the basic tweed amp setting, so the modelling isn't as detailed as what's provided on the UAD version.

Amp it up

Without any doubt, the Fender '55 Tweed Deluxe is arguably the most authentic and sonically detailed emulation we've seen of any guitar amplifier to date, complete with a level of dynamism and musicality to match. As you'd expect, there's a massive hit on DSP resources, so that even on an OCTO-powered UAD device, you'll struggle to run more than eight instances. Ultimately, you quickly realise why the plug-in is best run on the front end of an Apollo interface, where you simply 'record through' the plug-in, rather than run multiple instances. For non-Apollo users, though, there's always the option of bouncing/freezing the track, if you want to claw back some DSP resources.

> Arugably the **most authentic** and **sonically detailed** emulation we've seen of **any guitar amp**

As always, the best plug-ins – especially those that use distortion – come at a premium, but given the sheer amount of musicality you get back from the Fender '55 Tweed Deluxe, it will be an investment that reaps dividends. It's certainly an excellent and logical choice for Universal Audio's modelling expertise, and may indicate a string of products (a Fender Twin would make us very happy!) that could follow in future software updates. MTF

MTF Verdict

+ Superbly authentic sound
+ Dynamic response
+ Concentration on 'tone'
+ Flexible mic controls

- Big hit on UAD DSP resources
- Makes other tweed models sound inferior!

The Fender '55 Tweed Deluxe illustrates just how good amplifier modelling can sound, capturing all the unique sonic traits of a truly classic vintage amp.

9/10

Reel People Present Soul Vocals

Publisher **Loopmasters**
Price **£24.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Key Features

- 14 song construction kits
- 80 to 127BPM
- Male and female vocals
- Ad-libs, leads and layered harmonies
- Recorded at the Reel People studio

Reel People Present Soul Vocals is a new pack of 12 vocal-construction kits in 24-bit WAV and Apple Loops formats, for use in R'n'B, deep house and beyond. Each kit contains several male and female harmonies and ad-libs, plus a comped file with all the main layers combined. All of the harmonies are double-tracked to provide a richer sound, with each take hard panned so that they can be separated, if you so desire. The ad-libs are in mono, but there are several versions to offer up variation. Lyrics-wise, it's not going to win any awards, but the phrases are uplifting without sounding too cheese-heavy. With that in mind, some will prefer to use the pack as a source of layered, wordless harmonies and ad-lib chops. The tempos range from 84 to 127BPM, meaning the parts could be suitable for many genres, from hip-hop or drum 'n' bass to house styles. MTF

MTF Verdict

A well performed and thought-out pack that features some great male and female harmonic layers and a decent selection of soulful ad-libs.

8/10

Double Bass Mania VIII: Pure Modern Metal

Publisher **Beta Monkey Music**
Price **$39.99**
Contact **lost@betamonkeymusic.com**
Web **www.betamonkeymusic.com**



Key Features

- Modern, thrash and death-metal grooves
- 1.22GB of 24-bit audio
- 600+ Loops at 80 to 110BPM
- Acidized WAV, Apple Loops and REX formats
- 256MB of multi-velocity drum hits

Choice 9/10 MusicTech

If you're after some intense metal drum grooves to add power and excitement to your tracks, then you might want to check out Double Bass Mania VII from Beta Monkey Music. Recorded and mixed by one of metal's leading engineers, the library contains over 1GB of deep and punchy toms, fierce snares and fast but tight kick-drum rolls. Tempos range from 80 to 110BPM, with inspiration drawn from death and thrash alongside more modern metal styles. All loops come from the same session for easy drag-and-drop arrangement, and you also get 256MB of individual multi-velocity hits taken from the same session. The playing really is world class, with some blistering fills and high-octane beats delivering straight-up metal grooves alongside more complex syncopated patterns and alternative time signatures. MTF

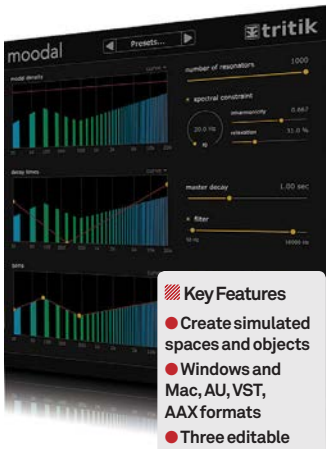
MTF Verdict

A large collection of varied, fierce and brutal metal drum grooves and fills, with a tight and punchy performance and production finish.

9/10

Moodal

Publisher **Tritik**
Price **€75**
Contact **via website**
Web **www.tritik.com**



Key Features
- Create simulated spaces and objects
- Windows and Mac, AU, VST, AAX formats
- Three editable curves for density, decay and gain
- Scalable GUI with two themes
- Up to 1,000 resonant filters

Moodal is an effect plug-in with a unique resonance engine that allows you to simulate reverb spaces such as rooms and halls, or objects like bottles and metal plates. The scalable user interface features three editable curves for Modal Density, Decay Times and Gains that allow you to control different parts of the spectrum – for example, creating a dense high-frequency cluster, with the low end decaying quickly and the high end ringing out for longer. You can create bell-like sounds by using less resonant filters, or have up to 1,000 (although this will use more CPU). You can also play with the harmonicity and the way the spectrum is distributed, add a LP/HP filter, and control the width. This is a decent plug-in for sound design, although it would be more flexible and interesting if there was an LFO to modulate the spectral parameters. **MTF**

MTF Verdict

A unique plug-in for creating simulated reverb and object effects that could be of interest to those looking for new sound-design tools.

8/10

Analogue Techno

Publisher **UNDRGRND Sounds**
Price **£34.95**
Contact **info@undrgrndsounds.com**
Web **www.undrgrndsounds.com**



Key Features
- Ambient and driving techno at 123BPM
- 1.16GB worth of 24-bit audio
- 678 loops and 387 one-shots
- Five Custom Kits for Ableton, Battery 4, Kong and Maschine 2+
- Produced and processed with analogue kit

Excellence 10/10 MusicTech

Analogue Techno is the first release from UNDRGRND Sounds, a new label which focuses on quality underground dance-music samples. The pack includes a whopping 1.16 GB of 24-bit audio, ranging from deep and ambient, to more driving and heavyweight forms of techno. There are 10 folders in total, packed with beefy, pulsating bass loops, punchy, glitch-laden drum and percussion grooves, and a mixture of dark and dissonant, and deep and beautiful pads, synth lines and textures. There are also some complex sound-designed FX and transitions, plus a large collection of processed drum hits. The production throughout is excellent, with movement, detail and precision in every single loop and hit. Anyone interested in deep, cutting-edge techno and electronica should definitely take a look. **MTF**

MTF Verdict

A large, high-quality pack of exquisitely produced techno nuggets, with plenty of inspiring sounds and grooves to explore.

10/10

Stefano Ritteri House Experiments

Publisher **Loopmasters**
Price **£29.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Key Features
- 728MB worth of 24-bit audio
- 139 loops and 237 one-shots
- 49 patches for EXS24, Kontakt, HALion, NN-XT, Kong and SFZ
- 120BPM
- Written and produced by Stefano Ritteri

Italian house maestro Stefani Ritteri has teamed up with Loopmasters to create a new pack of infectious house loops. There are a total of 139 loops and 237 one-shots that cover thumping basslines, retro-tinged drum loops, Latin-inspired percussion grooves, and squelchy leads and stabs. A range of analogue synths was used, including a Korg MS-20, Arturia MiniBrute, and Moog Sub 37, plus some classic drum machines. Although the riffs are relatively simple, the analogue sound shines through, and Ritteri lets repetitive loops run for multiple bars with subtle internal movement. The one-shot drums, FX, multi-sampled bass and synth sounds are tight, punchy and usable, but nothing out of the ordinary. In fact, for a pack called House Experiments, we found it a little tame and predictable. However, these are solid, satisfying loops and sounds and would work well as the backbone to a tech-house track. **MTF**

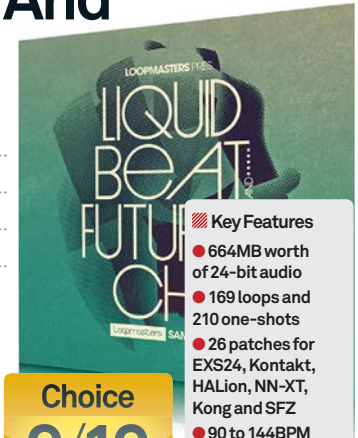
MTF Verdict

A decent collection of simple but effective analogue bass and synth loops, and tight, bouncing drum grooves.

8/10

Liquid Beat And Future Chill

Publisher **Loopmasters**
Price **£24.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Key Features
- 664MB worth of 24-bit audio
- 169 loops and 210 one-shots
- 26 patches for EXS24, Kontakt, HALion, NN-XT, Kong and SFZ
- 90 to 144BPM
- Mixture of live and synthetic elements

Choice 9/10 MusicTech

Liquid Beat and Future Chill is a new pack from Loopmasters, filled with soothing melodies, laid-back basslines and wonky beats. The pack weighs in at 664MB in your choice of Wav, Apple Loops, Live Pack and ReFill formats, and contains 169 loops and 210 one-shots. Everything here has an organic edge, with live Rhodes and piano riffs sitting alongside reverb-drenched arpeggio synth runs, and quirky beats constructed from chopped-up breaks. The original programming style and use of processing gives everything a fresh feel, and the looseness adds a human touch to every loop. There's also a small but characterful selection of drum and FX hits, and 10 reasonable multi-sampled bass and synth sounds. If you're looking for something a little different to kick start some ideas, then this could be a good place to start. **MTF**

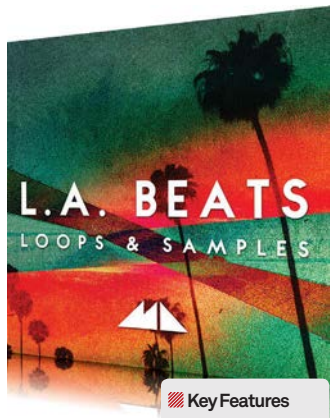
MTF Verdict

Quirky and loose production provide a fresh-sounding and inspiring collection of song starting riffs and beats.

9/10

L.A. Beats

Publisher **Mode Audio**
Price **£18**
Contact **info@modeaudio.com**
Web **www.modeaudio.com**



Key Features
- 613MB of 24-bit audio
- Hip-hop and trap inspired samples
- 156 drum, bass, synth, keys, guitar and FX loops
- 139 MIDI loop files
- 88 chunky drum, percussion and FX loops

L.A. Beats is a crunchy new collection from Mode Audio that features resonant analogue-synth riffs and skewed, chunky beats from 70 to 86BPM. The bulk of the pack includes 156 loops covering drums, hats, percussion and fills, synth bass, glistening arps and leads, jazzy organ and keys chords, guitar licks, and vinyl noise beds. It's an eclectic mix of hip-hop and trap sounds, with an energetic and loose feel. In addition, there's a folder of tail samples that can be used with their corresponding loops, or as samples in their own right. There are also 139 MIDI files that offer a great way to use some of that live feel with your own instrument sounds, and a decent folder of 88 beefy, saturated drum, percussion and FX hits. **MTF**

MTF Verdict

Despite some occasionally heavy-handed production finish, this is a versatile and characterful collection of loose-sounding loops and hits.

8/10

Producer's Guide To Vocals

Publisher **Producertech**
Price **£29.95**
Contact **via website**
Web **www.producertech.com**



Key Features
- Over one-and-half hours of video
- 11 modules
- Written and presented by Ian Bland
- Vocal writing, recording and production tips
- Uses Logic X and Ableton Live

Veteran producer Ian Bland, aka Blandy, has teamed up with Producertech for an 11-part course on working with vocals in a house track. The chapters start with a look at songwriting, and finding and recording a vocalist, before moving on to discuss compression, EQ, reverb, delay, chorus and special effects, panning and balancing, and finally vocal chops. Bland looks at native effects in both Live and Logic X, and plug-ins by Slate Digital, UAD, and Native Instruments. If you've already got experience working with vocals and are looking for in-depth tips on different mix techniques and mixing settings, then you may wish to look elsewhere. However, if the whole process of working with vocals is a new area for you, then this is a good, all-round introduction to get you started. **MTF**

MTF Verdict

Although a little technically light in places, this is a good introduction to working with singers, recording and producing vocals, with some creative ideas to get you thinking.

8/10

Drum Werks Groove Rock

Publisher **Beta Monkey Music**
Price **$29.99**
Contact **lost@betamonkeymusic.com**
Web **www.betamonkeymusic.com**



Key Features
- Hard, classic, alt- and pop-rock drum grooves
- 896MB of 24-bit audio
- 211 grooves and 130 fills at 90 to 120BPM
- Acidized WAV and Apple Loops formats; REX available soon
- 200MB of multi-velocity drum hits

Choice 9/10 MusicTech

Drum Werks XVII is a new drum-groove pack in Beta Monkey's 'L.A. Three' series that focuses on hard rock, classic rock, and alt- and pop-rock styles. Every loop has been expertly performed and recorded with the same settings, making it easy to mix-and-match grooves, even with other packs from the same series. You'll find a range of beats and fills across four folders, with tempos ranging from 90 to 120BPM, and each fill has been named to indicate which drums were used. So although you only get one drum sound, there is a vast amount of subtle pattern variety in the large number of grooves on offer. You also get a folder of multi-velocity drum and cymbal hits taken from the same session, but you'll have to load these into your sampler of choice to get the most out of them. **MTF**

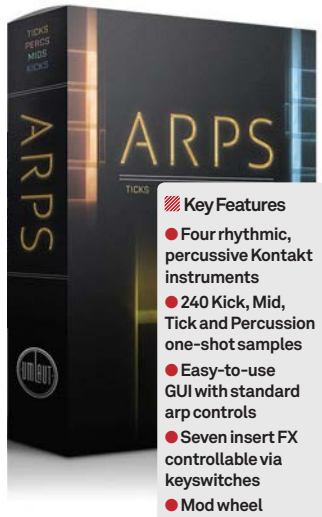
MTF Verdict

Uncluttered, driving rock grooves with a punchy sound, which would work well as the backbone to a song.

9/10

Arps

Publisher **Umlaut**
Price **$99 (works with Kontakt 5.5.1 or higher, or the free Kontakt Player)**
Contact **via website**
Web **www.umlautaudio.com**



Key Features
- Four rhythmic, percussive Kontakt instruments
- 240 Kick, Mid, Tick and Percussion one-shot samples
- Easy-to-use GUI with standard arp controls
- Seven insert FX controllable via keyswitches
- Mod wheel controls reverse samples

Arps is an intuitive percussive instrument for Kontakt or the Free Kontakt Player that creates rhythmic underscores or driving grooves for any composition. The sounds are loosely divided across four .nki instruments as Kicks, Mids, Ticks, and Percussion hits using a wide variety of acoustic and electronic samples. The GUI's main section has standard controls for Kontakt's arpeggiator and a large window for drawing in note velocities. A nice touch is the addition of seven insert FX, which can be performed using keyswitches, and the mod wheel can introduce reversed samples to the pattern. The streamlined layout makes it easy to get quick results, but there are also a handful of snapshots for each instrument to get you started. It seems like a missed opportunity not to have any more complex presets that use combinations of all four instruments, though. **MTF**

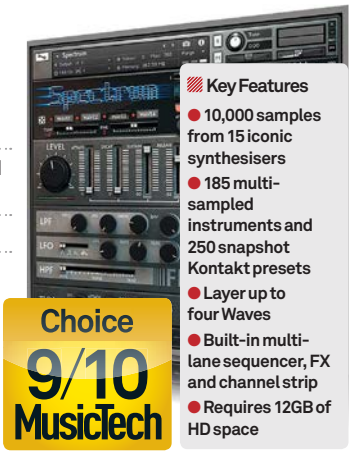
MTF Verdict

A simple but effective scoring tool for adding cinematic percussive and rhythmic patterns to your tracks.

8/10

Spectrum

Publisher **Wave Alchemy**
Price **£59.95 (requires Kontakt 5.5.1 or above)**
Contact **via website**
Web **www.wavealchemy.co.uk www.loopmasters.com/labels/4-Wave-Alchemy**



Choice 9/10 MusicTech

Key Features
- 10,000 samples from 15 iconic synthesisers
- 185 multi-sampled instruments and 250 snapshot Kontakt presets
- Layer up to four Waves
- Built-in multi-lane sequencer, FX and channel strip
- Requires 12GB of HD space

Wave Alchemy has spent over 18 months producing its new release, Spectrum. The Kontakt instrument includes 185 characterful, multi-sampled waveforms that can be layered into four slots, with controls for amp and filter envelopes, a choice of four filters, a gentle HPF, unison and width controls, LFOs for the filter and the pitch and a useful randomise wave function. There's also an excellent in-depth Sequencer page for modulating a range of parameters, a chord memoriser function, and a fairly simple, but well-thought-out FX page. The waveforms include more raw waves alongside some satisfying, well-programmed analogue shapes that can be stacked up to build massive-sounding patches. Just be aware that some patches can take up a fairly hefty amount of RAM. **MTF**

MTF Verdict

A fantastic collection of unique and lively multi-sampled synth waves that can be layered in near endless ways to create modern and retro synth gold.

9/10

How to Create A Killer Live Set
Using Ableton Live & External Instruments

MusicGurus



Publisher **MusicGurus**
Price **£25**
Contact **via website**
Web **www.musicgurus.com**

Key Features
- Learn techniques for setting up an Ableton Live set
- 16 videos streamed online
- 2 hours and 16 minutes
- Uses Live plus an array of hardware
- Written and presented by Mark Burnett

Although there are no end of tutorials showing how to program and mix using Ableton Live, there aren't so many on the art of crafting and delivering a dynamic, pro live set. This new course from MusicGurus sees producer and Ableton Certified Trainer Mark Burnett work through his setup that includes Ableton Push, an Elektron RYTM, Korg volca beats, Vestax VCM 600 mixing desk and more. The 16 lessons start simply with some clear explanations on how to cable and set up the equipment and software, before moving on to more complex explanations on MIDI mapping, racks, macros, using microphones and loopers, and performance FX, before finally showing some sections of performance. Although things get specific to Burnett's own methods, this is an interesting insight. Use MUSICTECH20 to get a discount. **MTF**

MTF Verdict

Starts out simply but quickly develops into an in-depth look at a specific, pro live setup with useful tips and insights.

8/10

Culprate Presents Sampling Reality

Publisher **Loopmasters**
Price **£24.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Key Features
- 912MB in 24-bit Wav, Apple Loops, Ableton Live or ReFill formats
- 191 Audio, 195 REX2, and 23 MIDI loops, plus 484 hits
- 79 patches for Kontakt, EXS24, HALion, NN-XT, SFZ
- Produced by Culprate
- Parts taken from *Nightmares In Reality* EP

This new pack from producer Culprate fuses dubstep with 2-step garage, d'n'b and breaks influences across 912MB, with 191 loops, 484 single hits and 79 sampler patches. These loops cover some aggressive and powerful synth basses, a variety of industrial-strength beats, top loops and percussion grooves, plus some surprisingly deep-sounding chords and musical loops. Many of the thicker sounds are provided with individual layers, which cuts down on the number of unique riffs, but offers an insight into how Culprate sculpts his huge sound. There's also a decent collection of powerful bass, synth, drums, vox and FX one-shots and multi-samples. The only downer is the fact that the parts have been dissected from Culprate's previously released *Nightmares In Reality* EP, so you'll need to work extra hard to make some of the sounds and loops your own. **MTF**

MTF Verdict

A well-produced library with crisp, tight and powerful production throughout, which offers a few surprises beyond your average dubstep pack.

8/10

Defected Old School House Rogue D

Publisher **Loopmasters**
Price **£19.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Key Features
- House drum, bass and chord loops by Rogue D
- 225MB worth of 24-bit audio WAV, Apple Loops & REX2
- 122BPM
- 15 drum hits with one patch for Kontakt, EXS24, HALion, NN-XT, SFZ
- Rough and warm old-school sound

If you love the sound of warm, fat and saturated old-school house, then you might be interested in this new pack from Defected Records and veteran producer Rogue D. The collection features a relatively modest 225MB worth of 24-bit audio, with 101 drum, top and fill loops, 20 bass loops and 44 chord loops, plus a small collection of 15 drum machine hits. It's not a massive pack, and it won't win any awards for originality, but everything here – from the chunky, kick heavy drums to the warm and creamy filtered down basses and dusty chord chops – has an appealing rough and analogue-sounding edge to it. Our only gripe is that we found the layout of the drums and their component loops a little confusing, and would have liked some more kick-only parts to aid with arrangement. **MTF**

MTF Verdict

A small, simple but effective pack of warm and chunky house beats, basses and chord stabs.

8/10

KV Bala Krishnan Table Sessions



Publisher **Loopmasters**
Price **£24.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**

Key Features
- Indian percussion loops
- 1.18GB worth of 24-bit audio
- 478 WAV or Apple Loops files, plus 496 Rex2 files
- Tabla, dholak, salangai, finger cymbals and more
- Performed by world-renowned percussionist KV Bala Krishnan

Choice 9/10 MusicTech

Following on from the success of four Indian Sessions packs, Loopmasters once again team up with world renowned percussionist KV Bala Krishnan for a large collection of tabla and assorted percussion instrument loops. The pack includes 478 loops spread across over 80 key-labelled grooves, with tempos ranging from 80 to 175BPM. You'll find some great, tightly played tabla and dholak patterns alongside much simpler finger cymbal, tambourine, cabas, and salangai hits, plus ensemble files with the single loops combined. There's good use of stereo and the sound is crisp and well recorded, but the dynamics are left wide open, so you can process how you see fit. Aside from being useful for film and TV composers, the nature of the instrument also offers up some great sounds for tech-house basses or d'n'b shuffles. MTF

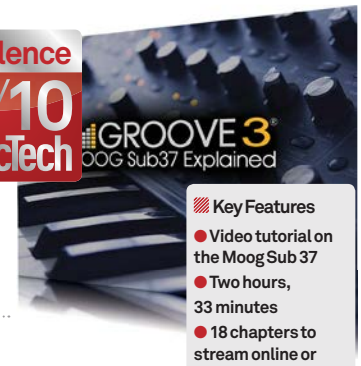
MTF Verdict

Tight-sounding Indian percussion in a wide variety of tempos that could prove a useful tool for media composers and dance-music producers alike.

9/10

Moog Sub 37 Explained

Excellence 10/10 MusicTech



Publisher **Groove 3**
Price **$30 (download), $15 (one month access to entire site)**
Contact **support@groove3.com**
Web **www.groove3.com**

Key Features
- Video tutorial on the Moog Sub 37
- Two hours, 33 minutes
- 18 chapters to stream online or download
- Includes tips for patch design
- Written and presented by Sami Rabia

Moog's Sub 37 is the perfect blend of analogue electronics and digital control. This two-and-a-half hour video tutorial from Groove 3, sees sound-design guru Sami Rabia exploring every facet of the instrument, starting with preset management and the Global menu, and moving on to the Oscillators, Filters, Mod Bus, Arpeggiator, Sequencer, software editor and more. The pace across the 18 videos is fairly quick, but easy to follow, with plenty of useful sound design tips spread throughout. In fact, there are four chapters dedicated to designing bass, lead, FX and pluck sounds at the end of the tutorial that help showcase the power of the synth. In terms of all-round training in a synth's parameters, settings, associated software and sound design, this is one of the best instrument tutorials we've seen. MTF

MTF Verdict

An excellent, densely packed look at every element of the synth, with plenty of useful sound design tips throughout.

10/10

Moog Model 15



Publisher **Moog Music**
Price **$29.99**
Contact **via website**
Web **www.moogmusic.com**

Key Features
- Faithful recreation of the Model 15 Modular Synth
- Includes ribbon controller and Animoog keyboard
- Loop recorder, arp, stereo ping-pong delay
- Inter-app Audio, Ableton Link, Bluetooth MIDI
- Compatible with iPhone 5s, iPod Touch 6, iPad Air or newer

Excellence 10/10 MusicTech

Moog's latest foray into the app world is a faithful recreation of the looks and sounds of the Model 15 modular synth. Available for newer generation iPads, and amazingly, iPod Touch and iPhone, the app contains all the oscillators, filters, envelopes and other associated modules from the hardware unit, which can be patched together at will, with bonuses including preset saving, MIDI over Bluetooth, Audiobus and Ableton Link support, a built-in recorder and arpeggiator, plus a ribbon controller and keyboard taken from Animoog for added expression. Although daunting at first, there are a couple of tutorials to get you started making sounds, and if you're looking to learn about programming modular systems, this is an excellent starting point. It also sounds great, with fine control over each dial giving that feel of true analogue precision. MTF

MTF Verdict

Initially daunting, but an excellent modular learning tool, and a superb-sounding analogue instrument with well-thought-out additional features.

10/10

Dark Underground Vocals



Publisher **Loopmasters**
Price **£24.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**

Key Features
- Dark, spoken hooks and phrases
- Wav, Apple Loops and REX2 formats
- 2,000+ Samples, 450+ MB
- 35 sampler patches for EXS24, Kontakt, HALion, NN-XT and SFZ
- Linking words, countdowns and rhythmic loops

Value €£$ MusicTech

Dark Underground Vocals from Loopmasters was created by ARTFX and looks to offer something different, with a harder edged collection of over 2,000 spoken male phrases in Wav, Apple Loops and REX2 formats. You'll find plenty of different single words and short phrases divided into 100, 140 and 174BPM folders, with each vocal performed or processed in a handful of different ways. Most of the hits have been heavily compressed, layered and slightly distorted to give an almost robotic and evil sound. You also get a folder with a countdown from 10 in five styles, a folder of bonus linking words to help create longer phrases, and 30 rhythmic, chopped vocal loops. This pack would be especially good for electronic music producers looking for lots of options when adding a dark vocal hook to a build-up or just before a heavy drop. MTF

MTF Verdict

An extensive and flexible collection of dark-sounding single words and short phrases, which is aimed at the harder side of electronic music.

8/10

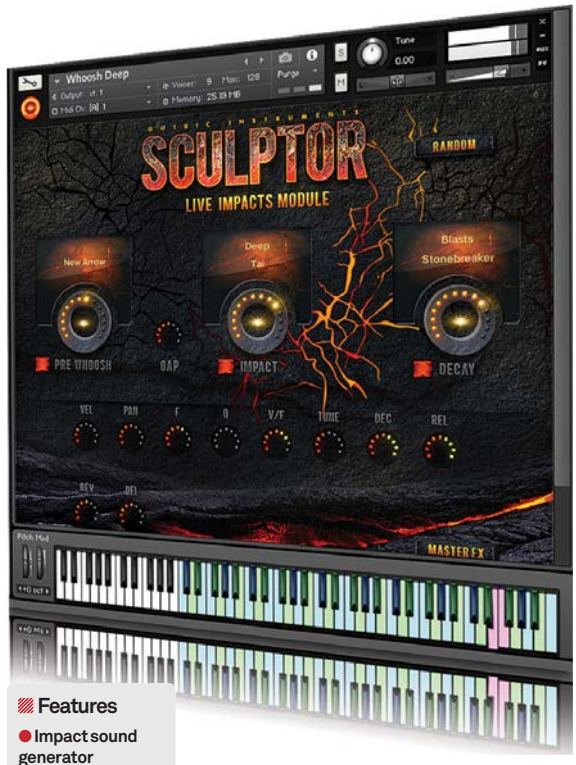
Gothic Instruments Sculpture

Developer **Gothic Instruments**

Price **£54.95**

Contact **+44(0)1837 55400**

Web **www.timespace.com**



What is not to like about a company that calls itself 'Gothic Instruments'? Luckily, its products are almost as good as its name and have made quite a mark. Dronar is an atmospheric modular software instrument that we reviewed a few issues back and loved.

The concept there was atmosphere aplenty, with film and soundtrack composers the target audience; and that still applies with Sculpture. Here, though, the story is a little simpler with everything being about impact – those tension-building slams that hit you in many a soundtrack: from action adventure films through to suspense and horror.

Features

- Impact sound generator
- 1.74GB (with 1.46GB of .aif samples)
- 82 presets
- GAP dial adjusts time between whoosh and impact
- Random button for new sounds

Sculpture is a Kontakt-based library – you'll need the full version – and like Dronar has a great front end with some graphical flourishes that make playing the instrument almost as exciting as the noises it makes.

Load up an impactful sound from the 82 presets, and you'll initially feel that Sculpture is a one-trick pony. It does the impact thing and it does it very well, but there's thankfully more on offer in terms of control. You get three main dials for Pre Woosh, Impact and Decay adjustments which, Pre Woosh aside, offer some incredibly dramatic edits over the main preset. But that's not all. A series of tuning and envelope dials lie beneath the main controls offering yet more control over the sound (the tuning can be particularly severe if you want it to be). Effects can also be adjusted, including a very comprehensive Master FX section.

So the bottom line is that Sculpture offers impact and delivers it and there's a lot more flexibility than you might think. Yes, it's about big, dramatic, attacking sounds, but if you put in a small amount of effort you can also take the library into some quite extraordinary areas. If you are making dramatic scores then you need Sculpture. Just work with it... **MTF**

MTF Verdict

Bang, crash and a side order of wallop. Sculpture does what it says it will do with aplomb.

8/10

Recording Unhinged Massy/Johnson

Publisher **Hal Leonard**

Price **$29.99**

Web **www.halleonardbooks.com**



Features

- 242 pages
- Chapters on engineers, odd recording places, good recording spaces, vocals, bass, drums, guitar, piano , strings, keys, percussion, production, and mixing. All twisted!

Excellence 10/10 MusicTech

You might initially think that Syvia Massy is telling you to rip up everything you know and start again. But with *Recording Unhinged*, it's a lot more than that. What her and co-author Chris Johnson are really saying is, think outside the box... and recording studio. There are rules, of course there are, but question them. Do you really need compression? Can you record on a train? Or on top of a mountain? (It's 'yes' to both of these.) Then there are the unconventional tips and techniques broken down for recording everything, chapter by chapter. In the opening of the vocal one, for example, Massy explains how she sits with a vocalist while recording, rather than in the control room, so they can see her reaction. It's filled with amazing insights from Massy, but also a host of quotes from just about every studio name. A stunning book. **MTF**

MTF Verdict

If you don't learn something new and unconventional from this book, you are not human.

10/10

Super Audio Cart

Manufacturer **Impact Soundworks**

Price **$149**

Contact **via website**

Web **www.korg.com**



Key Features

- Videogame sample library
- Samples from 8 systems (including SNES, NES, C64 and GameBoy)
- Over 6,000 samples from 630 sources
- 5 effects racks with 8 modules each

Choice 9/10 MusicTech

The genius of Impact Soundworks' Super Audio Cart is not so much the idea – to sample eight classic video game systems and bring their sonic prowess into a Kontakt/DAW environment – but more in just how well the developer has achieved it, and quite how much power SAC delivers.

You can run four layers of SAC in Kontakt and even this is implemented very easily, but it's what goes on behind the scenes that is the crucial and brilliant USP. The effects are stunning – including a dedicated custom SNESVerb – but the number of sound-parameter options take this into synthesis territory. So don't think it's just bleeps, whistles and 8-bit sound effects, as SAC goes much deeper. Classic video-game systems? Chiptune? Synths? It doesn't get much more geeky – nor better – than this. We've not had more fun and finally, our two favourite pastimes perfectly come together as one! **MTF**

MTF Verdict

Chiptuners have been messing with this stuff for decades, but now we all get to have a go and it's far more musical than you'll believe.

9/10

UVI Thorus Polyphase Modulator

Developer **UVI**

Price **€79**

Contact **via website**

Web **www.uvi.net**



If you thought modulating effects were dull and boring, then think again. UVI is pretty much trying to reboot your modulation thinking with Thorus and has created an effect that gives you plenty of the old-school but introduces so many simple parameters, oodles of power and such a lovely, simple interface that it almost makes this effect an instrument in its own right...

The installation process was relatively straightforward – as ever with iLok, there seems to be at least one stage too many – but thankfully, no dongle is required, just a free account.

Load the plug-in up and the first thing that strikes you is the simple, modern GUI and controls – Speed, Depth and so on – that show you exactly what you have dialled in.

So to the concept and, yes, it's modulation – fat choruses, swirling and vibrating filtering, it's all here. But there are several features that set Thorus apart: the first of which is the plug-in's eight-voice architecture, which allows more thickening or more extreme applications. The filtering and EQ side of the plug-in is fully featured, too, and with click-and-drag editing, is very easy to implement – a great feature about the plug-in as a whole, actually. Then there are the Standard and Wide modes that allow you to step up the stereo spread. But just be careful you don't overdo it.

Presets are spread over six sonic categories and display a large range of sounds – although there could be more – from subtle to very musical. This really is a plug-in that not only provides a classic gentle swirling, but also a creativity you might not have experienced on more standard chorus-like effects. It's a plug-in that you'll start with using subtly and then very likely use as an in-depth creative tool with certain instruments, almost as an extension of them, as there's so much on offer. But the best aspect is that GUI – it's simple and attractive, but more importantly, just easy to use and change, either by dial or by dragging. A great tool. **MTF**

Features

- Modulation plug-in (AU, VST, AAX)
- 400MB of hard disc space required
- iLok account required (free one, no dongle)
- 36 presets spread over Bass, FX, Guitar, Keys, Synth and Voice

MTF Verdict

Thorus takes the concept of modulation up a few levels and will aid creativity in many ways.

9/10

Beyerdynamic DT 1770 Pro

Developer **Beyerdynamic**

Price **€599**

Web **http://europe.beyerdynamic.com and www.polaraudio.co.uk**



Beyerdynamic's DT 770s have been regarded by some as being the standard in studio headphones, combining a great sound with good isolation, and winning them many fans among producers and engineers.

The DT 1770 is the latest update, what Beyerdynamic calls 'the evolution of a studio legend'. First things first, they're not cheap, at €599 which in post-Brexit Britain (ooh, getting a bit political there) is over 500 sheets for a set of headphones. But do we have to tell you again – no we don't – that throwing money at stuff to monitor your music, is a good thing to do? (As long as you get a true sound in return.)

Firstly, comfort levels are great. The soft foam pads and light weight offer a great feel and isolation – among the best we've noted on passive headphones. They do sit very well on your head and are not too cumbersome during extended sessions.

First impressions on the sound would suggest a slightly pronounced upper – it's a great sound, but we're questioning the overall accuracy. Time to compare to our two sets of reference phones: the half-the-price AT ATHR-70s and the double-the-price AKG K812s, probably our favourite headphones ever.

And you know what? The DT 1770s sit exactly where they should price wise, somewhere in between. The ATs are still our favourite long-term 'phones – light as you can want and extremely comfortable and the AKGs are as immersive and accurate as you like. But the DT 1770s offer a great mix of both, a lovely sound and comfort balance that will have you saying 'ooh listen to this' through them to anyone passing your studio, and easily accurate enough for you to pick out several mix elements and put them in the right damn place, which is what any monitoring (speaker or 'phone) is all about at the end of the day.

Yes they're expensive. But, as we always say when it comes to good accuracy, comfort and long-session wearing, they're worth it. **MTF**

Features

- Closed dynamic studio headphones
- Impedance: 250 ohms
- Frequency response: 5Hz to 40kHz
- SPL: 102 dBSPL (1mW/500Hz)
- Weight: 388g

MTF Verdict

The studio legend does indeed carry on with an added '1'. The DT1770s are damn fine...

9/10

Gear4 Stream 3

Manufacturer **Gear4**
Price **£129 per speaker**
Contact **info@gear4.com**
Web **www.gear4.com**



Bluetooth speakers like the Creative one below are superb for listening anywhere – and even, dare we say, for limited mixing on the move – but here's a wireless speaker system for your entire house. Yes, this is more about listening than music production but, on the face of it, it looks like it could be a revolution in the way that we consume music.

Gear4's Stream system enables a speaker system to be created in your home by placing one speaker per room (£129 each for the horizontal Stream 3 shown above; £99 for the vertical and more mobile Stream 1) and controlling it via Wi-Fi and a Stream app.

You can use an existing Wi-Fi router or set up a speaker network without it, as they include an internal router so can be a self-contained system.

Innovation MusicTech

Features

- Multi-room Wi-Fi speaker system
- Offline and online networks
- Tap into your library or internet radio stations
- Free Stream app to control it all
- Charge-out feature enables 'phone charging while streaming your music

Setting up is a little fiddly – there's a lot of button pushing and pairing and some confusion over whether you are connected to the internet or not to do certain tasks. You also rely on certain button flashes on the top and presses around the back and some of the symbols on said buttons aren't that obvious. But the Quick Start guide is your friend here – far better than using the app on its own, which does have its uses later. However, once it is all up and running, you don't need to do any of this again as your network can easily be recalled and it's easy to add more speakers to your network via the app.

On a basic level, then, you can use the Stream as a wireless speaker system for your iTunes library or home server. But it's the Stream app that unlocks more. Not only can you have the same music beamed into every room, but you can change it per room (that is per speaker), access internet radio or use Aux inputs to run external audio through the system.

Sound-wise, it's very good, given the size. The Stream 3s – the main building blocks of the system – have a good bass response. They do lack a little upper mid sparkle – a little boxy, perhaps – but they hit a decent volume when pushed, certainly enough to fill rooms, and we can imagine that the whole system cranked up throughout a house would provide a great party playback system, especially with different rooms tailored to different genres. Clearly, a club-style world beckons – in your house!

You've got to love the price, too. A system like this a few years back would have cost many thousands but now, you can fill your house for less than four figures. With your iPhone controlling it all, it's a house-sized jukebox. **MTF**

MTF Verdict

The future of home-audio listening is here and, beyond a fiddly start, it's great and very cost effective.

8/10

Creative Muvo 2c

Manufacturer **Creative Labs**
Price **£39.99**
Contact **via website**
Web **uk.creative.com**



Choice 9/10 MusicTech

Creative seems to have nailed this small speaker thing. We were impressed by the Muvo Mini, the company's Lego-type speaker, last year and now the Muvo 2c is booming out on our desktop. It's tiny, we mean really small, but the sound – helped by a rear woofer-type affair – is not. It packs a real punch for something so small. Ideal for listening, you may realise, but considering you can take inputs via Bluetooth, Aux, USB and SD, and more importantly, pair two together, you could even use these for pretty good mixing on the move. Okay, they're not going to give you pinpoint accuracy – but for mobile, wireless monitoring, if you need to go really small and convenient, these take some beating. **MTF**

Features

- Palm-size Bluetooth speaker
- Splash proof
- Two can be paired together
- Bluetooth, SD card, Aux and USB inputs
- Dimensions (HxWxD): 67x93x38
- Weight: 165g

MTF Verdict

Stupidly small, stupidly big sound and stupidly cheap. What more do you want? Splash proof, you say? Well, they're that, too, as it goes!

9/10

iRig HD 2

Manufacturer **IK Multimedia**
Price **£99.99**
Contact **+44 (0)1223 234414**
Web **www.ikmultimedia.com**



Choice 9/10 MusicTech

When we reviewed the original iRig HD – a simple but very high-quality guitar interface for iOS and USB – we loved the quality, but felt is could do with some added extras. Well, it's almost like IK read the criticism and responded [*that's exactly what happened* – Ed], as the company has added several features to version 2.

The great sound quality is there – 24-bit/96kHz A/D – and it performs exceptionally well and quietly in this regard. But now you also get a headphone out and level control; an FX Thru switch; an amp out and even a mic clip.

All of these extras are very welcome, especially as the price has gone up a little (blame Brexit). In use, the unit is still compact and lightweight and with a full version of AmpliTube – IK's almost legendary amp-sim software – it should be in every guitarist's utility belt. Do guitarists wear utility belts? Maybe they should. Maybe that's IK's next product. The IK iBelt. You heard it here first... **MTF**

Key Features

- 24-bit/96kHz interface
- 1/4" amp o/p jack with switchable o/p FX and thru
- 1/8" headphone out with preamp and level control
- Mic stand mounting clip and Velcro strip
- Comes with AmpliTube for Mac, PC and iOS
- Size (mm): 39 x 99 x 21
- Weight: 51g

MTF Verdict

An upgrade to an already fine product, iRig HD 2 adds all the refinements we wanted – although you do pay another £20 for them.

9/10

Save a huge 45%
When you subscribe to MT

If you're reading *MusicTech* from the USA, we've got a special winter offer just for you! Treat yourself and save 45%! Pay just $52.75 every 6 issues.



DIAL TOLL FREE
800 428 3003
QUOTE CODE MTP169

ORDER ONLINE AT
www.imsnews.com/musictech
QUOTE CODE MTP169

MusicTech magazine's subscriptions are now handled in North America by IMS News. Call at local rates, in your hours and pay in US dollars.

MusicTech FOCUS DVD ROM

ON YOUR FREE DVD

With each new version of Live, Ableton continues to set the bar when it comes to building a creative toolbox for both writing and performing your music, no matter what genre you work in. We've compiled a wide selection of content to help you delve deep into Live and start making music. There's over 90 minutes worth of videos and pro tuition, the latest software demos, freeware plug-ins, and promo videos showing off cutting-edge synths, effects and controllers. You'll also find royalty-free loops, samples and hits to use in your own productions, as well as all the files you need to follow along with the workshops at home...



MTF On the disc Over 90 mins of pro video tuition



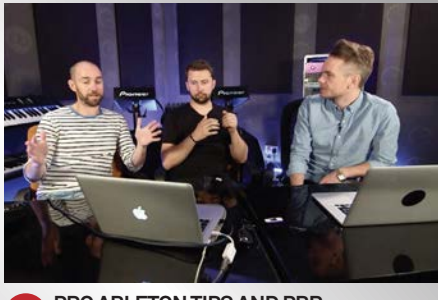
01 USING SAMPLE SLICES AND PUSH 2 WITH SIMPLER

Groove 3 has provided eight videos including how to chop and compose using sample slices within Simpler, how to set up macros knobs to control multiple parameters at once, and using Push 2 to control Simpler's various aspects and modes. **www.groove3.com**



02 PRODUCING 80S SYNTH POP

Groove 3 also takes a look at how to create an 80s synth-pop track using Live and a range of third-party instruments and plug-ins. Chapters include creating a chord pattern, a super wide and wet 80s clap, analogue bassline and synth lead melody. **www.groove3.com**



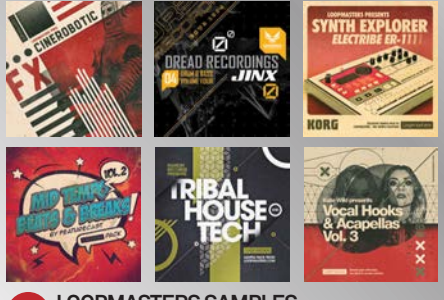
03 PRO ABLETON TIPS AND PBR STREETGANG TRACK MASTERCLASS

Point Blank Music School's Freddy Frogs offers tips on global sidechaining, using Link, creative rhythm programming, multiwarping and using Convolution Reverb Pro. There's also a track masterclass by PBR Streetgang on *12.32*. **www.pointbanklondon.com**



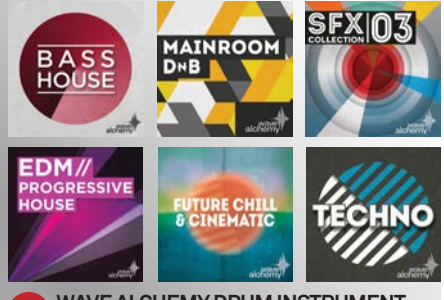
04 MAKE MUSIC WITH PUSH

Producertech's Rob Jones explores some top Push 2 techniques including using the pads and Note Sequencer to play and write basslines, and creating melodies with Slice Mode. Plus, we find out how to make a track using just Operator, with accompanying project files. **www.producertech.com**



05 LOOPMASTERS SAMPLES

A hand-picked library of royalty-free samples, including Cinerobotic FX, Dread Recordings Vol4 Jinx, Futurecast 2 Modern Breakz, Kate Wild Hooks & Acapellas Vol 3, Maison Records Tribal House & Tech and Synth Explorer ER1. Use the code MUSICTECH10 for an exclusive discount. **www.loopmasters.com**



06 WAVE ALCHEMY DRUM INSTRUMENT AND SAMPLES

A free version of the Revolution Kontakt instrument, plus loops from Bass House, Drum Tools 02, EDM & Progressive House, Future Chill & Cinematic, Mainroom DnB, SFX Collection 03, Tech House and Techno packs. **www.wavealchemy.com**

MTF Your Disc

MTF DVD 46 Ableton Live 2017

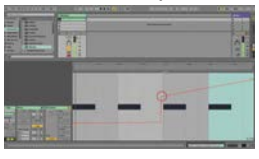


PROMOTIONAL VIDEOS

We've got over 900MB of videos showcasing the latest plug-ins and hardware, including a huge range of cutting-edge hardware synths and software instruments from Arturia, IK Multimedia, Korg, Rob Papen, Roland, Sonic Academy, Sonokinetic, Spectrasonics, Spitfire Audio and Wave DNA. You'll also find high-quality controllers, desks, microphones and interfaces from ROLI, Allen & Heath, sE Electronics and RME, plus top-of-the-range software effects and processors from FabFilter, iZotope, Sound Radix, Waves and Universal Audio.

USING OUR WORKSHOPS

Whether you're a seasoned pro or totally new to Live, we've got a host of Live workshops to help you improve your programming and mixing skills. Topics include using Live alongside iOS, using Ableton Link to sync with other devices, working with Sine waves, how to streamline your track count and more. Where appropriate, you'll find hi-res images, project files and audio for each of the workshops.





SOFTWARE DEMOS

Why not try out the latest plug-ins and software to use alongside Live's native effects. From innovative synths to complex, modern processors and tools, we've rounded up a range of demo and freeware software for you to try out. You'll find plenty of creative effects, instruments, EQs, distortion and saturation effects, stereo tools, dynamics processors and more to help you compose and fine-tune your tracks.

ROYALTY-FREE SAMPLES

We've rounded up a top-notch collection of royalty-free samples from Loopmasters and Wave Alchemy for you to use in your tracks. You'll find an eclectic range of beefy analogue basses, heavyweight d'n'b breaks, jackin' house drum loops, innovative FX, ambient textures and soulful vocals. All files come in 24-bit WAV, plus a Kontakt instrument for Wave Alchemy's Revolution drum machine – ready to be imported, warped, chopped and manipulated.

MusicTech DVD ROM On the disc

MusicTech FOCUS DVD ROM

YOUR DVD CONTENT FILES

ZIP FILES

To maximise the amount of content we can bring you on each DVD, the video, tutorial and samples files are supplied compressed ('zipped'). Mac users should be able to decompress ZIP files simply by double-clicking on them; PC users may need to download a utility such as WinZip (www.winzip.com).

TUTORIAL FILES

The software tutorials that feature in each issue of *MTF* are almost always accompanied by files and audio so you can work through them on your system. These files are zipped to reduce the space they occupy on the DVD. Download them to your hard drive and unzip them to access the individual files (remembering to eject the DVD to prevent your computer from slowing down).

WHAT IS ROYALTY-FREE?

Any *MTF* DVD content marked 'royalty-free' can be used in your own original compositions (even commercial ones). You may not, however, resell these samples in any other form.

DEFECTIVE DISCS

In the unlikely event that your disc is defective, please return it to: Disc Returns, Anthem Publishing, Suite 6, Piccadilly House, London, Bath BA1 6PL. We will endeavour to supply you with a replacement disc immediately. Please note that we're unable to provide technical support for the software on the *MTF* DVD – please check our website at www.musictech.net for any known problems.

MISSING YOUR DISC?

If your disc is missing, contact us at editorial@anthem-publishing.com with your full postal address and the issue number.

ANTHEM PUBLISHING



Push

Music at your fingertips

Ableton.com/push

Ableton

Making Music

74 Creative Strategies for Electronic Music Producers

Dennis DeSantis

Making Music
74 Creative Strategies for Electronic Music Producers

Dennis DeSantis

For Alison and Cecilia, who inspire everything I do.

Preface

What is this book?

For many artists, nothing inspires more existential terror than actually making art. The fear that we're not good enough or that we don't know enough results in untold numbers of creative crises and potential masterpieces that never get realized.

Electronic musicians used to be able to hide behind clunky, emerging technology as an excuse for inaction. But musicians today live in a golden age of tools and technology. A ninety-nine-cent smartphone app can give you the functionality of a million-dollar recording studio. A new song can be shared with the world as soon as it's finished. Tutorials for every sound design or music production technique can be found through a Google search. All of these developments have served to level the playing field for musicians, making it possible for a bedroom producer to create music at a level that used to be possible only for major-label artists.

But despite all of this, making music is still hard. Why?

Making Music was written both to answer this question and to offer ways to make it easier. It presents a systematic, concrete set of patterns that you can use when making music in order to move forward.

Each pattern is presented in the following way:

> A problem is stated. A problem is a roadblock that stops you from making progress with a particular piece of music. The problems in this book are real-world situations—you'll likely recognize many of them as things that have held you back in the past. A problem might appear at the beginning (e.g., you don't know how to start), in the middle (e.g., you've created lots of material but don't know how to organize it), or near the end (e.g., you keep making changes and

can't decide how to finish).

- > The problem is explained in more detail via examples and (sometimes) references to other patterns.
- > A solution is provided. A solution is a concrete instruction or small set of instructions that will solve the stated problem. Like the problems, the solutions are also real; if you apply the solution, the problem will be solved. Note that this requires you to actually do the solution; in most cases, reading it won't be enough to move you forward. Making Music can show you the way, but you still need to do the work.
- > The solution is explained in more detail via examples and (sometimes) references to other patterns.

Who is this book for?

If you make original music using computers and you've ever found yourself struggling to complete your musical projects, Making Music was written for you. While many of the patterns discussed here can probably be modified or directly applied to other types of music-making (such as composing for rock bands or string quartets), the goal of this book is to solve the specific problems that people have when working with machines, rather than with instruments or other people.

While no prior skills are really necessary to make use of this book, I've written it assuming you have a basic understanding of at least one digital audio workstation (DAW) or similar music production environment. No specific tool is required, and the problems and solutions discussed aren't specific to the workflow of any particular piece of technology. A basic understanding of the fundamentals of music—chords, scales, and concepts of rhythm—is useful but not a prerequisite.

Although it may not always be obvious, the patterns are all general enough that they can be used by musicians working with any genre of electronic music, from commercial dance music to the avant-garde. Although some of the explanations used refer to actual genres or even specific examples of music, I encourage you to read "around" these

descriptions in order to get at the essence of the pattern so that you can apply it to your own work.

Who's the author?

I am a musician with a background in a variety of different worlds. I studied classical composition, music theory, and percussion, although these days I primarily write electronic music in the direction of house and techno. I grew up just outside of Detroit, and all of the amazing music that came from and through that city was a big influence in my early musical development. Although this book aims to be as genre-neutral as possible, it's very much written from my personal perspective—the things I talk about here are a reflection of things I actually think about and employ in my own music.

How to use this book

Think of Making Music as something like a travel guide. There's no explicit order to the patterns, although things tend to be loosely grouped by concept. You can read and experiment with the various patterns as you need them in order to solve particular problems as they come up in your own work. Sometimes the patterns will explicitly relate to others, and I often refer to other patterns to help clarify the current one. So while it's not necessary to read the whole book cover to cover, doing so will probably help you to find relationships between the various patterns and to see them as a system, rather than as isolated examples.

How this book is organized

The patterns themselves are grouped into sections based on where in the writing process a particular musical problem is likely to occur. These are:

- Problems of beginning. These are problems that prevent you from starting at all. They include problems of inspiration, problems realizing the sounds you hear in your head, etc. The solutions include exercises to develop better active listening and explorations of various ways of playing with sound, harmony, melody, rhythm, and musical form.
- Problems of progressing. These are the most common roadblocks when working and occur once you've made something but are still a long way from the end. They include problems of fatigue, problems developing and varying material, etc. The solutions include exercises to generate new material, shape song structure, and stay in the creative flow.
- Problems of finishing. These problems occur when you feel that almost (but not quite) everything is in place, but you are unable to reach a satisfactory conclusion. The solutions include ideas for creating convincing arrangements and for creating powerful endings.

Why is this book needed?

There are already many ways—books, classes, video tutorials, software documentation, private teachers—to learn about music technology and music production. I'm a strong supporter of all of this, and I encourage anyone who's interested in this book to also take advantage of these resources. But almost all of them focus on the second half of the equation—technology or production—rather than the first half: music. Making Music is an attempt to help people who are comfortable with the basics of music production at a technical level but who still find music-making to be a difficult process (which I suspect is all of us!).

Finally…

This book will not teach you how to use a compressor, program a synthesizer, or make a great-sounding kick drum. Those aspects of music-making are already well covered. What it will teach you is how to make music using those tools, with a specific emphasis on solving musical problems, making progress, and (most importantly) finishing what you start.

While I hope you find Making Music inspiring, I hope even more that

what inspires you is the music you make using these patterns. Making Music is not a collection of vague aphorisms. Instead, it combines motivational ideas about the philosophy and psychology of music-making with hands-on tools and techniques that musicians of all kinds can use to really get work done.

Contents

Preface

Problems of Beginning
Three Ways to Start
Catalog of Attributes
Avoidance List
Active Listening
Listening to Music You Hate
Mise en Place
Arbitrary Constraints
“Write drunk; edit sober”
Choosing a Tempo
One Part at a Time
Different Tools
Presets as Starting Points
The Tyranny of the Default
Simple Sounds
Extended Techniques
Goal-less Exploration
Procrastination and Timeboxing
Collaboration
Thinking Like an Amateur
Scraps and Sketches
Problem-Solving as Distraction
Personal Workflow
Originality vs. Quality
Process vs. Product
On Work

Problems of Progressing
Breadth Before Depth
Foreground, Middle Ground, and Background
Fuzzy Boundaries
Creating Variation 1: Mutation of Clones
Creating Variation 2: Mutation Over Generations
Creating Variation 3: Note Transformations
Implied Rhythm in Short Loops
Asynchronous or Polyrhythmic Loops
Misusing Rhythmic Tools

3+3+2 Rhythm
Programming Beats 1: On Looseness
Programming Beats 2: Linear Drumming
Programming Beats 3: Ghost Notes
Programming Beats 4: Top, Bottom, Left, Right
Bass Lines and Kick Drums as a Single Composite
Drums to Pitches and Vice Versa
Tuning Everything
Silence and Noise
Sampling the Old-Fashioned Way
Creating Melodies 1: Contour
Creating Melodies 2: Using Motives
Multiple Simultaneous Melodies
Linear Rhythm in Melodies
Sound-Color Melody
The Rhythm of Lyrics
Creating Harmony 1: The Basics
Creating Harmony 2: Beyond Triads
Voice Leading and Inversions
Parallel Harmony
Harmonic Rhythm
Repetition and Insistence
The Rhythm of Automation
Humanizing With Automation Envelopes
Maximal Density
Deliberately Bad Listening
Randomness and Responsibility
Dramatic Arc

Problems of Finishing
Arranging as a Subtractive Process
The Power of Erasing
Formal Skeletons
Common Forms 1: Elements of Song Form
Common Forms 2: Layering as Form
Short Loops as Endings
Unique Events
Three Ways to End
Rendering as Commitment
Getting Feedback

Diminishing Returns
Fail Better

Problems of Beginning

Three Ways to Start

Problem:

You're staring into the void of a new empty project in your DAW, and you have absolutely no idea how to begin.

The blank slate might be the most intimidating place in a creative environment. Once we're in a flow, ideas tend to spawn additional ideas. But before we have anything, all options are available, so choosing seems impossible. One Part at a Time is certainly a way to move forward, but which one part comes first? When you have nothing, even choosing one part can feel like an impossible hurdle.

The simple-sounding answer is "It doesn't matter how you start; just do!" This might work well as a sports slogan, but it's perhaps too glib and dismissive to be really practical for creative work, in which the number of things to potentially "just do" are limitless and the path forward is not at all obvious. Here are three practical suggestions for how to begin from nothing.

Solution:

1. **Start with the foundation.** In most genres, we can think of the "bottom" of the music as being the low-pitched or purely rhythmic instruments, such as the bass and drums. On top of these are added instruments that are progressively higher in pitch. By starting with the bottom, we provide both a conceptual and musical foundation for everything else. The drums often provide the essential timekeeping elements, while the bass often provides the notes that define and anchor the chord progression. If you're working in more experimental genres that don't use these conventional instruments, it's still likely that some elements can be considered foundational—perhaps a droning layer or something that approaches a repetitive rhythm.
2. **Start with what you hear.** Many musicians never (or rarely) get spontaneous musical ideas—all of their music comes from active work. If you happen to be lucky enough to hear original musical ideas in your head, then you should absolutely use them as the basis for your own work. For example, maybe you have a melodic idea that you've been humming, or a rhythm that you've

tapped out on the table. Just because these ideas came to you outside of your active music-making context doesn't mean you should discard them. On the contrary, these accidental ideas are sometimes the most interesting ones you can have.

3. **Start with what you know.** If you play a "real" physical instrument, try using it to generate your ideas. Even if you're writing purely electronic music and have no plans to use instrument recordings in your work, your natural physical connection to your instrument may help you come up with more interesting and organic musical ideas than you can get from just working with a mouse and a MIDI controller. For example, guitarists tend to voice chords and approach harmony in a different way from keyboardists. But because keyboards are the de facto control surface for entering music into a DAW, many guitarists might never think to use their guitars in an electronic music context. Drummers rarely play the kind of beats that are used in many types of electronic music, but sitting behind a real drum kit might stimulate creative ideas that would feel alien on a pad controller. The trick with this approach is being able to accurately translate the acoustic idea to the electronic medium, but that's a good problem to have; much better than having no ideas at all.

Catalog of Attributes

Problem:

Creative musicians find inspiration in other music. While we seek to make music that is uniquely our own, every other piece of music we hear is automatically processed and becomes an unconscious part of our musical vocabulary. Taking too much is theft. Taking too little fails to acknowledge our influences.

For musicians who aim to find and develop a unique voice, there will always be an internal conflict to resolve when hearing music that is inspiring. To truly be original, is it necessary to ignore all external influence? How much can you "take" from other music before what you've made no longer feels like it's yours? What and where is the boundary between homage/inspiration and plagiarism/copying?

Unfortunately, these aren't questions with "right" answers. Objective legal issues aside, each artist needs to determine their own level of comfort when borrowing from other sources. But there are some strategies that let you infuse your own work with the "essence" of your inspiration, while simultaneously forcing you to make something new. One of these is to write a catalog of attributes.

Solution:

Listen carefully—and many times—to the piece that inspires you (the "source"). Study it, element by element and layer by layer, until you can write down a catalog of its attributes. Once the catalog feels complete, set aside the original source, instead referring only to the catalog as a template for your own new work (the "target").

Consider the attributes of sound, harmony, melody, rhythm, and form. Write something concrete about what you hear for each attribute. If you're comfortable with notation, feel free to use it in your catalog, but sparingly; the goal is to capture only the framework or scaffolding of the source, including the aspects that make it inspiring, but without simply recreating it. You should end up with a description, not a transcription.

The catalog's level of detail may vary depending on a variety of factors: your own ability to translate what you hear into words, the depth and complexity of the source, the amount of time you choose to spend, etc. What is important is not the actual level of specificity, but only that it gives you enough to use as a template without having to refer to the original again during your own creation process.

A basic catalog of attributes might look like this:

> 122 bpm
> Sound elements: drums, (808, four-on-the-floor, lots of filter motion on closed hats), bass line (FM-ish?), electric piano (distorted but dry), female vocals (breathy verses, full-voice choruses), lead synth (big supersaw, but only after the second chorus).
> Harmony: mostly D minor alternating with A major until breakdown. Breakdown section is in D major (sort of?). After breakdown, rest of track is in E minor alternating with B major.
> Melody: not much. Lots of D, with occasional jumps up to A and down to Bb.
> Rhythm: four-on-the-floor drums (basic house beat). Bass line is mostly offbeat eighth notes (trance influence?). Cool metallic hit on the "and" of beat 2 every four bars.
> Form: additive layering; drums start, then each element enters one by one. At breakdown, everything drops out except hi-hats and bass line, then rebuilds additively. Form is all 16- and 32-bar sections. (Verse 16, Verse 16, Chorus 16, Breakdown 32, Chorus 16, Chorus 16)

This catalog of attributes could describe an endless number of new works. In fact, you can probably hear music that fits these attributes in your head already. It is complete enough to serve as a template but not so descriptive as to allow for a recreation of any particular piece of

existing music; if two musicians read the catalog, there would be almost no possibility that they would use it to write the same thing. Now try to put yourself in the mindset of one of those musicians; using only the catalog as a recipe, make something new.

Avoidance List

Problem:

Sometimes when we listen back to a recently finished track, we find that it's too similar to something we've written before. We've managed to develop a personal style, but we're so bound by it that we feel unable to create things that really feel new.

We often discover this after listening back to a track a few days after it's finished. While working (or immediately after), we don't notice the similarities. But when we have a bit of distance from the new work, we're suddenly able to hear how it compares to the rest of the music we've done, and the similarities become painfully obvious. Often, it's a similarity in sound—the same types of samples or synth sounds keep coming up. Sometimes, it's a similarity in form—we rely on the same kinds of transitions or formal structures from track to track. It can feel like you're writing the same music over and over again.

Note that, depending on the genre you're working in, a high degree of consistency may be exactly what you're going for. In commercial styles like EDM, for example, predictability of structure and overall sound is a fundamental attribute of the style. Radical sound design and unusual forms wouldn't make sense in this music, and the interest comes from seeing how effectively artists can work within the strict boundaries that define the genre.

But if you're uncomfortable with the feeling that you're stuck on repeat, here's a potential solution.

Solution:

As soon as you realize that a new track is too similar to something

As soon as you realize that a new track is too similar to something you've done before, start writing a list of the specific musical attributes that make you feel this way. Carefully analyze all of the potential categories—sound, harmony, melody, rhythm, and form—and write down the specific technique that's been repeated. This will form an "avoidance list" for your next track, a list of things that you've decided ahead of time to not do.

If you're feeling inspired, try to come up with some alternatives to the attributes you'll avoid. This can serve two purposes: First, it gives you something to actually do, which will save time when you're working on the track. But more importantly, it might help you identify why you've been relying on a particular type of technique in the past. For example, if you've forbidden yourself from using tempo-synced dub delays, you might decide to experiment more with reverb. But perhaps the underlying reason for your use of delays in the first place is that you're uncomfortable with the idea of space and silence in your music. Maybe this problem can be solved by adding additional layers of musical material. Or maybe there isn't really a problem to solve, and your next track could make interesting use of silence that you would normally have simply filled in by force of habit.

You can think of the avoidance list as a kind of negative version of the Catalog of Attributes. The goal here is to find patterns in existing music that should be avoided, rather than emulated. The difference, of course, is that you're analyzing attributes in your own work rather than that of another artist.

An avoidance list might look like this:

> stay away from 123-127 bpm; try faster (or maybe slower?)
> no 808 samples; (try FM drums or processed acoustic samples)
> no "wood" sounds; (find different percussion elements—white noise or metal?)

> no (or less) filter cutoff/resonance sweeps; (try FM for timbral changes)

> no (or less) sidechaining; (write a bass pattern that avoids the kick, rather than relying on ducking)

> stop using only two chords; (spend time really working out a more comprehensive progression)

> no auto-tuned/chopped vocals (another organic source? non-pitched spoken word samples?)

> no drum-less breakdown; (find another way to create formal tension and release)

Now as you start a new track, you're forced to find new strategies rather than relying on techniques that you already know you're tired of hearing. You may find it difficult to work without these reliable elements of your vocabulary in place. But looking for alternatives will help you expand that vocabulary, while simultaneously ensuring that this track won't simply be a clone of the last one.

~ A note of caution: As with Arbitrary Constraints and Process vs. Product, you should feel free to abandon your pre-planned restrictions at any point in the writing process. If you find that you've eliminated techniques that really would sound better in the context of the music you're working on, you should absolutely allow yourself to use them. Restrictions, plans, and structure can be important learning tools and can help you to get out of ruts. But in every case, the most important thing is the song, and you should do whatever will make the best song possible, including deviating from the plan if necessary.

Active Listening

Problem:

Although you listen to a lot of music, you don't really have a sense that you're learning from what you listen to. You know what you like, but you don't really understand why you like it or how to extract compositional or technical ideas from what you hear so that you can reuse them in your own music.

For most people, listening to music is a passive experience. We turn music on but then engage in another primary activity. The music is playing, and we're loosely aware of it, but it's serving a decorative or soundtrack-like purpose for whatever else we're doing. Although this is certainly better than not listening to music at all, you can learn more if you spend at least some time engaged in active listening.

Solution:

Active listening simply means listening as the primary activity, and it's an important skill to develop. Rather than using music as the background for another activity, try listening without doing anything else. This requires time, quiet, and focus, which are skills you need for your own production work anyway. A good way to start is by just putting on some music and then turning your attention to it entirely. If you're listening at your computer, close any open applications (and, ideally, your eyes as well). At this point, you're not trying to listen with a particular focus, but rather a general one. If you can concentrate and avoid distraction, you'll be amazed by how much more you hear than in a passive listening state.

The next step in active listening is to start trying to deconstruct what's

happening in the music you're listening to. Here are some tips for doing this:

Listen in Layers

A great way to actively listen is to listen to the same piece multiple times and force yourself to focus on a different specific parameter each time. For example, spend one pass listening only for:

Sound: What are the timbral characteristics of this music? What instruments are used? What is the texture (dense vs. sparse)? Are there some specific production techniques that you recognize (either from your own or other music)? What kind of acoustic "space" is suggested by the music (dry vs. reverberant, near vs. far, etc.)?

Harmony: What key (if any) is the song in? What chords are used? Is there a chord progression that happens throughout, or does it change from section to section? If there are no overt chords (as in some minimal or experimental music), is harmony implied in another way?

Melody: What's happening in the melody? Does it have a wide or narrow range? What is its general contour: Angular, with lots of leaps? Stepwise, with motion mostly by one or two semitones? What instrument or voice has the melody? Does this ever change? If there is no overt melody (as in some minimal or experimental music), is melody implied in another way?

Rhythm: How are events distributed within short time ranges like a bar or phrase? Are there patterns that repeat, or do rhythmic gestures happen only once? Are rhythms and tempo overtly identifiable, or is the music free and largely arrhythmic? What instruments have the most impact on the rhythm? What do the less rhythmic instruments do?

Form: How does the song evolve over time? Are there clear sectional divisions or are there Fuzzy Boundaries between regions? What defines one section versus another? Do certain instruments play only in some sections or is the instrumentation the same in every section?

Additionally, if there are specific instrumental or vocal parts that you'd like to understand better, try spending an entire listening pass focusing entirely on only one part. For example, the best way to learn how the bass line works in a particular song is to tune out everything else and focus just on the bass line.

Listen in Chunks

By isolating and looping short durations of music, you can more easily focus on the specific parameters or instruments discussed earlier. The best tool for this is your DAW. Try loading the song you want to listen to into your DAW's timeline, adjust the project tempo to match the song, and then set the loop to a short region—one or two bars or, at most, a single musical phrase. Listen to this loop as many times as necessary in order to really hear what's happening in the parameter you're listening for. Then advance to the next chunk and repeat. When you're done, go back to the beginning and gradually expand the loop length so that you're covering a larger amount of time in a single listening pass. Listening in chunks like this is also a great way to learn or memorize a particular part by ear.

Listen Subjectively

In addition to helping you learn how a particular piece of music "works," active listening can also help you understand your subjective responses to music. For example, are there particular aspects of the song that sound familiar, nostalgic, emotional, etc.? Can you explain why

(perhaps with reference to the parameters discussed earlier)? When listening passively, it's common to have some kind of emotional response. But via active listening, you have a chance to understand what it is, specifically, that causes that response. And once you understand a technique or musical gesture, you'll be able to adapt it for use in your own music.

Listening to Music You Hate

Problem:

Although you listen to a lot of music, you tend to listen within a specific set of genres. And you pretty much exclusively listen to music that you like, or at least music that you expect you'll like based on your previous experience, recommendations, etc.

Even if you write exclusively in a single genre, you owe it to yourself to spend some time listening to a wide variety of music. Here are some tips for how (and why) to listen to music that you may not enjoy.

Solution:

Most people listen to music for entertainment. But producers and composers owe it to themselves to also listen in order to learn something. Inevitably, this will mean spending at least some time listening outside of your comfort zone, to music that you don't like.

This is more than just listening to genres that are closely related to what you normally listen to. For example, if you consider yourself primarily a techno producer, don't be too proud of yourself if you occasionally listen to house music. It's critical that you expand your range of listening to genres that you would never normally listen to for pleasure.

Why is this important? As you listen, consider the following questions:

Are there musical techniques you can use?

If you write in only one genre (and especially if you listen to only one genre), perhaps you've developed a set of working techniques that's

fairly consistent and "appropriate" for the kind of music you make. This may work for you, but it also exposes you to the risk of writing the same music over and over. Other types of music may be made in entirely different ways, and perhaps there are techniques from those worlds that you can appropriate for your own use.

As an example, consider opera (a genre that many electronic producers probably don't spend a lot of time listening to). A single opera may last for hours—a time scale that's closer to an entire DJ set than a single track. How does tension and release work over this amount of time? How does form work? What about harmony? Rhythm? Melody? Overall sound?

Are there production techniques you can use?

In addition to musical techniques, there is also a huge amount you can learn about production when listening to other genres.

Consider country music, for example. Whatever you might think of this music as entertainment, there is no doubt that contemporary country music is generally extremely well-produced and borrows a lot of mixing and production concepts from contemporary pop music. But there are some significant differences. For example, vocals in country music are often much louder than vocals in pop music (or in electronic genres with vocals). Also, pop music (and many electronic genres) tends to be quite bass-heavy, while country tends to emphasize guitars and other mid-range instruments.

Whether or not those production characteristics are useful in your own music depends on what you're trying to accomplish. But simply being aware of what's happening in a wide variety of styles might inspire your creativity or shake you out of your own stylistic ruts.

Why do people like this?

No matter what you're listening to, and no matter how terrible you think it is, someone out there loves it. And unless you're writing music exclusively for yourself, it's a useful intellectual exercise to try to understand why people like what they do. There are, of course, social pressures; people like things because they're popular, or because their friends like them, or because they represent a recognized signifier for a particular subculture. But there are also genuine musical preferences which vary widely (and wildly) between people. Even if you have no intention of writing music for the opera or country music fan, thinking about how other music works, and why disparate groups of people like what they do, can help you become a more well-rounded musician, which can only improve your music.

Mise en Place

Problem:

Getting into a creative mindset is hard enough, but once you're in it, something inevitably goes wrong. DAWs crash, plug-ins need to be updated, you're missing samples you absolutely need, you're out of coffee, the dog needs to be walked, etc. And once you've fixed all of the problems, huge amounts of time have gone by and your creative impulse is gone.

There are many things that can impede progress in the studio. Some of these things are legitimate issues (like equipment that suddenly fails), while others are distractions that we're able to convince ourselves are important enough to get in our way (like a new email).

But because the creative impulse is so fleeting, it's vital that we're able to actually get music made when ideas come. This means making sure that everything we'll need to work is set up and ready to go beforehand, so that we can focus when we need to. Here are some ideas.

Solution:

The term mise en place (roughly translated from French as "put in place") comes from the culinary world and refers to the careful preparation that goes into setting the physical (and mental) space prior to the actual work of cooking. This means making sure that the tools and ingredients you'll need are easily at hand, the kitchen is clean and safe for use, and you've thought ahead to the end of the recipe to ensure that nothing unexpected will stop you once you've started.

There are clear parallels between work in the kitchen and work in the

studio, and employing mise en place as part of your music-making process can help you stay focused when inspiration strikes.

Prepare Your Ingredients

Are your DAWs and plug-ins up to date? Do you have the samples you need? Are they named and organized in such a way that you can find them without needing to go on a time-consuming (and inspiration-destroying) search through all of your hard drives? Just as importantly, are the samples you don't need out of your way and easy to avoid? Are your controllers, audio interface, headphones, and speakers connected and working? Does any hardware need drivers? If you're using external instruments like synthesizers or effects, are they patched in and turned on? If you're using acoustic instruments like drums, are the mics set up and working?

Prepare Your Workspace

Is your chair comfortable? Is your desk uncluttered? Can you physically access any necessary hardware? If you're collaborating with other people, are they plugged in and ready to make noise? Are they comfortable and also in a creative mindset?

Prepare Yourself

Are you hungry or thirsty? How about any animals or children you're responsible for? Is there a critical phone call you need to make or email you need to send? Are your neighbors or roommates going to be bothered by the sound you make?

Nothing kills the creative flow more than having to stop and solve a non-creative problem. Making sure that these factors are dealt with

ahead of time means that you're able to fully embrace the music-making process, with less risk of distraction.

In addition to employing mise en place prior to starting work, there are some additional kinds of preparation and organization actions you can take that will help you be ready when inspiration strikes:

Make Templates

If you find that there are particular combinations of instruments or sounds that you're using often, save empty template projects with those instruments already loaded so that you can be ready to start creating as soon as you have an idea. For example, if you switch between your own solo work and working with other musicians, save a template for each scenario rather than rebuilding these environments from your DAW's default project over and over again. The goal is to get from idea to actual creation as quickly and effortlessly as possible.

Organize as You Go

Even a well-prepared environment can quickly fall into chaos during the heat of the creative moment. As you're working, try to stay as organized as you can (but without falling out of music-making). For example, name tracks before you record onto them, both so you can see at a glance what you're working on, and also so newly created clips on those tracks "inherit" useful names. If your DAW allows it, use color to create visual associations between related clips or tracks. And throw away anything that you've definitely decided not to use. Removing clutter from your project helps you to focus on the important things that remain (and may also reduce your CPU load or memory usage).

Always Be Ready to Capture

Great ideas might come to you when you're outside of the studio. Ideally, you're always prepared with a field recorder or a smartphone app to quickly capture interesting sounds, or even melodic or rhythmic ideas that you can hum or sing before you forget them. You can even use a notebook to capture ideas—in musical notation or in text, pictures, or any other system that will allow you to remember it. The important thing is that you have these capturing tools with you at all times.

Although mise en place can be a huge productivity boost, make sure that you find time for this preparation work that is not the same as your creative time. Don't start getting organized when you're actually inspired to make music. If you feel like making music, that should always take priority. It's much harder to find the energy to be creative than it is to find energy for mise en place, which you can make yourself do anytime. And it's much better to act quickly on a creative impulse—even if you're completely unprepared—than it is to postpone it, try to get organized, and then find you've lost your motivation. Use the time when you're not inspired to take care of mise en place.

Arbitrary Constraints

"My freedom thus consists in my moving about within the narrow frame that I have assigned to myself for each one of my undertakings. I shall go even further: my freedom will be so much the greater and more meaningful the more narrowly I limit my field of action and the more I surround myself with obstacles. Whatever diminishes constraint diminishes strength. The more constraints one imposes, the more one frees oneself of the claims that shackle the spirit."
— Igor Stravinsky, Poetics of Music

Problem:

Music production with a computer offers a limitless field of possibilities. Any sound can be made, manipulated, re-recorded, re-manipulated, etc. But while an infinite range of options might sound appealing, it also means that decision making is hard. The more options you see, the more you need to make active choices about which ones to pursue and which ones to ignore.

Limiting the field of possibilities isn't just about making it easier to work. It's also about making it possible to begin at all. If every possible starting direction is equally appealing, how could you ever choose one?

Solution:

Apply arbitrary constraints before starting to work. Create a "narrow frame" of possibilities, and then act entirely within that frame. Applying arbitrary constraints helps to limit your field of options, allowing you to move forward.

It's important to note that these constraints really are arbitrary; you're attempting to eliminate perfectly valid options rather than bad ones. Bad options are easy to deal with; your own musical sensibilities will reject them automatically. But valid options need special care because their

validity requires you to make active choices about whether to pursue or reject them. Eliminating them arbitrarily helps you avoid decision paralysis. Also, the fact that the constraints are arbitrary means that it doesn't really matter if you're wrong. If you realize during the course of your work that you've constrained yourself too much, or in the wrong direction, you can simply choose to remove the constraint.

Here are a number of ideas for arbitrary constraints, both related to the music itself and also to working method:

Musical Constraints

- **Make every sound from one sample.** An extreme restriction on available sound sources forces you to really think about the character and possibilities of the sounds you choose. Can you make a kick drum sample into a lush pad? How about a hi-hat? What kinds of processing could you use for these transformations?
- **Completely avoid an instrument that would be expected in the genre.** For example, the untitled Peter Gabriel album commonly called Melt has no cymbals. Prince's classic song "When Doves Cry" has no bass line. These kinds of restrictions force your decision making process into new directions. If an entire functional role is missing, how do you fill the gap? In the case of Melt, Gabriel had to find ways to propel time forward without relying on mainstays like hi-hats or ride cymbals. His solution was a range of unconventional percussion instruments. In the case of "When Doves Cry," Prince chooses to leave the sonic space unfilled. Consider both options in your own work.

Constraints on Time

- **Give yourself a deadline.** Nothing motivates like a due date. Since work always expands to fill the available time, it's necessary to actually put a limit on that time. If you find self-imposed deadlines to be too "soft," try having someone else assign the deadline for you, with the requirement that you show them the work at the end to ensure accountability. Or engage in a collective challenge, such as February Album Writing Month (http://fawm.org).
- **Schedule tasks as if they were appointments with yourself.** Try using a calendar to restrict specific types of work to specific times. For example: Sound design: 7-8pm, Form/song structure: 8-9pm, Mixing: 9-10pm

Timeboxing specific tasks serves two purposes: It forces you to narrow your focus while simultaneously eliminating the risk of non-musical distractions (Facebook, etc.). You wouldn't check your email in the middle of a business meeting, so treat these "appointments" with the same kind of care.

Constraints on Space

> **Change your venue.** If you're used to making music in a particular place, try moving to a different one. Simply moving from your bedroom studio to the local coffee shop forces your hand in a number of ways. First of all, you're limited by what you can take with you—no racks of external hardware, for example. Secondly, you're in an environment that is at least somewhat unfamiliar, so you're unable to fully relax (and thus become distracted). Many musicians avoid having a home studio altogether, instead choosing to rent studio space so that they can separate their working mindset from their living mindset.

Although you may not realize it at first, you're most likely already applying certain constraints to your work. For example, simply making the choice to work within a specific genre already eliminates a huge range of musical possibilities. To say that you're working on a "house track" as opposed to a "Balkan folktronica track" implies that you understand certain general characteristics (the instrumentation, rhythms, and harmonic language that identify a given piece of music as being "in" a particular genre) that describe one music as opposed to another.

“Write drunk; edit sober”

Problem:

Sometimes during music-making, we find moments of amazing creativity and flow. But because we have in mind the ultimate end goal of “a great track,” we sometimes stop mid-flow to edit, correct, or otherwise disrupt the creation process. And once we do, it can be difficult to get back into the flow.

Creativity is an intrinsically messy and uninhibited process. Editing, on the other hand, is about refinement and order. “Write drunk; edit sober”—a quote often attributed (most likely incorrectly) to author Ernest Hemingway—uses drunkenness versus sobriety as a metaphor about levels of self-control and the importance of working both with and without restraint. Both modes of working are necessary to create something good, but problems can arise when we don’t keep a clear division between the two. Allowing them to overlap runs the risk of breaking our creative flow.

Solution:

Force yourself to compartmentalize your work into two discrete phases. During the creation phase, your goal should be to capture as much material as possible. Only move forward in time. Delete nothing. Once you’ve recorded something, consider it finished. Better yet, forget about it entirely and move on to the next thing. Aim for speed and quantity, judge nothing, and be prepared to make lots of mistakes.

Once you’ve captured a lot of material (maybe after a set amount of time, or simply once you get bored) switch to the editing phase. Now, your goal is to carefully refine the material that you generated during the

creation phase. Resist the urge to generate anything new during this process, and instead ruthlessly delete, trim, reorder, or otherwise curate the material you made earlier. It's likely that you'll throw away the majority of what you made, and this is healthy. Most of what we make in a truly judgment-free creation mindset is likely to be terrible.

A side benefit of working so freely during the creation phase is that you may find things during the editing phase that are actually amazing but are unusable within the context of the project you're currently working on. When that happens, just save the material that's worth saving (but is wrong for the current song) into a "scraps" folder (see Scraps and Sketches). Then the next time you're beginning a new track but aren't in the mood to start from scratch, simply pull something interesting out of your scraps folder. Depending on how developed your scraps are, you might be able to pick one up and move directly into a new editing phase.

A good technical approach to working during the creation phase is to always be recording. Even if you're just improvising at the keyboard and have no plan or direction, be sure to capture every note you play. A common working method for electronic musicians is to use the DAW as the recording device but use hardware or other sources outside of the computer to generate sound. A collection of synthesizers generating MIDI and/or audio lets you get your hands on something that doesn't feel like an editing environment, allowing you to step away from the DAW (and its inherent bias towards editing). Press record, then play and tweak knobs, capturing everything you do as a kind of free-form jam. While jamming, try to forget that you're recording. Don't try to make something perfect. Simply indulge in the uninhibited freedom of exploring sound. Follow your instincts—if something is working, let that lead you in a direction. If something isn't working, abandon it, but do so without stopping the recording. The trick is to stay out of

judgment mode as much as possible. Simply capture as much as you can, following your instinctive sense of what's right. Don't worry about hard drive space. If you really need to reclaim the space, you can do that later during the editing phase.

Even if you work entirely in the computer, try to find some working methods that are somehow analogous to this physical division between instruments and editors. For example, maybe use one DAW as your creation space and another as your editing space. By treating these as distinct virtual environments, it may be easier to switch between the two modes of working. Alternatively, find a collection of instruments and effects that allow you to create lots of material in an evolving, organic, connected way. For example, modular synthesizers and arpeggiators, even virtual ones, can be great sources of material during the creation phase. Start your recording even before you've started building a modular patch, so that you can catch any happy accidents that occur on the way.

If your DAW allows for comprehensive routing options between tracks, try capturing both MIDI and audio simultaneously during the creation phase. This will give you more ways to edit later and provide more raw fodder for refinement. It will also save time in the editing process because you don't have to listen to everything twice—the MIDI and audio will contain analogous information, so you can listen through a single pass and then decide whether to rework the MIDI, the audio, or both.

The balance between play (creation) and work (editing) is necessary to generate good results. Like Subtractive Arranging, this approach works because it allows you to remove rather than create, and it's much easier to decide that something is bad once it exists than it is to make something good from nothing. It saves you from having to make the

“right” thing the first time, when you may be struggling to simply make anything at all.

~ Note: I wrote the first draft of this chapter in a text editor with an option called “Hemingway Mode,” which disables the Delete key. At the end of the creation phase, it was nearly 2000 words long. After editing, it’s about half that, and the paragraphs are almost completely reordered.

Choosing a Tempo

Problem:

You're often not sure how to choose the right tempo for a track in progress. Does it really make a difference if your track is at 124 bpm rather than 125, for example? And why do some ranges of tempos seem to allow for much more musical flexibility than others?

The choice of tempo depends on a variety of factors, including intended genre, intended audience, and (above all) personal preference. Here are some thoughts about tempo.

Solution:

If you already have a clear intention to write in a specific, well-established genre, then the range of "acceptable" tempos is probably already decided for you; tempo is one of the defining characteristics of genre. Here is a list of common tempo ranges for a number of electronic genres:

- Dub: 60-90 bpm
- Hip-hop: 60-100 bpm
- Downtempo: 90-120 bpm
- House: 115-130 bpm
- Techno/trance: 120-140 bpm
- Dubstep: 135-145 bpm
- Drum and bass: 160-180 bpm

Note that anything genre-related is not an exact science, and for each of these genres there are likely to be many exceptions and outliers. But as a general rule, if you're deliberately working in one of these genres,

choosing a tempo within these ranges should work.

So how do you decide on a more specific tempo? This is largely dependent on personal preference and musical context. Start with something in the middle of the range, but then try occasionally changing the tempo while working on the track and see if something just feels "right." You can also try some of the techniques suggested in Deliberately Bad Listening to see how the tempo feels in unusual monitoring contexts. Sometimes, a tempo will feel decidedly too fast or too slow when you listen from a different room, for example. And although the difference between, say, 124 and 125 bpm may be subtle, you may find that one is clearly preferable to the other for a particular context. If you don't get that sense, however, then don't spend more time than necessary worrying about it. Simply choose one and move on.

One interesting phenomenon that's been happening in some recent dubstep and drum and bass is a sense of continuous or fluid shift between half-time and double-time. Because the range of tempos in these genres is relatively fast, music at half speed may sound appropriate when layered with music at the "actual" tempo. A drum and bass track that is predominantly at 170 bpm, for example, might work perfectly well with a sampled breakbeat at 85 bpm. This slow/fast dichotomy allows for a lot of creative flexibility when working in these genres.

And of course, there are no rules when choosing tempo: The correct tempo is the one that makes the song work the best, and this is a creative decision only you can make.

One Part at a Time

"Do not worry. You have always written before and you will write now. All you have to do is write one true sentence. Write the truest sentence that you know."
— Ernest Hemingway, A Moveable Feast

Problem:

From the perspective of a blank slate, the idea of "a great song" seems overwhelming and very far away. How can you possibly get from nothing to a complete package—a finished song at the level of the music that inspires you?

You know that songs are made up of lots of parts, and the interplay of those parts is what makes music interesting. But how can you get an interesting interplay of multiple parts when you currently have nothing at all?

Solution:

The expression "a journey of a thousand miles begins with a single step" is clichéd but also totally applicable here. Thinking about a finished song from the perspective of a blank project is terrifying, and it's a recipe for paralysis.

So instead, forget about the song for now, and begin writing just one part. Of course, this is obvious: To begin, you have to begin! But treat this one part with the same kind of respect that you would a finished song. Instead of treating it like a building block, a task to accomplish, or a means to an end, try to focus on it as the goal itself. Before you even begin to think about adding more elements, make that one part amazing.

For example, imagine you'll begin with drums (as many producers do).

Instead of quickly sketching a pattern with stock sounds and sloppy timing before moving on to a bass line (also quickly sketched), spend the time necessary to completely finish the drums before adding any other elements.

If you've read other music-making tips (including the Breadth Before Depth chapter in this book), this may seem counterintuitive. We're commonly told to write by sketching quickly and broadly—many parts, but in little detail—and then treat refinement and editing as something that is done in passes over all of this material. One Part at a Time proposes the opposite approach; instead of working broadly, work deeply on one element—and only one element—until it's to a level that you would consider release-ready.

This way of working can be valuable for a number of reasons:

- **It helps you to develop the skill of focus.** A single part can be thought of as a tiny version of a complete song. To be successful, both have to satisfy the same basic criteria: They should be sonically (technically) proficient, they should flow well over time, they should remain interesting upon repeated listening, etc. Meeting these criteria over the course of minutes and with multiple tracks/parts is our ultimate goal, but if we can't make this happen over the course of seconds and with a single part, we're probably doomed. Making yourself finish something small helps you to learn what finishing feels like and helps you to internalize the steps necessary to get there. From there, finishing something larger is just an expanded version of the same process.
- **One great part can reveal more great parts.** Although what's being suggested here is to follow one idea to its end, another positive side effect of this working method is that one idea (especially one great-sounding, inspiring, and completed idea) can lead to others. Hearing a great drum part, with great sounding samples, perfectly placed notes, and attention to every detail, might make you start to imagine complementary bass lines, chords, or melodies.

Of course, it's important to remember not to be too strict in your thinking. (This is true for everything in this book!) If you find that you're getting bored with working on drums and are inspired to work on a bass line instead, follow your inspiration and work on the bass line.

But if you're bored with working on one part and aren't inspired to work on another one, try applying this technique as an exercise in discipline alone. And if this still isn't working, consider the possibility that the part is unsalvageable—sometimes the best solution is to start over.

Different Tools

Problem:

Your studio tools are like a second home. You've become so comfortable with a particular set of possibilities and workflows that you're able to easily get music made.

But you're not really happy with the results. Your work tends to follow predictable formulas, and you never seem to be able to expand your creative reach. You're pretty sure that your next track will really just be a variation of your last one. And although you're getting songs made, you don't feel like you're really making creative progress.

Gaining a comprehensive understanding of a focused and limited set of tools can be a real creative benefit. Many highly organized electronic musicians are good at resisting gear lust and instead learn a particular DAW inside and out. These same producers are usually equally good at limiting their acquisition of plug-ins and virtual instruments, and instead use either the native devices in their DAW or select plug-ins and/or a small collection of boutique hardware.

This strategy is powerful. It discourages activities like gear experimentation, which feel like music-making but which yield no results. And at the same time, it encourages deep understanding of the tools, which can help eliminate the need to reach for a tutorial in the middle of the creative process.

But these self-imposed limitations can come with a hidden cost—limiting yourself to a concise set of tools may allow you to fall into the mindset of how those tools "think." You may subconsciously begin to make the music suggested by your equipment rather than making the

music you really want to make.

Solution:

Even if you're a seasoned veteran with years of experience and carefully thought-out reasons for choosing your studio tools, try making a track using a completely different set of tools. Try changing everything, from your DAW to your plug-ins to your hardware instruments. This is essentially the opposite approach to Mise en Place—rather than carefully building a working environment, you're deliberately discarding one you've already built.

The goal here is to break yourself from routines you may not even realize you have. For example, if you've built templates or default presets in your DAW, no matter how much you change them in the course of writing a new track, you're still beginning from the same fixed starting point every time. By completely changing your tools, you're forced to work without these safe havens.

Additionally, you're forced to understand the mindset of these new tools. If you enter an unfamiliar creative space, your response to this lack of familiarity might manifest itself as a new creative direction.

For example, if you're a software-only producer, try making a track using hardware synthesizers. Although software instruments can have a strong and unique sonic character, our interaction with them is fundamentally the same from instrument to instrument—we navigate each parameter using the same "controller": a mouse or, at best, a generic hardware controller which is designed to be versatile rather than being focused.

Hardware, on the other hand, offers a physical connection that has a

unique physical identity which, in the best cases, is designed in tandem with the sonic identity of the instrument. A synthesizer from one manufacturer uses controls that feel different from those found on a synthesizer from another manufacturer. At some level, this tactile difference must influence the way we approach each instrument.

Furthermore, every tool offers not only its own unique set of possibilities but also its own unique set of limitations and constraints. If you're already comfortable with a particular set of tools, those limitations are well known to you, and you've probably stopped thinking about how they affect your daily work. But by exploring a new environment and following its constraints where they lead you, you'll be forced to make a new set of creative compromises—and, in turn, open a new set of creative possibilities.

Presets as Starting Points

Problem:

You can quickly make music using presets, loop libraries, sample packs, etc. But you're not necessarily comfortable with this approach and are unsure about whether or not it's "cheating."

There is a lot of disagreement and strong opinion about whether or not you can really take credit for your original music if you're not also programming your own sounds. Some electronic musicians are also highly skilled synthesizer programmers and/or sample manipulators, and they talk with pride about how the first thing they do when they get a new piece of equipment is erase all of the presets. Often, the underlying philosophy behind this approach to composition emphasizes sound as the principal parameter of the music. Many musicians who focus intensely on sound design also work with a fairly restricted harmonic, melodic, and rhythmic palette. For them, it is timbre choices that largely define their artistic signature. This is common in, for example, minimal techno and some experimental styles.

On the other end of the spectrum are electronic musicians who have no interest in sound design and instead create their music entirely by assembling existing loops and samples. For these musicians, the focus is generally more on harmony, melody, and rhythm, and less on timbre. This is not to suggest that the sounds are arbitrarily chosen or used without care, but rather that there may be a range of acceptable sounds for a particular song part, and that the notes are given more compositional attention than the sounds. This is common in, for example, trance and some house music. In an abstract sense, this is also common in more conventional genres such as classical music or rock. In these genres, the available selection of instruments (and thus the

essential palette of sounds) is generally a defining characteristic of the genre, and the composition is then largely about using those instruments in the service of melody, harmony, rhythm, and form.

Which approach is the right one? Perhaps neither. Consider, instead, a middle ground between the two that can maximize musical productivity while minimizing the feeling that you're not taking enough ownership of your music.

Solution:

Instead of approaching sound design as an all-or-nothing endeavor, use presets, samples, and loops as the point of departure for each song part, but with the implicit understanding that you will devote some part of your music-making time to fine-tuning the sounds to suit your particular taste and needs. Purists may always start from a synthesizer's default preset, but this feels like an unnecessary and time-consuming restriction. It means starting from the same state of controls, regardless of intended musical context, and then gradually tuning the sound "outwards" to gain distance from the default. Instead, consider beginning the writing process by using a preset that is at least in the same family as the instrument you're writing for. If you're working on a bass line, for example, explore your synthesizer's bass presets—guilt free. Once you've found one that's as close as possible to the sound you imagine in your head, work on the other parameters of the part: the melody, rhythm, etc. Then, after you have some music underway, come back to the sound again and begin tweaking it so that it takes on your own unique signature.

For musicians with absolutely no sound design experience, this may sound like an intimidating process. But once you learn just the very

basics of synthesizer theory, you'll find that this knowledge can be applied to almost any synthesizer. For example, filters and basic ADSR envelope controls can be found on almost any hardware or software synth or sampler. And by using nothing more than these parameters—this is usually no more than six total controls—you can come up with a huge range of variations from a given preset.

Commit at least enough of your music-making time to sound design so that you can be proud of the sounds you use. For the truly discerning electronic musician, it's unlikely that any preset will ever be the perfect choice for their particular musical context. But it might be close. From there, you may need only a few tweaks before you've found something that is uniquely your own.

The Tyranny of the Default

Problem:

Every time you're inspired to start a new song, you open your DAW and are immediately terrified by the blank project. Maybe you have a simple melody, bass line, or drum part in your head. But in order to hear it, you first have to load the appropriate instruments, set the tempo, maybe connect a MIDI controller, etc. By the time you've gotten your DAW to a state where you can actually record, you've either lost the motivation or you've forgotten your original musical idea.

Because DAWs have to cater to a wide range of users, they are often designed to work out of the box with a collection of default options and a basic screen layout that will be offensive to no one but probably also not optimal for anyone. This inevitably leads to a phenomenon that software developers call "the tyranny of the default": Since most users will never change their default software options, the seemingly small decisions made by developers may have a profound effect on the way users will experience the software every day.

Here's how to overcome the tyranny of the default in your own studio.

Solution:

Rather than allowing your DAW to dictate the environment in which you'll start each track, take the time to build your own default or template project. People often think of templates as blank slates containing a bare minimum of elements, and most default templates provided by DAWs are exactly that; maybe one or two empty tracks, perhaps a return track containing a single effect. But if you regularly start work in a similar way (and even if you don't), building a template

that's unique to your musical preferences and working style can save you lots of time when you start a new song, allowing you to more quickly capture an initial musical idea from your head into your DAW.

For example, many DAWs set a default tempo of 120 bpm for new projects. If you tend to work in a genre that is generally in a different range of tempos (see Choosing a Tempo), save yourself time by saving your template with a more appropriate tempo. Additionally, your DAW's default project likely makes a lot of assumptions about how many output channels you'll be using (usually two), as well as more esoteric settings like sample rate, bit depth, and even the interface's color scheme. If you prefer different settings, don't change them every time you start a new song. Instead, make these changes once and save them in your own template.

Additionally, if you regularly use a particular collection of instruments and/or effects, try pre-loading them into tracks in your DAW and saving them into your template. If you have a go-to sound that you use for sketching out ideas (maybe a sampled piano or a particular preset), preload that preset in your template and even arm the track for recording. This way you can be ready to play and record as soon as the project is loaded.

Some DAWs even allow you to create templates for different types of tracks. For example, if you regularly use a particular combination of effects on each track (such as a compressor and EQ), you could preload these devices—and even customize their parameters—into your default tracks. Then each time you create a new track in any project, you'll have these effects in place without needing to search through your library of devices.

If you regularly work in a variety of genres, you should consider making

multiple templates, each one customized for the different sounds and working methods you prefer. Even if your DAW doesn't natively support multiple templates, you can still create your own collection; you'll just need to remember to Save As as soon as you load one, so you don't accidentally overwrite it.

Some producers, recognizing the value of a highly customized template project, have even started selling templates containing nearly (or even completely) finished songs, with the stated goal that newer producers can use these to learn the production techniques of the pros. If that's really how you intend to use them, then these are a potentially valuable learning resource. But be careful to avoid just using these as "construction kits" for your own music. This is potentially worse than working from an empty default and is a gray area between original music and paint-by-numbers copying (or worse, outright plagiarism).

Simple Sounds

Problem:

You have a melodic, harmonic, or rhythmic idea in your head. But when you sit down at your DAW to try to capture it, you usually begin by trying to find or create the right sounds to match the idea. And somewhere along the way, the idea itself gets lost.

As electronic producers, often our first instinct when starting a new song is to collect a palette of sounds before beginning work on the actual notes and rhythms. And modern synthesizer presets are often designed to sound like entire productions just when playing a single note, with huge layers of effects, embedded rhythmic activity, or both.

But sometimes, this can be a distraction from your real goal, which is to create your own compelling musical ideas.

Solution:

Instead of starting by trying to find the perfect sounds, try starting with the simplest sounds you can find. General MIDI or other "generic" presets are good for this exercise.

Now, without taking time to tweak the sounds or add effects, just start working on notes and rhythms. It's likely that everything will sound uninspiring at first, and this is OK. The hope is that by working this way for a while, you'll come across musical ideas that are strong enough to transcend or overcome the sounds that play them. These ideas will have to be really good, since they won't be able to hide behind impressive synth patches or layers of effects. But really good ideas are what we're after—musical materials that are so strong that they inspire you on their

own merits. Consider that whole symphonies have been written by composers who work entirely at the piano.

Once you have a great collection of self-sufficient musical ideas, then you can start the process of finding great sounds to play them. As you were doing the hard part of the job—writing the music—you may have already had ideas about what kinds of sounds would be appropriate. This will make finding or creating the appropriate sounds even faster and more rewarding, because you'll know that they fit your own musical ideas, rather than suggesting musical ideas that may be more a function of the inherent sonic properties of the sound.

Although generic electronic sounds work well for this, an even better solution is to write using an acoustic instrument like a guitar or piano (if you play one). Acoustic instruments serve dual purposes here. As with General MIDI sounds, they help to get you out of sound-design thinking so you can focus on the music. But they also help you get out of DAW thinking entirely. By removing the computer from the picture for a while, we're more likely to avoid distraction and really force ourselves to write with our ears instead of our eyes.

Of course one challenge of working acoustically is that it makes it a little bit harder to capture great moments of inspiration that would otherwise automatically land in your DAW. In this case, it might help to set up a mic in the room and record everything you play as audio. Yes, it might take a bit more work to then sort out what you played later, but at least the ideas won't be lost.

This technique isn't for every producer. If your music is fundamentally about sound design, then it's possible that notes and rhythms aren't much of a factor in your work at all. In such cases, these techniques will probably just waste time. But if you're a producer working in a less

experimental genre, starting with simple sounds might help you get to better music.

Extended Techniques

Problem:

You're generally bored with the sounds that you've been using. Your real-world samples all sound too real, and your electronic sounds all sound too conventionally electronic. You're looking for something that bridges the gap between the rawness of acoustic sounds and the flexibility of synthesis. You've tried using effects to process your sounds but this just feels like adding a fresh coat of paint. You need something really new.

Sampling from the acoustic world brings a layer of "realness" to electronic music productions that is difficult to achieve using synthesis alone. And although samplers and effects can help to turn real-world sources into something radically different, here are some lo-tech ways to get entirely new sounds from acoustic sources.

Solution:

Although it's always possible to use samplers and effects to modify source sounds into something completely different, it can also be inspiring to work with a sound that's already unusual on its own. Field recordings, found sounds, and other types of non-instrumental source material have long been used as fodder for samplers, but there is also a world of new sound possibilities available from instruments you already know.

The term "extended techniques" refers to unconventional approaches to sound production using conventional instruments. Probably the most familiar extended technique is the "prepared piano" made famous by John Cage (among others). Cage's scores for prepared piano call for

objects such as bolts and rubber erasers to be placed on or between the piano strings, or on the hammers or dampers. The resultant sound varies widely depending on the particular type of preparation used, but often sounds like a diverse collection of metallic percussion instruments.

Another extended technique that can yield great results is the use of stringed instrument bows on other objects. A lot of contemporary percussion music, for example, calls for bowed vibraphone bars, cymbals, gongs, and other metal objects. In fact, almost any piece of resonant metal can be activated via bowing. If you have access to a bow, try bowing pots and pans, metal furniture, etc. And if you don't have access to a real bow, you can create makeshift bows out of fishing line.

There are a variety of possible extended techniques for woodwind and brass instruments. If you have access to these instruments (as well as adventurous people who can play them), try playing them without their reeds or mouthpieces, playing only the reed or mouthpiece, or even using the mouthpiece from one instrument on another "wrong" instrument. In some cases, this will produce no sound at all, but you can sometimes get very interesting timbres by resonating these instruments in new ways.

Woodwind and brass instruments are also capable of producing multiple simultaneous pitches, which are known as multiphonics. For woodwinds, multiphonics are produced by using certain unconventional fingerings, while on brass instruments they can be produced by humming additional pitches while playing normally.

In recent years, the concept of extended techniques has moved beyond the world of acoustic instruments and into the electronic domain as well. The practice known as circuit bending is the physical modification of electronic sound-making devices with the goal of creating sounds

those devices were never intended to make. Common circuit bending targets are cheap electronic instruments, children's toys, etc. Circuit bent instruments can be a great source of new timbres, particularly because they often produce very unusual sounds while still retaining the lo-fi character of the original device.

Goal-less Exploration

Problem:

You're sitting in front of your DAW, but you're uninspired. You've opened the DAW out of habit, or because you know you should, or because you have the abstract feeling of wanting to make music, but you have no real sense that you actually want to make anything in particular. This feels frustrating, and like wasted time. Maybe you should just close the DAW and try again another day.

Many of us are happy once a song is finished, but often we find the process of actually making the song to be painful, boring, or unrewarding. Sometimes we might reach a point where we see that the end of the song is in sight, and then we're inspired by the prospect of having something to show for our work. But until that point, it really does feel like work. How can you generate inspiration? Just opening a new project sometimes isn't enough.

Solution:

If you've already taken the step to open your DAW, you should congratulate yourself; this is farther than many musicians got today. Our computers (and even the real world) offer endless distractions that can prevent us from even beginning work. And although our underlying goal is to get music made, sometimes this is too much of a responsibility to bear. Even though you don't feel like working, once the DAW is open, there are a number of things you can do to seize the moment, even if it's a brief one.

Goal-less exploration is the process of simply finding a corner of your working space and letting yourself see what evolves from there. For

example, maybe there's a particular instrument or effect in your DAW that you've never taken the time to learn or that you've tried to use but never could find a place for in your music. Since the DAW is open anyway, try loading these devices and just experimenting with them. Without the pressure of trying to use them to make music, just spend some time learning what sorts of sounds they can make.

Another type of goal-less exploration can come from taking some time and going through your loop and sample library. If you're like most producers, you've probably collected far more samples than you've ever had time to really listen to. If you don't feel like working on music, explore those folders of samples and see what you actually have.

Working in this kind of free, non-directed way can sometimes unlock creative directions that you never would have found in your normal working process, when you might be tempted to use templates or known tools in order to get going more quickly. If goal-less exploration actually manages to turn into real, finished work, then all the better. But you're also not under any obligation to turn your explorations into a finished song, or even the beginning of one. Maybe you'll work for a long time and find that you've created nothing worth saving. This is OK. The point here is that you're doing something that is at least in the realm of music-making; this is already closer to where you want to be than, for example, Facebook.

There are so many things you can do in a DAW that aren't directly focused on the creation and completion of a song. If your DAW is open, but your normal creative inspiration isn't there, try simply exploring and see where it leads.

Procrastination and Timeboxing

Problem:

In the abstract, you know that you want to make music. But the actual process of doing it sometimes feels like torture.

You know that working on music can bring moments of bliss: When the work is going well, the music sounds great, and you're in a state of flow. But there seem to be as many or more moments of agony. Despite our best intentions, there are lots of reasons why we sometimes procrastinate, including fear of failure, fear of success, and simple laziness.

In the chapter called On Work, we discuss the inevitable reality that working is the only way to actually make progress. But here is a tip for actually getting to work.

Solution:

If you're a chronic procrastinator, you're not alone. Many creative (and non-creative) people suffer from task aversion and can find any excuse to avoid getting done the work that really needs to get done. One strategy for overcoming procrastination that's commonly used in the software development world is known as timeboxing.

Timeboxing simply means setting a fixed amount of time for a particular task. The amount of time you choose is up to you, but it should be short enough so that it's easily manageable by even the most determined procrastinators. For example, you might decide to spend 20 minutes on sound design. Next, set a timer, work only on sound design, and stop when the timer goes off. Finally, take a short break (five minutes or so).

Then repeat the process, perhaps with another unrelated type of task (e.g., drum programming, arranging, etc.). After four or five stretches of this timed work/break combo, you might want to take a longer break.

Short timeboxes work because they break apart intimidating, open-ended tasks into easily manageable chunks; no matter how painful creative work is, anyone can do it for 20 minutes. It's important that you really work during those 20 minutes—intensely and without interruption. But it's equally important that you stop at the end. No matter how productive you are or how close you feel to entering a real flow state, stick to the timebox. This may seem counterintuitive; after all, if you've finally managed to trick yourself into enjoying the process, doesn't it make sense to run with that as far as your attention span will allow? But the reason this is dangerous (at least at first) is because it runs the risk of setting you up for a long work session that eventually becomes frustrating or disappointing. When this happens, your memory of the session will be that it was both bad and long-lasting, which may further reinforce your procrastination tendencies. On the other hand, if you stop even while it's fun, you're more likely to be energized and ready to move forward again once the break is over.

After you spend a few days sticking to a systematic timeboxing routine, you may find that your latent procrastination starts to feel less overbearing, and you feel the urge to just do uninterrupted work. At this point, you might first try using longer and longer timeboxes. Eventually, you may find yourself able to focus without assistance at all and can then consider your procrastination cured. And if you find yourself slipping back to your old work-avoidance habits later, simply try using the timer again.

Collaboration

Problem:

You're considering the possibility of working with other musicians but aren't sure how to proceed. What do you look for in a collaborator? What should their role be in the creative process?

Musical collaboration can be a hugely rewarding catalyst for creativity, in which the resulting music is more than the sum of the individuals who made it. But establishing the right collaboration is critical; working with the wrong partner can be more painful than working alone. Here are some tips for finding great musical partners.

Solution:

When deciding on a partner for musical collaboration, the most important consideration is simply whether or not the partnership will result in better and/or more music. It's tempting to imagine that your best friend will also be your ideal musical partner, but personal compatibility and professional compatibility aren't necessarily related. Sure, the process will likely be more fun if you're working with someone with whom you have a good social connection. But many personal relationships have been destroyed by attempting to extend them into professional relationships; if the professional one fails, the personal one might as well. If your best friend happens to also have a strong work ethic and great musical and technical skills, then the partnership might make sense. But if not, consider partnering with someone else. Of course, even the most talented musician isn't going to work out in the long run if they're a horrible person. But when in doubt, aim for professional compatibility first and social compatibility second.

Whenever possible, try to work with people who can counterbalance your strengths and weaknesses. The most obvious example is the classic "producer plus vocalist" partnership. The producer deals with all of the musical and technical aspects of the song, while the vocalist brings a particular talent that the producer simply doesn't have. Beyond this obvious example, try to find partners who are good at a particular aspect of the production process that you're bad at or that you dislike. For example, if you have trouble with arrangements, try to find a partner who excels at them. If you're good at sound design and synth programming, don't worry too much about finding a partner who also has these skills (although two completely well-rounded collaborators might be the ideal package).

Although the traditional model of collaboration involves multiple musicians in the same studio, recent technological advancements also make it possible to collaborate with people remotely, and possibly even anonymously. For example, the classic model of the studio musician has changed in recent years, and many session musicians who used to show up for recording studio dates are now tracking parts for a variety of projects from their home studios, sometimes without ever meeting the artists for whom they're recording. This type of workflow is even easier for electronic musicians, because there's no need for acoustic treatment, microphone setups, etc. For example, perhaps you have an idea for a synth part that's beyond your ability to play. Using remote collaboration tools, you might find someone to record the part you need and deliver it back to you, entirely online.

Once you've found a good collaborative situation, it's important than everyone involved understands exactly how things like workload, money, ownership, and creative rights are to be divided. Misunderstandings can result in a lot of misery for all parties, so it's

best to have clear conversations and real, mutual understanding as early as possible.

Thinking Like an Amateur

Problem:

Ever since you started getting "serious" about your music-making, you've secretly started having less fun with it. You (vaguely) remember a time when you didn't feel this way; when you had no aspirations to be a professional, making music was always a great way for you to relax. But now that you've become concerned with "success," your sense of childlike joy at just making sounds has disappeared and has been overtaken by your desire to *finish songs*—or even scarier: *get famous*.

Although this book is mostly about finding ways to get music done, sometimes there are benefits to forgetting about that way of thinking entirely and instead just enjoying the process of making music as a goal in itself.

Solution:

When someone is referred to as an amateur, this is usually meant to imply that they're less qualified or less talented than a professional in the same field. An amateur, it's assumed, is someone who would have liked to be a professional but who was unable to reach that level. But despite these negative connotations, the word "amateur" actually just means "lover of," and there are many amateurs in all fields who are working at a very high level. And there are still many more who aren't necessarily great at what they do but are having a great time.

Think about something you consider a hobby, something (besides music) that you do with your free time. Maybe you run marathons, or brew beer, or take wildlife photographs. Whatever it is, have you ever even considered doing it professionally? Probably not. And most likely

this isn't because you're not good enough (and whether you are or not is probably irrelevant to your decision), but rather because the very fact that it's a hobby means that it's something you do that isn't work. Instead, it's a chance to spend time on something fun and fulfilling that doesn't saddle you with any outside pressure to succeed, earn a living, etc.

Electronic musicians, more so than musicians working in other genres, seem to have a more difficult time simply engaging with music as a hobby. Perhaps this is because tools like DAWs are fundamentally designed around a recording mentality. Think about people you've met who own an acoustic guitar. Just pulling it out and playing it for a few minutes while sitting on the couch may be the extent of their musical aspirations. And they don't see this as failure. They're not lamenting their inability to get gigs or write more music or get record deals. They're having exactly the relationship with music that they want. In fact, they're usually not even recording what they play; once it's in the air, it's gone.

By definition, being a professional means having to spend at least some amount of time thinking about the marketplace. Is there an audience for the music you're making? If not, you're guaranteed to fail. Amateurs, on the other hand, never have to think about this question at all. This frees them to make music entirely for themselves, on their own terms.

One easy way to do this is to put yourself into a musical context in which you actually are an amateur—by experimenting with a genre in which you have no prior experience. Are you a committed hip-hop producer? Try making a house track. Your expectations are bound to be lower, simply because you have no prior successes or failures against which to gauge your current work. Even if you hate the results, it's likely that you'll learn something from the experience.

Even if you do aspire to make a living out of creating original music, it might be helpful to think like an amateur in order to lower your stress and bring the fun back to your music-making time. Amateurs often have a genuinely more pleasurable experience than professionals working in the same field, and this is almost certainly because they're free from outside pressure. If you can instill this mindset into your own work, you'll probably have both better results and a better time.

Scraps and Sketches

Problem:

Your hard drive is filled with unfinished songs, and you know that most of them aren't worth finishing. Whenever you look at these songs, you're frustrated because you feel like you've failed and that you have nothing to show for all of the work and time you've spent on them.

We abandon music for a lot of reasons. Sometimes we realize that we're not happy with the direction the song is taking. Sometimes we just get bored, go do something else, and then never come back to the work in progress. While I'm a strong advocate of finishing bad songs anyway (see Fail Better), the reality is that most producers either have folders full of abandoned projects that will never be opened again, or they throw away work in progress if it stops being interesting. Here's an idea for making use of that unfinished material.

Solution:

Rather than considering your folder of unfinished projects as trash or wasted effort, consider it as a collection of scraps or sketches, ready for reuse in other musical contexts. This way, you can start a new song by extracting elements of abandoned or unfinished songs from the past, rather than facing the intimidating blank slate of a completely empty project. Forget entirely that these were once supposed to be "songs," and instead treat them as collections of musical ideas that are ready to be used again: your own custom—and completely unique—sound and loop library.

You may already be keeping a folder of scraps and sketches, either as a result of Goal-less Exploration or perhaps by collecting rejected audio

files created by Rendering as Commitment. The only difference here is that you're rehabilitating and recycling material that you once intended to be used in an actual, active composition. This can be incredibly liberating mentally, because it allows you to reject the idea that an unfinished song is a failure and instead embrace it as a collection of ammunition for the next song.

So throw away as little as you can. Even if you're bored after a single two-bar drum loop or bass line, save this work into your sketches folder. It's possible that it will become the basis of your next great song.

Problem-Solving as Distraction

Problem:

Every time you go to make music, you realize that there's something wrong with your setup; a plug-in you need isn't authorized, or you forgot to download a sample pack that you purchased, or your chair isn't comfortable. You dream of the day when you can hire a studio assistant to take care of all of the busy work that seems to be in your way every time you want to make music. But until then, you're not sure how to get anything done when all of these problems keep coming up. How do other people do it?

Technical problems in the studio can really kill the creative flow. Similarly, realizing that you don't fully understand a particular production process or how to use a particular piece of gear can be frustrating—particularly when you had planned to use those things in the course of your current song. When faced with these situations, here's how to get work done anyway.

Solution:

Ignore these problems. They're not as important as you think.

Steven Pressfield, in his classic book The War of Art, talks about a force that manifests in the life of creative people in order to keep them from getting work done. He calls this force Resistance. These problems you're finding in your studio are a manifestation of Resistance. It's a trap! Your mind is playing tricks on you to try to keep you from working. None of these problems are serious enough to take time away from your creative efforts. If they were, you would have solved them already, during your non-productive time. The only reason you notice

them when you sit down to work is that part of you doesn't want to sit down to work.

Technical problem solving, education, motivational reading (including this book), and other forms of work-related activities will never write music for you. They're important, but as discussed in On Work, they're nowhere near as important as actually doing the work.

Your creative time is essential; you should treat it with care and recognize when you're subconsciously trying to avoid it. If you suddenly realize that a plug-in you need isn't authorized, then use another one. You don't need it right now. You've made music without it before, and you can make music without it now. If you forgot to download a sample pack you purchased, use some samples you already have. If your chair isn't comfortable, work anyway. You can fix your chair on your next break or when the song's done (whichever comes first).

Nothing seems to stimulate the need to learn production techniques like sitting down to work. Do you really need to learn more about EQ or compression right now? Watch tutorial videos later; now is the time to work.

This is not to suggest that learning and problem solving aren't important. It's just that they belong to a different type of working time. The chapter Mise en Place discusses the importance of setting up your environment to be maximally conducive to getting work done. But as important as this is, it's even more important that it be done at the right time. And the right time is never when you're in the creative mindset.

Music-making time is sacred. Don't stop to learn and don't stop to fix something unless it's absolutely critical. Use what you have and know,

right now, to keep going forward.

Personal Workflow

Problem:

The more you read about music production techniques, the more you get the sense that you're doing everything wrong. For example, some tutorials advocate working on sound design, composition, arranging, and mixing in that order and as entirely separate processes. Others recommend doing them all at the same time and working each section of the song to completion in all aspects before moving to a different part of the song. But none of these solutions feels quite right, and now you're more confused and discouraged than you were before you read anything. What is *really* the optimum workflow?

Reading multiple tutorials or books about music production can make you feel suspicious that it's all just subjective. There seems to be so much disagreement from one source to the next that it all feels somewhat arbitrary. Who has the right answers?

Solution:

When it comes to optimal workflow, there is no objective right answer. The only way you can find a workflow that's optimal for you is by trying out various methods, keeping the parts that work, rejecting the parts that don't, and synthesizing your own solution out of what remains. When you're reading a tutorial, you're really reading about a workflow that has worked for that particular writer. A different writer may have had an entirely different experience, and so will have advocated an entirely different approach.

If you have trouble getting music done, it's worth looking at the workflows that other people advocate and giving them a try. But don't

expect that any one person's approach will yield better results for your particular case than an approach that you develop yourself. It's true that when trying out another method you may see a dramatic shift in your productivity (either positively or negatively). If things are better, then it's likely you've found a direction that's better than the one you were on before. If things are worse, you should probably abandon this direction and try something else. But in the end, even the best third-party workflow will benefit from your own personal tweaks and modifications.

Once you have the sense that you're going in the right direction, the key to optimizing your personal workflow is to practice it until it becomes a habit. Practice the steps just as you would practice an instrument, refining and adjusting as you go. Eventually, the workflow will stop being work and just become flow—the things you do unconsciously while creating.

This all sounds quite abstract, but there really aren't so many possible things to consider. At a fundamental level, music production workflows all involve answers to the following questions:

> What are the things I need to do?
> Is there an order I should do them in? If so, what is it?
> When should I move on to the next thing?
> When do I know I'm completely done?

Once you understand how to answer these questions—for yourself, not as absolutes—you will have developed your personal workflow. All that's left to do is make music with it.

Originality vs. Quality

Problem:

You're happy with the music you make, and other people are as well. But deep down, you don't believe it's truly original, and you feel that you should be working harder to find a sound that is uniquely your own.

Particularly among music critics, journalists, and bloggers, originality is regularly cited as a necessary component of good music. Pejorative terms like "derivative" or "formulaic" are often used to deride music that sounds like something else. But what critics write is not always what listeners hear, and there may be different ways of weighing musical value.

Solution:

What does originality actually mean in a musical context? As a descriptor, all it tells you is that something sounds unique, or that its influences are not obvious. But what the word doesn't convey is any information about quality, which is a far more valuable characteristic. "Original" and "good" are not mutually exclusive, but they are also not correlated. There is plenty of terrible music that sounds like nothing else, and in many cases the very characteristics that make it original are also what make it bad. Consider that what one artist thinks is original may simply be something that other artists have already tried and discarded.

Instead of aiming for an abstract goal like originality, aim instead for the concrete goal of quality. This is not to suggest that you should actively try to make your music sound like something else, but rather that you should try to find and strengthen your own voice through a process of

constant, varied listening, experimentation, synthesis of disparate influences, and—above all—hard work. If you only listen to a small variety of music, there are really only two conceivable outcomes for your own creations: Either you'll make things that reflect your limited exposure or you'll make things that you believe are original but are more than likely simply outside of your experience.

Even if you really only want to make music within a particular, narrow genre, you should still aim for a wide range of listening experiences. For example, imagine you only want to make old-school house tracks. Since a characteristic of that style is sampled phrases from jazz, soul, and R&B records, wouldn't an understanding of those genres be a valuable tool in your creative arsenal? Wouldn't understanding how jazz drummers and bass players approach rhythm and harmony help your own rhythm and harmony? And if you end up making old-school house music that fits perfectly into the confines of the genre, no one can claim that you're original. But if the music is good, that's all that matters.

In the end, the music that's remembered is not necessarily the music that's the most radically avant-garde. Instead, it's the music that's good. If you can achieve both quality and originality, then you're a rarity. But the primary goal should always be quality first.

Process vs. Product

Problem:

Often we read interviews with interesting musicians in which they talk about the monumental effort that goes into their creative work. Similarly, they may talk about how their music is rigorously structured or adheres to particular types of architectural processes that are carefully and painstakingly planned and then require immense work to carry out. When we then go to work on our own music, it may feel like we're "under thinking" or not investing enough effort. We may begin work but then stop and second-guess what we've made if we don't feel like we've done enough preparation to allow us to really begin "properly."

The idea of "process" in music refers to the development of a system or set of rules that allows some or all of the elements of the music to be generated or derived without requiring the composer to make purely intuitive creative decisions at every possible moment. There is a belief in our culture, often reinforced by interviews with celebrities, that there is a necessary and direct correlation between process and quality. For musicians who work intuitively, or via improvisation and experiment, this narrative can be intimidating. How can it be possible to create great work if we don't have a grand scheme and/or long hours of struggle?

Solution:

First, take every interview with a grain of salt. Interviews allow artists to share insights into their working methods, but interviews can also serve a valuable self-promotional function. There is nothing wrong with this —artists should take advantage of opportunities to present themselves. But readers should not expect an interview to reveal absolute truths, and stories about struggles, hard work, and rigorous planning might simply

serve to help an artist create and publicly project a romantic or idealized self-image.

Second, effort, rigor, pre-planning, and structure are not prerequisites for quality results. They may help you get there, but they are neither necessary to begin work nor, in themselves, evidence for quality. It's tempting to believe that things we've spent a lot of time and energy on are worth it, but in the case of music, only the results matter. Some listeners may indeed actively listen for musical structure, and the intricacy, depth, or cohesiveness of that structure may be what determines for them whether or not the music is "good." Likewise, some listeners may enjoy music because it sounds "difficult." Consider, for example, certain types of virtuosic metal or progressive rock, in which technical ability is a prerequisite. But there is no universal relationship between artistic effort and artistic quality. Many listeners take in music on a much more immediate and visceral level and enjoy particular music for reasons which may be more difficult to quantify.

Think of the process of creation as something like the scaffolding on a building. It's placed there to help the builders navigate the unsteadiness of a fragile work in progress. But once the building is completed, the scaffolding is removed and promptly forgotten. It is not, in itself, the thing being made. It may have curiosity value to other builders, who may have an abstract interest in the artifacts of construction. But good scaffolding cannot guarantee a good building any more than a good working process can guarantee a good song.

Furthermore, there is the danger that too much time invested in planning how you'll work will both prevent you from actually working and cause you to accept results that sound bad just because they fit the plan. For example, imagine you've spent hours crafting a four-minute song in which the second two minutes are a literal mirror image of the first two.

This is an interesting idea in the abstract. It provides a path forward: Once you've finished two minutes of music, you now have a system for how to finish two more through the simple application of a formula. But is it a good idea? Only the results can determine this. At the end, the product—not the process—is what you're left with, and the product needs to be strong enough to survive on its own.

This is not to suggest that you shouldn't experiment with employing processes in your music. Maybe a song with a mirrored form really will end up being amazing. And even if it isn't, you may end up learning something about how mirroring works as a musical process that can be applied in different ways in future works. What's important is that you don't fall so in love with the idea of what your music should be (or the work that you've put into it) that you fail to honestly judge it on the basis of how it really sounds when you've finished. Better yet: Listen critically at every stage of the process, and be willing to abandon conceptual directions if they are leading to bad musical results.

There is a rich history of brilliant music made by musicians who started without a grand scheme and simply followed where their experiments or improvisations led them. Furthermore, there is an equally rich tradition of brilliant music that was made very quickly and with seemingly little effort. Again, the important thing is the song—the final result of the work. What's unimportant (or, more accurately, what's not a reliable predictor of quality) is everything leading up to that point: the sketches and careful planning diagrams, the months spent in the studio, etc.

If those things really are the way that you get to a great result, then you should absolutely do them. But do not convince yourself that those things will guarantee a great result.

On Work

“Genius is one percent inspiration and ninety-nine percent perspiration.”
— Thomas Edison

“Inspiration is for amateurs—the rest of us just show up and get to work.”
— Chuck Close

Problem:

Sometimes, there are moments during the creative process when you get into a state of flow, music seems to come out naturally, and everything feels effortless. But this is extremely rare. Most of the time, working on music feels like a thankless chore.

How is it that some musicians seem to be able to produce endless amounts of music in minimal time, while you feel like every note is a struggle? What’s the secret to finding the mythical “flow” state that will enable you to create music effortlessly?

Solution:

The reality is that there are no shortcuts. The process of creating art is, fundamentally, a process of work. For most of us, every note really is a struggle, and this is simply the nature of the process. We have a romantic image of the artistic genius, who is able to spin endless amounts of material out of nothing, but these types of artists are extremely rare.

Rather than aspiring to this mysterious (and unrealistic) notion of what the process should be, try instead to embrace what it really is: hard work. There will be many points along the creative path when you’ll wish you were doing almost anything else. You will be plagued with

doubt, fear, boredom, and disappointment. Learn to be OK with being miserable, because this will be a regular part of your existence as a creative musician.

By all means, if you do happen to find moments of inspiration and real flow, you should embrace them and let them take you as far as they can. But even in these rare cases, it's usually only after a lot of painful effort that effortlessness takes over. Even the most fortunate, flow-prone artists don't just get there immediately. As painful as it is, every project, for everyone, requires real work.

Problems of Progressing

Breadth Before Depth

Problem:

Whenever you start making progress on a track, you suddenly become overwhelmed by the desire to get some particular aspect completely perfect. For example, maybe you're preoccupied with getting the sound of your kick drum just right. Lots of time can be spent in this phase, and it often becomes frustrating, sapping your will to continue working on the track as a whole.

Since you know that you'll eventually have to refine every aspect anyway, can there be a downside to doing at least some of that refinement as you go?

Solution:

Especially in the early and middle stages of your work on a track, when the ideas themselves may not exist yet, it can be detrimental to go into too much depth in any one particular area. Yes, you'll need to do detail work eventually. But the idea-generation phase is vital and very, very fragile. By definition, it's messy and doesn't hold up well when confronted with outside pressures. Idea generation requires experimentation, risk-taking, unbounded thinking, etc. Detail work, on the other hand, is an entirely different kind of working process and requires an entirely different mindset. Detail work requires narrow, focused thinking. It's often more about applying known processes than it is about exploring radical new directions.

When you're in idea-generation mode, it can be useful to work broadly—getting ideas out of your head and into the sequencer as quickly as possible—before working deeply on a single part. This way of working

can be valuable for a number of reasons:

- **It helps you to learn how to listen for potential rather than for perfection.** During broad idea generation, parts might sound bad for a variety of reasons. Maybe you have a terrible mix balance, or the wrong sounds, or even some wrong notes. But you've drawn inspiration from music that has none of these problems—finished, mastered tracks have a professional sheen that's miles ahead of where your particular track is right now. This can be discouraging, because even though we know that we can add polish at the end, we want to hear it right now. If your music doesn't compare to your inspiration at this moment, how can you be sure that you're going in the right direction? The key here is to practice learning to listen past the imperfections: Instead of thinking "this bass line isn't powerful enough," think "this bass line can be powerful enough after some sound design work and mixing. But how are the notes?" By hearing past the immediate lack of impact, you become a better judge of whether or not a particular part (or the whole track) is going in the right direction.
- **Creative time is short, and you have to move fast.** As mentioned previously, the idea-generation phase is fragile. It's the one part of the music creation process that you can't "force" to happen. This means that when it actually is happening, you need to squeeze out every idea that you can, working as quickly as possible and generating as much material as possible before your mind moves out of this phase. Once you've captured the ideas, the work of actually refining them can sometimes require much less truly creative energy.

This suggestion runs directly counter to the chapter called One Part at a Time. The reality is that there is no one way to work; different types of creative blocks may be solved in fundamentally different, and even opposing, ways.

Foreground, Middle Ground, and Background

Problem:

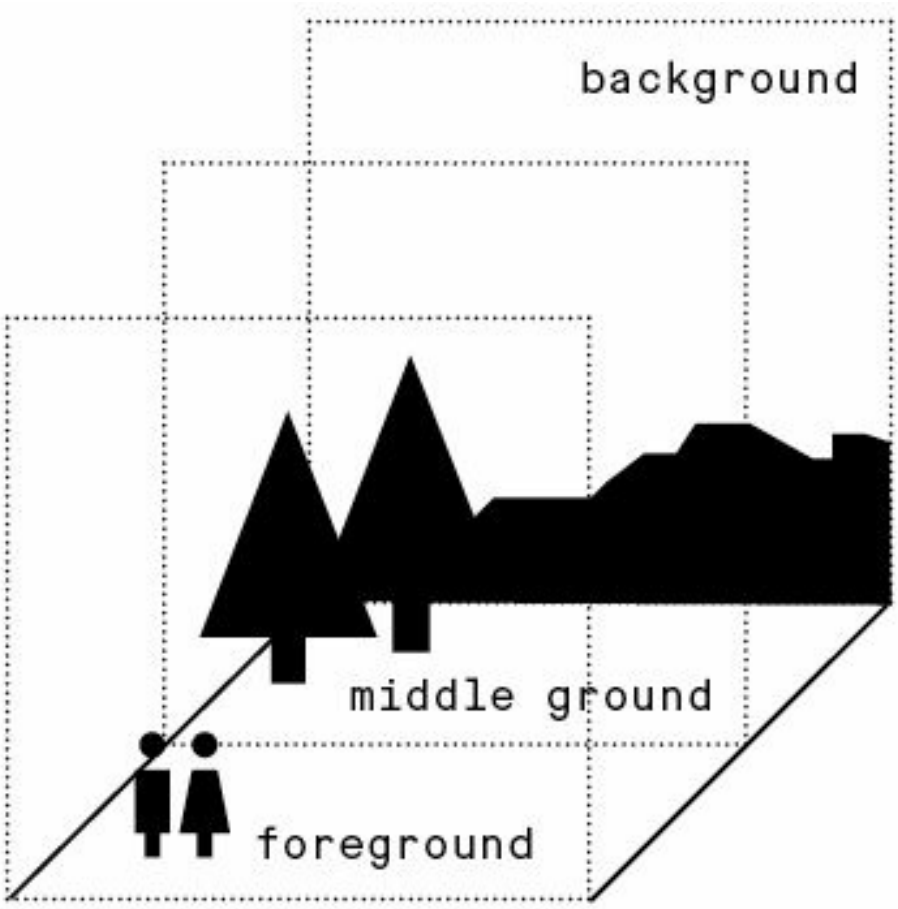
Your music has a feeling of "flatness" or two-dimensionality that is somewhat difficult to explain. The music that inspires you definitely doesn't suffer from this, although you find it equally difficult to express what it is, specifically, that's more satisfying about those inspiring songs.

Typically, we think of different instruments and/or voices in a song as having different levels of prominence. And we probably have some instinctive ideas about how to reinforce these levels of prominence when working on a mix. For example, important sounds are typically mixed louder than less important ones.

But creating a sense of varied "depth" in music is more complicated than just applying basic mixing rules. By thinking about depth even during the composition phase, we can complement our mixing decisions and create an even stronger sense that each element is fulfilling a specific functional role.

Solution:

The notion of foreground, middle ground, and background are well known to anyone who works in the visual arts: film, photography, painting, etc. In visual art, these layers refer to objects at different physical distances from the viewer's eye.



Typically, the objects in the foreground are the ones with the most prominence. In a photograph, for example, foreground objects are commonly the "subject" of the composition. They're often in focus, and we can see the complexity and fine detail. The background, on the other hand, contains objects in the far distance. We recognize them, but they might be blurry or otherwise less rich in detail. The middle ground contains everything else—objects which are generally not the primary focus of our attention but which are closer (and thus more prominent) than those in the background.

At a technical or production level, this concept can also be applied to an audio recording, and this is what mix engineers refer to when they talk about the "depth" of a mix. For example, sounds can be made to seem closer to the listener by making them louder, "brighter" (an increase in high-frequency content in comparison to other sounds), and/or "dryer" (an increase in the amount of original, unprocessed signal in comparison

to the amount of reverb applied to that signal). Likewise, sounds can be placed in the background by making them quieter, darker, or more reverberant.

But even before the mixing stage, we can think of our musical elements themselves as being divided into foreground, middle ground, and background. For example, in a track with a singer, we usually want the voice to be the most important element. We can try to force this to happen when mixing, but we can also make musical decisions to enhance the effect. For example, imagine a scenario in which the vocal melody is doubled note for note by a synth line that plays an octave higher. Our ears are naturally drawn to the highest sound in a mix, so we might subconsciously hear the synth line as the dominant voice. In an even more extreme example, imagine that our accompanying synth didn't double the voice but instead played a line that was more complicated, with faster rhythms, jagged jumps from note to note, etc. We would probably hear this as still more prominent.

We tend to hear things as being in the foreground if they're high, loud, fast, or subject to a rapid rate of change. For example, the drum groove of a song is often quite loud in comparison to the rest of the mix. But because it's also generally consistent and unvarying in relation to the other voices, we're able to place it in its appropriate place in the texture: the middle ground or background. When a drum fill happens, however, the sudden change in textural density and complexity causes us to switch our focus to the drums; they temporarily jump to the foreground.

When writing music, try to think about where each element should reside on this three-layered depth chart of foreground, middle ground, and background. Draw the elements in a box like the one in the previous picture if it helps you to visualize. Or simpler yet, just make a list. Here's an example of what a list of song parts for a minimal techno track

might look like if divided into foreground/middle ground/background:

Foreground

> Pitch-shifted spoken text (irregular, mostly non-rhythmic)
> Metal clang “interruptions” (irregular, high-pitched, loud)

Middle ground

> Synth bleeps (medium pitch, repeating patterns)
> Synth marimba chords (medium pitch, repeating patterns)
> Acid bass line (medium pitch, repeating patterns; sometimes comes to foreground via brighter filter settings, variations in the pattern)

Background

> Drums (steady-state)
> Sub bass (steady-state, low pitch)

Each element in the list is followed by decisions about musical parameters which will help to reinforce the selected depth. The rate of change in the middle ground and background elements is kept to a minimum, while foreground elements are allowed more freedom of motion.

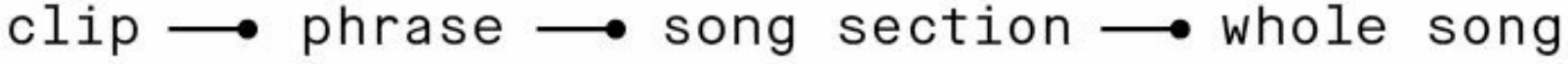
Additionally, try to experiment with making temporary changes in the depth at which certain voices reside, as in the drum fill example mentioned before. By varying the depth of elements in a song, we also draw attention to the fact that we’re using depth in the first place, which helps to reinforce the overall effect.

Fuzzy Boundaries

Problem:

DAWs tend to automatically structure material in blocks (often called "clips" or "regions") with clearly defined beginnings and endings. Musical structure is then created by moving the blocks around in time, but the sharp edges between the blocks also create a specific type of musical abruptness which may sound stiff, formulaic, and unrefined.

Most pieces of music divide time into sections, where each section contains a particular type of musical material that contrasts with that in the other sections. Additionally, these sections often exist within a hierarchy of sectional layers, with smaller contrasting subsections chained together to form the larger sections. In the abstract, this hierarchy might look something like this:



DAWs represent this musical structure in a very clear way; blocks of time appear as literal blocks on the screen. But the boundaries of these blocks suggest boundaries in time that are not always good representations of the musical results we want. Here are some ways to make the boundaries between formal sections less abrupt.

Solutions:

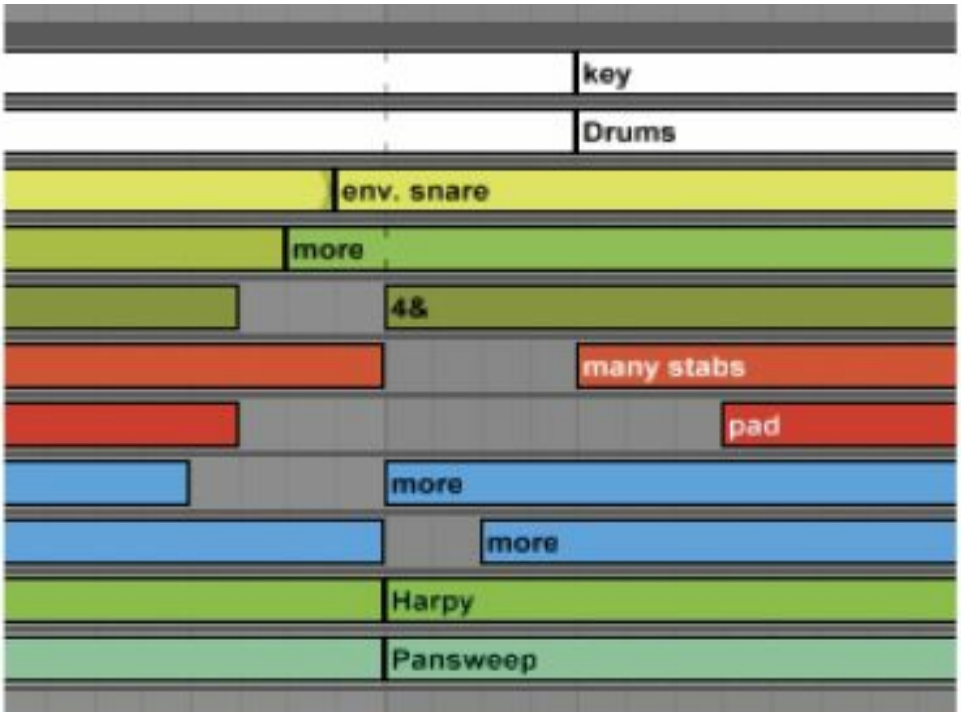
Abrupt formal boundaries look like a straight vertical line that cuts through the material in your tracks from top to bottom:



The most straightforward way to deal with this kind of abruptness is to simply make that line become jagged or disappear entirely by:

> extending material from the previous section in one or more (but not all) tracks.
> retracting material from the previous section in one or more (but not all) tracks.
> deleting material on either side of the formal boundary in one or more tracks.

For example, here's how this formal division might look after applying all of these techniques:

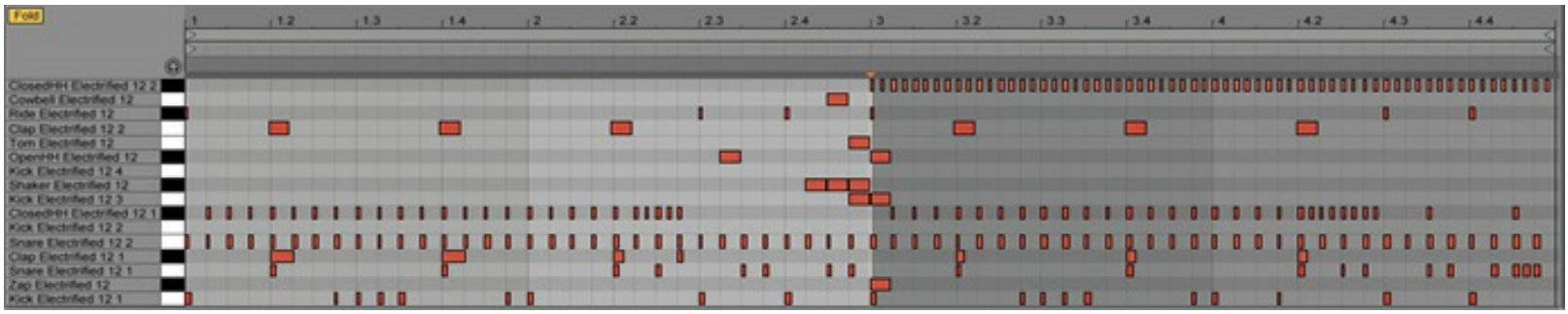


In this example, only the bottom two tracks have been left as they were in the original version. The top two tracks have had material extended beyond the formal boundary, while the next two tracks transition early. The following five tracks all use silence to blur the transition.

~ Note that this example is not a prescription for a particular set of actions; there's nothing inherent about this particular pattern of changes that will make sense in all cases. Of course, it's necessary to adapt what you do to the specifics of the material in your song.

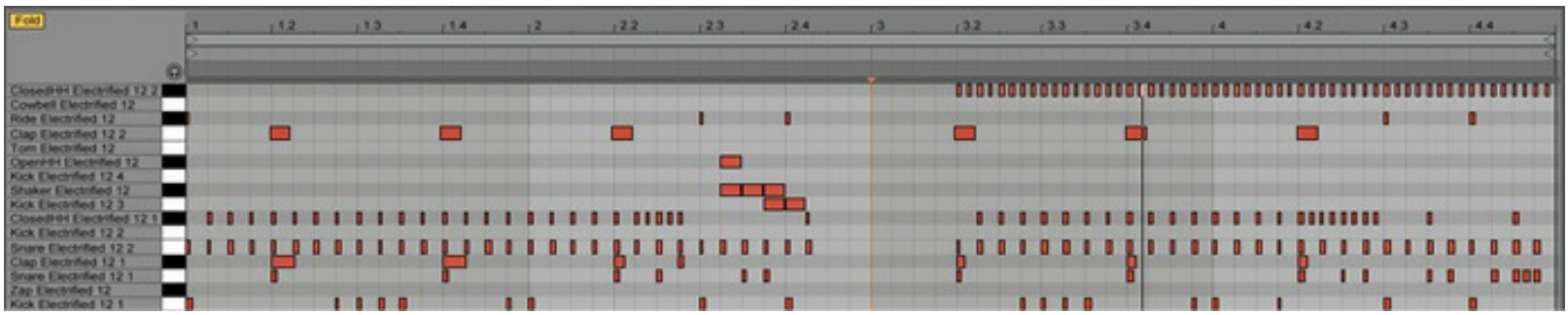
These kinds of "coarse-grained" changes to the length and position of clips can be effective. But sometimes you'll want to edit the material within the clips themselves. Here are some "fine-grained" ways to think about working with this material:

Anticipation - Drum fills are a common technique used to create a sense of transition from one section to another. But it's common to hear drum fills that lead out of one section and then end precisely on the downbeat of the next section. Here's an example of a "typical" drum fill that ends exactly on a downbeat. The energy peaks at the downbeat of bar 3:

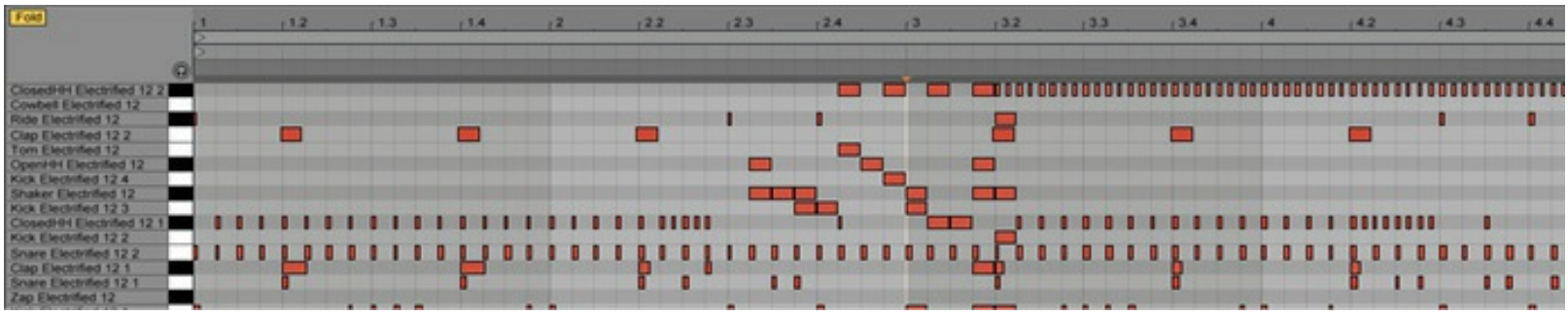


This is often musically effective, but it can sometimes feel formulaic and abrupt. Instead, try writing drum fills that anticipate, or end slightly earlier than expected. Reaching the peak of a fill slightly before the downbeat, for example, can create an interesting shift in the flow of musical time. This can be particularly effective if the actual downbeat is de-emphasized somehow, perhaps by stopping the drums entirely until

the second beat of the next section, and then re-entering with a “crash” or some other moment that emphasizes the sense of a musical ending. Here’s an example of a drum fill that anticipates the sectional boundary. The energy peaks before the downbeat, followed by silence and a re-entrance on beat 2:



Hesitation - Conversely, try writing a drum fill that lasts longer than expected, extending over the formal boundary between the sections and ending after the rest of the musical material has already changed to the new section. In these cases, it often works well if there’s no sharp emphasis on the end of the fill, and instead the drums of the new section simply take over as seamlessly as possible after the fill. Here’s an example of a drum fill that hesitates beyond the sectional boundary. The energy continues through the downbeat and peaks on beat 2:



Note that you can also apply anticipation and hesitation processes to musical elements besides drums. Try embellishing bass lines using these techniques, or extending (or shortening) sustained chords.

Creating Variation 1: Mutation of Clones

Problem:

You've come up with a great-sounding seed of a musical idea. But now that you have it, you find that you don't know how to proceed. Maybe it's a few notes or a few bars, but it's definitely not long enough to be a finished piece. You keep listening to the promising idea over and over again, unable to see a direction that will get you from here to a song.

There are a number of methods for generating many new ideas from the seed of one simple idea. Here is one recipe.

Solution:

Duplicate the initial idea a number of times (maybe eight or so) so that you have a number of identical copies. Edit the first duplicate until you have made only one meaningful change. You'll ultimately need to use your own taste and intuition to determine what "meaningful" means in the context of this particular idea, but the broad categories to consider are the fundamental parameters of music:

- Sound: changes to the timbre
- Harmony: changes to the chords
- Melody: changes to the foreground line
- Rhythm: changes to the timing of events
- Form: changes to the structure or distribution of smaller-scale components within the idea

You've succeeded when you can listen to the original and the edited duplicate back-to-back and clearly hear (or at least feel) that they're somehow different.

After making your one meaningful change to the first duplicate, move to

the second duplicate and again make one meaningful change. The change could be of the same type as the change you made to the previous edit or it could be a change of an entirely different musical parameter. Again, all that matters is that it is audibly different from the original (and, in this case, also audibly different from the previously edited version).

Repeat this process until you've made one meaningful change to each of the copies.

What you have now is a collection of "siblings," each of which is a direct descendant of the "parent" idea. Because you've made only one meaningful change to each duplicate, the relationship to the original will be clear.

Creating Variation 2: Mutation Over Generations

Problem:

You've come up with a great-sounding seed of a musical idea. But now that you have it, you find that you don't know how to proceed. Maybe it's a few notes or a few bars, but it's definitely not long enough to be a finished piece. You keep listening to the promising idea over and over again, unable to see a direction that will get you from here to a song.

There are a number of methods for generating many new ideas from the seed of one simple idea. Here is one recipe.

Solution:

Make a single duplicate of the initial idea. Edit this duplicate until you have made only one meaningful change. You'll ultimately need to use your own taste and intuition to determine what "meaningful" means in the context of this particular idea, but the broad categories to consider are the fundamental parameters of music:

- Sound: changes to the timbre
- Harmony: changes to the chords
- Melody: changes to the foreground line
- Rhythm: changes to the timing of events
- Form: changes to the structure or distribution of smaller-scale components within the idea

Once you're satisfied with the change that you've made to the duplicated idea, repeat the process. But this time, begin by creating a duplicate of the duplicate (rather than the original). Now make one meaningful change to this third-generation duplicate. Try not to go "backwards"; that is, don't simply undo the change that you made in the previous generation. Continue this process a number of times (maybe

eight or so), each time using the previous variation as the seed for the next one.

What you have now is a collection of "descendant" ideas, each of which is the direct offspring of the previous idea, but which can all be traced back to the original "ancestor." Although you've made only one meaningful change to each idea, all of them have also inherited the changes from all of the preceding generations and so may become increasingly remote from the original ancestor.

Creating Variation 3: Note Transformations

Problem:

You've come up with a great-sounding seed of a musical idea. But now that you have it, you find that you don't know how to proceed. Maybe it's a few notes or a few bars, but it's definitely not long enough to be a finished piece. You keep listening to the promising idea over and over again, unable to see a direction that will get you from here to a song.

There are a number of methods for generating many new ideas from the seed of one simple idea. Here are some recipes.

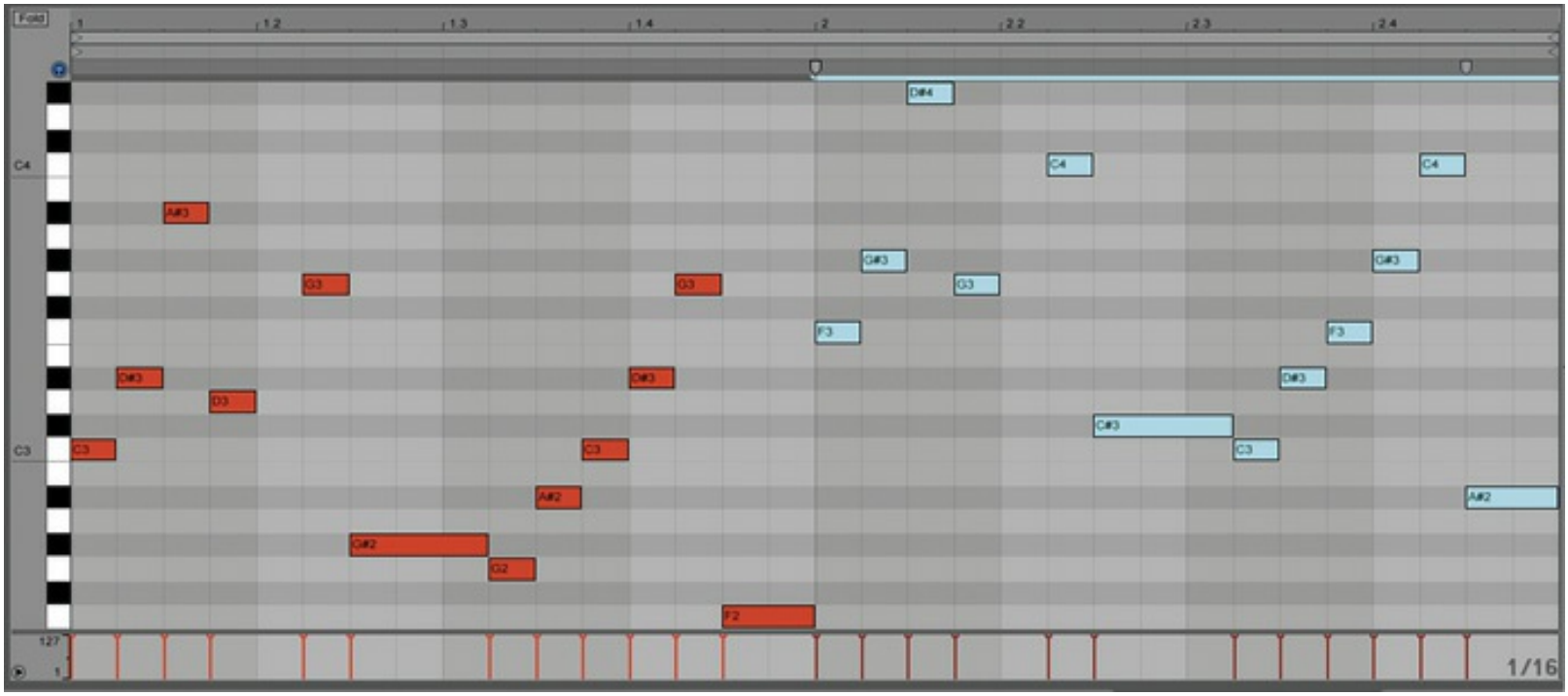
Solution:

Most DAWs offer a range of note transformation features that allow you to make predictable, rule-based changes to a selection of notes. The resulting patterns will usually have a clear, organic connection to the original pattern, although they can often sound quite different. Some types of transformations include:

Transposition

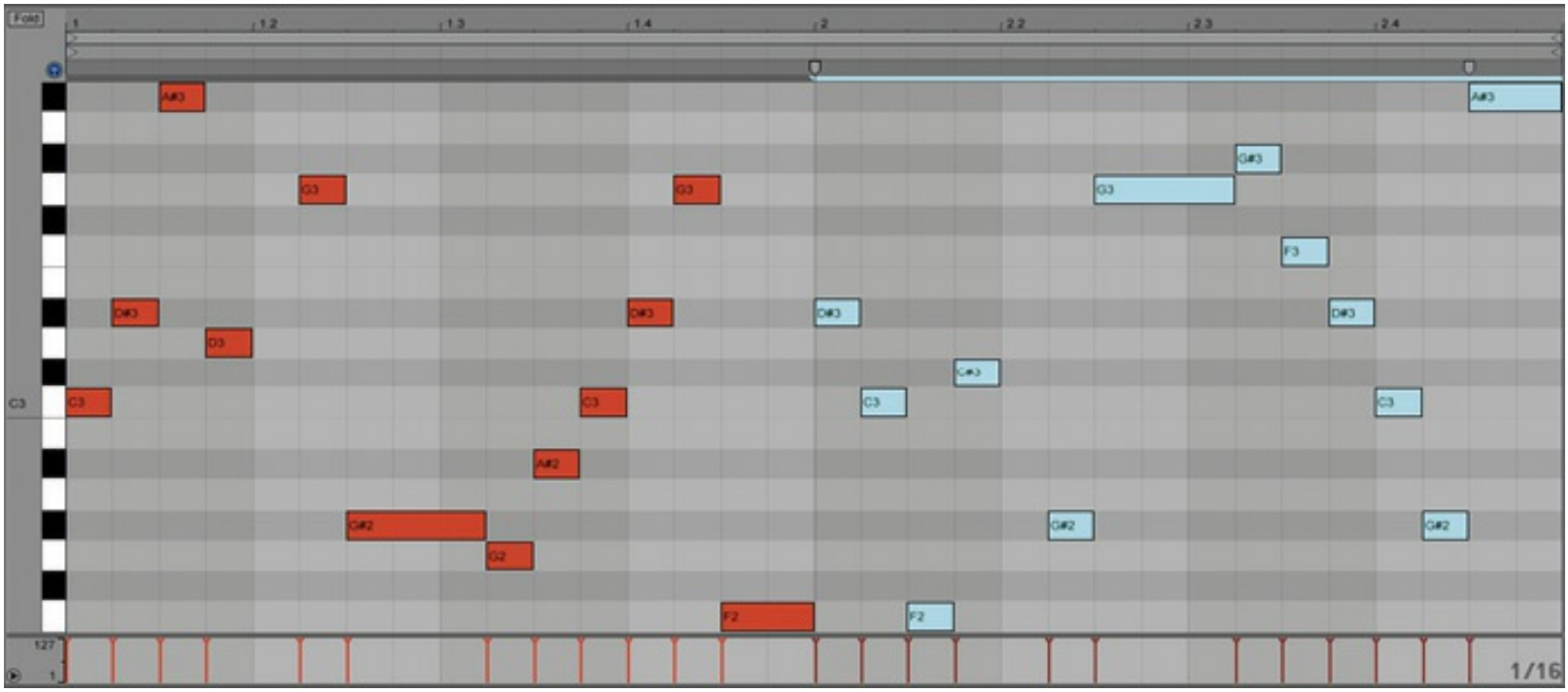
Transposition means shifting an entire pattern of notes up or down by a specific number of semitones. In a transposition operation, the relationship between all of the notes in the pattern will remain unchanged. Many DAWs provide a quick way to transpose MIDI notes, but you can also do this manually by simply selecting a group of notes and dragging them vertically or horizontally to a new position on the piano roll. Transposition can also be applied to audio samples, although there will generally be some change to the timbre of the audio as a result.

Here is a one-measure-long MIDI phrase, followed by a version of that phrase that has been transposed up by five semitones:



Inversion

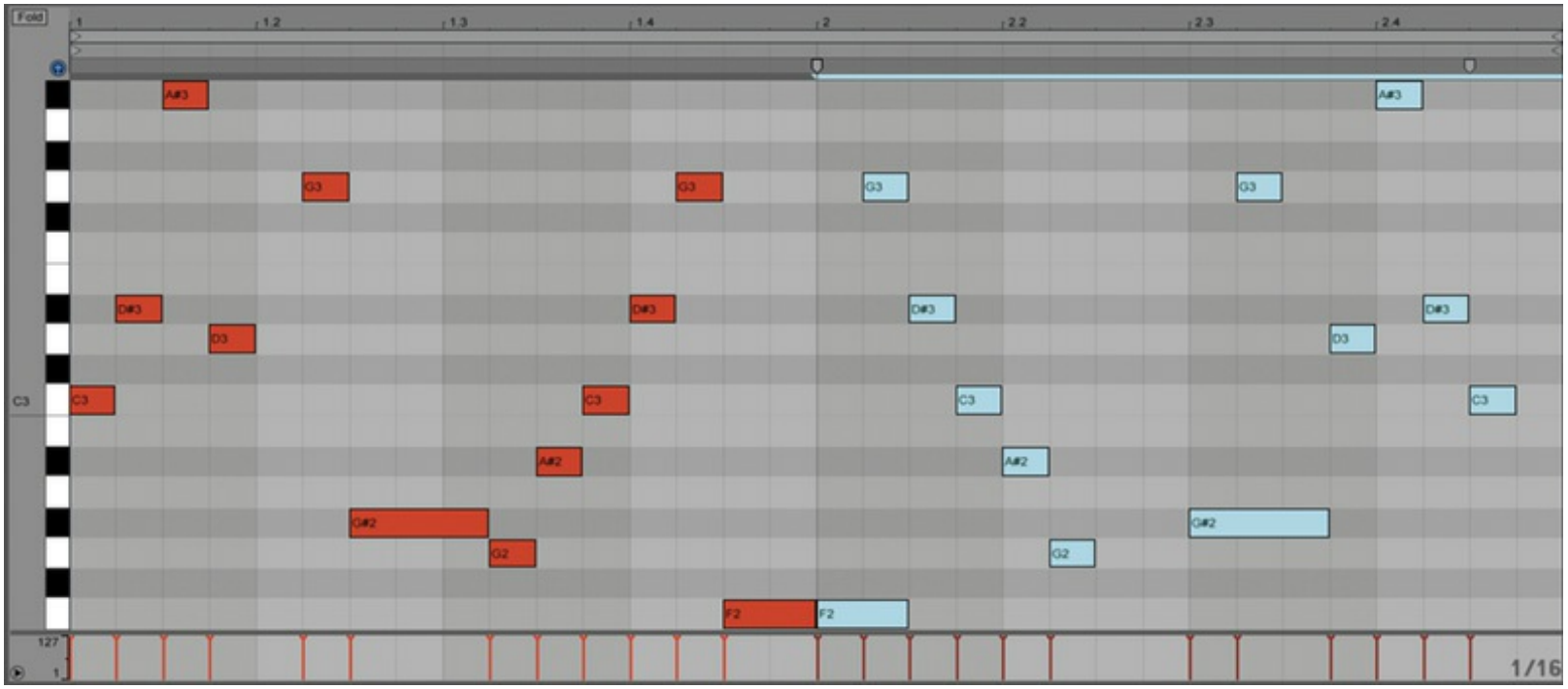
Inversion (in this context) is the process of flipping a collection of notes "upside-down" so that the lowest note becomes the highest note and vice versa. The shape of the pattern and interval distance between each of the notes is maintained, but in the opposite direction of the original. This process can be applied in the MIDI domain, either via an automatic process built in to the DAW or by manually moving the notes. For example, here is a one-measure-long phrase, followed by a version of that phrase that has been inverted:



Retrograde or Reverse

Retrograde means flipping a collection of notes backwards, so that the last note becomes the first note and vice versa. This process can be applied in the MIDI domain, either via an automatic process built in to the DAW or by manually moving the notes. Retrograde can also be applied in the audio domain, usually by applying a “Reverse” function to the sample. Note, however, that reversing a sample will also reverse the envelope contour of the audio itself, so the results will be quite different than a comparable process applied to the same MIDI.

Here’s an example of a one-measure-long MIDI phrase, followed by a version of that phrase that has been reversed:

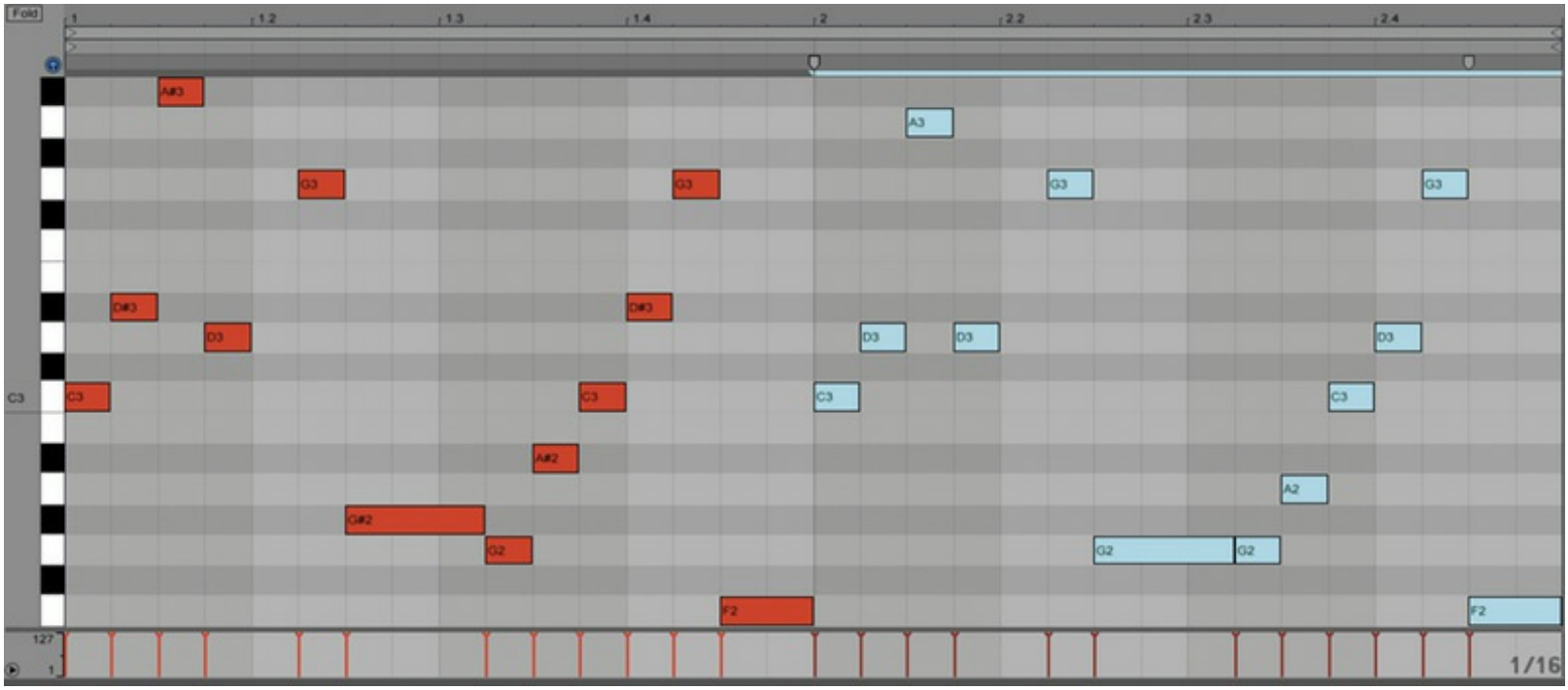


Note that in this particular retrograde algorithm, the distance between the note onsets is reversed. A true mirror image would mean that the note ends of the original became onsets in the reversed version. While literal mirroring is probably easier to understand conceptually, the results are often not particularly musically interesting.

Constraint to a Scale

A passage of MIDI notes can be selectively transposed so that the resulting pattern contains only notes that are within a particular scale. Many DAWs provide some type of device specifically for this purpose. These devices analyze incoming MIDI notes and remap them to an arbitrary pool of available notes before sending them out again.

Here's an example of a one-measure-long MIDI phrase containing a somewhat random collection of notes, followed by a version of that phrase that has been constrained to the pitches within the C Major scale:



Although the example above shows the actual notes that result from a scale-constraint process, note that many DAWs with MIDI processing devices apply their processing after the notes in the clip.

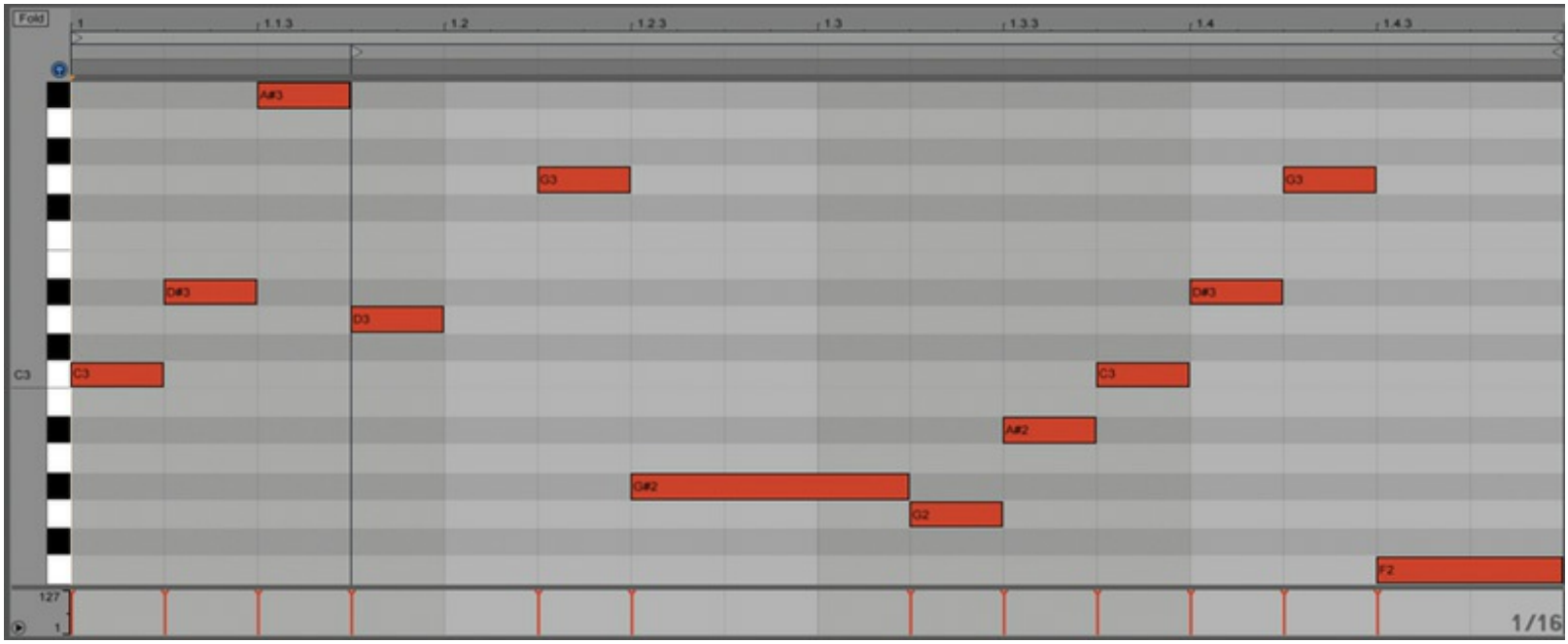
In order to get the notes you hear back into the original clip (as in the previous example), follow these steps:

1. Create a new MIDI track and set up your DAW's inter-track routing so that the new track records the output of the original track.
2. Copy the newly-recorded notes to the clipboard.
3. Paste them back into the original clip.
4. Disable or remove the scale-constraint device.

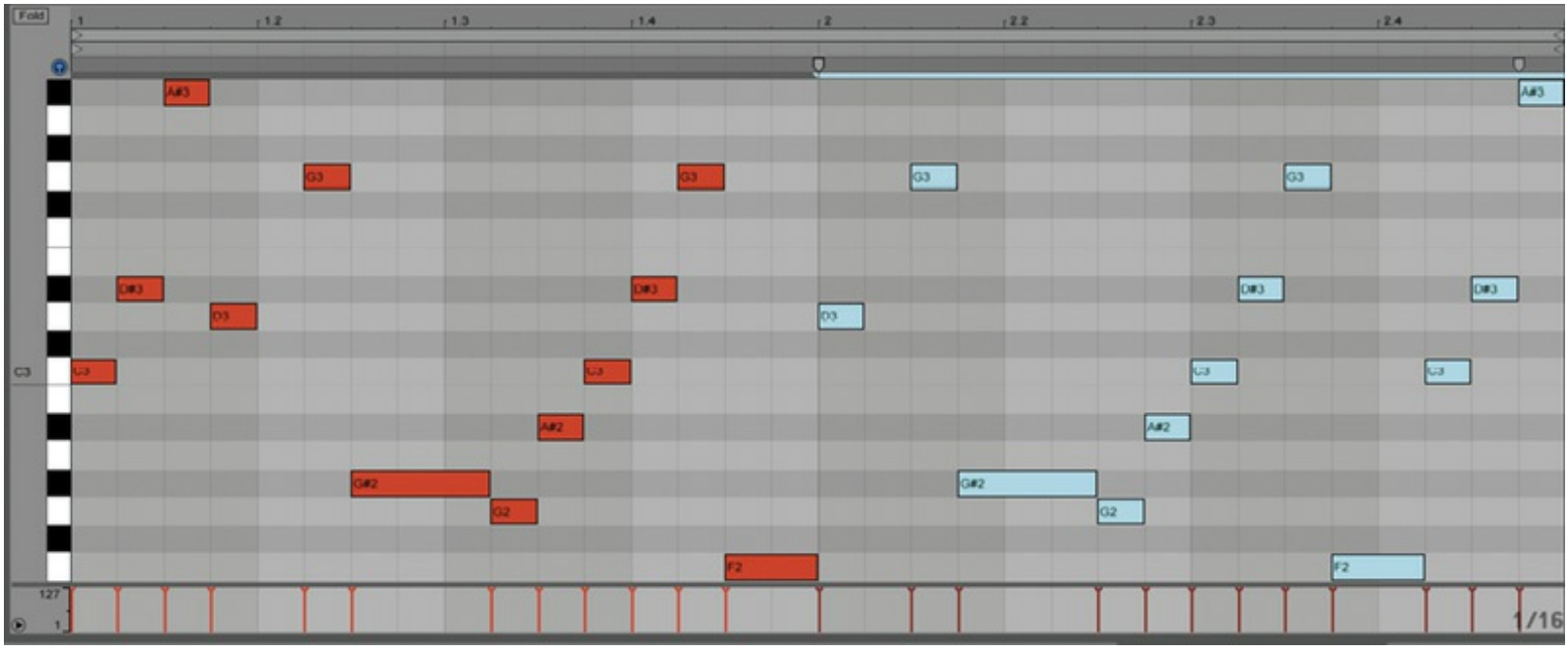
Time Shifting

A looped passage of material can be started at an arbitrary position within the passage. This is effectively the same as maintaining the exact pitches and rhythms of the original pattern, but shifting the pattern right or left to a new location. Some DAWs allow you to adjust the playback start position for a particular pattern independently of any other patterns. In these environments, simply moving the pattern's start

position achieves this effect. Here's an example of our original one-bar pattern with its start marker shifted three sixteenth notes to the right:

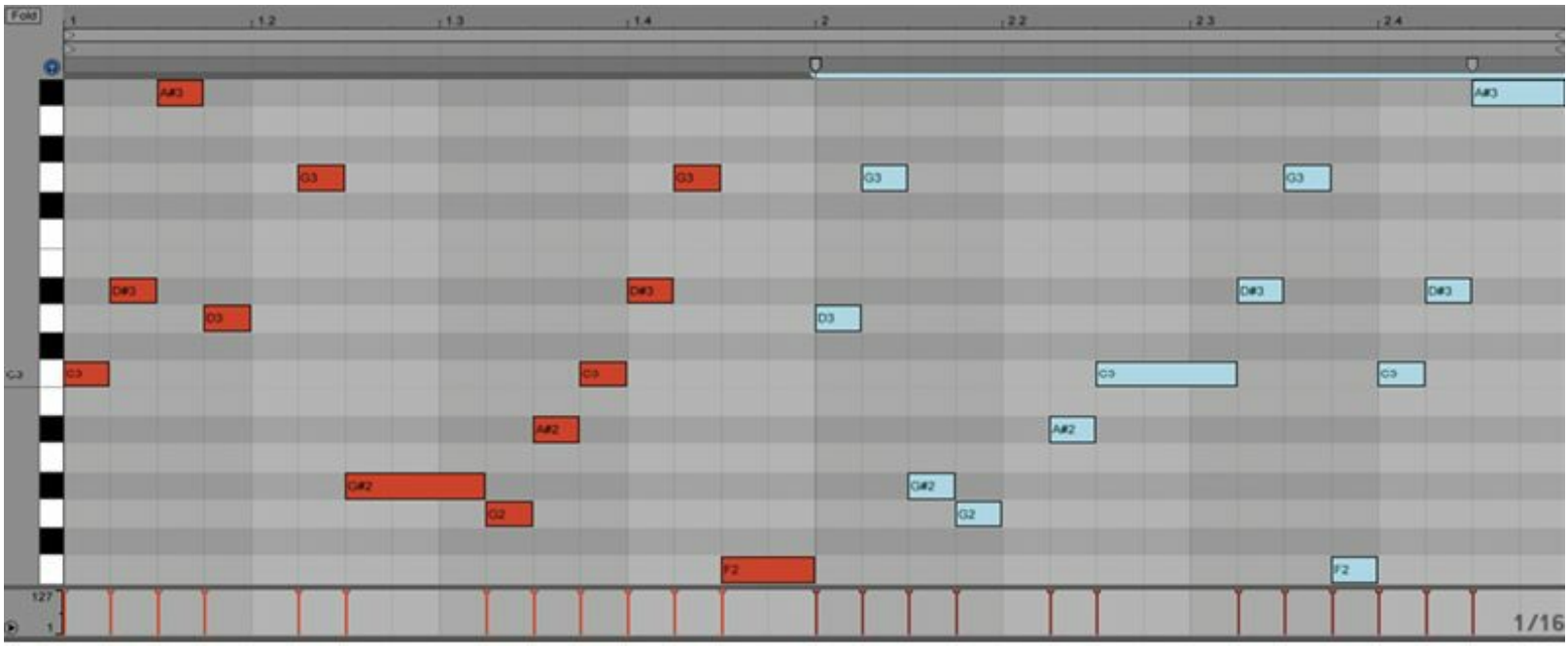


In other DAWs, you can manually copy the entire pattern and paste it to the new location with the desired offset. When shifting to the right, keep in mind that material near the end of the original pattern will need to be "wrapped around" to the beginning of the new pattern, and vice versa when shifting to the left.



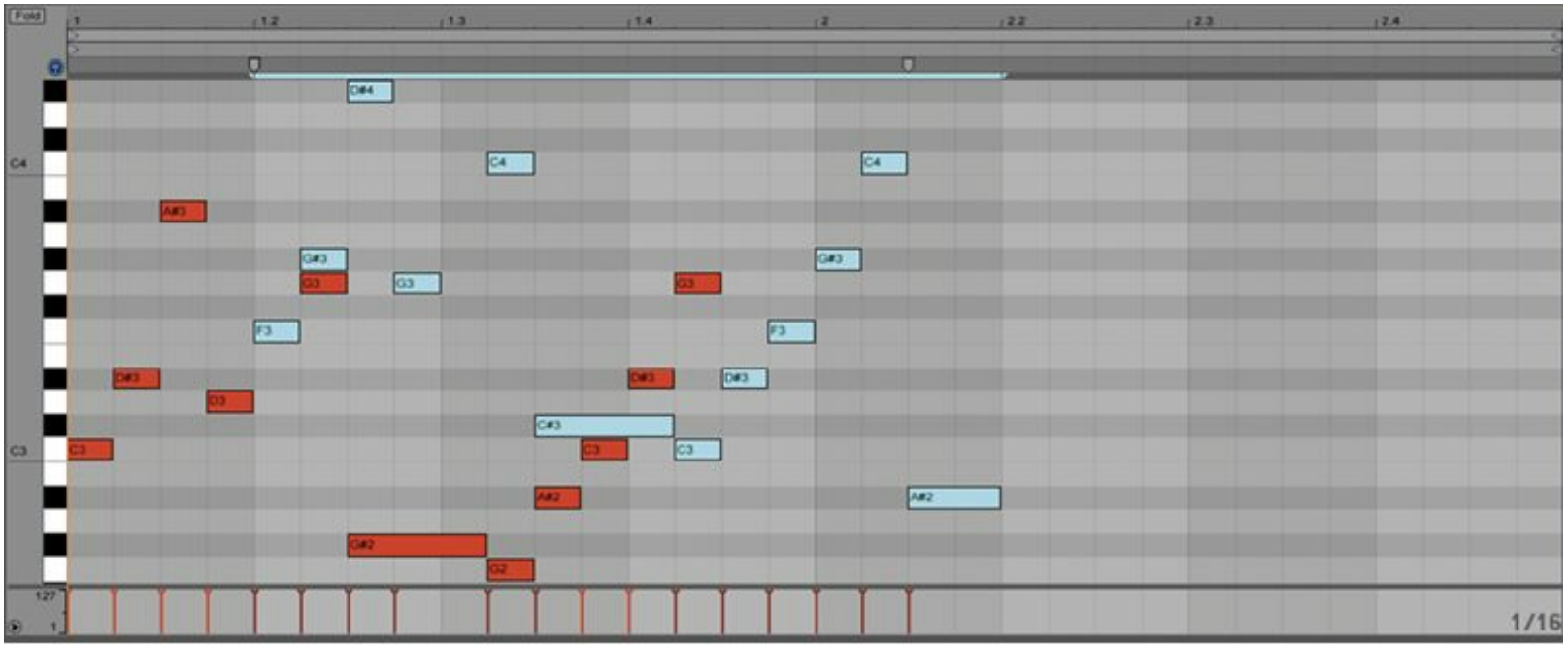
Pitch Rotation

Pitch rotation is a process whereby the rhythm of a passage of notes is retained but the pitches from the passage are shifted to the right or left in time. Here's an example of a one-measure-long MIDI phrase, followed by a version of that phrase that has been rotated to the right by three pitches. As with time shifting, pitches near the end of the pattern will need to be "wrapped around" to the beginning of the new pattern, and vice versa when shifting to the left.

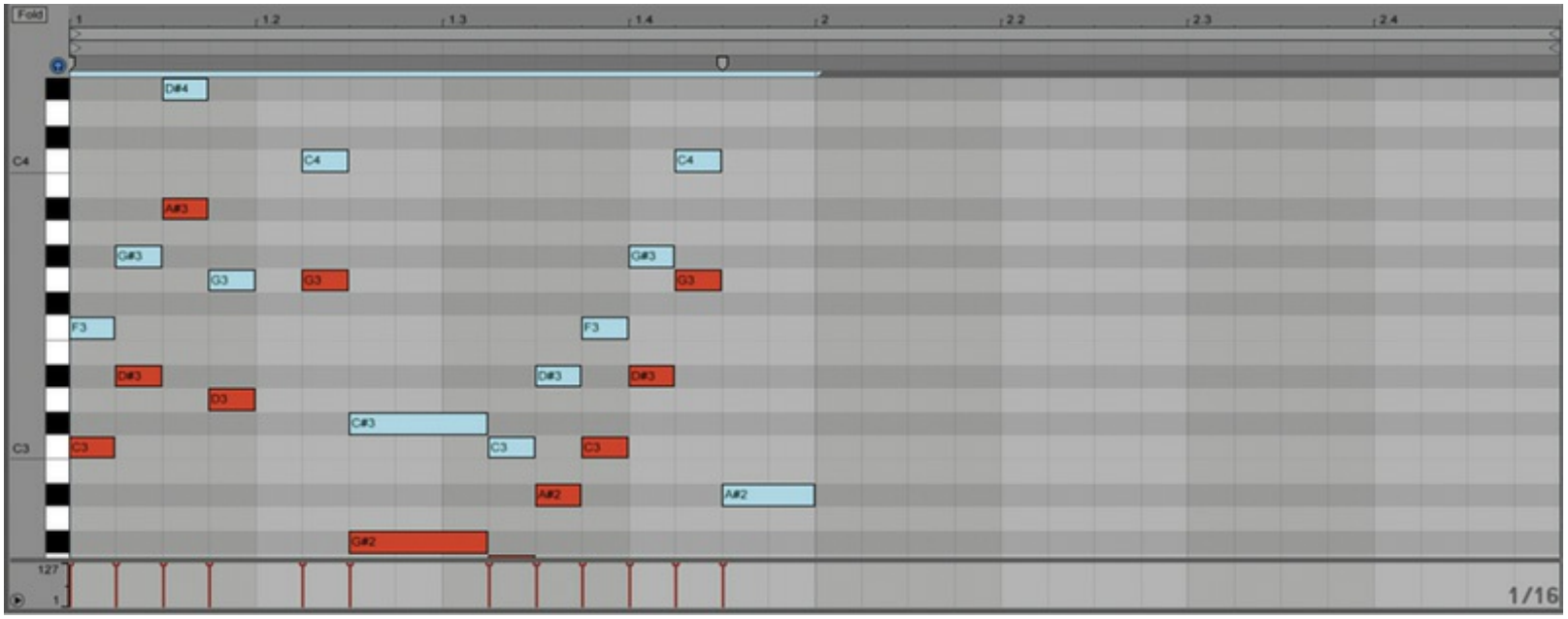


More Ways to Use the Transformations

Note that in all of these examples, the transformed version occurs immediately after the original. But of course, you could also treat the transformed version as an entirely separate, isolated pattern. Conversely, you could also overlap the transformation with the original to create a more complex pattern:



Or you could even superimpose them completely to create harmony:



Implied Rhythm in Short Loops

Problem:

You keep making the same kinds of beats over and over again, and you're looking for a way to expand your palette of rhythmic ideas.

We often develop a "vocabulary" of musical ideas that we rely on to create new music. Like spoken language, the richness of our musical expression is directly related to the depth of this vocabulary. When we know only a few words, we can say only a few things. Likewise, when we know only a few musical patterns, we're limited in our ability to create music that's different from the last time.

The most obvious place to get ideas for new beats is from other music, but here's an idea for finding rhythms in another way.

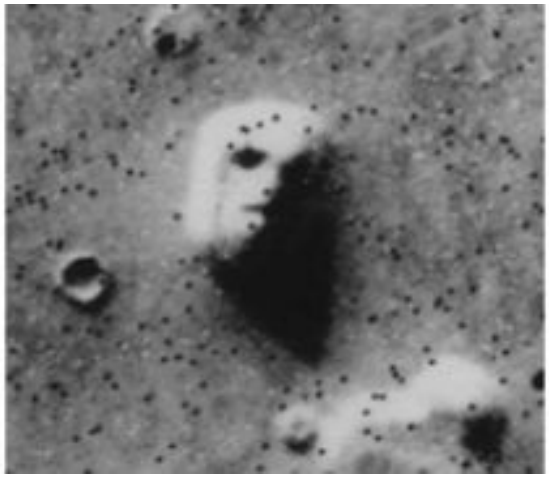
Solution:

Find or make an audio recording of anything. You'll get the best results from material that has at least some variation over time; a steady-state sine wave probably won't work well for our purposes here, but almost anything else is a viable source. It should be at least a few seconds long, but beyond this there are no restrictions on length. Very long material—minutes, hours, etc.—is fine.

Import this audio into your DAW and create a loop around an arbitrary portion of it. Now, while listening, begin to decrease the length of the loop until it's at most a few seconds long. One- or two-second loops seem to work best for this exercise.

As you continue to listen carefully to this loop for a while, you'll start

to notice rhythmic patterns emerge. These are the result of our inherent ability to find patterns in chaos, like hearing the sound of a ringing phone in the white noise of a shower. The phenomenon of finding patterns in unordered stimulus is known as pareidolia. For example, most people see a “face” in this image of a natural rock formation on Mars.



The bits and pieces of sound that make up the perceived rhythmic patterns may be the result of technical “errors” in your sample playback tool, such as loop boundaries that aren’t aligned with zero-crossing points in the sample. For our purposes here, these errors are exactly what we want; don’t try to fix them by enabling features with names like “snap to zero-crossings.”

If your original source material already had a clearly defined rhythm or tempo, try to loop a portion of the material that plays against this in a clearly audible way. For example, make the loop so that the repetitions are not synchronized with the original tempo. The goal isn’t to make use of the overt rhythms that exist in the large-scale source material, but rather to discover the rhythms that emerge from the repetition of tiny portions.

Good sources include:

> Existing music (ideally by you or by someone from whom you've secured rights, but anything can work technically). Remember, however, that you're not looking for the musical information from the original source. The goal is to find the implied rhythm heard only when listening repeatedly to tiny fragments.

> Field recordings of any kind—urban, industrial, or nature sounds work quite well for this.

> Recordings of speech. Rhythmic patterns emerge quite naturally from repeated fragments of speech. In fact, melodic patterns tend to emerge from short speech loops as well. For an interesting example of this in practice, listen to Steve Reich's Different Trains for string quartet and tape. For this piece, Reich listened for the implied melodies in short fragments of speech and then transcribed them to be played by the stringed instruments.

~ Bonus: In addition to audio, you can also use this technique with your own MIDI recordings, particularly if you've recorded extended improvised material without a metronome. Again, try to adjust the loop so that it plays against any rhythm or tempo that's suggested by the original material itself.

Once these rhythms have emerged from the loop, there are several different ways you can proceed. The most obvious is to simply use the looped sample as an element in your song. For an interesting example of this, listen to Machinedrum's "Baby It's U" from the album Vapor City, which uses hydrophone recordings of dolphins and shrimp as both the basis of the groove and as a rhythmic backdrop over which more conventional drums are eventually added. Another solution is to apply the emergent rhythms from the original sample to another instrument, as in the previously mentioned string quartet Different Trains.

Asynchronous or Polyrhythmic Loops

Problem:

Many electronic genres make extensive use of loops. When you listen to your favorite examples of loop-based music, there's always some sense of motion that keeps the music fresh and interesting, but your own attempts to work this way usually strike you as static and boring.

It seems like a paradox, but "simple"-sounding loop-based genres like minimal techno can be difficult to emulate. All of the elements are on the surface, and it's easy to hear through the few layers of production. But although the surface level of much of this music suggests that there isn't much happening, there's a subtlety and richness of detail at work that separates the good music from the bad.

Our typical mental model of a sequenced MIDI loop in a DAW is something like this: An instrument is played by one looping clip with a fixed length (probably equal to one or more complete bars). At some point, we might stop this clip and then start another clip in its place. So at any moment, any given instrument is either silent or is being triggered by one clip.

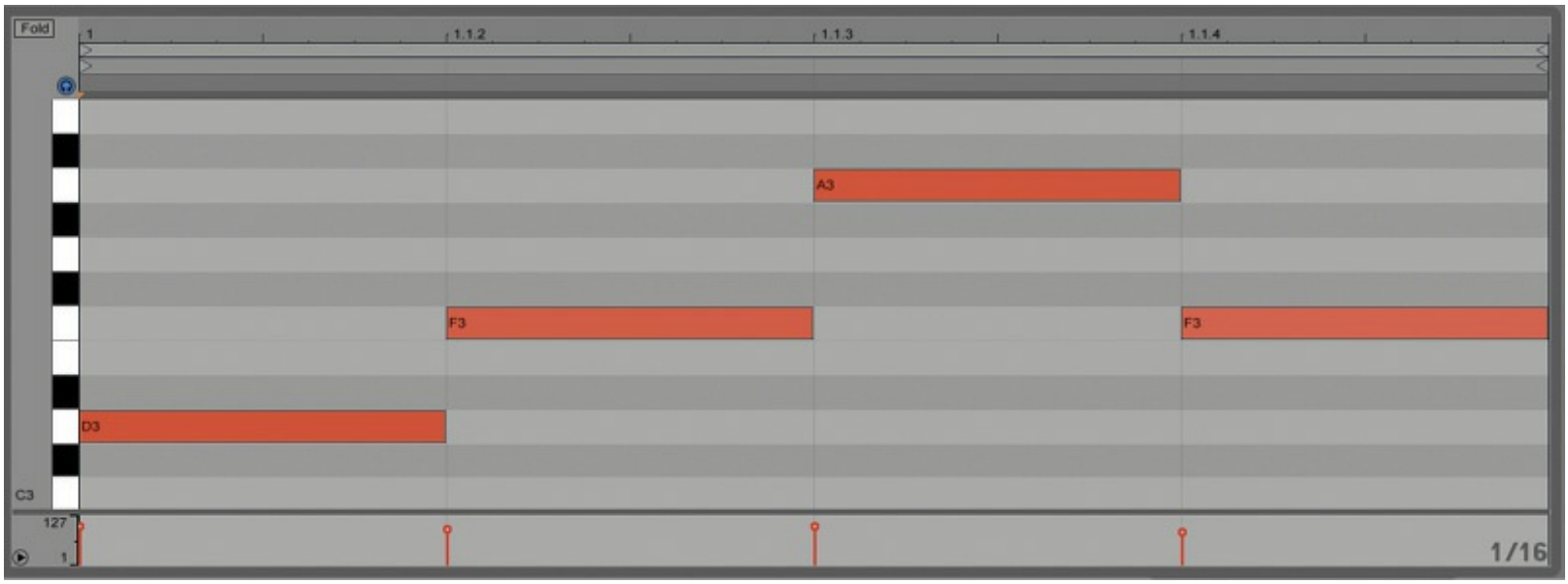
Here's a different way of thinking about looped clips that might make things more interesting.

Solution:

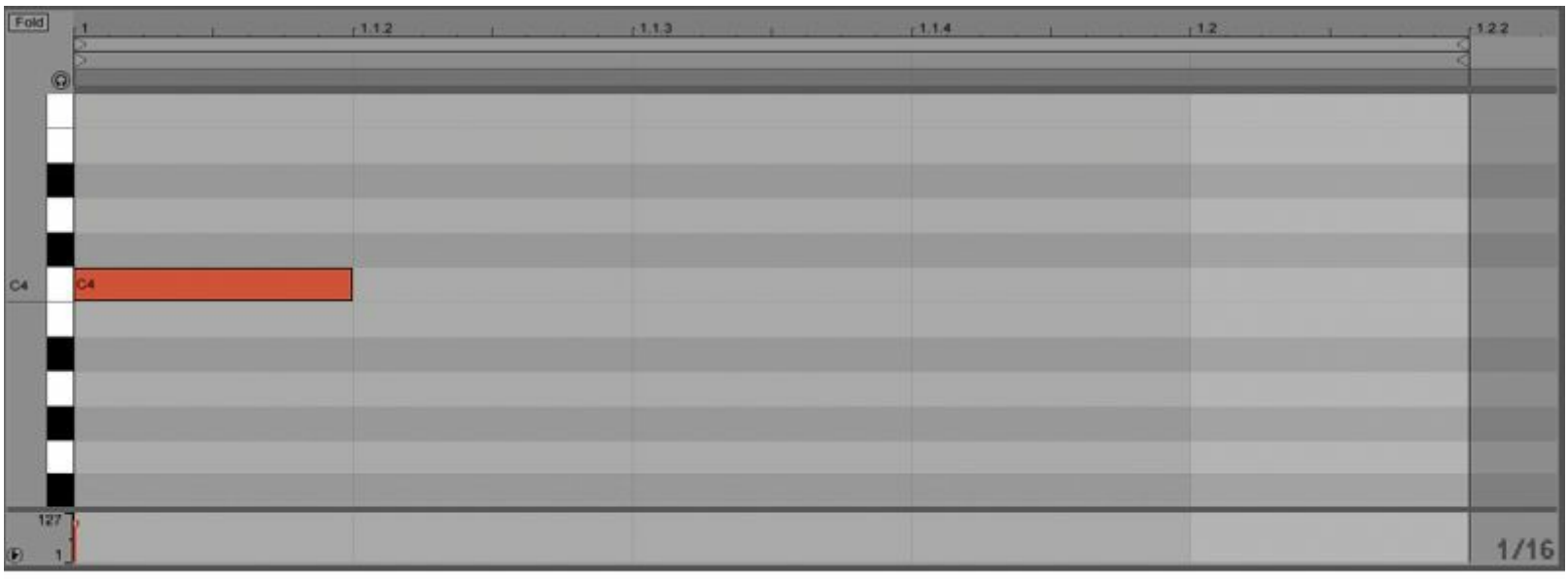
Instead of triggering an instrument with a single clip, try simultaneously triggering it with multiple clips of different lengths. In most DAWs, this will require some kind of inter-track routing. For example, one track could contain the instrument, while one or more additional tracks

contain no instruments but instead only the clips that will play the instrument on the first track. These “silent” clips will then have their output routed to the first track.

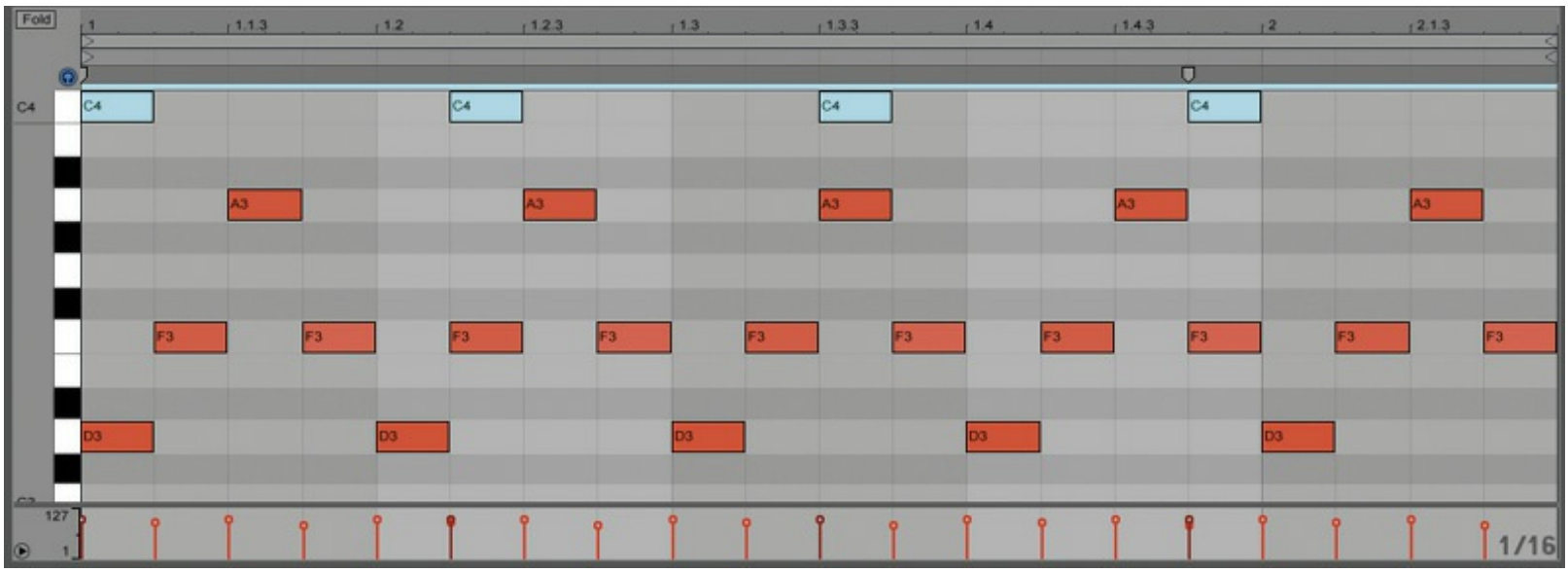
In this example, imagine an instrument is triggered by a simple looping MIDI pattern consisting of four sixteenth notes:



Simultaneously, the same instrument is being triggered by another loop that contains one sixteenth note followed by four sixteenths of silence. This pattern loops every five sixteenths; it is one sixteenth longer than the first pattern, so they will play out of sync with each other but in time with a shared pulse, a phenomenon known as polyrhythm.



The simultaneous juxtaposition of these two patterns creates an interesting auditory phenomenon. The ear can now hear three simultaneous patterns: the two that are actually being played and the resulting composite, which finally comes back into sync every 20 sixteenth notes:



Here are some variations on these concepts:

> Try adding more loops of differing lengths. For example, adding a third loop to the earlier example that repeats every three sixteenth notes will create a composite that is three times as long; the original patterns will realign after 60 sixteenth notes.

> Rather than letting the composite naturally realign, you could "artificially" realign it anywhere in an arrangement by simply pasting the clips so they begin at that point.

> Some DAWs allow automation to loop at a length that is "unlinked" from the length of the notes in the clip. For example, you could create an envelope that varies a synthesizer's filter cutoff and have this envelope loop at yet another length from the loops containing notes. Even if your DAW doesn't have this functionality within clips, you could manually create repeating automation patterns of any length when you build your arrangement.

> Similarly, if you have some kind of modulation source that can control parameters (like an LFO, for example), you could set this to a rate which is again asynchronous in relation to the looping notes.

> If you're using a plug-in instrument, try using duplicates of the instrument with slightly altered timbres, rather than actually using the exact same instrument. This will weaken the sense of a single composite pattern but may be more interesting anyway. You could even use this technique with hardware synths, provided you have enough of them (or you're willing to bounce to audio

after recording each pattern separately).

> When set against other instruments that are playing in more standard loop lengths, even a single "odd" loop can create the same kind of interesting effects. This approach is common in a lot of early acid tracks; listen to almost anything by Hardfloor for examples.

Layering of simultaneous asynchronous loops is a common hallmark of the so-called "Berlin School" of electronic music from the 70s and 80s, which was largely defined by multiple modular sequencers, each playing a monophonic loop consisting of a different number of notes.

A similar technique can be found in ambient pieces such as Brian Eno's Music for Airports, which is made from multiple tape loops of different lengths. This "analog" approach to the concept is truly asynchronous—if played forever, uneven tape loops are unlikely to ever resynchronize. You could achieve a similar effect in your DAW by offsetting one of your loop lengths so that it's not quite aligned to the metric grid. Try playing with the amount of offset until you find relationships which sound interesting. This can create some very unusual rhythmic effects.

Misusing Rhythmic Tools

Problem:

You have a variety of tools in your DAW arsenal—such as quantization, slicing, and beat-juggling effects—that are designed to help you work with rhythmic material. But you feel like there's more potential in these tools that you haven't yet discovered.

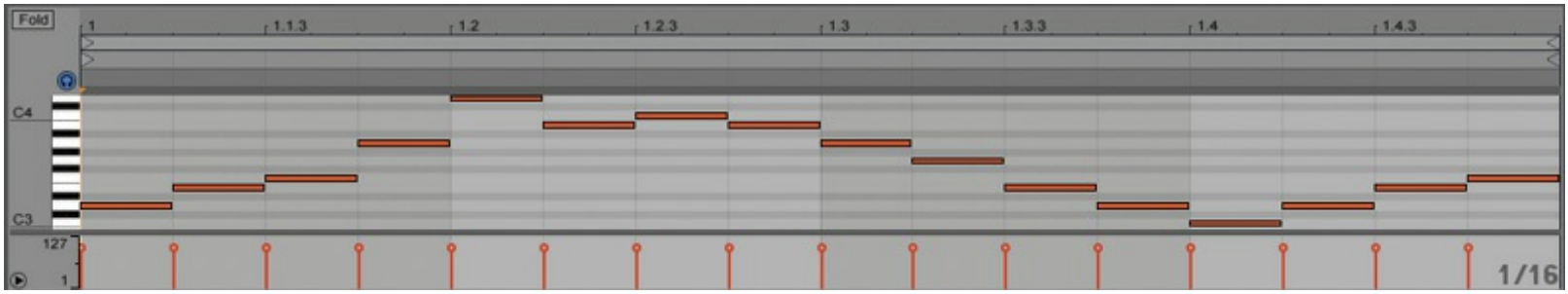
Some workflows and effects are clearly designed for use with very specific types of material. But there's no reason why you can't repurpose them for other uses. Here are some ideas for how to creatively misuse rhythmic tools to achieve unexpected results.

Solutions:

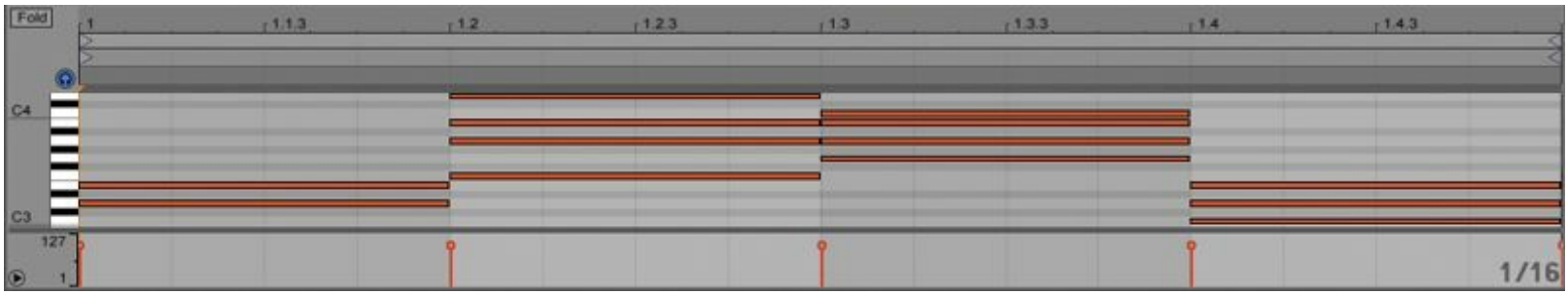
Misusing Quantization

Quantization is typically used to correct timing errors in manually played MIDI material and, in some DAWs, audio samples. But there are a number of ways to use quantization "incorrectly," sometimes with surprising results.

One interesting experiment: Try quantizing to a note value that is substantially slower or faster than the source material. If you played sixteenth notes, for example, try quantizing to a slower note value like quarter notes. If you originally played a monophonic figure, you'll end up generating chords. You can even apply successively larger quantization values to already quantized material to achieve the same effect. Here's a figure played in sixteenth notes:



And here's the same material after quantizing to quarter notes:



Another way to generate unexpected results from quantization is to start by playing your parts at a dramatically faster tempo than you actually intend. Because the errors in your playing will likely be more extreme than usual, applying quantization will then "correct" the note placements to unusual positions. When you then slow the tempo down to a more musically appropriate one, you may have created patterns that you never would have played intentionally.

Misusing Slicing

Slicing is typically used to separate rhythmic events (such as drum hits) from audio material such as drum beats or bass lines. But there's no reason why you can't slice any audio material, including entirely ambient pads, field recordings, or other non-rhythmic samples.

Another idea: If your DAW allows you to create or move markers in the original audio to determine the positions at which slices will be created, try placing the markers in deliberately "wrong" places.

In many DAWs, slicing generates a new instrument—and potentially a new effects chain—for each slice. This makes it easy to apply dramatically different types of processing to individual slices, regardless of their source. For this reason, slicing can often lead to interesting results even when you're slicing things in the "wrong" ways.

Misusing Beat-Juggling Effects

Effects that chop and rearrange incoming audio material in real time are normally used for creating variations of beats. But, as with the slicing examples discussed previously, you can also create interesting results by feeding these effects with any other material, including things that have no clear rhythm at all.

Although these are some examples for how to misuse specific workflows, consider applying this approach to all of your tools. Any effect, instrument, or workflow can potentially yield interesting results when used in unusual ways. Learn what your tools are supposed to do, but don't be afraid to make them do something else.

3+3+2 Rhythm

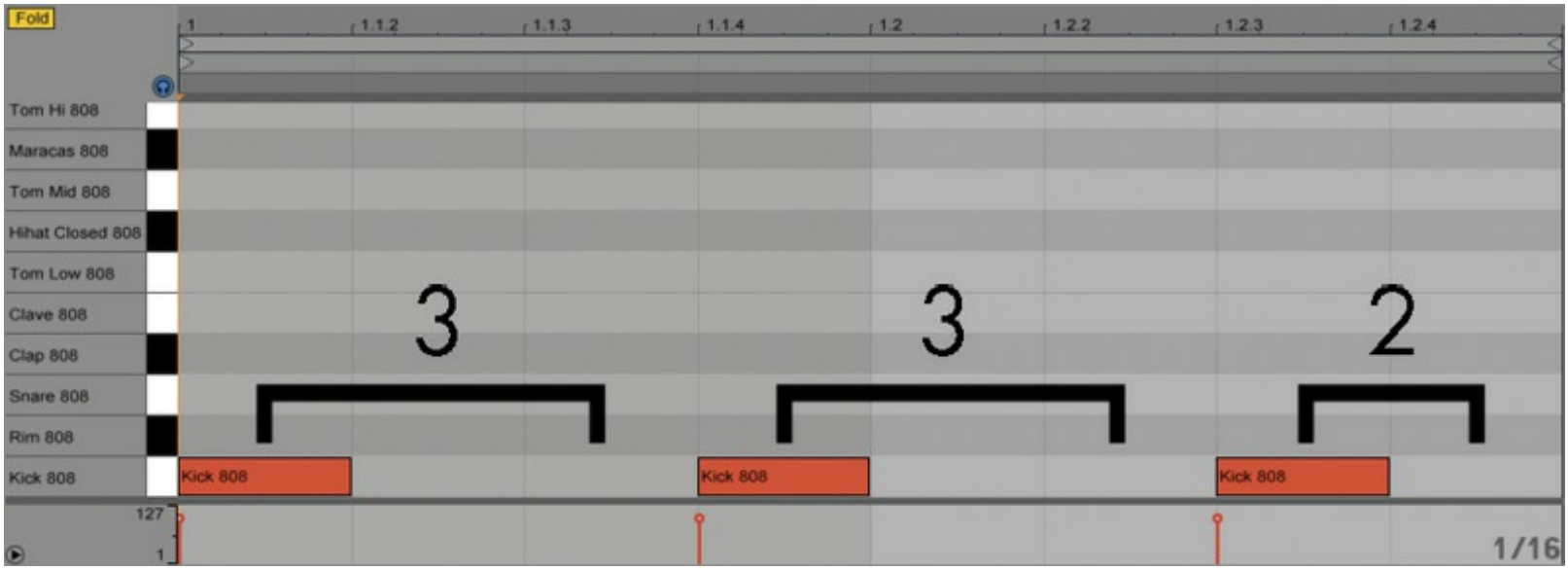
Problem:

There's a particular off-kilter, asymmetrical, funky rhythm that you keep hearing in music of all kinds, from hip-hop to footwork to rock and roll. What is it, and how can you use it in your own music?

For new producers, there's a tendency to be overwhelmed by the perception that all music is radically unique. The reality, however, is that there is a shared vocabulary of underlying musical patterns that artists frequently reuse across a variety of genres. Here's one of the most common.

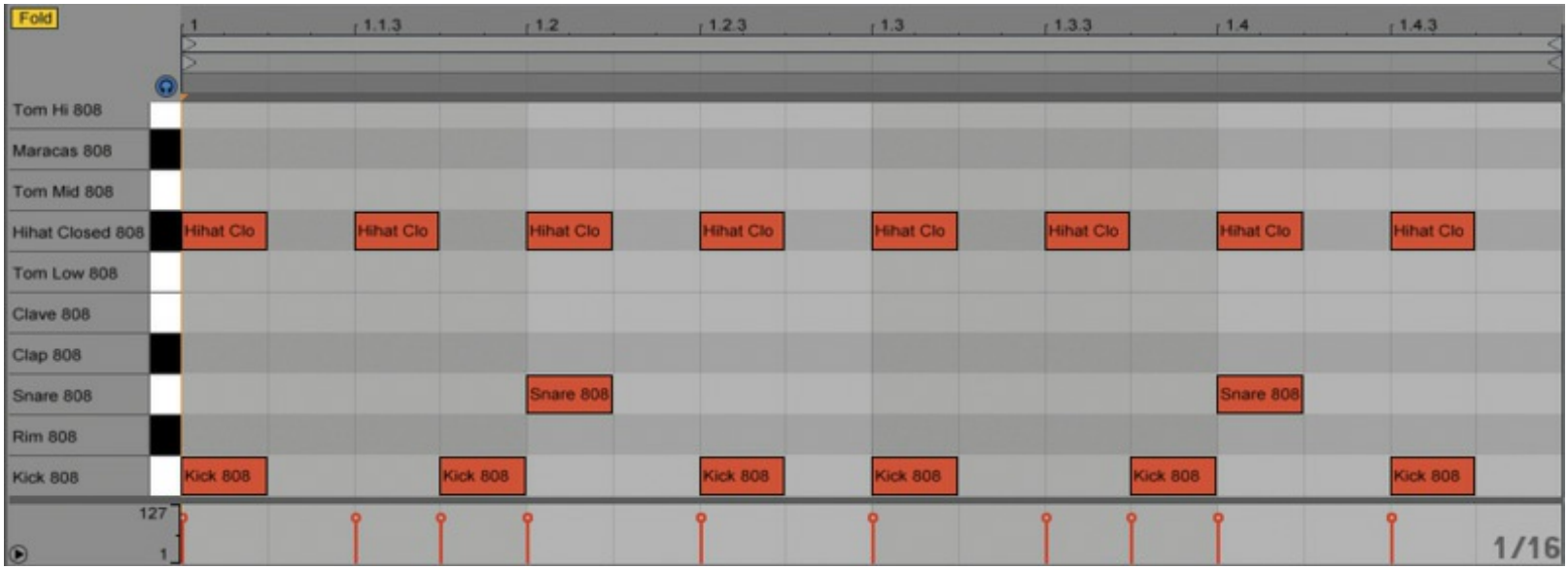
Solution:

The tresillo, or 3+3+2 pattern, is a widely used rhythmic pattern. Its origins are unknown, but it's a staple of African and Latin music and was eventually incorporated into early jazz, after which it made its way to all of jazz's descendants: R&B, rock, funk, and eventually modern electronic music. In its simplest form, the 3+3+2 pattern looks like this:

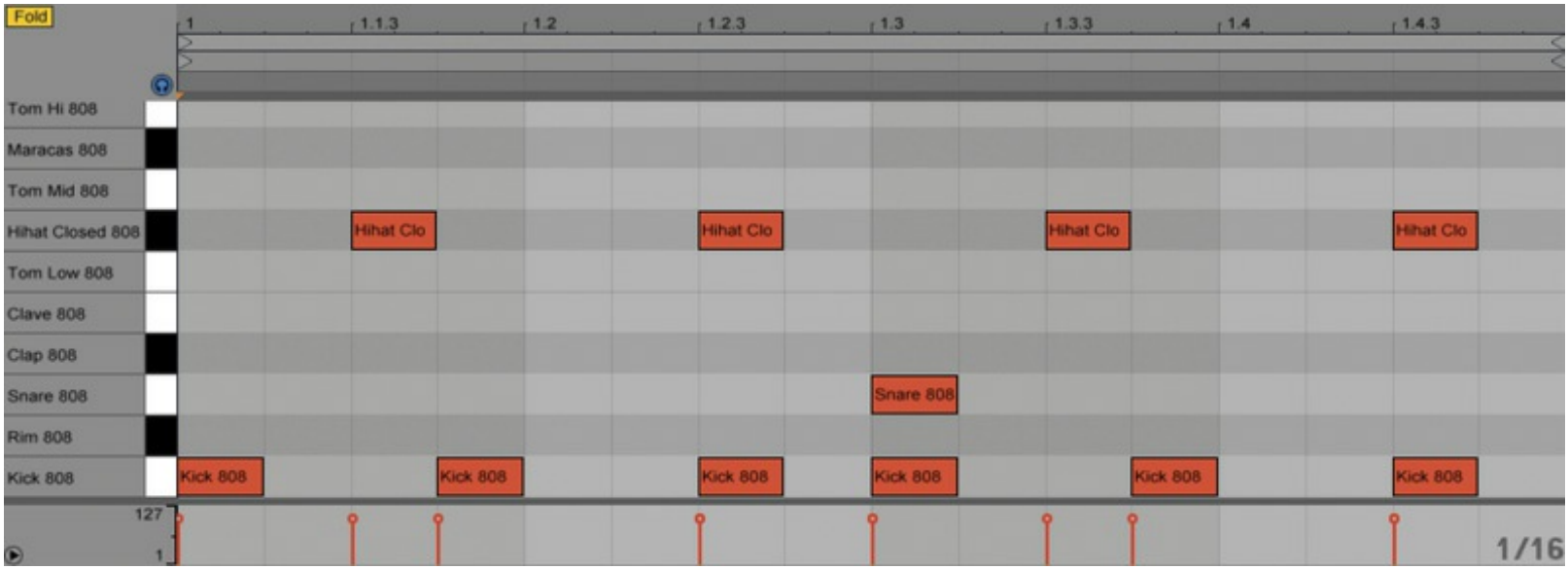


The “3+3+2” refers to the number of sixteenth notes between hits, and the pattern repeats every two beats. The pattern can also be played with other note durations; a half-time version would be measured in eighth notes, for example, and would repeat every four beats.

As a stand-alone pattern, 3+3+2 might not be so useful, but it really becomes interesting when overlaid against other patterns that are more symmetrical. For example, here’s the previous example as a kick drum pattern underlying a basic rock groove and expanded to fill a full bar:

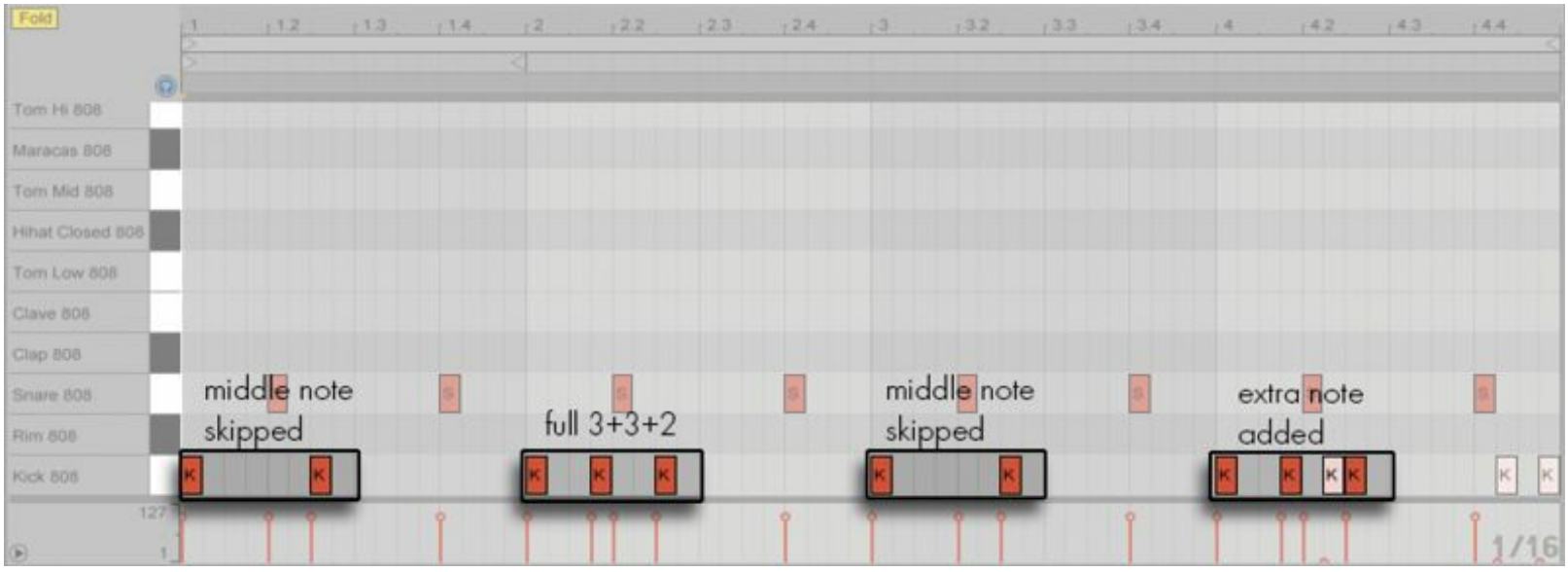


If the backbeats occur at half the speed (on beat 3 rather than beats 2 and 4), this same pattern begins to resemble a simplified version of the drum programming heard in a lot of bass music. With the addition of hi-hats on the offbeat eighth notes, the drum part for the track “No Think” by Sepalcure is based off this basic pattern:



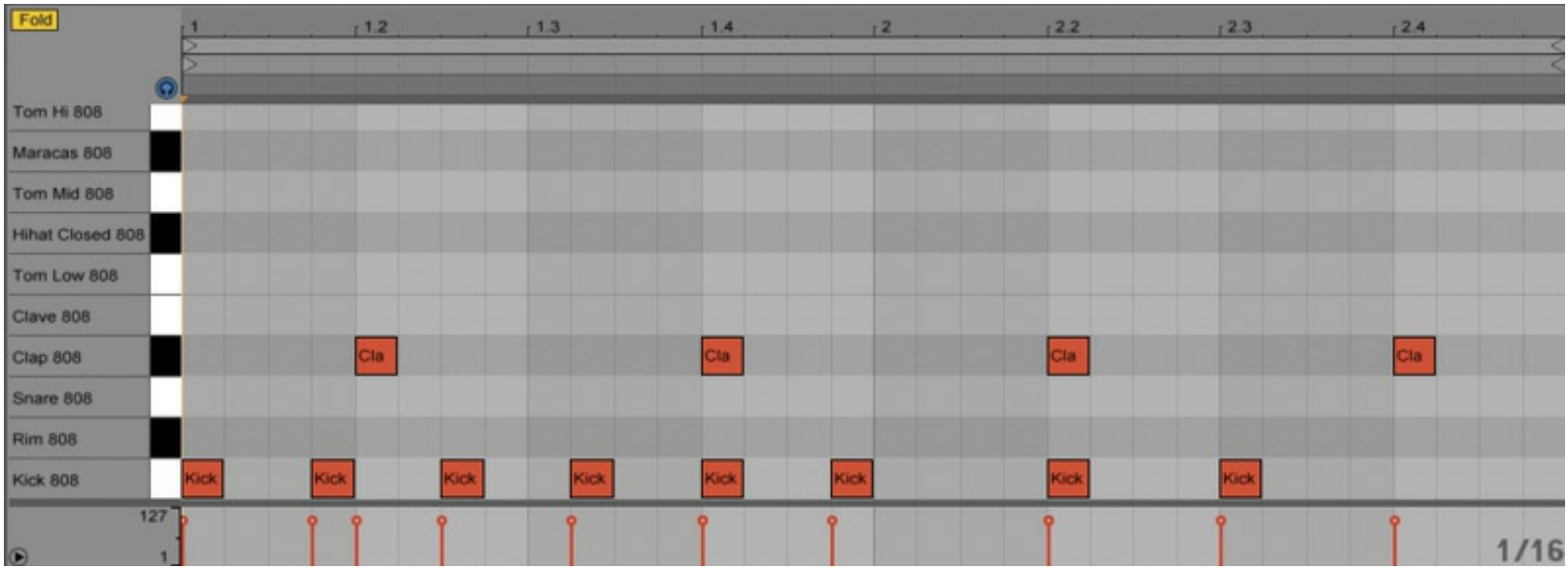
As you can see, there is a huge range of possibilities available just by combining the standard 3+3+2 rhythm with various other rhythms. Additionally, there are a number of ways to subtly vary the basic pattern, which can yield even more possibilities.

Skipping or adding one or more notes. By leaving out certain notes of the pattern, you can maintain the syncopation and asymmetry but in a more subtle way. Conversely, by adding notes to the pattern, you can create an embellished version. The main drum pattern in Lil Jon's "Turn Down for What" is an example of both processes at work. This is a four-bar phrase in which the kick drum pattern consists of three different types of 3+3+2 pattern:



The first and third phrases are identical, with the middle "3" of the 3+3+2 left unplayed. The second and fourth phrases play the whole pattern, but the fourth also adds an additional sixteenth immediately before the final "2." But because the 3+3+2 pattern has already been so strongly established, we hear this note purely as an ornament; the underlying 3+3+2 is still dominant.

Extending the "3." The "2" portion of the 3+3+2 pattern serves to resynchronize the gesture with the metric grid. But very interesting things can happen if you delay this resynchronization by repeating the "3" parts of the gesture additional times. An example of this can be heard in the track "Tenderly" by Disclosure. Here, the two-bar kick and clap pattern looks like this:



There are six kick drums in the first bar, which are all three sixteenths apart, followed by a pause and then a clear "realignment" with kicks on beats two and three in the second bar. This creates a strong contrast between rhythmic stability and instability, tension and release.

Now that you're familiar with the 3+3+2 rhythm, you'll start to notice it in various forms in all sorts of different kinds of music. And it's not just useful for drum patterns. You can use 3+3+2 to build bass lines, chords,

and melodies as well.

Programming Beats 1: On Looseness

Problem:

Your programmed drum beats just don't have the right feel. When you play them in by hand, they sound sloppy. But when you program them with the mouse or by quantizing your manual playing, they sound too perfect.

In certain genres of electronic music (like some techno, house, electro, and EDM) absolute quantization is completely appropriate for the style. It's common to hear drum parts in these styles with both perfect timing and completely consistent and unvarying dynamics. But for other styles (like most hip-hop), a looser, more organic feel is often more appropriate. In much of this music, drum parts are often directly sampled from recordings of real drummers. So in songs where programmed drums are desired instead, it often makes sense to try to emulate the human feel that comes naturally to actual humans. But while humanness isn't a characteristic that comes naturally to machines, many beat programmers don't have the drumming skills to program convincing drum parts that sound like something a real drummer would play. Here are some ideas that can help you with your beat programming, whether you're a drummer or not.

Solutions:

Apply Quantization in Small Amounts

Most modern DAWs provide controls that allow you to change the intensity or amount of quantization applied, usually represented as a percentage. For example, if you're quantizing to sixteenth notes at 100% quantization, the selected notes will be moved forward or

backward from their current position to the nearest sixteenth note. But quantizing to 50% will move the notes only half of the distance between their original position and the nearest "correct" note position. This means that human inaccuracies will be reduced rather than eliminated. This technique is probably the best way to retain a sense of human feel while still allowing you to correct for some sloppiness in your playing.

Apply Quantization to Only Certain Instruments

After recording a drum pattern with a MIDI controller, the typical approach is to apply quantization to the entire pattern, which means that all of the instruments in the drum kit will be shifted as a single unit. Instead of this, try selecting and quantizing only some of the instruments in the kit, leaving the others exactly as you played them. For example, try quantizing only the hi-hats, leaving the kick and snare alone. Or try the inverse, leaving the hi-hats loose but quantizing the kick and snare so that they're perfectly in time. Although any perfect quantization is not really a representation of how a real drummer would play, this technique can result in much more realistic-sounding drum patterns. You can also combine this technique with the previous one for even more possibilities.

Find Your "Natural" Tempo

A common technique when programming drum patterns in real time is to record at a slow tempo and then increase the tempo during playback. While it's true that it may be easier to play accurately at slower tempos, there's usually a bottom limit to this, and most drummers (and non-drummers) find that their rhythm is most accurate within a particular range of tempos—not too slow, and not too fast. This range varies from person to person however, and if you hope to improve your beat programming, experiment to find the tempos that work best for you.

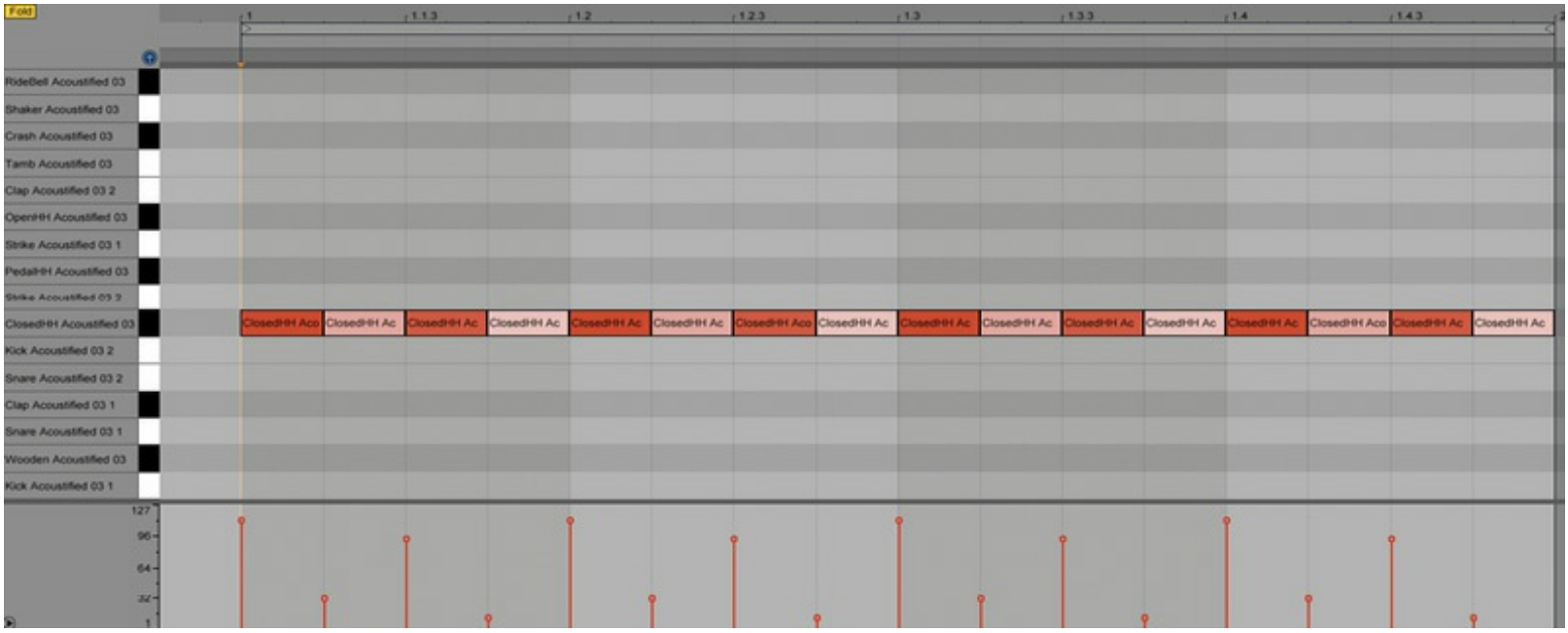
Once you've found something that feels comfortable, try programming your beats here and see if you can get the feel you want without applying any quantization afterwards. Keep in mind that timing errors are generally more perceptible if you play back at a slower tempo than you recorded, and less perceptible if you play back faster.

Humanize/Randomize

In addition to adjusting how much you quantize after recording, you can also sometimes get very musical results by applying "humanization" (subtle timing and/or velocity randomization) to your material. Even if you've quantized something to lock completely to the grid, humanization provides a way to reverse this effect, placing notes some distance away from the quantized positions according to whatever algorithm your DAW uses for this process. Note that applying randomization to material that hasn't been quantized will likely just exaggerate any timing inaccuracies that were already there; this process usually works best with material that's exactly on the grid.

Adjusting Velocity Instead Of/In Addition to Timing

Although timing is the most important variable that determines the feel of drum patterns, the relative volumes (velocities) of individual notes can also have a significant effect. For example, a hi-hat pattern in straight sixteenth notes sounds quite different if all of the velocities are identical than it does if there is some variation in volume from note to note. Most drummers will play a pattern like this with a slight emphasis on the first and third sixteenths, while playing slightly quieter on the second and fourth:

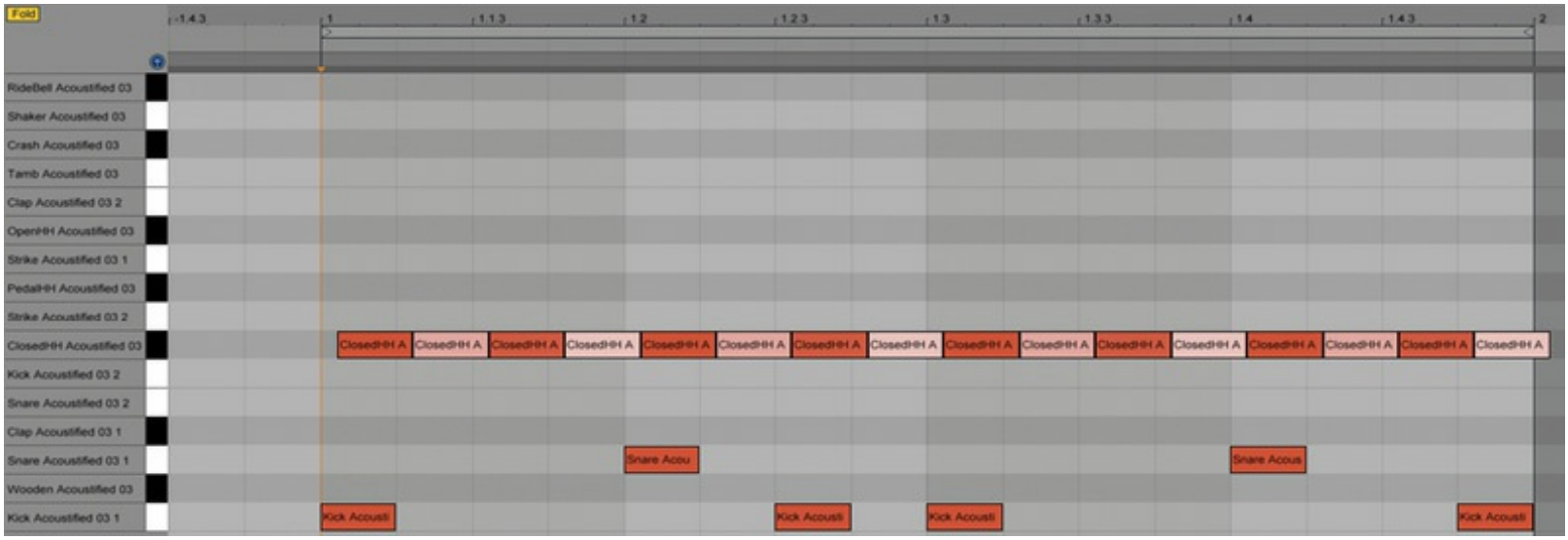


The image above shows a hi-hat pattern with perfect quantization but with velocities similar to what a drummer might play. Even without adjusting timing at all, you can often make a stiff, static groove come to life by carefully adjusting some of the velocities.

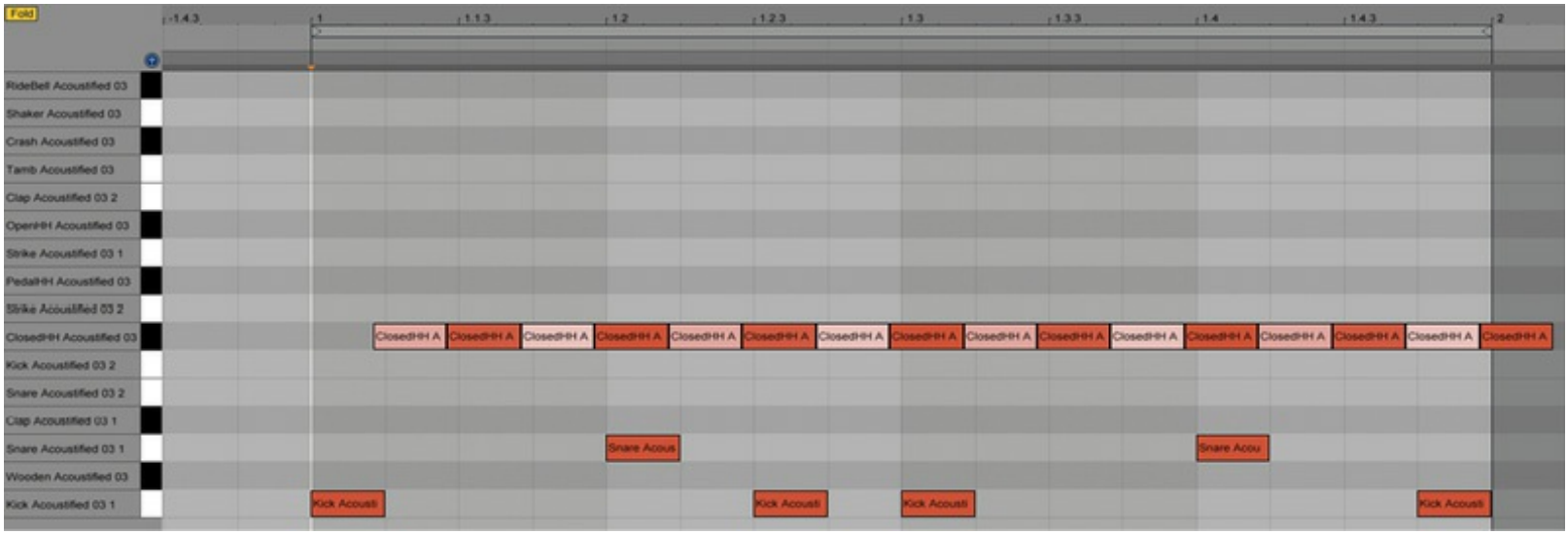
Manually Shift Specific Elements

Both for quantized and unquantized material, the surefire way to get the results you want is also the most time-consuming. Manually editing individual notes to change their relationship to each other and to the metric grid can ensure that everything happens exactly where you want it. Unfortunately, this assumes that you know what you're looking for, which may not be the case. Here are some ideas:

> Try dragging some or all of the snare drum or hi-hat notes slightly after their "intended" grid positions to create a heavy or "laid-back" feel. You'll often hear this "behind the beat" approach in funk drumming. Here's an example of hi-hats placed slightly after the beat:



> Conversely, placing them early can make the music sound nervous or "driving forward." You'll often hear ska or punk drummers play "ahead of the beat." Here's an example of hi-hats placed slightly before the beat. Note that the "first" note has been moved to the end of the pattern, so that it plays slightly before the loop point.



How far to move the notes is entirely up to you; there's no formula for this. Very small shifts tend to work well at slower tempos, while you'll need to make more drastic changes as the tempo increases. But in general, even a little bit of distance from the grid can be very effective.

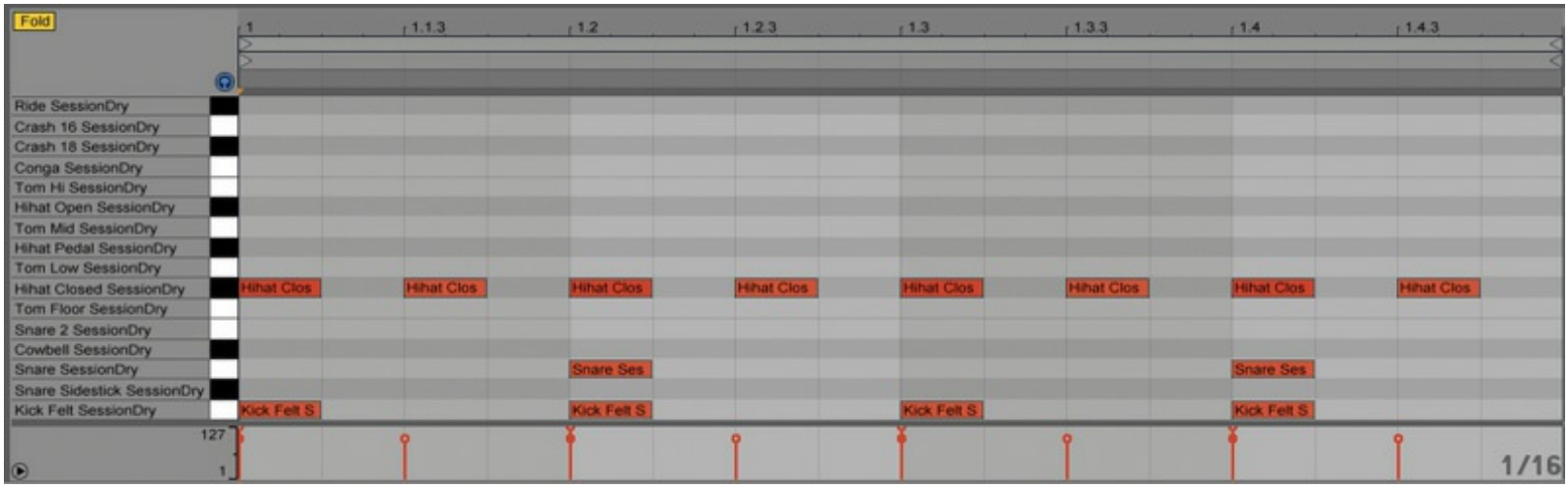
Keep in mind that if you move every instrument together, the listener won't hear any change. These subtle shifts are only audible if they occur in relation to other elements that have not been moved.

Programming Beats 2: Linear Drumming

Problem:

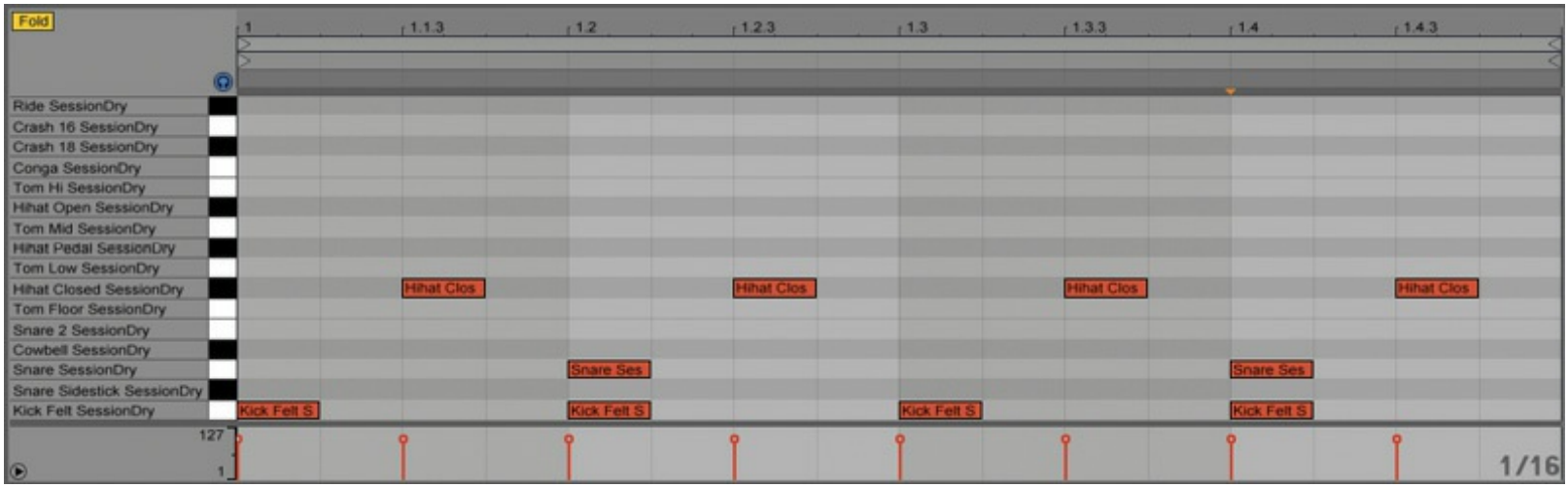
Your programmed drum beats tend to use the available instruments in "expected" ways: Hi-hats keep time, kick drums emphasize the beats, snares/claps are placed on or around the backbeats (beats 2 and 4). But in the music you admire, you sometimes hear other, more creative ways of working with drums. Sometimes it almost sounds like the drums are playing a "melody" of their own. But when you try to create patterns like this, it just sounds random and chaotic.

The various instruments in a drum kit have a tendency to be used in a somewhat standardized way. This is true across a variety of genres, both acoustic and electronic. For example, here's a very basic rock and roll drum pattern:



The kick drum plays on every beat, the snare on the backbeats, and the hi-hats on every eighth note.

This can be converted into a very basic house pattern simply by removing all of the hi-hat notes that appear on strong beats (leaving only the offbeat eighth notes):



One of the hallmarks of these kinds of conventional drum patterns is that they're polyphonic—multiple instruments can play at the same time. For a human drummer with four limbs, there's a maximum possible polyphony of four simultaneous "voices." Of course in the electronic realm, there's no such limitation, although experienced drum programmers who are working in styles that are meant to mimic acoustic drums will generally stick with this four-voice limitation in order to create parts that sound as realistic as possible.

A side effect of thinking about drum patterns as polyphonic textures is that we tend to treat at least one layer as unvarying. In the grooves discussed earlier, for example, note that each individual instrument maintains a symmetrical pattern that never changes. In many musical contexts, these kinds of simple, steady patterns are completely appropriate. But here's another way of writing drum patterns that can yield some interesting results.

Solution:

Acoustic drummers (particularly in some funk, R&B, and fusion contexts) sometimes use a type of playing called linear drumming. This simply means monophonic—no two instruments can play at the same time. Linear drumming is similar to the melodic technique called hocket

(see Linear Rhythm in Melodies). An extremely simple linear drumming pattern derived from the previous examples might look like this:



But rather than just taking a conventional pattern and removing overlapping elements, try thinking about how to use the monophonic requirement in creative ways. For example, treating a monophonic drum part as a single, quasi-melodic line of notes may cause you to think of each instrument as having the same level of importance, rather than subconsciously assigning them to functional roles (like “timekeeping” or “accenting”). Consider the following pattern:

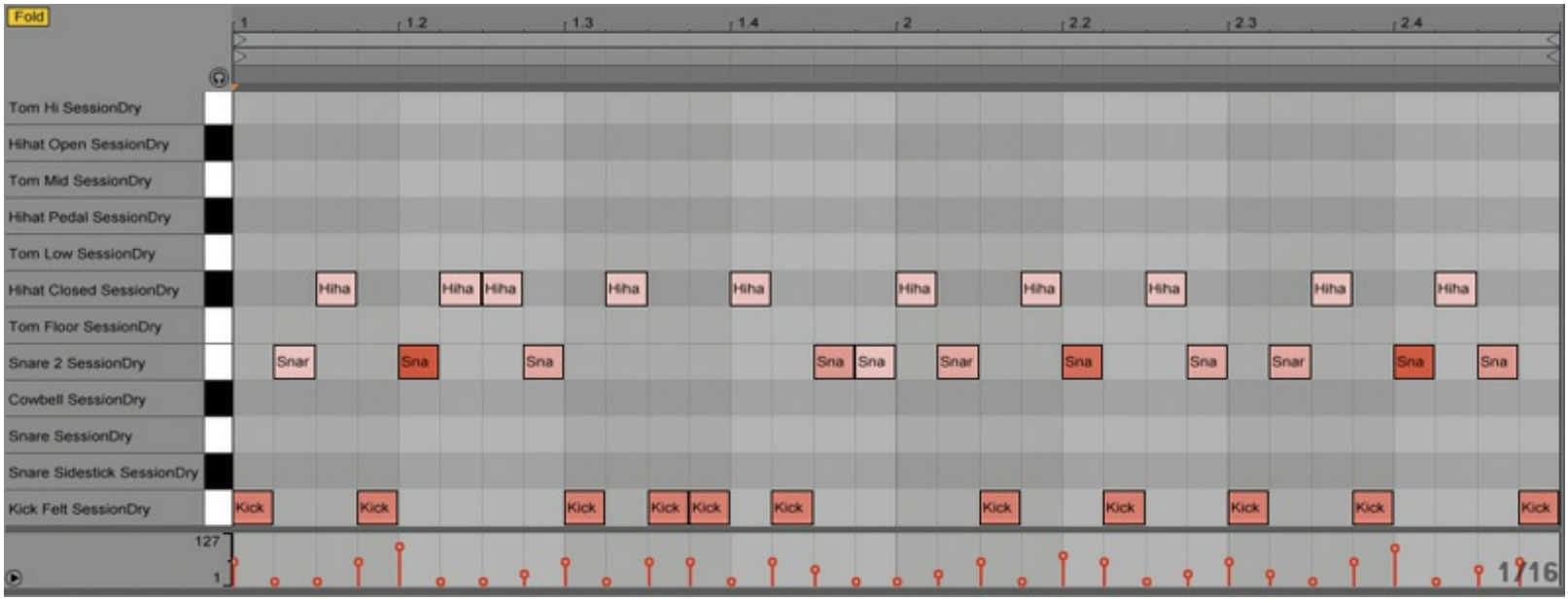


In the above pattern, sub-patterns in groups of three notes play against the expected flow of time, before “correcting” with a two-note pattern at the end of the measure.



These types of unusual note groupings treat each instrument as equally weighted, rather than using the overlaying of multiple voices to emphasize particular beat positions. Interestingly, patterns like this tend to be perceived very differently at different tempos. At a fast tempo such as 170 bpm, this pattern takes on a “rolling” character, and the composite line—the perceived fusing of the separate parts—becomes much more audible in relation to the individual notes. Linear patterns like this are common in drum and bass grooves, for example.

It’s also potentially interesting to add an additional level of restriction. You might decide, for example, that your pattern will be based entirely on sixteenth notes, and that every sixteenth-note position on the grid must contain a note. A complex two-measure version of such a pattern might look like this:



As with the previous pattern, the linear approach here helps to "democratize" the instruments. No one voice has dominance over the others, and the traditional roles of timekeeping elements vs. accenting element are eliminated. Also, notice that there's a lot of variation in note velocities in this example. Just by changing the volume of certain notes in relation to the others, you can make a single pattern of notes take on an entirely different character.

Linear drum patterns aren't right for every musical context, but they can offer a different way of thinking about beat programming that may help you come up with new patterns and variations that you might otherwise not have considered. And of course, as with all of the suggestions in this book, there's no reason to restrict yourself to strictly following the monophonic requirement. If a linear pattern would sound better with some instances of multiple simultaneous voices, go ahead and use them.

Programming Beats 3: Ghost Notes

Problem:

You're listening to a lot of acoustic drummers to get ideas for how to program more realistic drum parts, and you keep hearing very quiet sounds in between what are the clearly important hits. These are barely audible but still somehow sound intentional. And most importantly, they seem to make the groove work better. What are these sounds, and how can you use them in your own music?

In a lot of "purely" electronic music, there is often little to no variation in dynamics between notes played by a single instrument or even between all of the instruments. Each kick drum, clap, snare, etc. is generally triggered at the same volume. While this makes sense in many genres, it's quite different from how human drummers play, and if you're interested in making your drum parts sound more human, it may help to incorporate a technique drummers call ghost notes.

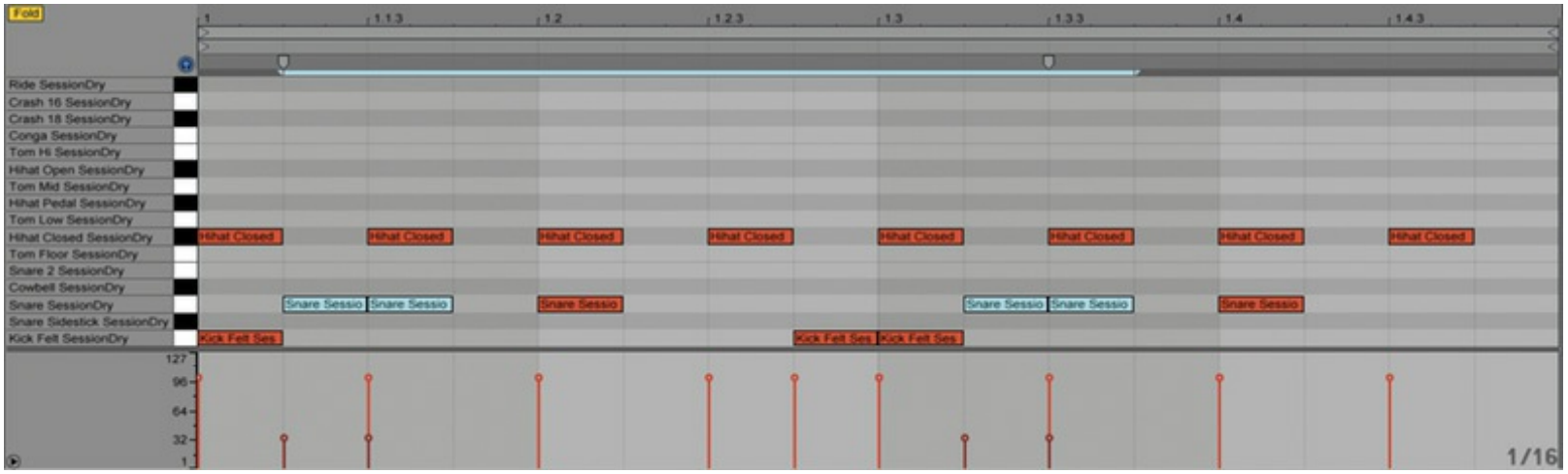
Solution:

Ghost notes are extremely quiet notes, normally played on the snare drum, that help to fill in the space between the prominent notes in the pattern. Ghost notes are nearly (but not completely) silent and are generally played as quietly as possible (although this varies in practice and between genres). For example, here's a basic one-bar rock drum pattern played without ghost notes:



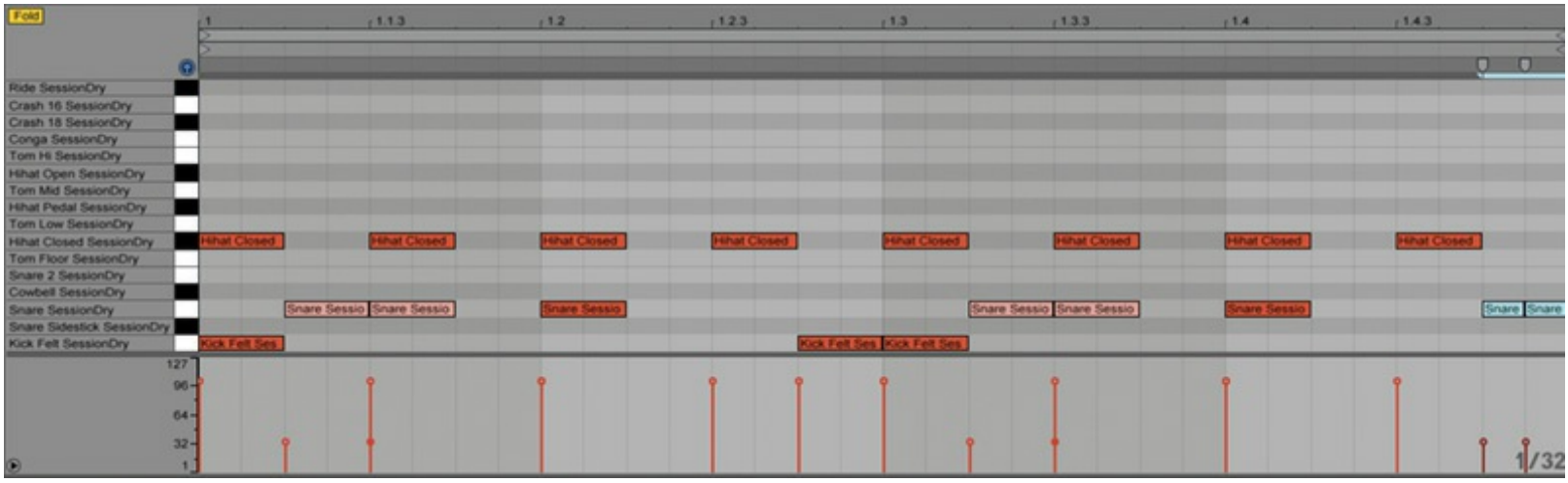
The hi-hat provides the regular “timekeeping” function, while the kick and snare provide accents that orient the listener to the place within the bar. This works fine and might be perfect for many musical situations. But in some cases, it might sound better if there’s a bit more activity in between the snare backbeats.

A drummer might play the pattern like this:



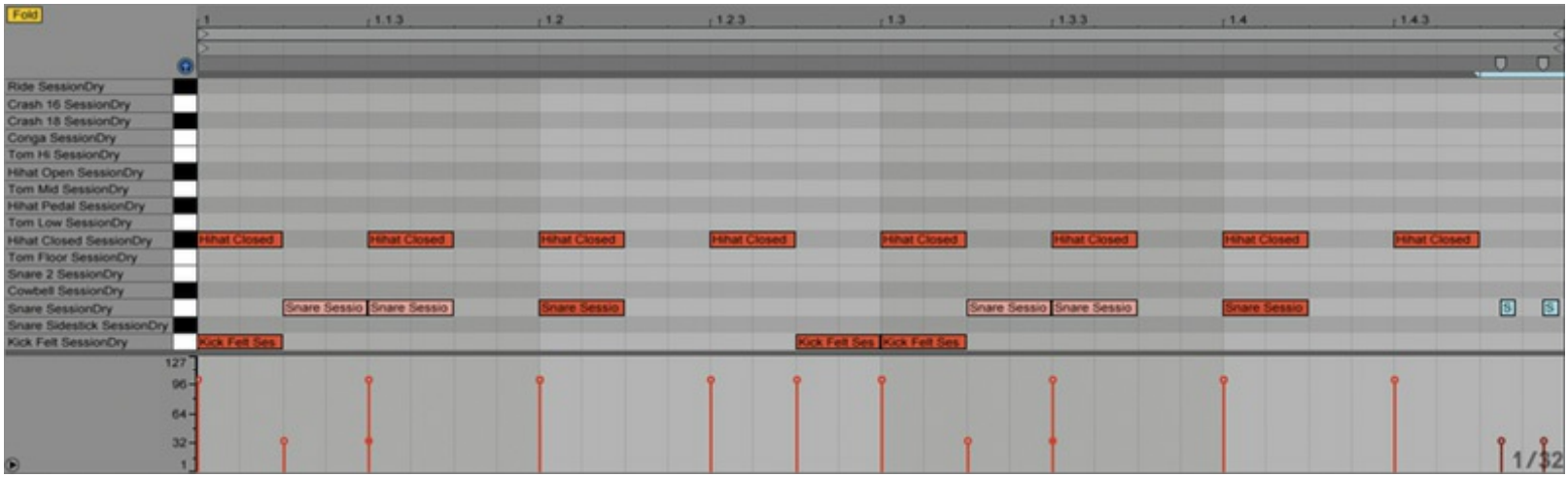
The additional low-velocity snare notes on the second and third sixteenths of beats one and three help to propel the musical time forward.

Another nice use of ghost notes is for adding very quick gestures that lead into or “anticipate” important structural points in the groove, such as the downbeat. Here’s an example:



The blue notes are the thirty-second notes that drive towards the downbeat and might be heard in combination with the downbeat's kick drum hit as a single musical gesture.

To exaggerate this effect even more, some drummers will even play these thirty-second notes quite a bit behind the beat (see [Programming Beats 1: On Looseness]):



Note that when programming ghost notes in a DAW, the actual musical result is entirely dependent on how velocity-sensitive the instrument is that's being triggered. If you're using a drum machine for example, there may be no velocity variation at all, in which case adding ghost notes will simply create a different pattern. Ghost notes are usually most effective if you're triggering multisampled acoustic drums, in which there are actually low-velocity samples that will respond the way a real drum

would respond. But of course, these techniques can also work with entirely electronic sounds as well. And although drummers usually play their ghost notes on the snare, you can also experiment with moving your ghost notes to an entirely different instrument.

Programming Beats 4: Top, Bottom, Left, Right

Problem:

From listening to a lot of music, you have a general understanding of how to program beats that sound similar to those in the music that inspires you. But you don't really have a sense of how the various drums in a drum kit relate to each other or the way human drummers think when they sit down at the drums and play. As a result, you're concerned that your programmed beats are either too mechanical sounding or are simply the result of your own interpretation and guesswork about what you hear in other music.

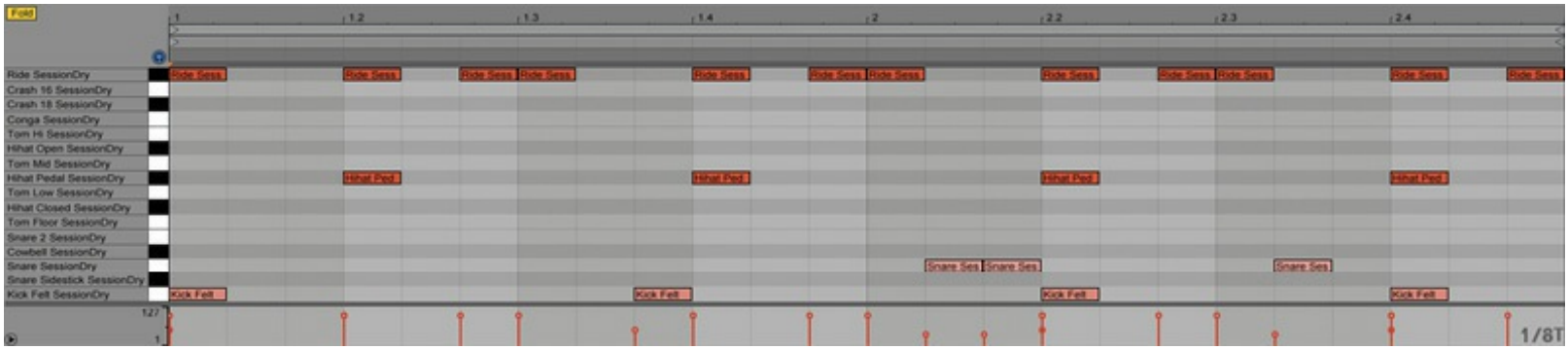
Even if you have no intention of writing "human"-sounding drum parts, it can be helpful to understand some of the physical implications of playing a real drum kit. Here are some ways that drummers approach their instrument.

Solution

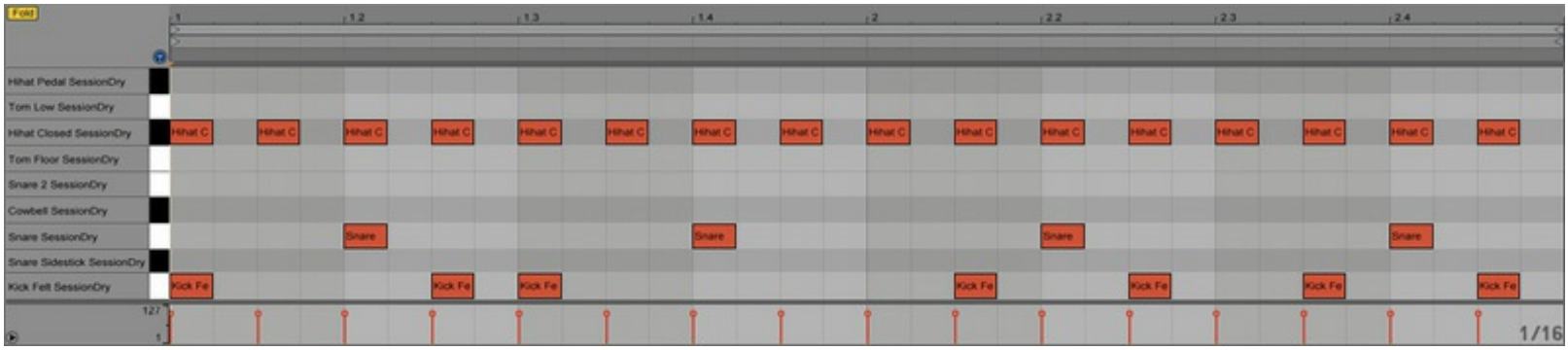
At a philosophical level, a drum kit can be thought of as divided into top and bottom halves. The top half includes all of the cymbals: the hi-hat, ride, crashes, and possibly more esoteric cymbals like splashes, Chinese cymbals, gongs, etc. These are the "top" half for two reasons: They're both physically higher than the drums, and they also occupy a higher range in the frequency spectrum. In contrast, the bottom half is the drums themselves: the kick, snare, and toms. (The snare is a special case and can be thought of as somewhere in between the top and the bottom in frequency. But for our purposes, let's consider it part of the bottom group).

Drummers tend to unconsciously approach beat making from either the

“top down” or the “bottom up,” depending primarily on genre. Jazz drumming beats, for example, are generally built from the top down, with the ride cymbal pattern being the most important element, followed by the hi-hat (played by the foot). In this context, the kick and snare drum serve to accent or interrupt the pattern which is established by the cymbals. A typical jazz drumming pattern might look like this:



In contrast, rock, pop, or R&B drumming beats are built from the bottom up, with the interplay between the kick and the snare comprising the most important layer and the hi-hat or ride cymbal patterns serving as a secondary element. A typical rock drumming pattern might look like this:



Note that in both jazz and rock beats, the cymbals generally play simple, repeating patterns, while the kick and snare play gestures that are more asymmetrical. But in jazz, those simple cymbal patterns are fundamental signifiers of the genre. In rock, the cymbal patterns are secondary in importance, while the asymmetrical kick and snare gestures are what define the music.

An awareness of these drumming concepts might give you some things to think about when writing your own electronic drum parts. Are you thinking from the top (cymbals) down, or from the bottom (kick and snare) up? Is the genre you're working in defined by repeating patterns (such as the steady four-on-the-floor kick drum of house and techno) or by asymmetrical gestures (such as the snare rolls used for buildups in trance)?

In addition to the top/bottom dichotomy, drummers also must make decisions along the left/right axis when determining how a particular pattern is divided between the left and right hands. On a drum kit, some of this is determined by the physical location of the instruments. But for an instrument like a hi-hat that can be reached by either hand, there is often a subtle difference in sound depending on how the pattern is played. For example, consider the following beat:



At slow-to-moderate tempos, most drummers would probably play the hi-hat part with one hand, leaving the other free for the snare drum. But once the tempo becomes too fast, it's no longer possible to play a continuous stream of sixteenth notes with one hand. At this point, many drummers would switch to playing the hi-hat with alternating sticking, each stroke with the opposite hand. But this requires some compromises: Beats two and four require both hands to be playing together, so the player must either move one hand very quickly between the snare and hi-hat or play at least two consecutive hi-hat notes with

the same hand. In both cases, there will likely be a slightly different resulting sound. Even the subtle physical differences between two drumsticks can result in a different sound versus when a pattern is played with a single hand.

Of course, none of these physical restrictions apply to the electronic domain by default. There's no inherent physical speed limit and no need for any notion of "alternating stickings." At any tempo, consecutive notes can sound completely identical if that's your intent. But if you'd like to apply some of the sonic characteristics that come about as a result of these human restrictions, you can do so manually. For example, you could try creating a very small change in velocity for every other note in a repeating pattern. Or with a bit more work, you could actually use a slightly different sound for every other note. Some software samplers have a feature called "round robin" that automatically plays a different sample with each key press.

Thinking like a drummer can be a useful exercise when writing beats for any genre—even ones that have no overt relationship to acoustic music at all.

Bass Lines and Kick Drums as a Single Composite

Problem:

You've learned all of the engineering tricks and technical solutions for a clean, well-defined low end. You use sidechain compression on your bass lines and EQ to remove unnecessary low-frequency information on every track besides kick and bass. But somehow, you're just not able to achieve the same clarity and definition in the bass as the tracks you admire. You know you're doing everything right technically, but things still sound muddy.

One of the most commonly discussed areas in music production is how to deal with issues in the low end. The solutions provided are almost always on the production side:

- Use EQ to create space for the bass in the kick drum's track, and vice versa.
- Use EQ on every other track to remove low end, which makes room for both kick and bass while reducing overall signal level.
- Use sidechain compression to automatically reduce the gain of the bass when the kick is sounding.

But what's discussed much less frequently is how to solve these problems on the musical side. Here are some compositional ideas for improving the clarity of low end.

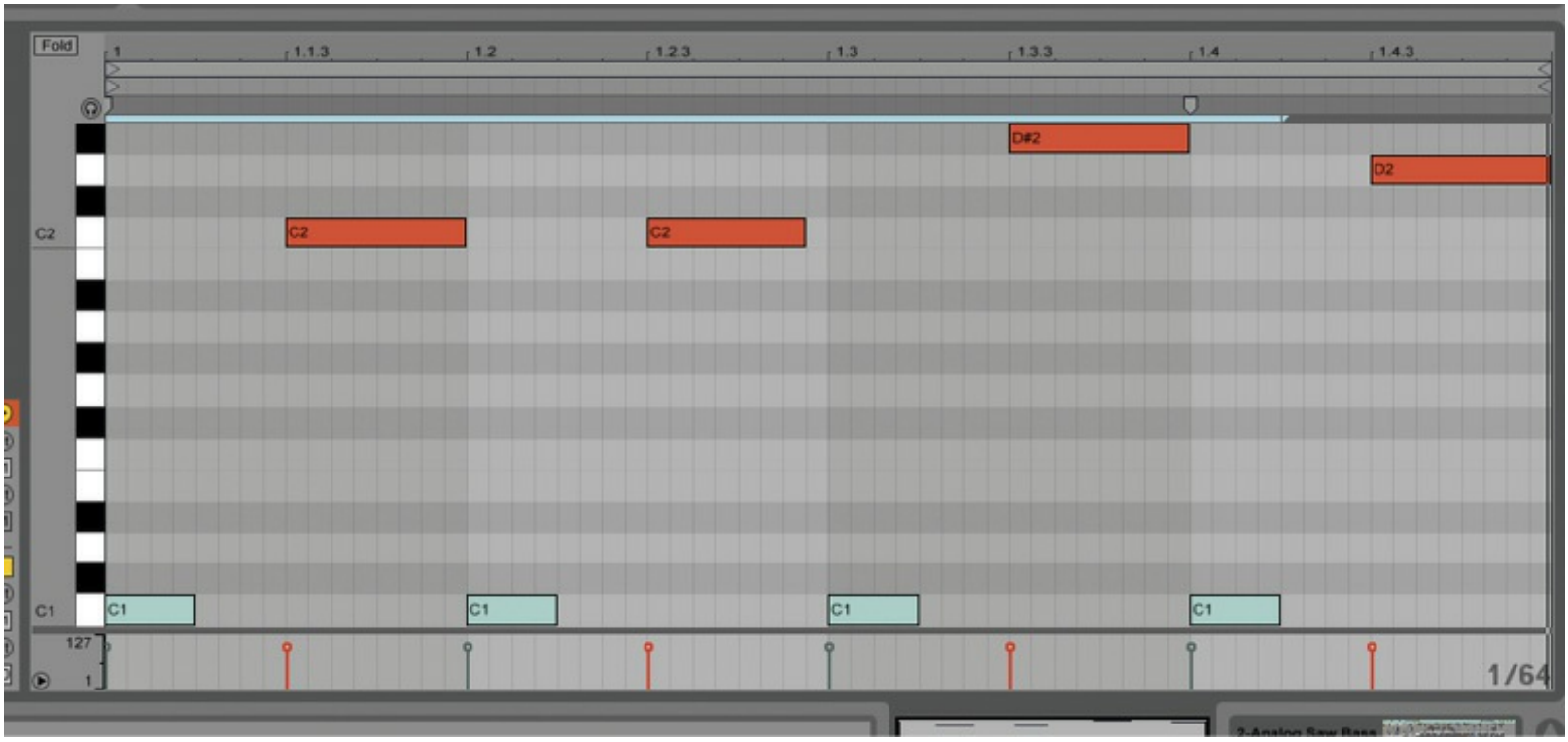
Solution:

Rather than thinking of your kick and bass as two separate musical lines, try to think of them as a single composite monophonic line. In this scenario, you not only can't have more than one bass note playing

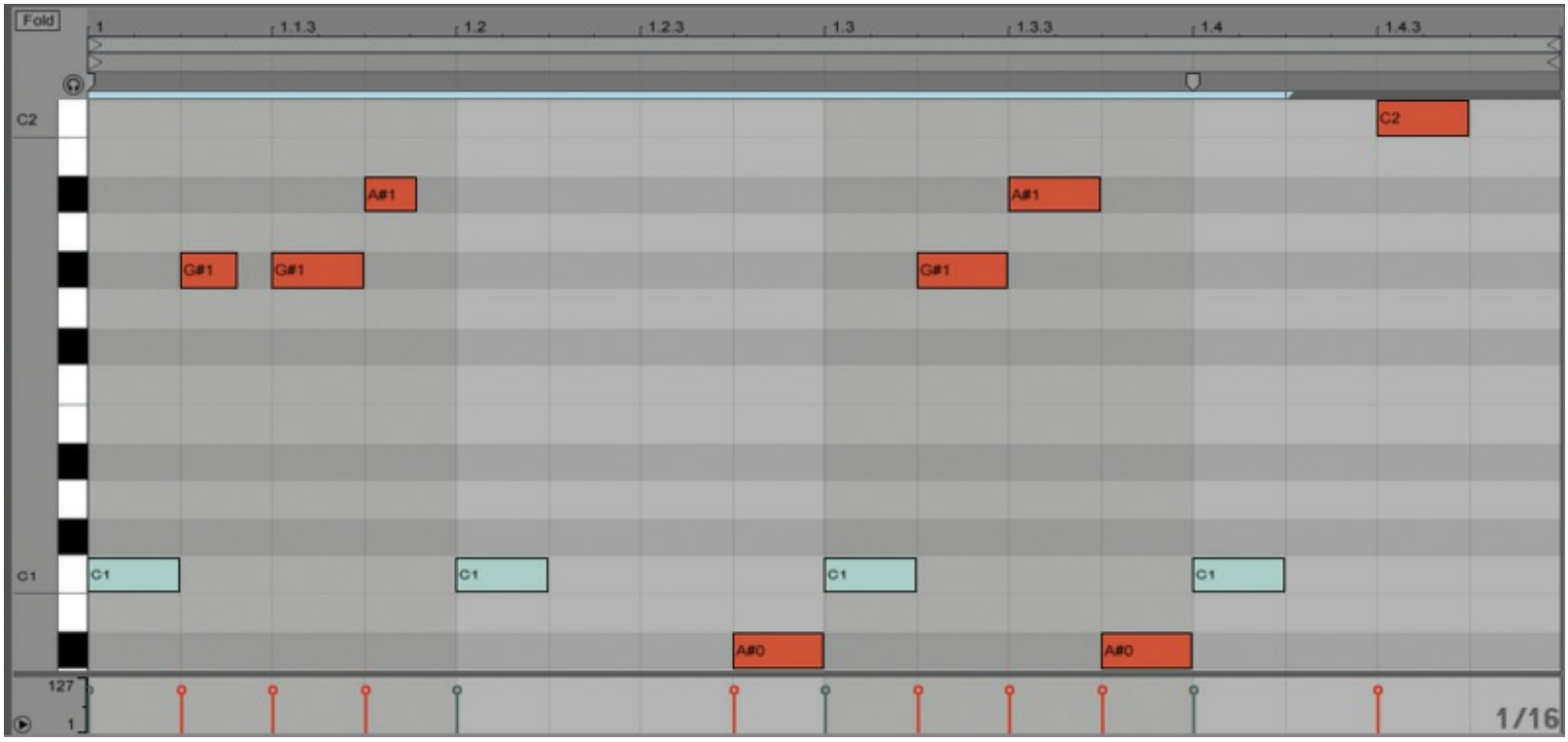
simultaneously, you also can't play a bass note at the same time as the kick. This requires you to make compositional decisions; how can you create a bass line that fits between the notes in your kick drum pattern (and vice versa)?

In some genres (such as trance and some flavors of commercial EDM), bass notes are commonly placed only on offbeats, while the kick drum notes occur on every downbeat. This is a simple way of creating a single composite line and is a defining characteristic of music in these genres.

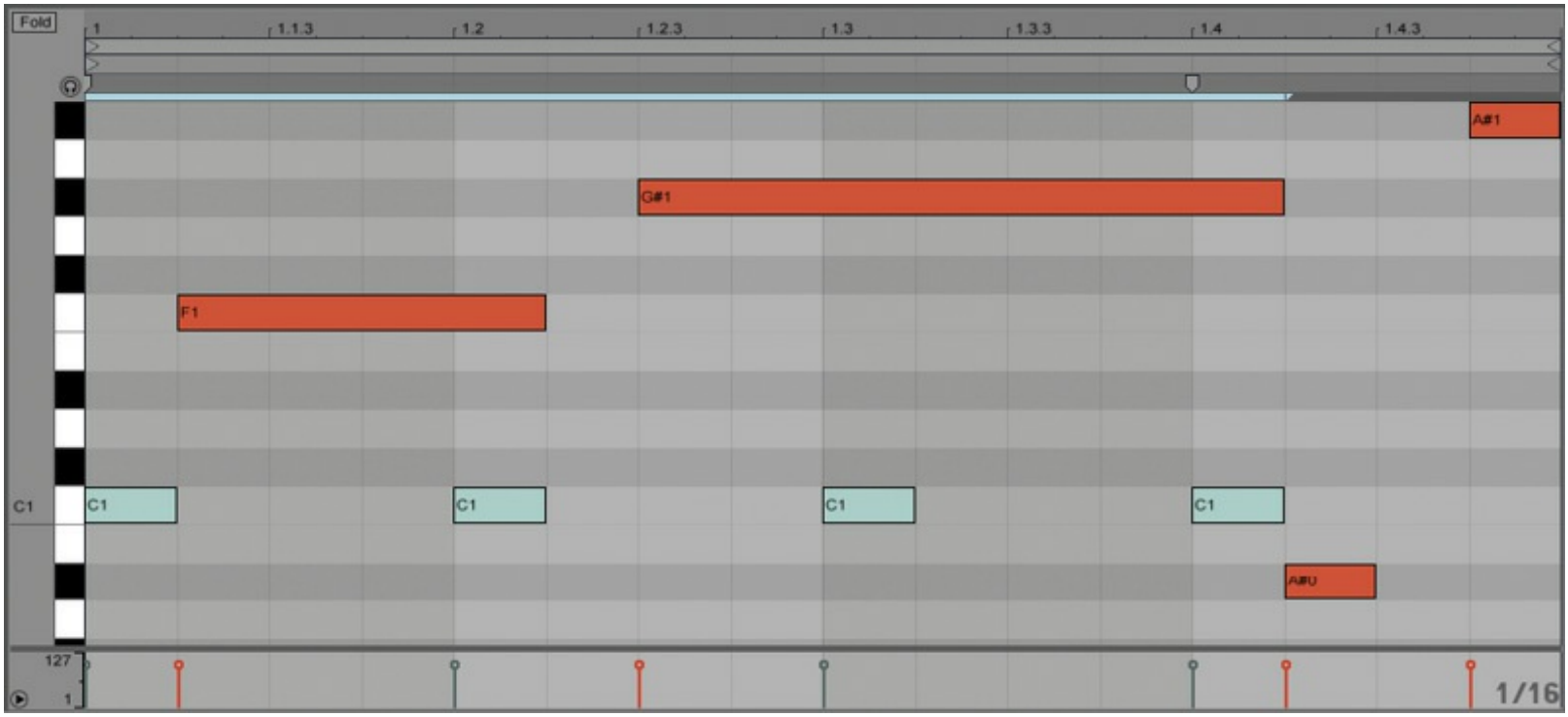
The image below is an example of a characteristic trance bass line (the red notes), with the kick drum pattern added below (the blue notes).



Other genres have less strict "rules" governing the relationship between kick and bass drum. In techno or house, for example, the kick drum is still often placed on every beat, but bass lines might be much more rhythmically intricate. Here's an example of a more detailed bass line (again shown alongside the kick drum notes) that still adheres to the notion of the kick and bass elements forming a monophonic composite.



Notice that, in both of the above examples, neither the beginnings nor the endings of the bass line notes ever overlap the kick drum notes (or each other). Provided your kick and bass sounds are short, these patterns should create a very clean, truly monophonic composite, which should help maintain sonic clarity in the low end. But the duration of sounds can have a big impact on the effectiveness of this type of writing, and what you see in your patterns may not represent the reality of what you hear. For example, 808-style kick drums often sustain for a long time, so even though you've programmed short notes in your MIDI editor, the resultant sounds may still overlap, causing low-end interference. On the other hand, there may be some situations in which zero overlap can sound clipped and unnatural, and sustained notes make more musical sense. In these cases, you can still achieve a lot of the benefits of this composite technique by at least making sure that the attacks of the notes never overlap. Here's an example of long, sustained bass notes but with the note onsets themselves still carefully composed to stay out of the way of the kick drum line.



If, after applying these techniques, you find that you're still having problems with low-end clarity, you can always go back and apply the production tips mentioned previously. And since you've already composed your way out of the most severe issues, techniques like sidechain compression and low-cut EQing may now have a much more pronounced effect.

~ Note: In the examples mentioned here, the kick drum pattern has been added purely as a visual reference. But it might make sense for you to actually do this in your own music by copying the notes of the kick drum clips, pasting them into your bass clips, and then deactivating or muting them so they don't make sound. This creates a clear visual relationship between the bass and kick lines, making it easier to write as if they were a single line.

Drums to Pitches and Vice Versa

Problem:

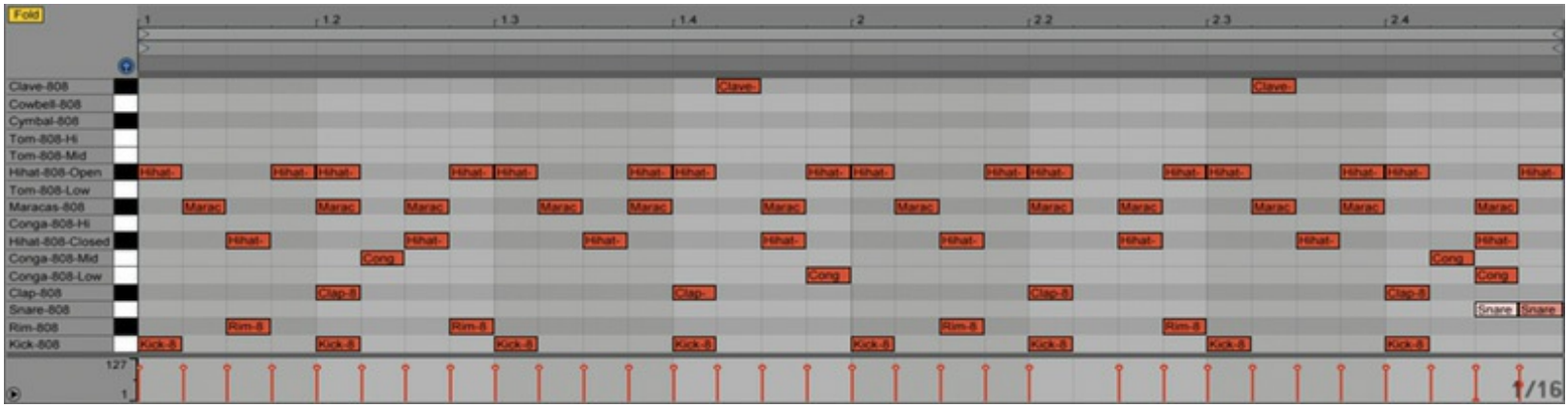
Sometimes you have great ideas for drum patterns but aren't able to find melodies or harmonies that fit. Other times, you have the opposite Problem You come up with great pitched ideas but can't write drum parts that work with them.

It's common to write bits and pieces of songs in no particular order. In fact, this kind of Goal-less Exploration can be a great way to spend creative time without the pressure of needing to finish a song. And ideas created in this way can often be saved as Scraps and Sketches to be revisited later. But sometimes, you really do want to finish the song but only seem to be able to compose one type of part.

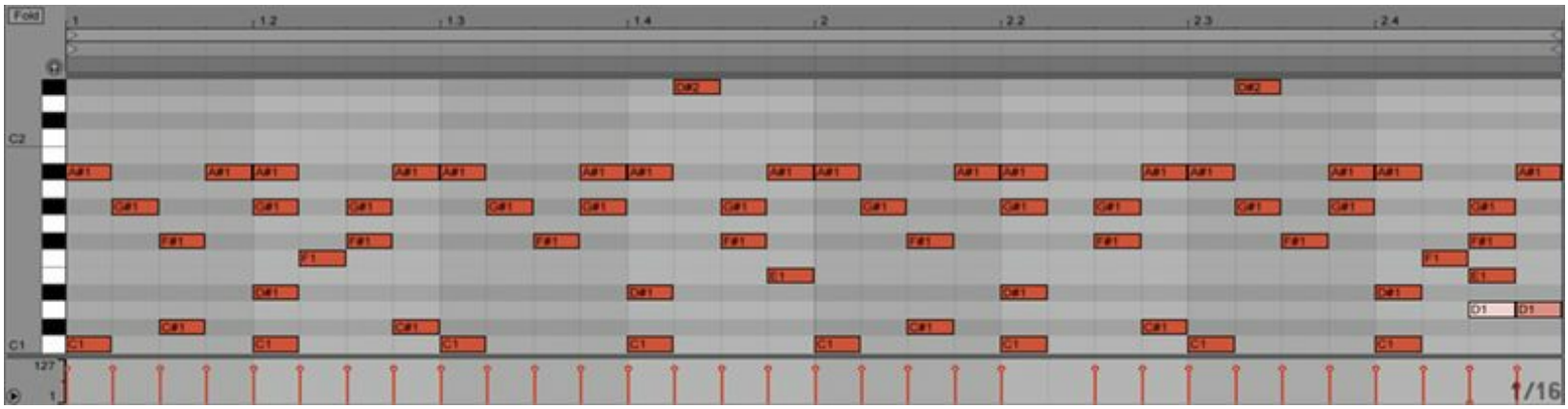
Solution:

One of the greatest benefits of working with MIDI is that the notes of a musical idea are a completely different type of data than the sounds triggered by those notes. This means that the notes can be repurposed to play an entirely different sound. It's common to create a bass line, for example, and then try it out through a number of different bass sounds before finding the one that fits the best. What's less common, however, is using the MIDI notes created for one instrument to trigger a completely different category of instrument. And doing this can yield some really interesting results.

For example, let's take this drum pattern, which might be appropriate for house or related genres:



By simply copying this clip to a track containing a pitched instrument, we end up with this:



The resulting notes are “mapped” from the notes that triggered the appropriate drum sounds in the original clip. These sound a bit strange when played as they are. They’re quite low in pitch, so the resulting chords sound somewhat thick and muddy. Also, the actual pitches aren’t particularly harmonious (although this might be exactly what you want, depending on the music you’re making). But the pattern is rhythmically and structurally interesting, and there are a number of things we can do to make the notes work better (see Creating Variation 3: Note Transformations), including simply transposing the whole pattern to a higher octave. Another way to get the notes to work together better is to selectively transpose some of them—either manually, by dragging them to new pitches, or automatically, through the use of some sort of scale-correcting MIDI effect (if your DAW has one). Now, in just a few steps, you’ve created a melodic or harmonic pattern that’s musically

related to the drum pattern (because it's rhythmically identical) and which might be useful as an idea in your song.

Of course, you can also follow this process in the other direction. If you have an interesting melodic or harmonic part, try copying the clip to a track containing drums. You'll probably end up with a lot of strange drum choices, but you'll likely get an interesting rhythmic pattern. By transposing the notes so that they play more appropriate drums, you might end up with an interesting drum part that you never would have written otherwise.

Some DAWs and plug-ins allow you to analyze recordings of audio and extract the pitch and rhythmic information into new MIDI clips. While this functionality is normally used to extract specific types of audio recordings into similar types of MIDI recordings, you might get interesting results if you use the "wrong" algorithm when converting. For example, if you're converting a piece of polyphonic audio, try telling your conversion utility that the material actually consists of drums instead. The tool will then try to figure out the pitches implied by the drums, which will likely result in harmonies that are completely unexpected (but which might be musically interesting).

What's exciting about these possibilities isn't just that you can create lots of new ideas from a small amount of source material, but also that the resulting ideas will be musically connected to their sources. While it might be boring to simply play the same rhythmic idea in the drums and a pitched part simultaneously, having real musical connections between the two allows you to alter one or the other in a way that makes the new part unique but still identifiable as a "descendant" of the original part.

Tuning Everything

Problem:

You're happy with all of the parts of your song, and the arrangement feels right. You're happy with all of the sounds as well—but only when you listen to them in isolation. When you listen to everything together, the energy and power that you hear in the individual parts seems to get blurred or lost. You've done all of the EQing and production tricks to make space for the various parts in the mix, but it still isn't working.

Sometimes there are musical factors, rather than production ones, that greatly affect whether or not a mix "gels" into a single, cohesive sound. One of the most important—and most overlooked—musical considerations is tuning. Here are some tips for ensuring that all of the elements in your mix are in tune with each other.

Solution:

Producers are generally pretty careful about making sure that the overtly pitched elements in their tracks are in tune with each other. Except in some experimental genres where unusual tuning relationships are a fundamental part of the musical concept, things like bass lines, harmony parts, and melodic elements sound terrible when their tunings don't match. But producers are often less careful when it comes to tuning elements like drums, which typically have a less definite pitch. In genres that use acoustic drums, this isn't usually an issue; acoustic drums are generally heard as unpitched. (The exception is toms, which often are tuned both to each other and to the key of the song.) But in electronic music, drum sounds are often generated via synthesis; kick drums are often made from pure sine waves, for example. And in these cases, drums may actually have a clearly audible pitch that should usually be

tuned to the other instruments in the song.

As mentioned earlier, electronic kick drums are the most obvious candidates for tuning, especially if they're made from sustained sine waves. And as with acoustic toms, electronic toms are also often clearly pitched. If you're using toms in your song, you probably don't want them all at the same pitch, so it might make sense to tune them in intervals that spell out the song's root chord. For example, a song in C minor might work well with the kick tuned to a C and toms tuned to an Eb and a G.

But even other percussion sounds with less obvious pitches can often still be tuned. Wooden and metallic sounds, such as sampled or synthesized woodblocks and cowbells, often have a clear pitch. And even cymbals, snares, claps, and other "noisy" sounds can sometimes be tunable. You may not hear a pitch when listening to these sounds for the first time, but over the course of a whole song, certain frequencies within almost any sound will start to come to the forefront and will likely be perceived as the pitch of that sound.

In situations where you're working with noisy sounds, the pitch may be hard to identify, but you may still hear that it's wrong. A spectrum analyzer, in conjunction with an EQ, can be really helpful in these cases. You can use the spectrum analyzer to find the frequencies that are louder than the average noise level, and then use EQ to either emphasize those frequencies (if they happen to be in tune) or attenuate them (if they're out of tune with respect to the harmonic and melodic parts).

If you've committed to tuning your drums to the key of your song, but you're writing a song that changes keys part way through, you'll need to make decisions about whether to retune the drums as well. In some cases, simply retuning the kick and toms might be enough. But you'll

need to carefully listen and think about how the drums work in relation to the other parts after the key change and adjust as necessary. In many cases, it may sound odd for the drums to suddenly be repitched. Ideally, drum tuning shouldn’t be noticeable to most listeners; drums that are in tune with the other instruments shouldn’t sound pitched. Instead, they should just make the whole song feel more like a single, cohesive entity.

Silence and Noise

Problem:

You understand that the fundamental components of music are sound, harmony, melody, rhythm, and form. But you're sure there's something else that's happening in the music that inspires you, and you can't quite find it in your own music.

It's common to think that the "music" is what you directly hear—the notes that play instruments that fill up what would otherwise be empty space. But a more interesting approach is to recognize and take advantage of the fact that silence and incidental noise can be made into fundamental parts of the musical texture. They can be as much a component of the sound of your music as a bass line.

Here are some ways that you can start to use silence and noise in your own music.

Solution:

Listen to other music and think about what's happening in the spaces between the notes. Is there ever actually complete silence? In most music, you'll probably find that at any given moment, there's always at least something you can hear. Even in sparse textures, there's usually a bit of reverb, delay, or other ambience that fills the gaps between notes. You can usually achieve this texture pretty easily by adding subtle spatial effects to certain elements.

But in some music, the space between the notes is truly silent. Listen, for example, to music by artists such as Atom™. Here, reverb is used sparingly, and each musical gesture seems to enter and exit against a

backdrop of empty space. The result is an overall texture that's extremely clean, surgically precise, and very "digital." To make music that evokes this feeling, you'll need to be very careful with spatial effects, as well as with the placement and duration of your sounds. Long sustained chords might not make sense in this context, nor will sounds with slow attacks or releases. It might help to think of everything as a short, percussive element.

At the other end of the spectrum, some music has no silence at all, with every gap between notes filled with some kind of ambience or noise that is so present that it's as important to the texture as the "intentional" sounds. Dub techno, as exemplified by artists such as Rhythm & Sound, is largely defined by this approach to space. Although the instrumental layers are quite sparse, there is a constant layer of noise that suggests the use of mysterious, ancient, broken equipment. This type of ambience might seem hard to achieve in modern, ultra-clean DAW environments, but here are some ways to create it:

> Record the sound of an open microphone in a quiet room (or even just a channel on your audio interface with nothing connected) and then dramatically boost the level of the recording. The inherent noise of mics, preamps, and audio interfaces can take on a new character when boosted enough to be heard as an intentional element in a mix.

> Sample the sound of a turntable's needle in the "runout" groove at the end of a vinyl record.

> Field recordings of almost any source can take on the character of abstract noise, provided they're processed in the right ways. Try applying lots of reverb to urban, factory, or nature recordings.

> Certain plug-ins are designed specifically to create artificial noise, although these can sometimes sound quite consistent (and thus artificial) unless heavily processed.

> If your DAW has this functionality, try creating feedback loops by sending the output of a return track back into itself, (carefully) adjusting the send level, and then recording the output. You can create very interesting types of noise with this technique—especially when using effects on the return track.

Whether you're aiming to incorporate noise as a compositional element

or are aiming for clinical, ultra-clean textures, it pays to think carefully about what's happening between the notes in your music and to make conscious decisions about how to make musical use of that space.

Sampling the Old-Fashioned Way

Problem:

You hear more and more music that uses samples in ways that sound like a deliberate under-leveraging of available technology. With characteristics such as “chipmunk vocals” and sampled breakbeats that sound like they’re pitched up, a lot of contemporary garage, footwork, and related styles use a set of techniques that sound like they’re taking advantage of the restrictions of vintage hardware. You’d like to be able to capture some of this nostalgic quality in your own music, but you can’t seem to figure out how it’s done using modern software tools.

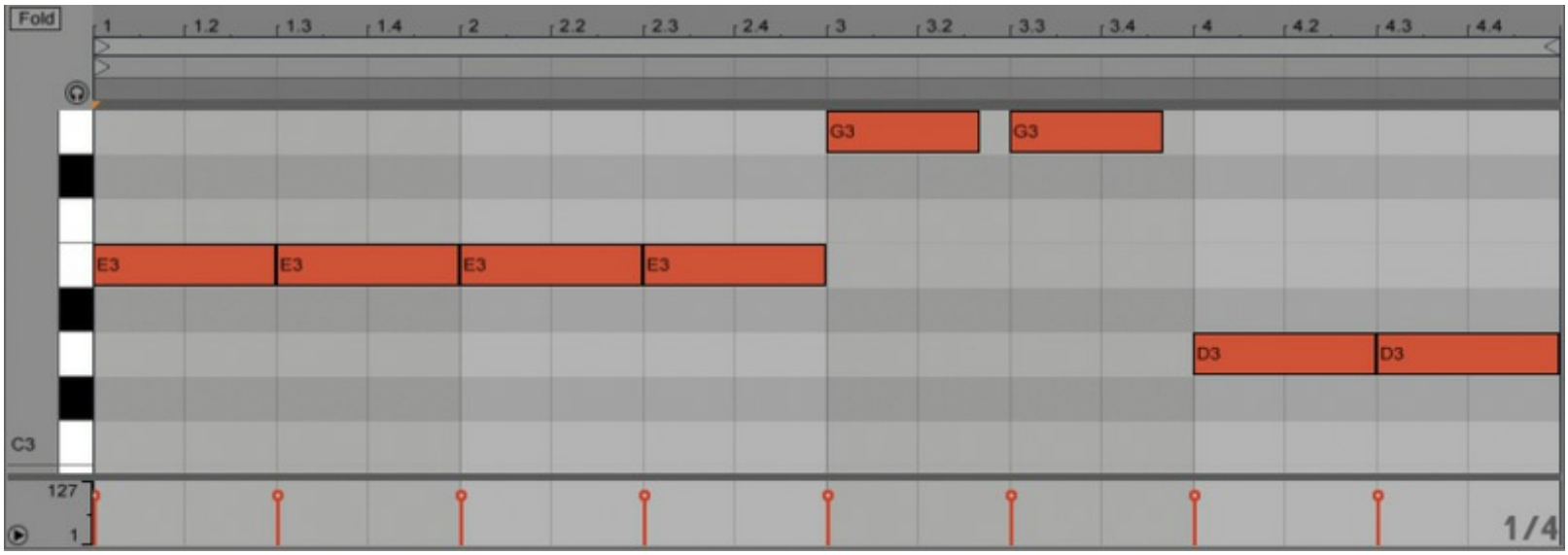
One of the most important technical developments in sampling technology was the advent of time-stretching: adjusting the pitch of a sample independently of its tempo (and vice versa). Prior to time-stretching, pitch and tempo were inextricably linked; the only way to make a sample play at a higher pitch was to play it faster. Modern time-stretching technology is great for preserving the realism of retuned samples and allows for considerable flexibility in reusing rhythmic samples at a range of tempos. But some of the artifacts and quirks in the older approach imparted a lot of interesting musical character. Here’s a technique for employing old-fashioned approaches to sampling in a creative way.

Solution:

In many modern DAWs, time-stretching is enabled by default for clips, under the assumption that this is the most flexible choice for most users in most musical situations. Although there is always a way to disable this functionality, either globally or on a per-clip basis, this usually just causes the clips to play back at their original tempo regardless of the

tempo of the song. To really see what can happen when time-stretching is eliminated, try playing the samples from a software or hardware sampler instead of placing them along your song's timeline. Now, samples played at different pitches on the keyboard will also play at different tempos. This can yield particularly interesting results when working with samples that have their own inherent rhythm (such as breakbeats or whole musical phrases).

The track "Leavin" by DJ Rashad uses this technique to great effect. The main hook in the track uses a sample of a two-note pattern: a dotted eighth note followed by a sixteenth. The four-bar phrase consists of this sample triggered every half note. For the first two bars the sample is at its root pitch (E). In bar three, it's triggered three semitones higher (G). In bar four, it's triggered two semitones below the root (D). In a sequencer, the played pattern might look like this:

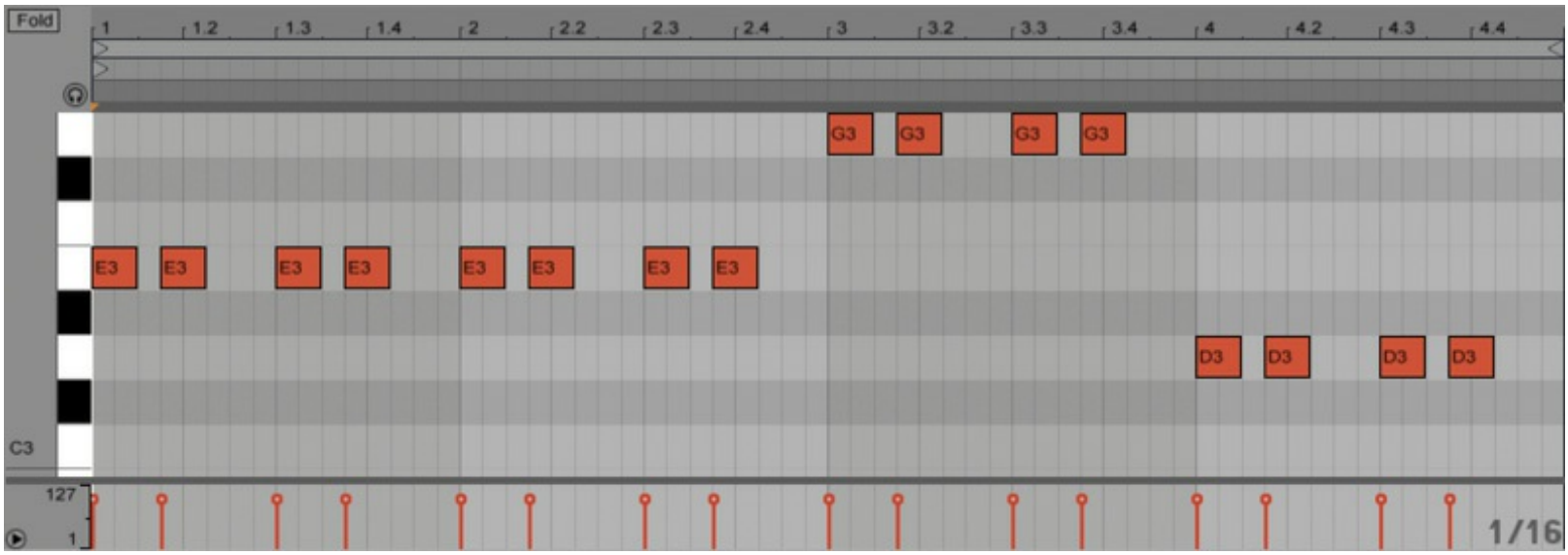


But because this is played using old-fashioned sampling technology, the transposed pitches play back the sample's two-note phrase at different speeds. What you hear is something closer to this:



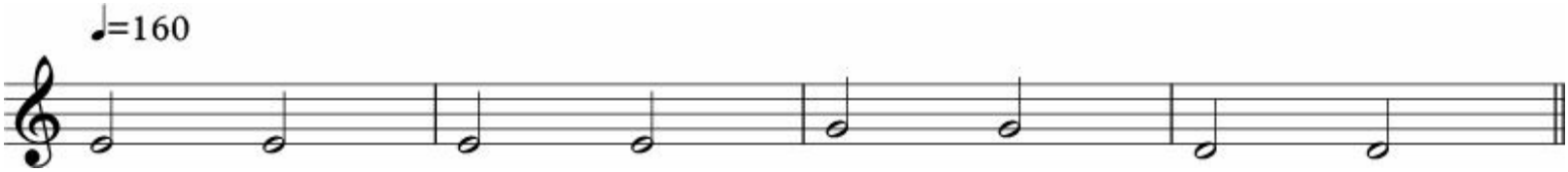
The first two bars play the dotted eighth/sixteenth pattern of the original sample. But the pattern in the next bar is transposed higher and thus plays faster—closer to the first two notes of a quarter note triplet. Finally the samples in the fourth bar play slower—closer to straight quarter notes.

If this sample were to have been triggered using a time-stretching sampler, there would be no rhythmic variation when transposing. The pitch would have changed, but the internal rhythm of the sample would have remained at a dotted eighth + sixteenth each time the sample was triggered. You'd hear something like this, which is much less interesting:

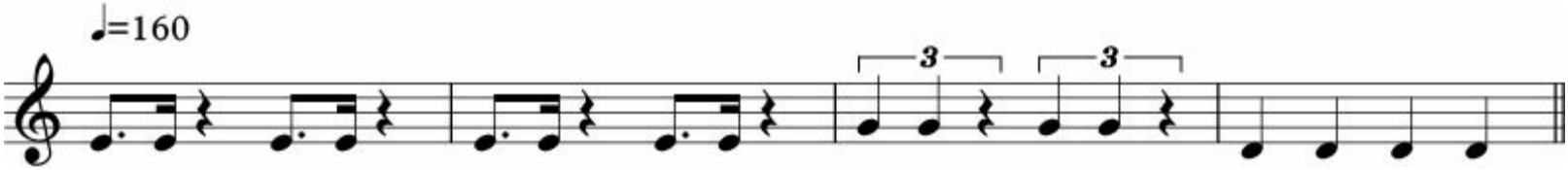


These examples may be easier to visualize in musical notation (but don't worry if you don't read music; these are simply notational representations of the three patterns shown previously).

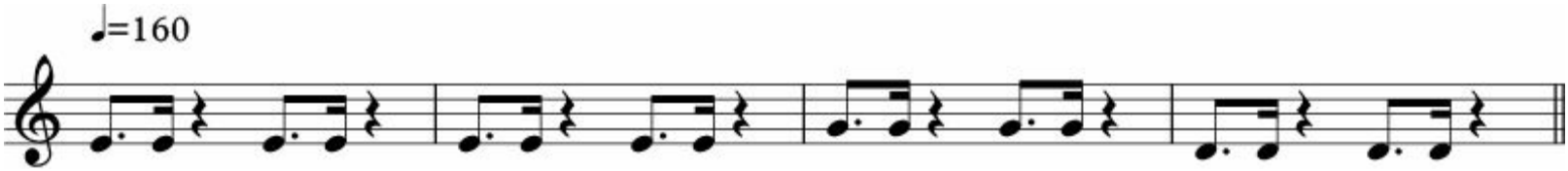
What is played:



What you hear (as a result of using old-fashioned, non-time-stretched sampling):



What you would hear (if a time-stretching sampler were used):



By deliberately using an old-fashioned approach to sampling, DJ Rashad made use of the limitations in the technology to create a more interesting musical result than he could have achieved by using a more modern approach.

Creating Melodies 1: Contour

Problem:

When you listen to music that inspires you, the melodies are strong, memorable, and hummable. In contrast, your melodies feel weak, aimless, and wandering.

Writing a good melody is challenging. Here are some ideas for writing better melodies by thinking about their contour.

Solution:

The term contour is used to refer to the shape of a melody over time—whether the notes rise to higher pitches, fall to lower ones, remain at the same pitch, or some combination of these. Additionally, we use contour to discuss whether a melody moves by adjacent notes (known as conjunct motion or motion by step) or by larger intervals (known as disjunct motion or motion by leap).

A few general rules:

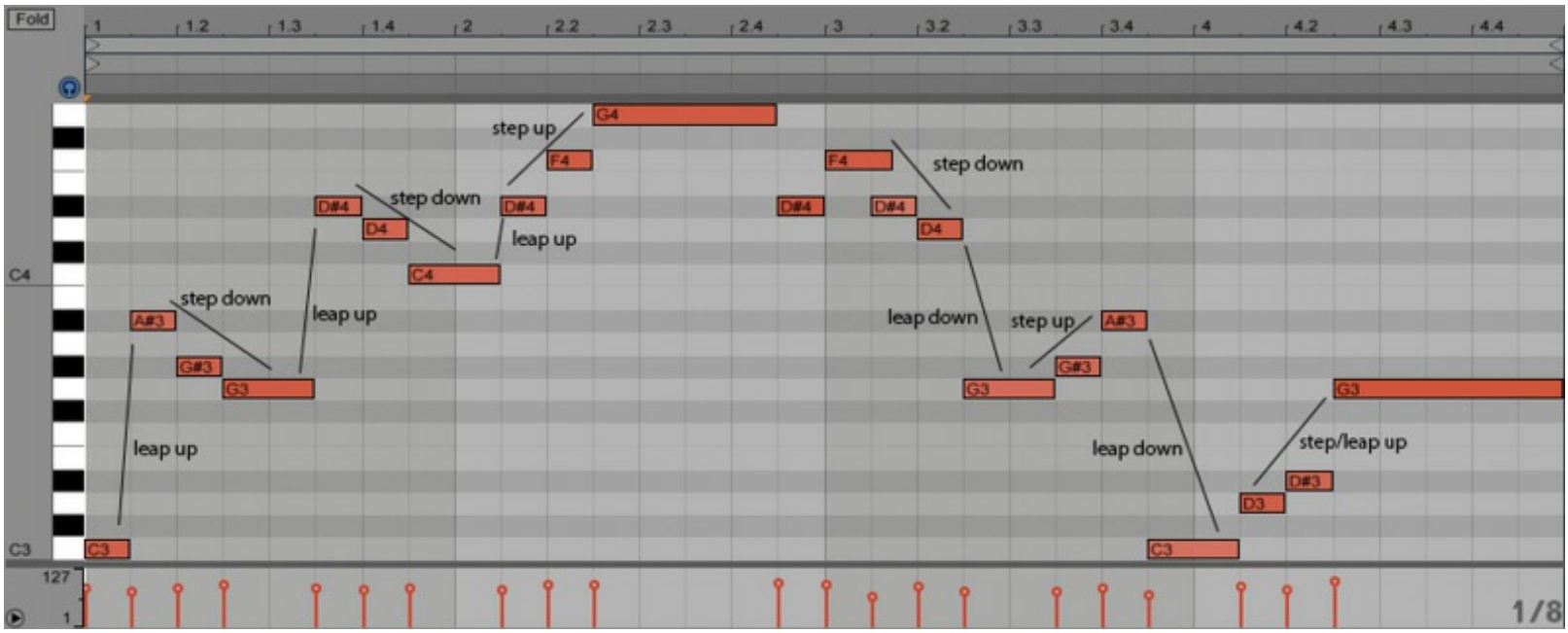
> Good melodies have a strong sense of balance between both aspects of contour: rise vs. fall and conjunct motion vs. disjunct motion. For example, if a melody rises for a while, it might make sense for it to then fall by roughly the same amount.

> If a melody has moved by step for a while, a good choice might be to then proceed by leap in the opposite direction. The converse also applies: Commonly, a melody that has moved by leap will then move by step in the opposite direction. One situation in which this might not be necessary is when a melody is simply an arpeggiation of a chord in the harmony. In these cases, the leaps between chord tones may sound "complete" even without stepwise resolution in the opposite direction.

> Good melodies often have a single peak note. That is, the highest pitch in the melody occurs only once. Furthermore, that peak note often (although certainly not always) occurs on a "strong" beat. (Assuming a 4/4 meter, the first and third beats are considered "strong" or "on" beats while the

second and fourth are considered “weak” or “off” beats. As you subdivide beats into shorter note values such as eighths and sixteenths, the same rule applies. That is, the odd-numbered subdivisions are perceived as stronger than the even-numbered ones.)

Let’s consider an example of a melody that (mostly) follows these rules:



The melody opens with a large leap up, which is immediately followed by two notes that step down. This pattern is then repeated in the next passage. Following this, we “break” the rules briefly to reach the peak note: A leap up is followed by two steps up. The second half of the melody also strays a bit more from the rules. Bar 3 begins with a descending stepwise figure which is then followed by a leap in the same direction. Next, we step up, and then leap down. The final gesture breaks the rules once again: We step up and then continue upward by leap to the last note.

The general contour of the melody is a kind of arc: We start from the lowest note in the passage and climb gradually to the highest note at roughly the halfway point of the melody, before gradually descending again to cover roughly the same amount of space in the second half.

As you listen to other melodies with contour in mind, you’ll find that

most (including the one discussed earlier) don’t follow these guidelines 100 percent of the time. But in general, you’ll find that the contour of most good melodies maintains a balance between ascent and descent and stepwise and leapwise motion.

Creating Melodies 2: Using Motives

Problem:

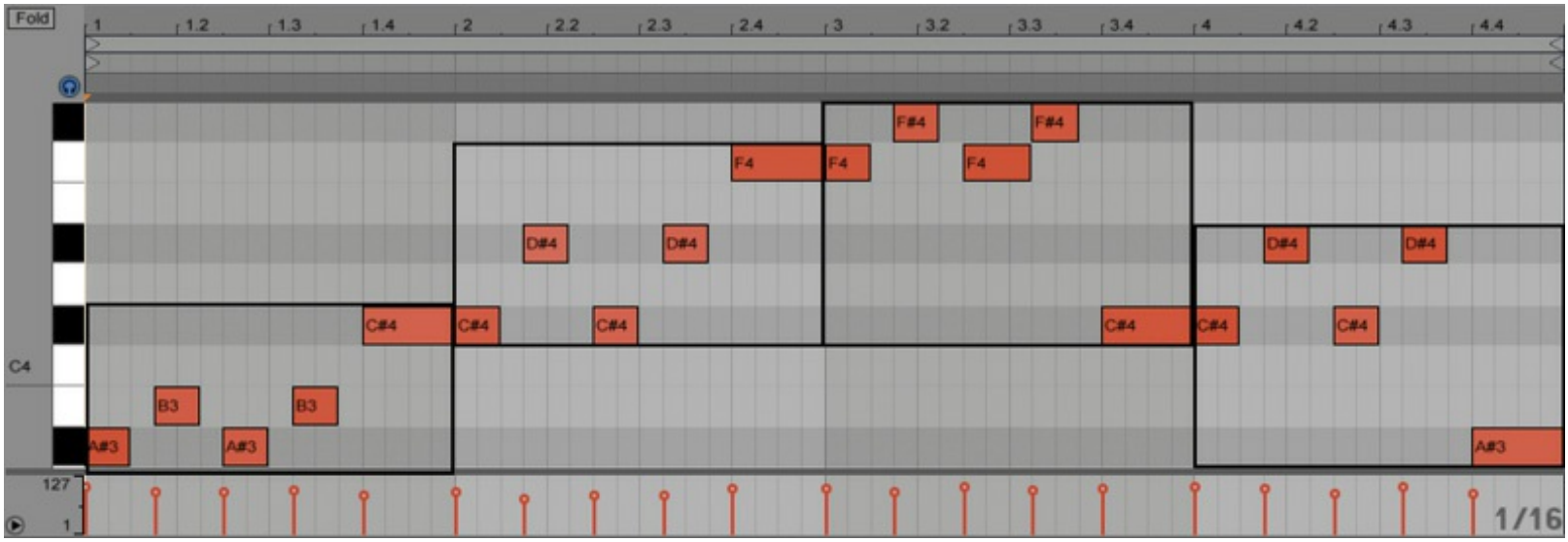
When you listen to music that inspires you, the melodies are strong, memorable, and hummable. In contrast, your melodies feel weak, aimless, and wandering.

Writing a good melody is challenging. Here are some ideas for writing better melodies by using motives.

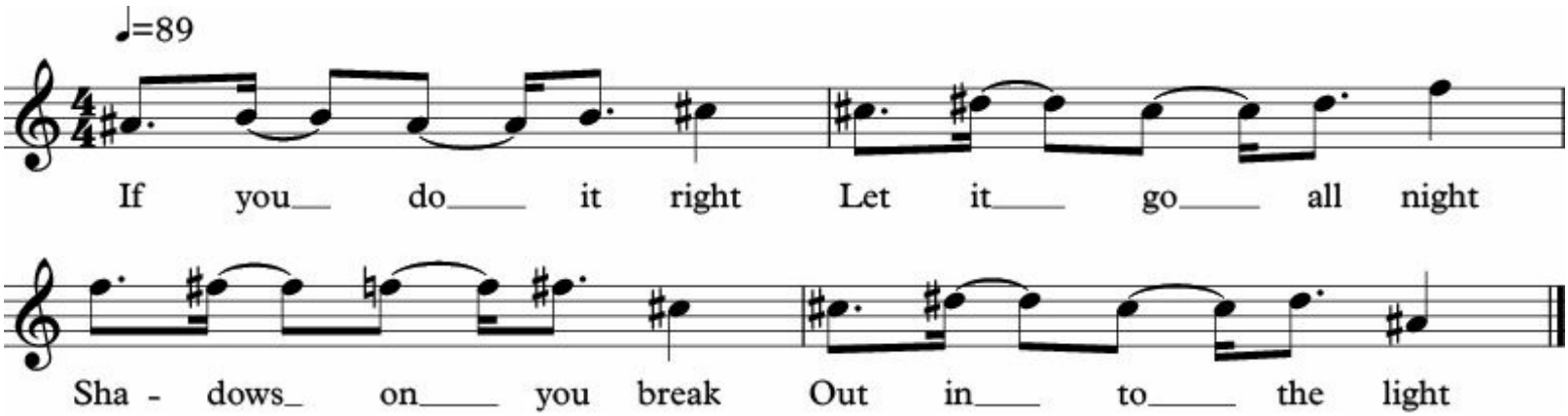
Solution:

Part of the reason writing melodies is so intimidating is that it's easy to imagine a melody as just a long string of notes without any internal logic. But melodies often have an inherent formal structure that is a sort of small-scale version of the formal structure used to construct a whole song. The smallest recognizable fragment of a melody is called a motive, and melodies are often constructed from just one or two motives. Motives are simple patterns that can be combined, repeated, and altered in various ways.

Here's an example of a melody from Daft Punk's "Doin' It Right" that's built from variations of just a single motive:



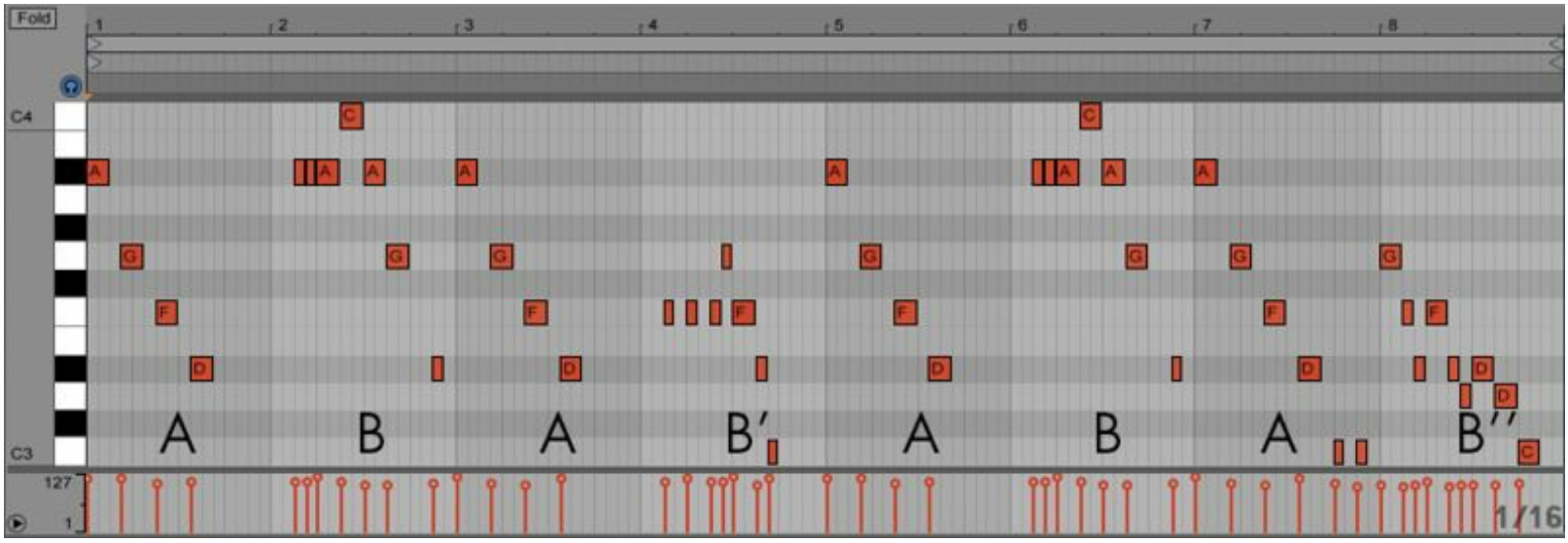
And here is the same example in musical notation:



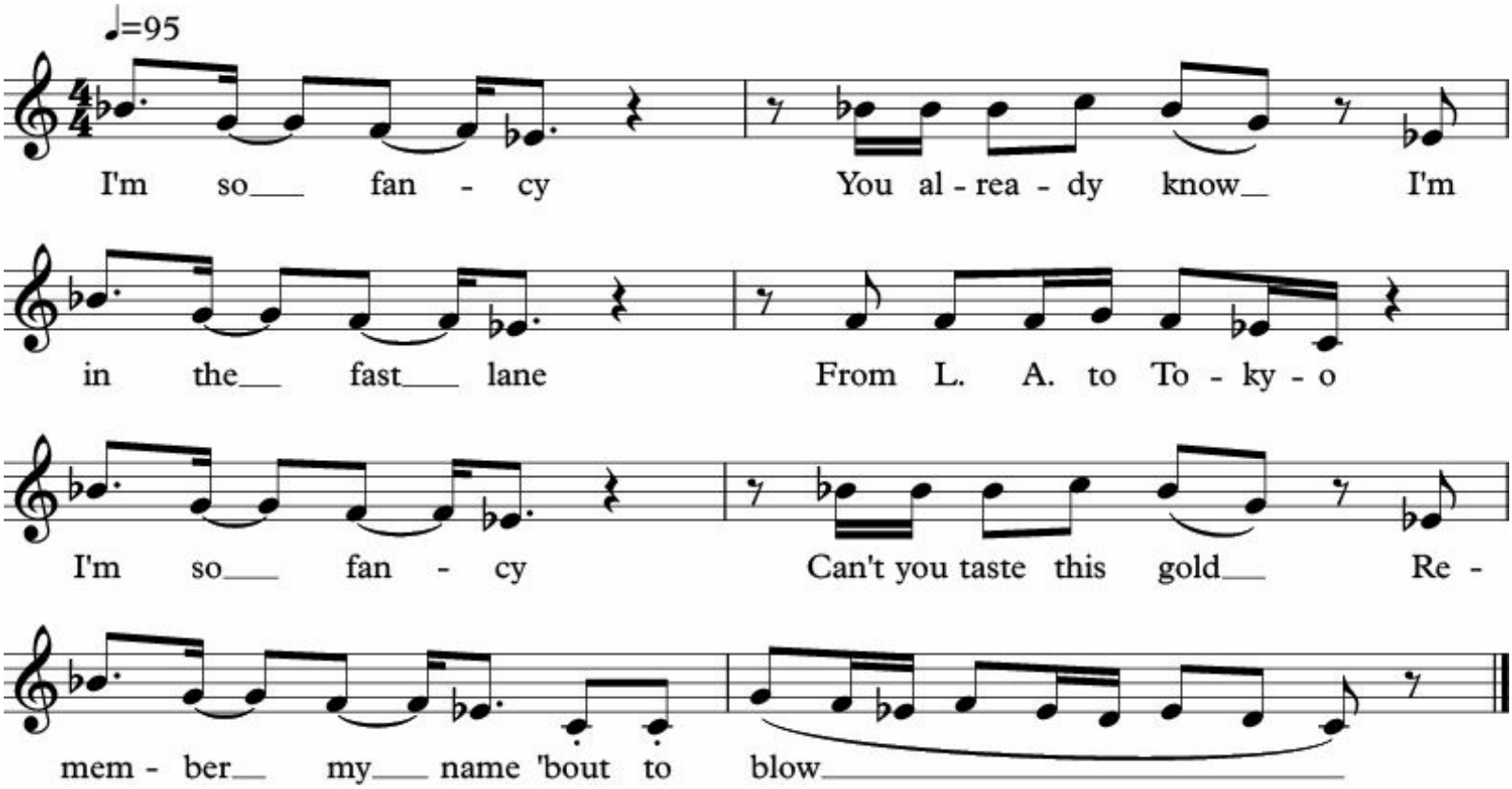
In the MIDI example, the boxes outline each of the four sections of this melody, and each is a variation of the same five-note motive. In the first two bars, the first four notes are an ascending stepwise motive of two notes, repeated twice, while the fifth note is another stepwise ascent. In the second two bars, we again have the initial four notes built from the repeated two-note motive, but the fifth note is now a leap downward. The contour of the whole melody forms an arc, beginning and ending on the same note (see Creating Melodies 1: Contour). In this example, a rich, usable melody is constructed simply by creating variations of a single five-note motive.

The melody in the chorus of Iggy Azalea's "Fancy" is a bit more complex but still uses variations of only two alternating motives. Here is

the melody as MIDI notes:



And here is the same example in musical notation:



Each bar contains one of the two motives and these alternate every other bar. In the MIDI example, these are labeled as A and B. The A motives are repeated identically each time they occur, while the B motives are subject to variation. Bar four (labeled B') is essentially just a transposed version of bar two, with a slight rhythmic embellishment. And bar eight

is just a more embellished version of bar four. Bar six is an identical repeat of bar two. The overall melody has a descending contour, as do each of the component motives.

As these examples show, there are many different ways to expand just one or two simple motives into a full melody through the use of variation. For some ideas about the kinds of variations you can use, see Creating Variation 3: Note Transformations. And identifying the motives in a melody can be a useful exercise when doing Active Listening.

Multiple Simultaneous Melodies

Problem:

In some music, you hear bass lines that are so detailed and interesting that they almost take on the quality of a melody. But when you try to do this in your own music, the parts collide and overlap in ways that just sound like a mess.

The combination of two or more intricate, independent, simultaneous melodies in a composition is called counterpoint. Here are some tips for how to use counterpoint effectively in your own work.

Solution:

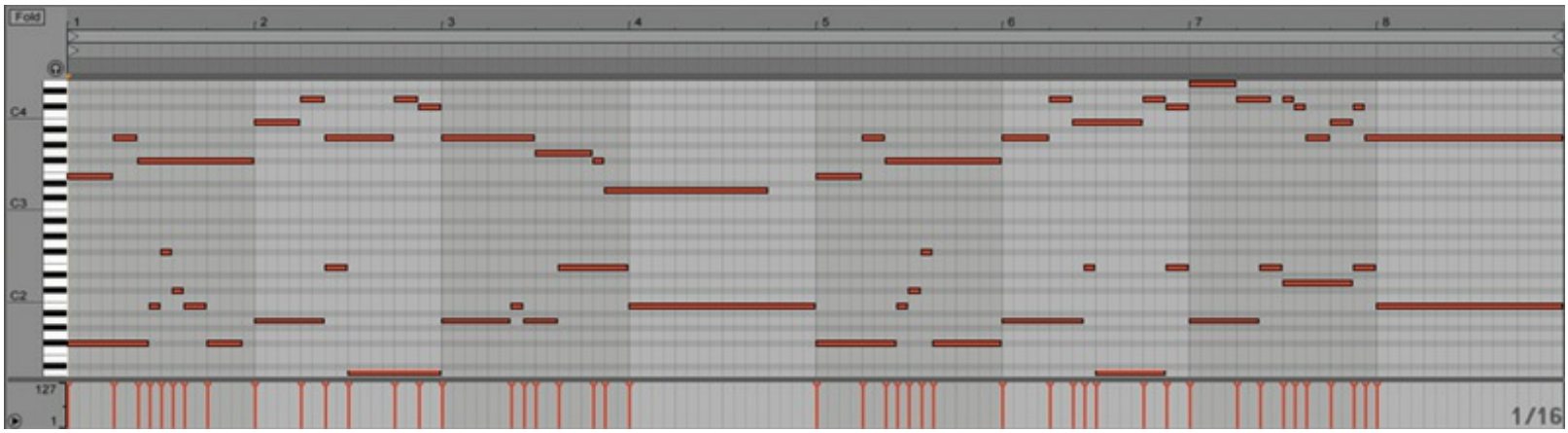
Counterpoint is an old practice, and music with independent melodic lines has been written and studied for hundreds of years. There are very strict rules for writing "correct" counterpoint in ancient styles of music, and although music today is generally less strict in all ways, we can still take advantage of some of these rules if we want to write interesting counterpoint in contemporary music.

One example of counterpoint that you've probably come across is the round, a type of composition in which the same melody occurs in multiple parts but starting at different times. The children's songs Row, Row, Row Your Boat and Frère Jacques are examples of rounds.

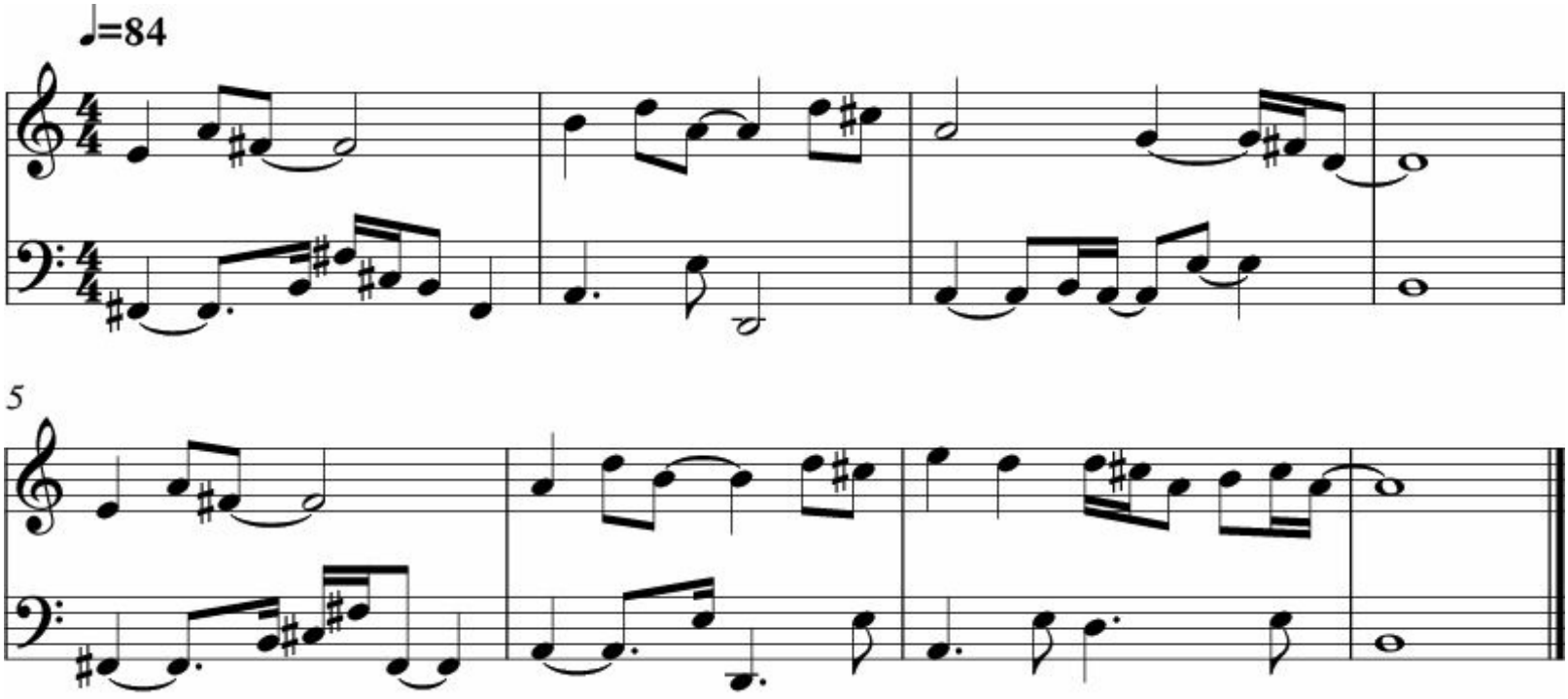
Technically, any music with multiple monophonic parts (such as a melody and a bass line) could be considered counterpoint. But in most music, there is one clearly dominant melody while the bass line is much more rhythmically static or serves only to support the harmony (or both). We wouldn't normally refer to this type of music as counterpoint

because the parts aren't given equal importance. True counterpoint (or at least good counterpoint) assumes that any of the parts involved in the counterpoint could function as a standalone melody even without the presence of the other parts.

An interesting example of counterpoint in electronic music is the Boards of Canada track "Roygbiv." The bass line is so intricate that, on first hearing, a listener might assume that it is really the track's melody. Only after the bass line has been repeated several times does the "real" melody finally enter. Here is a composite of the two parts:



And here is the same composite in musical notation:



There are a few specific aspects to this counterpoint that make it so effective and that can serve as good guidelines for your own work:

Make sure that each part's rhythm is independent of the other parts. When parts move from note to note at the same time, our ears tend to hear the result as chords. The independence of the horizontal lines is lost, and we hear the vertical result instead. Successful counterpoint does result in good chords, but this is secondary to the melodic aspect. Notice how this is applied in the Boards of Canada example. With the exception of certain downbeats, each voice only moves to a new note when the other voice is sustaining.

Use contrary motion. As a general guideline, counterpoint works best when the voices mostly move in opposite directions; one voice moves up while the other moves down. This is known as contrary motion.

Be careful when crossing voices. In this example, each part stays within its own range of pitches, never crossing into the range of the other. This is straightforward here because the counterpoint is between a high melody and a low bass line. But in some cases, you might write two parts that could potentially cross each other. Brief voice crossings are generally not problematic, but if the parts are consistently winding around each other, they can lose their sense of independence. If you do want the voices to cross while still retaining their identity, try making the timbres of the parts highly distinct.

Not all music benefits from these techniques. In some cases, one melody —or even none—is enough. But if you're looking for ways to generate more complexity in your music, counterpoint might be a nice option.

Linear Rhythm in Melodies

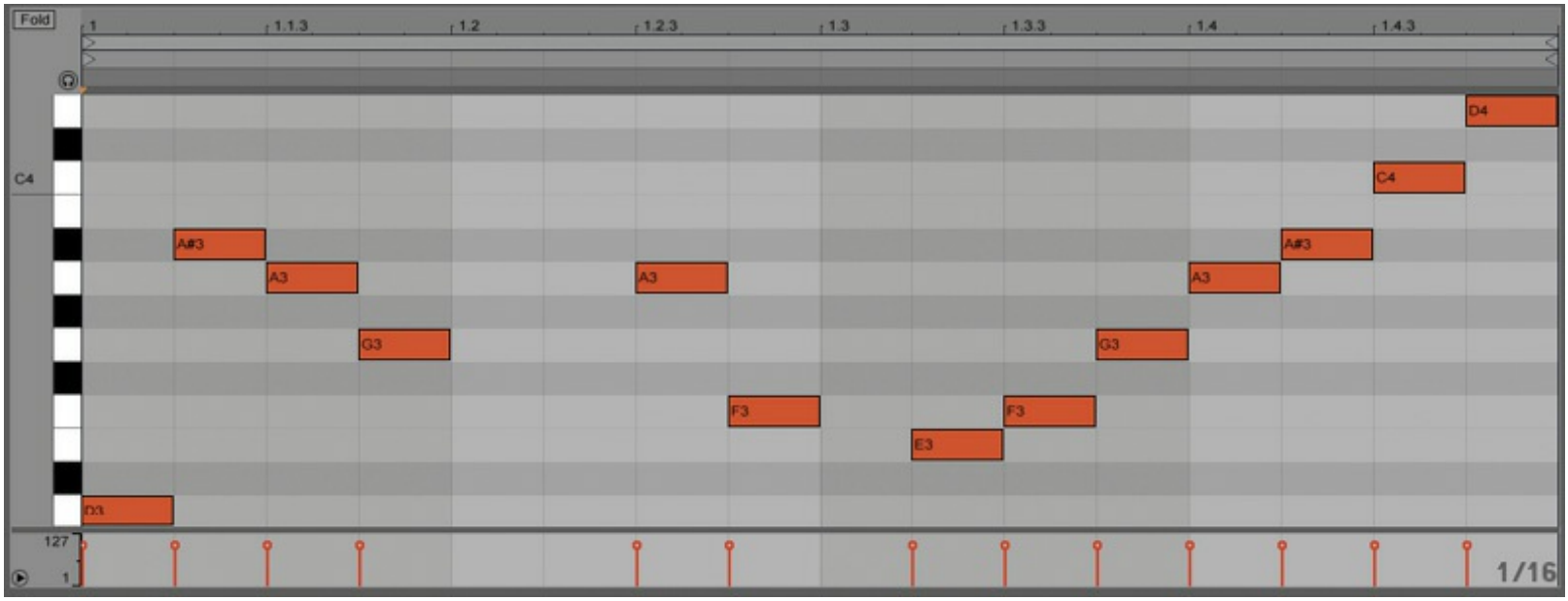
Problem:

You can come up with the beginnings of a good melody, but you're looking for ways to generate more variations and complexity. In general, your melodic ideas don't have the depth or richness of those you admire.

Melodies are hard. Almost by definition, they are the most important and memorable aspect of a good piece of music. But coming up with one that's worth remembering feels almost as challenging as writing the entire rest of the song. Here is a technique you can try for generating more interest in your melodic writing.

Solution:

Hocket is a compositional technique in which the notes of a melodic line quickly alternate between two or more different instruments, such that no two instruments are playing at the same time. It is the melodic equivalent of linear drumming (see Programming Beats 2: Linear Drumming) and can potentially turn bland melodies into interesting counterpoint. Hocket usually works best at fast tempos and when played by instruments with fast attacks and releases. It's a difficult technique for humans to play accurately but works really well for electronic music. As an example, let's begin with the simple loop below:



An easy recipe for turning this single voice into a hocketing pattern is as follows:

1. Duplicate the track, so that you have two identical copies of the pattern playing two separate (but so far, identical) instruments.
2. Modify the instrument in the new track so that it plays a different sound. This can be done by modifying or replacing the preset or even by replacing the instrument with a different one. As mentioned previously, try to find two instruments with similar volume contours. Short sounds with fast attacks work best.
3. Edit both patterns so that the melody is divided between them, such that no note is ever played by both instruments simultaneously. The way in which you divide the notes is entirely up to you, but really interesting hockets tend to alternate very quickly between the voices. The simplest example is probably literal alternation—the instruments trade off every other note. Here's how our original pattern might look if it was divided in this way:



You could also alternate by larger note groupings, such as every quarter note:



Or you could alternate by arbitrary time groupings, such as every third sixteenth (including rests):



Notice that in the last example, the pattern doesn't divide evenly into groups of three sixteenths, so the final alternation is cut short in the bottom voice.

These examples illustrate some possibilities with a two-voice hocket, but even more possibilities become available when you use three or more voices. In an extreme example, each note could be played by a separate instrument. Hocket can give even the simplest melody a whole new life by subjecting it to a rapidly changing cascade of new timbres.

Sound-Color Melody

Problem:

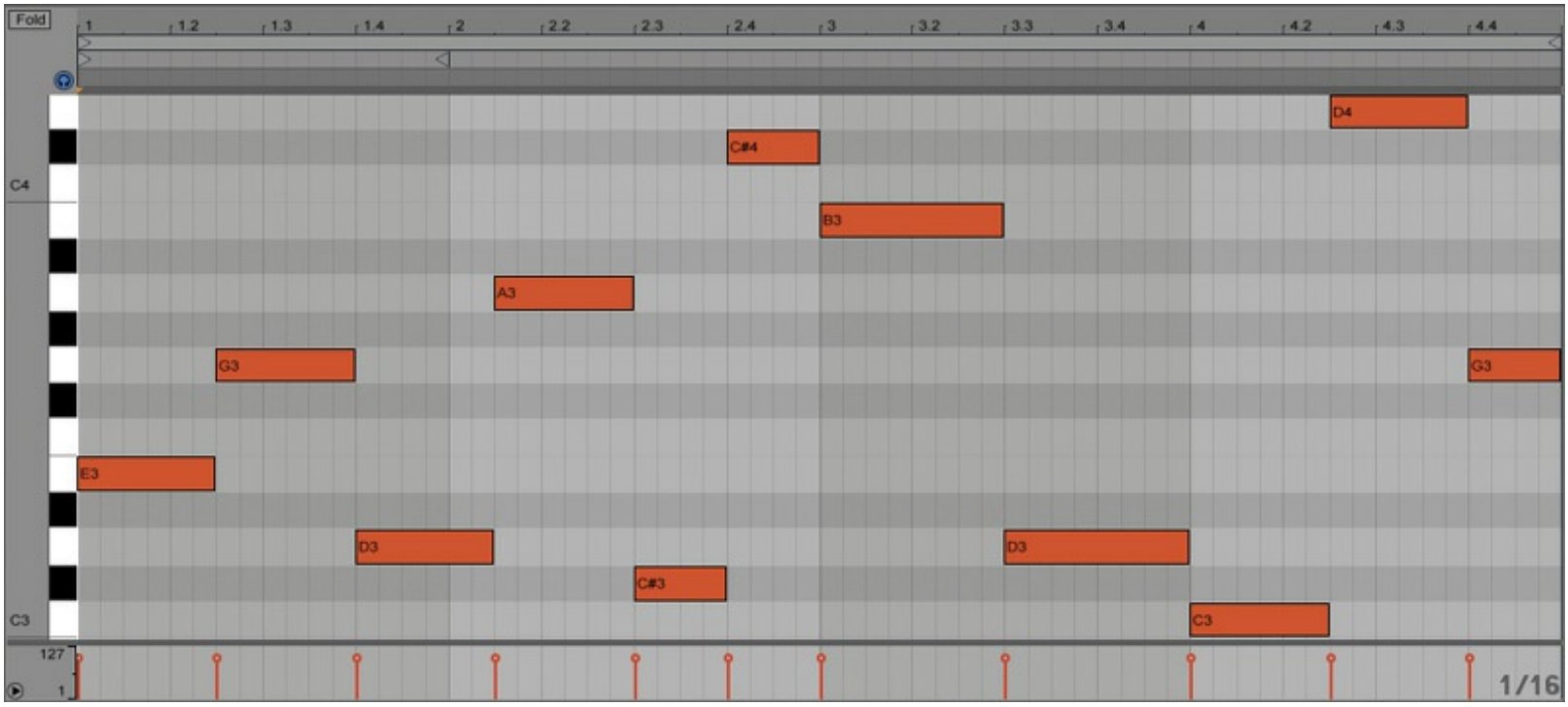
You can write a strong, memorable melody. But especially at slow tempos, even if the notes and rhythms are right, you regularly get bored with the sound. You've tried programming parameter automation so that the timbre changes over time, but you still have the sense that the sound is too static.

Long, slow melodies are difficult to write effectively in electronic music. By default, synthesizer sounds don't tend to vary much from note to note, and even with careful use of modulation or automation to create a dynamic, changing sound, there can still be a sense that the timbre of the melody isn't "breathing" or is too artificial sounding. Here is a technique you can use to create more timbral variation, particularly in slow melodies.

Solution:

The German word Klangfarbenmelodie ("sound-color melody") refers to the distribution of a melody (or even a single pitch) between multiple instruments. This is a similar concept to hocket (see Linear Rhythm in Melodies), although hocket generally refers to rapid, rhythmic transfer of melodic notes between voices at a fast tempo, while Klangfarbenmelodie is often slower and more subtle—one melodic voice gradually fading out as another fades in, for example.

Klangfarbenmelodie was originally developed as a technique for creating timbral variety in instrumental melodies, although it's perhaps even better suited for electronic music. For example, let's begin with the four-bar melody below:



The conventional way to treat a melody like this would be to have it play on a track that triggered a software or hardware instrument. For a richer sound, perhaps multiple instruments could be layered or stacked. But another approach might be to split the higher-pitched notes from the lower-pitched ones, and route the two halves through different instruments. This particular melody is well-suited to this kind of treatment because of its “wedge-like” distribution of pitches. For example, we might arbitrarily decide that the high voice will play E3 and above, while the low voice plays everything below E3. Many DAWs allow for creating “split points” or “zones” within a single MIDI track, which take care of the MIDI routing automatically. In these cases, you can simply assign the split points as desired, load the two instruments onto the same track, and play the clip. If your DAW doesn’t support this functionality, you could duplicate the track, modify or replace the second track’s instrument, and then delete notes as necessary from each pattern so that the melody is divided between them in an interesting and musical way.

Unlike hocketing, where abrupt, jagged timbral changes are the intended musical result, the idea with Klangfarbenmelodie is to create a smooth,

seamless exchange between voices. In the example melody discussed previously, another possibility might be to duplicate the track (or layer multiple instruments on a single track) but then use instruments that have “mirrored” volume envelopes. For example, one instrument might have a fast attack and a gradual decay, while the second instrument might have a slow attack. In this situation, the first instrument would supply the initial onset of the note, while the second instrument would fade in as the first faded away. Obviously, this requires carefully tuning the respective envelopes so that they make sense in the context of both the song’s tempo and the duration of the notes in the pattern. But when configured properly, the result can be a slow timbral shift across the duration of each note.

These examples illustrate some possibilities with Klangfarbenmelodie across two voices, but even more possibilities become available when you use three or more voices.

The Rhythm of Lyrics

Problem:

Your vocal melodies feel wrong somehow, as if there's a "conflict" between the rhythm of the melody and the natural flow of the lyrics.

Writing vocal parts is a fundamentally more complicated problem than writing instrumental music because words and sentences have their own implicit rhythm that needs to be consciously considered. Here are some tips for understanding how to get your musical and textual rhythms to work together (or to counter each other, if that's your intention).

Solution:

Scansion is the study of a text's inherent rhythm and the marking of each syllable in the text as being either strongly or weakly stressed. We use these stress patterns naturally and unconsciously when speaking and, unless we're poets, probably never need to give them much thought outside of a musical context.

As an example: Let's pretend this sentence is a lyric in your song. (Note: Artistically, this is probably not a good idea.) When you read it out loud, you will naturally apply a stress or "accent" to some syllables, while others will be spoken more softly. We'll mark the strong syllables with a / and the weak ones with a *. (By convention, these stress markings are placed over the first vowel in the syllable.):

```
/     * /     *   /  *    /  *  / *  /   *    /
Let's pretend this sentence is a lyric in your song.
```

Applying these markings can sometimes be subjective. For example, it

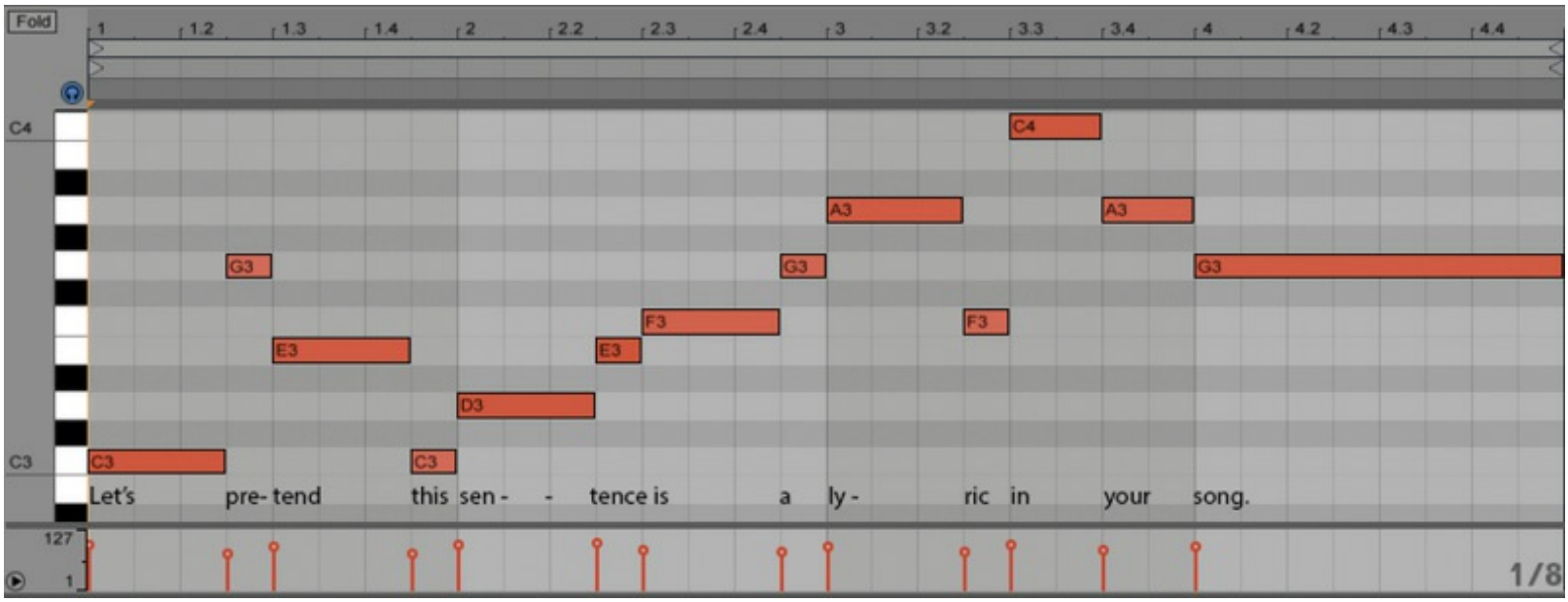
could be argued that “is” is weakly stressed (although it is probably stressed slightly more than the “a” that follows it). But now that we have a general sense of the inherent stress patterns of the text, we can relate this directly to the inherent stress patterns of musical beats. A couple of general (but by no means universal) rules:

> (Assuming a 4/4 meter), the first and third beats are considered “strong” or “on” beats while the second and fourth are considered “weak” or “off” beats.

> As you subdivide beats into shorter note values such as eighths and sixteenths, the same rule applies. That is, the odd-numbered subdivisions are perceived as stronger than the even-numbered ones.

Placing strong syllables on strong beats and weak syllables on weak beats usually results in a fairly natural-sounding vocal melody, while doing the opposite can often sound forced or stilted. Additionally, you can reinforce these natural alignments further by making stressed syllables louder, higher in pitch, or longer in duration.

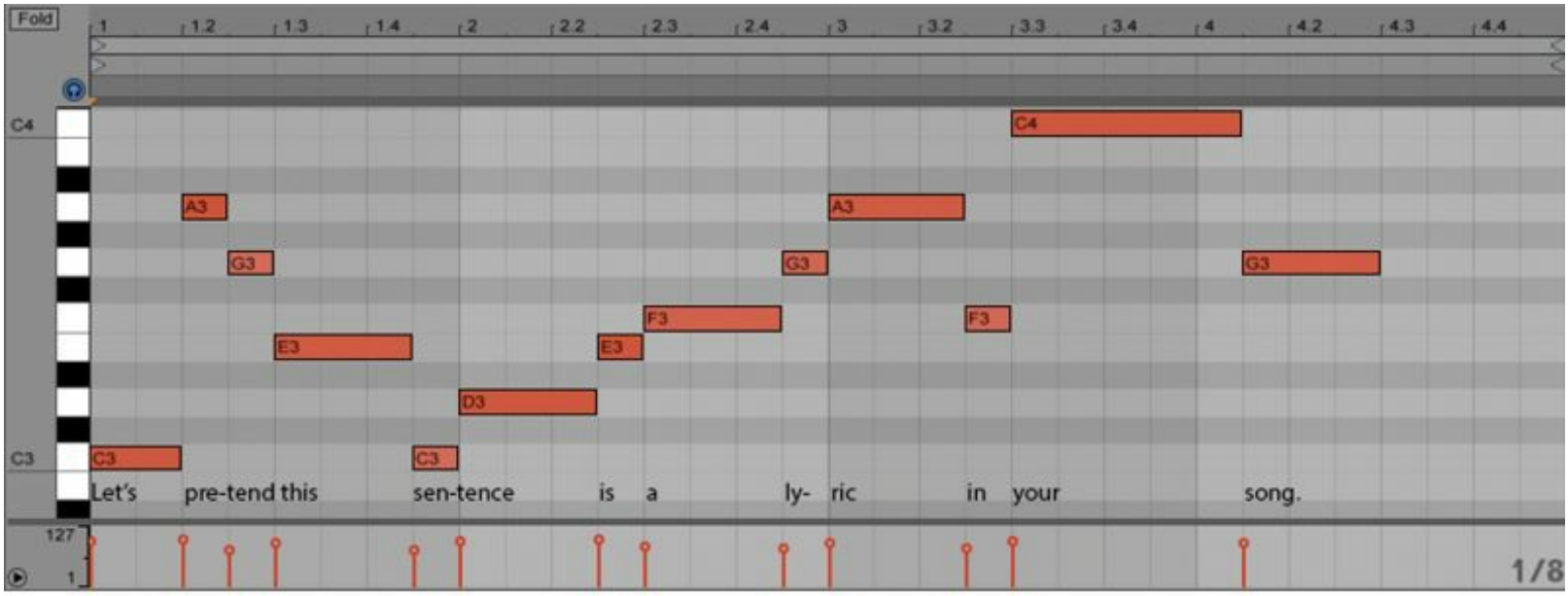
Here’s an example of a possible melody for this lyric:



It’s boring, but it “works” according to the scansion rules discussed earlier. Natural text accents occur on strong beat positions and are further emphasized by occurring on notes that last longer, while weakly

stressed syllables occur on weak beats and have short durations.

To contrast, let's take a look at a possible melody that goes against the natural stresses of the lyric:



This is nearly identical to the first melody, but with an additional note added on beat two of the first bar in order to shift each syllable to the left. Now we're breaking all the rules. The weakly stressed syllables happen almost entirely on strong beats and are sustained for long note durations, while the strongly accented syllables are short and on weak beats. This would be both difficult to sing and uncomfortable to listen to.

Although good vocal melodies are often significantly more nuanced and rhythmically intricate than both of these examples, you'll find that most of the time, good melodies follow the natural patterns of stress in the lyrics. Of course, there are exceptions. A lot of contemporary hip-hop, for example, uses syncopation and unusual stress patterns to go against the rhythm of the text. And it's always possible to apply more than one syllable of text to a single sustained note or, conversely, sustain a single syllable of text across more than a single pitch. But these are conscious and genre-specific musical decisions. If you find that your own vocal

writing tends to feel strained and unnatural, it might be because you're fighting against how the lyrics want to flow.

Creating Harmony 1: The Basics

Problem:

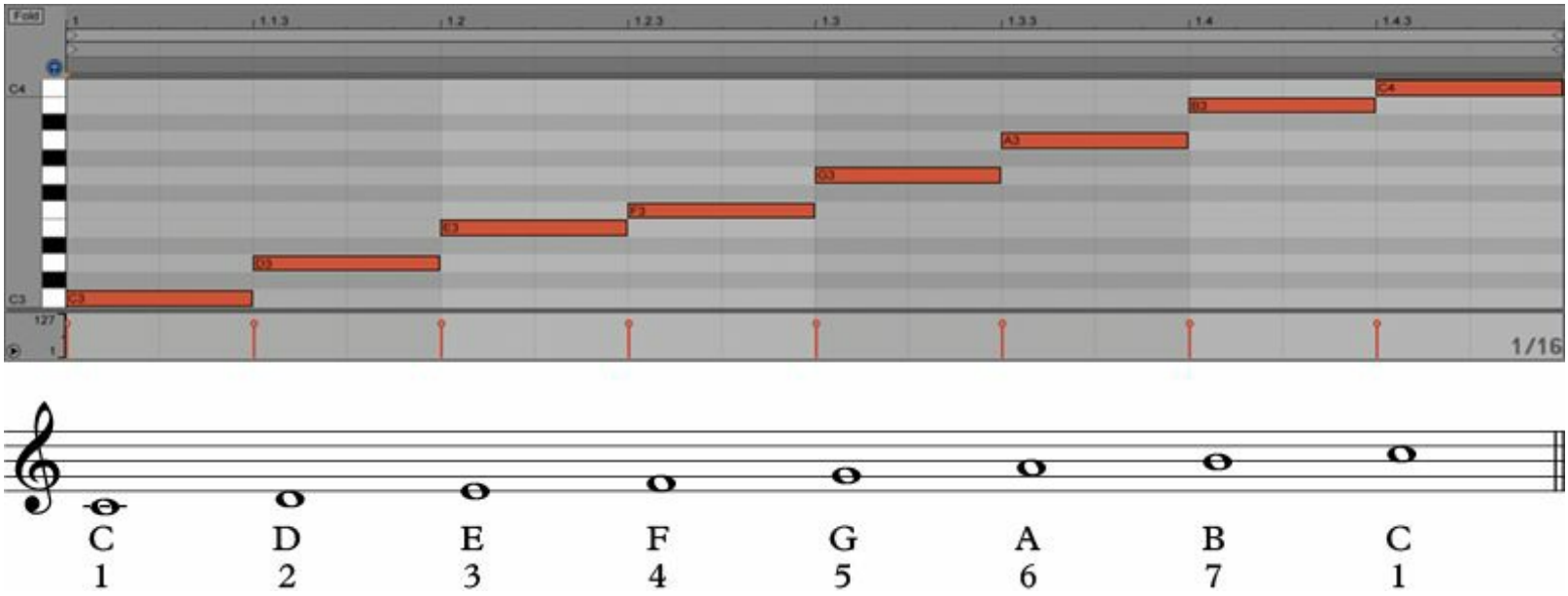
You don't know enough about how chords really "work" to understand how to get from one chord to another, and your attempts to create interesting chord progressions never match up to the music that inspires you.

Harmony is a big topic, and covering the full range of how chords work is beyond the scope of this book. But fortunately, the vast majority of what's commonly done with harmony in many types of music can be learned with just a bit of background and some basic terms. Here are some of the fundamentals.

Solution:

Most music is considered to be in a particular key. If we say that a song is "in C," this simply means that the note C sounds to most listeners like the most stable "home note" for the song. Key can be established in a few ways: either through Repetition and Insistence or, more commonly, through the use of chord combinations that sound, to most listeners, like they want to resolve back to the home note. This sense of tension and resolution is the basis of most successful chord progressions.

If we continue to assume C is our key note (or tonic), we can then determine the other notes in the scale of the key. The two most common qualities of scales are major and minor. These terms refer to the particular pattern of note distances (or intervals) between the adjacent notes in the scale. The C Major scale contains the following notes (shown in both a MIDI piano roll and conventional notation):

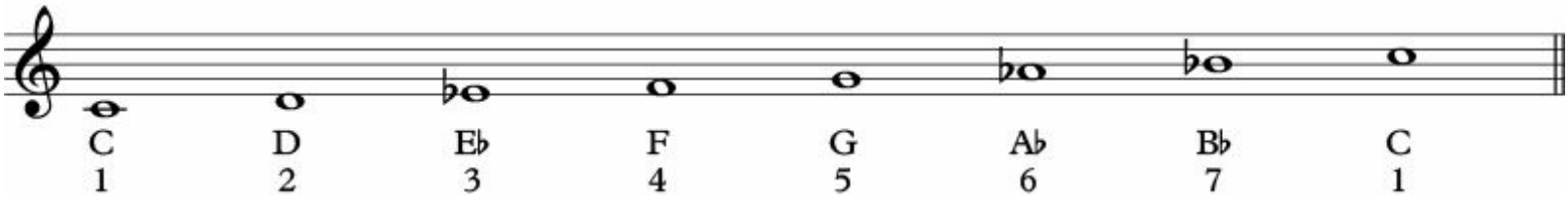


Although the examples in this chapter all use C as the tonic, you can easily transpose the examples in your DAW to try them out in any other key. Every major scale, regardless of starting note, contains seven notes (before repeating the first note again) and the same pattern of intervals between notes. That pattern is a particular combination of half steps (two immediately adjacent notes on the piano keyboard) and whole steps (notes that are two half steps apart): **Whole - Whole - Half - Whole - Whole - Whole - Half**:



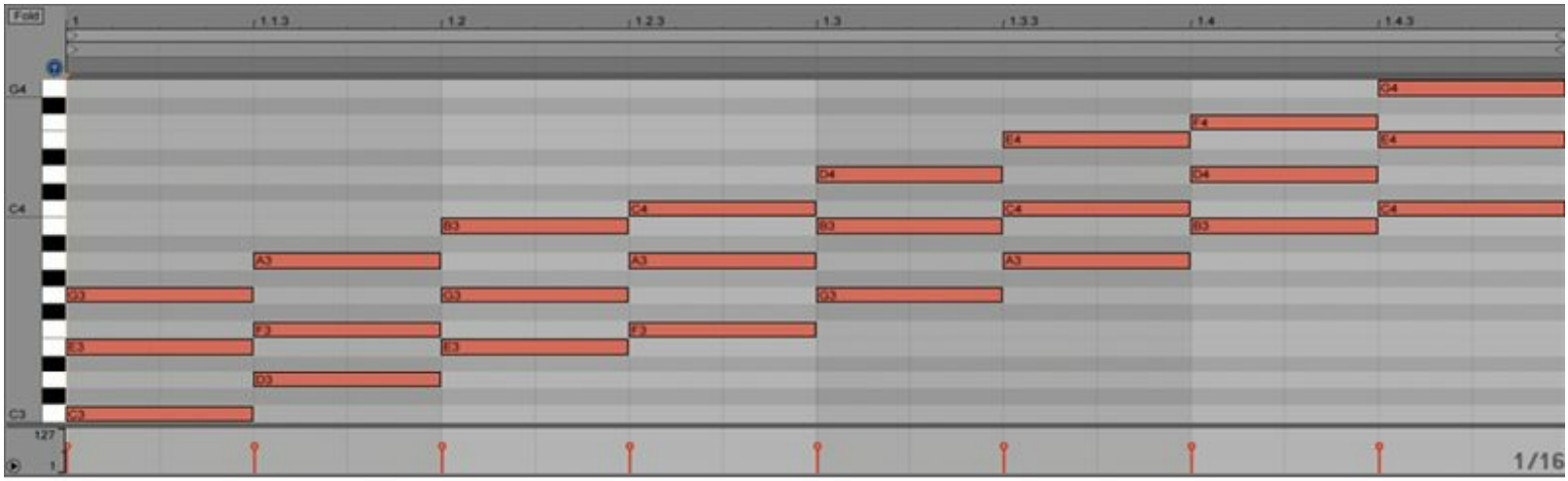
The minor scale contains a slightly different pattern of half and whole steps: **Whole - Half - Whole - Whole - Half - Whole - Whole**:

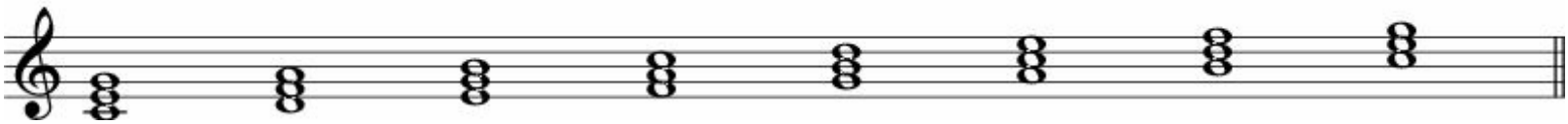




~ Note: Most DAWs use either only sharps or only flats in the piano roll. For our purposes here, consider D# = Eb, G# = Ab, and A# = Bb.

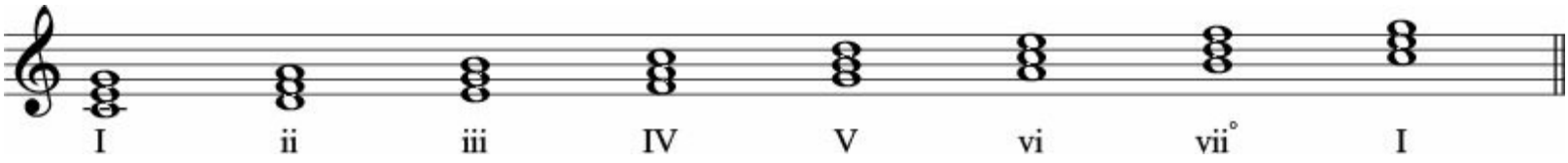
Now that we've established how major and minor scales are made, we can build chords on each of the notes in the scale. The most common chords are triads, which are built by adding the third and fifth notes above a starting note (or root). For example, in C Major, the triad built on C contains C (the root), E (the third note above C), and G (the fifth note above C). The triad built on A contains A (the root), C (the third note above A) and E (the fifth note above A). Here are the seven triads in C Major:



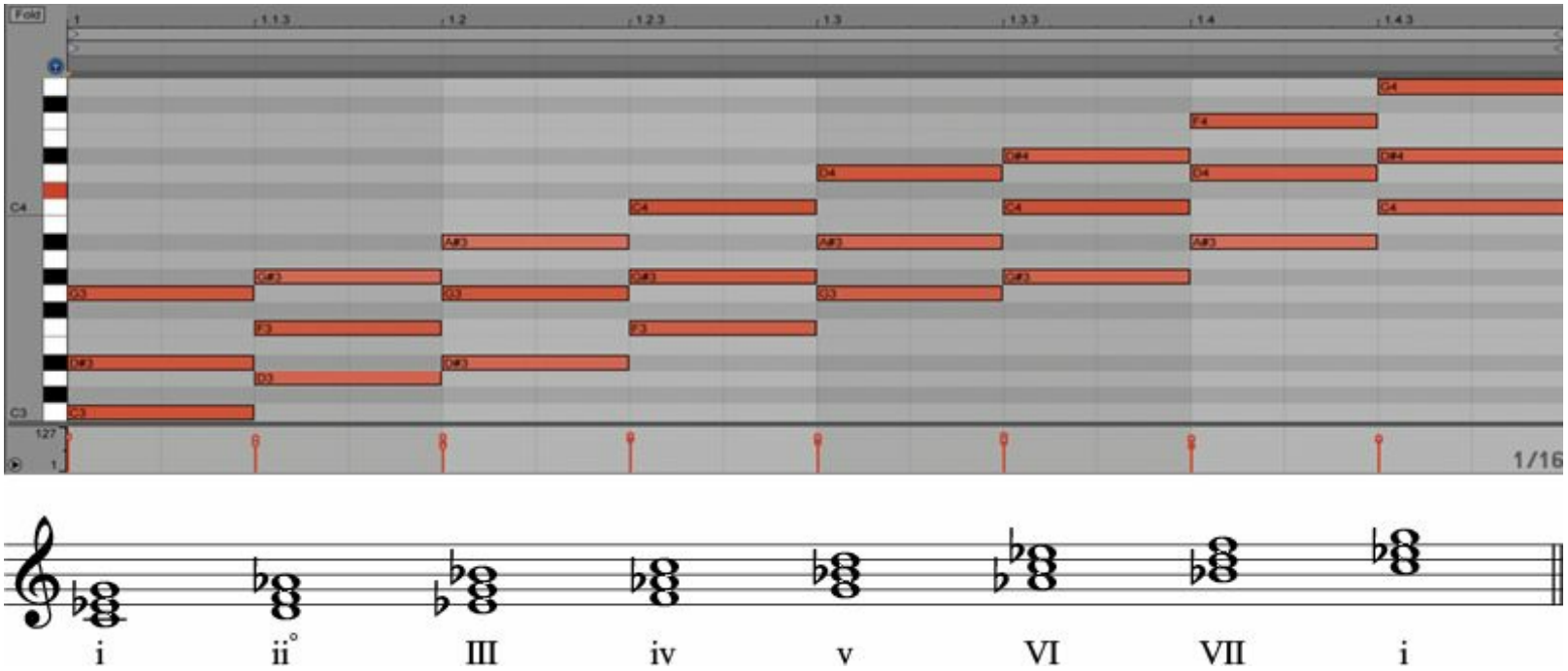


We referred earlier to the terms major and minor as being the quality of the key or scale. The triads in a particular key also have major or minor quality and, as with scales, this is determined by the interval distances between the notes. Major triads have four half steps between the root and the third note, while minor triads have only three. In both major and minor triads, there are seven half steps between the root and the fifth note. With this information, we can analyze and determine the quality of each triad in a particular key, simply by counting semitones.

Rather than using specific chord names (such as “C Major”), we can use Roman numerals for this type of chord analysis. This helps to emphasize an important aspect of this theory: Everything discussed here is completely transposable to any key, and all of the harmonic and scale relationships work exactly the same way regardless of which specific note is our tonic. By convention, major triads are labeled with capitalized Roman numerals, while minor triads are written with lower case Roman numerals. Here are the labels for the triads in the C Major scale:



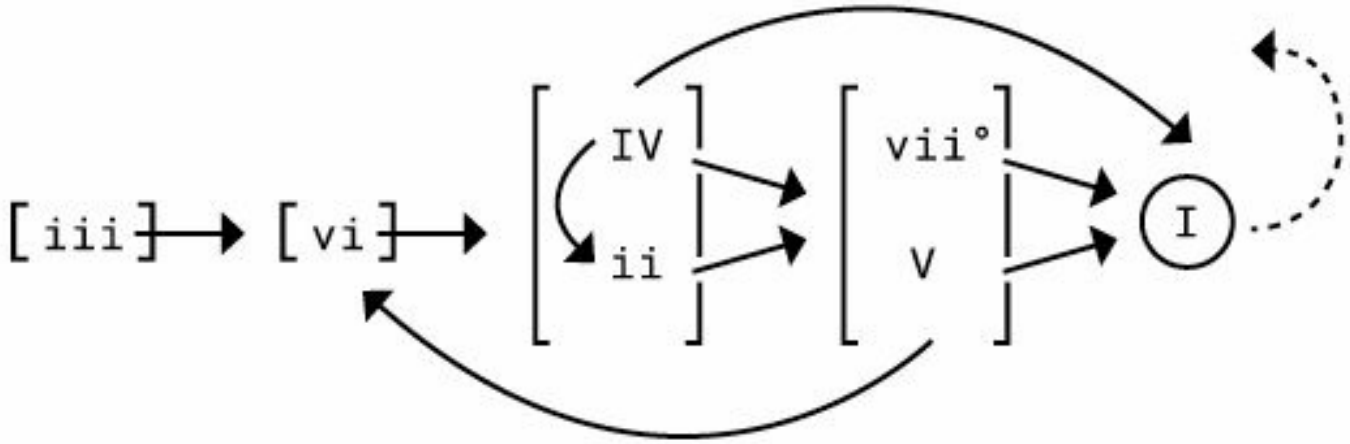
And the triads in the C minor scale:



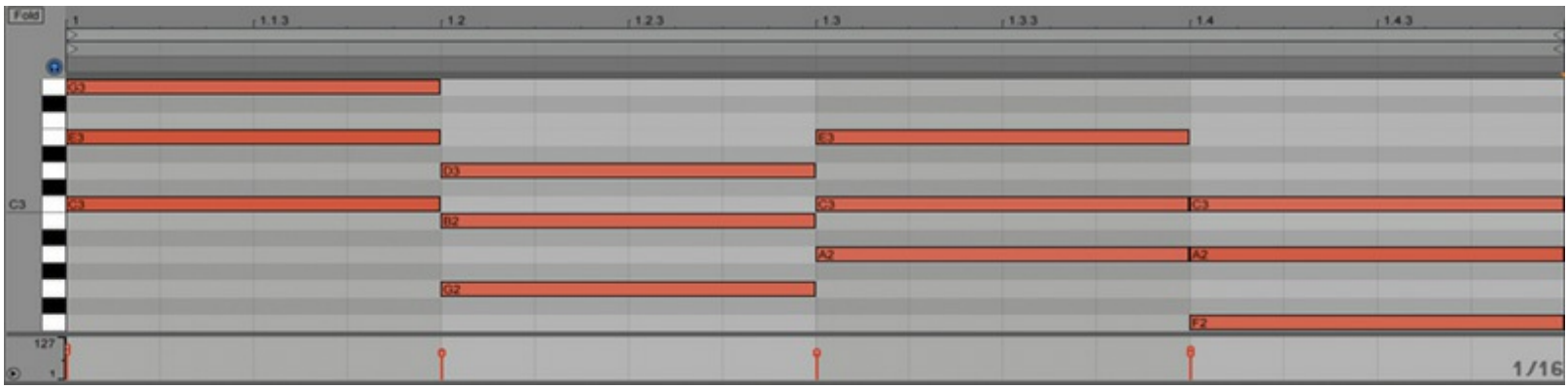
If you actually went through and measured the intervals in each of the triads above, you probably noticed that two of these chords don't fit into either the major or minor pattern. The seven chord in the major scale and the two chord in the minor scale are of a different quality, called diminished. Diminished triads, like minor triads, have three half steps between the root and the third note. But they have only six half steps between the root and the fifth note (as opposed to seven for the minor triad). Diminished triads are labeled with the superscript° after the Roman numeral.

It may sound surprising, but a huge amount of music, from the Classical period all the way to much contemporary pop and electronic music, uses only the seven chords that exist in a particular key (called the diatonic chords). And many songs use no more than two or three of these chords. But the choice and order of diatonic chords is what determines how the chord progression really sounds. What follows are a few general guidelines that can help you move from one chord to the next (Roman numerals refer to the major scale, but the patterns are similar for minor scales).

After a I chord, anything is possible; all chords within the key sound equally appropriate after the tonic chord. The V or viio chords sound (to most listeners) like they "want" to resolve back to I (although V can sometimes lead to vi). The ii and IV chords have a tendency to lead towards the V or vii^{o} (although IV also commonly leads back to I or to its "partner" ii). The vi chord leads to the ii or IV. And finally, the iii chord has a tendency to be followed by the vi. This series of chord resolutions can be summarized in this figure:



Using just these guidelines for diatonic triads, you have access to a huge number of usable chord progressions. For example, one of the most common progressions is I - V - vi - IV:



This is the progression in the verses of Imogen Heap's "Hide and Seek," the choruses of Adele's "Someone Like You," and many more. In fact, there are countless well-known songs that use this exact progression

("Four Chord Song" by Axis of Awesome is an enlightening mashup of 36 of them).

Chord progressions that follow these guidelines are said to be functional because each chord functions as a preparation for the next chord in the series. There is, of course, a much wider world of harmonic possibilities available beyond just using diatonic triads. But if you're interested in creating music that's within the realm of pop music, you can get quite far using only these seven chords.

~ Side note: A lot of the theory behind the construction of scales and triads may seem like arbitrary, rote memorization. Why did these particular combinations of whole and half steps come to define the scales that are commonly used? Why do certain chords have a feeling of "wanting" to move to other chords? Why do these scales and chords exist at all? Detailed answers are beyond the scope of this book but largely have to do with acoustics: The intervals of the major scale can be derived from fairly simple whole number frequency ratios that correspond to the overtone series. But perhaps even more importantly, much of this theory is descriptive rather than prescriptive: The theory was developed to help explain patterns in music that already existed, rather than to tell composers what they should or should not do in the future.

Creating Harmony 2: Beyond Triads

Problem:

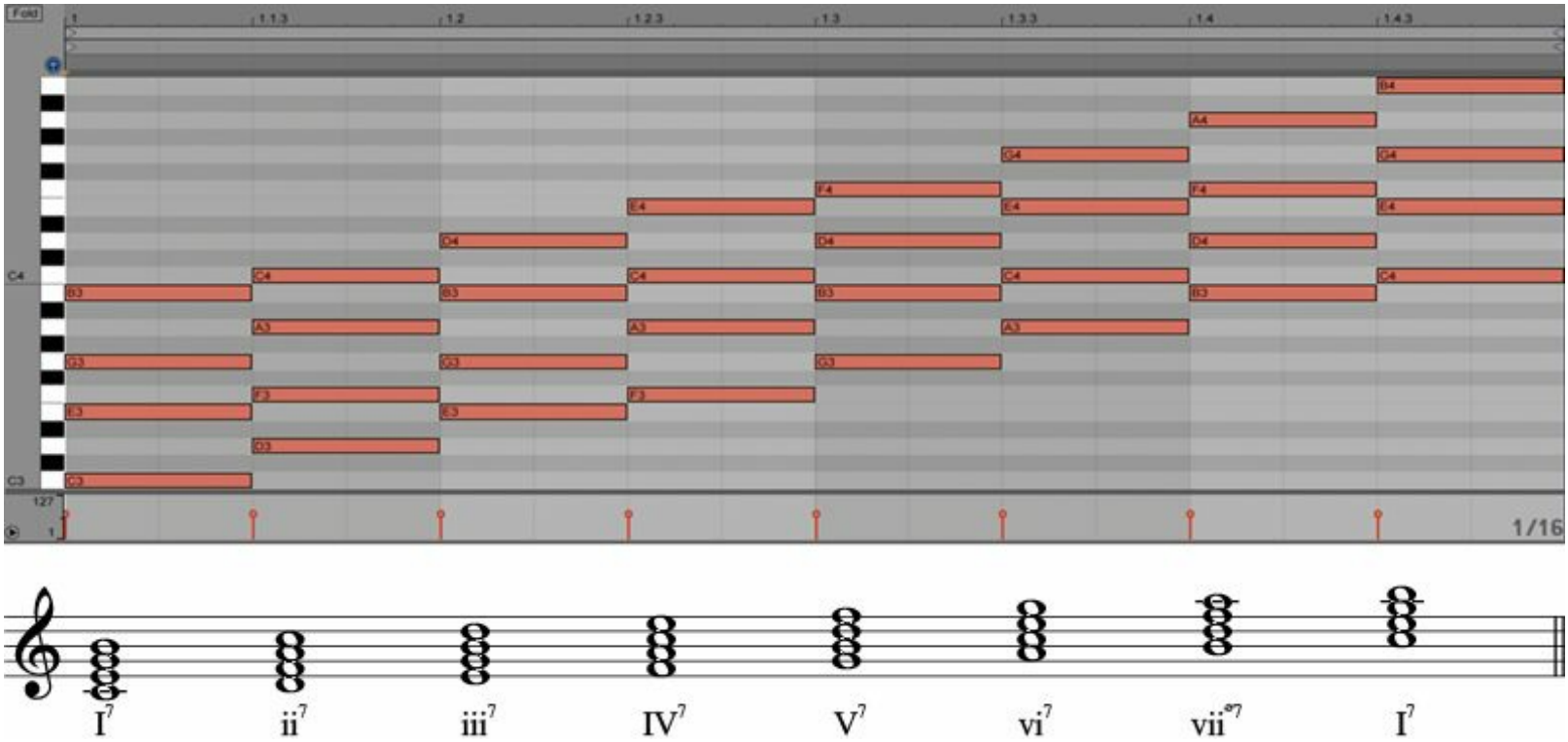
From reading the previous chapter, you understand the fundamentals of triads and diatonic harmony, as well as how to use these tools to create usable chord progressions. But these chords lack the complexity and "color" that you hear in genres like jazz and even some house music. What else is possible?

Solution:

There are many types of chords besides diatonic triads. But the basic tools that we use to build those triads can also be used to create a much wider range of more complex chords. Here are a number of these chords, as well as some ideas about how to use them:

Seventh Chords

We know that triads are built by choosing a starting note (or root) and then adding the third and fifth notes above it. But we can extend this pattern by adding the seventh note above the root. The four-note chord that results is (conveniently) called a seventh chord. Here are the diatonic seventh chords in C Major (shown in both a MIDI piano roll and conventional notation):



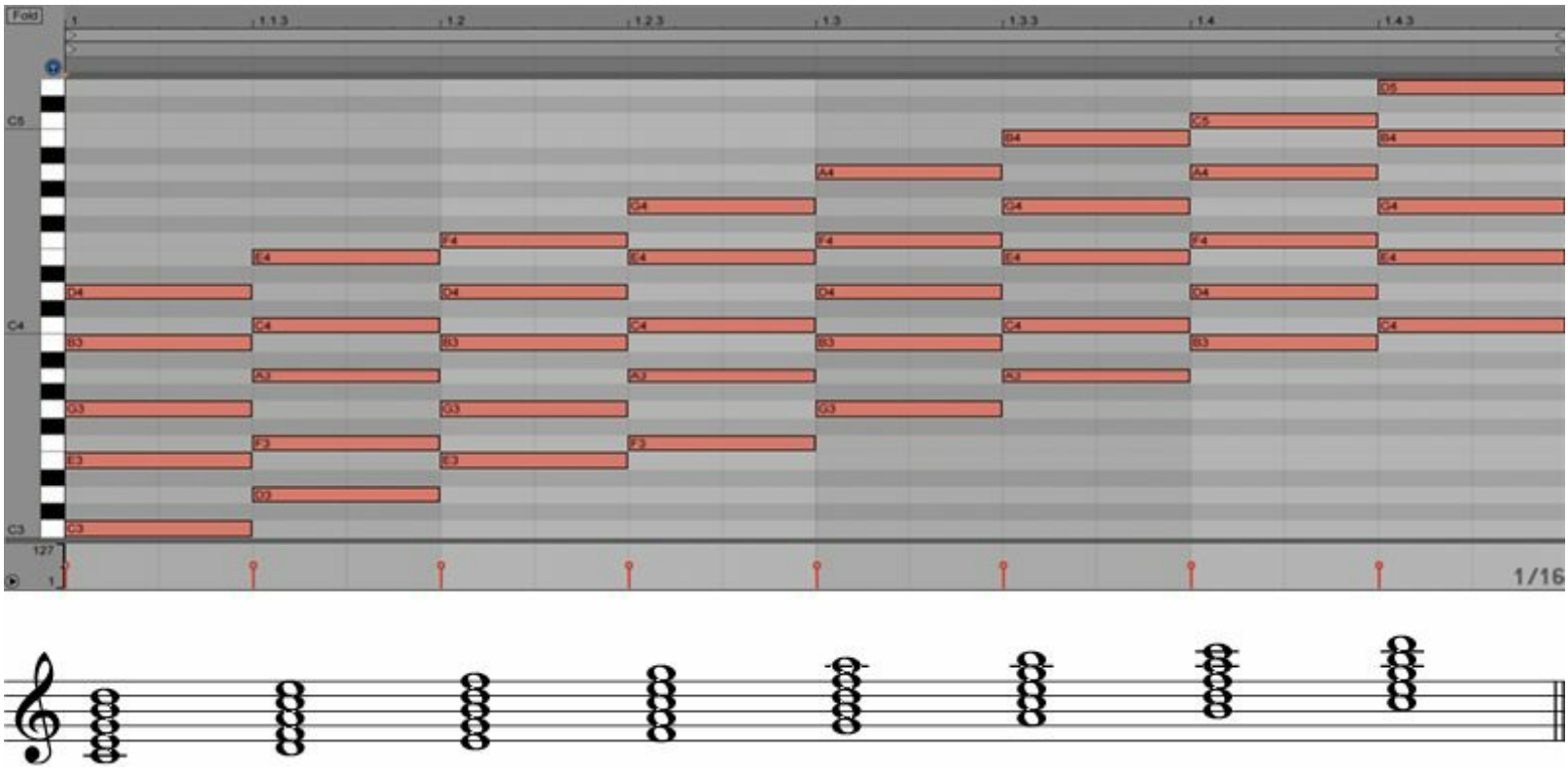
As with major, minor, and diminished triads, seventh chords also have a quality, although the situation becomes slightly more complex because the triad that serves as the basis of the chord does not necessarily have the same quality as the seventh itself. For example, the I and IV triads are both major, and so are the corresponding I7 and IV7 chords. This is because we define a major seventh as the interval distance of 11 half steps from the root to the seventh of the chord. But the V7 is very interesting. The V triad is major, but the seventh is minor—ten half steps between the root and the seventh. This combination of a major triad with a minor seventh on top is called a dominant seventh chord and has a strong feeling of "wanting" to resolve to the I chord. The viiø7 is another interesting case. Here, the triad is diminished while the seventh is minor. This is known as a half-diminished seventh chord.

With the exception of the V7, which is very commonly used to resolve back to I, seventh chords are used less frequently than triads in many genres. But they are the fundamental harmonic building blocks used in jazz and related genres. In fact, almost any "normal" functional chord progression (as explained in Creating Harmony 1: The Basics) can be

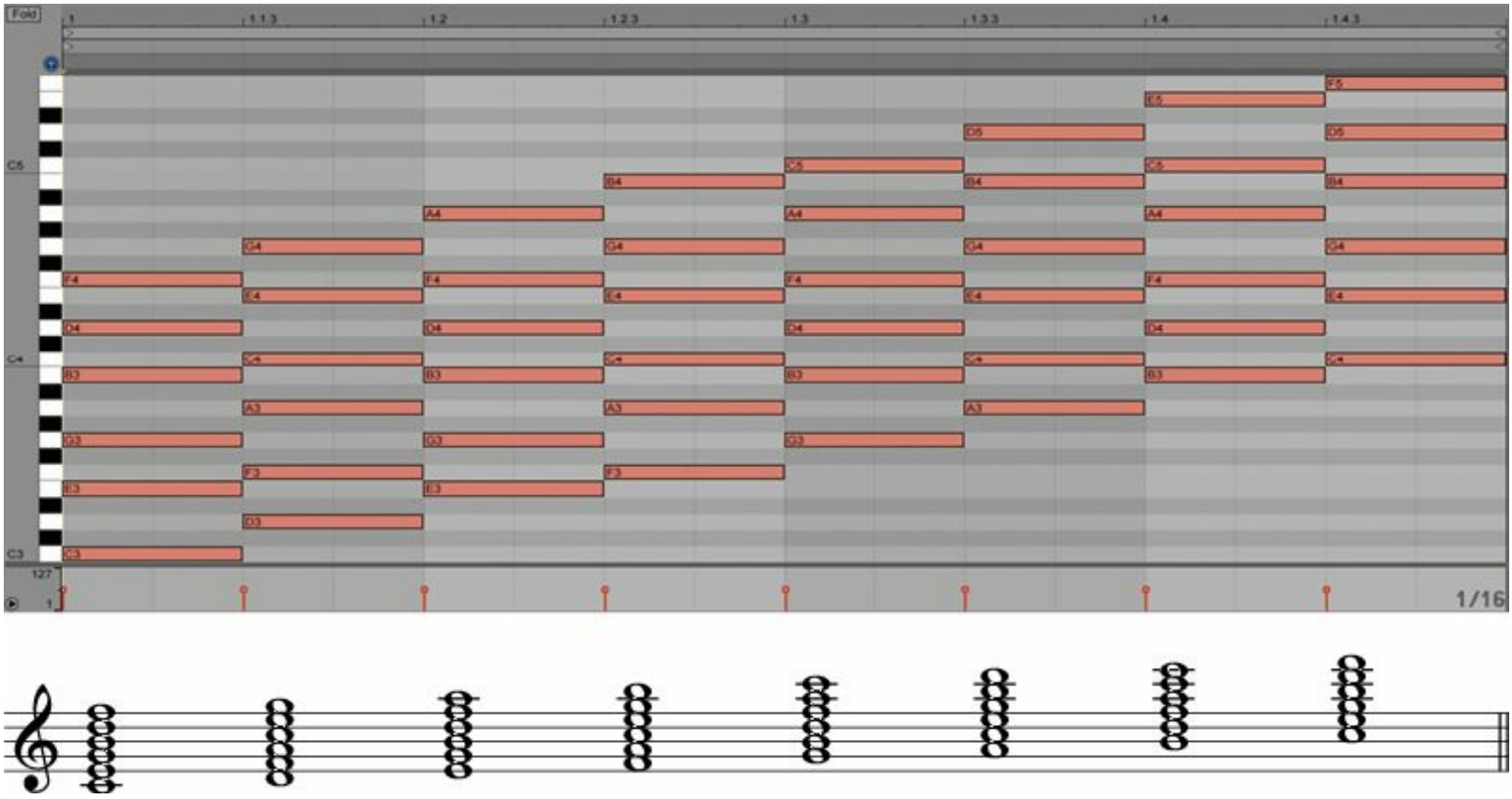
given a jazz flavor by replacing the triads with their corresponding diatonic seventh chords. Jazz often uses sevenths in “non-functional” ways as well, and jazz progressions often consist of chains of seventh chords that lead to one another primarily by smoothly connecting the individual voices within the chords and maintaining common tones whenever possible (rather than by use of predictable functionality). For more about how to make these kinds of progressions, see the chapter about Voice Leading and Inversions.

Further Extensions: 9ths, 11ths, and 13ths

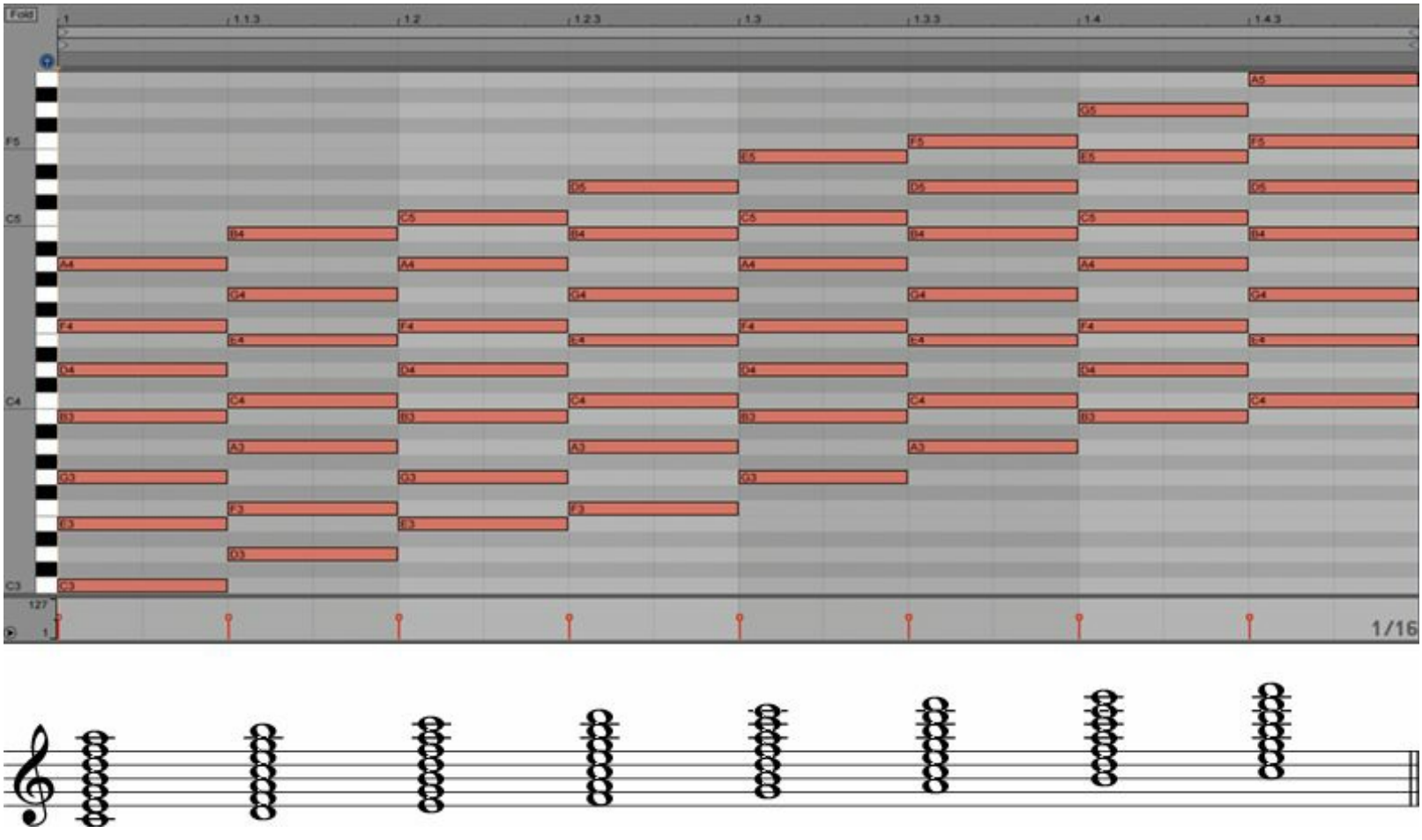
We can extend this method of constructing triads and seventh chords even further simply by adding additional thirds to the top. With one additional note, we get 9th chords:



Add one more note to get 11th chords:



And finally, yet one more note creates 13th chords:

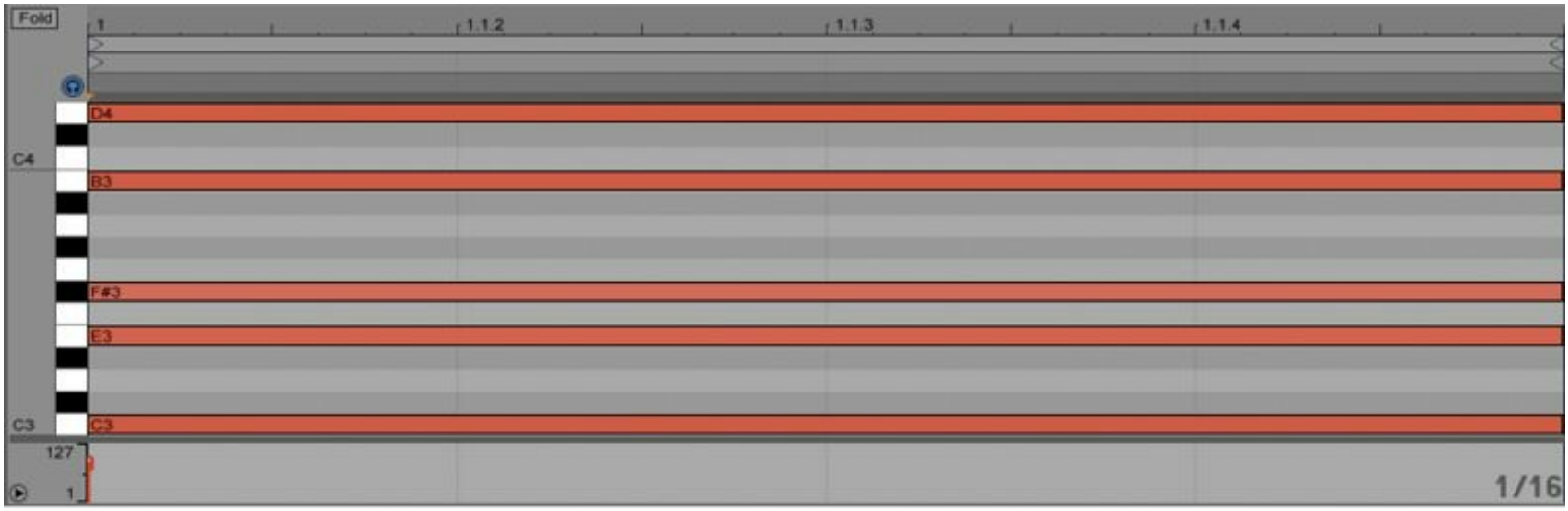


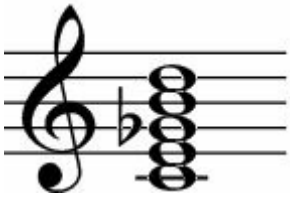
Note that I have left the Roman numeral analysis off of these chords.

This is because the quality of these chords quickly becomes very complex, and also very ambiguous—in many cases, there is no single "correct" analysis for these large chords. It's also because these types of harmonies are not normally used in a functional context (with the exception of sometimes serving as replacements for conventional progressions of triads).

It is rare for keyboard players to play every note of these very large, dense harmonies. This is partly because it's difficult to quickly navigate the keyboard while playing seven-note chords, but also because the sound of these harmonies can be implied by playing only the most "important" notes in the structure—commonly the highest notes in the extension, plus maybe the third. The root is commonly omitted entirely, usually because it's filled in by the bass or another instrument. Although it is no problem to program a DAW to play every voice of huge chords like this, you may still want to avoid doing so because you can quickly overpower a mix with such enormous blocks of harmony.

Although diatonic 9th, 11th, and 13th chords are common in jazz, it is also common to alter one or more of the notes in the chords to create even more interesting colors. Altering a chord simply means replacing one or more of the notes with a note that is one half step away. Some common alterations include lowering the fifth note, as in this 9th chord:





or raising the 11th.





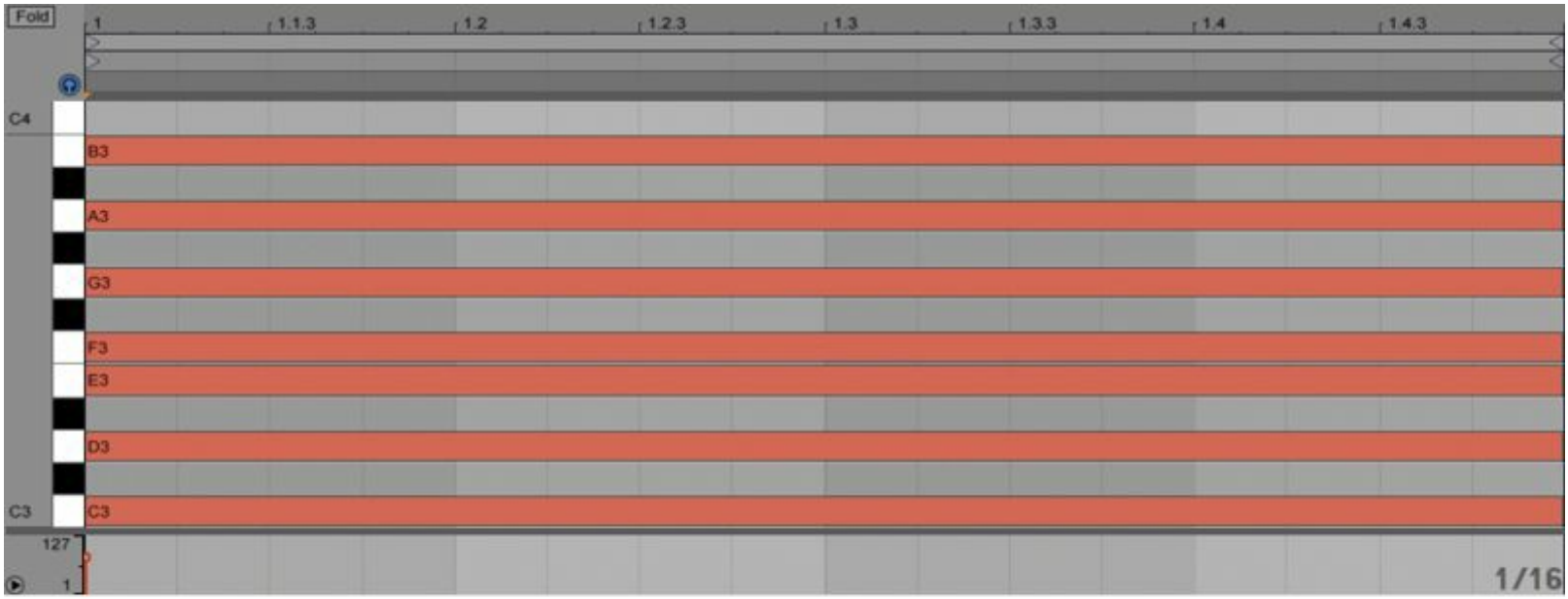
Notice that these chords are almost identical. In the first example, we've lowered the fifth of the chord, while in the second we've maintained the fifth while adding a raised 11th on top. But the lowered fifth (a Gb) and

the raised 11th (an F#) are the same pitch. This is an example of the kind of subtlety and ambiguity that gives these kinds of dense chords their sophisticated flavor.

You may have also noticed an interesting characteristic of the 13th chords—each of them contains every single note of the C Major scale, just starting in a different place. This leads to our next topic…

Non-Triadic Chords

Since 9th, 11th, and 13th chords all include intervals larger than an octave, we can create closely related harmonies by simply "folding" the upper extensions down so that they're compacted into a single octave. For example, compacting our 13th chord built on C results in this:





Although this contains exactly the same pitches as the original 13th, it is voiced using adjacent notes rather than every other note. Chords built on adjacent notes are called secundal chords or clusters.

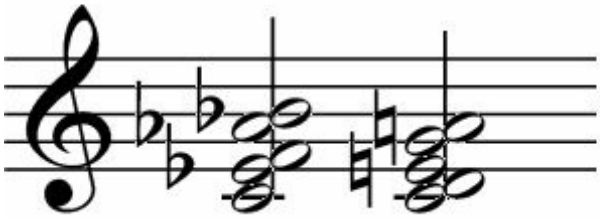
Now that we know we can build chords from thirds or from seconds, it follows that we can build chords from other intervals as well. Chords built by stacking notes that are five semitones apart are known as quartal chords, while stacking notes seven semitones apart yields quintal chords. Here are examples of five-note quartal and quintal chords built from C:





As with the extended chords discussed previously, these very “wide” chords can be compacted to fit within a single octave, often resulting (again) in secundal constructions. In fact, the two chords used in the previous example form a quite nice two-chord progression when condensed in this way:



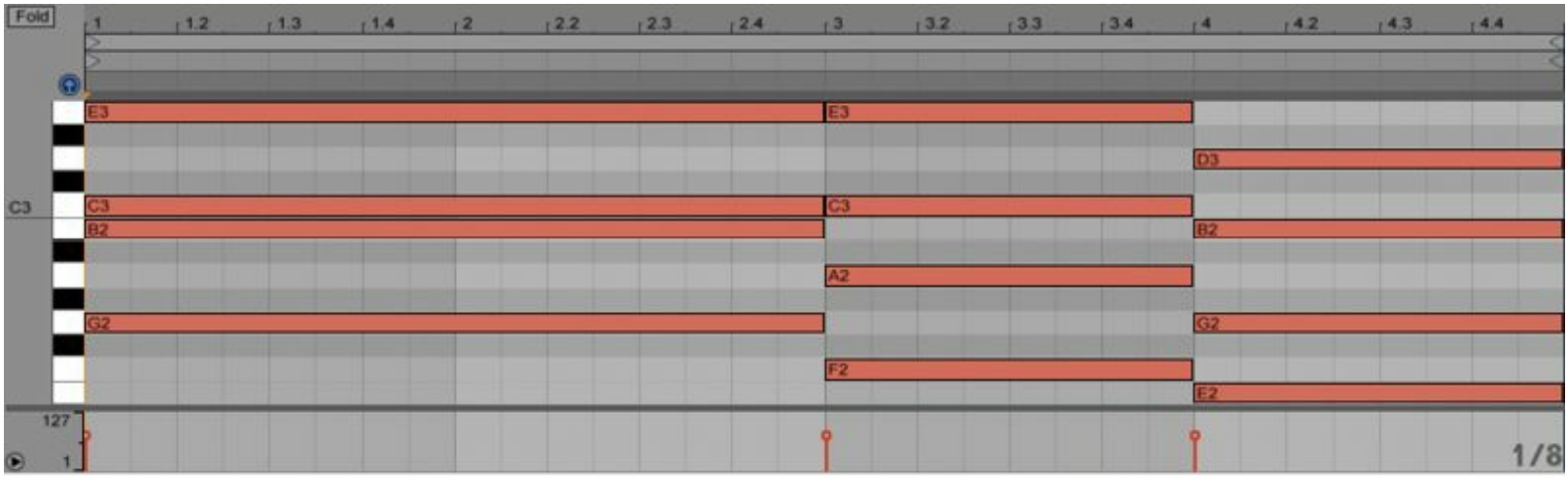


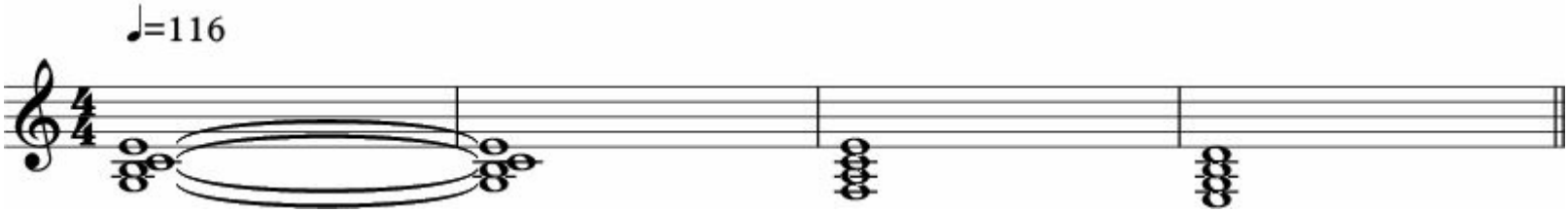
Note that these non-triadic harmonies usually aren't considered to have major or minor quality; in most contexts, they exist outside of the major/minor domain. Regardless, they are commonly used in jazz and modern classical music, again largely connected in progressions via voice leading rather than goal-oriented functionality.

This is just a sampling of the kinds of chords that are available when we expand our palette beyond conventional triads. But even using this expanded repertoire should allow you to come up with more colorful harmony than would be possible with triads alone.

A Real-World Example

The classic house track "Can You Feel It" by Mr. Fingers is a great example of some of these types of extended harmonies. The progression in the pad sound that occurs all through the track is only three chords, but they're exactly right. They also demonstrate how analysis of this type of harmony isn't always an exact science:





The song is in the key of A minor. So what are these chords?

1. **A minor 9:** At first glance, this labeling may seem strange. Why is this an A chord if it doesn't contain an A? The reason is the bass line, which consists of lots of repeated A and E notes underneath the harmony. Thus we have the bass line's root (A), plus the third (C), the fifth (E), seventh (G), and the ninth (B).
2. **F Major 7**: the root (F), the third, (A, both in the chord and in the bass line), the fifth (C), and the seventh (E).
3. **A minor 11** : the bass line's A, plus the fifth (E), the seventh (G), the ninth (B), and the 11th (D). Interestingly, the chord contains no third (C), which gives it an unusual, unstable color. And if we ignore the A in the bass, we could call this a simple E minor 7, which suggests a somewhat more functional progression: v - I. So is there a "right" analysis? Not really. The ambiguity is part of what makes the progression interesting.

Voice Leading and Inversions

Problem:

When you play chords on your keyboard, you find that the motion from one chord to the next sounds choppy or abrupt. Furthermore, you feel like you have to move your hands really far to get from one chord to another. But in the music that inspires you, the chords always sound like they move smoothly from one to another. What are your favorite artists doing that makes such a difference?

Creating smooth, flowing chord progressions often has less to do with what the actual chords are than it does with the nature of the voice leading from one chord to the next. Here's some information about how voice leading works and how to use it to create more satisfying harmony in your own music.

Solution:

Although we normally think of a chord as a vertical construction, we can think of the motion between chords as a series of horizontal constructions. That is, when one chord moves to another, the individual voices within those chords can be thought of as separate melodic lines. This relationship between the separate voices when moving between chords is known as voice leading.

Although voice leading is a big topic, we can use just a few basic voice leading principles to help create smoother, better-sounding chord progressions. In general, smooth voice leading results from minimizing the amount of motion from one note to the next within a single voice. This means moving the smallest possible number of semitones, as well as preserving common tones whenever possible. (Common tones are

notes that remain unchanged from one chord to the next.)

As an example, let's imagine a two-chord progression that moves from a C minor chord to an F minor chord. The simplest way to construct such a progression is probably something like this:



Both of these chords are in what's known as root position: The "roots" of the chords (C and F) are the lowest voices of the chords. For non-keyboard players root position chords are usually where the hands fall naturally. But progressions from one root position chord to another result in voice leading that's not ideal. In this example, no common tones are retained, and each voice needs to move a considerable distance (as shown by the black lines).

A smoother version of this progression might be something like this:



Here, the C minor chord is still in root position, but we've changed the order of the notes in the F minor chord to improve the voice leading. A reordering of a chord's notes is known as an inversion. In this case, we've inverted the F minor chord by placing the C (rather than the F) in the bass. This means that we maintain a common tone (the C) when moving between the chords while also decreasing the distance traveled by the upper voices.

We could achieve a similar effect by inverting the first chord while leaving the second one in root position:



Here, we've inverted the C minor chord by placing the Eb(rather than the C) in the bass. This means we maintain a common tone (the C) when moving between the chords, while again minimizing the distance that the other two voices travel.

By carefully using chord inversions, you can create smooth voice leading from one chord to the next, which is both easier to play and (generally) sounds better. But note that there are certain cases (as discussed in the chapter on Parallel Harmony), when you might not want to change the voicing of chords but instead move in parallel from one chord to another. The choice depends on genre and personal taste.

Parallel Harmony

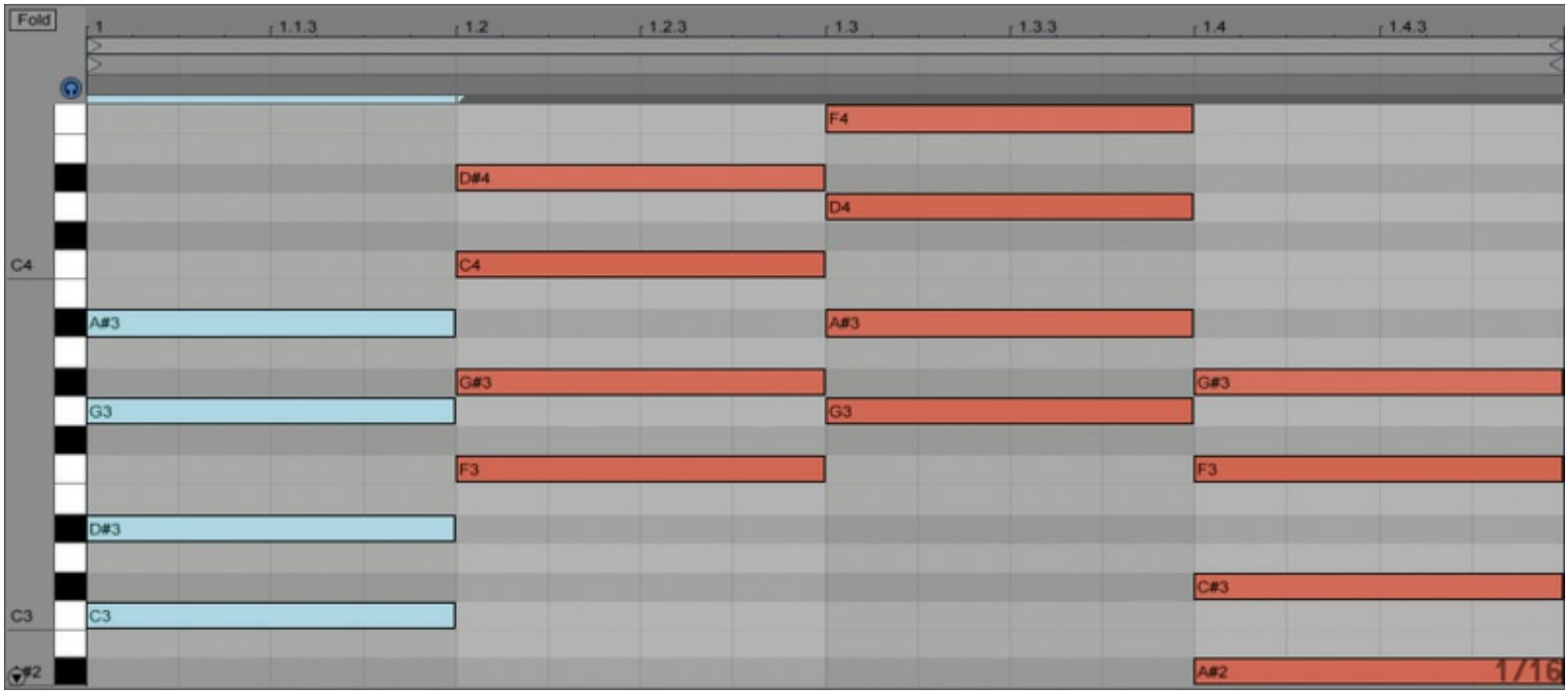
Problem:

There's a certain quality to the chord progressions in a lot of deep house, post-dubstep, and related styles that seems to be difficult to replicate in your own music.

Harmony in house music and its variants is often a tricky thing to dissect. The chord progressions are often quite repetitive but also seem strangely alien—they sound quite different from the way harmonic relationships work in any other music. Here is one way to unlock some of what's going on for use in your own music.

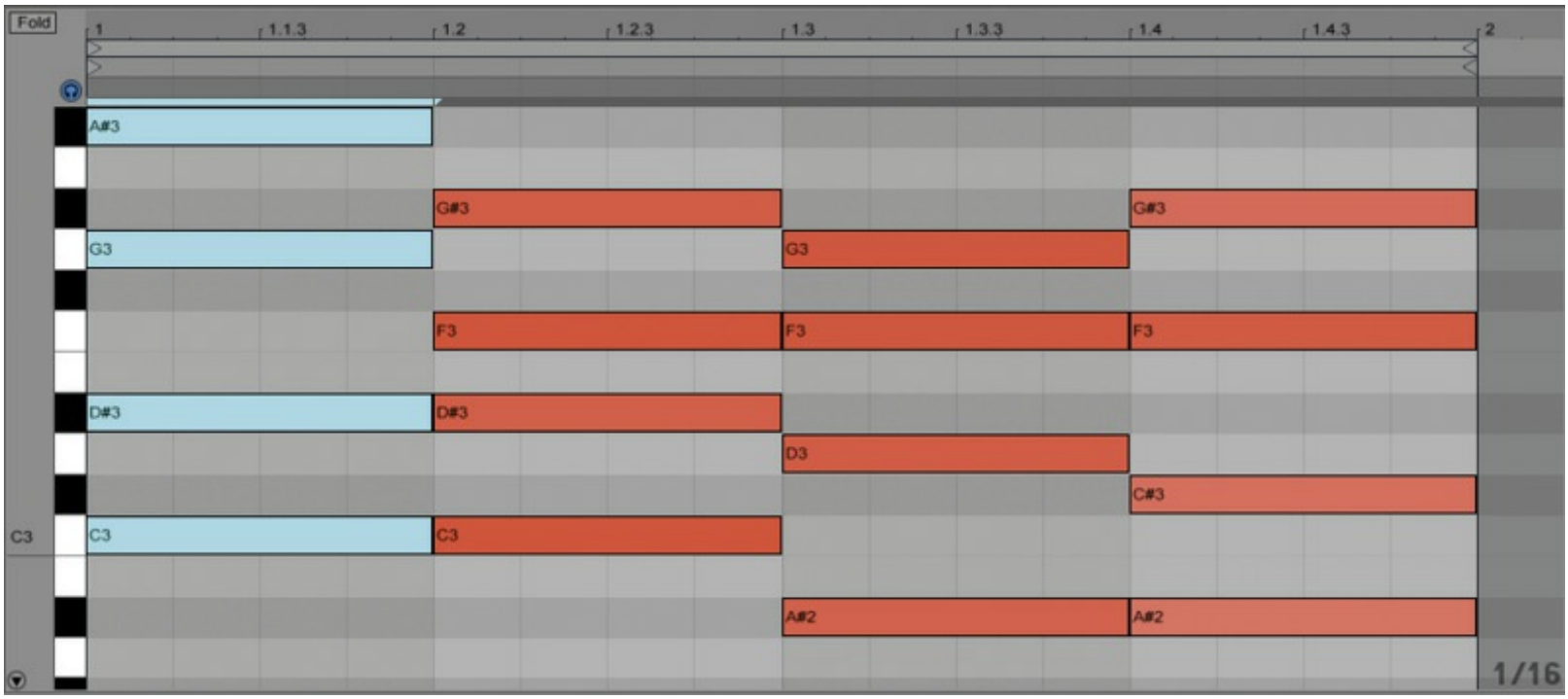
Solution:

A lot of house music chord progressions use a technique called parallel harmony, which refers to a method of moving from one chord to another in which each note moves by the same number of semitones and in the same direction. Parallel harmony is essentially what you would get if you copied a chord, pasted it somewhere else, and then transposed the copy. For example, here's a chord progression (Cm7-Fm7-Gm7-Bbm7) constructed using parallel harmony:



If you try to play this on a keyboard, you'll notice that you need to move your hand somewhat far to get from one chord to the next. A trained keyboard player might instead re-voice these chords to minimize hand movement, preserve notes that are shared between adjacent chords, and improve the Voice Leading.

Starting with the same voicing for the first chord, a more typical voicing of this progression might be something like:



But while traditional music theory would probably call the second version “better” than the first, the parallel version has a distinct sound which has found a place in a lot of electronic music. There are a number of technical and historical reasons for this, and you can use these techniques in your own music as well:

Sampled Chords

In house music, a common technique for creating harmony is to sample a single interesting chord from another song. As with sampled breakbeats, the source for these single chords is often classic soul or jazz tracks. By then playing the sampled chord at different pitches, the resulting harmony moves in parallel.

Chord Memory

A number of classic polyphonic synths from the 80s had a feature called “chord memory,” which was essentially an automatic parallel harmony generator. To use chord memory, the user would hold down a chord and press the chord memory button. After this, playing a single note on the keyboard would play the memorized chord, transposed as necessary. Many modern DAWs offer chord-generating MIDI processing tools that allow you to achieve a similar effect by automatically adding one or more notes (at specific intervals) to whichever note is played manually. This allows you to use any plug-in instrument to achieve parallel harmony.

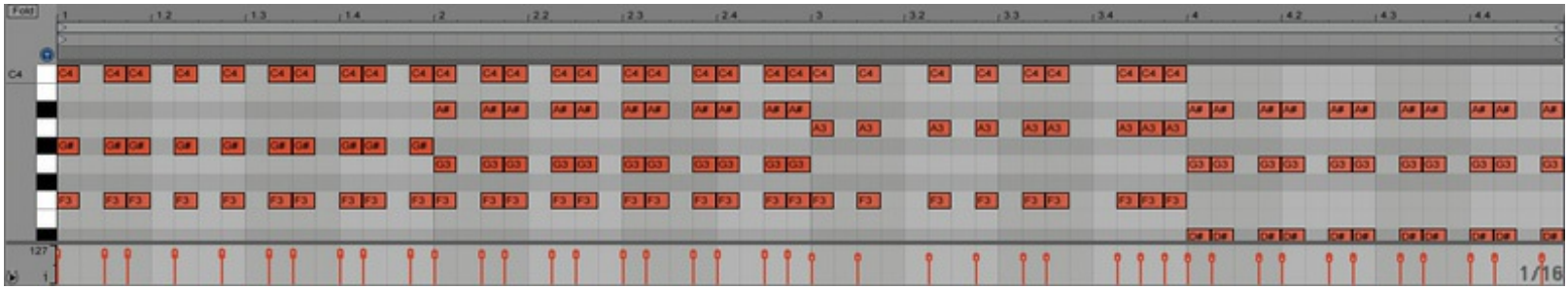
Harmonic Rhythm

Problem:

You have a number of great-sounding chords assembled together into a great-sounding chord progression. But you can't seem to find the right speed to move from one chord to the next. You're quite happy with the rhythm of the music itself, but this doesn't seem to offer any clues for how to control the speed of the harmonic changes.

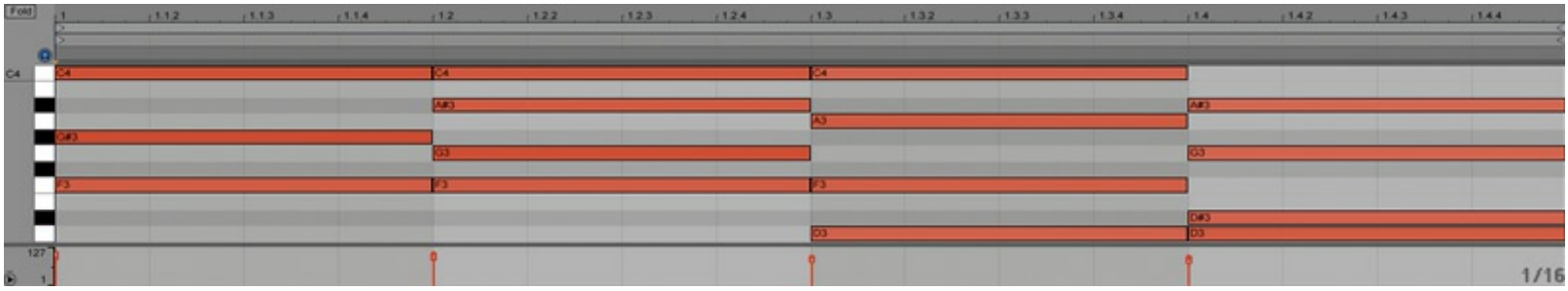
The rate at which the chords change in a piece of music is called the harmonic rhythm. This is a separate concept from the rhythm of individual notes and other events, which we can refer to as the surface rhythm. There is not necessarily any relationship between the surface rhythm and the harmonic rhythm. For example, it's certainly possible to have very active, fast surface rhythms that repeat or outline a single chord for a long time before changing to the next one.

Such a piece would have a slow harmonic rhythm. In the example below, a highly active, syncopated sixteenth-note pattern creates a fast surface rhythm. But the chords change only once per bar:



Likewise, we could have slow music but with a chord change on every note. This would likely result in a relatively fast harmonic rhythm. In the example below, the surface rhythm moves in quarter notes. But the chords also change once every quarter note. The result is a harmonic rhythm that's four times faster than the previous example but with a

slower surface rhythm, resulting in a slower apparent rate of activity.



Here are some ideas for how to use harmonic rhythm in interesting ways in your own music.

Solutions:

The examples discussed earlier are actually fairly common patterns that govern the relationship between surface rhythm and harmonic rhythm. As a general (but by no means universal) rule, music with a faster surface rhythm tends to have a slower harmonic rhythm, and vice versa. For example, if the instruments that define the harmony are moving in fast notes (like sixteenths), you'll generally get good results if the harmonic rhythm moves more slowly (perhaps one chord per bar), while if those instruments are moving in slow notes (like quarters), you might consider a faster harmonic rhythm (like one chord per beat or two beats).

Another interesting way to use harmonic rhythm is to vary it to create new formal sections. For example, for your bridge or breakdown section, you might use the same basic chord progression that you used in a verse or chorus, but at half the speed. In this case, it might also be interesting to slow down the surface rhythm as well, to create a sense of anticipation before returning to the original pace when returning to the "main" sections of the track.

Conversely, you might want to think about speeding up the harmonic rhythm as a way to end the song. For example, if your last section repeats a four-bar chord progression, try condensing the same progression into two bars to give a greater sense of forward motion. (This is a variation on the ideas discussed in Short Loops as Endings.)

~ Note: Harmonic rhythm is only affected by instruments that actually contribute to the harmony. Drums and other non-pitched instruments, for example, have no relevance here. It is certainly possible to have fast rhythmic activity in the drums while also having very slow rhythmic activity in pitched instruments. In this case, you will probably only want to consider the sonic effect of the pitched instruments when thinking about your harmonic rhythm.

Repetition and Insistence

Problem:

You struggle to create bass lines, chord progressions, and melodies that work well together. But some music seems to work perfectly without any of these relationships. The notes seem almost arbitrary, and yet they always sound "right." What's happening in this music and how can you use these techniques in your own work?

How is it that some music—like house or trance—depends so heavily on harmonic relationships, while other music—like acid or minimal techno—often seems to work without any notion of harmony at all?

Solution:

Repetition can have a powerful and unexpected effect on the listener. As long as it's relatively brief, the most angular, atonal musical pattern will eventually sound completely appropriate if it's repeated persistently enough. Through regular exposure and insistence, we can't help but find patterns in chaos. As discussed in Implied Rhythm in Short Loops, you can use this perceptual illusion to find interesting rhythms in any source material, but you can also use it to create melodic patterns and bass lines.

It's common in acid techno, for example, for bass lines to be comprised of notes that are essentially arbitrary. But because these patterns are both short and endlessly repeated, they sound perfectly natural. This doesn't work for every genre—if you're writing music that's rich in harmony, with long, sustained chords supported by a bass line, it's likely that the bass notes really matter. But you'll note that in acid and minimal techno, there's often no harmony at all. This allows for more

freedom in how melodies and bass lines can be constructed, and dissonance often works perfectly well in this music.

If you have access to a plug-in or instrument that can generate random sequences, try feeding a typical 303-style monophonic bass synth and see what kind of results you get. At first listen, some patterns might sound unusable. But allow yourself to listen to the pattern loop a number of times and you'll find that it will start to become musical. In principle, this will work with almost any pattern.

Of course, your ear needs to be the final judge of the results, and you're ultimately responsible for any random processes that you apply to your music (see Randomness and Responsibility). But in many cases, you can make almost any pattern work if you're willing to repeat it with enough insistence.

The Rhythm of Automation

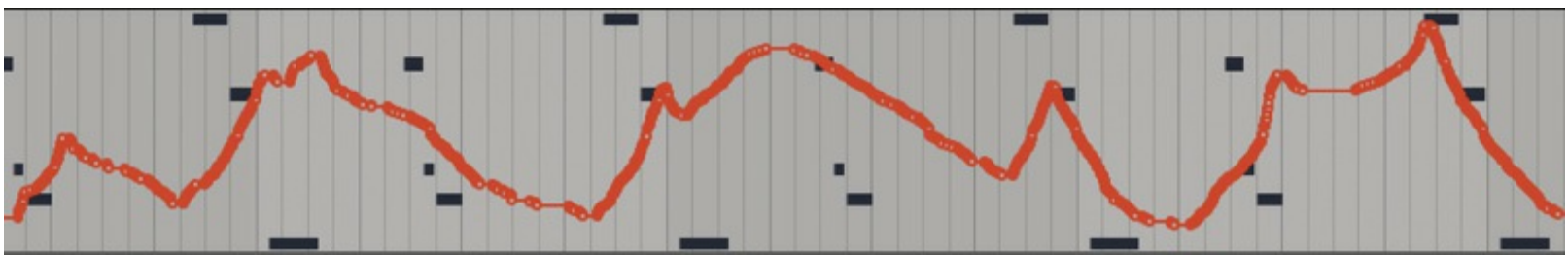
Problem:

You know that you can use automation to give more motion to your music, but you're not sure how to use it effectively or creatively. You end up falling back on the same standard patterns: long, smooth filter sweeps, slow, steady pitch rises, or simple adjustments to mix parameters to keep the parts in balance over the course of the arrangement. What are some ways that you can make more creative use of automation?

Automation is perhaps most commonly used to precisely sculpt a mix to ensure that levels will be perfectly balanced at all points of the song. But automation can be used for so much more than just simulating an engineer's hands on the mixer faders. Here are some ideas for using automation to add an additional layer of rhythmic interest to your music.

Solution:

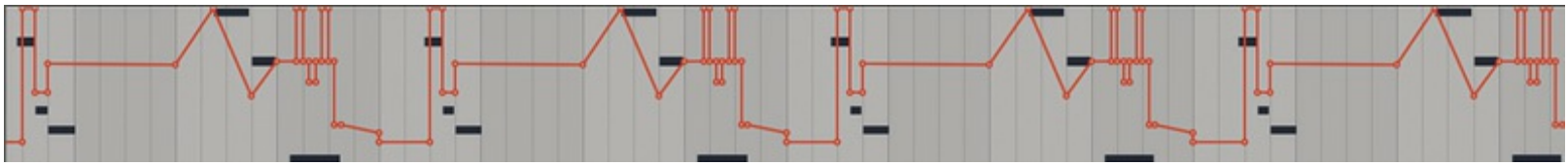
Most of the time, automation is recorded in real time; the producer moves a fader or knob as the transport runs, and the motion of that control is captured as a series of points that approximates a curve that is generally smooth. The resulting automation envelope may look something like this:



The overall shape here looks like something you'd imagine a human

would make; there are occasionally somewhat “sharp” changes, but there are no instant ones. Also, there seems to be no real relationship between the automation envelope and either the underlying metric grid or the note events. It looks like something that was created by moving a physical control by hand.

But there’s no reason why automation envelopes can’t have fundamentally different properties. If automation is created by manually placing breakpoints or envelope segments at specific metric positions, the resulting envelope will have a dramatically different character. Contrast the previous example with this one:



The sharp edges and instant changes look more like the waveforms used in synthesis than they do any kind of conventional parameter envelope. This is an envelope that is inherently rhythmic. By using envelopes like this, you can add an additional layer of activity that serves as a counterpoint to the rhythm created by the note or audio events themselves.

Rhythmic envelopes probably don’t make sense for all types of parameter automation. For example, real-time recordings of parameter changes will probably work best for things like the final mix adjustments, in which you’re just trying to control the relative volume balance of the various parts. But for any creative applications—synth parameters, filter or other effects settings, or even panning—you might find that you get more interesting results by using automation envelopes that are rhythmic.

Humanizing With Automation Envelopes

Problem:

When live musicians play instruments, there are subtle, constant variations in the tuning and length of notes that help to create the sense that the music is "alive." How can you recreate this effect in electronic music?

On most traditional instruments, all or most of the stages of the amplitude envelope are entirely under the control of the performer. And on some instruments (such as woodwinds, brass instruments, and stringed instruments without frets), the player also controls the tuning of each note. But most electronic instruments use an envelope generator to determine the amplitude characteristics of the sound, and use conventional equal-tempered tuning to determine pitch. This makes these instruments predictable and easy to play, but at the expense of some of the subtlety and expression that gives non-electronic music its human character. Here are some ideas about how to use your DAW's automation envelopes to provide some humanization to any electronic instrument.

Solution:

Most of the time, synthesizer programming involves setting the parameter values that define the sound and then mostly leaving them alone. Many producers will employ some automation to give the sound a sense of motion, but this is often applied to just a few controls—filter cutoff and maybe resonance. But by applying even subtle automation to a few more parameters, you can create much richer and more organic sounds.

Rather than treating amplitude envelope parameters as static values, try employing automation to change these values over the course of a phrase or even over your whole arrangement. In particular, by varying the attack and decay times, you can create subtle changes in the perceived duration of notes. In most cases, the actual parameter values used won't really matter much. This means you might also be able to achieve a similar effect by using parameter randomization tools (if your DAW provides them) rather than needing to create long automation envelopes. But even if you do choose to use automation, you'll likely find that this is substantially less work than editing the durations of individual MIDI notes.

Additionally, try using automation to create small variations in the overall tuning of the synth. It's critical that these changes be extremely small. The goal is to simulate the kinds of human intonation "errors" that give live instruments character, but tuning shifts of more than a few cents in either direction are likely to simply sound out of tune. And as with amplitude envelopes, the actual tuning values probably won't really matter, which means that you might be able to get the same effect via randomization.

By applying subtle automation or randomization to envelope and pitch parameters, you can emulate a lot of the subtle characteristics that make live instruments sound so organic. But there's no reason to stop with only these parameters. Modern synthesizers, samplers, and effects often have a huge number of adjustable parameters, many of which are either ignored outright or are set once per sound and then forgotten. Try experimenting with subtle automation changes to any of these parameters as well. You may find that the results sound artificial and strange, but it's possible that this is exactly the character your music needs.

Maximal Density

Problem:

You've completely finished the first section of a song but don't know how to go on from here. Considering how much time and effort you spent on the first part, adding a second section feels like creating an entirely new project. And since you know that there are many sections to go, the end of the song feels like it's an impossible distance away.

Moving forward with a track that's in progress can sometimes feel like an impossible task, especially if you're starting from the beginning and working your way from left to right towards the end. Here's a method for approaching the process "out of order" that may sometimes allow you to work faster.

Solution:

Instead of proceeding linearly from left to right in time, try starting with an arbitrary section in the middle of the song that is maximally dense; that is, a section in which every possible element that could ever happen simultaneously is actually happening simultaneously.

This song section may be purely hypothetical; by stacking every part on top of every other part you may end up with an unmusical mess which will never actually become a part of the song. If that's the case, begin subtracting elements one by one until you get to a level of density that is still high but which makes musical sense. This can then become the high point of your song.

Once you have this peak section, you can make duplicates of it and then try out various lower-density versions of the same basic material by

muting or deleting various elements or tracks and auditioning the result. Each time you find a combination that you like, copy and paste it so that it becomes a new song section. The act of arranging, then, is just a matter of organizing these sections of varying density in an order that makes sense with the ebb and flow that you want for the song.

Beginning with a section of maximal density is similar to Arranging as a Subtractive Process but is focused on the compositional stage rather than on the arranging stage. Although both ideas require that you have a pool of materials to work with, Maximal Density doesn't require you to lay them out into a song-length arrangement, but rather to stack the ideas vertically so that they comprise a single, dense section. From here, you can build your arrangement by creating variations that are texturally-reduced versions of the "master" section.

Note that this idea may only work in this very literal way if you're working in a genre such as minimal techno, in which there is not necessarily the expectation that the various song sections will be radically different from one another. In other genres in which there might be more contrast from one section to another, you may be able to take the idea of Maximal Density only so far. But even here, it can still be useful to create variations of a song part that might appear in variations throughout the song. You could even repeat the process for each section: Create a maximally dense verse, a maximally dense chorus, etc., each of which serves as the parent for smaller variations of the corresponding sections.

Deliberately Bad Listening

Problem:

The acoustics in your studio are perfect, or at least you know their imperfections well enough to account for them in your own listening. And when you listen back to your music during production, you feel like you're hearing everything there is to hear. But sometimes, when you leave the track playing and walk around for a moment, you find that the music suddenly takes on an entirely different perspective. What causes this feeling, and is there something in it that you can use to your creative advantage when composing?

Room acoustics, monitoring options, and listening positions are popular topics on the production side of electronic music-making, but there may also be things we can use (or misuse) in these areas to help us on the creative side.

Solution:

Instead of always trying to maximize the quality of your listening environment, try occasionally listening in a deliberately "bad" way. Particularly during the creation phase, doing this might help you hear things that are acoustic illusions which you might find musically interesting enough to actually incorporate into your music.

For example, rather than exclusively listening from the "sweet spot" in your studio, try regularly varying your listening position, and see what sorts of sounds emerge. Standing in the corner of your studio, for example, may dramatically change the way you perceive bass and other low-frequency components. Although you'd never want to do this when mixing, doing it while composing might reveal unusual bass elements

(such as phantom pitches) that sound great. At this point, you'll need to take note of what these artifacts are so that you can actually create them in your song.

A more extreme version of this idea is to listen from an entirely different room—either through a wall or through a closed or open doorway. Because of the way different frequencies transmit through walls, you are likely to hear a completely different mix when listening from outside of the studio. Again, this is not a suggested way to approach mixing your track, but the artifacts and imperfections you hear as a result of this "bad" approach to listening may be compositionally useful. (Note: Obviously, if your studio is completely soundproofed then this technique won't work unless you leave the door open.)

One interesting approach when composing with headphones is to take the headphones off and turn the volume up high enough so that you can still hear the music. You will generally lose all of the mid and low frequencies when listening this way, and all that will be left is the high-frequency content. But you may hear relationships between parts (or even artifacts like phantom notes and rhythms) that are entirely absent in the music itself.

Finally, another type of "bad" listening is to turn the volume down so that the music is just barely perceptible. This can yield similar results to the headphones suggestion mentioned earlier but will generally preserve a bit more low end. However, because our ears are not equally sensitive at all frequencies, you're likely to hear certain ranges more clearly than others at low volumes, and this too may help to reveal phantom elements that might otherwise go unnoticed.

Remember—with all of these techniques, you're listening for illusions: elements of the music that aren't really there, but which are suggested

by the strained or otherwise imperfect listening conditions. Once you find these (and provided you like them), you'll need to figure out how to get them into the music itself so that they can be heard by listeners in normal listening environments as well.

Randomness and Responsibility

Problem:

You have some "randomization" plug-ins that promise to create a never-ending supply of new ideas without much effort. In principle, you're excited by the potential of making lots of music with little work. But in practice, the results you're getting are unsatisfying for some reason.

There are some types of music, like classic IDM, breakcore, etc., in which constant change is a defining characteristic of the sound. Particularly with drum programming, music in these genres can be extremely time-consuming to build by hand, requiring lots of deliberately placed samples, micro-edits to parameter automation, etc. For this reason, there are a number of plug-ins and other tools available that process an incoming stream of audio and reorder it rhythmically on the fly (these often have the word "buffer" in their names or descriptions), apply effects to only specific time slices of the audio, or perform some combination of both processes.

On the other end of the musical spectrum are genres like minimal or dub techno, in which the overall rate of sonic and musical change is much slower. But even here, there are times when applying subtle randomness to some aspect of the music can kick you out of your normal patterns of production and help you find ideas that you wouldn't have found through your own deliberate action.

And when working in any genre, there may be times when you know you want something to change, but you don't care about the specific aspects of how it changes. For example, you may want the panning of a particular element to vary continuously throughout the track, but drawing in a detailed automation curve may be too time-consuming. In

this case, applying some kind of randomizer to the panning may give you what you want—a continuous state of change, without the need to specify the details.

But sometimes the results just don't work. Here's a solution.

Solution:

Randomization tools can be powerful, but they lack the most important element that makes your music work—your own taste. When you're in control of every element, you're making conscious micro- and macro-level decisions about each moment of the music. If you play a wrong note (or the right note at the wrong time), you hear it immediately and fix it. But when a tool has generated this wrong note for you, it's easy to miss.

To go back to the early IDM example, consider classic tracks by artists like Squarepusher and Aphex Twin. During this era, there were no plug-ins that would accomplish the micro-editing that's happening in this music. What you're hearing is a process that was done moment by moment, by hand.

This is not to suggest that randomization tools are "cheating" or should be avoided. There's nothing inherently wrong with finding ways to eliminate unnecessary work. But it's important to keep yourself fully invested in the process. Although the secrets behind what happens in your studio are yours, in the end, your name will be on this music. Even if it's made entirely by a machine via some sort of algorithmic composition process, it is still your job to evaluate the results that are generated, discard the ones that don't work, and keep the ones that do. The less you're involved in the creative minutiae, the more you need to

be involved in actively auditioning the results.

In terms of specific technical steps, try slowing things down, soloing them, or bouncing the results to audio so that you can really assess whether the randomized output has the musical effect you want. If you're working with randomizers that generate or alter MIDI data, try re-recording this output to another MIDI track so that you can capture the randomized material as "real" MIDI data, which you can then edit later. One nice way to test the limits of your randomization processes is to increase the amount of randomization until it's clearly too extreme, and then dial it back from there. The "right" amount is totally subjective and depends on the genre you're working in and, most importantly, your personal taste.

As always, you must be willing to throw material away if it doesn't sound right. Listen to—and take responsibility for—every moment.

Dramatic Arc

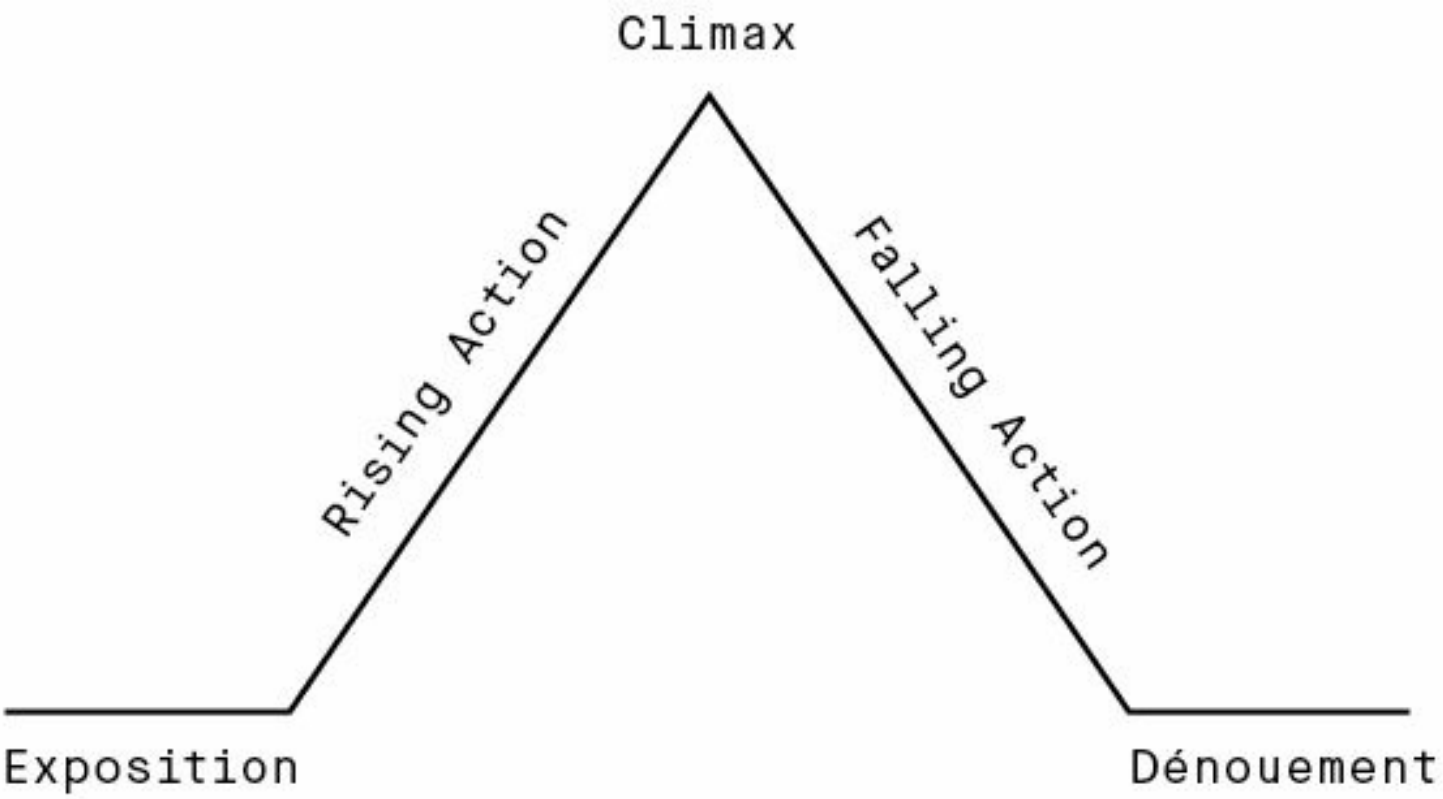
Problem:

You can't seem to create an arrangement that really feels like a convincing musical journey. You can put various song parts together and move them around, but nothing you do seems to create the feeling you get from listening to your favorite music. You find it very hard to create a sense of drama or tension and release. You don't understand how to structure musical time convincingly over the course of 4-10 minutes (and are amazed that a good DJ can do it over the course of hours!).

Solution:

There are many different ways to organize musical materials over time, but one of the most successful is a three-part structure called the dramatic arc that is also commonly used in theater, film, and other narrative or dramatic media.

The dramatic arc refers to three main sections: exposition, climax, and dénouement, which are connected by rising action at the beginning and falling action towards the end. A graphical representation of this arc (known as Freytag's pyramid) is below:



Exposition: In stories, this is the introduction of characters and the setting of the scene. In music, the exposition is the introduction of musical materials (melodies, harmonic progressions, rhythmic ideas, etc.) that will appear through the piece.

Rising action: This is the section when tension builds towards the climax. In stories, this usually means conflict between a protagonist and adversaries. In music, it might mean variations or mutations of the materials that were introduced in the exposition.

Climax: This is the peak of dramatic tension in the work. In a story, this is often the "turning point" for the protagonists, the point at which fortunes shift from bad to good or vice versa. In music, the climax is often marked by a sudden increase in textural density (e.g., "the drop" in dubstep or EDM).

Falling action: This is a section of relaxing tension after the climax. In stories, this section serves to resolve outstanding conflicts from the climax. In music, this section might mirror the activity from the rising

action section.

Dénouement: In stories, the dénouement is often a mirror of the exposition, in which characters return to normalcy after the events that make up the tension of the story. In music, this might mean a restatement of established musical materials (e.g., a repeated chorus) or a gradual dissolution (e.g., a fade out into silence).

While the dramatic arc is normally thought of as a formal structure for the piece as a whole, in music it can also be used as the structure for smaller- or larger-scale components. On the micro level, a single melodic idea can be structured as a dramatic arc; consider a melody that rises to a peak before falling again to its starting pitch. (An example like this is quite literal, with a melodic contour (see Creating Melodies 1: Contour) that visually matches the shape of Freytag's pyramid. Of course, it's also possible to build tension without literally ascending to a higher pitch.) On the macro level, a good DJ set often follows something like a dramatic arc as well, with tension building to a climax over the course of an hour or more, before resolving back to a lower energy level over a similar amount of time.

There are, of course, many types of musical forms that bear no resemblance to the conventional dramatic arc. Genres like dub or minimal techno, for example, generally have no clear build to a climax. In this music, dramatic tension is created in entirely different ways, often through careful attention to sound design and subtle changes in an otherwise static texture over time. But in some situations, using the dramatic arc as a model for musical form can work extremely well.

Problems of Finishing

Arranging as a Subtractive Process

Problem:

You have more than enough ideas to make up a finished song but don't know how to actually put the arrangement together. Even the process of arranging sounds like an intimidating commitment. How can you even begin, let alone finish?

In the context of music creation in a DAW, the arrangement refers to the layout of the parts of your song along a timeline. You may have assembled a rich pool of material, but the actual act of putting that material in some kind of order that unfolds over time is what will eventually turn that material into a finished song.

Getting from "pile of stuff" to "song" is a difficult process, both conceptually and technically. The most common way people approach creating an arrangement is the most obvious one: Gradually fill the empty arrangement with various combinations of the material you've made, moving from left (the beginning) to right (the end). In this workflow, the arrangement timeline is analogous to a blank canvas to which you apply paint until a finished painting appears from what was originally empty, white space.

This process works, of course. But facing emptiness can be scary. Even though you've already put in a considerable amount of time preparing the materials that you plan to use, you now face something that might feel like a reset to zero. Beginnings are hard, and a blank canvas (or empty timeline) can be a difficult mental bridge to cross.

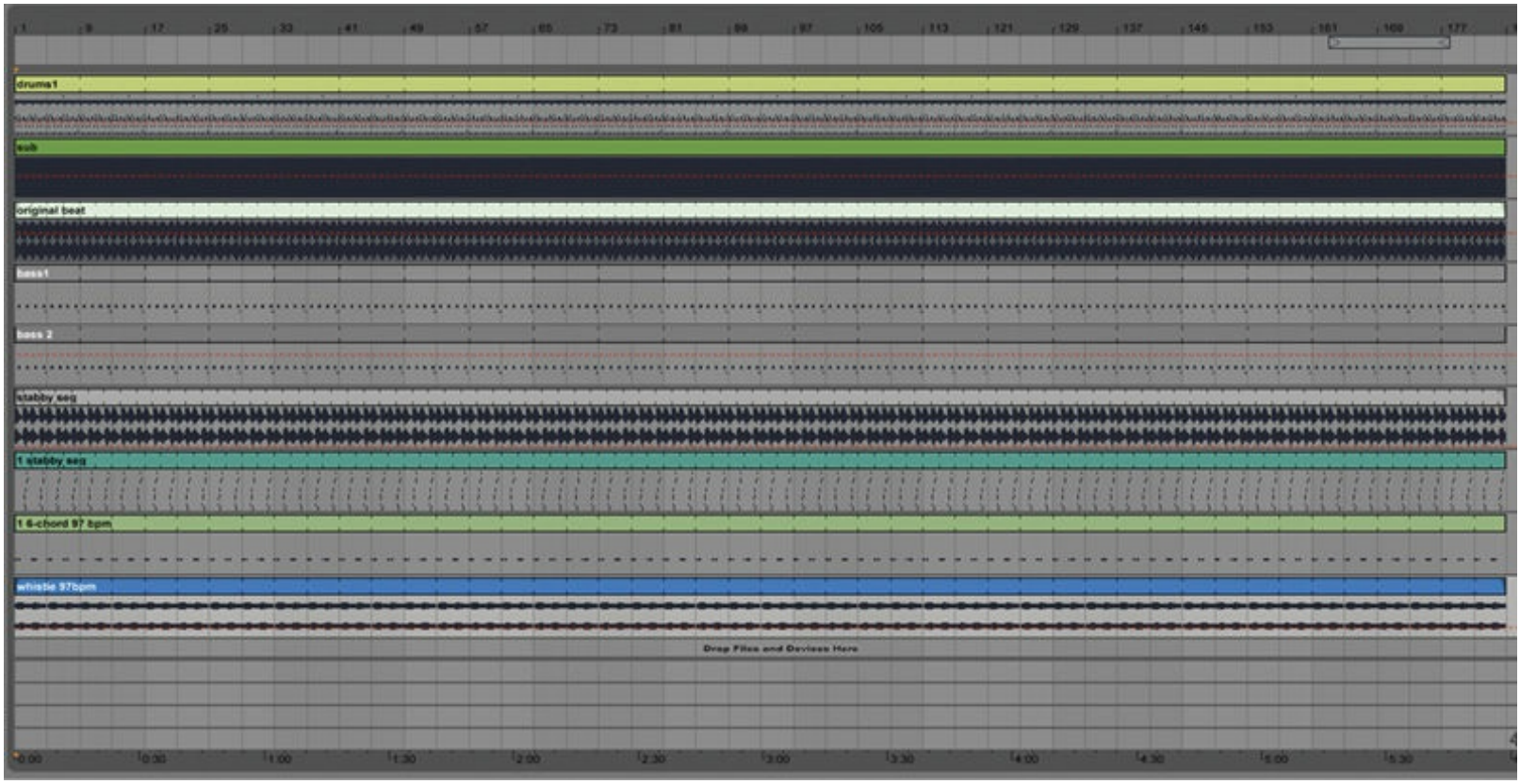
Solution:

- If you're finding that you're stuck at the arranging stage, here's one

If you're finding that you're stuck at the arranging stage, here's one process that might help: Start by immediately filling your entire arrangement with material, on every track. Spend as little time as possible thinking about this step; the goal right now isn't to try to create a good arrangement. You just want to start with something rather than nothing. It's OK that you don't know how long the song will eventually be. Just fill up an average song's worth of time (or even more) in whatever way is the fastest for your particular DAW—by copy/pasting blocks of clips over and over again, via a "duplicate" command, or (in some DAWs) by dragging the right edges of clips to extend them.

Perhaps you've already given some thought to how your material will be divided. Maybe you've named certain clips things like "Verse" and "Chorus" so that you can better organize them when arranging. Don't worry about any of that for now. In fact, don't even try to use all of the material you have. Just grab a pile of ideas from each track, and fill the empty space. This process should take no more than about 20 seconds. If you're spending more time than this, it probably means you're trying to make creative decisions. For example, maybe you're thinking "I already know that this chunk of ideas will go before this chunk of ideas, so to save time later, I'll just lay them out in that order now." Resist the temptation to organize anything in this phase, and simply move as fast as possible.

As an example, here's a 6-minute arrangement timeline, filled as quickly as possible.



Now that you've filled the timeline, the process of actually making your arrangement into music becomes one of subtraction rather than addition. If the traditional arranging workflow is analogous to painting, the subtractive workflow is analogous to sculpting. You're beginning with a solid block of raw material and then gradually chipping away at it, creating space where there used to be stuff, rather than filling space that used to be empty.

This can be a much more productive way to work for a number of reasons. For example, it's often easier to hear when something is bad than it is to imagine something good. If a particular combination of ideas doesn't make musical sense, you can generally feel this right away, and the steps to fix it may be obvious: Maybe an element is simply too loud, or the bass line clashes with the harmony. And because you're listening back to an actual flow of sound over time, you'll probably have an intuitive sense of when a song section has been going on for too long —your own taste will tell you that it's time for a change.

If you're working in a genre in which textural density tends to increase and decrease as the song progresses, you may already have your "thickest" sections of material finished at this point. You may find that you're actually able to work backwards from the end of the song towards the beginning, removing more and more elements as you go back in time.

~ Bonus tip: Most DAWs provide a way to insert or delete chunks of empty time in the middle of an arrangement. When using a subtractive process, these tools can be extremely helpful. For example, you may have finished editing work on what you originally thought would be two adjacent sections of material, but then realized that something else should come in between, or that the first section needs to be twice as long. Inserting time in the middle automatically shifts everything after this point to the right, which is much faster and safer than trying to cut and paste many tracks' worth of material manually. Or maybe you've realized that a section you've been editing is too long. If your DAW has it, use the delete time command to remove the excess material, which will cause everything to the right to automatically shift to the left to fill in the gap.

The Power of Erasing

"It seems that perfection is reached not when there is nothing left to add, but when there is nothing left to take away."
— Antoine de Saint-Exupéry

Problem:

You feel like your music lacks the "fullness" of the music that inspires you. But your attempts to solve the problem often end in music that just feels cluttered or messy.

It's tempting to think of a song as being something like an unlimited physical space, and that the way to fill the space involves adding more stuff. But in reality, musical space has boundaries, like a canvas: There is a limit to how much you can add before you're simply covering something that's already there.

Solution:

On the production side of music-making, it's common to hear about "making space" for each element in your mix. Normally, this means eliminating overlapping frequencies by using EQ to carve out a band of frequencies in one instrument that you'd like to be more dominant in another. But the same process can be applied on the composition side as well (and may even save you from having to do EQ surgery afterwards).

If your music feels "small" or "weak" or "lacking power," the real solution may be to take parts away rather than to add more. It may feel counterintuitive, but the reality is that each part you add requires the others to fight that much harder for your listener's attention. Of course, this paring-down strategy only works if the parts you have are as good as they can be. But if you allow yourself to fill musical space by adding

parts, you run the risk of treating each part with less care than you might otherwise, rather than really investing the time and energy into getting a few elements to be perfect.

This is not an argument for minimalism or for any kind of specific approach to genre. There's plenty of "minimal" music that fits that description because of factors other than textural density. Music with a slow rate of change, for example, is often heard as minimal, regardless of how many elements are happening. Likewise, plenty of maximally active and seemingly dense music (such as some commercial EDM) can still sound clear and open because of a conscious effort to limit the number of simultaneous parts. The important point is to add parts as necessary, but only as necessary, and to be ruthless about removing things that aren't really contributing to the music.

A sense of "fullness" can usually be created by carefully balancing just a few things. If each part is placed in its own space, both in terms of composition and production, then a song has a better chance of feeling "complete" than it does if many parts are competing for the same space.

In addition to brute-force removal of entire parts, you can often get good results by "thinning out" certain elements. Techniques for this include reducing the number of notes played by a particular instrument or shortening the decay time of sustained sounds so that they quickly get out of the way of the rest of the mix.

If you employ processes such as Arranging as a Subtractive Process or Maximal Density, you may have inadvertently created music that suffers from problems of over-layering. In these cases, you may need to do lots of erasing in order to get back to something that makes sense for your song. This is not to suggest that those processes don't work, but rather that you'll need to be extra careful and really think about whether or not

you’ve removed enough to keep the song musical.

Formal Skeletons

Problem:

You have all of the elements you'll need for a great song, but it seems that no matter how you put them together, the arrangement feels unsatisfying. When you listen to your favorite tracks, you never have this feeling; every part seems to last just the right amount of time, tension and release are balanced perfectly, and the song just "works" exactly as it should.

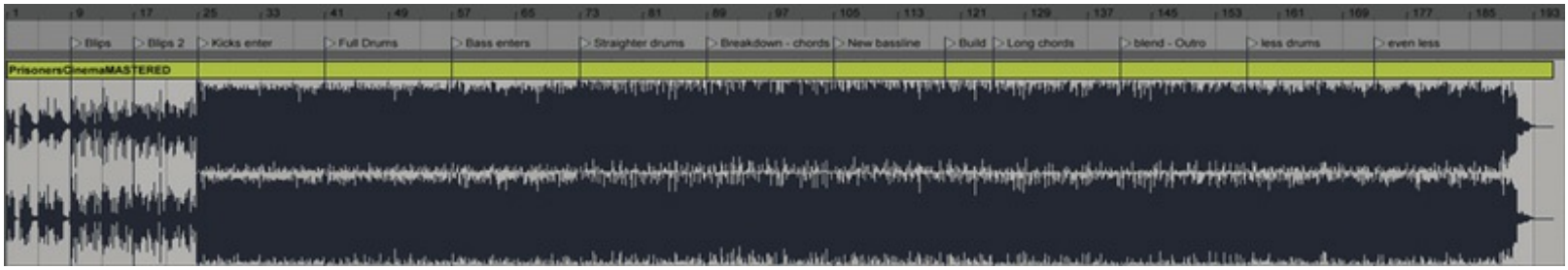
Creating successful arrangements requires having a great internal sense of proportion, an understanding of how long each song section needs to be in relation to the others. Luckily, this is a skill that can be learned, and one of the best ways to learn it is by studying and copying from great examples that already exist.

Solution:

Find a song that has a form that works and that's somehow in the same musical universe as the music you're writing. Then load that song into an empty project in your DAW, starting from the left edge (bar 1). Carefully adjust the tempo of your project so that it perfectly matches the tempo of the song you've loaded. This ensures that your DAW's beat grid will align perfectly with the actual beats in the song.

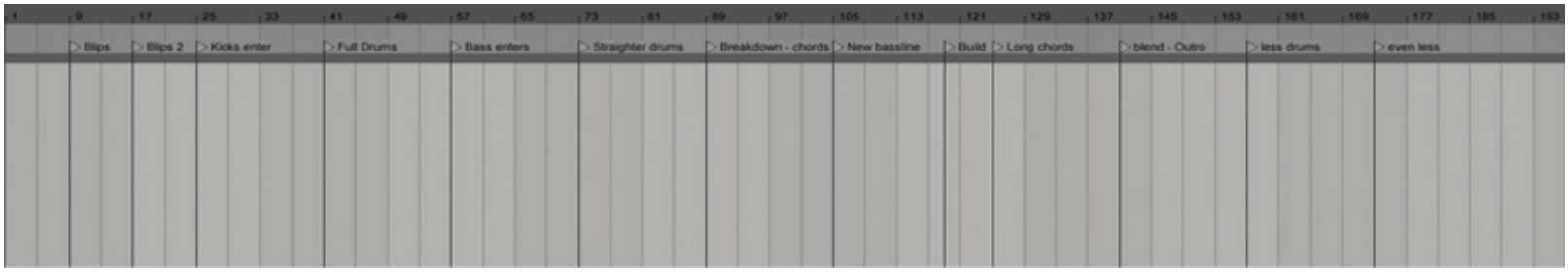
Once you have this set up, carefully listen to the song. When you hear a formal division (or even an interesting moment within a formal section), stop playback and place a marker at that point. Then continue listening from that point until you reach the next formal division, at which point you'll stop and add another marker. Continue this process until you reach the end of the song. If your DAW allows this, give the markers

descriptive or functional names that will help you to understand the musical significance of the marker without having to listen again. If the song is in a genre that uses conventional song structures, you might give these markers names like “Verse,” “Chorus,” etc. If not, you can simply give yourself clues like “Bass starts.” Here’s how a song might look when you’ve finished this process:



As with the above example, in most electronic genres, you’ll probably find that these formal divisions will occur at multiples of four or eight bars. This is by no means universal, but it can be a handy reference when working on your own arrangements—either by nature or because it simply occurs in so much music, listeners seem to be primed to expect formal divisions at these points.

Now you can simply delete the original audio file, leaving the markers behind as a kind of skeletal outline of an arrangement that you know is effective:



From here, you can arrange your song to fill in this formal outline. And because your materials are your own, there’s little chance that you’ll end up duplicating the song you’ve used as a reference.

~ Note: As with all of the ideas in this book, once you actually start employing this in practice, you may find that your own musical materials don't naturally fit into the arrangement skeleton perfectly. Maybe your breakdown section "wants" to be four bars longer, for example. In these cases, you must absolutely trust your instincts. The formal skeleton should be used to get you started, but you should absolutely feel free to move parts around once you're actually in the flow.

Common Forms 1: Elements of Song Form

Problem:

You have no problem creating musical ideas, but you find it difficult to organize those ideas into an arrangement that makes sense. You've heard terms like "verse," "chorus," and "bridge" to describe the sections of some songs, but you're never quite sure what these words really mean or whether or not they're relevant to your own musical work.

Understanding the various types of commonly used song sections can help your understanding of the music you hear, as well as provide options for the music you write. Here's more information about what these sections are, how they work, and how they relate to each other.

Solution:

Across many different genres, there are a handful of terms that are commonly used to refer to the various sections of a song. Much music, especially in genres that are related to pop music, is structured by combining these standard section types in a variety of ways. The most common sections are:

Verse or "A" Section: A song's verse is generally a recurring section—usually 16 or 32 bars in length—that serves as the main body of the song. In music with lyrics, the verse often tells the "story."

Chorus or "B" Section: The chorus is usually also recurring, and of comparable length to the verse. It acts as a contrast to the material of the verse and usually contains the "hook" of the song—a melodic idea that is intended to stick in the listener's head. Often, the chorus serves as a point of musical resolution, while the verse creates musical tension.

Another important distinction between verse and chorus: Recurring verses share the same music but they generally have different lyrics, while recurring choruses most often share both music and lyrics. Additionally, in music with lyrics, the chorus often contains the title of the song. As a general rule, the first chorus in a song occurs after a verse (although there are some songs that begin with a chorus).

Bridge or "C" Section: The bridge serves as a contrast to both the verse and chorus and typically occurs only once in a song. Musically, bridges are often substantially different from the rest of the music in the song; they may be in a different key, employ unusual chord progressions, or have a dramatically different level of textural density and energy. In some types of music, the bridge is used for instrumental solos. Generally, the bridge occurs only after at least one verse and one chorus.

The A, B, and C letter names are often used to create formal diagrams of particular songs, and this can be a useful tool in your own Active Listening or when writing a Catalog of Attributes. For example, one common form in commercial music is **ABABCB** or **Verse-Chorus-Verse-Chorus-Bridge-Chorus**. In some songs that use this basic form, there may be one or more additional choruses added to the end. But otherwise, this form is used unchanged in probably the majority of contemporary pop songs you'll hear on the radio. "Royals" by Lorde is an example of ABABCB form followed exactly.

Although there is a wide variety of possible song forms that can be made just from various combinations of verse, chorus, and bridge, these types of sectional constructions are less commonly used in more underground or experimental music. For example, most contemporary electronic genres without vocals tend to avoid conventional verse and chorus sections, and instead create formal contrast via addition and subtraction

of layers (see Common Forms 2: Layering as Form). But if you're working in genres more closely related to pop music, you can create a lot of music using only these few section types.

Common Forms 2: Layering as Form

Problem:

You have no problem creating musical ideas, but you find it difficult to organize those ideas into an arrangement that makes sense. You've heard terms like "buildup," "breakdown," and "drop" to describe the sections of some songs, but you're never quite sure what these words really mean or whether or not they're relevant to your own musical work.

Understanding how layering and textural density work can help your understanding of the music you hear, as well as provide options for the music you write.

Solution:

In contrast to the structure of song forms (see Common Forms 1: Elements of Song Form), many of the more "pure" electronic genres (such as some techno, trance, dubstep, etc.) avoid the use of sectional designs altogether. Instead, a sense of form in these genres is often established by continuous variation of textural density around a relatively small amount of material—what would sometimes comprise only a single section in more pop-oriented genres.

For example, in modern genres such as American dubstep, a single basic collection of material is often used from the beginning to the end of the track. Contrast, then, is created through layering: the careful addition and subtraction of parts in order to change the density of the underlying texture. There are a handful of terms that are commonly used to refer to the various parts of a song constructed in this way. These terms include:

Buildup: A track's buildup is a section—usually 16 or 32 bars in length, or some multiple of those—in which layers are added, increasing the music's textural density and perceived sense of energy. Buildups should create a sense of drive and forward motion and should end at a high point.

Breakdown: The breakdown is a section of lower energy, designed as a contrast to the buildup, and involves either a gradual or sudden reduction in density. Breakdowns are about removing elements with the goal of creating a sense of space. Some common breakdown techniques include removing all of the drum elements or, conversely, removing everything except drums. Breakdowns are often shorter than buildups but are commonly still in increments of 8 or 16 bars.

Drop: The drop is the climax and is generally the most texturally dense passage of music in the track. The drop can occur after either a buildup or a breakdown, depending on the overall energy contour you're trying to achieve. There are generally no more than one or two drops in a single track, although there are commonly more buildups and breakdowns.

Although the musical results are radically different and significantly more subtle than in American dubstep, similar types of layering principles occur in a variety of other electronic genres as well. Consider, for example, classic dub techno tracks like those released by labels such as Chain Reaction. In most of this music, there is no sense of sectional contrast. A bass line or drum pattern that appears at the beginning of the track is likely to appear throughout the entire track, and if new material is introduced at some point, it is an addition to the surrounding material rather than a large-scale replacement of it.

Layering as a form-defining process doesn't work for every kind of music. If you're working in genres that are closely related to pop music,

you will probably benefit from using the more conventional elements of song form (see Common Forms 1: Elements of Song Form). But for many types of "classic" electronic music, using a small amount of material subjected to ever-changing levels of textural density can be enough to create a sense of structure.

Short Loops as Endings

Problem:

You have an arrangement that's almost, but not quite, finished. The form makes sense, and the pacing, tension, and release all seem to be working. The only problem is that there seems to be no way to actually end it. What can you do to close off the final section when it seems like it could just go on forever?

A common problem when working on the late stages of a track is figuring out how to actually make it come to a real conclusion. A repeating chord progression in the closing phrase of a track, for example, may simply feel "right" if extended indefinitely. Leaving a loop playing continuously is, after all, a common way to work during the production phase of a track, and it's sometimes difficult to exit the producer mindset and enter the world of the listener, who (naturally) expects a track to end at some point. If you simply stop the music, then the ending feels abrupt and rushed. But if you go on too long, you risk ruining the pacing and balance of an otherwise well-made track. Here's a suggestion for bringing a track to a real, and satisfying, ending.

Solution:

One interesting way to end a track is to isolate a loop from somewhere in the song (the closing phrase usually works well) that is somewhat shorter than the phrase itself, and then repeat this loop a set number of times as a final musical gesture. For example, if the last passage of material in your song is an eight-bar chord progression, try isolating a one- or two-bar sub-passage within this material as a closing statement. For the listener, this material will already sound familiar, since it has been previously established in the closing phrase. But by using a short

snippet instead of the full passage, we can create the illusion of speeding up, or a sudden increase in forward motion that drives towards the ending.

Additionally, you may want to further reduce the size of the loop to continue this sense of acceleration into the ending. For example, our original eight-bar loop might first become two bars. Then after repeating this two bars for a little while, we might again reduce the two bars to one bar (or shorter). You will probably start to sense a natural ending point after repeating these shortened phrases a few times. If you still don't hear an obvious ending, try simply stopping the music after a fixed, even number of shortened repetitions (say 4, 8, or 16). Via trial and error, you can often come up with an ending that works.

To add even more drive to the ending, you could try adding some Unique Events to the final shortened loops. This will increase both the apparent speed of activity as well as the textural density, while also breaking up the "purity" of the loops.

Unique Events

Problem:

You've finished an arrangement, and you're generally happy with it. It has a variety of sections, you've created Fuzzy Boundaries between them, they all seem to last for the right amount of time, and there are contrasts and appropriate levels of tension and release. But somehow, it still sounds like "loop music"—repetitive. You can't consider it finished in this state, because it feels too predictable; it won't offer any surprises beyond the first listen.

Modern DAWs suggest a workflow that combines loops into larger arrangements, and many genres of electronic music are defined by repeating patterns. But perfectly repeated loops can become tedious after a while. Here is one technique for creating a sense of surprise within an otherwise loop-based context.

Solution:

At various points in your arrangement, insert unique events—sounds, gestures, or variations that occur only once and are never repeated. There are a few different types of unique events you can try:

- **Single *events***. These are short samples or sounds that can be strategically placed throughout your arrangement to add a layer of unpredictability. These can be almost anything—pitched or unpitched percussion sounds, single notes or chords played on an instrument that is otherwise never heard, etc. A good source of single events is the sample library you already have. Try finding unusual samples that would otherwise be out of context for your song. Depending on where you place these events, the effect on the listener can be quite varied. For example, placing single events at or near formal boundaries can make them feel like part of a transition, while placing them at formally insignificant moments can be quite jarring and interruptive, especially if they're also placed in odd rhythmic locations that play "against" the grid of the song.
- **Single *musical gestures***. These are short phrases or one-time alterations to otherwise

consistently repeating patterns. It's common to vary a musical phrase near a formal transition, but it can be especially interesting if you also create one-time variations within the middle of a phrase. Some DAWs offer various types of note-transformation tools, which can apply processes to a selection of MIDI notes to change some or all of them in a way that maintains a relationship to the original. These tools can provide an interesting way to create one-off musical gestures.

> **Single *processing gestures***. Consider applying dramatic, one-time changes to the effects processing on one or more tracks. For example, set up a chain of effects but leave them bypassed or off. Then use automation to make them active only once before disabling them again. Single processing gestures can be especially disruptive if they're quite short or if they occur in unusual places in a phrase.

Depending on the genre you're working in, you can be quite creative with these unique events. Unless you're working in a very commercial context, you don't generally need to worry that "weird" (i.e., out of key, out of rhythm, etc.) unique events will ruin the texture of the music. But note that you should probably use them sparingly within a single song. Even though each individual event occurs only once, using too many unique events in a song can create its own sense of predictability. Listeners will begin to expect that something jarring or unusual will occur, which reduces the effectiveness of the technique.

Three Ways to End

Problem:

You're staring at an arrangement that's working, but you see no way to come to a convincing ending.

Although endings are usually easier than beginnings, they can still be sources of considerable creative stress. After all, you've worked hard to build a strong arrangement that creates and resolves tension at the right times and in the right proportions. But all of this work could be lost if the ending doesn't "wrap up" the song in a satisfying way.

Here are three common ways to end songs.

Solutions:

1. **Mirror what you did before**. In many tracks, the work of creating the arrangement consists largely of adding and subtracting layers. Parts enter to increase the textural density and energy and then are removed to create sections of reduced energy. One way to write a satisfying conclusion is to create a mirror image of the initial "part accumulation" phase of your track. For example, if your track begins with:

> kick drum alone (8 bars)
> add other percussion (8 bars)
> add bass line (8 bars)
> add harmony and melody (8 bars),

you might then mirror this process to end the track as follows:

> remove harmony and melody (8 bars)
> remove bass line (8 bars)
> remove other percussion (8 bars)
> kick drum alone (8 bars).

2. **Loop and add/subtract**. As discussed in Short Loops as Endings, one way to end is to find a loop from somewhere in the song and simply repeat it. Insistence alone might be enough to drive

the track to a convincing ending, but you can also experiment with increasing the sense of drive by adding additional layers at each loop repeat to build density and tension. On the other hand, you can also subtract or "thin out" parts at each loop repetition (possibly through the mirroring process discussed previously) in order to minimize energy and let the song end with a whimper rather than a bang. Both options work well, and the choice of which to employ depends on taste and musical context.

3. **Fade out**. Fade outs have negative connotations among electronic musicians, possibly because tracks with fades can be a bit challenging to use in the context of a DJ mix. But there are definitely situations in which a simple fade into silence is the right choice. It's important to note the philosophical implication of the fade out: It suggests to the listener that the song might actually never end. With this in mind, it probably only makes sense to fade out if you've settled into closing material that would really work well if left looping indefinitely. Jagged, asymmetrical, or otherwise "harsh" passages probably don't make good material with which to fade out. Another important point to consider is where in the production process the fade out should be employed. If you plan to have your track mastered (or to master it yourself), it may be better to wait until the mastering stage to do the fade. This is because the sound of a mix fading can change dramatically depending on whether it happens before or after final mastering compression.

Rendering as Commitment

Problem:

You're nearly finished with the song. The arrangement is perfect, the mix is coming together, and you expect that you're just a few minor tweaks away from something you'll be really happy with. But then you start thinking "what if I could make these sounds even better?" So you go back to your synths and start tweaking knobs. One change leads to another, which leads to additional sound design work on another track, and now hours have passed and you're no closer to done—in fact, you feel like you've gone backwards, and you're just programming kick drum sounds again, like you were at the very beginning of your work on the track.

This scenario is especially common for people who work entirely in software. With total recall of every setting possible just by loading the project, we're able to easily go back and rework even the tiniest details of both the music and the sounds, regardless of what stage of the process we're at. In fact, there isn't even really a need to think in terms of "stages" anymore; we can seamlessly move between sound design, composition, arranging, and mixing right until the point at which we export the final file.

But this kind of flexibility can actually be a detriment to our real goal, which is getting music done.

Solution:

If you find that your desire to tweak the low-level details gets in the way of your ability to finish, consider adopting a workflow that simply takes away your ability to tweak. One way to do this is by rendering to audio

much earlier in the process than you normally would.

Although modern DAWs blur the line between editing MIDI and audio, there is still a huge range of latent possibility (and thus distraction) available just by having access to the source instruments. By rendering to audio—and then removing the instruments that made it—you're forced to change your mode of interaction with your material. You can't change patches anymore, or adjust parameters. This forces at least the "programming synthesizers" part of the sound design process to come to an end. By closing certain doors behind you, the only way you can move is forward—writing, arranging, and mixing the song. This is essentially an Arbitrary Constraint self-imposed for the sole purpose of eliminating possibilities.

Additionally, if you're collaborating with other people, rendering to audio streamlines project portability. It's no longer necessary for you and your collaborators to have exactly the same plug-ins, and you can be sure that shared files will open correctly on any machine. And this could also allow you to more explicitly divide up the collaborative responsibilities. For example, one of you could be entirely responsible for sound design and idea creation while the other handles arranging and mixing. By working only with rendered audio, the second person is forced to focus more on their specific roles and is unable to easily dip back into the sound design mindset.

Finally, if you're working with plug-in instruments rather than hardware ones, an additional side benefit of rendering early is that it significantly reduces your DAW's processing requirements. Working with audio is much less resource-intensive than working with virtual instruments, and constantly having to deal with an overloaded machine can itself be a damper on creativity, so working this way helps to solve this problem almost by accident.

And of course, this technique applies to more than just instruments. Including the sound of effects processing in your "early" audio renderings can be an even more extreme level of commitment. It may also help you to find new creative uses of effects, because you can be more free to capture the sound of a particular moment of processing experiments, rather than thinking of effects as locked to particular tracks. For example, you could work with an effect for a while, render the audio of its output, delete the effect, replace it with another one, and repeat.

If you own multiple DAWs, you could even consider switching between them for different stages of the production process, and rendering to audio becomes a necessary step in this case. For example, maybe you use one DAW for sound design, because you prefer the sound of some of its native instruments or effects. But you prefer another DAW for mixing because you like the look of its metering.

If you're really worried that rendering is too much of a commitment, but you still want to take advantage of the workflow benefits that rendering provides, you could try the intermediate step of saving multiple versions of your project, both pre- and post-rendering. This way, you can choose to work with the audio-only version but still go back to the older version containing the original instruments if there's a low-level change that you really need to make. The risk of working this way, however, is that any edits you make to one version will need to also be made to the other in order to keep them in sync.

Getting Feedback

Problem:

You recognize the value of feedback, but you're self-conscious about asking for other people's opinions of your music. In addition, you don't know *when* to ask for it; should you ask people to critique your in-progress tracks or only ask when things are finished (when you'll be more reluctant to make changes)? Finally, whom should you ask? Friends and family? Other producers? The anonymous internet?

Especially if you're young or new to making music, it can be difficult to have an internal sense of whether or not your work is any good. For new producers (and even for veterans), feedback from others can be helpful, but it's important to get feedback at the right time and from the right people. Here are some tips for how to improve your process of collecting feedback.

Solution:

If you're just asking for feedback because you're seeking praise, then you're not really looking to improve. The most important factor to consider when getting feedback on your creative work is whether or not you're prepared to accept criticism. And to be sure that you'll get criticism, you need to ask people who you can be sure will be honest with you. Your mother is likely to be a fan of everything you do. But unless she's also a musician, she may not be in a position to give you honest, usable feedback. Getting an ego boost can be valuable for your mental health, but beyond that it probably won't help you make better music, so be sure to consider whether or not family and friends are a good source of useful feedback or if they're instead just making you feel better.

The best people to give you constructive—and critical—feedback are probably going to be other people who are really invested in the kind of music you're trying to make: either other producers or, at least, very serious and knowledgeable fans. But even then, you need to consider the relationship between you and your critic. A teacher, for example, would probably give very different feedback to a student than that student would give to the teacher. Be careful when asking for feedback from people who are your subordinates or who might otherwise have a reason to stay on your good side. As with friends or relatives, they might just be telling you what they think you want to hear.

Getting helpful feedback from the internet can also be a challenge. Websites and forums that are made by and for music producers tend to yield feedback that is more constructive than sites that allow for truly anonymous comments (such as YouTube).

Your most valuable and astute critics, no matter how well intentioned, will probably find it difficult to objectively evaluate your music before it's finished. This is because they'll be evaluating it in comparison to tracks that are not only finished but also mastered. We naturally hear music as better even if it's just louder, and it's difficult to overcome this biological hardwiring when trying to critique music that's unfinished. One idea might be to have a "quick-and-dirty" mastering preset available that can at least get the level of an unfinished track into the general volume range of released material. This is, of course, not a substitute for real mastering. But if you're interested in getting feedback on work in progress, it will help to allow people to listen to the music without being distracted by the low volume.

Also, when asking for feedback, be sure to ask for specifics. Asking someone if they like the song will get you a yes or no answer, which isn't really of much use. But asking for details—about sounds, form,

and other fine details—will likely yield much more useful feedback about your musical decisions.

Remember that once you've released something into the world—whether on a label or just on your own via the internet—you'll get criticism no matter what. You may find it beneficial to hear some of this feedback ahead of time by asking for it early. Just make sure you do it at the right time and from the right people.

Diminishing Returns

“The perfect is the enemy of the good.”
— Voltaire

“Art is never finished, only abandoned.”
— Leonardo da Vinci

Problem:

You regularly notice that your rate of progress gets slower and slower as you get closer to the end of the song. The beginning creation phase is very fast, and this always feels inspiring. But endings seem to take forever. You’re constantly going back to refine, never quite sure that things are as good as they can be. As a result, you get increasingly discouraged as time goes on, and the final stages of work on a song are the most time-consuming and painful.

Most producers are perfectionists. We want our tracks to be better than the ones that inspire us, and at least as good as some abstract ideal that we have in the back of our mind for what our music should be. But in reality, perfection is unattainable, and continuing to tinker with a song in the very late stages may actually make things worse.

Solution:

As difficult as it might feel in the moment, it’s important to learn to recognize the point at which the song is “good enough.” This is not the point at which you can continue to make real improvements; if continuing to work is really making things better, then you should continue to work! Instead, this is the point at which continuing to work will yield meaningless or arbitrary results. Learning where this point is in the production process is different for everyone. But if you regularly

find yourself endlessly tweaking what is essentially a finished mix, you've probably reached the point of diminishing returns.

Most of the time, this late-stage tweaking is the result of a fear of commitment. Once we decide that the song is finished, we might be stricken with doubt: "If I had worked harder, would it have been better?" But it's important to realize that, by this stage, you've probably already made all of the hard decisions that need to be made. At best, continuing to tweak will probably just waste time that could be better spent getting to work on the next track. In the worst case, you might actually go backwards and make the track worse. This is because your initial decisions often prove to be the right ones, and the more time you spend second-guessing yourself, the more likely you are to override your instincts in a negative way.

Although it will take time and concentration, learn to find the point where the track isn't going to benefit from further work, and train yourself to stop there. In the long run, you'll finish more music without sacrificing quality.

Fail Better

“Ever tried. Ever failed. No matter. Try again. Fail again. Fail better.”
— Samuel Beckett, Worstward Ho

Problem:

The closer you get to finishing the track, the more you realize that it’s a failure. It will be impossible to turn this into something you’ll be proud of. Why bother finishing it at all? Wouldn’t it make more sense to just abandon it and start over on a completely different project?

It’s depressing to realize that you’ve made something bad. It’s even more depressing to realize it while you’re still working on it but after it’s beyond any hope of salvation. But in this situation, there are still valid reasons to keep going and finish the track anyway.

Solution:

Most producers have started far more tracks than they’ve finished. It’s much easier to give up in the middle of a project and move on to something new than it is to see a project through to the very end.

But what most producers don’t realize is that each stage of the music-making process is itself a thing that requires practice. We get to be better sound designers by designing sounds. We get to be better drum programmers by programming drums. And we get to be better song finishers by finishing songs. Because of this, the more songs we start but don’t finish, the more opportunities we miss out on to practice finishing. And as a result, we might continually improve at various aspects of the early stages, but we’ll never improve at actually getting things done.

If you realize very early that what you're working on isn't going to be successful, you probably have time to change directions and make things better. But if you're very late in the process of making the track before realizing that it's not good, it might be too late to fix it without completely gutting it (which is essentially the same as starting over). In these cases, forcing yourself to finish—no matter how painful—is often better than giving up. You'll not only get practice finishing, but you'll also get practice failing, in itself a valuable skill to learn in a subjective and unpredictable business like art. The better you get at finishing, and the better you get at coping with failure, the better your chances will be the next time you begin (and, hopefully, finish) a project.

If the track is really as bad as you think, maybe there is a natural end point that's earlier than where you'd stop with a track you were happy with. For example, it might not make sense to get your new track professionally mastered. And it might be a good idea to not share it with the public. Maybe it just goes right back into your Scraps and Sketches folder, to be pulled apart for use in other tracks later. But the important thing is that you actually finish the arrangement, if for no other reason than to practice, improve, and experience how it feels to finish.

Credits

Edited by Mark Garvey.

Proofread by Rose Knudsen.

Book design by Maggie Tang.

Special thanks to the following people for providing valuable feedback on early versions of this book: Savannah Agger, Gerhard Behles, Noel Bush, Michael Ford, Coleman Goughary, Ian Hobson, Alexander Joscht, Steffen Klein, Adam Mannegren, Takeshi Nishimoto, and Huston Singletary.

About the author

Dennis DeSantis is a composer, sound designer, percussionist, and the Head of Documentation for Ableton. His electronic music appears on labels such as Ghostly, Global Underground, Cocoon and Kanzleramt, and he has performed throughout North America, Europe, and Japan. Dennis has a Doctor of Musical Arts degree in Composition from the Eastman School of Music, and also has composition degrees from Yale University and Western Michigan University.

About Ableton

Ableton makes Live—a unique music creation and performance software—and Push, a hardware instrument for playing and composing with Live. Ableton was founded in 1999 and released the first version of Live in 2001. Since then, the company has received outstanding press, numerous awards and attracted a worldwide community of dedicated musicians, composers and DJs. The company headquarters are in Berlin, with an additional office in Los Angeles. Ableton is run by its original founders and currently has about 180 employees.

WILL PIRKLE

DESIGNING AUDIO EFFECT PLUG-INS IN C++

WITH DIGITAL AUDIO SIGNAL PROCESSING THEORY



Designing Audio Effect Plug-Ins in C++

Designing Audio Effect Plug-Ins in C++

With Digital Audio Signal Processing Theory

Will Pirkle

Focal Press
Taylor & Francis Group
NEW YORK AND LONDON

First published 2013
by Focal Press
70 Blanchard Road, Suite 402, Burlington, MA 01803

Simultaneously published in the UK
by Focal Press
2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN

Focal Press is an imprint of the Taylor and Francis Group, an Informa business

© 2013 Taylor and Francis

The right of Will Pirkle to be identified as author of this work has been asserted by him/her in accordance with sections 77 and 78 of the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this book may be reprinted or reproduced or utilised in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying and recording, or in any information storage or retrieval system, without permission in writing from the publishers.

Notices
Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data
Pirkle, William C., author.
Designing audio effect plug-ins in C++ with digital audio signal processing theory/Will Pirkle.
pages cm
Includes bibliographical references and index.
ISBN 978-0-240-82515-1 (paperback)
1. Computer sound processing. 2. Plug-ins (Computer programs)
3. C++ (Computer program language) I. Title.
MT723.P57 2013
006.4’5—dc23

2012024241

ISBN: 978-0-240-82515-1 (pbk)
ISBN: 978-0-123-97882-0 (ebk)

Typeset in Times
Project Managed and Typeset by: diacriTech

Dedicated to

my father and mother
C.H. Pirkle
and
J.V. Pirkle

Contents

Introduction

Chapter 1: Digital Audio Signal Processing Principles

1.1 Acquisition of Samples
1.2 Reconstruction of the Signal
1.3 Signal Processing Systems
1.4 Synchronization and Interrupts
1.5 Signal Processing Flow
1.6 Numerical Representation of Audio Data
1.7 Using Floating-Point Data
1.8 Basic DSP Test Signals
1.8.1 DC and Step
1.8.2 Nyquist
1.8.3 ½ Nyquist
1.8.4 ¼ Nyquist
1.8.5 Impulse
1.9 Signal Processing Algorithms
1.10 Bookkeeping
1.11 The One-Sample Delay
1.12 Multiplication
1.13 Addition and Subtraction
1.14 Algorithm Examples and the Difference Equation
1.15 Gain, Attenuation, and Phase Inversion
1.16 Practical Mixing Algorithm
Bibliography

Chapter 2: Anatomy of a Plug-In

2.1 Static and Dynamic Linking
2.2 Virtual Address Space and DLL Access
2.3 C and C++ Style DLLs
2.4 Maintaining the User Interface
2.5 The Applications Programming Interface

2.6 Typical Required API Functions
2.7 The RackAFX Philosophy and API
2.7.1 __stdcall
Bibliography

Chapter 3: Writing Plug-Ins with RackAFX
3.1 Building the DLL
3.2 Creation
3.3 The GUI
3.4 Processing Audio
3.5 Destruction
3.6 Your First Plug-Ins
3.6.1 Project: Yourplugin
3.6.2 Yourplugin GUI
3.6.3 Yourplugin.h File
3.6.4 Yourplugin.cpp File
3.6.5 Building and Testing
3.6.6 Creating and Saving Presets
3.6.7 GUI Designer
3.7 Design a Volume Control Plug-In
3.8 Set Up RackAFX for Use
3.9 Setup Preferences
3.9.1 Project: Volume
3.9.2 Volume GUI
3.9.3 Configure a Slider Control
3.9.4 Volume.h File
3.9.5 Volume.cpp File
3.10 Design a Volume-in-dB Plug-In
3.10.1 Project: VolumedB
3.10.2 VolumedB GUI
3.10.3 VolumedB.h File
3.10.4 VolumedB.cpp File
3.11 Design a High-Frequency Tone Control Plug-In
3.11.1 Project: SimpleHPF
3.11.2 SimpleHPF GUI
3.11.3 SimpleHPF.h File

3.11.4 SimpleHPF.cpp File
3.12 Design a High-Frequency Tone Control with Volume Plug-In
3.12.1 Project: SimpleHPF
3.12.2 SimpleHPF GUI
3.12.3 SimpleHPF.h File
3.12.4 SimpleHPF.cpp File
3.13 The User Plug-In Menu in RackAFX

Chapter 4: How DSP Filters Work
4.1 First-Order Feed-Forward Filter
4.2 Design a General First-Order Feed-Forward Filter
4.3 First-Order Feed-Back Filter
4.4 Design a General First-Order Feed-Back Filter
4.4.1 Project FeedBackFilter
4.4.2 FeedBackFilter GUI
4.4.3 FeedBackFilter.h File
4.4.4 FeedBackFilter.cpp File
4.5 Observations
4.5.1 General
4.5.2 Feed-Forward Filters
4.5.3 Feed-Back Filters
Bibliography

Chapter 5: Basic DSP Theory
5.1 The Complex Sinusoid
5.2 Complex Math Review
5.3 Time Delay as a Math Operator
5.4 First-Order Feed-Forward Filter Revisited
5.4.1 Negative Frequencies
5.4.2 Frequencies Above and Below ±Nyquist
5.5 Evaluating the Transfer Function H(ω)
5.5.1 DC (0 Hz)
5.5.2 Nyquist (π)
5.5.3 ½ Nyquist ($\pi/2$)
5.5.4 ¼ Nyquist ($\pi/4$)
5.6 Evaluating $e^{j\omega}$
5.7 The z Substitution

5.8 The *z* Transform
5.9 The *z* Transform of Signals
5.10 The *z* Transform of Difference Equations
5.11 The *z* Transform of an Impulse Response
5.12 The *Zeros* of the Transfer Function
5.13 Estimating the Frequency Response: *Zeros*
5.14 Filter Gain Control
5.15 First-Order Feed-Back Filter Revisited
5.16 The *Poles* of the Transfer Function
5.16.1 DC (0 Hz)
5.16.2 Nyquist (π)
5.16.3 ½ Nyquist (π/2)
5.16.4 ¼ Nyquist (π/4)
5.17 Second-Order Feed-Forward Filter
5.17.1 DC (0 Hz)
5.17.2 Nyquist (π)
5.17.3 ½ Nyquist (π/2)
5.17.4 ¼ Nyquist (π/4)
5.18 Second-Order Feed-Back Filter
5.18.1 DC (0 Hz)
5.18.2 Challenge
5.19 First-Order Pole-Zero Filter: The Shelving Filter
5.19.1 DC (0 Hz)
5.19.2 Challenge
5.20 The Bi-Quadratic Filter
Bibliography

Chapter 6: Audio Filter Designs: IIR Filters
6.1 Direct *z*-Plane Design
6.2 Single Pole Filters
6.2.1 First-Order LPF and HPF
6.3 Resonators
6.3.1 Simple Resonator
6.3.2 Smith-Angell Improved Resonator
6.4 Analog Filter to Digital Filter Conversion
6.4.1 Challenge

6.5 Effect of Poles or Zeros at Infinity
6.6 Generic Bi-Quad Designs
6.6.1 First-Order LPF and HPF
6.6.2 Second-Order LPF and HPF
6.6.3 Second-Order BPF and BSF
6.6.4 Second-Order Butterworth LPF and HPF
6.6.5 Second-Order Butterworth BPF and BSF
6.6.6 Second-Order Linkwitz-Riley LPF and HPF
6.6.7 First- and Second-Order APF
6.7 Audio Specific Filters
6.7.1 Modified Bi-Quad
6.7.2 First-Order Shelving Filters
6.7.3 Second-Order Parametric/Peaking Filter: Non-Constant-Q
6.7.4 Second-Order Parametric/Peaking Filter: Constant-Q
6.7.5 Cascaded Graphic EQ: Non-Constant-Q
6.7.6 Cascaded Graphic EQ: Constant-Q
6.8 Design a Resonant LPF Plug-In
6.8.1 Project: ResonantLPF
6.8.2 ResonantLPF GUI
6.8.3 ResonantLPF.h File
6.8.4 ResonantLPF.cpp File
6.9 The Massberg Analog-Matched Low-Pass Filter
6.9.1 First-Order Massberg LPF
6.9.2 Second-Order Massberg LPF
Bibliography
References

Chapter 7: Delay Effects and Circular Buffers
7.1 The Basic Digital Delay
7.2 Digital Delay with Wet/Dry Mix
7.2.1 Frequency and Impulse Responses
7.2.2 The Effect of Feedback
7.3 Design a DDL Module Plug-In
7.3.1 Project: DDLModule
7.3.2 DDLModule GUI
7.3.3 DDLModule.h File

7.3.4 DDLModule.cpp File
7.3.5 Declare and Initialize the Delay Line Components
7.3.6 DDLModule.h File
7.3.7 DDLModule.cpp File
7.4 Modifying the Module to Be Used by a Parent Plug-In
7.4.1 DDLModule.h File
7.4.2 DDLModule.cpp File
7.5 Modifying the Module to Implement Fractional Delay
7.5.1 DDLModule.cpp File
7.6 Design a Stereo Digital Delay Plug-In
7.6.1 Project: StereoDelay
7.6.2 StereoDelay GUI
7.6.3 StereoDelay.h File
7.6.4 StereoDelay.cpp File
7.7 Design a Stereo Crossed-Feedback Delay Plug-In
7.8 Enumerated Slider Variables
7.8.1 Constructor
7.8.2 PrepareForPlay()
7.8.3 UserInterfaceChange()
7.8.4 ProcessAudioFrame()
7.9 More Delay Algorithms
7.9.1 Advanced DDL Module
7.9.2 Delay with LPF in Feedback Loop
7.9.3 Multi-Tap Delay
7.9.4 Ping-Pong Delay
7.9.5 LCR Delay
Bibliography

Chapter 8: Audio Filter Designs: FIR Filters
8.1 The IR Revisited: Convolution
8.2 Using RackAFX's Impulse Convolver
8.2.1 Loading IR Files
8.2.2 Creating IR Files
8.2.3 The IR File Format
8.3 Using RackAFX's FIR Designer
8.4 The Frequency Sampling Method

8.4.1 Linear-Phase FIR Using the Frequency Sampling Method
8.5 Complementary Filter Design for Linear Phase FIR Filters
8.6 Using RackAFX's Frequency Sampling Method Tool
8.7 Designing a Complementary Filter
8.8 The Optimal (Parks-McClellan) Method
8.9 Using RackAFX's Optimal Method Tool
8.10 Design a Convolution Plug-In
8.10.1 Project: Convolver
8.10.2 Convolver.h File
8.10.3 Convolver.cpp File
8.11 Numerical Method FIR Filters
8.11.1 Moving Average Interpolator
8.11.2 Lagrange Interpolator
8.11.3 Median Filter
Bibliography

Chapter 9: Oscillators

9.1 Direct Form Oscillator
9.1.1 Initial Conditions
9.2 Design a Direct Form Oscillator Plug-In
9.2.1 Project: DirectOscillator
9.2.2 DirectOscillator GUI
9.2.3 DirectOscillator.h File
9.2.4 DirectOscillator.cpp File
9.2.5 Improving the Oscillator Design
9.3 The Gordon-Smith Oscillator
9.4 Wave Table Oscillators
9.5 Design a Wave Table Oscillator Plug-In
9.5.1 Project: WTOscillator
9.5.2 WTOscillator GUI
9.5.3 WTOscillator.h File
9.5.4 WTOscillator.cpp File
9.6 Adding More Wave Tables
9.6.1 WTOscillator.h File
9.6.2 WTOscillator.cpp File
9.6.3 WTOscillator GUI

9.6.4 WTOscillator.h File
9.6.5 WTOscillator.cpp File
9.7 Band-Limited Additive Wave Tables
9.7.1 WTOscillator GUI
9.7.2 WTOscillator.h File
9.7.3 WTOscillator.cpp File
9.7.4 Saw-Tooth
9.7.5 Square Wave
9.8 Additional Oscillator Features (LFO)
9.8.1 WTOscillator.h File
9.8.2 WTOscillator.cpp File
9.8.3 WTOscillator.h File
9.8.4 WTOscillator.cpp File
9.9 Bipolar/Unipolar Functionality
9.9.1 WTOscillator GUI
9.9.2 WTOscillator.cpp File
Bibliography

Chapter 10: Modulated Delay Effects
10.1 The Flanger/Vibrato Effect
10.2 The Chorus Effect
10.3 Design a Flanger/Vibrato/Chorus Plug-In
10.3.1 Project: ModDelayModule
10.3.2 ModDelayModule GUI
10.3.3 ModDelayModule.h File
10.3.4 ModDelayModule.cpp File
10.3.5 PrepareForPlay()
10.3.6 Challenge
10.4 Design a Stereo Quadrature Flanger Plug-In
10.4.1 Project: StereoQuadFlanger
10.4.2 StereoQuadFlanger GUI
10.4.3 StereoQuadFlanger.h File
10.4.4 StereoQuadFlanger.cpp File
10.4.5 Challenges
10.5 Design a Multi-Unit LCR Chorus Plug-In
10.5.1 Project: StereoLCRChorus

10.5.2 StereoLCRChorus GUI
10.5.3 StereoLCRChorus.h File
10.5.4 StereoLCRChorus.cpp File
10.6 More Modulated Delay Algorithms
10.6.1 Stereo Cross-Flanger/Chorus (Korg Triton®)
10.6.2 Multi-Flanger (Sony DPS-M7®)
10.6.3 Bass Chorus
10.6.4 Dimension-Style (Roland Dimension D®)
10.6.5 Deca-Chorus (Sony DPS-M7®)
Bibliography

Chapter 11: Reverb Algorithms

11.1 Anatomy of a Room Impulse Response
11.1.1 RT_{60}: The Reverb Time
11.2 Echoes and Modes
11.3 The Comb Filter Reverberator
11.4 The Delaying All-Pass Filter Reverberator
11.5 More Delaying All-Pass Filter Reverberators
11.6 Schroeder's Reverberator
11.7 The Low-Pass Filter–Comb Reverberator
11.8 Moorer's Reverberator
11.9 Stereo Reverberation
11.10 Gardner's Nested APF Reverberators
11.11 Modulated APF and Comb/APF Reverb
11.12 Dattorro's Plate Reverb
11.13 Generalized Feedback Delay Network Reverbs
11.14 Other FDN Reverbs
11.15 An Example Room Reverb
11.16 RackAFX Stock Objects
11.16.1 COnePoleLPF
11.16.2 CDelay
11.16.3 CCombFilter
11.16.4 CLPFCombFilter
11.16.5 CDelayAPF
11.17 Design the Room Reverb
11.17.1 Project: Reverb

11.17.2 Reverb GUI
11.17.3 Reverb.h
11.17.4 Reverb.cpp
11.18 Challenge
Bibliography
References

Chapter 12: Modulated Filter Effects

12.1 Design a Mod Filter Plug-In: Part I Modulated f_c
12.1.1 Project: ModFilter
12.1.2 ModFilter GUI
12.1.3 ModFilter.h File
12.1.4 ModFilter.cpp File
12.2 Design a Mod Filter Plug-In: Part II, Modulated f_c, Q
12.2.1 ModFilter GUI
12.2.2 ModFilter.h File
12.2.3 ModFilter.cpp File
12.3 Design a Mod Filter Plug-In: Part III, Quad-Phase LFOs
12.3.1 ModFilter GUI
12.3.2 ModFilter.cpp File
12.4 Design an Envelope Follower Plug-In
12.5 Envelope Detection
12.5.1 Project EnvelopeFollower
12.5.2 EnvelopeFollower GUI
12.5.3 EnvelopeFollower.h File
12.5.4 EnvelopeFollower.cpp File
12.6 Design a Phaser Plug-In
12.6.1 Project Phaser
12.6.2 Phaser GUI
12.6.3 Phaser.h File
12.6.4 Phaser.cpp File
12.7 Design a Stereo Phaser with Quad-Phase LFOs
12.7.1 Phaser GUI
12.7.2 Phaser.h File
12.7.3 Phaser.cpp File
Bibliography

References

Chapter 13: Dynamics Processing

13.1 Design a Compressor/Limiter Plug-In
13.1.1 Project: DynamicsProcessor
13.1.2 DynamicsProcessor: GUI
13.1.3 DynamicsProcessor.h File
13.1.4 DynamicsProcessor.cpp File
13.1.5 DynamicsProcessor.cpp File
13.2 Design a Downward Expander/Gate Plug-In
13.2.1 DynamicsProcessor.h File
13.2.2 DynamicsProcessor.cpp File
13.3 Design a Look-Ahead Compressor Plug-In
13.3.1 DynamicsProcessor: GUI
13.3.2 DynamicsProcessor.h File
13.3.3 DynamicsProcessor.cpp File
13.4 Stereo-Linking the Dynamics Processor
13.4.1 DynamicsProcessor: GUI
13.4.2 DynamicsProcessor.cpp File
13.5 Design a Spectral Compressor/Expander Plug-In
13.5.1 Project: SpectralDynamics
13.5.2 SpectralDynamics: GUI
13.5.3 Additional Slider Controls
13.5.4 Spectral Dynamics Buttons
13.5.5 Spectral Dynamics Metering
13.5.6 SpectralDynamics.h File
13.5.7 SpectralDynamics.cpp File
13.6 Alternate Side-Chain Configurations
Bibliography
References

Chapter 14: Miscellaneous Plug-Ins

14.1 Design a Tremolo/Panning Plug-In
14.1.1 Project: TremoloPanner
14.1.2 TremoloPanner: GUI
14.2 Design a Ring Modulator Plug-In
14.2.1 Project: RingModulator

14.2.2 RingModulator: GUI
14.2.3 RingModulator.h File
14.2.4 RingModulator.cpp File
14.3 Design a Wave Shaper Plug-In
14.3.1 Project: WaveShaper
14.3.2 WaveShaper: GUI
Bibliography

Appendix A: The VST® and AU® Plug-In APIs
A.1 Compiling as a VST Plug-In in Windows
A.2 Wrapping Your RackAFX Plug-In
A.3 Comparison of Objects/Methods/GUIs
A.4 VST Plug-In without Wrapping
A.4.1 Default GUI
A.4.2 Signal Processing
A.5 VST Plug-In with RackAFX Wrapping
A.5.1 Default GUI
A.6 AU Overview
A.6.1 Default GUI
A.6.2 Signal Processing

Appendix B: More RackAFX Controls and GUI Designer
B.1 The Alpha Wheel and LCD Control
B.2 The Vector Joystick Control
B.3 Using the sendUpdateGUI() Method
B.4 Using GUI Designer

Index

Introduction

When I started teaching in the Music Engineering Technology Program at the University of Miami in 1996, we were writing signal processing algorithms in digital signal processing (DSP) assembly language and loading them on to DSP evaluation boards for testing. We had also just begun teaching a software plug-in programming class, since computers were finally at the point where native processing was feasible. I began teaching Microsoft's DirectX® in 1997 and immediately began a book/manual on converting DSP algorithms into C++ code for the DirectX platform. A year later I had my first manuscript of what would be a thick DirectX programming book. However, I had two problems on my hands: first, DirectX is bloated with Common Object Model (COM) programming, and it seemed like the lion's share of the book was really devoted to teaching basic COM skills rather than converting audio signal processing algorithms into C++, creating a graphical user interface (GUI), and handling user input. More importantly, developers had dropped DirectX in favor of a new, lean, cross-platform compatible plug-in format called Steinberg VST®, written in "straight" C++ without the need for operating system (OS) specific components. After taking one look at the Virtual Studio Technology (VST) applications programming interface (API), I immediately dropped all DirectX development, wrote VST plug-ins all summer, and then switched to teaching it the following year. And, I put my now-obsolete manuscript on the shelf.

After half a dozen semesters teaching VST programming, a few things had become clear. For any given project, the students were spending more time dealing with setting up and maintaining a GUI than they were with the audio signal processing code. Instead of being on the Internet to look for cool processing algorithms, they were searching for knob and switch bitmaps. While I can certainly appreciate a nice-looking GUI, I was trying to teach audio signal processing and not graphic design. I next spent some time trying to come up with some kind of way to speed up the GUI programming, and I wrote a helper program that let students define lists of variable names, data types, minimum, maximum, and initial values. Then it would synthesize the code for handling part of the GUI interaction. The actual GUI itself was another problem, and I spent years trying to come up with a way to free the students (and myself) from dealing with the hassles of designing and maintaining a GUI. Around 2004, as a result of a consulting job, I needed to come up with a simple C++ based audio signal processing API—my own plug-in format. I would also need to write my own plug-in client, the software that would load and process data through my plug-ins. I was determined to write an API that was far simpler and leaner than even VST. And, I wanted my client software to handle 100% of the GUI details so I could focus on the signal processing work for my consulting client. This would also make great software to teach audio signal processing and plug-in programming simultaneously.

Since 2009 I've been using my RackAFX™ software in the classroom at both the graduate and undergraduate levels. My students never cease to amaze me with what they design. Currently, RackAFX runs on the Windows® OS and generates plug-ins that are compatible with it and optionally VST as well. You can develop your plug-in in RackAFX, then use it in any Windows VST client.

RackAFX runs in tandem with Microsoft Visual Studio compilers, even the free ones. It sets up your project files, writes GUI code for you, and allows you to remotely launch some compiler functions from its control surface. Once you understand how the RackAFX API works, learning other commercial plug-in APIs will be much easier because most plug-in APIs share similar methods and set-up sequences. And, since RackAFX plug-ins are written in C++, the plug-in objects can be compiled on different platforms and embedded (or wrapped) to operate in just about any signal processing environment. We are currently running RackAFX plug-ins in Apple's iOS® in one of my other classes.

In the fall of 2010 I began writing and handing out chapters, one at a time, to students in a graduate-level class. The chapters combined DSP theory, audio algorithms, and programming in the proportions I had always wanted, including difference equations and derivations that were fully worked out. The RackAFX API requires zero GUI programming. It isn't tied to a licensed product that I have to worry will become obsolete or will change in complexity or underlying design. Most importantly, if you can learn to write RackAFX plug-ins, you can learn to write for just about any other API on the platform you choose. See Appendix A for more information.

Those initial grad students helped shape the direction and flow of the book (perhaps without knowing it). I wanted the book to be aimed at people with programming skills who wanted to get into audio signal processing or the other way around. Academic types that are tired of using mathematical software to try to do real-time signal processing should also benefit. The API does not require a steep learning curve.

Chapter 1 presents the number systems, sample indexing, and basic block diagram algorithm components. Chapter 2 explains how plug-ins are packaged in software and Chapter 3 immediately starts you writing plug-ins. Chapters 4 and 5 are the basic DSP theory chapters. Feel free to skip these chapters if you already have a strong DSP background or if you don't care about it. Chapter 6 is a cookbook of infinite impulse response (IIR) and audio specific filter designs. Chapter 7 introduces delay lines and delay effects. The circular buffering they require is necessary for Chapter 8's finite impulse response (FIR) tools and convolution plug-in.

Chapter 9 introduces oscillator design, which is needed in Chapter 10's modulated delay effects: flanger, chorus and vibrato effects. Chapter 11 includes the analysis of a collection of reverberator modules and an example reverb design. Chapter 12 returns to the modulation theme with modulated filter effects, including my own phaser, while Chapter 13 introduces dynamics processor design. Chapter 14 is a clean-up chapter of miscellaneous effects algorithms. The two appendices can be read any time after Chapter 3, where you design your first plug-ins. Appendix A deals with the VST and Audio Units (AU)® formats, their similarities and differences, and how to use RackAFX objects inside them. Appendix B shows how to use the GUI designer to drag and drop GUI elements on your control surface—0% GUI coding, guaranteed. And, if you have the tools and skills to make and maintain your own GUI, there are options for that as well.

My colleagues Ken Pohlmann, Joe Abbati, Colby Leider, Chris Bennett, and Rey Sanchez had nothing but encouragement and enthusiasm for the book/software effort. Like many of the students, they watched RackAFX evolve—even change names—from a fairly bland, gray academic interface into a

highly customizable personal plug-in development lab with a drag-and-drop GUI designer. Of course those features and enhancements came from suggestions, comments, and critiques from the numerous students who used the software in class. The API would not be as robust today without their invaluable input and feedback. Stefan Sullivan tirelessly beta-tested the software, even in the middle of his thesis work; his comments led to many of the features in the current product. The RackAFX power users (students who went above and beyond the parameters of their projects and pushed the envelope of the API) include Mike Chemistruck, Greg Dion, Felipe Espic, Chen Zhang, Tim Schweichler, Phil Minnick, Sherif Ahmed, Scott Dickey, Matan Ben-Asher, Jay Coggin, Michael Everman, and Jordan Whitney.

I hope you enjoy learning the audio effects theory and plug-in design from the book and that each chapter opens up a new treasure box of cool audio gems you can use to make your own truly unique audio effects in your own audio effect laboratory. The software FAQs, sample code, and tutorials are all available at the website www.willpirkle.com and you are encouraged to upload your own plug-ins and videos as well. I can't wait to hear what you cook up in your lab!

Will Pirkle
June 1, 2012

CHAPTER 1

Digital Audio Signal Processing Principles

The first affordable digital audio devices began appearing in the mid-1980s. Digital signal processing (DSP) mathematics had been around since the 1960s and commercial digital recordings first appeared in the early 1970s, but the technology did not become available for widespread distribution until about 15 years later when the advent of the compact disc (CD) ushered in the age of digital audio. Digital sampling refers to the acquisition of data points from a continuous analog signal. The data points are sampled on a regular interval known as the sample period or sample interval. The inverse of the sample period is the sampling frequency. A compact disc uses a sampling frequency of 44,100 Hz, producing 44,100 discrete samples per channel each second, with a sample interval of about 22.7 microseconds (μS). While digital sampling applies to many different systems, this book is focused on only one of those applications: audio.

During the course of this book, you will learn both DSP theory and applications. This is accomplished by experimenting with your own DSP algorithms at the same time as you learn the theory using the included software, RackAFX™. The goal is to understand how the DSP algorithms translate into C++ code. The resulting plug-ins you write can be used to process audio in real time. Because plug-ins are software variations on hardware designs, it's worth examining how the hardware systems operate, the typical audio coding formats, and the algorithm building blocks you will be using to implement your own software versions.

1.1 Acquisition of Samples

The incoming analog audio signal is sampled with an analog-to-digital converter (ADC or A/D). A/Ds must accurately sample and convert an incoming signal, producing a valid digitally coded number that represents the analog voltage within the sampling interval. This means that for CD audio, a converter must produce an output every 22.7 μS. Figure 1.1 shows the block diagram of the input conversion system with LFP, A/D, and encoder.

Violating the Nyquist criteria will create audible errors in the signal in the form of an erroneously encoded signal. Frequencies higher than Nyquist will fold back into the spectrum. This effect is called aliasing because the higher-than-Nyquist frequencies are encoded “in disguise” or as an “alias” of the actual frequency. This is easiest explained with a picture of an aliased signal, shown in Figure 1.2.

The *sampling theorem* states that a continuous analog signal can be sampled into discrete data points and then reconstructed into the original analog signal without any loss of information—including inter-sample fluctuations—if and only if the input signal has been band-limited so that it contains no frequencies higher than one-half the sample rate, also known as the *Nyquist frequency* or *Nyquist rate*. Band-limited means low-pass filtered (LPF). Band-limiting the input signal prior to sampling is also known as adhering to the *Nyquist criteria*.



Figure 1.1: The input conversion system ultimately results in numerical coding of the band-limited input signal.

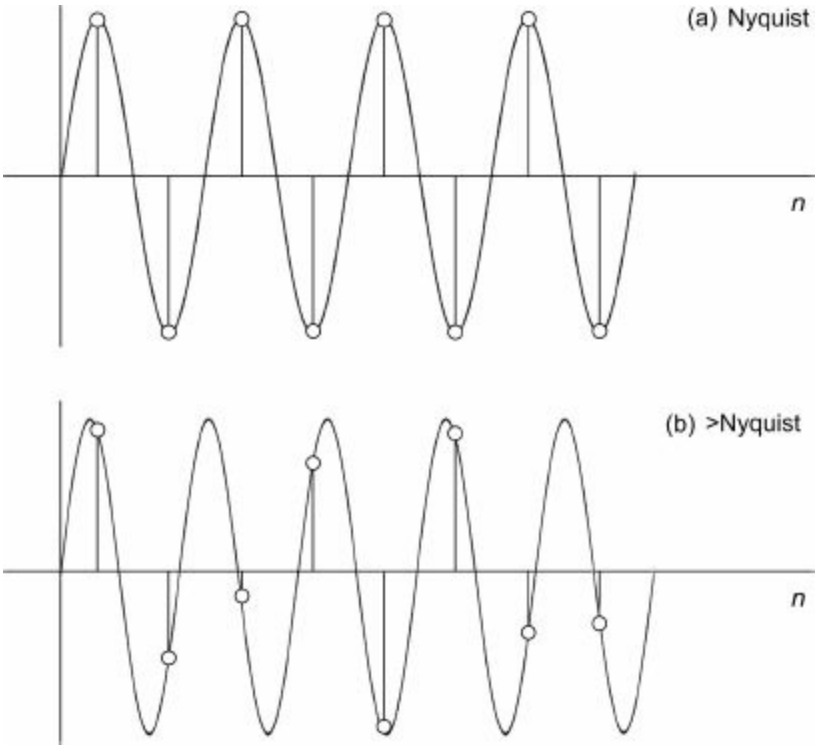


Figure 1.2: (a) The Nyquist frequency is the highest frequency that can be encoded with two samples per period. (b) Increasing the frequency above Nyquist but keeping the sampling interval the same results in an obvious coding error—the aliased signal is the result.

Once the aliased signal is created, it can never be removed and will remain as a permanent error in the signal. The LPF that band-limits the signal at the input is called the *anti-aliasing filter*. Another form of aliasing occurs in the movies. An analog movie camera takes 30 pictures (frames) per second. However, it must often film objects that are rotating at much higher rates than 30 per second, like helicopter blades or car wheels. The result is visually confusing—the helicopter blades or car wheels appear to slow down and stop, then reverse directions and speed up, then slow down and stop, reverse, and so on. This is the visual result of the high-frequency rotation aliasing back into the movie as an erroneous encoding of the actual event.

1.2 Reconstruction of the Signal

The digital-to-analog converter (DAC or D/A) first decodes the bit-stream, then takes the sampled data points or impulses and converts them into analog versions of those impulses. The D/A output is then low-pass filtered to produce the final analog output signal, complete with all the inter-sample fluctuations. As with the A/D diagram, the decoder and D/A are both inside the same device (a chip). Figure 1.3 shows the conceptual block diagram of the decoding end of the system.

The output filter is called the *reconstruction filter* and is responsible for re-creating the seemingly missing information that the original sampling operation did not acquire—all the inter-sample fluctuations. The reason it works is that low-pass filtering converts impulses into smeared out, wiggly versions. The smeared out versions have the shape $f(x) = \sin(x)/x$ which is also called the sinc() function and somewhat resembles the profile of a sombrero, as shown in Figure 1.4.

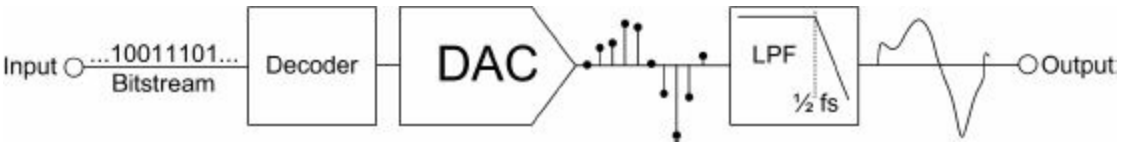


Figure 1.3: The digital bit-stream is decoded and converted into an analog output using a low-pass filter to reconstruct the analog signal.

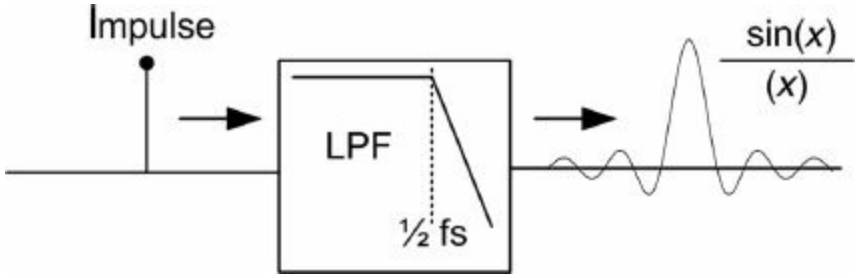


Figure 1.4: The ideal reconstruction filter creates a smeared output with a damped oscillatory shape. The amplitude of the sin(*x*)/*x* shape is proportional to the amplitude of the input pulse.

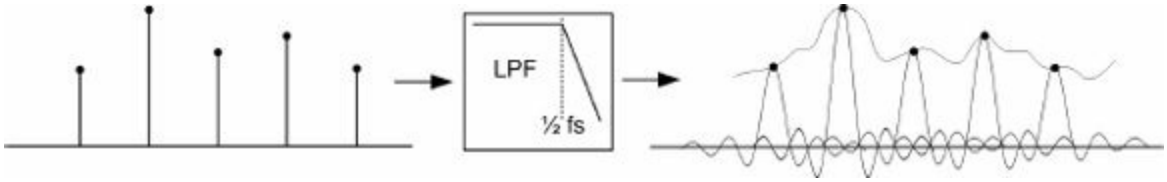


Figure 1.5: The sin(*x*)/*x* outputs of the LPF are summed together to reconstruct the original band-limited input waveform; the inter-sample information has been reconstructed.

When a *series* of impulses is filtered, the resulting set of sin(*x*)/*x* pulses overlap with each other and their responses all add up linearly. The addition of all the smaller curves and damped oscillations reconstructs the inter-sample curves and fluctuations (Figure 1.5).

1.3 Signal Processing Systems

Signal processing systems combine data acquisition devices with microprocessors to run mathematical algorithms on the audio data. Those algorithms are the focus of this book. Today's plug-ins are descendents of stand-alone hardware effects processors which are based on DSP chips. A DSP chip is a highly specialized processor designed mainly to run DSP algorithms. A DSP would function poorly as a central processing unit (CPU) for a personal computer because it only has a small but highly specialized instruction set. DSP devices (or just "DSPs") feature a core processor designed to multiply and accumulate data because this operation is fundamental to DSP algorithms. Because this process is repeated over and over, modern DSPs use pipelining to fetch the next data while simultaneously processing the current data. This technique greatly improves the efficiency of

the system. A typical signal processing system consists of the following components (Figure 1.6):

- Data acquisition devices (A/D and D/A)
- A DSP chip
- Memory to store the algorithm program (program memory)
- Memory to store the algorithm data (data memory)
- A user interface (UI) with buttons, knobs, switches, and graphic displays

Music synthesis devices are an embedded version of a CPU+DSP arrangement. The microcontroller or microprocessor manages the UI and keyboard, while the DSP performs the audio synthesis and effects algorithms. Practical implementations now involve multiple DSPs, some of which are custom designed specifically for sound synthesis. An example block diagram is shown in Figure 1.7.

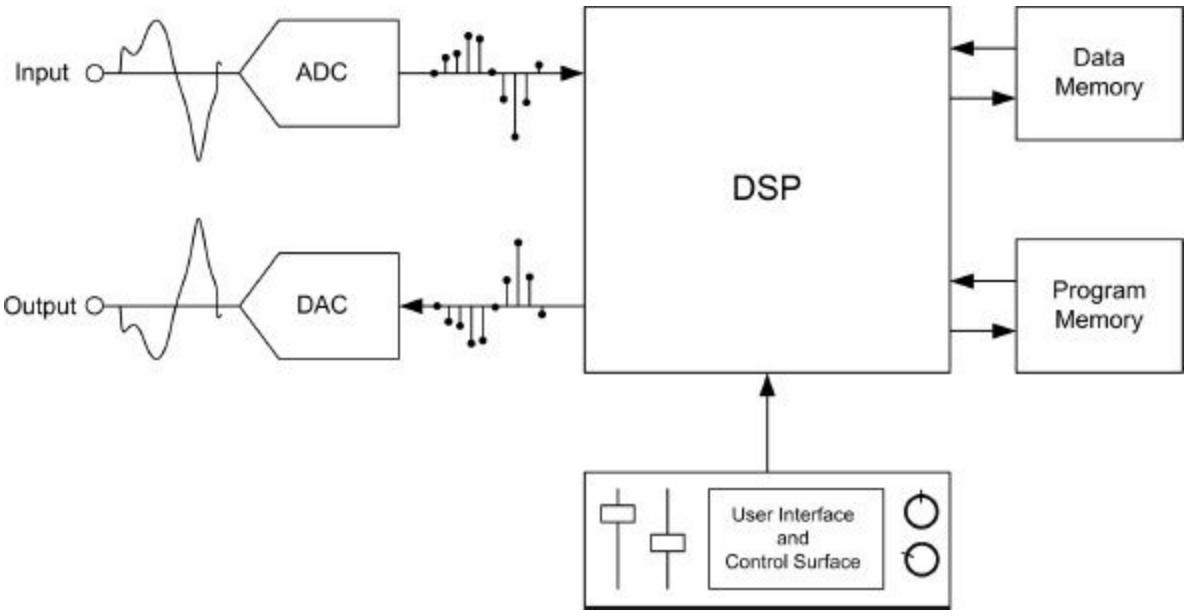


Figure 1.6: A simple signal processing system. The algorithm in this case is inverting the phase of the signal; the output is upside-down.

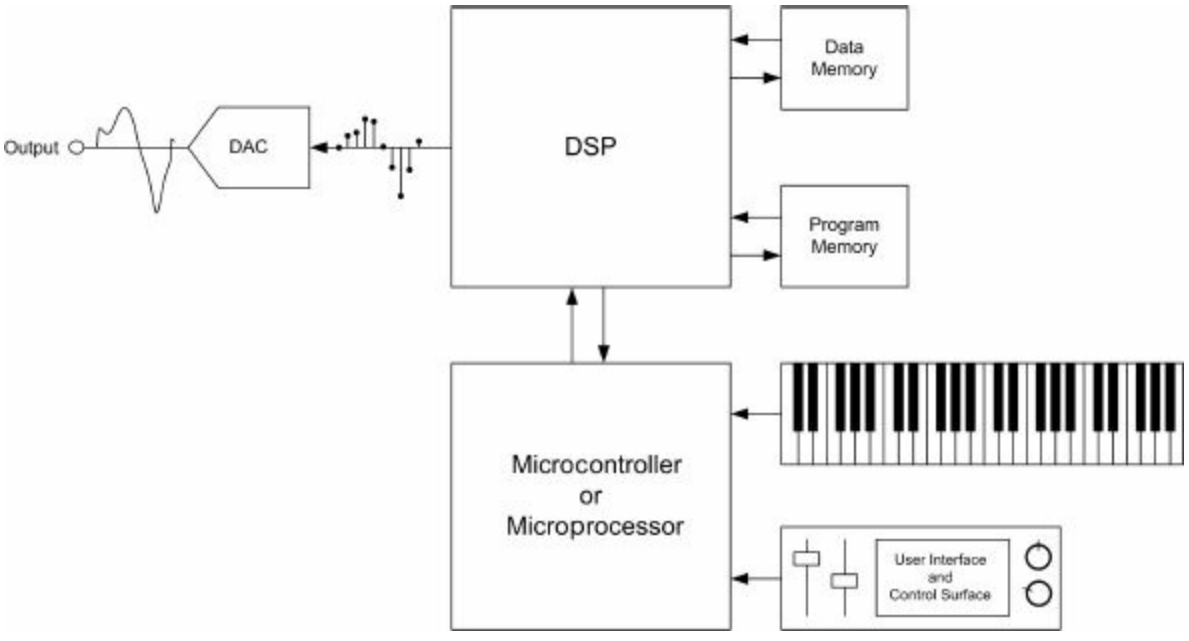


Figure 1.7: Block diagram of a synthesizer.

1.4 Synchronization and Interrupts

There are two fundamental modes of operation when dealing with incoming and outgoing audio data: synchronous and asynchronous. In synchronous operation, all audio input and output data words are synchronized to the same clock as the DSP. These are typically simple systems with a minimum of input/output (I/O) and peripherals. More-complex systems usually involve asynchronous operation, where the audio data is not synchronized to the DSP. Moreover, the audio itself might not be synchronous, that is, the input and output bit-streams might not operate on the same clock. A purely synchronous system will be more foolproof, but less flexible.

An asynchronous system will almost always be interrupt-based. In an interrupt-based design, the processor enters a wait-loop until a processor interrupt is toggled. The processor interrupt is just like a doorbell. When another device such as the A/D has data ready to deliver, it places the data in a predesignated buffer, and then it rings the doorbell by toggling an interrupt pin. The processor services the interrupt with a function that picks up the data, and then goes on with its processing code. The function is known as an *interrupt-service routine* or an *interrupt handler*. The interrupt-based system is the most efficient use of processor time because the processor can't predict the exact clock cycle when the data will be ready at the input.

Another source of interrupts is the UI. Each time the user changes a control, clicks a button, or turns a knob, the updated UI control information needs to be sent to the DSP so it can alter its processing to accommodate the new settings. This is also accomplished via interrupts and interrupt handlers. The interrupt is sometimes labeled INT, INTR, or IRQ (interrupt request line) in block diagrams or schematics.

1.5 Signal Processing Flow

Whether the processing is taking place on a DSP chip or in a plug-in, the overall processing flow, also known as the signal processing loop, remains basically the same. For a DSP chip, the processes in the loop will be encoded in the program memory's set of instructions, often in a nearly pure-inline fashion for optimization of the processor's cycles. For a plug-in, each processing block more or less takes on the form of a function in code, allowing for maximum flexibility.

The processing loop in Figure 1.8 consists of

- A one-time initialization function to set up the initial state of the processor and prepare for the arrival of data interrupts
- An infinite wait-loop, which does nothing but wait for an interrupt to occur
- An interrupt handler which decodes the interrupt and decides how to process—or ignore—the doorbell
- Data reading and writing functions for both control and data information
- A processing function to manipulate and create the audio output
- A function to set up the variables for the next time around through the loop, altering variables if the UI control changes warrant it

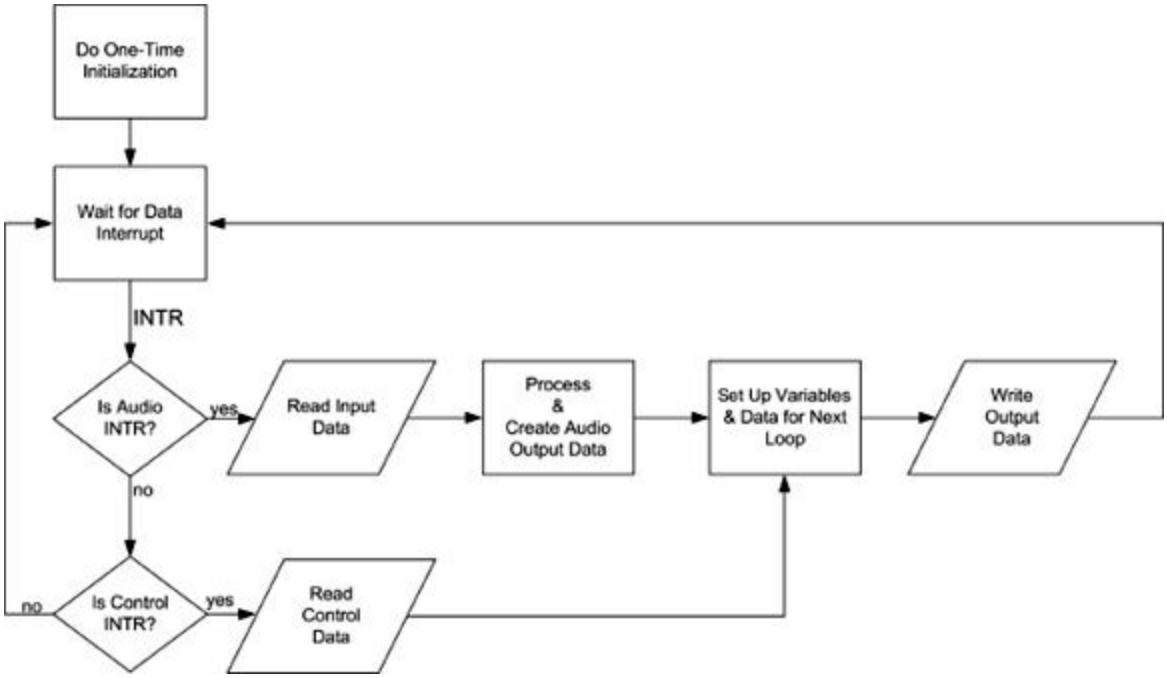


Figure 1.8: The flowchart for an audio signal processing system.

For the sampling theorem to hold true, the audio data must be arriving and leaving on a strictly timed interval, although it may be asynchronous with the DSP's internal clock. This means that when the DSP does receive an audio INTR it must do all of the audio processing and handle any UI interrupts before the next audio INTR arrives in one sample interval. The interrupt handling scheme is prioritized such that the audio data interrupt is the most important. Thus, while servicing the audio data interrupt, the DSP might be set up to ignore all other interrupts (except the reset interrupt) until the audio data is finished processing.

1.6 Numerical Representation of Audio Data

The audio data can be numerically coded in several different ways. Basic digital audio theory reveals that the number of quantization levels available for coding is found by Equation 1.1.

$$q = 2^N \tag{1.1}$$

where N is the bit depth of the system.

Thus, an 8-bit system can encode 2^8 values or 256 quantization levels. A 16-bit system can encode 65,536 different values. Figure 1.9 shows the hierarchy of encoded audio data. As a system designer, you must first decide if you are going to deal with unipolar (unsigned) or bipolar (signed) data. After that, you need to decide on the data types.

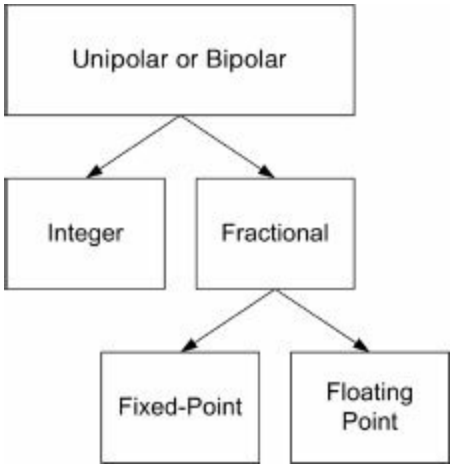


Figure 1.9: The hierarchy of numerical coding.

- Unipolar or *unsigned* data is in the range of 0 to +Max, or −Min to 0, and only has one polarity (+ or −) of data, plus the number zero (0).
- Bipolar or *signed* data varies from −Min to +Max and is the most common form today. It also includes the number zero (0).
- Integer data is represented with integers and no decimal place. Unsigned integer audio varies from 0 to +65,535 for 16-bit systems. Signed integer audio varies from −32,768 to +32,767 for the same 16-bit audio. In both cases, there are 65,536 quantization levels.
- Fractional data is encoded with an integer and fractional portion, combined as int.frac with the decimal place separating the parts (e.g. 12.09).

Within the fractional data subset are fixed- and floating-point types. Fixed-point data fixes the number of significant digits on each side of the decimal point and is combined as int-sig-digits.frac-sig-digits. "8.16" data would have 8 significant digits before the decimal place and 16 afterwards. Floating-point data has a moving mantissa and exponent which code the values across a range predetermined by the Institute of Electrical and Electronics Engineers (IEEE). The positive and negative portions are encoded in 2's complement so that the addition of exactly opposite values (e.g., −0.5 and +0.5) always results in zero. Figure 1.10 reveals how the quantization levels are encoded. The dashed lines represent the minimum and maximum values.

A fundamental problem is that the number of quantization levels will always be an even number since 2^N is always even. In bipolar systems, you would very much like to have the number zero (0) encoded as the number 0. If you do that, then you use up one of your quantization levels for it. This leaves an odd number of levels which cannot be split exactly in half. That creates the anomaly you see in Figure 1.10—there are always more negative (−) quantization levels than positive ones for bipolar coding. For the unipolar case, there is no value which exactly codes the zero or midpoint level; in Figure 1.10 it is midway between 127 and 128.

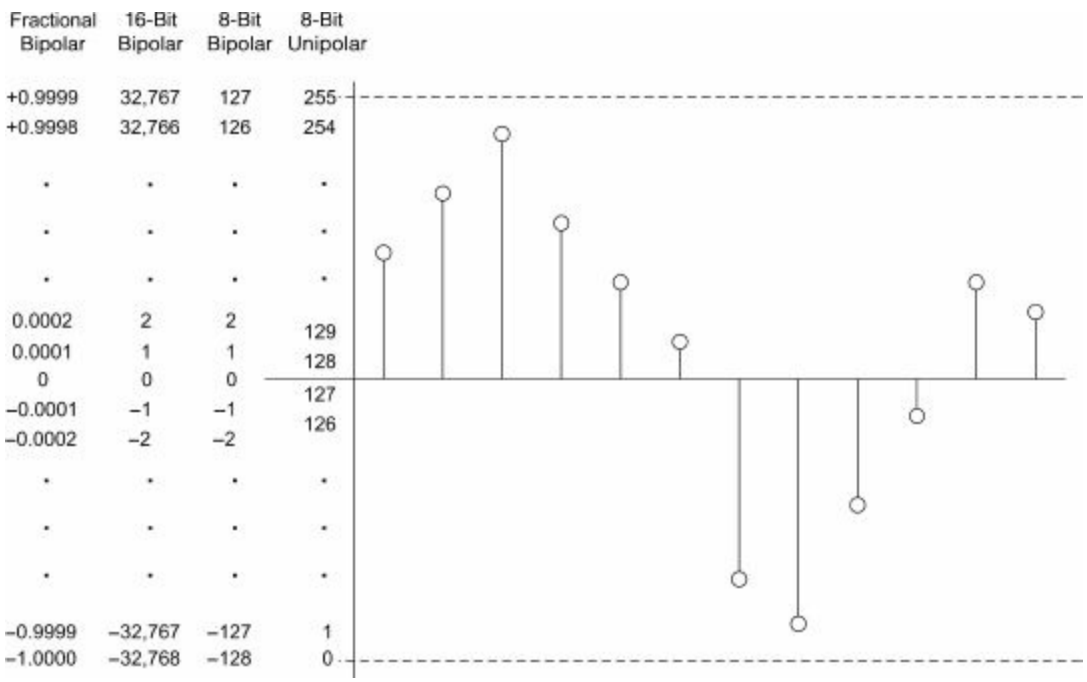


Figure 1.10: A comparison of several different types of data representations. The floating-point version is fixed for a range of −1.0 to +0.9999, though any range can be used.

This slight skewing of the data range is unavoidable if you intend on using the number zero in your algorithms, and that is almost guaranteed. In some systems the algorithm limits the negative audio data to the second most negative value. This is because phase inversion is common in processing algorithms, either on purpose or in the form of a negative-valued coefficient or multiplier. If a sample came in with a value of −32,768 and it was inverted, there would be no positive version of that value to encode. To protect against that, the −32,768 is treated as −32,767. The audio data that travels from the audio hardware adapter (DSP and sound card) as well as that stored in WAV files is signed-integer based. However, for audio signal processing, we prefer to use floating-point representation.

1.7 Using Floating-Point Data

In many audio systems, the DSP and plug-in data is formatted to lie on the range of −1.0 to +1.0 (which is simplifying the real range of −1.0 to +0.9999). In fact the plug-ins you code in this book will all use data that is on that same range. The reason has to do with overflow. In audio algorithms, addition and multiplication are both commonplace. With integer-based numbers, you can get into trouble quickly if you mathematically combine two numbers that result in a value that is outside the range of known numbers.

Consider the 16-bit integer bipolar format on the range of −32,768 to +32,767. Most of the values on this range, when multiplied together, will result in a product that is outside these limits. Addition and subtraction can cause this as well, but only for half the possible values. However, numbers between −1.0 and +1.0 have the interesting property that their product is always a number in that range. Converting an integer value to a fractional value along the normalized range of −1.0 to +1.0 in an *N*-bit digital audio system is easy, as is the reverse conversion shown in Equation 1.2:

$$\text{Fraction} = \frac{\text{Integer}}{2^N} \qquad \text{Integer} = \text{Fraction} * 2^N \tag{1.2}$$

where *N* is the bit depth of the system.

1.8 Basic DSP Test Signals

You need to know the data sequences for several fundamental digital signals in order to begin understanding how the DSP theory works. The basic signal set consists of

- Direct Current (DC) and step: DC is a 0Hz signal
- Nyquist
- ½ Nyquist
- ¼ Nyquist
- Impulse

The first four of these signals are all you need to get a ballpark idea of the frequency response of some basic DSP filters. The good news is that all the sequences are simple to remember.

1.8.1 DC and Step

The DC or 0 Hz and step responses can both be found with the DC/step input sequence: {…0, 0, 1, 1, 1, 1, 1, 1, 1, 1…}.

This signal in Figure 1.11 contains two parts: the step portion where the input changes from 0 to 1 and the DC portion where the signal remains at the constant level of 1.0 forever. When you apply this signal to your DSP filter and examine the output, you will get two pieces of information. The step portion will tell you the transient attack time and the DC portion will give you the response at DC or 0 Hz.

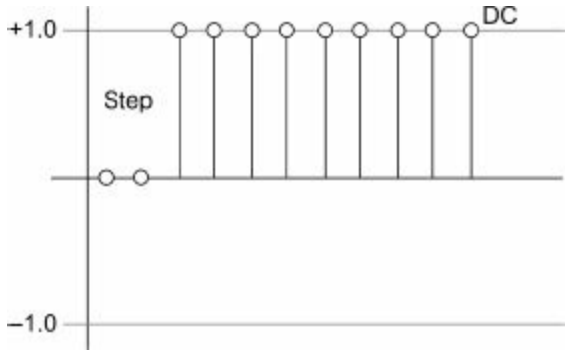


Figure 1.11 Representation of the DC/Step sequence.

1.8.2 Nyquist

The Nyquist input sequence represents the Nyquist frequency of the system and is independent of the actual sample rate. The Nyquist sequence is {…−1, +1, −1, +1, −1, +1, −1, +1…}.

The Nyquist frequency signal in Figure 1.12 is the highest frequency that can be encoded. It contains the minimum of two samples per cycle with each sample representing the maximum and minimum values. The two-sample minimum is another way of stating the Nyquist frequency as it relates to the sampling theorem.

1.8.3 ½ Nyquist

The ½ Nyquist input sequence in Figure 1.13 represents the ½ Nyquist frequency of the system and is independent of the actual sample rate. The signal is encoded with four samples per cycle, twice as many as Nyquist. The ½ Nyquist sequence is {…−1, 0, +1, 0, −1, 0, +1, 0, −1, 0, +1, 0, −1, 0, +1, 0…}.

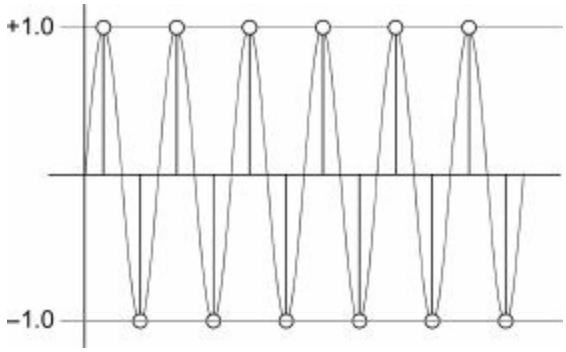


Figure 1.12 The Nyquist sequence.

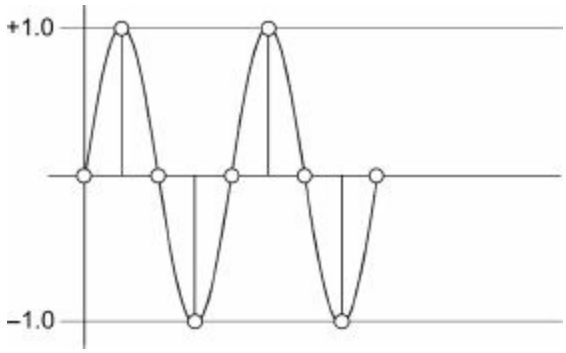


Figure 1.13 The ½ Nyquist sequence has four samples per cycle.

1.8.4 ¼ Nyquist

The ¼ Nyquist input sequence in Figure 1.14 represents the ¼ Nyquist frequency of the system and is independent of the actual sample rate. It is encoded with eight samples per cycle. The ¼ Nyquist sequence is {…0.0, 0.707, +1.0, 0.707, 0.0, −0.707, −1.0, −0.707, 0.0…}.

1.8.5 Impulse

The impulse shown in Figure 1.15 is a single sample with the value 1.0 in an infinitely long stream of zeros. The *impulse response* of a DSP algorithm is the output of the algorithm after applying the impulse input. The impulse sequence is {…0, 0, 0, 0, 1, 0, 0, 0, 0,…}.

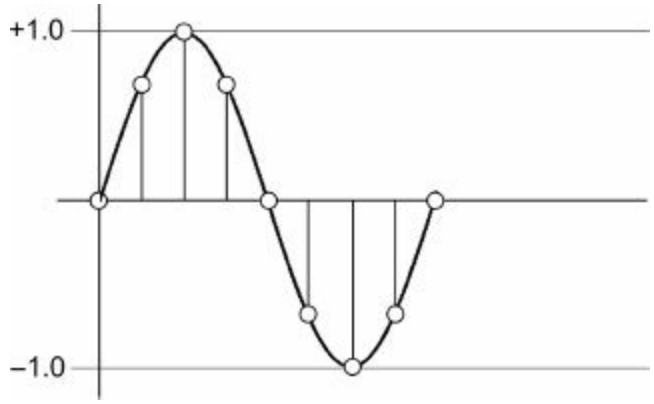


Figure 1.14 ¼ Nyquist sequence has eight samples per cycle.

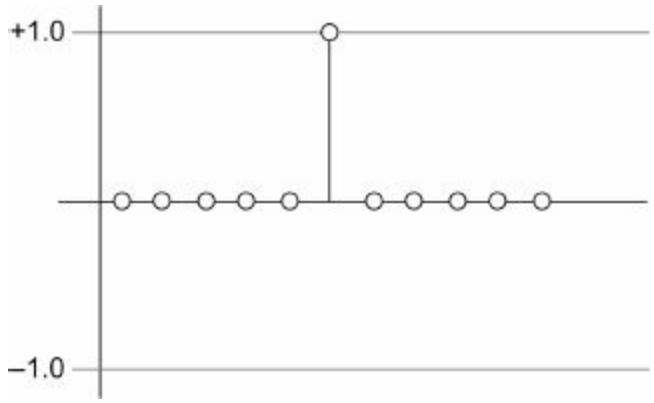


Figure 1.15 The impulse is a single nonzero sample value in a sea of zeros.

1.9 Signal Processing Algorithms

In the broadest sense, an algorithm is a set of instructions that completes a predefined task. The signal processing loop in Figure 1.8 is a picture of an algorithm for processing audio and control (UI) data in real time. In the specialized case of audio signal processing, an algorithm is a set of instructions that operates on data to produce an audio output bit-stream. Most of the exercises in this book involve processing incoming audio data and transforming it into a processed output. However, synthesizing a waveform to output also qualifies and in this special case, there is no real-time audio input to process. Most of the plug-ins in this book use the effects model, where an input sequence of samples is processed to create an output sequence, as shown in Figure 1.16.

Conventions and rules:

- $x(n)$ is always the input sequence; the variable n represents the location of the nth sample of the x-sequence.
- $y(n)$ is always the output sequence; the variable n represents the location of the nth sample of the y-sequence.
- $h(n)$ is the impulse response of the algorithm; a special sequence that represents the algorithm output for a single sample input or impulse.
- For real-time processing, the algorithm must accept a new input sample (or set of samples), do the processing, then have the output sample(s) available before the next input arrives; if the processing takes too long, clicks, pops, glitches, and noise will be the real-time result.

1.10 Bookkeeping

You can see that there are already three sequences to deal with: the input, output, and impulse response, all of which are coded with the same variable n to keep track of the location of samples within the sequence. The first step is to decide how to use n to do this bookkeeping task. Using it to represent the absolute position in the sequence would quickly become tiresome—how do you deal with indexing numbers like $x(12{,}354{,}233)$? Figure 1.17 shows an input signal, $x(n)$, starting from $t = 0$ or $x(0)$. The $x(0)$ sample is the first sample that enters the signal processing algorithm. In the grand scheme of things, $x(0)$ will be the oldest input sample ever. Indexing the numbers with absolute position is going to be a chore as the index values are going to become large, especially at very high sample rates.

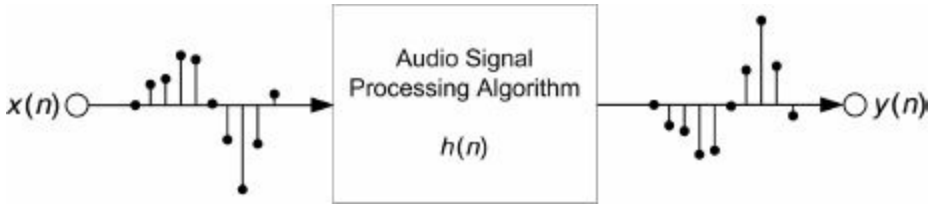


Figure 1.16: An audio signal processing algorithm that converts an input sequence $x(n)$ into an output sequence $y(n)$.

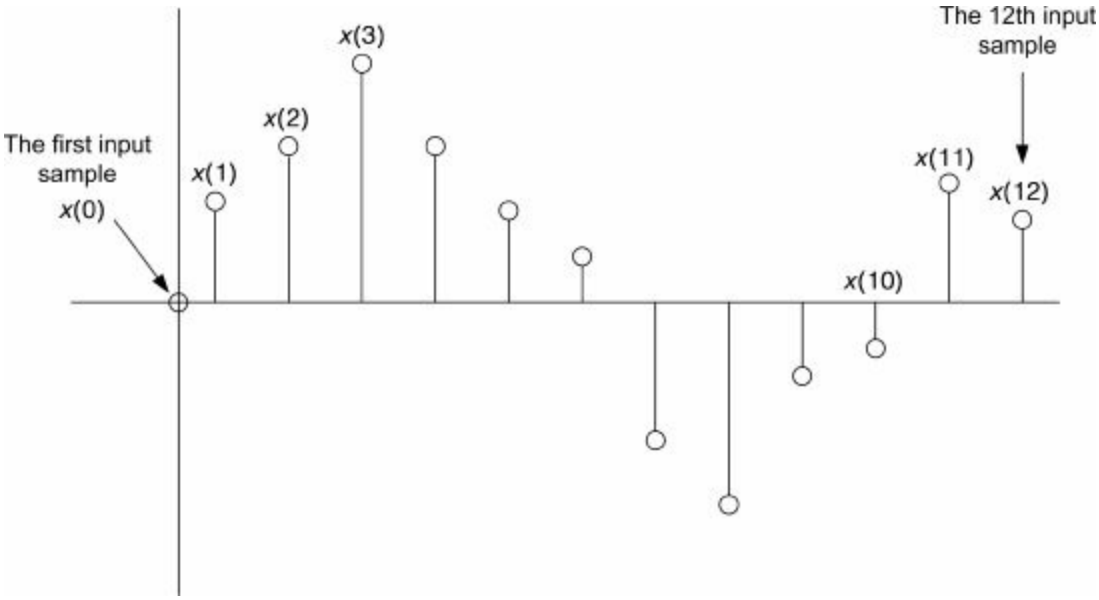


Figure 1.17: Using the absolute position within the sequence is one way to keep track, but the index values are going to get very large very quickly.

Another problem with dealing with the absolute position of samples is that algorithms do not use the sample's absolute position in their coding. Instead, algorithms use the position of the current sample and make everything relevant to that sample. On the next sample period, everything gets reorganized in relation to the current sample again. It might sound confusing at first, but it is a much better method of keeping track of the samples and more importantly, defining the I/O characteristics of the algorithm, called the *transfer function*. Figure 1.18 shows the input signal frozen at the current time, $x(n)$, and the other samples are indexed based on its location. One sample period later (Figure 1.19) you can see the frame of reference has shifted to the right and that $x(n)$ has now become $x(n - 1)$.

Bookkeeping rules:

- The current sample is labeled "*n*."
- Previous samples are negative, so $x(n-1)$ would be the previous input sample.
- Future samples are positive, so $x(n+1)$ would be the next input sample relative to the current one.
- On the *next* sample interval, everything is shuffled and referenced to the new current sample, $x(n)$.

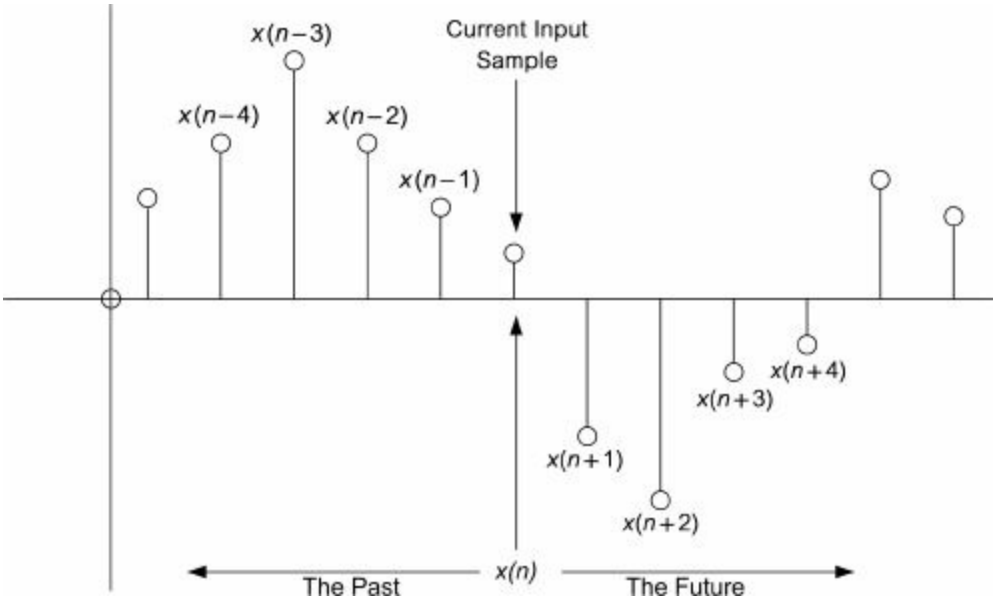


Figure 1.18: DSP algorithms use the current sample location as the reference location and all other samples are indexed based on that sample. Here you can see the current state of the algorithm frozen in time at the current input sample $x(n)$.

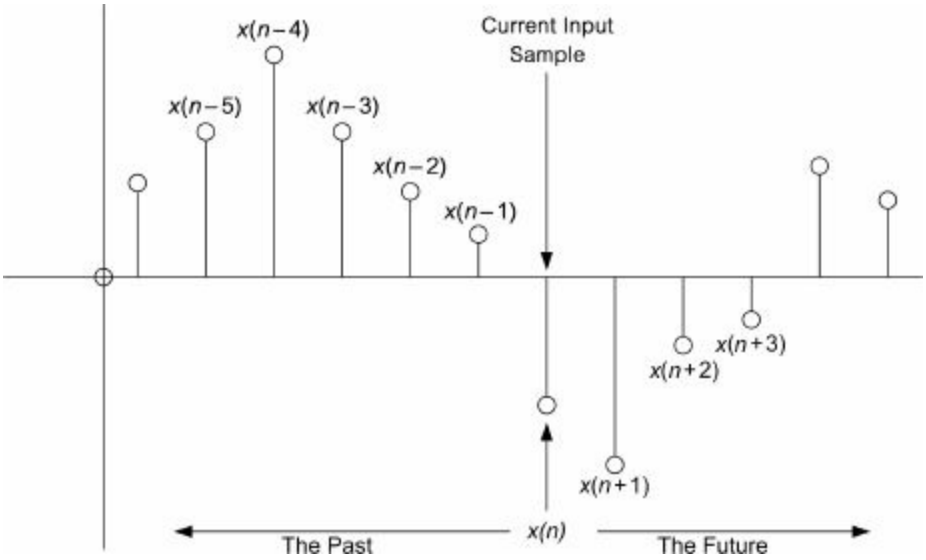


Figure 1.19: One sample period later, everything has shifted. The previous $x(n)$ is now indexed as $x(n-1)$ and what was the next sample, $x(n+1)$ now becomes the current sample.

1.11 The One-Sample Delay

Whereas analog processing circuits like tone-controls use capacitors and inductors to alter the phase and delay of the analog signal, digital algorithms use time-delay instead. You will uncover the math and science behind this fact later on in Chapters 4 and 5 when you start to use it. In algorithm

diagrams, a delay is represented by a box with the letter z inside. The z-term will have an exponent such as z^{-5} or z^{+2} or z^{0}—the exponent codes the delay in samples following the same bookkeeping rules, with negative (−) exponents representing a delay in time backward (past samples) and positive (+) representing delay in forward time (future samples). You call z the *delay operator* and as it turns out, time-delay will be treated as a mathematical operation.

You are probably asking yourself how you can have a positive delay toward the future, and the answer is that you can't for real-time signal processing. In real-time processing you never know what sample is going to come in next. However, in non-real-time processing (for example, an audio file that you are processing offline) you do know the future samples because they are in the file. Figure 1.20 shows two common ways to denote a one-sample delay in an algorithm block diagram.

Delay rules:

- Each time a sample goes into the delay register (memory location), the previously stored sample is ejected.
- The ejected sample can be used for processing or deleted.
- The delay elements can be cascaded together with the output of one feeding the input of the next to create more delay time.

If a sample $x(n)$ goes into the one-sample delay element, then what do you call the sample that is ejected? It's the previous sample that came in, one sample interval in the past. So, the output sample is $x(n-1)$. In Figure 1.21 you can see how delay elements can be cascaded with outputs taken at multiple locations generating multiple samples to use in the algorithm.

1.12 Multiplication

The next algorithm building block is the *scalar multiplication* operation. It is a sample-by-sample operator that simply multiplies the input samples by a coefficient. The multiplication operator is used in just about every DSP algorithm around. Figure 1.22 shows the multiplication operator in action. The inputs are simply scaled by the coefficients.

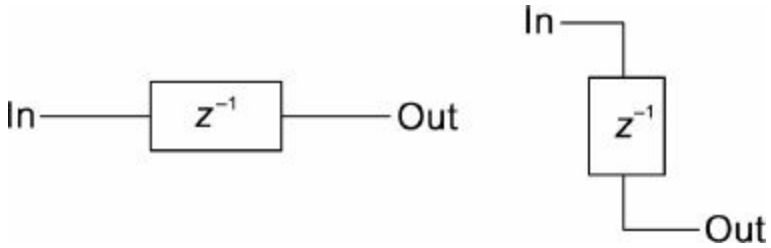


Figure 1.20: Two common ways to draw a delay; the one-sample delay is represented with the z^{-1}. Both versions are equivalent.

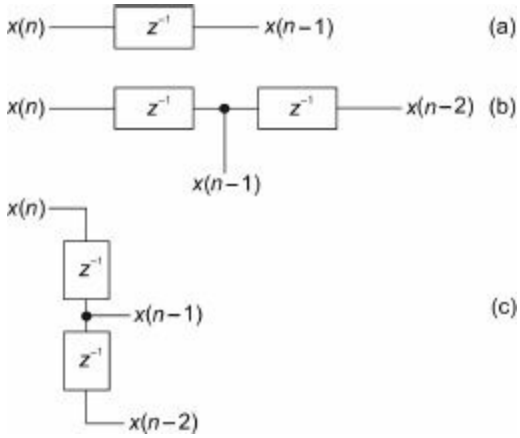


Figure 1.21: Three delay algorithms: (a) one-sample delay, (b) two one-sample delays cascaded, producing two different outputs, $x(n-1)$ and $x(n-2)$, notice that (c) is functionally identical to (b).

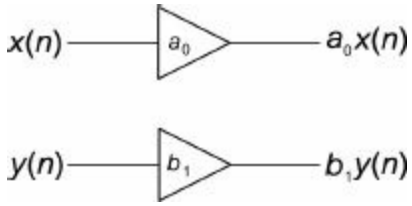


Figure 1.22: The multiplication operator is displayed as a triangle and a coefficient letter or number inside or just above it.

1.13 Addition and Subtraction

Addition and subtraction are really the same kind of operation—subtracting is the addition of a negative number. There are several different algorithm symbols to denote addition and subtraction. The operation of *mixing* signals is really the mathematical operation of addition. Figure 1.23 shows several ways of displaying the addition and subtraction operation in block diagrams.

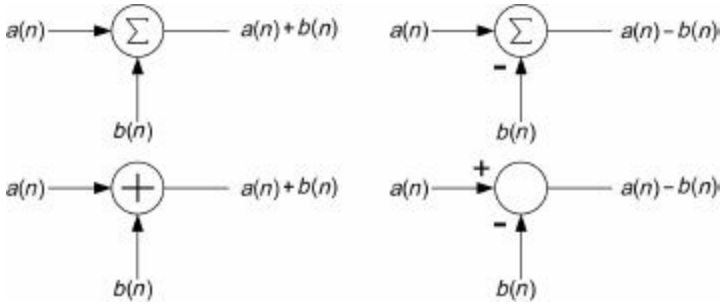


Figure 1.23: Addition and subtraction diagrams for two input sequences $a(n)$ and $b(n)$; these are all commonly used forms of the same functions.

1.14 Algorithm Examples and the Difference Equation

By convention, the output sequence of the DSP algorithm is named $y(n)$ and the mathematical equation that relates it to the input is called the *difference equation*. Combining the operations will give you a better idea of what the difference equation looks like. Figure 1.24 shows the difference equations for several combinations of algorithm building blocks. The output $y(n)$ is a mathematical combination of

inputs.

1.15 Gain, Attenuation, and Phase Inversion

As shown in Figure 1.25, a simple coefficient multiplier will handle the three basic audio processing functions of gain, attenuation, and inversion. If the coefficient is a negative number, phase inversion will be the result. If the coefficient has a magnitude less than 1.0, attenuation will occur, while amplification occurs if the magnitude is greater than 1.0.

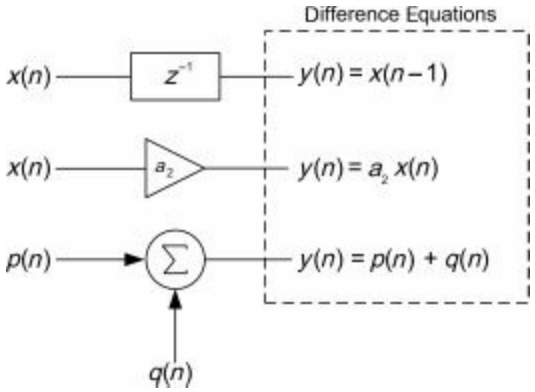


Figure 1.24: The difference equations relate the input(s) to the output via mathematical operations

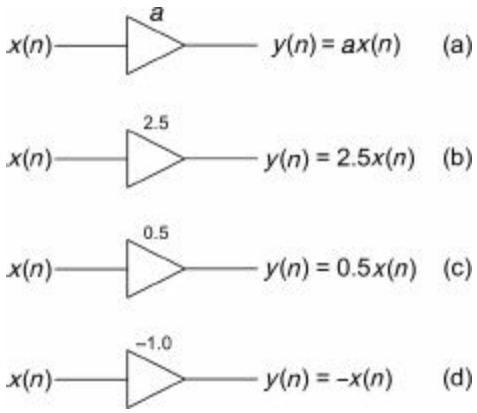


Figure 1.25: Examples of simple multiplier algorithms. Notice the different notation with the coefficient placed outside the triangle; this is another common way to designate it. (a) Simple scalar multiplication by an arbitrary value "*a*". (b) Gain is accomplished with a coefficient magnitude greater than one. (c) Attenuation reduces the size of the input value with a coefficient that is less than one. (d) Phase inversion turns the signal upside down by using a negative coefficient; a value of −1.0 perfectly inverts the signal.

1.16 Practical Mixing Algorithm

A problem with mixing multiple channels of digital audio is the possibility of overflow or creating a sample value that is outside the range of the system. You saw that by limiting the bipolar fractional system to the bounds of −1.0 to +1.0, multiplication of any of these numbers always results in a number that is smaller than either, and always within the same range of −1.0 to +1.0. However, addition of signals can easily generate values outside the ±1 limits. In order to get around this

problem, N-channel mixing circuits incorporate attenuators to reduce the size of the inputs, where the attenuation value is 1/N. When mixing two channels, the attenuators each have a value of ½ while a three-channel mixer would have attenuators with a value of 1/3 on each mixing branch. If all channels happen to have a maximum or minimum value at the same sample time, their sum or difference will still be limited to ±1. Figures 1.26 and 1.27 show the generalized mixing and weighted-sum algorithms.

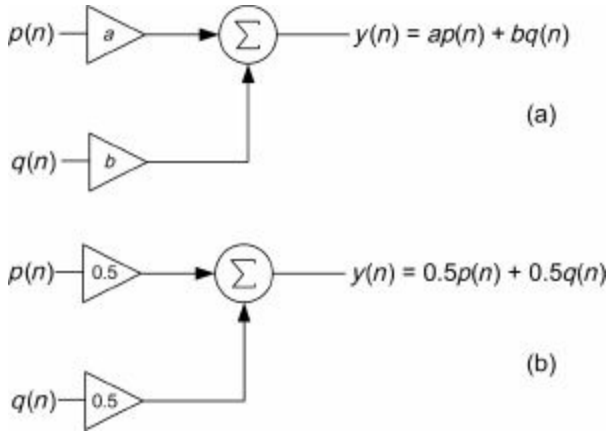


Figure 1.26: (a) Generalized mixer/summer with a separate coefficient on each line and (b) a normalized mixer which will not overflow or clip.

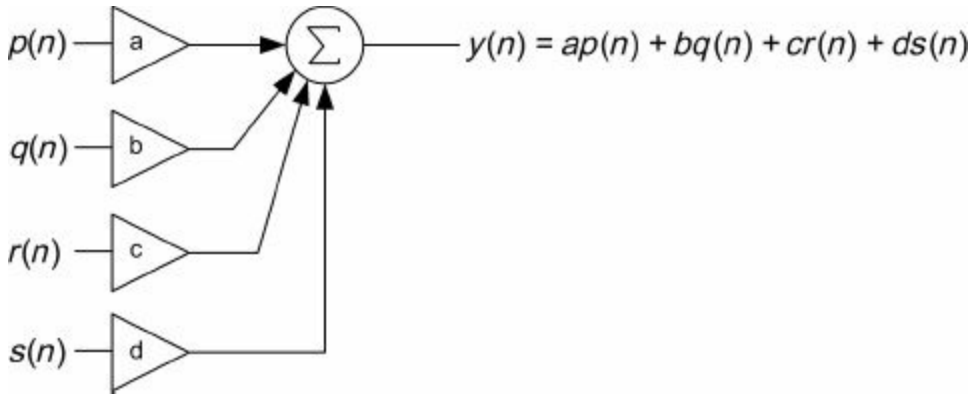


Figure 1.27: An example of a weighted-sum algorithm; each sample from each channel has its own weighting coefficient, a–d.

In the next chapter, you will be introduced to the anatomy of a plug-in from a software point of view. In Chapters 6 through 14, you will learn how DSP theory allows you to combine these building blocks into filters, effects, and oscillators for use in your own plug-ins.

Bibliography

Ballou, G. 1987. *Handbook for Sound Engineers*, pp. 898–906. Indiana: Howard W. Sams & Co.

Jurgens, R. K., ed. 1997. *Digital Consumer Electronics Handbook*, Chapter 2. New York: McGraw-Hill.

Kirk, R. and Hunt, A. 1999. *Digital Sound Processing for Music and Multimedia*, Chapter 1. Massachusetts: Focal Press.

KORG, Inc. 1991. *KORG Wavestation SR Service Manual*. Tokyo, Japan: KORG Inc.

Limberis, A. and Bryan, J. 1993. An architecture for a multiple digital-signal processor based music synthesizer with dynamic voice allocation. *Journal of the Audio Engineering Society*, Preprint No. 3699.

Pohlmann, K. C. 2011. *Principles of Digital Audio*, pp. 16–30. New York: McGraw-Hill.

Stearns, S. D. and Hush, D. R. 1990. *Digital Signal Analysis*, pp. 44–52. Englewood Cliffs, NJ: Prentice-Hall.

CHAPTER 2

Anatomy of a Plug-In

A plug-in is a software component that interfaces with another piece of software called the client in order to extend the client's capabilities in some way. For example, Internet browsers use plug-ins that implement added functions like searching and text messaging. In computer audio systems, a plug-in is usually an audio effect of some kind. However, a plug-in could also implement an oscilloscope or frequency analyzer function. Synthesizer plug-ins extend the client's functionality by working as musical instruments.

In order to start writing plug-ins, you need to know how the plug-in connects to and communicates with the client. Windows® plug-ins are typically packaged as dynamic link library, or DLL, files. Apple® plug-ins are packaged in a *bundle* which is configured as a *component*. The main difference between the two is in the lingo their designers use to describe them. Rather then try to accommodate both DLL and component labels during this discussion, we will just use DLL to describe both in conceptual form. Specific programmatic differences in the formats will be addressed and can be found on the website www.willpirkle.com.

2.1 Static and Dynamic Linking

C++ compilers include sets of precompiled libraries of functions for you to use in your projects. Perhaps the most common of these is the math library. If you try to use the sin() method you will typically get an error when you compile stating that “sin() is not defined.” In order to use this function you need to link to the library that contains it. The way you do this is by placing #include <math.h> at the top of your file. Depending on your compiler, you might also need to tell it to link to math.lib. When you do this, you are statically linking to the math.h library, a precompiled set of math functions in a .lib file. Static linking is also called implicit linking. When the compiler comes across a math function, it replaces the function call with the precompiled code from the library. In this way, the extra code is compiled *into* your executable. You cannot un-compile the math functions. Why would you do this? Suppose a bug is found in the sin() function and the math.h library has to be re-compiled and redistributed. You would then have to re-compile your software with the new math.h library to get the bug fix.

The solution is to link to the functions at run time. This means that these precompiled functions will exist in a separate file, which your executable will know about and communicate with, but only after it starts running. This kind of linking is called dynamic linking or explicit linking and is shown in Figure 2.1. The file that contains the precompiled functions is the DLL. It is more complicated because extra steps must be taken during run-time operation rather than relying on code compiled directly into the executable. The advantage is that if a bug is found in the library, you only need to redistribute the newly compiled file rather than re-compiling your executable. The other advantage is

that the way this system is set up—a client that connects to a component at run time—works perfectly as a way to extend the functionality of a client without the client knowing anything about the component when it is compiled. This is simplified when the DLL is loaded into the same virtual address space as the client. If you already understand this, you can skip the next section; otherwise, read on.

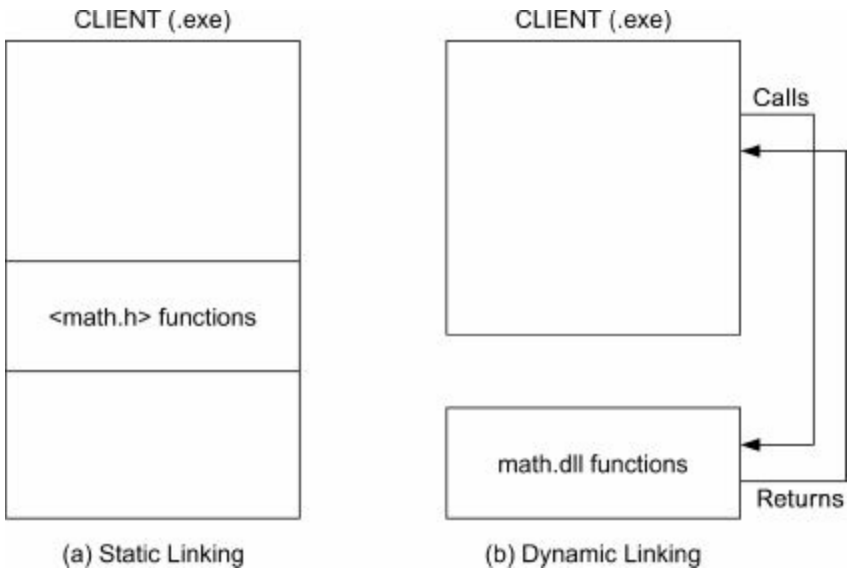


Figure 2.1: (a) In static linking the functions are compiled inside the client. (b) In dynamic linking the functions are located in a different file. This requires a communications mechanism between the client and DLL to call functions and return information.

2.2 Virtual Address Space and DLL Access

When you start a program (client) in a Windows 32-bit operating system (OS), the first thing that happens is the OS creates a new virtual machine and virtual address space. The maximum amount of memory that can be accessed by a 32-bit microprocessor is 2^{32} or about 4 gigabytes of memory. The client executable believes it has a full 4 gigabytes of memory and that its executable is loaded into part of this memory block. This block of memory is also called the process address space. The OS is responsible for maintaining this illusion for the client as well as all the other executables that are running on the system. This is done through virtualization of the microprocessor, memory, and peripherals.

While that topic could be the subject of another book, the main thing you need to know is that typically when the client loads the DLL and begins the communication process, the DLL is loaded into the same virtual address space as the client. This means that the client code might as well have the DLL code compiled into it since the addressing requires no translation. It should be noted that a DLL *can* be loaded into another process space or even on another machine across a network. In this case it is called an "out of process DLL"; however, inter-process communication is complicated and requires OS and/or network function calls. We will not be considering out of process DLLs here. With the DLL in the same process address space as the client, there is no extra overhead and the communication uses in-process addressing. Both types are shown in Figure 2.2.

In order to use the code in a DLL the client must perform two activities:

1. Load the DLL into the process address space.
2. Establish the communication mechanism for functions to be called.

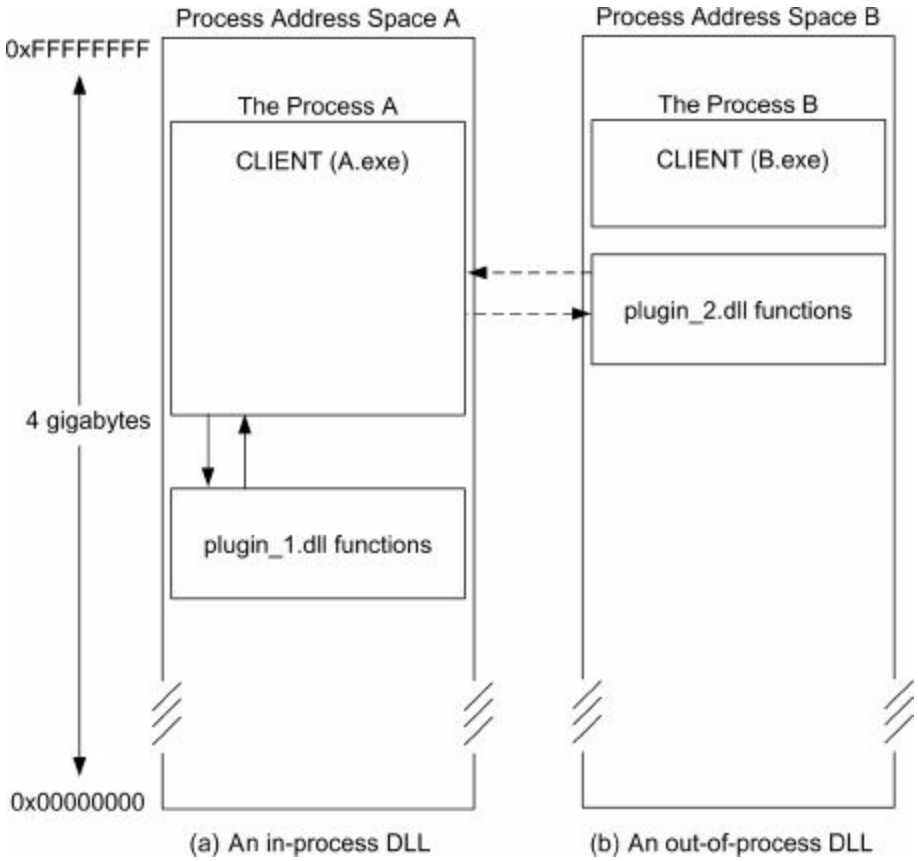


Figure 2.2: The process address space is a 4-gigabyte memory map starting at address 0x00000000 and going to 0xFFFFFFFF. When the client is launched it is placed inside the memory space. (a) An in-process DLL communicates directly with the client while (b) an out-of-process DLL communication requires OS intervention.

In the Windows OS, the three functions that are used to complete these tasks are

1. LoadLibrary(): Loads the DLL into the process address space.
2. GetProcAddress(): Retrieves pointers to functions in the DLL.
3. FreeLibrary(): Unloads the DLL from the process address space.

2.3 C and C++ Style DLLs

A DLL written in the C programming language consists of a set of stand-alone functions. There is no *main*() function. The functions can be defined and implemented in one. c file or can be broken into an interface file (.h) and implementation file (.c)—either way, the DLL performs a set of isolated functions. A problem with using the C programming language to write a DLL is the persistence of data. In C (and C++) the curly brackets {} define the *scope* of a function.

A fundamental problem is that the data declared inside a function cannot persist from one call to the next. One solution involves using global variables, which is generally frowned upon. A better solution to this problem is for the DLL to dynamically declare a data structure (e.g., using malloc()) that will hold all the persistent variables and then pass a pointer to this data structure back to the

client to maintain. During subsequent calls to the DLL, the client passes the pointer back to the DLL as a function argument so that it may operate on the persistent data. This is shown in Figure 2.3. When the DLL is no longer needed, it clears the memory by deleting the structure.

In the C++ programming language, the *class* data type defines an object that is a collection of member variables and member functions which can operate on those variables. By packaging a plug-in as a C++ class, you get several advantages; first, all of the benefits of C++ (encapsulation, polymorphism, etc.) are available during the coding process. Second, rather than allocating a data structure and returning a pointer to it, the DLL can create a new instance of the plug-in object and pass a pointer to the *object* to the client. Now, the client can simply call functions on the object—it does not have to communicate with the DLL again until it is time to either unload the DLL or, better yet, create another instance of the plug-in object. This leads to a third advantage over the C-based DLL: the client can create multiple plug-ins easily. The DLL can serve up multiple instances of the object. Sometimes, the plug-in is referred to as a *server* and this becomes another kind of client-server system. This is illustrated in Figure 2.4.

Any variable declared inside a function, after the first open curly bracket { is only defined for the duration of the function. After the closing curly bracket } is encountered, the variable ceases to exist.

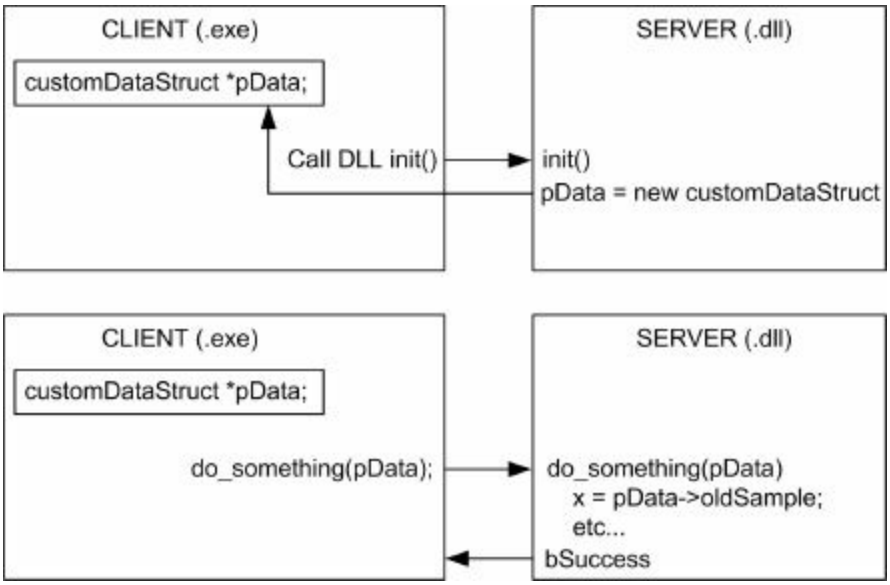


Figure 2.3: In a C-style DLL, the client first calls an initialization function and the DLL returns a pointer to a dynamically allocated data structure (pData), which the client stores. Later, the client calls a function do_something() and passes the pointer back to the DLL as a function argument. The DLL accesses the stored data, uses it, and then typically responds with a success flag.

2.4 Maintaining the User Interface

Most plug-ins have an associated graphical user interface (GUI or UI) with controls for manipulating the device. There are several schemes, but in general, when a new instance of the plug-in is created, a new child window is created with the GUI embedded in it. Whenever a GUI control changes state, a function on the plug-in is called. The client or GUI passes the plug-in information about which control

changed and the new value or state of the control. The plug-in then handles the message by updating its internal variables to affect the change in signal processing. Generally, the GUI appearance (the position of the sliders or knobs or the states of switches) is controlled automatically by the client or the GUI code itself. There are three different ways the GUI can be handled:

1. The client creates, maintains, and destroys the GUI and forwards control-change messages to the plug-in.
2. The client creates, maintains, and destroys the GUI but the GUI communicates directly with the plug-in.
3. The plug-in creates, maintains, destroys, and communicates directly with the GUI, independent of the client.

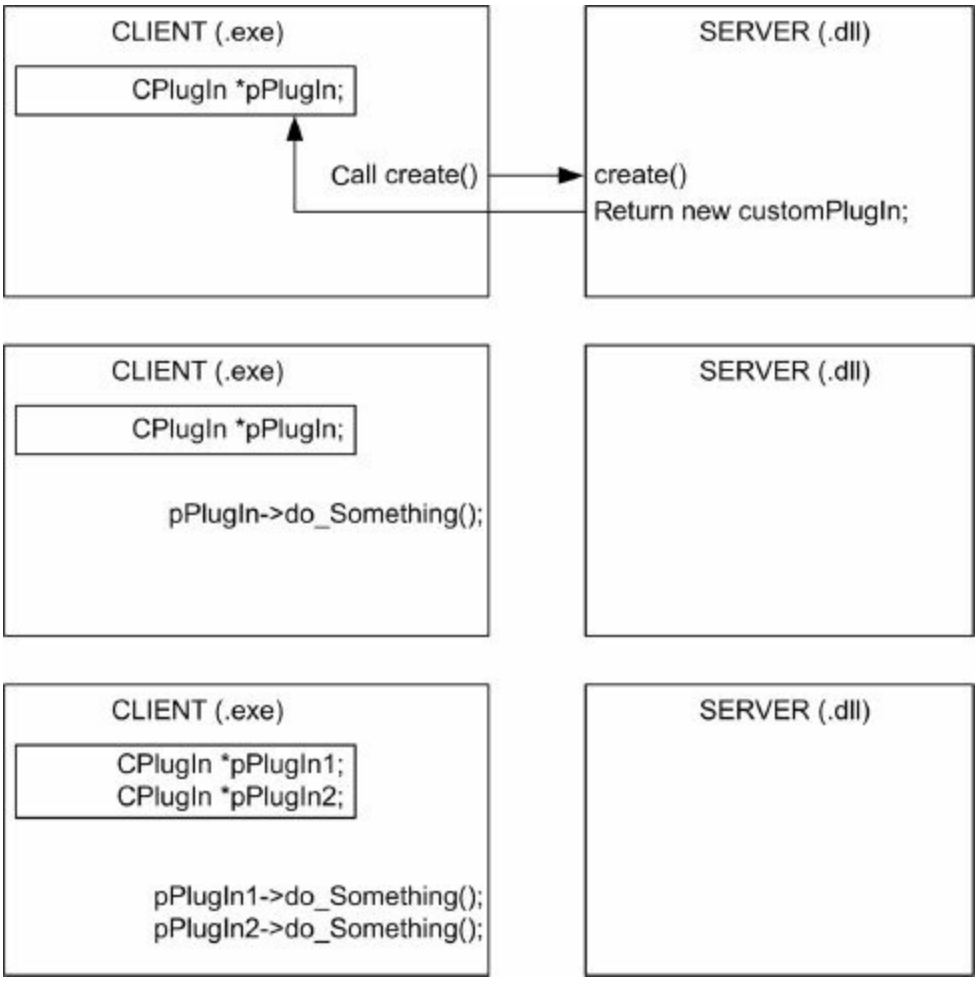


Figure 2.4: In a C++ plug-in, the client calls a creation function and the DLL (server) creates the object and passes a pointer to it back to the client. The client uses this pointer for future calls to the plug-in without having to bother communicating with the DLL. The client might also create multiple instances of the plug-in and then use the resulting pointers to implement processing functions.

Figures 2.5 through 2.7 show the three basic GUI scenarios. The first difference is in who creates, maintains, and destroys the GUI. When the client creates and maintains the GUI, it creates it as a child window which has the benefit that if the client is minimized or closed, the child windows will be hidden or destroyed accordingly. Therefore, the first two scenarios are more common. Some plug-ins can accommodate stand-alone GUI operation in addition to the client-child configuration. The second

difference is in how the communication flows: indirectly routed through the client or directly from the GUI to the plug-in. The RackAFX software uses the second paradigm; the client creates the GUI but the GUI communicates directly with the plug-in when controls are modified.

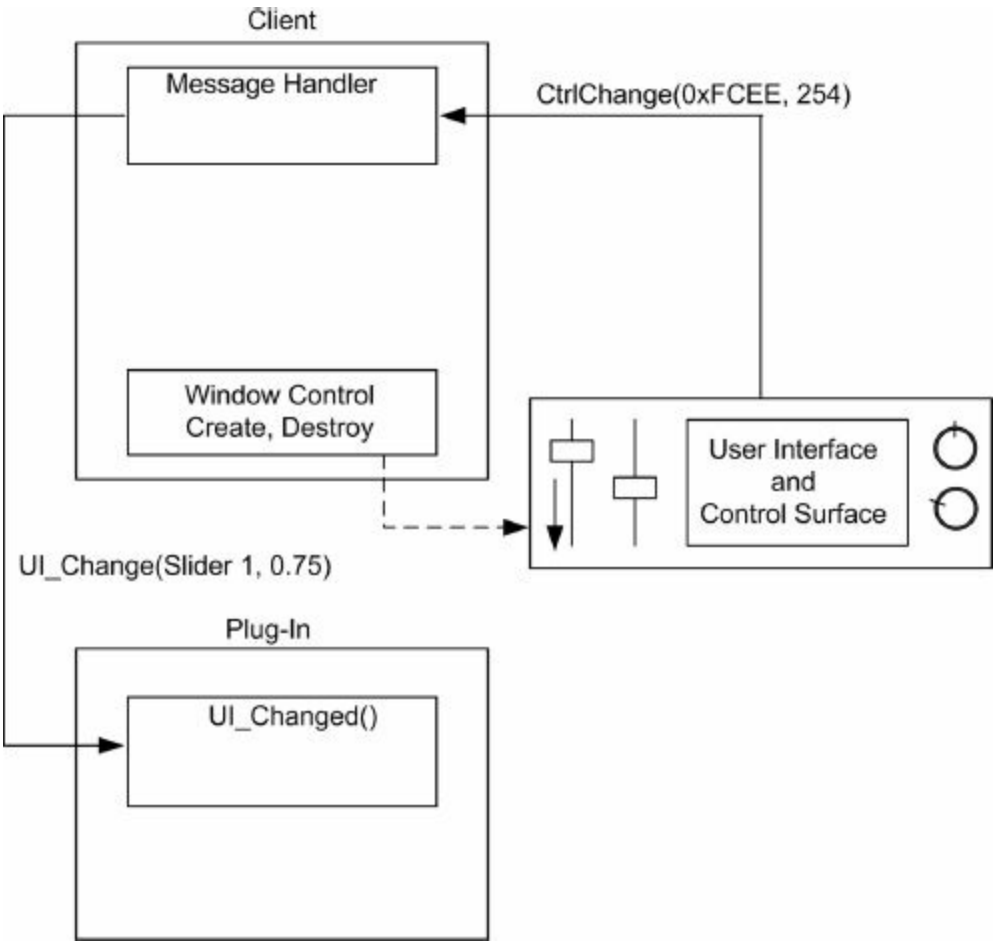


Figure 2.5: In this scenario, the client maintains the GUI and receives control change messages from it; the client then optionally translates that information before forwarding it to the plug-in.

2.5 The Applications Programming Interface

In order for the client-server scheme to work, both the client and DLL/plug-in must agree on the naming of the functions. This includes the creation function and all of the functions that the client will be able to call on the plug-in object. The plug-in might have other functions that the client doesn't know about, but they must agree on a basic set of them. Additionally, rules must be set up to define the sequence of function calls; the plug-in's author (that's you) will need to understand how the client intends to use the object.

The client must make sure that once it establishes these rules it adheres to them in future versions to avoid breaking the plug-in. On the other hand, the plug-in must also promise to implement the basic required functions to make the plug-in work. So, you can see that there is an implied *contract* between the client and DLL server. This contract is the *applications programming interface* or API. It is a definition of the functions an object must implement to be considered a proper plug-in as well as any additional functions that may be called or overridden. It defines the function prototypes and describes how the functions will be called and used. The API is written by the client manufacturer and is made available to programmers who want to create plug-ins for that target client. Some

examples are Direct-X®, VST®, AU®, and AAX®. Each of these manufacturers publishes an API that describes the contract with the plug-in developers. However, the basic operation and core functionality are generally the same.

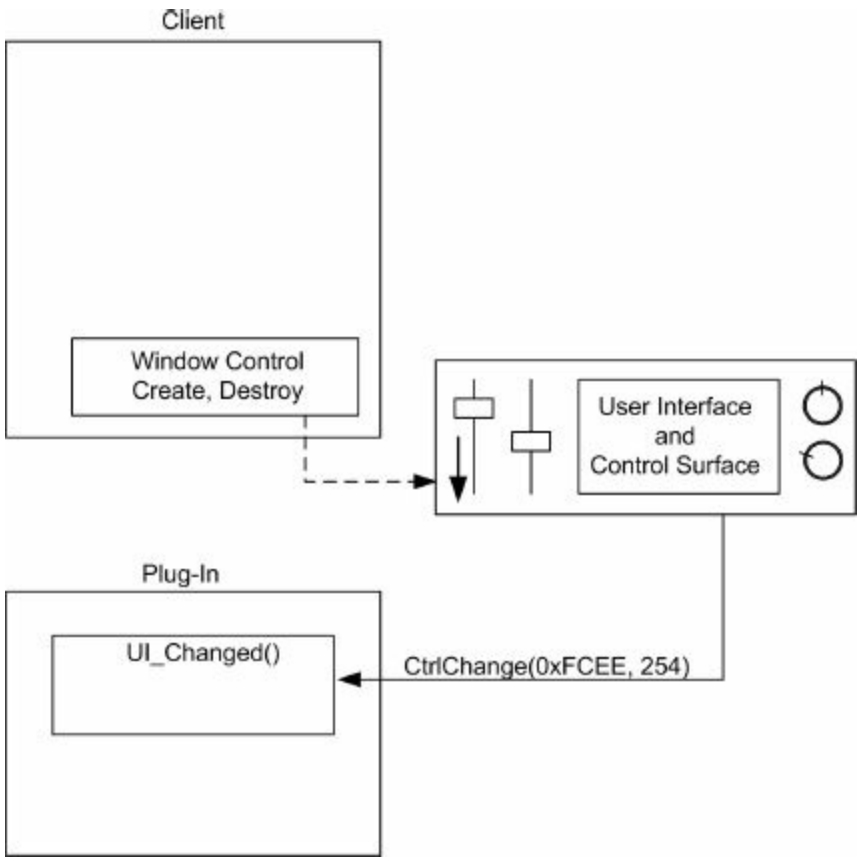


Figure 2.6: In this scenario, the client maintains the GUI, which communicates directly with the plug-in.

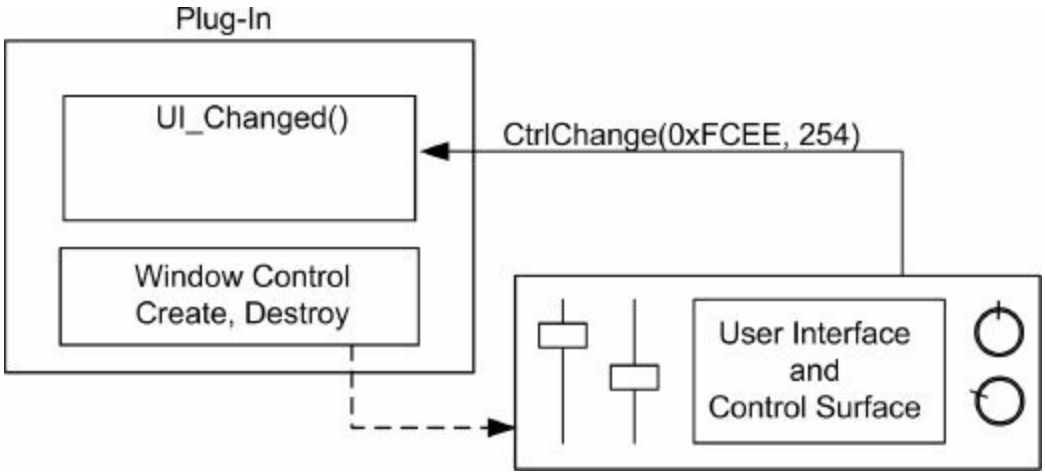


Figure 2.7: In another scenario, the plug-in maintains and communicates directly with the GUI without interference from the client.

C++ is especially useful here. Since the plug-in is an instance of a C++ object, the client manufacturer can specify that the plug-in is a derived class of a special base class that it defines. The base class is made to be abstract, containing virtual functions that the derived class overrides. These virtual functions provide the common functionality of the plug-in. There are two options here:

1. The manufacturer defines the base class as abstract and then provides default implementations of the virtual functions. Typically, the default implementations do nothing but return a success code. The plug-in authors then override whichever methods they need to. For example, the plug-in might not care about responding to Musical Instrument Digital Interface (MIDI) messages, so the default implementation of the MIDI function will suffice.
2. The manufacturer defines the base class as a *pure abstract* base class by making one or more of the virtual functions *pure* virtual functions. A pure abstract base class cannot be instantiated; only derived classes that implement all the pure virtual functions can. This forms a binding contract between the plug-in developer and the client manufacturer since the derived class won't work properly unless it implements the pure abstract functions that the client specifies.

The RackAFX software uses the first method, supplying default implementations for all virtual functions. As the plug-in author, you only override the functions you need to. But what are the typical required functions and were do they come from?

2.6 Typical Required API Functions

Plug-ins are designed after the hardware devices that they replace. The audio processing loop is the same as the hardware version you saw in Chapter 1. Figure 2.8 shows a software variation on that flowchart.

Although the various plug-in APIs are different in their implementations, they share a common set of basic operations. Table 2.1 details the responsibility of each function.

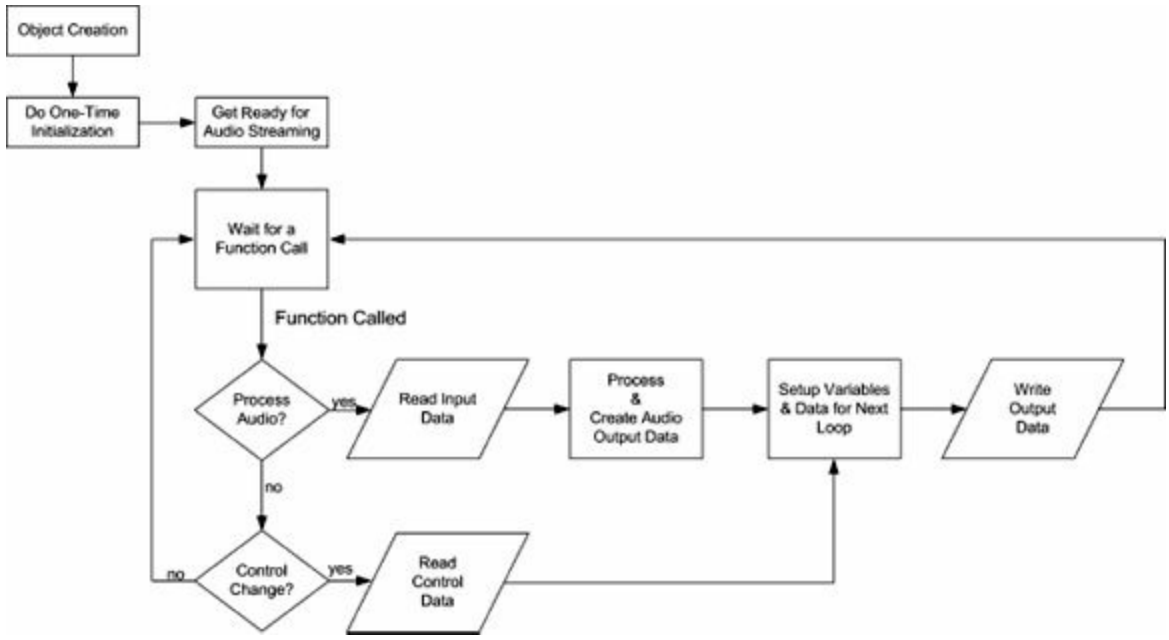


Figure 2.8: The software version of the DSP audio-processing loop. Most APIs also include functions to get or set information about the controls.

Table 2.1: The typical core operations that plug-in APIs share.

Function	Responsibility

One-time initialization	Called once when the plug-in is instantiated, this function implements any one-time-only initialization, usually consisting of initializing the plug-in variables, GUI, and allocating memory buffers dynamically.
Destruction	Called when the plug-in is to be destroyed, this function de-allocates any memory declared in the one-time initialization and/or in other functions that allocate memory. If there are any owned child-windows, the plug-in destroys them here.
Prepare for streaming	Called after the user has hit the play button or started audio streaming but before the data actually flows. This function is usually used to flush buffers containing old data or initialize any variables such as counters that operate on a per-play basis (not found in some APIs).
Process audio	The main function that does the actual signal processing. This function receives audio data, processes it, and writes out the result. This is the heart of the plug-in.
Get UI control info	Called to get information about a UI control—its name or label.
Set UI control info	Called when the plug-in needs to change the control information like its name or label.
Get UI control value	Called to get the value for the UI control that will set its appearance on the control surface.
Set UI control value	Called when the user makes a change to the plug-in's controls, this function is the message handler to deal with the user input. It usually causes a change or update in the plug-in's internal variables.

2.7 The RackAFX Philosophy and API

The fundamental idea behind the RackAFX software is to provide a platform for rapidly developing real-time audio signal processing plug-ins with a minimum of coding, especially with regard to the UI. In fact, most of the details of the connection between the RackAFX plug-in and the RackAFX UI screen are hidden from the developer so that he or she may concentrate more on the audio signal processing part and less on the UI details.

The RackAFX API specifies that the plug-in must be written in the C++ language and therefore takes advantage of the base class/derived class paradigm. The RackAFX API specifies a base class called CPlugIn from which all plug-ins are derived.

- RackAFX will automatically write C++ code for you to create a blank plug-in by creating a derived class of CPlugIn.
- As you add and remove controls from the control surface, the RackAFX client will automatically update your C++ code accordingly.
- This lets you focus on the signal processing and not the UI, making it a great tool for both rapid plug-in development and for teaching how to write plug-ins.

- After learning RackAFX, you will be able to understand other companies' APIs and learn to write plug-ins in their formats quickly and easily.
- Because the plug-in objects you create are written in C++, you can easily move them around between other APIs or computer platforms. You can wrap them to work easily in other systems too.

You only need to implement five functions in RackAFX to create a plug-in:

1. Constructor
2. Destructor
3. prepareForPlay()
4. processAudioFrame()
5. userInterfaceChange()

Figure 2.9 shows where these functions fit into the real-time audio processing loop.

2.7.1 __stdcall

In the RackAFX code you will see the qualifier __stdcall preceding each function prototype as well as implementation. The __stdcall calling convention is there for future compatibility with other compilers as well as other third-party software. The __stdcall is a directive that lets the compiler know how the stack will be cleaned up after function calls; it has no effect on the math, logic, or audio processing, so you can essentially ignore it.

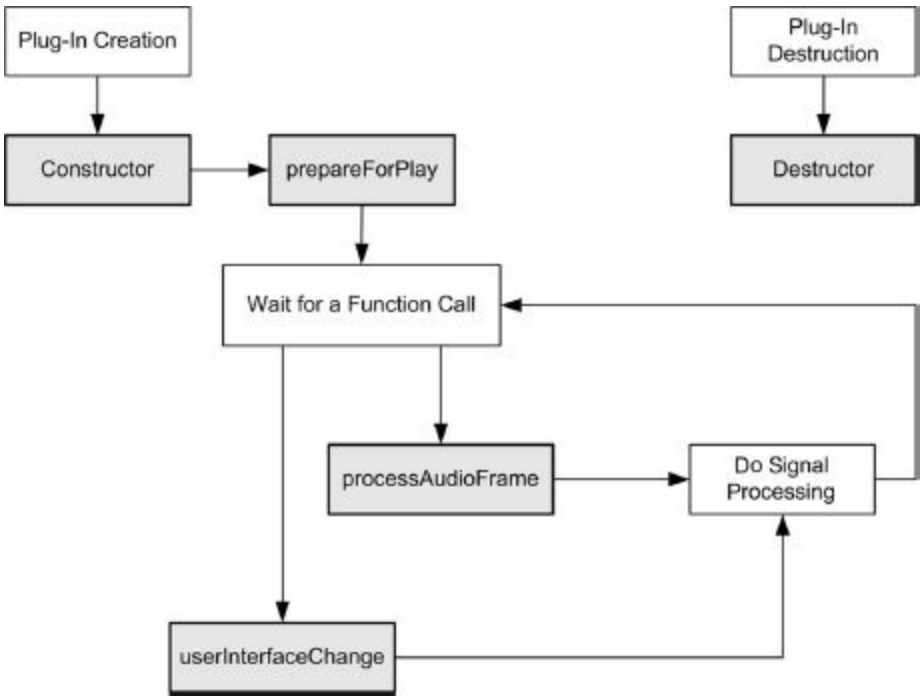


Figure 2.9: The RackAFX C++ plug-in version of the real-time audio processing loop in Figure 2.8.

Here is part of the interface file for the CPlugIn object plugin.h, which defines the contract or base class object interface for the primal six methods:

```
/*
        RackAFX(TM) Rapid Plug-In Development (RaPID) Client
        Applications Programming Interface
        Base Class Object Definition
        Copyright(C) Will Pirkle 2002-2012

        In order to write a RackAFX Plug-In, you need to create a C++ object that is
        derived from the CPlugIn base class. Your Plug-In must implement the
        constructor, destructor and virtual Plug-In API Functions below.
*/

// RackAFX abstract base class for DSP filters
class CPlugIn
{
public:
        // Plug-In API Member Methods:
        // The followung 5 methods must be impelemented for a meaningful Plug-In
        //
        // 1. One Time Initialization
        CPlugIn();
        // 2. One Time Destruction
        virtual ~CPlugIn(void);

        // 3. The Prepare For Play Function is called just before audio streams
        virtual bool __stdcall prepareForPlay();

        // 4. processAudioFrame() processes an audio input to create an audio
        output
        virtual bool __stdcall processAudioFrame(float* pInputBuffer,
                                                 float* pOutputBuffer,
                                                 UINT uNumInputChannels,
                                                 UINT uNumOutputChannels);
        // 5. userInterfaceChange() occurs when the user moves a control.
        virtual bool __stdcall userInterfaceChange(int nControlIndex);
```

The five functions in Table 2.2 are the core RackAFX API—implement them and you have a legitimate RackAFX plug-in. Best of all, the RackAFX plug-in designer will write and provide default implementations of all these functions for you. You need only to go in and alter them to change your plug-in's behavior. See Appendix A for a comparison of the RackAFX API and other commercially available formats as well as notes on using RackAFX plug-in objects inside API wrappers for other formats.

Table 2.2: The RackAFX API core functions.

RackAFX Function	Remarks
CPlugIn () Parameters: • none	The constructor for the plug-in object, this function is the one-time initialization function.
~CPlugIn () Parameters: • none	The destructor for the plug-in.

prepareForPlay() Parameters: • none	Function is called just before audio begins streaming. The audio file's sample rate, bit depth, and channel counts are extracted and then set by the client just before calling this method.
processAudioFrame() Parameters • pInputBuffer • pOutputBuffer • uNumInputChannels • uNumOutputChannels	The most important function in the API; this is where the audio processing is handled. You might do all the processing in this function or use it to call sub-functions. You are responsible for writing the data to the output buffer via pOutputBuffer. A pointer to one frame of audio input data. A frame is a set of channels as defined by uNumInputChannels. A pointer to one frame of audio output data. A frame is a set of channels as defined by uNumOutputChannels. The number of input channels in this frame of data. Currently, this value will be either 1 (mono) or 2 (stereo). The number of output channels in this frame of data. Currently, this value will be either 1 (mono) or 2 (stereo).
userInterfaceChange() Parameters: • nControlIndex	Function is called after the user changes a control in the RackAFX UI. RackAFX will automatically update the variable linked to this control prior to calling this method. The index of the control that was moved and whose value RackAFX has changed.

Bibliography

Apple Computers, Inc. 2011. *The Audio Unit Programming Guide*. https://developer.apple.com/library/mac/#documentation/MusicAudio/Conceptual/AudioUnitProgrammingGuide/Introduction/Introduction.html. Accessed August 7, 2012.

Bargen, B. and Donnelly, P. 1998. *Inside DirectX*, Chapter 1. Redmond, WA: Microsoft Press.

Coulter, D. 2000. *Digital Audio Processing*, Chapters 7–8. Lawrence, KS: R&D Books.

Petzold, C. 1999. *Programming Windows*, Chapter 21. Redmond, WA: Microsoft Press.

Richter, J. 1995. *Advanced Windows*, Chapters 2 and 11. Redmond, WA: Microsoft Press.

Rogerson, D. 1997. *Inside COM*, Chapters 1–2. Redmond, WA: Microsoft Press.

Steinberg.net. *The Steinberg VST API*. http://www.steinberg.net/nc/en/company/developer/sdk_download_portal.html. (Note: you must create a free developer's account to download the API.) Accessed August 7, 2012.

Writing Plug-Ins with RackAFX

The RackAFX plug-in designer will help you write your plug-in. When you create a new RackAFX project, it will set up a new Visual C++ project folder for you and populate your project with all the files you will need. It will automatically create a new derived class based on the name of your project. When you set up graphical user interface (GUI) controls like sliders and buttons, it will write and maintain the code for you. You will be switching back and forth between RackAFX and your C++ compiler. There are buttons on the RackAFX GUI that will let you jump to the compiler as well as launch compiler functions like rebuilding and debugging. You will use RackAFX to maintain your GUI and your compiler to write the signal processing code.

3.1 Building the DLL

RackAFX sets up your compiler to deliver your freshly built dynamic link libraries (DLL) to the /PlugIns folder in the RackAFX application directory. If you ever want to see, move, or delete your DLL you can find this folder by using the menu item PlugIn > Open PlugIns Folder. Each time you create a new project, RackAFX creates a pass-through plug-in by default; you are urged to build and test the DLL right after creating the new project to check your audio input/output (I/O) and any other connections. You then write over the pass-through code with your own processing algorithm.

After a successful build, you use RackAFX to test and debug the plug-in. You tell RackAFX to load the DLL and create your plug-in. The client needs to handle four basic operations during the lifecycle of your component:

1. Creation of the plug-in
2. Maintaining the UI
3. Playing and processing audio through the plug-in
4. Destruction of the plug-in

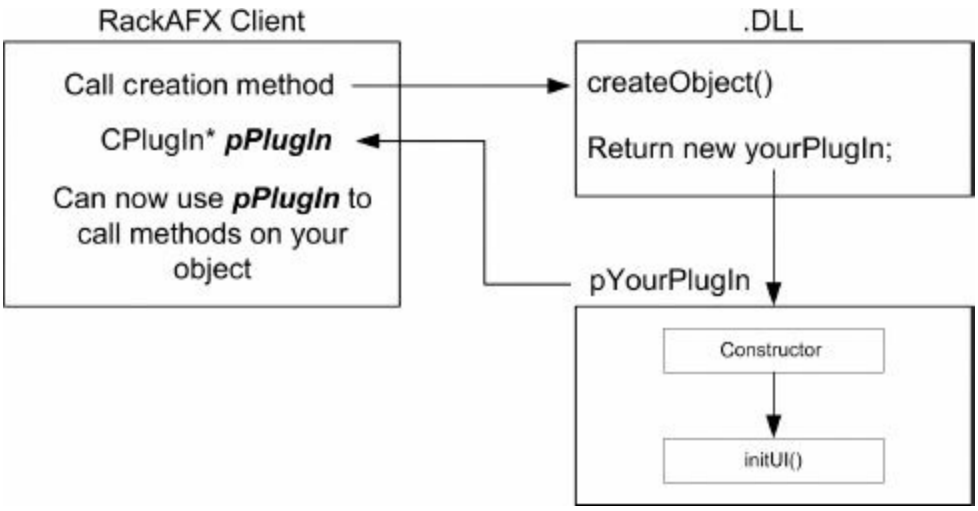


Figure 3.1: The *new* operator in createObject() dynamically creates your plug-in, which calls your constructor; the constructor in turn calls initUI() to create and initialize the user controls.

3.2 Creation

When you load a plug-in in RackAFX, you are actually passing the system a path to the DLL you've created. RackAFX uses an operating system (OS) function call to load the DLL into its process space. Once the DLL is loaded, RackAFX first runs a compatibility test, then requests a pointer to the creation method called createObject(). It uses this pointer to call the method and the DLL returns a newly created instance of your plug-in cast as the `PlugIn*` base class type. From that point on, the RackAFX client can call any of the base class methods on your object. Figure 3.1 shows the flow of operation during the creation phase.

Your constructor is where all your variables will be initialized. The very first line of code in the constructor has been written for you; it calls initUI(), which is a method that handles the creation and set up of your GUI controls. You never need to modify the initUI() method; RackAFX maintains this code for you.

3.3 The GUI

When you set up GUI elements such as sliders and buttons, RackAFX adds member variables to the .h file of your derived plug-in class. Each slider or button group controls one variable in your code. You set up each control with minimum, maximum, and initial values as well as supplying the variable name and data type. As the user moves the control, RackAFX calculates the variable's new value and delivers it to your plug-in automatically, updating it in real time. In some cases, this is all you will need and there is nothing left to write. In other cases, you will need to perform more calculations or logic processing in addition to just changing the control variable. So, in addition to changing and updating your internal GUI variable, RackAFX will also call the userInterfaceChange() method on your plug-in, shown in Figure 3.2.

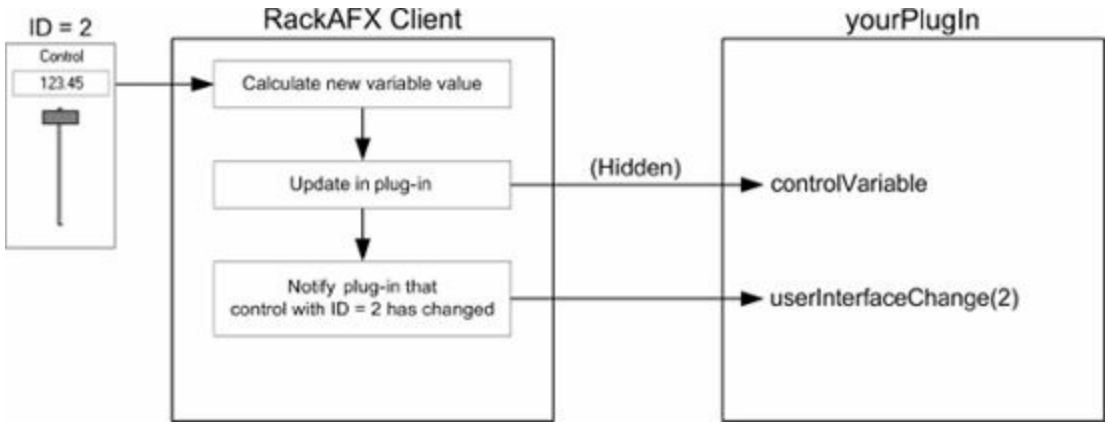


Figure 3.2: The sequence of events when the user moves the control with ID = 2 starts with a hidden change to the underlying linked variable, followed by a call to userInterfaceChange() passing the control's ID value as the parameter.

3.4 Processing Audio

When the user loads an audio file and hits the Play button, a two-step sequence of events occurs. First, the client calls prepareForPlay() on the plug-in. The plug-in will do any last initializations it needs before audio begins streaming. prepareForPlay() is one of the most important functions to deal with. Your plug-in has variables declared in it (see PlugIn.h) that contain information about the currently selected audio file:

```
// information about the current playing-wave file
int m_nNumWAVEChannels;
int m_nSampleRate;
int m_nBitDepth;
```

Just prior to calling prepareForPlay(), the client sets these values on your plug-in object. The reason this is done at this point is that the user can load multiple audio file types of varying channels (mono or stereo), sample rates, and bit depths at any time; thus, this is a per-play method. Many algorithms require these values to be known before certain things can be created or initialized. Almost all filtering plug-ins require the sample rate in order to calculate their parameters correctly. After prepareForPlay() returns, audio begins streaming. When audio streams, the client repeatedly calls processAudioFrame(), passing it input and output buffer pointers as shown in Figure 3.3. This continues until the user hits Pause or Stop.

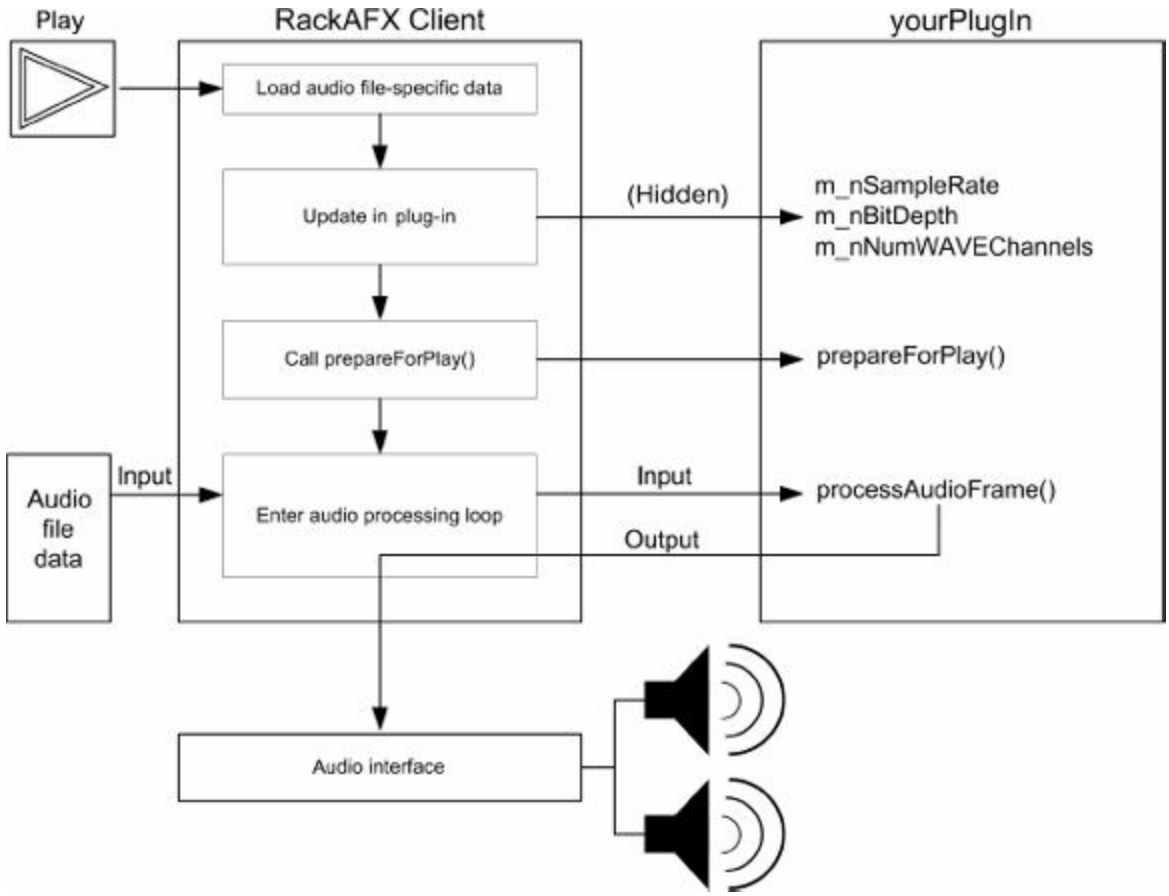


Figure 3.3: The sequence of events during the play/process operation; audio data from the file is processed in the plug-in and sent to the audio adapter for monitoring.

3.5 Destruction

When the user unloads the DLL either manually or by loading another plug-in, the client first deletes the plug-in object from memory, which calls the destructor. Any dynamically declared variables or

buffers need to be deleted here. After destruction, the client unloads the DLL from the process space.

3.6 Your First Plug-Ins

You can break the audio processing down into two very fundamental types:

1. Processing that only works on the current audio sample; requires no memory elements
2. Processing that requires the current *and* past audio samples; requires memory elements

We'll start with the first type and make a simple volume control. After that, we'll design a simple Audio Equalizer (EQ) control that will require memory elements. You will need the following installed on your computer:

- RackAFX
- Microsoft Visual C++ Express® 2008 or 2010 (both are free from Microsoft)
- Microsoft Visual C++ Professional® 2008 or 2010

There is no advantage to having the full version of Visual C++ (aka VC++) in RackAFX programming unless you plan on using your own GUI resources. Make sure that Visual C++ is installed on the same machine as RackAFX. See the website www.willpirkle.com for updates on supported compiler platforms. Once you understand the flow of writing and testing your plug-ins, you will find that you can move easily and swiftly through the rest of the book's projects because they all follow the same design pattern and the design chapters will use the same conventions for each project.

3.6.1 Project: Yourplugin

The first step will always be the creation of a new project. In this phase, RackAFX creates the C++ project directory and files along with a derived class based on the project name.

3.6.2 Yourplugin GUI

Next, you lay out your GUI controls based on the algorithm you are following and decide on the variable data types and names that will connect the GUI elements to your plug-in. This generally starts with writing the difference equation(s) for the algorithm. Variables in the difference equation will map to member variables and GUI controls in your plug-in. Abstracting the GUI from the algorithm requires that you decide which parameters you want the user to be able to adjust, as well as the variable names, min, max, and initial values, and data types. You can change your mind later and remove GUI elements or alter their parameters. A full-featured GUI designer allows you to finalize your plug-in's controls and package them in a neat and clean GUI. Often during the prototyping phase, you set up many sliders and controls and work on the voicing of the plug-in. Then, you hide some of them for the final product, only allowing the user access to a specific set of controls over a specific set range of values. This last step is part of the voicing of the final product.

3.6.3 Yourplugin.h File

The next phase involves adding your own member variables and member methods to the derived plug-in class. The variables and methods will depend on the algorithm you are implementing. In the first plug-in there are no extra variables or methods to supply.

3.6.4 Yourplugin.cpp File

In this step, you will add the new member method implementations (if you have any). Then, you will step through the .cpp file, altering and filling in the base class methods, typically in this order:

1. Constructor
2. prepareForPlay()
3. userInterfaceChange()
4. processAudioFrame()

Once these methods have been implemented, you will technically have a working plug-in.

3.6.5 Building and Testing

Finally, you will build the DLL, finding and fixing any issues. After the build succeeds, you can load it into the RackAFX client. You can use audio files, the oscillator, or your sound adapter input as the audio stimulus for your plug-in. You can run your plug-in in three modes: Insert, Mono > Stereo AUX, or Stereo > Stereo AUX to mimic the various ways a plug-in is used in a digital audio workstation (DAW).

3.6.6 Creating and Saving Presets

The presets are created and maintained on the main RackAFX UI. After you load your plug-in you can move the controls as you like and then save them as a preset. You use the Save Preset button on the toolbar. The presets will be saved inside a file until the next time you compile your plug-in; after that, the presets will be built into the DLL. You can add, modify, or delete the presets any time the plug-in is loaded.

3.6.7 GUI Designer

Once you have debugged and finished your plug-in, you can optionally use the GUI designer to create a compact, visually appealing GUI. See Appendix B and the website www.willpirkle.com for the latest details on using the GUI designer tools. In general, the book skips the GUI designer phase because it is so open ended; you are free to layout your final GUI however you wish. Check the website often for the latest news and example GUIs as well as video tutorials.

3.7 Design a Volume Control Plug-In

The easiest meaningful plug-in to write is a volume control, which uses a single scaling variable, a_2, depicted in Figure 3.4.

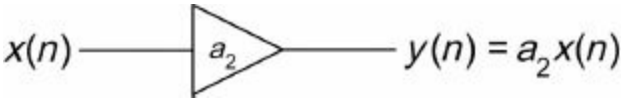


Figure 3.4: The volume control block diagram.

> *Coefficients* in a block diagram (or transfer function or algorithm) become *float member variables* in your plug-in code.

- $a_2 = 0$: Mute
- $a_2 = 1.0$: Max volume

The output samples $y(n)$ are a scaled version of the input $x(n)$ and the scaling factor is named a_2. a_2 is called a *coefficient* in the algorithm. The algorithm states that a_2 will vary between 0 (mute) and 1.0 (full volume).

3.8 Set up RackAFX for Use

Start the RackAFX software. You will start in prototype view, where you will see a blank control surface as shown in Figure 3.5. Your GUI may look slightly different or have different background images.

The control surface is what you use to create your UI. It is full of assignable controls you can connect to your plug-in's variables. The surface consists of:

- 40 assignable sliders (continuous controls).
- Universal LCD control with 1024 more continuous controls.
- Project controls (open, load, edit, rebuild, debug, jump to C++).
- Vector joystick (advanced, see website for more details).
- Assignable buttons.
- 10 assignable LED meters.
- Plug-in routing controls to test insert and aux effect modes.
- Prototype tab, the main GUI.
- GUI designer tab, which opens the designer for editing; you must have GUI controls declared first. See the website for more details.

The menu and toolbar consist of two parts: the left and right side. The left side (Figure 3.6) implements the majority of the software functionality, while the right side (Figure 3.7) maintains lists.

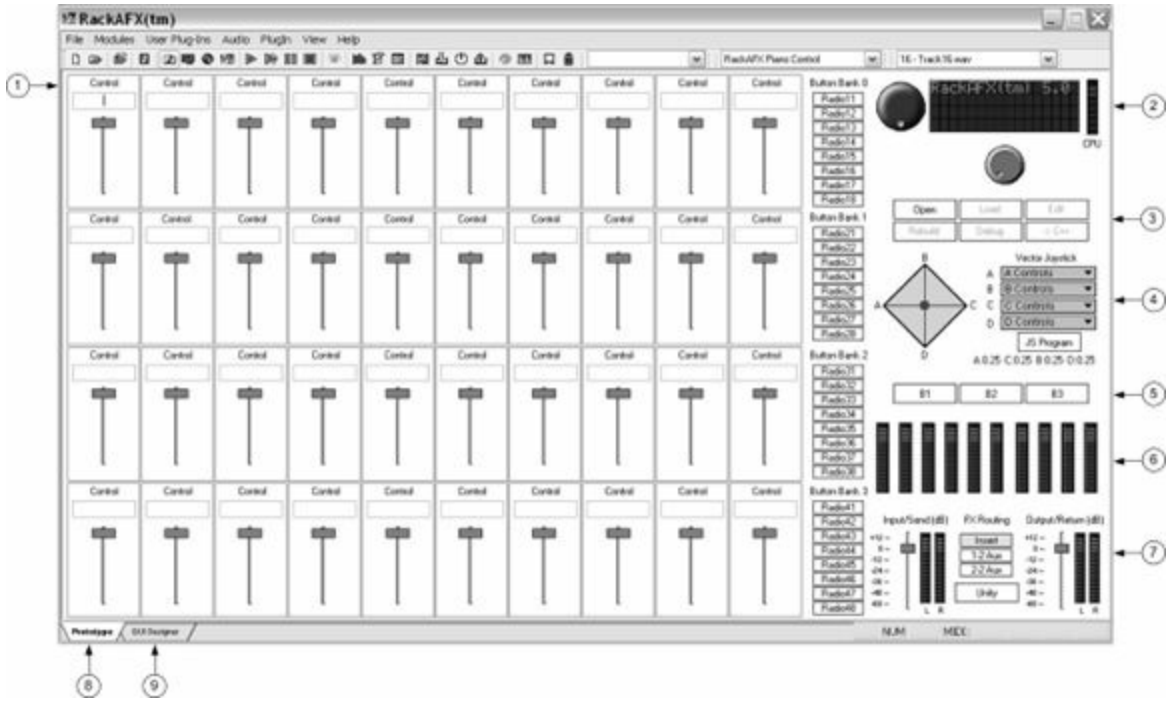


Figure 3.5: When you start RackAFX, it opens in prototype view. It features the control surface and plug-in routing controls.

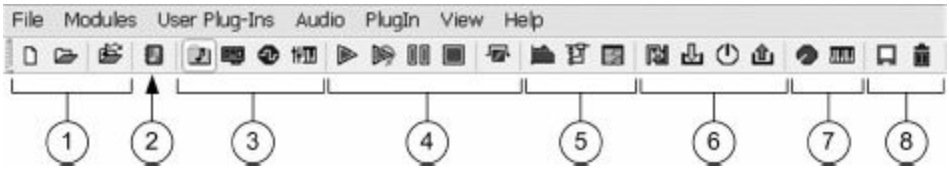


Figure 3.6: The menu and toolbar on the left handle most of your plug-in development.



Figure 3.7: The dropdown boxes on the right let you store and recall presets, choose a MIDI input controller (advanced), and keep track of the audio files you have been using.

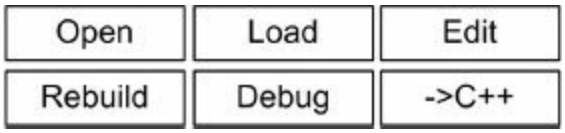


Figure 3.8: The Project/Compiler buttons make it easy to work with projects and control your compiler.

The menu items include:

- File: Manage projects by creating, editing or clearing the project.
- Modules: Built-in plug-ins that you can use for analysis and testing.
- User plug-ins: Each new plug-in you design gets loaded into this menu; you can audition or show off your plug-in in a standalone fashion.
- Audio: Manage all audio commands.

- Plug-in: Tools for loading/unloading and presets.
- View: Access the different windows.
- Help: Help information.

The toolbar items include:

1. New project, open project folder, open audio file
2. Set up low-level audio
3. Audio input mode: File, analog audio input, oscillator, user oscillator/synth
4. Transport controls: Play, loop, pause, stop, bypass
5. Additional windows: Analyzer, block diagram, status window
6. Plug-in tools: Synchronize code, load, reset, unload
7. GUI windows: Custom GUI, RackAFX MIDI Piano
8. Presets: Save, delete

Finally, there is a bank of buttons that allow you to manipulate your projects as well as control the C++ compiler shown in Figure 3.8. The buttons are set up as follows:

- Open: Open an existing project.
- Load: Load/unload the DLL from process space.
- Edit: Change an existing project's settings.
- Rebuild: Rebuild the project.
- Debug: Launch the debugger.
- ->C++: Jump to the C++ compiler and restore if minimized.

3.9 Setup Preferences

Before you start working on projects, take some time to configure your preferences. This is where you will choose your C++ compiler and set your default directories. Choose View > Preferences to get the interface shown in Figure 3.9.

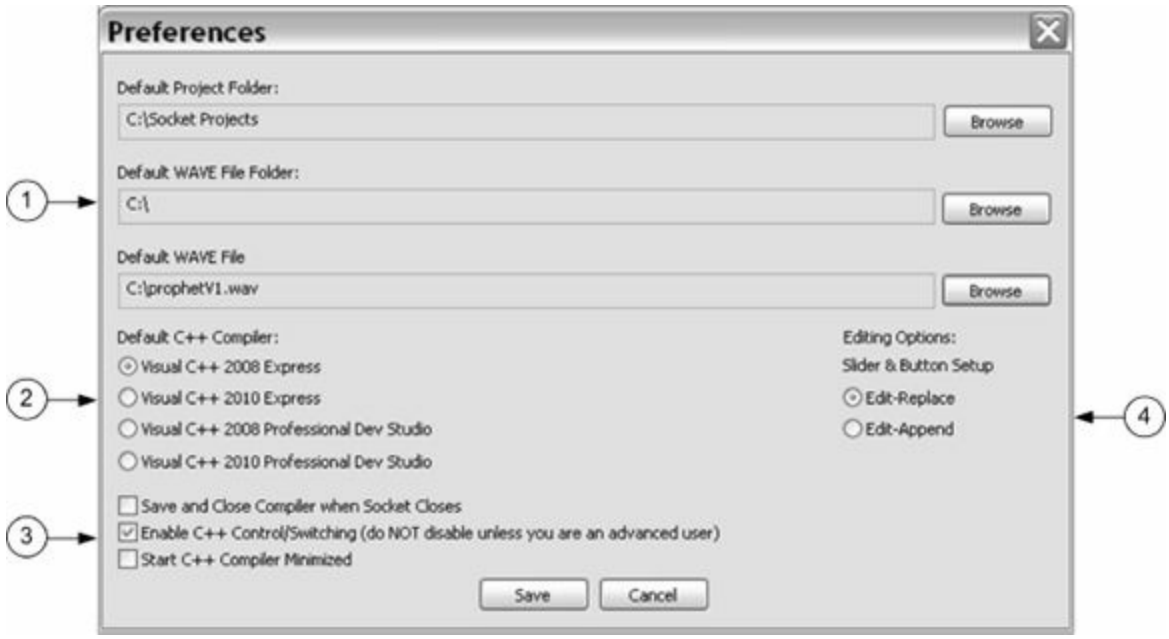


Figure 3.9: The preferences interface.

In the preferences you need to:

1. Choose your default folders for projects, WAVE files, and default WAVE files. You can use whatever directory you want for your project folder and you can also open projects from any other folder at any time; the default is simply for conveniently grouping all your projects together.
2. Choose a Visual C++ compiler.
3. Set the C++ options. Enable C++ Control/Switching should be left on for all but the most advanced users. C++ control/switching allows RackAFX to control Visual Studio, save files, launch the debugger, and so on.
4. Set up the edit options when entering information in the GUI slider/button dialogs.

3.9.1 Project: Volume

Create a new project with the menu, toolbar, or Ctrl+N and name it "Volume." The New/Edit Project window will appear and you can enter your project name. As you enter the name, the .h and .cpp files will be automatically named.

Notice the two C++ files listed in Figure 3.10—these are the interface (.h) and implementation (.cpp) files that RackAFX will create for you. They will contain a derived object named CVolume which will contain the bulk of your plug-in code. When you hit OK, several things happen. If you have C++ Control enabled in your preferences (it's the default) then you will see your C++ compiler start up. In Visual C++ you will see a new project and solution named "Volume." If you expand the Volume project then you can see the files that RackAFX wrote for you. Your derived class is contained in Volume.h and Volume.cpp. Before continuing, it's worth taking a peek into the RackAFXDLL.cpp file and locating the creation mechanism createObject():

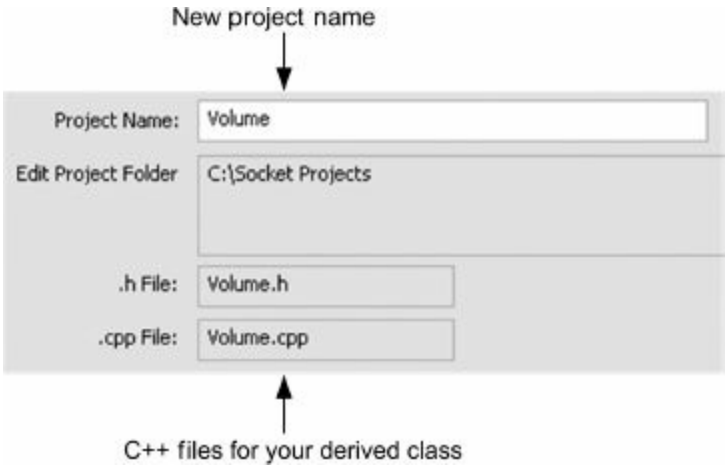


Figure 3.10: The top section of the New/Edit Project window. Notice that your project name becomes the name of a C++ object, so you will receive errors if you name the project in a way that produces illegal C++ syntax. Below this are more advanced settings that we will cover later.

```
//RackAFX Creation Function
DllExport CPlugIn* createObject()
{
	CPlugIn* pOb = new CVolume; // ***
	return pOb;
}
```

This is the method that the client calls on your DLL—you create the plug-in object with the *new* operator and return the pointer to the client. The DllExport specifier is OS-specific for calling a method in a DLL.

3.9.2 Volume GUI

You need to make sure you know the difference equation for each algorithm you want to implement. The difference equation relates the input and output samples and is what you are going to convert to C++ in order to make your plug-in. For this example, the equation is

$$y(n) = a_2 x(n) \tag{3.1}$$

> Each slider or button control on the UI will map to and control a *member variable* in your plug-in.

In RackAFX, you can see that all the sliders and buttons are disabled; the sliders don't move and the edit boxes won't accept text. You first need to set up some controls to create your UI or control surface.

Now, decide how to handle the difference equation. Notice the use of Hungarian notation for handling the variable names. See the website for more information if you are not familiar with this kind of naming convention—you are certainly free to name your variables whatever you like.

- Let's have only one volume coefficient and share it between the channels so that each channel has the same volume control.
- Let's have one slider control the volume coefficient.
- The coefficient a_2 will become a float member variable in the code; let's decide to name it "m_fVolume."
- We'll name the slider "Volume" on the GUI and link it to the variable m_fVolume.
- Slider minimum = 0.
- Slider maximum = 1.0.
- Initial setting = 0.75.

3.9.3 Configure a Slider Control

Choose the first slider in the upper left and right-click just inside the outer bounding box; a slider properties box appears (Figure 3.11). Note: Your dialog may look slightly different.

You need to fill out the slider properties with the proper values. You will notice that the uControlID value is 0 for this slider. This is the ID number that will link the slider to a variable in the object. You

cannot edit this cell. Start with the control name and enter "Volume." Hit Enter to advance to the next cell. For this version of the plug-in there are no units, so use backspace to delete it. The next cell is one of the most important—it is the data type for the variable that the slider will be linked with; the choices are available from a drop-down list. As shown in Figure 3.12, you can select the data type with the mouse, or you can just type the first letter (e.g., "d" for double) while the box is highlighted.

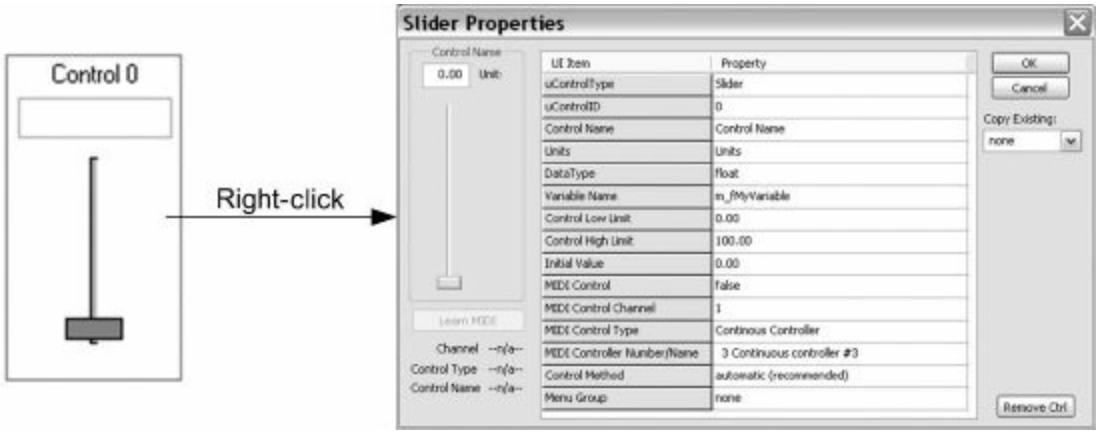


Figure 3.11: Right-click inside the bounding box of a slider and the slider properties window appears. This is how you configure the slider and link it to a variable.

You can create variables of the following types:

- float
- double
- integer
- enum: An enumerated Unsigned Integer (UINT) using a list of strings for the enum, for example {LPF, HPF, BPF, Notch}. We will work with enumerated UINTs later in the book.

We decided to use the *float* data type and to name the variable m_fVolume—it is really important to decide on a name and stick with it. Changing the variable name later can be tricky, so do yourself a favor and plan ahead. The completed slider properties are shown in Figure 3.13. The low and high limits are set and the initial value is set to 0.75. Do not worry about the MIDI control or other properties for this slider; see the website for details on these enhancements. After you are finished, choose the OK button to save the code.

Our control surface is pretty simple—it just consists of a single volume slider. If you realize you've made a mistake or left something out, just right-click in the slider box and fix the problem. You can also remove the slider by clicking the Remove Ctrl button on the properties window.

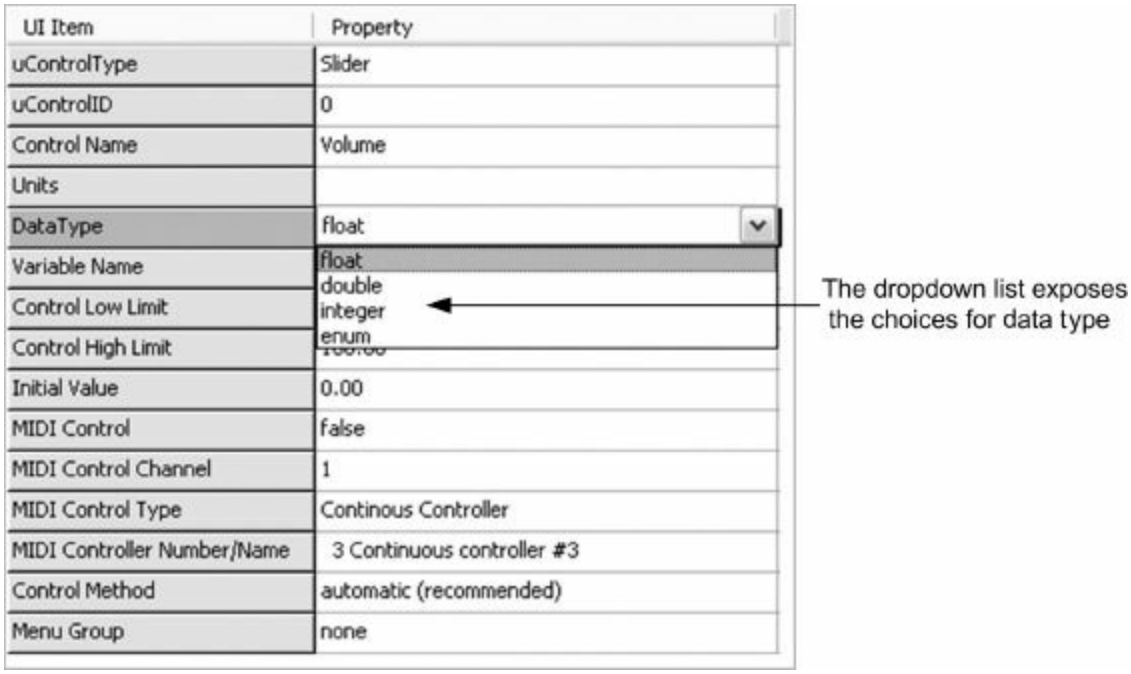


Figure 3.12: The data type is selected from a dropdown list control.

UI Item	Property
uControlType	Slider
uControlID	0
Control Name	Volume
Units	
DataType	float
Variable Name	m_fVolume
Control Low Limit	0.00
Control High Limit	1.00
Initial Value	0.75
MIDI Control	false
MIDI Control Channel	1
MIDI Control Type	Continous Controller
MIDI Controller Number/Name	3 Continuous controller #3
Control Method	automatic (recommended)
Menu Group	none

Figure 3.13: The completed slider properties.

As you add, edit, or remove controls from the main UI you will notice that RackAFX will flash to the compiler and back as it writes the code for you. You might use this flashing as a signal that the code update is synchronized. If you don't like it, minimize the compiler and the flashing will not occur. There is a special check-box in View > Preferences to start the compiler minimized for this very reason.

Your plug-in code will use the index value 0 (uControlID in the properties dialog) to map to the m_fVolume variable, which is controlled by the slider named "Volume" on the UI.

3.9.4 Volume.h File

Before we add the code, look around the plug-in object files (volume.h and volume.cpp) to get a better understanding of what's going on and what you'll need to modify. First, open the volume.h file and look inside:

```
// RackAFX abstract base class for DSP filters
class CVolume : public CPlugIn
{
public:  // plug-in API Functions
         //


        // 1. One Time Initialization
        CVolume();

        <SNIP SNIP SNIP>

        // 7. userInterfaceChange() occurs when the user moves a control.
        virtual bool userInterfaceChange(int nControlIndex);

        // Add your code here: ---------------------------------------------- //

        // END OF USER CODE ------------------------------------------------- //

        // ADDED BY RACKAFX -- DO NOT EDIT THIS CODE!!! ---------------------- //
        // **--0x07FD--**
        float m_fVolume;

        // **--0x1A7F--**
        // ------------------------------------------------------------------ //
};
```

Aside from the main plug-in functions we discussed in Chapter 2, you will see some more commented areas of code. In the first part marked // **Add your code here:** you can add more variables or function definitions just like you would in any .h file. Try to keep your code in the denoted area to make it easier to find and read. The area below that says:

// ADDED BY RACKAFX—DO NOT EDIT THIS CODE!!!

is very important—you will see your member variable m_fVolume declared in this area. This is the portion of the .h file that RackAFX modifies when you add, edit, or delete controls from your control surface. It is imperative that you let RackAFX maintain this part of the C++ code. There are several other portions of code in the .cpp file that have similar warnings and interesting hex symbols (0x1A7F, etc.); do not edit the code contained between the hex codes or commented areas.

You will see the notation **<SNIP SNIP SNIP>** frequently in the printed code as a reminder that code has been cut out for easier reading.

RackAFX writes C++ code for you! But, you have to be careful not to alter the RackAFX C++ code in any way. You can always tell if the code is RackAFX code because there will be warning comments and strange hex codes surrounding the RackAFX code. The RackAFX code is left for you to see only as a backup to your debugging and should never be altered by anyone except RackAFX.

In this case, check to verify that RackAFX added the float member variable m_fVolume as you anticipated. Next, move on to the volume.cpp file and have a look at it, starting from the top.

3.9.5 Volume.cpp File

Constructor and destructor

The constructor is the One-Time-Init function and is set up to:

- call initUI(): This is where your GUI controls are initialized; m_fVolume is initialized to 0.75 inside this function. It is important to make sure this remains the first line of the constructor so that your GUI variables are always initialized first.
- Set the plug-in name variable: This is what you will see in the user plug-in menu and on the GUI windows.
- Set the plug-in defaults (snipped out here); you will rarely need to change these variables.
- Give you a place to finish any of your own initializations at the end.

The destructor is empty because nothing has been allocated dynamically in this plug-in.

```
CVolume::CVolume()
{
	// Added by RackAFX - DO NOT REMOVE
	//
	// Setup the RackAFX UI Control List and Initialize Variables
	initUI();
	// END InitUI

	// built in initialization
	m_PlugInName = "Volume";

	// Default to Stereo Operation:
	// Change this if you want to support more/less channels

	<SNIP SNIP SNIP>

	// Finish initializations here
}

/* destructor()
	Destroy variables allocated in the contructor()

*/
CVolume::~CVolume(void)
{

}
```

prepareForPlay()

There is nothing to write yet since there are no custom controls or other allocations.

processAudioFrame()

This function is where the signal processing action happens. Above the definition is a comment block

as a reminder of how to get the audio data samples into and out of the I/O buffers. Currently, RackAFX only supports mono and stereo plug-ins. The left and right channels are accessed using the normal array-indexed C++ pointer mechanism. Of special note is the reminder that all values in and out are (and should be) between −1.0 and +1.0.

```
/* processAudioFrame

// ALL VALUES IN AND OUT ON THE RANGE OF -1.0 TO + 1.0

LEFT INPUT - pInputBuffer[0];
RIGHT INPUT - pInputBuffer[1]

LEFT OUTPUT - pInputBuffer[0]
LEFT OUTPUT - pOutputBuffer[1]

*/
bool __stdcall CVolume::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
UINT uNumInputChannels, UINT uNumOutputChannels)
{
	// output - input -- change this for meaningful processing
	//
	// Do LEFT (MONO) Channel; there is always at least one input/one output
	(INSERT Effect)
	pOutputBuffer[0] - pInputBuffer[0];

	// Mono-In, Stereo-Out (AUX Effect)
	if(uNumInputChannels -- 1 && uNumOutputChannels -- 2)
		pOutputBuffer[1] - pInputBuffer[0];

	// Stereo-In, Stereo-Out (INSERT Effect)
	if(uNumInputChannels -- 2 && uNumOutputChannels -- 2)
		pOutputBuffer[1] - pInputBuffer[1];

	return true;

}
```

Take a look at the function as RackAFX wrote it for you—it is designed to pass audio through unmodified. In this case, you simply write the output buffer with the data from the input buffer. In your processAudioFrame() function, get used to always processing the first channel then optionally processing the second one. This makes it easy to write mono/stereo capable plug-ins and will also make it easier to extend when RackAFX has more channel options. Because the code is already done, you could compile it right now and try it in RackAFX as a sanity check to make sure your audio hardware is set up properly. In fact, we'll do that right after examining the last few functions in the file (I promise you will write code shortly).

userInterfaceChange()

Perhaps the second most important function is userInterfaceChange(), which is called when the user changes a control on the control surface:

```
/* ADDED BY RACKAFX -- DO NOT EDIT THIS CODE!!! ---------------------------- //
      **--0x2983--**

      Variable Name             Index
-------------------------------------------
      m_fVolume                 0

-------------------------------------------

      **--0xFFDD--**
// ------------------------------------------------------------------------------ */
// Add your UI Handler code here ------------------------------------------------ //
//
```

As with processAudioFrame(), there is a "hint" comment above the function definition which reminds you how RackAFX has mapped your variable to a control index. In this case, the m_fVolume variable is mapped to index 0.

```
bool __stdcall CVolume::userInterfaceChange(int nControlIndex)
{
      // decode the control index, or delete the switch and use brute force calls
      switch(nControlIndex)
      {
            case 0:
            {
                  break;
            }

            default:
                  break;
      }

      return true;
}
```

userInterfaceChange() implements the first part of a switch/case statement in case you need to decode the control index and do something to the data before finally altering your code to reflect the change. Often, you will have nothing to write here either.

Build the plug-in. Since the default plug-in behavior is to simply pass audio unaffected, you can build the project now and test it in RackAFX to make sure everything is working properly. Rebuild your project from the compiler or from RackAFX's Rebuild button. You should get a clean build with no errors.

> You should *always* build and test your brand-new project first before modifying any code! You want to do this to make sure there are no C++ errors (you might have inadvertently hit a key or changed some code), as well as to make sure your audio system is working and you can hear the audio data.

At this point, you have built a DLL which is designed to serve up CVolume objects when the client requests them. The problem is that RackAFX doesn't yet know your CVolume plug-in is available. During the debugging phase, you need to get used to manually loading and unloading the DLL. You do this with the Load button or the toolbar/menu item. After you hit Load, RackAFX calls the appropriate functions to load your DLL into its address space. You will see the control surface change to reflect that your plug-in is loaded. You will also see the Load button change to read Unload. When it is time

to go back to C++, modify, and recompile, you'll need to unload the project first so you can reload it in its later state.

Use Audio > Load Audio File to load a test file. Then use the transport control buttons to play, loop, and pause or stop the file. The volume control should have no effect since we haven't written any code yet. Make sure you get audio at this point before continuing; if you don't, check your audio adapter settings.

In order to make the volume control slider work, we need to wire it into the processing code. The volume slider is directly mapped to the volume coefficient m_fVolume; as the slider moves from 0.0 to 1.0, so does the volume coefficient. So, the algorithm is simple to write: just multiply the input audio sample by the volume coefficient and set the output to that value. Switch to your C++ compiler and find the processAudioFrame() function. Modify it by doing the multiplication described above, which implements the difference equation for the filter.

```
bool __stdcall CVolume::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
                                  UINT uNumInputChannels, UINT uNumOutputChannels)
{
	// output = input -- change this for meaningful processing
	//
	// Do LEFT (MONO) Channel; there is always at least one input/one output
	// (INSERT Effect)
	pOutputBuffer[0] = pInputBuffer[0]*m_fVolume;

	// Mono-In, Stereo-Out (AUX Effect)
	if(uNumInputChannels == 1 && uNumOutputChannels == 2)
		pOutputBuffer[1] = pInputBuffer[0]*m_fVolume;

	// Stereo-In, Stereo-Out (INSERT Effect)
	if(uNumInputChannels == 2 && uNumOutputChannels == 2)
		pOutputBuffer[1] = pInputBuffer[1]*m_fVolume;

	return true;
}
```

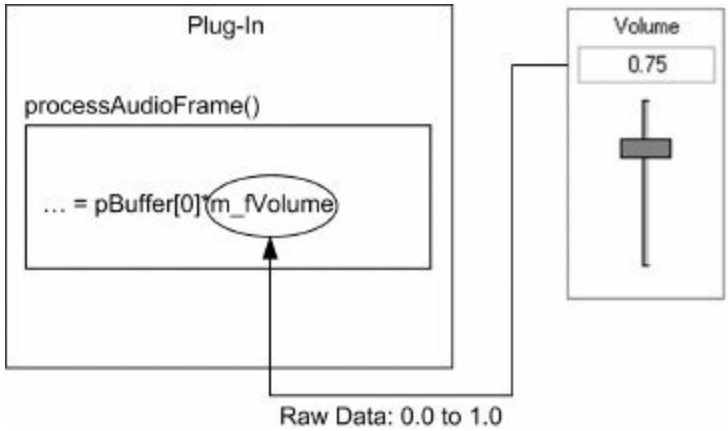


Figure 3.14: Here is the connection between the slider and the variable in the calculation. The processAudioFrame() function is using the raw slider data directly.

There are only three lines to modify, one for the first channel and another two for the other routing combinations. The modification is shown in **bold** where you are scaling the input by the volume coefficient. Can you see how this relates to the difference equation? If not, stop now and go back to figure it out. Now, rebuild the project and reload it into RackAFX. Try the slider and you will see that

the volume does indeed change. Congrats on your first plug-in!

What makes this plug-in so easy and quick to develop is that the slider volume control maps *directly* to a variable that is used in the processAudioFrame() function, as depicted in Figure 3.14. This means the data coming from the slider can be used directly in the algorithm. The data coming from the slider and controlling m_fVolume is said to be *raw data*. You use the raw value to affect the signal processing algorithm.

3.10 Design a Volume-in-dB Plug-In

This next example will show you how to *cook* your raw data to be used in the signal processing algorithm. The VolumedB plug-in will also be a volume control, but will operate in dB instead of using a raw multiplier. You may have noticed that your previous volume control didn't seem to do much in the upper half of the throw of the slider. This is because your ears hear logarithmically and so linear deflections of the slider do not correspond to linear changes in perceived loudness. To fix this, we'll design another plug-in that will operate in decibels (dB). The block diagram is identical to the first project, only the control range of values has changed.

- $a_2 = -96$ dB: Mute
- $a_2 = 0$ dB: Max volume

Now, the volume control is specified in dB, so you need a formula to convert the dB value to a scalar multiplier for the algorithm. You should memorize the dB equations now if you haven't already, since they will reappear over and over in audio plug-ins. This is the *cooking formula* that will take the raw data from the slider −96 to 0 dB and cook it into a variable we can use in our scalar multiplication:

$$\begin{aligned} dB &= 20\log(x) \\ x &= 10^{\frac{dB}{20}} \end{aligned} \tag{3.2}$$

You set up the cooking formula by implementing the userInterfaceChange() function which is always called when the user changes anything on the control surface. When the user moves the slider, you cook the data. This is shown conceptually in Figure 3.15.

The cooking function simply converts the dB into a scalar multiplier. In this case, the cooking function is short enough to simply leave inside the userInterfaceChange() function; as the cooking functions become more complex, you will probably want to break them out into separate functions, which are called from userInterfaceChange(). Remember the two RackAFX rules you've learned so far:

> *Coefficients* in a block diagram (or transfer function or algorithm) become *float member variables* in your plug-in code. Each slider or button control on the UI will map to and control a *member variable* in your plug-in.

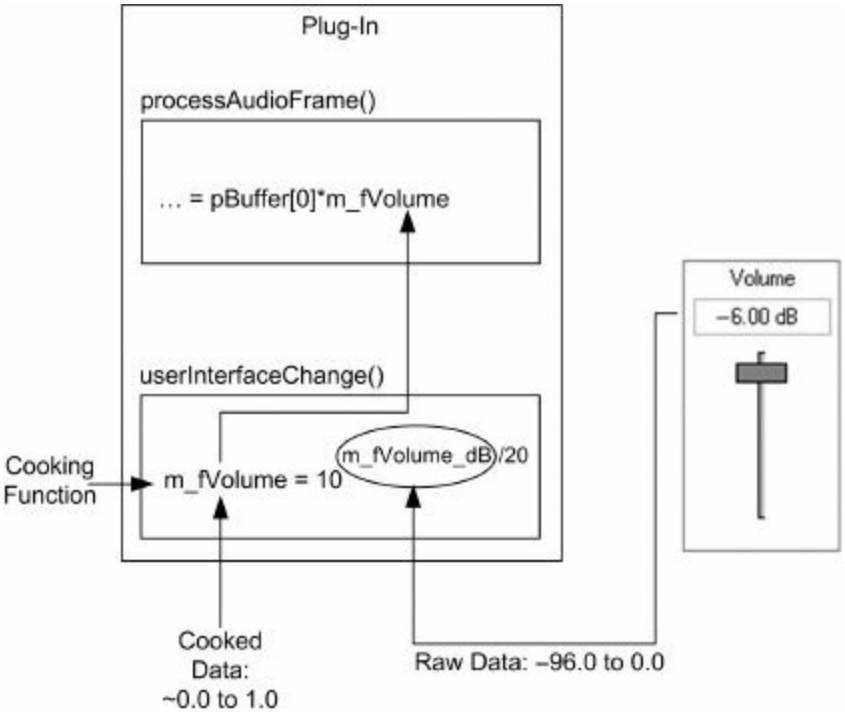


Figure 3.15: The volume-in-dB plug-in will have a single slider that generates values between −96 and 0.0 dB; you need to cook the raw dB values to use in your plug-in.

In the first plug-in, the variable was shared between the slider and the algorithm. Now we need two variables, one for the raw slider data and the other for the cooked algorithm processing. We will name them as follows:

- m_fVolume_dB: The raw slider data
- m_fVolume: The cooked value used in the algorithm

3.10.1 Project: VolumedB

Using the same method as before, create a new project named "VolumedB." As before, you don't have to worry about the advanced options at the bottom of the new project window. Your compiler will automatically start.

3.10.2 VolumedB GUI

We only need a single slider control. It will be adjustable from −96 to 0 dB. Set up the GUI in RackAFX by choosing a slider and right-clicking inside the bounding box. Set the slider according to Table 3.1.

3.10.3 VolumedB.h File

RackAFX has written the code and declared the variable float m_fVolume_dB but we still need to declare our second variable named m_fVolume, which stores the cooked data. Open the VolumedB.h file and declare the variable in the user declarations area:

```
// abstract base class for DSP filters
class CVolumedB : public CPlugIn
{
public:  // plug-in API Functions

        <SNIP SNIP SNIP>

        // Add your code here: ---------------------------------------------- //

        // our Cooked Volume Multiplier
        float m_fVolume;

        // END OF USER CODE ------------------------------------------------- //
```

Table 3.1: The slider properties for the VolumedB project.

Slider Property	Value
Control name	Volume
Units	dB
Variable type	float
Variable name	m_fVolume_dB
Low limit	−96
High limit	0
Initial value	−6

3.10.4 VolumedB.cpp File

Constructor

- Cook and initialize the member variable.
- Use the pow() function.

```
CVolumedB::CVolumedB()
{
        // Added by RackAFX - DO NOT REMOVE
        //
        // Setup the RackAFX UI Control List and Initialize Variables
        initUI();
        // END InitUI

        <SNIP SNIP SNIP>

        // Finish initializations here

        // Cook the raw data:
        m_fVolume = pow(10.0, m_fVolume_dB/20.0);
}
```

prepareForPlay()

There is nothing to do here because the volume variable does not need to be reset on each play event.

userInterfaceChange()

- Decode the control ID value.
- Cook the raw data using the formula.
- When the plug-ins get more complex, you can create separate cooking functions and then share the functions as needed.
- Make sure you check your control ID values from the comments so that they match properly.

```
bool __stdcall CVolumedB::userInterfaceChange(int nControlIndex)
{
        // decode the control index, or delete the switch and use brute force calls
        switch(nControlIndex)
        {
               case 0:
               {
                    // Cook the raw data:
                    m_fVolume = pow(10.0, m_fVolume_dB/20.0);
               }

               default:
                    break;
         }

         return true;
}
```

processAudioFrame()

- Implement the difference equation.

```
bool __stdcall CVolumedB::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
                                  UINT uNumInputChannels, UINT uNumOutputChannels)
{
       // output = input -- change this for meaningful processing
       //
       // Do LEFT (MONO) Channel; there is always at least one input/one output
       // (INSERT Effect)
       pOutputBuffer[0] = pInputBuffer[0]*m_fVolume;

       // Mono-In, Stereo-Out (AUX Effect)
       if(uNumInputChannels == 1 && uNumOutputChannels == 2)
              pOutputBuffer[1] = pInputBuffer[0]*m_fVolume;

       // Stereo-In, Stereo-Out (INSERT Effect)
       if(uNumInputChannels == 2 && uNumOutputChannels == 2)
              pOutputBuffer[1] = pInputBuffer[1]*m_fVolume;

       return true;
}
```

Build and test your plug-in. You should now hear a smooth volume transition as you move the slider. Depending on your sound system and volume levels, you might not hear much below −40 dB.

3.11 Design a High-Frequency Tone Control Plug-In

This example will show you how to implement the last of the digital signal processing (DSP) algorithm building blocks: the delay element (z^{-N}), where N is the number of samples of delay. In this example, $N = 1$, so we are dealing with a one-sample delay, a fundamental building block for many DSP algorithms. Although we won't get to the theory of how this delay creates a tone control until the

next chapter, it's worth it to go ahead and build a plug-in that uses it. After this exercise, you will be able to build almost any kind of DSP filter that uses one-sample-delay elements—and that's a lot of DSP filters. The later examples in the book will not be as detailed regarding the operation of RackAFX and your C++ compiler, so make sure you understand how to manipulate the software as needed to complete the project. Figure 3.16 shows the block diagram for the filter.

The design equation is as follows:

$$0.0 \le a_1 \le 0.49$$
$$a_0 = a_1 - 1.0 \tag{3.3}$$

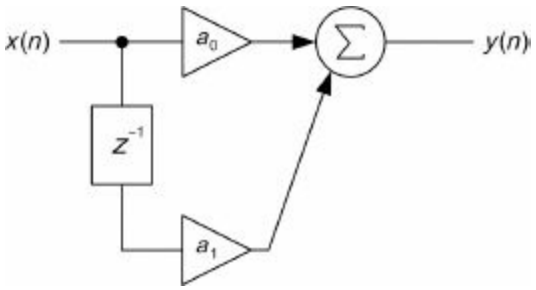


Figure 3.16: The HF tone control block diagram.

You already know that the coefficients a_0 and a_1 will become float member variables in our plug-in. But what about the one-sample-delay element, z^{-1}? In hardware, this would be a register to store the sample for one clock period. In software, it simply becomes another float variable, but it must be able to persist over the lifetime of a processing cycle. Therefore, like the coefficients, it will become a float *member* variable of our plug-in object.

You will do an example using arrays of floats in Chapter 7 when you implement digital delay lines that require long strings of z^{-1} elements. For now, we need to implement the single z^{-1} element in the block diagram. Right away, we need to decide if we are going to support multichannel (stereo) operation.

The last rule is really important and it is easy to get into trouble if you do not follow it. Unless the algorithm specifically deals with multichannel data, you will need to implement a separate algorithm for each channel, which means having separate coefficient and data (z^{-1}) elements for each channel. Even if you might be able to share the coefficients, you can never share the delay elements. We will need to declare variables for the following:

- Left channel a_0 and a_1 variables
- Left channel z^{-1} variable
- Right channel a_0 and a_1 variables
- Right channel z^{-1} variable

Delay elements will become *float member variables* in your plug-in object. For single-delay elements, you can simply assign separate variables. For multiple-sample-delay elements you may also use float arrays to store the data.

A DSP filtering algorithm, which is only described in mono or single channel format, that is, one input, $x(n)$, and one output, $y(n)$, cannot share delay elements between multiple channels. This means that you must duplicate your algorithms so that you have one set of variables for the left channel and one for the right channel.

3.11.1 Project: SimpleHPF

This plug-in is going to implement a very primitive HF (high frequency) tone control that behaves like a high-pass filter. It will attenuate low frequencies, leaving only the highest frequencies intact. Using the same method as before, create a new project named "SimpleHPF." Check your compiler to make sure the project was created properly.

3.11.2 SimpleHPF GUI

This simple plug-in will only have one slider that will control the value of the a_1 coefficient. The other coefficient is calculated from it. The specifications show that a_1 varies between 0 and 0.49. Set up the slider according to the properties in Table 3.2.

3.11.3 SimpleHPF.h File

To figure out what the CSimpleHPF object is going to have to do, first write the difference equation. Examine it and figure out which components are going to become coefficients and which are going to be memory locations. Also, figure out any intermediate variables you might need. You can figure out the difference equation by using the rules you learned in Chapter 1 to chart the input and output signals. Make sure you understand how this equation is formed from the block diagram in Figure 3.17.

The difference equation is as follows:

$$y(n) = a_0x(n) + a_1x(n - 1) \tag{3.4}$$

Next, figure out which block diagram components become variables in your C++ code. The coefficients $a_0$and a_1 will become a float member variable in the code. Even though we might be tempted to share the coefficients, these are separate left and right algorithms that have separate delay elements, so let's implement two sets, one each for the left and right channels.

Table 3.2: The slider properties for the SimpleHPF project.

Slider Property	Value
Control name	a1
Units	
Variable type	float
Variable name	m_fslider_a1
Low limit	0.0

High limit	0.49
Initial value	0

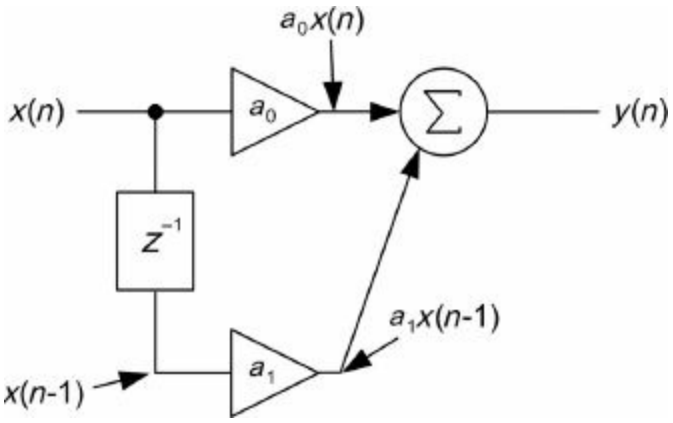


Figure 3.17: The HF tone control block diagram with annotations showing the signal math.

I named mine:

m_f_a0_left
m_f_a1_left
m_f_a0_right
m_f_a1_right

The z^{-1} element will also need to become a member variable and we will definitely need one for each channel because these can never be shared. I named mine

m_f_z1_left
m_f_z1_right

The slider will only modify its own m_fSlider_a1 value. We will calculate the values for the other coefficients using it. We will need to modify the userInterfaceChange() function just like the preceding example to wire the slider into the algorithm. Jump to your C++ compiler and go to the SimpleHPF.h file to add your member variables. Notice the variable that RackAFX added in the code below:

```
// 5. userInterfaceChange() occurs when the user moves a control.
virtual bool userInterfaceChange(int nControlIndex);

// Add your code here: ---------------------------------------------------- //
float m_f_a0_left;
float m_f_a1_left;

float m_f_a0_right;
float m_f_a1_right;

float m_f_z1_left;
float m_f_z1_right;

// END OF USER CODE ------------------------------------------------------- //

// ADDED BY RACKAFX -- DO NOT EDIT THIS CODE!!! ------------------------------ //
// **--0x07FD--**
float m_fSlider_a1;
// **--0x1A7F--**
// ------------------------------------------------------------------------ //
```

3.11.4 SimpleHPF.cpp File

Constructor

- Set our coefficient values to match the initialized slider settings.
- Calculate the new a_0 values.
- Clear out the z^{-1} variables.

```
CSimpleHPF::CSimpleHPF()
{
        <SNIP SNIP SNIP>

        // Finish initializations here
        m_f_a1_left = m_fSlider_a1;
        m_f_a1_right = m_fSlider_a1;

        m_f_a0_left = m_f_a1_left - 1.0;
        m_f_a0_right = m_f_a1_right - 1.0;

        m_f_z1_left = 0.0;
        m_f_z1_right = 0.0;
}
```

prepareForPlay()

```
bool __stdcall CSimpleHPF::prepareForPlay()
{
        // Add your code here:
        m_f_z1_left = 0.0;
        m_f_z1_right = 0.0;

        return true;
}
```

processAudioFrame()

The logic for the signal processing of *one channel* will be as follows:

- Read the delayed value $x(n-1)$ out of the z^{-1} element.
- Implement the difference equation.
- Write the current input $x(n)$ into the delay variable; it will be $x(n-1)$ next time around.
- Do this for both channels.

```
bool __stdcall CSimpleHPF::processAudioFrame(float* pInputBuffer, float*
pOutputBuffer, UINT uNumChannels)
{
	// Do LEFT (MONO) Channel
	//
	// Input sample is x(n)
	float xn = pInputBuffer[0];

	// READ: Delay sample is x(n-1)
	float xn_1 = m_f_z1_left;

	// Difference Equation
	float yn = m_f_a0_left*xn + m_f_a1_left*xn_1;

	// WRITE: Delay with current x(n)
	m_f_z1_left = xn;

	// Output sample is y(n)
	pOutputBuffer[0] = yn;
```

OK, now it's your turn to implement the other channel. Give it a try by yourself before proceeding. You should have something like this for the rest of the function:

```
	// Mono-In, Stereo-Out (AUX Effect)
	if(uNumInputChannels == 1 && uNumOutputChannels == 2)
		pOutputBuffer[1] = yn;

	// Stereo-In, Stereo-Out (INSERT Effect)
	if(uNumInputChannels == 2 && uNumOutputChannels == 2)
	{
		// Input sample is x(n)
		float xn = pInputBuffer[1];

		// Delay sample is x(n-1)
		float xn_1 = m_f_z1_right;

		// Difference Equation
		float yn = m_f_a0_right*xn + m_f_a1_right*xn_1;

		// Populate Delay with current x(n)
		m_f_z1_right = xn;

		// Output sample is y(n)
		pOutputBuffer[1] = yn;
	}

	return true;
}
```

Flush out delay elements in preparation for each play event in the plug-in. You generally do not want old data sitting inside these storage registers. The only exceptions are delay-looping effects where you exploit the old data. This is done in prepareForPlay().

userInterfaceChange()

- Store the new a_1 values.
- Cook the slider data to get the a_0 values.

```
bool __stdcall CSimpleHPF::userInterfaceChange(int nControlIndex)
{
	switch(nControlIndex)
	{
		case 0:
		{
			// save a1
			m_f_a1_left = m_fSlider_a1;
			m_f_a1_right = m_fSlider_a1;

			// calculate a0
			m_f_a0_left = m_f_a1_left - 1.0;
			m_f_a0_right = m_f_a1_right - 1.0;

			break;
		}
		default:
			break;
	}
	return true;
}
```

Build and load the project, open an audio file, and test your plug-in to make sure it's working properly. This plug-in is a simple low-cut filter and with the slider all the way down, you should hear no difference in the music. When you move the slider up, you will lose more and more bass frequencies, allowing only the high frequencies to pass. The cut-off frequency is approximately 11 kHz and the slider controls the amount of low frequency cut. You should easily be able to hear the effect, even over small speakers. Now, to get a little more information about the plug-in's operation, use RackAFX's analysis tools. Stop the music from playing with the transport control. Launch the analyzer window by clicking on the Analyzer button in the toolbar or choose View > Analyzer. The analyzer pops up as shown in Figure 3.18 (yours may look slightly different).

The analyzer is a powerful tool for checking your plug-in's performance. The basic controls are

1. Scope/spectrum analyzer.
2. Basic graphing options.
3. Scope controls.
4. Real-time response buttons, which let you measure the frequency, phase, impulse, and step responses of the plug-in (audio must not be streaming to use these).

Click on the Frequency button and you will get a frequency response plot of the filter. Move the slider all the way down and you should get a flat response, as shown in Figure 3.19. If you move the slider all the way up so that $a_1 = 0.49$, you get the response in Figure 3.20.

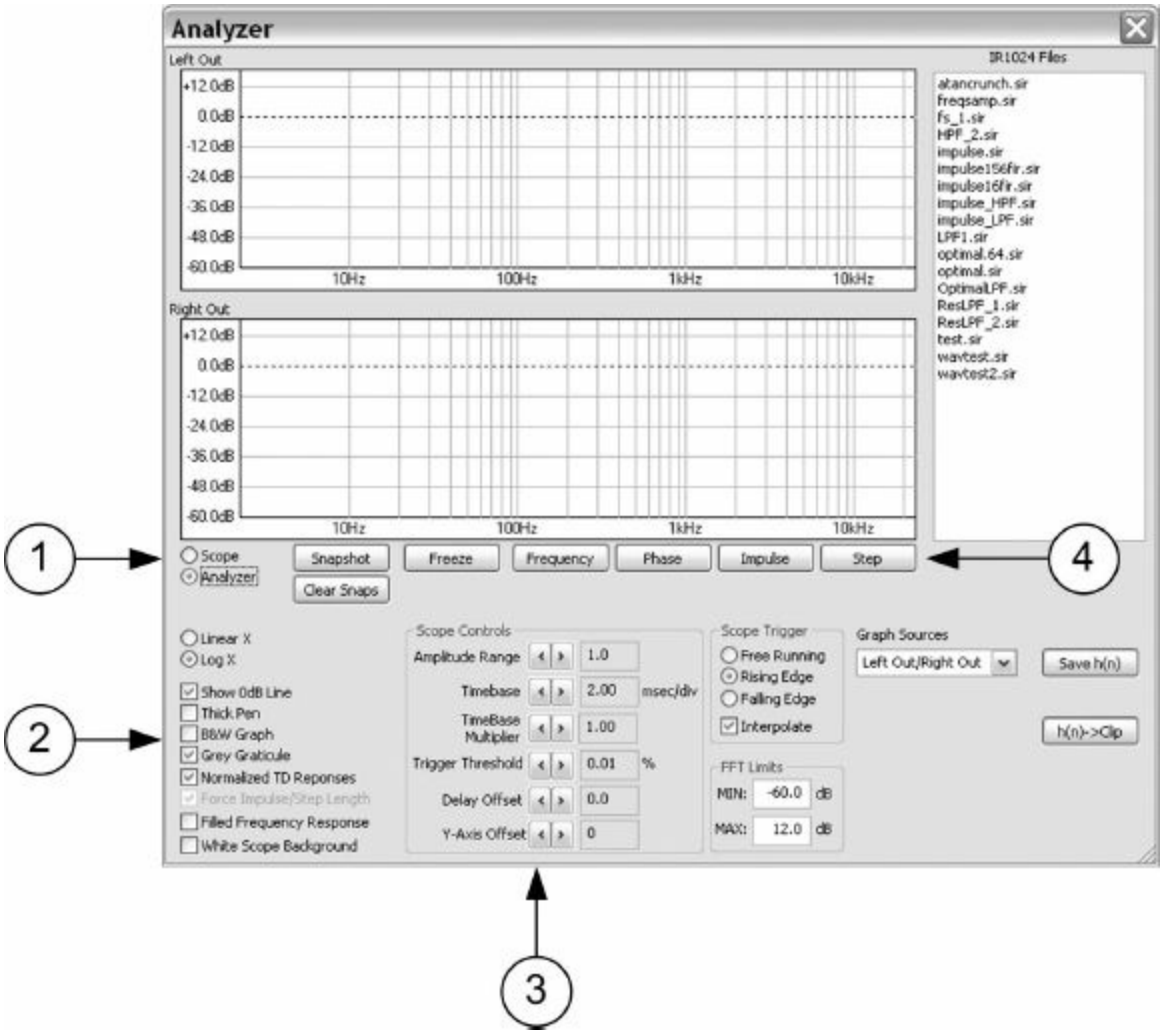


Figure 3.18: The audio analyzer.

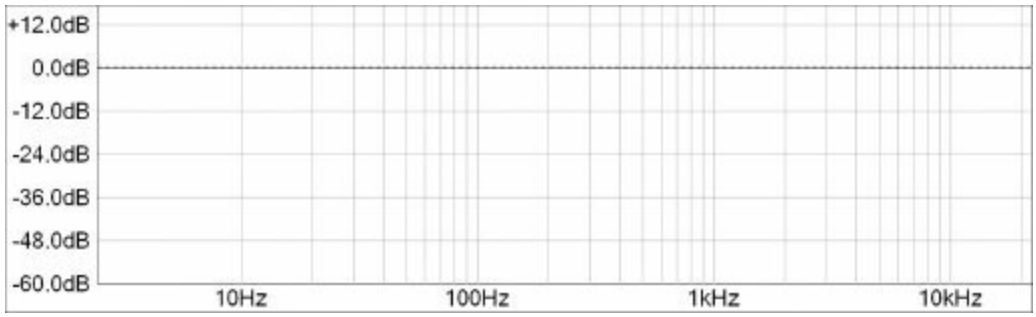


Figure 3.19: A flat frequency response with $a_1 = 0.0$.

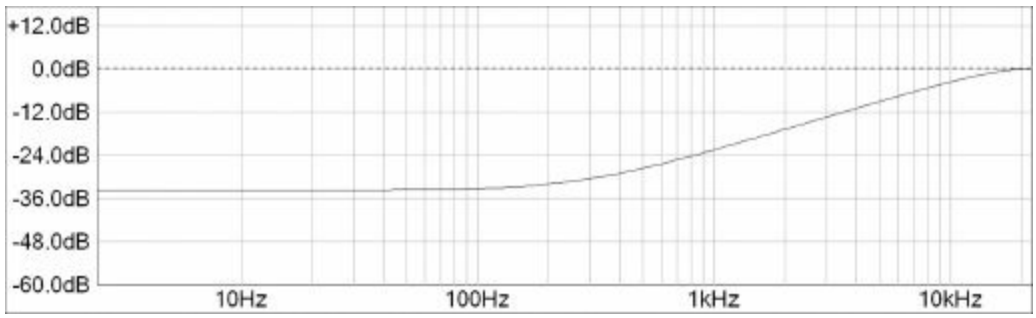


Figure 3.20: A filter that appears to boost high frequencies. You can see that it is really cutting the low frequencies instead; this is something you might only have realized by using the audio analyzer.

3.12 Design a High-Frequency Tone Control with Volume Plug-In

This final example will show you how to deal with more than one slider control by simply adding a volume-in-dB control to the block diagram. The plan is to add another slider to the existing plug-in; the new slider will control the overall volume of the plug-in in dB. You already know how to implement both parts of it, so this exercise is really more about adding new controls to an existing project.

3.12.1 Project: SimpleHPF

Open your SimpleHPF project in RackAFX using the Open button or the menu/toolbar items.

3.12.2 SimpleHPF GUI

Add the new volume slider: Right-click on the second slider group and add a new slider for the volume control, in dB, and use the exact same variable name and settings as in the VolumedB project. You should end up with a GUI like that in Figure 3.21.

3.12.3 SimpleHPF.h File

Add your own m_fVolume variable to handle the cooked volume data, just as before.

```
// Add your code here: ------------------------------------------------------ //
	float m_f_a0_left;
	float m_f_a1_left;

	float m_f_a0_right;
	float m_f_a1_right;

	float m_f_z1_left;
	float m_f_z1_right;

	float m_fVolume;
```

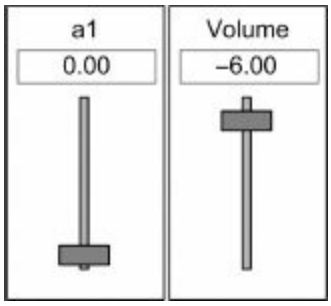


Figure 3.21: The new SimpleHPF GUI with added volume-in-dB control.

```
// END OF USER CODE ------------------------------------------------------------ //

// ADDED BY RACKAFX -- DO NOT EDIT THIS CODE!!! ------------------------------- //
//  **--0x07FD--**

float m_fSlider_a1;
float m_fVolume_dB;

// **--0x1A7F--**
// ------------------------------------------------------------------------------ //
```

3.12.4 SimpleHPF.cpp File

Constructor

- Cook the volume data after the filter initializations.

```
CSimpleHPF::CSimpleHPF()
{
        <SNIP SNIP SNIP>
        m_f_a0_left = -1.0;
        m_f_a1_left = 0.0;

        m_f_a0_right = -1.0;
        m_f_a1_right = 0.0;

        m_f_z1_left = 0.0;
        m_f_z1_right = 0.0;

        m_fVolume = pow(10.0, m_fVolume_dB/20.0);
}
```

prepareForPlay()

There is nothing to add here because the volume variable does not need to be reset on each play event.

processAudioFrame ()

- Add the volume control scaling *after* the filtering operation.

```
// Do LEFT (MONO) Channel; there is always at least one input/one output
// (INSERT Effect)
// Input sample is x(n)
float xn = pInputBuffer[0];

// READ: Delay sample is x(n-1)
float xn_1 = m_f_z1_left;

// Difference Equation
float yn = m_f_a0_left*xn + m_f_a1_left*xn_1;

// WRITE: Delay with current x(n)
m_f_z1_left = xn;

// Output sample is y(n)
pOutputBuffer[0] = yn*m_fVolume;

// Mono-In, Stereo-Out (AUX Effect)
if(uNumInputChannels == 1 && uNumOutputChannels == 2)
	pOutputBuffer[1] = yn*m_fVolume;

// Stereo-In, Stereo-Out (INSERT Effect)
if(uNumInputChannels == 2 && uNumOutputChannels == 2)
{
	// Input sample is x(n)
	float xn = pInputBuffer[1];

	// Delay sample is x(n-1)
	float xn_1 = m_f_z1_right;

	// Difference Equation
	float yn = m_f_a0_right*xn + m_f_a1_right*xn_1;

	// Populate Delay with current x(n)
	m_f_z1_right = xn;

	// Output sample is y(n)
	pOutputBuffer[1] = yn*m_fVolume;
}
```

userInterfaceChange()

- Cook the volume data.
- Make sure you check your control ID values in case you chose different sliders than I did.

```
bool __stdcall CSimpleHPF::userInterfaceChange(int nControlIndex)
{
	// decode the control index
	switch(nControlIndex)
	{
		case 0:
		{
			m_f_a1_left = m_fSlider_a1;
			m_f_a1_right = m_fSlider_a1;

			m_f_a0_left = m_f_a1_left - 1.0;
			m_f_a0_right = m_f_a1_right - 1.0;

			break;
		}
		case 1:
		{
			m_fVolume = pow(10.0, m_fVolume_dB/20.0);
			break;
		}
		default:
			break;
	}
	return true;
}
```

Build and test your code to make sure the plug-in works as expected. You now understand the basics of writing a plug-in from a block diagram and design equations.

Add some presets

Now that you have a GUI with more than one control, try adding some presets. If you look at the preset list in the toolbar, you will see the first one is named Factory Preset. This preset contains your initial settings for the GUI controls. You cannot delete this preset and it is automatically updated whenever you add, edit, or delete controls. This preset takes you back to your initial state. With your plug-in loaded and (optionally) audio streaming through it, adjust the controls to affect the signal. Hit the Save Preset button on the toolbar or choose it from the plug-in menu. A box will pop up allowing you to name the preset. You can also overwrite an existing preset. You can store up to 32 presets in your plug-in.

3.13 The User Plug-In Menu in RackAFX

As you write more plug-ins, you will notice that they begin to automatically populate the user plug-in menu item in RackAFX. By now, you should have three plug-ins in this menu. There are a few things you need to understand about this menu.

- It allows you to play with the plug-ins without having to open your compiler and manually load and unload the DLLs.
- You can select different plug-ins from this menu while audio is streaming and they will automatically slot in and out, so you can audition or show off your plug-ins quickly.
- It allows RackAFX to behave just like other plug-in clients by loading all the DLLs it finds in its PlugIns folder at once when you start the software. *This can be dangerous!*

That last item poses a problem during the course of development—if you write a DLL that does bad

things in the constructor, such as hold pointers with garbage values or try to access memory that hasn't been allocated, it may crash RackAFX when it first starts up. If your DLL behaves really badly, you might even wound the OS too. This is a difficult issue to avoid without complicated rules for commissioning and decommissioning the plug-in. Additionally, you will have the same problem if you are developing a commercial plug-in and you are using a third-party client; most of these are designed to first open all the DLLs in their plug-in folder and check to make sure the plug-ins can be instantiated. If you write a bad DLL, you might also crash these clients and/or the OS. In at least one commercially available client, if your plug-in crashes during startup, it will not be loaded again in future launches. When RackAFX loads your DLL, it does some error checking to try to make sure your plug-in is legal, but it can't check the validity of your construction code.

If RackAFX crashes each time you open it, remove the last DLL you were working on from the PlugIns folder. Alternatively, you can remove all the DLLs—you will want to copy them and restore them later when you find the bad DLL that caused the crashing. Writing a DLL is challenging and fun, but since you are writing a component, you can wind up with crashes like this.

CHAPTER 4

How DSP Filters Work

During the course of this book you will learn how to implement the following effects:

- EQs/tone controls
- Delay
- Flanger/chorus
- Compressor/limiter/tremolo
- Reverb
- Modulated filters/phaser

The EQ/tone control theory is the most difficult of all the effects to explain in a simple way. These effects are based on DSP filter theory which involves complex algebra, that is, the algebra of complex numbers. Complex numbers contain real and imaginary parts. There are two basic ways to explain basic DSP theory. The first is intuitive and involves no complex math but requires some bookkeeping and can be tedious. The second method uses complex algebra to solve the problem. We'll start with the intuitive method, and then (optionally, if you desire) make the leap into complex math. Don't worry—we don't want to get mired down in theory and forget the fun part, which is making audio effects. If you skip the second part, you will still be able to code EQ and tone control plug-ins, but you will better understand where the equations come from if you know a little theory too. The transfer functions you will learn along the way will reappear in many effects algorithms.

In Figure 4.1 you can see the −6 dB/octave roll-off indicative of a first-order filter. A digital version of the analog filter should have the same shape and roll-off. A key difference is that the digital filter will not operate beyond the Nyquist frequency.

The phase response plot shows the relative phases of different frequencies upon exiting the filter. During the filtering process, the phases of different frequencies get shifted forward or backward. In Figure 4.2, the 1 kHz frequency is shifted by −45 degrees compared to the input. At very high frequencies, the phase shift approaches −90 degrees. This phase shift is not a side effect of the filtering but an integral part of how it works.

To understand the ramifications of the phase shifting, consider a complex waveform entering the filter. Fourier showed that a complex, continuous waveform could be decomposed into a set of sinusoids with different frequencies and amplitudes. Figure 4.3 shows a filter in action. The input is decomposed into four sinusoids, a fundamental, and three harmonics. The peak-amplitudes of the components are shown as dark bars to the left of the *y* axes.

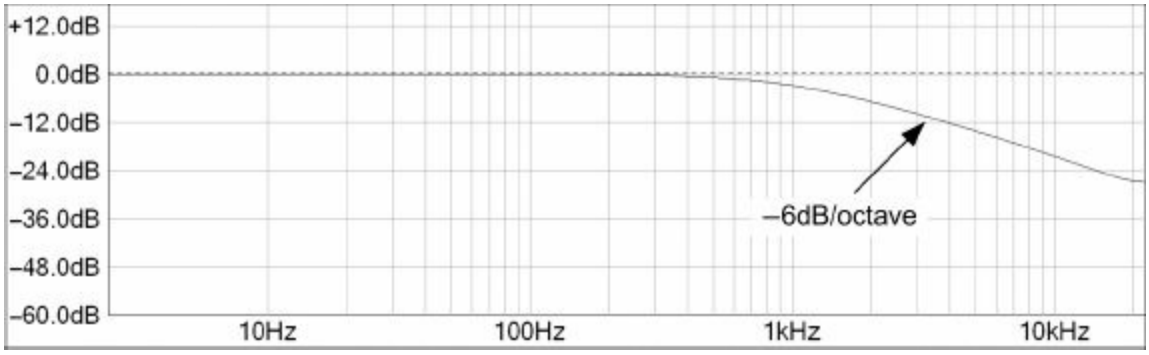


Figure 4.1: The fundamental analysis tool for a DSP filter is its frequency response plot. This graph shows how the filter amplifies or attenuates certain bands of frequencies.

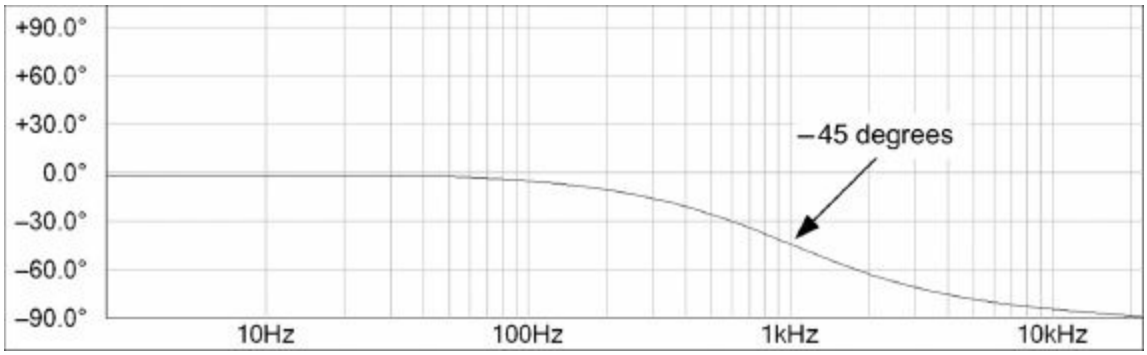


Figure 4.2: The phase response plot of the analog filter in Figure 4.1 shows how the phase of the output is shifted across frequencies.

We observe several features of the output:

- The composite waveform is smoothed out.
- The amplitude and phase shift of the fundamental are unchanged.
- The three harmonics have decreasing amplitudes and more phase shift as you go higher in frequency.

Figure 4.4 shows the same filter with the information plotted differently; here, the amplitudes and phases are plotted against frequency rather than dealing with a set of time-domain sinusoids. You can see by the output frequency response plot that this filter is a kind of low-pass filter. Its curve is similar to Figure 4.1, the analog equivalent.

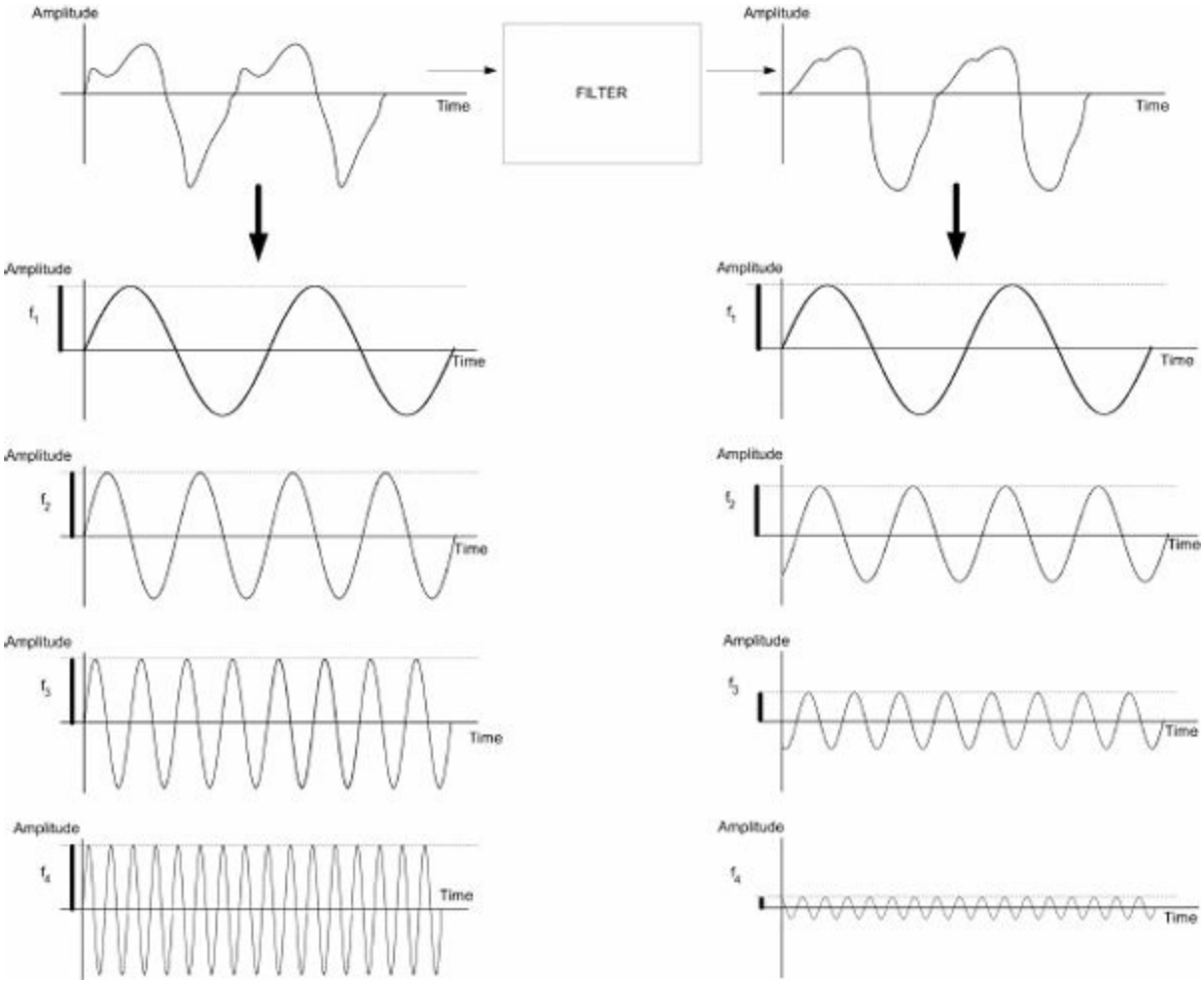


Figure 4.3: A complex waveform is filtered into a smoothed output. The input and output are decomposed into their Fourier-series components.

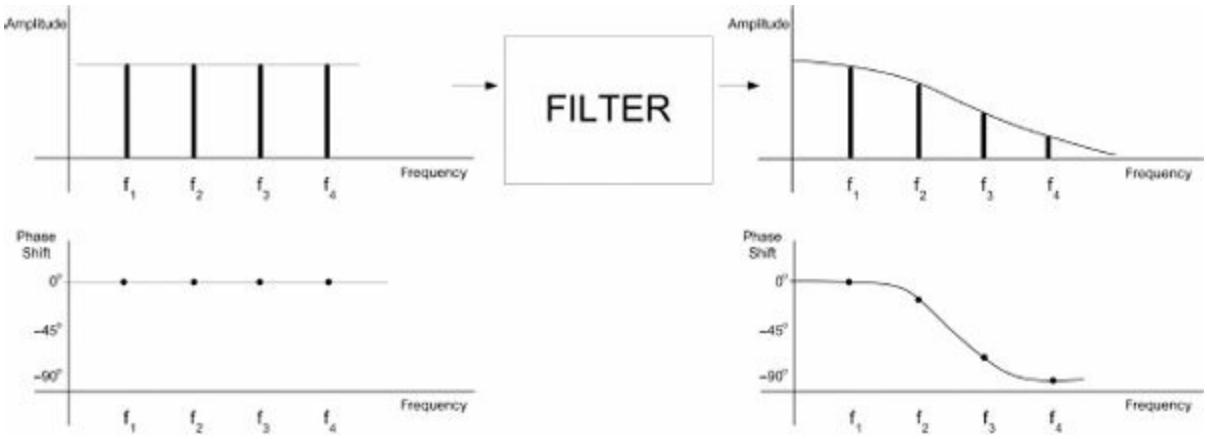


Figure 4.4: The same information is plotted as frequency and phase responses.

4.1 First-Order Feed-Forward Filter

In order to get a grip on the nature of digital filtering, start with the first-order feed-forward filter shown in a block diagram in Figure 4.5. You've already seen a version of it in the HPF tone control you coded in the last chapter.

The difference equation is as follows:

$$y(n) = a_0x(n) + a_1x(n - 1) \tag{4.1}$$

You can tell why it's called feed forward—the input branches feed forward into the summer. The signal flows from input to output. There is no feedback from the output back to the input. Now, suppose we let the coefficients a_0 and a_1 both equal 0.5 in Figure 4.6.

In order to analyze this filter we can go the easy but tedious route or the difficult but elegant route. Let's start with the easy way. In order to figure out what this does, you apply the five basic digital test signals you learned in Chapter 1 to the filter and then manually push the values through and see what comes out. You only need a pencil and paper or a simple calculator. The five waveforms we want to test are:

1. DC (0 Hz)
2. Nyquist
3. ½ Nyquist
4. ¼ Nyquist
5. Impulse

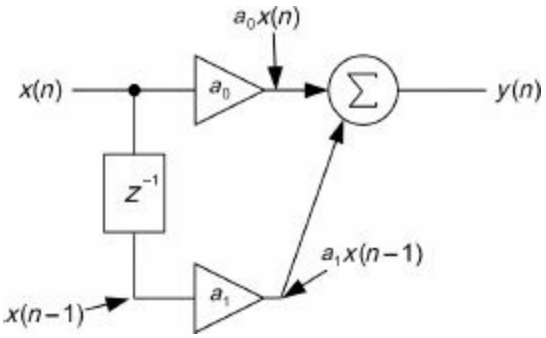


Figure 4.5: The first-order feed-forward filter.

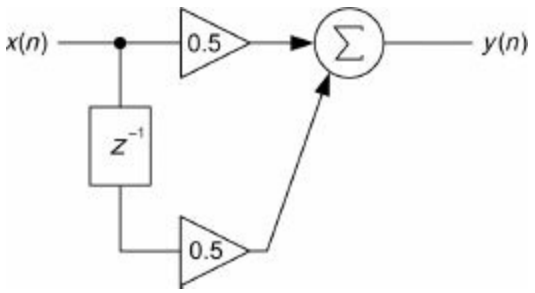


Figure 4.6: What kind of filter is this? What are its frequency and phase responses?

For each audio sample that enters there are two phases to the operation:

1. Read phase: The sample is read in and the output is formed using the difference equation and the previous sample in the delay register.
2. Write phase: The delay element is overwritten with the input value—the sample stored in the single z^{-1} register is effectively lost.

Start with the DC/step input and begin sequentially applying the samples into the filter shown in Figures 4.7 through 4.10.

Now, observe the amplitude and phase shift of the input versus output—the output amplitude

eventually settles out to a constant 1.0 or unity gain condition, so at DC or 0 Hz, the output equals the input. However, there is a one sample delay in the response, causing the leading edge of the step-input to be smeared out by one sample interval. This time smearing is a normal consequence of the filtering.

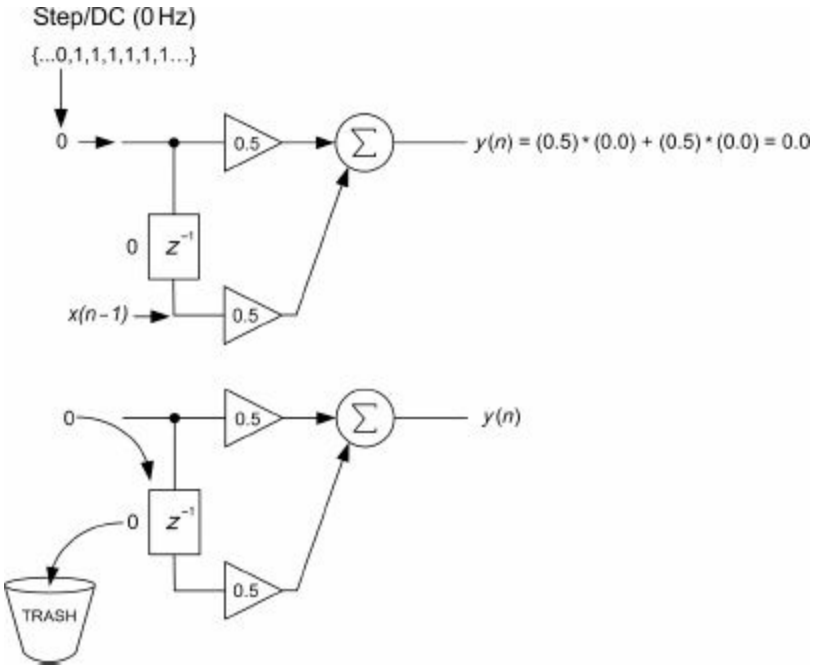


Figure 4.7: On the first iteration the input sample is used in the difference equation to create the output, $x(n) = 0$, $y(n) = 0$ and then the input is shifted into the delay register.

In a feed-forward filter, the amount of time smearing is equal to the maximum delayed path through the feed-forward branches.

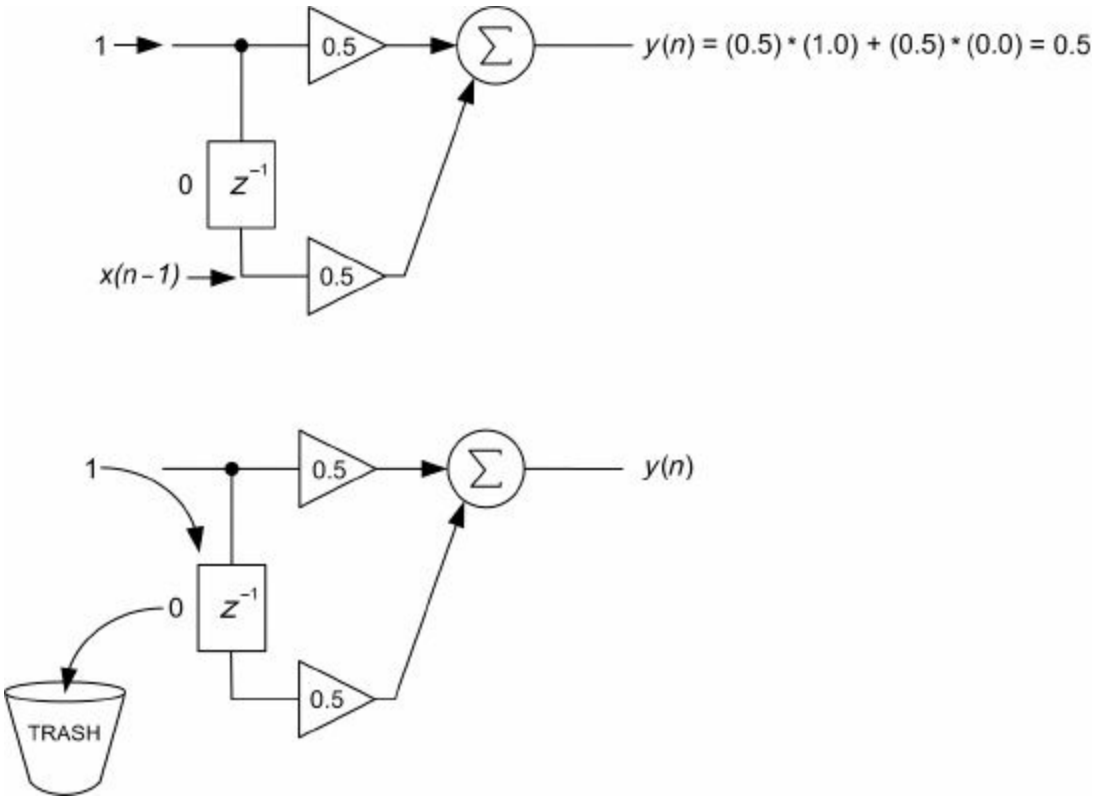


Figure 4.8: The process continues with each sample. Here the input 1.0 is combined with the previous input; the second output $y(n) = 0.5$.

Next, repeat the process for the Nyquist frequency (DC and Nyquist are the easiest, so we'll do them first). The filter behaves in an entirely different way when presented with Nyquist (Figures 4.11 through 4.14).

Now, make your observations about amplitude and phase. The amplitude at Nyquist eventually becomes zero after the one-sample-delay time. The phase is hard to tell because the signal has vanished. Why did the amplitude drop all the way to zero at Nyquist? The answer is one of the keys to understanding digital filter theory: the one-sample delay introduced exactly 180 degrees of phase shift at the Nyquist frequency and caused it to cancel out when recombined with the input branch through a_0.

Delay elements create phase shifts in the signal. The amount of phase shift depends on the amount of delay as well as the frequency in question.

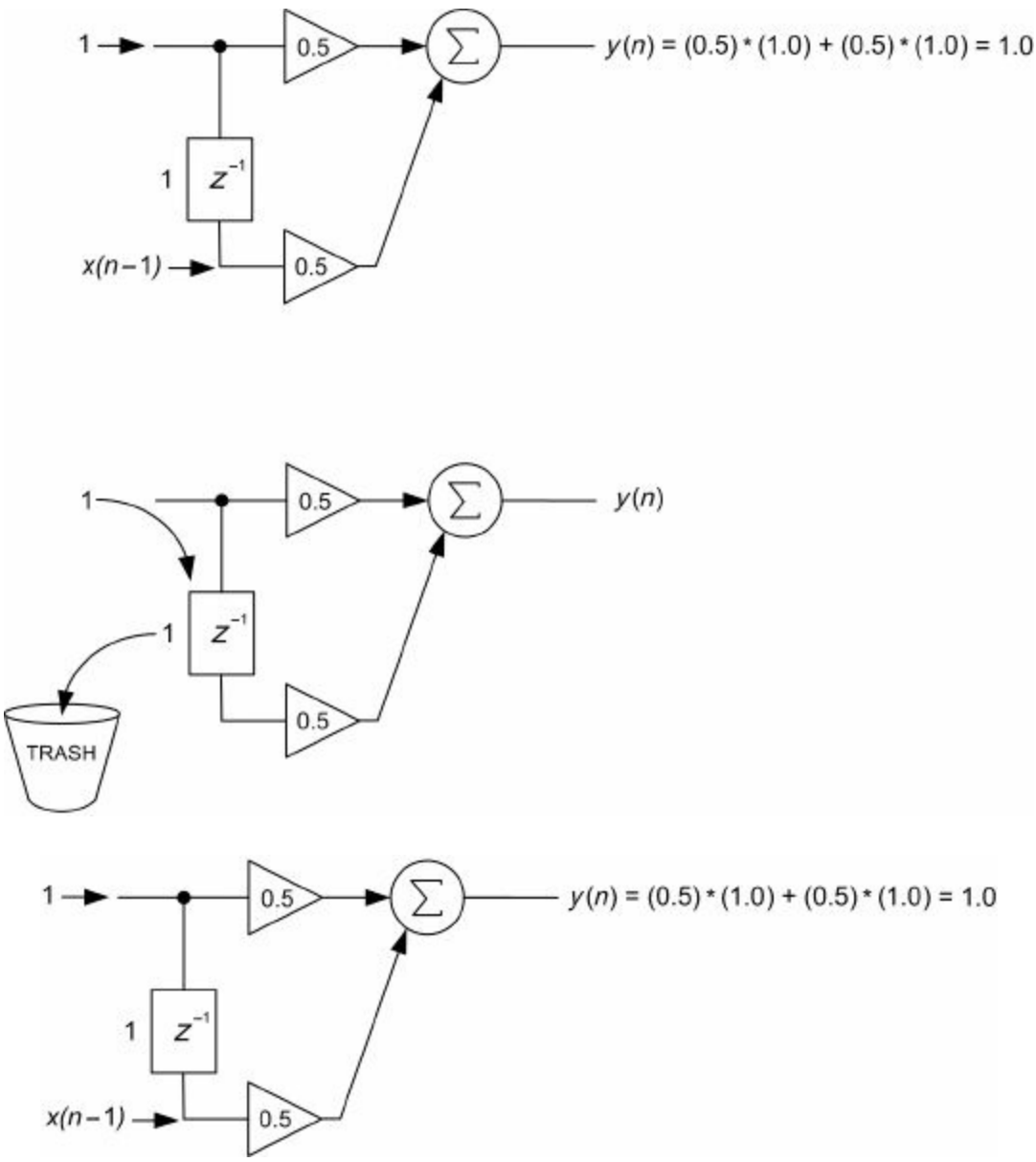


Figure 4.9: The sequence continues until we observe a repeating pattern; 1.0 is repeating here.

In the case of Nyquist, the one-sample delay is exactly enough to cancel out the original signal when

they are added together in equal ratios. What about other frequencies like ½ and ¼ Nyquist? They are a bit more laborious to work through but worth the effort. By now you can see how the data moves through the filter, so let's use a table (Table 4.1) and move the data through *it* instead. The ½ Nyquist sequence is $x(n) = \{\ldots 0, +1.0, 0.0, -1.0, 0.0, +1.0, 0.0, \ldots\}$.

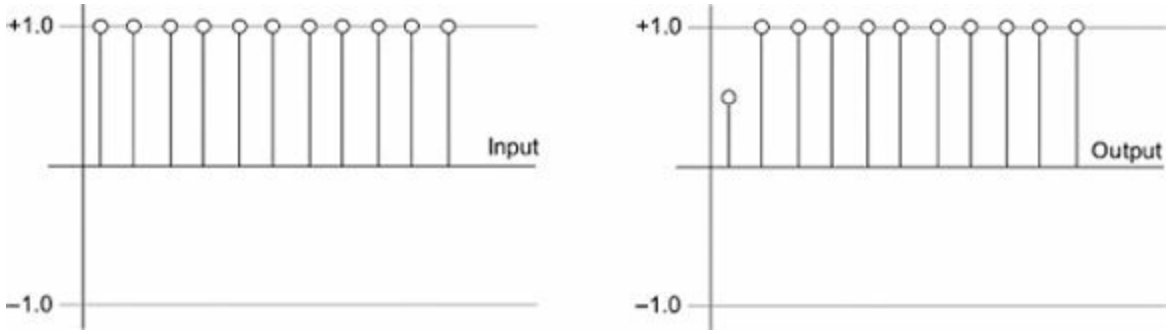


Figure 4.10: The input and output sequences for the filter in Figure 4.6 at DC or 0 Hz.

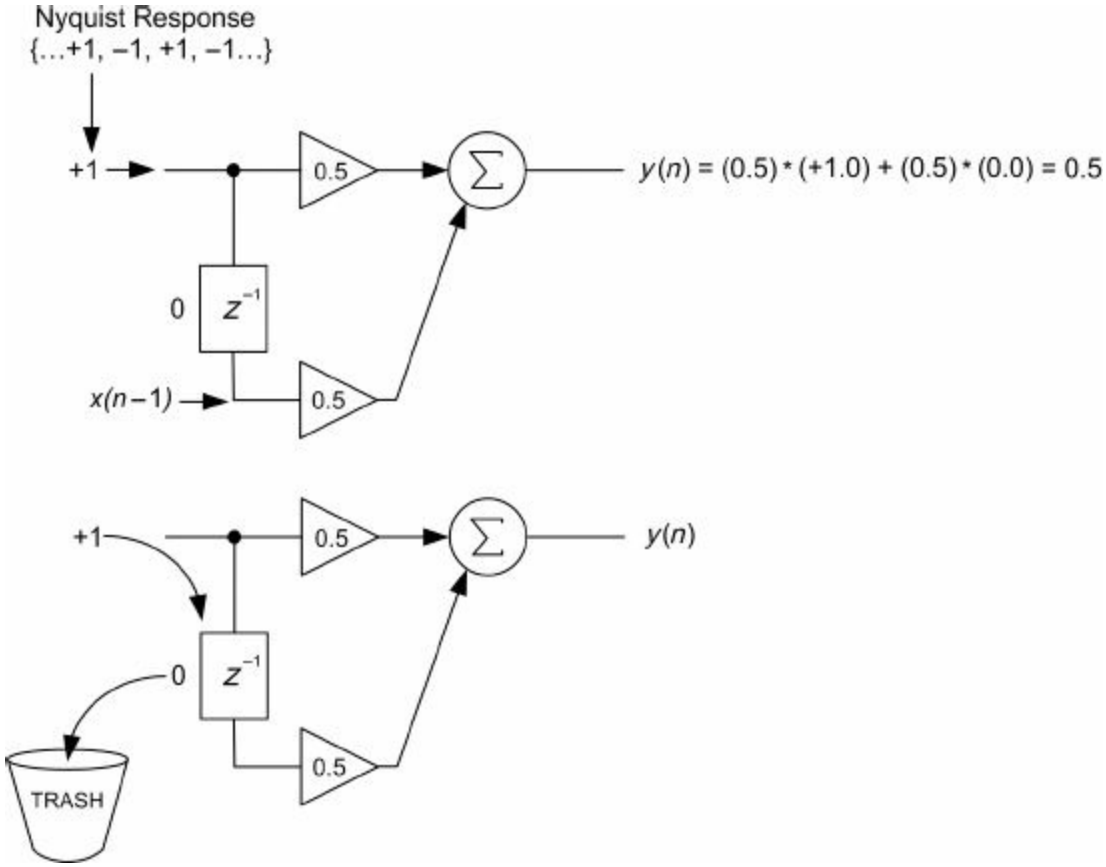


Figure 4.11: The Nyquist sequence is applied to the filter. Notice how the delay element has been zeroed out. The output for the first iteration is $y(n) = 0.5$.

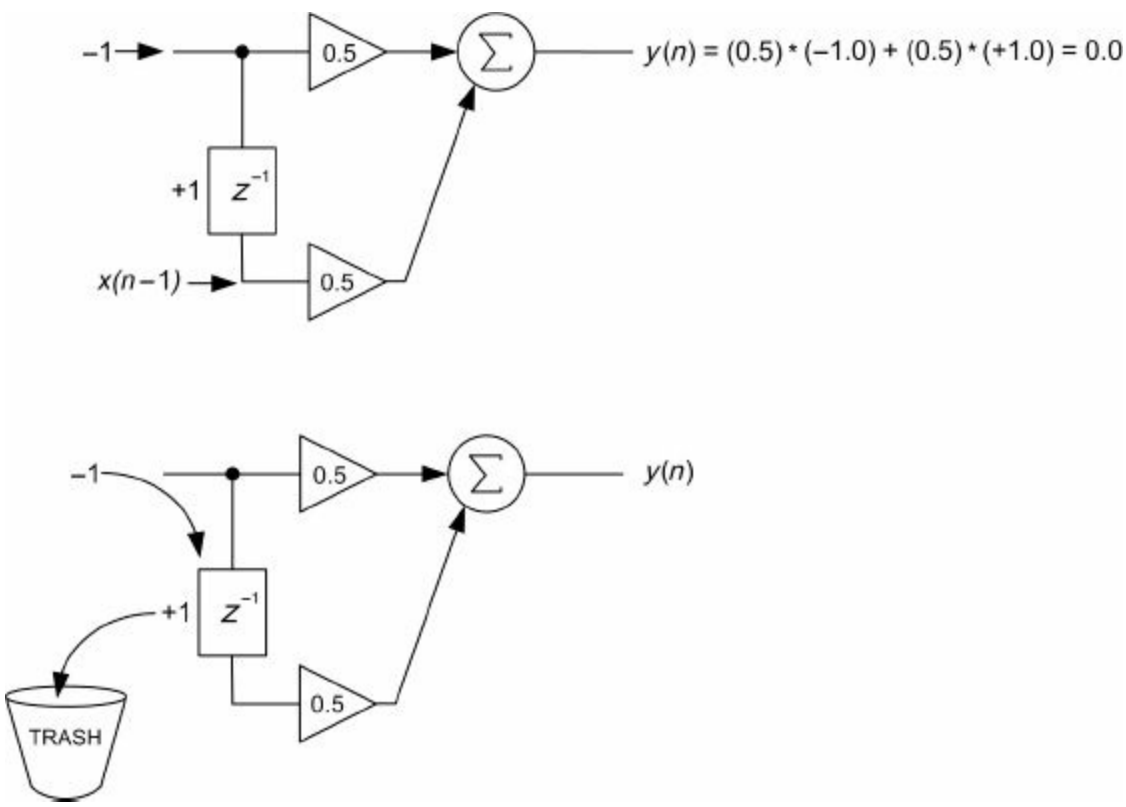


Figure 4.12: The second iteration at Nyquist produces an output $y(n) = 0$.

Table 4.1: The manual labor continues as we work through the ½ Nyquist frequency.

$x(n)$	$x(n-1)$	$y(n) = 0.5x(n) + 0.5x(n-1)$
0	0	0
1	0	0.5
0	1	0.5
−1	0	−0.5
0	−1	−0.5
1	0	0.5
0	1	0.5
−1	0	−0.5
0	−1	−0.5

Can you see how $x(n)$ becomes $x(n-1)$ for the next row? The $x(n-1)$ column holds a one-sample-delayed version of the input $x(n)$. The output is $y(n) = \{0, +0.5, +0.5, -0.5, -0.5, +0.5, +0.5\}$.

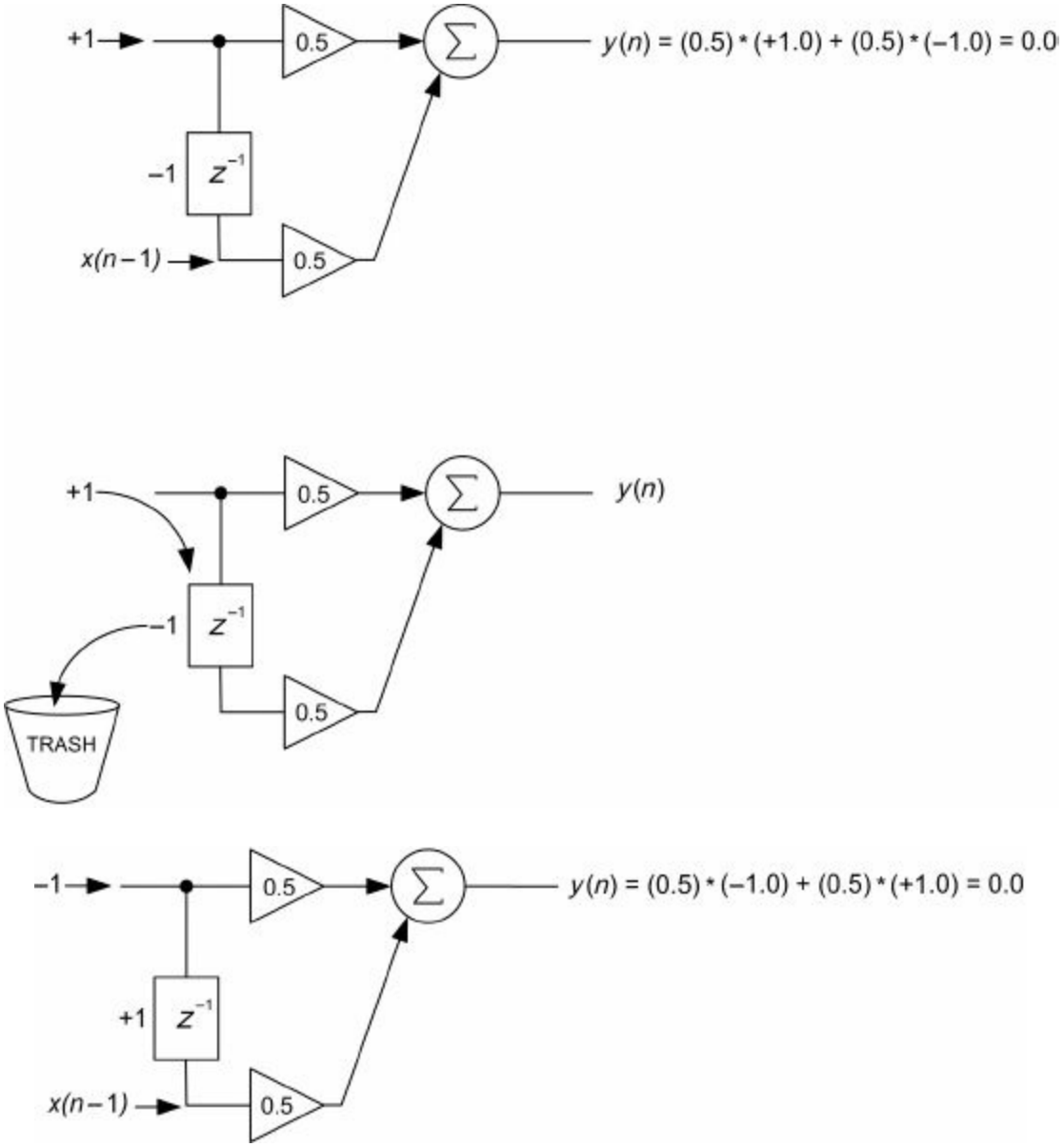


Figure 4.13: Continuing the operation at Nyquist, we see that eventually the output settles to a repeating 0, 0, 0, 0 sequence.

Next we observe the amplitude and phase relationship from input to output in Figure 4.15. At first it might seem difficult to figure out the sequence {…−0.5, −0.5, +0.5, +0.5, …}. ½ Nyquist is also encoded with a repeating sequence of four values (0, 1, 0, −1).

Work through ¼ Nyquist the same way (Table 4.2). The ¼ Nyquist frequency sequence is $x(n)$ = {0, 0.707, 1, 0.707, 0, −0.707, −1, −0.707, 0, …}.

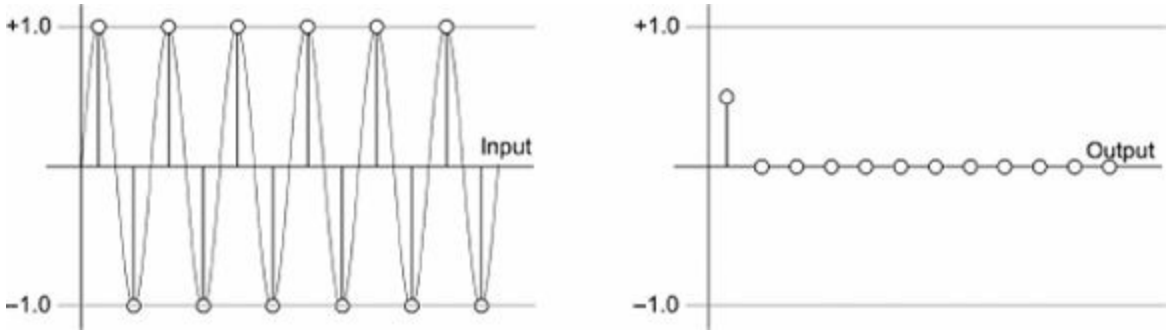


Figure 4.14: The input and output sequences for the filter in Figure 4.6 at Nyquist.

Table 4.2: ¼ Nyquist input/output.

$x(n)$	$x(n-1)$	$y(n) = 0.5x(n) + 0.5x(n-1)$
0	0	0
0.707	0	+0.354
1	0.707	+0.854
0.707	1	+0.854
0	0.707	+0.354
−0.707	0	−0.354
−1	−0.707	−0.854
−0.707	−1	−0.854
0	−0.707	−0.354

The output is $y(n) = \{\ldots+0.354, +0.854, +0.854, +0.354, -0.354, -0.854, -0.854, -0.354, +0.354, \ldots\}$. Analysis of the output sequence reveals the phase-shifted and slightly attenuated output signal at ¼ Nyquist. Both the phase shift and the attenuation are smaller than ½ Nyquist. As you can see in Figure 4.16 there is also one sample of time smearing at the start of the signal.

Finally, apply the impulse sequence and find the impulse response. The impulse response is the third analysis tool. The impulse response defines the filter in the time domain like the frequency response defines it in the frequency domain. The basic idea is that if you know how the filter reacts to a single impulse you can predict how it will react to a series of impulses of varying amplitudes. Take a Fast Fourier Transform (FFT) of the impulse response and you get the frequency response. An inverse FFT converts the frequency response back into the impulse response. For this filter the impulse response is simple (Table 4.3).

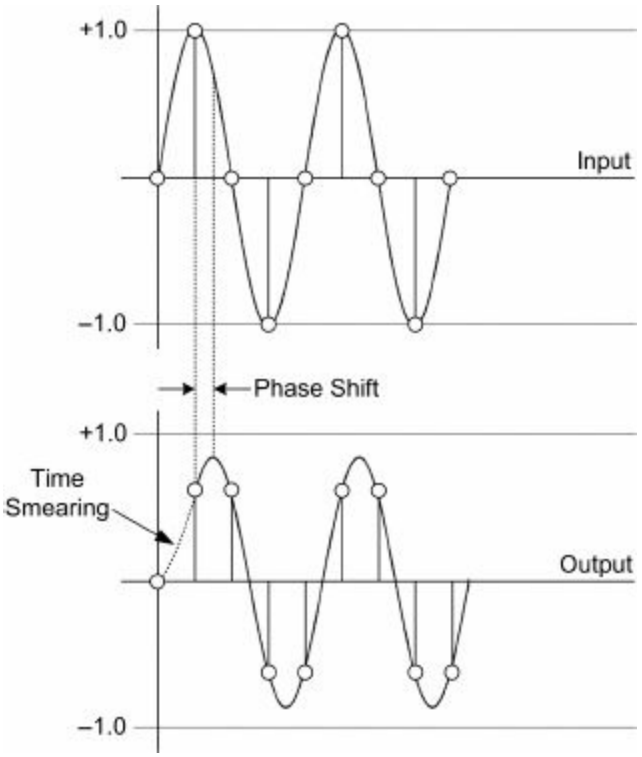


Figure 4.15: The input/output relationship in time at ½ Nyquist. The ½ Nyquist frequency is attenuated almost by one-half. The output is also phase shifted by 45 degrees. The leading edge of the first cycle is smeared out by one sample's worth of time.

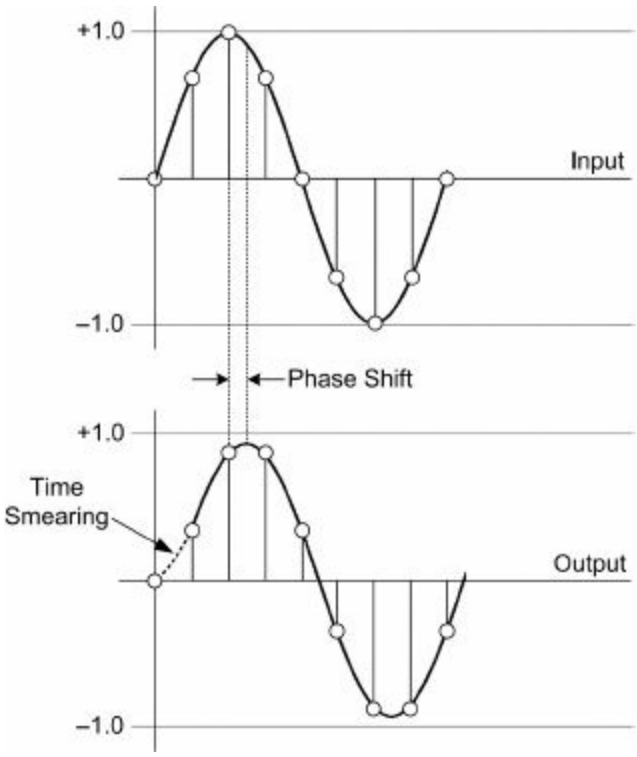


Figure 4.16: The input/output relationship at ¼ Nyquist.

Table 4.3: The impulse response input/output relationship.

$x(n)$	$x(n-1)$	$y(n) = 0.5x(n) + 0.5x(n-1)$
0	0	0
1	0	0.5
0	1	0.5
0	0	0
0	0	0
0	0	0

Here you can see that the impulse is flattened and smeared out. It is actually two points on a $\sin(x)/(x)$-like curve, as shown in Figure 4.17. Now, you can combine all the frequency amplitude and phase values into one big graph, as shown in Figure 4.18.

We observe that this digital filter is a low-pass variety with a typical low pass filter (LPF) magnitude response. However the phase response is quite interesting—it is linear instead of nonlinear like the analog example at the beginning of the chapter. In fact, this simple filter is a *linear phase filter*.

Figures 4.19 and 4.20 show the measured frequency and phase responses for this filter. Compare it with our estimation. Notice the linear phase produces a straight line only when plotted on the linear

frequency axis. What makes this filter a low-pass filter? It is a combination of the coefficients and the filter topology (first-order feed-forward). There are three basic topologies: feed forward (FF), feed back (FB), and a combination of FF/FB. Once the topology has been chosen, it's really the *coefficients* that determine what the filter will do.

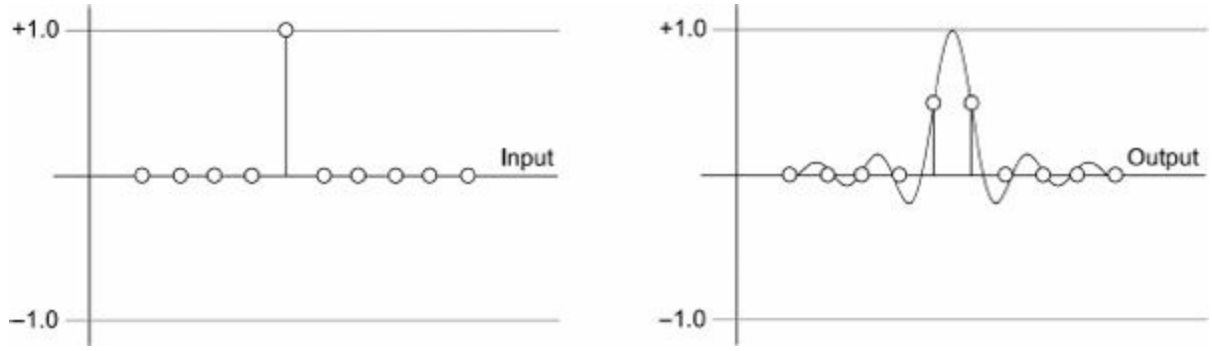


Figure 4.17: The time domain plots of the impulse response input/output.

A feed-forward filter will be a linear phase filter if its coefficients are symmetrical about their center. In this case (0.5, 0.5) is symmetrical. Another example would be (−0.25, 0, −0.25).

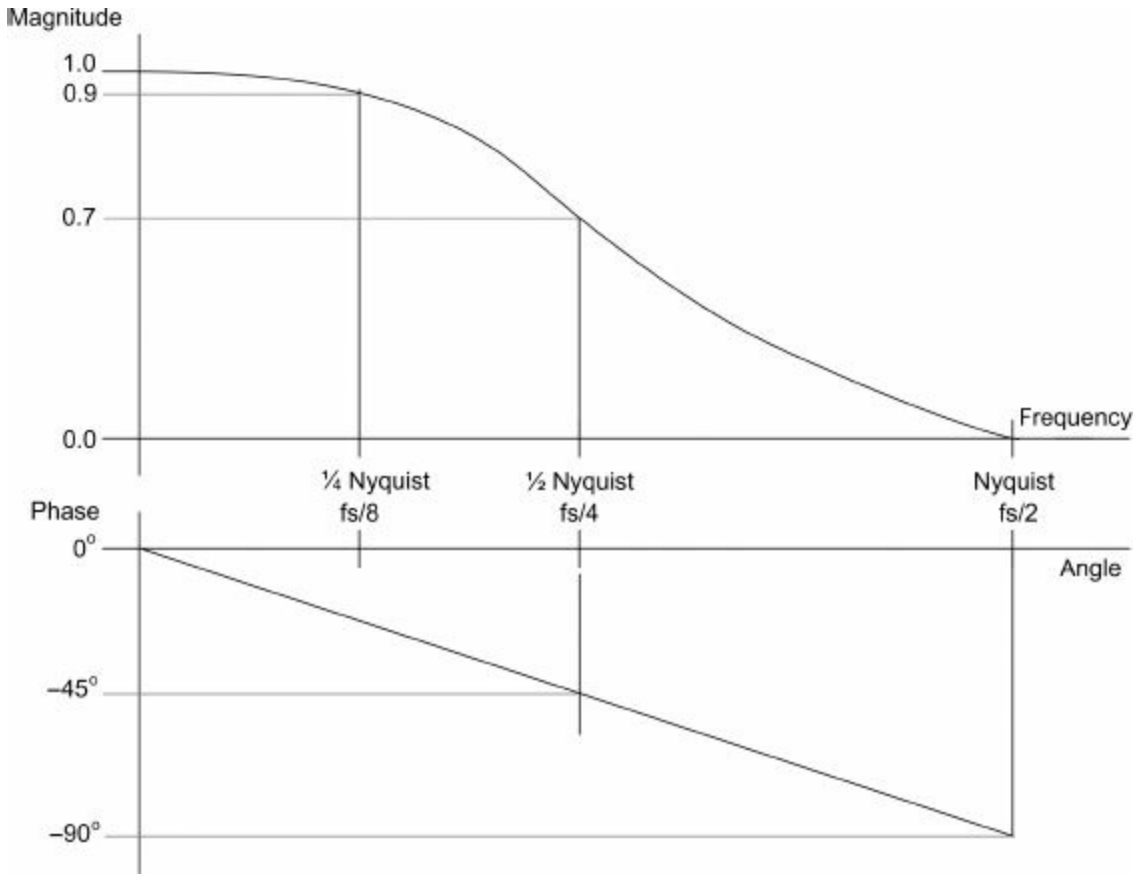


Figure 4.18: Final frequency and phase response plots for the digital filter in Figure 4.6. Notice that this is a linear frequency plot since ½ Nyquist is halfway across the *x*-axis. The phase at Nyquist pops back up to 0 degrees since there is no Nyquist component present (output = 0).

4.2 Design a General First-Order Feed-Forward Filter

To illustrate this and crystallize it as a concept, modify your SimpleHPF filter so that two sliders control the a_0 and a_1 coefficients directly. Also, alter the range of values they can control to (−1.0 to +1.0). Then, experiment with the two values and watch what happens in the analyzer. How to do this is described next.

Open your SimpleHPF project and modify the user interface (UI). First, change the values for the a_1 slider to match the new low and high limits. As usual, you right-click inside the slider's bounding box and alter the limits and initial value (shown in bold) as in Table 4.4.

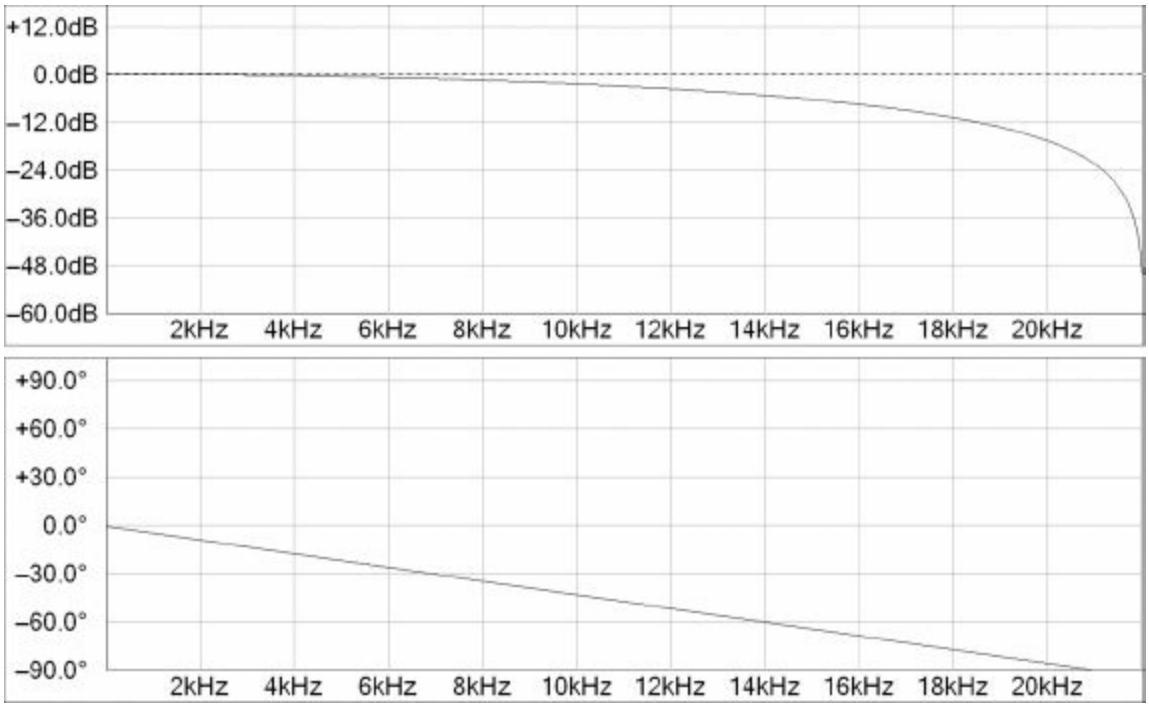


Figure 4.19: Measured frequency and phase response plots for the filter you just analyzed by hand. These are plotted with a linear frequency base.

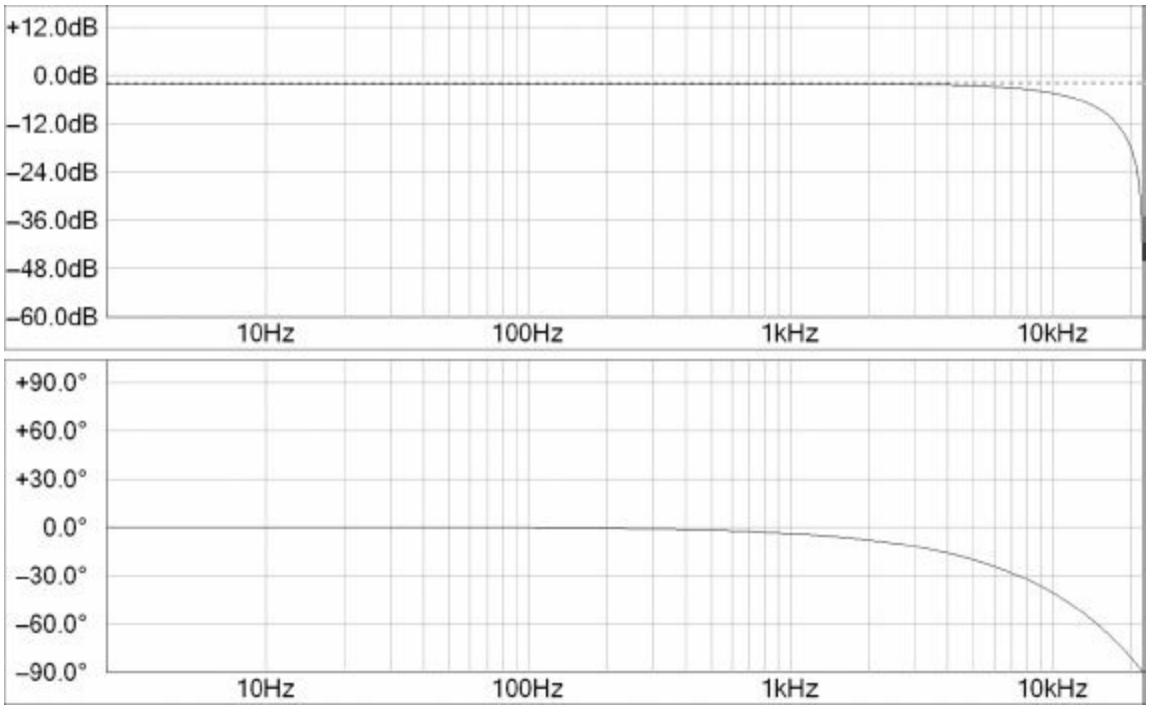


Figure 4.20: Measured frequency and phase response plots with log frequency base.

Table 4.4: The altered a_1 slider properties.

Slider Property	Value
Control Name	a1
Units	
Variable Type	float
Variable Name	m_fSlider_a1
Low Limit	−1.0
High Limit	+1.0
Initial Value	0.0

Now add a new slider for the a_0 coefficient just below the a_1 slider (Table 4.5).

Table 4.5: The new a_0 slider properties.

Slider Property	Value
Control Name	a0
Units	
Variable Type	float
Variable Name	m_fSlider_a0
Low Limit	−1.0
High Limit	+1.0
Initial Value	1.0

Change the userInterfaceChange() function to directly map the slider values to the coefficient values. The new slider has a control ID of 10; always check your nControlIndex value since it might be different depending on your UI.

```
        Variable Name                   Index
-----------------------------------------------
        m_fSlider_a1                    0
        m_fVolume_dB                    1
        m_fSlider_a0                    10

-----------------------------------------------
bool CSimpleHPF::userInterfaceChange(int nControlIndex)

{
        switch(nControlIndex)
        {
```

```
        case 0:
                // direct map to the a1 Slider
                m_f_a1_left = m_fSlider_a1;
                m_f_a1_right = m_fSlider_a1;
                break;

        case 1:
                // cook the Volume Slider
                m_fVolume = pow(10.0, m_fVolume_dB/20.0);
                break;

        case 10:
                // direct map to the a0 Slider
                m_f_a0_left = m_fSlider_a0;
                m_f_a0_right = m_fSlider_a0;
                break;

        default:
                ; // do nothing
        }
        return true;
}
```

Rebuild the DLL and load it. Place the volume control at 0 decibels (dB) (maximum) and set a_0 and a_1 to 1.0 (maxima); then open the analyzer and hit the Frequency button—play with the different values for a_0 and a_1. Figure 4.21 shows the response with various coefficient values.

After you play around with the controls, there are several things to note from this experiment:

- You can get a low-pass or high-pass response, or anything between, including a flat response (a_0 = 1.0, a_1 = 0.0).

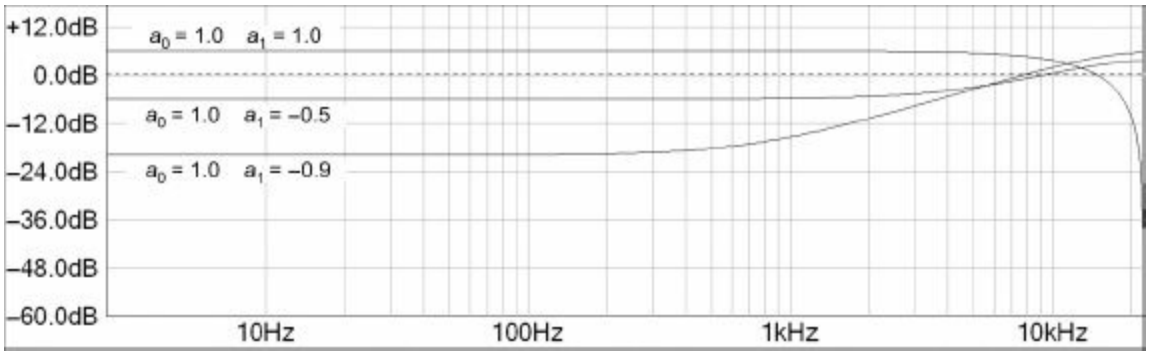


Figure 4.21: Three different combinations of coefficient settings yield three different filters.

The topology of the filter determines its difference equation. The coefficients (a_N) of a filter determine its filter frequency and phase response and therefore its type (HPF, LPF, etc.) and its sonic qualities. Your plug-in implements the difference equation in processAudioFrame(). Your plug-in calculates the coefficients in userInterfaceChange() in response to the user making changes to the control surface.

- Inverted settings give the identical frequency response but the phase responses will be inverted (use the Phase button and have a look for yourself).
- You can get gain out of the filter.
- You can also get attenuation out of the filter.

4.3 First-Order Feed-Back Filter

A first-order feed-back filter is shown in Figure 4.22.

The difference equation is as follows:

$$y(n) = a_0x(n) - b_1y(n-1) \tag{4.2}$$

You can see the feed-back nature of the filter; the output $y(n)$ is fed back into the summer through a one-sample-delay z^{-1} element. Notice that the feedback coefficient has a negative sign in front of it and the difference equation reflects this with the $-b_1$ term. The negative sign is for mathematical convenience and will make more sense in the next chapter when we analyze the difference equations in more detail. Also notice that there is no b_0 coefficient—there will not be a b_0 coefficient in any of the feed-back filters. To analyze this without math requires going through the same procedure as before, analyzing the amplitude and phase of the basic test signals. Another option would be to code it as a plug-in in RackAFX and map the coefficients directly to the sliders as you did in the previous example. Then, you can experiment with the coefficients and see how they affect the filter's frequency, phase, DC/step, and impulse responses.

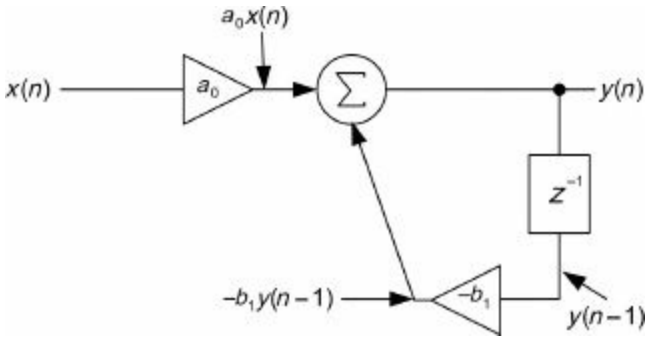


Figure 4.22: First-order feed-back filter block diagram.

4.4 Design a General First-Order Feed-Back Filter

4.4.1 Project FeedBackFilter

Create a new RackAFX project called “FeedBackFilter.” The plan is for this filter to let you directly control the coefficients with UI sliders and use the analyzer to check the resulting filters.

4.4.2 FeedBackFilter GUI

The GUI will consist of two sliders, one for the a_0 coefficient and the other for the b_1 coefficient. You can use any two sliders you like, but make sure you keep track of the control index values later on. Figure 4.23 shows my version of the GUI. Right-click near the sliders you want to use and set them up according to Table 4.6.

4.4.3 FeedBackFilter.h File

Add the z^{-1} elements and a_0, b_1 coefficient variables for the right and left channels in your .h file:

```
// Add your code here: ---------------------------------------------------- //
float m_f_a0_left;
float m_f_b1_left;
```

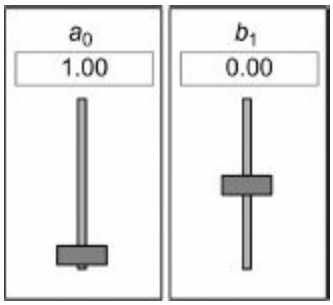


Figure 4.23 The GUI for the FeedBackFilter project.

Table 4.6: The a_0 slider properties.

Slider Property	Value
Control Name	b1
Units	
Variable Type	float
Variable Name	m_fSlider_b1
Low Limit	−1.0
High Limit	+1.0
Initial Value	0.0

Slider Property	Value
Control Name	a0
Units	
Variable Type	float
Variable Name	m_fSlider_a0
Low Limit	−1.0
High Limit	+1.0

Initial Value	1.0

```
float m_f_a0_right;
float m_f_b1_right;

float m_f_z1_left;
float m_f_z1_right;
// END OF USER CODE ------------------------------------------------------------ //
```

4.4.4 FeedBackFilter.cpp File

Constructor

- Initialize the internal a_0 and b_1 variables to match our GUI variables.
- Zero out the delay line elements.

```
CFeedBackFilter::CFeedBackFilter()
{
        // Added by RackAFX - DO NOT REMOVE
        //
        // Setup the RackAFX UI Control List and Initialize Variables
        initUI();
        // END InitUI

        <SNIP SNIP SNIP>

        // Finish initializations here
        //
        // setup our coefficients
        m_f_a0_left = m_fSlider_a0;
        m_f_b1_left = m_fSlider_b1;

        m_f_a0_right = m_fSlider_a0;
        m_f_b1_right = m_fSlider_b1;

        // flush the memory registers
        m_f_z1_left = 0.0;
        m_f_z1_right = 0.0;
}
```

prepareForPlay()

- Flush the z^{-1} storage registers.

```
bool __stdcall CFeedBackFilter::prepareForPlay()
{
        // Add your code here:
        m_f_z1_left = 0.0;
        m_f_z1_right = 0.0;

        return true;
}
```

processAudioFrame()

- Implement the difference equation in processAudioFrame(); notice the (−) sign in the difference equation too.

```
bool __stdcall CFeedBackFilter::processAudioFrame(float* pInputBuffer, float*
pOutputBuffer, UINT uNumInputChannels, UINT uNumOutputChannels)
{
	// Do LEFT (MONO) Channel; there is always at least one input/one output
	// (INSERT effect)
	// Input sample is x(n)
	float xn = pInputBuffer[0];

	// Delay sample is y(n-1)
	float yn_1 = m_f_z1_left;

	// Difference Equation
	float yn = m_f_a0_left*xn - m_f_b1_left*yn_1;

	// Populate Delay with current y(n)
	m_f_z1_left = yn;

	// Output sample is y(n)
	pOutputBuffer[0] = yn;

	// Mono-In, Stereo-Out (AUX effect) -- COPY for now
	if(uNumInputChannels == 1 && uNumOutputChannels == 2)
		pOutputBuffer[1] = yn;

	// Stereo-In, Stereo-Out (INSERT effect)
	if(uNumInputChannels == 2 && uNumOutputChannels == 2)
	{
		// Input sample is x(n)
		float xn = pInputBuffer[1];

		// Delay sample is x(n-1)
		float yn_1 = m_f_z1_right;

		// Difference Equation
		float yn = m_f_a0_right*xn - m_f_b1_right*yn_1;

		// Populate Delay with current y(n)
		m_f_z1_right = yn;

		// Output sample is y(n)
		pOutputBuffer[1] = yn;
	}
	return true;
}
```

userInterfaceChange()

- Update the coefficients when the user changes a slider.

- Check your nControlIndex values to make sure you are processing the correct control.

```
bool CFeedBackFilter::userInterfaceChange(int nControlIndex)
{
	// add your code here
	switch(nControlIndex)
	{
	case 0:
		// map the a0 Slider
		m_f_a0_left = m_fSlider_a0;
		m_f_a0_right = m_fSlider_a0;
		break;
	case 1:
		// map the b1 Slider
		m_f_b1_left = m_fSlider_b1;
		m_f_b1_right = m_fSlider_b1;
		break;

	default:
		; // do nothing
	}

	return true;
}
```

Build and load the plug-in into RackAFX. Open the analyzer and use the Frequency, Phase, Impulse, and Step buttons to analyze the output. Play with the controls. Notice first that the a_0 slider only controls the gain and phase of the signal; in the frequency response this just moves the curve up or down and it disappears when $a_0 = 0.0$, which makes sense. The real action is with the b_1 control; try the examples in Figures 4.24 through 4.26.

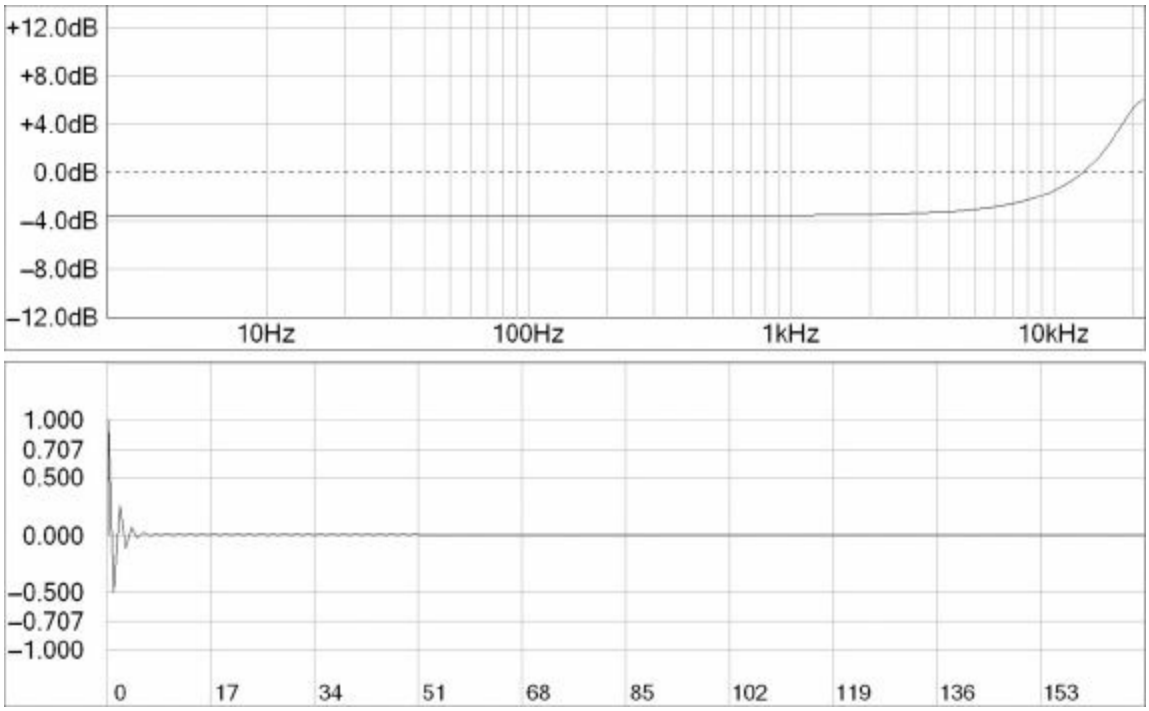


Figure 4.24: $a_0 = 1.0$ and $b_1 = 0.5$; the frequency response is high-pass/low-shelf in nature and

has gain above 11 kHz, while the impulse response shows slight ringing.

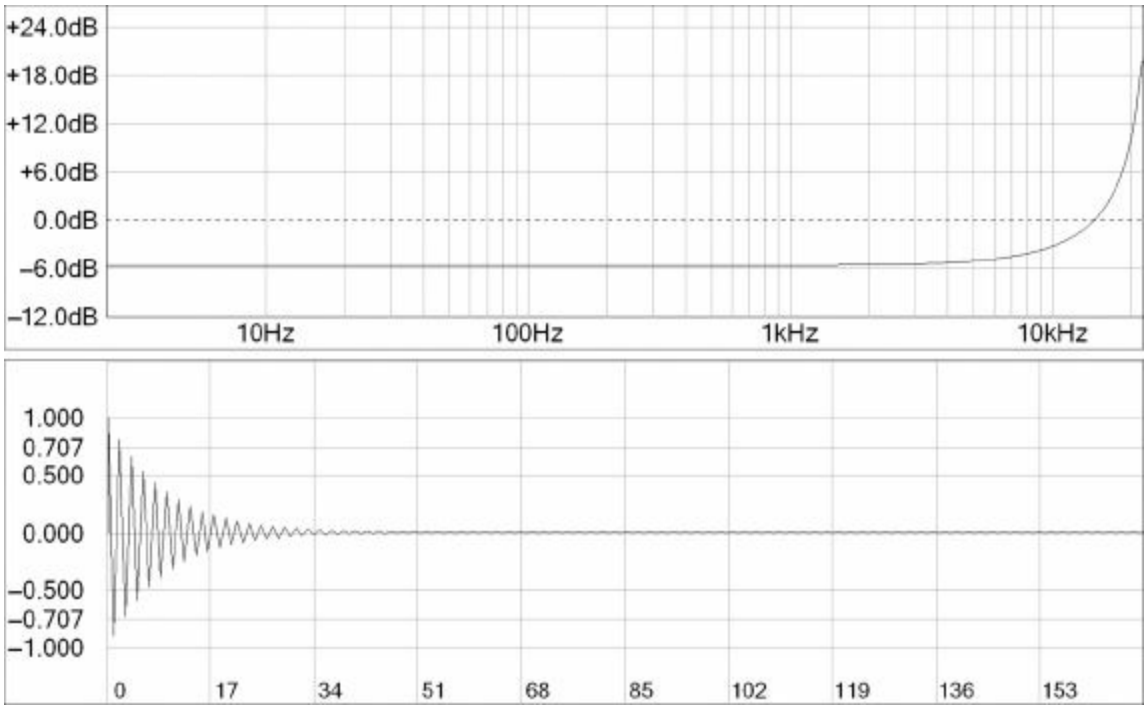


Figure 4.25: $a_0 = 1.0$ and $b_1 = 0.9$; the frequency response has a steeper high-pass response than before and has gain of +20 dB at Nyquist, while the impulse response shows considerable ringing.

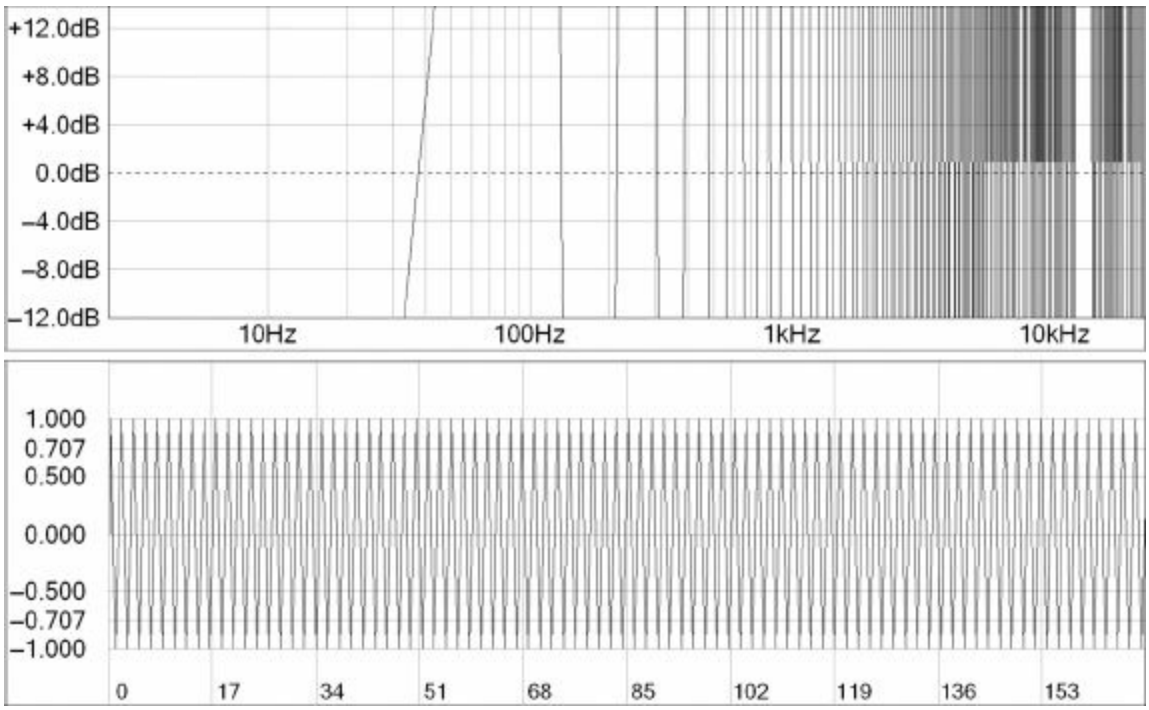


Figure 4.26: $a_0 = 1.0$ and $b_1 = 1.0$; the frequency response has blown up while the impulse response rings forever.

What happened in that last filter? Why was there no frequency response? The filter became unstable and *blew up*. It blew up because the b_1 coefficient was 1.0, which introduced 100% positive feedback into the loop. The output recirculated through the feedback loop forever causing the infinite

ringing seen in the step and impulse responses. Also notice that as the b_1 variable was increased, the gain at Nyquist also increased. With $b_1 = 1.0$, the gain at Nyquist is actually infinite.

4.5 Observations

In doing these exercises, you have made a lot of progress—you know how to implement both feed-forward and feed-back filters in a plug-in. You also have a good intuitive knowledge about how the coefficients control the filter type. Plus you got to blow up a filter, so that is pretty cool. Here are some observations.

4.5.1 General

- Feed-forward and feed-back topologies can both make high-pass or low-pass-shaped filter responses.
- The coefficients of the filter ultimately determine the kind of filter and what it sounds like.
- The phase shift introduced by the delay element(s) is responsible for the filtering operation.

4.5.2 Feed-Forward Filters

- Operate by making some frequencies go to zero; in the case of $a_0 = 1.0$ and $a_1 = 1.0$, the Nyquist frequency went to zero; this is called a *zero of transmission* or a *zero frequency* or just a *zero*.
- The step and impulse responses show smearing. The amount of smearing is exactly equal to the total amount of delay in the feed-forward branches.
- Don't blow up.
- Are called *finite impulse response* (FIR) filters because their impulse responses, though they may be smeared, are always finite in length.

4.5.3 Feed-Back Filters

- Operate by making some frequencies go to infinity; in the case of $a_0 = 1.0$ and $b_1 = 1.0$, the Nyquist frequency went to infinity and with $a_0 = 1.0$ and $b_1 = -1.0$, DC or 0 Hz went to infinity; this is called a *pole of transmission* or a *pole frequency* or just a *pole*.
- The step and impulse responses show overshoot and ringing or smearing depending on the coefficients. The amount of ringing or smearing is proportional to the amount of feedback.
- Can blow up (or go unstable) under some conditions.
- Are called *infinite impulse response* (IIR) filters because their impulse responses can become infinite.

The problem now is that we want to be able to specify the filter in a way that makes sense in audio—a low-pass filter with a cut-off of 100 Hz or a band-pass filter with a Q of 10 and center frequency of 1.2 kHz. What we need is a better way to analyze the filtering than just randomly trying coefficients, and we need methods to come up with filter coefficients based on how we specify the filter. In the next chapter, we'll work on the basic DSP theory that will make this happen.

Bibliography

Dodge, C. and Jerse, T. 1997. *Computer Music Synthesis, Composition and Performance*, Chapters 3 and 6. New York: Schirmer.

Steiglitz, K. 1996. *A DSP Primer with Applications to Digital Audio and Computer Music*, Chapter 4. Menlo Park, CA: Addison-Wesley.

CHAPTER 5

Basic DSP Theory

You want to get a grip on the underlying digital signal processing (DSP) theory of filters for several reasons. It helps to understand the anatomy of the filter because you have to implement it in code; a deeper understanding of the theory can only help your coding strategy. Also, the same DSP filter analysis and mathematical models can be applied to other effects including delay, chorusing, reverb, and compression. In order to intuitively understand the foundation of DSP theory, you need to review some math and engineering concepts.

5.1 The Complex Sinusoid

The analysis and design of digital filters uses the sinusoid as its basic stimulus function. Since Fourier showed that a signal can be decomposed into sinusoids, if you know how the system reacts to a sinusoidal stimulus at a bunch of different frequencies, you can plot the frequency and phase responses like you did by hand in the last chapter. This is akin to the impulse response—since the input signal is a train of impulses of varying amplitudes, if you know how the filter responds to a single impulse, you can figure out how it will respond to multiple impulses. You also did this by hand when you took the impulse response of the low-pass filter.

Everyone is familiar with the sine and cosine functions—sine and cosine are related by an offset of 90 degrees and the sine function starts at zero, whereas cosine starts at 1.0. In Figure 5.1 you can identify the sine and cosine waveforms by their starting position. But what about the sine-like waveform that starts at an arbitrary time in the lower plot? Is it a sine that has been phase shifted backwards or a cosine that has been phase shifted forward? You have to be careful how you answer because sine and cosine have different mathematical properties; their derivatives are not the same and it usually becomes difficult when you try to multiply them or combine them in complex ways. Add a phase offset to the argument of the sin() or cos() function and then it really turns into a mess when you do algebra and calculus with them. What you need is a function that behaves in a sinusoidal manner, encapsulates both sine and cosine functions, and is easy to deal with mathematically. Such a function exists, and it's called the complex sinusoid:

$$\text{Complex sinusoid} = e^{j\omega t} \tag{5.1}$$

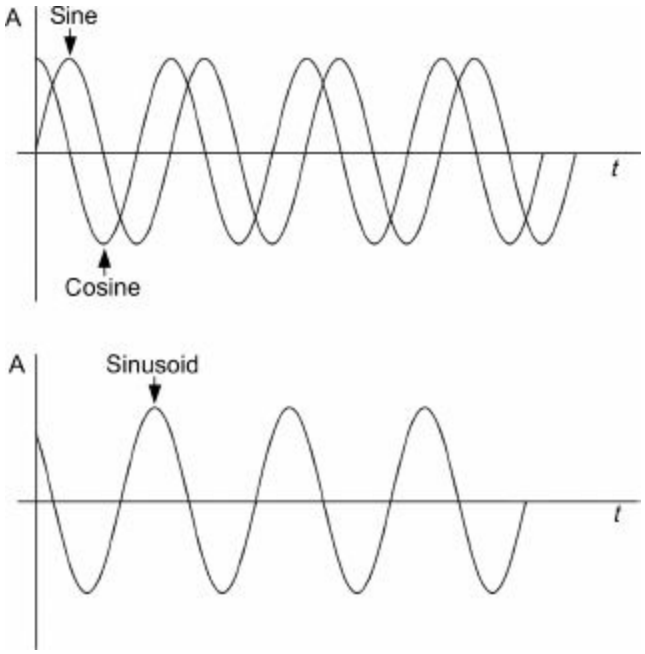


Figure 5.1: Sine, cosine, and sinusoid signals.

Euler's equation is shown below:

$$e^{j\theta} = \cos(\theta) + j\sin(\theta)$$
$$j = \sqrt{-1} \tag{5.2}$$

You can see that it includes both sine and cosine functions. The *j* term is the imaginary number, the square root of −1 (mathematicians call it *i* but since that represents current in engineering, we rename it *j* instead). The *j* is known as the phase rotation operator; multiplying a function by *j* rotates the phase by 90 degrees.

Suppose you want to shift the phase of a waveform by 180°, thereby inverting it. Mathematically, you can do this by multiplying the waveform by −1, inverting the values of all its points. Suppose you wanted to invert the waveform again (which would bring it back to its original shape)—you could do that by multiplying by −1 again. But suppose that you only wanted to shift the phase by 90°? Is there a number (η) you could use as a multiplier to shift by 90°? In other words,

90-degree shifted waveform = (originalwaveform) (η)

You don't know what η is yet, but you can figure it out. Suppose you then wanted to shift the waveform by another 90 degrees, which would be the same as shifting the original waveform by 180 degrees. You would multiply it by η again. You'd then have the following:

$$\begin{aligned} \text{180-degree shifted waveform} &= (\text{originalwaveform})\,(\eta)(\eta) \\ &= (\text{originalwaveform})\,(\eta)^2 \\ &= (\text{originalwaveform})\,(-1) \end{aligned}$$

This leads to Equation 5.3:

$$\eta^2 = -1$$
$$\eta = \sqrt{-1} = j \tag{5.3}$$

So, you can perform a conceptual 90-degree phase shift by multiplying a waveform by j. A −90-degree phase shift is accomplished by multiplying by $-j$. Some other useful relationships with j are

$$j^2 = -1$$
$$\frac{1}{j} = -j \tag{5.4}$$

Euler's equation is complex and contains a real part (cos) and imaginary part (sin), and the plus sign (+) in the equation is not a literal addition—you can't add real and imaginary numbers together. In a complex number of the format A + jB, the two parts coexist as part of one complex number.

For our purposes, we replace the θ with ωt instead, where ω is the frequency in radians/second and t is the time variable. Plug in various values for t and you get the plot in Figure 5.1 when you plot the sine and cosine in the same plane. So, we will reject using the sin() and cos() functions independently and adopt the complex sinusoid as a prepackaged mixture of the two. The reason is partly mathematical—as it turns out, e is simple to deal with mathematically. You only need to learn four rules (Equation 5.5) in addition to Euler's equation.

Euler's equation: $e^{j\omega t} = \cos(\omega t) + j\sin(\omega t)$

The four rules:

$$e^a e^b = e^{(a+b)} \text{ or } e^{at}e^{bt} = e^{(a+b)t}$$
$$\frac{e^a}{e^b} = e^{(a-b)} \text{ or } \frac{e^{at}}{e^{bt}} = e^{(a-b)t}$$
$$\frac{d(e^{at})}{dt} = ae^{at} \tag{5.5}$$
$$\int e^{at}dt = \frac{1}{a}e^{at}$$

> So, what Euler's equation is really describing is a sine and cosine pair of functions, coexisting in two planes that are 90 degrees apart. The word *orthogonal* is the engineering term for *90 degrees apart*.

The two equations in Equation 5.5 demonstrate that e behaves like a polynomial ($x^2 * x^5 = x^7$) even when the argument is a function of time, t. Equation 5.5 also shows how simply it behaves in calculus—multiple derivatives or integrations are done by simply multiplying the argument's constant (a or 1/a) over and over. Before we leave this topic, make sure you remember how to deal with complex numbers—you'll need it to understand where the frequency and phase responses come from.

5.2 Complex Math Review

Because a complex number has both real and imaginary parts, it cannot be plotted on a single axis. In

order to plot a complex number, you need two axes: a real and imaginary axis. The two axes are aligned at right angles, just like the x- and y-axes in two-dimensional geometry. They are orthogonal. The x-dimension is the real axis (Re), and the y-dimension is the imaginary axis (Im). So, a complex number is plotted as a point on this x-y plane, also called the *complex plane*. Complex numbers are usually written in the form $A + jB$ where A is the real part and B is the imaginary part. Notice that the notation always stays in the form $A + jB$ even when common sense might contradict it. For the point in the second quadrant, you still write $-2 + j2$ even though $j2 - 2$ might look nicer. Also notice that you write $2 - j1$ instead of just $2 - j$. The points are plotted the way you would plot (x, y) pairs in a plane (Figure 5.2).

This example plots points as $(x + jy)$ pairs; this is called the Cartesian coordinate system. You can also use polar notation to identify the same complex numbers. In polar notation, you specify the radius (R) and angle (θ) for the vector that results from drawing a line from the origin of the axes out to the point in question. For example, consider the point $2 + j3$ above. You could plot this point in polar form as shown in Figure 5.3.

We are leaving the Cartesian form $(2 + j3)$ for reference. Normally, all you would see is R and θ. Most engineers prefer polar notation for dealing with complex numbers. The polar number is often written as $R < \theta$. Fortunately for us, the conversion from Cartesian to polar notation is simple. Starting with a complex number in the form $A + jB$, you can find the resulting radius and angle from Equations 5.6 and 5.7.

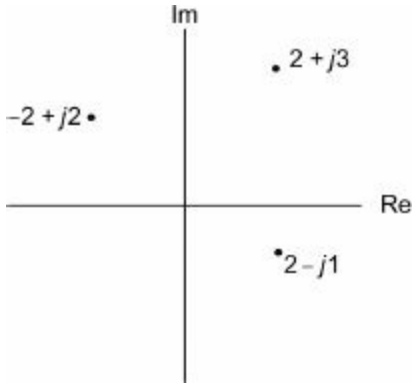


Figure 5.2: Several points plotted in the complex plane using the Cartesian (x, y) coordinate system.

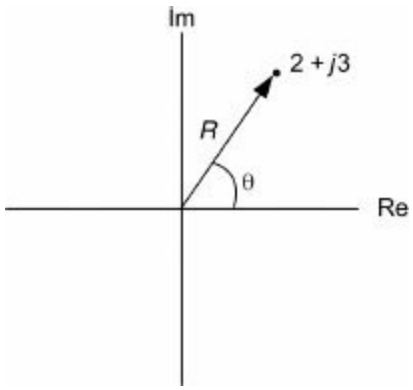


Figure 5.3: Plotting $2 + j3$ in polar form.

$$R = \sqrt{A^2 + B^2} \tag{5.6}$$

$$\theta = \tan^{-1}\left[\frac{B}{A}\right] \tag{5.7}$$

The radius (R) is sometimes called the *magnitude* and the angle θ is called the *argument*. You sometimes only care about the square of the magnitude (called the "magnitude-squared" or $|R|^2$) (Equation 5.8).

$$|R|^2 = A^2 + B^2 \tag{5.8}$$

Equation 5.9 shows how to extract the magnitude and phase from a transfer function.

$$H = \frac{\text{num}}{\text{denom}}$$

then

$$|H| = \frac{|\text{num}|}{|\text{denom}|} \tag{5.9}$$

and

$$\text{Arg}(H) = \text{Arg}(\text{num}) - \text{Arg}(\text{denom})$$

The frequency response plots of filters are actually magnitude responses of a complex function called the *transfer function* of the filter. The phase response plots are actually argument responses of this function. The transfer function is complex because it contains complex numbers; many transfer functions are actually quite simple. We use the letter H to denote a transfer function.

5.3 Time Delay as a Math Operator

The next piece of DSP theory you need to understand is the concept of time delay as a mathematical operator. This will be pretty easy since we are going to exploit the simple mathematical behavior of the $e^{j\omega t}$ function. First, consider a complex sinusoid and a delayed complex sinusoid (Figure 5.4).

How does the delay of n seconds change the complex sinusoid equation? Since positive time goes in the positive x direction, a delay of n seconds is a shifting of $-n$ seconds. In the complex sinusoid equation, you would then replace t with $t-n$. In other words, any point on the delayed curve is the same as the nondelayed curve minus n seconds. Therefore, the delayed sinusoid is

$$\text{Delayed sinusoid} = e^{j\omega(t-n)} \tag{5.10}$$

But, by using the polynomial behavior of e and running Equation 5.5 in reverse, you can rewrite it as shown in Equation 5.11:

$$e^{j\omega(t-n)} = e^{j\omega t}e^{-j\omega n} \tag{5.11}$$

It's a subtle mathematical equation but it says a lot: if you want to delay a complex sinusoid by n seconds, multiply it by $e^{-j\omega n}$—this allows us to express time delay as a mathematical operator.

In the last two sections you've learned that phase rotation and time delay can both be expressed as mathematical operators.

> Time delay can be expressed as a mathematical operator by multiplying the signal to be delayed n seconds by $e^{-j\omega n}$. This is useful because $e^{-j\omega n}$ is not dependent on the time variable. In discrete-time systems, the n refers to samples rather than seconds.

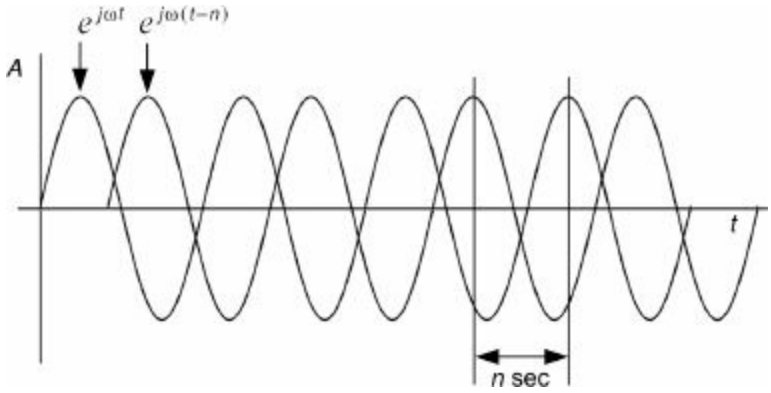


Figure 5.4: A complex sinusoid $e^{j\omega t}$ and another one delayed by n seconds.

5.4 First-Order Feed-Forward Filter Revisited

Being able to express delay as the mathematical operation of multiplication by $e^{-j\omega n}$ means you can take the block diagram and difference equation for a DSP filter and apply a sinusoid to the input in the form of $e^{j\omega t}$ rather than having to plug in sequences of samples as you did in Chapter 4. Then, you can see what comes out of the filter as a mathematical expression and evaluate it for different values of ω (where $\omega = 2\pi f$ with f in Hz) to find the frequency and phase responses directly rather than having to wait and see what comes out then try to guesstimate the amplitude and phase offsets. Consider the first-order feed-forward filter from the last chapter but with $e^{j\omega t}$ applied as the input signal, shown in a block diagram in Figure 5.5.

The difference equation is as follows:

$$y(t) = a_0 e^{j\omega t} + a_1[e^{j\omega t} e^{-j\omega 1}] \tag{5.12}$$

Figure 5.5 shows the familiar block diagram but this time with the input $x(t)$ and output $y(t)$ instead of $x(n)$ and $y(n)$. Notice the delay element has been replaced by $e^{-j\omega 1}$ since there is a one-sample delay. When you apply the complex sinusoid $e^{j\omega t}$ to the input, the difference equation uses the delay-as-multiplication operation to produce the output. With a little math you can arrive at Equation 5.13.

$$\begin{aligned} y(t) &= a_0 e^{j\omega t} + a_1[e^{j\omega t} e^{-j\omega 1}] \\ &= e^{j\omega t}(a_0 + a_1 e^{-j\omega 1}) \\ \text{and the input } x(t) &= e^{j\omega t} \text{ so} \\ y(t) &= x(t)\,(a_0 + a_1 e^{-j\omega 1}) \end{aligned} \tag{5.13}$$

The transfer function is defined as the ratio of output to input, therefore

$$\frac{y(t)}{x(t)} = a_0 + a_1 e^{-j\omega 1}$$

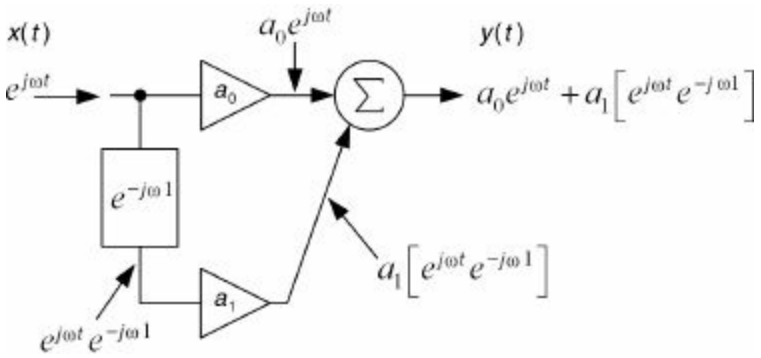


Figure 5.5: Block diagram of a first-order feed-forward filter with signal analysis.

The *transfer function* of the filter is the ratio of output to input. The *frequency response* of the filter is the magnitude of the transfer function evaluated at different frequencies across its spectrum. The *phase response* of the filter is the argument (or angle) of the transfer function evaluated at different frequencies across its spectrum.

To produce the frequency and phase response graphs, you evaluate the function for various values of ω then find the magnitude and argument at each frequency. The evaluation uses Euler's equation to replace the *e* term and produce the real and imaginary components.

What is so significant about this is that the transfer function is not dependent on time even though the input and output signals are functions of time. The transfer function (Equation 5.14) is only dependent on frequency ω, so we call it $H(\omega)$.

$$H(\omega) = a_0 + a_1 e^{-j\omega 1} \tag{5.14}$$

Notice that the transfer function is complex.

But what values of ω are to be used in the evaluation? We know that $\omega = 2\pi f$, but do we really care about the frequency in Hz? In Chapter 4 when you analyzed the same filter, you applied DC, Nyquist, ½ Nyquist, and ¼ Nyquist without thinking about the actual sampling frequency. This is called *normalized frequency* and is usually the way you want to proceed in analyzing DSP filters. The actual sampling rate determines Nyquist but the overall frequency range (0 Hz to Nyquist) is what we care about. To normalize the frequency, you let $f = 1$ Hz in $\omega = 2\pi f$, then ω varies from 0 to 2π or across a 2π range. There is also one detail we have to be aware of: negative frequencies.

5.4.1 Negative Frequencies

You may have never thought a frequency could be negative, but it can. When you first learned about the concept of a waveform's frequency, you were taught that the frequency is $1/T$, where T is the period, as shown in Figure 5.6.

The reason the frequencies came out as positive numbers is because the period is defined as $t_2 - t_1$, which makes it a positive number. But, there's no reason you couldn't define the period to be the other way around: $T = t_1 - t_2$, except that it implies that time is running backwards. Mathematically, time

can run backwards. This means that for every positive frequency that exists, there is also a negative "twin" frequency. When you look at a frequency response plot you generally only look at the positive side. Figure 5.7 shows a low-pass response up to the highest frequency in the system, Nyquist.

However, in reality, the filter also operates on the negative frequencies just the same in a mirror image. In this case, as the negative frequencies get higher and higher, they are attenuated just like their positive counterparts (Figure 5.8). And it makes sense too. If you take an audio file and reverse it in time, then run it through a low-pass filter, the same frequency filtering still occurs.

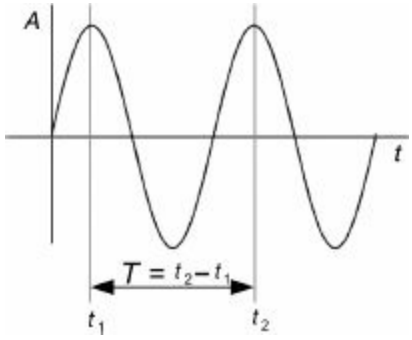


Figure 5.6: The classic way of defining the period, *T*.

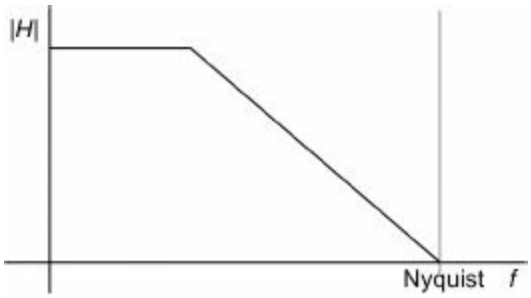


Figure 5.7: The classic way of showing a frequency response plot only shows the positive portion.

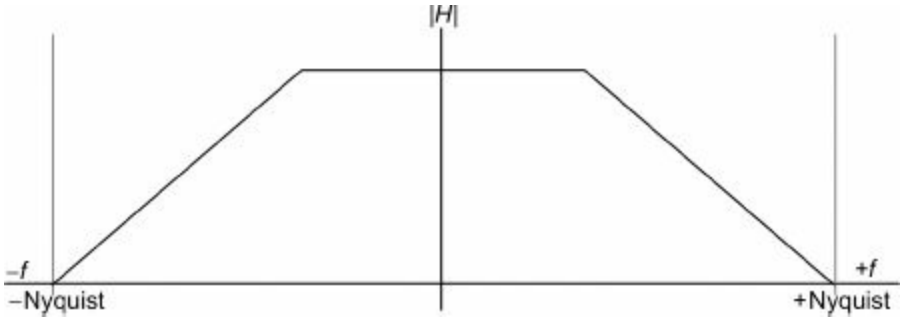


Figure 5.8: The more complete frequency response plot contains both positive and negative sides.

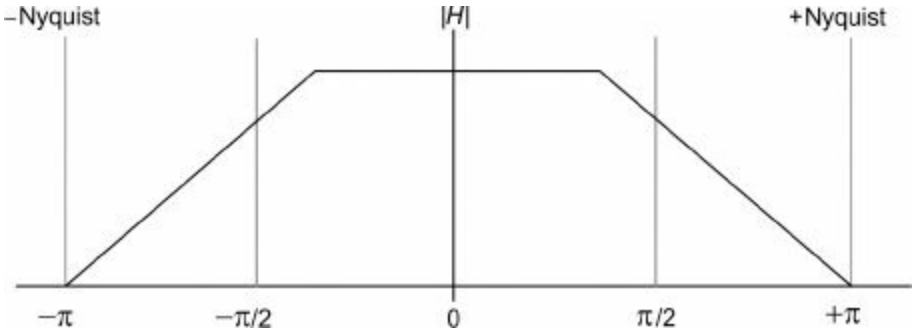


Figure 5.9: One way to divide the 2π range of frequencies includes both positive and negative frequencies.

For filter evaluation, ω varies on a 0 to 2π radians/second range and one way to think about this 2π range is to split it up into the range from −π to +π corresponding to −Nyquist to +Nyquist (Figure 5.9).

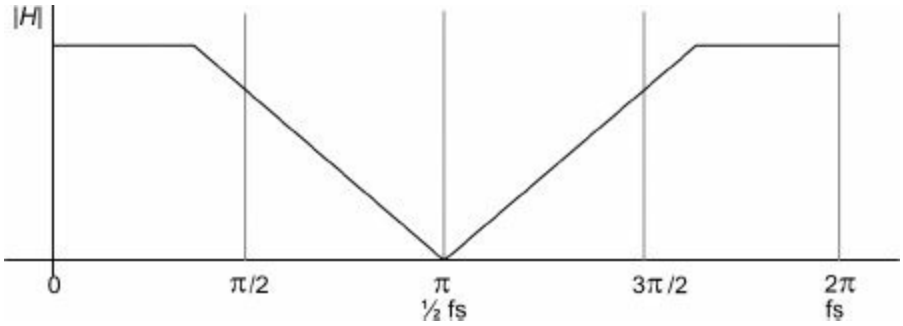


Figure 5.10: Mapping the 0 to 2π range of frequencies across the 0 to fs range.

To evaluate the transfer function, let ω vary from 0 to π and get the first half of the response. The other half is a mirror image of the data.

5.4.2 Frequencies Above and Below ±Nyquist

The sampling theorem sets up the Nyquist criteria with regards to completely recovering the original, band-limited signal without aliasing. However, frequencies above Nyquist and all the way up to the sampling frequency are also allowed mathematically. And in theory, any frequency could enter the system and you could sample it without limiting Nyquist. For a frequency or phase response plot, the frequencies from Nyquist up to the sampling frequency are a mirror image about Nyquist. This is another way to divide up the 2π range by going from 0 Hz to the sampling frequency (Figure 5.10).

Notice that in either method the same information is conveyed as we get both halves of the curves, and in both cases, Nyquist maps to π and 0 Hz to 0 and positive frequencies map to the range 0 to π.

5.5 Evaluating the Transfer Function H(ω)

DSP filter transfer functions will contain $e^{-j\omega n}$ terms that need to be evaluated over the range of 0 to π; the way to do this is by using Euler's equation to decompose the sinusoid into its real (cos) and imaginary (sin) components. Then, evaluate the cos and sin terms at the frequency in question. In the

last chapter you manually calculated the input/output relationship of a filter by cranking through the filter operation, one step at a time. In this improved method, you only need to solve the transfer function equation. Start with the block diagram in Figure 5.11.

The transfer function is as follows:

$$H(\omega) = a_0 + a_1 e^{-j\omega 1} \tag{5.15}$$

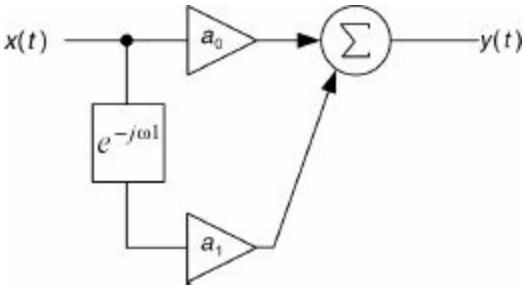


Figure 5.11: First-order feed-forward block diagram.

Table 5.1: Sine and cosine function evaluations at DC, ¼ Nyquist, ½ Nyquist, ¾ Nyquist, and Nyquist.

Frequency ω	cos(ω)	sin(ω)
0	1.0	0.0
π/4	0.707	0.707
π/2	0.0	1.0
3π/4	0.707	–0.707
π	–1.0	0.0

Use the filter coefficients $a_0 = 0.5$, $a_1 = 0.5$. You can use Table 5.1 to help with the evaluation. Evaluate at the following frequencies:

- DC: 0
- Nyquist: π
- ½ Nyquist: π/2
- ¼ Nyquist: π/4

Evaluation is a two-step process for each frequency:

1. Use Euler's equation to convert the *e* terms into real and imaginary components.
2. Find the magnitude and argument of the complex equation.

5.5.1 DC (0 Hz)

$$
\begin{aligned}
H(\omega) &= 0.5 + 0.5e^{-j\omega 1} \\
&= 0.5 + 0.5(\cos(\omega) - j\sin(\omega)) \\
&= 0.5 + 0.5(\cos(0) - j\sin(0)) \\
&= 0.5 + 0.5(1 - j0) \\
&= 1.0 + j0
\end{aligned}
\tag{5.16}
$$

Find the magnitude and phase at this frequency:

$$
\begin{aligned}
|H(\omega)| &= \sqrt{(a + jb)(a - jb)} \\
&= \sqrt{(1 + j0)(1 - j0)} \\
&= 1.0 \\
\text{Arg}(H) &= \tan^{-1}(b/a) \\
&= \tan^{-1}(0/1) \\
&= 0.0^{\circ}
\end{aligned}
\tag{5.17}
$$

Compare these mathematical results (Equations 5.16 and 5.17) with the graphical ones from the last chapter (Figure 5.12).

5.5.2 Nyquist (π)

$$
\begin{aligned}
H(\omega) &= 0.5 + 0.5e^{-j\omega 1} \\
&= 0.5 + 0.5(\cos(\omega) - j\sin(\omega)) \\
&= 0.5 + 0.5(\cos(\pi) - j\sin(\pi)) \\
&= 0.5 + 0.5\,(-1 - j0) \\
&= 0 + j0
\end{aligned}
\tag{5.18}
$$

$$
\begin{aligned}
|H(\omega)| &= \sqrt{(a + jb)(a - jb)} \\
&= \sqrt{(0 + j0)(0 - j0)} \\
&= 0.0 \\
\text{Arg}(H) &= \tan^{-1}(b/a) \\
&= \tan^{-1}(0/0) \\
&= 0^{\circ}
\end{aligned}
\tag{5.19}
$$

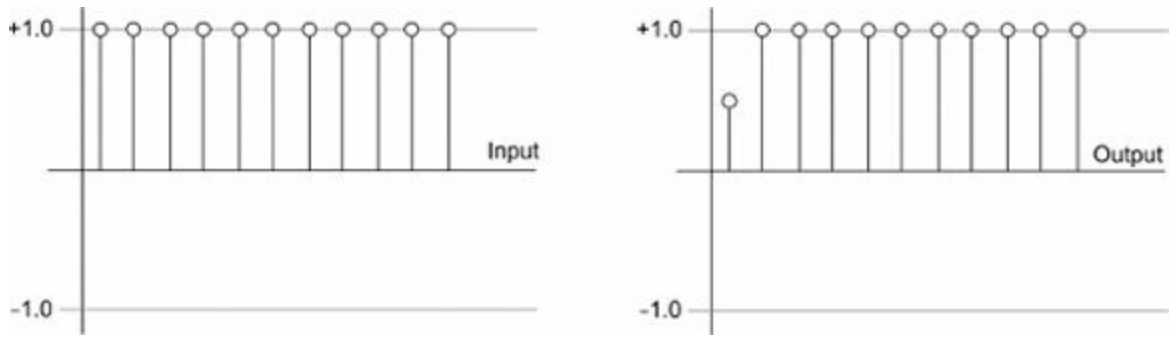


Figure 5.12: The graphical results show the same information. The magnitude is 1.0 and the phase shift is 0.

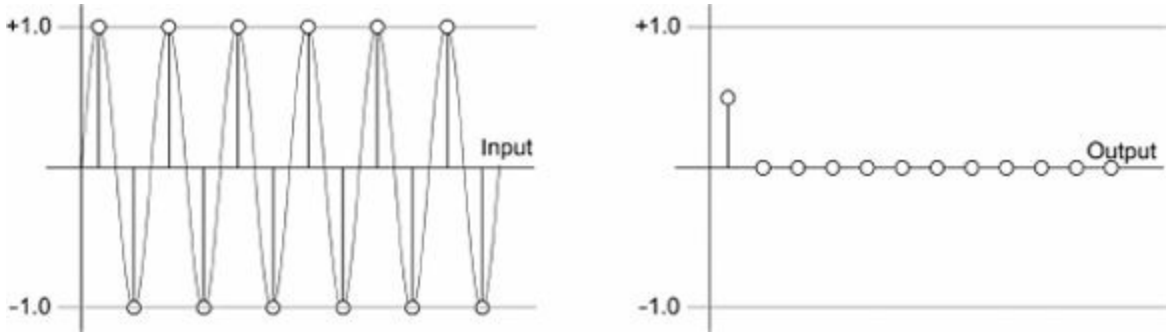


Figure 5.13: The graphical results show the same information at Nyquist—the magnitude is 0 and there is no phase shift since there is nothing there to shift.

The inverse tangent argument is 0/0 and the phase or Arg(*H*) is defined to be 0 under this condition. The C++ function you use is *arctan2*(*im, re*), which performs the inverse tangent function; it will also evaluate to 0 in this case. Now, compare our results to the last chapter's graphical results (Figure 5.13).

5.5.3 ½ Nyquist (π/2)

$$\begin{aligned} H(\omega) &= 0.5 + 0.5e^{-j\omega 1} \\ &= 0.5 + 0.5(\cos(\omega) - j\sin(\omega)) \\ &= 0.5 + 0.5(\cos(\pi/2) - j\sin(\pi/2)) \\ &= 0.5 + 0.5(0 - j1) \\ &= 0.5 - j0.5 \end{aligned} \tag{5.20}$$

$$\begin{aligned} |H(\omega)| &= \sqrt{(a + jb)(a - jb)} \\ &= \sqrt{(0.5 + j0.5)(0.5 - j0.5)} \\ &= \sqrt{0.25 + 0.25} = \sqrt{0.5} \\ &= 0.707 \end{aligned} \tag{5.21}$$

$$\begin{aligned} \text{Arg}(H) &= \tan^{-1}(b/a) \\ &= \tan^{-1}(-0.5/0.5) \\ &= -45^{\circ} \end{aligned}$$

Compare this to the last chapter's graphical results (Figure 5.14); the magnitude is 0.707 with a phase shift of −45 degrees, and the results agree.

5.5.4 1/4 Nyquist (π/4)

$$\begin{aligned} H(\omega) &= 0.5 + 0.5e^{-j\omega 1} \\ &= 0.5 + 0.5(\cos(\omega) - j\sin(\omega)) \end{aligned} \tag{5.22}$$

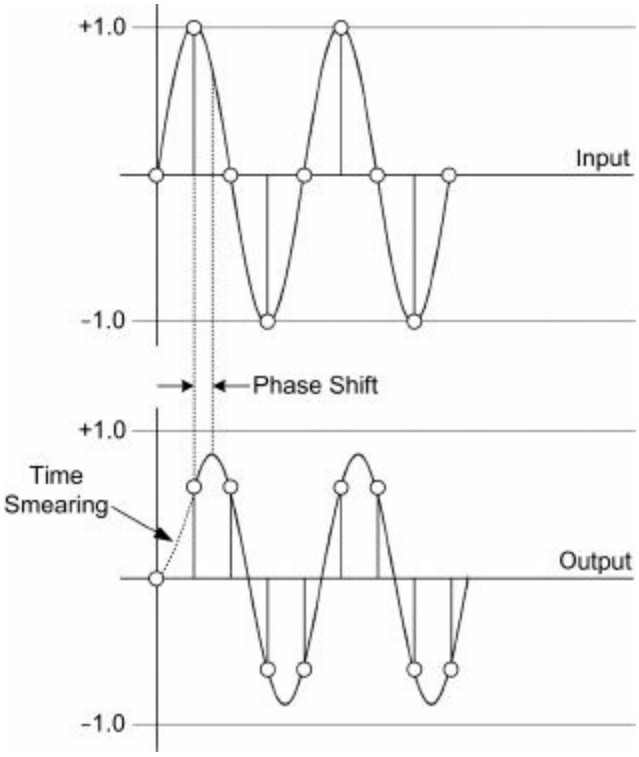


Figure 5.14: Graphical results from the last chapter at ½ Nyquist.

$$
\begin{aligned}
&= 0.5 + 0.5(\cos(\pi/4) - j\sin(\pi/4)) \\
&= 0.5 + 0.5(0.707 - j0.707) \\
&= 0.853 - j0.353 \\
|H(\omega)| &= \sqrt{(a + jb)(a - jb)} \\
&= \sqrt{(0.853 + j0.353)(0.853 - j0.353)} \\
&= \sqrt{0.728 + 0.125} = \sqrt{0.853} \\
&= 0.923 \qquad (5.23) \\
\text{Arg}(H) &= \tan^{-1}(b/a) \\
&= \tan^{-1}(-0.353/0.853) \\
&= -22.5^{\circ}
\end{aligned}
$$

Compare to the last chapter's graphical results (Figure 5.15); you can see how much more accuracy we get with the mathematical calculation. The magnitude and phase shift look about right when compared to the graphs.

Now, you can combine all the evaluations together and sketch out the frequency response of the filter (Figure 5.16).

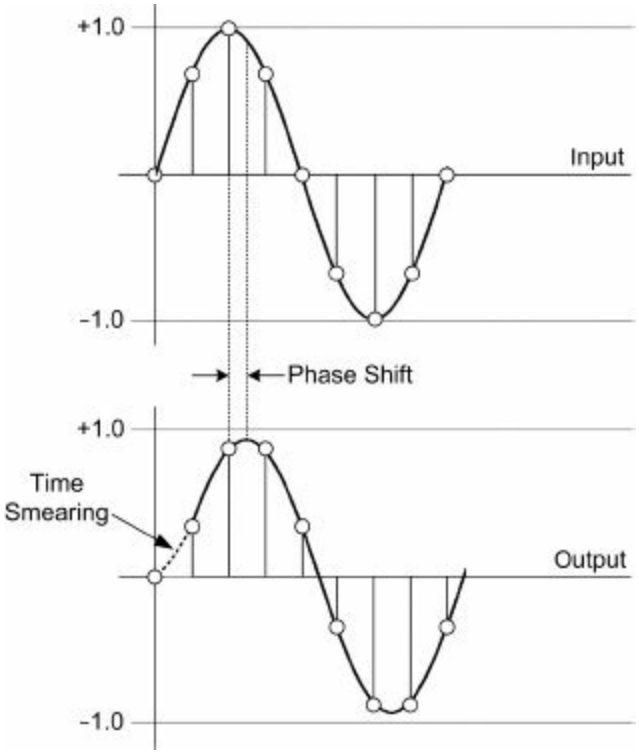


Figure 5.15: Graphical results from the last chapter at ¼ Nyquist.

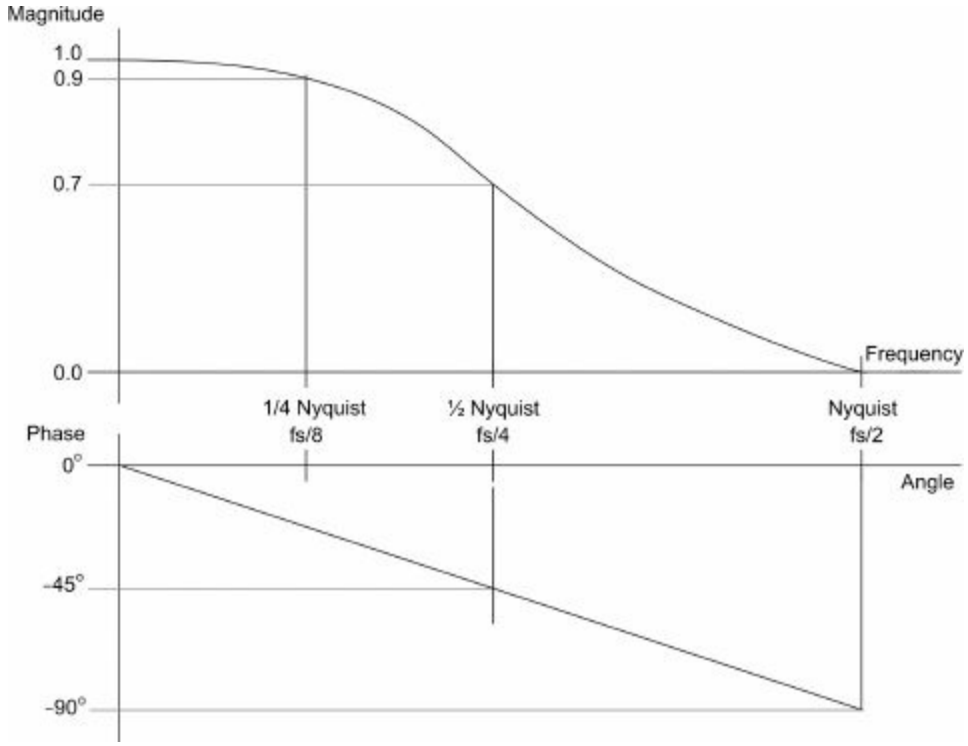


Figure 5.16: The final composite frequency and phase response plots show the same results as the last chapter, but with a lot less work.

Hopefully, this quick example has convinced you that it is better to do a little complex math than have to analyze and design these filters by brute force analysis of time domain input sequences.

5.6 Evaluating $e^{j\omega}$

In the evaluation of the transfer function, you had to substitute values of ω from 0 to π into the $e^{j\omega}$

terms of the equation. But what would the plot look like if you evaluate a single $e^{j\omega}$ term? You saw that the use of Euler's equation produced the real and imaginary components of the term and now it's time to plot them over the range of 0 to π.

If you evaluate $e^{j\omega}$ over more frequencies and plot the resulting values in the complex plane, you get an interesting result. The frequencies in Table 5.2 from 0 to +π map to an arc that is the top half of a circle with radius = 1.0, shown in Figure 5.17. Remember, the magnitude is the radius and the argument is the angle when using polar notation, which simplifies the analysis. You don't have to keep track of the real and imaginary parts. The evaluation at ω = π/4 is plotted on the curve. The circle this arc is laying over would have a radius of 1.0 and is called the *unit circle*. If you evaluate $e^{j\omega}$ over the negative frequencies that correspond to 0 to −π, you get a similar but inverted table (Table 5.3).

Table 5.2: The magnitude and angle of $e^{j\omega}$ from DC to Nyquist.

Frequency π	$e^{j\omega} = \cos(\omega) + j\sin(\omega)$	$\lvert e^{j\omega} \rvert$	$\text{Arg}(e^{j\omega})$
DC (0Hz)	1 + j0	1.0	0
¼ Nyquist	0.707 + j0.707	1.0	π/4
½ Nyquist	0 + j1	1.0	π/2
Nyquist	–1 + j0	1.0	π

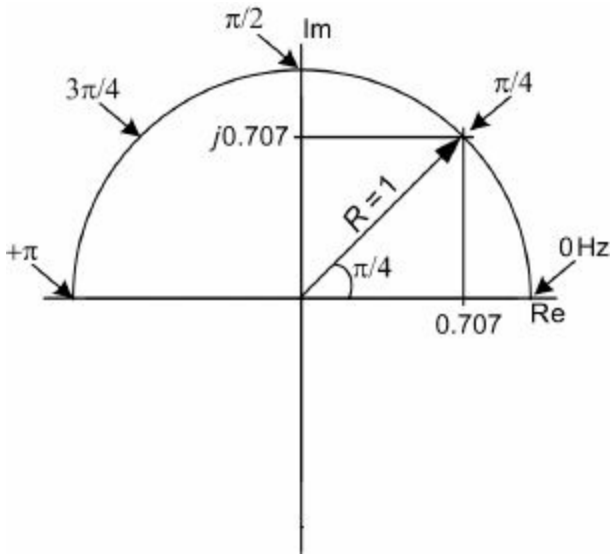


Figure 5.17: The positive frequencies map to the upper half of the unit circle.

This table translates to a mapping across the lower half of the same unit circle (Figure 5.18). The negative frequencies increase as you move clockwise from 0 Hz, the radius stays 1.0 during the entire arc.

Why bother to evaluate $e^{j\omega}$? It will be useful very soon when we start picking apart the transfer functions in an effort to figure out how to design filters. It also shows the limited "frequency space" of the digital domain. All the frequencies that could exist from −Nyquist to +Nyquist map to outline of a simple unit circle. In contrast the analog domain has an infinitely long frequency axis and an infinite

frequency space.

Table 5.3: The magnitude and angle of $e^{j\omega}$ from DC to –Nyquist.

Frequency ω	$e^{j\omega} = \cos(\omega) + j\sin(\omega)$	$\lvert e^{j\omega} \rvert$	$\text{Arg}(e^{j\omega})$
DC (0Hz)	$1 + j0$	1.0	0
−¼ Nyquist	$0.707 - j0.707$	1.0	$-\pi/4$
−½ Nyquist	$0 - j1$	1.0	$-\pi/2$
−Nyquist	$-1 + j0$	1.0	$-\pi$

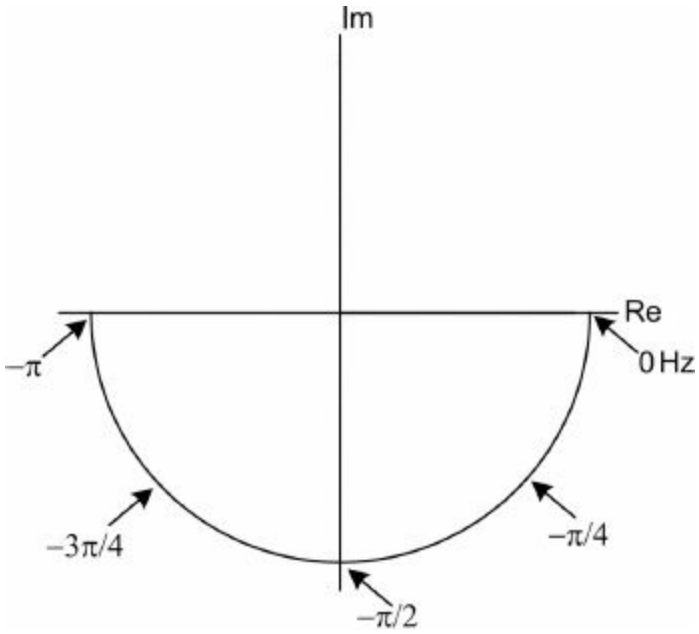


Figure 5.18: The negative frequencies map to the lower half of the unit circle.

The *order* of a filter is the order of the polynomial in the transfer function that describes it. The order of the polynomial is the maximum absolute exponent value found in the equation.

5.7 The z Substitution

It's going to get messy having to write $e^{j\omega}$ so often and we know $e^{j\omega}$ behaves like a polynomial mathematically. So, we can simplify the equations by making a simple substitution of Equation 5.24.

$$z = e^{j\omega} \tag{5.24}$$

This is just a substitution right now and nothing else. Making the substitution in Equation 5.24 and noting the resulting transfer function is now a function of z, not ω, we can write it like Equation 5.25:

$$H(z) = a_0 + a_1 z^{-1} \tag{5.25}$$

The reason this is useful is that it turns the transfer function into an easily manipulated polynomial in

z. In this case, the polynomial is a first-order polynomial (the highest exponent absolute value is 1) and this is the real reason the filter is named a first-order filter—it's the polynomial order of the transfer function.

5.8 The z Transform

The z substitution does a good job at simplifying the underlying polynomial behavior of $e^{j\omega}$ and it lets us use polynomial math to solve DSP problems. But, there is an interesting application of $z = e^{j\omega}$ that simplifies the design and analysis of digital filters. In the graph from Chapter 1 (Figure 5.19) you can see how the indexing of the samples determines their place in time. The future samples have positive indices and the past (delayed) samples have negative indices.

We've seen that $e^{j\omega}$ is the delay operator, so $e^{-j\omega 1}$ means one sample of delay, or one sample *behind* the current one. Likewise, $e^{-j\omega 2}$ would be two samples behind and $e^{+j\omega 4}$ indicates 4 samples *ahead* or in the future. That means that $e^{j\omega}$ could also be used as a book-keeping device since it can relate the position in time of something (a sample, for us).

The rules for implementing the z transform on a discrete signal or difference equation are easy and can often be done by inspection. The current sample $x(n)$ or $y(n)$ transforms into the signal $X(z)$ or $Y(z)$. Instead of thinking of the *sample* $x(n)$ you think of the *signal* $X(z)$, where $X(z)$ is the whole signal—past, present, and future.

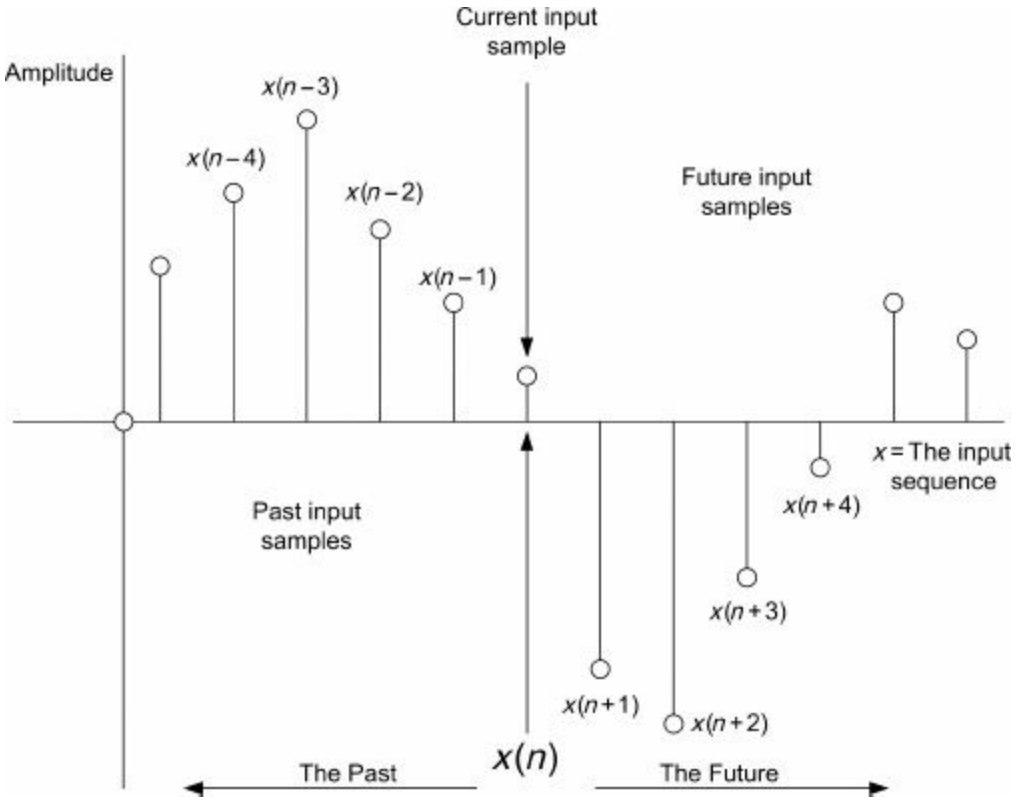


Figure 5.19: Our book-keeping rules shown graphically.

The *z transform* changes a sequence of samples in n to a sequence of samples in z by replacing the indices …n−1, n, n+1… with … z^{-1}, z, z^{+1}… This works because multiplication by $z = e^{j\omega}$ represents the operation of delay or time shift. The resulting transformed sequence now is a function of the complex frequency $e^{j\omega}$, therefore it transforms things from the *time domain* into the *complex*

frequency domain.

- $x(n) \rightarrow X(z)$
- $y(n) \rightarrow Y(z)$

Instead of thinking of the sample $x(n-1)$ as being delayed by one sample, you think of the signal $X(z)$ delayed by one sample, z^{-1}. The delayed signals are the result of the whole signal multiplied by the z^{-N} terms:

- $x(n-1) \rightarrow X(z)\, z^{-1}$
- $y(n-1) \rightarrow Y(z)\, z^{-1}$
- $x(n-2) \rightarrow X(z)\, z^{-2}$
- $y(n+6) \rightarrow Y(z)\, z^{+6}$

You can see that this concept relies on the ability to express delay as a mathematical operator. It not only allows us to express an *algorithm* based on z, it also lets us express a *signal* based on z. Mathematically, the z transform is

$$X(z) = \sum_{n=-\infty}^{n=+\infty} x(n) z^{-n} \tag{5.26}$$

5.9 The z Transform of Signals

Remember, $x(n)$ is the sequence of samples just like the ones you used for analysis. Figure 5.20 shows an example. This simple, finite length signal consists of five samples. The remaining zero samples don't need to be counted.

$$x(n) = \{0, 0.25, 0.5, 0.75, 1.0\}$$

The sequence $x(n)$ could also be written $x(n) = \{x(0), x(1), x(2), x(3), x(4)\}$, so using Equation 5.26 we transform $x(n)$ into $X(z)$ and write Equation 5.27:

$$X(z) = 0z^0 + 0.25z^{-1} + 0.5z^{-2} + 0.75z^{-3} + 1.0z^{-4} \tag{5.27}$$

You could read Equation 5.27 as follows: "The whole signal $X(z)$ consists of a sample with an amplitude of 0 at time 0 followed by a sample with an amplitude of 0.25 one sample later and a sample with an amplitude of 0.5 two samples later and a sample with an amplitude of 0.75 three samples later and . . ." This should shed light on the fact that the transform really involves the book-keeping of sample locations in time and that the result is a polynomial. You can multiply and divide this signal with other signals by using polynomial math. You can mix two signals by linearly combining the polynomials.

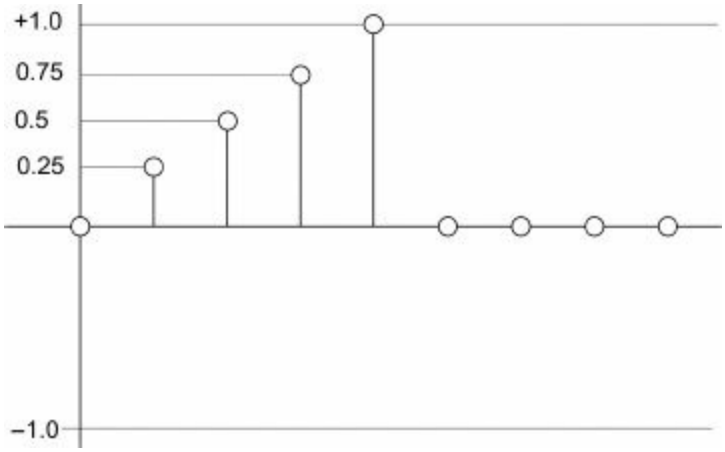


Figure 5.20: A simple signal for analysis.

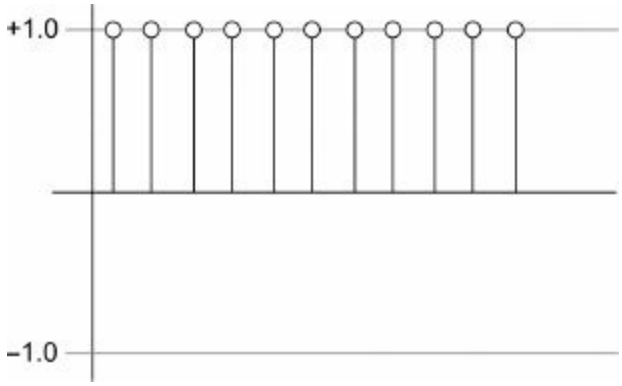


Figure 5.21: The DC signal is infinite in length.

Let's do one more example regarding the transformation of an input signal. This time, let's choose the DC signal—it goes on forever. Figure 5.21 shows the DC signal with the first sample at 1.0 and all preceding samples at 0.0 with a sequence of $x(n) = \{1, 1, 1, 1, 1\ldots\}$.

Using Equation 5.26 you can directly write the z transform as (remember, $1z^0 = 1$) in Equation 5.28:

$$X(z) = 1 + z^{-1} + z^{-2} + z^{-3} + \ \ldots\ + z^{-\infty} \tag{5.28}$$

While that looks like an ugly, infinitely long equation, it can also be represented in a *closed form*. In fact, a closed form representation exists for this polynomial using a polynomial series expansion, as shown in Equation 5.29:

$$\begin{aligned} X(z) &= 1 + z^{-1} + z^{-2} + z^{-3} + \ \ldots\ + z^{-\infty} \\ &= \frac{1}{1 - z^{-1}} \\ &= \frac{z}{z - 1} \end{aligned} \tag{5.29}$$

5.10 The z Transform of Difference Equations

The z transform of signals is interesting, but something fascinating happens when you take the z transform of a difference equation, converting it into a transfer function in z all at once. And, the same easy rules apply. Let's do that with the basic first-order feed-forward filter. The difference equation is

$$y(n) = a_0x(n) + a_1x(n-1) \quad (5.30)$$

Taking the z transform:

$$Y(z) = a_0X(z) + a_1X(z)z^{-1}$$
$$Y(z) = X(z)[a_0 + a_1z^{-1}]$$
$$H(z) = \frac{Y(z)}{X(z)} = a_0 + a_1z^{-1}$$

This is a really useful result—you got the final transfer function in just a handful of steps, using the simple z transform rules. Let's recap what you had to do before you learned how to take the z transform of a difference equation:

- Redraw the block diagram with $e^{-j\omega n}$ operators in the n-sample delay elements.
- Apply the complex sinusoid $e^{-j\omega t}$ to the input.
- Find out what comes out, $y(t)$, and formulate the transfer function $H(\omega)$.
- Apply the z substitution to the transfer function.

Taking the z transform does all these steps at once and we're left with the same simple polynomial in z. If we evaluate the transfer function for different values of $z = e^{j\omega}$, we can find the frequency and phase plots. You'll get more practice taking the z transforms of more difference equations soon.

5.11 The z Transform of an Impulse Response

The z transform of a difference equation results in the transfer function. But what if you don't have the difference equation? Suppose you only have a black box that performs some kind of DSP algorithm and you'd like to figure out the transfer function, evaluate it, and plot the frequency and phase responses. It can be done without knowing the algorithm or any details about it by taking the impulse response of the system. You apply the input sequence $x(n) = \{1, 0, 0, 0, 0\ldots\}$ and capture what comes out, which we'll call $h(n)$. If you take the z transform of the impulse response, you get the transfer function expanded out into a series form.

In fact, this is exactly what the RackAFX software's audio analyzer does—it takes the impulse response of the filter and then runs a z transform on it to make the magnitude and phase plots you see on the screen. Mathematically, Equation 5.31 is identical to Equation 5.26 except we've changed the signal from X to H:

> The z transform of the impulse response $h(n)$ is the transfer function $H(z)$ as a series expansion. Evaluate the transfer function to plot the frequency and phase responses.

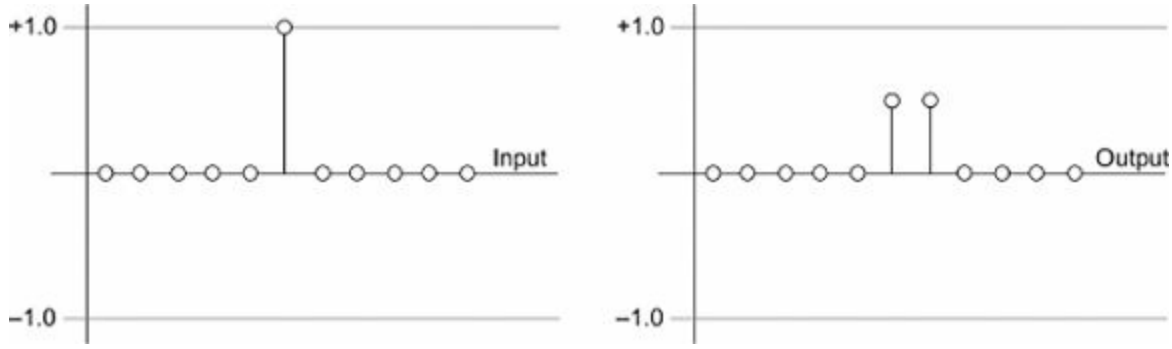


Figure 5.22: The impulse response from the first-order feed-forward filter.

$$H(z) = \sum_{n=-\infty}^{n=+\infty} h(n)z^{-n} \tag{5.31}$$

Try this on the first-order feed-forward filter we've been working on; you already have the impulse response "captured" from the last chapter (Figure 5.22).

The impulse response is $h(n) = \{0.5, 0.5\}$. Applying the z transform yields Equation 5.32:

$$\begin{aligned} H(z) &= 0.5z^0 + 0.5z^{-1} \\ &= 0.5 + 0.5z^{-1} \end{aligned} \tag{5.32}$$

Notice that this is the identical result as taking the z transform of the difference equation and the filter coefficients (0.5, 0.5) *are* the impulse response {0.5, 0.5}.

5.12 The Zeros of the Transfer Function

When we used the coefficients $a_0 = a_1 = 0.5$ we wound up with a filter that completely destroys the Nyquist frequency and you saw how its output became 0 in both the manual and complex sinusoid evaluations. We noted that feed-forward filters have *zeros of transmission* or *zero frequencies* or just *zeros* when their output becomes zero. In both the manual and complex sinusoid evaluations, we just got lucky when we stumbled upon this value as Nyquist happened to be one of the signals we were testing or evaluating. There's a way to precisely find these critical frequencies by using the polynomial result of the z transform. You probably remember factoring polynomials in high school or college. When you did that, you set the polynomial equal to 0 and then you factored to find the roots of the polynomial. What you were really doing was finding the zeros of the polynomial, that is, the values of the dependent variable that make the polynomial become zero. You can do the same thing with the transfer function by setting it equal to zero and then factoring the polynomial. Suppose $a_0 = a_1 = 0.5$ and we factor the transfer function in Equation 5.32 to get Equation 5.33:

$$\begin{aligned} H(z) &= 0.5 + 0.5z^{-1} \\ &= 0.5 + \frac{0.5}{z} \end{aligned} \tag{5.33}$$

You can find the zero by inspection—it's the value of z that forces $H(z)$ to be 0 and in this case there is a zero at $z = -1.0$. But what does it mean to have a zero at −1.0? This is where the concept of evaluating $e^{j\omega}$ comes into play. When you did that and plotted the various points, noting they were

making a unit circle in the complex plane, you were actually working in the z-plane, that is, the plane of $e^{j\omega}$. The location of the zero at $z = -1.0$ is really at the location $z = -1.0 + j0$ purely on the real axis and at Nyquist. In Figure 5.23 the zero is shown as a small circle sitting at the location $z = -1.0$.

There are several reasons to plot the zero frequencies. First, you can design a filter directly in the z-plane by deciding where you want to place the zero frequencies first, then figuring out the transfer function that will give you those zeros. Secondly, plotting the zeros gives you a quick way to sketch the frequency response without having to evaluate the transfer function directly. You can estimate a phase plot too, but it is a bit more involved. So, you have two really good reasons for wanting to plot the zeros; one for design, the other for analysis.

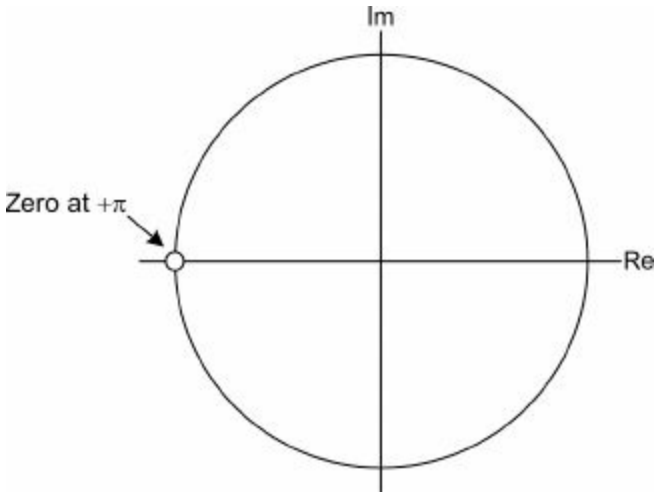


Figure 5.23: Zero is plotted in the z-plane at its location on the real axis $z = -1 + j0$.

5.13 Estimating the Frequency Response: Zeros

An interesting property of the z-plane and z transform is that you can measure the frequency response graphically on the z-plane. In the simplest case of only one zero, the method is as follows:

- Locate each evaluation frequency on the outer rim of the unit circle.
- Draw a line from the point on the circle to the zero and measure the length of this vector. Do it for each evaluation frequency.
- The *lengths* of the lines will be the *magnitudes* at each frequency in the frequency response.

In Figure 5.24 you can see the complete operation, first drawing and measuring the lines—you can use graph paper and a ruler, if you want—then building the frequency response plot from them. Notice also that the magnitude of a line drawn from Nyquist to the zero at −1 has a length of zero. The lengths of the vectors are the mathematical definition of magnitude and you are evaluating the whole filter at once. These z-plane plots are going to be useful for filter design. You can also derive the phase response, which involves measuring the angles of incidence of each vector on the zero. With multiple zeros, it becomes cumbersome. But, estimating the frequency response is pretty simple, even for more complex filters.

You might notice that even though this frequency response looks like the one we produced earlier, the gain values are not the same. In this filter, the gain is 2.0 at DC, and in ours, it's half of that. In fact, this filter's magnitudes at the evaluation frequencies are all twice what ours are.

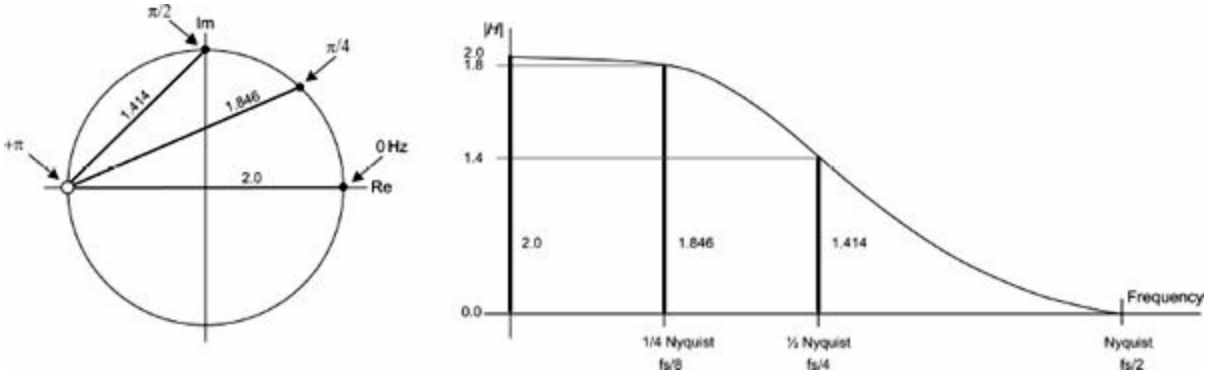


Figure 5.24: The geometric interpretation shows how the length of each vector from the evaluation frequency to the zero is really a magnitude in the response.

5.14 Filter Gain Control

The last thing you need to do is remove the overall gain factor from the transfer function so that overall filter gain (or attenuation) can be controlled by just one variable. This is actually pretty simple to do, but requires re-working the transfer function a bit. The idea is to pull out the a_0 variable as a multiplier for the whole function. This way, it behaves like a volume knob, gaining the whole filter up or down. The way you do it is to normalize the filter by a_0 (Equation 5.34):

$$\begin{aligned} H(z) &= a_0 + a_1 z^{-1} \\ &= \frac{a_0}{a_0} + \frac{a_1}{a_0} z^{-1} \\ &= a_0 \left[1 + \frac{a_1}{a_0} z^{-1} \right] \\ \text{Let } \alpha_1 &= \frac{a_1}{a_0} \\ H(z) &= a_0 [1 + \alpha_1 z^{-1}] \end{aligned} \tag{5.34}$$

By normalizing by a_0 and using the α_1 variable you can produce a transfer function that looks basically the same in the polynomial but pulls a_0 out as a scalar multiplier—a gain control. Where is the zero of the new transfer function in Equation 5.35?

$$\begin{aligned} H(z) &= a_0 [1 + \alpha_1 z^{-1}] \\ &= a_0 \left[1 + \frac{\alpha_1}{z} \right] \end{aligned} \tag{5.35}$$

Once again, by inspection we can tell that if $z = -\alpha_1$ then the function will become 0 regardless of the value of a_0. This transfer function has a zero at $-\alpha_1$. If you plug our values of $a_0 = a_1 = 0.5$, you still get the same zero at −1.0. The difference is in the gain of the filter.

After extracting the magnitude response from the z-plane plot, scale it by your a_0 value. This makes the response in Figure 5.24 match our others because everything gets multiplied by $a_0 = 0.5$. The idea of controlling the gain independently of the magnitude response is useful in audio filters, so we will keep it and use it in all future analyses.

At this point, you have all the DSP theory tools you need to understand the rest of the classical DSP filters (first-order feed back, second-order feed forward and feed back) as well as many filter design techniques. The rest of the chapter will be devoted to applying these same fundamentals to the other classical DSP filters but we will move much more quickly, applying each analysis technique to the other algorithms. For example, we will dispense with the evaluation of $e^{j\omega}$ terms and start off directly in the z transform of the difference equations.

> The *graphical interpretation* method of evaluating a filter in the z-plane assumes the filter is normalized so that $a_0 = 1.0$.

5.15 First-Order Feed-Back Filter Revisited

Now let's go through the same analysis technique on the first-order feed-back filter from the last chapter. We can move much more quickly now that we have the basic DSP theory down. There will be many similarities but also several key differences when dealing with feed-back designs. You already saw that the feed-back topology can blow up or ring forever and that the feed-forward design cannot. We will find a way to figure out if this is going to happen and how to prevent it. Start with the original first-order feed-back filter (block diagram in Figure 5.25) and its difference equation.

The difference equation is as follows:

$$y(n) = a_0x(n) - b_1y(n-1) \tag{5.36}$$

Step 1: Take the z transform of the difference equation

This can be done by inspection, using the rules from Section 5.8 (Figure 5.26). Therefore, the z transform is shown in Equation 5.37.

$$Y(z) = a_0X(z) - b_1Y(z)z^{-1} \tag{5.37}$$

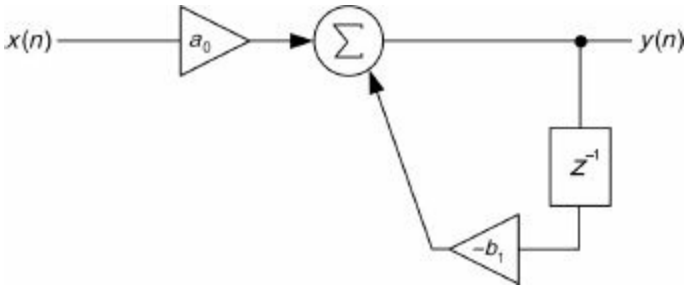


Figure 5.25: The first-order feed-back filter.

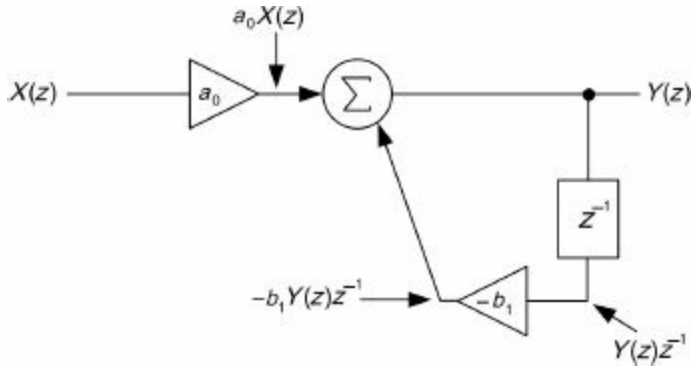


Figure 5.26: Pushing the input $X(z)$ through the algorithm produces the z transform.

Step 2: Fashion the difference equation into a transfer function

Now apply some algebra to convert the transformed difference equation to $H(z)$. The process is always the same: separate the $X(z)$ and $Y(z)$ variables, then form their quotient (Equation 5.38)

$$Y(z) = a_0X(z) - b_1Y(z)z^{-1}$$

Separate variables:

$$\begin{aligned} Y(z) + b_1Y(z)z^{-1} &= a_0X(z) \\ Y(z)[1 + b_1z^{-1}] &= a_0X(z) \\ Y(z) &= \frac{a_0X(z)}{1 + b_1z^{-1}} \end{aligned} \tag{5.38}$$

From $H(z)$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a_0}{1 + b_1z^{-1}}$$

Step 3: Factor out a_0 as the scalar gain coefficient

In this case, this step is simple since pulling a_0 out is trivial, as in Equation 5.39. However, in more complex filters this requires making substitutions as you did in the last section.

$$\begin{aligned} H(z) &= \frac{a_0}{1 + b_1z^{-1}} \\ &= a_0\frac{1}{1 + b_1z^{-1}} \end{aligned} \tag{5.39}$$

5.16 The Poles *of the Transfer Function*

The next step in the sequence is to do a quick estimation of the frequency response using the graphical interpretation method in the z-plane. The pure feed-forward filter you analyzed produced zeros of transmission or zeros at frequencies where its output becomes zero. A pure feed-back filter produces *poles* at frequencies where its output becomes infinite. We were able to make this happen by applying 100% feed-back in the last chapter. For the simple first-order case, finding the poles is done by

inspection.

> When the denominator of the transfer function is zero, the output is infinite. The complex frequency where this occurs is the pole frequency or pole.

Examining the transfer function, we can find the single pole in Equation 5.40:

$$\begin{aligned} H(z) &= a_0 \frac{1}{1 + b_1 z^{-1}} \\ &= a_0 \frac{1}{1 + \frac{b_1}{z}} \\ &= a_0 \frac{z}{z + b_1} \end{aligned} \tag{5.40}$$

By rearranging the transfer function, you can see that the denominator will be zero when $z = -b_1$ and so there is a pole at $z = -b_1$. You might also notice something interesting about this transfer function—it has a z in the numerator. If $z = 0$, then this transfer function has a zero at $z = 0$. This zero is called a *trivial zero* because it has no impact on the filter's frequency response. Thus, you can ignore the zero at $z = 0$. In fact, you can also ignore poles at $z = 0$ for the same reason.

And, if you look back at the transfer function in the feed-forward filter in Equation 5.35 you can see that it also had a pole in Equation 5.41:

$$\begin{aligned} H(z) &= a_0[1 + \alpha_1 z^{-1}] \\ &= a_0\left[1 + \frac{\alpha_1}{z}\right] \end{aligned} \tag{5.41}$$

Pole at $z = 0$

The poles are plotted in the z-plane in the same manner as the zeros but you use an **x** to indicate the pole frequency. In Equation 5.40, the pole is at $-b_1 + j0$ and so it is a real pole located on the real axis in the z-plane. For this filter let's analyze it with $a_0 = 1.0$ and $b_1 = 0.9$. You wrote a plug-in and implemented this filter in the last chapter. The results are shown in Figure 5.27.

Let's see how this is estimated first, and then we can do a direct evaluation as before.

Step 4: Estimate the frequency response

The single pole is plotted on the real axis at $z = -0.9 + j0$ and a trivial zero at $z = 0 + j0$ (Figure 5.28). In the future, we will ignore the trivial zeros or poles.

> A pole or zero at $z = 0$ is trivial and can be ignored for the sake of analysis since it has no effect on the frequency response.

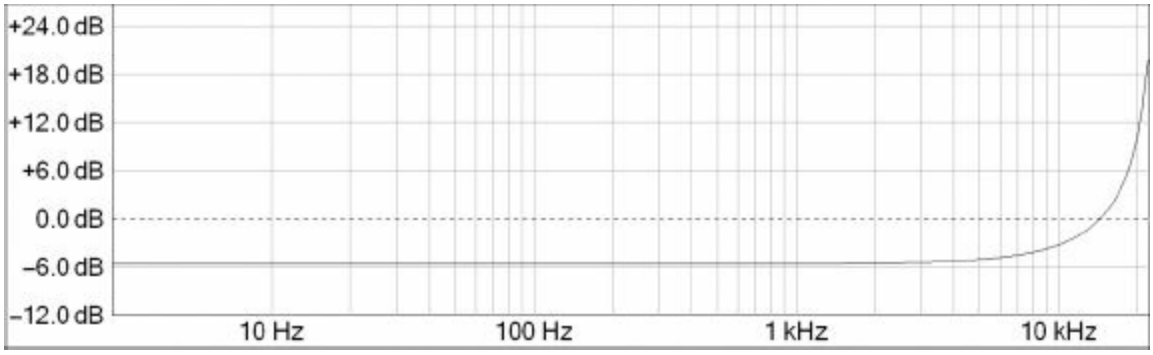


Figure 5.27: The frequency response of the first-order feed-back filter with these coefficients.

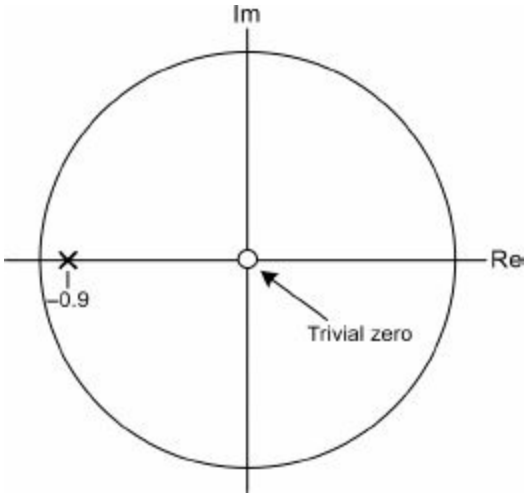


Figure 5.28: The pole is plotted in the *z*-plane along with the trivial zero.

In the simplest case of only one pole, the method for estimating the frequency response is as follows:

- Locate each evaluation frequency on the outer rim of the unit circle.
- Draw a line from the point on the circle to the pole and measure the length of this vector. Do it for each evaluation frequency.
- The *inverse of the lengths* of the lines will be the *magnitudes* at each frequency in the frequency response in Equation 5.42.

$$\text{Magnitude} = \frac{1}{L_p} \qquad (5.42)$$

L_p = length from evaluation frequency to pole

Thus, the mechanism is the same as for the one-zero case, except you take the inverse of the length of the vector. This means that as you near the pole, the vector becomes shorter, but the amplitude becomes larger—exactly opposite of the zero case. You can see from Figure 5.29 that our estimate is pretty close to what we saw in the real world when we coded the filter and tested it. Note the gain at Nyquist is 10.0; convert that to dB and you get +20 dB of gain, which is what we expect.

Step 5: Direct evaluation of frequency response

Now you can evaluate the filter the same way as before using Euler's equation to separate the real and imaginary components from the transfer function. Evaluate at the following frequencies:

- DC: 0
- Nyquist: π
- ½ Nyquist: π/2
- ¼ Nyquist: π/4

First, get the transfer function in a form from Equation 5.43 to use for all the evaluation frequencies:

$$\begin{aligned} H(z) &= a_0 \frac{1}{1 + b_1 z^{-1}} \\ a_0 &= 1.0 \quad b_1 = 0.9 \\ H(z) &= \frac{1}{1 + 0.9 z^{-1}} \\ z &= e^{j\omega} \end{aligned} \tag{5.43}$$

Next, let's make a simplification in the magnitude response equation and use the reduced form in Equation 5.44. Next, evaluate.

$$\begin{aligned} |a + jb| &= \sqrt{(a + jb)(a - jb)} \\ &= \sqrt{a^2 + b^2} \end{aligned} \tag{5.44}$$

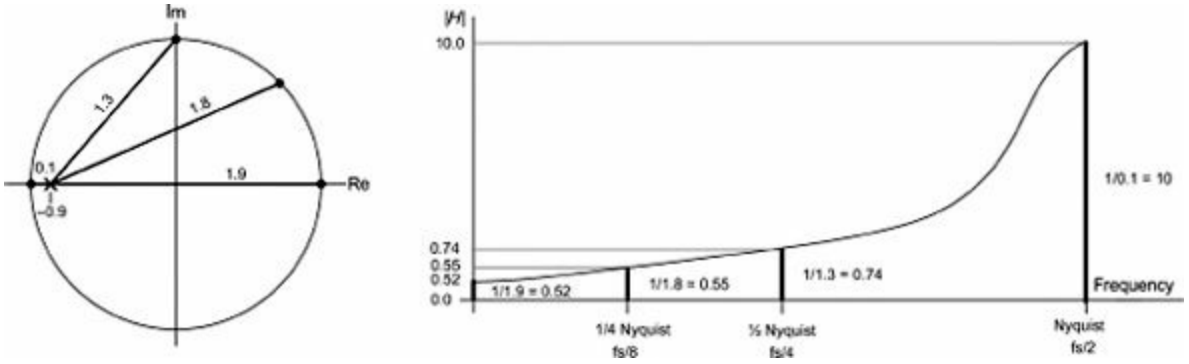


Figure 5.29: Estimating the frequency response of the first-order feed-back design.

5.16.1 DC (0 Hz)

$$\begin{aligned} H(\omega) &= \frac{1}{1 + 0.9e^{-j\omega 1}} \\ &= \frac{1}{1 + 0.9[\cos(\omega) - j\sin(\omega)]} \\ &= \frac{1}{1 + 0.9[\cos(0) - j\sin(0)]} \\ &= \frac{1}{1 + 0.9[1 - j0]} \\ &= \frac{1}{1.9} \\ &= 0.526 + j0 \end{aligned} \tag{5.45}$$

$$
\begin{aligned}
|H(\omega)| &= \sqrt{a^2 + b^2} \\
&= \sqrt{0.526^2 + 0^2} \\
&= 0.526 \\
\text{Arg}(H) &= \tan^{-1}(b/a) \\
&= \tan^{-1}(0/0.526) \\
&= 0.0^\circ
\end{aligned}
\tag{5.46}
$$

5.16.2 Nyquist (π)

$$
\begin{aligned}
H(\omega) &= \frac{1}{1 + 0.9e^{-j\omega 1}} \\
&= \frac{1}{1 + 0.9[\cos(\omega) - j\sin(\omega)]} \\
&= \frac{1}{1 + 0.9[\cos(\pi) - j\sin(\pi)]} \\
&= \frac{1}{1 + 0.9[-1 - j0]} \\
&= \frac{1}{0.1} \\
&= 10 + j0
\end{aligned}
\tag{5.47}
$$

$$
\begin{aligned}
|H(\omega)| &= \sqrt{a^2 + b^2} \\
&= \sqrt{10.0^2 + 0^2} \\
&= 10.0 \\
\text{Arg}(H) &= \tan^{-1}(b/a) \\
&= \tan^{-1}(0/10.0) \\
&= 0.0^\circ
\end{aligned}
\tag{5.48}
$$

5.16.3 ½ Nyquist (π/2)

$$
\begin{aligned}
H(\omega) &= \frac{1}{1 + 0.9e^{-j\omega 1}} \\
&= \frac{1}{1 + 0.9[\cos(\omega) - j\sin(\omega)]} \\
&= \frac{1}{1 + 0.9[\cos(\pi/2) - j\sin(\pi/2)]} \\
&= \frac{1}{1 + 0.9[0 - j1]} \\
&= \frac{1}{1 - j0.9}
\end{aligned} \tag{5.49}
$$

$$
\begin{aligned}
H(\omega) &= \frac{1}{1 - j0.9} \\
|H(\omega)| &= \frac{|1|}{|1 - j0.9|} \\
&= \frac{1}{\sqrt{a^2 + b^2}} \\
&= \frac{1}{\sqrt{1 + 0.81}} \\
&= 0.743 \\
\text{Arg}(H) &= \text{Arg(Num)} - \text{Arg(Denom)} \\
&= \tan^{-1}(0/1) - \tan^{-1}(-0.9/1) \\
&= +42^{\circ}
\end{aligned} \tag{5.50}
$$

5.16.4 ¼ Nyquist (π/4)

$$
\begin{aligned}
H(\omega) &= \frac{1}{1 + 0.9e^{-j\omega 1}} \\
&= \frac{1}{1 + 0.9[\cos(\omega) - j\sin(\omega)]} \\
&= \frac{1}{1 + 0.9[\cos(\pi/4) - j\sin(\pi/4)]} \\
&= \frac{1}{1 + 0.636 - j0.636} \\
&= \frac{1}{1.636 - j0.636}
\end{aligned} \tag{5.51}
$$

$$
\begin{aligned}
H(\omega) &= \frac{1}{1.636 - j0.636} \\
|H(\omega)| &= \frac{|1|}{|1.636 - j0.636|} \\
&= \frac{1}{\sqrt{1.636^2 + 0.636^2}} \\
&= 0.57 \\
\text{Arg}(H) &= \text{Arg(Num)} - \text{Arg(Denom)} \\
&= \tan^{-1}(0/1) - \tan^{-1}(-0.636/1.636) \\
&= +21^{\circ}
\end{aligned}
\tag{5.52}
$$

Make a special note about how we have to handle the magnitude of a fraction with numerator and denominator. You need to use the two equations in Equation 5.9 to deal with this. The main issue is that the phase is the difference of the Arg(numerator) − Arg(denominator). If the numerator was a complex number instead of 1.0, you would need to take the magnitude of it separately then divide. The final composite frequency/phase response plot is shown in Figure 5.30. You can see that the phase behaves linearly until it gets near the pole, and then it behaves nonlinearly. This is not a linear phase filter.

Step 6: *z* transform of impulse response

From Chapter 4 you will recall that the impulse response rings for this particular set of coefficients (Figure 5.31). Finding the impulse response by hand is going to be tedious. There are a lot of points to capture and it could take many lines of math before the impulse settles out. Fortunately, you can use RackAFX to do the work for you. The frequency and phase response plots are made using a *z* transform of the impulse response.

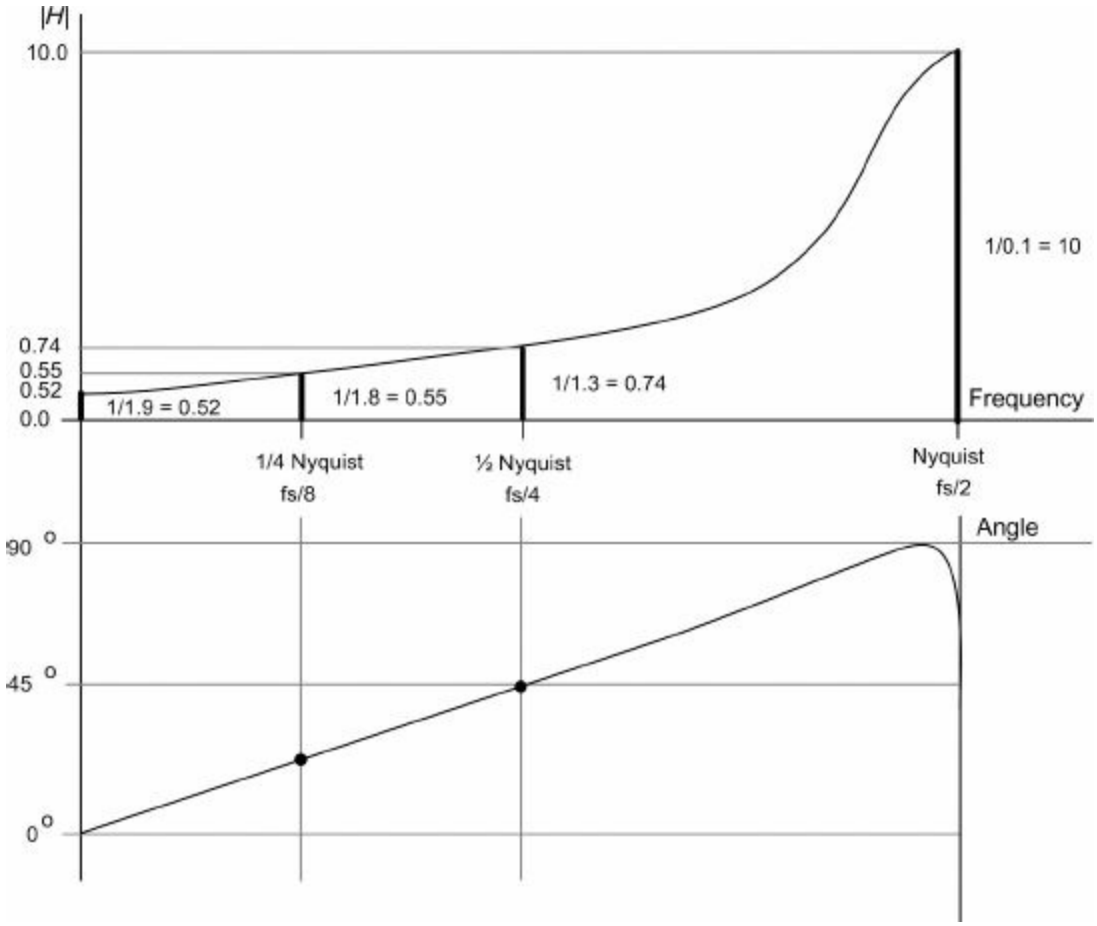


Figure 5.30: The final frequency and phase response plots.

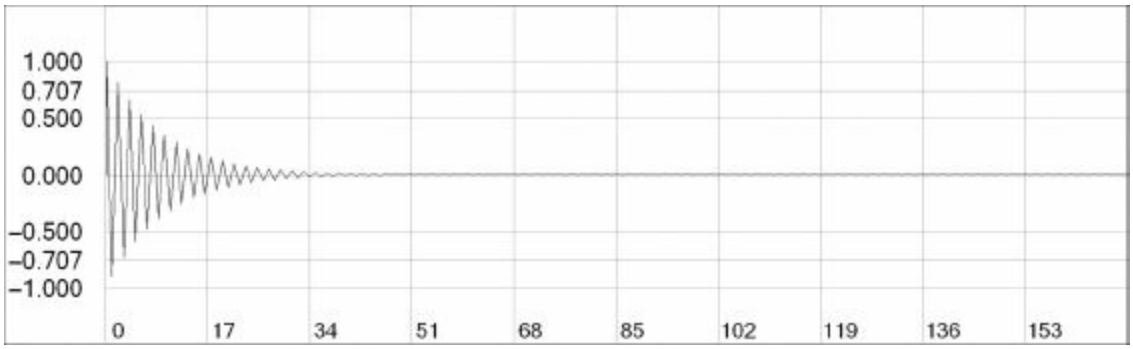


Figure 5.31: The impulse response of the filter in question.

In Figure 5.32 we observe excellent agreement with our evaluation; the response is down −6 dB at DC (0.52) and +20 dB at Nyquist (10) and the phase is 45 degrees at π/2. You followed six steps in evaluation of this filter:

1. Take the *z* transform of the difference equation.
2. Fashion the difference equation into a transfer function.
3. Factor out a_0 as the scalar gain coefficient.
4. Estimate the frequency response.
5. Direct evaluation of frequency response.
6. *z* transform of impulse response as a final check.

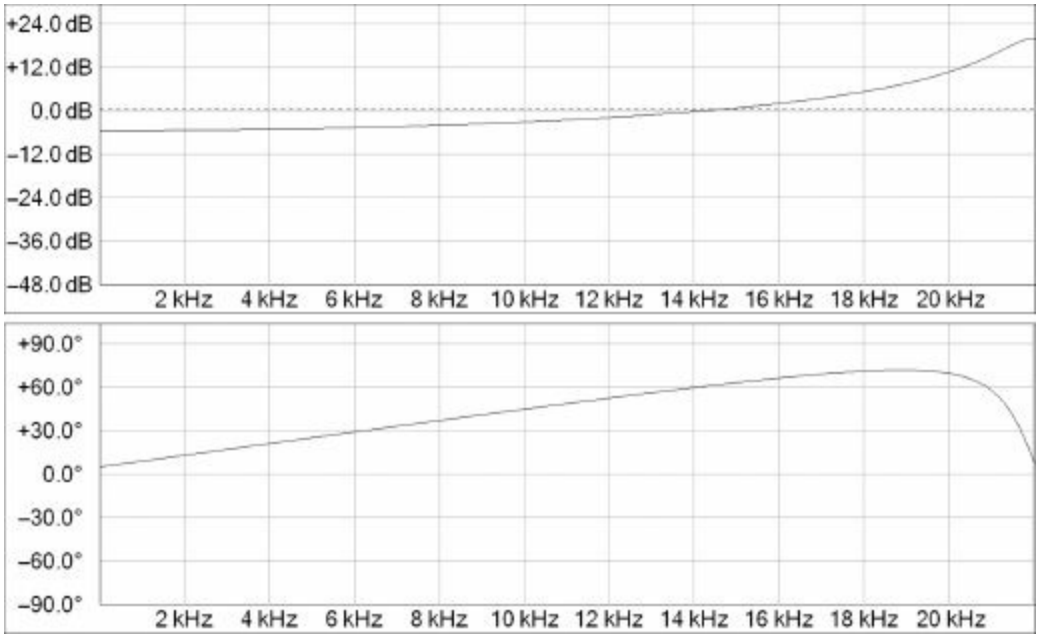


Figure 5.32: RackAFX's frequency and phase responses are taken from the *z* transform of the impulse response. Compare these measured results to the response we predicted using direct evaluation in Figure 5.30—notice these are plotted in dB rather than raw magnitudes.

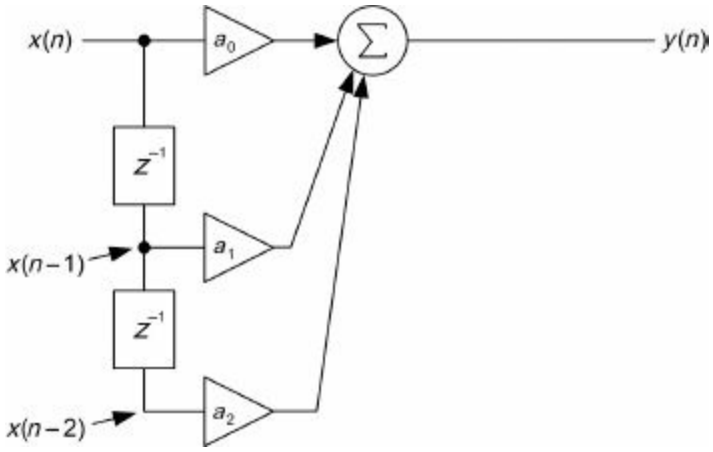


Figure 5.33: Second-order feed-forward filter.

5.17 Second-Order Feed-Forward Filter

Analysis of the second-order feed-forward filter proceeds much like the first-order filters you've seen so far, but there's a bit more math we have to deal with. The topology of a second-order feed-forward filter is shown in the block diagram in Figure 5.33.

The difference equation is as follows:

$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2) \tag{5.53}$$

Steps 1 & 2: Take the *z* transform of the difference equation and fashion it into a transfer function

We can combine steps to save time. The *z* transform can be taken by inspection, using the rules from Section 5.8, and then you need to get it into the form $Y(z)/X(z)$ for the transfer function in Equation

5.54.

$$\begin{aligned} y(n) &= a_0x(n) + a_1x(n-1) + a_2x(n-2) \\ Y(z) &= a_0X(z) + a_1X(z)z^{-1} + a_2X(z)z^{-2} \\ &= X(z)[a_0 + a_1z^{-1} + a_2z^{-2}] \end{aligned} \tag{5.54}$$

Form the transfer function $H(z)$:

$$H(z) = \frac{\text{output}}{\text{input}} = \frac{Y(z)}{X(z)} = a_0 + a_1z^{-1} + a_2z^{-2}$$

Step 3: Factor out a_0 as the scalar gain coefficient

We'll need to make some substitutions to get this in the form we are used to, shown in Equation 5.55:

$$\begin{aligned} H(z) &= a_0 + a_1z^{-1} + a_2z^{-2} \\ \text{Let } \alpha_1 &= \frac{a_1}{a_0} \quad \alpha_2 = \frac{a_2}{a_0} \\ H(z) &= a_0(1 + \alpha_1z^{-1} + \alpha_2z^{-2}) \end{aligned} \tag{5.55}$$

Step 4: Estimate the frequency response

First, this is a pure feed-forward filter, so you know there will only be nontrivial zeros; there are no poles to deal with. This transfer function is a second-order function because of the z^{-2} term. In fact, this is a quadratic equation. In order to find the poles or zeros, you need to first factor this equation and find the roots. The problem is that this is a complex equation, and the roots could be real, imaginary, or a combination of both. The mathematical break that we get is that our coefficients α_1 and α_2 are *real numbers*. The only way that could work out is if the locations of the zeros are complex conjugates of one another. When you multiply complex conjugates together the imaginary component disappears. So, with some algebra, you can arrive at the deduction shown in Equation 5.56:

$$H(z) = 1 + \alpha_1z^{-1} + \alpha_2z^{-2} \tag{5.56}$$

can be factored as

$$H(z) = (1 - Z_1z^{-1})(1 - Z_2z^{-1})$$

where

$$\begin{aligned} Z_1 &= \text{Re}^{j\theta} = a + jb \\ Z_2 &= \text{Re}^{-j\theta} = a - jb \end{aligned}$$

This analysis results in two zeros, Z_1 and Z_2, located at complex conjugate positions in the z-plane. Figure 5.34 shows an arbitrary conjugate pair of zeros plotted in the z-plane. You can see how they are at complementary angles to one another with the same radii. Remember, any arbitrary point in the z-plane is located at $\text{Re}^{j\theta}$ and the outer rim of the circle is evaluated for R = 1 and θ between $-\pi$ and $+\pi$ (or 0 to 2π).

But, how do the complex conjugate pair of zeros at $Re^{j\theta}$ and $Re^{-j\theta}$ relate to the coefficients α_1 and α_2? The answer is to just multiply everything out, use Euler's equation, and compare functions as shown in Equation 5.57:

$$\begin{aligned} H(z) &= a_0(1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}) \\ &= a_0(1 - Z_1 z^{-1})(1 - Z_2 z^{-1}) \end{aligned} \tag{5.57}$$

where

$$\begin{aligned} Z_1 &= Re^{j\theta} \\ Z_2 &= Re^{-j\theta} \\ (1 - Z_1 z^{-1})(1 - Z_2 z^{-1}) &= (1 - Re^{j\theta} z^{-1})(1 - Re^{-j\theta} z^{-1}) \\ &= 1 - Re^{j\theta} z^{-1} - Re^{-j\theta} z^{-1} + R^2 (e^{j\theta} e^{-j\theta}) z^{-2} \end{aligned}$$

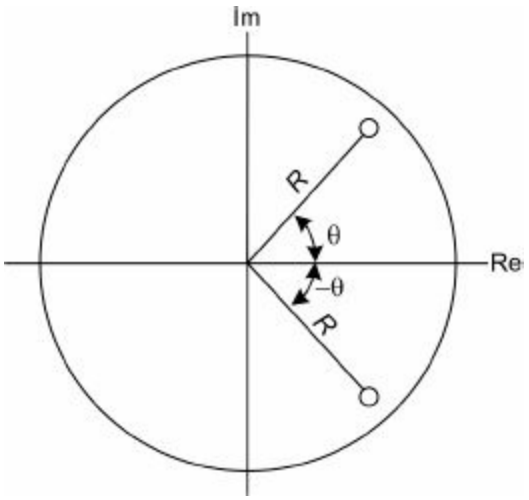


Figure 5.34: A complementary pair of zeros in the *z*-plane.

$$\begin{aligned} \text{noting that } (e^{j\theta} e^{-j\theta}) &= e^{j\theta - j\theta} = e^0 = 1 \\ &= 1 - (Re^{j\theta} + Re^{-j\theta}) z^{-1} + R^2 z^{-2} \\ &= 1 - R(\cos(\theta) + j\sin(\theta) + \cos(\theta) - j\sin(\theta)) z^{-1} + R^2 z^{-2} \\ &= 1 - 2R\cos(\theta) z^{-1} + R^2 z^{-2} \end{aligned}$$

compare functions:

$$\begin{aligned} H(z) &= a_0(1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}) \\ &= a_0(1 - 2R\cos(\theta) z^{-1} + R^2 z^{-2}) \end{aligned}$$

then

$$\begin{aligned} \alpha_1 &= -2R\cos(\theta) \\ \alpha_2 &= R^2 \end{aligned}$$

Equation 5.57 shows how the coefficients α_1 and α_2 create the zeros at the locations $Re^{j\theta}$ and $Re^{-j\theta}$. Once again you see that the coefficients *are* the filter—they determine the locations of the zeros, and these determine the frequency and phase responses of the filter. To estimate, we'll need some

coefficients to test with. Use the following: $a_0 = 1.0$, $a_1 = -1.27$, $a_2 = 0.81$. Now, calculate the location of the zeros from Equation 5.35; since $a_0 = 1.0$, then $\alpha_1 = -1.27$ and $\alpha_2 = 0.81$. Start with α_2 then solve for α_1 as shown in Equation 5.58. The zeros are plotted in Figure 5.35.

$$R^2 = \alpha_2 = 0.81 \qquad (5.58)$$
$$R = \sqrt{0.81} = 0.9$$

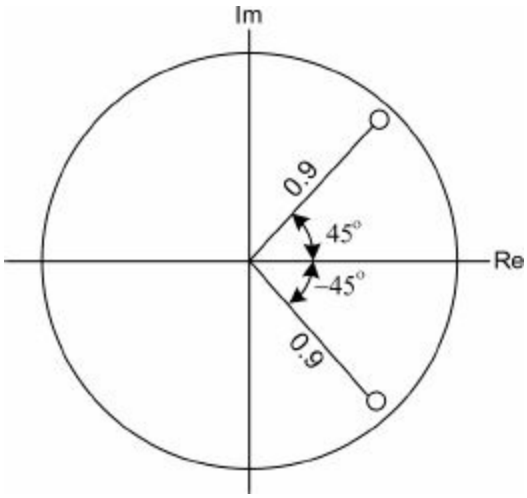


Figure 5.35: The complementary pair of zeros in the *z*-plane at radii 0.9 and angles ±45 degrees.

then

$$\begin{aligned} -2R\cos(\theta) &= -1.27 \\ 2(0.9)\cos(\theta) &= 1.27 \\ \cos(\theta) &= \frac{1.27}{2\,(0.9)} \\ \theta &= \arccos(0.705) \\ \theta &= 45^\circ \end{aligned}$$

Evaluating the frequency response of the complex pair is similar to before, but with an extra step. When estimating the frequency response with more than one zero:

- Locate each evaluation frequency on the outer rim of the unit circle.
- Draw a line from the point on the circle to *each* zero and measure the length of these vectors. Do it for each evaluation frequency.
- For each evaluation frequency, the magnitude of the transfer function is the *product of the two vectors* to each zero pair.

Mathematically, this last rule looks like Equation 5.59:

$$|H(e^{j\omega})|_\omega = a_0 \prod_{i=1}^{N} U_i \qquad (5.59)$$

where

N = filter order

U_i = geometric distance from the point ω on the unit circle to the *i*th pole

Follow the progression in Figures 5.36 through 5.39 through the four evaluation frequencies, starting at DC (0 Hz). Finally, put it all together in one plot and sketch the magnitude response in Figure 5.40.

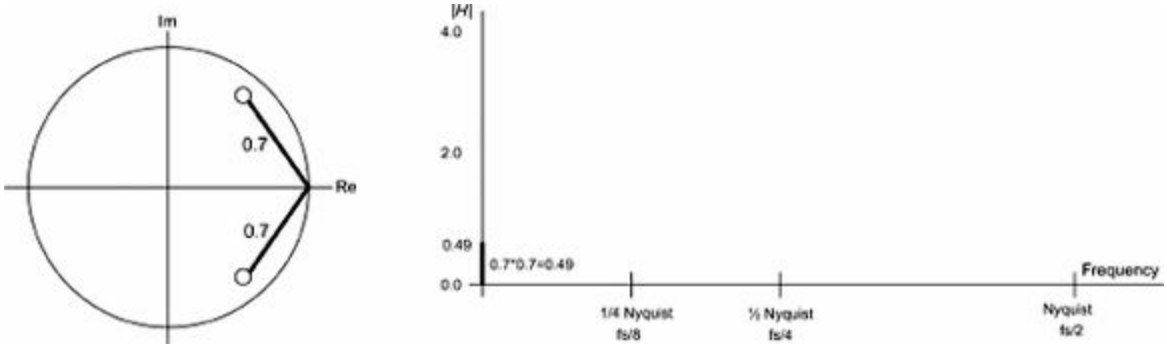


Figure 5.36: The magnitude response at 0 Hz (DC) is the product of the two vectors drawn to each zero, or 0.49.

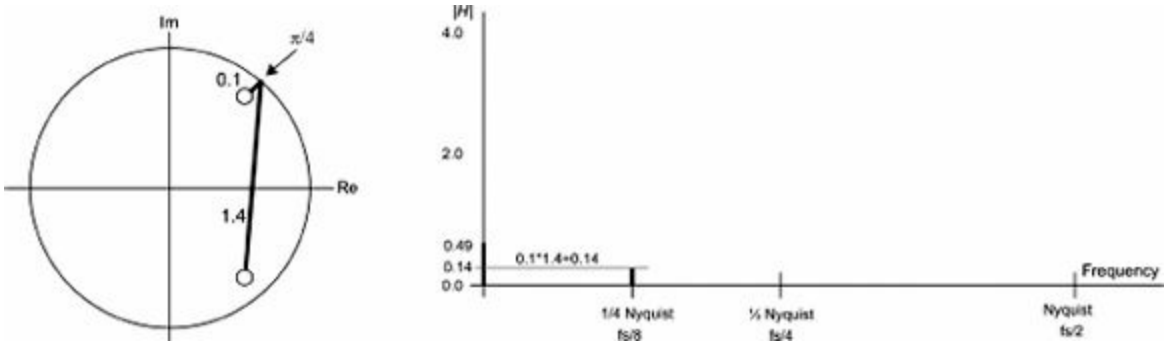


Figure 5.37: At ¼ Nyquist, the two vectors multiply out to 0.14.

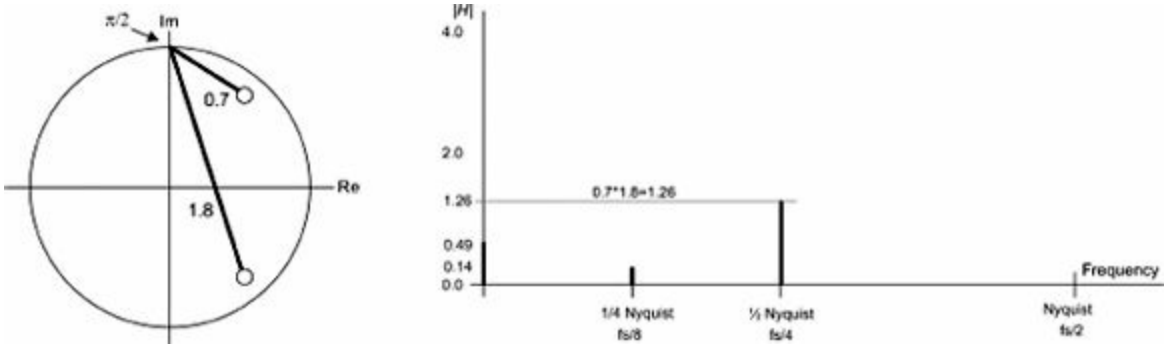


Figure 5.38: At ½ Nyquist, the response reaches 1.26 as the vectors begin to stretch out again.

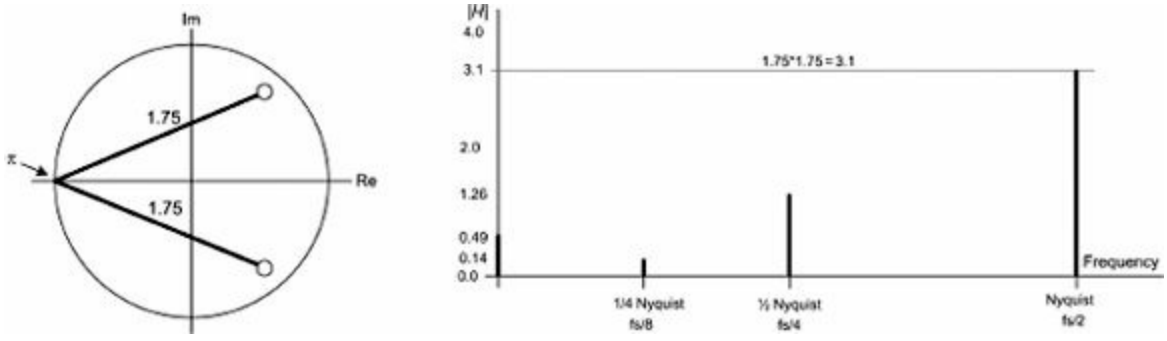


Figure 5.39: At Nyquist, the response reaches a maximum of 3.7 as the vectors stretch out to

their longest possible lengths.

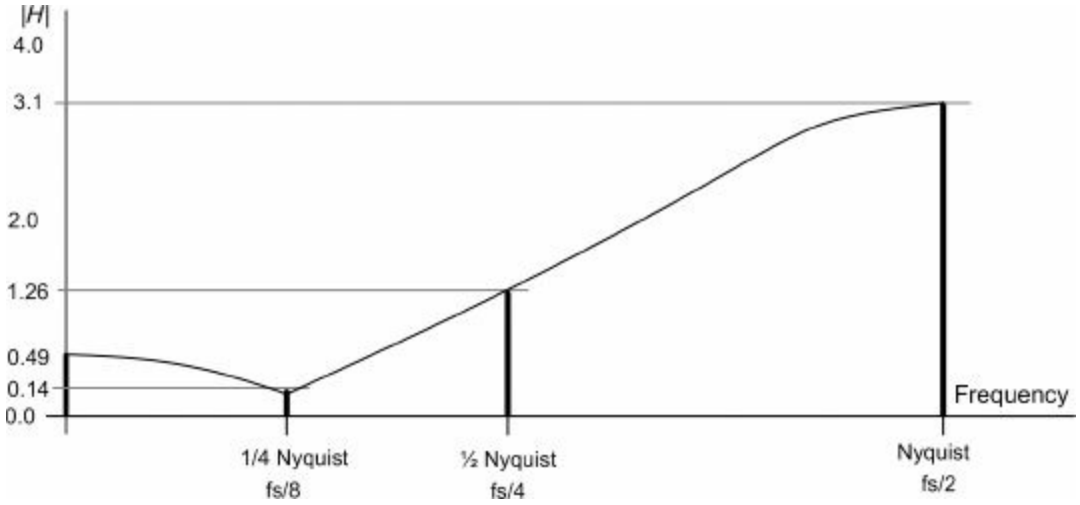


Figure 5.40: The combined response reveals a band-stop (notch) type of filter. The minimum amplitude occurs at the zero frequency, where the vector product is the lowest; this is where the smallest vector is obtained when evaluating on the positive frequency arc.

Step 5: Direct evaluation

Now you can evaluate the filter the same way as before using Euler's equation to separate the real and imaginary components from the transfer function. Evaluate at the following frequencies:

- DC: 0
- Nyquist: π
- ½ Nyquist: π/2
- ¼ Nyquist: π/4

First, get the transfer function in the form of Equation 5.60 to use for all the evaluation frequencies:

$$H(z) = a_0(1 + \alpha_1 z^{-1} + \alpha_2 z^{-2})$$

$$\text{Where: } \alpha_1 = \frac{a_1}{a_0} \quad \alpha_2 = \frac{a_2}{a_0}$$

$$H(z) = 1 - 1.27z^{-1} + 0.81z^{-2}$$

$$\text{Let: } z = e^{j\omega} \tag{5.60}$$

$$H(\omega) = 1 - 1.27e^{-j1\omega} + 0.81e^{-j2\omega}$$

Apply Euler's Equation:

$$H(\omega) = 1 - 1.27[\cos(\omega) - j\sin(\omega)] + 0.81[\cos(2\omega) - j\sin(2\omega)]$$

Now evaluate for each of our four frequencies in Equations 5.61 through 5.68.

5.17.1 DC (0 Hz)

$$
\begin{aligned}
H(\omega) &= 1 - 1.27[\cos(\omega) - j\sin(\omega)] + 0.81[\cos(2\omega) - j\sin(2\omega)] \\
&= 1 - 1.27[\cos(0) - j\sin(0)] + 0.81[\cos(2*0) - j\sin(2*0)] \\
&= 1 - 1.27[1 - j0] + 0.81[1 - j0] \\
&= 1 - 1.27 + 0.81 \\
&= 0.54 + j0
\end{aligned} \tag{5.61}
$$

$$
\begin{aligned}
|H(\omega)| &= \sqrt{a^2 + b^2} \\
&= \sqrt{0.54^2 + 0^2} \\
&= 0.54 \\
\text{Arg}(H) &= \tan^{-1}(b/a) \\
&= \tan^{-1}(0/0.54) \\
&= 0.0^{\circ}
\end{aligned} \tag{5.62}
$$

The exact magnitude is 0.54, which is pretty close to our estimated value of 0.49 using the *z*-plane graphical method.

5.17.2 Nyquist (π)

$$
\begin{aligned}
H(\omega) &= 1 - 1.27[\cos(\omega) - j\sin(\omega)] + 0.81[\cos(2\omega) - j\sin(2\omega)] \\
&= 1 - 1.27[\cos(\pi) - j\sin(\pi)] + 0.81[\cos(2\pi) - j\sin(2\pi)] \\
&= 1 - 1.27[-1 - j0] + 0.81[1 - j0] \\
&= 1 + 1.27 + 0.81 \\
&= 3.08
\end{aligned} \tag{5.63}
$$

$$
\begin{aligned}
|H(\omega)| &= \sqrt{a^2 + b^2} \\
&= \sqrt{3.08^2 + 0^2} \\
&= 3.08 \\
\text{Arg}(H) &= \tan^{-1}(b/a) \\
&= \tan^{-1}(0/3.08) \\
&= 0.0^{\circ}
\end{aligned} \tag{5.64}
$$

The exact magnitude is 3.08, which is pretty close to our estimated value of 3.1 using the *z*-plane graphical method.

5.17.3 ½ Nyquist (π/2)

$$
\begin{aligned}
H(\omega) &= 1 - 1.27[\cos(\omega) - j\sin(\omega)] + 0.81[\cos(2\omega) - j\sin(2\omega)] \\
&= 1 - 1.27[\cos(\pi/2) - j\sin(\pi/2)] + 0.81[\cos(2\pi/2) - j\sin(2\pi/2)] \\
&= 1 - 1.27[\cos(\pi/2) - j\sin(\pi/2)] + 0.81[\cos(\pi) - j\sin(\pi)] \\
&= 1 - 1.27[0 - j1] + 0.81[-1 - j0] \\
&= 1 + j1.27 - 0.81 \\
&= 0.19 + j1.27
\end{aligned} \tag{5.65}
$$

$$
\begin{aligned}
|H(\omega)| &= \sqrt{a^2 + b^2} \\
&= \sqrt{0.19^2 + 1.27^2} \\
&= 1.28 \\
\text{Arg}(H) &= \tan^{-1}(b/a) \\
&= \tan^{-1}(1.27/0.19) \\
&= 82^\circ
\end{aligned}
\tag{5.66}
$$

The exact magnitude is 1.28, which is pretty close to our estimated value of 1.26 using the *z*-plane graphical method.

5.17.4 ¼ Nyquist (π/4)

$$
\begin{aligned}
H(\omega) &= 1 - 1.27[\cos(\omega) - j\sin(\omega)] + 0.81[\cos(2\omega) - j\sin(2\omega)] \\
&= 1 - 1.27[\cos(\pi/4) - j\sin(\pi/4)] + 0.81[\cos(2\pi/4) - j\sin(2\pi/4)] \\
&= 1 - 1.27[\cos(\pi/4) - j\sin(\pi/4)] + 0.81[\cos(\pi/2) - j\sin(\pi/2)] \\
&= 1 - 1.27[0.707 - j0.707] + 0.81[-j1] \\
&= 0.11 + j0.08
\end{aligned}
\tag{5.67}
$$

$$
\begin{aligned}
|H(\omega)| &= \sqrt{a^2 + b^2} \\
&= \sqrt{0.11^2 + 0.08^2} \\
&= 0.136 \\
\text{Arg}(H) &= \tan^{-1}(b/a) \\
&= \tan^{-1}(0.08/0.11) \\
&= 36^\circ
\end{aligned}
\tag{5.68}
$$

The exact magnitude is 0.136, which is pretty close to our estimated value of 0.14 using the *z*-plane graphical method.

Step 6: *z* transform of impulse response

This second-order feed-forward filter is actually pretty easy to examine for its impulse response. For an impulse stimulus, the impulse response $h(n)$ is $h(n) = \{1.0, -1.27, 0.81\}$. Taking the *z* transform of the impulse response is easy, as shown in Equation 5.69:

$$
\begin{aligned}
H(z) &= \sum_{n=-\infty}^{n=+\infty} h(n)z^{-n} \\
&= 1.0z^0 - 1.27z^{-1} + 0.81z^{-2} \\
&= 1 - 1.27z^{-1} + 0.81z^{-2}
\end{aligned}
\tag{5.69}
$$

This is exactly what we expect. This should help you understand two more very important details about pure feed-forward filters.

In a pure feed-forward filter:

- The coefficients $\{a_0, a_1, a_2, \ldots\}$ are the impulse response, $h(n)$.
- The transfer function is the z transform of the coefficients.

Finally, I'll use RackAFX to verify the frequency and phase response from our analysis by using a plug-in I wrote for second-order feed-forward filters. Figure 5.41 shows the frequency and phase response plots.

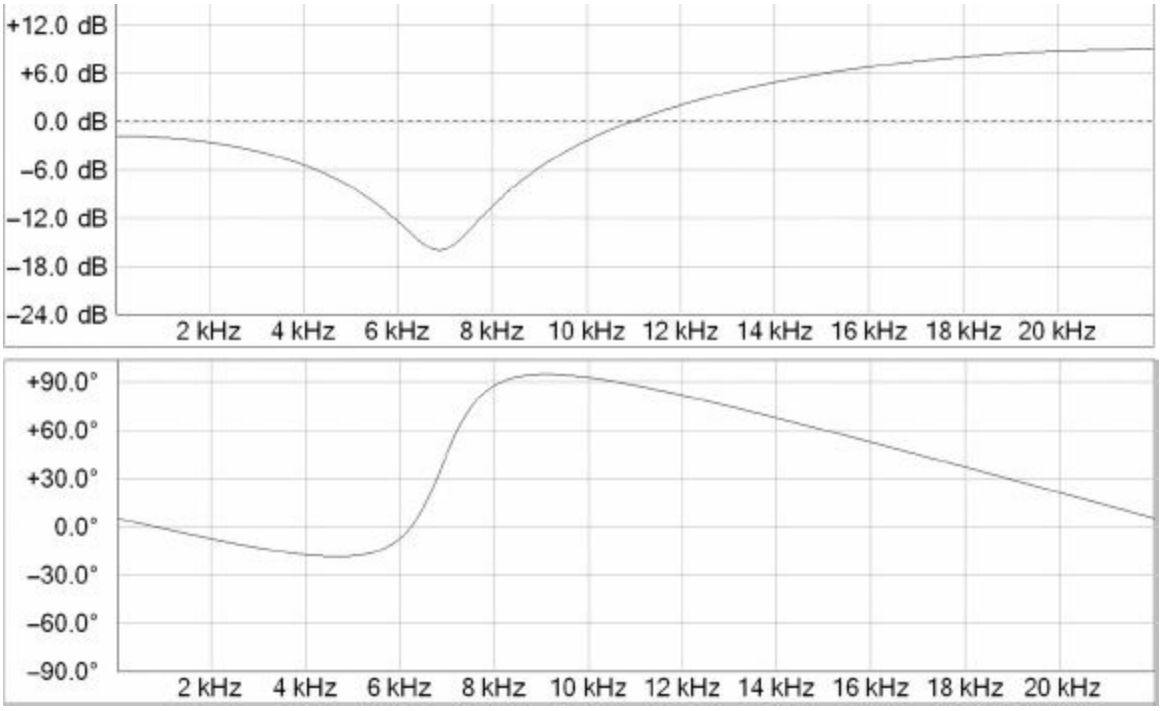


Figure 5.41: Plots using RackAFX's z-transform of the impulse response.

5.18 Second-Order Feed-Back Filter

Analysis of the second-order feed-back filter starts with the block diagram and difference equation. Figure 5.42 shows the topology of a second-order feed-back filter.

The difference equation is as follows:

$$y(n) = a_0x(n) - b_1y(n-1) - b_2y(n-2) \tag{5.70}$$

Steps 1 to 3: Take the z transform of the difference equation to get the transfer function, then factor out a_0 as the scalar gain coefficient

We'll continue to combine steps. Once again, the z transform can be taken by inspection using the rules from Section 5.8, and then you need to get it into the form $Y(z)/X(z)$ for the transfer function in Equation 5.71.

$$y(n) = a_0x(n) - b_1y(n-1) - b_2y(n-2)$$
$$Y(z) = a_0X(z) - b_1Y(z)z^{-1} - b_2Y(z)z^{-2}$$

Separate variables:

$$Y(z) + b_1 Y(z)z^{-1} + b_2 Y(z)z^{-2} = a_0 X(z)$$
$$Y(z)[1 + b_1 z^{-1} + b_2 z^{-2}] = a_0 X(z) \tag{5.71}$$

Form transfer function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a_0}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

Factor out a_0:

$$H(z) = a_0 \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

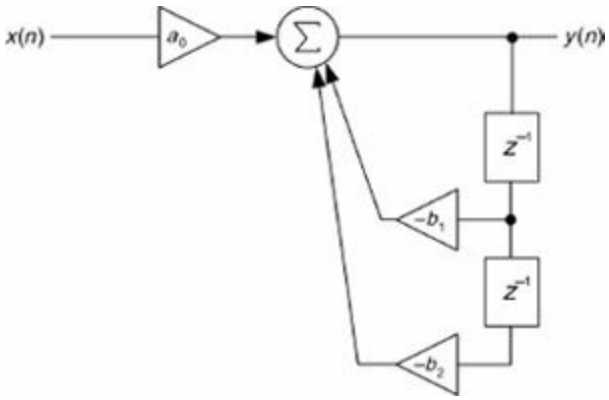


Figure 5.42: Second-order feed-back filter.

Step 4: Estimate the frequency response

Now you can see why the feed-back filter block diagrams have all the *b* coefficients negated. It puts the quadratic denominator in the final transfer function in Equation 5.71 in the same polynomial form as the numerator of the feed-forward transfer function. Thus, you can use the same logic to find the poles of the filter; since the coefficients b_1 and b_2 are real values, the poles must be complex conjugates of each other, as shown in Equation 5.72:

$$H(z) = a_0 \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

can be factored as

$$H(z) = a_0 \frac{1}{(1 - P_1 z^{-1})(1 - P_2 z^{-1})}$$

where

$$P_1 = \mathrm{R}e^{j\theta}$$
$$P_2 = \mathrm{R}e^{-j\theta}$$
$$(1 - P_1 z^{-1})(1 - P_1 z^{-1}) = 1 - 2R\cos(\theta)z^{-1} + R^2 z^{-2} \tag{5.72}$$

therefore

$$H(z) = a_0 \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}}$$
$$= a_0 \frac{1}{(1 - 2R\cos(\theta) z^{-1} + R^2 z^{-2})}$$

and

$$b_1 = -2R\cos(\theta)$$
$$b_2 = R^2$$

This results in two poles, P_1 and P_2, located at complex conjugate positions in the z-plane.

Figure 5.43 shows an arbitrary conjugate pair of poles plotted in the z-plane. You can see how they are at complementary angles to one another with the same radii.

To estimate we'll need some coefficients to test with. Use the following: $a_0 = 1.0$, $b_1 = -1.34$, $b_2 = 0.902$. Now, calculate the location of the poles from Equation 5.72:

$$R^2 = b_2 = 0.902 \tag{5.73}$$
$$R = \sqrt{0.902} = 0.95$$

then

$$-2R\cos(\theta) = -1.34$$
$$2(0.95)\cos(\theta) = 1.34$$

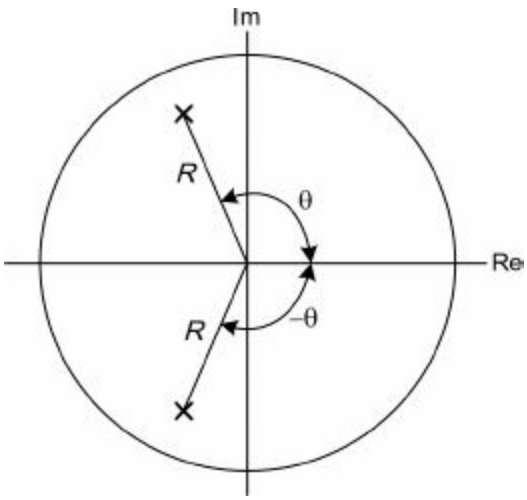


Figure 5.43: A complementary pair of poles in the z-plane.

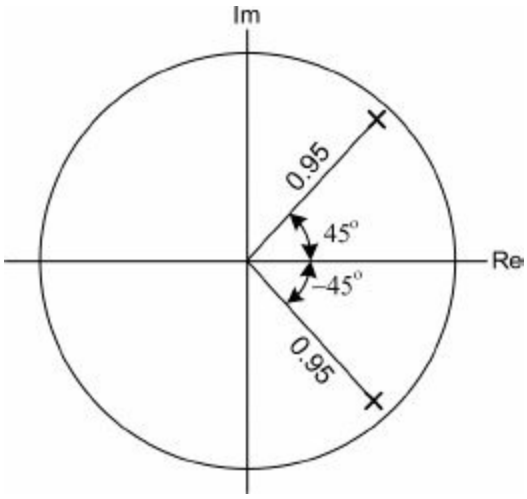


Figure 5.44: The poles of the filter.

$$\cos(\theta) = \frac{1.34}{2(0.95)}$$
$$\theta = \arccos(0.707)$$
$$\theta = 45°$$

Figure 5.44 shows the complex conjugate pair of poles plotted in the *z*-plane at angles ±45° and radii of 0.95. Evaluating the frequency response of the complex pair is similar to before, but with an extra step. When estimating the frequency response with more than one pole:

- Locate each evaluation frequency on the outer rim of the unit circle.
- Draw a line from the point on the circle to *each* pole and measure the length of these vectors. Do it for each evaluation frequency.
- For each evaluation frequency, the magnitude of the transfer function is the *product of the inverse lengths* of the two vectors to each pole pair.

Mathematically, this last rule looks like Equation 5.74:

$$|H(e^{j\omega})|_{\omega} = a_0 \frac{1}{\prod_{i=1}^{N} V_i} \tag{5.74}$$

where

N = the filter order

V_i = the geometric length from the point(ω) on the unit circle to the *i*th pole

Thus, the process is the same as with the zeros, except that you take the inverse of the length to the pole.

For feed-forward filters:

- The closer the frequency is to the zero, the more attenuation it receives.
- If the zero is *on* the unit circle, the magnitude would go to zero at that point.

For feed-back filters:

- The closer the evaluation frequency is to the pole, the more gain it receives.
- If a pole is *on* the unit circle, the magnitude would theoretically go to infinity, and it would produce an oscillator, ringing forever at the pole frequency.

You blew up the first-order feed-back filter as an exercise in Chapter 4. All feedback filters are prone to blowing up when their poles go outside the unit circle. We can now continue with the estimation process for our standard four evaluation frequencies. This time, we'll convert the raw magnitude values into dB. The reason for doing this is that there will be a very wide range of values that will be difficult to sketch if we don't use dB. Follow the evaluation sequence in Figures 5.45 through 5.48. Finally, you can put it all together to form the frequency response plot in Figure 5.49.

In a digital filter:

- Zeros may be located anywhere in the *z*-plane, inside, on, or outside the unit circle since the filter is always stable; it's output can't go lower than 0.0.
- Poles must be located inside the unit circle.
- If a pole is on the unit circle, it produces an oscillator.
- If a pole is outside the unit circle, the filter blows up as the output goes to infinity.

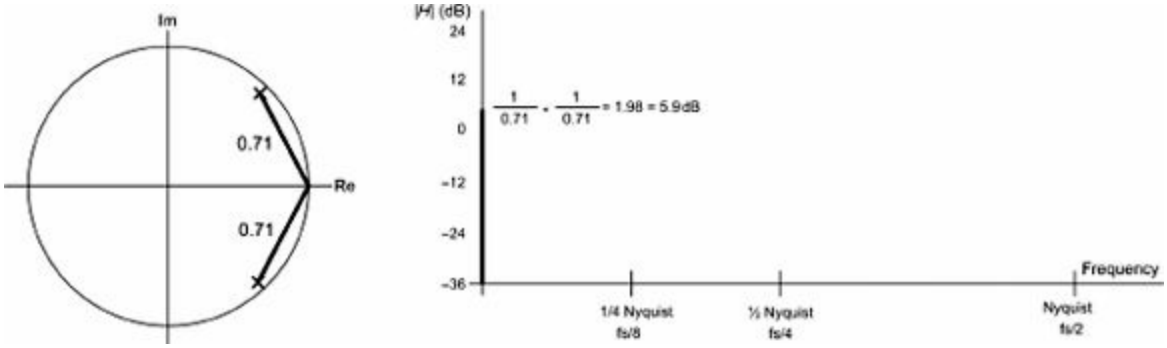


Figure 5.45: The magnitude response at 0 Hz (DC) is the product of the inverse of the two vectors drawn to each zero, or (1/0.71)(1/0.71) = 5.9 dB.

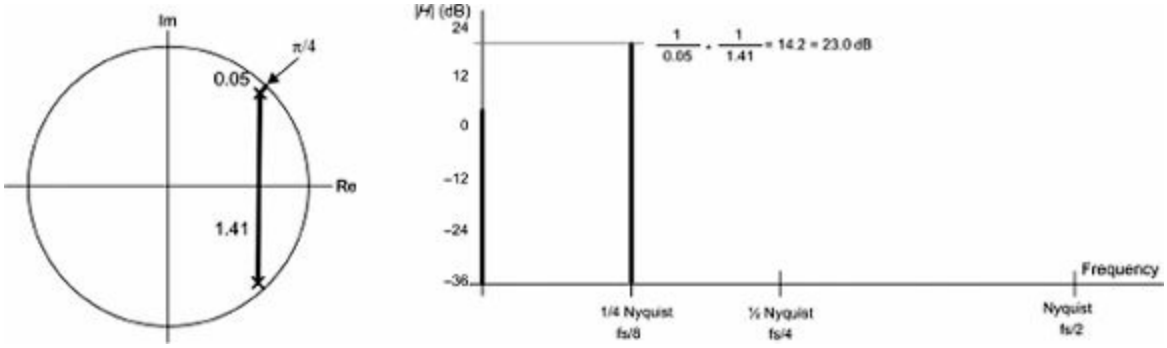


Figure 5.46: The magnitude response at 1/4 Nyquist is a whopping +23 dB since the inverse of 0.05 is a large number.

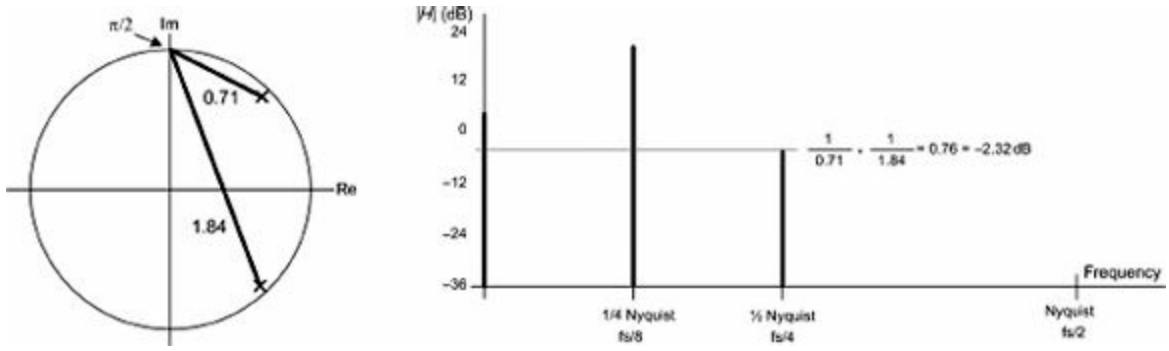


Figure 5.47: The magnitude response at π/2 is −2.98 dB.

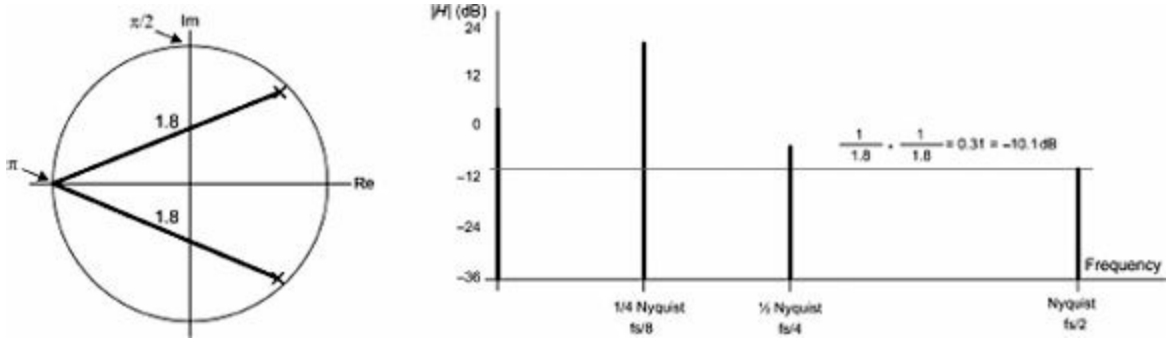


Figure 5.48: The magnitude response at Nyquist is −10.1 dB.

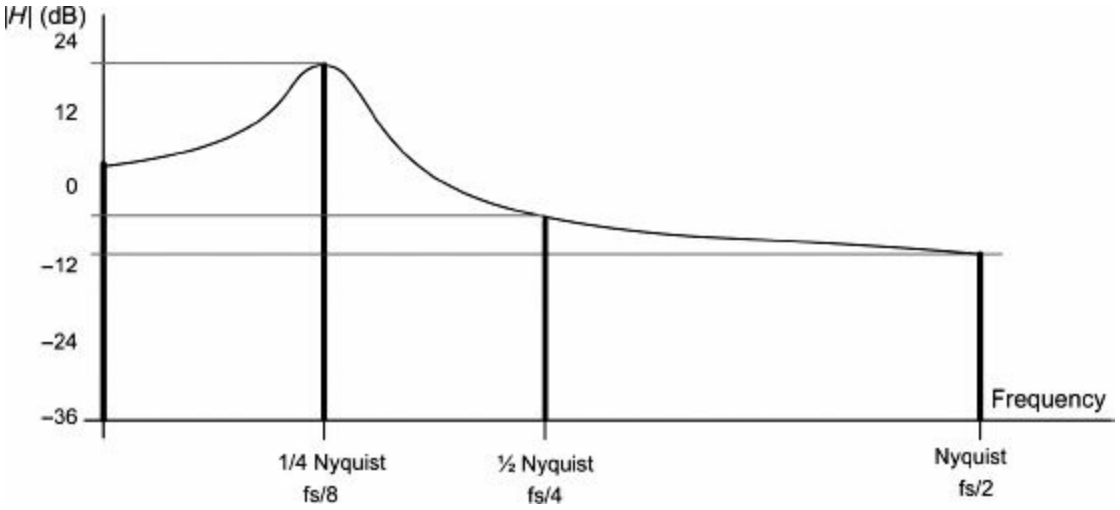


Figure 5.49: The composite magnitude response of the filter shows that it is a resonant low-pass filter; the resonant peak occurs at the pole frequency.

Step 5: Direct evaluation

Now you can evaluate the filter the same way as before using Euler's equation to separate the real and imaginary components from the transfer function. Evaluate at the following frequencies:

- DC: 0
- Nyquist: π
- ½ Nyquist: π/2
- ¼ Nyquist: π/4

First, get the transfer function in a form to use for all the evaluation frequencies:

$$H(z) = a_0 \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{5.75}$$
$$= \frac{1}{1 - 1.34 z^{-1} + 0.902 z^{-2}}$$
$$\text{Let } z = e^{j\omega}.$$
$$H(\omega) = \frac{1}{1 - 1.34 e^{-j1\omega} + 0.902 e^{-j2\omega}}$$

Apply Euler's equation: 1

$$H(\omega) = \frac{1}{1 - 1.34 e^{-j1\omega} + 0.902 e^{-j2\omega}}$$
$$H(\omega) = \frac{1}{1 - 1.34[\cos(\omega) - j\sin(\omega)] + 0.902[\cos(2\omega) - j\sin(2\omega)]}$$

Now evaluate for each of our four frequencies starting with DC.

5.18.1 DC (0 Hz)

$$H(\omega) = \frac{1}{1 - 1.34[\cos(\omega) - 1 - j0\, j\sin(\omega)] + 0.902[\cos(2\omega) - j\sin(2\omega)]}$$
$$= \frac{1}{1 - 1.34[\cos(0) - j\sin(0)] + 0.902[\cos(2*0) - j\sin(2*0)]}$$
$$= \frac{1}{1 - 1.34[1 - j0] + 0.902} \tag{5.76}$$
$$= \frac{1}{1 - 1.34 + 0.902}$$
$$= \frac{1}{0.562 + j0}$$

$$H(\omega) = \frac{1}{0.562 + j0}$$
$$|H(\omega)| = \frac{|1|}{|0.562 + j0|}$$
$$= \frac{1}{\sqrt{a^2 + b^2}} \tag{5.77}$$
$$= \frac{1}{\sqrt{0.562^2}} = 1.78 = 5.00 \text{ dB}$$
$$\text{Arg}(H) = \text{Arg(Num)} - \text{Arg(Denom)}$$
$$= \tan^{-1}(0/1) - \tan^{-1}(0/0.562)$$
$$= 0^\circ$$

Remember that for magnitudes of fractions, you need to take the magnitude of the numerator and denominator separately; also for phase, the total is the difference of the Arg(num) and Arg(denom). The direct evaluation yields 5.0 dB and shows our sketch evaluation was a little off at 5.9 dB.

Table 5.4: Challenge answers.

Frequency (ω)	$\|H(\omega)\|$	Arg(H)
Nyquist (π)	–10.2 dB	0.0°
½ Nyquist (π/2)	–2.56 dB	–85.8°
¼ Nyquist (π/4)	+123.15 dB	–40.3°

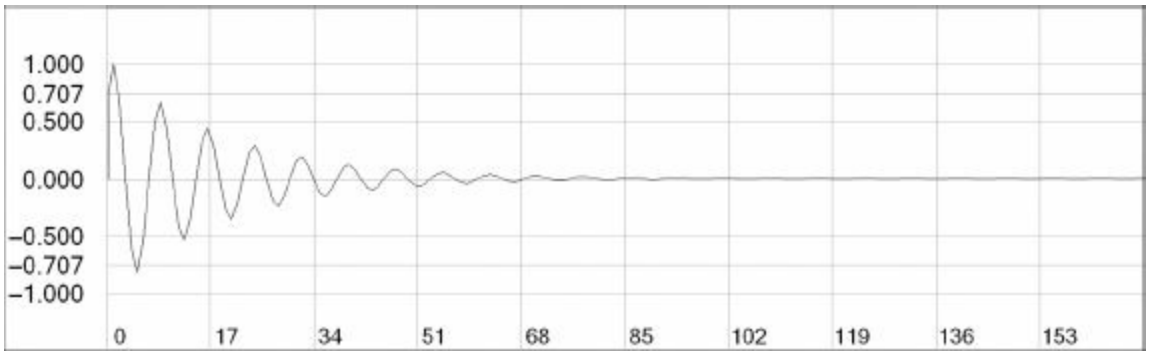


Figure 5.50: Impulse response of the filter from RackAFX.

5.18.2 Challenge

Finish the rest of the direct evaluation calculations on your own. The answers are in Table 5.4. The exact evaluation once again produces results pretty close to our estimated version. You can get the complete answers at the RackAFX websites.

Step 6: *z* transform of impulse response

The impulse response for this filter is shown in Figure 5.50. Once again, finding the impulse response by hand is going to be tedious. Instead, I will use RackAFX to do the analysis so we can compare our estimated and direct evaluation results. The measured responses are shown in Figure 5.51.

5.19 First-Order Pole-Zero Filter: The Shelving Filter

The first-order pole-zero filter consists of a first-order pole and first-order zero in the same algorithm. The topology in the block diagram in Figure 5.52 is a combination feed forward and feed back since it contains both paths.

The difference equation is as follows:

$$y(n) = a_0x(n) + a_1x(n-1) - b_1y(n-1) \tag{5.78}$$

Steps 1 to 3: Take the *z* transform of the difference equation to get the transfer function, then factor out a_0 as the scalar gain coefficient

$$Y(z) = a_0X(z) + a_1X(z)z^{-1} - b_1Y(z)z^{-1} \tag{5.79}$$

Separate variables:

$$Y(z) + b_1Y(z)z^{-1} = a_0X(z) + a_1X(z)z^{-1}$$
$$Y(z)[1 + b_1z^{-1}] = X(z)[a_0 + a_1z^{-1}]$$

Form the transfer function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1z^{-1}}{1 + b_1z^{-1}}$$

Factor out a_0:

$$H(z) = a_0\frac{1 + \alpha_1z^{-1}}{1 + b_1z^{-1}}$$

where

$$\alpha_1 = \frac{a_1}{a_0}$$

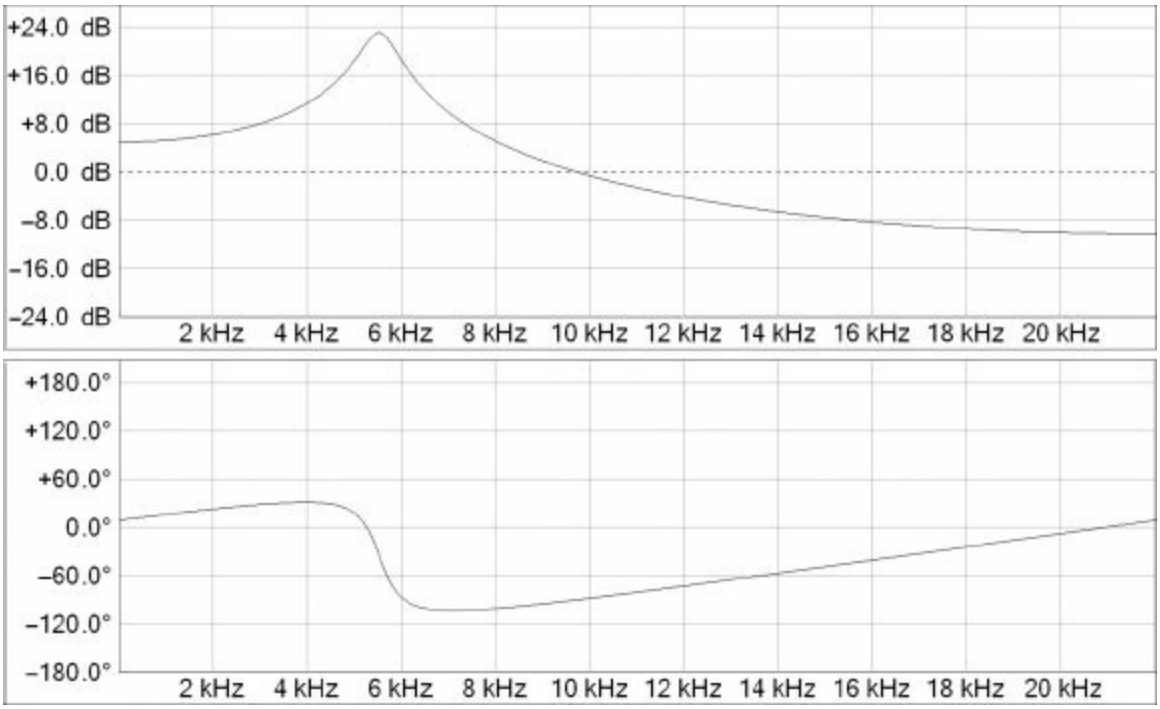


Figure 5.51: RackAFX's frequency and phase responses are taken from the *z*-transform of the impulse response.

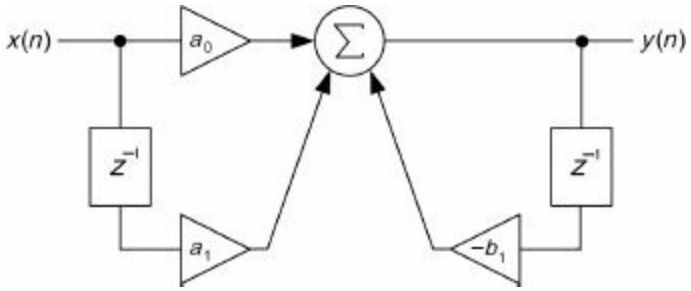


Figure 5.52: First-order pole-zero filter.

Step 4: Estimate the frequency response

This transfer function has one pole and one zero and both are first order. Like the other first-order cases, we can find the pole and zero by inspection of the transfer function:

$$H(z) = a_0 \frac{1 + \alpha_1 z^{-1}}{1 + b_1 z^{-1}} = a_0 \frac{1 + \frac{\alpha_1}{z}}{1 + \frac{b_1}{z}} \tag{5.80}$$

In the numerator, you can see that if $z = -\alpha_1$ the numerator will go to zero and the transfer function will go to zero. In the denominator, you can see that if $z = -b_1$ the denominator will go to zero and the transfer function will go to infinity. Therefore we have a zero at $z = -\alpha_1$ and a pole at $z = -b_1$. For this example, use the following values for the coefficients: $a_0 = 1.0$, $a_1 = -0.92$, $b_1 = -0.71$. Then, $\alpha_1 = -0.92$, and so we now have a zero at $z = -\alpha_1 = 0.92 + j0$ and a pole at $z = -b_1 = 0.71 + j0$. The pole/zero pair are plotted in Figure 5.53.

Evaluating the frequency response when you have mixed poles and zeros is the same as before, but you have to implement both magnitude steps.

- Locate each evaluation frequency on the outer rim of the unit circle.
- Draw a line from the point on the circle to *each* zero and measure the length of these vectors. Do it for each evaluation frequency.
- Draw a line from the point on the circle to *each* pole and measure the length of these vectors. Do it for each evaluation frequency.

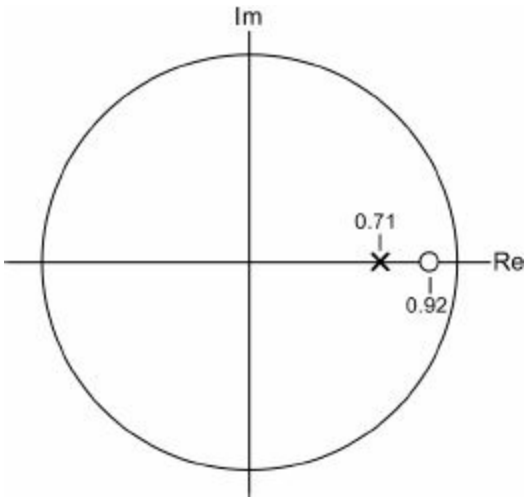


Figure 5.53: The pole and zero are both purely real and plotted on the real axis in the *z*-plane.

- Multiply all the zero magnitudes together.
- Multiply all the *inverse* pole magnitudes together.
- Divide the zero magnitude by the pole magnitude for the final result at that frequency.

Mathematically, this last rule looks like Equation 5.81:

$$|H(e^{j\omega})|_{\omega} = a_0 \frac{\prod_{i=1}^{N} U_i}{\prod_{i=1}^{N} V_i} \tag{5.81}$$

where

N = the filter order

U_i = the geometric length from the point(ω)on the unit circle to the ith zero

V_i = the geometric length from the point(ω)on the unit circle to the ith pole

Equation 5.81 is the final, generalized magnitude response equation for the geometric interpretation of the pole/zero plots in the z-plane. For completeness, here's the equation for calculating the phase response of a digital filter using the geometric method:

$$\text{Arg}(H(e^{j\omega}))|_{\omega} = \sum_{i=1}^{N} \theta_i - \sum_{i=1}^{N} \phi_i \tag{5.82}$$

where

N = the filter order

θ_i = the angle between the ith zero and the vector U_i

ϕ_i = the angle between the ith pole and the vector V_i

Equations 5.81 and 5.82 together complete the DSP theory for pole/zero interpretation for estimating the frequency and phase responses of any filter. So, let's do the analysis for this filter—by going through the math, you can see the tug of war going on between the pole and zero. Follow the analysis sequence in Figures 5.54 through 5.58.

Step 5: Direct evaluation

You can evaluate the filter the same way as before using Euler's equation to separate the real and imaginary components from the transfer function. Evaluate at the following frequencies:

- DC: 0
- Nyquist: π
- ½ Nyquist: π/2
- ¼ Nyquist: π/4

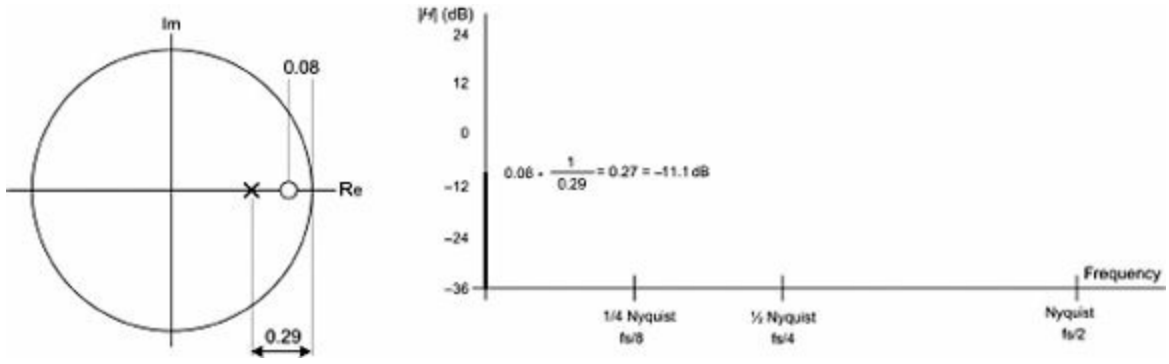


Figure 5.54: The magnitude response at DC is −11.1 dB. Look at the equation and you can see the zero value bringing down the total while the pole value is trying to push it back up. In this case, the zero wins and the response is down almost 12 dB. Geometrically, you can see this is because the zero is closer to the evaluation point and so it has more effect on the outcome.

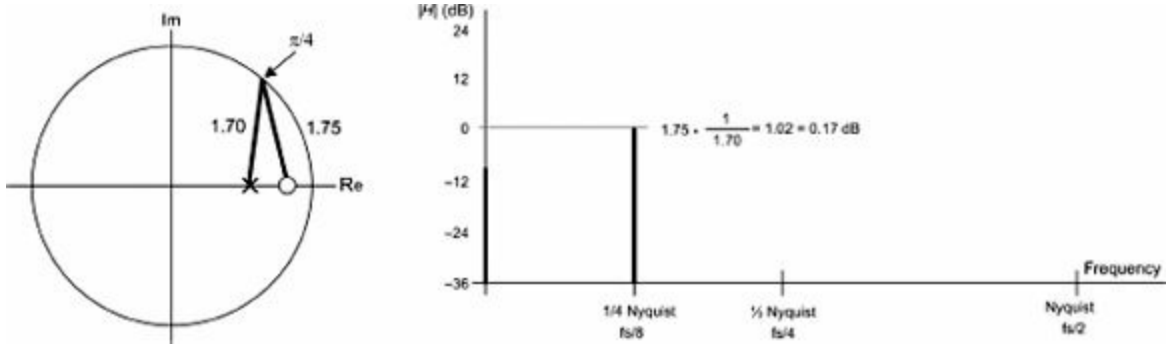


Figure 5.55: The magnitude response at π is almost unity gain because the pole and zero distances are almost the same. The tug of war ends in stalemate here at 0.17 dB of gain.

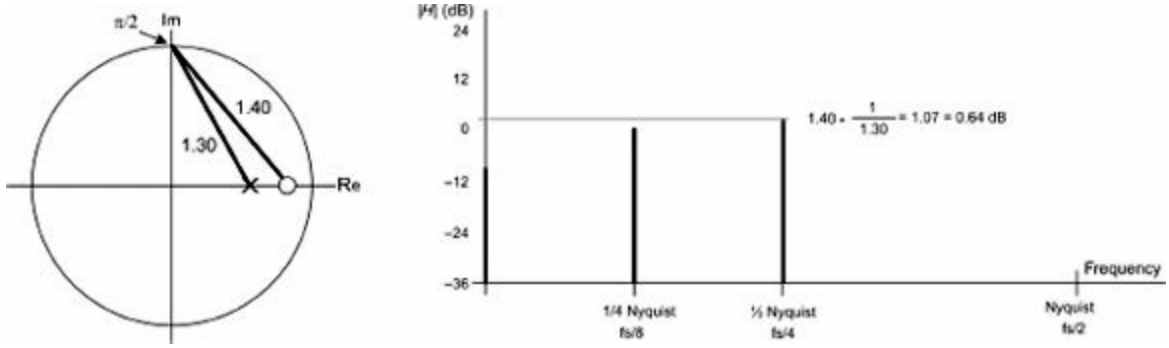


Figure 5.56: With the pole slightly closer to the evaluation frequency, the magnitude response at π/2 picks up a bit to +0.64 dB.

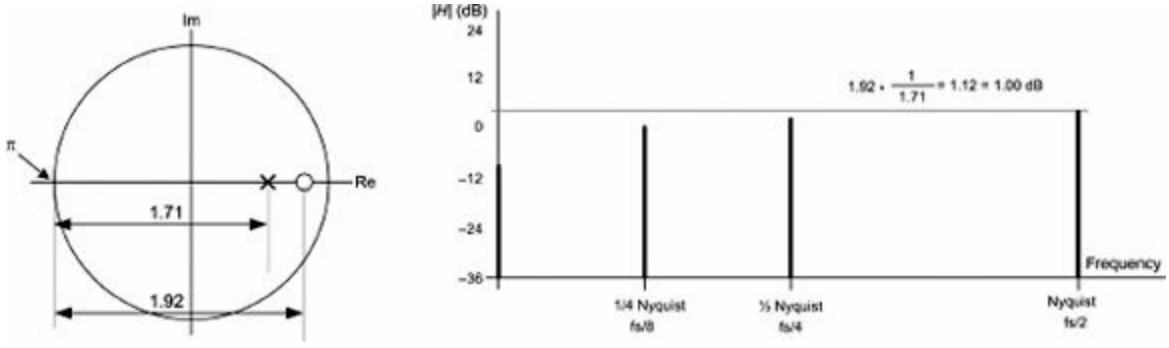


Figure 5.57: At π/4 the pole/zero ratio favors the pole and the response perks up to 1.0 dB; notice that this is the frequency where the pole is clearly dominating, but just barely.

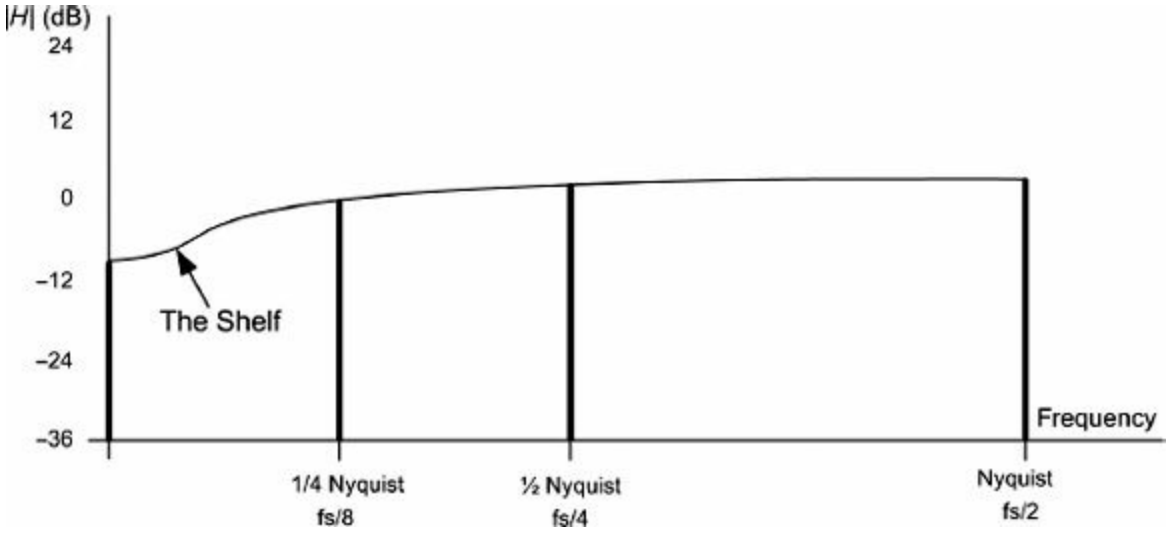


Figure 5.58: The composite frequency response plot shows a −12 dB low shelving filter response, a useful filter in audio.

First, get the transfer function in a form to use for all the evaluation frequencies (Equation 5.83). Then evaluate at our four frequencies.

$$H(z) = \frac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1}} \tag{5.83}$$
$$= \frac{1 - 0.92 z^{-1}}{1 - 0.71 z^{-1}}$$
$$\text{Let } z = e^{j\omega}$$
$$H(\omega) = \frac{1 - 0.92 e^{-j1\omega}}{1 - 0.71 e^{-j1\omega}}$$

Apply Euler's equation:

$$H(\omega) = \frac{1 - 0.92 e^{-j1\omega}}{1 - 0.71 e^{-j1\omega}}$$
$$H(\omega) = \frac{1 - 0.92[\cos(\omega) - j\sin(\omega)]}{1 - 0.71[\cos(\omega) - j\sin(\omega)]}$$

5.19.1 DC (OHz)

$$H(\omega) = \frac{1 - 0.92[\cos(\omega) - j\sin(\omega)]}{1 - 0.71[\cos(\omega) - j\sin(\omega)]}$$
$$= \frac{1 - 0.92[\cos(0) - j\sin(0)]}{1 - 0.71[\cos(0) - j\sin(0)]}$$
$$= \frac{1 - 0.92[1 - j0]}{1 - 0.71[1 - j0]} \tag{5.84}$$
$$= \frac{0.08 + j0}{0.29 + j0}$$

$$
\begin{aligned}
|H(\omega)| &= \frac{|0.08 + j0|}{|0.29 + j0|} \\
&= \frac{\sqrt{a^2 + b^2}}{\sqrt{a^2 + b^2}} \\
&= \frac{\sqrt{0.08^2}}{\sqrt{0.29^2}} = 0.276 = -11.2 \text{ dB} \qquad (5.85) \\
\text{Arg}(H) &= \text{Arg(Num)} - \text{Arg(Denom)} \\
&= \tan^{-1}(0/0.08) - \tan^{-1}(0/0.29) \\
&= 0°
\end{aligned}
$$

By now this should be getting very familiar; the only difference in this situation is that we have to evaluate the numerator and denominator, but the method is the same.

15.19.2 Challenge

Finish the rest of the direct evaluation calculations on your own. The answers are in Table 5.5.

Table 5.5: Challenge answers.

Frequency (ω)	$\|H(\omega)\|$	Arg(H)
Nyquist (π)	1.00 dB	0.0°
½ Nyquist (π/2)	0.82 dB	7.23°
¼ Nyquist (π/4)	0.375 dB	16.60°

Thus, once again the direct evaluation backs up the estimation from the z-plane. Because we have a feedback path, extracting the impulse response will be tedious but we can use RackAFX's pole/zero filter module to analyze the impulse response. Figure 5.59 shows the measured impulse response, while Figure 5.60 shows the frequency and phase responses.

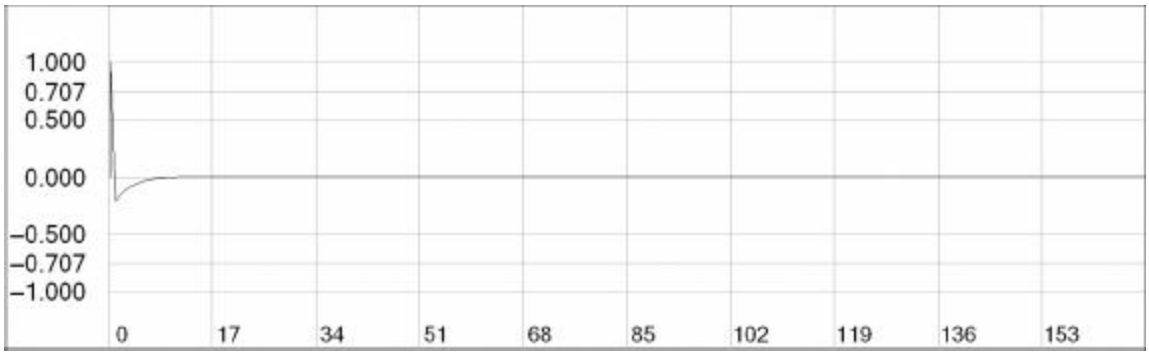


Figure 5.59: Impulse response of the first-order shelving filter.

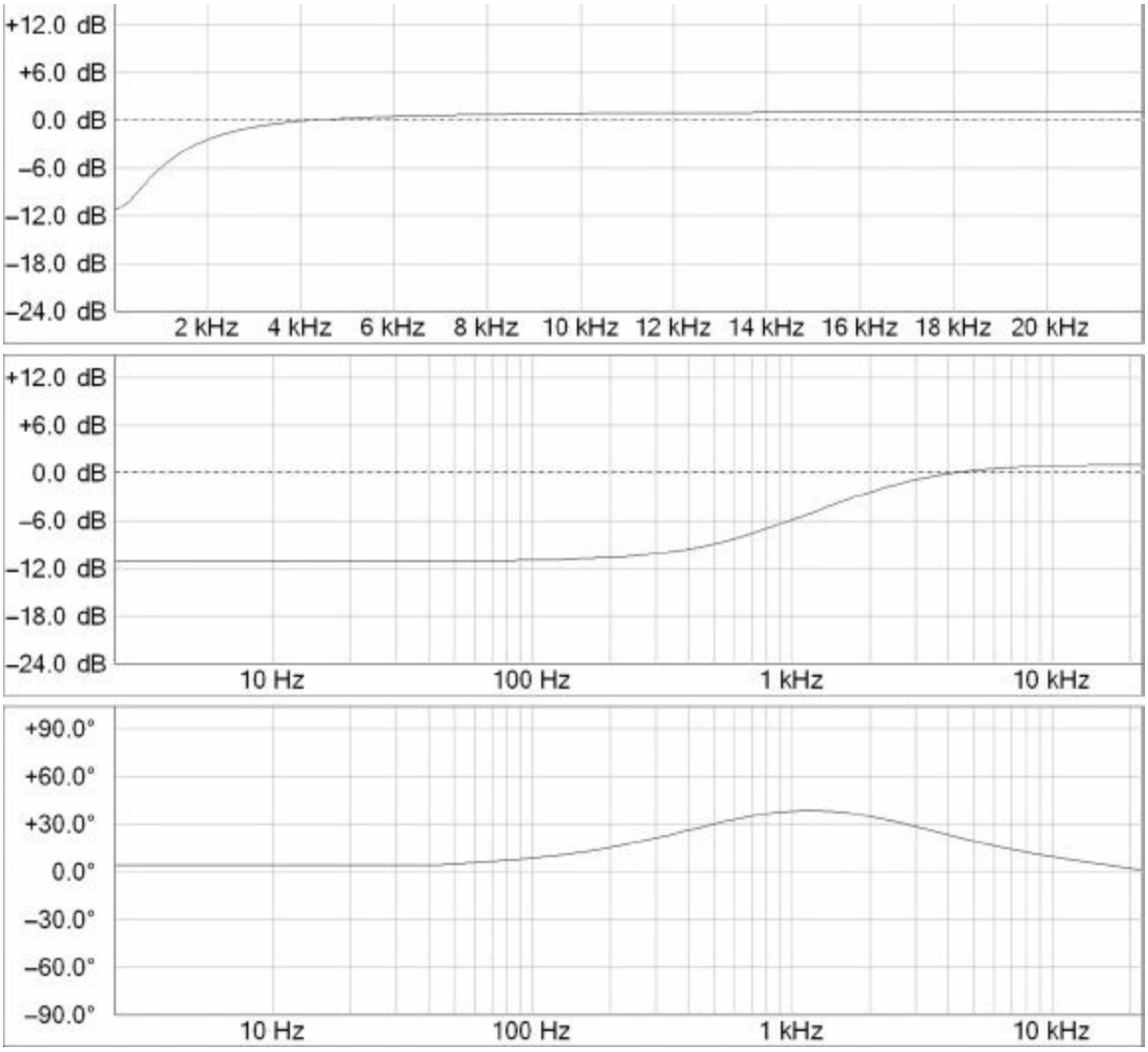


Figure 5.60: Frequency and phase responses of the first-order shelving filter.

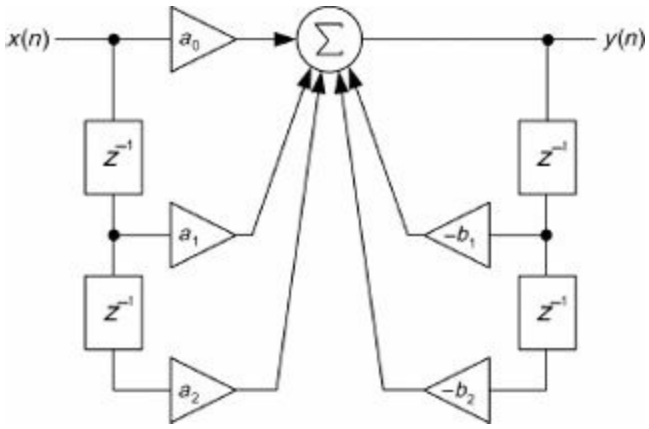


Figure 5.61: The bi-quad.

A first-order shelving filter is a pole-zero design. The shelf will be located in the region between where the zero dominates and where the pole dominates. When neither really dominates, the response is approximately unity gain. The RackAFX frequency and phase plots are shown in Figure 5.60; the log frequency plot has been included to reveal the textbook shelving filter curve.

5.20 The Bi-Quadratic Filter

The last filter topology to study is the bi-quadratic (or bi-quad) filter. The bi-quad consists of two second-order components: a second-order feed-forward and a second-order feed-back filter combined together as shown in the block diagram in Figure 5.61. The resulting transfer function will have two quadratic equations, thus the name.

The difference equation is as follows:

$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2) - b_1y(n-1) - b_2y(n-2) \tag{5.86}$$

Steps 1 to 3: Take the z transform of the difference equation to get the transfer function, then factor out a_0 as the scalar gain coefficient

$$\begin{aligned} y(n) &= a_0x(n) + a_1x(n-1) + a_2x(n-2) - b_1y(n-1) - b_2y(n-2) \\ Y(z) &= a_0X(z) + a_1X(z)z^{-1} + a_2X(z)z^{-2} - b_1Y(z)z^{-1} - b_2Y(z)z^{-2} \end{aligned}$$

Separate variables:

$$\begin{aligned} Y(z) + b_1Y(z)z^{-1} + b_2Y(z)z^{-2} &= a_0X(z) + a_1X(z)z^{-1} + a_2X(z)z^{-2} \\ Y(z)[1 + b_1z^{-1} + b_2z^{-2}] &= X(z)[a_0 + a_1z^{-1} + a_2z^{-2}] \end{aligned}$$

Form transfer function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1z^{-1} + a_2z^{-2}}{1 + b_1z^{-1} + b_2z^{-2}}$$

Factor out a_0

$$H(z) = a_0\frac{1 + \alpha_1z^{-1} + \alpha_2z^{-2}}{1 + b_1z^{-1} + b_2z^{-2}} \tag{5.87}$$

where

$$\begin{aligned} \alpha_1 &= \frac{a_1}{a_0} \\ \alpha_2 &= \frac{a_2}{a_0} \end{aligned}$$

Step 4: Plot the poles and zeros of the transfer function

The bi-quad will produce a conjugate pair of zeros and conjugate pair of poles from the numerator and denominator respectively. Calculating these locations is the same as in the pure second-order feed forward and feed back topologies. All you need to do is plot them in the same unit circle. The transfer function becomes (by simple substitution from previous sections):

$$H(z) = a_0\frac{1 - 2R_z\cos(\theta)z^{-1} + R_z^2z^{-2}}{1 - 2R_p\cos(\phi)z^{-1} + R_p^2z^{-2}} \tag{5.88}$$

Figure 5.62 shows a pair of poles and a pair of zeros plotted together. Each has its own radius, R_z and

R_p, and angle, θ and φ. The same kind of pole/zero tug of war goes on with the bi-quad, only now there are more competing entities.

Estimating the frequency response is complicated by the additional poles and zeros, but the rules are still the same:

- Locate each evaluation frequency on the outer rim of the unit circle.
- Draw a line from the point on the circle to *each* zero and measure the length of these vectors. Do it for each evaluation frequency.
- Draw a line from the point on the circle to *each* pole and measure the length of these vectors. Do it for each evaluation frequency.
- Multiply all the zero magnitudes together.
- Multiply all the *inverse* pole magnitudes together.
- Divide the zero magnitude by the pole magnitude for the final result at that frequency.

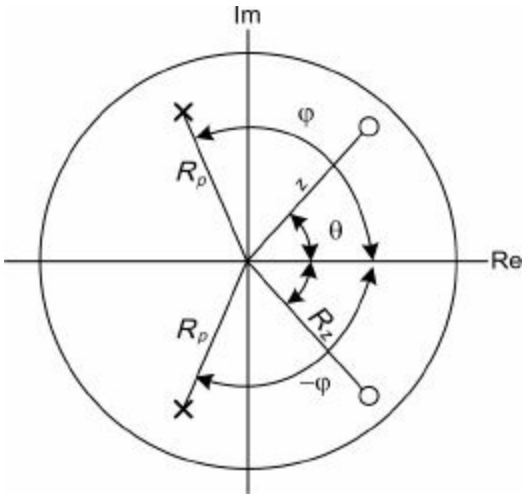


Figure 5.62: Second-order poles and zeros.

Using the following coefficients, $a_0 = 1.0$, $a_1 = 0.73$, $a_2 = 1.00$, $b_1 = -0.78$, $b_2 = 0.88$, we can directly find the pole and zero locations from Equation 5.88 (note that because a_0 is 1.0, you don't have to calculate the α terms). The pairs of poles and zeros are plotted in Figure 5.63.

Zeros are calculated as follows:

$$a_2 = R_z^2 = 1.00$$
$$R_z = \sqrt{1.00} = 1.00$$

and

$$\begin{aligned} a_1 &= -2R\cos(\theta) = 0.73 \\ 2(1.00)\cos(\theta) &= 0.73 \\ \cos(\theta) &= 0.365 \\ \theta &= \arccos(0.365) \\ \theta &= 68.6^\circ \end{aligned} \tag{5.89}$$

Poles are calculated as follows:

$$b_2 = R_p^2 = 0.88$$

$$R_p = \sqrt{0.88} = 0.94$$

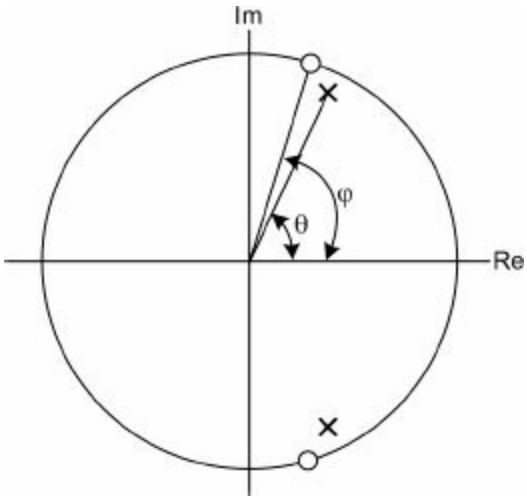


Figure 5.63: Pole/zero plot for the example bi-quad filter.

and

$$\begin{aligned} b_1 &= -2R\cos(\phi) = -0.78 \\ 2\,(0.94)\cos(\phi) &= 0.78 \\ \cos(\phi) &= \frac{0.78}{2(0.94)} \\ \phi &= \arccos(0.414) \\ \phi &= 65.5^\circ \end{aligned} \tag{5.90}$$

The poles and zeros are in close proximity to each other. The zero is directly on the unit circle (R_z = 1.0), so we expect a notch to occur there. The pole is near the unit circle but not touching it, so we expect a resonance there.

We are not going to go through the full response estimation or direct evaluation since it's just repetitive algebra at this point. But, we can use RackAFX's bi-quad module to set up the filter and evaluate it (Figure 5.64). The frequency response clearly shows the resonant peaking due to the pole, then the sudden notch due to the nearby zero.

In Figure 5.64, it's easy to locate the places where the pole or zero dominates. In the low frequencies, the pole dominates and at high frequencies the zero dominates. This is an example of a direct *z*-plane design where we place a pole and zero pair directly in the *z*-plane, then calculate the coefficients. In the next chapter, we will examine some basic Infinite Impulse Response (IIR) design techniques and the direct *z*-plane method will be the first.

Final Review Notes

In this chapter you learned the basics of DSP theory; specifically, you learned the sequence:

1. Take the *z* transform of the difference equation.

2. Fashion the difference equation into a transfer function.
3. Factor out a_0 as the scalar gain coefficient.
4. Estimate the frequency response.
5. Direct evaluation of frequency response.
6. z transform of impulse response as a final check.

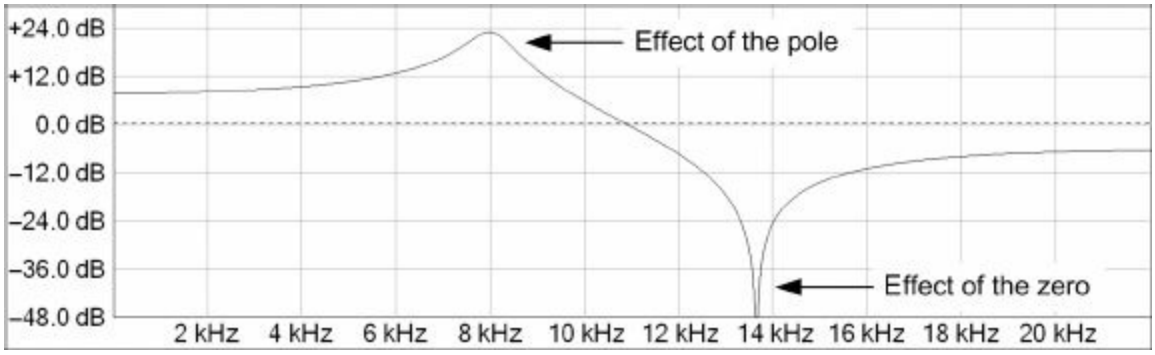


Figure 5.64: The plot from RackAFX shows the resonant peak and notch formed by the pole and zero.

For geometric estimation, the frequency response and phase responses of a filter can be found with Equations 5.91 and 5.92.

$$|H(e^{j\omega})|_\omega = a_0 \frac{\prod_{i=1}^{N} U_i}{\prod_{i=1}^{N} V_i} \tag{5.91}$$

where

N = the filter order

U_i = the geometric length from the point(ω)on the unit circle to the ith zero

V_i = the geometric length from the point(ω)on the unit circle to the ith pole

$$\text{Arg}(H(e^{j\omega}))|_\omega = \sum_{i=1}^{N} \theta_i - \sum_{i=1}^{N} \phi_i \tag{5.92}$$

where

N = the filter order

θ_i = the angle between the ith zero and the vector U_i

ϕ_i = the angle between the ith pole and the vector V_i

For direct evaluation, you simply plug in various values for frequency and crank through the algebra. We applied this same routine to feed-forward, feed-back, and combination types of algorithms, and

then used RackAFX to check our results. We also classified the filters into IIR and Finite Impulse Response (FIR) types.

IIR Filters

- Any filter with a feed-back path is IIR in nature, even if it has feed-forward branches as well.
- The feed-back paths in IIR filters produce poles in the *z*-plane and the poles cause gain to occur in the magnitude response.
- An IIR filter can blow up when its output steadily increases toward infinity, which occurs when the poles are located outside the unit circle.
- If the IIR filter also has feed-forward branches it will produce zeros as well as poles.
- IIR filters can have fast transient responses but may ring.

FIR Filters

- The FIR filter only has feed-forward paths.
- It only produces zeros in the *z*-plane.
- The FIR filter is unconditionally stable since its output can only go to zero in the worst case.
- FIR filters will have slower transient responses because of the time smearing they do on the impulse response.
- The more delay elements in the FIR, the poorer the transient response becomes.

Bibliography

Ifeachor, E. C. and Jervis, B. W. 1993. *Digital Signal Processing: A Practical Approach*, Chapter 3. Menlo Park, CA: Addison-Wesley.

Kwakernaak, H. and Sivan, R. 1991. *Modern Signals and Systems*, Chapter 3. Englewood Cliffs, NJ: Prentice-Hall.

Moore, R. 1990. *Elements of Computer Music*, Chapter 2. Englewood Cliffs, NJ: Prentice-Hall.

Oppenheim, A. V. and Schafer, R. W. 1999. *Discrete-Time Signal Processing* (2nd ed.), Chapter 3. Englewood Cliffs, NJ: Prentice-Hall.

Orfanidis, S. 1996. *Introduction to Signal Processing*, Chapters 10–11. Englewood Cliffs, NJ: Prentice-Hall.

Steiglitz, K. 1996. *A DSP Primer with Applications to Digital Audio and Computer Music*, Chapters 4–5. Menlo Park, CA: Addison-Wesley.

Audio Filter Designs: IIR Filters

It's time to put the theory into practice and make some audio filters and equalizers (EQs). You know that the coefficients of a filter determine its frequency response and other characteristics. But how do you find the coefficients? There are two fundamental ways to find the coefficients of the infinite impulse response (IIR) filter:

- Direct *z*-plane design
- Analog filter to digital filter conversion

This chapter uses the following filter naming conventions:

- LPF: Low-pass filter
- HPF: High-pass filter
- BPF: Band-pass filter
- BSF: Band-stop filter

6.1 Direct z-Plane Design

In this first category of design techniques, you manipulate the poles and zeros directly in the *z*-plane to create the response you want. You take advantage of the simple equations that relate the coefficients to the pole/zero locations. Consider the bi-quad. Equation 6.1 shows the numerator. Equation 6.2 shows the denominator.

$$\begin{aligned} H(z) &= a_0(1 + \alpha_1 z^{-1} + \alpha_2 z^{-2}) \\ &= a_0(1 - 2R\cos(\theta)z^{-1} + R^2 z^{-2}) \end{aligned} \tag{6.1}$$

then

$$\begin{aligned} \alpha_1 &= -2R\cos(\theta) \\ \alpha_2 &= R^2 \end{aligned}$$

$$\begin{aligned} H(z) &= a_0\left[\frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}}\right] \\ &= a_0\left[\frac{1}{(1 - 2R\cos(\theta)z^{-1} + R^2 z^{-2})}\right] \end{aligned} \tag{6.2}$$

then

$$\begin{aligned} b_1 &= -2R\cos(\theta) \\ b_2 &= R^2 \end{aligned}$$

For the numerator or denominator, the a_1 or b_1 coefficients are in direct control over the angles of the zeros or poles. The distance R to the zeros or poles is determined by both a_1, a_2 or b_1, b_2. For first-order filters, the coefficients only control the location of the pole and zero on the real axis. There are

no conjugate pairs. However, careful placement of the pole and zero can still result in useful audio filters.

6.2 Single Pole Filters

A block diagram of a single pole filter is shown in Figure 6.1.

The difference equation is as follows:

$$y(n) = a_0 x(n) - b_1 y(n-1) \tag{6.3}$$

6.2.1 First-Order LPF and HPF

Specify:

- f_c, the corner frequency

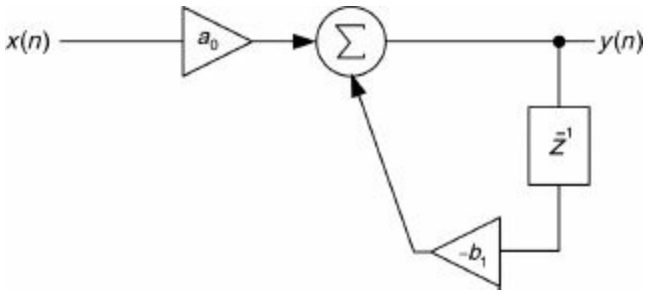


Figure 6.1: The first-order feed-back filter and difference equation.

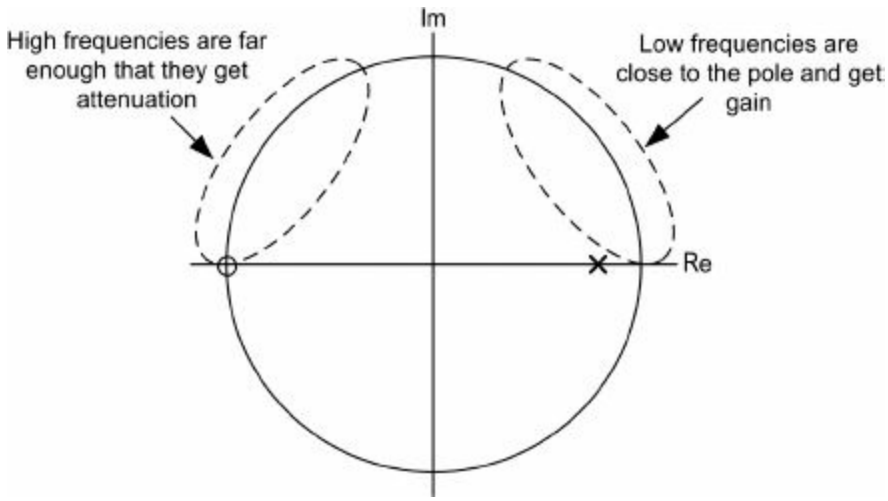


Figure 6.2: The first-order LPF has a single pole and zero on the real axis.

The design equations are as follows:

LPF	HPF
$\theta_c = 2\pi f_c / f_s$	$\theta_c = 2\pi f_c / f_s$
$\gamma = 2 - \cos(\theta_c)$	$\gamma = 2 + \cos(\theta_c)$
$b_1 = \sqrt{\gamma^2 - 1} - \gamma$	$b_1 = \gamma - \sqrt{\gamma^2 - 1}$
$a_0 = 1 + b_1$	$a_0 = 1 - b_1$

(6.4)

These simple first-order filters work by moving the pole back and forth across the real axis while holding a fixed zero at Nyquist. When the pole is on the right side of the unit circle, low frequencies are gained up due to their close proximity to the pole. High frequencies are close to the zero and get attenuated (Figure 6.2). This produces the low-pass response in Figure 6.3.

When the pole/zero pair are reversed, the opposite effect happens and the high frequencies are now boosted with low frequencies attenuated. There is also a zero at 0 Hz. This produces the high-pass response. These are simple but very useful designs and are found in many effects including delay and reverb algorithms.

6.3 Resonators

A resonator is a band-pass filter that can be made to have a very narrow peak. The simple version uses a second-order feed-back topology.

6.3.1 Simple Resonator

A block diagram of a simple resonator is shown in Figure 6.4.

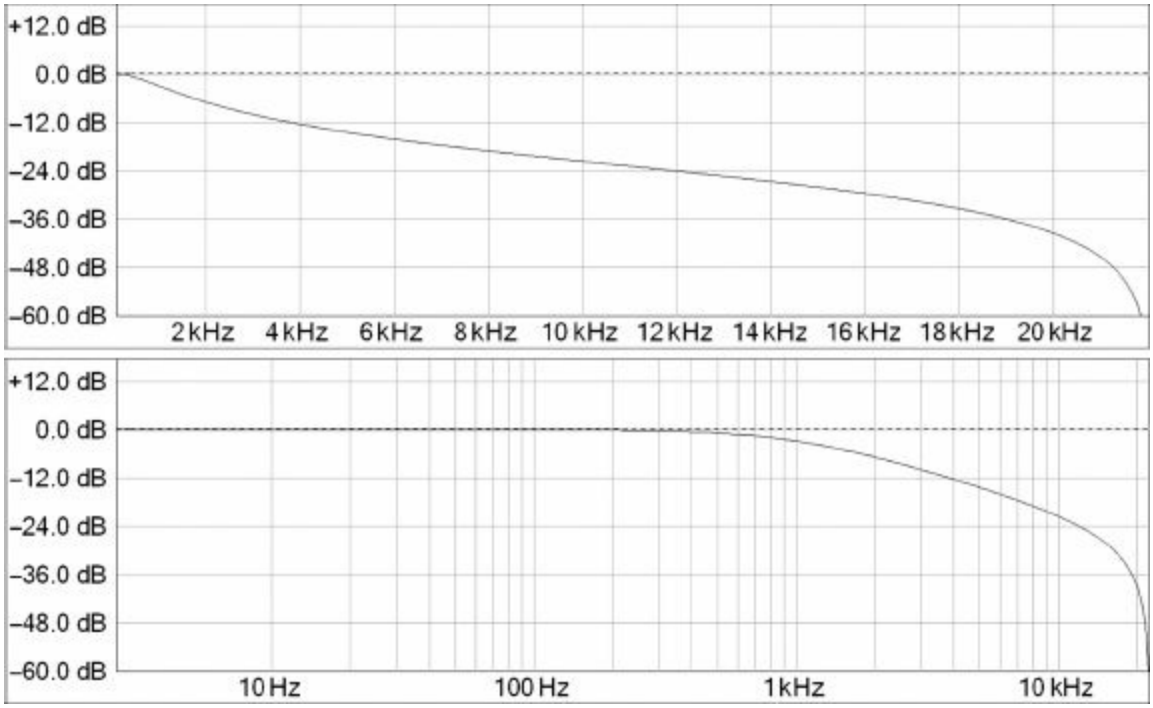


Figure 6.3: Linear and log frequency responses of the one-pole low-pass filter with f_c = 1 kHz.

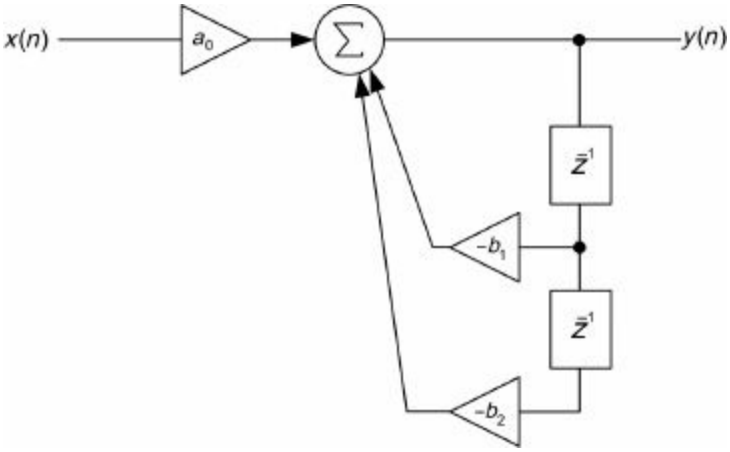


Figure 6.4: Second-order feed-back filter block diagram.

The difference equation is as follows:

$$y(n) = a_0x(n) - b_1y(n-1) - b_2y(n-2) \tag{6.5}$$

Specify:

- f_c, center frequency
- BW, 3 dB bandwidth; or Q, quality factor

The design equations are as follows:

$$\begin{aligned} \theta_c &= 2\pi f_c/f_s \\ W &= f_c/Q \\ b_2 &= e^{\left(-2\pi\frac{BW}{f_s}\right)} \\ b_1 &= \frac{-4b_2}{1+b_2}\cos(\theta_c) \\ a_0 &= (1-b_2)\sqrt{1-\frac{b_1^2}{4b_2}} \end{aligned} \tag{6.6}$$

The resonator works by simple manipulation of the conjugate poles formed with the second-order feed-back network. The b_2 term controls the distance out to the pole which makes the resonant peak sharper (when the pole is close to the unit circle) or wider (when the pole is farther away from the unit circle). The b_1 and b_2 terms control the angle of the pole, which controls the center frequency, shown in Figure 6.5. The a_0 term is used to scale the filter so its peak output is always normalized to unity gain or 0 dB.

The drawback to this design is that it is only symmetrical in frequency response at one frequency, π/2, when the low-frequency and high-frequency magnitude vectors are symmetrical. At all other frequencies, the response is shifted asymmetrically. When the pole is in the first quadrant, it is nearer to the low frequencies than high frequencies and so the low end is boosted (Figure 6.6). The opposite happens (Figure 6.7) when the pole moves to the second quadrant and the high frequencies get the boost. The solution to the asymmetry problem is a really useful filter, especially for making filter banks of extremely narrow band-pass filters.

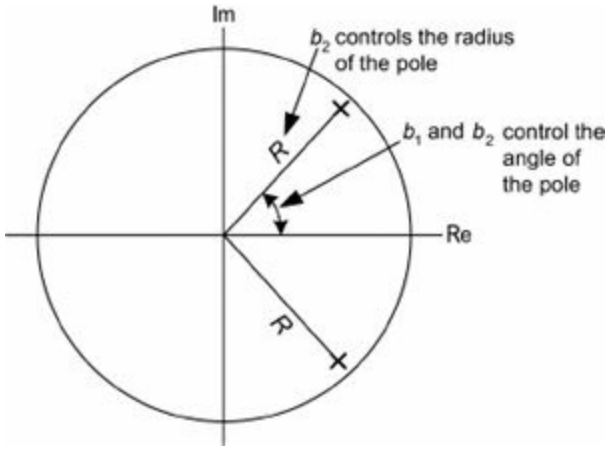


Figure 6.5: The location of the resonator's conjugate poles are determined by the coefficients.

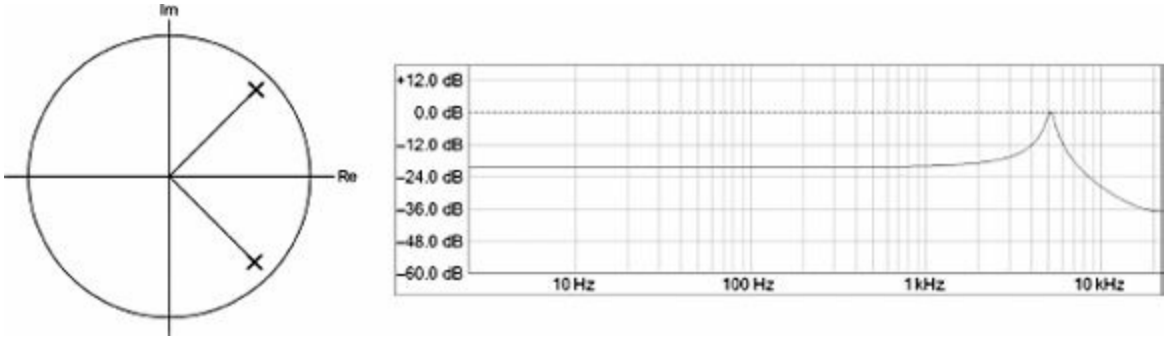


Figure 6.6: The asymmetrical response shape when the pole is in the first quadrant; notice the difference in gain at DC versus Nyquist. This filter has f_s = 44.1 kHz, $f_c = \pi/4$ = 5.5125 kHz, and Q = 10.

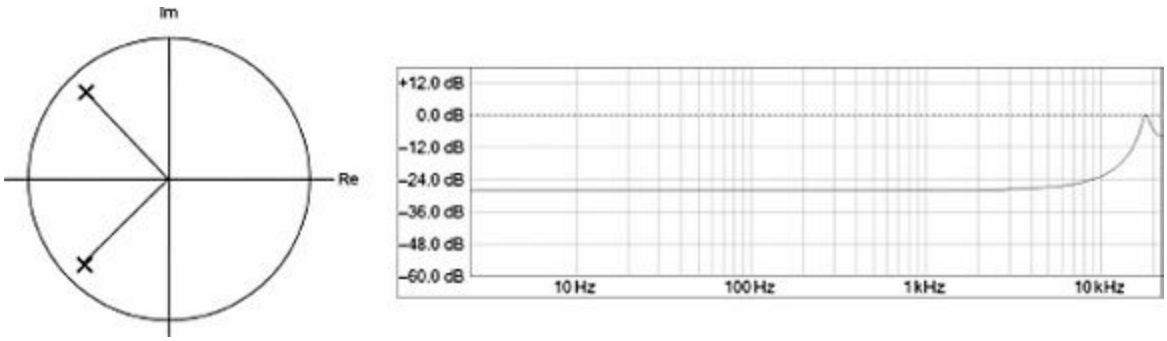


Figure 6.7: The asymmetrical response shape when the pole is in the second quadrant; notice the difference in gain at DC versus Nyquist. This filter has *fs* = 44.1 kHz, $f_c = 3\ \pi/4$ =16.5375 kHz, and Q = 10.

6.3.2 Smith-Angell Improved Resonator

A block diagram of a Smith-Angell improved resonator is shown in Figure 6.8.

The difference equation is as follows:

$$y(n) = a_0x(n) + a_2x(n-2) - b_1y(n-1) - b_2y(n-2) \qquad (6.7)$$

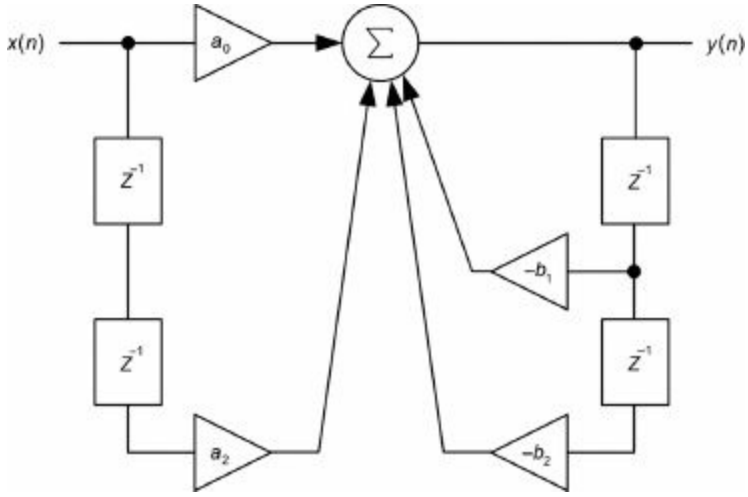


Figure 6.8: Block diagram for the Smith-Angell improved resonator.

Specify:

- f_c, center frequency
- BW, 3 dB bandwidth; or Q, quality factor

The design equations are as follows:

$$\begin{aligned} \theta_c &= 2\pi f_c / f_s \\ BW &= f_c / Q \\ b_2 &= e^{\left(-2\pi \frac{BW}{f_s}\right)} \\ b_1 &= \frac{-4b_2}{1 + b_2} \cos(\theta_c) \\ a_0 &= 1 - \sqrt{b_2} \\ a_2 &= -a_0 \end{aligned} \tag{6.8}$$

This design is also gain normalized with a_0 and as before, the radius is set with b_2 first, then b_1 is calculated using the b_2 coefficient and the desired pole frequency. The filter is not truly normalized to 0.0 dB; there is actually a slight fluctuation of less than 1 dB.

The improved resonator is also named the Smith-Angell resonator, after its inventors. They improved the design by dropping two zeros into the filter, one at $z = 1$ and one at $z = -1$ in order to pin down DC and Nyquist with a response of zero. This forces the filter to become somewhat more symmetric (but not entirely) and has the advantage of making the band pass even more selective in nature (Figure 6.9).

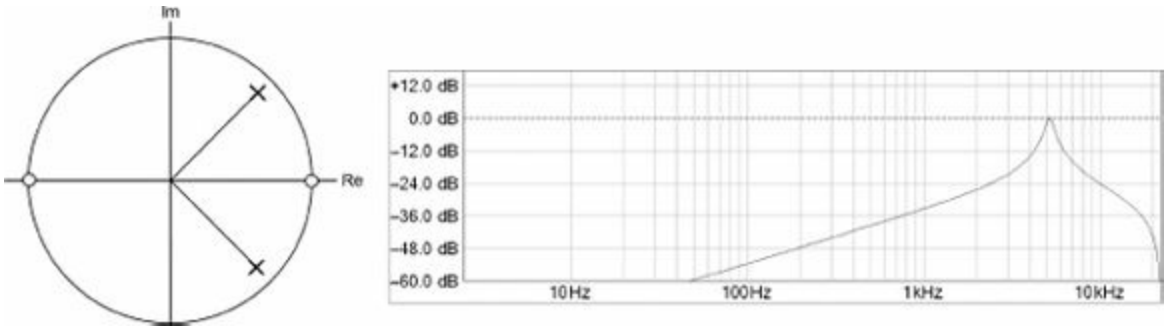


Figure 6.9: The effect on the resonator shape with the added zeros to hold down DC and Nyquist. This filter has f_s = 44.1 kHz, $f_c = \pi/4$ = 5.5125 kHz, and Q = 20 (notice how close the poles are to the unit circle).

6.4 Analog Filter to Digital Filter Conversion

A more widely used approach to filter design is to first start with the classic analog designs and then convert them into digital versions. There's good reason to do this because there are many excellent analog designs already done for us. We just need a way to make them work in our digital world. While analog filter design is outside the scope of the book, there are many similarities between the two design worlds. For example, both analog and digital filter design involve a transfer function that you manipulate to produce poles and zeros. They also both use a transform to get from the time domain to the frequency domain. A fundamental difference is that in analog, there is no Nyquist restriction and all frequencies from −∞ to +∞ are included. Also, in analog, reactive components or circuits like inductors, capacitors, and gyrators (a circuit that simulates an inductor) are used to create the phase shift that is at the heart of the filtering operation. Table 6.1 summarizes the similarities and differences.

Table 6.1: The differences between analog and digital filter design technique.

Digital	Analog
• Uses a transfer function to relate I/O	• Uses a transfer function to relate I/O
• Delay elements create phase shift	• Reactive components create phase shift
• Uses the z transform (sampled time to frequency)	• Uses the Laplace transform (continuous time to frequency)
• Poles and zeros in the z-plane	• Poles and zeros in the s-plane
• Nyquist limitation	• All frequencies from −∞ to +∞ allowed
• Poles must be inside the unit circle for stable operation	• Poles must be in the left-hand part of the s-plane for stable operation

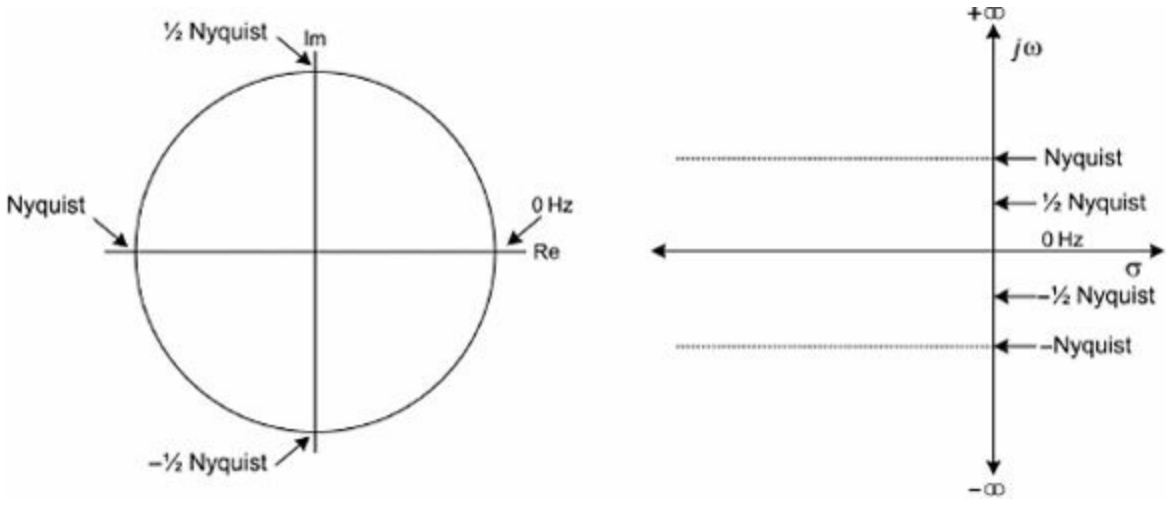


Figure 6.10: Comparison of the unit circle in the *z*-plane with the *s*-plane.

In Figure 6.10 you can see the *s*-plane on the right—it is also a complex plane. The real axis is named the σ axis and the imaginary axis is the *j*ω axis. The *j*ω axis is the frequency axis and it spans −∞ to +∞ rad/sec. The unit circle maps to the portion on the *j*ω axis between −Nyquist and +Nyquist. In order to transform an analog filter design into a digital filter design, we need a mapping system to get the poles and zeros from the *s*-plane to become poles and zeros in the *z*-plane. Once we have that, the coefficients that produce those poles and zeros in the digital locations can be calculated. In the analog world, poles that are close to the *j*ω axis will result in poles that are close to the unit circle in the digital world. The problem is that poles and zeros can exist anywhere along the *j*ω axis, even at frequencies outside the digital Nyquist zone. It is common for analog designs to have a pole or a zero at −∞ and/or +∞ on either the σ and/or *j*ω axis.

In Figure 6.11 you can see the problem: the pair of zeros close to the *j*ω axis at ½ Nyquist will map to locations close to the unit circle at the ½ Nyquist angles. But what about the pair of poles that are outside the Nyquist frequency in the *s*-plane? The other way to think about the issue is that in the analog *s*-plane, the entire left-hand side, including all the infinite frequencies, must map to the *interior* of the unit circle in the *z*-plane. Also, the entire right-hand plane must map to the *exterior* of the unit circle, as shown in Figure 6.12.

So, what we need is a mathematical device to make the transformation from the *s*-plane to the *z*-plane. It must map an infinitely large area into the unit circle. Fortunately, many of these mathematical transforms already exist. The one best suited for us is called the bilinear transform, and we can use it to transform an analog transfer function, *H*(*s*) into a digital one, *H*(*z*).

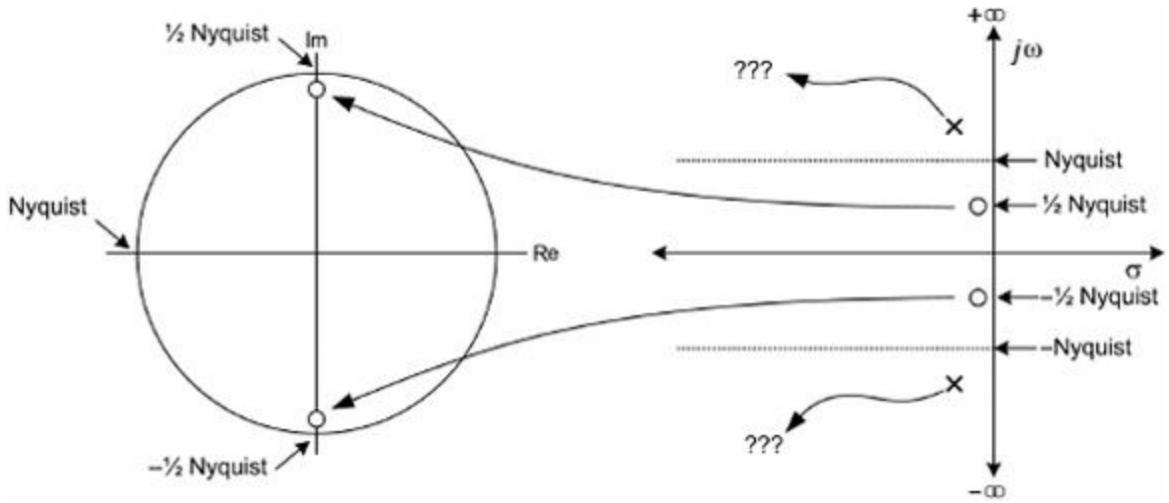


Figure 6.11: Mapping the poles and zeros from the analog *s*-plane to the digital *z*-plane.

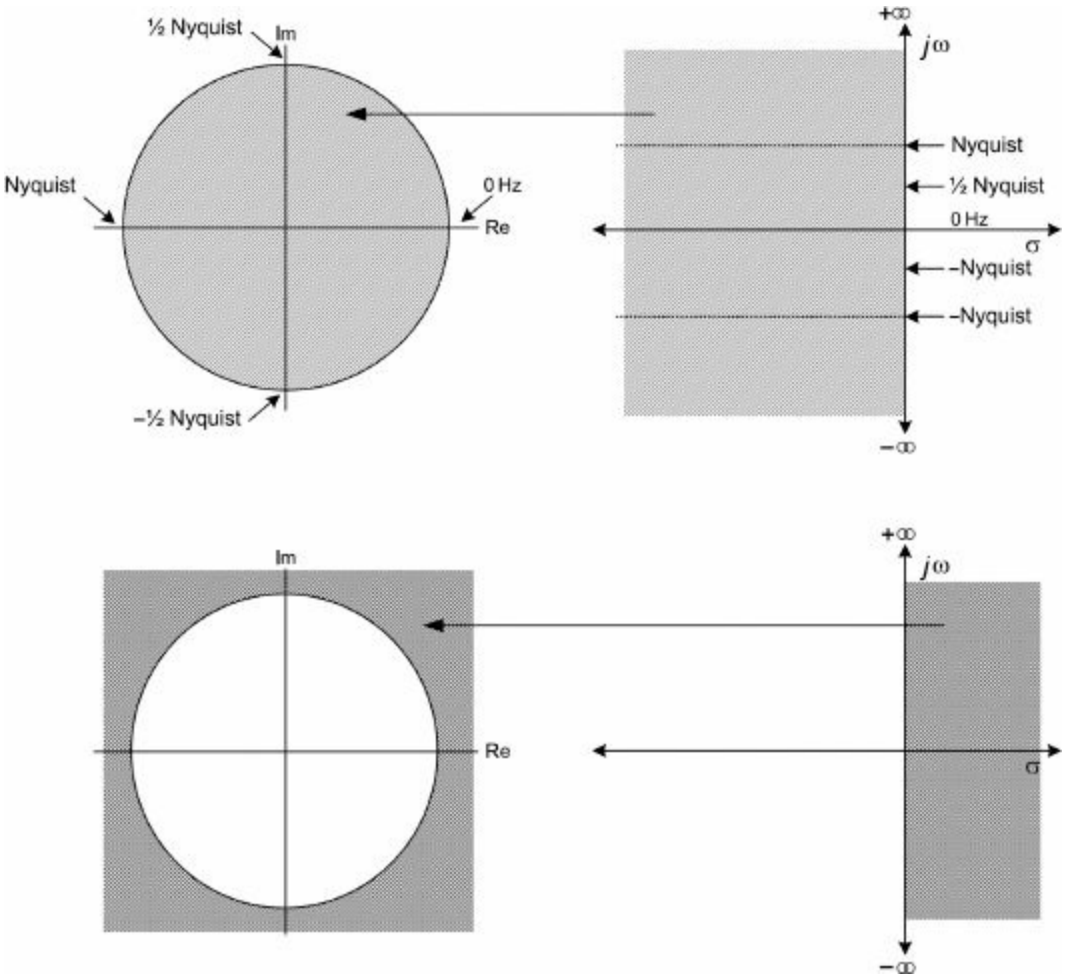


Figure 6.12: Mapping the infinitely large left-hand plane into the finite space inside the unit circle and the right-hand plane into the exterior of the unit circle.

We wish to convert an analog transfer function $H(s)$ into a sampled equivalent $H(z)$. Effectively, this means we need a way to sample the continuous s-plane to produce the sampled z-plane version. In other words, we need to create a sampled analog transfer function $H_s(s)$ where the subscript s stands for "sampled." We seek to find a function $g(z)$ such that Equation 6.9 holds true.

$$H(s) \xrightarrow{s=g(z)} H(z) \tag{6.9}$$

Since the sample interval is T, then the term $e^{j\omega T}$ would correspond to one sample in time. So, if we evaluate $H(z)$ by letting $z = e^{j\omega T}$ then we will arrive at the sampled analog transfer function $H_s(s)$ in Equation 6.10:

$$H(s) \xrightarrow{s=g(z)} H(z) \xrightarrow{z=e^{j\omega T}} H_s(s) \tag{6.10}$$

To solve, we note that $s = j\omega$ (the analog definition of s) and $z = e^{j\omega T}$ to get Equation 6.11:

$$\begin{aligned} z &= e^{sT} \\ \ln(z) &= \ln(e^{sT}) \\ \ln(z) &= sT \end{aligned} \tag{6.11}$$

or

$$sT = \ln(z)$$

We now have the relationship between s and z, but we need to deal with taking the natural log of z, which doesn't have a closed form solution. If we use the Taylor series expansion for ln(), we get Equation 6.12 for Re(z) > 0:

$$sT = 2\left[\frac{z-1}{z+1} + \frac{1}{3}\left(\frac{z-1}{z+1}\right)^3 - \frac{1}{5}\left(\frac{z-1}{z+1}\right)^5 + \frac{1}{7}\left(\frac{z-1}{z+1}\right)^7 - \ldots\right]$$

Taking the first term only

$$s = \frac{2}{T}\frac{z-1}{z+1} \tag{6.12}$$

This first term *approximation* of the general solution is the bilinear transform. The bilinear transform we use is Equation 6.13:

$$s = \frac{z-1}{z+1} \tag{6.13}$$

The $2/T$ term washes out mathematically, so we can neglect it. This equation does the mapping by taking values at the infinite frequencies and mapping them to Nyquist. So, a pole or zero at $-\infty$ or $+\infty$ is mapped to Nyquist. The other frequencies between Nyquist and $-\infty$ or ∞ Hz are squeezed into the little region right around Nyquist, just inside the unit circle, as shown in Figure 6.13.

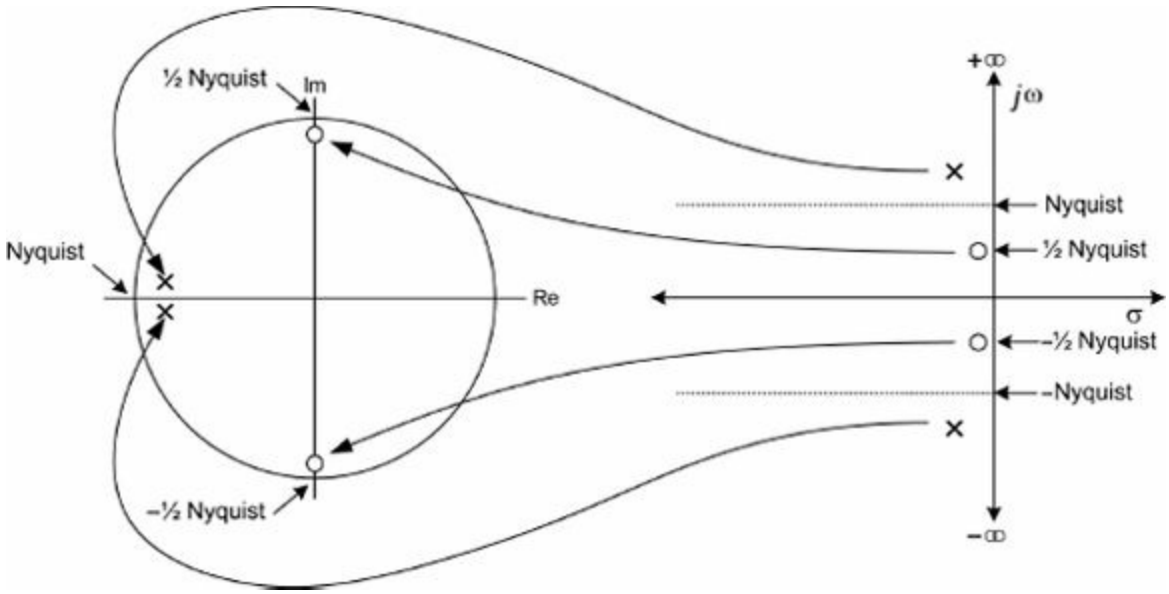


Figure 6.13: The bilinear transform maps the area outsize the Nyquist zone on the left-hand plane to an area just inside the circle near Nyquist.

The bilinear transform maps analog frequencies to their corresponding digital frequencies nonlinearly via the tan() function (Equation 6.14):

$$\omega_a = \tan\left[\frac{\omega_d/f_s}{2}\right]$$

where

ω_a = the analog frequency
ω_d = the mapped digital frequency
f_s = the sample rate

(6.14)

The tan() function is linear at low values of ω but becomes more nonlinear as the frequency increases. At low frequencies, the analog and digital frequencies map closely. At high frequencies, the digital frequencies become warped and do not map properly to the analog counterparts. This means that a given analog design with a cutoff frequency f_c might have the wrong digital cutoff frequency after the conversion.

The solution is to pre-warp the *analog* filter so that its cutoff frequency is in the wrong location in the analog domain, but will wind up in the correct location in the digital domain. To pre-warp the analog filter, you just use the same equation (Equation 6.14) applied to the cutoff frequency of the analog filter. When you combine all the operations you get the bilinear *Z* transform, or BZT. The step-by-step method of conversion is as follows (Ifeachor 1993):

1. Start with an analog filter with a normalized transfer function $H(s)$—"normalized" means the cutoff frequency is set to $\omega = 1$ rad/sec; this is the typical way analog transfer functions are described and specified.
2. Choose the cutoff frequency of the digital filter ω_d; get the pre-warped analog corner frequency using Equation 6.15:

$$\omega_a = \tan\left[\frac{\omega_d/f_s}{2}\right] \tag{6.15}$$

3. Scale the normalized filter's cutoff frequency out to the new analog cutoff ω_a by replacing s with s/ω_a in the analog transfer function.
4. Apply the bilinear transform by replacing s with Equation 6.13: $z - 1$

$$s = \frac{z-1}{z+1} \tag{6.13}$$

5. Manipulate the transfer function $H(z)$ algebraically to get it into the familiar form so you can identify the coefficients (format the numerator and denominator to match the transfer functions you studied in the last chapter); this is often the most difficult part.

NOTE: This only works for LPF designs but, fortunately, there are easy conversions. The difference is in the step where you scale the analog filter's cutoff frequency just before applying the bilinear transform. How it works is described next.

For LPF and HPF:

- Specify ω_d, the desired digital cutoff frequency.
- Calculate the pre-warped analog cutoff with Equation 6.14:

$$\omega_a = \tan\left[\frac{\omega_d/f_s}{2}\right]$$

where

$$\begin{aligned} \omega_a &= \text{the analog frequency} \\ \omega_d &= \text{the mapped digital frequency} \\ f_s &= \text{the sample rate} \end{aligned} \tag{6.14}$$

For BPF and BSF:

- Specify ω_{dL} and ω_{dH}, the lower and upper corner frequencies of the digital filter.
- Calculate the two analog corner frequencies with Equation 6.16:

$$\begin{aligned} \omega_{aL} &= \tan\left[\frac{\omega_{dL}/f_s}{2}\right] \\ \omega_{aH} &= \tan\left[\frac{\omega_{dH}/f_s}{2}\right] \\ \omega_0^2 &= \omega_{aL}\omega_{aH} \\ W &= \omega_{aH} - \omega_{aL} \end{aligned} \tag{6.16}$$

Next, scale the filter with Equation 6.17:

Filter Type	Scaling Factor
LPF	$s = \frac{s}{\omega_a}$
HPF	$s = \frac{\omega_a}{s}$
BPF	$s = \frac{s^2 + \omega_0^2}{Ws}$
BSF	$s = \frac{Ws}{s^2 + \omega_0^2}$

(6.17)

Example:

Convert the basic resistor-capacitor (RC) analog LPF in Figure 6.14 into a digital LPF. The sample rate is 44.1 kHz and the desired cutoff frequency is 1 kHz.

Step 1: Get the normalized $H(s)$:

$$\begin{aligned} H(j\omega) &= \frac{1}{j\omega RC + 1} \\ \text{Let } s &= j\omega \\ \text{Normalize by setting } RC &= 1 \\ H(s) &= \frac{1}{s+1} \end{aligned} \tag{6.18}$$

Step 2: Calculate the pre-warped analog cutoff:

$$\begin{aligned} f_c &= 1 \text{ kHz} \\ \omega_d &= 2\pi f_c = 6283.2 \text{ rad/sec} \end{aligned} \tag{6.19}$$

$$\omega_a = \tan\left(\frac{\omega_d T}{2}\right) = \tan\left[\frac{(6283.2)(1/44100)}{2}\right] = 0.07136 \tag{6.20}$$

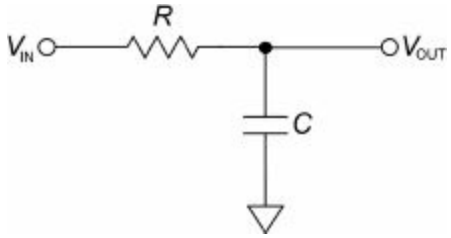


Figure 6.14: A simple analog RC low-pass filter.

Step 3: De-normalize the analog transfer function $H(s)$ with the appropriate factor:

$$\begin{aligned} H(s) &= \frac{1}{s+1} \\ s &= \frac{s}{\omega_a} \text{ because this is an LPF} \\ H'(s) &= \frac{1}{s/\omega_a + 1} = \frac{1}{s/0.07136 + 1} \\ &= \frac{0.07136}{s + 0.07136} \end{aligned} \tag{6.21}$$

Step 4: Apply the BZT:

$$\begin{aligned} H(z) &= H'(s)\big|_{s=(z-1)/(z+1)} \\ &= \frac{0.07136}{\frac{z-1}{z+1} + 0.07136} \\ &= \frac{0.07136(z+1)}{z - 1 + 0.07136(z+1)} \\ &= \frac{0.07136z + 0.07136}{z - 1 + 0.07136z + 0.07136} \\ &= \frac{0.07136z + 0.07136}{z + 0.07136z - 0.9286} \end{aligned} \tag{6.22}$$

Get into standard $H(z)$ format:

$$\begin{aligned} &= \frac{0.07136 + 0.07136z^{-1}}{1 + 0.07136 - 0.9286z^{-1}} \\ H(z) &= \frac{0.0667 + 0.0667z^{-1}}{1 - 0.8667z^{-1}} = \frac{a_0 + a_1z^{-1}}{1 + b_1z^{-1}} \end{aligned}$$

Equation 6.22 is in the format that we need with the numerator and denominator properly formed to observe the coefficients. From the transfer function, you can see that:

- $a_0 = 0.0667$
- $a_1 = 0.0667$
- $b_1 = -0.8667$

The difference equation is Equation 6.23:

$$y(n) = 0.0667x(n) + 0.0667x(n-1) + 0.8667y(n-1) \quad (6.23)$$

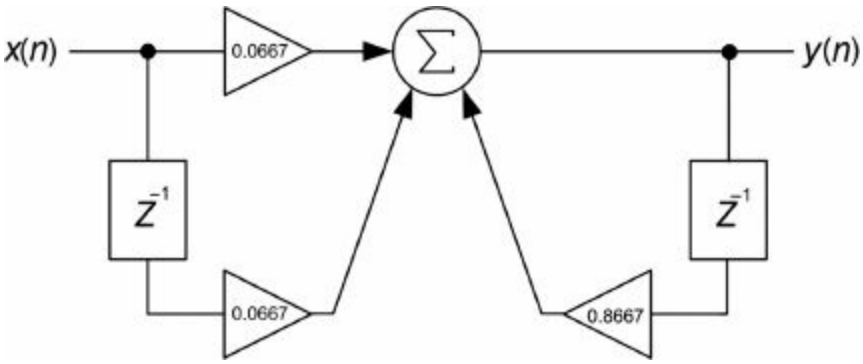


Figure 6.15: The digital equivalent of the analog RC low-pass filter.

The block diagram of the completed design reveals a pole-zero filter. This makes sense—the original analog filter had a pole at $s = -1$ and a zero at infinity. The digital equivalent has a pole at $z = 0.8667$ and a zero at $z = -1$ (Nyquist) both on the real axis (Figure 6.15).

6.4.1 Challenge

The analog transfer function of a second-order LPF is Equation 6.24:

$$H(s) = \frac{1}{s^2 + (1/Q)s + 1} \quad (6.24)$$

The analog LPF has the following specifications: $Q = 1, f_c = 1$ kHz, $f_s = 44.1$ kHz. Apply the bilinear transform and some algebra to find the coefficients. (Answer: $a_0 = 0.0047$, $a_1 = 0.0095$, $a_2 = 0.0047$, $b_1 = -1.8485$, $b_2 = 0.8673$.)

6.5 Effect of Poles or Zeros at Infinity

In the analog transfer function (Equation 6.19) of the previous example, you can see that there is an analog pole at $s = -1$ since that would make the transfer function go to infinity, and there is a zero at $s = \infty$ because that will cause $H(s)$ to become 0.0. There is also a zero at $s = -\infty$. Interestingly, these two infinity values are in the same location because the reality is that the sand $j\omega$ axis actually wrap around an infinitely large sphere and touch each other at $\pm\infty$. So, in this first-order case engineers only show the single zero at infinity and they choose to use the one at $-\infty$ so this transfer function's pole and zero would be plotted like Figure 6.16 in the s-plane. For low-pass zeros at infinity, the bilinear transform maps the zero at infinity to $z = -1$ (Nyquist) (Figure 6.17).

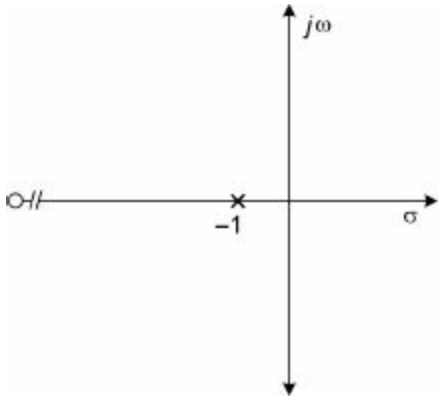


Figure 6.16: The pole at $s = 21$ and the zero at $s = \pm\infty$ plotted in the s-plane.

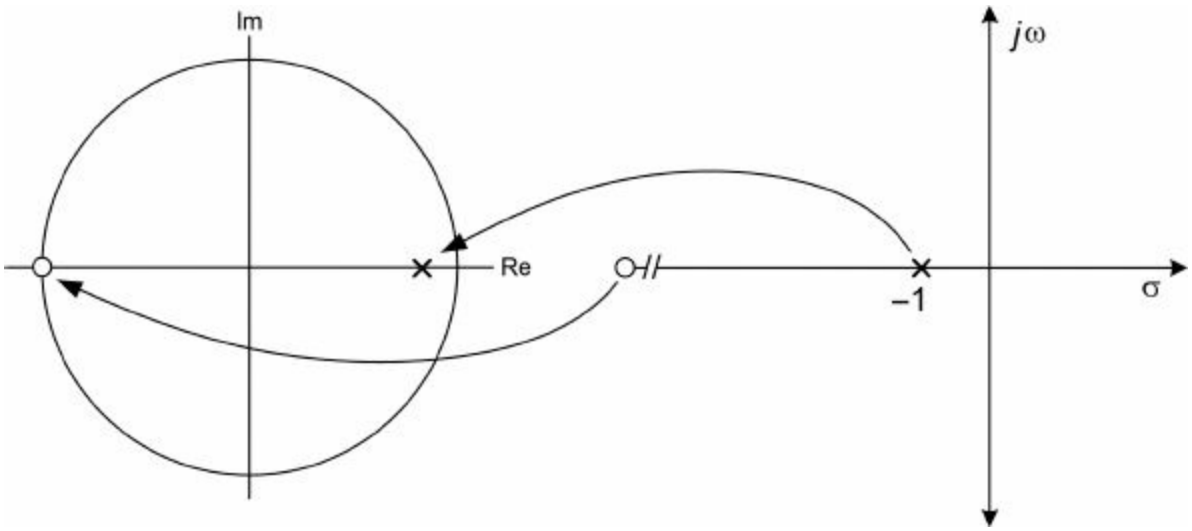


Figure 6.17: The bilinear transform maps real zeros at infinity to the Nyquist frequency in the z-plane.

Next consider the second-order analog low-pass filter transfer function:

$$H(s) = \frac{1}{s^2 + (1/Q)s + 1} \tag{6.25}$$

which factors as

$$H(s) = \frac{1}{(s - P)(s - P^*)}$$

This transfer function has a pair of conjugate poles at locations P and P^* or $(a + bj\omega)$ and $(a - bj\omega)$ as well as a pair of zeros at $\pm\infty$. The bilinear transform maps the poles on the left side of the s-plane to locations inside the unit circle. Once again, it maps the zeros at $\pm\infty$ to $z = -1$ or the Nyquist frequency (Figures 6.18 and 6.19).

There are several methods of overcoming this problem (Orfanidis 1997; Massberg 2011). The Massberg approach is discussed further in Section 6.9. Even with the error in the BZT, it is still an excellent tool for converting existing designs. You will notice that in the design methods, we start with the desired cutoff frequency (and Q, where applicable) and calculate the coefficients.

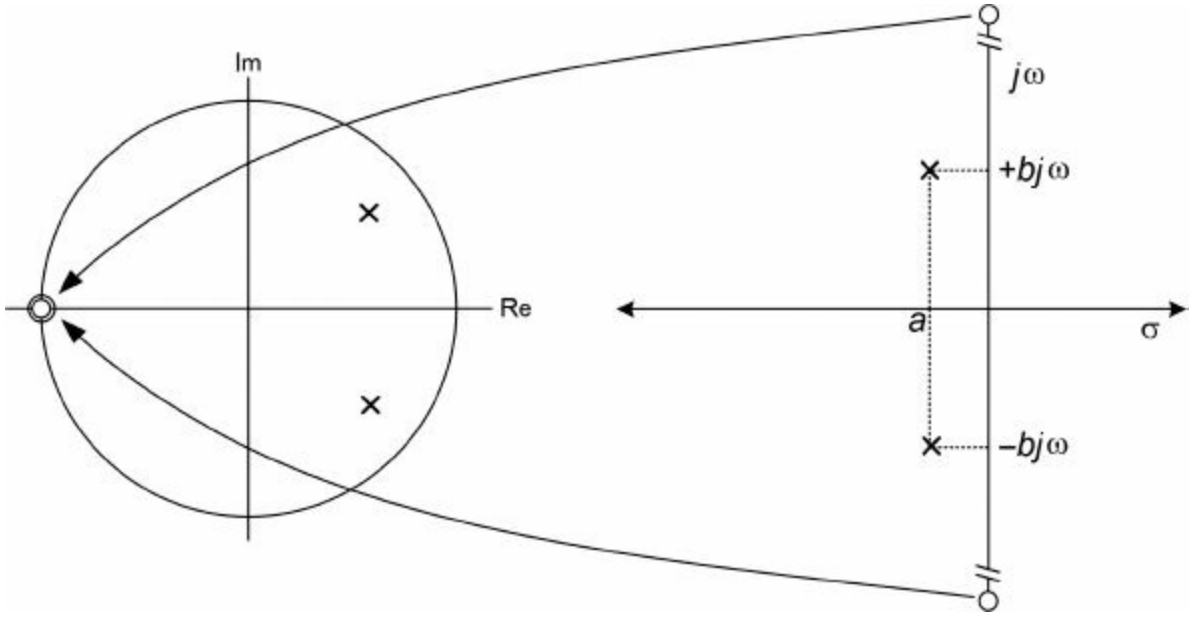


Figure 6.18: The bilinear transform maps imaginary zeros at infinity to Nyquist in the *z*-plane.

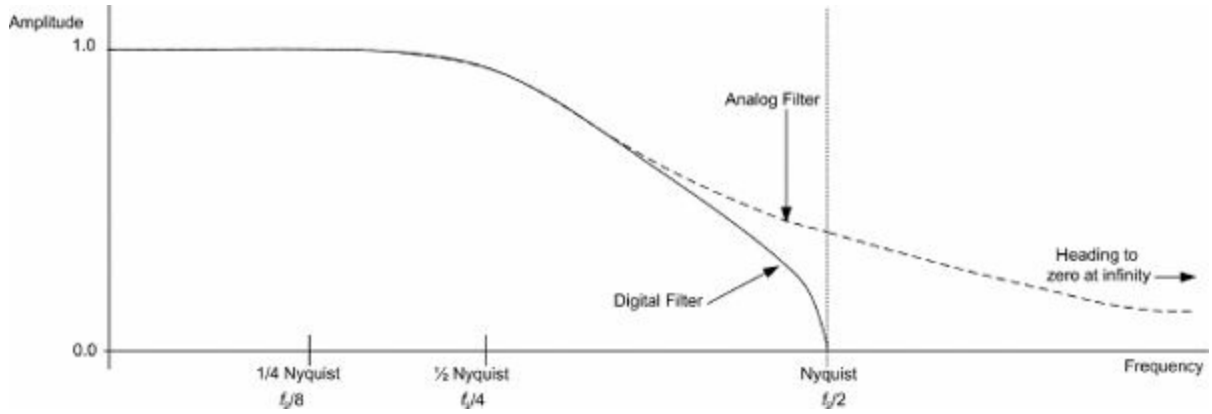


Figure 6.19: The zero at infinity causes an error in the upper part of the frequency. The error is worst at the Nyquist frequency; the analog error is exaggerated for the purposes of illustration.

6.6 Generic Bi-Quad Designs

The following classical analog filters are converted to digital and implemented as bi-quad topologies:

- LPF (low-pass filter)
- HPF (high-pass filter)
- BPF (band-pass filter)
- BSF (band-stop filter)
- Second-order Butterworth LPF and HPF
- Second-order Linkwitz–Riley LPF and HPF (good for crossovers)
- First- and second-order all-pass filters (APF)

Low-pass and high-pass filters are characterized by their corner frequency f_c and (for second-order and higher) their Q or resonant peaking value. A Q of 0.707 is the highest value Q can assume before peaking occurs. It is called the Butterworth or maximally flat response. With a Q of 0.707 the −3 dB

point of the filter will be exactly at f_c. For these designs, once the Q rises above 0.707, it will correspond to the peak frequency and not the −3 dB frequency. Equations 6.26 through 6.29 relate the Q, peak frequency, −3 dB frequency, and the peak magnitude values.

$$f_{-3\text{dB}} = f_c\sqrt{\left(\frac{1}{2Q^2} - 1\right) + \sqrt{\left(\frac{1}{2Q^2} - 1\right)^2 + 1}} \tag{6.26}$$

$$f_c = \frac{f_{-3\text{dB}}}{\sqrt{\left(\frac{1}{2Q^2} - 1\right) + \sqrt{\left(\frac{1}{2Q^2} - 1\right)^2 + 1}}} \tag{6.27}$$

$$\text{Peak gain} = \frac{Q^2}{\sqrt{Q^2 - 0.25}} (Q > 0.707 \text{ only}) \tag{6.28}$$

$$\text{Peak}_{\text{dB}} = 20\log(\text{peak gain}) \tag{6.29}$$

Band-pass, band-stop, graphic, and parametric EQs are specified by their center frequency f_c and Q or bandwidth (Equation 6.30). Depending on the filter, the arithmetic (Equation 6.31) or geometric (Equation 6.32) mean is used to relate f_c and the band edges f_U and f_L.

$$Q = \frac{f_c}{BW} \tag{6.30}$$

$$\text{Arithmetic mean} = f_c = \frac{f_U - f_L}{2} \tag{6.31}$$

$$\text{Geometric mean} = f_c = \sqrt{f_U f_L} \tag{6.32}$$

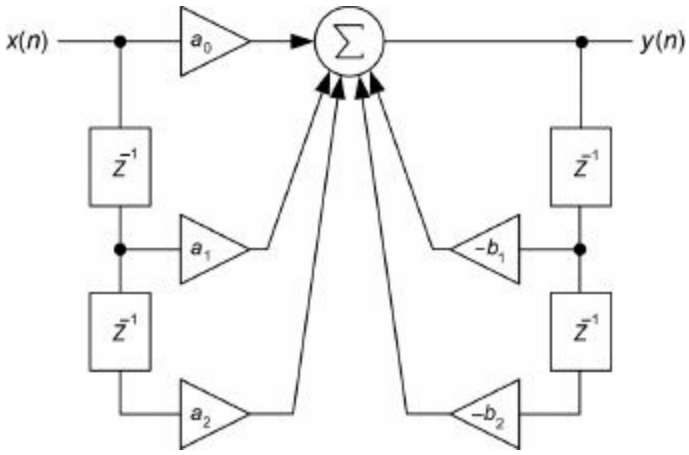


Figure 6.20: Generic bi-quad structure.

The block diagram is shown in Figure 6.20.

The difference equation is as follows:

$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2) - b_1y(n-1) - b_2y(n-2) \tag{6.33}$$

6.6.1 First-Order LPF and HPF

Specify:

- f_c, corner frequency; see Figure 6.21 for examples

The design equations are as follows:

$$\begin{array}{ll} \text{LPF} & \text{HPF} \\ \theta_c = 2\pi f_c / f_s & \theta_c = 2\pi f_c / f_s \\ \gamma = \dfrac{\cos\theta_c}{1 + \sin\theta_c} & \gamma = \dfrac{\cos\theta_c}{1 + \sin\theta_c} \\ a_0 = \dfrac{1 - \gamma}{2} & a_0 = \dfrac{1 + \gamma}{2} \\ a_1 = \dfrac{1 - \gamma}{2} & a_1 = -\left(\dfrac{1 + \gamma}{2}\right) \\ a_2 = 0.0 & a_2 = 0.0 \\ b_1 = -\gamma & b_1 = -\gamma \\ b_2 = 0.0 & b_2 = 0.0 \end{array} \tag{6.34}$$

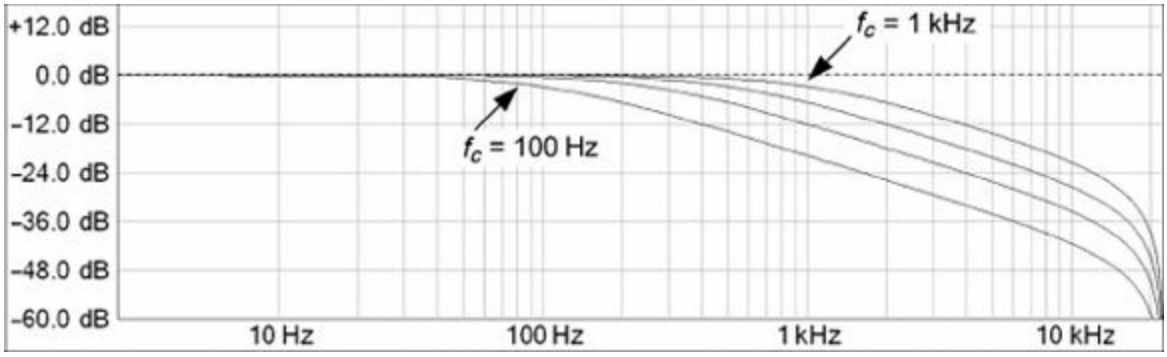


Figure 6.21: First-order LPF with f_c = 100 Hz, 250 Hz, 500 Hz, and 1 kHz.

6.6.2 Second-Order LPF and HPF

Specify:

- f_c, corner frequency
- Q, quality factor controlling resonant peaking; Figure 6.22 shows the effect of Q and peaking

The design equations are as follows:

$$
\begin{array}{ll}
\text{LPF} & \text{HPF} \\
\theta_c = 2\pi f_c / f_s & \theta_c = 2\pi f_c / f_s \\
d = \dfrac{1}{Q} & d = \dfrac{1}{Q} \\
\beta = 0.5\dfrac{1 - \dfrac{d}{2}\sin\theta_c}{1 + \dfrac{d}{2}\sin\theta_c} & \beta = 0.5\dfrac{1 - \dfrac{d}{2}\sin\theta_c}{1 + \dfrac{d}{2}\sin\theta_c} \\
\gamma = (0.5 + \beta)\cos\theta_c & \gamma = (0.5 + \beta)\cos\theta_c \\
a_0 = \dfrac{0.5 + \beta - \gamma}{2.0} & a_0 = \dfrac{0.5 + \beta + \gamma}{2.0} \\
a_1 = 0.5 + \beta - \gamma & a_1 = -(0.5 + \beta + \gamma) \\
a_2 = \dfrac{0.5 + \beta - \gamma}{2.0} & a_2 = \dfrac{0.5 + \beta + \gamma}{2.0} \\
b_1 = -2\gamma & b_1 = -2\gamma \\
b_2 = 2\beta & b_2 = 2\beta
\end{array}
\tag{6.35}
$$

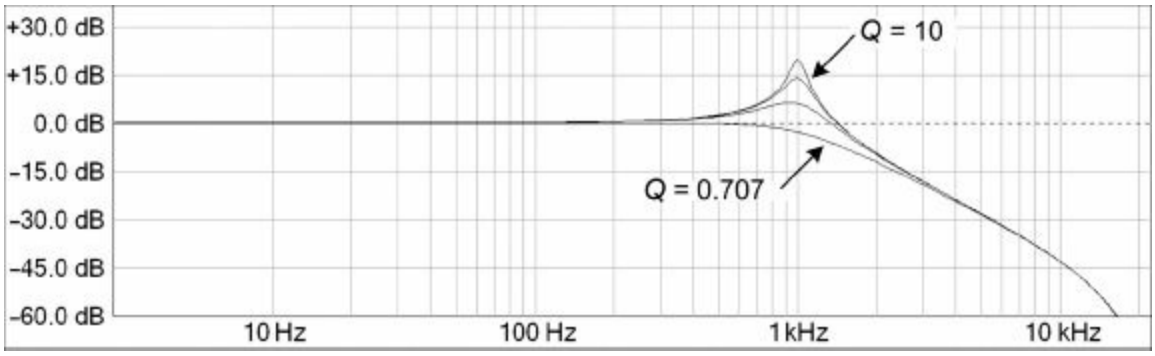


Figure 6.22: Second-order LPF responses: f_c = 1 kHz, Q = 0.707, 2, 5, 10. Notice that as the Q rises above 0.707, f_c becomes the peak frequency.

6.6.3 Second-Order BPF and BSF

Specify:

- f_c, corner frequency
- Q, quality factor controlling width of peak or notch $51/BW$; Figure 6.23 shows the BSF version

Note: These filter coefficients contain the tan() function, which is undefined at $\pi/2$ and then flips negative between $\pi/2$ and π. The argument is $\theta_c/2Q$, so care must be taken with these filters to ensure that $\theta_c/2Q$ does not fall on the range of $\pi/2$ to π. One solution is to clamp the value of $\theta_c/2Q$ so that it never equals or exceeds $\pi/2$.

The design equations are as follows:

BPF	BSF
$\theta_c = 2\pi f_c / f_s$	$\theta_c = 2\pi f_c / f_s$
$\beta = 0.5\dfrac{1 - \tan(\theta_c/2Q)}{1 + \tan(\theta_c/2Q)}$	$\beta = 0.5\dfrac{1 - \tan(\theta_c/2Q)}{1 + \tan(\theta_c/2Q)}$
$\gamma = (0.5 + \beta)\cos\theta_c$	$\gamma = (0.5 + \beta)\cos\theta_c$
$a_0 = 0.5 - \beta$	$a_0 = 0.5 + \beta$
$a_1 = 0.0$	$a_1 = -2\gamma$
$a_2 = -(0.5 - \beta)$	$a_2 = 0.5 + \beta$
$b_1 = -2\gamma$	$b_1 = -2\gamma$
$b_2 = 2\beta$	$b_2 = 2\beta$

(6.36)

6.6.4 Second-Order Butterworth LPF and HPF

Specify:

• f_c, corner frequency

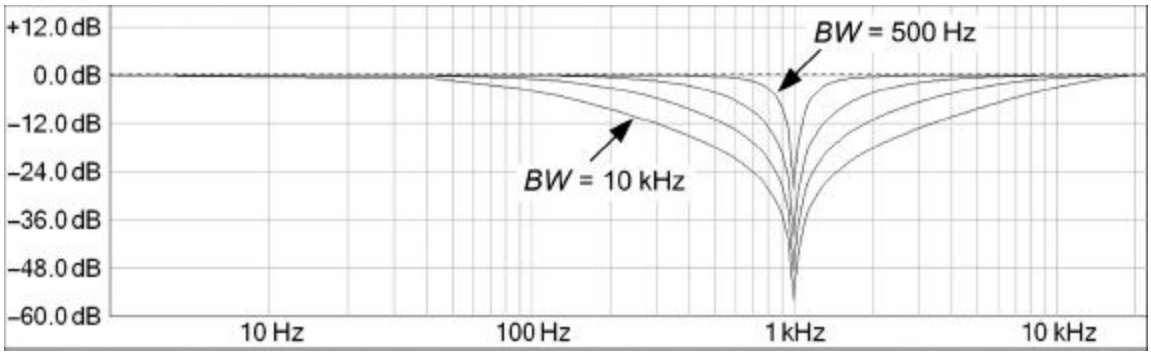


Figure 6.23: Second-order BSF responses: f_c = 1 kHz, bandwidth = 500 Hz (narrowest), 2 kHz, = kHz, and 10 kHz (widest).

Butterworth low-pass and high-pass filters are specialized versions of the ordinary second-order low-pass filter. Their Q values are fixed at 0.707, which is the largest value it can assume before peaking in the frequency response is observed.

The design equations are as follows:

LPF	HPF
$C = \dfrac{1}{\tan(\pi f_c / f_s)}$	$C = \tan(\pi f_c / f_s)$
$a_0 = \dfrac{1}{1 + \sqrt{2}C + C^2}$	$a_0 = \dfrac{1}{1 + \sqrt{2}C + C^2}$
$a_1 = 2a_0$	$a_1 = -2a_0$
$a_2 = a_0$	$a_2 = a_0$
$b_1 = 2a_0(1 - C^2)$	$b_1 = 2a_0(C^2 - 1)$
$b_2 = a_0(1 - \sqrt{2}C + C^2)$	$b_2 = a_0(1 - \sqrt{2}C + C^2)$

(6.37)

6.6.5 Second-Order Butterworth BPF and BSF

Specify:

- f_c, corner frequency
- BW, bandwidth of peak/notch = f_c/Q

Butterworth BPF and BSF are made by cascading (BPF) or paralleling (BSF) a Butterworth LPF and Butterworth HPF.

Note: These filter coefficients contain the tan() function, which is undefined at π/2 and then flips negative between π/2 and p. The argument is $\pi f_c BW/f_s$, so care must be taken with these filters to ensure that $\pi f_c BW/f_s$ does not fall on the range of π/2 to π. One solution is to clamp the value of $\pi f_c BW/f_s$ so that it never equals or exceeds π/2.

The design equations are as follows:

BPF	BSF
$C = \dfrac{1}{\tan(\pi f_c BW/f_s)}$	$C = \tan(\pi f_c BW/f_s)$
$D = 2\cos(2\pi f_c/f_s)$	$D = 2\cos(2\pi f_c/f_s)$
$a_0 = \dfrac{1}{1+C}$	$a_0 = \dfrac{1}{1+C}$
$a_1 = 0.0$	$a_1 = -a_0 D$
$a_2 = -a_0$	$a_2 = a_0$
$b_1 = -a_0(CD)$	$b_1 = -a_0 D$
$b_2 = a_0(C-1)$	$b_2 = a_0(1-C)$

(6.38)

6.6.6 Second-Order Linkwitz–Riley LPF and HPF

Specify:

- f_c, corner frequency (−6 dB)

Second-order Linkwitz–Riley LPFs and HPFs are designed to have an attenuation of −6 dB at the corner frequency rather than the standard −3 dB, shown in Figure 6.24. When these filters are placed in parallel with the same cutoff frequency, their outputs sum exactly and the resulting response is flat. They are often used in crossovers. We use them for the spectral dynamics processor later on.

The design equations are as follows:

LPF	HPF
$\theta_c = \pi f_c / f_s$	$\theta_c = \pi f_c / f_s$
$\Omega_c = \pi f_c$	$\Omega_c = \pi f_c$
$\kappa = \dfrac{\Omega_c}{\tan(\theta_c)}$	$\kappa = \dfrac{\Omega_c}{\tan(\theta_c)}$
$\delta = \kappa^2 + \Omega_c^2 + 2\kappa\Omega_c$	$\delta = \kappa^2 + \Omega_c^2 + 2\kappa\Omega_c$
$a_0 = \dfrac{\Omega_c^2}{\delta}$	$a_0 = \dfrac{\kappa^2}{\delta}$
$a_1 = 2\dfrac{\Omega_c^2}{\delta}$	$a_1 = \dfrac{-2\kappa^2}{\delta}$
$a_2 = \dfrac{\Omega_c^2}{\delta}$	$a_2 = \dfrac{\kappa^2}{\delta}$
$b_1 = \dfrac{-2\kappa^2 + 2\Omega_c^2}{\delta}$	$b_1 = \dfrac{-2\kappa^2 + 2\Omega_c^2}{\delta}$
$b_2 = \dfrac{-2\kappa\Omega_c + \kappa^2 + \Omega_c^2}{\delta}$	$b_2 = \dfrac{-2\kappa\Omega_c + \kappa^2 + \Omega_c^2}{\delta}$

(6.39)

All pass filters have interesting designs that yield equally interesting results. Their frequency responses are perfectly flat from DC to Nyquist. However, the phase responses are the same as those in first- and second-order LPFs. You get all of the phase shift but none of the frequency response change. These filters can be found in crossovers and reverb algorithms. They are also used for the phaser effect. APFs are designed with pole-zero pairs whose pole/zero radii are reciprocals of one another. For a first-order APF, the pole lies somewhere on the real axis inside the unit circle at radius R. The zero lies outside the unit circle at radius $1/R$ (Figure 6.25).

If you think about the geometric interpretation and analysis of the transfer function, as you move around the unit circle, your vector distances from the pole and zero will always be reciprocals, or 1/each other. The amplitude response is then flat but the phase response *does* change because it is not based on the distance to the point in question but rather on the angle of incidence of the ray line drawn from the analysis point to the pole or zero. The second-order APF has complementary poles and zeros also at reciprocal radii (Figure 6.26).

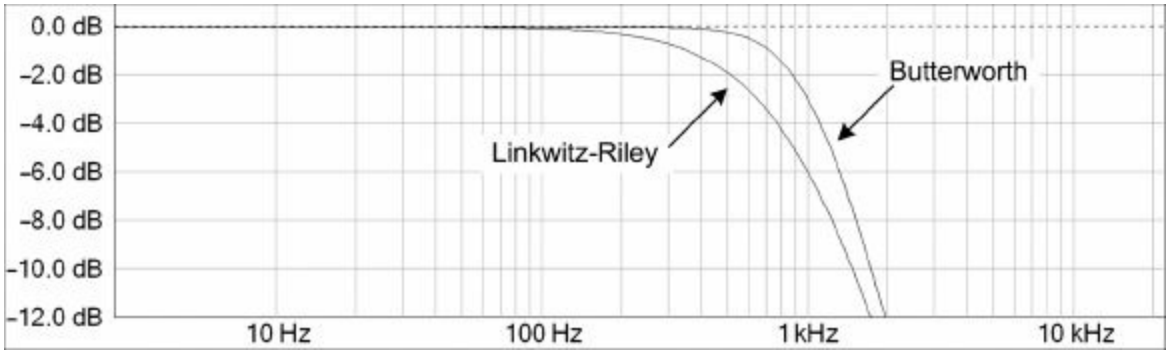


Figure 6.24: A comparison between Butterworth and Linkwitz–Riley filters each with f_c = 1 kHz; the Linkwitz–Riley filter is down −6 dB at f_c.

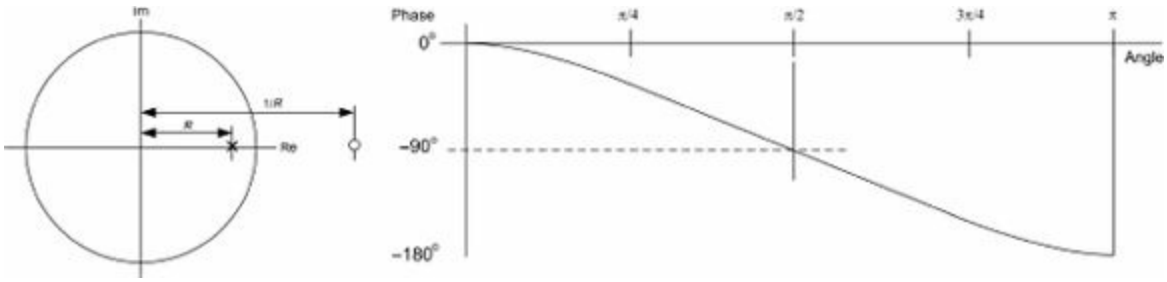


Figure 6.25: The first-order APF has a flat frequency response but shifts the phase of the signal by −90 degrees at f_c (π/2 in this example).

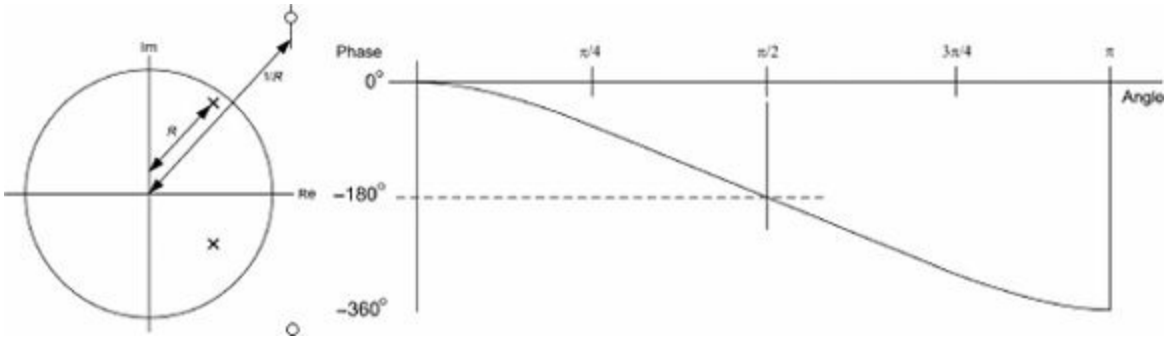


Figure 6.26: The second-order APF adds another −90 degrees of phase shift at f_c (π/2 here).

6.6.7 First- and Second-Order APF

Specify:

- f_c, corner frequency
- Q, steepness of phase shift at f_c (second-order only)

The design equations are as follows:

First-Order APF	Second-Order APF
$\alpha = \dfrac{\tan(\pi f_c/f_s) - 1}{\tan(\pi f_c/f_s) + 1}$	$\alpha = \dfrac{\tan(\pi Q/f_s) - 1}{\tan(\pi Q/f_s) + 1}$
	$\beta = -\cos\theta_c$
$a_0 = \alpha$	$a_0 = -\alpha$
$a_1 = 1.0$	$a_1 = \beta(1 - \alpha)$
$a_2 = 0.0$	$\alpha_2 = 1.0$
$b_1 = \alpha$	$b_1 = \beta(1 - \alpha)$
$b_2 = 0.0$	$b_2 = -\alpha$

(6.40)

6.7 Audio Specific Filters

The basic classical filters provide many functions in audio and can be very musical (e.g., resonant low-pass filters in synthesizers) but you also need filters that are very audio specific. These filters are found all throughout plug-ins and have been used in mixing consoles and other audio gear for decades. These designs are often not found in DSP textbooks because of their specific audio-only functions. These filters include:

- Shelving filters
- Parametric EQ
- Graphic EQ

These all require a slightly modified bi-quad structure. The reason is that these filters require mixing the original, unfiltered input directly with the output in a mix ratio. The ratio is controlled by two more coefficients, c_0 and d_0.

6.7.1 Modified Bi-Quad

You can see that the filter in Figure 6.27 is a bi-quad with two extra coefficients, c_0 and d_0, which make the wet-dry mix ratio needed for these filter types.

6.7.2 First-Order Shelving Filters

Low shelving

Specify:

- f_c, low shelf frequency
- Low-frequency gain/attenuation in dB

High shelving

Specify:

- f_c, high shelf frequency
- High-frequency gain/attenuation in dB

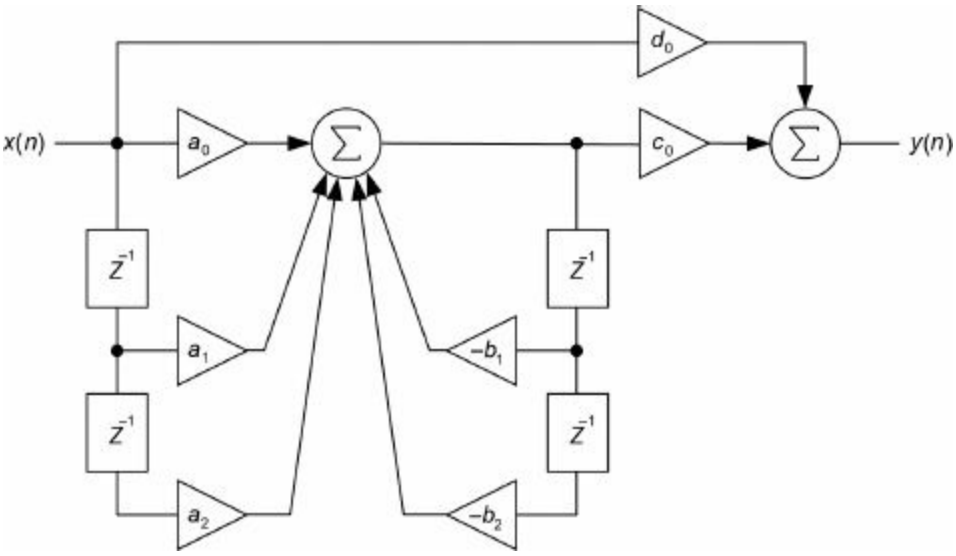


Figure 6.27: The modified bi-quad required for audio filters.

The design equations are as follows:

Low Shelving	High Shelving
$\theta_c = 2\pi f_c / f_s$	$\theta_c = 2\pi f_c / f_s$
$\mu = 10^{Gain(\text{dB})/20}$	$\mu = 10^{Gain(\text{dB})/20}$
$\beta = \frac{4}{1+\mu}$	$\beta = \frac{1+\mu}{4}$
$\delta = \beta \tan(\theta_c/2)$	$\delta = \beta \tan(\theta_c/2)$
$\gamma = \frac{1-\delta}{1+\delta}$	$\gamma = \frac{1-\delta}{1+\delta}$
$a_0 = \frac{1-\gamma}{2}$	$a_0 = \frac{1+\gamma}{2}$
$a_1 = \frac{1-\gamma}{2}$	$a_1 = -\left(\frac{1+\gamma}{2}\right)$
$a_2 = 0.0$	$a_2 = 0.0$
$b_1 = -\gamma$	$b_1 = -\gamma$
$b_2 = 0.0$	$b_2 = 0.0$
$c_0 = \mu - 1.0$	$c_0 = \mu - 1.0$
$d_0 = 1.0$	$d_0 = 1.0$

(6.41)

Shelving filters are used in many tone controls, especially when there are only two, bass and treble, which are almost always implemented as shelf types. The filters have a corner frequency and gain or attenuation value. Figure 6.28 shows a family of shelving filter response curves.

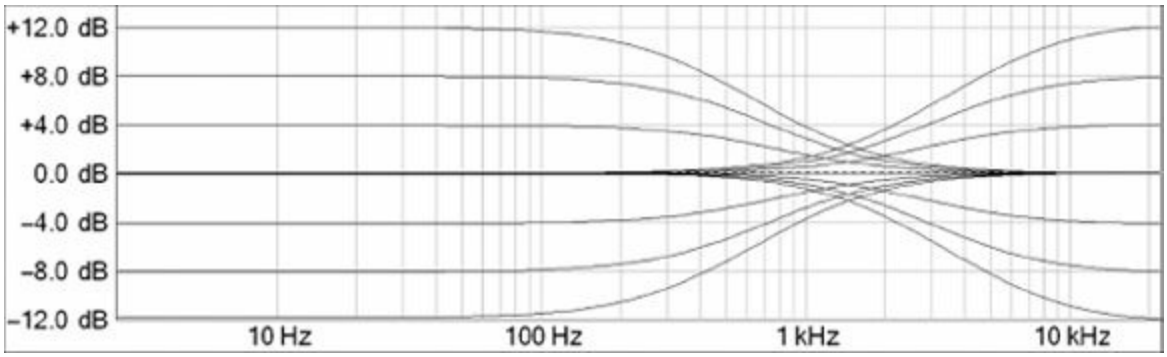


Figure 6.28: Low and high shelving filter responses. The low shelf frequency = 400 Hz, high shelf frequency = 5 kHz, with a variety of boost/cut settings.

6.7.3 Second-Order Parametric/Peaking Filter: Non-Constant-Q

Specify:

- f_c, center frequency
- Q quality factor
- Gain/attenuation in dB

Parametric EQs allow you to adjust the center frequency, Q and boost or cut creating any arbitrary bumps or notches in the frequency response (Figure 6.30). The parametric EQ is a variation on the ordinary band-pass and band-stop filters that generates symmetrical boost/cut curves and mixes in the dry signal to create the final response. A true digital parametric EQ not only has independent controls,

but each control only varies one coefficient in the filter. The parametric EQs in this section afford the same frequency response but adjustments in any parameter require a recalculation of all the coefficients. These filters are also called peaking filters.

This parametric EQ is not constant-Q, which means the bandwidth varies depending on the boost/cut value. Some analog filters have the same issue, although there is occasional debate over whether or not this is desirable in an EQ design. Figure 6.29 shows this EQ with three different boost curves with a center frequency of 1 kHz and $Q = 1.0$; therefore, the bandwidth should also be 1000 Hz.

Note: These filter coefficients contain the tan() function, which is undefined at $\pi/2$ and then flips negative between $\pi/2$ and π. The argument is $\theta_c/2Q$, so care must be taken with these filters to ensure that $\theta_c/2Q$ does not fall on the range of $\pi/2$ to π. One solution is to clamp the value of $\theta_c/2Q$ so that it never equals or exceeds $\pi/2$.

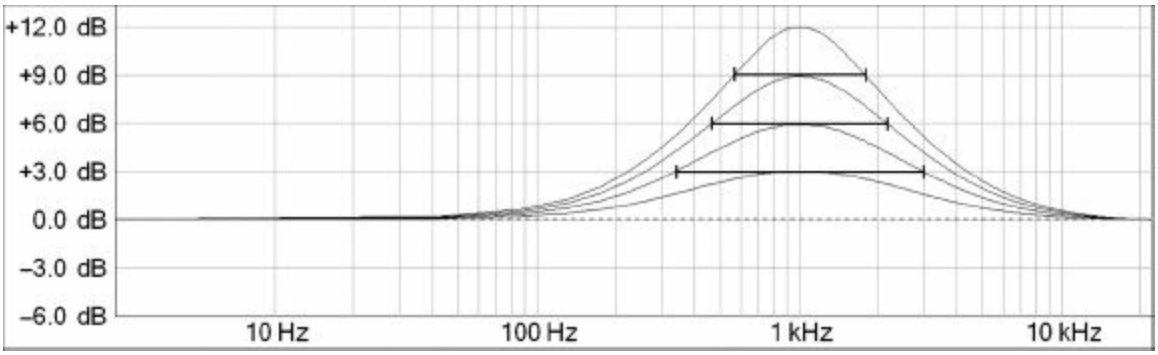


Figure 6.29: The non-constant-Q peaking filter has different bandwidths for different gain values; the bandwidth widens as the gain is reduced. $f_c = 1$ kHz, $Q = 0.707$.

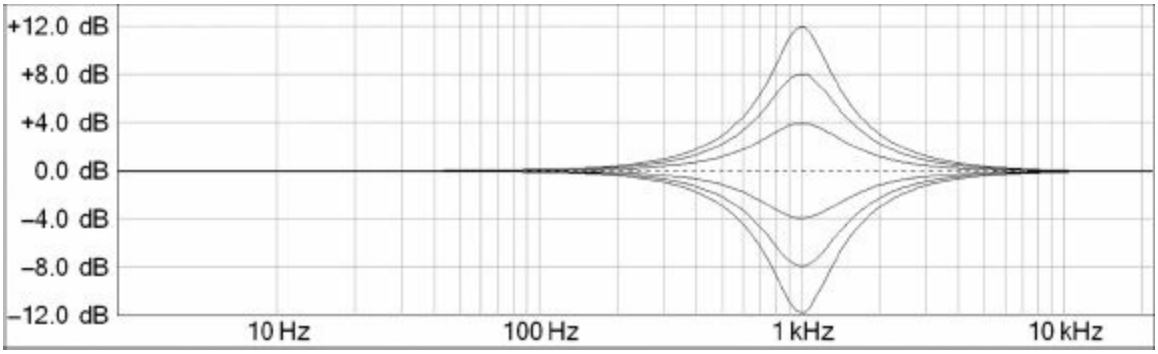


Figure 6.30: A set of responses for the non-constant-Q parametric/peaking filter with a variety of boost/cut settings. $f_c = 1$ kHz, $Q = 2$.

The design equations are as follows:

$$
\begin{aligned}
\theta_c &= 2\pi f_c / f_s \\
\mu &= 10^{Gain(\mathrm{dB})/20} \\
\zeta &= \frac{4}{1+\mu} \\
\beta &= 0.5\frac{1-\zeta\tan(\theta_c/2Q)}{1+\zeta\tan(\theta_c/2Q)} \\
\gamma &= (0.5+\beta)\cos\theta_c \\
a_0 &= 0.5-\beta \\
a_1 &= 0.0 \\
a_2 &= -(0.5-\beta) \\
b_1 &= -2\gamma \\
b_2 &= 2\beta \\
c_0 &= \mu - 1.0 \\
d_0 &= 1.0
\end{aligned}
\tag{6.42}
$$

6.7.4 Second-Order Parametric/Peaking Filter: Constant-Q

Specify:

- f_c, center frequency
- Q, quality factor
- Gain/attenuation in dB

This design creates an almost perfect constant-Q filter with only a small amount of error for low-boost (or cut) values (Figure 6.31). The effect of the constant-Q design is clearly evidenced in the frequency response plot (Figure 6.32).

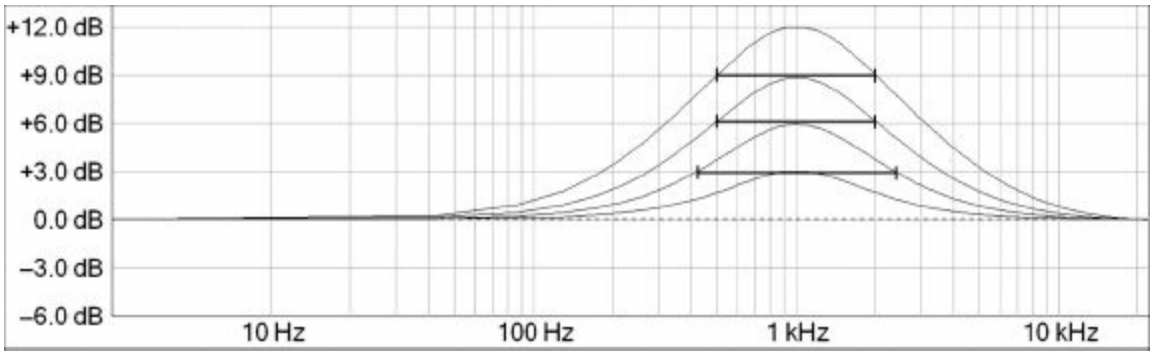


Figure 6.31: The constant-Q peaking filter preserves the bandwidth over most of the range of boost/cut values. f_c = 1kHz, Q = 0.707.

The design equations are as follows:

$$K = \tan(\pi f_c / f_s)$$
$$V_0 = 10^{boost/cut(\text{dB})/20}$$
$$d_0 = 1 + \frac{1}{Q}K + K^2 \qquad (6.43)$$
$$e_0 = 1 + \frac{1}{V_0 Q}K + K^2$$

	Boost	Cut
$\alpha = 1 + \frac{V_0}{Q}K + K^2$	$a_0 = \frac{\alpha}{d_0}$	$a_0 = \frac{d_0}{e_0}$
$\beta = 2(K^2 - 1)$	$a_1 = \frac{\beta}{d_0}$	$a_1 = \frac{\beta}{e_0}$
$\gamma = 1 - \frac{V_0}{Q}K + K^2$	$a_2 = \frac{\gamma}{d_0}$	$a_2 = \frac{\delta}{e_0}$
$\delta = 1 - \frac{1}{Q}K + K^2$	$b_1 = \frac{\beta}{d_0}$	$b_1 = \frac{\beta}{e_0}$
$\eta = 1 - \frac{1}{V_0 Q}K + K^2$	$b_2 = \frac{\delta}{d_0}$	$b_2 = \frac{\eta}{e_0}$
	$c_0 = 1.0$	$c_0 = 1.0$
	$d_0 = 0.0$	$d_0 = 0.0$

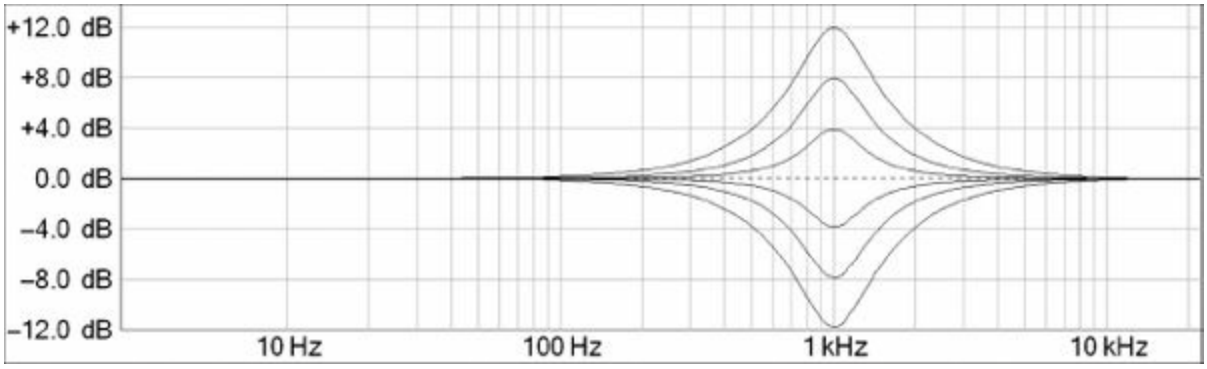


Figure 6.32: A set of responses for the constant-Q parametric/peaking filter with a variety of boost/cut settings. f_c = 1 kHz, Q = 2.

6.7.5 Cascaded Graphic EQ: Non-Constant-Q

Specify:

- f_c, center frequency
- Gain/attenuation in dB

The graphic EQ is a fixed Q variation on the parametric EQ. In a graphic EQ, the user only has control over the boost and cut of a set of filters tuned to fixed center frequencies. The Q that is used will depend on the boost or cut settings as well as the number of bands of EQ. For non-constant-Q graphic EQs, the method of finding the proper Q can be difficult because the bandwidth will change for various settings of the boost/cut. The value for Q is sometimes chosen so that when all the

controls are set to the maximum boost or cut values, the frequency response is as flat as possible.

The Q for constant-Q filters is easy to calculate knowing the number of bands split across the 10 octaves of audio. It is a good starting point for developing a non-constant-Q EQ plug-in; experimentation may be needed to find the ideal Q for your particular design. You might also allow the user to adjust the value within reason. The equation relating Q to the number of EQ bands is Equation 6.44.

$$N = \frac{\text{Number of modules}}{10}$$
$$Q = \frac{\sqrt{2^N}}{2^N - 1} \tag{6.44}$$

For a 10-band graphic EQ, $Q = 1.414$, while for a 31-band (aka 1/3 octave) EQ, $Q = 4.32$. The center frequencies of the bands are usually found with the following International Organization for Standardization (ISO) standard equation:

$$q = \frac{\text{Number of bands}}{10}$$
$$f_n = 1000 * 2^{n/q}$$
$$n = 0, \pm 1, \pm 2, \pm 3, \text{etc ...} \tag{6.45}$$

For a 10-band EQ, this works out (with rounding) to 32 Hz, 64 Hz, 125 Hz, 250 Hz, 500 Hz, 1 kHz, 2 kHz, 4 kHz, 8 kHz, and 16 kHz.

The cascaded graphic EQ topology is shown in Figure 6.33. It consists of a simple series cascade of each of the filter modules. The non-constant-Q graphic uses the same non-constant-Q peaking filter algorithm, but with the Q fixed according to Equation 6.42. You can get a good feel for how the Q affects the overall response by looking at a plot with all controls set to the maximum boost (or cut). For a 10-band non-constant-Q graphic, we observe the combined response shown in Figure 6.34.

6.7.6 Cascaded Graphic EQ: Constant-Q

Specify:

- f_c, center frequency
- Gain/attenuation in dB

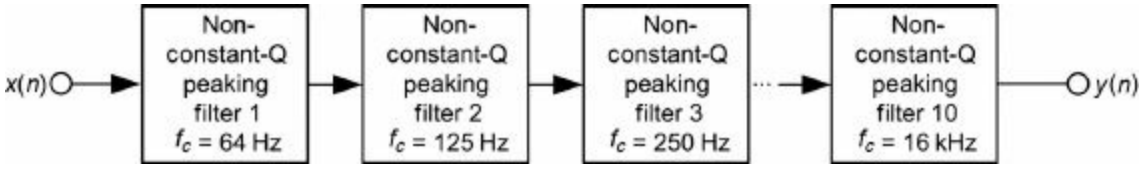


Figure 6.33: The cascaded graphic EQ topology; this is for a 10-band design using the ISO center frequencies. The design equations for each module use the non-constant-Q algorithm above.

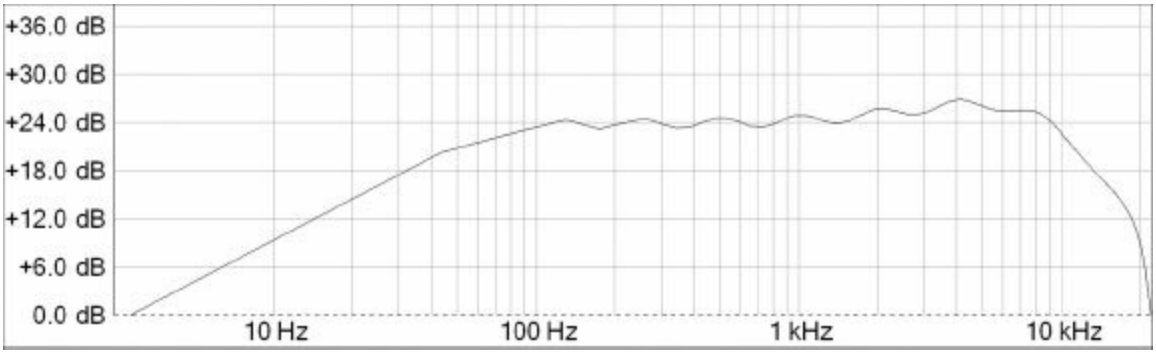


Figure 6.34: At the prescribed constant-Q value of $Q = 1.414$ we observe rippling and an increased high-frequency response with all controls at maximum boost.

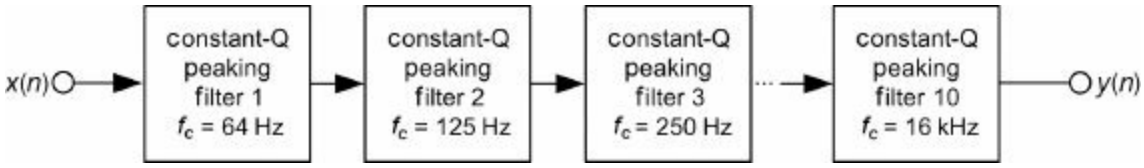


Figure 6.35: The cascaded graphic EQ topology; this is for a 10-band design using the exact unrounded ISO center frequencies. The design equations for each module use the constant-Q algorithm above.

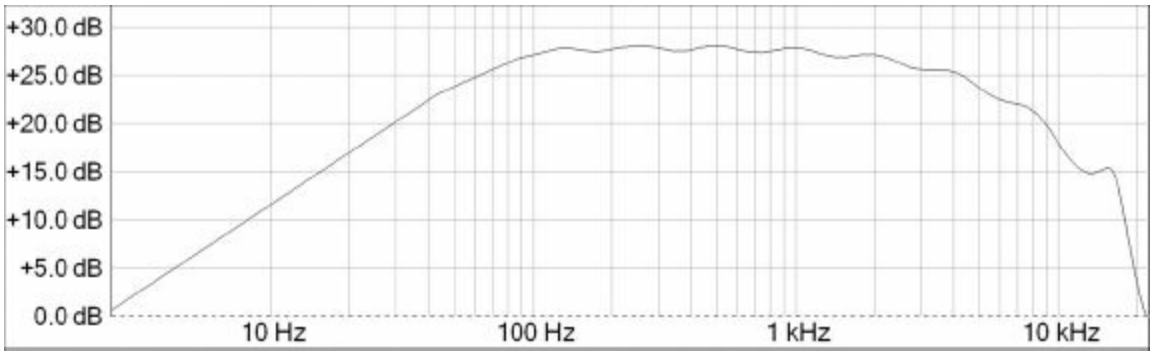


Figure 6.36: The 10-band constant-Q graphic EQ with the prescribed $Q = 1.414$ produces less low-frequency rippling and a decreased high-frequency response with all controls at maximum boost.

The constant-Q graphic EQ follows the same design pattern (Figure 6.35) except that you use the constant-Q peaking filter in each of the modules. The equations for finding the center frequencies and Q are the same as above. Bohn (1986) recommends not rounding the ISO center frequencies but rather use the exact values. Figure 6.36 shows the composite response with all controls set to full boost.

6.8 Design a Resonant LPF Plug-In

We'll continue with a resonant LPF design. The filter will use the second-order bilinear transform design and the bi-quad structure for implementation. After that you will redesign the filter using the Massberg technique for analog emulation and examine the differences. Note: This filter is found in RackAFX's module menu item under “HP/LP Filter” for you to experiment with and use to check your own implementations.

For this project, we'll be using our first built-in object. I've already prepared a bi-quad object for you

to use in this (or any other) project. It is not a RackAFX plug-in but rather a simple C++ object that accomplishes only one task: the bi-quad structure realization. The bi-quad can be used to make any first- or second-order feed-forward or feed-back type filter. Let's take a moment to check out the CBiquad object. You can find the interface in the pluginconstants.h file and the implementation in the pluginobjects.cpp files, respectively. The CBiquad object has the members and variables in Table 6.2.

Table 6.2: The CBiquad object interface.

CBiquad	
Member Variables	Purpose
protected: float m_f_Xz_1 float m_f_Xz_2 float m_f_Yz_1 float m_f_Yz_2	Implements the four delay elements needed for the bi-quad: x(n - 1), x(n - 2), y(n - 1) and y(n - 2) in these protected variables
public: float m_f_a0 float m_f_a1 float m_f_a2 float m_f_b1 float m_f_b2	The filter coefficients
Member Functions	
void flushDelays()	Initialize all delay elements with 0.0s
float doBiQuad(float f_xn)	Do one cycle of filter operation; return value is y(n)
Parameters	
• float f_xn	Input: the current input sample, x(n)

You can see that this object is simple; it only has two functions, one to reset and the other to do the bi-quad calculation. The coefficients must be calculated externally—this object has no knowledge of the filter type that it is implementing, the sample rate, and so on. You will see that this simple object will also save you a lot of coding for filters in later chapters. We will need to set up the user interface (UI) and then link the variables to the bi-quad objects in userInterfaceChange().

6.8.1 Project: ResonantLPF

Create the project and name it “ResonantLPF,” then add the sliders for the graphical user interface (GUI).

6.8.2 ResonantLPF GUI

Figure 6.37 shows what your final GUI will look like. You will need the controls shown in Table 6.3.

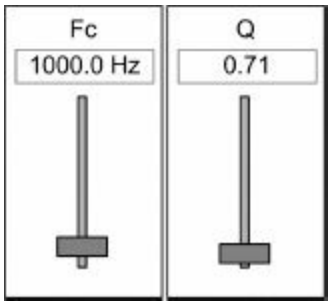


Figure 6.37: The resonant LPF GUI.

Table 6.3: The GUI controls for the resonant LPF.

Slider Property	Value
Control Name	Fc
Units	Hz
Variable Type	float
Variable Name	m_f_fc_Hz
Low Limit	100
High Limit	5000
Initial Value	1000

Slider Property	Value
Control Name	*Q*
Units	
Variable Type	float
Variable Name	m_f_Q
Low Limit	0.5
High Limit	20
Initial Value	0.707

6.8.3 ResonantLPF.h File

RackAFX provides you with the built-in bi-quad object named CBiquad and you don't need to do anything at all to add it to your project. You can just declare your bi-quad objects directly in the .h file like you would any other object:

```
CBiquad m_LeftLPF; // or whatever you like to name it
```

Here's my .h file with the left and right LPF objects declared:

```
// Add your code here: ------------------------------------------------------ //
CBiquad m_LeftLPF;
CBiquad m_RightLPF;

// END OF USER CODE --------------------------------------------------------- //
```

We'll also need a function to calculate the bi-quad's coefficients (a_0, a_1, a_2, b_1, and b_2) and we can share this between the left and right objects (i.e., both LPFs will have the same cutoff and Q, so they will also have the same coefficients). Declare this now:

```
// Add your code here: ------------------------------------------------------ //
CBiquad m_LeftLPF;
CBiquad m_RightLPF;

void calculateLPFCoeffs(float fCutoffFreq, float fQ);

// END OF USER CODE --------------------------------------------------------- //
```

6.8.4 ResonantLPF.cpp File

Write the calculateLPFCoeffs() function in the .cpp file by using Equation 6.35. I have used the same intermediate coefficient names here too.

```
void CResonantLPF::calculateLPFCoeffs(float fCutoffFreq, float fQ)
{
	// use same terms as in book:
	float theta_c = 2.0*pi*fCutoffFreq/(float)m_nSampleRate;
	float d = 1.0/fQ;

	// intermediate values
	float fBetaNumerator = 1.0 - ((d/2.0)*(sin(theta_c)));
	float fBetaDenominator = 1.0 + ((d/2.0)*(sin(theta_c)));

	// beta
	float fBeta = 0.5*(fBetaNumerator/fBetaDenominator);

	// gamma
	float fGamma = (0.5 + fBeta)*(cos(theta_c));

	// alpha
	float fAlpha = (0.5 + fBeta - fGamma)/2.0;

	// left channel
	m_LeftLPF.m_f_a0 = fAlpha;
	m_LeftLPF.m_f_a1 = 2.0*fAlpha;
	m_LeftLPF.m_f_a2 = fAlpha;
	m_LeftLPF.m_f_b1 = -2.0*fGamma;
	m_LeftLPF.m_f_b2 = 2.0*fBeta;

	// right channel
	m_RightLPF.m_f_a0 = fAlpha;
	m_RightLPF.m_f_a1 = 2.0*fAlpha;
	m_RightLPF.m_f_a2 = fAlpha;
	m_RightLPF.m_f_b1 = -2.0*fGamma;
	m_RightLPF.m_f_b2 = 2.0*fBeta;
}
```

prepareForPlay()

- Flush the bi-quad buffers.
- Cook the raw variables.

```
bool __stdcall CResonantLPF::prepareForPlay()
{
        // Add your code here:
        m_LeftLPF.flushDelays();
        m_RightLPF.flushDelays();

        // calculate the initial values
        calculateLPFCoeffs(m_f_fc_Hz, m_f_Q);

        return true;
}
```

userInterfaceChange()

- If the user moves either slider then we have to recalculate all the coefficients because they are interdependent on both f_c and Q.
- There is no need to check the slider nControlIndex value here.

```
bool __stdcall CResonantLPF::userInterfaceChange(int nControlIndex)
{
        // update coeffs for next time
        calculateLPFCoeffs(m_f_fc_Hz, m_f_Q);

        return true;
}
```

processAudioFrame()

- Call the methods on the filter objects to do the processing.
- They will automatically update their own delay lines and maintain themselves so the code is simple.

```
bool __stdcall CResonantLPF::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
                                    UINT uNumInputChannels, UINT uNumOutputChannels)
{
       // Do LEFT (MONO) Channel; there is always at least one input/one output
       // (INSERT Effect)
       pOutputBuffer[0] = m_LeftLPF.doBiQuad(pInputBuffer[0]);

       // Mono-In, Stereo-Out (AUX Effect)
       if(uNumInputChannels == 1 && uNumOutputChannels == 2)
              pOutputBuffer[1] = pOutputBuffer[0]; // Just copying

       // Stereo-In, Stereo-Out (INSERT Effect)
       if(uNumInputChannels == 2 && uNumOutputChannels == 2)
              pOutputBuffer[1] = m_RightLPF.doBiQuad(pInputBuffer[1]);

       return true;
}
```

Build and test your resonant LPF. Use the analyzer's frequency and phase response features to verify that all is working as you would expect. You now have a working resonant LPF, but the Nyquist

response is always clamped to 0.0, so let's investigate a technique to make the response more closely match the original analog filter.

6.9 The Massberg Analog-Matched Low-Pass Filter

At the 131st Audio Engineering Society Convention in 2011, Michael Massberg presented a solution to the clamping problem, producing first- and second-order low-pass filters that very closely match their analog counterparts near and at the Nyquist frequency. It takes advantage of the fact that a shelving filter with a very high upper shelf frequency has a roll-off portion that very closely resembles the roll-off of a low-pass filter. In a nutshell, the idea is to design a shelving filter whose upper shelf frequency is above Nyquist. In the range from DC to Nyquist, it will resemble a low-pass filter (Figure 6.38). A unique matching scheme is applied to force the two responses to match their gains exactly at a point halfway between the gain at DC and the gain at Nyquist in decibels, with a slight error at points above that frequency. The result is a filter that has an excellent approximation to the analog response it is trying to mimic.

Massberg used first- and second-order shelving filters to produce analog-matched low-pass filters applying the bilinear transform to produce the final result. Figure 6.39 shows a comparison of the standard digital LPF and the Massberg LPF.

6.9.1 First-Order Massberg LPF

Specify:

- f_c, corner frequency

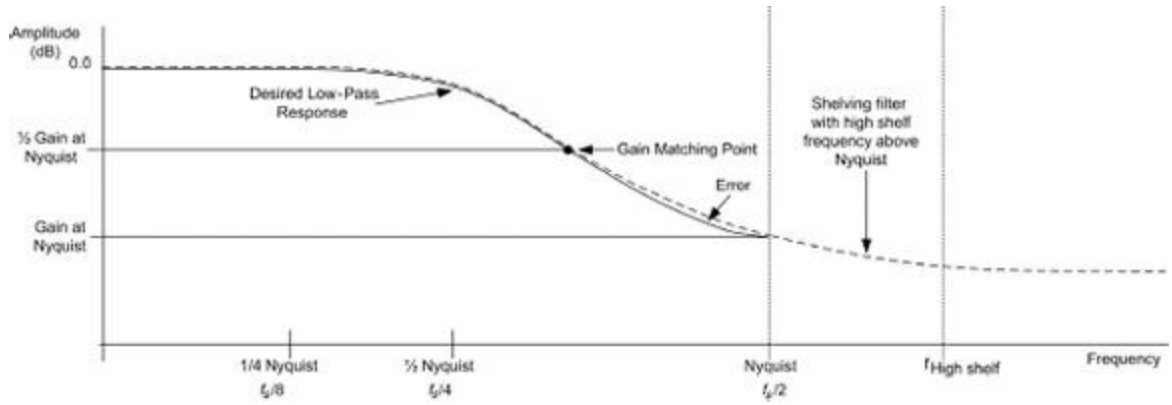


Figure 6.38: The Massberg filter uses a shelving filter as a prototype. In the range from DC to Nyquist, it very closely resembles the desired low-pass filter. The two curves are displaced slightly.

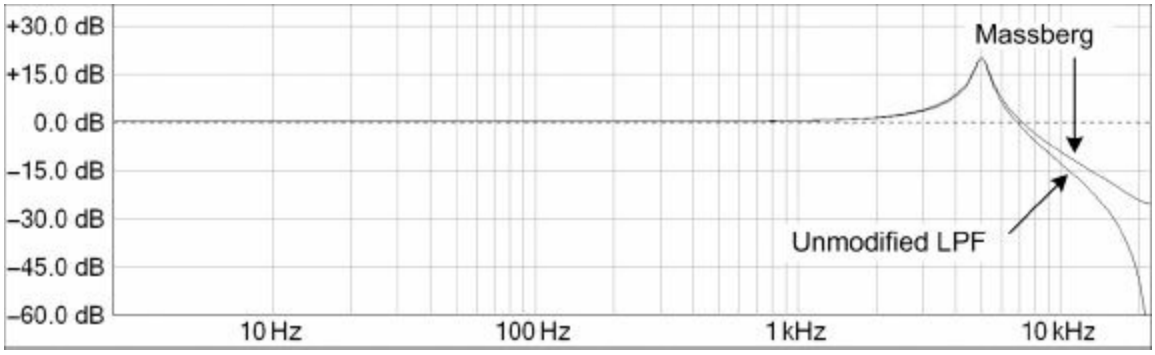


Figure 6.39: The Massberg and unmodified LPF responses with f_c = 5 kHz and Q = 10. The difference in the high-frequency response is evident.

The design equations are as follows:

$$
\begin{aligned}
g_1 &= \frac{2}{\sqrt{4 + \left(\frac{f_s}{f_c}\right)^2}} ac \\
g_m &= \max(\sqrt{0.5}, \sqrt{g_1}) \\
\omega_m &= \frac{2\pi f_c \sqrt{1 - g_m^2}}{g_m} \\
\Omega_m &= \tan\left(\frac{\omega_m}{2 f_s}\right) \\
\Omega_s &= \Omega_m \frac{\sqrt{(g_m^2 - g_1^2)(1 - g_m^2)}}{1 - g_m^2} \\
\gamma_0 &= \Omega_s + 1 \\
\alpha_0 &= \Omega_s + g_1 \\
\alpha_1 &= \Omega_s - g_1 \\
\beta &= \Omega_s - 1 \\
a_0 &= \frac{\alpha_0}{\gamma_0} \\
a_1 &= \frac{\alpha_1}{\gamma_0} \\
a_2 &= 0 \\
b_1 &= \frac{\beta_1}{\gamma_0} \\
b_2 &= 0
\end{aligned}
\tag{6.46}
$$

6.9.2 Second-Order Massberg LPF

Specify:

- f_c, corner frequency

- Q, quality factor controlling resonant peaking

The design equations are as follows:

$$\theta_c = 2\pi f_c / f_s$$

$$g_1 = \frac{2}{\sqrt{\left(2 - \left(\frac{\sqrt{2}\pi}{\theta_c}\right)^2\right) + \left(\frac{2\pi}{Q\theta_c}\right)^2}}$$

Calculate Ω_s depending on the value of Q:

$Q > \sqrt{5}$?	$Q \le \sqrt{5}$?
$g_r = \frac{2Q^2}{\sqrt{4Q^2 - 1}}$	$\omega_m = \theta_c \sqrt{\frac{2 - \frac{1}{2Q^2} + \sqrt{\frac{1 - 4Q^2}{Q^4} + \frac{4}{g_1}}}{2}}$
$\omega_r = \theta_c \sqrt{1 - \frac{1}{2Q^2}}$	$\Omega_m = \tan\left(\frac{\omega_m}{2}\right)$
$\Omega_r = \tan\left(\frac{\omega_r}{2}\right)$	$\Omega_s = \frac{\theta_c (1 - g_1^2)^{1/4}}{2}$ (6.47)
$\Omega_s = \Omega_r \left(\frac{g_r^2 - g_1^2}{g_r^2 - 1}\right)^{1/4}$	$\Omega_s = \min(\Omega_s, \Omega_m)$

Calculate the pole and zero frequencies (ω), gains (g) and Qs

$$\omega_p = 2 \arctan(\Omega_s) \qquad \omega_z = 2 \arctan\left(\frac{\Omega_s}{\sqrt{g_1}}\right)$$

$$g_p = \frac{1}{\sqrt{\left(1 - \left(\frac{\omega_p}{\theta_c}\right)^2\right) + \left(\frac{\omega_p}{Q\theta_c}\right)^2}} \qquad g_z = \frac{1}{\sqrt{\left(1 - \left(\frac{\omega_z}{\theta_c}\right)^2\right) + \left(\frac{\omega_z}{Q\theta_c}\right)^2}}$$

$$Q_p = \sqrt{\frac{g_1(g_p^2 - g_z^2)}{(g_1 + g_z^2)(g_1 - 1)^2}} \qquad Q_p = \sqrt{\frac{g_1^2(g_p^2 - g_z^2)}{g_z^2(g_1 + g_p^2)(g_1 - 1)^2}}$$

$$\gamma_0 = \Omega_s^2 + \frac{1}{Q_p}\Omega_s + 1$$

$$\alpha_0 = \Omega_s^2 + \frac{\sqrt{g_1}}{Q_z}\Omega_s + g_1$$

$$\alpha_1 = 2(\Omega_s^2 - g_1) \qquad \beta_1 = 2(\Omega_s^2 - 1)$$

$$\alpha_2 = \Omega_s^2 - \frac{\sqrt{g_1}}{Q_z}\Omega_s + g_1 \qquad \beta_2 = \Omega_s^2 - \frac{1}{Q_p}\Omega_s + 1 \tag{6.48}$$

$$a_0 = \frac{\alpha_0}{\gamma_0}$$

$$a_1 = \frac{\alpha_1}{\gamma_0} \qquad b_1 = \frac{\beta_1}{\gamma_0}$$

$$a_2 = \frac{\alpha_2}{\gamma_0} \qquad b_2 = \frac{\beta_2}{\gamma_0}$$

Challenge: Modify your digital resonant LPF plug-in to add the Massberg filter option, then experiment with high-fidelity music and listen for the differences in sound.

Bibliography

Allred, R. 2003. Second-order IIR filters will support cascade implementations: 5 part digital audio application tutorial. *EE Times Design Guide*. http://www.eetimes.com/design/audio-design/4009473/Second-order-IIRFilters-will-support-cascade-implementations—5-Part-digital-audio-application-tutorial. Accessed August 15, 2012.

Berners, D. P. and Abel, J. S. 2003. Discrete-time shelf filter design for analog modeling. *Journal of the Audio Engineering Society*, preprint 5939.

Bohn, D. 1986. Constant-Q graphic equalizers. *Journal of the Audio Engineering Society* 34(9): 611–25.

Bohn, D. 2008. Bandwidth in octaves versus Q in band-pass filters. Application Note 170. Mukilteo, WA: Rane Corp.

Dodge, C. and Jerse, T. 1997. *Computer Music Synthesis, Composition and Performance*, Chapter 6. New York: Schirmer.

Giles, M., ed. 1980. *The Audio/Radio Handbook*. Santa Clara, CA: National Semiconductor Corp.

Kwakernaak, H. and Sivan, R. 1991. *Modern Signals and Systems*, Chapter 9. Englewood Cliffs, NJ: Prentice-Hall.

Lane, J. et al. 2001. *DSP Filters*, Chapters 4–10 and 20. Carmel, IN: Howard W. Sams & Co.

Lane, J. and Hillman, G. 1991. *Implementing IIR/FIR Filters with Motorola's DSP56000/DSP56001*. APR7/D Rev1. Schomberg, ON: Motorola, Inc.

Leach, M. 1999. *Introduction to Electroacoustics and Audio Amplifier Design*, Chapter 6. Dubuque, IA: Kendall-Hunt.

Lindquist, C. 1977. *Active Network Design*, Chapter 2. Miami: Steward & Sons.

Lindquist, C. 1999. *Adaptive and Digital Signal Processing*, Chapter 5. Miami: Steward & Sons.

Massberg, M. 2011. Digital low-pass filter design with analog-matched magnitude response. *Journal of the Audio Engineering Society*, preprint 8551 (Massberg 2011).

Moore, R. 1990. *Elements of Computer Music*, Chapter 2. Englewood Cliffs, NJ: Prentice-Hall.

Motorola, Inc. 1991. *Digital Stereo 10-Band Graphic Equalizer Using the DSP56001*. APR2/D. Schomberg, ON: Motorola, Inc.

Oppenheim, A. V. and Schafer, R. W. 1999. *Discrete-Time Signal Processing*, 2nd ed., Chapter 7. Englewood Cliffs, NJ: Prentice-Hall.

Smith, J. O. and Angell, J. B. 1982. A constant gain digital resonator tuned by a single coefficient. *Computer Music Journal* 4(4): 36–40.

Zöler, U. 2011. *Digital Audio Effects*, 2nd ed., Chapter 2. West Sussex, U.K.: John Wiley & Sons.

References

Ifeachor, Emmanuel C. and Jervis, Barrie W. 1993. *Digital Signal Processing: A Practical Approach*. Menlo Park: Addison-Wesley. pp. 398–400.

Orfanidis, Sophacles. 1997. Digital Parametric Equalizer Design with Prescribed Nyquist-Frequency Gain. *Journal of the Audio Engineering Society* 45(6): 444–55.

Delay Effects and Circular Buffers

Before we can start looking at some finite impulse response (FIR) algorithms, we need to deal with the concept of long delay lines or circular buffers. Not only are they used for the delay effects, but also they are needed to make long FIR filters. In this chapter we'll take a break from the DSP filter algorithms and develop some digital delays. If you think back to the infinite impulse response (IIR) filters you've worked on so far you will remember that after implementing the difference equation, you need to shuffle the z^{-1} delay element values. You do this by overwriting the delays backwards, like this:

```
m_f_z2 = m_f_z1;
m_f_z1 = xn;          // xn is the input sample
```

Suppose you had a filter that was higher than a second-order one and you had to implement z^{-4} using discrete variables for each delay element. You might wind up writing something like this to do the shuffling:

```
m_f_z4 = m_f_z3;
m_f_z3 = m_f_z2;
m_f_z2 = m_f_z1;
m_f_z1 = xn;          // xn is the input sample
```

But what happens when the delay line gets really long, like z^{-1024} or, for a 1-second digital delay, z^{-44100}? It's going to be difficult to implement the delay shuffling this way. Not only would it be tedious to code, it would also be very inefficient to have to implement all those read/write operations each sample period. The answer to the problem of long delay lines is called circular buffering.

Digital signal processors (DSPs) have an interesting feature in the address generation unit (AGU), which allows you to declare a buffer of data to be addressed circularly. This kind of addressing is not built into the C++ programming language's array access operation. We will have to code the mechanism ourselves. When you declare a buffer, the addressing is linear.

This means that the pointer will add or subtract the offset you provide and move linearly to the next location. Suppose you declare a buffer of floats like this:

```
float Buffer[1024];
```

and a pointer to the buffer:

```
float* pBuffer = &Buffer[0];
```

and then you enter a loop which accesses the data, incrementing by five addresses each time through the loop:

```
for(int n=0; n<someNumber; n+=5)
{
        float data = pBuffer[n];
}
```

What happens if the pointer is accessed outside the buffer, as shown in Figure 7.1 ? Usually a crash or debug halt. You always have to be careful when setting up a loop like that one to make sure the pointer never accesses outside the buffer. But what if you are stuck with a certain sized buffer and a different sized loop so that you are never sure if the pointer is going to go outside the buffer on the next iteration? In a DSP chip, when you declare a buffer to be circular, you are setting up a pointer access mechanism. Reads and writes are made with a pointer, which is incremented or decremented by some amount to generate the next access location. If the amount of offset causes the pointer to go outside the bounds of the buffer, it automatically wraps to the other side, including the amount of offset, as depicted in Figure 7.2.

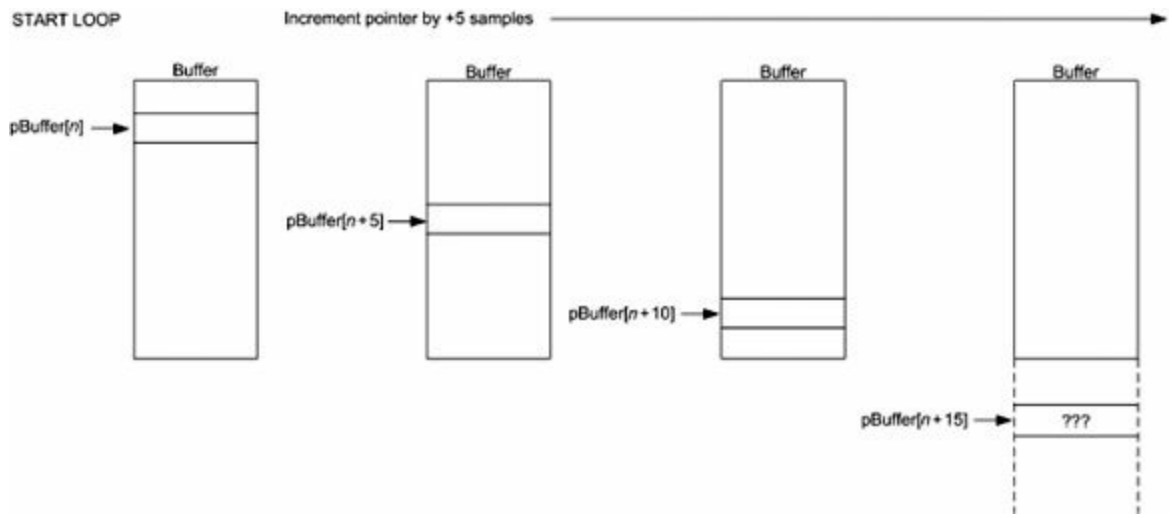


Figure 7.1: After several accesses, the pointer goes outside the buffer into an unknown location, usually resulting in a crash.

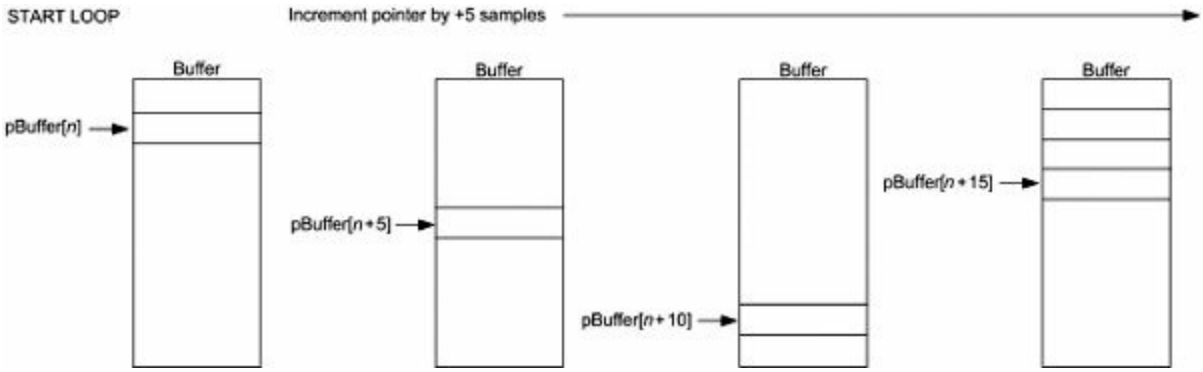


Figure 7.2: In a circular buffer, the pointer is automatically wrapped back to the top and offset by the proper amount to continue the access-by-five-samples loop.

Circular buffers are useful in audio signal processing. You can create circular buffers of audio samples or buffers of coefficients and loop through and access them automatically. Let's start with the most obvious use and make a digital delay effect.

7.1 The Basic Digital Delay

The digital delay effect or DDL (digital delay line) consists of a long buffer for storing audio samples. The samples enter one end of the buffer and come out the other end after *D* samples of delay, which corresponds to *D* sample periods. A feedback path allows for delay regeneration, or repeated echoes, as shown in Figure 7.3. With the feedback control *fb* set to 0, there is only a single delayed sample. With any other value, repeated echoes will form at the rate corresponding to the length of the delay line. The block diagram is shown in Figure 7.3.

The difference equation is as follows:

$$y(n) = x(n-D) + fb * y(n-D) \quad (7.1)$$

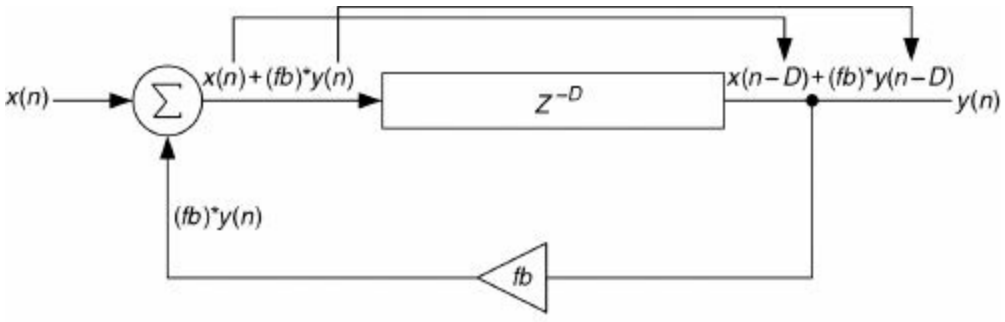


Figure 7.3: Basic DDL.

From the difference equation in Equation 7.1, you can see that the output consists of an input sample delayed by *D* samples plus a scaled version of the output at that time, $fb * y(n-D)$. The sequence of accessing the delay line during the processAudioFrame() function is as follows:

1. Read the output value, y(n), from the DDL and write it to the output buffer.
2. Form the product $fb * y(n)$.
3. Write the input value, $x(n) + fb * y(n)$, into the delay line.

You might notice that something is missing here: the shuffling of the samples through the delay. If we use a circular buffer as the delay line, then we don't have to shuffle data around, but we do have to keep track of the read and write access locations in the buffer and wrap the pointers as needed. In order to understand how the buffer operates to make a delay, consider a circular buffer that we've been writing samples into each sample period and automatically wrapping the pointer (or index) back to the top of the buffer as needed. The code would look something like this:

```
// buffer size, 1 second of audio data
int nBufferLength = 44100;

float Buffer[nBufferLength];
.
.
.
. // somewhere else in code:
int nIndex = 0;
.
.


      // inside a loop of some kind:
 pBuffer[nIndex] = audioSample;

 nIndex++;

 // if we go outside the buffer
 if(nIndex >= nBufferLength) // if we hit nBufferLength, we are one sample outside
        nIndex = 0;     // wrap the pointer back to the top for the next iteration
 .
 .
```

Suppose this has been going on for some time, and we are left with the buffer looking like Figure 7.4 *after* the last write access and just before we increment the pointer index. If pBuffer is pointing to the current sample value $x(n)$

- Where is the $x(n-1)$ sample (the youngest delayed value)?
- Where is the oldest sample in the delay?

In Figure 7.5 the youngest sample, $x(n-1)$, is in the location just before pBuffer[i], that is pBuffer[i −1]. The oldest sample is found by working backwards to the top of the buffer, wrapping back to the bottom, and locating the oldest sample written; it is at pBuffer[i+1]. If the pointer is accessing the samples sequentially from top to bottom, then the youngest sample is just above $x(n)$ and the oldest is just below it. It is easy to understand that the youngest sample is $x(n-1)$ but why is the oldest sample labeled $x(n-D+1)$ rather than $x(n-D)$?

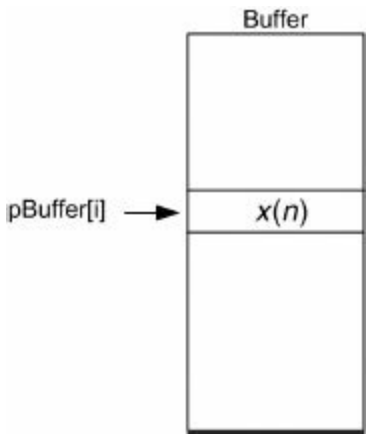


Figure 7.4: After many loops through the buffer, pBuffer[i] points to *x* (*n*).

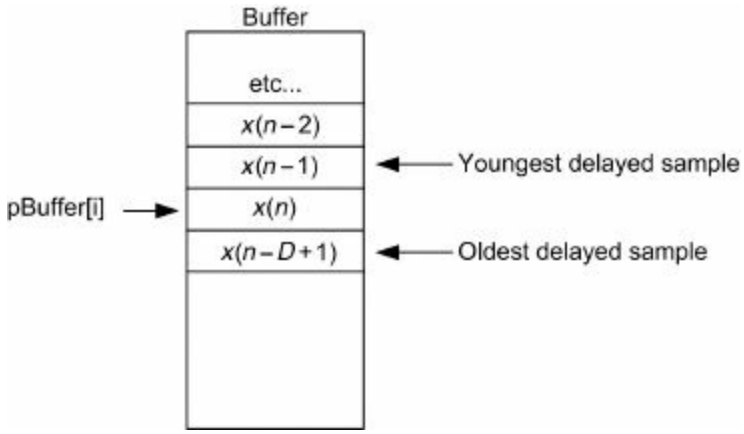


Figure 7.5: The youngest and oldest samples in the delay line.

The answer to the question is that we overwrote the actual oldest sample, $x\,(n-D)$, when we wrote in $x\,(n)$. This is one of the reasons for our rule about always performing reads before writes in our algorithms. This means that to get the oldest sample in the delay, you must first read out pBuffer[i] before writing into it. In other words, *before* the write access, the buffer looks like Figure 7.6; you can see the oldest value $x\,(n-D)$ is actually the location of the current write operation.

The steps for creating the delay line in your plug-in are as follows:

1. Decide on the maximum amount of delay you need to provide.
2. Declare read and write index values.
3. Declare a float buffer for each channel, right and left: for very long delay lines this is traditionally done with the *new* operator in the constructor of the plug-in.
4. Initialize the buffers with 0.0 using the memset() function.

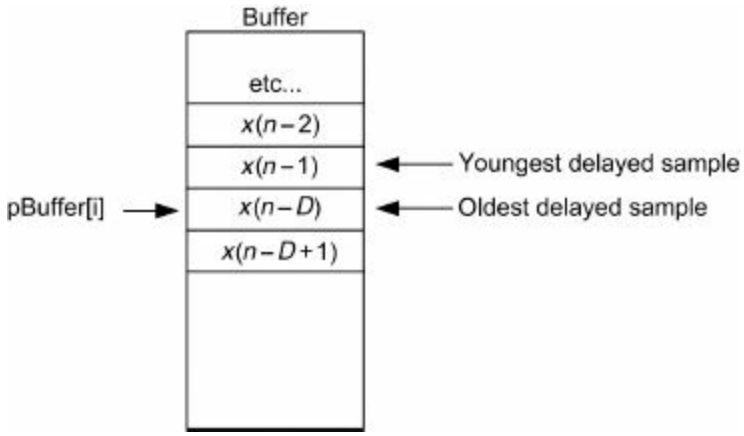


Figure 7.6: The location of the oldest audio sample $x\,(n-D)$.

During the processAudioFrame() function, you will need to

- Use the read and write index values to operate the delay.
- Increment or decrement the indices according to the algorithm you are using.
- Check to see if you need to wrap the index values for the next sample period.

In order to use the delay line, the user needs to provide a value for the delay amount in samples (in Section 7.3 we will allow them to enter the delay in milliseconds instead, but we will convert this to

a sample count inside the delay). In your code, there are two basic ways to do this:

1. Subtract the delay amount (in samples) from your write index, wrapping the pointer backwards if needed, to locate the delayed sample.
2. Store a read index and offset the two indices when the user changes the delay amount.

The second option is better because you only have to offset the index values and perform the subtraction/wrap when the user changes the delay value, instead of each time through the processAudioFrame() loop. Suppose the user selects 100 samples of delay time. You have declared two indices, m_nRead and m_nWrite, to use for buffer. During the processAudioFrame() function you will need to do the following five steps.

Step 1: Read out the delayed audio data, $d\ (n-D-100)$, which is 100 samples behind the current $x\ (n)$ sample time; this value is $y\ (n)$, the current output value (Figure 7.7).

```
float yn = pBuffer[m_nRead];
```

Step 2: Form the input combination input + feedback * output:

```
float xn = pInputBuffer[0] + m_fFeedBack*yn;
```

m_fFeedBack is declared in your .h file; this example code is for processing the left channel, pInputBuffer[0].

Step 3: Write the input data into the delay line at the m_nWrite location (Figure 7.8).

Step 4: Increment the read and write pointers by one to set up for the next time through the function; check for wrapping and do that if necessary. Can you think of other ways to do this?

```
// inc/wrap
m_nWrite++;
if(m_nWrite >= m_ nBufferLength)
       m_nWrite = 0;

m_nRead++;
if(m_nRead >= m_ nBufferLength)
       m_nRead = 0;
```

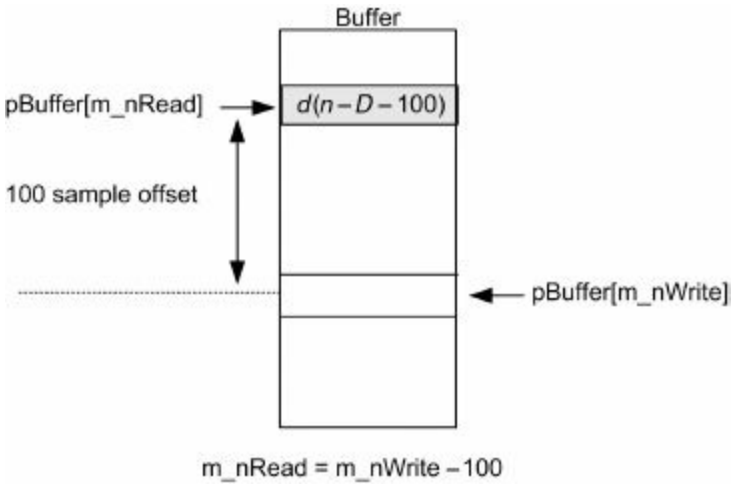


Figure 7.7: The delayed sample is read at location *d*(*n*–*D*–100), 100 samples before the current write location.

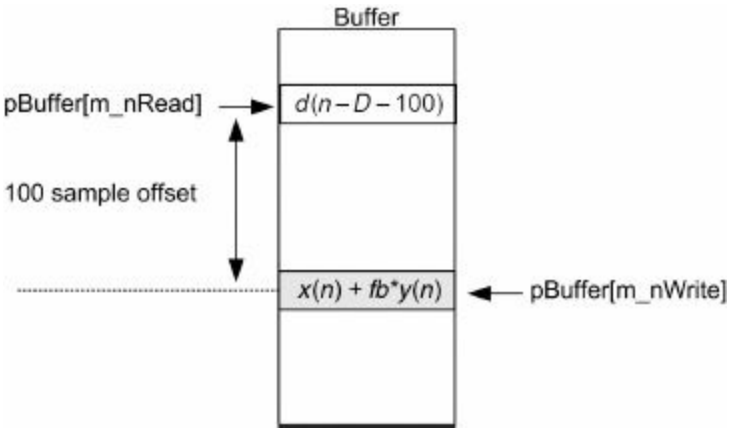


Figure 7.8: The delayed sample plus feedback is written into the current write location.

Notice that we wrap if the incremented index hits nBufferLength because this references the location just one sample outside the delay line.

Step 5: In the event that the user changes the delay time, you need to recalculate the m_nRead index to accommodate it. Note that once set up, the m_nWrite index is never changed except to increment it through the line. In this example, the user has selected nSamples of delay in the plug-in.

```
// user selects nSample delay
// first subtract the index
m_nRead = m_nWrite - nSamples;

//the check and wrap BACKWARDS if the index is negative
if (m_nRead < 0)
      m_nRead += nBufferLength;  // amount of wrap is Read + Length
```

7.2 Digital Delay with Wet/Dry Mix

Although there may be some instances where you don't need a wet/dry mix, in general when you make a delay plug-in, you still want to hear your unaffected signal. This can be done by using the plug-in as an Auxiliary (AUX) Send effect rather than an insert effect. But, to provide both capabilities, you need to provide the user with a wet/dry ratio control. The block diagram is slightly modified as shown in Figure 7.9.

The difference equation is as follows:

$$y(n) = \text{dry} * x(n) + \text{wet} * [x(n-D) + fb * y(n-D)] \tag{7.2}$$

7.2.1 Frequency and Impulse Responses

Consider the basic delay with *no* feedback applied and with the wet/dry ratio at 1:1. The block diagram and difference equation would then reduce down to Figure 7.10.

The difference equation is as follows:

$$y(n) = x(n) + x(n-D) \tag{7.3}$$

To find the frequency response, first take the *z* transform of the difference equation and form the

transfer function:

$$\begin{aligned} y(n) &= x(n) + x(n-D) \\ Y(z) &= X(z) + X(z)z^{-D} \\ &= X(z)(1+z^{-D}) \\ H(z) &= \frac{Y(z)}{X(z)} = 1+z^{-D} \end{aligned} \tag{7.4}$$

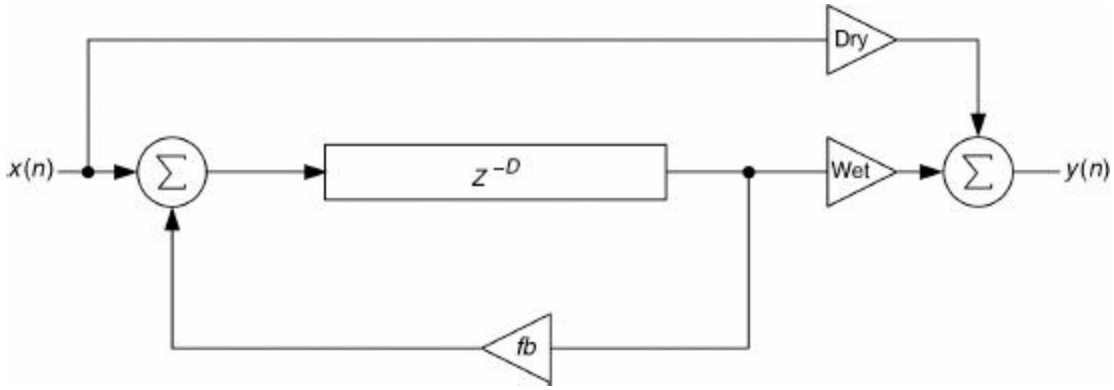


Figure 7.9: A more useful digital delay incorporates wet and dry controls.

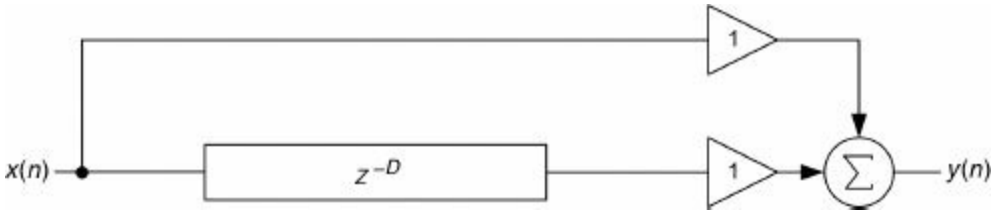


Figure 7.10: The simplified DDL of *D* samples delay.

Next, calculate the poles and zeros of the transfer function. We can see that this is a pure feed-forward filter in its current state so there are only zeros. We need to find the roots of the equation for $H(z)$:

$$\begin{aligned} 0 &= 1 + z^{-D} \\ &= z^{D} + 1 \\ z^{D} &= -1 \end{aligned}$$

$$\begin{aligned} &\text{Let } z = e^{j\omega} \\ &e^{jD\omega} = -1 \end{aligned} \tag{7.5}$$

Apply Euler
$\cos(D\omega) + j\sin(D\omega) = -1$

The roots (zeros) of the function lie wherever $D\omega$ causes the equation to evaluate to $-1 + j0$, and we know this will happen when Equation 7.6 holds true.

$$\begin{aligned} &\cos(\Theta) + j\sin(\Theta) = -1 + j0 \\ &if \\ &\Theta = \pm\pi, \pm 3\pi, \pm 5\pi, \text{etc...} \end{aligned} \tag{7.6}$$

Notice that both $\pm\pi$ produce the desired result of $-1 + j0$ as well as all the odd multiples of π: $\pm 3\pi$,

±5π, and so on. So the actual solution to find the roots becomes Equation 7.7:

$$\begin{aligned}&\cos(D\omega) + j\sin(D\omega) = -1 + j0\\&\text{if}\\&D\omega = \pm\pi, \pm 3\pi, \pm 5\pi, \ldots, \pm N\pi\\&\text{until}\\&N > D-1\end{aligned} \tag{7.7}$$

or

$$\text{zeros at } \omega = \pm\frac{k\pi}{D} \quad k = 1, 3, 5, \ldots, D$$

After $N > D - 1$, the whole mathematical sequence repeats again, cycling through odd multiples of p. This means that there are D zeros spread equally around the unit circle. This makes sense—the fundamental theorem of algebra predicts D roots for a polynomial of order D. Now consider the simple case of $D = 2$ samples; we get Equation 7.8:

$$\begin{aligned}&\cos(2\omega) + j\sin(2\omega) = -1\\&\text{if}\\&\omega = \pm\frac{k\pi}{D} \quad k = 1, 3, 5, \ldots, D\\&\omega = \pm\frac{\pi}{2}\end{aligned} \tag{7.8}$$

There are two zeros, one at $+\pi/2$ and the other at $-\pi/2$. Plot those on the unit circle in the z-plane and you can see what the frequency response will be, shown in Figure 7.11. You can see from Figure 7.11 that the zeros produce a notch (zero of transmission) at $\pm\pi/2$. In fact, when the delay time is very small, your ears hear the effect as a frequency response change; your ears cannot discriminate two samples that are only 23 uS apart as separate echoes. Now consider what would happen if we increase the delay amount to four samples, as in Figure 7.12. Finally, what would happen if the delay is an odd value, like $D = 5$ (Figure 7.13)?

$$\begin{aligned}&\cos(D\omega) + j\sin(D\omega) = -1 + j0\\&\cos(4\omega) + j\sin(4\omega) = -1 + j0\\&\omega = \pm\frac{\pi k}{D} \quad k = 1, 3, 5, \ldots, D\\&\omega = \pm\frac{\pi}{4}, \pm\frac{3\pi}{4}\end{aligned} \tag{7.9}$$

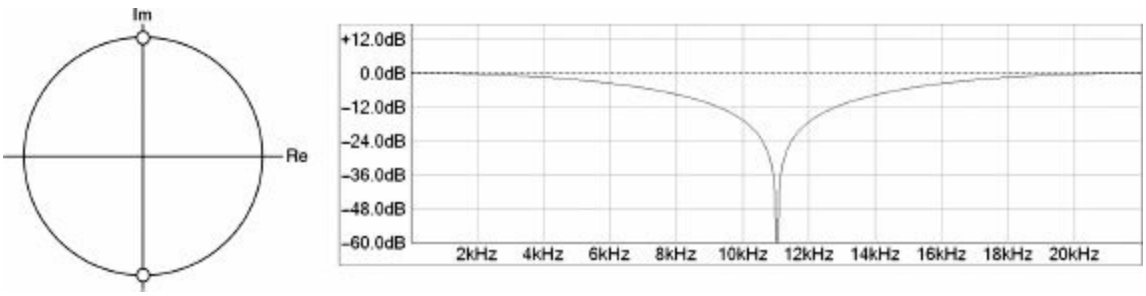


Figure 7.11: The z-plane pole/zero plot and resulting frequency response.

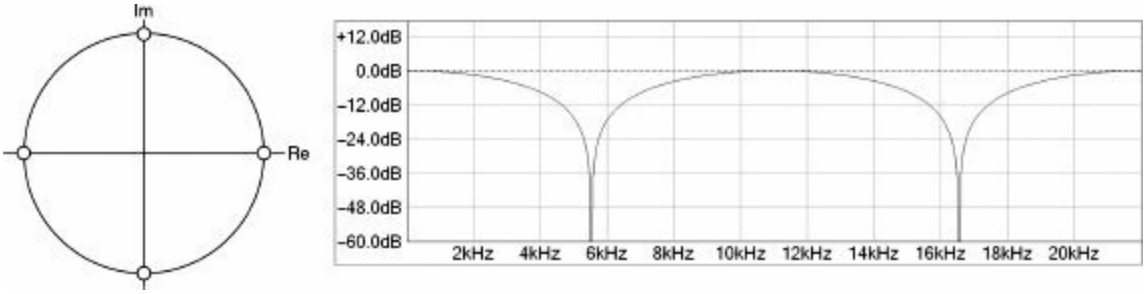


Figure 7.12: The *z*-plane pole/zero plot and resulting frequency response for *D* = 4 samples.

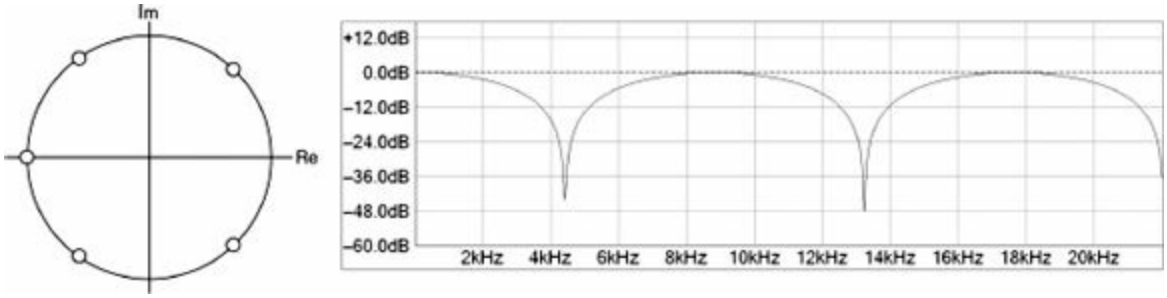


Figure 7.13: The *z*-plane pole/zero plot and resulting frequency response for *D* = 5 samples.

$$\begin{aligned}
&\cos(D\omega) + j\sin(D\omega) = -1 + j0 \\
&\cos(5\omega) + j\sin(5\omega) = -1 + j0 \\
&\omega = \pm\frac{k\pi}{5} \quad k = 1, 3, 5, \ldots, D-1 \\
&\omega = \pm\frac{\pi}{5}, \pm\frac{3\pi}{5}, \pm\frac{5\pi}{5} \\
&\omega = \pm\frac{\pi}{5}, \pm\frac{3\pi}{5}, \pm\pi
\end{aligned} \tag{7.10}$$

This kind of frequency response in Figure 7.13 is called *inverse comb filtering*. As we add more and more samples of delay, we add more and more notches to the response. You can use the built in module in RackAFX to experiment. Figures 7.14 and 7.15 show the frequency response for 32 samples of delay—it's an inverse comb filter with 16 zeros in the positive frequency domain.

7.2.2 The Effect of Feedback

When you add feedback to the delay, two things happen: first, for long delays your ear will hear discrete repeating echoes that decay away after the signal is removed. As the delay time gets shorter and shorter, the echoes merge, begin to ping, and then the actual filtering (which has been going on all the time) now dominates what you hear.

In Figure 7.16, you can see the effect of feedback on the impulse response. The initial impulse is the one through the dry path, and the next identical-sized one is the first output of the delay line. The echoes decay, being reduced 90% on each trip through the filter. To understand the effect on frequency response, write the transfer function and take the *z* transform. To make it easier to find the pole frequencies, let the feedback value become 100% or 1.0, as shown in the block diagram in Figure 7.17; even though we know this would result in oscillation, it will make calculating the frequencies easier.

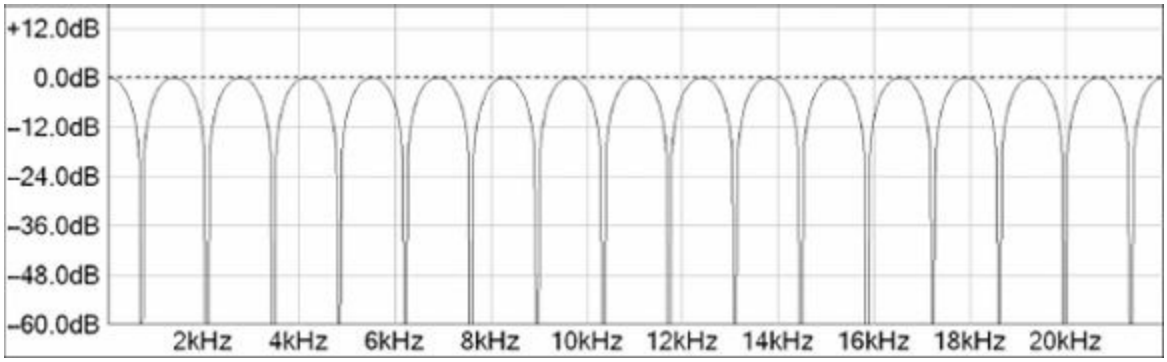


Figure 7.14: Frequency response (linear) with *D* = 32 samples.

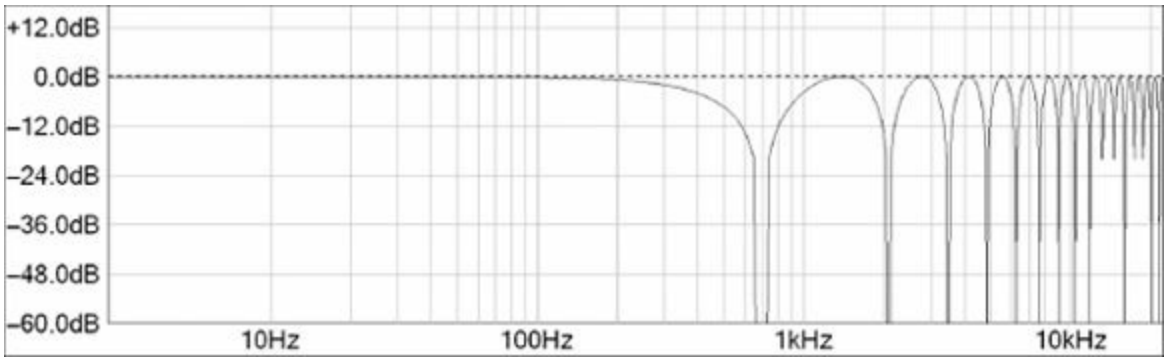


Figure 7.15: Frequency response (log) with *D* = 32 samples.

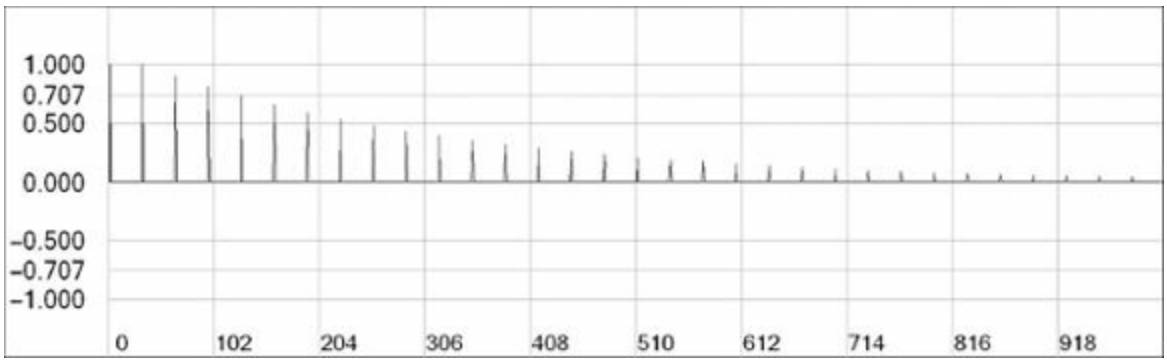


Figure 7.16: Impulse response with 90% feedback, 32 samples of delay.

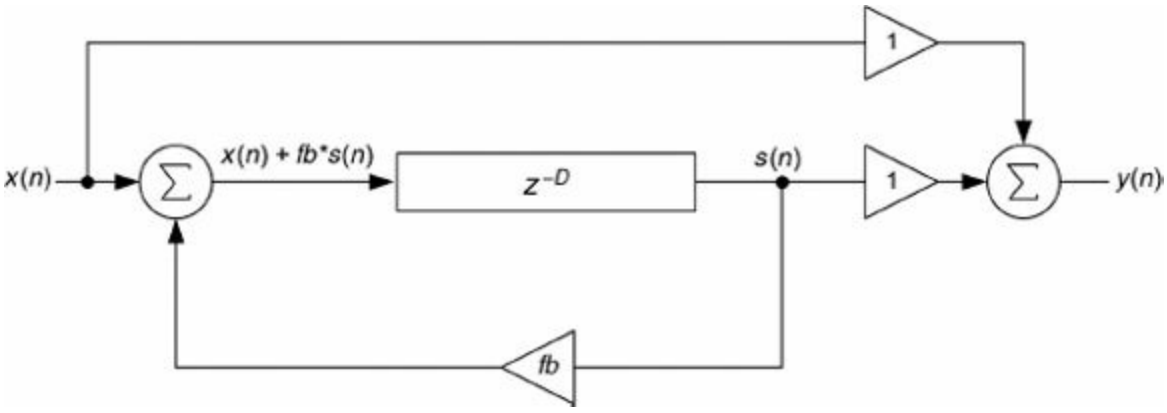


Figure 7.17: Block diagram of the DDL with feedback.

The difference equation is as follows:

$$y(n) = x(n) + x(n-D) - fb*x(n-d) + fb*y(n-D) \tag{7.11}$$

To derive the difference equation, label the output of the delay line $s(n)$ (Equation 7.12):

$$\begin{aligned}
&\text{The input to the delay line} = x(n) + fb^{*}s(n)\\
&\text{therefore}\\
&s(n) = x(n-D) + fb^{*}s(n-D)\\
&\text{and}\\
&y(n) = x(n) + s(n)\\
&\quad = x(n) + x(n-D) + fb^{*}s(n-D)
\end{aligned} \tag{7.12}$$

$$\begin{aligned}
&\text{rearranging:}\\
&s(n) = y(n) - x(n)\\
&s(n-D) = y(n-D) - x(n-D)
\end{aligned} \tag{7.13}$$

Substituting Equation 7.13 into Equation 7.12 gives you the following difference equation:

$$\begin{aligned}
y(n) &= x(n) + x(n-D) + fb^{*}[y(n-D) - x(n-D)]\\
&= x(n) + x(n-D) + fb^{*}y(n-D) - fb^{*}x(n-D)
\end{aligned} \tag{7.14}$$

To find the transfer function, separate variables and take the z transform:

$$\begin{aligned}
y(n) - fb^{*}y(n-D) &= x(n) + x(n-D) - fb^{*}x(n-D)\\
Y(z) - fbY(z)z^{-D} &= X(z) + X(z)z^{-D}[1 - fb]\\
Y(z)[1 - fbz^{-D}] &= X(z)[1 + z^{-D} - fbz^{-D}]\\
H(z) = \frac{Y(z)}{X(z)} &= \frac{1 + z^{-D} - fbz^{-D}}{1-fbz^{-D}}\\
H(z) &= \frac{1 + (1-fb)z^{-D}}{1-fbz^{-D}}
\end{aligned} \tag{7.15}$$

The new transfer function has both zeros (which we already calculated) and poles. The poles are caused by the feedback and will occur whenever the denominator becomes zero. If we let $fb = 1.0$, then equation Equation 7.15 reduces to Equation 7.16.

$$\begin{aligned}
H(z) &= \frac{1}{1 - z^{-D}}\\
0 &= 1 - z^{-D}\\
0 &= z^{D} - 1\\
z^{D} &= 1
\end{aligned}$$

$$\begin{aligned}
&\text{Let } z = e^{j\omega}\\
&e^{jD\omega} = 1
\end{aligned} \tag{7.16}$$

$$\begin{aligned}
&\text{Apply Euler}\\
&\cos(D\omega) + j\sin(D\omega) = 1
\end{aligned}$$

$$\begin{aligned}
&\cos(\Theta) + j\sin(\Theta) = 1 + j0\\
&\text{if}\\
&\Theta = 0, \pm 2\pi, \pm 4\pi, \pm 6\pi, \text{etc...}
\end{aligned}$$

$$\cos(D\omega) + j\sin(D\omega) = 1 + j0$$

if

$$D\omega = 0, \pm\frac{2\pi}{D}, \pm\frac{4\pi}{D}, \pm\frac{6\pi}{D}, \ldots, \pm\frac{N\pi}{D}$$

until

$$N > D-1 \tag{7.17}$$

or

$$\text{poles at } \omega = \pm\frac{k\pi}{D} \quad k = 0, 2, 4, 6, \ldots, D$$

Equation 7.17 shows that the poles will occur at even multiples of π and DC (0 Hz) when Euler's equation becomes $1+j0$; the analysis is nearly identical to the zero case. Consider the case of four samples:

$$\begin{aligned}
&\cos(D\omega) + j\sin(D\omega) = 1 + j0 \\
&\cos(4\omega) + j\sin(4\omega) = 1 + j0 \\
&\omega = \pm\frac{\pi k}{D} \quad k = 0, 2, 4, 6, \ldots, D \\
&\omega = 0, \pm\frac{\pi}{2}, \pm\frac{4\pi}{4} \\
&\omega = 0, \pm\frac{\pi}{2}, \pm\pi
\end{aligned} \tag{7.18}$$

Figure 7.18 shows the effect of 100% feedback – the response is technically infinite at the pole frequencies. It produces a comb filter (with the teeth of the comb pointing up) rather than the inverse comb filter you saw when examining the zeros. The amount of feedback will affect the radius of the poles and the zeros, but not the pole or zero frequencies, which are only dependent on the amount of delay. Consider the transfer function with a feedback value of 0.75:

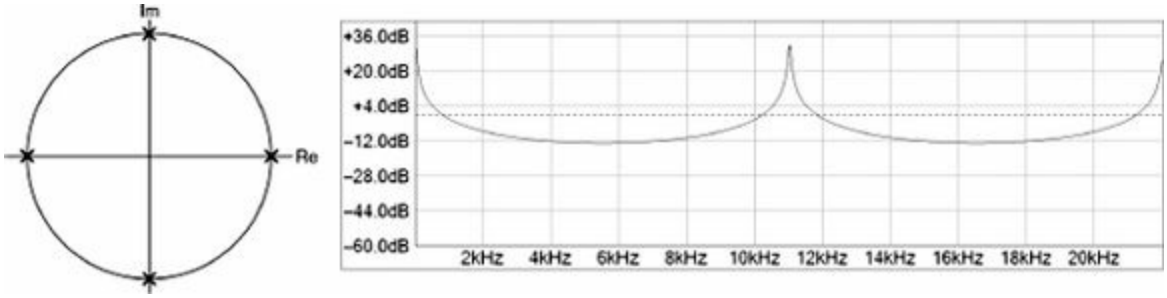


Figure 7.18: The *z*-plane pole/zero plot and resulting frequency response for $D = 4$ samples with $fb = 1.0$. The peaks theoretically go to infinity.

$$\begin{aligned}
H(z) &= \frac{1 + (1-fb)z^{-D}}{1 - fbz^{-D}} \\
H(z)|_{fb=0.75} &= \frac{1+0.25z^{-D}}{1-0.75z^{-D}}
\end{aligned} \tag{7.19}$$

The poles will have a radius of 0.75, while the zeros will have a radius of 0.25. This will result in

the z-plane plot and frequency response in Figure 7.19. You can see that the lowered radius results in less gain at the pole frequencies. The peaks are now softer and the overall gain is reduced down to about +8 dB from infinity. If you continue to drop the feedback to 50% (0.5) then the poles and zeros will be distributed at equal radii, as shown in Figure 7.20.

As the feedback gets smaller and smaller, the response will turn into the pure inverse comb filtering when the poles disappear (by converging on $z = 0$) and the response goes to 0.0 at the zero frequencies. What would happen if we inverted the feedback? This would mean that the feedback value is a negative percentage. Consider –50% feedback and look at the transfer function in Equation 7.20:

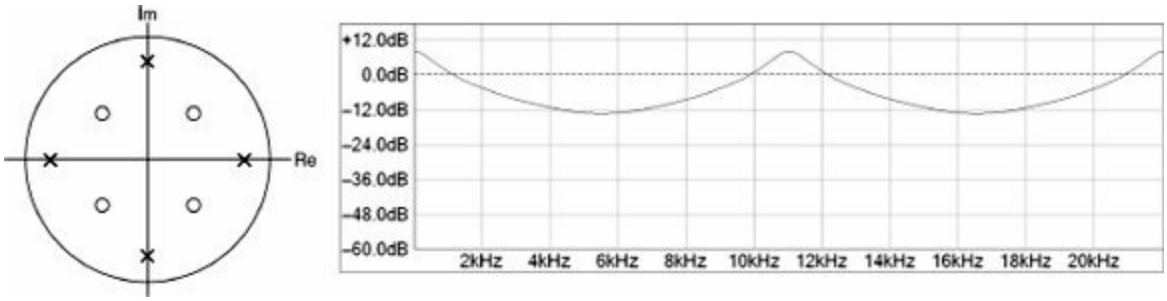


Figure 7.19: The z-plane pole/zero plot and resulting frequency response for $D = 4$ samples, 75% feedback.

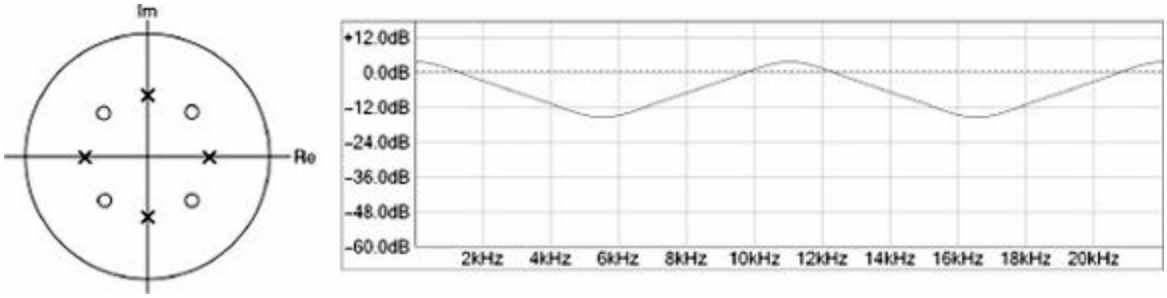


Figure 7.20: The z-plane pole/zero plot and resulting frequency response for $D = 4$ samples, 50% feedback.

$$H(z) = \frac{1 + (1-fb)z^{-D}}{1-fbz^{-D}}$$
$$H(z)|_{fb=-1} = \frac{1+1.5z^{-D}}{1+0.5z^{-D}} \qquad (7.20)$$

If you look at Equation 7.20, you can figure out that the pole frequencies are going to lie at the zero frequencies (notice the signs of the coefficients). The zeros will be at a radius of 1.5, while the poles will be at 0.5. A frequency lying on the unit circle will be under the influence of all four poles and zeros.

For the four-sample delay, a feedback value of –62% will make the frequency response perfectly flat, but with –3 dB of attenuation seen in Figure 7.21. The poles will have radii of 0.62 with the zeros at radii of 1.38. This means you can create a delay that has no comb/inverse comb filtering, but only at this particular value. Other negative feedback values will give varying degrees of cancellation. In practice, the poles will dominate and small peaks can appear at high inverted feedback values. In the

time domain, the echoes will alternate between positive and negative values each time they are flipped in the feedback path, shown in Figure 7.22.

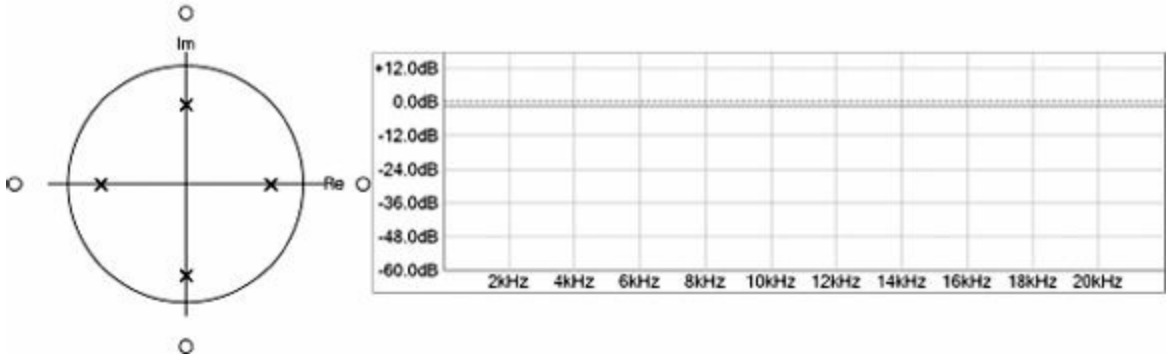


Figure 7.21: At –62% feedback with 4 samples of delay, the frequency response becomes flat but slightly attenuated.

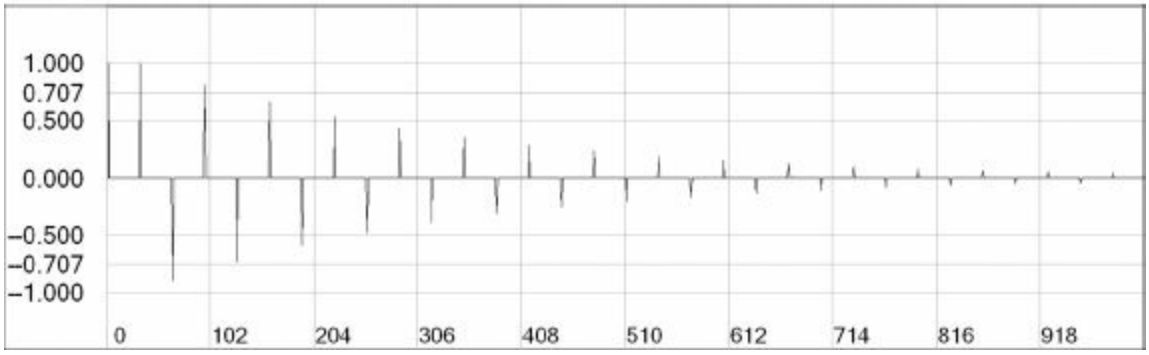


Figure 7.22: The effect of inverted feedback on the impulse response; feedback is –90% here.

7.3 Design a DDL Module Plug-In

In the previous projects, it was easy enough to simply declare left and right delay elements and coefficients for our simple filters. However, as the filters become more complex, this becomes more tedious and is also bad coding practice since we have replicated some code. For educational purposes, it is better to split the code out at first, but now it's time to think modularly. More complicated delay effects like the stereo cross feedback or LCR (left center right) delay will be easier to implement if we have a flexible delay module to work with. In this project, you will create a DDL module that you can use in many other projects. It is going to be a slight modification on the above DDL + wet/dry control. Here are the specifications:

- Implements an *n*-sample delay line, user controllable, up to 2 seconds of delay.
- Delay is given in milliseconds.
- Has feedback capable of –100% to +100% operation.
- Has wet/dry mix ratio control; 50/50 is an equal mix while 100/0 is full wet (delay only).
- Allows the feedback sample to be taken from the delay line, or supplied “outside” the module; the reason for this will become evident later.
- Allows access to the current feedback output sample. This allows you to insert other effects into the feedback path by using the switch to choose feedback_in; it also allows for cross coupling the feedback paths of stereo modules.

The block diagram is given in Figure 7.23.

For this design, let's implement a mono-only version. We can then make stereo versions by making two of our member variables the module object. That's right—you can use one plug-in inside another by making it a member object. First, let's get started on the DDL module. Note: For the initial DDL modules, don't worry about the feedback path switching. We need to get the delay debugged and running first.

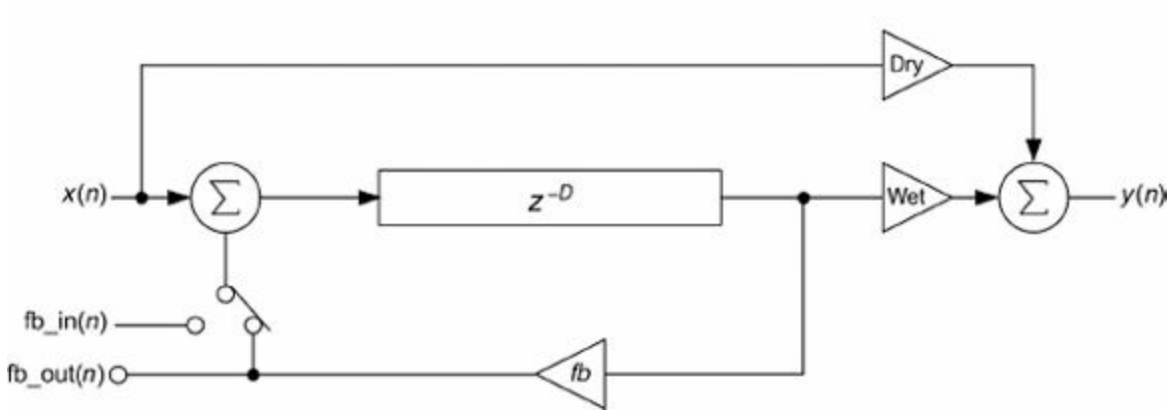


Figure 7.23: Our more flexible DDL module; a switch allows the user to choose where the feedback sample is taken. Here it is using the normal output feedback path. The feedback output is available for use at the fb_out pin.

7.3.1 Project: DDLModule

By now, you should be getting good at RackAFX programming, so we can move more quickly through the process. Create the project and add the sliders.

7.3.2 DDLModule GUI

Here is the final graphical user interface (GUI) for the DDL module in Figure 7.24. You can use my variable names or make up your own. You will need the controls shown in Table 7.1. We do not need a switch for the feedback option on the UI; it will only be needed by the super plug-in that includes this module as a member object.

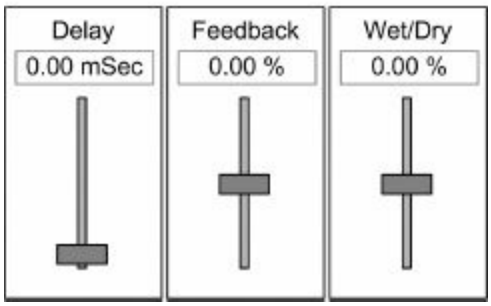


Figure 7.24: The DDL Module GUI. Table 7.1: GUI controls for the DDL module

Table 7.1: GUI controls for the DDL module

Slider Property	Value
Control Name	Delay
Units	mSec
Variable Type	float
Variable Name	m_fDelay_ms
Low Limit	0
High Limit	2000
Initial Value	0

Slider Property	Value
Control Name	Feedback
Units	%
Variable Type	float
Variable Name	m_f_Feedback_pct
Low Limit	–100
High Limit	100
Initial Value	0

Slider Property	Value
Control Name	Wet/dry
Units	%
Variable Type	float
Variable Name	m_f_WetLevel_pct
Low Limit	0
High Limit	100
Initial Value	50

7.3.3 DDLModule.h File

In the .h file, add the cooked variables, m_fDelayInSamples, m_fFeedback, and m_fWetLevel:

```
// Add your code here: ------------------------------------------------------ //
float m_fDelayInSamples;
float m_fFeedback;
float m_fWetLevel;
// END OF USER CODE --------------------------------------------------------- //


// ADDED BY RACKAFX -- DO NOT EDIT THIS CODE!!! ------------------------------ //
//   **--0x07FD--**

float m_fDelay_ms;
float m_fFeedback_pct;
float m_fWetLevel_pct;
// **--0x1A7F--**
// ------------------------------------------------------------------------- //
```

Note: I named the RackAFX controls with _ms and _pct appended to remind me that these values need to be cooked to be used.

Note: The delay time cooked variable is a float, m_fDelayInSamples rather than an integer number of samples. This is because we will allow *fractional delay* in a future version of the module. For now, though, we will treat it as an integer.

7.3.4 DDLModule .cpp File

Constructor

• Initialize variables.

```
CDDLModule::CDDLModule()
{
  <SNIP SNIP SNIP>
  // Finish initializations here
  m_fDelayInSamples = 0;
  m_fFeedback = 0;
  m_fDelay_ms = 0;
  m_fFeedback_pct = 0;
  m_fWetLevel = 0;

}
```

The formula for figuring out the delay time in samples from the delay time in milliseconds is Equation 7.21.

$$\text{Samples delay} = (D \text{ mSec})\left[\frac{\text{sample rate}}{1000}\right] \tag{7.21}$$

Cooking the feedback value is easy—just divide by 100 to convert the percent to a raw multiplier. The same thing is true for the wet level value. In this project, we'll introduce the concept of a cooking function to handle the work. Because we are planning on using this plug-in as a module for future development, it will be a good idea. From this point on, you should get in the habit of making a cooking function. In this case, we will keep it simple and have the cooking function recalculate all the plug-in's variables regardless of which ones actually change. This is an effort to educate first. You

can always go back and streamline your functions and code after the plug-in has been tested and is functioning. First, declare the cooking function in the .h file:

```
// Add your code here: ------------------------------------------------------ //
float m_fDelayInSamples;
float m_fFeedback;
float m_fWetLevel;

void cookVariables();
// END OF USER CODE --------------------------------------------------------- //
```

Write the function:

```
// function to cook the variables
void CDDLModule::cookVariables()
{
  m_fFeedback = m_fFeedback_pct/100.0;
  m_fWetLevel = m_fWetLevel_pct/100.0;
  m_fDelayInSamples = m_fDelay_ms*((float)m_nSampleRate/1000.0);
}
```

- The end of the constructor()
- prepareForPlay()
- userInterfaceChange()

Constructor

```
CDDLModule::CDDLModule()
{
	<SNIP SNIP SNIP>

	// Finish initializations here
	<SNIP SNIP SNIP>
	m_fFeedback_pct = 0;
	m_fWetLevel = 0;
	cookVariables();
}
```

prepareForPlay()

```
bool __stdcall CDDLModule::prepareForPlay()
{
	// cook
	cookVariables();

	return true;
}
```

Notice the big change made here—rather than a switch/case statement, we just go ahead and cook all the data.

userInterfaceChange()

```
bool __stdcall CDDLModule::userInterfaceChange(int nControlIndex)
{
	// cook
	cookVariables();

	return true;
}
```

Now that the UI details are taken care of, we can get to the business of declaring the delay line, initializing it, and finally implementing the effect.

7.3.5 Declare and Initialize the Delay Line Components

For a delay line, you will need the following variables:

- A float* which points to a buffer of samples
- An integer read index
- An integer write index
- An integer that is the size of the buffer in samples

Add them to your .h file:

```
// Add your code here: ------------------------------------------------------ //
float m_fDelayInSamples;
float m_fFeedback;
float m_fWetLevel;

float* m_pBuffer;
int m_nReadIndex;
int m_nWriteIndex;
int m_nBufferSize;
// END OF USER CODE --------------------------------------------------------- //
```

The delay line will be created dynamically. It will be destroyed in the destructor. The problem is that we don't yet know what the sample rate will be; we won't know that until the user loads a new file and begins playing it. Just before RackAFX calls your prepareForPlay() function, it sets the sample rate on your plug-in's m_nSampleRate variable. Therefore, we will have to dynamically create and flush out the buffer each time prepareForPlay() is called. In the constructor, we set the m_pBuffer to NULL as a flag to know that it is uninitialized, as well as zero the buffer size and read and write index values.

Constructor

```
CDDLModule::CDDLModule()
{
<SNIP SNIP SNIP>
        m_fFeedback_pct = 0;
        m_fWetLevel = 0;

        // reset
        m_nReadIndex = 0;
        m_nWriteIndex = 0;

        // no buffer yet because we don't have a sample rate yet
        m_pBuffer = NULL;
        m_nBufferSize = 0;

        // cook
        cookVariables();
}
```

prepareForPlay()

- Create the buffer now that we know the sample rate.

```
bool __stdcall CDDLModule::prepareForPlay()
{
        // setup our delay line
        m_nBufferSize = 2*m_nSampleRate;     // 2 seconds delay @ fs

        // delete it if it exists
        if(m_pBuffer)
                delete [] m_pBuffer;

        // create the new buffer
        m_pBuffer = new float[m_nBufferSize];

        return true;
}
```

Destructor

- Delete the buffer.

```
CDDLModule::~CDDLModule(void)
{
        // delete buffer if it exists
        if(m_pBuffer)
                delete [] m_pBuffer;
}
```

To initialize the buffer with 0.0, use the the memset function. That memset() flushes the buffer of data, and we need to do this each time prepareForPlay() is called so we don't play out old data at the onset. We are going to be flushing and resetting the buffer in several places in code, so it is also a good thing to make into a function.

7.3.6 DDLModule.h File

```
// Add your code here: ------------------------------------------------------ //
float m_fDelayInSamples;
float m_fFeedback;
float m_fWetLevel;

void cookVariables();
void resetDelay();
// END OF USER CODE --------------------------------------------------------- //
```

7.3.7 DDLModule.cpp File

```
// function to flush buffer and set Write pointer back to top
// read pointer will be calculated based on write pointer location
void CDDLModule::resetDelay()
{
	// flush buffer
	if(m_pBuffer)
		memset(m_pBuffer, 0, m_nBufferSize*sizeof(float));

	// init read/write indices
	m_nWriteIndex = 0; // reset the Write index to top
	m_nReadIndex = 0; // reset the Read index to top
}
```

You can also modify the cooking function to add the code for updating the read index; set the read and write indices depending on the amount of delay time, so you can use a modified version of the code from Section 7.1. Note: The delay in samples is cast to an integer using the casting method (int).

```
// function to cook the variables
void CDDLModule::cookVariables()
{
	m_fFeedback = m_fFeedback_pct/100.0;
	m_fWetLevel = m_fWetLevel_pct/100.0;
	m_fDelayInSamples = m_fDelay_ms*((float)m_nSampleRate /1000.0);

	// subtract to make read index
	m_nReadIndex = m_nWriteIndex - (int)m_fDelayInSamples; // cast as int!

	// check and wrap BACKWARDS if the index is negative
	if (m_nReadIndex < 0)
		m_nReadIndex += m_nBufferSize;	// amount of wrap is Read + Length
}
```

prepareForPlay()

- Reset the delay in prepareForPlay() after creating the buffer and before cookVariables().

```
bool __stdcall CDDLModule::prepareForPlay()
{
	// setup our delay line
	m_nBufferSize = 2*m_nSampleRate;		// 2 seconds delay @ fs

	// delete it if it exists
	if(m_pBuffer)
		delete [] m_pBuffer;

	// create the new buffer
	m_pBuffer = new float[m_nBufferSize];

	// reset
	resetDelay();

	// then cook
	cookVariables();

	return true;
}
```

processAudioFrame()

- The flowchart for the processAudioFrame() function is shown in Figure 7.25.

Note: We have one minor detail to deal with, and this is going to happen when we use the delay line in a read-then-write fashion. If the user has chosen 0.00 mSec of delay, then the write pointer and read pointer will be the same. This also occurs if the user selects the *maximum* delay value since we want to read the oldest sample before writing it. So, we need to make a check to see if there is no delay at all and deal with it accordingly.

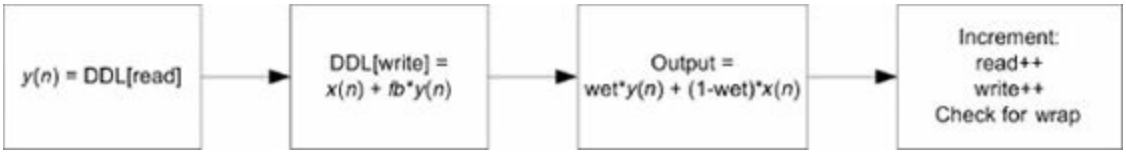


Figure 7.25: Flowchart for processAudioFrame().

```
bool __stdcall CDDLModule::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
UINT uNumInputChannels, UINT uNumOutputChannels)
{
	// SYNC CODE: DO NOT REMOVE - DO NOT PLACE CODE BEFORE IT
	WaitForUIChangeDone();
	setProcessAudioDone(false);
	// END SYNC CODE

	// Do LEFT (MONO) Channel
	// Read the Input
	float xn = pInputBuffer[0];

	// Read the output of the delay at m_nReadIndex
	float yn = m_pBuffer[m_nReadIndex];

	// if zero delay, just pass the input to output
	if(m_fDelayInSamples == 0)
		yn = xn;

	// write the input to the delay
	m_pBuffer[m_nWriteIndex] = xn + m_fFeedback*yn;

	// create the wet/dry mix and write to the output buffer
	// dry = 1 - wet
	pOutputBuffer[0] = m_fWetLevel*yn + (1.0 - m_fWetLevel)*xn;


	// incremnent the pointers and wrap if necessary
	m_nWriteIndex++;
	if(m_nWriteIndex >= m_nBufferSize)
		m_nWriteIndex = 0;

	m_nReadIndex++;
	if(m_nReadIndex >= m_nBufferSize)
		m_nReadIndex = 0;

	// Mono-In, Stereo-Out (AUX Effect)
	if(uNumInputChannels == 1 && uNumOutputChannels == 2)
		pOutputBuffer[1] = pOutputBuffer[0]; // copy MONO!

	// DDL Module is MONO - just do a copy here too
	// Stereo-In, Stereo-Out (INSERT Effect)
	if(uNumInputChannels == 2 && uNumOutputChannels == 2)
		pOutputBuffer[1] = pOutputBuffer[0];  // copy MONO!

	// SYNC CODE: DO NOT REMOVE
	setProcessAudioDone();

	return true;
}
```

Build and test the module in RackAFX to make sure it works. You should get a generic delay effect with up to two seconds of delay. We only have two items to take care of to complete this first version of the module. The first is easy: we need to provide the user of the module with a way to choose external feedback and have access to the feedback variable. The second is more difficult: we need to be able to handle fractional delay.

In order for someone to use this module inside another plug-in and get access to the feedback path, we need to provide some functions that will only be called by a plug-in parent. We also need to modify the processAudioFrame() function to use an externally supplied feedback sample.

7.4 Modifying the Module to Be Used by a Parent Plug-In

It's actually pretty easy to modify this module to work as an internal module for a plug-in. We need to provide a variable for the feedback sample and allow the user to set this value. We also need to provide a way to allow the user to get the current feedback value. Finally, we need a switch to allow the user to select the external feedback mode. That switch will be in the form of a boolean flag. After we get the variables set up, we can modify the processAudioFrame() function to use the external feedback sample.

7.4.1 DDLModule.h File

Declare the following new variables:

```
bool m_bUseExternalFeedback; // flag for enabling/disabling
float m_fFeedbackIn;         // the user supplied feedback sample value
```

Also, declare and implement the get/set functions. They are so short you can just put them in the .h file rather than implementing them in the .cpp file.

```
// Add your code here: ------------------------------------------------------ //
float m_fDelayInSamples;
float m_fFeedback;
float m_fWetLevel;

void cookVariables();
void resetDelayLine();

float* m_pBuffer;
int m_nReadIndex;
int m_nWriteIndex;
int m_nBufferSize;

bool m_bUseExternalFeedback; // flag for enabling/disabling
float m_fFeedbackIn;         // the user supplied feedback sample value

// current FB is fb*output
float getCurrentFeedbackOutput(){return m_fFeedback*m_pBuffer[m_nReadIndex];}

// set the feedback sample
void setCurrentFeedbackInput(float f){m_fFeedbackIn = f;}

// enable/disable external FB source
void setUsesExternalFeedback(bool b){m_bUseExternalFeedback = false;}
// END OF USER CODE ---------------------------------------------------------- //
```

The current feedback sample output is found at m_pBuffer[m_nReadIndex] and multiplied by the feedback coefficient as per the block diagram. By allowing the user to get and set the feedback, we allow them to break into the loop and insert other effects, or use the feedback output for some other purpose altogether.

7.4.2 DDLModule.cpp File

processAudioFrame()

- Modify the function to allow the use of externally supplied feedback samples:

```
bool __stdcall CDDLModule::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
UINT uNumInputChannels, UINT uNumOutputChannels)
{
	// Do LEFT (MONO) Channel
	// Read the Input
	float xn = pInputBuffer[0];

	// Read the output of the delay at m_nReadIndex
	float yn = m_pBuffer[m_nReadIndex];

	// if zero delay, just pass the input to output
	if(m_fDelayInSamples == 0)
		yn = xn;

	// write the input to the delay
	if(!m_bUseExternalFeedback)
		m_pBuffer[m_nWriteIndex] = xn + m_fFeedback*yn; // normal
	else
		m_pBuffer[m_nWriteIndex] = xn + m_fFeedbackIn; // external feedback

	// create the wet/dry mix and write to the output buffer
	// dry = 1 - wet
	pOutputBuffer[0] = m_fWetLevel*yn + (1.0 - m_fWetLevel)*xn;

	etc…
```

You can see that the change required is very minor—just a switch to change the feedback sample value. Of course if the user enables this option, then they are responsible for placing meaningful data in the fFeedbackIn variable.

Rebuild and test the code to make sure it still works properly. In the next section, we will use the module to make two different plug-ins:

1. Stereo digital delay
2. Stereo crossed-feedback delay

7.5 Modifying the Module to Implement Fractional Delay

Before we work on the bigger projects, we need to take care of the problem of fractional delay. We would like the user to be able to get any amount of delay they want. By implementing only sample-based delay, we are limiting the user to choosing delays that are multiples of the sample period, about 23 uSec. You might think that is enough accuracy; however, there are several instances where this won't be enough. The first is the case of a delay that is beats-per-minute (BPM) synchronized (e.g., delay is set to a multiple of the song's BPM to create synchronized echoes or percussion effects). The second case is that of modulated delays like chorus and flanging, which require smooth modulations from one delay time to another. Linear interpolation will provide acceptable results.

Suppose our delay is in a state where we have calculated our delay position to be at sample location 23.7183 samples. We need to find the value of the data at the location 0.7183 between sample 23 and sample 24. In Figure 7.26 you can see a graphic representation of the interpolation method. Since it's linear interpolation, a line is drawn between the adjacent samples and the interpolated value is found on the line at 0.7183 the distance between the two samples.

In polynomial interpolation such as LaGrange interpolation, a curve is drawn between the points (or a series of points), and then the interpolated value is found on that curve. There are several ways to

implement the linear interpolation but the easiest method is to treat it like a DSP filter. Another way of thinking about interpolation is as a weighted sum. For example, if the interpolation point is 0.5 between samples 1 and 2, then the interpolated value is made up of 50% of sample 1 plus 50% of sample 2. In the above case, our interpolated distance is 0.7183, so we can view the output as

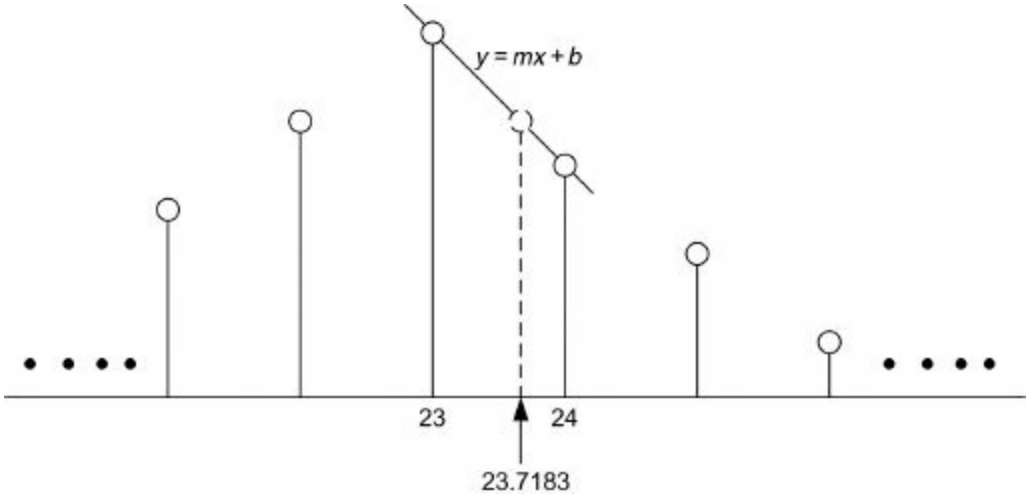


Figure 7.26: Linear interpolation of sample values.

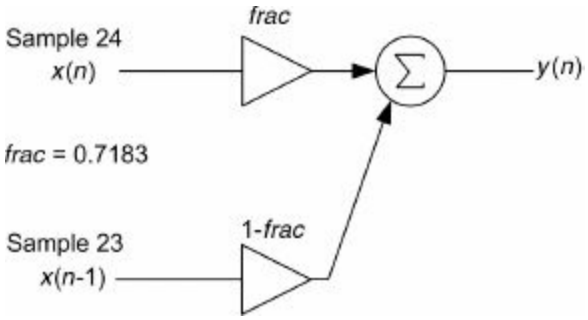


Figure 7.27: Linear interpolation as a kind of feed-forward filter. The z^{-1} element is removed since we do not know if the interpolated samples will always be exactly one sample apart. For example, what if the next fractional delay sample is at location 56.2394?

interp_output = (0.7183)(Sample 2) + (0.2817)(Sample 1)

Here is a linear interpolation function you can use; it is already declared in your pluginconstants.h file:

```
float dLinTerp (float x1, float x2, float y1, float y2, float x);
```

You give it a pair of data points ($x1$,$y1$) and ($x2$,$y2$), plus a distance between them on the x-axis (x), and it returns the interpolated value using the weighted sum method. The first part of the code checks for a potential divide by zero fault that technically should not happen. You should also be aware that linear interpolation is a form of feed-forward filtering. The block diagram of the above function would look like Figure 7.27.

Thus, in the processAudioFrame(), we need to find the two sample values that our actual delay time needs, then do the interpolation. It is pretty straightforward except the situation where we are interpolating across the wrap boundary (from the last sample in the buffer to the first one). Suppose the user enters a delay time that corresponds to 2.4 samples of delay. In the cookVariables() function,

we locate the read index to be two samples before the write pointer because we cast the value to an integer, stripping out the fractional part. The actual delay we really want is 0.4 samples between the current sample and the one before it. In other words, $x\ (n - 2.4)$ rather than $x\ (n - 2)$. You can see that we need a sample that is between $x\ (n - 2)$ and $x\ (n - 3)$; in Figure 7.28 it happens to be a distance of 0.4 between them.

We need to modify our code as follows:

- We are already calculating and updating the first sample that we need to interpolate; it is located with the index value at m_nReadIndex, so there is nothing to change in the cooking function.
- In the processAudioFrame() function, we need to interpolate between our current read location m_nReadIndex and the location just *before* it in the buffer, m_nReadIndex-1.
- Since we are only focused on a single pair of samples at any time and we know they are one sample apart, we can use the values 0 and 1 for the interpolation function as $x1 = 0$, $x2 = 1$; then we interpolate the fractional distance between them. You can use m_nRead-Index but you will get in trouble when you are interpolating across the wrap boundary of the buffer.
- We will need to check for a wrap condition backward if m_nReadIndex–1 takes us outside the top of the buffer.

We can get the fractional value from our m_fDelayInSamples in several ways; here is one of them:

```
float fFracDelay = m_fDelayInSamples – (int)m_fDelayInSamples
```

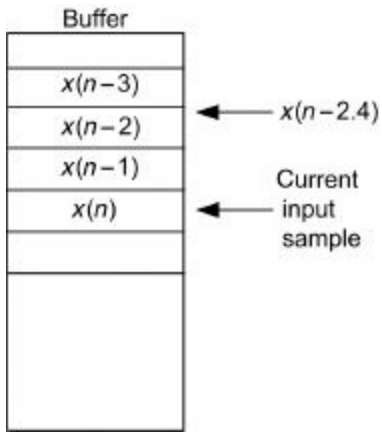


Figure 7.28: Fractional interpolation.

It really only comes down to locating the sample 1 behind our current read index, then using the linear interpolation code to get the fractional value. There are two extreme conditions to consider:

- At the maximum delay time, the read and write indices will be equal and the fractional part of the delay in samples will be 0 so no interpolation will occur—this is OK.
- If the delay in samples is less than 1, the read and write indices will also be equal, but this will be a problem. In this case, we need to interpolate between the current $x(n)$ and the sample at $x(n - 1)$, one sample behind the read location. Branching will be necessary to catch this event and handle it.

7.5.1 DDLModule.cpp File

processAudioFrame()

- Modify the code to do the interpolation.

```
bool __stdcall CDDLModule::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
UINT uNumInputChannels, UINT uNumOutputChannels)
{
	// Do LEFT (MONO) Channel
	// Read the Input
	float xn = pInputBuffer[0];

	// Read the output of the delay at m_nReadIndex
	float yn = m_pBuffer[m_nReadIndex];

	// if delay < 1 sample, interpolate between input x(n) and x(n-1)
	if(m_nReadIndex == m_nWriteIndex && m_fDelayInSamples < 1.00)
	{
		// interpolate x(n) with x(n-1), set yn = xn
		yn = xn;
	}
	// Read the location ONE BEHIND yn at y(n-1)
	int nReadIndex_1 = m_nReadIndex - 1;
	if(nReadIndex_1 < 0)
		nReadIndex_1 = m_nBufferSize-1; // m_nBufferSize-1 is last location
	// get y(n-1)
	float yn_1 = m_pBuffer[nReadIndex_1];

	// interpolate: (0, yn) and (1, yn_1) by the amount fracDelay
	float fFracDelay = m_fDelayInSamples - (int)m_fDelayInSamples;

	// linerp: x1, x2, y1, y2, x
	float fInterp = dLinTerp(0, 1, yn, yn_1, fFracDelay); // interp frac between them

	// if zero delay, just pass the input to output
	if(m_fDelayInSamples == 0)
		yn = xn;
	else
		yn = fInterp;

	// write the intput to the delay
	if(!m_bUseExternalFeedback)
		m_pBuffer[m_nWriteIndex] = xn + m_fFeedback*yn; // normfInterpal
	else
		m_pBuffer[m_nWriteIndex] = xn + m_fFeedbackIn; // external feedback
		    sample
}
```

Now, build and test the module. Try a variety of delay settings. If you hear a repetitive click in your output that is exactly the same rate as the delay time, then there is something wrong during the interpolation across the wrap boundary of buffer[0] to buffer[size–1]. Only advance to the next section when you have this project fully debugged and functional. We will be including it in the next project, a stereo digital delay.

7.6 Design a Stereo Digital Delay Plug-In

In this project, we use two DDL modules in one parent plug-in. RackAFX makes it easy to do this by allowing you to add other plug-in components (.h and .cpp files) into a new project. It will automatically #include the components too. However, if you use external modules or other files you might need to manually #include these. In Figure 7.29 you can see that we now have two DDL

modules declared as member objects of the new plug-in. The plug-in implements its own interface of three sliders, which we use to control our modules.

7.6.1 Project: StereoDelay

Create a project named "StereoDelay." When you create the project, you have the option of including other modules in your code, seen in Figure 7.30. RackAFX finds all of the existing RackAFX projects in the default directory you supply and lists them here. You use the Add button to move them into your project. If you have a project located in another directory that is not the default, you will need to move the files on your own (copy them to the new project directory and #include them in the new <project>.h file and add them into the compiler). RackAFX will automatically copy them and #include whichever modules you choose. In this case, choose the DDL module.

When you use a plug-in as a module for another parent plug-in you must create and implement a new UI. The child plug-in objects will *not* expose their sliders to RackAFX, but you can manipulate the UI variables. All other aspects of the child objects work as expected. In this plug-in, we will implement another UI to control the modules. See Appendix A.2 for advanced control of the UI variables.

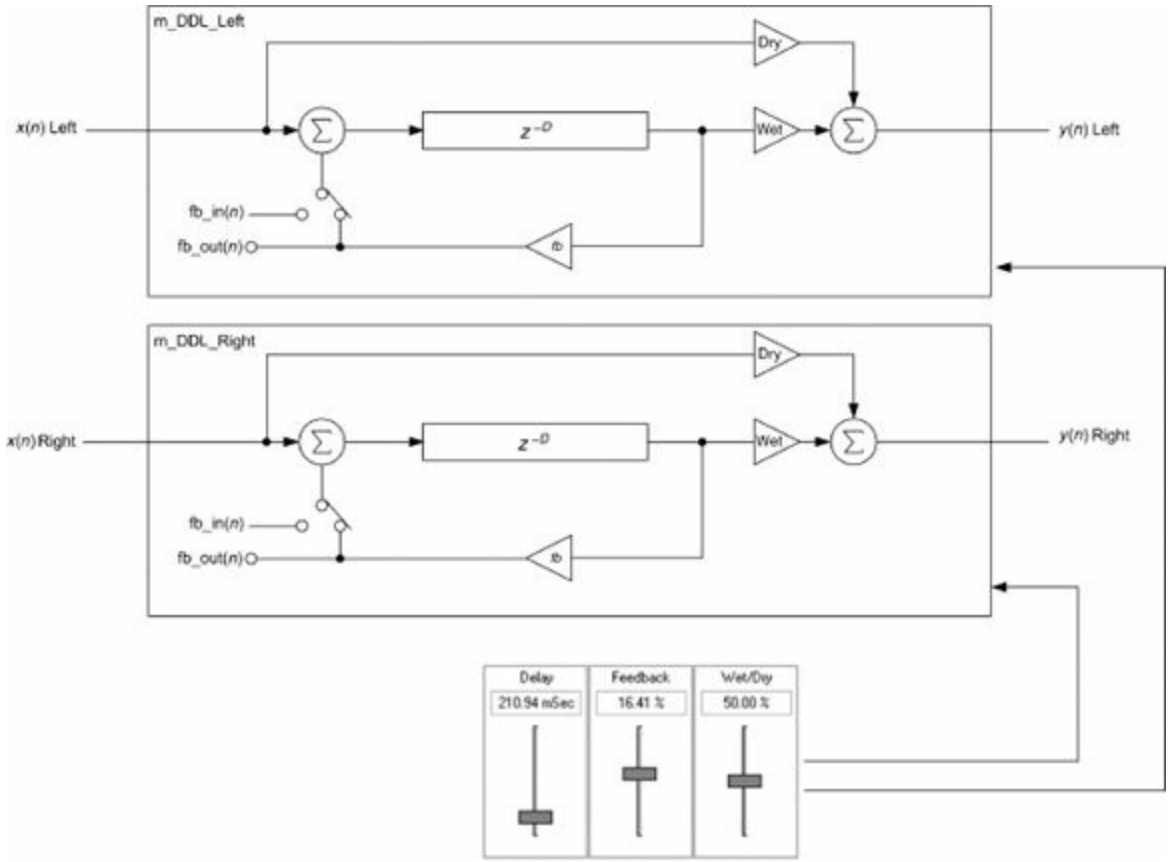


Figure 7.29: Block diagram of our stereo delay.



Figure 7.30: Adding existing modules can be done programmatically through RackAFX.

You can see in Figure 7.30 that I added "DDLModule.h" to the project. After completing the new project dialog, check your new <plugin>.h file:

```
#pragma once

#include "pluginconstants.h"

#include "DDLModule.h"
#include "plugin.h"

// abstract base class for DSP filters
class CStereoDelay : public CPlugIn
{
public:  // Plug-In API Functions
         //
         // 1. One Time Initialization
         CStereoDelay();

etc …
```

7.6.2 StereoDelay GUI

Your GUI will look like that in Figure 7.29 and you use the same setup from the DDLModule in Table 7.1.

7.6.3 StereoDelay.h File

In the .h file, declare two member objects of type CDDLModule. Also, add a function called setDelayVariables() to transfer our global delay variables to the member objects and optionally have the member objects cook the data:

```
// Add your code here: ----------------------------------------------------- //
CDDLModule m_DDL_Left;
CDDLModule m_DDL_Right;

// function to pass our variables to member delays
void setDelayVariables(bool bCook);
// END OF USER CODE ------------------------------------------------------- //
```

Our DDLModules are just C++ objects, so you can treat them as such and call their member functions and set their member variables. Before implementing the function above, go ahead and add your UI sliders, exactly the same as before with the same variable names. You will have something like this:

```
// ADDED BY RACKAFX -- DO NOT EDIT THIS CODE!!! ------------------------------ //
// **--0x07FD--**

   float m_fDelay_ms;
   float m_fFeedback_pct;
   float m_fWetLevel_pct;

// **--0x1A7F--**
// ---------------------------------------------------------------------------- //
```

7.6.4 StereoDelay.cpp File

Write the setDelayVariables() function. The Boolean flag allows you to optionally call the cooking functions on the members:

```
void CStereoDelay::setDelayVariables(bool bCook)
{
       // forward our variables over to the member objects
       m_DDL_Left.m_fDelay_ms = m_fDelay_ms;
       m_DDL_Right.m_fDelay_ms = m_fDelay_ms;

       m_DDL_Left.m_fFeedback_pct = m_fFeedback_pct;
       m_DDL_Right.m_fFeedback_pct = m_fFeedback_pct;

       m_DDL_Left.m_fWetLevel_pct = m_fWetLevel_pct;
       m_DDL_Right.m_fWetLevel_pct = m_fWetLevel_pct;

       // cook, if desired
       if(bCook)
       {
              m_DDL_Left.cookVariables();
              m_DDL_Right.cookVariables();
       }
}
```

Constructor

- Initialize the delay variables and cook them; this version does not use the external feedback option, so set the bool accordingly.

```
CStereoDelay::CStereoDelay()
{
       <SNIP SNIP SNIP>

       // Finish initializations here
       m_DDL_Left.m_bUseExternalFeedback = false;
       m_DDL_Right.m_bUseExternalFeedback = false;

       // set and (true) cook the delays
       setDelayVariables(true);

}
```

prepareForPlay()

- Set the delay variables.
- Forward the calls to prepareForPlay() on the member objects, which will cook them.

The DDL module will handle flushing the buffers and so on.

```
bool __stdcall CStereoDelay::prepareForPlay()
{
       setDelayVariables(false);

       m_DDL_Left.prepareForPlay();
       m_DDL_Right.prepareForPlay();

     return true;
}
```

userInterfaceChange()

- Set the delay variables and cook them whenever one of our controls changes. This could be streamlined for better efficiency.

```
bool __stdcall CStereoDelay::userInterfaceChange(int nControlIndex)
{
       // set and cook the variables
       setDelayVariables(true);

       return true;
}
```

processAudioFrame()

- Forward the processAudioFrame() call to the member objects.
- Remember that we need to send it only one channel at a time and make sure it is the correct one. Note the “address of” operator (&) to point to the single memory location we pass it.

```
bool __stdcall CStereoDelay::processAudioFrame(float* pInputBuffer, float*
pOutputBuffer, UINT uNumInputChannels, UINT uNumOutputChannels)
{
       // Do LEFT (MONO) Channel; there is always at least one input/one output
       // forward call to sub-object: pInput, pOutput, 1 input ch, 1 output ch
       m_DDL_Left.processAudioFrame(&pInputBuffer[0], &pOutputBuffer[0], 1, 1);

       // Mono-In, Stereo-Out (AUX Effect)
       if(uNumInputChannels == 1 && uNumOutputChannels == 2)
              pOutputBuffer[1] = pOutputBuffer[0]; // just copy

       // Stereo-In, Stereo-Out (INSERT Effect)
       if(uNumInputChannels == 2 && uNumOutputChannels == 2)
              // forward call to sub-object pInput, pOutput, 1 input ch, 1 output ch
              m_DDL_Right.processAudioFrame(&pInputBuffer[1], &pOutputBuffer[1],1,1);
       return true;
}
```

Rebuild and test the project and you now have a stereo version of the previous project. Hopefully, you have a better idea of how powerful it can be to create modules that are combined, though it does take a bit of extra work on the module since you have to think ahead and implement functions or variables that are not required for standalone (simple) plug-in operation. Next, we'll exercise the module by converting this plug-in to a crossed feedback delay. We're going to do this in a specific way so we can later use an enumerated variable to switch between normal and crossed-feedback operation.

7.7 Design a Stereo Crossed-Feedback Delay Plug-In

A crossed-feedback delay is implemented by crossing the feedback paths of the two delay lines. We will add this functionality to the existing StereoDelay project. This is where we will use the external feedback option that we built into the module. In Figure 7.31 you can trace the feedback paths to see that they are crossed into the opposite delay line. You will be surprised at how easy it is to convert the delay into a crossed-feedback delay:

- Enable the external feedback function with the boolean flag.
- Use the getCurrentFeedbackOutput() and setCurrentFeedbackInput() functions to "cross the beams" of the feedback paths.

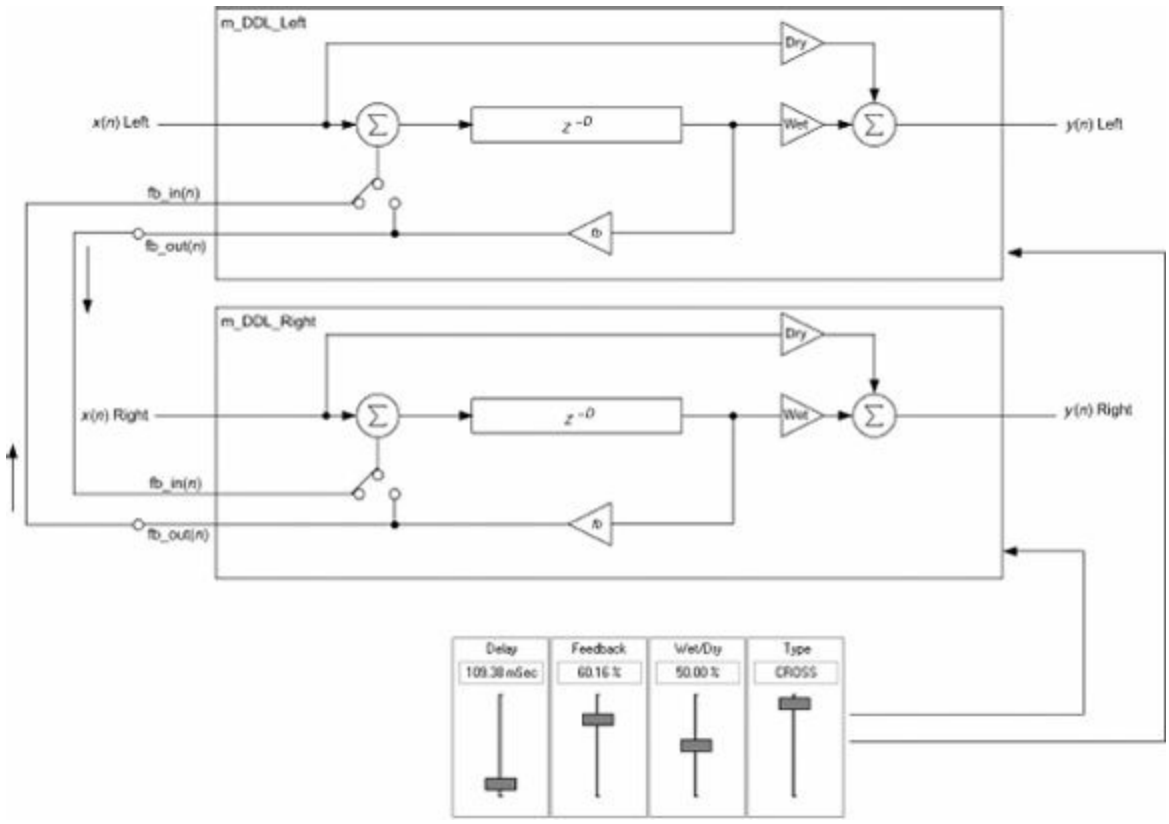


Figure 7.31: Block diagram of the crossed-feedback delay.

7.8 Enumerated Slider Variables

You can see from Figure 7.31 that there is a new slider control for the GUI to select between normal and crossed-feedback operation. You can take advantage of RackAFX's enumerated UINT variable to make a slider that selects between multiple enumerated values, like a multi-position switch. In our case, we only have two positions right now: normal and cross.

Right-click on a slider to bring up the properties dialog and add a new slider/variable combination, as shown in Figure 7.32. Choose "enum" for the data type and create a variable called m_uDelayType—the variable will be an unsigned integer type (UINT). The low, high, and initial value cells will no longer be editable; RackAFX will figure them out depending on what you type in the enumerated variable box. You type in the values separated by commas. The first value will be the default value.

In the box at the bottom, type in the strings which will be shown in the slider control (keep them

short); you can have as many enumerated variables as you want, but in this case, we only need NORM and CROSS. Go to your plug-in's .h file to see the new variables:

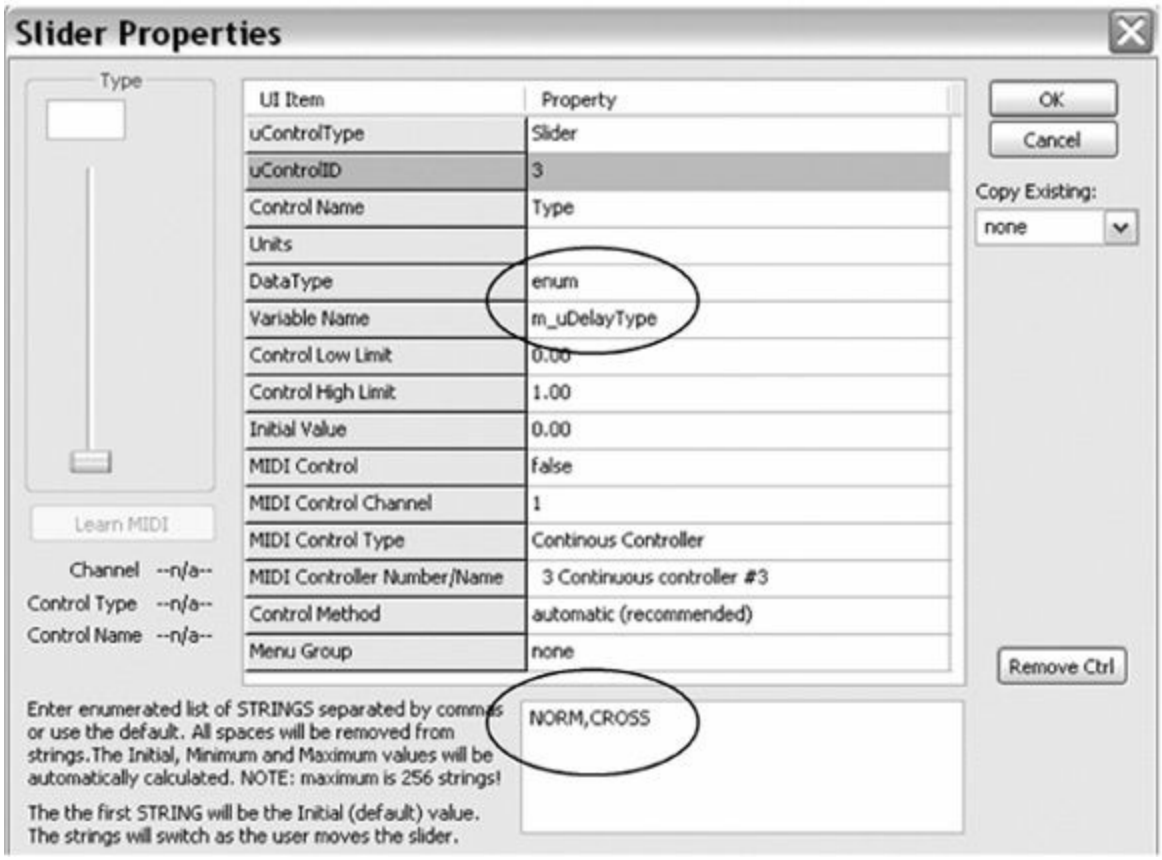


Figure 7.32: The enumerated UINT data type.

```
// ADDED BY RACKAFX -- DO NOT EDIT THIS CODE!!! ------------------------------ //
//**--0x07FD--**

float m_fDelay_ms;
float m_fFeedback_pct;
float m_fWetLevel_pct;
UINT m_uDelayType;
enum{NORM,CROSS};
// **--0x1A7F--**
// ------------------------------------------------------------------------ //
```

For enumerated variables, the first one in the list will be the default, with the slider at the bottom. The other strings will appear in sequence when you move the slider up. The UINT will automatically be updated. We can use this as a directly controlled variable so there's nothing to add in userInterfaceChange(). Make the edits discussed next.

7.8.1 Constructor

- Initialize the delay type to NORM.

```
CStereoDelay::CStereoDelay()
{
	<SNIP SNIP SNIP>

	// Finish initializations here
	m_DDL_Left.m_bUseExternalFeedback = false;
	m_DDL_Right.m_bUseExternalFeedback = false;

	// set and (true) cook the delays
	setDelayVariables(true);

	// init the delay type
	m_uDelayType = NORM;
}
```

7.8.2 PrepareForPlay()

Nothing to do; we don't care what mode we're in and don't want to reset the mode each time the user plays audio.

7.8.3 UserInterfaceChange()

Nothing to do; we are using this as a direct control variable.

7.8.4 ProcessAudioFrame()

- Use the enumerated variable in a switch/case statement to modify the feedback as required. For CROSS operation:
  - Set the external feedback flag to true.
  - Get the left feedback output and write it to the right feedback input.
  - Get the right feedback output and write it to the left feedback input.
- Call the processAudioFrame() functions on the DDL modules.

```
bool __stdcall CStereoDelay::processAudioFrame(float* pInputBuffer, float*
pOutputBuffer, UINT uNumInputChannels, UINT uNumOutputChannels)
{

	switch(m_uDelayType)
	{
	case CROSS:
		// CROSS FB DELAY --- NOTE: MUST HAVE STEREO FILE! ---------------- //
		m_DDL_Left.m_bUseExternalFeedback = true;
		m_DDL_Right.m_bUseExternalFeedback = true;

		// cross the Feedbacks
		m_DDL_Left.setCurrentFeedbackInput(m_DDL_Right.getCurrent
		FeedbackOutput());
		m_DDL_Right.setCurrentFeedbackInput(m_DDL_Left.getCurrent
		FeedbackOutput());
		break;

	case NORM: // clear the flags
		m_DDL_Left.m_bUseExternalFeedback = false;
		m_DDL_Right.m_bUseExternalFeedback = false;
		break;

	default: // deault is NORM operation; clear the flags
		m_DDL_Left.m_bUseExternalFeedback = false;
		m_DDL_Right.m_bUseExternalFeedback = false;
		break;
	}

	// SHARED CODE ------------------------------------------------------- //
	// Do LEFT (MONO) Channel; there is always at least one input/one output
	// forward call to sub-object: pInput, pOutput, 1 input ch, 1 output ch
	m_DDL_Left.processAudioFrame(&pInputBuffer[0], &pOutputBuffer[0], 1, 1);

	// Mono-In, Stereo-Out (AUX Effect)
	if(uNumInputChannels == 1 && uNumOutputChannels == 2)
		pOutputBuffer[1] = pOutputBuffer[0]; // just copy


	// Stereo-In, Stereo-Out (INSERT Effect)
	if(uNumInputChannels == 2 && uNumOutputChannels == 2)
		// forward call to sub-object pInput, pOutput, 1 input ch, 1 output ch
		m_DDL_Right.processAudioFrame(&pInputBuffer[1], &pOutputBuffer[1], 1, 1);

	return true;
}
```

Rebuild and test the plug-in; exercise the new delay type control as well.

Recapping some of the things you learned:

- Declaring, flushing, and manipulating delay lines as circular buffers.
- Implementing fractional delay times using interpolation.
- Using a plug-in as a child module for a parent plug-in.
- How to add an enumerated UINT variable for type selection or using a slider as a multi-pole switch.

7.9 More Delay Algorithms

Here are some ideas for a more complex DDL module and other delay plug-ins.

7.9.1 Advanced DDL Module

The more flexible module in Figure 7.33 allows the parent to have access to the input and pre-wet output of the delay line and the input and output of the feedback path. Another module (low-pass filter, for example) could be inserted here or it could be used for more crossed-delay modes.

7.9.2 Delay with LPF in Feedback Loop

Analog delays suffer a high frequency loss on each path through the delay line/feedback path. This can be modeled with a LPF in the feedback path, as shown in Figure 7.34. You can try first- or second-order filters with or without resonance for a variety of interesting delay effects. For other ideas, try changing the type (LPF, high-pass filter, band-pass filter, band-stop filter) or location (before or after the *fb* coefficient). Be careful with adding resonant or high-gain filters in the feedback loop. You could also inject some filtered white noise into the feedback path to simulate the noisy characteristics of typical analog delays.

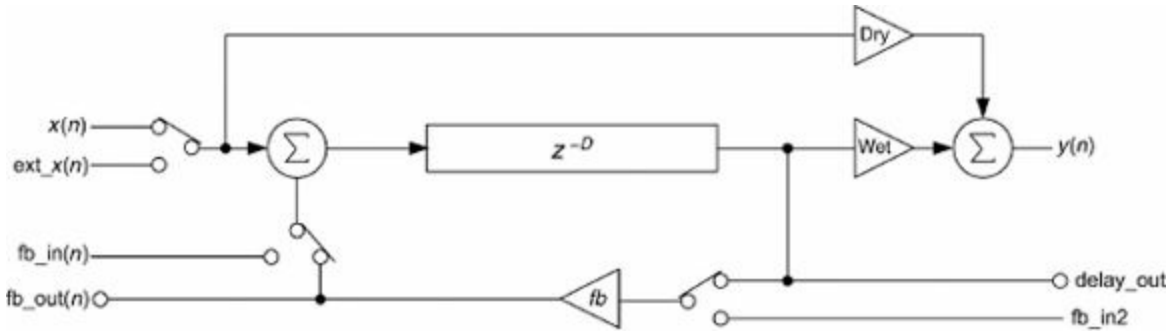


Figure 7.33: A DDL module with more input and feedback path options.

7.9.3 Multi-Tap Delay

The multi-tap delay line shown in Figure 7.35 produces four separate delays from one unit. Only the longest delay is fed back in this version. You create the multiple taps by simply declaring more read index values (m_nReadIndexTap_1, m_nReadIndexTap_2, etc..); the multiple taps are used and incremented exactly the same as the single tap case.

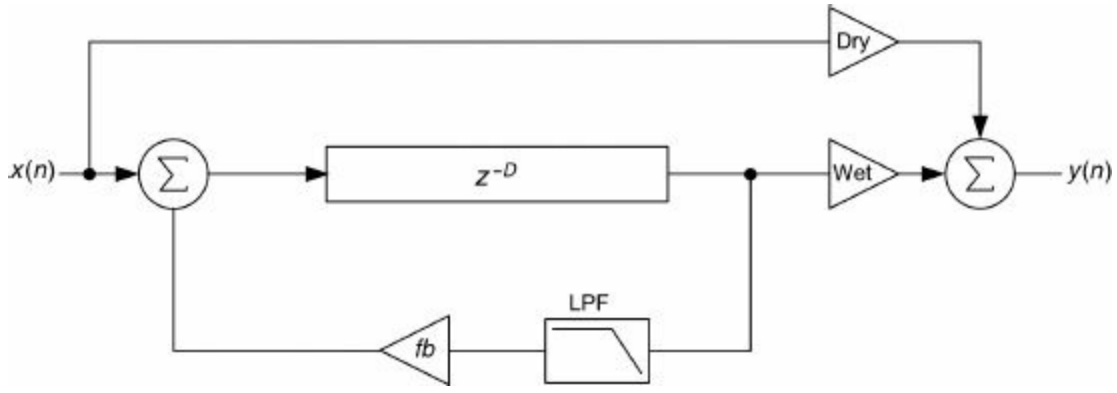


Figure 7.34: An analog delay modeled with an LPF in the feedback loop.

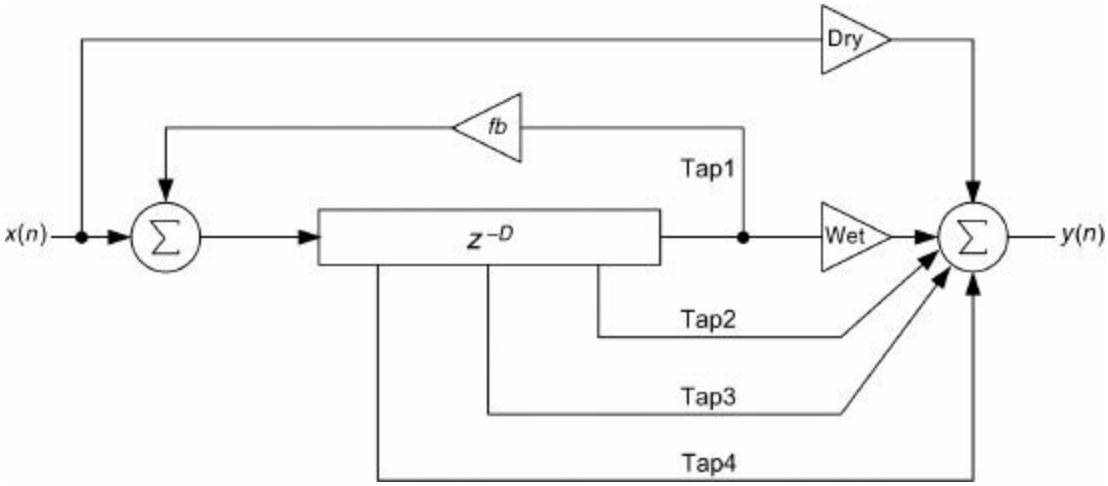


Figure 7.35: A four-tap multi-tap delay line.

You can also experiment with multiple feedback paths, filters, or setting your tap times using BPM and musical rhythms (e.g., delays on eighth and quarter notes).

7.9.4 Ping-Pong Delay

Figure 7.36 shows how the ping-pong delay cycles the delay back and forth between the left and right channels.

7.9.5 LCR Delay

The LCR delay in Figure 7.37 based on the Korg Triton® has both LPF and HPF in the feedback path (both are switchable or bypass-able) for multiple feedback tone shaping options.

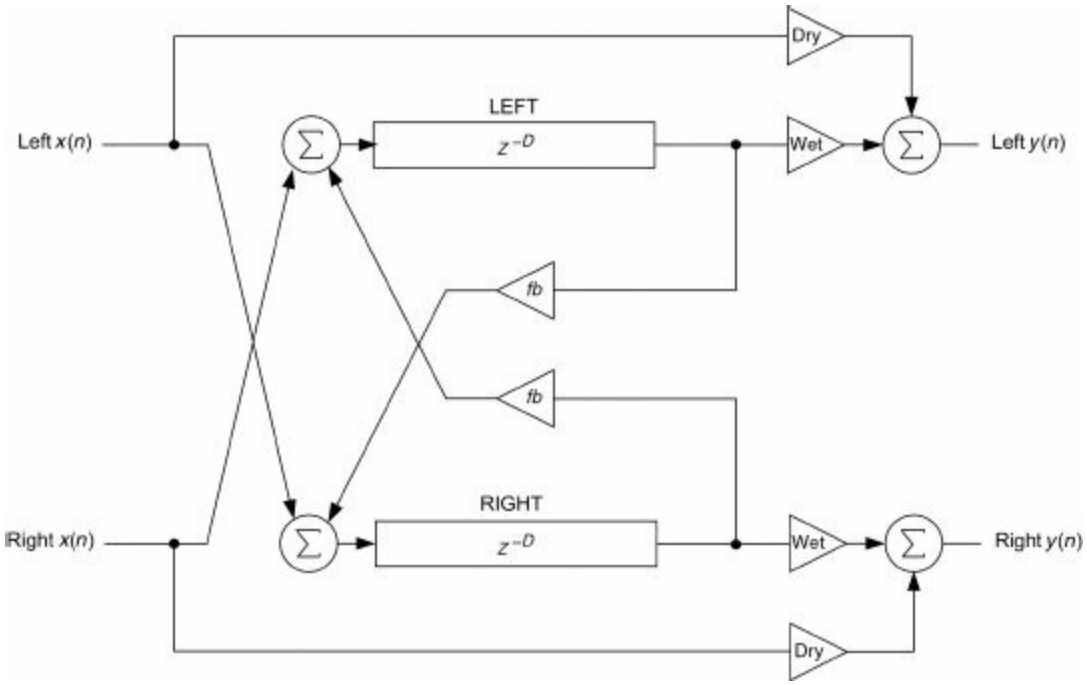


Figure 7.36: The ping-pong delay builds on the cross-feedback delay by crossing the inputs as well as the feedback paths to produce the back and forth ping-pong effect. You will probably want to design the advanced DDL module first and use its input, output, and feedback ports.

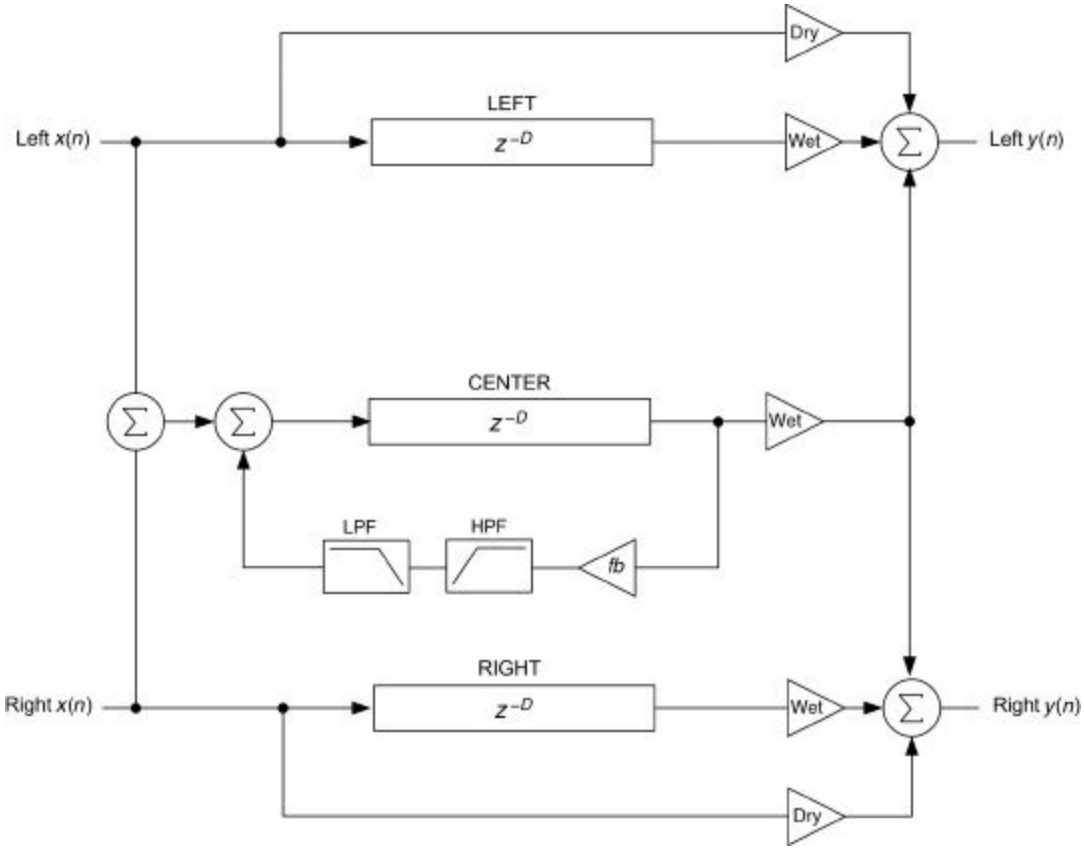


Figure 7.37: The LCR delay, based on the LCR delay in the Korg Triton®.

Bibliography

Coulter, D. 2000. *Digital Audio Processing*, Chapter 11. Lawrence, KS: R&D Books.
DSP56KFAM/AD. Schomberg, ON: Motorola, Inc.
Korg, Inc. 2000. *Triton-Rack Parameter Guide*, Tokyo: Korg, Inc.
Motorola, Inc. 1992. *DSP56000 Digital Signal Processor Family Manual*.
Roads, C. 1996. *The Computer Music Tutorial*, Chapter 3. Cambridge, MA: The MIT Press.

Audio Filter Designs: FIR Filters

Infinite impulse response (IIR) filters have several attractive properties:

- They only require a few delay elements and math operations.
- You can design them directly in the *z*-plane.
- You can use existing analog designs and convert them to digital with the Bilinear z-Transform (BZT); the fact that IIR topologies somewhat resemble the signal flow in analog filters emphasizes their relationship.
- You can get extreme resonance/gain by placing poles very near the unit circle.
- You can make filters, EQs, and so on with controls that link to the coefficients directly or indirectly for real-time manipulation of the plug-in.

Their main drawback is that they can become unstable and blow up, or oscillate. Their impulse responses (IRs) can be infinite. Finite impulse response (FIR) filters have all zeros and a finite IR. They are unconditionally stable so their designs can never blow up. However, they can put out all zeros, a constant value, a series of clicks or pulses, or other erroneous output, but they don't actually go unstable.

You can also make a linear phase filter with an FIR, just like the simple feed-forward (FF) topology you analyzed in Chapters 5 and 6; a linear phase filter is impossible to make with an IIR topology, although you can approximate it by adding phase compensation filters. However, the one thing that separates FIR filters from all other kinds, including analog counterpart filters, is that their coefficients a_0, a_1, …, a_N are the IR of the filter. You proved that when you manually pushed an impulse through the simple FF filter in Chapter 4, and then again when taking the *z* transform of the IR of the same filter in Chapter 5.

8.1 The IR Revisited: Convolution

In Chapter 1 you saw how the digitized signal was reconstructed into its analog version by filtering through an ideal low-pass filter (LPF). When the series of impulses is filtered, the resulting set of $\sin(x)/x$ pulses overlap with each other and their responses all add up linearly. The addition of all the smaller curves and damped oscillations reconstructs the inter-sample curves and damped fluctuations (Figure 8.1).

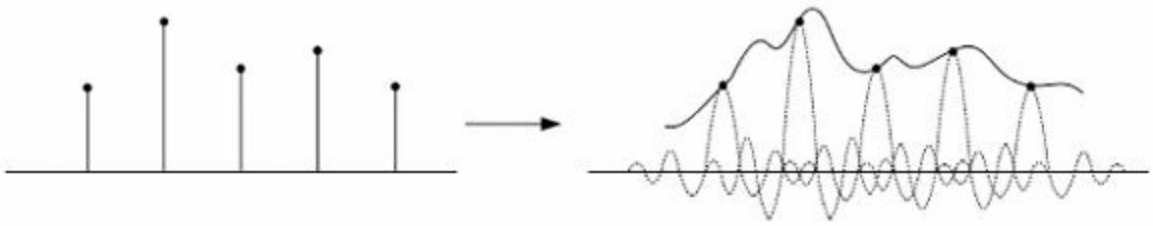


Figure 8.1: The sin(*x*)/*x* outputs of the LPF are summed together to reconstruct the original band-limited input waveform.

In the time domain, you can see how the IR of each sample is overlaid on the others and that the summing together of the peaks and valleys of the sin(*x*)/*x* shape ultimately creates the portions in between the samples which appeared to have been lost during the sampling process.

The process of overlaying the IR on top of the input stream of impulses $x(n)$ and adding up the results to get the final time domain output $y(n)$ is called convolution. Convolution is a mathematical operation used in many fields of science; digital audio processing is just one of them. The mathematical symbol for convolution is * which can be confusing because this is used to represent the multiplication operation in C/C++. In the above example, you convolved the input signal $x(n)$ with the IR $h(n)$ by overlaying the $h(n)$ signal on top of each input impulse, then summing up signals to create the final output $y(n)$. Mathematically, you would write this as Equation 8.1:

$$y(n) = x(n) * h(n) \tag{8.1}$$

Visually, it's easy to understand the concept of overlaying the signals and adding them to get the final result, but how do you describe that mathematically? The answer is that this kind of operation is a special circumstance of a more generalized operation of convolution. Mathematically, convolution for discrete signals is described in Equation 8.2:

$$c(n) = f(n) * g(n) = \sum_{m=-\infty}^{+\infty} f(n)g(n-m) \tag{8.2}$$

In this case, f and g are two generalized signals and neither of them has to be an IR. Convolution is commutative, so that $f * g = g * f$, or Equation 8.3:

$$\begin{aligned} c(n) &= f(n) * g(n) = \sum_{m=-\infty}^{+\infty} f(n)g(n-m) \\ c(n) &= g(n) * f(n) = \sum_{m=-\infty}^{+\infty} g(n)f(n-m) \end{aligned} \tag{8.3}$$

> If you know how a system affects one single impulse, you can exactly predict how it will affect a stream of impulses (i.e., a signal) by doing the time domain overlay. If you have the IR of a system, you have the *algorithm* for the system coded in a single function.

The operation this equation is describing is not simple. On the right-hand side of Equation 8.3 the function $f(n)$ is one signal while the function $g(n - m)$ represents the second signal reversed in time. The multiplication/summation of the two across $-\infty$ to $+\infty$ describes the process of sliding the two signals over each other to create overlapping areas. On each iteration, the area under the curve of overlap between $g(n - m)$ and $f(n)$ is computed. This results in a third (output) signal $c(n)$. This signal $c(n)$ is made up of the overlapping area of the two input signals. This operation is shown graphically in Figures 8.2 and 8.3.

Thus, the convolution of two arbitrary signals is quite involved mathematically. If the two signals

have any complexity at all, the resulting convolution signal is generally not distinguishable as a linear combination of the two. If you know the IR of a system $h(n)$, you can convolve it with the input signal $x(n)$ to produce the output signal $y(n)$. This is the equivalent of multiplying the transfer function $H(z)$ with the input signal $X(z)$ to produce the output $Y(z)$. Thus convolution in the time domain is multiplication in the frequency (z) domain (Equation 8.4).

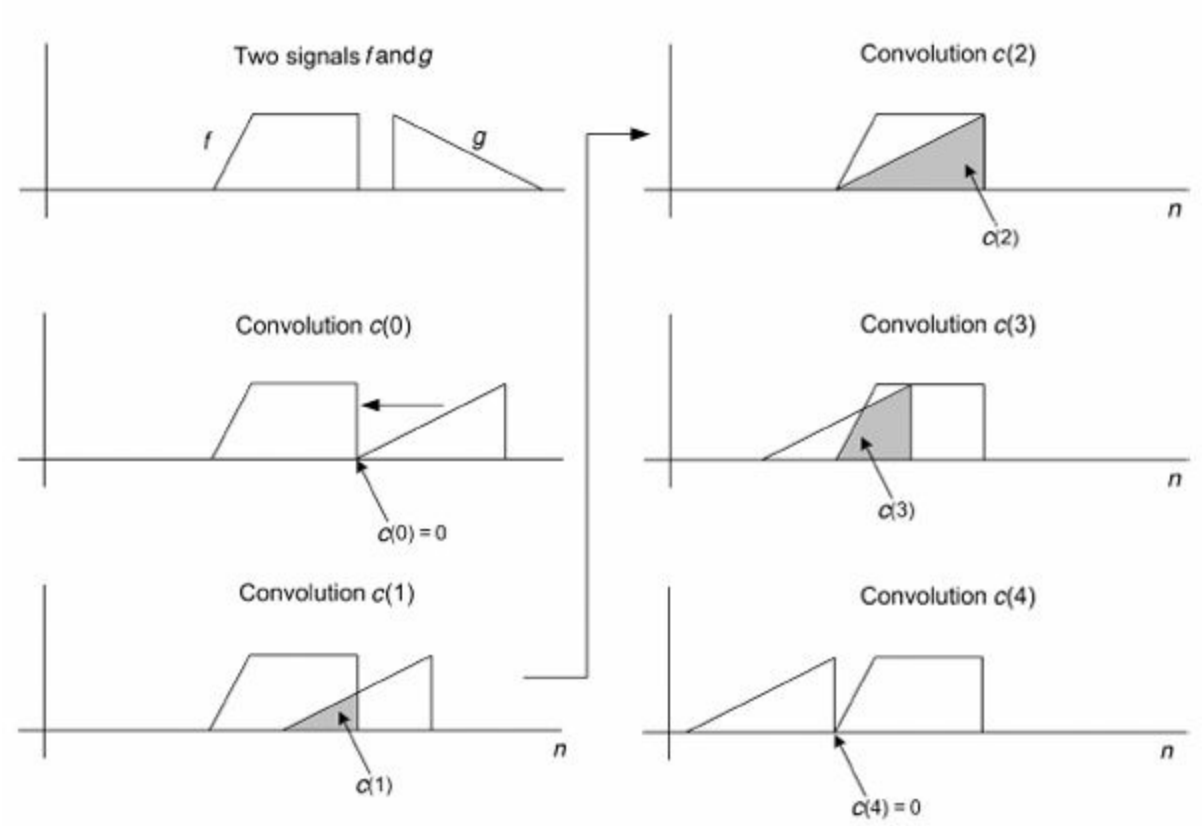


Figure 8.2: Two signals *f* and *g* are convolved. These are discrete signals but the sample symbols have been removed to make it easier to see; instead they are shown as continuous. In the first step, *c*(0), one of the signals is reversed and the two are pushed up next to each other. As the convolution progresses through each step *c*(1) to *c*(4), the overlapping areas are calculated and stored.

$$y(n) = x(n) * h(n) \leftrightarrow Y(z) = X(z)H(z) \tag{8.4}$$

To understand how a FF filter implements convolution first rearrange the block diagram. Let's consider a long FIR filter with $N + 1$ coefficients in Figure 8.4.

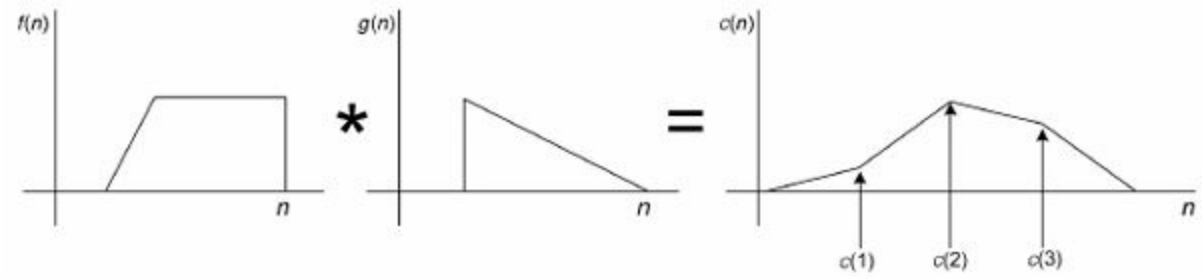


Figure 8.3: The convolution of *f* and *g* results in *c*(n), consisting of five samples *c*(0) through *c*(4), which represent the overlap areas.

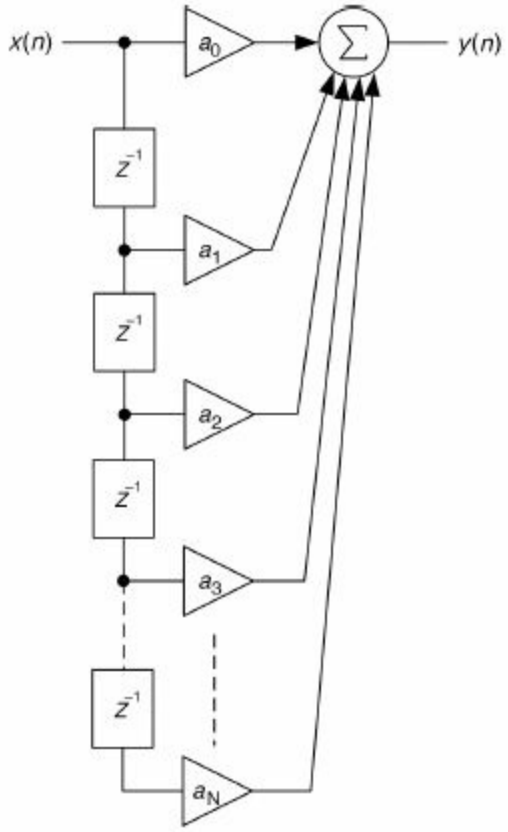


Figure 8.4: The familiar FIR feed-forward structure expanded out to N delay taps with $N+1$ coefficients. It is important to see that there is one less delay element than coefficients since a_0 is multiplied against the original undelayed signal.

Next, mentally rotate the structure so it looks like Figure 8.5. In Figure 8.5 you can see that at any given time, a portion of the input signal $x(n)$ is trapped in the delay line. On each sample period, the input signal slides to the right and a summation is formed with the product of the coefficients and the samples $x(n - d)$. The words "sliding, summation and product" are key here—they're the same words used to describe convolution.

In Figure 8.6, the input signal $x(n)$ moves through the delay line being convolved with the IR on each sample period. Since each sample in the delay line is an impulse, and each impulse is symmetrical when reversed, this is the same as conceptually overlapping the IR on top of each sample and scaling by the sample value. The result is the final summation of all the peaks and valleys of the IR with the delayed signal $x(n)$. Thus, an FIR filter exactly implements discrete convolution in the time domain. This ultimately gives us a whole new way to filter a signal—by convolving against an impulse.

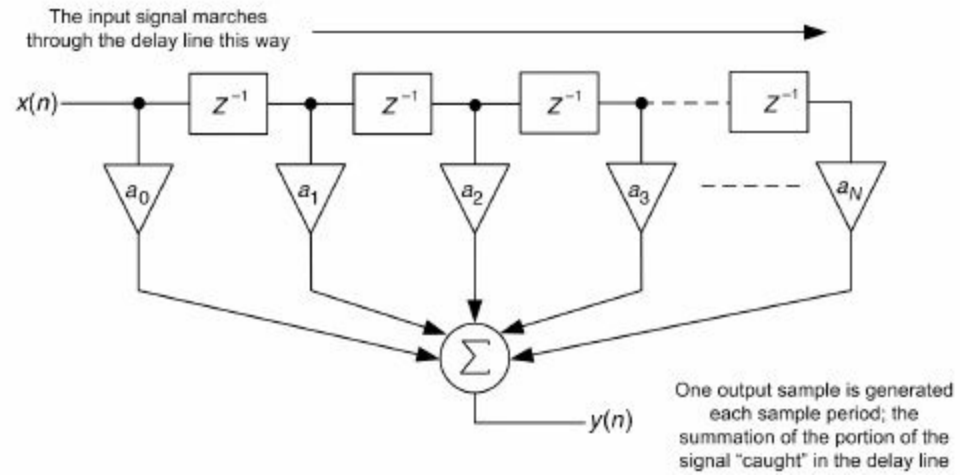


Figure 8.5: This rotated version is sometimes called a transverse delay line or a transverse structure.

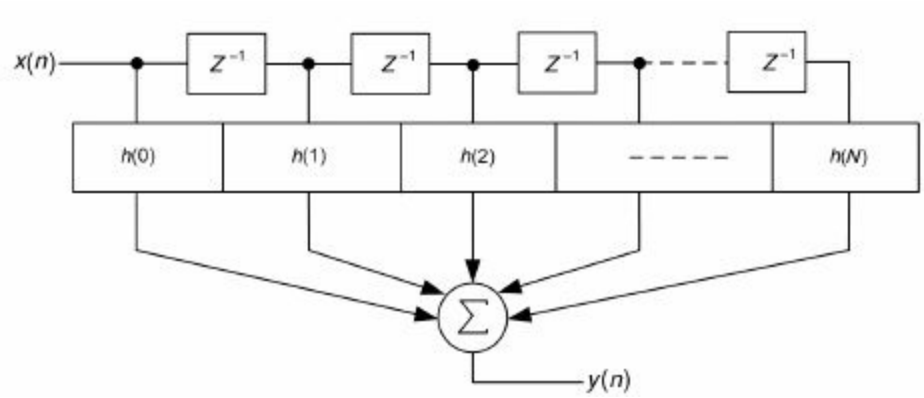


Figure 8.6: You can also think of the coefficients as being frozen in the *h*(*n*) buffer while the input signal marches one sample to the right on each iteration.

So, if an ideal LPF has an IR in the shape sin(*x*)/*x* and we sample the IR, we get a discrete IR *h*(*n*). The more samples we take, the more accurate our version of the IR becomes. This introduces the first way to design a FIR filter: find an IR you like, sample it, and convolve with it to produce the filtered output.

8.2 Using RackAFX's Impulse Convolver

RackAFX has a built-in module to do impulse convolution and a directory of IR files that you can experiment with. The impulses are stored in a directory called IR1024 and they are all 1024-point IRs. Some of them came from RackAFX itself—you can save IRs of any plug-in you make, then load them into the convolver module. You will also learn to write your own convolution plug-in and tell RackAFX that your software would like to receive IRs any time a user loads or creates one using the built-in tools.

First, let's look at the Impulse Convolver tool. Open the modules menu item in RackAFX. There are two built-in FIR modules: the Impulse Convolver 1024 and the FIR Designer located at the bottom of the list. Choose the Impulse Convolver 1024 module. The analyzer will then appear with the IR directory populated and additional IR buttons enabled. These will allow you to load an IR file into the convolver.

8.2.1 Loading IR Files

On the right side of the analyzer (Figure 8.7), you will see a box full of the IRs in your IR1024 directory. You might not have the exact same list as this one but you will have the file "optimal.64.sir" in the list. All the IR files are named with the ".sir" suffix and must be created in RackAFX or loaded using the RackAFX IR file format (see the website for details). RackAFX automatically saves your IRs as. wav files so you can use them in other applications. You can find these .wav files in your IR1024 directory. If you want to convolve with your own .wav files, see the website for code examples.



Figure 8.7: The IR files are visible in the analyzer window.

At the bottom right, you will see the buttons for loading and saving IR files. The first two, Save $h(n)$ and Load $h(n)$, will save and load the .sir files from the IR1024 directory. The lower two buttons save and load the IR to the clipboard. The IR is actually C++ code, and you can use the clipboard to paste this code directly into your own source code. You might do this to hard-code a particular IR or for testing purposes, or to set a default IR.

To get started, click on the file named "optimal.64.sir" and then click the Load h(n) File button (you can also double-click on the file name to load it). The IR for the file will load into the analyzer as well as the Impulse Convolver module itself. You will automatically be switched into the impulse view to see the imported file data (Figure 8.8). Next, click on the Frequency Response button to see the filter's frequency response (Figure 8.9). This LPF was designed to have a cutoff frequency of 500 Hz and a very steep roll-off of about 50 dB/oct. Load a wave file and play the file through the convolver. It will implement this 500-Hz LPF exactly. Try loading some other IR files and playing audio through them.

8.2.2 Creating IR Files

You can capture the IR of any RackAFX plug-in, including your own. As an example let's capture the IR of a built-in module first. Open the module named "HP/LP Filter" from the module menu. Then, open the analyzer, click on the Frequency button and adjust the slider controls to give you a unique shape. For example, I will make a highly resonant LPF by setting the Q to 12 (Figure 8.10).

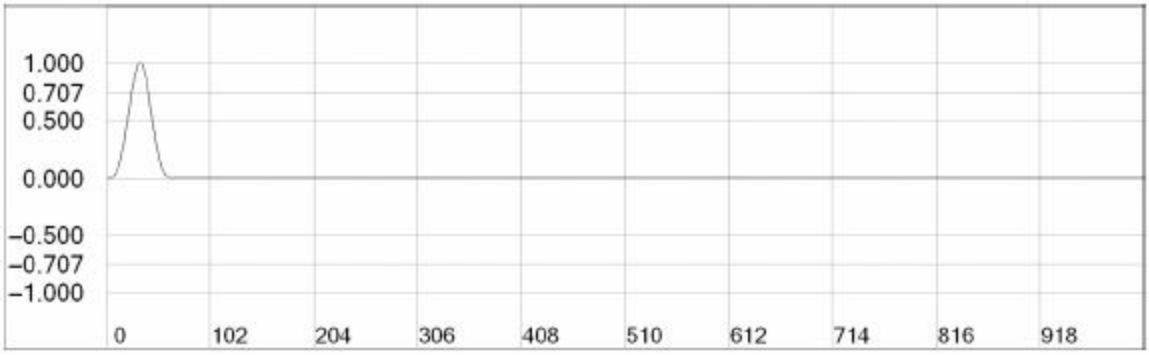


Figure 8.8: The IR for the optimal.64.sir file.

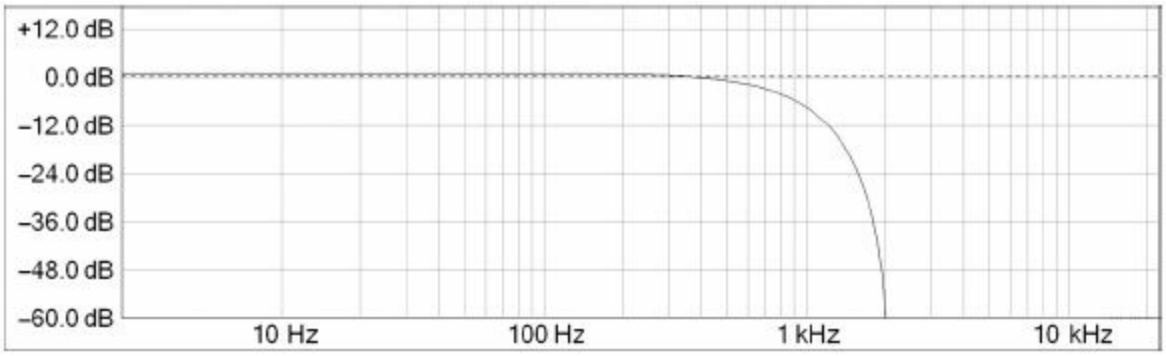


Figure 8.9: The frequency response for the optimal.64.sir file.

Click on the Impulse button in the analyzer. The IR of the filter is shown in Figure 8.11. This IR completely captures the filter at these particular settings (f_c = 1 kHz, Q = 12). If we store the IR of the IIR filter, we can load it into the convolver and turn it into a FIR filter instead.

Click on the Save h(n) File button and name it. It will then appear in the list of available IRs. Play a wave file through the module and remember what it sounds like. Now, go back and open the Impulse Convolver 1024 module from the module menu. You will see your freshly created IR in the list. Double-click on it to load it and you will see the original IR. Click on the Frequency button and you will see the original frequency response. Next, play a wave file through the convolver and you should hear something that is remarkably similar to the original IIR filter.

But, how similar is it? The IR convolver module can convolve up to 1024-point IRs. If the IR of the original filter is longer than 1024 points, then the resulting FIR filter will not be exactly perfect. However, if it is shorter than 1024 points (meaning that the IR becomes 0.0 and remains that way at some point in the IR duration) then the resulting FIR will be a dead-on accurate version of the original. If you look at the IR for the original filter, you can see that it is still just barely ringing right there at the end, so we are not getting an exact duplicate, but upon listening you should hear that they are nearly identical.

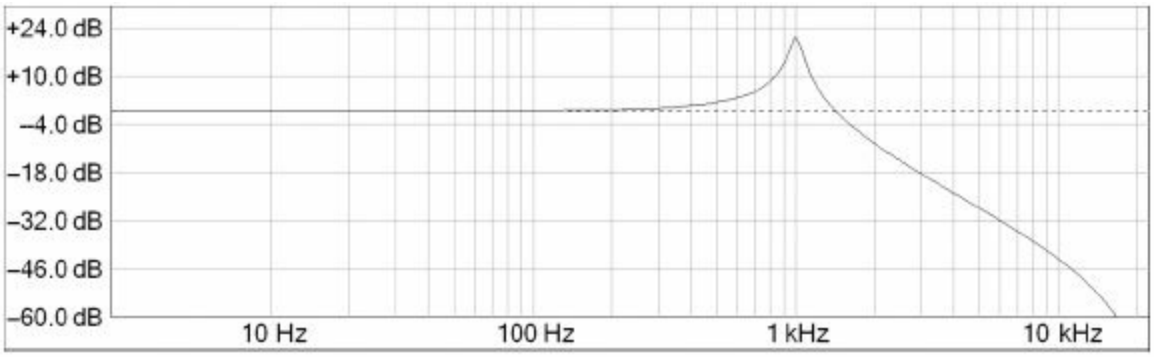


Figure 8.10: The frequency response for the resonant LPF test filter.

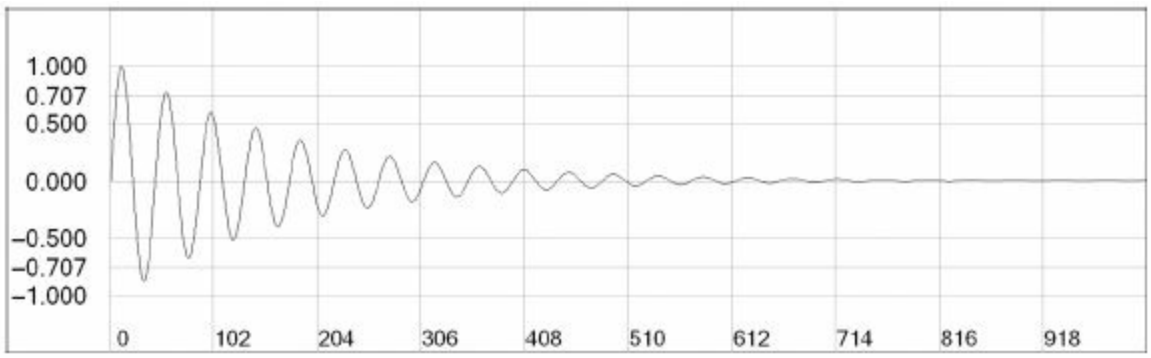


Figure 8.11: The ringing IR of the resonant LPF.

8.2.3 The IR File Format

The IR file actually contains C++ code and you can quickly understand how it works by using the clipboard functions. In the analyzer window that you still have pulled up, click on the button $h(n)$ → Clipboard and after the success message, open a text editor or a C++ compiler. Then, use the operating system (OS) paste function to paste the clipboard data into your editor. You will see something like this at the top:

```
// h(n) Impulse Response
// Length = 1024

m_nIRLength = 1024;
m_h_Left[0] = 0.00000000;
m_h_Left[1] = 0.00503618;
m_h_Left[2] = 0.01998402;
m_h_Left[3] = 0.03938961;
m_h_Left[4] = 0.05777339;
m_h_Left[5] = 0.07477701;
m_h_Left[6] = 0.09007415;
m_h_Left[7] = 0.10337673;

etc…
```

This is the IR data starting with the length (1024) and then the left and right channels respectively. If you scroll to the bottom, you will see the last few samples of the IR:

```
m_h_Right[1016] = -0.00000850;
m_h_Right[1017] = 0.00004104;
m_h_Right[1018] = 0.00008918;
m_h_Right[1019] = 0.00013495;
m_h_Right[1020] = 0.00017745;
m_h_Right[1021] = 0.00021589;
m_h_Right[1022] = 0.00024953;
m_h_Right[1023] = 0.00027775;
```

This was my resonant LPF and you can see that it is still ringing, even after 1024 samples. If you listen to a wave file through a filter like this, you can hear pinging sounds at the peak resonant frequency. These pinging noises are the ringing of the filter. If you look at the C++ code you can tell that the IR data appears to be some kind of member variable information for a C++ object because of the "m_" Hungarian notation used to describe the data.

> All RackAFX plug-ins already have two default IR arrays declared as m_h_Left[1024] and m_h_Right[1024] and another variable m_nIRLength that defines how much of the 1024 point IR buffer is being used. The FIR designer will let you create IRs with variable sizes–in many cases, you don't need all 1024 points to describe the IR of the system. You tell RackAFX that you want it to populate your IR arrays by setting a flag m_bWantIRs in your plug-in constructor. When a user loads or creates an IR in the analyzer, it is automatically delivered to your plug-in.

8.3 Using RackAFX's FIR Designer

RackAFX has a powerful built-in module called FIR Designer that lets you use two popular methods to design FIR filters: the optimal method and the frequency sampling method. The optimal method is sometimes called the "Parks–McClellan algorithm." When the module first opens, you will see the new controls at the right side, as shown in Figure 8.12.

This module creates IRs. You can save the IRs to a file or the clipboard using the same buttons as before. If your plug-in has the m_bWantIRs flag set, any time the user hits the Calculate button to make a new IR, it will automatically be delivered and copied into the plug-in's default IR arrays. Even though the FIR Designer defaults to the optimal method for design, let's begin with the frequency sampling method since the optimal method relies on it.

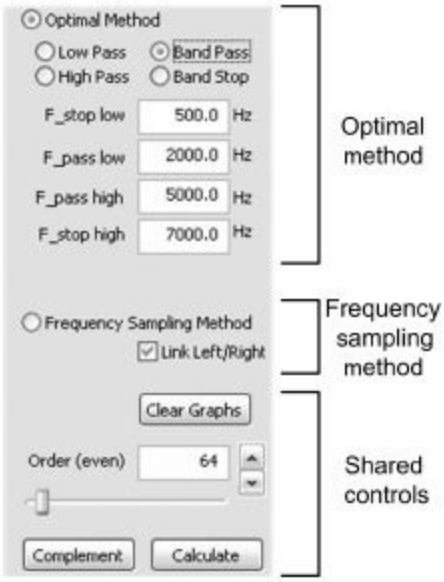


Figure 8.12: The FIR designer controls consist of three parts. The order slider and edit box and the Calculate and Complement buttons are shared between both methods. You enable the method of choice using the radio button controls.

8.4 The Frequency Sampling Method

The frequency sampling method is really interesting because it lets you design any kind of frequency response you want—it can be any arbitrary shape and it does not have to be a classical filter type (LPF, high-pass filter [HPF], band-pass filter [BPF], band-stop filter [BSF]), but you can make these types if you want to. The frequency sampling method involves these steps:

1. Decide on a desired frequency response and plot it in the frequency domain.
2. Sample the frequency response at evenly spaced intervals determined by the filter order you choose.
3. Take the inverse discrete cosine-transform (DCT) of the sampled frequency response to get the sampled IR (the DCT is simply the real part of the fast Fourier transform [FFT]).
4. Load the sampled IR into a convolver and go.

8.4.1 Linear-Phase FIR Using the Frequency Sampling Method

Choose:

N = number of coefficients

Calculate:

For N = odd:

- $(N + 1)/2$ = number of samples in frequency domain, starting at 0 Hz

For N = even

- $N/2$ = number of samples in frequency domain, starting at 0 Hz $\Delta f = f_s/N$ = frequency spacing, starting at 0 Hz

Calculate the filter coefficients a_0 to $a_{N/2}$ with Equation 8.5:

$$\text{For } N = \text{odd}$$
$$a_n = \frac{1}{N}\left[H(0) + 2\sum_{i=1}^{(N-1)/2} |H(i)| \cos\left[2\pi i\left[n - \frac{N-1}{2}\right]/N\right]\right]$$
$$\text{For } N = \text{even} \tag{8.5}$$
$$a_n = \frac{1}{N}\left[H(0) + 2\sum_{i=1}^{N/2-1} |H(i)| \cos\left[2\pi i\left[n - \frac{N-1}{2}\right]/N\right]\right]$$

Note: This produces half the coefficients; the other half are a mirror image, as shown in the example below. Because the IR is guaranteed to be symmetrical about its center, it produces a linear phase FIR filter every time.

Example: Design an LPF with a cutoff of 5.5 kHz, $f_s = 44.1$ kHz, $N = 16$.

Solution:

1. $N = 16$, which produces eight sampled points in the frequency domain with a spacing of 2.756 kHz.
2. Sample the plot, producing the magnitude response, $H(i)$ (Figure 8.13).

For this plot notice that:

- The plot is linear in frequency. To design a frequency sampling method filter by hand, it is often easier to start with a linear frequency axis.
- The point at Nyquist is not used; it is there to pin down the response that would occur after Nyquist.

The sampled frequency response is read directly off the plot: (0 dB = 1.0, −60 dB = 0.001).

$$|H(i)| = \{1.0, 1.0, 1.0, 0.001, 0.001, 0.001, 0.001, 0.001\}$$

1. Use Equation 8.5 to extract IR $h(n)$, which are the coefficients, a_n.

$a_0 = 0.04858366$
$a_1 = 0.00364087$
$a_2 = -0.05199205$
$a_3 = -0.07047625$
$a_4 = -0.02194221$
$a_5 = 0.08695625$
$a_6 = 0.21101949$
$a_7 = 0.29421023$
$a_8 = 0.29421023$
$a_9 = 0.21101949$
$a_{10} = 0.08695625$
$a_{11} = -0.02194221$

$a_{12} = -0.07047625$

$a_{13} = -0.05199205$

$a_{14} = 0.00364087$

$a_{15} = 0.04858366$

Notice the symmetry about a_7 to a_8 boundary.

2. Measure the response (Figure 8.14).

The resulting filter is guaranteed to exactly match the desired frequency response at the sampled points only. In between the sampled points, the frequency response can do anything; rippling in the pass band and stop band can occur as shown here. You can see that this is a pretty bad rendition of our desired response.

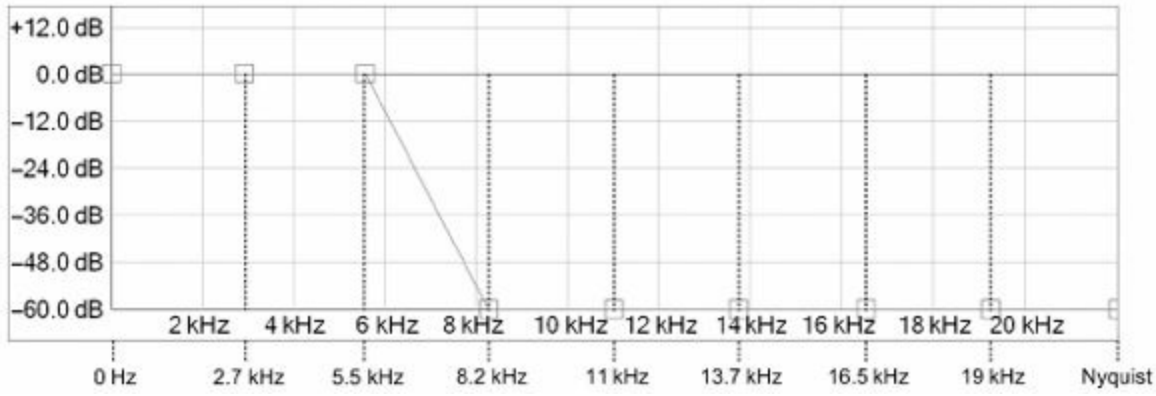


Figure 8.13: Here is our prototype LPF with cut-off at 5.5 kHz; its response becomes −60 dB (0.001) at 8.2 kHz, which is a steep roll-off.

The first lobe in the stop band only produces about −14 dB of attenuation, which is poor considering that we wanted a filter that would have a stop-band attenuation of −60 dB. To improve the response, you have several choices:

- Relax the specifications.
- Add more points to the desired response.
- Increase the filter order.

Relax the specifications and roll off the response less steeply by changing the point at 8.2 kHz to −12 dB (0.25) instead of −60 dB (Figure 8.15). Now, the sampled frequency response is

$$|H(i)| = \{1.0, 1.0, 1.0, 0.25, 0.0, 0.0, 0.0, 0.0\}$$

The resulting magnitude response $|H(f)|$ is shown in Figure 8.16.

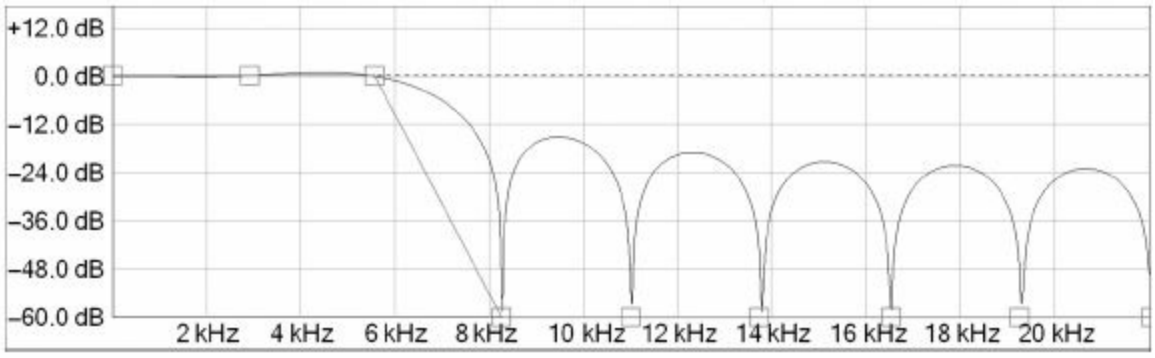


Figure 8.14: The resulting low-pass filter magnitude response.



Figure 8.15: The same design with specifications relaxed; the slope is less steep.

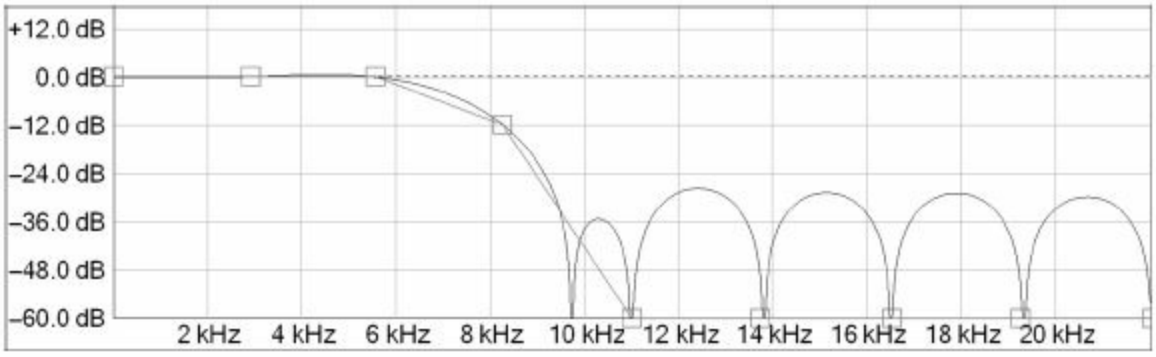


Figure 8.16: The relaxed magnitude response shows improved stop-band attenuation. Now, the first lobe in the stop band has moved to a magnitude of about −30 dB, an improvement of about 15 dB.

8.5 Complementary Filter Design for Linear Phase FIR Filters

This technique results in a complementary filter, rotated about the center of the Nyquist bandwidth, that is, rotated about Nyquist/2. To convert an LPF to HPF or vice versa on a linear phase FIR:

For N = even

- Multiply the even-numbered coefficients by −1.

For N = odd

- Multiply the odd-numbered coefficients by −1.

This will rotate the frequency response around Nyquist/2 such that an LPF will become an HPF.

However, they will not share the same cutoff frequency, but will rather be mirror images of each other. Thus, the first filter design above with a cutoff point of 5.5 kHz would produce an HPF with a cutoff frequency 5.5 kHz *below Nyquist*. Table 8.1 shows the coefficients, while Figure 8.17 shows the frequency response.

Table 8.1: The LPF and complementary HPF coefficients for the current design.

Low-Pass Filter	Complementary High-Pass Filter
$a_0 = 0.02598719$	$a_0 = -0.02598719$
$a_1 = 0.01331121$	$a_1 = 0.01331121$
$a_2 = -0.02018474$	$a_2 = 0.02018474$
$a_3 = -0.05761\ 180$	$a_3 = -0.05761\ 180$
$a_4 = -0.04827195$	$a_4 = 0.04827195$
$a_5 = 0.04957144$	$a_5 = 0.04957144$
$a_6 = 0.20692666$	$a_6 = -0.20692666$
$a_7 = 0.33027202$	$a_7 = 0.33027202$
$a_8 = 0.33027202$	$a_8 = -0.33027202$
$a_9 = 0.20692666$	$a_9 = 0.20692666$
$a_{10} = 0.04957144$	$a_{10} = -0.04957144$
$a_{11} = -0.04827195$	$a_{11} = -0.04827195$
$a_{12} = -0.05761180$	$a_{12} = 0.05761\ 180$
$a_{13} = -0.02018474$	$a_{13} = -0.02018474$
$a_{14} = 0.01331121$	$a_{14} = -0.01331121$
$a_{15} = 0.02598719$	$a_{15} = 0.02598719$

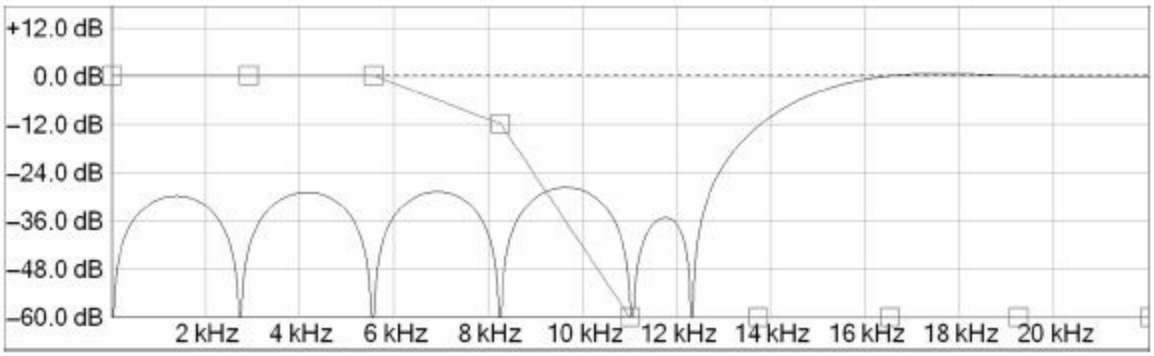


Figure 8.17: The complementary HPF has its cut-off point at 16.5 kHz and you can see the rotation about the ½ Nyquist frequency.

8.6 Using RackAFX's Frequency Sampling Method Tool

RackAFX allows you to implement frequency sampling method filters by drawing responses directly on the analyzer's frequency response plots. You can place as many points as you like as well as choose the order (even only) of the filter. The previous examples were done using the frequency sampling method tool. Let's do an example similar to the last one but in the log frequency domain instead. First, click on the radio button marked frequency sampling method and you will see a red horizontal line appear on the graphs. Make sure the frequency axis is log. In this design, let's try to make a resonant LPF with a peak frequency around 1 kHz.

You will see two boxes on the red line, one at DC, the other at Nyquist. These points cannot be removed. To enter points and move them, use the following rules:

- Right-click on the red line to add a new point.
- Click on the new point and drag it up or down.
- To remove a point, right-click on it and choose “delete point” from the pop-up menu.
- Hit the Calculate button when you are done to view the resulting filter.

Hit the Frequency Sampling button and note the filter order of 64 (if it is not 64, change it with the slider or the edit control). Create a resonant LPF with a peak frequency of 1 kHz. In Figure 8.18 you can see the bandwidth of the peak is 1.5 kHz and the response goes to −60 dB at = kHz. Next, hit the Calculate button to get Figure 8.19.

You can use the order slider to slowly increase the filter order until the response matches your sampled points as closely as possible. You can move the slider, or type an order into the edit box and hit the Tab button to inject it. You can also use the small up/down arrow buttons to advance the order up or down in even increments. Typically, the way you would use this is to first get close with the slider, then use the up/down arrows to fine tune the design. Figure 8.20 shows the same filter with the order set to 164.

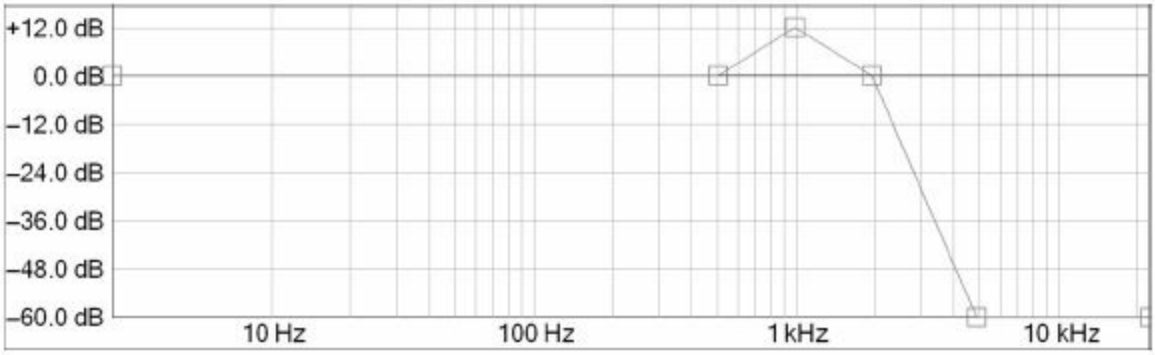


Figure 8.18: The prototype resonant LPF sampled points.

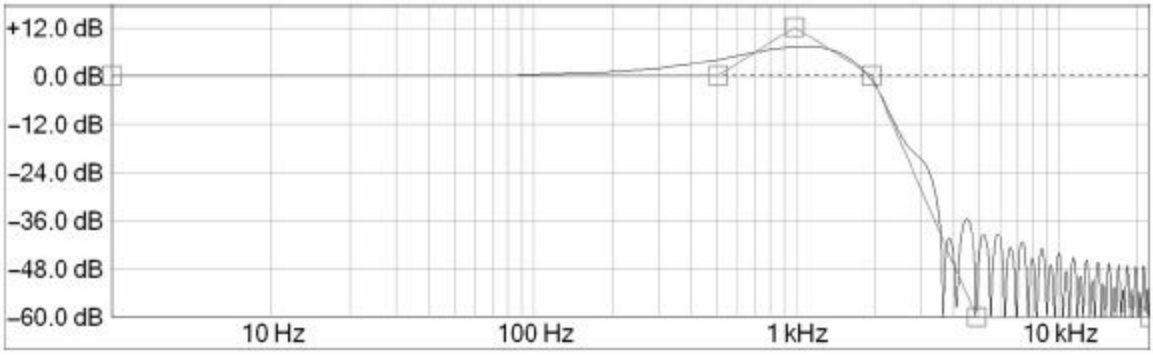


Figure 8.19: The 64-tap FIR filter produces marginal results with poor stop-band attenuation.

Play an audio file through the new filter and listen to the resonant LPF characteristics. Here are some interesting things you can do in RackAFX while the audio file is playing or looping:

- You can move the order control; the filter will be updated in real time and you can hear the results.
- You can add or remove points on the desired response or move them around, then hit Calculate to update the filter in real time, and you can also hear the results.
- You can save the IR as a file, then load it into the Impulse Convolver module just as before.

8.7 Designing a Complementary Filter

You can convert any design into a complementary design by hitting the Complementary button. With the current 164th-order resonant LPF, first switch to the linear scale (Figure 8.21).

Hit the Complement button to create the complementary HPF filter. The original design points are left to show you the complementary nature of the filter. You can clearly see the rotation about ½ Nyquist here (Figure 8.22). You can perform this operation while audio files are playing in RackAFX as well. If you look at the IRs in the .sir file you can see the operation of negating the odd-numbered coefficients.

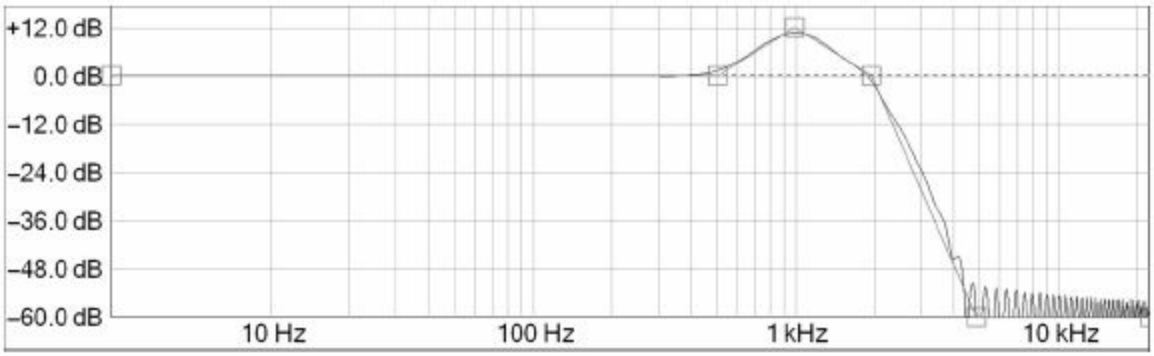


Figure 8.20: The 164th-order FIR filter produces an excellent match to the specifications.

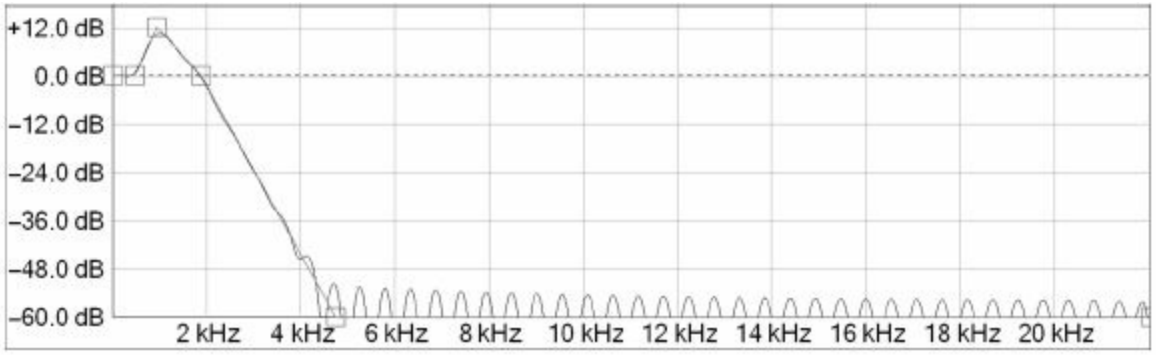


Figure 8.21: The linear domain frequency response of the 1-kHz LPF.

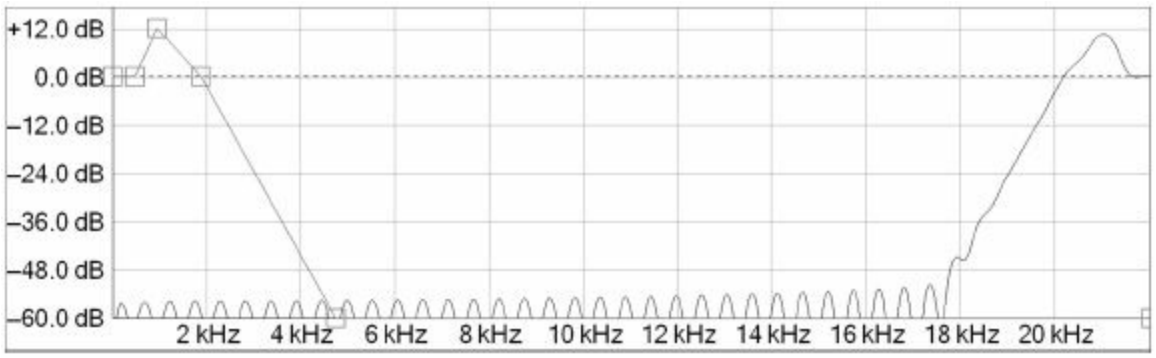


Figure 8.22: The complementary HPF shows the mirror image nature of complementary filter designs; the new resonant peak is 1 kHz below Nyquist.

8.8 The Optimal (Parks–McClellan) Method

FIR filters produce all zeros in the *z*-plane. You can see the effect of these zeros in the frequency sampling method filters we designed—they create the notches and humps in the frequency response. In general, we seek to minimize the humps in the pass band and set some kind of limit for the largest lobe we see in the stop band. In the frequency sampling filters, we changed the design specifications or the order of the filter to get a response that satisfied our initial desired response to some extent. The optimal method is an algorithm for designing classical filters (LPF, HPF, BPF, BSF). You can create filters with extremely steep roll-off edges that would be difficult to synthesize in the analog domain. The word "optimal" comes from the engineering phrase "optimal in the Chebychev way," which describes a filter design with a certain kind of rippling in the pass band and stop band.

The LPF in Figure 8.23 might be specified like this:

- Pass-band ripple: less than 0.1 dB
- Stop-band attenuation: greater than −24 dB

The pass-band ripple (0.1 dB) and stop-band attenuation (−24 dB) are converted into weighting values using Equation 8.6:

$$\begin{aligned} \text{Pass-band weight} &= (10^{\text{pb\_ripple(dB)}/20}) - 1 \\ \text{Stop-band weight} &= (10^{\text{sb\_atten(dB)}/20}) \end{aligned} \tag{8.6}$$

The optimal method uses these weighting values in its design calculation. The weightings give

acceptable error tolerances in each band. It the weights are equal, then there will be an equal deviation from the ideal in both bands. This is called an *equiripple* design. The specifications for the optimal method are:

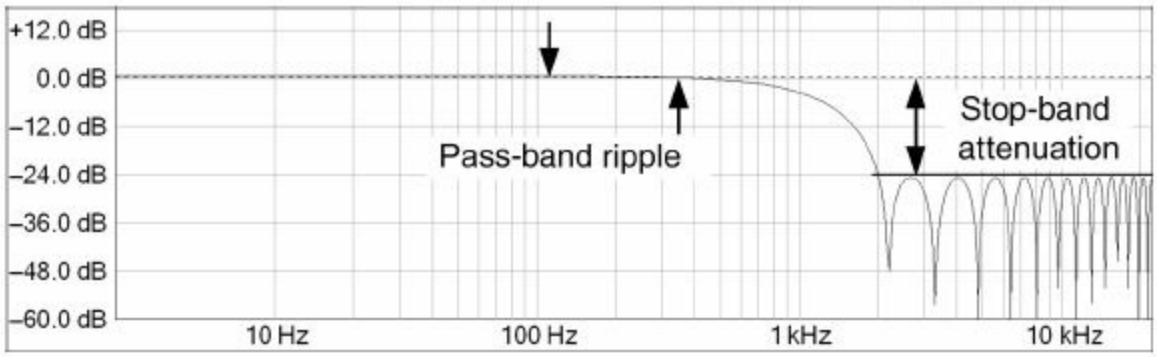


Figure 8.23: An FIR LPF can be specified by the acceptable ripple in the pass band (barely observable here) and the minimum stop-band attenuation.

- The pass-band and stop-band frequencies
- The maximum pass-band ripple and minimum stop-band attenuation
- The order of the filter (the number of zeros in the frequency response)

The algorithm works by first distributing the zeros out equally across the frequency axis. This is going to produce a series of humps in the pass band (the ripple) and lobes in the stop band. The ripples and lobes will produce a set of maxima and minima frequencies. The problem is that for a given specification, the locations of the extremal frequencies are not known before hand. The optimal method must use a search algorithm to iteratively move the locations around until a solution is found that is within the tolerance of the specifications. The optimal method uses an algorithm called the Remez exchange method to iteratively move the zeros around until a solution converges. It must be noted that the Remez exchange algorithm is not guaranteed to converge for a given set of specifications.

Once the Remez exchange method finds the locations of the extremal frequencies, the resulting $H(f)$ graph is sampled using the frequency sampling method to produce the IR, which is the set of FIR filter coefficients. To summarize the method:

- Specify the filter parameter and order.
- Equally distribute zeros across the spectrum.
- Set the pass-band and stop-band weights.
- Use the Remez exchange algorithm to iteratively find the location of the zeros.
- Sample the resulting frequency response to produce the final IR.

8.9 Using RackAFX's Optimal Method Tool

RackAFX lets you design any of the four classical filter types using the equiripple weighting for the pass band and stop band. BPF and BSF filters have an additional band edge to specify, but the meaning of pass-band ripple and stop-band attenuation is the same for all filter types. You can start with an LPF design by clicking on the Optimal button. Try the following filter specifications:

- Type: LPF
- F_pass low: 1 kHz (the low edge cut-off frequency)
- F_stop low: 2 kHz (the lowest frequency that must receive the required stop band attenuation)
- Filter order: 16

Now, use the Calculate button to generate the filter. You can see from Figure 8.24 that the filter is not performing exactly to specifications. Although the pass band looks good, the stop band does not. Next, begin increasing the order of the filter using the slider or nudge buttons until you get the stop-band attenuation you desire; for −48 dB of attenuation, you will need a 104th-order filter as shown in Figure 8.25. Finally, check the IR and you will see a truncated version of the sin $x(x)/x$ function (Figure 8.26).

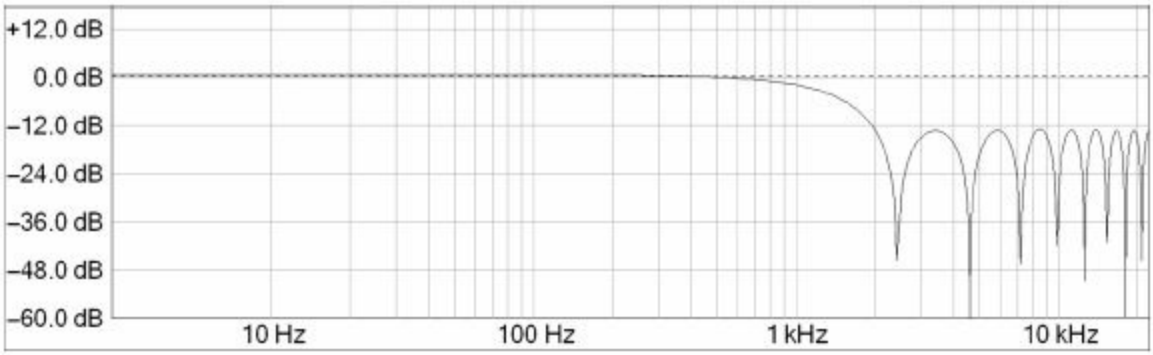


Figure 8.24: The pass-band ripple is well within spec; however, the stop-band attenuation is only getting to about −12 dB.

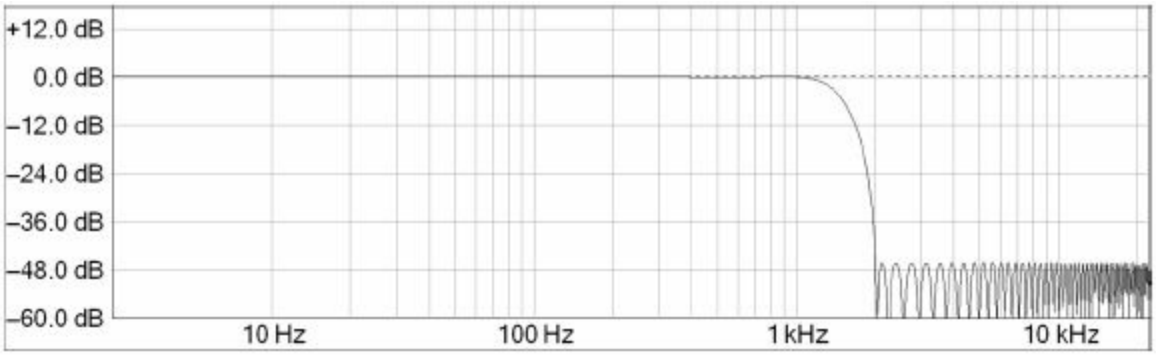


Figure 8.25: The 104th-order optimal method LPF produces the desired stop-band attenuation.

The Remez exchange algorithm is not guaranteed to converge. You will receive an error message if it does not converge. Even if the algorithm does converge, the resulting IR is not guaranteed to be finite. Increasing the filter order will not necessarily produce a better design.

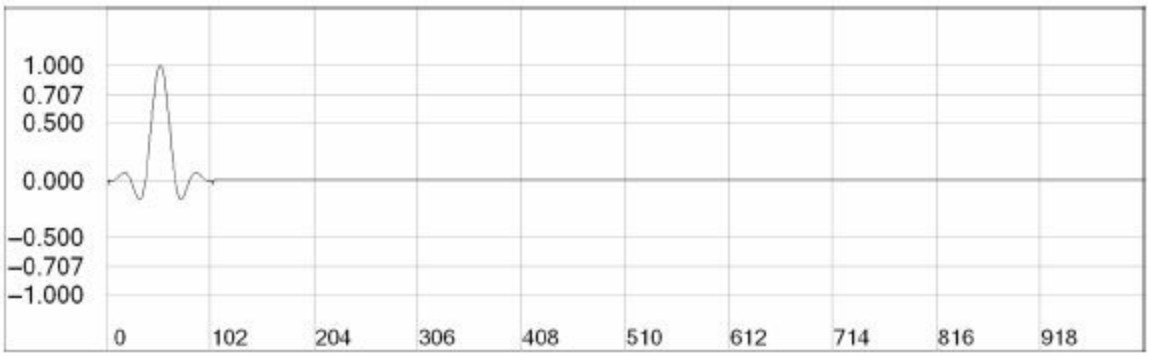


Figure 8.26: The IR for the 104th-order optimal method LPF.

Experiment:

- Load a wave file and audition the filter.
- Try the other filter types (HPF, BPF, BSF).
- Adjust the order control noting when the Remez exchange algorithm fails to converge or the filter blows up.
- Try to find the lowest possible filter order to just match the desired specifications.
- Save IR files with various filters you design.
- Copy and paste the IR code into your own convolution plug-in.

8.10 Design a Convolution Plug-In

In order to implement the convolution (FIR) algorithm you need to use the delay line theory from the last chapter. The filters often need hundreds or thousands of delay elements and you know that a circular buffer works perfectly for storing and updating a sequence of $x(n)$, $x(n-1)$, $x(n-2)$… In addition, the IR will need to be stored in a buffer and accessed sequentially with a pointer like the input buffer. The convolution equation in Equation 8.2 accumulates from $-\infty$ to $+\infty$ which uses both past and future data. We can only use past data and so we only need half of the equation. The generalized FIR convolution equation is Equation 8.7:

$$y(n) = h_0x(n) + h_1x(n-1) + h_2x(n-2) + \ldots + h_M(n-M) \tag{8.7}$$

The number of delay elements required is $M-1$ since the first term $h_0x(n)$ operates on the current, undelayed input signal. So, a 64-tap FIR requires 63 delay elements. Remember from Chapter 7 that when we access a circular buffer and write the current input sample, we are overwriting the oldest sample, $x(n-M)$ but we can use this to our advantage in this case by using a 64-element circular buffer to implement a 64-tap FIR and by writing in the first sample before doing the convolution operation. This will give us a buffer with $x(n)$ through $x(n-M-1)$ lined up and ready for access. We will have an identically sized buffer to store the IR, $h(n)$. During the convolution operation we will zip through both buffers at the same time, accumulating the product of each operation. The only tricky thing is that the IR buffer will be reading sequentially from top to bottom exactly once each sample period to create the sequence $h(0)$, $h(1)$, $h(2)$, and so on. The input buffer will be circular and reading *backwards* to create the sequence $x(n)$, $n(n-1)$, $x(n-2)$, and so on, shown graphically in Figure 8.27.

If you look at your base class file, PlugIn.h, you will find the declarations of your built-in IR buffers and variables:

```
// impulse response buffers!
float m_h_Left[1024];
float m_h_Right[1024];

// the length of the IR from 0 to 1024
int m_nIRLength;

// flag to set to request impulse responses from the UI
bool m_bWantIRs;
```

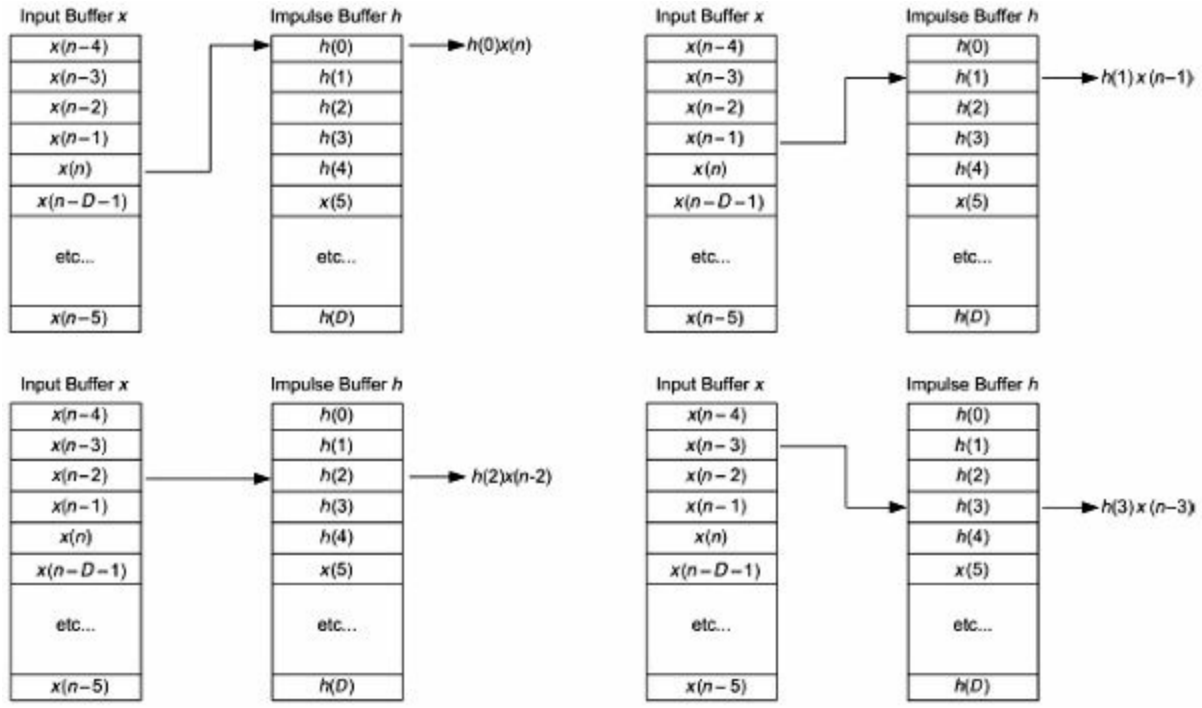


Figure 8.27: The first four multiplication operations for a convolution operation shows how one buffer reads backward while the other reads forward.

Table 8.2: IR variables

Variable	Description
m_h_Left[1024]	The IR buffer for the left channel
m_h_Right[1024]	The IR buffer for the right channel
m_nIRLength	The length of the current convolution
m_bWantIRs	A flag to tell RackAFX to populate your IR buffers automatically whenever the user loads an IR file or creates a filter with the FIR Designer tool

These variables are shown in Table 8.2.

8.10.1 Project: Convolver

Create a new RackAFX project; I named mine “Convolver.” It has no GUI elements to set up.

8.10.2 Convolver.h File

Declare the variables you need to implement a stereo convolution and remember that the IR buffers and length variables are already declared for you. We need the following:

- Buffer to hold the *x* input for the left channel
- Buffer to hold the *x* input for the right channel
- Read index for the IR delay line (buffers)
- Read index for the input (*x*) buffers
- Write index for sequentially writing input samples into the delay lines

```
// Add your code here: ----------------------------------------------- //
//
// pointers to the left and right input buffers
float* m_pBufferLeft;
float* m_pBufferRight;

// read index for delay lines (input x buffers)
int m_nReadIndexDL;

// read index for impulse response buffers
int m_nReadIndexH;

// write index for input x buffer
int m_nWriteIndex;
```

8.10.3 Convolver.cpp File

Constructor

- Create the buffers.
- Flush the buffers; reset to 0.0.
- Reset all indices.

```
CConvolver::CConvolver()
{
	<SNIP SNIP SNIP>

	// Finish initializations here

	// set our max buffer length for init
	m_nIRLength = 1024;  // 1024max

	// dynamically allocate the input x buffers and save the pointers
	m_pBufferLeft = new float[m_nIRLength];
	m_pBufferRight = new float[m_nIRLength];

	// flush x buffers
	memset(m_pBufferLeft, 0, m_nIRLength*sizeof(float));
	memset(m_pBufferRight, 0, m_nIRLength*sizeof(float));

	// flush IR buffers
	memset(&m_h_Left, 0, m_nIRLength*sizeof(float));
	memset(&m_h_Right, 0, m_nIRLength*sizeof(float));

	// reset all indices
	m_nReadIndexDL = 0;
	m_nReadIndexH = 0;
	m_nWriteIndex = 0;
}
```

You can see that we've allocated the memory for input buffers of maximum size 1024. We've also flushed out all buffers by setting all data to 0.0 with memset() and finally reset the indices to all point to the tops of the buffers.

Destructor

- Since we allocated memory in the constructor, we need to remove it in the destructor:

```
CConvolver::~CConvolver(void)
{
	// free up our input buffers
	delete [] m_pBufferLeft;
	delete [] m_pBufferRight;
}
```

prepareForPlay()

- We also need to flush the buffers and reset the indices:

```
bool __stdcall CConvolver::prepareForPlay()
{
	// Add your code here:
	// flush buffers
	memset(m_pBufferLeft, 0, m_nIRLength*sizeof(float));
	memset(m_pBufferRight, 0, m_nIRLength*sizeof(float));

	// reset indices
	m_nReadIndexDL = 0;
	m_nReadIndexH = 0;
	m_nWriteIndex = 0;

	return true;
}
```

processAudioFrame()

- Implement the convolution loop.

The loop will need to do the following operations (these are not streamlined in the code for ease of reading; you can always improve on the code by combining steps together). You are implementing the operation in Figure 8.27:

- Read the current sample $x(n)$ and write it into the buffer.
- Reset the delay line read pointer to the current input sample, $x(n)$; the pointer needs to be reset because its value will change and be destroyed later in the loop.
- Reset the IR read index to point to the top of the buffer at location 0.
- Set up the accumulator and create the convolution loop.
- After forming each product $h(i)x(n - i)$, *increment* the IR buffer read pointer and *decrement* the delay line read pointer.
- Check for a wrap after the delay line pointer is decremented.
- Write the output sample.
- Process the second (right) channel the same way.
- Increment the delay line write index and wrap if necessary.

```
bool __stdcall CCConvolver::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
UINT uNumInputChannels, UINT uNumOutputChannels)
{
	// Do LEFT (MONO) Channel; there is always at least one input/one output
	// Read the Input
	float xn = pInputBuffer[0];

	// write x(n) -- now have x(n) -> x(n-1023)
	m_pBufferLeft[m_nWriteIndex] = xn;

	// reset: read index for Delay Line -> write index
	m_nReadIndexDL = m_nWriteIndex;

	// reset: read index for IR - > top (0)
	m_nReadIndexH = 0;

	// accumulator
	float yn_accum = 0;

	// convolve:
	for(int i=0; i<m_nIRLength; i++)
	{
		// do the sum of products
		yn_accum += m_pBufferLeft[m_nReadIndexDL]*m_h_Left[m_nReadIndexH];

		// advance the IR index
		m_nReadIndexH++;

		// decrement the Delay Line index
		m_nReadIndexDL--;

		// check for wrap of delay line (no need to check IR buffer)
		if(m_nReadIndexDL < 0)
			m_nReadIndexDL = m_ nIRLength -1;
	}

	// write out
	pOutputBuffer[0] = yn_accum;
```

Now you should try to write the code for the second (right) channel. After you're done, check it against the following code:

```
	// Mono-In, Stereo-Out (AUX Effect)
	if(uNumInputChannels == 1 && uNumOutputChannels == 2)
		pOutputBuffer[1] = pOutputBuffer[0]; // just copy
```

```
	// Stereo-In, Stereo-Out (INSERT Effect)
	if(uNumInputChannels == 2 && uNumOutputChannels == 2)
	{
		// Read the Input
		xn = pInputBuffer[1];

		// write x(n) -- now have x(n) -> x(n-1023)
		m_pBufferRight[m_nWriteIndex] = xn;

		// reset: read index for Delay Line -> write index
		m_nReadIndexDL = m_nWriteIndex;

		// reset: read index for IR - > top (0)
		m_nReadIndexH = 0;

		// accumulator
		yn_accum = 0;

		// convolve:
		for(int i=0; i<m_nIRLength; i++)
		{
			// do the sum of products
			yn_accum +=
				m_pBufferRight[m_nReadIndexDL]*m_h_Right
				[m_nReadIndexH];
			// advance the IR index
			m_nReadIndexH++;

			// decrement the Delay Line index
			m_nReadIndexDL--;

			// check for wrap of delay line (no need to check IR buffer)
			if(m_nReadIndexDL < 0)
				m_nReadIndexDL = m_nIRLength-1;
		}
		// write out
		pOutputBuffer[1] = yn_accum;
	}

	// incremnent the pointers and wrap if necessary
	m_nWriteIndex++;
	if(m_nWriteIndex >= m_nIRLength)
		m_nWriteIndex = 0;

	return true;
}
```

Increment the buffer write pointer after the end of the right channel processing. The reason is that it is shared between both left and right (just like the IR buffer read index is shared between the left and right IR buffers). Build and test the code; in order to quickly test the code, you can use a feature of the FIR Designer and copy an IR to the clipboard, then paste it into the constructor. Open the FIR Designer and use the default optimal method LPF design—it will automatically be calculated when you first see the FIR Designer interface. Hit the button h(n) → Clipboard to copy the IR code to your Windows clipboard. Now, go to the constructor and use the paste function Ctrl+V to load in the IR code. Do this at the end of all the initialization stuff you wrote in Step 2. It should look like this:

Constructor

```
CConvolver::CConvolver()
{
	<SNIP SNIP SNIP>

	// reset all indices
	m_nReadIndexDL = 0;
	m_nReadIndexH = 0;
	m_nWriteIndex = 0;

	// h(n) Impulse Response
	// Length = 1024
	m_nIRLength = 65;

	m_h_Left[0] = -0.00057406;
	m_h_Left[1] = -0.00016395;
	m_h_Left[2] = -0.00015001;
	m_h_Left[3] = -0.00007587;
	m_h_Left[4] = 0.00007552;
	m_h_Left[5] = 0.00029672;
	m_h_Left[6] = 0.00063462;
	m_h_Left[7] = 0.00108109;
	etc...

	m_h_Right[1018] = 0.00000000;
	m_h_Right[1019] = 0.00000000;
	m_h_Right[1020] = 0.00000000;
	m_h_Right[1021] = 0.00000000;
	m_h_Right[1022] = 0.00000000;
	m_h_Right[1023] = 0.00000000;
}
```

This IR has 65 samples and will create a 65-tap FIR filter. Compile the dynamic link library (DLL) and load it into RackAFX. Play a wave file through it; it's an LPF at 1 kHz, so this will be easy to verify by ear. Open the analyzer window and look at the frequency and IRs—these will also be identical to the original design. Now that you have verified that your convolution works properly, you can set the IR flag in your constructor to let RackAFX know to deliver your IRs when the user makes, loads, or calculates them. You can also remove or comment-out the default IR code you pasted in to do the initial testing:

```
CConvolver::CConvolver()
{
	<SNIP SNIP SNIP>

	// reset all indices
	m_nReadIndexDL = 0;
	m_nReadIndexH = 0;
	m_nWriteIndex = 0;

	// set the flag for RackAFX to load IRs into our convolver
	m_bWantIRs = true;
}
```

Now, build and load your DLL. Next, open the analyzer window and you will see two new buttons on

the right below the IR/FIR design area. Because your plug-in wants IRs, RackAFX reveals these two hidden buttons on the analyzer panel allowing you to use either the FIR Designer or the IR directory to create or load IR files. As soon as this window opens, it delivers the current IR loaded, which is the optimal LPF design by default. If you change the design it will remember it if you close the analyzer.

Load a wave file and hit Play—this will play the optimal LPF design. Now, hit one of the other buttons (HPF, BPF, or BSF) and the IR will automatically be calculated and loaded into your plug-in, even if an audio file is playing. Test the other IR tools, like the frequency sampling method and the IR Directory, and listen to the resulting filters.

8.11 Numerical Method FIR Filters

The last type of FIR filters consists of numerical methods and is not based on IRs or convolution, even though many of them implement a convolution loop. There are books full of numerical methods for performing all kinds of operations, from integration and differentiation to moving averages and interpolation. In this section we will examine a few algorithms and if it interests you, get a book on numerical methods and you will have a plethora of algorithms to try out. Some of these filters can be implemented as FIR topologies while others are better suited for a straight mathematical function call.

8.11.1 Moving Average Interpolator

Choose:

N = number of points to interpolate, minimum = 2

Calculate:

$$a_k = \frac{1}{N}, \text{ where } 0 \leq k \leq N-1 \tag{8.8}$$

The moving average interpolator (or MA filter) in Figure 8.28 implements a sliding window of N samples wide, over which the input samples are averaged. Each sample gets an equal weight of $1/N$.

Example: Design a five-point MA filter.

Solution:

$N = 5$, so $a_k = 1/5 = 0.20$

To code this in RackAFX, you would flush the IR buffers, then set the first five values of each impulse array to 0.2 and set the m_nIRLength variable to 5:

```
// disable RackAFX IRs for the MA filter test
m_bWantIRs = false;

// h(n) Impulse Response
// MA Filter N = 5
```

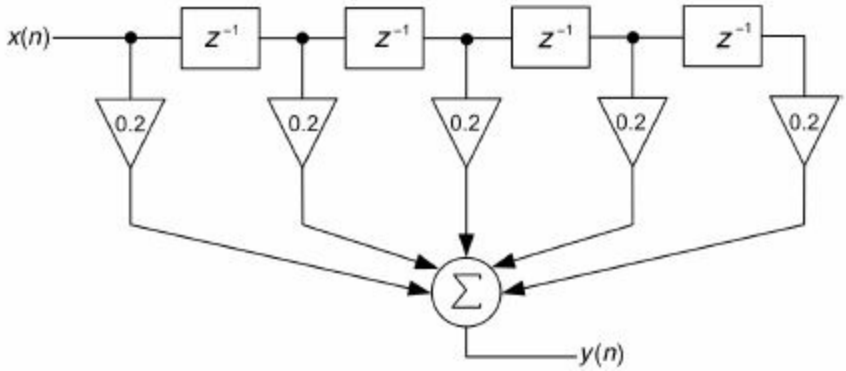


Figure 8.28: The implementation of a five-point MA filter.

```
    m_nIRLength = 5;

    m_h_Left[0] = 0.2;
    m_h_Left[1] = 0.2;
    m_h_Left[2] = 0.2;
    m_h_Left[3] = 0.2;
    m_h_Left[4] = 0.2;

    m_h_Right[0] = 0.2;
    m_h_Right[1] = 0.2;
    m_h_Right[2] = 0.2;
    m_h_Right[3] = 0.2;
    m_h_Right[4] = 0.2;
}
```

Of course you could also write a function to calculate and populate the IR buffers, but this one is short enough to code by hand if you want.

The frequency response (Figure 8.29) of an MA filter is always an LPF. The more samples that are averaged, the more stop-band attenuation and the more zeros get inserted into the *z*-plane. For this filter there is a pair of complex conjugate zeros which produce two null points in the response.

You can see this is a pretty poor LPF compared to some of the optimal filter designs, but these filters can be very useful, for example to smooth the response of an audio detector output or to insert in the feedback path of a delay line module to gently roll off high frequencies on each pass through the delay. You can get better stop-band attenuation by increasing the order of the filter. This will also effectively pull in the corner frequency, which looks like about 2.5 kHz here. Experiment with higher-order MA filters, or add a slider to let the user adjust the window size. This exact filter already exists as a RackAFX module called a "Smoothing Filter" for you to experiment with.

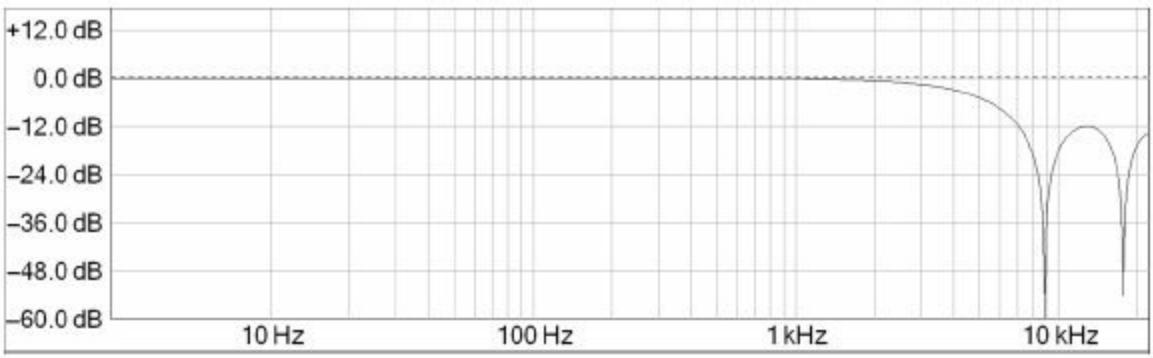


Figure 8.29: The MA filter with a window of 5 samples.

8.11.2 Lagrange Interpolator

Choose:

N = number of points to interpolate, minimum = 2

Calculate:

$$a_j = \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k} \tag{8.9}$$

The Lagrange interpolator uses a polynomial of order $j - 1$ to interpolate across the window of points that you give it. The window is of length j in the above equation. This is a complex filter because the coefficients change every sample period and they are based on the window of input values, x_0 to x_j. This filter can be implemented as a pure math function call. To facilitate your coding, a Lagrange interpolation function is implemented in your pluginconstants.h file:

```
/*
Function:  lagrpol()     implements n-order Lagrange Interpolation
Inputs:    double* x     Pointer to an array containing the x-coordinates of the
                         input values
           double* y     Pointer to an array containing the y-coordinates of the
                         input values
           int n         The order of the interpolator, this is also the length of
                         the x,y input arrays
           double xbar   The x-coorinates whose y-value we want to interpolate

Returns    The interpolated value y at xbar. xbar ideally is between the middle two
           values in the input array, but can be anywhere within the limits, which
           is needed for interpolating the first few or last few samples in a table
           with a fixed size.
*/
```

8.11.3 Median Filter

Choose:

N = number of points in window = odd

Calculate:

Acquire samples in windows of *N* values, then sort and choose the median value as the output.

The median filter (Figure 8.30) is a very interesting and somewhat strange algorithm. It has no IR or frequency response. It smoothes an input signal, which is an LPF type of operation, but it preserves transient edges, which is very un-LPF in nature. It has applications in noise reduction without losing high-frequency transients. Its central algorithm uses a sorting mechanism to sort the window of data by amplitude. The median value is chosen from the sort operation as the output. When the next sample arrives, the window is re-sorted and the next median value is obtained. To understand how it

smoothes a signal without affecting high-frequency transients, consider the following example.

Example: Design a five-point median filter and test with example.

Consider this input sequence: $x = \{1, 2, 1, 0, 1, 2, 3, 3, 2, 1, 9, 8, 9, 9, 7, 5, 5, 4\}$ in Figure 8.31.

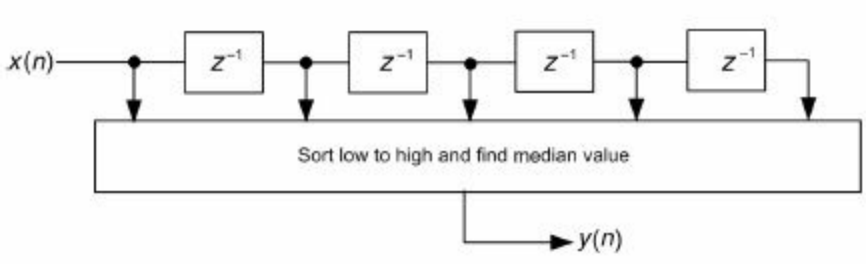


Figure 8.30: The block diagram of a five-point median filter implementation.

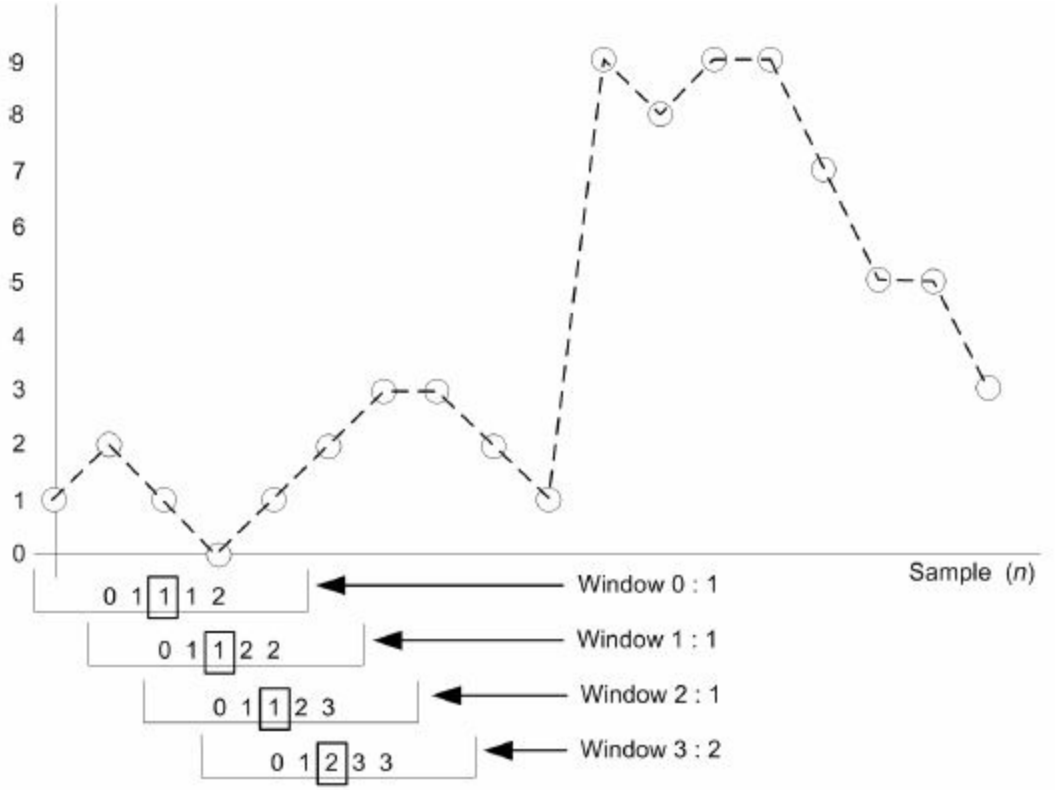


Figure 8.31: The first four windows of the median filter produce an output sequence {1,1,1,2}.

You can see a transient edge where the signal jumps from 1 to 9 and then another transient where it drops from 9 to 7 to 5. The first window operates on the first five samples and sorts them from low to high. Then, the median value is chosen as the output. The median value is shown in a box in the center of each window. You can see the smoothing effect immediately—the first three samples out of the filter are all 1, even though the first three samples vary from 1 to 2. Figures 8.32 and 8.33 show the result of median filtering the signal in Figure 8.31.

FIR filters can be complicated to design and long convolutions in direct form are slow. You can use the FIR design tools when you need to create linear-phase filters with very steep roll offs (optimal method) or filters with arbitrary frequency responses (frequency sampling method). You might also want to investigate other FIR designs such as the windowing method and the recursive frequency sampling method.

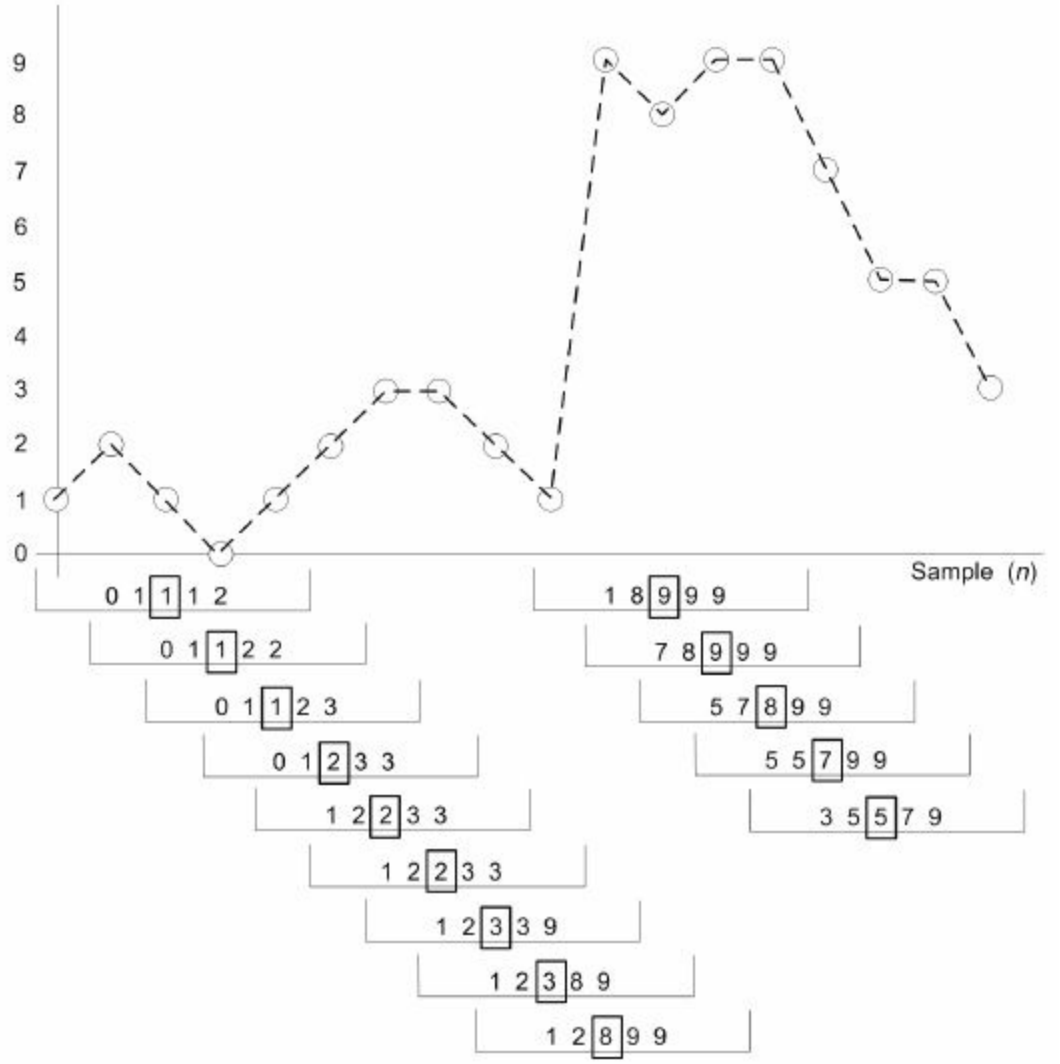


Figure 8.32: The complete set of median filter outputs for the sliding-window-of-five operation.

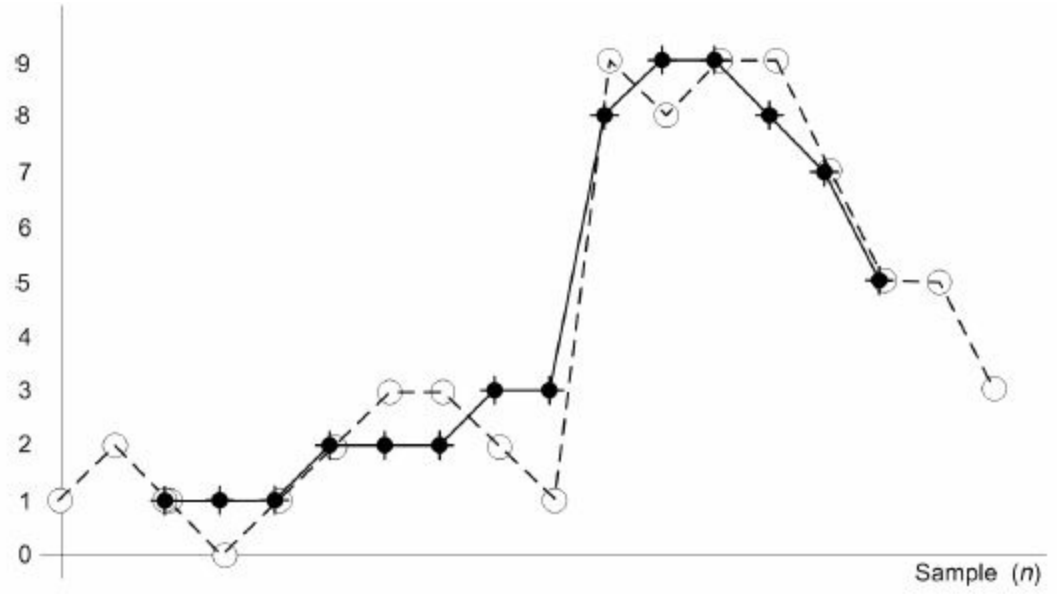


Figure 8.33: Input and output sequences plotted together and shifted to show the smoothing of the steady-state portions and preservation of the transient edges.

Bibliography

Ifeachor, E. C. and Jervis, B. W. 1993. *Digital Signal Processing, A Practical Approach*, Chapters 4 and 6. Menlo Park, CA:

Addison-Wesley.
Kwakernaak, H. and Sivan, R. 1991. *Modern Signals and Systems*, Chapters 3 and 9. Englewood Cliffs, NJ: Prentice-Hall.
Lindquist, C. 1999. *Adaptive and Digital Signal Processing*, Chapter 10. Miami: Steward & Sons.
Oppenheim, A. V. and Schafer, R. W. 1999. *Discrete-Time Signal Processing*, 2nd ed., Chapter 7. Englewood Cliffs, NJ: Prentice-Hall.

CHAPTER 9

Oscillators

Oscillators find several uses in audio effects and plug-ins. The obvious use is as an audio test signal like the one RackAFX provides on the main interface. Additive synthesis of musical sounds uses multiple sinusoidal oscillators at harmonic frequencies to create complex waveforms. Wavetable synthesis stores a periodic waveform in a table for interpolation and playback when the musician strikes a key or a MIDI message is sent. Low-frequency oscillators (LFOs) are used in the design of modulated delay lines and modulated filters. Oscillators broadly fall into two categories: direct calculation and table lookup. We desire oscillators that have several important features:

- Stability over a wide range of frequencies
- No aliasing
- Purity of sinusoid (low THD+N) for sinusoidal oscillators
- Quadrature phase outputs
- Simplicity of calculation

9.1 Direct Form Oscillator

We can make a sinusoidal oscillator by placing a pair of poles directly on the unit circle in the z-plane. This produces a sinusoid at the pole angle (or frequency). The radius of the pole is always 1.0. Using the fundamental digital signal processing (DSP) z-plane equations, we can directly write the transfer function and difference equations:

$$\begin{aligned} H(z) &= a_0\left[\frac{1}{1 + b_1z^{-1} + b_2z^{-2}}\right] \\ &= a_0\left[\frac{1}{1 - 2R\cos(\Theta)z^{-1} + R^2z^{-2}}\right] \\ y(n) &= 2\cos(\Theta)y(n-1) - y(n-2) \end{aligned} \tag{9.1}$$

Since the pole radius is 1.0, then the b_2 coefficient is 1.0 as well. The b_1 coefficient is then $-2\cos(\theta)$, where θ is the pole frequency from 0 to π.

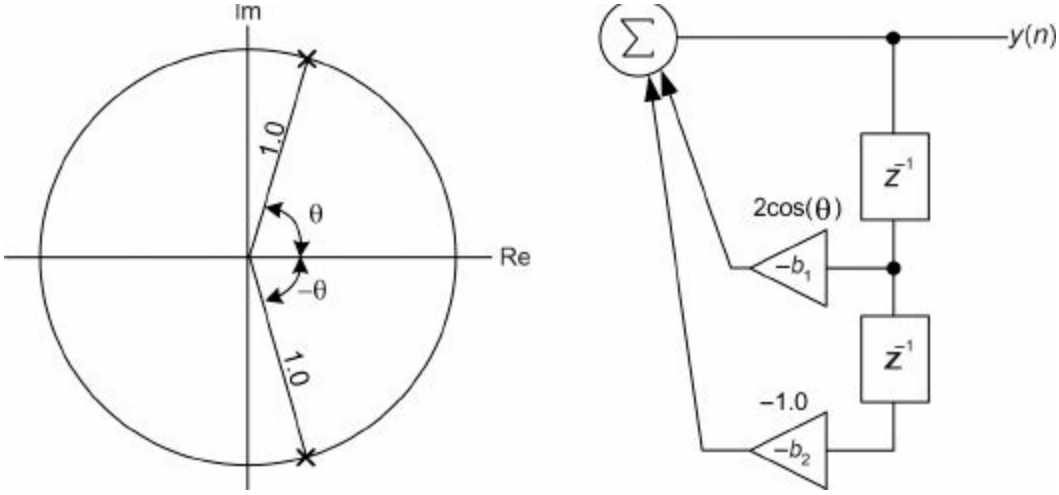


Figure 9.1: The direct form sinusoidal oscillator *z*-plane and block diagram.

You can see that the block diagram in Figure 9.1 has no input. Oscillators do not have inputs; instead they have *initial conditions* which, once started, will cause eternal oscillation. DSP theory suggests that kick-starting the system with an impulse will cause it to self-oscillate, which it does. However, the amplitude and phase are dependent on the frequency, θ. For an oscillator, we would like to precisely control the amplitude and starting phase so that they are exact. If we are trying to create a sinusoid with an initial phase offset of 0 degrees, then the first sample out of our oscillator at $n = 0$ would be 0.0, and the second sample out would be $\sin(\omega nT)$ where n 51 and $T = 1/f_s$.

9.1.1 Initial Conditions

Suppose we want to generate the above sinusoid, oscillating at 500 Hz. If we want the first sample out to be 0.0, then continue rising after that, we need to initialize the delays as if a sinusoid had been oscillating, up to the point when we turn on the oscillator. So we can set our initial state by preloading those last two samples into the delays in our oscillator/filter, as shown in Figure 9.2.

The sinusoid is oscillating at $\sin(\omega nT)$ where $n = 0, 1, 2, 3\ldots$ after the oscillator starts and T is the sample period. The previous two samples with $f_o = 500$ Hz and $f_s = 44{,}100$ Hz would be $\sin(\omega(-1T))$ and $\sin(\omega(-2T))$, which are –0.0712 and –0.1532, respectively. The direct form oscillator block diagram is shown in Figure 9.3.

The difference equation is as follows:

$$y(n) = -b_1 y(n-1) - b_2 y(n-2) \tag{9.2}$$

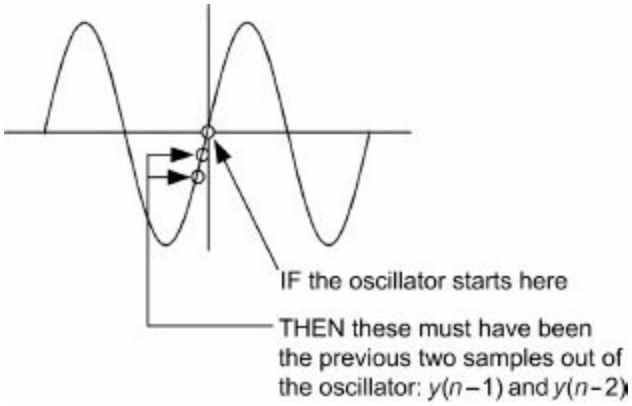


Figure 9.2: Initial conditions that would have produced a sinusoid whose first output sample is 0.0.

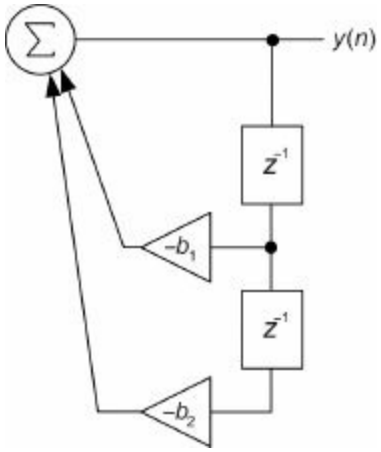


Figure 9.3: The direct form oscillator is really the feedback side of the bi-quad structure.

Specify:

- f_o, desired oscillation frequency

The design equations are as follows:

$$\theta = \frac{2\pi f_o}{f_s}$$
$$b_1 = -2\cos(\theta)$$
$$b_2 = 1.0 \tag{9.3}$$

Initial conditions:
$$y(n-1) = \sin(-1\theta)$$
$$y(n-2) = \sin(-2\theta)$$

9.2 Design a Direct Form Oscillator Plug-In

In our first version of the direct form oscillator we are going to make it as simple as possible by restarting the oscillator when the user changes the oscillator frequency. This means we are going to recalculate the initial conditions as if the oscillator was starting from a phase of 0 degrees and the first sample out would have a value of 0.0. After we have that up and running, we will modify it to change frequency on the fly, automatically back-calculating the initial conditions for any given output sample. Here are the oscillator's specifications:

- Monophonic sinusoidal oscillator.
- We will need to implement a second-order feed-back block.
- We will need a slider for the user to control the oscillation frequency in Hz.
- We will need a cookFrequency() function to calculate our coefficients and reset our initial conditions when the user moves the frequency slider.
- The oscillator's range will be 200 Hz to 6 kHz.

- We need to assign Start and Stop buttons.

9.2.1 Project: DirectOscillator

Create a new project in RackAFX. On the new project dialog, make sure you check the box "Output Only Synthesizer Plug-In"—you must select this for any synthesis plug-in; you will not be able to play a wave file through it (Figure 9.4).

9.2.2 DirectOscillator GUI

Add a frequency slider (Figure 9.5) to the user interface (UI) with the parameters in Table 9.1. Next, set up the assignable buttons to create start and stop functions. Right-click inside the box labeled "Assignable Buttons" but do not click on a button, only in the area in between. A dialog pops up (Figure 9.6) that lets you set the button names. Choose "Start," "Stop," and make the last one blank with " " by entering the strings in the edit boxes. The buttons will automatically use the text you enter. Do not check the "Latching" buttons. See the website for notes on how to use latching buttons.

Table 9.1: The GUI control for the DirectOscillator plug-in

Slider Property	Value
Control Name	Frequency
Units	Hz
Variable Type	float
Variable Name	m_fFrequency_Hz
Low Limit	200
High Limit	6000
Initial Value	1000

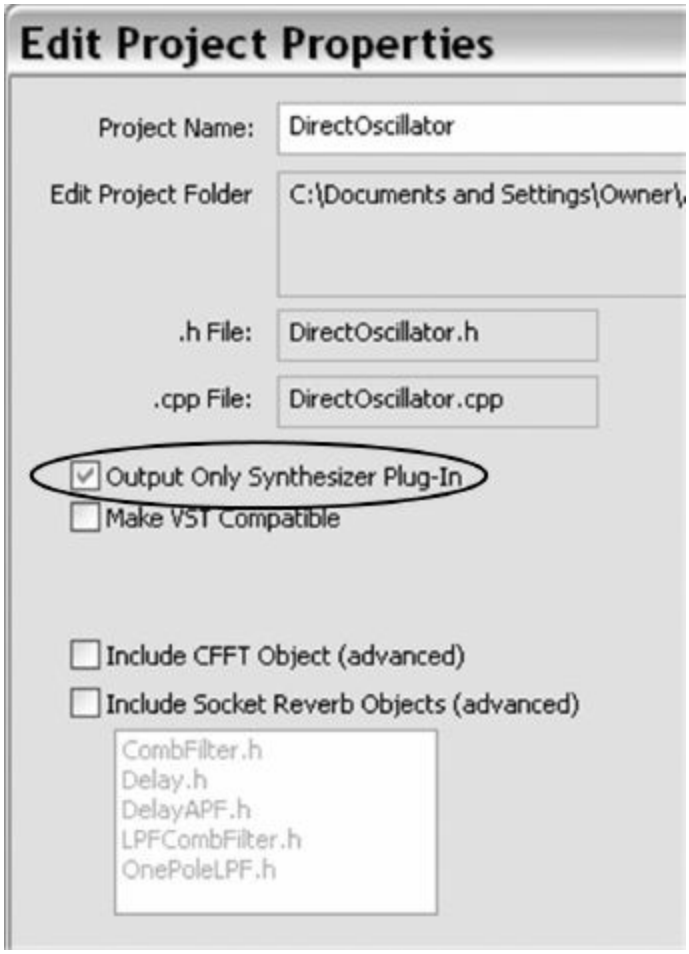


Figure 9.4: Make sure you check the "Output Only Synthesizer Plug-In" box for your oscillators.

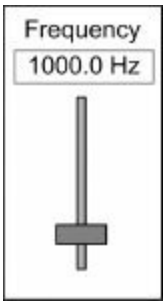


Figure 9.5: The DirectOscillator GUI.

These assignable buttons will trigger your userInterfaceChange() function with their nControlIndex values of 50, 51, and 52. See the userInterfaceChange() function for more details.



Figure 9.6: The assignable button setup; notice that the last button is blank. RackAFX will hide it when you load the plug-in.

9.2.3 DirectOscillator.h File

In the.h file, declare the variables you need to implement the oscillator:

```
// Add your code here: ------------------------------------------------------ //
//
// coefficients, 2nd Order FB
float m_f_b1;
float m_f_b2;

// delay elements, 2nd Order FB
float m_f_y_z1;
float m_f_y_z2;

// flag to start/stop oscillator
bool m_bRunOscillator;

// function to cook the Frequency and set initial conditions
void cookFrequency();
//
// END OF USER CODE --------------------------------------------------------- //
```

9.2.4 DirectOscillator.cpp File

Constructor

- Turn off the oscillator by setting the flag you declared in the.h file. You will need to do these two steps any time you make an oscillator or other output-only plug-in.

```
CDirectOscillator::CDirectOscillator()
{
	<SNIP SNIP SNIP>

	// Finish initializations here
	// turn off
	m_bRunOscillator = false;

	// set 0
	m_f_b1 = 0;
	m_f_b2 = 0;

	// flush memory
	m_f_y_z1 = 0;
	m_f_y_z2 = 0;

	// call the cooking function to calc coeffs
	// and set initial states
	cookFrequency();
}
```

Implement the cooking function according to the design equations:

```
void CDirectOscillator::cookFrequency()
{
	// Oscillation Rate = theta = wT = w/fs
	float f_wT = (2.0*pi*m_fFrequency_Hz)/(float)m_nSampleRate;

	// coefficients according to design equations
	m_f_b1 = -2.0*cos(f_wT);
	m_f_b2 = 1.0;

	// set initial conditions so that first sample out is 0.0
	m_f_y_z1 = sin(-1.0*f_wT);  // sin(wnT) = sin(w(-1)T)
	m_f_y_z2 = sin(-2.0*f_wT);  // sin(wnT) = sin(w(-2)T)
}
```

prepareForPlay()

- Call the cooking functions to initialize the oscillator:

```
bool __stdcall CDirectOscillator::prepareForPlay()
{
	// Add your code here:
	// calc coeffs and initial conditions
	cookFrequency();

	retur true;
}
```

processAudioFrame()

- Implement the second-order feed-back filter.
- Check the oscillator flag. If the oscillator is not running you *must* send 0.0 samples to the output stream and then return without further processing. This is done at the very beginning of the function.

```
bool __stdcall CDirectOscillator::processAudioFrame(float* pInputBuffer, float*
                                                    pOutputBuffer, UINT uNumChannels)
{
        //
        // output = input -- change this for meaningful processing
        //
        // Do LEFT (MONO) Channel
        //
        // if not running, write 0s and bail
        if(!m_bRunOscillator)
        {
                // zero it all out
                pOutputBuffer[0] = 0.0;

                // Mono-In, Stereo-Out (AUX Effect)
                if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                        pOutputBuffer[1] = 0.0;

                // Stereo-In, Stereo-Out (INSERT Effect)
                if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                        pOutputBuffer[1] = 0.0;

                return true;
        }
        // otherwise, do the oscillator
        // do difference equation y(n) = -b1y(n-2) - b2y(n-2)
        pOutputBuffer[0] = -m_f_b1*m_f_y_z1 - m_f_b2*m_f_y_z2;

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0];

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0];

        // shuffle memory
        m_f_y_z2 = m_f_y_z1;
        m_f_y_z1 = pOutputBuffer[0];

        return true;
}
```

userInterfaceChange()

- Call the frequency cooking function and implement the start and stop functions.
- Make sure your nControlIndex values match your switch/case statement.

```
bool __stdcall CDirectOscillator::userInterfaceChange(int nControlIndex)
{
	// cook variables
	switch(nControlIndex)
	{
	case 0: // freq
		cookFrequency();
		break;

	case 50: // start
		m_bRunOscillator = true;
		break;

	case 51: // stop
		m_bRunOscillator = false;
		break;
	default:
		break;
	}
	return true;
}
```

Build and load the dynamic link library (DLL). Then, hit the Start and Stop buttons to control your oscillator. Play with the frequency slider to change the value. You will notice a lot of clicking due to the discontinuities that occur each time the oscillator is reset to re-trigger from the 0.0 value. Other than that problem, the output should be a pure sinusoid with very low distortion and noise.

9.2.5 Improving the Oscillator Design

One of the drawbacks of oscillators is their initial conditions usually must be met (there are noise generators and other devices that can start up in a random state). In this case, we must preload the delay elements with samples that would have occurred before the initial output sample. If the user changes the frequency of oscillation, the initial states must change too. The problem is that the oscillator can be in any state when the user changes the controls. Suppose the user decreased the frequency of oscillation a bit and that the oscillator had some values in $y(n-1)$ and $y(n-2)$. Then to set up the new initial state, you can share the $y(n-1)$ value between the two sinusoids and calculate a new $y(n-2)$ value. The next sample out of the oscillator is at the new rate, using the new initial state you created (Figure 9.7).

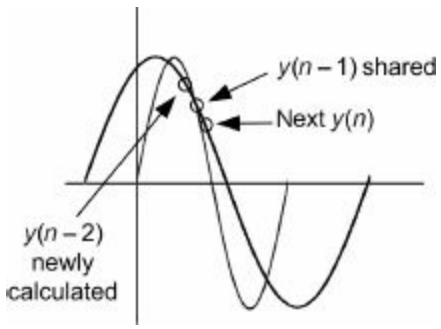


Figure 9.7: The new initial state samples, $y(n-1)$ and $y(n-2)$; we share the $y(n-1)$ sample as if it was from the new frequency and alter the $y(n-2)$ value to relocate it on the new sinusoid.

The initial conditions from the design equations are in Equation 9.4:

Initial conditions:

$$y(n-1) = \sin(\omega nT) \quad (9.4)$$
$$y(n-2) = \sin(\omega(n-1)T)$$

Our problem is that we do know the new frequency, ω, and since the sample rate is also known, then we also know ωT, but we don't know what sample interval n we are on. However, we can figure it out as follows:

- Take the inverse sin of the $y(n-1)$ delay element.
- Find the value of n by dividing it by ωT.
- arcsine() returns a value between $-\pi/2$ to $+\pi/2$ which is the rising edge of the sinusoid like the original initial conditions.
- If we are on the rising edge, calculate the new $y(n-2)$ value as $\sin((n-1)\omega T)$.
- If we are on the falling edge, calculate the new $y(n-2)$ value as $\sin((n+1)\omega T)$.

It is simple to change in the cookFrequency() function as follows, so make the change in your code. Note the use of commenting out the original initial condition code in case you want to refer to it later. Also note the logic to see if we are rising or falling. We need this because the arcsine() function only returns the angle for half a sinusoid. If you think about it, this makes sense. If you give it a value of 0.707, it can't tell if the angle is 45 degrees or 135 degrees, both of which produce a sine of 0.707.

```
void CDirectOscillator::cookFrequency()
{
	// Oscillation Rate = theta = wT = w/fs
	float f_wT = (2.0*pi*m_fFrequency_Hz)/(float)m_nSampleRate;

	// coefficients according to design equations
	m_f_b1 = -2.0*cos(f_wT);
	m_f_b2 = 1.0;

	// set initial conditions so that first sample out is 0.0
//	m_f_y_z1 = sin(-1.0*f_wT);  // sin(wnT) = sin(w(-1)T)
//	m_f_y_z2 = sin(-2.0*f_wT);  // sin(wnT) = sin(w(-2)T)

	// re calculate the new initial conditions
	// arcsine of y(n-1) gives us wnT
	double wnT1 = asin(m_f_y_z1);
	// find n by dividing wnT by wT
	float n = wnT1/f_wT;

	// re calculate the new initial conditions
	// asin returns values from -pi/2 to +pi/2 where the sinusoid
	// moves from -1 to +1 -- the leading (rising) edge of the
	// sinewave. If we are on that leading edge (increasing)
	// then we use the value 1T behind.
	//
	// If we are on the falling edge, we use the value 1T ahead
	// because it mimics the value that would be 1T behind
	if(m_f_y_z1 > m_f_y_z2)
		n-=1;
	else
		n+=1;

	// calculate the new (old) sample
	m_f_y_z2 = sin((n)*f_wT);
}
```

Build and load the DLL. Then, hit the Start and Stop buttons to control your oscillator. Play with the frequency slider to change the value. If you change the slider too much you will still hear a click on some of the discontinuities due to the fact that the frequencies are so far apart that it produces a noticeable shift in the waveform.

Now that you understand how to code a direct calculation oscillator in RackAFX, next are a few more example oscillator designs for you to try out. Remember that you must set the flag m_bOutputOnlyPlugIn to make the oscillators work properly in RackAFX. You must provide a start/stop mechanism as well. Sample code is included to help you along.

9.3 The Gordon–Smith Oscillator

The Gordon–Smith oscillator uses a pair of delay elements arranged in a circular configuration to produce both sin and cos oscillations. The two outputs are almost perfectly in quadrature phase (90 degrees apart) only differing by half a sample period. Its sinusoidal oscillation is pure enough to be used in audio test equipment. And, because each output only uses one delay element, there are no initial states to update when the frequency is changed; the single $y(n-1)$ sample is shared with the new frequency the same way as in the direct form oscillator. Only the coefficient e needs to be updated. A small amplitude variation is observed when the oscillation frequency changes, but it is

small enough to not cause clicks or pops in the output. It sounds just as smooth as the direct form oscillator when the frequency is adjusted slowly.

The two outputs are labeled $y(n)$ and $y_q(n)$ where the "q" stands for quadrature. Therefore, there are two difference equations. The difference equation for $y_q(n)$ must be solved first because $y(n)$ is dependent on it. A Gordon–Smith oscillator block diagram is shown in Figure 9.8.

The difference equations are as follows:

$$\begin{aligned} y_q(n) &= y_q(n-1) - \epsilon y(n-1) \\ y(n) &= \epsilon y_q + y(n-1) \end{aligned} \tag{9.5}$$

Specify:

- f_o, desired oscillation frequency

The design equations are as follows:

$$\begin{aligned} \theta &= \frac{2\pi f_o}{f_s} \\ \epsilon &= 2\sin(\theta/2) \end{aligned} \tag{9.6}$$

Initial conditions:

$$\begin{aligned} y(n-1) &= \sin(-1\theta) \\ y_q(n-1) &= \cos(-1\theta) \end{aligned}$$

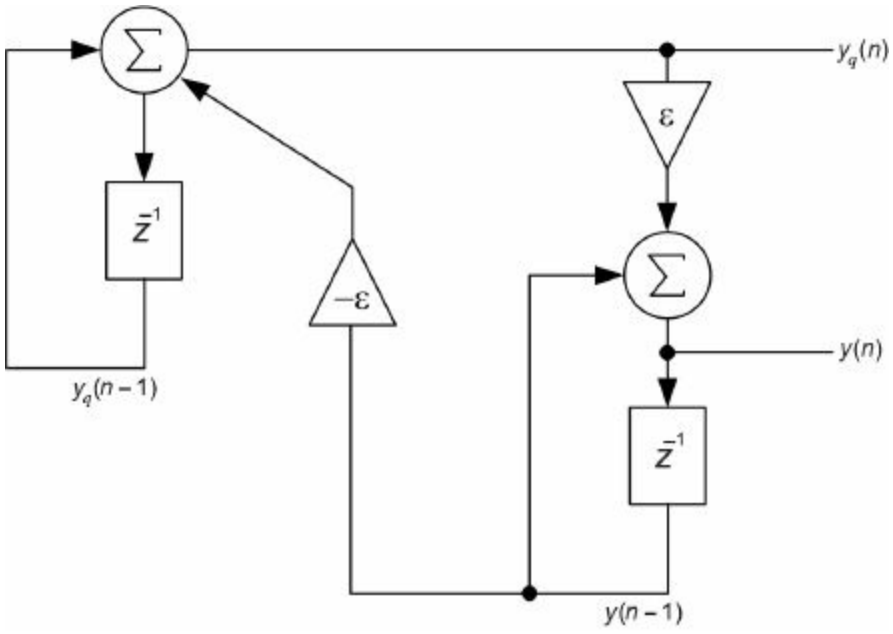


Figure 9.8: The Gordon–Smith Oscillator.

The C++ code for the Gordon–Smith oscillator looks as follows (two memory elements have been declared, m_f_yn_z and m_f_yq_z, as well as a coefficient m_fGorSmithEpsilon). In cookFrequency():

```
// calculate HS Epsilon
```

```
float f_wT = (2.0*pi*m_fFrequency_Hz)/(float)m_nSampleRate;
m_fGorSmithEpsilon = 2.0*sin(f_wT/2.0);
```

In processAudioFrame():

```
// form yq(n) first
float f_yqn = m_f_yq_z - m_fGorSmithEpsilon*m_f_yn_z;

// y(n)
float f_yn = m_fGorSmithEpsilon*f_yqn + m_f_yn_z;

// shuffle delays
m_f_yq_z = f_yqn;
m_f_yn_z = f_yn;

// write out
pOutputBuffer[0] = f_yn;

etc…
```

9.4 Wave Table Oscillators

A wave table oscillator is, as its name implies, a table-based system for creating periodic signals. A circular buffer is preloaded with one cycle, minus one sample, of a waveform. The waveform may be mathematical (sin, cos, pulse, square, saw, triangle, etc.) or it might be arbitrary or a recorded signal. The basic idea is that you read through the table and output samples from it. Consider a table of 1024 samples consisting of one cycle of a sinusoid, minus one sample (Figure 9.9).

Suppose you start at $i = 0$ and during each sample period, you read out one value and advance to the next. At the end of the buffer, you wrap around and start all over. If you did read out one value per sample period, what would be the resulting frequency of the waveform?

The answer is $f_{table} = f_s/L$ when the index increment is exactly 1.0 through the table. For a 1024-point wave table at a 44,100 Hz sample rate, the table frequency is 43.06 Hz. If you happen to really need a super precise sinusoid at exactly 43.06 Hz, then this method will produce nearly perfect results. The only factor is the precision of the sinusoid loaded into the table. If you had a saw-tooth waveform stored in the table, it too would have a fundamental frequency of 43.06 Hz.

Most likely, you are going to want to make any arbitrary frequency you like, perhaps with an emphasis on musical pitch frequencies. With the exception of the note A, these are going to be floating-point numbers with fractional parts. To make any frequency, you calculate the *inc* value with Equation 9.7:

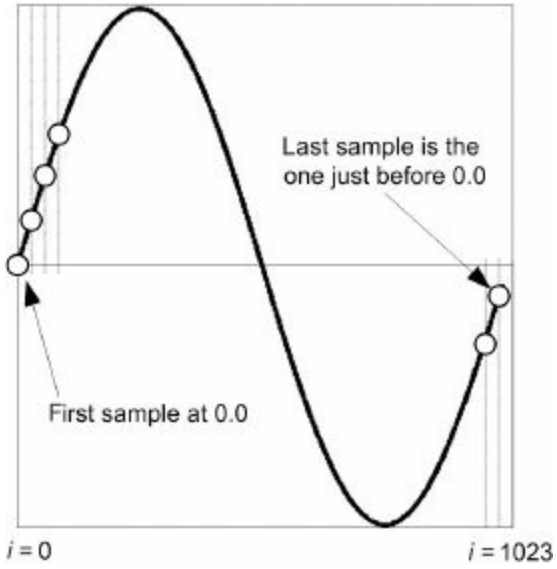


Figure 9.9: One cycle minus one sample of a sinusoid. The table is indexed with the value *i* which starts at 0; the last entry in the table is at *i* = 1023 and it is the sample just before the waveform starts all over again.

$$inc = L\frac{f_{\text{desired}}}{f_s} \tag{9.7}$$

L is the table length and f_{desired} is the target frequency. The increment value you get back will be used to skip through the table, moving forward by *inc* during each sample interval. If *inc* is less than 1.0, then the desired frequency is below the table frequency, and if it is above 1.0 it must be greater than the table frequency. If the *inc* is 2.0, then the resulting frequency is twice the table frequency. Most likely, the *inc* value is going to be noninteger and will therefore consist of an integer part and a fractional part. For example, if *inc* = 24.9836 then the integer part would be 24 and the fractional part 0.9836. Here, the integer part is called *int* and the fractional part is called *frac*.

As with the delay module you built in Chapter 8, there are several options for dealing with the fractional part of the increment value. You could:

- Truncate the value, and forget *frac*.
- Linearly interpolate the table *frac* distance between *int* and *int* + 1.
- Use polynomial interpolation or another interpolation method instead of linear interpolation.

If you truncate the *inc* value, then you have multiple problems—the note you synthesize won't be exactly in tune. Additionally, it will be distorted because of the inaccuracy in the transcription out of the table. Linear and polynomial interpolation both overcome these problems, though there is still distortion in the output. The industry standard is a fourth-order Lagrange interpolation on the wave table, where the neighboring four points (two to the left and two to the right) of the target interpolated value are used.

Before we code a wave table oscillator, there is one thing you need to be aware of: the danger of a wave table is that the cycle of data stored inside might be full of aliasing components. In other words, you can create a table of data that could have never made it past the input low-pass filter (LPF) if it were an analog signal being sampled. We will be doing just that and creating some signals that do

alias, on purpose. Then, we'll examine a way to synthesize common signals without aliasing. The only meaningful signal that won't alias is a sinusoid, so we can start there.

9.5 Design a Wave Table Oscillator Plug-In

To code a wave table oscillator you need to create a table of data and initialize it in the constructor. A floating-point index value is used for *inc* and it will keep track of the current read location. Linear interpolation will be used to extract samples from the table. The circular buffer will run as long as the oscillator is engaged. Thus, we need to start with a plug-in design similar to the direct form oscillator. We can use the assignable buttons to trigger the oscillator to simulate MIDI note-on and note-off messages, and we can provide a slider for frequency control.

9.5.1 Project: WTOscillator

Create a new RackAFX project and make sure to check the synthesizer plug-in box (if you forget, you can always change it later).

9.5.2 WTOscillator GUI

The final GUI will feature a frequency control and multiple radio button controls for waveform, mode, and polarity. Ultimately, it will have the controls shown in Figure 9.10.

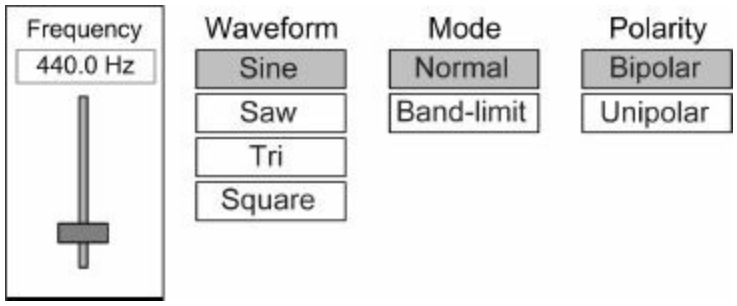


Figure 9.10: The final WTOscillator GUI.

Table 9.2: GUI controls for the wave table oscillator plug-in.

Slider Property	Value
Control Name	Frequency
Units	Hz
Variable Type	float
Variable Name	m_fFrequency_Hz
Low Limit	25
High Limit	4200
Initial Value	440

First, add a frequency slider to the UI and connect it to a variable named m_fFrequency_ Hz with the limits 25 Hz to 4.2 kHz and an initial setting of 440 Hz (Table 9.2). The limits are chosen as such because they are close to the lower and upper fundamental frequencies of the notes on a standard (88 key) piano or synthesizer. Next, set up the assignable buttons to create start and stop functions just like you did for the DirectOscillator plug-in.

9.5.3 WTOscillator.h File

Declare the variables you need to implement the oscillator:

```
// Add your code here: ------------------------------------------------------ //
//
// Array for the Table
float m_SinArray[1024];        // 1024 Point Sinusoid

// current read location
float m_fReadIndex;            // NOTE its a FLOAT!

// reset the read index
void reset()
{
     m_fReadIndex = 0.0;
}

// our inc value
float m_f_inc;

// our cooking function
void cookFrequency();

// our note on/off message
bool m_bNoteOn;

// END OF USER CODE ---------------------------------------------------------- //
```

You can see that we've added the necessary ingredients (array, read index, *inc*, note-on/off), as well as two functions:

- cookFrequency() to update the *inc* value when the frequency changes.
- reset(), which just relocates the read index to the top of the buffer.

9.5.4 WTOscillator.cpp File

Constructor

- Turn off the oscillator by setting the flag you declared in the.h file.

```
CWTOscillator::CWTOscillator()
{
	<SNIP SNIP SNIP>
	// Finish initializations here
	// setup array
	for(int i = 0; i < 1024; i++)
	{
		// sample the sinusoid, 1024 points
		// sin(wnT) = sin(2pi*i/1024)
		m_SinArray[i] = sin( ((float)i/1024.0)*(2*pi) );
	}

	// clear variables
	m_fReadIndex = 0.0;
	m_f_inc = 0.0;

	// silent
	m_bNoteOn = false;

	// initialize inc
	cookFrequency();
}
```

In the constructor, you set the flags and set up the wave table by sampling it for 1024 points. This version is set up to produce the one-cycle-minus-one-sample waveform we desire. Write the cooking function:

```
void CWTOscillator::cookFrequency()
{
	// inc = L*fd/fs
	m_f_inc = 1024.0*m_fFrequency_Hz/(float)m_nSampleRate;
}
```

prepareForPlay()

- Reset the oscillator.
- Cook the variables.

```
bool __stdcall CWTOscillator::prepareForPlay()
{
	// Add your code here:
	// reset the index
	reset();

	// cook curent frequency
	cookFrequency();

	return true;
}
```

userInterfaceChange()

- Handle the slider control to cook the variables.
- Handle the start/stop buttons to turn the flag on/off and cook variables if turning on.
- Make sure your nControlIndex variables match with your GUI controls.

```
bool __stdcall CWTOscillator::userInterfaceChange(int nControlIndex)
{
	// add your code here
	switch(nControlIndex)
	{
	case 0:
		cookFrequency();
		break;

	// note on
	case 50:
		reset();
		cookFrequency();
		m_bNoteOn = true;
		break;
	// note off
	case 51:
		m_bNoteOn = false;
		break;

	default:
		break;
	}
	return true;
}
```

processAudioFrame()

- Implement the table look up.
- Notice that as with the delay line, we shift the frame of reference of the interpolated points to be between $n = 0$ and $n = 1$, then just use the *frac* component to find the value between the samples.

```
bool __stdcall CWTOscillator::processAudioFrame(float* pInputBuffer, float*
		pOutputBuffer, UINT uNumChannels, UINT uNumOutputChannels)
{
	// Do LEFT (MONO) Channel
	// if not running, write 0s and bail
	if(!m_bNoteOn)
	{
		pOutputBuffer[0] = 0.0;

		// Mono-In, Stereo-Out (AUX Effect)
		if(uNumInputChannels == 1 && uNumOutputChannels == 2)
			pOutputBuffer[1] = 0.0;

		// Stereo-In, Stereo-Out (INSERT Effect)
		if(uNumInputChannels == 2 && uNumOutputChannels == 2)
			pOutputBuffer[1] = 0.0;

		return true;
	}
	// our output value for this cycle
	float fOutSample = 0;

	// get INT part
	int nReadIndex = (int)m_fReadIndex;

	// get FRAC part
	float fFrac = m_fReadIndex - nReadIndex;
```

```
        // setup second index for interpolation; wrap the buffer if needed
        int nReadIndexNext = nReadIndex + 1 > 1023 ? 0 : nReadIndex + 1;

        // interpolate tht output (x1,x2,y1,y2,frac) - notice the way we set x1 and x2
        // to 0 and 1, then only use the frac value while nReadIndex and nReadIndexNext
        // acquire the values from the table
        fOutSample = dLinTerp(0, 1, m_SinArray[nReadIndex], m_SinArray[nReadIndexNext],
                              fFrac);

        // add the increment for next time
        m_fReadIndex += m_f_inc;

        // check for wrap
        if(m_fReadIndex > 1024)
                m_fReadIndex = m_fReadIndex - 1024;

        // write out
        pOutputBuffer[0] = fOutSample;

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = fOutSample;

        // Stereo-In, Stereo-Out (INSERT Effect)

        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                pOutputBuffer[1] = fOutSample;

        return true;
}
```

Build and run the oscillator and check the pitch using the analyzer window. Adjust the oscillation frequency with the slider. Add a volume control slider if you wish.

9.6 Adding More Wave Tables

The sinusoidal oscillator is interesting, but you can implement any kind of table you want. You will see that adding more tables is actually easy since they all use the same frequency of oscillation equation. In this section we will add the following table types:

- Saw-tooth
- Triangle
- Square

These three waveforms will be mathematically as close as you can get in a discrete time system, but they will alias when played back. However, we still want to implement them because we can use the wave table oscillator as an LFO for upcoming effects like chorus and flanger. To add more tables we need to do the following:

- Add more arrays to store the tables.
- Initialize the arrays in the constructor.
- Provide the user with a way to switch oscillator types.

9.6.1 WTOscillator.h File

Start with the new arrays in the .h file:

```
// Add your code here: -----------------------------------------------------
//
// Array for the Table
float m_SinArray[1024];             // 1024 Point Sinusoid
float m_SawtoothArray[1024];        // saw
float m_TriangleArray[1024];        // tri
float m_SquareArray[1024];          // sqr

// current read location
float m_fReadIndex;                 // NOTE its a FLOAT!
```

9.6.2 WTOscillator.cpp File

Constructor

- These three new wave tables will be bipolar like the sinusoid table and swing between –1.0 and +1.0, and they will all start with the first sample = 0.0.

```
CWTOscillator::CWTOscillator()
{
	m_PlugInName = "WTOscillator";

	// set our oscillator flag:
	this->m_bOutputOnlyPlugIn = true;

	// slope and y-intercept values for the
	// Triangle Wave
	// rising edge1:
	float mt1 = 1.0/256.0;
	float bt1 = 0.0;

	// rising edge2:
	float mt2 = 1.0/256.0;
	float bt2 = -1.0;

	// falling edge:
	float mtf2 = -2.0/512.0;
	float btf2 = +1.0;

	// Sawtooth
	// rising edge1:
	float ms1 = 1.0/512.0;
	float bs1 = 0.0;

	// rising edge2:
	float ms2 = 1.0/512.0;
	float bs2 = -1.0;
```

```
// setup arrays
for(int i = 0; i < 1024; i++)
{
        // sample the sinusoid, 1024 points
        // sin(wnT) = sin(2pi*i/1024)
        m_SinArray[i] = sin( ((float)i/1024.0)*(2*pi) );

        // saw, triangle and square are just logic mechanisms
        // can you figure them out?

        // Sawtooth
        m_SawtoothArray[i] =i < 512 ? ms1*i + bs1 : ms2*(i-511) + bs2;

        // Triangle
        if(i < 256)
                m_TriangleArray[i] = mt1*i + bt1; // mx + b; rising edge 1
        else if (i >= 256 && i < 768)
                m_TriangleArray[i] = mtf2*(i-256) + btf2; // mx + b; falling edge
        else
                m_TriangleArray[i] = mt2*(i-768) + bt2; // mx + b; rising edge 2

        // square:



        m_SquareArray[i] = i < 512 ? +1.0 : -1.0;
}

etc…
```

9.6.3 WTOscillator GUI

In order to let the user adjust the table, you can use the enumerated unsigned integer type (UINT) variable in the slider or the radio buttons. I will use the radio buttons and set them up here—right-click in the area around the buttons you want to label and fill in the properties with Table 9.3. The enumerated UINT variable lets you make the switch between oscillators. You set up the UINT variable name and the string of labels for the radio buttons here.

Table 9.3: The button properties for the oscillator type control.

Slider Property	Value
Control Name	Osc Type
Units	
Variable Type	enum
Variable Name	m_uOscType
enum string	sine, saw, tri, square

Note: Here, we use a radio button list with an enumerated string.

9.6.4 WTOscillator.h File

In the .h file, you can see where RackAFX added the variable:

```
// ADDED BY RACKAFX -- DO NOT EDIT THIS CODE!!! ------------------------------//
// **--0x07FD--**

float m_fFrequency_Hz;
UINT m_uOscType;
enum{sine,saw,tri,square};

// **--0x1A7F--**
// ------------------------------------------------------------------------ //
```

The m_uOscType is the switch variable and the enumerated list {sin, saw, tri, square} represents the integer set {0, 1, 2, 3}, which maintains the state of the switch.

9.6.5 WTOscillator.cpp File

processAudioFrame()

- A switch/case statement is used to choose the table to interpolate. Every other facet of the operation is the same as before, only the table is different.

```
bool __stdcall CDirectOscillator::processAudioFrame(float* pInputBuffer, float*
                    pOutputBuffer, UINT uNumChannels, UINT uNumOutputChannels)
{
       // Do LEFT (MONO) Channel
       <SNIP SNIP SNIP>
       // setup second index for interpolation; wrap the buffer if needed
       int nReadIndexNext = nReadIndex + 1 > 1023 ? 0 : nReadIndex + 1;

       // interpolate the output
       switch(m_uOscType)
       {
              case sine:
                     fOutSample = dLinTerp(0, 1, m_SinArray[nReadIndex],
                                   m_SinArray[nReadIndexNext], fFrac);
                     break;

              case saw:
                     fOutSample = dLinTerp(0, 1, m_SawtoothArray[nReadIndex],
                                   m_SawtoothArray[nReadIndexNext], fFrac);
                     break;
              case tri:
                     fOutSample = dLinTerp(0, 1, m_TriangleArray[nReadIndex],
                                   m_TriangleArray[nReadIndexNext], fFrac);
                     break;
              case square:
                     fOutSample = dLinTerp(0, 1, m_SquareArray[nReadIndex],
                                   m_SquareArray[nReadIndexNext], fFrac);
                     break;
              // always need a default
              default:
                     fOutSample = dLinTerp(0, 1, m_SinArray[nReadIndex],
                                   m_SinArray [nReadIndexNext], fFrac);
                     break;
       }
       // add the increment for next time
       m_fReadIndex += m_f_inc;
       <SNIP SNIP SNIP>
       return true;
}
```

Build and test the oscillator. Use the analyzer window to view the waveforms in time and frequency. Notice the large amount of upper harmonics in the non-sinusoidal waveforms—some of those

harmonics are actually aliased backwards. We'll need a way to get around the aliasing if we want to synthesize these waveforms for listening; however, we can still use this oscillator as a very effective LFO if we want.

9.7 Band-Limited Additive Wave Tables

If you looked at the analyzer's fast Fourier transform (FFT) window you saw an abundance of upper harmonics in the saw-tooth, triangle, and square waves. That is because mathematically, these waveforms contain harmonics out to infinite frequency. Equations 9.8 through 9.10 show the Fourier series equations for these waveforms.

$$\begin{aligned} y(n)_{SAW} &= \sum_{k=1}^{\infty}(-1)^{k+1}\frac{1}{k}\sin(k\omega nT) \\ &= \left[\sin(\omega nT) - \frac{1}{2}\sin(2\omega nT) + \frac{1}{3}\sin(3\omega nT) - \frac{1}{4}\sin(4\omega nT) + \ldots\right] \end{aligned} \tag{9.8}$$

The saw-tooth waveform has both even and odd harmonics scaled according to $(1/k)$: ∞_1

$$\begin{aligned} y(n)_{TRI} &= \sum_{k=0}^{\infty}(-1)^{k}\frac{1}{(2k+1)^2}\sin((2k+1)\omega nT) \\ &= \left[\sin(\omega nT) - \frac{1}{9}\sin(3\omega nT) + \frac{1}{25}\sin(5\omega nT) - \frac{1}{49}\sin(7\omega nT) + \ldots\right] \end{aligned} \tag{9.9}$$

The triangle waveform has only odd harmonics. The $(-1)^k$ term alternates the signs of the harmonics. The harmonic amplitudes drop off at a rate given by $1/(2k+1)^2$ which is exponential in nature.

$$\begin{aligned} y(n)_{SQUARE} &= \sum_{k=0}^{\infty}\frac{1}{2k+1}\sin((2k+1)\omega nT) \\ &= \left[\sin(\omega nT) + \frac{1}{3}\sin(3\omega nT) + \frac{1}{5}\sin(5\omega nT) + \frac{1}{7}\sin(7\omega nT) + \ldots\right] \end{aligned} \tag{9.10}$$

The square wave is also composed of odd harmonics like the triangle wave. The harmonic amplitudes drop off at a rate of $1/(2k+1)$, which is not as severe as the triangle wave. Therefore, the square wave has higher-sounding harmonics and a more gritty texture.

Using these Fourier series equations, you can implement Fourier synthesis (or additive synthesis) to build up a waveform that is band-limited to Nyquist. Let's modify the existing wave table oscillator to optionally use band-limited tables instead. The first version will be somewhat primitive, but will be a beginning source that you can use to make much more advanced synthesizer oscillators. This version will also span the frequencies of the piano.

The highest note on a piano is C8, vibrating at about 4186 Hz. Five times this value is 20,930 Hz, which is just barely below Nyquist. This means that we can synthesize our tables with the fundamental plus the next five harmonics (also called partials) of the series given in Equations 9.8 through 9.10 and create our band-limited tables. The tables will therefore potentially contain the fundamental, 2nd, 3rd, 4th, 5th, and 6th harmonics. In order to do this, you need to modify the plug-in to allow for another mode: normal or band-limited. You can do this with another enumerated UINT

data type, using either a slider or radio-button bank. I will use another radio button bank in this example.

9.7.1 WTOscillator GUI

Start with the UI and add another enumerated UINT by right-clicking inside the second bank of radio buttons and filling out the properties in Table 9.4.

Table 9.4: The button properties for the mode switches between "normal" and "bandlimited" on the UI.

Slider Property	Value
Control Name	Osc Type
Units	
Variable Type	enum
Variable Name	m_uOscType
enum string	normal, bandlimit

9.7.2 WTOscillator.h File

This new information appears in the .h file:

```
// ADDED BY RACKAFX -- DO NOT EDIT THIS CODE!!! ------------------------------ //
// **--0x07FD--**

float m_fFrequency_Hz;
UINT m_uOscType;
enum{sine,saw,tri,square};
UINT m_uTableMode;
enum{normal,bandlimit};

// **--0x1A7F--**
// ------------------------------------------------------------------------ //
```

As long as we are in the .h file, we need to add more arrays for our band-limited tables. We want to keep these separate from the other tables to provide both modes of operation.

```
// Add your code here: ------------------------------------------------------ //
//
// Array for the Table

float m_SinArray[1024];                  // 1024 Point Sinusoid
float m_SawtoothArray[1024];             // saw
float m_TriangleArray[1024];             // tri
float m_SquareArray[1024];               // sqr

// band limited to 5 partials
float m_SawtoothArray_BL5[1024];         // saw, BL = 5
float m_TriangleArray_BL5[1024];         // tri, BL = 5
float m_SquareArray_BL5[1024];           // sqr, BL = 5

// current read location
float m_fReadIndex;                      // NOTE its a FLOAT!
```

9.7.3 WTOscillator.cpp File

Constructor

- Initialize the tables according to the Fourier series equations. One of the problems with the additive method is that the final tables may be slightly outside the range of (−1.0 to +1.0), so we need to make normalized tables by saving the maximum value, then dividing each table by its maximum.

Here is the code for the five-partial tables:

```
CWTOscillator::CWTOscillator()
{
	<SNIP SNIP SNIP>

	// rising edge2:
	float ms2 = 1.0/512.0;
	float bs2 = -1.0;

	// setup arrays
	// to keep track of max-es for normalization later
	float fMaxTri = 0;
	float fMaxSaw = 0;
	float fMaxSqr = 0;

	for(int i = 0; i < 1024; i++)
	{
		// sample the sinusoid, 1024 points
		// sin(wnT) = sin(2pi*i/1024)
		m_SinArray[i] = sin( ((float)i/1024.0)*(2*pi) );

		<SNIP SNIP SNIP>

		// square: 1st sample is zero!
		if(i==1)
			m_SquareArray[i] = 0.0;
```

```
		else
			m_SquareArray[i] = i < 512 ? +1.0 : -1.0;
		// zero to start, then loops build the rest
		m_SawtoothArray_BL5[i] = 0.0;
		m_SquareArray_BL5[i] = 0.0;
		m_TriangleArray_BL5[i] = 0.0;

		// sawtooth: += (-1)^g+1(1/g)sin(wnT)
		for(int g=1; g<=6; g++)
		{
			double n = double(g);
			m_SawtoothArray_BL5[i] += pow((float)-1.0,(float)(g+1))*
					(1.0/n)*sin(2.0*pi*i*n/1024.0);
		}

		// triangle: += (-1)^g(1/(2g+1+^2)sin(w(2n+1)T)
		// NOTE: the limit is 3 here because of the way the sum is constructed
		// (look at the (2n+1) components
		for(int g=0; g=3; g++)
		{
			double n = double(g);
			m_TriangleArray_BL5[i] += pow((float)-1.0, (float)n)*
					(1.0/pow((float)(2*n + 1),
					(float)2.0))*
					sin(2.0*pi*(2.0*n + 1)*i/1024.0);
		}

		// square: += (1/g)sin(wnT)
		for(int g=1; g<=5; g+=2)
		{
			double n = double(g);
			m_SquareArray_BL5[i] += (1.0/n)*sin(2.0*pi*i*n/1024.0);
		}
		// store the max values
		if(i == 0)
		{
			fMaxSaw = m_SawtoothArray_BL5[i];
			fMaxTri = m_TriangleArray_BL5[i];
			fMaxSqr = m_SquareArray_BL5[i];
		}
		else
		{
			// test and store
			if(m_SawtoothArray_BL5[i] > fMaxSaw)
				fMaxSaw = m_SawtoothArray_BL5[i];

			if(m_TriangleArray_BL5[i] > fMaxTri)
				fMaxTri = m_TriangleArray_BL5[i];


			if(m_SquareArray_BL5[i] > fMaxSqr)
				fMaxSqr = m_SquareArray_BL5[i];
		}
	}
	// normalize the bandlimited tables
	for(int i = 0; i < 1024; i++)
	{
		// normalize it
		m_SawtoothArray_BL5[i] /= fMaxSaw;
		m_TriangleArray_BL5[i] /= fMaxTri;
		m_SquareArray_BL5[i] /= fMaxSqr;
	}
	etc…
}
```

Because each sample in the table is built additively, you first zero out the value, then enter the loops

to accumulate the additive signal in the sample slot. To increase the number of partials, you just increase the limit in the for() loops above.

processAudioFrame()

- A switch/case statement is used to change between normal and band-limited versions.

```
bool __stdcall CWTOscillator::processAudioFrame(float* pInputBuffer, float*
                                        pOutputBuffer, UINT uNumInputChannels,
                                        UINT uNumOutputChannels)
{
      // SNIP SNIP SNIP>

      // setup second index for interpolation; wrap the buffer if needed
      int nReadIndexNext = nReadIndex + 1 > 1023 ? 0 : nReadIndex + 1;

      // interpolate the output
      switch(m_uOscType)
      {
            case sine:
                  fOutSample = dLinTerp(0, 1, m_SinArray[nReadIndex],
                                        m_SinArray[nReadIndexNext], fFrac);
                  break;

            case saw:
                  if(m_uTableMode == normal)     // normal
                       fOutSample = dLinTerp(0, 1, m_SawtoothArray[nReadIndex],
                                             m_SawtoothArray[nReadIndexNext],
                                             fFrac);
                  else                           // bandlimited
                       fOutSample = dLinTerp(0, 1,
                                        m_SawtoothArray_BL5[nReadIndex],
                                        m_SawtoothArray_BL5[nReadIndexNext],
                                        fFrac);
                  break;
```

```
            case tri:
                if(m_uTableMode == normal)   // normal
                    fOutSample = dLinTerp(0, 1, m_TriangleArray[nReadIndex],
                                          m_TriangleArray[nReadIndexNext],
                                          fFrac);
                else                         // bandlimited
                    fOutSample = dLinTerp(0, 1,
                                          m_TriangleArray_BL5[nReadIndex],
                                          m_TriangleArray_BL5[nReadIndexNext],
                                          fFrac);
                break;
            case square:
                if(m_uTableMode == normal)   // normal
                    fOutSample = dLinTerp(0, 1, m_SquareArray[nReadIndex],
                                          m_SquareArray[nReadIndexNext],
                                          fFrac);
                else                         // bandlimited
                    fOutSample = dLinTerp(0, 1, m_SquareArray_BL5[nReadIndex],
                                          m_SquareArray_BL5[nReadIndexNext],
                                          fFrac);
                break;

            // always need a default
            default:
                fOutSample = dLinTerp(0, 1, m_SinArray[nReadIndex],
                                      m_SawtoothArray[nReadIndexNext], fFrac);
                break;
        }
        // add the increment for next time
        m_fReadIndex += m_f_inc;
        etc…
}
```

Build and test the oscillator. Switch between the different modes and listen to the results, then use the analyzer to compare the oscillators in frequency and time. Next are some screenshots showing the differences for the saw-tooth and square waves.

9.7.4 Saw-Tooth

Figure 9.11 shows a normal saw-tooth waveform. Figure 9.12 shows a band-limited saw-tooth waveform.

9.7.5 Square Wave

Figure 9.13 shows a normal square wave. Figure 9.14 shows a band-limited square wave.

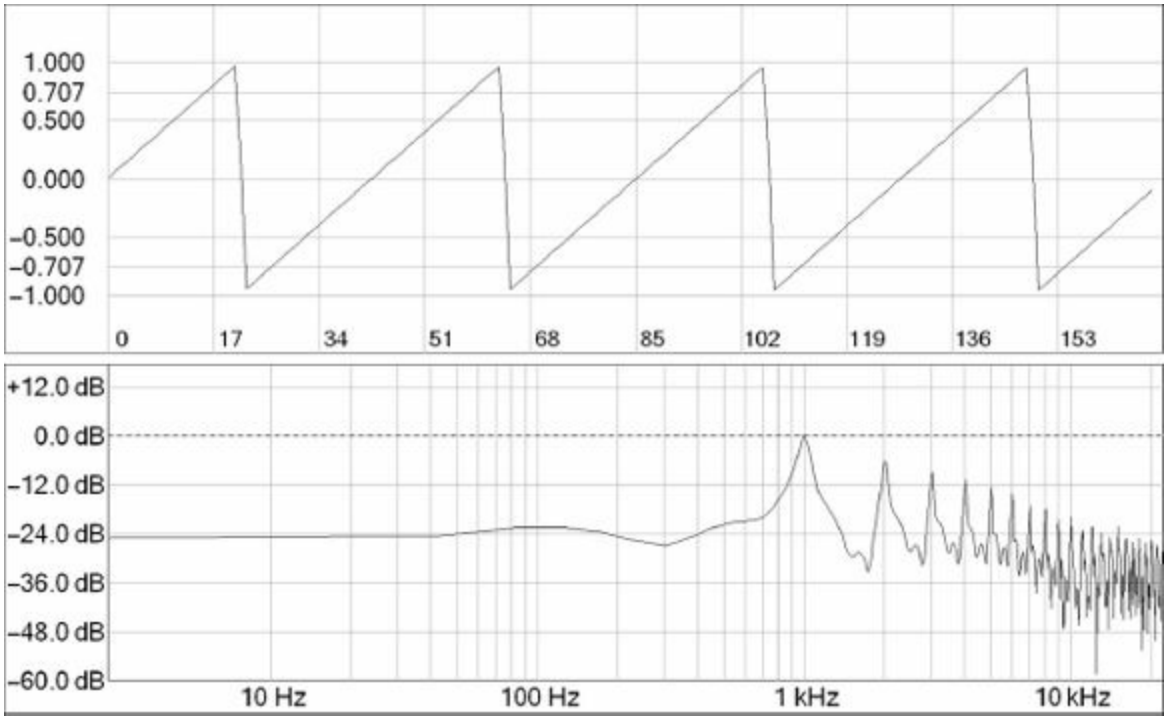


Figure 9.11: The mathematically perfect saw-tooth waveform with $f = 1$ kHz in the time domain (top) and frequency domain (bottom), which shows aliasing in the frequencies between 10 kHz and Nyquist.

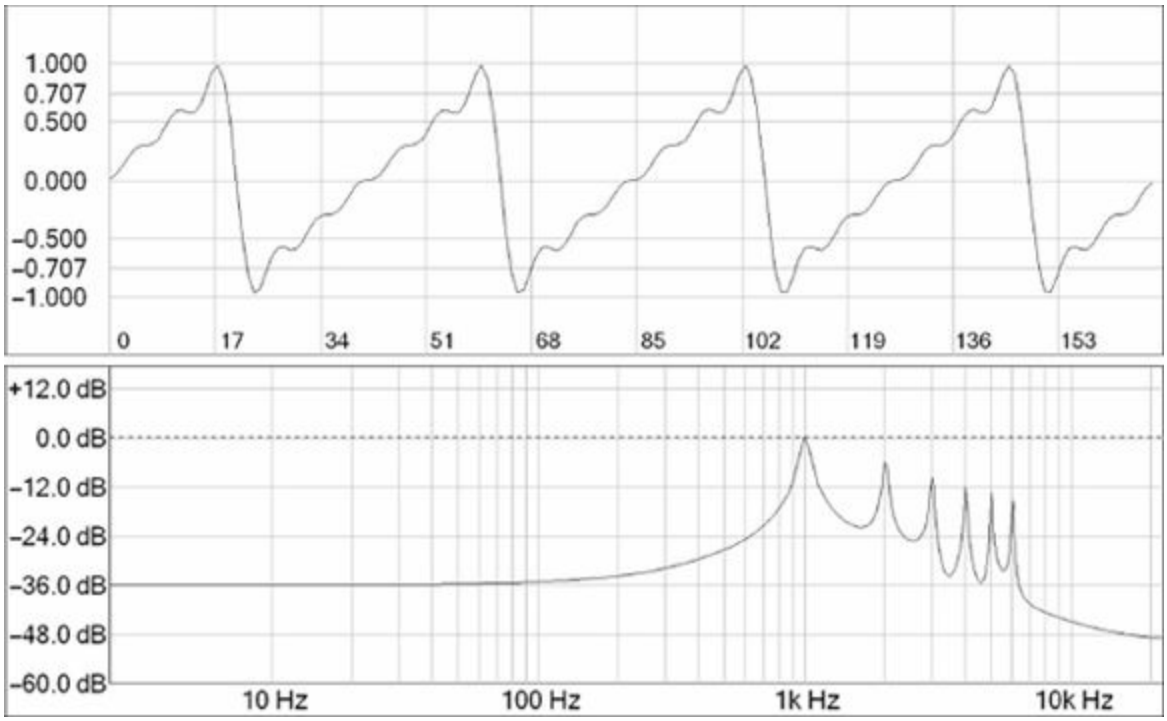


Figure 9.12: The five-harmonic band-limited saw-tooth waveform with $f = 1$ kHz in the time domain (top) and frequency domain (bottom). The aliasing is gone for this 1 kHz signal; it would reappear when the frequency was raised to the point that the highest harmonic went outside the Nyquist boundary. The fundamental plus the five harmonic peaks are clearly visible.

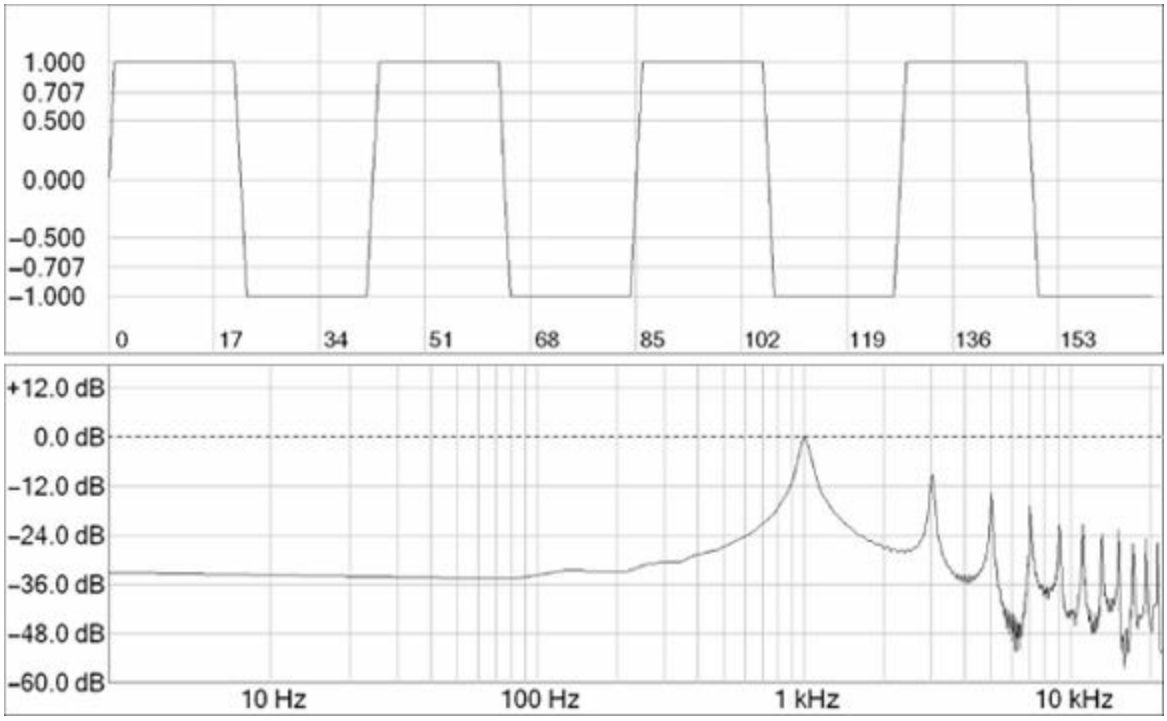


Figure 9.13: The mathematically perfect square wave with $f = 1$ kHz in the time domain (top) and frequency domain (bottom), which shows aliasing in the frequencies between 10 kHz and Nyquist. The square wave contains only the odd harmonics.

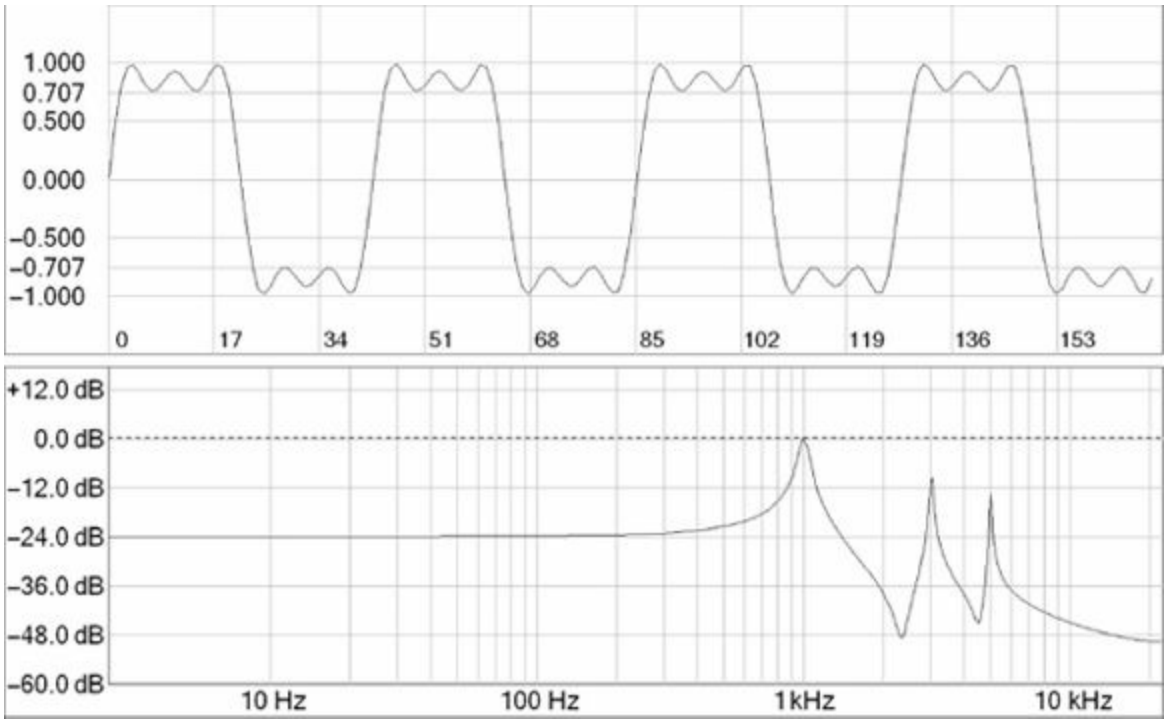


Figure 9.14: The five-harmonic band-limited square wave with $f = 1$ kHz in the time domain (top) and frequency domain (bottom). The aliasing is gone for this 1 kHz signal. There are only two harmonics (the third and fifth) because the next harmonic would be outside our limit.

9.8 Additional Oscillator Features (LFO)

For the modulated delay lines in the next chapters, we will need to use LFOs with a couple of additional properties. Specifically, we need

- A quadrature phase output
- Unipolar or bipolar operation
- Option to invert the output by 180 degrees
- A doOscillate() function that returns both outputs

All of these are easy to add to the current project. To make a quadrature phase output, we only need to declare a second read index which samples the table along with the current one. The quad phase read index will be initialized one-quarter of the way through the table to set up the quad phase output (Figure 9.15).

9.8.1 WTOscillator.h File

- Add a new variable for the inverted output option.
- Add a new variable for the quad phase read index; initialize it in reset().

```
// Add your code here: ------------------------------------------------------

<SNIP SNIP SNIP>
// current read location
float m_fReadIndex;                    // NOTE its a FLOAT!
float m_fQuadPhaseReadIndex;           // NOTE its a FLOAT!
```

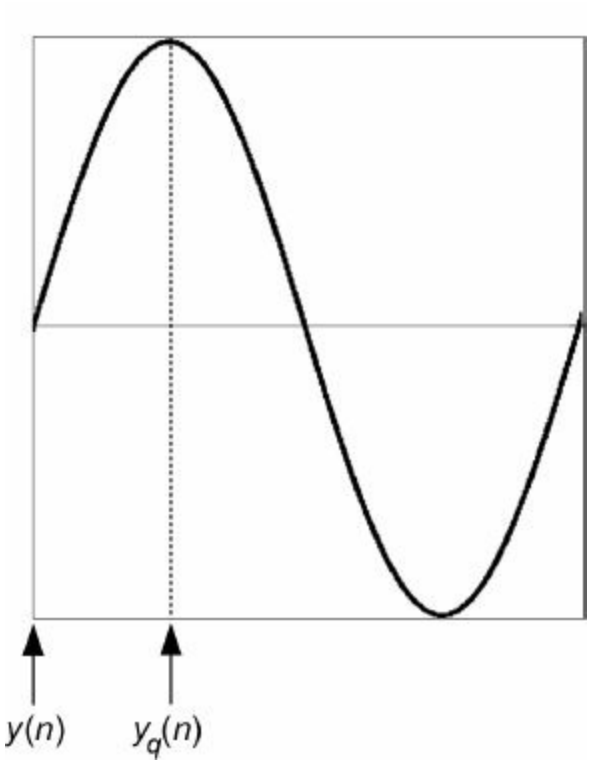


Figure 9.15: The concept of two read index values separated by one-quarter of the buffer; as long as they advance together by the same distance they will stay in quad phase. We currently have a bipolar oscillator which can easily be converted to a unipolar output by dividing by 2, then shifting the whole waveform up by 0.5.

```
// invert output
bool m_bInvert;

// reset the read index
void reset()
{
	m_fReadIndex = 0.0;				// top of buffer
	m_fQuadPhaseReadIndex = 256.0;		// 1/4 of our 1024 point buffer
}
```

9.8.2 WTOscillator.cpp File

Constructor

- Initialize the new index variable in the constructor; it will be reset to the proper value later.

```
CWTOscillator::CWTOscillator()
{
	<SNIP SNIP SNIP>
	// clear variables
	m_fReadIndex = 0.0;
	m_fQuadPhaseReadIndex = 0.0;
	m_f_inc = 0.0;

	// silent
	m_bNoteOn = false;

	// norm phase
	m_bInvert = false;
	etc…
}
```

9.8.3 WTOscillator.h File

Add the doOscillate() function. It needs to be called externally in future projects. It also needs to provide both quad phase outputs. It will be a mono function. It requires no input argument and we can test it from within the plug-in by sending the two phases out to the left and right channels, then view it with the analyzer. In the .h file, add the following:

```
// Add your code here: ----------------------------------------------------//
//
// funciton to do the Oscillator; may be called by an external parent Plug-In too
//  pYn is the normal output
//  pYqn is the quad phase output
void doOscillate(float* pYn, float* pYqn);

// Array for the Table

float m_SinArray[1024];				// 1024 Point Sinusoid
float m_SawtoothArray[1024];			// saw
float m_TriangleArray[1024];			// tri
float m_SquareArray[1024];			// sqr

Etc…
```

9.8.4 WTOscillator.cpp File

Add the new function doOscillate() to the .cpp file. Cut and paste the contents of the

processAudioFrame() for the left channel only. Just about everything is duplicated for the quad phase output. Of course, you can go in and streamline this code for more efficiency, but you can clearly see the operation of the two outputs here. Add the stuff in bold to the code you copied from processAudioFrame():

```
void CWTOscillator::doOscillate(float* pYn, float* pYqn)
{
	// IMPORTANT: for future modules, bail out if no note-on event!
	if(!m_bNoteOn)
	{
		*pYn = 0.0;
		*pYqn = 0.0;
		return;
	}
	// our output value for this cycle
	float fOutSample = 0;
	float fQuadPhaseOutSample = 0;

	// get INT part
	int nReadIndex = (int)m_fReadIndex;
	int nQuadPhaseReadIndex = (int)m_fQuadPhaseReadIndex;

	// get FRAC part - NOTE no Quad Phase frac is needed because it will be the same!
	float fFrac = m_fReadIndex - nReadIndex;


	// setup second index for interpolation; wrap the buffer if needed
	int nReadIndexNext = nReadIndex + 1 > 1023 ? 0 : nReadIndex + 1;
	int nQuadPhaseReadIndexNext = nQuadPhaseReadIndex + 1 > 1023 ? 0 :
	                              nQuadPhaseReadIndex + 1;

	// interpolate the output
	switch(m_uOscType)
	{
		case sine:
			fOutSample = dLinTerp(0, 1, m_SinArray[nReadIndex],
			                      m_SinArray[nReadIndexNext], fFrac);
			fQuadPhaseOutSample = dLinTerp(0, 1,
			                      m_SinArray[nQuadPhaseReadIndex],
			                      m_SinArray[nQuadPhaseReadIndexNext], fFrac);
			break;

		case saw:
			if(m_uTableMode == normal)     // normal
			{
				fOutSample = dLinTerp(0, 1, m_sawtoothArray[nReadIndex],
				        m_SawtoothArray[nReadIndexNext], fFrac);
				fQuadPhaseOutSample = dLinTerp(0, 1,
				        m_SawtoothArray[nQuadPhaseReadIndex],
				        m_SawtoothArray[nQuadPhaseReadIndexNext], fFrac);
			}
			else                                        // bandlimited
			{
				fOutSample = dLinTerp(0, 1,
				        m_SawtoothArray_BL5[nReadIndex],
				        m_SawtoothArray_BL5[nReadIndexNext], fFrac);
				fQuadPhaseOutSample = dLinTerp(0, 1,
				        m_SawtoothArray_BL5[nQuadPhaseReadIndex],
				        m_SawtoothArray_BL5[nQuadPhaseReadIndexNext],
				fFrac);
			}
			break;
		case tri:
		// ETC… FOLLOW THE SAME LOGIC AS THE OTHER TABLES //
```

```
        case SQUARE:
        // ETC… FOLLOW THE SAME LOGIC AS THE OTHER TABLES //
        // always need a default
        default:
                fOutSample = dLinTerp(0, 1, m_SinArray[nReadIndex],
                                m_SinArray [nReadIndexNext], fFrac);
                fQuadPhaseOutSample = dLinTerp(0, 1,
                                m_SinArray[nQuadPhaseReadIndex], m_SinArray
                                [nQuadPhaseReadIndexNext], fFrac);
                break;
    }

    // add the increment for next time
    m_fReadIndex += m_f_inc;
    m_fQuadPhaseReadIndex += m_f_inc;

    // check for wrap
    if(m_fReadIndex > 1024)
        m_fReadIndex = m_fReadIndex - 1024;

    if(m_fQuadPhaseReadIndex > 1024)
        m_fQuadPhaseReadIndex = m_fQuadPhaseReadIndex - 1024;

    // invert?
    if(m_bInvert)
    {
        fOutSample *= -1.0;
        fQuadPhaseOutSample *= -1.0;
    }

    // write out
    *pYn = fOutSample;
    *pYqn = fQuadPhaseOutSample;
}
```

process AudioFrame()

- Use the doOscillate() function.

```
bool __stdcall CWTOscillator::processAudioFrame(float* pInputBuffer, float*
                    pOutputBuffer, UINT uNumInputChannels, UINT uNumOutputChannels)
{
	// Do LEFT (MONO) Channel
	// if not running, write 0s and bail
	if(!m_bNoteOn)
	{
		pOutputBuffer[0] = 0.0;
		if(uNumOutputChannels == 2)
			pOutputBuffer[1] = 0.0;
		return true;
	}

	// some intermediate variables for return
	float fY = 0;
	float fYq = 0;

	// call the oscilator function, return values into fY and fYq
	doOscillate(&fY, &fYq);

	// write fY to the Left
	pOutputBuffer[0] = fY;

	// write fYq to the Right
	if(uNumOutputChannels == 2)
	{
		pOutputBuffer[1] = fYq;
	}
	return true;
}
```

Build and test the code. Figure 9.16 clearly shows the quad phase output: the sin() in the left channel, cos() is in the right channel.

9.9 Bipolar/Unipolar Functionality

9.9.1 WTOscillator GUI

Add the bipolar/unipolar switch to the UI using the next bank of radio buttons with the properties in Table 9.5. Edit it like you did with the previous buttons and give it the enumerations. I named my variable “m_uPolarity” and used “bipolar” and “unipolar” as my string/enums. The default will be bipolar.

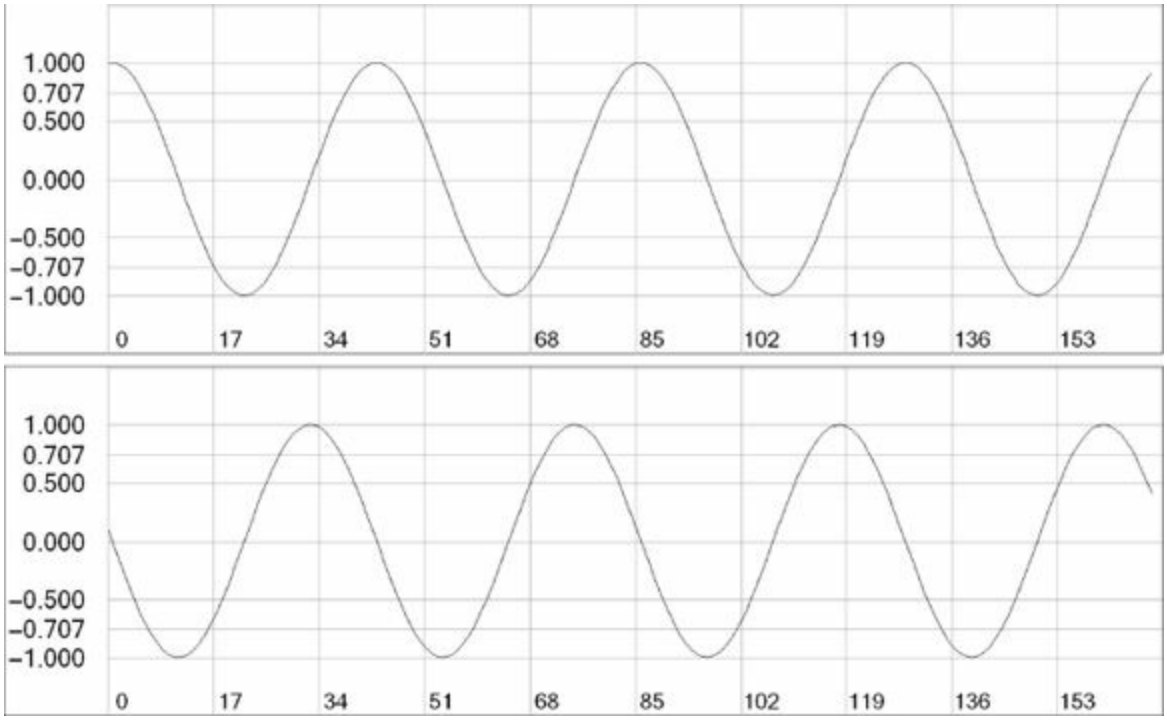


Figure 9.16: The quadrature-phase outputs of the left and right signals. Note: You must put the scope in free running mode to see the phase differences between channels.

Table 9.5: The button properties for the polarity control.

Slider Property	Value
Control Name	Osc Type
Units	
Variable Type	enum
Variable Name	m_uPolarity
enum string	bipolar, unipolar

9.9.2 WTOscillator.cpp File

Add the bipolar/unipolar functionality to the very last part of doOscillate() to divide by 2 then shift by 0.5 as follows:

```
void CWTOscillator::doOscillate(float* pYn, float* pYqn)
{
	<SNIP SNIP SNIP>
	// write out
	*pYn = fOutSample;
	*pYqn = fQuadPhaseOutSample;

	// create unipolar; div 2 then shift up 0.5
	if(m_uPolarity == unipolar)

	{
		*pYn /= 2.0;
		*pYn += 0.5;

		*pYqn /= 2.0;
		*pYqn += 0.5;
	}
}
```

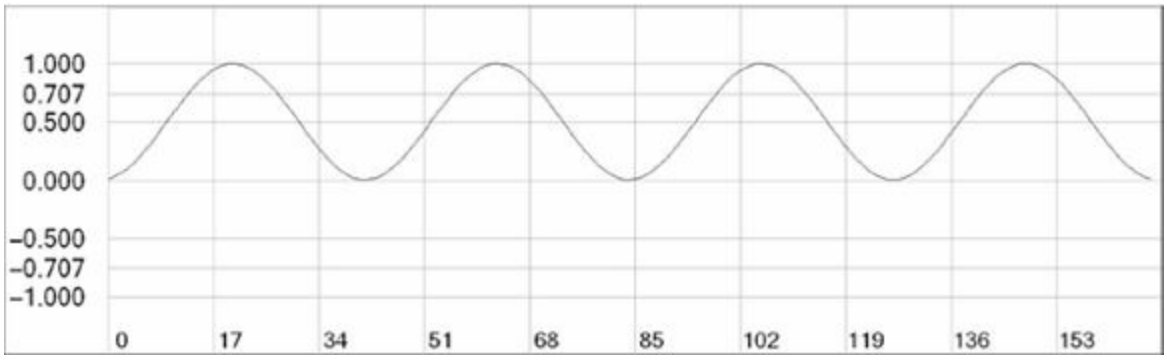


Figure 9.17: Unipolar operation, sinusoid.

Build and test the code; Figure 9.17 shows the output for a unipolar sinusoid.

Here are some projects to try:

- Higher-resolution tables; each octave of the piano fingerboard gets its own table with as many harmonics as the upper note of the octave permits.
- A two-oscillator synth module with a detuning control for one of the oscillators.
- An oscillator that only plays musical pitches.

In the next chapter, we'll use our new wave table oscillator to make modulated delay effects.

Bibliography

Dattorro, J. 2003. Effect design, Part 3, oscillators: Sinusoidal and pseudonoise. *Journal of the Audio Engineering Society*, 50(3): 114–146.

Gordon, J. W. and Smith, J. O. 1985. A sine generation algorithm for VLSI applications. *Proceedings from the International Computer Music Conference*, pp. 165–168.

Moore, R. 1990. *Elements of Computer Music*, Chapter 3. Englewood Cliffs, NJ: Prentice-Hall.

Roads, C. 1996. *The Computer Music Tutorial*, Chapter 2. Cambridge, MA: The MIT Press.

Modulated Delay Effects

Perhaps the most interesting kinds of audio effects, from both listening and engineering standpoints, are the modulated delay effects. They include the chorus and flanger/vibrato; additionally, modulated delays are also found in some reverb algorithms. These time-varying filters are actually quite sophisticated and fun to implement, and we have gone to great lengths to create useful sub-modules. These include the digital delay effect or DDL (digital delay line) module and the wave table oscillator. If you can design, build, and implement modulated delay effects, then you are well on your way to proficiency in audio effects coding. These effects have parameters and coefficients which change on every sample period.

Figure 10.1a shows the standard delay line or DDL. The output is fed back via the *fb* coefficient. The delay is constant until the user changes it to another fixed value. In the modulated delay line (MDL) in Figure 10.1b the amount of delay is constantly changing over time. The system only uses a portion of the total available delay amount.

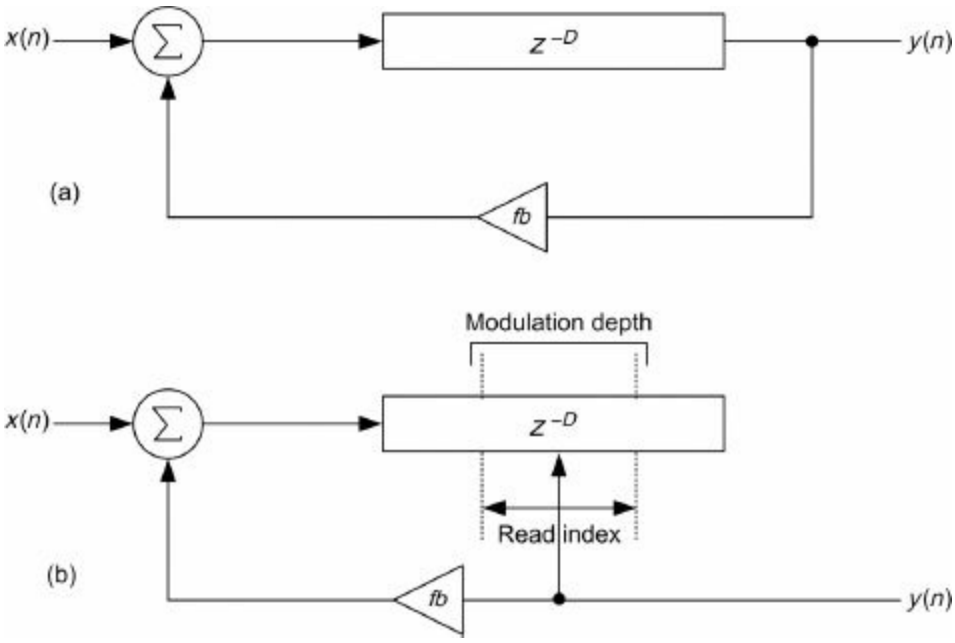


Figure 10.1: (a) A static delay and (b) modulated delay effect.

The *modulation depth* relates to the size of the portion of the delay line that is being read. The *modulation rate* is the speed at which the read index moves back and forth over the modulated delay section. A low-frequency oscillator (LFO) is used to modulate the delay amount and the LFO can be just about any kind of waveform, even noise. The most common waveforms are triangle and sinusoid. In order to make a modulated delay effect, you first need to make a delay line, and then modify it to constantly change and update the read location on each sample interval.

10.1 The Flanger/Vibrato Effect

The flanger effect gets its name from the original analog method of running two tape machines slightly out of sync with each other by placing a thumb on the flange (the ring that holds the tape) of one machine's reel and applying pressure to slow down the tape ever so slightly. Other variations include rubbing the flange circularly. The effect creates a time-varying delay that falls out of sync, then back in sync with the first tape machine. The effect has been compared to the whoosh of a jet engine, or the sound of a phaser effect. No matter how you define it, the flanger has a unique sound. The effect is heard on numerous recordings.

In order to simulate the flanger effect, you need a delay line whose read index moves away from, then back to the top or starting point of the delay. When the read pointer is back to the top, there is no delay and as it moves away from the write index, the pitch is modulated down. When the index turns around and moves back, the pitch is modulated up—this is the Doppler effect in practice. When the output of the modulated delay is sent out 100% wet, with no dry signal, the effect is that of vibrato—the pitch shifts up and down as the pointer moves back and forth. When the dry signal is mixed with the wet signal, the pitch shifting changes into the flanging effect. The amount of delay time varies between 0 and about 7–10 mSec; at a 44,100 Hz sample rate, this corresponds to around 308 to 410 samples. Feedback is usually added to increase the effect by creating resonances or anti-resonances, depending on whether the feedback is normal or inverted.

Figure 10.2 shows the two states of the modulator with increasing and decreasing pitch shifting due to the Doppler effect. Figure 10.3 shows alternate ways of diagramming the same modulator. The vibrato and flanger are the same effect underneath the hood—it's only the mix ratio that determines which effect you have. Because the delay is constantly moving around we are once again faced with options on how to handle the fractional delay that is required; after you calculate the number of samples of delay (which will most likely have a fractional component) you can

- Truncate the value and just use the integer part.
- Round the value up or down.

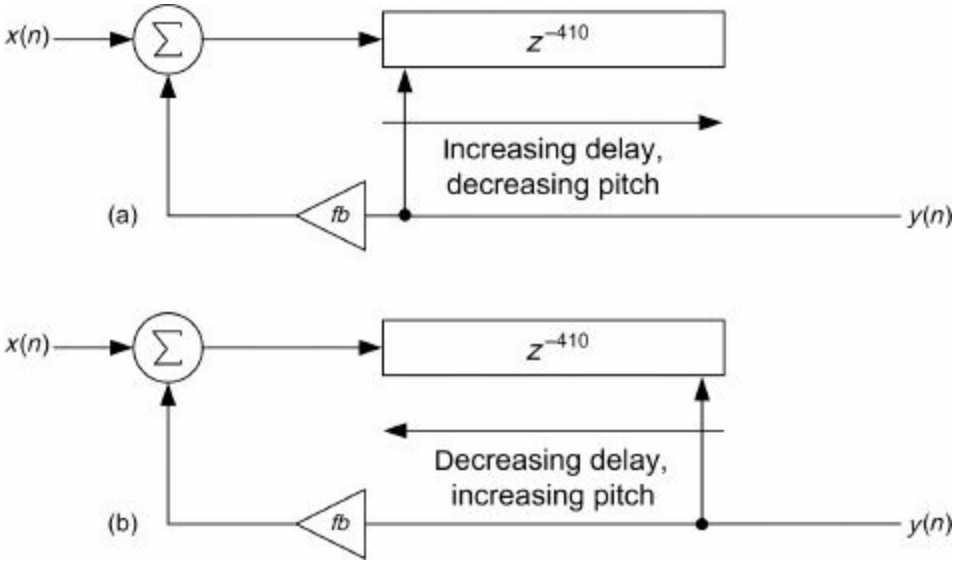


Figure 10.2: (a) As the delay increases, the pitch drops. At the end of the delay, the index turns around and begins heading back toward the write index (b), increasing the pitch; the maximum delay here is about 10 mSec at 44,100 Hz.

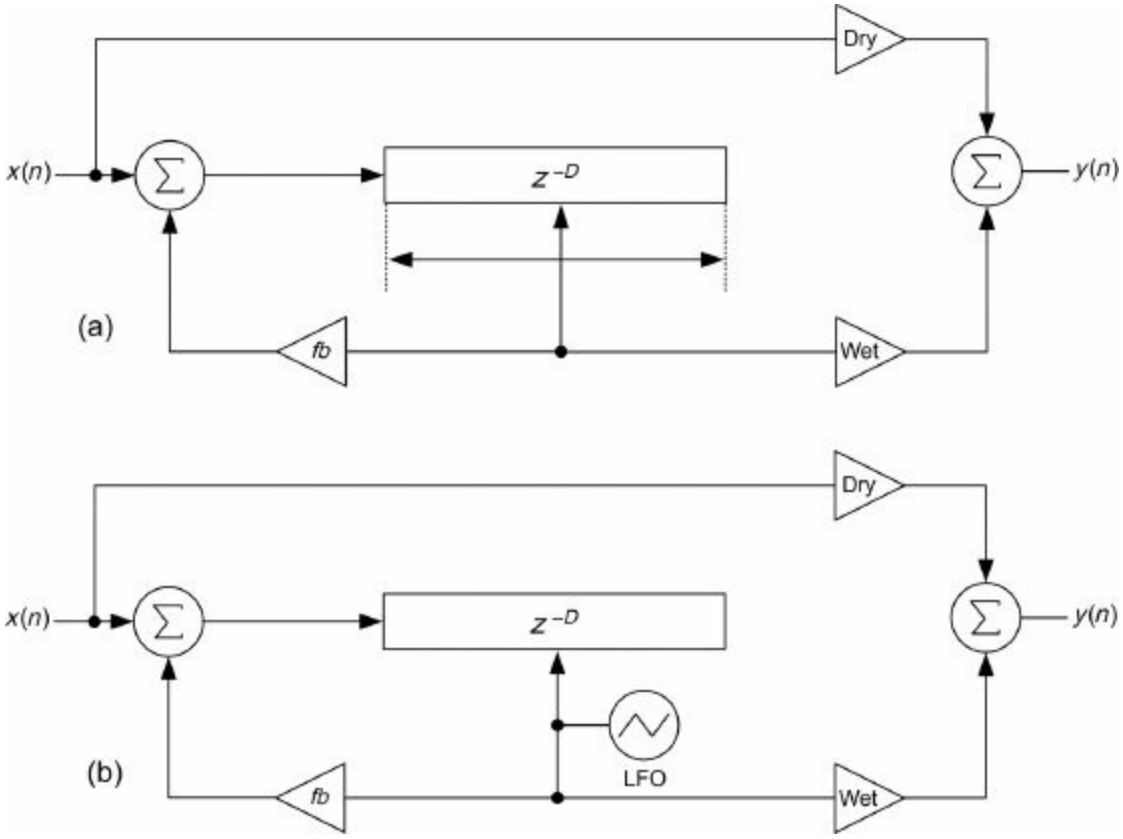


Figure 10.3: (a) The simplest form of the flanger/vibrato effect. The dotted lines show the limits of the delay modulation, from 0.0 (no delay) to the full range, z^{-D} samples. (b) An alternate version shows the LFO connection to the modulation index to the delay line.

- Interpolate the delayed value using linear or polynomial interpolation.
- Use an all-pass filter to make the fractional delay (see the Bibliography).

The flanger/vibrato controls consist of:

- Modulation depth: how much of the available range is used.
- Modulation rate: the rate of the LFO driving the modulation.
- LFO type: usually sinusoidal or triangular, but may be anything.
- Feedback: as with the normal delay, the feedback has a big effect on the final product.
- Wet/dry control: 100% wet = vibrato; 50/50 = flanger.

As shown in Figure 10.4, the flanger/vibrato technically always starts with 0.0 samples of delay. Because of this, we'd like to have a LFO that can generate a unipolar output from 0.0 to +1.0, which we can then map to a range of delays, from 0 to about 410 samples or so. Figure 10.5 shows a stereo version of the flanger. A common LFO modulates both delay lines, but the phases are off by 90 degrees, putting them in quadrature or quad-phase. The effect is interesting as the two LFOs seem to chase each other, one on the left and the other on the right.

The flanger technically modulates the delay right down to 0 samples. For analog tape flanging, this occurs when the two tape machines come back into perfect synchronization. This is called *through-zero flanging* or TZF. This means that for an instant, the output is double the input value. For low frequencies and/or high feedback values, this can cause a problem when the flanger delay time is between 0 and about 0.1 mSec and the waveform is at a peak location. In this case, large valued

samples near each other on the waveform sum together, producing an increased and sometimes distorted bass response.

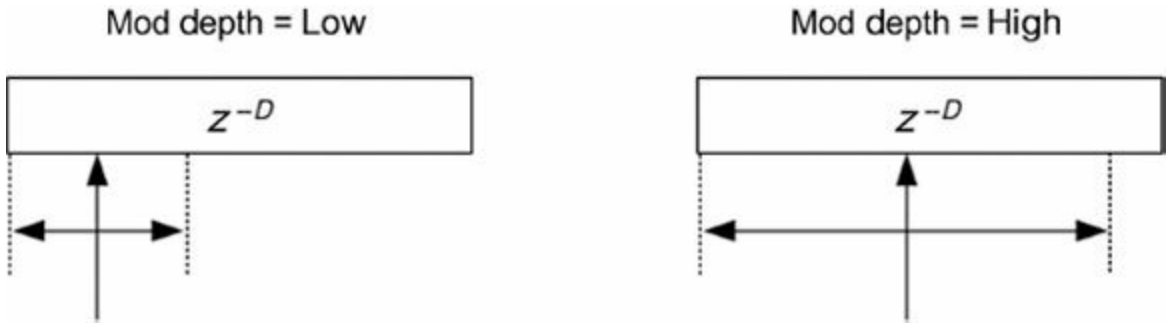


Figure 10.4: Two modulation depth settings; zero is always the minimum delay.

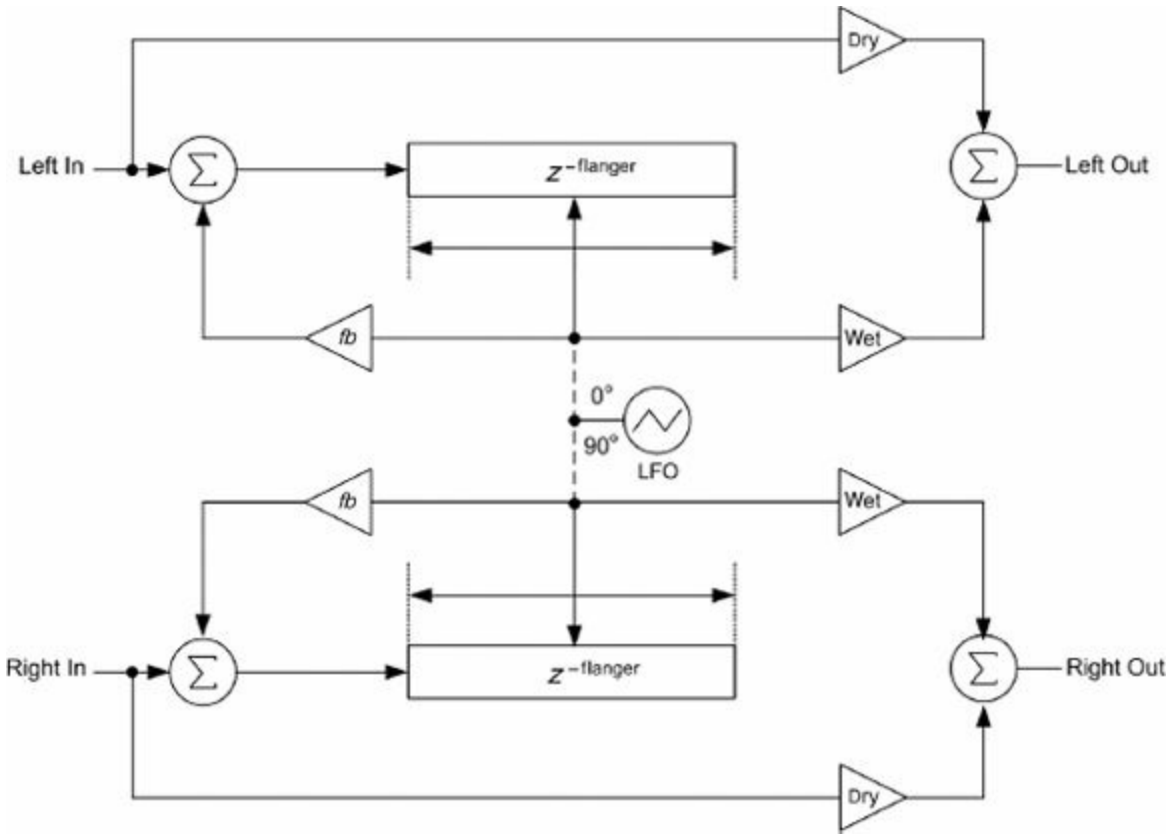


Figure 10.5: A stereo quadrature flanger; the dotted lines at the LFO show its 90-degree offset modulation. The max depth spans the entire delay line.

Table 10.1: Delay times for the flanger and chorus effects.

	Min Delay(mSec)	Max Delay(mSec)	Feedback(%)
Flanger	0	7–10	−90 to +90
Chorus	7	20–40	Typically 0

10.2 The Chorus Effect

The chorus effect appeared in the late 1970s and was used on countless recordings during the 1980s,

becoming a signature sound effect of that decade. The chorusing effect can produce a range of effects from a subtle moving sheen to a swirling sea-sick de-tuned madness. Like the flanger, it is also based on a modulated delay line, driven by an LFO. Although different manufacturers adopted different designs, the basic algorithm for a single chorusing unit is the same. The differences between a flanger and chorus are shown in Table 10.1.

- Analog chorus units typically do not use any feedback (because our DDL module already has feedback, we can keep it for experimentation purposes).
- The range of delay times over which the device operates is the most significant difference.

The read index is modulated about the center of the pointer but the center is set to avoid the keep-out zone (Figures 10.6 and 10.7). Also, the modulation depth and the location of the center of the delay area vary across manufacturers (Figure 10.8).

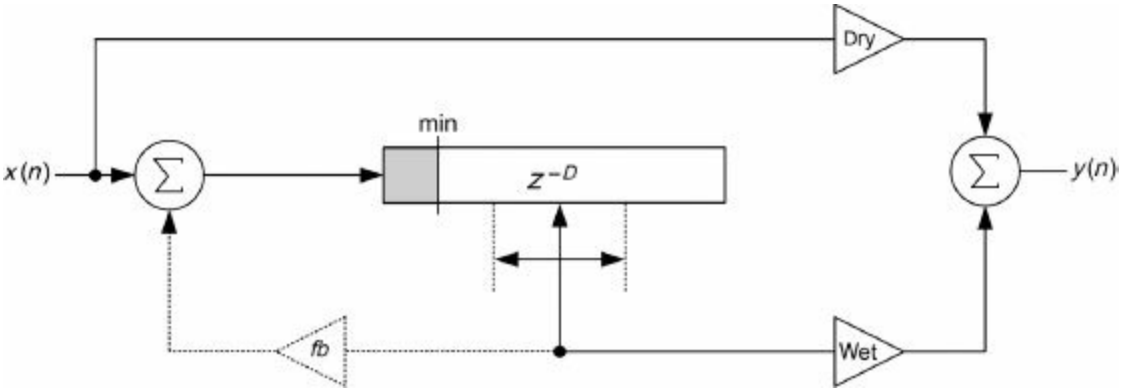


Figure 10.6: The basic chorus module. Note the feedback path is optional and not found in typical analog chorus units. The gray area is the flanger keep-out zone.

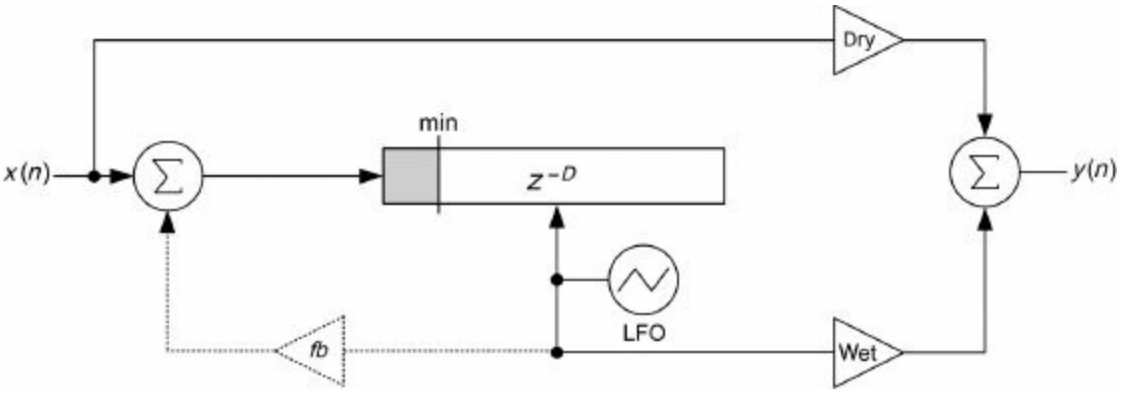


Figure 10.7: An alternative block diagram that describes the same chorus module.



Figure 10.8: (a) and (b) are two different chorus modules with the same maximum depth (dotted lines) but a different center of operation.

Many variations on the basic chorus module exist, including:

- Multiple chorus modules with different centers of operation
- Different LFO frequency or LFO phases applied to different modules
- Series modules
- Parallel modules
- Bass modules with a low-pass filter/high-pass filter on the front end

Figure 10.9 shows the stereo quad chorus with the left and right LFOs out of sync by 90 degrees. In this permutation, the maximum depths of the left and right channels are independent, as well as the ability to adjust them. The flowchart for the basic modulated delay effect for each processAudioFrame() is shown in Figure 10.10.

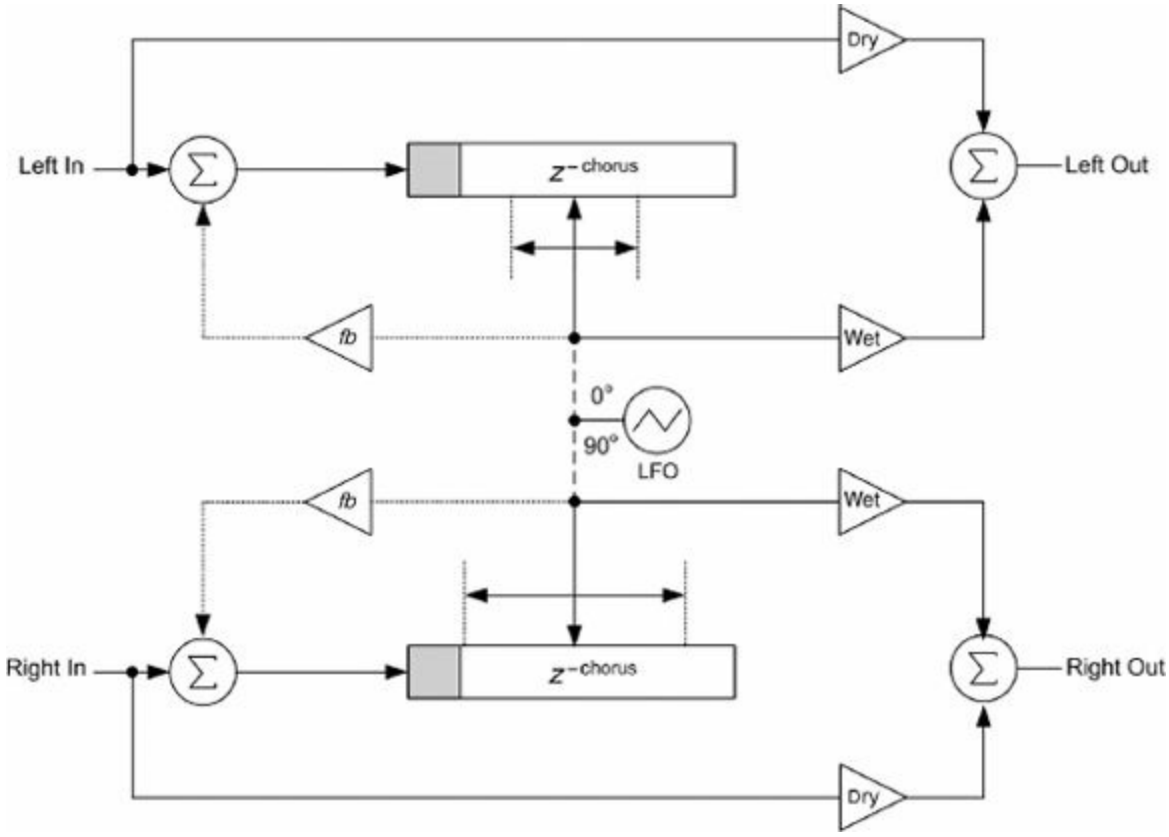


Figure 10.9: A stereo quadrature chorus; feedback paths are optional.

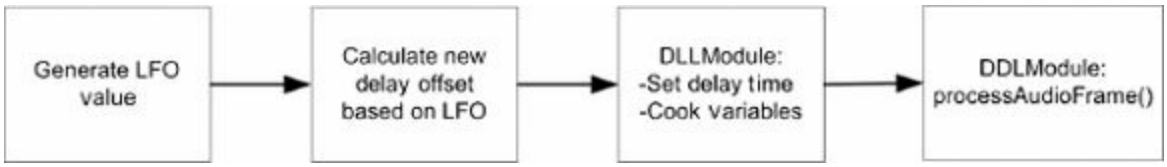


Figure 10.10: The flowchart for modulated delay effects.

You will use the DDL module to do most of the low-level work and generate the delayed values. The LFO will generate values that will be used to calculate the current delay offset. Since the LFO is varying, the delay offset will need to be recalculated on each sample interval.

10.3 Design a Flanger/Vibrato/Chorus Plug-In

Now we get down to the business of coding these effects. Because of the large number of variations,

we should start by coding a module that can be used in larger parent plug-in objects. Figure 10.11 shows the block diagram.

We will provide several user interface (UI) components to control the plug-in and their variables are available for a parent plug-in to use as well. We will use the existing DDL module and wave table oscillator plug-ins. Since the DDL module has feedback path control, there will be many ways to combine these modules into sophisticated units. The controls we will provide for the user are discussed next. The plan of operation is as follows:

- LFO: needs to be set up to run unipolar and with the non-band-limited tables.
- The chorus delay offset control will effectively move the center of operation for the modulated delay.
- The vibrato mode will force the feedback value to 0.0% since it does not sound good with feedback.
- prepareForPlay()
  - Reset the LFO and start it by setting the m_bNoteOn flag; also set up the LFO and DDL modules.
  - Set the DDL m_nSampleRate variable; it needs this value to initialize its buffer properly.
  - Forward the prepareForPlay() call to the child objects, LFO and DDL.

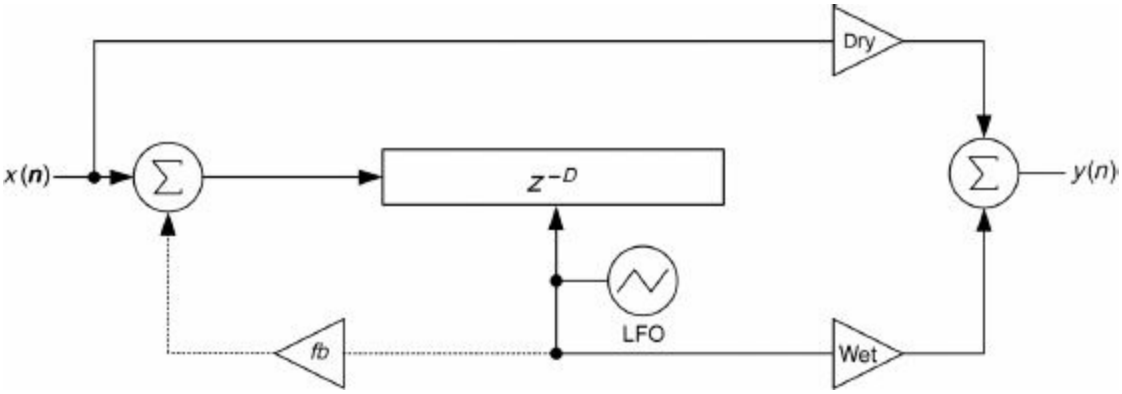


Figure 10.11: Generic modulated delay block diagram; it can be any type of flanger/vibrato/chorus.

- processAudioFrame()
  - Call the doOscillate() function on the LFO.
  - Use the LFO value to calculate a delay offset in milliseconds for the DDL module.
  - Write the new delay time to the DDL module and call its cooking function.
  - Call processAudioFrame() on the DDL module to get the current output sample.
- userInterfaceChange()
  - Call the update functions for the LFO and DDL.
  - Call the type switching function to change from flanger to vibrato and chorus.

We'll go ahead and plan on some helper functions to make this process easier. This is not going to be the most streamlined or efficient code, but once you have it mastered you can always go back and improve it. Although this is a difficult project, you only need two member variables and four functions to accomplish the design.

The member objects are as follows:

```
CWTOscillator m_LFO;    // our LFO
CDDLModule m_DDL;       // our delay line module
```

The member variables are as follows:

```
float m_fMinDelay_mSec;   // min delay time
float m_fMaxDelay_mSec;   // max delay time
```

The member functions are as follows:

```
float updateLFO();   // change/set the type and rate
float updateDDL();   // change/set the wet/dry and feedback
float cookModType();   // set up the min/max delay times
float calculateDelayOffset(float fLFOSample);   // convert the LFO to a delay value
```

10.3.1 Project: ModDelayModule

Start a new project and make sure to include your WTOscillator and DDLModule objects (Figure 10.12).



Figure 10.12: Project settings for the ModDelayModule.

Table 10.2: The GUI components, variable names, and values.

Slider Property	Value
Control Name	Depth
Units	%
Variable Type	float
Variable Name	m-fModDepth-pct
Low Limit	0
High Limit	100
Initial Value	50

Slider Property	Value
Control Name	Rate
Units	Hz
Variable Type	float

Variable Name	m-fModFrequency-Hz
Low Limit	0.02
High Limit	5
Initial Value	0.18

Slider Property	Value
Control Name	Feedback
Units	%
Variable Type	float
Variable Name	m-fFeedback-pct
Low Limit	-100
High Limit	100
Initial Value	0

Slider Property	Value
Control Name	Chorous Offset
Units	mSec
Variable Type	float
Variable Name	m-fChorousOffset
Low Limit	0
High Limit	30
Initial Value	0

Button Property	Value
Control Name	Mod Type
Units	
Variable Type	enum
Variable Name	m_uModType
Enum String	Flanger, Vibrato, Chorus

Button Property	Value
Control Name	LFO
Units	
Variable Type	enum
Variable Name	m-uLFOType
Enum String	tri, sin

10.3.2 ModDelayModule GUI

Set up the graphical user interface (GUI) in RackAFX with Table 10.2.

10.3.3 ModDelayModule.h File

Add the member objects, variables, and function declarations to the .h file:

```
// Add your code here: ----------------------------------------------------- //
CWTOscillator m_LFO;   // our LFO
CDDLModule m_DDL;      // our delay line module

// these will depend on the type of mod
float m_fMinDelay_mSec;
float m_fMaxDelay_mSec;
// functions to update the member objects
void updateLFO();
void updateDDL();

// cooking function for mod type
void cookModType();

// convert a LFO value to a delay offset value
float calculateDelayOffset(float fLFOSample);

// END OF USER CODE ------------------------------------------------------- //
```

10.3.4 ModDelayModule.cpp File

Constructor

- Initialize all variables.

```
CModDelayModule::CModDelayModule()
{
      <SNIP SNIP SNIP>

      // Finish initializations here
      m_fMinDelay_mSec = 0.0;
      m_fMaxDelay_mSec = 0.0;
      m_fChorusOffset = 0.0;

      m_LFO.m_fFrequency_Hz = 0;
      m_LFO.m_uOscType = m_uLFOType; // triangle  enum{tri,sin};

      m_DDL.m_bUseExternalFeedback = false;
      m_DDL.m_fDelay_ms = 0;

}
```

Write the update() functions:

```
// LFO function to set:
//                    - the LFO Frequency
//                    - the oscillator type
void CModDelayModule::updateLFO()
{
        // set raw data
        m_LFO.m_fFrequency_Hz = m_fModFrequency_Hz;
        m_LFO.m_uOscType = m_uLFOType;

        // cook it
        m_LFO.cookFrequency();
}
// DDL function to set:
//                    - the DDL Feedback amount (disabled for Vibrato)
void CModDelayModule::updateDDL()
{
        // test and set if needed
        if(m_uModType != Vibrato)
                m_DDL.m_fFeedback_pct = m_fFeedback_pct;

        // cook it
        m_DDL.cookVariables();
}
```

Write the cookModType() function using the information in Table 10.3—this method is called when the user changes the effect type.

```
// cookMod() function:
/*
            Min Delay (mSec)     Max delay (mSec)     Wet/Dry(%)     Feedback(%)
Flanger     0                    7-10                 50/50          -100 to +100
Vibrato     0                    7-10                 100/0          0
Chorus      5                    20-40                50/50          -100 to +100

*/
void CModDelayModule::cookModType()
{
       switch(m_uModType)
       {
              case Flanger:
              {
                     m_fMinDelay_mSec = 0;
                     m_fMaxDelay_mSec = 7;
                     m_DDL.m_fWetLevel_pct = 50.0;
                     m_DDL.m_fFeedback_pct = m_fFeedback_pct;
                     break;
              }
              case Vibrato:
              {
                     m_fMinDelay_mSec = 0;
                     m_fMaxDelay_mSec = 7;
                     m_DDL.m_fWetLevel_pct = 100.0;
                     m_DDL.m_fFeedback_pct = 0.0; // NOTE! no FB for vibrato
                     break;
              }
```

Table 10.3: The default delay line settings for the various effects.

	Min Delay(mSec)	Max Delay(mSec)	Wet/Dry(%)	Feedback(%)
Flanger	0	7-10	50/50	-100 to +100
Vibrato	0	7-10	100/0	0

Chorus	7	20-40	50/50	Typically 0

```
            case Chorus:
            {
                m_fMinDelay_mSec = 5;
                m_fMaxDelay_mSec = 30;
                m_DDL.m_fWetLevel_pct = 50.0;
                m_DDL.m_fInputAttenuation = 1.0;
                break;
            }

            default: // is Flanger
            {
                m_fMinDelay_mSec = 0;
                m_fMaxDelay_mSec = 7;
                m_DDL.m_fWetLevel_pct = 50.0;
                m_DDL.m_fFeedback_pct = m_fFeedback_pct;
                break;
            }
        }
}
```

Write the calculateDelayOffset() function. It works by taking the LFO value from 0 to 1.0 and mapping it on to the range from min to max delay. Then, this value is scaled by the depth percent value so that at 100% the maximum range is used. For the chorus, we effectively move the operating range and chorus center by adding a delay offset to the minimum value, then using that as the starting (reference) point in the calculation.

```
// calculateDelayOffset():
/*
        fLFOSample: a value from 0.0 to 1.0 from the LFO object

        returns: the calculated delay time in mSec for each effect

        NOTES: - the range for the flanger/vibrato is simply mapped from min to max
                 starting at min: fLFOSample*(m_fMaxDelay_mSec - m_fMinDelay_mSec)) +
                 m_fMinDelay_mSec

               - the range for the Chorus includes the starting offset
                 fStart = m_fMinDelay_mSec + m_fChorusOffset;
*/
if(m_uModType == Flanger || m_uModType == Vibrato)
{
        // flanger 0->1 gets mapped to 0->maxdelay
        return (m_fModDepth_pct/100.0)*(fLFOSample*(m_fMaxDelay_mSec -
                m_fMinDelay_mSec)) + m_fMinDelay_mSec;
        }
        else if(m_uModType == Chorus)
        {
                // chorus adds starting offset to move delay range
                float fStart = m_fMinDelay_mSec + m_fChorusOffset;
                return (m_fModDepth_pct/100.0)*(fLFOSample*(m_fMaxDelay_mSec -
                        m_fMinDelay_mSec)) + fStart;
        }
}
```

10.3.5 PrepareForPlay()

- Forward the call to prepareForPlay() on the member objects.
- Reset and start up the LFO in unipolar mode.

```
bool __stdcall CModDelayModule::prepareForPlay()
{
	// Add your code here:
	m_LFO.prepareForPlay();

	// DDL needs to know sample rate to initialize its buffer
	m_DDL.m_nSampleRate = m_nSampleRate;
	m_DDL.prepareForPlay();

	m_LFO.m_uPolarity = 1;   // 0 = bipolar, 1 = unipolar
	m_LFO.m_uTableMode = 0;  // normal, no band limiting
	m_LFO.reset();           // reset it

	// initialize
	cookModType();
	updateLFO();
	updateDDL();

	// start the LFO!
	m_LFO.m_bNoteOn = true;

	return true;
}
```

userInterfaceChange()

- Call the update functions and cook the mod type. We won't bother with the switch/case statement since changes to any slider will require updates of the LFO and DDL.
- Make sure your nControlIndex values match your GUI components.

```
bool __stdcall CModDelayModule::userInterfaceChange(int nControlIndex)
{
	// change the min/max limits; set wet/dry and Feedback
	if(nControlIndex == 41) // 41 is mod type switch
		cookModType();

	// else just call the other updates which handle all the rest
	//
	// frequency and LFO type
	updateLFO();

	// Wet/Dry and Feedback
	updateDDL();

	return true;
}
```

processAudioFrame()

- Call the doOscillate() function on the LFO.
- Use the LFO value to calculate a delay offset in milliseconds for the DDL module.
- Write the new delay time to the DDL module and call its cooking function.
- Call processAudioFrame() on the DDL module to get the current output sample.

```
bool __stdcall CModDelayModule::processAudioFrame(float* pInputBuffer, float*
                                        pOutputBuffer, UINT uNumInputChannels,
                                        UINT uNumOutputChannels)
{
        // Do LEFT (MONO) Channel
        //
        // 1. Get LFO Values, normal and quad phase
        float fYn = 0;
        float fYqn = 0;
        m_LFO.doOscillate(&fYn, &fYqn);

        // 2. calculate delay offset
        float fDelay = 0.0;
        if(m_uLFOPhase == quad)
                fDelay = calculateDelayOffset(fYqn); // quadrature LFO
        else
                fDelay = calculateDelayOffset(fYn); // normal LFO

        // 3. set the delay & cook
        m_DDL.m_fDelay_ms = fDelay;
        m_DDL.cookVariables();

        // 4. get the delay output one channel in/one channel out
        m_DDL.processAudioFrame(&pInputBuffer[0], &pOutputBuffer[0], 1, 1);

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0];

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0];

        return true;
}
```

Build and test the module. Be sure to listen to the flanger as the delay hits the zero point.

10.3.6 Challenge

Add another radio button switch to turn on or off TZF that we employ. To turn off TZF, set the minimum delay time for flanging to a nonzero value (try 0.1 mSec to begin with) in the cookModType() method. You now have a working modulated delay module that can be combined in many ways to create a wide variety of modulated delay effects. Let's try a couple of them now.

10.4 Design a Stereo Quadrature Flanger Plug-In

The block diagram of the stereo quad flanger is shown in Figure 10.5 and it consists of two identical flanger delay lines running off of LFO values that are in quadrature phase. Since our mod delay module has a built-in LFO and delay, we can assemble this plug-in quickly.

10.4.1 Project: StereoQuadFlanger

Make sure to add all the existing modules when you create the new StereoQuadFlanger project:

- ModDelayModule.h.

- DDLModule.h (because ModDelayModule # includes it).
- WTOscillator.h (because DDLModule #includes it).

10.4.2 StereoQuadFlanger GUI

We will use a simpler UI consisting of the controls shown in Table 10.4.

10.4.3 StereoQuadFlanger.h File

In the .h file, declare two instances of the ModDelayModule, one for the left and one for the right. Also add one helper function to initialize and update the modules. These two member variables and the one function are all that you need.

```
// Add your code here: ------------------------------------------------------ //
CModDelayModule m_ModDelayLeft;
CModDelayModule m_ModDelayRight;

void updateModDelays();
// END OF USER CODE --------------------------------------------------------- //
```

Table 10.4: The GUI elements for the StereoQuadFlanger.

Slider Property	Value
Control Name	Depth
Units	%
Variable Type	float
Variable Name	m-fModDepth-pct
Low Limit	0
High Limit	100
Initial Value	50

Slider Property	Value
Control Name	Feedback
Units	%
Variable Type	float
Variable Name	m-fFeedback-pct
Low Limit	−100
High Limit	100
Initial Value	50

Slider Property	Value
Control Name	Rate
Units	Hz

Variable Type	float
Variable Name	m-fModFrequency-Hz
Low Limit	0.02
High Limit	5
Initial Value	0.18

Button Property	Value
Control Name	LFO
Units	
Variable Type	enum
Variable Name	m-uLFOType
Enum String	tri, sin

10.4.4 StereoQuadFlanger.cpp File

Add the one helper function updateModDelays(); this is also the function that forces the two mod lines into quad phase:

```
// update all parameters of the ModDelayModules:
/*
        - setup for stereo quad phase
        - mod depth
        - mod rate
        - feedback
        - mod type = 0 = flanger
        - LFO type
        - call the update() functions to cook everything
*/
void CStereoQuadFlanger::updateModDelays()
{
        // setup quad phase
        m_ModDelayLeft.m_uLFOPhase = 0; // 0: normal
        m_ModDelayRight.m_uLFOPhase = 1; // 1: quad phase

        m_ModDelayLeft.m_fModDepth_pct = m_fModDepth_pct;
        m_ModDelayRight.m_fModDepth_pct = m_fModDepth_pct;
```

```
        m_ModDelayLeft.m_fModFrequency_Hz = m_fModFrequency_Hz;
        m_ModDelayRight.m_fModFrequency_Hz = m_fModFrequency_Hz;

        m_ModDelayLeft.m_fFeedback_pct = m_fFeedback_pct;
        m_ModDelayRight.m_fFeedback_pct = m_fFeedback_pct;

        // FLANGER!
        m_ModDelayLeft.m_uModType = 0;
        m_ModDelayRight.m_uModType = 0;

        m_ModDelayLeft.m_uLFOType = m_uLFOType;
        m_ModDelayRight.m_uLFOType = m_uLFOType;

        // cook them
        m_ModDelayLeft.updateLFO();
        m_ModDelayLeft.updateDDL();

        m_ModDelayRight.updateLFO();
        m_ModDelayRight.updateDDL();
}
```

prepareForPlay()

- Forward the call to the member objects.
- updateModDelays().

```
bool __stdcall CStereoQuadFlanger::prepareForPlay()
{
        // Add your code here:
        // call forwarding!
        m_ModDelayLeft.prepareForPlay();
        m_ModDelayRight.prepareForPlay();

        // don't leave this out - it inits and cooks
        updateModDelays();

        return true;
}
```

userInterfaceChange()

- updateModDelays().

```
bool __stdcall CStereoQuadFlanger::userInterfaceChange(int nControlIndex)
{
       // just do a brute force update of all
       updateModDelays();
}
```

processAudioFrame()

- Forward the processAudioFrame() function to the member objects to do the processing.

```
bool __stdcall CStereoQuadFlanger:: processAudioFrame(float* pInputBuffer, float*
                                        pOutputBuffer, UINT uNumInputChannels,
                                        UINT uNumOutputChannels)
{
      // Do LEFT (MONO) Channel
      m_ModDelayLeft.processAudioFrame(&pInputBuffer[0], &pOutputBuffer[0], 1, 1);

      // Mono-In, Stereo-Out (AUX Effect)
      if(uNumInputChannels == 1 && uNumOutputChannels == 2)
           pOutputBuffer[1] = pOutputBuffer[0];

      // Stereo-In, Stereo-Out (INSERT Effect)
      if(uNumInputChannels == 2 && uNumOutputChannels == 2)
           m_ModDelayRight.processAudioFrame(&pInputBuffer[1], &pOutputBuffer[1],
           1, 1);

      return true;
}
```

Build and test the project. The code is so sparse because we already did most of the work. The only tricky part about this is making sure you have the objects properly initialized and cooked and that you forward the calls correctly. If you have errors or crashes, it's probably because your objects are not properly initialized.

10.4.5 Challenges

- Add a control to toggle TZF mode.
- Add a control to let you turn on and off quad phase (it's currently hard-coded “on”).
- Add more LFO shapes.

10.5 Design a Multi-Unit LCR Chorus Plug-In

This LCR or left-center-right chorus (based on the Korg Triton®) uses three chorusing modules run by three independent LFOs, each with its own depth and rate controls (Figure 10.13). If we play it right, we can code this with a minimum of effort, but we have to be very careful about book-keeping since we have many variables here. The UI will also be more complicated, with three sliders per chorus module: depth, rate, and feedback. The LFO type is fixed as a triangle for all units. L is in quad phase, R is in inverse-quad phase, and C is normal.

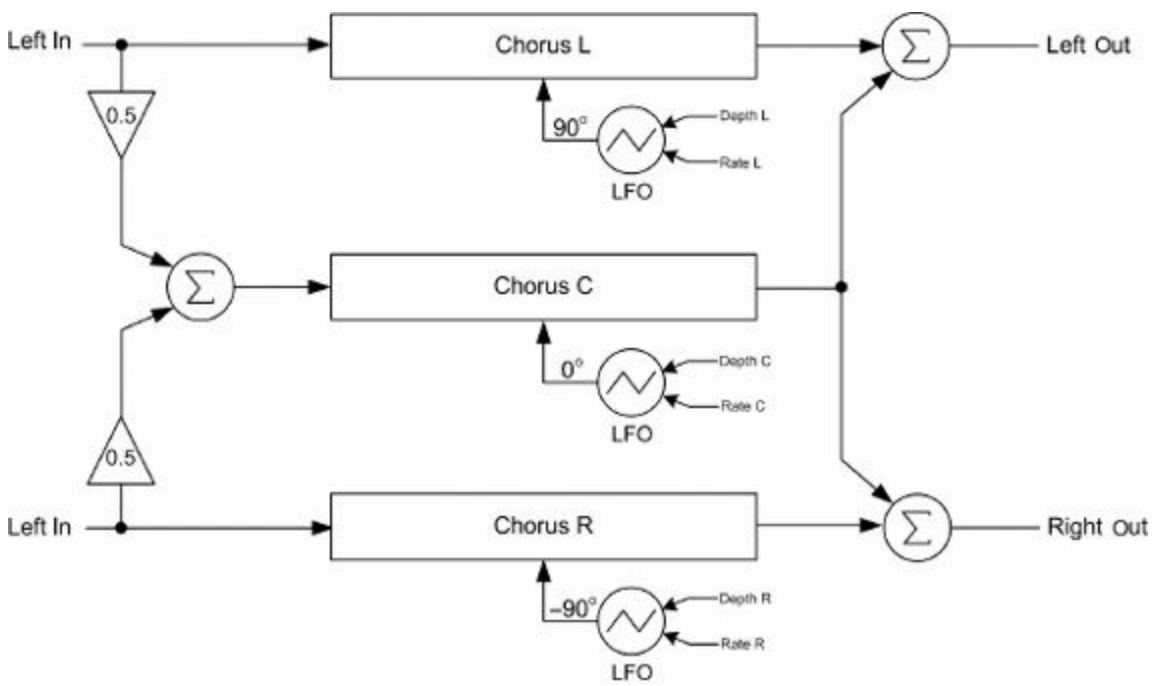


Figure 10.13: The LCR chorus.

10.5.1 Project: StereoLCRChorus

Make sure to add all the existing modules when you create the new project:

- ModDelayModule.h.
- DDLModule.h (because ModDelayModule #includes it).
- WTOscillator.h (because DDLModule #includes it).

10.5.2 StereoLCRChorus GUI

This project contains repeated sets of controls. There are depth, rate, and feedback values for each of the three modules. The sets of controls share the same min and max values. You will have nine slider controls in all. Use Table 10.5 to set up the controls.

10.5.3 StereoLCRChorus.h File

Declare your member objects in the .h file. Also, declare an update() function to modify the parameters all at once.

```
// Add your code here: ---------------------------------------------------- //
//
// One module for each LCR
```

Table 10.5: GUI elements for the LCR chorus.

Depth (%)	Rate (Hz)	Feedback (%)
Min = 0	Min = 0.02	Min = −100

Max = 100	Max = 5	Max = +100
Initial L = 50	Initial L = 0.18	Initial L = 0
Initial C = 50	Initial C = 0.27	Initial C = 0
Initial R = 50	Initial R = 0.49	Initial R = 0
Variable names:	**Variable names:**	**Variable names:**
m_fModDepth_pct_L	m_fModFrequency_Hz_L	m_fFeedback_pct_L
m_fModDepth_pct_C	m_fModFrequency_Hz_C	m_fFeedback_pct_C
m_fModDepth_pct_R	m_fModFrequency_Hz_R	m_fFeedback_pct_R

```
CModDelayModule m_ModDelayLeft;
CModDelayModule m_ModDelayCenter;
CModDelayModule m_ModDelayRight;

// function to transfer out variables to it and cook
void updateModules(); // you could split this out into smaller functions
// END OF USER CODE ------------------------------------------------------------ //
```

10.5.4 StereoLCRChorus.cpp File

Constructor

There is nothing to initialize in the constructor because we have no variables; the member objects will initialize themselves at construction time. Implement the updateModules() function next. We need to set up the LFO phases and invert flags according to the block diagram. We also need to put the modules in chorus mode and call the internal update functions:

```
// updateModules()
/*
        - set LFO Phases
        - set LFO inversion (right only)
        - Mod Depths
        - Mod Freqs
        - Feedback
        - Mod Types - chorus (2)
        - LFO Types - triangle (0)
        - call updaters
*/
void CStereoLCRChorus::updateModules()
{
        // setup quad phase
```

```
        m_ModDelayLeft.m_uLFOPhase = 1; // 1: quad phase
        m_ModDelayCenter.m_uLFOPhase = 0;// 0: normal
        m_ModDelayRight.m_uLFOPhase = 1; // 1: quad phase

        m_ModDelayLeft.m_LFO.m_bInvert = false;
        m_ModDelayCenter.m_LFO.m_bInvert = false;
        m_ModDelayRight.m_LFO.m_bInvert = true; // this one is inverted

        m_ModDelayLeft.m_fModDepth_pct = m_fModDepth_pct_L;
        m_ModDelayCenter.m_fModDepth_pct = m_fModDepth_pct_C;
        m_ModDelayRight.m_fModDepth_pct = m_fModDepth_pct_R;

        m_ModDelayLeft.m_fModFrequency_Hz = m_fModFrequency_Hz_L;
        m_ModDelayCenter.m_fModFrequency_Hz = m_fModFrequency_Hz_C;
        m_ModDelayRight.m_fModFrequency_Hz = m_fModFrequency_Hz_R;

        m_ModDelayLeft.m_fFeedback_pct = m_fFeedback_pct_L;
        m_ModDelayCenter.m_fFeedback_pct = m_fFeedback_pct_C;
        m_ModDelayRight.m_fFeedback_pct = m_fFeedback_pct_R;

        // CHORUS!
        m_ModDelayLeft.m_uModType = 2;
        m_ModDelayCenter.m_uModType = 2;
        m_ModDelayRight.m_uModType = 2;

        m_ModDelayLeft.m_uLFOType = 0; // triangle
        m_ModDelayCenter.m_uLFOType = 0;// triangle
        m_ModDelayRight.m_uLFOType = 0;// triangle

        // cook them
        m_ModDelayLeft.updateLFO();
        m_ModDelayCenter.updateLFO();
        m_ModDelayRight.updateLFO();

        m_ModDelayLeft.updateDDL();
        m_ModDelayCenter.updateDDL();
        m_ModDelayRight.updateDDL();
}
```

prepareForPlay()

- Forward the calls to prepareForPlay() to the member objects.

```
bool __stdcall CStereoLCRChorus::prepareForPlay()
{
        // Add your code here:
        // call forwarding!
        m_ModDelayLeft.prepareForPlay();
        m_ModDelayCenter.prepareForPlay();
        m_ModDelayRight.prepareForPlay();

        // don't leave this out - it inits and cooks
        updateModules();

        return true;
}
```

UserInterfaceChange()

- Just do a brute force update of all modules (you can streamline this later):

```
bool __stdcall CStereoLCRChorus::userInterfaceChange(int nControlIndex)
{
	// add your code here
	// brute force update
	updateModules();

	return true;
}
```

ProcessAudioFrame()

- Split and sum the L + R to feed the center chorus.
- Split and sum the outputs (L = L + C) and (R = R + C).

```
bool __stdcall CStereoLCRChorus:: processAudioFrame(float* pInputBuffer, float*
                                                    pOutputBuffer, UINT
                                                    uNumInputChannels, UINT
                                                    uNumOutputChannels)
{
	// declare some output variables
	float fChorusOut_L = 0;
	float fChorusOut_C = 0;
	float fChorusOut_R = 0;

	// get the left and right inputs; note the setup for mono
	float fLeftIn = pInputBuffer[0];
	float fRightIn = pInputBuffer[0];

	// if stereo
	if(uNumInputChannels == 2)
		fRightIn = pInputBuffer[1];

	// form the center channel
	float fCenterIn = 0.5*fLeftIn + 0.5*fRightIn;

	// call the processAudioFrame()
	m_ModDelayLeft.processAudioFrame(&fLeftIn, &fChorusOut_L, 1);
	m_ModDelayCenter.processAudioFrame(&fCenterIn, &fChorusOut_C, 1);
	m_ModDelayRight.processAudioFrame(&fRightIn, &fChorusOut_R, 1);


	// sum to create Left Out
	pOutputBuffer[0] = fChorusOut_L + fChorusOut_C;

	// Mono-In, Stereo-Out (AUX Effect)
	if(uNumInputChannels == 1 && uNumOutputChannels == 2)
		pOutputBuffer[1] = pOutputBuffer[0];

	// Stereo-In, Stereo-Out (INSERT Effect)
	if(uNumInputChannels == 2 && uNumOutputChannels == 2)
		pOutputBuffer[1] = fChorusOut_R + fChorusOut_C;

	return true;
}
```

Build and test the code. Try to find the most interesting settings for different instruments then go back and set your own initial values accordingly. Be sure to save lots of presets—this plug-in can produce a wide variety of sounds.

10.6 More Modulated Delay Algorithms

In addition to the extra algorithms in Sections 10.1 and 10.2, next are some more designs for you to experiment with. Remember to figure out the UI first then code it using the modules we've already built.

10.6.1 Stereo Cross-Flanger/Chorus (Korg Triton®)

Like the quadrature version, the only difference between the flanger and chorus is in the module setting—the same block diagram is used for both. The LFO can be engaged in normal or quadrature phase (Figure 10.14).

10.6.2 Multi-Flanger (Sony DPS-M7®)

The DPS-M7 has some intensely thick modulation algorithms. This one has two flanger circuits that can be combined in parallel or series on either channel. The channels are also cross mixable. Each module marked “flanger” contains a complete flanger module: depth, rate, feedback, and wet/dry. Note also the use of pre-delays with feedback too. All LFOs and pre-delays are independent and fully adjustable (Figure 10.15).

10.6.3 Bass Chorus

The bass chorus in Figure 10.16 splits the signal into low-frequency and high-frequency components and then only processes the high-frequency component. This leaves the fundamental intact. The comb filtering of the chorus effect smears time according to how much delay is being used. For the bass guitar, this results in a decreased fundamental with an ambiguous transient edge or starting point. Because bass players need to provide a defined beat, the bass chorus will preserve this aspect of the instrument. If you want to implement this effect, use the Linkwitz–Riley low-pass and high-pass filters to split the signal. Invert the output of one of the filters—it doesn't matter which one—so that their phase responses sum properly.

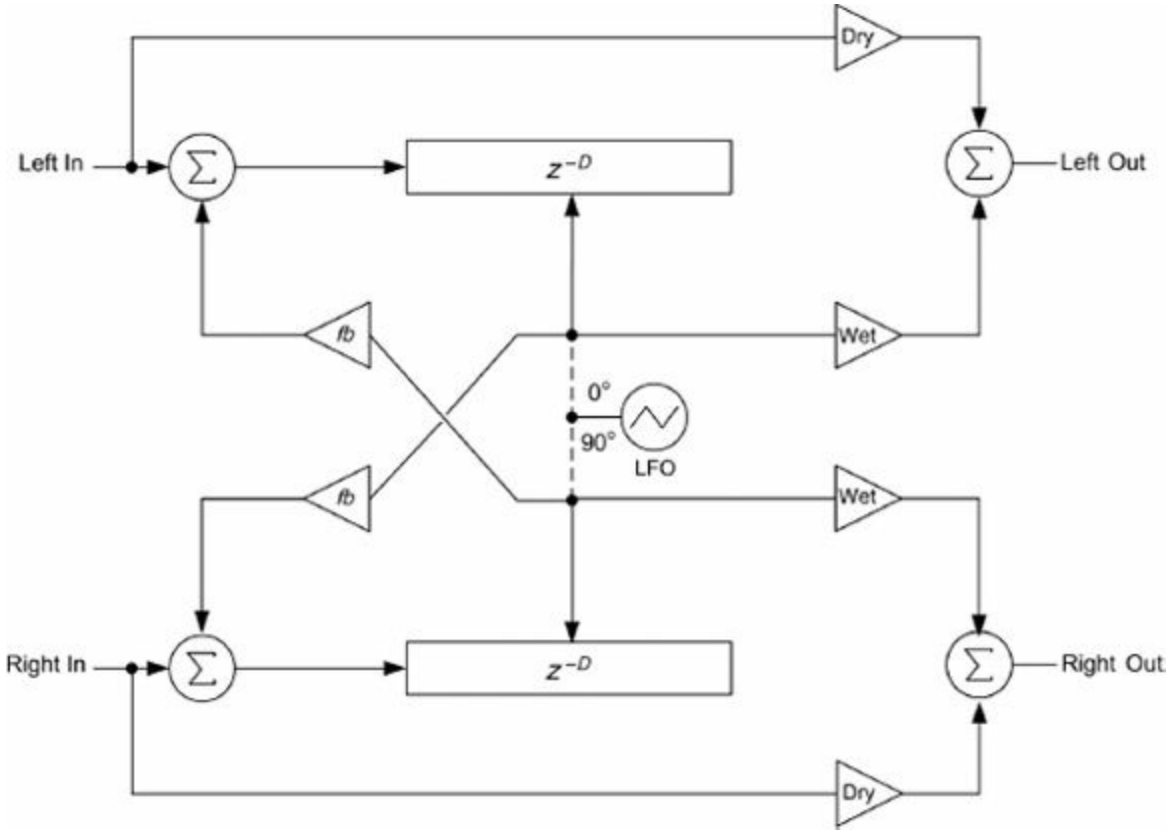


Figure 10.14: A stereo cross-flanger/chorus.

10.6.4 Dimension-Style (Roland Dimension D®)

This chorus unit (Figure 10.17) is based on the Roland Dimension D® chorus. Known for its subtle transparent sound, it features a shared but inverted LFO and an interesting output section where each output is a combination of three signals:

1. Dry
2. Chorus output
3. Opposite channel chorus output, inverted and high-pass filtered

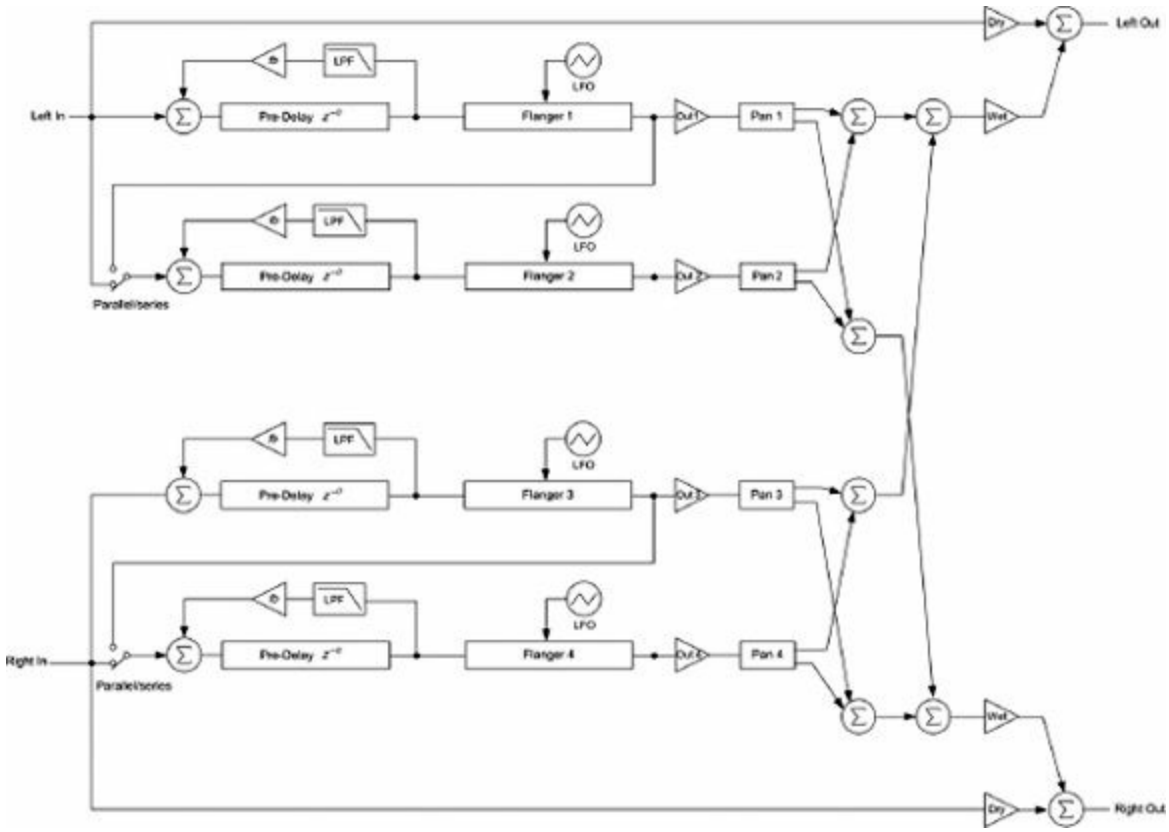


Figure 10.15: DPS M7 multi-flanger.

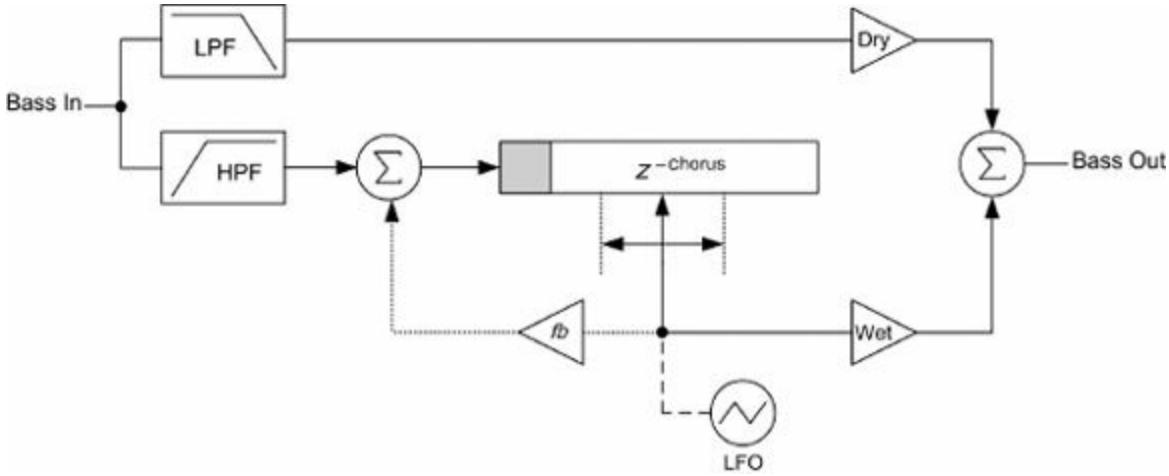


Figure 10.16: A bass chorus.

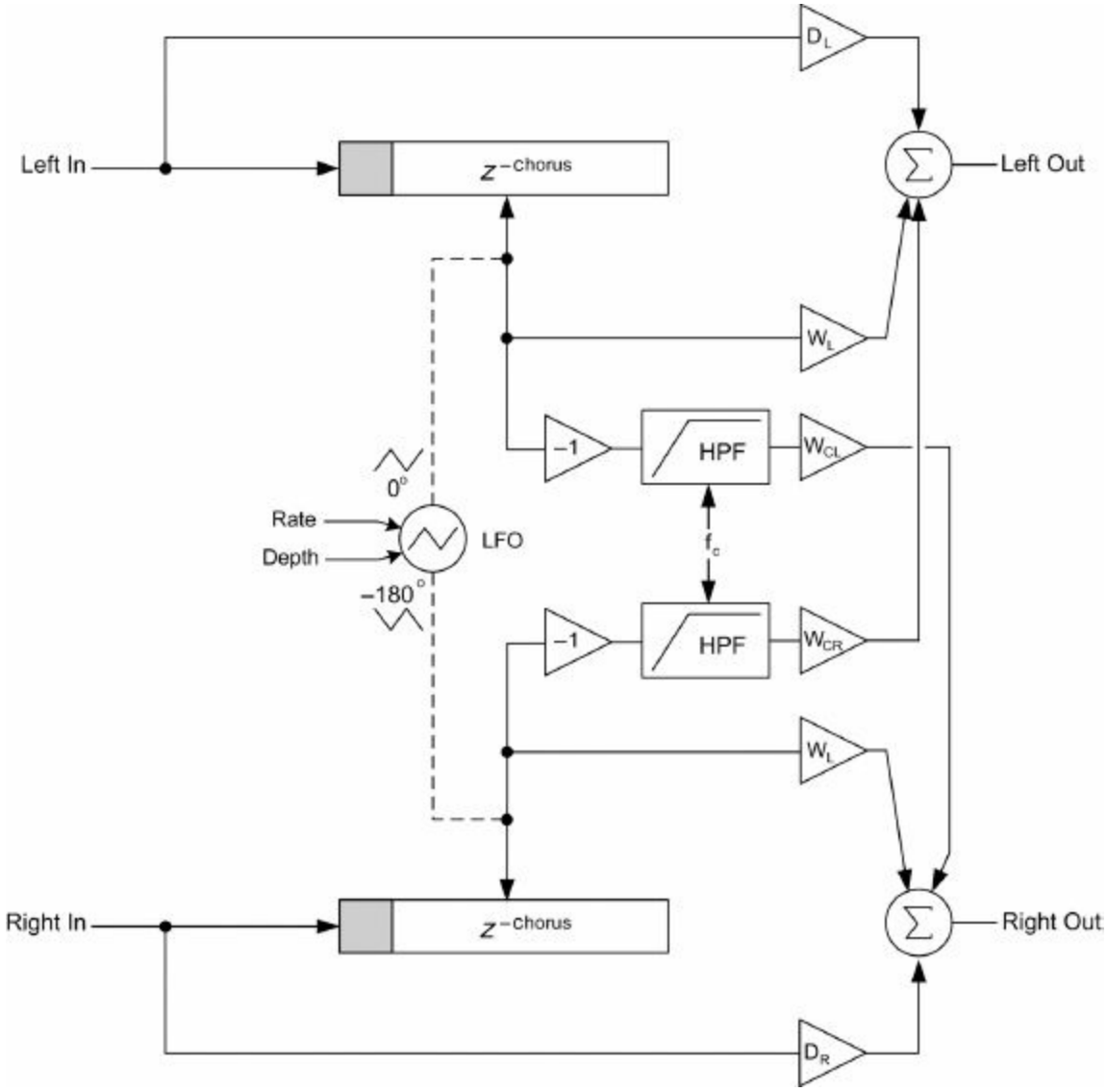


Figure 10.17: A dimension-style chorus.

The controls on the original unit consisted of four switches only; these were hardwired presets and changed

- The LFO rate (either 0.25 Hz or 0.5 Hz)
- The depth
- For one preset, the wet/dry mix ratios for the two channels (W_L vs D_L, W_R vs D_R)

Here I have added two more level controls for experimentation, W_{CL} and W_{CR}, which are the filtered, inverted chorused signals from the opposite (crossed) channels. Experiment with various rate, depth, and output mixture combinations as well as different high-pass filter cutoff frequencies.

10.6.5 Deca-Chorus (Sony DPS-M7®)

The deca-chorus has 10 (deca) chorus units, 5 per channel. Each chorus has its own pre-delay, LFO, gain, and pan control. It can also run in a mono mode (10 chorus units in parallel) (Figure 10.18).

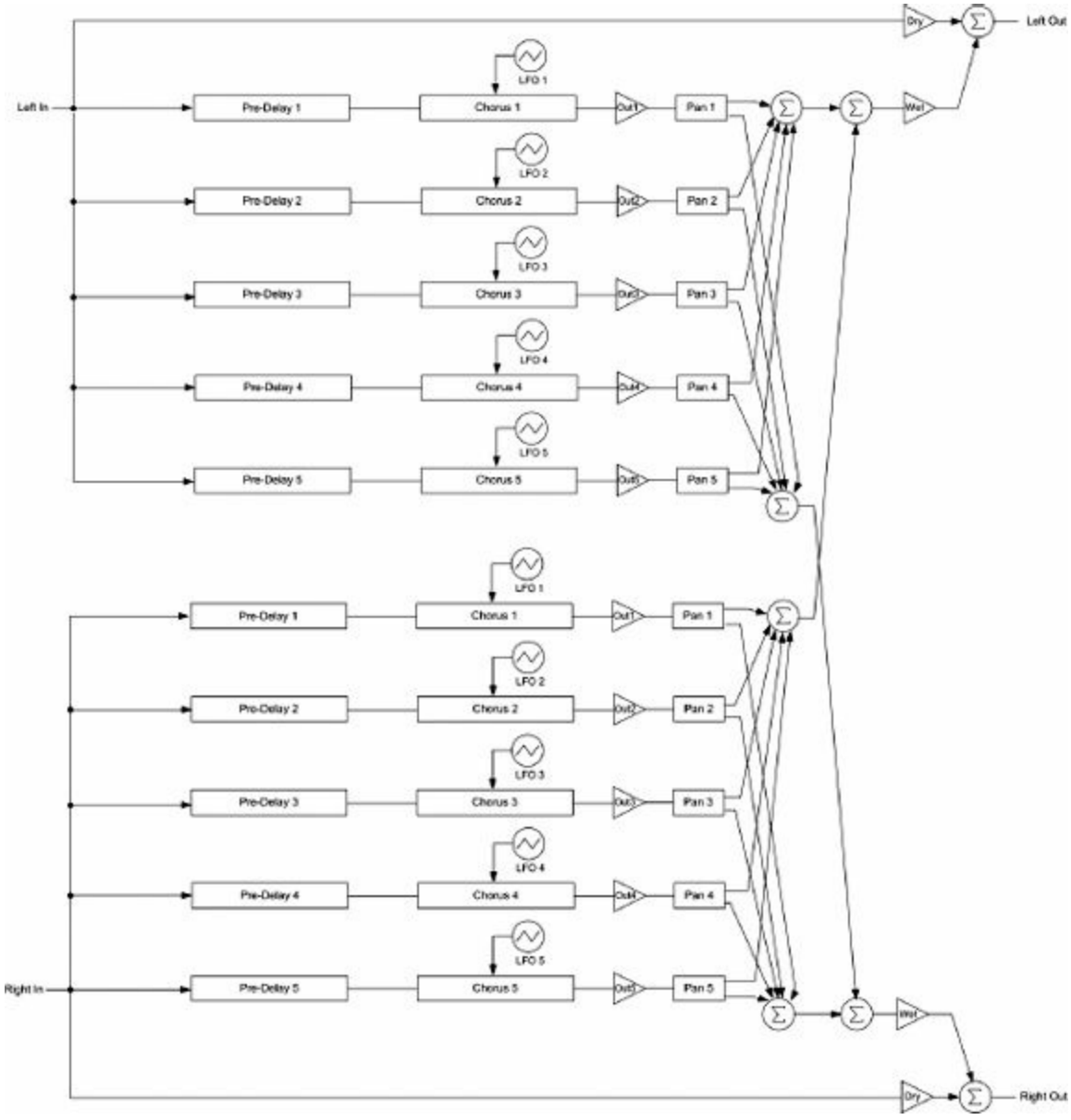


Figure 10.18: DPS-M7 deca chorus.

Bibliography

Cole, M. 2007. "Roland dimension C clone for Eventide 7000, Orville and H8000." Accessed August 2012 from http://www.eventidestompboxes.com/forummedia/PATCHES/Orville/ProFXalgorithms/Roland%20Dimension%20C.doc.

Coulter, D. 2000. *Digital Audio Processing*, Chapter 11. Lawrence, KS: R&D Books.

Dattorro, J. 1997. Effect design part 2: Delay line modulation and chorus. *Journal of the Audio Engineering Society*, 45(10): 764–786.

Korg, Inc. 2000. *Triton Rack Parameter Guide*. Tokyo: Korg Inc.

Phillips, D. 1991. *Wavestation SR Reference Guide*. Tokyo: Korg Inc.

Roads, C. 1996. *The Computer Music Tutorial*, Chapter 3. Cambridge, MA: The MIT Press.

Roland Corporation. 1990. "Boss CH-1 schematics." Accessed August 2012 from http://superchorusmods.blogspot.com/2009/08/schematics.html.

Sony, Corp. 1995. *DPS-M7 Manual*. Tokyo: Sony Corp.

White, P. 2001. "Understanding and emulating vintage dffects." *Sound on Sound Magazine*. Accessed August 2012 from http://www.soundonsound.com/sos/jan01/articles/vintage.asp.

CHAPTER 11

Reverb Algorithms

Reverb algorithms might represent the Holy Grail of audio signal processing. They have an appeal that seems universal, perhaps because we live in a reverberant world. Our ears are time-integrating devices that use time domain cues and transients for information, so we are sensitive to anything that manipulates these cues. In this chapter, we discuss reverb algorithms as applied mainly to room simulation. There are two general ways to create the reverberation effect:

- Reverb by direct convolution—the physical approach.
- Reverb by simulation—the perceptual approach.

In the physical approach, the impulse response of a room is convolved with the input signal in a large finite impulse response (FIR) filter. For large rooms, these impulse responses might be = to 10 seconds. In the mid 1990s, Gardner developed a hybrid system for fast convolution that combined direct convolution with block fast Fourier transform (FFT) processing (Gardner 1995). Around the same time, Reilly and McGrath (1995) described a new commercially available system that could process 262,144-tap FIR filters for convolving impulses over = seconds in length. The required processing power is sometimes too large to be practical in a plug-in. Aside from the computing expense, another drawback to this approach is that an impulse response is frozen in time, and measures the room at one location only, under a certain set of conditions. To create a reverb unit (or *reverberator*) that is general enough to provide many different reverbs of different spaces would require a large library of impulse response files. In addition, its parameters can't be adjusted in real time.

The perceptual approach aims to simulate the reverberation with enough quality to fool the ears and give the same perception of real reverb but with much less processing power. The advantages are numerous, from the minimal processing required to the ability to vary many parameters in real time. Browne (2001) proposed a hybrid system combining a short impulse response convolution along with recursive filtering, combining both approaches. There are several key engineers who developed much of the theory still in use today. These include Schroeder's (1962) initial work in the early 1960s with continued research and contributions from Moorer, Griesinger, Gerzon, Gardner, Jot, Chaigne, Smith, Roscchesso, and others across the decades. Most of this chapter is owed to their work in the field. We will focus on the perceptual approach and try to find computationally efficient algorithms for interesting reverberation effects.

Griesinger (1989) states that it is impossible to perfectly emulate the natural reverberation of a real room and thus the algorithms will always be approximations. It seems that the area of reverberation design has the most empirically derived or trial-and-error research of just about any audio signal processing field. There is no single “correct” way to implement a reverb algorithm, so this chapter focuses on giving you many different reverberator modules to experiment with.

11.1 Anatomy of a Room Impulse Response

The first place to start is by examining impulse responses of actual rooms. There are several popular methods for capturing the impulse response, from cap pistols and balloons to deconvolution of chirp signals and pseudo random sequences. The resulting time domain plot is useful for investigating the properties of reverberation.

Figure 11.1 shows the impulse response plots for two very different spaces; a large concert hall and a cathedral. The initial impulse is followed by a brief delay called the *pre-delay*. As the impulse pressure wave expands, it comes into contact with the nearby structures—walls, floor, and ceiling—and the first echoes appear. These initial echoes, called *early reflections*, are important to the simulation of reverb because of the auditory cues we get from them. The pressure wave continues to expand and more reflections occur, with reflected signal upon reflected signal piling on top of each other while decaying in energy. The resulting reverb "tail" is called *late reverberation* or *subsequent reverberation*.

The top impulse response in Figure 11.1 is a modern concert hall designed for a pleasing reverberant quality. The initial reflections are from irregularly shaped back and side walls and they pile up in amplitude; they are also piling up in density, shown as the impulse gets "fatter" in the middle section. The dense reverberant tail follows, which decays in an exponential fashion, and the sound pressure energy is absorbed by the seats and acoustical room treatments. The cathedral is a different story. After a short pre-delay, a few large early reflections arrive from the nearby structures but they don't increase in amplitude or density in the same way as the hall. The reverb's decay is also much longer in time as there is little in the cathedral to absorb the sound pressure.

The block diagram in Figure 11.2 shows the three components we observe. However, there is debate as to whether or not this is a good way to break the problem down. Some feel that the reverb algorithm should not need to separate out the early reflections from the late reverberation. In other words, a good algorithm will create all of the reverberation aspects at once.

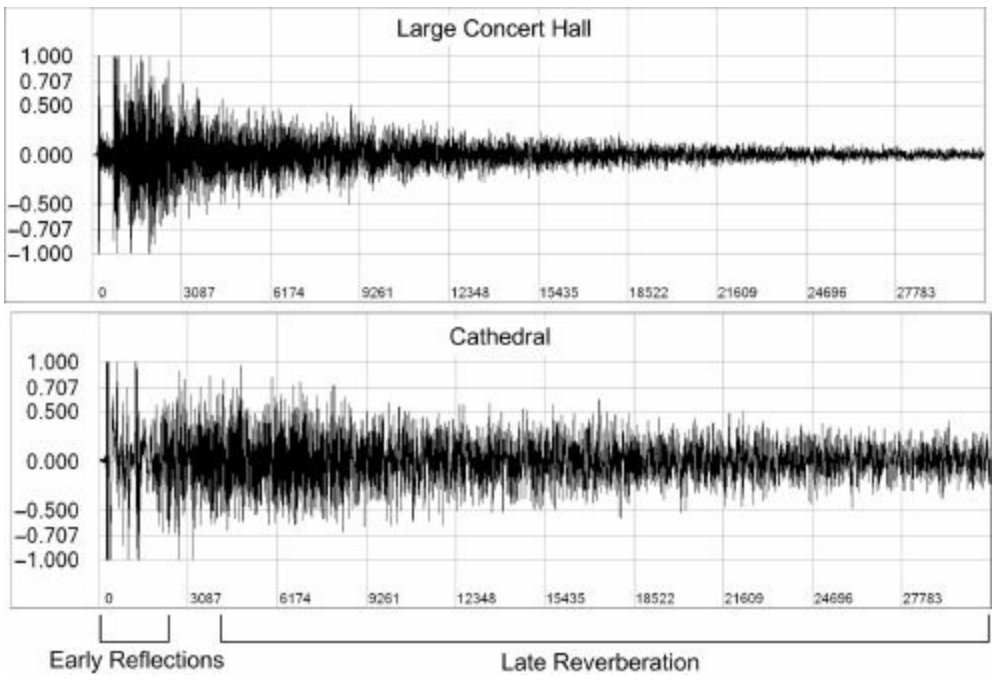


Figure 11.1: The impulse responses of a large hall and cathedral.

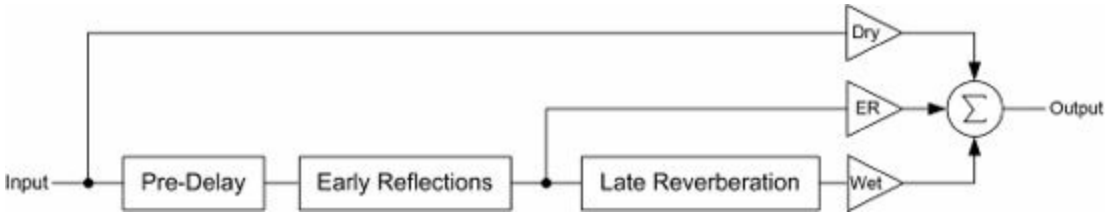


Figure 11.2: A generalized model of a reverb algorithm.

11.1.1 RT_{60}: The Reverb Time

The most common measurement for reverb is the RT_{60} reverb time. Reverb time is measured by first stimulating the room into a reverberant state, then turning off the source and plotting the pressure-squared level as a function of time. The amount of time it takes for this energy decay curve to drop by 60 dB is the *reverb time*, or RT_{60}. Sabine's (1973) pioneering work in this area leads to the following formula in Equation 11.1:

$$RT_{60} = 0.5\frac{V_R}{S_R A_{RAve}} \tag{11.1}$$

where

V_R = volume of room in ft^3

S_R = surface area of room in ft^2

A_{RAve} = average absorption coefficient

Sabine measured and tabulated the absorption coefficients for various materials. The units are given in *Sabines*. A room made of several materials is first partitioned to find the partial surface area of each material type, then the average is found by weighting the areas with the absorption coefficients and summed. The reverb time is probably the most common control, found on just about every kind of reverb plug-in.

11.2 Echoes and Modes

In Schroeder's early work, he postulates that a natural sounding artificial reverberator has both a large echo density and a colorless frequency response. The echo density is simply the number of echoes per second that a listener experiences in a given position in the reverberant environment (Equation 11.2).

$$E_D = \frac{\text{echoes}}{\text{second}} \tag{11.2}$$

If the echo density is too low the ear discerns the individual echoes and a fluttering sound is the result. As the echo density increases, the echoes fuse together, producing a wash of sound. Schroeder (1962) postulated that for a natural-sounding reverb, the echo density needs to be 1000 echoes/second. Greisinger (1989) increases this to 10,000 echoes/second or more. Statistically, the

echo density is shown in Equation 11.3, which reveals that the echoes build up as the square of time

$$\text{Echo density} \propto \frac{4\pi c^3}{V_R} t^2 \qquad (11.3)$$

where

c = speed of sound
t = time
V_R = volume of room

Reverb is often modeled statistically as decaying white noise, which implies that ideal reverberant rooms have flat frequency responses. A room's geometry can be used to predict its frequency response. The dimensions of the room can be used to calculate the frequencies that will naturally resonate as the wave bounces back and forth between parallel surfaces. An untreated rectangular room will have multiple resonances and anti-resonances.

Given a rectangular room with length, width, and height of l, w, and h, the resonances are found in Equation 11.4, a well-known and useful equation for predicting room resonances, also called *modes* (Beranek 1986). As the frequency increases, so do the number of resonances.

$$f_e = \frac{c}{2}\sqrt{\left(\frac{n_x}{l}\right)^2 + \left(\frac{n_y}{w}\right)^2 + \left(\frac{n_z}{h}\right)^2} \qquad (11.4)$$

where

n_x, n_y, n_z = half wave numbers 0, 1, 2, 3...
l, w, h = length(x), width(y) and height(z) of the room

Above a certain frequency the resonances overlap and fuse together. Each resonant frequency has its own envelope whose curve is bell (band-pass filter [BPF]) shaped, meaning that it has a peak resonance at the mode frequency, but is still excitable at frequencies around it and thus the quality factor (Q) of the curve relates to this excitability. The number of resonances increases with frequency. Figure 11.3 shows a fictitious room example with the bell-shaped resonances.

The *modal density* is the number of resonant peaks per Hz. Physicists call the resonant frequencies or modes *eigenfrequencies* (note this is not an acoustics-specific term; an eigenfrequency is the resonant frequency of any system). Schroeder's second postulation is that for a colorless frequency response, the modal density should be 0.15 eigenfrequencies/Hz or one eigenfrequency every 6.67 Hz or approximately 3000 resonances spread across the audio spectrum. Thus, it makes sense that good reverberant environments have interesting geometries with many nonparallel walls. The multitude of resonances is created by the many paths an impulse can take from the source to the listener. Kuttruff (1991) derived the approximation for the modal density as it relates to the volume of the room and modal frequency. Equation 11.5 shows that the resonances build up as the square of frequency.

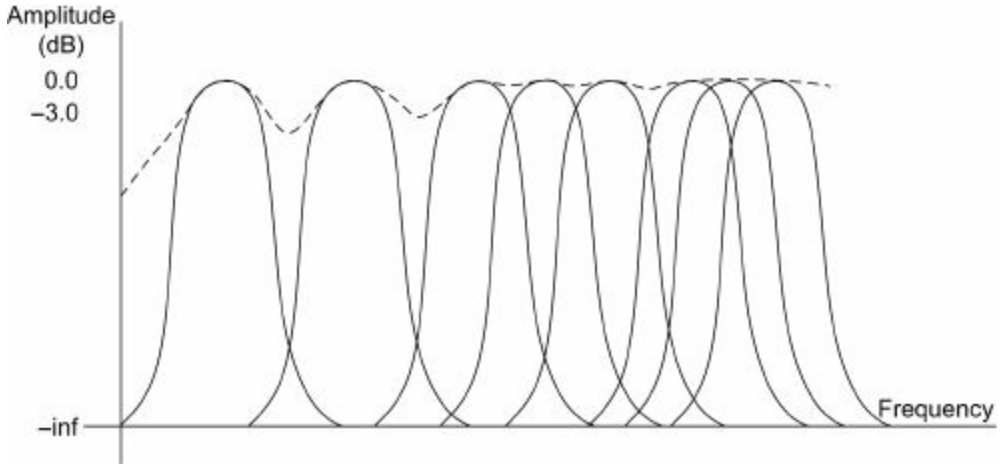


Figure 11.3: The resonances of a room all contribute to create the final frequency response. When the bandwidths of the modes overlap, they fuse together. The dotted line represents the combined frequency response.

Schroeder's rules for natural reverb:

- Echo density: At least 1000 echoes/sec (Greisinger: 10,000 echoes/sec)
- Modal density: At least 0.15 eigenfrequencies/Hz

In physical rooms we know that:

- Echo density increases with the square of time.
- Modal density increases with the square of frequency.

$$D_m = \frac{4\pi V_R}{c^3} f_m^2 \qquad (11.5)$$

where

V_R = volume of the room

f_m = modal frequency in question

The *energy decay relief* plot (or EDR) shows how the energy decays over both frequency and time for a given impulse response of a room. Figure 11.4 shows a very simple fictitious EDR.

In Figure 11.4, notice that the frequency axis comes out of the page; low frequencies are in the back. It also shows that this room has a resonant peak, which forms almost right away in time. In an EDR, the modal density is shown across the *z*-axis (frequency) while the echo density appears across the *x*-axis (time). This simplified plot shows just one resonance and no echo density build-up for ease of viewing. Figure 11.5 shows the EDR of an actual room, in this case a theater. The eigenfrequencies are visible as the ridges that run perpendicularly to the frequency axis.

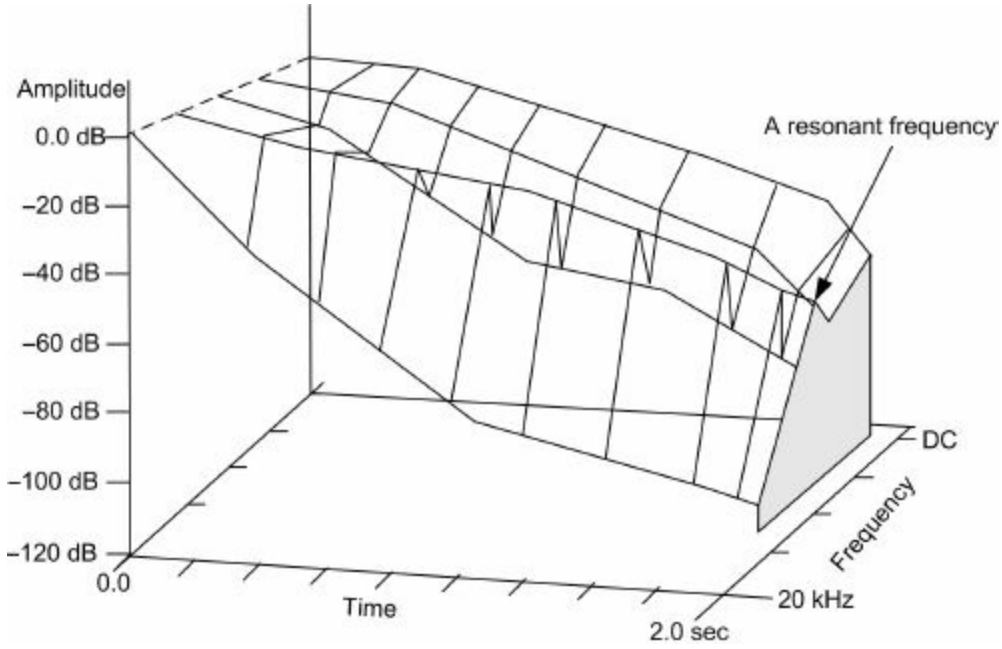


Figure 11.4: The EDR shows time (*x*-axis), frequency (*z*-axis) and amplitude (*y*-axis) of the energy decay of a room.

Figure 11.6 shows the EDR for a cathedral. In this case, the echo pile-ups are clearly visible running perpendicular to the time axis. Comparing both EDRs shows that both rooms have high echo and modal densities; therefore, they should be good reverberant spaces. Both EDRs show an initial high frequency roll-off and, especially in the theater's case, the high frequencies decay faster than the low frequencies. The high-frequency decay is a property of the treatment of the room surfaces along with the fact that the air absorbs high frequencies more easily than low frequencies. The high-frequency energy decay in the theater is caused by the carpet, seats, and acoustic treatment which the cathedral lacks. Therefore, for a good quality reverb, we will need to take this high-frequency decay into account along with the echo and modal density.

Reverb algorithms are typically made of arrangement of modules called *reverberator modules*. From our basic observations we can tell that good reverberator modules are going to produce dense echoes along with a large number of resonances. If the resonances are distributed properly across the spectrum, the reverb will sound flat. If they are not, there will be metallic pinging and other annoyances that will color the frequency response in an unnatural way. The majority of the rest of the chapter is devoted to revealing, analyzing, and explaining these building blocks and the algorithms that use them. The best place to start is with Schroeder's reverb modules.

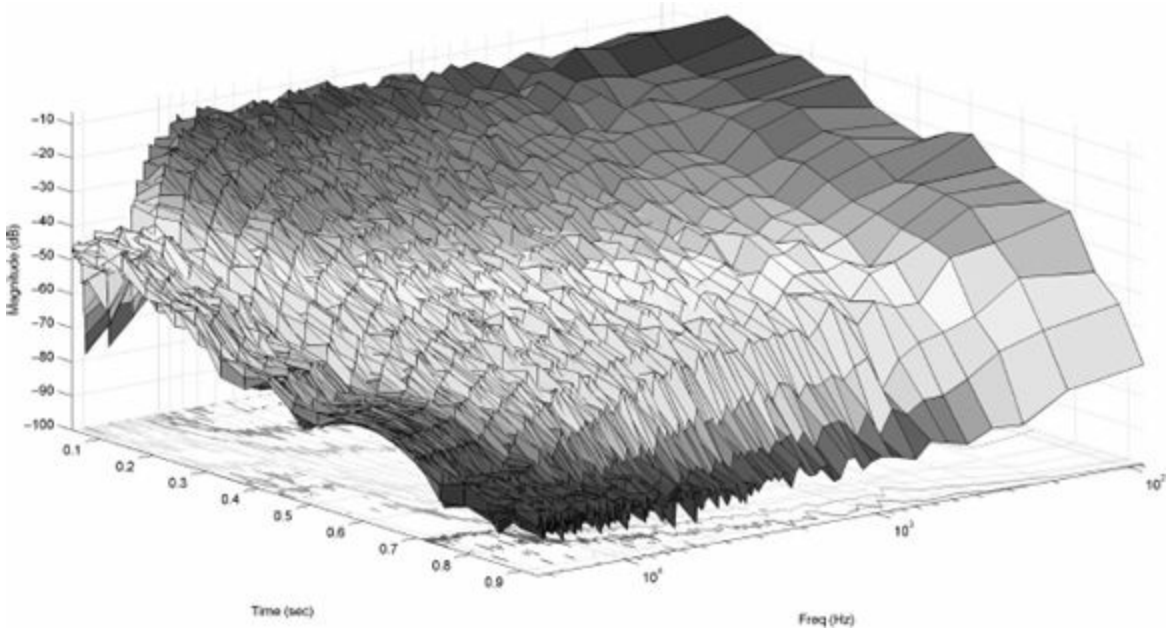


Figure 11.5: An EDR for a theater.

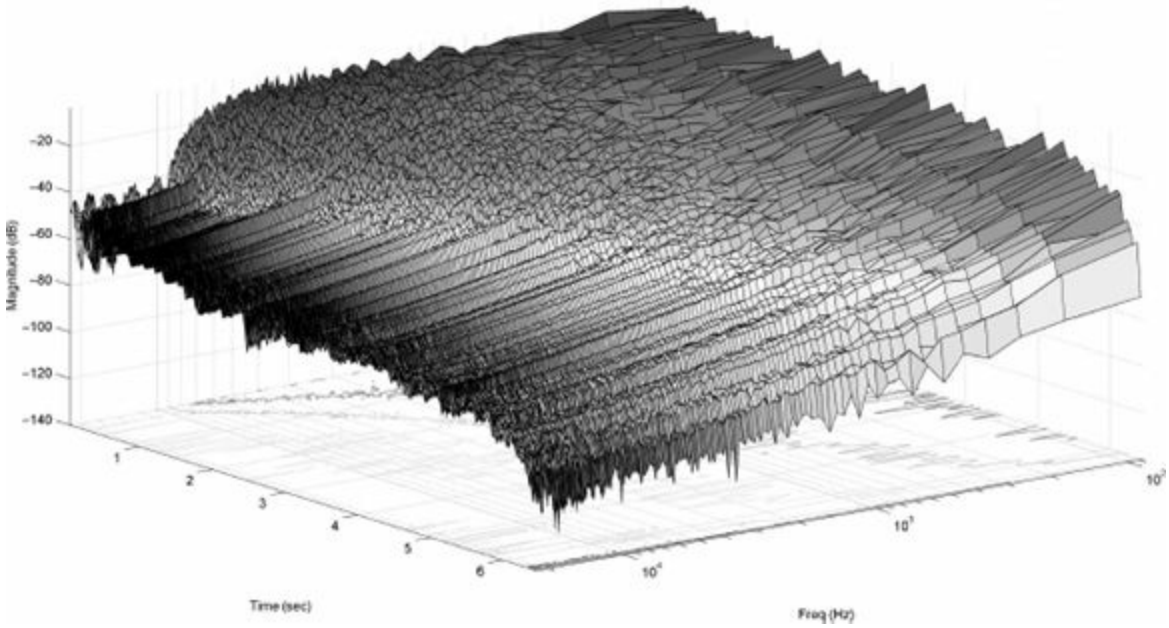


Figure 11.6: An EDR for a cathedral.

11.3 The Comb Filter Reverberator

One of the reverberator modules that Schroeder proposed is a comb filter. Remember that we are looking for modules that produce echoes and the comb filter with feedback will do just that. The comb filter reverberator is familiar looking because it's also the delay with feedback algorithm you studied in Chapter 7. We derived the difference equation and transfer function for the comb filter in Figure 11.7 in Chapter 7:

$$y(n) = x(n-D) + gy(n-D)$$

and (11.6)

$$H(z) = \frac{z^{-D}}{1-gz^{-D}}$$

We also performed a pole-zero analysis and generated frequency plots. We showed that the feedback path caused a series of poles evenly spaced around the inside of the unit circle. The resulting impulse response is easy to predict as the echo recirculates through the feedback path, being attenuated by multiplying by *g* each time through the loop.

While Figure 11.8 might look simple, the results certainly trend in the right direction. The frequency response plot shows a set of resonant peaks that could correspond to a room's eigenfrequencies and the impulse response shows a set of decaying impulses, mimicking the energy loss as the echoes bounce off of surfaces. The modal density for a comb filter is given by Jot and Chaigne (1991) in Equation 11.7.

$$M_d = \frac{D}{f_s}$$
$$\Delta f = \frac{f_s}{D} \qquad (11.7)$$

where

D = the delay length

f_s = the sample rate

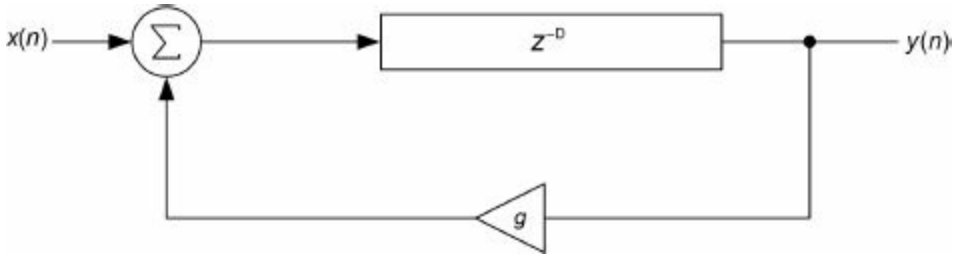


Figure 11.7: The basic comb filter.

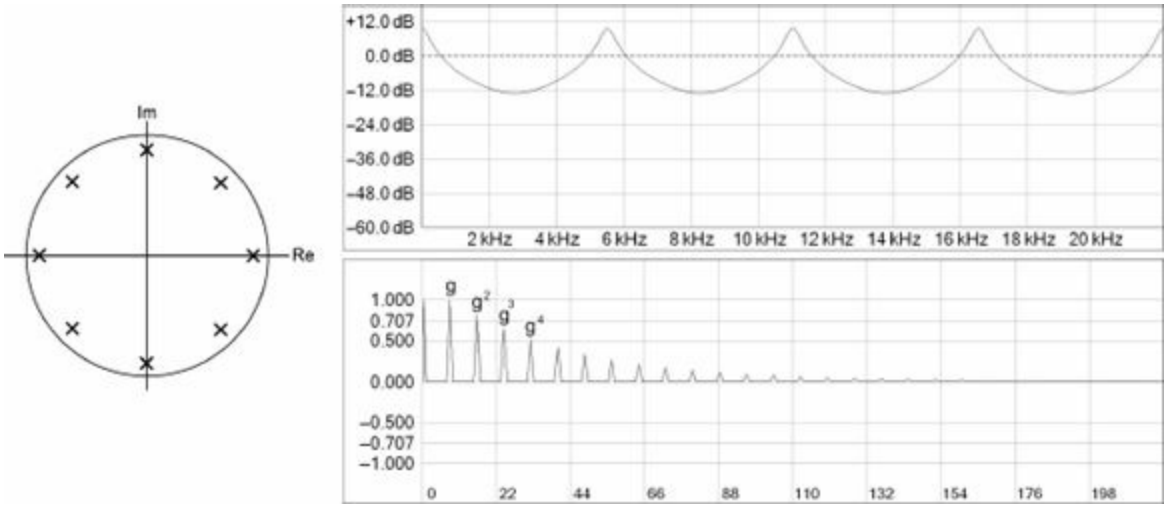


Figure 11.8: The poles in the *z*-plane produce the resonances. The feedback that creates those poles also recirculates and scales the echoes by *g*. In this example, *D* = 8 samples, feedback = 80%.

So, the comb filter produces D resonances across DC to f_s (or $D/2$ resonances from DC to Nyquist) with each resonance separated by $\Delta f = f_s/D$. This is exactly what we found when we analyzed the comb filter in Chapter 7. In order to get Schroeder's desired 0.15 eigenfrequencies/Hz from DC to 20 kHz at f_s = 44,100, you would need D = 6622. This also makes sense because we need about 3000 ($D/2$) resonances to cover the range from DC to 20 kHz. The problem is that while we will get the desired number of resonances, they will be linearly spaced rather than piling up as the square of frequency, as Equation 11.5 predicts. Additionally, it will take 6622 samples (~150 mSec with a 44.1 kHz sample rate) before the reverberator begins outputting samples. Schroeder was able to achieve high density by placing comb filters in parallel, then summing the results. This would then cause the comb filter responses to overlap, however, care must be taken when choosing the delay lengths—if they are mathematically related then resonances or anti-resonances will color the sound.

Figure 11.9 shows an example of four comb filters in parallel. Each has its own delay time ($D1$–$D4$) and gain ($g1$–$g4$). However, care must be taken with the gain values. As you remember from Chapter 7, the value of g alone determines the pole radii. If the poles of each filter have the same radii, their impulses will decay at the same rate. This is desirable since it mimics an uncolored reverberation pattern. If the pole radii are different for the different comb filters then the resulting reverberation will have undesirable coloration in it. The pole radii are given in Equation 11.8 by Gardner (Kahrs and Bradenberg 1998). The relationship between the pole radii, the delay D and the reverb time RT_{60} is given in Equation 11.9.

$$r = g^{1/D} \tag{11.8}$$

where

g = the feedback gain
D = the delay length

$$RT_{60} = \frac{3DT_s}{\log(1/g)}$$

or

$$1/g = 10^{\frac{3DT_s}{RT_{60}}}$$

or

$$g = 10^{\frac{-3DT_s}{RT_{60}}} \tag{11.9}$$

T_s is the sample period

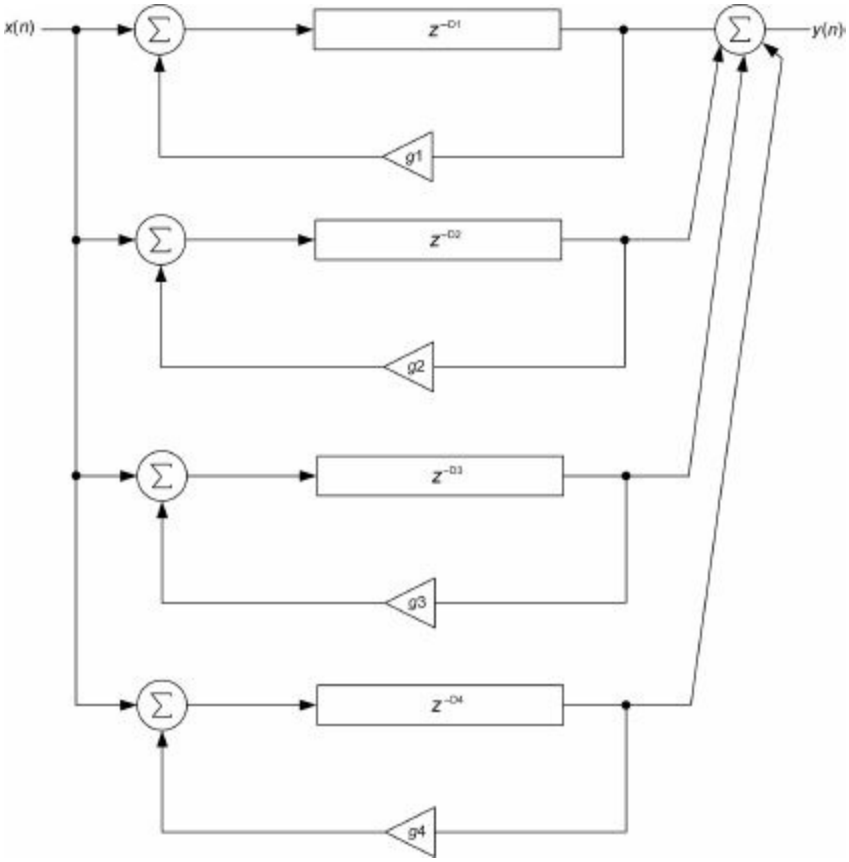


Figure 11.9: Four comb filters in parallel.

This means we can control the reverb time by using the gain factor *or* the delay length *D*. The tradeoff is that if we increase *g* to increase reverb time, the poles get very near the unit circle, causing resonances. If we increase *D*, then the echoes become distinctly audible rather than smearing together. This means that there is a tradeoff of the modal density versus the echo density in the design. The modal density for a parallel bank of *N* comb filters in Equation 11.10 is given by Jot (1991). From Equation 11.10 you can see that the modal density remains constant across all frequencies, which is not what happens in real rooms where the modal density increases with the square of the frequency (Equation 11.5).

$$M_d = \sum_{i=0}^{N} D_i T_s$$

where

D_i = the delay length of the ith comb filter
T_s = the sample period (11.10)

or

$$M_d = N\overline{D}T_s$$

where

$\overline{D}$ = the mean delay length of all filters averaged

The echo density for the parallel combs with delay lengths close to each other is given in Equation 11.11 (Jot 1991).

$$E_d = \sum_{i=0}^{N} \frac{1}{D_i T_s}$$

where

D_i = the delay length of the ith comb filter

or

$$E_d = \frac{N}{\overline{D} T_s} \tag{11.11}$$

where

$\overline{D}$ = the mean delay length of all filters averaged

Knowing the desired M_d and E_d you can then calculate the number of parallel comb filters N with average delay time DT_s with Equation 11.12.

$$N = \sqrt{E_d M_d}$$
$$\overline{D} T_s = \sqrt{M_d / E_d} \tag{11.12}$$

Plugging Schroeder's values of $M_d = 0.15$ eigenfrequencies/Hz and $E_d = 1000$ echoes/sec into Equation 11.12 yields $N = 12$ and the average delay time $DT_s = 12$ mSec.

11.4 The Delaying All-Pass Filter Reverberator

Schroeder also proposed the delaying all-pass filter (APF) as a reverberator unit. The impulse response is a decaying set of echoes but the frequency response is technically flat. Schroeder's APF is shown in Figure 11.10. In Figure 11.11, you can see that we get echoes that decay faster than the comb filter and at a different rate. It has a flat magnitude response due to the APF reciprocal zero/pole radii. However, as Gardner points out, our ears only perform a short-time integration, whereas the APF requires an infinite integration time to achieve the flat response, so we will still hear some timbral coloration in this unit.

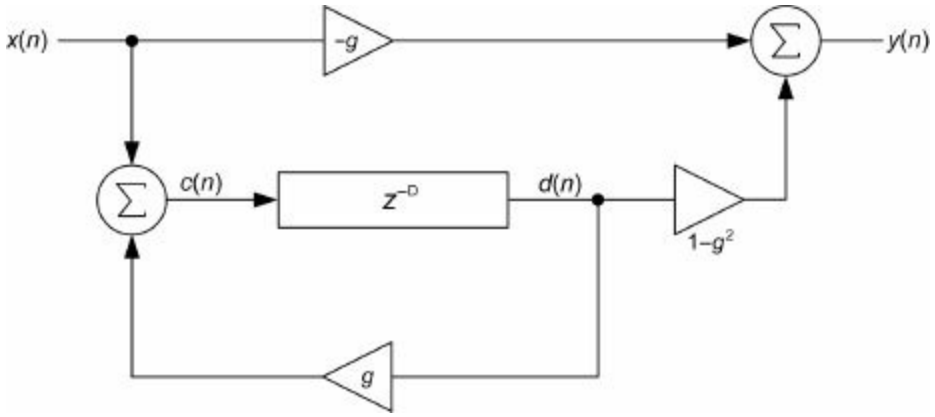


Figure 11.10: Schroeder's APF reverberator.

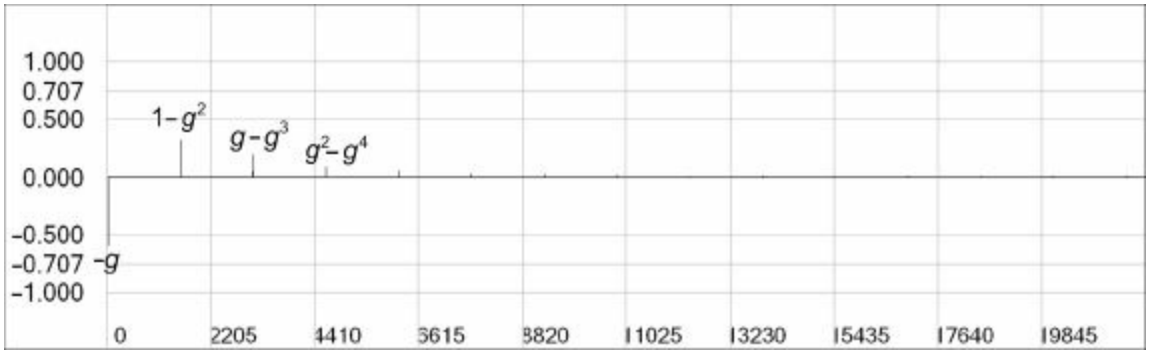


Figure 11.11: Impulse response of Schroeder's APF reverberator; the frequency response is flat. The delay time was $D = 35$ mSec with $g = 0.6$.

Inspection of the block diagram in Figure 11.10 reveals that this is a rather complex feed-back/feed-forward structure. We need to extract the difference equation so we can synthesize the reverb unit. To start, we label the nodes $c(n)$ and $d(n)$ in the block diagram, then fashion the output $y(n)$ with respect to them:

$$\begin{aligned} c(n) &= x(n) + gd(n) \\ d(n) &= c(n)z^{-D} \\ y(n) &= -gx(n) + (1-g^2)d(n) \end{aligned} \tag{11.13}$$

Now, we expand out the internal nodes and use the familiar time and frequency shifting properties of the z transform to continue Equation 11.14.

$$\begin{aligned} d(n) &= c(n)z^{-D} \\ &= x(n-D) + gd(n-D) \\ y(n) &= -gx(n) + (1-g^2)[x(n-D) + gd(n-D)] \end{aligned} \tag{11.14}$$

Examining the last term in Equation 11.14, $d(n-D)$, we can rearrange Equation 11.13 to continue and get Equation 11.15:

$$\text{so} \qquad d(n) = \frac{y(n) + gx(n)}{(1-g^2)} \qquad d(n-D) = \frac{y(n-D) + gx(n-D)}{(1-g^2)} \tag{11.15}$$

Substituting Equation 11.15 back into Equation 11.14 you get the difference equation in Equation 11.16:

$$\begin{aligned} y(n) &= -gx(n) + (1-g^2)\left[x(n-D) + g\left(\frac{y(n-D) + gx(n-D)}{(1-g^2)}\right)\right] \\ &= -gx(n) + (1-g^2)x(n-d) + gy(n-D) + g^2x(n-D) \\ &= -gx(n) + x(n-d) + gy(n-D) \end{aligned} \tag{11.16}$$

11.5 More Delaying All-Pass Filter Reverberators

There are multiple ways to synthesize the delaying APF. Figure 11.12 shows another one. D eriving

the difference equation is easier for this version; we only need to define one extra node, $w(n)$ in Equation 11.17:

$$\begin{aligned} w(n) &= x(n) + gw(n-D) \\ y(n) &= -gw(n) + w(n-D) \\ &= -g[x(n) + gw(n-D)] + [x(n-D) + gw(n-2D)] \\ &= -gx(n) - g^2 w(n-D) + x(n-D) + gw(n-2D \end{aligned} \tag{11.17}$$

grouping:

$$= -gx(n) + x(n-D) - g^2 w(n-D) + gw(n-2D)$$

In order to finish, find $y(n-D)$ from the second line in Equation 11.18 and notice that this is the same as the last two terms in Equation 11.17.

$$y(n) = -gw(n) + w(n-D)$$

then

$$y(n-D) = -gw(n-D) + w(n-2D) \tag{11.18}$$

and

$$gy(n-D) = -g^2 w(n-D) + gw(n-2D)$$

Thus, the last two terms in Equation 11.17 can be replaced by $gy(n-D)$, and the final difference equation matches Schroeder's APF:

$$y(n) = -gx(n) + x(n-D) + gy(n-D) \tag{11.19}$$

Figure 11.13 shows another version of the same APF. The proof that it has the same difference equation is easy if you look at the node $w(n)$ in Equation 11.20:

$$\begin{aligned} w(n) &= x(n) + gw(n-D) \\ y(n) &= -gw(n) + w(n-D) \end{aligned} \tag{11.20}$$

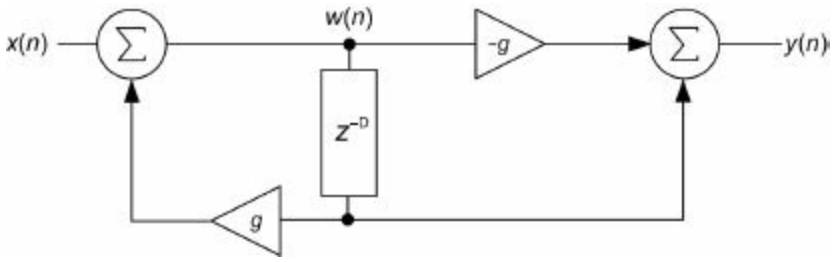


Figure 11.12: Another version of the delaying all-pass reverberator.

These are the first two lines in the derivation for the delaying APF above. Figure 11.14 shows yet another structure. To figure this one out, once again find the internal node $w(n)$ as it relates to the output in Equation 11.21:

$$\begin{aligned} w(n) &= x(n) + gy(n) \\ y(n) &= -gx(n) + w(n-D) \end{aligned}$$

then

$$w(n-D) = x(n-D) + gy(n-D) \tag{11.21}$$

and

$$y(n) = -gx(n) + x(n-D) + gy(n-D)$$

Equation 11.21 is identical to Equation 11.20, but care must be taken in the synthesis of this in code—you must form the internal nodes first to avoid a zero-delay loop (the $x(n) + gy(n)$ term above). Inverted APFs simply swap signs on the g coefficients. This has the effect of inverting the impulse response while keeping the frequency response theoretically flat. By combining both inverting and regular delaying APFs, you can try to achieve a more uncorrelated echo density.

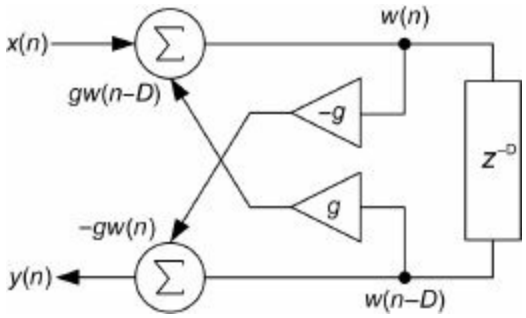


Figure 11.13: Alternate version of the delaying APF; the transfer function and difference equations are identical to the other structures.

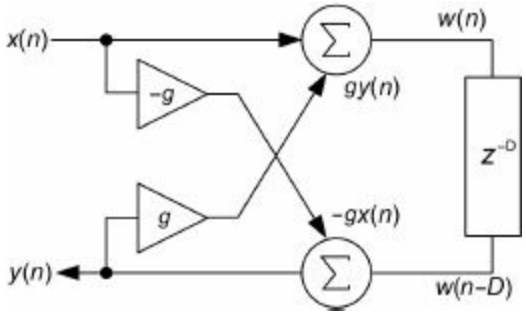


Figure 11.14: Another delaying APF structure.

11.6 Schroeder's Reverberator

Schroeder combined a parallel comb filter bank with two APFs to create his first design. The comb filters produce the long series of echoes and the APFs multiply the echoes, overlaying their own decaying impulse response on top of each comb echo. The resulting reverberation unit, shown in Figure 11.15, sounds marginal but it is very simple to implement.

Schroeder suggests that the ratio of smallest to largest delay in the comb filters should be about 1:1.5 and originally specified a range of 30–45 mSec in total. The 1:1.5 ratio rule turns out to be useful for just about any set of parallel comb filters.

The comb filters should have the following properties:

- Choose the delays to have the 1:1.5 ratio above.
- Choose delay times that have no common factors (e.g., 2191, 2971, 3253, 3307).
- Set the gain values according to Equation 11.9.

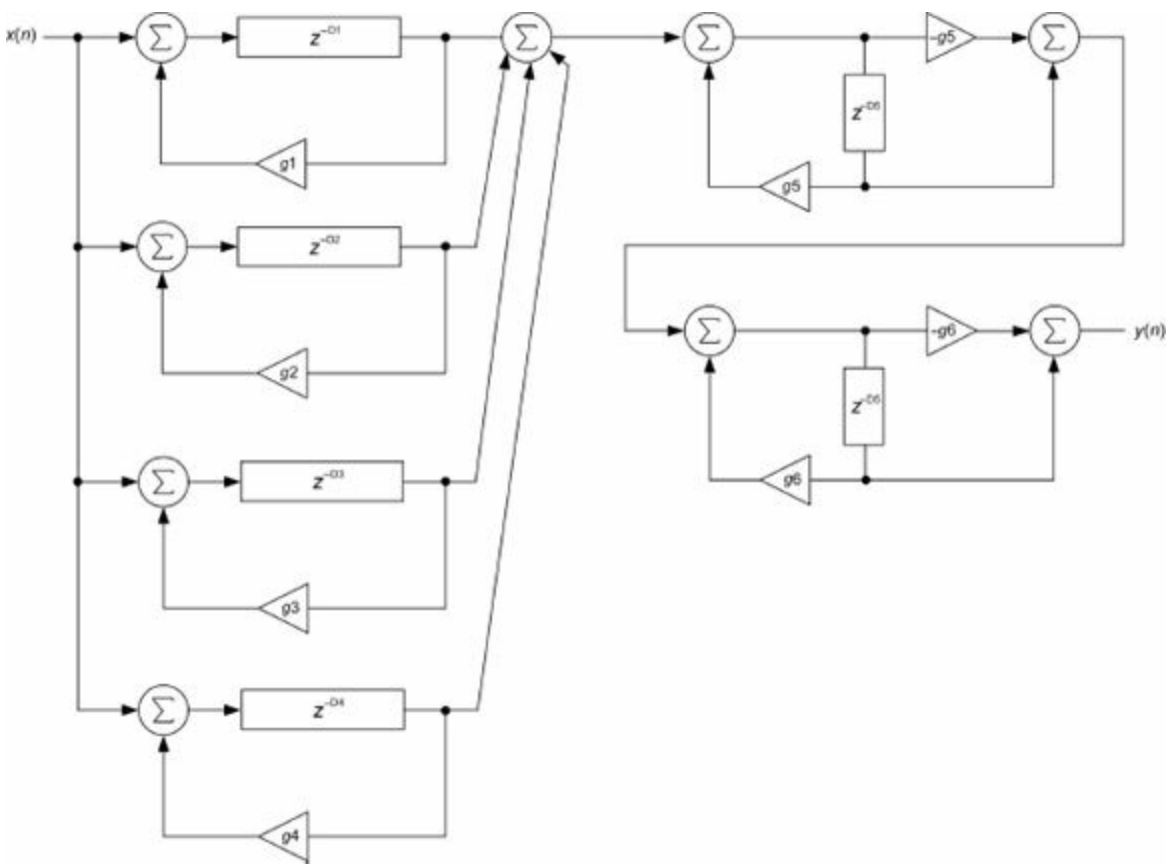


Figure 11.15: Schroeder's original reverb design.

The APFs should have the following properties:

- Choose delays that are much shorter than the comb filters, 1 mSec to = mSec.
- Set both gain values the same, between 0.5 and 0.707.

11.7 The Low-Pass Filter–Comb Reverberator

One of the things missing from the original Schroeder reverb is the fact that in a real room, high frequencies decay much more rapidly than low frequencies, as shown in the EDRs of Figure 11.6 (Moorer 1979). Placing a low-pass filter (LPF) in the comb filter's feedback path will roll off the high-frequency content of successive echoes exponentially, which is what we want. The LPF–comb reverberator block diagram is shown in Figure 11.16.

The LPF is chosen to be a one-pole feedback type (infinite impulse response [IIR]) so that it will introduce not only low-pass filtering, but also its own impulse response into the echoes going through the feedback path. In Figure 11.17 the LPF is shown in the dotted box; notice that it is turned around backward to follow the flow of the feedback path.

To find the difference equation, it is easier to start in the frequency domain with the z transforms of the comb and first-order feed-forward filter, since we are already familiar with them by now (Equation 11.22).

$$H_C(z) = \frac{z^{-D}}{1 - g_1 z^{-D}}$$
$$H_{LP}(z) = \frac{1}{1 - g_2 z^{-1}} \tag{11.22}$$

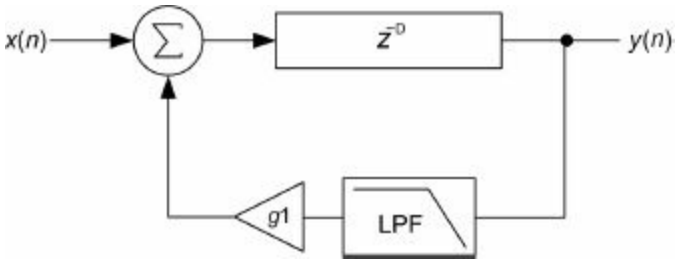


Figure 11.16: The LPF and comb filter combination.

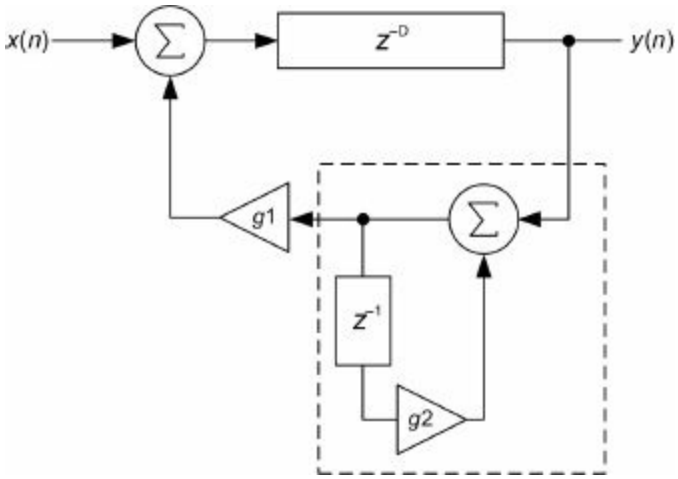


Figure 11.17: The LPF–comb filter expanded.

Filtering the feedback loop in the frequency domain is done by simply multiplying the LPF transfer function by the feedback term $g_1 z^{-D}$ in Equation 11.23.

$$\begin{aligned} H(z) &= \frac{z^{-D}}{1 - H_{LP}(z) g_1 z^{-D}} \\ &= \frac{z^{-D}}{1 - g_1 z^{-D} \dfrac{1}{1 - g_2 z^{-1}}} \\ &= \frac{z^{-D}}{1 - \dfrac{g_1 z^{-D}}{1 - g_2 z^{-1}}} \\ \frac{Y(z)}{X(z)} &= \frac{z^{-D}}{1 - \dfrac{g_1 z^{-D}}{1 - g_2 z^{-1}}} \end{aligned} \tag{11.23}$$

The next step is to separate variables and multiply out the denominator:

$$
\begin{aligned}
Y(z)\left[1-\frac{g_1 z^{-D}}{1-g_2 z^{-1}}\right] &= X(z)z^{-D} \\
Y(z)[1-g_2 z^{-1}-g_1 Y(z)z^{-D}] &= X(z)z^{-D}(1-g_2 z^{-1}) \\
Y(z)-g_2 Y(z)z^{-1}-g_1 Y(z)z^{-D} &= X(z)z^{-D}-g_2 X(z)z^{-D}z^{-1} \\
Y(z)-g_2 Y(z)z^{-1}-g_1 Y(z)z^{-D} &= X(z)z^{-D}-g_2 X(z)z^{-D-1}
\end{aligned}
\tag{11.24}
$$

Lastly, take the inverse z transform by inspection:

$$
Y(z)-g_2 Y(z)z^{-1}-g_1 Y(z)z^{-D} = X(z)z^{-D}-g_2 X(z)z^{-D-1}
$$

then

$$
y(n)-g_2 y(n-1)-g_1 y(n-D) = x(n-D)-g_2 x(n-D-1) \tag{11.25}
$$

and

$$
y(n) = x(n-D)-g_2 x(n-D-1)+g_2 y(n-1)+g_1 y(n-D)
$$

In order for the filter combination to remain stable (after all, it is a massive feedback system) g_1 and g_2 can be related as in Equation 11.26:

$$
g = \frac{g_2}{1-g_1}
$$

where

$$
g < 1.0 \tag{11.26}
$$

Because the RT_{60} determines the pole radii, and therefore the value of g_1, you can rearrange Equation 11.26 as Equation 11.27:

$$
g_2 = g(1-g_1)
$$

where

$$
g < 1.0 \tag{11.27}
$$

11.8 Moorer's Reverberator

Moorer proposed a similar design to Schroeder's reverberator which uses a parallel bank of LPF–comb filters. Because the LPFs remove energy from the system, more units are needed in parallel for a given reverb time. Moorer's reverb sounds better because it mimics the high-frequency losses of a real room.

In Figure 11.18 you can see the differences from Schroeder's reverb—there are more comb units and only one all-pass on the output. The same care must be taken to ensure the pole radii of the comb filters are still the same, using Equation 11.9 to set them according to the desired reverb time. Table 11.1 shows Moorer's preset values for a reverb time of about 2 seconds.

Table 11.1: Some settings for Moorer's reverb.

Comb Filter	Delay (mSec)	g_{COMB} (48 kHz)	g_{LPF} (48 kHz)
1	50	0.46	0.4482
2	56	0.47	0.4399

3	61	0.475	0.4350
4	68	0.48	0.4316
5	72	0.49	0.4233
6	78	0.50	0.3735
All-Pass Filter	**Delay (mSec)**	*g*	–
1	6	0.7	–

Note: RT_{60} 2 seconds, total $g = 0.83$

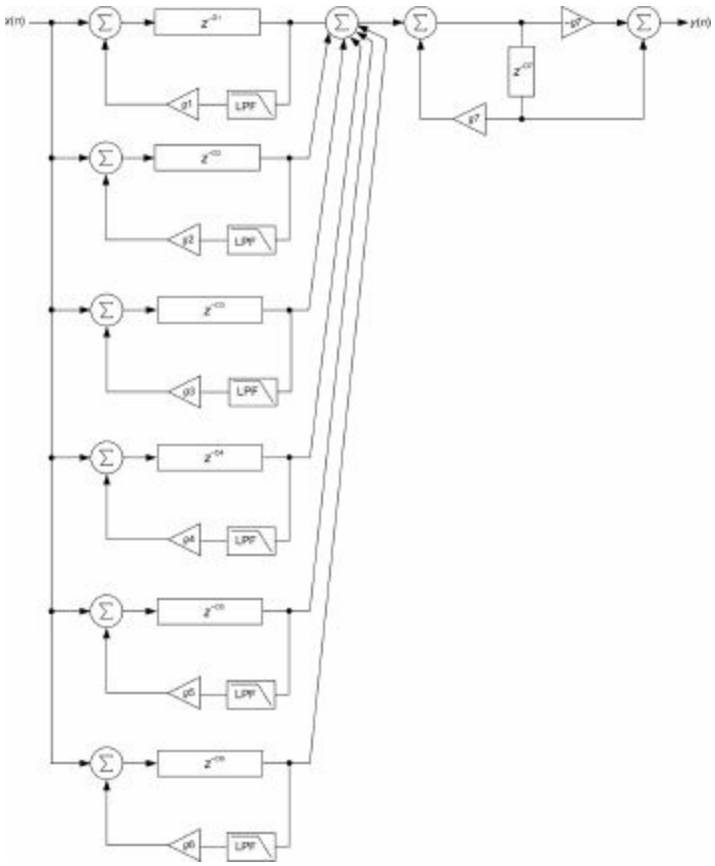


Figure 11.18: Moorer's reverb.

11.9 Stereo Reverberation

Conducting listening tests, Schroeder (1984) found that listeners overwhelmingly preferred a stereo reverb to a mono version of the same algorithm. Both Schroeder and Moorer's reverbs can be adapted for stereo usage. In fact, this scheme can be used with just about any reverb consisting of comb and APFs. The first thing to note is that mathematically, there is no reason why you can't place the APFs before the comb filters since their transfer functions multiply in the *z* domain. Then, the individual outputs of the comb filters can be combined through a mixing matrix to provide the left and right outputs. The mixing matrix is an array of weighting values for each comb filter's output. Jot proposed that the matrix should be orthogonal for the most uncorrelated output. The mixing matrix is shown in Equation 11.28. The rows are the outputs of the comb filters and the columns are left and right. Figure 11.19 shows a mixing matrix for the left channel of a Schroeder reverb unit.

$$g\begin{bmatrix} +1 & +1 \\ +1 & -1 \\ +1 & +1 \\ +1 & -1 \end{bmatrix} \tag{11.28}$$

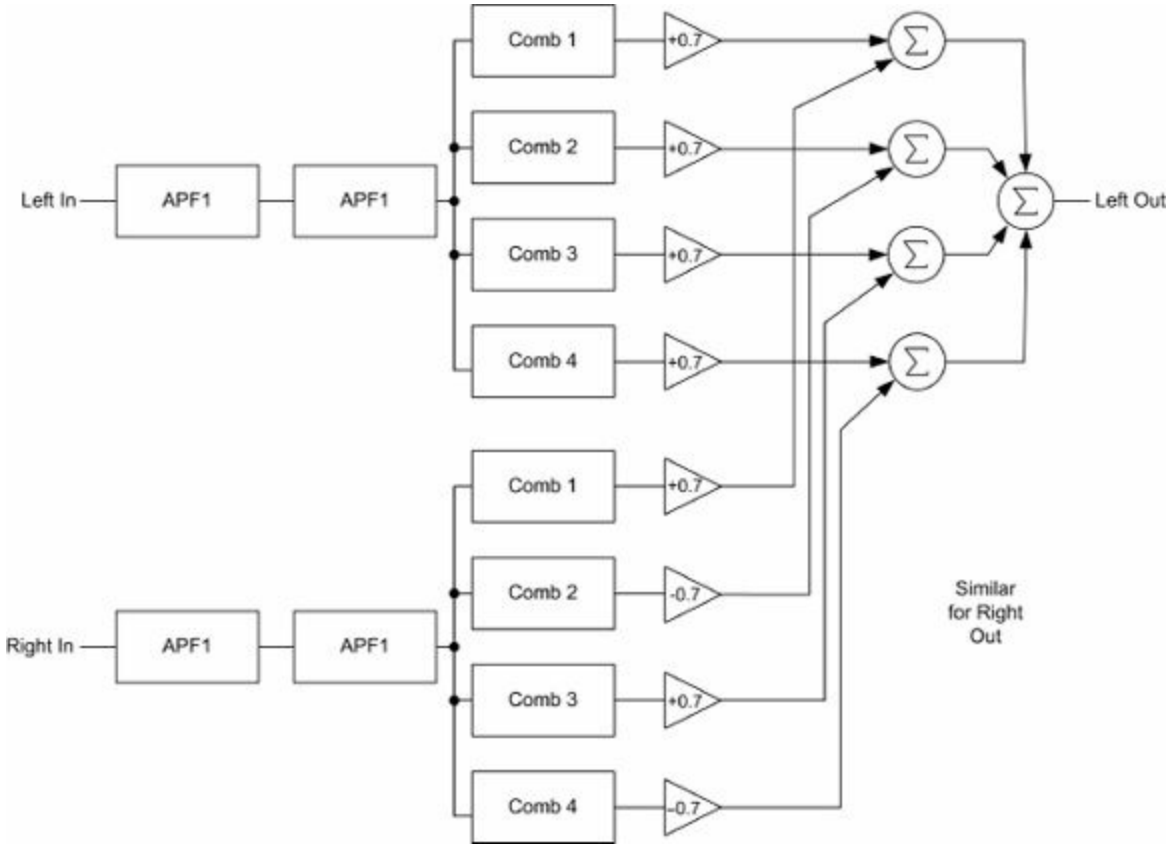


Figure 11.19: A stereo implementation for a Schroeder reverb—only one channel is shown for clarity; the right output is fashioned in a similar way. Notice the mix matrix values do not have to be ±1.0 but need to follow the orthogonality of alternating signs.

11.10 Gardner's Nested APF Reverberators

Schroeder experimented with reverbs made only from delaying APF modules in series. The abundant time smearing suggested that this might be a viable option. Unfortunately the consensus was that this reverb sounded metallic and that it didn't build up the echo density fast enough. Schroeder's original series APFs started with the maximum delay in the first module (about 100 mSec) and then decreased by 0.3 for each successive module, through five modules. Gardner (1992) noted that Schroeder (1962), Gerzon (1972), and Moorer (1979) all experimented with *nesting* multiple all-pass structures, but without success. He suggested a strategy for nesting delaying APF modules along a single transversal delay line. In order to understand how this works, re-fashion the delaying APF structure using the transversal delay line structure you saw in Chapter 8. Figure 11.20 shows the same delaying APF structure as shown in Figure 11.13 but with the delay replaced by a set of single delay elements.

The sequence of operation—reading before writing—is key for implementing this design. Specifically, you need to

1. Read the output of the last delay cell, $x(n - D)$.

2. Read the value of the first delay cell, $x(n)$.
3. Form the new value for the last cell $-gx(n) + x(n - D)$ and write it back into the cell as $y(n)$.
4. Form the new value for the first cell $x(n) = x(n) + gy(n)$.

Gardner's idea was to nest multiple APF structures inside each other so that they *shared* the same delay line. This would produce layers of embedded echoes. Figure 11.20 shows a delaying APF with a total delay time of z^{-5}. Nesting another APF with a delay of z^{-3} inside it results in Figure 11.21.

Gardner also devised a new schematic representation of his nested filter structures that removes the clutter of the delay cells, gain, and summation components. Figure 11.22 shows a nested APF structure. The outer APF has a delay time $D2$ and gain g_2 while the inner APF uses $D1$ and g_1 as its values. Additionally, pre-and post-delays may be added to the transversal delay line before and after the nested structure as shown in Figure 11.23.

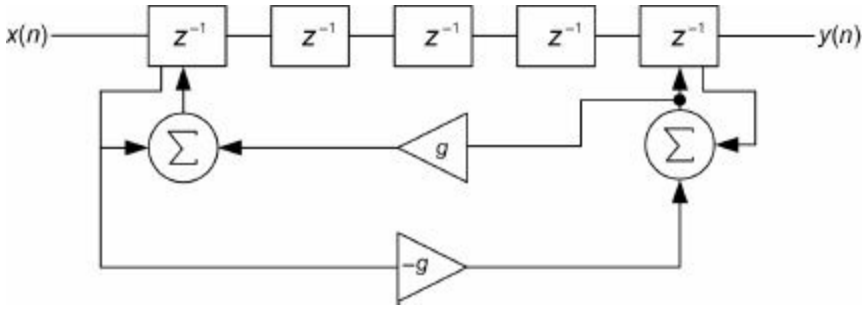


Figure 11.20: The delaying APF structure sitting across a transversal delay line.

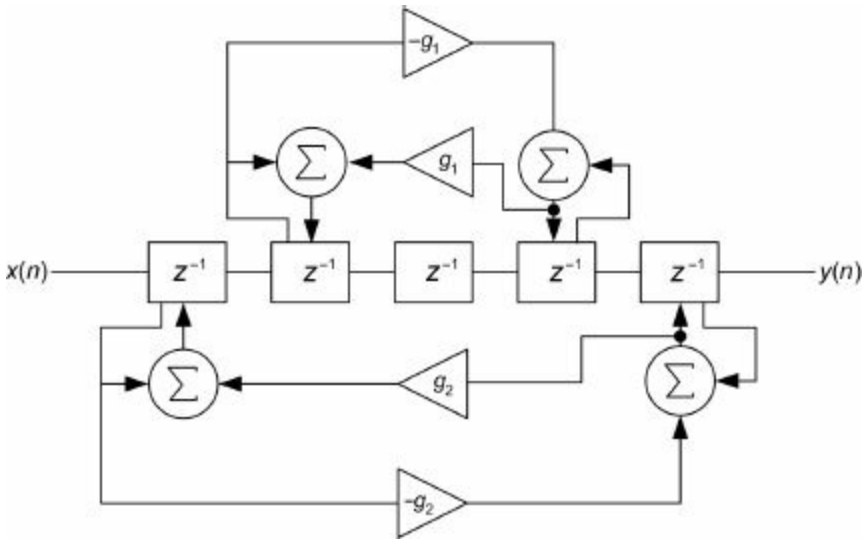


Figure 11.21: Two delaying APF structures sharing the same delay line.

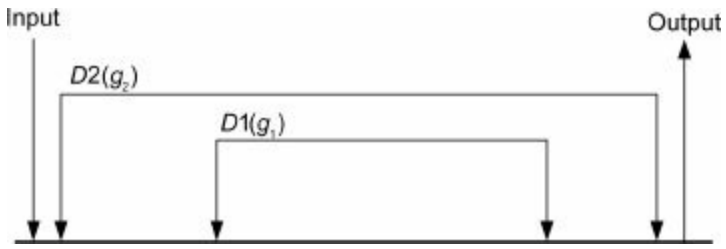


Figure 11.22: A nested APF schematic; Gardner gave the delay times in mSec.

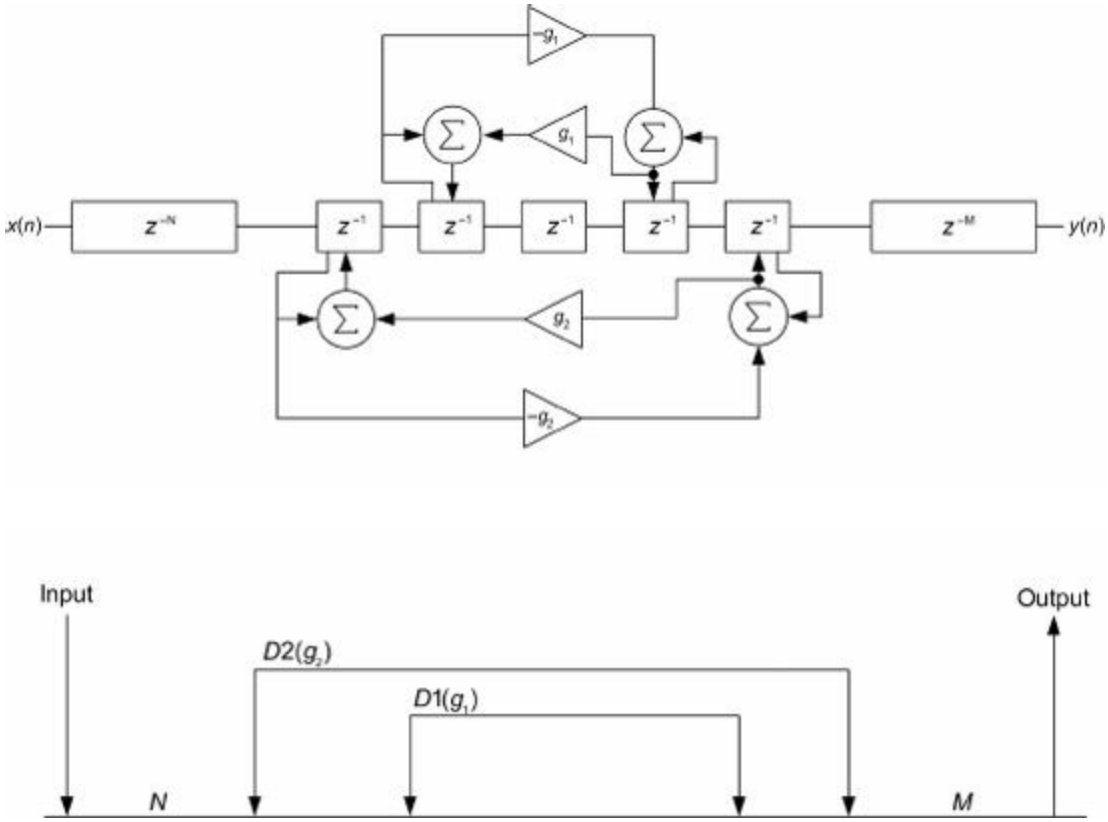


Figure 11.23: Pre- and post-delays of length *N* and *M* have been added to the nested APF structure. The second diagram shows how Gardner would lay this out schematically.

Consider the nested APF structure in Figure 11.24. It has an outer APF with a delay time of 35 mSec with $g = 0.3$ and two inner APFs with delay and gain values of 22 mSec ($g = 0.4$) and 8.3 mSec ($g = 0.6$). This is actually the first part of one of Gardner's larger designs. Figure 11.25 shows the impulse responses as each APF is added to the structure.

You can certainly see how the echo density begins to grow as you add each APF unit. We note several things about the nesting system:

- In the example, the first nested APF is 22 mSec in length; because of the commutative property of the delay operator, it doesn't matter where the 22 mSec delay is placed within the 35 mSec outer element.
- The 8.3 mSec APF is not nested inside the 22 mSec APF; it comes anywhere after it but still inside the outer APF.

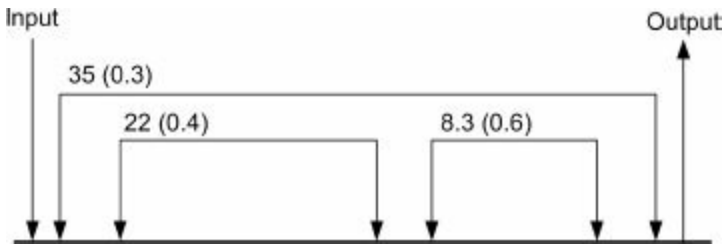


Figure 11.24: Three APFs—an outer filter and two nested units.

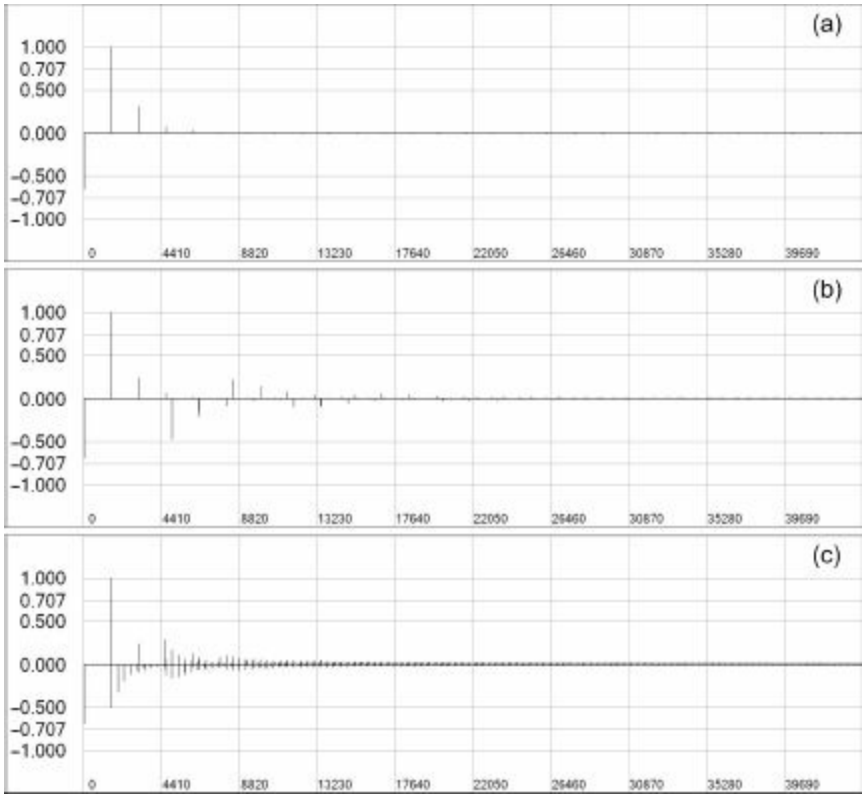


Figure 11.25: (a) The output of the single outer APF with delay of 35 mSec and gain of 0.3 shows the typical APF impulse response. (b) The first nested APF with delay of 22 mSec has been added. (c) The second nested APF with delay of 8.3 mSec has been added to the first two.

- Echo density piles up as time increases.
- The system can produce ringing or instabilities.
- The system still sounds metallic.

Gardner's solution to the ringing or metallic sound was to create a feedback path around the entire system with an LPF in the loop, thus embedding the nested APFs in a massive comb filter. He also notes that his reverb designs were ultimately arrived at by trial-and error. Figure 11.26 shows Gardner's three reverb designs for small, medium, and large room simulations. In each algorithm, the reverb time is controlled by the loop gain or “g” control (notice there are two of them in the medium room algorithm).

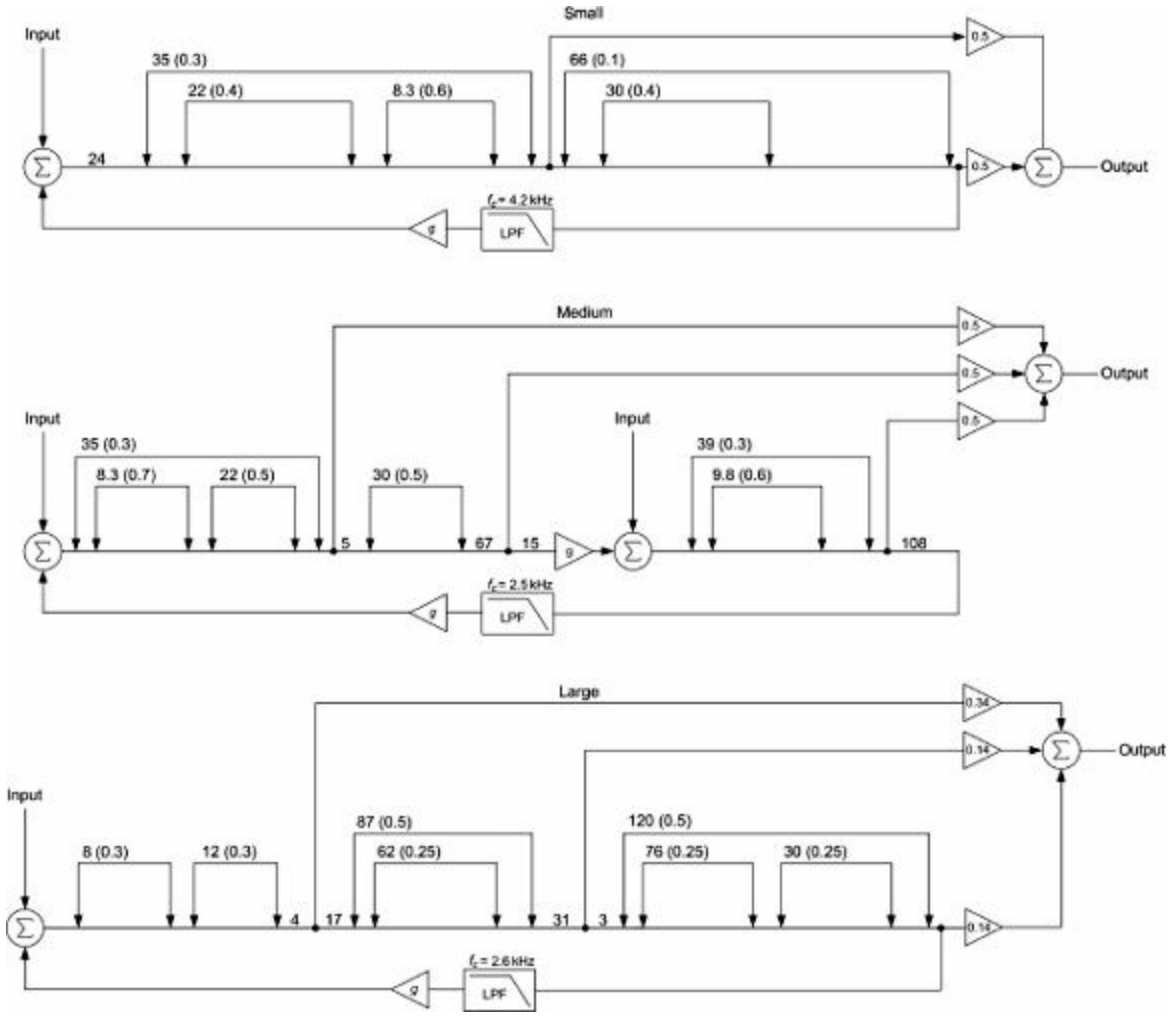


Figure 11.26: From top to bottom these are Gardner's nested APF reverb algorithms for small, medium, and large room simulations. Notice that the medium room algorithm in the middle has two input locations. Also notice the use of pre-and post-delays on some of the nested modules.

11.11 Modulated APF and Comb/APF Reverb

The modulated delay line can be used to further increase time smearing and echo density by using it in an APF module. The low-frequency oscillator (LFO) rate is kept low (<1 Hz) and the depth also very low (10–20 samples) to ensure that the detuning and chorusing effect is not overly obvious. In 1997, Dattorro presented a plate reverb algorithm "in the style of Greisinger," revealing the use of a modulated APF. He also notes that the modulated filter technically mimics a room whose walls are slowly moving back and forth.

Frenette (2000) showed that the use of modulated delay lines could reduce the computational requirements without a perceptual reduction in quality of reverb. Frenette used modulated comb filters (chorus modules) in addition to modulated APFs to further reduce the overall complexity and memory requirements. The modulated APF is shown in Figure 11.27.

The modulated APF in Figure 11.27 modulates only the very end of the delay line, producing a delay described by Equation 11.29:

$$Delay = z^{(-D + u(nT))} \tag{11.29}$$

where $u(nT)$ is the discrete time modulation signal (LFO). The inverting version can be implemented with a swap of the g coefficients, as with the other APFs.

11.12 Dattorro's Plate Reverb

Dattorro's plate reverb algorithm in Figure 11.28 has a block diagram that reveals its figure-8 recirculating tank circuit. This figure-8 circuit could be applied to the other reverb block diagrams to generate countless variations on this theme.

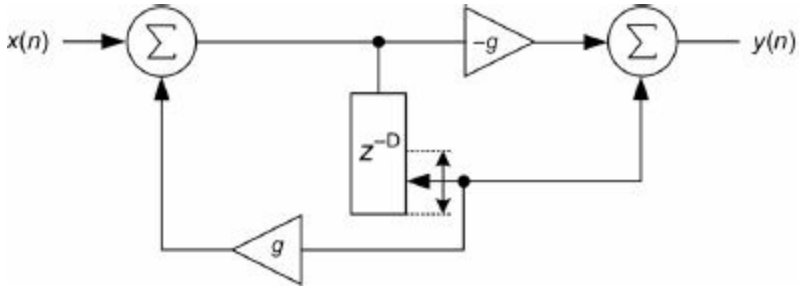


Figure 11.27: A modulated APF.

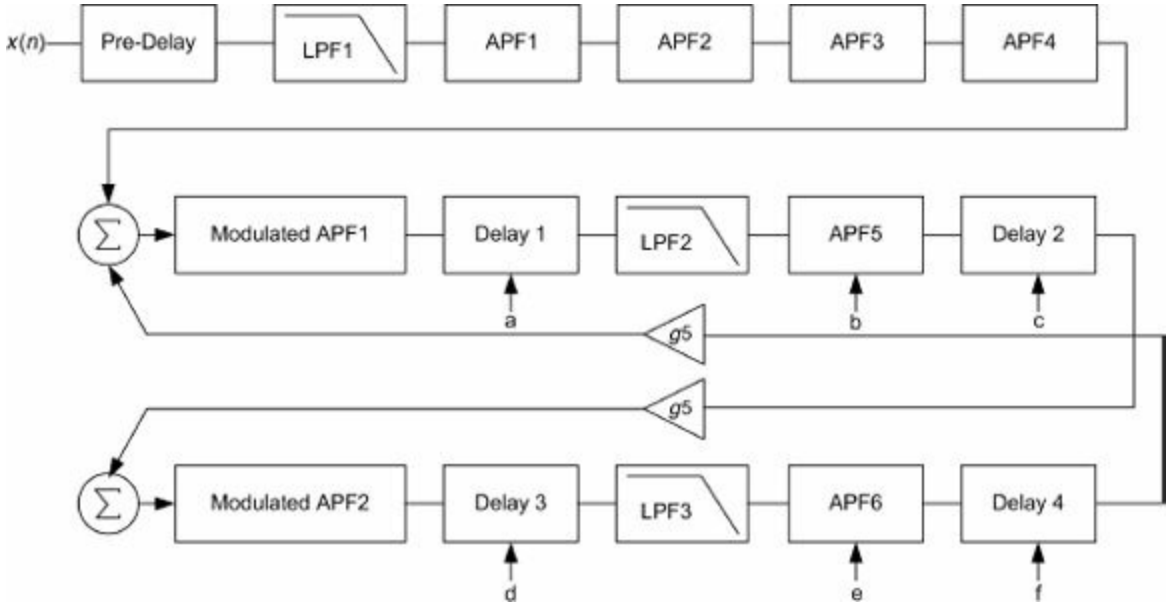


Figure 11.28: Block diagram of Dattorro's reverb.

You might notice something strange about Figure 11.29—it has no output $y(n)$ node. In fact, the left and right outputs are taken from various locations within the delay lines, marked a–f in the diagram. This is a mono-in, stereo-out reverberator. The first LPF is marked "diffusion" while the second pair (LPF2 and LPF3) are designated "damping." The first filter controls the diffusion in the series APFs while the second pair controls the high-frequency roll-off in the tank circuit. These LPFs are all DC-normalized single pole feedback filters shown in Figure 11.30.

Table 11.2 gives the various values for the filters, followed by the equations that give the left and right outputs. The original design was for a sample rate of 29.8 kHz. Values for f_s = 44.1 kHz are calculated and given in the tables. Figure 11.30 shows the entire reverb algorithm block diagram. Table 11.3 lists the control ranges and defaults.

Table 11.2: Gain and delay values for Dattorro's plate reverb.

APF Dx	**Delay (samples) f_s = 29.8 k**	**Delay (samples) f_s = 44.1 k**	*g* (index)	*g* (value)
1	142	210	1	0.75
2	107	158	1	0.75
3	379	561	2	0.625
4	277	410	2	0.625
9	2656	3931	3	0.5
10	1800	2664	3	0.5
Fixed Delay *Dx*	**Delay (samples) f_s = 29.8 k**	**Delay (samples) f_s = 44.1 k**	–	–
7	4217	6241	–	–
8	4453	6590	–	–
11	3136	4641	–	–
12	3720	5505	–	–
Mod APF *Dx*	**Delay (samples) f_s = 29.8 k**	**Delay (samples) f_s = 44.1 k**	g (index)	g (value)
5	908 +/–8	1343 +/–12	4	0.7
6	672 +/–8	995+/–12	4	0.7

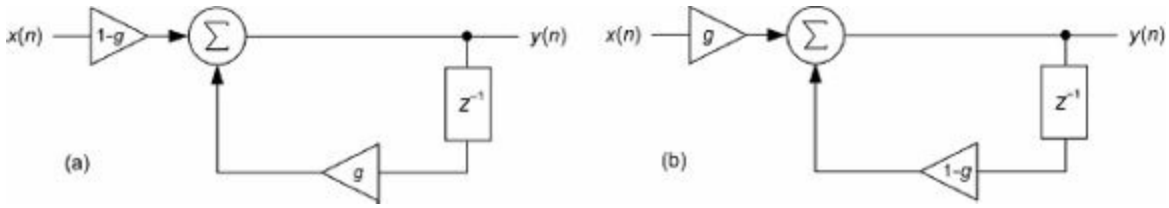


Figure 11.29: (a) DC-normalized one-pole filter, easy to use in reverb algorithms, including the comb filter and LPF type module. (b) This version merely reverses the effect of the slider or control.

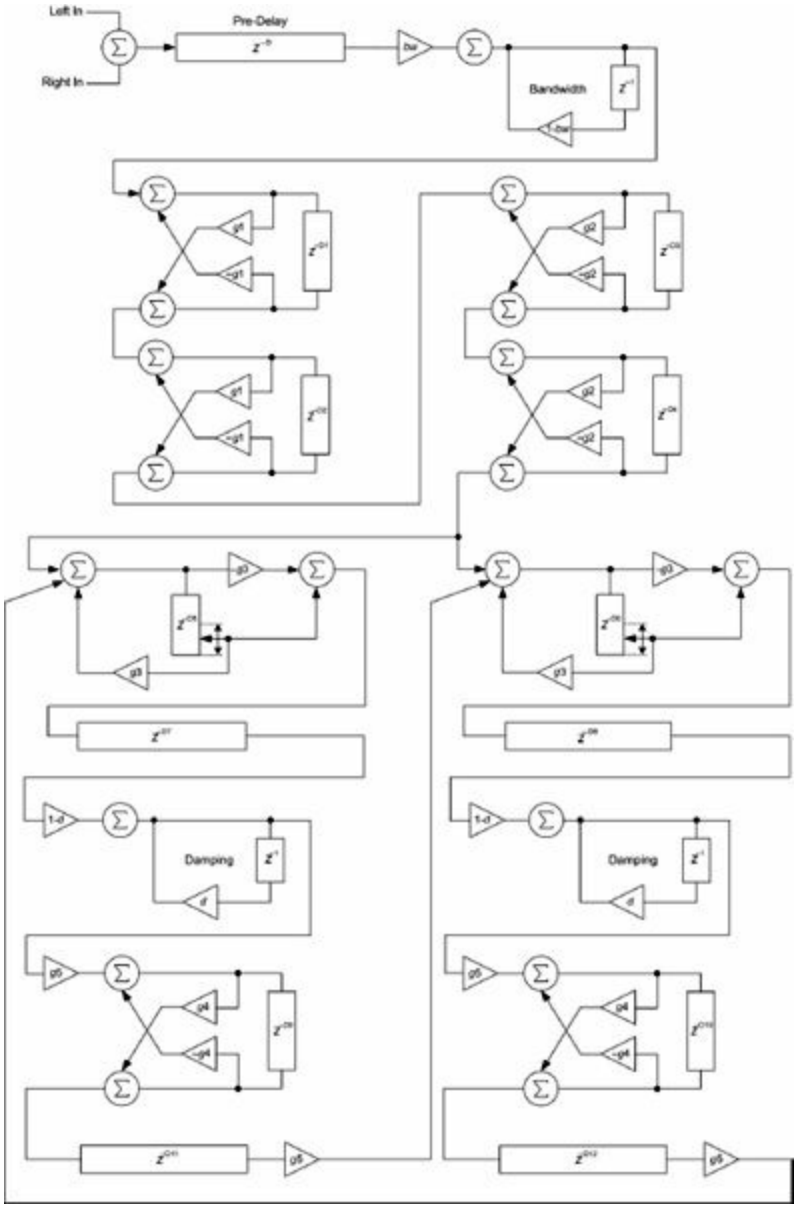


Figure 11.30: Dattorro's plate reverb algorithm.

Table 11.3: Control values for Dattorro's plate reverb.

Control	Range	Default
Decay (g^5)	0.0–1.0	0.5
Bandwidth	0.0–1.0	0.9995
Damping	0.0–1.0	0.0005

The left and right outputs (Equation 11.30) are summed from points within the delay lines labeled a–f in Figure 11.28.

f_s = 29.8 kHz:

$$y_L = a[266] + a[2974] - b[1913] + c[1996] - d[1990] - e[187] - f[1066] \quad (11.30)$$
$$y_R = d[353] + d[3627] - e[1228] + f[2673] - a[2111] - b[335] - c[121]$$

f_s = 44.1 kHz:

$$y_L = a[394] + a[4401] - b[2831] + c[2954] - d[2945] - e[277] - f[1578]$$
$$y_R = d[522] + d[5368] - e[1817] + f[3956] - a[3124] - b[496] - c[179]$$

11.13 Generalized Feedback Delay Network Reverbs

Another approach to reverberation is to realize that the listener is experiencing dense echoes caused by multiple reflections off of surfaces along with potentially constructive or destructive interference. The generalized *feedback delay network* (FDN) approach began with Gerzon's work in 1972 on preserving energy in multichannel reverberators. It was continued by Stautner and Puckette (1982) as well as Jot and Chaigne (1991). Generally speaking, the idea is to model the room with some number of delay lines with potential feedback paths to and from every delay line in the system. The inputs and outputs of each delay line are also scaled by some values b and c. Consider a simple version that consists of two delay lines with feedback paths to and from each one shown in Figure 11.31.

If you look at Figure 11.31 and think about Schroeder's parallel comb filter bank, you can see that this is a variation, indeed a generalization, on the structure. In the general FDN, every possible feedback path is accounted for. Additionally, each delay line has input and output amplitude controls. If you let $b1$, $b2$, $c1$, and $c2$ all equal 1.0 and set $g12$ and $g22$ to 0.0, you get Schroeder's parallel comb filter network exactly. Notice how the feedback coefficients are labeled:

- $g11$: Feedback from delay line 1 into delay line 1
- $g12$: Feedback from delay line 1 into delay line 2
- $g21$: Feedback from delay line 2 into delay line 1
- $g22$: Feedback from delay line 2 into delay line 2

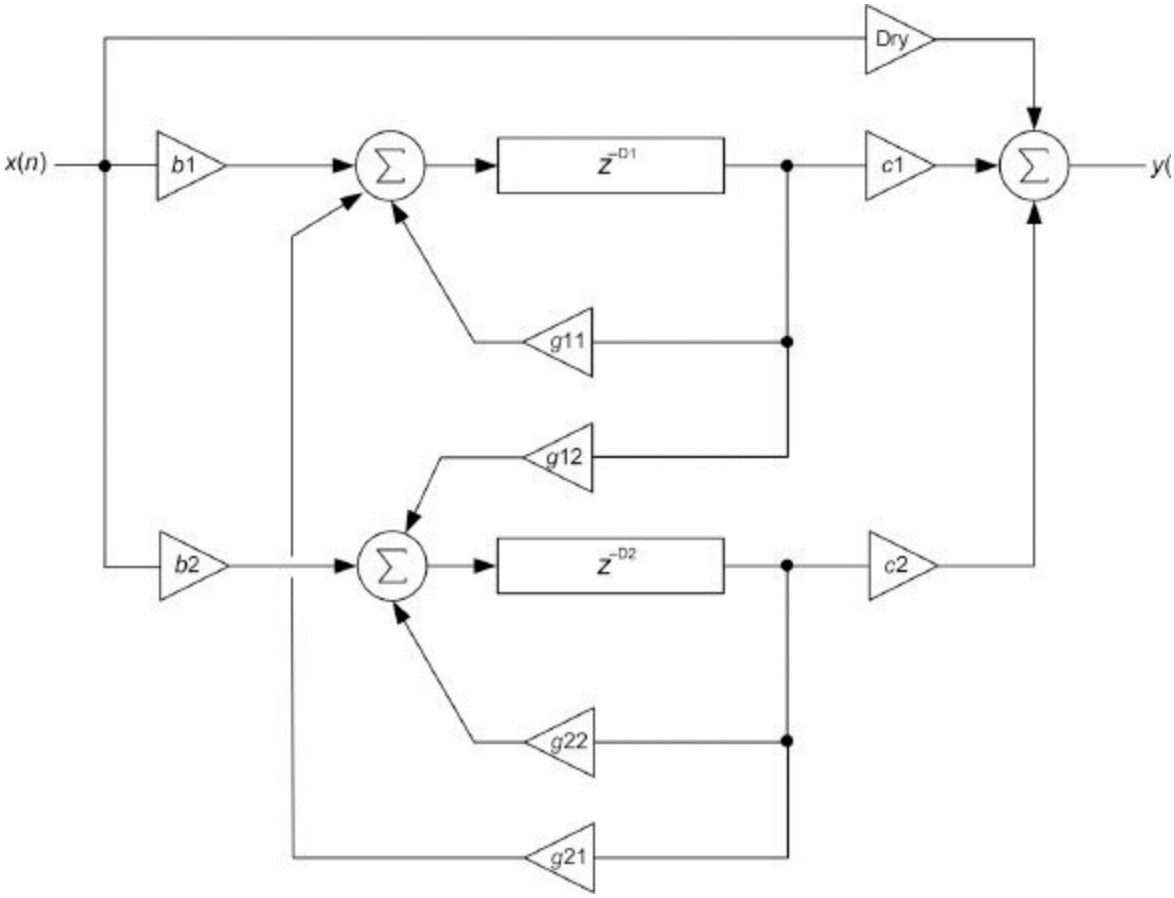


Figure 11.31: A two-delay-line feedback network.

An equivalent way to look at Figure 11.31 is shown in Figure 11.32. You can see that the feedback coefficients have been grouped together in what looks like a linear algebra matrix. The total energy is preserved if the matrix of coefficients is *unitary*. A unitary matrix multiplied by its *transpose matrix* results in the *identity matrix*. The transpose matrix is formed by turning the rows into columns and vice versa. These matrices are square. The identity matrix is a square matrix whose diagonal values are all 1. An FDN that uses a unitary matrix for its coefficients is called a *unitary feedback delay network* or UFDN.

Examine the gain matrix G in Equation 11.31:

$$G = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Then multiplying it by its transpose yields

$$GG^T = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{11.31}$$

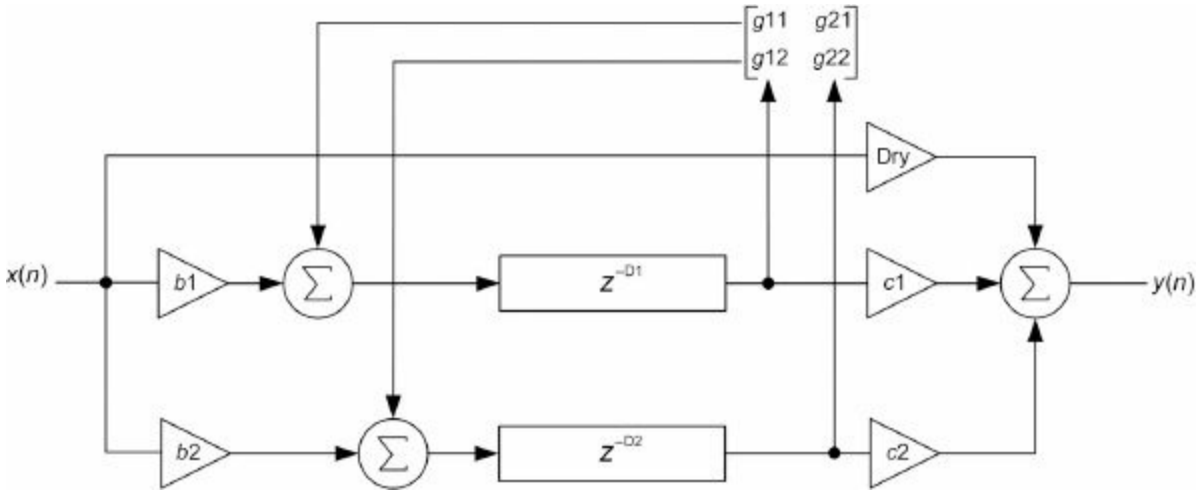


Figure 11.32: Another version of the two-delay-line FDN.

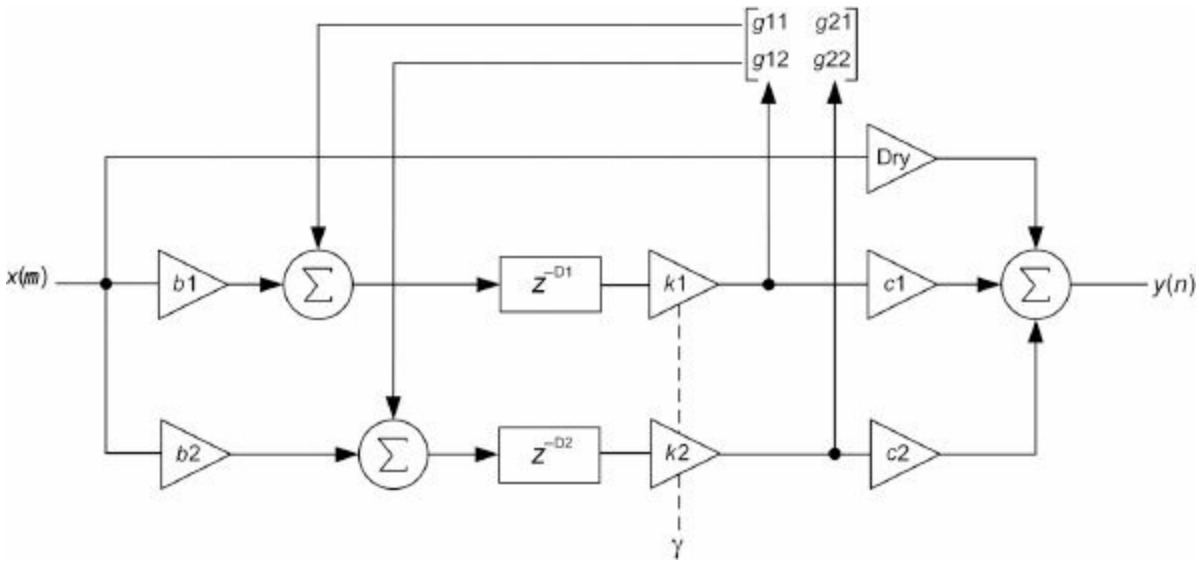


Figure 11.33: The FDN with decay factor control.

It is a unitary matrix. This reverberator would ring forever because of 1.0 gain values. Jot proposed adding an absorptive coefficient *k* to each delay line. For a colorless reverb, the value for *k* in Equation 11.32 is given by Jot.

$$k = \gamma^M \tag{11.32}$$

where

M = the length of the delay

γ = the decay factor, set by the user

In Figure 11.33, each delay line undergoes the proper attenuation to keep the reverb colorless. However, it does not include the frequency dependent absorptive losses we noted from the energy decay relief diagrams. To accomplish this, Jot then suggested inserting LPFs instead of the static attenuators. The LPF magnitudes are cleverly chosen in relation to the frequency-dependent reverb time, $RT_{60}(\omega)$ in Equation 11.33:

$$20\log(|h|) = \frac{-60T_s}{RT_{60}(\omega)}M \tag{11.33}$$

where

M = delay length
h = the magnitude of the filter at some frequency ω

The problem with this setup is that the pole radii are no longer the same (circular in the z plane) and we know that this produces coloration in the signal. The solution (Jot 1992) was to add a correction filter at the end of the whole system, $T(z)$, whose magnitude is proportional to the inverse square root of the frequency-dependent reverb time, $RT_{60}(\omega)$. This is shown in Figure 11.34.

Finally, Figure 11.35 shows a generalized version of the FDN. It does not include the absorptive loss components (either static k values or H_{LPFs}) for clarity. The feedback matrix is always square, $N \times N$, where N is the number of delay lines. Because it is square, it can be made to be unitary.

As an experiment, you can try to implement a four delay line version using the unitary matrix Equation 11.34 and gain coefficients proposed by Stautner and Puckette (1982), who were researching multichannel reverb algorithms.

$$G = \frac{g}{\sqrt{2}}\begin{bmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \tag{11.34}$$

where

$$|g| < 1.0$$

You can also try to de-correlate the four delay line outputs by using Jot's orthogonal matrix and set the coefficients c_N according to Equation 11.35:

$$C = g_2\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \tag{11.35}$$

where

$$|g_2| < 1.0$$

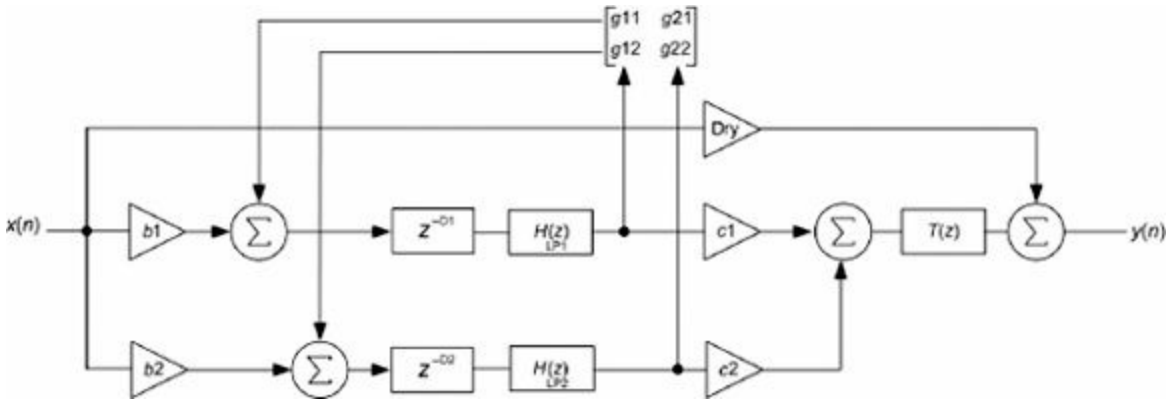


Figure 11.34: Addition of absorptive LPFs to the delay lines, plus a correction filter $T(z)$ at the end.

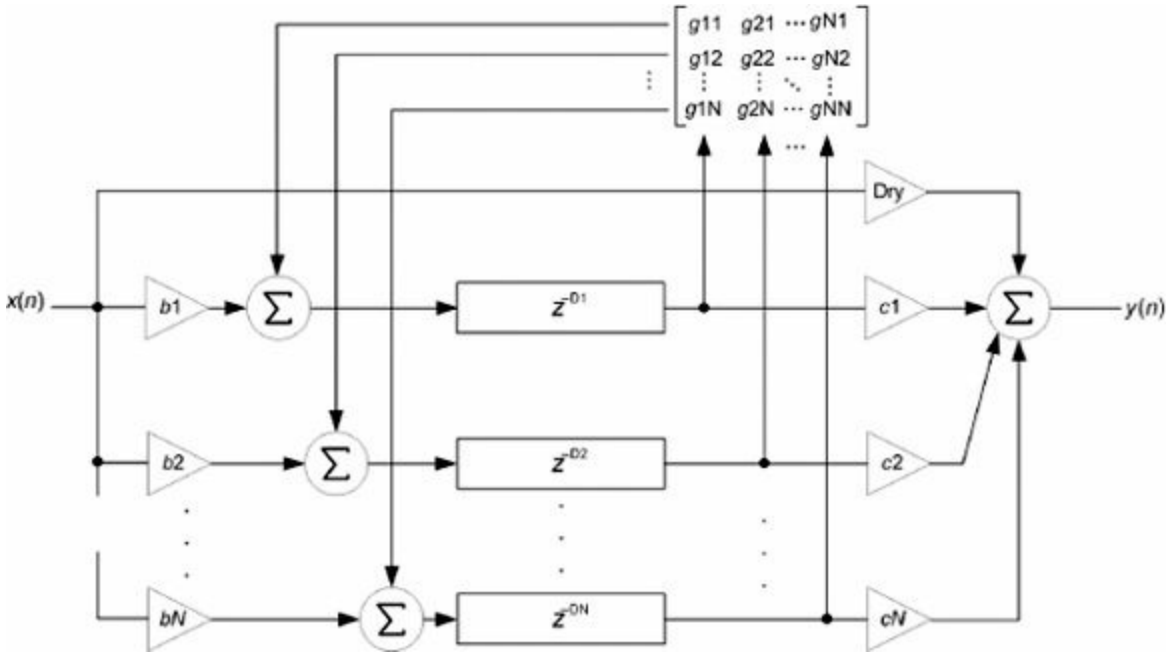


Figure 11.35: A generalized, *N*-delay-line FDN.

11.14 Other FDN Reverbs

Smith (1985) developed a variation on the FDN theme with his waveguide reverberators. Each waveguide consists of a two delay lines that move in opposite directions with coupling coefficients for connecting waveguides together. Any number of waveguides may be connected in any geometric pattern or shape, leading to a generalized set of scattering matrices. The scattering matrices are similar to the feedback matrices—they contain the coefficients that control the waveguide junction gains. The scattering matrices may also be made to be lossless by adhering to mathematical conditions involving the matrices' eigenvalues and eigenvectors (Smith and Rocchesso 1994).

Dahl and Jot (2000) proposed another UFDN type of reverb algorithm based on a structure they call the *absorbent all-pass filter* (AAPF). Figure 11.36 shows the block diagram of the AAPF which consists of a standard delaying APF with an LPF inserted in signal path. They combined an early reflections block that consisted of a multi-tapped delay line with a late reverberation block. In their late reverberation model, they use series AAPFs in a UFDN loop as shown in Figure 11.37. The block marked M is the unitary feedback matrix that mixes and creates the inputs for the filter loops.

Chemistruck, Marcolini, and Pirkle (2012) also experimented with FDNs for reverb algorithms. They used the genetic algorithm (GA) to generate coefficients for feedback delay and filtering blocks in their proposed system. A random unitary matrix generator was coded to seed the GA so that the starting point is a UFDN. Their overall algorithm block diagram consists of a four-delay line FDN followed by a diffusion block, as shown in Figure 11.38. The GA was only used to find the delay network coefficients while the diffusion block remained constant throughout.

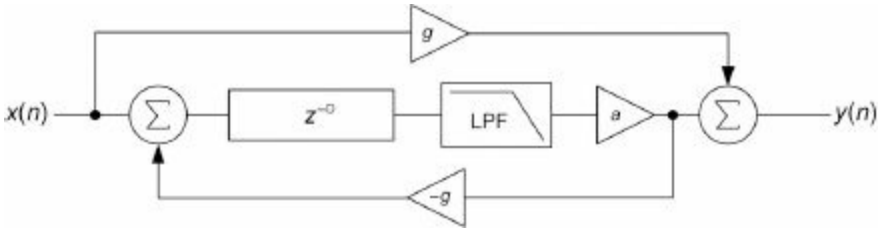


Figure 11.36: The absorbent APF features an LPF and attenuator in line with the delay element.

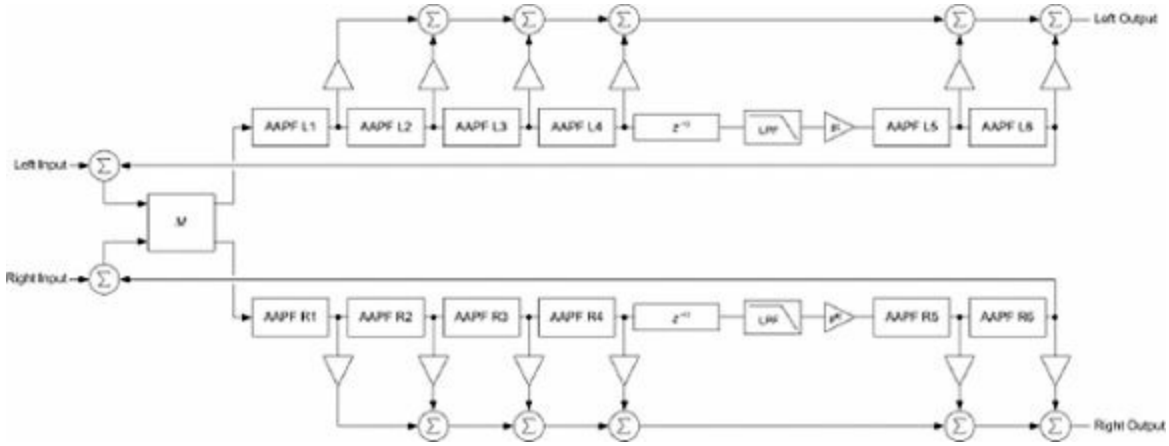


Figure 11.37: Dahl and Jot's late reverberator using a unitary feedback matrix *M* and absorbent APFs.

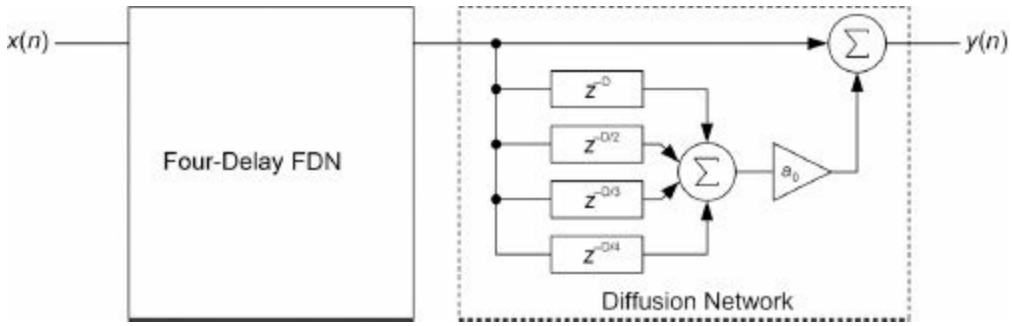


Figure 11.38: The FDN and diffusion network from the GA reverb.

Two different FDNs were used; the normal four-delay line and the delay and LPF in series (Figure 11.35, but without the correction filter on the output). An impulse response was taken for the target listening environment. The GA fitness function used the time domain envelope of the target impulse response for the matching criteria on the output of the system. During each iteration, the top 35% of offspring candidates were kept and inter-mated. Mutation rates of 10% for the delay coefficients and 5% for the LPF cutoff frequencies were used. More details can be found in the original reference.

We've analyzed many types of reverberator building blocks and complete algorithms. By now you should be able to see that reverb design is fairly open-ended. An example room reverb design follows.

11.15 An Example Room Reverb

For an example reverb, we have assembled a combination of reverberator modules to implement a mono-in, stereo-out reverb. The reverb is based on many of the modules and classic algorithms we've

studied so far. The idea is to produce a realistic-sounding algorithm that uses several of the more common reverberator modules so that you can use it as a springboard for your own designs. It is a good place to start your experiments because it features a variety of reverberator modules to help you get a better understanding of how the parameters change the overall reverb sound. Figure 11.39 shows the stereo impulse response for the factory preset; notice the pre-delay, early reflections, and late reverb sections are all present. Figure 11.40 shows the block diagram and graphical user interface (GUI) components for the example reverb. The complete algorithm consists of the following:

- Pre-delay with output attenuation.
- Input LPF ("bandwidth").
- Two input APF diffusion modules.
- Two parallel comb filter banks; four comb filters per bank; one bank for left output, one for right.

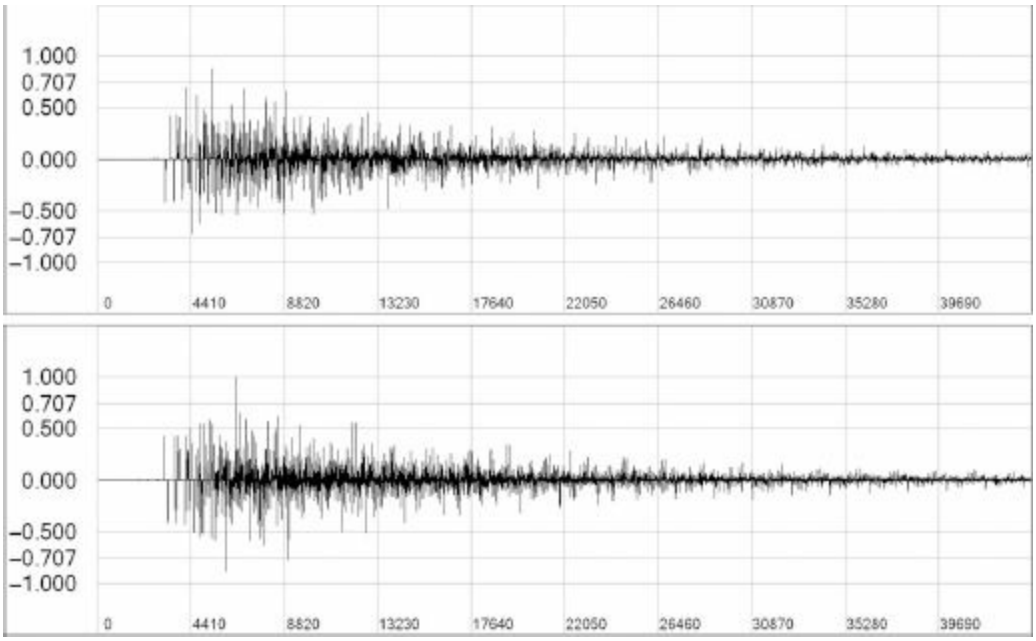


Figure 11.39: A stereo impulse response of the reverb plug-in, 100% wet output.

- Two comb filters in each bank are LPF–comb filters.
- Each parallel comb filter bank feeds an output LPF ("damping").
- The output LPF damping control also adjusts the LPF–comb filter *g* values.
- Each output LPF feeds an output APF diffusion module.
- A single RT_{60} Control for all comb filter gain variables.
- Individual APF and LPF controls for all parameters.

You can see many familiar components of this algorithm. The inputs are summed together to form a mono-signal and the left and right outputs are taken from two separate parallel comb filter echo generation units. The input bandwidth and output damping filters are the same as the Dattorro version but they also control the LPF–comb filter cutoffs. The parallel comb filter banks are reminiscent of the Schroeder and Moorer algorithms. These reverb algorithms often contain multiple instances of the same kinds of filters—APF, Comb, LPFComb, and so on. RackAFX provides you with several of these stock objects to use in your plug-ins. This will allow you to quickly assemble variations on the algorithms. Here are some key points for the design:

- We follow Schroeder's rule of about 1:1.5 ratio for shortest to longest comb filter delay (27.50 to 39.34 is our range, almost exactly 1:1.5).
- The RT_{60} time is used to calculate the gain variables for the comb filters for a less colored sound.
- The diffusion APFs alternate the sign of the *g* coefficient to create inverted/normal diffusion.

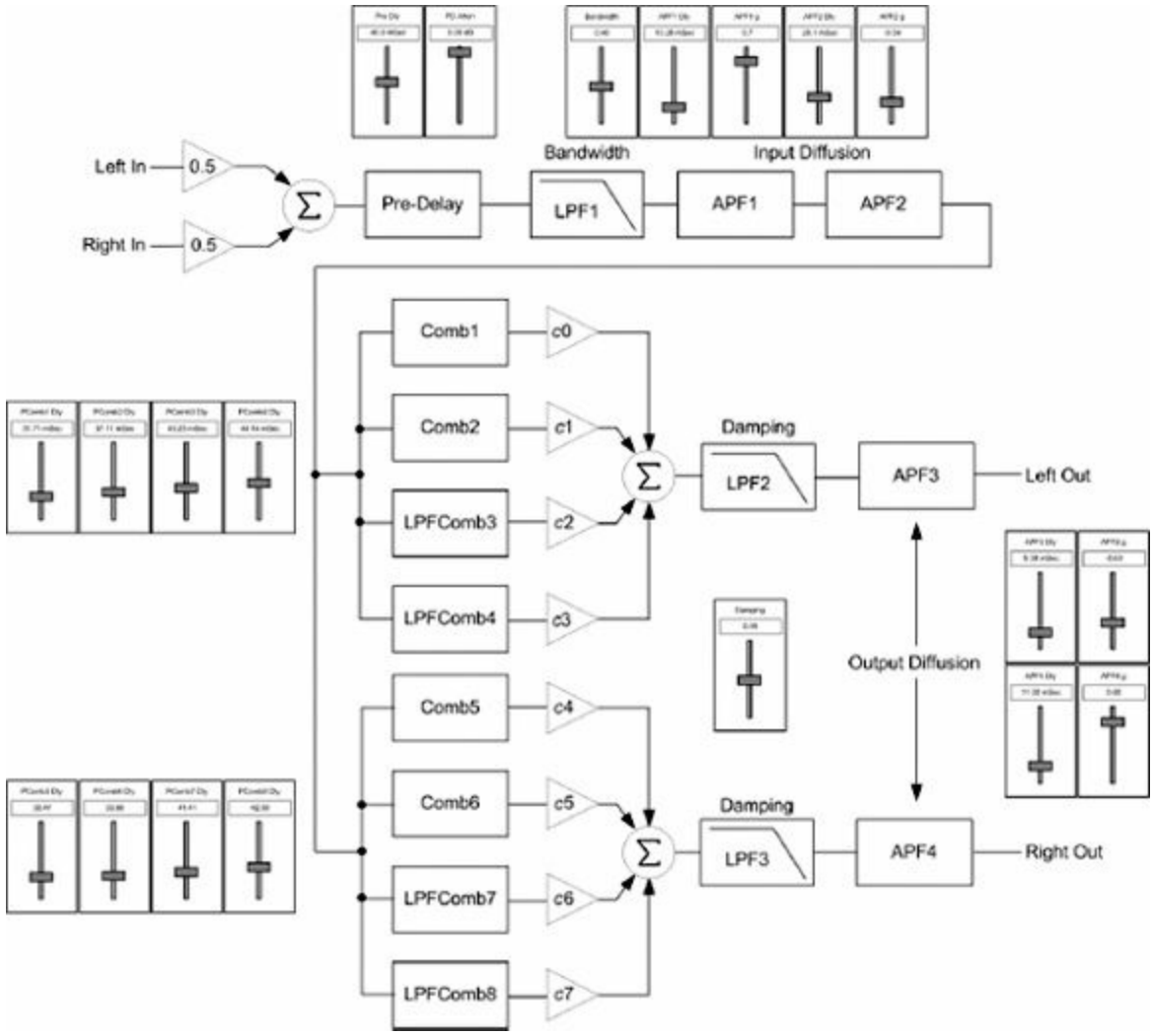


Figure 11.40: The room reverb algorithm.

- Adjusting the input APF delays has an effect on early reflections.
- Adjusting the output APF delays has a tonal coloration effect on the final reverb.
- The key to less ringing/metallic sound is getting the parallel comb filter delays just right, and not violating the 1:1.5 rule.
- The parallel comb filter attenuators, *c*0—*c*7 are hard-coded as 0.15, but there are several ways you could experiment with this, or allow the user to control it. They also have alternating signs, which was found to greatly reduce pinging.
- You can also provide unequal weighting such that the shorter (or longer) combs are more (or less) emphasized.

11.16 RackAFX Stock Objects

You can use any or all of the following stock objects in your plug-in—you just have to let RackAFX know when you create your project initially. The stock objects consist of the following:

- COnePoleLPF
- CDelay
- CCombFilter
- CLPFCombFilter
- CDelayAPF

When you tell RackAFX to add them to your project, RackAFX automatically copies the files, #includes them, and adds them to your C++ project file. Here's a quick reference guide to the objects. With these objects, coding our example reverb unit will be much easier than you think.

11.16.1 COnePoleLPF

Implements a one-pole LPF with a single coefficient, *g* in Figure 11.41. Table 11.4 shows object members.

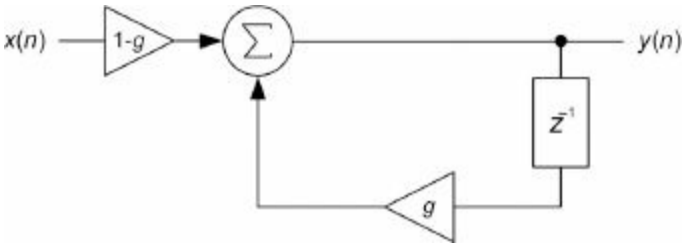


Figure 11.41: The one-pole LPF.

Table 11.4: The COnePoleLPF object.

Member Variables	**Purpose**
float m_fLPF_g float m_fLPF_z1	Implements the one and only gain coefficient *g*. Register to hold the single sample delay, z^{-1}.
Member Functions	
void setLPF_g(float fLPFg) Parameters: • float fLPFg	set() function for gain coefficient. Input: the new gain value.
void init()	Function to initialize and clear out the z^{-1} register.
bool processAudio(float* pInput, float* pOutput) Parameters: • float* pInput • float* pOutput	Process one input sample. Input: pointer to a float input sample. Output: pointer to the output sample destination.

11.16.2 CDelay

Implements a delay of *D* samples with an output attenuator. This is also the base class for other stock objects in Figure 11.42. Table 11.5 shows object members.

Table 11.5: The CDelay object.

Member Variables	**Purpose**
float m_fDelayInSamples	Implements the one and only gain coefficient *g*
float m_fOutputAttenuation	Output attenuation variable
float* m_pBuffer	Pointer to our dynamically declared buffer
int m_nReadIndex	Current read location
int m_nWriteIndex	Current write location
int m_nBufferSize	Max buffer size
int m_nSampleRate	Sample rate
float m_fDelay_ms	Delay in mSec, set by the parent plug-in
float m_fOutputAttenuation_dB	Output attenuation in dB, set by the parent plug-in
Member Functions	
void init(int nDelayLength) Parameters: • int nDelayLength	Declare the buffer and initialize it with 0.0s Input: the buffer length in samples
void cookVariables()	Function to cook delay (nSec) and attenuation (dB) to the variables actually used in the calculation
void resetDelay()	Reset the pointers to top, flush buffer
void setDelay_mSec(float fmSec) Parameters: • float fmSec	Called by parent to set the delay in mSec Input: the desired delay time
void setSampleRate(int nFs) Parameters: • int nFs	Set the sample rate; called by parent Input: the sample rate integer
void setOutputAttenuation_dB(float fAttendB) Parameters: • float fAttendB	Set the output attenuator; called by parent Input: the attenuation in dB
	Read the delay with the user defined delay time without

float readDelay()	incrementing any pointers or writing input data
float readDelayAt(float fmSec) Parameters: • float fmSec	Read the delay at an arbitrary delay time without incrementing any pointers or writing input data; caller must make sure that this does not exceed max delay time Input: the delay time to read
void writeDelayAndInc(float fDelayInput) Parameters: • float fDelayInput	Write the input value and increment the pointer indices Input: the input audio sample
bool processAudio(float* pInput, float* pOutput) Parameters: • float* pInput • float* pOutput	Processes one input sample Input: pointer to a float input sample Output: pointer to the output sample destination



Figure 11.42: The delay with output attenuator.

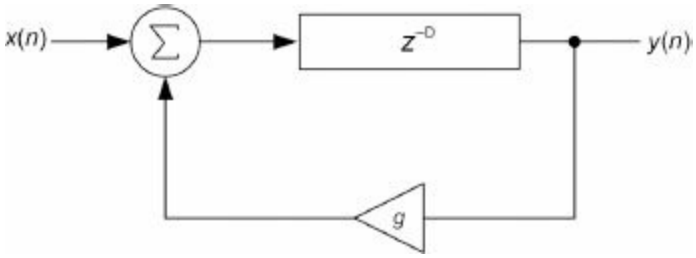


Figure 11.43: The comb filter.

11.16.3 CCombFilter

Implements a *D*-sample comb filter with feedback coefficient *g* in Figure 11.43. Table 11.6 shows object members.

Table 11.6: The CCombFilter object.

Member Variables	Purpose
float m_fComb_g	The one and only feedback gain coefficient g
Member Functions	
void setComb_g(float fCombg)	set() function for feedback gain coefficient

Parameters: • float fCombg	Input: the new feedback gain value
void setComb_g_with_RTSixty(float fRT) Parameters: • float fRT	Set the feedback gain using the RT60 time Input: RT60 time in mSec
CDelay Base Class Overrides	
bool processAudio(float* pInput, float* pOutput) Parameters: • float* pInput • float* pOutput	Process one input sample Input: pointer to a float input sample Output: pointer to the output sample destination

11.16.4 CLPFCombFilter

Implements a *D*-sample comb filter with LPF in feedback loop (with coefficient *g* 2) and feedback coefficient *g* 1 in Figure 11.44. Table 11.7 shows object members.

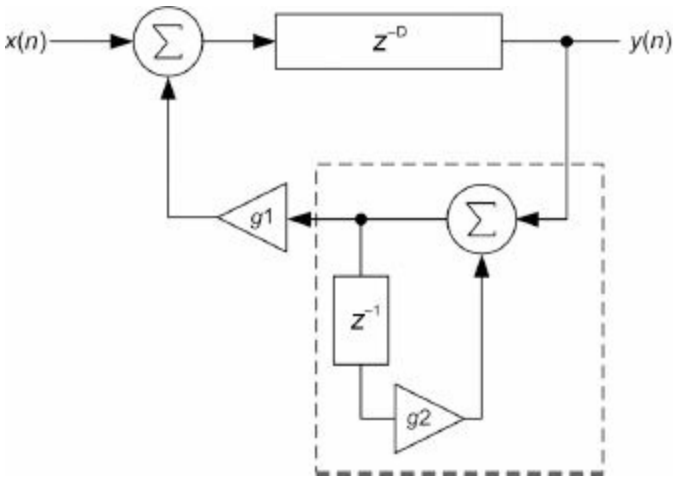


Figure 11.44: The LPF–comb filter.

Table 11.7: The CLPFCombFilter object.

Member Variables	Purpose
float m_fComb_g	The one and only feedback gain coefficient *g*1
float m_fLPF_g	The LPF gain coefficient, *g*2
float m_fLPF_z1	Register for one pole LPF
Member Functions	
void setComb_g(float fCombg) Parameters: • float fCombg	set() function for feedback gain coefficient Input: the new feedback gain value

void setComb_g_with_RTSixty(float fRT) Parameters: • float fRT	void setComb_g_with_RTSixty(float fRT) Parameters: float fRT
void setLPF_g(float fOverAllGain) Parameters: • float fOverAllGain	Set the LPF gain knowing an overall gain value; uses Equation 11.27 Input: the overall gain in Equation 11.27
CDelay Base Class Overrides	
void init(int nDelayLength) Parameters: • int nDelayLength	Dynamically create and init the buffer Input: delay length in samples
bool processAudio(float* pInput, float* pOutput) Parameters: • float* pInput • float* pOutput	Process one input sample Input: pointer to a float input sample Output: pointer to the output sample destination

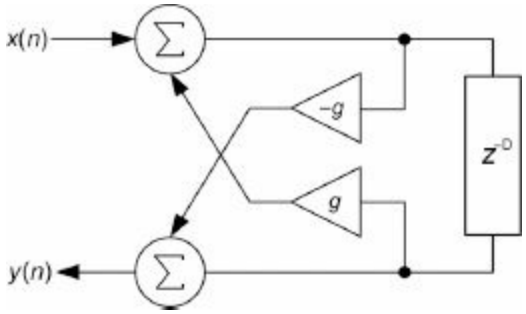


Figure 11.45: The delaying APF.

11.16.5 CDelayAPF

Implements a *D*-sample delaying APF in Figure 11.45. Table 11.8 shows object members.

Table 11.8: The CDelayAPF object.

Member Variables	Purpose
float m_fAPF_g	The one and only gain coefficient g
Member Functions	
void setAPF_g(float fAPFg) Parameters: • float fAPFg	set() function for gain coefficient Input: the new gain value

CDelay Base Class Overrides	
bool processAudio(float* pInput, float* pOutput) Parameters: • float* pInput • float* pOutput	Process one input sample Input: pointer to a float input sample Output: pointer to the output sample destination

11.17 Design the Room Reverb

11.17.1 Project: Reverb

Start a new project named “Reverb” and be sure to check the box to automatically include the stock reverb objects.

11.17.2 Reverb GUI

The first thing to do is decide on a user interface to control the reverb's modules. Since this is a learning tool as well, we will give the user many more controls than they would normally find on a reverb for the purpose of experimentation. Refer back to Figure 11.37, which has the block diagram and GUI components with it. There are going to be a lot of sliders to set up since this is a complex design (Table 11.9). However the use of the stock reverb objects is going to make the rest of the design easy.

Table 11.9: UI Control Properties for the Room Reverb: Pre-Delay

Slider Property	Value
Control Name	Pre Delay
Units	mSec
Variable Type	float
Variable Name	m_fPreDelay_mSec
Low Limit	0
High Limit	100
Initial Value	40

Slider Property	Value
Control Name	Pre Dly Atten
Units	dB
Variable Type	float
Variable Name	m_fPreDelayAtten_dB
Low Limit	−96
High Limit	0
Initial Value	0

Input Diffusion

Slider Property	Value
Control Name	Bandwidth
Units	
Variable Type	float
Variable Name	m_fInputLPF_g
Low Limit	0
High Limit	1
Initial Value	0.45

Slider Property	Value
Control Name	APF1 Dly
Units	mSec
Variable Type	float
Variable Name	m_fAPF_1_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	13.28

Slider Property	Value
Control Name	APF1 g
Units	
Variable Type	float
Variable Name	m_fAPF_1_g
Low Limit	−1
High Limit	1
Initial Value	0.7

Slider Property	Value
Control Name	APF2 Dly
Units	mSec
Variable Type	float
Variable Name	m_fAPF_2_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	28.13

Slider Property	Value
Control Name	APF2 g
Units	
Variable Type	float
Variable Name	m_fAPF_2_g
Low Limit	−1
High Limit	1
Initial Value	−0.7

Parallel Comb Filter Bank 1

Slider Property	Value
Control Name	PComb1 Dly
Units	mSec
Variable Type	float
Variable Name	m_fPComb_1_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	31.71

Slider Property	Value
Control Name	PComb2 Dly
Units	mSec
Variable Type	float
Variable Name	m_fPComb_2_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	37.11

Slider Property	Value
Control Name	PComb3 Dly
Units	mSec
Variable Type	float
Variable Name	m_fPComb_3_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	40.23

Slider Property	Value

Control Name	PComb4 Dly
Units	mSec
Variable Type	float
Variable Name	m_fPComb_4_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	44.14

Parallel Comb Filter Bank 2

Slider Property	Value
Control Name	PComb5 Dly
Units	mSec
Variable Type	float
Variable Name	m_fPComb_5_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	30.47

Slider Property	Value
Control Name	PComb6 Dly
Units	mSec
Variable Type	float
Variable Name	m_fPComb_6_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	33.98

Slider Property	Value
Control Name	PComb7 Dly
Units	mSec
Variable Type	float
Variable Name	m_fPComb_7_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	41.41

Slider Property	Value
Control Name	PComb8 Dly

Units	mSec
Variable Type	float
Variable Name	m_fPComb_8_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	42.58

Input Diffusion and Damping

Slider Property	Value
Control Name	Damping
Units	
Variable Type	float
Variable Name	m_fLPF2_g2
Low Limit	0
High Limit	1
Initial Value	0.5

Slider Property	Value
Control Name	APF3 Dly
Units	mSec
Variable Type	float
Variable Name	m_fAPF_3_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	9.38

Slider Property	Value
Control Name	APF3 g
Units	
Variable Type	float
Variable Name	m_fAPF_3_g
Low Limit	−1
High Limit	1
Initial Value	−0.6

Slider Property	Value
Control Name	APF4 Dly
Units	mSec

Variable Type	float
Variable Name	m_fAPF_4_Delay_mSec
Low Limit	0
High Limit	100
Initial Value	11

Slider Property	Value
Control Name	APF4 g
Units	
Variable Type	float
Variable Name	m_fAPF_4_g
Low Limit	−1
High Limit	1
Initial Value	0.6

Reverb Output

Slider Property	Value
Control Name	Reverb Time
Units	mSec
Variable Type	float
Variable Name	m_fRT60
Low Limit	0
High Limit	5000
Initial Value	1000

Slider Property	Value
Control Name	Wet/Dry
Units	%
Variable Type	float
Variable Name	m_fWet_pct
Low Limit	0
High Limit	100
Initial Value	50

Build the UI using the list of controls above; the variable names I used in the project are listed in the table. You can use your own variable names but you will need to make sure they are correctly implemented. Figure 11.46 shows my GUI; I grouped sets of controls together, following the algorithm block diagram.

11.17.3 Reverb.h

In the .h file, declare the member objects plus one cooking function that will change all modules at once (it's not streamlined but it is simple to cook all at once). There is a stock object for each block in the algorithm.

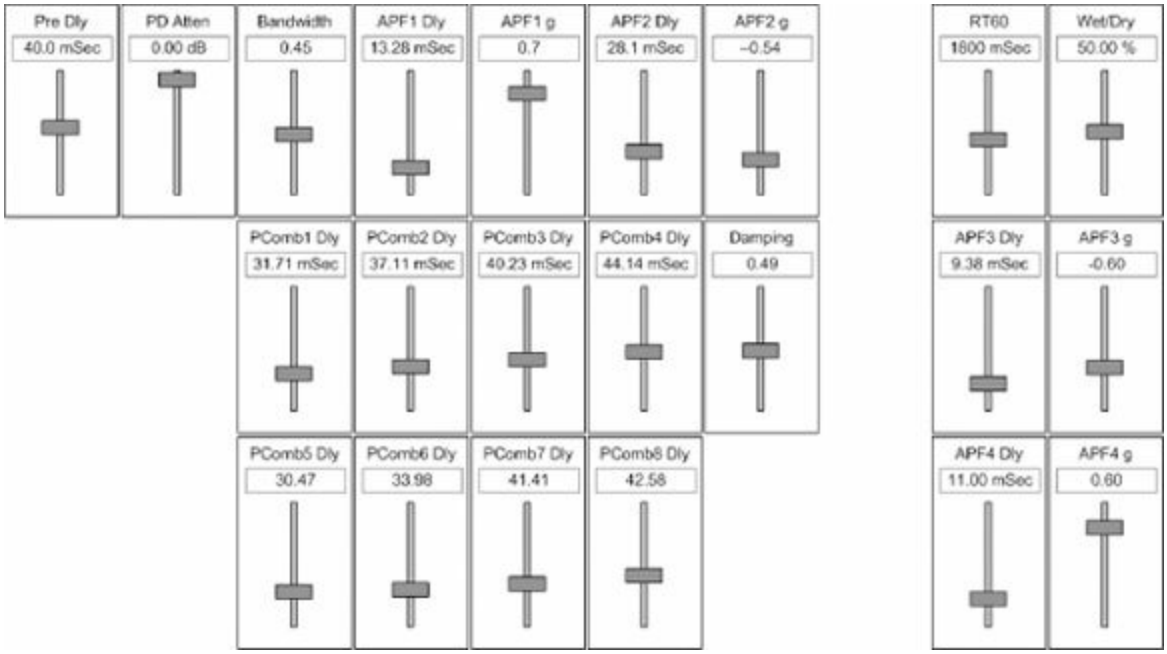


Figure 11.46: The completed reverb prototype GUI.

```
// Add your code here: ---------------------------- //
//
// Pre-Delay Block
CDelay m_PreDelay;

// input Diffusion
COnePoleLPF m_InputLPF;
CDelayAPF m_InputAPF_1;
CDelayAPF m_InputAPF_2;

// parallel Comb Bank 1
CCombFilter m_ParallelCF_1;
CCombFilter m_ParallelCF_2;
CLPFCombFilter m_ParallelCF_3;
CLPFCombFilter m_ParallelCF_4;

// parallel Comb Bank 2
CCombFilter m_ParallelCF_5;
CCombFilter m_ParallelCF_6;
CLPFCombFilter m_ParallelCF_7;
CLPFCombFilter m_ParallelCF_8;

// damping
COnePoleLPF m_DampingLPF1;
COnePoleLPF m_DampingLPF2;

// output diffusion
CDelayAPF m_OutputAPF_3;
CDelayAPF m_OutputAPF_4;

// function to cook all member object's variables at once
void cookVariables();

// END OF USER CODE ----------------------------- //
```

11.17.4 Reverb.cpp

Write the only extra function, cookVariables():

```
// function to cook all variables at once
void CReverb::cookVariables()
{
	// Pre-Delay
	m_PreDelay.setDelay_mSec(m_fPreDelay_mSec);
	m_PreDelay.setOutputAttenuation_dB(m_fPreDelayAtten_dB);

	// input diffusion
	m_InputAPF_1.setDelay_mSec(m_fAPF_1_Delay_mSec);
	m_InputAPF_1.setAPF_g(m_fAPF_1_g);
```

```
        m_InputAPF_2.setDelay_mSec(m_fAPF_2_Delay_mSec);
        m_InputAPF_2.setAPF_g(m_fAPF_2_g);

        // output diffusion
        m_OutputAPF_3.setDelay_mSec(m_fAPF_3_Delay_mSec);
        m_OutputAPF_3.setAPF_g(m_fAPF_3_g);

        m_OutputAPF_4.setDelay_mSec(m_fAPF_4_Delay_mSec);
        m_OutputAPF_4.setAPF_g(m_fAPF_4_g);

        // comb filters
        // set delays first...
        m_ParallelCF_1.setDelay_mSec(m_fPComb_1_Delay_mSec);
        m_ParallelCF_2.setDelay_mSec(m_fPComb_2_Delay_mSec);
        m_ParallelCF_3.setDelay_mSec(m_fPComb_3_Delay_mSec);
        m_ParallelCF_4.setDelay_mSec(m_fPComb_4_Delay_mSec);
        m_ParallelCF_5.setDelay_mSec(m_fPComb_5_Delay_mSec);
        m_ParallelCF_6.setDelay_mSec(m_fPComb_6_Delay_mSec);
        m_ParallelCF_7.setDelay_mSec(m_fPComb_7_Delay_mSec);
        m_ParallelCF_8.setDelay_mSec(m_fPComb_8_Delay_mSec);


        // …then calculate comb g's from RT60:
        m_ParallelCF_1.setComb_g_with_RTSixty(m_fRT60);
        m_ParallelCF_2.setComb_g_with_RTSixty(m_fRT60);
        m_ParallelCF_3.setComb_g_with_RTSixty(m_fRT60);
        m_ParallelCF_4.setComb_g_with_RTSixty(m_fRT60);
        m_ParallelCF_5.setComb_g_with_RTSixty(m_fRT60);
        m_ParallelCF_6.setComb_g_with_RTSixty(m_fRT60);
        m_ParallelCF_7.setComb_g_with_RTSixty(m_fRT60);
        m_ParallelCF_8.setComb_g_with_RTSixty(m_fRT60);

        // LPFs
        m_DampingLPF1.setLPF_g(m_fLPF2_g2);
        m_DampingLPF2.setLPF_g(m_fLPF2_g2);
        m_InputLPF.setLPF_g(m_fInputLPF_g);

        // LPF-comb filters
        m_ParallelCF_3.setLPF_g(m_fLPF2_g2);
        m_ParallelCF_4.setLPF_g(m_fLPF2_g2);
        m_ParallelCF_7.setLPF_g(m_fLPF2_g2);
        m_ParallelCF_8.setLPF_g(m_fLPF2_g2);

}
```

Constructor

There is nothing to do here because the child-object constructors handle these details.

prepareForPlay()

- Initialize all the objects with their max delay times; all delay times except the pre-delay have maximum values of 100 mSec.
- Initialize the pre-delay for its maximum of 2 seconds.
- Reset all delays.
- Set the sample rate for the delays.
- Cook all the variables at once.
- Init the delays to flush buffers.

```
bool __stdcall CReverb::prepareForPlay()
{
    // Add your code here:
      // up to 2 seconds predelay
      m_PreDelay.init(2.0*(m_nSampleRate));

      // init up to 100 mSec
      m_InputAPF_1.init(0.1*(m_nSampleRate));
      m_InputAPF_2.init(0.1*(m_nSampleRate));

      // 100 mSec each max
      m_ParallelCF_1.init(0.1*(m_nSampleRate));
      m_ParallelCF_2.init(0.1*(m_nSampleRate));
      m_ParallelCF_3.init(0.1*(m_nSampleRate));
      m_ParallelCF_4.init(0.1*(m_nSampleRate));
      m_ParallelCF_5.init(0.1*(m_nSampleRate));
      m_ParallelCF_6.init(0.1*(m_nSampleRate));
      m_ParallelCF_7.init(0.1*(m_nSampleRate));
      m_ParallelCF_8.init(0.1*(m_nSampleRate));


      // 100 mSec each max
      m_OutputAPF_3.init(0.1*(m_nSampleRate));
      m_OutputAPF_4.init(0.1*(m_nSampleRate));

      // init the three LPFs
      m_InputLPF.init();
      m_DampingLPF1.init();
      m_DampingLPF2.init();

      // Call all delay resets
      m_PreDelay.resetDelay();
      m_InputAPF_1.resetDelay();
      m_InputAPF_2.resetDelay();

      m_ParallelCF_1.resetDelay();
      m_ParallelCF_2.resetDelay();
      m_ParallelCF_3.resetDelay();
      m_ParallelCF_4.resetDelay();
      m_ParallelCF_5.resetDelay();


      m_ParallelCF_6.resetDelay();
      m_ParallelCF_7.resetDelay();
      m_ParallelCF_8.resetDelay();

      m_OutputAPF_3.resetDelay();
      m_OutputAPF_4.resetDelay();

      // set sample rates on combs (needed to calc g values)
      m_ParallelCF_1.setSampleRate(this->m_nSampleRate);
      m_ParallelCF_2.setSampleRate(this->m_nSampleRate);
      m_ParallelCF_3.setSampleRate(this->m_nSampleRate);
      m_ParallelCF_4.setSampleRate(this->m_nSampleRate);
      m_ParallelCF_5.setSampleRate(this->m_nSampleRate);
      m_ParallelCF_6.setSampleRate(this->m_nSampleRate);
      m_ParallelCF_7.setSampleRate(this->m_nSampleRate);
      m_ParallelCF_8.setSampleRate(this->m_nSampleRate);

      // flush buffers
      m_InputLPF.init();
      m_DampingLPF1.init();
      m_DampingLPF2.init();

      // cook everything
      cookVariables();
      return true;
}
```

userInterfaceChange()

- Call the one and only cooking function

```
bool __stdcall CReverb::userInterfaceChange(int nControlIndex)
{
	// add your code here
	cookVariables();
	return true;
}
```

processAudioFrame()

In this function, you tie all the member objects together by creating variables to pass into/out of their processAudio() functions. By stepping through the code, you can see just how the series and parallel elements are connected. You can always go back and make this code more streamlined by removing intermediate variables, and so on. The four outputs of each comb filter bank are mixed at 15% ratios with alternating signs.

```
bool __stdcall CReverb::processAudioFrame(float* pInputBuffer, float* pOutputBuffer, UINT
                                          uNumInputChannels, UINT uNumOutputChannels)
{
	//
	// output = input - change this for meaningful processing

	//
	// Form our input = L + R (if there is a R)
	//
	float fInputSample = pInputBuffer[0];
	if(uNumInputChannels == 2)
	{
		// mix
		fInputSample += pInputBuffer[1];
		// attenuate by 0.5
		fInputSample *= 0.5;
	}

	// begin the series/parallel signal push
	// Pre-Delay
	float fPreDelayOut = 0;
	m_PreDelay.processAudio(&fInputSample, &fPreDelayOut);

	// Pre-Delay Out -> fAPF_1_Out
	float fAPF_1_Out = 0;
	m_InputAPF_1.processAudio(&fPreDelayOut, &fAPF_1_Out);

	// fAPF_1_Out -> fAPF_2_Out
	float fAPF_2_Out = 0;
	m_InputAPF_2.processAudio(&fAPF_1_Out, &fAPF_2_Out);

	// fAPF_2_Out -> fInputLPF
	float fInputLPF = 0;
	m_InputLPF.processAudio(&fAPF_2_Out, &fInputLPF);
```

```
	// comb filter bank
	// variables for each output
	float fPC_1_Out = 0;
	float fPC_2_Out = 0;
	float fPC_3_Out = 0;
	float fPC_4_Out = 0;
	float fPC_5_Out = 0;
	float fPC_6_Out = 0;
	float fPC_7_Out = 0;
	float fPC_8_Out = 0;
	float fC1_Out = 0;
	float fC2_Out = 0;

	// fInputLPF -> fPC_1_Out, fPC_2_Out, fPC_3_Out, fPC_4_Out
	m_ParallelCF_1.processAudio(&fInputLPF, &fPC_1_Out);
	m_ParallelCF_2.processAudio(&fInputLPF, &fPC_2_Out);
	m_ParallelCF_3.processAudio(&fInputLPF, &fPC_3_Out);
	m_ParallelCF_4.processAudio(&fInputLPF, &fPC_4_Out);

	// fInputLPF -> fPC_5_Out, fPC_6_Out, fPC_7_Out, fPC_8_Out
	m_ParallelCF_5.processAudio(&fInputLPF, &fPC_5_Out);
	m_ParallelCF_6.processAudio(&fInputLPF, &fPC_6_Out);


	m_ParallelCF_7.processAudio(&fInputLPF, &fPC_7_Out);
	m_ParallelCF_8.processAudio(&fInputLPF, &fPC_8_Out);

	// form outputs: note attenuation by 0.15 for each and alternating signs
	fC1_Out = 0.15*fPC_1_Out - 0.15*fPC_2_Out + 0.15*fPC_3_Out - 0.15*fPC_4_Out;
	fC2_Out = 0.15*fPC_5_Out - 0.15*fPC_6_Out + 0.15*fPC_7_Out - 0.15*fPC_8_Out;

	// fC1_Out -> fDamping_LPF_1_Out
	float fDamping_LPF_1_Out = 0;
	m_DampingLPF1.processAudio(&fC1_Out, &fDamping_LPF_1_Out);

	// fC2_Out -> fDamping_LPF_2_Out
	float fDamping_LPF_2_Out = 0;
	m_DampingLPF2.processAudio(&fC2_Out, &fDamping_LPF_2_Out);

	// fDamping_LPF_1_Out -> fAPF_3_Out
	float fAPF_3_Out = 0;
	m_OutputAPF_3.processAudio(&fDamping_LPF_1_Out, &fAPF_3_Out);

	// fDamping_LPF_2_Out -> fAPF_4_Out
	float fAPF_4_Out = 0;
	m_OutputAPF_4.processAudio(&fDamping_LPF_2_Out, &fAPF_4_Out);

	// form output = (100-Wet)/100*x(n) + (Wet/100)*fAPF_3_Out
	pOutputBuffer[0] = ((100.0 - m_fWet_pct)/100.0)*fInputSample +
	                   (m_fWet_pct/100.0)*(fAPF_3_Out);

	// Do RIGHT Channel if there is one
	if(uNumOutputChannels == 2)
	{
		// form output = (100-Wet)/100*x(n) + (Wet/100)*fAPF_4_Out
		pOutputBuffer[1] = ((100.0 - m_fWet_pct)/100.0)*fInputSample +
		                   (m_fWet_pct/100.0)*(fAPF_4_Out);
	}
	return true;
}
```

Build and test the example reverb module. Use RackAFX's impulse response tool in the analyzer to tweak your settings. Adjust the impulse response time according to your RT_{60} times. Notice how the impulse response can drastically change with only a slight modification of the parameters. Also, try adjusting the mix coefficients for the comb filter outputs by alternating signs or using different

weightings. Visit the website www.willpirkle.com for more example reverb algorithms and code.

11.18 Challenge

Design your own reverb. Start by implementing some of the classics (Schroeder, Moorer) and some of the more recent versions (Gardner, Jot, Dattorro) combining different modules. Or start with the reverb design here and modify it. For example, try replacing the APFs with nested all-pass modules or add or remove comb filters. You can easily identify ringing and oscillations using the impulse response tool, so keep it open as you experiment.

Bibliography

Griesinger, D. 1995. How loud is my reverberation? *Journal of the Audio Engineering Society*, preprint 3943, pp. 1–11.

Roads, C. 1996. *The Computer Music Tutorial*, Chapter 3. Cambridge, MA: The MIT Press.

References

Beranek, L. 1986. *Acoustics*. New York: American Institute of Physics.

Browne, S. 2001. "Hybrid reverberation algorithm using truncated impulse response convolution and recursive filtering." Master's diss., University of Miami, Miami, FL.

Chemistruck, M., Marcolini, K., and Pirkle, W. 2012. Generating matrix coefficients for feedback delay networks using genetic algorithms. *Journal of the Audio Engineering Society*, preprint N/A.

Dahl, L. and Jot, J-M. 2000. A reverberator based on absorbent all-pass filters. *Proceedings of the COST G-6 Convention on Digital Audio Effects*, DAFX-00, pp. 1–6.

Dattorro, J. 1997. Effect design part 1: Reverberators and other filters. *Journal of the Audio Engineering Society* 45(9): 660–665.

Frenette, J. 2000. "Reducing artificial reverberation requirements using time-variant feedback delay networks." Master's diss., University of Miami, Miami, FL.

Gardner, W. G. 1992. "The virtual acoustic room." Master's diss., Massachusetts Institute of Technology, Boston, MA.

Gardner, W. G. 1995. Efficient convolution without input-output delay. *Journal of the Audio Engineering Society* 43(3): 127–135.

Gerzon, M. A. Synthetic stereo reverberation. *Studio Sound Magazine*, January 14, 1972.

Griesinger, D. 1989. Practical processors and programs for digital reverberation. *7th International Conference of the Audio Engineering Society*, pp. 187–195.

Jot, J-M. and Chaigne, A. 1991. Digital delay networks for designing artificial reverberators. *Journal of the Audio Engineering Society*, preprint 3030, pp. 1–16.

Jot, J-M. 1992. "Design and implementation of a sound spatializer based on physical and perceptual models." PhD diss., Telecom, Paris.

Kahrs, M. and Brandenberg, K. 1998. *Applications of Digital Signal Processing to Audio and Acoustics*, Chapter 3. Boston: Kluwer Academic Publishers.

Kuttruff, H. 1991. *Room Acoustics*. New York: Elsevier.

Moorer, J. A. 1979. About this reverberation business. *Computer Music Journal* 3(2): 13–28.

Reilly, A. and McGrath, D. S. 1995. Convolution processing for realistic reverb. *Journal of the Audio Engineering Society*, preprint 3977, pp. 1–8.

Sabine, W. 1973 "Reverberation." Lindsay, R. B., ed., *Acoustics: Historical and Philosophical Development*. Stroudsburg, PA: Dowden, Hutchinson & Ross.

Schroeder, M. 1962. Natural-sounding artificial reverberation. *Journal of the Audio Engineering Society* 10(3): 209–213.

Schroeder, M. 1984. Progress in architectural acoustics and artificial reverberation: Concert hall acoustics and number theory. *Journal of the Audio Engineering Society* 32(4): 194–202.

Smith, J. O. 1985. A new approach to digital reverberation using closed waveguide networks. *Proceedings of the 1985 International Computer Music Conference*, pp. 47–53.

Smith, J. O. and Rocchesso, D. 1994. Connections between feedback delay networks and waveguide networks for digital reverberation. *Proceedings of the 1994 International Computer Music Conference, pp. 376–377.*

Stautner, J. and Puckette, M. 1982. Designing multi-channel reverberators. *Computer Music Journal* 6(1): 52–65.

CHAPTER 12

Modulated Filter Effects

Modulated filter effects alter one or more filter parameters with other signals called control signals. The filter parameters might include cutoff frequency, *Q*, bandwidth, filter type, or overall gain. The control signals are usually periodic low-frequency oscillators (LFOs), a signal envelope, an envelope generator (EG) or even another audio signal. In the modulated delay effects, the LFO control signal changed the delay amount of the signal. In modulated filter effects, the filter parameters are the ones that are changing. In this chapter you will design and implement the following effects:

- Mod filter using an LFO
- Envelope follower using an audio detector
- Phaser

The phaser is a specialized effect that uses a bank of all-pass filters (APFs) to try to brute force delay the signal to make a flanging effect. An LFO modulates the depth of the APFs. In

Figure 12.1 you can see a simple modulated filter. The filter type is a low-pass filter (LPF). The control signal is an LFO. The control parameter is the cutoff frequency of the filter.

In Figure 12.2, an EG moves the filter parameter. An EG is triggered by some external event, such as a note-on or -off event, or possibly triggered when the input level crosses a threshold. A modulated filter might have multiple parameters controlled by multiple control signals. For example, Figure 12.3 shows another LPF that has two modulation sources for two different parameters.

In Figure 12.3 you can see a new module labeled envelope detector; it detects and follows the peak, mean-squared (MS), or root-mean-squared (RMS) value of the control signal which is the input signal here. The control signal could be taken from other sources such as another audio signal or a complex LFO signal. The effect in Figure 12.2 has a special name too: it is called an envelope follower.

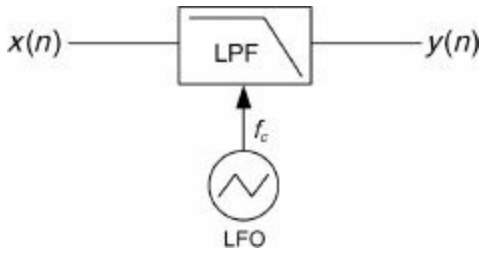


Figure 12.1: A simple LFO-modulated LPF.

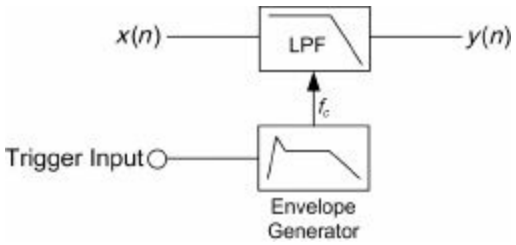


Figure 12.2: An envelope generator can also modulate a filter parameter.

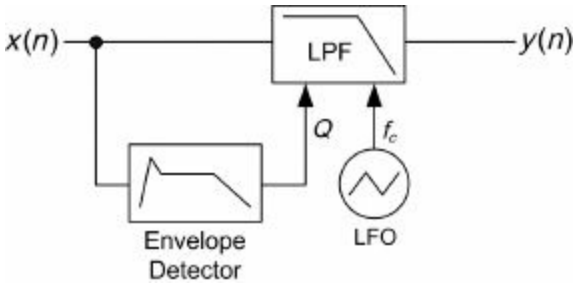


Figure 12.3: A doubly modulated LPF with both f_c and Q controls.

12.1 Design a Mod Filter Plug-In: Part I Modulated f_c

For our first effect design, we'll start with a modulated second-order LPF and modulate the cutoff frequency with an LFO. Then, we can increase the complexity by adding another LFO to control the Q and even give the option to run the two LFOs in quadrature phase. We can use the second-order digital resonant LPF you've already designed from Chapter 6 for the filter. Notice that for this initial project, we will hold the LPF Q constant at 2.0. And, we will introduce a built-in RackAFX object to handle the LFO for us. The block diagram is shown in Figure 12.4.

The parameters are as follows:

- f_c mod rate (0.2 to 10 Hz)
- f_c mod depth (0 to 100%)
- LFO type (sine, tri, saw, square)
- LPF Q: fixed at 2.0
- f_c mod range: 100 Hz to = kHz
- LPF is fixed as second-order bi-quad variety

By now, you should be getting really good at building RackAFX plug-ins. If you haven't just been cutting and pasting the code then you will have no problems with the next chapters. We can use the digital resonant LPF from Chapter 6 along with the built-in wave table oscillator object to quickly implement the mod filter effect. This project will use two built-in objects:

1. CBiquad for the filter
2. CWaveTable for the LFO

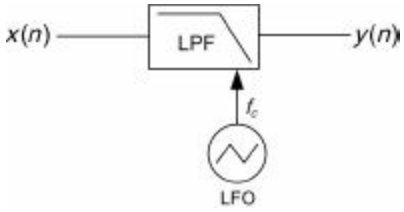


Figure 12.4: The mod filter block diagram.

You used CBiquad in Chapter 6 to begin your filtering work. The CWaveTable object was taken directly from your WTOscillator object. You can find the definition and implementation in the pluginconstants.h and pluginonbjects.cpp files. Table 12.1 shows the object's members and attributes. The flow chart for processAudioFrame() is shown in Figure 12.5.

12.1.1 Project: ModFilter

Create the project; because we are using built-in objects for the filter and LFO there are no member objects to add.

12.1.2 ModFilter GUI

For the initial design, you will need the following slider controls in Table 12.2. Note that these LFO waveform enumerations follow exactly that of the built-in CWaveTable object to make mapping the button control easy.

12.1.3 ModFilter.h File

Declare instances of the BiQuad and WaveTable objects. You don't need to #include anything since these are built-in:

```
// Add your code here: ------------------------------------------------------ //
// BiQuad Objects
CBiquad m_LeftLPF;
CBiquad m_RightLPF;

// one LFO for the fc
CWaveTable m_fc_LFO;
// END OF USER CODE --------------------------------------------------------- //
```

We are also going to need some variables and functions to control the effect. Specifically, we need:

- Minimum f_c variable
- Maximum f_c variable
- Function to calculate the current f_c given a LFO value
- Functions to calculate the bi-quad coefficients for left and right channels independently (we want to calculate them separately for the third part of the project where we allow for quad-phase LFOs)

Table 12.1: The CWaveTable object interface.

CWaveTable	
Member Variables	**Purpose**
float m_SinArray[1024]; float m_SawtoothArray[1024]; float m_TriangleArray[1024]; float m_SquareArray[1024];	Arrays to store the non-band limited wave tables
float m_SawtoothArray_BL5[1024]; float m_TriangleArray_BL5[1024]; float m_SquareArray_BL5[1024];	Arrays to hold the band-limited wave tables
float m_fReadIndex; float m_fQuadPhaseReadIndex;	Indices for reading the wave table
float m_f_inc;	Table increment value for current output frequency
int m_nSampleRate;	Sample rate *IMPORTANT: Needed for calculation of table inc; do not forget it!*
float m_fFrequency_Hz;	Oscillator frequency
bool m_bInvert;	Flag to invert output
UINT m_uOscType; enum{sine, saw, tri, square};	Enumerated unsigned integer type (UINT) for osc waveform type
UINT m_uTableMode; enum{normal, bandlimit};	Enumerated UINT for table mode (we will use normal mode in this plug-in since the oscillator is a LFO)
UINT m_uPolarity; enum{bipolar, unipolar};	Enumerated UINT for osc polarity (we will use unipolar for this plug-in)
Member Functions	
void doOscillate(float* pYn, float* pYqn) Parameters:	The oscillate function
• float* pYn • float* pYqn	Output: pYn is the normal output Output: pYqn is the quadrature phase output
bool prepareForPlay( )	The prepareForPlay( ) function for the oscillator; same use as a

	normal plug-in
void reset ( )	Reset the pointers to top
void setSampleRate(int nSampleRate) Parameters:	Called by parent to set the sample rate in Hz
• int nSampleRate	Input: the current sample rate
void cookFrequency( )	Calculates the new *inc* value for a changed oscillation frequency

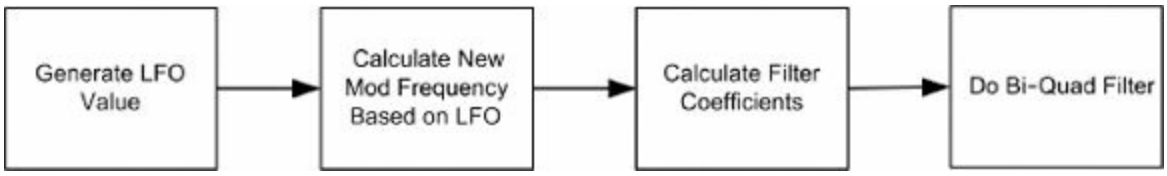


Figure 12.5: The flowchart for the mod filter process function.

Table 12.2: GUI controls for the ModFilter.

Slider Property	Value
Control Name	Mod Depth fc
Units	%
Variable Type	float
Variable Name	m_fMod_Depth_fc
Low Limit	0
High Limit	100
Initial Value	50

Slider Property	Value
Control Name	Mod Rate fc
Units	Hz
Variable Type	float
Variable Name	m_fModRate_fc
Low Limit	0.2
High Limit	10
Initial Value	1.0

Slider Property	Value
Control Name	LFO
Units	

Variable Type	enum
Variable Name	m_uLFO_waveform
Enum String	sine, saw, tri, square

Add the following to the .h file:

```
// Add your code here: ------------------------------------------------------ //
// BiQuad Objects
CBiquad m_LeftLPF;
CBiquad m_RightLPF;

// one LFO for the fc
CWaveTable m_fc_LFO;

// min and max values (to make it easy to change later)
float m_fMinCutoffFreq;
float m_fMaxCutoffFreq;

// function to calculate the fc given the LFO sample
float calculateCutoffFreq(float fLFOSample);

// calculate the coeffs for a given filter
void calculateLPFCoeffs(float fCutoffFreq, float fQ, CBiquad* pFilter);

// END OF USER CODE ---------------------------------------------------------- //
```

12.1.4 ModFilter.cpp File

Constructor

- Initialize the rate min and max values. The objects are self initializing upon creation.

```
CModFilter::CModFilter()
{
        <SNIP SNIP SNIP>

        // Finish initializations here

        // set our Min and Max Modulation points
        m_fMinCutoffFreq = 100.0;
        m_fMaxCutoffFreq = 5000.0;
}
```

Write the calculateCutoffFrequency() function. This function behaves exactly like the function in the flanger to calculate a new delay offset given the LFO sample and mod depth values. Here, we simply calculate the cutoff frequency between the min and max values according to the LFO sample and depth.

```
float CModFilter::calculateCutoffFreq(float fLFOSample)
{
       return (m_fMod_Depth_fc/100.0)*(fLFOSample*(
               m_fMaxCutoffFreq - m_fMinCutoffFreq)) + m_fMinCutoffFreq;
}
```

Next, we need to implement the cooking function for the LPFs to set the new coefficients for a given f_c and Q value. This is identical to the resonant LPF you designed in Chapter 6:

```
void CModFilter::calculateLPFCoeffs(float fCutoffFreq, float fQ, CBiquad* pFilter)
{
	// use same terms as book
	float theta_c = 2.0*pi*fCutoffFreq/(float)m_nSampleRate;
	float d = 1.0/fQ;

	float fBetaNumerator =  1.0 - ((d/2.0)*(sin(theta_c)));
	float fBetaDenominator = 1.0 + ((d/2.0)*(sin(theta_c)));

	float fBeta = 0.5*(fBetaNumerator/fBetaDenominator);

	float fGamma = (0.5 + fBeta)*(cos(theta_c));

	float fAlpha = (0.5 + fBeta - fGamma)/2.0;

	// apply to filter
	pFilter->m_f_a0 = fAlpha;
	pFilter->m_f_a1 = 2.0*fAlpha;
	pFilter->m_f_a2 = fAlpha;
	pFilter->m_f_b1 = -2.0*fGamma;
	pFilter->m_f_b2 = 2.0*fBeta;
}
```

prepareForPlay()

- Flush delays in LPFs.
- Set up the LFOs by setting the initial rate and LFO type (the LFO type will match our LFO_waveform variable).

```
bool __stdcall CModFilter::prepareForPlay()
{
	// Add your code here:
	//
	// Flush the LPFs
	m_LeftLPF.flushDelays();
	m_RightLPF.flushDelays();
	// Note we do NOT need to init the cutoff and Q;
	// these are done in processAudioFrame()

	// setup LFO
	m_fc_LFO.m_fFrequency_Hz = m_fModRate_fc;
	m_fc_LFO.m_uPolarity = 1;// 0 = bipolar, 1 = unipolar
	m_fc_LFO.m_uTableMode = 0; // normal, no band limiting
	m_fc_LFO.m_uOscType = m_uLFO_Waveform; // our own variable
	m_fc_LFO.setSampleRate(m_nSampleRate); // really important!

	// the LFO prepareForPlay() calls reset() and cookFrequency()
	m_fc_LFO.prepareForPlay();

	return true;
}
```

userInterfaceChange()

- If the user moves the rate slider, we need to change the LFO rate.
- If the user clicks on a radio button, we need to change the LFO waveform type.
- For now, simply recalculate everything when any control is moved; you can streamline this later.

```
bool __stdcall CModFilter::userInterfaceChange(int nControlIndex)
{
	// brute force update all
	m_fc_LFO.m_fFrequency_Hz = m_fModRate_fc;
	m_fc_LFO.m_uOscType = m_uLFO_Waveform;
	// cook to calculate
	m_fc_LFO.cookFrequency();

	return true;
}
```

processAudioFrame()

- Calculate a new LFO value.
- Use the LFO value to calculate a new f_c value.
- Use the f_c value to calculate new filter coefficients.
- Do the bi-quad routines on the input samples.

```
bool __stdcall CModFilter::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
                                             UINT uNumInputChannels, UINT
                                             uNumOutputChannels)
{
	//
	// output = input -- change this for meaningful processing
	//
	float fYn = 0; // normal output
	float fYqn = 0; // quad phase output

	// call the LFO function; we only need first output
	m_fc_LFO.doOscillate(&fYn, &fYqn);

	// use the LFO value to calculate the updated fc value
	float fc = calculateCutoffFreq(fYn);

	// use the fc value and a preset Q = 2.0
	// to calculate the LPF coefficients for each channel
	calculateLPFCoeffs(fc, 2.0, &m_LeftLPF);
	calculateLPFCoeffs(fc, 2.0, &m_RightLPF);


	// do the BiQuad operation on the Left LPF
	pOutputBuffer[0] = m_LeftLPF.doBiQuad(pInputBuffer[0]);

	// Mono-In, Stereo-Out (AUX Effect)
	if(uNumInputChannels == 1 && uNumOutputChannels == 2)
		pOutputBuffer[1] = pOutputBuffer[0]; // just copy

	// Stereo-In, Stereo-Out (INSERT Effect)
	if(uNumInputChannels == 2 && uNumOutputChannels == 2)
		pOutputBuffer[1] = m_RightLPF.doBiQuad(pInputBuffer[1]);

	return true;
}
```

Build and test the plug-in. You will hear the f_c modulated by the LFO; adjust the LFO rate and depth controls and try changing the LFO waveform type as well; triangle and sine seem to be the most common choices, and the saw produces an interesting pulsing effect.

12.2 Design a Mod Filter Plug-In: Part II, Modulated f_c, Q

In the second design, we will modify the current plug-in to include the modulation of the LPF Q value. We will use a second, independent LFO for the Q but will share the same LFO waveform type with the f_c LFO. The block diagram is shown in Figure 12.6.

The parameters are as follows:

- f_c mod rate (0.2 to 10 Hz)
- f_c mod depth (0 to 100%)
- Q mod rate (0.2 to 10 Hz)
- Q mod depth (0 to 100%)
- LFO type (sine, tri, saw, square)
- f_c mod range: 100 Hz to = kHz
- Q mod range: 0.5 to 10
- LPF is fixed as second-order bi-quad

12.2.1 ModFilter GUI

Add more sliders to the user interface (UI) for the new LFO as per Table 12.3.

Table 12.3: Additional controls for the second LFO.

Slider Property	Value
Control Name	Mod Depth Q
Units	%
Variable Type	float
Variable Name	m_fMod_Depth_Q
Low Limit	0
High Limit	100
Initial Value	50

Slider Property	Value
Control Name	Mod Rate Q
Units	Hz
Variable Type	float
Variable Name	m_fModRate_Q
Low Limit	0.2
High Limit	10
Initial Value	0.5

12.2.2 ModFilter.h File

We need to add the following new functions and variables, basically just duplicating the ones for the f_{c} $_{m}$modulation:

- LFO object for *Q* modulation
- calculateQ() to calculate the new *Q* value from the LFO sample value
- Min and max *Q* values

```
// Add your code here: ----------------------------------------------------- //
CBiquad m_LeftLPF;
CBiquad m_RightLPF;

CWaveTable m_fc_LFO;
CWaveTable m_Q_LFO;

float m_fMinCutoffFreq;
float m_fMaxCutoffFreq;

float m_fMinQ;
float m_fMaxQ;

// functions to calculate the fc or Q for a given LFO sample
float calculateCutoffFreq(float fLFOSample);
float calculateQ(float fLFOSample);

// calculate the coeffs for a given filter
void calculateLPFCoeffs(float fCutoffFreq, float fQ, CBiquad* pFilter);

// END OF USER CODE ------------------------------------------------------------
```

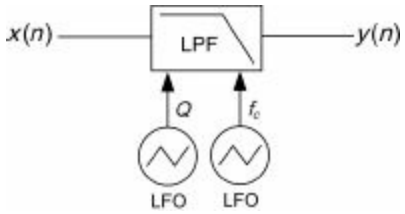


Figure 12.6: The mod filter with second LFO for Q.

12.2.3 ModFilter.cpp File

Constructor

- Initialize the min and max *Q* values.

```
CModFilter::CModFilter()
{
        m_PlugInName = "ModFilter";

        // Default to Stereo Operation:
        // Change this if you want to support more/less channels
        m_uMaxInputChannels = 2;
        m_uMaxOutputChannels = 2;

        // Finish initializations here
        // set our Min and Max Modulation points
        m_fMinCutoffFreq = 100.0;
        m_fMaxCutoffFreq = 5000.0;
        m_fMinQ = 0.577;
        m_fMaxQ = 10;
}
```

Write the calculateQ() function, which behaves the same way as calculateCutoffFreq(); you only need to change the mod depth source and min and max values:

```
float CModFilter::calculateQ(float fLFOSample)
{
	return (m_fMod_Depth_Q/100.0)*(fLFOSample*(m_fMaxQ - m_fMinQ)) + m_fMinQ;
}
```

prepareForPlay()

- Set up the *Q* LFO by setting the initial rate and LFO type (the LFO type will match our m_uLFOwaveform variable).

```
bool __stdcall CModFilter::prepareForPlay()
{
	// Add your code here:
	<SNIP SNIP SNIP>

	// the LFO prepareForPlay() calls reset() and cookFrequency()
	m_fc_LFO.prepareForPlay();

	// now do the Q LFO
	m_Q_LFO.m_fFrequency_Hz = m_fModRate_Q;
	m_Q_LFO.m_uPolarity = 1; // 0 = bipolar, 1 = unipolar
	m_Q_LFO.m_uTableMode = 0; // normal, no band limiting
	m_Q_LFO.m_uOscType = m_uLFO_Waveform;
	m_Q_LFO.setSampleRate(m_nSampleRate); // really important!

	// this calls reset() and cookFrequency()
	m_Q_LFO.prepareForPlay();

	return true;
}
```

userInterfaceChange()

- Update the *Q* LFO when the user changes this value.

```
bool __stdcall CModFilter::userInterfaceChange(int nControlIndex)
{
	// add your code here
	// brute force update all
	m_fc_LFO.m_fFrequency_Hz = m_fModRate_fc;
	m_fc_LFO.m_uOscType = m_uLFO_Waveform;

	m_Q_LFO.m_fFrequency_Hz = m_fModRate_Q;
	m_Q_LFO.m_uOscType = m_uLFO_Waveform;

	// cook to calculate
	m_fc_LFO.cookFrequency();
	m_Q_LFO.cookFrequency();

	etc…
}
```

processAudioFrame()

- Calculate a new f_c LFO value.
- Use the LFO value to calculate a new f_c value.
- Calculate a new Q LFO value.
- Use the LFO value to calculate a new Q value.
- Use the f_c and Q values to calculate new filter coefficients.
- Do the bi-quad routines on the input samples.

```
bool __stdcall CModFilter::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
                                             UINT uNumInputChannels, UINT
                                             uNumOutputChannels)
{
        //
        // output = input -- change this for meaningful processing
        //

        float fYn = 0; // normal output
        float fYqn = 0; // quad phase output

        // call the fc LFO function; we only need first output
        m_fc_LFO.doOscillate(&fYn, &fYqn);

        // calculate fc
        float fc = calculateCutoffFreq(fYn);

        // call the Q LFO funciton
        m_Q_LFO.doOscillate(&fYn, &fYqn);

        // calculate the new Q
        float fQ = calculateQ(fYn);

        // use to calculate the LPF
        calculateLPFCoeffs(fc, fQ, &m_LeftLPF);
        calculateLPFCoeffs(fc, fQ, &m_RightLPF);

        // do the BiQuads
        pOutputBuffer[0] = m_LeftLPF.doBiQuad(pInputBuffer[0]);

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0]; // just copy

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                pOutputBuffer[1] = m_RightLPF.doBiQuad(pInputBuffer[1]);

        return true;
}
```

You can see that we only had to add a few lines of code to get the second modulation source working. Build and test the plug-in. You will hear the f_c and Q modulated by the two LFOs; adjust the LFO rate and depth controls and try changing the LFO waveform type too. The Q modulation can be more subtle to hear, but if you use RackAFX's oscillator and pump white noise through it, the modulation becomes clearly audible. We'll finish the plug-in by making one more modification: the ability to place the right and left LPF modulation sources in quadrature phase.

12.3 Design a Mod Filter Plug-In: Part III, Quad-Phase LFOs

In the third design iteration, we will modify the current plug-in to allow for quadrature phase LFOs. The block diagram is given in Figure 12.7.

The parameters are as follows:

- f_c mod rate (0.2 to 10 Hz)
- f_c mod depth (0 to 100%)
- Q mod rate (0.2 to 10 Hz)
- Q mod depth (0 to 100%)
- LFO type (sine, tri, saw, square)
- LFO phase (normal, quadrature)
- f_c mod range: 100Hz to 5kHz
- Q mod range: 0.577 to 10
- LPF is fixed as second-order bi-quad

In order to switch the LFOs into quadrature phase, we only need to add one more radio button control and make a slight modification to the processAudioFrame() function.

12.3.1 ModFilter GUI

Add the Radio Button Control in Table 12.4. Note: This is a direct control variable—there is nothing extra to add in userInterfaceChange() or prepareForPlay() since we only use it in the single processAudioFrame() function.

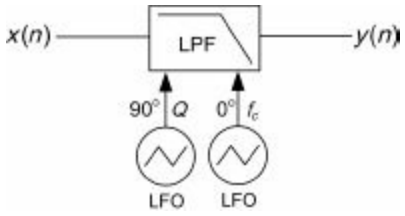


Figure 12.7: The mod filter with quad-phase LFOs.

12.3.2 ModFilter.cpp File

processAudioFrame()

- Check the enumerated variable and use the appropriate LFO output sample.

```
bool __stdcall CModFilter::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
                                             UINT uNumInputChannels, UINT
                                             uNumOutputChannels)
{
        //
        // output = input -- change this for meaningful processing
        //
        float fYn = 0; // normal output
        float fYqn = 0; // quad phase output

        // call the LFO function; we only need first output
        m_fc_LFO.doOscillate(&fYn, &fYqn);

        // calculate both fc values (can be streamlined!)
        float fc = calculateCutoffFreq(fYn);
        float fcq = calculateCutoffFreq(fYqn);

        // get the Q LFO output
        m_Q_LFO.doOscillate(&fYn, &fYqn);

        // calculate both Q values
        float fQ = calculateQ(fYn);
        float fQq = calculateQ(fYqn);
        // use the fc and Q to calculate the Left LPF coeffs
        calculateLPFCoeffs(fc, fQ, &m_LeftLPF);
        calculateLPFCoeffs(fc, fQ, &m_RightLPF);

        // test the Phase variable; if not NORM, use the
        // quad phase LFO values on the Right channel
        if(m_uLFO_Phase == NORM)
                calculateLPFCoeffs(fc, fQ, &m_RightLPF);
        else
                calculateLPFCoeffs(fcq, fQ, &m_RightLPF);

        // do the BiQuads
        pOutputBuffer[0] = m_LeftLPF.doBiQuad(pInputBuffer[0]);

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0]; // just copy

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                pOutputBuffer[1] = m_RightLPF.doBiQuad(pInputBuffer[1]);

        return true;
}
```

Table 12.4: The quad phase control.

Button Property	Value
Control Name	LFO
Units	
Variable Type	enum
Variable Name	m_uLFO_Phase
Enum String	NORM, QUAD

Build and test the new plug-in. When you change the phase to quadrature, it will be easy to hear the two modulations chasing each other across the two speakers. Ideas for future plug-ins include:

- Change the filter type (high-pass filter, band-stop filter, band-pass filter).
- Use a double-modulator: modulate one LFO with *another* LFO, then use that to control a parameter.

12.4 Design an Envelope Follower Plug-In

While the straight-up mod filter is pretty interesting, it has a mechanical quality about it due to the LFO modulations. This might not be an issue for a synthesizer, but it does not sound musically dynamic. We can correlate the modulation to the musical input signal by implementing an envelope follower, which uses the amplitude envelope of the input signal to control the filter parameters. For example, we can code it so that when the input signal has a high amplitude, the f_c is modulated *up* to a higher value, and of course as the plug-in creators, we can also do the opposite kind of modulation. The envelope follower requires an audio envelope detector, and you can use one built into RackAFX. We will use the same second-order LPF as the filter to modulate; then you can try different filter types on your own. First, take a look at the block diagram in Figure 12.8.

The parameters are as follows:

- Pre-gain: 0 to +20 dB of pre-gain to drive the detection circuit
- Threshold (the value the signal has to cross to engage the detector)
- Envelope attack time (mSec)
- Envelope release time (mSec)
- Up/down mod control (determines whether positive amplitude excursions result in positive or negative f_c excursions)
- Digital/analog detection mode: changes detector time constant
- User-adjustable Q
- LPF is fixed as second-order bi-quad

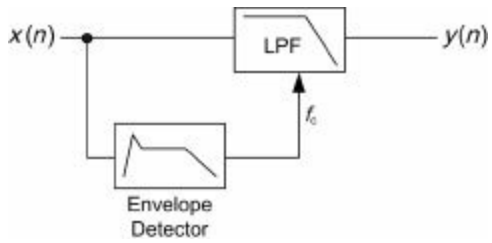


Figure 12.8: The envelope detector block diagram.

We can use the same CBiquad objects for the LFO, but we need another object to handle the envelope detection. This object is built in; the interface is in pluginconstants.h and the implementation is found in the pluginobjects.cpp file. The CEnvelopeDetector has the member variables and functions shown in Table 12.5.

This is a pretty extensive plug-in with multiple controls and interactions. Let's start with the detector itself. The CEnvelopeDetector is based on a simple detector listed in the references (musicdsp.org). The detector mimics an analog peak, mean-square, and RMS detector all in one. To use the detector, follow these steps:

1. Create an instance of the detector, for example, CEnvelopeDetector m_Detector.
2. Initialize all parameters at once with the init() function, for example, m_Detector.init((float)m_nSampleRate, 10.0, 250.0, true, DETECT_MODE_RMS, false).
3. This code initializes the detector with the following parameters:
   a. Sample rate = object's m_nSampleRate
   b. Attack time = 10 mSec
   c. Release time = 250 mSec
   d. Analog time constant = true
   e. Detect RMS signal level
   f. Linear, *not* log detection
4. To change any of the parameters on the fly, use the individual functions:
   a. m_Detector.setAttackTime()
   b. m_Detector.setReleaseTime()
   c. m_Detector.setDetectMode()
   d. m_Detector.setTCModeAnalog()

The attack and release times simply adjust these common parameters. The detection mode can be set for peak, MS, or RMS. Because audio signals must be on the range of −1.0 to +1.0 in RackAFX, this means that the squaring function of the MS and RMS modes will produce a value that is less than the instantaneous value. In RMS mode, the MS value is square-rooted, which makes the resulting envelope even smaller. Thus, you need to use care with the MS and RMS modes. In this envelope follower, we will use the RMS value to trigger our dynamic filter.

The time constant determines how fast the detector rises or falls to its maximum (1.0) and minimum (0.0) values. In an analog circuit, the resistor–capacitor (RC) rise and fall times are only 63.3% of the edge. In digital systems, the value is usually 80% to 99%. The envelope detector has a control to allow you to change the time constant from the very fast (digital) version or the slower (analog) version. We'll try both types in the envelope follower plug-in.

Table 12.5: CEnvelopeDetector object definition.

CEnvelopeDetector	
Function: Implements an Envelope Detection Unit	
Member Variables	**Purpose**
int m_nSample;	A z^{-1} delay element for the detector
float m_fAttackTime; float m_fReleaseTime;	The attack and release times for the detector in mSec
float m_fSampleRate;	The sample rate, needed for detection timing
UINT m_uDetectMode;	The detection mode according to these pre-coded enumerated values: DETECT_MODE_PEAK = 0

	DETECT_MODE_MS = 1 DETECT_MODE_RMS = 2
Member Functions	
void init(arguments) Parameters:	The initialization function, called once before prepareForPlay()
• float samplerate • float attack_in_ms • float release_in_ms • bool bAnalogTC	Input: the sample rate in Hz Input: the attack time in mSec Input: the release time in mSec Input: If true: set the attack and release time constants using the analog definition (time to rise/fall to 36.7% of min/max) — If false: set the attack and release time constants using the 1% definition (time to rise/fall to 1% of min/max); the digital version is about 4X faster
• UINT uDetect Mode	Input: uDetect mode can be one of three constants: DETECT_MODE_PEAK DETECT_MODE_MS DETECT_MODE_RMS
• bool bLogDetector	Input: true if you want the detector to be logarithmic (usually only used with LED metering)
void setTCModeAnalog(bool bAnalogTC) Parameters:	
• bool bAnalogTC	Input: sets the time constant variable; see above for more info
void setAttackTime(float attack_in_ms) Parameters: • float attack_in_ms	Change the attack time on the fly Input: the attack time in mSec.
void setReleaseTime(float release_in_ms) Parameters: • float release_in_ms	Change the release time on the fly Input: the release time in mSec
void setDetectMode(UINT uDetect) Parameters: • UINT uDetect	Change the detection mode on the fly Input: the detect mode; see m_uDetectMode above
void prepareForPlay()	The prepareForPlay() function for this object
float detect(float fInput)	The detection function; it produces the current envelope

Parameters: • float fInput	output for a given input, x(n) Input: the current input sample x(n)

12.5 Envelope Detection

Software envelope detectors are based on the analog circuits that preceded them. The two basic types of detectors are half-wave and full-wave, depending on whether they track either the positive or negative portion of the input waveform (half-wave) or both portions (fullwave). Many of the classic hardware devices used the half-wave version. Either the positive or negative portion may be tracked. The circuit consists of a diode to half-wave-rectify the signal, followed by a "tank" capacitor. The capacitor charges and discharges along with the audio signal. The attack knob is a potentiometer in series with the diode; it controls the rate at which the cap charges. The release knob is a potentiometer connected to ground through which the cap discharges. The RC combinations (one for the charging side and the other for the discharging side) create the attack and release times of the detector (Figure 12.9). The positive portion of the waveform is detected and tracks the input according to the attack and release times (Figures 12.10 and 12.11).

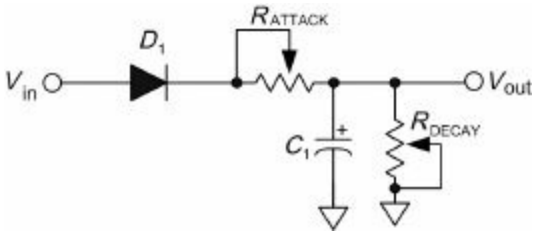


Figure 12.9: A simple positive half-wave detector.

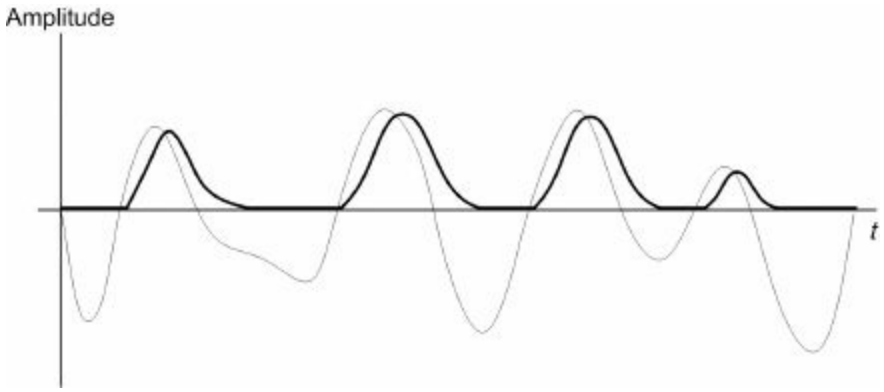


Figure 12.10: Detector output (solid dark line) with short attack and release times.

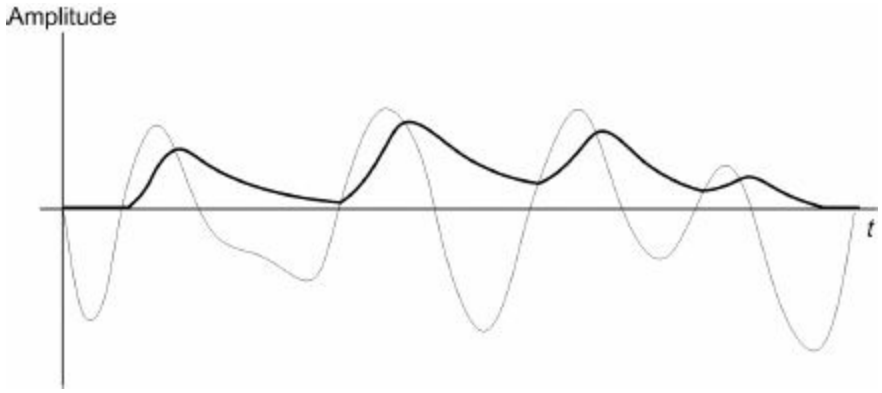


Figure 12.11: Detector output (solid dark line) with medium attack and release times.

In analog RC circuits, the cap charges and discharges exponentially, as shown in Figure 12.12. The analog attack time is calculated as the time it takes to reach 63.2% of the full charge, while the release time is the time to release from the fully charged state down to 36.8% of the full charge. Different combinations of R and C change these times. A digital envelope detector differs in the percent levels; it uses 99% and 1% as the attack and release destination values. The curves remain exponential.

It is well worth noting that there are several approaches for designing a software algorithm for the analog envelope detector. Orfanidis (1996) suggests using a full-wave rectifier followed by an LPF with separate attack and release time constants. Zöler (2011) suggests a mean-squarer followed by an LPF with only a single time constant. In either case, a running average is constantly measured and updated on each sample interval. The CEnvelopeDetector (Figure 12.13) implements a similar method using a half-wave rectifier followed by an LPF whose input is scaled by the attack or release time, depending on whether the signal is rising above the current envelope output or falling below it. You can find the original code source in the references (musicdsp.org).

The attack and release times are conveniently calculated from your controls in milliseconds. The calculations are shown in Equation 12.1:

$$\tau_a = e^{TC/(\text{attack\_in\_mSec}*\text{SampleRate}*0.001)}$$
and
$$\tau_r = e^{TC/(\text{release\_in\_mSec}*\text{SampleRate}*0.001)} \quad (12.1)$$
where
$$TC_{\text{analog}} = \log(0.368)$$
$$TC_{\text{digital}} = \log(0.01)$$

The code for the difference equation is:

```
if(fInput> m_fEnvelope)
	m_fEnvelope = m_fAttackTime * (m_fEnvelope - fInput) + fInput;
else
	m_fEnvelope = m_fReleaseTime * (m_fEnvelope - fInput) + fInput;
```

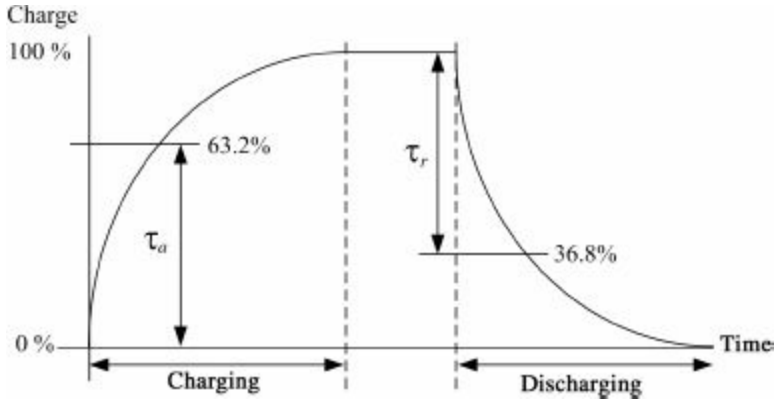


Figure 12.12: Analog attack and release times.

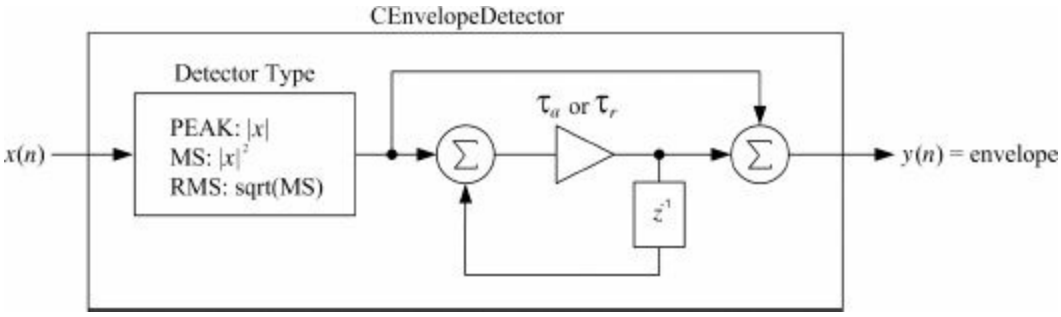


Figure 12.13: The CEnvelopeDetector block diagram.

Compare this to the block diagram in Figure 12.13 and you can see how it implements the detector.

12.5.1 Project EnvelopeFollower

Create the project; there are no member plug-ins to add.

12.5.2 EnvelopeFollower GUI

The GUI (Table 12.6) needs to have controls for the envelope detector:

- Pre-gain
- Threshold
- Attack time
- Release time
- Time constant (analog or digital)

Table 12.6: The GUI controls for the envelope follower plug-in.

Slider Property	Value
Control Name	Pre-gain
Units	dB
Variable Type	float
Variable Name	m_fPreGain_dB
Low Limit	0
High Limit	20
Initial Value	12

Slider Property	Value
Control Name	Attack Time
Units	mSec
Variable Type	float

Variable Name	m_fAttack_mSec
Low Limit	10
High Limit	100
Initial Value	25

Slider Property	Value
Control Name	Q
Units	
Variable Type	float
Variable Name	m_fQ
Low Limit	0.5
High Limit	20
Initial Value	5

Slider Property	Value
Control Name	Threshold
Units	
Variable Type	float
Variable Name	m_fThreshold
Low Limit	0
High Limit	1
Initial Value	0.2

Slider Property	Value
Control Name	Release Time
Units	mSec
Variable Type	float
Variable Name	m_fRelease_mSec
Low Limit	20
High Limit	250
Initial Value	50

Button Property	Value
Control Name	Time Constant
Units	

Variable Type	enum
Variable Name	m_fTimeConstant
Enum String	analog, digital

Button Property	Value
Control Name	Direction
Units	
Variable Type	enum
Variable Name	m_fDirection
Enum String	UP, DOWN

And, it needs controls for the modulated filter:

- *Q*
- Direction of modulation (up or down)

12.5.3 EnvelopeFollower.h File

We need to add the following new functions and variables; most can be taken directly from your ModFilter project. We will need:

- An LPF bi-quad for each channel (left, right)
- An envelope detector for each channel (left, right)
- A function to calculate a new LPF cutoff value from an envelope value
- Functions to calculate the bi-quad coefficients from the LPF cutoff and *Q* values
- Min/max LPF cutoff variables

Add the following to your .h file:

```
// Add your code here: ------------------------------------------------------ //
// two LPF BiQuads for the filters
CBiquad m_LeftLPF;
CBiquad m_RightLPF;

// function to calculate the new fc from the Envelope Value
float calculateCutoffFreq(float fEnvelopeSample);

// calculate the coeffs for a given filter
void calculateLPFCoeffs(float fCutoffFreq, float fQ, CBiquad* pFilter);

// min/max variables
float m_fMinCutoffFreq;
float m_fMaxCutoffFreq;

// envelope detectors
CEnvelopeDetector m_LeftDetector;
CEnvelopeDetector m_RightDetector;

// END OF USER CODE ---------------------------------------------------------- //
```

Notice that everything except the envelope detectors is straight from your last project.

12.5.4 EnvelopeFollower.cpp File

Constructor

- Initialize the rate min and max values. The member-objects are self initializing upon creation.

```
CEnvelopeFollower::CEnvelopeFollower()
{
        <SNIP SNIP SNIP>

        // Finish initializations here
        // set our Min and Max Modulation points
        m_fMinCutoffFreq = 100.0;
        m_fMaxCutoffFreq = 5000.0;
}
```

Write the calculateCutoffFrequency() function. It is similar to the previous modulation functions except that we will allow both up and down directions of modulation. Can you figure out how the code works?

```
float CEnvelopeFollower::calculateCutoffFreq(float fEnvelopeSample)
{
        // modulate from min upwards
        if(m_uDirection == UP)
                return fEnvelopeSample*(m_fMaxCutoffFreq - m_fMinCutoffFreq) +
                        m_fMinCutoffFreq;
        else // modulate from max downwards
                return m_fMaxCutoffFreq - fEnvelopeSample*(m_fMaxCutoffFreq -
                        m_fMinCutoffFreq);

        return m_fMinCutoffFreq;
}
```

Write the cooking functions for the LPFs to set the new coefficients for a given f_c and Q value. This is identical to the ModFilter project.

```
void CEnvelopeFollower::calculateLPFCoeffs(float fCutoffFreq, float fQ, CBiquad*
            pFilter)
{
        /* use same code as the Mod Filter */
}
```

prepareForPlay()

- Flush delays in LPFs.
- Initialize the detector objects.

```
bool __stdcall CEnvelopeFollower::prepareForPlay()
{
	// Add your code here:
	m_LeftLPF.flushDelays();
	m_RightLPF.flushDelays();

	// init the envelope detectors
	// set all params at once with this function;
	// false - Digital Time Constant NOT Analog one
	if(m_uTimeConstant == digital)
	{
		m_LeftDetector.init((float)m_nSampleRate, m_fAttack_mSec, m_fRelease_mSec,
			false, DETECT_MODE_RMS, false);
		m_RightDetector.init((float)m_nSampleRate, m_fAttack_mSec,
			m_fRelease_mSec, false, DETECT_MODE_RMS, false);
	}
	else
	{
		m_LeftDetector.init((float)m_nSampleRate, m_fAttack_mSec, m_fRelease_mSec,
			true, DETECT_MODE_RMS, false);
		m_RightDetector.init((float)m_nSampleRate, m_fAttack_mSec,
			m_fRelease_mSec, true, DETECT_MODE_RMS, false);
	}

	return true;
}
```

You can see that the detectors are initialized all at once, using the m_uTimeConstant to control the boolean value passed to the init function. The detection is set as MS (DETECT_ MODE_MS).

userInterfaceChange()

- Decode the control ID and then call the appropriate function; we simply pass our slider values or radio button command to the functions.
- Note: Make sure your nControlIndex values match with your UI; these mappings are for my project.

```
bool __stdcall CEnvelopeFollower::userInterfaceChange(int nControlIndex)
{
	switch(nControlIndex)
	{
		case 2: // Attack
		{
			m_LeftDetector.setAttackTime(m_fAttack_mSec);
			m_RightDetector.setAttackTime(m_fAttack_mSec);
			break;
		}
		case 3: // Release
		 {
			m_LeftDetector.setReleaseTime(m_fRelease_mSec);
			m_RightDetector.setReleaseTime(m_fRelease_mSec);
			break;
		}
```

```
            case 41: // Time Constant
            {
                    if(m_uTimeConstant == digital)
                    {
                            m_LeftDetector.setTCModeAnalog(false);
                            m_RightDetector.setTCModeAnalog(false);
                    }
                    else
                    {
                           m_LeftDetector.setTCModeAnalog(true);
                           m_RightDetector.setTCModeAnalog(true);
                    }
            }

            default:
                    break;
        }

        return true;
}
```

processAudioFrame()

- The envelope follower's processAudioFrame() function will operate in the sequence shown in Figure 12.14.

```
bool __stdcall CEnvelopeFollower::processAudioFrame(float* pInputBuffer, float*
                                                    pOutputBuffer, UINT
                                                    uNumInputChannels, UINT
                                                    uNumOutputChannels)
{
        //
        // Do LEFT (MONO) Channel; there is always at least one input, one output
        float fGain = pow(10, m_fPreGain_dB/20.0);
        float fDetectLeft = m_LeftDetector.detect(fGain*pInputBuffer[0]);

        // set mod freq to minimum (un-triggered)
        float fModFreqLeft =m_fMinCutoffFreq;

        // if threshold triggered, calculate new LPF cutoff
        if(fDetectLeft >= m_fThreshold)
                fModFreqLeft = calculateCutoffFreq(fDetectLeft);

        // use the mod freq and user supplied-Q to calc the fc
        calculateLPFCoeffs(fModFreqLeft, m_fQ, &m_LeftLPF);
```

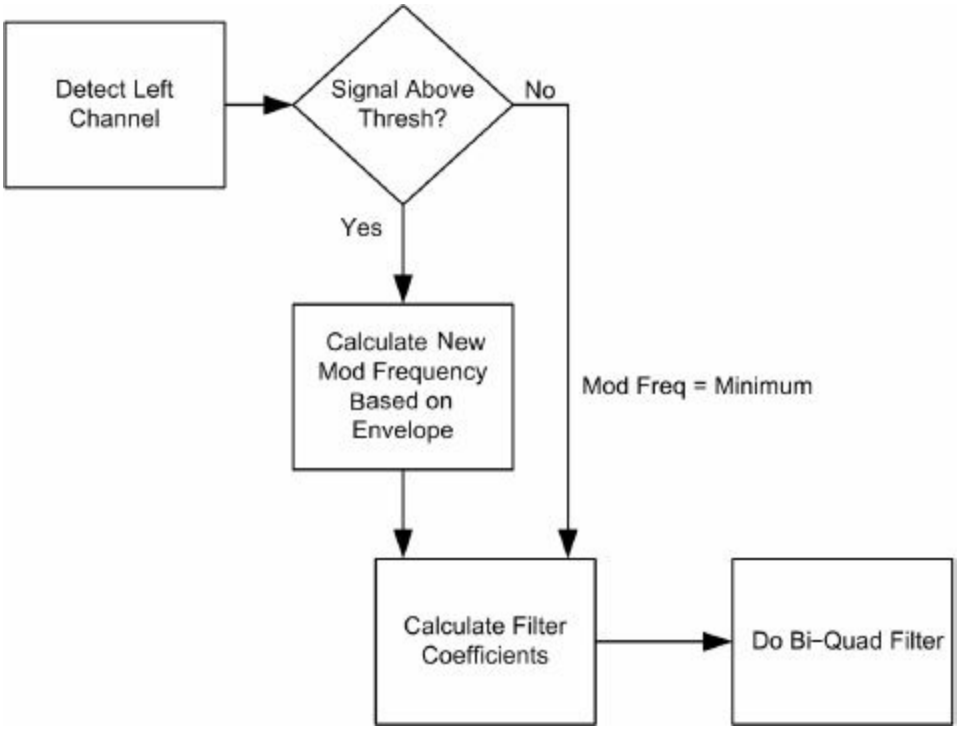


Figure 12.14: The flowchart for the left channel of the envelope follower (right channel is same).

```
        // do the BiQuads
        pOutputBuffer[0] = m_LeftLPF.doBiQuad(pInputBuffer[0]);

        // detect the other channel
        float fDetectRight = m_RightDetector.detect(fGain*pInputBuffer[1]);

        // set mod freq to minimum (un-triggered)
        float fModFreqRight = m_fMinCutoffFreq;

        // if threshold triggered, calculate new LPF cutoff
        if(fDetectLeft >= m_fThreshold)
                fModFreqRight = calculateCutoffFreq(fDetectRight);

        // use the mod freq and user supplied-Q to calc the fc
        calculateLPFCoeffs(fModFreqRight, m_fQ, &m_RightLPF);

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer [0];

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                        pOutputBuffer[1] = m_RightLPF.doBiQuad(pInputBuffer[1]);

        return true;
}
```

Build and test the plug-in. Try all the controls. Note: Beware—because of the high pre-gain values possible combined with high-Q values, you can easily overload the audio. If you hear distortion, adjust the pre-gain value. Test this out with guitar, keyboard, or drum loops. Ideas for more plug-ins include:

- Change the filter type (high-pass filter, band-stop filter, band-pass filter).
- Modulate the Q in the same direction, or opposite direction, as the f_c.

- Modulate the f_c or Q with an LFO and the other parameter with the envelope.
- Allow the user to select the detection mode: peak, MS, or RMS.
- Allow the user to change between linear and log detection (see CEnvelopeDetector's init() method).

12.6 Design a Phaser Plug-In

The phaser effect will be on par with the reverb algorithms as among the more complex in the book—the theory is not that difficult, but a nice analog sounding phaser requires many nearly duplicated components, in this case APFs. The phaser effect was an attempt at creating a flanging effect before the necessary analog integrated circuits were available. A flanger uses a modulated delay line to alter the delay time. A phaser uses APFs in an attempt to do the same. APFs have a delay response that is flat across the first one-third or so of the DC-to-Nyquist band. In fact, APFs can be designed to perform fractional delay as an alternative to interpolation. The phaser didn't sound like a flanger at all. The reason is that in a flanger, the notches in the inverse-comb filtering are mathematically related as simple multiples of each other. This is not the case for the phaser. The phaser is still used today in electronic music. Like the flanger, we will get the best results by first looking at the original analog design. Our phaser topology will be based off the analog counterpart in Figures 12.15 and 12.16. Even if you can't read schematics, it's worth a moment to take a look at the overall architecture since we'll be emulating it in digital form. The complete block diagram is shown in Figure 12.17.

The phase of the signal is shifted by 90 degrees at the frequency given by Equation 12.2:

$$f_p = \frac{1}{2\pi C_1(R_1 // r_{DS})} \tag{12.2}$$

The positive control voltage is a triangle wave, which alters the resistance of the field effect transistor (FET) that is in parallel with R_1. This in turn changes the 90 degree phase shift frequency. The values of C_1 vary from 1 uF to 0.06 uF, and r_{DS} varies from 100 Ω (FET on) to 10 K (FET off). The input signal is run through a cascade of such modules. The output of the last module, or stage, is then mixed with the dry input signal to create the effect. You only need to decide on the number of stages and the oscillator to complete the design.

The things to note about the analog design are:

- There are six first-order APF modules.
- The APF modules are in series, one feeding the next.
- The depth control is a blend of the dry and filtered signals; a depth of 100% is actually a 50/50 mix of wet and dry.

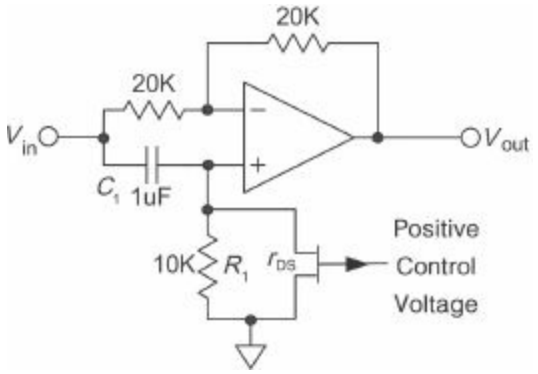


Figure 12.15: A voltage-controlled first-order all-pass filter module.

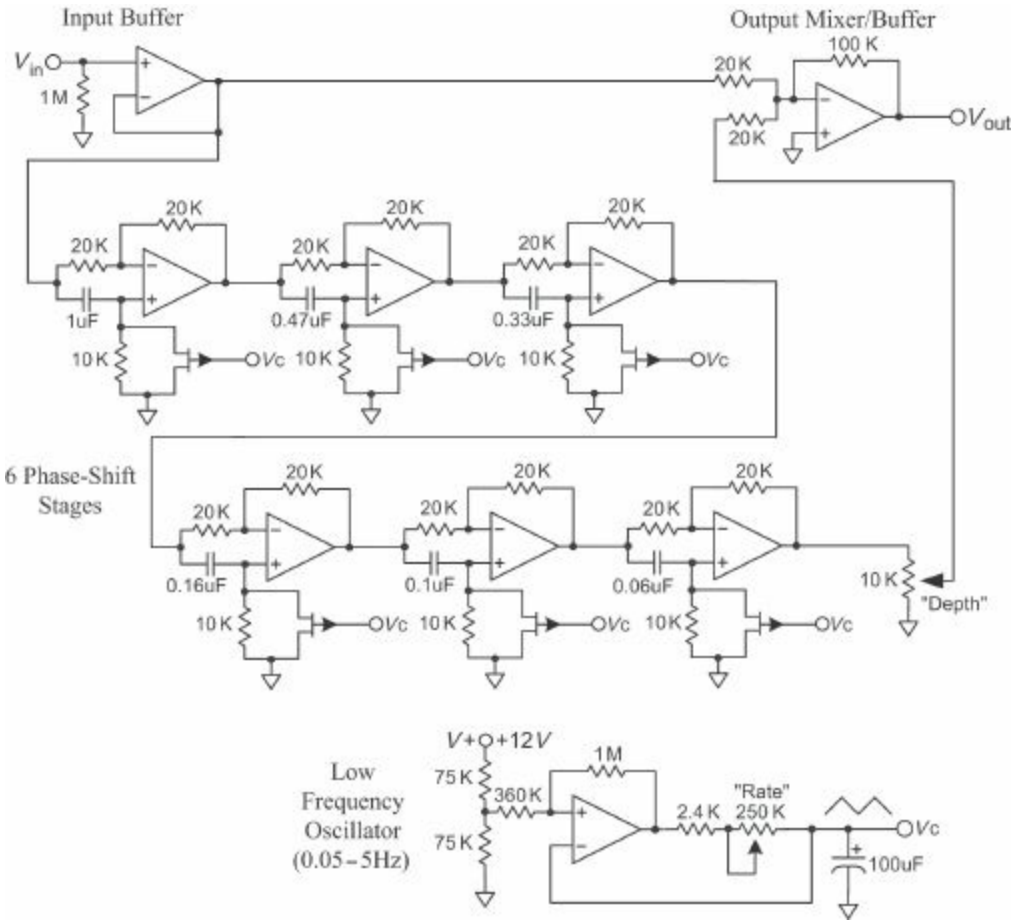


Figure 12.16: The phaser circuit (from National Semiconductor).

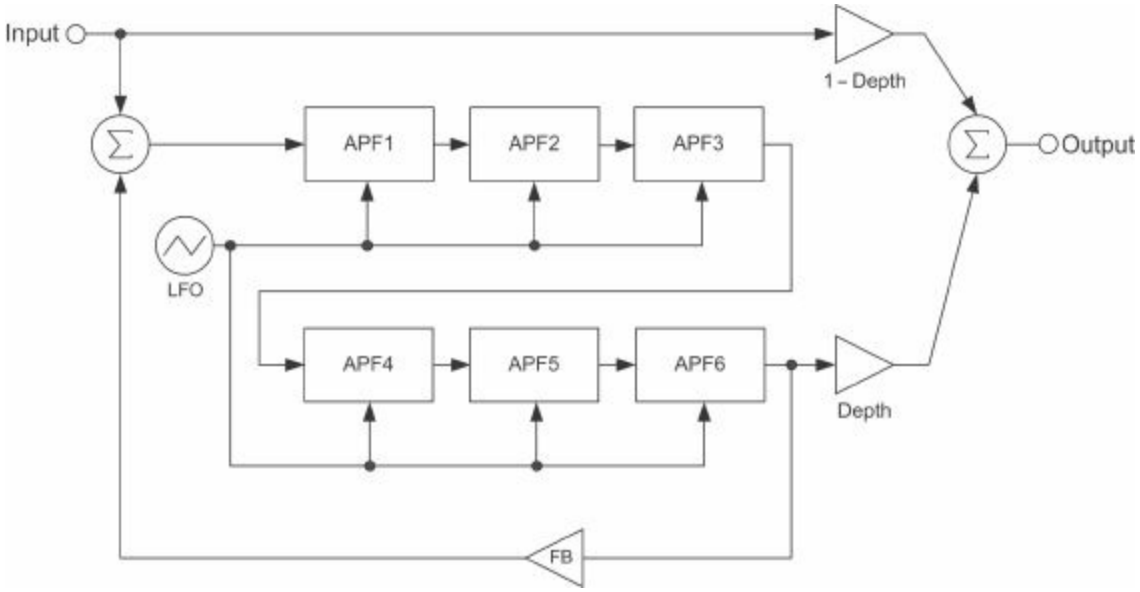


Figure 12.17: The block diagram of the analog phaser design. The feedback path has been added to the original National Semiconductor design; the feedback will be named "Intensity".

- Other phaser models also included a feedback path from the filtered output back to the filter chain input to increase intensity of the effect.
- The six APF modules have the frequency ranges in Table 12.7.

Because this is a complex design, we will break it down into two parts: first, we'll design an abbreviated version in Figure 12.18 with only three APF modules and no feedback control. Then, we'll add the other three APF stages and the feedback control.

The design equations for first-order APFs from Chapter 6 are given next.

Table 12.7: Minimum and maximum phase rotation frequencies for the phaser APFs.

APF	Minimum f_c	Maximum f_c
1	16 Hz	1.6 kHz
2	33 Hz	3.3 kHz
3	48 Hz	4.8 kHz
4	98 Hz	9.8 kHz
5	160 Hz	16 kHz
6	260 Hz	26 kHz*

*We will have to cut off at ½ Nyquist instead.

Specify:

- f_c, corner frequency

$$
\begin{aligned}
\theta_c &= 2\pi f_c / f_s \\
\alpha &= \frac{\tan(\pi Q / f_s) - 1}{\tan(\pi Q / f_s) + 1} \\
\beta &= -\cos\theta_c \\
a_0 &= -\alpha \\
a_1 &= \beta(1-\alpha) \\
a_2 &= 1.0 \\
b_1 &= \beta(1-\alpha) \\
b_2 &= -1.0
\end{aligned}
\tag{12.3}
$$

We'll need to implement the APF using the built-in BiQuad object. The built-in oscillator will be used for the LFO; many aspects of the design and code chunks can be borrowed from your ModFilter projects. Specific to this design is that it:

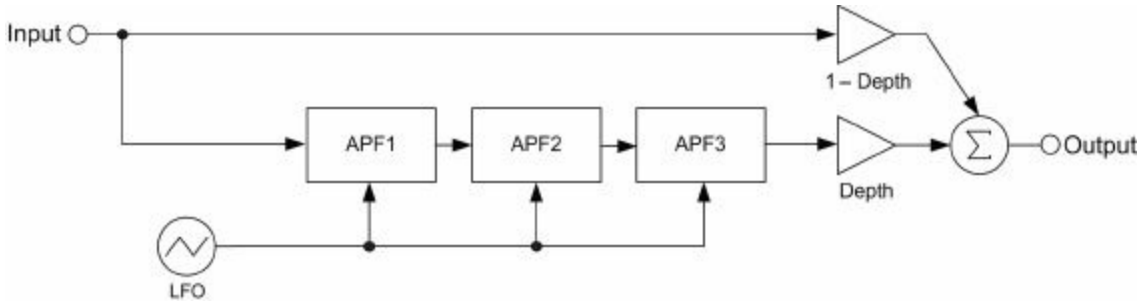


Figure 12.18: The block diagram of the simplified, first part of the design.

- Needs multiple APFs, each with its own range of frequencies
- Needs only one LFO to control all APFs
- Needs to calculate all APF cutoff frequencies and coefficients at the same time

Let's get started with the usual steps.

12.6.1 Project Phaser

Create a new project. There are no member plug-ins to add.

12.6.2 Phaser GUI

Add the controls from Table 12.8 to your GUI.

Table 12.8: The phaser GUI elements.

Slider Property	Value
Control Name	Rate
Units	Hz
Variable Type	float

Variable Name	m_fModRate
Low Limit	0.02
High Limit	10
Initial Value	0.5

Slider Property	Value
Control Name	Depth
Units	%
Variable Type	float
Variable Name	m_fMod_Depth
Low Limit	0
High Limit	100
Initial Value	50

Button Property	Value
Control Name	LFO Type
Units	
Variable Type	enum
Variable Name	m_fuLFO_Waveform
Enum String	sine, saw, tri, square

12.6.3 Phaser.h File

Declare instances of the BiQuad and WaveTable objects. Also, declare the functions to calculate cutoff frequencies and filter coefficients. We will calculate all the APF coefficients in one pair of functions (one for left and one for right). Because there is so much repeated code, we will also add a helper function to calculate the APF coefficients for each bi-quad. You don't need to #include anything since these are built in. Add these to your .h files

```
// Add your code here: ------------------------------------------------------ //
CBiquad m_LeftAPF_1;
CBiquad m_RightAPF_1;

CBiquad m_LeftAPF_2;
CBiquad m_RightAPF_2;

CBiquad m_LeftAPF_3;
CBiquad m_RightAPF_3;

// function to calculate the new fc from the Envelope Value
float calculateAPFCutoffFreq(float fLFOSample, float fMinFreq, float fMaxFreq);

// Two functions to calculate the BiQuad Coeffs: APF
void calculateFirstOrderLeftAPFCoeffs(float fLFOSample);
void calculateFirstOrderRightAPFCoeffs(float fLFOSample);

// helper function for APF Calculation
void calculateFirstOrderAPFCoeffs(float fCutoffFreq,
                                  CBiquad* pBiQuadFilter);

// min/max variables
float m_fMinAPF_1_Freq;
float m_fMaxAPF_1_Freq;

float m_fMinAPF_2_Freq;
float m_fMaxAPF_2_Freq;

float m_fMinAPF_3_Freq;
float m_fMaxAPF_3_Freq;

// LFO Stuff
CWaveTable m_LFO;

// END OF USER CODE --------------------------------------------------------- //
```

12.6.4 Phaser.cpp File

Constructor

- Initialize the rate min and max values. The objects are self-initializing upon creation.
- Initialize the first three APF min/max pairs according to the table. We'll add the others once we have tested it and find it works properly.

```
CPhaser::CPhaser()
{
        <SNIP SNIP SNIP>

        // Finish initializations here
        // set our Min and Max Modulation points
        m_fMinAPF_1_Freq = 16.0;
        m_fMaxAPF_1_Freq = 1600.0;

        m_fMinAPF_2_Freq = 33.0;
        m_fMaxAPF_2_Freq = 3300.0;

        m_fMinAPF_3_Freq = 48.0;
        m_fMaxAPF_3_Freq = 4800.0;
}
```

Write the calculateAPFCutoffFreq() function. This function behaves exactly like the function in the

mod filter except there is no mod depth. Also note that we will call this function repeatedly for each APF unit, so we need to send it the min/max values to compute with; the function itself is trivial and just like all the previous mod filters:

```
float CPhaser::calculateAPFCutoffFreq(float fLFOSample, float fMinFreq, float fMaxFreq)
{
      return fLFOSample*(fMaxFreq - fMinFreq) + fMinFreq;
}
```

Implement the helper function for calculating the APF coefficients for a given f_c.

```
void CPhaser::calculateFirstOrderAPFCoeffs(float fCutoffFreq,
                                           CBiquad* pBiQuadFilter)
{
       // coeff calculation
       float alpha_num = tan(pi*fCutoffFreq/(float)m_nSampleRate) - 1.0;
       float alpha_den = tan(pi*fCutoffFreq/(float)m_nSampleRate) + 1.0;
       float alpha = alpha_num/alpha_den;

       // set on target filter
       pBiQuadFilter->m_f_a0 = alpha;
       pBiQuadFilter->m_f_a1 = 1.0;
       pBiQuadFilter->m_f_a2 = 0.0;
       pBiQuadFilter->m_f_b1 = alpha;
       pBiQuadFilter->m_f_b2 = 0.0;
}
```

Implement the cooking functions below for the right and left banks of APFs using the helper function above:

```
void CPhaser::calculateFirstOrderLeftAPFCoeffs(float fLFOSample)
{
       // APF1 - fc -> Bi Quad
       float fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_1_Freq,
                                                  m_fMaxAPF_1_Freq);
       calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_LeftAPF_1);

       // APF2 - fc -> Bi Quad
       fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_2_Freq,
                                                  m_fMaxAPF_2_Freq);
       calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_LeftAPF_2);

       // APF3 - fc -> Bi Quad
       fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_3_Freq,
                                                  m_fMaxAPF_3_Freq);
       calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_LeftAPF_3);
}
```

```
void CPhaser::calculateFirstOrderRightAPFCoeffs(float fLFOSample)
{
	// APF1 - fc -> Bi Quad
	float fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_1_Freq,
	                                           m_fMaxAPF_1_Freq);
	calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_RightAPF_1);

	// APF2 - fc -> Bi Quad
	fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_2_Freq,
	                                           m_fMaxAPF_2_Freq);
	calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_RightAPF_2);

	// APF3 - fc -> Bi Quad
	fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_3_Freq,
	                                           m_fMaxAPF_3_Freq);
	calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_RightAPF_3);
}
```

prepareForPlay()

- Flush the delays in the APFs.
- Set up the LFO by setting the initial rate, LFO type (the LFO type will match our m_uLFOWaveform variable).

```
bool __stdcall CPhaser::prepareForPlay()
{
	// Add your code here:
	m_LeftAPF_1.flushDelays();
	m_RightAPF_1.flushDelays();

	m_LeftAPF_2.flushDelays();
	m_RightAPF_2.flushDelays();

	m_LeftAPF_3.flushDelays();
	m_RightAPF_3.flushDelays();

	// setup LFO
	m_LFO.m_fFrequency_Hz = m_fModRate;
	m_LFO.m_uPolarity = 1; // 0 = bipolar, 1 = unipolar
	m_LFO.m_uTableMode = 0; // normal, no band limiting
	m_LFO.m_uOscType = m_uLFO_Waveform;
	m_LFO.setSampleRate(m_nSampleRate); // really important!

	// the LFO prepareForPlay() calls reset() and cookFrequency()
	m_LFO.prepareForPlay();

	return true;
}
```

userInterfaceChange()

- If the user moves the rate slider, we need to change the LFO rate and toggle the oscillator type if they use the buttons.
- Note: Make sure your nControlIndex values match your UI.

```
bool __stdcall CPhaser::userInterfaceChange(int nControlIndex)
{
	// decode the control index, or delete the switch and use brute force calls
	switch(nControlIndex)
	{
		case 0:
		{
			m_LFO.m_fFrequency_Hz = m_fModRate;
			m_LFO.cookFrequency();
			break;
		}
		case 41:
		{
			m_LFO.m_uOscType = m_uLFO_Waveform;
			break;
		}
		default:
			break;
	}

	return true;
}
```

processAudioFrame()

- Calculate a new LFO value.
- Cascade our three APFs one after another, feeding the output of each into the input of the next until done.
- Use the depth control to mix the output.
- Calculate the depth so that 100% depth gives a 50/50 mix ratio (done by dividing depth percent by 200.0 instead of 100).

```
bool __stdcall CPhaser::processAudioFrame(float* pInputBuffer, float* pOutputBuffer, UINT
                                          uNumInputChannels, UINT uNumOutputChannels)
{
	// Do LEFT (MONO) Channel; there is always at least one input, one output
	float fYn = 0; // normal output
	float fYqn = 0; // quad phase output

	// mod depth is at 100% when the control is at 50% !!
	float fDepth = m_fMod_Depth/200.0;

	// call the LFO function; we only need first output
	m_LFO.doOscillate(&fYn, &fYqn);

	// use the LFO to calculate all APF banks
	calculateFirstOrderLeftAPFCoeffs(fYn);

	// do the cascaded APFs
	float fAPF_1_Out = m_LeftAPF_1.doBiQuad(pInputBuffer[0]);
	float fAPF_2_Out = m_LeftAPF_2.doBiQuad(fAPF_1_Out);
	float fAPF_3_Out = m_LeftAPF_3.doBiQuad(fAPF_2_Out);
```

```
        // form the output
        pOutputBuffer[0] = fDepth*fAPF_3_Out + (1.0 - fDepth)*pInputBuffer[0];

        // calculate
        calculateFirstOrderRightAPFCoeffs(fYn);

        // do the cascade
        fAPF_1_Out = m_RightAPF_1.doBiQuad(pInputBuffer[1]);
        fAPF_2_Out = m_RightAPF_2.doBiQuad(fAPF_1_Out);
        fAPF_3_Out = m_RightAPF_3.doBiQuad(fAPF_2_Out);

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0];

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                pOutputBuffer[1] = fDepth*fAPF_3_Out + (1.0 - fDepth)*pInputBuffer[1];

        return true;
}
```

Build and test the plug-in. You should be able to easily hear the phaser effect, though it might not be as strong as you are used to with other devices. We're going to fix this next by added more APF stages, feedback, and the option for quadrature phase LFO operation.

12.7 Design a Stereo Phaser with Quad-Phase LFOs

Adding the extra APFs is just more of the same cut-and-paste operation; you only need to change the range of frequencies and adjust the cooking and processing functions for that. The addition of the feedback is simple—we need another slider on the UI and a feedback z^{-1} storage device. The quadrature phase is easy to implement because our LFO already produces this output for us; we only need a UI change and a branch statement in the code.

12.7.1 Phaser GUI

Update the GUI with the new controls in Table 12.9.

Table 12.9: Additional controls for the quad-phase phaser plug-in.

Button Property	Value
Control Name	LFO Type
Units	
Variable Type	enum
Variable Name	m_fuLFO_Phase
Enum String	NORM, QUAD

Slider Property	Value

Control Name	Intensity
Units	%
Variable Type	float
Variable Name	m_fFeedback
Low Limit	0
High Limit	99
Initial Value	50

12.7.2 Phaser.h File

We need to add three more APF object sets (three for left, three for right) plus the feedback storage elements, one each for left and right. Add the following to your .h file:

```
// Add your code here: -----------------------------------------------------

<SNIP SNIP SNIP>

CBiquad m_LeftAPF_3;
CBiquad m_RightAPF_3;

CBiquad m_LeftAPF_4;
CBiquad m_RightAPF_4;

CBiquad m_LeftAPF_5;
CBiquad m_RightAPF_5;

CBiquad m_LeftAPF_6;
CBiquad m_RightAPF_6;

// function to calculate the new fc from the Envelope Value
float calculateAPFCutoffFreq(float fLFOSample, float fMinFreq, float fMaxFreq);

<SNIP SNIP SNIP>

float m_fMinAPF_3_Freq;
float m_fMaxAPF_3_Freq;

float m_fMinAPF_4_Freq;
float m_fMaxAPF_4_Freq;

float m_fMinAPF_5_Freq;
float m_fMaxAPF_5_Freq;

float m_fMinAPF_6_Freq;
float m_fMaxAPF_6_Freq;

// LFO Stuff
CWaveTable m_LFO;

// Feedback Storage
float m_fFeedbackLeft;
float m_fFeedbackRight;

// END OF USER CODE ---------------------------------------------------------
```

12.7.3 Phaser.cpp File

Constructor

- Initialize the min and max APF values as per the table.

```
CPhaser::CPhaser()
{
        <SNIP SNIP SNIP>
        m_fMinAPF_3_Freq = 48.0;
        m_fMaxAPF_3_Freq = 4800.0;

        m_fMinAPF_4_Freq = 98.0;
        m_fMaxAPF_4_Freq = 9800.0;

        m_fMinAPF_5_Freq = 160.0;
        m_fMaxAPF_5_Freq = 16000.0;

        m_fMinAPF_6_Freq = 220.0;
        m_fMaxAPF_6_Freq = 22000.0;

        m_fFeedbackLeft = 0.0;
        m_fFeedbackRight = 0.0;
}
```

Modify the cooking functions to calculate the new APF coefficients:

```
void CPhaser::calculateFirstOrderLeftAPFCoeffs(float fLFOSample)
{
        <SNIP SNIP SNIP>

        // APF3 – fc -> Bi Quad
        fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_3_Freq,
                                             m_fMaxAPF_3_Freq);
        calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_LeftAPF_3);

        // APF4 – fc -> Bi Quad
        fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_4_Freq,
                                             m_fMaxAPF_4_Freq);
        calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_LeftAPF_4);

        // APF5 – fc -> Bi Quad
        fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_5_Freq,
                                             m_fMaxAPF_5_Freq);
        calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_LeftAPF_5);

        // APF6 - fc -> Bi Quad
        fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_6_Freq,
                                             m_fMaxAPF_6_Freq);
        calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_LeftAPF_6);

}
```

```
void CPhaser::calculateFirstOrderRightAPFCoeffs(float fLFOSample)
{
	<SNIP SNIP SNIP>

	// APF3 - fc -> Bi Quad
	fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_3_Freq,
	                                     m_fMaxAPF_3_Freq);
	calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_RightAPF_3);

	// APF4 - fc -> Bi Quad
	fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_4_Freq,
	                                     m_fMaxAPF_4_Freq);
	calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_RightAPF_4);

	// APF5 - fc -> Bi Quad
	fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_5_Freq,
	                                     m_fMaxAPF_5_Freq);
	calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_RightAPF_5);

	// APF6 - fc -> Bi Quad
	fCutoffFreq = calculateAPFCutoffFreq(fLFOSample, m_fMinAPF_6_Freq,
	                                     m_fMaxAPF_6_Freq);
	calculateFirstOrderAPFCoeffs(fCutoffFreq, &m_RightAPF_6);
}
```

prepareForPlay()

- Flush the new APF buffers.
- Reset the feedback variables.

```
bool __stdcall CPhaser::prepareForPlay()
{
	// Add your code here:
	<SNIP SNIP SNIP>

	m_LeftAPF_3.flushDelays();
	m_RightAPF_3.flushDelays();

	m_LeftAPF_4.flushDelays();
	m_RightAPF_4.flushDelays();

	m_LeftAPF_5.flushDelays();
	m_RightAPF_5.flushDelays();

	m_LeftAPF_6.flushDelays();
	m_RightAPF_6.flushDelays();

	m_LFO.m_fFrequency_Hz = m_fModRate;
	m_LFO.m_uPolarity = 1; // 0 = bipolar, 1 = unipolar
	m_LFO.m_uTableMode = 0; // normal, no band limiting
	m_LFO.m_uOscType = m_uLFO_Waveform;
	m_LFO.setSampleRate(m_nSampleRate); // really important!

	// the LFO prepareForPlay() calls reset() and cookFrequency()
	m_LFO.prepareForPlay();

	m_fFeedbackLeft = 0.0;
	m_fFeedbackRight = 0.0;

	return true;
}
```

processAudioFrame()

- Add more cascaded APFs to each channel.
- Store and use the feedback values.
- Check for quad-phase LFO and use the quad LFO output on right channel if needed.

```
bool __stdcall CPhaser::processAudioFrame(float* pInputBuffer, float* pOutputBuffer, UINT
                                          uNumInputChannels, UINT uNumOutputChannels)
{
        //
        // output = input -- change this for meaningful processing
        //
        // Do LEFT (MONO) Channel; there is always at least one input, one output


        float fYn = 0; // normal output
        float fYqn = 0; // quad phase output

        // mod depth is at 100% when the control is at 50% !!
        float fDepth = m_fMod_Depth/200.0;
        float fFeedback = m_fFeedback/100.0;

        // call the LFO function; we only need first output
        m_LFO.doOscillate(&fYn, &fYqn);

        // use the LFO to calculate all APF banks
        calculateFirstOrderLeftAPFCoeffs(fYn);

        // do the cascaded APFs
        float fAPF_1_Out = m_LeftAPF_1.doBiQuad(pInputBuffer[0] +
                                                m_fFeedbackLeft*fFeedback);

        float fAPF_2_Out = m_LeftAPF_2.doBiQuad(fAPF_1_Out);
        float fAPF_3_Out = m_LeftAPF_3.doBiQuad(fAPF_2_Out);
        float fAPF_4_Out = m_LeftAPF_4.doBiQuad(fAPF_3_Out);
        float fAPF_5_Out = m_LeftAPF_5.doBiQuad(fAPF_4_Out);
        float fAPF_6_Out = m_LeftAPF_6.doBiQuad(fAPF_5_Out);


        // for next sample period
        m_fFeedbackLeft = fAPF_6_Out;

        pOutputBuffer[0] = fDepth*fAPF_6_Out + (1.0 - fDepth)*pInputBuffer[0];

        // use the fc and Q to calculate the Left LPF coeffs
        if(m_uLFO_Phase == QUAD)
                calculateFirstOrderRightAPFCoeffs(fYqn);
        else
                calculateFirstOrderRightAPFCoeffs(fYn);

        // do the cascaded APFs
        fAPF_1_Out = m_RightAPF_1.doBiQuad(pInputBuffer[1] +
                                                m_fFeedbackRight*fFeedback);
        fAPF_2_Out = m_RightAPF_2.doBiQuad(fAPF_1_Out);
        fAPF_3_Out = m_RightAPF_3.doBiQuad(fAPF_2_Out);
        fAPF_4_Out = m_RightAPF_4.doBiQuad(fAPF_3_Out);
        fAPF_5_Out = m_RightAPF_5.doBiQuad(fAPF_4_Out);
        fAPF_6_Out = m_RightAPF_6.doBiQuad(fAPF_5_Out);

        // for next sample period
        m_fFeedbackRight = fAPF_6_Out;

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0];
```

```
        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                pOutputBuffer[1] = fDepth*fAPF_6_Out + (1.0 - fDepth)
                *pInputBuffer[1];
        return true;
}
```

Build and test the plug-in. Try out the different modes; turn up the intensity and depth controls for very intense effects. This phaser should be one of the finest you have heard as it is not like some digital phasers that only use one giant APF. The cascaded APFs are the way to go with this effect. The saw LFO with copious feedback sounds very much like a classic pulsating "vibe" effect.

Bibliography

Anderton, C. 1981. *Electronic Projects for Musicians*. London: Wise Publications.

Dodge, C. and Jerse, T. 1997. *Computer Music Synthesis, Composition and Performance*. Chapter 2. New York: Schirmer.

Giles, M., ed. 1980. *The Audio/Radio Handbook*. Santa Clara, CA: National Semiconductor Corp.

Roads, C. 1996. *The Computer Music Tutorial*. Chapter 2. Cambridge, MA: The MIT Press.

References

musicdsp.org. "Envelope follower with different attack and release." Accessed 2008, http://www.musicdsp.org/archive.php?classid=0#205.

Orfanidis, S. 1996. *Introduction to Signal Processing*, Chapter 8. Englewood Cliffs, NJ: Prentice-Hall.

Zöler, U. 2011. *DAFX—Digital Audio Effects*, Chapter 4. West Sussex, U.K.: John Wiley & Sons.

CHAPTER 13

Dynamics Processing

Dynamics processors are designed to automatically control the amplitude, or gain, of an audio signal and consist of two families: compressors and expanders. Technically speaking, compressors and expanders both change the gain of a signal after its level rises *above* a predetermined threshold value. A compressor reduces the gain of the signal once it goes over the threshold. An expander raises the gain of a signal after it crosses above the threshold. With the exception of noise reduction systems, true expanders are rare since they can easily cause instabilities, runaway gain, and distortion. What is normally called an "expander" today is technically a *downward expander*. A downward expander reduces the gain of a signal after it drops *below* the threshold. We will be designing a downward expander but will use the common lingo and refer to it simply as an expander. Both compressors and expanders require the user to decide on the threshold of operation as well as a *ratio* value that tells the device how much gain reduction to implement.

Figure 13.1 shows the input/output transfer functions for the compressor family. The line marked 1:1 is unity gain. A given input x(dB) results in the same output y (dB). Above the threshold, the output level is compressed according to the ratio. For example, above the threshold on the 2:1 line, every increase in 1 dB at the input results in an increase of only 0.5 dB at the output. On the 4:1 line, each 1 dB input increase results in a 0.25 dB output increase. On the ∞:1 line, increasing the input amplitude beyond the threshold results in no output increase at all, thus y(dB) is limited to the threshold value. This version of the device is called a *limiter* and represents the most extreme form of compression.

Figure 13.2 shows the input/output transfer functions for the expander family. The ratios are reversed. You can see that as the signal falls below the threshold, it is attenuated by the ratio amount. For example, below the threshold on the 1:2 line, every decrease in 1dB at the input results in a decrease of 2 dB at the output. On the 1: ∞ line, when the input falls below the threshold it receives infinite attenuation, that is, it is muted. This version of the device is called a *gate* (or noise gate) and represents the most extreme form of downward expansion. Perhaps the most important aspect of both families is that their gain reduction curves are linear for logarithmic axes. These devices operate in the *log domain*. The two families of dynamics processors yield four combinations: compression and limiting, and expansion and gating. We will implement all four of them.

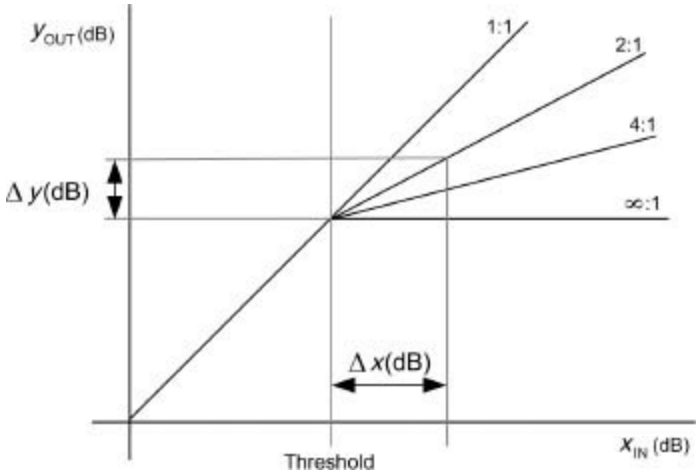


Figure 13.1: The generalized input/output transfer curves for a compressor at various reduction ratios. The ratio is in the form Δ*x*:Δ*y*.

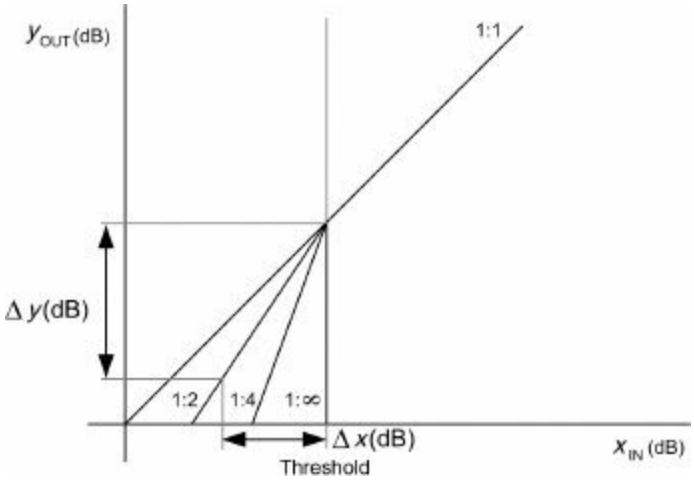


Figure 13.2: The generalized input/output transfer curves for a downward expander at various reduction ratios. The ratio is in the form Δ*x*:Δ*y*.

Figure 13.3 shows the feed-forward and feedback topologies for dynamics processors. There is some debate as to which is best; we'll stay out of that argument and design the feed-forward version first. Then, you can design a feedback version for yourself. Both designs include a detector/computer and a *digitally controlled amplifier/attenuator* (DCA). The detector analyzes the input signal and computes a new gain value to be applied to the signal. The detector/computer block lies in the *side-chain* of the circuit.

Because the compressor and downward expander are both gain reduction devices, an additional stage of make-up gain is required. In some designs, the make-up gain is calculated in the side-chain while in others it sits outside of the DCA. Both accomplish the same task. We are going to add another gain stage at the input (similar to the mod filter in Chapter 12) to drive the detector; this pre-gain block is useful when the source material is very low in amplitude, making it difficult to trigger the detector. Figure 13.4 shows the block diagram of the first dynamic processor plug-in.

Figure 13.4 labels the input and control signals as follows:

- $x(n)$, $y(n)$ = input/output
- $d(n)$ = detector output

- $G(n)$ = gain value for current sample
- d_o = threshold in dB
- ρ = ratio

The CEnvelopeDetector object you used in the last chapter has a log output mode so we can combine the detector and log conversion blocks together, as shown in the dotted box in Figure 13.4. These feed the gain calculation block, which uses the threshold (in dB) and the ratio to set the gain for a given detector output. Orfanidis (1996), Zöler (2011), and Reiss (2011) have given equations that describe the gain reduction calculation. The Zöler equations (Equations 13.1 and 13.2) seem to be the most concise. These equations require additional slope variables (*CS* and *ES*) that are calculated from the ratio. In the compressor, *CS* varies between 0 and 1 as the ratio r varies from 1 to ∞. In the downward expander, *ES* varies between 0 and −∞ as the ratio r varies from ∞ to 1.

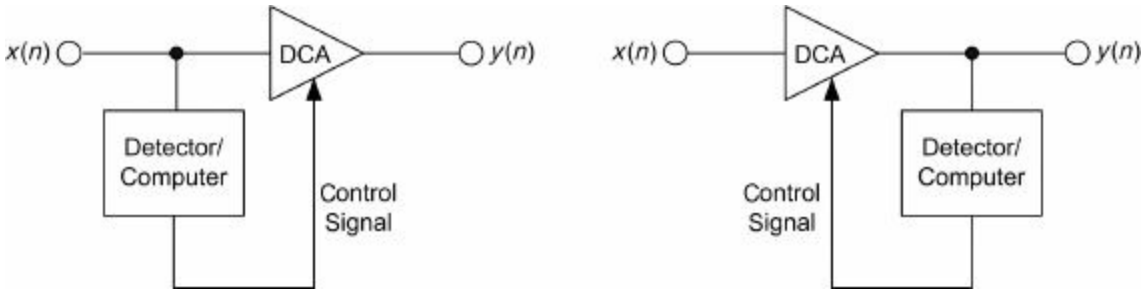


Figure 13.3: The feed-forward and feedback topologies for a dynamics processor. A detector calculates the peak or RMS value of the signal to create a control signal. The DCA attenuates the input signal according to its control value.

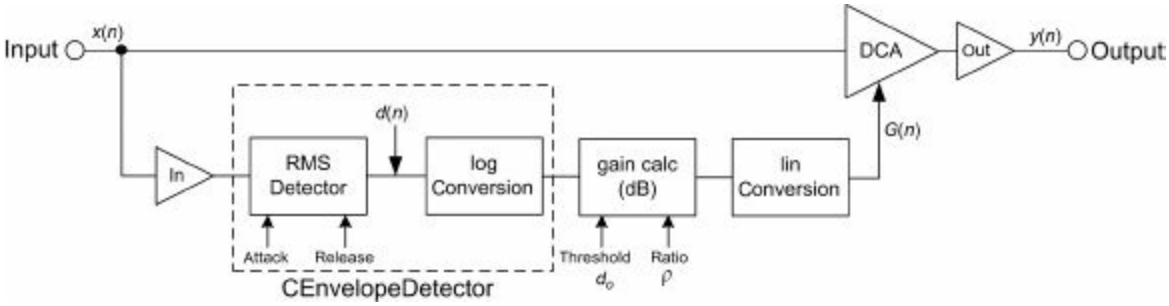


Figure 13.4: A generalized feed-forward topology for the compressor/downward expander families.

Compressor gain (dB):

$$G(n) = f(d(n))$$

where

$$CS = \begin{cases} 1 - (1/\rho) & \text{for compression} \\ 1.0 & \text{for limiting} \end{cases}$$

$$f = \begin{cases} CS \cdot (d_0 - d(n)) & \text{if } d(n) \geq d_0 \\ 0.0 & \text{if } d(n) < d_0 \end{cases} \quad (13.1)$$

Downward expander gain (dB):

$$G(n) = f(d(n))$$

where

$$ES = \begin{cases} (1/\rho) - 1 & \text{for downward expansion} \\ 0.0 & \text{for gating} \end{cases}$$
$$f = \begin{cases} ES \cdot (d_0 - d(n)) & \text{if } d(n) < d_0 \\ 0.0 & \text{if } d(n) \geq d_0 \end{cases} \quad (13.2)$$

The point in the input/output plot where the signal hits the threshold, thereby engaging the device, is called the *knee* of the compression curve. With a hard knee the signal is either above the threshold or below it and the device kicks in and out accordingly. The soft-knee compressor allows the gain control to change more slowly over a width of dB around the threshold called the *knee width* (*W*). In this way, the device moves more smoothly into and out of compression or downward expansion. Figure 13.5 compares hard- and soft-knee curves.

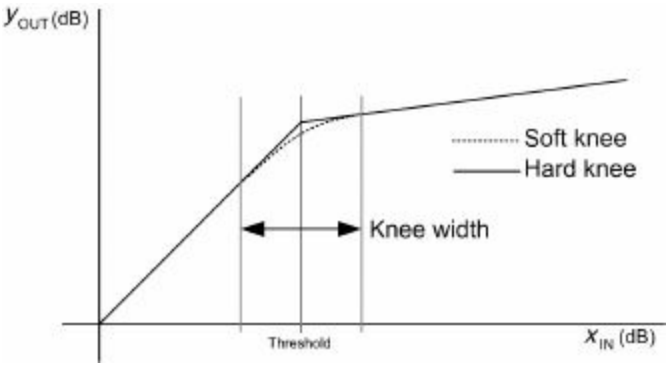


Figure 13.5: Hard and soft-knee compression curves.

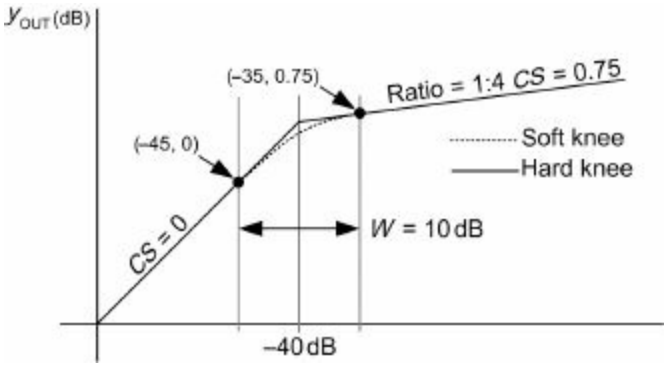


Figure 13.6: Setting up the endpoints for interpolating the *CS* value for a soft-knee compressor.

There are several approaches to generating the soft-knee portion of the curve when calculating the gain reduction value. Curve fitting by interpolating across the knee width is popular. There are a variety of different ways to implement the curve fitting. We already have a Lagrange interpolator in the software, so we can use this to fit a second-order polynomial curve to the knee. This is done by altering the *CS* or *ES* values. The two values above and below the threshold are the *x* values, while the two values of the slope variable at the end points are the *y* components. When the detector signal is in the knee zone around the threshold, we use its value to calculate a new slope. Figure 13.6 shows an example. The knee width (*W*) is 10 dB with a −40 dB threshold. For a compression ratio of 1:4, *CS* is 0.75. Therefore, the soft-knee *CS* will vary between 0 at −45 dB to 0.75 at −35 dB. The

Lagrange interpolation will find the *CS* value for a given input detection value across this range.

13.1 Design a Compressor/Limiter Plug-In

We are going to build a complete dynamics processor with the following features and functions:

- Compressor
- Limiter
- Expander
- Gate
- Analog or digital time constant
- Attack, release controls for detector
- Threshold, ratio controls for gain computer
- Knee-width control: 0 = hard knee processing
- Gain reduction meter to view the reduction

You'll notice a new feature in this plug-in: the ability to monitor a signal and display it on a meter. We will start with the compressor/limiter function and build up from there.

13.1.1 Project: DynamicsProcessor

Create a new project called “Dynamics Processor.” You don't need any stock objects or other advanced features.

13.1.2 DynamicsProcessor: GUI

You need to add sliders and radio button controls to implement the eight control blocks required for the plug-in. Table 13.1 shows the graphical user interface (GUI) slider properties for the design.

Table 13.1: The GUI controls for the dynamics processor plug-in.

Slider Property	Value	Slider Property	Value
Control Name	Det Gain	Control Name	Threshold
Units	dB	Units	dB
Variable Type	float	Variable Type	float
Variable Name	m_flnputGain_dB	Variable Name	m_fThreshold
Low Limit	−12	Low Limit	−60
High Limit	20	High Limit	0
Initial Value	0	Initial Value	0

Slider Property	Value	Slider Property	Value

Control Name	Attack Time	Control Name	Release Time
Units	mSec	Units	mSec
Variable Type	float	Variable Type	float
Variable Name	m_fAttack_mSec	Variable Name	m_fRelease_mSec
Low Limit	1	Low Limit	20
High Limit	300	High Limit	5000
Initial Value	20	Initial Value	1000

Slider Property	Value	Slider Property	Value
Control Name	Ratio	Control Name	Output Gain
Units		Units	dB
Variable Type	float	Variable Type	float
Variable Name	m_fRatio	Variable Name	m_fOutputGain_dB
Low Limit	1	Low Limit	0
High Limit	20	High Limit	20
Initial Value	1	Initial Value	0

Slider Property	Value
Control Name	Knee Width
Units	dB
Variable Type	float
Variable Name	m_fKneeWidth
Low Limit	0
High Limit	20
Initial Value	0

Button Property	Value
Control Name	Processor
Units	
Variable Type	enum
Variable Name	m_uProcessorType
Enum String	COMP, LIMIT, EXPAND, GATE

Button Property	Value
Control Name	Time Constant

Units	
Variable Type	enum
Variable Name	m_uTimeConstant
Enum String	Digital, Analog

13.1.3 DynamicsProcessor.h File

We need to add the following new functions and variables; most can be taken directly from your ModFilter project. There's not much to the basic device. You will need the following:

- An envelope detector for each channel (left, right).
- A function to calculate a compression gain value, given the detector input; this function will also implement the limiting function.

```
// Add your code here: ------------------------------------------------------ //
// envelope detectors
CEnvelopeDetector m_LeftDetector;
CEnvelopeDetector m_RightDetector;

// calculate the compressor G(n) value from the Detector output
float calcCompressorGain(float fDetectorValue, float fThreshold, float fRatio,
                         float fKneeWidth, bool bLimit);
// END OF USER CODE ---------------------------------------------------------- //
```

13.1.4 DynamicsProcessor.cpp File

We need to implement Equation 13.1 and also need to add the code to provide the Lagrange interpolation when the soft-knee function is enabled in the calcCompressorGain() function. The gain calculations simply follow Equation 13.1. See the Lagrange interpolation function in the pluginconstants.h file for documentation. You provide it with *x* and *y* arrays for the end points along with the order (two, since we have two endpoints) and the *x* value to calculate the interpolated *y* value.

```
// calculate the compressor G(n) value from the Detector output
float CDynamicsProcessor::calcCompressorGain(float fDetectorValue, float fThreshold,
                                             float fRatio, float fKneeWidth, bool bLimit)
{
	// slope variable
	float CS = 1.0 - 1.0/fRatio; // [Eq. 13.1]

	// limiting is infinite ratio thus CS->1.0
	if(bLimit)
		CS = 1;

	// soft-knee with detection value in range?
	if(fKneeWidth > 0 && fDetectorValue > (fThreshold - fKneeWidth/2.0) &&
	   fDetectorValue < fThreshold + fKneeWidth/2.0)
	{
		// setup for Lagrange
		double x[2];
		double y[2];
		x[0] = fThreshold - fKneeWidth/2.0;
		x[1] = fThreshold + fKneeWidth/2.0;
		x[1] = min(0, x[1]); // top limit is 0dBFS
		y[0] = 0;      // CS = 0 for 1:1 ratio
		y[1] = CS;    // current CS

		// interpolate & overwrite CS
		CS = lagrpol(&x[0], &y[0], 2, fDetectorValue);
	}

	// compute gain; threshold and detection values are in dB
	float yG = CS*(fThreshold - fDetectorValue); // [Eq. 13.1]

	// clamp; this allows ratios of 1:1 to still operate
	yG = min(0, yG);

	// convert back to linear
	return pow(10.0, yG/20.0);
}
```

prepareForPlay()

- Initialize the detector objects; note the boolean flags that set the analog/digital and log or linear calculation; we use log.

```
bool __stdcall CDynamicsProcessor::prepareForPlay()
{
	// Add your code here:
	// init the envelope detectors
	// set all params at once with this function; see function definition
	if(m_uTimeConstant == Digital)
	{
		m_LeftDetector.init((float)m_nSampleRate, m_fAttack_mSec, m_fRelease_mSec,
				false, DETECT_MODE_RMS, true);
		m_RightDetector.init((float)m_nSampleRate, m_fAttack_mSec,
				m_fRelease_mSec, false, DETECT_MODE_RMS, true);
	}
	else
	{
		m_LeftDetector.init((float)m_nSampleRate, m_fAttack_mSec, m_fRelease_mSec,
				true, DETECT_MODE_RMS, true);
		m_RightDetector.init((float)m_nSampleRate, m_fAttack_mSec,
				m_fRelease_mSec, true, DETECT_MODE_RMS, true);
	}

	return true;
}
```

userInterfaceChange()

- Set the new attack time on the detectors.
- Set the new release time on the detectors.
- Set the time constant mode on the detectors.
- Note: Make sure you check your variable control ID values to match your UI.

```
bool __stdcall CDynamicsProcessor::userInterfaceChange(int nControlIndex)
{
	// decode the control index, or delete the switch and use brute force calls
	switch(nControlIndex)
	{
		case 2:
		{
			m_LeftDetector.setAttackTime(m_fAttack_mSec);
			m_RightDetector.setAttackTime(m_fAttack_mSec);
			break;
		}
```

```
            case 3:
            {
                    m_LeftDetector.setReleaseTime(m_fRelease_mSec);
                    m_RightDetector.setReleaseTime(m_fRelease_mSec);
                    break;
            }

            case 42:
            {
                    if(m_uTimeConstant == Digital)
                    {
                            m_LeftDetector.setTCModeAnalog(false);
                            m_RightDetector.setTCModeAnalog(false);
                    }
                    else
                    {
                            m_LeftDetector.setTCModeAnalog(true);
                            m_RightDetector.setTCModeAnalog(true);
                    }
        }
        default:
                break;
    }

    return true;
}
```

processAudioFrame()

- Follow the block diagram to complete the operations.
- Apply input gain to detector.
- Detect input sample.
- Calculate gain.
- Apply dynamic gain reduction and static make-up gain.

```
bool __stdcall CDynamicsProcessor::processAudioFrame(float* pInputBuffer, float*
                                                     pOutputBuffer, UINT
                                                     uNumInputChannels, UINT
                                                     uNumOutputChannels)
{
        //
        // output = input -- change this for meaningful processing
        //
        // Do LEFT (MONO) Channel; there is always at least one input, one output
        // calculate gains
        float fInputGain = pow(10.0, m_fInputGain_dB/20.0);
        float fOutputGain = pow(10.0, m_fOutputGain_dB/20.0);

        // detect left channel
        float fLeftDetector = m_LeftDetector.detect(pInputBuffer[0]);
```

```
        // gain calc
        float fGn = 1.0;

        // branch
        if(m_uProcessorType == COMP)
                fGn = calcCompressorGain(fLeftDetector, m_fThreshold, m_fRatio,
                                         m_fKneeWidth, false);
        else if(m_uProcessorType == LIMIT)
                fGn = calcCompressorGain(fLeftDetector, m_fThreshold, m_fRatio,
                                         m_fKneeWidth, true);

        // form left output and apply make up gain
        pOutputBuffer[0] = fGn*pInputBuffer[0]*fOutputGain;

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0];

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
        {
                // detect right channel
                float fRightDetector = m_RightDetector.detect(pInputBuffer[1]);

                // gain calc
                float fGn = 1.0;

                // branch
                if(m_uProcessorType == COMP)
                        fGn = calcCompressorGain(fRightDetector, m_fThreshold, m_fRatio,
                                                 m_fKneeWidth, false);
                else if(m_uProcessorType == LIMIT)
                        fGn = calcCompressorGain(fRightDetector, m_fThreshold, m_fRatio,
                                                 m_fKneeWidth, true);

                // form right output and apply make up gain
                pOutputBuffer[1] = fGn*pInputBuffer[1]*fOutputGain;
        }

        return true;
}
```

Build and test the plug-in. Try all the controls. Test the analog and digital time constants. Try extreme settings and listen for artifacts called “pumping and breathing.” The soft-knee function can sometimes be subtle, so you may have to try a variety of different audio files to hear it. It would be nice to have a visual way of seeing the gain reduction, so we will use an advanced RackAFX feature that will enable a metering device. To make it even better, we can invert the meter so that gain reduction results in a downward meter view—common in dynamics processor devices, both analog and digital.

In RackAFX, right click on one of the LED meters in the meter section of the main UI, shown in Figure 13.7. The meters accept a value between 0.0 and 1.0 and display accordingly. This is perfect for us since our gain values are also fluctuating on that same range. You only need to declare a floating point variable to set the meter; RackAFX does everything else. A dialog box will pop up that allows you to name the metering variable you are going to attach to the meter as well as customize its look and behavior. You will be adding two meters, one for the left and one for the right channel gain reduction.

GRL stands for gain reduction for the left and GRR stands for gain reduction for the right channel. Set up the two meters according to Table 13.2.

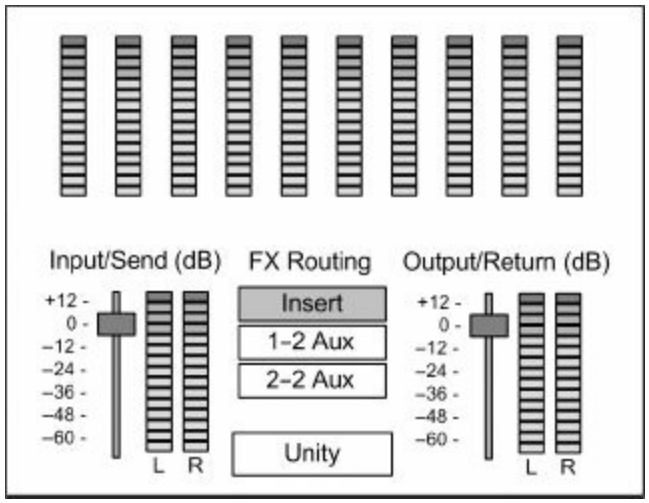


Figure 13.7: The 10 assignable meters in RackAFX.

Table 13.2: Meter settings for the dynamics processor.

Meter Blurb	GRL	Meter Blurb	Value
Invert Meter	true	Invert Meter	GRR
Enable Meter	true	Enable Meter	true
Meter Cal	linear	Meter Cal	linear
Detect Mode	PEAK	Detect Mode	PEAK
Meter Variable	m_fGRMeterValue_L	Meter Variable	m_fGRMeterValue_R
Attack Time	0	Attack Time	0
Release Time	0	Release Time	0

The gain reduction value we will be metering is linear, from 0 to 1.0. The gain reduction value will have its own attack and release times according to our settings, so we want to monitor the PEAK variable type with no attack or release time.

All you need to do is make sure the variables m_fGRMeterValue_L and m_fGRMeterValue_R are reset to the current gain reduction value on each sample interval, and the meter will track it. The meter value needs to go up as the gain reduction goes down to visualize it properly. To set the meter, add one line of code to your processAudioFrame() function for each of the left and right channels.

13.1.5 DynamicsProcessor.cpp File

processAudioFrame()

- Add the metering variables to track the gain reduction values.

```
bool __stdcall CDynamicsProcessor::processAudioFrame(float* pInputBuffer, float*
                                    pOutputBuffer, UINT uNumInputChannels,
                                    UINT uNumOutputChannels)
{
       // Do LEFT (MONO) Channel; there is always at least one input/one output

       <SNIP SNIP SNIP>

       // form left output and apply make up gain
       pOutputBuffer[0] = fGn*pInputBuffer[0]*fOutputGain;

       // set the meter to track 1-gain value
       m_fGRMeterValue_L = 1.0 - fGn;

       <SNIP SNIP SNIP>

              // form right output and apply make up gain
              pOutputBuffer[1] = fGn*pInputBuffer[1]*fOutputGain;

              // set the meter to track 1-gain value
              m_fGRMeterValue_R = 1.0 - fGn;
       }

       etc…
```

Build and test the plug-in now and you'll get a visual cue about how much gain reduction is taking place. Try the analog limiter setting and adjust the attack, release, and threshold values to give smooth limiting (note the ratio slider doesn't work in limiter mode). Once you have that running properly, we'll finish off the dynamics processor by adding the downward expansion/gating capability.

13.2 Design a downward expander/gate plug-in

We'll add on to the current project. Here is what is needed:

- A function to calculate downward expander gain
- Branching in the processAudioFrame() function to implement all four dynamics processor operations

13.2.1 DynamicsProcessor.h File

```
// Add your code here: ------------------------------------------------- //
// envelope detectors
CEnvelopeDetector m_LeftDetector;
CEnvelopeDetector m_RightDetector;

// calcualte the compressor G(n) value from the Detector output
float calcCompressorGain(float fDetectorValue, float fThreshold, float fRatio,
                                float fKneeWidth, bool bLimit);

// calculate the downward expander G(n) value from the Detector output
float calcDownwardExpanderGain(float fDetectorValue, float fThreshold, float
                                fRatio, float fKneeWidth, bool bGate);

// END OF USER CODE ---------------------------------------------------- //
```

Add the following function declaration for the downward expander calculation:

13.2.2 DynamicsProcessor.cpp File

Implement the downward expander function including the soft-knee operation.

```
// calculate the downward expander G(n) value from the Detector output
float CDynamicsProcessor::calcDownwardExpanderGain(float fDetectorValue, float
                                                   fThreshold, float fRatio, float
                                                   fKneeWidth, bool bGate)
{
        // slope variable
        float ES = 1.0/fRatio - 1; // [Eq. 13.2]

        // gating is infinite ratio; ES -> -1.0
        if(bGate)
                ES = -1;

        // soft-knee with detection value in range?
        if(fKneeWidth > 0 && fDetectorValue > (fThreshold - fKneeWidth/2.0) &&
           fDetectorValue < fThreshold + fKneeWidth/2.0)

        {
                // setup for Lagrange
                double x[2];
                double y[2];
                x[0] = fThreshold - fKneeWidth/2.0;
                x[1] = fThreshold + fKneeWidth/2.0;
                x[1] = min(0, x[1]); // top limit is 0dBFS
                y[0] = ES;  // current ES
                y[1] = 0;  // 1:1 ratio

                // interpolate the value
                ES = lagrpol(&x[0], &y[0], 2, fDetectorValue);
        }

        // compute gain; threshold and detection values are in dB
        float yG = ES*(fThreshold - fDetectorValue); // [Eq. 13.2]

        // clamp; this allows ratios of 1:1 to still operate
        yG = min(0, yG);

        // convert back to linear
        return pow(10.0, yG/20.0);
}
```

processAudioFrame()

- Similar to the compressor, add the branching code to calculate the downward expansion gain.

```
bool __stdcall CDynamicsProcessor::processAudioFrame(float* pInputBuffer, float*
                                                     pOutputBuffer,
                                                     UINT uNumInputChannels,
                                                     UINT uNumOutputChannels)
{
        // Do LEFT (MONO) Channel; there is always at least one input/one output

        <SNIP SNIP SNIP>

        // gain calc
        float fGn = 1.0;

        // branch
        if(m_uProcessorType == COMP)
                fGn = calcCompressorGain(fLeftDetector, m_fThreshold, m_fRatio,
                                         m_fKneeWidth, false);
        else if(m_uProcessorType == LIMIT)
                fGn = calcCompressorGain(fLeftDetector, m_fThreshold, m_fRatio,
                                         m_fKneeWidth, true);
        else if(m_uProcessorType == EXPAND)
                fGn = calcDownwardExpanderGain(fLeftDetector, m_fThreshold, m_fRatio,
                                               m_fKneeWidth, false);

        else if(m_uProcessorType == GATE)
                fGn = calcDownwardExpanderGain(fLeftDetector, m_fThreshold, m_fRatio,
                                               m_fKneeWidth, true);

        // form left output and apply make up gain
        pOutputBuffer[0] = fGn*pInputBuffer[0]*fOutputGain;

        <SNIP SNIP SNIP>

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
        {
               // detect right channel
               float fRightDetector = m_RightDetector.detect(pInputBuffer[1]);

               // gain calc
               float fGn = 1.0;

               // branch
               if(m_uProcessorType == COMP)
                       fGn = calcCompressorGain(fRightDetector, m_fThreshold, m_fRatio,
                                                m_fKneeWidth, false);
               else if(m_uProcessorType == LIMIT)
                       fGn = calcCompressorGain(fRightDetector, m_fThreshold, m_fRatio,
                                                m_fKneeWidth, true);
               else if(m_uProcessorType == EXPAND)
                       fGn = calcDownwardExpanderGain(fLeftDetector, m_fThreshold,
                                                      m_fRatio, m_fKneeWidth, false);
               else if(m_uProcessorType == GATE)
                       fGn = calcDownwardExpanderGain(fLeftDetector, m_fThreshold,
                                                      m_fRatio, m_fKneeWidth, true);

               // form right output and apply make up gain
               pOutputBuffer[1] = fGn*pInputBuffer[1]*fOutputGain;
               etc…
```

Build and test the plug-in now and try all four modes with several different input files; drums and guitars make good test files. The analog time constant along with the soft-knee settings generally provide a smoother, less obtrusive gain reduction function.

13.3 Design a Look-Ahead Compressor Plug-In

A problem with the detector circuit is that for any non-zero attack and release time settings, the detector output lags behind the input in time. Remember from Chapter 12 how the output of the detector tracks the input signal, as shown in Figure 13.8. The dark line is the detector output, which is clearly lagging behind the input.

Since the detector output lags, the gain reduction actually misses the event that needs to be attenuated. In the digital world, we can accommodate for this detector lag by using a look-ahead technique. We can't look ahead into the future of a signal, but we can delay the present. If we insert a delay line in the forward signal path (not the side-channel) we can make up for the detector delay so that the detector begins charging before the signal actually gets to the DCA, as shown in Figure 13.9.

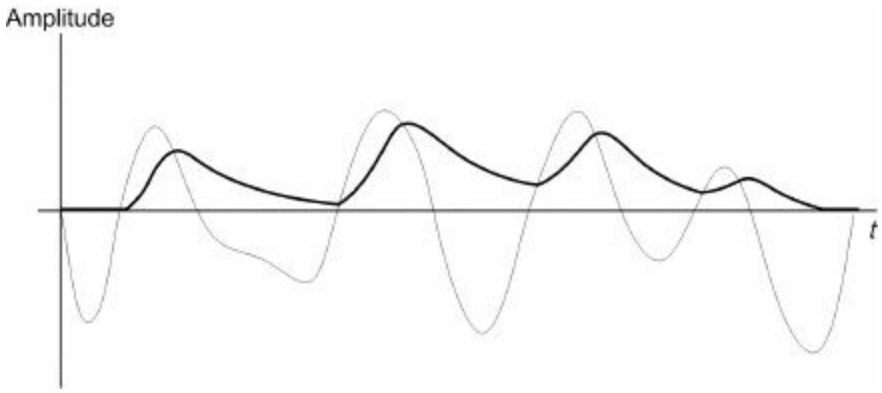


Figure 13.8: The detector output with the medium attack and medium decay times.

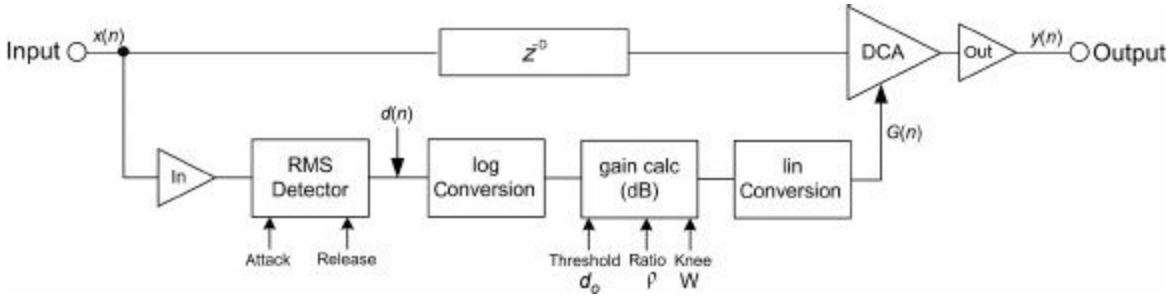


Figure 13.9: The look-ahead compressor block diagram.

You can use the CDelay object that comes as a "stock reverb object" and add it to your project now for our look-ahead pre-delay. Do this by using the Edit button and edit the project. Check the box marked "Stock Reverb Objects" and then hit OK. Your compiler will ask you if you want to reload the project, so answer yes. You will then have the reverb objects included in your project. We'll add a single new slider to control the look-ahead time. Remember when using the CDelay-based objects that you need to make sure you initialize the object to have the same maximum delay time as your slider. The amount of look-ahead delay will depend on the compressor settings. We'll make the maximum value match our maximum attack time of 300 mSec.

13.3.1 DynamicsProcessor: GUI

Add a new slider for the look-ahead delay time in mSec using Table 13.3.

Table 13.3: The look-ahead slider properties.

Slider Property	Value
Control Name	Look-Ahead
Units	mSec
Variable Type	float
Variable Name	m_fLookAheadDelay_mSec
Low Limit	0
High Limit	300
Initial Value	0

13.3.2 DynamicsProcessor.h File

Declare the delays for the look-ahead:

```
// Add your code here: ------------------------------------------------------ //
// envelope detectors
CEnvelopeDetector m_LeftDetector;
CEnvelopeDetector m_RightDetector;

<SNIP SNIP SNIP>

// Delay lines for a look-ahead compressor
CDelay m_LeftDelay;
CDelay m_RightDelay;
// END OF USER CODE --------------------------------------------------------- //
```

13.3.3 DynamicsProcessor.cpp File

prepareForPlay()

- Initialize the delays now that we know the sample rate.
- Set the current delay time in mSec.
- Flush the delays.

```
bool __stdcall CDynamicsProcessor::prepareForPlay()
{
	// init lookahead delays
	m_LeftDelay.init(0.3*(float)m_nSampleRate);
	m_RightDelay.init(0.3*(float)m_nSampleRate);

	// set the current value
	m_LeftDelay.setDelay_mSec(m_fLookAheadDelay_mSec);
	m_RightDelay.setDelay_mSec(m_fLookAheadDelay_mSec);

	// flush delays
	m_LeftDelay.resetDelay();
	m_RightDelay.resetDelay();

	etc…
```

userInterfaceChange()

- Set the look-ahead delay time on the delay elements.
- Note: Make sure your control IDs match your GUI.

```
bool __stdcall CDynamicsProcessor::userInterfaceChange(int nControlIndex)
{
      <SNIP SNIP SNIP>

            case 7:
            {
                  m_LeftDelay.setDelay_mSec(m_fLookAheadDelay_mSec);
                  m_RightDelay.setDelay_mSec(m_fLookAheadDelay_mSec);
                  break;
            }
      etc…
```

processAudioFrame()

- Delay the signal going to the DCA, not the detector side-chain.

```
bool __stdcall CDynamicsProcessor::processAudioFrame(float* pInputBuffer, float*
                                                     pOutputBuffer,
                                                     UINT uNumInputChannels,
                                                     UINT uNumOutputChannels)
{
      <SNIP SNIP SNIP>

      // delay the left DCA after gain calculation
      float fLookAheadOut = 0;
      m_LeftDelay.processAudio(&pInputBuffer[0], &fLookAheadOut);

      // form left output and apply make up gain
      pOutputBuffer[0] = fGn*fLookAheadOut*fOutputGain;

            <SNIP SNIP SNIP>

            // delay the right DCA after gain calculation
            float fLookAheadOut = 0;
            m_RightDelay.processAudio(&pInputBuffer[1], &fLookAheadOut);

            // form right output and apply make up gain
            pOutputBuffer[1] = fGn*fLookAheadOut*fOutputGain;

            // set the meter to track 1-gain value
            m_fGRMeterValue_R = 1.0 - fGn;
      }
      return true;
 }
```

Build and test the plug-in now and try the look-ahead feature. It will take some critical listening to adjust the look-ahead value; start by matching the attack time, and then tweak it from there.

13.4 Stereo-Linking the Dynamics Processor

You can see from the block diagram and your own code that our stereo dynamics processor shares all of the controls: gains, attack, release, threshold, and ratio. However, the ultimate value of the dynamic gain factor $G(n)$ really depends on the audio content of each channel, which is fed into the independent left and right detectors. If the left and right channels have very different signals, then the resulting very different $G(n)$ values for left and right can create a confusing output. Stereo-linking a dynamics processor means that you tie together the two detector outputs, sum and average them, and

then apply that one signal to the log converter. The resulting $G(n)$ value is applied to both left and right channels. Figure 13.10 shows how the right channel's gain computation has been removed since it now shares the left channel's side-chain path. In order to modify the existing processor, we need to do two things: modify the UI to add a stereo link switch and add more branching in processAudioFrame() to do the linking.

13.4.1 DynamicsProcessor: GUI

For a change, I'll use a slider control in enum mode to make a switch for the stereo link. Remember that you can use sliders to make enumerated-list variables in addition to the buttons. Use Table 13.4 to set up the slider.

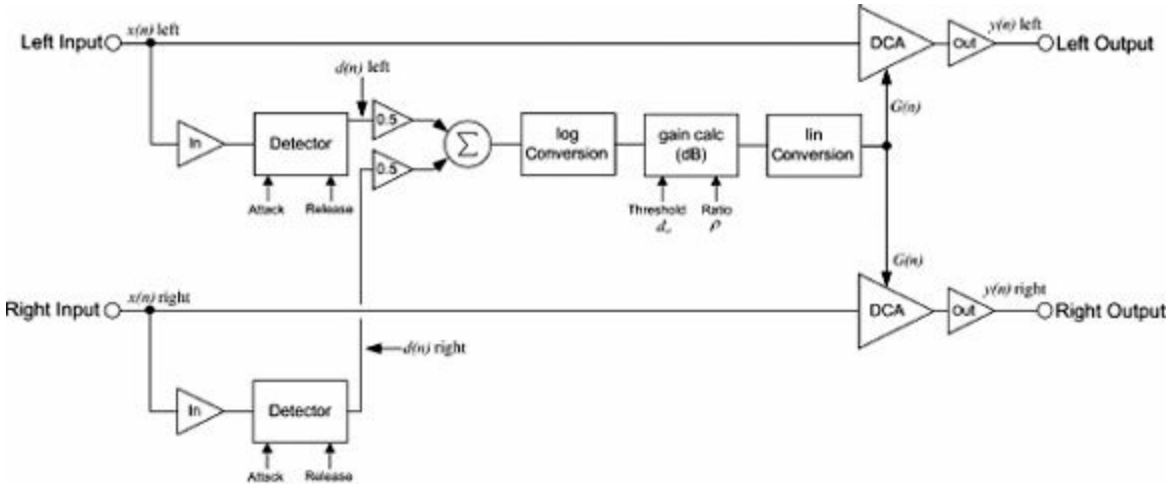


Figure 13.10: The stereo-linked dynamics processor.

NOTE: The detectors are running in log output mode. This means that we need to convert their values back to linear before summing them. After they are summed, they must be converted back to the log domain prior to being used in the gain calculation function.

Table 13.4: The stereo link slider properties.

Slider Property	Value
Control Name	Stereo Link
Units	enum
Variable Type	m_uStereoLink
Enum String	ON, OFF

13.4.2 DynamicsProcessor.cpp File

There are many different ways to do this branching and you should definitely try to figure out your own way. My code is for education purposes and not very streamlined either. For the branching code, I chose to set up the stereo link and right channel detection early on.

```
bool __stdcall CDynamicsProcessor::processAudioFrame(float* pInputBuffer, float*
                                                     pOutputBuffer,
                                                     UINT uNumInputChannels,
                                                     UINT uNumOutputChannels)
{
	// calculate gains
	float fInputGain = pow(10.0, m_fInputGain_dB/20.0);
	float fOutputGain = pow(10.0, m_fOutputGain_dB/20.0);

	// setup stereo link
	float xn_L = fInputGain*pInputBuffer[0];
	float xn_R = 0.0; // for later

	// do left
	float fLeftDetector = m_LeftDetector.detect(xn_L);
	float fRightDetector = fLeftDetector; // use in case of mono file


	// check for right side; can use again in right side code
	if(uNumOutputChannels == 2)
	{
		// get the right side
		xn_R = fInputGain*pInputBuffer[1];

		// detect it
		fRightDetector = m_RightDetector.detect(xn_R);
	}

	// start with Left
	float fLinkDetector = fLeftDetector;
	float fGn = 1.0;

	if(m_uStereoLink == ON) // this works even with mono files
	{

		// detectors output log values; convert to linear to sum them
		fLinkDetector = 0.5*(pow(10.0, fLeftDetector/20.0) + pow(10.0,
		                     fRightDetector/20.0));

		// convert back to log
		fLinkDetector = 20.0*log10(fLinkDetector);
	}
	// branch
	if(m_uProcessorType == COMP)
		fGn = calcCompressorGain(fLinkDetector, m_fThreshold, m_fRatio,
		                         m_fKneeWidth, false);
	else if(m_uProcessorType == LIMIT)
		 fGn = calcCompressorGain(fLinkDetector, m_fThreshold, m_fRatio,
		                          m_fKneeWidth, true);

	etc… same for other branches

	// delay the left DCA
	float fLookAheadOut = 0;
	m_LeftDelay.processAudio(&pInputBuffer[0], &fLookAheadOut);

	<SNIP SNIP SNIP>
```

```
// Stereo-In, Stereo-Out (INSERT Effect)
if(uNumInputChannels == 2 && uNumOutputChannels == 2)
{
        float fGn = 1.0;

        // if not linked, overwrite the variable
        if(m_uStereoLink == OFF)
                fLinkDetector = fRightDetector;

        // branch
        if(m_uProcessorType == COMP)
                fGn = calcCompressorGain(fLinkDetector, m_fThreshold, m_fRatio,
                                         m_fKneeWidth, false);
        else if(m_uProcessorType == LIMIT)
                fGn = calcCompressorGain(fLinkDetector, m_fThreshold, m_fRatio,
                                         m_fKneeWidth, true);
        else if(m_uProcessorType == EXPAND)
                fGn = calcDownwardExpanderGain(fLinkDetector, m_fThreshold,

        etc… same for other branches

        // delay the righr DCA
        float fLookAheadOut = 0;
        m_RightDelay.processAudio(&pInputBuffer[1], &fLookAheadOut);

        etc…
```

13.5 Design a Spectral Compressor/Expander Plug-In

In this project, you will design a spectral dynamics processor. A spectral processor splits up the input signal into two or more frequency bands, then applies signal processing independently to each band. Finally, the processed filtered outputs are summed to produce the final output. In this way, you can apply signal processing to only one band of frequencies, or apply different types of processing to different bands. In this design, we will create a two-band spectral compressor/expander. We will use complementary low-pass filter (LPF) and high-pass filter (HPF) units to split the incoming audio into two bands: low-frequency (LF) and high-frequency (HF). Then, we will process each band independently and recombine the outputs as shown in Figure 13.11. You can compress, say, only the HF content to squash cymbals and other sibilance. Or you can smooth out the bass by compressing it slightly, or any combination of both that you like.

Take a look at Figure 13.8 and check out the features. The input LPF and HPF are adjusted with a single f_c control. This means their cut-off points are always overlapping. The outputs are processed through two independent dynamics processors with independent make-up gains, then the results are summed back together. This means the UI is going to contain a lot of sliders. But, RackAFX will be taking care of most of the GUI details. We need to focus on the input filter-bank first.

For the input LPF and HPF we will be using a special variant on the Butterworth filter called the Linkwitz–Riley filter (Equation 13.3). It has the same second-order roll-off as the Butterworth, but the cutoff point is −6 dB rather than −3 dB so that recombining the filtered signals results in a flat response. Remember that filters alter the phase response as well, so recombining them in parallel offers a special problem with how the phases recombine. *For the two signals to recombine properly, you must invert one of the filter outputs*. It doesn't matter whether you invert the LPF or HPF output as long as you invert one of them. This is specific to the Linkwitz-Riley filters. The filter design equations are shown from Chapter 6.

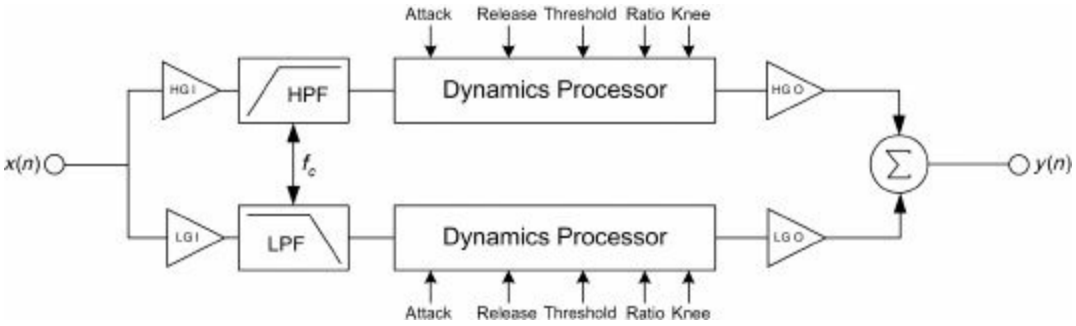


Figure 13.11: A two-band spectral compressor.

$$
\begin{array}{ll}
\text{LPF} & \text{HPF} \\
\theta_c = \pi f_c / f_s & \theta_c = \pi f_c / f_s \\
\Omega_c = \pi f_c & \Omega_c = \pi f_c \\
\kappa = \dfrac{\Omega_c}{\tan(\theta_c)} & \kappa = \dfrac{\Omega_c}{\tan(\theta_c)} \\
\delta = \kappa^2 + \Omega_c^2 + 2\kappa\Omega_c & \delta = \kappa^2 + \Omega_c^2 + 2\kappa\Omega_c \\
a_0 = \dfrac{\Omega_c^2}{\delta} & a_0 = \dfrac{\kappa^2}{\delta} \\
a_1 = 2\dfrac{\Omega_c^2}{\delta} & a_1 = \dfrac{-2\kappa^2}{\delta} \\
a_2 = \dfrac{\Omega_c^2}{\delta} & a_2 = \dfrac{\kappa^2}{\delta} \\
b_1 = \dfrac{-2\kappa^2 + 2\Omega_c^2}{\delta} & b_1 = \dfrac{-2\kappa^2 + 2\Omega_c^2}{\delta} \\
b_2 = \dfrac{-2\kappa\Omega_c + \kappa^2 + \Omega_c^2}{\delta} & b_2 = \dfrac{-2\kappa\Omega_c + \kappa^2 + \Omega_c^2}{\delta}
\end{array}
\tag{13.3}
$$

Because these are second-order filters, we can use RackAFX's built-in bi-quad to do the filtering; all we need to do is supply the calculation of the coefficients. Thus, we're going to need to design a stereo dynamics processor with independent attack, release, threshold, ratio, and gain make-up controls. We will need a shared f_c control for the two filters. Although this plug-in shares most of the features of the previous project, for simplicity there are a few modifications:

- This design will not include the look-ahead function.
- The knee width will be a common shared control that both bands will use.

13.5.1 Project: SpectralDynamics

Create a new project named “SpectralDynamics.” Since we are not implementing the look-ahead feature you don't have to add the stock objects.

13.5.2 SpectralDynamics: GUI

The GUI is basically a double version of the previous design without the look-ahead function. However, we also need to add a cutoff frequency slider to control the point where the bands split and

a knee-width control. Tables 13.5 and 13.6 show all the controls you will need. In Table 13.5, all of the settings are identical to the previous project so only the variable names are shown.

NOTE: Unlike the last plug-in, the processor will be hard-coded in stereo link mode. We can then run the detectors in *linear* operation mode, perform the summation, and then convert the linked detector output to dB prior to calling the gain calculation functions. This only serves to simplify the processAudioFrame() method a bit.

Table 13.5: Duplicated sets of controls.

Control	Variable Name
LF Detector gain	m_LF_DetectorGain_dB
LF Threshold	m_LF_Threshold
LF Attack Time	m_LF_Attack_mSec
LF Release Time	m_LF_Release_mSec
LF Ratio	m_LF_Ratio
LF Make Up Gain	m_LF_MakeUpGain_dB

Control	Variable Name
HF Detector gain	m_HF_DetectorGain_dB
HF Threshold	m_HF_Threshold
HF Attack Time	m_HF_Attack_mSec
HF Release Time	m_HF_Release_mSec
HF Ratio	m_HF_Ratio
HF Make Up Gain	m_HF_MakeUpGain_dB

Table 13.6: The two additional sliders for cutoff frequency and knee-width.

Slider Property	Value
Control Name	fc
Units	Hz
Variable Type	float
Variable Name	m_fFilterBankCutoff
Low Limit	100
High Limit	8000
Initial Value	1000

Slider Property	Value
Control Name	Knee Width
Units	dB
Variable Type	float
Variable Name	m_fKneeWidth
Low Limit	0
High Limit	20
Initial Value	0

13.5.3 Additional Slider Controls

Note: These are duplicated sets of controls that are set up exactly like the previous project, with the exception of the variable names.

13.5.4 Spectral Dynamics Buttons

You need the same two button banks as the previous project, one for the processing type and the other for the time-constant mode.

13.5.5 Spectral Dynamics Metering

This plug-in will feature extensive monitoring using the LED meter bank to indicate the LF and HF input, gain reduction, and output levels. The gain reduction meters are inverted. You can also choose different color schemes for the meters. Other than color and inversion, the meters are set up identically to the previous project. You will need to set up six meters to connect to the following variables:

- m_fMeterLFIn
- m_fMeterHFIn
- m_fMeterLFGr
- m_fMeterHFGr
- m_fMeterLFOut
- m_fMeterHFOut

The metering is especially important for the spectral processing because of the differences in energy in the bands; you will get a clearer picture of the equalization nature of this effect. The middle two meters show the gain reduction and are colored green and red and inverted. The input and output use the Volume Unit or “VU meter” color scheme. Figure 13.12 shows the GUI after all the controls have been added.

13.5.6 SpectralDynamics.h File

Because the bulk of the work is in the UI code, you only need a few more components to add to the original dynamics processor. You need the left and right detectors and the gain calculation functions as before. You also need four bi-quads: an LPF/HPF pair for each of the left and right channels. Finally, you need one function to calculate all the bi-quad coefficients at once for a given f_c value.

```
// Add your code here: ------------------------------------------------- //
// envelope detectors X2
CEnvelopeDetector m_Left_LF_Detector;
```

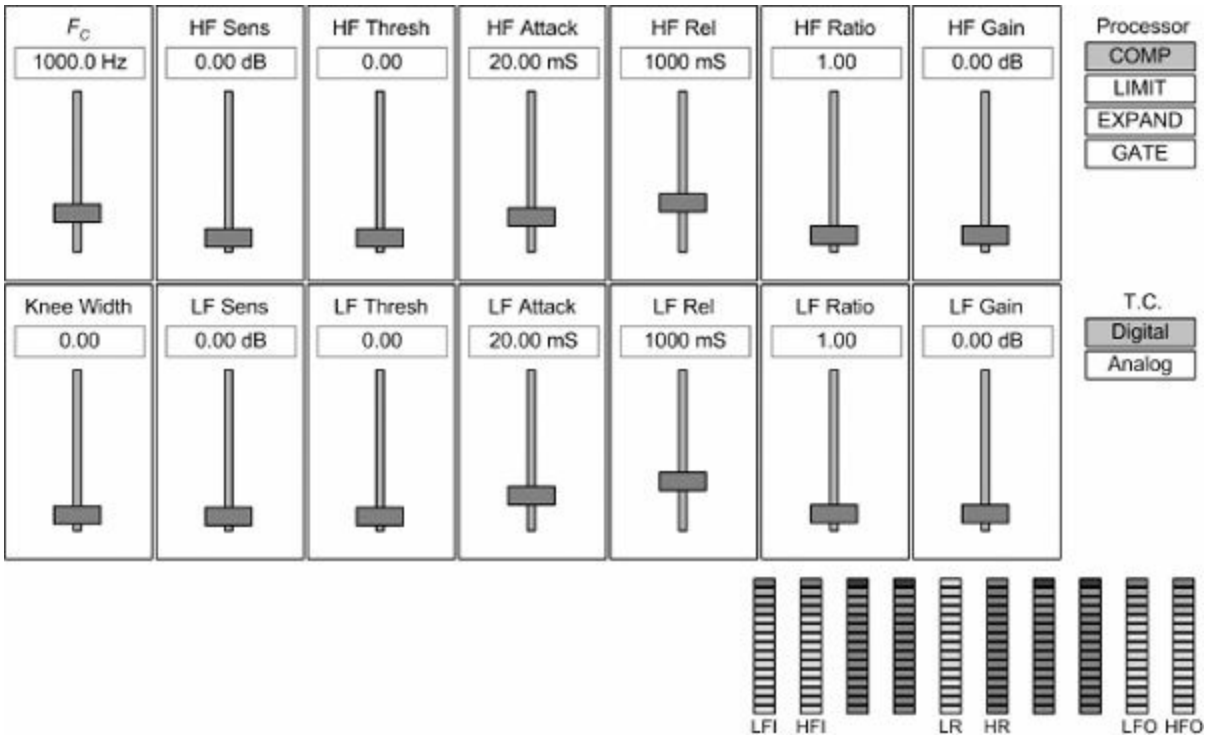


Figure 13.12: The final SpectralDynamics GUI sliders, buttons, and meters.

```
CEnvelopeDetector m_Right_LF_Detector;
CEnvelopeDetector m_Left_HF_Detector;
CEnvelopeDetector m_Right_HF_Detector;

// calculate the compressor G(n) value from the Detector output
float calcCompressorGain(float fDetectorValue, float fThreshold, float fRatio);

// calculate the downward expander G(n) value from the Detector output
float calcDownwardExpanderGain(float fDetectorValue, float fThreshold, float
                               fRatio);
// our input filter banks X2
CBiquad m_LeftLPF;
CBiquad m_LeftHPF;
CBiquad m_RightLPF;
CBiquad m_RightHPF;

// function to set all the filter cutoffs
void calculateFilterBankCoeffs(float fCutoffFreq);

// END OF USER CODE ---------------------------------------------------- //
```

The detector variables are the same as your previous dynamics processor but are split into LF and HF sections or bands.

13.5.7 SpectralDynamics.cpp File

The calcCompressorGain() and calcDownwardExpanderGain() functions are *identical* to the last project you did, so you can cut and paste them. The calculateFilterBankCoeffs() function is pretty straightforward and follows the design equations for the Linkwitz–Riley filters. Since all the filters share the same cutoff, they are pretty easy to deal with.

```
// all filters have same cutoff frequency; only need LPF and HPF calcs
void CSpectralDynamics::calculateFilterBankCoeffs(float fCutoffFreq)
{
	// Shared Factors:
	float omega_c = pi*fCutoffFreq;
	float theta_c = pi*fCutoffFreq/(float)m_nSampleRate;

	float k = omega_c/tan(theta_c);
	float k_squared = k*k;

	float omega_c_squared = omega_c*omega_c;

	float fDenominator = k_squared + omega_c_squared + 2.0*k*omega_c;

	float fb1_Num = -2.0*k_squared + 2.0*omega_c_squared;
	float fb2_Num = -2.0*k*omega_c + k_squared + omega_c_squared;

	// the LPF coeffs
	float a0 = omega_c_squared/fDenominator;
	float a1 = 2.0*omega_c_squared/fDenominator;
	float a2 = a0;
	float b1 = fb1_Num/fDenominator;
	float b2 = fb2_Num/fDenominator;

	// set the LPFs
	m_LeftLPF.m_f_a0 = a0;
	m_LeftLPF.m_f_a1 = a1;
	m_LeftLPF.m_f_a2 = a2;
	m_LeftLPF.m_f_b1 = b1;
	m_LeftLPF.m_f_b2 = b2;

	// right
	m_RightLPF.m_f_a0 = a0;
	m_RightLPF.m_f_a1 = a1;
	m_RightLPF.m_f_a2 = a2;
	m_RightLPF.m_f_b1 = b1;
	m_RightLPF.m_f_b2 = b2;
```

```
        // the HPF coeffs
        a0 = k_squared/fDenominator;
        a1 = -2.0*k_squared/fDenominator;
        a2 = a0;
        b1 = fb1_Num/fDenominator;
        b2 = fb2_Num/fDenominator;

        // set the HPFs
        m_LeftHPF.m_f_a0 = a0;
        m_LeftHPF.m_f_a1 = a1;
        m_LeftHPF.m_f_a2 = a2;
        m_LeftHPF.m_f_b1 = b1;
        m_LeftHPF.m_f_b2 = b2;

        // right
        m_RightHPF.m_f_a0 = a0;
        m_RightHPF.m_f_a1 = a1;
        m_RightHPF.m_f_a2 = a2;
        m_RightHPF.m_f_b1 = b1;
        m_RightHPF.m_f_b2 = b2;
}
```

prepareForPlay()

- Flush delays in all filters.
- Calculate the initial setting of the filters.
- Init the detectors—identical to the previous project except that there are more of them.

```
bool __stdcall CSpectralDynamics::prepareForPlay()
{
    // Add your code here:
    // Flush the filters
    m_LeftLPF.flushDelays();
    m_LeftHPF.flushDelays();
    m_RightLPF.flushDelays();
    m_RightHPF.flushDelays();

    // calculate the Coeffs all at once!
    calculateFilterBankCoeffs(m_fFilterBankCutoff);

    // init detectors
    // set all params at once with this function; false = Digital Time Constant NOT
    // NOTE: Setting detector for linear operaton so we can sum the results;
    //       we have to convert back to log manually
    if(m_uTimeConstant == Digital)
    {
        m_Left_LF_Detector.init((float)m_nSampleRate, m_LF_Attack_mSec,
                                m_LF_Release_mSec, false, DETECT_MODE_RMS,
                                false);

        m_Right_LF_Detector.init((float)m_nSampleRate, m_LF_Attack_mSec,
                                m_LF_Release_mSec, false, DETECT_MODE_RMS,
                                false);

        m_Left_HF_Detector.init((float)m_nSampleRate, m_HF_Attack_mSec,
                                m_HF_Release_mSec, false, DETECT_MODE_RMS,
                                false);
```

```
        m_Right_HF_Detector.init((float)m_nSampleRate, m_HF_Attack_mSec,
                                 m_HF_Release_mSec, false, DETECT_MODE_RMS,
                                 false);
    }
    else
    {
        m_Left_LF_Detector.init((float)m_nSampleRate, m_LF_Attack_mSec,
                                m_LF_Release_mSec, true, DETECT_MODE_RMS,
                                false);

        m_Right_LF_Detector.init((float)m_nSampleRate, m_LF_Attack_mSec,
                                m_LF_Release_mSec, true, DETECT_MODE_RMS,
                                false);

        m_Left_HF_Detector.init((float)m_nSampleRate, m_HF_Attack_mSec,
                                m_HF_Release_mSec, true, DETECT_MODE_RMS,
                                false);

        m_Right_HF_Detector.init((float)m_nSampleRate, m_HF_Attack_mSec,
                                m_HF_Release_mSec, true, DETECT_MODE_RMS,
                                false);
    }
    return true;
}
```

userInterfaceChange()

- Decode the nControlIndex and apply changes to filter and detectors.
- Basically the same as the last project except the addition of the filter and the fact that there are four times as many detectors.

```
bool __stdcall CSpectralDynamics::userInterfaceChange(int nControlIndex)
{
    // decode the control index, or delete the switch and use brute force calls
    switch(nControlIndex)
    {
        // filter cutoff
        case 0:
        {
            calculateFilterBankCoeffs(m_fFilterBankCutoff);
            break;
        }

        // HF Attack
        case 3:
        {
            m_Left_HF_Detector.setAttackTime(m_HF_Attack_mSec);
            m_Right_HF_Detector.setAttackTime(m_HF_Attack_mSec);
            break;
        }
```

```
            // LF Attack
            case 13:
            {
                    m_Left_LF_Detector.setAttackTime(m_LF_Attack_mSec);
                    m_Right_LF_Detector.setAttackTime(m_LF_Attack_mSec);
                    break;
            }

            // HF Release
            case 4:
            {
                    m_Left_HF_Detector.setReleaseTime(m_HF_Release_mSec);
                    m_Right_HF_Detector.setReleaseTime(m_HF_Release_mSec);
                    break;
            }

            // LF Release
            case 14:
            {
                    m_Left_LF_Detector.setReleaseTime(m_LF_Release_mSec);
                    m_Right_LF_Detector.setReleaseTime(m_LF_Release_mSec);
                    break;
            }

            // Time Constant
            case 42:
            {
                  if(m_uTimeConstant == Digital)
                  {
                        m_Left_HF_Detector.setTCModeAnalog(false);
                        m_Right_LF_Detector.setTCModeAnalog(false);
                        m_Right_HF_Detector.setTCModeAnalog(false);
                        m_Right_LF_Detector.setTCModeAnalog(false);
                  }
                  else
                  {
                        m_Left_HF_Detector.setTCModeAnalog(true);
                        m_Right_LF_Detector.setTCModeAnalog(true);
                        m_Right_HF_Detector.setTCModeAnalog(true);
                        m_Right_LF_Detector.setTCModeAnalog(true);
                  }
            }
            default:
                  break;
      }

      return true;
}
```

processAudioFrame()

- Calculate input/output gain values from their dB UI versions.
- Filter the input into LF and HF components; *invert one filter's output*.
- Detect the LF and HF components separately.
- Calculate the dynamic gain factors $G(n)$ for LF and HF components.
- Implement the gain reduction for each band.
- Recombine the outputs of each band.

```
bool __stdcall CSpectralDynamics::processAudioFrame(float* pInputBuffer, float*
                                                    pOutputBuffer, UINT
                                                    uNumInputChannels, UINT
                                                    uNumOutputChannels)
{
        // Do LEFT (MONO) Channel; there is always at least one input/one output
        float fLeftInput = pInputBuffer[0];
        float fRightInput = uNumInputChannels == 1 ? fLeftInput : pInputBuffer[1];

        // setup the input and output gains
        float fLFGain = pow(10.0, m_LF_DetectorGain_dB/20.0);
        float fHFGain = pow(10.0, m_HF_DetectorGain_dB/20.0);

        float fLFOutputGain = pow(10.0, m_LF_MakeUpGain_dB/20.0);
        float fHFOutputGain = pow(10.0, m_HF_MakeUpGain_dB/20.0);


        // split the signal into m_Left LF and HF parts
        float fLeft_LF_Out = m_LeftLPF.doBiQuad(fLeftInput*fLFGain);
        float fLeft_HF_Out = m_LeftHPF.doBiQuad(fLeftInput*fHFGain);

        // invert ONE of the outputs for proper recombination
        fLeft_HF_Out *= -1.0;

        // send these to the detectors: NOTE OUTPUTS ARE LINEAR
        float fLeft_LF_Detector = m_Left_LF_Detector.detect(fLeft_LF_Out);
        float fLeft_HF_Detector = m_Left_HF_Detector.detect(fLeft_HF_Out);

        // split the signal into m_Left LF and HF parts
        float fRight_LF_Out = m_RightLPF.doBiQuad(fRightInput*fLFGain);
        float fRight_HF_Out = m_RightHPF.doBiQuad(fRightInput*fHFGain);

        // invert ONE of the outputs for proper recombination
        fRight_HF_Out *= -1.0;

        // send these to the detectors: NOTE OUTPUTS ARE LINEAR
        float fRight_LF_Detector = m_Right_LF_Detector.detect(fRight_LF_Out);
        float fRight_HF_Detector = m_Right_HF_Detector.detect(fRight_HF_Out);
        // This is the stereo linking of the detector paths;
        // The detectors were set for linear operation so we could sum them
        // they must be converted back to dB before use in the gain calculation
        float LFDetectorSum = 0.5*(fRight_LF_Detector, fLeft_LF_Detector);
        float HFDetectorSum = 0.5*(fRight_HF_Detector, fLeft_HF_Detector);

        // convert back to dB
        LFDetectorSum = 20.0*log10(LFDetectorSum);
        HFDetectorSum = 20.0*log10(HFDetectorSum);

        // sum for input metering
        m_fMeterLFIn = 0.5*(fLeft_LF_Out + fRight_LF_Out);
        m_fMeterHFIn = 0.5*(fLeft_HF_Out + fRight_HF_Out);

        // calculate the gain factors
        float fGn = 1.0;

        // --- LF BAND ---------------
        // branch: all are LF: detect, thresh, ratio
        if(m_uProcessorType == COMP)
                fGn = calcCompressorGain(LFDetectorSum, m_LF_Threshold, m_LF_Ratio,
                                         m_fKneeWidth, false);
        else if(m_uProcessorType == LIMIT)
                fGn = calcCompressorGain(LFDetectorSum, m_LF_Threshold, m_LF_Ratio,
                                         m_fKneeWidth, true);
        else if(m_uProcessorType == EXPAND)
                fGn = calcDownwardExpanderGain(LFDetectorSum, m_LF_Threshold,
                                         m_LF_Ratio, m_fKneeWidth, false);
```

```
        else if(m_uProcessorType == GATE)
                fGn = calcDownwardExpanderGain(LFDetectorSum, m_LF_Threshold,
                                               m_LF_Ratio, m_fKneeWidth, true);

        // create left and right LF outputs
        float fLFOutputL = fGn*fLeft_LF_Out*fLFOutputGain;
        float fLFOutputR = fGn*fRight_LF_Out*fLFOutputGain;

        // gain reduction meter
        m_fMeterLFGr = 1.0 - fGn;

        // --- HF BAND --------------
        // branch: all are HF: detect, thresh, ratio
        if(m_uProcessorType == COMP)
                fGn = calcCompressorGain(HFDetectorSum, m_HF_Threshold, m_HF_Ratio,
                                               m_fKneeWidth, false);
        else if(m_uProcessorType == LIMIT)
                fGn = calcCompressorGain(HFDetectorSum, m_HF_Threshold, m_HF_Ratio,
                                               m_fKneeWidth, true);
        else if(m_uProcessorType == EXPAND)
                fGn = calcDownwardExpanderGain(HFDetectorSum, m_HF_Threshold,
                                               m_HF_Ratio, m_fKneeWidth, false);
        else if(m_uProcessorType == GATE)
                fGn = calcDownwardExpanderGain(HFDetectorSum, m_HF_Threshold,
                                               m_HF_Ratio, m_fKneeWidth, true);
        // create left and right HF outputs
        float fHFOutputL = fGn*fLeft_HF_Out*fHFOutputGain;
        float fHFOutputR = fGn*fRight_HF_Out*fHFOutputGain;

        // meter output
        m_fMeterLFOut = 0.5*(fLFOutputL + fLFOutputR);
        m_fMeterHFOut = 0.5*(fHFOutputL + fHFOutputR);

        // meter GR
        m_fMeterHFGr = 1.0 - fGn;

        // sum outputs
        float fLeftOutput = fLFOutputL + fHFOutputL;
        float fRightOutput = fLFOutputR + fHFOutputR;

        // write the outputs
        pOutputBuffer[0] = fLeftOutput;

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = fRightOutput;

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                pOutputBuffer[1] = fRightOutput;
}
```

Build and test the plug-in. This is a complex device with many interactions. It can sound a lot like an equalizer because it is a very special case of one. Be careful with gain and threshold settings.

Ideas for future projects might include:

- Add separate left, right, HF, and LF meters instead of summing them.
- Add the look-ahead function.
- Split the input into three or more bands (tricky—make a band-pass filter by cascading two Linkwitz–Riley filters, and be careful how you sum the band edges) and run dynamics processing on each band.

- Make a feed-back topology compressor.
- Add parallel compression—a dry path that mixes with the effected signal in a user-chosen ratio.
- Try alternate side-chain configurations (next).

13.6 Alternate Side-Chain Configurations

As Reiss points out, there are several different configurations for the contents of the side-chain. The version we've used so far is a standard linear timing placement configuration. One alternate configuration is called the biased configuration, in which the attack and release levels start at the threshold, not zero. Figure 13.13 shows this configuration. Reiss notes that this has the effect of allowing the release part of the envelope to fade out when the input drops below the threshold. If you want to try to implement this one, you will need to make a modification to the built-in CEnvelopeDetector object to set the low limit levels of the attack and release times—which depend on the linear magnitude of the signal—to match the threshold (which is in dB).

The other two configurations involve the location of the attack and release controls. In the post-gain configuration shown in Figure 13.14 the attack and release controls follow the output of the gain computer and are not intermixed with the RMS detector. The post-gain configuration can also be implemented in RackAFX. You will need to use two CEnvelopeDetectors. The first one runs in RMS mode with zero attack and zero release times. It feeds the gain calculator. The second detector follows the gain computer, runs in PEAK mode and has the user-controlled attack and release controls connected to it. Example code is available at the www.willpirkle.com.

A second variation called log-domain, shown in Figure 13.15, places the timing controls after the gain computer but before the conversion to the linear domain. It could also be implemented using an extra detector placed in the proper location in the side-chain.

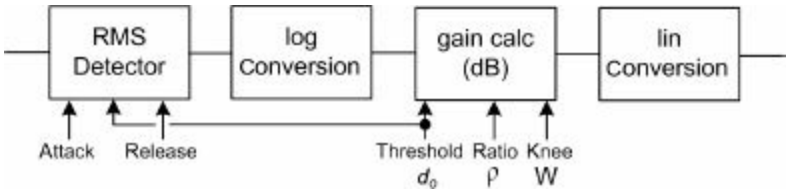


Figure 13.13: In the biased configuration, the threshold is used to set the attack and release starting levels.

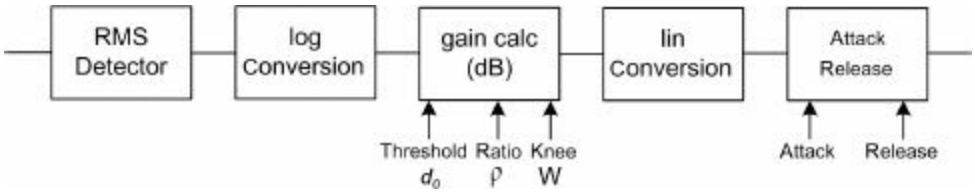


Figure 13.14: The post-gain configuration with timing controls placed after the gain computer.

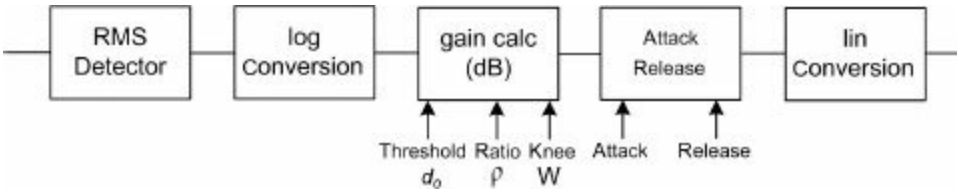


Figure 13.15: The log-domain timing configuration places the timing controls after the gain computer but before linear conversion.

Bibliography

Ballou, G. 1987. *Handbook for Sound Engineers*, pp. 850–860. Carmel, IN: Howard W. Sams & Co.

Floru, F. 1998. "Attack and Release Time Constants in RMS-Based Feedback Compressors." *Journal of the Audio Engineering Society* preprint 4703: 1–4.

Pirkle, W. C. 1995. "Applications of a New High-Performance Digitally-Controlled Audio Attenuator in Dynamics Processing." *Journal of the Audio Engineering Society* preprint 3970: 1–8.

References

Orfanidis, S. 1996. *Introduction to Signal Processing*, Chapters 10–11. Englewood Cliffs, NJ: Prentice-Hall.

Reiss, J. 2011 (October). Under the hood of a dynamic range compressor. Paper presented at the 131st Audio Engineering Society Convention, New York (Tutorials).

Zöler, U. 2011. *DAFX—Digital Audio Effects*, Chapter 4. West Sussex, U.K.: John Wiley & Sons.

Miscellaneous Plug-Ins

There are a few effects left over that didn't have an exact chapter to reside in, so they are presented here. Interestingly, they are some of the simplest to implement but can have a massive sonic impact on the audio they process. These effects include

- Tremolo
- Auto-panning
- Ring modulation
- Wave shaping

14.1 Design a Tremolo/Panning Plug-In

The tremolo is a modulated amplitude effect that uses a low-frequency oscillator (LFO) to directly modulate the output. The LFO waveform is usually triangular or sinusoidal. If the LFO is a square wave, it produces a gapping effect, where intermittent chunks of audio are alternatively muted then restored to unity gain. An auto-panning algorithm pans the signal from left to right following an LFO. Since they are both amplitude-based effects we can combine them into one plug-in. The block diagrams are shown in Figure 14.1.

We will allow the user to switch between tremolo and auto-panning modes. For the tremolo, the LFO will be in unipolar mode, swinging between 0 and 1.0, and the LFO depth will modulate the amplitude of the input signal. The only tricky part is to decide how far below 1.0 the effect will reduce gain. For the tremolo, we define the depth as follows:

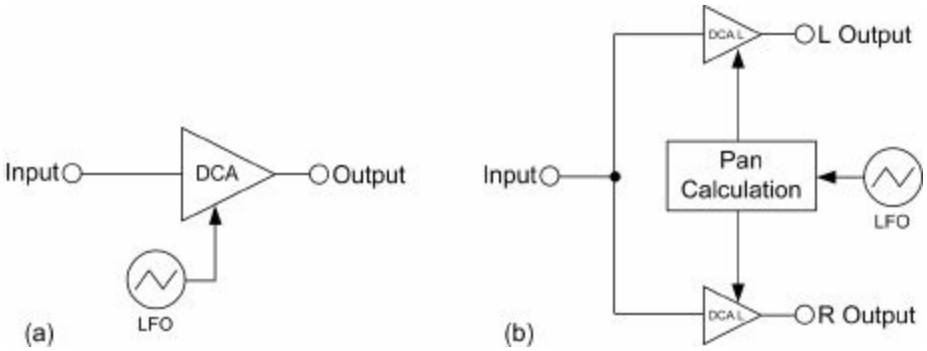


Figure 14.1: (a) A tremolo effect uses an LFO to modulate the amplitude of the input. (b) The auto-panning plug-in uses the LFO to calculate scaling values for the two digitally controlled amplifiers DCA L and DCA R to pan the signal from the left to the right.

- Depth = 100%, the gain will swing from 0.0 to 1.0
- Depth = 75%, the gain will swing from 0.25 to 1.0

- Depth = 50%, the gain will swing from 0.5 to 1.0

For auto-panning, a depth of 0 yields no effect while 100% pans the signal through the full left–right stereo field. The panning is calculated using the constant power pan rule (Equation 14.1). In fact, any plug-in you design that has a pan control on it needs to use the same calculation. Instead of linearly adjusting the left and right gains, they follow the curves of the first quarter cycle of a sin/cos pair. This way, they overlap at the 0.707 points. It should be noted there are other panning schemes, but this one seems to be the most common. The LFO must be in bipolar mode for this to work easily.

$$\begin{aligned} p &= \frac{\pi}{4}(\text{LFO} + 1) \\ \text{Gain}_L &= \cos(p) \\ \text{Gain}_R &= \sin(p) \\ &\text{LFO is bipolar} \end{aligned} \tag{14.1}$$

14.1.1 Project: TremoloPanner

Create the project and name it "TremoloPanner." There are no other objects or options to add.

14.1.2 TremoloPanner: GUI

We need rate and depth controls for the LFO and buttons to change the LFO type and the operational mode of the plug-in shown in Table 14.1.

TremoloPanner.h File

We need to add a wave table object and two gain calculation functions, one for the tremolo and one for the panner; notice the pass-by-pointer method used to return the left and right gain values for the panner.

```
// Add your code here: ------------------------------------------------------ //
// a single LFO
CWaveTable m_LFO;

// a function to calculate the amplitude based on LFO
float calculateGainFactor(float fLFOSample);

void calculatePannerGainFactor(float fLFOSample, float* pLeftVolume,
                               float* pRightVolume);
// END OF USER CODE --------------------------------------------------------- //
```

Table 14.1: TremoloPanner graphical user interface (GUI) controls.

Slider Property	Value	Slider Property	value
Control Name	Mod Rate	Control Name	Mod Depth
Units	Hz	Units	%
Variable Type	float	Variable Type	float
Variable Name	m_ModRate	Variable Name	m_ModDepth

Low Limit	0.02	Low Limit	0
High Limit	25	High Limit	100
Initial Value	5	Initial Value	50

Button property	Value
Control name	LFO
Units	
Variable Type	enum
Variable Name	m_uLFO_Waveform
Enum String	sine, saw, tri, square

Button property	Value
Control name	Mode
Units	
Variable Type	enum
Variable Name	m_uMode
Enum String	Tremolo, Panner

TremoloPanner.cpp File

The tremolo gain calculation will:

- Multiply the LFO sample by depth/100.0.
- Add the value 1.0-depth/100.0 to the result.

That provides the mapping we need for the effect.

```
float CTremoloPanner::calculateGainFactor(float fLFOSample)
{
	// first multiply the value by depth/100
	float fOutput = fLFOSample*(m_fModDepth/100.0);

	// then add the value (1 - m_fModDepth/100.0)
	fOutput += 1 - m_fModDepth/100.0;

	return fOutput;
}

void CTremoloPanner::calculatePannerGainFactor(float fLFOSample,
                                               float* pLeftVolume, float* pRightVolume)
{
	// calc sin/cos quadrant location
	float fPan = ((m_fModDepth/100.0)*fLFOSample + 1)*pi/4.0;

	// equal power calculation
	*pLeftVolume = cos(fPan);
	*pRightVolume = sin(fPan);
}
```

prepareForPlay()

- Initialize the LFO.
- Forward the prepareForPlay() method to the LFO.

```
bool __stdcall CTremoloPanner::prepareForPlay()
{
	// Add your code here:
	m_LFO.m_fFrequency_Hz = m_fModRate;
	m_LFO.m_uPolarity = m_uMode == Panner ? 0 : 1; // 0 = bipolar, 1 = unipolar
	m_LFO.m_uTableMode = 0; // normal, no band limiting
	m_LFO.m_uOscType = m_uLFO_Waveform;
	m_LFO.setSampleRate(m_nSampleRate); // really important!

	// the LFO prepareForPlay() calls reset() and cookFrequency()
	m_LFO.prepareForPlay();

	return true;
}
```

userInterfaceChange()

- Decode the control index and update variables as needed.
- Note: Make sure you check your nControlIndex values to match your GUI.

```
bool __stdcall CTremoloPanner::userInterfaceChange(int nControlIndex)
{
	// decode the control index, or delete the switch and use brute force calls
	switch(nControlIndex)
	{
		case 0:
		{
			m_LFO.m_fFrequency_Hz = m_fModRate;
			m_LFO.cookFrequency();
			break;
		}

		// 41: LFO Waveform change
		case 41:
		{
			m_LFO.m_uOscType = m_uLFO_Waveform;
			break;
		}

		case 42:
		{
			if(m_uMode == Panner)
				m_LFO.m_uPolarity = 0;// 0 = bipolar, 1 = unipolar
			else // is tremolo
				m_LFO.m_uPolarity = 1;// 0 = bipolar, 1 = unipolar
			break;
		}
	}
}
```

processAudioFrame()

- Generate the LFO value.
- Calculate the channel gain values according to the mode.

In processAudioFrame() we need to generate a new LFO value, calculate the new gain factor, then

implement the gain function.

```
bool __stdcall CTremoloPanner::processAudioFrame(float* pInputBuffer,
                                                 float* pOutputBuffer,
                                                 UINT uNumInputChannels,
                                                 UINT uNumOutputChannels)
{
	// Do LEFT (MONO) Channel; there is always at least one input/one output
	float fYn = 0; // normal output
	float fYqn = 0; // quad phase output

	// call the LFO function; we only need first output
	m_LFO.doOscillate(&fYn, &fYqn);

	// setup necessary variables
	float fGnL = 1.0;
	float fGnR = 1.0;
	float fMonoIn = 0.0;

	// branch
	if(m_uMode == Tremolo)
		fGnL = calculateTremoloGainFactor(fYn);
	else // panner sums inputs
	{
		if(uNumInputChannels == 1)
			fMonoIn = pInputBuffer[0];
		else
			fMonoIn = 0.5*(pInputBuffer[0] + pInputBuffer[1]);

		calculatePannerGainFactor(fYn, &fGnL, &fGnR);
	}

	// do MONO (Left) channel
	pOutputBuffer[0] = pInputBuffer[0]*fGnL;

	// Mono-In, Stereo-Out (AUX Effect)
	if(uNumInputChannels == 1 && uNumOutputChannels == 2)
		pOutputBuffer[1] = pOutputBuffer[0]; // just copy

	// Stereo-In, Stereo-Out (INSERT Effect)
	if(uNumInputChannels == 2 && uNumOutputChannels == 2)
	{
		// branch
		if(m_uMode == Tremolo)
		{
			// do right channel, use same gain as left
			pOutputBuffer[1] = pInputBuffer[1]*fGnL;
		}

		else
		{
			// do right channel, its value
			pOutputBuffer[1] = pInputBuffer[1]*fGnR;
		}
	}

	return true;
}
```

Build and test the plug-in. Try raising the LFO rate maximum from 20 Hz to something like 200 or 2 kHz; the results are quite interesting.

14.2 Design a Ring Modulator Plug-In

Ring modulation is accomplished by multiplying the input signal against a sinusoid called the

carrier, as shown in Figure 14.2. This is technically amplitude modulation or frequency mixing. The resulting signal contains sum and difference pairs of each frequency present in the signal. For example, if the carrier is 250 Hz and the input signal is 1 kHz, the output will consist of two frequencies, 1 kHz – 250 Hz = 750 Hz and 1 kHz + 250 Hz = 1.25 kHz. The original input signal at 1 kHz is gone. For complex audio signals, this process happens for every frequency component that the signal contains. The resulting output can sound nothing like the input. Drums change their pitches and full spectrum signals become obliterated. Because of the sum and difference pair, aliasing will occur when the sum signal goes outside Nyquist. One way to handle the aliasing is to band-split the signal just like the spectral compressor and only process the lower frequencies. Though not a perfect solution, it will at least alleviate some of the aliasing. The ring modulator project here only implements the frequency modulation without addressing aliasing problems.

14.2.1 Project: RingModulator

Create a project and name it "RingModulator." There are no other objects or options to add.

14.2.2 RingModulator: GUI

We need frequency and depth controls for the carrier oscillator in Table 14.2. The depth control will increase or decrease the amount of modulation.

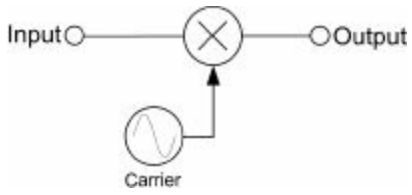


Figure 14.2: The ring modulator block diagram.

Table 14.2: RingModulator plug-in GUI controls.

Slider Property	Value	Slider Property	Value
Control Name	Mod Freq	Control Name	Mod Depth
Units	Hz	Units	%
Variable Type	float	Variable Type	float
Variable Name	m_fModFrequency	Variable Name	m_fModDepth
Low Limit	100	Low Limit	0
High Limit	5000	High Limit	100
Initial Value	1000	Initial Value	50

14.2.3 RingModulator.h File

We only need to add a single wave table object to implement the carrier oscillator.

```
// Add your code here: ------------------------------------------------ //

// the Carrier Oscillator
CWaveTable m_Oscillator;

// END OF USER CODE ---------------------------------------------------- //
```

14.2.4 RingModulator.cpp File

prepareForPlay()

- Initialize the oscillator.
- Forward the preareForPlay() method to the oscillator.

```
bool __stdcall CRingModulator::prepareForPlay()
{
	// Add your code here:
	m_Oscillator.m_fFrequency_Hz = m_fModFrequency;
	m_Oscillator.m_uPolarity = 0; // 0 = bipolar, 1 = unipolar
	m_Oscillator.m_uTableMode = 0; // normal, no band limiting
	m_Oscillator.m_uOscType = 0; // 0 = sin()
	m_Oscillator.setSampleRate(m_nSampleRate); // really important!

	// the prepareForPlay() calls reset() and cookFrequency()
	m_Oscillator.prepareForPlay();

	return true;
}
```

userInterfaceChange()

- If the frequency slider moves, change the oscillator frequency and call the cooking function.
- The depth control is a direct-control variable, so it does not need any attention here.

```
bool __stdcall CRingModulator::userInterfaceChange(int nControlIndex)
{
	// decode the control index, or delete the switch and use brute force calls
	switch(nControlIndex)
	{
		case 0:
		{
			m_Oscillator.m_fFrequency_Hz = m_fModFrequency;
			m_Oscillator.cookFrequency();
			break;
		}

		default:
			break;
	}

	return true;
}
```

processAudioFrame()

- Generate a new carrier oscillator value.
- Multiply the input signal against it.

```
bool __stdcall CRingModulator::processAudioFrame(float* pInputBuffer, float*
                                                 pOutputBuffer, UINT uNumInputChannels,
                                                 UINT uNumOutputChannels)
{
        // Do LEFT (MONO) Channel; there is always at least one input/one output

        // generate carrier
        float yn, yqn;
        m_Oscillator.doOscillate(&yn, &yqn);

        // multiply
        pOutputBuffer[0] = pInputBuffer[0]*yn*(m_fModDepth/100.0);

        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0];

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pInputBuffer[1]*yn*(m_fModDepth/100.0);

        return true;
}
```

Build and test the plug-in. You may find it to be a remarkable effect or you may think it sounds awful. Some things you could do to modify this project include:

- Use multiple carrier frequencies.
- Use series or parallel multiple modulation.
- Use an LFO to modulate the carrier frequency.

14.3 Design a Wave Shaper Plug-In

Wave shaping is a method of nonlinear processing that adds harmonics to the input signal as a result. It is used in synthesizers and distortion algorithms. Because harmonics are added to the signal, wave shapers suffer from aliasing problems, like the ring modulator. The process of wave shaping is simple because no memory elements are required. You treat the input signal x as the argument to a function. The function is often a trigonometric or polynomial function. An interesting function to use is the arctangent. It is part of a family of functions called *sigmoid functions*. Sigmoid functions produce S-shaped curves. The amplitude transfer function of a vacuum tube generally resembles a sigmoid, so this function can be used to try to mimic a tube sound. Figure 14.3 shows the arctangent function $\text{atan}(kx)$, where k controls the amplitude of the input value and thus the amount of nonlinear processing applied. The exact equation is shown in Equation 14.2, where a normalization factor has been added to restrict the output to the range of −1 to +1. You can see in Figure 14.3 that with $k = 1$, the input/output relationship is nearly linear. As k increases, the S-shaped curve emerges and adds gain. For example, with $k = 5$ and $x(n) = 0.25$, $y(n)$ is about 0.6.

$$y(n) = \frac{1}{\arctan(k)} \arctan(kx(n)) \tag{14.2}$$

Asymmetrical distortion can be implemented easily by using two different k-values, one for positive input samples and the other for negative ones. This asymmetrical distortion is found in Class-A tube circuits. Cascading multiple stages will result in more harmonic distortion. In some amplifiers, many stages are cascaded in series. Class A tube circuits also invert the signal. This means that the

asymmetrical distortion curves are also inverted in between each stage. The resulting sound is quite different than simply cascading the modules without inversion. We will design a plug-in that will implement the arctangent wave shaping and allow the user to adjust the following:

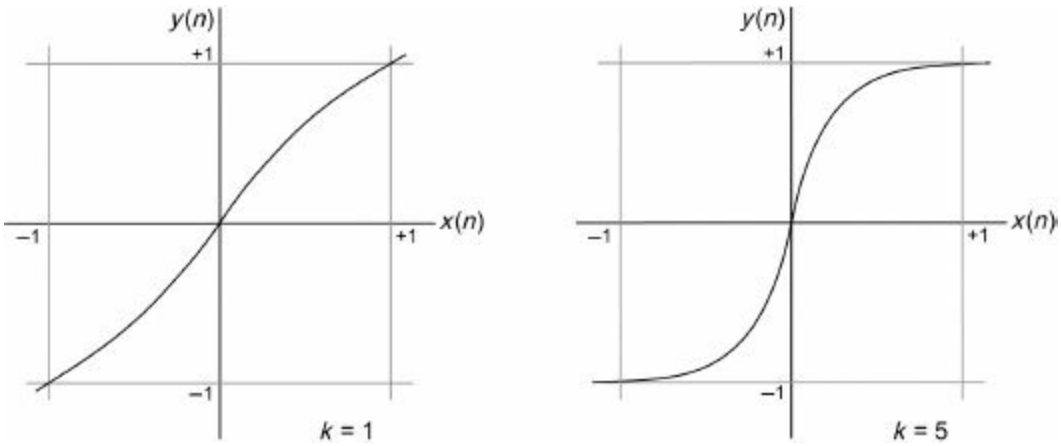


Figure 14.3: The input/output relationship for Equation 14.1 with $k = 1$ and $k = 5$.

- The *k*-value for both the positive and negative halves of the input signal
- The number of stages in series, up to 10
- Inverting or not inverting every other stage when cascaded

14.3.1 Project: WaveShaper

Create a project and name it "WaveShaper." There are no other objects or options to add.

14.3.2 WaveShaper: GUI

Add the controls in Table 14.3 to implement the GUI we will need for the WaveShaper project.

WaveShaper.h File

There's nothing to do in the .h file as we require no additional variables, memory storage spaces, methods, or objects!

WaveShaper.cpp File

The only method that needs to be altered is processAudioFrame().

Table 14.3: WaveShaper plug-in GUI controls.

Slider Property	Value	Slider Property	Value
Control Name	+Atan Exp	Control Name	–Atan Exp
Units		Units	
Variable Type	float	Variable Type	float
Variable Name	m_fArcTanKPos	Variable Name	m_fArcTanKNeg

Low Limit	0.10	Low Limit	0.10
High Limit	20	High Limit	20
Initial Value	1	Initial Value	1

Slider Property	Value	Slider Property	Value
Control Name	Stages	Control Name	Invert Stages
Units		Units	
Variable Type	float	Variable Type	enum
Variable Name	m_nStages	Variable Name	m_uInvertStages
Low Limit	1		
High Limit	10		
Initial Value	1	Enum String	OFF, ON

```
bool __stdcall CWaveShaper::processAudioFrame(float* pInputBuffer, float* pOutputBuffer,
                                              UINT uNumInputChannels,
                                              UINT uNumOutputChannels)
{
	// Do LEFT (MONO) Channel; there is always at least one input/one output
	// (INSERT Effect)
	float f_xn = pInputBuffer[0];

	// cascaded stages
	for(int i=0; i<m_nStages; i++)
	{
		if(f_xn >= 0)
			f_xn = (1.0/atan(m_fArcTanKPos))*atan(m_fArcTanKPos*f_xn);
		else
			f_xn = (1.0/atan(m_fArcTanKNeg))*atan(m_fArcTanKNeg*f_xn);

		// invet every other stage
		if(m_uInvertStages == ON && i%2 == 0)
			f_xn *= -1.0;
	}

	pOutputBuffer[0] = f_xn;
```

```
        // Mono-In, Stereo-Out (AUX Effect)
        if(uNumInputChannels == 1 && uNumOutputChannels == 2)
                pOutputBuffer[1] = pOutputBuffer[0];

        // Stereo-In, Stereo-Out (INSERT Effect)
        if(uNumInputChannels == 2 && uNumOutputChannels == 2)
        {
                float f_xn = pInputBuffer[0];

                // cascaded stages
                for(int i=0; i<m_nStages; i++)
                {
                        if(f_xn >= 0)
                                f_xn = (1.0/atan(m_fArcTanKPos))*atan(m_
                                fArcTanKPos*f_xn);
                        else
                                f_xn = (1.0/atan(m_fArcTanKNeg))*atan(m_
                                fArcTanKNeg*f_xn);

                        // invet every other stage
                        if(m_uInvertStages == ON && i%2 == 0)
                                f_xn *= -1.0;
                }

                pOutputBuffer[1] = f_xn;
        }

        return true;
}
```

Build and test the plug-in. Note that in order to fully replicate a tube circuit you will need much more than just the distortion component, as tube circuits are amplitude sensitive and band limited. However, this plug-in is a good place to start for experimenting with nonlinear processing.

Bibliography

Black, H. S. 1953. *Modulation Theory*. New York: Van Nostrand-Reinhold.

Dodge, C. and T. Jerse. 1997. *Computer Music Synthesis, Composition and Performance*, Chapter 4. New York: Schirmer.

Roads, C. 1996. *The Computer Music Tutorial*, Chapter 2. Cambridge, MA: The MIT Press.

U.S. Navy. 1973. *Basic Electronics*, *Rate Training Manual NAVPES, 10087-C*. New York: Dover Publications.

The VST® and AU® Plug-In APIs

Once you learn how to design plug-ins with RackAFX you will find that learning other plug-in application programming interfaces (APIs) is fairly straightforward. The main reason is that most of the plug-in APIs share very similar structuring and function calls as RackAFX. An analogy might be learning to fly: once you have the fundamentals of solo flight internalized, learning to fly different *kinds* of airplanes is a matter of understanding the peculiarities of each one. You don't need to worry about the principles of lift, drag, pitch, and yaw after you've mastered them. The current popular plug-in APIs include Steinberg's Virtual Studio Technology (VST), Apple Computer's Audio Units (AU), and Avid's Avid Audio eXtension (AAX)® formats. If you want to write for these APIs, the first step is to set up a developer account with the manufacturers via their websites. Once you have an account you can download their API documentation and SDKs (software development kits), which include sample code templates. The VST and AU APIs are publicly documented, while AAX is not. Like RackAFX, VST plug-ins are written in straight C++ without the need for external frameworks or foundation classes. AU plug-ins are not and require a good working knowledge of Mac programming to develop. Once you have a RackAFX plug-in, how can you make it work in other plug-in APIs?

A.1 Compiling as a VST Plug-In in Windows

The simplest way to use your RackAFX plug-in in a third-party client is to take advantage of the fact that you can set your project to compile as a dynamic link library (DLL) that is compatible with both RackAFX and VST on the Windows operating system (OS). You can enable this functionality when you create your project or later in development. If you already have a project in development, use File → Edit Project or the Edit button on the user interface (UI). On the edit project dialog, check the "Make VST Compatible" box, as shown in Figure A.1. The resulting DLL will work with both RackAFX and any Windows VST client. Just copy the DLL from your /PlugIns folder to the designated VST folder for the client.

The GUI options are as follows:

- Use VST client's GUI: VST clients will provide you with a *default interface* whose look and feel are determined by the client's manufacturer. These range from a simple bank of sliders to more elaborate GUIs. The client will query your plug-in for control information and build the GUI at run time. You do not have to provide any extra code for this option.

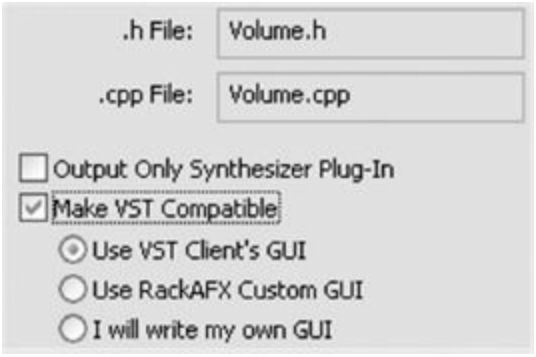


Figure A.1: The VST Compatible switch reveals three options for the GUI.

- Use RackAFX custom GUI: If you have the proper version of RackAFX, this button will be enabled. Choosing this option will allow the GUI you create with GUI designer (Appendix B) to be used by the VST client. It will look and work identically to the one you create in RackAFX. You do not have to provide any extra code for this option.
- I will write my own GUI: If you have the skills to do this, then you can check this box and write your own GUI. The client will call a method on your plug-in to launch your GUI. All details of the GUI are your responsibility and you must provide the code and have the necessary resource editor.

For the last option, first look in the PlugIn.h file and find the flag and three methods you need to override.

```
// set this to TRUE if you have your own GUI to use; not for RackAFX GUIs
bool m_bUserCustomGUI;
```

First, set the m_bUserCustomGUI flag to "true" in your constructor. This will tell the client that you have your own GUI. Next, override the following three methods:

```
// for user (NOT RackAFX) generated resources/GUIs
virtual void __stdcall showGUI(HWND hParentWnd);
virtual void __stdcall hideGUI(HWND hParentWnd);
virtual void __stdcall refreshGUI(HWND hParentWnd);
```

- The client calls showGUI() to show it.
- The client calls hideGUI() to hide it.
- The client calls refreshGUI() to update it. See Appendix B for more information about updating.

In the three methods, the argument is the client's window handle. You will need to create your child window using the client's handle as your parent. If you don't know what that means, you probably shouldn't be using this option.

A.2 Wrapping Your RackAFX Plug-In

Your RackAFX plug-in will compile on Windows (Visual Studio) or Mac (Xcode) because it is pure C++ with some #define statements added to split out the Windows- and Mac-specific components. The audio signal processing and GUI component objects are all done in C++ so your plug-in object can be wrapped with an outer C++ object that translates function calls from the VST or AU client into RackAFX function calls. This is exactly what happens when you choose the "Make VST Compatible"

option; when the client requests a VST object, it receives one with your plug-in embedded inside as a member object. The wrapper object then forwards calls to your plug-in and returns information or audio data to the client, as shown in Figure A.2. Your plug-in exposes an interface of public methods (gray boxes). The wrapper exposes its API specific methods (white boxes) which the client calls. The translation usually involves a simple forwarding method call. Sometimes more than one wrapper method connects to a single plug-in method. Other times, the wrapper method only needs to return a success code or special string so there is nothing to call on the plug-in object.

If you want to write a wrapper object, it is fairly straightforward, as you will see in Section A.3 when we compare the different API function calls. However, you need to know a bit more about how your GUI objects are designed so you can use the GUI controls you set up in RackAFX. Alternatively, you can implement your own scheme. When you set up a control in RackAFX, you are really creating a new C++ object that is added to a linked list. The linked list is a member of your CPlugIn class named "m_UIControlList." It contains a list of C++ CUICtrl objects. A CUICtrl object can represent one of the following:

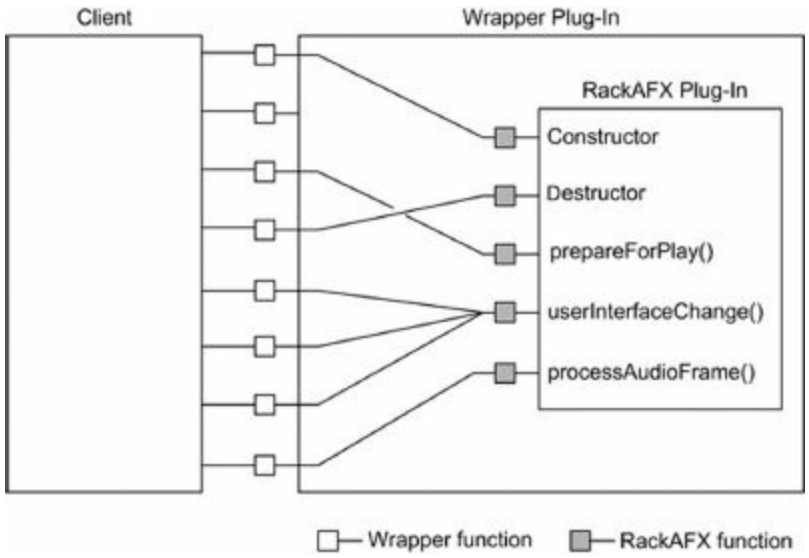


Figure A.2: A wrapper object exposes a client specific interface and routes calls to and from the plug-in.

- Slider
- Radio button bank
- Meter

The simplest way to explain is by example. Look in your initUI() method for any plug-in and you will see the instantiation and initialization of the GUI objects. Here's the first part of the volume slider code from the very first project. The highlights are in bold. Remember, *do not ever edit this code manually*.

```
m_fVolume = 0.750000;
CUICtrl ui0;
ui0.uControlType = FILTER_CONTROL_CONTINUOUSLY_VARIABLE;
ui0.uControlId = 0;
ui0.fUserDisplayDataLoLimit = 0.000000;
ui0.fUserDisplayDataHiLimit = 1.000000;
ui0.uUserDataType = floatData;
ui0.fInitUserIntValue = 0;
ui0.fInitUserFloatValue = 0.750000;
ui0.fInitUserDoubleValue = 0;
ui0.fInitUserUINTValue = 0;
ui0.m_pUserCookedIntData = NULL;
ui0.m_pUserCookedFloatData = &m_fVolume;
ui0.m_pUserCookedDoubleData = NULL;
ui0.m_pUserCookedUINTData = NULL;
ui0.cControlUnits = "                    ";
ui0.cVariableName = "m_fVolume";
ui0.cEnumeratedList = "SEL1,SEL2,SEL3";
ui0.dPresetData[0] = 0.000000;ui0.dPresetData[1] <SNIP SNIP SNIP>
ui0.cControlName = "Volume";
```

The very first line initializes the underlying variable, m_fVolume, as you set it up when you created the slider. The control type is FILTER_CONTROL_CONTINUOUSLY_VARIABLE, which means a continuous controller. The uControlID is 0—this is the value that is passed to userInterfaceChange() when a control is manipulated. This "unique ID" paradigm for identifying the control is universal among the other plug-in APIs. The connection to the variable itself is via a pointer. Since this is a float variable, m_pUserCookedFloatData is set to the address of the underlying variable. This is how RackAFX manipulates it for you. When you wrap your plug-in you can use this to control the variable as well. Of course, you can always use other schemes. The units variable codes several items and will need to be trimmed of white spaces before you use it (do not alter its length in the object, however). Thus, you can see the fundamentals about each variable: the ID, minimum, maximum, and initial value, and the name and units. These are all used in the other plug-in APIs as well.

RackAFX automatically sorts the GUI component objects by type. The sliders and radio button banks are supported by the default GUIs for the VST and AU formats but the meters are not. The m_UIControlList is defined in pluginconstants.h and exposes several methods for you to use:

RackAFX automatically sorts the GUI component objects by type. The sliders and radio button banks are supported by the default GUIs for the VST and AU formats but the meters are not. The m_UIControlList is defined in pluginconstants.h and exposes several methods for you to use:

```
int count();                // returns the total number of GUI objects
int countLegalVSTIF();      // returns the number of legal VST Default GUI objects
CUICtrl* getAt(int nIndex); // returns a pointer to a GUI object at a list index
```

The CPlugIn object also has some helper functions for you to use. Once you know the ID values for the controls, you can use the following CPlugIn methods to access the objects:

```
void setParameter (UINT index, float value);      // set a parameter 0 -> 1
float getParameter (UINT index);                  // get a parameter 0 -> 1
```

These two methods are very important because they mimic the way a VST client will be getting or setting values through the default interface. *The index value is the index in the linked list and not the ID value*. Finally, for Windows plug-ins, you will often need to know the name of the directory where your DLL exists for dealing with presets or loading/saving custom information. CPlugIn exposes the method for you to use to get that folder name.

```
char* getMyDLLDirectory();     // return the DLL's owning directory
```

A.3 Comparison of Objects/Methods/GUIs

RackAFX, VST, and AU share many similarities. First, they all require that you derive your plug-in from a base class (or classes) that the manufacturer specifies. They also implement the same basic set of functionality that you've been coding all along. The base classes are shown in Table A.1. The comparison of methods is shown in Table A.2.

Table A.1: Comparison of the functions and the methods that implement them.

API	Base Class(es)
RackAFX	CPlugIn
VST	AudioEffect, AudioEffectX
AU	AUBase, AUEffectBase, AUKernelBase

Table A.2: Comparison of the functions and the methods that implement them.

Function	RackAFX	VST	AU
Instantiation	CPlugIn()	AudioEffectX()	AUEffectBase() AUKernelBase()
Destruction	~CPlugIn()	~AudioEffectX()	~AUEffectBase() ~AUKernelBase()
Initialization	prepareForPlay()	resume()	Initialize()
Prepare For Play	prepareForPlay()	setSampleRate() updateSampleRate() resume()	N/A
Process Audio	processAudioFrame()	processReplacing()	Process() ProcessBufferList()
Get UI Control Value	getParameter() (optional)	getParameter()	GetParameter()
Set UI Control Value	userInterfaceChange() (optional)	setParameter()	SetParameter()
Update GUI (see			

Appendix B)	sendUpdateGUI()	updateDisplay()	N/A
Get the DLL's Directory	getMyDLLDirectory()	getDirectory()	N/A
Show a Custom GUI	showGUI()	dispatcher()	AudioUnitGetProperty()

RackAFX, VST, and AU all provide for a default standard interface—a GUI that the user does not have to code. For RackAFX, you can optionally design your own custom GUI with the GUI designer (Appendix B) that requires no coding; alternatively, you can design your own (see Section A.1). VST and AU also allow you to design your own GUI. VST provides a GUI class of objects that you can optionally use in your coding that allows for cross-platform compatibility. AU only runs on Mac and does not provide cross-platform compatibility. The default interface setup for RackAFX involves you simply filling in a properties dialog for the components you want; the GUI object code is written for you. In VST and AU, you must set up a way to describe your controls. In all three formats, when you load your plug-in the client queries it for its default interface controls and then builds the GUI at run time. In RackAFX, the client simply iterates through your linked list of objects and sets up the GUI.

A.4 VST Plug-in without Wrapping

Let's start by pretending to write a VST plug-in from scratch *without* wrapping any RackAFX code—of course, you can always cut and paste processing or cooking functions directly from your source, since it is pure C++. The pretend plug-in will be a delay with feedback called CDelay.

A.4.1 Default GUI

First, you define an enumerated list of unsigned integer-type (UINT) constant values in your Plugin.h file where "Plugin" is the name and you have derived it from the AudioEfffectX base class. These constants are the control ID values the host will use when querying about, getting, or setting a control variable. A delay plug-in might have delay time, feedback, and wet/dry mix controls, so you might define your enum as follows:

```
enum
{
        uDelayTime,      /* ID = 0 */
        uFeedBack,       /* ID = 1 */
        uWetDry,         /* ID = 2 */

        uNumParams       /* Always last = 3 = number of controls */
};
```

Next, you declare a *parallel set of member variables* that will be used in your plug-in. These are for cooked data values.

```
float m_fDelayTime;
float m_fFeedBack;
float m_fWetDry;
```

Finally, you declare the minimum and maximum values for each control.

```
float m_fMinDelayTime, m_fMaxDelayTime;
float m_fMinFeedBack, m_fMaxFeedback;
float m_fMinWetDry, m_fMaxWetDry;
```

The last item in the enumeration keeps track of the control count by virtue of being last in the list. During operation, the VST client will use these ID values passed as the variable "index" in the following methods. First, there are three methods to fill out that describe the name, display, and units for each control.

1. getParameterName: You create the name of the control as a string.

```
void CDelay::getParameterName (long index, char *label)
{
        switch (index)
        {
                case uDelayTime:strcpy (label, "Delay"); break;
                case uFeedBack: strcpy (label, "FeedBack"); break;
                case uWetDry:   strcpy (label, "Wet/Dry"); break;
        }
}
```

2. getParameterDisplay: You create a string that represents the value to display on the UI; the float2String() method is a helper function that is provided in the VST object file.

> All default VST controls use values from 0.0 to 1.0 regardless of how they map in your actual plug-in. You must write the cooking and uncooking functions that will convert your internal variable values to and from values from 0.0 to 1.0.

```
void CDelay::getParameterDisplay (long index, char *text)
{
        switch (index)
        {
                case uDelayTime:float2string (m_fDelayTime, text);   break;
                case uFeedBack: float2string (m_fFeedBack, text);    break;
                case uWetDry:   float2string (m_fWetDry, text);      break;
        }
}
```

3. getParameterLabel: You create a string that represents the units for the control.

```
void CDelay::getParameterLabel (long index, char *label)
{
        switch (index)
        {
                case uDelayTime:strcpy (label, "mSec");    break;
                case uFeedBack: strcpy (label, "%");       break;
                case uWetDry: strcpy   (label, "%");       break;
        }
}
```

Next, there are two methods you need to implement to allow the client to get and set your control variables. However, these methods deal with *raw VST client data*.

1. getParameter: You return the raw data (0–1) for the control.

```
float CDelay::getParameter (long index)
{
        float fRawData = 0;

        switch (index)
        {
                case uDelayTime:
                {
                      float fDiff = m_fMaxDelayTime - m_fMinDelayTime;
                      fRawData = (m_fDelayTime - m_fMinDelayTime)/fDiff;
                      break;
                }

                // same thing for the other variables
        }

        return fRawData;
}
```

2. setParameter: You cook the raw data (0–1) for use in your plug-in.

```
void CDelay::setParameter (long index, float value)
{
       switch (index)
       {
              case uDelayTime:
              {
                     m_fDelayTime= (m_fMaxDelayTime - m_fMinDelayTime)* value +
                                   fMinDelayTime;
              }

              // same thing for the other variables
       }
}
```

Thus, every time you add or remove a variable, you have to update your enumerated list, change the variables and min/max pairs, and update the five functions above.

A.4.2 Signal Processing

You will notice that you've already had to write a lot of code just to deal with the UI; you have not written any code to do any meaningful processing. In RackAFX, you alter three functions for signal processing:

1. Constructor
2. prepareForPlay()
3. processAudioFrame()

Constructor

- Init all variables.
- Set our input and output channel count (this is also done in the CPlugIn constructor).
- Set a unique four-character identifier (required by VST).
- Call the base class's constructor to pass it the number of presets (0 for now) and number of controls.

- The audioMasterCallback you see is a mechanism for client 4 plug-in communication; its details are hidden from developers.

```
CDelay::CDelay (audioMasterCallback audioMaster)
        : AudioEffectX (audioMaster, 0, uNumParams) // 0 presets, uNumParams controls
{
        m_fDelayTime = 1000;
        m_fFeedback = 0;
        m_fWetDry = 50;

        // VST Specific setup
        setNumInputs (2);

        setNumOutputs (2);
        setUniqueID ('WDly');
}
```

prepareForPlay() → setSampleRate() and resume()

prepareForPlay() is special for RackAFX because it is called after the sample rate is set. In VST, this is done in setSampleRate(). We also use prepareForPlay() to flush out our delay line buffers and do any other preparations for the next run of the plug-in. The resume() function is called when the VST plug-in is turned on so we can flush buffers there.

```
void CDelay::setSampleRate(float sampleRate)
{
        // Left
        if(m_pLeftBuffer)
                delete m_pLeftBuffer;

        // create a 2 second buffer
        m_pLeftBuffer = new float[2*(int) sampleRate];

        // Right
        if(m_pRightBuffer)
               delete m_pRightBuffer;

        // create a 2 second buffer
        m_pRightBuffer = new float[2*(int) sampleRate];

        // flush buffers
        memset (m_pLeftBuffer, 0, 2*(int) sampleRate * sizeof (float));
        memset (m_pRightBuffer, 0, 2*(int) sampleRate * sizeof (float));
}

void CDelay::resume()
{
        // flush buffer
        memset (m_pLeftBuffer, 0, 2*(int) sampleRate * sizeof (float));
        memset (m_pRightBuffer, 0, 2*(int) sampleRate * sizeof (float));

        // reset read/write indices, etc…
        m_nReadIndex = 0;
        m_nWriteIndex = 0;

        etc…
}
```

processAudioFrame() → processReplacing()

The fundamental difference in the processing functions is in how the data arrives. In RackAFX, the default mechanism is that it arrives in frames, where each frame is a sample from each channel. The frames arrive as pointers to buffers. In VST, the audio is processed in *buffers* of frames, rather than single frames. Figure A.3 shows a two-channel in and two-channel out VST setup.

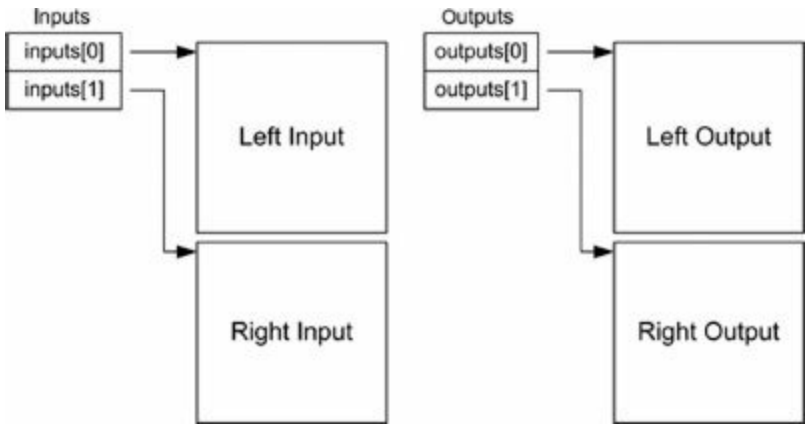


Figure A.3: VST audio data is delivered in buffers of frames.

Instead of passing a pointer to a buffer, you pass a pointer to a buffer of pointers, or float** variables. The processReplacing() method also passes you a count of the number of frames. You use these variables to iterate through and access your audio data. VST was designed to allow you to have any number of input and output channels, which you specify in the constructor. The parameters that are passed to the processReplacing() method are as follows:

- float** inputs: A pointer to a buffer of pointers to input buffers
- float** outputs: A pointer to a buffer of pointers to output buffers
- Long sampleframes: The count of frames in each buffer

```
void CDelay::processReplacing(float **inputs, float **outputs, long sampleframes)
{
        // pick up pointers to buffers
        float *inL = inputs[0];
        float *inR = inputs[1];
        float *outL = outputs[0];
        float *outR = outputs[1];

        UINT uFrameCount = 0;

        while (--sampleFrames >= 0)
        {
                // pick up samples from buffers
                float fxnL = inL[uFrameCount];
                float fxnR = inR[uFrameCount];

                // calculate and write out
                outL[uFrameCount] = doLeftDelayFunction(fxnL);
                outR[uFrameCount] = doRightDelayFunction(fxnR);


                // inc frame counter
                uFrameCount++;
        }
}
```

You can now see the similarities and differences in RackAFX and VST:

- There's more overhead for dealing with the GUI—even if you write your own GUI, you must provide the default implementation functions because not all VST clients have custom GUI capability.
- VST uses setSampleRate() and resume() to change the sample rate or restart the plug-in.
- VST processes in buffers of frames rather than frames.

A.5 VST Plug-In with RackAFX Wrapping

To wrap the RackAFX plug-in you first need to create a new AudioEffectX derived class. Then, you make one of the member variables a RackAFX plug-in object. Suppose you have a delay plug-in named "CStereoDelay" and you want to embed it into the VST plug-in. In the .h file you simply declare an instance of it: CStereoDelay m_Delay.

A.5.1 Default GUI

You still need to fill in those five VST functions, but you will be using your plug-in to supply the information.

getParameterName

- Copy the name of object; you use the getAt() method to find the UI object in the linked list.

```
void CDelay::getParameterName (long index, char *label)
{
        // get the GUI object
        CUICtrl* pUICtrl = m_Delay.m_UIControlList.getAt(index);
        if(!pUICtrl) return;

        // copy the ControlName
        strcpy (label, pUICtrl->cControlName);
}
```

getParameterDisplay

- Access the variable via the pointer; this example assumes all parameters are float variables, but you can always check the value of the uUserDataType variable to find out what kind of variable the object stores and branch accordingly.

```
void CDelay::getParameterDisplay (long index, char *text)
{
        // get the GUI object
        CUICtrl* pUICtrl = m_Delay.m_UIControlList.getAt(index);
        if(!pUICtrl) return;

        // use the helper function to copy the data
        float2string (*pUICtrl->m_pUserCookedFloatData, text);
}
```

getParameterLabel

- You create a *string* that represents the units for the control.

```
void CDelay::getParameterLabel (long index, char *label)
{
        // get the GUI object
        CUICtrl* pUICtrl = m_Delay.m_UIControlList.getAt(index);
        if(!pUICtrl) return;

        // copy the ControlUnits
        strcpy (label, pUICtrl->cControlUnits);
}
```

setParameter/getParameter

- Here is where the CPlugIn helper functions work for you; all you need to do is forward the call.

```
float CDelay::getParameter(long index)
{
        return m_Delay.getParameter(index);
}
void CDelay::setParameter(long index, float value)
{
        m_Delay.setParameter(index, value);
}
```

Constructor

For the constructor you will need to handle one minor detail: you need to know how many legal VST controls you have. Unfortunately, you need to pass these to the AudioEffect base class constructor before your member object plug-in is constructed. You will need to count your slider controls (including any embedded in the LCD control in Appendix B) and radio button bank controls. You can also look through the initUI() method and count all control objects with either the FILTER_CONTROL_CONTINUOUSLY_VARIABLE or FILTER_CONTROL_RADIO_SWITCH_VARIABLE as the uControlType. In this case, we have three legal VST controls.

```
CDelay::CDelay (audioMasterCallback audioMaster)
        : AudioEffectX (audioMaster, 0, 3) // 0 presets, 3 controls
{
        // child object is now instantiated
        //
        // VST Specific setup; get info from embedded plug-in
        setNumInputs (m_Delay.m_uMaxInputChannels);
        setNumOutputs (m_Delay.m_uMaxOutputChannels);
        setUniqueID ('WDly');
}
```

processReplacing

- In order to use processReplacing() you need to combine samples from the input/output (I/O) buffers into frames to send the processAudioFrame() function. This version is for a stereo insert effect.

```
void CDelay::processReplacing(float **inputs, float **outputs, long sampleframes)
{
	// pick up pointers to buffers
	float *inL =  inputs[0];
	float *inR =  inputs[1];
	float *outL = outputs[0];
	float *outR = outputs[1];

	UINT uFrameCount = 0;

	// mini frames
	float fInputFrame[2];
	float fOutputFrame[2];

	while (--sampleFrames >= 0)
	{
		// pick up samples from buffers
		fInputFrame[0] = inL[uFrameCount];
		fInputFrame[1] = inR[uFrameCount];

		// calculate and write out (...2, 2) = 2 in/2 out
		m_Delay.processAudioFrame(&fInputFrame[0], &fOutputFrame[0], 2, 2);

		// inc frame counter
		uFrameCount++;
	}
}
```

A.6 AU Overview

VST was a bit more complex in that you had to do some of the GUI work yourself. Whether you write one from scratch or wrap the RackAFX plug-in, you are still stuck with the default GUI. AU plug-ins are another order of magnitude in complexity. The actual audio processing function is fairly simple. The large majority of the work goes into setting up the default GUI and maintaining your presets. Unlike both RackAFX and VST, AU plug-ins are not ANSI C/C++; they require the Audio Units and Core Foundation components, so you need to be familiar with Mac programming if you are going to get into AU programming. AU uses two base class objects together to implement a plug-in. When you create a new Audio Unit plug-in in XCode, it sets up the derived classes for you, one from each base class. You then override methods on both of them.

- AUEffectBase handles the GUI controls and any other non-audio-processing details.
- AUKernelBase handles the processing and reset controls.

A.6.1 Default GUI

Suppose we have a simple volume plug-in with just one volume control. AU calls a control a "parameter." Similar to VST, you first define an enumerated list of UINT constant values in your Plugin.h file where "Plugin" is the name and you have derived it from the AUEffectBase base class. These constants are the control ID values, just as before.

```
enum {
	kParam_One =0,
	//Add your parameters here... increment kNumberOfParameters

	kNumberOfParameters=1
};
```

In the same file, you create constant definitions for the initial value and name.

```
// set the initial value
static const float kDefaultValue_ParamOne = 0.5;

// set the name; CFStringRef is a Core Foundation String object
static CFStringRef kParameterOneName = CFSTR("Volume");
```

You declare your controls by filling in the GetParameterInfo() method. It is beyond the scope of this book to analyze the code line by line, but you can get a feel for how it works by looking at the way the objects are set up.

```
OSStatus AUVolume::GetParameterInfo(AudioUnitScope       inScope,
                                    AudioUnitParameterID    inParameterID,
                                    AudioUnitParameterInfo  &outParameterInfo)
{
        <SNIP SNIP SNIP>

        switch(inParameterID)
        {
            case kParam_One:
                AUBase::FillInParameterName (outParameterInfo, kParameterOneName,
                false);
                outParameterInfo.unit = kAudioUnitParameterUnit_LinearGain;
                outParameterInfo.minValue = 0.0;
                outParameterInfo.maxValue = 1;
                outParameterInfo.defaultValue = kDefaultValue_ParamOne;
                break;
            default:
                result = kAudioUnitErr_InvalidParameter;
                break;
            }

etc…
```

Here, when the Client queries the plug-in on its first parameter (kParam_One) it fills out a structure about the control. You can see the minimum, maximum, and default values being set. The "unit" variable of the structure is set to kAudioUnitParameterUnit_LinearGain. A linear gain control moves between 0 and 1.0. Another example is kAudioUnitParameterUnit_Boolean, which specifies a boolean control variable. What is interesting here is that you don't declare the actual variable that stores the control data. The variable is a float variable that the framework stores for you. You tell the framework what kind of variable it represents. When you need to get or set the variable, you use GetParameter() and SetParameter(), which get and set a float data type.

A.6.2 Signal Processing

The AU processing function is simply named Process(). The audio data moves in buffers. The buffers can contain any arbitrary number of channels with the simplest systems being $N \times N$ or equal input/outputs. Although the inNumChannels variable is passed to the function, it is always set to 1.0 for the current version of AU. This is OK for mono or when you don't need to process channels independently. A boolean flag is passed to the method, indicating the mute condition and the function simply returns without processing. The parameters that are passed to the Process() method are as follows:

- const Float32: *inSourceP (a pointer to an input buffer)
- Float32: *inDestP (a pointer to the output buffer)
- UInt32: InSamplesToProcess (number of samples in the input buffer)
- UInt32: inNumChannels (number of channels to process)
- bool: &ioSilence (indicates no processing, if true)

Here is the AU process function for a volume plug-in.

```
void AUVolume::AUVolumeKernel::Process(const Float32     *inSourceP,
                                       Float32     *inDestP,
                                       UInt32      inFramesToProcess,
                                       UInt32      inNumChannels,
                                       bool        &ioSilence)

{
        // setup counter
        UInt32 nSampleFrames = inFramesToProcess;

        // setup pointers to I/O
        const Float32 *sourceP = inSourceP;
        Float32 *destP = inDestP;

        // use our own Get method
        Float32 gain = GetParameter( kParam_One );

        while (nSampleFrames-- > 0) {
                Float32 inputSample = *sourceP;

                sourceP += inNumChannels;

                // here's where you do your DSP work
                Float32 outputSample = inputSample * gain;

                *destP = outputSample;
                destP += inNumChannels;
        }
}
```

If you need to process multi-channel audio data independently, then you need to remove the Process() method override and replace it by overriding ProcessBufferLists().

```
ProcessBufferLists(AudioUnitRenderActionFlags &ioActionFlags,
                   const AudioBufferList &inBuffer,
                   AudioBufferList &outBuffer,
                   UInt32 inFramesToProcess)
```

This method passes data back and forth by using AudioBufferList objects. These objects contain pointers so buffers of interleaved data along with a variable that tells you how many channels the buffer holds. For example, for the input buffer:

```
inBuffer.mNumberBuffers = the number of channels in this Buffer List
inBuffer.mBuffers[0].mData = a float* to the buffer for channel 0
inBuffer.mBuffers[1].mData = a float* to the buffer for channel 1, etc…
```

You would typically write something like this to extract them (in this case, we know the channel count is 2):

```
// create the mini-frame arrays
float **inputs = new float*[2 * sizeof(float*)];
inputs[0] = (float*)inBuffer.mBuffers[0].mData;
inputs[1] = (float*)inBuffer.mBuffers[1].mData;

float **outputs = new float*[2 * sizeof(float*)];
outputs[0] = (float*)outBuffer.mBuffers[0].mData;
outputs[1] = (float*)outBuffer.mBuffers[1].mData;

// call your processing function
doProcess(inputs, outputs, inFramesToProcess);
```

In this case, your processing function is just like the VST processReplacing()—it is taking pointers to buffer pointers as arguments, then it would split out the data as in the VST version above. Lastly, the AUKernelBase method Reset() is implemented to reset your plug-in to its just-instantiated state.

More RackAFX Controls and GUI Designer

RackAFX has several optional graphical user interface (GUI) controls that were not covered in the book chapters. It also has a GUI designer that lets you convert your slider-based default GUI into a fully customized version. The website www.willpirkle.com is always the best way to get the latest information on the GUI designer and any additional controls that may be added at a later date. The two additional GUI controls are the alpha wheel and liquid crystal display (LCD) and the vector joystick control.

B.1 The Alpha Wheel and LCD Control

In the upper right of the main RackAFX UI is a control that represents an alpha wheel, LCD matrix view, and the control knob. During the course of designing your project, you've noticed that the LCD is used to give you feedback about the state of your project and compiler. However, it can also be used as a GUI control just like the other sliders. The LCD control allows you to store up to 1024 continuous slider controls inside it. Anything you can set up with a normal slider control will work as one of the LCD embedded controls. And, the GUI designer lets you place one of these controls on your own GUI. There are several uses for the control:

- It's a different way to store the continuous controls.
- If you run out of sliders, you can always put 1024 more of them in the LCD.
- You can use the enumerated list variable to create a set of presets or other global functions or configuration controls.

To set up the LCD control, right-click on it when you are in development mode (not with a plug-in loaded). A box pops up as shown in Figure B.1. Click on New to open the very familiar slider properties dialog. Fill it out just as you would any other slider control. After adding a few more controls, you might have something resembling Figure B.2, with three controls embedded. You can edit the controls, remove them, or move them up and down the list (which changes their sequence in the LCD control).

You treat these embedded slider controls just as any other slider, so there's nothing to add to your normal operational code. When you run your plug-in, use the alpha wheel to select one of the controls and the value knob to adjust its value. Note there is currently no edit control connected to the LCD for manual entry of control values. Figure B.3 shows the LCD control with the alpha wheel adjusted for the bass control.

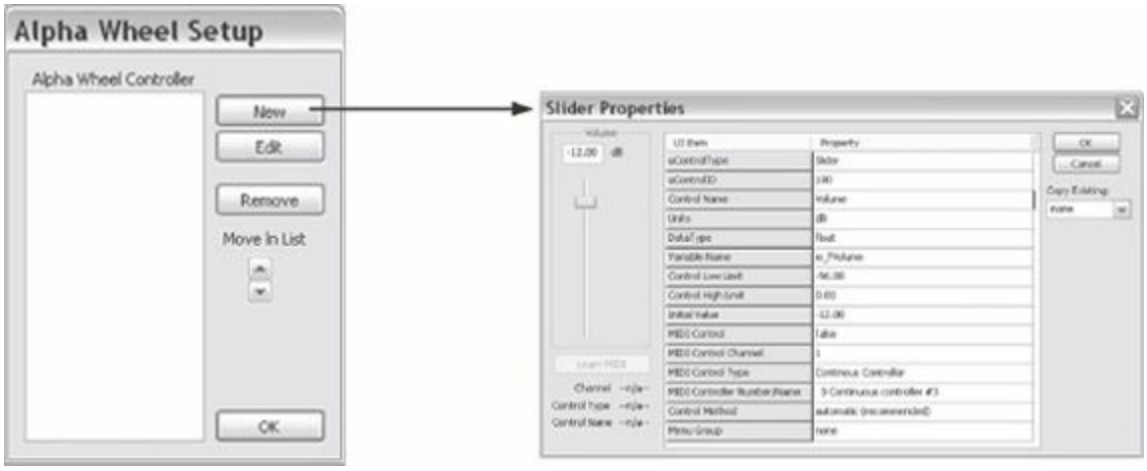


Figure B.1: The New button pops up the slider properties to add a control to the LCD.



Figure B.2: The LCD is loaded with three controls.

An alternative way to use the LCD control is for global operations (like effects configurations) or to store presets, either because you need more than RackAFX provides or you need to write your own methods for loading and storing data. You can use the enumerated list variable to create up to 256 individual string names. You can also take advantage of a feature of the LCD control: You can use the underscore (_) to separate strings in your list (you can't have spaces in an enumerated name). When the LCD control uses them, it replaces the underscores with single white spaces. This allows you much more flexibility than the radio buttons or normal sliders provide. For example, you could set up a preset list with names like “Heavy_Lead,” “Liquid_Pad,” “Lush_Space,” and so on. Then, you can use the up/down arrows to place your “presets” control at the top of the control list. Now, as shown in Figure B.4, the user can select one of your custom presets (remember, it's up to you to code your own preset format) and then scroll to adjust the other controls. The underscores have been replaced with spaces.

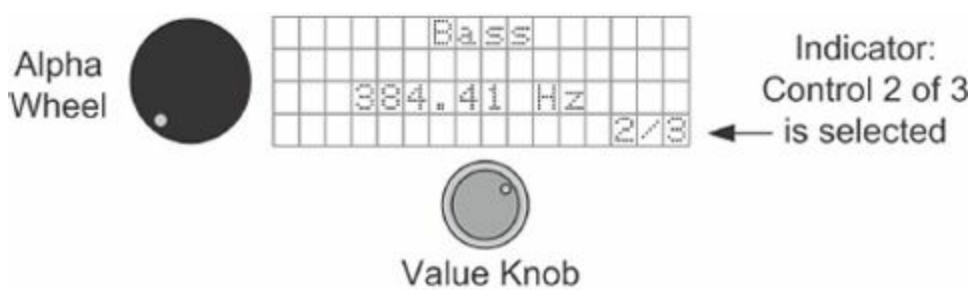


Figure B.3: The alpha wheel and LCD control is reminiscent of vintage synthesizer and rack effects.

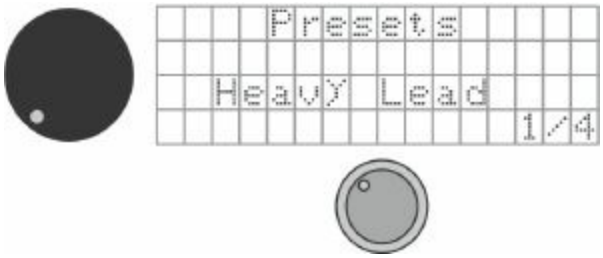


Figure B.4: The underscores are automatically removed from enumerated list variables in the LCD control.

B.2 The Vector Joystick Control

RackAFX also has a dedicated vector joystick control that communicates with your plug-in. Dave Smith designed the first vector synthesizers for Sequential Circuits, Inc. and then later for Yamaha and Korg. The vector joystick is a combination of a traditional *x*/*y* track-pad and an exponential joystick. The calculation documentation is available on the website. In the original synthesizers, the vector joystick was used to program or alter in real time the mixture of two or four different waveforms. The position of the joystick determines the mix ratios, which are not linear but rather exponential. In RackAFX, you can use the joystick to alter up to four different parameters at once, exponentially. In addition, a linear *x*/*y* value of the joystick location is available for your plug-in. Figure B.5 shows the vector joystick and components.

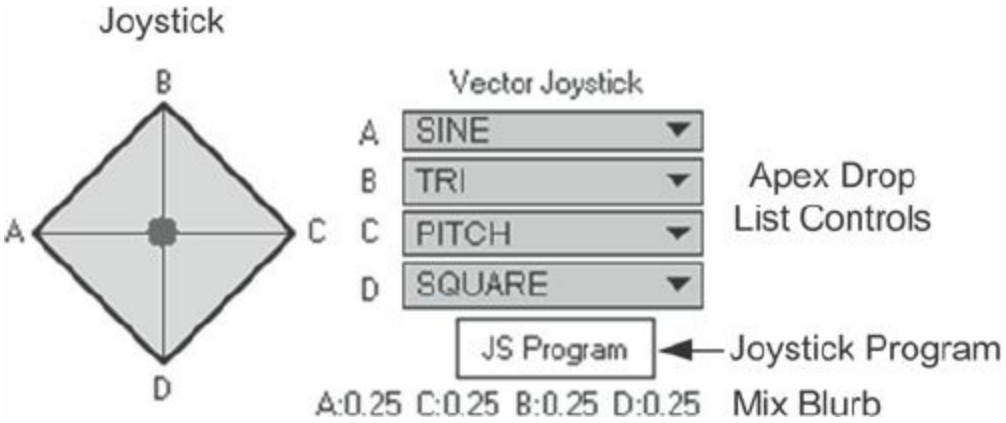


Figure B.5: The RackAFX vector joystick.

The joystick is arranged in a diagonal with four corners labeled A, B, C, D, clockwise; this is in keeping with Dave Smith's original lingo. As the joystick moves around, a new set of mix ratios is generated and appears in the mix blurb. The mix ratios will always add up to 1.0. In the center, the

ratios are all 0.25. A second set of values also exists, the AC-mix and the BD-mix, which are linear. On the AC axis, 0 is at the A apex and 1.0 is at the C corner. On the BD axis, 0 is at D and 1.0 is at B. In the center position, the AC-mix is 0.5 and the BD-mix is also 0.5. Figure B.6 shows the joystick roughly three-quarters of the way from the origin to the A corner. However, the mix blurb shows the value of A to be 0.79, but the other three have dropped off substantially. With the joystick all the way in the A corner, the ratios are {A, B, C, D} = {1, 0, 0, 0} and interestingly, as Figure B.7 demonstrates, when the joystick is halfway between A and B on the outer edge of the diamond, the A/B ratio is 50/50 while the rest are 0. In Figure B.7, the AC-mix is 0.25 and the BD-mix is 0.75.

By sweeping around the field, you can generate many different combinations. You could use these combinations as mix ratios for something in your plug-in. Or, you could use them to control any parameter or other aspect of the plug-in that you like. You are responsible for mapping the 0.0 to 1.0 values to your plug-in's parameters. The vector joystick does not connect to any variable in your plug-in; like the assignable buttons, these passive controls simply tell you when they've moved. When the joystick moves, a function called joystickControlChange() is called.

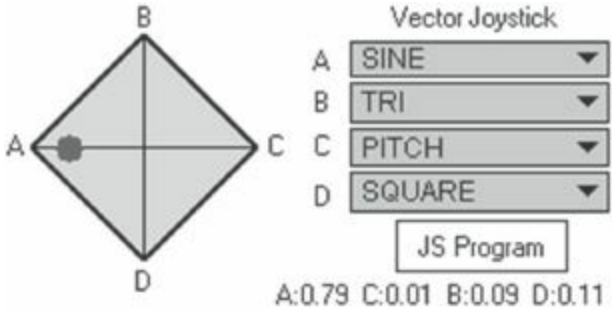


Figure B.6: The exponential mix ratios are evident in the vector joystick.

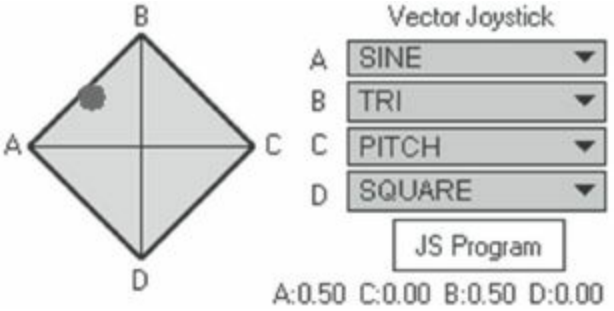


Figure B.7: This position results in the 50/50 mix of A/B only. AC-mix = 0.25, BD-mix = 0.75.

```
/* joystickControlChange

        Indicates the user moved the joystick point; the variables are the relative
        mixes of each axis; the values will add up to 1.0
                        B
                        |
                A       x       C
                        |
                        D
        The point in the very center (x) would be:
        fControlA - 0.25
        fControlB - 0.25
        fControlC - 0.25
        fControlD - 0.25

        AC Mix - projection on X Axis (0 -> 1)
        BD Mix - projection on Y Axis (0 -> 1)
*/

bool __stdcall CVolume::joystickControlChange(float fControlA,
                                              float fControlB,
                                              float fControlC,
                                              float fControlD,
                                              float fACMix,
                                              float fBDMix)
{
        // add your code here

        return true;
}
```

You can see that RackAFX leaves you a comment on the coding of the joystick location. The six variables arrive as function parameters and you use them however you wish. In Figure B.5 you can also see four drop list controls to the right of the joystick. I have already set them up to have something meaningful in them for the purpose of demonstration. These controls are generally used for two purposes:

1. They describe the current "meaning" of the apex lettering system; in Figure B.5 the A-corner is the "SINE" corner.
2. They allow the user to change the meaning of the apex in the plug-in by selecting different values.

You set up the drop list boxes when you are in development mode by clicking on them and choosing the item "Select To Setup," which allows you to connect these drop list boxes to different enumerated lists. They follow the same rules as the other enumerated lists, however, like the LCD control, you can use the underscore (_) to create a blank space in the string at run time. When the user selects an item from the box, the client calls your userInterfaceChange() function, passing it the parameters as described in the comment section for that method. You can then handle those messages as you wish.

Joystick Drop List Boxes	Index
Drop List A	60
Drop List B	61
Drop List C	62
Drop List D	63

The joystick program is an advanced user's control for automating movement of the joystick in reaction to some event. Please see the website for tutorials and sample code for using the program.

B.3 Using the sendUpdateGUI() Method

This is a perfect place to explain how to use the sendUpdateGUI() method. You can force the user interface (UI) (RackAFX or a custom graphical user interface [GUI]) to change its settings based on your internal variable values. This is opposite the normal flow where the GUI calculates and alters your variables. In this case, you alter your own variable and tell the GUI to update accordingly. This can be used to move sliders in real time or toggle radio buttons programmatically. The joystick drop lists present an interesting use of this method. Suppose you wanted to have four identical lists that represent the joystick corners, such as {A, B, C, D} = {SINE, SAW, TRIANGLE, NOISE} and the user will move the joystick to produce a mixture of low-frequency oscillators (LFOs) in your plug-in. You would like the lists to contain the same values, but you would like to initialize them with other values than the first one. For example, you might want the initial state of the boxes to show SINE for A, SAW for B, NOISE for C, and TRIANGLE for D (notice this is different than the order of the list too). When you set up the enumerated list, you give it a list variable to connect to (e.g., m_uApexA) and a list that goes along with that. By clicking "Use Existing String List," then double-clicking on any of the items in the list, you can legally share your string lists. This is something we did not cover in the book chapters because it is a more advanced feature. Then, in your code (prepareForPlay() or wherever you need to) you can alter your own unsigned integer-type (UINT) variables then call sendUpdateGUI() to show the changes. For example, to initialize the drop lists as described above, you would add this code to your prepareForPlay() method:

```
// set my variables
m_uApexA = SINE;
m_uApexB = SAW;
m_uApexC = NOISE;
m_uApexD = TRIANGLE;

// tell the GUI to change accordingly
sendUpdateGUI();
```

Theoretically, you can do this from any point in the code *except the constructor* (because the messaging pointer isn't assigned until after construction), even from the joystick handler if needed. However, you should never initiate client communication like this in the processAudioFrame() method as it relies on a slow window-messaging system. This is true for other plug-in formats as well. Remember that you can use the sendUpdateGUI() method to alter the GUI for any of the RackAFX 4 Plug-In connected variables.

B.4 Using GUI Designer

RackAFX has a powerful GUI designer that lets you arrange visually appealing GUIs in a matter of just a few moments. The GUI designer is always being updated with more features, so always check the latest additions at the website. Depending on which version you have, your GUI designer will look more or less like that shown in Figure B.8. The flow of operations is as follows:

1. In prototype view (the main RackAFX view), you assemble your plug-in. You create controls, give them min, max, and initial values, connect them to variables, and so on. Because this is a development mode, you will probably change some controls, add or remove them, and so on. Also, because we are prototyping, we can set up a bunch of controls we really would not want the user to be able to adjust in the final plug-in. An excellent example is the reverb plug-in; once we've got those parallel comb filters tuned, we don't want the user messing with them.
2. After your plug-in is complete, debugged, and ready to go, you click on the GUI Designer tab to reveal the blank GUI surface, shown in Figure B.8.
3. You drag and drop controls from the left side and arrange them however you like on the surface. Because they have transparent backgrounds, they can be overlapped.
4. For the slider, radio buttons, and knob controls, you must connect the control with the variable in your plug-in that it will control.
5. The LCD and vector joystick controls don't need any additional setup.

Suppose after testing we decide that the final GUI for the reverb project needs to have the following controls, while hiding all the others:

- Pre-delay time
- Pre-delay attenuation
- Bandwidth

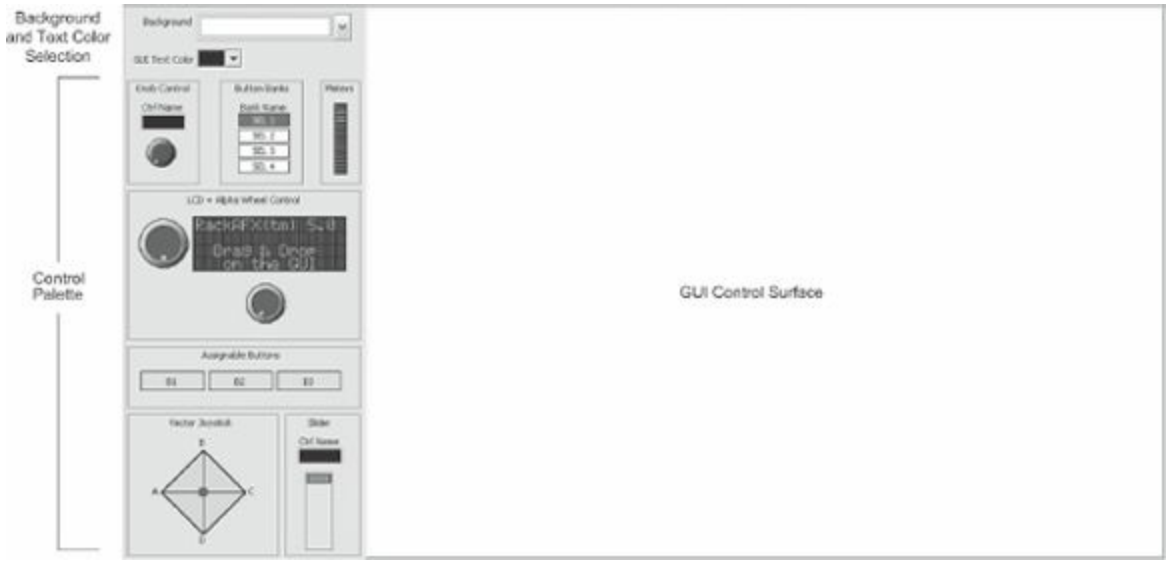


Figure B.8: The GUI Designer.

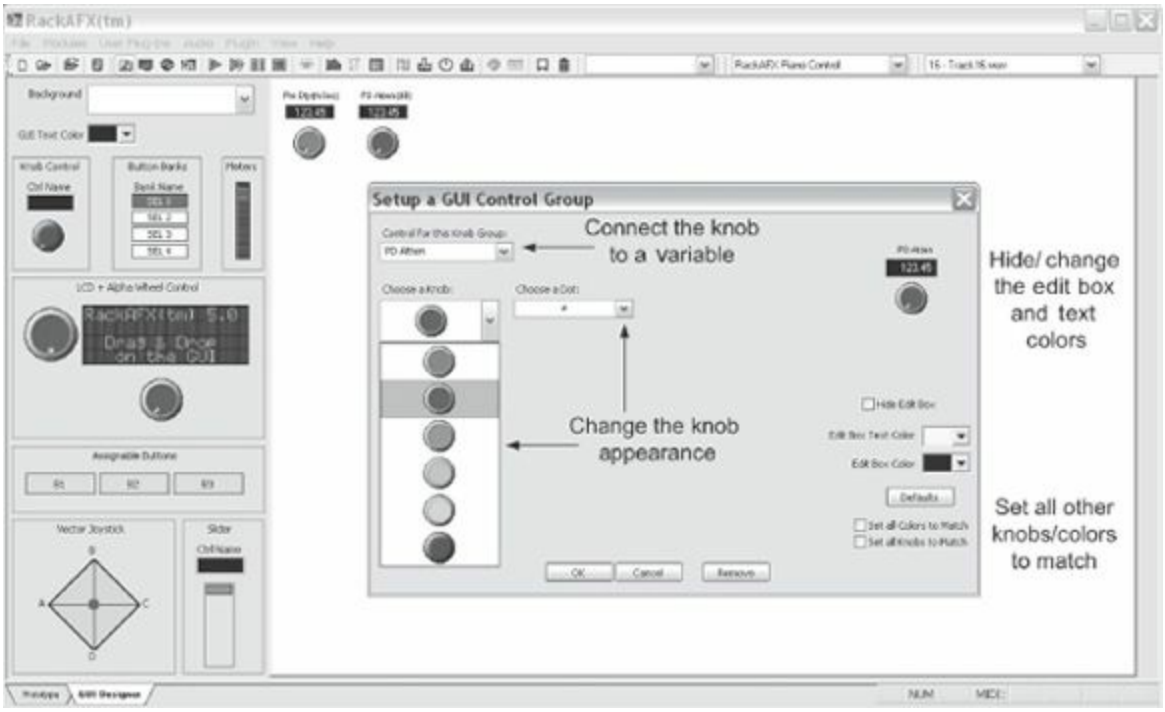


Figure B.9: Customizing a knob by right-clicking it.

- Damping
- RT60
- Wet/dry

Any of the prototype's sliders can be dragged and dropped either as knobs or sliders using the palette on the left. Figure B.9 shows the GUI designer after one knob control has been placed and mapped to the pre-delay time. A second knob has just been dropped next to it. Right-clicking on the knob itself pops up another customization box. Here you can do the following:

- Connect the knob to a variable via the drop list.
- Change the appearance of the knob and dot combination.
- Change the edit box behavior, hide it, or change its colors.
- Set all the other knobs to match this one.

You can do similar things for the radio buttons, meter, or the slider control, as shown in the pop-up boxes in Figures B.10 and B.11. The current GUI designer rules are as follows:

- For sliders, you link to a control variable, just like the knobs. You can customize many aspects of the slider control's appearance.
- When you run out of continuous controls (sliders) you can't drag any more knobs/slider controls.
- Each time you drag a knob or slider, you must set it up before dragging any other knobs or sliders.
- The joystick can have its colors altered and you can set the program button to look like the assignable buttons. There can be only one joystick.
- You can change the IN and OUT button bitmaps for the assignable buttons. There can be only one assignable button bank.
- You can add any continuous control (slider) to the GUI LCD group; it is a separate entity from the one on your main RackAFX GUI. Here I added bandwidth and damping. There can be only one LCD control.

- When you drag meter controls, RackAFX starts with the leftmost meter on the main GUI and automatically links your control variable for you. It then moves from left to right as you drag and drop meters. If you need to skip meters, add all the meters first, then remove the ones you don't want.
- When you add radio button controls, RackAFX starts with the topmost button bank on the main GUI and automatically links your control variable for you. It then moves from top to bottom as you drag the radio buttons. If you need to skip button banks, add all of them first, then remove the ones you don't want. The buttons banks are highly customizable.

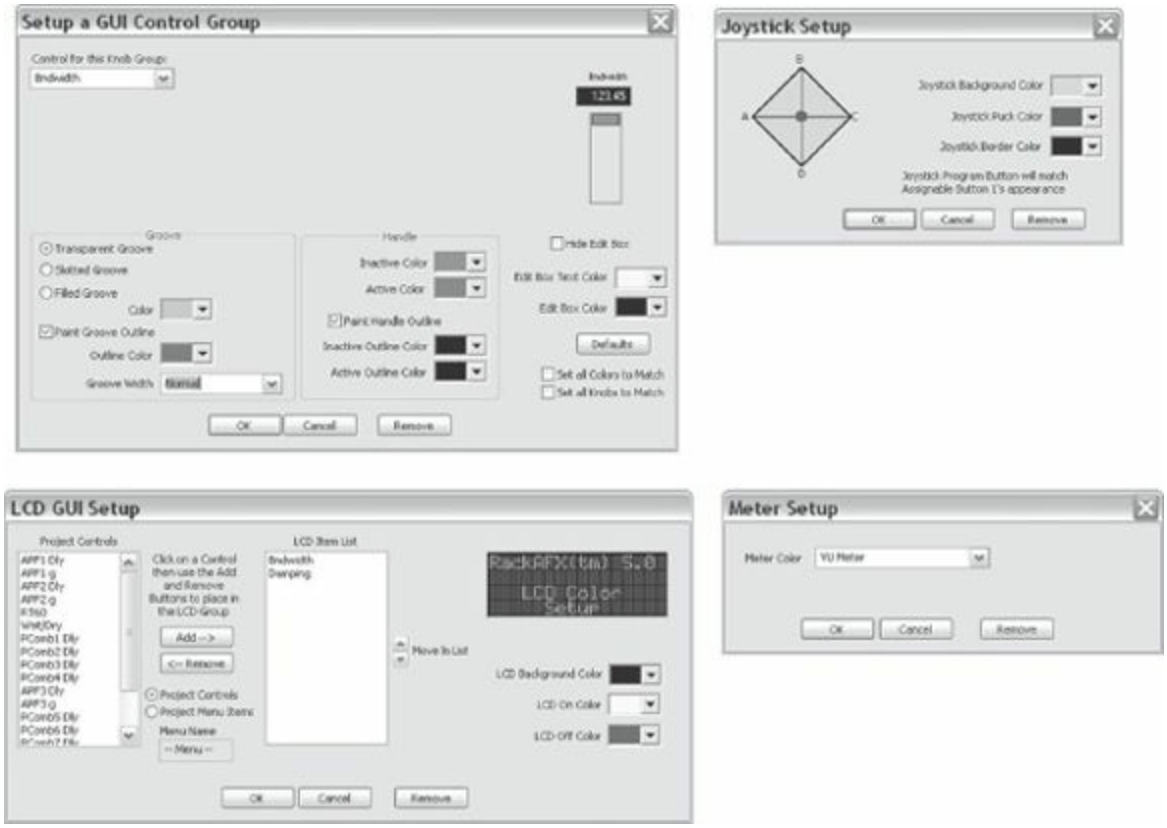


Figure B.10: The slider, joystick, LCD, and meter GUI customization dialog boxes.

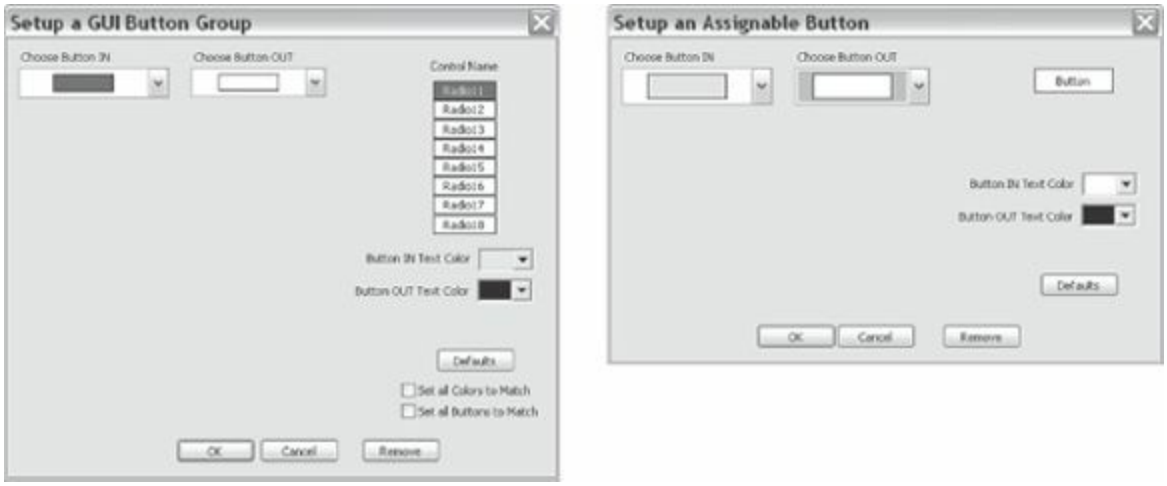


Figure B.11: The Radio and Assignable button GUI customization dialog boxes.

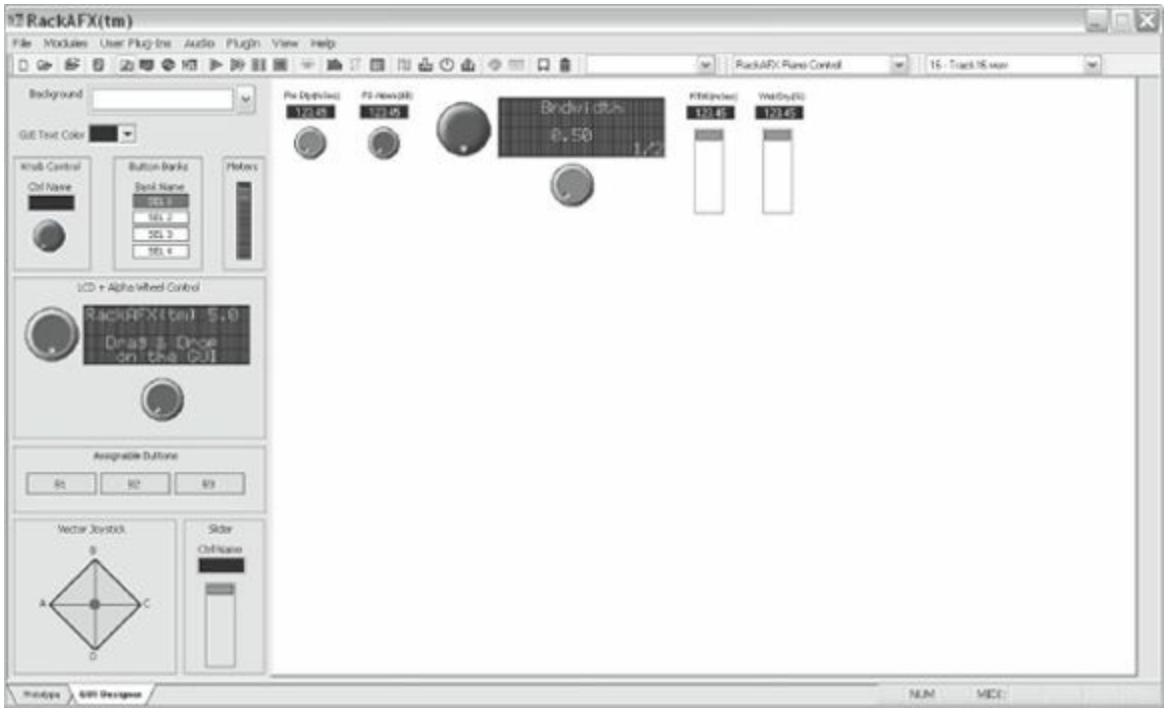


Figure B.12: The finished reverb GUI.

Figure B.12 shows my finished reverb GUI. I chose the white background just to make it easier to see in print. After you are done, go back to the prototype view and rebuild your plug-in. You need this rebuild to embed the new GUI information into your code. The GUI styles are coded into UINT arrays in your control variables as well as the plug-in object.

See the website for more details on the encoding. Now, load your plug-in. You will notice that the blue knob button on the toolbar is now active. You can open your GUI with that button or the menu item View S Custom Plug-In GUI. Figure B.13 shows the finished GUI. The designer automatically crops it to fit the controls you've placed on it. When you alter the new GUI controls, the RackAFX controls change as well (but not the other way around).

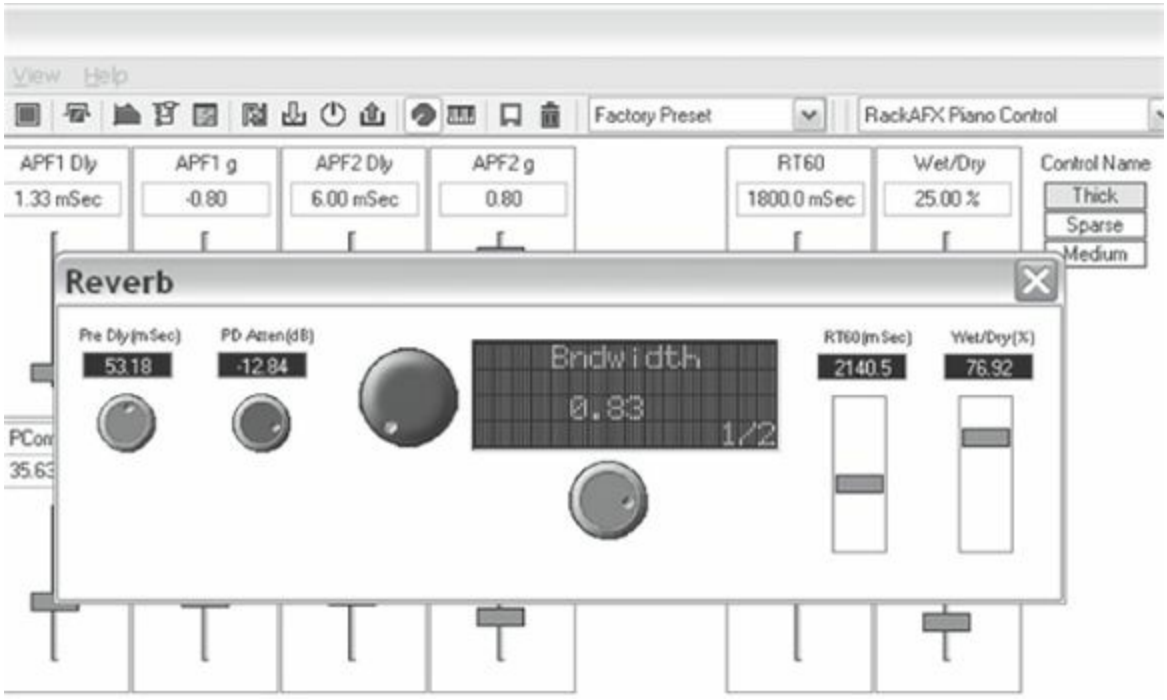


Figure B.13: The reverb GUI in action!

If you have enabled your plug-in to be compiled to produce a Virtual Studio Technology (VST) plug-in (Appendix A) then your RackAFX custom GUI will be available in any Windows VST client that supports custom GUIs (which is just about all of them). You launch it in whatever way the client digital audio workstation (DAW) requires.

Index

A

absorbent all-pass filters (AAPFs) 390
addition operation 17
address generation unit (AGU) 207
all-pass filter (APF) reverberators 368–71
alpha wheel GUI controls 519
analog-to-digital conversion 1, 170–8
anti-aliasing filters 3
applications programming interfaces (APIs) 27–9, 501
    functions typically required of 29–30
    and RackAFX philosophy 31–3
asynchronous operation 6
attack times of envelope detectors 428–9
attenuation function and attenuators 18–19
AU programming and processing 515–18
audio filter designs 253–87
audio specific filters 188–96

B

beats-per-minute (BPM) synchronization 235
bilinear transform 172–5, 179–80
bipolar/unipolar functionality 324–6
bi-quadratic filters (bi-quads) 157–62
    design of 181–8
    modified 189
Bohn, D. 196
book-keeping and book-keeping records 13–15, 114–15
Browne, S. 357
Butterworth filters 184–7, 475

C

C programming language 24
C++ programming language 24, 29, 31
Cartesian coordinate system 100–1
cascaded graphic filters 194–6

Chaigne, A. 385
Chemistruck, M. 390
child plug-in objects 239
chorus effect 331–4
circular buffers 209–10, 273
comb filter reverberator 364–8
comb filters 221
comparison of objects, GUIs and functions 505–6
complementary filter design 266–70
complex conjugates 133–4, 144
complex numbers 100–1, 112
complex sinusoid function 97–100
compressor/limiter plug-in design 457–65
compressors 453–6
conjugate poles of resonators 167–70
constant-Q filters 192–4
convolution plug-ins 273–81
convolution process 254–8, 274
cooking functions 57, 227, 230–1

D

Dahl, L. 390
Dattorro, J. 381
Dattorro plate reverb 382–5
Decca chorus 354
delay effects 207–9
    modulated 327–54
delay operators 16
delaying all-pass filter reverberators 368–71
difference equations 18
    z transform of 117–18
digital audio technology, history of 1
digital delay line (DDL) 209–15, 327
    advanced form of module 248
    module plug-in 224–33
digital signal processing (DSP): address generation unit 207
    basic theory of 97–122
    chips 4
    filters 71–95

practical application of theory 123–62
test signals 10–12
digital-to-analog converters (DACs) 3
digitally-controlled amplifiers/attenuators (DCAs) 454
dimension-style choruses 351–3
direct *z*-plane design techniques 163–4
Doppler effect 328
downward expander/gate plug-in design 466–8
dynamic link library (DLL): access to 22
building of 35
dynamic linking 21–2
dynamic processing 453–87

E

early reflections 358
echoes 360
eigenfrequencies 361
energy decay relief (EDR) plots 362–4
enumerated slider variables 245–8
envelope detection 426–36
envelope follower plug-in design 425–8
Euler's equation 98–9, 104–7, 112, 138–9, 148, 152, 155, 215, 221
expanders 453–6

F

feed-back filters: first-order 88–95, 123–4
second-order 142–9
feed-forward filters 119
first-order 74–88, 94–5, 103–6
second-order 132–41
feedback 221–2
effect on impulse response 218–19, 222–3
feedback delay network (FDN) reverbs 385–91
generalized 385–9
finite impulse response (FIR) filters 161–2, 253, 262, 266–7, 281–7
flanger/vibrato effect 328–31
flanger/vibrato/chorus plug-in 334–42
floating-point data 9–10
fractional delay 235–9

Frenette, J. 381–2
frequency response: direct evaluation 127–30
estimation 121, 133–8, 143–7, 151–2, 158–60
frequency response plots 71–2, 83–5, 104, 111
frequency sampling method 263–6
full-wave envelope detectors 428

G

gain control 122–3
gain function 18
Gardner, W. G. 357, 366–8, 377–81
Gardner's nested APF reverberators 377–81
gates 453
genetic algorithm (GA) 390–1
Gerzon, M.A. 377
Gordon–Smith oscillator 299–301
graphical user interfaces (GUIs) 25–8, 35
for DDL Module 225–6
Griesinger, D. 358, 360

H

half-wave envelope detectors 428
hard-knee compression curves 456

I

impulse response: effect of feedback on 218–19, 222–3;
see also room impulse response infinite impulse response (IIR) filters 161–204, 207, 253
interpolation functions 236–7
interrupt-based designs 6–7
inverse comb filtering 217–18

J

Jot, J.-M. 385–90

K

knee of the compression curve 456
Korg Triton 250–1, 345, 350
Kuttruff, H. 361–2

L

Lagrange interpolator 284
late reverberations 358
limiters 453
linear interpolation 236–8
linear phase filters 83, 266–7
Linkwitz–Riley filters 186–7, 475, 479
liquid crystal display (LCD) GUI controls 519
look-ahead compressor plug-in design 468–71
low-frequency oscillation (LFO) 328, 331
low-pass filter–comb reverberator 373–5

M

Mac programming 501, 503, 506, 515
McGrath, D. S. 357
Marcolini, K. 390
Massberg, Michael 201
Massberg analog-matched low-pass filter 201–4
median filter algorithm 284–7
member objects 224
mixing algorithms 19
mod filter plug-in design 412–25
modal density 361–2
modes (room resonances) 360–1
modulated APF reverberator 381–2
modulated delay effects 327–54
modulated delay line (MDL) 327
modulated filter effects 411–51
Moorer, J. A. 375, 377
Moorer's reverberator 365–71
moving average (MA) interpolator 282–3
multi-tap delay 249–50
multi-unit LCR chorus plug-in 345–50
multiplication operation 16–17

N

National Semiconductor 438
negative frequencies 104–5, 113
nested filter structures 379–81

normalized frequency 104
numerical coding of audio data 7–9
numerical method FIR filters 281–7
Nyquist frequency 1–2, 11–12, 76–7, 95

O

one-sample delay 15–16
optimal method for designing filters 270–3
order of a filter 114
Orfanidis, S. 429, 455
oscillators 289–326
    bipolar/unipolar functionality 324–6
    direct-form 289–99
    wave-table type 301–8

P

panning *see* tremolo/panning plug-in design
parametric filters 191–4
parent plug-ins 233, 239
Parks–McClellan algorithm 262, 270
peaking filters *see* parametric filters phase inversion function 18
phase response plots 71–2, 83–5, 101, 104, 111
phaser plug-in design 436–45
ping-pong delay 250
plate reverb algorithm 382–5
plug-in, definition of 21
polar notation 100–1
pole-zero filters 149–57
poles of the transfer function 124–32, 145
poles or zeros at infinity, effect of 178–80
polynomials, factoring of 119–20
pre-delay 358
Puckette, M. 385, 388
'pumping and breathing' artifacts 463

R

RackAFX software 29–70
    design of tone control plug-ins 58–69
    design of volume control plug-ins 40–1, 54–8

destruction of plug-in objects 38
dynamic link library 35–6
FIR filter designer 262
frequency sampling method tool 267–9
graphical user interfaces 36–7, 525–30
impulse convolver 258–62
optimal method tool 271–3
processing audio 37–8
setting up plug-ins for use and setup preferences 41–54
stock objects 394–8
testing 38–40
user menu 69–70
reconstruction filters 3
Reilly, A. 357
Reiss, J. 455, 486
release times of envelope detectors 428–9
Remez exchange algorithm 271–3
resonant LPF plug-in 196–201
resonators 165–70
reverb algorithms 357–409
reverb modules 363
reverb time 359–60
ring modulator plug-in design 494–6
Roland Dimension D chorus 351–3
room impulse response 358–60
room resonances *see* modes
room reverb: design of 398–408
example of 391–3
RT_{60} reverb time 359–60

S

Sabine, W. 359–60
samples of audio signal, acquisition of 1–3
sampling theorem 2, 7, 106
saw-tooth waveform 317–18
scalar multiplication operation 16–17
scattering matrices 389–90
Schroeder, M. 357, 360–4, 368, 370, 376–7, 385
Schroeder's reverberator 372–3

‘sendUpdate GUI()’ method 525
shelving filters 156–7, 189–90, 201
sigmoid functions 497
signal-processing algorithms 13
signal-processing flow 6–7
signal-processing systems 4–5
single pole filters 164–5
slider controls 245
Smith, Dave 522
Smith, J. O. 389
Smith-Angell improved resonator 168–70
soft-knee compression curves 456–7
software development kits (SDKs) 501
Sony DPS–M7 354
spectral compressor/expander plug-in design 475–86
square wave form 317, 319
static linking 21–2
Stautner, J. 385, 388
stereo crossed feedback delay plug-in 244
stereo digital delay plug-in 239–44
stereo-linking a dynamics processor 472–4
stereo phaser with quad-phase LFOs 446–51
stereo quadrature flanger plug-in 342–5
stereo reverberation 376–7
subsequent reverberations 358
subtraction operation 17
synchronous operation 5–6

T

through-zero flanging (TZF) 330
time delay as a mathematical operator 102
transfer functions 101, 104
    evaluation of 106–12
transverse delay line 257
tremolo/panning plug-in design 489–94
trivial zeros 125

U

unitary feedback delay networks (UFDNs) 386, 390

V

vector joystick control 519–24
vibrato *see* flanger/vibrato/chorus plug-in;
    flanger/vibrato effect virtual address space 22–3

W

wave shaper plug-in design 497–500
wave table oscillators 301–8
wave tables: addition of 308–12
        band-limited 312–19
waveguide reverberators 389
wet/dry mix 214–15
white noise 360
'willpirkle.com' website 519
Windows operating system 501–3
wrapping of plug-in objects 503–5, 512–14

Z

z substitution 114
z transform 114–21, 133, 141–2
        of difference equations 117–18
        of impulse responses 118–19
        of signals 116–17
        zero frequencies 119–20, 145
Zöler, U. 429, 455

Monograph Series in

UNDERWATER ACOUSTICS

Sponsored by Office of Naval Research

Editorial Board: Jeffrey A. Simmen, Ph.D., Chair
Homer P. Bucker, Ph.D.
Ira Dyer, Ph.D.
Finn B. Jensen, Ph.D.
Ellen S. Livingston, Ph.D.

For further volumes:
http://www.springer.com/series/5535

Monograph Series in

UNDERWATER ACOUSTICS

In 2001 the Office of Naval Research began a program to publish a selected monograph series in the field of underwater acoustics. The series publishes in-depth reviews and analysis on the state of understanding of the physics of sound in selected areas of research.

The monograph series covers the most important topics in underwater sound in an in-depth, critical, scholarly, and comprehensive fashion. It is intended to provide researchers with an understanding of the physics of underwater sound, based on both theory and experiment. The discussion of important unanswered questions is encouraged. Topics of particular importance in the field are low-frequency sound in the deep ocean, shallow-water acoustics, and ambient noise.

Sponsored by:

Office of Naval Research
One Liberty Center
875 North Randolph Street, Suite 1425
Arlington, VA 22203-1995
E-mail: onrpao@onr.navy.mil

Series Editors:

Jeffrey A. Simmen, Ph.D.
Applied Physics Laboratory
University of Washington
1013 NE 40th Street
Seattle, WA 98105-6698
E-mail: simmen@apl.washington.edu

Homer P. Bucker, Ph.D.
Space and Naval Warfare Systems
Center, retired
808 Moana Drive
San Diego, CA 92106
E-mail: bucker@nosc.mil

Ira Dyer, Ph.D.
Weber Shaughness Professor
of Ocean Engineering, Emeritus
Massachusetts Institute of Technology
Cambridge, MA 02139

Finn B. Jensen, Ph.D.
Emeritus Scientist
NATO Undersea Research Centre
19126 La Spezia, Italy
E-mail: jensen@nurc.nato.int

Ellen S. Livingston, Ph.D.
Associate Director
Ocean and Undersea Science
U.S. Office of Naval Research Global
86 Blenheim Crescent
West Ruislip, Middlesex HA4 7HB
United Kingdom
E-mail: ellen.livingston@onrg.navy.mil

Boris Katsnelson · Valery Petnikov
James Lynch

Fundamentals of Shallow Water Acoustics

Springer

Boris Katsnelson
Voronezh State University
Universitetskaya Square 1
Voronezh 349006, Russia
katz@phys.vsu.ru

Valery Petnikov
A.M. Prokhorov General Physics Institute
Russian Academy of Sciences
Moscow 119991, Russia
petniko@kapella.gpi.ru

James Lynch
Woods Hole Oceanographic Institution
Woods Hole 02543, MA, USA
jlynch@whoi.edu

ISBN 978-1-4419-9776-0 e-ISBN 978-1-4419-9777-7
DOI 10.1007/978-1-4419-9777-7
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011944833

© Springer Science+Business Media, LLC 2012
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.
The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Series Preface

The efficacy of sound to penetrate the seas made acoustic systems in the past century the leading tools for sensing objects in and measuring properties of the seas. For over 60 years, the US Office of Naval Research (ONR) has been a major sponsor of undersea research and development at universities, national laboratories, and industrial organizations. Appropriately ONR is the sponsor of this monograph series.

The intent of the series is to summarize recent accomplishments in, and to outline perspectives for, underwater acoustics in specific fields of research. The general field has escalated in importance and spread broadly with richness and depth of understanding. It has also, quite naturally, become more specialized. The goal of this series is to present monographs that critically review both past and recent accomplishments in order to address the shortcomings in present understanding. In this way, these works will bridge the gaps in understanding among the specialists and favorably color the direction of new research and development. Each monograph is intended to be a stand-alone advanced contribution to the field. We trust that the reader will also find that each is a critical introduction to related specialized topics of interest as well.

ONR has sponsored the series through grants to the authors. Authors are selected by ONR based on the quality and relevance of each proposal and the author's experience in the field. The editorial board, selected by ONR, has, at times, provided independent views to ONR in this process. Its sole official role, however, is to judge the manuscripts before publication and to assist each author at his request through the process with suggestions and broad encouragement.

Jeffrey A. Simmen, Ph.D.
Homer P. Bucker, Ph.D.
Ira Dyer, Ph.D.
Finn B. Jensen, Ph.D.
Ellen S. Livingston, Ph.D.

Preface and Acknowledgments

Shallow water acoustics is a very active area of underwater acoustics research, as the continental shelves and slopes have great economic, social, and military importance to humans, and all these areas interact with ocean sound. However, books dedicated to this specific topic are in short supply, and indeed the book "Shallow Water Acoustics" published over two decades ago in Russian by the two Russian authors of this present book was the first volume specifically dedicated to this branch of underwater acoustics. The authors then had this book translated into English, and it represented the state of the art up to about 1995. However, a *huge* amount of research has been done in this area in the last 15 years, and so we thought that it would be a good thing for the three of us to collaborate on a book that would update the coverage of the field, as well as point out more strongly where we thought the future might lie. As the Office of Naval Research (ONR) was publishing a book series on underwater sound, and was amenable to the idea, we signed on with them and produced this book, guided by a technical Editorial Board that reviewed and critiqued our work.

It may seem strange to some that a book on underwater sound was written jointly by an American and two Russians, but in fact it is a rather natural development given the history of the last 50 years. During the Cold War era, both the USA and Russia developed considerable expertise in underwater sound, and had large communities working in the area. And though (obviously) any transfer of classified work was and is strictly prohibited by both countries, the transfer of basic research knowledge was and still is fairly open. Indeed, American researchers studied the works of the famous Russian acoustician Leonid Brekovskikh, while Russian researchers were just as familiar with the textbooks by Bob Urick and Clarence Clay. When the Cold War subsided in around 1990, actual meetings of Russian and American investigators became common, and indeed that is when the three authors of this book first met, at a port call of the Russian research vessel Akademik Sergey Vavilov in Boston, MA. What struck us all was just how much we had in common scientifically – we spoke one common language well, even if the Russians' English was poor and the American's Russian nonexistent. We were excited about the basic

acoustics science, and as the international doors had opened sufficiently, we started to collaborate.

In the post-Cold War era, Russian support for experimental ocean acoustics dropped precipitously, and seagoing efforts became few and far between. However, a strong community with a good understanding of experiments and a sterling knowledge of theory existed, and started to look outwards for collaboration and funding. US funding also dropped some, but ONR still maintained a vibrant ocean acoustics program, including a strong experimental and technological component. Given the data coming out of the US program, and the analysis capabilities of the Russian researchers, a natural partnership was possible, and indeed came to fruition in many cases, not just for this book. This bit of technological symbiosis has been good for all concerned, and we fervently hope it will continue.

Regarding the book itself, we have written a book that is intended for graduate students, postdocs, and more advanced researchers, that is also a cross between a research monograph and a textbook. As there are few textbooks in this area (see the above remarks), we thought that covering the elements of the field, but at an advanced level, would be useful On the other hand, we also wanted to provide a monograph that showed the state of the art in research, and also tried to point the way for future efforts. This made the book a bit of a balancing act, and we hope that it has succeeded.

We also have couched much of the physics of this book in terms of the modal picture, which we think is the most natural one for shallow water. That is not to say that we have avoided ray theory or WKB, but that our emphasis has been on lower frequency modal work. We also note that some of the physics can be explained in the ray picture as well as the mode, and indeed one of our Editorial Board members, Dr. Ira Dyer, is a strong advocate of looking at things that way. Our feeling is that, once you have understood the physics in one picture or the other, you can understand it both ways – the picture translation is not so hard.

One might also notice a bias in the experimental data discussed to the Barents Sea and the East Coast of the USA. This is because these are the places that were most frequented experimentally by the authors and their home Institutions (Voronezh State University, the General Physics Institute and the Woods Hole Oceanographic Institution). We are most familiar with the data from these areas, and so naturally used them first for examples.

Next, you will note in reading this book that it speaks with both a Russian and an American (New Jersey) accent, depending on the section. We have tried hard to make things grammatically correct, and clear as well, but the "linguistic lilt" of the various sections we left intact. As all of us are fans of the great physicist Richard Feynman, we think that speaking in your own voice is no crime, and has a noble precedent. Feynman would have contended, as we do, it is what you say about the science, and not the regional accent that counts.

As with any larger piece of work, many people were involved and contributed to its completion. From the Russian side, the authors would like to acknowledge Fyodor V. Bunkin, Yuri A. Kravtsov, and the many other colleagues from the A.M. Phokhorov General Physics Institute and the Voronezh State University

whose work in the execution of large-scale experiments on the shelf of the Barents Sea and the interpretation of the results was invaluable. From the American side, the author also warmly acknowledges the numerous colleagues with whom he has gone to sea, analyzed data, and argued over what it all means at meetings, workshops, seminars, and other venues. Regarding funding, all the authors give their heartfelt thanks to the US Office of Naval Research and the Robert W. Morse Senior Scientist Chair at WHOI, both of which paid for the majority of this work. From the Editorial side, the efforts of the Editorial Review Board (Drs. Ralph Goodman (deceased), Ira Dyer, Jeffrey Simmen, and Finn Jensen have been enormously helpful. The enthusiasm and project management of Dr. Ellen Livingston is also warmly appreciated. Also, our great thanks to Ms. Sheila Hurst for her editorial work and skills, and Ms. Gretchen McManamin for her administrative assistance.

Finally, our thanks to our wives and families, who put up with our often erratic and travel heavy oceanographer lifestyles, which are prerequisites to writing such a book. The only *really* hard thing about being an oceanographer is being away from such wonderful families for long periods of time.

Voronezh, Russia Boris Katsnelson
Moscow, Russia Valery Petnikov
Woods Hole, MA, USA James Lynch

Contents

1 What Is Shallow Water Acoustics? 1
1.1 Deep Versus Shallow 1
1.2 Past and Present of Shallow Water Acoustics 4
1.3 The Future of Shallow Water Acoustics 10
1.4 Some "Old Favorite" Research Areas Revisited and Updated 14

2 Coastal Oceanography, Geology, and Biology 17
2.1 The Coast as Acoustic Waveguide 17
2.2 Properties of Sea Water: Vertical Stratification and Its Seasonal Variability 18
2.3 Horizontal Stratification and Its Variability: Fronts and Eddies, Surface Ducts, and Storm Surges 23
2.4 Dynamics of the Ocean Surface: Surface Waves 32
2.5 Dynamical Processes Inside the Ocean: Tides and Internal Waves 34
2.6 Experimental Studies of Coastal Internal Waves 43
2.7 Coastal Geology and Geophysics 51
2.8 Acoustics of Sediments 55
2.9 Bottom Roughness 63
2.10 Solid and Multicomponent Layered Bottom Models 65
2.11 Acoustics of Biological Objects in a Coastal Area 70

3 Foundations of the Theory of Propagation of Sound 79
3.1 Field of a Point Source in a Layered Waveguide with Absorbing Boundaries 79
3.2 The Pekeris Model 86
3.3 Perturbation Theory and WKB Methods 94

3.4 Ray Description of the Sound Field and Ray-Mode Connections 102
3.4.1 Ray Theory 102
3.4.2 Rays as Interfering Modes 105
3.4.3 Modes as Interfering Rays 107
3.4.4 Distinguishing Between Ray and Mode Arrivals 109
3.5 Mode Coupling in a Shallow Water Waveguide with Small Inhomogeneities 113
3.6 Mode Coupling in Shallow Water with Smooth Inhomogeneities 118
3.7 Horizontal Refraction in Shallow Water (The 3D Problem) 124
3.8 Parabolic Equation 138

4 Examples Illustrating the Characteristics of Waveguide Propagation 145
4.1 A General Transmission Loss Example, Including a Simplified Theory of Shallow Water Attenuation 145
4.2 Simplified Description of Solid, Multicomponent, Poro-Elastic, and Layered Bottom Models 157
4.3 Optimum Frequency 161
4.4 Interference Structure and Interference Invariant of the Sound Field 165
4.5 Waveguide Dispersion of Sound Signals in Shallow Water 171
4.6 Averaged Description of the Sound Field in a Waveguide 181
4.6.1 Channel with Constant Sound Speed and Variable Depth 184
4.6.2 Irregular Near-Bottom Sound Channel 187

5 Sound Field in Shallow Water with Random Inhomogeneities 193
5.1 Structure and Models of Different Kinds of Random Inhomogeneities 193
5.2 Description of a Random Sound Field by Coherent and Incoherent Parts 196
5.3 Equations for Mode Interactions 198
5.4 Equations for Intensity 204
5.5 The Diffusion Equation and Averaged Decay Laws 209
5.6 Some Examples 212
5.6.1 Waveguide of Constant Depth 212
5.6.2 Irregular Waveguide 216
5.6.3 Dependence of the Intensity on Depth 219

5.7 Sound Field Fluctuations in the Presence of Background Internal Waves 222
5.8 Models and Statistics of Intensity Fluctuations 230
5.8.1 Intensity Fluctuations: Stochastic Modeling..... 231
5.8.2 Experimental Measurements and the Statistics of Intensity Fluctuations 233
5.8.3 Broadband Fluctuations..... 236

6 Low-Frequency Bottom Reverberation in Shallow Water 239
6.1 Introduction: Sound Backscattering by the Sea Bottom 239
6.2 Mode Theory of Bottom Reverberation in a Regular Waveguide 244
6.3 Numerical Simulation of Low-Frequency Bottom Reverberation..... 250
6.4 Long-Range Reverberation Studies Using Extended Arrays 253
6.5 Long-Range Reverberation in a Randomly Inhomogeneous Waveguide..... 259

7 The Inverse Problem 267
7.1 General Considerations: The Linear Inverse Problem 267
7.2 Bottom Versus Water Column Influences on the δq_l and Inverse..... 272
7.3 The Generalized Inverse Solution, Its Error, and an Example from the Corpus Christi..... 278
7.3.1 GEMINI Experiment..... 278
7.4 Examples of Nonlinear Bottom Property Inversion Using Various Data Types..... 284
7.4.1 Horizontal Interference Structure and Bottom Parameters Estimation 285
7.4.2 Using the Vertical Interference Pattern for Bottom Parameter Estimation..... 288
7.5 Broadband Experiments and the Frequency Dependence of the Bottom Parameters..... 290
7.6 Bottom Geoacoustic Inversions Using Ambient Noise Sources 298
7.6.1 Inversion for Bottom Properties from Ship Self-Noise..... 299
7.6.2 Inversion for Bottom Properties from Random "Ship of Opportunity" Noise..... 301
7.6.3 Inversion for Bottom Properties from Random Surface Noise 304

8 Signal Processing 305
8.1 Fundamentals of Data Processing Techniques 305
8.2 Matched Field Processing in Shallow Water 313
8.2.1 Conventional MFP, Bartlett Beamformer 316
8.2.2 Two (Analytic) Examples of the Parameter Ambiguity Function 317
8.2.3 Maximum Likelihood Beamformer 319
8.2.4 Variable Coefficient Likelihood Beamformer 320
8.2.5 Comparison of Processors for Simple Examples 321
8.3 Spatial Coherence of the Sound Field in Shallow Water and Array Signal Gain 324
8.4 Mode Filtration 334
8.5 Time-Reversal Mirror of the Sound Field in Shallow Water 348
8.6 Acoustic Uncertainty 358

9 Noise Field in Shallow Water 363
9.1 Introduction. Model of a Noise Source: General Equations 363
9.2 Relationship Between the Continuous and the Discrete Components of the Field of Surface Noise Sources in a Waveguide 374
9.3 Effect of the Frequency Dependence of the Bottom Absorption Coefficient on the Noise Spectrum 377
9.4 Vertical Directivity of the Sound Field and the Effect of Random Inhomogeneities 381
9.5 Effect of the Sound Speed Profile on the Vertical Distribution of the Noise Field Intensity 391

10 Equipment for Shallow Water Acoustics and Experimental Considerations 397
10.1 The Frequency Band Used for Large-Scale Acoustic Monitoring 397
10.2 Low-Frequency Sources of Sound 399
10.3 Receiving Acoustic Arrays and the Design of Large-Scale Acoustic Monitoring Experiments on the Shelf 413

11 The Future 425
11.1 Introduction 425
11.2 Physical Oceanography 425
11.3 Bottom Acoustic Properties 428
11.4 New Directions in Theory and Modeling of the Sound Propagation 430
11.4.1 3D Problems 430
11.4.2 Interference Invariant 430
11.4.3 Dislocations of the Wave Field 431

11.5 Physical Phenomena, Concerned with Stochastic Effects 431
11.6 Inverse Problems 432
11.6.1 Water Layer..... 432
11.6.2 Bottom Layer 433
11.7 Signal Processing 434
11.8 New and Prospective Developments in Oceanographic and Acoustical Equipment and Survey Methods..... 435
11.8.1 AUV, Gliders, Etc. and the Construction of Detailed Oceanographic Models 435

Appendix A Waves and Signals 439

Appendix B Modal Decomposition of the Sound Field from a Point Source (Green Function) 473

Appendix C Mode Coupling Equations..... 481

Appendix D Empirical Orthogonal Functions 487

Appendix E Scattering at Localized Inhomogeneities in the Waveguide. Approximate Description 493

Appendix F Reflection of a Plane Wave from Half-Space 499

Main Provisional Notation..... 503

References..... 509

Abbreviations 533

Index..... 535

Chapter 1
What Is Shallow Water Acoustics?

1.1 Deep Versus Shallow

To many ocean acousticians, shallow water is "water a few acoustic wavelengths in depth, where the normal mode description of the sound field is efficient." To some of our physical oceanographer friends, shallow water is taken as the portion of the sea that extends from the shore to the continental shelf break. A rather jaundiced geologist once described the entire water column (including shallow water) as "a bothersome thin layer of fluid that obscures the really interesting part of the ocean." All these definitions have some merit (including the last, if you are a geologist), but they are *not* the definition we will use. Rather, we will be looking at the region from the end of the surf zone out to the continental shelf break (and even onto the slope to ~500 m depth) as our working definition of shallow water. This is done for pragmatic reasons, having to do with the types of sonar systems that work for given purposes in those depths. Acoustically, we will be looking at sound and sonar systems working from ~50 Hz up to about 5 kHz. Incorporating these limits on depth and frequency bounds our technical area of "shallow water acoustics," but even with these limits, the field is a rather vast one.

Having stated what we will be examining, we next answer the "graduate student's Nemesis question" – why bother with this topic at all?! The primary answer is quite simple. The roughly 70 years of serious research that has been devoted to shallow water acoustics (since just before World War II) have been largely motivated by naval defense issues. Anti-submarine warfare (ASW) and mine warfare (MWF) have been particularly important topics in shallow water. But defense, the major application, is not at all the only application of shallow water acoustics at the present. Basic science research questions in biology (marine mammals and fish), geology (seabed properties and mapping), and physical oceanography (temperature and current measurements) are all being pursued using shallow water acoustic techniques. Commercial and industrial applications (shipping and oil exploration) use underwater sound in shallow water. Understanding anthropogenic noise pollution in the coastal ocean is also a direct use of

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_1,
© Springer Science+Business Media, LLC 2012

shallow water acoustics technology. So there are plenty of valid reasons to pursue this topic, in addition to its intrinsic technical fascination (which we also hope to convey).

Another first-order question worth asking is: "Why would a book be devoted specifically to *shallow water* acoustics? Surely a general ocean acoustics book can cover the ocean, whether deep or shallow?!" The answer here is again a simple one – No! The shallow water environment is remarkably different from the deep water environment. Once one gets past the initial step of writing down the basic wave equation, then the boundary conditions, the sound speed profile, the medium inhomogeneities, and many other factors become drastically different. Indeed, shallow water is a far more *complicated* acoustic medium than deep water (at least in our opinions), and is richer in phenomena to explore and exploit. Let us look at some examples.

The first, and most important, difference between deep and shallow water acoustics is the boundaries, i.e., the surface and bottom. In shallow water, one generally has to take into account the interaction of the sound with the surface and/or the bottom, whereas in deep water, one very often avoids acoustic interaction with these boundaries. The surface boundary, the one which is either the well-behaved friend or the feared foe of sea-going oceanographers, is perhaps the simpler and better understood of the two boundaries – but it is still anything but trivial. On a flat-calm day, the sea surface *is* a trivial pressure release surface boundary condition (give or take 15 p.s.i. of air pressure) and easy to deal with, physically and theoretically. On a rough day, however, the sea surface now presents a wind-fetch-dependent frequency-directional roughness spectrum, underlain by a seething layer of bubbles and bubble clouds. And then there is ice to consider if one works in the polar regions. Modeling this boundary condition (which one usually cannot escape, as opposed to many deep-water scenarios) is definitely *non*-trivial. The ocean bottom presents even more of a challenge, since it is both less accessible and (without specialized tools) impenetrable. Moreover, since the water column sound speed profiles are often downward refracting in shallow water, the bottom is an almost unavoidable boundary condition for sound propagation. In addition to a surface roughness profile akin to the one seen in sea surface scattering, the bottom presents irregular bathymetry, sub-bottom inclusions, variable materials, and all the fascinating layering and internal structure that coastal geology can provide. Indeed, the geological structure of the bottom is a vast field of study in its own right and far more than just a boundary condition for acoustic uses!

Just as the surface and bottom contributions in shallow water differ from those deep water, so does the physical oceanography. Rather than the plain-vanilla ocean one might expect from the well-known "Pekeris model" (Pekeris 1948; see Fig. 1.1 for a portrait photo of Chaim Pekeris) of a shallow water waveguide (an isovelocity water column over an isovelocity bottom halfspace), the coastal ocean abounds with phenomena creating complex range, depth, and time-dependent sound speed profiles. Coastal fronts, eddies, internal waves (linear and nonlinear), river buoyancy plumes, the turbulent edges of western boundary currents, and other

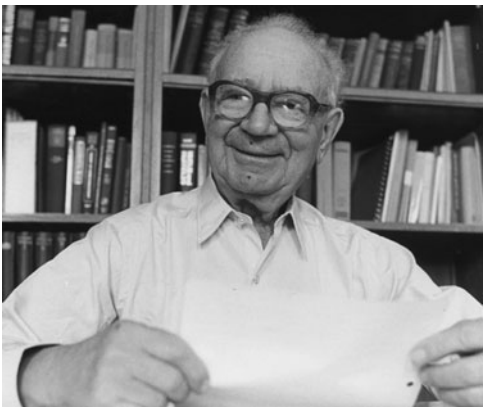


Fig. 1.1 C. Pekeris, 1908–1993

phenomena abound and create structure on a variety of scales – some small enough to create challenges to measurement and modeling, but still large enough and strong enough to affect significantly acoustics in the frequency range we are interested in. Many of these ocean phenomena are specific to shallow water, and the rest are certainly modified by the shallowness of the seas, so as to be distinct from deep water.

Not to slight biology, we should mention fish and marine mammals. In deep water, one detects some ambient noise from whale and dolphin vocalizations, but the density of fish and marine mammals is low overall so that the animals do not appreciably produce, absorb, or scatter sound. However, shallow water is the primary habitat for most marine life, and in addition to making more noise, fish schools can significantly absorb and scatter sound, as has been shown by both theory and experiment (Diachok et al. 2004; Makris et al. 2006).

There are acoustic differences between deep and shallow water, as well as physical environmental ones. The two main differences are (1) stronger reverberation and attenuation and (2) stronger 3D effects in shallow water (both random and deterministic ones). First, as already mentioned, in shallow water, the boundaries are close and so one is almost assured of interacting with them. Thus backscattering, reverberation, and bottom attenuation are more prominent. (That is not to say that one does not ever see such reverberation in deep water. As an example, deep-water upward refracting profiles such as those in polar regions or in the Mediterranean will force sound to interact with the surface, and so create reverberation.) Second, there are 3D acoustic effects. In deep water, only the bathymetry of seamounts and islands is a serious 3D acoustic concern, a very limited bit of the ocean's area. In shallow water, the slope of the shelf, the shelf break and the canyons can have significant 3D acoustic effects. Moreover, horizontal refraction by the water column itself can also be appreciable, as has been seen recently by both theory and experiment. Indeed, shallow water acoustics has created a small revival in 3D acoustic work, which was largely ignored for many years.

1.2 Past and Present of Shallow Water Acoustics

This book is, in general terms, about where we have been, where we are, and where we believe we are going in shallow water acoustics. Though more complete details will be presented in later chapters, it would be helpful here to present a glimpse into what we will be discussing, both as a roadmap and (more importantly) as motivation. To do this, we will look briefly at the "world line" of ocean acoustics from past to future, with focus on the aspects of theory, experiment, modeling, and prediction.

The Past (*up to early 1970s*). If you ask any practitioner of ocean acoustics when shallow water acoustics was "born" as a distinct subdiscipline of ocean acoustics, nine of ten would likely answer "in 1948, with the publication of Chaim Pekeris' seminal paper on shallow water normal modes." This answer, though an acceptable one, is not fully correct. Pekeris' paper was a masterful summation, extension, and clarification of previously classified work that was done during World War II, including his own.

Shallow water acoustics had its genesis during early World War II, when sonar engineers noticed that shallow water broadband signals, such as those from explosions, had a strange looking frequency dispersion curve (low frequencies arrived first, followed by high and medium frequencies) and that they could not model these broadband arrivals very well using ray theory. Declassified World War II reports clearly show modal dispersion curves for shallow water [e.g., the NDRC report "Physics of Sound in the Sea," (1945)] and discuss the use of normal mode theory to explain such curves.

At the risk of being a little controversial, we would claim that, beyond the Pekeris model pointing the way to using modal physics for shallow water, there was only a small amount of purely "shallow water acoustics" work after World War II and until 1970. The technology and measurement techniques were basically the same as those for deep water, just transplanted to different water depths. Progress was made, but in the more general frameworks of physical oceanography and ocean acoustics. Better climatologies of sound speed profiles for the coastal ocean were obtained by the oceanographers, though with an emphasis on larger scale oceanography and subtidal frequency processes. Improvements were made to ray tracing and normal mode codes to predict propagation. For calculation purposes, not only in acoustics, the Fast Fourier Transform (FFT) of Cooly and Tukey quickly penetrated into different areas in the 1960s. Experimentally, we began to see the applications of vertical and horizontal line arrays, containing many hydrophones.

Bottom geology surveys were also made, culminating (for acousticians) in Hamilton's very useful summary paper describing bottom sediment profiles (Hamilton 1985). But the overall emphasis, driven by the Cold War, was on deep "blue water." The continental shelf seas off the USA, USSR, and China were not totally ignored, but neither were they the prime focus of interest for ocean acoustics.

The Near-Present (*1970–1985*). The 1970s represent a breakpoint for many branches of science and technology because of two things: computers and satellites. In computers, mainframes had reached the size, maturity, and availability to allow

most researchers to perform large calculations including serious numerical models of the ocean and acoustic propagation. Mainframes also allowed one to do array signal processing, heralding the era of towed arrays and the cabled Sound Surveillance System (SOSUS). As importantly, the personal computer revolution started later in the 1970s [Triumph of the Nerds, a 2002 DVD, is an entertaining and informative popular history], allowing one to incorporate very compact, very capable data processing and recording devices into ocean instruments. The satellite revolution begat satellite oceanography – visual, synthetic aperture radar (SAR), advanced very high-resolution radiometry (AVHRR), altimetric and scatterometric satellites provided spectacular surface views of the ocean's temperature, currents, waves, and eddies. Overall, these technological breakthroughs helped push ocean acoustics into the modern era. More to the point of this book, they also highlighted some significant differences between deep and shallow water, and led to a more frequent branching off of the two areas.

Let us first look at the effect of computers. For the existing ray and mode programs of the early 1970s, improved computers allowed finely vertically layered and range-dependent programs to be developed.

Additionally, the faster computers and the development of FFT technologies allowed the development of new and even more powerful computational techniques. The "fast field program" developed by Kutschale (1973) and later improved into the SAFARI/OASIS wavenumber integration codes (Schmidt 1988) provided an extremely fast, full wave way to calculate acoustic fields in complicated, layered media, even those including roughness and shear. Another revolutionary theoretical approach, including a calculation code developed by Tappert (1977) in the 1970s, was the parabolic equation (PE) code. This approach used the FFT to march an initial solution out to far ranges in a very short time and made doing three-dimensional calculations via a number of 2D slices (the so-called *N* by 2D) feasible. The fast field programs, being an evaluation of the Hankel transform wavenumber integral, were immediately useful for shallow water, and have been a staple of the trade ever since. The PE, coming originally from a "paraxial approximation" that just looked at low angles, was a deep-water tool at first. However, some ingenious high-angle approximations such as the Pade approximation (for an overview, see Jensen et al. 1994) quickly extended the technique to the large range of angles needed for shallow water acoustics, and again, this tool became a fixture in the community. Indeed, the PE can accommodate mode coupling, diffraction, and many other complicated scattering effects, and is now perhaps the most widely used field computation tool in shallow water acoustics. The 1970s also saw another landmark 3D computational tool developed: the "vertical modes and horizontal rays" technique of Barridge and Weinberg (1977). However, as 3D calculations were a bit ahead of their time in the 1970s, this gem was largely ignored for many years and has just recently become a common tool for researchers.

In addition to the acoustic tools that capable computers enabled, advanced regional physical oceanography models were also made possible, these holding the promise of providing crucial environmental input to the acoustics. Solution of

the 4D, nonlinear Navier–Stokes with complicated boundary conditions and forcing has been a technical challenge for ages, but one that oceanographers, given the computational power and background from the meteorological sciences, felt ready to attempt seriously in the 1970s. Shallow water ocean models in that era were often depth integrated, and thus not really useful for ocean acoustics, but were a start in the right direction.

Satellite images provided an entirely new look at the ocean, but three oceanographic entities especially stood out for shallow water acoustics: the fronts, the eddies, and the internal wave field, particularly the nonlinear internal wave field. Visual and advanced very high-resolution radiometry images showed numerous ocean fronts (shelf break, upwelling, tidal mixing, etc.), a turbulent eddy field (often spun off from large western boundary currents such as the Gulf Stream and Kuroshio), and some very large and unexpectedly common internal waves propagating across the shelf. The strengths and 3D structure of these features would have to be determined by in situ observations, but their existence and commonness were readily apparent in the satellite images. Indeed, much of the fieldwork currently being pursued is in following up these initial remote observations.

Achievements in ocean acoustics during that period, including shallow water, were summarized in many monographs. Some good ones are Ocean acoustics, ed. DeSanto (1979); Wave propagation and underwater acoustics, ed. Keller and Papadakis (1977); and Ocean acoustics (ed) Brekhovskikh (1974) (in Russian).

The present era (*1985–2008*). In around 1985, both political and technical events occurred that pushed underwater acoustics research more heavily toward shallow water. Specifically, the Cold War began to wind down, which somewhat diminished the US, Russian, and Chinese research emphasis on deep water. This left the underwater acoustics community freer to look at shallow water issues.

During this period, Chinese researchers conducted a significant amount of research in the Yellow Sea, paying particular attention to the influence of ocean dynamics. Especially interesting is their series of experiments through the late 1980s on the acoustic effects of nonlinear internal waves. The Yellow Sea experiments heralded the global beginning of a rather intense period of shallow water experimentation that concentrated on not only the specific oceanography issue mentioned above, but also on bottom interaction and inversion, signal processing in shallow water, and even marine biology. Indeed, this period was characterized not only by its activity, but also by its intensely interdisciplinary nature, a welcome development that allowed the shallow water acoustician to *explain* the causes of structures seen in the acoustic data, not just to report and catalog them. Moreover, it entrained excellent practitioners from other disciplines who were just as eager to study the coastal region. We will give a short synopsis of this "Golden Age of Shallow Water Experimentation" in the following discussion, with more details to follow in later chapters. We will then turn to theory, models, and prediction.

Experimental efforts. Let us begin with the oceanography. Coastal fronts have been studied in numerous shallow water acoustics/oceanography experiments, among

them being the 1992 Barents Sea Polar Front experiment, the 1996–1997 Shelfbreak PRIMER experiments, and the recent (2006) SW06 experiment that we participated in. The drastic change in the acoustic field strength going across a front, where one often goes from low angle, ducted propagation to high angle, strong bottom-interacting propagation, was a prime impetus for these studies. Coastal eddies are also acoustically important, but are harder to measure. To date, only two studies that included Sea Soar, Scanfish, and AUV glider oceanographic measurements along with the acoustics might be considered "coastal eddy resolving." Specifically, the PRIMER experiments and the SW06 experiment obtained sufficient 3D measurements of the coastal ocean interior to resolve the eddy structure fully in a region of shallow water acoustic transmissions, and so can provide some insight into these effects. Internal waves, particularly nonlinear coastal internal waves, have received perhaps the lion's share of the experimental attention. Numerous recent experiments have looked at their effects. A sampling of those experiments that the authors have been directly involved in includes the 1995 Shallow Water Acoustic Random Medium experiment (SWARM), the 1995–1996 PRIMER experiments, the 2000–2001 Asian Seas Acoustics Experiment (ASIAEX), and the recent Shallow Water 2006 (SW06) experiment. (Many other experiments have been performed on these topics. We just report on our own in detail for convenience.) Technically, these experiments have examined the strong mode coupling by internal waves, their frequency and azimuthal scattering characteristics, ducting and reflection of sound by internal waves, and other issues.

Russian experimental efforts, though less than during the Cold War era, were also pursued, with an emphasis on internal wave research, and concentrating on low-mode acoustics propagation in areas of the Barents Sea and the Japan Sea shelf. In the Russian experiments, vertical source and receiver arrays were generally towed with an emphasis on providing good radiation and reception of individual modes.

Bottom interaction has always been a prime concern for shallow water acoustics, and shallow water "bottom inversion" has received a lot of attention by the US and European investigators. Such experiments as the 1985 GEMINI cruises off Corpus Christi, Texas, and the above-cited PRIMER and ASIAEX demonstrated new methods to obtain bottom properties based on acoustic data – and these were just a few of many worldwide experiments; again, we mostly use as illustrations the experiments that we have participated in.

Signal processing is yet another huge area of science and technology that interacts with shallow water acoustics. The list of topics that signal processing is important for is about as large as the list of topics that we treat; like physics, signal processing is a basic discipline. However, two of the more active areas in recent years should give one some feel for how this discipline "enters the fray" in shallow water acoustics, specifically horizontal array coherence and time-reversed acoustics.

Coherence of a signal across an array determines the gain of an array, a crucial quantity for detection and observation of signals of any sort. The ocean, especially the shallow ocean, is a strong scatterer of sound, which distorts acoustic wavefronts, and so can degrade array performance in a complicated, fluctuating way.

Quantification of the coherence of arrays, particularly towed horizontal arrays, was crucial for a variety of naval applications, and so many experiments were funded and performed to ascertain this quantity. This measurement is not an easy one, demanding a long towed or fixed horizontal array in shallow water. (Vertical arrays are far easier to deploy in shallow water experimentally.) Horizontal array experiments to determine horizontal array coherence in shallow water have recently been performed in the South China Sea (ASIAEX), the Russian Barents Sea experiments, and the SW06 experiment. Again, detailed results of these experiments will be presented later in the text.

Well known in different "wave research areas," especially in optics, time-reversed effects have recently enjoyed a renaissance in shallow water acoustics. By recording a signal on an array and then retransmitting a time-reversed replica of the received signal, one can receive the (nearly) exact signal transmitted at the source back at the source, with virtually no environmental effects distorting it. This is almost as astounding as Taylor's demonstration that one can "undo" the stirring of a drop of tracer dye in water between two concentric cylinders simply by counter-rotating the cylinders the exact number of turns used to stir the liquid originally. Reversibility in a complicated medium is an unexpected and always fascinating phenomenon. There are also practical applications of this effect, as we will discuss later.

Fish biology has been somewhat of a side issue in shallow water acoustics to date (though that is changing), due to one simple factor: it is extremely difficult to know adequately where the fish are for use in scientific studies! Knowing where one can probably find a school of fish for the purposes of catching some commercially is not the same problem as trying to construct a 4D picture of what the fish distribution, including species, is for incorporation into scientific propagation and scattering models. Acoustic methods have recently provided some first glimpses into fish distributions over a sizable area of the US continental shelf, using a large active source and a large receiving array (Makris 2006). However, in terms of a standard survey technique, this method requires a large amount of resources and funding, and so other methods still need to be developed if the biology is to be factored routinely into shallow water experiments (as it should be, as it can render some experimental results ambiguous otherwise).

In concluding this section, we will note that experimental studies play a crucial role, just as in other branches of acoustics and in physics as a whole. In many respects, this is caused by the very strong medium variability on the continental shelf and its margins. Quantifying the strongly anisotropic, inhomogeneous, and time-varying medium that shallow water acoustics involves is a major challenge – in fact, it is probably *the* major challenge for the field.

Theory (*acoustics*). As mentioned, the most popular "theories" used currently to describe acoustic propagation and scattering in the coastal ocean are normal modes, rays, parabolic equation, wavenumber integration, and the vertical mode-horizontal ray hybrid used for 3D work. Each of these was improved and refined for shallow water usage during the period of interest – we will look at a few highlights of this development here, again in the context of background and motivation.

Normal mode theory was significantly extended by the introduction of "coupled normal mode theory," which could handle the range dependence of the medium in a slice between source and receiver, and even fully 3D effects in theory (Pierce 1965; Milder 1969). This represented the basic extension of Pekeris' 1D model to 2D and 3D, and allowed mode theory to accommodate the full ocean and seabed medium. Elastic medium (shear) effects were also included, but have not proved as popular in normal mode theory as in other frameworks, due to computational intensity.

Ray theory, often ignored in shallow water for frequencies less than 1,000 Hz, is still a very useful tool for understanding the physics of shallow water sound, and indeed, if one includes the effects of beam displacement can often closely approximate full-wave normal mode results (Tindle and Weston 1980). One of the more interesting developments in ray theory during this era was the ray-mode picture correspondence, which showed how one could go from one picture to the other using the constructive interference of modes to create rays and vice-versa. This was especially useful in modeling the arrivals of broadband signals, where the earlier arrivals in shallow water would often be modes, whereas the later arrivals would be rays (Munk and Wunsch 1983). This arrival structure is, as practitioners will recognize, the exact opposite of the deep water case, where one sees steep angle ray arrivals first, followed by a loud, low-angle (low-mode) modal coda.

The parabolic equation, once it was extended to the high-angle regime, was an "instant classic" in shallow water, and indeed is probably the most widely used code for creating acoustic field representations for everyday purposes. Initially, the parabolic equation was criticized as "providing no physical insight, just an answer" (by those who can find a reason to criticize anything, to our minds), but over the decades it was found that one could also project out modes and rays from the PE field, and so one eventually had speed, full wave accuracy, and physical insight all from one model.

Wavenumber integration techniques, based on the Hankel transform representation of the acoustic field, produce both the pressure field and depth-dependent Green function for shallow water acoustic purposes. While this method is intrinsically less suited to range-dependent problems, it excels and outperforms other methods including shear and boundary roughness effects.

Finally, we come to the vertical mode–horizontal ray technique that was developed by Weinberg and Burridge, and then promptly ignored (by and large) for several decades. This insightful model of fully 3D propagation was far ahead of the data stream in shallow water, which only lately has included oceanographic as well as bathymetric refraction and reflection. This theory has recently been extended to include PE instead of ray theory in the horizontal, which gives it even more power as an "almost full wave" approach. (It ignores mode coupling, so it is as good as the adiabatic approximation).

We should note that our "theory" discussion here has somehow talked a lot about computer models and computation. In that these are the modern embodiments of the analytic theory and its practical vehicles for computation, we feel that there is no real discrepancy in doing this.

Models (*combined ocean and acoustics*) *and prediction*: As discussed, in shallow water acoustics, one needs both ocean and bottom information (and in some cases, fish) to make sense of the acoustic propagation and scattering in the medium. However, the bottom is, aside from some ripples due to surface waves, pretty much fixed, and the biology is still (at present) very hard to measure, so we are left with the water column state as a first-order, tractable time-dependent problem to solve. In the early days of ocean computer models, just getting a good representation of the ocean state down to mesoscale resolution, usually in deep water, was the principal goal, and one which was accomplished only after decades of concentrated effort. However, just producing an ocean model was not enough – researchers wanted a few more basic capabilities. Specifically, they wanted the models to (1) assimilate ocean data so as to improve the model's estimate, especially in real time; (2) interact with other disciplines such as acoustics, biology, chemistry, etc.; and (3) have predictive capabilities, in both forecast and hindcast modes. The volume "Oceanography and Acoustics: Prediction and Propagation Models" (Robinson 1994) summarized a lot of the early work, up to about 1994 where the oceanographic models and acoustics merged. Since then, many individuals and groups have worked on this problem, extending the deep ocean models into the coastal zone, using finer grids, "nested domains," and other numerical devices. At present, coastal ocean models can assimilate data well and examine oceanographic scales down to about the internal tide level (a few kilometers wavelength and ~12-h periodicity). Acoustic codes, including fully 3D codes, can be incorporated into these ocean models rather readily. However, this set of developments is still in its youth (though not in its infancy), and there are some exciting challenges for the future, as we will discuss.

1.3 The Future of Shallow Water Acoustics

The near future (*2011–2020*). One of the advantages of having worked in an active area for ~30 years (as all the authors have) is that you get a sense of how much progress has been made, and thus also have some idea of how much progress can and should be made in the future. The area of shallow water acoustics is still a very active one, and should stay so for many years to come – the coastal ocean is too important to human activities, for this not to be so. That being said, let us look briefly at where we think shallow water acoustics is going, in the next decade at least, as after that our crystal ball becomes murky, and some unforeseen technical advance can (and probably will) render our predictions conservative and outmoded.

Taking up where we just left off, let us look at computers and numerical modeling. The paradigm for computer technology advancement has been Moore's Law. To quote the Wikipedia directly (why not – this is the twenty-first century):

"Moore's Law describes an important trend in the history of computer hardware: that the number of transistors that can be inexpensively placed on an integrated circuit is increasing exponentially, doubling approximately every 2 years.

The observation was first made by Intel cofounder Gordon E. Moore in a 1965 paper. The trend has continued for more than half a century and is not expected to stop for a decade at least and perhaps much longer.

Almost every measure of the capabilities of digital electronic devices is linked to Moore's Law: processing speed, memory capacity, even the resolution of LCD screens and digital cameras. All of these are improving at (roughly) exponential rates as well. This has changed dramatically the *usefulness* of digital electronics in nearly every segment of the world economy. Moore's Law is a driving force of technological and social change in the late twentieth and early twenty-first centuries."

So, with the added computer capabilities that we should have in the near future at least, what will we be able to do? From the viewpoint of ocean acoustic models, the added capabilities would let us do broadband, 3D, and fully elastic models with impunity – things which we can do now when we need to, but which are still considered computationally intensive. However, this is an incremental advance, and not a basic one. It is in physical oceanographic modeling that more computer power will best make itself felt. At present, ocean models with fully consistent dynamics (as opposed to nested models, which "graft" smaller scale models onto larger scale ones, letting the large scale force the small, but generally with no feedback) are limited to mesoscale phenomena and larger, due to the dramatic increase in the number of grid points with the fineness of the space–time grid used. But ocean acoustics is sensitive to sound speed gradients, which can be very large over small scales in the coastal ocean. The challenge, and the progress to be made, is in modeling these finer and finer ocean scales so that acoustics users can eventually do numerical propagation through a very realistic, dynamically consistent model of the ocean. In that space and time scales can be tens of meters and 1–2 min, this is anything but a trivial challenge to the oceanographers.

Keeping with the technology theme, let us consider one of the other great advances of modern technology: robots. We live in an age where computational abilities, low-power microelectronics, sensor technology, pattern recognition and artificial intelligence, and mechanical devices have evolved to a very high state, thus allowing us to make autonomous systems (robots) with very substantial capabilities – no longer mere toys or research laboratory curiosities. The two premier robotic technologies for exploring the oceans are Autonomous Underwater Vehicles (AUVs) and ocean gliders. The first are faster, propeller-driven vehicles, whereas the second are slower ones, driven by weights and buoyancy engines and tilted fins that convert up and down motions to forward thrust. These vehicles are already carrying a wide variety of sensors for oceanographic and acoustic sensing – but we are only at the infancy of this technology, and there is a *lot* more to come! Again, let us peer into our near-future crystal ball and see if we can pick out the likely advances, in AUVs at least.

One of the big criticisms of current propeller-powered AUVs is their "persistence," i.e., their ability to carry out long missions for more than a few days. Gliders do better on this count (weeks instead of days), but are hindered by their slowness and small payload capability. It is not hard to envision that with better battery technology, optimized propulsion versus hotel power budgets, the combination/

convergence of AUV and glider modes of propulsion, better propellers, docking stations, and mother vehicle "battery recharges" (and data dumps), among other things, the next generation of AUVs/gliders should have far more durability and payload capability. This seems a rather safe bet. In addition, the more durable, "smarter" AUVs/gliders (in terms of preprogrammed intelligence, sensing and adaptive behavior, and mission capability) should also be a safe bet. Groups of vehicles that intelligently coordinate with one another are already being experimented with, and the acoustic communications and control theory needed to make this work routinely in the future are well under development. The vision of a swarm of robot vehicles loaded with advanced sensors being deployed from a mother vehicle (robotic, of course) and surveying a region of the ocean (deep or shallow, by the way) does not at all seem farfetched at this point in time. Moreover, these should be cost effective compared to large ships with large science parties manning them (the daily cost of a large oceanographic ship for 2009 is ~25–30 K USD). This might sadden the romanticists who yearn for a return to the days of reversing thermometers deployed from wooden sailing ships (see, e.g., "The A Boat," Densmore 1995), but for those of us who enjoy seeing good ocean data first and foremost, life will improve. (By the way, we stopped short of self-replicating, evolving ocean robots – this thought was just a bit scary since we could all too easily be replaced!)

Another technical advance that has crept up on oceanography is "ocean observatories," which are being implemented more and more, especially along the coastal ocean (which has the largest economic and societal impact). These observatories not only center around fixed nodes of instrumentation (including acoustics), but can also have mobile and distributed sensors associated with them, including satellites, radar and CODAR, gliders and AUVs, and so on, as well as interact with data-assimilating numerical models.

These observatories began concentrating on small areas and regional concerns, but are now looking to integrate measurements over a variety of scales, often via linking with each other. A particularly large initiative called the Integrated Ocean Observing System (IOOS) has recently been initiated by NSF and, if successful, should be the harbinger of much more work in that vein.

So far we have concentrated on the technology that will enable measurement and modeling advances in shallow water acoustics, but one question still remains: what are the new acoustic questions that these advances will address (in addition to the old, hard ones we will continue to deal with)? We have one or two thoughts in that direction, as well.

Four new areas that we think will come to the forefront in shallow water acoustics are (1) the continental slope and the canyons often found along it; (2) ocean uncertainty (i.e., the ocean variability that we are not able to measure or capture); (3) measuring the marine biology relevant to acoustics; and (4) mid-frequency acoustics.

The first area, the continental slope and its canyons, is perhaps one of the hardest to deal with in terms of acoustics, oceanography, geology, and biology. It is the fully

3D acoustics problem, with complicated bathymetry, sub-bottom stratification, water column profiles, and loads of fish (the continental shelf break is prime fishing ground). The canyons that routinely cut across the slope are even trickier in terms of their oceanography and acoustics, but have been less explored, mainly due to their inherent complexity and difficulty to sample. These areas are important ones in terms of their economic and naval impact, and we feel that they soon will be a prime area for shallow water research.

The second area of shallow water acoustics that we expect to get a lot of future attention (as it is already getting a lot of present attention) is the area of "quantifying uncertainty." We have finally, if belatedly and grudgingly, come to the realization that we are invariably resource limited in measuring the ocean, and that in any given application, there will be an inherent resolution attached to the measurement that is imposed strictly by our finite resources. We know from both theory and practice that we cannot measure all ocean processes at all scales, even though some of those scales may be important to the application at hand. Even the interpolation provided by ocean numerical models cannot fix this. And so the question arises, "How do we deal with this uncertainty?" Shallow water, with its great complexity, really brings this home. The oceanographic internal waves, fronts, eddies, filaments, and other structures found there, along with the complicated bottom, will likely always be beyond the reach of measuring on a "realization" basis, and so will force us to some statistical or other approach.

Next, there is the question of quantifying our finny friends, unless of course overfishing makes this a moot point [complete overfishing and extinction have been predicted by some by 2048 given the current trends (Blakemore and Sandell 2006)]. We have gotten to the point where we have some skill in looking at coastal oceanography and geology, but marine biology has remained more elusive in regard to routine measurement and quantification of fish schools and marine mammals. The technology of AUVs, gliders, and observatories may hold some hope for quantifying fish to the point where we will know in a given experiment how they will affect acoustics, but we have a long way to go.

Finally, we come to so-called mid-frequency shallow water acoustics (1–10 kHz), which we include in this book up to ~5 kHz. This frequency range has traditionally been somewhat of an orphan between the high- and low-frequency research programs. This is likely due to its physics being a hybrid of low-frequency effects (many bottom and surface interactions before the signal is lost to noise) and high-frequency effects (ray theory is a good descriptor, by and large). Mid-frequency systems have numerous applications: naval (ASW and acoustic communications), marine mammal (whale and dolphin hearing and vocalization), fisheries (stock survey), and geology (echo-sounding), and deserve attention as an important frequency range in shallow water. This recognition is just beginning to be achieved, and the recent SW06 experiment featured a large mid-frequency research component.

1.4 Some "Old Favorite" Research Areas Revisited and Updated

In June 2007, a career celebration was held for Massachusetts Institute of Technology (MIT) professor Ira Dyer, one of the pioneers in underwater acoustics and a colleague still active after a long, productive career. Ira's lecture topic was "Shallow water acoustics: basic challenges," and he presented a very concise summation of what he felt some of the ongoing challenges were in shallow water acoustics. The topics he pursued were not so much the 10-year crystal ball predictions we just made, but rather solid extensions of past and current work while looking into the future. These are topics that we have looked at for years, and in some cases decades, but which still have a lot of depth and unsolved puzzles to them.

The first problem that Professor Dyer noted was "quantifying the azimuthal anisotropy of the shallow water waveguide," a topic that the authors of this book are actively involved in, and which we address in the text. It is obvious that the shallow water waveguide breaks azimuthal symmetry, in that it has (to first order) an alongshore symmetric geometry. The shelf-break bathymetry, the shelf-break front, and the shoreward propagating internal waves and internal tide all strongly define a "preferred direction" in the x–y plane, making azimuthal symmetry along acoustic paths that interact with these features impossible in the mathematical sense. Additionally, complicated sub-bottom stratigraphy and directional seas can also destroy azimuthal symmetry. But the practical question, as in all symmetry breaking, is "by how much?" It is well known that the shelf-break bathymetry and shelf-break fronts are strong features, as well as large (directional) surface wave fields. More subtle is the question of how "directional" are the effects of internal waves and tides, sub-bottom strata, ocean eddies, and ocean fine structure. Calculational studies so far indicate that this question has a complicated answer that depends on frequency, source/receiver depth, and the details of the oceanographic and bottom features. However, there is a dearth of data on the azimuthal dependence of propagation (e.g., transmission loss), noise, and other acoustic field variables. Some recent data on this problem have been obtained in the context of the SW06 experiment, as will be discussed later, but more detailed experiments need to be done before we can say that we have a firm handle on this seemingly simple, but very important, question.

A second "long-lived" question is "where do you place the random/deterministic boundary when describing the shallow water medium?" We can do deterministic calculations of the propagation through and scattering by a very complicated medium at this point in time, given a realization of the environment to use as input. The practical trouble is not computation (though it is a factor), but in obtaining a very detailed environmental input. As mentioned before, there are many fine-scale, but acoustically strong, processes in shallow water (e.g., nonlinear internal waves and sub-bottom discontinuities) that cannot be routinely measured. We thus have to ascribe, in the absence of detailed realizations, a random medium

nature to these scatterers and processes. However, these processes often have a high degree of structure, and so some measure of this structure should be incorporated as well. Where and how to do this is still a research issue. One study (Lynch et al. 2004) addresses this very issue in terms of modeling the fluctuations in sound exposure that marine mammals are subject to because of finer scale oceanography.

Related to the last issue, and in some sense a possible solution to it, is the use of an effective medium to describe the water column and (especially) the bottom when one has a complicated, spatially and (for the water) temporally varying medium that cannot be measured or modeled easily. Let us discuss two examples of where such an approach might be applied, noting that we will only be posing the problem here, and not giving its solution.

Describing the sub-bottom materials and stratigraphy for the purposes of shallow water acoustics over the frequency range of 50–5,000 Hz has been a very challenging and, in many ways, very frustrating enterprise. The bottom can change its acoustic properties significantly over a few meters (as a function of x, y, z), and measurements of the bottom show both large-scale and small-scale variation. Thus, any set of regional measurements is apt to show significant variability and a distribution of properties from measurement to measurement. To add further confusion to this picture, bottom measurements are often made using transmissions over a sizable path (a few kilometers), thus giving an average over that path. The result of this measurement will vary depending on path length and the center and orientation of the path. To add icing to the cake, such "longer path" measurements often have an oceanographic scattering component in them that can significantly affect the results of the bottom estimate. As an example, scattering by internal waves can increase or decrease the transmission loss seen so that both sound speed and attenuation measurements could effectively combine ocean and seabed into one medium. One can attempt to separate the ocean and seabed effects on bottom property estimates, but this is new work, and not perfected yet. (So do we make an effective bottom model that also has "unknown and uncorrectable oceanography contributions" in it? We would hope not.)

As an example of how the "complicated medium" problem has been handled throughout the years, let us look at how the U.S. Navy has modeled the bottom for practical sonar systems. Initially, bottom loss was modeled using half a dozen or so "bottom type"-based bottom loss versus grazing angle curves, with the bottom type typically being determined by grab samples of the near-surface sediments. This was adequate for the initial days of sonar systems (where higher frequency systems would mostly sense the near-surface part of the sediments), but did a poor job with some of the very basic aspects of bottom loss such as its frequency dependence. Thus in the 1970s, a "Bottom Loss UpGrade" (BLUG) model (Michelson et al. 1981) was introduced to calculate the bottom loss curves from a simple geoacoustic model. This helped to get much closer to the basic physics of the bottom loss, but still had very simplistic ways of dealing with the layering and material properties (including an interesting device called the "stainless steel layer" which accounted for some medium scattering effects by inserting a surface layer with the sound speed of stainless steel). As computers and ocean calculation codes became better,

it became possible to put in a multilayered model of the bottom, including shear and layer roughness, which can vary in range as well. This would seem to answer the problem, except for the fact that these models now ask for far too much experimental information for their creation. In reality, one is left with figuring out how to produce a layered structure that is commensurate with the resolution and variance of the existing measurements. Moreover, the question to be addressed is how good of a model is needed for a given user task. Progress *is* being made in this area, as we get more savvy about how to match our models, measurements, resolution estimates, and user requirements, but there is still some interesting work left to be done.

Chapter 2
Coastal Oceanography, Geology, and Biology

2.1 The Coast as Acoustic Waveguide

From the viewpoint of low- and mid-frequency shallow water acoustics, the ocean shelf is a waveguide, limited by a pressure release boundary above (the ocean's surface) and an absorbing boundary below (the ocean bottom/seabed). In describing the sound field radiated from a source (whether point or distributed), both the waveguide's interior and boundaries play an essential role. To describe this waveguide's interior and boundaries, we must delve into the areas of coastal physical oceanography, geology and geophysics, and biology. These areas are vast scientific fields on their own, and so we limit ourselves here to touching on only the elements of them that are most germane to shallow water acoustics. The rest we leave to the reader to explore independently. Toward this, we initially provide three physical oceanography references, which should give the reader a starting point: Pond and Pickard (1983); Bowden (1983); and Cushman-Roisin (1994). We will start by looking in the first six sections of this chapter (Sects. 2.1–2.6) at physical oceanography, then devoting Sects. 2.7–2.10 to marine geology, and finally ending in Sect. 2.11 with some discussion of marine biology.

In looking at the shallow water acoustic waveguide in general, the sound speed in both the water and the bottom depends mainly on depth (the z coordinate) and as a "beginners rule" varies slowly in the horizontal coordinates and in time (more precisely, geotime). Thus, it is commonly taken that the sound-speed field $c(x, y, z)$ is to first order only a function of z. This allows us to look initially at the vertical variability of the sound speed at different geographical locations and seasons and make both sensible intercomparisons and distributions of variability.

In that vein, distilled results of different observations and measurements of sound-speed profiles in various areas of the world ocean are shown in Fig. 2.1. We can see that the water column sound-speed profiles differ depending on geographical location and season, whereas the seabed sound-speed profiles are more simply determined by depth (overburden pressure) and material composition. We will attempt to explain the details of these profiles in the sections that follow.

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_2,
© Springer Science+Business Media, LLC 2012

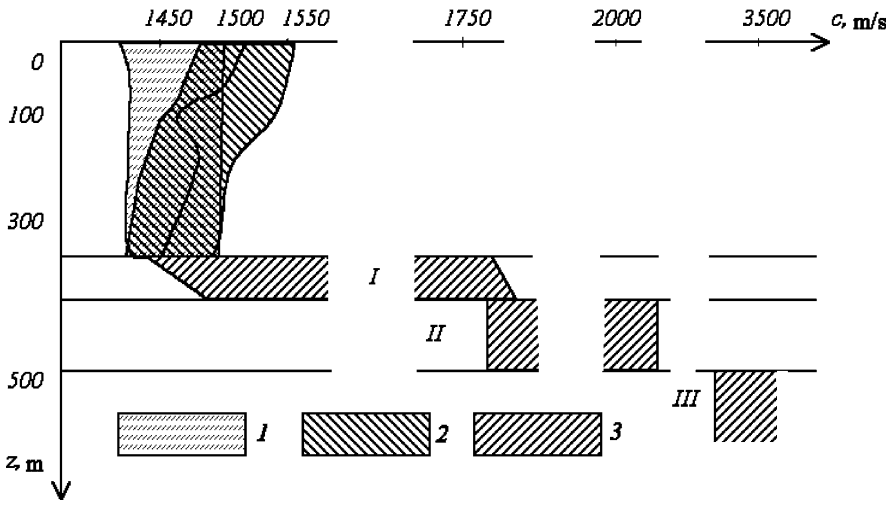


Fig. 2.1 Sound-speed profiles in different areas of the world's coastal ocean and for different seasons. 1 – higher latitude, winter sound-speed profiles, 2 – lower latitude year round and summer mid-latitude profiles. 3 – bottom sound speed. The nonmonotonic curve included shows a sound-speed profile with a possible pure refraction duct

For analytical calculations and estimation, we can select a small numbers of "canonical" (i.e., generic or typical) sound-speed profiles that can be used to characterize a large fraction of the shallow water environment (Kuperman and Lynch 2004). However, these profiles, shown in Fig. 2.2, are merely first-order simplified representations of the reality of the shallow water seas (as represented by Fig. 2.1). In order to know when they can be used or when they should be augmented, we must first discuss the real environment. We will do that next, approaching the coastal oceanography first, and then moving to the coastal marine geology.

2.2 Properties of Sea Water: Vertical Stratification and Its Seasonal Variability

To a good approximation, we can consider seawater as a two-component medium, consisting of a solvent (water) and various salts. The concentration of salt (salinity) is usually denoted as S and is measured in pro-mille (denoted as ‰) or in parts per thousand (ppt). The average value of salinity in deep ocean seawater is about 35‰. Variations of salinity within the shelf zone are in the order of 20–35 ppt.

The thermodynamic state of sea water is characterized by temperature T°, density ρ, salinity S, and pressure P and is described by the state equation

$$\rho = \rho(T^\circ, S, P). \tag{2.1}$$

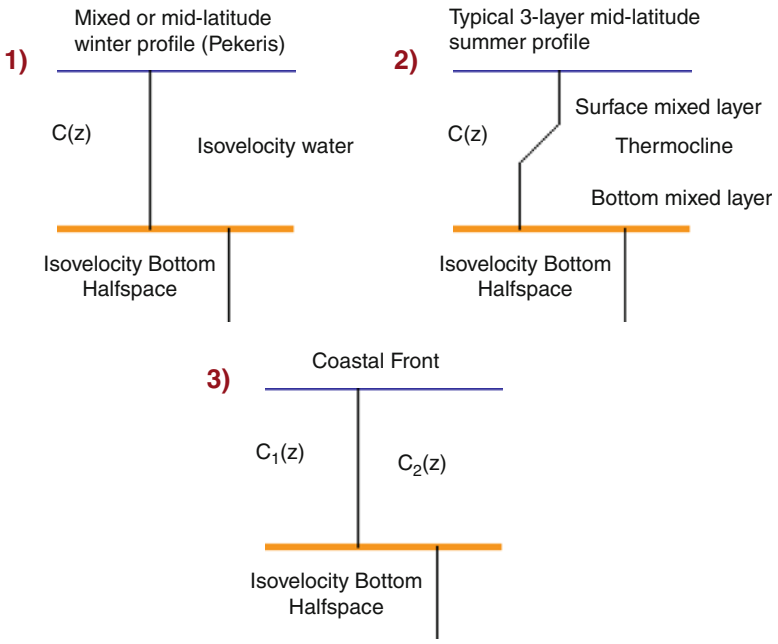


Fig. 2.2 Three simple sound-speed profiles which characterize much of shallow water acoustics. The first profile is the so-called "Pekeris waveguide," typical of mixed water column conditions over a simple bottom. The second profile is the "three-layer model," typical of low-latitude or summer mid-latitude waters over a simple bottom. The third profile is that of an enormously simplified coastal front, again over a simple bottom

Due to the complexity of the composition of seawater, this equation can be written only empirically. In physical oceanography, the parameters temperature, salinity, and pressure are basic, whereas other observed characteristics (sound speed, heat capacity, conductivity, etc.) are expressed through these three parameters. As is well known, these parameters depend on both space and time (we will denote this "slow" time as geotime, T, in contrast with the "fast" time, t, used in forming expressions for the high-frequency oscillations of the wave fields).

One of the main characteristics of seawater is its bulk adiabatic compressibility (defined by the relative variation of the density with pressure)

$$\beta_a = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial P}\right)_a \tag{2.2}$$

which also depends on the basic thermodynamic parameters and, in relating it to acoustics, generally depends on the frequency of the pressure oscillations (of sound) as well. The sound speed is connected to β_a by $c = 1/\sqrt{\rho\beta_a}$. In our case, $\beta_a \sim 4 - 5 \times 10^{-10}\,\mathrm{Pa}^{-1}$. β_a depends on T° and S, and according to experimental data (e.g., Babii 1983), compressibility decreases with both increasing temperature and salinity, whereas sound speed correspondingly increases.

Table 2.1 Coefficients of equation of state

β_a [Pa^{-1}]	β_T [$°K^{-1}$]	$\beta_S(\circ/\circ\circ)^{-1}$	$\partial c/\partial T$[m s^{-1} $°K^{-1}$]	N [cph]
$4-5\times10^{-10}$	1.3×10^{-4}	8×10^{-4}	4.6	<10

This variability of β_a determines the dependence of the sound speed upon the thermodynamic parameters; of particular importance, the dependence of the sound speed on temperature goes as $\frac{\partial c}{\partial T^\circ} \sim 4.6$ m/(s degree), as a rough "rule of thumb."

We also need to define the thermal expansion coefficient

$$\beta_T = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T^\circ}\right)_{S,P} \tag{2.3}$$

and the coefficient of variation of the density as a function of salinity

$$\beta_S = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial S}\right)_{P,T}. \tag{2.4}$$

With these quantities defined, we can then write down the differential equation of state of seawater as

$$\frac{d\rho}{\rho} = \beta_T dT^\circ + \beta_a dP + \beta_S dS \tag{2.5}$$

where (typical) numerical values of the coefficients are shown in the Table 2.1 and N is the Brunt–Vaisala frequency (see Appendix A). This result will be used in our further work.

Temperature variability is a good first-order proxy for sound-speed variability in shallow water, since the well-known equation of state (Urick 1975) gives that the sound speed is approximately

$$c = 1449.2 + 4.623T^\circ - 0.0546(T^\circ)^2 + 1.391(S-35) + 0.017z \tag{2.6}$$

where c is in m/s, T° is in degrees Celsius, S is salinity in ppt or psu, and z is in meters. Since depth effects only add about 0.017 s^{-1} to this equation, they are often ignored to first order in shallow water. Salinity also is a generally weak sound-speed signal (though not always, as coastal waters show the largest variation in S). For the range of T° and S typically found in coastal waters ($T^\circ \sim$ 6–25°C and $S \sim$ 20–35 ppt), it is easily seen that temperature effects dominate. However, to get at the temperature field theoretically, the physical oceanographer has to consider the salinity just as much as the temperature, since the density field is the most critical parameter in ocean flows and structure, and density is about equally dependent on temperature and salinity effects. Thus, (2.5) is the critical state equation for physical oceanography, whereas (2.6) is the important state equation for ocean acoustics.

The temperature distribution in the shallow water column is determined by both the large-scale water masses found at a particular location and the competition between solar heating, medium stratification and vertical mixing, the latter of which is induced by wind stress on the surface as well as wave and current stress on the bottom. In modeling the vertical temperature distribution, we will consider here a simple yet physical and useful model, published by James (1977), which is also described in the coastal oceanography textbook by Bowden (1983). This variability in the vertical temperature profile with geotime, i.e., the function $T^{\circ}(z, T)$, is central to shallow water acoustics.

We begin with the one-dimensional vertical diffusion equation for temperature,

$$\frac{\partial T^{\circ}}{\partial T} = \frac{\partial}{\partial z}\left(K_z \frac{\partial T^{\circ}}{\partial z}\right), \tag{2.7}$$

where T is the geotime and K_z is the vertical eddy diffusion coefficient of heat, dependent on depth and geotime. To take into account that mixing is inhibited by density stratification/stability, K_z is modified to

$$K_z = K_0(1 + \sigma Ri)^{-p}, \tag{2.8}$$

where $g' = K_0 g$ is the coefficient for neutral stability (no stratification present) and Ri is the gradient Richardson number, given by

$$Ri = g\beta_{\mathrm{T}} \frac{\partial T^{\circ}}{\partial z} \Bigg/ \left(\frac{\partial U}{\partial z}\right)^2. \tag{2.9}$$

In (2.8) and (2.9), σ, p are empirically determined constants, on the order of $\sigma = 0.3$ and $p = 1.5$, respectively; U is the horizontal current magnitude; and β_{T} is the coefficient of thermal expansion of water.

The solution of (2.7) is subject to surface and bottom boundary conditions.

At the surface,

$$\left[K_z \frac{\partial T^{\circ}}{\partial z}\right]_{z=0} = \frac{\tilde{Q}}{\rho c}, \tag{2.10a}$$

where $\tilde{Q}$ is the flux of heat downward through the surface (in particular, the average heat flux from the Sun is about 0.14 W/cm^2). At the bottom, the turbulent diffusion of heat is inhibited, so

$$\left[K_z \frac{\partial T^{\circ}}{\partial z}\right]_{z=H} = 0. \tag{2.10b}$$

The temperature gradient, $\partial T^{\circ}/\partial z$, is then determined via calculation. The heat input $\tilde{Q}$ and the wind speed are the meteorological inputs to the model (the wind

speed being one of the factors in K_z, as we will see). The term $(\partial U/\partial z)^2$ in (2.9) is obtained by assuming that the bottom and surface boundary layers have logarithmic velocity profiles, the surface one caused by the wind stress and the bottom one caused by the tidal currents. For instance, the near-bottom velocity profile will look like

$$u = \frac{u_*}{k_0} \ln \frac{z}{z_0}, \tag{2.11}$$

where z is the distance from the bottom, $u_* = (\tau/\rho)^{1/2}$ is the so-called friction velocity (τ being the stress due to the near-bottom tidal current), z_0 is a roughness parameter, and k_0 is von Karman's constant, approximately equal to 0.4. A similar form is easily found in the literature for the surface boundary layer. This form will be familiar to many readers as the "law of the wall."

The neutral eddy diffusivity (at $Ri = 0$) is expressed as the sum of surface and bottom parts as well, i.e.,

$$K_0 = K_{0\mathrm{W}} + K_{0\mathrm{T}} \tag{2.12}$$

where $K_{0\mathrm{W}}$ is proportional to the wind speed and $K_{0\mathrm{T}}$ is proportional to the tidal current speed and the depth of the water.

With all this in hand, the model is run. When a thermocline is formed at a certain depth, $z = z_0$, it is assumed that wind mixing cannot penetrate below that depth and that bottom (tidal current) mixing cannot penetrate above it. Moreover, as is also usual in mixing models, if $\partial T^\circ/\partial z$ becomes positive, indicating unstable stratification, the model mixes the temperature instantaneously over a range of depth, thus restoring stability to the profiles. By integrating (2.7) numerically, starting from given initial conditions, one can compute the evolution of the temperature profile with time, and indeed James' results look exactly like our three-layer model of shallow water, i.e., a surface layer, a thermocline, and a bottom mixed layer, as seen in Fig. 2.3.

We should make two notes about the temperature stratification model presented above, particularly from the viewpoint of an acoustics "model user." First, and most importantly, while this model is useful for understanding the physics of how the surface mixed layer, bottom mixed layer, and thermocline arise, it is relatively weak for predictive use. Mixed layer models are demanding in their inputs for wind stress, insolation, initial stratification, etc., and typically, this much detailed input is hard to get. For acoustics users, climatology probably gives a good enough first-order estimate of stratification, and an in situ XBT or CTD gives the best improvement to that. Second, the model discussed above is one of the simpler ones, and more complicated models do exist which include salinity effects, more detailed models of turbulence and mixing, and even three dimensions. We refer the reader to the classic article by Price et al. (1986) as an example of a more sophisticated 1D approach.

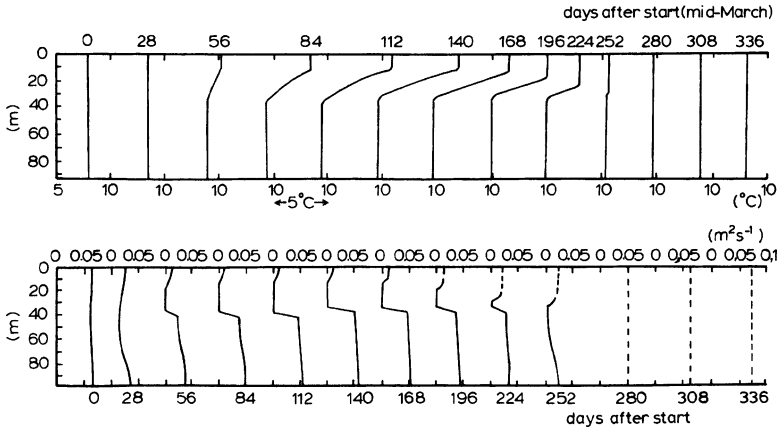


Fig. 2.3 Development of a seasonal thermocline, computed from the 1D model discussed. *Top panel* shows temperature profiles at 28-day intervals, the *bottom panel* the corresponding coefficients of eddy diffusion at the same times. With permission from I.D.James, Estuar. Coastal. Mar. Sci, 7, 197–202. Copyright Academic Press (London) Ltd

Extension of the shelf sound-speed profile results to the slope is straightforward, at least to first order. One just gets rid of the bottom mixed layer in the shelf profile, and extends the downward refracting gradient layer to the typical downward refraction one gets in the first kilometer depth due to the exponential temperature decrease, using the climatology of the region as a guide. This gives the profile shown in Fig. 2.3 top panel. However, this is a generic mean profile and does not include the effects of fronts, eddies, IWs, fine structure, etc. These are often present on the slope, and so the above "prescription" must be used with extreme care.

One also notes that down to the depth of about 750 m, the depth of the mid-latitude deep sound channel axis, this profile is downward refracting. This means that sound coming off the shelf onto the slope will "hug the bottom," giving high sound levels near the bottom, and a shadow zone just above that. However, past the depth of 750 m (it often can be down to ~1 km), the sound will no longer be driven into the bottom, but rather the field will "lift off" the bottom and continue its travel in the well-known deep ocean SOFAR axis. This can be modified by the shelf break front and other structure; so again, one must look at this effect exercising care.

2.3 Horizontal Stratification and Its Variability: Fronts and Eddies, Surface Ducts, and Storm Surges

In the above discussion, we concentrated on the vertical variability of the temperature field and discussed a simple model of how that variability arises. This vertical structure is the most important feature of the shallow water column, as the water column and bottom are approximately horizontally stratified (comprised of

vertically stacked layers) over the propagation scales of interest, which reach to about 50 km in shallow water. However, horizontal stratification is a broad-brush first approximation only, and in many shallow water scenarios, there is appreciable sound-speed variability in the horizontal direction, as well as in the vertical. Perhaps the strongest horizontal variability in shallow water is due to shallow water fronts, and so we address that topic next.

There are four separate kinds of shallow water fronts which are of potential interest to underwater acousticians. They are (1) upwelling fronts, (2) tidal mixing fronts, (3) buoyant plume fronts, and (4) shelf break fronts. These fronts all have somewhat different dynamics and characteristics, which in turn determine what type of horizontal temperature/sound-speed gradients they produce and where on the continental shelf they are observed. We will look at all four of these fronts and present examples where possible.

Upwelling fronts. Upwelling of water at the coasts is a very well-known phenomenon, particularly since it has a profound effect on fishing activity, the Peruvian anchovy fisheries being perhaps the most famous example (Bowden 1983). Specifically, upwelling is the process whereby cool water from depths on the order of 50–300 m is pumped upward into the surface layer due to the effect of wind, thereby bringing nutrient salts into the photic regions near the sea surface. From the point of view of acoustics, the biologically important transport of salts to the surface (a salinity signal) is almost irrelevant; however, the transport of cold, deep water to the normally warm surface layer is not, as this temperature signal provides an appreciable sound-speed signal, and indeed an acoustic "front."

Physically, upwelling is caused by two things: the first is the wind-induced Ekman transport of near-surface water away from the coast (surface Ekman transport being produced by the balance between the wind stress and the Coriolis force). The second is the continuity equation demanding that the amount of (warm) surface water that gets transported seaward must be replaced with an equal amount of deeper (cold) water being transported shoreward and eventually to the surface, so as to balance the masses. Ekman transport is to the right of the wind in the Northern Hemisphere and to the left of it in the Southern Hemisphere, so the direction of the wind along the coast is critical. We should note that if the wind direction is such that it is pushing the warm surface waters onshore, rather than offshore, an analogous process called downwelling occurs. This pushing of warm water onshore tends to create a deeper surface warm layer near the coast, rather than a sharp front. The basic mathematics of both the Ekman transport and the continuity equation are to be found in numerous elementary oceanography textbooks, and we would refer the reader to the treatments by Bowden (1983) and Cushman-Roisin (1994) for details.

Though one should use a more exact dynamical model to construct the geometry of an upwelling front, we can create a first-order estimate of upwelling frontal structure useful for acoustics from rather simple dynamics. In the top and bottom panels of Fig. 2.4, we show a two-layer ocean before and after an upwelling event, with the initial warm surface layer being of depth H, density ρ_1, and sound speed c_1. This layer overlies a (mathematically) infinitely deep layer with density ρ_2 and

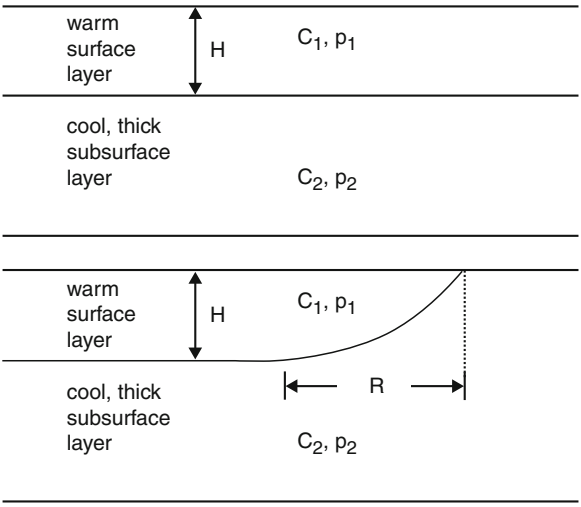


Fig. 2.4 *Top panel* shows the two-layer system of warm surface layer overlying a deep cooler layer near the coast. *Lower panel* shows result of upwelling, where deep cooler water is brought up to the surface due to the warmer surface layer being pushed seaward. The distance R over which the front extends is a good dynamic approximation, but the exact shape of the front needs a more exact dynamical treatment to determine – our figure above is an "artist's conception" in that respect

sound speed c_2. After upwelling occurs, the cold lower layer rises from depth H to the surface over a distance R, which is given by Cushman-Roisin (1994) as

$$R = (g'H)^{1/2}/f_C, \tag{2.13}$$

where

$$g' = \frac{(\rho_2 - \rho_1)}{\rho_2} g \tag{2.14}$$

is the "reduced gravity" (see Appendix A.3) and

$$f_C = 2\Omega_E \sin\theta \tag{2.15}$$

is the so-called Coriolis parameter, where Ω_E is the rotational rate of the earth and θ is the latitude. Thus, one has a horizontal sound-speed gradient which is sharp across the surface layer, and has a depth average value (over the layer depth) of $(c_1 - c_2)/R$. Here, R is typically on the order of 5–10 km, and Δc is of the order 10–50 m/s so that the average horizontal gradient is of the order 1–10 m/s/km or $10^{-3} - 10^{-2}\,c^{-1}$. This is an appreciable horizontal gradient for acoustics.

Tidal mixing fronts. Whether or not a thermocline develops in the late spring or early summer depends critically on whether or not there is enough turbulent kinetic energy generated throughout the water column to mix heat downward at the rate at which it is being received at the surface. Both surface winds and tides can create turbulence and mixing, but the winds are an intermittent mechanism, whereas the tides are a constant one. Thus, we will consider here the case where the mixing is due to tides alone.

From a rather simple derivation (Bowden 1983), one can show that there is complete vertical mixing of the water column when

$$\frac{H}{U_t^3} \sim 70 - 100 \,\text{per}\,\text{m}^{-2}\,\text{s}^{-3}, \tag{2.16}$$

where H is the water column depth in meters and U_t is the tidal current speed in m/s. This form easily lets us see the depths to which one expects full mixing. Taking U_t equal approximately to 0.5–0.7 m/s (about one to one and one-half knots) as typical tidal current magnitudes, full mixing will be seen in water columns with depths between ~10 and 35 m. Past this point, a two (or more)-layer system will evolve. For water depths greater than this mixing depth, the lower layer will be at some sound speed c_{mixed}, whereas above the mixed region, the water sound speed will be c_{upper} (in the two-layer model). This idealized system agrees well with real world measurements of this effect, as shown by comparing our simple model with data, as seen below (Bowden 1983), in Fig. 2.5.

Buoyant plume fronts. River outflow plumes create fronts that are well known to even lay observers, as large river discharges after storms often contain copious amounts of mud and fine sediment, which show the edges of the plume sharply against the much clearer, deeper water. However, these plume fronts are not as acoustically strong as the other three types, for reasons we now discuss. The main signal of a buoyant (freshwater) river plume front is salinity, which may be ~20–30 ppt compared to 35 ppt in the deep ocean. This creates up to an ~15–20 m/s sound-speed contrast between the buoyant plume and its surroundings, a not insignificant disturbance. However, these strong salinity signals only extend over the top meter or two of the water column near their origin. These plumes then travel many tens of kilometers outside the river mouth, and the initial freshwater pulse is spread out both horizontally (by wind and currents) and vertically (by mixing), thus diluting it. By the time river salinity signals have traveled an appreciable distance, the sound-speed disturbance is only 5 m/s or less. This is weak as far as coastal horizontal fronts are concerned. However, the near-surface acoustic duct that they can create can be very effective in ducting sound at higher frequencies, as we will discuss later in our section on surface ducts.

Shelf break fronts. If the buoyant plume fronts are the least important ones to acoustics, their distant offshore cousins, the shelf break fronts, are perhaps the most important. The shelf break regions of the world (the lines where the continental shelves dip abruptly into the continental slopes, usually at depths of 100–200 m) are often the boundaries between very cool (but somewhat fresher) coastal waters and very warm (but somewhat saltier) deep waters influenced by large boundary current systems such as the Gulf Stream and Kuroshio. An example of a well-measured shelf break front from the 1996 PRIMER experiment (Gawarkiewicz et al. 2004; Sperry et al. 2003) is shown in Fig. 2.6. This front obviously has a lot of structure, unlike our simple cartoon in Fig. 2.2(3), and serves

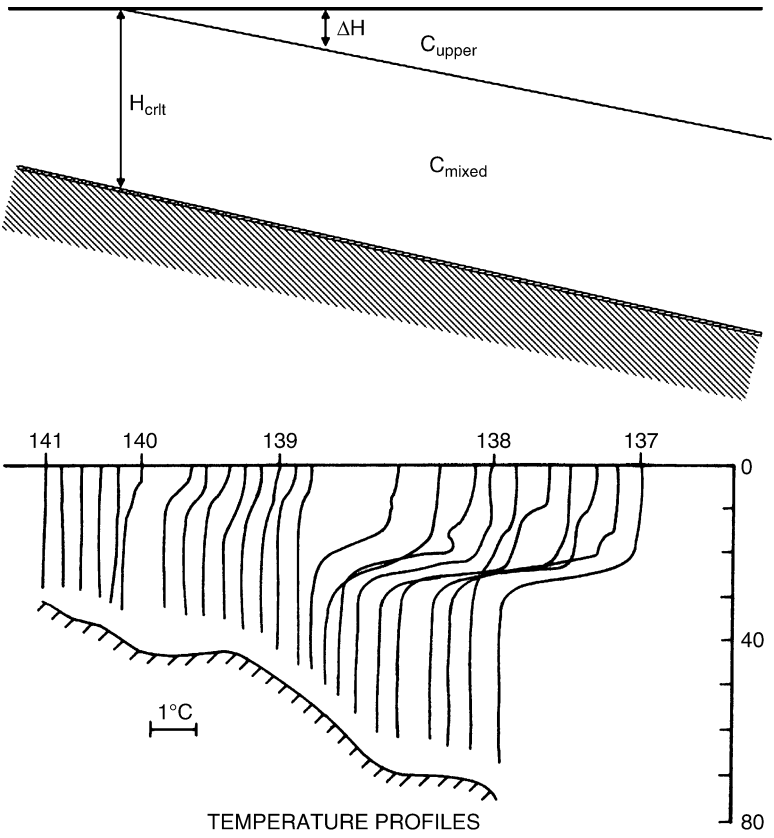


Fig. 2.5 *Top panel* – Simple schematic of a coastal mixing front where the bathymetry is deepening. Past the critical mixing depth, a surface layer forms, resulting in a front. *Lower panel* – Data showing a coastal mixing front from Bowden, referencing Schumacher et al. (1979). Horizontal axis is degrees in longtitude

to warn us that such simple models will often need expansion to cope with the intricacies of the real world.

Shelf break fronts are rather complicated entities, and their genesis, maintenance, structure, and variability are still topics of great interest to the physical oceanographic community. We will make only a brief mention of the genesis and maintenance of these fronts, as these topics are more of interest to physical oceanographers, but will discuss their structure and variability in more detail, as these are of direct acoustic importance.

Historically, the existence of shelf break fronts was explained by *assuming* that there were two horizontally distinct water masses near the shelf break, and that the front and the flow (mostly along the shelf) associated with it were generated and maintained by a mechanism called "geostrophic adjustment," whereby the density gradient force and Coriolis forces balance each other. An alternative to this "almost assuming the answer" type of theory was given by Gawarkiewicz and Chapman

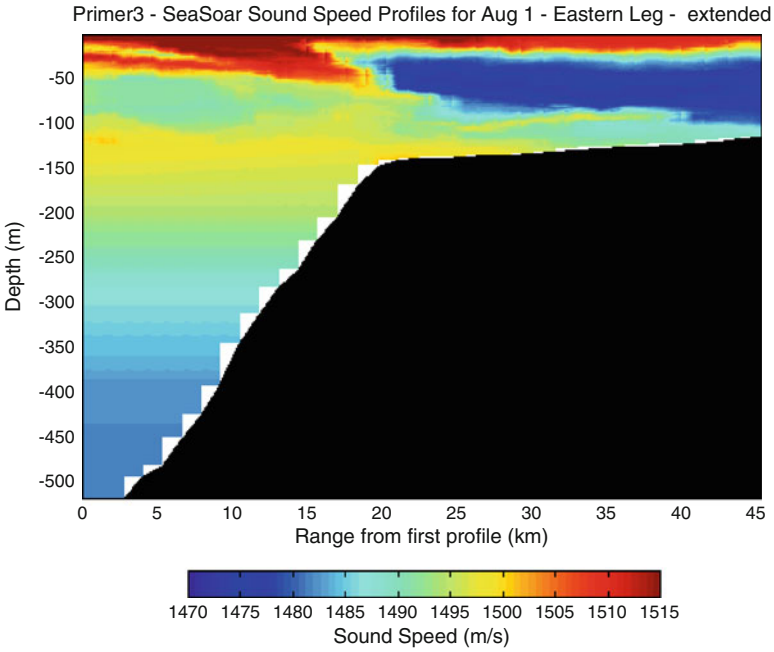


Fig. 2.6 Shelf break front in the Mid-Atlantic Bight measured by SeaSoar undulating CTD as part of the PRIMER experiment in 1996. Warm slope water (to the *left*) meets cold shelf water (to the *right*), creating the front. Much additional structure is seen, in the form of (1) a warm surface mixed layer, (2) a "foot of the front" protrusion of warm, salty water onto the shelf near the bottom, and (3) a 0–50-m depth "downwelling cell" of very warm water just seaward of the shelf break. These additional features all have significant acoustic impact

(1992), who showed that by looking at how the bottom friction affects flows near the shelf break, one could generate and maintain shelf break fronts with absolutely no assumption of horizontally distinct water masses. The Gawarkiewicz/Chapman theory provides a very satisfying explanation of these fronts, but like our mixed layer model above, it is too simple to provide all the acoustically important details that one sees in the real ocean, as, for example, in Fig. 2.6. Larger, data-assimilating "primitive equation" regional oceanographic models can give better pictures of real frontal structure, but as was the case with the mixed layer (discussed above), the amount of input needed to make such models describe a front in adequate detail can be large. Because of this, our predictive capability, even while using large models, must honestly be described as limited at this time.

The dynamics of shelf break fronts are interesting for multidisciplinary acousticians to learn, but the space–time structure of the shelf break front is of far more practical concern. The toy "vertical wall" model that we showed in Fig. 2.2(3) is fine for examining the acoustical physics of fronts in a simple way, but is inadequate in general. In particular, it misses many of the important features of the real front, particularly their across-shelf slope, their along-shelf structure, and

their variability and inherent stability with respect to perturbations. We examine these next. It is observed that coastal fronts, including shelf break fronts, have a bottom-to-top slope on the order of 10^{-2} (Bowden 1983). For shelf break fronts, the bottom or "foot" of the front is at or slightly inshore of the shelf break, with the surface expression being $\sim 10^2 H$ (where H is the water depth at the foot) seaward. This gives us about a 10-km frontal width for $H = 100$ m, a not unreasonable number considering five to ten kilometers is a common result. Going along the front in the along-shelf direction, the first-order assumption based on simple models is that the front is laterally symmetric, i.e., has an infinite correlation length in that direction. But, again, that is a gross oversimplification, and in fact along-shelf correlation length can be as short as the across-shelf length, as it was for the PRIMER experiment front shown in Fig. 2.6. In that experiment, a radially symmetric Gaussian correlation function with scale length ~10 km actually fit the frontal and near-frontal oceanography structure rather well (Sperry et al. 2003). The exact correlation length of shelf break fronts in the along-shelf direction is generally a function of how much the system is perturbed by eddies and other turbulent ocean processes so that it is hard to ascribe a number a priori. Our guess of "between the across-shelf scale length and infinity" is rather unsatisfactory, but it is probably reasonable. As to the stability of these fronts, they are found to reform after being severely perturbed in hours to days, so we can ascribe some real persistence and permanence to them.

Eddies and filaments. Continental shelves, particularly broad western continental shelves, are often located in close proximity to large, strong deep-water current systems. Two spectacular western boundary current examples are the Gulf Stream and the Kuroshio in the Northern Hemisphere, which border the US and Asian shelves, respectively. These currents routinely spawn large eddies and filaments, many of which travel into shallower water (to a distance determined in large part by their vertical extent compared to the shelf bathymetry) and create large sound speed disturbances. Thus, knowing what the onshore eddy and filament radiation field is in a coastal region is important to shallow water acoustics.

For the purposes of acoustics, we would like to have the space, time, and sound-speed perturbation scales of these features. In a somewhat oversimplistic view, the sound-speed perturbation is dictated simply by the differences between the water masses of the eddy/filament and the shelf water. In the Mid-Atlantic Bight (MAB), sound-speed differences of 30–40 m/s between shelf and eddy waters are typical. Obviously, the perturbation magnitude will be highly regionally dependent.

In terms of the (horizontal) spatial scales of the eddies and filaments, one goes from the comparatively large scales of the deep-water features (e.g., ~50 km diameter for Gulf Stream eddies and filaments) to smaller scales on the shelf (5–7 km was typically observed in the MAB). The radius R of these objects is roughly given by the internal deformation radius defined previously, $R = (g'H)^{1/2}/f_C$. In the vertical direction, one can see a large variety of scales, going from a few meters to full water depth. Again, the variability seen is regionally dependent and complicated.

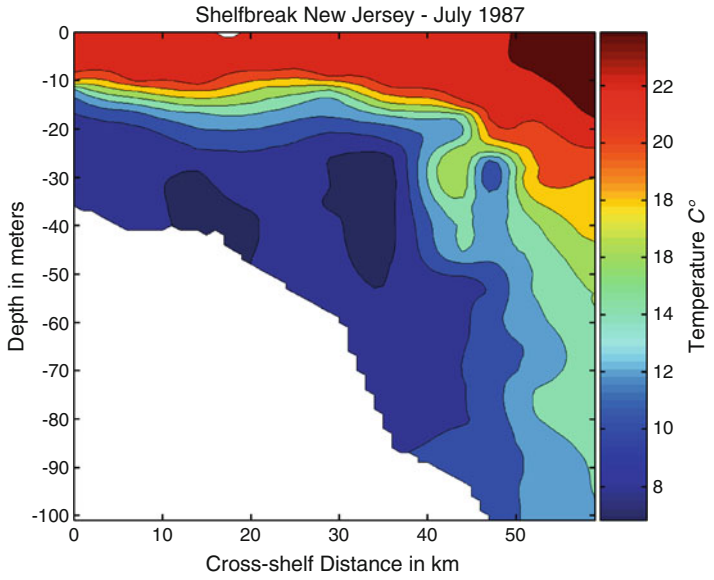


Fig. 2.7 Horizontal acoustic duct created by ocean filamentary feature. Horizontal temperature differences over ~5 km are from 7 to 10°C, or up to 40 m/s in sound speed

In that eddies and filaments are turbulence phenomena, the examination of a turbulence spectrum would seem to be a sensible way to characterize these irregular features. Rather interestingly, looking at the 2D wavenumber spectra of MAB eddies and filaments from satellite, Advanced Very High Resolution Radiometer (AVHRR) thermal imagery has produced k^{-2} spectra, characteristic of sharp-edged features. Vertical wavenumber spectra are harder to obtain, and we know of no readily available estimates of the wavenumber spectrum for shelf eddies and filaments. This is an area that needs further investigation for acoustics purposes, as well as for the other oceanographic disciplines.

We should note that filamentary features can also create acoustic ducts; an example is shown in Fig. 2.7. In this figure, one sees a warm water filament isolating (in the x–z plane) a parcel of cold water, which creates a duct between the warm filament and the warm front. This duct "into the page" way extends for many kilometers and is likely to be curved in the x–y plane as well.

Surface ducts. There are three types of surface ducts one encounters in shallow water that might be of importance to low- and mid-frequency acoustic propagation (1) bubble cloud ducts, (2) pressure gradient ducts, and (3) surface buoyancy plume ducts.

It is well known that a surface bubble layer is created by wave breaking and spilling during wind and storm events, and that this layer has an exponentially decaying profile and a depth on the order of a meter. Farmer and other ocean bubble researchers have made simultaneous measurements of the bubble population and size distribution as a function of depth and time with multifrequency inverted echo

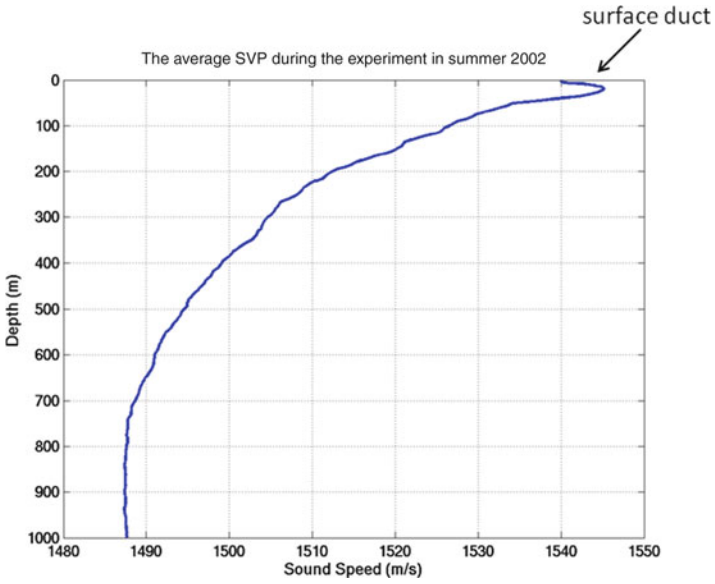


Fig. 2.8 Surface duct created by Mississippi River outflow off the coast of Louisiana. The duct of diluted fresh water is only 5 m/s in strength

sounders, which allow the calculation of the resulting (dispersive) sound-speed anomaly profile. Using the surface ambient noise that is created by such wave events, Farmer and his colleagues were able to show the trapping of some of the ambient sound in the 40 Hz–20 kHz range, with the distinctive modal structure one expects from waveguide propagation (Farmer and Vagle 1989). This is obviously an intermittent and highly variable sound duct, but a real one nonetheless, particularly as frequency increases.

A pressure gradient surface duct can be created when the water column is isothermal and isohaline over the top 10–20 m (typical of a well-mixed surface mixed layer), giving the 0.017 s^{-1} pressure gradient a chance to form a slightly upward refracting surface duct. This duct can channel sound from near-surface sources quite effectively, especially in calmer conditions where surface scattering out of the duct is unlikely to happen. This duct is often called "the surface duct" by mid-frequency sonar operators, as this has the biggest effect in that frequency range. Again, this is an intermittent duct that depends upon a restricted range of conditions for its formation. The range and extent of such a duct are likely to be dictated by the correlation length of the surface mixed layer, a number not easily available.

Buoyant river plumes are a common feature on continental shelves, and are well studied at this point in time. An example of a river plume surface layer signal is shown in Fig. 2.8. It was observed by DeRuiter et al. (2006) that this plume was able to trap seismic airgun signals with frequencies over 300 Hz, the cutoff frequency of this particular duct. Thus, we would not only note these ducts as being viable shallow water features, but would also caution that they are (at present) hard to predict.

Storm surges (*coastally trapped waves*). There is perhaps one more rather large-scale ocean feature that bears discussion here, i.e., storm surges, which can raise the sea level several meters. It is well known that storms push water away from them, and that the water pushed by the storm can precede the storm in arrival. This is due to the storm surge being describable by a Kelvin wave, a coastally trapped wave that has wave speeds given by $c = (gH)^{1/2}$, which gives $c = 25$–45 m/s in waters 60–200 m deep. Since storms typically travel at 10–15 m/s, the surge usually outstrips the storm and is seen first. The coastally trapped Kelvin wave also has the highest amplitude at the shore and decays away in amplitude exponentially with distance from the shore. Thus one sees the large, and often perilous, surges near the shore (several meters), but less of an effect further out (1–2 m). The state of the art in forecasting storm surges is actually rather good at this point, as they have a pronounced human impact. Thus, for the purposes of shallow water acoustics, we can take this as a known and understood process.

2.4 Dynamics of the Ocean Surface: Surface Waves

Surface waves are perhaps the most familiar phenomena from physical oceanography and also of great importance to shallow water acoustics, since they introduce a range-dependent water depth and waveguide boundary condition. Again, a vast literature on the topic exists, and we must constrain our treatment here to the "useful nuggets" for SW acoustics. A short section on the basic theory of surface waves is presented in Appendix A; here, we will only consider shallow water applications of surface waves.

Let us first look at the case of 1D surface waves to illustrate some simple concepts before moving to the more realistic case. Considering only waves traveling in the x direction, the elevation of the sea surface $\zeta(x,t)$ may be written in a discrete sum form as

$$\zeta(x,t) = \sum_{n=1}^{\infty} a_n \exp[i(\tilde{q}_n x - \Omega_n t + \varphi_n)], \tag{2.17}$$

where a_n is the amplitude of the nth component of the wave spectrum which has wave number $\tilde{q}_n$, frequency Ω_n, and a random phase φ_n which ranges from 0 to 2π uniformly. The dispersion relation relating the frequency and wavenumber is [see (A.35)]:

$$\Omega^2 = g\tilde{q} \tanh \tilde{q}H, \tag{2.18}$$

where g is acceleration of gravity and H is depth. We should note that for coastal waters, where the depths we consider are 10–200 m, the dispersion relation most generally falls between the long wave (shallow water) and short wave (deep water)

limits, and that one is generally well advised to use the full dispersion relation in (2.18), rather than one of the limiting cases. However, the deep-water dispersion relation $\Omega^2 = g\tilde{q}$ is reasonable for many shallow water cases; so if one needs to approximate, this is probably the better form.

If we now extend the wavenumber spectrum to two dimensions, i.e., the x–y plane [see (A.62)–(A.66)], we can define a wavenumber–frequency spectrum, $G(\tilde{\vec{q}}, \Omega)$, where $\tilde{\vec{q}} = (\tilde{q}\cos\alpha, \tilde{q}\sin\alpha)$ is the magnitude of the 2D wavenumber vector, and α is the azimuthal direction. The most usual measurement of waves is the "polar" spectrum containing both frequency and direction information, which we can relate to the wavenumber–frequency spectrum by

$$G(\Omega, \alpha) = 2\int_0^\infty G(\tilde{\vec{q}}, \Omega)\tilde{q}d\tilde{q}. \tag{2.19}$$

This quantity is the one that is measured by standard instruments and is of most use to acoustics. For this spectrum, the variance of the sea surface height is determined as

$$\sigma^2 = \int_0^{2\pi}\int_0^\infty G(\Omega, \alpha)d\Omega d\alpha. \tag{2.20}$$

An often-quoted measure of wave displacement, the significant wave height H_s, which tells one what the average height of the biggest 1/3 of the waves is, is related to this by

$$H_s = 4(\sigma^2)^{1/2}.$$

Often wave buoys do not have tilt sensors in them and just measure the frequency content of the surface waves via accelerometers. In this case, one has effectively integrated the frequency-directional spectrum over angle, giving

$$G(\Omega) = \int_0^{2\pi} G(\Omega, \alpha)d\alpha. \tag{2.21}$$

Having defined formally what the wave spectra are, it is also useful to describe some "popular parameterizations" of these quantities, as these are often employed in calculations. This is generally done by multiplying a wave-directional spectrum by a wave frequency spectrum. Two popular parameterizations of the wave frequency spectrum are the Pierson–Neuman and the Pierson–Moscovitz spectra. The Pierson–Neuman spectrum is

$$G_{PN}(\Omega) = C\Omega^{-6}\exp\left[-2\left(\frac{g}{\Omega v}\right)^2\right], \tag{2.22}$$

where $C = 2.4\,\mathrm{m}^2/\mathrm{s}^5$, and v is the wind speed in m/s. The more recent Pierson–Moscovitz spectrum is given by

$$G_{\mathrm{PM}}(\Omega) = 8.1(10^{-3})g^2\Omega^{-5}\exp[-0.74(g/\Omega v)^4]. \tag{2.23}$$

The directional spectrum, denoted as $G(\tilde{q},\alpha)$, is often given by a cosine-to-a-power form, i.e.,

$$G(\tilde{q},\alpha) = \begin{cases} b\cos^{v(\tilde{q})}\alpha, & |\alpha| \le \pi/2 \\ 0, & |\alpha| > \pi/2 \end{cases}. \tag{2.24}$$

In (2.24), $v(q)$ varies from 10 at low frequencies to 2 at the higher frequencies. The factor b is determined by the following normalization condition:

$$\int_{-\pi}^{\pi} G(\tilde{q},\alpha)\mathrm{d}\alpha = 1. \tag{2.25}$$

In addition to the cosine form, a sech^2 form has also been used for coastal spectra, and we refer the reader to Donelan et al. (1985) if this form is of interest.

As a last note about surface wave spectra in shallow water, the distance from the measurement site to the shore, called the fetch, often needs to be considered. A good account of the evolution of coastal wave spectra (including fetch and other coastal effects) can be found in Hasselmann et al. (1980), which describes the results of the Joint North Sea Wave Project (JONSWAP).

2.5 Dynamical Processes Inside the Ocean: Tides and Internal Waves

Tides. There are two types of tides of interest to shallow water acoustics: barotropic and baroclinic. A barotropic tide affects the entire water column uniformly, whereas a baroclinic tide produces a nonuniform vertical structure. The barotropic tides are the most familiar to us and so we will address them first.

The barotropic tide produces three effects of interest to shallow water acoustics (1) a periodic increase and decrease of the water depth, (2) a current field, and (3) an advection of ocean structure along the tidal ellipse. The amplitude and phase of the tidal signal (both in water depth variation and current) are a strong function of where one is geographically, as the local coastline and bathymetry are major factors in the tidal equations. Thus, we cannot provide any more useful guidance to the reader than to consult local tidal tables when doing an experiment. Moreover, these tables will probably be far more accurate in describing water depth variation than the current, as tide gauges are common along coastlines, whereas current measurements are sparse. (However, this is fortuitous, in that current is a

second-order determinant of sound speed.) As to the periodicity of the tidal signal, that too is locally variable, though there are two major tidal components that dominate – semidiurnal (12.4 h) and diurnal (25 h). One most often sees the semidiurnal tide, though in many locations the diurnal tide or mixed diurnal–semidiurnal tides are observed. Again, this is locally variable and the rule is to look at local tide charts. Since the barotropic tide is a well-known and long investigated phenomenon, we will simply refer the reader to some of the standard introductory texts for more detail. Specifically, the texts by Pond and Pickard (1983) and Bowden (1983) provide a good starting point. We now turn to the acoustic effects of barotropic tides.

The first tidal effect, the water depth variation, is often of importance acoustically even though it would seem that the ~1-m variation typical of coastal regions should be negligible. But an ~1% perturbation of the water column depth can perturb modal eigenvalues and travel times just as much as a significant perturbation in either the bottom or the water column sound speeds, and so must be taken carefully into account when using modal inverse methods for obtaining medium properties. In general, one must look to how the acoustic quantity used scales with water depth to see how the tidal signal affects a particular application. Turning to the currents, there is again a wide local variability, but an average value of ~0.5 m/s (or 1 knot) would not be far off the mark for coastal waters. As this is a fraction of a percent of the water column sound speed (~1,500 m/s), the current usually does not need to be considered when looking at the sound speed profile of a particular shallow water region. Current-related source-speed effects, however, can be considerable in that speeds of fish, marine mammals, ships, AUVs, etc., are of the order 1–10 m/s, which makes the tidal current a 5–50% effect in reference frames where it can be noticed. Finally, we mention the horizontal advection of ocean structure by tidal currents, particularly the temperature structure to which acoustics is most sensitive. If we take ocean structures such as eddies, fronts, and internal waves to be frozen in shape, the tides will move these features around their tidal ellipses. The largest excursions are represented by the time integral of the tidal velocity components (along-shelf and across-shelf being the generally preferred directions) over a half period, as the tide then reverses direction. These excursions are of the order of 1–10 km, which is significant, as the coastal ocean commonly has significant horizontal sound-speed structure gradients over these scales.

We next turn to the baroclinic tides, or the so-called internal tides in shallow water. Our dealings with the baroclinic tide will be brief in this section, simply because the internal tide is an internal wave of tidal frequency, and so is most appropriately included in our discussion of internal waves. However, a few points should still be made here, on both the oceanography and the acoustics, before we pass to the internal wave section. First, the linear coastal internal tide often looks like a long wavelength (10–40 km), two-layer system (less dense water above more dense water) sinusoidal wave propagating at a speed of (as a rough average) 0.25–0.5 m/s from the shelf break toward the shore, with an amplitude of 5–10 m. Second, the internal tides can have “critical latitudes” beyond which they cannot exist as propagating disturbances, i.e., they become evanescent,

decaying waves. Third, the baroclinic tides are far more variable than the barotropic tides (due to their dependence of their propagation properties on the complex thermohaline and current structure of the ocean through which they move) and so are far less predictable. One does not find internal tide atlases, though there is now a fairly extensive literature on such tides, and indeed an atlas for nonlinear internal tides recently appeared (Jackson and Apel 2002; Jackson 2004).

Acoustically, linear internal tides tend to create a slowly evolving range dependence of the acoustic waveguide, and so can generally be accounted for using adiabatic normal mode theory. However, the nonlinear internal tides create sharp horizontal gradients in the ocean structure, producing acoustic effects which are much more substantial and interesting.

Internal waves. Elements of the general theory of internal waves in a stratified liquid are considered in Appendix A. In this chapter, we will consider properties of the internal waves on the oceanic shelf and their connection with the acoustic properties of a shallow water waveguide. To begin, let us revisit the properties of the seawater taking into account its compressibility.

We begin with an equilibrium density depth distribution $\rho = \rho_0(z)$, formed by taking into account the compressibility of water. We then consider the adiabatic displacement of a selected volume dV (the mass of this volume is $dm = \rho_0(z)dV$) by depth dz. Its density at the depth $z + dz$ is changed by $d\rho = \beta_a \rho_0^2(z) g dz$ [see (2.2)]. Using the connection between the compressibility and the sound speed, we can write the resulting density variation of the displaced volume as

$$\delta\rho = \left(\frac{\partial\rho_0}{\partial z} - \frac{\rho_0 g}{c^2}\right)\delta z \tag{2.26}$$

which contains a part (the second term on the right side) formed by hydrostatic pressure.

The "Archimedes force" dF_A (i.e., the buoyancy, or restoring force) exerted on this volume has the value

$$dF_\mathrm{A} = \rho_0 dV\left(\frac{g}{\rho_0}\frac{d\rho_0}{dz} - \frac{g^2}{c^2}\right)dz. \tag{2.27}$$

So for the frequency of harmonic oscillations [the Vaisala frequency, introduced in (A.104)], we have the expression

$$\tilde{N}^2(z) = \frac{g}{\rho_0}\frac{d\rho_0}{dz} - \frac{g^2}{c^2}. \tag{2.28}$$

We now introduce the so-called potential density $\rho_\mathrm{p}(z)$ by the equation

$$\frac{\partial\rho_\mathrm{p}}{\partial z} = \frac{\partial\rho_0}{\partial z} - \frac{\rho_0 g}{c^2}. \tag{2.29}$$

If water is taken to be incompressible or gravitation is ignored, then the real density and potential density are the same. In other words, the potential density can be formed by the compressible density if we throw out gravitation. The difference between the real density and the potential density is called the adiabatic density. The stability of the water column is determined by the potential density, as well as the oscillations in the vertical plane and the Vaisala frequency. In a similar way, we can also introduce the potential and adiabatic distributions for the temperature and salinity depth dependence.

The second term on the right side of (2.28) is of the order ~4×10^{-5} s^{-2}, whereas the first term can be ~10^{-3}–10^{-4} s^{-2}. Thus, for shallow water (or more exactly for upper ocean layers – several hundred meters), we can neglect the second term in the Vaisala frequency and thus use (A.94). We will use (A.94) in our following discussions of internal waves.

Let us now consider some numerical estimates. In shallow water, we can typically have temperature jumps of several degrees (3–5°C) over a depth interval of 10–20 m. This means that the temperature gradient can be 0.1–0.5°C/m. From this, it is possible to estimate the relative density gradient and then the Vaisala frequency $\tilde{N} \approx \sqrt{\frac{g}{\rho_0}\frac{d\rho_0}{dz}} = \sqrt{g\beta_T \frac{dT^\circ}{dz}} \leq 10.$ cph.

Using the aforementioned connections between the thermodynamic parameters of seawater, (2.2)–(2.6), we can establish the relation between internal waves and the fluctuations of the sound-speed profile, an important relation for underwater acoustics.

We begin with some equilibrium distribution of the water layer parameters: $\rho_0(z)$, $T^\circ(z)$, $c(z)$. If we have a displacement of a selected water volume by δz, then the corresponding fluctuation of the water density at a given depth (neglecting small corrections due to compressibility) is $\delta\rho = \frac{d\rho_0}{dz}\delta z$; correspondingly, the sound speed will be changed by $\delta c = \frac{\partial c}{\partial z}\delta z$.

As mentioned, density and temperature fluctuations are connected by $\delta\rho = \frac{\partial\rho_0}{\partial T^\circ}\delta T^\circ = \rho_0\beta_T\delta T^\circ$. For seawater, we have that $\beta_T \sim 1.3 \times 10^{-4}\,(^\circ\mathrm{C})^{-1}$. If we use these, i.e., the coefficient of thermal expansion and the variability of the sound speed with temperature, then we obtain

$$\frac{\delta c}{c} = \frac{1}{c}\frac{\partial c}{\partial T^\circ}\delta T^\circ = \frac{1}{c}\frac{\partial c}{\partial T^\circ}\frac{1}{\rho_0\beta_T}\delta\rho = \frac{1}{c}\frac{\partial c}{\partial T^\circ}\frac{1}{\rho_0\beta_T}\frac{\partial\rho}{\partial z}\delta z = \frac{1}{gc}\frac{\partial c}{\partial T^\circ}\frac{1}{\beta_T}\tilde{N}^2(z)\delta z.$$

If we denote

$$\tilde{Q} = \frac{1}{gc}\frac{\partial c}{\partial T^\circ}\frac{1}{\beta_T} \tag{2.30}$$

then

$$\delta c = \tilde{Q}c\tilde{N}^2(z)\delta z. \tag{2.31}$$

Using the range of values we considered, we can estimate numerically that $\tilde{Q} \approx 2.4\,\mathrm{s}^2/\mathrm{m}$. In the monograph edited by Flatte (1979), it is shown that this constant does not vary significantly with temperature and depth.

Varieties of internal waves. Due to the rotation of the Earth, the so-called inertial waves of frequency $\Omega_i = 2\Omega_\mathrm{E} \sin\theta$ exist, where $\Omega_\mathrm{E} \sim 0.04$ cph is the rotation frequency of the earth and θ is the latitude. These inertial waves define the lowest propagating wave frequency in the IW band, whereas the biggest Vaisala frequency defines the top of the propagating IW band. The inertial frequency varies from two cycles per day at the poles to zero cycles per day at the equator, whereas the maximum buoyancy frequency is usually on the order of 10–20 cph. This means that in the equation determining the internal waves, (A.108), $\Omega_i \ll N$ in the denominator. However, we cannot simply throw this term out, because (as will be seen in the following discussion) inertial oscillations often appear strongly in the IW spectrum, providing they are obtained via long-time observations (several days or more). Thus, we can neglect inertial waves for most experiments with durations on the order of several hours, but we should take inertial waves into account for experiments on the order of 50–100 h or more. We would also note that inertial oscillations are due to water being accelerated, with the dominant mechanism being sudden wind events. Additionally, the particle paths for inertial waves are horizontal so that their effect on acoustics is actually minimal.

Two flavors of propagating IWs are found in stratified coastal waters: linear and nonlinear waves. The linear waves, found virtually everywhere, obey a standard linear wave equation for the displacement of the surfaces of constant density (isopycnal surfaces). The nonlinear IWs are generated under somewhat more specialized circumstances than the linear waves and obey a wide variety of wave equations (Apel et al. 1995). The simplest description is the familiar Korteweg-deVries equation (KdV), which governs the horizontal components of the nonlinear internal waves. Both types of internal waves can be illustrated by a simple two-layer model (see Appendix A), i.e., the three-layer model with the thermocline shrunk to zero extent. Using the two-layer model, we can look at some typical wave parameters. For a more or less common situation, where $H_1 = H_2$ and $\Delta\rho/\rho = 2.0 \times 10^{-3}$, one obtains that ratio of speeds of internal and surface waves is $\tilde{c}_\mathrm{int}/\tilde{c}_\mathrm{surface} = 1/45$; typical wave speeds would be $\tilde{c}_\mathrm{int} = 0.8\,\mathrm{m/s}$ and $\tilde{c}_\mathrm{surface} = 36\,\mathrm{m/s}$. However, while the small density contrast between ocean layers makes the internal wave speed small, it also contributes to making internal wave amplitudes large compared to surface waves. Typically, internal wave amplitudes are of order 5–10 m in coastal regions; however, some coastal nonlinear internal waves can have amplitudes of 100 m or more! Also, the dispersion relations for surface and internal waves show that the internal waves will generally have much longer wavelengths than surface waves. The highest frequency linear internal waves have $\lambda \approx 100\,\mathrm{m}$, whereas the low-frequency "internal tides" have wavelengths on the order of tens of kilometers. By comparison, surface wave wavelengths are tens to hundreds of meters.

Linear internal waves. We next look at the spectrum of the linear internal waves in shallow water, following Colosi et al. (2001). In doing this, we will look at a continuously stratified medium, as opposed to the two-layer simplification. Currently, a popular approach is to use a wavenumber spectrum much like the well-known Garrett and Munk (1975) spectral description appropriate for deep water (Yang and Yoo 1999). A prime descriptor of the IW is the amplitude of the displacement of a water layer of constant density, as a function of the coordinate in the horizontal plane and time, $\zeta_{\mathrm{IW}}(\vec{r}, t)$ (see Appendix A). We consider the internal wave field to be a linear superposition of waves of the form,

$$\zeta_{\mathrm{IW}}(\vec{r}, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} d\tilde{q}_x d\tilde{q}_y \sum_{l=1}^{\infty} S_l(\vec{\tilde{q}}) \Phi_l(z; \tilde{q}) \exp\left[i \left(\vec{\tilde{q}} \vec{r} - \Omega_l(\tilde{q}) t \right) \right], \tag{2.32}$$

where $\Omega_l(\tilde{q})$ is the internal wave dispersion relation, $\Phi_l(z; \widetilde{q})$ are the internal wave eigenmodes, and $S_l(\vec{\tilde{q}}) = S_l(\widetilde{q}_x, \widetilde{q}_y)$ are the amplitude distribution of the internal waves, taken to be complex zero mean Gaussian random variables with cross spectral density

$$\left\langle S_l(\vec{\tilde{q}}) S_m(\vec{\tilde{q}}') \right\rangle = G_l^{\mathrm{s}}(\vec{\tilde{q}}) \delta_{lm} \delta(\vec{\tilde{q}} - \vec{\tilde{q}}'). \tag{2.33}$$

The quantity $G_l^{\mathrm{s}}(\vec{\tilde{q}})$ is the spectral density of the random field of internal waves. We consider this value to be the main characteristic of the random IW field. In the following discussion, it will be called the "spectrum" of the IW. The modes $\Phi_l(z; \widetilde{q})$ in (2.32) are obtained by solving the linear internal wave equations (A.108)–(A.110) with the appropriate input for $N(\widetilde{z})$.

We note that there are several variants of the mode (A.108) extant in the literature, depending on exactly what physics are included in the modes. One commonly sees (A.108) without the Coriolis parameter in the denominator, which ignores the effects of the Earth's rotation (as we discussed). One can also add the important effect of vertical current shear to (A.108), which results in the so-called Taylor–Goldstein equation. Equation (A.108) is easily solved with standard finite difference techniques. It should be noted that shallow water has a very "red" IW modal spectrum. Because of this, one generally needs to look at only the lowest few modes, often just the first.

The function $G_l^{\mathrm{s}}(\vec{\tilde{q}})$ is called the spectrum of the internal waves, and has the physical meaning of the energy density of the internal waves in a given gravitational mode for a given wave vector in the horizontal plane. It is one of the most important characteristics of the internal wave field. In deep water, this spectrum has been best described by the work of Garrett and Munk (1975). In shallow water, there have been many different attempts to extend this spectrum. However, it does not appear possible to find a common spectrum appropriate for all shallow water areas, where there are significant sources and sinks, directional waves, etc. But one still needs something to work with in SW, and so we describe one of these efforts as an example.

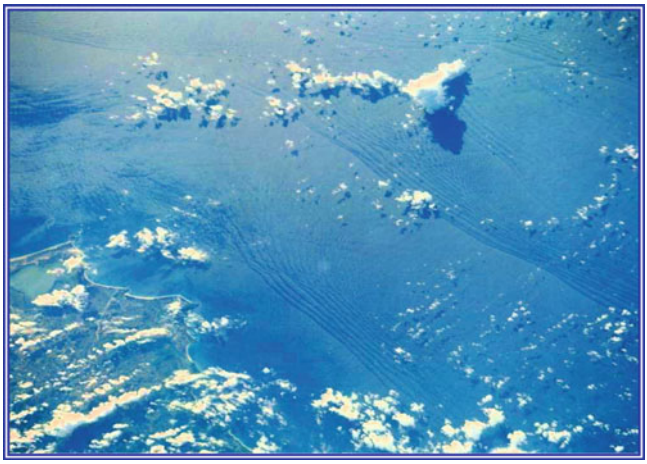


Fig. 2.9 An observation showing the parameters of internal solitary waves as presented in the work by Xiaofeng Li et al. (2000)

As mentioned, Yang and Yoo (1999) have shown that a useful model, which fits the comparatively sparse existing data on shallow water linear internal waves, looks like the GM spectrum, with

$$G_j^{\mathrm{S}}(\tilde{q}_x, \tilde{q}_y) = \varsigma_0^2 \frac{\tilde{N}_0}{\tilde{N}(z)} \frac{M_l}{(j^2 + j_*^2)^q} M_{\tilde{q}} \frac{\tilde{q}_j^{p-2} (\tilde{q}_x^2 + \tilde{q}_y^2)^{1/2}}{(\tilde{q}_x^2 + \tilde{q}_y^2 + \tilde{\sigma}_z^2)^{\frac{p+1}{2}}}, \tag{2.34}$$

$$\tilde{q}_j = \frac{\pi f j}{\tilde{N}_{0B}} \quad \tilde{N}_{0B} = \int_0^H N(z) dz, \tag{2.35}$$

where H is depth of the waveguide, $j^* \sim 1$, $q \sim 1$, $p \sim 2.7$, f_{C} is the Coriolis parameter, and M_j, $M_{\tilde{q}}$ are normalizations to provide a depth-dependent displacement variance of $\varsigma_0^2 \tilde{N}_0 / \tilde{N}$. We note that for the GM model, one has $j^* \sim 3$, $q \sim 1$, $p \sim 3$, $\zeta_0 \sim 7.3$ m, and $\tilde{N}_0 \sim 3$ cph. Also, note that (2.34) is a homogeneous, isotropic spectrum, which only depends on the magnitude of the horizontal wavenumber. An efficient scheme for the numerical evaluation of such spectra was provided by Colosi and Brown (1998) and could be adapted for shallow water environments of variable water depth and stratification.

In closing this section, we will note that an isotropic directional spectrum may not be a good assumption for the linear wave field that is found at the tail of a nonlinear internal wave train. This will be discussed later in the book.

Nonlinear internal waves. As mentioned many times, nonlinear internal waves are also common on continental shelves. Observations of coastal nonlinear internal waves are commonly shown, and we present two typical ones below, in Figs. 2.9 and 2.10 (airborne photos).

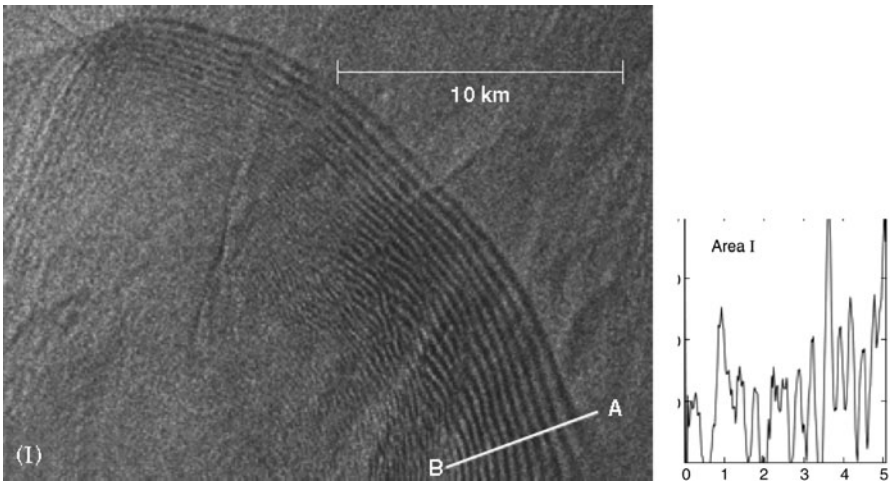


Fig. 2.10 Solitary waves often exist in the form of packets (or trains) propagating toward the coast line. An illustration of one of these trains on the NJ shelf is presented above

Table 2.2 Results of observations of coastal nonlinear internal waves

			Average wavelength per packet (m)		Number of wave crests				
N	Number of packets	Length (km)	1	2	1	2	Group velocity (m/s)	Tide period (min)	Water depth (m)
I	1	–	410	–	11	–	–	12:40	100
II	2	11.2	560	580	6	11	0.25	12:40	50
III	2	13.8	560	780	7	9	0.30	12:40	35
IV	2	9.5	630	760	9	7	0.21	12:40	35

The results of processing observations such as the one above in Fig. 2.10 are presented in Table 2.2, showing typical values of coastal internal solitary wave trains. We would note that these values, while "typical," also vary with location, times, and season.

One finds a vast literature available on this topic of nonlinear internal waves, indeed far larger than that concerning linear internal waves (which are oft-times harder to observe in shallow water!). The reader interested in more detail is referred to the review article by Apel et al. (1995) and the ONR workshop report authored by Duda and Farmer (1999) for initial treatments and a source of references.

For nonlinear internal waves, function $\zeta(\vec{r}, z, t)$ cannot be presented as a product of the vertical mode on the plane wave, similar to (2.32). Dependence on horizontal coordinates is determined by the nonlinear equation. Let us consider a function of displacement for a fixed depth, for example, $z_{\max}$, where function $\Phi_l(z)$ has a maximum. So we have a function of the horizontal coordinates

$$\zeta(\vec{r},t) = \zeta(\vec{r}, z_{\max}, t). \tag{2.36}$$

This function satisfies a nonlinear equation, depending on our model of the water layers. Many different equations and their solutions are given in the literature (Apel et al. 1995). For our purposes, we will just concentrate on one of the simplest yet most useful nonlinear internal wave equation for shallow water, the KdV equation, which depends only on one spatial coordinate: $\zeta(x,t)$. In the limits of small nonlinearity, weak dispersion, and long waves, one can derive expressions for the displacement of the isopycnal surfaces from their equilibrium levels due to internal waves. In Appendix A, a nonlinear KdV equation is obtained for a simple two-layered liquid. For arbitrary stratification, a similar equation can be used:

$$\frac{\partial\zeta}{\partial t} + \tilde{c}^{\text{ph}}\frac{\partial\zeta}{\partial x} + \alpha\zeta\frac{\partial\zeta}{\partial x} + \beta\frac{\partial^3\zeta}{\partial x^3} = 0 \tag{2.37}$$

where the parameters α, β can be expressed through the waveguide characteristics

$$\alpha = \frac{3\tilde{c}^{\text{ph}}}{2Q}\left\langle\left(\frac{d\Phi}{dz}\right)^3\right\rangle, \quad \beta = \frac{\tilde{c}^{\text{ph}}}{2Q}\langle\Phi^2\rangle, \quad \text{and} \quad Q = \left\langle\left(\frac{d\Phi}{dz}\right)^2\right\rangle. \tag{2.38}$$

In (2.38), Φ is the vertical mode function defined in (A.107)–(A.110), and $\tilde{c}^{\text{ph}}$ is the linear wave phase speed. Equation (2.39) has a very well-known solitary wave (single soliton) solution, which is

$$\zeta(\vec{r},z,t) = \zeta_0 \sec^2\left(\frac{x-\tilde{c}t}{\Lambda}\right). \tag{2.39}$$

In (2.39), ζ_0 is the amplitude, $\tilde{c}$ is the nonlinear wave velocity, and Δ is the characteristic width of the soliton. These are related to the linear speed and displacement via the relations

$$\tilde{c} = \tilde{c}^{\text{ph}} + \frac{\alpha\zeta_0}{3} \quad \text{and} \quad \Lambda^2 = \frac{12\beta}{\alpha\zeta_0}. \tag{2.40}$$

The single soliton solution is a very useful one for acoustics calculations, in that it can be implemented simply. One can also make *ad hoc* trains of solitons by stringing these individual solitons together spatially. The effects of these *ad hoc* soliton train solutions on acoustics are often quite close to that of more elegant oceanographic models, giving them a decided appeal for simple calculational studies.

For the continuously stratified medium case, we can numerically solve the KdV equation to get a time and space evolving train of solitons. The solution methods are standard and give results such as the one shown in Fig. 2.11.

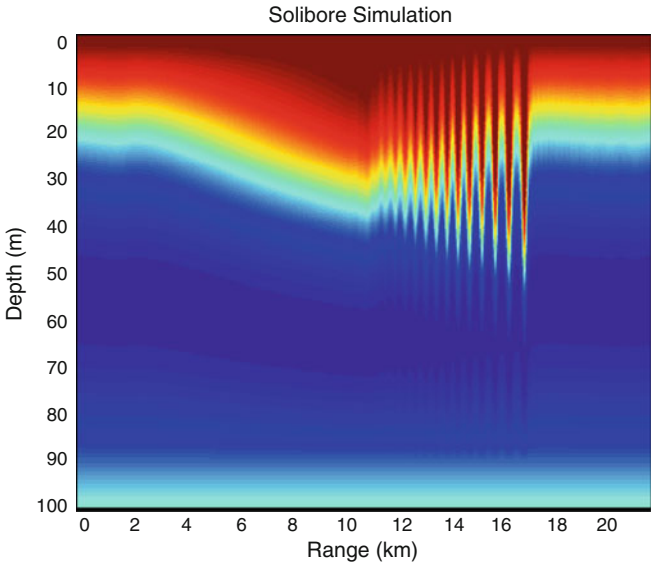


Fig. 2.11 Simulation of a train of solitons using the KdV equation. "Solibore" is a phrase coined to denote the fact that the train of solitons generated has both sharp solitons and a bore-like depression of the main thermocline

In concluding the discussion of internal waves, we will merely restate that the solutions shown above are some of the simplest ones available and as such should be useful for first-order acoustic calculations. However, there are many more complications and nuances of the internal wave field, and we would caution the reader that these simple solutions do not fully capture the real wave field. For this, one may have to pursue more detailed alternatives.

2.6 Experimental Studies of Coastal Internal Waves

We have so far considered the general properties of internal waves in a shallow water region. Next, we will illustrate our analysis by examples of detailed observations of the internal wave field in two shelf areas (the New York (NY) Bight and Kamchatka). Of the most practical use, we will show how it is possible to construct a model of the sound-speed field in a SW waveguide based on measured data; more exactly, we will see how we can get the spatial distribution of the sound speed using temporal data obtained at a fixed point (or several fixed points). In modeling the IW field for shallow water acoustics, it is best to use robust characteristics of the IW field, as measured in an experiment for a given region. There are generally two easy and robust measurements one can make: temperature

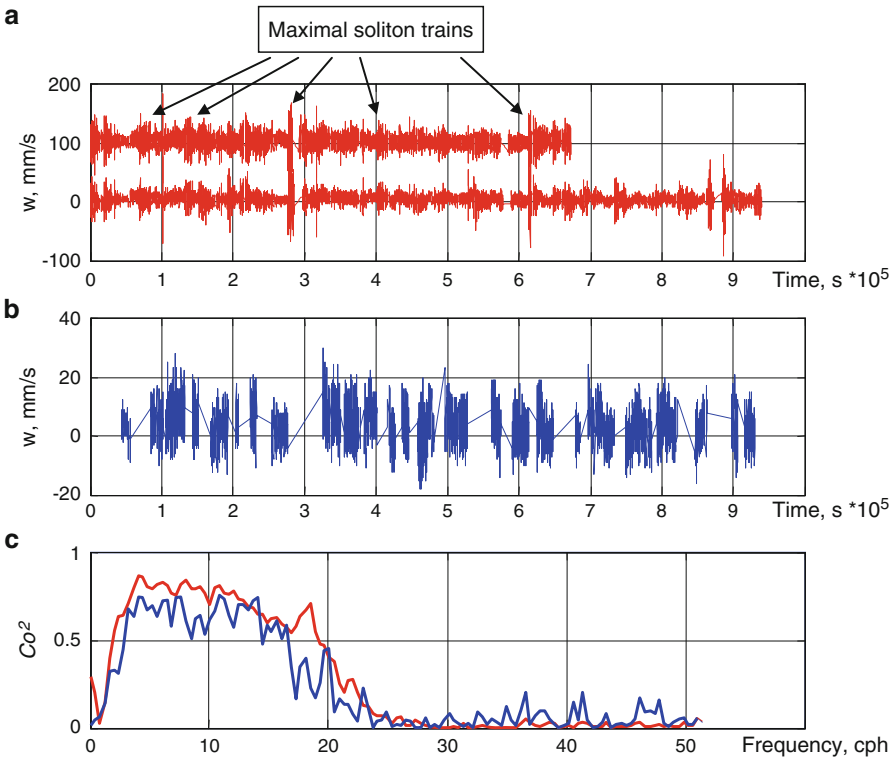


Fig. 2.12 (**a**) Portions of the vertical current velocity records for depth 17 m (*upper line*) and 28.7 m (*bottom line*), (**b**) – record of the "background" segments for depth 28.7 m, (**c**) – coherence function squared (Co^2) between records including soliton trains (*red line*), and between the same records in the background segments (*blue line*)

(simple, cheap thermistor devices abound) and current (e.g., using ADCPs to get vertical profiles of three-component currents is now common). We will look first at current measurements, and then go to temperature.

In discussing the NY Bight, we will examine current measurements of internal waves using some older current velocity meters (the so-called three-axis propeller type). These devices are generally placed at a few vertical points and give values of the three components of velocity $\vec{v}(\vec{u}, w)$ as a function of time. For our work, the vertical component $(w = \partial\zeta/\partial t)$ is obviously the most important one to consider. In Fig. 2.12, we show typical records of the vertical current velocity $w(\vec{r}, z, t)$ including the IW field. These two records were measured simultaneously in the Middle Atlantic Bight (MAB) (Kuzkin et al. 2006). The two current meters were placed at the same location but at different depths, $z_1 = 17$ and $z_2 = 28.7$ m, respectively. The water depth was $\approx$70 m. As may be seen from Fig. 2.12, soliton trains (the pieces with significant amplitude) compose a notable portion of the records.

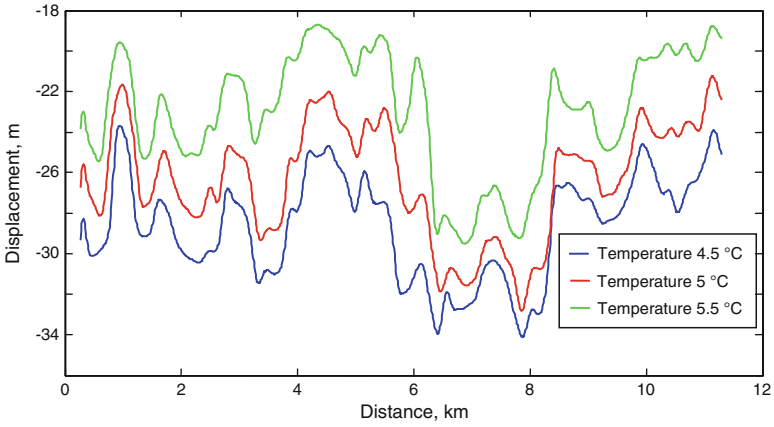


Fig. 2.13 Vertical displacements of isothermal surfaces off Kamchatka

To analyze these results, we will consider the soliton trains and the background waves as independent wave fields in a first approximation. We will also assume that the soliton train is a deterministic process and that the linear background IW field is a random field. Of course, there is significant variability in the nonlinear wave field from tidal period to tidal period (Colosi et al. 2001). However, over one tidal period, one can take the soliton train as a deterministic process for our current purposes. Based on this assumption, we separate the experimental record into background waves and soliton trains. Notice that there is some degree of subjectivity in the choice of the threshold for the separation of background and solitary waves. In order to define this threshold for our case, we can estimate the phase velocity of linear internal waves for our region. Using a simple two-layered model (A.123) and taking the appropriate parameters (depth $H \sim 70$ m, depth of thermocline $H_1 \sim 15$ m, etc.), we get $\tilde{c}^{\mathrm{ph}} \sim 35$ cm/s. We then take the threshold for the amplitude of the vertical current velocity to be less by one order of magnitude or 3.5 cm/s for frequencies less than 10 cph. (This corresponds to an amplitude of the vertical displacements of ~2 m.) For greater amplitudes, we assume that we have deterministic soliton trains, whereas for smaller amplitudes we suppose that we are dealing with a random realization of the background IWs. Results of this separation are shown in Fig. 2.12 for the vertical current velocity records. The coherence ratio[1] for the records with soliton trains and without trains is plotted in Fig. 2.12 as well. We see that a high coherence ratio is found in both cases. This is because the first gravity mode almost invariably dominates the IW field, both for the solitons and for the linear waves.

We can produce additional arguments in favor of this statement. In Fig. 2.13, we show results of measurements of the vertical displacement $\zeta(r, z, t)|_{z,t=\mathrm{const}}$ mode at

[1] Notice that coherence function of two random processes $w_1 = w(x, y, z_1, t)$ and $w_2 = w(x, y, z_2, t)$ for given values $x,\ y,\ z_1,\ z_2$ is defined in the Appendix.

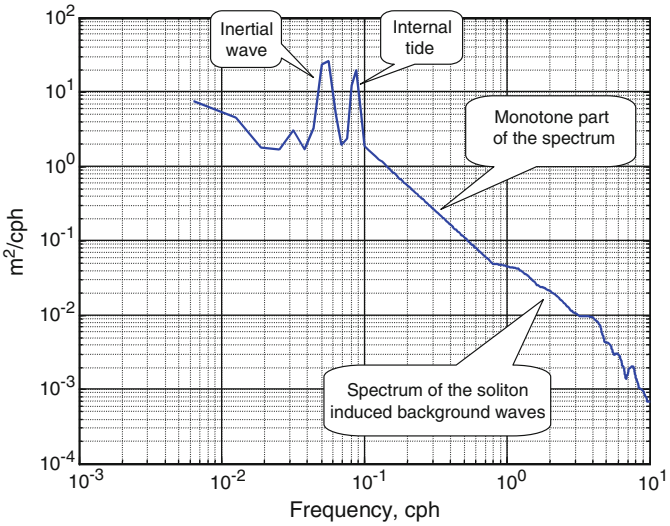


Fig. 2.14 The temporal spectrum of the background linear internal wave field

different depths near Kamchatka. The measurement was carried out using a towed vertical thermistor string. As seen from the figure, the isothermal surfaces oscillate synchronously at different depths. This can take place only if the field is dominated by the first mode.

After the separation of background and solitary waves, we have the samples of the vertical displacements caused by the soliton trains which now can be used to study the effect of intense nonlinear IWs on sound propagation in shallow water. Further details on this topic are given in Chap. 5 of this book. On the other hand, we also have a realization $\zeta(\vec{r}, z, t)$ of the random field. We consider the value $\zeta(x, y, z, t)$ as a random field at the fixed point $\vec{r}, z$ and homogeneous in the horizontal plane so that the power spectrum does not depend on (x, y). We denote it as $G^{\mathrm{t}}(z, \Omega)$; for the background IW, it is shown in Fig. 2.14.

This (averaged and smoothed) spectrum has some important features. First of all, one clearly sees the inertial and M2 internal tidal components. Second, one sees a monotonically decreasing section that falls off as f to the minus 1.75. And finally, there is the high frequency part of the spectrum ($\Delta\Omega = 0.8$–10 cph). This last part of the spectrum can be shown to be due to soliton-induced background waves.

It is important to note that in shallow water, the IW field can be strongly anisotropic. The surface manifestations of the IW field are the main evidence we have of this statement. One example of the IW field surface manifestation is shown in Fig. 2.15 for the MAB. This figure presents a mosaic of sea surface images superimposed on a geographic grid. The images were obtained using side-looking airborne radar. The figure clearly illustrates the complexity of the spatial structure of the IW field in shallow water. One easily sees the system of well-pronounced lines, corresponding to the soliton trains, propagating in the onshore direction.

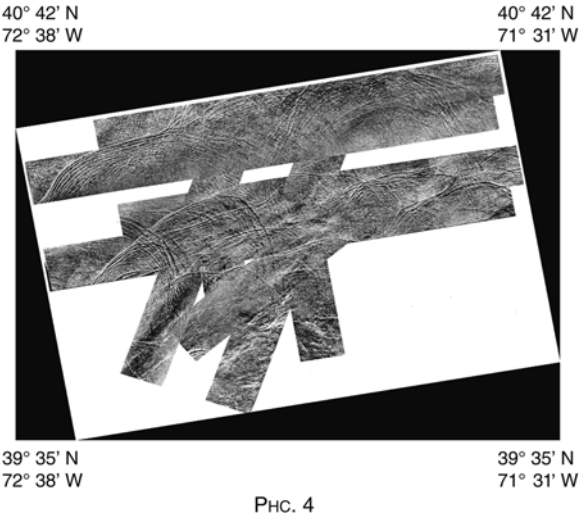


Fig. 2.15 Mosaic of radar images of MAB internal waves

Based on spectral processing of the image, we can see that for this region, the separation between adjacent trains is ~10–20 km and that each train is 1–3 km long. All the trains have an almost plane wave front, with a large radius of curvature of the order 10–20 km. The assumption of a totally plane wave front is usually used for sound propagation simulations in the presence of soliton trains. The spatial spectrum of the image in Fig. 2.15 is presented in Fig. 2.16 (Petnikov et al. 2004). In interpreting this figure, we suppose that the degree of darkness of a small surface element in the photo is proportional to a depth-averaged amplitude of the internal wave (or energy per given area). We see that the maximum of this spectrum is at wavelengths of more than several hundred meters, an expected result. We also see a remarkable anisotropy of the background internal wave field, expected for solitons, but less so for the linear field. This latter result is perhaps not so startling if we note that the soliton train tails consist of linear IW and have roughly the same directional property (including the quasi-plane wave front) as the soliton trains. This clearly demonstrates that the high-frequency background IWs can be induced by soliton trains. Notice that the tidal period internal wave has a well-defined direction and that this wave is the main source of the soliton trains. The monotonic part of the spectrum (see Fig. 2.12) is also formed by the tidal harmonics, and thus we can assume that the IWs in this part of the spectrum propagate in the same direction. Thus, we can state that the background IW of every wave length, except perhaps inertial waves, have the same spatial characteristics and propagate in essentially the direction of the parent soliton train. In describing the anisotropy characteristics of the random background IW field, the directivity function $\Theta(\varphi)$ is usually used. For a single mode assumption (when only the first gravity mode exists), $\Theta(\varphi)$ is equal to

$$\Theta(\varphi) = \int_{-\infty}^{\infty} G_1^s(\tilde{q}, \varphi)\tilde{q}d\tilde{q}, \tag{2.41}$$

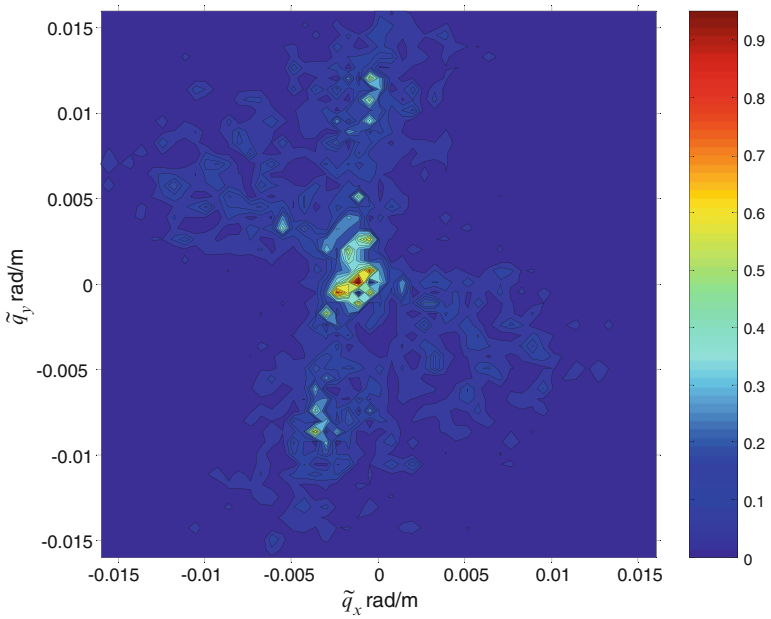


Fig. 2.16 2D (averaged) wavenumber spectrum of IWs

where $G_1^{\mathrm{s}}(\tilde{q}, \varphi)$ is the spatial power spectrum of the background IW field in polar coordinates, and $\tilde{q}_x = \tilde{q}\cos\varphi$, $\tilde{q}_y = \tilde{q}\sin\varphi$. Experimental estimates of $\Theta(\varphi)$ can be made using the radar images of the IW. In doing so, the following assumptions are made:

- The spatial spectrum can be represented as the product of the two functions:

$$G_1^{\mathrm{s}}(\tilde{q}, \varphi) = \tilde{G}(\tilde{q})\Theta(\varphi). \tag{2.42}$$

- Up to a constant factor, $G_1^{\mathrm{s}}(\tilde{q}, \varphi)$ is equal to $G_{\mathrm{si}}^{\mathrm{s}}(\tilde{q}, \varphi)$ for the frequency band $\Delta\Omega$. $G_{\mathrm{si}}^{\mathrm{s}}(\tilde{q}, \varphi)$ is the spatial power spectrum of the surface image of the background IW (without the trains of solitons).

  In this case, $\Theta(\varphi)$ is equal to

$$\Theta(\varphi) = \int_{\tilde{q}_{\min}}^{\tilde{q}_{\max}} G_{\mathrm{si}}^{\mathrm{s}}(\tilde{q}, \varphi)\tilde{q}d\tilde{q}, \tag{2.43}$$

where $\tilde{q}_{\min}$ and $\tilde{q}_{\max}$ are determined by solving (1.98) for frequencies close to the local inertial frequency Ω_i and the buoyancy frequency correspondingly.

Directivity functions for two realizations of the linear background field are plotted in Fig. 2.17 for the IW shown in Fig. 2.15. (The functions are normalized

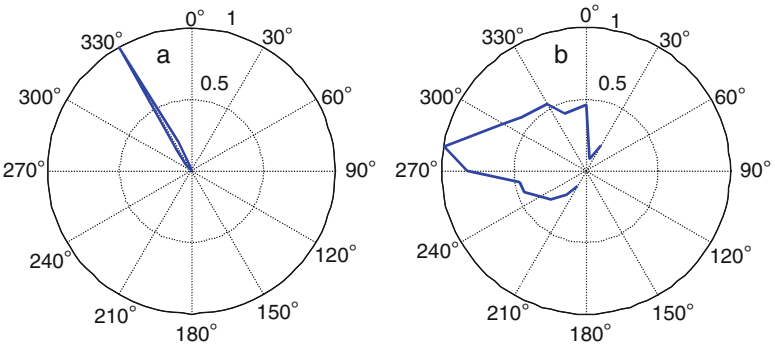


Fig. 2.17 Angular dependence $\Theta(\varphi)$ for the region of the New York Bight in the points 40°23′N 72°30′W (**a**) and 40°13′N 71°47′W (**b**)

here to the maximum value.) For this example, we chose two different areas: one where the IWs have quasi-plane wave fronts (Fig. 2.17a) and one where the wave fronts have a complex form (Fig. 2.17b). As one would expect, the very narrow directivity function corresponds to the IWs with quasi-plane wave fronts, whereas the wave fronts with complex form have a broad beam pattern. Figure 2.17 makes it clear that in shallow water, the background IW field can vary from narrow to broad. This is consistent with our conjecture that the (background) linear internal waves at the tail end of the soliton internal tides leave anisotropic spectra, whereas background waves that are not influenced by the nonlinear field have broader, more isotropic spectra. We note that one also can have areas in the multiple soliton trains going in many directions – in this case, a broad linear spectrum is also expected.

We now turn back to examining the methods of modeling the sound-speed perturbation in the presence of background IW. We can rewrite expressions like (2.32) in a single-mode approximation as

$$\zeta(x,y,z,t) = \int_{\tilde{q}_{\min}}^{\tilde{q}_{\max}} \int_{-\pi}^{\pi} S_1(\tilde{q},\varphi)\Phi_1(\tilde{q},z)\exp\{i(\tilde{q}(x\cos\varphi + y\sin\varphi) - \Omega(\tilde{q})t)\}\tilde{q}d\tilde{q}d\varphi \tag{2.44}$$

where the random field is assured to be statistically homogeneous

$$\langle S_1(\tilde{q},\varphi)S_1(\tilde{q}',\varphi')\rangle = G_1^s(\tilde{q},\varphi)\tilde{q}^{-1}\,\delta(\tilde{q}-\tilde{q}')\,\delta(\varphi-\varphi') \tag{2.45}$$

and $\Omega(\tilde{q})$ is the dispersion relation for the first gravity mode. For estimation of $G_1^s(\tilde{q},\varphi)$, we will assume that the eigenfunction Φ_1 depends only slightly on frequency, or equivalently depends only weakly on the wave number $\tilde{q}$. Note that this is not a strong assumption for shallow water.

Using this assumption, we can write the following simple expression:

$$\left\langle \left|\zeta(\vec{r},z,t)\right|^2 \right\rangle = \Phi_1^2(z)\int_{\tilde{q}_{\min}}^{\tilde{q}_{\max}}\int_{-\pi}^{\pi} G_1^{\mathrm{s}}(\tilde{q},\varphi)\tilde{q}d\varphi d\tilde{q} = 2\int_{\Omega_i}^{N(z)} G^{\mathrm{t}}(z,\Omega)d\Omega, \quad (2.46)$$

where $\left\langle |\zeta|^2 \right\rangle$ is the square of the vertical displacement. Furthermore, by using assumption (2.42) we finally obtain the formula needed for the calculation of $\tilde{G}(\tilde{q})$:

$$\tilde{G}(\tilde{q}) = \frac{2}{\tilde{q}\Phi_1^2(z)} G^{\mathrm{t}}(\Omega(\tilde{q}),z)\frac{d\Omega(\tilde{q})}{d\tilde{q}}\left(\int_{-\pi}^{\pi}\Theta(\varphi)d\varphi\right)^{-1}. \quad (2.47)$$

On the right-hand side of (2.47), one has the physical values which are measured by experiment ($G^{\mathrm{t}}(\Omega,z)$, $\Theta(\varphi)$) or can be calculated via experimental data ($\Phi_1(z)$, $\Omega(\tilde{q})$). Specifically, for the calculation of $\Phi_1(z)$ and $\Omega(\tilde{q})$, we should use (A.98)–(A.100) and the experimental measurement of $N(z)$.

From the above discussion, an algorithm for the computer simulation of a sound-speed perturbation caused by background IWs would be along the following lines:

1. First, generate a 2D, delta-correlated Gaussian random field of vertical displacements $\zeta_\delta(x,y)$ with mean value equal to zero and variance equal to one.
2. Then, calculate the 2D spectrum $S_\delta(\tilde{q}_x,\tilde{q}_y)$ and the corresponding spectrum $S_\delta(\tilde{q},\varphi)$:

$$S_\delta(\tilde{q}_x,\tilde{q}_y) = \frac{1}{(2\pi)^2}\iint_{\tilde{S}}\zeta(x,y)\exp\{-i(\tilde{q}_x x+\tilde{q}_y y)\}dxdy, \quad (2.48)$$

   where $\tilde{S}$ is the area occupied by the internal wave field. We have also used the common knowledge that the random variable $S_\delta(\tilde{q},\varphi)$ has the same statistical characteristics as $\zeta_\delta(x,y)$.
3. Calculate $G_1^{\mathrm{s}}(\tilde{q},\varphi)$ by using (2.47) and the experimental dependences of $G^{\mathrm{t}}(\Omega,z)$ and $\Theta(\varphi)$.
4. Compute $S_1(\tilde{q},\varphi)$ $\left(S_1(\tilde{q},\varphi) = S_\delta(\tilde{q},\varphi)\sqrt{G_1^{\mathrm{s}}(\tilde{q},\varphi)/B_\delta}\right)$, where B_δ is a normalization factor $\left(B_\delta \cong 1/\pi G_{\max}^2\right)$. In this manner, we achieve the relation (2.45).
5. Calculate $\zeta(x,y,z,t)$ using (2.44).
6. Calculate the sound-speed perturbations using $\zeta(x,y,z,t)$.

By way of example, some calculated random realizations of $\zeta(x,y,z,t)$ are shown in Figs. 2.18 and 2.19. Here, we used the spectrum $G^t(\Omega,z)$ shown in Fig. 2.14 and model the dependence $\Theta(\varphi)$ as $\Theta(\varphi) = (\cos\varphi)^n$. Notice that $n = 400$ corresponds to the narrowest directivity function we have seen recorded in an experiment (Kuzkin et al. 2006).

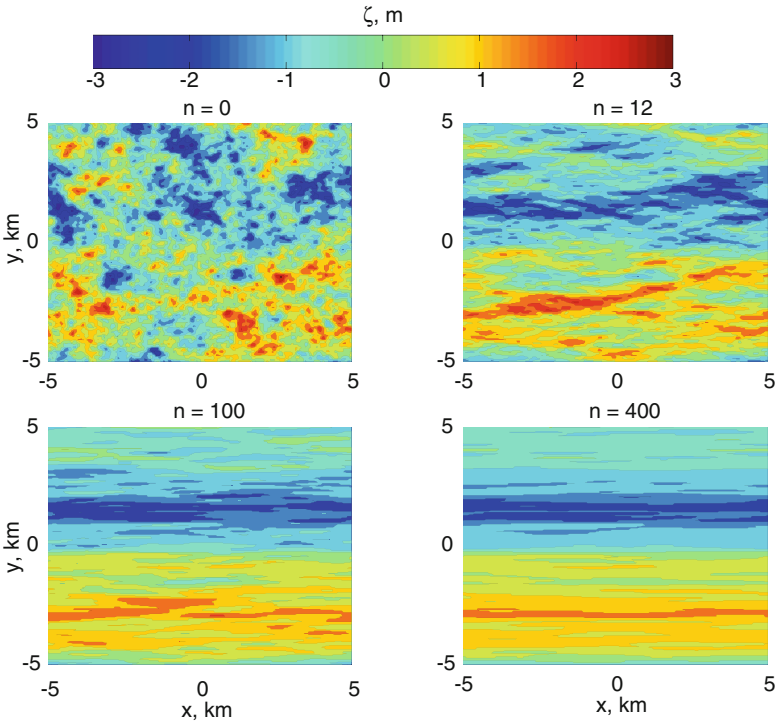


Fig. 2.18 Random vertical displacements at the instant $t = 0$ s, for different values of exponent n of the directivity function (**a**) $n = 0$, (**b**) $n = 12$, (**c**) $n = 100$, (**d**) $n = 400$

2.7 Coastal Geology and Geophysics

The fact that low-frequency sound interacts strongly with the bottom is a defining feature, if not the defining feature of shallow water acoustics. The close proximity (in wavelengths) of the surface and bottom boundaries defines a thin (shallow) waveguide. It is eventually seen that the interaction of the sound with these boundaries is of roughly equal importance to its interaction with the volume (the water column). The effects of the bottom are primarily (1) the reflection of sound, (2) the attenuation of the sound that interacts with it, which dictates the very important propagation loss versus range curve, and (3) the scattering of the sound through rough surface and seabed interior scattering.

It is obvious that in order to understand acoustic propagation and scattering in a shallow water waveguide, one must understand the properties of the bottom. In the following sections, we will discuss these geological properties, especially those most germane to modeling the bottom for acoustics. In that the ocean bottom is both complicated and difficult to measure, the study of the bottom is far from complete at

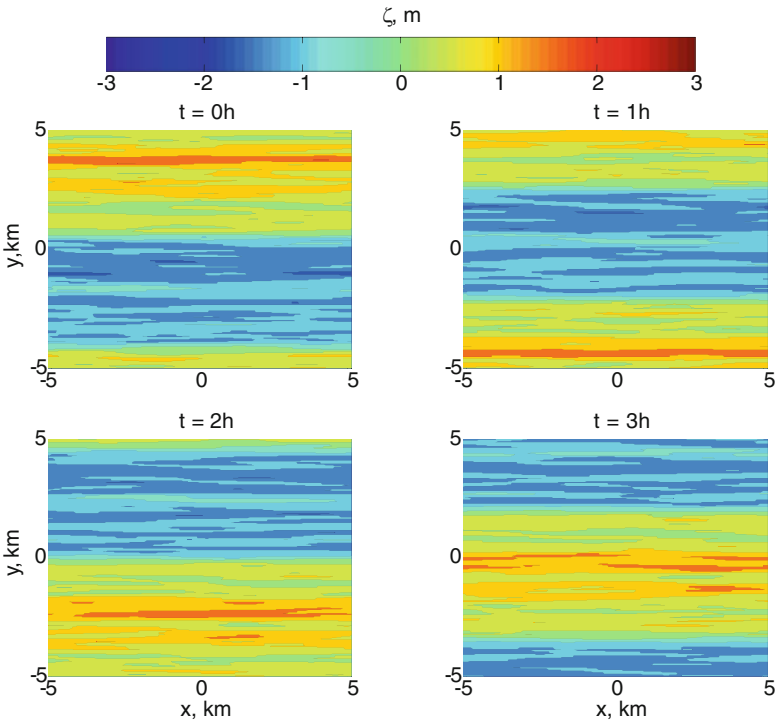


Fig. 2.19 Temporal variation of internal wave field

this point in time, both in terms of geology and acoustics, and so what we will be describing is still a very active area of research (see, e.g., Seibold and Berger (1996) for an overview of the geology).

Shallow water acoustics takes place, by definition, on the continental shelf. The continental shelf is defined as the part of the seafloor that extends from the shoreline out to the shelf break (the latter being the point where there is a large change in bottom slope, leading to abyssal depths). Shelf breaks typically occur at depths of 130–200 m so that the shelf is the relatively flat (with average slopes of less than one degree) area from 0 m to 130–200 m depth.

Shelves can be either broad or narrow, depending on how they formed and what the plate tectonic activity is in their vicinity. A narrow shelf (e.g., the U.S. Pacific coast shelf) can be as narrow as 1 km, whereas a broad shelf (e.g., the U.S. Atlantic coast) can be as wide as 100–200 km, where the distance is measured perpendicular to the shoreline (Table 2.3).

Continental shelves that have little or no seismic activity (“passive margins”) can accumulate sediments over long geological periods and tend to be broad. Shelves which have significant earthquake or faulting activity (“active margins”) that inhibits the accumulation of sediment tend to be narrow.

Table 2.3 Depths and breadths of several oceanic shelves

Place	Average depth (m)	Breadth (km)
Mexican Bight	70	250
Barents Sea	200	1,200
North Sea	180	500
Hudson Bight	100	1,000

Although shelves are considered flat to first order, they often have significant large-scale relief (horizontal scale order of 10 km or more) in areas reflecting (1) water currents that transport sediment in both erosional and depositional modes, (2) the rise and fall of sea level over geological time, (3) glacial scours, and (4) numerous other phenomena. The Hudson Canyon offshore of the Hudson River in New York is a classic example of how geological processes can impose significant bottom relief on an otherwise flat shelf.

The medium scale bathymetry/roughness (horizontal scales of roughly 100 m–10 km) on the shelf will be discussed in detail in a following section, so we will just mention it here.

There is also the "microtopography" (small-scale roughness of order 100 m or less) of the bottom to consider in shallow water acoustics. Though the effect of surface waves in creating bedforms is most significant near the shore, low-frequency surface waves can redistribute bottom sediments at up to 100 m depth, and so the small-scale (bedform) bottom roughness spectrum should be considered a dynamic, changing entity. A useful rule of thumb for determining how small the lateral scale of microtopography one should consider for forward propagation studies is that it be a small fraction (say 0.1) of the acoustic Fresnel zone radius R_F. For a straight line path of distance R connecting a source and a receiver, the Fresnel zone radius is given by $R_F = (\lambda R)^{1/2}$, where λ is the acoustic wavelength (Flatte 1979). For a 100-Hz source transmitting over 20 km (a typical scenario), 0.1 R_F is about 55 m. Scales smaller than this should be looked at as rough surface scattering contributors, whereas larger scales should be treated as bathymetry. This means that a narrow beam echosounder (of order 5° beamwidth) with a bottom patch ensonification width of 10–20 m in 100 m water should be adequate for low-frequency bathymetry surveying, but that as one goes to propagation frequencies near a kilohertz or to closer ranges, the bathymetry map may be undersampled.

Finally, there is the issue of how the surface sediments are distributed on shelves. It is a common observation that the coarsest sediments are deposited nearer to the shore due to the fact that the fall velocity (and thus deposition rate) is larger for large sediments. Thus, the larger terrigenous sediments that are washed into the sea stay near the shore. However, this is a rule with some notable exceptions. First, it is often observed in mid-latitudes that there is a considerable amount of coarse sediment near the shelf break. These are sediments that were transported to the sea from rivers when sea level was lower. (Indeed, sea level has risen some 100–130 m over the past 20,000 years!) They are generally called relict sediments. There is also the case where a river spews forth a large amount of mud and silt,

Table 2.4 A brief précis from Hamilton (1980)

Sediment type	Mean grain diameter (mm)	Density (g/cm^3)	Porosity (%)	Velocity (m/s)	Velocity ratio
Coarse sand	0.5285	2.034	38.6	1,836	1.201
Fine sand	0.1593	1.941	45.6	1,749	1.145
Very fine sand	0.0960	1.856	50.0	1,702	1.115
Silty sand	0.0490	1.772	55.3	1,646	1.078
Sandy silt	0.0308	1.771	54.1	1,652	1.080
Silt	0.0237	1.740	56.3	1,615	1.057
Sand–silt–clay	0.0172	1.596	66.3	1,579	1.033
Clayey silt	0.0077	1.488	71.6	1,549	1.014
Silty clay	0.0027	1.421	75.9	1,520	0.994

Most columns are self-explanatory. The velocity ratio is the velocity in sediment to the velocity in sea water at 23°C, 1 atm, and at the salinity of the sediment pore water

causing a delta or plume offshore of it. The Mississippi, Nile, and Amazon rivers have very famous plumes, with a large amount of fine material close to shore.

The above discussion simply points out that one must seriously consider the geology in trying to understand/predict/measure the sediments in shallow water.

Shelf sediments. In describing the bottom for shallow water acoustic purposes, the main geological entities one needs to ascertain are the bathymetry and the stratigraphy. The bathymetry is, in principle, easily mapped by modern echosounding systems, so we will not belabor it here. The stratigraphy, which includes the layering of the sediments, their types, their age, their mode of deposition, and their geographic extent, is a more challenging study, and indeed is one of the significant open problems in shallow water acoustics. We will first look at the types of sediments found in shallow water in this section, before turning to layering structure and other necessary parts of the stratigraphic record.

Though over a quarter-century old, one of the most useful papers in the ocean acoustician's collection is Edwin Hamilton's "Geoacoustic modeling of the sea floor" (Hamilton 1980). This paper contains much of the basic information one needs to create simple first-order acoustic models of the seabed in a wide range of locations, including our prime area of interest, the continental shelf. Two of the most useful items in this paper describe continental terrace (shelf and slope) sediment properties, which we paraphrase in Table 2.4. The numbers in these tables show what types of sediments are dominant on the shelf (sands, silts, and clays, in various combinations), their sizes (from larger sand grains to fine silty clay particles), three principal geoacoustic properties (the density, sound speed, and the water–sediment interface sound-speed ratio), and two prime poro-elastic medium properties (the mean grain diameter and the porosity). The entries in this table are, to repeat, first-order estimates, but they provide a good starting point for more refined bottom models.

Though the acoustic properties of these sediments are probably already familiar to the reader, we briefly note them anyway. It is seen that there is an increase in

sound speed and density as one goes from the softer, smaller sized sediments (muds and silts) to the larger sediments (sands). Moreover, the velocity ratio increases in the same way as one goes to larger sediments. One of the more interesting entries in this table is the velocity ratio for silty clay, which is less than one, and indicates that the near-surface sediment actually creates a low speed acoustic duct in this case (e.g., see Frisk 1994).

We note, at the risk of triviality, that this table is somewhat incomplete. It omits the calcium carbonate sediments that one gets at low latitudes from coral remnants, the larger pebbles (and even rocks and boulders) one gets from glacial debris at higher latitudes, shell hash that is common on all shelves, and other contributors.

Knowing what sediments are found on the continental shelf in a general sense is not sufficient to give one the precise knowledge that is needed to understand acoustics at a given site. To do this, one needs to know the local sediment types and their variation with depth down to (as a rule of thumb) 2–3 acoustic wavelengths into the sediment. Thus for 100-Hz propagation, one would like to have the properties of the top 30–45 m into the sediment. It is relatively easy to ascertain the properties of the surface sediments via grab samples, cores (box, piston, etc.), accelerometer impact probes, surface reflectivity surveys, photographs, etc. However, sampling deeper sediments becomes problematic. Coring or other such direct probing is the only foolproof way to identify subsurface material, and even this can be difficult. Long cores of near-surface sedimentary material are harder to get on the shelf, particularly in regions with noncohesive sands, where obtainable core lengths may only be a few meters.

One might argue that acoustic probing could be used to get at the type of subsurface material, but this is also difficult, as acoustic profiling primarily senses impedance (the product of density and sound speed), which can be an ambiguous measurement of material type.

Vertical layering and horizontal structure of shelf sediments. In addition to the material type of the sediment, one also needs to know the layering structure of the sediment for acoustics purposes. This component of the problem is what most people commonly think of as "stratigraphy," as it is the direct description of the layers or strata. A typical stratigraphic section of interest to shallow water acoustics is shown in Fig. 2.20. This low-frequency echosounder image shows sloping layering, remnant riverbeds, cut and truncated layers, and other interesting structure that goes well beyond the "simple horizontal stratification" ansatz that is often invoked in modeling the shallow seabed.

2.8 Acoustics of Sediments

In the science of stratigraphy, geologists use the chemistry (especially radiochemistry), the biology (e.g., skeletal fragments), the physics (e.g., magnetics, fluid properties, etc.), and the geometry (the strata and their discontinuities) of rock

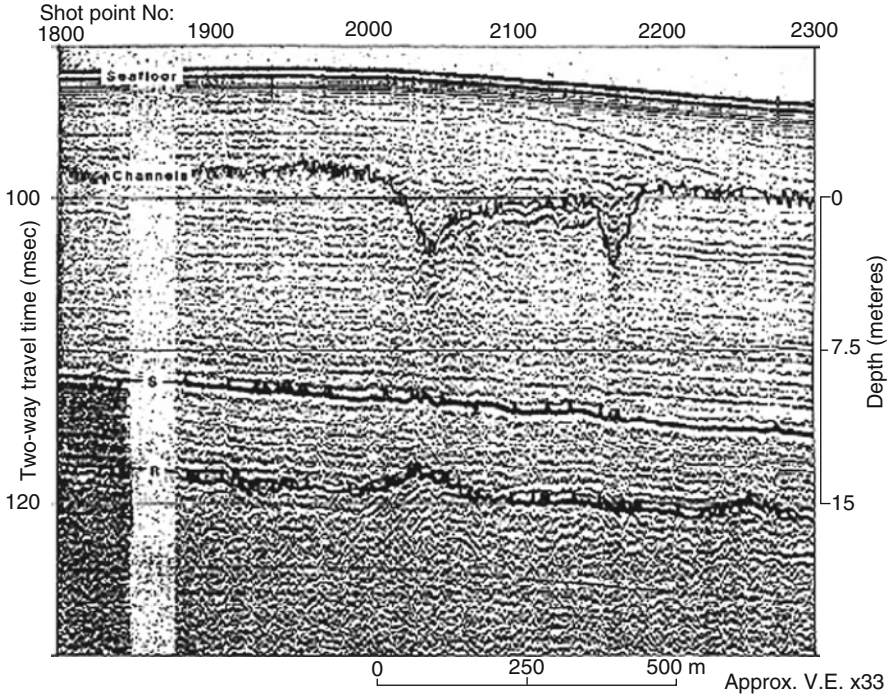


Fig. 2.20 Stratigraphic section from off the coast of New Jersey, showing seafloor, remnant river channels, and the so-called R and S stratigraphic interface reflectors. From Carey, IEEE JOE

and sediment layers as clues in the scientific detective story of the history of the Earth and planets. This is a subject of enormous interest, and indeed the progress over the last century in this area has been staggering.

However, from the pragmatic viewpoint of a shallow water acoustician trying to obtain the acoustic properties of the lower waveguide boundary, the full science of stratigraphy is simply "too much information." What acousticians mainly need is a much simpler thing – the sound-speed, density, and attenuation profiles in the seabed which are (again, to first order) primarily a function of depth. Thus, we will look at this 1D model first and then return to geological stratigraphy (e.g., Fig. 2.20), which will provide the full 3D corrections to the simplified model.

We will look here at two representatives of the acousticians "simplified view" of continental shelf stratigraphy (1) the vertical region model of Hamilton (1980) and (2) the three-region model presented by Katsnelson and Petnikov (2002).

Hamilton's treatment of how to create sediment geoacoustic models is epitomized in his Appendix A, "An example of a geoacoustic model," in which he creates an Abyssal Plain bottom geoacoustic model. While abyssal environments typically have a thick layer of soft sediments compared to shallow water shelf environments, the use of regression equations (based on physical principles, laboratory, and field data) to create vertical sound-speed, density, and attenuation

profiles, and the breaking of the profile into the three regions of (1) sediment, (2) sedimentary rock, and (3) basement are a basic methodology in bottom acoustics. Indeed, Hamilton's work is some of the earliest and most influential in this field, and it is acknowledged that much of the later work in the field builds upon his lead.

As an example of work that builds upon the "Hamiltonian framework" (with due apologies to the physics community), the simple Katsnelson and Petnikov model (see Fig. 2.1) prescribes that coastal sediments have three distinct vertical regions (1) a top layer of unconsolidated sediments, describable by a simple fluid model, of order 10-m thick; (2) a layer of semiconsolidated sediments, extending from 10 to 100 m depth, and also described by a fluid model; and (3) a basement of consolidated sediments, for which solid, elastic medium properties may be important. Sound velocity profiles (and density and attenuation, which we will leave tacit from here) in these regions are based on the material composition of the sediments and are computed from regression equations such as Hamilton's while incorporating any other available historical data, such as cores or acoustic surveys. There may or may not be a discontinuity in sound speed between layers I, II, and III. Layer thicknesses are also somewhat adjustable.

In constructing vertical sound-speed profiles as per the above prescriptions, we note again that we are not using three isovelocity regions, but rather three sediment type regions, each of which has a variable (and not necessarily continuous) vertical sound-speed profile. In describing this with a layered model, we are thus invoking an N-layer model, where N is infinitely large in the limit.

Let us now look at an example of a shallow water geoacoustic model, using some work by Lynch, Frisk, and Rajan off Corpus Christi, Texas (Lynch et al. 1991). Interestingly, one of the motivations of this work was to see how well an a priori geoacoustic model created from high-quality historical data, Hamilton's regressions, etc., could predict newly measured low-frequency acoustic transmission loss data. The a priori model for compressional wave speed and the acoustical inverse model inferred are shown in Fig. 2.21. It is noted that the acoustic inverse result and the a priori data model are different, even though the acoustic inverse model provides an excellent "best least squares fit" to the acoustic transmission loss data, as does the a priori data. This is simply a reflection of the fact that the historical model and the acoustic inverse result have very different vertical resolutions, with the historical data in fact having the better resolution in this case. This is particularly evident at sharp interfaces in the sediment. This simple intercomparison of two "good quality" bottom geoacoustic profile estimates brings out an important point. That is, in creating a "best" bottom model, one needs to pay careful attention to the resolution, variance, and bias of the measurement and model components that make up the estimate. A weighted average that keeps the best pieces of all the component data is what should be used.

One or two more details should be noted on the resolution available, using the Corpus Christi data. First, we see that the historical geological data, which includes core data (with very high vertical resolution, of order centimeters) and high-frequency vertical echosounder data (order $\Delta z = c\tau/2$, where τ is a pulse length on the order of a millisecond, thus giving $\Delta z \sim 1$ meter), give a far better look at the sharp, subsurface

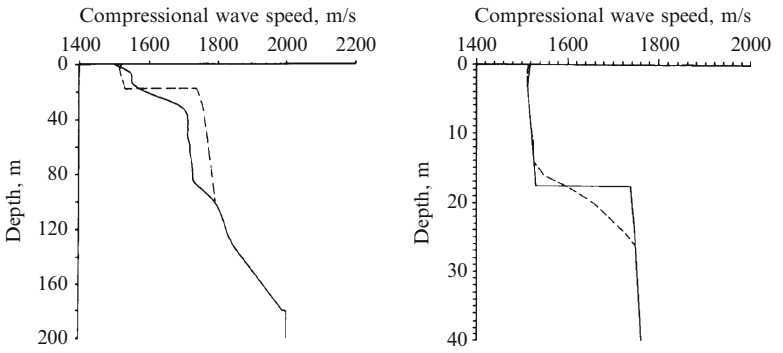


Fig. 2.21 *Left panel* shows an intercomparison of a high-resolution compressional wave speed profile ("Matthews model," *dashed line*) and the result of a low-frequency acoustic inverse (*solid line*). It is seen that the LF inverse shows smoother interfaces versus depth. The *right panel* shows the result of looking at a specific sharp interface ("Matthews model," *solid line*) with only low-frequency acoustic data, i.e., the smoothing of the interface. In this panel, the sharp profile (*dashed line*) is convolved with the vertical resolution kernel of the low-frequency inverse to produce the smoothed result. From Lynch et al., JASA

interfaces that were encountered. This is because the low-frequency acoustic data (a $p(r)$ vs. r data set used for a Hankel transform inverse) have vertical resolution in the bottom given by the differences in turning points (or e-folding lengths) of a sparse set of normal modes, of order ~1 m near the surface and 10 s of meters deeper down. However, near the surface, the LF acoustic data set actually was the better descriptor, in an interesting way that one might not think of, namely, in the temporal dimension. The LF acoustics was able to see a change in the near-surface sound speed due to the change in temperature of the pore water with season, an effect that was not resolved in the historical data (Rajan and Frisk 1992).

The simplified view of bottom modeling described above was sufficient for shallow water acoustics for a number of years, and indeed is still sufficient for many applications today where only modest accuracy is needed. However, both coastal geology and shallow water acoustics have "gone to the next level" of sophistication, namely, a fully 3D medium description, and so we should look at what these recent advances have added to our ability to model the bottom in shallow water. Let us start from the geology viewpoint.

The first thing an ocean acoustician realizes when looking at a 3D geological description of the continental shelf sediments is that he or she is in a whole new world which speaks an entirely different language. Instead of "geoacoustic model" or "sound-speed profile," one sees vocabulary like stratigraphy, morphology, lithology, sequences, facies, strata, tracts, regressive and transgressive sequences, and lowstand and highstand systems. In addition, one encounters the geologic time scale names like Holocene epoch, Quaternary period, and Cenozoic era (all three of which we belong to!). To add to the acoustician's confusion, one finds that, until rather recently, most geological stratigraphy studies were made with low-frequency seismic arrays, which have rather poor resolution of the near-surface sediments

(e.g., the top 100–200 m) which are of most interest to shallow water acousticians. Thus, one can easily get discouraged in trying to translate the geologist's findings into acoustical terms – which would be too bad, as the more recent work in geology has a lot to tell the shallow water acoustician.

Unfortunately, there is no "quick fix" to introducing geology to acousticians; rather, we will make the recommendation (which also holds for physical oceanography and biology, two other marine science areas with strong acoustic links) that the modern ocean acoustician should invest the time needed to get an overview of those areas, at least at the level of an introductory course. Modern ocean acoustics is more and more concerned with detailed effects of the 3D transmission/scattering medium, which come directly from geology, physical oceanography, and biology. Again, we might recommend two basic texts, which the reader can then supplement from his or her own topical interests. They are (1) "Marine Geology" by Kennett (1982) and (2) "The Sea Floor" by Seibold and Berger (1996).

However, even a rudimentary knowledge of geology is still not enough to translate geology to acoustics – one rather critical piece is still missing, namely, what geoacoustic description of a geologically complex medium is adequate/reasonable/usable/etc.? We are now endeavoring to go beyond simple 1D horizontally stratified seabeds (where the sound speed is only a function of z) to a fully 3D seabed. How one describes a 3D geologic/geoacoustic field is still an interesting question, so we see that we have two problems to solve (1) translation of the geologist's map into acoustic variables and (2) a "reasonable" way to express a complicated 3D field of variables for use by acousticians.

By way of example of a modern 3D dataset, let us look at a recent research result from a combined watergun, chirp sonar, and core survey of a section of the East China Sea that was obtained as part of the 2000–2001 Asian Seas International Acoustics Experiment (ASIAEX). Two color plots showing (a) the surface sediment layer thickness for the area and (b) the thickness of the layer just underneath the 0–3-m sandy sediment layer are presented in Figs. 2.22 and 2.23, taken from Miller et al. (2004). The nomenclature "surface layer" from the article is the same as the acoustics usage, so the cross-disciplinary translation is not a problem. The discussion in the article then describes the second layer as "the difference between the transgressive systems tract depth and the sequence boundary," but it is clear that this is a "second layer" in acoustics terms. Thus, the geology-to-acoustics translation is perhaps not as hard as it seems, at least for this case.

The sediment maps in Figs. 2.22 and 2.23 give a top view of the sediment's complexity, just as Fig. 2.20 gives a side view. A spectral/correlation analysis of the ASIAEX data by Abbot, Dyer, Emerson (2006) shows that both layers have roughly isotropic wavenumber spectra, but different correlation lengths, i.e., 2 km for the top layer and 4 km for the bottom layer. Figs. 2.22 and 2.23 make it clear that the structure of shallow water sediments can be complicated.

We will next look at how we describe such complicated structure, i.e., the solution to the "geoacoustic model/description" problem. It is apparent that storing the complete set of 3D geoacoustic parameters at every spatial point is overly computer memory intensive and awkward. Thus, a more compact solution should

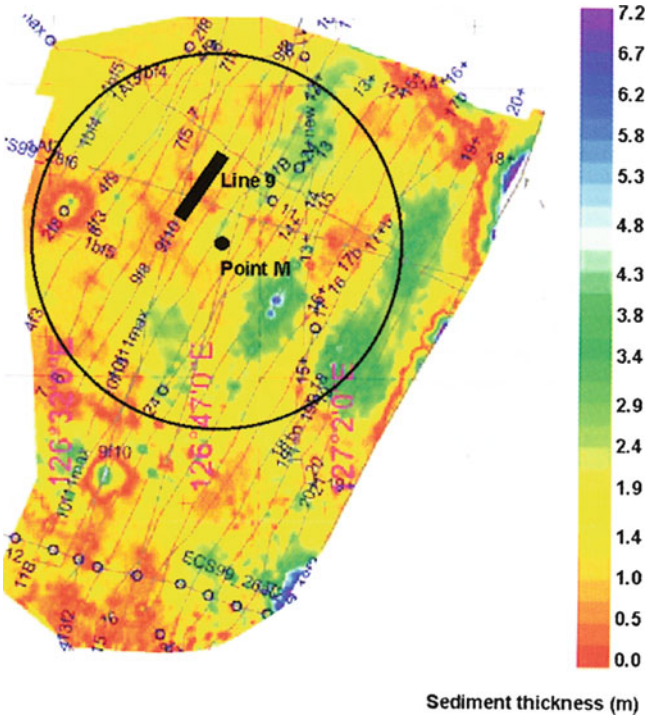


Fig. 2.22 Surface sediment layer thickness (m) at the ASIAEX East China Sea experimental site, inferred from a survey made by L. Bartek. Miller et al. (2004) (with permission from IEEE)

be sought. Three possible solutions come quickly to mind (1) a low-pass filtered layer boundary map, (2) a power spectral (or equivalently autocorrelation function) representation of rough layer boundaries, and (3) an EOF representation. All these have their strengths and weaknesses, which we now discuss.

If one is interested only in a smoothed representation of the bottom layering (range dependence), then it is a simple task to low-pass filtered 2D layer thickness maps, such as those in Figs. 2.22 and 2.23, and store the numbers on a regular grid. This approach is computationally easy but has two weaknesses. First, it (obviously) ignores the high-frequency horizontal variability of the sediments and second, it assumes that the layers maintain their identity (material properties and position relative to each other) over the region mapped. This latter assumption is the one most likely to be violated in shallow water, where the geology is complicated and individual layers can appear or disappear.

The second way of mapping the sediment layering is really an extension of the first, where one superposes some roughness upon the smooth map of #1, using the data to derive an appropriate power spectrum. This is also a computationally tractable way of representing the data. It has the added advantage that one can calculate some of the finer scale scattering effects from the bottom, using the

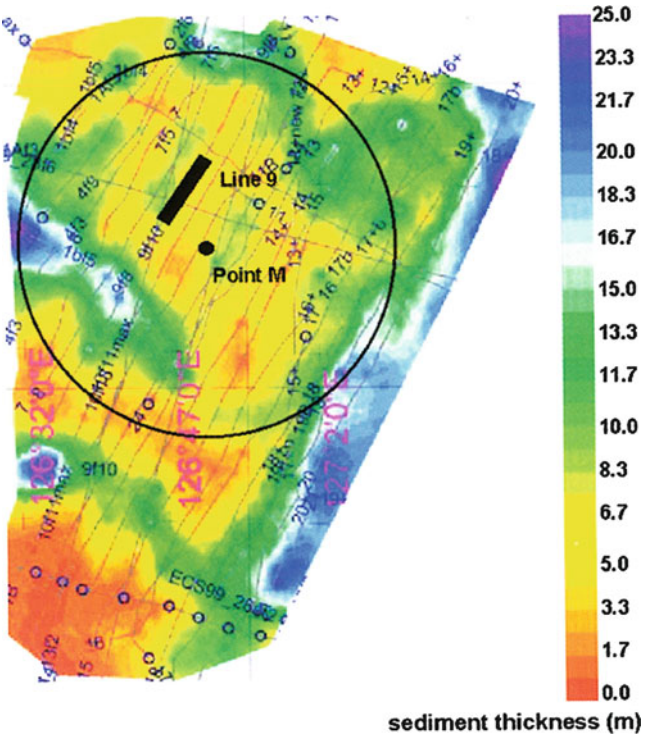


Fig. 2.23 Layer two sediment layer thickness (m) at the ASIAEX East China Sea experimental site, inferred from a survey made by L. Bartek. Miller et al. 2004 (with permission from IEEE)

realizations of either the roughness or the acoustic power equations. It has the disadvantage that the power spectrum loses the phase of the signal, which in this case means it does not reproduce the true realization of the finescale layering structure at a given site. If one does not need the exact finescale regional layering structure, this is adequate. This method also has the weakness of #1 in that it assumes that the layers maintain their identity. We will give an extended discussion of this approach in the next section.

The third method of dealing with bottom layering structure is to use empirical orthogonal functions, which are also discussed in the context of creating useful maps of fully 3D water column oceanography. The advantage of this technique is that it creates a map with the best fidelity to the local region, i.e., it maintains the proper layering structure, including any discontinuities. If one needs the exact local structure, the EOF representation allows one to get precisely the right answer at the data points and also provides the ability to interpolate between them in simple and sensible ways. The major flaw with the EOF representation for the bottom structure is that it is best suited to a red spectrum (like the ocean water column structure). When applied to the bottom, it is seen (J. Miller, private communication) that many modes need to be kept in order to account for most of the variance. This makes the

field representation by EOFs far less economical than for the oceanography; however, the fidelity and the added ability to interpolate the data perhaps compensate for this.

In considering the detailed layering of the bottom sediment, there is also the interesting observation that a number of fine, thin layers can (for some purposes) be replaced by a larger "effective layer" (Avseth et al. 2005). This replacement of fine structure by larger scale "effective structure" is an interesting, and still not fully explored, area in ocean acoustic research.

We should also note that the above discussion does not constrain the properties of the layered medium. One can use a fluid model, solid model, poro-elastic model, etc. Moreover, in all these methods, extrapolation outside the measurement region can only be trusted out to a horizontal and vertical correlation length. Past there, one should revert to the mean-layering profile.

There are other possible representations of the bottom stratigraphy that we have not explored here, e.g., fractal representations. (Fractal roughness will be treated in a later section.) We do not claim to have exhausted the possibilities here; rather, we have just shown a few reasonable ones.

The final topic we will discuss in this section is the implications of this work for both "useful descriptors" of bottom layering and inverse models. The question of "how many local layers (at a given x, y geographical coordinate, since we are now looking at 3D) are adequate," both for the acoustics calculations one wishes to do and to explain the data one takes, is an interesting and important one. The first problem is actually easy enough to address via a sensitivity study. If one is interested in predicting some aspect of the acoustic field, e.g., the magnitude of the pressure field as a function of range for a given acoustic transmission frequency, one can do a sensitivity study that varies the number of layers of the "real bottom" and then judge what number of layers gives adequate resolution for the task at hand. This can be done by averaging layers over some vertical scale, eliminating deeper layers ["hidden depths" (A. Williams 1970)], etc. In the horizontal, one should estimate what the effects of diffraction are to see whether or not to eliminate or consolidate layers.

On the contrary, the question of how many (local) layers are appropriate for an inverse calculation is a different issue. In doing an acoustic inverse (the process of which will be discussed in detail later), one is generally facing an under-determined problem when trying to determine the vertical profile of a given geoacoustic property (e.g., sound speed) at a given geographical point. A common tactic used to address this is to break the local vertical profile into N thin layers, where N is large, and then solve for the property in each layer, using the inverse constraints (e.g., best least squares fit to the data and minimum norm solution) to eliminate the under-determinedness. While this supplies a solution (a number) at every layer depth, these layers are not necessarily resolved by the inverse. Indeed, one needs to look at the vertical resolution kernel of the inverse to find out just how many real (resolved) layers inversion of the data will support. One way to do this is to propagate the layers down from the surface using the resolution kernel the inverse provides. For instance, if near the surface, the resolution length is 1 m, and below

that it is 10 m, and below that it is 10,000 m; we are probably justified to make a three-layer model (the bottom layer being an effective infinite half-space) with interfaces at 0 m, 1 m, and 10 m. What the resolution is versus the depth into the bottom will depend on the dataset used. For instance, given a combination of low-frequency acoustic data and 3.5-kHz vertical echosounder data, the resolution may be better modeled with a different set of layers than for just the low-frequency data alone. Our main points in this last discussion are (1) the layering used in creating a model should be dictated by the vertical resolution of the data available and (2) creating such layered models is still somewhat of an art form, and the answer forces should be dictated by the user's needs.

2.9 Bottom Roughness

In considering bottom properties in shallow water, bottom roughness is certainly another important issue. In this section, we take the phrase bottom roughness to operationally mean the random perturbation of ocean depth $h(r)$ with typical horizontal and vertical scales 10–1000 m and 0.1–30 m, respectively. Other operational scales are also usable, and we should quote the old ocean acoustics joke "one man's roughness is another man's bathymetry." As previously noted, the acoustic frequency used and Fresnel radius/bottom footprint are the practical determinants of this boundary. Mathematically, we assume that depth can be represented as a sum of two terms $H(r) + h(r)$, where $H(r)$ is the macroscale bottom contour, which can be considered as a deterministic process in comparison with $h(r)$.

It should be noted that in shallow water, there exists bottom roughness with smaller horizontal and vertical scale, which is the result of the transport of sediments (Trowbridge and Nowell 1994). However, these microscale roughness elements do not have much effect on low-frequency sound propagation on continental shelves, and so will be ignored for the remainder of this discussion. (At medium frequencies, these scales can be important, however.)

We now look at the bottom roughness in a spectral/correlation function approach. As a first approximation, $h(r)$ can be considered as a stationary, anisotropic, zero mean, Gaussian random field completely specified by its two-point correlation function (Goff and Jordan 1988)

$$B_h(\vec{r}) = \sigma_h^2 \Phi(\hat{r}(\vec{r}))/\Phi(0), \tag{2.49}$$

where σ_h is root mean square (RMS) depth variation, $\Phi(\hat{r}) = \hat{r}^{\nu} K_{\nu}(\hat{r})$, K_{ν} is a modified Bessel function of the second kind and order ν, and $\hat{r} = \sqrt{\hat{q}_{11}x^2 + 2\hat{q}_{12}xy + \hat{q}_{22}y^2}$, where $\hat{q}_{ij}$ are the Cartesian elements of a positive-definite, symmetric matrix $\mathbf{Q}$ and have dimensions of $(\text{length})^{-2}$. $\mathbf{Q}$ is expressed in terms of its ordered eigenvalues $\hat{k}_n^2 \geq \hat{k}_s^2$ and its normalized eigenvectors $\vec{e}_n$ and $\vec{e}_s$ which are orthogonal to each other, i.e.,

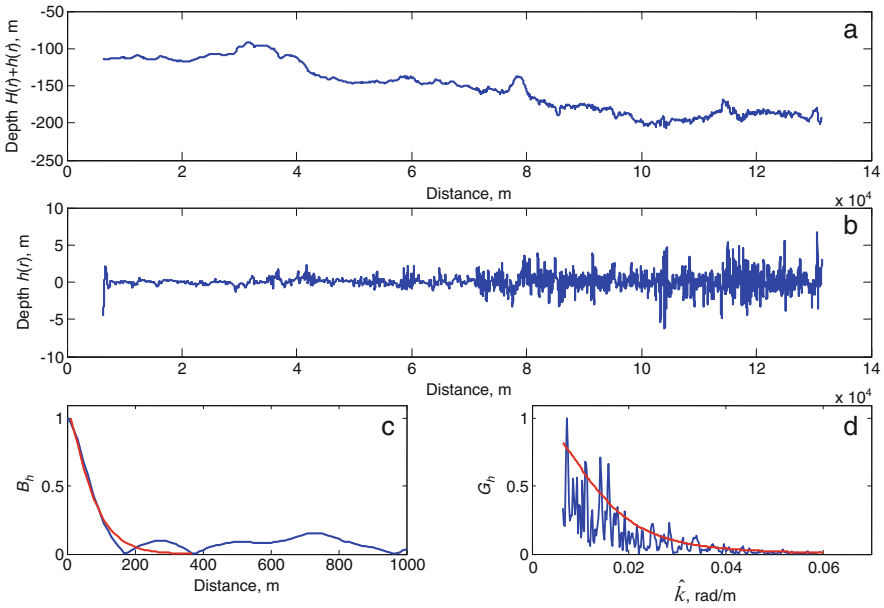


Fig. 2.24 Bottom roughness recorded in Barents Sea. (**a**) $H(r)$+$h(r)$, (**b**) $h(r)$, (**c**) and (**d**) correlation function of random field $h(r)$ and its spatial power spectrum (*blue lines*), respectively. Correlation function (**c**) and power spectrum (**d**) calculated in accordance with (3.1) and (3.4) are shown by *red lines*. All correlation functions and power spectra are normalized to the maximum values

$$\mathbf{Q} = \hat{k}_n^2 \vec{e}_n \vec{e}_n^{\mathrm{T}} + \hat{k}_s^2 \vec{e}_s \vec{e}_s^{\mathrm{T}}. \tag{2.50}$$

Notice that $\hat{k}_s$ characterizes the correlation radius of the random field $h(r)$ in the horizontal direction, determined by the azimuth φ_s. This direction corresponds to the smaller characteristic wavenumber $\hat{k}_s$. The characteristic values define an aspect ratio

$$\widehat{a} = \widehat{k}_n / \widehat{k}_s, \tag{2.51}$$

which is unity for the case of an isotropic random field.

In accordance with (A.28) of our reference, the spatial power spectrum of bottom roughness is equal to

$$G_h(\hat{\vec{k}}) = (2\pi)^{-1} \nu \sigma_h^2 |\mathbf{Q}|^{-1/2} \left[u^2(\hat{\vec{k}}) + 1 \right]^{-(\nu+1)}, \tag{2.52}$$

where

$$u(\hat{\vec{k}}) = \left[\hat{\vec{k}}^{\mathrm{T}}\mathbf{Q}\hat{\vec{k}}\right]^{1/2} = \sqrt{\left(\hat{k}/\hat{k}_s\right)^2\cos^2(\varphi-\varphi_s) + \left(\hat{k}/\hat{k}_n\right)^2\sin^2(\varphi-\varphi_s)}. \quad (2.53)$$

In practice, the parameters found in (2.49) through (2.53) are characterized by strong geographic variability. As an example, in Fig. 2.24a, b we show the bottom roughness recorded in one of the regions of the Barents Sea. The RMS value was equal to $\sigma_h \approx 1\,\mathrm{m}$. The corresponding correlation function and spatial power spectrum are also plotted in Fig. 2.24c, d, as the functions $B_h(r)$ and $G_h(\widehat{k})$ calculated by using (2.49) and (2.52) with $\hat{k}_s = 0.02$. Since we did not have data concerning the anisotropy of the roughness field (which in general one does not), we assume in this calculation that the $h(r)$ shown in Fig. 2.24 is isotropic, i.e., $\hat{r} = r$ and $a = 1$. We also assume that[2] $\nu = 1$. As can be seen from Fig. 2.24, the model matches the experimental spectrum and correlation function more or less reasonably. We note that the correlation distance of bottom roughness shown in Fig. 2.24c is equal to $L_r \cong 1/\hat{k}_s = 50\,\mathrm{m}$.

The data in Fig. 2.24 also display some obvious spatial nonstationarity (e.g., the high wavenumber roughness increases with the range), demonstrating some of the pitfalls in imposing "simple" processing on a complicated ocean bottom.

2.10 Solid and Multicomponent Layered Bottom Models

In the previous sections, we considered the bottom as a multilayered fluid medium. A more realistic and, naturally, more complex model for the bottom is the so-called multilayered elastic solid bottom model (a single-layer elastic solid bottom model being a first approximation). In this model, two types of waves dominate (1) the compressional wave, which is the usual longitudinal acoustic wave that we have studied in fluids, and (2) the shear wave, which is a transverse wave that can exist only in a solid medium. The linear elastic medium is, in the most basic form, described by Hooke's law, which is a tensor equation relating the stress tensor (the imposed forces on the solid) to the strain tensor (the resulting deformation of the solid). For an isotropic, linear elastic medium, only two constants are needed to specify the stress–strain relation completely. The two most usual are the Lame coefficients λ, μ (Mavko et al. 2003; Brekhovskikh 1980). Other possibilities are the bulk modulus, Young's modulus, and the P-wave modulus. We refer the reader interested in the various forms of Hooke's law to the above references, which are two among many.

[2] Notice that parameter ν connected with Hausdorff (fractal) dimension D of the stochastic process $h(r)$ is $D = 3 - \nu$. For the sea bottom, D generally lies in the range from 2 to 2.5 (Goff and Jordan, 1988).

Table 2.5 Geoacoustic properties of elastic solid bottom material typical of shallow water

Sediment	Density (g/cm^3)	Speed of longitudinal waves c_l (m/s)	Attenuation coefficient of longitudinal waves α_l (dB/m kHz)	Speed of shear waves c_s (m/s)	Attenuation coefficient of shear waves α_s (dB/m kHz)
Aleurite–clay, mud	1.9	1,520–1,600	4–10	300–600	
Chalk	2–2.2	2,000–3,000	0.8–1	400–1,500	0.1–4
Basalt	2.3–2.5	3,000–5,000	0.02–0.08	1,600–2,500	0.02–0.1
Coarse sand	2.0	1,800	0.47	250	

Within the framework of the geoacoustic model, we characterize an elastic bottom by five geoacoustic parameters: density, compressional (longitudinal) wave speed and attenuation, and shear (transverse) wave speed and attenuation. The speeds of the compressional and shear waves can be expressed through the Lame parameters as

$$c_l = \sqrt{\frac{\lambda + 2\mu}{\rho}}, \; c_s = \sqrt{\frac{\mu}{\rho}}. \tag{2.54}$$

In order to get a feel for the solid bottom parameters of sediments in shallow seas, we present Table 2.5.

At this point, one has to decide whether to "take the plunge" into the depths of elastic media theory or to expend one's efforts elsewhere. For the purposes of shallow water acoustics, we would generally advise the latter course, not out of laziness, but because of the numbers presented in Table 2.5. Specifically, one notes that in the sands, clays, and silts that dominate the bottom material in shallow water, the shear wave speed is rather low. Only in chalk (e.g., carbonate coral reef material) or basalt (rock bottoms, as opposed to sediments) does one see high shear speeds. We will show below that only for high shear speeds (i.e., less usual bottom material) does one need to include shear effects in a detailed manner.

Our argument is based on Fig. 2.25, which is taken from the text by Brekhovskikh and Lysanov (1991). In this figure, one sees that for shear speeds typical of sands, clays, and muds (like curve 3), the effects of shear on the magnitude of the plane wave reflection coefficient are comparatively small. This means that we might be able to account for shear effects by some sort of perturbation to the compressional wave problem, and indeed we consider just that in a later section of this book. Specifically, shear provides an effective attenuation mechanism for the compressional wave, which converts partially to shear wave energy, which is then lost quickly in the sediment.

Next, we will note that there are also interface waves (Scholte waves) that are created by the existence of shear in the bottom sediments. These can be useful in obtaining bottom shear properties, and again we refer the reader to Brekhovskikh and Lysanov (1991) for further details.

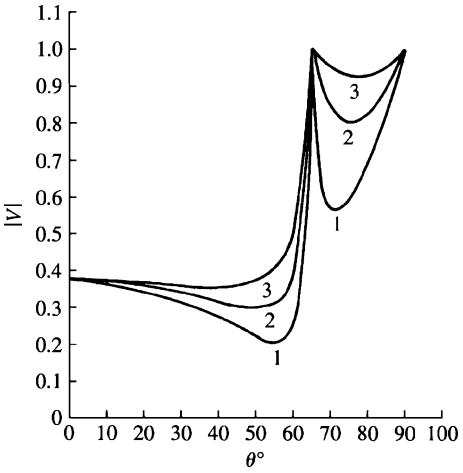


Fig. 2.25 Magnitude of the reflection coefficient for three different bottom shear speeds. For curve 1, $c_s = 750$ m/s, for curve 2, $c_s = 600$ m/s and for curve 3, $c_s = 450$ m/s. Bottom density is twice water and sediment compressional wave speed is 1,666 m/s. From Brekhovskikh and Lysanov. θ° is angle of incidence

In closing this section, we will try to make some amends to the reader for not treating the cases where the bottom material has a high shear speed, e.g., carbonate and basaltic bottom. Many good references on the topic exist, and we would refer the reader to the texts by Brekhovskikh, Jensen et al., and Mavko et al. as good starting points. The theory of linear elastic media is well developed, understood, and implemented (via computer models) at this point in time so that the reader can expect a reasonably good answer to any questions about these effects.

Multiphase bottom models. It is obvious to anyone who has ever walked on a sandy beach washed by waves (the swash zone) or who has trod on the bottom of a muddy bay or silty river that the bottom sediments in the ocean are not pure solid or pure liquid, but rather a two-phase mixture. The question then arises how does this mixture behave compared to the pure liquid or pure solid cases that we have considered so far? Rather luckily for ocean acoustics, the study of poro-elastic media has been pursued for many years by the soil mechanics and groundwater hydrology communities, who are interested in buildings and structures being placed on such media, reservoir behavior, and other such practical engineering problems. Thus, a body of literature applicable to the ocean bottom existed long before ocean acousticians needed to refine their fluid and solid models of the bottom.

One of the most well-known poro-elastic medium theories is the so-called Biot theory, which is also often called the Biot–Stoll theory, acknowledging the significant extensions of the theory by Stoll (1989). We will discuss some of the background and the modern uses of this theory in shallow water in the next section, but first we would like to show just *why* we need to go to this next level of sophistication in modeling the seabed. We will first look at theory predictions and then at experiments to provide this justification.

Our theoretical bolster for looking at poro-elastic media comes from the work of Stoll (1989). In Figs. 2.26 and 2.27, we show (respectively) the compressional wave

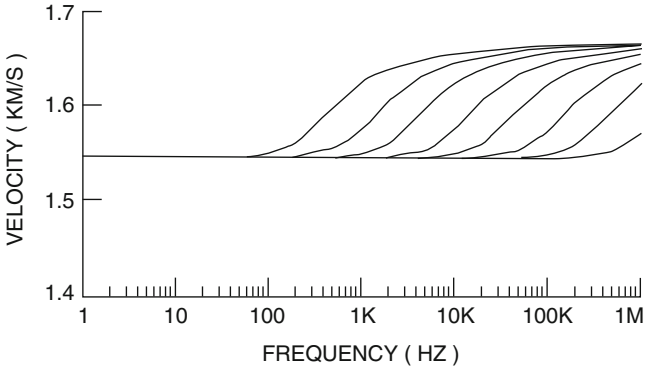


Fig. 2.26 Biot–Stoll model velocity dispersion curve

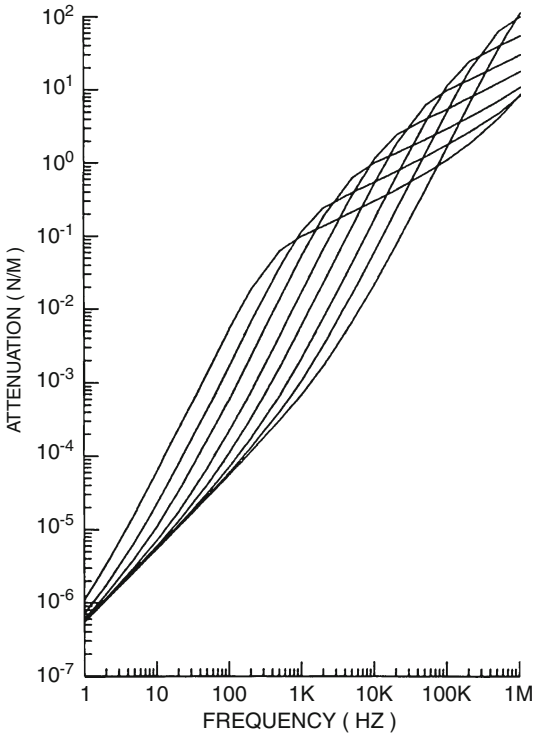


Fig. 2.27 Biot–Stoll model attenuation dispersion curve

velocity and compressional wave attenuation versus frequency. In Fig. 2.26, we see that the Biot–Stoll theory predicts *dispersion* (frequency dependence) of the medium sound speed, an effect that does not occur in either the pure liquid or solid cases. In Fig. 2.27, we see variability in the power law for attenuation versus frequency, something which does not happen for the simple liquids or solids, which

predict only linear frequency dependence. It is thus obvious that poro-elastic theory is predicting effects that are not seen in the simpler theories, and so we should now try to see if such effects are observable by experiment. (Before leaving the curves in Figs. 2.26 and 2.27, we should note that the different curves in each feature represent given variations in the parameters $\widehat{\alpha}$ (the pore size parameter) and $\widehat{k}$ (the coefficient of permeability) in such a way that $\widehat{k}/\widehat{\alpha}^2 = \text{constant}$.)

We now look for experimental evidence of poro-elastic medium effects. In looking for these, we can examine either compressional wave effects or shear wave effects. As shear waves are hard to measure in sediments, we prefer to look at compressional waves. We have the choice of the real part (sound speed) or the imaginary part (attenuation) to consider, or both. Present theoretical predictions of poro-elastic soundspeed dispersion effects show that the signal sizes should be only a few tens of meters per second in real marine sediments. While measurable, this is small and often masked by experimental error, so we should thus turn to compressional wave attenuation. Luckily, in this case, the signals are *not* so small, as we shall see.

As mentioned, the simple fluid and elastic models predict a linear attenuation dependence with frequency, whereas poro-elastic models predict different dependencies. It is possible to identify at least three mechanisms for the loss of sound intensity in a homogeneous poro-elastic bottom:

- Inelastic losses in the sediment skeleton
- Viscous losses, associated with the relative movement of the interstitial fluid and the skeleton
- Losses due to volume scattering

These mechanisms give different dependencies of the absorption coefficient on frequency: from a proportionality to the square root of the frequency for the model of viscous losses to a proportionality to the fourth power of the frequency, associated with volume scattering, when the size of the scatterers is much less than the wavelength of the sound wave. It is obvious that the measured frequency dependence will represent some weighted combination of these mechanisms. Thus, site-specific studies are needed to explain the frequency dependence of the sound absorption in the bottom for a given shallow water region and for a given frequency band.

As a result of considering complex poro-elastic bottom models, various authors have proposed an exponential frequency dependence for the coefficient

$$\alpha(f) = \alpha_0 (f/f_0)^q. \tag{2.55}$$

Here, α is the determined sound absorption in the sea bottom. It is related to the bottom refractive index via $n_1^b = c(H)/[c_1(1 + i\alpha/2)]$, where c_1 is sound speed in the bottom, α_0 and f_0 are normalization coefficients, and q is the most important parameter, the exponential determining the frequency dependence. All these parameters depend on real, experimental conditions (i.e., the type of bottom). The

main parameter q has been the subject of significant interest for many authors who have measured it in diverse areas, mainly via the best fit between experimental and theoretical acoustical data. In Table 2.6 (courtesy of J. Holmes [J. Holmes et al. (2005)]), a compilation of the results from the different authors referenced above is presented. Intriguingly, the q parameter tends to cluster in a range of values between 1.5 and 2.0, the latter value being what the simple Biot theory would predict. It is thus obvious from the data that poro-elastic medium effects are important, and moreover that their exact explanation needs more than even the simpler forms of the Biot theory. We will examine this topic further in a later section. We will also look further at the measurements of q in a later chapter dedicated to inverse theory.

Two variants of the Biot theory that have attracted considerable recent attention in the underwater acoustics community are those by Buckingham (1997) and Pierce et al. (2004; Holmes et al. 2005). The former theory has been most extensively applied to higher frequency bottom dispersion data (with great success), whereas latter is tailored more toward low-frequency data.

2.11 Acoustics of Biological Objects in a Coastal Area

To a shallow water acoustician working at frequencies between 50 and 5,000 Hz, marine biology means pretty much two things: fish and marine mammals. We will disregard smaller flora or fauna as "weak effects" for this frequency range, and concentrate solely on these larger animals. Again, some basic references to start from might be (1) "Biological Oceanography" (Miller 2004) and (2) "Marine Mammals and Noise" (Richardson et al. 1995).

As to why we would consider fish and marine mammals, there are a variety of reasons. Starting with fish, it is well known that they can both significantly attenuate sound in forward propagation and backscatter sound, leading to significant volume reverberation. The former can lead to ambiguity and error in measurements of propagation loss and fluctuations (i.e., is it the ocean, the bottom, or the fish causing the signal?), whereas the latter can be a significant source of clutter in sonars. For many years, the effects of fish were ignored, both because they were intermittent/transient signals which were deemed *probably* not important in most transmission and reverberation experiments and sonar usages. (The emphasis is on the probably, because if a large fish school was present, the measurements could exhibit a substantial (unknown) error.) Also, the measurement of fish schools and classification of their members were just not advanced enough for the purposes of shallow water acoustic experiments, where one needs to know the fish distribution along an acoustic propagation path (be it one-way forward or two-way backscatter) during each transmission. The ship-borne surveys standardly used by fisheries and marine biologists (even with acoustic backscatter instruments) were just too slow, of low resolution, and space–time aliased to be of real value to shallow water acoustic work. However, this situation is probably changing, as we will discuss.

Table 2.6 Compilation of frequency-dependent attenuation results (courtesy of Holmes et al., unpublished manuscript 2005)

Reference	Frequency range (Hz)	Bottom type	Estimated critical angle (degrees)	Frequency dependent exponent (n) following af^n	Experimental technique[a]
Ingenito (1973)	400–750	Sand	19	1.75	M
Beebe et al. (1982)	100–600	Medium to coarse sand	29	1.76	TL
Beebe et al. (1982)	25–250	Coarse sand with gravel	35	1.57	TL
Zhou (1985)	80–800	Sand–silt–clay	19	1.84	M
Zhou et al. (1987)	100–1,000	Fine sand and silt	21	1.6	INV
Tatersall and Chizhik (1993)	100–8,000	Medium sand	24	2.0[b]	TL
Tappert and Yamamoto (1993)	50–800	Sand–silt–clay	19	2.0	TL
Carey and Evans (1998)	500–1,000	Sand–silt–clay	18	1.5	TL
Rozenfeld et al. (2001)	47–604	Sand–silt	23	1.8	TL
Peng et al. (2004)	100–500	Very fine sand	21	1.65	TL
Zhou et al. (2004a)	100–700	Very fine sand	22	1.63	INV
Knobles et al. (2004)	25–800	Fine sand	21	2.0	TL

[a]*M* Modal, *TL* Transmission Loss, *INV* Inversion

[b]Paper actually reports agreement with Biot theory, which gives $n = 2.0$ in the 100–1,000 Hz range

Another reason for considering fish in shallow water acoustics is to use shallow water acoustic techniques to *measure* the fish schools. The absorption and backscattering of sound by the fish can be used as signals in tomographic and beamforming imaging schemes, respectively (see Diachok et al. 2004; Makris et al. 2006). We will discuss these schemes in more detail shortly.

Turning to marine mammals, the population density of these animals is not so high that they will ever significantly attenuate or backscatter acoustic signals in our frequency range. However, the use of acoustic techniques to track these animals, for both population and behavior studies, is very valuable scientifically. Another reason that marine mammals are important for shallow water acoustics is because they are protected (under the US law) under the Marine Mammal Protection Act and, in cases of certain species, the Endangered Species Act. In order to perform shallow water acoustics with active sources, US researchers need to know where the animals are likely to be, and how the sounds being employed would affect them. Exposure levels need to be estimated, and based on this estimate, permits may or may not be required and issued. It is worth noting that estimating exposures and their effects is still a science/technology under development, as will be discussed further. Finally, the vocalizations of these animals can be considered a noise source for those using sonars for purposes other than marine mammal research.

Scattering and absorption of sound by fish. The study of the scattering and absorption of sound by fish is, at this point in time, a rather well-developed field. While there are certainly a number of open questions yet to be answered, there is also enough knowledge and technology available to enable a researcher to calculate what the scattering cross-section of sound is for a given frequency and species of fish (Clay and Medwin 1977). From a basic physics viewpoint, there are three salient regimes of scattering from a fish of size L (1) low-frequency, $L/\lambda \ll 1$, called the Rayleigh or "point scatterer" regime, (2) mid-frequency, $L/\lambda \approx 1$, called the resonant or "organ pipe" regime, and (3) high-frequency, $L/\lambda \gg 1$, called the geometric optics regime. The most important regime, for both scattering and absorption, is the resonant regime. At this frequency, the effect of the fish is greatest, often by several orders of magnitude over the other regimes. The resonance, it is well known, is due to the gas-filled swim bladder that most fishes have. This bladder accounts for ~90% of the sound scattering from the fish as a whole. The theory of this resonant scattering has evolved through the last few decades from that of a simple spherical gas bubble in water, roughly the size of the swim bladder, to that of an extended, irregular gas-bearing bladder encased in a fleshy housing that can be treated as an elastic medium with the correct shape and acoustic properties. Needless to say, calculations of the scattering cross-sections of fish with these more complicated geometries and material properties necessitate sophisticated scattering physics and computational codes – however, this is possible today, if one wants exact numbers. It can be argued (Diachok et al. 2004) that the "free bubble" approximation may be quite adequate for certain calculations (e.g., the resonant frequency); however, this probably should be examined on a case-by-case, species-by-species basis.

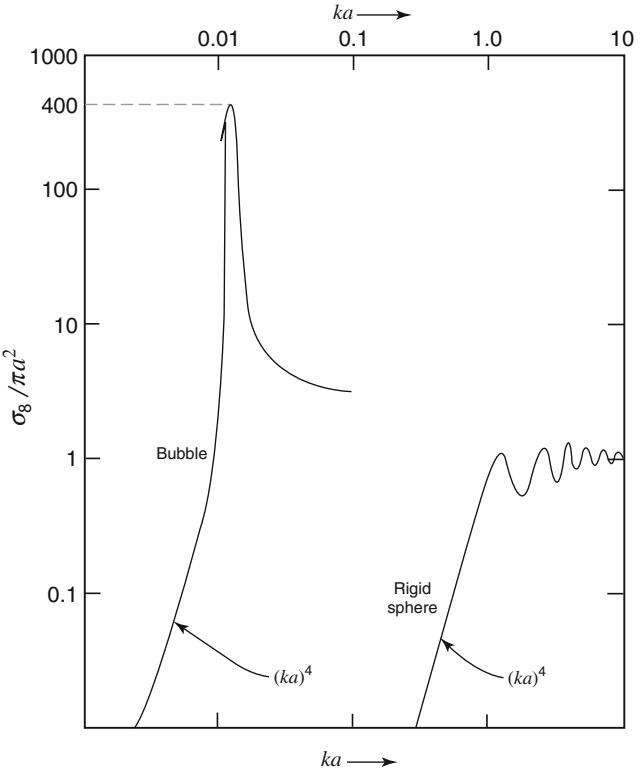


Fig. 2.28 Example of bubble and rigid sphere scattering cross-sections

It would be helpful to glance at what the cross-sections and attenuations are for fish, just to get an order-of-magnitude feel for them. In Fig. 2.28, we show (from Clay and Medwin 1977) a plot of the total scattering cross-section of a bubble compared to that of a rigid sphere. We use the dimensionless quantity *ka* instead of frequency in the ordinate, and the ratio of the total cross-section to the geometrical cross-section πa^2 in the abscissa, conventions widely used in scattering theory. One immediately sees that the sphere has a small total cross-section that peaks at (approximately) the geometric value, and a smooth transition from the Rayleigh scattering regime to the geometric regime. The bubble, on the other hand, has a whopping peak at the $ka = 1$ resonance, some 400 times bigger than the geometric value. It is obvious that resonance scattering is the main story for fish. We also note that the bubble scattering is approximately isotropic in direction so that the relations between the backscattering, forward scattering, and total cross-sections are simple. Thus, our lack of previous discussion of the directionality of the scattering.

The next part of the story, the attenuation, also shows sharp resonant effects, as seen in Fig. 2.29 from Diachok's work. In this figure, which does use real fish sizes and acoustic frequencies (not *ka* as before), we see that for a distribution of sardines

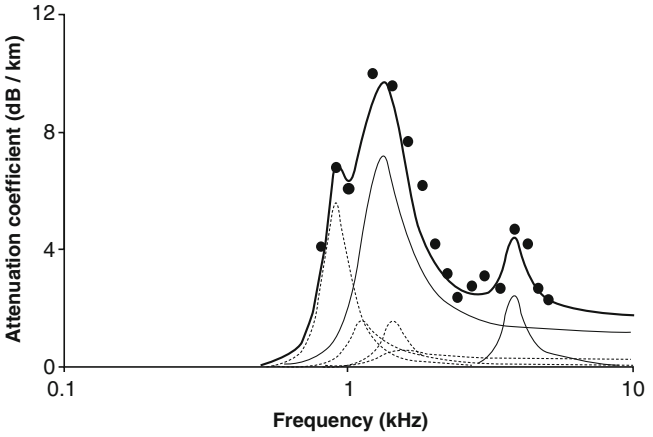


Fig. 2.29 Attenuation by fish vs. frequency, showing resonances (from Diachok)

(in age and size) ensonified at night, there is a distinct absorption peak at ~1.2 kHz, a wavelength that resonates with the dominant size fish in the distribution. The attenuations are seen to be substantial in the frequency band of interest to this book, and so if fish schools such as those measured by Diachok are present, biology needs to be part of propagation and scattering experiments, and the systems using them.

Population distributions of fish and surveying fish populations acoustically. The scattering and absorption cross-sections for fish are one part of the acoustic problem one needs to address. The distribution (in x, y, z, and t) of the fish and their species classification are another. In acoustic propagation studies, we need these quantities to factor out the effects of fish. Biologists and fisheries researchers are just as interested in these distributions and classifications for biological and economic reasons. To date, the standard method of surveying fish populations has been ship transects using acoustic echosounders ("fish finders") and tow nets to classify the species and give ground truth for the acoustic profiles. This method works to an extent, but it has low resolution and is severely space–time aliased. One might see the day–night migration of fish populations from the water column to the bottom, or get a rough idea of the dimensions of a fish school across the transect direction with these methods, but they still do not come close to a "snapshot" or series of snapshots of what is going on. Diachok has proposed using the resonant absorption of the fish to do an ocean CAT scan of fish populations, which he calls "resonance absorption tomography." However, the sorting out of fish attenuation from other ocean effects and the large number of slices needed to do tomography with good resolution make us think that this is an idea that is theoretically possible, but perhaps impractical in implementation. Recently, Makris et al. (2006) *were* able to produce a snapshot of fish shoals in the Atlantic using a pulsed, low-frequency, high power source array coupled with a very capable towed array receiver. The pulses echoed back strongly from

the fish schools, providing a "snapshot" with each pulse. The travel time of the pulse gave the range, and the steered beams the azimuth, producing beautiful 2D (x–y) time series views of the fish shoals over tens of kilometers with good resolution, and giving many new insights into their dimensions, structure, variability, biomass, etc. This combination of a low(er)-frequency pulsed source and a towed, steered array looking at backscatter seems to be the way to go in the future if one wishes to map fish populations effectively over large areas. We would make one suggestion as a possible future improvement of this method. As it stands, Makris' method requires large equipment and 2–3 ships to work (one for the source array, one for the towed receiver array, and one to net survey the area for ground truth as to species). Perhaps, by using several small AUVs equipped with mid-frequency (resonant) sources and directional receivers for this work, one could perform a survey equivalent to that using larger (and admittedly more range-capable) equipment. AUVs have already used high-frequency side-scan sonar and cameras to survey and classify fish populations at close ranges; so this is merely an extension and combination of the previous work by Makris and the AUV community.

Shallow water acoustics and marine mammals. Marine mammals enter into shallow water acoustics in (at least) five ways. First, one is interested in localizing marine mammals using their vocalizations. Tracking marine mammal "singers" with sound in the coastal waters that they frequent is, at this point in time, a well-worn path, but one which still has some interesting offshoots left. Second, one might regard marine mammal vocalizations as noise if using sonar for other purposes. The authors can testify to the amazingly interesting cacophony of sound one can hear in coastal waters (e.g., the Barents Sea, where we all have worked) due to seals, dolphins, and whales. This is a beautiful chorus to listen to, unless you are a sonar operator trying to listen to something else! Third, there is the ambient noise field in shallow water to consider, most especially that due to wind, waves, and shipping and pleasure craft. This acts as a hindrance to marine mammal communications. Next, there is the aspect of marine mammal protection via hazard mitigation – a new technology, but potentially important. And finally, there is the issue of regulation of active sources which are used by natural resource exploration vessels, science vessels, and the military.

Turning to the first topic, localization of vocalizing marine mammals in coastal waters using passive acoustic receivers, we would refer the reader to the article by Cato et al. in Medwin's "Sounds in the Sea" (2005) volume as a nice introduction, after which one will be comfortable in approaching the rather vast bioacoustic literature that exists on this topic. We would just note that while the basic acoustics and signal processing used for localization (time-delay cross-correlation, use of multipath and intensity information, etc.) has remained the same through the years, the technology being employed for this task has improved. AUVs with hydrophones, large-scale naval test ranges with *many* calibrated hydrophones, and even whale tags have been used to track whales and whale pods acoustically, with ever-increasing success. Being able to locate and track marine mammals (and fish) is one of the hardest tasks in marine biology, and passive acoustics is a tool that has provided some measure of success.

As to the sounds that the marine mammals make, many of these are in the range of frequencies of interest to us here (50–5,000 Hz), and many are quite loud – the mammals often need to communicate over many kilometers in shallow water, and evolution has tailored their source levels and frequencies to meet this need. For instance, baleen whales (e.g., finbacks and humpbacks) typically vocalize under 1 kHz and have source levels in the 150–190 dB re 1 μPa-m range, similar to source levels often used for research in shallow water. Odontocete whales often vocalize in our "medium frequency" (1–5 kHz) range. For example, killer whales emit pulses from ~1 to 6 kHz, with peak levels of 160 dB re 1 μPa-m. There is much more to this topic (as always), and we would again refer the interested reader to the volume "Marine mammals and noise" by Richardson et al. (1995) as a basic reference to start with.

The third piece of shallow water acoustics of interest to marine mammal studies is the "general ambient noise field," i.e., the background susurrus due to wind, waves, distant shipping, drilling platforms, pleasure craft, and the like. However, the word "din" or "racket" is perhaps preferable to "susurrus," as the ocean is a noisy place – often due to natural causes, but increasingly due to anthropogenic noise. It has been estimated that the low-frequency ambient noise in the Northern Hemisphere has increased by 10–20 dB over the 10–1,000-Hz band (a number that is rather hard to make exact, due to undersampling issues) solely due to anthropogenic noise, a rather startling finding. (The 2003 National Research Council volume on "Ocean noise and marine mammals" makes an excellent starting point for such an overview.) Whatever the exact number, it is clear that human activities have added to the shallow water noise field (especially, since most anthropogenic maritime activity is coastal) and that this has affected marine mammals. Again, we will forego much detail here in favor of references to a well-established literature.

Mitigation is another active area of acoustics that intersects with marine mammals and shallow water. Two typical mitigation issues that come up are (1) avoidance of ship strikes to marine mammals in shallow water and (2) strategies for "ramping up" louder active sources that could affect marine mammals in their immediate vicinity. Regarding the former topic, there does not seem to be any low- or medium-frequency signal that a ship can send to a surfaced marine mammal to say "Danger! Get out of the way!" at least at present. Warning these animals of a pending collision would be a very useful thing, but unfortunately right whales got their name by being absolutely unresponsive to human activity in their vicinity, which made them the "right whale" to hunt in the heyday of the whaling industry. A more useful tool is the forward-looking obstacle avoidance sonar, which can be purchased commercially. This is higher frequency equipment (e.g., one can look at the system by FarSounder, Inc.) and so is a bit beyond the purview of this book.

Finally, there is the issue of estimating the exposure of marine mammals to low- and medium-frequency sound in shallow water. This has been a major research issue, both from the point of view of noise models and databases (again, see the National Academy 2003 reportsas a starting point) and also from the point of estimating the levels from individual military, industrial, and research sources.

Looking at models and databases, the recent Effects of Sound on the Marine Environment (ESME) model created by the Office of Naval Research is a good

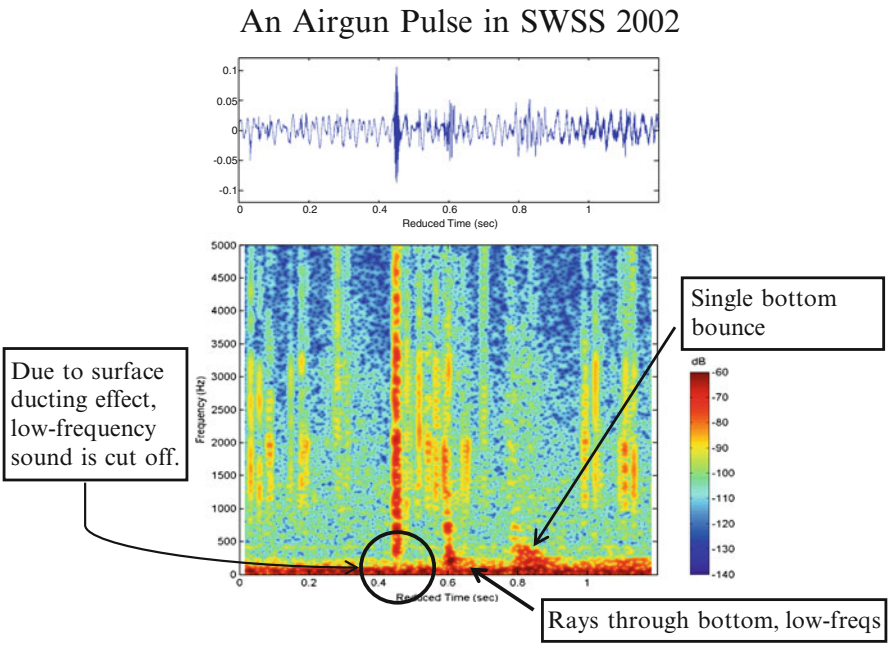


Fig. 2.30 Spectrogram showing energy in acoustic band from seismic source (from DeReuter et al.)

example of an interdisciplinary model (acoustics, biology, physical oceanography, and marine geology) that attempts to calculate exposures and their effects (from both ambient noise and discrete sources) on marine mammals using the latest research results as much as possible (see the IEEE Journal of Oceanic Engineering Special Issue on this topic in 2004 for detailed papers). Moreover, this model has also inspired more research in the areas that were deemed "underdeveloped" for that purpose – a truthful admission that such model outputs are estimates that can be improved.

As an example of direct shallow water research that deals with marine mammal exposures from discrete sources, we will take the work by DeRuiter et al. (2006). In her paper, DeRuiter showed that finescale coastal oceanography effects (in this case, a near-surface duct) could produce some exposure of marine mammals at the surface to the higher frequency components of a seismic airgun signal (see Fig. 2.30). This somewhat surprised the seismic community, which did not worry about detailed oceanography very much – but it was less of a surprise to coastal, shallow water acousticians, who routinely see such effects. Indeed, much of the material in the rest of this book is concerned with just such effects.

We will leave the topic of marine mammals for now, with the realization that all we have done is to point out some major directions in a huge field. Like all the sciences that shallow water acoustics interacts with, marine biology is not only a vast one, but also an important one for acoustic practitioners.

Chapter 3
Foundations of the Theory of Propagation of Sound

3.1 Field of a Point Source in a Layered Waveguide with Absorbing Boundaries

In this chapter, we will consider some simple models of the shallow water waveguide. Such models allow us to obtain and understand the main features of SW sound propagation quickly. Such simple models can also be perturbed to take into account more realistic properties of the environment, thus giving them far more power than one might think at first.

The main quantity that we shall be detailing is the sound pressure, generated by some source, as a function of spatial coordinates and time. The theory in this section presupposes a harmonic time dependence for the source (a "pure-tone or CW source") and, consequently, for the resulting field. This corresponds to simply inserting a factor $\exp(-i\omega t)$ on time-dependent quantities (where ω is the angular frequency, related to the linear frequency by $f = \frac{\omega}{2\pi}$). Work with signals that are modulated or generated by broadband sources (e.g., explosions) presupposes explicit use of Fourier analysis (i.e., we are working with a signal level corresponding to linear acoustics, and those signals satisfying the linear superposition principle).

The coordinate system to be used for this problem is shown in Fig. 3.1, with the z-axis pointing downward. We shall use the notation $\vec{R}$ for the radius vector of the observation point with coordinates (x, y, z). When the problem has cylindrical symmetry, which is often the case, we shall use the cylindrical coordinate system (again see Fig. 3.1):

$$r, \vartheta, z: \quad r = \sqrt{x^2 + y^2}, tg\vartheta = \frac{y}{x}.$$

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_3,
© Springer Science+Business Media, LLC 2012

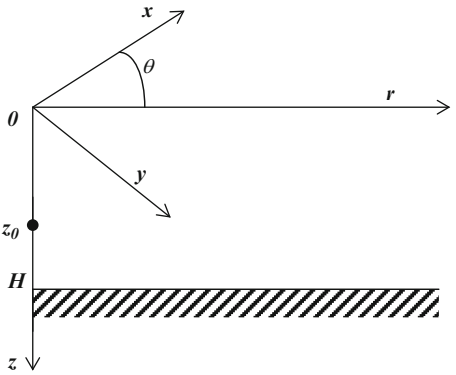


Fig. 3.1 Model of shallow water waveguide

Let us now consider a model of shallow water in the form of a water layer of depth H with density ρ[1] and sound-speed dependence $c(z)$ overlying a fluid, homogeneous half-space of density ρ_1 and sound speed c_1. We characterize the absorptive properties of the bottom by a number $\alpha>0$, determined as follows: for a sound speed in the bottom c_1 and a real-valued bottom refractive index, we define a complex-valued wave vector in the bottom k_1 as

$$k_1 = \frac{2\pi f}{c_1}\left(1 + i\frac{\alpha}{2}\right) = kn_1\left(1 + i\frac{\alpha}{2}\right). \tag{3.1}$$

Here, $n_1 = \frac{c(H)}{c_1} < 1$ is the refractive index of sound in the bottom, and $k = \frac{2\pi f}{c(H)}$.

The absorption coefficient of sound in the bottom, measured in decibels per kilometer, is related to the parameter α by the equation $\hat{\beta}\,(\mathrm{dB/km}) = 27.3\alpha f/c_1$ (where frequency is in hertz, and speed is in km/s). For the case where α can be assumed to be frequency independent, this coefficient gives a linear frequency dependence for the absorption of sound in the bottom. Other descriptions of bottom attenuation are also commonly used: the coefficient of sound absorption in the bottom over frequency, $\hat{\beta}_f(\mathrm{dB/km\,Hz}) = 27.3\alpha/c_1$, is standard; in addition, the coefficient of absorption of sound in the bottom divided by a wavelength is also employed, which is given by the expression $\hat{\beta}_\lambda\,(\mathrm{dB}/\lambda) = 27.3\alpha$.

We now address the problem of determining the sound field[2] due to a point source of unit amplitude, located at coordinates $x = x_0 = 0$, $y = y_0 = 0$, $z = z_0$. This field can be expressed as the solution of the inhomogeneous equation

[1] The dependence of density on the spatial coordinates can be easily taken into account using a change of functions (see Sect. 3.3). The possibility of some depth dependence of the bottom properties is also discussed.

[2] This field is the Green function of the waveguide, determined by the general equation (A.162); in what follows, when not required in the problem, we shall omit the arguments $\vec{R}_0$, ω.

$$[\nabla^2 + k^2(z)]\Psi(\vec{R}) = -\delta(\vec{R} - \vec{R}_0), \tag{3.2}$$

satisfying the radiation condition at infinity. Here, $\vec{R}_0(0, 0, z_0)$ is the radius vector of the source and $k(z) = \frac{2\pi f}{c(z)}$. We can also consider this problem when the right-hand side of (3.2) is equal to zero. In this case, i.e., the homogeneous equation for the function $\Psi(\vec{R})$ as $\vec{R} \to \vec{R}_0$, we have the limit

$$\Psi(\vec{R}) \to \Psi_0(\vec{R}) = \frac{\exp(ik(z_0)|\vec{R} - \vec{R}_0|)}{4\pi|\vec{R} - \vec{R}_0|}. \tag{3.3}$$

Both problems are part of the construction of the Green function for the wave equation, corresponding to an outgoing wave. This problem additionally has the interface boundary conditions

$$\Psi|_{z=0} = 0 \tag{3.4}$$

determined by the pressure-release surface, and the conditions of continuity of the pressure and of the vertical component of the particle velocity across the transition from the water layer to the bottom

$$\Psi|_{z\to H-} = \Psi|_{z\to H+} \frac{1}{\rho}\frac{\partial\Psi}{\partial z}|_{z\to H-} = \frac{1}{\rho}\frac{\partial\Psi}{\partial z}|_{z\to H+}, \tag{3.5a}$$

where the notation $z \to H-$ $(z \to H+)$ denotes the limit as z tends to the boundary from the water and from the bottom layer, respectively. Sometimes, this condition is written as the continuity of the "logarithmic derivative" $\frac{1}{\rho\Psi}\frac{\partial\Psi}{\partial z}$ on transition through the boundary, i.e.,

$$\frac{1}{\rho\Psi}\frac{\partial\Psi}{\partial z}|_{z\to H-} = \frac{1}{\rho\Psi}\frac{\partial\Psi}{\partial z}|_{z\to H+}. \tag{3.5b}$$

Given cylindrical symmetry, (3.2) can be rewritten in cylindrical coordinates independent of the angle ϑ. In this case, we obtain for $\Psi(r, z)$ the so-called modal decomposition (see Appendix B)

$$\Psi(r, z) = \frac{i}{4}\sum_l \Psi_l(z_0)\Psi_l(z)H_0^{(1)}(\xi_l r), \tag{3.6}$$

where $H_0^{(1)}(\xi_l r)$ is Hankel's function, the functions $\Psi_l(z)$ are waveguide modes or eigenfunctions/normal modes, and the values ξ_l are the eigenvalues of the Sturm–Liouville problem in the region $0<z<H$:

$$\frac{d^2\Psi_l(z)}{dz^2} + \left[k^2(z) - \xi_l^2\right]\Psi_l(z) = 0, \tag{3.7a}$$

$$\Psi_l(z)|_{z=0} = 0, \tag{3.7b}$$

$$\left[\Psi_l(z) + g(\xi_l)\frac{d\Psi_l(z)}{dz}\right]_{z=H} = 0 \tag{3.7c}$$

Here,

$$g(\xi) = \frac{m_1}{\sqrt{\xi^2 - k_1^2}}, \tag{3.8}$$

where $m_1 = \frac{\rho_1}{\rho}$. The function $g(\xi)$ is determined by the properties of the bottom[3] and can be computed even for complicated models of the bottom, e.g., those taking into account shear waves, multilayered structure, etc.

The normalization condition for the waveguide modes has the form

$$\int_0^H \Psi_l^2(z)dz + \frac{\Psi_l^2(H)}{2m\sqrt{\xi_l^2 - k_1^2}} = 1. \tag{3.9}$$

Equation (3.7a) can be extended into the region $z>H$ to search for the eigenfunctions. In this case, instead of boundary condition (3.7c) we have a corresponding continuity condition on the bottom, for example, the solution in this region will have the form $\Psi_l(z) = \Psi_l(H)e^{-\sqrt{\xi_l^2 - k_1^2}(z-H)}$, and the normalization condition (3.9) becomes the condition

$$\int_0^H \Psi_l^2(z)dz + \frac{\rho}{\rho_1}\int_H^\infty \Psi_l^2(z)dz = 1. \tag{3.10}$$

Similarly, we obtain the orthogonality condition for the functions $\Psi_l(z)$:

$$\int_0^H \Psi_m(z)\Psi_l(z)dz = -\frac{\Psi_m(H)\Psi_l(H)}{m(\sqrt{\xi_m^2 - k_1^2} + \sqrt{\xi_l^2 - k_1^2})}. \tag{3.11}$$

[3] The function $g(\xi)$ is related to the impedance of the bottomZ_1 by the equation $g(\xi) = iZ_1/\omega\rho$, where the impedance is defined according to Brekhovskikh (1980); the change in sign from Brekhovskikh (1980) is due to the direction of the z-axis. It also is related to the bottom reflection coefficient by $V = (g\sqrt{k^2(H) - \xi^2} + i)/(g\sqrt{k^2(H) - \xi^2} - i)$.

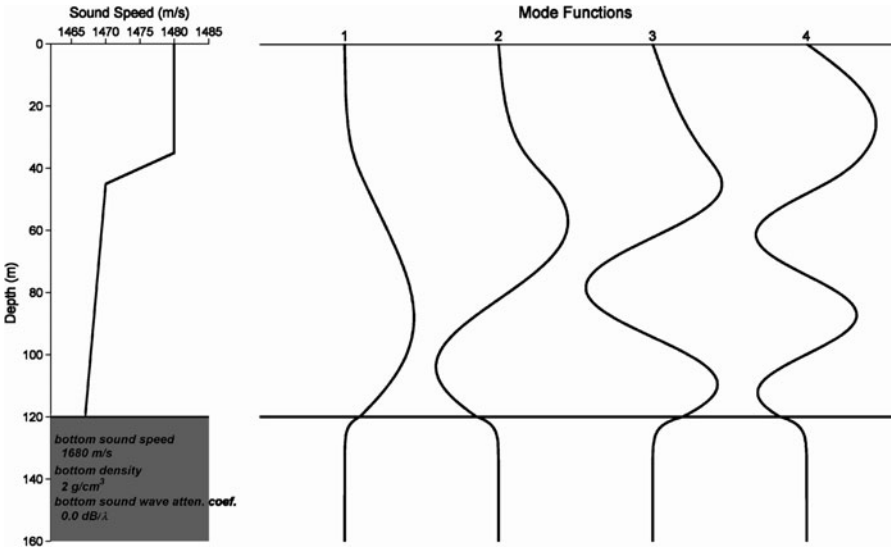


Fig. 3.2 Example of modes shapes for typical waveguide

Hence, (3.6) gives the field of a point source in a waveguide. The summation in the formula extends over all possible values of ξ_l. The values $\mathrm{Re}(\xi_l)>\mathrm{Re}(k_1)$ correspond to the wave vectors of the so-called propagating (trapped) normal wave modes; these modes decay in the bottom. Eigenvalues with $\mathrm{Re}(\xi_l)<\mathrm{Re}(k_1)$ correspond to the so-called leaky modes (modes of the continuous spectrum). The number of normal modes propagating is determined by the physical parameters of the problem: the sound frequency, the waveguide parameters (depth, the sound-speed profile), etc. An example calculation for four trapped modes in 120 m water is shown in Fig. 3.2.

Of course, because of the presence of absorption, a complex impedance appears in the boundary conditions and formally the problem becomes non-self-adjoint. The eigenvalues and eigenfunctions of the (nonproper) Sturm–Liouville problem (3.7) are then complex:

$$\Psi_l(z) = \mathrm{Re}\Psi_l + i\mathrm{Im}\Psi_l, \quad \xi_l = q_l + i\frac{\gamma_l}{2}. \tag{3.12}$$

The small absorption (the small imaginary part of the refractive index in the bottom, small being in comparison with the real part) leads to a small imaginary part of the eigenfunctions and eigenvalues in comparison with their real parts; however, while in practical calculations one can neglect the imaginary part of the eigenfunctions in comparison with their real parts, one cannot do this for the eigenvalues, since the amount of sound energy absorbed increases exponentially with distance.

The value q_l is the wave number of the individual normal mode and the coefficient γ_l is called the modal attenuation coefficient. The phase and group velocities of the mode(s) are associated with these quantities via

$$c_l^{\text{ph}} = \left(\frac{q_l}{\omega}\right)^{-1}, \quad c_l^{gr} = \left(\frac{dq_l}{d\omega}\right)^{-1}. \tag{3.13}$$

In shallow water acoustic problems, the sound field is considered as a rule to be at very large distances from the source, $kr \gg 1$. In this case, we can use the asymptotic expressions in (3.6) for the Hankel functions

$$H_0^{(1)}(\xi_l r) \approx \sqrt{\frac{2}{\pi \xi_l r}} \exp[i(\xi_l r - \pi/4)]. \tag{3.14}$$

Using this expression, we get the Green function at large distances from the source. If we remember that for low frequency ($f \geq 50$ Hz) sound waves we have $k \geq 0.1$ m^{-1}, for distances as small as $r \geq 100$–200 m we can already use the asymptotic formula (3.14). Thus, for the Green function (3.6), we now have

$$\Psi(r,z) = \sqrt{\frac{i}{8\pi}} \sum_l \frac{\Psi_l(z_0)\Psi_l(z)}{\sqrt{\xi_l r}} e^{i\xi_l r}. \tag{3.15}$$

This Green function is the "outgoing wave" Green function. It is also possible to construct a Green function, corresponding to "incoming" waves, or containing both types of waves. The latter situation takes place routinely in irregular waveguides, where we must take into account both outwardly radiated and back scattered waves. We will come back to this situation later.

It is worth discussing our solution somewhat further. First, we can consider (3.6) and (3.15) as the sound field of point source decomposed in a series of waveguide modes at the point of observation $\Psi_l(z)$, these modes also being a set of orthogonal functions. Every member of the sum (3.6) or (3.15) is also a solution of the wave equation, so we can introduce the "modal sound pressure," i.e., the partial sound pressure provided by each mode. Furthermore, because the depth dependence is due solely to the eigenfunction, we can introduce "modal amplitude," which does not depend on depth. The modal phase and cylindrical spreading loss are also easily separated out in this form of the sound field. In other words, we can write

$$\Psi(r,z) = \sum_l \frac{C_l \Psi_l(z)}{\sqrt{\xi_l r}} e^{iq_l r}, \tag{3.16}$$

where $C_l = \sqrt{\frac{i}{8\pi}} \Psi_l(z_0) e^{-\frac{\gamma_l}{2} r}$.

We see that expression $C_l e^{iq_l r}/\sqrt{\xi_l}$ is the modal coefficient of the sound field (up to the cylindrical spreading factor). For the case where the waveguide does not

change in range and if there is no significant attenuation, the amplitude $\left|C_l e^{i\xi_l r}/\sqrt{\xi_l}\right| = \Psi_l(z_0)/\sqrt{8\pi|\xi_l|}$ does not depend on range. Thus, up to cylindrical spreading and attenuation, the modal amplitude remains the same and is determined solely by the excitation coefficient of the mode. For a point source, the excitation coefficient is proportional to the value of the mode at the point of the source. We note that for an irregular waveguide, this coefficient should be taken as a function of horizontal coordinates (or range and polar angle), and (in the nonstationary case) can depend on time.

In Sect. A.3, we introduced the intensity of the sound field passing through a waveguide (A.170). Let us also introduce the spectral intensity $\vec{I}^h_\omega$ in the horizontal plane of a source with the spectrum $S(\omega)$

$$\vec{I}^h_\omega = \frac{2\pi i}{\rho\omega}|S(\omega)|^2(\Psi\nabla_-\Psi^* - \Psi^*\nabla_-\Psi), \tag{3.17}$$

where $\nabla_- = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$. In the cylindrically symmetric case, we can get the radial component of the spectral intensity I^r_ω of the point source if we substitute the modal decomposition (3.16) into (3.17), taking into account that $\xi_l = q_l + \frac{\gamma_l}{2}$ and $\gamma_l \ll q_l$. We obtain

$$I^r_\omega = \frac{2\pi|S(\omega)|^2}{\rho\omega r}\sum_{l,m}\frac{C_l C^*_m}{\sqrt{q_l q_m}}(q_l + q_m)\Psi_l\Psi_m e^{i(q_l - q_m)r}. \tag{3.18}$$

A simple and informative quantity is the spectral intensity integrated over depth (a waveguide intensity or waveguide flux)

$$\vec{I}^w_\omega = \int \vec{I}^h_\omega dz. \tag{3.19}$$

For our case, taking into account the orthogonality of the eigenfunctions, we get for the radial component

$$I^w_\omega = \frac{4\pi|S(\omega)|^2}{\rho\omega r}\sum_l |C_l|^2. \tag{3.20}$$

On the basis of this relationship, we can define the spectral modal intensity (the power carried by a mode through the waveguide for a unit length across the waveguide)

$$I^w_{l\omega} = \frac{4\pi}{\rho\omega r}|S(\omega)|^2|C_l|^2. \tag{3.21}$$

If we multiply expressions (3.20) and (3.21) by $2\pi r$, we get both the total and modal power, carried by the trapped modes over all directions:

$$\bar{W}_{\omega} = \frac{8\pi^2}{\rho\omega}|S(\omega)|^2 \sum_l |C_l|^2 \tag{3.22}$$

and

$$\bar{W}_{l\omega} = \frac{8\pi^2|S(\omega)|^2}{\rho\omega}|C_l|^2. \tag{3.23}$$

Next, let us find the field in the waveguide $P(r,z)$ generated by a monochromatic acoustic source with power $\bar{W}_0$. This corresponds to some nonunit amplitude A, the factor in front of the δ-function on the right-hand side of (3.2) or in front of $\Psi_0(\vec{R})$ in (3.3). We can determine the power of the source (or the connection between $\bar{W}_0$ and A) as follows. Consider a volume in the neighborhood of the source, sufficiently small $\left(\vec{R} \to \vec{R}_0\right)$ so that the medium in this region may be assumed to be homogeneous ($c = c(z_0) = c_0,\ \rho = \rho(z_0) = \rho_0$), and also neglect the effect of the waveguide boundaries. The time-dependent field generated by the source in this neighborhood is $A\Psi_0(\bar{R})\exp(-i\omega t)$. Calculating the power using (A.161), and integrating over the surface surrounding the source, we obtain

$$A = \sqrt{8\pi\rho_0 c_0 \bar{W}_0}. \tag{3.24}$$

Taking this into account and substituting (3.24) in (3.16), we obtain for the field of a source of power W_0, a time-dependent expression for the pressure $P(r,z,t)$:

$$P(r,z,t) = \sqrt{i\rho_0 c_0 \bar{W}_0} \sum_l \frac{\Psi_l(z_0)\Psi_l(z)}{\sqrt{\xi_l r}} \exp[i(\xi_l r - \omega t) - \beta r]. \tag{3.25}$$

This expression also takes into account the absorption in the water. We shall also use (3.25) in considering the complex envelope $P(r,z)$ of the field of an acoustic source, simply by omitting the factor $e^{-i\omega t}$. Also, if absorption is not significant, we can omit the quantity β.

3.2 The Pekeris Model

The simplest model of a shallow water waveguide having all the main characteristic properties is the model proposed by Chaim Pekeris (1948). In this model, the water layer is assumed to have a constant depth, sound speed, and density $H = \text{const}$, $c(z) = c = \text{const}$, $\rho = \text{const}$, respectively, and the bottom parameters are also

assumed to be constants c_1, ρ_1, α. The Pekeris model is often used for putting forward qualitative arguments and giving quantitative estimates in the description of various phenomena in shallow water. A detailed analysis of the behavior of the sound field in this model (in other words, in a two-layer waveguide) is given in many places in the literature, e.g., Gazarian (1958), Brekhovskikh (1980), DeSanto (1979). We shall consider only those properties and relationships which we shall use later for estimates and analysis.

Eigenfunctions $\Psi_l(z)$ can be easily constructed from the modal decomposition (3.7) because (3.7) has an analytical solution when k is constant. However, in our work here, we shall obtain the Green function Ψ using the method discussed in Appendix B. In this case, (B.15) and (B.16) for the construction of the field of a point source have analytic solutions

$$U(\xi, z) = \sin\sigma(z-H) - g\sigma\cos\sigma(z-H), \quad V(\xi, z) = \sin\sigma z. \tag{3.26}$$

Here, we have defined $\sigma = \sqrt{k^2 - \xi^2}$. The Wronskian of this system of solutions is

$$W(\xi) = -\sigma(\sin\sigma H + g\sigma\cos\sigma H), \tag{3.27}$$

and its zeros (the eigenvalues of the normal modes) are defined by the equation

$$\tan\sqrt{k^2 - \xi_l^2}H = -m_1\frac{\sqrt{k^2 - {\xi_l}^2}}{\sqrt{{\xi_l}^2 - k_1^2}}. \tag{3.28}$$

Equation (3.28) has an infinite number of roots in the complex plane ξ (the real and imaginary axes are Reξ and Imξ, respectively), whose position depends on the parameters of this equation, or, more precisely, on the quantity $v = \frac{H}{\pi}\sqrt{k^2 - k_1^2}$.

It is convenient to carry out a preliminary analysis of (3.28) in the limiting case $k_1 = k$. For the poles of the integrand in (B.21), we obtain the equation

$$\tan\sqrt{k^2 - \xi_l^2}H = -im_1. \tag{3.29}$$

Since for $z>H$, there can only exist outgoing waves, then in accordance with the choice of the signs of the roots,

$$\sqrt{\xi_l^2 - k_1^2} = -i\sqrt{k_1^2 - \xi_l^2}. \tag{3.30}$$

We then seek solutions of (3.28) setting $\sqrt{k^2 - \xi_l^2} = \sigma_l' + i\sigma_l''$. Equation (3.29) gives two families of points

$$\tan(\sigma_l' H) = 0, \quad \tanh(\sigma_l'' H) = -m_1 \tag{3.31}$$

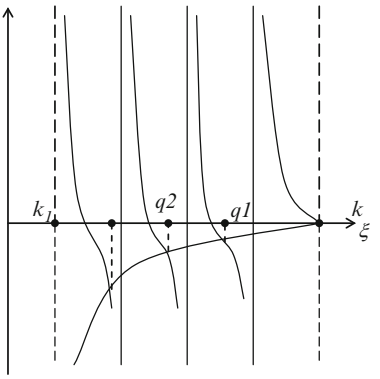


Fig. 3.3 Graphical solution of the transcendental equation

and also

$$\operatorname{ctan}(\sigma_l' H) = 0, \quad \tanh(\sigma_l'' H) = -\frac{1}{m_1}. \tag{3.32}$$

Since we are assuming that $m_1 > 1$, it follows that only the second family remains. That is, $\sigma_l' = \left(l + \frac{1}{2}\right)\frac{\pi}{H}$. Moreover, from (3.32) it is clear that $\sigma_l'' = \operatorname{Im}\sqrt{k^2 - \xi_l^2} = -\frac{1}{H}\operatorname{arctanh}\frac{1}{m_1} < 0$, which corresponds to an unphysical sheet of the complex plane. Moving to the values of ξ_l, we obtain

$$\xi_l^2 = k^2 + (\sigma_l'')^2 - \left[\left(l + \frac{1}{2}\right)\frac{\pi}{H}\right]^2 - i(2l+1)\frac{\pi\sigma''_l}{H}.$$

It is clear that since ξ_l^2 are located in the upper half-plane, the poles are found in the first and third quadrants of an unphysical sheet of the complex plane, and for most values of the indices are located almost equidistantly along the imaginary axis. When $k > \operatorname{Re}(k_1)$, there may exist roots of (3.28) lying on a physical sheet of the complex plane.

The solution of the equation for real k_1 (i.e., when there is no absorption) can be illustrated graphically (Fig. 3.3). It is clear that the poles (all of which are located between the points k_1 and k) are determined by the intersection points of branches of the tangent (the function $\tan\sqrt{k^2 - \xi^2}H$) and the function $f(\xi) = -m_1\frac{\sqrt{k^2-\xi^2}}{\sqrt{\xi^2-k_1^2}}$. As the difference $k - k_1$ increases, and, correspondingly, the point $\xi = k_1$ moves to the left relative to the point $\xi = k$, new poles appear, or more precisely, poles of the unphysical sheet move onto the physical sheet. The first of these appears for $\sqrt{k^2 - k_1^2} > \pi/2H$. The total number of poles M is determined by the number of branches of the function $\tan\sqrt{k^2 - \xi^2}H$, lying between the points $\xi = k_1$ and $\xi = k$, and is equal to

$$M = E\left\{(H/\pi)\sqrt{k^2 - k_1^2}\right\}, \tag{3.33}$$

where $E\{\}$ is the integral part of the function. We note further that, from Fig. 3.3, it is clear that for arbitrary variations of the parameters of the waveguide, the value of the transverse wavenumber belongs to the interval $(l - 1/2)\frac{\pi}{H} < \sigma_l < l\frac{\pi}{H}$. It can be written in the form

$$\sigma_l = \frac{l^*\pi}{H}, \tag{3.34}$$

where l^* is a number such that $l - 1/2 \le l^* \le l$. The extreme values correspond to the limiting cases of a rigid and a pressure-release bottom, respectively. Another variant corresponds to a change in depth, i.e.,$\sigma_l = \frac{l\pi}{H^*}$, where H^* is the "effective depth."

For the case of fixed values of depth and constant sound speed in the bottom and in the water layer, the parameter determining the number of normal modes is the sound frequency. We define the quantity $f = f_l$, determined by the equation $2f_l H\sqrt{c^{-2} - c_1^{-2}} = l - 1/2$, as the cut-off frequency (or critical frequency) for modes with index l. This means that if the frequency $f < f_l$, then there exists $l - 1$ propagating modes, and when the frequency passes through the value $f = f_l$, the mode with index l appears.

In the general case of a complex value of k_1, the poles are located in the upper half-plane. Assuming that the absorption is low ($\alpha \ll 1$), the poles are located near the real axis, and our earlier considerations concerning their number and behavior remain valid.

Let us now briefly consider separately the two main component fields of a point source for the Pekeris model, i.e., the contributions of the discrete and continuous spectra, respectively.

The contribution of the discrete spectrum of the point source Ψ_d is determined by the expression[4]

$$\Psi_d(r, z) = \frac{i}{4}\sum_{l=1}^{M} \Psi_l(z_0)\Psi_l(z)H_0^{(1)}(\xi_l r), \tag{3.35}$$

where $\Psi_l(z)$ are the eigenfunctions

$$\Psi_l(z) = N_l\begin{cases} \sin(\sigma_l z), & 0 \le z \le H, \\ \sin(\sigma_l H)e^{-\kappa_l(z-H)}, & z > H. \end{cases} \tag{3.36}$$

where $\kappa_l^2 = \xi_l^2 - k_1^2$.

[4] In other sections we shall neglect the contribution of the continuous spectrum, and the notation Ψ will be used for the field generated by the discrete spectrum.

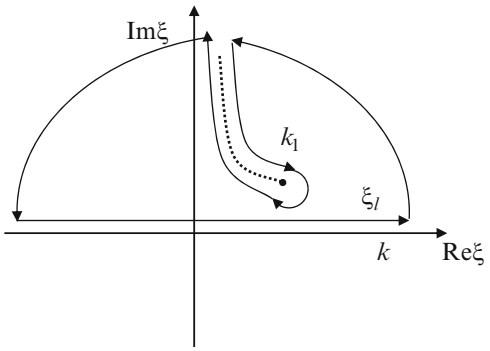


Fig. 3.4 Contour of integration in the complex plane ξ for computation of the Green function. The branch point in the complex plane is the value of the wave vector in the bottom. The complex eigenvalues shown are zeros of the Wronskian

The normalization factors for the eigenfunctions of the discrete spectrum are determined similarly to those in the preceding section using (3.10)

$$N_l = \left(\frac{H}{2} + \frac{m_1}{2\kappa_l}\frac{k^2 - k_1^2}{\kappa_l^2 + {m_1}^2\sigma_l^2}\right)^{-1/2}. \tag{3.37}$$

Let us now consider the contribution of the continuous spectrum (denoted by Ψ_c). As discussed earlier, it is determined by the integral over the branch line, beginning at the branch point $\xi = k_1$. For ease of computation, the branch line is often deformed. There are a number of different ways of transforming the branch line. For an asymptotic estimate of the integral, the branch line is drawn in the direction of steepest descent (and thus a lateral wave is defined); in Pekeris (1948), the branch line was taken to be a straight line running parallel to the imaginary axis through the point $\xi = k_1$. Let us here consider as the branch line the upper part of the hyperbola $\mathrm{Re}\xi \times \mathrm{Im}\xi = \frac{\alpha}{2}k^2{n_1}^2$, passing through a branch point (see Fig. 3.4). A section of this type corresponds to the so-called EJP method (Ewing. Jardetzky and Press 1957). Neglecting absorption in the lower half-space, the section line represents the region from the point $\xi = k_1$ to zero and from zero to infinity along the imaginary axis. In this case, the quantity $\sqrt{\xi^2 - k_1^2}$ becomes purely imaginary and we can use the expression

$$g(\xi) = \frac{im_1}{\sqrt{k_1^2 - \xi^2}}. \tag{3.38}$$

The sum of integrals over the section is thus given by the expression

$$\begin{aligned}\Psi_c(r,z) = &\frac{1}{4\pi}\int_{k_1}^{i\infty}\frac{U^{(+)}(\xi,z_>)V^{(+)}(\xi,z_<)}{W^{(+)}(\xi)}H_0^{(1)}(\xi r)\xi d\xi \\ &+\frac{1}{4\pi}\int_{i\infty}^{k_1}\frac{U^{(-)}(\xi,z_>)V^{(-)}(\xi,z_<)}{W^{(-)}(\xi)}H_0^{(1)}(\xi r)\xi d\xi,\end{aligned} \tag{3.39}$$

where the signs (+) and (−) denote the values of the functions on the left- and right-hand segments of the section, respectively. They are distinguished by the sign of the root $\sqrt{k_1^2 - \xi^2}$ in the impedance. Substituting the values of the functions into this expression, we obtain for the contribution of the continuous spectrum

$$\Psi_c(r,z) = \frac{i}{4}\int_{k_1}^{i\infty} \frac{m_1}{\pi\kappa} \frac{\sin(\sigma z)\sin(\sigma z_0)}{\sin^2(\sigma H) + \frac{{m_1}^2\sigma^2}{\kappa^2}\cos^2(\sigma H)} H_0^{(1)}(\xi r)\xi d\xi. \tag{3.40}$$

The meaning of the integrand can easily be understood by reaching a comparison on how this was done in the previous section. For values $\xi^2 < k_1^2$, we define the modes of the continuous spectrum $\Psi_\xi(z)$, "sewing together" the solution on the boundary $z = H$:

$$\Psi_\xi = N_\xi \begin{cases} \sin\sigma z, & 0<z<H \\ \sin\sigma H\cos\kappa(z-H) + i\frac{m_1\sigma}{\kappa}\cos\sigma H\sin\kappa(z-H), & z>H \end{cases}. \tag{3.41}$$

Here, ξ, σ and κ take on a continuous set of values. Let us now calculate the overlap integral of the two different functions of the continuous spectrum:

$$\int_0^H \Psi_\xi(z)\Psi_{\xi'}(z)dz + \frac{\rho}{\rho_1}\int_0^\infty \Psi_\xi(z)\Psi_{\xi'}(z)dz = \pi A_\xi^2 \frac{\delta}{m_1}(\sin^2\sigma H + \frac{{m_1}^2\sigma^2}{\delta^2}\cos^2\sigma H)\delta(\xi^2 - (\xi')^2). \tag{3.42}$$

In the above, we have used the equations

$$\int_0^\infty \sin x\tau d\tau = P\frac{1}{x}, \quad \int_0^\infty \cos x\tau d\tau = \pi\delta(x), \tag{3.43}$$

where P denotes the Cauchy Principal Value integral.

The functions $\Psi_\xi(z)$ will be normalized by the δ-function if we take

$$N_\xi = \left[\frac{\pi\kappa}{m_1}\left(\sin^2\sigma H + \frac{{m_1}^2\sigma^2}{\kappa^2}\cos^2\sigma H\right)\right]^{-1/2}. \tag{3.44}$$

Thus, the field of a point source, taking into account the continuous spectrum, can be written in the form

$$\Psi(r,z) = \frac{i}{4}\sum_{l=1}^{M}\Psi_l(z_0)\Psi_l(z)H_0^{(1)}(\xi_l r) + \frac{i}{4}\int_{k_1}^{i\infty}\Psi_\xi(z_0)\Psi_\xi(z)H_0^{(1)}(\xi r)\xi d\xi. \quad (3.45)$$

As mentioned earlier, the continuous spectrum, as a rule, gives a small contribution to the field, and at sufficiently large distances (a few water depths, usually) it is neglected; however, there exists situations when it has to be taken into account. This continuum is studied in great detail in the literature (e.g., Brekhovskikh 1980; Brekhovskikh and Godin 1992; Gazarian 1958; Stickler and Ammicht 1980; Kuperman and Ferla 1985). Of particular interest is the simple, physically insightful solution of the branch line integral by the so-called virtual modes (Tindle et al. 1976).

When absorption in the bottom is taken into account, the value of k_1^2 becomes complex, with a small imaginary part $\alpha k^2 {n_1}^2$. In this case, the imaginary part produces a pole ξ_l, moving in the upper half-plane: $\xi_l = q_l + i\gamma_l/2$, where the coefficient γ_l is called the mode attenuation coefficient. The numerical value of α is usually two or three orders of magnitude less than unity so that $\gamma_l \ll q_l$. Assuming a low absorption in the bottom, one can define the absorption coefficient of a mode of the discrete spectrum γ_l. Here, the real values σ_l and κ_l have to be replaced by complex expressions. Retaining the notation σ_l and κ_l for the real parts of these quantities, we have, respectively, the complex quantities $\sigma_l - i\frac{q_l\gamma_l}{2\sigma_l}$ and $\kappa_l + i\frac{\gamma_l q_l - \alpha k^2 {n_1}^2}{2\kappa_l}$. Substituting these expressions in the transcendental (3.28), we obtain for the mode attenuation coefficient in the Pekeris model

$$\gamma_l = \alpha\frac{N_l^2\sigma_l^2 k^2 {n_1}^2 m_1}{2\beta_l\kappa_l(\kappa_l^2 + {m_1}^2\sigma_l^2)}. \quad (3.46)$$

If we consider the dependence of the mode spectrum on the index of the mode, the distance between the modes and the absorption coefficient in the Pekeris model grows with the index of the mode.

Let us now consider the behavior of the mode attenuation coefficients as a function of frequency. This can be done based on (3.46), where the normalization coefficient is determined by (3.44). Suppose that for a given mode index, the frequency tends from above to the critical frequency for that mode index, remaining somewhat larger. In this case, $q_l \to kn_1$ and, thus accordingly, for the attenuation coefficient, $\gamma_l \to \alpha k n_1$. The meaning of the latter limit is that when the frequency approaches the critical frequency, the mode becomes a mode of the continuous spectrum, i.e., it is "unconstrained" and propagates in the bottom to a greater extent. Accordingly, the attenuation coefficient tends to the coefficient of sound absorption in the bottom at that given frequency. For this same mode if the frequency grows, then as can be easily seen from (3.46), the value of γ_l tends monotonically to zero, in proportion to $1/f^2$ for large values of the frequency. This is also easy to understand, since as the mode frequency increases, it penetrates less and less into the bottom and the proportion of the function propagating in the bottom tends to zero.

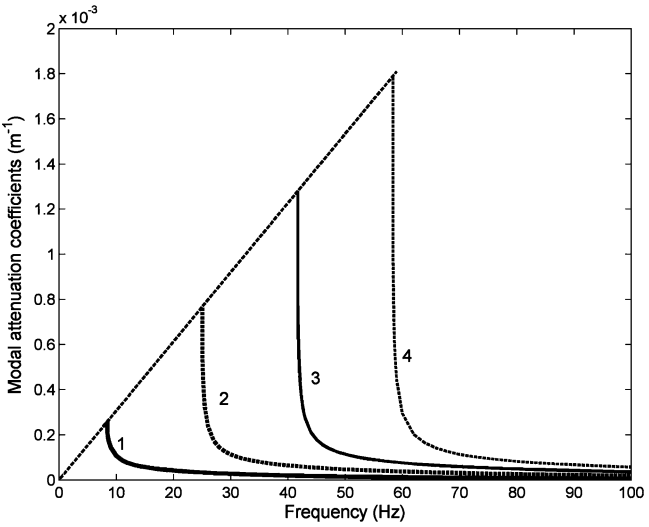


Fig. 3.5 Frequency dependence of the mode attenuation coefficients. The straight dashed line corresponds to linear dependence of the sound absorption coefficient in the bottom, determined by our condition for an absorptive bottom. Waveguide parameters are $H = 100$ m, $c = 1{,}500$ m/s, $c_1 = 1{,}680$ m/s, $\rho_1/\rho = 2$, $\alpha \sim 0.015$

Figure 3.5 shows the behavior of the mode attenuation coefficients as a function of frequency for some typical model parameters.

Analogously, we can consider the behavior of the mode attenuation coefficient as a function of the depth H. If we fix some mode with index l and decrease the depth H, then as H nears some critical depth H_l, it is clear from (3.46) and (3.30) that the coefficient γ grows, giving in the limit a value $\alpha k n_1$, which is the coefficient of sound absorption in the bottom. The limit of decreasing depth can also be interpreted as looking at the propagation of a mode in a waveguide with smoothly decreasing depth, for example, in a coastal wedge moving toward shore. In this situation, the absorption of the mode increases, and at the point where the critical depth is reached, the mode transfers to the continuous spectrum and is quickly absorbed by the bottom. The direction of energy flow in the bottom in the region of the critical depth is determined by the vector $(k\sqrt{1 - {n_1}^2}, \sigma_l)$ in the water layer and, in the language of rays, corresponds to the incidence of the ray at the angle of total internal reflection. The greater the mode index, the further from the coast its exit to the bottom occurs. Numerical analysis of such propagation based on a parabolic equation (PE) method has been carried out, for example, in Jensen (1981). As the depth increases, the attenuation coefficient of the mode decreases as H^{-3}.

Having looked closely at the phase velocities of the modes, we should note that it is also possible (if tedious) to calculate the group velocities c_l^{gr} of the propagating modes. This gives

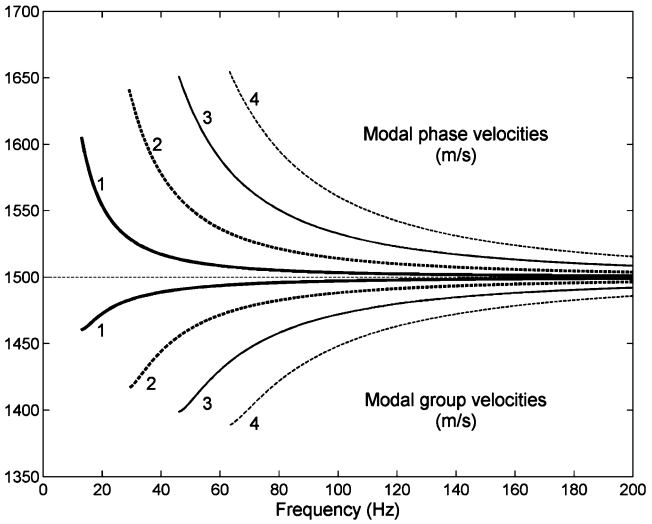


Fig. 3.6 Frequency dependence of phase and group velocities above cut-off for Pekeris model with parameters as in the Fig. 3.5

$$c_l^{gr} = \frac{d\omega}{dq_l} = c^2 \frac{q_l}{\omega} \frac{\frac{m_1\sigma_l}{\kappa_l^2} + \frac{\kappa_l H}{\sigma_l}\left(1 + \frac{{m_1}^2\sigma_l^2}{\kappa_l^2}\right) + \frac{m_1}{\sigma_l}}{\frac{m_1\sigma_l}{\kappa_l^2} n_1^2 + \frac{\kappa_l H}{\sigma_l}\left(1 + \frac{{m_1}^2\sigma_l^2}{\kappa_l^2}\right) + \frac{m_1}{\sigma_l}}. \tag{3.47}$$

Typical frequency dependence of phase and group velocities is shown in Fig. 3.6 for frequencies above cut-off for every mode. The behavior of the aforementioned functions for frequency below cut-off is discussed in the literature (Frisk 1994).

To conclude this section, we would note that despite its simplicity, the Pekeris model is not only used quite often for estimates and qualitative arguments, but can also be successfully applied to the analysis of concrete problems; see, for example, the studies of noise in Buckingham (1980).

3.3 Perturbation Theory and WKB Methods

As seen in the previous sections, the main task in the construction of the sound field in a shallow water waveguide is to find (calculate) the waveguide modes (or eigenfunctions) $\Psi_l(z)$ and the mode eigenvalues ξ_l of the Sturm–Liouville problem (3.7a) with boundary conditions (3.7b–3.7c). In the previous sections, these functions were constructed in analytical form for the simple case $c(z) = c$ = constant (the Pekeris model). However, even for this model, eigenvalues cannot be found analytically as a rule. In spite of this model giving the salient features of sound propagation in shallow water, we still need more realistic models of the sound-speed profile.

Generally speaking, (3.7) has analytical solutions only for a rather limited number of profiles $c(z)$ (or $k(z)$). For example, the linear profile $c(z) = c_s(1 - az)$ has solutions of (3.7) given by the Airy functions (Brekhovskikh 1974); there are also other types of profiles, giving analytical solutions of (3.7) (Brekhovskikh 1974). However, for real (experimentally measured) sound-speed profiles $c(z)$, we generally cannot solve (3.7) and so we should either solve this equation approximately or numerically. As a rule, when comparing with experiment, we should solve this equation numerically, because approximate methods do not give the necessary accuracy; however, to carry out estimations to understand or interpret results, we can use approximate methods. The most widespread approximate methods to solve (3.7) are perturbation theory (PT) and WKB methods, which are used under different conditions.

Perturbation theory is used in the calculation of corrections to some "reference" solution, essentially to the eigenvalues and eigenfunctions. In PT, the equations for the corrections can be represented in various forms (Burov and Sergeev 1991, 1992; Davydov 1965; Turbiner 1984). In underwater acoustics, PT is also used to calculate corrections to the spectrum due to changes in the waveguide characteristics, both in the water column and in the bottom. Based on perturbation theory, a method for recovering waveguide characteristics (in particular, the sound-speed profile and density in the bottom) has been proposed (Rajan et al. 1987). The actual formulae of perturbation theory for shallow water acoustics exhibit properties resulting from the non-self-adjoint nature of the Sturm–Liouville–Louisville problem, due to absorption. This contributes small corrections to the basic formulae in comparison with the well-known Rayleigh–Schrodinger formulae (Turbiner 1984). Some works have also considered different forms for the higher-order perturbative corrections for underwater acoustics (Burov and Sergeev (1992); Gindler 1990).

To formulate the theory of perturbations, it is both convenient and interesting to use the model of a shallow water waveguide with no division between the water and bottom layer, and simply represent the propagation in the medium as a dependence of the wave vector (sound speed) and the density on depth: $k(z)$, $\rho(z)$. To construct a perturbation theory which enables us to take into account the change in density, which is often important for shallow water acoustics, we have to use a wave equation for a medium with variable (in this case, depending on just one coordinate) density. The equation for the eigenfunctions in this case will have a somewhat different form from (3.7) [e.g., Isakovich (1973)]:

$$\frac{d^2\Psi_l(z)}{dz^2} + \rho(z)\frac{d}{dz}[\rho^{-1}(z)]\frac{d\Psi_l(z)}{dz} + [k^2(z) - \xi_l^2]\Psi_l(z) = 0, \tag{3.48}$$

where $\Psi_l(0) = 0$, and $\Psi_l(z) \to 0$ as $z \to \infty$. The eigenfunctions must satisfy the normalization condition

$$\int_0^\infty \frac{\rho_s}{\rho(z)}\Psi_m(z)\Psi_l(z)dz = \delta_{ml}, \tag{3.49}$$

where ρ_s is some fixed value of the density (in this case, that near the water surface), and δ_{ml} is the Kronecker delta symbol. To construct our perturbation theory, we make the following useful substitution in (3.7a):

$$\tilde{\Psi}_l(z) = \sqrt{\frac{\rho_S}{\rho(z)}}\Psi_l(z) \tag{3.50}$$

as a result of which we have the equation for the function $\tilde{\Psi}_l(z)$,

$$\frac{d^2\tilde{\Psi}_l(z)}{dz^2} + [\tilde{k}^2(z) - \xi_l^2]\tilde{\Psi}_l(z) = 0, \tag{3.51}$$

where

$$\tilde{k}^2(z) = k^2(z) - \frac{1}{2}\sqrt{\frac{\rho(z)}{\rho_S}}\frac{d^2}{dz^2}\sqrt{\frac{\rho_S}{\rho(z)}}. \tag{3.52}$$

The functions $\tilde{\Psi}_l(z)$ are normalized according to the condition (3.49)

$$\int_0^\infty \tilde{\Psi}_m(z)\tilde{\Psi}_l(z)dz = \delta_{ml}. \tag{3.53}$$

Let $k^0(z)$ and $\rho^0(z)$, respectively, be the "unperturbed" values of the wave vector and the density of the medium, for which the solution of the Sturm–Liouville problem (3.48) is known, being denoted here by $\tilde{\Psi}_l^0(z)$ and ξ_l^0. According to our previous work, we have the unperturbed quantity $k^0(z)$ and spectra $\tilde{\Psi}_l^0(z)$ and ξ_l^0. We shall assume that the quantities $k^0(z)$ and $k(z)$ are complex with a small imaginary part (in comparison with the real part). We shall further assume that the quantities $k(z)$ and $\rho(z)$ exhibit only small differences from their unperturbed values (we shall make the criteria for the smallness of these differences more precise in what follows):

$$k^2(z) - [k^0(z)]^2 = \delta k^2(z), \quad \rho(z) - \rho^0(z) = \delta\rho(z). \tag{3.54}$$

Correspondingly, we have the perturbation $\tilde{k}^2(z) - [\tilde{k}^0(z)]^2 = \delta\tilde{k}^2(z)$ which can be expressed in terms of perturbations of the density $\delta\rho(z)$ and of the sound speed, $\delta\tilde{k}^2(z)$ separately. Using standard procedure [see Schiff (1968), and also see our later sections on inverse theory], the following expressions, to second order in the eigenvalues and to first order in the eigenfunctions, can be obtained:

$$\xi_l^2 = (\xi_l^0)^2 + V_{ll} + \sum_{m\neq l}\frac{V_{ml}^2}{(\xi_l^0)^2 - (\xi_m^0)^2}, \tag{3.55a}$$

$$\tilde{\Psi}_l(z) = \tilde{\Psi}_l^0(z) + \sum_{m \neq l} \frac{V_{ml}}{\left(\xi_l^0\right)^2 - \left(\xi_m^0\right)^2} \tilde{\Psi}_m^0(z), \tag{3.55b}$$

where we have denoted

$$V_{ml} = \int_0^\infty \tilde{\Psi}_m^0(z) \delta\tilde{k}^2(z) \tilde{\Psi}_l^0(z) dz = \int_0^\infty \frac{\rho_S}{\rho^0(z)} \Psi_m^0(z) \delta\tilde{k}^2(z) \Psi_l^0(z) dz.$$

Let us consider in more detail the correction $\delta\xi_l = \xi_l - \xi_l^0$ to the propagation constant of a mode with index l, which in first-order perturbation theory has the form

$$\delta\xi_l = \frac{1}{2\xi_l^0} \int_0^\infty \frac{\rho_S}{\rho^0(z)} \left[\Psi_l^0(z)\right]^2 \delta\tilde{k}^2(z) dz. \tag{3.56a}$$

Because of absorption, the value of this correction is complex, and its imaginary part gives a correction to the mode attenuation coefficient for modes with index. In particular, we can consider the coefficient of absorption of sound in the bottom, which generally depends on depth, as a perturbation to the real problem without absorption, that is, $\delta\tilde{k}^2 = ik^2 {n_1}^2(z)\alpha(z)$ for $z>H$, which gives us the mode absorption coefficients:

$$\gamma_l = \frac{k^2}{q_l} \int_H^\infty \frac{\rho_S}{\rho^0(z)} \left[\Psi_l^0(z)\right]^2 {n_1}^2(z)\alpha(z) dz, \tag{3.56b}$$

where the q_l^0 denotes real eigenvalues of the unperturbed problem. For realistic parameters of the shallow water waveguide, calculations of the mode absorption coefficients using (3.56b) give completely acceptable accuracy, which makes perturbation theory a convenient means of calculating and studying the absorption of sound in the bottom.

Another perturbation theory application is to calculate the influence of internal waves on the spectrum of eigenvalues. This calculation is useful for understanding the propagation of sound waves in the presence of internal waves, both linear background and soliton-like ones. The perturbation to the wave vector $\delta\tilde{k}^2$ can be found on the basis of (2.31) for the perturbation of the sound speed in the presence of IW:

$$\delta c = \tilde{Q} c_0 \tilde{N}^2(z) \zeta(x, y, z),$$

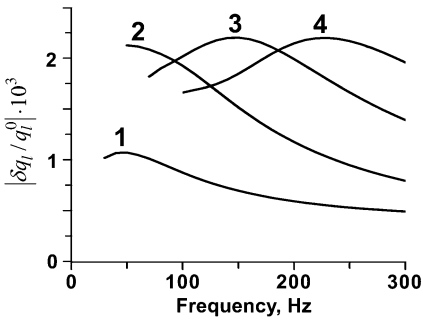


Fig. 3.7 Relative corrections to eigenvalues calculated using perturbation theory as a function of frequency for waveguide with unperturbed sound speed profile shown in the figure

e.g., here $\tilde{N}(z)$ is the buoyancy frequency, and $\tilde{Q}$ is a coefficient, which in shallow water can be taken as ~2.4 m^2/s. If $\delta\tilde{k}^2 \sim -2k_0^2\delta c/c_0$, then we can get for the correction to eigenvalue

$$\delta q_l = -\frac{k_0^2}{q_l^0}\int_0^H \tilde{N}^2(z)[\Psi_l^0(z)]^2\zeta(\vec{r},z)dz. \tag{3.57}$$

Here, the correction to the wave vector depends on one's local position in the horizontal plane due to the large-scale structure of the internal waves. It is usual that the lowest gravitational modes give the main contribution to the displacement of the water layer in internal shallow water waves, so in simplified form $\zeta(\vec{r},z) = \zeta(\vec{r})\Phi(z)$, where $\Phi(z)$ is the first gravitational mode normalized to $\max\Phi = 1$. So our correction to the wave vector can be written as

$$\delta q_l = -\frac{k_0^2}{q_l^0}\int_0^H \tilde{N}^2(z)[\Psi_l^0(z)]^2\Phi(z)dz\zeta(\vec{r}). \tag{3.58}$$

In Fig. 3.7, we show relative corrections to eigenvalues $\left|\delta q_l/q_l^0\right|$ as a function of frequency for several of the lowest acoustic modes for waveguide parameters corresponding to the Atlantic shelf of the US (the area of the SWARM'95 experiment (Apel et al. 1997). The amplitude of the internal waves is taken to be ~10 m. We note that there is rather complex behavior to these frequency dependencies. Due to the frequency dependence of the waveguide modes $\Psi_l^0(z)$, there is different overlap versus frequency of the modes with the narrow thermocline layer where the function $N(z)$ is not small.

For perturbation theory to be applicable, the corrections calculated from (3.56) must be much less than the characteristic distances between eigenvalues: $|\delta q_l, \delta q_{l+1}| \ll \left|q_l^0 - q_{l+1}^0\right|$. This condition determines the order of magnitude of the quantity $\delta\tilde{k}^2$ which we shall take as a perturbation, or equivalently gives bounds for the other parameters of the problem. Of course, this estimate is different in each specific case. Let us consider, for example, the case when the Pekeris model is

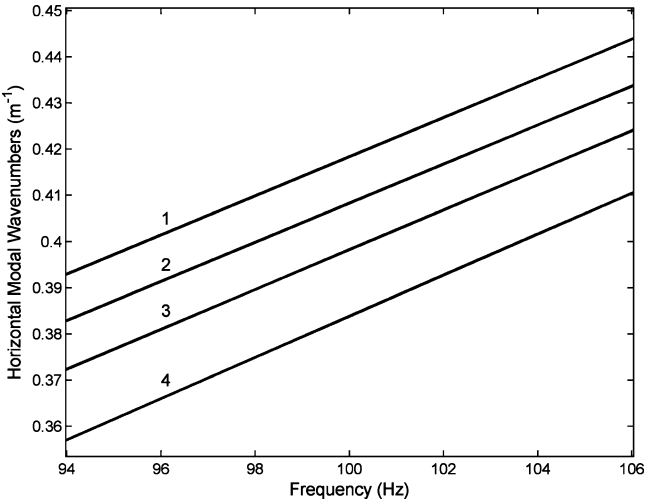


Fig. 3.8 Frequency dependence of unperturbed eigenvalues (modal wave numbers) for mentioned model as a function of frequency

adopted for the unperturbed waveguide and the perturbation is the addition of δc_1 to the sound speed in the seabed, taking place at depths from H to $H + \delta H$. If we suppose that the minimum distance between eigenvalues in the Pekeris model is estimated by $\min|\xi_{l+1} - \xi_l| \approx \frac{\pi^2}{kH^2}$, then the condition for the applicability of perturbation theory has the form

$$\frac{\pi^2}{kH^2} \gg \frac{\delta c_1}{c_1} \frac{1 - \exp(-2\delta H k\sqrt{1 - {n_1}^2})}{m_1 H\sqrt{n_1^{-2} - 1}}. \tag{3.59}$$

The possibility of using perturbation theory thus clearly depends on many parameters of the problem and in particular on the sound frequency. For example, in the case studied by Rajan et al. (1987), where $H = 40$ m, $c = 1{,}500$ m/s, $c_1 = 1{,}900$ m/s, $\delta c_1 = 100$ m/s, $\delta H = 40$ m, and $m_1 = 1.6$, the above estimate gives a condition for applicability of perturbation theory for frequencies $f \ll 1,800$ Hz. By fixing the frequency, one can also obtain a condition for the applicability of other parameters.

As in the example of internal waves as perturbation, the distance between the corresponding eigenvalues can also be seen from Fig. 3.8. We see that typical distances are $\left|q_l^0 - q_{l+1}^0\right| \sim 0.01\,\mathrm{m}^{-1}$, as well as that $|\delta q_l| < 10^{-3}\mathrm{m}^{-1}$.

Equation (3.56), as in the work of Frisk (1988) and Rajan et al. (1987), can be viewed differently, if we suppose that the left-hand side of that equation, that is, the spectrum of the problem, is known via measurements, while the perturbation of the sound-speed profile, that is, the function $\delta k^2(z)$ in the integrand, is unknown. In this case, (3.56) emerges as an integral equation for the unknown function $\delta k^2(z)$, or more precisely, as a system of integral equations. Thus, given some initial model of

the waveguide considered to be unperturbed and the measured spectrum of the real waveguide, by solving the system of integral equations one can find the correction $\delta k^2(z)$, which then gives the "true" sound-speed profile. That is, one can solve the inverse problem in that formulation. Of course, such a solution is only possible if the original parameters of the problem (and, of course, it is not just the sound speed which may vary) are sufficiently close to the final ones. Thus, for successful application of this algorithm, the initial model of the waveguide must be "a good guess." Several examples of recovery of unknown functions $c(z), \rho(z)$ of this type were analyzed in Rajan et al. (1987) and Lynch et al. (1996). For different calculations, the perturbation theory can also be used to find corrections for small variations of the sound-field parameters, for example, frequency (Jensen et al. 1994). In the same vein, one may use a perturbation of frequency to find a useful, integral expression for group velocity.

Another approximate method to construct eigenfunctions and eigenvalues is the well-known WKB method (Brekhovskikh and Lysanov 1991). This method is applicable if the dependence of $k(z)$ (or $c(z)$) is sufficiently smooth – more exactly, if the wavelength of the field is much less than the typical vertical space scale of the sound-speed profile. Obviously, the accuracy of this method increases with the frequency or with the increasing scale of variability of the sound-speed profile. For this reason, the WKB method is far more widely used in deep water acoustics; however, in shallow water, it can also be used for estimation and interpretation of results. Within the framework of this method, we can solve (3.7) for real eigenvalues (without attenuation):

$$\frac{d^2\Psi_l(z)}{dz^2} + \left[k^2(z) - q_l^2\right]\Psi_l(z) = 0, \tag{3.60a}$$

$$\Psi_l(z)|_{z=0} = 0, \tag{3.60b}$$

$$\left[\Psi_l(z) + g(q_l)\frac{d\Psi_l(z)}{dz}\right]_{z=H} = 0. \tag{3.60c}$$

Imaginary part corrections can be obtained on the basis of the real solutions.

According to the WKB method, in order to construct normal modes, we should first find two linearly independent solutions of (3.60a). These solutions can have an oscillatory or monotonic (exponentially decreasing or increasing) character depending on the sign of the expression $q - k(z)$ and depending on whether they are upper or lower turning points. More exactly, oscillating solutions have the form

$$\Psi_l^{\pm} \sim \frac{1}{\sqrt{\sigma_l(z)}} e^{\pm i\int \sigma_l dz}, \tag{3.61}$$

where $\sigma_l(z) = \sqrt{k^2(z) - q_l^2}$. These solutions are found in the region where $q_l < k(z)$ (or $c_l^{\mathrm{ph}} > c(z)$). Points where $q_l = k(z)$ (or $c_l^{\mathrm{ph}} = c(z)$) are called turning points, and

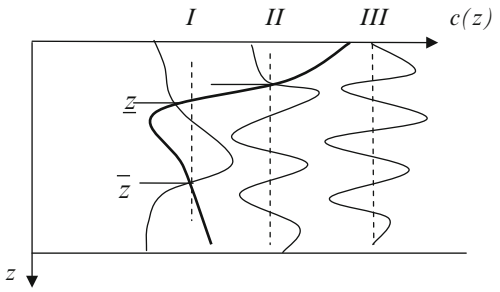


Fig. 3.9 Three types of WKB modes (thin lines for sound speed profile, shown by heavy line): with both turning points within the channel (I) oscillating function between the turning points; lower turn point at the bottom (II); and oscillating solution between the surface and bottom (III). *Dotted lines* denote position of phase velocities of the corresponding modes. Character of modal behavior is shown for case II

they will be denoted as $\bar{z}$ or $\underline{z}$. Their exact positions depend on the type of acoustic channel. In shadow zones where $q_l > k(z)$, solutions have an exponentially decreasing or increasing form:

$$\Psi_l^{\pm} \sim \frac{1}{\sqrt{|\sigma_l(z)|}} e^{\pm \int |\sigma_l(z)| dz}. \tag{3.62}$$

To construct waveguide modes on the basis of the solutions (3.61) and (3.62), we should first join together the solutions (the oscillating and monotonic functions) at the turning points. This is actually a rather interesting exercise, as both solution types are formally inapplicable at the turning points. There are different ways to solve this problem (Brekhovskikh 1980; Kravtsov and Orlov 1990), for example, by using Airy functions solutions and by joining solutions (3.61) and (3.62).

Eigenvalues can be found by using the periodicity conditions for the phase integral $\int \sigma_l(z) dz$. This "Bohr quantization condition" depends on the position and type of the turning points and also the boundary conditions. In actual conditions in shallow water, we occasionally will meet a fully internal waveguide (where both turning points are inside the water column, like the case shown in Fig. 3.9). For more typical conditions with sound-speed profile having the form $c(z)$ as shown in Fig. 3.9 case II, the lower turning point is at the bottom. In this case, the phase integral has the form

$$\int_z^H \sigma_l(z) dz = \pi l + \phi_l, \tag{3.63}$$

where the added phase ϕ_l is defined by the bottom type. Let us next write out waveguide modes for case III in Fig. 3.9, where the mode has an oscillating character in all the space of the waveguide

$$\Psi_l(z) = \frac{N_l}{\sqrt{\sigma_l(z)}} \sin\left[\int_0^z \sigma_l(z)dz\right]. \tag{3.64}$$

Boundary conditions give

$$\tan\left[\int_0^H \sigma_l(z)dz\right] + g(q_l) = 0 \tag{3.65}$$

or

$$\int_0^H \sigma_l(z)dz = \pi l + \arctan(g(q_l)). \tag{3.66}$$

For this case, the phase correction $\phi_l = \arctan(g(q_l))$. We note that for a soft bottom $\phi_l = 0$, and for a rigid bottom $\phi_l = \frac{\pi}{2}$.

The normalization coefficient can also be found in the ordinary way. If the upper turning point is not at the surface, we should add an expression for the eigenfunction (3.64) with an exponentially decreasing part, and glue this part together with the oscillating part. This can be done, for example, using Airy functions. Expressions for the ϕ_l can depend on the eigenvalue and the quantization condition, and often will be more complex than (3.65).

3.4 Ray Description of the Sound Field and Ray-Mode Connections

3.4.1 Ray Theory

The ray description of the sound field is used mostly for the description of the sound field in deep water or for high frequencies. However, for some purposes it can be used in low- and mid-frequency shallow water acoustics.

In ray theory, the sound field from a point source (Green's function) is found using the form

$$\Psi(\vec{r}, z) = A(\vec{r}, z)e^{i\theta(\vec{r},z)}, \tag{3.67}$$

where A is the amplitude, and θ is the phase (eikonal). Curves orthogonal to the iso-phase surfaces are called rays. Both the amplitude and eikonal are functions of the spatial

coordinates (time dependence is taken to be harmonic). The corresponding eikonal equation and transport equation have the well-known forms:

$$\begin{aligned}(\nabla\theta)^2 &= k^2(\vec{r}, z),\\ 2\nabla\theta\nabla A + A\Delta\theta &= 0.\end{aligned} \tag{3.68}$$

Let us consider ray expressions for the sound field in shallow water. For simplicity, we suppose that our waveguide is regular (a model corresponding to Fig. 2.1) with a constant depth and sound-speed profile, depending on depth ($k^2 = k^2(z)$). We will work in cylindrical coordinates r, ϑ, z (Fig. 3.1). Given cylindrical symmetry, the variable ϑ disappears and the sound field can be presented in the form

$$\Psi(r, z) = \sum_n A_n(r, z) e^{i[\theta_n(r,z)]}. \tag{3.69}$$

Summation over the subindex n is added to take into account that we can have several rays, radiating from the source and arriving at the receiver. In the following descriptions, we will omit this subscript when it is possible.

As was mentioned previously, rays are curves normal to constant surfaces of the eikonal; in the case of cylindrical symmetry, these trajectories are constrained to the "vertical" plane containing the source and receiver. In other words, the trajectories are described by two cylindrical coordinates r, z: $\vec{R} = \vec{R}(r, z)$. It is possible to write out equations for the ray trajectory, or more exactly for the unit vector tangent to its trajectory: $\vec{\tau}(r, z) = \nabla\theta(r, z)/|\theta(r, z)|$:

$$\frac{d\vec{R}}{ds} = \vec{\tau}, \quad \frac{d\vec{\tau}(r, z)}{ds} = \frac{1}{k(z)}\nabla_\perp k(z), \tag{3.70}$$

where s is the length of the curve along the ray ($ds = \sqrt{dr^2 + dz^2}$) and $\nabla_\perp = \nabla - \vec{\tau}(\vec{\tau}\nabla)$ is the gradient transverse to the ray trajectory. More comfortable for use are equations for the ray's trajectory, given by the expressions $r = r(s)$, $z = z(s)$:

$$\frac{d^2r}{ds^2} = 0, \; \frac{d}{ds}\left[k(z)\frac{dz}{ds}\right] = \frac{dk(z)}{dz}. \tag{3.71}$$

The parameter s can be considered as one of the ray coordinates; for example, we can consider coordinates s, ϑ, χ_0. Let us find an equation for rays radiated from a source placed at the point $r = 0$, $z = z_0$, where χ_0 is the grazing angle of the ray outgoing from the source.

Using simple evaluations, we can get an equation for the function $r = r(z)$:

$$\left(\frac{dr}{ds}\right)^2 = \frac{k_0^2\cos^2\chi_0}{k^2(z) - k_0^2\cos^2\chi_0}. \tag{3.72}$$

In this case, we find the trajectory of the rays and the eikonal in the forms

$$r(z) = ND \pm \int_{z_0}^{z} \frac{k^2 dz}{\sqrt{k^2 - k_0^2 \cos^2 \chi_0}}, \tag{3.73}$$

$$\theta(r, z) = k_0 r \cos \chi_0 \pm \int_{z_0}^{z} \sqrt{k^2 - k_0^2 \cos^2 \chi_0} dz, \tag{3.74}$$

where N is number of ray cycles, denoted as D. The sign before the integrals is chosen depending on which is greater, z_0 or z. The amplitude of each ray can be found as a divergence, taking into account cylindrical spreading:

$$A = \frac{A_0}{\sqrt{J}}, \quad J = r \left| \frac{\partial(r, z, \vartheta)}{\partial(s, \vartheta, \chi_0)} \right|. \tag{3.75}$$

The expressions we have previously derived or quoted give descriptions of the acoustic field for ocean acoustics, with the ray picture being the easiest to envision and well adapted to deep water acoustics and the mode picture being a full-wave (nonapproximate) solution to the Helmholtz equation that is well tailored to shallow water acoustics. That there is some connectivity between these two solutions to the wave equation is obvious, since they both produce acoustic fields that are approximately the same, especially at higher frequencies. Moreover, mode descriptions invariably talk about the "equivalent ray angle" (the Brillouin angle of a given mode), and about modes being the interference of the up-and-down going plane waves at that angle (rays in an isovelocity medium). But this is just the beginning of the connection. Over the past decades, a number of researchers have explored the deeper connections between rays and modes, most notably L. Felsen and C. Tindle (Tindle and Guthrie, 1974; Guthrie and Tindle 1976; Felsen and Marcuvitz 1994). Felsen's treatment is very deep and mathematical, whereas Tindle's is much simpler and more physical. In keeping with the tone of this book, we will follow the more transparent work of Tindle, and also recommend that the reader interested in greater mathematical depth look into the work of Felsen.

In presenting the next section, we will look at three topics (1) rays as interfering modes, (2) modes as interfering rays, and (3) how to distinguish ray arrivals from mode arrivals. For the first two topics, we will follow Tindle, whereas for the latter we will follow the work by Munk and Wunsch (1983).

3.4.2 *Rays as Interfering Modes*

Let us consider a shallow water waveguide with a typical sound-speed profile (see Fig. 2.1). The sound field from a point source at a fixed frequency (the Green function) is from (3.6)

$$\Psi(r,z) = \frac{i}{4}\sum_{l}\Psi_l(z_0)\Psi_l(z)H_0^{(1)}(\xi_l r). \tag{3.76}$$

In order to go to the ray description of the sound field, we should remember that ray language operates with the concept of a ray phase front, so we should analyze the phase for the expression (3.76). In making the connection between rays and modes, we start with the WKB representation of the solution for a normal mode, i.e.,

$$\Psi_e(z) \sim \frac{1}{\sigma_l^{1/2}}\sin\left[\int_{z_l}^{z}\sigma_l dz\right] \tag{3.77}$$

for the area below the turning point $z_l<z<H$, where z_l is turning point for the lth mode. We have the usual relation between the horizontal, vertical, and total wave numbers

$$\sigma_l^2(z) = \frac{\omega^2}{c(z)^2} - q_l^2. \tag{3.78}$$

The WKB solutions give a good, higher frequency approximation to the normal mode away from the turning points, where the vertical wave number goes to zero and the amplitude of the solution blows up. However, we will be interested only in the phase of the WKB solution, which does not suffer this problem, so we can take that (3.78) holds everywhere.

Using the WKB forms (3.77) for the modes $\Psi_l(z_0)$ and $\Psi_l(z)$, and the asymptotic form of the Hankel function in (3.76), we can write the phase for each lth component in (3.76) as

$$\phi_l = \pm\int_{z_0}^{z}\left(\frac{\omega^2}{c^2} - q_l^2\right)^{1/2} dz + q_l r. \tag{3.79}$$

In addition to the +/− induced by the square root, there are also two more possibilities for the phase in (3.79), corresponding to the integral from z_0 to z going directly or via the upper turning point. These four possibilities correspond to the usual eigenray arrival group (i.e., up–down, down–up, up–up, and down–down angles at source and receiver).

In order to transition to the ray description, we suppose that the frequency is rather high and that we have a large number of waveguide modes with close values of wave numbers. In this case, the interference pattern of the sound field in the waveguide is formed by interfering terms in the sum (3.76) with close values of their phases. Let's take some mode number l and consider the structure of the sound field formed by a group of modes with numbers from l to $l + \Delta l$. The variation of the phase corresponding to a variation of mode number is $\Delta\phi_l = \phi_{l+\Delta l} - \phi_l$. This corresponds to a variation of wave number $\Delta q_l = q_{l+\Delta l} - q_l$, and

$$\Delta\sigma_l = -(q_l/\sigma_l)\Delta q_l \tag{3.80}$$

The key to the whole business is in the next step, when one realizes that successive mode sum terms will interfere constructively provided the change in phase $\Delta\phi$ with change in mode number Δl is some integer multiple of 2π. Thus, the constructive interference condition is

$$\Delta\phi_l = \pm q_l \Delta q_l \int_{z_s}^{z} \frac{1}{\sigma_l} dz + \Delta q_l r = 2N\pi, \tag{3.81}$$

where N is some integer number. But this strong reinforcement will occur only over a small range of mode numbers since ϕ_l varies rapidly with l. This range increases as frequency increases, and the modes move "closer together" in wave number.

Thus, we see that a small group of modes will interfere constructively at a depth and range given by

$$r = \pm \int_{z_0}^{z} \left[\frac{(\omega/q_l)^2}{c^2(z)} - 1 \right]^{-1/2} dz + N \frac{2\pi}{\Delta q_l}. \tag{3.82}$$

This equation defines a line with coordinates (r, z) where we have constructive interference or from another perspective – the equation of a ray. The second term on the right-hand side of (3.82) is the well-known "mode cycle distance," whereas the first term on the right-hand side is the "additional distance" term which accounts for the part of the path in addition to (or subtraction from) an integer number of cycle distances.

This picture of a "ray" created from a constructively interfering group of modes is interesting in that the rays are not sharp, line-like geometrical rays, but rather "fuzzy rays" which have a finite width (see Fig. 3.11). These fuzzy rays may be interpreted as beams which represent the rays at a finite frequency. A discussion of just how many modes one should use to create a given fuzzy ray is given, in terms of a "resonant mode sum," in Guthrie and Tindle (1976). Creation of fuzzy rays is a rather simple academic exercise to try if one looks at the eigenrays and modes of an isovelocity, rigid bottom waveguide, and we leave this to the reader to experiment with. In terms of practical use,

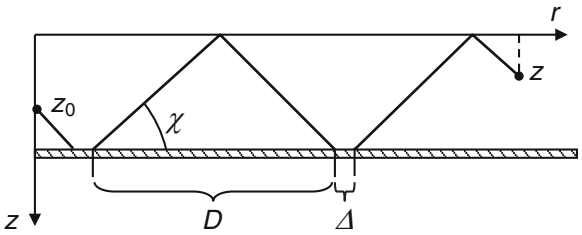


Fig. 3.10 Rays in the Pekeris waveguide, taking into account displacement of rays by bottom reflection

one may use the fuzzy rays in ray tracing much like one uses the Gaussian beam approach (Jensen et al. 1994), though this is not a common use.

3.4.3 Modes as Interfering Rays

Taking a direct cue from the argument above, where constructively interfering neighboring modes conspired to produce a ray (albeit a fuzzy one), we can flip the coin and see if one may take constructively interfering, neighboring eigenrays and somehow produce a mode. Not surprisingly, one can. Again, we follow the lead of Tindle and his collaborators (Tindle and Bold 1981) using the example of a Pekeris waveguide of depth H and sound speeds c and c_1. Figure 3.10 shows the geometry we will consider.

Let us consider (3.69) giving the expression for the sound field as a sum of several rays outgoing from the source at depth z_0 and propagating to the receiver at distance r and depth z. The phase of these rays at the receiver is θ_{Nm} where the numbering of rays can be chosen in different ways. We will label these rays, using the number of reflections N of rays from the bottom (in other words, N is the number of full ray cycles at the distance from the source to the receiver) and m, which denotes an index going from 1 to 4 which accounts for the four types of eigenrays in an arrival group, as before. This phase can be simply expressed as

$$\theta_{Nm} = k\frac{2HN + \tilde{z}}{\sin\chi} + \Delta k \cos\chi N + N\varphi + (N+m)(-\pi). \tag{3.83}$$

In (3.83), the terms correspond to (1) the phase accumulation for the total ray trajectory including additions due to the depths of the source and receiver, for example, for the case of our figure, $\tilde{z} = z_0 + z$; (2) the ray beam displacement upon reflection, Δ (see Brekhovskikh 1980); (3) the phase changes $N\varphi$ due to N bottom reflections (φ is the phase of the reflection coefficient of the bottom); and (4) the

phase change $(N+m)(-\pi)$ due to the surface reflections. Note that m corresponds to u.c. directly, only with values of 1, 0, -1 to address correctly the surface reflection phase changes. Equation (3.83) can be modified using the equality $1/\sin\chi = \cos\chi/\tan\chi + \sin\chi$, so we have

$$\theta_{Nm} = k\cos\chi\frac{2HN+\tilde{z}}{\tan\chi} + N\Delta k\cos\chi + N\varphi + (N+m)(-\pi) + k(2HN+\tilde{z})\sin\chi. \tag{3.84}$$

Taking into account that

$$\frac{2HN+\tilde{z}}{\tan\chi} + N\Delta = r, \tag{3.85}$$

where r is the distance from the source to receiver, we can write an expression for the phase

$$\theta_{Nm} = kr\cos\chi + N\varphi + (N+m)(-\pi) + k(2HN+\tilde{z})\sin\chi. \tag{3.86}$$

If we denote $q = k\cos\chi$ and $\sigma = k\sin\chi$ (the longitudinal and transverse wave numbers), then the phase is

$$\theta_{Nm} = qr + N\varphi + (N+m)(-\pi) + \sigma(2HN+\tilde{z}), \tag{3.87}$$

where the ray displacement is

$$\Delta = -\frac{\partial\varphi}{\partial q}. \tag{3.88}$$

The idea behind the construction of modal structure from rays is that at the receiver, we have constructive interference if the rays are coming in one and the same phase (or differing by $2\pi l$, where l is an integer number). For example, if we have rays with the numbers N and $N+1$, the difference in phase should be $\delta\theta_{Nm} = 2\pi l$. We can construct this difference working with the value N like a continuous variable. It means that we consider

$$\delta\theta_{Nm} = \frac{\partial\theta_{Nm}}{\partial N}\times 1. \tag{3.89}$$

Simple calculations give

$$\delta\theta_{Nm} = 2H\sigma + \varphi - \pi + \frac{\partial q}{\partial N}\left[r + N\frac{\partial\varphi}{\partial q} + (2HN+\tilde{z})\frac{\partial\sigma}{\partial q}\right]. \tag{3.90}$$

If we take into account that $\partial\sigma/\partial q = -q/\sigma$, then the term in square brackets becomes zero. And thus we have

$$2H\sigma + \varphi - \pi = 2l\pi. \tag{3.91}$$

Using the well-known expression for the phase of the reflection coefficient in the Pekeris model,

$$\varphi = -2\arctan(\rho\sigma_1/\rho_1\sigma), \tag{3.92}$$

where the indices 1, 2 refer to the water column and bottom, respectively, we get

$$\sigma H - \arctan(\rho\sigma_1/\rho_1\sigma) = (l + 1/2)\pi. \tag{3.93}$$

Or in another form,

$$\tan(\sigma H) = -m_1\sigma/\sigma_1. \tag{3.94}$$

This equation, which is the condition for neighboring eigenrays to interfere constructively, is also immediately recognized as the eigenvalue equation for the normal modes in the Pekeris model. Thus, the constructive interference of neighboring eigenrays (including the beam displacement, an important point) produces the normal mode equation, saying that these solutions are consistent with it! We note that this does not create a mode, per se, but just informs us that the ray picture, including beam displacement, can be equivalent to the mode picture in certain cases. One formally can create a mode by looking at the constructive interference of up- and down-going rays at the equivalent mode angle, as mentioned before.

3.4.4 Distinguishing Between Ray and Mode Arrivals

Although it is perhaps not a well-posed question to ask whether an arrival one sees on a record is a ray or a mode (since, as we have seen, you can often use either description to a good approximation), it does make some sense to ask whether or not one can resolve a ray or mode arrival using either arrival time discrimination (i.e., the time dispersion of broadband pulse arrivals) or with hydrophone arrays (i.e., spatial wave number processing). The former topic was addressed in depth in a paper by Munk and Wunsch (1983) in the context of the deep water ocean acoustic tomography problem, and their results are easily transportable to shallow water acoustics. The latter problem, array methods, is discussed in our later chapter on signal processing. We would note that, to our knowledge, there has not been a paper discussing the combination of travel time and spatial discrimination of ray versus mode arrivals, which we think would be both straightforward and useful.

In Munk and Wunsch's paper, a great deal of effort is expended in finding analytic expressions for the travel times of rays and modes, particularly for cases that are of interest to deep-water tomography. This effort, while useful pedagogically (like some of the effort we expend on the Pekeris model in shallow water), can be duplicated by numerical calculations, and so we will take the position here that any ray or mode travel times between a source and a receiver are easily calculable. At this point, we can begin looking at where rays and modes are resolvable.

The simplest case to consider is where rays are and are not resolvable. One advantage here is that the criterion, like ray theory, should be frequency independent. If we look at a broadband pulse with a bandwidth Δf, and consider that the duration of pulse $\tau \sim 1/\Delta f$, we get the simple criterion that

$$\Delta_n t \sim 1/\Delta f, \tag{3.95}$$

where $\Delta_n t$ is the travel time difference between the n-th and $(n + 1)$-th ray arrivals. In theory, rays fall into a simple resolved/unresolved classification scheme, dependent only on the bandwidth of the equipment used.

The situation for modes, due to the frequency dependence and dispersion of their travel times, is somewhat different. In this case, the time bandwidth product – TBW $= 1$ criterion leads to the relation

$$\Delta_l t \times \Delta_l f \sim 1 \tag{3.96}$$

being an absolute criterion for whether one can resolve modes using travel time regardless of the bandwidth employed. In this equation, $\Delta_l t$ is the travel time difference between adjacent modes at a given frequency and $\Delta_l f$ is the frequency difference between adjacent modes for a given travel time. For this product less than one, the modes are unresolvable. For this product greater than one, it again depends upon experimental bandwidth whether or not one can resolve modes. Specifically, if one is in the "theoretically resolvable" regime, the criterion

$$\Delta_l t \sim 1/\Delta f \tag{3.97}$$

becomes the limiting equation, very similar to the ray criterion.

We can illustrate the temporal resolvability of modes and rays using the Pekeris model of the shallow water waveguide. In this case, the arrival times of separate modes is determined by $t_l = r/c_l^{gr}$, where the group velocity can be estimated as $c_l^{gr} \sim cq_l/k$ (3.47). In Fig. 3.11, we show a set of curves for the frequency dependence of arrival times for the 10 lowest modes. If we establish a resolvable time $\tau \sim 0.1$ s, then the area of resolvability of modes ($\Delta_l t > 0.1$ s) is placed above the curve shown in Fig. 3.11. Rays in the Pekeris waveguide can be constructed very simply, on the basis of straight lines, outgoing from the source and its images above the surface and below the bottom. They are characterized by the grazing

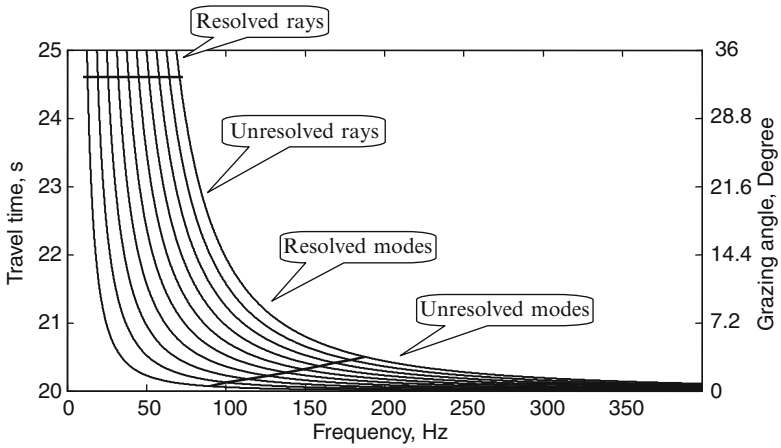


Fig. 3.11 Areas of resolvability of rays and modes for the Pekeris waveguide. Depth $H = 200$ m, distance to the receiver $r = 30$ km. Curves for arrival time as a function of frequency are shown for 10 modes. Right vertical axis shows grazing angles of the rays, having arrival time, shown in the left vertical axis

angle χ_n between the direction of the nth ray and the bottom. The arrival time of the nth ray is $t_n \sim r/(c\cos\chi_n)$, and correspondingly,

$$\Delta t_n \sim \left[r\sin\chi_n/(c\cos^2\chi_n)\right]\Delta\chi_n, \tag{3.98}$$

where $\Delta\chi_n$ is the angle between the grazing angles for the nth and $n + 1$th rays. It can be simply estimated from geometrical considerations as

$$\Delta\chi_n \sim H\cos\chi_n/r \tag{3.99}$$

so that

$$\Delta t_n \sim H\tan\chi_n/c. \tag{3.100}$$

So the condition for the resolvability of rays is $H\tan\chi_n/c > 0.1$ s.

We see that the resolvability of rays depends directly on the depth H (within the framework of our model). The smaller the depth, the worse the resolvability, or in other words, we can most easily resolve rays for the larger grazing angles. (We note that in shallow water, due to attenuation, we only deal with comparatively small grazing angles.) For example, for a depth of 200 m, we are only able to resolve rays having grazing angles of more than ~32°. In our diagram (Fig. 3.11), this area is placed above the straight line, which is parallel to the frequency axis. On the right side of the figure, we draw a vertical axis, denoting the grazing angles of the rays.

In the paper (Kravtsov et al. 1988), the problem of the resolvability of rays/modes using horizontal and vertical arrays was also discussed. An example of the Pekeris

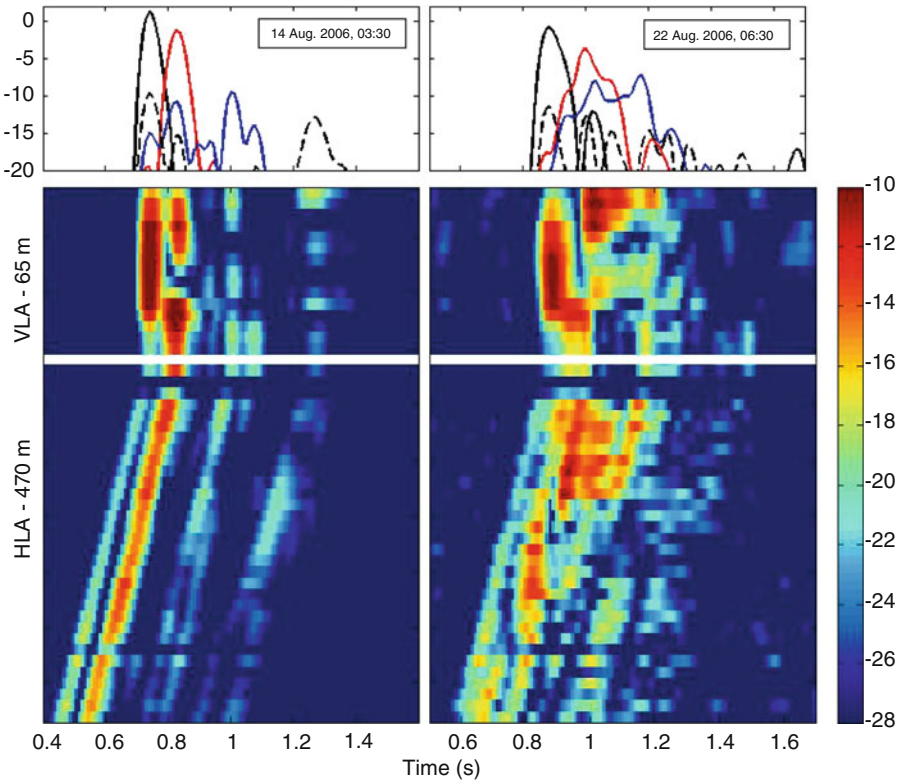


Fig. 3.12 An example from the SW06 experiment of modal arrivals seen on a combination vertical/horizontal line array. The *top panels*, which deal with array mode filtering results, can be disregarded here. The *middle panels* show the arrival time structure on the vertical array. The *left middle panel* shows a very clear travel time separation of modes 1–4, whereas the *right middle panel* does not show time separation of the modes past the first due to the time spreading due to scattering. The *bottom panel* shows the arrivals on a bottom lying horizontal array, with the same general modal separation in time as the vertical array, but with no simple way (solely from the horizontal display) to determine which mode is which

waveguide was examined, showing that depending on the distance to the source, we can resolve either rays or modes. A "critical" distance to separate these zones is estimated to be $r_* \sim H^2/\lambda$ and these are called near ($r \ll r_*$) and far ($r \gg r_*$) zones.

Moreover, in shallow water, vertical and horizontal arrays are often used in conjunction with broadband signals so that the combination of array and travel time resolution needs to be addressed (see Fig. 3.12). It would seem that there is more work one could do on this "limits to resolution" problem, incorporating spreading and arrays into a more generalized ambiguity diagram, and perhaps someone will follow this path in future work.

3.5 Mode Coupling in a Shallow Water Waveguide with Small Inhomogeneities

In the previous sections, we considered the sound field in the waveguide assuming the sound-speed profile, bottom properties and water depth to be constant along an acoustic track. This waveguide is denoted as a "regular" range independent one. In this waveguide, the sound field can be expressed as an expansion (series) of waveguide modes, which are solutions of a Sturm–Liouville equation. The weights or amplitudes of the modes, denoted in (3.16) as C_l, either do not depend on the range r in the absence of attenuation, or depend on r exponentially if there is attenuation in the waveguide. Irrespective of attenuation, in a regular waveguide modes propagate independently on each other (which is their main feature). The number of modes remains the same during propagation. In the presence of a perturbation of any nature, destroying the regular character of the waveguide (i.e., giving range dependence), this situation does not take place. However, if these perturbations are small (in some sense), then the representation of the sound field as a sum of waveguide modes can still be maintained. That is, the modes themselves and/or their amplitudes change smoothly (or change only a little) during propagation of the signal in the waveguide. Variations of the mode amplitudes are interpreted as transferring energy from one mode to another, and we thus discuss "mode coupling" or "mode interaction." The exact nature and description of this interaction depend on the type of perturbation.

Some of the types of perturbations regularly encountered are as follows:

1. Small and comparatively sharp volume inhomogeneities and/or small roughness of the bottom and surface of the sea (see Fig. 3.13).
2. Smooth variations of the waveguide parameters along an acoustical track (see Fig. 3.14).
3. Both types of perturbations taking place – small (possibly sharp) inhomogeneities on a background of smooth (and possibly not small) variations of the waveguide parameters. This classification corresponds to a separation of the perturbation into large and small scale, which will prove useful.

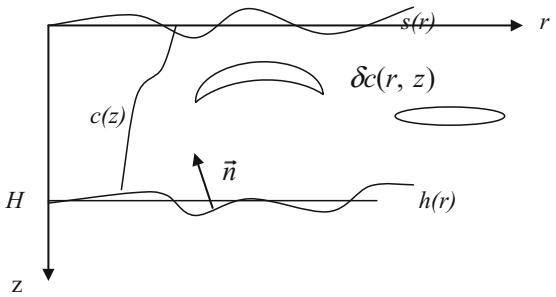


Fig. 3.13 Model of shallow water with small inhomogeneities

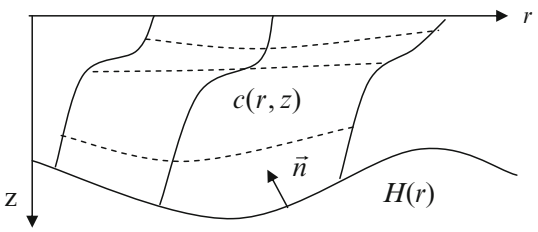


Fig. 3.14 Shallow water model with smooth inhomogeneities. Isolines of sound speed are shown by dotted lines

Mathematically, taking mode coupling into account corresponds to the solution of a system of coupled linear equations of the second order for the modal amplitudes. This system, for a smooth change of amplitude and in the forward scattering approximation, can also be reduced to a system of the first order. The coefficients of mode coupling are determined by the character of the perturbation; in particular, they should be real for smooth perturbations (nonadiabatic) and imaginary for small perturbations. The condition of energy conservation leads to the condition of symmetry for the mode coupling coefficients.

Let us look at a simple case, assuming approximate cylindrical symmetry, which implies that all our values depend only on the two variables r, z. Let the bottom have a variable depth $H(r)$, the sound speed in water be $c(r, z)$, the bottom sound speed be c_1, and the densities ρ and ρ_1 be constant. In principle, we should also take into account surface variations due to wind waves, described by a displacement $s(r)$. If we consider the effects of mode coupling for long distances and use cylindrical spreading, we can make the replacement

$$\Psi(r, z) = \frac{1}{\sqrt{r}}\phi(r, z). \tag{3.101}$$

Making this substitution in (3.2), we get

$$\frac{\partial^2\phi}{\partial z^2} + \frac{\partial^2\phi}{\partial r^2} + k^2(r, z)\phi = 0. \tag{3.102}$$

We now consider small perturbations of the waveguide parameters on the background of a regular waveguide. In this case, we have the following waveguide parameters: surface $s(r)$, depth $H + h(r)$, sound-speed profile $c(r, z) = c(z) + \delta c(r, z)$, and correspondingly $k^2(r, z) = k^2(z) + \delta k^2(r, z)$, where $|h(r)|$, $|s(r)| \ll H$, $|\delta c| \ll c$, and $|h'| \ll 1$.

We also have the boundary conditions

$$\phi\big|_{z=s(r)} = 0, \tag{3.103}$$

$$\phi|_{z=H+h-} = \phi|_{z=H+h+}, \tag{3.104}$$

$$\frac{1}{\rho}\frac{\partial\phi}{\partial n}\Big|_{z=H+h-} = \frac{1}{\rho_1}\frac{\partial\phi}{\partial n}\Big|_{z=H+h+} \tag{3.105}$$

where $\vec{n}$ is the normal to the bottom (see Fig. 3.13). Solutions of (3.109) can be found in the form of a series over the eigenfunctions of the regular waveguide $\Psi_n(z)$, or in the region $0<z<H$

$$\phi(r,z) = \sum_n a_n(r)\Psi_n(z), \tag{3.106}$$

where

$$\frac{d^2\Psi_n}{dz^2} + \left[k^2(z) - \xi_n^2\right]\Psi_n = 0 \tag{3.107}$$

with boundary conditions

$$\Psi_n(0) = 0 \tag{3.108}$$

$$\Psi_n^- = \Psi_n^+; \quad \frac{1}{\rho}\frac{\partial\Psi_n^-}{\partial z} = \frac{1}{\rho_1}\frac{\partial\Psi_n^+}{\partial z},$$

where we use the notation $F^{\pm} = F|_{z=H\pm}$.

It should be noted that for the functions Ψ_n and for the reduced sound pressure ϕ, we have different boundary conditions at the bottom – in (3.108) we have a derivative with respect to z, whereas the boundary conditions (3.105) contain derivatives with respect to the normal to the bottom.

The eigenfunctions are normalized as

$$\int_0^\infty \frac{\rho}{\rho(z)}\Psi_n(z)\Psi_m(z)dz = \delta_{nm}. \tag{3.109}$$

After substituting (3.106) into (3.102) and subsequent evaluation (see Appendix C), we get the so-called system of coupled modes for a modal amplitude a_n. If we denote

$$S_{mn} = -\frac{\partial\Psi_n^0}{\partial z}\frac{\partial\Psi_m^0}{\partial z}s, \tag{3.110}$$

$$B_{mn}^{(1)} = h\left\{\left(\frac{1}{\eta} - 1\right)\frac{\partial\Psi_n^-}{\partial z}\frac{\partial\Psi_m^-}{\partial z} + \Psi_m^-\Psi_n^-\left[\eta(\xi_n^2 - k_1^2) - (\xi_n^2 - k_H^2)\right]\right\}$$
$$B_{mn}^{(2)} = h'(\eta - 1)\Psi_m^-\Psi_n^-, \tag{3.111}$$

where

$$\frac{\partial\Psi_n^0}{\partial z} = \frac{\partial\Psi_n}{\partial z}\Big|_{z=0}$$

and

$$V_{nm} = \int_0^\infty \frac{\rho}{\rho(z)} \Psi_m \delta k^2 \Psi_n dz, \tag{3.112}$$

then

$$a''_m + \xi_m^2 a_m = -\sum_n B_{mn}^{(2)} a'_n - \sum_n (V_{mn} + B_{mn}^{(1)} + S_{mn}) a_n. \tag{3.113}$$

Of course, the relationship between the mode coupling coefficients V_{mn}, $B_{mn}^{(1,2)}$, S_{mn} depends on the actual conditions encountered. For example, in the presence of volume inhomogeneities, we have the terms in (3.111) and $B_{mn}^{(1,2)}$ in (3.113) to take into account, e.g., the bottom roughness.

If the right side of (3.113) is equal to zero, we have

$$a_m(r) = \frac{C_m^+}{\sqrt{\xi_m}} e^{iq_m r} + \frac{C_m^-}{\sqrt{\xi_m}} e^{-iq_m r} \tag{3.114}$$

and the sound field is a sum of uncoupled modes. The second term in (3.114), containing the constant C_m^-, describes waves propagating to the source (reflected waves). In the absence of perturbations this term should be equal to zero, but in the presence of perturbations, this term should be taken into account. As a rule, we neglect this term (the forward scattering approximation). Let us write the mode coupling equations for this case. We start with

$$a_m(r) = \frac{C_m(r)}{\sqrt{\xi_m}} e^{iq_m r} \approx \frac{C_m e^{iq_m r}}{\sqrt{q_m}}. \tag{3.115}$$

For this case, the decomposition of the Green function over modes [using (3.101) and (3.106)] has the form (3.16). Let us also suppose that in this case, the variation of the coefficient of modal amplitude $C_m(r)$ is a comparatively slow function of range so that we can separate the slow and fast variability, i.e.,

$$\left|\frac{C'_m}{C_m}\right| \ll q_m, \left|\frac{q'_m}{q_m}\right| \ll q_m, \left|\frac{(C'_m)'}{C'_m}\right| \ll q_m. \tag{3.116}$$

The meaning of this approximation is that we can speak about waveguide modes only at distances greater than the spatial scale of the modal interference, which is $\geq \frac{2\pi}{|q_n - q_m|}$. Only for these distances does it make sense to speak about the modal description of the sound field and mode coupling.

Then we have

$$a'_m = \frac{C'_m}{\sqrt{q_m}} e^{iq_m r} + iq_m \frac{C_m}{\sqrt{q_m}} e^{iq_m r} \approx iq_m \frac{C_m}{\sqrt{q_m}} e^{iq_m r}, \tag{3.117}$$

where we used that

$$\left| \frac{C'_m}{C_m} \right| \ll q_m. \tag{3.118}$$

For the second derivative,

$$a''_m = \left(2i \frac{q_m}{\sqrt{q_m}} C'_m - \frac{q_m^2}{\sqrt{q_m}} C_m \right) e^{iq_m r}. \tag{3.119}$$

Let us now put these expressions into the mode coupling system. In doing this, we get the first-order system

$$C'_m + \frac{\gamma_m}{2} C_m = \sum_n (\tilde{V}_{mn} + \tilde{B}_{mn} + \tilde{S}_{mn}) e^{-i\Delta q_{mn} r} C_n, \tag{3.120}$$

where

$$\tilde{S}_{mn} = -\frac{1}{2\sqrt{q_n q_m}} \frac{\partial \Psi^0_m}{\partial z} \frac{\partial \Psi^0_n}{\partial z} s, \tag{3.121}$$

$$\tilde{V}_{nm} = \frac{1}{2\sqrt{q_n q_m}} \int_0^\infty \frac{\rho}{\rho(z)} \Psi_n \delta k^2 \Psi_m dz\,, \tag{3.122}$$

$$\begin{aligned} \tilde{B}_{nm} = & \frac{h}{2\sqrt{q_n q_m}} \left\{ \left(\frac{1}{\eta} - 1 \right) \frac{\partial \Psi^-_n}{\partial z} \frac{\partial \Psi^-_m}{\partial z} + \Psi^-_n \Psi^-_m \left[\eta(\xi_n^2 - k_1^2) - (\xi_n^2 - k_H^2) \right] \right\} \\ & + \frac{iq_n h'}{2\sqrt{q_n q_m}} (\eta - 1) \Psi^-_n \Psi^-_m \end{aligned} \tag{3.123}$$

and $\Delta q_{ml} = q_m - q_l$.

System (3.120) with the appropriate mode coupling coefficients will be used for the analysis of the influence of small inhomogeneities, both random and deterministic.

3.6 Mode Coupling in Shallow Water with Smooth Inhomogeneities

Let us now consider the case of mode coupling in the situation where the waveguide parameters change smoothly along an acoustic track (or in a more general sense, when the parameters depend smoothly on the horizontal coordinates, as per Fig. 3.14). A more exact criterion of "smoothness" will be discussed later. In terms of absolute value, the variation can be significant and comparable with the value of the parameter itself (for example, $\delta H \sim H$). In a real situation on the shelf, typical values of depth variation can be of the order 100 m over 10 km. In this case, we can also use the modal decomposition of the sound field, where after the excitation of distinct waveguide modes at the source we have a superposition of different waveguide modes at the receiver.

The theory of this type of mode coupling has a different form compared with the example presented above. The sound field is presented as decomposition in a series of the so-called adiabatic or local modes, which are the solution of the range-dependent Sturm–Liouville problem. The equation for the sound field has the form (3.102), where $H(r)$ and $k^2(r,z)$ now cannot be decomposed into a regular part and a small perturbation.

In this case, we solve (3.102) with the boundary conditions

$$\phi|_{z=0} = 0, \tag{3.124}$$

$$\phi\big|_{z=H(r)-} = \phi\big|_{z=H(r)+}, \tag{3.125}$$

$$\frac{1}{\rho}\frac{\partial\phi}{\partial n}\Big|_{z=H(r)-} = \frac{1}{\rho_1}\frac{\partial\phi}{\partial n}\Big|_{z=H(r)+}, \tag{3.126}$$

where $\vec{n}$ is the normal to the seabed (see Fig. 3.14).

In order to construct solutions of (3.102), we introduce the local eigenfunctions of the Sturm–Liouville problem, $\Psi_l(r,z)$, now depending on the horizontal coordinates, i.e.,

$$\frac{d^2\Psi_l}{dz^2} + \left[k^2(r,z) - \xi_l^2(r)\right]\Psi_l = 0 \tag{3.127}$$

with boundary conditions

$$\begin{aligned} \Psi_l(0) &= 0, \\ \Psi_l^- &= \Psi_l^+, \\ \frac{1}{\rho}\frac{\partial\Psi_l^-}{\partial z} &= \frac{1}{\rho_1}\frac{\partial\Psi_l^+}{\partial z}, \end{aligned} \tag{3.128}$$

where we again use the notation $F^{\pm} = F|_{z=H\pm}$.

The functions $\Psi_l(r,z)$ are the solutions of (3.107), (3.108) for every cross-section of the waveguide and are called the adiabatic or local normal modes. Correspondingly, the local modes are normalized for each value of range:

$$\int_0^\infty \frac{\rho}{\rho(z)} \Psi_l(r;z)\Psi_m(r;z)dz = \delta_{lm}. \tag{3.129}$$

Again, the functions Ψ_l and the reduced sound pressure φ have different boundary conditions at the bottom – in (3.128) we have derivatives with respect to z, whereas the boundary condition (3.126) contains derivatives with respect to the normal to the bottom.

Similar to our previous case, we should modify the boundary condition at the bottom, i.e.,

$$\frac{\partial}{\partial n} = n_r \frac{\partial}{\partial r} + n_z \frac{\partial}{\partial z} \approx H' \frac{\partial}{\partial r} + \frac{\partial}{\partial z} \tag{3.130}$$

and correspondingly,

$$\frac{\partial \phi^-}{\partial z} - \eta \frac{\partial \phi^+}{\partial z} = H' \left(\frac{\partial \phi^-}{\partial r} - \eta \frac{\partial \phi^+}{\partial r} \right). \tag{3.131}$$

Similar to our previous case, we can get an integral equation for the reduced sound field $\phi(r,z)$

$$H' \left[\eta \frac{\partial \phi^+}{\partial r} - \frac{\partial \phi^-}{\partial r} \right] \Psi_l^- + \int_0^\infty \frac{\rho}{\rho(z)} [\xi_l^2 \phi + \frac{\partial^2 \phi}{\partial r^2}] \Psi_l(r,z)dz = 0. \tag{3.132}$$

Let us find solutions of (3.132) in the form

$$\phi(r,z) = \sum_l a_l(r)\Psi_l(r;z). \tag{3.133}$$

In this case, from the integral equation (3.132), we can get a system for the values of $a_e(r)$:

$$a_l'' + a_l \xi_l^2 = -\sum_m [B_{lm} a'_m + D_{lm} a_m], \tag{3.134}$$

where

$$D_{lm} = \int_0^\infty \frac{\rho}{\rho(z)} \Psi_l \frac{\partial^2 \Psi_m}{\partial r^2} dz + (\eta - 1) H' \Psi_l^- \frac{\partial \Psi_m^-}{\partial r}, \tag{3.135}$$

$$B_{lm} = 2\int_0^\infty \frac{\rho}{\rho(z)}\Psi_l \frac{\partial \Psi_m}{\partial r} dz + (\eta - 1)H'\Psi_l^- \Psi_m^-. \tag{3.136}$$

As a rule, $|D_{lm}| \ll |B_{lm}|$ and we can neglect the second term on the right-hand side of (3.134).

In the simplest case, where we can neglect mode coupling of any nature, we have zero on the right side of (3.134) and so the WKB solution of the homogeneous equation for a_l has the form

$$a_l(r) = \frac{C_l^+}{\sqrt{\xi_l(r)}} e^{i\int \xi_l dr} + \frac{C_l^-}{\sqrt{\xi_l(r)}} e^{-i\int \xi_l dr}, \tag{3.137}$$

where the $C_l^\pm$ are constant. Physically, the positive and negative signs correspond to waves propagating to the left and right; if we have a point source, these waves correspond to radiated and reflected signals. For this case, we also the mode set at the source remains unchanged in energy during propagation. This approximation is the so-called adiabatic approximation, where the sound field is the sum over modes, which depends locally on the waveguide parameters. Instead of system (3.134), we can also get a system of equations of the first order in the forward scattering approximation. Simply speaking, this approximation supposes that the modal amplitude has the form (3.137) where we neglect the term C_l^- and suppose that C_l^+ depends on range. A more comfortable view of the coefficient results if we let the part of the exponential factor containing the attenuation be included in the coefficient $C_l^+ \exp(-\int \gamma_l/2dr) = C_l(r)$. Next, since $\gamma_l \ll q_l$, we can neglect γ_l in the denominator. So the amplitude a_l has the form

$$a_l(r) = \frac{C_l(r)}{\sqrt{q_l(r)}} e^{i\int q_l(r')dr'}. \tag{3.138}$$

Similar to the previous section, we assume that the modal amplitude $C_l(r)$ is a comparatively slow function of range, so we can separate slow and fast variability, i.e.,

$$\left|\frac{C_l'}{C_l}\right| \ll q_l, \left|\frac{q'_l}{q_l}\right| \ll q_l, \left|\frac{(C'_l)'}{C'_l}\right| \ll q_l. \tag{3.139}$$

Again, the meaning of this approximation is that we can speak about waveguide modes only at distances greater than the modal interference scale; in other words, $r \geq \Lambda_{lm} = 2\pi|q_l - q_m|^{-1}$.

Then we take

$$a'_l = \frac{C'_l}{\sqrt{q_l}} e^{i\int q_l dr} + iq_l \frac{C_l}{\sqrt{q_l}} e^{i\int q_l dr} \approx iq_l \frac{C_l}{\sqrt{q_l}} e^{i\int q_l dr}. \tag{3.140}$$

We used that

$$\left|\frac{C'_l}{C_l}\right| \ll q_l. \tag{3.141}$$

So for the second derivative, we can write

$$a''_l = \left(2i\frac{q_l}{\sqrt{q_l}}C'_l - \frac{q_l^2}{\sqrt{q_l}}C_l\right)e^{i\int q_l dr}. \tag{3.142}$$

Let us combine these expressions while keeping in mind that in this approximation we should neglect the terms D_{mn} in the mode coupling system. We get

$$C'_l + \frac{\gamma_l}{2}C_l = -i\sum_m U_{lm}e^{-i\int \Delta q_{lm}dr}C_m, \tag{3.143}$$

where

$$U_{lm} = \sqrt{\frac{q_m}{q_e}}\int_0^\infty \frac{\rho}{\rho(z)}\Psi_l\frac{\partial\Psi_m}{\partial r}dz. \tag{3.144}$$

The coefficients of coupling U_{lm} describe the nonadiabatic interaction of the modes. Neglecting this mode coupling again leads to the adiabatic approximation, which for this case gives the expression for the sound field:

$$\Psi(r,z) = \frac{1}{\sqrt{r}}\sum_l \frac{C_l^0}{\sqrt{q_l}}\Psi_l(r,z)e^{i\int[q_l(r')+i\frac{\gamma_l(r')}{2}]dr'}. \tag{3.145}$$

For the example of a point source, placed at the point $(0, z_0)$, we have often seen the expression

$$\Psi(r,z) = \sum_l \frac{i}{\sqrt{8\pi q_l r}}\Psi_l(0,z_0)\Psi_l(r,z)e^{i\int\left[q_l(r')+i\frac{\gamma_l(r')}{2}\right]dr'}. \tag{3.146}$$

In concluding this section, we revisit the fact that range variations of the waveguide parameters (sound-speed profile, water depth, and bottom properties) can have two scales of variability – small variations appearing on the background of a smooth or large-scale variation. Mathematically, this corresponds to using the equation $\bar{H}(r) = H(r) + h(r)$ for the range dependence of depth, $c(r,z) + \delta(r,z)$ for the sound speed in the water layer and holding c_1, and the densities ρ and ρ_1 fixed. In other words, we select both smooth and small variations of the depth and sound-speed profile waveguide parameters: $h \ll H$, $\delta c \ll c$. The smallness of

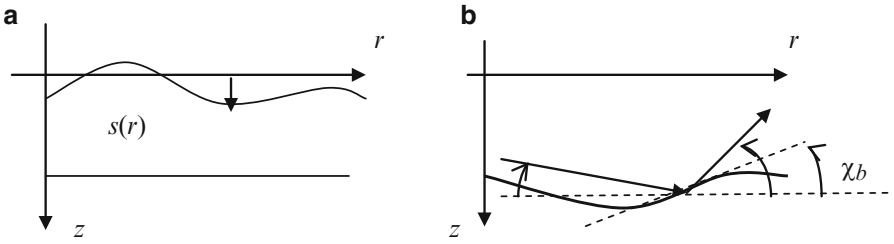


Fig. 3.15 (**a**) shape of rough surface, (**b**) Reflection of rays from an inhomogeneous bottom

derivatives $|d\bar{H}/\mathrm{d}r| = |\bar{H}'|$, $|\partial c/\partial r|$ is also assumed, the meaning of which will be discussed later. In terms of oceanography, this corresponds to considering mesoscale and short-period fluctuations, respectively.

One very important question is that of the applicability of the adiabatic approximation (3.137). To ascertain this, we should estimate the correction provided by nonadiabatic interactions to the adiabatic solution. If we integrate (3.143), we get

$$C_l(r) = C_l(0) + i\sum_m \sqrt{\frac{q_m(r)}{q_l(r)}} C_m(r) \frac{U_{ml}(r)}{\Delta q_{ml}(r)} [\exp(-i\Delta q_{lm}(r)) - 1], \qquad (3.147)$$

where $\Delta q_{ml} = q_m - q_l$.

For (3.146) to be applicable, as can be seen from (3.147), $|U_{ml}|$ must be small in comparison with $|\Delta q_{ml}| = |q_m - q_l|$, i.e.,

$$\tilde{\kappa} \approx \frac{|U_{ml}|}{|q_m - q_l|} \ll 1. \qquad (3.148)$$

The inequality (3.148) can be given in another form, if we take into account that $1/|\xi_{ml}|$ is of the order of the value of the cycle distance in the channel Λ and $U_{ml} \approx \tilde{L}_k^{-1}$, where $\tilde{L}_k$ is the characteristic scale of the variation of the waveguide properties [for example, the scale of the variation of the wave number $k(r)$]. Then, the inequality (3.148) takes the form

$$\tilde{\kappa} \approx \Lambda |U_{ml}| \approx \Lambda/\tilde{L}_k \ll 1. \qquad (3.149)$$

The parameter $\tilde{\kappa}$ is often called the degree of nonadiabacity of the waveguide.

As an example, let us estimate the coupling coefficient U_{ml} resulting from a change in depth due to a surface wave disturbance, within the framework of a very simple model of a channel with a constant sound speed ($k = \text{const}$). The specifics are as follows: assume that the boundaries of our waveguide are a bottom of a constant depth and a surface which depends on both space coordinates and time, $s(r,t) \le z \le H$ (see Fig. 3.15a). The function $s(r,t)$ describing the wave elevation as a function of distance is assumed to be smooth. The dependence on time is also

assumed smooth and slowly changing so that we can consider the problem in the "frozen approximation" and the argument t can be omitted. In this case, the solution of (3.127) with the simplest boundary conditions on the bottom $\partial\Psi_l/\partial z|_H = 0$, corresponding to $g \to \infty$, and $\Psi_l\big|_{z=s(r)} = 0$ on the surface has the form

$$\Psi_l(r;z) = \sqrt{\frac{2}{H-s(r)}}\sin\left[\frac{\left(l+\frac{1}{2}\right)\pi}{H-s(r)}[z-s(r)]\right]. \tag{3.150}$$

Substituting (3.150) in U_{ml}, we find

$$|U_{ml}| = 2\frac{s'}{H}\frac{\left(l+\frac{1}{2}\right)\left(m+\frac{1}{2}\right)}{|m-l|(m+l+1)}. \tag{3.151}$$

Obviously, the coupling will be maximal for adjacent modes with $l = m \pm 1$:

$$\left|U_{m,m\pm1}\right| = 2\left|\frac{s'}{H}\right|\frac{\left(m\pm1+\frac{1}{2}\right)^2}{(2m+1\pm1)} \approx m\left|\frac{s'}{H}\right|. \tag{3.152}$$

The coupling of mode m with distant modes $|m-l| \gg 1$ has the value $\left|U_{m,\infty}\right| \approx 2|s'/H|$. For an interference length scale $|\Delta q_{ml}|$ and for indices m, l which are not too large (this corresponds to a small angle of inclination for equivalent rays), we obtain the estimate

$$|q_m - q_e| \approx \frac{\pi^2}{2H^2k}|m-l|(m+l+1). \tag{3.153}$$

The condition for the applicability of the adiabatic approximation then takes the form

$$\tilde{\kappa} = 2k|s'|H\frac{\left(l+\frac{1}{2}\right)^2}{(m-l)^2(m+l+1)^2} \ll 1. \tag{3.154}$$

Clearly, this quantity has its greatest value for adjacent modes with indices $|m-l| = 1$. In this case, the following inequality must be satisfied

$$\tilde{\kappa} \approx k|s'H/2| \ll 1. \tag{3.155}$$

We note that in the case where we have an irregular channel with a variable depth, $H = H(r)$ (Fig. 3.14), the condition for the applicability of the adiabatic approximation has the analogous form

$$\begin{aligned} \tilde{\kappa} &\approx k|H'H/2| \ll 1 \\ H' &\approx \frac{dH}{dr}. \end{aligned} \tag{3.156}$$

Substituting typical values $f = 300$ Hz, $H = 200$ m, $|s'| \approx 10^{-3}$, $c \approx 1{,}500$ m/s, we obtain the value $\tilde{\kappa} \approx 0.1$. If we consider a general frequency f, then for the same parameters of the channel, $\tilde{\kappa} \approx 4 \times 10^{-4} f$. In other words, for this depth, mode coupling can be neglected up to frequencies $f \leq 10^3$ Hz. For higher frequencies, mode coupling would have to be taken into account.

Considering that H' characterizes the angle of inclination of the bottom χ_b (see Fig. 3.15b) and that kH/π is the number of propagating modes M, the adiabaticity parameter κ for a given model of the channel can be written as $\pi M \chi_b$. Naturally, the greater the number of propagating modes, the closer their phase velocities will be and the stronger the coupling between them.

The same idea can be expressed in a different way. When there are M modes propagating in the channel, the average angle between adjacent "Brillouin rays" is $\approx \pi/M$. When a ray inclined at angle χ_b is reflected from a surface, the direction of the ray (its grazing angle) changes by $\approx \chi_b$. Clearly, mode coupling will be insignificant if the change in the grazing angle $\approx \chi_b$ is less than the angle between Brillouin rays. This condition can be written as $\chi_b \ll \pi/M$, which corresponds approximately to the previous adiabaticity condition.

In the case of a more complicated waveguide model, where not only the depth changes with distance, one has to check whether or not it is possible to use the transverse sections method for a given dependence of the waveguide parameters on the coordinates.

3.7 Horizontal Refraction in Shallow Water (The 3D Problem)

When calculating the sound field in a waveguide, we previously (Sects. 3.1 and 3.2) assumed that the problem had cylindrical symmetry, which enabled us to reduce it to a 2D problem, with r and z coordinates only. In other words, we assumed that in the horizontal plane, the propagation of sound takes place in a straight line in the radial direction. If the parameters of the waveguide vary smoothly and moderately in the radial direction, then the sound field is the sum of independently propagating adiabatic normal modes (or local modes) with coefficients depending smoothly on the longitudinal coordinate r. As far as the 3D picture is concerned, the wave fronts are cylindrical surfaces. But a real waveguide most often does not have exact cylindrical symmetry, and so the sound does not propagate in an exact straight line (in the horizontal plane) and the wave fronts have a different form. The problem thus becomes fully three dimensional (denoted as 3D) and the field at the receiver depends on one vertical coordinate (z) and two horizontal ones ($\vec{r} = (x, y)$). Assuming that the waveguide parameters (depth and sound-speed profile) vary as functions of the horizontal coordinates, the propagation becomes more complex than in the previous sections concerned with mode coupling, since besides mode coupling, the deviation of the propagation direction from a straight line now must also be taken into account.

Let us now obtain the equations describing this fully 3D case. We shall assume that we have the waveguide model described in Sect. 3.1 (water layer on a fluid, absorbing half-space) with parameters now depending on the coordinates x, y, and z. In order to avoid cumbersome calculations, we will assume that we have comparatively smooth variability of the waveguide parameters in the x, y coordinates, and we will also exclude small variations of depth and volume inhomogeneities. Additionally, we will neglect terms in the mode coupling due to deviations of the normal derivative. (These terms could however be included in the theory as in the previous sections).

We first write (3.2) and (3.3) for the Green function in a Cartesian coordinate system, assuming a smooth dependence on x and y

$$\left[\nabla_-^2 + \frac{\partial^2}{\partial z^2} + k^2(\vec{r}, z)\right]\Psi(\vec{r}, z) = 0, \tag{3.157}$$

$$\Psi(\vec{r}, z) \to \frac{\exp(ik_0|\vec{R} - \vec{R}_0|}{4\pi|\vec{R} - \vec{R}_0|} \text{as } \vec{R} \to \vec{R}_0.$$

We also should remember our notation for the gradient in horizontal plane $\nabla_- = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$, and note that $\vec{R} = (x, y, z), \vec{R}_0 = (0, 0, z_0)$, and $k_0 = k(0, 0, z_0)$.

We will now search for a solution of (3.157) in the form of a decomposition into local modes, $\Psi_l(\vec{r}; z) = \Psi_l(x, y; z)$ depending [unlike the functions in (3.114), (3.115)] on the two horizontal components via the dependence of the waveguide parameters on these components, that is,

$$\Psi(\vec{r}, z) = \sum_l a_l(\vec{r})\Psi_l(\vec{r}; z), \tag{3.158}$$

where local modes are determined by

$$\begin{aligned} \frac{\partial^2\Psi_l(\vec{r}; z)}{\partial z^2} + \left[k^2(\vec{r}; z) - \xi_l^2(\vec{r})\right]\Psi_l(\vec{r}; z) &= 0, \\ \Psi_l(\vec{r}; z)|_{z=0} &= 0, \\ \left[\Psi_l(\vec{r}; z) + g(\xi_l(\vec{r}))\frac{\partial\Psi_l(\vec{r}; z)}{\partial z}\right]_{z=H(\vec{r})} &= 0. \end{aligned} \tag{3.159}$$

Substituting (3.165) in (3.164) and carrying out calculations similar to those in previous sections, we obtain a system of coupled equations for the functions $a_l(\vec{r})$

$$\Delta_- a_l(\vec{r}) + \xi_l^2(\vec{r})a_l(\vec{r}) = -\sum_m \left[\vec{B}_{lm}\nabla_- a_m(\vec{r}) + D_{lm}a_m(\vec{r})\right]. \tag{3.160}$$

Here, $\Delta_- = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$.

For simplicity on the right-hand side of the equation, we neglect bottom variations in the terms containing mode coupling. The modes coupling coefficients thus become

$$\vec{B}_{lm} = 2\int_0^\infty \frac{\rho}{\rho(z)}\Psi_l \nabla_- \Psi_m dz,$$
$$D_{lm} = \int_0^\infty \frac{\rho}{\rho(z)}\Psi_l \nabla_-^2 \Psi_m dz. \tag{3.161}$$

Thus, the system (3.160) is simply the extension of system (3.134) for the 3D case. If variations of the waveguide parameters are sufficiently smooth, we can neglect the second coefficient in (3.160) and the system thus has the form

$$\Delta_- a_l(\vec{r}) + \xi_l^2(\vec{r}) a_l(\vec{r}) = -\sum_m \vec{B}_{lm} \nabla_- a_m(\vec{r}), \tag{3.162}$$

where the mode coupling coefficients can be expressed through media parameters. We note that for this case, the mode coupling coefficients $\vec{B}_{lm}$ are vector valued in the horizontal plane.

The next step in our derivation is to cast the modal amplitudes $a_l(\vec{r})$ in the ray theory form

$$a_l(\vec{r}) = A_l(\vec{r}) \exp[i\theta_l(\vec{r})]. \tag{3.163}$$

In other words, we express the modal amplitudes in terms of modulus and phase, both depending on the coordinates in the horizontal plane. Generally speaking, the line of constant phase ($\theta_l(\vec{r}) = \text{const}$) is not a circle here, and if we introduce rays as a lines normal to this phase curve, the rays are not straight lines.

Let us put (3.163) in (3.162). We obtain

$$\nabla_- a_l(\vec{r}) = (\nabla_- A_l + i\nabla_- \theta_l \cdot A_l) \exp[i\theta_l] \tag{3.164}$$

and

$$\begin{aligned}\Delta_- a_l &= \exp[i\theta_l]\left\{i\nabla_-\theta_l \cdot [\nabla_- A_l + i\nabla_-\theta_l \cdot A_l] + \nabla_-^2 A_l + iA_l\nabla_-^2\theta_l + i\nabla_-\theta_l\nabla_- A_l\right\} \\ &= \exp[i\theta_l]\left\{-(\nabla_-\theta_l)^2 \cdot A_l + \nabla_-^2 A_l + iA_l\nabla_-^2\theta_l + 2i\nabla_-\theta_l\nabla_- A_l\right\} \\ &\approx \exp[i\theta_l]\left\{-(\nabla_-\theta_l)^2 \cdot A_l + 2i\nabla_-\theta_l\nabla_- A_l + iA_l\nabla_-^2\theta_l\right\}.\end{aligned}$$

As usual, we neglect the second derivative of the amplitude along the acoustic track. Substituting in (3.162), we get the equation

$$\exp[i\theta_l]\left\{-(\nabla_-\theta_l)^2\cdot A_l + 2i\nabla_-\theta_l\nabla_- A_l + iA_l\nabla_-^2\theta_l\right\}$$
$$+ q_l^2 A_l\exp(i\theta_l) + iq_l\gamma_l A_l\exp(i\theta_l) = -\sum_m \vec{B}_{lm}\exp(i\theta_m)[\nabla_- A_m + i\nabla_-\theta_m\times A_m]. \tag{3.165}$$

Strictly speaking, this expression is a system of two vector equations for the two functions $A_l(\vec{r})$ and $\theta_l(\vec{r})$. We have some freedom in choosing the equation for one of these functions. Let us have the phase satisfying the eikonal equation for the phase function

$$(\nabla_-\theta_l)^2 - q_l^2 = 0 \tag{3.166}$$

and have a system of coupled equations (the "transport equations") for the modulus functions

$$2\nabla_-\theta_l\nabla_- A_l + A_l\nabla_-^2\theta_l + q_l\gamma_l A_l = -\sum_m \vec{B}_{lm}\exp[i(\theta_m-\theta_l)\nabla_-\theta_m\cdot A_m. \tag{3.167}$$

As was said previously, (3.166) can be considered as determining 2D rays (or rays in the horizontal plane or horizontal rays), as curves orthogonal to the lines of constant phase $\theta_l(x,y) = \text{const}$. For each separate mode number, we have corresponding horizontal rays. In general, there are M families of horizontal rays coming out of the source, each corresponding to a given vertical mode, which can reach the point of observation. These rays have separate trajectories; therefore, they have different amplitudes and eikonals. Thus for calculation of the sound field in this approximation, we must carry out a summation over all horizontal eigenrays. We will introduce a second index, for example, A_{lj}, θ_{lj}, where j denotes additional horizontal rays for the lth vertical mode. In easily understandable situations, we will omit the second index. These (3.166), (3.167) are extensions of mode coupling (3.134).

To understand and illustrate the basic ideas of horizontal mode refraction, we can neglect the mode coupling terms on the right-hand side of (3.167). This case, corresponding to the adiabatic approximation in the previous section, is called the approximation of horizontal rays (HR) and vertical modes (Burridge and Weinberg 1977; Brekhovskikh and Lysanov 1991). For this case, instead of (3.167) for the amplitude and the eikonal, we get the traditional 2D equations of ray acoustics:

$$(\nabla_-\theta_l)^2 = (q_l^0)^2 n_l^2(\vec{r}), \tag{3.168a}$$

$$2\nabla_- A_l\nabla_-\theta_l + A_l\nabla_-^2\theta_l + q_l\gamma_l A_l = 0, \tag{3.168b}$$

where we have introduced the refraction index for horizontal rays $n_l(\vec{r}) = q_l(\vec{r})/q_l^0$ (q_l^0 is the amplitude of the wave vector of the lth mode in the absence of horizontal

refraction). Ray trajectories in the horizontal plane can, as usual, be found from the ordinary differential equations.

If we take new variable s – the length of the ray along the path ($ds = \sqrt{dx^2 + dy^2}$), then we can write equations for the ray trajectory for the lth vertical mode in the horizontal plane, given by the equation $\vec{r}_l = \vec{r}_l(s)$:

$$\frac{d\vec{r}_l}{ds} = \vec{\tau}_l, \quad \frac{d\vec{\tau}_l(\vec{r})}{ds} = \frac{1}{q_l(\vec{r})}\nabla_\perp q_l(\vec{r}), \tag{3.169}$$

where the operator $\nabla_\perp = \nabla_{\vec{r}} - \vec{\tau}(\vec{\tau}\nabla_{\vec{r}})$ denotes a gradient in the transverse direction, (orthogonal to $\vec{\tau}$) and $\vec{\tau}_l(\vec{r}) = \nabla_- \theta_l(\vec{r})/|\theta_l(\vec{r})|$ is a unit vector in the horizontal plane, tangent to the horizontal ray.

These equations must be accompanied by conditions at the source $\vec{r}_0 = (x_0, y_0)$ such that $\vec{r}_l(t; \vec{r}_0, \vec{\tau}_0)|_{s=0} = \vec{r}_0$ and $\vec{q}_l(\vec{r}_l, t; \vec{r}_0, \vec{\tau}_0)\big|_{\vec{r}_l=\vec{r}_0, s=0} = q_l(x_0, y_0)\vec{\tau}_0$.

By solving (3.169), we find the ray trajectory and in turn, after integration along this trajectory, we can find the eikonal (for simplicity, we take $\theta_l(\vec{r}_0, 0) = 0$) and the amplitude $A_l(s)$ from the transport equation, giving us

$$\theta_l(\vec{r}_0, s) = \int_0^s q_l(\vec{r}_l(s))ds, \tag{3.170a}$$

$$A_l(s) = \frac{A_l(0)}{\sqrt{J}} \exp\left[-\frac{1}{2}\int_0^s \gamma_l(\vec{r}(s'))ds'\right], \tag{3.170b}$$

where $J(t)$ is the well-known geometrical factor (the Jacobian) which describes the divergence of the horizontal rays. $J(t) = d\Omega_t/\mathrm{d}\Omega_0$ is specifically the ratio of the lengths of the corresponding pieces of the wave fronts along a ray tube. This value (i.e., the amplitude of the sound field) at the receiver can be estimated if we know either the "density" of rays in the horizontal plane or the value of the cross-section. Here, integration is carried out along the ray path. Because the spatial scale of q_l in the horizontal plane is $|q_l/\nabla_- q_l|$, this value must be much greater than a wavelength for HR theory to be applicable.

Instead of the length of the ray path, we can use a time variable – it can be called the arrival time t, or the time of propagation along the ray path. It is concerned with the group velocity of the l-th vertical mode $c_l^{gr} = (dq_l/d\omega)^{-1}$ so that $dt = ds/c_l^{gr}$. The corresponding equations have the form

$$\frac{d\vec{r}_l}{dt} = c_l^{gr}\vec{\tau}_l, \quad \frac{d\vec{q}_l}{dt} = c_l^{gr}\nabla_- q_l, \tag{3.171}$$

where $\vec{q}_l = q_l\vec{\tau}_l$ is the wave vector, tangent to the horizontal ray for the l-th vertical mode.

Equation (3.171) describe space–time horizontal rays (Kravtsov and Orlov 1990) because we use the arrival time as the ray parameter. Using these equations, one can find both the trajectories of the horizontal rays and arrival times of the corresponding modal pulses. We define $\vec{\tau}_0 = (\cos\varphi_0, \sin\varphi_0)$, where φ_0 is the horizontal ray angle at the source. For a fixed point source, φ_0 and t can be considered as ray coordinates. Also, if φ_0 is fixed, a separate horizontal ray can be obtained; for fixed t, a wave front exists in the horizontal plane. As a function of the arrival time, the eikonal is

$$\theta_l(\vec{r}, t) = \int_0^t c_l^{gr} q_l[\vec{r}(t')]dt'. \tag{3.172}$$

The amplitude of the HRs can be written as:

$$A_l(t) = \frac{A_l(0)}{\sqrt{J}} \exp\left[-\frac{1}{2}\int_0^t \gamma_l(\vec{r}(t'))c_l^{gr}dt'\right]. \tag{3.173}$$

Let us now consider the energetic characteristics of the sound field in a waveguide for the 3D problem. In Sect. 3.1, we introduced the horizontal component of spectral intensity $\vec{I}_\omega^h(x, y)$ (3.17) which in our case depends on both horizontal coordinates. The modal expansion of this quantity is

$$\vec{I}_\omega^h = \frac{2\pi i}{\rho\omega}|S(\omega)|^2 \sum_{l,m} [A_l \nabla_- A_m - A_m \nabla_- A_l - 2iA_l A_m \nabla_- \theta_l] e^{i(\theta_l - \theta_m)} \Psi_l \Psi_m. \tag{3.174}$$

The waveguide spectral intensity (flux through the waveguide) is

$$\vec{I}_\omega^w = \frac{2\pi}{\rho\omega}|S(\omega)|^2 \sum_l A_l^2 \nabla_- \theta_l \tag{3.175}$$

and correspondingly the modal spectral waveguide intensity is

$$\vec{I}_{l\omega}^w = \frac{2\pi|S(\omega)|^2 A_l^2 \nabla_- \theta_l}{\rho\omega}. \tag{3.176}$$

The total and modal power can also be written as

$$\vec{W}_\omega = \frac{2\pi}{\rho\omega}|S(\omega)|^2 \sum_l \int_0^{2\pi} A_l^2 \nabla_- \theta_l \vec{r} d\varphi \tag{3.177}$$

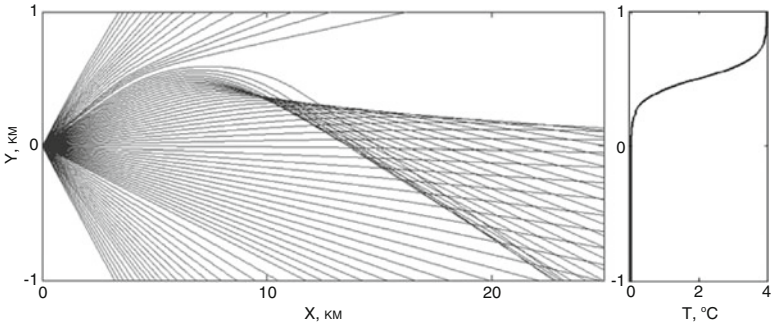


Fig. 3.16 Set of horizontal rays for the first vertical mode incoming from the source in shallow water (Barents sea model) in presence of temperature front. Distribution of temperature along *Y* axis is shown in the *right panel*. Frequency 300 Hz

and

$$\bar{W}_{l\omega} = \frac{2\pi}{\rho\omega}|S(\omega)|^2 \int_0^{2\pi} A_l^2 \nabla_- \theta_l \vec{r} d\varphi. \tag{3.178}$$

In Burridge and Weinberg (1977) and Brekhovskikh and Lysanov (1991), analytic horizontal rays for the useful example of a point source were constructed in a wedge-shaped region. In this case, the horizontal rays are hyperbolas. They have caustics, which are also hyperbolas.

An essential feature of horizontal refraction in our description is the frequency and modal dependence of the refraction index [in the right-hand side of the eikonal equation (3.168a)] and, in turn, of the parameters of the horizontal rays. This means that for different frequencies and mode numbers, we have a different redistribution of the amplitude (intensity) of the sound waves propagating in a medium where horizontal refraction is manifested (i.e., in a spatially anisotropic medium). As an example, we show trajectories of horizontal rays in shallow water (Fig. 3.16) in the area of a temperature front, where we have a significant variation of temperature (and correspondingly in sound-speed profile) in the horizontal plane. In our model, the variation of temperature takes place along the *Y*-axis and the temperature front is placed along the *X*-axis. This model is close in form to the Barents Sea Polar Front, considered in Jin et al. (1996). This pattern is constructed for a fixed frequency (300 Hz) and for the first vertical mode. The source is located at the zero point. We can see a varying distribution of the density of rays (correspondingly of intensity) in the horizontal plane; in particular, in the area "behind" the temperature front there is a shadow zone. Also, in the area at a distance of about 20 km, there is an increase of the sound field due to a crossing of "direct" and "refracted" rays at the point of observation. It should be noted that for another frequency and mode number, this modal multiplication interference pattern will change (Katsnelson et al. 2007). Also, in the sense of space–time rays, we should observe

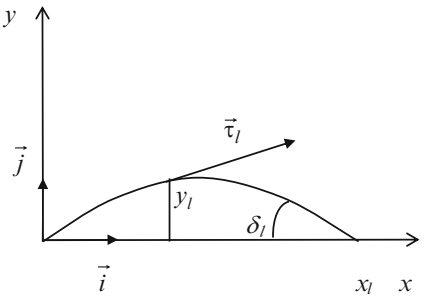


Fig. 3.17 Horizontal ray in the *XY* plane

some delay in the arrival of different horizontal rays, giving a rather complex pattern for the space–time distribution in the horizontal plane.

It also makes sense to solve the equations for the trajectory of a horizontal ray in the general case of a shallow water waveguide using perturbation theory, assuming "smallness" of the horizontal refraction. A perturbation theory for rays (Kravtsov et al. 1984a, b; Kravtsov and Orlov 1990) for our model of shallow water is constructed as follows. Suppose the source is located at a point with coordinates $(0, 0, z_0)$, and the receiver is at the point with coordinates $(x_1, 0, z_1)$ (see Fig. 3.17). We shall assume that the unperturbed horizontal ray is a straight line along the x-axis, or in other words, we choose the X-axis to coincide with an unperturbed ray. The field of the point source, at the point of observation, neglecting horizontal refraction, is determined by expression (3.146):

$$\Psi(x_1, 0, z) = \sum_l \frac{A_l(0,0)}{\sqrt{J}} \Psi_l(x_1, 0; z) \exp\left\{ \int_0^{x_1} \left[iq_l(x,0) - \frac{\gamma_l(x,0)}{2} \right] dx \right\}, \quad (3.179)$$

corresponding to a linear horizontal ray.

We shall now determine first-order corrections, due to the dependence on the transverse coordinate y of the physical parameters of the problem, i.e., the sound speed and the depth of the waveguide. We do this based on the calculation of corrections to the function q_l in the integrand and integration of (3.179) with its corresponding contributions along the unperturbed ray (Kravtsov and Orlov 1990), which is possible if we assume smallness of the displacement of the ray in the horizontal plane. (In the case where the horizontal displacements are very large, a more accurate form of perturbation theory has to be used (Kravtsov and Orlov 1990), where the corrections are iteratively found to the previously constructed "perturbed" ray.) In the following discussion, we suppose that in the regions where the perturbed ray propagates we can restrict decomposition of the refraction index in a horizontal plane or value $q_l(\vec{r})$ by terms linear in the variable y.

Thus, the corrections to $q_l(\vec{r})$ are determined by the equation

$$q_l(\vec{r}) = q_l^0(\vec{r}) + q_l^1(\vec{r}) = q_l(x,0) + \frac{\partial q_l}{\partial y}\Big|_{y=0} y = q_l(x) + q_{ly}(x) y, \quad (3.180)$$

where we used the notation

$$q_{ly}(x) = \frac{\partial q_l(x,y)}{\partial y}\big|_{y=0},$$
$$q_l(x) = q_l(x,0). \tag{3.181}$$

Thus, corrections to the eikonal and ray trajectory will be sought in the form

$$\theta_l(\vec{r}) = \theta_l^0(\vec{r}) + \theta_l^1(\vec{r}),$$
$$\vec{r}_l(s) = \vec{i}x + \vec{r}_l^1(x),$$
$$\vec{\tau}_l(s) = \vec{i} + \vec{\tau}_l^1(x),$$
$$\theta_l^0(\vec{r}) = \int_0^x q_l(x',0)dx'. \tag{3.182}$$

We substitute these expressions into the ray equation, taking into account the fact that the unperturbed ray coincides with the x-axis and that the transverse gradient is determined by the partial derivative with respect to y.

For first-order corrections, we have the equation:

$$\frac{d^2\vec{r}_l^1(x)}{dx^2} = \vec{j}v_l(x), \tag{3.183}$$

where $v_l(x) = q_{ly}(x)/q_l(x)$, and $\vec{j}$ is the unit vector along the y axis.

After double integration, we have

$$\vec{r}_l^1(x) = \vec{r}_l^1(0) + \vec{C}x + \vec{j}\int_0^x dx' \int_0^{x'} dx'' v_l(x'') = \vec{r}_l^1(0) + \vec{C}x + \vec{j}\int_0^x (x-x')v_l(x')dx'. \tag{3.184}$$

The constants $\vec{r}_n^1(0)$ and $\vec{C}$ are determined by the conditions that our ray should go from the point $(0,0)$ on the (x,y) plane and come to the point of observation $(x_1,0)$, or $\vec{r}_l^1(0) = \vec{r}_l^1(x_1) = 0$. Then,

$$\vec{r}_l^1(x) = -\vec{j}\frac{x}{x_1}\int_0^{x_1}(x_1-x')v_l(x')dx' + \vec{j}\int_0^x (x-x')v_l(x')dx'. \tag{3.185}$$

Using this expression, we can find the first-order correction to the eikonal at the point of observation. This correction is obtained after substituting the expression for the horizontal ray taking into account the correction (3.180):

$$\theta_l(x_1,0) = \int_0^{x_1} q_l(\vec{r}_l^0 + \vec{r}_l^1)dx = \int_0^{x_1} q_l(x)dx + \int_0^{x_1} q_{ly}(x)|\vec{r}_l^1|dx. \tag{3.186}$$

After substituting $\vec{r}_n^1$ in the second integral, we get the following corrections to eikonal and to the tangent vector:

$$\theta_l^1(x_1) = \int_0^{x_1} q_{ly}(x) \left[-\frac{x}{x_1} \int_0^{x_1} (x_1 - x')v_l(x')dx' + \int_0^{x} (x - x')v_l(x')dx' \right] dx, \quad (3.187)$$

$$\vec{\tau}_l^1(x_1) = \frac{d\vec{r}_l^1(x)}{dx}|_{x=x_1} = -\vec{j}\frac{1}{x_1} \int_0^{x_1} (x_1 - x)v_l(x)dx. \quad (3.188)$$

The value of the last vector determines the angle of deflection of the horizontal ray at the point of observation $(x_1, 0)$:

$$\delta_l = \frac{1}{x_1} \int_0^{x_1} (x_1 - x)v_l(x)dx. \quad (3.189)$$

From these formulae, it is clear that the corrections are determined by integrals along the unperturbed ray of a transverse derivative of waveguide parameters. Of course, if the waveguide has a small, fluctuating bottom slope or change in sound speed, then these perturbations will largely be compensated for in the integration along a ray, and the corrections at the point of receipt will be insignificant. If the waveguide does have significant transverse gradients (for example, if the track passes along a coastline, where there is a wedge, or along a temperature front), then the perturbations will accumulate and the corrections to the eikonal may be significant (see our example in Fig. 3.16). As another example, if we have a soliton-like internal wave with a long wave front, then as sound propagates between internal wave fronts, such refraction may even lead to the formation of a waveguide in the horizontal plane (Katsnelson and Pereselkov 2000).

We can estimate characteristic values of the horizontal refractive indices for typical shallow water conditions. Suppose for an initial example that along the whole track, the transverse gradient of the sound-speed profile and the depth of the water are more or less constant. For this case, we can easily estimate the values of the angle of horizontal refraction, the values to be added to the eikonal (the additional phase excursion) and also the maximum horizontal deflection of a ray from the track for a mode with index l. If the values of $v_l(x) \approx \bar{v}_l = \text{const}$ for the transverse variations of the waveguide m, then the parameters can be taken outside the integrals in (3.184)–(3.189), and estimates of the lateral deflection of the ray $y_l(x)$ can easily be obtained:

$$y_l(x) \approx \frac{\bar{v}_l}{2} x(x - x_1). \quad (3.190)$$

In this approximation, the horizontal ray has a parabolic shape. A ray in the horizontal plane is deflected from a straight line (the x-axis) toward decreasing values of q_l as a function of y, in other words, in the opposite direction to the transverse gradient of q_l. Since the change in eigenvalue, as the sound-speed profile or the depth varies, depends on the mode index and may be either positive or negative for different indices, horizontal rays corresponding to different mode indices may be deflected from a straight line in different directions. The maximum deflection of a ray trajectory from the direction of the track is equal to

$$\bar{y}_l = \frac{\bar{v}_l x_1^2}{8}. \tag{3.191}$$

The additional phase excursion (of the eikonal) due to horizontal refraction can be estimated as

$$\theta_l \approx \frac{q_{ly}^2}{12 q_l} x_1^3. \tag{3.192}$$

We note that the addition to the eikonal depends on the mode index, which in a real waveguide may lead to a change in the field interference pattern at the receiving antenna when compared with the case where there is no horizontal refraction. We can also estimate the angle in the horizontal plane between a ray at the receiver and the direction of the track:

$$\delta_l = -\frac{1}{x_1} \int_0^{x_1} (x_l - x) v_l(x) dx \approx -\frac{q_{ly} x_1}{2 q_l}. \tag{3.193}$$

The effect of variations in the waveguide depth and the sound-speed profile can be estimated separately, using simple models. For example, let us assume that the waveguide depth varies as $(H = H(y))$ for a constant sound speed $q_l = \sqrt{k^2 - \sigma_l^2}$, that $\sigma_l = \frac{l^* \pi}{H}$, and that $q_{ly} = \frac{\sigma_l^2}{q_l} \frac{H_y}{H}$, where $(H_y = \frac{\partial H}{\partial y})$. Taking into account our previous work on the direction of deflection of a ray, it is clear that for all eigenrays in the horizontal plane, the pattern is deflected in the direction of decreasing depth. Substituting the transverse gradient of the eigenvalues into our previous equations, we obtain

$$\bar{y}_l \approx \frac{\pi^2 l^2}{8 k^2 H^2 \tilde{L}_H} x_1^2, \quad \theta_l \approx \frac{x_1^3}{12 k} \left(\frac{\pi^2 l^2}{H^2 k \tilde{L}_H} \right)^2, \quad \delta_l \approx \frac{x_1}{2 \tilde{L}_H} \frac{\sigma_l^2}{q_l^2}, \tag{3.194}$$

where $\tilde{L}_H = \frac{H}{H_y}$ is the characteristic scale of the change in depth.

We can now estimate the values of the parameters obtained in typical shallow water conditions. For example, if $H = 100$ M, $f = 300$ Hz, $H_y = 10^{-3}$, and $x_1 = 30$ km, then the deviation of the horizontal ray with mode number l from the straight-line

direction is approximately $\bar{y}_l \approx l^2(m)$, the phase shift is $\sim 10^{-4} l^2$, and the angle at which the horizontal ray reaches the point of observation is $\delta_l \approx 10^{-4} l^2$. If we introduce the grazing angle of a vertical ray with the bottom, $\chi_l \approx \frac{\sigma_l}{q_l}$, into the ray approximation, then the horizontal refraction can be expressed in terms of this angle as $\delta_l \approx \frac{x_1}{2L_H}\chi_l^2$. Clearly, horizontal rays corresponding to higher modes experience greater horizontal refraction, and the dependence on the mode index in this example is very sharp.

Let us now consider a waveguide with a negative gradient of the sound speed (downward refracting) and calculate the horizontal refraction for rays corresponding to near-bottom modes. Suppose we have a sound-speed profile $c(y,z) = c_S[1-\varepsilon(y)z]$ or $k^2(y,z) = k_S^2[1+2\varepsilon(y)z]$, where c_S is the sound speed on the surface and c_H is the sound speed near the bottom. The eigenvalues of near-bottom rays (which do not interact with the surface) can be determined from the WKB formulae to be

$$q_l = \sqrt{k_H^2 - \sigma_l^2}, \sigma_l = \left[3(l+\frac{1}{2})\pi\varepsilon(y)k_S^2\right]^{1/3}. \tag{3.195}$$

Thus, $q_{ly} = -\frac{\sigma_l^2}{3q_l}\frac{\varepsilon_y}{\varepsilon}$. If, analogously to earlier work, we introduce the spatial scale of the change of sound speed in the transverse direction (actually the spatial scale of the change in gradient)

$\tilde{L}_c = \frac{c}{c_y} = \frac{a}{a_y}$ and the grazing angle of a ray with the bottom, then we obtain the estimate $\delta_l \approx -\frac{x_1}{3L_c}\chi_l^2$ for the horizontal angle of refraction. It is clear that, as in the previous case, modes with high mode indices deviate more, and that the direction of deviation of a horizontal ray coincides with the direction of the increasing gradient.

As an example of a calculation of a ray trajectory using perturbation theory, we consider a ray path in the presence of internal solitons, having a planar wave front (directed along the X axis) and traveling in the Y direction with some velocity. For this case, the squared refraction index in a horizontal plane can be represented as the sum of an unperturbed part (equal to 1) and a perturbation, denoted as μ_l. In accord with (3.168a), $\mu_l = 2\delta q_l/q_l^0$. Assuming a plane wave front and a constant shape of the ISW envelope, the perturbation depends only on y and on geotime; thus we can express the position of the IW train as

$$n_l^2 = 1 + \mu_l(y), \tag{3.196}$$

where the correction μ_l can be found in different ways (numerically on the basis of the solution of the Sturm–Liouville problem, or analytically, for example, within perturbation theory). The acoustic track (Fig. 3.18), taken as the X' axis, is placed at an angle α with X. We then calculate the ray trajectory from the source to the point of observation $x' = r$, $y' = 0$. To do this, we solve (3.183), which has the form

$$\frac{d^2y'}{dx'^2} = \frac{1}{2}\frac{d\mu_l}{dy'}. \tag{3.197}$$

The solution to this problem with the appropriate conditions at the source and receiver is

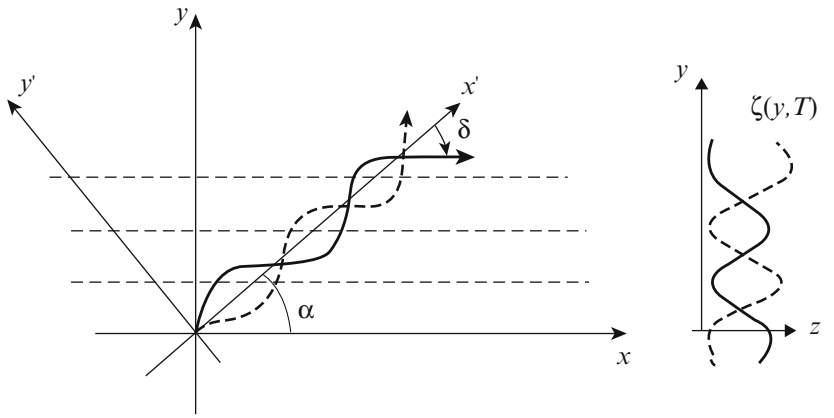


Fig. 3.18 Horizontal rays in presence of internal wave, corresponding to two limit positions. Envelopes of internal waves are shown in the *right side*

$$y'(x') = \frac{\cos\alpha}{2\sin^2\alpha}\left[\int_0^{x'\sin\alpha} \mu_l(y)dy - \frac{x'}{r}\int_0^{r\sin\alpha} \mu_l(y)dy\right]. \tag{3.198}$$

This ray path is shown in Fig. 3.18 for the two positions of the envelope of the internal waves shown in the right panel. During the motion of internal waves, the shape of the ray coming to the point of observation changes and the horizontal angle fluctuates. The amplitude of these fluctuations $\Delta\delta$, which can be seen as the difference between maximal deviations for different positions of envelope, can be estimated from the trajectory equation. If the horizontal angle $\delta \approx \frac{dy'}{dx'}|_{x'=r}$, then $\Delta\delta_l \sim \frac{\mu_l^0}{\alpha}$, where μ_l^0 is the maximal value of the perturbation for a given amplitude of internal waves. It is clear that horizontal refraction should increase with increasing amplitude of the IW and with decreasing angle α. If for a numerical estimate we take the values of the corrections calculated in (Sect. 3.3) (see Fig. 3.7), then for an angle about 5° (or 0.1 rad), we get $\Delta\delta \sim 2^\circ - 3^\circ$ which in principle can be measured in the experiment. Indeed, measurements of this "angular spread" in arrivals have been seen in the recent SW06 measurements (Lynch and Tang 2008).

It is interesting to note that the manifestation of horizontal refraction is connected not only to the bending of the horizontal rays, which in the absence of a perturbation are considered to be straight lines. A more remarkable manifestation takes place as a result of the change in the density of the rays (or divergence, as described by the Jacobian J). This variation can be roughly estimated for internal waves traveling approximately perpendicular to an acoustic track. As an example of the application of perturbation theory, we will show results of a calculation for a ray trajectory in the presence of a train of internal waves.

Another situation, which can allow us to construct an interesting trajectory of horizontal rays, takes place if the perturbation can be considered as depending only

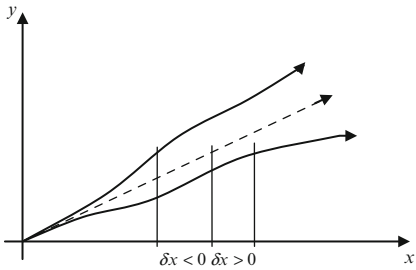


Fig. 3.19 In the presence of the perturbations trajectories of horizontal rays are shown by solid lines. Without the perturbations, trajectory of ray is shown by dotted line

on one coordinate (denoted as y). In this case, we can construct the trajectory using Snell's law. This approach is possible for several models of coastal wedges, or intense internal waves if they have approximately planar wave fronts.

Snell's law for a function $\varphi_l(y)$ describing the angle of horizontal rays with the axis x (see Fig. 3.19) has the form

$$\frac{\cos\varphi_l^0}{\cos\varphi_l(y)} = 1 + \frac{\mu_l(y)}{2}. \tag{3.199}$$

For a ray trajectory, determined by a function $y(x)$, or in a more convenient form $x(y)$, we have[5]

$$\frac{dx}{dy} = \cot\varphi_l(y). \tag{3.200}$$

Thus, the trajectory in the form $x(y)$ of a horizontal ray, incoming in the plane is

$$x = \int_0^y \frac{\cos\varphi_l}{\sqrt{1-\cos^2\varphi_l}}\,dy'. \tag{3.201}$$

Using (3.199), we get

$$x = \cos\varphi_l^0 \int_0^y \frac{dy'}{\sqrt{\mu_l + \sin^2\varphi_l^0}}. \tag{3.202}$$

Let us consider the case of appreciable angles where $\mu_l \ll \sin^2\phi_l^0$. Usually $|\mu_l| \leq 1 - 3 \times 10^{-3}$, and so this inequality will be true for angles $\varphi > \sqrt{\mu} \sim 0.02 - 0.05$, or in other words, angles more than 4°–5°. In this case, we can simplify this equation to

[5] Strictly speaking we should write for every vertical mode $x_l(y)$ and $y_l(x)$. However, for simplicity we will omit subscript.

$$x = y\cot\varphi_l^0 - \frac{\cot\varphi_l^0}{2\sin^2\varphi_l^0}\int_0^y \mu_l(y')dy'. \tag{3.203}$$

We can see that the second term on the right-hand side gives the correction δx to a straight-line trajectory in the horizontal plane (see Fig. 3.19)

$$\delta x = -\frac{\cot\varphi_l^0}{2\sin^2\varphi_l^0}\int_0^y \mu_l(y')dy'. \tag{3.204}$$

This indicates that if in the absence of a perturbation we have a homogeneous angular distribution of rays, then in the presence of a perturbation depending on the sign of δx, we have either an increase or a decrease of density of the rays, and in turn an increase or a decrease of the sound intensity. In the case of internal waves, where the perturbation of the sound speed δc and in turn μ_l depends on time, we have temporal variations of the sign of δx and corresponding space–time fluctuations of intensity. These fluctuations can easily reach 5–6 dB, as observed experimentally (Badiey et al. 2005).

In conclusion, we note that as the above estimates show, for typical shallow water conditions, and for low (weakly decaying) modes propagating over tracks of several tens of kilometers, the horizontal angle of refraction may reach several degrees. The sound-speed profile (gradient) may vary markedly at these distances, and when these refractive quantities vary simultaneously, their mutual compensation or reinforcement may play a role. In Doolittle et al. (1988), horizontal refraction was measured experimentally in a coastal wedge region. The inclination was very significant, with $H_y \approx 10^{-2}$ so that the horizontal angle of refraction amounted to several degrees. See also estimations in Kuz'kim (1996).

3.8 Parabolic Equation

The Parabolic Equation (PE) method is unarguably one of the most useful approximations for solving the problem of wave propagation in inhomogeneous media. PE in underwater acoustics is an extension of a method first invented by Leontovich and Fock (1946) [also see Fock (1965)]. The extension was first done by Tappert (1977) and since has been extended by many other authors. We will illustrate the derivation of the PE using the time independent wave equation for the Green function in an inhomogeneous waveguide

$$[\nabla^2 + k^2(x,y,z)]\Psi(x,y,z) = 0 \tag{3.205}$$

with the appropriate boundary conditions at the source.

At the core of PE, there is an assumption that the sound field (or more exactly the spatial spectrum of the sound field) is concentrated in some definite direction. In underwater acoustics, this situation is common since the horizontal angles of propagating rays and modes are often of the order of only a few degrees, and so we can consider the sound field in the underwater channel as a narrow beam, concentrated in the direction of the acoustic track. If we suppose that this direction is the X-axis in a Cartesian coordinate system (the Z-axis being directed down), then mathematically this corresponds to the following form of the solution of (3.205):

$$\Psi(x,y,z) = F(x,y,z)e^{ik_0 x}, \tag{3.206}$$

where k_0 is some reference (average) value of the wave vector. In order to get an equation for the function $F(x,y,z)$, we should know the spatial scales of variability of this function. To understand these scales, we will consider first the simple example of a homogeneous medium, where $k = k_0 = \text{const}$ and will analyze a plane wave, propagating in the direction defined by the vector $\vec{k_0} = (k_x, k_y, k_z)$. Assuming a plane wave in the form (3.206), we can analyze the factor F. Suppose that the direction of propagation is close to the x-axis; for this case, the components k_y, k_z are small. If we suppose that $k_y/k_0,\ k_z/k_0 \sim \varepsilon \ll 1$ (the parameter determining the variability in the y and z directions), we can write for the plane wave

$$e^{i\vec{k_0}\vec{R}} = e^{ik_x x + ik_y y + ik_z z} = e^{i\sqrt{k_0^2 - k_y^2 - k_z^2}x + ik_y y + ik_z z} \sim e^{ik_0 x} \times e^{-i\frac{k_y^2 + k_z^2}{2k_0}x + ik_y y + ik_z z}. \tag{3.207}$$

The variability in the X-direction is determined by the parameter $(k_y^2 + k_z^2)/2k_0^2 \sim \varepsilon^2$.

On the basis of these assertions, we can also introduce new variables for the factor F, denoted as $x' = \varepsilon^2 x$, $y' = \varepsilon y$, $z' = \varepsilon z$. With this modification, we will attempt to find $F = F(x', y', z')$, which now explicitly includes the small parameter. We substitute (3.206) in (3.205) to obtain

$$\varepsilon^2 \frac{\partial^2 F}{\partial y'^2} + \varepsilon^2 \frac{\partial^2 F}{\partial z'^2} + \varepsilon^4 \frac{\partial^2 F}{\partial x'^2} + (k^2 - k_0^2)F + \varepsilon^2 2ik_0 \frac{\partial F}{\partial x'} = 0. \tag{3.208}$$

If we neglect terms of order $\sim\varepsilon^4$, and also come back to the coordinates x, y, z, then we get

$$2ik_0 \frac{\partial F}{\partial x} + \frac{\partial^2 F}{\partial y^2} + \frac{\partial^2 F}{\partial z^2} + (k^2 - k_0^2)F = 0. \tag{3.209}$$

This equation is called PE. Physically speaking, the neglect of terms of order $\sim \varepsilon^4$ in (3.208) corresponds to neglecting waves propagating in the negative X-direction (reflected waves).

In order to understand the structure of the PE solution and its difference from the solutions of the wave equation (3.2), it is useful to consider (3.209) in a

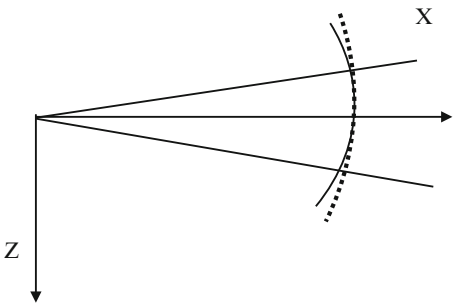


Fig. 3.20 Wave fronts of Green functions Ψ_0 of Helmholtz equation (3.3) (*dotted line*) and wave front of Green function of PE equation (3.212) (*solid line*)

homogeneous medium $k = k_0 = \text{const}$. In this case, the amplitude (denoted as F_0) satisfies equation

$$2ik_0 \frac{\partial F_0}{\partial x} + \frac{\partial^2 F_0}{\partial y^2} + \frac{\partial^2 F_0}{\partial z^2} = 0. \tag{3.210}$$

Solutions of this equation can be simply constructed as

$$F_0(x, y, z) = \frac{1}{4\pi x} e^{ik_0 \frac{y^2 + (z - z_0)^2}{2x}} \tag{3.211}$$

and so the sound field of a point source in the PE approximation (the Green function) according to (3.206) is

$$\Psi_0(x, y, z) \sim \frac{1}{4\pi x} e^{ik_0 \left[x + \frac{y^2 + (z - z_0)^2}{2x}\right]}. \tag{3.212}$$

It is interesting to compare this value with the exact Green function (3.3). We can see in Fig. 3.20 that these functions are close to each other in regions where $x \gg |y|, |z - z_0|$, in other words in the small angle (paraxial) region. Next, we address the wave fronts of these expressions. We see that the wave front of function (3.3) is spherical, whereas the wave front of the PE Green function has a parabolic shape. In the area near the x-axis, these fronts are close (see Figs. 3.20 and 3.21).

Using solution (3.212), we can construct a sound field having finite spatial extent in the plane $x = 0$, for example, a Gaussian form:

$$F(0, y, z) = \bar{F} e^{-\frac{y^2 + (z - z_0)^2}{2a^2}}, \tag{3.213}$$

where $\bar{F} = \text{const}$ is the amplitude and $a = \text{const}$ is the width of the Gaussian beam.

The function $F(x, y, z)$ can then be constructed using the Green function (3.212), and has the form

$$F_{\mathrm{G}}(x, y, z) = \bar{F} \frac{a^2}{a^2 + ix/k_0} \exp\left[-\frac{y^2 + (z - z_0)^2}{2(a^2 + ix/k_0)}\right]. \tag{3.214}$$

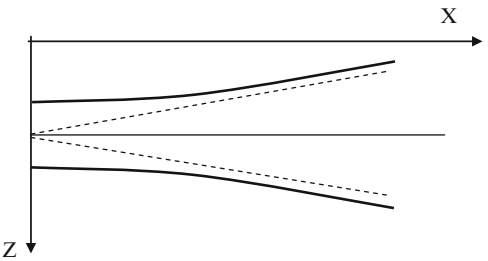


Fig. 3.21 Gaussian beam (lines of constant intensity) F_G (3.214)

This function is the so-called "Gaussian beam," which has a Gaussian distribution of amplitude across its axis (in our case the straight line $y = 0$, $z = z_0$). The spatial width of this distribution increases during propagation, and the maximal amplitude decreases as a function of x. The surface of constant amplitude is a hyperbolic cone; in the XZ plane, lines of constant amplitude are hyperbolas (see Fig. 3.21).

These types of solutions of the parabolic equation are widely used in diffraction theory and the theory of laser radiation. There are many types of similar beams e.g., Hermitian–Gaussian beams (Jensen et al. 1994). If we deal with inhomogeneous media, we should solve (3.209). In diffraction theory, where we deal with unbounded media, it is possible to construct solutions similar to the aforementioned Gaussian beams in inhomogeneous media, using one of the basic curvilinear ray paths as the beams direction and transverse rays as directions of field diffusion.

For underwater acoustics, we can derive applicability criteria for the PE to model the shallow water channel. In the simplest case, we assume cylindrical symmetry for the problem. In this case, we should solve (3.7) which has a solution of the form

$$\Psi(r, z) = \frac{\phi(r, z)}{\sqrt{r}}. \tag{3.215}$$

For long enough distances, we have the 2D wave equation

$$\frac{\partial^2 \phi}{\partial r^2} + \frac{\partial^2 \phi}{\partial z^2} + k_0^2 n^2(r, z)\phi = 0, \tag{3.216}$$

where $k = k_0 n$.

One main feature of PE is the separation of fast variability from slow variability. We assume that in the r direction, the prime direction of propagation of the sound field, we are close to a plane wave, i.e.,

$$\phi(r, z) = F(r, z)e^{ik_0 r}, \tag{3.217}$$

where $F(r,z)$ is a factor that is slowly varying along the direction of propagation. If we substitute (3.217) into (3.216) and neglect the term $\frac{\partial^2 F}{\partial r^2}$ in the following calculations, we get

$$2ik_0\frac{\partial F}{\partial r}+\frac{\partial^2 F}{\partial z^2}+k_0^2[n^2(r,z)-1]F=0, \tag{3.218}$$

which is the simplest case of a parabolic equation. For a shallow water waveguide, this equation should be accompanied by the appropriate boundary conditions at the bottom and surface.

In Fig. 3.22, we show an example of the sound field calculated using different methods including the PE method. (We actually compare three calculations in this figure, giving the sound field in the PE, mode, and ray approximations. We see a satisfying consistency between methods).

For the 3D shallow water problem, we consider the following approach, similar to the approach discussed in Sect. 3.6, which we call "vertical modes and PE in the horizontal plane." According to this approach, the solution of (3.209) should be found in the form of (3.158). Neglecting mode coupling of the amplitudes, we have

$$\Delta_{-}a_l(\vec{r})+\xi_l^2(\vec{r})a_l(\vec{r})=0. \tag{3.219}$$

Using the PE, we find solutions of (3.219) in the form

$$a_l(\vec{r})=F_l(\vec{r})e^{iq_l^0 x},$$

where q_l^0 is some eigenvalue, corresponding to a given mode number. For the function $F_{l,}$ we can get the equation

$$\frac{\partial F_l}{\partial x}=\frac{i}{2q_l^0}\frac{\partial^2 F_l}{\partial y^2}+\frac{iq_l^0}{2}\left(n_q^2(x,y)-1\right)F_l, \tag{3.220}$$

where $n_q(x,y)=q_l(x,y)/q_l^0$ is the refraction index in the horizontal plane for the vertical mode with number l.

Using this method, it is possible to study many different 3D problems in shallow water acoustics, for example, sound propagation in the presence of different hydrodynamic perturbations (internal solitons, temperature fronts) and also propagation near the continental shelf.

As an example of using the PE approach in the horizontal plane we show calculations of the behavior of the sound field amplitude for the SWARM'95 experiment (Badiey et al. 2005). In Fig. 3.23, horizontal rays and amplitudes of the sound field calculated by PE in the horizontal plane for the first vertical mode are shown. For these calculations, we took two different positions of the train of internal waves (the shapes and positions of the envelope are shown in the left panel of the figure). Consistent with our previous theory for these two positions, we see

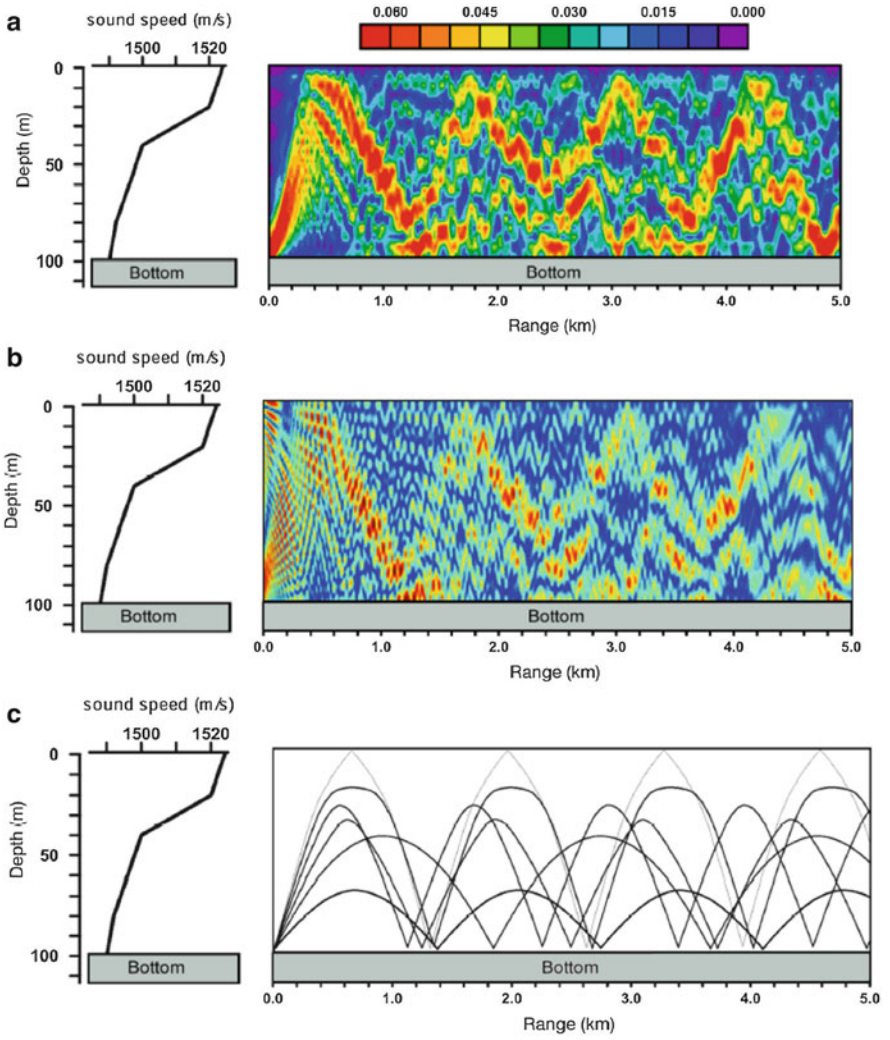


Fig. 3.22 Sound field in shallow water waveguide. Interference (**a** – modes summation, **b** – Parabolic equation) and ray patterns (**c**) for the sound field from a point source. *Left panels* show the sound speed profile. The source is taken to be on the bottom, frequency $f = 250$ Hz. *Bottom parameters* $c_1 = 1{,}750$ m/s, $\rho_1 = 1.8$ g/cm^3, $\alpha = 0.02$. Scale of colour gradation is shown in (**a**)

different behavior of the horizontal rays and correspondingly the PE solutions. These can be called focusing [(b) and (d)] and defocusing [(a) and (c)] situations. As a result, we have increasing and decreasing of the sound field intensity at the receiver (the acoustic track is shown in the figure by a heavy straight line).

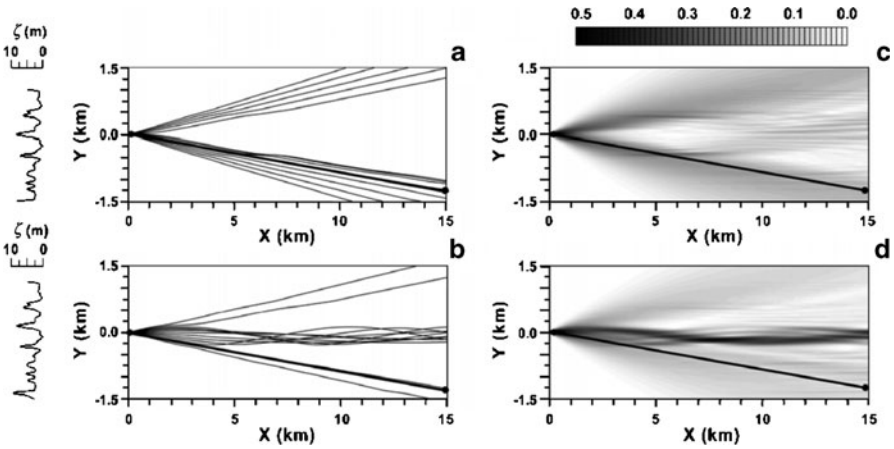


Fig. 3.23 Calculation of the acoustic field in the horizontal plane corresponding to two different positions of the soliton's train, showing focusing and defocusing by horizontal refraction. Positions of internal waves are shown in the *left panel*. In the Fig's (**a**, **b**) horizontal rays are shown, in the fig's (**c**, **d**) results of PE in horizontal plane. Sound speed profile and waveguide parameters are taken for the SWARM'95 experiment (Fig. 4.11), frequency is 60 Hz, mode 1

What we have shown here is just a small amount of small angle PE theory. The field is a huge one, and there are numerous papers considering the many theoretical aspects of and applications of the PE method [see, e.g., Jensen et al. (1994)]. We would also note that PE creates a "full field" solution efficiently, but does not provide the physical insight that the ray and mode decompositions do.

Chapter 4
Examples Illustrating the Characteristics of Waveguide Propagation

4.1 A General Transmission Loss Example, Including a Simplified Theory of Shallow Water Attenuation

As we have often pointed out, when considering the propagation of sound in a shallow water waveguide, it is of primary importance to take into account interaction of the sound with the bottom. The hardest quantity to quantify in this interaction is absorption in the bottom, which leads to a fall in the signal intensity with distance. Formally, this decay is due to the complex eigenvalues in the Sturm–Liouville problem (3.7), or equivalently the constants of propagation ξ_l having an imaginary part $\gamma_l = 2\mathrm{Im}\xi_l$ (i.e., the attenuation coefficients of the normal modes). Despite the fact that the values of γ_l are easily obtained numerically, though not experimentally, a more detailed study of their analytic behavior is advisable, particularly as a function of the parameters of the problem. We should first look at the attenuation as a function of mode number l, since this dependence is crucial to explain the structure of the field formed at a large distance from the source, including its modal composition and total intensity. Moreover, in performing approximate calculations of the averaged intensity, simplified analytic formulae for γ_l are required. These simplified formulae can be obtained and interpreted in the framework of ray (or WKB) theory.

In ray theory, the absorption of sound is of a discrete nature, as energy is lost at the instant a ray is reflected from the bottom; this is typically described by loss per reflection. These losses can also be associated with the mode attenuation coefficient if one uses a spatial scale of interference $\Lambda_l \approx 2\pi|q_l - q_{l+1}|^{-1}$ to connect the pictures. The decay of the mode per interference wavelength Λ_l is precisely equal to the losses on reflection. In turn, Λ_l corresponds to the cycle distance D_l of an equivalent ray, up to a correction (if it is needed, which often it is not) for the "beam displacement" of the ray upon reflection. In this approach, one can obtain the dependence of the quantity γ on the mode index l or, in ray theory, on the grazing angle of the ray χ, corresponding to the given mode index.

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_4,
© Springer Science+Business Media, LLC 2012

These and related questions have been considered many times in the literature (Krupin 1972; Katsnelson et al. 1984; Weston 1994). In Tindle and Weston (1980), Tindle and Bold (1981), Tindle and Dean (1985), Tindle et al. (1987), the connection between γ, D, and the losses on reflection was considered based on the Pekeris model. In Koch et al. (1983) this problem was solved for an arbitrary profile $c(z)$ based on numerical analysis. In Kuperman and Jensen (1980), the results of an approximate computation of γ_l were compared with those obtained based on the solution of a Sturm–Liouville problem for a concrete hydro-acoustic experiment which was made. The importance of taking into account the displacement of the beam on reflection in a number of cases was first pointed out in Tindle and Weston (1980). In Krupin (1972) and Weston (1994), the connection between ray and mode representations was obtained based on the WKB method; in Macpherson and Daintith (1967), the stratification and inhomogeneity of the bottom were also taken into account. All the works cited, except for Katsnelson (1986), consider regular waveguides.

Let us now present the derivation for γ_l, based mainly on the results of Krupin (1972), and Tindle and Weston (1980), using as a framework the waveguide model previously formulated. We first multiply (3.7) by $\Psi_l(z)$ and then integrate both sides of the equation from 0 to H. Taking into account the boundary conditions, we have

$$-\frac{\Psi_l^2(H)}{g(\xi_l)}-\int_0^H(\Psi_l'(z))^2dz+\int_0^H k^2(z)\Psi_l^2(z)dz-\xi_l^2\int_0^H\Psi_l^2(z)dz=0. \tag{4.1}$$

We shall further assume that the imaginary corrections are small in comparison with the real parts of these quantities so that the eigenfunctions of the problem are real. In this approximation,[1]

$$\frac{1}{g(\xi_l)}=\frac{\sqrt{\xi_l^2-k^2{n_1}^2(1+i\alpha)}}{m_1}=\frac{\sqrt{q_l^2-k^2{n_1}^2}}{m_1}+i\frac{q_l\gamma_l-k^2{n_1}^2\alpha}{2m_1\sqrt{q_l^2-k^2{n_1}^2}}. \tag{4.2}$$

Substituting this expression into (4.1) and separating out the imaginary part, we obtain

$$\gamma_l q_l\left[\int_0^H\Psi_l^2(z)dz+\frac{\Psi_l^2(H)}{2m_1\sqrt{q_l^2-k^2{n_1}^2}}\right]=\frac{\alpha k^2{n_1}^2}{2m_1\sqrt{q_l^2-k^2{n_1}^2}}\Psi_l^2(H). \tag{4.3}$$

[1] We used a liquid bottom here for simplicity. In the following, we will show that the expression for actual shallow water conditions can be reduced to the liquid bottom case by redefinition of the bottom parameters.

Taking into account the normalization of the eigenfunctions, we obtain for the attenuation coefficients

$$\gamma_l = \frac{k^2 {n_1}^2 \alpha}{2m_1 q_l \sqrt{q_l^2 - k^2 {n_1}^2}} \cdot \Psi_l^2(H) = \frac{k^2 n_1^2 \alpha}{2m_1^2 q_l} g_l \Psi_l^2(H), \tag{4.4}$$

where $g_l = g(q_l) = \frac{m_1}{\sqrt{q_l^2 - k^2 n_1^2}}$.

Equation (4.4) can also be obtained from perturbation theory if for the integrand in (3.56b) one uses the expression for the eigenfunction employed in the derivation of equation (3.56b).

Let us now write down the real part of (4.1):

$$-\frac{\Psi_l^2(H)}{m_1}\sqrt{q_l^2 - \frac{\omega^2}{c_1^2}} - \int_0^H (\Psi_l')^2 dz + \omega^2 \int_0^H c^{-2}(z)\Psi_l^2(z)dz - q_l^2 \int_0^H \Psi_l^2(z)dz = 0. \tag{4.5}$$

Here, for $k^2(z)$, we have substituted an expression in terms of the sound frequency and the sound speed profile in water. If in expression (4.5) the squares of the eigenvalues q_l and the frequency ω are taken to be variables, it then has the functional form

$$F(q_l^2, \omega^2) = 0, \tag{4.6}$$

where

$$\frac{\partial F}{\partial q_l^2} dq_l^2 + \frac{\partial F}{\partial \omega^2} d\omega^2 = 0. \tag{4.7}$$

Assuming that the eigenvalues have weak frequency dependence and calculating the partial derivatives, we obtain for the normalized functions

$$\frac{q_l}{\omega}\frac{dq_l}{d\omega} = \int_0^H \frac{\Psi_l^2(z)}{c^2(z)} dz + \frac{\Psi_l^2(H)}{2m_1 c_1^2 \sqrt{q_l^2 - k^2 {n_1}^2}}. \tag{4.8}$$

Replacing the last term in this expression using (3.9) and taking into account the definitions of the phase and group velocities of the modes from (3.13), we obtain

$$c_l^{\mathrm{gr}} c_l^{\mathrm{ph}} = \left[c_1^{-2} + \int_0^H \left(c^{-2}(z) - {c_1}^{-2}\right)\Psi_l^2(z)dz \right]^{-1}. \tag{4.9}$$

For the simple limit case of the Pekeris model, where $c(z) = c = \text{const}$ and $\Psi_l(H) = 0$, we get the well-known relationship $c_l^{\text{gr}} c_l^{\text{ph}} = c^2$.

For the mode attenuation coefficient (4.4), a simple estimation formula can be obtained using the ray approximation. To find the value of $\Psi_l(H)$, we simply use the WKB method to represent the function in the water layer. Here, we shall neglect the absorption from the discussions of Sect. 3.3 and assume the eigenvalues q_l and the eigenfunctions to be real

$$\Psi_l(z) = \frac{A_l}{\sqrt{\sigma_l(z)}} \sin[\phi_l(z)], \quad \phi_l(z) = \int_{\underline{z}}^{H} \sigma_l(z) dz + \zeta_l, \tag{4.10}$$

where $\sigma_l(z) = \sqrt{k^2(z) - q_l^2}$, $\underline{z}$ is a turning point of the corresponding ray, and the phase correction ζ_l depends on the nature of the turning point and the acoustical properties of the sea bottom. We note that in shallow water, a ray may undergo reflection from the surface in which case, the value $\underline{z} = 0$ should be taken for the initial lower limit of the integral. $\zeta_l = \zeta_{0l}$ and the value ζ_{0l} lies in the range from $\pi/2$ to 0 for rigid and pressure-release bottoms, respectively. If a ray does not reach the surface (and there is a turning point of another type), then $\underline{z} \neq 0$ and $\zeta_l = \zeta_{0l} - \pi/4$. From this analysis, two types of modes and corresponding rays exist in shallow water, which can be called bottom-surface modes (rays) and bottom modes (rays), respectively. The value of A_l is determined by the normalization equation and by the boundary conditions. From the boundary conditions, we obtain

$$\tan\phi_l(H) + \sigma_l(H) g(q_l) = 0.$$

In what follows, we shall write down the functions without arguments for $z = H$, $\sigma_l = \sigma_l(H), k = k(H)$, etc. The normalization condition gives the value for A_l.

$$\begin{aligned} A_l &= \left[\int_{\underline{z}}^{H} \frac{\sin^2\phi_l(z)}{\sigma_l(z)} dz + \frac{\sin^2\phi_l}{2m_1\sigma_l\sqrt{q_l^2 - k^2 n_1^2}} \right]^{-\frac{1}{2}} \\ &= \left\{ \frac{1}{2} \left[\int_{\underline{z}}^{H} \frac{dz}{\sigma_l(z)} - \int_{\underline{z}}^{H} \frac{\cos 2\phi_l(z)}{\sigma_l(z)} dz + \frac{\sigma_l g_l^3}{m_1^2(1 + \sigma_l^2 g_l^2)} \right] \right\}^{-\frac{1}{2}} \\ &= \left\{ \frac{1}{2} \left[\int_{\underline{z}}^{H} \frac{dz}{\sigma_l(z)} + \frac{k^2(1 - n_1^2) g_l^3}{\sigma_l m_1^2(1 + \sigma_l^2 g_l^2)} \right] \right\}^{-\frac{1}{2}}. \end{aligned} \tag{4.11}$$

To compute the second integral, we assumed that the function $\sigma_l(z)$ is smooth in comparison with the cosine and thus have taken it out from under the integral.

Substituting (4.11) and (4.10) into (4.4), we obtain via the WKB method the following expression for γ_l:

$$\gamma_l = k^2 {n_1}^2 \alpha \frac{\dfrac{\sigma_l g_l^3}{{m_1}^2 q_l (1+\sigma_l^2 g_l^2)}}{\displaystyle\int_z^H \frac{dz}{\sigma_l(z)} + \frac{k^2(1-{n_1}^2) g_l^3}{\sigma_l {m_1}^2 (1+\sigma_l^2 g_l^2)}} \approx 2k^2 n_1^2 \alpha \frac{\sigma_l g_l^2}{m_1^2 D_l}. \tag{4.12}$$

We now obtain the ray interpretation of (4.12). For this, we need to determine the connection between the modes of the waveguide and the rays with corresponding Brillouin angles. This question has been studied in great detail in the literature (Brekhovskikh 1965; Tindle and Weston 1980; Weston 1994), so we shall not reproduce this analysis but merely state the connection between the eigenvalues and the grazing angle χ

$$\cos \chi_l(H) = q_l / k(H). \tag{4.13a}$$

Thus, the angle goes between the direction of the ray, corresponding to the mode with index l and the bottom. Other "ray" characteristics, such as the cycle distance of the ray in the waveguide D_l

$$D_l = 2 \int_{z_t}^H \frac{q_l}{\sqrt{k^2(z) - q_l^2}} dz \tag{4.13b}$$

or the time of propagation along a ray cycle T_l

$$T_l = 2 \int_{z_t}^H \frac{k(z)}{c(z)\sqrt{k^2(z) - q_l^2}} dz, \tag{4.13c}$$

can be expressed in terms of the grazing angle and the parameters of the waveguide. In subsequent formulae in this section, we shall omit the mode index (l) on the corresponding quantities. We shall assume that the dependence on the mode index enters through the value of the corresponding angle χ. In what follows, we shall also need the reflection coefficient V of a plane wave incident at the lower medium with wave vector (q, σ). Its value is easily obtained using the boundary condition (3.5) in Appendix F:

$$V = \frac{\sigma g + i}{\sigma g - i}. \tag{4.14}$$

When there is no absorption, the value of $g(q)$ is real, $|V| = 1$, and the argument is determined by the expression

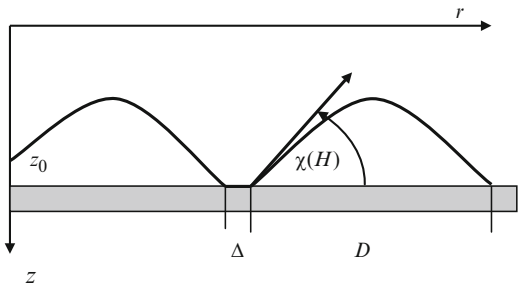


Fig. 4.1 Ray path, corresponding to the l-th normal mode. Cycle distance D, ray displacement Δ and grazing angle at the bottom $\chi(H)$ (or simply χ) are shown (index l is omitted)

$$\varphi_V = \arg V = -2\arctan\left(\frac{1}{\sigma g}\right). \tag{4.15}$$

In the presence of weak absorption, we obtain for $|V|$ to first order in α

$$|V| \approx 1 - \alpha k^2 {n_1}^2 \frac{\sigma g^3}{{m_1}^2(1+\sigma^2 g^2)}. \tag{4.16}$$

Let us consider the denominator of (4.12) in more detail. It is easy to see that the integral $\int dz/\sigma(z)$, corresponding to the phase change in the water layer, determines the horizontal displacement of the ray D on passing through the water layer (in other words, the cycle distance of the ray in the channel)

$$\int_{\underline{z}}^{H} \frac{dz}{\sigma(z)} = \frac{D}{2q}. \tag{4.17}$$

The meaning of the second term in the denominator of expression (4.12) becomes clear if we evaluate the phase of the coefficient of (4.15), differentiate it with respect to q, and then let $q = q_l$, giving

$$\frac{d\varphi_V}{dq} = -2\frac{qk^2(1-{n_1}^2)g^3}{\sigma {m_1}^2(1+\sigma^2 g^2)} = -\Delta, \tag{4.18}$$

where Δ is the horizontal displacement of the beam on reflection (see Fig. 4.1). Substituting these results into (4.12), we finally obtain for the mode attenuation coefficient the well-known expression

$$\frac{\gamma}{2} = \frac{1-|V|}{D+\Delta} = -\frac{\ln|V|}{D+\Delta}. \tag{4.19}$$

When $1 - |V|$ is small, we have $-\ln|V| \approx 1 - |V|$. In most cases, the displacement of the beam is small compared with D (see the estimated values of Δ at the end of the section) and can be neglected. As a result, we obtain the frequently used formula, see, for example, Brekhovskikh and Lysanov (1991),

$$\gamma/2 = -\ln|V|/D. \tag{4.20}$$

By way of example, let us consider the attenuation coefficient γ as a function of grazing angle χ in the Pekeris model. $V(\chi)$ is given by the standard Fresnel equation (Appendix F)

$$V(\chi) = \frac{m_1 \sin\chi - \sqrt{{n_1}^2(1+\mathrm{i}\alpha) - \cos^2\chi}}{m_1 \sin\chi + \sqrt{{n_1}^2(1+\mathrm{i}\alpha) - \cos^2\chi}}, \tag{4.21}$$

where typical values of n_1 lie in the range 0.9–1.0 corresponding to the region $\chi \leq 0.3$ for total reflection angles. In many cases of interest, only small angles χ play a role. Therefore, we give a decomposition of V in a power series in χ up to the third order, which we shall need in subsequent discussions

$$\begin{aligned} V(\chi) = -1 + \frac{2m_1}{\sqrt{{n_1}^2(1+\mathrm{i}\alpha) - 1}}\chi - \frac{2{m_1}^2}{{n_1}^2(1+\mathrm{i}\alpha) - 1}\chi^2 \\ - \frac{m_1({n_1}^2(1+\mathrm{i}\alpha) - 6m_1^2 + 2)}{3({n_1}^2(1+\mathrm{i}\alpha) - 1)^{3/2}}\chi^3 + \ldots \end{aligned} \tag{4.22}$$

The cycle distance of a ray in the Pekeris model is defined by the expression $D = 2H\mathrm{ctg}(\chi)$. For γ, we thus obtain the known expression

$$\gamma = \frac{s}{H}\chi^2, \quad s = \mathrm{Re}\,\frac{2m_1}{\sqrt{{n_1}^2(1+\mathrm{i}\alpha) - 1}} \approx \frac{m_1 n_1^2}{(1-n_1^2)^{3/2}}\alpha. \tag{4.23}$$

From (4.23), it is clear that the values of the mode attenuation coefficient may grow sharply as the angle χ grows. For small angles χ_l, we can write $\chi_l = \frac{\pi l^*}{kH}$ and thus,

$$\gamma_l = \frac{s\pi^2(l^*)^2}{k^2H^3}. \tag{4.24}$$

It is clear from this that the coefficient γ_l grows as the square of the mode index. One practical consequence of these relations is that in a waveguide corresponding to the Pekeris model, as the sound propagates the high modes die out very quickly and so the far field is essentially determined by the first few modes.

The dependence of the ray cycle distance D on the grazing angle is determined by the sound speed profile. The function $\gamma(\chi)$ is also modified by the shape of this profile. For example, if the sound speed profile has a mainly negative gradient, then because of refraction the ray is deflected toward the bottom and may not even touch the surface. A channel with this characteristic will be called a near-bottom channel; as the grazing angle of a ray grows, its cycle distance increases and the attenuation coefficient $\gamma(\chi)$ may thus decrease. The decrease (and in general the nonmonotonic nature) of the attenuation coefficient with increasing mode index in a waveguide with a negative sound speed gradient was first established and studied in Krupin (1972). He has shown that depending on the value of the sound speed gradient and starting from the least mode number, the attenuation coefficients have the same order of magnitude and can initially even decrease with mode index. Higher mode indices may be even more strongly extinguished. Physically, this leads to the fact that a group of weakly decaying modes may contribute most strongly to the field at large distances from the source, which is very important for ocean acoustics.

Simple analytic formulae describing this phenomenon can be obtained for the case of a channel with linear sound speed dependence $c(z) = c_S(1 - \varepsilon z)$ with an absorptive, homogeneous bottom. This sound speed profile can and will be used as a good approximation to real sound speed profiles with negative gradients for qualitative estimations. In this case, the cycle distance of bottom rays D grows as a function of grazing angle, until the ray grazes the surface, while for rays that are reflected from the surface (bottom-surface rays), it decreases as χ grows. This is described by the equation

$$D(\chi) = 2\mathrm{Re}\left(\frac{c_H \tan\chi}{c_S\varepsilon} - \frac{c_H}{c_S\varepsilon\cos\chi}\sqrt{1 - \frac{c_S^2}{c_H^2}\cos^2\chi}\right). \tag{4.25}$$

Here, $c_H = c_S(1 - \varepsilon H)$ is the sound speed at $z = H$. The critical angle corresponding to a ray grazing the surface has the value $\chi_c \sim \sqrt{\frac{2c_S\varepsilon H}{c_H}} \sim \sqrt{2\varepsilon H}$. For $\chi < \chi_c$, the expression within the square root in (4.25) is negative and D is given by the first term. For $\chi > \chi_c$, the second term in (4.25) leads to a decrease in the cycle distance.

The number of modes M, corresponding to rays not touching the surface in a channel with a linear sound speed profile, can be coarsely estimated based on the condition that for the turning point of this ray (modes with index greater than M) $\underline{z} = 0$. A semiclassical quantization condition for such modes is described by the expression $\int_0^H \sqrt{k_S^2(1 + 2\varepsilon z) - q_M^2}dz \approx M\pi$ (where $k_S = 2\pi f/c_S$). Calculating the elementary integral, and given that at the turning points $\underline{z} = 0$ and $k_S^2 - q_M^2 = 0$, we obtain

$$M \approx \frac{kH}{\pi}\sqrt{\frac{\Delta c}{c_S}} \sim \frac{kH}{\pi}\sqrt{\varepsilon H}, \tag{4.26}$$

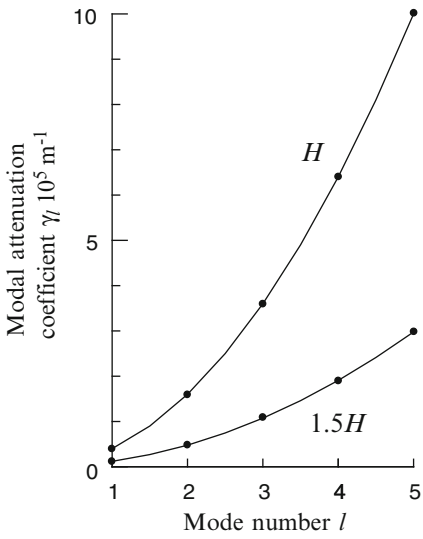


Fig. 4.2 Dependence of γ_l on mode index and grazing angle for the Pekeris model. $s = 0.1$, $H = 100$ m, $f = 100$ Hz, $c = 1{,}500$ m/s; depth values are shown in the figure

where $\Delta c = (c_S - c_H) = c_S \varepsilon H$ is the drop in the sound speed between the bottom and the surface. This estimate can be used for an arbitrary sound speed profile with a negative gradient.

Using expression (4.25), it is easy to show that the attenuation coefficients of bottom rays have the values

$$\gamma = \frac{s c_S \varepsilon}{c_H} \sim s\varepsilon \tag{4.27}$$

($\chi < \sqrt{2\varepsilon H}$) and are only weakly dependent on grazing angle. For bottom-surface rays ($\chi > \sqrt{2\varepsilon H}$), γ grows as a function of grazing angle.

We see from (4.27) that in shallow water areas with different negative gradients, we should expect different transmission losses. For example, the sound speed profiles in three shallow water areas: the Atlantic shelf of the USA (area of the SWARM'95 and SW06 experiments), the Barents sea, and the South China Sea (Asiaex experiment), can be roughly described by the gradients $\varepsilon_A \sim 5 \times 10^{-4}\,\mathrm{m}^{-1}$, $\varepsilon_B \sim 8 \times 10^{-5}\,\mathrm{m}^{-1}$, and $\varepsilon_C \sim 7 \times 10^{-4}\,\mathrm{m}^{-1}$; so attenuation in these areas should differ significantly in character (different bottom properties can also modify this simple ratio). Correspondingly, typical ray cycle distances for the first area are estimated as 1,200 m, where the Barents Sea area is about 4,000 m.

Different features of the attenuation coefficients are illustrated in Figs. 4.2–4.5. Figure 4.2 shows the behavior of γ as a function of mode index for the Pekeris waveguide. For a channel with a linear dependence of $c(z)$, the behavior of the attenuation coefficient is shown in Fig. 4.3, and discrete values of γ_l corresponding to individual modes can be obtained based on the quantization rules.

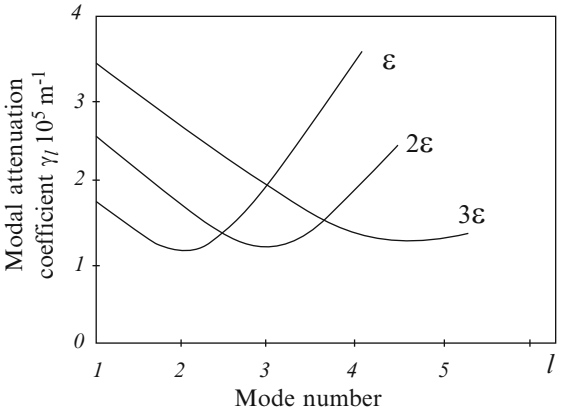


Fig. 4.3 Dependence of γ_l on mode index for a waveguide with sound speed linearly decreasing with depth (negative gradient: $dc/dz < 0$) for various values of the gradient, shown on the curves $s = 0.1$, $H = 100$ m, $f = 300$ Hz, $c_H = 1{,}500$ m/s; $\varepsilon = 6 \times 10^{-5}$ m^{-1}

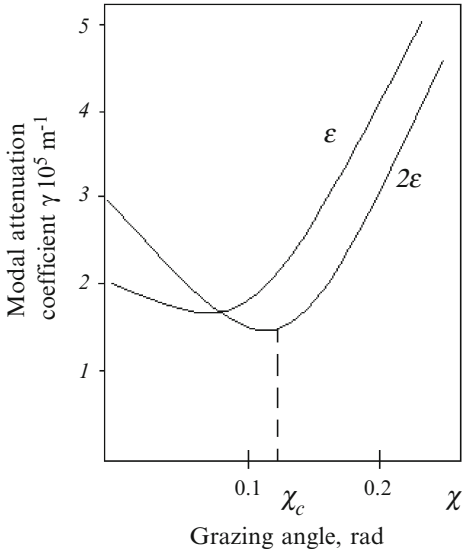


Fig. 4.4 Dependence of γ on grazing angle χ, calculated via (4.20) for a waveguide with a sound speed profile with a negative gradient for various values of gradient (shown in the figure). $s = 0.1$, $H = 100$ m, $f = 300$ Hz, $c_H = 1{,}500$ m/s; $\varepsilon = 6 \times 10^{-5}$ m^{-1}

One can also consider the dependence of the quantity γ on the value of the sound speed gradient. This dependence is shown in Fig. 4.4, from which it is clear that the attenuation coefficients of the individual modes at first decrease comparatively sharply down to some minimum value and then grow proportionally with the gradient of the sound speed according to (4.23). The decrease in the attenuation coefficient γ takes place at different times and nonuniformly for the different modes. For each fixed index l (or grazing angle χ of the equivalent ray), there is a value of gradient ε_l where the decay of the given mode is minimal. Consideration of the gradient ε_l gives a value

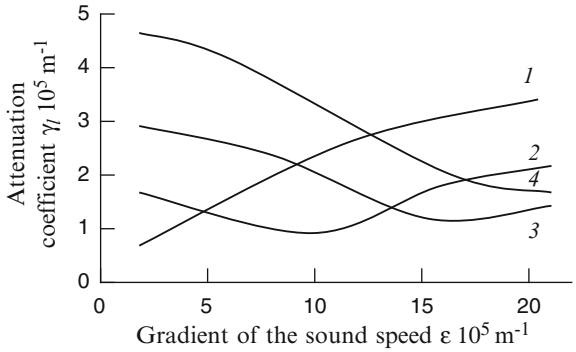


Fig. 4.5 Dependence of γ_l on the gradient ε of the sound speed for the first four modes. $s = 0.1$, $H = 100$ m, $f = 300$ Hz, $c_H = 1{,}500$ m/s

$\varepsilon_l \sim \chi_l^2/(2H)$, i.e., at the point where the given ray grazes the surface. Accordingly, if we fix the value of the gradient, then for a given ε, there are one or more modes whose decay is less than others, and which can have very high indices. Rays corresponding to these modes graze the surface or approach very close to it.

For a layer with linear $c(z)$, the indices of the modes corresponding to given angles can be estimated from the quantization rules. Assuming that the grazing angles are small, the estimate has the form

$$\chi_l \approx (3c_S\varepsilon l/2f)^{1/3}. \tag{4.28}$$

Hence, the index of the mode with the least decay can be estimated as

$$\bar{l} \approx \frac{2f}{3c_S\varepsilon}\left(\frac{2c_S\varepsilon H}{c_H}\right)^{3/2} \sim \frac{2fH\sqrt{\varepsilon H}}{c_H}. \tag{4.29}$$

This expression agrees with (4.26), which means that the least decay in this waveguide occurs for the near-bottom mode with the largest index, which is the mode which grazes the sea surface. For example, if $H \sim 120$ m, $c_H = 1{,}470$ m/s, $\varepsilon = 8 \times 10^{-5}$ 1/m (approximate Barents Sea parameters), then at the frequency $f \sim 200$ Hz, we have $l \sim 3 - 4$, i.e., there are approximately three modes which give roughly identical contributions to the far field. It is interesting to note that for the Atlantic shelf area $H \sim 80$ m, $c_H = 1{,}500$ m/s, and $\varepsilon = 5 \times 10^{-4}$, we have approximately the same number of energy-carrying modes for the same frequency (this is in accordance with experimental observation, see Sect. 4.4). However, in the China Sea ($H \sim 40$ m, $c_H = 1{,}500$ m/s, $\varepsilon = 7 \times 10^{-4}$ 1/m), there is no more than 1 mode, propagating at 200 Hz frequency.

Figure 4.5 shows the results of a calculation of the mode attenuation coefficients (imaginary parts of the eigenvalues) obtained via a precise solution of the Sturm–Liouville problem (3.25) with a linear sound speed profile in the water

layer $c(z) = c_S(1 - \varepsilon z)$. The bottom is assumed to be a fluid, homogeneous half space. Results obtained from this numerical calculation agree well with the approximate estimates, except for a small range of values of the parameters ε (individually for each mode) where γ is minimal. This region corresponds to values of ε for which the ray grazes the surface and thus where ray theory does not strictly apply.

In concluding this section, let us consider what the value of the horizontal displacement of a ray Δ might be in real conditions. Assuming small grazing angles for the case of homogeneous layers of water and sediment, the following estimate for Δ is easily obtained:

$$\Delta \approx 2m \Big/ \left(\chi \sqrt{k^2 - k_1^2} \right). \tag{4.30}$$

Let us take the Pekeris model with the values $c = 1{,}500$ m/s, $c_1 = 1{,}600$ m/s, $f \approx 250$ Hz, $m_1 \approx 2$, and $H = 150$ m. Then for an angle $\chi = 0.02$ radian, corresponding to the lowest angle for a waveguide mode, the displacement of the ray upon reflection is in the order of $\Delta \sim 40$ m, whereas the cycle distance of this ray is equal to approximately 7.5 km. It is clear that in this case, the displacement of the ray can be neglected.

If there is a small negative sound speed gradient in the water – for example, if $\varepsilon \sim 8 \times 10^{-5}$, then the angle corresponding to the fundamental mode is $\chi \approx 0.05$ and thus $\Delta \sim 10$ m. The ray cycle distance for the channel with depth ~120 m under these conditions is $D \approx 4$ km. Consequently, the value of Δ is again significantly less than D, and can be neglected.

The beam displacement is more pronounced if there is a layer of sediments with a positive sound speed gradient $c_1\varepsilon_1$. The smaller this gradient, the larger the beam displacement $\Delta \sim 2\chi/\varepsilon_1$. In this case, the correction to the cycle distance D and correspondingly to the mode attenuation coefficient γ is somewhat greater than for an homogeneous bottom, but still remains small in absolute value, since gradients in the bottom are typically at least an order of magnitude greater than the value of the gradient in the water layer.

The theory studied in this section enables us to estimate the transmission loss of sound in shallow water due to interaction with the bottom. For example, for typical values of bottom parameters $m_1 \approx 2$, $c/c_1 \approx 0.83$, $\alpha \approx 0.02$, for a depth $H \approx 100$m, at a frequency $f \approx 100$ Hz, and for a value of the mode attenuation coefficient (4.24) for the fundamental mode $l^* \approx 1$, we obtain $\gamma_1 \approx 6.4 \times 10^{-3}$ 1/km. If we then take into account the fact that at a large distance from the source, because of the difference in the attenuation coefficients, just one fundamental mode remains (in the Pekeris model), then the overall decay of the observed field corresponds to the decay of this mode. In this case, we are examining $\beta = 10 \lg \gamma_1 \approx 0.03$dB/km. Similarly, one can estimate the law for transmission loss in a waveguide with a negative sound speed gradient (a very common case). Here, at large distances, a group of approximately identically decaying modes remains, with attenuation coefficients determined by (4.27). For the earlier example, with sound speed gradient $\varepsilon \approx 2 \times 10^{-4}$ 1/m, we obtain $\gamma_1 \sim 10^{-2}$ 1/km or

$\beta \sim 0.04$ dB/km. This value is in complete agreement with results of experiments in shallow water regions. We also mention that the analogous value for a deep water underwater sound channel is considerably less, $\beta \approx 0.003-0.007$. dB/km, and the absorption of sound by the water makes the dominant contribution to this decay. Thus, the absorption of sound by water in shallow water channels can legitimately be neglected.

4.2 Simplified Description of Solid, Multicomponent, Poro-Elastic, and Layered Bottom Models

In the previous sections, we considered the construction of the sound field due to a point source on the assumption that the bottom is modeled by a fluid, homogeneous, absorptive half-space. A more realistic and more complex model for bottom is the solid bottom model, which we mentioned in Chap. 2. This model can be used for both "hard" elastic solids, such as rocks, and two-phase media, such as water plus sediment. However, in the latter case, more sophisticated and complete descriptions should also be examined [Biot-Stoll (Stoll 1989)].

Let us now consider the solid bottom model. Within the framework of this approach, the bottom is characterized by density, compressibility, and shear modulus. This means that in the bottom, two types of waves can propagate – both longitudinal and shear waves, with the velocities c_l, c_s correspondingly. Typical values of bottom parameters, including shear, are shown in Sect. 2.11.

For this case, the boundary conditions are changed; more exactly, we have another function $g(\xi)$ and another reflection coefficient to consider. From our table, we see that as a rule for sediments (thus excluding the very rigid basalt material), we have a relationship between speeds $c_s < c < c_l$. This means that in accordance with Snell's law (see Fig. 4.6),

$$\frac{\cos\chi}{c} = \frac{\cos\chi_l}{c_l} = \frac{\cos\chi_s}{c_s}; \tag{4.31}$$

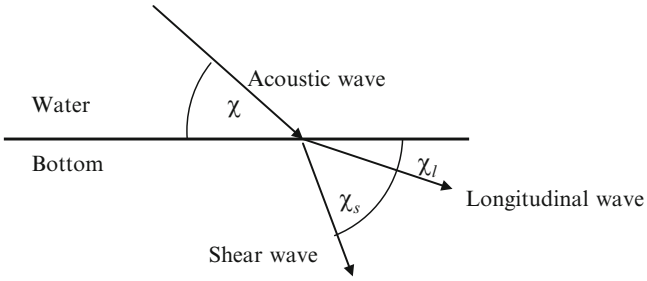


Fig. 4.6 Diagram of rays on the boundary between the water and a solid bottom

thus for the wave vectors, $k_l<q<k<k_s$ for small grazing angles of the sound rays in water. This means that for shear waves, $\chi_s>\chi$ and there is no total internal reflection. Physically, this implies that for any incident sound waves in the water, shear waves will carry energy away from the boundary. This provides attenuation (loss of energy) even if $\alpha = 0$, and the greater the speed of the shear waves, the more the loss. For our work, the function $g(\xi)$ can be calculated based on the theory of waves in elastic media (Brekhovskikh 1980):

$$g(\xi) = \frac{m_1}{k_s^4}\left[\frac{\left(2\xi^2 - k_s^2\right)^2}{\sqrt{\xi^2 - k_l^2}} - 4\xi^2\sqrt{\xi^2 - k_s^2}\right]. \tag{4.32}$$

The limit of a liquid bottom follows by letting $k_s \to \infty$, and in this limit, (4.32) reverts to (3.8).

Let us find the dependence of the modulus of the reflection coefficient on the grazing angle (in mode language, on the longitudinal wave number). In Fig. 4.7, we can see that for realistic parameters, the modulus of the reflection coefficient is practically constant in the area beyond the total reflection angle. In the region of small angles (total internal reflection for a liquid bottom), shear waves give a small contribution to the coefficient of attenuation. Indeed, in a simplified picture, we could introduce an effective coefficient of attenuation including both mechanisms: the real attenuation (α) and the shear wave attenuation. At the same time, we should also take into account the frequency dependence of this coefficient. Within the framework of a simplified approach to the modal attenuation coefficient, we should expand the reflection coefficient in a series for small grazing angles, i.e.,

$$V = -1 + \frac{dV}{d\chi}\Big|_{\chi=0}\chi = -1 + V_0{}'\chi. \tag{4.33}$$

Using the expression for the reflection coefficient, we get

$$V_0' = \frac{2i(\sigma g)'}{(\sigma g + i)^2}\Big|_{\chi=0} = -2i\sigma_0' g\big|_{\chi=0} = -2ikg(k). \tag{4.34}$$

If we take into account the relationship between the reflection coefficient and the modal attenuation coefficient, we see that

$$\gamma = -2\frac{\ln|V|}{D} \approx \frac{2s\chi}{D}, \tag{4.35}$$

where we define

$$s = -\frac{d\ln|V|}{d\chi}\Big|_{\chi=0}. \tag{4.36}$$

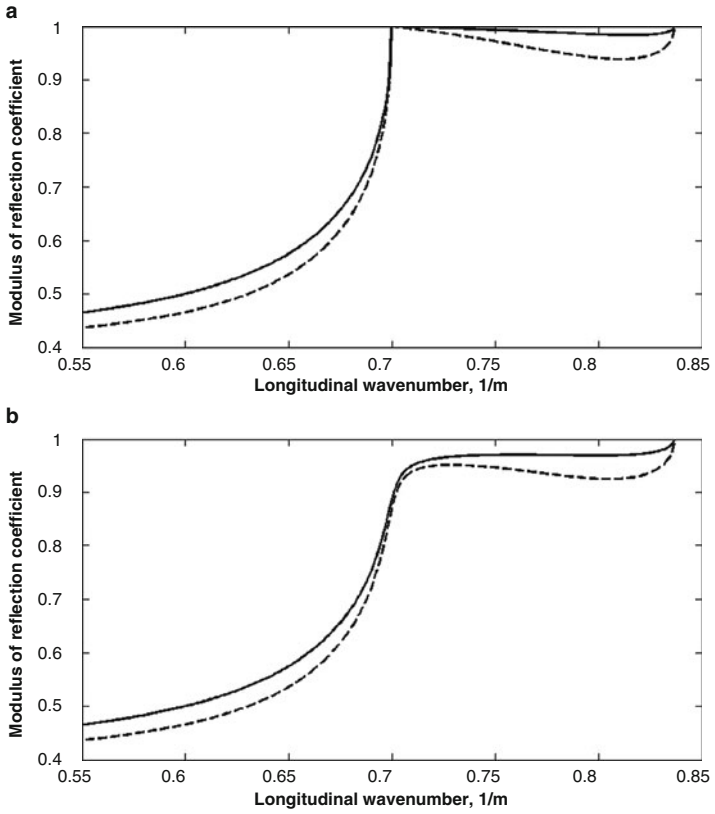


Fig. 4.7 Modulus of reflection coefficient for model with parameters $m_1 = 2$, $c = 1{,}500$ m/s, $c_l = 1{,}800$ m/s, frequency 200 Hz, speed of shear waves $c_s = 300$ m/s (*continuous line*) and 400 m/s (*dashed line*), (**a**) attenuation coefficient of longitudinal waves $\alpha = 0$, (**b**) $\alpha = 0.01$

In our simplified form, we can separate the dependence on bottom parameters and mode number. Taking the derivative, and using

$$\sqrt{q^2 - k_1^2} = -i\sqrt{k_1^2 - q^2},$$

we get

$$s = \mathrm{Im}[2kg(k)]. \tag{4.37}$$

We see that even in the absence of attenuation, for the case of a solid bottom, we have loss due to shear waves carrying energy into the bottom. We can now estimate the parameters according to (4.32) as

$$g(k) = \frac{m_1}{k_s^4}\left[\frac{(2k^2 - k_s^2)^2}{\sqrt{k^2 - k_1^2}} - 4k^2\sqrt{k^2 - k_s^2}\right] \tag{4.38}$$

and from (4.37),

$$s = \mathrm{Im}[2kg(k)] = \frac{8m_1k^3}{k_s^4}\sqrt{k_s^2 - k^2} \sim \frac{8m_1k^3}{k_s^3} = \frac{8m_1c_s^3}{c^3}. \tag{4.39}$$

In this approximation, s does not depend on frequency and for our values of parameters, $s \sim 0.128$.

In the simplest approach, we can use the model of a liquid bottom, but supposing that coefficient of loss now includes the generation of shear waves at the boundary. The frequency dependence of the attenuation remains the same as the imaginary part of the wave number in the bottom. Not only a little more complex (but also a simplified) approach is seen in the work Tindle and Zhang (1992), where the authors propose a "liquid equivalent bottom" with effective parameters of density and sound speed, which provides the same structure of the sound field in the waveguide. In order to construct an "equivalent liquid bottom," it is necessary to change the density and the attenuation bottom coefficient. The weak point of this approach (in our opinion) is the dependence of the attenuation coefficient on the water layer parameters, along with the bottom parameters and frequency. A more elaborate approach to constructing an equivalent model of the bottom introduces the concept of complex density, thus providing a more precise sound field in the water layer. As to numerical values, for more or less real conditions, it is possible to use the equivalent bottom concept if the shear wave speed is less than 600 m/s. In particular, we can see from our table that for aleurit-clay and coarse sand sediments we could use simple, effective liquid bottom, but for the rigid basalt bottom, we must use the solid model.

If we take into account both loss mechanisms, then the parameter s is given by the relationship

$$s = 8m_1(c_s/c)^3\sqrt{1-(c_s/c)^2} + m_1\alpha(c/c_1)^2\frac{1-2(c_s/c)^2}{\left[1-(c/c_1)^2\right]^{\frac{3}{2}}} \sim \frac{8m_1}{n_s^3} + \frac{m_1\alpha n_1^2}{\left(1-n_1^2\right)^{\frac{3}{2}}}. \tag{4.40}$$

We can see that if $c_s \to 0$, we get the result (4.23). The above theory allows us to use the model of a liquid bottom with effective parameters in many cases.

Both the liquid and solid bottom models usually suppose that the bottom is a homogeneous medium. In practice, this means that parameters of bottom models are considered as averaged along comparatively long acoustic tracks. If, however, we are dealing with a comparatively small region for which a more or less regular (along the track) layer structure of the bottom can be established, the model can be made more complicated. Specifically, we can use the assumption that the bottom consists of a given number of fluid/elastic homogeneous layers lying one above the other. This model often corresponds to the experimental observations of bottom structure.

Let us now consider theoretically the model of a bottom consisting of fluid/solid homogeneous absorptive layers. The spatial region $z>H$ is divided up by planes $z_n = H_n = H + h_n$, where h_n is the thickness of the layer with index n. We suppose that there are N layers, each of which has density ρ_n, sound speed c_n, and attenuation coefficient α_n (and the additional shear parameters for solid layers). In this case, the boundary conditions for the normal mode with index l in the water layer have the same form as in (3.7c):

$$\Psi_1(H) + g_1(\xi)\frac{d\Psi_1}{dz}|_{z=H} = 0. \tag{4.41}$$

In the above, the value of $g_1(\xi)$ is determined from a recursive system of sequentially decreasing indices, which is derived by matching the values $\rho_n \frac{\Psi_l}{\Psi_l'}$ at the boundary layers $n-1$ and n:

$$g_n = \frac{\rho_n}{\rho_{n-1}\kappa^{(n)}}\frac{\kappa^{(n)}g_{n+1} - \tanh\kappa^{(n)}h_n}{1 - \kappa^{(n)}g_{n+1}\tanh\kappa^{(n)}h_n}, \quad g_{N+1} = \frac{\rho_{N+1}}{\rho_N\delta_{N+1}}, \tag{4.42}$$

where $\kappa^{(n)} = \sqrt{\xi^2 - k_n^2}$, which is complex valued because k_l is complex. Note that in what follows we shall also use the quantity $\sigma^{(n)} = \sqrt{k_n^2 - \xi^2} = i\kappa^{(n)}$. When $N = 0$, we obtain the case of a homogeneous, fluid bottom, which was considered earlier. An analogous but more cumbersome relation is obtained for boundary conditions in the case of solid layers.

A detailed numerical analysis of the behavior of the characteristics of the waveguide modes in a layered waveguide was given in Ageeva and Krupin (1980; 1981) in which the effect of various factors were considered. We further note that generally speaking, the bottom layer (layers) may be inhomogeneous with depth, and large proportions may have a positive gradient of the sound speed. However, the model considered here may be used in this case, by splitting the inhomogeneous layer into a certain number of homogeneous layers, as has been done in a number of works (Ageeva et al. 1994; Zhuravlev et al. 1986; Porter and Reiss 1984, 1985; Porter 1992). Sometimes an inhomogeneous layer can be considered by approximating the dependence of the sound speed on z, for example, by a linear function, and then constructing the solution in this layer via Airy functions. Finally, the function $g(\xi)$ can be constructed numerically for even more complex bottom models.

4.3 Optimum Frequency

It was D. Weston who first reported the so-called optimum frequency in sound propagation in shallow water (1976). He did this on the basis of experiments with broadband signals (25–6,400 Hz) in the North Sea at a depth of 70 m. In his case, minimal losses were observed at the frequency ~200 Hz. Analogous results were

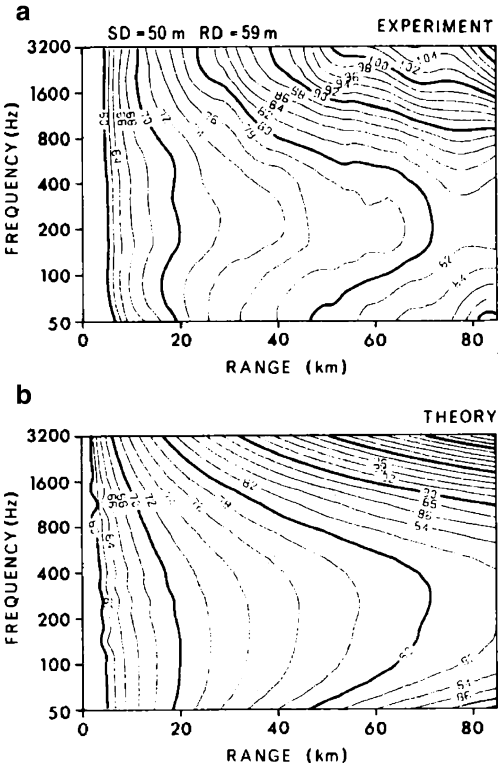


Fig. 4.8 Curves of constant loss for experimental conditions in a particular shallow water region (Jensen and Kuperman 1983) demonstrating the existence of an optimal frequency in sound propagation (**a**) experiment, (**b**) theory

obtained by him in further experiments in the Bristol Channel. According to Weston, shallow water regions can be divided into two types, with different loss at low frequencies but more or less the same loss for high frequencies. Areas with high losses are characterized by rocky bottoms, whereas low losses are typical for sandy or muddy bottoms.

Kibblewhite and Denham (1966) measured losses in sound propagation in an area off New Zealand at a depth of about 70 m. The bottom was almost flat and effectively liquid. Experiments were carried out in winter (with an almost constant sound speed profile) and in summer (with a negative gradient of the water sound speed profile). In the first case, the minimal loss took place in the frequency band 50–150 Hz. Approximately the same situation (but with higher loss overall) took place in summer.

Jensen and Kuperman (1983) have also explored the existence of an optimal frequency in shallow water sound propagation. In Fig. 4.8a, b, we see the results of experiment (a) and theoretical calculations (b), illustrating the existence of an optimal frequency, which for the waveguide investigated is about 300 Hz.

According to Aikal (1972), the optimal frequency depends upon the type of sediment; for example, sand gives an optimal frequency in the band 400–800 Hz, whereas for muddy (clay) bottoms, this frequency is found in the band 50–100 Hz.

Gershfeld and Eller (1985) estimated the optimum frequency by using a simple waveguide model. We will carry out estimates of the existence and value of the optimum frequency for the same simple model, also taking into account bottom shear waves (see Sect. 4.2). The frequency dependence of the modal attenuation coefficient is determined by waveguide properties and, most importantly, by the sound speed profile. For the Pekeris waveguide, these coefficients are given in (3.46) and can be written as a function of frequency as

$$\gamma_1 = \frac{sc^2l^2}{4f^2H^3}. \tag{4.43}$$

Here, the parameter s can include both attenuation in the sediments and the influence of shear waves [see (4.39)]. If we also take into account the loss in the water layer, the total loss has the form

$$\gamma_1 = \frac{sc^2l^2}{4f^2H^3} + \frac{0.013(f/1,000)^2}{1+(f/1,000)^2}. \tag{4.44}$$

In this case, there exists a frequency for minimal loss for a given mode number. It can be simply estimated as

$$f_{\text{opt}} \sim 100\left(\frac{sc^2l^2}{5.2H^3}\right)^{1/4}. \tag{4.45}$$

The optimal frequency increases with the mode number; however, that is an asymptotic value, since frequency dependence is also contained in the mode excitation coefficients (if we deal with a point source). The total intensity of the sound field is summed from the modal intensities, which have the form

$$|P_l| = \sqrt{\rho_0 c_0 W_0}\,\frac{|\Psi_l(z_0)\Psi_l(z)|}{\sqrt{q_l r}}\exp[-(\gamma_l/2+\beta)r]. \tag{4.46}$$

For comparatively close ranges, the eigenvalue as a function of frequency plays the main role and is in the denominator. For long distances, the exponential factor predominates. In Fig. 4.9a, b, we show experimental and theoretically examples of transmission loss for the two lowest modes, treatable within the framework of the Pekeris model.

In a waveguide with a real sound speed profile, the optimal frequency will be a little stretched. In particular, for sound speed profiles with a negative gradient, the modal attenuation coefficient does not depend on frequency ($\gamma \approx s\varepsilon$).

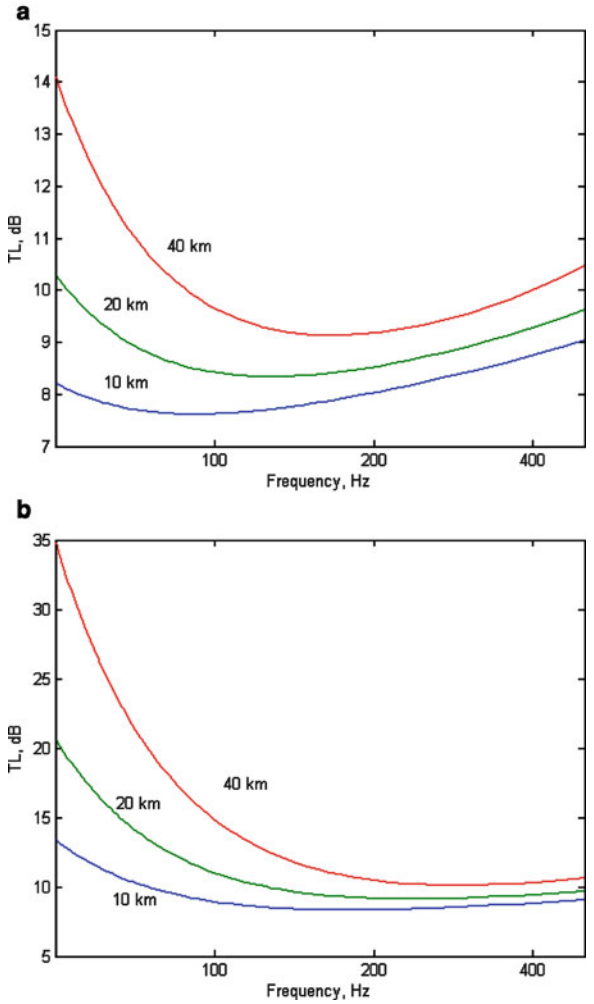


Fig. 4.9 Transmission loss for the first (**a**) and second (**b**) modes. Parameters $c = 1.5$ km/s, $c_1 = 1.7$ km/s, $c_s = 0.2$ km/s, $\alpha = 0.01$, $H = 100$ m

Approximately the same situation takes place for waveguide modes (rays) that do not touch the surface (near-bottom rays). As frequency decreases, the mode transforms into surface reflected and the loss increases with the decreasing frequency, so we have a new band with minimal loss. The optimal frequency, as well as the loss in the frequency domain for different ranges, also depends on other waveguide parameters: depth, sound speed profile, etc. Additionally, as a result of the mode summation (each mode of which has its own optimal frequency), some broadening of the frequency curve takes place.

4.4 Interference Structure and Interference Invariant of the Sound Field

In this section, we introduce the concept of the interference invariant of the sound field (also often called the "wavedguide invariant") (Chuprov 1982; Zverev and Orlov 1984; Kuperman and Lynch 2004). We discuss this concept briefly using simple examples.

As was shown, the acoustic field of a CW source in a SW waveguide has the form

$$P(\vec{r},z,t)=\sqrt{\rho_0 c_0 W_0}\sum_l C_l\frac{\Psi_l(z)}{\sqrt{q_l r}}\exp[i(q_l r-\omega t)] \tag{4.47}$$

or in short form, for the time independent amplitude only

$$P(\vec{r},z)=\sum_l \frac{P_l}{\sqrt{q_l r}}\exp[iq_l r], \tag{4.48}$$

where

$$P_l=\sqrt{\rho_0 c_0 W_0}C_l\Psi_l(z).$$

The complex sound field can be written in the polar form

$$P(\vec{r},z)=|P|\exp(i\varphi) \tag{4.49}$$

$$|P(\vec{r},z)|=\sqrt{\left[\sum_l \frac{P_l}{\sqrt{q_l r}}\cos(q_l r)\right]^2+\left[\sum_l \frac{P_l}{\sqrt{q_l r}}\sin(q_l r)\right]^2},$$

$$\varphi(r,z)=\arctan\left[\frac{\sum_l \frac{P_l}{\sqrt{q_l r}}\sin(q_l r)}{\sum_l \frac{P_l}{\sqrt{q_l r}}\cos(q_l r)}\right].$$

The intensity of the sound field, using (4.48), is thus

$$I=\frac{1}{2\rho c r}\sum_{l,m}\frac{P_l P_m}{\sqrt{q_l q_m}}\cos(\Delta q_{lm} r). \tag{4.50}$$

If we take into account that $\Lambda_{lm}=2\pi/|\Delta q_{lm}|$ and that cos is an even function,

$$I=\frac{1}{2\rho c r}\sum_{l,m}\frac{P_l P_m}{\sqrt{q_l q_m}}\cos\left(2\pi\frac{r}{\Lambda_{lm}}\right). \tag{4.51}$$

We now consider the sound field at comparatively long distances from the source, more exactly for $r \gg \Lambda_{lm}$, where a stable interference pattern is formed. As seen from (4.50) and (4.51), the properties of the interference structure of the sound field are determined by the values of Λ_{lm}.

We can estimate the interference beating scale for the Pekeris waveguide using (3.34). We obtain

$$\Delta q_{lm} = q_l - q_m = \sqrt{k^2 - \left(\frac{l^*\pi}{H}\right)^2} - \sqrt{k^2 - \left(\frac{m^*\pi}{H}\right)^2} = \frac{\pi^2}{H^2}\frac{(m^*)^2 - (l^*)^2}{q_l + q_m} \approx k\left(\frac{\pi}{kH}\right)^2 \frac{(m^*)^2 - (l^*)^2)}{2}.$$

For example, if $f = 100$ Hz and $H = 100$ m, we have $q_{lm} \sim 10^{-3}((m^*)^2 - (l^*)^2)$ m.

It is clear that in typical situations, for shallow water acoustics $r \gg \Lambda_{lm}$. We see also that the value Δq_{lm} can be written in the form

$$\Delta q_{lm} \approx \alpha_{lm} c H^{-2} f^{-1}, \tag{4.52}$$

where we define $\alpha_{lm} = \pi((m^*)^2 - (l^*)^2)/4$.

Let us consider this value for a near-bottom channel, corresponding to the sound speed profile linearly changing with depth as $c(z) = c_s(1 - \varepsilon z)$ or

$$k^2(z) = k_s^2(1 + 2\varepsilon z). \tag{4.53}$$

Modal eigenvalues can be obtained using the WKB method, i.e.,

$$\int_{\underline{z}}^{H} \sqrt{k_s^2(1 + 2\varepsilon z) - q_l^2}\, dz = l^*\pi \tag{4.54}$$

so that

$$q_l^2 = k_H^2 - \left(3k_s^2 \varepsilon l^* \pi\right)^{2/3} \tag{4.55}$$

and

$$\Delta q_{lm} = q_l - q_m \approx \left(9\varepsilon^2 k \pi^2\right)^{1/3} \left[(l^*)^{2/3} - (m^*)^{2/3}\right]. \tag{4.56}$$

In this case, we also have a relationship which is similar to (4.52)

$$\Delta q_{lm} \approx \alpha_{lm} \varepsilon^{2/3} f^{1/3}, \tag{4.57}$$

where we now define $\alpha_{lm} = (18\pi^3 c)^{1/3}[(l^*)^{2/3} - (m^*)^{2/3}]$.

It is easy to see that for both waveguides, the following relationship is true

$$\frac{\partial \Delta q_{lm}}{\partial \omega} = -\frac{1}{\beta_0}\frac{\Delta q_{lm}}{\omega}, \tag{4.58}$$

where we define the β_0 parameter as

$$\beta_0 = -\frac{\Delta q_{lm}}{\omega}\Big/\frac{\partial \Delta q_{lm}}{\partial \omega} = -\Delta s^{\mathrm{ph}}_{lm}/\Delta s^{\mathrm{g}}_{lm}. \tag{4.59}$$

In the above, the phase and group slowness (1/velocity) are denoted by $s^{\mathrm{ph}}_{lm,} = 1/v^{\mathrm{ph}}_{lm,}$ $s^{\mathrm{g}}_{lm} = 1/v^{\mathrm{g}}_{lm,}$ v^{ph}_{lm} and v^{g}_{lm} are the phase and group velocities of the modes with the given mode numbers. For the Pekeris model and for a near-bottom channel, β_0 has the values $\beta_0 = 1$, $\beta_0 = -1/3$, respectively.

Generally speaking, not only is relationship (4.59) true for any group of waveguide modes that have dispersion dependences "of the same type," but it can also be seen from (4.58) and (4.59) that each group has a β_0 value of its own. It is this value that is called the space-frequency interference invariant of the sound speed. To explain the physical meaning of this invariant, let us calculate the following values:

$$\begin{aligned}\frac{\partial I}{\partial r} &= -\frac{1}{2r}\sum_{l,m}\frac{P_l P_m}{\sqrt{q_l q_m}}[\Delta q_{lm}\sin(\Delta q_{lm} r) + (1/r)\cos(\Delta q_{lm} r)] \\ &\approx -\frac{1}{2r}\sum_{l,m}\frac{P_l P_m}{\sqrt{q_l q_m}}\Delta q_{lm}\sin(\Delta q_{lm} r) = \frac{\omega}{2r}\sum_{l,m}\frac{P_l P_m}{\sqrt{q_l q_m}}\Delta s^{\mathrm{ph}}_{lm}\sin(\Delta q_{lm} r),\end{aligned} \tag{4.60}$$

$$\begin{aligned}\frac{\partial I}{\partial \omega} &= -\frac{1}{2r}\sum_{l,m}\frac{P_l P_m}{\sqrt{q_l q_m}}\left[r\frac{\partial \Delta q_{lm}}{\partial \omega}\sin(\Delta q_{lm} r) + (1/2q_l q_m)\frac{\partial (q_l q_m)}{\partial \omega}\cos(\Delta q_{lm} r)\right] \\ &\approx -\frac{1}{2r}\sum_{l,m}\frac{P_l P_m}{\sqrt{q_l q_m}} r\frac{\partial \Delta q_{lm}}{\partial \omega}\sin(\Delta q_{lm} r) = \frac{1}{2}\sum_{l,m}\frac{P_l P_m}{\sqrt{q_l q_m}}\Delta s^{\mathrm{gr}}_{lm}\sin(\Delta q_{lm} r).\end{aligned} \tag{4.61}$$

From relationship (4.59), we have the following equality:

$$r\frac{\partial I}{\partial r} \approx -\beta_0 \omega \frac{\partial I}{\partial \omega}. \tag{4.62}$$

We can interpret this relationship in the following way. Let us denote intensity as a function of range and frequency as $I(r, \omega)$. If we are at an area (line) where $I = \mathrm{const}$, then for any point with coordinates (r_0, ω_0) we have

$$\frac{\partial I}{\partial r}\delta r + \frac{\partial I}{\partial \omega}\delta \omega = 0, \tag{4.63}$$

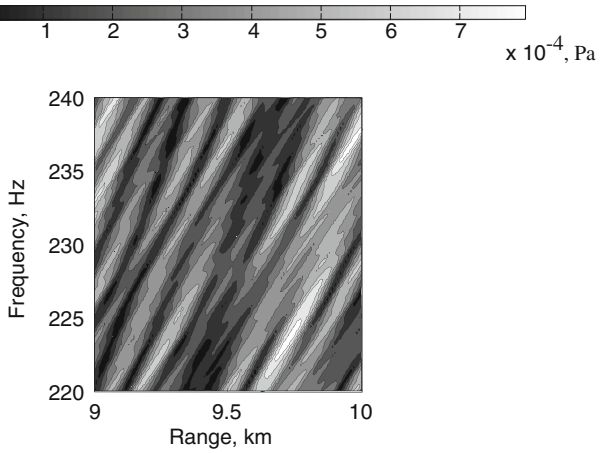


Fig. 4.10 Brightness values of the sound field amplitude

where $\delta\omega = \omega - \omega_0$ and $\delta r = r - r_0$ are increments of frequency and distance. Taking into account our previous relationships, we obtain

$$\tan\chi_0 = \frac{\delta\omega}{\delta \mathrm{r}} = \beta_0 \frac{\omega_0}{\mathrm{r}_0}. \tag{4.64}$$

The value $\delta\omega/\delta r$ is the slope of the line corresponding to a fixed level of the sound field in the (ω, r) coordinate space. The space-frequency invariant of the sound field defines the angle of inclination χ_0 of this line. Equation (4.64) also finds practical use in the calculation of broadband transmission loss, as will be discussed in a later section.

It is significant and useful that the interference pattern of the sound field exhibits a striation structure in the (ω, r) plane. The interference invariant β_0 is one of the main characteristics of this structure. In a regular waveguide with a fixed number of modes, the value β_0 does not depend on distance r, frequency ω and depth z. It is in this respect that β_0 is an invariant of the sound field in the waveguide.

An example of an interference pattern for the SW sound field ("brightness" values of the sound field amplitude $|P(r,\omega)|$, $|P(r,\omega)| = \sqrt{2\rho cI}$) is shown in Fig. 4.10. For this example, the waveguide depth is equal to $H = 72$ m; the vertical profile of sound speed is plotted in Fig. 4.11. [This sound speed profile was recorded in the experiment by Gasparovic and Etkin (1994a).] The bottom parameters are the following: density $\rho = 1.8$ g/cm^3 and bottom refractive index $n_1 = 0.84(1 + 0.007i)$. The source and receiver depths are equal to 50 and 60 m, respectively. The image shown in Fig. 4.10 is sampled every 25 m in range and every 0.5 Hz in frequency. As seen from Fig. 4.10, two striation structures are observed in this case. One of them corresponds to the bottom modes and the other to the bottom-surface modes. Each group of modes has a dispersion dependence of its own.

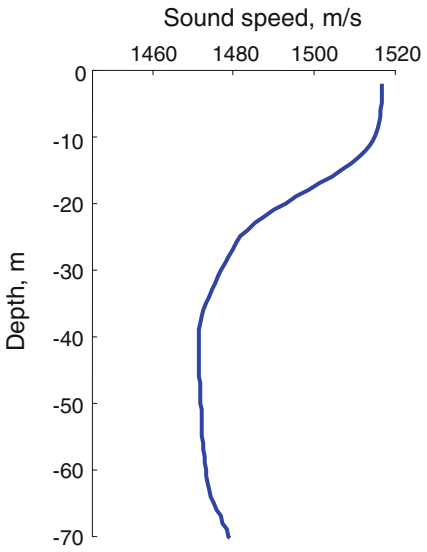


Fig. 4.11 Vertical profile of sound speed in the Middle Atlantic Bight

It is important to note that the value β_0 is generally calculated by using the interference pattern of the sound field. To do this, it is necessary to calculate the directivity function of the corresponding brightness image by using the approach[2] seen in Chap. 2 for processing the surface manifestation of IW. In implementing this approach, the spectrum $\tilde{w}(\rho\cos\chi, \rho\sin\chi)$ is first calculated in polar coordinates:

$$\tilde{w}(\rho\cos\chi, \rho\sin\chi) \equiv \tilde{w}(u,\tau) = \int_0^{\Delta R}\int_0^{\Delta\Omega} w(r,\omega)\exp[\mathrm{i}(ur+\tau\omega)]drd\omega, \tag{4.65}$$

where $w(r,\omega) = |p(r,\omega)| - \overline{|p(r,\omega)|}$, and $\overline{|p(r,\omega)|}$ is the mean amplitude of the sound field, averaged over the spatial and frequency beating bands. Correspondingly, the directivity function is equal to

$$\tilde{\Theta}[\chi(\beta)] = (4\pi E_0)^{-1}\int_0^{\Delta P} |\tilde{w}(\rho\cos\chi, \rho\sin\chi)|^2\rho d\rho, \tag{4.66}$$

where $\beta = (r_0/\chi_0)\tan\chi$, and $E_0 = \int_0^{2\pi}\int_0^{\Delta P} |\tilde{w}(\rho\cos\chi, \rho\sin\chi)|^2\rho d\rho d\chi$. The maximum value of (4.66) occurs when $\chi = \chi_0$, i.e., when $\beta = \beta_0$. The function $\tilde{\Theta}(\beta)$ is plotted

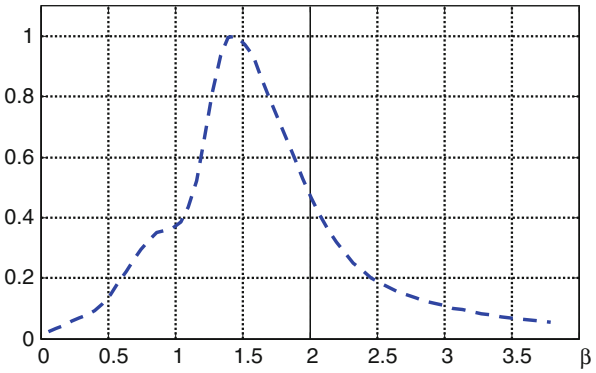


Fig. 4.12 The directivity function $\tilde{\Theta}(\beta)$ normalized to the maximum value

in Fig. 4.12 for the interference pattern shown in Fig. 4.10. As seen from the figure, the directivity function has two maxima corresponding to the bottom-surface modes and to the bottom modes. The corresponding values of β_0 are equal to 0.9 and 1.457, respectively. The maxima have different values because the bottom modes dominate in this numerical example. It is notable that for the bottom-surface mode, the value of β_0 is close to the value $\beta_0 = 1$ obtained for the Pekeris model.

We can also introduce the concept of the interference invariant with respect to other waveguide parameters, not only to the distance r (Grachev 1993). Let us consider some parameter η (for example, depth). Let us then consider a shift of a point with a constant value of intensity on the plane (ω, η). The equation we must consider has a form similar to that of our previous work, i.e.,

$$\frac{\partial I}{\partial \eta}\delta\eta + \frac{\partial I}{\partial \omega}\delta\omega = 0. \tag{4.67}$$

We have also that

$$\frac{\partial I}{\partial \eta} = -\frac{2}{2r}\sum_{l,m}\frac{P_l P_m}{\sqrt{q_l q_m}} r \frac{\partial \Delta q_{lm}}{\partial \eta}\sin(\Delta q_{lm} r); \tag{4.68}$$

now we need to establish a connection between arbitrary wave numbers as a function of frequency and the aforementioned parameter, i.e.,

$$\eta\frac{\partial \Delta q_{lm}}{\partial \eta} = -\gamma\omega\frac{\partial \Delta q_{lm}}{\partial \omega}. \tag{4.69}$$

For this case, the line of constant intensity is determined by

$$\eta\frac{\partial I}{\partial \eta} = -\gamma\omega\frac{\partial I}{\partial \omega}, \tag{4.70}$$

which results in the equation for a straight line

$$\frac{\delta\omega}{\delta\eta} = \gamma\frac{\omega}{\eta}. \tag{4.71}$$

In general, if we take the dependence of the value of Δq_{lm} as a function of frequency and our "arbitrary parameter" as

$$\Delta q_{lm}(\omega,\eta) = \alpha_{lm}\Omega(\omega)\mathrm{H}(\eta), \tag{4.72}$$

then

$$\eta\frac{\mathrm{H}'}{\mathrm{H}} = -\gamma\omega\frac{\Omega'}{\Omega} = \frac{\gamma}{\beta_0} \tag{4.73}$$

and

$$\Delta q_{lm}(\omega,\eta) = \alpha_{lm}\omega^{-1/\beta_0}\eta^{\gamma/\beta_0}, \tag{4.74}$$

where the parameter γ depends on the waveguide properties and can be considered as an interference invariant as well, see example of depth variations in (Kuz'kin et al. 1998).

4.5 Waveguide Dispersion of Sound Signals in Shallow Water

It is common knowledge, see, for example, Clay and Medwin (1977), that in clear seawater without bubbles the sound speed does not depend appreciably on frequency. Thus, when considering the propagation of sound in water, we neglect the dispersion of sound, i.e., the dependence of the sound speed c on frequency ω. This is inherently assumed in using propagation equations in the form of (A.146). However, waveguide propagation of sound is characterized by a special form of sound signal dispersion, due to the dispersive properties of the propagation channel itself. The essence of the matter is that each normal mode in the waveguide propagates with its own group velocity c_l^{gr} which, in turn, depends on frequency. Thus, for propagation in a sound channel, one can separately identify the intermode dispersion due to the different velocities of the individual modes and the intramode dispersion due to the frequency dependence of the group velocities of the individual modes. Strictly speaking, both the intermode and the intramode dispersion cause distortion of signal shapes which should be taken into account in underwater applications at a large distance. On the contrary, this distortion also allows us to separate pulse signals corresponding to individual modes that can be used for various applications.

To begin with, let us consider intermode dispersion and demonstrate the group velocity behavior of normal modes in shallow water. For this purpose, we will

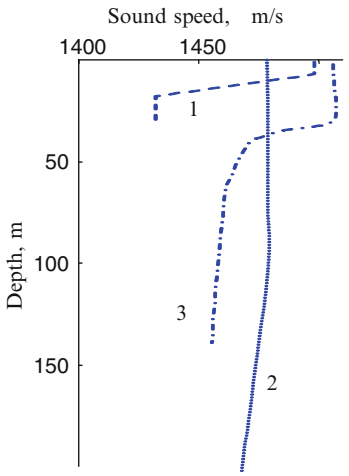


Fig. 4.13 Vertical sound speed profiles. 1 – North Sea, 2 – Barents Sea, 3 – Sea of Japan

consider three model waveguides, approximately corresponding to the environment in the areas of the Atlantic shelf of the USA (Apel et al. 1997) (profile 1), the Barents Sea shelf (Bunkin et al. 1989b; Katsnelson and Petnikov 2002) (profile 2), and the Yellow Sea (Rogers et al. 2000) (profile 3) with different depths (Fig. 4.13). Roughly, we can take the bottom parameters as being approximately the same: sound speed, density, and attenuation coefficient are set equal to $c_1 = 1{,}573$ m/s, $\rho_1 = 1.8$ g/cm^3, and $\hat{\beta}_\lambda = 0.4\,\mathrm{dB}/\lambda$, respectively. For these waveguides, frequency dependences of the group velocities for the first four modes are plotted in Fig. 4.14a–c. As seen from this figure, for comparatively low frequency, the group velocities increase as a function of frequency and decrease with mode number. For higher frequency, c_l^{gr} depends in a more complex form on frequency and mode number. Also, we can see that for different conditions, the boundary between these areas with different behavior changes. Also, for high frequency, the group velocities of the different modes are close to each other and roughly decrease with mode number.

In order to understand these features of waveguide dispersion, let us examine a simplified expression for the group velocity using (4.8). Let us also use the WKB expressions for the eigenfunctions (4.10). Then, applying (4.17) and (4.18), we can write:

$$A_l = \sqrt{\frac{2q_l}{D_l + \Delta_l}}. \tag{4.75}$$

For this case, substituting (4.10) into (4.18), we can get a similar approximation:

$$\frac{q_l}{\omega}\frac{dq_l}{d\omega} = \frac{2q_l}{D_l + \Delta_l}\left[\int_0^H \frac{1}{\sigma_l(z)c^2(z)}dz - \frac{\sigma_l g_l^3(1 - n_1^2)}{m_1^2 c^2(1 + \sigma_l^2 g_l^2)}\right]. \tag{4.76}$$

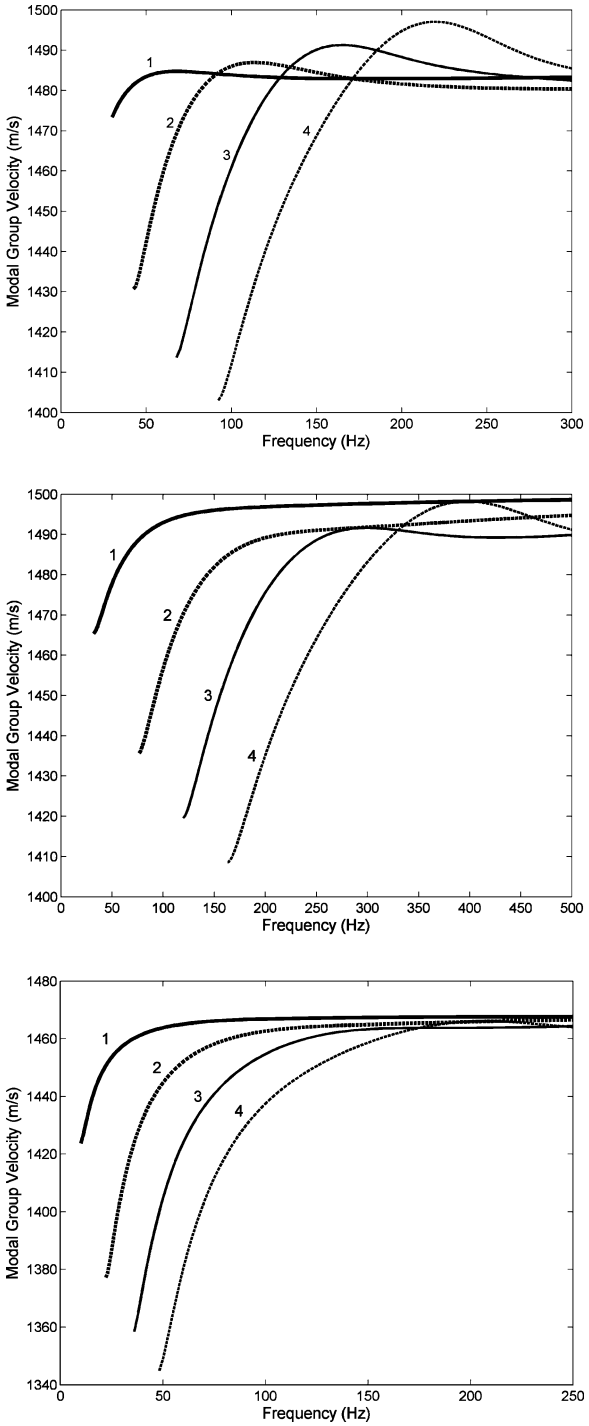


Fig. 4.14 Frequency-dependence of the group velocities for individual modes. The mode numbers are in the figure

Let us attempt a ray interpretation of this expression, assuming we have some correspondence between the mode number and the ray grazing angle. Then for the first and the second terms in square brackets on the right-hand side of (4.76), we can write the following expressions (respectively):

$$2\omega \int_{\underline{z}}^{H} \frac{dz}{\sigma_l(z)c^2(z)} = T_l, \tag{4.77}$$

$$\frac{\partial \varphi_V}{\partial \omega} = -\frac{\sigma_l g_l^3 (1 - n_1^2)}{m_1^2 c^2 (1 + \sigma_l^2 g_l^2)} \omega = \tau_l, \tag{4.78}$$

where T_l is defined as the time of propagation along a ray cycle, and the value $\hat{\tau}_l$ can be considered as the time delay of the sound pulse upon reflection, similar to the displacement of the beam (Jin et al. 1993). So a simplified expression for the group velocities can be expressed as

$$c_l^{\mathrm{gr}} = \frac{D_l + \Delta_l}{T_l + \tau_l}. \tag{4.79}$$

All parameters mentioned depend on grazing angle, and this dependence is initially determined by waveguide properties, depth, sound speed profile, and bottom parameters. If we neglect the displacement of the beam and the time delay upon reflection, we have the obvious relationship for the group velocity:

$$c_l^{\mathrm{gr}} = \frac{D_l}{T_l}. \tag{4.80}$$

In the ray approximation, we can get an explicit dependence on the mode number (or in ray language on the grazing angle). Specifically, we can use (4.13a–c) to get the group velocity as a function of grazing angle. In the simplest case, for the Pekeris model, we obtain

$$c^{\mathrm{gr}}(\chi) = \frac{D(\chi)}{T(\chi)} = \frac{2H/\tan\chi}{2H/c\sin\chi} = c\cos\chi. \tag{4.81}$$

We see that the group velocity as a function of grazing angle decreases, in contrast to the phase velocity, which increases. We note also that for the Pekeris model,

$$\cos\chi \cong 1 - 2^{-3} c^2 f^{-2} (l^*/H)^2, \tag{4.82}$$

and so c^{gr} increases with increasing frequency and decreases with increasing mode number:

$$c^{\mathrm{gr}} \cong c\left[1 - 2^{-3}c^2 f^{-2}(l^*/H)^2\right]. \tag{4.83}$$

Recall that parameter l^* lies in the range from $l - 0.5$ to l (see Sect. 3.2) and that the exact value of c^{gr} is determined by (3.47) for the Pekeris model.

Another situation worth examining is the waveguide with a negative gradient sound speed profile. For bottom rays (modes), the ray cycle distance D and the time of propagation along a ray cycle T are equal to

$$D(\chi) = \frac{2c_H}{\varepsilon c_s}\tan\chi, \tag{4.84}$$

$$T(\chi) = \frac{1}{\varepsilon c_s}\ln\left(\frac{1+\sin\chi}{1-\sin\chi}\right), \tag{4.85}$$

respectively. Thus, the group velocity equals

$$c^{\mathrm{gr}}(\chi) = 2c_H\frac{\tan\chi}{\ln[(1+\sin\chi)/(1-\sin\chi)]}. \tag{4.86}$$

Here, we have a rather small increase in the group velocities as a function of angle, so we can see that for bottom rays (modes), the group velocities are almost the same and $c^{\mathrm{gr}} \sim c_H$.[3] How can we understand this? Let us consider (4.9). For the Pekeris waveguide, the expression on the right-hand side depends weakly on mode number, because the area occupied by each mode is one and the same (between the surface and the bottom). This means that if the phase velocity increases, then the group velocity decreases. In a near-bottom channel, the area occupied by the bottom modes increases with mode number and it is possible that the phase and group velocities can grow simultaneously, or at least that the group velocities do not decrease with mode number (grazing angle) and do not increase with frequency. However, this situation takes place only for bottom modes. For bottom-surface modes, the χ dependencies of the group velocities are similar to the dependencies for the Pekeris waveguide.

Now the group velocity behavior shown in Fig. 4.14a–c becomes understandable. For low frequency, the second, third, and fourth modes are bottom-surface modes and their $c_l^{\mathrm{gr}}(f)$ trends are analogous to those of the Pekeris waveguide. As frequency increases, these modes become bottom modes and the shape of $c_l^{\mathrm{gr}}(f)$ changes. We should remark that in region of high frequency for a near-bottom channel, we have

the opposite dependence of the group velocity on the mode number compared to the Pekeris model – the group velocity increases with mode number.

Let us further examine the signal transformation and distortion caused by waveguide dispersion in long-range sound propagation in shallow water. First, let us consider the propagation of a sound signal at low frequency in the waveguide where the difference in group velocities is clearly defined. Specifically, let us consider the propagation of a narrow band (10 Hz) sound signal with carrier frequency 60 Hz as exemplified in Fig. 4.15a. This signal, received at a distance 14 km from the sound source, is shown in Fig. 4.15b. Using color gradation, the amplitude of the signal for different depths is shown in Fig. 4.15c. (The amplitudes of the signals shown in Fig. 4.15 were normalized to maximum values.) These signals were recorded in the SWARM95 experiment (Apel et al. 1997), which was carried out in a waveguide corresponding roughly to the model waveguide described above. More exactly, these signals were obtained by using narrow band ($\Delta f = 10$ Hz) filters on the broadband signals radiated by an air gun and received both at a small distance $\cong 1$ m from this source (Fig. 4.15a) and at a large distance $L \sim 14$ km (Fig. 4.15b).

From a formal point of view, the signal shown in Fig. 4.15b,c can be written using (A.165), where $S(\omega)$ is the spectrum of the narrow band signal, shown in Fig. 4.15a and $\Psi(\vec{R}, \vec{R}_0, \omega)$ is the Green function (see Appendix B). As seen from Fig. 4.15b, c in the reception at 14 km, we observe several signals, each corresponding to the individual waveguide modes. In other words, we see that due to the difference in group velocities, the separation of the signal into individually arriving waveguide modes takes place. Regarding numerical values, it is easy to see that the difference in arrival times of the waveguide modes $\Delta t_{12} \sim 0.2$ s and $\Delta t_{23} \sim 0.4$ s is in good accordance with rough estimations $\Delta t_{lm} = (v_l^{\mathrm{gr}} - v_m^{\mathrm{gr}})L^{-1}$ if we take group velocities of modes from Fig. 4.14 for the corresponding frequency of 60 Hz.

For higher frequency, and in other regions of shallow water where the difference in group velocities is not as great, the time separation of signals corresponding to individual waveguide modes does not always happen. In this case, we have a more complex signal structure than that shown in Fig. 4.15b.

Let us next consider propagation in the framework of a numerical experiment for a waveguide corresponding to the Barents Sea (Katsnelson et al. 2004a). (This work will also allow us to demonstrate some possible techniques to separate modes.) For this case, the depth is equal to $H = 120$ m, the sound velocity in the sea bottom is $c_1 = 1{,}750$ m/s, the bottom density is $\rho_1 = 1.9$ g/cm^3, and $\alpha = 0.01$. The vertical profile of sound speed measured in the experiment is shown in Fig. 4.16. Notice that for mode separation, we should use the signal with minimal duration $\tau \cong 1/\Delta f$, or equivalently use a correlation detector at the reception point. For a correlation detector, the output signal envelope is equal to

$$B(t) \cong \left| \int_{\Delta\omega} \Psi(r, z, \omega, t) |S(\omega)|^2 \mathrm{e}^{-i\omega t} d\omega \right|. \tag{4.87}$$

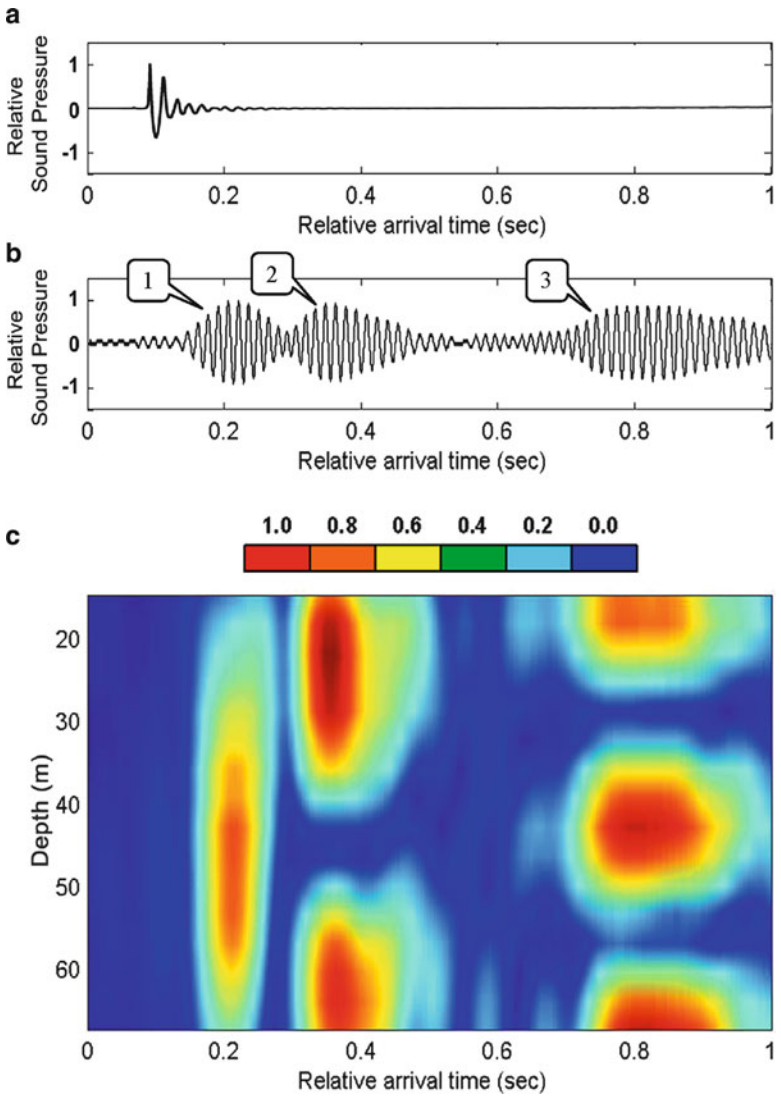


Fig. 4.15 Signal transformation and distortion caused by waveguide dispersion. The signal received at a distance 1 m (**a**) and 14 km (**b**) from sound source. The envelope of the signals received at a distance 14 km and at different depth (**c**). In panel b, the mode numbers are shown in the figure

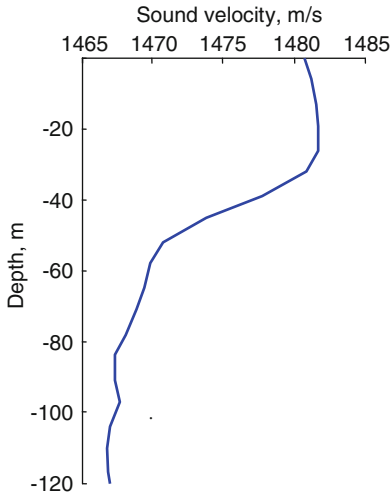


Fig. 4.16 Sound velocity profile in the Barents Sea

For an example, let us consider a chirp[4] signal with carrier frequency $f = 240$ Hz and $\Delta f = \Delta\omega/2\pi = 20$ Hz. For this waveguide and for this carrier frequency, the group velocities c_l^{gr} and the attenuation coefficients γ_l of the individual modes are shown in Fig. 4.17. As seen from the figure, as well as in the previous case, at least five bottom modes are available with almost the same values of c_l^{gr}. These modes have the smallest values of the attenuation coefficient, which are also close in value. Furthermore, as the mode number increases, the group velocities decrease and the attenuation coefficients increase.

Now let us suppose that the depths of the sound source and receiver are equal to $z_0 = 40$ m and $z = 75$ m, respectively.[5] For this case, the envelope of the signal $B(t)$ is plotted in Fig. 4.18. As the figure suggests, the waveguide dispersion causes noticeable signal distortion at ranges of several tens of kilometers; however, the separation of the pulses, corresponding to the individual modes, takes place only at a distance of more than 200 km. It should also be noted that such a picture is only observed in the absence of strong random inhomogeneties, which cause mode coupling. With random inhomogeneties, the signal distortions become far more complex.

In a waveguide with constant sound speed (Pekeris model), we have similar signal distortions (Katsnelson and Petnikov 2002) with the difference being that the arrival time of the mode pulses increases monotonically with mode number, and the

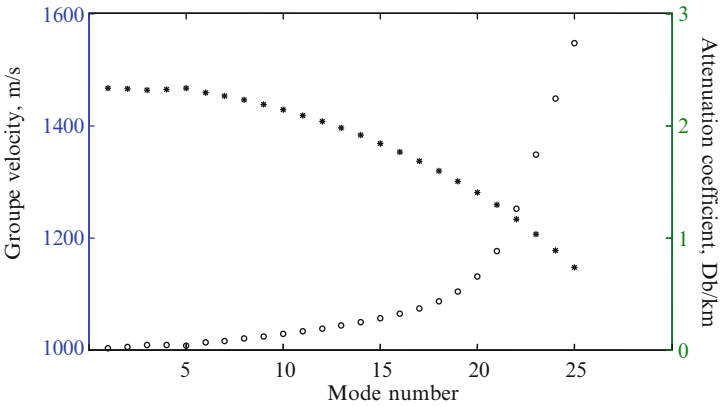


Fig. 4.17 Dependence of the group velocity (*asterisks*) and the attenuation coefficient (*circles*) on the mode number

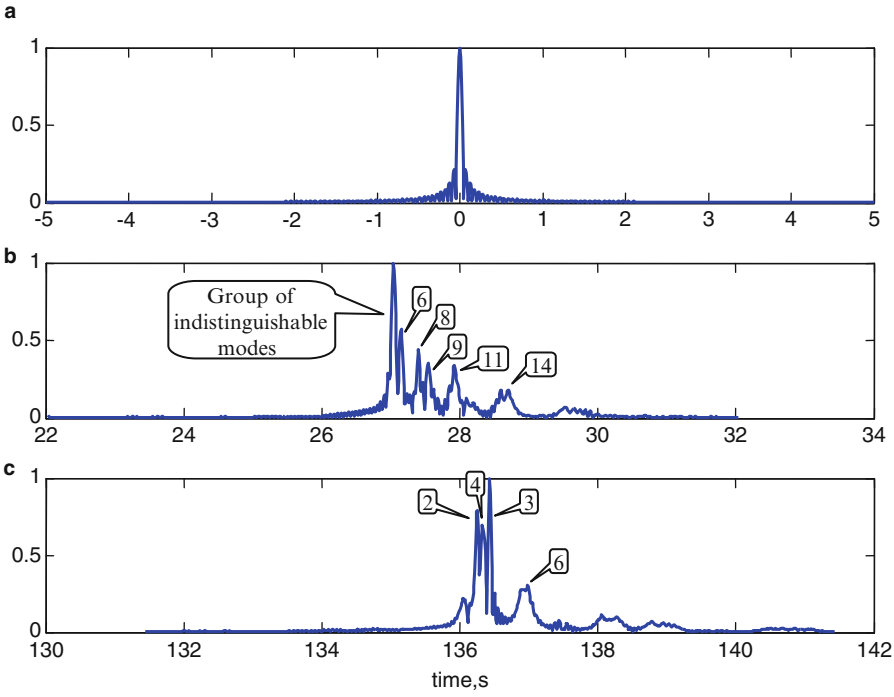


Fig. 4.18 Dependence of $B(t)$ normalized by its maximum value in the absence of random inhomogeneities: (**a**) the autocorrelation function of the emitted signal ($\Psi(r,z,\omega,t) = 1$); the distance is $r = 40$ km (**b**) and $r = 200$ km (**c**) The mode numbers are in the figure

pulse separation of the lowest modes begins to become more detectable. For the Pekeris model, we can write a simple expression for the distance r_{bm} where we can separate pulses. Indeed, the separation takes place when the difference in the time of arrival of two adjacent modes $\Delta\tau = \left(r_{\text{bm}}/c_{l+1}^{\text{gr}} - r_{\text{bm}}/c_{l}^{\text{gr}}\right)$ satisfies $\Delta\tau \geq \tau \cong 1/\Delta f$, where τ is the duration of the mode pulse at the output of correlation detector, neglecting intramode dispersion. (Notice that τ is also the pulse duration at the output of the correlation detector near the sound source if we can neglect waveguide dispersion). Correspondingly, by using (4.83), we get

$$r_{\text{bm}} \geq \frac{4H^2f^2}{lc\Delta f}. \tag{4.88}$$

From here on, we will take $l^* = l$. Specifically, for our numerical example ($H = 120$ m, $f = 240$ Hz, and $\Delta f = 20$ Hz) and for $l = 3$, $c = 1{,}470$ m/s, the distance is found to be $r_{\text{bm}} \geq 38$ km.

It is important to note as well that at the receiving point, the full signal duration τ_1 (including all modes) does not depend strongly on range. This is a typical situation in shallow water and is true because, on one hand, the signal duration increases with distance through the difference in the mode group velocities, but on the other hand, the number of modes decreases with distance because of attenuation and the signal duration decreases correspondingly.

Another peculiarity of wave dispersion is connected with intramode dispersion, which can also cause distortion of mode pulses. Usually, this distortion is detectable only for modes with relatively large mode numbers and at relatively large distances from sound source. (See, for example, the third mode in Fig. 4.15 and the sixth mode in Fig. 4.17.) Indeed, strong distortion takes place only when the distance r_{im} satisfies $r_{\text{im}}/c_l^{\text{gr}}(f_1) - r_{\text{im}}/c_l^{\text{gr}}(f_2) \geq \Delta\tau$, where f_1 and f_2 are the limiting frequencies in the signal spectrum ($f_2 - f_1 = \Delta f$). For the Pekeris model, this distance r_{im} is equal to

$$r_{\text{im}} \geq \frac{4H^2f^3}{\Delta f^2 cl^2}. \tag{4.89}$$

When the following inequality holds, i.e.,

$$\frac{f}{2\Delta fl} > 1, \tag{4.90}$$

then r_{im} is greater than r_{bm} and, consequently, there exists an interval of distances wherein the signals corresponding to individual modes are easily distinguished according to their propagation time, and also contain no marked dispersion distortions. In particular, for the parameters used above r_{bm}, $r_{\text{im}} \geq 150$ km.

In a waveguide with a negative gradient sound speed profile, r_{im} further increases for bottom modes because the group velocities of these modes depend only slightly on the frequency (see Fig. 4.14) for $f >$ 200-300 Hz.

4.6 Averaged Description of the Sound Field in a Waveguide

As has been seen, the sound field in a shallow water waveguide has a complex interference structure characterized by many spatial scales. A typical pattern, obtained via numerical modeling, is shown in Fig. 3.22, and displays the sound field from an acoustic point source lying on the bottom of a model waveguide. The sound intensity is shown in color gradations for the characteristic case of a "summer" sound speed profile with a negative gradient (near-bottom sound channel). It is interesting to note that the result of the interference between modes corresponds to a ray pattern (shown in Fig. 3.22c). It is further noteworthy that as one moves away from the source, the sound field pattern becomes less regular, taking on a rather blurred structure which may appear random, despite the fact that this pattern is completely deterministic.

With these points in mind, we note that when one studies the propagation of sound in an underwater sound channel, it is often not the fine interference structure of the field (for which it may be very difficult to compare measured and calculated values) but certain average characteristics, relating mostly to the square of the sound pressure averaged over some regions (modulo a sound intensity constant), which are of interest (Brekhovskikh 1965; Borodin 1992; Brekhovskikh and Lysanov 1991; Denham 1986). By the intensity, we mean the amount of sound energy passing through a unit area in unit time. The spatial scale of averaging is chosen to be greater than the largest scale of the interference of normal modes, which is the longest cycle distance of rays in the channel. As a result of this averaging, interference (with spatial scales on the order of the wavelength) and zonal (with scale D) variations of the sound field vanish, leaving a comparatively smooth dependence of the intensity on the distance from the sound source.

This averaging will be most effective if a very large number of normal waves propagate in the channel (the waveguide is highly multimodal). Precise parameters characterizing the extent to which such an approximation is applicable will be indicated below.

In ray theory, this averaging corresponds to summing the energy of ray tubes having the cycle distance of rays coming from the source. An analogous averaged pattern is obtained if one averages not over distance at fixed frequency, but over frequency at a fixed distance, in other words, if one considers the field of a wideband source. When one averages over the spatial interval $\Lambda_l \approx 2\pi|q_{l+1} - q_l|^{-1}$, variations due to the presence of terms of the type $\exp[\mathrm{i}(q_{l+1} - q_l)r]$ vanish. Analogously, when one averages over frequency, the characteristic interval will be $\Delta\omega \approx \frac{2\pi}{r}\left[\frac{d(q_{l+1}-q_l)}{d\omega}\right]^{-1}$, and if we take $\frac{dq_l}{d\omega} \approx \frac{d}{d\omega}\sqrt{\frac{\omega^2}{c^2} - \sigma_l^2} \approx \frac{\omega}{q_l c^2}$, then we obtain

that $\frac{\Delta\omega}{\omega} \approx \frac{\Lambda_l}{r}$ (Borodin 1992). For a number of model underwater sound channels, the dependence of the averaged intensity on the distance r can be obtained in analytic form (Brekhovskikh and Lysanov 1991; Weston 1976; 1980a, b, c, d). In particular, Brekhovskikh (1965) established that for the Pekeris model with an absorbing bottom, the intensity falls with distance as $r^{-3/2}$. The nature of this fall off, which is intermediate between the cylindrical and spherical spreading laws, is due to the considerable role of the absorbing bottom, upon which rays are multiply reflected.

The case of an irregular bottom was considered by Weston (1976) where, neglecting losses in propagation, a number of useful equations for $\bar{I}(r)$ were obtained. Focke and Weston (1978) considered the role of caustics in the averaging of the sound field, while Katsnelson et al. (1985) compared the results of calculations based on averaged equations for a specific track using experimental results.

Below, we consider a generalized method for obtaining an averaged description of the sound intensity, in conditions where the rays undergo multiple reflections from absorbing boundaries. This theory is also applicable for waveguides with the depth varying along the track and a variable sound speed profile.

Suppose that in a waveguide with parameters (the depth $H(r)$, the bottom characteristics, and the sound speed profile $c(r,z)$) varying sufficiently smoothly along an acoustic track, a source is located, as usual, at the point with coordinates $r = 0, z = z_0$ and has a total emitted power W_0.

We represent the sound field in such an irregular channel similarly to (3.25), taking into account the dependence of the eigenvalues and eigenfunctions on distance as per (3.145). We then write down an expression for the sound pressure at the observation point $P(r,z)$, separating, in what follows, the real and imaginary parts of the eigenvalues. In the expression within the square root in the denominator, the imaginary part of the eigenvalue is small compared with the real part and is neglected. Thus, we get

$$P(r,z) = \sqrt{\rho_0 c_0 W_0} \sum_{l=1}^{N} \frac{\Psi_l(0,z_0)\Psi_l(r,z)}{\sqrt{r q_l(r)}} \exp\left[i \int_0^r q_l(r')dr' - \int_0^r \frac{\gamma_l(r')}{2} dr' + i\frac{\pi}{4} \right]. \tag{4.91}$$

Here, c_0 and ρ_0 denote the sound speed and density at the source point. We shall calculate the "local" intensity at the observation point as

$$I = \frac{1}{\rho c} \left\langle \left\{ \mathrm{Re}\left[P(r,z) \mathrm{e}^{-i\omega t} \right] \right\}^2 \right\rangle. \tag{4.92}$$

Here, $\langle\rangle$ denotes averaging over an intermediate time interval, greater than the period of the sound oscillations. The values of the density ρ and the sound speed c are taken at the observation point, and we assume that the inhomogeneity of the medium is sufficiently smooth that the wavefront is locally planar, i.e., $|\nabla p| \sim k$.

The averaging over the spatial interval is carried out in two steps. First, it is performed over a longitudinal scale D (in the neighborhood of the receiver) greater than the interval of the interference variations, as a result of which we obtain

$$\bar{I} = \frac{\bar{W}_0}{2r}\frac{\rho_0 c_0}{\rho c}\sum_l \frac{\Psi_l^2(0,z_0)\Psi_l^2(r,z)}{q_l(r)}\exp\left[-\int_0^r \gamma_l(r')dr'\right]. \tag{4.93}$$

We then average over a vertical transverse scale $\approx H$ in the neighborhood of the receiver and the source, for which we use the WKB method (4.10) and (4.11) to represent the functions $\Psi_l(0,z_0), \Psi_l(r,z)$. We obtain

$$\bar{I} = \frac{2\bar{W}_0}{r}\sum_{l=1}^{M}\frac{q_l(0)\exp\left(-\int_0^r \gamma_l(r')dr'\right)}{\sigma_l(0,z_1)\sigma_l(r,z)D_l(0)D_l(r)}, \tag{4.94}$$

where we neglect the differences in density and sound speed at the source point and the observation point in the factor in front of the intensity, since the error incurred by neglecting these is extremely small.

Expressions (4.93), (4.94) can be used to compute the average intensity. However, to obtain analytic expressions, we can move to a ray representation if we replace the sum over the discrete variable l by integration over a continuous variable. This variable might, for example, be the angle from the horizontal axis χ_0 at which a ray emerges from the source. To do this, we shall use a representation of l in terms of the ray invariant J (Brekhovskikh and Lysanov 1991):

$$dl = \frac{\omega}{2\pi}dJ. \tag{4.95}$$

Here,

$$J = \oint \frac{\sin\chi(r,z)}{c(r,z)}dz = \oint \left[c^{-2}(r,z) - c^{-2}(r,z_0)\cos^2\chi(r)\right]^{1/2}dz, \tag{4.96}$$

where the integration is performed over a waveguide section. Here, $\chi(r)$ is the ray grazing angle on the source horizon, which naturally varies with distance. The angle formed by the ray with the horizontal plane (grazing angle) $\chi(r,z)$ depends, generally speaking, on z and r; in what follows, we shall use the notation $\chi(r) = \chi(r,z_0), \chi_0 = \chi(0,z_0)$, and for the grazing angle on the bottom $\chi_H(r) = \chi(r,H(r))$ (see Fig. 4.19). By virtue of (4.96), we have

$$dJ = \frac{\sin\chi_0}{c(0,0)}Dd\chi_0$$

and as a result,

$$\bar{I} = \frac{2}{r} \int_{-\pi/2}^{\pi/2} \frac{E(\chi_0, r) \cos \chi_0}{D(\chi_0, r) \sin \chi(\chi_0, r)} d\chi_0. \tag{4.97}$$

Here, and later in this section, we have normalized the total power of the source to $\bar{W}_0 = 4\pi$, i.e., we have assumed that the source radiates unit power into a unit solid angle. The attenuation factor $E(\chi_0, r) = \exp\left(-\int \gamma_l dr\right)$, which thus can also be replaced by a ray equation if we use (4.20):

$$E(\chi_0, r) = \exp\left[-\int_0^r \gamma_l(r')dr'\right] = \exp\left[-2\int_0^r \frac{\ln|V(\chi_H, r')|}{D(\chi_0, r')} dr'\right]. \tag{4.98}$$

Instead of the mode index in (4.97), we now have the angle at which a ray leaves the source χ_0, which determines the grazing angle of that ray on the bottom χ_H, upon which, in turn, the attenuation coefficient in (4.98) depends. Equation (4.98) can also be obtained purely via ray theory.

Thus, to compute $\bar{I}(r)$ from (4.97), we need to know the forms of the functions $D(\chi_0, r), \chi(\chi_0, r), \chi_H(\chi_0, r)$ and $E(\chi_0, r)$. Below, we shall find these in the adiabatic approximation, based on the notion of the ray invariant, (4.96). Because the value of J is constant along the track, we can express J explicitly in terms of the above quantities, and thus compute these dependences explicitly. The ray cycle distance, according to Brekhovskikh and Lysanov (1991), is equal to

$$D(\chi_0, r) = \oint \frac{dz}{\tan \chi(\chi_0, r, z)}. \tag{4.99}$$

It can also be computed if we know $\chi(\chi_0, r, z)$. After the quantities χ_H, D have been found, they are substituted in expressions (4.97) and (4.98) and the integration is performed.

4.6.1 Channel with Constant Sound Speed and Variable Depth

Let us consider a simple underwater sound channel in which the sound speed is constant ($c = \text{const}$) and the absorbing bottom has a variable depth (Fig. 4.19). For such a channel, within a single ray cycle, the angle χ does not depend on z so that the ray invariant J is equal to

$$J = 2\frac{H(r)\sin\chi(r)}{c} = 2\frac{H(0)\sin\chi(0)}{c} = 2\frac{H_0 \sin\chi_0}{c}. \tag{4.100}$$

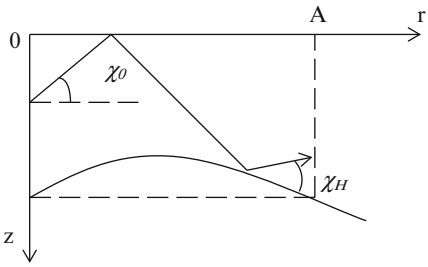


Fig. 4.19 Channel with constant sound speed, bounded by an absorbing bottom of variable depth

Hence, we can express $\chi(r)$ in terms of $\chi_0, H(r)$ and $\chi_H(r)$. The grazing angle $\chi_H(r)$ in the adiabatic approximation is equal to $\chi(r)$. Furthermore, we can compute the ray cycle distance

$$D(\chi_0, r) = \frac{2H(r)}{\chi(\chi_0, r)} = \frac{2H^2(r)}{H_0 \sin\chi_0}\sqrt{1 - \frac{H_0^2 \sin^2\chi_0}{H^2(r)}}. \tag{4.101}$$

To compute the attenuation coefficient of a ray, we use (4.23) in which we take χ_H for χ. Restricting our analysis solely to the contribution of rays with small grazing angles, we have

$$\gamma = sH_0\chi_0^2/H(r), \tag{4.102}$$

$$D(\chi_0, r) = 2H^2(r)/H_0\chi_0. \tag{4.103}$$

Thus, after substituting formulae (4.98), (4.102), and (4.103) into expression (4.97), we obtain

$$\begin{aligned}\overline{I}(r) &= \frac{1}{rH(r)}\int_{-\pi/2}^{\pi/2} d\chi_0 \exp\left[-sH_0^2\int_0^r \frac{dr'}{H^3(r')}\chi_0^2\right] \\ &= \frac{\sqrt{\pi/s}}{r_0 H_0 H(r)}\left[\int_0^r H^{-3}(r')dr'\right]^{-1/2}.\end{aligned} \tag{4.104}$$

For a constant depth ($H(r) = H_0 = \text{const}$), (4.104) becomes the expression found in Brekhovskikh (1965) and Brekhovskikh and Lysanov (1991):

$$\overline{I_0} = \sqrt{\pi/sH_0}\, r^{-3/2}. \tag{4.105}$$

Significantly, in expression (4.104), the intensity at the point r is determined by the integral of the whole bottom profile. Using (4.104), one can consider the

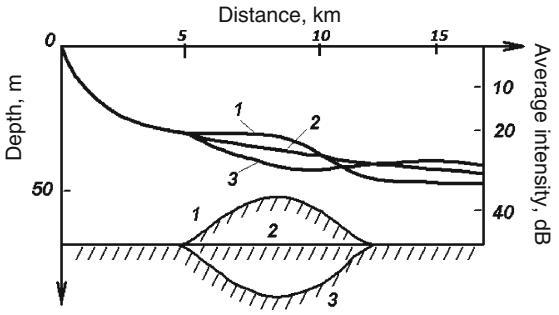


Fig. 4.20 Graph of $\bar{I}(r)$ in presence of a "hill" and a "trough". 1–3 profiles of the bottom and the corresponding averaged intensity curves

behavior of the intensity for various bottom profiles. Results for a number of profiles were obtained in Katsnelson and Kulapin (1984). For example, for a sound channel with linearly varying depth $H(r) = H_0 \pm \varepsilon r$, calculations using (4.104) give the expression

$$\bar{I}(r) = \sqrt{\frac{\pi}{sH_0}} \frac{r^{-3/2}}{\sqrt{1 \pm \varepsilon r/2H_0}} \tag{4.106}$$

which differs from the law for transmission loss in a regular channel by the factor $(1 \pm \varepsilon r/2H_0)^{-1/2}$. Let us consider the role of this factor. If, for example, the channel gets shallower with distance (minus sign), then the intensity of the sound field at distance r from the source will, naturally, be greater than $\overline{I_0}$, but less than when the conventional adiabatic factor $(1 - \varepsilon r/H_0)^{-1}$ is used, which is inversely proportional to the local depth of the waveguide. This is because the absorption coefficient (4.23) increases when the waveguide shoals so that the grazing angle χ grows and the ray cycle distance D decreases. Analogously, one can consider a deepening waveguide. Here, the absorption coefficient is less than that in a regular waveguide, and due to the decreased loss in comparison with that in the regular waveguide, the intensity decreases more slowly as the depth grows.

Thus, (4.104), obtained in the adiabatic approximation, indicates that the average field at a given point is determined by the whole of the bottom profile along the area the track passed through. This conclusion can be illustrated by the following example. Suppose the water depth in section A (Fig. 4.19) is the same as at $r = 0$. What change in the field in section A is induced by the presence of a hill between the source and section A? In the adiabatic approximation, when the field characteristics are determined by local values of the waveguide parameters and there is no absorption, the field in section A does not change. Meanwhile, it follows from (4.104) that the integral in the presence of a hill increases and, consequently, the average intensity decreases. Analogously, if there is a trough between sections O and A, the field in section A increases in comparison with the case of a regular sound channel. Figure 4.20 illustrates these arguments and shows the behavior of the $\bar{I}(r)$ curves for the above channels.

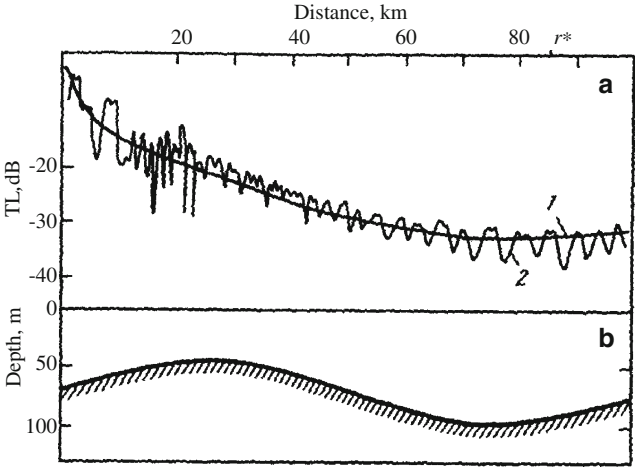


Fig. 4.21 Calculated values of the relative sound intensity $\mathrm{TL} = 10\lg(I/I_0)$ from a point source (**a**) and bottom profile (**b**). 1 – Based on averaged equation (4.104); 2 – using summation of modes; parameter values for the calculation are $f = 100$ Hz, $c = 1{,}500$ m/s, $c_1 = 1{,}580$ m/s, $s = 0.1$, $z_0 = z = 50$ m

Figure 4.21 shows an example of a calculation of the averaged intensity for a channel of variable depth. This figure shows the result of calculations using the full summation of modes, as studied in Sect. 3.2. From the figure, it is clear that in the region of applicability of formula (4.104), where the number of propagating modes is large, there is a good agreement between the curves. The distance r^*, determining the range of applicability of (4.97) [and respectively of (4.104)], is computed via the equation

$$\int_0^{r^*} H^{-3}dr \le 16\pi^{-2}s^{-1}\lambda^{-2}. \tag{4.107}$$

4.6.2 *Irregular Near-Bottom Sound Channel*

Let us next consider an irregular near-bottom sound channel (Fig. 4.22). Here, the sound propagation takes place by successive reflections of refracted rays from the bottom. We shall assume that the sound speed varies linearly with depth and that the waveguide parameters vary smoothly along the track: $c(r,z) = c_s(r)(1-\varepsilon(r)z), H = H(r)$. The ray cycle distance is easily found from

$$D(\chi_H, r) = \frac{2c_H}{\varepsilon c_S}\tan\chi_H, \tag{4.108}$$

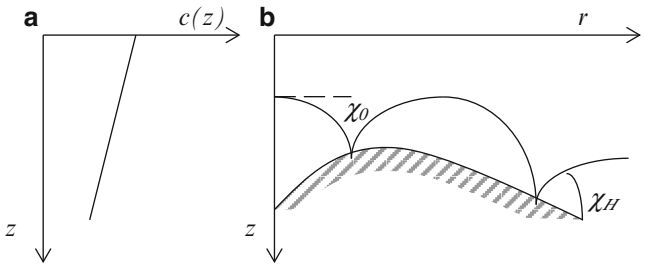


Fig. 4.22 Irregular near-bottom sound channel with an homogeneous absorbing fluid bottom and variable depth. (**a**) sound speed profile; (**b**) bottom profile

where $c_H = c(r, H(r)) = c_S(1 - \varepsilon H)$. To calculate the averaged intensity using (4.97), we need to know the dependence of the grazing angle $\chi_H(r)$ on the angle χ_0. This dependence is determined using the ray invariant (4.96)

$$J = \frac{1}{\varepsilon c_S}\left(\ln\frac{1+\sin\chi_H}{1-\sin\chi_H} - 2\sin\chi_H\right). \tag{4.109}$$

In real conditions, the main contribution to the resulting field comes from rays with small grazing angles; thus, in expressions (4.108) and (4.109), we can restrict ourselves to terms of the first nonvanishing order in χ_H:

$$J \approx \frac{2}{3\varepsilon c_S}\chi_H^3 \approx \frac{2}{3\varepsilon_0 c_{S0}}\chi_{H0}^3. \tag{4.110}$$

Here, $\varepsilon_0 = \varepsilon(0), c_{S0} = c_S(0)$, and $\chi_{H0} = \chi_H(0)$. The dependence of $\chi_H(r)$ on the angle at which the ray leaves the source can be determined using Snells' law and the value of J in the plane $r = 0$, i.e.,

$$\chi_H = (\varepsilon c_S/\varepsilon_0 c_{S0})^{1/3}\arccos(n_s\cos\chi_0), n_s = c_{H0}/c_{S0}. \tag{4.111}$$

Assuming additionally that the angle χ_0 is small, we can decompose our expressions (4.111) in a series in χ_0:

$$\chi_H(\chi_0, r) \approx \left(\frac{\varepsilon c_S}{\varepsilon_0 c_{S0}}\right)^{1/3}\left(\arccos n_s + \frac{n_s}{2\sqrt{1-n_s^2}}\chi_0^2\right). \tag{4.112}$$

From (4.111) and (4.112), it is clear that $\chi_0 = 0$ corresponds to the minimum grazing angle $\chi_{H\min} = (\varepsilon c_S/\varepsilon_0 c_{S0})^{1/3}\arccos n_s$. The factor $E(\chi_0, r)$, determining the sound loss from reflection, has the form (4.98), where $D(\chi_H, r)$ is determined by expression (4.108), and for $V(\chi_H)$, we take the Fresnel reflection coefficient (4.22)

with local grazing angle $\chi_H(r)$. We proceed in the usual way, decomposing the integrand in (4.98) into a power series in χ_H

$$2\frac{\ln|V(\chi_H)|}{D(\chi_H)} = \frac{\varepsilon c_S}{c_H}\left(-s + v\chi_H^2\right), \tag{4.113}$$

where

$$\begin{aligned} s &= -\frac{\ln|V(\chi)|}{\tan\chi}\Big|_{\chi=0} = \mathrm{Re}\frac{2m_1}{\sqrt{{n_1}^2(1+i\alpha)-1}}; \frac{d}{d\chi}\left[\frac{\ln|V(\chi)|}{\tan\chi}\right]_{\chi=0} = 0; \\ v &= \frac{1}{2}\frac{d^2}{d\chi^2}\left[\frac{\ln|V(\chi)|}{\tan\chi}\right]_{\chi=0} = \mathrm{Re}\frac{[3{n_1}^2(1+i\alpha)-2{m_1}^2]}{3[{n_1}^2(1+i\alpha)-1]^{3/2}}. \end{aligned} \tag{4.114}$$

Using (4.111) and (4.112), we can obtain a further decomposition of expression (4.113) in a series in χ_0:

$$2\frac{\ln|V(\chi_H)|}{D(\chi_H)} = -u(r) - w(r)\chi_0^2. \tag{4.115}$$

Here,

$$\begin{aligned} u(r) &= -\frac{\varepsilon c_S}{c_H}\left[-s + v\left(\frac{\varepsilon c_S}{\varepsilon_0 c_{S0}}\right)^{2/3}\arccos^2 n_s\right]; \\ w(r) &= -\frac{\varepsilon c_S v n_s}{c_H}\left(\frac{\varepsilon c_S}{\varepsilon_0 c_{S0}}\right)^{2/3}\frac{\arccos n_s}{\sqrt{1-n_s^2}}. \end{aligned} \tag{4.116}$$

Substituting expressions (4.98), (4.115), and (4.96) into (4.97), we finally have

$$\begin{aligned} \bar{I}(r) &= \frac{2}{r}\int_{-\pi/2}^{\pi/2}\frac{\cos\chi_0 \exp\left[-\int_0^r\left(u(r')+w(r')\chi_0^2\right)dr'\right]}{D(\chi_H)\sin\chi(\chi_H,r)}d\chi_0 \\ &= \frac{\varepsilon c_S\sqrt{\pi}}{c_H r}\left(\mathrm{tg}\chi_{H\min}\sqrt{1-\frac{c_S^2\cos^2\chi_{H\min}}{c_H^2}}\right)^{-1}\left[\int_0^r w(r')dr'\right]^{-1/2}\exp\left(-\int_0^r u(r')dr'\right). \end{aligned} \tag{4.117}$$

Equation (4.117) is analogous to (4.104). As in the latter, the averaged intensity in the section r is not determined locally but by the overall parameters of the channel, including all previous acoustic tracks up to the point r. Using (4.112) and (4.113), we can relate the parameters u, w, s, v to the values of the refractive index n and the density ratio m.

For the case of a regular waveguide [constant values of c_H, c_S, ε in (4.116)], instead of functions we have constants $u(r) = u, w(r) = w$, and (4.117) becomes

$$\bar{I}(r) = \frac{\varepsilon c_S}{c_H r^{3/2}} \sqrt{\frac{\pi}{w}} \left(\tan\chi_{H\min} \sqrt{1 - \frac{c_S^2 \cos^2 \chi_{H\min}}{c_H^2}} \right)^{-1} \exp(-ur). \tag{4.118}$$

This is a well-known result, showing that an exponential damping e^{-ur} is imposed upon the "three over two" law $I \approx r^{-3/2}$ (Brekhovskikh and Lysanov 1991; Katsnelson et al. 1984). As in the case of a channel with a constant sound speed, the criterion for applicability of the averaged equations is the existence of a large number of propagating modes. However, for the near-bottom channel, there is a small difference, namely, that the set of all "near-bottom" modes [whose number is determined by (4.26)], having the same attenuation coefficient, determines the exponential factor in (4.117), while the other factor controlling the decrease in intensity, of type $r^{-3/2}$, is determined by the width $\left(\int_0^r w dr'\right)^{-1/2}$ of the Gaussian angular distribution in the integrand in (4.117).

Thus, the distance r^*, determining the boundaries of applicability of (4.117), is determined by the condition that a sufficiently large number of modes should exist at this width. This criterion takes the form $\int_0^{r^*} w dr \leq \left(\frac{2kH}{\pi}\right)^2$.

The next model to consider is a channel with variable depth $H = H(r)$, having a linear sound speed profile $c = c_S(r)[1 - \varepsilon(r)z]$ and bounded by the surface $z = 0$. The difference from the previous case is that reflection from the surface is now taken into account. The bottom is a homogeneous fluid half-space with density ρ_H, and a sound speed c_H which is complex, owing to absorption.

For this model, the dependence of the absorption coefficient on the grazing angle is nonmonotonic, i.e., up to some angle $\chi_{0\min}$, corresponding to reflection of the ray from the surface, absorption decreases, and then it grows. The absorption coefficient $\gamma/2 = -\ln|V|/D$ behaves as shown in Fig. 4.3.

Finding the averaged law for the transmission loss in this model requires the computation of the integral (4.97) with a nonmonotonic dependence of $\gamma(\chi_0)$. Of course, at large distances, the angular spectrum of the radiation will be narrowed, and concentrated in the neighborhood of the angle $\chi_{0\min}$ corresponding to minimum absorption. Based on the above analysis, we conclude that for different regions corresponding to different distances from the source, the averaged dependence of the loss with distance will be different in nature. Namely, at small distances from the source, propagation will be of a spherical nature, and correspondingly, the law for the decrease in intensity will be spherical. Furthermore, in the presence of a sufficiently large number of excited modes, the law for the transmission loss will be of type $r^{-3/2}$ (or a suitable modification of that in the irregular case), with the possible imposition of exponential damping upon that if there are near-bottom modes in the channel. The spatial interval of this dependence is determined by the distance r^*, estimates of which were obtained in the Pekeris model and for a near-bottom channel. For example, for a typical case with $H = 100$ m, $f = 200$ Hz, and $s = 0.2$, we obtain a value for the characteristic distance of $r^* \approx 80$ km.

At distances greater than or equal to r^*, the field contains very few modes, namely, those having the smallest attenuation coefficients, and the decrease in the field with distance is exponential. Of course, in addition to this, one must also take into account the attenuation of sound due to absorption in the water, which is proportional to the square of the frequency, but this attenuation is small at the frequencies with which we are dealing (see the remarks at the end of Sect. 4.1). We note in conclusion that the nature of the decrease in the average intensity also depends on the positions of the receiver and the source. For example, for a source located near the surface and a receiver sufficiently far away from it, the averaged law for the transmission loss is given by a function of type $r^{-5/2}$, which is explained by the change in the emission distribution for modes near the boundary (there is a dipole nature of the emission). These questions were considered in detail in Grachev (1983), Denham (1986), and Weston (1980a, b, c, d, 1994).

Chapter 5
Sound Field in Shallow Water with Random Inhomogeneities

5.1 Structure and Models of Different Kinds of Random Inhomogeneities

Up to this point now we have considered sound propagation in shallow water as being through a deterministic, inhomogeneous medium. It is well known that these inhomogeneities can have a considerable effect on the propagation of sound. Moreover, we know from oceanography that these random inhomogeneities are primarily concentrated in the top kilometer of the water column (e.g., Babii 1983; Shavcho 1982, 1984), which leads to a particularly strong effect of random inhomogeneities in shallow water. Recently, great attention has been paid, both experimentally (Zhou et al. 1996; Headrick et al. 2000) and theoretically (Dozier and Tappert 1978; Gorskaya and Raevskii 1984; Ashley et al. 1987; Kuznetsova 1988; Derevyagina and Katsnel'son 1995; Creamer 1996; Beran and Frankenthal 1992, 1996; Frankenthal 1998), to the study of shallow water as a randomly inhomogeneous medium. Also, the propagation of waves of different kinds in a randomly inhomogeneous medium has been studied for a long time (beginning with Rayleigh) [see, for example, the monographs (Chernov 1975; Rytov et al. 1989; Ishimaru 1978; Flatte 1979)], etc. The major theme of this chapter will thus be the transition from the complex, detailed oceanography (see Chap. 2) to appropriate random medium representations, and then the description of propagation and scattering in these representations.

The oceanic medium is characterized by various space–time scales of variability [a detailed analysis can be found in Dickey (1991)]. These scales range from dimensions of the order of 1 cm and 1 s in time for molecular processes, to several hundreds and thousands of kilometers and several months in time for synoptic systems and seasonal changes. Of course, there is a certain amount of coarseness and arbitrariness about the division of such space–time variability into "deterministic" and "random" parts, associated with the scales of averaging. In our case, the choice of those scales is dictated by the scales of the space–time variability of the low-to-medium frequency sound field, i.e., we deem phenomena with characteristic spatial scales of from one meter (or less) to several tens of meters to be randomly

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_5,
© Springer Science+Business Media, LLC 2012

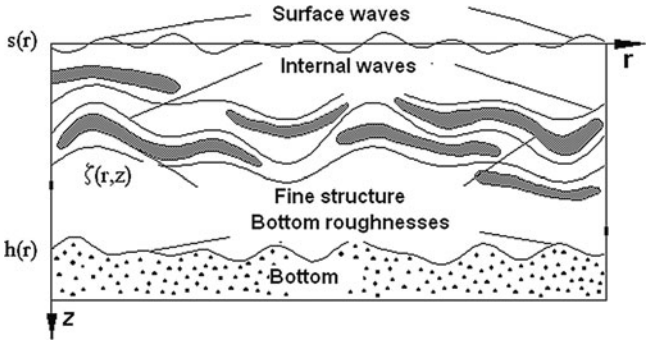


Fig. 5.1 Random inhomogeneities in shallow water

inhomogeneous. As far as temporal characteristics are concerned, due to the very small period of sound oscillations, or the small transit times along an acoustic track, we can, as a rule, work in the approximation of a "frozen" medium. Thus, the main phenomena in shallow water giving rise to random inhomogeneities are taken to include the following (see Fig. 5.1).

- The fine (thermohaline) structure (Fedorov 1976, 1978). This gives rise to fluctuations of the sound speed in the water layer $\delta c(r,z)$ with amplitude $\delta c_0 \sim 1\,\mathrm{m/s}$, horizontal scales of variability $L_r \sim 100 - 500\,\mathrm{m}$, vertical scales of variability $L_z \sim 0.1 - 10\,\mathrm{m}$, and a variance of the fluctuations which varies as a function of depth in the range 10^{-2}–$1\ \mathrm{m^2/s^2}$. For calculations of the sound speed fluctuations, the following correlation function was used in the above authors' works:

$$B_c(r,z;r',z') = \langle \delta c(r,z)\delta c(r',z')\rangle = \delta c_0^2 B_d(z,z')\exp\left(-|r-r'|/L_r\right), \tag{5.1}$$

  where $B_d(z,z')$ is the correlation coefficient in the vertical direction with a radius of at most 10 m.
- Internal waves, displacing surfaces of constant density $\zeta(\vec{r},z)$. One can show (see, for example, Flatte (1979) or Sect. 2.5) that these displacements lead to a change in the sound speed profile of the form [see (2.31)]:

$$\delta c(\vec{r},z) = c(z)\tilde{Q}\tilde{N}^2(z)\zeta(\vec{r},z), \tag{5.2}$$

  where $\tilde{Q} \approx 2.4\,\mathrm{s^2/m}$[1] is a constant, determined by the water properties, and $\tilde{N}(z) = \left(\frac{g}{\rho}\frac{d\rho}{dz}\right)^{1/2}$ is the buoyancy frequency, determined by the density stratification. Equation (5.2) allows us to associate the correlation characteristics (for example

[1] We remark that the coefficient Q depends on water properties (temperature, salinity etc.) and for another depth and region can be different (see Flatte et al. 1979).

the spatial spectrum) of the displacements with the correlation characteristics of the sound speed field. We now introduce the energy spectral density of the internal waves $G_l^s(\tilde{\vec{q}})$ which was introduced in the Sect. 2.5: $\left\langle S_l(\tilde{\vec{q}})S_m(\tilde{\vec{q}}')\right\rangle = G_l^s(\tilde{\vec{q}})\delta_{lm}\delta(\tilde{\vec{q}}-\tilde{\vec{q}}')$, where the displacement is represented in the form of a series in the vertical modes of the internal waves $\Phi_{l\tilde{\vec{q}}}(z)$:

$$\zeta(\vec{r},z) = \sum_l \int S_l(\tilde{\vec{q}})\tilde{\Phi}_{l\tilde{\vec{q}}}(z)\exp(i\tilde{\vec{q}}\vec{r})d\tilde{\vec{q}}, \tag{5.3}$$

l is the index of the gravitational mode of the internal waves, and $\tilde{\vec{q}}$ is its wave vector in the horizontal plane. For $G_l^s(\tilde{\vec{q}})$ in deep water, there exists an expression, proposed by Garrett and Munk (1975), giving the energy distribution of the gravitational modes and isotropic in space. As mentioned in Chap. 2, in shallow water this spectrum is not suitable because of the presence of considerable anisotropy and the predominance of the first gravitational mode. This corresponds to the exclusion of the sum from expression (5.3) (only the gravitational mode $\Phi_{1\tilde{\vec{q}}}(z)$ remains) and integration over the angle (with some weighting function to describe anisotropy) in the horizontal plane. Using the experimentally observed fact that the energy density of internal waves of comparatively small amplitude (the background linear internal waves) in shallow water decreases as a function of approximately $\tilde{q}^{-p}$ $(p \sim 2)$, for background waves in shallow water, we adopt the spectrum $G_l^s(\tilde{q}) = 2G_0\left\langle\zeta_0^2\right\rangle\tilde{q}^{-p}$. Here G_0 is a normalization constant, and $\left\langle\zeta_0^2\right\rangle$ is the variance at some fixed depth. Estimates of the longitudinal correlation length using this theoretical form give values from several hundred meters to a kilometer for background internal waves.

- *Bottom roughness*. According to the results of numerous experimental studies (e.g., Kuperman and Jensen (ed) 1980; Hampton 1973) the surface of the ocean bed is rather smooth, and the depth at comparatively large horizontal distances usually differs by at most by a few tens of meters from some average level. In addition, the characteristic horizontal scales of roughnesses vary over broad ranges. It is useful to distinguish the following characteristic scales (1) general relief, characteristic of the main geomorphological formations (ridges, valleys, troughs etc.), with longitudinal dimensions 10^4–10^5 m, (2) mesorelief, including relief forms of length 10^2–10^3 m, (3) microrelief, formed by ripple marks, boulders, hillocks, etc., with spatial scales of 10^{-2}–10 m. In this approach, general relief corresponds to regular variations of the parameters of a matched waveguide $H(r)$, and the other two components, mesorelief and microrelief, correspond to random fluctuations of the depth of the waveguide $h(r)$. Example of bottom roughness and correlation function was considered in Sect. 2.9. This random function is also characterized by a spatial spectral density, associated with the correlation function

$$\langle h(r)h(r')\rangle = \int_0^\infty B_h(\tilde{q})\cos(|r-r'|\tilde{q})d\tilde{q}. \tag{5.4}$$

At the present time, there are not so many available experimental data relating to the statistical properties of the shallow water sea bed as a randomly rough surface. According to our studies in one region of the Barents Sea, the mesoscale relief, captured by high-precision echo sounding, was characterized by a spectral density of deviations well approximated by the expression:

$$B_h(\tilde{q}) = 2\langle h_0^2 \rangle \frac{\tilde{q}}{a^2} \exp(-\tilde{q}^2/a^2), \tag{5.5}$$

where $\langle h_0^2 \rangle$ is the variance of the vertical deviations from the average level, and a is the spectral width, with an associated longitudinal correlation length $L_r \sim 1/2a$. In this case, we obtained values $\langle h_0^2 \rangle \sim 2\,\mathrm{m}^2$, $a \sim 2 \times 10^{-3}\,\mathrm{m}^{-1}$, $L_r \sim 10^3\,\mathrm{m}$.

- Sea surface disturbance, transforming the upper boundary of the waveguide into a randomly rough surface. Neglecting time dependence (supposing slow time variability) we take a displacement described by the random function $\zeta(\vec{r}) = \zeta(r, \theta)$ (θ being the azimuthal angle in the horizontal plane). This phenomenon is very well studied (e.g., Phillips 1977). Let us look at the spatial spectrum of a surface disturbance for a fixed direction, determined by the azimuthal angle θ. The correlation function $B_\zeta(r, \theta; r', \theta)$ and spectral density $G_\zeta(\tilde{q}, \theta)$, are determined by the equation

$$B_\zeta(r, \theta; r', \theta) = \langle \zeta(r, \theta)\zeta(r', \theta) \rangle = \frac{1}{2} \int_0^\infty G_\zeta(\tilde{q}, \theta) \cos(|r - r'|\tilde{q}) d\tilde{q}, \tag{5.6}$$

and is often described by theory based on the dispersion relation for surface waves $\Omega(\tilde{q})$ combined with the Pierson–Moscowitz (Pierson and Moscowitz 1969) frequency spectrum:

$$G_\zeta(\Omega) = a g^2 \Omega^{-5} \exp\left[-b(g/\Omega u)^4\right], \tag{5.7}$$

where $a = 8.1 \times 10^{-3}$, $b = 0.74$, $g = 9.8\,\mathrm{m/s}^2$, and u is the wind speed in m/s. In typical shallow water conditions ($u \leq 10 - 12\,\mathrm{m/s}$, depth $H \sim 100 - 200\,\mathrm{m}$), the dispersion equation $\Omega^2 = g\tilde{q}$ can be used for surface waves. In particular, for wind speed $u \sim 10\,\mathrm{m/s}$, $\theta \sim 0$, and the variance of the vertical displacements is $\langle \zeta^2 \rangle \sim 0.25\,\mathrm{m}^2$.

5.2 Description of a Random Sound Field by Coherent and Incoherent Parts

To describe the sound field in a random medium, we need to introduce two additional concepts: the coherent (average) and incoherent (fluctuating) parts of a random function. Let us examine this for a complex amplitude that can be either dependent or independent of time:

$$P(\vec{R},t) = \langle P(\vec{R},t)\rangle + \delta P(\vec{R},t), \tag{5.8}$$

where the bracket $\langle\rangle$ denotes averaging over a suitable ensemble of realizations. In this equation, $\langle \delta P(\vec{R},t)\rangle = 0$. The quantity $\langle P(\vec{R},t)\rangle$ is called the coherent part of the sound field; the quantity $\delta P(\vec{R},t)$ is called the incoherent (fluctuating) part. Correspondingly, we can consider the intensity of the sound field

$$I(\vec{R},t) = \frac{1}{2\rho c}\left|P(\vec{R},t)\right|^2. \tag{5.9}$$

It can also be divided as above. After ensemble averaging, we have

$$\langle I\rangle = \frac{1}{2\rho c}|\langle P\rangle|^2 + \frac{1}{2\rho c}\left\langle |\delta P|^2\right\rangle. \tag{5.10}$$

We can thus define the coherent and incoherent parts of average intensity as

$$\langle I\rangle = I_{\mathrm{c}} + I_i, \tag{5.11}$$

where

$$I_{\mathrm{c}} = \frac{1}{2\rho c}|\langle P\rangle|^2, \quad I_i = \frac{1}{2\rho c}\left\langle |\delta P|^2\right\rangle.$$

The correlation function of the sound field amplitude is defined according to (A.22). If we use our division into coherent and incoherent parts, we have for the correlation function

$$B(\vec{R},t;\vec{R}',t') = \frac{1}{2}\langle P(\vec{R},t)\rangle\langle P^*(\vec{R}',t')\rangle + \frac{1}{2}\langle \delta P(\vec{R},t)\delta P^*(\vec{R}',t')\rangle. \tag{5.12}$$

This allows us to introduce the correlation function of the fluctuating part

$$B_i(\vec{R},t;\vec{R}',t') = \frac{1}{2}\langle \delta P(\vec{R},t)\delta P^*(\vec{R}',t')\rangle, \tag{5.13}$$

which is also called the covariance.

The intensity can be divided into its average and fluctuating parts

$$I = \langle I\rangle + \delta I \tag{5.14}$$

so that the variance of the intensity can be obtained as

$$\sigma_I^2 = \langle I^2\rangle - \langle I\rangle^2. \tag{5.15}$$

We note that this last expression contains moments of the fourth order in pressure. For the characterization of fluctuations, the so-called scintillation index or *SI* (Ishimaru 1978) can be used[2]:

$$SI^2 = \frac{\langle I^2 \rangle - \langle I \rangle^2}{\langle I \rangle^2}.$$

5.3 Equations for Mode Interactions

To calculate the sound field in an irregular waveguide, it is common to represent it in the form of a decomposition into matched modes (3.106, 3.114), where the coefficients of the decomposition $C_l(r)$ may be assumed to be constant in the case of a smooth variation of the waveguide parameters. The smoothness criterion (i.e., the ability to neglect mode interactions) was previously formulated for the deterministic variation of these parameters [e.g., see the inequalities (3.148), (3.149), and (3.155–156)]. In the case where the interaction cannot be neglected, we must look for the $C_l(r)$ as functions of distance, and these functions describe mode coupling, i.e., a transfer of energy from one mode to another. In this case, the coupling is due to the deterministic variations of the waveguide parameters. In addition, we need to consider the case where the interaction of the modes may be due to stochastic variations of the waveguide parameters, e.g., the random inhomogeneities in the water column or its boundaries (roughness of the bottom or the surface) discussed in previous sections. This type of interaction causes the coefficients of the decomposition $C_l(r)$ to become random functions of the coordinate r, and, accordingly, the sound field itself, (3.106), becomes a random function of the coordinates. In order to determine the characteristics of the sound field, we then have to construct its various statistical moments. In this chapter, we shall work mainly with the second moment, the probability distribution of the intensity across the waveguide modes. Equations determining the variation of these moments with distance can be obtained in the form of a system of the so-called transport equations (kinetic equations) and, moreover, under a number of assumptions, may be brought into closed form, in the guise of a so-called diffusion approximation. This approach, apparently first proposed in Marcuse (1972) for optical waveguides, and for acoustic waveguides in Kohler and Papanikolau (1977) and Dozier and Tappert (1978), was used for the analysis of propagation in deep water in Artel'nyi and Raevskii (1984, 1987a, 1987b). Scattering by random inhomogeneities in deep-water waveguides was also considered in Flatte (1979) and Beran and Whitman (1978).

[2] In some literature, there is another definition $SI = \left(\langle I^2 \rangle - \langle I \rangle^2\right) \Big/ \langle I \rangle^2$.

Understanding the specifics of shallow water propagation, when taking into account the effect of random homogeneities, involves both quantitative (different characteristic values of the waveguide parameters) and qualitative differences from the deep water case. As we have already indicated on a number of occasions, when sound propagates in shallow water, absorption in the bottom, which is taken into account in terms of the mode attenuation coefficients, plays a crucial role. Of course, the variation of the mode structure of the field due to scattering (mode coupling) by random inhomogeneities is accompanied by its variation due to nonuniform (over modes) absorption, which complicates the reconstruction of the spatial structure of field in shallow water in comparison with deep water. Using the apparatus of the transport equations (kinetic equations) and/or the diffusion approach, we can include absorption in the bottom in a natural manner in the equations describing the variation of the field. This allows us to obtain an averaged picture of the variation of the spatial distribution of the statistical characteristics of the sound field as it propagates in the waveguide.

Formally, the presence of random inhomogeneities in the waveguide parameters listed above means that the values of $k^2(r,z)$ in (3.2) have fluctuating parts (or that the boundary conditions gain a fluctuating part[3]). In what follows, we shall use the "random inhomogeneous refractive index" $n_{\mathrm{R}}(r,z)$, composed of a square average part $n^2(r,z)$ and a fluctuating part $\mu(r,z)$ (we assume for now that $\mu = 0$ for $z>H$, i.e., that random inhomogeneities occur only in the water column). Thus,

$$k^2(r,z) = k^2 n_{\mathrm{R}}^2(r,z) = k^2[n^2(r,z) + \mu(r,z)], \quad |\mu| \ll \langle n_{\mathrm{R}}^2 \rangle = n^2, \quad \langle \mu \rangle = 0. \quad (5.16)$$

Typical SW environments lead to values of $\approx 10^{-2} - 10^{-3}$ for μ. Despite the smallness of μ, the effects of random inhomogeneities may not be small over sufficiently long tracks. The random part of the refractive index can be described by the correlation function

$$B(r,r';z,z') = \langle \mu(r,z)\mu(r',z') \rangle. \quad (5.17)$$

As far as the randomly inhomogeneous refractive index field is concerned, we shall assume that the following relation between the spatial scales of the random inhomogeneities holds. Specifically, let us suppose that the correlation radius of the random inhomogeneities in the longitudinal direction is $R_{||}$ and that it is $R_{\perp}$ in the transverse direction. Moreover, we take that the characteristic scale of variations of the waveguide parameters in the longitudinal direction is $\tilde{L}$ and that the characteristic distances of observation are r. We shall assume that

$$R_{\perp} \ll R_{||} \ll \tilde{L}, r. \quad (5.18)$$

[3] Stochastic boundary conditions lead to analogous equations for mode interactions and so will not be considered separately in what follows.

With this assumption, we can obtain equations describing the variation of the sound field from the source, propagating out from the source in the waveguide. We shall suppose, as in Sect. 3.2, that the sound propagates in a waveguide whose depth $H(r)$ varies smoothly with the distance, and that the deterministic (average) part of the refractive index also varies smoothly with distance. The sound field satisfies an equation analogous to (B.5) with corresponding boundary conditions on the surface (pressure-release boundary) and on the bottom (matching condition with the lower half-space, which may be chosen to be stratified). As in the previous section, we shall be interested in the intensity of the sound field generated by an acoustic source with frequency ω and power $\bar{W}_0$; therefore, we shall be dealing with a sound pressure with complex amplitude, analogous to (3.25) or (4.91) (the temporal dependence $\sim e^{-i\omega t}$ is omitted)

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right)+\frac{\partial^2}{\partial z^2}+k^2\left(n^2(r,z)+\mu(r,z)\right)\right]P(r,z)=0. \tag{5.19}$$

As before, we represent the complex amplitude of the sound field in the form of a decomposition into the modes of an unperturbed (without random inhomogeneities) waveguide:

$$P(r,z)=\sqrt{\rho_0 c_0 \bar{W}_0}\sum_l \frac{\Psi_l(r;z)}{\sqrt{r q_l(r)}}\left[C_l^+(r)\,e^{\,i\int_0^r q_l(\rho)d\rho}+C_l^-(r)\,e^{\,-i\int_0^r q_l(\rho)d\rho}\right], \tag{5.20}$$

where $\Psi_l(r;z)$ are the eigenfunctions of the matched waveguide, which we determined earlier in (3.127) and (3.128). The difference from the analogous formulas (3.137) and (3.133) is that here the coefficients of the decomposition $C_l^{\pm}(r)$ are random functions. Moreover, the exponents in the integrands contain the real part of an eigenvalue. We note that both the real and the imaginary parts of the eigenvalues are smooth functions of the coordinate r. Substituting the decomposition (5.20) in (5.19), we obtain a system of stochastic differential equations for the random functions $C_l^{\pm}(r)$. Neglecting backscattering terms, coefficients containing C_l^- are dropped from the system of equations, and the remaining coefficients C_l^+ are then just denoted by C_l. In addition, the following simplifying approximations are made. In obtaining the derived quantities, we will neglect any adiabatic dependence on r: $\frac{\partial q_l}{\partial r}, \frac{\partial \Psi_l}{\partial r}$; also, we assume that $\left|\frac{d^2C_l}{dr^2}\right| \ll \left|q_l\frac{dC_l}{dr}\right|$. Taking this into account, we obtain a system of stochastic differential equations:

$$\frac{dC_l(r)}{dr}=-\frac{\gamma_l(r)}{2}C_l(r)+i\sum_m V_{lm}(r)\,e^{\,i\int_0^r q_{ml}(\rho)d\rho}C_m(r), \tag{5.21}$$

where the matrix elements of the mode coupling interaction V_{lm} are determined by the equation

$$V_{lm}(r) = \frac{k^2}{2\sqrt{q_l q_m}} \int_0^\infty \frac{\rho}{\rho(r,z)} \Psi_l(r;z)\mu(r,z)\Psi_m(r;z)dz \tag{5.22}$$

and are random functions of r. Also, $q_{ml}(r) = q_m(r) - q_l(r)$.

The derivation of the equations for the average characteristics of the sound field is performed in the standard way. Instead of (5.21), we write down the equivalent integral equation

$$C_l(r) = C_l(0) - \int_0^r \frac{\gamma_l(r')}{2} C_l(r')dr' + i\sum_m \int_0^r V_{lm}(r')\, e^{i\int_0^{r'} q_{ml}(\rho)d\rho} C_m(r')dr'. \tag{5.23}$$

After one iteration [where we replace any C on the right-hand side with the full expression (5.23)], we obtain

$$\begin{aligned} C_l(r) = C_l(0) &- \int_0^r \frac{\gamma_l(r')}{2} C_l(r')dr' + i\sum_m \int_0^r V_{lm}(r')\, e^{i\int_0^{r'} q_{ml}(\rho)d\rho} C_m(0)dr' \\ &- i\sum_m \int_0^r dr' \int_0^{r'} dr'' \frac{\gamma_m(r'')}{2} V_{lm}(r')\, e^{i\int_0^{r'} q_{ml}d\rho} C_m(r'') \\ &- \sum_{m,n} \int_0^r dr' \int_0^{r'} dr'' V_{lm}(r')V_{mn}(r'')\, e^{i\int_0^{r'} q_{ml}(\rho)d\rho + i\int_0^{r''} q_{nm}(\rho)d\rho} C_n(r'') \end{aligned} \tag{5.24}$$

Averaging over instantaneous realizations, we obtain the following equation:

$$\begin{aligned} \langle C_l(r)\rangle = \langle C_l(0)\rangle &- \int_0^r \frac{\gamma_l(r')}{2}\langle C_l(r')\rangle dr' - i\sum \int_0^r dr' \int_0^{r'} dr'' \frac{\gamma_m(r'')}{2}\langle V_{lm}(r')C_m(r'')\rangle\, e^{i\int_0^{r'} q_{ml}d\rho} \\ &- \sum_{m,n} \int_0^r dr' \int_0^{r'} dr'' \langle V_{lm}(r')V_{mn}(r'')C_n(r'')\rangle\, e^{i\int_0^{r'} q_{ml}(\rho)d\rho + i\int_0^{r''} q_{nm}(\rho)d\rho}. \end{aligned} \tag{5.25}$$

In the Burre approximation, we separate the average expression from that in the integrand

$$\langle V_{lm}(r')V_{mn}(r'')C_n(r'')\rangle = \langle V_{lm}(r')V_{mn}(r'')\rangle\langle C_n(r'')\rangle$$
$$\langle V_{lm}(r')C_l(r'')\rangle = \langle V_{lm}(r')\rangle\langle C_l(r'')\rangle = 0 \quad . \tag{5.26}$$

The correlation of the random perturbation function in (5.26) can be represented using (5.22) and (5.17), giving

$$\langle V_{lm}(r')V_{mn}(r'')\rangle = \frac{k^4}{4q_m\sqrt{q_l q_n}} \int_0^\infty dz' \int_0^\infty dz'' \frac{\rho^2}{\rho(r',z')\rho(r'',z'')}$$
$$\Psi_l(r';z')\Psi_m(r';z')B(r',r'';z',z'')\Psi_m(r'';z'')\Psi_n(r'';z'')\,. \tag{5.27}$$

We shall further use the assumption that the correlation function B is a sharper function of its arguments $r' - r''$ than the dependence of the eigenfunctions on the longitudinal coordinate. Consequently, in the integral (5.27), we can set the value of the longitudinal coordinate equal to r' for the eigenfunctions. By virtue of the slowness of the density variations, the same can be done for the functions of ρ. This means that the correlation in (5.27) can be represented in the form $\langle V_{lm}(r')V_{mn}(r'')\rangle = B_{lm,mn}(r';r'-r'')$, where the function $B_{lm,mn}$ is a smooth function of the first argument and a sharp function of the second argument. After differentiating with respect to r, we obtain

$$\frac{d\langle C_l(r)\rangle}{dr} = -\frac{\gamma_l(r)}{2}\langle C_l(r)\rangle - \sum_{m,n}\int_0^r B_{lm,mn}(r;r-r')\,e^{i\int_0^r q_{ml}(\rho)d\rho + i\int_0^{r'} q_{nm}(\rho)d\rho}\langle C_n(r')\rangle dr'. \tag{5.28}$$

Using the smoothness of the function $\langle C_n(r')\rangle$ in comparison with the correlation, we can take it out of the integral for values $r = r'$. For sufficiently large values of r, the correlation function is very small and so the integral in (5.25) can be extended to infinity. We then obtain

$$\frac{d\langle C_l(r)\rangle}{dr} = -\frac{\gamma_l(r)}{2}\langle C_l(r)\rangle - \sum_{m,n}\left[\int_0^\infty B_{lm,mn}(r;r-r')\,e^{i\int_r^{r'} q_{nm}(\rho)d\rho}dr'\right]\langle C_n(r)\rangle e^{i\int_0^r q_{nl}(\rho)d\rho}. \tag{5.29}$$

With the notation

$$A_{ln}(r) = \sum_m \int_0^\infty B_{lm,mn}(r;r-r')\,e^{i\int_r^{r'} q_{nm}(\rho)d\rho}dr', \tag{5.30}$$

we finally obtain a system of equations for the average amplitudes:

$$\frac{d\langle C_l(r)\rangle}{dr} = -\frac{\gamma_l(r)}{2}\langle C_l(r)\rangle - A_{ll}\langle C_l(r)\rangle - \sum_{n\neq l} A_{ln}(r)\, e^{i\int_0^r q_{nl}(\rho)d\rho}\langle C_n(r)\rangle. \quad (5.31)$$

The right-hand side of the system of equations describes the transfer of energy from the modes with index n to the mode with index l as a result of mode-coupling interactions (scattering by random inhomogeneities). The system (5.31) specifically describes the change in the coherent component of the field (or the average amplitude). In the sum on the right-hand side of (5.31), we separate out the diagonal term, corresponding to scattering at zero angle. In the language of modes, this corresponds to the interaction of the mode with index l with the same mode of index l as a result of random inhomogeneities.

The coefficient $A(r)$ is, generally speaking, complex; this corresponds to attenuation of the coherent part of the amplitude and also to the change in its phase under the effects of statistical perturbation. Thus, we can introduce an additional coefficient Γ_l for the attenuation of the coherent component of the mode with index l due to scattering by random inhomogeneities

$$\Gamma_l(r) = 2\,\mathrm{Re}\, A_{ll}(r) = \sum_m a_{lm}(r), \quad (5.32)$$

where we have denoted

$$a_{lm}(r) = 2\,\mathrm{Re}\int_0^\infty B_{lm}(r,r')\, e^{i\int_r^{r+r'} q_{lm}(\rho)d\rho} dr', \quad (5.33)$$

$$B_{lm}(r,r') = B_{lm,ml}(r,r') = \langle V_{lm}(r)V_{ml}(r-r')\rangle. \quad (5.34)$$

The sum on the right-hand side of (5.31) can be omitted if we consider sufficiently large distances, since it contains rapidly oscillating terms. Generally speaking, if perturbation of the sound-speed profile depends periodically on the spatial coordinate and the spatial period is close to some of the scales of interference beating determined by the differences between the eigenvalues, then the interaction between separate modes [terms under the sum in (5.31)] can be significant. In this case, we have the so-called "resonance" interaction or coupling between modes (Rogers et al. 2000).

In the other (nonresonant) case for the amplitudes of the average (coherent) component, we obtain the following mode equation:

$$\frac{d|\langle C_l(r)\rangle|}{dr} = -\frac{\gamma_l(r)+\Gamma_l(r)}{2}|\langle C_l(r)\rangle|. \quad (5.35)$$

5.4 Equations for Intensity

In this section, we will obtain equations for the averaged square quantities of the field. As a preliminary, we note that since the sound field $P(r,z)$ in (5.20) is random, then when averaging squared quantities, it is sensible to identify mean (coherent) and fluctuating (incoherent) components. Thus,

$$C_l(r) = \langle C_l(r)\rangle + \delta C_l(r), \quad \langle \delta C_l(r)\rangle = 0, \tag{5.36}$$

respectively, and for the squares (of the mode intensities), we get

$$\left\langle |C_l(r)|^2 \right\rangle = |\langle C_l(r)\rangle|^2 + \left\langle |\delta C_l(r)|^2 \right\rangle. \tag{5.37}$$

If we define the square of the modulus of the mean amplitude to be a quantity proportional to the intensity of the coherent component, for the mode with index l

$$|\langle C_l(r)\rangle|^2 = U_l(r). \tag{5.38}$$

Then from (5.35), we immediately have the equation

$$\frac{dU_l(r)}{dr} = -[\gamma_l(r) + \Gamma_l(r)]U_l(r), \tag{5.39a}$$

which must be supplemented by the "initial" conditions, determined by the source

$$U_l(0) = |C_l(0)|^2. \tag{5.39b}$$

Let us now consider the variation of the averaged square of the mode amplitudes, in other words, the total intensity of the sound, delivered by the mode with index l traveling along the waveguide. For this, we shall integrate (5.23) from r to $r+\Delta r$ and iterate it once, thus obtaining the integral equation

$$\begin{aligned} C_l(r+\Delta r) = C_l(r) &- \int_r^{r+\Delta r} \frac{\gamma_l(r')}{2} C_l(r')dr' + i\sum_m \int_r^{r+\Delta r} V_{lm}(r')\, e^{i\int_r^{r'} q_{ml}(\rho)d\rho} C_m(r)dr' \\ &- i\sum_m \int_r^{r+\Delta r} dr' \int_r^{r'} dr'' \frac{\gamma_m(r'')}{2} V_{lm}(r')\, e^{i\int_r^{r'} q_{ml}(\rho)d\rho} C_m(r'') \\ &- \sum_{m,n} \int_r^{r+\Delta r} dr' \int_r^{r'} dr'' V_{lm}(r')V_{mn}(r'')\, e^{i\int_r^{r'} q_{ml}(\rho)d\rho + i\int_r^{r''} q_{nm}(\rho)d\rho} C_n(r''). \end{aligned} \tag{5.40}$$

We then multiply this rather lengthy expression by its complex conjugate and consider only quadratic terms in the perturbation. After averaging over realizations, we obtain

$$\left\langle |C_l(r+\Delta r)|^2 \right\rangle = \left\langle |C_l(r)|^2 \right\rangle - \frac{1}{2}\int_r^{r+\Delta r} \gamma_l(r')\left[\left\langle C_l(r')C_l^*(r)\right\rangle + \left\langle C_l(r)C_l^*(r')\right\rangle\right]dr'$$
$$+ \sum_{m,m'} \int_r^{r+\Delta r} dr' \int_r^{r+\Delta r} dr'' \langle V_{lm}(r')V_{lm'}(r'')\rangle \left\langle C_m(r')C_{m'}^*(r'')\right\rangle e^{i\int_r^{r'} q_{ml}(\rho)d\rho - i\int_r^{r''} q_{m'l}(\rho)d\rho}$$
$$- 2\,\mathrm{Re}\left\{ \sum_{m,n} \int_r^{r+\Delta r} dr' \int_r^{r'} dr'' \langle V_{lm}(r')V_{mn}(r'')\rangle \left\langle C_n(r'')C_l^*(r)\right\rangle e^{i\int_r^{r'} q_{ml}(\rho)d\rho + i\int_r^{r''} q_{nm}(\rho)d\rho} \right\} \tag{5.41}$$

In what follows, we shall use the simplification that the exponentials in the first and second sums in (5.41) will oscillate rapidly for $m' \neq m$ and $n \neq l$, respectively. Discarding terms with rapidly oscillating exponentials, we find that in the sums, only the quantities with $m' = m$ and $n = l$ remain. Let us denote

$$\left\langle |C_l(r)|^2 \right\rangle = W_l(r). \tag{5.42}$$

The quantity $W_l(r)$ on the right-hand side of (5.42) has the sense of the weight of the mode with index l (the term "population density" used in quantum mechanics is more appropriate) in the modal decomposition of the field, and determines the intensity produced by the mode with index l.

We shall also use the assumption that the variation of the mode amplitudes is comparatively slow (in comparison with the correlation scale of the inhomogeneities). Subsequent transformations of expression (5.41) will be based on assumptions about the "discretization step" Δr. We shall suppose that there exists a scale Δr, greater than the longitudinal radius of correlation of the random inhomogeneities $R_{||}$, but less than the characteristic scale of deterministic longitudinal variations of the waveguide parameters $\tilde{L}$. In more detail, $\tilde{L}_H \approx |H/H'|$, $\tilde{L}_c \approx |c/c'|$ (for the depths and sound-speed profile) and correspondingly for the scales of variation in the eigenfunctions and eigenvalues, $\tilde{L}_q \approx |q/q'|$ (where the prime denotes the derivative with respect to r). One can easily give an approximate estimate of the extent to which these conditions are satisfied. As we have already mentioned, typical values of $R_{||}$ are 100 m, whereas $\tilde{L}_H, \tilde{L}_c$ and, $\tilde{L}_q$ are of the order of several kilometers so that the inequality

$$R_{||} \ll \Delta r \ll \tilde{L}_H, \tilde{L}_c, \tilde{L}_q, r \tag{5.43}$$

is completely feasible in real conditions. This means that we can take C_l, W_l, and q_{lm} out from under the integrals in the range from r to $r + \Delta r$ for

most values of the argument r. Thus, for the mode intensities, we obtain a system of transfer equations (kinetic equations):

$$W_l(r+\Delta r) = W_l(r) - \gamma_l(r)W_l(r)\Delta r + \Delta r \sum_m a_{lm}(r)[W_m(r) - W_l(r)], \tag{5.44}$$

where the coefficients a_{lm} are defined in (5.33).

If we are interested in the field at a sufficiently large distance from the source, according to the inequalities (5.43), Δr may be assumed to be infinitely small. After moving to the limit $\Delta r \to 0$ in (5.44), we obtain an equation for the change in energy in the mode with index l.

$$\frac{dW_l}{dr} = -\gamma_l W_l + \sum_m a_{lm}(W_m - W_l), \tag{5.45a}$$

$$W_l(0) = |C_l(0)|^2. \tag{5.45b}$$

We note that all the quantities in (5.45a, b) depend on r. Summing over l on the right- and left-hand sides of (5.45a, b), we obtain an equation determining the decrease in the total sound intensity as it propagates in the waveguide:

$$\frac{d}{dr}\sum_l W_l(r) = -\sum_l \gamma_l(r)W_l(r). \tag{5.46}$$

We note that if $\gamma_l = 0$, then as $r \to \infty$, we have $\frac{dW_l}{dr} \to 0$, and from the equality to zero of the right-hand side of (4.38), we see that $W_m = W_l = \text{const}$, i.e., the mode intensities are smoothed out (equipartitioned) due to mode interactions. Using the quantity $W_l(r)$, we can find a number of characteristics of the sound field in the waveguide. For example, the vertical correlation function, averaged in the longitudinal direction over a sufficiently large interval, is

$$\begin{aligned} B(r;z,z') &= \frac{1}{\sqrt{\rho\rho' cc'}}\left\langle \operatorname{Re} p(r,z)\, e^{-i\omega t} \operatorname{Re} p(r,z')\, e^{-i\omega t}\right\rangle \\ &= \frac{\rho_0 c_0 \bar{W}_0}{2r\sqrt{\rho\rho' cc'}} \sum_l \frac{W_l(r)}{q_l(r)} \Psi_l(r;z)\Psi_l(r;z') \\ &= \frac{\bar{W}_0}{2r} \sum_l \frac{W_l(r)}{q_l(r)} \Psi_l(r;z)\Psi_l(r;z'). \end{aligned} \tag{5.47}$$

We recall in the above that $c = c(z), c_0 = c(z_0), c' = c(z')$, and analogously for the equations for ρ. The last identity was obtained by neglecting the difference in the values of the density and the sound speed at different points in the coefficient

(analogously to Sect. 4.6). The normalization in (5.47) was chosen so that for $z = z'$, one obtains from (5.47) that the sound intensity averaged over the longitudinal interference structure is

$$I(r,z) = \frac{1}{\rho c}\left\langle \left[\operatorname{Re} p(r,z)\, e^{-i\omega t}\right]^2 \right\rangle = \frac{\bar{W}_0}{2r}\sum_l \frac{W_l(r)}{q_l(r)}\Psi_l^2(r,z). \tag{5.48}$$

Comparing this equation with (4.94), we see that without random inhomogeneities $W_l(r) = \Psi_l^2(0,z_0)\exp\left[-\int\limits_0^r \gamma_l(r')dr'\right]$. An expression for the intensity, averaged over the position of the receiver, can be obtained using the WKB approximation for $\Psi_l(r;z)$:

$$\bar{I} = \frac{\bar{W}_0}{r}\sum_l \frac{W_l(r)}{\sigma_l(r,z)D_l(r)}, \quad \sigma_l(r,z) = \sqrt{k^2n^2(z) - q_l^2(r)}.$$

$$D_l(r) = 2q_l(r)\int\limits_{\underline{z}}^{\bar{z}} \frac{dz}{\sigma_l(r,z)}. \tag{5.49}$$

Equation (5.49) also describes the intensity averaged over the position of the source if we take that $W_l(0) = \frac{2q_l(0)}{\sigma_l(0,z_0)D_l(0)}$.

Analogously to the derivation of (5.48), one can obtain the averaged intensity of the coherent component of the sound field

$$I_{\mathrm{c}}(r,z) = \frac{\bar{W}_0}{2r}\sum_l \frac{U_l(r)}{q_l(r)}\Psi_l^2(r,z), \tag{5.50}$$

where we define the difference between the total and the coherent components to be the incoherent (fluctuating) component of the intensity

$$I_n(r,z) = I(r,z) - I_{\mathrm{c}}(r,z) = \frac{\bar{W}_0}{2r}\sum_l \frac{W_l(r) - U_l(r)}{q_l(r)}\Psi_l^2(r,z). \tag{5.51}$$

If we again use the WKB approximation for the modes, then we can obtain expressions analogous to (5.49) for the coherent and the incoherent components. Thus, the mode intensities of the sound field in a randomly inhomogeneous waveguide vary according to (5.39a, b) and (5.45a, b). It is clear that as the sound propagates, energy is redistributed between modes (for example, the total intensity is smoothed) and that there is also redistribution between the coherent and incoherent components. More precisely, the proportion of the coherent component is decreased and the fluctuating component grows.

Let us consider a simple example of the coupling of intensities for the hypothetical case where we have only two propagating modes. The equations then have the form

$$\frac{dW_1}{dr} = -\gamma_1 W_1 + a(W_2 - W_1),$$
$$\frac{dW_2}{dr} = -\gamma_2 W_2 + a(W_1 - W_2). \tag{5.52}$$

We can also add the initial conditions

$$W_{1,2}(0) = W_{1,2}^0 \tag{5.53}$$

If we assume that the strength of the interaction does not depend on range, this system can be solved analytically. Taking into account the initial conditions, we have

$$W_1(r) = \frac{W_1^0}{2K}\left[(\tilde{\gamma}_1 + k_1)\, e^{k_2 r} - (\tilde{\gamma}_1 + k_2)\, e^{k_1 r}\right] + \frac{aW_2^0}{2K}\left(e^{k_1 r} - e^{k_2 r}\right) \tag{5.54a}$$

$$W_2(r) = W_1^0 \frac{(\tilde{\gamma}_1 + k_1)(\tilde{\gamma}_1 + k_2)}{2aK}(e^{k_2 r} - e^{k_1 r}) + W_2^0 \frac{1}{2K}\left[(\tilde{\gamma}_1 + k_1)\, e^{k_1 r} - (\tilde{\gamma}_1 + k_2)\, e^{k_2 r}\right], \tag{5.54b}$$

where

$$\tilde{\gamma}_{1,2} = \gamma_{1,2} + a, \quad k_{1,2} = -\frac{\tilde{\gamma}_1 + \tilde{\gamma}_2}{2} \pm K, \quad K = \sqrt{\left(\frac{\tilde{\gamma}_1 - \tilde{\gamma}_2}{2}\right)^2 + a^2}.$$

We can consider an even simpler instructive case where $W_1^0 = 1$, $W_2^0 = 0$. The existence of inhomogeneities leads to the excitation of the second mode due to mode coupling, and moreover, both modes should decrease as a result of attenuation. Thus we see

$$W_1(r) = \frac{(\tilde{\gamma}_1 + k_1)}{2K}\, e^{k_2 r} - \frac{(\tilde{\gamma}_1 + k_2)}{2K}\, e^{k_1 r}, \tag{5.55a}$$

$$W_2(r) = \frac{a}{K}\, e^{-\frac{\tilde{\gamma}_1 + \tilde{\gamma}_2}{2} r} \sinh Kr. \tag{5.55b}$$

In this case, we see a nonmonotonic behavior of the second mode. Initially, the amplitude of this mode increases due to mode coupling and the transfer of energy from the first mode. Next, the decrease in both modes takes place due to attenuation. Plots describing the behavior of the modal intensity for this case are shown in Fig. 5.2.

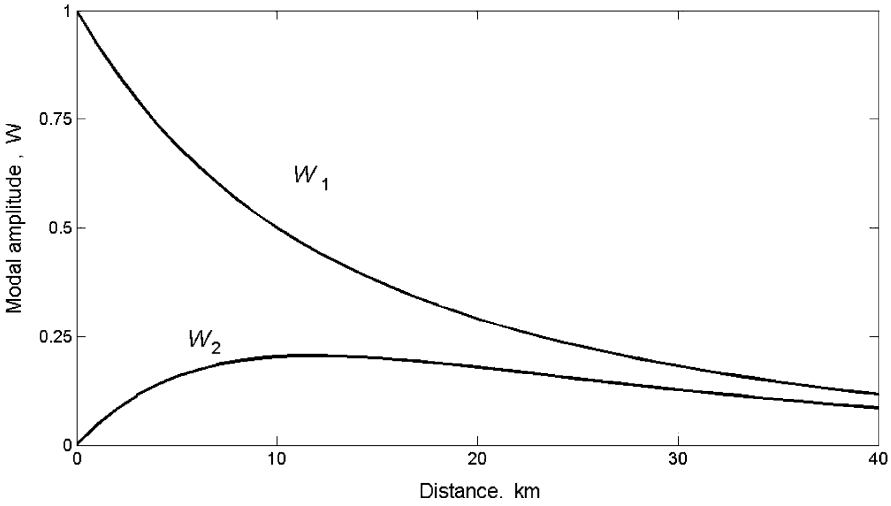


Fig. 5.2 Modal amplitudes a for two-mode waveguide. $W_1^0 = 1$, $W_2^0 = 0$, $\gamma_1 = 3 \times 10^{-5}\mathrm{m}^{-1}$, $\gamma_2 = 6 \times 10^{-5}\mathrm{m}^{-1}$, $a = 5 \times 10^{-5}\mathrm{m}^{-1}$

5.5 The Diffusion Equation and Averaged Decay Laws

The nature of the solution of the system of (5.39a, b) and (5.45a, b) is determined by the values of the elements a_{lm}, which describe the mode interaction. For a very broad class of random inhomogeneities, the nature of the interactions is such that only adjacent modes interact, i.e., the only modes effectively interacting with mode l are modes $l-1$ and $l+1$. In the language of rays, this corresponds to the predominance of scattering at small angles. The possibility of using this approximation is determined by the dependence of the correlation function for the random inhomogeneities on the transverse coordinates, defining the integral (5.27) and, correspondingly, (5.45a, b). In particular, it is clear that if we have δ-correlation with respect to z, then all modes interact effectively and the above approach is not suitable. At the same time, if the transverse timescale is large, then we can go beyond the limits of (5.18) so that the region of applicability of this approach lies at some intermediate values of $R_\perp$. In Sidenko (1989), it is shown based on numerical analysis that at values ($R_\perp/H \leq 0.1$), which often arise in real situations and when there are not too many high modes, the contribution of nonadjacent modes is at most 10%. This topic is also discussed in Kohler and Papanikolau (1977).

Thus, in the system of transfer equations for the intensity, we can be left with the interaction of adjacent modes only. In this case, the system of equations can be written in the form

$$\begin{aligned}
\frac{dW_1(r)}{dr} &= -\gamma_1(r)W_1(r) + a_{21}(r)[W_2(r) - W_1(r)],\\
\frac{dW_l(r)}{dr} &= -\gamma_l(r)W_l(r) + a_{l+1,l}(r)[W_{l+1}(r) - W_l(r)] - a_{l,l-1}(r)[W_l(r) - W_{l-1}(r)],\\
\frac{dW_M(r)}{dr} &= -\gamma_M(r)W_M(r) + a_{M,M-1}(r)[W_M(r) - W_{M-1}(r)].
\end{aligned} \tag{5.56}$$

where M is the number of propagating modes.

The system of (5.45a, b) or (5.56) can be solved numerically and then the quantities $W_l(r)$ can be used to find the characteristics of the sound field. The authors have done this (Katsnelson et al. 1998) to calculate the intensity of sound propagating on an acoustic track, taking into account inhomogeneities in the bottom. The latter was measured with high precision echosounding, which was also used to construct the precise mesoscale relief of the track. Calculations of the intensity were made based on numerical solution of the system (5.45a, b) or (5.56) along the track in partition areas with step length less than the characteristic longitudinal scale of the inhomogeneities, of the order of 10 m. The intensity as a function of distance to the source thus constructed is, strictly speaking, a deterministic quantity. In addition, this dependence is extremely irregular and gives an idea of the physical medium fluctuations; this is due to the statistical roughness of the bottom in this case.

In addition, equations can be obtained which may enable us to construct analytic and semianalytic solutions. One helpful step in this direction is the assumption that the number of propagating modes is sufficiently large so that the distances between eigenvalues are sufficiently small. Then the discrete variable, the mode index n, can be replaced by a continuous variable p. This hypothesis corresponds to the transition to ray formulae with an averaged description of the propagation of sound (Sect. 4.6), and has the same extent of applicability. In this case, the system of (5.56) is a finite difference approximation to some differential equation with continuous variables. Accordingly, we make the substitution

$$\begin{aligned}
\Delta l &= (l+1) - l = l - (l-1) \to dp,\\
\Delta F_l &= F_{l+1} - F_l \to dF.
\end{aligned} \tag{5.57}$$

Denoting $a_p(r) = a_{p,p-1}(r) = a_{p-1,p}(r)$ from (5.56) for indices $2 \le n \le N$, we obtain a differential equation

$$\frac{\partial W_p(r)}{\partial r} = -\gamma_p W_p(r) + \frac{\partial}{\partial p}\left(a_p \frac{\partial W_p(r)}{\partial p}\right), \tag{5.58}$$

where for $a_p(r)$ we can obtain the expression

$$a_p(r) = \frac{k^4}{q_p q_{p\pm1}} \int_{-\infty}^{\infty} dr' \int_0^{H(r)} \frac{\rho}{\rho(z)} dz \int_0^{H(r)} \frac{\rho}{\rho(z')} dz'$$

$$\Psi_p(r,z)\Psi_p(r+r',z')B(r,r';z,z')\Psi_{p\pm1}(r,z)\Psi_{p\pm1}(r+r',z')\, e^{i\int_r^{r+r'} (q_{p\pm1}-q_p)dr'}$$

$$\sim \frac{k^2}{q_p^2} \int_{-\infty}^{\infty} dr' \int_0^{H(r)} \frac{\rho}{\rho(z)} dz \int_0^{H(r)} \frac{\rho}{\rho(z')} dz'\ \Psi_p^2(r,z)\Psi_p^2(r+r',z')B(r,r';z,z') \tag{5.59}$$

The limits of integration in (5.59) come from the fact that we are considering random inhomogeneities in the water column only. The first and last equations of the system (5.56) lead to boundary conditions for the parameter p, the permissible values of which correspond precisely to the mode indices from 0 to some value M, determined by the sound frequency and the waveguide parameters. Modes with indices greater than M are "outflowing energy" and constitute one of the mechanisms for the loss of sound in the waveguide. The "de-excitation" of the modes when this index is reached is described in Kohler and Papanikolau (1977) as radiation losses and corresponds to a certain term on the right-hand side of the last equation of the system (5.56). Specifically, this attenuation corresponds to the coefficient γ_M. For the case of shallow water, in the presence of marked absorption in the bottom, the role of these radiation losses is small, since as a result of growing with the mode index the coefficients γ_l may decay long before the indices corresponding to the exit of a wave from the waveguide are reached. Thus, the range of variation of the parameter p is defined by the inequality $0 \le p \le p_c$, where p_c is determined by the number of propagating modes. One can compare, for example, the grazing angle of a ray corresponding to a given mode, to the continuous variable p. In particular, for a Pekeris waveguide, the continuous variable p can be determined using the relation $p = \frac{kH}{\pi} \sin\chi \approx \frac{kH}{\pi}\chi$, where χ is the grazing angle of a ray corresponding to the mode with index p.

The boundary conditions at $p = 0$ define the behavior of the coefficient a_p in this region. From (5.59) one can establish that this quantity behaves proportionally to k^2as a function of the wave vector. The condition of the finiteness of the right-hand side of the expression in (5.58) leads to the condition

$$\left(\frac{\partial W_p}{\partial p}\right)_{p=0} = 0. \tag{5.60}$$

A second condition, whereby mode attenuation in a shallow water waveguide is extended to $p \to \infty$, gives

$$W_{p=\infty} = 0. \tag{5.61}$$

Thus, if we need to compute any of the observed characteristics of the sound field based on $W_p(r)$, instead of summing over modes, we can integrate over the

continuous variable p. In particular, the averaged intensity (5.49) will be determined by the equation

$$\bar{I} = \frac{\bar{W}_0}{r} \int dp \frac{W_p(r)}{\sigma_p(z) D_p} \tag{5.62}$$

which is analogous to (4.98) describing the averaged law for the decay in a ray approximation. If we use the relation given earlier, between the continuous variable p and the ray grazing angle χ, these formulae can be reduced to each other. If we use the ray approach from the beginning, then an equation analogous to (5.62) can be obtained by considering the propagation of sound in the waveguide as a diffusion of the intensity at the ray grazing angle, as is done in free space (Chernov 1975).

To conclude this section, we give analogous equations for the coherent and incoherent components of the averaged intensity. These comprise a differential equation

$$\frac{dU_p(r)}{dr} = -(\gamma_p + \Gamma_p) U_p(r), \tag{5.63}$$

analogous to (5.39a) and integral equations

$$\bar{I}_{\mathrm{c}} = \frac{\bar{W}_0}{r} \int dp \frac{U_p(r)}{\sigma_p(z) D_p}, \quad \bar{I}_{\mathrm{n}} = \frac{\bar{W}_0}{r} \int dp \frac{W_p(r) - U_p(r)}{\sigma_p(z) D_p}, \tag{5.64}$$

where the functions $U_p(r)$ and $\Gamma_p(r)$ are analogous to $U_l(r)$ and $\Gamma_l(r)$ (5.32) with continuous values of the index, having the same sense as for $W_p(r)$.

In the next section, we will consider a number of examples of the solutions of (5.58) for simple models of a shallow water sound channel. In these simple models, one can obtain a generalization of the averaged decay laws if the channel includes random inhomogeneities (whose detailed nature is not important in this case). It is only important that the characteristics of the inhomogeneities satisfy the conditions of applicability of the theory given earlier [see (5.43) and (5.18)].

5.6 Some Examples

5.6.1 Waveguide of Constant Depth

Let us consider the case of a waveguide having a constant depth along the track ($H = \text{const}$) and a constant profile of the average sound speed [refractive index $n = 1$ in (5.16)], i.e., the Pekeris model plus random inhomogeneities. The bottom, as in Sect. 3.2, is a homogeneous fluid, absorptive half-space with some given values of the density and sound speed and a corresponding absorption coefficient α. The

modes of this "deterministic" waveguide can be constructed in analytic form from (3.36) with the appropriate formulae for the normalization coefficients and the eigenvalues. The mode attenuation coefficient γ_l appearing in (5.56), which is very important for us, is determined by (4.24). Based on what has been done, instead of the discrete variable for the mode index l, we will introduce a continuously varying p and construct the function γ_p. We do this by simplifying (3.71), if possible. The simplifying factor is the assumption that we have sufficiently many modes propagating at small grazing angles, which means that assuming $\sigma_l \approx \frac{l\pi}{H} \ll q_l \approx k$, $N_l^2 \approx \frac{2}{H}$, and $\kappa_l \approx k\sqrt{1-n^2}$ $(n = \frac{c}{c_1})$ for γ_p, we have

$$\gamma_p = \frac{s\pi^2 p^2}{k^2 H^3}. \tag{5.65}$$

In the above, $s = \alpha \frac{n^2 m}{(1-n^2)^{3/2}}$ or, which is equivalent for small α, it is determined by (4.23). The diffusion coefficient $a_p = a_{p,p-1} = a_{p,p+1}$ can be computed by substituting into expression (5.59) the function $\Psi_l(z)$ obtained according to (3.36) and also the correlation function $B(r, z; r', z')$, which is based on some specific model of the inhomogeneities. Analogous to Kohler and Papanikolau (1977), one can show that the dependence of the diffusion coefficient on p appears in the form of a correction $\approx p\pi/kH$. Bear in mind that because of absorption, we are interested in the region of values $p \ll kH/\pi$. Further, we are subject to the condition that the longitudinal scale of the correlation of the random inhomogeneities $R_{||}$ is small in comparison with the ray cycle distance D. Thus, this dependence can be neglected. Therefore, we shall assume in this section that the diffusion coefficient a does not depend on the parameter p in the theory.

Thus we write

$$\begin{aligned} \frac{\partial W_p(r)}{\partial r} &= -\frac{s\pi^2}{k^2 H^3} p^2 W_p(r) + a\frac{\partial^2 W_p(r)}{\partial p^2}, \\ W_p(0) &= W_0(p), \\ \frac{dW_p}{dp}\Big|_{p=0} &= 0. \end{aligned} \tag{5.66}$$

The "initial" conditions $W_0(p)$ are determined by the sound source. Since, due to the growth of the absorption coefficient γ_p with the index p, the distribution function decreases rapidly as $p \to \infty$, we may suppose that the variable p ranges from 0 to ∞. Because the coefficient γ_p is an even function of p, the function W_p can be evenly extended to the region of negative values of p so that when calculating the integrals over p, we must then take a factor of ½ in front of the integral.

Equation (5.66) has the form of a Schrödinger equation for a harmonic oscillator, whose solution can be given for arbitrary initial conditions $W_0(p)$:

$$W_p(r) = \frac{1}{\sqrt{4\pi a r_0 \sinh(r/r_0)}} \int_{-\infty}^{\infty} W_0(q) \exp\left\{\frac{1}{2ar_0}\left[-\frac{p^2+q^2}{2}\coth(r/r_0) + \frac{pq}{\sinh(r/r_0)}\right]\right\} dq \tag{5.67}$$

In particular, if the source is a nondirectional point source, then from (5.48) we obtain $W_0(p) = H^{-1}$ and

$$W_p(r) = \frac{1}{H\sqrt{\cosh(r/r_0)}} \exp\left[-\frac{p^2}{4ar_0}\tanh(r/r_0)\right], \tag{5.68}$$

where

$$r_0 = \frac{kH}{2\pi}\sqrt{\frac{H}{as}}. \tag{5.69}$$

Clearly, $W_p(r)$ is the product of an amplitude factor $\mathrm{ch}^{-1/2}(r/r_0)$ times a Gaussian distribution, whose width (i.e., the effective number of modes propagating) decreases as r increases, but is bounded below by a limiting width equal to $2\sqrt{ar_0}$. The characteristic spatial scale of these variations is r_0, the parameter which takes into account the role of the random inhomogeneities in the waveguide.

Let us next consider the limiting cases. If $r \ll r_0$, then $W_p(r) \approx H^{-1}$ $\exp(-\pi^2 sp^2 r/k^2H^3)$, which corresponds to attenuation of an individual mode in the Pekeris waveguide. The factor in front of r in the exponential is the mode attenuation coefficient (5.65). In this limit, the diffusion coefficient a in formula (5.67) vanishes, which means that at distances less than r_0, random homogeneities are not manifest. If $r \gg r_0$, then $W_p(r)$ takes the form $(r^{-3/2}\pi/H)$ $\exp(-r/2r_0 - p^2/4ar_0)$. The intensity transported along the waveguide decreases exponentially, with mode attenuation coefficients $(2r_0)^{-1}$, independent of p, and the width of the mode intensity distribution (which is still Gaussian) becomes a fixed constant. Since the width of the distribution determines the number of modes propagating in the waveguide, it is clear that there infinitely many modes at $r = 0$, while the number decreases due to mode attenuation at large indices, but not to one as in Brekhovskikh (1980), because of the mode transformations by random inhomogeneities. The dependence of the width of the distribution on the waveguide parameters, specifically its growth as H and a increase and its decrease as s grows, is evident.

Let us now compute the average intensity according to (5.48), where the summation over modes must be replaced by integration over the continuous variable p, (5.62). The normalization of the emitted power is chosen as in Chap. 4 for the derivation of (4.98), $W_0 = 4\pi$. Substituting (5.68) in (5.62), we obtain

$$\bar{I} = \frac{2\pi}{kH^2 r}\sqrt{\frac{\pi r_0 a}{\sinh(r/r_0)}}. \tag{5.70}$$

This equation gives the average decay law for the intensity along the waveguide and generalizes the 3/2 law (Brekhovskikh 1965). The characteristic parameter here is r_0. If $r \ll r_0$, then from (5.70) we obtain the $r^{-3/2}$ law, describing the decay in the

intensity of sound in an underwater sound channel without random inhomogeneities.

Let us now consider the behavior of the averaged coherent component of the sound field intensity, which is obtained from (5.64), where the distribution function for the coherent component U_p is computed from (5.63). The attenuation coefficient for the coherent component in our approximation based on formula (5.32) is $\Gamma_p = a_{p,p-1} + a_{p,p+1} = 2a$. Substituting this and the attenuation coefficient γ_p from (5.65), we obtain

$$\frac{\partial U_p(r)}{\partial r} = -\left(\frac{s\pi^2 p^2}{k^2 H^3} + 2a\right) U_p(r), \tag{5.71}$$

$$U_p(0) = W_p(0) = W_0(p) = H^{-1}. \tag{5.72}$$

Equation (5.71) with the initial conditions (5.72) is easily solved, giving

$$U_p(r) = \frac{1}{H} \exp\left[-\left(\frac{s\pi^2 p^2}{k^2 H^3} + 2a\right) r\right]. \tag{5.73}$$

From this, one can find the averaged intensity of the coherent component

$$\bar{I}_c = \sqrt{\frac{\pi}{sH}} r^{-3/2} \exp(-2ar). \tag{5.74}$$

Comparing (5.74) and (4.106), we see that the decay of the coherent component is determined by the "3/2 law" with an exponential factor determining an additional decrease due to scattering by random inhomogeneities. Figure 5.3 shows the curves for the decay of the given components of the intensity for a model sound channel, characterized by the parameters indicated in the figure caption. As can be seen from the figure, the behavior of the intensity of the wave field in a randomly inhomogeneous waveguide is associated with the operation of several mechanisms. These include absorption by the bottom, redistribution of the intensity between modes, and redistribution of the intensity between the coherent and the incoherent component. Near the source, where the effects of scattering by random inhomogeneities are small, the intensity of the sound field is concentrated mainly in the coherent component (in the language of modes, this means that the fluctuations are small, and a comparatively stable interference structure should be observed). As the observer moves away from the source, considerable losses of the total intensity and its coherent component occur, whereas the incoherent component grows with the inflow of energy from the coherent component (the fluctuations grow, both in the relative sense and in absolute value). At a certain distance from the source, the incoherent component (fluctuations of intensity) reaches a maximum and then decreases because its loss to the bottom is not compensated by an inflow of energy from the coherent component due to scattering. At large distances, all the

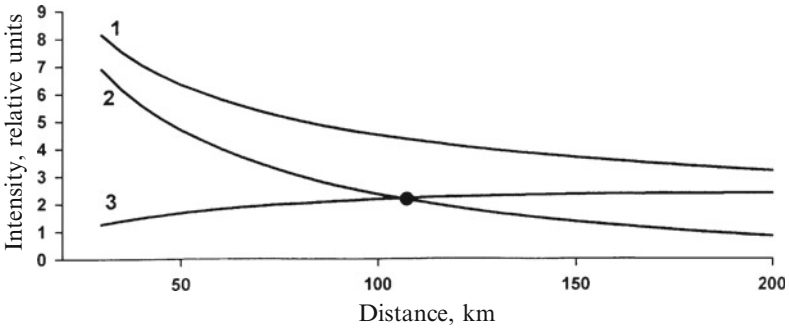


Fig. 5.3 Averaged decay laws of sound with frequency $f = 300\,\text{Hz}$ in a waveguide with random inhomogeneities. *Curve* (1) total intensity, (2) coherent component, (3) incoherent component. Waveguide parameters: $H = 150\,\text{m}$, $c = 1,500\,\text{m/s}$, dimensionless coefficient of absorption in the bottom $s = 0.1$, diffusion coefficient $a = 3 \times 10^{-6}\,\text{m}^{-1}$

components of the intensity decrease monotonically. Based on the above analysis, we can introduce a characteristic distance r_{cog} (the scale of coherence) at which the coherent component and the incoherent component are equal; for $r > r_{\text{cog}}$, the radiation is dominated by the incoherent component. In the language of modes, we can say that at distances less than the scale of coherence, we observe a comparatively stable interference structure, while at large distances fluctuations predominate and the field has a substantially stochastic nature. For our model case (a Pekeris model with random inhomogeneities), we obtain the estimate $r_{\text{cog}} \sim 0.35/a$, which gives $r_{\text{cog}} \sim 100\,\text{km}$ for the parameters of our waveguide.

5.6.2 *Irregular Waveguide*

Let us now consider the example of a very simple irregular waveguide with constant refractive index $(n = 1)$ and depth, which is a function of the coordinate $r : H = H(r)$. We shall assume that this dependence is "sufficiently smooth" (varies little over a ray cycle distance). This problem can be analyzed based on a decomposition of the amplitude of the sound field $\Psi(r, z)$ into transverse matched modes $\Psi_l(r, z)$, which depend on r as a parameter. Analogously, the quantities $\sigma_p, q_p, \Psi_p, D_p$, and γ_p also exhibit dependence on r, which can be formally obtained by substituting $H(r)$ in place of H. As noted in Sects. 3.2 and 3.4, this approach can be used for the case of a sufficiently smooth variation of the waveguide depth with distance, which is determined by the nonadiabaticity parameter $v = kHH'/2$. If we assume that $H' \sim 10^{-3}$, $H \approx 100\,\text{m}$, and $k \approx 2$, the parameter is small for several tens of modes propagating for those waveguide parameters, and the above approach is legitimate.

The dependence of the parameters upon r means that in accordance with the diffusion equation for the mode intensity $W_p(r)$, the absorption and diffusion

coefficients will depend on r. For γ_p, this dependence is obtained from (5.65); we assume that a does not have such a dependence. This assumption corresponds, for example, to a scattering layer of constant width in the waveguide, or to the fact that wave transformation is caused by boundary roughnesses. On the contrary, given the presence of a dependence $a(r)$, by using a simple replacement of variables, this dependence can be transferred to a coefficient in front of the first term on the right-hand side of (5.66). Thus,

$$\begin{aligned} \frac{\partial W_p(r)}{\partial r} &= -\frac{s\pi^2}{k^2H^3(r)}p^2W_p(r) + a\frac{\partial^2 W_p(r)}{\partial p^2}, \\ W_p(0) &= W_0(p), \\ \frac{dW_p}{dp}|_{p=0} &= 0. \end{aligned} \tag{5.75}$$

Since in a smoothly irregular waveguide the index of a matched mode (or in our case the parameter p) can be mapped to an adiabatic invariant, (5.75) is interpreted as a diffusion equation with an adiabatic invariant. An analogous approach has been used in problems concerning the propagation of radio waves in the ionosphere by Gurevich and Tsedilina (1979).

A solution of (5.75) for a nondirectional point source has the form

$$W_p(r) = \frac{1}{H_0}\exp\left[-2a\int_0^r u(r')dr' - u(r)p^2\right], \tag{5.76}$$

where $H_0 = H(0)$, and u is a function satisfying the Riccati equation

$$\frac{du}{dx} + qu^2 = \frac{1}{qf^3(x)}, \tag{5.77}$$

$$u(0) = 0, \quad x = \frac{r}{r_0}, \quad r_0 = \frac{kH_0}{2\pi}\sqrt{\frac{H_0}{as}}, \quad f(x) = \frac{H(r)}{H_0}, \quad q = 4ar_0.$$

It is clear that the distribution of the intensity by mode remains Gaussian with width $u^{-1/2}$ (as $r \to 0$, the width tends to infinity, which corresponds to all the excited modes). Let us now consider some subcases. If $H = \text{const}$, then the solution of (5.77) is the function $u = (4ar_0)^{-1}\text{th}(r/r_0)$, in agreement with (5.68). For $a = 0$, $u = \frac{s\pi^2}{k^2}\int_0^r H^{-3}(r')dr'$, and the exponent corresponds to the attenuation coefficient of an adiabatic mode, obtained in the semiclassical approximation. Equation (5.77) is easy to solve numerically. For a specific example, we assume that the waveguide depth varies linearly: $f(x) = 1 + \varepsilon x$, where ε is a dimensionless parameter, $\varepsilon > 0$ corresponds to a deepening channel, and $\varepsilon < 0$ indicates a shoaling channel. The results of numerical calculations for (5.77) are shown in Fig. 5.4. Let us first

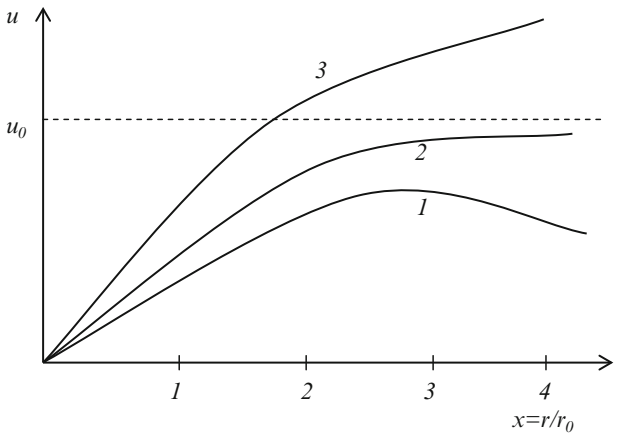


Fig. 5.4 Graphs of $u(x)$ for a waveguide with linearly varying depth (1) deepening waveguide $(\varepsilon = 0, 1)$; (2) regular $(\varepsilon = 0)$; (3) shoaling $(\varepsilon = -0, 1)$; for all curves $(4ar_0 = 1)$; *dashes* – value of $u_0 = (4ar_0)^{-1} = 1$, corresponding to the asymptotic value of u in a regular channel for $(H = H_0)$

consider a deepening channel. Curve 1 clearly behaves nonmonotonically; it grows at first, reaching a maximum at some value x, depending on q, and then falls. Recalling that $u^{-1/2}$ is the width of the Gaussian distribution W_p, the behavior of $u(x)$ is interpreted as follows. At the point $x = 0$, $u = 0$, that is, the width of the distribution is infinite, which corresponds to the fact that all modes (rays) are present. As propagation occurs, because of absorption in the bottom, modes with high indices (steep rays) are strongly absorbed, and since for $r<r_0$ no random inhomogeneities are manifest, the packet becomes narrower. For $r>r_0$, the diffusion mechanism takes effect, leading to the appearance of modes with large indices. In a channel of constant depth, the birth and attenuation of modes cancel each other out, and at large distances, the function $u(x)$ approximates asymptotically to a constant. In a deepening channel after some maximum value is reached, $u(x)$ decreases as $x \to \infty$; in other words, the distribution becomes broader and new modes appear. This happens because the ray grazing angle with the boundaries in a deepening channel $\chi(r) = (H_0/H(r))\chi(0)$ decreases and, consequently, its absorption by the boundary decreases in such a way that the appearance of new rays due to scattering by random inhomogeneities is not compensated for by absorption. In the case of a shoaling channel, the angle $\chi(r)$ grows as r increases and absorption at the boundary dominates. Here, the function grows sharply as r increases; in other words, the distribution narrows.

We note that in a deepening channel, there exists a distance x_m, at which the number of propagating modes is maximal. At that point $qu^2_{\max} = 1/qf^3(x_m)$, $u_{\max} = (1/q)f^{-3/2}(x_m)$. If the waveguide has a complicated depth profile, then by decomposing it into segments in which it is purely shoaling or deepening, one can gain a qualitative understanding of the behavior of the distribution of intensity. An example of such a waveguide is given in Fig. 5.5, where the behavior of the

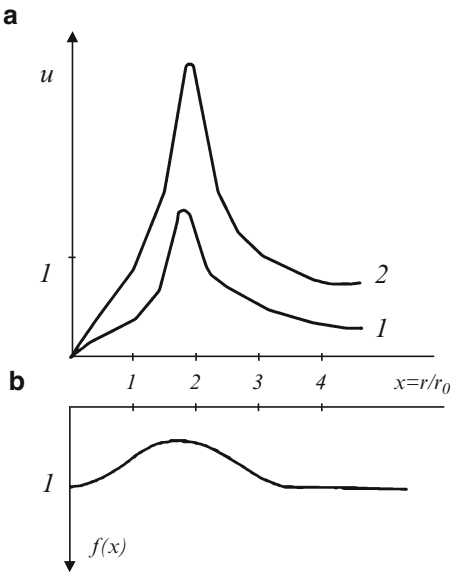


Fig. 5.5 Graphs of $u(x)$ for an irregular channel (**a**) and its profile (**b**). (1) $4ar_0 = 1$; (2) $4ar_0 = 0.5$

curves can be understood from the above discussions. The average intensity has the form

$$\bar{I} = \frac{1}{rH(r)}\sqrt{\frac{\pi}{u(r)}}\exp\left[-2a\int_0^r u(r')dr'\right]. \tag{5.78}$$

In the limiting case $a = 0$, Eq. (4.104) follows from (5.78). For $H = H_0$, (5.78) becomes (5.70).

It is clear from (5.78) that the behavior of the average energy density is determined by the function $u(x)$. The presence of random inhomogeneities leads to a faster exponential decay where, even if u decreases with distance (in a deepening channel), the integral in the exponential grows. An analysis of the coherent component of the intensity in this case can be performed analogously to the previous section.

5.6.3 *Dependence of the Intensity on Depth*

Let us now consider the reconstruction of the vertical (depth) dependence of the intensity of sound due to a point source in shallow water with random inhomogeneities. Suppose that the average part of the square of the refractive index in the water layer is described by a linear dependence on depth

$$n^2(z) = n_0^2(1 + 2\varepsilon z), \tag{5.79}$$

where n_s is the refractive index at the surface. We know that for the case of a negative sound-speed gradient, unlike in the examples just considered, the absorption coefficient depends on p in a different way from (5.65), which has to do with the presence of groups of near-bottom modes (rays) with approximately the same attenuation coefficient. A characteristic dependence of γ_p was shown in Fig. 4.3. Equation (5.58) can only be solved numerically, based on a representation of the distribution $W_p(r)$ as a sum over the eigenfunctions of the Sturm–Liouville problem. We obtain

$$W_p(r) = \sum_n P_n(p) \exp(-\lambda_n r), \tag{5.80}$$

where $P_n(p)$ and λ_p are determined by the equation

$$\frac{d}{dp}\left(a_p \frac{dP_n}{dp}\right) + (\lambda_n - \gamma_p)P_p = 0 \tag{5.81}$$

with boundary conditions $P_n'(0) = P_n(p_{\max}) = 0$. It is clear that as r increases, only one term remains in the sum (5.80), namely, that corresponding to the least value of the parameter λ. We refer to this as the asymptotic solution [in Artel'nyi and Raevskii (1987a, b), the analogous limit is called the self-similar limit; see also Whitman and Beran (1978)]. Accordingly, the distribution of the field intensity with depth tends to some "asymptotic limit":

$$\bar{I}^*(r, z) = \frac{\exp(-\lambda^* r)}{2\pi\rho c r} \int_0^{p_{\max}} dp \frac{P^*(p)}{\alpha_p(z) D_p}, \tag{5.82}$$

where λ^* is the minimal eigenvalue and $P^*(p)$ is the corresponding eigenfunction.

To obtain further estimates, we shall assume that the diffusion coefficient is independent of the mode index. To estimate the extent of the self-similar regime, one can use the following relation: $\min(\lambda_n - \lambda^*)r > 1$ for $n > 1$; in particular, for the Pekeris waveguide,

$$r > R_c = \frac{kH}{4\pi}\sqrt{\frac{H}{a_p s}}. \tag{5.83}$$

For a waveguide with a negative sound-speed gradient, the estimates will be rather different. We shall suppose that the effects of scattering are so small that in the self-similar regime only "bottom" modes remain, i.e., $4a/(s\varepsilon^2 H^3 k^2) \ll 1$. This condition is well satisfied for typical parameters $f = 300\,\text{Hz}$, $H \sim 100\,\text{m}$, $s \sim 0.1$,

$a_p \sim 10^{-5}\,\mathrm{m}^{-1}$, $\varepsilon \sim 1.2 \times 10^{-4}\,\mathrm{m}^{-1}$, and $c_S \sim 1,500\,\mathrm{m/s}$. In this case, we obtain the simple expressions for the estimates of the distances of the self-similar regime and of the distribution of intensity:

$$r > R_c = \frac{4}{3}\frac{\varepsilon k^2 H^3}{\pi^4 a_p}, \quad 0 < z < H - \left(\frac{3\pi}{2kn_s}\right)^{2/3} (2\varepsilon)^{-1/3}$$

$$\overline{I^*}(r,z) = \frac{I_0 \exp\left(-\lambda^* r\right)}{2\pi\rho c r} \frac{Hn(H)\sqrt{2\varepsilon}}{2\pi\sqrt{H-z}} \arccos\left(\sqrt{\frac{H-z}{H}}\right), \tag{5.84}$$

where I_0 is some normalization value for the intensity.

As one might expect, the speed of the decay in intensity is determined by the gradient of the refractive index, and the longitudinal scale of coherence grows as the gradient of the sound speed increases. This has to do with the fact that for large values of ε, the number of mixed modes decreases. We note that because of the need to satisfy the above inequality, the transition from (5.84) to (5.78) in the limit does not take place. For typical shallow water conditions of a drop in the sound speed between the bottom and the surface of $\geq 10\,\mathrm{m/s}$, we have $R_c \geq 70\,\mathrm{km}$. We also note that the sense of the limiting regime is that at large distances, an equilibrium distribution of $W_p(r)$ is obtained over the modes: the "dying out" of the higher modes as a result of greater absorption than for lower modes is compensated for by a transfer of energy to them as a result of scattering by inhomogeneities. As a result, the shape of the curve of the depth dependence is unchanged with distance. Figure 5.6 shows numerical calculations of the dependence of the sound field intensity on depth at different distances from the source. In these calculations, both waveguide boundaries were assumed to be pressure-release boundaries. The calculation was carried out for the parameter set $f = 300\,\mathrm{Hz}$, $H = 100\,\mathrm{m}$, $s = 0.1$, $a_p = 10^{-5}\,\mathrm{m}^{-1}$, and $\varepsilon = 1.2 \times 10^{-4}\,\mathrm{m}^{-1}$, $c_S = 1,500\,\mathrm{m/s}$. The nature of the dependence of the intensity on depth can be understood to be the result of scattering by inhomogeneities of the energy concentrated in the modes with the least attenuation, namely, the near-bottom modes in this case. Because of this, the graph of the field intensity becomes smoother and, as a result of the redistribution of energy over the modes, most of the energy is concentrated at depths with a lower sound speed. A more important detail is the broadening of the interference structure. It is clear that at comparatively small distances, the vertical interference structure is little disrupted, whereas at large distances (for our values, starting from ~100 km), it is practically blurred. In Fig. 5.6, curve 1, for a distance of 200 km, already corresponds to the limiting distribution (for the chosen values of the waveguide parameters, the longitudinal scale of coherence is ~170 km). We note that unlike at small distances from the source, where there is a stable interference structure, at the distances of the limiting regime, this constant interference field no longer exists. Thus, in "sounding" (or recovering the waveguide parameters) at such distances, one needs to use the statistical characteristics of the field.

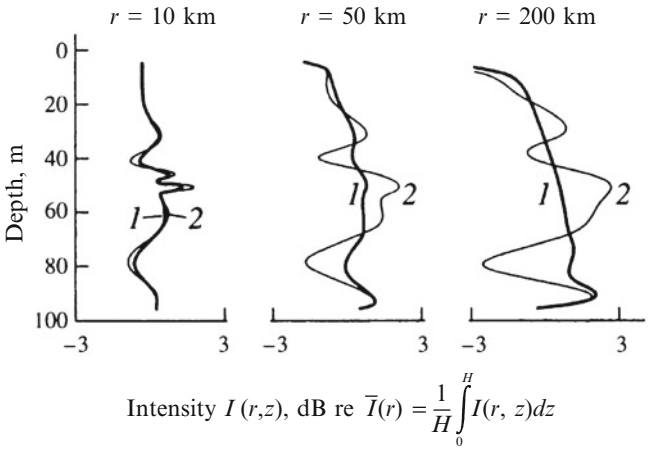


Fig. 5.6 Variation of the depth dependence of the intensity of sound at frequency $f = 300$ Hz in a waveguide with linear sound speed profile $c(z) = c_S(1 - \varepsilon z)$, $c_s = 1,500$ m/s, $\varepsilon = 2.5 \times 10^{-4}\,\mathrm{m}^{-1}$, $H = 100$ m, depth of source 50 m, diffusion coefficient $a = 10^{-5}\,\mathrm{m}^{-1}$, dimensionless absorption coefficient in bottom $s = 0.1$. Distances from the source are shown in the figure. Curves 1 and 2 for each distance describe intensity with and without random inhomogeneities, respectively

5.7 Sound Field Fluctuations in the Presence of Background Internal Waves

As mentioned above, a random field of background linear internal waves is universally present in shallow water. In Chap. 2, we described the main features of this field as well as a numerical algorithm for computer simulation of the random vertical displacements $\zeta(r, \vartheta, z, T)$ caused by IWs. Here, r, ϑ, z are cylindrical coordinates; we use the value of the geotime T to denote the slow temporal dependence of $\zeta(r, \vartheta, z, T)$ compared to the typical time scale of the sound field variation $(2\pi/\omega)$. The random displacements in turn produce sound-speed perturbations in the acoustical waveguide and thus affect -range sound propagation. The methods of sound field calculations for a randomly inhomogeneous waveguide, which take into account mode coupling, were just described. Here we would like to demonstrate and discuss some peculiarities of sound propagation connected with the anisotropic background internal waves as another example of random medium propagation. Specifically, we would like to demonstrate some of the peculiarities of the amplitude fluctuations of the sound field and the fluctuations of the interference invariant of the sound field. More precisely, we will take the phrases "amplitude fluctuations" and "fluctuations of the interference invariant" to mean the temporal variation of $|P(r, z, \vartheta, T)|_{r,z=\mathrm{const}}$ (or what is the same – the temporal amplitude variation of a CW signal at the reception point) and the temporal variation of $\beta_0(T)$, respectively. We assume that these variations take place along a stationary acoustic track with an unmoving sound source in the presence of background IWs. It should be noted that in this case, the coefficients C_m involved in (3.41) depend on the distance r, the depth z, and the geotime T.

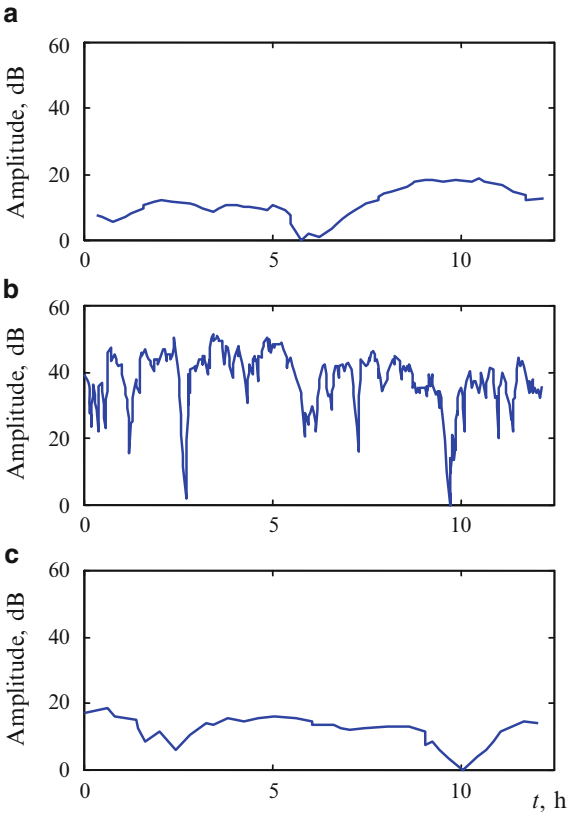


Fig. 5.7 Fluctuations of sound field amplitude, recorded on stationary acoustic tracks. (**a**) North Sea, $f = 295$ Hz, $r = 1.1$ km (Essen et al. 1987); (**b**) Barents Sea $f = 300$ Hz, $r = 70$ km, (Katsnelson and Petnikov 2002) (**c**) Sea of Japan, $f = 315$ Hz, $r = 16$ km, (Rutenko 1997)

In beginning this section, it is important to note the following. At the present time, fluctuations of the sound field caused by internal waves have been recorded in experiments scores of times. As an example, see Fig. 5.7, where we show amplitude fluctuations,[4] which were recorded at three different regions of shallow water. The vertical sound-speed profiles corresponding to these three regions are shown in Fig. 4.13. (For this example, we chose 12-h fragments of recordings of the

[4] In measurements, i.e., in work with real CW signals, the amplitudes are determined somewhat differently: $|P(r,z,t)| = \sqrt{p^2_{\cos}(r,z,t) + p^2_{\sin}(r,z,t)}$ where $p_{\sin}(r,z,t) = \overline{p(r,z,t) * \sin \omega t}$ and $p_{\cos}(r,z,t) = \overline{p(r,z,t) * \cos \omega t}$ are the sine and cosine quadrature components of the real (measured) field $p(r,z,t)$, related to the field $P(r,z,t)$ by the equation $p(r,z,t) = \mathrm{Re}(P(r,z,t))$. The overline denotes averaging over the period. Simple transformations show that these two definitions are equivalent.

amplitudes of pure-tone signals. We can see from Fig. 5.7 that the fluctuations measured in different experiments are very different in both amplitude and frequency. This is due to the geographic and seasonal variability of the near-bottom sound channel, including the variability of the internal wave field itself, which is different in different regions of shallow water and for different seasons. However, the results of these experiments to date do not enable us to determine the influence of purely linear internal waves on the fluctuations of the sound field, including the influence of the anisotropy of these waves. The reason for this is that in most shallow water experiments, the separation of the fluctuations caused by linear background internal waves and the fluctuations caused by (also common) trains of solitons was not carried out. Also, the directivity function of the internal waves was generally not measured, and so the angle between the direction of the sound and the internal wave propagation was simply unknown. Finally, experiments where the propagation of the same sound signal was studied on two mutually perpendicular stationary acoustic tracks are not common. Some of these experimental shortcomings *were* addressed in the recent, large-scale Shallow Water 2006 (SW06) experiment off New Jersey – we will discuss those new results in Chap. 11.

Given a lack of detailed experimental knowledge, we can do the next best thing and demonstrate some of the characteristics of sound propagation in a random anisotropic background internal wave field by using numerical experiments. For our calculations, we will use the temporal power spectrum $G^{\mathrm{t}}(\Omega)$ shown in Fig. 5.10. We will simulate the random vertical displacements $\zeta(r,\vartheta,z,T)$ by using the algorithm described previously (see Chap. 2) and then calculate the corresponding amplitude variations of the sound field by applying (3.106, 3.114). For the calculation $C_m(r,\vartheta,T)$, we will use (3.120) in the following form:

$$\frac{dC_m(r,\vartheta,T)}{dr} = -\frac{\gamma_m}{2}C_m(r,\vartheta,T) + i\sum_{l=1}^{M}\tilde{V}_{lm}(r,\vartheta,T)C_l(r,\vartheta,T)\exp[i(q_l - q_m)r], \tag{5.85}$$

where according to (3.122) and (2.31)

$$\tilde{V}_{lm} = \frac{-Qk^2}{\sqrt{q_l q_m}}\int_0^H \Psi_l(z)\tilde{N}(z)\zeta(r,\vartheta,T)\Psi_m(z)dz; \tag{5.86}$$

i.e., we assume that only volume inhomogeneties connected with the background internal waves are present and that $\eta \cong 1$, $\tilde{N}(z)$ is Vaisala frequency.

For the calculations to follow, we use the following parameters: acoustic track length (distance between source and receiver) $r = 10\,\mathrm{km}$; depth $H = 72\,\mathrm{m}$; bottom parameters of density $\rho_1 = 1.8\,\mathrm{g/cm^3}$ and bottom refractive index $n_1 = 0.84(1 + 0.007i)$; frequencies of $f = 100$ and $240\,\mathrm{Hz}$; source power $W_0 = 1\,\mathrm{kW}$; source depth $z_0 = 50\,\mathrm{m}$; and receiver depth $z = 40\,\mathrm{m}$. The sound-speed profile we will use is plotted in Fig. 4.11. (This profile corresponds to the water stratification observed during JUSREX (Gasparovic and Etkin 1994a, b). Recall that this

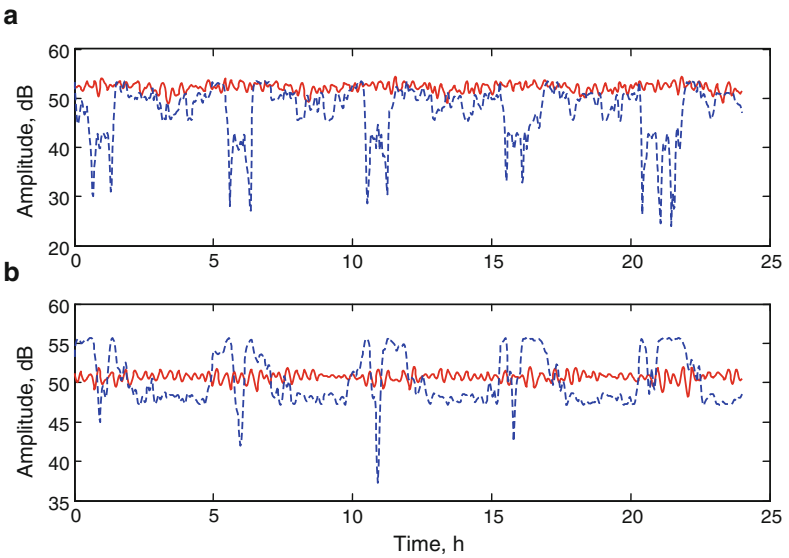


Fig. 5.8 Fluctuations of sound field amplitude. IWs cross the stationary acoustic track at right angles (for *blue lines*) and propagate in the same direction as the sound waves (for *red lines*). Frequencies are equal to 240 Hz (**a**) and 100 Hz (**b**)

experiment was carried out in Mid-Atlantic Bight.) The directivity function of the internal wave field is set equal to $\theta(\vartheta) = (\cos\vartheta)^{400}$. (Notice that this directivity function corresponds to the sharpest function measured by processing the radar images shown in Fig. 2.17).

Let us now carry out the numerical experiments for the two most important directions of internal wave propagation, along and across a stationary acoustic track.

The results of numerical modeling are shown in Fig. 5.8. Figure 5.8 displays the fluctuations of the sound field amplitude for two different angles $\tilde{\vartheta}$ between the directions of the sound and internal wave propagation. For this numerical example, a common random realization of the background internal wave field was used for both the longitudinal $\left(\tilde{\vartheta} = 0^\circ\right)$ and transverse $\left(\tilde{\vartheta} = 90^\circ\right)$ directions of internal wave propagation. The figure strongly suggests that when the internal waves cross the stationary acoustic track at a right angle, the amplitude fluctuations exceed those for the case where the sound waves and internal waves propagate in the same direction. A similar effect was described in Katsnelson and Petnikov (2002), where a simpler model was used for the simulation of the vertical displacements due to the background linear IW field.

In order to study this effect in more detail, let us estimate the power spectrum of the amplitude fluctuations

$$G_{\mathrm{a}}(\omega) = \left\langle \frac{2}{T_d} \left| \int_{-T_d/2}^{T_d/2} \Delta P(T) \exp(i\omega T) dT \right|^2 \right\rangle. \tag{5.87}$$

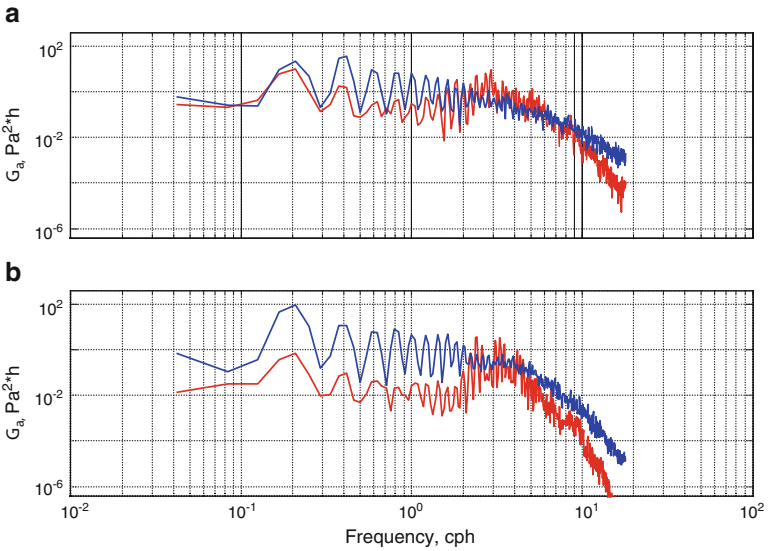


Fig. 5.9 The power spectra of amplitude fluctuations of the sound field. IWs cross the stationary acoustic track at right angle (*blue lines*) and propagate in the same direction as sound waves (*red lines*). Frequencies are equal to 240 Hz (**a**) and 100 Hz (**b**)

Here, $\Delta P(T) = |P(T)| - \overline{|P(T)|}$, the overbar denotes time averaging for a time T_d, and T_d is the duration of a realization (i.e., the duration of an emitted pulsed CW signal). The result of these calculations is shown in Fig. 5.9, where T_d and the number of independent random realization are equal to 24 h and 10 correspondingly. In other words, to calculate the spectrum shown in Fig. 5.9, we first calculated ten independent random realizations of the IW field with duration 24 h each, after which we calculated ten realizations of sound field $|P(T)|$. Finally, we estimate the power spectrum $G_a(\omega)$ by using (5.87). The spectrum estimation confirms the conclusion that the sound field fluctuations depend on the direction of IW propagation; however, this is true only at frequency band $\Delta\Omega \cong 0.1$ to 1 cph. When IWs cross the stationary acoustic track at right angles, the low-frequency fluctuations of the value $\Delta P(T)$ exceed the similar fluctuations for the case where the sound waves and internal waves propagate in the same direction. Presumably, the reason for this is that for transverse propagation, the spatial correlation length of the vertical displacements matches or exceeds the length of the stationary acoustic track for the frequency band $\Delta\Omega \cong 0.1 - 1$ cph. We should keep in mind the scales of the correlation radius. For the lowest frequencies, $\Delta\Omega \leq 0.1$ cph, the correlation radius in both the transverse and longitudinal directions matches or exceeds the length of the stationary acoustic track. (The size of the internal wave packet is comparable to the track length.) Correspondingly, for the lowest internal wave frequencies, the effects of internal wave anisotropy on acoustics are not so strong.

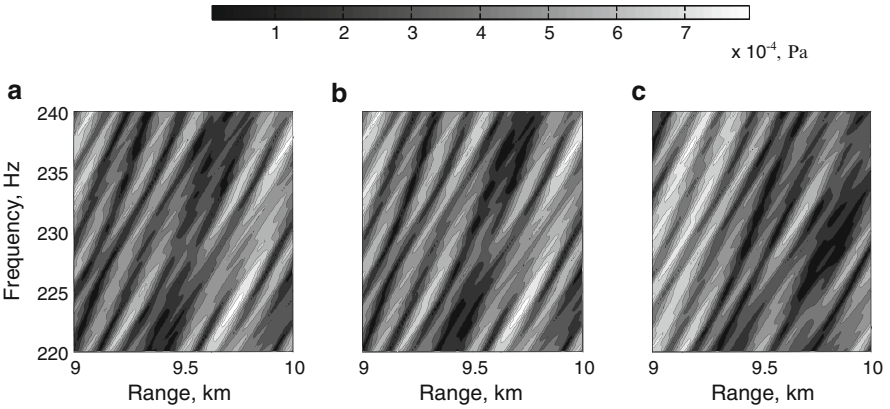


Fig. 5.10 Brightness values of the sound field amplitude without IWs (**a**), with IWs propagating in the same direction as the acoustic waves (**b**), and with IWs propagating at right angles to the propagation direction of acoustic waves (**c**)

Next, let us examine the behavior of sound field interference invariant β_0 in the presence of an anisotropic field of background internal waves.[5] For the frequency and range bands equal to $\Delta f = 20\,\text{Hz}$ ($f_0 = 230\,\text{Hz}$) and $\Delta r = 1\,\text{km}$ ($r_0 = 9.5\,\text{km}$), calculated interference patterns of the sound field are shown in Fig. 5.9 for a common random realization of the IW field at a given time, but the different directions of sound propagation. f_0 and r_0 are the means of the frequency and range, respectively. (For these calculations, the waveguide parameters and statistical characteristics of the IW field mentioned above were used.) As seen from the figure, IWs cause detectable variations of the interference pattern and consequently a variation of β_0. To demonstrate this variation in more detail, we calculated the sequence of functions $\tilde{\Theta}_n(\beta) = \tilde{\Theta}(\beta, T_n) + 2n$, i.e., we calculated the function $\tilde{\Theta}(\beta)$ for random realizations of the IW field at different times. Here, T_n is equal to $T_n = 0, 2, 4, \ldots, 24\,\text{h}$. The results of the calculations are shown in Fig. 5.10. Remembering that the maximum of $\tilde{\Theta}(\beta)$ occurs where $\beta = \beta_0$, we can clearly see the temporal variations $\beta_0(T)$ in the figure. Slightly greater variations are seen when the IWs propagate across the acoustic track (see Fig. 5.9) than in the case of longitudinal propagation. For the 90% confidence interval in the Student test, the averaged values $\overline{\beta_0}$ and $\overline{\Delta\beta}$ are equal to $1.415 \le \overline{\beta_0} \le 1.517$, $0.716 \le \overline{\Delta\beta} \le 0.930$ and $1.495 \le \overline{\beta_0} \le 1.596$, $0.755 \le \overline{\Delta\beta} \le 0.851$ for along and across IW propagation, respectively. ($\Delta\beta$ is the width of the function $\tilde{\Theta}(\beta)$ at the ½ level of the peak.) It is interesting to note that for the same distance $\Delta r = 1\,\text{km}$ ($r_0 = 9.5\,\text{km}$), Rouseff (2001) obtained about the same variations of β_0. In so doing, he used a somewhat

[5] It should be noted that the β_0-behavior is just being studied experimentally at the present day. The β_0-variations caused by background IW were first calculated by Rouseff and Spindel (Rouseff 2001; Rouseff and Spindel 2002).

different isotropic low-mode model of IW field, with a modified Garrett–Munk spectrum. In particular, j^* was equal to zero and $\Delta f = 20$ Hz ($f_0 = 410$ Hz).

Notice that the value $\Delta\beta$ determines the fuzziness of the crest direction, corresponding to a fixed level of the sound field. In the first approximation, $\Delta\beta$ is equal to

$$\Delta\beta = \sqrt{\left.\frac{\tilde{\Theta}(\beta)}{d^2\tilde{\Theta}(\beta)/d\beta^2}\right|_{\beta=\beta_0}}. \tag{5.88}$$

It is obvious that as long as the fuzziness is not too great, the interference pattern holds its shape when the distance and frequency are varied simultaneously. As the extreme value decreases, the width $\Delta\beta$ increases. By this is meant that the size of the space–frequency interference pattern, where the lines of fixed sound level hold their direction, decreases. In the limit where $\tilde{\Theta}(\beta_0) \to 1$, the width $\Delta\beta$ tends to infinity. In this case, the lines collapse to points and the interference pattern becomes unstable. This means that the value $\Delta\beta$ may be considered as a measure of the stability of the interference pattern under changes of distance and frequency. For the same numerical example without internal waves, which was discussed above in Chap. 4, $\Delta\beta$ is equal to $\Delta\beta = 0.833$.

The statistical estimations of $\overline{\beta_0}$ and $\overline{\Delta\beta}$ allow one to conclude that the background internal waves do not affect the stability of the interference pattern of the sound field, since with respect to the statistical error, the width $\Delta\beta$ does not differ from its value without internal waves. At the same time, for across wave propagation (along the internal wave wave crest), internal waves can cause detectable variations of the interference invariant β_0 in spite of the small amplitudes typical of linear background IWs. This is presumably because the long-wave length IWs cause synchronous thermocline oscillations along the acoustic track during propagation. These oscillations have significant influence on the dispersion $c_l^{\mathrm{gr}}(\omega)$, and the extent to which this is effective depends on the thermocline displacement (upward or downward from the equilibrium position).

At the same time, for the distance $r_0 \cong 10$ km, mode coupling has no time to change the mode composition significantly and so because of this, the interference invariant is scarcely affected. However, the situation changes when the distance increases. For the same parameters for the numerical experiment, $\overline{\beta_0(r_0)}$ and $\overline{\Delta\beta(r_0)}$ are shown in Fig. 5.11 without internal waves and in the presence of internal waves. As is evident from the figure, the interference invariant $\overline{\beta_0}$ and the width $\overline{\Delta\beta}$ rise with distance increase. This is true for the waveguide both without IW and with IW. The reason is that for small distances, the sound field is formed dominantly by the modes with high mode numbers for which $\beta_0 \approx 1.5$. With increasing distance, the mode number is reduced due to high mode attenuation (stripping). The field is then largely formed by modes with low numbers, which leads to a β_0 increase. At the distance $r_0 \approx 100$ km, the value β_0 converges to $\beta_0 = 3$, which is characteristic of a sound channel with a linear sound-speed profile (Chuprov 1982). As to the $\Delta\beta$ value, its increase is also caused by the decrease in the number of interfering modes forming the sound field at large distances.

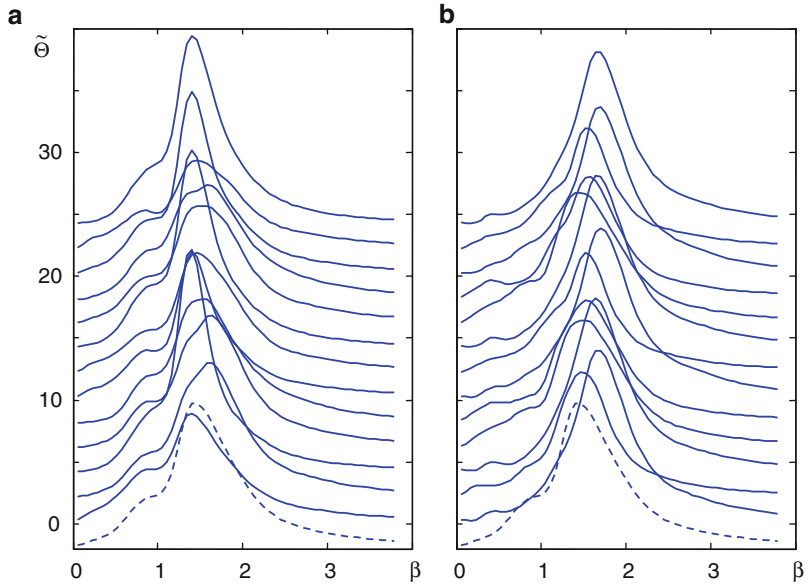


Fig. 5.11 Function $\tilde{\Theta}(\beta)$ at different times. The function was computed at 2 h intervals for IWs propagating in the same direction as the acoustic waves (**a**), for IWs propagating at right angles to the propagation direction of acoustic waves (**b**). $\tilde{\Theta}(\beta)$ without internal waves is plotted as a *dashed line*

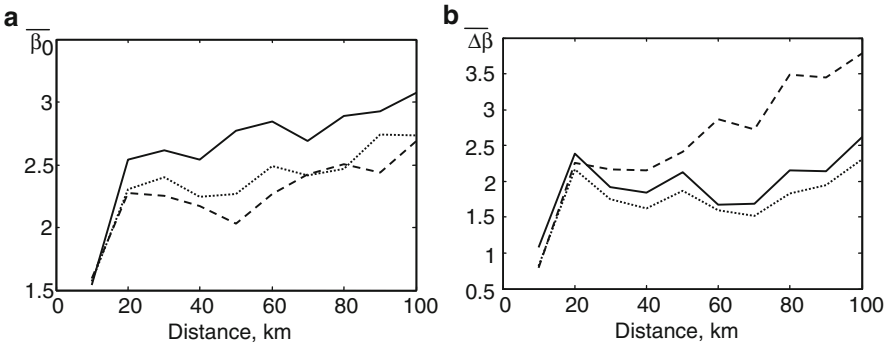


Fig. 5.12 Dependences of $\overline{\beta_0}(r_0)$ (**a**) and $\overline{\Delta\beta}(r_0)$ (**b**) without IW (solid lines), for transverse IW propagation (*dotted lines*) and for longitudinal IW propagation (*dashed lines*)

At the same time, Fig. 5.12 demonstrates the profound effect of internal waves on the interference invariant at large distances. This effect is connected with the modification of the modal structure of the sound field with distance. Due to internal waves, the modification takes place not only through high mode attenuation, but also because of strong mode coupling. As this takes place, an outflow of energy

occurs from the low modes with small attenuation to the high modes with large attenuation. Asymmetry of the energy flow enriches the mode spectrum for the high numbers and depletes it for the low numbers. As a result, with internal waves included the mode composition differs from that of the regular waveguide. This distinction is best in evidence for longitudinal internal wave propagation, where, as was mentioned above, mode coupling is greater.

5.8 Models and Statistics of Intensity Fluctuations

It is well known that for a known signal in Gaussian noise, the optimal detector is a correlator processor (or matched filter), whereas for an unknown signal in Gaussian noise, the optimal detector is a square law detector (Urick 1975). Since we are often searching for unknown signals and since $I = p^2/\rho c$ is the acoustic square law quantity of interest, the intensity is a very important entity to understand. Moreover, we often are interested in transient signals, where we integrate the instantaneous intensity over the duration over the signal, T, so that we are looking at variants of the instantaneous intensity such as the energy flux density, given by (A.177)

$$F = \int_0^T I dt = \frac{1}{\rho c} \int_0^T p^2 dt. \tag{5.89}$$

Additionally, we might be interested in peak intensity, average intensity, or intensity integrated over a vertical or horizontal array – the useful and interesting variants of intensity are plentiful.

Intensity fluctuations are generally quantified in two distinct ways. Specifically, in addition to looking at the variation of the intensity in our standard logarithmic (dB) units, another commonly used metric for intensity fluctuations is the *scintillation index, SI*:

$$SI^2 = \frac{\langle I^2 \rangle}{\langle I \rangle^2} - 1. \tag{5.90}$$

In looking at the intensity fluctuations, there are two limiting regimes. First, there is the weak scattering regime, where both the dB and SI metrics of intensity fluctuation approach zero. And second, there is the strongly scattered, saturated scattering regime (full random phases), where the dB variation approaches 5.6 dB (see below) and the *SI* approaches one. How one transitions from the weak to the saturated regime is a research topic of current interest, as we will discuss.

5.8.1 Intensity Fluctuations: Stochastic Modeling

One of the simplest things that can be done in the absence of very detailed knowledge of the medium and its scattering physics is to employ a stochastic approach. Specifically, one makes a few (hopefully reasonable) assumptions about the statistical nature of the medium and the scattering processes, and then goes on to model/predict the field fluctuations and then compare these to data. This approach has both power and pitfalls.

A classic statistical model treatment of what the intensity fluctuations should be in a strongly interfering multipath environment, such as shallow water, was put forth by Dyer (1970). In his paper, he showed that for a large number of equal intensity, uncorrelated, randomly interfering multipaths, the variance of the intensity (in log or dB units) should be 5.6 dB (or equivalently a scintillation index of one). This is a very simple model, yet a physically appealing one. For a sufficiently large number of (energetically significant) acoustic normal modes in shallow water (i.e., high enough frequencies), which are scattered independently (not exactly true, but OK for many cases) and have random phases (likely a fair assumption in the saturated, strongly scattered regime), one should approach this simple limiting number. This work, which also had an implicit assumption that the time-bandwidth product equals one, was later generalized by Makris (1996) for cases with a large time bandwidth product. In this case, the standard deviation of the transmission loss goes as $4.34\sqrt{(1/\mu)}$, where μ is the time-bandwidth product. A further extension of this work treating broadband signals and the approach to saturation was recently provided by Colosi and Baggeroer (2004), as will be discussed.

A simple demonstration of "Dyer's 5.6 dB number" can be produced by looking at the intensity distribution of a multimode interference pattern over the length of the longest mode cycle distance. This pattern samples the phase uniformly from 0 to 2π, mimicking a truly random phase distribution, and given enough energetic modes (not necessarily equal amplitude – but that is a weak constraint), this works rather well.

In keeping with the theme of random medium scattering and modes, we next turn to the work of Dozier and Tappert (1978). In their work, using coupled mode power equations (the so-called *master* equations), they showed that in a lossless medium, scattering due to internal waves would eventually (as a function of frequency, scattering strength and range) redistribute the modes so that each mode had equal energy and random phase. Thus, even though a source might have excited only a few modes preferentially, as the sound propagated in range it would be progressively scattered into the other modes until, at a certain range, energy equipartition would be achieved. This eventual equipartition and population of the full modal spectrum are seen to set up Dyer's conditions for a 5-dB intensity variance; Dozier and Tappert's work in fact would give the saturation range at which one would achieve Dyer's conditions, given the added assumption that the scattering strength of the internal wave field is constant with range.

However, one assumption in Dozier and Tappert's work is likely *not* to hold well in shallow water – the assumption of no appreciable modal attenuation over the

propagation path. While this assumption might be defended approximately for deep-water work, the high bottom attenuation seen in shallow water makes this assumption poor for the coastal ocean. To correct this, Creamer (1996) considered the modal scattering case including attenuation; this case is also considered in our Sect. 5.3. In this case, mode stripping competes with mode repopulation by scattering. This leads to a very different range evolution of the field fluctuation statistics with range, not surprising in retrospect. What *was* surprising was that Creamer's work indicated that one should see a very high scintillation index at large ranges – in fact, an exponentially increasing scintillation index. Mathematically, this results from a mismatch in the evolution of the second moment and the fourth moment of the acoustic pressure, with the second moment decaying more rapidly than the fourth. Experimentally, however, the transition to this exponential regime past SI = 1 comes at ranges of ~50–100 km or more, longer than one typically achieves in practice. Thus, one sees SI = 1 as a practical limit to the fluctuation strength.

One effect that has *not* been considered in stochastic calculations, but is probably of some importance to the amplitude/intensity fluctuation statistics one sees, is the range, azimuth, and time dependence of the field scattering the sound, e.g., internal waves. Especially in the case of coastal solitons, there is a very time- and space-dependent scattering field, and this needs to be considered. As nonstationary scattering is hard to accommodate in these purely statistical theories, this might be considered the point of diminishing returns where one abandons the purely statistical approach in favor of more deterministic models, or at least ensembles of deterministic calculations.

We would also note that the statistical theories which rely on mode coupling are looking at across-internal-wave front acoustic propagation paths. The along-wave front paths do not couple modes, and so need a different description, as we have discussed.

Furthermore, we note that we have not talked about deterministic descriptions of scattering and intensity fluctuations in this section. There is copious discussion of that topic elsewhere in the book so that this section just concentrates on random, statistical descriptions.

5.8.2 *Experimental Measurements and the Statistics of Intensity Fluctuations*

Dyer's prediction for the variance of the intensity in the saturated regime has been shown to hold in the context of shallow water acoustics by, among others, Pasewark et al. (2002) using the across-shelf path SWARM '95 data. These data show a variance of about 5.8 dB much of the time, within experimental error of Dyer's result. This is likely due to the strong acoustic mode coupling by the internal wave soliton field present keeping the waveguide populated with the maximum number of trapped modes with random phases; were the scattering weaker, both the number of modes would decrease (stripping effects would depopulate the trapped modal spectrum) and the phases would be less random (a standard weak scattering effect).

Dyer's main results get us through the first two moments of the probability density function (pdf), but what about the rest of the moments, or better the entire probability distribution function itself? There are many applications where the tails of the pdf are as important as the mean and variance, and so we should devote some effort to looking here.

From basic theory, it is well known that for saturated scattering, the complex acoustic field can be interpreted as a large number of random, independent contributions. The central limit theorem says that the real and imaginary components will approach zero mean Gaussian, random variables. The phases of the components will be uniformly distributed between zero and 2π, and the combined amplitude, a, will show a Rayleigh distribution, given by

$$\hat{w}(a) = \frac{a}{\sigma^2} \exp\left[\frac{a^2}{2\sigma^2}\right] \quad \text{for } a \geq 0 \tag{5.91}$$

where σ^2 is the variance. This is a simple one parameter distribution, well known to ocean acousticians.

Turning to the intensity, for saturated scattering, it is given by a two degree of freedom chi-square distribution (or a negative exponential), given by

$$\hat{w}(I) = \frac{1}{\langle I \rangle} \exp\left[-\frac{I}{\langle I \rangle}\right]. \tag{5.92}$$

If one considers the weak scattering regime, as opposed to the saturated scattering, we have been discussing so far, theory indicates that we should expect a log-normal distribution of the instantaneous intensity, given by

$$\hat{w}(I) = \frac{1}{I\sigma_{\log I}\sqrt{2\pi}} \exp\left[-\frac{[\log(I/m)]^2}{2\sigma^2_{\log I}}\right], \tag{5.93}$$

where m is the median and $\sigma_{\log I}$ is the standard deviation of the log-intensity.

An experiment that showed both the weak and strong scattering regimes of intensity fluctuations was the 1996 summer PRIMER experiment, conducted on the New England shelf break (Fredericks et al. 2005). In this experiment, 400-Hz signals were sent over 42.2 km and 59.6 km paths through an environment that featured both strong nonlinear internal waves and also intense low-frequency oceanographic variability due to the shelf break front and local eddy field. The internal wave field was a "high-frequency oceanography" signal, as opposed to the front and eddies, so it was possible by using high-pass filtering (discerning fluctuations faster and slower than 2 h period) to separate the acoustic scattering signatures of both these components of the oceanography.

In Fig. 5.13, we show an example of the separation of the total (top panel) fluctuation signal from the longer path into the high- and low-frequency components

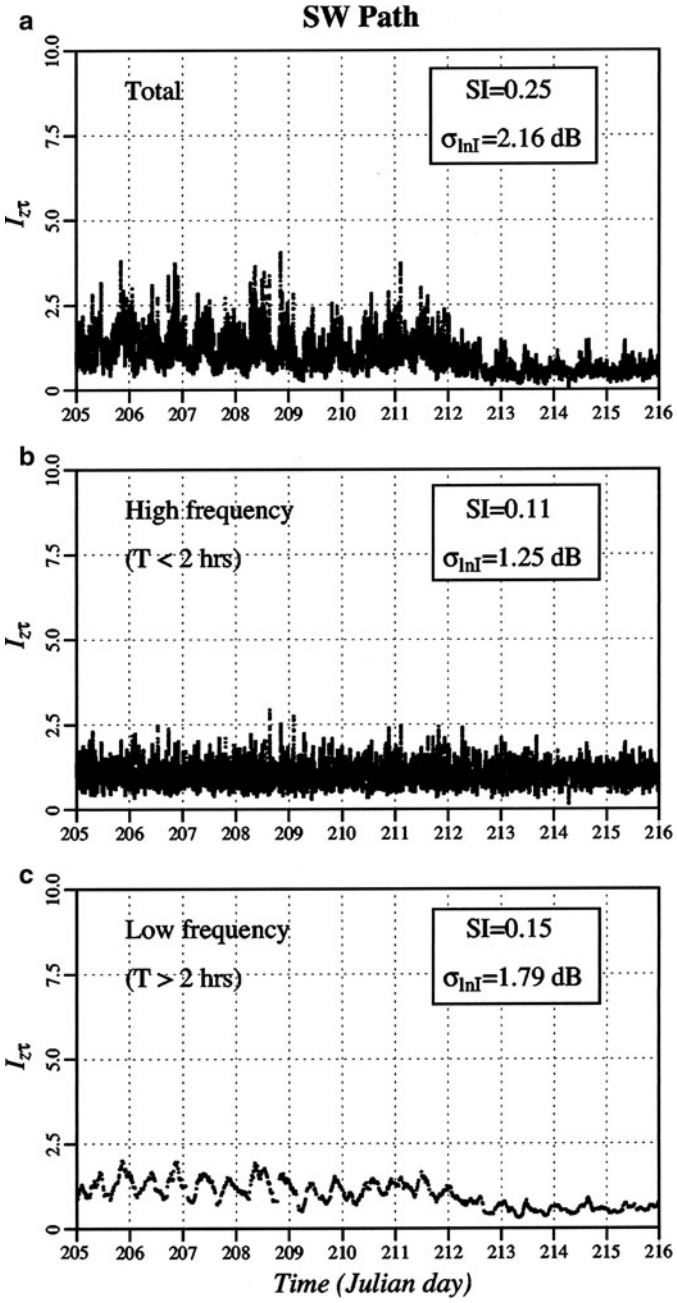


Fig. 5.13 Time series of depth and arrival time integrated energy for an acoustic path (the southwest path) in the PRIMER experiment. In panel (**a**) the unfiltered time series is shown, while panels (**b**) and (**c**) display the high and low frequencies, respectively

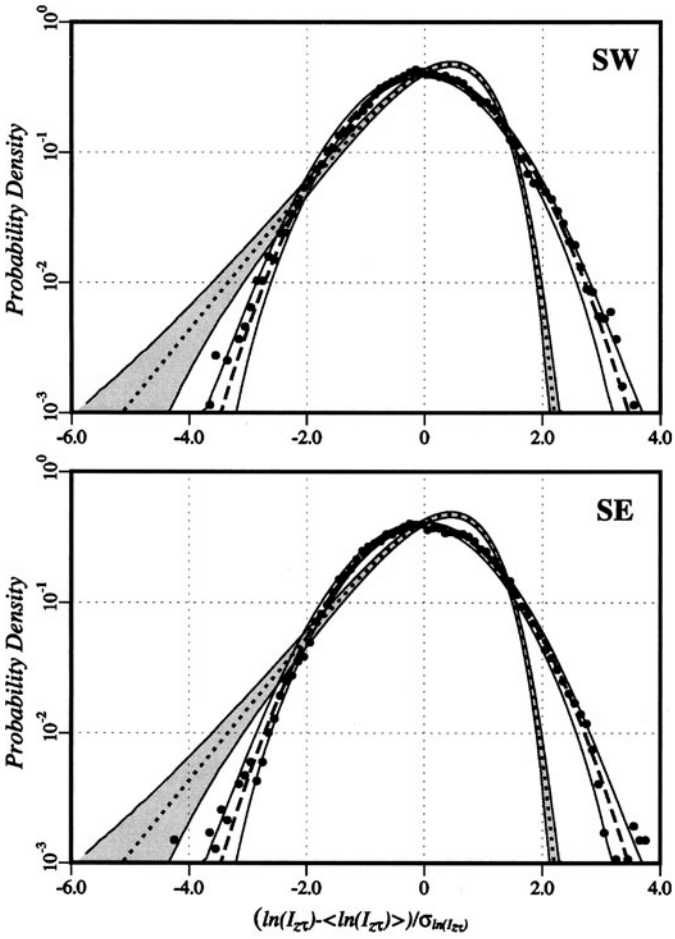


Fig. 5.14 Probability density functions of the high frequency component of the integrated intensity for the southwest and southeast transmission paths (*asterisk*) in the PRIMER experiment and the theoretical log-normal and exponential distributions (*gray* in-filling). Sampling error confidence intervals (95%) on the theoretical distributions (*dash*) are bound by *solid lines*

(middle and lower panels). We see that the high-frequency signal looks like a strong, fairly stationary, structureless scattering noise, whereas the weaker low-frequency time series looks like a tidal signal combined with some intermittent events. One would guess that the high-frequency, strong signal would show an exponential distribution, and indeed it does, as shown in Fig. 5.14. (This figure also shows the statistics for the shorter path, which are similar – both paths show saturated high frequency scattering.) The low-frequency component of the signal, on the contrary, shows a log-normal distribution (not displayed here) for both paths, indicating (along with the dB variation and SI) that the scattering by that component of oceanography is unsaturated.

Besides the statistics of the instantaneous intensity, there are also the peak intensity statistics and time and space averaged intensity statistics to consider. Deriving exact statistical distributions for all these fluctuations is a task that, to our knowledge, is not complete at present. However, due to the effects of (1) picking only the peak values and thus ignoring deep fades and (2) averaging a signal over time and/or space, the intensity fluctuations for all these cases are weaker overall. Thus, one would expect to be in the unsaturated scattering regime, and that log-normal statistics might be a good first approximation. This was shown to be the case in the Fredericks et al. (2005) paper and is probably a good first-order guideline to use, albeit not an exact result.

5.8.3 Broadband Fluctuations

As mentioned previously, our work has been primarily concerned with narrowband signals. However, broadband signals are just as important in practice, and so we turn to their treatment here. Specifically, we will look at the extensions of the narrowband statistics work by Makris (1996), Colosi and Baggeroer (2004), and Harrison and Harrison (1995). The first two works share the same physical underpinning: that for broadband signals, not all the multipaths are mutually interfering, as their arrival times differ and the pulses may be separable given a certain bandwidth. The last work points out an interesting way of obtaining broadband fluctuation calculations using narrowband calculations and doing range averaging as a proxy for a finite bandwidth.

In Makris' work, as mentioned, it was shown that the standard deviation of the transmission loss goes as $4.34\sqrt{(1/\mu)}$, where μ is the time-bandwidth product, assumed to be large. It was also shown, consistent with our earlier discussion, that the broadband fluctuation statistical distribution is log-normal for a large time-bandwidth product $(\mu>4)$, due to the averaging out of the fluctuations that the large frequency band provides.

In Colosi and Baggeroer, expressions are derived for the broadband weak and saturated regime SI, and the approach to saturation is also explored. It is seen clearly in the SI expressions that the main physical quantities governing the scintillations are (1) the number N of the interfering multipaths, be they rays or modes, (2) the bandwidth $Q = \Delta\omega/\omega_0$ of the signal, which tells you what time resolution between arrivals is possible, (3) the signal phase variability Φ^2, which describes how much the multipaths wander in time, which in turn increases their possibility of mutually overlapping one another and interfering, and (4) the various moments of the amplitude distribution, $\langle a^n \rangle$. The third quantity is where the scattering physics comes explicitly into play, and estimating this quantity remains a significant research challenge. Estimating the moments of the amplitude distribution is also a nontrivial problem, especially as the tails of the distribution can (and do) play an important role in the sonar equation.

The approach to saturation from either $SI > 1$ (geometric regime) or $SI < 1$ (diffractive regime) is a very interesting one, as one is quite often in the situation where the scattering is between the weak and saturated regimes. Colosi and Baggeroer (2004) find, for the narrowband and broadband cases, respectively, that the SI has the limiting behaviors:

$$SI^2 \cong 1 + \frac{1}{N}\left(\frac{\langle a^4 \rangle}{\langle a^2 \rangle^2} - 2\right) \quad \text{Narrowband,} \tag{5.94a}$$

$$SI^2 \cong 1 + \frac{1}{N}\left(Q\Phi\frac{\langle a^4 \rangle}{\langle a^2 \rangle^2} - 2\right) \quad \text{Broadband.} \tag{5.94b}$$

In examining these expressions, one sees that the sign of the second term on the right-hand side will determine the approach from above or below, and that this sign is determined by the competition between the second and fourth moments of the amplitude distribution (ala Creamer's previously discussed result) as well as other the three factors discussed before. We will not pursue detailed cases, but rather refer the reader to the original paper.

Finally, we note Harrison's paper (1995), which gives a prescription for how one can spatially average a narrowband calculation (e.g., a CW transmission loss versus range curve) over a given range bin, and by the range-frequency averaging equivalency discussed elsewhere in this book, obtain a TL curve whose fluctuations are those appropriate to a broadband transmission. This can significantly reduce calculational loads for broadband problems.

Chapter 6
Low-Frequency Bottom Reverberation in Shallow Water

6.1 Introduction: Sound Backscattering by the Sea Bottom

An extremely important part of shallow water acoustics is the study of long-range, low-frequency reverberation. In particular, one is studying acoustic wave backscattering by medium inhomogeneities, which are generally separated by a distance of a few to several tens of kilometers from a sound source and receiver. Interest in this area of study is generated by the fact that low-frequency reverberation affects the results of many methods of large-scale acoustic monitoring of the shelf areas of the World Ocean. This reverberation, along with its main role as an undesired noise signal, can also play the role of an additional source of environmental information. It is important to note that long-range reverberation in shallow water is mainly associated with acoustic backscattering by the sea bottom, since the backscattering coefficient of the sea surface at low frequencies ($f \leq 500$ Hz) is as a rule lower by 10–30 dB than the similar coefficient characterizing the sea bottom. The volume backscattering coefficient is also lower by several tens of decibels. Therefore, this chapter is mostly devoted to bottom reverberation, although the calculation methods described are in large part applicable to surface reverberation. We will only consider long-range reverberation here; that is, the scattered acoustic field is analyzed at a reception point at a distance r from the bottom scattering region, and r satisfies the condition $r \gg H$, where H is the waveguide depth.

Low-frequency, long-range reverberation is a complex physical phenomenon depending on both sound backscattering characteristics and the features of long-range propagation of acoustic waves in a near-bottom waveguide (Grigor'ev et al. 2004). Here, the backscattering characteristics are the frequency and angular dependence of the backscattering coefficient and its statistical properties. The propagation features are the sound attenuation in the waveguide, the waveguide dispersion, and the fluctuations of the forward and scattered acoustic signals, caused by mesoscale hydrodynamic perturbations. Let us now consider these characteristics and the propagation features in the context of low-frequency reverberation.

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_6,
© Springer Science+Business Media, LLC 2012

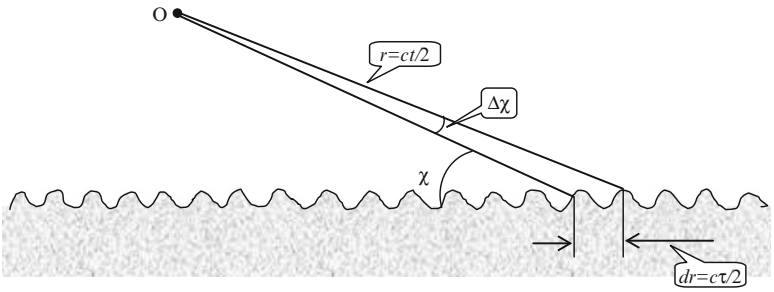


Fig. 6.1 Experimental configuration for backscattering coefficient measurements; O is the source/receiver point

The main parameter characterizing sound backscattering by the sea bottom is the backscattering coefficient [see, for example, Buntchuk and Zhitkovskii (1980)] given by

$$m_b = \frac{\langle W^{sc} \rangle}{I^{in} A_{sc}}, \tag{6.1}$$

where $\langle W^{sc} \rangle$ is the average acoustic power scattered by a unit bottom surface area in a given direction per unit solid angle, A_{sc} is the ensonified surface area, and I^{in} is the intensity of the incident acoustic wave near the scattering area. Averaging is carried out over a set of various realizations of the bottom surface. In addition to the scattering coefficient, the scattering strength $S_b = 10 \lg m_b$ is often used as well.

The quantities m_b and S_b can depend on the grazing angle χ^{in} of the incident wave with respect to the bottom, the frequency f, and the grazing angle χ^{sc} of the scattered wave, as well as on the bottom surface relief and the bottom structure. Before discussing these dependencies, we note a very important circumstance in studying long-range reverberation. Most known, in-situ measurements of the backscattering coefficient, in particular at rather low frequencies, were carried out using a scheme (shown in Fig. 6.1) employing a combined nondirectional sound source and receiver. Pulsed signals of short duration τ were almost invariably used. The distance r to the bottom scattering area, shaped as a circular ring, was smaller than the sea depth. In other words, although the sound receiver was in the far-field (Fraunhofer) region with respect to the scattering area, the latter was within the "line of sight." Thus, there was specifically no "waveguide" (multiply reflected) propagation of sound in these measurements. The same measurement features are also inherent in other common experimental methods for determining the coefficient m_b, for example, those applying directional source/receiver arrays or explosive sound sources.

In our measurement scheme, the area A_{sc}, from which scattered signals arrive at the (nonacoustic) emission–reception point O with a delay within τ, is calculated by the following formula[1]:

$$A_{sc} = 2\pi r \cos\chi \frac{c\tau}{2\cos\chi} = \pi c\tau r, \tag{6.2}$$

where $r = ct/2$ is the distance from the point O to the scattering area, t is the total time of forward and backscattered signal propagation, and c is the velocity of sound, which for such experiments is generally assumed to be independent of depth so that $\chi = \chi^{in} = \chi^{sc}$. Taking into account the fact that the intensity varies with distance the same as in free space ($I^{in} = I_0(r_0/r)^2$; $\langle W^{sc}\rangle = \langle I^{sc}\rangle r^2$), (6.1) takes on the form commonly used in experiments for determining the backscattering coefficient [see Buntchuk and Zhitkovskii (1980)]:

$$m_b = \frac{\langle I^{sc}\rangle r^3 10^{0.2\beta_w r}}{I_0 \pi c\tau r_0^2}, \tag{6.3}$$

where I_0 is the emitted signal intensity at the reference distance r_0 from the source, $\langle I^{sc}\rangle$ is the averaged scattered signal intensity, and β_w is the sound attenuation coefficient in water (dB/m). The appropriate grazing angle is estimated in terms of the total propagation time t: $\chi = \arcsin(t_0/t)$, where t_0 is the time of sound propagation to the bottom and back by the shortest path.

Currently, there is a large body of experimental data measurements of the backscattering coefficient for the shallow sea bottom. Let us attempt to characterize these results. First of all, the wide range of values of this coefficient should be noted. This is due to the fact that sound backscattering by the bottom is caused not only by bottom relief undulations, but also by the structure and properties of the upper layer of bottom sediments which, as is well known, are distinguished by their wide variety. In particular, the shelf contains all sediment types, from fine clays and silts to coarse-grained sands, pebbles, cockleshells, and even boulders. For representative samples of the backscattering coefficient for various sediments, as well as its dependence on frequency and grazing angle, we note the data reviews of How-Kin and Chesterman (1968) and Buntchuk and Zhitkovskii (1980). As is noted in these reviews, for most in-situ measurements, the backscattering coefficient is almost independent of frequency. This feature is inherent to all shallow water regions and is independent of the seabed material type. To illustrate this, Fig. 6.2 shows the results obtained by various authors. We see that only the results of McKinney and Anderson (1964) differ from the general tendency.

Analysis of the experimental data also shows that the backscattering coefficient decreases versus the various bottom types in the series stony soil, silt and siltstones, and sand. However, the differences are small; in particular, the difference between average

[1] In this measurement scheme, the value of τ is selected to be sufficiently small so that $\Delta\theta << \theta$.

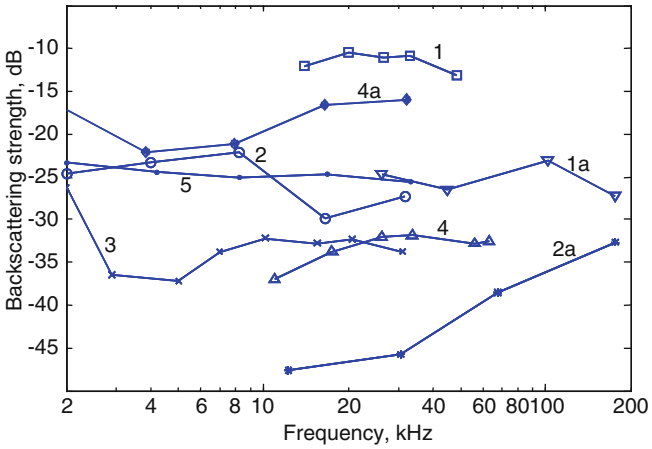


Fig. 6.2 Frequency dependence of the backscattering strength for various types of bottom sediments (1) stony bottom, $\chi = 30°$ (Urick 1954); (1a) the same, $\chi = 10°$ (McKinney and Anderson 1964); (2) sand $\chi = 70°$ (Buntchuk and Zhitkovskii 1980), (2a) the same, $\chi = 10°$ (McKinney and Anderson 1964); (3) siltstones, $\chi = 10°$ (Volovov and Zhitkovskii 1977); (4) silt, $\chi = 30°$ (Urick 1954); (4a) the same, $\chi = 70°$ (Buntchuk and Zhitkovskii 1980); and (5) the data averaged over regions with different bottom types, $\chi = 40°$ (Buntchuk and Zhitkovskii 1980)

coefficients m_b for stony and sandy bottoms is about 10 dB; for silty and sandy bottoms, it is only 2–4 dB. It is important to note that these differences take place for grazing angles χ less than 50°.

The averaged angular dependences of the backscattering strength $S_b(\chi)$ for shallow areas with the above-mentioned three basic types of bottom sediments are shown in Fig. 6.3. We can see that S_b increases quite smoothly for grazing angles up to $\chi \cong 45$–$50°$ and then grows abruptly. The curves in Fig. 6.3, as well as detailed analyses of the results of nonaveraged experimental dependences $S_b(\chi)$ measured in various shelf areas [see, for example, Volovov and Zhitkovskii (1977); McKinney and Anderson (1964); How-Kin and Chesterman (1968); Urick (1954)], show that the angular dependence of the backscattering coefficient, even at small χ, does not comply with the well-known Lambert law $m_b = g_{sc}^2 \sin^2\chi$, which describes bottom sound scattering in the deep ocean very well. Here, g_{sc} is a constant factor depending on the bottom sediment type. The angular dependence of $m_b(\chi)$ at small grazing complies more with the empirical Lommel–Seeliger law $m_b = g_{sc}^2 \sin\chi$. This difference from the classical Lambert law is apparently caused by the following factor. Sound backscattering in regions with a flat bottom, which is typical of shallow water, arises largely as a result of the interaction of the acoustic waves with inhomogeneities in the upper sediment layer. These inhomogeneities are pronouncedly anisotropic: they are large scale in the horizontal plane and small scale over depth. For the case of sound scattering by such random volume inhomogeneities, the dependence $m_b(\chi)$ complies with the Lommel–Seeliger law, and the coefficient m_b depends slightly on frequency. This was shown for the first time in Lysanov (1980).

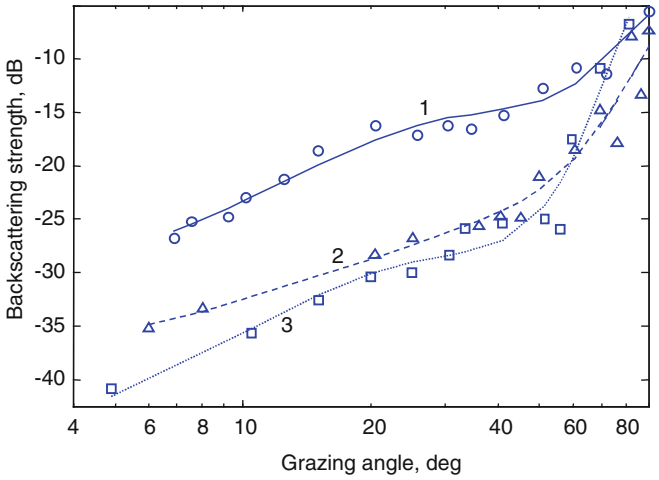


Fig. 6.3 Averaged angular dependence of the backscattering strength for shallow areas with three basic types of bottom sediments (1) stony bottom, (2) siltstones and silts, and (3) sand. The data are averaged over the frequency range 10–100 kHz

A variety of models, which describe sound scattering by bottom relief undulations and inhomogeneities of the upper layer of the sea bottom, have been constructed for the more general bistatic case, where the sound source and receiver are at different spatial locations [see, for example, Yamamoto (1996) and Williams and Jackson (1998)]. The dependences of the sound scattering coefficient depend on three angles in this case: $m_b = g_{sc}^2 F^2(\chi^{in}, \chi^{sc}, \Delta\vartheta^{sc})$. Here, $\Delta\vartheta^{sc}$ is the horizontal angle between vertical planes in which the vector of incident and scattered acoustic waves lies (for backscattering, $\Delta\vartheta^{sc} = 180°$). The coefficient m_b calculated using these bistatic models agrees adequately with experimental data and, moreover, weakly depends on frequency at backscattering and complies with the Lommel–Seeliger law at small angles $\chi = \chi^{in} = \chi^{sc}$. To illustrate this, Fig. 6.4 shows experimental and calculated backscattering strengths obtained in Jackson and Briggs (1992), as well as the $S_b(\chi)$ dependence complying with this law.

Rather than going into the details of the rather complex bistatic models, we will present only the simplified dependences of the function F at $\Delta\vartheta^{sc} = 180°$:

$$F^2\left(\chi^{in}, \chi^{sc}\right) = \sin^{1/2}\left(\chi^{in}\right)\sin^{1/2}\left(\chi^{sc}\right), \tag{6.4}$$

$$F^2\left(\chi^{in}, \chi^{sc}\right) = \sin\left(\frac{\chi^{in} + \chi^{sc}}{2}\right), \tag{6.5}$$

$$F^2\left(\chi^{in}, \chi^{sc}\right) = \left(\frac{2\sin(\chi^{in})\sin(\chi^{sc})}{\sin(\chi^{in}) + \sin(\chi^{sc})}\right). \tag{6.6}$$

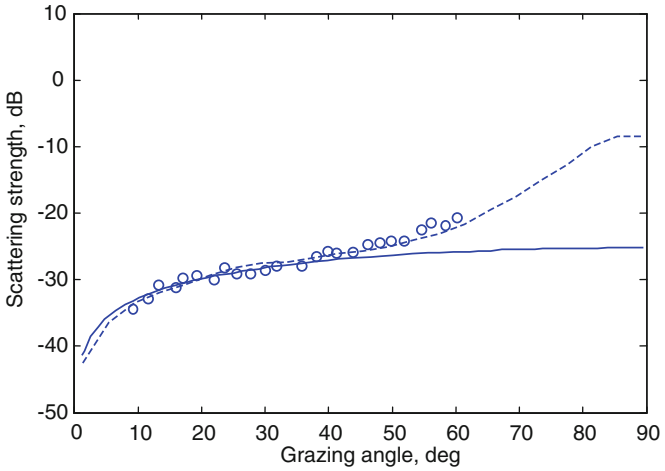


Fig. 6.4 Angular dependence of the backscattering strength for the "Quinault" shallow water region near the western coast of Washington State; experimental data (*circles*) (frequency is 25 kHz), calculated dependence (*dashed curve*) (Jackson and Briggs 1992), and the empirical dependence corresponding to the Lommel–Seeliger law (*solid curve*)

These dependences were proposed in Jin and Zhang (1990) and Li et al. (2003) for small grazing angles. All the functions (6.4)–(6.6) are identical to the empirical dependence defined by the Lommel–Seeliger law at $\chi = \chi^{in} = \chi^{sc}$. We also note that the function F is given by

$$F^2\left(\chi^{in}, \chi^{sc}\right) = \sin\left(\chi^{in}\right)\sin\left(\chi^{sc}\right) \tag{6.7}$$

for isotropic scattering, corresponding to the Lambert law.

6.2 Mode Theory of Bottom Reverberation in a Regular Waveguide

The conventional approach to the calculation and modeling of long-range, low-frequency reverberation in shallow water is based on the modal description of an acoustic field in a waveguide. This approach was first proposed in Bucker and Morris (1968). Soon after, a long series of studies were initiated by the Chinese scientists Shang and Chang (1975), Tang (1989), Zhang and Jin (1987), Jin and Zhang (1990) and by Kudryashov (1999), where this approach was developed to estimate reverberation in various shallow water areas. The mode theory of bottom reverberation is clearly stated in general form in Ellis (1995), where the influence of

waveguide intermode dispersion[2] on the characteristics of long-range shallow water reverberation was analyzed for the first time. This theory postulates independent scattering of the individual waveguide modes incident on a scattering area of the bottom. Each mode is associated with a mode-forming (Brillouin) beam, and the scattering process is described in beam terms. The scattered acoustic field also consists of Brillouin beams corresponding to similar modes. Following this logic, one can separate individual components $P^{in}_{m\mu}$ of the acoustic field of a point source, incident on the scattering area of the sea bottom. Let us assume for now that the source emits narrow-band pulsed signals. Let us further consider the m-th mode of the incident acoustic field, which is scattered into the μ-th mode and is then detected by the receiver at the instant t (i.e., t is the time of an acoustic signal propagation from the sound source to the scattering area and back to the sound receiver). In the measured scattered field, the component $P^{sc}_{m\mu}$ is identified, which corresponds to the μ-th mode and depends only on $P^{in}_{m\mu}$. Here, $P^{in}_{m\mu}$ and $P^{sc}_{m\mu}$ are the complex amplitudes of the acoustic field components. For the cylindrical coordinate system used with a downward z axis and the source at the waveguide[3] point with coordinates $(0, z_0)$, the quantity $P^{in}_{m\mu}$ is given by (3.25)

$$P^{in}_{m\mu}(r_{m\mu}, z) = \sqrt{\rho_0 c_0 W_0}\,\frac{\Psi_m(z_0)\Psi_m(z)}{\sqrt{\xi_m r_{m\mu}}}\exp(i\pi/4)\exp(i\xi_m r_{m\mu}), \tag{6.8}$$

where m and μ are the mode indices of the incident and scattered fields, respectively $(m,\ \mu = 1, 2, \ldots M)$, $z = H$, and

$$r_{m\mu} = t\left(1\Big/c^{\mathrm{gr}}_m + 1\Big/c^{\mathrm{gr}}_\mu\right)^{-1}. \tag{6.9}$$

It is assumed that the sound receiver is at the point with coordinates $(0, z_r)$; that is, the source and receiver are at the same horizontal distance from the scattering area, but are separated in depth.[4]

The scattering of the acoustic field $P^{in}_{m\mu}$ is shown schematically in Fig. 6.5. To determine the acoustic field at the reception point, we will invoke some known relations used by Grigor'ev et al. (2001) to solve the scattering problem in a waveguide. In these relations, the excitation coefficients $b_{m\mu}(z)$ of the waveguide

[2] As for the intramode dispersion, it manifests itself at appreciably longer distances than typical distances to the bottom scattering region, affecting long-range reverberation in shallow water (see Chap. 4).

[3] To simplify the basic theory, the waveguide is hereafter assumed to be horizontally homogeneous, although the present theory of long-range reverberation is also valid for a smoothly inhomogeneous waveguide.

[4] This assumption is not key and was introduced only to simplify the equations. The theory for bistatic scattering, when sound source and receiver are located at distinct points $(\Delta\vartheta^{sc} \neq 180^\circ)$, will be described below.

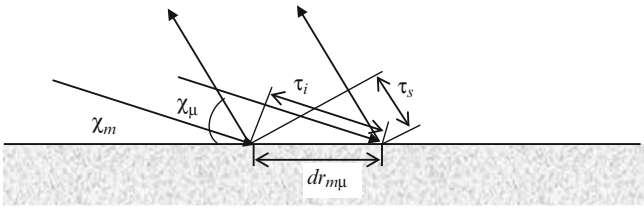


Fig. 6.5 Configuration for mode-forming beam scattering

modes of the scattered field are expressed in terms of the mode amplitudes of the incident field and the mode transformation matrix $T_{m\mu}$ (in our case, it is the scattering matrix appropriate to the bottom area). According to these relations, the component $P^{sc}_{m\mu}$ can be written as

$$P^{sc}_{m\mu}(r_{m\mu}, z_r) = \sqrt{\frac{1}{8\pi}}\exp(i\pi/4)\frac{b_{m\mu}(z)\Psi_\mu(z_r)}{\sqrt{\xi_\mu r_{m\mu}}}\exp(i\xi_\mu r_{m\mu}). \tag{6.10}$$

The coefficients $b_{m\mu}(z)$ are given by

$$b_{m\mu}(z) = \Psi_\mu(z)T_{m\mu}(r_{m\mu})P^{in}_{m\mu}(r_{m\mu}, z) \tag{6.11}$$

and the mode transformation matrix is defined by the bottom scattering properties. In most papers devoted to long-range reverberation, it is assumed that scattering from the sea bottom is deterministic and completely defined by the backscattering coefficient. In this case, the quantity $T_{m\mu}$ is written as

$$T_{m\mu}(r_{m\mu}) = \sqrt{m_b dS_{m\mu}} = g_{sc}F(\chi_m, \chi_\mu)\sqrt{dA_{m\mu}}, \tag{6.12}$$

where $dA_{m\mu}$ is the bottom area (a circular ring for a nondirectional sound source) where rescattering from mode m into mode μ occurs (see Fig. 6.5), and χ_m and χ_μ are the grazing angles of the modal beams. The area $dA_{m\mu} = 2\pi r_{m\mu}dr_{m\mu}$. The quantity $dr_{m\mu}$ is determined by solving the simple equation $\tau = \tau_{in} + \tau_{sc}$, which follows from the condition that the difference in the propagation times of signals reflected from the inner and outer boundaries of the ring is equal to τ. Here, τ_{in} and τ_{sc} are the delays of the incident and reflected waves, respectively (see Fig. 6.1). Since $\tau_{in} = dr_{m\mu}\cos\chi_m/c$, $\tau_{sc} = dr_{m\mu}\cos\chi_\mu/c$, where $\cos\chi_{m,\mu} = q_{m,\mu}/k$, and the quantity $dr_{m\mu}$ can be written as

$$dr_{m\mu} = \tau\left(\frac{1}{c^{ph}_m} + \frac{1}{c^{ph}_\mu}\right)^{-1}, \tag{6.13}$$

where c_m^{ph} and c_μ^{ph} are the phase velocities of the m-th and μ-th modes, respectively ($c_{m,\mu}^{ph} = 2\pi f/q_{m,\mu}.$).

The time dependence of the amplitude $P^{sc}(t)$ of the scattered acoustic field at the reception point during the emission of a short ($\tau \ll t$) pulsed signal is determined by summing relations (6.9) and (6.10) over all possible values of m and μ. Taking into account (6.11) and (6.12), $P^{sc}(t)$ is given by

$$P^{sc}(t) = \frac{g_{sc}}{2}\exp(i\pi/4)\sum_{m,\mu}^{M}\frac{\Psi_\mu(z_r)\Psi_\mu(z)F(\chi_m,\chi_\mu)\sqrt{dr_{m\mu}}P_{m\mu}^{in}(r_{m\mu},z)}{\sqrt{\xi_\mu}}\exp(i\xi_\mu r_{m\mu}), \tag{6.14}$$

where t and $r_{m,\mu}$ are connected by the expression (6.9) and $z = H$.

The average intensity I^{sc} of the scattered acoustic field, averaged over the time interval t_a, where $t_a \geq \Lambda_l/c$, and where Λ_l is the characteristic interference beating distance of the incident and scattered acoustic fields, is of interest for sonar applications. Taking into account relations (6.10), (6.13), and (6.14), the quantity I^{sc} is given by

$$I^{sc} = \frac{\overline{(P^{sc}(t))^2}}{\rho(z_r)c(z_r)} \cong \frac{W_0 g_{sc}^2\tau}{4}\sum_{\mu}^{M}\frac{|\Psi_\mu(z_r)|^2|\Psi_\mu(z)|^2}{|\xi_\mu|}$$
$$\times\sum_{m}^{M}F^2(\chi_m,\chi_\mu)\frac{|\Psi_m(z_0)|^2|\Psi_m(z)|^2}{|\xi_m|}\frac{1}{r_{m\mu}}\frac{c_m^{ph}c_\mu^{ph}}{c_m^{ph}+c_\mu^{ph}}\exp\left(-\gamma_m r_{m\mu}-\gamma_\mu r_{m\mu}\right). \tag{6.15}$$

Hereafter, we will suppose that $\rho(z_r) \cong \rho_0$ and $c(z_r) \cong c_0$. We also note that the overbar denotes time averaging. Observe that in (6.15) and in any following equations where we calculate the averaged intensity of the scattered field, the spatial averaging over the interference beating distance takes place for both the incident and the scattered sound fields.

We note that waveguide dispersion is neglected in many papers [see, for example, Zhang and Jin (1987)] when calculating the average intensity of the scattered field; that is, it is assumed that $c_m^{ph} = c_\mu^{ph} = c_m^{gr} = c_\mu^{gr} = c$. In this case, $r_{m\mu} = r = tc/2$ and $dr_{m\mu} = c\tau/2$, and so expressions like (6.15) become simpler, e.g.,

$$I^{sc} = \frac{\overline{(P^{sc}(t))^2}}{\rho(z_r)c(z_r)} \cong \frac{W_0 g_{sc}^2\tau c}{8r}\sum_{\mu}^{M}\frac{|\Psi_\mu(z_r)|^2|\Psi_\mu(z)|^2}{|\xi_\mu|}$$
$$\times\sum_{m}^{M}F^2(\chi_m,\chi_\mu)\frac{|\Psi_m(z_0)|^2|\Psi_m(z)|^2}{|\xi_m|}\exp\left(-\gamma_m r-\gamma_\mu r\right). \tag{6.16}$$

Expression (6.16) is commonly used in shallow water acoustics to simulate and estimate long-range, low-frequency reverberation. This expression can also be used to estimate the coefficient g_{sc} by considering the experimentally measured dependence of $I^{sc}(r)$. We note that such an estimate is generally necessary for each region studied, since the known values of g relate mostly to shallow water frequencies above 1 kHz and grazing angles of more than 10° (see Sect. 6.1). But, on the contrary, long-range reverberation in shallow water is mainly observed in the frequency range 50–500 Hz, and the grazing angles of mode-forming beams responsible for this reverberation are often equal to only a few degrees.

At this point, it is appropriate to explain that it is not practical to measure the backscattering coefficient at low frequency and at small grazing angles by using the standard method described in Sect. 6.1. Specifically, at frequencies in the order of 100 Hz, there is no way to obtain angular resolution of the mildly sloping incident and scattered sound rays to within an accuracy of ~1°. Because of this, some investigators, for example, Cable et al. (1997), Grigor'ev et al. (2004), have used the integrated backscattering coefficient $\langle m_b \rangle$ and integrated backscattering strength $\langle S_b \rangle$ which are equal to

$$\langle m_b \rangle = \frac{\langle W^{sc} \rangle}{I^{in} A_{sc}} = \frac{\langle p_o^{sc} \rangle^2 r_0^2}{\langle p^{in} \rangle^2 A_{sc}}, \quad \langle S_b \rangle = 10 \lg \langle m_b \rangle, \tag{6.17}$$

respectively. Here, the angle brackets also denote averaging over grazing angle. In other words, here $\langle W^{sc} \rangle$ is the averaged acoustic power scattered by a bottom surface area A_{sc} per unit solid angle, with averaging done not only over a set of various realizations of the bottom area, but also over grazing angle. $\langle p_0^{sc} \rangle$ is the rms sound pressure of the scattered field in the vicinity of the bottom scattering area at the distance r_0. (Traditionally, r_0 is equal to 1 m.) $\langle p^{in} \rangle$ is the rms sound pressure of the incident wave. The values $\langle m_b \rangle$ and $\langle S_b \rangle$ can be measured via experiment. With this in mind, we rewrite (6.17) in a form that is suitable for measurement:

$$\langle S_b \rangle = 20 \lg \frac{\langle p^{sc} \rangle}{p_{st}} - 20 \lg \frac{p_0}{p_{st}} + 40 \lg \frac{p_0}{\langle p^{in} \rangle} - 10 \lg \frac{A_{sc}}{r_0^2}, \tag{6.18}$$

where $\frac{p_0}{p_{st}}$ is the ratio of the levels at the source to a standard reference level. Here, we assume that sound source and receiver are colocated and that $\langle p^{sc} \rangle = \langle p_0^{sc} \rangle (\langle p^{in} \rangle / p_0)$. $\langle p^{sc} \rangle$ is the rms sound pressure of the scattered field at the reception point at a large distance from the scattering area. This equation can also be written in terms of the familiar sonar equation:

$$\bar{S} = \mathrm{RL} - \mathrm{SL} + 2\mathrm{TL} - 10 \lg \frac{A_{sc}}{r_0^2}, \tag{6.19}$$

where RL is reverberation level ($\mathrm{RL} = 20\lg(\langle p^{sc}\rangle/p_{st})$), SL is signal (source) level ($\mathrm{SL} = 20\lg(p_0/p_{st})$), and TL is transmission loss ($\mathrm{TL} = 20\lg(p_0/\langle p^{in}\rangle)$).

It is also possible to write the relationship for the integrated backscattering strength by using the mode theory of bottom reverberation. Neglecting waveguide dispersion gives the following expressions for the averaged intensity of incident I^{in} and scattered I^{sc} fields:

$$I^{in} = W_0 \sum_m \frac{|\Psi_m(z_0)|^2 |\Psi_m(z)|^2}{r|\xi_m|} \exp(-\gamma_m r), \tag{6.20}$$

$$I^{sc} = \langle W^{sc}\rangle \sum_\mu \frac{|\Psi_\mu(z_r)|^2 |\Psi_\mu(z)|^2}{r|\xi_\mu|} \exp(-\gamma_\mu r). \tag{6.21}$$

Comparing (6.21) and (6.16) yields $\overline{W_{\mathrm{a}}^{sc}}$. Furthermore, substituting (6.20) in (6.17), we obtain

$$\langle S_b\rangle = 10\lg \frac{g_{sc}^2 \sum\limits_{m,\mu} \frac{|\Psi_\mu(z_r)|^2 |\Psi_\mu(z)|^2 |\Psi_m(z_0)|^2 |\Psi_m(z)|^2}{|\xi_m||\xi_\mu|} F^2(\chi_m, \chi_\mu) \exp(-\gamma_m r - \gamma_\mu r)}{8\pi \sum\limits_\mu \frac{|\Psi_\mu(z_r)|^2 |\Psi_\mu(z)|^2}{|\xi_\mu|} \exp(-\gamma_\mu r) \sum\limits_m \frac{|\Psi_m(z_0)|^2 |\Psi_m(z)|^2}{|\xi_m|} \exp(-\gamma_m r)}. \tag{6.22}$$

As is evident from (6.22), the integrated backscattering strength does not solely represent the scattering properties characteristic of the sea bottom. It also depends on the waveguide parameters and also includes the distance r. The angular dependence $\langle S_b(\chi)\rangle$ is not available by definition. The frequency dependence $\langle S_b(f)\rangle$ usually measured in experiments concerning long-range reverberation (Cable et al. 1997) can differ from the similar dependence $S_b(f)$. On the contrary, when the function $F(\chi^{in}, \chi^{sc})$ is known, it is possible to estimate the coefficient g_{sc} by using (6.22) and the measured value $\langle S_b\rangle$.

Furthermore, we note that it is often necessary to solve the problem of sound scattering by the sea bottom simultaneously with the problem of scattering at localized inhomogeneities in the waveguide. Thus, it is important that the above-considered mode theory also be developed for the case of scattering at localized inhomogeneities in waveguides, including the case when the source and receiver are placed at different points and the angle ($\Delta\vartheta^{sc} \neq 180^\circ$). This was done in Kravtsov et al. (1983) and Makris and Ratital (2001), where the problem of scattering in the waveguide was solved in the general form using the so-called scattering amplitude. This approximate approach to a problem of scattering at localized inhomogeneities in the waveguide is described in more detail in Appendix E.

6.3 Numerical Simulation of Low-Frequency Bottom Reverberation

The theoretical model of backscattering described above makes it possible to carry out numerical experiments and to understand the dependence of bottom reverberation on bottom scattering properties, and the features of low-frequency sound propagation in a waveguide. Examples of calculations of the average intensity of the scattered field are shown in Figs. 6.6–6.8. Figure 6.6 shows the dependence of I^{sc} on the distance r to the bottom scattering area, both taking and not taking into account the waveguide dispersion (see expressions (6.15) and (6.16), respectively). A waveguide of depth $H = 120$ m with a constant sound velocity $c = 1{,}475$ m/s was used in the numerical simulation. The bottom parameters, that is, the complex refractive index and the density, were taken as $n_1 = 0.84(1 + 0.006i)$ and $\rho_1 = 1.9$ g/cm^3, respectively. The function $F(\chi_m, \chi_\mu)$ was calculated using relation (6.7). The coefficient g_{sc} had the value $g_{\text{sc}} = 0.19$ ($S_{\text{b}} \cong -30$ dB at $\chi = 10°$). The acoustic signal frequency was $f = 230$ Hz, the pulse duration was $\tau = 0.1$ s, and the radiated power was $W_0 = 2$ kW. It was assumed that the source and receiver were colocated at the depth $z_0 = z_r = 120$ m.

We can see in Fig. 6.6 that the dispersion properties of the propagation channel cause a change in the average intensity of less than 1 dB, which is insignificant in practical terms. However, as shown in Ellis (1995), this small difference is observed only for the case of a sea bottom whose scattering properties do not change drastically with distance[5]: $\Delta g_{sc}/g_{sc} \leq 1$ at the distance $\Delta r = c\tau_1/2$, where τ_1 is

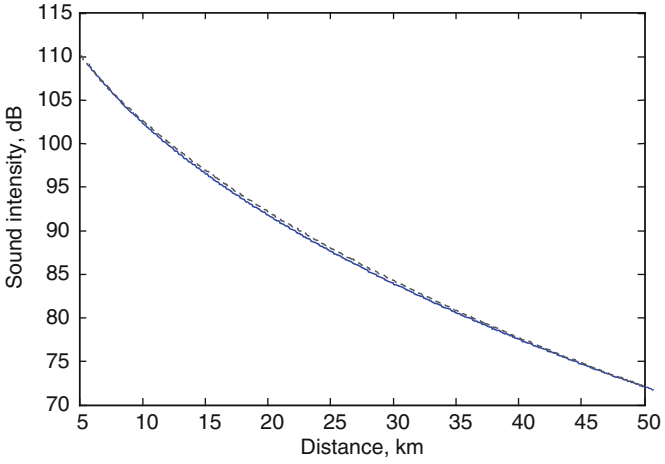


Fig. 6.6 Dependence of the average intensity of the scattered field on the distance to the scattering area. The *solid* and *dashed curves* were constructed by taking and not taking into account the dispersion properties of the waveguide, respectively

[5] Such changes are possible, e.g., in the cases of an underwater relief feature or when a sandy bottom contains regions with rocky outcrops.

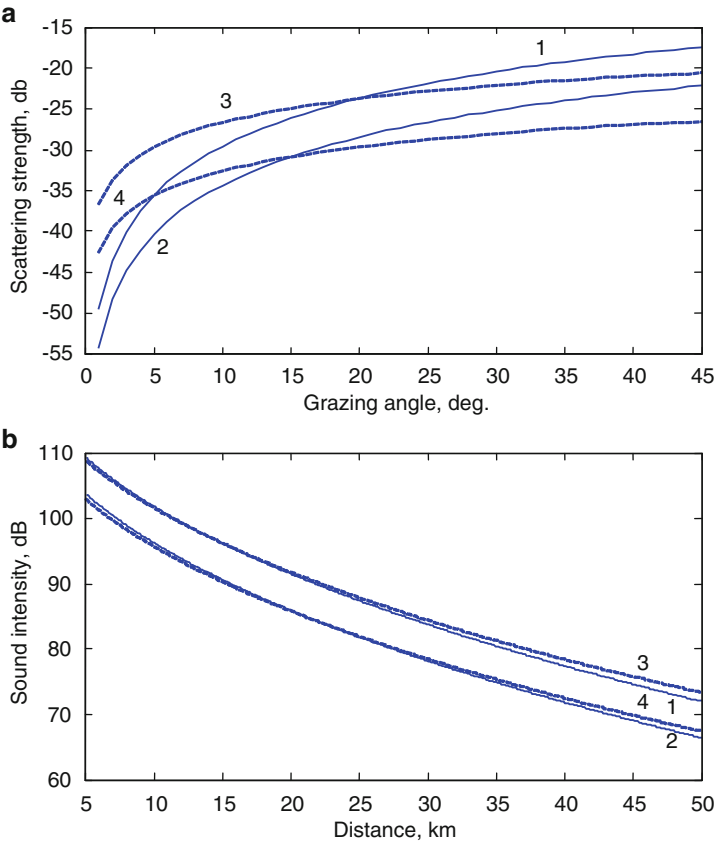


Fig. 6.7 (**a**) Angular dependence of the backscattering strength at the sea bottom and the corresponding dependence of the average intensity (**b**) Dependence of the scattered field on the distance to the scattering area. *Solid curves* (1,2) correspond to (6.7) dependence (the Lambert law) with the coefficients g_{sc}=0.19 and 0.1, respectively; dashed curves (3,4) correspond to dependence (6.4) with the coefficients $g_{sc} = 0.11$ and 0.056, respectively

the duration of a signal that has passed the distance r in the waveguide. It should also be emphasized that the value of τ_1 is much longer than that of τ due to waveguide dispersion (see Chap. 4). Therefore, a more correct (from a physical viewpoint) modification to the approximate relation (6.16) involves the substitution of $dr_{m\mu} = c\tau_1/2$ in (6.15), and the introduction of the effective power $W_{ef} = W_0\tau/\tau_1$ for the source. This change reflects the fact that the ensonified bottom area increases due to waveguide dispersion, whereas the energy density incident on it decreases. We thus deal as though we had a source emitting lower intensity but longer signals. This physical interpretation is possible also due to the fact that the two factors have competing effects on τ_1 in shallow water. On the one hand, the signal duration increases with the distance from the source due to intermode dispersion. On the other hand, the number of energy-carrying modes

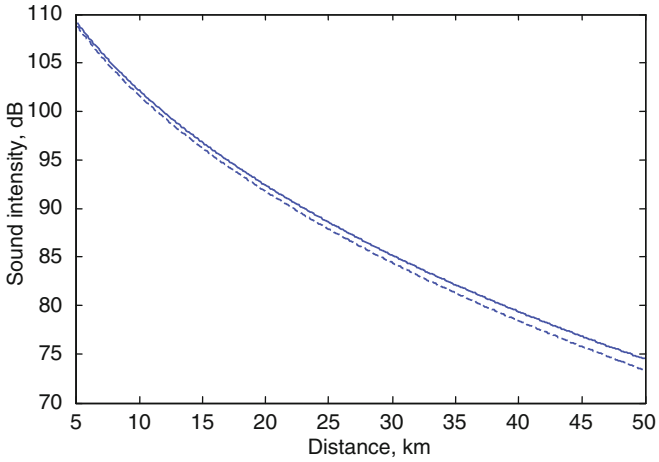


Fig. 6.8 Dependence of the average intensity of the scattered field on the distance to the scattering area for the isovelocity waveguide (*dashed curve*) and the waveguide with the sound velocity profile shown in Fig. 4.16 (*solid curve*)

decreases with distance due to rapid strengthening of waveguide mode attenuation as the mode index increases, and thus the effective value of τ_1 decreases. As a result, the duration τ_1 depends weakly on distance if the latter is of the order of several tens of kilometers.

Figure 6.7 shows $I(r)$ variations for various values of the coefficients g_{sc} and for various dependences of $F(\chi^{in}, \chi^{sc})$ corresponding to the Lambert and Lommel–Seeliger laws. The calculations were carried out for the above-discussed model waveguide, and the acoustic signal parameters were also unchanged. We can see that the $I(r)$ variation weakly depends on the form of the function $F(\chi^{in}, \chi^{sc})$. With appropriate selection of the coefficient g_{sc}, the $I^{sc}(r)$ variations are almost identical for both laws of $F(\chi^{in}, \chi^{sc})$ variation. The numerical experiments also show that the calculated values of $I^{sc}(r)$ are almost identical when using the same value of the coefficient g, but different model dependences $F(\chi^{in}, \chi^{sc})$. Thus, in the absence of a reliable experimental measurement for $F(\chi^{in}, \chi^{sc})$, the bottom reverberation can still apparently be simulated and analyzed using either the Lambert or the Lommel–Seeliger laws. The latter, as mentioned above, is more appropriate to the actual situation in shallow water. We also note that different values of the coefficient g_{sc} should be taken in this case (see Fig. 6.7).

Figure 6.8 shows the dependence of the scattered field intensity for a waveguide with a vertical sound velocity profile having a negative gradient, typical of the shallow sea. The sound velocity profile, measured in a field experiment in the Barents Sea (Grigor'ev et al. 2004), is shown in Fig. 4.16. Other waveguide and acoustic signal parameters were the same as in previous calculations. The function $F(\chi^{in}, \chi^{sc})$ was calculated using (6.4), and the coefficient g_{sc} was 0.11. We can see in Fig. 6.8 that the change in the sound velocity profile does not have a significant

effect on the reverberation signal intensity when compared with the isovelocity model. It is interesting to note that the $I^{sc}(r)$ curve for the waveguide with a variable sound velocity profile in general lies above the similar curve calculated for the isovelocity waveguide. This seems to be due to the fact that reverberation signals are mostly generated due to scattering of the acoustic modes with the highest mode numbers, and thus with relatively large grazing angles for the modal beams. In the isovelocity waveguide, these modes are characterized by stronger attenuation.

6.4 Long-Range Reverberation Studies Using Extended Arrays

Nowadays, the so-called modal beam-forming arrays are commonly applied to acoustics studies on the continental shelf. These arrays are extended vertical emitting and receiving chains tuned to excite and detect individual modes of the acoustic field in the waveguide (see Sect. 8.4 and Chap. 10). Such chains must be sufficiently long and overlap a major part of the acoustic waveguide to achieve efficient single-mode resolution. It is obvious that the use of such matched arrays significantly improves the measurements for long-range reverberation (Grigor'ev et al. 2004). Indeed, the use of such arrays in a regular waveguide allows the neglect of waveguide dispersion,[6] and thus the calculation procedure for the scattered acoustic field must be modified.

As an example, let us consider the features of long-range reverberation for a "mode-forming" array tuned to the first waveguide mode. Hereafter, we will assume that the mode-forming array is located at the point with coordinate $(r = 0)$. To facilitate the comparison of reverberation signals received by an individual receiver and by a receiving array, we assume that the sum of the magnitudes of transmission coefficients over all the channels is equal to unity $\left(\sum_{i_r}^{I_r} \hat{k}_{i_r} \equiv \sum_{i_r}^{I_r} N_r \Psi_1^*(0, z_{i_r}) = 1\right)$. This gives the normalization factor

$$N_r = \left[\sum_{i_r=1}^{I_r} |\Psi_1(0, z_{i_r})|\right]^{-1}. \tag{6.23}$$

Here, i_r is the receiving hydrophone index, $i_r = 1, 2, \ldots, I_r$, where I_r is the number of array receivers. The previously discussed mode theory of bottom reverberation allows the derivation of an expression for the reverberation signal $s_{i_r}^{sc}(t)$ measured by each receiving element of the array, $s_{i_r}^{sc}(t) = \hat{k}_{i_r} P_{i_r}^{sc}(t)$, where $P_{i_r}^{sc}$ is the scattered field at the reception point, representing a sum of scattered fields

[6] The intermode dispersion is absent if one has the excitation and reception of individual waveguide modes.

(6.14) from each sound source. We note that according to expression (6.23), the dimensions of all the $s_{i_r}^{sc}(t)$ are in Pascals. Summing signals from all the receiving elements, we obtain the following expression for the receiving array response:

$$s^{sc}(t) = \frac{g_{sc}}{2} \exp(i\pi/4) \sum_{m,\mu}^{M} \frac{\hat{C}_\mu \Psi_\mu(z) F(\chi_m, \chi_\mu) \sqrt{c\tau/2}}{\sqrt{\xi_\mu}} P_m^{in} \exp(i\xi_\mu r), \tag{6.24}$$

where $t = 2z/c$, P_m^{in} is equal to P_m for $l = 1$ (10.8)

$$\hat{C}_\mu = N_r \sum_{i_r=1}^{I_r} \Psi_1^*(z_{i_r}) \Psi_\mu(z_{i_r}). \tag{6.25}$$

For our case, the magnitude of the weight factor $\hat{C}_1$ [see expressions (6.8) and (10.10)] for the first mode significantly exceeds the magnitudes of coefficients $\hat{C}_{m,\mu}$ for the indices of other modes $|\hat{C}_1| \gg |\hat{C}_{m,\mu}|$, where $m, \mu = 2, 3, ..., M$ for the case of extended emitting and receiving arrays overlapping a major part of the waveguide. Thus, the first mode is emitted and received. The effect of an increase in the pulse duration, associated with the intermode dispersion, is thus absent.

By analogy with (6.16), an expression can also be derived for the quantity $I_a^{sc} = \overline{(s^{sc}(t))^2}/\langle\rho\rangle\langle c\rangle$, which is the average intensity of the acoustic field detected by the mode-forming array

$$I_a^{sc} = \frac{g_{sc}^2 \tau c}{8r\langle\rho\rangle\langle c\rangle} \sum_{\mu}^{M} \frac{|\hat{C}_\mu|^2 |\Psi_\mu(z)|^2}{|\xi_\mu|} \times \sum_{m}^{M} F^2(\chi_m, \chi_\mu) \frac{|\hat{C}_m|^2 |\Psi_m(z)|^2}{|\xi_m|} \exp(-\gamma_m r - \gamma_\mu r). \tag{6.26}$$

Here, $\langle\rho\rangle$ and $\langle c\rangle$ are the density and sound speed averaged over the array aperture, respectively.

The values of I_a^{sc} calculated for the waveguide with the sound velocity profile given in Fig. 4.16 are shown in Fig. 6.9. The function $F(\chi^{in}, \chi^{sc})$ was calculated using (6.4), and the coefficient g_{sc} was taken as $g_{sc} = 0.11$. The acoustic signal parameters were the same as in the calculation shown in Fig. 6.8. It was assumed that the lengths of the receiving and radiating arrays were identical, each equal to 87 m; that each consisted of 30 equally spaced elements; and that the lower source–receiver pair was at the sea bottom. For comparison, the values of I^{sc} calculated for a combined solitary point source and receiver at the sea bottom are also shown in Fig. 6.9. The radiated powers of the array and the solitary source were taken as equal. We can see in Fig. 6.9 that the reverberation signals for the mode-forming arrays are significantly smaller in magnitude than those for the case of the nondirectional source and receiver, which excite and receive the entire mode

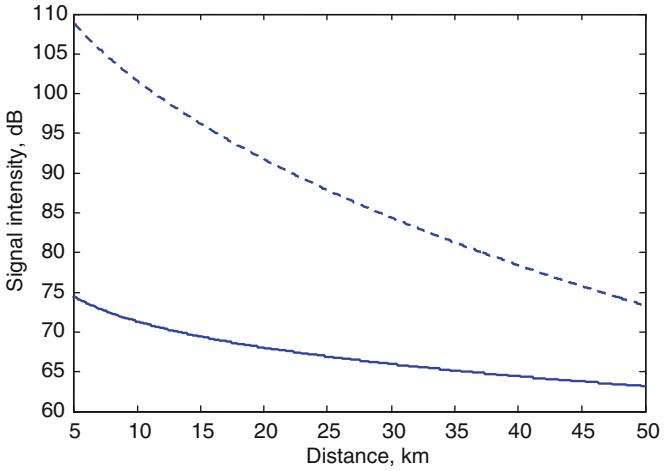


Fig. 6.9 Levels of reverberation signals calculated in using "mode forming" arrays (*solid curve*) and a point source and receiver of sound (*dashed curve*)

spectrum of the waveguide. This is caused by there being a minimum value of the backscattering coefficient for the Brillouin beams of the first mode. It is also noteworthy that the difference between I_{a}^{sc} and I^{sc} decreases as the distance to the scattering area increases, which is simply associated with a decrease in the number of modes contributing to the long-range reverberation. At long distances, only the modes with the lowest mode indices and low backscattering coefficients survive in the waveguide because of attenuation.

An exciting possibility for using vertical emitting and receiving arrays is connected with focusing the sound field in shallow water. It is common to use a time-reversal mirror (or phase conjugate array) for this purpose (Kuperman et al. 1998). To focus the sound field, we should use a vertical source–receiver array (or SRA, a chain with compound sound sources and receivers) and a supporting tonal sound source located at the point (r, z) where we want to focus the sound field (see Sect. 8.5). Using this equipment, we can obtain a focused sound field satisfying the following expression:

$$P(r,z) = \sum_{m}^{M} P_m^{in} \equiv \sum_{m}^{M} \hat{C}_m \frac{\Psi_m(z)}{\sqrt{\xi_m r}} \exp(i\xi_m r). \tag{6.27}$$

Here, in distinction to (10.8), the coefficient $\hat{C}_m$ is equal to

$$\hat{C}_m = \sqrt{N_s W_0} \exp(i\pi/4) \sum_{j=1}^{J} \sqrt{\rho_j c_j} P_{ss}^{*}(0, z_j) \Psi_m(z_j), \tag{6.28}$$

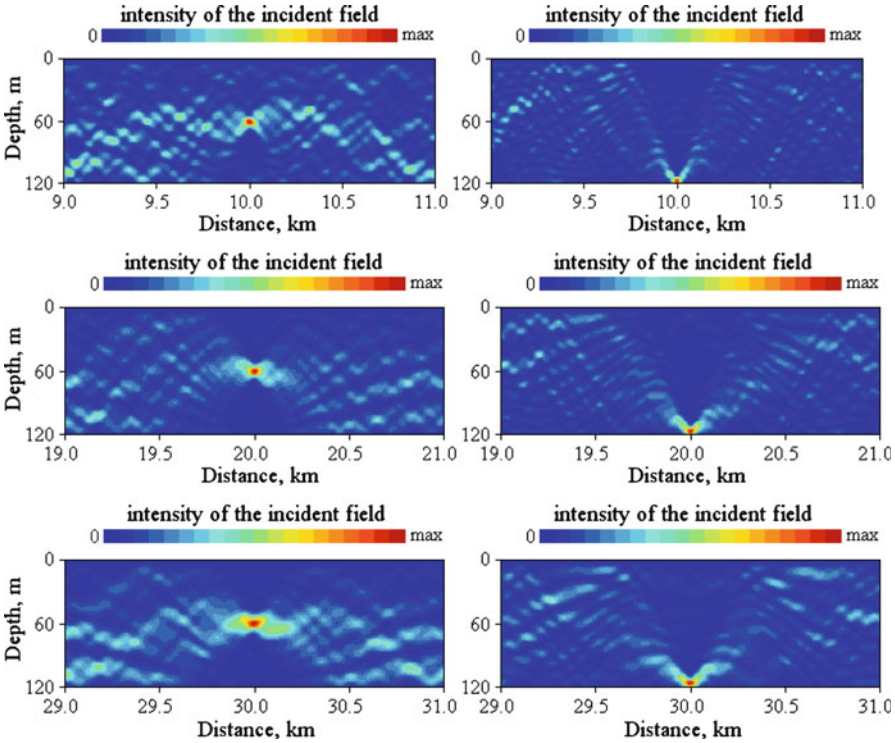


Fig. 6.10 Intensity of the focused sound field in the waveguide at different distances from the SRA and at different depths

where $P^*_{ss}(0, z_j)$ is the complex conjugate sound field of the supporting sound source at the points where the sound sources–receivers of the SRA are located and N_s is a normalization factor. If the full acoustic power radiated by the vertical array is equal to W_0, then N_s is given by

$$N_s = \left[\sum_{j=1}^{J} \left| P_{ss}(0, z_j) \right|^2 \right]^{-1}. \tag{6.29}$$

By way of example, the intensity of the sound field $I = P^2(r, z)/\rho c$ focused at different depths in the model waveguide discussed above in this section is plotted in Fig. 6.10. (For this waveguide, the vertical profile of sound speed is shown in Fig. 4.16.) Here, for the computer simulation, we used the SRA described above in this section and tuned it to focus the sound field [see (6.27) and (6.28)].

As indicated in the figures, we can focus the sound either field in the middle of the waveguide or on the sea bottom (or at any other spot). It is apparent that the

backscattered sound field varies in value for these two situations.[7] However, this difference can be detected only after some special processing of the scattered signal. Let us explain why. Ignoring waveguide dispersion, and similar to (6.14), we can write the following expression for the backscattered sound field at the reception point $(0, z_r)$ when we use a focused incident field (6.27):

$$P^{sc}(t) = \frac{g_{sc}}{2}\exp(i\pi/4)\sum_{m,\mu}^{M}\frac{\Psi_\mu(z_r)\Psi_\mu(z)F(\chi_m,\chi_\mu)\sqrt{dr_{m\mu}}P_m^{in}(r,z)}{\sqrt{\xi_\mu}}\exp(i\xi_\mu r), \tag{6.30}$$

where $t = 2r/c$, $z = H$, and the values $P_m^{in}(r,z)$ are determined by (6.27). Correspondingly, when we use a point receiver, we can calculate or measure the nonaveraged $I_{nonav}^{sc} = \left(P^{sc}(t)\right)^2/\rho_r c_r$ and the averaged $I_{av}^{sc} = \overline{\left(P^{sc}(t)\right)^2}/\rho_r c_r$ sound intensity of the scattered field when we use a focusing SRA. (Note that ρ_r and c_r are the density and sound speed at the receiving point, respectively). We obtain

$$I_{av}^{sc} = \frac{\overline{\left(P^{sc}(t)\right)^2}}{\rho_r c_r} = \frac{g_{sc}^2\tau c}{8r\rho_r c_r}\sum_{\mu}^{M}\frac{\left|\Psi_\mu(z_r)\right|^2\left|\Psi_\mu(H)\right|^2}{\left|\xi_\mu\right|}$$
$$\times\sum_{m}^{M}F^2(\chi_m,\chi_\mu)\frac{\left|\hat{C}_m\right|^2\left|\Psi_m(H)\right|^2}{\left|\xi_m\right|}\exp(-\gamma_m r - \gamma_\mu r). \tag{6.31}$$

Calculated r-dependences $I_{nonav}^{sc}(r)$ and $I_{av}^{sc}(r)$ are plotted in Fig. 6.11 for the same waveguide and for the same focusing points which are shown in Fig. 6.10 (In the calculation, $z_r = 120\,\text{m}$). As can be seen from Fig. 6.11, the effect of the backscattered field increasing when the sound field is focused on the sea bottom is not detected on both the averaged and nonaveraged measures of sound intensity. This rather unexpected result is explained by the interferences of the sound field in shallow water which mask the backscattered field increase. (See the blue line in Fig. 6.11.) On the contrary, if we do not take into account the sound interference (primarily the interferences of the incident sound field), we cannot perform focusing of the sound field at all. That is why we cannot see the effect of sound focusing in the red line in Fig. 6.11.

That we cannot see this effect in Fig. 6.11 does not mean that it is absent. It only means that we should carry out another spatial averaging of the backscattered sound field, and should not average the incident sound field. Specifically, we can see this effect if we average the sound intensity $\hat{I}_{nonav}^{sc}$ over depth (over all receivers of the SRA) and over a horizontal range with a spatial scale which is equal to the horizontal extent of the focal spot in the waveguide, Δr_f (also called the depth of field). (In an experiment, instead of using horizontal spatial averaging, we can use

[7] The low-frequency, long-range reverberation for a focused sound field was first studied in Kim et al. (2004).

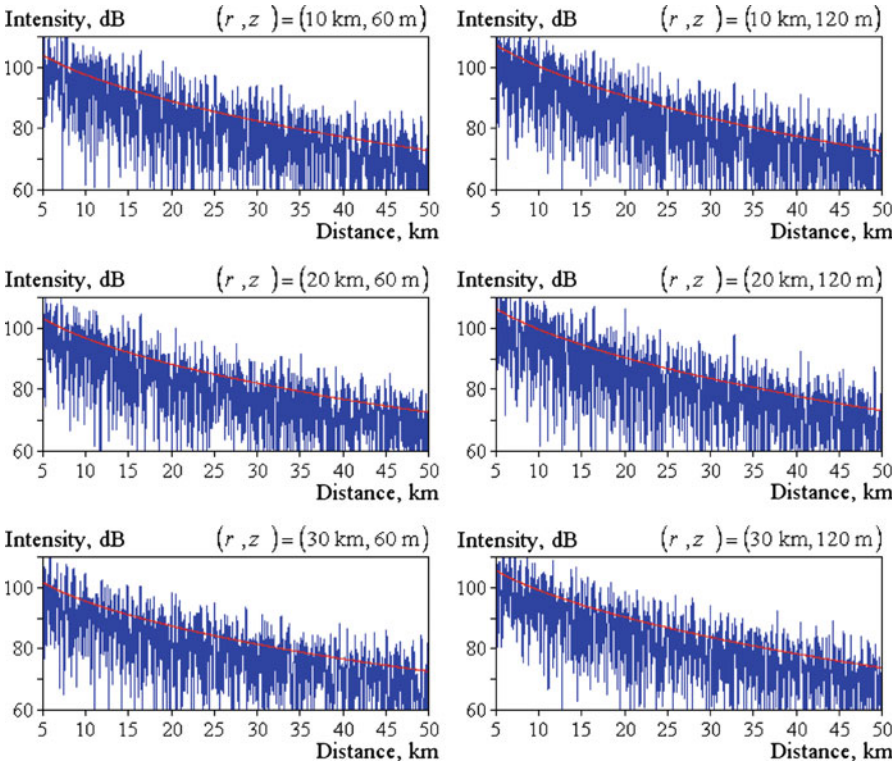


Fig. 6.11 Averaged (*red line*) and nonaveraged (*blue line*) intensity of the scattered field for an array focused sound field at the point with coordinates (r, z)

temporal averaging with a temporal scale $\Delta r_f/c$.) In other words, this effect can be observed in the r-dependence of $\hat{I}^{sc}_{av}$, which is equal to

$$\hat{I}^{sc}_{av}(r) = \frac{1}{I_r}\int_0^R \sum_{i_r}^{I_r} I^{sc}_{nonav}(r', z_{i_r}) D(r' - r) dr', \tag{6.32}$$

where

$$D(x) = \begin{cases} 1/\Delta r_f, & |x| \le \Delta r_f/2 \\ 0, & |x| > \Delta r_f/2 \end{cases}. \tag{6.33}$$

As an example, the calculated dependence of $\hat{I}^{sc}_{av}(r)$ for different focusing points is plotted in Fig. 6.12. Here, computer simulation was carried out for the same parameters as for the calculations in Fig. 6.11. (The value Δr_f was equal to 100 m.) As seen from the figure, sound field focusing on the sea bottom causes a noticeable increase in the reverberation level. Experimentally, such an increase was observed by Kim et al. (2004).

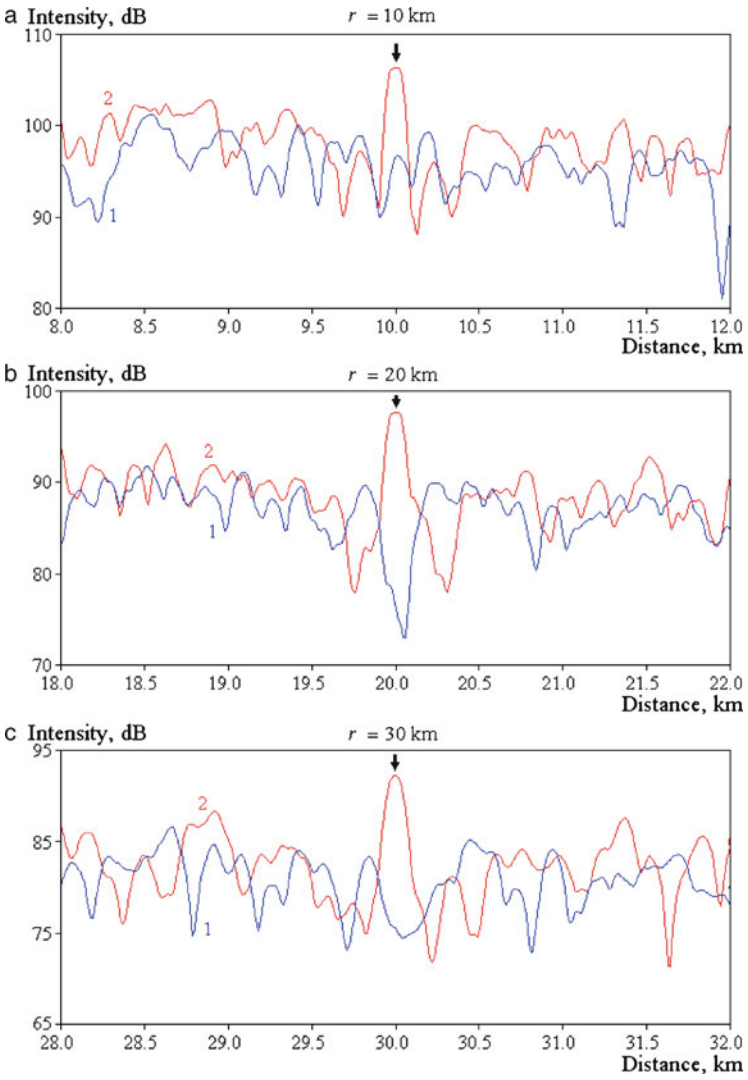


Fig. 6.12 Levels of reverberation signals $\hat{I}_{av}^{sc}(r)$ for a SRA focused at 10 km (**a**), 20 km (**b**), and 30 km (**c**) from the SRA. *Blue lines* and *red lines* correspond to a focal depth of 60 m and 120 m (sea bottom) respectively

6.5 Long-Range Reverberation in a Randomly Inhomogeneous Waveguide

The mode theory of long-range reverberation described in Sect. 6.2 is valid for the regular acoustic waveguide (a near-bottom sound channel) with a constant backscattering coefficient at the sea bottom. Let us now consider how to construct a similar

theory for the randomly inhomogeneous waveguide with a random coefficient m_b [see (6.1)]. Such a randomly inhomogeneous channel is characteristic of shallow water with a randomly inhomogeneous bottom and in the presence of pronounced internal waves. The principal difference of the theory for this case from our previous theory consists of considering mode interactions during the propagation of the forward and scattered acoustic fields, as well as the stochastic character of the sound reflection from the sea bottom. One of the basic results of the theory developed is that the simultaneous effects of waveguide dispersion and mode interaction not only increase the signal duration up to τ_1, but also change its mode structure. This is experimentally confirmed by the shape of a signal detected in an experiment in the Barents Sea (Fig. 6.13) (Katsnelson et al. 2004a). It is important to recall that the intermode dispersion and the peculiarities of waveguide mode attenuation in shallow water result in the fact that the effective signal duration is virtually independent of distance (see Chap. 4). The effective radiated power decreases by a factor of τ/τ_1 in this case. Thus, the ensonified bottom area is no longer partitioned into separate rings of radius $r_{m\mu}$ and width $dr_{m\mu}$ [see expressions (6.9) and (6.13)], but is represented by a single ring of width $\Delta r = c\tau_1/2$.

Thus, by analogy with (6.8), the component of the acoustic field incident on the scattering area for a solitary sound source and corresponding to the m-th mode in a waveguide with random inhomogeneities is given by

$$P_m^{in} \cong \sqrt{8\pi\rho_s c_s W_0} C_m(r, \vartheta, T) \frac{\Psi_m(z)}{\sqrt{q_m r}} \exp(iq_m r). \tag{6.34}$$

As before, $\Psi_m(z)$ and $\xi_m(r)$ are the eigenfunction and the propagation constant of the m-th mode for a waveguide with no inhomogeneities, and ϑ is the angle in the cylindrical coordinate system, defining the direction to the scattering ring on the bottom. The coefficients C_m are determined by solving the set of equations for coupled mode interaction (3.120) at the distance from the source ($r = 0$) to the bottom area (r). In this case, the initial conditions given for a point source are equal to

$$C_m(0, \vartheta, T) = \frac{\exp(i\pi/4)}{\sqrt{8\pi}} \Psi_m(z_s). \tag{6.35}$$

It should be noted that the coefficients $C_m(r, \vartheta, T)$ depend on the type and characteristics of the random inhomogeneities. They also remain almost unchanged over the time t of the reverberation signal propagating to the scattering area and back. In such cases, the model of a frozen medium is valid and we use the value T to denote slow temporal dependence of the coefficients $C_m(r, \vartheta, T)$.

We represent the bottom scattering surface as an ensemble of randomly distributed and independent point sources with a given distribution of "force" and size. Such a physical model of the scattering surface corresponds to sound back-scattering at both random undulations and random inhomogeneities of the upper sediment layer. In this case, the descriptor of the bottom scattering area [see relation (6.12)] takes the form

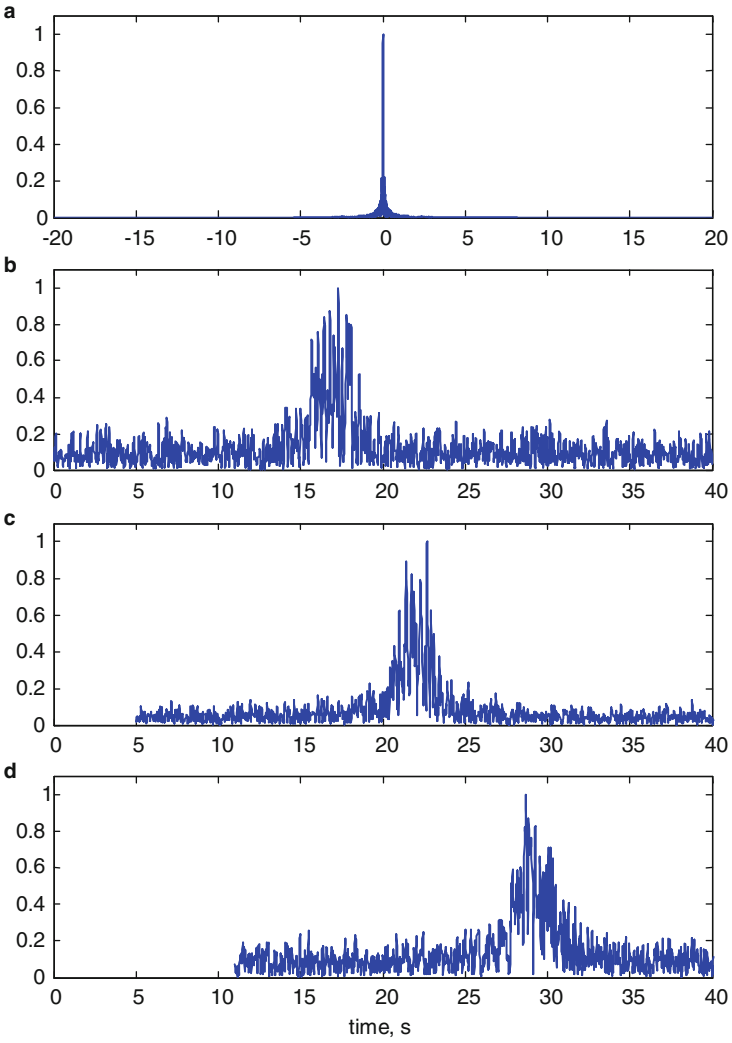


Fig. 6.13 Envelopes of signals at the correlation receiver output, normalized to maximum values at distances from the sound source of 1 m (**a**), 23.5 km (**b**), 31.1 km (**c**), and 39.6 km (**d**)

$$T_{m\mu}(r) = \varsigma_{m\mu}(r)F(\chi_m, \chi_\mu)\sqrt{dA}, \tag{6.36}$$

where $\varsigma_{\mu m}(r)$ is a stochastic process describing the relative fluctuations of the field scattering from the m-th mode to the μ-th mode, and dA is the elementary area of the ensonified surface. The statistical characteristics of the scattering matrix are symmetric with respect to permutation of the indices $T_{\mu m} = T_{m\mu}$ (Young-Chung Cho 1980).

As a consequence of the assumptions of the uniformity and isotropy of the random bottom inhomogeneities, the field $\varsigma_{m\mu}(r)$ is uniform and isotropic. This field is

characterized by the independent scattering of the acoustic waves by the various bottom areas. Therefore, it can be written as

$$\varsigma_{m\mu}(r) = \hat{\vartheta}_{m\mu}(r)\exp\left[i\widehat{\phi}_{m\mu}(r)\right], \tag{6.37}$$

where $\hat{\vartheta}_{m\mu}$ and $\widehat{\phi}_{m\mu}$ are statistically independent, the phase $\widehat{\phi}_{m\mu}$ is uniformly distributed in the interval $(0, 2\pi)$ (the probability density is $\hat{w}(\widehat{\phi}_{m\mu}) = 1/2\pi$), and the amplitude $\hat{\vartheta}_{m\mu}$ is Rayleigh distributed, i.e.,

$$\hat{w}\left(\hat{\vartheta}_{m\mu}\right) = \frac{\hat{\vartheta}_{m\mu}}{\hat{\sigma}^2_{m\mu}}\exp\left(-\frac{\hat{\vartheta}^2_{m\mu}}{2\hat{\sigma}^2_{m\mu}}\right), \tag{6.38}$$

where $\left\langle\hat{\vartheta}_{m\mu}\right\rangle = \sqrt{\pi/2}\hat{\sigma}_{m\mu}$ and $\left\langle\hat{\vartheta}^2_{m\mu}\right\rangle = 2\hat{\sigma}^2_{m\mu}$. Hereafter, we consider (under the assumption of our model) that the random field $\varsigma_{m\mu}(r)$ is statistically independent of the mode indices; that is, $\hat{\sigma}_{m\mu} = \hat{\sigma}$. We note that such an assumption is valid for sound scattering obeying the Lambert law and $g_{sc} = \sqrt{2}\hat{\sigma}$.

The elementary ensonified area is taken as a circular ring sector with inner radius $r_{\breve{\beta}}$, width r_c, and angular size $\Delta\vartheta$ ($\Delta\vartheta \gg r_c/r_{\breve{\beta}}$). The width of the entire ensonified area is $dr = c\tau_1/2$. This area is a ring of radius r and consists of many elementary areas so that $r \le r_{\breve{\beta}} \le r + dr$. We also assume that r_c is much smaller than the scale of the interference beating of the acoustic field so that the incident and scattered fields are constant within the elementary area. It is also reasonable that $r_c \ll dr$. Let us introduce the following designations: $\breve{\alpha}$ is the angular sector index; $\breve{\alpha} = 1, 2, ..., \breve{A}$, $\breve{A}$ is the number of sectors; $\breve{A} = 2\pi/\Delta\vartheta$; and $\breve{\beta}$ is the elementary ring index, $\breve{\beta} = 1, 2, ..., \breve{B}$, where $\breve{B} = dr/r_c$. Then the number of elementary areas is $\breve{A} * \breve{B}$, and the scattered field at the reception point is determined by summing the scattered fields from all the elementary statistically independent areas[8]:

$$P^{sc}(t,T) \cong \sum_{\breve{\beta}=1}^{\breve{B}}\sum_{\breve{\alpha}=1}^{\breve{A}}\sum_{\mu=1}^{M} C_\mu\left(0, \vartheta_{\breve{\alpha}}, T\right)\frac{\Psi_\mu(z_r)}{\sqrt{q_\mu(0)r_{\breve{\beta}}}}\exp(iq_\mu r_{\breve{\beta}}), \tag{6.39}$$

where $t = 2r/c$.

The complex amplitude $C_\mu(0, \vartheta_{\breve{\alpha}}, T)$ of the μ-th mode is determined by solving the mode coupling equations (3.120) at the distance $r = r_{\breve{\beta}}$ (from the scattering area to the source at $r = 0$) with the initial condition

$$C_\mu\left(r_{\breve{\beta}}, \vartheta_{\breve{\alpha}}, T\right) = \frac{\exp(\mathrm{i}\pi/4)}{\sqrt{8\pi}}\Psi_\mu\left(r_{\breve{\beta}}, z\right)\sum_{m=1}^{M} T_{\mu m}\left(r_{\breve{\beta}}, \vartheta_{\breve{\alpha}}\right)p^i_m\left(r_{\breve{\beta}}, \vartheta_{\breve{\alpha}}, T\right). \tag{6.40}$$

[8] We emphasize that P^{sc} depends on time T as a parameter.

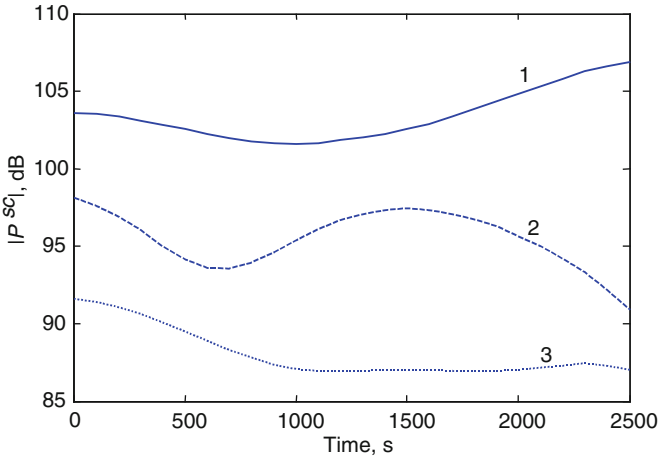


Fig. 6.14 T-dependence of the magnitude of the scattered field amplitude at various distances to the scattering bottom area: $r = 10$ (1), 20 (2), and 30 km (3)

Relation (6.61) makes it possible to estimate the size and temporal variability of long-range reverberation signals in a randomly inhomogeneous waveguide. In contrast to expression (6.14) accounting for only the rapid temporal variations of the reverberation signals caused by the acoustic field interference in the waveguide, relation (6.39) makes it possible to carry out the following:

1. Estimate slow signal variations associated with changes in the medium characteristics.
2. Describe the variability of scattered signals caused by random bottom inhomogeneities. In practice, such variability is observed during acoustic soundings from moving sound emitters and receivers. This is due to the fact that long-range reverberation is caused by various sea bottom areas sensing signals emitted at various instants of time.
3. Estimate slow signal variations associated, in particular, with internal waves. Nowadays, one can simulate random pycnocline vibrations $\varsigma(r, z, \vartheta, T)$ in a large area in shallow water. An example of such a simulation based on the field of background IWs on the continental shelf is described in Chap. 5.

Examples of calculations showing the slow time variations in the amplitude of the scattered acoustic field in the presence of background IWs (simulated using the technique described in Chap. 2) are given in Fig. 6.14. The calculations were carried out for several fixed distances to the scattering area specifically, $r = 10$, 20, and 30 km. It was assumed that the sound source and receiver are colocated and are placed on the sea bottom. The sound velocity profile taken for these calculations is shown in Fig. 4.16. The other parameters of the waveguide and the acoustic signals correspond to our recently discussed model waveguide. The effective signal duration was $\tau_1 = 3$ s (see Fig. 6.13). The width of the elementary

circular ring was assumed to be $r_c = 10$ m. Thus, the value of $\breve{B}$ was 220, and the value of $\breve{A}$ was taken to be equal to 20. The function $F(\chi^{in}, \chi^{sc})$ was calculated using relation (6.7) (the Lambert law). The value of $\hat{\sigma}$ was taken to be equal to 15. The time spectrum $G^t(\Omega, z)$ was calculated on the basis of the data of Kozubskaya et al. (1999), and corresponded to the spectrum of background IWs for an area of the Barents Sea. Here, we assume that the internal wave field is isotropic and that $\Theta(\vartheta) = 1$. For calculation of the coefficients $C_{m,\mu}$, we used (3.106 and 3.115) and the forms (5.21 and 5.22) as well. As may be seen from Fig. 6.14, the scattered acoustic field exhibits significant fluctuations with time in the presence of IWs.

Examples of calculations of the scattered signal amplitude at the reception point for various statistical samples of the bottom are shown in Fig. 6.15. The calculations were carried out using relation (6.39). To separate the effects associated with the stochastic character of scattering, the calculations were carried out in the absence of IWs, that is, for an unchanging sound velocity profile (see Fig. 4.16). In all other respects, the numerical experiment was carried out using the same parameters as were indicated in this section. Figure 6.15a–c shows that the amplitudes of the scattered signals exhibit significant fluctuations as the scattering area properties vary. For this reason, and mainly due to the interference of the waveguide modes, the value $|P^{sc}|$ varies significantly with time t, or, what is the same [see (6.9)], with the distance to the scattering area. Figure 6.15a–c also show that rather stable interference beats are observed in the scattered signal in the case of a relatively uniform distribution of the backscattering coefficient and long-range reverberation. This is caused by a relatively small number of energy-carrying modes at long distances from a sound source. We note that such beats will occur when observing long-range reverberation of any type: bottom, bulk, and surface with a relatively uniform spatial distribution of the sound backscattering coefficient. The validity of this statement follows from experimental data. In particular, interference beats of the scattered signal, slowly varying in time, were observed for several days in the experiments described in Weston and Hocking (1990) (see Fig. 6.16). In this case, the distance to the scattering area was several tens of kilometers at a sea depth of 40–70 m and with a probing pulse frequency of 1 kHz. The temporal variations of the interference beats were probably associated with tides.

We once again emphasize that the characteristics of low-frequency bottom reverberation on the sea shelf vary widely, primarily caused by the dependence of the backscattering coefficient on the bottom relief and the properties of the upper layer of the bottom sediments. Due to technical problems, no direct measurements of the features of the coefficient m_b have been carried out at low frequencies ($\leq$500 Hz) and at small grazing angles ($\leq 20°$). Extrapolation of the available experimental data to larger frequencies and grazing angles allows the following conclusions (1) The backscattering coefficient weakly depends on frequency and (2) the angular dependence of this coefficient obeys the empirical Lommel–Seeliger law as a rule. These conclusions are also confirmed by calculational models of sound backscattering by the sea bottom.

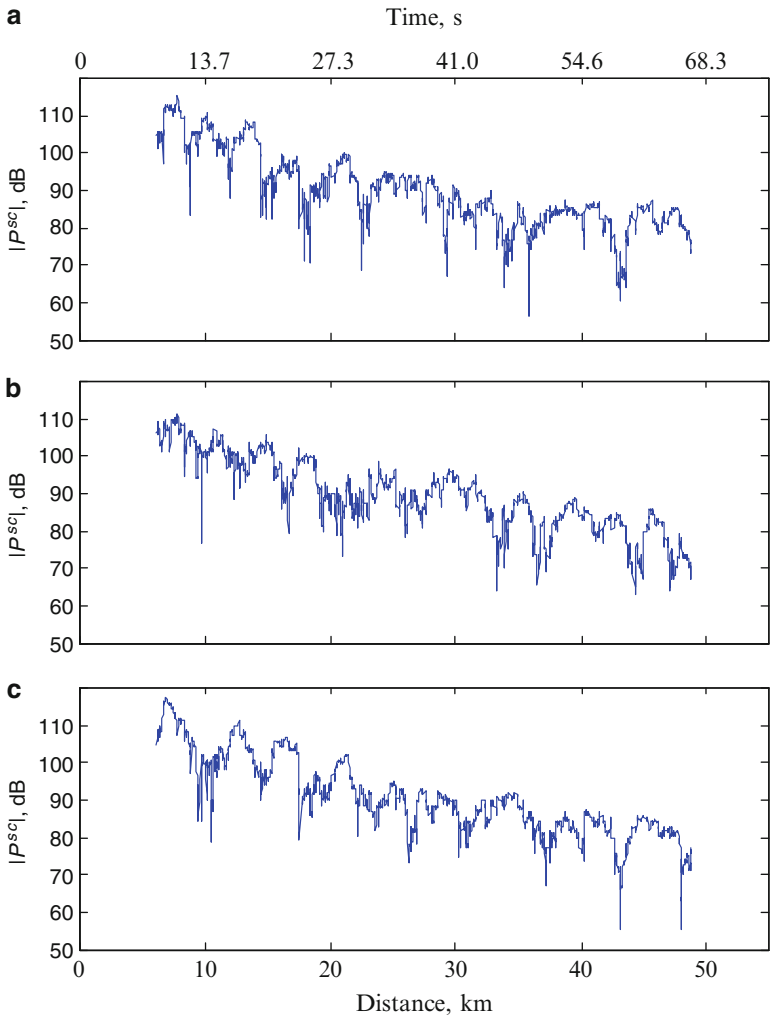


Fig. 6.15 r–t-dependence of the magnitude of the scattered field amplitude for various random realizations (a,b,c) of the scattering area

At the same time, numerical experiments simulating long-range, low-frequency reverberation show that the average intensity of the scattered acoustic signals at the reception point depends weakly on both the sound velocity profile in shallow water and the angular dependence of the backscattering coefficient. This intensity significantly decreases in the case of extended arrays tuned to the radiation and reception of acoustic signals corresponding to the lowest waveguide modes.

It should be noted as well that we can reduce long-range reverberation in shallow water significantly by focusing the sound field at some appreciable distance from the sea bottom.

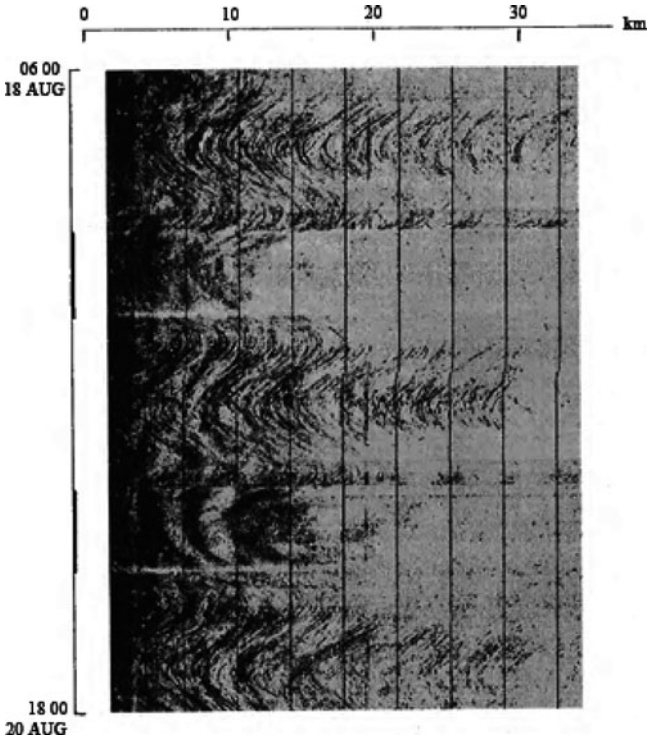


Fig. 6.16 Brightness form of the sonar record including three types of diffuse interference pattern in shallow-water reverberation, 18–20 Aug 1963, times BST (GMT+1)

It is also important to note that in a randomly inhomogeneous waveguide that is characteristic of the continental shelf, long-range reverberation signals can exhibit significant variations with time as the source and receiver positions vary, even for a statistically uniform sea bottom.

Chapter 7
The Inverse Problem

7.1 General Considerations: The Linear Inverse Problem

In considering the propagation of sound through fluid or elastic media (or light through optical media, or heat through materials, etc.), it is usual to first address the so-called "forward problem." Specifically, one calculates the sound (light, heat) field assuming knowledge of both the medium properties and of the physics governing the radiation, propagation, diffusion, attenuation, etc. of the corresponding waves.

However, while we know the basic wave equations for shallow-water acoustics rather well (though their variants for scattering processes might be less straightforward), actually knowing the medium properties in the shallow ocean and seabed is another issue entirely. The ocean and seabed vary significantly spatially in shallow-water regions, and the water column varies significantly in time as well. (And in fact, even the top of the seabed can vary significantly in shallow water, as seen by looking at seabed roughness changes before, during, and after the passage of coastal storms.) Thus in doing even the simple forward problem to generate acoustic fields, we encounter some real difficulty. The easy way to address medium complexity for shallow-water acoustics over the years has been to use a simple "average model" of the medium, which includes the salient features, but ignores or smoothes over fine detail. If a more sophisticated answer is desired, then Monte Carlo computer calculations, using an appropriate set of possible realizations of the medium (more detailed variations about the average profile), are often employed. Due to improved computational power, this type of statistical approach is routine nowadays, and probability density function (pdf) representations of the medium properties and Bayesian approaches are becoming more popular. Incorporation of medium complexity/variability via moment propagating equations has also been pursued, though not as commonly as the other approaches.

Consideration of the forward problem also informs us of some of the issues that will be encountered in the inverse problem (1) should we solve for the "average medium" using a deterministic approach (where the medium variability about the

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_7,
© Springer Science+Business Media, LLC 2012

average gets absorbed as "noise" or error), or (2) should we solve for the medium statistical properties, such as searching for spectra, pdf's, moments, EOF coefficients, or other such entities.

In the coming sections, we will primarily discuss the deterministic inversion of low-frequency acoustic propagation data for *average medium* properties, and consign the variability about it to noise/error. In doing this inverse problem, we will also limit ourselves mostly to *linear* inversion techniques. We do this for two good reasons (1) the linear techniques most clearly show the physics of the inverse via the kernel function (see below) and (2) the linear techniques most easily provide error estimates via the resolution and variance. That being said, linearization of a problem can also be inappropriate for some cases, and nonlinear techniques are routinely used nowadays (e.g., simulated annealing or genetic algorithms [Special Issue of IEEE]). We also note that we will not be delving much into the intricacies of inverse theory, which is a vast topic on its own. Rather, we will just recommend the reader to some standard books on the topic, which include (Menke 1989, Wunsch 1996, Parker 1994, Tarantola 2005, Aster et al. 2005). Journal references will also be provided in the sections that follow. Shallow-water inverse techniques that we do not explicitly describe in this portion of the text (e.g., Bayesian, nonlinear) will be discussed in Chap. 11, where we look to current and future trends in shallow-water acoustics.

We now proceed directly to the (linear) inverse problem. In general, we are interested in how specific data, d, are related to a specific model, m, via some function G (the operator/kernel) which contains the physics in it. Mathematically we write the simple form

$$G(m) = d. \tag{7.1}$$

For our purposes, d will often be the acoustic field (or an aspect of it) recorded at some receiver or array of receivers, m will be the ocean and/or seabed properties we wish to retrieve, and G will be the (assumed known) propagation or scattering physics equation(s) that connect(s) the two. In the case of linear systems, we can cast (7.1) as an operator (matrix) equation

$$\mathbf{Gm} = \mathbf{d} \tag{7.2}$$

where the data $\mathbf{d}$ are in an $(n \times 1)$ vector, the model $\mathbf{m}$ is in an $(m \times 1)$ vector, and the kernel $\mathbf{G}$ is an $(n \times m)$ matrix. How we obtain these vector and matrix quantities for some useful shallow-water acoustics cases will soon be discussed. But, first we discuss some example solutions to (7.2) and the errors associated with them.

As a "typical" shallow-water inverse problem, we will use an example from normal mode theory in which a perturbation to the sound speed, either in the water column or the bottom, causes a perturbation to the acoustic modal eigenvalues. The perturbation theory used to derive this example was shown previously, so we will just use the result to display the technique.

The equation we refer to is the first order perturbation theory integral (3.56a) which relates a sound speed perturbation $\delta c(z)$ from some (background) reference profile $c^0(z)$ to modal eigenvalue perturbations $\delta\xi_l$ (more exactly, we will deal with the perturbation of the real part, δq_l). Specifically,

$$\delta q_l = -\frac{1}{q_l^0}\int_0^\infty \frac{\rho_s}{\rho^0(z)}\left[\Psi_l^0(z)\right]^2[k^0(z)]^2\frac{\delta c(z)}{c^0(z)}dz \tag{7.3}$$

where q_l^0 is the real part of the l-th eigenvalue for the assumed background geoacoustic model, $\delta q_l = q_l - q_l^0$ is the difference between the measured eigenvalue q_l and the background model eigenvalue, $\rho^0(z)$ is the density profile for the background geoacoustic model, $\Psi_l^0(z)$ is the normalized mode function for the lth mode of the background model, and $k^0(z)$ is $\omega/c^0(z)$, where ω is the angular frequency. Equation (7.3) is a Fredholm integral equation of the first kind, a standard mathematical form in inverse theory. It relates data we measure (the δq_l) to the model we seek (the $\delta c(z)$) via the integral kernel (the rest of the r.h.s. of (7.3) beside the $\delta c(z)$). To convert (7.3) to the form of (7.2) is quite simple. One merely discretizes the integral in (7.3) via some numerical quadrature scheme, such as Simpsons Rule, which turns the integral into a sum over m pieces (or levels in z_j) for each mode l. Thus we have a matrix for $\mathbf{G}$ and vectors for $\delta\mathbf{q}$ with components δq_l and $\delta\mathbf{c}$ with components $\delta c(z_j)$.

This simple relation, $\delta\mathbf{q} = \mathbf{G}\delta\mathbf{c}$, showing how a change in sound speed relates to a change in modal eigenvalues is, at this point in time, almost a "classical" example of the properties of linear inversions in shallow water. It bears some examination of its terms.

We begin with examining the δq_l. The main thing to note here is that the number of modes that we generally have for data is limited in number, on the order of ten or less. This is due to there being few modes at low frequencies, and due to the inherent difficulty in filtering out (resolving) more than a few modes as the frequency increases. One also might naively think that each frequency one uses provides an entirely new data set, as the modal inverse considered looks at individual frequencies. And while this is mathematically correct, it is misleading. The modal dispersion relation $q_l = q_l(\omega)$ strongly relates the values of the mode eigenvalues at neighboring frequencies, so that the data sets $q_l(\omega)$ and $q_l(\omega + \Delta\omega)$ are nearly (but not quite) linearly dependent for small $\Delta\omega$. Thus, the information content in the two different frequencies is nearly the same, and one has to look at more widely separated frequencies to get significantly different information.

The next piece of the simple inverse equation is the kernel. While this would seem would seem to be the most mundane piece of the puzzle, it still has some inherent interest. The most basic question one can ask of the kernel is: Is it complete and correct? In our case, we are looking at a modal eigenvalue perturbation due to a sound speed perturbation, but we are also tacitly ignoring the effects of density, attenuation, and shear. To the extent that these quantities affect the real part of q_l for

the waveguide we are looking at, i.e., have effects that look like sound speed effects, we are "truncating" the problem in the kernel, i.e., leaving out some of the physics. Whether or not this is an acceptable truncation is part of the "art form" in constructing the kernel. As to the issue of the "correctness" of the kernel, we merely refer to the degree of approximation used in the wave equation (e.g., modal vs. ray or parabolic equation approaches) and also the degree of linearization required to obtain a perturbative form. Both give errors in the kernel which may or may not be acceptable.

Finally, we come to the issue of the model, which just may be the thorniest (and thus the most interesting) part of doing inverse theory in shallow-water acoustics. In Tarantola's "Inverse Problem Theory" book, this problem is called the "parameterization of the system" which is very nicely described as "discovery of a minimal set of model parameters whose values completely characterize the system (from a given point of view)." How one parameterizes the ocean and seabed variability is quite flexible, but there is an underlying theme to it. Specifically, we can construct very simple models with 1–2 parameters to describe a system (e.g., a seabed sediment model that has only the sound speed at the water sediment interface and the gradient of the sound speed in the bottom as parameters), but this is only done by incorporating our a priori knowledge/prejudices and at the serious risk of incurring model errors by oversimplification. The advantage gained is that the problem becomes even-determined or over-determined. On the other hand, we can assume almost no a priori information, and try to solve for a much larger model space. This makes the model more flexible, but at the cost of the system of equations being highly underdetermined. The former case is rather simple to deal with, using standard least squares approaches, so we will not pursue it further. Let us look at the (more intricate) latter case next.

In the perturbation equation example we are examining, we are trying to determine a piecewise continuous function $\delta c(z)$ from zero (the air–sea interface) to infinity (realistically, deep into the seabed). This is an infinite amount of information we are requesting, and while discretizing the integral lessens the dimension of the model space, it does so by compromising resolution, so that we have not really obtained any "free" advantage. Thus we see that the modal inverse is using about ten data points to determine a very large number of unknowns, and so our problem is highly underdetermined. This is the case for a simple horizontally stratified medium (a 1D problem). If we consider a 2D range-dependent medium, we basically square the indeterminacy. For 3D, we can often (though not always) use $N \times$ 2D slices, so this often limits the spatial indeterminacy to 2D at most. There is also the issue of time variability – however, if we consider each inverse to be a time snapshot (usually appropriate for media which can be considered as "frozen" for acoustic propagation between source and receiver), this issue can be avoided for now.

Underdetermined problems are common in science, and the standard way of dealing with them is to add "reasonable constraints" to the problem which reduce the indeterminacy and thus provide a unique solution. (It is well known from linear algebra that underdetermined systems can allow an infinity of solutions, which is useless. One good, physically sane, and hopefully correct solution is all that we desire!) Let us look at these constraints next.

To begin with, we want a solution that will produce a best fit to the scant few data points that we have, in the least squares sense. Mathematically, this just corresponds to minimizing the L_2 norm fitting error $\|\mathbf{K}\|$. This value can be defined in different ways, for example as a norm of mismatch:

$$\|\mathbf{K}\| = \|\mathbf{G}(\mathbf{m}) - \mathbf{d}\|.$$

Further we can consider different forms of the norm (using the mismatch between modeled and measured data) in finding a vector $\hat{m}$ providing the best fit to the data.

This constraint on the problem is a data–space requirement, which still leaves a lot of freedom in the space of the model, $\hat{m}$. In the spirit of perturbation theory, we would also expect the deviation of the model from the background to be small. Mathematically, this suggests that we minimize the norm of the model. With these two constraints, in both the data and model space, we can obtain a unique solution to the problem that has a "common sense" flavor to it – but we must still admit that we have no assurance that the solution matches physical reality.

In addition to the minimum norm constraint mentioned above, we can also impose a variety of other constraints on the model. For instance, we can ask for the smoothest model or flattest model or spikiest model by multiplying the model vector by a matrix that incorporates constraints on the model's derivatives (Menke 1989). We can also impose a priori information such as the exact value of the sound speed at some point (due to thermistor or core sample measurements, perhaps) via Lagrange multipliers. The topic of constraining inverses is again a large one, and so we will again refer the reader to the literature for more information.

Another question we would like to address while looking at the components of the linear inverse equation is: What is the data "illumination" of the model space vs. frequency, mode number, and depth? In more simple terms, we would like to know what frequencies to use to best see various ocean and seabed features. Our question can be phrased mathematically: How much information about $\delta c(z_0)$ does a given set of $\delta q_l(\omega_0)$ provide? To answer this, we are guided by the perturbation relation

$$\delta q_l(\omega_0) = \int G_l(z, \omega_0)\delta c(z)dz \tag{7.4}$$

where

$$G_l(z, \omega_0) = -\frac{\rho_s}{q_l^0 \rho^0(z) c_0(z)} \left[\Psi_l^0(z)\right]^2 [k^0(z)]^2.$$

Let us consider how a unit change in a model point, $\delta c(z_0) = L\delta c_0 \delta(z - z_0)$, (where L is constant of dimension of length) changes the data, the modal eigenvalues. From the sifting property of the delta function, we get

$$\delta q(\omega_0) = LG(z_0, \omega_0)\delta c_0 \tag{7.5}$$

or that the nth eigenvalue perturbation at depth z_0 is simply proportional to the kernel at depth z_0, for a given frequency ω_0. Since the kernel in this case is proportional to the square of the mode functions, we see that the modal structure of the acoustic field determines what is seen for the inverse, a physical result we can easily intuit based on simple ensonification considerations.

We note that this simple replacement of a model point with a delta function will give a similar result for all linear inverses, which implies that a study of the kernel can be useful in determining what we can see and cannot see experimentally.

7.2 Bottom Versus Water Column Influences on the δq_l and Inverse

In our perturbation integral, (7.3), there are contributions from both the water column and the seabed, i.e., from 0 to H (water column) and from H to ∞ (seabed). This creates a real practical difficulty in doing the inverse problem, which is usually posed as: "given that we know the sound speed in the water column perfectly (from CTD's, XBT's, or other simple measurements), infer the bottom sound speed profile" or "given that we know the bottom properties well (from cores, grab samples, etc.), infer the sound speed perturbations in the water column due to ocean features." Very rarely (if ever) do we try to jointly estimate ocean and seabed properties – that is generally just too much to ask from a limited data set. It is also more usual to ask for the seabed properties from a shallow-water modal inverse, though low-frequency shallow-water acoustic tomography for the water column perhaps has some potential, and is being explored by various investigators.

One difficulty commonly encountered is that we really do not know the water column perfectly when trying to invert for seabed properties, and vice versa. For example, there have been a number of "Hankel transform experiments" performed to date (e.g., Frisk and Lynch 1984) in which investigators have measured the δq_l at a few frequencies, and then used these data to invert for seabed properties. In doing these experiments, the local water column sound speeds were measured by CTD's, XBT's, thermistor chain strings, etc., so that one would think that the ocean sound speed *was* reasonably constrained. However, vertical time series profiles of the ocean sound speed at a few spatial points are very often not adequate, in that internal waves, fronts, eddies, etc. can change the shallow-water sound speed profile drastically over horizontal distances of hundreds of meters, typically shorter than the horizontal extent of the above noted experiments. Moreover, as was first shown by Frisk (2005) and will be further discussed below, the water column variability can often be the dominant contribution to the modal eigenvalue perturbation measured, and so errors/uncertainties in the water column sound speed have to be minimized if one is to perform a good inverse for the bottom sound speed. If the errors in the water column sound speed are not accounted for, they can then cause significant errors in the bottom estimate, as we will also show (Lin et al. 2006).

Let us consider the relative contributions to corrections of the eigenvalues due to perturbations of the sound speed profile in the water layer and in the bottom. We suppose for now that the variation of the sound speed profile $\delta c(z)$, which generally speaking can be complex due to absorption, is real. In first order of perturbation theory, this correction is given by (7.3) as for the unperturbed (reference values) of the longitudinal wave numbers we take the real values q_l^0, obtained from solution to the Sturm–Liouville problem (Frisk 1994). In this case

$$\delta q_l = -\frac{1}{q_l^0}\int_0^\infty \frac{\rho_s}{\rho(z)}\left[\Psi_l^0(z)\right]^2 k^2(z)\frac{\delta c(z)}{c(z)}dz. \tag{7.6}$$

The integral on the right side of the equation can be split into two parts – one part within the water column and one part extending through the bottom. In the framework of the shallow-water model of Sect. 3.1, where we take the density in the water and bottom to be a constant, and assuming a constant sound speed in the bottom, we get

$$\delta q_l = -\frac{1}{q_l^0}\left\{\int_0^H \left[\Psi_l^0(z)\right]^2 k^2(z)\frac{\delta c(z)}{c(z)}dz + \frac{\Psi_l^0(H)k_1^2}{c_1 m_1}\int_H^\infty \mathrm{e}^{-2\kappa_l^0(z-H)}\delta c(z)dz\right\}. \tag{7.7}$$

The relative contribution of these corrections is the ratio of the first and second terms in parentheses.

The influence of the bottom layers decreases exponentially with depth, so for the second integral only a narrow sediment layer plays much of a role. For estimation of the aforementioned ratio, we suppose that the perturbations to the sound speed profile in the bottom and in the water column are constant, δc_w and δc_b. In this case the second integral can be easily calculated and we have

$$\delta q_l = -\frac{1}{q_l^0}\left\{\int_0^H \left[\Psi_l^0(z)\right]^2 \frac{k^2(z)}{c(z)}dz\delta c_\mathrm{w} + \frac{\left[\Psi_l^0(H)\right]^2 k_1^2}{2\kappa_l^0 c_1 m_1}\delta c_\mathrm{b}\right\}. \tag{7.8}$$

Proceeding further, we can use the simplification that the sound speed and correspondingly wave numbers within water layer change smoothly in comparison with the eigenfunction, so we can take these outside of the integral:

$$\delta q_l \approx -\frac{1}{q_l^0}\left\{\frac{k^2}{c}\int_0^H \left[\Psi_l^0(z)\right]^2 dz\delta c_\mathrm{w} + \frac{\left[\Psi_l^0(H)\right]^2 k_1^2}{2\kappa_l^0 c_1 m_1}\delta c_\mathrm{b}\right\}. \tag{7.9}$$

Using the normalization condition, we obtain

$$\int_0^H \left[\Psi_l^0(z)\right]^2 dz = 1 - \frac{\left[\Psi_l^0(H)\right]^2}{2\kappa_l^0 m_1}. \tag{7.10}$$

For typical conditions, the second term in this expression is much less that the first one. If we now consider the ratio of corrections, we have

$$\frac{\delta q_l^{\mathrm{b}}}{\delta q_l^{\mathrm{w}}} \sim \frac{\left[\Psi_l^0(H)\right]^2 k_1^2 c}{2\kappa_l^0 c_1 m_1 k^2} \frac{\delta c_{\mathrm{b}}}{\delta c_{\mathrm{w}}}. \tag{7.11}$$

For the Pekeris model in particular

$$\Psi_l \leq \sqrt{2/H}, \quad \kappa_l \sim \sqrt{k^2 - k_1^2}.$$

For typical waveguide parameters of $H \sim$ 50–150 m, $f \sim$ 100–200 Hz, $m_1 \sim 2$, $c = 1{,}500$ m/s, $c_1 \sim$ 1,700–2,000 m/s, we have

$$\frac{\delta q_l^{\mathrm{b}}}{\delta q_l^{\mathrm{w}}} \sim 2 \times 10^{-2} \text{ to } 2 \times 10^{-3} \frac{\delta c_{\mathrm{b}}}{\delta c_{\mathrm{w}}}. \tag{7.12}$$

We see that a correction due to a perturbation of the sound speed in the water layer is greater than the same in the bottom layer by two to three orders of magnitude. This implies that, in doing inverses for the bottom, one must have good knowledge of the water column, as we discuss next.

Recent work (Lin et al. 2006) has shown how unaccounted for water column error can be transferred into bottom property inverse estimates. We will paraphrase part of his recent paper on this topic in this section.

We slightly retool our formal development from before by looking at modal eigenvalue perturbation data, but now with the range included as a parameter. We also will use $q_l(\omega, r)$ to denote our data in this case, to differentiate it from the arguments in the last section, which treated a less general (range independent) waveguide. The linear perturbation integral for the modal eigenvalue data is then expressed as

$$\delta q_l(\omega, r) = \int_0^D G_l(\omega, z, r)\delta c(z, r) dz, \tag{7.13}$$

where $\delta q_l(\omega, r) = q_l(\omega, r) - q_l^0(\omega, r)$ is the small change of the lth acoustic modal eigenvalue due to the perturbation of sound speed c. D is the depth of the bottom boundary condition (and *not* the water column depth – rather, the "basement" depth), and $G_l(\omega, z, r)$ is the integrand kernel for the ℓth modal eigenvalue that will be separated into water and bottom portions. As we saw previously, the integral kernel only depends on the background (unperturbed) model c^0. In what Lin calls "the

equivalent transform method," the background model c^0 is the environment model contaminated with the water column mismatch Δc_{W}, i.e., the true environment is $c = c^0 + \Delta c_{\mathrm{W}}$. Thus the errors in the acoustic data due to unsuspected water column mismatch can be determined at a local range r from the perturbation integral

$$\delta q_l(\omega, r) = \int_0^{h(r)} G_{l_{\mathrm{W}}}(\omega, z, r)\delta c_{\mathrm{W}}(z, r)dz, \tag{7.14}$$

where $h(r)$ is the local water depth and G_l is the water column portion of the integral kernel. In order to eliminate the unsuspected data errors $\Delta\xi_l$ (i.e., treat our water column sound speed data as if it were correct), a term $\delta c_{\mathrm{B}}{}^*(z, r)$ must be added into the sound speed in the bottom, which is just the geoacoustic inversion error transferred from the water column mismatch $\delta c_{\mathrm{W}}(z, r)$. Hence the induced error term appears as the following:

$$\delta q_l(\omega, r) = \int_{h(r)}^{D_{\max}} G_{l_{\mathrm{B}}}(\omega, z, r)\delta c_{\mathrm{B}}{}^*(z, r)dz, \tag{7.15}$$

where $G_{l\mathrm{B}}$ is the bottom portion of the integrand kernel, and $D_{\max}$ indicates the maximum depth penetrated by the modal energy over the frequency range of interest. Equating the r.h.s. terms of (7.2) and (7.3) leads to a useful result for the error transformation:

$$\int_{h(r)}^{D_{\max}} G_{l_{\mathrm{B}}}(\omega, z, r)\delta c_{\mathrm{B}}{}^*(z, r)dz = \int_0^{h(r)} G_{l_{\mathrm{W}}}(\omega, z, r)\delta c_{\mathrm{W}}(z, r)dz. \tag{7.16}$$

Let us introduce the matrix form of (7.16). First, we introduce a discrete set of range values r_i and then consider (7.16) for every value of r_i. Next we divide the depth $0<z<h(r_i)$ into n intervals via the points z_k $(k = 1, 2...n)$. Correspondingly, we divide the interval $h(r_i)<z<D$ into m points z_k $(k = n + 1, 2...n + m)$. Next we denote the maximum mode number in our problem as N. So we obtain the following matrices:

$$\mathbf{G}_{\mathrm{B}} = \begin{pmatrix} G_{\mathrm{B}1}(\omega, z_{n+1}, r_i) & \ldots & G_{\mathrm{B}1}(\omega, z_{n+m}, r_i) \\ \ldots & \ldots & \ldots \\ G_{\mathrm{B}N}(\omega, z_{n+1}, r_i) & \ldots & G_{\mathrm{B}N}(\omega, z_{n+m}, r_i) \end{pmatrix}$$

$$\mathbf{G}_{\mathrm{W}} = \begin{pmatrix} G_{\mathrm{W}1}(\omega, z_1, r_i) & \ldots & G_{\mathrm{W}1}(\omega, z_n, r_i) \\ \ldots & \ldots & \ldots \\ G_{\mathrm{W}N}(\omega, z_1, r_i) & \ldots & G_{\mathrm{W}N}(\omega, z_n, r_i) \end{pmatrix}. \tag{7.17}$$

The dimensions of the matrices $\mathbf{G}_\mathrm{B}$ and $\mathbf{G}_\mathrm{W}$ are $N \times m$ and $N \times n$ correspondingly; they are generally different.

To calculate the integrals in (7.16), we can use approximate methods, for example take equal intervals $\Delta z = z_{k+1} - z_k = D/(n+m) = \text{const}$. In this case, we take the values of the sound speed at the point of division $\delta c_\mathrm{W}(r_i, z_k)$, $\delta c^*_\mathrm{B}(r_i, z_k)$ as components of vectors:

$$\delta \mathbf{c}_\mathrm{W}(r_i) = \begin{pmatrix} \delta c_\mathrm{W}(r_i, z_1) \\ \ldots \\ \delta c_\mathrm{W}(r_i, z_n) \end{pmatrix}, \ \delta \mathbf{c}^*_\mathrm{B}(r_i) = \begin{pmatrix} \delta c^*_\mathrm{B}(r_i, z_{n+1}) \\ \ldots \\ \delta c^*_\mathrm{B}(r_i, z_{n+m}) \end{pmatrix}. \tag{7.18}$$

Thus (7.16) can be written in matrix form

$$\mathbf{G}_\mathrm{B} \delta \mathbf{c}^*_\mathrm{B}(r_i) = \mathbf{G}_\mathrm{W} \delta \mathbf{c}_\mathrm{W}(r_i). \tag{7.19}$$

Formally, we can write a solution for the bottom part of the sound speed profile (or for the matrix $\delta c^*_\mathrm{B}(r_i, z_k)$). Solving (7.19) by taking a generalized inverse of $\mathbf{G}_\mathrm{B}$ gives the inversion errors due to water column mismatch $\delta c^*_\mathrm{B}(r_i, z_k)$ as the following:

$$\delta \mathbf{c}^*_\mathrm{B}(r_i) = \mathbf{G}_\mathrm{B}^{-1} \mathbf{G}_\mathrm{W} \delta \mathbf{c}_\mathrm{W}(r_i). \tag{7.20}$$

The above equation represents the "equivalent transform" of the water column mismatch error into inversion error. Therefore, we can define a transform matrix $\mathbf{H}_m = \mathbf{G^{-1}}_{m\mathrm{B}}\, \mathbf{G}_{m\mathrm{W}}$, and the equivalent transform can be briefly expressed as

$$\delta \mathbf{c}^*_\mathrm{B}(r_i) = \mathbf{H}_\mathrm{BW} \delta \mathbf{c}_\mathrm{W}(r_i). \tag{7.21}$$

The above equations are for the case of range-dependent water column mismatch. If a geoacoustic inversion is done by using range-averaged acoustic data, such as modal travel-times, the inversion errors will become range-averaged:

$$\overline{\delta c_\mathrm{B}{}^*}(z, R) = \frac{1}{R} \int_0^R \delta c_\mathrm{B}{}^*(z, r) dr, \tag{7.22a}$$

or in discrete form

$$\overline{\delta \mathrm{c}_\mathrm{B}{}^*} = \frac{1}{N} \sum_{i=1}^{N} \delta c_\mathrm{B}(r_i)^*. \tag{7.22b}$$

Before ending this section, we note a few points. First, since (7.19) will generally describe an under-determined system in practice, some specific generalized inverse

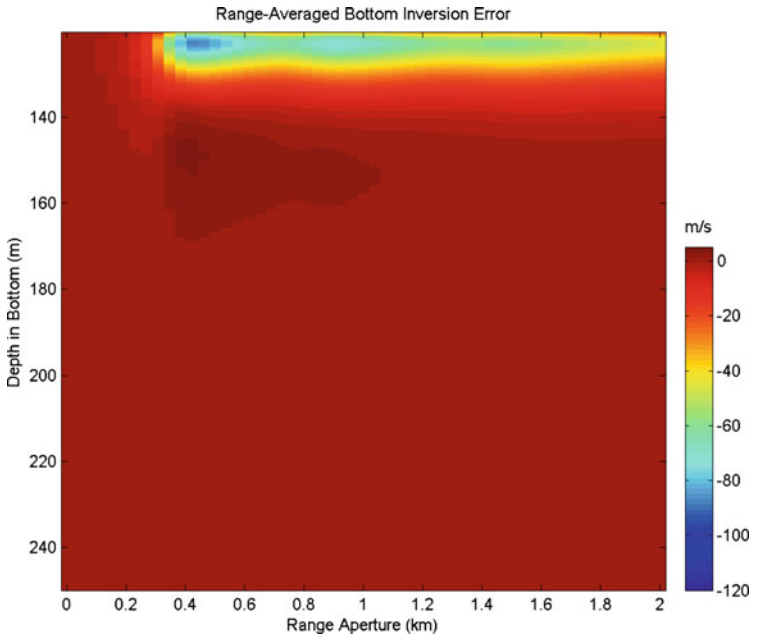


Fig. 7.1 Courtesy IEEE. JOE. The range-averaged inversion error in test case II, $\overline{\delta c_B{}^*}(z, R)$, due to the water column mismatch shown in Fig. 7.15c. From Figure 19 (Lin et al. 2006)

scheme, such as the singular value decomposition or SVD (Menke 1989, Aster et al. 2005), has to be employed to obtain the inverse matrix, $\mathbf{G}_B^{-1}$. This will be examined in detail soon. Second, if one considers that the modification of the inverse due to water column mismatch is seen not only in the bottom but also in the water, the transform matrix will become

$$\mathbf{H}_{BW} = \mathbf{G}^{-1}\mathbf{G}_W \tag{7.23}$$

where $\mathbf{G}$ includes both the water column and bottom portions of the integrand kernel, and the water column mismatch will equivalently transfer into the whole depth including both water and bottom. Third, this equivalent transform method can be applied to other kinds of inverse approaches, even though we only described the wave number integration approach in this section. Lin et al. (2006) discusses other approaches in his paper.

As an example, we show Fig. 7.1 from Lin et al. (2006), which gives the magnitude of the errors in the bottom profile that can be generated by not accounting for internal wave solitons in a coastal ocean environment. The reader interested in detail is referred to this article – however, the salient detail to notice is that the magnitude of the error can be quite considerable – easily 100 m/s, which is on the order of the size of the bottom sound speed perturbation we wish to estimate. We will also note that, given that one suspects that there is unaccounted for water column variation, a joint

inverse for the water column and bottom properties can reduce the bottom error. Lin and Lynch are also pursuing the use of a "reduced data set" via data nullspace projection, where data which are likely to have the most mismatch error and non-Gaussian statistics are preprocessed to reduce their effect (Lin and Lynch, private communication). This has shown promising results.

7.3 The Generalized Inverse Solution, Its Error, and an Example from the Corpus Christi

7.3.1 GEMINI Experiment

We have briefly discussed the linear inverse problem equations at this point, and shown an example. We now look how one finds a solution to the linear inverse problem, what the usual error estimates associated with it are, and then show an example from an experiment that used a wave number inverse to find bottom properties.

As mentioned, the inverse problem has been the subject of a vast amount of research, and many inverse methods and their variants exist. We will consider here only one of the possibilities, the singular value decomposition/generalized inverse solution, as it is one of the simplest, most popular, and perhaps most powerful of the linear techniques. It is particularly suited for work with ill-conditioned and/or rank deficient systems, which is the type of problem we usually have to deal with in shallow-water acoustics. In doing this, we will largely follow the text in (Aster et al. 2005), as it is the most modern and inclusive text on this area that we have found.

We begin by noting that the kernel of the linear inverse, $\mathbf{Gm} = \mathbf{d}$, $\mathbf{G}$ can be factored into the form

$$\mathbf{G} = \mathbf{USV}^T \tag{7.24}$$

where $\mathbf{U}$ is an $m \times m$ orthogonal matrix whose columns form a basis spanning the data space, $\mathbf{V}$ is an $n \times n$ orthogonal matrix whose columns form a basis spanning the model space, and $\mathbf{S}$ is a diagonal matrix with nonnegative elements called the "singular values." This matrix decomposition is called the "singular value decomposition" (SVD) and is easily implemented via the MATLAB command **svd**. Quite frequently, some of the singular values are zero, and in this case, we consider the first p singular values that are nonzero in a particular way. Specifically, we create matrices $\mathbf{U}_p$, $\mathbf{V}_p$ and $\mathbf{S}_p$ where $\mathbf{U}_p$ contains the first p columns of $\mathbf{U}$, $\mathbf{V}_p$ contains the first p columns of $\mathbf{V}$ and $\mathbf{S}_p$ is a $p \times p$ diagonal matrix composed of the positive singular values. From these matrices, we can create a general solution to the inverse problem, called the Moore–Penrose pseudoinverse, which is given by

$$\mathbf{G}^+ = \mathbf{V}_p \mathbf{S}_p^{-1} \tag{7.25}$$

so that

$$\mathbf{m}^{+} = \mathbf{G}^{+}\mathbf{d}. \tag{7.26}$$

The pseudoinverse has many useful properties, a few of which we will note here. First, it is easily implemented via the MATLAB command **pinv** which generates $\mathbf{G}^{+}$. MATLAB also gives the user the flexibility of choosing a tolerance for rejecting the smallest singular values, which can create significant error in the solution unless dealt with properly. Finally, the generalized inverse solution always exists.

Having obtained a solution, it is desirable to place error bounds upon it, so as to determine its quality. The two metrics for error that are commonly used are the resolution and the covariance. We will look at these briefly.

If we start with the mathematically useful, but sometimes disputable, assumption that measurement errors are independent and normally distributed, we can write that the covariance of the model solution is

$$\mathrm{Cov}(m^{+}) = G^{+}\mathrm{Cov}(d)(G^{\dagger})^{\mathrm{T}}$$

or in terms of the SVD matrices

$$\mathrm{Cov}(m_{\dagger}) = \sigma^2 V_p {S_p}^{-2} V_p^T. \tag{7.27}$$

The resolution of the model estimate is also obtained from the generalized inverse as

$$\mathbf{R}_m = \mathbf{G}^{+}\mathbf{G} = \mathbf{V}_p\mathbf{V}_p^T. \tag{7.28}$$

The trace of $\mathbf{R}_m$ in (7.28) can be used as a quick estimate of how good the resolution is. If the trace is close to n, then $\mathbf{R}_m$ is close to the identity matrix, and the resolution is almost perfect. If the resolution is not so good, one often uses the "peak width" in the rows of $\mathbf{R}_m$ as a measure of the resolution, called the "resolution length."

There are numerous other aspects to the linear inverse which we have ignored here, such as the stability of the inverse, how to effectively truncate the SVD, "regularization" methods, etc. Again, we refer anyone interested in doing an inverse problem to the texts we referenced previously.

Having looked at what a shallow-water inverse problem is, and how one could solve it, we now turn to a concrete example. Specifically, we examine an inversion experiment for the bottom sound speed performed off Corpus Christi, Texas, as a test of how well one can predict and invert for bottom properties (Lynch et al. 1991). In this experiment, a horizontal synthetic aperture array was created (see Fig. 7.2) to estimate the modal wave numbers, our "data" for (7.4)–(7.6) above. Using a Hankel

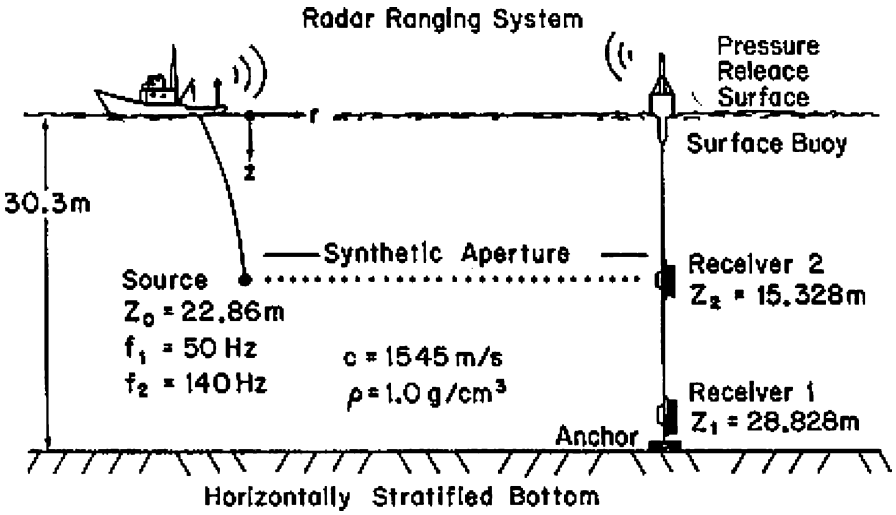


Fig. 7.2 Experimental configuration for the GEMINI experiment performed off Corpus Christi, Texas in September, 1985, courtesy JASA

transform (implemented approximately by a fast field program) on the $p(r)$ data created by the experiment produces a depth-dependent Green's function, whose peaks are the modal eigenvalues we desire.

For the water depth at the main site (30.3 m) and the frequencies used (50 and 140 Hz), we obtain only 2–3 trapped mode eigenvalues, and thus are clearly dealing with the underdetermined problem discussed above. The water column was well mixed due to a severe storm passing by just before the experiment, so our worries about having unknown water column variability became a nonissue. The water was very close to an isovelocity 1,545.5 m/s. The bottom was previously well measured by two other methods – a broadband experiment by Rubano a decade previously (Rubano 1980) and a geological survey by (Matthews et al. 1985). Thus, we felt that we knew the bottom to "within a perturbation," an ideal setup for using linear inverse methods.

Using linear inversion techniques very similar to the ones discussed above, we were able to retrieve the bottom model shown in Fig. 7.3, which has two range segments due to our accounting for the range dependence of the waveguide.

The resolution and variance of this bottom model were also estimated. The resolution was seen to be very good near the surface (order of 5 m resolution length) and then progressively degraded to about 30 m with depth, until at ~140 m depth, the resolution length became infinite, showing that no appreciable energy was illuminating the bottom at that depth (see Fig. 7.4). The RMS error (square root of the variance) versus depth was found to increase slowly from 10 to 20 m/s over the top 140 m of the sediment. We should note that there is a tradeoff between resolution and variance when doing such inverse procedures, so that our exact error numbers will somewhat reflect our "prejudices" in doing this tradeoff. The rather

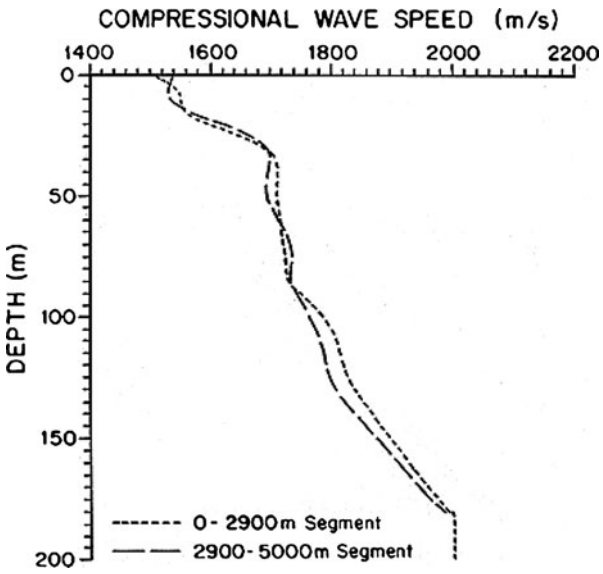


Fig. 7.3 Soundspeed in the bottom off Corpus Christi, Texas estimated from modal eigenvalue data, courtesy JASA

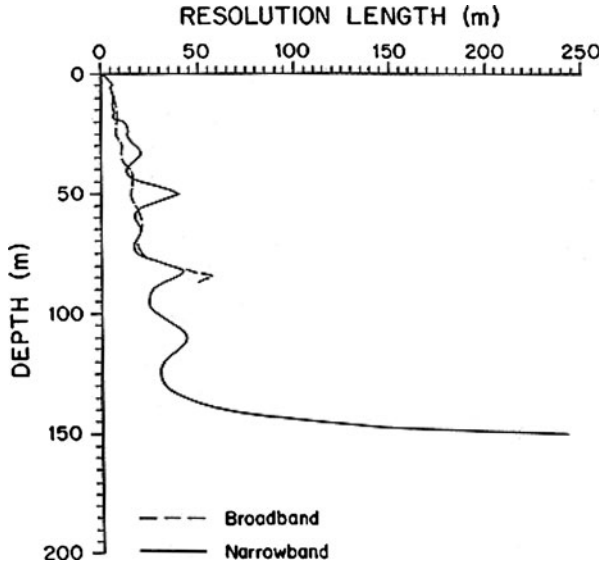


Fig. 7.4 Resolution length versus depth into the bottom for the Corpus Christi, Texas site examined by the GEMINI experiment. The narrowband result corresponds to the experiment discussed above. The broadband result (which is very consistent with the narrowband result) comes from using broadband data collected by Rubano a decade previous to the GEMINI experiment at the exact same site. Courtesy JASA

satisfying outcome of this exercise was that the inverse, even when working from a relatively crude "background model" guess at the bottom, gave good agreement with the detailed geological survey, certainly to within the accuracy that we could expect from the data.

Our presentation so far has examined one particular type of inverse for shallow-water environmental properties – a modal wave number inverse for the bottom sound speed. However, we note that there many more data types and many more bottom and water column properties that we can invert for, so that our illustrative work above represents the tip of the iceberg. We will look here at a few more of the forms one can work with, note which others may be of use, and also show how first order, time-independent perturbation theory is used to generate the linear forms that we can invert.

In trying to invert for bottom or water column properties, we pragmatically will use any and all available data that may illuminate the model space we are interested in. There are a few more common data types that we should mention in particular, which have been used in inversions by various investigators. The first one is modal group velocity, which is measured by finding the travel time of a resolvable mode over a known source-to-receiver distance. To find the perturbation of group velocity (or its useful reciprocal, the group slowness), we do not have to go through the procedures outlined above, as the answer is almost in our hands already. One simply takes the frequency (ω) derivative of our perturbation equation for δq_l to get the form for $\delta c_l^{\mathrm{g}}(\omega)$, the group velocity perturbation, as shown in Lynch et al. (1991). In Lynch et al. (1991), it was shown that the inverse results from both a wave number perturbation technique and a group velocity perturbation technique agree well at a common site (the Corpus Christi site discussed above), as one would hope! Another type of modal data that might be useful for inversions is the modal amplitude. Due to the somewhat more difficult nature of obtaining modal amplitudes as opposed to modal eigenvalues or group velocities, as well as the more complicated expressions one generates perturbing the modal wavefunctions, this type of data has not been exploited as much to date. However, efforts are being made in Thode (2004), Poole et al. (2008), and others to utilize such data, and there should be results in this area soon.

Another very common data type is transmission loss versus range at a given frequency, $\mathrm{TL}(r, \omega)$. Rajan (1992) has shown how one may set up a perturbation problem using the difference between measured TL and that predicted by a background model. It is interesting that again we see a data set that has a potentially infinite number of points even at one frequency, since the r variable is continuous. However, as with the frequency variable, the number of actual degrees of freedom that are contained in the data is finite. There is also the interesting technicality that the difference between the measured and predicted TL can be (and usually is) quite large in the regions of the nulls, giving us a highly nonlinear behavior to the difference at certain ranges. These technical intricacies aside, the TL is a common and easy measurement to make, and so inversions using it are quite useful.

In cases where one can linearize a problem, the mechanics of linear inverse theory are available, easy to use, and reliable. However, not all problems can be reduced to linear, and given that we do not ever wish to discard useful data, we must

find ways to deal with nonlinear inversion problems. At this point in time, many such methods exist, and while they are less easy to use than linear methods, and less transparent, they are still extremely useful nonetheless. We look at a small number of them here.

In cases of mild nonlinearity, it is well known that iterated use of linear inversions will converge to an answer, as the local minimum is also the global one. This is probably the simplest approach to nonlinearity. However, when the degree of nonlinearity increases, other approaches are needed.

The simplest approach to the general nonlinear problem is the brute force computation method, in which one generates all possible models over the space of allowable ones, and then calculates what data that is produced by these models by using a forward model of some sort. (In our case, this would usually be some acoustics code that would generate a field realization.) The model that gives the best fit to the data (perhaps along with some other cost function criteria) wins the competition, and is chosen. Even in the age of high speed computing this is wasteful and inefficient (though admittedly robust), and so we look to other ways to find a global minimum.

Monte Carlo, random sampling of the model space is often used, with some adaptations that allow for more efficient sampling, e.g., algorithms which determine optimal step size, etc. One of the more popular Monte Carlo variants is called "simulated annealing," which is designed to find the maximum likelihood point of any probability density function (e.g., the posterior probability distribution of the model, which is what the random search begins to produce). This approach incorporates a Metropolis Algorithm (Tarantola 2005) to try to avoid unproductive areas of the model space, which produce maxima or shallow local minima in the cost function. While being somewhat computer intensive, and requiring some close attention by the user, this technique gives good answers and insights into "model parameter sensitivity," and is one of the more widely used nonlinear techniques. A very nice application of this technique to shallow-water acoustics, in the context of bottom parameter inversion, was shown by Collins et al. (1992). In their study, they applied this technique to the same Corpus Christi data set that was inverted using linear techniques (Lynch et al. 1991). Using the nonlinear methods allowed them to invert a wider range of the data than were available to the linear inverse methods (e.g., the continuum modes) and also to comment in detail on the sensitivity of the data to various models parameters.

Another popular type of nonlinear inverse is the so-called "genetic algorithm," which is based on a biological analogy with genetic populations. This algorithm does not have the simple to interpret feature of the Metropolis Algorithm incorporated into it, but produces similar results to annealing, and is an equivalent tool in the minds of many researchers. A good example of the use of genetic algorithms for shallow-water acoustics can be found in the bottom inverse work (Potty et al. 2003), who use the technique to invert modal group velocity dispersion curves for bottom geoacoustic properties.

The Bayesian approach, named after Thomas Bayes and his renowned theorem, is qualitatively different from the approaches we have discussed so far, in that in the

Bayesian approach, the model we desire is a random variable, and the solution obtained is a probability distribution function for the model parameters. In our previous work, we searched for a specific, deterministic model $\hat{m}_{\text{true}}$ along with a description of what errors there were likely to be in the model we actually obtained. Differences in methodology aside, there is a deep philosophical difference between the two approaches, and perhaps the first thing that would be useful to do is to note when and where each approach should be used. It is the opinion of the authors that in cases where one is dealing with individual realizations of the ocean or seabed, such as the bottom geoacoustic properties at a fixed point or a snapshot of the features of the ocean at a particular time, and moreover where one is not worried about averaging over fine-scale, unresolvable detail, then inversion for a deterministic model makes sense. Moreover, it is usually easier to do. However, if one is looking at data which inherently contains a number of realizations of either the ocean or seabed (e.g., data covering more than one correlation length or time), then a Bayesian approach may be more appropriate, as one is inherently looking at a distribution of local "true" models. Of course, both the deterministic and Bayesian approaches also have a close correspondence with each other, so that one is not comparing completely dissimilar quantities. For instance, the maximum a posteriori estimate of the model taken from the Bayesian approach will often correspond to the deterministic model estimate. And conversely, the error spread that one sees in the deterministic model will usually correspond to the moments of the probability distribution function that obtains for the model in the Bayesian approach.

We will not rehash the machinery of Bayesian inversion here, but rather recommend the reader to the books by Tarantola, Aster et al, and Leonard and Hsu (op. cit.) for detailed discussions. We would also point the reader to recent work by (Dosso 2002; Dosso and Nielsen 2002) as good examples of how Bayesian methods are applied to shallow-water acoustic bottom geoacoustic property inversions.

7.4 Examples of Nonlinear Bottom Property Inversion Using Various Data Types

In the previous sections we considered reconstruction of waveguide properties (bottom properties) on the basis of the measurement of eigenvalues of separate modes. But more commonly, we measure the sound field as a function of spatial coordinates and time. In (7.1) for the value d, we can have sets of functions of different variables, depending on the statement of the experiment. For example if we have a vertical receiving array along a stationary acoustic track, receiving for a long time, we will have a set of functions I, each giving the temporal dependence for each depth (z_i), $i = 1, 2...I$ where I is the number of hydrophones. In another experimental setup, where we have a fixed receiving array and a moving source, we have I functions of range r to the source. It is possible to have yet another set of data using combinations of these experiments and (or) work with broadband signals, etc.

This means that in the matrix form we have a matrix column **d**, where $d_i = f_i(r, t)$ (and frequency dependence also can be added). In this case, the procedure for bottom property reconstruction becomes essentially nonlinear. Bottom parameter reconstruction can be achieved on the basis of the best fit between modeled and measured data using the minimization of fitting error, similar to the approaches discussed previously.

Let us consider some examples of bottom parameter reconstruction based on a direct relationship between these parameters and the sound field in a near bottom sound channel. We note that we are dealing with bottom parameter reconstruction in the framework of some known, a priori bottom model and that we will be trying to determine the model parameters averaged over a large range along the acoustic track. The relationship between these parameters and sound field was demonstrated previously in this book.

The foundation of the method we will use here is the comparison between the measured and the calculated interference pattern of the sound field using a quantitative comparison test. In this method, it is possible to use interference patterns of various types e.g., (1) the spatial dependence of the sound intensity at given depth as a function of distance between acoustic source and receiver in the water layer (the horizontal interference pattern), (2) the depth dependence of the intensity (the vertical interference structure) and (3) the interference pattern in the frequency domain when the sound sources and receivers are held in place and wide-band signals are used. It is assumed that the best agreement between the measured and calculated interference patterns corresponds to the true values of the bottom parameters.

7.4.1 Horizontal Interference Structure and Bottom Parameters Estimation

Let us first consider an example of these methods via the reconstruction of some bottom parameters in the Barents Sea (Katsnelson and Petnikov 2002). In this region, the sea bottom is represented by layers of unconsolidated sediment overlying basic rock (basalt, granite, etc.). The thickness of the upper layer is generally several tens of meters. Its shear modulus is practically equal to zero. One characteristic property of the sediment layer is here the existence of abrupt random inhomogeneities. These include layered (intermittent and tapered) structures of length up to tens of kilometers, vertical channels associated with the venting of gases, and diapirs (dome-shaped folds, in which rocks with high plasticity are extruded from below). As a result, for most regions of the Barents Sea, we can consider the sea bottom as liquid, randomly inhomogeneous medium with mean effective parameters (sound speed c_1, density ρ_1, absorption coefficient $\hat{\beta}$) which need to be reconstructed. In the language of our previous work, the vector $\hat{m}$ (the model) has three components: $c_1, \rho_1, \hat{\beta}$. For this bottom model, the coefficient $\hat{\beta}$ is

hardest to determine (Kuperman and Ingenito 1980). According to the nature of the bottom, $\hat{\beta}$ may vary by a factor of 20–30, whereas the variability of the sound speed c_1 and the density ρ_1 amount to only 10–20% of their mean values. It is not an overstatement to say that finding the coefficient $\hat{\beta}$ is one of the primary tasks in shallow-water acoustics.

In experiments, the spatial dependence of the sound field intensity is commonly measured using a stationary receiver and a CW multitone acoustic source towed away from it. Because of the reciprocity theorem, the dependence of the sound field intensity on the distance r obtained in this way is equivalent to the dependence of the intensity on the distance to a stationary source, from a receiver at a fixed depth. Usually, this dependence is constructed using a logarithmic scale, in other words, when speaking about the horizontal interference pattern, we have in mind the function $\delta(r,z) = 10\lg\frac{I(r,z)}{I(r_0^*,z)}$. Here, r_0^* is a reference distance at the beginning of the acoustic track (i.e., at the beginning of the bottom area to be reconstructed). The quantity δ is very irregular as a function of spatial coordinates, being the result of the interference of spatial oscillations over different scales. In our scheme, we experimentally measure $\delta_{\mathrm{e}}(r,z)$ and also calculate the functions $\delta_{\mathrm{th}}(r,z)$. Using the notation of the first section, $\delta_{\mathrm{e}}(r,z)$ at different depths can be interpreted as components of the vector $\mathbf{d}$, and the values $\delta_{\mathrm{th}}(r,z)$ are considered in turn as values of $\mathbf{B}(\mathbf{m})$. In order to construct a method of "best fitting" between $\mathbf{d}$ and $\mathbf{B}(\mathbf{m})$, we should find criteria to compare these values, in other words to construct some norm for the difference $\|\mathbf{d} - \mathbf{B}(\mathbf{m})\|$. One such criterion might be the following (Bunkin et al. 1989a). Let us represent the function $\delta(r,z)$ in the form of a sum of mean and fluctuating parts $\delta(r,z) = \bar{\delta}(r,z) + \tilde{\delta}(r,z)$, where $\bar{\delta}$ corresponds to the intensity $\bar{I}(r,z)$ averaged over a longitudinal spatial interval greater than the maximum scale of the interference oscillations (e.g., the mode cycle distance), and $\tilde{\delta}$ corresponds to the fluctuating part of the intensity $\tilde{I}(r,z) = I(r,z) - \bar{I}(r,z)$. According to this, the "proximity coefficient" of the functions will also consist of two terms and is defined by the following expressions:

$$K = \bar{K} + \tilde{K}, \qquad \delta = \bar{\delta} + \tilde{\delta},$$
$$\bar{K} = \frac{\int_0^r \bar{\delta}_{\mathrm{e}}\bar{\delta}_{\mathrm{th}}\mathrm{d}r}{\int_0^r \bar{\delta}_{\mathrm{e}}^2\mathrm{d}r + \int_0^r \bar{\delta}_{\mathrm{th}}^2} \quad \tilde{K} = \frac{\int_0^r \tilde{\delta}_{\mathrm{e}}\tilde{\delta}_{\mathrm{th}}\mathrm{d}r}{\int_0^r \tilde{\delta}_{\mathrm{e}}^2\mathrm{d}r + \int \tilde{\delta}_{\mathrm{th}}^2\mathrm{d}r}. \tag{7.29}$$

It is easy to see that $0 \le K \le 1$ and the maximum value of K (ideally 1) corresponds to the best agreement of calculation and experiment.

By way of example, experimental and calculated functions $\delta_{\mathrm{e}}(r,z)$ and $\delta_{\mathrm{th}}(r,z)$ are plotted in Fig. 7.5 for a radiation frequency of 100 Hz. (Here r_0^* and z equaled 0.5 km and 65 m respectively.) In constructing the curves, the values of $I(r,z)$ were multiplied by r/r_0^*, which enabled us to correct the decrease in I with distance due to cylindrical spreading of the wavefront, and also to show the interference patterns more graphically. The measurements were performed in the Barents Sea using the participation of

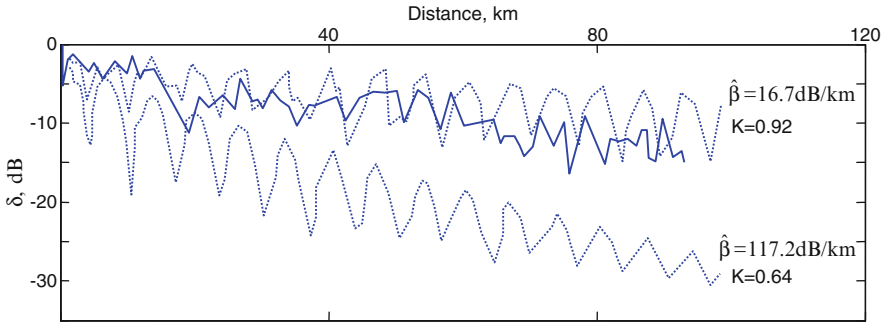


Fig. 7.5 Calculated (*dotted lines*) and experimental (*solid line*) dependence of the relative intensity of a point sound source on distance

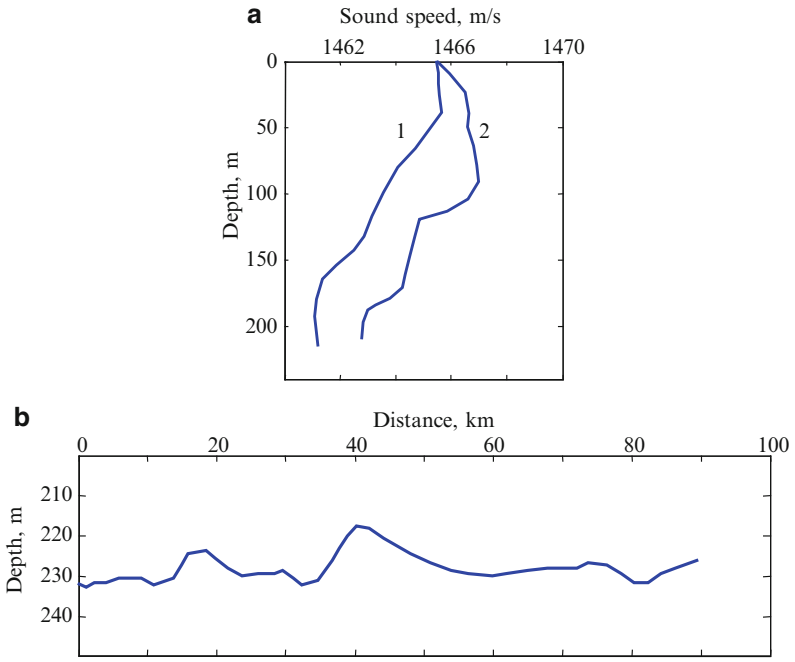


Fig. 7.6 Sound speed profiles $c(z)$ at the start (1) and the end (2) of an acoustic track (**a**), changes in depth of the waveguide $H(r)$ along the track (**b**)

two ships, one of which towed sound sources at a depth ≅40 m, while the other lowered a vertical chain of hydrophones located at depths of $z = 65, 80, 95$ and 125 m. Two CW signals, with frequencies of 100 and 230 Hz, were radiated simultaneously. Profiles of the sound speed and the water depth in the measurement region are shown in Fig. 7.6.

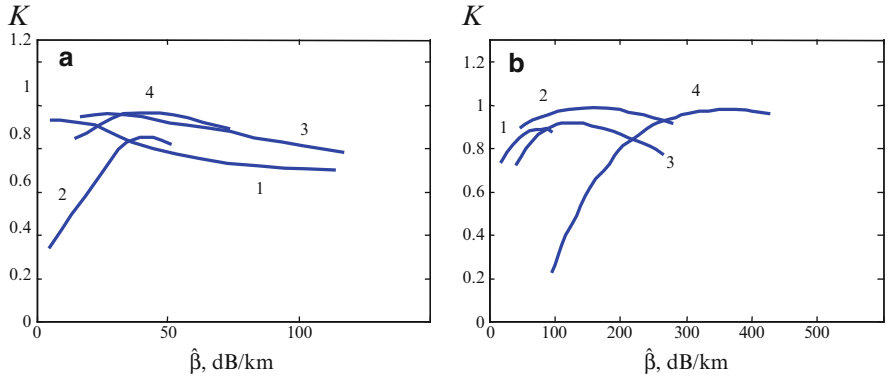


Fig. 7.7 Dependence of the values of the numerical proximity criterion K on the parameter $\hat{\beta}$ for frequency 100 (**a**) and 230 Hz (**b**). 1–4 – receiver depth in meters: 1 – 125, 2 – 95, 3 – 80, 4 – 65

In Fig. 7.5 one of the calculated curves corresponds to optimal $\hat{\beta}$ (when K is maximum), and the other to an arbitrary $\hat{\beta}$. The other parameters of the bottom in the calculations were taken to be $\rho_1 = 2$ g/cm^3, $c_1 = 1{,}630$ m/s, which correspond to typical values of the density and sound speed in the region studied. (The method discussed here was used only for determination of the sound absorption coefficient $\hat{\beta}$.) One can clearly see that for nonoptimal values of $\hat{\beta}$, the experimental and numerical curves diverge strongly. From this and analogous graphs, one can construct the dependence of $K(\hat{\beta})$ for different values of the depth of the receivers. Such graphs for frequencies 100 and 230 Hz are shown in Fig. 7.7. Evidently, most of the $K(\hat{\beta})$ have a very clearly defined maximum. This maximum is found in a single range of values of $\hat{\beta}$ for different depths. According to our method of bottom parameter reconstruction, this value can be considered as the true one.

7.4.2 *Using the Vertical Interference Pattern for Bottom Parameter Estimation*

It is interesting to compare the results of bottom parameter inversion using the sound field interference pattern in the frequency domain with a similar scheme using the comparison between measured and calculated vertical interference patterns (Katsnelson and Petnikov 2002). The intercomparison can be done by using the data from the experiment described above. During this experiment, CW signals of 100 Hz frequency were radiated continuously as well as other signals. For this purpose an additional sound source was placed on the sea bed. A vertical receiver chain of ten hydrophones was also installed on the sea bed, of 66 m length. The lowest hydrophone was at a distance of 18.5 m from the bottom. The processing of the received signals involved identifying the amplitudes $|P_e(z_i)|$ and the phases

$\varphi_e(z_i)$ of the sound field for subsequent comparison with the theoretical values, $|P_{th}(z_i)|$ and $\varphi_{th}(z_i)$, where z_i are the hydrophone depths. We note that the theoretical and experimental values are determined by (7.30) and (7.31) respectively:

$$P(z_i,t) = |P_{th}(z_i)|e^{i[\varphi_{th}(z_i)-\omega t]} \tag{7.30}$$

$$\begin{aligned} |P(z_i)| &= \sqrt{p_{cos}^2(z_i) + p_{sin}^2(z_i)} \\ \varphi(z_i) &= \arctan(p_{sin}(z_i)/p_{cos}(z_i)), \end{aligned} \tag{7.31}$$

where $p_{sin}(z_i) = \langle p(z_i,t) * \sin\omega t\rangle$ and $p_{cos}(z_i) = \langle p(z_i,t) * \cos\omega t\rangle$ are the sine and cosine quadrature components of the real (measured) field $p(z_i,t)$, which can be related to the field $P(z_i,t)$ using our earlier work. The angle brackets denote averaging over the period. Simple transformations show that these two amplitude and phase definitions are equivalent from a physical point of view. In our scheme, a simple model was used in which the bottom was assumed to be a homogeneous fluid half space with three effectively unknown parameters: sound speed c_1, density ρ_1, and absorption coefficient $\hat{\beta}_f$. Comparison of theoretical and experimental values was performed based on the quantitative criteria:

$$\begin{aligned} \tilde{\sigma}_\varphi &= \frac{1}{J_h}\sqrt{\sum_{i=1}^{I}(F_e(z_i) - F_{th}(z_i))^2} \\ \tilde{\sigma}_P &= \frac{1}{J_h}\sqrt{\sum_{i=1}^{I}\left(|P_e(z_i)|^2 - |P_{th}(z_i)|^2\right)^2}. \end{aligned} \tag{7.32}$$

Here J_h is the number of hydrophones, and $F(z_i) \equiv \cos(\varphi(z_i))$. We note that in order to eliminate an arbitrary constant phase difference between theory and experiment, a constant phase value was added to each theoretical value of $\varphi_{th}(z_i)$, ensuring equality of the phases at the first hydrophone ($\varphi_{th}(z_1) = \varphi_e(z_1)$). The search for the parameters was done independently for the phase and amplitude distribution of the sound field, as these are independent data. The values of the bottom parameters corresponding to the best agreement between theory and experiment for the phase distribution (minimal value $\tilde{\sigma}_\varphi$) were as follows: $c_1 = 1{,}800$ m/s, $\hat{\beta}_f = 0.25$ dB/(km Hz) ($\alpha = 0.016$), and $\rho_1 = 1.8$ g/cm^3. An analogous procedure for the amplitude distribution (minimal value $\tilde{\sigma}_P$) yielded the following set of parameters: $c_1 = 1{,}780$ m/s, $\hat{\beta}_f = 0.23$ dB/(km Hz), $\rho_1 = 1.8$ g/cm^3. The theoretical vertical dependences $F_{th}(z_i)$ and $|P_{th}(z_i)|$ corresponding to these parameters are shown in Fig. 7.8. The figure also shows the corresponding experimental forms of $F_{th}(z_i)$ and $|P_{th}(z_i)|$.

As can be seen, the estimate obtained from the coefficient α is slightly lower than the value obtained by processing the interference pattern in the frequency domain.

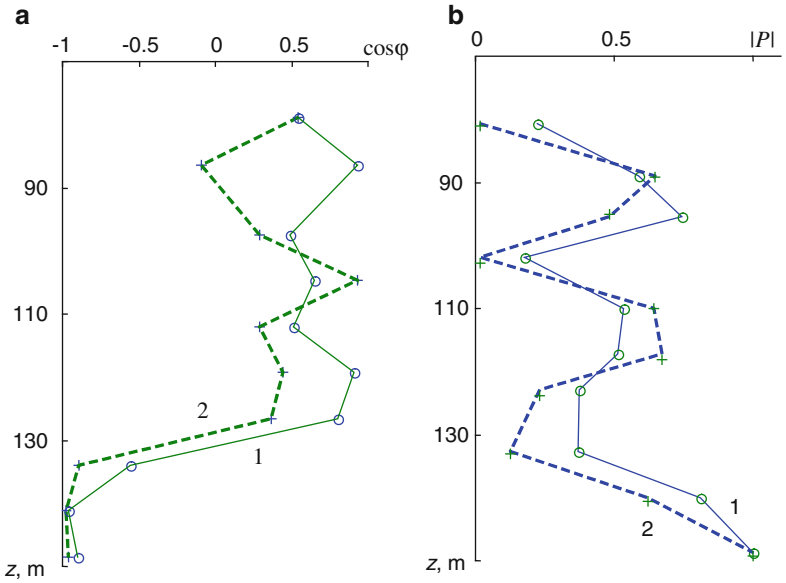


Fig. 7.8 Vertical distribution of cosφ (**a**) and amplitude |P| (**b**). 1 – experiment, 2 – theory

However, it is important to keep in mind that the vertical interference pattern depends only weakly on the sound absorption [the corresponding minima of $\tilde{\sigma}_{\varphi}(\alpha)$ and $\tilde{\sigma}_{P}(\alpha)$ are weak over the range of possible α variation (Katsnelson and Petnikov 2002)]. Because of this, the accuracy of the α estimate is not very high. A similar situation takes place with respect to the density ρ_1. In contra-distinction, the vertical interference pattern depends strongly on the sound speed in the bottom. As a result the accuracy of estimation of the bottom sound speed is good enough in this case to recommend the inverse method for determining c_1.

We now discuss our previous observation that the bottom parameters obtained using the amplitude and phase distribution gave similar but not coincident values. These values also differ slightly from the values obtained by processing the interference pattern in the frequency domain. The observed spread of values can most likely be related to the fact that the parameters of the waveguide vary along the track, and so their values at specific ranges can be considered as random quantities.

7.5 Broadband Experiments and the Frequency Dependence of the Bottom Parameters

In the example we treated, previously the absorption coefficient $\hat{\beta}$ showed an approximately linear dependence on frequency. Correspondingly, the absorption coefficient normalized to frequency $\hat{\beta}_f$ $(\hat{\beta}_f = \hat{\beta}/f)$ does not depend on frequency

and is equal to $\hat{\beta}_f \cong 0.17 - 0.5$ dB/(km Hz). (The corresponding value of the dimensionless parameter α describing the imaginary part of the bottom refractive index equals $\alpha = \frac{c_1[\text{km/s}]}{27.3}\hat{\beta}_{af}[\text{dB/km Hz}] = 0.01 - 0.03$.) Naturally, such a dependence does not necessarily occur in other shallow-water regions and in other frequency bands. To determine this frequency dependence, it is best to apply a bottom parameter inverse method which uses an interference pattern in the frequency domain. For a different region of the Barents Sea, this method of frequency-dependent reconstruction of $\hat{\beta}_f(f)$ was used (Grigor'ev et al. 2001). In this case, the reconstruction was carried out for a relatively small region of the bottom ≈14 km in length, where according to data from a Parasound parametric echo-sounder, the upper layer of bottom sediments presents a more or less homogeneous medium, without the large inhomogeneities mentioned above. For this reason, a different and physically more adequate bottom model was used, based on the role of different mechanisms of sound absorption. The bottom parameters determining the frequency dependence were also refined by comparing the calculated sound field in the water layer with the experimental data. It was suggested that the bottom sediments were best represented as a two-component medium, i.e., water plus mineral particles. The main parameter that can be used to characterize the sediments is the porosity, which is the ratio of the volume occupied by seawater to the total volume of the bottom sample under consideration. The geoacoustic parameters of the bottom sediments, specifically the propagation velocity of longitudinal waves and the density in the sea floor, are thus expressed through the values of porosity. These can be adequately described by expressions from (Aikal 1972), which are

$$\begin{aligned} c_1 &= c_H\left(1.631 - 1.78\kappa + 1.2\kappa^2\right) \\ \rho_1 &= \rho_H(2.604 - 1.606\kappa), \end{aligned} \tag{7.33}$$

where c_H and ρ_H are the sound velocity in water and the water density near the sea floor, respectively. Within the framework of this model, it is possible to separate two basic mechanisms that are responsible for sound attenuation in the sea floor. The first mechanism is sound absorption via the transformation of sound energy to heat. We will denote the sound absorption coefficient connected with this mechanism by the coefficient $\hat{\beta}_a$ ($\hat{\beta}_{af}$ and α_a respectively). The second mechanism is sound scattering by volume inhomogeneities. In this case, the sound energy dissipates from the beam in various directions. The contribution of this mechanism is characterized by the coefficient β_v ($\hat{\beta}_{vf}$ and α_v respectively).

According to the data in the literature (Hamilton 1976), the quantity $\hat{\beta}_{af}$ is determined by the semiempirical expressions:

$$\hat{\beta}_{af}\left(\frac{\text{dB}}{\text{km Hz}}\right) = \begin{cases} 0.2747 + 0.527\kappa,\ 0<\kappa<0.472, \\ 4.903 + 1.7688\kappa,\ 0.472<\kappa<0.52, \\ 3.3232 - 4.89\kappa,\ 0.52<\kappa<0.65, \\ 0.7602 - 1.487\kappa + 0.78\kappa^2,\ 0.65<\kappa<0.9 \end{cases}. \tag{7.34}$$

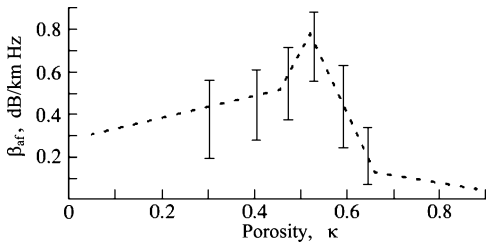


Fig. 7.9 Spread in attenuation versus porosity k

This absorption mechanism can be considered as mainly the result of friction between particles. An increase in the sound frequency means that each particle performs more vibrations per unit time, which makes the loss proportional to frequency. Figure 7.9 demonstrates the dependence of the absorption coefficient $\hat{\beta}_{af}$ on porosity according to empirical equation (7.31). The scatter of the experimental data for $\hat{\beta}_{af}$ obtained in different regions of the ocean is also displayed in this figure.

We note that, in the theory of water-saturated porous media (Hamilton 1976, Biot 1956a, b), the absorption mechanism due to the motion of the pore liquid relative to the ground skeleton is considered in addition to the losses caused by friction between particles. In this case, a viscous friction force arises, which opposes the relative motion of the liquid and solid phases. This also leads to a dissipation of the wave energy and its transformation to heat. At low frequencies, this mechanism leads to a quadratic dependence of the absorption coefficient on frequency (Biot 1956a, b). However, measurements performed for sediments on the continental shelf indicate the relative smallness of the contribution of this mechanism to the total loss. We should indicate that in highly porous sediments (suspensions), particles do not contact each other but are kept at a distance by the electrochemical forces. In this case, friction between particles is absent, and as a result, the frequency dependence of the absorption coefficient is not linear. This situation is found for deep-water clays (Smith 1974), but is not a common shallow-water scenario.

Strictly speaking, the scattering responsible for propagation loss is due not only to volume inhomogeneities but to rough surface inhomogeneities as well (Lysanov 1980, Ivakin and Lysanov 1985). However, according to echo-sounder data for the case we are examining, the surface inhomogeneities [small roughness of the bottom relief, comparable to the acoustic wave length at low frequencies (<100 Hz)] were very small and only volume inhomogeneities needed to be taken into account in the model. The value $\hat{\beta}_v$ is determined as the ratio of the sound power W scattered by a unit volume in all directions to the intensity I_0 of the incident plane wave:

$$\hat{\beta}_v = \frac{W}{I_0}. \tag{7.35}$$

For volume inhomogeneities of various dimensions, one can obtain (Chuprov and Shvachko 1984, Brekhovskikh and Lysanov 1991) that the total coefficient of volume scattering depends on frequency according to a power law with an exponent varying from two to four, depending on the characteristic size of the inhomogeneities, i.e., $\hat{\beta}_v \sim f^2$ to f^4. This implies for the bottom model that

$$\alpha_v = \alpha_v^0 (f/f_0)^{b-1} \tag{7.36}$$

where $b = 2$ to 4. Notice that after reconstruction of the bottom parameters according to the value of b, it is possible to qualitatively evaluate the size of inhomogeneities that make the greatest contribution to the losses caused by the volume scattering of sound.

In doing this inverse procedure, the specific values of α_v^0 and b are assumed to be such that (7.35) closely describes the frequency dependence of the signal in the band under consideration. In this case, the parameters and b are, generally speaking, different for the same bottom but in different frequency bands, because the variation of the frequency changes the relationship between the dimensions of the inhomogeneities and the wavelength, which changes the characteristics of the scattering.

Thus, assuming that the losses in the bottom are caused mainly by the two indicated mechanisms, we have for the frequency dependence of the coefficient α:

$$\alpha = \alpha_a + \alpha_v = \alpha_a + \alpha_v^0 (f/f_0)^{b-1}. \tag{7.37}$$

The coefficients in (7.33) are calculated by examining the experimental interference pattern of the acoustic field in the frequency domain. The basis for this calculation is the following expression for the modulus of the sound field received at a point with the coordinates (r, z) and produced by a tonal source located at the point $(0, z_0)$:

$$|S_{\text{th}}(r, z; \omega)| = C|\Psi(r, z, 0, z_0; \omega)| \tag{7.38}$$

where

$$\Psi(r, z, 0; z_0, f) \approx +\sqrt{\frac{i}{8\pi}} \sum_n \Psi_n(z_0)\Psi_n(z) \frac{\exp(\mathrm{i}\xi_n r)}{\sqrt{\xi_n r}} \tag{7.39}$$

Here, the coefficient C characterizes the spectral power $W(f)$ and the initial phase of the radiator, $C = \sqrt{8\pi\rho c(z_0)W(f)}$. We note that the frequency dependence of the received signal in (7.39) is determined by all the terms on the right-hand side of the equation, because the eigenfunctions $\Psi_n(z)$ and the eigenvalues ξ_n all depend on frequency.

So using (7.38) at fixed positions for the source and receiver and assuming that the eigenfunctions and eigenvalues depend on frequency as a parameter, we

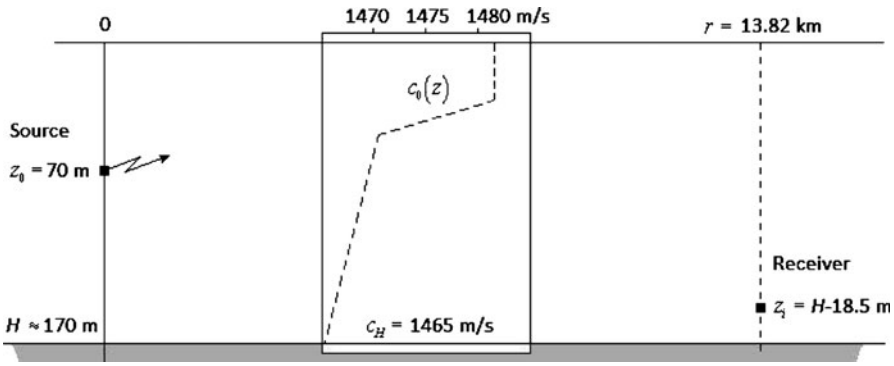


Fig. 7.10 Model of shallow water waveguide and experimental conditions

calculate a set of dependences $|S_{\text{th}}(r, z; \omega)|$ (i.e., we determine the interference pattern of the sound field in the frequency domain) for different values of the coefficients in (7.36). The $c(z)$ profile measured in the experiment (Fig. 7.10) is used in these calculations. The interference pattern of the sound field measured in the same experiment at a stationary track in the Barents Sea was used for the experimental dependence $|S_{\text{ex}}(r, z; \omega)|$. A list of the major parameters of the experiment is also shown in Fig. 7.10. The experiment was performed with the help of a piston-type radiator, which excited wide-band pulsed signals with a linear frequency modulation in the band 25–95 Hz. The duration of each signal was 40 s. The sound source was located at a depth of 70 m. The signal reception was performed by a single hydrophone at a depth of 151.5 m. The source and hydrophone were deployed from the two ships, each of which was braced between two anchored mooring buoys. The distance between the source and the receiver was 13.82 km. The sea depth varied weakly along the path, and was equal to $H \sim 170$ m. The main experimental data set used for the bottom parameter reconstruction was the spectrum of the received signal $|S_{\text{ex}}(r, z; \omega)|$ (one example of which is shown in Fig. 7.11a) as averaged in frequency bins of 5 Hz (shown in Fig. 7.11b).

This spectrum represented the interference pattern of the sound field in the frequency domain at the observation point. It should be noted that, after averaging, the signal spectrum did not depend on time; i.e., it did not change from pulse to pulse during the entire period of measurement. In other words, the averaged interference pattern obtained in this manner did not depend on the fluctuations of the waveguide parameters caused by the mesoscale variability of the medium (tides and internal waves), and so could be used for the determination of the acoustic characteristics of the sea floor. This is why the $|S_{\text{th}}(r, z; \omega)|$ were also averaged in frequency over a 5 Hz band.

In terms of our idea of obtaining the best fit between the vector $\hat{d}$ (the set of $|S_{\text{ex}}(r, z; \omega)|$ for different frequencies) and the vector $\hat{B}(\hat{m})$ which for our case is the set of theoretical data $|S_{\text{th}}(r, z; \omega)||$, we calculate the rms deviation between them.

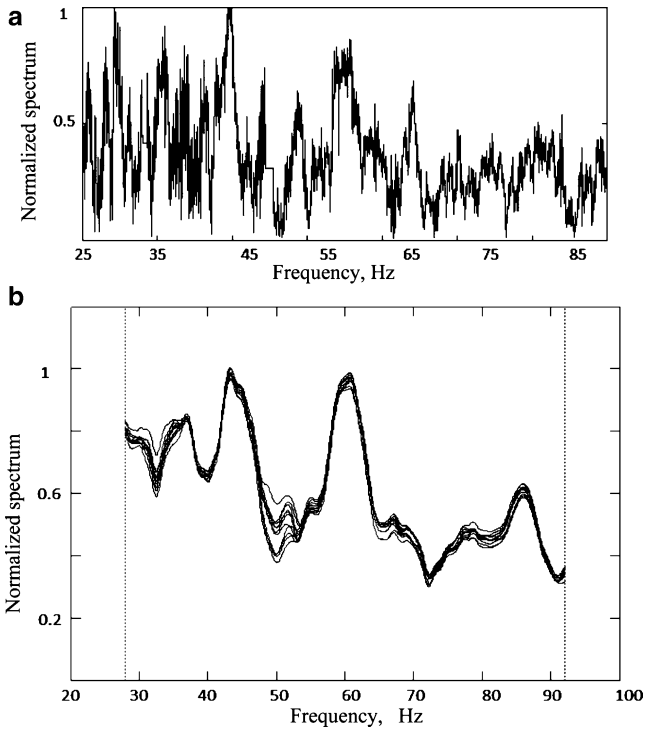


Fig. 7.11 (**a**) One example of the experimental spectrum. (**b**) 10 examples from sets of averaged spectra

The rms deviation $\tilde{\sigma}$ of the theoretical dependence from the experimental one is given by the following expression:

$$\tilde{\sigma} = \frac{1}{M}\sqrt{\sum_{j=1}^{M}\left[\left|S_{\mathrm{e}}(r,z;\omega_j)\right| - \left|S_{\mathrm{th}}(r,z;\omega_j)\right|\right]^2}. \tag{7.40}$$

Here M is the number of frequency bins of the sound field amplitude. The minimum $\tilde{\sigma}$ (ideally, equal to zero) corresponds to the best choice of the bottom model, and the parameters of the latter can be considered close to reality. We take the quantities κ, α_a, α_v^0, and b as the unknown characteristics of the bottom and assume that the velocity and density in the bottom are defined through the porosity by (7.33). As to the connection of porosity with the absorption coefficient defined by (7.34), the absorption calculated according to the averaged equation (7.35) can differ widely from real data corresponding to a specific region because of the large scatter of experimental data obtained in different regions of the ocean (see Fig. 7.9). Therefore, we assume that the coefficient α_a must be determined independently, along

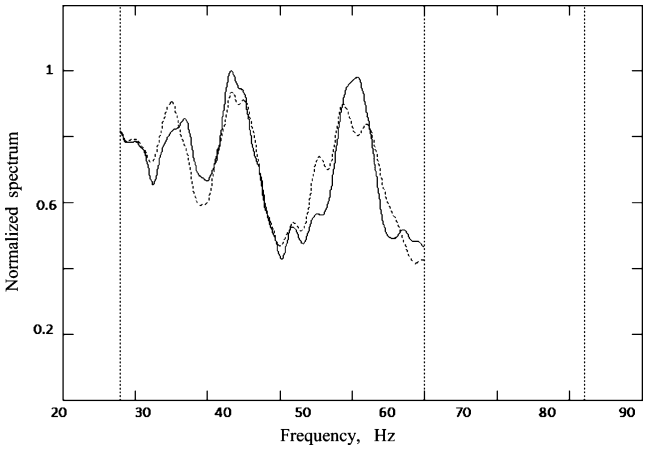


Fig. 7.12 Averaged experimental spectrum (*solid line*) and theoretical spectrum (*dotted line*) calculated for optimal bottom parameters. Both spectra are averaged using a sliding 5 Hz window

with the porosity and the parameters of volume scattering. In this case, the complete list of values upon which the theoretical spectrum depends is as follows: z_0, z, r, $c_0(z)$, c_H, H, κ, α_a, α_v^0, b. This set of parameters can be considered as a data vector $\hat{d}$. The values of the first six parameters can be measured directly, while the last four parameters are determined from the matching conditions. An essential consideration for the matching procedure used in this study is the effect of the variation of single parameters on the spectrum. For example, the variation of the porosity κ leads to a shift of the spectral pattern as a whole in frequency. The other three parameters (α_a, α_v^0, b) mainly influence the energy characteristics of the spectrum, but do not cause a shift in frequency. Thus, in the first stage of the procedure, we determine the value of porosity in such way that the interference peaks of the theoretical spectra coincide in frequency with the corresponding experimental peaks. Confining ourselves to the interval of realistic values, the unknown value of the ground porosity lies within the limits 0.3–0.4. The next stage in the inverse is a search for the optimal absorption parameters. The optimal value of the power index is sought in the region $b = 2$ to 4, together with the determination of the coefficients α_a and α_v^0 by an exhaustive search.

As calculations show, the best coincidence of the theory and experiment is observed in the 28–70 Hz frequency band with the effective value of porosity being equal to 0.3. Examples of spectra which correspond to the best fit are shown in Fig. 7.12.

According to the empirical formulae in (7.33), this corresponds to a relative density of $\rho_1/\rho_H = 2.1$ and to a sound velocity in the sea floor $c_1 = 1,765$ m/s. The values obtained for the specific coefficients α_a and α_v are as follows: $\alpha_a = 0.005$ and $\alpha_v = 9.3 \times 10^{-8} f^3$, and thus $\alpha = 0.005 + 9.3 \cdot 10^{-8} f^3$ ($\hat{\beta}_f[\mathrm{dB/km\,Hz}] = 0.084 +$

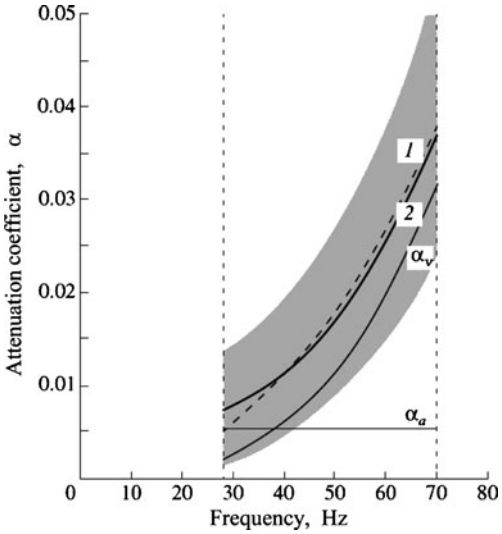


Fig. 7.13 Frequency dependences of the loss coefficients α. The *dashed line* 1 corresponds to the result of (Grigor'ev et al. 1996) $\alpha = 2.8 \times 10^{-6} f^{2.24}$ without separation of the loss mechanisms. The *thick line* 2 shows the total loss according to the results of this paper, $\alpha = \alpha_a + \alpha_v$. The *thin lines* represent the frequency dependences of the coefficients of absorption α_a and scattering α_v. The *gray area* shows the scatter of data due to the fluctuations of the experimental spectra

$1.6 \times 10^{-6} f^3$), where f is the frequency in Hz. The frequency dependences of (α_a, α_v, α) are displayed in Fig. 7.13.

Figure 7.13 also shows the boundaries of the spread in the values of the parameter $\tilde{\sigma}$ due to the random fluctuations of the received spectra. The curves shown also demonstrate the similarity of the results obtained by determining the frequency dependence of the absorption coefficient on the basis of the technique used here and the results presented in (Grigor'ev et al. 1996), where the same experimental data were used, but a simpler bottom model was used. Furthermore, we can make some conclusions about the mechanisms for losses in the sea floor. One can see from Fig. 7.13 (in particular) that, at frequencies lower than 40 Hz, the losses in the sea floor are caused mainly by absorption, whereas at frequencies higher than 40 Hz, the volume scattering dominates.

The numerical value of the sound absorption coefficient $\hat{\beta}_{af}$ is $\hat{\beta}_{af} = \sim 0.084$ dB/(km Hz). This value is much smaller than the one calculated by the empirical formula (7.34) (0.4 dB/(km Hz)), although it generally lies within the scatter of the data. It should be noted that other methods used for the determination of the absorption coefficient also lead to values much lower than those obtained from the empirical formula (7.34). Presumably, the explanation of this is connected with the specific types of sediment that occur in the Barents Sea. They consist mainly of highly porous clay mixed with stones. As a result, the average porosity is low (~0.3) because of the presence of the solid inclusions, and absorption is

determined mainly by the clay filler; i.e., it is characterized by the values corresponding to absorption for the case of porosity equal to 0.7–0.9. Turning to Fig. 7.9, we can conclude that in the Barents Sea the absorption coefficients for sediments with porosity ~0.3 (thin clay with stone inclusions) and ~0.8 (homogeneous clay, silt) can approximately coincide and are equal to ~0.1 dB/(km Hz). As one can see from the same figure, the scatter of experimental data at $\kappa \sim 0.3$ confirms the reasoning given above. At the same time, the wide difference between the minimum value of absorption and the average coefficient of absorption for the case of porosity equal to 0.3 demonstrates, in our opinion, the particular features of the sediment in the Barents Sea; in contrast for other regions of the ocean, the typical sediment has the form of coarse-grained sand with a porosity of 0.3. For this case, the sound absorption is 35 times greater than for homogeneous clay or, for the case under study, for the inhomogeneous sediment of the Barents Sea type. On the whole, we feel that the results of this study favor the use of the two-component model of the sea floor for describing the sound propagation in many regions of the Barents Sea, and in other similar situations.

In concluding of this section, it is interesting to present the results of measurements of many different authors, as recently compiled by Zhou et al. (2004), estimating the frequency dependence of bottom absorption coefficient. (This is similar to the compilation by Holmes et al. presented earlier.) In combining these results, the frequency dependence of the attenuation coefficient was approximated by power function $\alpha \sim f^b$. In Fig. 7.14, this dependence, obtained in experiments at 16 different shallow-water sites, is shown using a logarithmic scale; in other words, we can directly see values of the parameter b. As a result of averaging all the data in a frequency band up to 1 kHz, the regression indicates that the dependence can be approximated as $\alpha \sim 0.38 f^{1.84}$. This is in good agreement with both Jixun data and the Holmes et al. compilation. In closing, we will remark that in our theory, the value of b is different for different frequency bands so that one should take care in extrapolating such results.

7.6 Bottom Geoacoustic Inversions Using Ambient Noise Sources

In an era when the usage of active sound sources in the sea is being curtailed, often arbitrarily, by external regulation, one alternative to using such sources is to use the unregulated ambient sound sources that exist, be they natural or man-made. We will discuss three popular variants of this technique here, two of which use anthropogenic noise and one which uses naturally produced noise. Specifically, we will look at inversion for bottom properties from (1) self-noise from a ship or AUV that is towing a horizontal array, (2) noise from ships of opportunity, using single hydrophones and arrays, and (3) natural noise, using a vertical array receiver. We note that these inverses do not require a very deep understanding of the

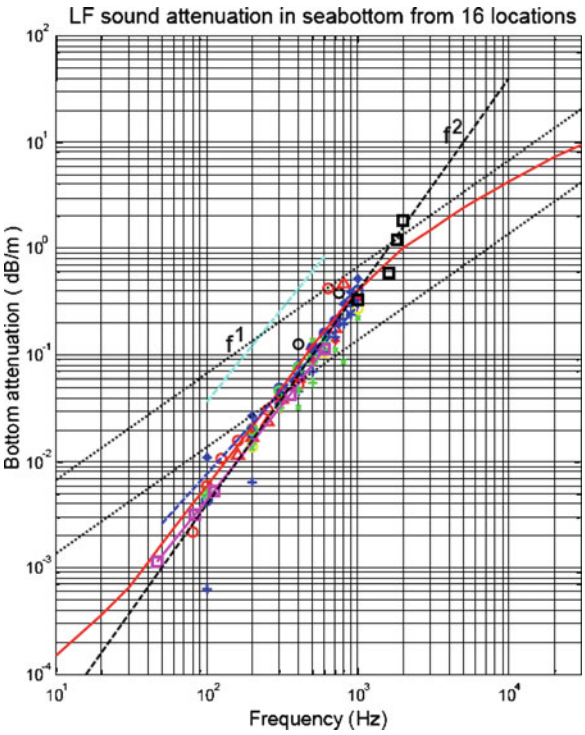


Fig. 7.14 Frequency dependence of absorption coefficient on a logarithmic scale

shallow-water noise field, so that we can look at them before going into the detailed physics of the noise in a later chapter.

7.6.1 Inversion for Bottom Properties from Ship Self-Noise

In Fig. 7.15, we show a diagrammatic view of the dominant ship noise arrivals that are seen by an array towed close behind a moving ship. These are the four arrivals of the first ray group, which intuitively are at the lowest grazing angles and contain the most energy. Two of the arrivals (direct and SR) have no interaction with the bottom, whereas the other two (BR and SBR) each have one bottom interaction. The addition of these four rays will cause an interference pattern in range, part of which will be seen along the length of the array. If the bottom were a perfect absorber, the pattern seen would simply be a segment of the well-known Lloyd's mirror. However, the addition of the bottom interacting paths makes this pattern more complex, and it is the specific shape of this complex pattern that contains information about the bottom. Depending on the type of sediment, the length of the array, and the distance of the array from the ship, the bottom interacting arrivals

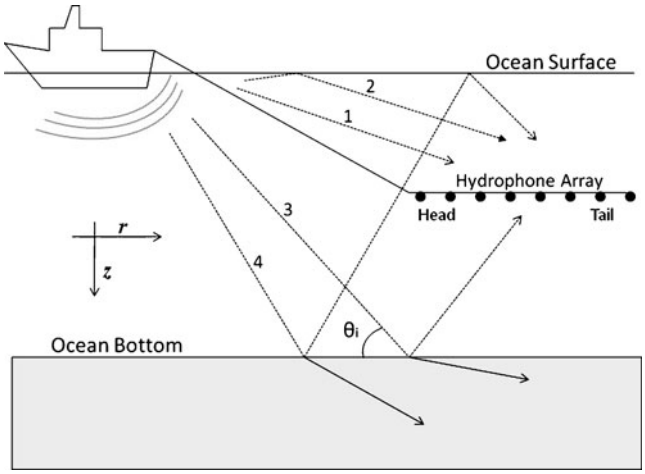


Fig. 7.15 Four dominant (ray) arrivals seen behind a ship towing a horizontal array receiver

may be above or below the critical grazing angle. For arrivals below the critical grazing angle, one has that the interference pattern is strongly affected by the bottom reflection coefficient phase, which has an almost linear dependence on θ_i with a gradient set by the critical angle (and thus the seabed sound speed, if one uses an isovelocity half space model, as we do here.) If the array is close to the ship or short or both, one or both of the bottom interacting arrivals can be above critical grazing angle, and for these rays the reflection coefficient phase is zero. In this case, the reflection coefficient amplitude controls the bottom arrival's contribution to the field pattern.

The key to the inverse from such data is that a particular pattern of amplitudes received along the array can be, in principle, associated with a unique value of the bottom half space velocity. Also, the bottom density, while affecting the near-field in absolute terms through the impedance contrast, does not significantly affect the interference field pattern along the array. Further, due to the closeness of the array to the ship, bottom attenuation is not a significant factor. Thus, one is poised to measure the bottom sound speed with such a system.

An example of the inversion of such data to determine the bottom sound speed is given in (Battle et al. 2003). In this paper, wave number integration is used to create a bottom model, as this technique is well adapted to creating the virtual modes needed to describe the bottom reflected energy above the critical grazing angle, and also because the environment can be treated as approximately range dependent over the short distances between source and receiver. The field so computed is used as a replica field to the data for matched field processing using a standard Bartlett processor, and the maximum beamformer output is used as a criterion for the best

match. As the bottom sound speed is a parameter in the bottom model, it is thus determined by the maximum output. There are, of course, other details that need to be considered in such an inverse (array shape, noise, etc.), but the discussion above gives the basic idea.

7.6.2 *Inversion for Bottom Properties from Random "Ship of Opportunity" Noise*

One of the most common types of noise encountered in shallow water is ship and boat noise. Almost all research recordings of sound are contaminated by such noise, an example of which is shown in Fig. 7.16a (with a computer model of it being shown in Fig. 7.16b). In Fig. 7.16, we see a time–frequency sonogram of the well known "bathtub rings" that are formed by the mutual interference of the broadband frequencies of a ship signature as the ship source approaches and then recedes from the receiver. These interference rings have information about the bottom in them, and again the question becomes: How do we extract the bottom information from such data? There are, in fact, a number of ways in which one can extract bottom information from the data presented in Fig. 7.17, and we will discuss a small (but important) subset of them here.

One of the most popular ways of looking at, and extracting information from, data such as are seen in Fig. 7.17, is the so-called "waveguide invariant," β, which we have discussed previously in our text. Let us have broadband noise signal from the ship, central frequency and width are f, Δf correspondingly. A useful representation of this invariant is presented in (Heaney 2004), which is

$$\frac{\Delta f/f}{\Delta r/r} = \frac{\frac{1}{c_0^{\mathrm{ph}}} - \frac{1}{c_{\max}^{\mathrm{ph}}}}{\frac{1}{c_0^{\mathrm{gr}}} - \frac{1}{c_{\max}^{\mathrm{gr}}}} = \beta \tag{7.41}$$

where c_0^{ph}, $c_{\max}^{\mathrm{ph}}$ and c_0^{gr}, $c_{\max}^{\mathrm{gr}}$ are minimal and maximal values of the phase and group velocities in given channel.

In the Pekeris waveguide, $\beta = \cos^2\vartheta_{\mathrm{critical}} \sim 1$, so that one may relate the slope of the interfering intensity fringes to the critical angle, and thus (in the context of the Pekeris model bottom half space) the bottom sound speed. Of course, one immediately notices that (1) has source range rather than clock time as a variable, and so it is obvious that one experimental task is to relate the range to time, i.e., determine $r(t)$. We will assume that this can be done by some ranging device, which could include radar, acoustic triangulation, acoustic focusing, or others.

Having used the slope of the striations as the first variable, one can choose either the spacing of the fringes in range or in frequency as the second. The frequency

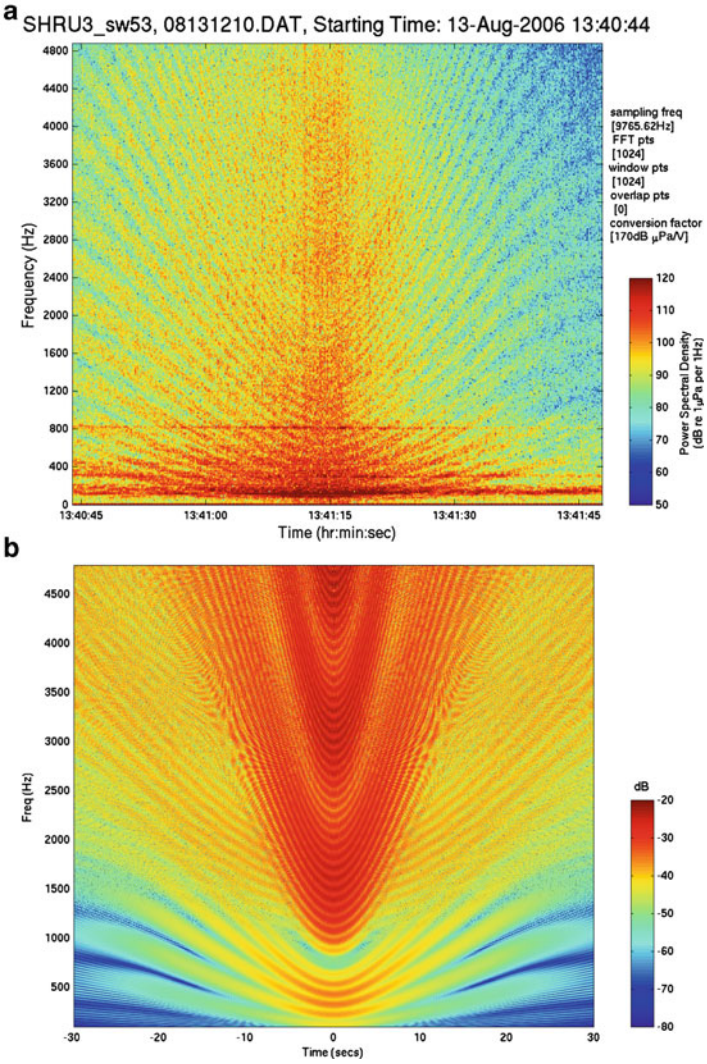


Fig. 7.16 (**a**) "Bathtub ring" ship noise pattern seen on the bottom hydrophone during the Shallow Water 2006 (SW06) experiment. (**b**) "Bathtub ring" ship noise pattern calculated for the (**a**) bottom hydrophone

spread at a given range is perhaps the simplest physical choice, as by using the time-bandwidth relation $\Delta f = \frac{1}{\Delta t}$ it represents the time spacing between broadband ray arrivals for a source at the given range, a familiar picture. This spacing of the ray arrivals is determined by the waveguide depth, which for a perfectly reflecting

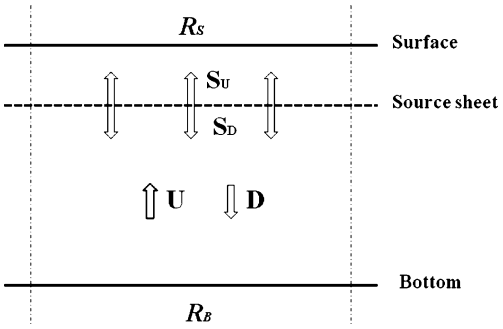


Fig. 7.17 Up and down going energy fluxes, used to determine the bottom reflection coefficient

bottom would just be the water depth. However, the energy for a real bottom penetrates into the bottom, so that the depth given by the spacing of real arrivals has both the water depth *and* the depth of penetration of the sound into the bottom included into it. As this effective depth of penetration depends on the material properties of the sediment, it is data for an inverse for bottom geoacoustic properties.

A final piece of data that is available from the ring interference pattern is the slope of the incoherent transmission loss versus range. For many shallow-water environments, the transmission loss (TL) can be approximated as

$$\mathrm{TL} = A + B\lg(r) - ar. \tag{7.42}$$

The constant A is irrelevant for relative TL (i.e., slope), and if we assume cylindrical spreading, $B = 10$, which allows that term to be subtracted out from the data. This just leaves us with the αr term. For sufficiently long ranges, this term can be estimated reasonably, but as α is a small number overall, one must be careful in estimating it.

Heaney and Cox (2002) and Heaney (2004) have used the above data in an inverse scheme that they call "Rapid Geoacoustic Characterization" in which one makes a best fit to the data using a bottom parameterization based on the work of Hamilton (1985) and Bachman (1985). This makes sense in the context of a fluid medium, since the above parameters have sensitivity to the compressional wave sound speed and attenuation, and the density as well. The parameterized model, in turn, extrapolates the data further into the bottom than the inverse can resolve or sense.

There are other, somewhat more elaborate, ways of using ship-of-opportunity data than the simple method we have described above. We will just refer the reader to the literature for these, noting the works by Koch and Knobles (2004), Chapman et al. (2006), and Turgut et al. (2010) as being representative.

7.6.3 *Inversion for Bottom Properties from Random Surface Noise*

Use of surface wind/wave noise to get at bottom properties is another inverse problem that has received a fair share of attention from the shallow-water acoustics community. A very simple, but compelling and physically insightful, argument of how one might accomplish this is given in (Harrison and Simons 2002). In Fig. 7.17, we show a diagram, based on that paper, of the up and down going plane wave fluxes that are created by a planar sheet of sources somewhere below the sea surface. The sources create plane wave fluxes S_U and S_D in the up and down directions. (The horizontal propagation can be ignored in this argument.) The surface and bottom have power reflection coefficients V_S and V_B respectively. The net fluxes U and D are to be determined. Just below the source, the total downward flux consists both of the direct source component and a component of the energy reflected back downwards from the surface, i.e., $D = S_D + V_S(S_U + U)$. And by the definition of the bottom power reflection coefficient, we have $U = DR_B$. If we now put a receiver in this field, we can measure both U and D, and thus infer R_B. And since the original field consisted of plane waves, it also means we can identify R_B as the plane wave reflection coefficient! So the experimental task becomes resolving upgoing versus downgoing energy, one that has been successfully addressed by steered vertical arrays, and which is nothing new for the underwater sound community. Both modeling and real data demonstrations of the efficacy of this technique are presented in the Harrison and Simon paper, and it must be said that in the authors' opinions this is perhaps one of the simplest, yet most elegant, methods for inferring bottom information that has been yet devised.

A very exciting and promising extension of this technique has been made by Siderius et al. (2006), who built on a conjecture by Rickett and Claerbout: "By cross correlating noise traces recorded at two locations at the surface, we can construct the wave field that would be recorded at one of the locations if there were a source at the other." Given sufficient time averaging, this cross correlation of noise produces the eigenray arrivals between the two hydrophones. If one considers two (or more) vertically separated hydrophones, one should see that correlating the surface noise will give a "passive fathometer" that sees the eigenray arrivals from both the bottom and the prominent subbottom layers. Siderius and his collaborators have shown that such a fathometer can in fact out-perform some standard fathometers and, in a drifting configuration, provide beautiful pictures of the range-dependent layering of the ocean bottom.

Chapter 8
Signal Processing

Signal processing is another huge technical area that interacts with shallow water acoustics. Thus, we will only consider here those methods which are directly applicable to shallow water acoustics. We will assume the reader is familiar with the basic texts in this area.

In this chapter, we will begin by considering methods of acoustical data processing. Specifically, we will analyze experimental data obtained by different devices (vertical and horizontal arrays, single hydrophones, etc.) called the "raw data." We first consider how to convert these data to forms appropriate for theoretical analysis, to facilitate comparison with the results of calculations. This is comprised by the more or less standard techniques of sampling and digitization used to get the spectra and correlation and coherence functions of the received signals. Also, these "raw data" can be used directly for the determination of the parameters of the environment (so-called matched field processing or MFP) or for the extraction of more detailed spatial information like the modal decomposition (mode filtering). Even more sophisticated methods of data processing can be used for the control of the sound field in the waveguide, as for instance, methods of focusing the sound signals (e.g., the time reversal mirror).

8.1 Fundamentals of Data Processing Techniques

The spatial processing of acoustic signals measured in field experiments and the creation of special acoustic signals to be radiated for specific purposes on the sea shelf are just two examples out of many of signal processing for shallow water acoustics. In doing such processing, it is important first of all to determine the commonly used characteristics of the acoustic signal such as its amplitude, phase, spectral density, etc. Shallow water signal processing often deals with random signals, as the signal from a remote sound source passes through a randomly inhomogeneous

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_8,
© Springer Science+Business Media, LLC 2012

medium in field experiments and is detected against an acoustic noise background, not to mention the case when the acoustic noise itself is the object of study. Techniques for measuring these signal characteristics, including techniques for estimating measurement accuracy as a whole, are well known and similar to measurement techniques in other fields of experimental physics. Techniques are described in many monographs [see, e.g., Bendat and Piersol (1986); Max (1981)], to which we refer our readers. In this book, we will briefly consider some basic relations commonly used when processing such acoustic data.

First, we note that digital signal processing is used in the vast majority of cases. In this type of processing, the first step is digitization of the continuous electrical signal $u(t)$ coming from a receiving hydrophone. As a result, the signal is observed periodically at strictly defined time points, rather than continuously. (In practice, a standard processing device, the analog-to-digital converter (ADC), transforms electric voltage to a numerical value.) The interval T_e between time points is called the digitization (or discretization) period and correspondingly, the quantity $F_e = 1/T_e$ is called the digitization (or discretization) frequency. In the case of ideal discretization, the signal observation time is infinitesimal, i.e., discretization is carried out using infinitely short pulses, whose set forms the so-called comb function.

$$D(t) \equiv \sum_{n=-\infty}^{\infty} \delta(t - n/F_e), \tag{8.1}$$

where $\delta(t)$ is the Dirac function. In this case, the signal $\hat{u}(t)$ coming from the ADC output can be written as $\hat{u}(t) = u(t)D(t)$. According to the convolution theorem [see, e.g., Max (1981)] the spectrum of the signal $\hat{u}(t)$ is given by

$$S_{\hat{u}}(\omega) = \int_{-\infty}^{\infty} S_u(\omega')S_D(\omega - \omega')d\omega', \tag{8.2}$$

where $S_u(\omega)$ and $S_D(\omega)$ are the spectra of signals $u(t)$ and $D(t)$, respectively. We can get an expression for the spectrum $S_D(\omega)$ if we use the well-known relationship

$$\sum_{n=-\infty}^{\infty} \delta(t - n/F_e) = F_e \sum_{n=-\infty}^{\infty} \mathrm{e}^{-i2\pi F_e nt}$$

and substitute it into formula (A1), i.e.,

$$S_D(\omega) = F_e \sum_{n=-\infty}^{\infty} \delta(\omega - 2\pi n F_e). \tag{8.3a}$$

Then, substituting (8.3a) into (8.2), we obtain:

$$S_{\hat{u}}(\omega) = F_e \sum_{n=-\infty}^{\infty} S_u(\omega - 2\pi n F_e). \tag{8.3b}$$

As can be seen from (8.3), as a condition we must have

$$F_e \geq 2F_{\mathrm{m}} \tag{8.4}$$

where F_{m} is the maximum frequency in the signal spectrum of $u(t)$, i.e., the spectrum $S_u(\omega)$ lies in the frequency range $[-F_{\mathrm{m}}, F_{\mathrm{m}}]$. The spectrum of $\hat{u}(t)$ is thus a periodically recurring spectrum, $S_u(\omega)$. This means that knowing the spectrum $S_{\hat{u}}(\omega)$, we can uniquely reconstruct both the spectrum $S_u(\omega)$ itself and the signal $u(t)$. Thus, since condition (8.4) is satisfied, we can uniquely reconstruct the function $u(t)$ by its discrete values $u(n/F_e)$ $(n = 0, \pm 1, ..., \pm\infty)$. The latter statement is in fact the well-known Shannon theorem,[1] and (8.4), which defines the digitization frequency, is one of the basic relations in digital signal processing. In the literature, the frequency $F_e/2$ is usually given the name Nyquist frequency.

For unique reconstruction of the signal by a set of its discrete values at the ADC output, the Shannon interpolation formula:

$$u(t) = \sum_{n=-\infty}^{\infty} u(n/F_e) \frac{\sin \pi F_e[t - (n/F_e)]}{\pi F_e[t - (n/F_e)]}, \tag{8.5}$$

can be used, whose detailed derivation is given, e.g., in Max (1981). The use of ordinary linear interpolation between discrete $u(n/F_e)$ values can cause appreciable errors, especially if the digitization frequency only slightly exceeds the value of $2F_{\mathrm{m}}$. Sufficiently accurate signal reconstruction using linear interpolation generally requires an increase in the digitization frequency. For example, to reconstruct a sinusoidal signal with an error no more than ε, the digitization frequency should satisfy the inequality $F_e \geq f(2.2/\sqrt{\varepsilon})$ (Max 1981). Here f is the signal frequency, $\varepsilon = (\max(u(t)) - u_1(t))/\max(u(t))$, and $u_1(t)$ is the reconstructed signal. For $\varepsilon = 0.01$, $F_e \geq 22f$, i.e., the discretization frequency more than ten times exceeds its minimum possible value as defined by the inequality (8.4).

In practice, signals are digitized using a train of finite-duration pulses, which, strictly speaking, differ from the comb function $D(t)$. However, when processing the acoustic data we are concerned within this book, rather slow functions of time are used. As a result of this processing, the inequality:

$$t_{\mathrm{d}} \ll \frac{1}{F_e} \leq \frac{1}{2F_{\mathrm{m}}} \tag{8.6}$$

is valid, where t_{d} is the digitization pulse duration, which typically does not exceed several microseconds for modern ADCs. When this inequality is satisfied, the error associated with the practical use of (8.5) is negligible (Max 1981).

[1] In the Russian literature, this theorem is referred to as the Kotelnikov theorem.

When collecting and processing acoustic data, another practical problem arises; this is associated with large amounts of data collected during multiday observations using autonomous receiving systems. Even with the terabyte level computer storage available at the time this is being written, multichannel acoustic array data sets can quickly overwhelm computer memory. This drawback is sometimes mitigated, however, in the circumstance where narrow-band signals are used in acoustics experiments. These can be written as

$$u(t) = A(t)\cos(\omega t + \varphi(t)),$$

where $A(t)$ and $\varphi(t)$ are slow functions of time, and the carrier frequency, ω, is accurately known. In this case, the signal is passed through a synchronous detector before digitization which separates the sine

$$p_{\sin}(t) = \langle u(t)\sin(\omega t + \varphi_0)\rangle = -A(t)\sin(\Delta\varphi(t))$$

and cosine

$$p_{\cos}(t) = \langle u(t)\cos(\omega t + \varphi_0)\rangle = A(t)\cos(\Delta\varphi(t))$$

quadrature components of the signal. Here, the angle brackets indicate low-frequency filtering of the bracketed signal or, which is almost the same, averaging of the signal over a time much longer than $2\pi/\omega$. $\Delta\varphi(t)$ is the difference between the phases of the received and reference signals ($\Delta\varphi(t) = \varphi(t) - \varphi_0$). It is clear that digitization of the quadrature components allows the use of much smaller frequencies for F_e and thus decreases the collected data set. At the same time, passing to the quadrature components makes it easy to determine the signal amplitude and phase:

$$A(t) = \sqrt{p_{\sin}^2(t) + p_{\cos}^2(t)}, \tag{8.7}$$

$$\Delta\varphi(t) = \arctan\left(\frac{p_{\sin}(t)}{p_{\cos}(t)}\right). \tag{8.8}$$

It should be emphasized that the quadrature components carry all the information on the received signal contained in the frequency band $\Delta\omega$. (We recall that $\Delta\omega$ is the signal spectral width for $u(t)$.)

Another important signal processing concern is with achieving optimum signal separation against the noise background. Since this topic is extremely important, we will give a brief derivation [For more details see, e.g., Gonorovskii (1977)]. Let's assume a signal with the spectrum $S(\omega)$ and noise with a spectral density function $G(\omega)$ [see Appendix A, (A.8)]. We can pose the problem of optimum signal separation against a noise background as the construction of a special filter with a transfer function $K(\omega)$ which provides the highest signal-to-noise ratio (SNR). To search for

the optimum transfer function, we shall initially construct independent expressions for the signal and noise, and then provide an expression for their ratio. Let us determine the peak signal observed at the instant t_0 at the filter output using an expression well known from circuit theory:

$$u_{\text{out}}(t_0) = \left| \int_{-\infty}^{\infty} S(\omega)K(\omega)\mathrm{e}^{-\mathrm{i}\omega t_0} d\omega \right|. \tag{8.9}$$

Under the assumption that the noise is white with spectral density $G(\omega) = G_0$, the root-mean-square value of the interference at the filter output is given by:

$$\sigma = \sqrt{\int_{-\infty}^{\infty} G(\omega)|K(\omega)|^2 d\omega} = \sqrt{G_0 \int_{-\infty}^{\infty} |K(\omega)|^2 d\omega}. \tag{8.10}$$

Hence, the ratio of the peak signal to the root-mean-square value of the interference at the filter output is given by

$$\frac{u_{\text{out}}(t_0)}{\sigma} = \frac{\left| \int_{-\infty}^{\infty} S(\omega)K(\omega)\mathrm{e}^{-\mathrm{i}\omega t_0} d\omega \right|}{\sqrt{G_0 \int_{-\infty}^{\infty} |K(\omega)|^2 d\omega}}. \tag{8.11}$$

We then invoke the Schwartz inequality

$$\left| \int_a^b F_1(x)F_2(x)dx \right|^2 \leq \int_a^b |F_1(x)|^2 dx \int_a^b |F_2(x)|^2 dx \tag{8.12}$$

where $F_1(x)$ and $F_2(x)$ are complex functions in general. This inequality becomes an equality only when the condition:

$$F_2(x) = CF_1^*(x), \tag{8.13}$$

is satisfied, where C is any constant factor.

Using (8.12), we can pass from the equality (8.11) to the inequality

$$\frac{u_{\text{out}}(t_0)}{\sigma} \leq \frac{\left[\int_{-\infty}^{\infty} |S(\omega)|^2 d\omega \int_{-\infty}^{\infty} |K(\omega)|^2 d\omega \right]^{\frac{1}{2}}}{\left[G_0 \int_{-\infty}^{\infty} |K(\omega)|^2 d\omega \right]^{\frac{1}{2}}} = \frac{1}{\sqrt{G_0}} \left[\int_{-\infty}^{\infty} |S(\omega)|^2 d\omega \right]^{\frac{1}{2}}. \tag{8.14}$$

Taking into account that the bracketed expression on the far right-hand side is simply nothing but the total energy E of the input signal, we write the latter expression as:

$$\frac{u_{\text{out}}(t_0)}{\sigma} \leq \sqrt{\frac{E}{G_0}}. \tag{8.15}$$

Finally, it follows from expression (8.13) that this inequality becomes an equality under the condition:

$$K(\omega)\mathrm{e}^{-\mathrm{i}\omega t_0} = CS^*(\omega). \tag{8.16}$$

It follows that an optimum filter providing the largest S/N ratio should have the following transfer function:

$$K(\omega) = CS^*(\omega)\mathrm{e}^{i\omega t_0}. \tag{8.17}$$

Thus, according to (8.9), the output signal of such a filter is given by:

$$u_{\text{out}}(t) = \int_{-\infty}^{\infty} S(\omega)K(\omega)\mathrm{e}^{-i\omega t}d\omega = C\int_{-\infty}^{\infty} |S(\omega)|^2\mathrm{e}^{-i\omega(t-t_0)}d\omega. \tag{8.18}$$

Expressions (8.17) and (8.18) are very important relations for processing acoustic signals. We note also that the expression on the right-hand side of (8.18) is the autocorrelation function of the input signal received at the instant t_0. This means that if a correlation receiver calculating the correlation function between the input and reference signals is used as a device for processing, such a device becomes optimum (i.e., provides the maximum SNR), when an input signal copy in the absence of noise is used as a reference signal.

To estimate the reliability of measurements of acoustic signals received in field experiments, well-known relations of mathematical statistics are invoked [see, e.g., Bendat and Piersol (1986)]. In what follows, we will present estimates of the reliability of measurements of the mean value μ_x and of the variance σ_x^2 of the random variable x, which are a result of such measurements. The definition of μ_x and σ_x are given by

$$\mu_x \equiv E(x) = \int_{-\infty}^{\infty} x\hat{w}(x)dx, \tag{8.19}$$

$$\sigma_x^2 = \int_{-\infty}^{\infty} (x-\mu_x)^2\hat{w}(x)dx, \tag{8.20}$$

where $\hat{w}(x)$ is the probability density of the random variable x. If x is understood to be the acoustic signal[2] $u(t)$, which is received in a field experiment at time points $t_n = \hat{t}_0 + nT_r$ ($\hat{t}_0$ is the measurement start time and nT_r is an integer multiple of the emitted signal period $n = 1, 2, 3, ..., \infty$), this quantity is as a rule normally distributed and the function $\hat{w}(x)$ is given by

$$\hat{w}(x) = \left(\sigma_x\sqrt{2\pi}\right)^{-1} \exp\left[-\frac{(x-\mu_x)^2}{2\sigma_x^2}\right]. \tag{8.21}$$

The latter statement follows from the central limit theorem, according to which the distribution of the normalized sum of independent random variables is asymptotically normal. We note that exactly such a case takes place in shallow water acoustics, since the received sound signal represents a sum of the signal passed through a randomly inhomogeneous medium with uncorrelated perturbations and natural acoustic noise. This noise, as shown in Sec. Ch. 9 of this book, is combined from noise signals from many uncorrelated sources.

Since the μ_x and σ_x values are as a rule unknown, they are estimated using the sample mean:

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{8.22}$$

and the sample variance:

$$s_x^2 = \frac{1}{N-1}\sum_{n=1}^{N} (x_n - \bar{x})^2 \tag{8.23}$$

where the x_n are results of single instantaneous measurements of the x value and N is the number of measurements. Under the assumption that the random process $x(t)$ is stationary, $\bar{x}$ and s_x^2 converge to μ_x and σ_x^2, respectively, for an infinite number of measurements. If the number of measurements is limited, a confidence interval including the μ_x and σ_x^2 values can be determined with a set probability using the $\bar{x}$ and s^2 values. In particular, for μ_x, such an interval is written as

$$\bar{x} - \frac{s_x}{\sqrt{N}}\tilde{t}_{\hat{w}} < \mu_x < \bar{x} + \frac{s_x}{\sqrt{N}}\tilde{t}_{\hat{w}}, \tag{8.24}$$

where $\tilde{t}_{\hat{w}}$ is the fractile of the well-known Student distribution. We recall that the fractile is a $\tilde{t}_{\hat{w}}$ value, below which all values of the random variable $\tilde{t}$ obey the Student distribution with a set parameter N with the probability $\hat{w}$. Thus, inequality (8.24)

[2] The quadrature components $p_{\sin}(t)$ and $p_{\cos}(t)$ of the signal can also be taken as this random variable.

defines a confidence interval including the true value of μ_x with the probability $\hat{w}$. We note that, assuming that x is normally distributed, it is valid to use normal distribution fractiles smaller than $\tilde{t}_{\hat{w}}$ in expression (8.24).

For σ_x^2, a similar confidence interval can be written as

$$\frac{N}{\chi^2_{\hat{w}_2}} s_x^2 < \sigma_x^2 < \frac{N}{\chi^2_{\hat{w}_1}} s_x^2 \tag{8.25}$$

where $\chi^2_{\hat{w}}$ is the fractile of another known distribution, the Chi-square distribution, which like the Student distribution, depends on N. In other words, the value σ_x^2 lies in the interval of values defined by inequality (8.25) with the probability $w = w_2 - w_1$. The derivation of inequalities (8.24) and (8.25) is given in standard monographs devoted to the analysis of experimental data [see, e.g., Bendat and Piersol (1986)].

We note that if the random variable x is understood as the measured amplitude of the received signal (8.7) at time points t_n (although, strictly speaking, this quantity obeys the generalized Rayleigh distribution), such a distribution is close to normal at large ratios $\mu_x/\sigma_x \gg 1$. Hence, to estimate the measurement accuracy for the mean signal amplitude and its variance, inequalities (8.24) and (8.25) can be used.

The close-to-normal distribution at $\mu_x/\sigma_x \gg 1$ (here x corresponds to the signal amplitude) is also valid for the phase φ' of the quadrature components of the signal [see (8.8)] under the additional condition that the mean value of the random variable φ' is close to zero. The latter condition is easily satisfied by a change in the phase φ_0 of the reference signal of the synchronous detector. Under these conditions, we again can apply inequalities (8.24) and (8.25) to estimate the measurement accuracy of the statistical characteristics of the received signal phase.

A similar approach is used to estimate the measurement accuracy of the spectral density $G_0(\omega)$ of a random acoustic signal. It is conventionally assumed that $G_0(\omega)$ is nonzero only for positive frequencies $(\omega \geq 0)$ and is given by

$$G_0(\omega) = \begin{cases} 2G(\omega), & \omega \geq 0 \\ 0, & \omega < 0 \end{cases}, \tag{8.26}$$

where $G(\omega)$ is defined by (A.14) and (A.15). It is also commonly assumed that the input signal is a stationary and ergodic random process, for which the following relation is valid

$$G_0(\omega) = 2 \lim_{T_\mathrm{d} \to \infty} \frac{1}{T_\mathrm{d}} E\left[|S(\omega, T_\mathrm{d})|^2 \right], \tag{8.27}$$

where $S(\omega, T_\mathrm{d})$ is the spectrum of the experimentally received acoustic signal, calculated using the finite Fourier transform

$$S(\omega, T_\mathrm{d}) = \frac{1}{2\pi} \int_0^{T_\mathrm{d}} u(t) \mathrm{e}^{i\omega t} d\omega \tag{8.28}$$

where T_d is the realization duration.

The spectral density of the experimentally measured signal is conventionally understood as average values of the random variable

$$\tilde{G}_0(\omega) = \frac{2}{T_d}|S(\omega, T_d)|^2, \tag{8.29}$$

defined as

$$\bar{G}_0(\omega) = \frac{2}{NT_d}\sum_n^N |S_n(\omega, T_d)|^2. \tag{8.30}$$

Here N is the number of nonoverlapping (in time) realizations of the signal of duration T_d. We note that, in contrast to the cases considered earlier, here we deal with the multidimensional random variable $\tilde{G}_0(\omega)$ (its dimension is the product $\Delta f T_d$, where Δf is the signal bandwidth), which does not converge to the true spectral density $G_0(\omega)$ even for very large T_d.

To a first approximation, to estimate the measurement accuracy of the spectral density, an inequality similar to (8.25) can be used (Bendat and Piersol 1986),

$$\frac{2N}{\chi^2_{\hat{w}_2}}\bar{G}_0(\omega) < G_0(\omega) < \frac{2N}{\chi^2_{\hat{w}_1}}\bar{G}_0(\omega). \tag{8.31}$$

Thus, true values of the spectral density lie in the interval defined by inequality (8.31) with the probability $w = w_2 - w_1$. We note that inequality (8.31) differs from (8.25) by a factor of 2 in its left- and right-hand parts. The reason is that summation of two random independent multidimensional variables takes place when calculating the squared spectra magnitudes on the right-hand side of (8.30), i.e., the real and imaginary parts of spectra $|S_n(\omega)|^2 = (\mathrm{Re}(S_n(\omega)))^2 + (\mathrm{Im}(S_n(\omega)))^2$. As a result, the number of statistical degrees of freedom increases twofold. It should be also emphasized that the confidence interval defined by relation (8.31) increases slightly depending on detailed features of the spectral analysis. For example, if each signal realization is smoothed by a special time window (e.g., the Hann window), in such an analysis the confidence interval is increased by a factor of $\sqrt{2}$. This time window is used to diminish the systematic error in the spectrum calculation, associated with energy leakage from adjacent frequencies (see Bendat and Piersol (1986) for more detail).

8.2 Matched Field Processing in Shallow Water

Traditionally, processing of the sound field on the basis of the plane wave approximation works well in free space or in situations where a homogeneous space might be a good approximation to the environment. In reality, sound propagation in the

ocean waveguide, especially in shallow water, cannot be taken as having plane wave characteristics. However, in this case there is method of optimal signal processing that can use and exploit the realistic structure of the sound field. This technique is called matched field processing (MFP).

MFP involves the correlation of the actual acoustic pressure field measured at the receiver array with a predicted field based on a postulated source position and (or) assumed ocean model. A high degree of correlation or "match" between the measured field and the predicted field indicates likely source localization or estimates of the parameters of the waveguide. Thus, an increased complexity of the ocean structure provides a greater variability of the acoustic field, which actually aids the estimation procedure. However, MFP is also predicated on having sufficient knowledge of the ocean structure, which allows us to make an accurate prediction of the acoustic pressure field. When real-world (limited) environmental data are used to calculate the predicted fields, a "mismatch" generally occurs between the measured data and the predicted pressure field, leading to errors in the estimator.

There have been numerous studies in this area; as initial material, we would refer the reader to: the review of Baggeroer et al. (1993), the special issue of IEEE Ocean Engineering (1993), Doolittle et al. (1993), Fizell (1987), and Hamson and Heitmeyer (1989).

In this section, we consider the main ideas of MFP and compare the different beamformers and ambiguity functions that form the MFP technique. To begin with let us consider a set of receiving sensors (hydrophones) placed (as a rule) in a vertical line array at depths z_i, where the total number I can be several tens of hydrophones. In our notation, the complex sound field due to a unit point source of frequency ω in a regular waveguide at a distance r from the source, at hydrophone number i [coordinates r_i, z_i $(i = 1, 2, \ldots I)$] is given by the expression

$$\Psi_i = \Psi(r_i, z_i) = \sum_l \Psi_l(z_0)\Psi_l(z_i)H_0^{(1)}(\xi_l r_i). \tag{8.32}$$

These values can be considered as the components of a vector:

$$\Psi = \begin{pmatrix} \Psi_1 \\ \Psi_2 \\ \vdots \\ \Psi_I \end{pmatrix}. \tag{8.33}$$

If we take an analogous vector for the experimentally measured pressure, then the complex amplitude at the frequency ω is constructed as in our previous work, see (8.7), (8.8), as:

$$\Psi_i^{\mathrm{e}} = A_i \exp(i\varphi_i') \tag{8.34}$$

The corresponding vector for this signal is:

$$\Psi^{\mathbf{e}} = \begin{pmatrix} \Psi_1^{\mathrm{e}} \\ \Psi_2^{\mathrm{e}} \\ \vdots \\ \Psi_I^{\mathrm{e}} \end{pmatrix}. \tag{8.35}$$

We also suppose that this sound field is produced by a source of unit amplitude. Let's now introduce the scalar product of two vectors, of dimension J

$$(\Psi, \Phi) = \Psi^{+}\Phi = \sum_{i=1}^{I} (\Psi_i)^{*}\Phi_i \tag{8.36}$$

where the superscript + denotes the Hermitian conjugate.

If $\Psi = \Phi$, then (8.36) gives the norm of the vector $\|\Psi\|$:

$$\|\Psi\|^2 = \sum_{i=1}^{I} |\Psi_i|^2. \tag{8.37}$$

We now also introduce the normalized vectors $\mathbf{w}$

$$\mathbf{w} = \left(\frac{\Psi}{\|\Psi\|} \right). \tag{8.38}$$

Obviously $\|\mathbf{w}\| = 1$. Equation (8.33) for the modal decomposition of the received sound field can also be presented in matrix form, using the following matrices:

$$\mathbf{M_S} = \begin{pmatrix} \Psi_1(z_0) \\ \Psi_2(z_0) \\ \vdots \\ \Psi_L(z_0) \end{pmatrix}, \tag{8.39}$$

$$\mathbf{H}(r_i) = \begin{pmatrix} H_0^{(1)}(\xi_1 r_i) & 0 & \cdots & 0 \\ 0 & H_0^{(1)}(\xi_2 r_i) & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & H_0^{(1)}(\xi_L r_i) \end{pmatrix}, \tag{8.40}$$

$$\mathbf{W} = \begin{pmatrix} \Psi_1(z_1) & \Psi_1(z_2) & \cdots & \Psi_1(z_I) \\ \Psi_2(z_1) & \Psi_2(z_2) & \cdots & \Psi_2(z_I) \\ \cdots & \cdots & \cdots & \cdots \\ \Psi_L(z_1) & \Psi_L(z_2) & \cdots & \Psi_L(z_I) \end{pmatrix}. \tag{8.41}$$

Thus, (8.33) takes the form

$$\Psi = \mathbf{H}\mathbf{M}_{\mathbf{S}}\mathbf{W}. \tag{8.42}$$

As we said, the basis of MFP is the construction of the some expression which contains both the experimental data and the parameters of some theoretical model. This expression (8.42) allows us both to carry out the comparison between experiment and model data and also to estimate unknown model parameters. To achieve the last goal, we take some set of parameters of the model (e.g., input parameters, like the source coordinate and/or waveguide parameters) as "fitting parameters." Their estimation is carried out by maximization of another function (the ambiguity function), which has a maximum value when there is an exact coincidence between the experimental data and the theoretical model.

A key element of this approach is the ambiguity function (or estimator, or beamformer). There are many different expressions for this beamformer, some of which we will discuss below. Let us denote the set of unknown (or hypothesized) parameters of the source and/or waveguide by vector $\mathbf{a}$. In this case, the vector of the sound field, calculated for different depths of the receiver will be denoted as $\Psi(\mathbf{a})$. Let's then consider some forms of the ambiguity functions using these vectors. We start with the simplest case, the so-called Bartlett Beamformer.

8.2.1 Conventional MFP, Bartlett Beamformer

For this beamformer, we take the squared scalar product of the vector of theoretical data (the replica vector) $\Psi(\mathbf{a})$ and the "experimental" vector Ψ^{e} as the value determining the closeness of experimental and measured data. More exactly, for the replica vector we will take the normalized vector $\mathbf{w}(\mathbf{a})$. Additionally, for real experiments, we should carry out some averaging over ensembles (or over time, assuming ergodicity), which gives us

$$B_{\mathrm{BE}}(\mathbf{a}) = \left\langle \left| (\mathbf{w}^{+}(\mathbf{a})\Psi^{\mathrm{e}}) \right|^2 \right\rangle. \tag{8.43}$$

This result is called the traditional MFP beamformer (processor) by (e.g., Baggeroer et al. 1993) or the Bartlett processor by (e.g., Daugherty and Lynch 1990). After substitution of the components of the vectors we created, we get

$$B_{\mathrm{BE}}(\mathbf{a}) = \sum_{i,k=1}^{I} \left\langle w_i (\Psi_i{}^{\mathrm{e}})^* w_k^* \Psi_k{}^{\mathrm{e}} \right\rangle = \sum_{i,k} w_i(\mathbf{a}) K_{ik} w_k^*(\mathbf{a}) \tag{8.44}$$

where the value

$$K_{ik} = \left\langle \Psi_i^{\mathrm{e}} (\Psi_k^{\mathrm{e}})^* \right\rangle \tag{8.45}$$

is the correlation (or covariance) matrix. This simple physical interpretation is one of the main attractions of the MFP approach. In the matrix form, we have for the Bartlett beamformer

$$B_{\mathrm{BE}}(\hat{a}) = \mathbf{w}^{+}(\mathbf{a})\mathbf{K}\mathbf{w}(\mathbf{a}), \tag{8.46}$$

$$\mathbf{K} = \langle \Psi^{\mathrm{e}} \Psi^{\mathrm{e}+} \rangle.$$

We note that in the deterministic case the averaging brackets can be omitted.

Let us now examine the properties of the function $B_c(\mathbf{a})$ and another beamformer with the purpose of finding the unknown parameters of the waveguide and the source. Let's take a replica field $\mathbf{w}(\mathbf{a}')$ with other values of the vector of parameters $\mathbf{a}'$ as the "experimental" vector and consider the ambiguity function, (8.46), that it produces. More exactly, we take modulus of this expression:

$$\theta(\mathbf{a}, \mathbf{a}') = |(\mathbf{w}^{+}(\mathbf{a})\mathbf{w}(\mathbf{a}'))|. \tag{8.47}$$

This function is equal to 1 if $\mathbf{a} = \mathbf{a}'$. It is called "two parameter ambiguity function" in Baggeroer et al. (1993), and can help estimate the position and values of the mainlobe and sidelobes of a beamformer. It is analogous to the resolution kernel found in inverse theory.

8.2.2 *Two (Analytic) Examples of the Parameter Ambiguity Function*

1. It is interesting and instructive to consider the simplest cases for beamformers. For a first example, consider estimation of the source position in free space using a vertical line array. Consider the source at the point (r, z). In this case

$$\Psi_i = \frac{\exp\left|\mathrm{i}k\sqrt{r^2 + (z - z_i)^2}\right|}{\sqrt{r^2 + (z - z_i)^2}} = \frac{\exp(\mathrm{i}kr_i)}{r_i} \tag{8.48}$$

where we denote $r_i = \sqrt{r^2 + (z - z_i)^2}$. The normalized vector w_i is given by

$$w_i = \frac{\exp(\mathrm{i}kr_i)}{r_i \sqrt{\sum_{k=1}^{I} r_k^{-2}}}. \tag{8.49}$$

The ambiguity function without noise is thus given as

$$\theta(r,z;r',z') = \left| \sum_{i=1}^{I} \frac{\exp\{ik(r_i - r'_i)\}}{r_i r'_i \sqrt{\sum_{k=1}^{I} r_k^{-2} I \sum_{k=1}^{J} r'_k{}^{-2}}} \right|. \tag{8.50}$$

With this simple mathematics, we can draw figures illustrating the behavior of this function for different positions of the source.

2. Next, let us construct the ambiguity function used to estimate the angle of propagation (direction of propagation) of a plane wave in free space. A simple plane wave, propagating at an angle α with the x-axis, has the form

$$P(x,z) = \exp[ik(x\cos\alpha + z\sin\alpha)]. \tag{8.51}$$

The field source at a set of receivers placed parallel to z-axis is given by

$$\Psi_i = \exp[ik(x\cos\alpha + z_i\sin\alpha)]. \tag{8.52}$$

The normalized vector thus becomes

$$w_i = \frac{1}{\sqrt{I}} \exp[ik(x\cos\alpha + z_i\sin\alpha)] \tag{8.53}$$

and so two-parameter ambiguity function is simply

$$\theta(\alpha,\alpha') = \frac{1}{I} \left| \sum_{i=1}^{I} \exp\{ik[x(\cos\alpha - \cos\alpha') + z_i(\sin\alpha - \sin\alpha')]\} \right|. \tag{8.54}$$

In particular, if $\alpha' = 0$, then we obtain

$$\theta(\alpha,0) = \frac{1}{I} \left| \sum_{i=1}^{I} \exp\{ik[x(\cos\alpha - 1) + z_i\sin\alpha]\} \right|. \tag{8.55}$$

Again, this is simply evaluated to explore resolution and parameter sensitivity.

Of course, due to the inaccuracy of the models and the existence of noise, experimental values cannot be equal to theoretical for any parameters of the waveguide and source. There is the simple way to take this circumstance into account. Let us suppose that experimentally measured values consist of both desired and noise components, and as the desired component we can take the theoretical value, i.e.,

$$\Psi_i^{\mathrm{e}} = \sigma_{\mathrm{s}}\Psi_i + \sigma_{\mathrm{n}}\Psi_i^{\mathrm{n}} \tag{8.56}$$

where $\sigma_{s,n}$ is the amplitude of the desired and noise signals, and Ψ_i^n is a random vector, determining the noise at the receiver. Assuming that the noise is uncorrelated, Gaussian and white, covariance matrix can be found as:

$$\left\langle \Psi_i^n \right\rangle = 0, \quad \left\langle \Psi_i^n (\Psi_k^n)^* \right\rangle = \delta_{ik}$$

and

$$K_{ik} = \sigma_s^2 \Psi_i (\Psi_k)^* + \sigma_n^2 \delta_{ik}.$$

Or, in matrix form

$$\mathbf{K} = \sigma_s^2 \Psi \Psi^+ + \sigma_n^2 \mathbf{I} \tag{8.57}$$

where $\mathbf{I}$ is the unit matrix of dimension $I \times I$.

For this case, we can study the behavior of function (8.47) in the presence of noise. The Bartlett processor in this case is

$$B_{BE}(\mathbf{a}) = \sigma_s^2 |\mathbf{w}^+(\mathbf{a})\Psi|^2 + \sigma_n^2. \tag{8.58}$$

If we use the normalized vector (8.38) and the two-parameter ambiguity function (8.47), then we get

$$B_{BE}(\mathbf{a}, \mathbf{a}') = \sigma_s^2 \|\Psi\|^2 |\theta(\mathbf{a}, \mathbf{a}')|^2 + \sigma_n^2. \tag{8.59}$$

It is generally useful to introduce the average SNR for separate hydrophones, i.e.,

$$\text{SNR} = \frac{\sigma_s^2 \|\Psi\|^2}{J\sigma_n^2}. \tag{8.60}$$

For the Bartlett processor, we can then get a useful expression, which we will write up to constant (as it can be used in a logarithmic scale)

$$B_{BE}(\mathbf{a}, \mathbf{a}') = \text{SNR} \times J |\theta(\mathbf{a}, \mathbf{a}')|^2 + 1. \tag{8.61}$$

Expression (8.54) can easily be calculated for different models of the environment, and allows one to study the existence and sizes of various mainlobes and sidelobes.

8.2.3 Maximum Likelihood Beamformer

Another variety of ambiguity function is connected with the maximum likelihood method (MLM) of parameter estimation, originally proposed and used in Capon (1969) and Hinish (1973). Another name for this technique is "Minimum variance,

Distortionless Filter MFP" as denoted in Baggeroer et al. (1993). The essence of this method is that in processing the signals, we can use a beamformer in the form of (8.43) or (8.46); however, as a matching vector (weighting vector) we should take $\mathbf{w}_{\mathrm{MLM}}$, which is not the theoretical vector but is constructed according to some rule of optimization. Specifically, the vector $\mathbf{w}_{\mathrm{MLM}}$ is constructed using the Lagrange multiplier method based on the expression

$$F = \mathbf{w}_{\mathrm{MLM}}^{+}\mathbf{K}\mathbf{w}_{\mathrm{MLM}} + \lambda(\mathbf{w}_{\mathrm{MLM}}^{+}\mathbf{w}_{\mathrm{MLM}} - 1). \tag{8.62}$$

So we obtain the connection between the weighting vector $\mathbf{w}_{\mathrm{MLM}}$ and the theoretical vector as

$$\mathbf{w}_{\mathrm{MLM}} = \frac{\mathbf{K}^{-1}\mathbf{w}}{\mathbf{w}^{+}\mathbf{K}^{-1}\mathbf{w}}. \tag{8.63}$$

In this case, the ambiguity function can be expressed through the normalized replica vector

$$B_{\mathrm{MLM}} = \left[\mathbf{w}^{+}\mathbf{K}^{-1}\mathbf{w}\right]^{-1}. \tag{8.64}$$

If we use expression (8.57) for the covariance matrix and also use the Woodbury identity for matrices, i.e.,

$$(\mathbf{A} + \mathbf{B}\mathbf{C}^{+})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{C}^{+}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}^{+}\mathbf{A}^{-1}, \tag{8.65}$$

where $\mathbf{A}$ is an $M \times M$ matrix, $\mathbf{B}$ and $\mathbf{C}$ are $M \times N$ matrices, and $\mathbf{I}$ is the $N \times N$ identity matrix, then we can get an expression for MLM processor, i.e.,

$$B_{\mathrm{MLM}}(\mathbf{a}, \mathbf{a}') = \frac{1 + \mathrm{SNR} \times I}{1 + \mathrm{SNR} \times I(1 - |\theta(\mathbf{a}, \mathbf{a}')|^2)}. \tag{8.66}$$

We see that if there is no correlation between the measured field and the replica, the estimator reduces approximately to the noise level (which on our scale corresponds to $B = 1$). The $1 - |\theta(\mathbf{a}, \mathbf{a}')|^2$ term, referred to as the "estimator–subtractor" term by Van Trees (1968), is responsible for achieving the desired sidelobe minimization.

8.2.4 Variable Coefficient Likelihood Beamformer

We next introduce a third processor, which we will refer to as the variable coefficient likelihood method (VCLM) (Daugherty and Lynch 1990). It has similar characteristics to the MLM processor, but is more robust to environmental mismatch. We define the VCLM processor as

$$B_{\mathrm{VCLM}}(\mathbf{a}) = \left[\mathbf{w}^{+}(\mathbf{a})\mathbf{K}^{-\alpha}\mathbf{w}(\mathbf{a})\right]^{-1/\alpha}, \tag{8.67}$$

where α is the variable coefficient. For the case of white noise and a single source, the VCLM estimator depending on two parameters can be rewritten as

$$B_{\mathrm{VCLM}}(\mathbf{a}, \mathbf{a}') = \frac{1 + \mathrm{SNR} \times I}{\left[|\theta(\mathbf{a}, \mathbf{a}')|^2 + (1 + \mathrm{SNR} \times I)^{\alpha}(1 - |\theta(\mathbf{a}, \mathbf{a}')|^2)\right]^{1/\alpha}}. \tag{8.68}$$

This is similar in appearance to a low-pass filter where the SNR and the exponent α determine the cutoff point. When the correlation coefficient is equal to 1, the output of the estimator is equal to the signal plus noise. When the correlation coefficient is less than 1, but greater than the cutoff point, the estimator is still approximately equal to the signal-plus-noise value. Below the cutoff point, the estimator performs much like a normal MLM estimator, except that the "estimator–subtractor" term has a variable exponent to control the extent of sidelobe expression. We will note that the selection of the optimal exponent value is accomplished by finding the α which maximizes the difference $B_{\mathrm{VCLM}}\left(\alpha, \mathrm{SNR}, |\theta|_{\max}\right) - B_{\mathrm{MLM}}\left(\mathrm{SNR}, |\theta|_{\max}\right)$. This requires one to estimate the SNR and the maximum correlation coefficient between the source and replica fields, two physical and interesting properties.

In closing this section, we will just add the observation that in considering these methods in more detail, we routinely examine the singular value decomposition (SVD) of the covariance matrix, finding eigenvalues and eigenvectors of matrix **K** (see, for example, Baggeroer et al. 1993).

8.2.5 Comparison of Processors for Simple Examples

We will now illustrate the properties of the different ambiguity functions we have considered for the two simple examples of free space and a waveguide. Let us consider a source with frequency 150 Hz, placed at some depth and distance from a vertical line array. As the vertical array we take 21 hydrophones spaced 2.5 m from each other ($\lambda/4$), so our array is of 50 m length. In Figs. 8.1 and 8.2, we show ambiguity functions constructed on the basis of the simple examples. The plane wave approximation is acceptable if the source is placed in the far field. The incident angle we use is 0° and the average SNR at the receiver is 10 dB. In Fig. 8.1, the ambiguity functions for a BE processor (dashed line), an MLM processor (solid line), and a VCLM processor (dashed-dotted line) are plotted. In this case, the VCLM and MLM curves are the same. This result occurs because there is an exact match between the source and replica field at 0°. The exponent of the VCLM processor is automatically selected as 1 by the optimization algorithm, and the method thus reduces to the MLM processor.

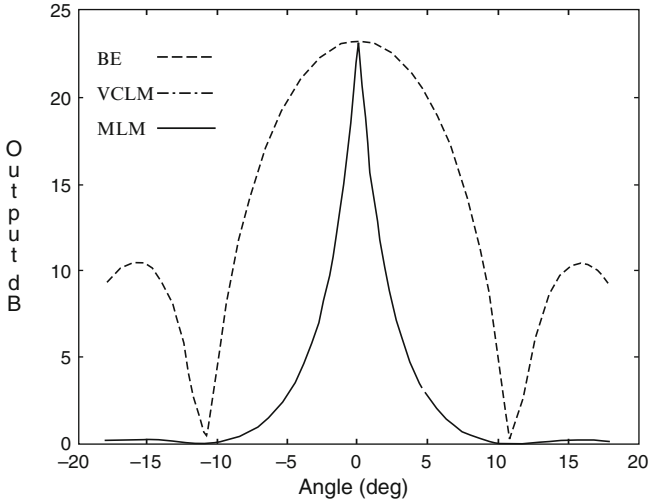


Fig. 8.1 Plane wave ambiguity function for a BE processor (*dashed line*), a MLM processor (*solid line*) and a VCLM processor (*dash-dot line*) for the cases of (**a**) a perfect match and (**b**) a mismatch caused by using the plane wave assumption with a source close to the array

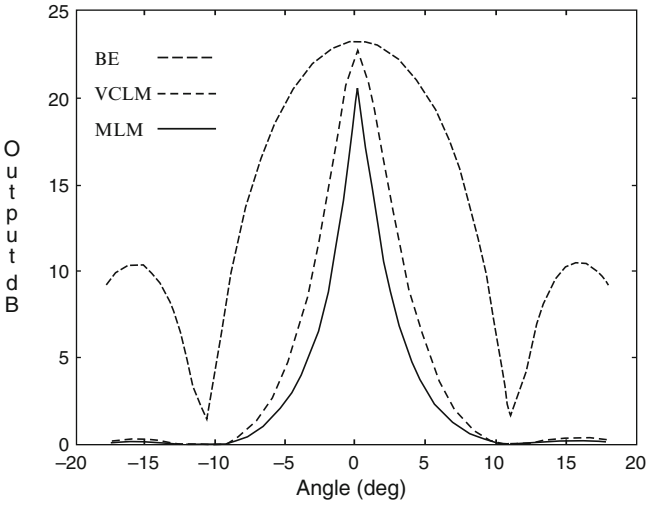


Fig. 8.2 Plane wave ambiguity function for a BE processor (*dashed line*), a MLM processor (*solid line*) and a VCLM processor (*dash-dot line*) for the cases of (**a**) a perfect match and (**b**) a mismatch caused by using the plane wave assumption with a source close to the array

Next, the source is moved closer to the array (range 1,000 m) so that the incident field has a distinct curvature, which prevents a perfect match with any plane wave replicas. The incident angle remains 0° and the signal strength is adjusted so that the average SNR at the receiver is still 10 dB. As seen in Fig. 8.2, the VCLM

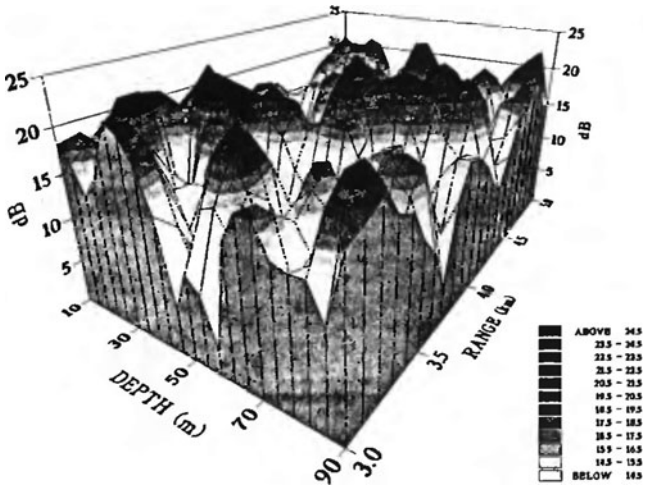


Fig. 8.3 Ambiguity surface using the BE processor in an ideal case. True source location is at 4 km range and 54 m depth

beamformer maintains a large peak since the algorithm has selected a new exponent value that maximizes the peak relative to all other values. In general, the VCLM processor will always work better than the MLM processor and also shows the most improvement at high SNR and low correlation between the source and replica fields.

Although the previous examples used simple plane and spherical waves for illustrative purposes, the results are easily extrapolated to more complicated processes, as we will show in our shallow water waveguide examples. As a more complex example, we will consider the localization of a source of 150 frequency in a Pekeris waveguide. The channel parameters for this case are $\rho_1/\rho = 1.8$, $c_1 = 1,620\,\text{m/s}, c = 1,500\,\text{m/s}, H = 100\,\text{m}$. The receiver array is again as described earlier, the SNR is 10 dB, and the source is placed at 54 m depth and 4 km distance from the array. The ambiguity function is computed over a large r–z place grid to determine the magnitude and position of the source peak, as well as to quantify the background statistics. The grid size is also chosen such that the ambiguity function will be sampled at the exact source position. This is ensured by using 21 sampling points in the range interval 3–5 km ($\Delta r = 100\,\text{m}$ and there are 21 points in the depth interval 10–90 m). The peak-to-background ratio (PBR) expressed in dB is then calculated for the "measured" data.

Let's now analyze the performance characteristics of MFP using the BE processor and the MLM processor in the perfectly matched case. The propagation model used to synthetically produce the "measured" signal is the same as that used to produce the replica fields (8.42). We display a contour plot that shows the top 10 dB of the ambiguity surface so that a better look at sidelobe suppression can be obtained.

The BE processor ambiguity surface shown in Fig. 8.3 has a global peak at the true source position, but is also has many ambiguous peaks. The magnitude of the source peak is 24.5 dB, but mainly as a result of increased background levels,

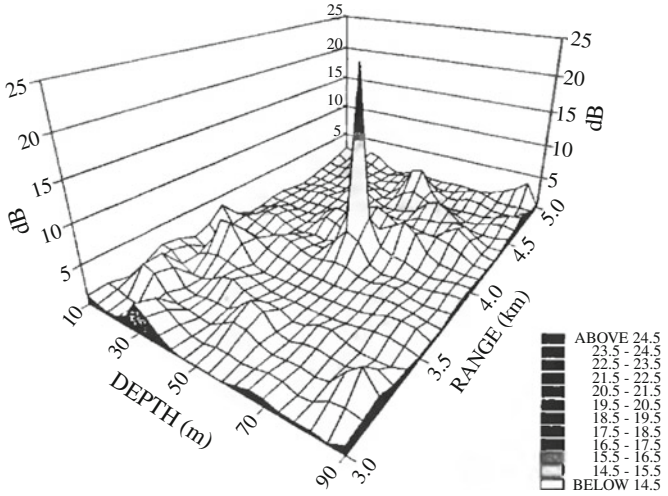


Fig. 8.4 Ambiguity surface using a MLM processor in an ideal case. True source location is at 4 km range and 54 m depth

the PBR is only 12 dB. In Fig. 8.4, the MLM processor produces an ambiguity surface that clearly displays the true source location with a peak magnitude 24.5 dB above the background, and with all ambiguous peaks reduced by more than 10 dB compared to the BE. In fact the largest ambiguous peak is 19 dB less than the source peak. Moreover, throughout our study, the BE processor could not unambiguously locate the source.

8.3 Spatial Coherence of the Sound Field in Shallow Water and Array Signal Gain

Some of the main tools for generating, measuring, and recording the sound field on the sea shelf are linear vertical and horizontal radiating and receiving acoustic arrays. The technical implementation of these arrays and their parameters will be considered in Chap. 10. Here we shall discuss only their basic characteristics which are important from the viewpoint of signal processing. First and foremost of such characteristics are the beam pattern and the array signal gain (ASG).

In the general case, the beam pattern $\hat{D}(\vec{e})$ of the receiving array is taken as the ratio of voltages at the array output between when a signal comes from a point nondirectional source placed in the direction given by the vector $\vec{e} = \vec{R}/|\vec{R}|$ and when it comes from a certain chosen direction usually corresponding to a signal maximum,

$$\hat{D}(\vec{e}) = \frac{U(\vec{e})}{U(\vec{e}_0)} = B_{a0}(\vec{e}) \exp(\mathrm{i}\vartheta_{ph}(\vec{e})), \tag{8.69}$$

where $B_{a0}(\vec{e})$ and $\vartheta_{ph}(\vec{e})$ are generally referred to as the amplitude[3] and phase beam patterns, respectively. In further discussions, we will operate only with $B_{a0}(\vec{e})$; for brevity, we refer it to as the beam pattern. Here it is assumed that the receiving array center is at the coordinate origin, and the source is in the far-field region at a point with radius vector $\vec{R}$. One more important assumption is that the medium is assumed to be homogeneous and deterministic.

For the receiving array, the ASG is understood to be the following quantity

$$K_a = 10 \lg \frac{W_s^a / W_n^a}{W_s^{sr} / W_n^{sr}}, \tag{8.70}$$

where W_s^a and W_n^a are the average signal and noise powers at the array output (it is assumed that the point sound source is positioned in the $\vec{u}_0$ direction), and W_s^{sr} and W_n^{sr} are the average signal and noise powers at the output of a single nondirectional receiver, respectively. It is clear that the beam pattern and ASG are related with each other. Indeed, if we assume that the single nondirectional receiver is a "tightly coiled" grouped array (for a discrete array, this means that receivers are spaced by a minimum possible distance much smaller than the sound wavelength), then $W_s^a = W_s^{sr}$ and

$$W_n^a = \widehat{A} \int_\Omega \left\langle U_{na}^2(\vec{e}) \right\rangle d\hat{\Omega} = \widehat{A} A_1^2 \int_\Omega I_n(\vec{e}) B_{a0}^2(\vec{e}) d\hat{\Omega}, \tag{8.71}$$

$$W_n^{sr} = \widehat{A} \int_\Omega \left\langle U_{nsr}^2(\vec{e}) \right\rangle d\hat{\Omega} = \widehat{A} A_1^2 \int_\Omega I_n(\vec{e}) d\hat{\Omega}, \tag{8.72}$$

where $U_{na}(\vec{e})$ and $U_{nsr}(\vec{u})$ are, respectively, the array and single receiver output voltages from a noise sound source placed in the $\vec{u}$ direction, $I_n(\vec{u})$ is the average noise intensity in a given direction, $\hat{\Omega}$ is the solid angle, and $\widehat{A}$ and A_1 are the constant coefficients determined by the characteristics of the array. In this case, the receiving array gain is given by

$$K_a = 10 \lg \frac{W_n^{sr}}{W_n^a} = 10 \lg \frac{\int_\Omega I_n(\vec{e}) d\hat{\Omega}}{\int_\Omega I_n(\vec{e}) B_{a0}^2(\vec{e}) d\hat{\Omega}}. \tag{8.73}$$

[3] Sometimes the amplitude beam pattern is called the beamformed output of array.

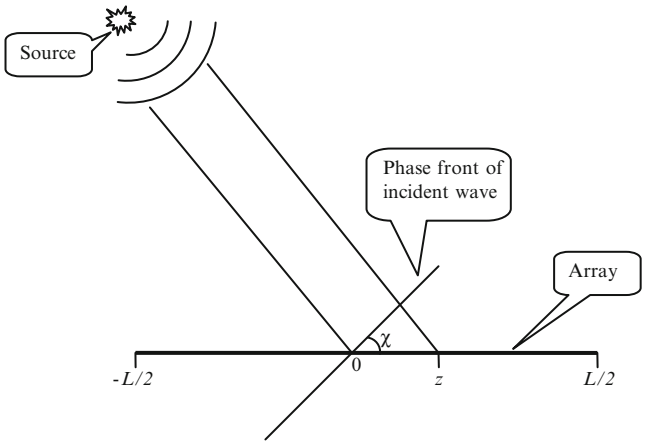


Fig. 8.5 Sound signal reception by using horizontal linear array

For isotropic noise,

$$K_{\mathrm{a}} = 10\lg\frac{W_{\mathrm{n}}^{\mathrm{sr}}}{W_{\mathrm{n}}^{\mathrm{a}}} = 10\lg\frac{4\pi}{\int\limits_{\Omega} B_{\mathrm{a}0}^2(\vec{e})d\hat{\Omega}}. \tag{8.74}$$

Further, let us consider the simplest version of the linear continuous receiving array of length L with identical sensitivity along its aperture. For simplicity, we suppose that the array is in free space.[4] To determine its directional pattern and the gain, we will use the spherical coordinate system with the z axis directed along the array aperture. Let us designate the longitude and latitude in the adopted coordinate system as φ and χ and assume that a plane wave with amplitude P_0 and zero phase is incident on such an array at angle χ (see Fig. 8.5). The output voltage from such an array is

$$U(\chi) = A\int\limits_{-L/2}^{L/2} P_0\exp(ikz\sin\chi)dz = AP_0L\frac{\sin\left(\frac{\pi L}{\lambda}\sin\chi\right)}{\frac{\pi L}{\lambda}\sin\chi}, \tag{8.75}$$

where A is the normalization factor defining the sound pressure–electric signal conversion factor for the array under consideration. The $U(\chi)$ maximum takes place

[4] We note here that the further derived expressions, to a first approximation, are also valid for the linear array being in the waveguide, first of all, for the horizontal array. However, in some cases, features in waveguide sound propagation should be taken into account for extended arrays (see, e.g., Eliseevnin (1981), Eliseevnin and Tuzhilkin (2001) and Sect. 8.4).

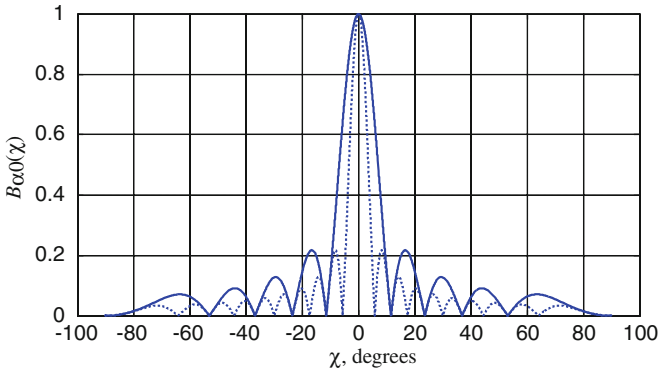


Fig. 8.6 Beam pattern of line array. $L = 5\lambda$ – *solid line*, $L = 10\lambda$ – *dotted line*

at $\chi = 0$ and is $U(0) = AP_0L$. Hence, the array directional pattern $B_{a0}(\chi)$ in the plane passing through the sound source and array aperture is written as

$$B_{a0}(\chi) = \left| \frac{\sin\left(\frac{\pi L}{\lambda} \sin\chi\right)}{\frac{\pi L}{\lambda} \sin\chi} \right|. \tag{8.76}$$

The $B_{a0}(\chi)$ curve at $L = 5\lambda$ and $L = 10\lambda$ is shown in Fig. 8.6.

The ASG of such an array at $L \gg \lambda$, according (8.74), is given by[5]

$$K_a = 10\lg\left\{ 4\pi \left[\int_0^{2\pi} \int_{-\pi/2}^{\pi/2} \left(\frac{\sin\left(\frac{\pi L}{\lambda} \sin\chi\right)}{\frac{\pi L}{\lambda} \sin\chi} \right)^2 \cos\chi d\chi d\varphi \right]^{-1} \right\} \cong 10\lg\frac{2L}{\lambda}. \tag{8.77}$$

For a constant sensitivity along the array aperture, the angle χ_0 defining the direction to the sound source, at which $U(\chi)|_{\chi=\chi_0} = \max$, is $\chi_0 = 0$. In a similar way, it can be shown that, if the array sensitivity varies along the aperture by $\exp(ikz \sin\chi_0)$, $B_{a0}(\vartheta)$ is given by

$$B_{a0}(\chi) = \left| \frac{\sin\left(\frac{\pi L}{\lambda}(\sin\chi - \sin\chi_0)\right)}{\frac{\pi L}{\lambda}(\sin\chi - \sin\chi_0)} \right|, \tag{8.78}$$

and the angle χ_0 is usually referred to as the array compensation or "steering" angle. At small compensation angles, $\chi_0 \ll 90°$, and for a large array, $L \gg \lambda$, the gain is also close to the value of $K_a \cong 10\lg(2L/\lambda)$.

[5] We note that, for our notation, the angle $\chi = \pi/2 - \vartheta$, where ϑ is the polar angle in the spherical coordinate system, defines the direction to the sound source. $d\hat{\Omega} = \sin\vartheta d\vartheta d\varphi$.

In practice, receiving arrays are discrete rather than continuous, i.e., they consist of several nondirectional receivers (hydrophones). In this case, with identical receivers composing the array, the directional pattern is written as

$$B_{\mathrm{a}0}(\chi) = \left| \frac{\sin\left(I\frac{\pi d}{\lambda}(\sin\chi - \sin\chi_0)\right)}{I\sin\left(\frac{\pi d}{\lambda}(\sin\chi - \sin\chi_0)\right)} \right|, \tag{8.79}$$

where I is the number of receivers and d is the distance between them. We note that for $d \approx \lambda/2$ and for a large number of receivers, $I \gg 1$, (8.79) and (8.78) are almost identical for angles χ corresponding to the mainlobe of the beam pattern (see Fig. 8.6). According to (8.74), the ASG of such an array at small angles χ_0 is given by

$$K_{\mathrm{a}} = 10\lg\left\{ 4\pi \left[\int_{-\pi/2}^{\pi/2} \int_{0}^{2\pi} \left(\frac{\sin I\frac{\pi d}{\lambda}(\sin\chi - \sin\chi_0)}{I\sin\frac{\pi d}{\lambda}(\sin\chi - \sin\chi_0)} \right)^2 \cos\chi d\chi d\varphi \right]^{-1} \right\} \cong 10\lg I. \tag{8.80}$$

This formula can also be derived from (8.70), using quite different considerations (see Urick 1975). That is, the quantities $W_{\mathrm{s}}^{\mathrm{a}}$ and $W_{\mathrm{n}}^{\mathrm{a}}$ for the array under consideration can be written as

$$W_{\mathrm{s}}^{\mathrm{a}} = \widehat{A}\left\langle [s_1(t) + s_2(t) + \cdots + s_I(t)]^2 \right\rangle, \tag{8.81}$$

$$W_{\mathrm{n}}^{\mathrm{a}} = \widehat{A}\left\langle [n_1(t) + n_2(t) + \cdots + n_I(t)]^2 \right\rangle, \tag{8.82}$$

where $s_1(t),\ s_2(t), \ldots, s_I(t)$ and $n_1(t),\ n_2(t), \ldots, n_I(t)$ are output voltages of signal and noise from individual array receivers. This suggests that

$$\frac{W_{\mathrm{s}}^{\mathrm{a}}}{W_{\mathrm{n}}^{\mathrm{a}}} = \frac{(\langle s_1 s_1\rangle + \langle s_1 s_2\rangle + \cdots + \langle s_1 s_I\rangle) + (\langle s_1 s_2\rangle + \langle s_2 s_2\rangle + \cdots + \langle s_2 s_I\rangle) + \cdots}{(\langle n_1 n_1\rangle + \langle n_1 n_2\rangle + \cdots + \langle n_1 n_I\rangle) + (\langle n_2 n_1\rangle + \langle n_2 n_2\rangle + \cdots + \langle n_2 n_I\rangle) + \cdots}. \tag{8.83}$$

For simplicity, the time dependence in formula (8.83) is omitted. For identical receivers, $\langle s_1^2\rangle = \langle s_2^2\rangle = \cdots = \langle s_n^2\rangle \equiv \langle s^2\rangle$ and $\langle n_1^2\rangle = \langle n_2^2\rangle = \cdots = \langle n_I^2\rangle \equiv \langle I^2\rangle$. Here the quantities $\langle s^2\rangle$ and $\langle I^2\rangle$ are understood as average signal and noise from a single receiver. We can rewrite formula (8.83) in the following form

$$\frac{W_{\mathrm{s}}^{\mathrm{a}}}{W_{\mathrm{n}}^{\mathrm{a}}} = \frac{\langle s^2\rangle}{\langle n^2\rangle} \frac{\sum_j \sum_i (\rho_{\mathrm{s}})_{ij}}{\sum_j \sum_i (\rho_{\mathrm{n}})_{ij}}, \tag{8.84}$$

where $(\rho_s)_{ij} = \left(\langle s_i s_j \rangle\right)/(\langle s_i^2 \rangle \langle s_j^2 \rangle)^{1/2}$. and $(\rho_n)_{ij} = (\langle n_i n_j \rangle)/(\langle n_i^2 \rangle \langle n_j^2 \rangle)^{1/2}$. are the cross correlation coefficients at the output of the ith and jth array receivers for the signal and noise, respectively. From this, according to (8.70), it follows that the ASG is

$$K_a = 10 \lg \frac{\sum\limits_j \sum\limits_i (\rho_s)_{ij}}{\sum\limits_j \sum\limits_i (\rho_n)_{ij}}. \tag{8.85}$$

For a completely correlated signal $\left((\rho_s)_{i,j} = 1\right)$ and uncorrelated noise

$$(\rho_n)_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

$$K_a = 10 \lg I. \tag{8.86}$$

As follows from (8.76), (8.78), (8.79), and (8.86), as the array length (number of receivers) increases, the directional pattern narrows and the ASG increases. However, for a long array length in a shallow sea, this systematic feature is violated even for a horizontal array, where there are no length limitations associated with the sea depth. The cause is the lowering of the signal correlation over the array aperture, i.e., $((\rho_s)_{i,j})$ already is not unity at $|i - j| \gg 1$. In turn, this is because the sound from the source propagates to the ith and jth receivers via appreciably different trajectories, along which propagation characteristics vary randomly. In other words, for large arrays, it should be considered that shallow water is a random-homogeneous isotropic medium even to a first approximation, and the signal received by the array becomes a random value characterized by some spatial correlation function. It is natural that, by definition (8.69), the beam pattern of such an array is also random and its behavior is characterized by the behavior of the random variable $U(\vec{u})$. Let us consider this in more detail using the example of a linear horizontal continuous receiving array with uniform sensitivity over its aperture. Let us further assume that the stochastic nature of the signal is first of all caused by random fluctuations of its phase. In this case, we can rewrite (8.75) as

$$U(\chi) = A \int\limits_{-L/2}^{L/2} P_0 \exp(ikz \sin\chi + i\phi(z)) dz, \tag{8.87}$$

where $\phi(z)$ is a random variable. Making the change of variables $\tilde{z} = 2z/L$ and denoting $\pi L \sin\chi/\lambda = \vartheta$, we obtain

$$\frac{2U(\chi)}{P_0 AL} \equiv f(\vartheta) = \int\limits_{-1}^{1} \exp(i\vartheta\tilde{z} + i\widetilde{\phi}(\tilde{z})) d\tilde{z}. \tag{8.88}$$

Correspondingly, the beam pattern of the array under consideration is also a random variable and can be written as

$$B_{\mathrm{a}}(\chi) \equiv \widehat{B}_{\mathrm{a}}(\vartheta) = \frac{|U(\chi)|}{|U_0(0)|} = \frac{|f(\vartheta)|}{2}. \tag{8.89}$$

In the absence of random medium perturbations, $\phi(z) = 0$ and $B_{\mathrm{a}}(\chi) = B_{\mathrm{a}0}(\chi)$.

Let us further derive an expression for the ASG of such an array in a random, homogeneous isotropic medium and with isotropic noise. Assuming that the noise and fluctuating signal are statistically independent, and using expressions (8.71) and (8.72), we can write the following expressions for the quantities entering (8.70),

$$W_{\mathrm{s}}^{\mathrm{a}} = \widehat{A} < U^2(0) > = \widehat{A} A^2 L^2 P_0^2 \frac{\left\langle |f(0)|^2 \right\rangle}{4}, \tag{8.90}$$

$$W_{\mathrm{s}}^{\mathrm{sr}} = \widehat{A} U_0^2(0) = \widehat{A} A^2 L^2 P_0^2, \tag{8.91}$$

$$\begin{aligned} W_{\mathrm{n}}^{\mathrm{a}} &= \widehat{A} \int\limits_{\Omega} \left\langle U_{na}^2(\vec{u}) \right\rangle d\Omega = \widehat{A} A_1^2 I_n \int\limits_0^{2\pi} \int\limits_{-\pi/2}^{\pi/2} \left\langle B_a^2(\chi) \right\rangle \cos\chi d\chi d\varphi \\ &= \widehat{A} A_1^2 I_n \frac{2\lambda}{L} \int\limits_{-\pi L/\lambda}^{\pi L/\lambda} \left\langle \frac{|f(\vartheta)|^2}{4} \right\rangle d\vartheta, \end{aligned} \tag{8.92}$$

$$W_{\mathrm{n}}^{\mathrm{sr}} = \widehat{A} \int\limits_{\Omega} \left\langle U_{\mathrm{nsr}}^2(\vec{u}) \right\rangle d\Omega = \widehat{A} A_1^2 I_{\mathrm{n}} 4\pi. \tag{8.93}$$

Hence, the ASG becomes

$$K_{\mathrm{a}} = 10 \lg \left\{ \frac{\left\langle |f(0)|^2 \right\rangle 2\pi L}{\lambda} \left[\int\limits_{-\pi L/\lambda}^{\pi L/\lambda} \left\langle |f(\vartheta)|^2 \right\rangle d\vartheta \right]^{-1} \right\}. \tag{8.94}$$

As follows from (8.94), the behavior of the ASG is mainly defined by the function $f(\vartheta)$ and random variable $\widetilde{\phi}(\tilde{z})$. We further assume that the random variable $\widetilde{\phi}(\tilde{z})$ is normally distributed and has the following statistical characteristics: $\left\langle \widetilde{\phi}(\tilde{z}) \right\rangle = 0$, $\left\langle \widetilde{\phi}^2(\tilde{z}) \right\rangle = \sigma^2 = \widehat{\alpha}$, and

$$R_{\tilde{\phi}} = \left\langle \widetilde{\phi}_1 \widetilde{\phi}_2 \right\rangle / \sigma_1 \sigma_2 = \exp\left(-(\tilde{z}_1 - \tilde{z}_2)^2 / C^2 \right), \tag{8.95}$$

$$C = 2L_{\tilde{\phi}}/L, \tag{8.96}$$

where $\widehat{\alpha}$ is the variance, $R_{\tilde{\phi}}$ is the correlation coefficient, and $L_{\tilde{\phi}}$ is the radius of the spatial correlation of the random variable $\widetilde{\phi}(\tilde{z})$. In this case, it can be shown that (Shifrin 1971; Carey and Evans 1998)

$$\langle f(0)\rangle = \left\langle \int_{-1}^{1} \exp(i\phi(x))dx \right\rangle = \exp(-\widehat{\alpha}/2), \tag{8.97}$$

$$\begin{aligned}\left\langle |f(0)|^2 \right\rangle &= \left\langle \int_{-1}^{1}\int_{-1}^{1} \exp(i(\widetilde{\phi}(\tilde{z}_1) - \widetilde{\phi}(\tilde{z}_2))d\tilde{z}_1 d\tilde{z}_2 \right\rangle \\ &= \int_{-1}^{1}\int_{-1}^{1} \exp\left(-\tilde{\alpha}(1 - R_{\tilde{\phi}}\right)d\tilde{z}_1 d\tilde{z}_2\end{aligned} \tag{8.98}$$

at $L_{\tilde{\phi}} \gg L$, $R_{\tilde{\phi}} \to 1$, and $\left\langle |f(0)|^2 \right\rangle \to 4$, which corresponds to a homogeneous medium, where fluctuations of the received signal phase over the array aperture are absent, and the ASG is defined by expression (8.77). As the array length L increases, an interesting situation arises, when L and $L_{\tilde{\phi}}$ become comparable, $L \le L_{\tilde{\phi}}$, and $C \ge 2$. Here, to estimate the value of $\left\langle |f(0)|^2 \right\rangle$, we can use the following approximate expressions (Carey and Evans 1998)

$$R_{\tilde{\phi}} = \exp\left(-(\tilde{z}_1 - \tilde{z}_2)^2/C^2\right) \cong 1 - |\tilde{z}_1 - \tilde{z}_2|^2/C^2, \tag{8.99}$$

$$\widehat{\alpha}[R_{\tilde{\phi}} - 1] \cong -\widehat{\alpha}|\tilde{z}_1 - \tilde{z}_2|^2/C^2, \tag{8.100}$$

$$\left\langle |f(0)|^2 \right\rangle \cong \int_{-1}^{1}\int_{-1}^{1} \exp\left(-\widehat{\alpha}|\tilde{z}_1 - \tilde{z}_2|^2/C^2\right)d\tilde{z}_1 d\tilde{z}_2. \tag{8.101}$$

We note that the estimate (8.101) at $C>1.5$ and $\widehat{\alpha} \approx 0.5 - 1.0$ differs from the exact value (8.98) by less than 4%.

On the other hand, as Shifrin (1971) showed, at $\pi LC/2\lambda \gg 1$ and $\pi L/\lambda \gg 1$, we have

$$\int_{-\pi L/\lambda}^{\pi L/\lambda} \left\langle |f(\vartheta)|^2 \right\rangle d\vartheta \cong \int_{-\pi L/\lambda}^{\pi L/\lambda} f_0^2(\vartheta)d\vartheta \cong 4\pi, \tag{8.102}$$

$$K_{\mathrm{a}} = K_{\mathrm{a}}|_{\phi(\tilde{z})=0} + 10\lg\frac{\left\langle |f(0)|^2 \right\rangle}{4}, \tag{8.103}$$

where $f_0(\vartheta) \equiv f(\vartheta)|_{\phi(\tilde{z})=0}$ and $K_{\mathrm{a}}|_{\phi(\tilde{z})=0}$ are defined by expression (8.77).

Let us now introduce the definition of the relative signal gain RSG, characterizing the ASG decrease caused by random inhomogeneities, as

$$K_{\mathrm{a}}^{\mathrm{r}} \equiv K_{\mathrm{a}} - K_{\mathrm{a}}|_{\phi(\tilde{z})=0} = 10\lg\frac{\left\langle |f(0)|^2 \right\rangle}{4}. \tag{8.104}$$

Then, taking into account (8.101), $K_{\mathrm{a}}^{\mathrm{r}}$ is written

$$\begin{aligned} K_{\mathrm{a}}^{\mathrm{r}} &\cong 10\lg\left(\frac{1}{4}\int\limits_{-1}^{1}\int\limits_{-1}^{1}\exp\left(-\widehat{\alpha}|\tilde{z}_1-\tilde{z}_2|^2/C^2\right)d\tilde{z}_1 d\tilde{z}_2\right) \\ &\equiv 10\lg\left(\frac{1}{L^2}\int\limits_{-L/2}^{L/2}\int\limits_{-L/2}^{L/2}\exp\left(-|z_1-z_2|^2/{L_h}^2\right)dz_1 dz_2\right), \end{aligned} \tag{8.105}$$

where $L_h = L_\phi/\sqrt{\widehat{\alpha}}$.. The quantity L_h has the meaning of the spatial coherence radius, when the spatial coherence function of the sound field is Gaussian. In a similar way, we can derive an expression for L_h when R_ϕ, hence, the spatial coherence function of the signal exponentially depends on the distance $|z_1 - z_2|$. In this case, $L_h = L_\phi/\widehat{\alpha}$.

Assuming that a sea medium is random inhomogeneous, we can believe that all statistical characteristics of the linear array under consideration depend only on the difference magnitude $|z_1 - z_2|$ and are independent of the coordinates z_1 and z_2 themselves. In this case, we can rewrite formula (8.105) in the more convenient form (Carey and Evans 1998):

$$K_{\mathrm{a}}^{\mathrm{r}} \cong 10\lg\left(\frac{L_h^2}{L^2}\int\limits_{-L/L_h}^{L/L_h}\left(\frac{L}{L_h}-\Delta z\right)\exp\left(-\Delta z^2\right)d\Delta z\right), \tag{8.106}$$

where $\Delta z = |z_1 - z_2|/L_h$. We note that, in the case of a discrete linear array, formula (8.106) can be rewritten as (Carter et al. 1973; Carey and Evans 1998)

$$K_{\mathrm{a}}^{\mathrm{r}} \cong 10\lg\left(\frac{1}{n^2}\sum_{i=-(I-1)}^{i=(I-1)}(I-|i|\exp\left[-(|i|d/L_h)^2\right]\right). \tag{8.107}$$

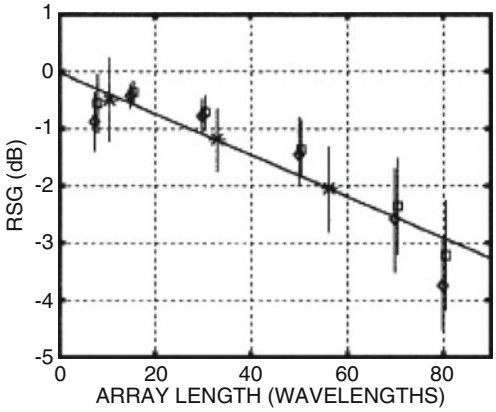


Fig. 8.7 RSG plotted as a function of array length in wavelength for the CW (*Asterisk* 8–12 km) and Impulsive (*Open square* 5–22 km; *open diamond* 23–46 km)

Using (8.106) and (8.107), we can estimate the decrease in the ASG for a linear array, associated with the random inhomogeneities of a medium. In particular, at $L/L_h = 2$, $K_a^r \cong -2$ dB. We note that, in the case of $L/L_h = 2$, a K_a^r value close to $L/L_h = 2$ takes place for the exponential dependence of the spatial coherence function on Δz as well.

On the other hand, (8.106) and (8.107) can also be used to estimate the parameter L_h, which is an important medium characteristic on the continental shelf. As an example, we present the results of measurements of the relative signal gain[6] (see Fig. 8.7), carried out near the coast of South Korea (Rozenfeld et al. 2001). These measurements were performed using both CW and impulse sound sources. Despite a certain difference between the results for different sound sources, which is probably caused by a difference in the signal processing techniques for different sources, K_a^r was -2 dB at an array length of $\cong 58\lambda$. Hence, the estimated spatial coherence radius is $\cong 29\lambda$. We note that this value is close to the typical value of L_h for the sea shelf. Of course, this value depends on the distance to the sound source inducing the sound field at the receiving array position. In particular, a review of available experimental data in Carey et al. (2006) allows the conclusion that on the average $L_h \cong 30\lambda$ at distances of 9–45 km and in a frequency range of 50–800 Hz. This value is typical of the shallow-water acoustic waveguide in the summer time with sharp thermocline. In this case, the random perturbations of the medium affecting sound propagation are caused by internal waves and inhomogeneities of the upper layer of bottom sediments. One also notes that L_h shows a distribution function in time, which can be gleaned when looking at long enough time series (Duda et al. 2011).

[6] We recall that K_a^r is determined by relation (8.78), whereas K_a is measured experimentally, and $K_a|_{\phi(\tilde{z})=0}$ is calculated by (8.77) and (8.80).

8.4 Mode Filtration

We know that the sound field in the shallow water waveguide at comparatively low frequencies is a composition of waveguide modes (3.25). The number of modes in real situation can be up to a few tens. For example, for the point source in the simple case of waveguide with nonvariable parameters complex sound field at fixed frequency is

$$P(r,z,t) = \sum_l C_l \Psi_l(z) \exp[i(q_l r - \omega t)] \tag{8.108}$$

where the coefficients C_l are the modal amplitudes at the point of observation. Cylindrical spreading, attenuation due to the bottom and water layer absorption processes are included in the coefficients C_l. For many theoretical and practical goals, it is necessary to know different parameters of the separate modes: wave vectors and attenuation coefficients, shape of eigenfunctions, group velocities, the energy, etc., carried by separate modes. Mathematically, it means one needs to find coefficients C_l (for broadband signal they depend on frequency), the values $\xi_l = q_l + i\gamma_l/2$ and the functions $\Psi_l(z)$. This can be done on the basis of the measured signal signatures at the vertical or horizontal arrays. These methods and tools are described by the term "modes filtering."

It has long been known that one can filter normal modes (i.e., receive a particular mode to the exclusion of the others) using the standard beamforming techniques of (1) amplitude shading a vertical array, (2) steering a vertical or horizontal array, or (3) focusing a horizontal array (Clay and Medwin 1977; Lynch 1983). In this section, we will look at these techniques, discuss their realms of usefulness, and show some examples. Modes filtering is also important to acoustic tomography, which uses the measurement of parameters of sound signals of separate modes. Modal acoustic tomography is an extension of the original idea of ray acoustic tomography of the ocean (Munk 1979) which can be applied to the shallow sea (Kravtsov and Petnikov 1986; Zaitsev et al. 1989). It is based on the separation of acoustic signals corresponding to individual modes and the measurement of their propagation times. In this case, from the viewpoint of signal processing techniques, the major difficulty in implementation of modal acoustic tomography is just associated with separation of signals corresponded to the most energetic first modes. In some shallow water areas, for not too large vertical gradients of the sound speed profile (such a profile can occur, e.g., in the Barents Sea), low mode signals propagate with approximately the same velocity. In these cases, vertical receiving arrays (VRA) were first experimentally used to separate mode signals (Miller et al. 1993; Zverev et al. 2006). It should also be noted that it is possible to use vertical emitting arrays for modal acoustic tomography as well (Luchinin and Khil'ko 2005), which provide excitation of a given mode group in a waveguide.

Mode selection using amplitude shading. In that this topic has been addressed elsewhere in this book, we will only highlight those aspects that have not been discussed.

As was discussed, mode filtration using amplitude shading of a vertical array works by exploiting the orthogonality integral for the normal modes (3.11). When the array element responses are adjusted to match the vertical amplitude pattern of a particular mode (at a given frequency), the array produces a maximum power output for that mode, but (almost) no power from the other modes.

So if amplitude of the sound field at the fixed distance is (8.69), then modal amplitude has the form

$$C_l \cong \int_0^H P(r,z)\Psi_l(z)dz. \tag{8.109}$$

(Here we assume that the right side of (3.11) close to zero.) The amplitude $P(r,z)$, corresponding to (3.25) does not contain factor $\mathrm{e}^{-\mathrm{i}\omega t}$. Practically, for using this formula we should know the eigenvalues and waveguide modes and also the depth distribution of the sound field for a fixed range. The eigen functions $\Psi_l(z)$ can be found via solution of the Sturm–Liouville problem (3.7), or from experiment.

This mode selection scheme has proven very useful in experimental situations, where vertical arrays are popular. However, it is not without shortcomings, and we discuss those next. The principal effects that degrade the performance of this particular modal beamformer are:

1. Finite aperture effects (will be as shown below)
2. The measured sound field $P(r,z)$ can only be known at a finite number of depths z_i (depths of hydrophones), so instead of the integral (8.109) we have a sum
3. Finite SNR
4. Unaccounted for water column fluctuations
5. Unaccounted for array distortions
6. Hydrophone calibration errors (hydrophone sensitivity)

The fact that we use the approximate formula (8.109) or that the vertical carry does not span the entire interval $0 \le z \le \infty$ [see (B.32) and (B.33)] means that one is truncating the orthogonality integral that is the basis of the technique, which will introduce some form of truncation error. In practice, moored vertical arrays usually avoid the top 10 m of the water column (due to shipping and surface wave induced motion) and the bottom (due to penetrability). As has been shown in detail in, e.g., Lynch (1985), this truncation leads to reduced power in the mode of interest, which is compensated for by leaking power into unwanted modes. Thus, mode resolution degrades as the aperture decreases, a well-known effect.

We consider influence of truncated array via a simple example, using the waveguide shown in Fig. 8.8 and six trapped modes seen in Fig. 8.9. In order to illustrate influence of reduced length of an antenna, we compare the integrals $A_{lm} = \int_0^H \Psi_l(z)\Psi_m(z)dz$ and $\tilde{A}_{lm} = \int_0^{\tilde{H}} \Psi_l(z)\Psi_m(z)dz$ where $\tilde{H} = 0.8H$. This comparison is given in Fig. 8.10 in the

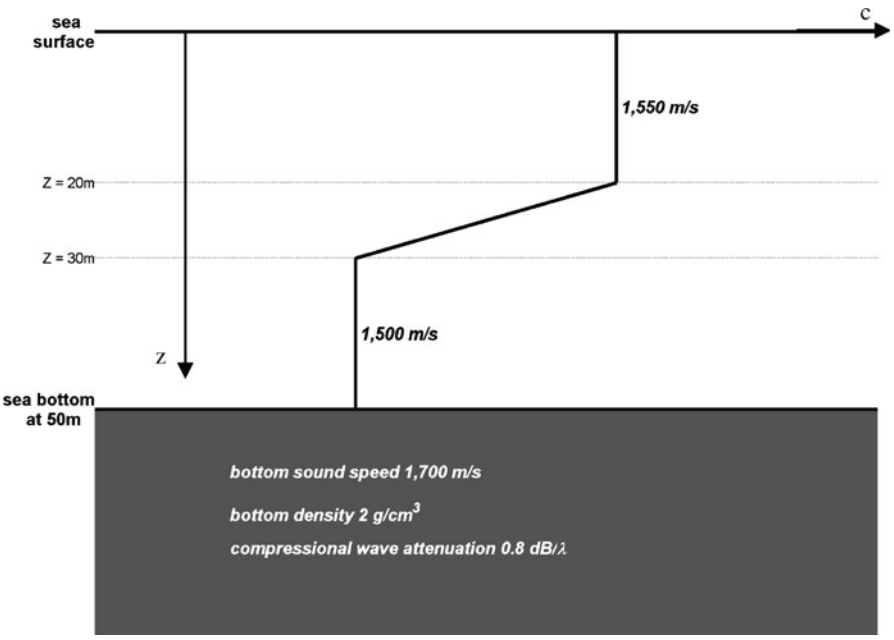


Fig. 8.8 Downward refracting waveguide used to generate six trapped modes used in examples below

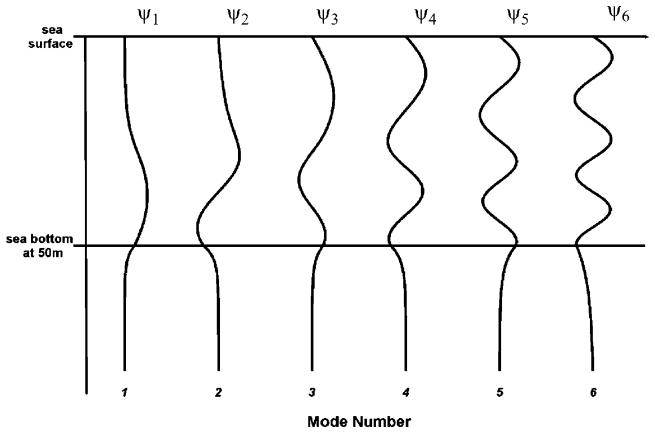


Fig. 8.9 Six trapped modes used for examples of mode filtration. Frequency is equal to 200 Hz

form of diagram. Here the values of A_{lm} and $\tilde{A}_{lm}$ correspond to the lengths of vertical sections at the points with Cartesian coordinates (l, m) for an array spanning the full water columns (dots) and an array spanning 80% of the water column (missing the top portion) (asterisk). It is seen that the array spanning the full water column sees all the energy (minus a small fraction in the bottom), whereas the truncated array

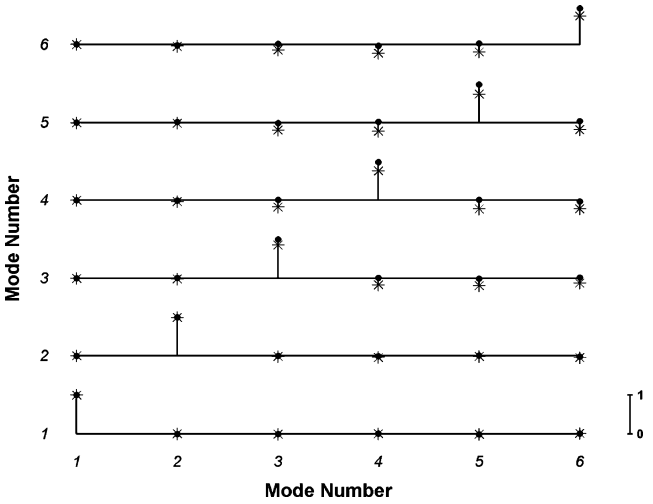


Fig. 8.10 Test for orthogonality of waveguide modes. The *dots* are the result from a full water column scanning array. The *asterisks* are the result of a truncated array

misses seeing some energy near the surface. This leads to the truncated array filtering the low modes well (as their energy is concentrated below the thermocline), but not doing as well for the highest modes, which have significant near surface energy.

Finite SNR presents yet another limitation on mode forming, particularly for higher modes. The higher order modes tend to have lower SNR simply due to their higher attenuation (mode stripping). But, additionally, they have spatially closer nulls, which are less easy to resolve at low SNR, as the noise fills the nulls in.

Unaccounted for water column sound speed fluctuations cause a "mismatch" of the replica mode we use to create the amplitude shading, and thus one also loses power in the mode desired to "sidelobe leakage" to other modes. Unaccounted for array tilt/deformation has a similar effect, in that this causes one to "misplace" the proper amplitude shading.

Another common source of error in this type of mode filtration is hydrophone calibration error (due to lack of calibration!) This again causes spurious amplitude shading.

Interestingly, a failure to Nyquist sample[7] the vertical oscillations of the mode functions is *not* a fatal error in this technique. Rather, the rule of thumb (Clay and Medwin 1977) is that one must have as many hydrophones as one has modes; but

[7] Notice that sampling the instantaneous sound field distribution along a vertical array by using a set of hydrophones is similar to digitization of the continuous electrical signal as described in Sect. 8.1. With equidistant hydrophone spread, the wavenumber $F_N^S = 0.5/l_a$ plays the role of the Nyquist frequency. Here l_a is the distance between hydrophones.

their spatial distribution is unspecified. One could in theory put all the hydrophones in a very short segment and still filter modes. This rather odd property is due to the fact that one can, in theory, discriminate mode eight from mode one (or any other mode) based on measuring a fraction of a vertical oscillation cycle, as each individual mode function segment has its own distinct curvature. This works in theory for infinite SNR, but in practice is a poor idea! Measurements of the alternations in sign of the amplitude of the mode functions or of the positions of mode nulls are much more robust than measurements of curvature over a short segment.

Mode separation using steered and focused arrays. Another very popular and common way of filtering normal modes in shallow water acoustics is via use of a steered array, which can be either vertical or horizontal in orientation (Clay and Medwin 1977). Each of the two orientations has its own advantages and drawbacks, so we will look at each one separately. We will start with the vertical array.

While the water depth H places a hard limit on the array length (~100 m for a typical shelf), the vertical geometry is still useful for steered beamforming. Beamformed output ($B^{\mathrm{v}}_{\mathrm{a}0}(\chi)$) is equal to

$$B^{\mathrm{v}}_{\mathrm{a}0}(\chi) = \left| \int_0^H P(r,z) \mathrm{e}^{-ikz \sin\chi} dz \right| \tag{8.110}$$

where χ can be considered as the grazing angle and k is the wave vector. In accordance with the transverse modal structure of the sound field, this beamformer should have maxima at the points (in grazing angle) corresponding to the modes.

The 3 dB beamwidth (in degrees) of ~50 λ/D (Urick 1975) comes to ~3° at 150 Hz for a 100 m long array, which is adequate to at least separate low modes (0–10° grazing angle) from high modes (10°–20° or 10°–30°, depending on critical angle.) Also, the mainlobe is at broadside (0° grazing angle) for this geometry, and so has the smallest width right in the heart of the trapped mode angular regime. Additionally, the vertical steered array geometry allows one to look equally well in any azimuthal direction, an advantage not shared by the horizontal geometry.

In terms of defects, the limited aperture is perhaps the most restrictive. One also needs to account for the vertical variability in the sound speed profile when doing the beamforming, but this is a simple correction and often one does not even need to use it (rather, an average of $c(z)$ is adequate). Array tilt and distortion can bias the beam direction estimate or broaden the mainlobe slightly, but overall, does not seriously hurt performances. Given a broad beam to begin with, small fluctuations in it can mostly be ignored.

Horizontal steered arrays are commonly used by submarines, seismic exploration ships, and academic researchers (the latter often using the synthetic aperture arrays). Steering a horizontal array in shallow water produces peaks at the angles corresponding to the individual normal modes, provided the source is not broadside to the array. While it is possible to have a very long real aperture array in shallow water

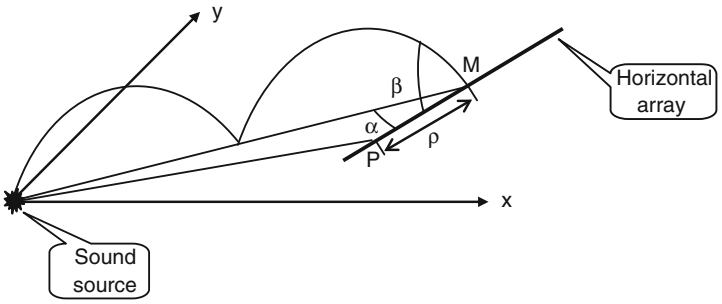


Fig. 8.11 Steered horizontal array in the shallow water waveguide

(and near surface seismic streamers can be quite long), deeper towed arrays have to worry about dragging on the bottom, and so in practice are limited in aperture. A solution to this shortcoming, useful for cases where the medium does not change over the course of the observation and there is no appreciable motion of the source, is to create a synthetic aperture array. This is often done by having a source move away from a receiver (or vice versa) and then using the separation distance as a virtual array element position. This technique has been employed by numerous researchers, but we think a somewhat novel variant of it is worth discussing here.

Let's state the problem in the following form (Fig. 8.11). Let us have a horizontal array of the length L, placed at the bottom ($z = H$) at a distance r from the source (we define the distance to be to the middle point M of the array). The angle between the direction of sound propagation and the direction of the antenna is taken to be α. The main entity which we construct to process acoustical data received by an array is the beamformed output of the array, which is defined as the intensity spatial Fourier transform over the antenna's length:

$$B_{a0}^{h}(\alpha,\beta) = \left| \int_{-L/2}^{L/2} P(\vec{r}+\vec{\rho},H)\mathrm{e}^{ik\rho\cos\beta}d\rho \right| \tag{8.111}$$

where we use variable ρ as the distance from point of observation P to the middle point along the antenna $-L/2<\rho<L/2$.

If we suppose that the distance to the source is much bigger than the length of the antenna ($r \gg L$) or $|\vec{r}+\vec{\rho}| \sim r - \rho\cos\alpha$ and neglect the curvature of the sound wave front (it is supposed to be plane and orthogonal to $\vec{r}$ so that the phase shift at the point P is $q\rho\cos\alpha$), then using the modal decomposition of the sound field we can get

$$B_{a0}^{h}(\alpha,\beta) = \left| \sum_{l} \frac{C_l \sin[(q_l\cos\alpha - k\cos\beta)L/2]}{(q_l\cos\alpha - k\cos\beta)L/2]} \Psi_l(H)\mathrm{e}^{iq_l r} \right|. \tag{8.112}$$

We consider $B^{\mathrm{h}}_{\mathrm{a}0}(\alpha, \beta)$ as a function of two angles, the angle α characterizing the direction of the antenna in the horizontal plane and the hypothetical angle β (see Fig. 8.11).

Properties of the beamformer output are directly related to the problem of source localization. For example, in free space we had only one term in $B^{\mathrm{h}}_{\mathrm{a}0}(\alpha, \beta)$, which has a maximum at $\beta = \alpha$, where the ray path coincides with a straight line joining the source and the receiver and β gives the direction to the source. In the case of a waveguide, $B^{\mathrm{h}}_{\mathrm{a}0}(\alpha, \beta)$ (8.112) has a number of maxima where $q_l \cos\alpha - k\cos\beta = 0$ or for the set of angles $\cos\beta_l = (q_l/k)\cos\alpha$. Each maximum corresponds to individual mode. What this means is we can select individual modes by tuning to these separate maxima.

The geometrical meaning of the angle β is shown in Fig. 8.11. A mode can be described by a Brillouin ray path, and β is the angle between the Brillouin ray and the direction of an antenna. We thus can find the direction to the source (the angle α) if we know the modal eigenvalues q_l.If we know the source position, we can also infer the modal eigenvalues (if we can resolve and identify the modal peaks). From yet another viewpoint, if both values of the q_l and the location of the source are unknown, we can only determine the relative positions of the maxima in the spectrum.

Next, we note that our chances of resolving the maxima in $B^{\mathrm{h}}_{\mathrm{a}0}(\alpha, \beta)$ (or in the spatial spectrum) depend on the angle α. We see from (8.112) that if $\alpha = 0$ (the antenna is directed at endfire onto the source) then $\cos\beta_l = (q_l/k)$ and the angles β_l are equal to χ_l, where χ_l are the grazing angles of the Brillouin rays, corresponding to the waveguide modes. The usual range of their values is from zero to the total reflection angle, though higher angle "virtual modes" can also be seen under some conditions. Increasing the angle α (going to a broadside geometry) makes the resolution of the modal angles decrease to zero: if $\alpha = \pi/2$ all the angles $\beta_l = \pi/2$ as well.

In Fig. 8.12, we show output of an array for the situation where the waveguide model sound speed profile is as shown in Fig. 8.8. We see distinct maxima in the angle distribution, corresponding to the eigenvalues of the waveguide. What this means is that we can select the waveguide modes by using beamforming. Again, we note that if $\alpha = \pi/2$ then this way of processing does not work, and in contrast the best geometry for this method is $\alpha = 0$, when the horizontal array is directed along the acoustic track.

This example also suggests the usefulness of this simple beamformer in discriminating between near-surface sound sources and near-bottom sound sources in a downward refracting environment. In such an environment, near surface sources excite high order modes, but not low modes, whereas near bottom sources excite both types of modes. (This is related to the "noise notch" we will discuss in Chap. 9). By steering the array through all angles, from 0° to past critical angle, we can look directly at the distribution of energy between high and low modes. This is shown via a numerical example in Fig. 8.12. We note that it is not necessary to resolve the modes individually in angle in order to discriminate between having more energy in the high versus the low modes.

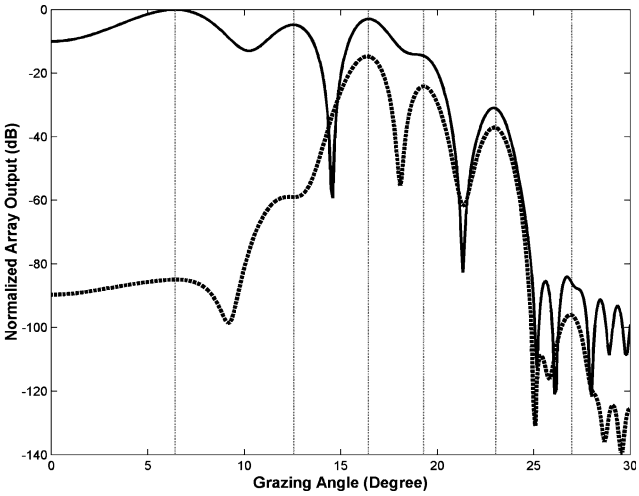


Fig. 8.12 Output of steered horizontal array in shallow water for array directed onto the source $\alpha = 0$. *Solid line* is for source near surface, *dotted line* is for source near bottom



Fig. 8.13 Deployment of REMUS AUV carrying a towed acoustic array in shallow water off Nantucket Sound, MA

We next note that, if we have stable unmoving source, radiating a signal over a long time, we can collect acoustical data along rather long distance using single receiver moving along an acoustic track. The processing is the same as that we applied for the real horizontal array, but our "effective array" will usually be much longer. In Fig. 8.13, we show a REMUS 100 AUV which has a small (6 m active acoustic segment) towed array trailing behind it. (REMUS 100 AUV is described in Chap. 10 more detail.) This AUV/towed array system was able to create a synthetic aperture array of 4 km length with very little "distortion" of the array due to vehicle or array towing effects, and perhaps represents the most precise synthetic aperture horizontal array measurement system available today.

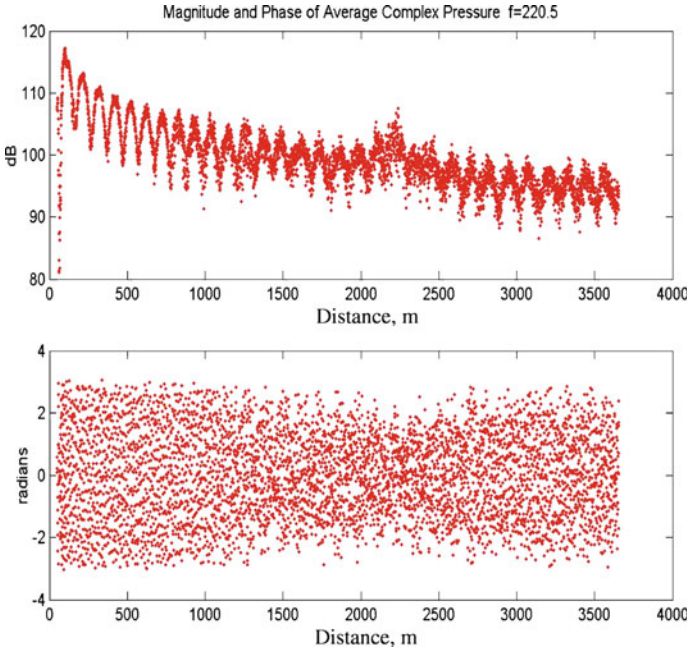


Fig. 8.14 (*top panel*) Magnitude of the pressure field versus range measured by REMUS AUV and a towed array off Nantucket Sound, MA. Two mode interference pattern is clearly evident. (*bottom panel*) Phase of the pressure field in *top panel*

The $P(r)$ field produced by the array (listening to a fixed source at 220.5 Hz) is shown in Fig. 8.14. A clear, two mode interference pattern is seen over the experiment's range (with a little noise in the middle due to a passing ship).

By Hankel transforming the pressure field, an operation basically equivalent to array steering, one sees distinct peaks at the horizontal wavenumbers corresponding to the modes (see Fig. 8.15).

There are two shortcomings of horizontal steered arrays that we should mention. The first, already discussed, is that real aperture systems must be used for tracking quickly moving sources, and such systems are of limited aperture in practice. The second shortcoming is that the low modes are seen at endfire with such an array, which means a broader, lower resolution beam. As the low modes have the closest spacing in horizontal wavenumber, this makes their resolution more problematic. This is shown for a two-mode system in Fig. 8.16.

Normal modes can also be resolved by using a horizontal linear array (HLA) with a source placed in the nearfield of the array, at broadside aspect (optimally). If the source is within $L^2/2\lambda$ of the array, wavefront curvature effects are pronounced, and one has to focus the array to produce a sharp image of the source. In doing this with a line array (not a necessity – other array shapes also work) (Lynch 1983, 1985),

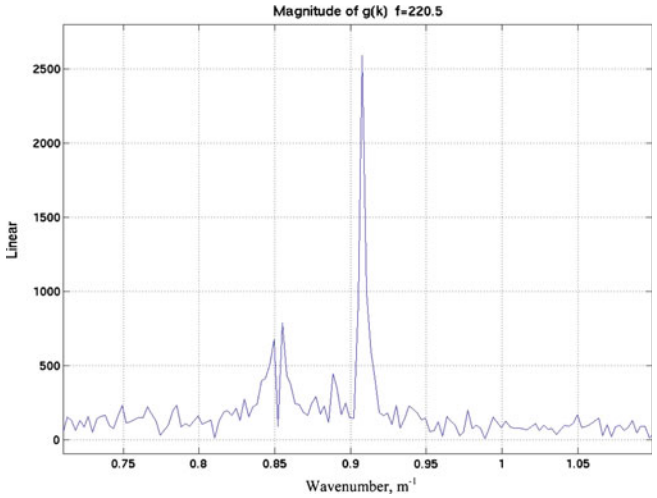


Fig. 8.15 Hankel transform of the pressure field in the previous figure, showing the horizontal wavenumbers (m^{-1}) of the two trapped modes

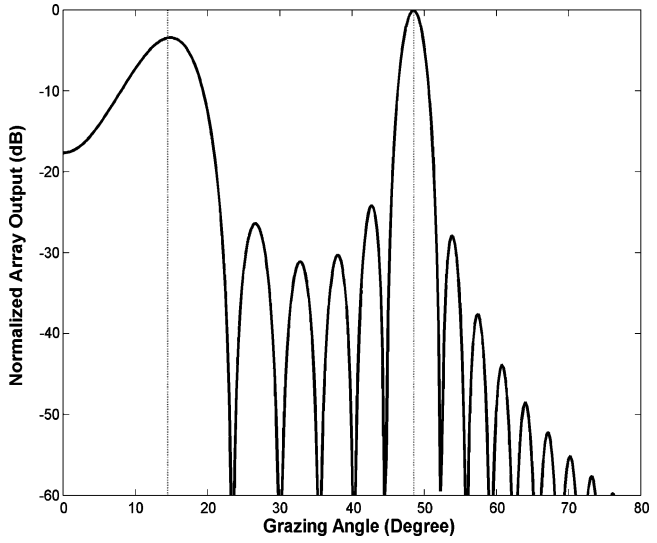


Fig. 8.16 Output of a steered, endfire array in a waveguide with two trapped modes. Note the broad beamwidth at low angles

one cancels the phase accumulation from the source to each of the $i = 1, 2, \ldots I$ array elements. A quadratic phase variation across the array minimizes this phase curvature to first order. However, this cancels the phase curvature only for a free space point source at distance ρ from the array for the wavenumber $k = \omega/c$.

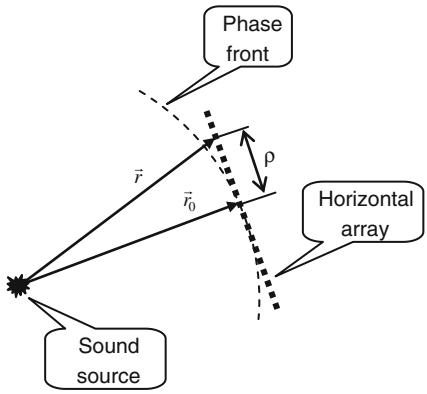


Fig. 8.17 Scheme of the focused array. *Dashed line* denotes circular wave front, *dotted straight line* denotes horizontal array, oriented perpendicular to an acoustic track

Let's consider the working principles of a focused HLA in more detail, when the HLA is placed in a zone where the curvature of the wave front should be taken into account (Fig. 8.17). In the simplest case, when HLA is perpendicular to an acoustic track case we have some phase mismatch at the antenna due to curvature of the wave front in comparison with a plane wave $\delta\varphi = k(r - R) = k\rho^2/2r$ and so we introduce the beamformed output for focused array

$$B^{\mathrm{h}}_{\mathrm{a}0}(r, r_0) = \left| \int_{-L/2}^{L/2} P(\vec{r} + \vec{\rho}, H) \mathrm{e}^{-ik\frac{\rho^2}{2r_0}} d\rho \right|. \tag{8.113}$$

In free space, this output can provide the distance to the source, because the function $B^{\mathrm{h}}_{\mathrm{a}0}(r, r_0)$ has maximum if $r = r_0$.

In a waveguide, the sound pressure under the integral (8.113) is sum of waveguide modes, and after some simple calculations we have

$$B^{\mathrm{h}}_{\mathrm{a}0}(r, r_0) = \left| \sum_l C_l \Psi_l(H) \mathrm{e}^{iq_l r} \int_{-L/2}^{L/2} \exp\left[i \left(\frac{q_l}{r} - \frac{k}{r_0} \right) \rho^2 \right] \right| d\rho. \tag{8.114}$$

Similar to previous case, we see that this dependence has a number of focal maxima, corresponding to different modes at the points $r_0^l = kr/q_l$. Corresponding larger apparent focal distances simply reflect the added path length the modal multipaths see compared to the straight line path, a straightforward geometric effect. A numerical example of this is shown in Fig. 8.18. As illustrated in this figure, we can separate individual maxima. Again, this means that we can select the waveguide modes by using beamforming. We note that this form of mode filtration is rarely used, in that one rarely has a source extremely close to a large aperture array.

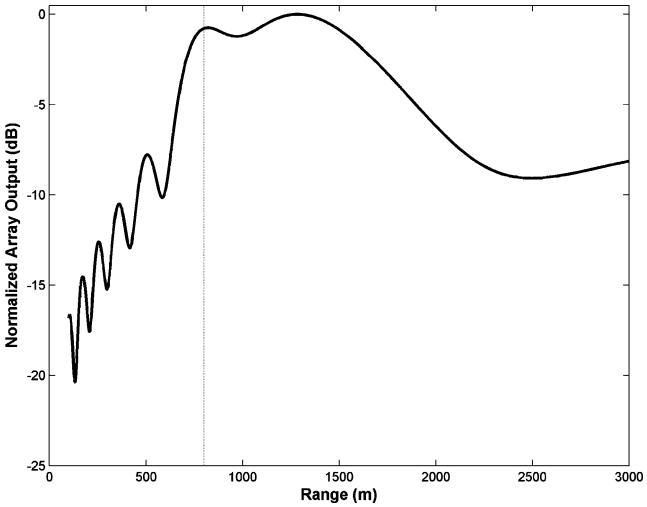


Fig. 8.18 Output of focused beamformer in shallow water waveguide with two modes present

Finally, we can write the expression for the beamformed output in the general situation, joining steered and focused arrays:

$$B^{\mathrm{h}}_{\mathrm{a}0}(\alpha, r, \beta, r_0) = \left| \int\limits_{-L/2}^{L/2} P(\vec{r} + \vec{\rho}, H) \mathrm{e}^{-\mathrm{i}k\left(\rho \sin\beta + \frac{\rho^2}{2r_0}\right)} d\rho \right| \tag{8.115}$$

which can be calculated either in free space or in a waveguide.

As a final technique, we note that the mode filtration also can be performed by using waveguide dispersion properties. In other words (see Sect. 4.5) we can separate individual modes by using the difference in arrival time for signals corresponding to these modes. With vertical chain of receivers, this method can be used as well even though you do not know the group velocities and eigenfunctions $\Psi_l(z)$ exactly. Let us clarify this statement by using data from the SWARM'95 experiment. As was demonstrated earlier (see Fig. 4.15c), we obtained mode signals $\tilde{P}(\vec{r}, z, t, \omega_{\mathrm{c}})$. Specifically, the processed signal can be written as follows[8]:

$$\tilde{P}(\vec{r}, z, t, \omega_{\mathrm{c}}) = 2 \int\limits_{0}^{\infty} F(\omega, \omega_{\mathrm{c}}) S_r(\vec{r}, z, \omega) \mathrm{e}^{-i\omega t} d\omega \tag{8.116}$$

[8] Here we used expression (8.116) instead of (4.88) because of we did not know the spectrum of the radiated signal in this experiment. As was mentioned before (Sect. 4.3), expression (4.88) allows us to select the mode signal in the best way.

where $S_r(\vec{r}, z, \omega)$ is the spectrum of the received signal. Recall that

$$S_r(\vec{r}, z, \omega) = \Psi(\vec{r}, \omega, z) S(\omega). \tag{8.117}$$

Here, $F(\omega, \omega_c)$ is the frequency characteristic of a band pass filter with band center ω_c. In particular, $F(\omega, \omega_c)$ is equal:

$$F(\omega, \omega_c) = \frac{1}{\Delta\omega\sqrt{2\pi}} \exp\left\{ -\frac{(\omega - \omega_c)^2}{2\Delta\omega^2} \right\}, \tag{8.118}$$

where $\Delta\omega$ is the bandwidth and $\Delta\omega \ll \omega_c$.

These mode signals are shown in Fig. 8.16 as well for $\omega_c/2\pi = 60$ and 90 Hz. Recall that the maxima in arrival times show the arrivals of different mode signals due to their differences in group velocities, which can be used for mode filtering. After integration of the pulse envelope (amplitude) we get a function of depth and frequency ω_c, which is proportional to the amplitude of mode l, denoted as $\tilde{\Psi}_l(z, \omega_c)$:

$$\tilde{\Psi}_l(z, \omega_c) = \int_{t_l}^{t_l + \Delta t} |\tilde{p}(\vec{r}, z, t, \omega_c)| dt. \tag{8.119}$$

We should compare these eigenfunctions obtained from experimental data with the theoretically calculated values $\Psi_l(z, \omega_c)$. For this we should normalize the selected modes using some relationship. To normalize experimental functions to compare with the analytical ones, we use

$$\tilde{\Psi}_l^n(z, \omega_c) = \tilde{\Psi}_l(z, \omega_c) \Big/ \int_0^H \left[\tilde{\Psi}_l(z, \omega_c) \right]^2 dz. \tag{8.120}$$

The results of mode filtering for the modal shape using the data shown in Fig. 8.19c are presented in Fig. 8.20. Good agreement between experimentally extracted and theoretically calculated modes is obtained. What this means is we have additional method for identification of mode signals by comparing $\tilde{\Psi}_l^n(z, \omega_c)$ and $\Psi_l(z, \omega_c)$. In other words, we can place the mode number and the number of the received mode signal into a one-to-one correspondence because the depth dependence of $\Psi_l(z, \omega_c)$ represents a rough but stable characteristic way to determine mode number uniquely. Recall that $\mathrm{Re}\Psi_l(z)$ has l zeros, not counting the first one, remembering the theorem on the number of zeros of the eigenfunction (Titchmarsh 1958).

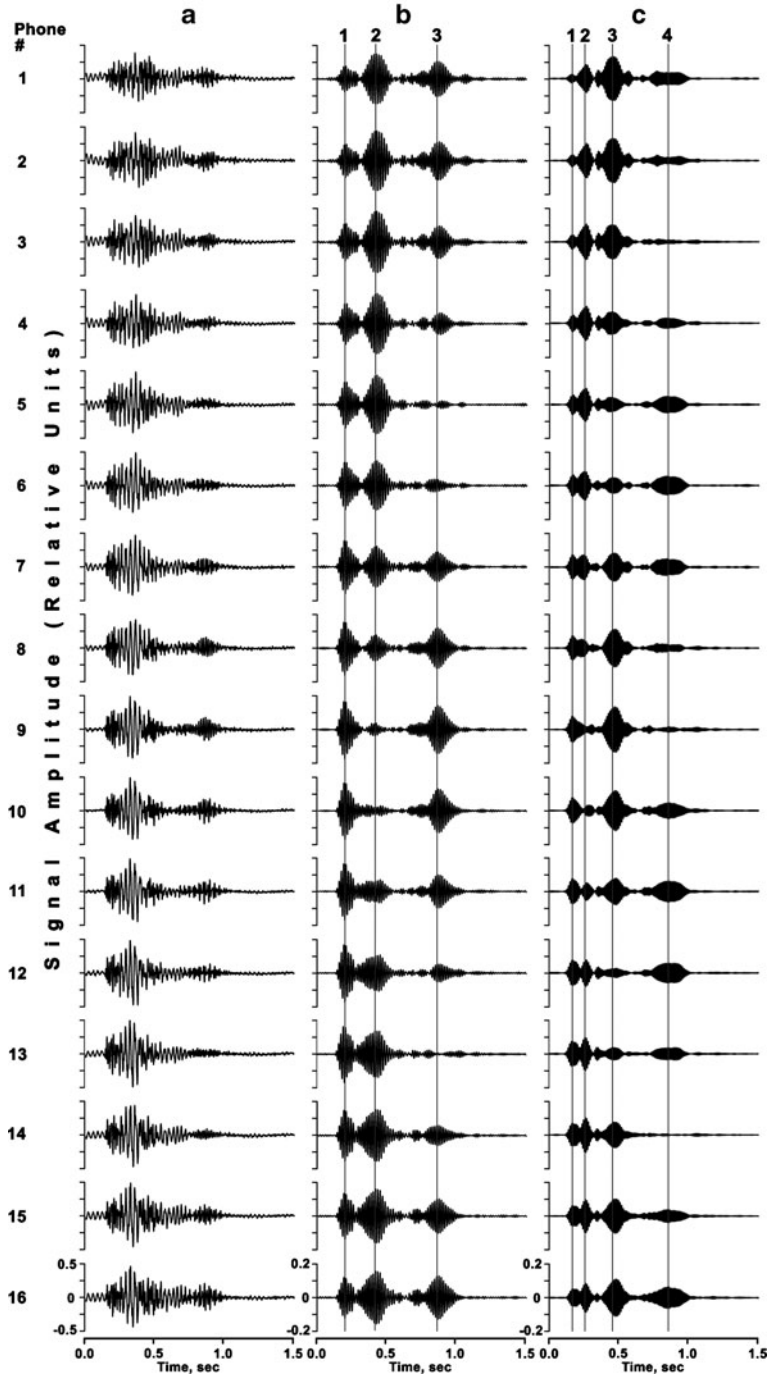


Fig. 8.19 Received airgun signals for all 16 hydrophones of VLA (**a**) raw data, (**b**) band pass filter $\omega_c/2\pi = 60$ Hz, (**c**) band pass filter with $\omega_c/2\pi = 90$ Hz

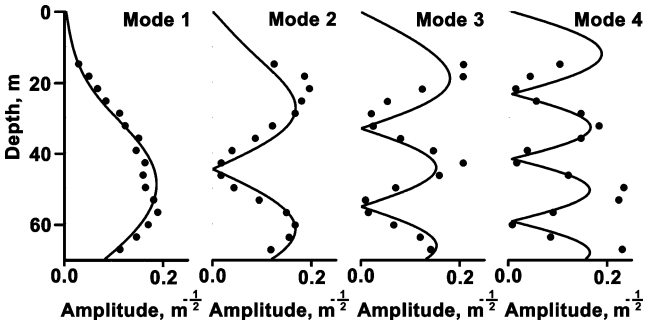


Fig. 8.20 Depth dependence of the first four modes. *Dots* show averaged values of experimental data $\tilde{\Psi}_l^n(z)$ and the *solid lines* show the theoretical calculations $\Psi_l(z)$ based on averaged sound speed profile

8.5 Time-Reversal Mirror of the Sound Field in Shallow Water

Recently, the time reversal of waves has been successfully used for sound field focusing in shallow water acoustics (Dargeiko et al. 1984; Jackson and Dowling 1991, 1992; Dowling 1993, 1994; Kuperman et al. 1998). This method, which in its broadband version is called the "time-reversal mirror," is based on the CW technique of acoustic wave phase conjugation. From the historical viewpoint, it originates from the well-known phenomenon of wave phase conjugation used in nonlinear optics (Zel'dovich et al. 1985), and was first employed in underwater acoustics in the time reversed experiments of Parvelescu and Clay (1965).

The essence of the method, i.e., its version adapted to applications, consists in using two spatially coincident arrays, which span most of the acoustic waveguide, together with a "probe sound source" positioned at the assumed point of focusing. The arrays are vertical arrays of hydrophones and sound sources, i.e., receiving and emitting arrays [Generally, these two arrays are referred to by the single term "source/receiver array" (SRA)]. With this geometry, sound field focusing is reduced to three sequential operations (1) reception of the signal from the "probe sound source" at each hydrophone, (2) time reversal of the received signal (in oceanic acoustics at frequencies below several kilohertz, this is achieved using computer data processing, where the main element is the calculation of the complex–conjugate spectrum of the received signal[9]), and (3) emission of the time-reversed signal.

Let us consider such focusing in more detail, applied to a shallow water acoustic waveguide. As was done in Kuperman et al. (1998), we first analyze focusing of the

[9] We note that in ultrasonics, at frequencies of several megahertz, this sound field focusing method is also well known (see, e.g. Fink 1995). Focusing is achieved simply by reproducing a digital copy of the received signal in the reverse sequence.

monochromatic sound field, when a probe source (PS) at the point with coordinates r_0, z_0 emits a simple harmonic sound signal with frequency ω. To simplify the expressions used, we assume that the SRA is positioned at the coordinate origin and its receiving elements are arranged at points $(0, z_j)$, where j is the receiving hydrophone index. Then, according to (3.25), the received signal is given by

$$U(0, z_j, t) = N_\mathrm{r}\sqrt{\rho_0 c_0 W_0}\sum_l \frac{\Psi_l(z_0)\Psi_l(z_j)}{\sqrt{\xi_l r_0}}\exp[i(q_l r_0 - \omega t + \pi/4) - \gamma_l r_0/2] \tag{8.121}$$

where N_r is the sound pressure-to-electrical signal conversion factor of the receiver.

In this case, according to (3.3), the time-reversed sound signal transmitted by the jth array element can be written as

$$P_\mathrm{r}(0, z_j, t) = N_\mathrm{s}\mathrm{Re}(U(0, z_j, -t)\frac{\exp(ikz_j|\vec{R}|)}{4\pi|\vec{R}|} \tag{8.122}$$

where N_s is a conversion factor for the source, which depends on the source power. Once again, using formula (3.25), we can write the total sound signal from all sources at distance r from the array and at depth z as

$$P(r, z, t) \cong \hat{K}\sum_j\sum_m\sum_l \frac{\Psi_l(z_0)\Psi_l(z_j)\Psi_m(z_j)\Psi_m(z)}{\sqrt{\xi_l\ \xi_m r r_0}}\exp[i(q_m r - q_l r_0 - \omega t] - (\gamma_m r + \gamma_l r_0/2)]. \tag{8.123}$$

We note that (8.123) ignores the dependence of the sound speed and density on depth at the place where the array is positioned. The proportionality factor $\hat{K}$ (8.123) is given by $\hat{K} = N_\mathrm{s}N_\mathrm{r}\sqrt{\rho_0 c_0 W_0}$. Taking into account the eigenfunction normalization (3.10) and orthogonality (3.11) conditions, as well as neglecting sound attenuation, we can simplify (8.123) to read

$$P(r, z, t) \cong \hat{K}\rho\sum_m \frac{\Psi_m(z_0)\Psi_m(z)}{q_m\sqrt{r r_0}}\exp[iq_m(r - r_0) + i\omega t]. \tag{8.124}$$

There is one more important relation that can be exploited now, the completeness of the system of eigenfunctions, which is stated as:

$$\sum_m \frac{\Psi_m(z)\Psi_m(z_0)}{\rho_0} = \delta(z - z_0). \tag{8.125}$$

Substitution of (8.125) into (8.124) shows clearly that the sound field at $r = r_0$ is localized at depth $z = z_0$, i.e., focusing occurs, displayed by

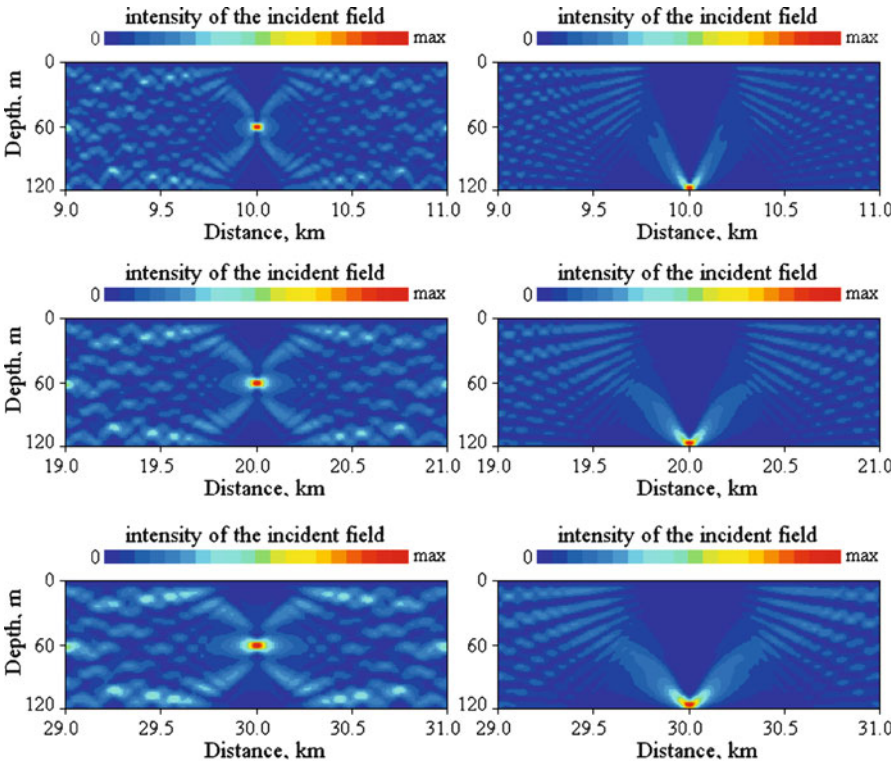


Fig. 8.21 Intensity of the focused simple harmonic sound field in the waveguide at the different distances from SRA and at the different depths

$$P(r_0, z, t) \approx \hat{K}\rho\rho_0 \frac{\delta(z - z_0)}{k} \exp(\mathrm{i}\omega t). \tag{8.126}$$

Relation (8.126) is valid under one more important assumption: that the sound field at distance r_0 is formed by a limited number of the low, energy-carrying modes, for which the eigenvalues are approximately equal to each other and are close to the water wavenumber k.

Even without worrying about or adhering to all the above-mentioned assumptions, sound field focusing is confirmed easily by numerical simulation results. As an example, Fig. 8.21 shows focusing in a waveguide corresponding to the Pekeris model, which includes accounting for sound attenuation in the bottom. The model parameters are as follows: the waveguide depth is $H = 120$ m, the speed of sound in the waveguide is 1,475 km/s, and the refractive index in the bottom is $n_1 = 0.84(1+0.006\mathrm{i})$. The signal frequency used is $f = 230$ Hz. It was assumed that focusing is achieved using a uniformly spaced SRA spanning the whole waveguide. The distance between array elements is 3 m; the number of elements used is 41. Another example of focusing in a similar waveguide, with the

sound speed profile shown in Fig. 4.16 and a vertical array arranged below the thermocline, is shown in Fig. 6.11 (the array length here is 87 m). We can see that "adequate" focusing takes place even for the case of an array not spanning the whole acoustic waveguide.

Let us next consider the case (important for acoustic applications) of focusing of an emitted pulsed signal of duration τ (Kuperman et al. 1998). Let us also assume that it is a simple signal, whose spectral width is $\Delta\omega \cong 2\pi/\tau$. To write the signal received by the jth hydrophone, we invoke the (A.165)

$$u(0, z_j, t) = N_r A \int_{-\infty}^{\infty} \Psi(0, r_0, z_0, z_j, \omega) S(\omega) \exp(-i\omega t) d\omega, \tag{8.127}$$

where $\Psi(0, r_0, z_0, z_j, \omega)$ is the Green's function[10] defined by relation (3.16), which depends on frequency as a parameter; A is the coefficient calculated using relation (3.24), and $S(\omega)$ is the spectrum of the PS signal. We note that (8.121) is a special case of (8.127), for the case of a tonal signal emission at frequency ω' in a waveguide with fixed parameters. Hence, the simple harmonic signal spectrum is $S(\omega) = \delta(\omega' - \omega)$. We recall once again (see Sect. 4.5 for more details) that the pulsed signal (8.88) contains all the dispersive distortions caused by waveguide propagation. For example, duration τ_1 becomes much longer than τ. Thus, the simple signal with the parameter $D = \tau\Delta\omega \cong 1$ becomes complex with $D = \tau_1\Delta\omega \gg 1$ (where the parameter D is called the signal complexity). Let us now assume that the PS emits a signal at the instant $t = 0$; then the time-reversed signal can be written as $u(0, z_j, T_1 - t)$, where T_1 is the signal propagation time satisfying the condition $T_1 > 2\tau$ and

$$\begin{aligned} u(0, z_j, T_1 - t) &= N_r A \int_{-\infty}^{\infty} \Psi(0, r_0, z_0, z_j, \omega) S(\omega) \exp\{-i\omega(T_1 - t)\} d\omega \\ &= \int_{-\infty}^{\infty} [\Psi^*(0, r_0, z_0, z_j, \omega) e^{i\omega T_1} S^*(\omega)] e^{-i\omega t} d\omega. \end{aligned} \tag{8.128}$$

In deriving relation (8.128), the well-known property of the Fourier transform of real functions of time (the emitted signal and pulse characteristics of the waveguide) was used, written as $S(-\omega) = S^*(\omega)$, and $\Psi(0, r_0, z_0, z_j, -\omega) = \Psi^*(0, r_0, z_0, z_j, \omega)$, respectively. The square bracketed term in (8.89) is the complex conjugate spectrum of the received signal. Thus, to reverse the signal in time, it is sufficient to calculate its complex conjugate spectrum and then to calculate the signal itself using the inverse Fourier transform.

[10] Hereafter in this section, the Green's function is also defined for negative frequencies, i.e., $\Psi(0,r_0,z_0,z_j,-\omega) = \Psi^*(0,r_0,z_0,z_j,\omega)$.

We then once again invoke (A.165) to calculate the sound signal at depth z and at distance r from the array:

$$p(r,z,t) = \hat{K}\sum_{j=1}^{J}\int_{-\infty}^{\infty}\Psi(0,r,z_j,z,\omega)\Psi^*(0,r_0,z_j,z_0,\omega)\mathrm{e}^{i\omega T_1}S^*(\omega)\mathrm{e}^{-i\omega t}d\omega. \quad (8.129)$$

Relation (8.129) is a generalization of (8.123) for the case of narrow-band acoustic signals.

A fundamental difference takes place between tonal and broadband sound field focusing, i.e., between sound field focusing using phase conjugation and a time-reversal mirror. Specifically, not only the spatial localization of the field occurs at the focusing point, as follows from (8.126), but also its temporal pulse compression almost to the pulse's initial duration τ, i.e., dispersive distortions arising during the signal propagation from the PS to the SAR are compensated! To prove this fact, let us assume that the spectrum of the emitted pulse is close to uniform, with spectral density S_0 in the frequency band $\Delta\omega$. Thus, the sound signal at the focal point can be written in the simplified form:

$$p(r,z,t) = \hat{K}S_0\sum_{j=1}^{J}\int_{\Delta\omega}\Psi(0,r_0,z_j,z_0,\omega)\Psi^*(0,r_0,z_j,z_0,\omega)\mathrm{e}^{-i\omega(t-T_1)}d\omega. \quad (8.130)$$

For this case, the function $p(r,z,t)$belongs to the class of so-called positive definite functions. Such functions cannot have complexity D larger than unity. At the same time, as can be seen from (8.121), this function of time features the same spectral width $\Delta\omega$ as the pulse emitted by the PS. This means that the duration of the signal $p(r,z,t)$ is close to τ.

Specific peculiarities of the broadband sound field focusing in shallow water can result in focusing by using single time-reversal element; what this means is that a vertical array need not be used.[11] However, the quality of focusing is much better when a vertical time-reversal array is used (see Lunkov et al. (2010) for details). For a tonal sound field, it is impossible to focus the sound field by using single time-reversal element. On the other hand, the quality of focusing is approximately the same for tonal and impulse sound fields (for phase conjugation and for the time-reversal mirror) if the same vertical array covering the waveguide is used and the frequency of the tonal sound is equal to the mean frequency of the chirped signals used for the time-reversal mirror (see Lunkov et al. 2010).

It is also interesting to consider (8.129) and (8.130) from the viewpoint of MFP. It is known [see, e.g., Tolstoi and Clay (1966)] that the largest possible SNR is

[11] Notice that in the first experiment with "time-reversal mirror" (Pazvulescu 1995) sound field focusing was carried out with single time-reversal element.

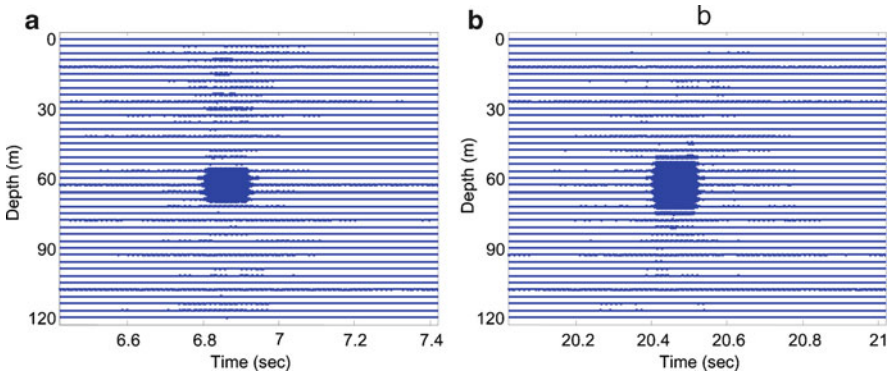


Fig. 8.22 Simulation of sound pulse focusing at the distance 10 km (**a**) and 30 km (**b**). SRA spans the waveguide. Time is measured from the point in time at which sound pulse is radiated by SRA

achieved at the output of an optimum correlation receiver for such processing. The signal at the receiver output is written as

$$B(t) = \int_{-\infty}^{\infty} S_1(\omega)S_2(\omega)\mathrm{e}^{-i\omega t}d\omega = \int_{-\infty}^{\infty} S_1(\omega)S_1^*(\omega)\mathrm{e}^{-i\omega t}d\omega \tag{8.131}$$

where $S_1(\omega)$ is the received signal spectrum and $S_2(\omega) = S_1^*(\omega)$ is the reference signal spectrum, which should be used in the matched processing. If the received signal is considered to be the PS signal which has traveled the distance r_0 in the waveguide (e.g., the signal received by the source/receiver array), then

$$S_1(\omega) = S(\omega)\Psi(r_0, z_0, 0, z_j). \tag{8.132}$$

Substituting (8.132) into (8.131), we obtain a relation very close to (8.129),

$$B(t) = \int_{-\infty}^{\infty} \Psi(0, r_0, z_j, z_0, \omega)S(\omega)\Psi^*(0, r_0, z_j, z_0, \omega)S^*(\omega)\mathrm{e}^{-i\omega t}d\omega. \tag{8.133}$$

However, to implement the optimum receiver using relation (8.133), the waveguide parameters must be known, whereas the signal corresponding to (8.129) is obtained as a consequence only of focusing. To optimally separate the focused pulse from the noise background, it is sufficient to multiply the spectrum of the signal $p(r_0, z, t)$ by the value of $S^*(r, z, \omega)$, which is known. Thus, as noted in Parvulescu (1995), the oceanic waveguide plays the role of a self-matched filter during focusing.

Focusing of a narrow-band rectangular pulse via numerical experiments[12] is shown in Fig. 8.22. Here the pulse duration was set to $\tau \cong 0.1$ s. The carrier

[12] Such numerical experiments were carried out for the first time in Kuperman et al. (1998).

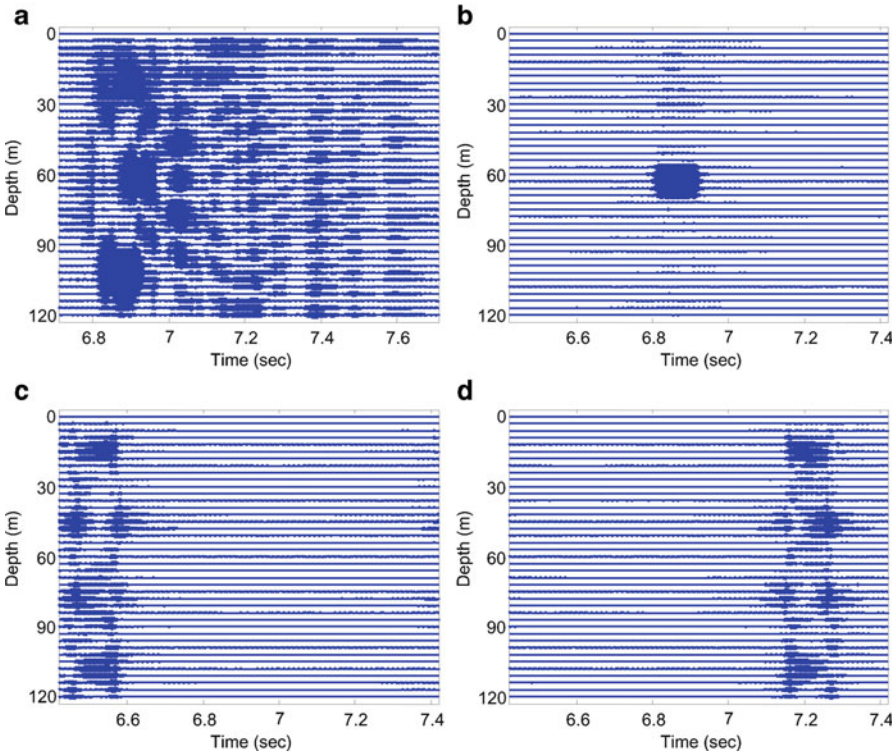


Fig. 8.23 Simulation of sound pulse focusing at the distance 10 km. (**a**) The pulse data received on the SRA for PS at depth 60 m. (**b**,**c**,**d**) The data received from the time reversed transmission of pulse shown in (**a**). (**b**) The data received at various depths in focal planes; the data received at various depth in the plane lying at distances of 1 km closer (**c**) and farther (**d**) than the focusing plane arranged at 10 km from the array. Time is measured from the point in time at which sound pulse is radiated by probe source (**a**) and SRA (**b**,**c**,**d**)

frequency and waveguide parameters were taken to be the same as when we were simulating tonal radiation focusing in the Pekeris waveguide, where the SRA spanned the whole waveguide (see above). Figure 8.22 shows the pulse received at various depths in the focal planes at distances of 10 and 30 km from the array. The focusing depth (the PS depth) is $z_0 = 60$ m. To illustrate the focusing quality, Fig. 8.23 shows the pulses received at the same depths, but in planes lying at distances of 1 km closer and farther than the focusing plane at 10 km from the array. The pulse received by the array is also shown there. We can see that the time reversal of the signal provides focusing of the sound pulses in the shallow waveguide, including significant distances from the array. However, as the distance increases, the quality of focusing does decrease. This is true both for the simple harmonic sound field (see Fig. 8.21) and the pulsed sound field (see Fig. 8.22).

It should be emphasized once again that high quality of narrow-band pulse focusing is retained even with a limited array aperture. The numerical results of

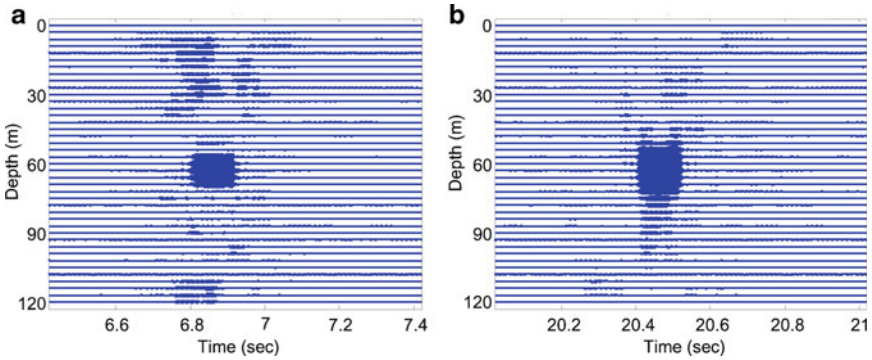


Fig. 8.24 Simulation of sound pulse focusing at the distance 10 km (**a**) and 30 km (**b**). The length of SRA is equal to 0.7 H. Time is measured from the point in time at which sound pulse is radiated by SRA

such focusing (in the same waveguide) are shown in Fig. 8.24. Here it was assumed that the array length is $L = 87$ M $\cong 0.7\ H$. The array is placed at the bottom.

The feasibility to focus narrow-band sound field in a shallow waveguide was also confirmed by the field experiments of Kuperman et al. (1998). Figure 8.25 shows the experimental results at a distance to the focal point (the PS point) of ≈6.3 km. The emission frequency was 445 Hz and the waveguide depth was 130 m. A sharp thermocline, where the sound speed changed by 4 m/s, was seen at a depth of 20 m. The pulse duration was $\tau = 50$ ms. Focusing was carried out using a 20-element SRA of aperture $L = 72$ m. Sound pulses were detected at various depths in the vertical plane near the focal point using an additional vertical receiving array VRA. The distance between this array and the PS point was 40–75 m. We can see in Fig. 8.25 that the sound pulse focusing is good despite the relatively small array aperture of $L = 0.55\ H$. This result is of significant practical importance, since it is clear that ideal focusing of the sound field in a randomly inhomogeneous waveguide using time reversal requires the use of a continuous 2D source/receiver array, clustered around the PS (Jackson and Dowling 1991). However, as follows from Fig. 8.25, even a vertical array with limited length provides good results. At the same time, we note that no intense internal waves causing horizontal refraction (see Sect. 3.7) were detected in that experiment. Such 3D refraction cannot be compensated for by time reversal of the acoustic waves using a vertical linear array.

So far, the frozen medium approximation has been used when analyzing sound field focusing and in numerical experiments, i.e., we assumed that the waveguide parameters remain unchanged in time. It is clear that this approximation is not valid in practice, even in the case when the time of sound signal propagation from the PS to the receiving array does not exceed 1 min. Indeed, waveguide parameters change over this time, first of all due to surface waves. In this case, all features of sound wave propagation in the waveguide, including scattering at random inhomogeneities, are not compensated for by time reversal. On the way from the

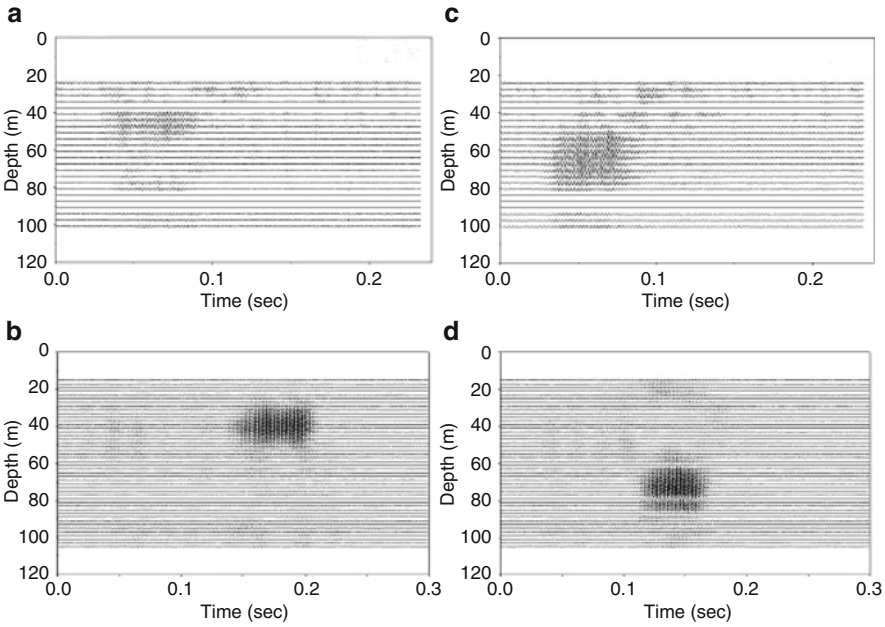


Fig. 8.25 Experimental results of sound pulse focusing at the distance ≈6.3 km. (**a**) The pulse data received on the SRA for PS at depth of 40 m. (**b**) The data received on the VRA from the time reversed transmission of pulses shown in (**a**). The VRA is 40 m inbound from the focus as determined by DGPS. (**c**) The pulse data received on the SRA for PS at depth of 75 m. (**d**) The data received on the VRA from the time reversed transmission of pulses shown in (**c**). The VRA is 40 m outbound from the focus as determined by DGPS

array to the focal point, the sound passes through a different medium than on its propagation from the PS to the array. The focusing quality is thus deteriorated, and the focal spot parameters begin to randomly vary in time from one probe pulse to another. This effect is illustrated by Fig. 8.26 from Kuperman et al. (1998). Here the averaged focal spot is shown, as well as the average level of sound field and its standard deviation at various depths in the focal plane. These results were obtained under field conditions using a measurement time of 1 h.

Fortunately, an analysis of Fig. 8.26 shows that temporal fluctuations of the focal spot are in general insignificant in actual practice, and at least at a distance of ~6 km from the array, adequate focusing quality is observed. Furthermore at later time the same authors Hodgkiss et al. (1999) demonstrated in the other full-scale experiment that the sound field focusing with satisfactory quality can be obtained after ten days since initial focusing. The distance between SRA and focal spot was equal to 15.2 km. The emission frequency was 445 Hz.

It is significant that the full-scale experiments mentioned above were carried out for "closed" region of ocean shelf (Mediterranean Sea) with small temporal variability of near bottom sound channel. At the same time "open" regions of ocean shelf are characterized by the great temporal variability caused by internal

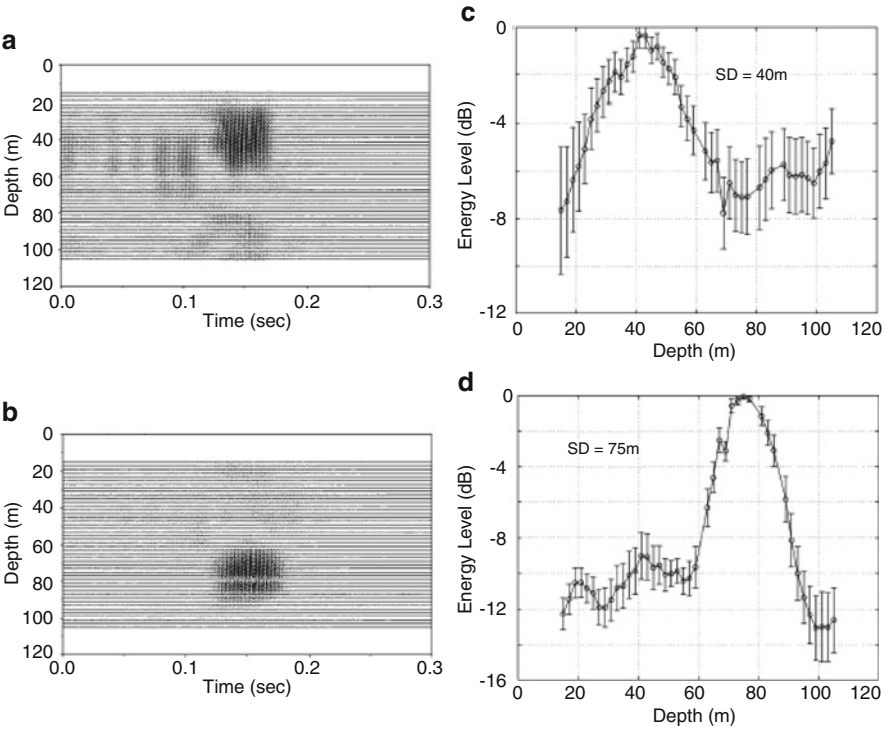


Fig. 8.26 Experimental results on stability of the focal region. (**a**) Pulse arrival structure at VRA for probe source at 40 m depth averaged over 1 h. (**b**) Pulse arrival structure at VRA for probe source at 75 m depth averaged over 2 h. (**c**) Mean and standard deviation of energy in a 0.3-s window for 40 m probe source. (**d**) Mean and standard deviation of energy in a 0.3-s window for 75 m probe source

waves. According to the numerical estimations (Dungan and Dowling 2000) the background internal waves for "open" shelf can decay focal spot in five minutes. The estimations were performed for the site of the SWARM experiment (Apel et al. 1997) (New Jersey shelf). The distance between SRA and focal spot was equal to 10 km. The emission frequency was 500 Hz. The power spectrum of internal waves recorded in the SWARM experiment was used for the estimations.

In conclusions of this section notice that the position of focal spot (the distance r between SRA and focal spot) can be varied by a relative simple method. For this purpose we should only vary the carrier frequency f of sound signal. According to (4.64) the relation between the frequency variation Δf and the distance variation Δr is equal to

$$\Delta r \cong \frac{\Delta f r}{f \beta_0}$$

where $\Delta f \ll f$ and $\Delta r \ll r$.

8.6 Acoustic Uncertainty

A very interesting area of research over the past few years has been the study of "Uncertainty" (unaccounted for ocean variability), i.e., our inability to know everything about the ocean environment and the effect that lack of knowledge has on sonar systems and sonar signal processing. This is not an admission of defeat on the part of the environmental sensing community, but rather a sane, practical, and pragmatic realization that we will always have limited resources for ocean sensing, which in turn implies finite resolution and nonzero variance of our ocean estimates. Given that practical constraint, what is the best we can do?

In addressing that last question, let us go back to the basic ocean/seabed/biology system that we are trying to measure, and see wherein the problem lies. And, to try to use a simple, common feature as an example, let's consider surface waves.

In looking at the sea surface, we know that we can take occasional photographic snapshots of the surface, and with some clever processing of the glints and shadows, can get a "realization" of the 2D (x–y plane) surface wave field at various instants. This is useful for some purposes (e.g., testing the ergodic hypothesis), but is not a product that we have the time, patience, or resources to produce routinely. Looking at every wave, all the time is not a pursuit for mere mortals.

That has not bothered us, however, because we are quite adept at this point in time in treating the problem statistically, i.e., in terms of spectra. This allows us to distill (average) our 2D realizations into a rather compact entity such as the 2D frequency-directional spectrum of the waves, a common product of wave analysis. So, if we just drop back to the spectral level, can we know "everything there is to know" about the wave field, at least in this domain? Sadly, no. At this point, we hit the next level of constraint, specifically (1) our measurements have finite aperture in space and time and (2) the ocean waves are nonstationary in space and time. The combination of these two conditions means that we will be constrained to obtaining a spectral estimate in space and time, and will never have the "exact spectrum."

But again, there is good news. We are rather good at this point in time in the science of spectral estimation, and are well aware of its practical limitations due to the finite windows and nonstationary signal effects. So we can create an estimate with "error bars."

The surface wave story can go even further. In many instances, we can use our understanding of the physics of a process, such as waves, to augment and extend the information that the experimental data provides. If, for instance, there is a "universal surface wave spectrum" (e.g., the Pierson–Moscowitz spectrum for fully developed seas), we can take a rough measurement estimate of the spectrum, match it to the universal form, and then happily extend our measurement into areas we have not directly measured in this instance. Or going back one more step, if we know how the wind forcing produces a wave spectrum, and measure the wind forcing somehow, then we can get the waves from this "indirect" measurement. So, it is obvious that understanding the physics of the waves and their forcing (i.e., process studies) buys us some additional information and power in our measurement systems. This is the much ballyhooed interface of models and measurements. So, are we home free?

Again, no. For this approach to deliver "perfect results," we would need perfect knowledge of all the physics of surface waves and/or perfect measurements of the wind forcing function. Neither one is possible – even with models, there will be uncertainty in the model physics and the model forcing functions. (Think of how well we understand and measure breaking waves, bubble clouds, and Langmuir cells, for instance.) Again, this is not any cause for despair. We can only improve on the estimates produced by our measurements by (correctly!) incorporating models of the physics and dynamics of the systems we are measuring. (Please note the "correctly." It is quite possible to degrade your estimate by incorporating incorrect models of the system).

So, even for one of the best known processes in oceanography, surface waves, we see that we are going to have to settle for less than perfect knowledge. When we start getting to somewhat less well understood (or easily measurable) processes such as ocean internal waves, fronts, and eddies, or bottom geoacoustic profiles, or fish schools, we will have even more uncertainty in our environmental estimates. So what do we do?

There are two obvious first steps to approaching uncertainty problems (1) optimize your sampling of the process given the measurement resources you expect to have and (2) try hard to understand the process physics/oceanography/seabed/biology in the area you want to estimate, so that you can get the most additional information from models. These are strictly common sense recommendations and ones that have been pursued throughout the years.

Before getting to the topic of how to approach this uncertainty for ocean acoustics purposes, let us just note one or two more things. First, there is the matter of the deterministic or statistical view of a system, and how that interacts with uncertainty. In a practical sense, we can deterministically measure the features that our system can resolve (in space and time), and then can look at either realizations or spectra as descriptors. Where we draw the line between deterministic and statistical descriptions often depends on our "comfort zone" in handling large amounts of data versus the simpler spectra. For processes below our resolution, we have to look to models or a priori knowledge to supply some idea of the variability (variance) that will be caused. For instance, if we have a nonlinear internal wave field present on the shelf, but have only taken temperature measurements that resolve to 2 km (adequate for mesoscale features and the internal tide, but not individual internal waves) and the M2 tidal frequency (again, adequate for the internal tide/bore part of a soliton wave train, but not the individual waves), what do we do? We have aliased the higher frequency oceanography in the basic measurements, so we obviously must appeal to either previous knowledge (climatology) or models (the wave physics) or both to supply the missing information. At this point, we hit the interesting point: how much of the variability due to poorly measured processes can we estimate with our a priori knowledge, and how much is just plain unknown? This again bumps into the issue of how good our models and other a priori's are, which in this case puts an absolute bound on our uncertainty. Our measurements, models, and databases can get us a reasonable estimate up to this point, but past there "we don't know what we don't know." Thus, there is an "estimable uncertainty" that can be filled in by models, physics, previous data, etc., and then there is the stone wall of what we just, flat out,

do not know, the "unestimable uncertainty." It is our hope to minimize the "unestimable uncertainty," which at best we can put sanity bounds on (as an extreme example, the soundspeed in water is unlikely to exceed 2,000 m/s). We also note that this discussion can be put in terms a measured ocean signal and an unmeasured noise, some of the latter of which we can estimate via our models and a priori knowledge. The size and statistics of the "unestimated" noise are, as before, *terra incognita*.

Having talked about our ocean/seabed/biology estimation problem, i.e., the environmental input that determines the acoustic field from a given source, let's now look at how we can address it. A first approach, and one that jives nicely with the Bayesian inverse approach described in an earlier chapter as well as our discussion earlier, is to describe the environmental inputs to acoustic propagation models as probability density functions (PDFs), which in turn produces a statistical description of the acoustic field (e.g., the mean, variance, and other higher order moments). Knowing what the mean of the acoustic field is, and its expected fluctuations, gives a powerful input to sonar signal processors. Of course, this approach runs right into the difficulties we discussed in estimating the surface wave spectrum – we can only estimate the PDF of a given environmental variable, but never know it exactly. This is especially true of the higher order moments of the distribution (the tails), which are harder to estimate due to their sensitivity to error and lack of data, but even the lower moments can be challenging.

Past looking at the statistics of the environmental inputs to acoustics models, we can also go directly to the statistics of the primary acoustics quantities themselves. In this instance, we turn, following the work of Abbot and Dyer (2002) and Abbot et al. (2006), to the sonar equation. The terms like transmission loss (TL) and noise level (NL) in the sonar equation are directly affected by fluctuating environmental propagation conditions, and thus the SNR will show variability due to them (SNR = SL–NL–TL in a simple approach, where SL is source level). Abbot and Dyer have proposed putting the (estimated) PDFs of these acoustics terms into the sonar equation for use in probability of detection calculations, thus creating a "predictive probability of detection" that firmly incorporates environmental variability into the sonar equations. Looking at a graphic representation of this concept in Fig. 8.27, we see that the PPD approach gives a smooth probability of detection curve versus range, versus the (unrealistic) "detect/no-detect" verdict at a given range that the older approach gives.

Given that we might want to incorporate environmental variability into the sonar equation in this way, let's look at a real-world example of "coping with uncertainty" using some data taken during the SW06 experiment off New Jersey, courtesy of Abbot, Emerson, and Gedney. In Fig. 8.28, we see two examples of transmission loss versus azimuthal angle for two near-shelfbreak circular transmission loss tracks. These curves show some interesting detail, namely (1) they are fairly similar to first order, indicating an approximate translational invariance in conditions between the two tracks (which were offset from each other ~10 km along shelf), (2) there is a bit more loss in the lower right hand third of the two plots, where the acoustic track crosses the shelfbreak front, and so propagation conditions change, and (3) there is a distinct scalloping pattern to the TL versus azimuthal angle, which has an ~15° null

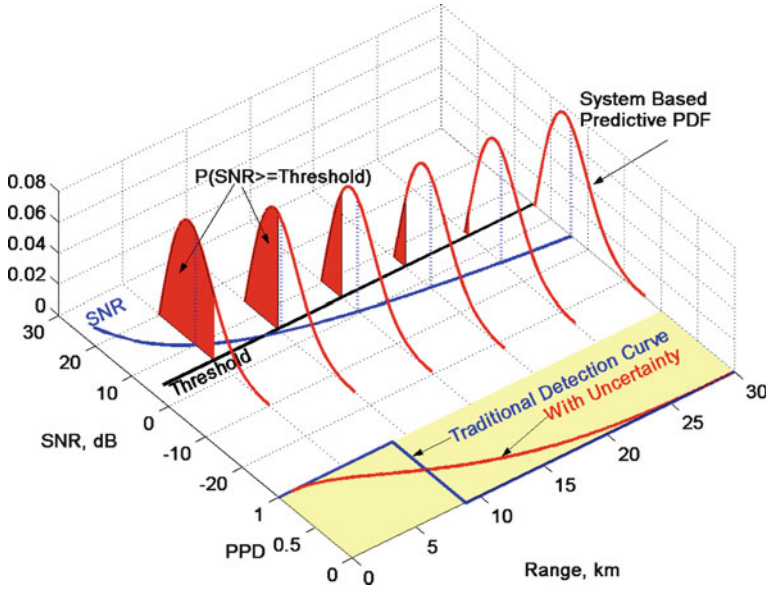


Fig. 8.27 Schematic of how the Predictive Probability of Detection (PPD) approach (Abbot and Dyer 2002; Abbot et al. 2006) compares with a traditional single SNR curve and a threshold (here at 5 dB)

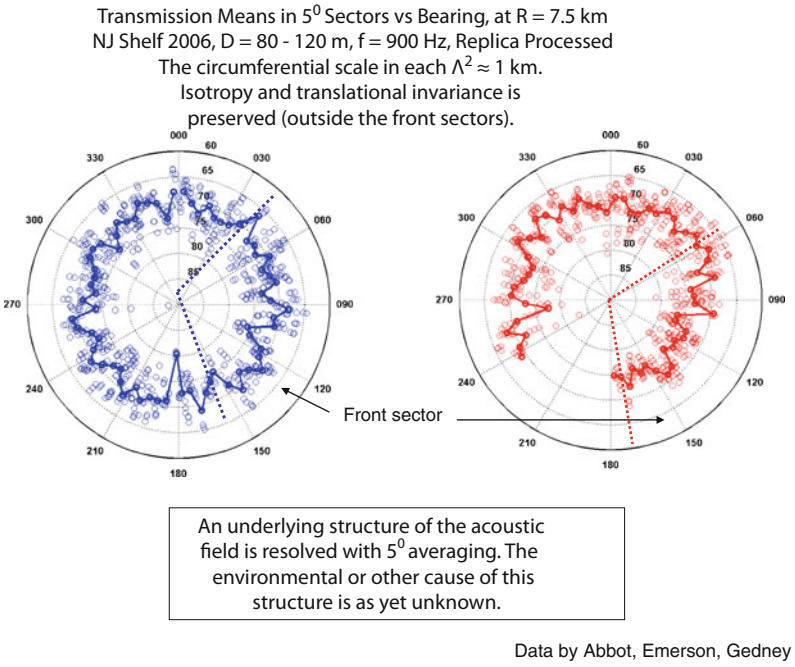


Fig. 8.28 Transmission loss versus azimuthal angle for two near-shelfbreak circular acoustic tracks measured during the SW06 experiment

spacing in both cases (though this is a bit irregular). These are interesting physical characteristics of the mean curves, but in terms of the PPD formalism, do we really need them? To get a PDF, we just bin the TL measurements (either in a small angular sector or for all angles, if we ignore azimuth) and insert the result in the PPD sonar equation. Once we have the PDF, we are in good shape, aren't we? The answer in this case is "yes and no." Yes, in that our histogram estimate of the PDF is a good working start for the time and place that we sampled. But it will not work beyond the decorrelation time and distance of the medium, so that if we came back to the same place 2 days later, or looked 20 km away at the same time, we might easily see a different PDF estimate. The medium variability determines the PDF approximation we measure, so just going to the PDF formalism has not obviated our uncertainty problem, but just recast it in a form more consistent with the real sonar detection and estimation problem.

So, we are now at the point of trying to understand the environmental physics that dictates the PDF of the TL or NL, which is basically the same problem as before, only looking at the acoustic quantities versus the environmental ones. In the case of the three properties of the curves mentioned earlier, we have some rather good hypotheses that we believe can explain the features of Fig. 8.27 curves, and so will give us some modeling and predictive capability. Regarding the first property, translational invariance of the curves along-shelf, we see that we are propagating through coastal bathymetry, bottom geology, a coastal frontal system, and a system of nonlinear internal waves that is, to first order, translationally invariant alongshelf. This would produce translational invariance for the TL results. The second effect, the greater loss seen in the curves as the sources go across the shelfbreak front, is a standard propagation effect seen when one goes from cooler to warmer water, and thus goes to higher angles of bottom interaction. The slight deviation between the two curves in the frontal region is likely due to the front being "less invariant" alongshelf, i.e., subject to meanders, eddy influences, etc. Turning to the third effect, the "scalloping" of the curves in azimuth, we strongly suspect that this is due to the Bragg scattering of the sound from the internal wave field as first described by Rubenstein, In this case, the mode cycle distance equals the spacing between waves at a given angle, thus giving more mode coupling and thus increased bottom interaction. The null spacing and size of the effect are consistent with this mechanism, and we are in the process of matching a few sets of data to the theory to test if our hypothesis is correct. This (eventual) understanding of the physics will allow us to make more focused measurements of the environment (e.g., concentrate on key variable like front position, timing of the internal wave packets, etc.), and also to approximate the PDFs better in areas where we have little data, or even no data.

There one more question which we will pose here, but not answer. That is, in estimating the PDF for the PPD method, what errors do we make in the estimate, and how serious are these to the detection problem? Getting the mean TL right is critical to the old "detection range" methods, but this result gets smoothed out some in the new method. Also, the tails of the detection distributions also get a bit smoothed out due to the integration (over the PDF) inherent in this method, so what does this do to the criticality of knowing the PDF exactly. These are questions that are worth pursuing, but we cannot say that we know the answers quite yet.

Chapter 9
Noise Field in Shallow Water

9.1 Introduction. Model of a Noise Source: General Equations

One of the most important problems in ocean acoustics is the study (measurement, forecasting) of noise, that is, of the sound fields formed by various wideband, distributed sources. These sources include wind-generated waves (surface disturbance noises), various human constructions which generate industrial and mechanical noises (ships, drilling platforms, etc.), marine organisms (biological noises), earthquakes, and volcanic eruptions (seismic noises), meteorological noises (rain, thunder), noises from under the ice, etc. The effects of these sources, and the characteristics of the noise field they generate, depend on many factors including weather, geographical position, time of day, and season, and are manifested in various frequency bands. A good example of course is that mechanical noises have a marked maximum at the "electrical" frequencies (50 and 60 Hz) of the alternating current mains.

The noise field has a dual role. On the one hand, it represents an undesirable interference for the operation of various ocean acoustic devices, while on the other hand it provides researchers with a free tool with which to study propagation in the ocean medium itself, using the signal characteristics of the noise.

The more or less systematic study of underwater noise began more than 50 years ago, after the end of the Second World War, and since that time a large number of both experimental and theoretical results have been obtained. There is a vast literature devoted to the study of various aspects of this problem, including advanced monographs, in particular (Urick 1975; Ross 1976; Furduev 1974; Kuryanov and Klyachin 1974). Historically, the first works were devoted to the study of noise in deep water, and the interest has only recently been carried over to shallow water. We note that in what follows, we shall be interested in the following observable characteristics of the noise field, usually at a fixed frequency (spectral characteristics): space–time variations of the level, depth dependence of the intensity, space–time coherence, and vertical and horizontal directivity. We note that

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_9,
© Springer Science+Business Media, LLC 2012

these and other characteristics can be calculated theoretically if one has adequate information about the propagation medium and the noise sources. Specifically, one needs to know the level (intensity) and angular distribution of emission from the sources, generally speaking as a function of the spatial coordinates and of time, together with the parameters of the acoustic waveguide.

Although the theory of sound propagation in shallow water presented earlier enables us to solve this problem in principle, in reality many of the data needed to model the noise are either unknown or at least insufficiently well known. As an example, let us first consider the space–time distribution and intensities of the noise sources. To obtain these quantities, we must know the generation mechanisms in the ocean, such as surface waves, the formation and collapse of bubbles in diverse conditions, biological sources, industrial and shipping noise, etc. There is considerable literature describing the different mechanisms for noise generation (e.g., Ross 1976; Mendus and Postnov 1993; Furduev 1966; Wilson 1979; Oguz 1994); however, not all of the noise source mechanisms have been experimentally quantified. This is because it is hard experimentally to place a receiver in close proximity to some natural noise sources (e.g., ice cracks, breaking waves, etc.) which are highly variable and intermittent in space and time.

However, some properties of ocean noise can be modeled with high accuracy. These are properties which are determined by general properties of the propagation medium and by certain characteristics of the sources, which are of a general nature for all types of sources. For example, as mentioned in Yang and Kwang (1997), the noise in the azimuthal (horizontal) direction will have a maximum in the direction of the shipping lanes when one is studying shipping noise, or along the line of a storm if the noise is generated mainly by wind waves.

Furthermore, at low frequencies (below 1 kHz), noise sources can be considered to be point sources, which because of their proximity to the surface emit as effective dipoles. In this case, the vertical direction of the noise is determined by the properties of the waveguide (the propagation medium) and the absolute intensity of the noise source (source level) is not required (Yang and Kwang 1997; Hamson 1985). It has also been noted that in a lossy waveguide, the vertical scale of coherence is determined mainly by the angle of total internal reflection of a ray hitting the bottom (Carbone et al. 1998), which makes it possible in principle to determine the characteristics of the bottom from the observed characteristics of the noise.

In this chapter, the main noise source we shall consider is wind, which is responsible for the formation of a significant fraction of the noise field at comparatively low frequencies (up to 1–2 kHz) in open areas of the sea. We shall not go into detailed hypotheses about noise generation mechanisms here, but rather will consider a simplified model which can replicate the space–time distribution of the noise field in a shallow water waveguide. Accordingly, we shall consider the sound source to be a distribution of individual emitters on the surface, characterized by their intensity, direction and other (generally statistical) parameters. This approach, which was apparently first proposed by Cron and Sherman (1962) for a half-space, was subsequently generalized to the case of shallow water waveguides. Using this approach,

good results for the observable characteristics of the noise field have been obtained: intensity vs. depth distribution (Kuperman and Ferla 1985; Derevyagina et al. 1994); vertical and horizontal direction (Yang and Kwang 1997; Hamson 1985; Carbone et al. 1998); spatial correlation functions (Buckingham 1980; Kuperman and Ingenito 1980; Abdullaev et al. 1986); and the spectral distribution of intensity at a fixed depth (Kuperman and Ferla 1985). The effects of various ocean and sediment environmental factors on the noise field characteristics have also been considered: an absorbing bed (Derevyagina et al. 1988a; Carbone et al. 1998); the sound speed profile (Lyubchenko and Petnikov 1991); random homogeneities of various kinds (Derevyagina and Katsnelson 1995), including, in particular, internal waves (Yang and Kwang 1997).

As before, our approach is based on a modal decomposition of the sound field, which is justified for comparatively low frequencies.

Up to now, we have assumed in this book that the sources of interest are quasi-chromatic point sources. However, in working with the noise field, we must now deal with a wideband signal distributed over some spatial region. Let us briefly consider what an appropriate description of the sound field for this case could be.

As a model of the sources of surface noise, we shall consider a set of randomly distributed monopoles in the horizontal plane, which are located at a small depth z_s below the sea surface. For the characteristics of the surface sources of noise, we introduce the complex (and generally speaking, random) function $Q(\vec{r}, t)$ of the surface density of the emitted signal (a complex signal emitted from a unit surface area). The surface spectral density of the emitted signal is

$$Q_\omega(\vec{r}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathrm{Re} Q(\vec{r}, t) e^{i\omega t} dt, \tag{9.1}$$

$$Q(\vec{r}, t) = 2 \int_{0}^{\infty} Q_\omega(\vec{r}) e^{-i\omega t} d\omega, \tag{9.2}$$

and is also a random function of the space–time variables. We shall assume that $Q(\vec{r}, t)$ is a stationary process in time and is also homogeneous in the XY plane, and is characterized by the joint spectral correlation function

$$B_q(\vec{r}, \omega; \vec{r}', \omega') = \left\langle Q_\omega(\vec{r}) Q^*_{\omega'}(\vec{r}') \right\rangle = B_q(\vec{r} - \vec{r}')\delta(\omega - \omega'), \tag{9.3}$$

where the angular brackets denote averaging over realizations. The quantity $Q_\omega(\vec{r})$ can be interpreted as the number of sources per unit surface area with equal amplitude emitting signals in a unit frequency interval. The spatial correlation function $B_q(\vec{r} - \vec{r}')$, which is generally frequency-dependent, and characterizes the spectral intensity of the emission, is sometimes written in the form $B_q(\vec{r} - \vec{r}') = q_\omega^2 N(\vec{r} - \vec{r}')$. In this expression, the source spectrum level q_ω (Kuperman and Ingenito 1980),

measured in dB/μPa/$\sqrt{\text{Hz}}$ and the spatial density of the distribution of the sources $N(\vec{r}-\vec{r}')$, normalized by the number of sources N_0: $\int N(\vec{r})d\vec{r} = N_0$, are distinguished. If we additionally introduce the spectral component of the monopole volume pulsation speed, emitting the noise field V_ω, then the source spectrum level is related to this quantity by the equation: $q_\omega = \rho\omega V_\omega$. Accordingly, the emitted power from a unit surface in a unit frequency interval (spectral power density) is $W_\omega = q_\omega^2/8\pi\rho c$.

The complex amplitude of the sound field $P(\vec{R},t)$, formed by such randomly distributed sources, satisfies the equation:

$$\left(\Delta - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)P(\vec{R},t) = -Q(\vec{r},t)\delta(z-z_s). \tag{9.4}$$

The sound field at a fixed frequency (the spectral amplitude of the sound field $S(\vec{R},z_s;\omega)$), taken at the point with cylindrical coordinates (r,z), is given by the equation:

$$\left[\Delta + \frac{\omega^2}{c^2(z)}\right]S(\vec{r},z,z_s;\omega) = -Q_\omega(\vec{r})\delta(z-z_s) \tag{9.5}$$

with the appropriate boundary conditions on the bottom and the surface. Using the Green's function for the waveguide, the solution of (9.5) can be written in the form

$$S(\vec{r},z,z_s;\omega) = \int \Psi(\vec{R},\vec{R}_s;\omega)Q_\omega(\vec{r}_s)d\vec{r}_s, \tag{9.6}$$

where $\vec{R}_s = (\vec{r}_s, z_s)$. In particular, in a homogeneous space

$$S(\vec{r},z,z_s;\omega) = \int \frac{\exp\left(ik\left|\vec{R}-\vec{R}_s\right|\right)}{4\pi\left|\vec{R}-\vec{R}_s\right|}Q_\omega(\vec{r}_s)d\vec{r}_s. \tag{9.7}$$

Historically, the first model for studying the sound field due to sources distributed on the sea surface was the model of sources uniformly distributed on the surface of a homogeneous half-space, proposed by Cron and Sherman (1962). To examine this model, we imagine as an individual source, a point monopole of unit amplitude, located at a small depth z_s below the sea surface, emitting a monochromatic signal of frequency ω, where the radius vector of this point in the horizontal plane is $\vec{r}_s$ (see Fig. 9.1).

The complex amplitude of the field at a given frequency, given by the equation $P_s(\vec{R},t) = P_s(\vec{R})\exp(-i\omega t)$, at an observation point with coordinates $\vec{R} = (\vec{r},z)$ at a large distance from the plane of the surface in comparison with the wavelength, is equal to

$$P_s(\vec{R}) = \frac{e^{ik\left|\vec{R}-\vec{R}_s^+\right|}}{4\pi\left|\vec{R}-\vec{R}_s^+\right|} + V(\vartheta)\frac{e^{ik\left|\vec{R}-\vec{R}_s^-\right|}}{4\pi\left|\vec{R}-\vec{R}_s^-\right|}, \tag{9.8}$$

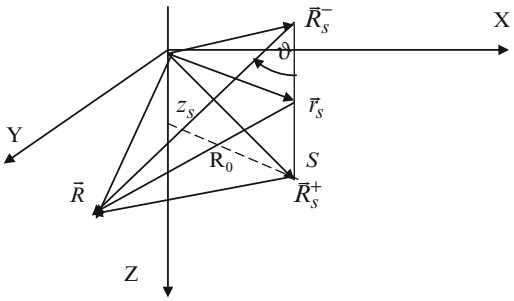


Fig. 9.1 Formation of the field from an individual source located at the point S near the sea surface, with coordinates $\vec{r}_s, z_s$. The point of observation is given by the radius vector $\vec{R}$. The radius vector from the source to the point of observation is $\vec{R}_s^+$, and the radius vector to the point of observation from the point corresponding to the image source is $\vec{R}_s^-$

where $\vec{R}_s^{\pm} = (\vec{r}_s, \pm z_s)$, $V(\vartheta)$ is the coefficient of reflection, and ϑ is the angle from the vertical, defined by the condition $z + z_s = \left|\vec{R} - \vec{R}_s^-\right| \cos\vartheta$. Assuming the depth of the sources is small, (one quarter of a wavelength is a common choice) we shall suppose that all the angles from the vertical of the vectors drawn from the source and its image source are equal, and also that $\left|\vec{R} - \vec{R}_s^{\pm}\right| \approx R_0(1 \pm z_s \cos\vartheta)$, $R_0 = \sqrt{(\vec{r} - \vec{r}_s)^2 + z^2}$.

Using these approximations, we obtain for the field of an individual effective source (formed by the source and its image)

$$P_s(\vec{R}) = \frac{e^{ikR_0}}{4\pi R_0}\left[e^{-ikz_s\cos\vartheta} + V(\vartheta)e^{ikz_s\cos\vartheta}\right] = g(\vartheta)\frac{e^{ikR_0}}{4\pi R_0}. \tag{9.9}$$

Equation (9.9) is a special case of the known properties of a field formed by sets of monopoles occupying a bounded area of space. It allows us to consider the distribution of the sources as the set of all effective sources, located on the surface, each of which has a certain directivity pattern. $g(\vartheta)$ is a function determined by the boundary conditions and by the conditions of emission, which combine to determine the directivity of an effective point source. In particular, if we have a pressure-release sea surface ($V(\vartheta) = 1$), then an individual effective source is a dipole and its directivity is determined by the function $g(\vartheta) = ikz_s\cos\vartheta$. We note that the expressions (9.8) and (9.9) do not include inhomogeneous waves and are approximately valid only for sufficiently large distances from the surface.

The expression (9.9) was used in Cron and Sherman (1962) to determine the correlation function. This was done as follows. Let us suppose that a receiver is located at the point with coordinates $r = 0$, z, that is, on the z-axis. Uncorrelated sources, each of whose directivity patterns is determined by the function $g(\vartheta)$, lie in the XY-plane. An individual source is located at the point S, and the geometry of the problem is shown in Fig. 9.2. The field of the individual source at the point of

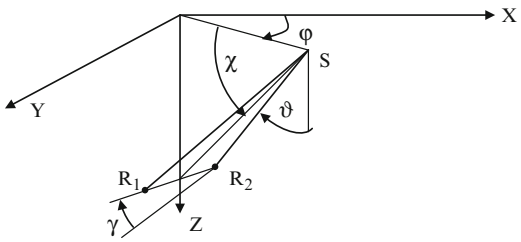


Fig. 9.2 Diagram of the coordinates used in the calculation of the noise correlation function. The angles χ from the horizontal (or ϑ from vertical) drawn from the point of observation are assumed to be identical for all radius vectors

observation is given by (9.9). The intensity of the total field from all sources, distributed in the XY-plane, is:

$$I(z) = 2\pi \int_0^\infty \frac{1}{2\rho c} |P_s(\vec{r})|^2 r dr = \frac{1}{16\pi\rho c} \int_0^{\pi/2} |g(\vartheta)|^2 \tan\vartheta d\vartheta \tag{9.10}$$

which is Cron and Sherman's result for the intensity.

Let us now consider the spatial correlation function of the noise, for which we introduce two different receivers at the points R_1 and R_2, located, for convenience, in the YZ-plane (see Fig. 9.2). The distance between the receivers is denoted by d and the angle of this section with the horizontal axis (Y-axis) is denoted by γ. The spatial correlation function, normalized to unity the maximum, and measured by summing the sound pressures at the distinct receivers, is determined using the equation:

$$\begin{aligned} B(\vec{R}_1, \vec{R}_2) &= \frac{\int_0^\infty r dr \int_0^{2\pi} d\varphi \left| P(\vec{R}_1) + P(\vec{R}_2) \right|^2}{\int_0^\infty r dr \int_0^{2\pi} d\varphi \left[\left| P(\vec{R}_1) \right|^2 + \left| P(\vec{R}_2) \right|^2 \right]} - 1 \\ &= \frac{\mathrm{Re} \int_0^\infty r dr \int_0^{2\pi} d\varphi P(\vec{R}_1) P^*(\vec{R}_2)}{I(z)}. \end{aligned} \tag{9.11}$$

Here, $\vec{R}_1$, $\vec{R}_2$ are the radius vectors from the individual emitters, to the mean of the left- or right-hand points of observation, respectively. In introducing this formula, we have neglected the difference in the coordinates of the observation points everywhere, except in the phases of the emitted field. Substituting the expression for the emitted field and integrating, we obtain Cron and Sherman's expression

$$B(\vec{R}_1,\vec{R}_2)=B(d,\gamma)=\frac{\int\limits_0^{\pi/2}|g(\vartheta)|^2\tan\vartheta\cos(kd\sin\gamma\cos\vartheta)J_0(kd\cos\gamma\sin\vartheta)d\vartheta}{\int\limits_0^{\pi/2}|g(\vartheta)|^2\tan\vartheta d\vartheta}. \tag{9.12}$$

It is very interesting to note that this expression does not contain z, i.e., it does not depend on the depth of the receiver. From it, one can obtain correlation functions for various cases of the orientation and directivity of the noise sources. In general, it is customary to assume that the directivity of the source corresponds to that of multipoles distributed on the surface, i.e., to the function $g(\vartheta)\sim\cos^m\vartheta$. As examples, for the horizontal ($\gamma=0$) and the vertical $\gamma=\pi/2$ correlation functions, the above expression simplifies to

$$B(d,0)=\frac{\int\limits_0^{\pi/2}\cos^{2m}\vartheta\tan\vartheta J_0(kd\sin\vartheta)d\vartheta}{\int\limits_0^{\pi/2}\cos^{2m}\vartheta\tan\vartheta d\vartheta}=\frac{2^m m!J_m(kd)}{(kd)^m}, \tag{9.13}$$

$$B(d,\pi/2)=\frac{\int\limits_0^{\pi/2}\cos^{2m}\vartheta\tan\vartheta\cos(kd\cos\vartheta)d\vartheta}{\int\limits_0^{\pi/2}\cos^{2m}\vartheta\tan\vartheta d\vartheta}=2m\int\limits_0^1 x^{2m-1}\cos(kdx)dx. \tag{9.14}$$

These expressions are easily computed, and numerical results are discussed in Cron and Sherman.

Equation (1.139), which was used to obtain the results, is approximate. It would be more accurate to work with a decomposition of the field into plane waves, as then we would be dealing with precise solutions of the wave equation, as in Kuryanov (1963). It is well known that the sound field from a point source in the plane $z_s=0$ can be decomposed into plane waves (Brekhovskikh 1980):

$$\frac{e^{ikR_0}}{4\pi R_0}=\frac{i}{8\pi^2}\int\frac{1}{\sqrt{k^2-\xi^2}}e^{i\vec{\xi}(\vec{r}-\vec{r}_s)+ik_z z}d\vec{\xi},$$
$$R_0=\sqrt{(\vec{r}-\vec{r}_s)^2+z^2}. \tag{9.15}$$

Accordingly, the complex amplitude of the field of an effective directional point source in the plane can be written as:

$$P_s(\vec{r},z)=\frac{i}{8\pi^2}\int\frac{g(\xi)}{\sqrt{k^2-\vec{\xi}^2}}e^{i\vec{\xi}(\vec{r}-\vec{r}_s)+ik_z z}d\vec{\xi} \tag{9.16}$$

At large distances, this field becomes (9.9). Suppose now that the sources in (9.16) are distributed in the plane $z_s=0$ with some density $Q_\omega(\vec{r}_s)$. Then the total field at frequency ω, and at some given distance from the plane, takes the form

$$\begin{aligned}S(\vec{r},z,0;\omega)&=\int d\vec{r}_s P_s(\vec{r},z)Q_\omega(\vec{r}_s)=\frac{i}{8\pi^2}\int\int d\vec{r}_s d\vec{\xi}\frac{Q_\omega(\vec{r}_s)}{\sqrt{k^2-\xi^2}}g(\xi)e^{i\vec{\xi}(\vec{r}-\vec{r}_s)+ik_z z}\\&=\frac{i}{2}\int d\vec{\xi}\frac{Q_\omega(\vec{\xi})}{\sqrt{k^2-\xi^2}}g(\xi)e^{i\vec{\xi}\vec{r}+ik_z z}=\int d\vec{\xi}\bar{F}_\omega(\vec{\xi})e^{i\vec{\xi}\vec{r}+ik_z z},\end{aligned} \tag{9.17}$$

where $Q_\omega(\vec{\xi})=\frac{1}{(2\pi)^2}\int d\vec{r}_s Q_\omega(\vec{r}_s)e^{-i\vec{\xi}\vec{r}_s}$ is the spatial spectrum of the source distribution function. Kuryanov's (1963) work uses a distribution of the field in the plane of the sources and its spatial spectrum. For our case, this is

$$\bar{F}_\omega(\vec{\xi})=\frac{i}{2}\frac{Q_\omega(\vec{\xi})}{\sqrt{k^2-\xi^2}}g(\xi). \tag{9.18}$$

The correlations of these functions are related; specifically, assuming a δ-correlation of the spectral distribution function, we have the relation

$$\begin{aligned}\left\langle\bar{F}_\omega(\vec{\xi})\bar{F}^*_{\omega'}(\vec{\xi}')\right\rangle&=\frac{1}{4}\frac{\left\langle Q_\omega(\vec{\xi})Q^*_{\omega'}(\vec{\xi}')\right\rangle}{\sqrt{(k^2-\xi^2)(k^2-\xi'^2)}}g(\xi)g(\xi')\\&=\frac{g^2(\xi)\bar{B}_q(\vec{\xi})}{4(k^2-\xi^2)}\delta(\omega-\omega')\delta(\vec{\xi}-\vec{\xi}'),\end{aligned} \tag{9.19}$$

where $\bar{B}_q(\vec{\xi})$ is the spatial spectral density, associated with the correlation function of the sources in the plane

$$B_q(\vec{\rho}_s)=\int d\vec{\xi}\bar{B}_q(\vec{\xi})e^{i\vec{\xi}\vec{\rho}_s}. \tag{9.20}$$

Accordingly

$$\bar{B}_q(\vec{\xi})=\frac{1}{(2\pi)^2}\int d\vec{\rho}_s B_q(\vec{\rho}_s)e^{-i\vec{\xi}\vec{\rho}_s}. \tag{9.21}$$

The function $f(\vec{\xi})$ used in Kuryanov (1963) is related to our notation by

$$\left\langle \bar{F}_{\omega}(\vec{\xi})\bar{F}^{*}_{\omega'}(\vec{\xi}')\right\rangle = f(\vec{\xi})\delta(\vec{\xi}-\vec{\xi}')\delta(\omega-\omega'). \tag{9.22}$$

The correlation function for the field in the plane of the sources

$$\langle S(\vec{r},0,0;\omega)S^{*}(\vec{r}',0,0;\omega')\rangle = B_{\omega}(\vec{r}-\vec{r}')\delta(\omega-\omega'),$$
$$B_{\omega}(\vec{\rho}) = \int d\vec{\xi} f(\vec{\xi})e^{i\vec{\xi}\vec{\rho}}. \tag{9.23}$$

Let us find the correlation function for the field

$$\langle S(\vec{r},z,0;\omega)S^{*}(\vec{r}',z',0;\omega')\rangle = B_{\omega}(\vec{r},z;\vec{r}',z')\delta(\omega-\omega') = B_{\omega}(\vec{R};\vec{R}')\delta(\omega-\omega'). \tag{9.24}$$

From expression (7.30) we obtain,

$$B_{\omega}(\vec{r},z;\vec{r}',z') = \int\int d\vec{\xi}d\vec{\xi}' f(\vec{\xi})\delta(\vec{\xi}-\vec{\xi}')e^{i(\vec{\xi}\vec{r}+k_z z)}\left[e^{i(\vec{\xi}'\vec{r}'+k_z' z')}\right]^{*} \tag{9.25}$$

In this integral, inhomogeneous waves must also be taken into account, which gives, in principle, a dependence on the coordinates and z and z' for the correlation function. At sufficiently large distance from the emitting planes, inhomogeneous waves can be neglected and the correlation function has the form

$$B_{\omega}(\vec{r}-\vec{r}',z-z') = \int d\vec{\xi} f(\vec{\xi})e^{i[\vec{\xi}(\vec{r}-\vec{r}')+k_z(z-z')]}. \tag{9.26}$$

Clearly, for $z = z'$ the correlation function at an arbitrary distance from the plane coincides with the correlation function (9.25) for the field in the plane, which is defined in Kuryanov as the "transport" correlation function of the field from the plane. For the spectral intensity of the noise field at the point of observation with coordinates $\vec{r}$, z:

$$I_{\omega}(\vec{r},z) = \frac{1}{2\rho c}B_{\omega}(0,0) = \frac{1}{2\rho c}\int d\vec{\xi} f(\vec{\xi})$$
$$= \frac{1}{2\rho c}\int\int k^2 f(\vartheta,\varphi)\cos\vartheta\sin\vartheta d\vartheta d\varphi = \int J(\vartheta,\varphi)d\Omega, \tag{9.27}$$

where we have introduced the ray intensity

$$J(\vartheta,\varphi) = \frac{k^2 f(\vartheta,\varphi)\cos\vartheta}{2\rho c} = \frac{g^2(\vartheta)\left|\tilde{Q}\right|^2}{8\rho c\cos\vartheta}. \tag{9.28}$$

Let us now consider the spatial correlation function of the noise (9.25) at a fixed frequency (the spectral correlation function) in an arbitrary inhomogeneous medium. Using the Fourier transform: one can associate the space–time correlation function with the functions $B_\omega(\vec{R},\vec{R}')$ as

$$B(\vec{R},\vec{R}',\tau)=\left\langle p(\vec{R},t+\tau)p(\vec{R}',t)\right\rangle=\int\limits_{-\infty}^{\infty}B_\omega(\vec{R},\vec{R}')e^{-i\omega\tau}d\omega. \tag{9.29}$$

The correlation function of the complex field can also be expressed in terms of the spectral correlation function:

$$\left\langle P(\vec{R},t)P^*(\vec{R}',t+\tau)\right\rangle=4\int\limits_{0}^{\infty}B_\omega(\vec{R},\vec{R}')e^{-i\omega\tau}d\omega. \tag{9.30}$$

Following Kuperman and Ingenito (1980), let us find an expression for the correlation function. For this, we shall use the Hankel transformation of the Green's functions $\bar{\Psi}(\xi,z,z_s)$:

$$\Psi(\vec{R},\vec{R}';\omega)=\int d\vec{\xi}\bar{\Psi}(\xi,z,z')e^{i\vec{\xi}(\vec{r}-\vec{r}')} \tag{9.31}$$

where $\vec{\xi}$ is a wavevector in the horizontal plane. Using, in addition, (9.24), (9.6) and (9.20), we obtain

$$\begin{aligned}
B_\omega(\vec{R},\vec{R}')&=\frac{1}{(2\pi)^4}\int d\vec{r}_s\int d\vec{r}'_s\int d\vec{\xi}\int d\vec{\xi}'B_q(\vec{r}_s-\vec{r}'_s),\\
&\quad\bar{\Psi}(\xi,z,z_s)\bar{\Psi}^*(\xi',z',z_s)e^{i[\vec{\xi}(\vec{r}-\vec{r}_s)-\vec{\xi}'(\vec{r}'-\vec{r}'_s)]}\\
&=\frac{1}{(2\pi)^4}\int d\vec{r}_s\int d\vec{\rho}\int d\vec{\xi}\int d\vec{\xi}'B_q(\vec{\rho}),\\
&\quad\bar{\Psi}(\xi,z,z_s)\bar{\Psi}^*(\xi',z',z_s)e^{i[\vec{\xi}(\vec{r}-\vec{r}_s)-\vec{\xi}'(\vec{r}'-\vec{r}_s-\vec{\rho})]}\\
&=\frac{1}{(2\pi)^2}\int d\vec{\rho}\int d\vec{\xi}B_q(\vec{\rho})\times\bar{\Psi}(\xi,z,z_s)\bar{\Psi}^*(\xi,z',z_s)e^{i\vec{\xi}(\vec{r}-\vec{r}'-\vec{\rho})}\\
&=\int d\vec{\xi}B_q(\vec{\xi})\times\bar{\Psi}(\xi,z,z_s)\bar{\Psi}^*(\xi,z',z_s)e^{i\vec{\xi}(\vec{r}-\vec{r}')}\\
&=\frac{1}{2\pi}\int d\vec{\rho}B_q(\vec{\rho})\int\limits_0^\infty\xi d\xi J_0(\xi|\vec{r}-\vec{r}'-\vec{\rho}|)\bar{\Psi}(\xi,z,z_s)\bar{\Psi}^*(\xi,z',z_s).
\end{aligned} \tag{9.32}$$

This equation was derived in Kuperman and Ingenito (1980), where expressions were also obtained for the correlation function of the noise for various forms of

directivity functions. The simplest case corresponds to δ-correlation in the horizontal plane and to dipole sources in the vertical plane.

Using the connection between Bessel functions and Hankel functions, we obtain

$$B_\omega(\vec{R},\vec{R}') = \frac{1}{4\pi}\int d\vec{\rho} B_q(\vec{\rho}) \int\limits_{-\infty}^{\infty} \xi d\xi H_0^{(1)}(\xi|\vec{r}-\vec{r}'-\vec{\rho}|)\bar{\Psi}(\xi,z,z_s)\bar{\Psi}^*(\xi,z',z_s). \quad (9.33)$$

In particular, for the noise intensity

$$I_\omega(\vec{R}) = \frac{2}{\rho c} B_\omega(\vec{R},\vec{R}) = \frac{1}{2\rho c\pi}\int d\vec{\rho} B_q(\vec{\rho}) \int\limits_{-\infty}^{\infty} \xi d\xi H_0^{(1)}(\xi\rho) \int\limits_{-\infty}^{\infty} \xi d\xi H_0^{(1)}(\xi\rho)\left|\bar{\Psi}(\xi,z,z_s)\right|^2$$

$$= \frac{2}{\rho c}\int d\vec{\xi} B_q(\vec{\xi})\left|\bar{\Psi}(\xi,z,z_s)\right|^2. \quad (9.34)$$

Expressions (9.32)–(9.34) include the Cron–Sherman formulae as a subcase if, for the Green's function $\bar{\Psi}(\xi,z,z_s)$, one uses the expression for a homogeneous half-space.

A transformation of expressions (9.32)–(9.34) associated with the modal decomposition of the Green's function can also be written, viz.

$$\bar{\Psi}(\xi,z,z_s) = \sum_l \frac{\Psi_l(z)\Psi_l(z_s)}{\xi^2-\xi_l^2} + \int\limits_{k_1}^{i\infty} d\eta \frac{\Psi_l(z)\Psi_l(z_s)}{\xi^2-\eta^2}, \quad (9.35)$$

which includes both the discrete and the continuous spectra. Assuming that the main contribution to the noise field comes from the discrete spectrum (the validity of which assumption will be discussed below), that is, keeping only the first term in (9.35) and substituting it into (9.33), we obtain, after integration in the complex plane, the expression for the correlation function:

$$B_\omega(\vec{R},\vec{R}') = \frac{1}{4}\sum_{l,m}\int d\vec{\rho} B_q(\vec{\rho}) \frac{\Psi_l(z)\Psi_l(z_s)\Psi_m(z')\Psi_m(z_s)}{\xi_l^2-(\xi_m^*)^2}$$
$$\left[H_0^{(1)}(\xi_l|\vec{r}-\vec{r}'-\vec{\rho}|) + H_0^{(1)}\left(\xi_m^*|\vec{r}-\vec{r}'-\vec{\rho}|\right)\right], \quad (9.36)$$

which, as can be easily seen, specifically depends on the difference in coordinates in the horizontal plane $\vec{r}-\vec{r}'$ and on the depths z and z'. Retaining only the largest terms in the sum, that is, where the denominator contains a difference of eigenvalues with the same indices $\xi_l^2-(\xi_l^*)^2 = 2iq_l\gamma_l$, we obtain the expression

$$B_\omega(\vec{r}-\vec{r}';z,z') = \frac{1}{4}\sum_l \int d\vec{\rho} B_q(\vec{\rho}) \frac{\Psi_l(z)\Psi_l(z')\Psi_l^2(z_s)}{q_l\gamma_l} J_0(q_l|\vec{r}-\vec{r}'-\vec{\rho}|). \quad (9.37)$$

As a rule, noise sources are assumed to be δ-correlated, that is

$$B(\vec{\rho}) = q_\omega^2 N_0 \frac{\delta(\rho)}{2\pi\rho}. \tag{9.38}$$

Carrying out the integration in (9.33), we obtain

$$B_\omega(\vec{r} - \vec{r}'; z, z') = \frac{q_\omega^2 N_0}{4} \sum_l \frac{\Psi_l(z)\Psi_l(z')\Psi_l^2(z_s)}{q_l \gamma_l} J_0(q_l |\vec{r} - \vec{r}'|). \tag{9.39}$$

The vertical correlation function, corresponding to the case $\vec{r} = \vec{r}'$ is

$$B_\omega(z, z') = \frac{q_\omega^2 N_0}{4} \sum_l \frac{\Psi_l(z)\Psi_l^2(z_s)\Psi_l(z')}{q_l \gamma_l}. \tag{9.40}$$

In turn, for the intensity of the noise, we have

$$I_\omega(z) = \frac{q_\omega^2 N_0}{2\rho c} \sum_l \frac{\Psi_l^2(z)\Psi_l^2(z_s)}{q_l \gamma_l}. \tag{9.41}$$

We note that this intensity corresponds to taking into account only the contribution of the discrete spectrum. One can also obtain the intensity including the contribution of the continuous spectrum.

9.2 Relationship Between the Continuous and the Discrete Components of the Field of Surface Noise Sources in a Waveguide

An important question in calculating the overall noise field is: what is the role of the continuous spectrum in the noise field from surface sources? It is well known that for a point source, the role of the continuous spectrum decreases rapidly as the receiver moves away from the source. However, in the present analysis, the role of the fields formed by both closely located and distant sources is not at all clear, and so requires detailed consideration. Of course, if the noise source is directly above the receiver, then the proportion of "direct" rays is large and, accordingly, the contribution of the continuous spectrum may be large. As the "sound-suppressing" region increases, the proportion of sound from distant sources, i.e., the discrete spectrum, grows. Thus, the question of the role of the continuous spectrum is associated with the study of the field from a bounded region. Analogously, there is a need to analyze the role of other factors: the frequency of the field, the refractive index in the bottom, and the overall depth. It seems clear that, the role of the continuous spectrum depends on the specific

problem; in some waveguide conditions, this contribution is significant, whereas in others it can be neglected. It is intuitive for example that the contribution of the continuous spectrum must grow with the sound frequency, since this corresponds to an effective increase in the waveguide depth. It is almost impossible to give a general answer to the question of the role of the continuous spectrum, however, and so we shall attempt to consider this question only for typical shallow water conditions and comparatively low frequencies.

According to the results of Kuperman and Ingenito (1980), the relative contributions of the discrete and continuous spectra are determined by the losses in the bottom. If the losses are large, the "near field," i.e., the continuous spectrum, predominates, whereas in the case of small bottom loss, the contribution of the "far field," i.e., the discrete spectrum, is dominant. Accordingly, in the calculations of the noise field at frequencies up to 1 kHz, Buckingham (1980) takes into account only the discrete spectrum of the field. The role of the continuous spectrum was also analyzed in Hamson (1985), where the response of vertical and horizontal receiver arrays to the noise field was analyzed. There, it was noted that the relative contribution of the components is also determined by the sector of angles in the vertical directional diagram: near the horizontal direction (for small angles from the horizontal) the discrete component dominates, whereas near the vertical direction, the continuous component dominates. It was also shown there that there is a dependence on the absorptive properties of the bottom. All the results there were given for a frequency of 480 Hz. The frequency dependence of the relative contribution of the continuous and discrete fields to the noise was considered in Kuperman and Ferla (1985), where for a specific region, the effect of the wind noise was considered. It was shown that for a comparatively low wind speed the contribution of the discrete component dominates at low frequencies (up to 200 Hz), whereas for higher frequencies the main contribution comes from the continuous spectrum. The scattering of sound by surface waves played a significant role in this analysis.

Therefore, let us consider the intensity of the noise field taking into account both the discrete and continuous spectra. Substituting the expression for the Green's function in the form of a decomposition into waveguide modes (3.6), we obtain for the sound field an expression in the form of a sum of two components, corresponding to the discrete and continuous spectra:

$$S(r, z, z_s; \omega) = S^{\mathrm{d}}(r, z, z_s; \omega) + S^{\mathrm{c}}(r, z, z_s; \omega), \tag{9.42}$$

where

$$S^{\mathrm{d}}(r, z; \omega) = \frac{i}{4} \sum_l \Psi_l(z)\Psi_l(z_s) \int Q_\omega(\vec{r}_s) H_0^{(1)}(\xi_l |\vec{r} - \vec{r}_s|) d\vec{r}_s,$$

$$S^{\mathrm{c}}(r, z; \omega) = \frac{i}{4} \int_{i\infty}^{k_1} \xi d\xi \Psi_\xi(z)\Psi_\xi(z_s) \int Q_\omega(\vec{r}_s) H_0^{(1)}(\xi |\vec{r} - \vec{r}_s|) d\vec{r}_s. \tag{9.43}$$

In this notation, $|P(r,z,t)|^2/2\rho c$ is the intensity (energy flux density), and $B_\omega(\vec{R},\vec{R})/2\rho c$ is the spectral density of the noise power. This quantity can be found by substituting (9.41) and (9.42) in (9.34), where we assume that the point of observation is at some depth on the z-axis, i.e., $\vec{r}=0$. Then

$$I_\omega(z)=B_\omega(0,z;0,z)/2\rho c=\frac{q_\omega^2 N_0}{32\rho c}\int d\vec{r}\left|\sum_l \Psi_l(z)\Psi_l(z_s)+\int_{i\infty}^{k_1}\xi d\xi\Psi_\xi(z)\Psi_\xi(z_s)\right|^2. \tag{9.44}$$

If we neglect the cross terms between the discrete and continuous spectrum, then the intensity of the noise field can be approximated in the form of a sum of the intensities of the discrete and continuous components

$$I_\omega(z)=I_\omega^{\mathrm{d}}(z)+I_\omega^{\mathrm{c}}(z), \tag{9.45}$$

where

$$\begin{aligned} I_\omega^{\mathrm{d}}(z)&=\frac{q_\omega^2 N_0}{32\rho c}\int d\vec{r}\left|\sum_l \Psi_l(z)\Psi_l(z_s)\right|^2,\\ I_\omega^{\mathrm{c}}(z)&=\frac{q_\omega^2 N_0}{32\rho c}\int d\vec{r}\left|\int_{i\infty}^{k_1}\xi d\xi\Psi_\xi(z)\Psi_\xi(z_s)\right|^2. \end{aligned} \tag{9.46}$$

Let us now assess the role of the continuous spectrum for the noise field in a simple waveguide model. We shall suppose that the medium in question is a Pekeris waveguide, where the absorption coefficient of sound in the bottom depends on frequency according to (3.1). For the Pekeris model, the functions describing the continuous spectrum were given in Sect. 3.2.

To determine the contribution of the continuous spectrum in the second integral in (9.46), the integral is calculated numerically by direct integration; whereas the intensity of the discrete spectrum component was computed based on expression (9.41). Figure 9.3 shows $\delta_n(f)=I_f^{\mathrm{d}}/I_f^{\mathrm{c}}$ as a function of frequency for waveguide parameters: $m_1=1.4$, $n_1=c/c_1=0.83$, $\alpha=0.025$, and $H=100\,\mathrm{m}$. Clearly, for low frequencies $f\le 300-500\,\mathrm{Hz}$, we have $\delta_n\gg 1$, so that in this frequency regime the continuous spectrum can be neglected. These results are in qualitative agreement with the analysis of Kuperman and Ferla (1985), who also concluded that, while the role of the continuous spectrum grows with frequency, the frequency boundary corresponding to the dominance of a particular part of the spectrum depends strongly on specific conditions. A more precise analysis for the relative contributions of the continuous and discrete components of the noise intensity also involves an analysis of the shape and dimensions of the emitting region, which we will not pursue here.

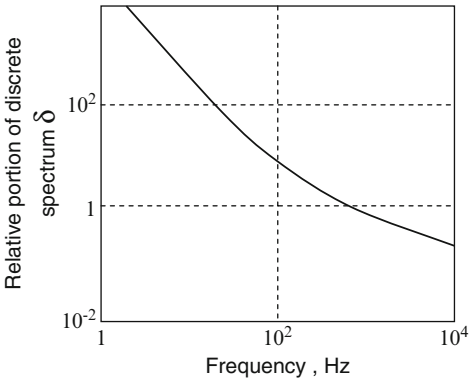


Fig. 9.3 Frequency dependence of the contribution of the discrete spectrum

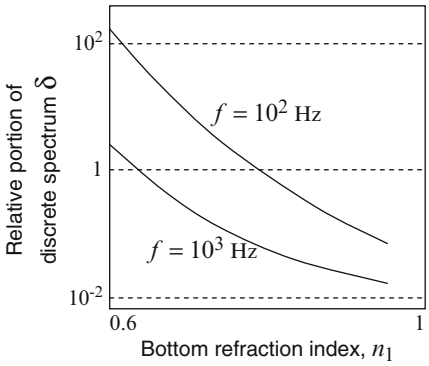


Fig. 9.4 Dependence on the refractive index of the bottom of the contribution of the discrete spectrum to the noise field

It is also interesting to consider the relative contribution of the continuous spectrum as a function of the refractive index of the bottom n_1, which determines the number of trapped modes. It is intuitively clear that the closer n_1 is to unity, the fewer propagating modes there are and the smaller the proportion of the discrete spectrum in the intensity; naturally, there is also a significant frequency dependence, since the frequency also determines the number of trapped modes. This property is illustrated in Fig. 9.4, which shows the change in δ_n as a function of the refractive index n_1 for some representative frequencies, using the above parameters.

9.3 Effect of the Frequency Dependence of the Bottom Absorption Coefficient on the Noise Spectrum

One of the most important characteristics of the noise field is its spectral intensity at the point of reception, measured in decibels relative to 1 μPa/$\sqrt{\text{Hz}}$. Many works are devoted to this quantity, in both deep (Knudsen et al. 1948; Ross 1976; Wenz 1962) and shallow water (Kuperman and Ferla 1985; Piggott 1965), and the significant

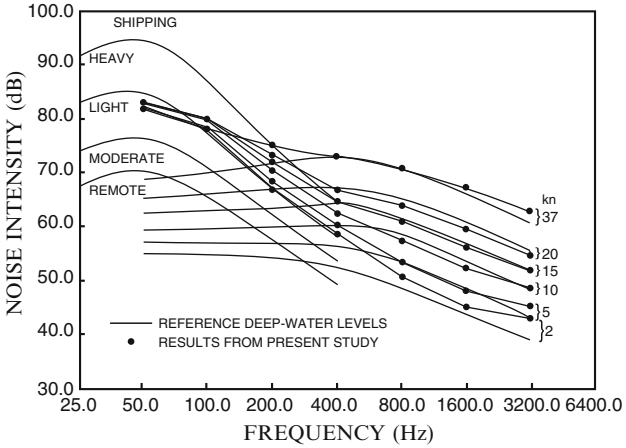


Fig. 9.5 Noise spectra in deep (*continuous curves*) and shallow (shown by *dots*) water for various wind speeds, compared with analogous spectra in deep water [from the results of Kuperman and Ferla (1985)]

difference of the spectra of deep and shallow water is noted. This difference is manifest in the (usually) higher noise level in shallow water compared to deep water in the comparatively low frequency band. At higher frequencies there is an approximately identical drop off of intensity with frequency of around 5 dB/octave. This apart, there is also a far greater variability of shallow-water noise as a function of environmental conditions (weather, geographical position, etc.). Figure 9.5 shows the results of experimental studies of the spectral intensity of the noise in a shallow region of the Mediterranean Sea, given in Kuperman and Ferla (1985), in comparison with noise spectra in deep water based on the results of Ross (1976). Moreover, Kuperman and Ferla (1985), Crouch and Burt (1972) and Perrone (1969) studied the dependence of the spectral characteristics on the wind speed. The physically natural growth of the sound intensity due to a wind-generated disturbance, with increasing wind speed v_w, was demonstrated. Here, the dependence of the spectral intensity on the wind speed is exponential in nature, i.e., $I_\omega = \text{const} \cdot (v_w)^m$ with the exponent m varying from 0.4 to 4 as a function of frequency and propagation conditions.

Of course, the greater variability of noise in shallow water, in comparison with deep water, is due to the vastly greater effect of the bottom. Let us analyze, in particular, how the frequency dependence of the absorption coefficient (see, for example, Chap. 6) is manifested in the observed spectrum of the noise field. We shall consider this effect in a simple model of shallow water consisting of a water layer of thickness H with density ρ and sound speed $c(z)$, and a bottom in the form of a homogeneous fluid absorbing half-space with density ρ_1, and sound speed c_1, with the absorption coefficient depending on frequency according to an exponential law $\alpha = \alpha_0 (f/f_0)^q$, where the parameters α_0, f_0, q are determined by the bottom type. For example, according to Kuperman and Ingenito (1980), for sand $\alpha \approx 0.009 f^{1.78}$ dB/km, whereas for clay $\alpha \approx 0.0065 f^{1.3}$ dB/km, where the frequency is measured in Hz.

If we assume that the effective sources are distributed at a small depth in comparison with the wavelength, z_s, then the spectral intensity of the noise field in this case is determined by (9.41). We also assume that we are dealing with a homogeneous distribution of sources on the sea surface. The spectral intensity of the noise can then be rewritten in the form

$$I_\omega(z) = \frac{q_\omega^2 N_0}{2\rho c} z_s^2 \sum_l \frac{\Psi_l^2(z)[\Psi_l'(0)]^2}{q_l \gamma_l}. \tag{9.47}$$

Instead of the unknown parameter being the depth of the sources, we can introduce another equivalent parameter, the moment of the effective dipole source. In this picture, we shall be dealing with the spectral density of the dipole distribution. In other words, we have N_0 dipoles with moment $M_\omega = 2z_s V_\omega$ per unit area (where V_ω is the spectral component of the monopole volume pulsation speed of an individual source). Such a model is often used and in this case, we may suppose that the noise field is formed by dipoles actually distributed on the sea surface:

$$I_\omega(z) = \frac{\rho\omega^2 N_0 M_\omega^2}{8c} \sum_l \frac{\Psi_l^2(z)[\Psi_l'(0)]^2}{q_l \gamma_l}. \tag{9.48}$$

For subsequent transformations of (9.48), we will assume that we can use the WKB approximation for the eigenfunctions and the eigenvalues. We note that in connection with the field of sources on the surface, the lower turning point in formulae (4.10) will be $z = 0$. Thus

$$\Psi_l(z) = 2\sqrt{\frac{q_l}{\sigma_l(z) D_l}} \sin \int_0^z \sigma_l(z')dz', \quad \gamma_l \approx \frac{2s\sigma_l(H)}{kD_l}, \tag{9.49}$$

where s is determined by (4.23). For values of the eigenfunction at the point z_s, such that $z_s \ll \lambda$, we obtain

$$\Psi_l(z_s) = 2\sqrt{\frac{q_l \sigma_l(0)}{D_l}} z_s. \tag{9.50}$$

Expression (9.48) thus takes the form

$$I_\omega(z) = \frac{\rho\omega^2 k M_\omega^2 N_0}{2cs} \sum_l \frac{q_l \sigma_l(0)}{\sigma_l(z)\sigma_l(H) D_l} \sin^2 \int_0^z \sigma_l(z')dz'. \tag{9.51}$$

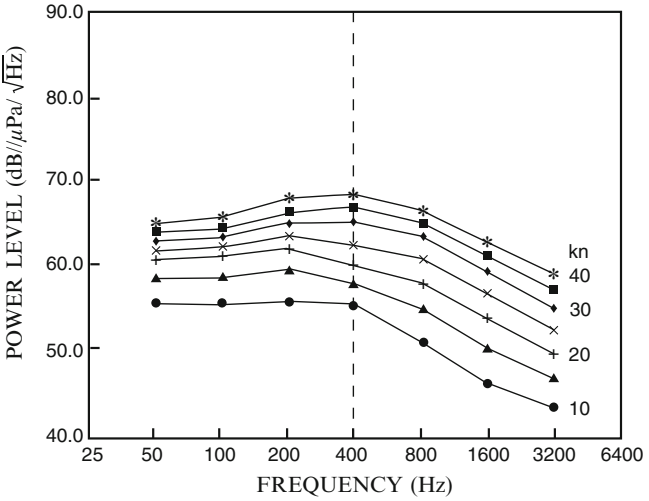


Fig. 9.6 Spectral level of acoustic radiation in the frequency band from 25 Hz to 3.2 kHz as a function of wind speed (shown at the knotes) [based on Kuperman and Ferla (1985)]

If we consider this expression as a function of frequency, then it is clear that the sum on the right-hand side of the equality depends weakly on frequency, so that the frequency dependence of the intensity is mainly determined by the function

$$G(f) = \frac{4\pi^3 \rho f^3 M_\omega^2 N_0}{c^2 s}. \tag{9.52}$$

This factor contains both information about the sources (their spectral level) and information about the waveguide. As far as the spectral level of emission is concerned, it is the result of a "microscopic" mechanism, responsible for the emission of noise (for example, wind-generated disturbances). This is an entire topic of research in its own right, both experimental (for example, Wolf and Ingenito 1982) and theoretical, based on various model ideas about the source mechanism. Kuperman and Ferla (1985) were able to recover the spectral level of wind noise emission based on measurements of the spectral intensity of the noise in the Mediterranean Sea. These results for q_ω are shown in Fig. 9.6. Clearly, this quantity depends on the wind strength and acoustic frequency, and is very smooth over a broad frequency interval. It has a practically constant value and only shows a small maximum at frequency ~400 Hz; the subsequent fall in level in the region up to 1 kHz amounts to at most 5 dB. If we take a similar dependence for the sources in our case, then carrying out the calculations according to (9.51), we obtain results for two types of bottom (sand and clay), which are shown in Fig. 9.7. The frequency is on a logarithmic scale starting at 30 Hz. Clearly, the nature of the fall in intensity with frequency (the slope of the curves) varies; this variation amounts to around 5 dB at ~400 Hz, which is in excellent agreement with the experimentally recorded value.

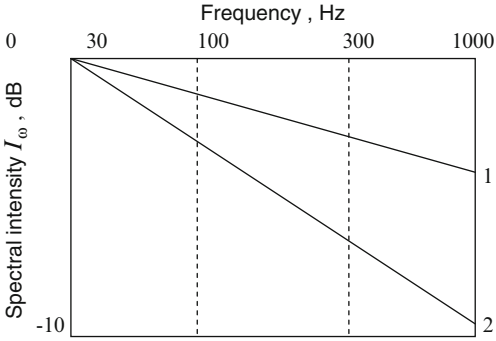


Fig. 9.7 Frequency spectrum of intensity relative to the level at 31 Hz for various types of bottom: 1 – for clay, 2 – for sand

9.4 Vertical Directivity of the Sound Field and the Effect of Random Inhomogeneities

Another important measurable characteristic of the noise field is the directivity of the noise, that is the power (intensity) flux density receivable per unit solid angle; in normalized form, this quantity gives the directivity pattern of the noise field. In our formulation of the problem, where we assume anisotropic waveguide properties, the directivity pattern in the vertical plane is of interest. For the simple Cron–Sherman model, this quantity can be obtained directly from (9.10). Assuming a solid angle element $d\Omega = 2\pi \sin\vartheta d\vartheta$, (9.10) can be rewritten in the form:

$$I(z) = \frac{1}{32\pi^2 \rho c} \int_0^{\pi/2} \frac{|g(\vartheta)|^2}{\cos\vartheta} d\Omega = \int I(\vartheta) d\Omega, \tag{9.53}$$

where

$$I(\vartheta) = \frac{1}{32\pi^2 \rho c} \frac{|g(\vartheta)|^2}{\cos\vartheta} \tag{9.54}$$

is the intensity transported into a unit solid angle. Often, when it is only required to determine the overall form of this dependence, a normalized quantity is used (called the directivity pattern), where the nature of the normalization can be chosen in different ways, for example, at the maximum value or with unity taken to be at $\vartheta = 0$ or at $\vartheta = \pi/2$. From (9.54), it is clear that the directivity pattern of the noise in the Cron–Sherman model coincides with the directivity pattern of the emitter, modulo the cosine of the vertical angle. The appearance of the cosine is due to the angle at which a surface radiator element is visible from the point of observation. This simple relationship between the directivity patterns of the received and the emitted signal is due to the fact that in our calculations, we have not taken into account either absorption or scattering by the medium, which, naturally, changes the characteristics

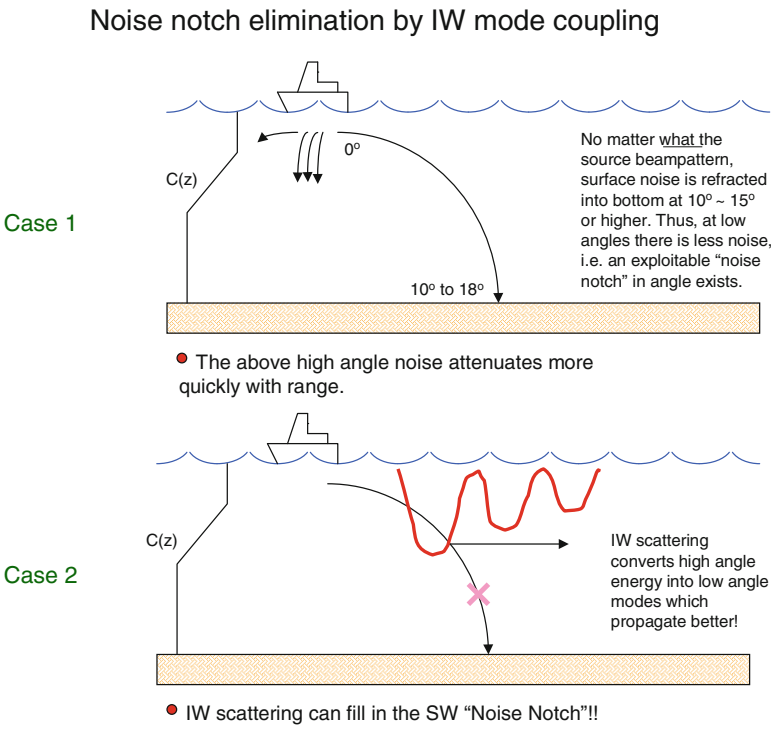


Fig. 9.8 *Top panel* shows a downward refracting shallow water sound-speed profile resulting in surface generated sound interacting with the bottom at 10–15° grazing angle, and thus not being seen by beamformers that steer to lower grazing angles. The *bottom panel* shows the filling in of the noise notch by scattering (redistribution of modal energy to lower modes and angles), in this case by solitons

of the directivity of the propagation of sound. In a more general formulation of the problem, one has to explain how the vertical angular distribution of the radiation from the sources is transformed, as a result of oceanic factors (absorption, scattering, interaction with the bottom and with the surface, etc.), to the observed dependence of the intensity field on vertical angle.

In low latitudes or summer mid-latitudes, the shallow water sound velocity profile is usually downward refracting. This causes a significant amount of the low frequency noise produced at the surface (predominantly shipping or surface wave noise) to interact with the bottom at angles greater that zero degrees, and in fact typically at 10–15° or greater, as shown in Fig. 9.8. This means that no surface noise will be encountered by a beam former that is steered to vertical angles lower than the smallest bottom (grazing) angle that is dictated by Snell's law. This phenomenon occurs in both deep and shallow water, and is called the "noise notch."

This noise notch can also be "filled in" by scattering effects, due to the ocean surface, volume, and bottom. As has been previously discussed, some of the most significant scattering seen in shallow water comes from the volume scattering by

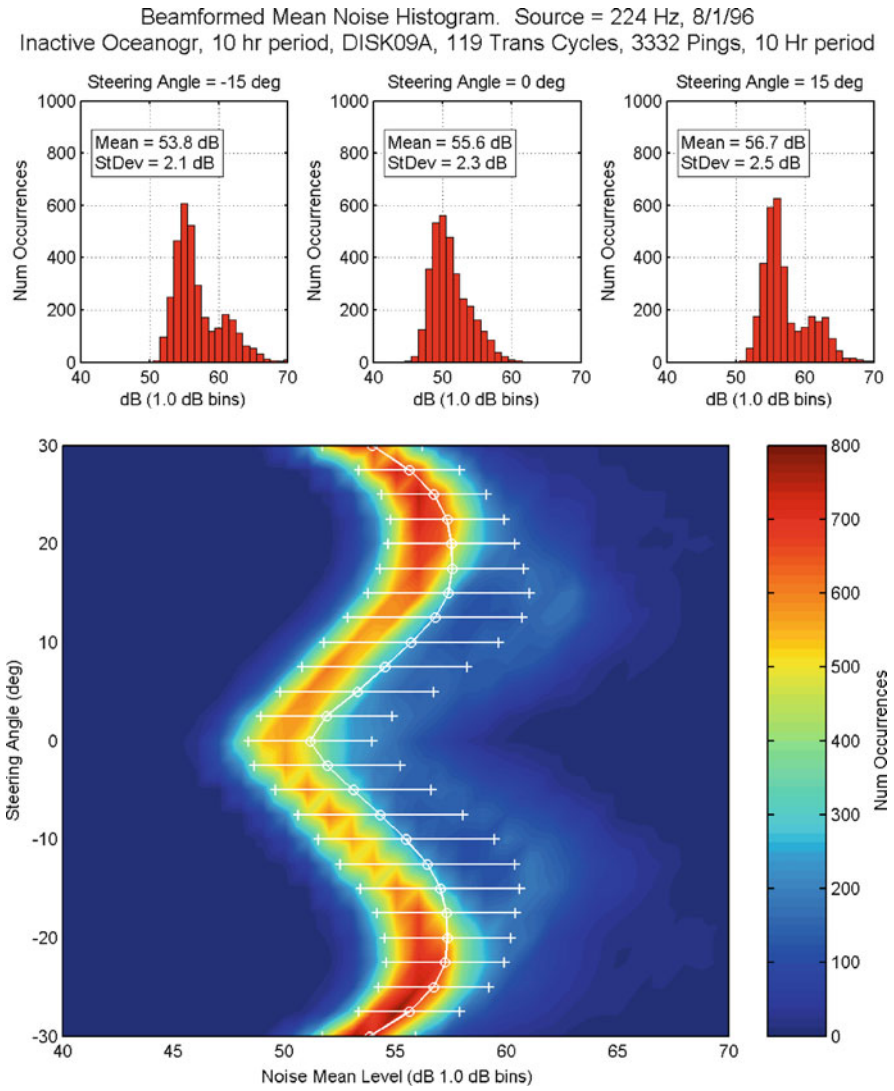


Fig. 9.9 Shallow water noise notch observed during the summer New England shelfbreak PRIMER experiment, 8/1/96

internal waves, especially nonlinear internal waves. This effect is shown in the bottom panel of Fig. 9.8. Moreover, the scattering that fills in the notch redistributes high-order normal modes into low-order modes, the latter of which decay very slowly in range. Thus the filling in of the notch, which occurs locally at the sites of the internal waves, can extend over a large distance away from the internal waves.

A rather nice experimental example of the shallow water noise notch is shown in Fig. 9.9 (courtesy Mr. Phil Abbot, OASIS Inc.). This shows an almost 7 dB difference

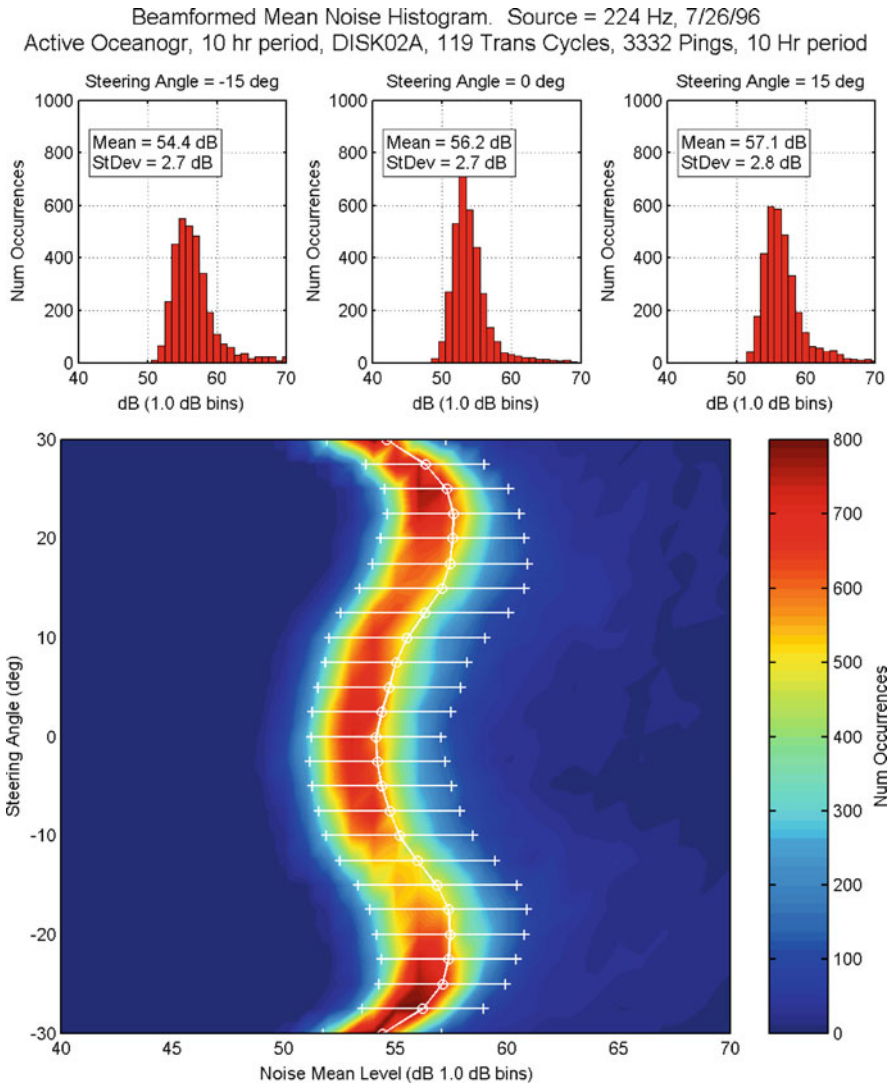


Fig. 9.10 Nearly filled in noise notch, due to the scattering of sound by nonlinear internal waves

in the noise at 224 Hz between the 0° point and the peak at ~22° (the bottom critical incidence angle). The filling in of the noise notch can be seen in noise data taken about a week earlier (see Fig. 9.10), on 7/26/96, where the difference between the zero degree beam and the 20° beam is only 1–2 dB. Indeed, there are times during which the noise notch is entirely filled in. One of the most interesting questions about the fluctuations of the noise notch is how they correlate to the environmental conditions, particularly the time and space variations in the scattering field. Work with low frequency acoustic data from both the 1996 PRIMER experiment and the

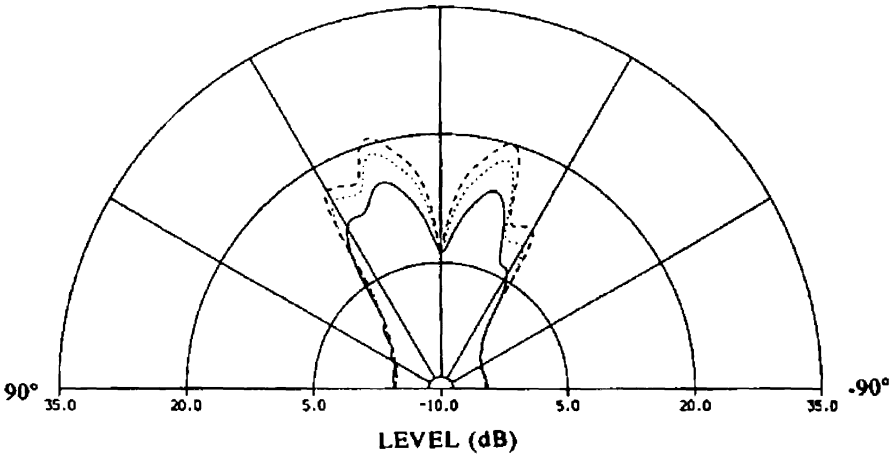


Fig. 9.11 Angular dependence of the noise field intensity as measured by vertical arrays with aperture 8, 16, and 24 wavelengths, respectively, corresponding to the following curves: *continuous* – 16 hydrophones; *dots* – 32 hydrophones; *dashes* – 48 hydrophones [from the results of Yang and Kwang (1997)]

2001 ASIAEX South China Sea experiment points toward there being a significant correlation between the coastal soliton field and the noise notch fluctuations (P. Abbot and T. Duda private communication).

For problems in a waveguide, where the sound field is a sum of propagating modes, the angular distribution of intensity is replaced by the modal distribution. In this case, one can speak about the angular distribution having in mind the corresponding Brillouin angles for the propagating modes. The vertical directivity of the noise is a very important and informative quantity; thus, great attention is paid to its study, both experimentally (Buckingham 1980; Buckingham and Jones 1987; Carbone et al. 1998; Cavanagh and Renner 1980; Aredov 1988) and theoretically (Yang and Kwang 1997; Hamson 1985). In particular, Buckingham and Jones (1987) presented experimental data from which it was clear that in the vertical plane the intensity distribution may have a maximum at zero angle from the horizontal. Yang and Kwang (1997) present results from measurements of the angular dependence of the intensity, some of which are shown in Fig. 9.11. For the case of deep water, Cavanagh and Renner (1980) considered the connection between the vertical directivity and the vertical intensity distribution of the noise field in a ray approximation.

Certain properties of the angular distribution of the intensity can be easily understood from simple considerations, for example, the value of the angle χ_0 from the horizontal, at which the intensity is zero (becomes small) coincides with the critical angle of reflection from the bottom, so that a knowledge of that dependence provides additional information about the properties of the bottom. In deep water, a technique for determining properties of the coefficient of reflection from the bottom from the directivity characteristics of the noise field was proposed

and realized in Aredov and Furduev (1994). Carbone et al. (1998) undertook a detailed study of the effects of the waveguide parameters on the vertical directivity for shallow water. They showed, in particular, that the source directivity characteristics and properties of the bottom have the greatest effect on the noise directivity, where one intrinsic special point in the angular dependence of the intensity is given by the value of the total internal reflection angle. Absorption in the bottom due to the presence of shear waves also has a significant effect. These factors change the angular dependence of the reflection coefficient of the bottom and thus the angular characteristics of the received noise also. The sound speed profile affects the noise directivity to a lesser extent. However, its effect is highly typical. One can qualitatively analyze (in the language of rays) the properties of the angular distribution for different types of sound speed profiles: let us consider a negative gradient (near-bottom channel) and a positive gradient (near-surface channel). In the first case, rays emanating from sources on the surface tend toward the bottom, and at the point or receipt, there are no rays propagating horizontally (with zero grazing angle). This corresponds to the presence of a trough in the angular dependence (the noise notch effect). For the case of a surface channel, rays emanating from sources tend toward the surface and so horizontally directed rays occur at the point of receipt. For certain values of the waveguide parameters and characteristics of the emitters, this may even lead to the appearance of a maximum at 0°. Corresponding numerical examples, indicating the possibilities for determining parameters of the bottom from the angular characteristics of the noise are given in Carbone et al. (1998). Hamson (1985) presents numerical calculations of the characteristics of the antenna array directivity for various environmental conditions in the case of a pressure-release surface and a solid bottom, for Summer and Winter sound speed profiles at a fixed frequency $f = 480$ Hz. In addition to the sound speed profiles, the parameters of the bottom and the characteristics of the radiators, the various mechanisms for scattering of sound (random inhomogeneities in the water column, and also roughness of the bottom and the surface) are important. These factors have been incorporated in a general formulation of the problem including scattering by Kuryanov and Klyachin (1984), within the framework of the theory of radiation transport. Based on numerical analysis, Kuznetsova (1989) gives an interpretation of the experimental data obtained in Buckingham and Jones (1987), taking into account both attenuation in the bottom and absorption in the water, and also scattering by a rough ocean surface.

In this section, in the context of a shallow-water model, we will obtain equations determining how the redistribution of the angular intensity takes place as a result of angle-dependent absorption in the bottom and scattering by random inhomogeneities.

We shall consider a waveguide mode where we have a water layer of thickness H, with sound speed $c(z)$ and density ρ, overlying a homogeneous fluid half-space (c_1, ρ_1, α). The sound speed in the water column is a random function of the space coordinates and time, $c = \langle c \rangle + \delta c$, which gives expression (5.16) for the refractive index. In the (r, z) coordinate system, the receiver is located at the point $(0, z)$.

We are interested in the angular distribution of the intensity of the noise, arriving from the whole of the waveguide surface. When there are no inhomogeneities, this distribution can be found from expression (9.41), and it is clear that the modal intensity is proportional to the expression $q_\omega^2 N_0 \Psi_l^2(z_s)/2\rho c q_l \gamma_l$, which, in the WKB approximation and assuming a dipole nature of the noise sources, gives $2N_0 M_\omega^2 q_l \sigma_l(0)/\rho c D_l \gamma_l$. Let us now extend our description of the noise field from a distributed source, taking into account the random inhomogeneities. In these discussions, we will adopt the diffusion approach. Our analysis will be somewhat simplified, including the assumption of azimuthal symmetry, and taking into account only the discrete spectrum (in other words, the far field is assumed to be dominant). In this case, for the spectral component of the sound field at the point $(r, 0)$, we shall use expression (9.6), where as the Green's function we take a random function expressed as a decomposition into modes:

$$S(0, z, z_s; \omega) = \int Q_\omega(\vec{r}) \sum_l c_l(r) \frac{\Psi_l(z) \exp[\mathrm{i}(q_l r + \pi/4)]}{\sqrt{8\pi q_l r}} d\vec{r}, \tag{9.55}$$

where the mode amplitudes $c_l(r)$ are random functions of the coordinates. Corresponding stochastic equations can be obtained for these amplitudes, from which, after some averaging, one obtains equations for the quantities $W_l(r) = \left\langle |c_l(r)|^2 \right\rangle$, which are proportional to the intensity transported by the mode with index l.

Using the assumption (9.38) that the sources in the plane are δ-correlated, we obtain for the vertical correlation function

$$B_\omega(z, z') = \frac{q_\omega^2 N_0}{4} \sum_l \int_0^\infty dr W_l(r) \frac{\Psi_l(z)\Psi_l(z')}{q_l}. \tag{9.56}$$

When there are no random inhomogeneities, $W_l(r) = \Psi_l^2(z_s) \exp(-\gamma_l r)$ and from (9.56), we obtain (9.26). The spectral energy density (mean square pressure) or the average intensity is also expressed in terms of $W_l(r)$:

$$I_\omega(z) = \frac{q_\omega^2 N_0}{2\rho c} \sum_l \int_0^\infty dr W_l(r) \frac{\Psi_l^2(z)}{q_l}. \tag{9.57}$$

A system of equations for determining $W_l(r)$ was also given in Chap. 4, where the value of $W_l(0)$ for a monopole was given for the "initial" conditions. If we consider the dipole as our source, then $W_l(0) = \frac{2q_l \sigma_l(0)}{D_l}$. Assuming a sufficiently large number of propagating modes, we can move from a discrete variable (the mode index) to a continuous variable and obtain the diffusion equation (4.47) for $W_p(r)$, where the diffusion coefficient a_p is determined by the waveguide parameters and by the random inhomogeneities. For example, when we have a waveguide of constant depth, with a constant average sound speed, instead of the mode index, we consider

the continuous variable $p : 0 \le p \le p_{\max} \sim kH/\pi$. The value of $p_{\max}$ is determined by the angle of total internal reflection from the bottom χ_c. The connection between p and the natural physical quantity, the grazing angle of a sound ray with the bottom, is determined in the simple case of a rigid bottom by the relations:

$$p = \frac{kH}{\pi} \sin\chi, \quad p_{\max} = \frac{kH}{\pi} \sin\chi_c. \tag{9.58}$$

In the case of a continuous variable, the summation formulae of (9.39)–(9.41) are replaced by integrals, where the value of $W_p(r)$ is determined from (5.58) with the appropriate initial conditions. Assuming dipole sources in a waveguide with constant sound speed, $W_p(0) = \frac{2\pi^2 z_s^2}{H^3} p^2$. For the boundary conditions, we must take

$$\frac{\partial W_p(r)}{\partial p}\big|_{p=0} = 0, \quad W_p(r)\big|_{p=p_{\max}} = 0. \tag{9.60}$$

In the calculation, we also perform averaging over the position of the receiver, which gives a smooth dependence on depth for the intensity.

Thus, (9.57) gives the total intensity of the noise field at depth z. The integral analogue of this formula has the form

$$I_\omega(z) = \frac{q^2 N_0}{\rho c} \int_0^{p_{\max}} dp \int_0^{\infty} dr \frac{W_p(r)}{\sigma_p(z) D_p}. \tag{9.61}$$

Based on this equation, the angular distribution of the intensity, in which we are interested, has the form

$$I_\omega(p,z) = \frac{q^2 N_0}{\rho c} \int_0^{\infty} \frac{W_p(r)}{\sigma_p(z) D_p} dr = \frac{q^2 N_0}{\rho c} \frac{Y_p(z)}{\sigma_p(z) D_p(z)}, \tag{9.62}$$

where

$$Y_p = \int_0^{\infty} W_p(r) dr. \tag{9.63}$$

We shall consider the value of Y_p as a function of the parameter p (or the angle in the vertical plane). We note that Y_p satisfies an ordinary differential equation, which can be obtained by integrating (5.58) over r from 0 to ∞, assuming $W_p(\infty) = 0$:

$$\frac{d}{dp}\left(a_p \frac{dY_p}{dp}\right) - \gamma_p Y_p + W_p(0) = 0, \tag{9.64}$$

The "initial conditions" for Y_p have the form

$$\frac{dY_p}{dp}\Big|_{p=0} = Y_p\Big|_{p=p_{\max}} = 0. \tag{9.65}$$

We can now obtain concrete results within the model, and when $\langle c\rangle = \text{const}$, correspondences between the unperturbed functions and the eigenvalues (i.e., without taking random inhomogeneities into account) are obtained with the Pekeris model. We will further assume that the noise field is due to uniformly distributed point dipoles (or monopoles located close to the surface). We additionally assume, analogous to our works in Sect. 5.4, that the propagation of the noise in the waveguide takes place only in a small sector of grazing angles, so that the diffusion coefficient for these values can be assumed to be independent of p: $a_p = a = \text{const}$. The loss coefficient corresponding to the Pekeris model was given in (4.24); thus the equation for the function Y_p has the form

$$a\frac{d^2Y_p}{dp^2} - \frac{s\pi^2}{k^2H^3}p^2Y_p = -\frac{2\pi^2 z_s^2}{H^3}p^2 \tag{9.66}$$

with the initial conditions (9.65). The solution of this equation is expressed in terms of a Bessel function with an imaginary argument. According to (9.62) one can obtain for $I_\omega(p)$ the expression (for the Pekeris model, independent of depth)

$$I_\omega(p) = \frac{\rho\omega^3 N_0 M_\omega^2}{4Hsc^2}\left[1 - \sqrt{\frac{p}{p_{\max}}}\frac{I_{-1/4}\left(\hat{\varepsilon}\frac{\pi^2p^2}{k^2H^2}\right)}{I_{-1/4}\left(\hat{\varepsilon}\frac{\pi^2p_{\max}^2}{k^2H^2}\right)}\right], \tag{9.67}$$

where $\hat{\varepsilon} = \frac{k}{2\pi}\sqrt{\frac{sH}{a}}$.

Since, in what follows, we shall be interested only in the form of the vertical directivity of the noise, we represent (9.67) in the form:

$$I_\omega(p) = I_\omega(0)F(p) \tag{9.68}$$

where the function $F(p)$ is normalized so that $F(0) = 1$ defines the vertical directivity of the noise field. If in expression (9.67) we change the argument to the angle from the horizontal χ, using (9.58), then for the directionality function we obtain the expression

$$F(\chi) = \frac{C}{\cos\chi}\left[1 - \sqrt{\frac{\sin\chi}{\sin\chi_0}}\frac{I_{-1/4}(\hat{\varepsilon}\sin^2\chi)}{I_{-1/4}(\hat{\varepsilon}\sin^2\chi_0)}\right] \tag{9.69}$$

where $C = \left\{1 - 2^{1/4}\Gamma(3/4)\left[(\hat{\varepsilon}\sin^2\chi_0)^{1/4}I_{-1/4}(\hat{\varepsilon}\sin^2\chi_0)\right]^{-1}\right\}^{-1}$.

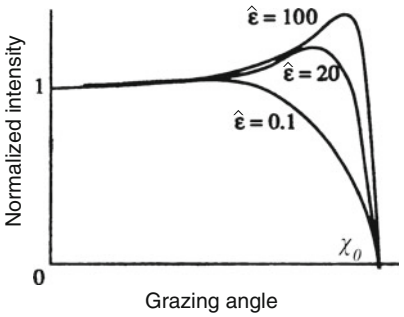


Fig. 9.12 Dependence of the noise intensity I on the angle from horizontal (grazing angle). Values of the parameter $\hat{\varepsilon}$ are shown in the figure, and χ_0 corresponds to the total reflection angle

It is clear that the nature of the dependence on the angle χ is determined by the parameter $\hat{\varepsilon}$, including both the geometric parameters of the channel, the absorption coefficient γ and the diffusion coefficient, which takes effective account of the random inhomogeneities. Depending on the values of the parameters of the problem, the coefficient $\hat{\varepsilon}$ ranges from zero to infinity.

Let us consider the limiting cases. If $\hat{\varepsilon} \to 0$, which corresponds to the situation when absorption is small in comparison with scattering (diffusion of the modes), then

$$F(\chi) = \begin{cases} \cos^{-1}\chi\left(1 - \frac{\sin^4\chi}{\sin^4\chi_0}\right), & \chi > \chi_0 \\ 0, & \chi < \chi_0 \end{cases}. \tag{9.70}$$

The other limit, $\hat{\varepsilon} \to \infty$, corresponds to large absorption or a comparatively small effect from random inhomogeneities. Here the limiting case shows the behavior:

$$F(\chi) = \begin{cases} \cos^{-1}\chi, & \chi < \chi_0 \\ 0, & \chi > \chi_0 \end{cases}. \tag{9.71}$$

We note that the intensity in (9.71) is nonzero at $\chi = 0$, although the intensity of an individual dipole source tends to zero as $\chi \to 0$. This value is obtained as the result of an integration over the whole surface, taking into account the nature of the attenuation. In other model situations, the results can be different.

The general case of an arbitrary value of the parameter $\hat{\varepsilon}$ is found to have values between the above limiting cases. Figure 9.12 shows a number of curves for $F(\chi)$. The behavior corresponding to the parameter values $\hat{\varepsilon} = 0.1$ and $\hat{\varepsilon} = 100$ practically coincide with the upper and lower limiting cases. The qualitative nature of the dependence obtained in (9.69) corresponds to the numerical results obtained in Kuznetsova (1989): in the absence of inhomogeneities, there is a minimum in the region of small angles χ. This is due to the minimum in the angular distribution of the intensity of radiation of an individual source. In the presence of random inhomogeneities, there is a jump in the energy in the region of small grazing angles. As the concentration of random inhomogeneities increases (growth of the diffusion coefficient), the minimum becomes less pronounced and for sufficiently large values of a becomes a maximum.

9.5 Effect of the Sound Speed Profile on the Vertical Distribution of the Noise Field Intensity

Experimental data from measurements of the vertical distribution of the intensity of low frequency noise (Morris 1978; Lyubchenko and Petnikov 1991), together with theoretical studies of this question (Kuperman and Ingenito 1980; Weston 1980d; Klyachin and Kuryanov 1974) attest to the dependence of average values of the noise level on depth, associated with changes in the sound speed profile.

The best known measurements of the depth dependence of the noise intensity in deep water are given in Morris (1978). The data presented there have served as material for theoretical analysis in various works (Arase and Arase 1967; Weston 1980a, b, c, d). Based on an averaged approach in the framework of ray theory, the depth distribution of the intensity was considered in Weston 1980a, b, c, d, where it was shown that in the presence of a deep-water sound channel the "law of conjugate depths" is satisfied, that is the intensities are identical at depths with the same sound speed up to the so-called critical depth, that is the depth at which the sound speed is equal to the sound speed on the surface. The maximum value of the noise intensity occurs somewhat below the critical depth and the minimum value of the intensity occurs on the channel axis. These general laws are also described by theory (Arase and Arase 1967) and agree with calculated data (Tonoyan 1991). The spatial distribution of the noise intensity was considered in Klyachin and Kuryanov (1974) in the framework of the theory of radiation transport, where, in particular, agreement with Morris's experiment was also obtained.

The question of the connection between the depth dependence of the noise intensity and the sound speed profile in shallow water has been studied theoretically by Kuperman and Ingenito (1980) who considered three types of sound speed profile: constant, linear with a negative gradient, corresponding to a near-bottom sound channel, and surface duct. The curves of intensity against depth, in their nature, approximately reflect the sound speed profile in the waveguide, for the constant and linear depth dependence. As the frequency increases, the slope of the curves increases. For a near-surface waveguide, the nature of the dependence is somewhat different, though the maximum intensity value occurs at the surface, in the region with minimum sound speed.

Qualitatively, this behavior can be understood from the following arguments. Naturally, in a waveguide with constant sound speed, all the modes excited by sources on the surface have on average a constant amplitude with depth and thus the intensity distribution with depth will be approximately constant, apart from in areas near the bottom and the surface, where boundary conditions play a role. If the sound speed profile has a negative gradient, the situation is somewhat ambivalent. Only some of the modes (rays) are excited by the surface sources, namely only those which graze or reflect from the surface. Each of these modes has a smooth envelope (for example, in the WKB method, this envelope is proportional to $[\sigma(z)]^{-1/2}$, where $\sigma(z)$ is the transverse (vertical) wave number). This quantity decreases toward the bottom for all modes, leading to a decrease in the intensity

on summation. Near-bottom modes, which might contribute near the seabed, are not excited by surface sources. However, a relative increase in the noise intensity near the bottom and even the occurrence of a maximum there may arise in the presence of a mechanism for the delivery of energy from near-surface excited high modes to low modes, for example, scattering on random inhomogeneities. For a near-surface channel, the clear maximum intensity value on the surface is due to the presence of near-surface modes (rays) being strongly excited by surface sources.

Let us now consider experimental results from measurement of the vertical profile of the noise field intensity in shallow-water conditions, using an example from the Barents Sea. A detailed description of the experiment we consider is given in Lyubchenko and Petnikov (1991). The depth dependence of the intensity of the noise field was obtained on a vertical drifting array of 40 m length, comprised of 11 equidistantly positioned receivers (hydrophones). The upper end of the array was positioned at a depth of 40 m. As one moved horizontally from the point of observation, the sound speed profile changed somewhat, exhibiting a close to linear dependence, and indicating some local oceanographic feature. The slope of the bottom in the given region was insignificant, amounting to less than 0.007°. The wind speed at the time of measurement was 7.6 m/s. The rate of drift of the antenna was a rather brisk 2 knots, which led to a deviation of the antenna from the vertical by approximately 30°. Measurement of the average slope angle of the array was performed using two depth sensors located at the ends of the receiving antenna. Figure 9.13 shows graphs of the vertical distributions of the average values of the noise level and the corresponding sound speed distribution with drift. The noise field was averaged in the 200–600 Hz band. The averaging time for acquisition of the average spectral noise levels was 20 min, and the number of signals averaged was 60.

Taking into account the inclination of the antenna, comparison of the vertical sound speed profiles and profiles of the average noise intensity shows that minimum values of the sound speed profile correspond to minimum values in the vertical distribution of the noise. Similar measurements of the vertical distribution of the average intensity values of the noise field were performed in shallow-water regions, where there is no underwater sound channel and the sound speed profile is characterized by a small negative gradient. In this case, within experimental error, the vertical distribution of the intensity of the noise was monotonic and practically constant.

To allow some interpretation of the experimental dependence of the noise intensity, let us consider a model of a regular (constant depth) waveguide with the model sound speed profile shown in Fig. 9.14.

$$c(z) = \begin{cases} c = 1,480\,\text{m/s}, & 0<z<h, \\ c' = 1,445\,\text{m/s}, & h<z<h+d, \\ c'' = 1,450\,\text{m/s}, & h+d<z<H. \end{cases} \tag{9.72}$$

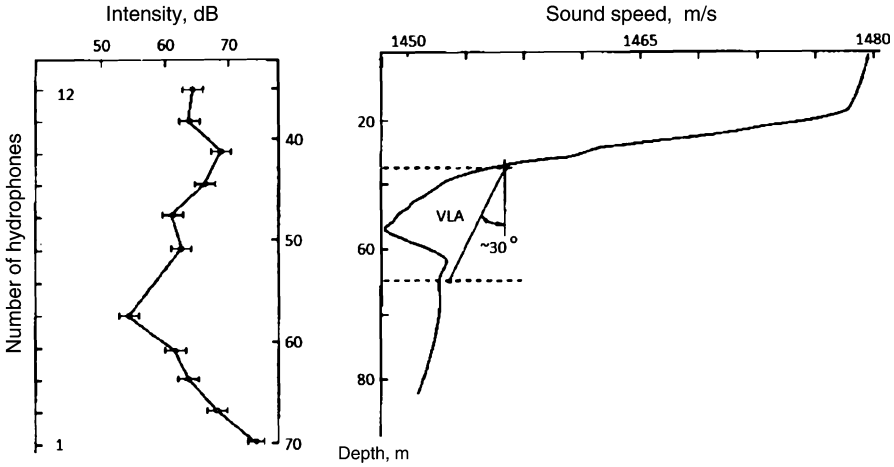


Fig. 9.13 Sound speed profile in the region of the experiment of Lyubchenko and Petnikov (1991), (the right panel) and corresponding measured distribution of the noise intensity, averaged in the 200–600 Hz band (the left panel) *VLA* vertical line array. *Left panel* shows part of water layer denoted by dashed lines in the *right panel*

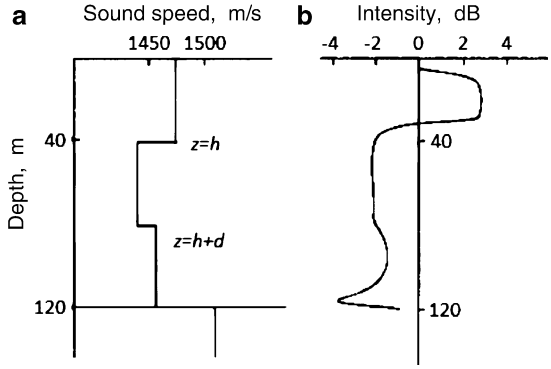


Fig. 9.14 Model sound speed profile (**a**) and the corresponding intensity distribution, averaged in the 200–600 Hz band

We further suppose that the water column with this sound speed profile overlies a fluid, homogeneous, absorbing half-space with geoacoustic parameters c_1, ρ_1, α. To calculate the intensity, we use (9.47) together with a summation of the modal intensities. In this case, the eigenfunctions have the form:

$$\Psi(z) = \begin{cases} \sin \sigma z, \quad 0<z<h, \\ B \sin \sigma'(z-h) + C \cos \sigma'(z-h), \quad h<z<h+d, \\ G \sin \sigma''(z-h-d) + E \cos \sigma''(z-h-d), \quad h+d<z<H, \\ V \exp[-\kappa(z-H)], \quad z>H. \end{cases} \tag{9.73}$$

where

$$B = \frac{\sigma}{\sigma'}\cos\sigma h, C = \sin\sigma h, G = \frac{\sigma'}{\sigma''}(B\cos\sigma' d - C\sin\sigma' d),$$
$$E = B\sin\sigma' d + C\cos\sigma' d, V = G\sin\sigma''(H-h-d) + E\cos\sigma''(H-h-d).$$

σ, σ', σ'' are the transverse (vertical) wavenumbers in the corresponding regions of depth of the waveguide: $\xi = \sqrt{k^2 - \sigma^2} = \sqrt{k'^2 - \sigma'^2} = \sqrt{k''^2 - \sigma''^2}$, and the normalization coefficient is found from condition (3.33).

The eigenvalues are found from the transcendental equation obtained from the continuity condition

$$\tan\sigma''(H-h-d) + \frac{\frac{m\sigma''}{\kappa}R + \frac{\sigma''}{\sigma'}T}{R - \frac{\sigma''}{\sigma'}T} = 0, \tag{9.74}$$

where $R = \frac{\sigma}{\sigma'} - \tan\sigma h \times \tan\sigma' d$ and $T = \frac{\sigma}{\sigma'}\mathrm{tg}\sigma' d + \mathrm{tg}\sigma h$.

Based on this model, numerical calculations of the depth dependence of the noise field intensity, averaged over frequency in the band 200–600 Hz were performed. The dependence $I(z)$, averaged in the band 200–600 Hz, for a chosen sound speed profile and bottom parameters $c_1 = 1,525$ m/s, $\rho_1 = 1.7$ g/cm^3, $\alpha = 0.015$ is shown in Fig. 9.14. Clearly, the intensity curve approximately restates the sound speed profile except very close to the bottom, as previously noted. A qualitative explanation of the nature of this behavior was given earlier. It is interesting to note that scattering by random inhomogeneities, which may lead to the redistribution of the intensity across the modes, and thus, by depth, does not (and cannot!) occur in this case. Almost "by omission," this stresses the effects of scattering by surface waves. In particular, the low order, near-bottom modes which play a significant role in the redistribution of the field are not excited on the surface and, hence, do not interact with high-order, near-surface modes (which are strongly excited).

Let us further consider how the change in sound speed profile with depth affects the averaged noise intensity for a negative (overall) gradient. We shall compute the intensity of the noise in a model waveguide with a sound speed profile represented by a two-layer function (Fig. 9.15a). This figure shows the vertically averaged distribution of the intensity (Fig. 9.15b) and also the nonaveraged dependence of the intensity at frequency $f = 300$ Hz (Fig. 9.15c). As we see in Fig. 9.15, the jump in the sound speed $\Delta c = c - c'$ at depth $z = h$ corresponds closely to a change in the averaged intensity ΔI (In our case $\Delta c \sim 30$ m/s at depth 40 m, and the corresponding jump is $\Delta I \sim 6$ dB). Let us determine the value of this change as the difference between the maximum value $I_{\max}$ above the level $z = h$ and the minimum value below that level. The quantities $I_{\max}$ and $I_{\min}$ are determined far from the waveguide boundary, and $\Delta I = I_{\max} - I_{\min}$. The dependence of ΔI on the jump in the sound speed Δc is shown in Fig. 9.16 for two different values of the sound speed in the bottom. Clearly, the change in the noise field intensity at

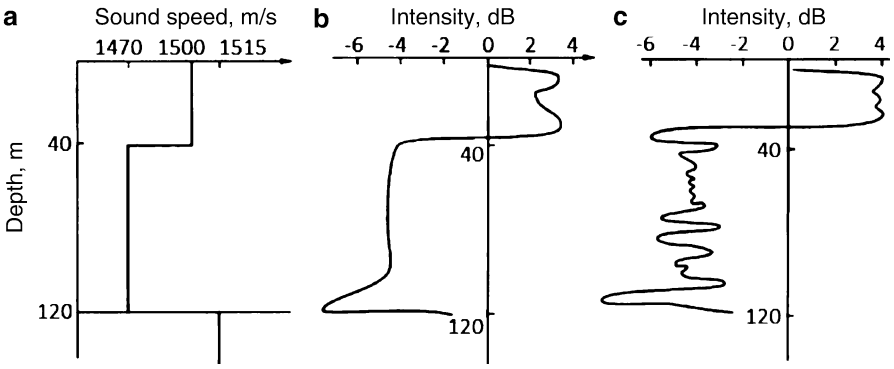


Fig. 9.15 Model sound speed profile (**a**) and intensity distribution at 300 Hz (**c**). *Middle panel* is averaged in the band 200–600 Hz (**b**)

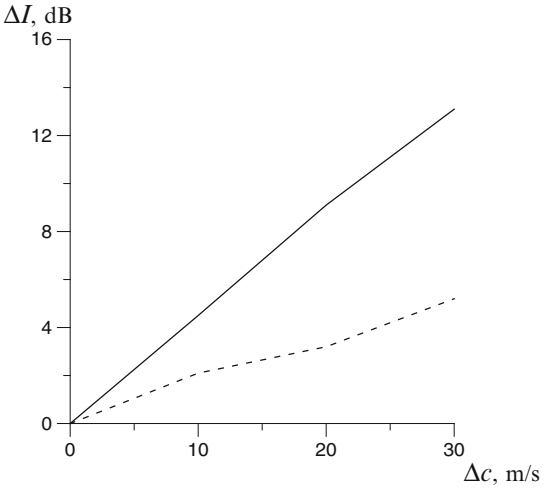


Fig. 9.16 Dependence of the jump in the averaged noise intensity on the decrease in sound speed. Dashed curve corresponds to $c_1 = 1,515$ m/s, continuous curve to $c_1 = 1,505$ m/s

the level corresponding to the jump in the sound speed will be greater the less the sound speed in the bottom differs from the sound speed at the surface of the waveguide. The parameters in the calculation are the following: $c = 1{,}500$ m/s, the continuous curve corresponds to $c_1 = 1{,}505$ m/s, and the dashed curve to $c_1 = 1{,}515$ m/s.

Thus, results from the study of noise given in this chapter demonstrate the possibility of using measurements of noise characteristics to determine bottom geoacoustic parameters in shallow water what was described in our earlier chapters on inverse theory.

Chapter 10
Equipment for Shallow Water Acoustics and Experimental Considerations

10.1 The Frequency Band Used for Large-Scale Acoustic Monitoring

In this chapter, we consider the applications of sound signals with frequencies less than 500 Hz. The use of such low frequencies is primarily motivated by the strong attenuation of acoustic waves in shallow water due to the interaction with the bottom. For example, for a negative gradient sound speed profile, the transmission loss of acoustic signals (TL) where $\mathrm{TL} = 20\lg(\bar{p}/p_0)$ can be described in the following equation by the three terms inside square brackets, each corresponding to different decay laws for the sound field with distance

$$\mathrm{TL} = [20\lg(r_0/r_1)] + [15\lg(r_1/r^*) - \beta_1(r^* - r_1)] + [10\lg(r^*/r) - \beta(r - r^*)], \tag{10.1}$$

where $\bar{p}$ is the mean acoustic pressure at the receiver, defined in terms of the mean intensity (4.95) of the sound field[1] $\left(\bar{p}(r) = \sqrt{\bar{I}\rho c}\right)$, and p_0 is the acoustic pressure at distance $r_0 = 1$ m from the sound source:

$$\left(p_0 = \sqrt{I\rho c} = \sqrt{\frac{W_0\rho c}{4\pi r_0^2}}\right). \tag{10.2}$$

The first term corresponds to the spherical spreading of the wave front at the start of the dissipation (3.3), the second to the so-called "three over two law" with additional damping, and the third to cylindrical spreading with additional damping.

[1] In experiments, $\bar{p}$ is defined as $\bar{p} = \langle |p(\vec{R}, t)| \rangle$, where the angle brackets denote averaging over a spatial interval greater than the period of the interference pulsations, and over a time greater than the period of oscillation.

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_10,
© Springer Science+Business Media, LLC 2012

The expression (10.1) is strictly empirical. However, as shown in Sect. 4.6, the second term corresponds to the situation when a large number of modes are present in the waveguide propagation, whereas the third term corresponds to a number of near-bottom modes with the same attenuation coefficient [here, the coefficient β is given by $\beta = 10^4 u \lg e$ dB/km, where u is defined in (4.117)]. The distances r_1 and r^* define the dissipation interval $[r_1$ to $r^*]$, over which the "three over two" law is satisfied. The distance r_1 (in oceanic acoustics it is sometimes called the transitional distance) can be estimated using the simple relation $r_1 \cong H^2/2\lambda$, i.e., the distance where the Fresnel radius of a ray is equal to half the depth of the waveguide (Kravtsov and Orlov 1990). For a waveguide of 100 m depth, the transitional distance is equal to 333 m at 100 Hz (wavelength 15 m) and 1,000 m at 300 Hz (wavelength 5 m). The quantities r^*; β_1 and β depend on the wavelength of the sound and the waveguide parameters, including the acoustic properties of the bottom. For example, for one region with a negative gradient sound speed profile (Bunkin et al. 1984), these quantities were found equal to $r^* = 23$ km, $\beta_1 = 0.8$ dB/km, and $\beta =$ dB/km, for a waveguide depth of 100 m and a frequency of 100 Hz. At 300 Hz, these same parameters were equal to $r^* = 13$ km, $\beta_1 = 1.3$ dB/km, and $\beta = 0.38$ dB/km. These numbers show that the transmission loss increases sharply with frequency. For a source level of $p_0 = 190$ dB[2] relative to 1 μPa, the sensing range for the waveguide studied in the work of Bunkin et al. (1984) was $r = 235$ km at 100 Hz and $r = 185$ km at 300 Hz (estimates of r obtained using (10.1) assume that the signal-to-noise ratio (SNR) at the receiver is at least 20 dB, and that the accumulation time is equal to 1 s).

These estimates demonstrate the orders of magnitude for the range of acoustic sensing in shallow water. These limiting ranges are significantly less than in deep water, where experiments on the ultra long-range propagation of sound at distances of several thousand kilometers have been performed (Munk et al. 1994). On the other hand, problems in acoustic monitoring in shallow water usually only require relatively small sensing distances.

We once again stress that the significant growth of the attenuation with frequency means that the use of high frequencies for acoustic sensing of shallow water at large distances holds little or no promise, despite the possibility of increasing the source power with frequency. In fact, in the above example, to (theoretically) obtain a sensing range of 235 km at only 300 Hz requires a source level of 209 dB ($W_0 \cong 7$ kW), a considerable increase.

[2] The emission level of 190 dB corresponds to the acoustic power ($W_0 = 100$ W) of typical nondirectional low-frequency sound sources, used in experiments on acoustic tomography of the ocean.

10.2 Low-Frequency Sources of Sound

A wide variety of low-frequency sound emitters are presently used for acoustics purposes on the sea shelf. These sources operate using different physical principles and can differ significantly in their basic parameters, i.e., the operational frequency band Δf, the emitted acoustic power W_0, the weight, and the dimensions. In the case of self-contained sources, the basic parameters also include the source efficiency (the acousto-electric efficiency factor $\eta_{a/e} = W_0/W_e$, where W_e is the active component of the electric power taken by the source from the power supply) and the time of autonomous operation the source is capable of. Description of all possible low-frequency sound source designs goes far beyond the framework of this book. Detailed information on this topic can be found, for example, in the work by Rimsky-Korsakov et al. (1984). Here it should be mentioned as well the fundamental monograph by Sherman and Butler (2007) where the general description of linear electroacoustic transducers is given. Different electroacoustic transducers (piezoelectric, electrostrictive, magnetostrictive, electrostatic, variable reluctance, and moving coil) can work as both sound source and sound receiver, and in this monograph the interested reader can find the structural variations of the high-frequency sound sources and sound receivers including receiving and emitting high-frequency arrays based on these transducers. However, as noted in this monograph: "long range active sonar surveillance presents the greatest challenge to transducer development because of the need for high power at low frequency." In this connection we describe just a few basic low-frequency source designs implementing different operational principles (including designs which do not use electroacoustic transducers), and give their typical characteristics.

In discussing powerful low-frequency radiators of sound, we first examine the sound sources with electromagnetic transducers described by Bogolyubov et al. (2000) and Penkin (2001). The traditional design of such sources and their appearance is shown in Fig. 10.1. In simple terms, elastic metal diaphragms are located at the butt ends of a cylinder, and diaphragm vibrations in opposite directions are excited by an electromagnet. The preset radiation frequency f (in the range of 50–500 Hz), the operational frequency band, and the radiation power are determined by the choice of the dimensions and rigidity of the diaphragm, which is manufactured from special alloys that insure its continuous operation over many months. There are also designs (Penkin 2001), where the diaphragms are made of relatively cheap varieties of steel, and additional strength and rigidity of the emitting elements are provided by employing flat springs attached to the external surfaces of the diaphragms. Typical dimensions of sound sources with electromagnetic transducers and radiating diaphragms are on the order of 0.4–0.8 m in diameter and 0.25–0.5 m in thickness.

Radiators employing emitting pistons attached to a rigid case by elastic gaskets also belong to this type of sound sources. The design of such a source (Bogolyubov et al. 2000), used in the Acoustic Thermometry of Ocean Climate (ATOC) Program, is shown in Fig. 10.2. The diameter of the emitting piston of this source is 1.57 m,

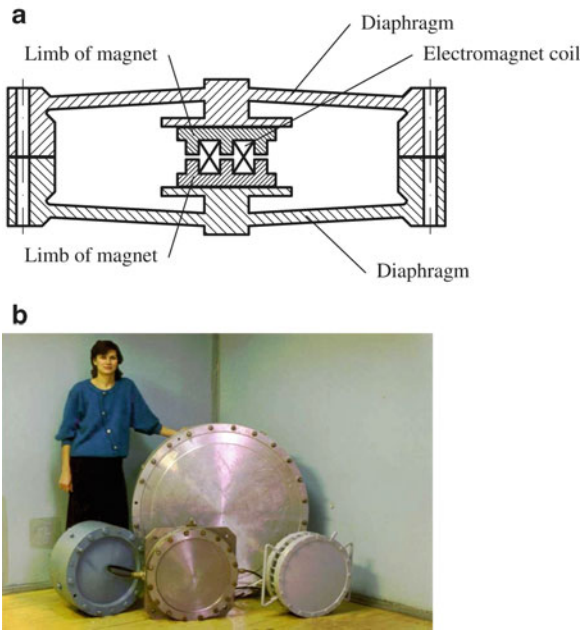


Fig. 10.1 Sound sources with an electromagnetic transducer and with a radiating diaphragm (**a**) schematic sketch, (**b**) appearance

the total source diameter is 2.64 m, and the thickness is 1.52 m. The case of emitting pistons instead of diaphragms provides one the opportunity (if desired) to reduce the operational frequency down to $f \cong 20$ Hz and to expand (because of increasing the effective area of the radiating surface) the operational frequency band.

We note that such sources have, as a rule, air compensators offsetting the large hydrostatic pressure encountered when operating at depths of several hundred meters. When operating at a small constant depth, such sources are usually just pumped up before an experiment to a calculated pressure, without using compensators.

Typical parameters of such electromagnetic-type sound sources[3] are given in Table 10.1 (Bogolyubov et al. 2000).

Sound sources with electrodynamic vibroexciters (linear electromotors or just the simple electrodynamic loudspeaker commonly used for the reproduction of music and speech) belong to another type of radiators. A schematic of such a vibroexciter is shown in Fig. 10.3a (Rimsky-Korsakov et al. 1984). In the diagram we see that (1) a bias coil produces a uniform constant magnetic field in the annular gap between the central rod and a round orifice in the upper flat part of (2) a magnetic core, then (3) a movable coil suspended with the help of (4) centering springs oscillates in the annular

[3] The parameters of the sound source used in the ATOC Program are given in the third line of the table.

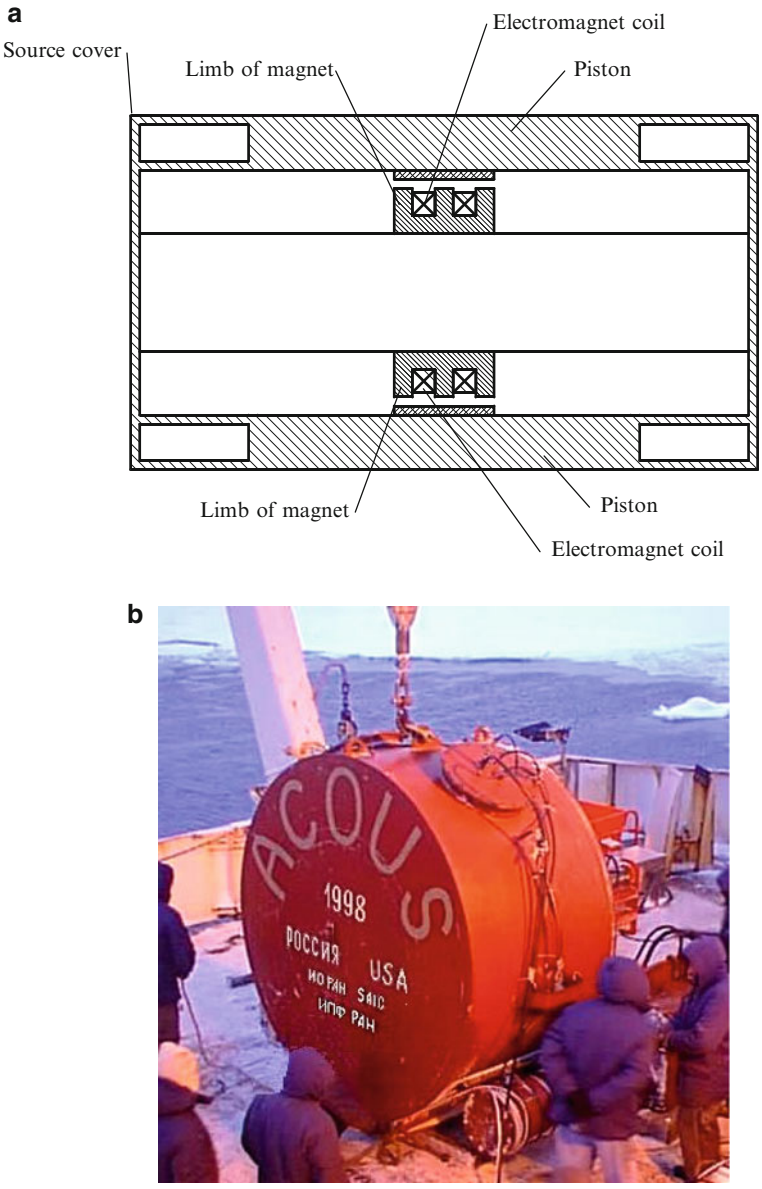


Fig. 10.2 Sound sources with an electromagnetic transducer and with a radiating piston. (**a**) Schematic sketch, (**b**) appearance

gap along it. Vibrations from (5) the platform attached rigidly to the moving coil are transmitted to a mechanical load (a radiating piston). To obtain high efficiency, a sound source with such vibroexciter must have a radiating piston with linear dimensions larger than the wavelength of the emitted sound. Therefore, at low

Table 10.1 Parameters of electromagnetic-type sound sources

Frequency (Hz)	Power (kW)	Frequency range (%)	Efficiency (%)	Mass (kg)
20	0.25	8	30	500
20	1	16	70	4,500
70	3	20	60	490
135	2	25	60	300
250	10	25	80	350
250	1	20	82	90
330	0.05	10	60	30
110	0.3	15	50	110
300	0.3	20	50	40
600	0.3	20	50	30

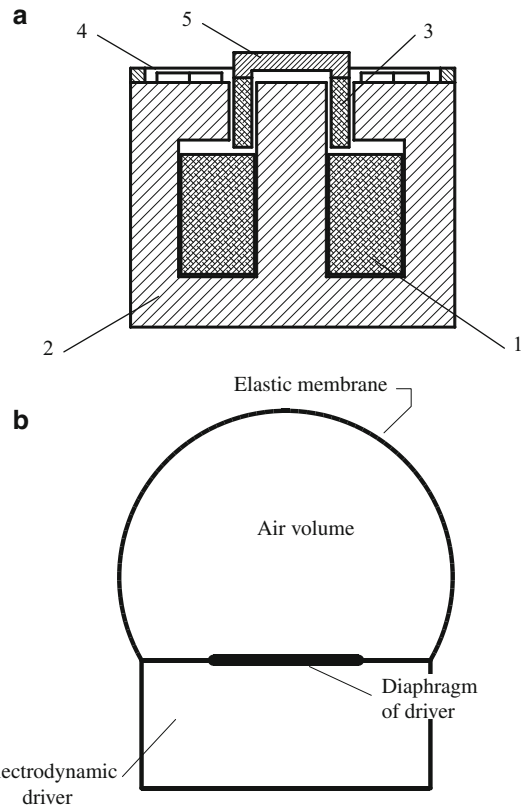


Fig. 10.3 Sound source with an electrodynamic vibroexciter, which is loaded to a resonator in the form of an air chamber with a flexible shell. (**a**) Exciter design: (1) electromagnet coil, (2) limb of magnet, (3) movable coil, (4) springs, (5) platform connected with diaphragm. (**b**) Assembled sound source

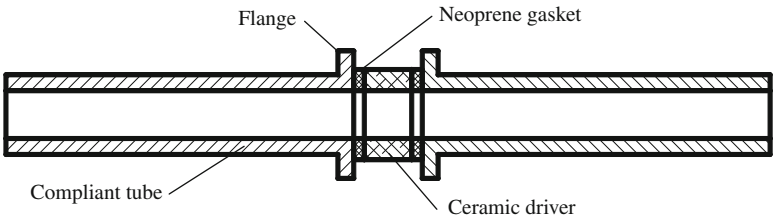


Fig. 10.4 Sound source with a cylindrical piezotransducer, which is coupled to a resonator in the form of a quarter-wavelength segment of tubes

frequencies an acoustic transformer is inserted between (5) the platform and the piston or a small piston is loaded with a resonator that is an air chamber with a flexible shell (Fig. 10.3b). Using such a resonator at frequency of $f \cong 47$ Hz, it is possible to obtain a radiated power of $W_0 \cong 6.5$ W with a source efficiency of 5% and a relative frequency bandwidth of about 15% (Sims 1960). The diameter of the flexible shell is 0.19 m in this case. We note here that an evident disadvantage of such radiators with an air chamber is the fact that the resonance frequency of the chamber depends on internal pressure and, therefore, on the submersion depth (as the system is pressure compensating).

Another common type sound source is the piezoceramic transducer variety. To increase the efficiency of such sources at low frequencies, a piezotransducer is often coupled to a passive acoustic resonator, which can be, for example, a quarter-wavelength section of tube as shown in Fig. 10.4. These sources described in the figure by Lee (1974) employ a driver in the form of a radially polarized piezoceramic ring with neoprene waterproofing. The tubes for the radiators (Lee 1974) were manufactured from common steel or aluminum and had a diameter of 8.9 cm. The principal resonance frequencies of the radiators were from 323 to 512 Hz, the tube lengths were from 76.2 to 45.7 cm, and their efficiency[4] varied from 10 to 30%. These sound sources can produce a rather high radiated power of several kilowatts, but their operational frequency band, determined by the high Q-factor of the resonator employed, is small $(\Delta f/f \leq 1\%)$.

Another sound source design employing piezoceramic transducers that is important from a practical point of view is that using the Helmholtz resonator and flexural piezotransducers (Henriques and Young 1980). The design of such sources is shown in Fig. 10.5. In Fig. 10.5a, the manufacture of a piezoactive transducer is shown in the form of a set of a large number of longitudinally or radially polarized segments which form the lateral surface of the resonator. In Fig. 10.5b, a transducer in the form of a piezoceramic flexural disk forms the resonator bottom. According to the

[4] It should be noted that in the case of sound sources with piezotransducers, the efficiency commonly taken is not the acousto-electric efficiency, but a somewhat smaller quantity $\chi_{a/e}$ determined as $\chi_{a/e} = W_0/i^2|z_e|$, where i is the current through the transducer and z_e is the total electric resistance including the active and reactive components.

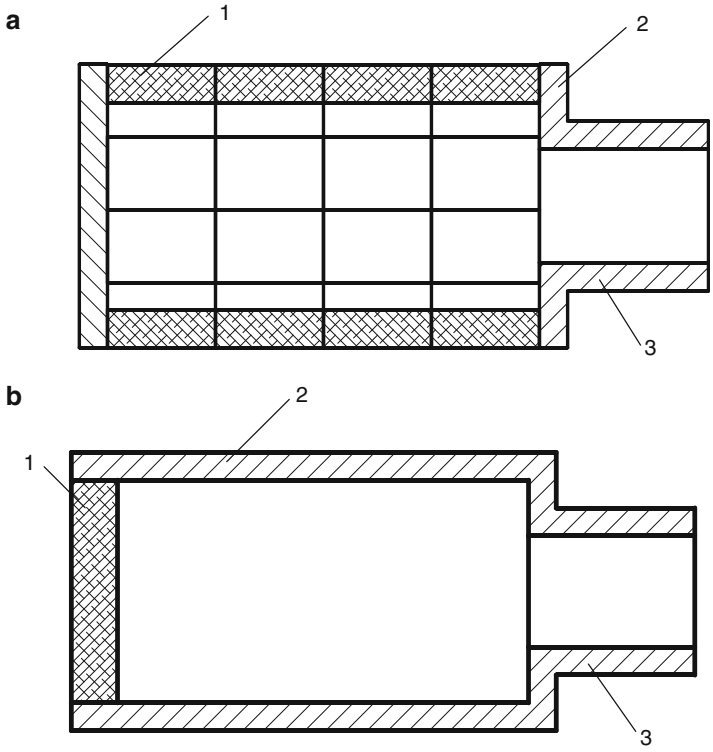


Fig. 10.5 Sound sources with Helmholtz resonators of various designs. (1) Piezotransducer, (2) case, and (3) mouth

estimates given by Henriques and Young (1980), the sound source shown in Fig. 10.5a can provide a radiated power of over 3 kW at $f = 300$ Hz, with an operational frequency band wider than that of sources using quarter-wavelength tubes $(\Delta f/f \leq 2\%)$ with smaller dimensions. (The diameter of the source is 0.5 m and the length is ≅90 cm.) Source efficiency in this case can attain 25%.

On the whole, the low-frequency sound sources with piezoceramic transducers described above operate effectively at a frequency of several hundreds of hertz, and have approximately the same power as the sources with electromagnets, which were described previously. However, the latter have a significantly wider operational frequency band and are also capable of operation at frequencies of only tens of hertz (Table 10.1). It is possible to increase the frequency band of the piezoceramic transducer sources by using a special resonator with tunable frequency; a source with such a resonator developed by Morozov and Webb (2003) is shown in Fig. 10.6. The distinguishing feature of this structure, when compared to the sound source shown in Fig. 10.4, is that the tubes have special slots in their lateral faces and coaxial sleeves that move and cover (or uncover) the slots. The tube resonance frequency varies according to the size of the slot opening. In doing this, the variation

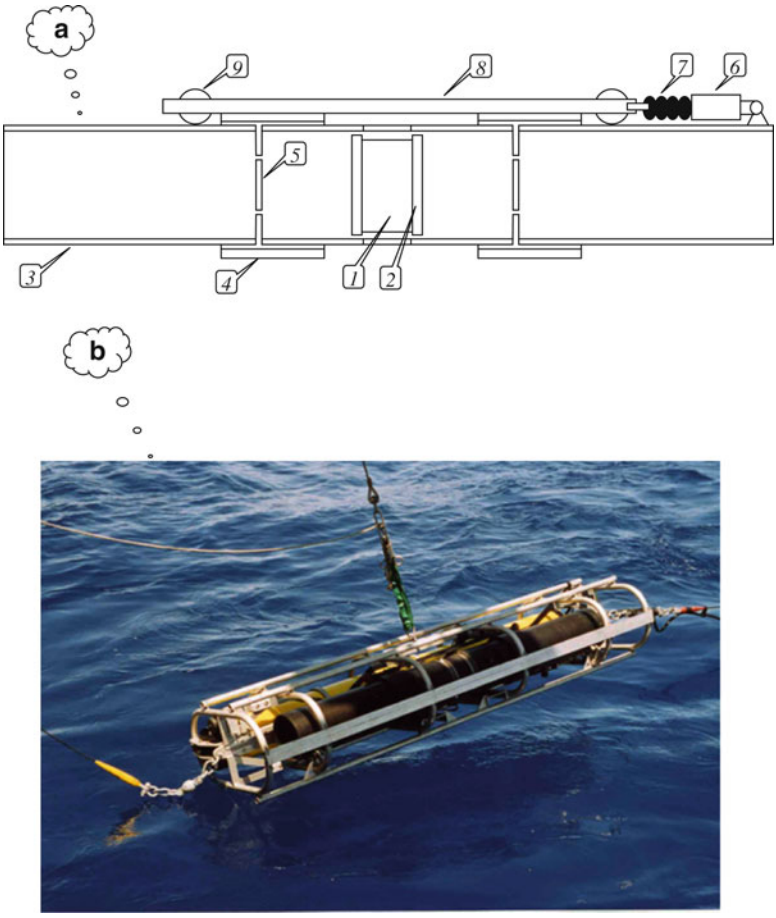


Fig. 10.6 Resonator tube tunable projector (one wheel, pair 9 of three is shown). (**a**) sketch: (1) a symmetrical Tonpilz transducer, driver with two (2) pistons separated by preloaded ceramic stacks; (3) resonator tube; (4) coaxial sleeves; (5) orifices in the form of slots; (6) electrical linear actuator, moving metal (8) bar with (9) wheels, along the resonator (3) pipe; (6) bar is fastened with the (4) sleeves; (7) rubber oil-filled bellows. (**b**) appearance

is executed synchronously with the frequency of the incoming electric signal, using a phase-locked loop system. Notice also that in this structure, a symmetrical Tonpilz transducer is used instead of a radially polarized piezoceramic ring. A low-frequency sound source with this structure has been successful operated in the frequency band of 185–350 Hz (Morozov and Webb 2003). The radiated power and efficiency were equal to $W_0 \cong 420$ W and $\cong 50\%$, respectively. The source mass, including the mass of the battery for remote operation, was 500 kg.

All the sound sources described above can operate in both pulsed and CW modes. They can emit signals with the preset amplitude and phase modulation that are needed for acoustical experiments in shallow water. However, independent

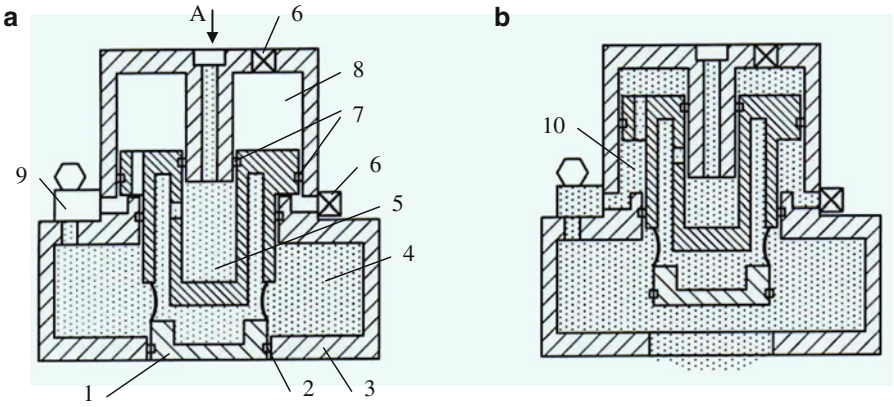


Fig. 10.7 Scheme of an air gun (**a**) before exhaust and (**b**) at the moment of exhaust. (1) Piston, (2) Teflon gasket ring, (3) case, (4) basic volume of the air storage chamber, (5) central cavity of the piston, (6) valve, (7) rubber gasket rings, (8) damping volume, (9) firing valve, and (10) volume of the chamber under the piston shoulder

of their design, they are inherently narrow-band and rather expensive radiators. To remedy this, air guns or explosive sound sources are used frequently in shallow water acoustics, to emit powerful wideband pulsed signals.

One possible design for an air gun is displayed in Fig. 10.7 (Rimsky-Korsakov et al. 1984). In this case, compressed air with pressure p_0 comes through the aperture A, and under the effect of this pressure a piston goes down and closes the air port. Orifices in the piston allow the compressed air to fill in the storage chamber. Further, with the help of an electromagnetic firing valve, a channel is opened which lets compressed air enter the chamber of larger diameter under the piston. Since the piston area now affected by compressed air in the chamber (10) is greater than that in the other chamber (5), the piston goes upward, and near its upper position it is slowed down, compressing air in the volume (6). (The valves are needed to let out excess air.) During the time of upward piston motion, the compressed air goes sharply out of the chamber through the now-open large-diameter air port and produces a wideband pulse of acoustic pressure. For this system, the energy maximum in the pulse spectrum is concentrated at the frequency f (Barger and Hamblen 1980), given by

$$f = 1.2 p_0^{-0.06} \rho^{-1/2} p_w^{0.91} E^{-1/3}, \tag{10.3}$$

where p_w is the pressure of the ambient sea-water, E the internal energy of air in the main chamber (4), given by $E = p_i V_i (\gamma_a - 1)^{-1}$, V_i the volume of chamber (4), and γ_a the ratio the thermal capacities of air at constant pressure and constant volume. As seen from (10.3), the radiation frequency increases as the source depth increases.

Another design for air gun sound sources is also well known (Barger and Hamblen 1980). In this case the air storage chamber is shaped as an elongated cylinder with a large number of air ports at its lateral surface. Each aperture has an individual valve, but all of them open simultaneously with the help of a solenoid,

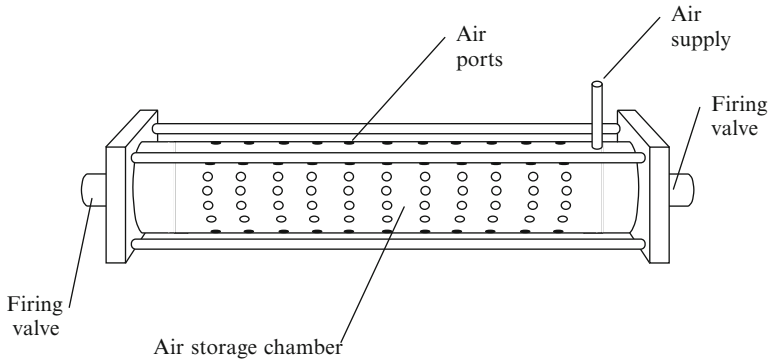


Fig. 10.8 Air gun with a cylindrical storage chamber

providing a compressed air exhaust (Fig. 10.8). For this, design the frequency f is not determined by (10.3) anymore, but rather depends on the ratio of the accumulated energy to the cylinder length.

The shape of a sound pulse emitted by an air gun with an air storage chamber shaped like a cylinder is shown in Fig. 10.9 as an example. Its spectrum is given there also. As one can see from the figure, the relative width of the signal spectrum exceeds 30% at a frequency of $f \cong 120$ Hz. On the whole, the operational frequency range of air guns lies within the interval 10–200 Hz.

In distinction to the sound sources considered above that operate in CW mode, it is convenient to characterize sources of pulsed signals not by their radiated power W_0 but by their energy spectral density G_f. In the particular case of air guns, the maximum values of energy spectral density at frequency f are about $G_f \cong 220$ dB with respect to 1 $\mu Pa^2/Hz$ for a volume $V_i = 0.02 - -0.016\,m^3$ and pressure $p_0 = 1.4 - 3.5 \times 10^6 Pa$. According to (10.2), this corresponds to a radiated power of an equivalent narrowband source of $W_0 = 84$ kW. For sources with a spherical air chamber, the value of S_f decreases 10–15 dB with frequency growth from 20 to 120 Hz. In the case of radiators with a cylindrical chamber this decrease is not observed (Barger and Hamblen 1980). We should note that there is a good reproducibility of signals from such sources. For example, for a series of 120 pulses measured in the course of the same study, the root-mean-square deviation the energy of spectral density at a given frequency f was just 0.15 dB.

The efficiency of air guns, which is the ratio of the radiated acoustic energy to the total energy stored in the air chamber, varies over a rather wide range and depends on many factors: the chamber dimensions, its shape, the frequency f, the submersion depth, etc. For example, for the case of a source with a spherical chamber of volume $V_i = 0.016\,m^3$ at pressure $P_i = 2.1 \times 10^6$ Pa and at a submersion depth of ≅61 m ($f = 20$ Hz), the efficiency is 20%. If the submersion depth increases to 427 m ($f = 100$ Hz), the efficiency drops down to a meager 2%. At the same time, the efficiency of the source with a cylindrical chamber of volume $V_i = 0.02\,m^3$ is 5% and does not depend on the submersion depth. The physical

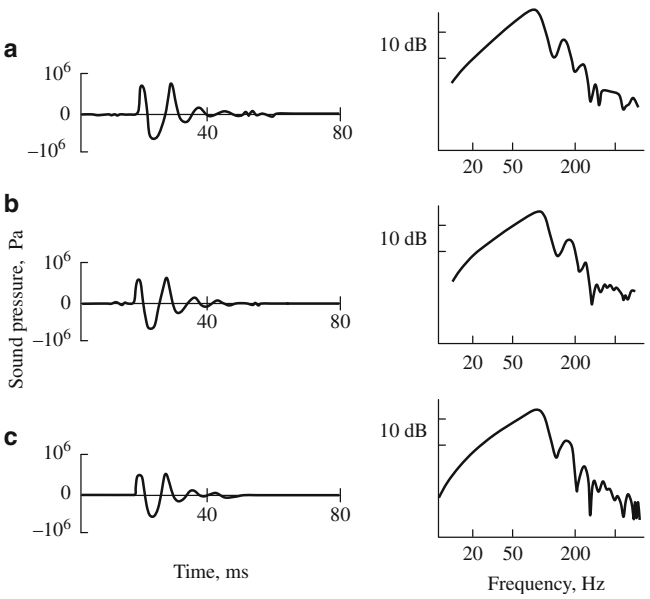


Fig. 10.9 Waveforms (left panel) and their energy spectra (right panel) for a cylindrical-cavity air gun with initial energy of 1.7 MJ. (**a**) Depth of 411 m, (**b**) depth of 442 m, and (**c**) depth of 472 m

reason for this difference is the different efficiency of the air blowout through the ports. For the case of a spherical chamber, as the frequency grows, not all the accumulated air has enough time to leave the chamber through a single large port when the chamber is open. The many small ports in the cylindrical chamber turn out to be more efficient for air exhaust at frequencies over 100 Hz.

Turning to explosive sources of sound, in this case the generation of acoustic waves is connected with a shock wave and the subsequent pulses of a gas bubble (Fig. 10.10). The time dependence of the pressure of the shock wave pulse is described by a simple function (Urick 1975):

$$p(t) = p_0 e^{-\frac{t}{t_0}}, \tag{10.4}$$

where, because of the considerable contribution of the nonlinearity of the medium at close ranges, p_0 and t_0 depend upon distance[5]:

$$p_0 = 5.15 \times 10^7 \left(\frac{w_0^{\frac{1}{3}}}{r}\right)^{1.13} \text{[pa]}, \tag{10.5}$$

[5] For typical explosive sound sources (where the mass of the explosive material is a few hundred grams), this dependence occurs over a distance of several hundred meters. For large distances p_0 and t_0 are time independent and the equation of linear acoustics holds.

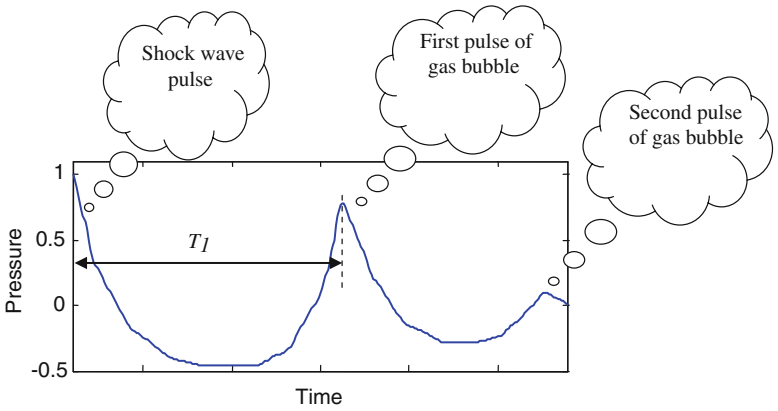


Fig. 10.10 Pressure waveform developed from a charge explosion

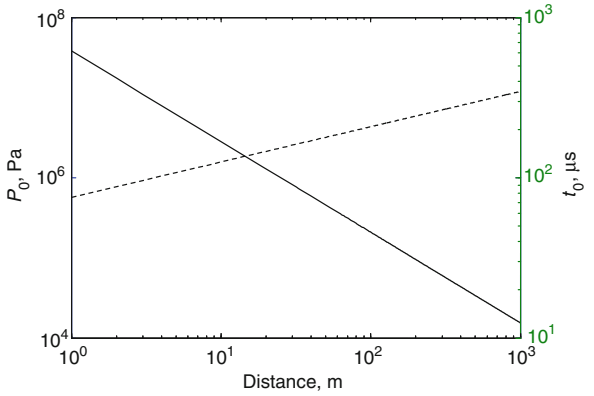


Fig. 10.11 Dependence of peak pressure P_0 and time constant t_0 on distance in meters

$$t_0 = 93 \times w_0^{\frac{1}{3}} \left(\frac{w_0^{\frac{1}{3}}}{r} \right)^{-0.22} \quad [\,\mu\text{s}], \tag{10.6}$$

where w_0 is the mass of the charge in kilograms. For a charge mass of $w_0 = 0.45$ kg, the dependences (10.5) and (10.6) are shown in Fig. 10.11.

The sound pressure of the first gas bubble pulse depends on the depth of the explosion, and is comparable in value with the pressure of the shock wave. The pressure of the second pulse is approximately a factor of 5 less, and subsequent pulses are negligibly small.

The characteristics of explosive sound sources have been described in detail, for example, by Urick (1975) and Weston (1965). Here we present only a few properties

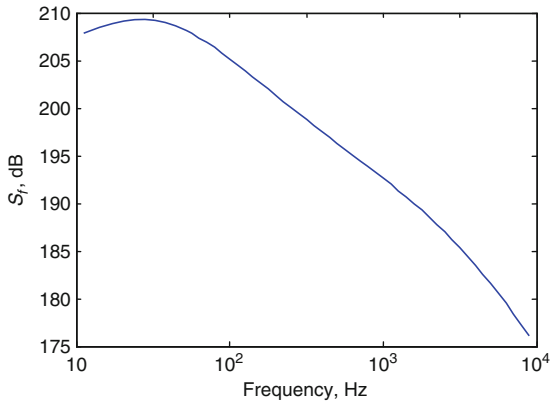


Fig. 10.12 Energy spectral density of a signal from an explosion (Weston 1963) with an explosive mass of 0.45 kg and an explosion depth of 36 m

of these sound sources, mentioned in the study of Vadov (1994), which are significant for acoustic sensing in shallow water at large distances.

- The spectrum of the emitted signal, up to a frequency of 3–5 kHz, is characterized by strong irregularity, associated with the interference between the shock wave and the gas bubble pulses. At large (greater than 10 km) distances, the irregularity is mainly determined by the first bubble pulse. The position of the first maxima on the frequency scale, f_0, and the distance between successive minimum in the spectrum are related to the period of the first bubble pulse T_1 by the equation: $f_0 T_1 = \Delta f T_1 = 1$. T_1 can be estimated from a formula due to Arons (1948):

$$T_1 = 0.48 \frac{K_0 w_0^{1/3}}{(z+10)^{5/6}}, \tag{10.7}$$

  where K_0 is a coefficient of proportionality (the value of K_0 is weakly dependent upon the type of the explosive material, and for TNT is equal to 4.36), and z is the depth of the explosion in meters. (We note that in the experiment of Vadov (1994), the period of pulsation varied according to the law $T_1 \sim 1/z^{1.1}$ i.e., it depended more strongly on depth than in (10.7)).
- The decay of the spectral density of the energy of the explosive signal with frequency (based on the envelope, without taking the interference structure into account) follows a law close to $1/f$ at frequencies less than 1 kHz, as exemplified in Fig. 10.12.
- The repetition rate of the spectral characteristics (mean-square spread of the levels) of explosive signals for frequency bands bounded by consecutive interference minima is generally 0.2–0.9 dB.

It should be noted that the mass of the charges, typically used in experiments, does not exceed several hundred grams. It is not usual to increase the mass of the charge beyond this size for the following reasons (1) the low effectiveness of such an

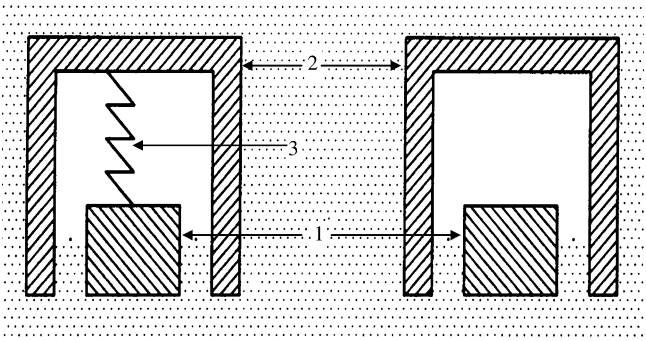


Fig. 10.13 A diagram of the source (**a**) mechanical spring version and (**b**) air spring version. (1) Oscillating rigid body, (2) chamber, and (3) spring

increase (level of the source level goes as $\sim w_0^{\frac{1}{3}}$), (2) the possibility of destroying or injuring marine organisms; (3) a significant increase in the cost of the individual charge; and (4) the complexity and danger of working with large charges.

A general feature of all the sound sources described above which operate at low frequencies is the fact that their linear dimensions are much smaller than the sound wavelength so that all of them are the so-called zeroth-order radiators or monopoles. The sound waves emitted by them are spherically symmetrical. In this context, it is appropriate to at least mention a one more original design for a low-frequency sound source, where the dipole oscillations of a solid are converted into the monopole radiation of the source as a whole (Donskoy and Blue 1994). One design for such a source is presented in Fig. 10.13. Here a solid (1) with positive buoyancy performs oscillations under the effect of an internal vibrator (an electric motor with an unbalanced mass). The solid is suspended at a spring (2) inside a specially designed chamber (3) with the center of gravity at the chamber mouth. This secures its vertical position with the bottom up and, correspondingly, maintains the presence of air in it. The presence of the spring (2) is not necessary in general and its role can be performed by the air layer itself. This source belongs to the class of narrowband sources of sound, which are capable of operation in both pulsed and CW modes. The operational (resonance) radiation frequency is equal to $\omega_r = 2\pi f = \sqrt{\omega_1^2 + \omega_2^2}$, where $\omega_{1,2}^2 = k_s/\tilde{M}_{1,2}$, k_s is the spring constant or the equivalent stiffness of the air layer,[6] $\tilde{M}_1$ is the mass of the oscillating solid, and $\tilde{M}_2$ is the mass of the chamber with the added mass of water. According to the estimate given by Donskoy and Blue (1994), this sound source can provide radiated power of $W_0 = 4\,\text{kW}$ at a frequency of $f = 70\,\text{Hz}$. The operational frequency band in this case is $\Delta f = 12\,\text{Hz}$.

To provide directional radiation at low frequencies, an array of several sources must be used, positioned at half-wavelength distances from each other. In other words,

[6] It is necessary to note that in the absence of a spring the radiation frequency depends on air pressure in the chamber and, therefore, the depth of the source submersion.

an extended linear array is formed for this purpose. One of such vertical arrays consisting of 20 electromagnetic-type radiators is described in the study by Bogolyubov et al. (2000). Each radiator of the chain had the following parameters: resonance frequency $f = 330\,\text{Hz}$, radiation power $W_0 = 10\,\text{W}$, and mass $\tilde{M} = 35\,\text{kg}$. The formation of a preset directivity pattern for such an array is accomplished by controlling the relative phase of the sound emitted by each source.[7] On occasion, amplitude control is used also for improving the directivity pattern. However, in contrast to analogous linear receiving arrays, it is necessary to control the phase and amplitude also taking into account the interference of the sources. More details on this interference can be found in the work by Barger and Hamblen (1980) and Bogolyubov et al. (1986), for the cases of air guns and electromagnetic-type radiators, respectively.

We should note that it is important for a whole class of acoustic experiments in shallow water not to form a single-lobe directivity pattern for a radiating vertical linear array as is usually done in a free space, but rather to try to excite one of the modes of the shallow-water acoustic channel. In other words, the problem is now to implement single-mode radiation. This technique is needed, for example, for low-mode differential tomography of shallow water, which is based on the excitation of one of the channel modes and detection of the signal corresponding to another mode. Since the transformation of energy from one mode to another occurs only in an inhomogeneous waveguide, the measured coefficient of transformation provides a test for inhomogeneities in shallow water within the framework of certain models (Nechaev and Khilko 1987). We again note that in the case of single-mode radiation, the level of long-range bottom reverberation decreases dramatically (see Sect. 6 of this book).

To implement single-mode operation (for example, to radiate the mode with the number l), it is necessary that the amplitude and phase of the sound pressure emitted by the j-th sound source in the chain be directly proportional to the complex-conjugate eigenfunction of the l-th mode, $N_s\Psi_l^*(0, z_j)$. In this case, according to (3.25), the sound pressure field in a homogeneous channel at the point with coordinates (r, z), which lies at a large distance from the vertical chain, is equal to

$$\begin{aligned} P(r,z) &= \sqrt{N_s W_0}\exp(i\pi/4)\sum\nolimits_j^J \sqrt{\rho_j c_j}\Psi_l^*(0,z_j) \\ &\quad \sum\nolimits_m^M \Psi_m(0,z_j)\frac{\Psi_m(r,z)}{\sqrt{\xi_m(r)r}}\exp\left(i\int_0^r \xi_m(r')dr'\right) \\ &\equiv \sum\nolimits_m^M \hat{C}_m \frac{\Psi_m(r,z)}{\sqrt{\xi_m(r)r}}\exp\left(i\int_0^r \xi_m(r')dr'\right) \equiv \sum\nolimits_m^M P_m, \qquad (10.8) \\ \hat{C}_m &= \sqrt{N_s W_0}\exp(i\pi/4)\sum\nolimits_j^J \sqrt{\rho_j c_j}\Psi_l^*(0,z_j)\Psi_m(0,z_j). \end{aligned}$$

[7] In the case of air guns, the variation of the radiation directivity is performed by introducing corresponding time delays.

We assume here that the sources are positioned at the points with the coordinates $(0, z_j)$, where j is the source number ($j = 1, 2, \ldots, J$, and J is the number of sources in the chain), and W_0 is the total power of all the sources. The coefficient N_s is equal to

$$N_s = \left[\sum_{j=1}^{J} \left|\Psi_l(0, z_j)\right|^2\right]^{-1}, \tag{10.9}$$

ρ_j and c_j are the density and the sound velocity at the point of the j-th source position. For a vertical source chain spanning most of the channel, the inequality $\hat{C}_{l=m} \gg \hat{C}_{l\neq m}$ is true due to the normalization (B.32) and the orthogonality of eigenfunctions (B.33). Thus the sound field $P(r, z)$ at the receiving point corresponds to the waveguide mode of number l. Thus the vertical array represents a single-mode radiator.

10.3 Receiving Acoustic Arrays and the Design of Large-Scale Acoustic Monitoring Experiments on the Shelf

Acoustic signals employed for large-scale monitoring in shallow water are generally received using vertical or horizontal linear arrays comprised of vertical or horizontal chains of point receivers (hydrophones) combined with a device for forming the directional pattern. The use of such arrays has a variety of advantages over a single point receiver. In particular, the use of a horizontal linear array makes it possible to steer a directional pattern toward the sources of acoustic signals, and thus significantly increase the SNR. For isotropic noises the array gain for a horizontal linear array is: $\mathrm{AG} = 10 \log n$, where n is the number of receivers in the array. Receivers are usually arranged at a distance of half the acoustic wavelength, $\lambda/2$, the spatial Nyquist criterion.

For a vertical array in a shallow water waveguide, the gain is less; however, in this case, the array allows better spatial selection individual normal modes (Sect. 8.3). As noted above, this is necessary for modal acoustic work in shallow water, including mode tomography. By analogy with the vertical single-mode radiating array, and according to (8.70), the receiver transmission coefficient is given by $\chi_i = N_r \Psi_m^*(r, z_i)$, where i is the receiving hydrophone index, $i = 1,2,\ldots,I$, I is the number of array receivers, z_i is the depth of the i-th receiver, and N_r is a proportionality factor.[8] In this case, the signal from a point sound source or

[8] Notice that factor N_r depends on the conversion ratio of sound pressure to electrical signal at the output of each receiver. It may vary within wide limits.

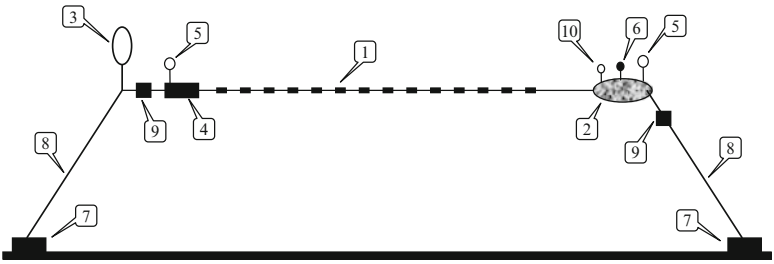


Fig. 10.14 Horizontal receiving array. (1) cine of array with hydrophones, (2) leakproof container with measuring equipment, (3) float for antenna tension, (4) leakproof container with auxiliary equipment, (5) acoustic beacons for navigation, (6) transducer elements of the acoustic-underwater communication system, (7) anchors, (8) kapron halyards, (9) releases, (10) flashing beacon

from an isotropic point scatterer at the output of the device forming the directional pattern is given by

$$
\begin{aligned}
u(r,z) &= \sqrt{N_{\mathrm{r}} W_0}\exp(i\pi/4)\sum\nolimits_i^I \sqrt{\rho c}\Psi_l^*(r,z_i) \\
&\quad \sum\nolimits_\mu^M \frac{\Psi_\mu(0,z_0)\Psi_\mu(r,z_i)}{\sqrt{\xi_\mu(r)r}}\exp\left(i\int_0^r \xi_\mu(r')dr'\right) \\
&\equiv \sum\nolimits_\mu^M \hat{C}_\mu \frac{\Psi_\mu(0,z_0)}{\sqrt{\xi_\mu(r)r}}\exp\left(i\int_0^r \xi_\mu(r')dr'\right), \qquad (10.10)
\end{aligned}
$$

$$
\hat{C}_\mu = \sqrt{N_{\mathrm{r}} W_0}\exp(i\pi/4)\sum\nolimits_i^I \sqrt{\rho c}\Psi_l^*(r,z_j)\Psi_\mu(r,z_j).
$$

Here it is assumed that the sound source or scatterer is at the point with coordinates $(0, z_0)$; ρ and c are the density and speed of sound at this point, respectively, and μ is the mode number. As for the single-mode radiating array, when the receiving array spans the entire waveguide, $\hat{C}_{l=\mu} \gg \hat{C}_{l\neq\mu}$, and the array essentially responds only to the l-th mode.

An example of a shallow water array (Korotin et al. 2000) is shown in Fig. 10.14. The array contains 64 equally spaced receivers. The total array length is 64 m. The array's operational frequency range is from 5 Hz to 10 kHz. The operational depth is down to 250 m.

Some crucial elements in array design should be noted, which are important to receiver systems set in shallow water. Among them are:

- A system for the horizontal linear positioning of the array. For this example, it includes the float 3 (Fig. 10.14), a 10-l water-filled tank for variable ballast (the tank is in the leakproof container 4), depth sensors, and a controller, which controls the ballasting.

- The use of sound receivers with rather small linear sizes, as well as a special technique for joining receivers with a communication cable, necessary to diminish flow noise. In particular, for the array design under consideration, the receiver diameter is only 22 mm, while the cable diameter is 20 mm.
- A digital device for data recording to a hard disk located in a leakproof container.
- A system of "signaling sensors" for the array (a leak sensor and a battery life sensor are very useful).
- Power supply units (batteries or accumulators) for maintaining autonomous array operation for several days to several weeks.
- A universal time system[9] using (at least) quartz and sometimes rubidium frequency standards, which allows measurement of the propagation time of acoustic signals with an accuracy of several tens of microseconds.
- Acoustic-underwater sound communications with a surface ship, which allows one to monitor array performance and control its operation.
- Recovery devices for retrieving the array from the sea bottom and getting it back aboard the ship (acoustic releases and a flashing beacon are basic). One can also add a radio beacon and a radar reflector, which are easily installed.
- Acoustic navigation beacons for accurately determining the coordinates of the array on the sea bottom, and perhaps the array elements themselves.

Vertical receiving arrays as a rule have the same design features, except for the system of horizontal linear positioning. The example of such a 16-element vertical receiving array (Miller et al. 1993) is shown in Fig. 10.15. Here an important structural element is inclinometers, which continuously measure the array inclination angle and direction. An acoustic navigation system is also often used instead of inclinometers in at-sea acoustic experiments. A typical acoustic navigation system consists of three expendable acoustic transponders positioned on the sea floor at a distance of a few hundred meters from the array. These periodically measure the coordinates of acoustic transponder beacons mounted on the array. The measurement accuracy is generally on the order of one meter (to perhaps three meters at worst).

A data transmission system via a radio channel to a surface ship or coastal station is also used often in autonomous acoustic receiving arrays, instead of recording data from the array receivers to a hard disk. The radio buoy providing such transmission is shown in Fig. 10.15. The main advantage of a radio channel is the opportunity for real-time processing of the received information. In this case, the prompt tuning of array operating conditions becomes possible as well. Unfortunately, the radio buoy range depends on sea surface roughness and does not exceed a few tens of kilometers. It seems to be optimum to simultaneously use both systems to get information from array sensors, i.e., the main data stream is recorded to an off-line hard disk, while the radio channel is used only for control and tuning. This was

[9] It should be noted that in acoustic experiments in the shallow water, such a system is as a rule installed at all devices of real-time data acquisition and processing.

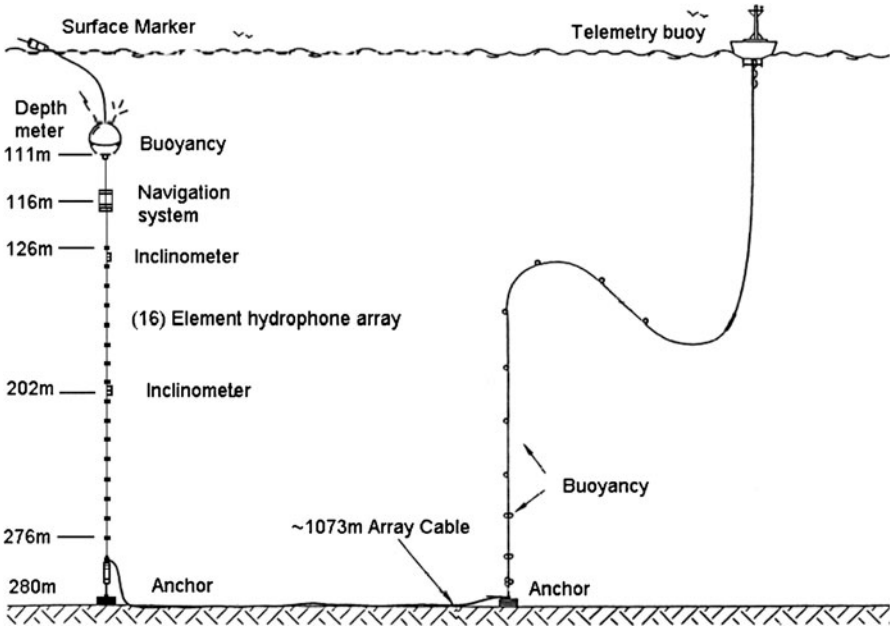


Fig. 10.15 Vertical line array

the mode of operation of one of vertical arrays in the 1995 SWARM experiment (Apel et al. 1997).

In acoustic experiments in shallow water, apart from the emission and reception of acoustic signals, synchronous measurements of the oceanographic characteristics are almost inevitably required. Some standard instrumentation used to this end is shown in Fig. 10.16. Among the oceanographic characteristics, one needs to sense the primary ones are the depth dependences of the seawater temperature $T^{\circ}(z)$ and the salinity $S(z)$.

Most commonly, these profiles are measured from a drifting or anchored ship using a depth-scanning CTD (conductivity–temperature–depth) probe incorporating a seawater conductivity sensor (the measured conductivity is used to determine the salinity), a temperature sensor, and a pressure (depth) sensor. In parallel with CTD "yoyo" stations, expendable bathythermographs (XBTs) are widely used also to measure $T^{\circ}(z)$. They are dropped from ships and transmit the data on the depth and temperature via wire during their descent to the sea bottom.

Mention should be made here of the new technology of underwater gliders with attached CTD sensor systems (Webb et al. 2001). These efficient autonomous underwater vehicles provide unique possibilities for measurements of the temperature $T^{\circ}(r, z, t)$ and salinity $S(r, z, t)$ fields in shallow water, and merit a somewhat more detailed discussion. An underwater glider dives periodically to a selected depth and then ascends to the surface; in so doing, its horizontal speed does not

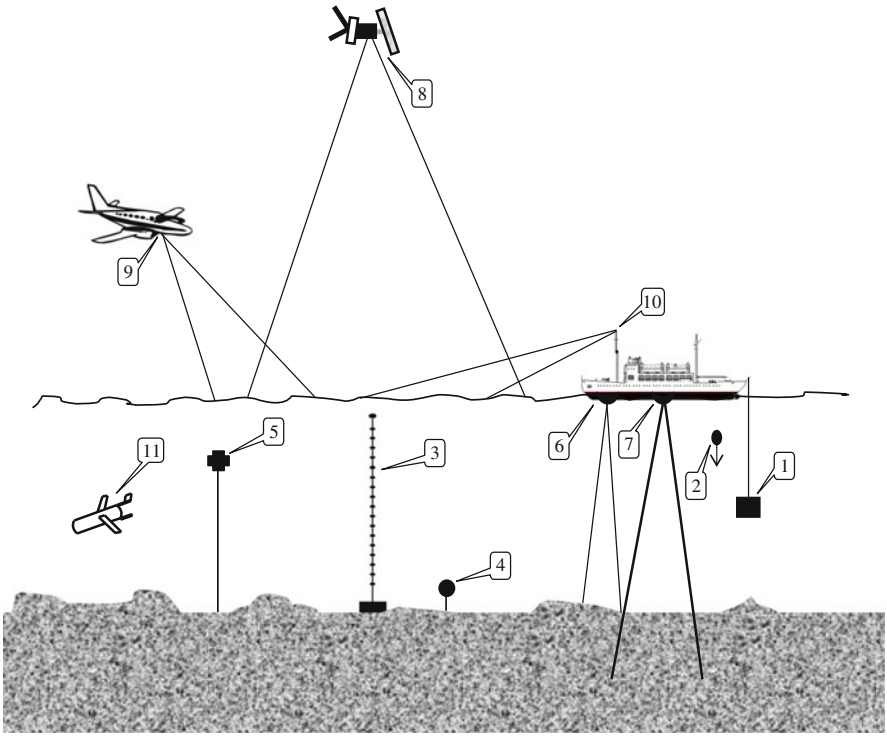


Fig. 10.16 Oceanographic/environmental instrumentation used in acoustic tomography experiments. (1) CTD sensor; (2) XBT, (3) thermistor string; (4) ADCP; (5) three-axis current meter, (6) narrow beam echo-sounder; (7) parametric echo-sounder; (8) synthesic aperture radar; (9) aircraft radar; (10) ship radar, (11) underwater glider

change. Thus this underwater vehicle glides with angle $-\chi_g$ or $+\chi_g$ relative to the horizon. Optimal values of the angle χ_g lie in the range from 30° to 45°. Another interesting attribute of this Webb vehicle is the application of a thermal engine which uses the difference between the ambient temperature (seawater temperature) near the sea surface and at large depths. Because of this difference the engine working fluid refrigerates to near freezing with descent and contracts. With ascent, the working fluid heats, melts and expands. These effects have made it possible to rapidly change the glider buoyancy from positive to negative and vice versa by using a system of bladders and regulated valves that open at predetermined instants of time (see Webb et al. 2001 for more detail). We note that with a buoyancy of only 50 g, the glider can obtain a horizontal speed of 0.25 m/s when it is moving at an angle of $\chi_g = 40°$. By this means, the main energy needed for propulsion is obtained from the surrounding environment. Consequently, this underwater vehicle is capable of operating for an appreciable length of time (perhaps up to 5 years). This is very important for shallow water acoustics and oceanography. To aid in such long-term monitoring, the glider has three antennas (GPS, line-of sight LAN,

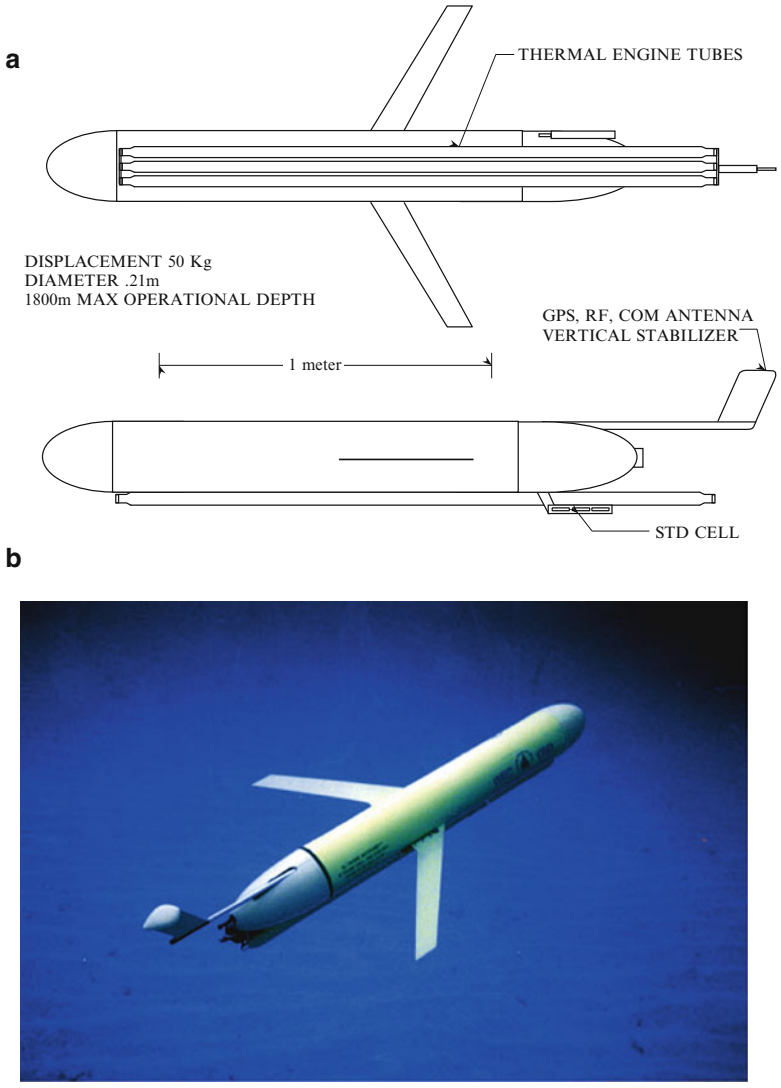


Fig. 10.17 Underwater glider (**a**) schematic sketch, (**b**) appearance

and ARGOS) on its tail fin. The first two antennas are used for the determination of the glider's position when surfaced and the third one is for data communication. A glider outline and photograph are shown in Fig. 10.17. Performance attributes of this underwater vehicle appear in Fig. 10.17 as well.

Underwater gliders are not the only submersible craft which are used at present for measuring the temperature $T^{\circ}(r,z,t)$ and salinity $S(r,z,t)$ fields in shallow water – other devices are used for these purposes as well, specifically autonomous underwater vehicles (AUVs). As an example, the REMUS (Remote Environmental



Fig. 10.18 Remote Environmental Monitoring UnitS (REMUS)

Monitoring UnitS) developed by the Oceanic Systems Laboratory at the Woods Hole Oceanographic Institution is one of the most widely used autonomous underwater vehicles in the world. In contrast to a glider, this vehicle has three motors for propulsion, yaw control and pitch control. As a result, a vehicle-like REMUS works at high speed (operating velocity 0.25–2.8 m/s) as compared to a glider, but has smaller endurance due to its power consumption. Typical endurances for the smaller vehicles (e.g., the REMUS 100, rated to 100-m water depth) are about half a day before recharging is needed. The larger vehicles, however, can hold proportionally more batteries, and endurances for these are now several days, and even weeks. The REMUS 100 vehicle is 160-cm long with a body diameter 19 cm. It weighs just under 37 kg in air, with a displaced volume of 35.7 l. Maximum operating depth is 100 m, with the larger vehicles in the family being rated to 600, 2,700, and 6,000 m respectively. Its mission path is specified by programming an embedded computer. Traditionally, in addition to CTD sensors REMUS has an acoustic Doppler current profiler, a side scan sonar, cameras, light scattering measurement devices, a digital compass and a yaw rate gyro. A REMUS photograph is shown in Fig. 10.18. It is significant that low- and medium-frequency sound sources and receivers can be installed on these vehicles as well. By this means, REMUSs can and are being used as research platforms for the types of shallow water acoustics we have discussed in this book.

The depth dependences of $T(z)$ and $S(z)$ are generally determined using the afore-mentioned devices. These are then used to calculate the vertical profile of sound speed in seawater using known empirical relations. We note that from among a variety of such relations, UNESCO recommends employing the formulas proposed by Chen and Millero (1977) and Millero and Li (1994). However, it should be emphasized that these relations do not provide the best calculational accuracy for the speed of sound in all (without exception) shelf regions and for all seasons.

As already mentioned in this book, the acoustic channel in shallow water represents a waveguide with a random spatiotemporally varying refractive index. In this case, the refractive index variations contain both quasi-determinate and random components. Moreover, the upper and lower waveguide boundaries are randomly rough, and the waveguide itself lies on a layered halfspace, whose acoustic characteristics are also in general randomly inhomogeneous. For this reason, instruments for measuring the parameters of such inhomogeneities are also used in physical experiments on the sea shelf. One particular class is devices for estimating the vertical displacements[10] of liquid layers, which produce sound speed variations in a stratified medium (5.2).

Frequently, such estimates are carried out using the data from thermistor chains (Fig. 10.16). These vertical strings of sensors allow measurements of the time variations of the seawater temperature $T^{\circ}(t)$ at given depths. The vertical displacement at depth z_0 is determined by using the simple relation

$$\zeta(r_0, z_0, t) = \frac{T^{\circ}(r_0, z_0, t) - \bar{T}^{\circ}(z_0)}{d\bar{T}^{\circ}/dz|_{z=z_0}}. \tag{10.11}$$

Thermistor chains often incorporate the same devices mentioned above (for data recording to a hard disk or data transmission via a radio channel; an acoustic navigation system; devices for releasing from the sea bottom, etc.), as linear vertical acoustic arrays. We also note that vertically distributed thermistor elements, rather than point ones, are sometimes used in such chains. This makes possible to significantly decrease the measurement error where there is highly nonlinear dependence in $T^{\circ}(z)$ (Koniaev and Sabinin 1973). In this case, the vertical size is chosen larger than the amplitude $A_{\zeta}(z)$ of the vertical oscillations of liquid, but smaller than the characteristic depth over which $A_{\zeta}(z)$ significantly changes. For example, in the study by Koniaev and Sabinin (1973), the vertical size of a distributed temperature sensor reached 40 m.

Sometimes, thermistor chains are combined with vertical linear acoustic arrays (e.g., Rutenko 1998). In this case, the array can also be accurately tuned to reception or radiation of individual waveguide modes using the data on temperature variations, as was discussed above. Such tuning becomes possible, since the speed of sound depends mainly on the seawater temperature. Having calculated deviations of the vertical profile of sound speed from the average, the eigenfunctions Ψ_m and eigenvalues ξ_m of the Sturm–Liouville boundary-value problem can be more accurately calculated, which is just what is required to tune the mode filtering vertical array. Such tuning can in principle be carried out in real time, using a processor in the array control unit.

[10] These displacements are often associated with internal waves.

The data measured by current velocity meters are also used to calculate vertical vibrations of the water column. Here, the simplest expression is used for $\zeta(t)$ estimations, which is strictly speaking valid in the linear approximation,

$$\zeta(t) \cong \int \frac{S_{v_z}(\omega)}{-i\omega} e^{-i\omega t} d\omega, \tag{10.12}$$

where $S_{v_z}(\omega)$ is the spectrum of the vertical current velocity. Current velocity meters, which typically measure the vertical and two mutually perpendicular horizontal velocities, also allow diagnostics of the current variations associated with a variety of large-scale oceanographic phenomena on the shelf (fluctuating fronts, eddies, etc.). All these features also cause variations in the refractive index in the acoustic waveguide. To measure them experimentally, two current meter types are commonly used. These are (1) contact propeller-type meters with a depth sensor, which allow measurements of the current velocity vector $\vec{v}(z)$ at a single point and (2) remote acoustic Doppler current profilers (ADCPs), which allow measurements of $\vec{v}(z)$ vertical sections (Serebryany 2003). (See devices (4) and (5) in Fig. 10.16.) All current meters incorporate a compass, as well as memory for off-line data recording.

To measure the vertical profile of the current velocity vector, ADCPs form narrow acoustic beams directed at an angle to each other. Velocity measurement in each beam is based on the Doppler effect from ultrasonic backscattering from fine particles or plankton in the seawater and moving together with it as Lagrangian tracers. ADCPs typically operate at frequencies of several hundred kilohertz, and employ four beams propagating at an angle of ~20* to the vertical. Thus, two horizontal and two vertical current components are measured. The double measurement of vertical components is used to estimate the so-called current error due to flow inhomogeneity within the area spanned by the ADCP beams at a given measurement depth. The depth resolution for vertical sections of current velocities is typically about 1 m, and the measurement accuracy is a few tenths of one percent of the current velocity. We note that $\vec{v}(z)$ is also often measured using an ADCP aboard a moving research ship. In this case, the ADCP measuring signals reflected from the sea bottom allows determination of $\vec{v}(z)$ in a fixed coordinate system.

The above-described oceanographic instruments (excepting gliders) are mainly used to estimate the temporal variations of the vertical displacements of the water and current velocities. To estimate spatial variations, quite different device types are used, which detect surface manifestations of IWs and large-scale flows. As an example, we recall that the orbital currents of IWs, by modulating small-scale surface wind waves, form a "striped" structure of surface roughness, which reproduces the spatial structure of the deep vertical displacements of liquid. This structure, as well as manifestations of large-scale flows, is distinct in radar images of the sea surface. Such images are generally obtained using radars operating in the frequency range from 1 to 100 GHz. The main advantage of these devices is that cloud cover has almost no effect on their efficiency. Furthermore, they can operate

at night in contrast to, e.g., devices operating in visible or infrared wavelength ranges. Radars are installed on ships, aircraft, or spacecraft (Fig. 10.16). In particular, to most clearly detect images associated with IWs, it is advantageous to use side-looking airborne radar with an array several meters in length extended along the aircraft fuselage. For an aircraft cruising at an altitude of about 2 km, such radars operating at a frequency of 13 GHz produce an image strip (illuminated area or swath) about 10-km long and with a spatial resolution pixel of 25 × 25 m (Bulatov et al. 2003). Under the (not fully true) assumption that the image brightness is proportional to the IW amplitude in the frequency range corresponding to soliton-like IWs, photographs obtained via radar make it possible to determine IW phase front curvature and propagation direction. Spectral analysis of individual parts of images allows calculation of the IW directivity function $\theta(\varphi)$ (Chap. 2).

Radars of another type based on the "aperture synthesis" method are used for spacecraft (Bulatov et al. 2003). In this method, special coherent (i.e., phase conserving) processing of scattered signals is carried out during a certain time Δt, for which the spacecraft moving with velocity v_s flies distance $l_s = v_s \Delta t$, which is in fact the synthesized array length. For an l of several kilometers, almost the same spatial resolution can be attained as for side-looking airborne radar. In this case, the surveillance strip is much larger, about 100 km. However, an obvious disadvantage of the spacecraft is that the spacecraft flies above the same place only once every several days. This does not allow detection of the rather rapid changes of the sea surface, which take place over intervals from tens of minutes to several hours.

To estimate another type of inhomogeneity, i.e., bottom relief roughness and volume inhomogeneities of the sea bottom, conventional narrow-beam echo-sounders and parametric echo-sounders (see devices (6) and (7) in Fig. 10.16) are used. In shallow water, with a beam width of $\Delta\varphi = 1^\circ - 3^\circ$, such echo-sounders allow detection of inhomogeneities with characteristic sizes L ($L \cong \Delta\varphi \times H$, when H is the sounding depth) comparable with an acoustic wavelength. We note that parametric echo-sounders typically operate at rather low frequencies (several kilohertz). At these frequencies, sound is weakly absorbed in the sea bottom. As a result, it becomes possible to detect echoes from sea bottom inhomogeneities at depths of several tens of meters from the water-bottom interface. In other words, it becomes possible to determine the bottom sediment structure at those depths, which (as was shown previously) plays an important role in long-range sound propagation in shallow water. An alternate method for solving this problem comes from the application of the seismic profiling method, well-known in geophysics, in which several towed linear receiving arrays several kilometers in length are used, as well as electric-spark sources emitting short pulses of several milliseconds in duration.

Unfortunately, both parametric echo-sounders and seismic profiling do not yield data on the absolute values of the density, sound speed, and attenuation coefficient for various sea-bottom regions. Therefore, experimental data are conventionally processed using some typical values of the indicated parameters obtained in the following ways.

- Direct geological core sampling, followed by measurements of the sample acoustic characteristics under laboratory conditions. (Sometimes, bottom drilling samples are used as well.)
- Special methods for signal processing and inverse problem solution (see, e.g., Tolstoy et al. 1998; Fokin and Fokina 2003).

However, it should be noted that core sampling over a large area, not to mention drilling, is an extremely expensive and laborious method. Moreover, the maximum depth from which such samples can be obtained as a rule does not exceed 10 m. As to the estimation of bottom inhomogeneity parameters by solving the inverse problem in acoustic wave scattering and reflection, the accuracy of these methods in many respects depends on characteristics of inhomogeneities themselves and the characteristics of the upper water layer. In other words, the accuracy can depend on the sea shelf region where the methods are used.

In closing this section, it is also worth noting that modern techniques of acoustic experimentation on the shelf are almost impossible without our present satellite global positioning system (GPS). Navigation receivers of this system, which provide a position determination accuracy no worse than $\cong 1\,\mathrm{m}$, are commonly used in all experimental devices or systems, or at least are used when positioning devices on the sea bottom. Using GPS, a universal time system is provided for experiments as well.

Chapter 11
The Future

11.1 Introduction

In Chap. 1, we briefly discussed the future of shallow water acoustics as part of motivation for this book. In this chapter, we will attempt to go into some more detail on the various topics that we touched there and in other portions of the book. We will do this in a broad topical way, treating oceanography, bottom acoustics, acoustic propagation theory, biology, stochastic acoustics, measurement techniques, inverse problems, and signal processing, among other topics.

11.2 Physical Oceanography

Let us now take a look at the various pieces of physical oceanography that enter into our acoustic environmental description, and see what the future might hold for them. Let us start with one of the older and more venerable oceanography topics, tides.

While we can reasonably claim that the barotropic tides are well understood in shallow water, the same is not exactly true for the baroclinic (internal) tides, and particularly their nonlinear evolution into entities like soliton trains or "solibores." While a good deal of the physics of these waves is understood, predicting them deterministically, and particularly their 4-D structure, is beyond our present capability (and may remain so, in which case useful statistical descriptions must be developed). Most present day, large-scale ocean computer models are just at the edge of incorporating the M2 internal tides, and certain specialized supercomputer models (e.g., the SUNTANS model) even look down to the soliton level, but it is not likely that we will see any full ocean model that allows us to do fast realizations of the full internal wave field on the continental shelf for acoustics usages. One possible way around this is the use of "nested domains" to address the problem, i.e., grafting a simpler internal wave model on top of a mesoscale regional oceanographic model. An algorithm for doing this has recently been proposed by Duda,

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7_11,
© Springer Science+Business Media, LLC 2012

Lermusieaux, and Lynch. Using the critical slopes to define the wave source regions, and then using an ocean numerical model to define the internal wave modal eigenvalues over the shelf, one can create an approach very similar to the Weinberg-Burridge "vertical modes and horizontal rays" for acoustics, except for an extended source region rather than a point one. An "*XY* plane index of refraction field" is created from the ocean internal wave modes, and then the rays are propagated out from the source points across the shelf. Nonlinear wave equations (e.g., the KdV equation) are solved along the ray paths, and then wave fronts are created from these solutions. This would create a fully 4-D field, with much of the correct spatial/temporal structure of the internal waves, which is key for input to acoustic models. It would, of course, miss some of the more detailed nonlinear dynamics of the waves that a full, first principles solution to the Navier Stokes equations would incorporate, but for acoustics work (and perhaps some other areas) it would perhaps be sufficient.

There is a wealth of 4-D spatial structure to the soliton field, including wave front curvature, a frequency-directional spectrum that is also a strong function of the XY position and time, truncated waves, crossing wave trains, and a strong link of the field to local bathymetry, density stratification, and currents. These 4-D characteristics of the soliton field make the acoustic propagation and scattering more rich and complex, in many ways. Shallow water researchers are currently exploring the nature of the individual mechanisms of propagation, ducting and scattering of sound by nonlinear internal waves, and we will show one or two recently published examples here. Specifically, we will look at the examples of (1) curved internal wave ducting of sound for a source between the internal waves and (2) curved internal wave shadowing and reflection of sound for a source just outside of internal wave train. These are two of a dozen such effects that are being explored, but should be illustrative of the class (Lynch et al. 2010).

An example of acoustic propagation between two curved, nonlinear internal waves is shown in Fig. 11.1. In this figure, we see a very pretty "light pipe" type of effect, where there are two trapped acoustic modes, the higher of which is ducted between the two curved solitons and the lower of which leaks out dispersively in a rainbow/prism type of effect. This ducting between curved waves can give a false bearing to an acoustic source, which is a concern for sonar systems. From the theory in Lynch et al. (2010), this type of ducting should be seen for larger internal waves, lower frequencies, and higher acoustic normal modes.

If one now puts an acoustic source just outside a curved internal wave train, one gets an entirely different set of acoustic effects, as illustrated in Fig. 11.2. In this example, one sees both shadowing by the curved train, and also a caustic, high-intensity region on the reflected energy side of the train. Moreover, this reflected energy has a distinct interference pattern associated with it, which is just the interference between the direct and side reflected path, i.e., a horizontal version of Lloyd's mirror. This again has some interesting sonar ramifications, in that, while some sonar operators may be savvy enough to realize that a single source can produce multiple target images via the separate normal modes, seeing two copies of, e.g., mode four coming at you from different directions, would seem to be two

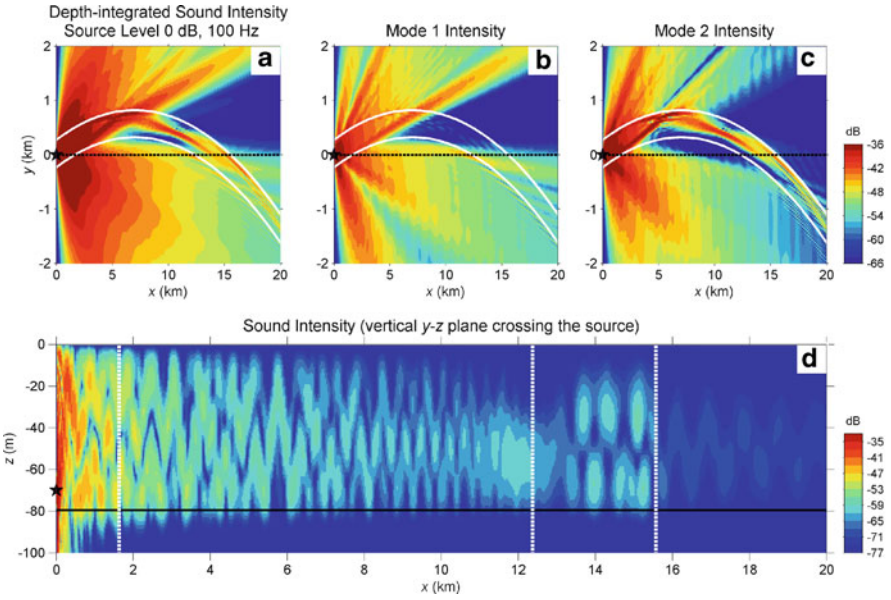


Fig. 11.1 An acoustic source placed in between two curved nonlinear internal waves. Intensity in *d*B relative to intensity of the source (**a**) Depth-integrated sound intensity at 100 Hz. (**b**) Vertical mode one intensity. (**c**) Vertical mode two intensity. (**d**) Vertical slice of sound intensity at $y = 0$

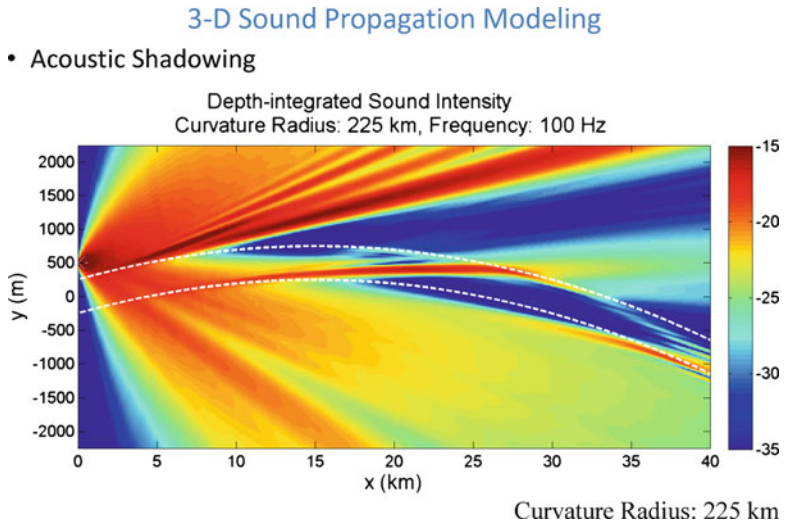


Fig. 11.2 An example of the acoustic shadowing, caustics, and the "horizontal Lloyd's mirror" effect that one sees from a source located just outside of an internal soliton train. Intensity in *d*B relative to intensity of the source

distinct sources. On the other hand, knowing that you are receiving a direct and reflected path actually allows a simple source localization via triangulation, so that a very savvy sonar operator can use this to his advantage!

There are numerous other acoustic scattering/refraction/reflection processes that can occur due to nonlinear internal waves, and these are being actively pursued by researchers. It would not be amiss to predict that we will be reasonably conversant with the physical mechanisms of sound scattering by the various sorts of internal wave solitons in the next few years. However, being able to propagate sound through an acoustics code that has as input a full realization of the coastal internal wave field is also a necessity, in that the overlap and interaction of the individual mechanisms take place at this larger scale. Without that larger picture view, our "real world" predictive capability will remain limited, and perhaps severely so.

The linear internal wave field description also can use improvement, as it certainly differs in shallow water from the deep ocean GM spectrum. It appears that in regions where there is significant soliton activity, the linear field is in fact the tail of the (very directional) soliton trains, and so the directional isotropy of GM is entirely inappropriate. On the other hand, in regions where there is no soliton activity (and these may be relatively rare, another interesting question), there may be a weak isotropic linear wave field. This is an open topic at present.

Eddies and filaments are yet another important acoustic effect in shallow water, as they can duct or refract sound significantly. Of particular note is the turbulence shed by the big, northern hemisphere western boundary currents, the Gulf Stream and Kuroshio, whose acoustic effects have been clearly seen on the shelf in the recent PRIMER (1996) and ASIAEX (2000) experiments. This eddy radiation is hard to predict deterministically, and so we need to get some measurements of the statistics of these fields for acoustic purposes. The classic $k^{-5/3}$ turbulence spectrum is just not adequate, and more work is needed here. The SW06 experiment has added another data set to address this question, and some information may be coming soon, at least for that region. It is interesting that acoustic fluctuations along a path might provide one of the best ways to measure this spectrum.

Surface ducts are also important acoustically in shallow water (when they exist), but they too are under-measured in 3-D, for the same reason as the mixed layer. It is hoped that some new data from surface buoys, AUVs or other instruments will be available in the near future to look at the 3-D structure of these ducts.

11.3 Bottom Acoustic Properties

For most purposes in low-frequency shallow water acoustics, a multi-layered geoacoustic model (compressional and shear wave speeds and attenuation and density) of the sub-bottom properties will suffice. However, even after a half century of efforts to obtain such models, the fact remains that we do not have a fast, simple, and cheap way to survey the sub-bottom for these properties. All experiments to date are ship and equipment intensive, which make survey work (particularly for 3-D sub-bottom models, which are part of what we are striving for) inefficient.

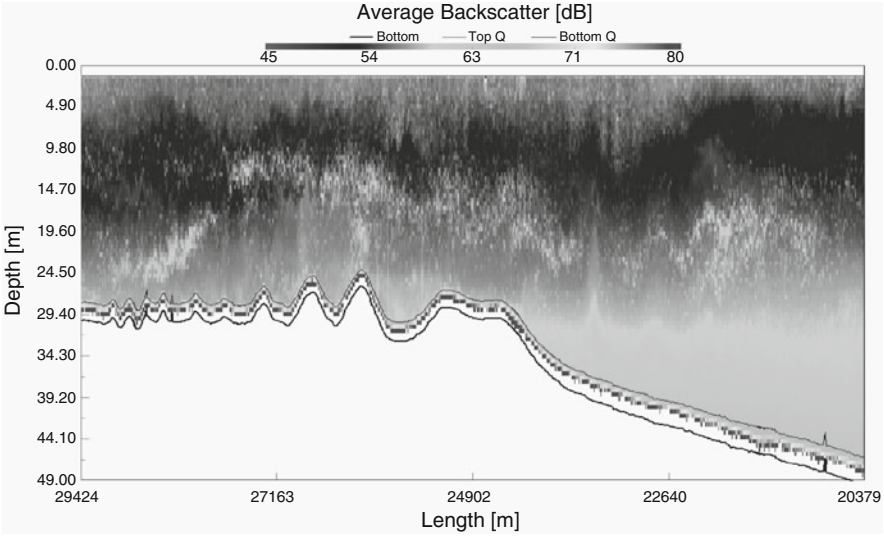


Fig. 11.3 Large sediment ripples in the Sea of Japan, courtesy Andrey Serebryany

It is the view of the authors that AUVs, carrying acoustic sources and receivers may eventually solve this problem, and allow researchers to "mow the lawn" to see 3-D spatial structure in the seabed.

Another question that should be addressed further is just what degree of structure does one need when working on a particular application and at a given frequency. The choice of "minimalist" models suited for a given task seems like an efficient way to do business. This is an optimization problem, but one that is not often addressed, leading to people often "overprescribing" what models or information they need (when in doubt, overkill works?).

Yet another question that could use more work is what averaging of the exact bottom structure to get an "effective" bottom model is justified and effective. This is again a "minimalist" consideration, but again, we think a necessary one to consider, especially as many of the seabed measurements that we do incorporate averaging of the medium a priori. This question is intimately related to the correlation scales and structure of the seabed, and so has a geological basis.

The recent work on poro-elastic media is very intriguing, and one area where this could be extended is to various mixtures of sediment types (which one sees naturally, anyway). Mud is also a material that is of interest to bottom acoustics, and has been little explored compared to sands (perhaps because of its relative abundance).

Another interesting future possibility is the study of what continental shelf sediment ripple fields do to acoustic propagation, particularly in the sense of 3-D propagation. Bottom roughness scattering has been extensively studied as a 2-D process, but the 3-D structure of some of the larger ripple fields and its 3-D effects on acoustics is just now being looked at. As an example of how large some of these features are, we refer the reader to Fig. 11.3, which is an interesting ripple

field that may have been created by ocean solitons. Whatever the generation mechanism, we see that there are ripple features up to 6 m high, or 20% of the water depth! Ripples over 10% of the water depth are candidates for 3-D acoustics effects, so these certainly would seem to qualify.

11.4 New Directions in Theory and Modeling of the Sound Propagation

11.4.1 3D Problems

One would think that acoustic propagation theory was a totally dead area, with no new interest (other than scattering problems, which are a somewhat different proposition). A 2-D, linear second-order partial differential equation is not exactly considered a challenge anymore, in an era when 4-D, nonlinear turbulent oceans and atmospheres are being modeled with ever increasing success. However, the recent observations that 3-D oceanography could create significant 3-D acoustics effects has created a need for some updates in the acoustic numerical models, and we are now looking at upgrading various acoustics models for 3-D. An efficient 3-D modeling technique was the "vertical modes and horizontal rays" formalism introduced by Weinberg and Burridge; this technique has recently been updated to "vertical coupled modes and horizontal parabolic equation" by the authors and other investigators. Moreover, the fully 3-D parabolic equation model (e.g., FOR3D by Ding Lee et al.) is also being looked at for upgrades, for instance to Cartesian coordinates. Fully 3-D PE will be needed for coastal, shelfbreak, and canyon applications, so more acoustic numerical modeling work now seems justified at this point in time. While this new modeling probably will not startle the overall numerical modeling world, it does at least show that there is something new in the area, and that shallow water acoustics has pushed the "state of the art" a bit.

11.4.2 Interference Invariant

For example, in recent years, the interference invariant β has been the topic of much attention, in that it is a rather easily observed and robust acoustical feature. It is our feeling that there is still much work that can be done in using observations of this invariant to invert for ocean and seabed properties, and that this should be pursued in the years to come.

11.4.3 Dislocations of the Wave Field

The concept of the dislocation of the wave field was originally introduced by Nye and Berry and was developed for its applicability to optics. This concept can help in the description of the complex interference structure of the sound field in shallow water waveguide, and probably will be helpful for better understanding the nature and properties of fluctuations of the sound field (intensity, amplitude, phase, different correlation functions, etc.)

11.5 Physical Phenomena, Concerned with Stochastic Effects

Scattering from random inhomogeneities in the ocean is always a rich topic. An area that should continue to attract interest is to infer the spectra of the inhomogenities from the acoustic fluctuation spectra, i.e., the inverse problem for spectra. It is hard to get in situ spatial sampling over many wavelengths of ocean features, and acoustics offers that possibility. Moreover, if it is possible to see Bragg resonances from certain scales by using an assortment of modes and frequencies, the technique becomes more robust.

The transition from focusing weak scattering to saturated scattering in shallow water (versus the type of oceanography scattered from) is another question worth quantifying. Many investigators have recently been exploring the varieties and strengths of scattering versus azimuthal angle in coastal transmissions through internal waves, and it is this type of study that will allow us to eventually predict the detailed fluctuation behavior of shallow water.

Shallow water acoustic reverberation from a flat-bottom geometry, while not perfectly known, is at least fairly well described at this point in time. The challenge, it seems to us, is in better describing reverberation near the continental slope, where one encounters sloping bathymetry, canyons, fronts and eddies, fish schools, and extremely variable sediment structure.

The low-frequency volume reverberation due to fish schools is a topic that has received much attention recently (Makris et al. 2006), and indeed, in addition to quantifying the "bio-clutter" volume reverberation that detracts from active low-frequency sonar performance, the use of low-frequency sources and towed arrays has provided researchers with an interesting new tool to look at the large-scale structure of fish schools, which can be on the scales of kilometers. This is an interesting combination of naval and basic biology applications. It is likely that sound field focusing and focal spot scanning have opened up fresh opportunities in this field.

The noise field in shallow water is often dominated by a single thing – shipping. And, this is not distant shipping, but rather loud, up-close and personal, in-your-face shipping, which blasts out enough noise to pretty much drown out any other signal of

interest. One area that is of interest both to this chapter and the last one is how one deals with such noise when doing the detection problem, e.g., how does one find a weak signal in the presence of a strong one, specifically in the context of shallow water.

While loud ship noise is obnoxious in the context of the detection problem, it can also be used as a signal for bottom inverses, for instance using the waveguide invariant as a tool. This has been pursued, and given the difficulty in permitting standard research sources nowadays, also should be pursued more in the future.

The noise notch in shallow water is still a topic that could bear further examination, in that "if open", one can discriminate between surface and subsurface sources using the comparison between the power in the high- and low-acoustic modes. However, when the noise notch closes due to scatterers (waves, internal waves, bottom roughness, etc.) redistributing the modes, then this technique becomes more problematic.

Finally, we should remark that 3-D oceanography and bathymetry can also substantially affect the shallow water noise field, and that there should be further work, both experimental and theoretical, in exploring this topic.

11.6 Inverse Problems

11.6.1 Water Layer

Inverting for the water column sound speed structure in shallow water is more complicated than deep water tomography because of two factors (1) strong bottom interaction and (2) strong mode coupling. Inverting for the water column in the presence of (partially) unknown bottom structure can introduce error into the water column structure estimate, in just the way we showed that water column variability could contaminate the bottom estimate. The strong mode coupling often found in shallow water, particularly when propagating through some interesting ocean or bottom structure, is even more of a concern. Mode coupling produces severe time spreading of the pulses typically used for travel time tomography to the extent that the multipath arrival structure is no longer resolvable, and one is reduced to using the "leading edge arrival" as the sole data point. This results in a very low resolution inverse (Sperry et al. 2003), of a quality not useful either for oceanographic work or for the making of good sound speed maps for acoustics purposes. Indeed, it can usually be shown using a cost versus resolution analysis that using conventional sensors to map the ocean structure in shallow water is far more cost effective and scientifically useful than acoustic tomographic imaging. The exception to this may be in trying to map the spectra of ocean structures using the spectra of acoustic fluctuations for long-range transmissions (order 30–50 km). In this case, one sees many wavelengths of the ocean process with a comparatively small amount of (acoustic) instrumentation.

11.6.2 Bottom Layer

The bottom inverse problem in shallow water is the more important one to address, in that direct, in situ measurements of the bottom are both difficult and costly, and usually access only the upper few meters of the bottom. Acoustic methods are the correct choice for remote sensing, and indeed much has been done in exploring such methods. But, again, much remains to be done.

Over the past few decades, linear inverse methods using both narrowband and broadband data (producing travel time dispersion curves, modal wavenumber spectra, etc.) have been popular and heavily exploited. These methods are simple and produce good error estimates (resolution and variance) along with the "answer" for the bottom structure. However, the linear techniques can quickly break down, and so nonlinear methods were pursued soon afterwards. The use of these methods (e.g., simulated annealing and genetic algorithms) has become "packaged" and simple to use at this point in time, so that one has a powerful and robust inverse technique close at hand. The shortcoming of these techniques is their failure to produce an error estimate along with an answer, and this is where we can see room for improvement of these techniques.

The inclusion of a random, uncertain medium pushes one towards probabilistic techniques, where the answer produced is a probability distribution. The leading contender in this area is Bayesian techniques, and we refer again to the work of Dosso and others as leading the way in shallow water acoustics applications. The Bayesian, probabilistic approaches also have their strengths and weaknesses, which are worth noting. The strength, again, is the ability to include medium variability/randomness as part of the answer produced, which addresses one of our big concerns in inverse theory. However, the answer produced has only part of the error analysis incorporated into it (the variance), and it would be useful to see the resolution eventually included as part of the answer as well. In addition, Bayesian techniques become analytically difficult in the presence of non-Gaussian processes (noise), which presently restrict their power. And finally, in order to get a full probability distribution function (PDF) for an answer, one must have a very large data set, which produces the full range of possible answers. This is asking far too much, and so one must be able to specify which part of the PDF is "illuminated" by the finite data set.

We have previously discussed the "art form" involved in choosing a bottom model, but it could not hurt to re-state that this is still an interesting problem in inverse theory, and one that is often cavalierly disregarded to the detriment of the inverse results produced.

We might note in closing this topic that a "cost versus resolution" analysis is very much part of an experiment for inverting effectively for ocean or seabed structure, and that using some of the tools of inverse theory to do this as part of the "antenna array" design for an experiment can be a good use of inverse theory's toolbox. This is an area that should be pursued far more than it is currently.

11.7 Signal Processing

Signal processing is a huge part of shallow water acoustics, and its interaction with the peculiarities of the shallow water environment provides much technical interest. The three main topics that we have treated here are matched field processing, modal beam forming, and time reversal mirrors. Each has come far, but in each there is still much to do.

Matched field processing (MFP) has one major enemy, mismatch, and that enemy is a strong one in shallow water. It is a given that we will never know any shallow water waveguide perfectly, and in many instances (e.g., operational navy work) the amount of knowledge of the waveguide may be rather small. Thus the signal processors work feverishly towards achieving their Holy Grail, "environmentally robust signal processing." An environmentally robust processor needs no a priori knowledge of the environment to work optimally, or gets whatever knowledge it needs from the acoustic signal it is processing. While in some cases, robust processors can be created, in most cases lack of environmental information results in loss of gain by the processor, an undesirable outcome. Such is the case for MFP, which in its simple forms can be extremely sensitive (un-robust!) to the lack of environmental input, which causes mismatch of the "correlator" signal. Creating robust versions of MFP, which can adapt to whatever level of environmental input is available, is an ongoing research area, and one that seems to be producing usable results.

Modal beam forming is a known technology at this point, but there are still a number of practical issues to be solved in creating sources for individual modes that are not too complicated for routine use. As an example, Buck et al. (1997) have created a "feedback loop" scheme that will create single modes reliably given a vertical array of sources and a nearby "feedback receiver" which talks to the source array and corrects it. This is an elegant and effective scheme on paper, but is rather too expensive and complicated for field work. A cheap, easy single mode generator for field work would be a rather welcome advance in shallow water acoustics.

The topic of time reversal (and its narrowband cousin, phase conjugation) mirroring has become a vibrant and lively one in the past decade, and we would think it fair to say that it is reasonably understood theoretically. At this point, we would say that finding clever and inventive applications for this technique would be the biggest future priority. We know of "tripwire" applications for intruder detection, and some beginning communications application, but think that there should be many more possible. As an example, sound field focusing using a time reversal mirror has opened up fresh opportunities for acoustic environmental monitoring. For example, phase variations in the focal spot can be used to fix point displacements of the seafloor (Iwase et al. 2006). These measurements can perhaps be useful in the future for earthquake prediction. It should also be noted that in all probability, for sound focusing using time reversal mirroring in shallow water, it is not necessary to use a vertical array in all cases. It seems reasonable (Zverev et al. 2008) to say that besides using vertical

arrays and narrow band signals, we can also use point sources and broadband signals. This is cheaper and opens up new possibilities for time reversal mirroring applications.

Finally, we come to a topic that we did not pursue in detail in the text, but that we think will become important in the coming decade – low-frequency shallow water communications. For things such as communicating to subsurface vehicles (AUVs, gliders, submarines, small submersibles) at large distances (5–50 km) that higher frequencies will not reach, low-frequency acoustics should be ideal. As long as one can live with a low bit rate (i.e., keep the messages simple or compressed!), this technique should be fine. There has been some preliminary work in this area by T.C. Yang (2003) of NRL, and by Chinese researchers (Wu, private communication), but by and large the field is rather open and new at this point in time.

11.8 New and Prospective Developments in Oceanographic and Acoustical Equipment and Survey Methods

11.8.1 AUV, Gliders, Etc. and the Construction of Detailed Oceanographic Models

The simple (spatial) 1-D mixed layer model that we discussed in Chap. 2 is interesting, but inadequate; it is easy to show that the fully 3-D (spatial) characterization of the surface mixed layer is important to know for acoustics applications. However, at this point in time, there is very little 3-D mixed layer data and the state of the art of the modeling is rather rudimentary. The lack of 3-D data is easy to understand in that one needs surface and subsurface measurements for mixed layer studies, which means having a spar buoy or surface float buoy. These are rather expensive moorings to begin with, and a 3-D field of them (i.e., many in the x–y plane) becomes prohibitively expensive and impractical. Perhaps inverted echosounders or another similar technology could help solve this sampling problem, but at present no simple, inexpensive sampling solution exists.

Fronts are another acoustically important piece of oceanography, in that they can cause strong mode coupling (for across front propagation) or induce 3-D acoustic propagation effects (for along front propagation). Again, it is a reliable, fully 3-D (spatial) description of the fronts that is lacking as input to the acoustics. Specifically, we are most lacking in data – the modeling of coastal fronts in 3-D *is* in reasonable shape, comparatively. And again the lack of data is a sampling issue. However, modern developments such as gliders, tow-yo CTDs (like Scanfish or SeaSoar), and lower cost, reliable sub-surface moorings have made the acquisition of 3-D frontal data feasible.

In some ways, measurement technology is perhaps the "most fun" of all the topics to consider in relation to "what will the future bring." Technology invariably transforms scientific and engineering fields, as we three authors (all in our mid-to-late fifties) can testify, having watched how the computer and microelectronic revolutions

have taken ocean acoustics observations and data analysis to heights that were undreamed of when we all started in the field. As an example, in SW06 we fielded an acoustic receiver array that had 3.5 TB of internal storage, which allows a 48 element array to take data at a high rate over two full months. Compare that to the 64 K memories that we started with (and that even Bill Gates thought should be sufficient back then!)

The computer revolution should continue to trot along, and though it may slow up (maybe!?), we can probably count on more memory and flops of processing power in the days to come. Software, too, will steadily improve. (Again, the authors would have killed to have MATLAB or MATHEMATICA available when they were students!)

The biggest revolution, in the author's eyes, will probably be the use of AUVs and gliders to make routine acoustics measurements. These vehicles already carry external and internal hydrophones. In the near future, they will also carry low-frequency sources for shallow water acoustics work, and also have the capability to deploy such sources as moorings, and then listen to them. Add docking capabilities for power "refueling" and data dumping, and these vehicles can (and will) be doing multi-month acoustics missions, with the data being transmitted remotely to shore. Coordinated use of groups (schools?) of AUVs is also being worked on, so that one may think of an array of these mobile instruments. We think that developments in this area will be very exciting.

AUVs and gliders will also be gathering the 3-D oceanography data we desire as inputs to our acoustics calculations, which will be combined with data assimilating ocean numerical models. This will allow us to both understand 3-D effects better and to reduce the uncertainty in our predictions of the acoustic field.

Source technology continues to be a sore point for shallow water acoustics. We really could use a cheap, efficient, and (reasonably) powerful low-frequency acoustic source for shallow water work. First of all, we hope that these sources will not use barometric compensation and also can be combined with source receivers. In doing so, acoustic power flux receivers (vector receivers) can be used along with the usual hydrophones. Vector receivers can provide detection advantages, especially in the presence of anisotropic ambient noise (Gordienko et al. 2007).

A rosy future would have combined source and receiver arrays that can be installed both vertically and horizontally. In this case, as was shown in the SW06 experiment, the combined vertical–horizontal array (L-array) exhibits perhaps the most promise for shallow water acoustics. We hope that this type of array will be combined with adequate temperature and salinity sensors, in which case we can use the data from these sensors for matched field radiation and matched field processing where it is necessary.

Another interesting new development is that of cheap, small, low-power rubidium clocks, which will allow one to cost-effectively perform underwater tasks that require high timing precision. This is probably just one example of many existing new technical developments that will impact shallow water acoustics in the years to come.

Finally, we should mention that with global warming, there should be a lot more shallow water soon, and that one would perhaps be well advised to use some of the very effective dry-land techniques (e.g., radar reflection, Rayleigh wave analysis, vibroseis, and even simple cores) to obtain "incipient bottom" properties before the actual submergence of parts of Florida, Louisiana, and Cape Cod (to use US examples).

Not being Verne's or DaVinci's, we will leave things here, with the hope that the robots that will be doing our measurements in the near future are very intelligent ones, and that they will continue to advance the field that their poor, small-memory, low-flop biological predecessors have intrepidly started.

Appendix A
Waves and Signals

Introduction

Here we discuss signals, waves, and fields of various nature propagating in shallow water: acoustic signals and waves, waves on the sea surface, ocean internal waves, as well as assorted other ocean waves and fields. In order to provide a general framework for this discussion, we describe here some mathematical techniques that are useful for handling these physical concepts. These discussions might already be familiar to the reader, and indeed can be found in many places in the standard signal processing and physics literature. Nonetheless, we include them for the sake of both completeness and notation.

Analytic Signals and Waves

To begin with, we consider signals $u(t)$, which are functions of time, where the time axis extends from $-\infty$ to $+\infty$. Like all physical observables, the value of $u(t)$ should be real. We will also wish to consider$u(t)$ in a frequency domain representation for many purposes. $u(t)$ has the Fourier spectrum $S_{\mathrm{u}}(\omega)$ [or simply $S(\omega)$]

$$S(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} u(t)\, \mathrm{e}^{i\omega t} dt$$

$$u(t) = \int_{-\infty}^{\infty} S(\omega) \mathrm{e}^{-i\omega t} d\omega\,. \tag{A.1}$$

Generally speaking $S(\omega)$ is complex function, and due to $u(t)$ being real satisfies the relationship $S(\omega) = S^*(-\omega)$, where the * denotes complex conjugation.

In both signal processing and theoretical physics, it is usually more comfortable to work with a complex (analytic) signal, which has $u(t)$ as its real part:

B. Katsnelson et al., *Fundamentals of Shallow Water Acoustics*,
The Underwater Acoustics Series, DOI 10.1007/978-1-4419-9777-7,
© Springer Science+Business Media, LLC 2012

$U(t) = u(t) + iv(t)$. There are various ways to construct this signal, the most convenient and straightforward being the Hilbert transform (Oppenheim and Willsky 1997)

$$v(t) = -\frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{u(s)}{t-s} ds,$$
$$u(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{v(s)}{t-s} ds. \tag{A.2}$$

where $P\int$ denotes the Cauchy principal value of the integral. The spectrum of the real part of the analytic signal is given by (A.1). It is straightforward to find the spectrum of the imaginary part of the analytic signal:

$$v(t) = -i \int_{-\infty}^{\infty} \mathrm{sgn}(\omega) S(\omega)\, e^{-i\omega t} d\omega, \tag{A.3}$$

where

$$\mathrm{sgn}(\omega) = \begin{cases} 1, & \omega > 0 \\ -1, & \omega < 0 \end{cases}.$$

Thus the spectrum of the imaginary part of the analytic signal is

$$S_v(\omega) = \begin{cases} -iS(\omega), & \omega > 0 \\ iS(\omega), & \omega < 0 \end{cases}. \tag{A.4}$$

We can see that the amplitude of the real and imaginary parts of spectrum is the same

$$|S_u(\omega)| = |S_v(\omega)|. \tag{A.5}$$

The spectrum of the analytic signal is thus:

$$S_w = S + iS_v \tag{A.6}$$

or

$$S_w(\omega) = \begin{cases} 2S(\omega), & \omega > 0 \\ 0, & \omega < 0 \end{cases} \tag{A.7}$$

and so the analytic signal can be written in terms of the spectrum of the real signal as:

$$U(t) = 2\int_0^\infty S(\omega)\mathrm{e}^{-i\omega t}d\omega. \tag{A.8}$$

Indeed, this expression can be taken as the definition of the analytic signal, following (Gabor 1946).

Let us consider the often used comb function, consisting of a sequence of δ-pulses

$$u(t) = \sum_{n=-\infty}^{\infty} \delta(t - n/F). \tag{A.9}$$

It is a periodic function with period $T = 1/F$ or frequency $\Omega = 2\pi F$. It can be decomposed into a Fourier series as:

$$u(t) = \sum_{n=-\infty}^{\infty} a_n\,\mathrm{e}^{-i2\pi Fnt}, \tag{A.10}$$

where

$$a_n = F\int_{-1/2F}^{1/2F} \delta(t)\exp\left(i2\pi Fnt\right)dt = F. \tag{A.11}$$

So as the Fourier decomposition, we get

$$\sum_{n=-\infty}^{\infty} \delta(t - n/F) = F\sum_{n=-\infty}^{\infty} \exp(-i2\pi Fnt).$$

In a more traditional form, this formula is

$$2\pi\sum_{n=-\infty}^{\infty} \delta(t - 2\pi n) = \sum_{n=-\infty}^{\infty} \exp(-itn). \tag{A.12}$$

If we multiply both sides by an arbitrary function $u(t)$ and integrate, we obtain the so-called Poisson formula

$$\sum_{n=-\infty}^{\infty} u(2\pi n) = \sum_{n=-\infty}^{\infty} S(n), \tag{A.13}$$

where S denotes spectrum of the function $u(t)$.

Let us now consider the important situation where $u(t)$ is a random function (a random signal), and for simplicity suppose it is also a zero-mean process, i.e. that $\langle u \rangle = 0$. In this case, the spectral function $S(\omega)$ is also a random one and (A.1) should be interpreted stochastically. We consider only stationary processes, determined by expression

$$\langle S(\omega)S(\omega') \rangle = G(\omega)\delta(\omega + \omega'), \tag{A.14}$$

or

$$\langle S(\omega)S^*(\omega') \rangle = G(\omega)\delta(\omega - \omega'), \tag{A.15}$$

where $G(\omega) > 0$ is some function of frequency. In this case, the correlation function of the random signal $B(t, t') = \langle u(t)u(t') \rangle$ depends only on difference $t - t' = \tau$, and $B(\tau) = \langle u(t)u(t + \tau) \rangle$. It satisfies the Wiener–Khinchin theorem (Leon-Garcia 1994):

$$B(\tau) = \int_{-\infty}^{+\infty} G(\omega)\, \mathrm{e}^{-i\omega\tau} d\omega, \tag{A.16}$$

and correspondingly

$$G(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} B(\tau)\, \mathrm{e}^{i\omega\tau} d\tau. \tag{A.17}$$

Using the above relationships at $\tau = 0$ (zero lag), we see that

$$\left\langle u^2(t) \right\rangle = \int_{-\infty}^{+\infty} G(\omega) d\omega. \tag{A.18}$$

The value $G(\omega)$ is called the spectral density of the signal. If $\left\langle u^2 \right\rangle$ has the units of energy, then $G(\omega)$ is the power spectral density. In reality, we deal only with positive values of frequency, and so the frequency decomposition of the measured correlation function has the form

$$B(\tau) = \int_{0}^{\infty} G^+(\omega) \cos \omega\tau \, d\omega \tag{A.19}$$

and the measured spectral density $G^+(\omega)$ is

$$G^+(\omega) = \begin{cases} 2G(\omega), & \omega \geq 0 \\ 0, & \omega < 0 \end{cases}. \tag{A.20}$$

Similar to (A.7), we can also consider $U(t)$ as a complex random signal. In this case, the correlation function is

$$B(\tau) = \frac{1}{2}\langle U(t)U^*(t+\tau)\rangle. \tag{A.21}$$

Also, if we have two random signals $u_1(t)$ and $u_2(t)$ then we can introduce the cross-correlation function, which for a stationary process has the form:

$$B_{12}(\tau) = \langle u_1(t)u_2(t+\tau)\rangle = \frac{1}{2}\langle U_1(t){U_2}^*(t+\tau)\rangle. \tag{A.22}$$

Correspondingly we can introduce cross-spectral density of mentioned signals:

$$\langle S_1(\omega)S_2(\omega')\rangle = G_{12}(\omega)\delta(\omega+\omega') \tag{A.23}$$

and

$$B_{12}(\tau) = \int_{-\infty}^{\infty} G_{12}(\omega)\,\mathrm{e}^{-i\omega t}d\omega. \tag{A.24}$$

Also we can introduce cross-spectral density in area of positive frequency

$$G_{12}^{+}(\omega) = \begin{cases} 2G_{12}(\omega), & \omega>0 \\ 0, & \omega<0 \end{cases}. \tag{A.25}$$

We can use also normalized value of cross-spectral density which is called coherence

$$\mathrm{Co}^2(\omega) = \frac{|G_{12}(\omega)|^2}{G_1(\omega)G_2(\omega)}. \tag{A.26}$$

Let us now consider expressions for waves in some physical space. Such waves can be written as a real function of the space and time coordinates, $u(\vec{R},t)$, where u can denote different physical values – displacement of a water layer for surface and internal waves or sound pressure for acoustical waves. This function should be the solution of some specific wave equation describing the propagation of waves of different natures. The mathematical form of these wave equations can be essentially different (in particular we will see this in the equations, describing the propagation of surface, internal, and acoustical waves in water) and can be linear or nonlinear. In the linear case, we can write the form

$$\hat{L}u(\vec{R},t) = 0, \tag{A.27}$$

where $\hat{L}$ is some linear operator which is responsible for the wave's propagation, including the mechanisms of absorption, attenuation, and diffusion, and in turn their corresponding parameters. If this medium is homogeneous, then $\hat{L}$ is an operator with constant coefficients, and its solutions have the form of a plane wave $e^{i(\vec{k}\vec{R}-\omega t)}$. In an inhomogeneous medium, the solution has a more complex form.

The function $u(\vec{R},t)$ can be decomposed as a Fourier integral for a given point in the space:

$$\begin{aligned} S(\vec{R},\omega) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} u(\vec{R},t)\,e^{i\omega t}dt \\ u(\vec{R},t) &= \int_{-\infty}^{\infty} S(\vec{R},\omega)\,e^{-i\omega t}d\omega. \end{aligned} \tag{A.28}$$

Correspondingly, we can construct an analytical field (signal) similar to (A.8):

$$U(\vec{R},t) = 2\int_{0}^{\infty} S(\vec{R},\omega)\,e^{-i\omega t}d\omega. \tag{A.29}$$

In solving a specific problem, equation (A.27) must be accompanied by boundary and/or initial conditions. Also, on the right side of (A.27) we must include some function of space and time if we wish to include a source for the waves, described by the complex function in the right side of Eq. (A.27). For complex waves, we have

$$\hat{L}U(\vec{R},t) = -Q(\vec{R},t). \tag{A.30}$$

A powerful and popular way to solve the inhomogeneous wave equation above is to employ the Green function $\Psi(\vec{R},\vec{R}';t,t')$. It satisfies an equation similar to (A.30), i.e.,

$$\hat{L}\Psi(\vec{R},\vec{R}';t,t') = -\delta(\vec{R}-\vec{R}')\delta(t-t'), \tag{A.31}$$

where $\hat{L}$ operates on the $\vec{R},t$ variables. In terms of the Green function, the solution of (A.30) has the form

$$U(\bar{R},t) = \int_{-\infty}^{\infty}\int \Psi(\vec{R},\vec{R}';t,t')Q(\vec{R}',t')dt'd\vec{R}'. \tag{A.32}$$

In a formal way, we can write the solution of (A.31) as

$$\Psi(\vec{R},\vec{R}';t,t') = -\hat{L}^{-1}\delta(\vec{R}-\vec{R}')\delta(t-t'), \tag{A.33}$$

where the task then becomes to find the inverse operator $\hat{L}^{-1}$ and then $\Psi(\vec{R},\vec{R}';t,t')$.

If we use a standard Fourier expansion of the δ functions:

$$\Psi(\vec{R},\vec{R}';t,t') = -\int d\vec{k} \int_{-\infty}^{\infty} d\omega \hat{L}^{-1} \mathrm{e}^{i\left[\vec{k}(\vec{R}-\vec{R}')-\omega(t-t')\right]}, \tag{A.34}$$

when $\vec{k}$ is a 3D wavevector. We then just apply the wave equation operator to the plane waves to generate $\Psi(\vec{R},\vec{R}';t,t')$.

We can use this operator expression directly, by first describing the wave propagation in homogenous free space. In this case, the partial solutions of the wave equation are the plane waves $\mathrm{e}^{i(\vec{k}\vec{R}-\omega t)}$, where ω is function of $\vec{k}$: $\omega = \omega(\vec{k})$ (dispersion relationship). The function $u(\vec{R},t)$ can be decomposed in a space–frequency integral

$$\begin{aligned} u(\vec{R},t) &= \int d\vec{k} \int_{-\infty}^{\infty} d\omega\, S(\vec{k},\omega)\, \mathrm{e}^{i(\vec{k}\vec{R}-\omega t)}, \\ S(\vec{k},\omega) &= \frac{1}{(2\pi)^4} \int d\vec{R} \int_{-\infty}^{\infty} dt\, u(\vec{R},t)\, \mathrm{e}^{-i(\vec{k}\vec{R}-\omega t)}, \end{aligned} \tag{A.35}$$

where $S(\vec{k},\omega)$ is the spectral amplitude of the wave field. We note that the wavevector can also depend on frequency according to the dispersion law $\vec{k} = \bar{k}(\omega)$.

For an analytic signal, we have

$$U(\vec{R},t) = 2\int d\vec{k} \int_{0}^{\infty} d\omega\, S(\vec{k},\omega)\, \mathrm{e}^{i(\vec{k}\vec{R}-\omega t)}. \tag{A.36}$$

Equation (A.36) describes the propagation of pulses. The most important characteristics of pulse propagation are the group and phase velocities. In vector form, the group velocity is defined as

$$\vec{c}^{\,\mathrm{gr}} = \left(\frac{\partial\omega}{\partial k_x}, \frac{\partial\omega}{\partial k_y}, \frac{\partial\omega}{\partial k_z}\right), \tag{A.37}$$

or in a more compact symbolic form

$$\vec{c}^{\,\mathrm{gr}} = \frac{\partial\omega}{\partial\vec{k}}.$$

For the phase velocity, we have the definition

$$\vec{c}^{\,\mathrm{ph}} = \frac{\omega\vec{k}}{k^2}. \tag{A.38}$$

Again, let us consider random wave fields. For this case, the function $u(\vec{R},t)$ and correspondingly $U(\vec{R},t)$ should be considered as random functions of several variables.

One of the main tools for the characterization of random fields is the correlation function:

$$B(\vec{R},t;\vec{R}',t') = \frac{1}{2}\left\langle U(\vec{R},t)U^*(\vec{R}',t')\right\rangle. \tag{A.39}$$

We note that the dependence of the correlation function on the random field can be called: stationary $(B(\vec{R},\vec{R}';t-t'))$, homogeneous $B(\vec{R}-\vec{R}';t,t')$, or simultaneously stationary and homogeneous $B(\vec{R}-\vec{R}';t-t') = B(\vec{\rho},\tau)$.

For simplicity, let us consider a stationary and homogeneous random wave field. As usual, the complex amplitude can be presented as a decomposition in plane waves – in this case the dispersion law is contained in implicit form in the function $S(\vec{k},\omega)$ in (A.35). The function $S(\vec{k},\omega)$ is random here, and is called the spectral amplitude or spectrum of the random field. For stationary and homogeneous fields,

$$\left\langle S(\vec{k},\omega)S^*(\vec{k}',\omega')\right\rangle = G(\vec{k},\omega)\delta(\omega-\omega')\delta(\vec{k}-\vec{k}'), \tag{A.40}$$

where the function $G(\vec{k},\omega)$ is called the spectral density of the random field. For the correlation function of the random field, we have the relationships:

$$B(\vec{\rho},\tau) = \int d\vec{k}\int\limits_{-\infty}^{\infty} d\omega\, G(\vec{k},\omega)\,\mathrm{e}^{i(\vec{k}\vec{\rho}-\omega\tau)}, \tag{A.41}$$

$$G(\vec{k},\omega) = \frac{1}{(2\pi)^4}\int d\vec{\rho}\int\limits_{-\infty}^{\infty} d\tau\, B(\vec{\rho},\tau)\,\mathrm{e}^{-i(\vec{k}\vec{\rho}-\omega\tau)}. \tag{A.42}$$

In the material that follows, we will use the mathematical concepts developed here for two- or three-dimensional situations. If k and ω are connected by a dispersion law, we will use the notation $G(\vec{k})$ for spectral density.

Surface Waves

Having discussed some general properties of waves and fields, let us delve into specific examples. To begin, let us consider elements of the theory of linear and nonlinear surface waves [a more detailed description can be found in Brekhovskikh

and Goncharov (1994)]. These waves can be derived for a heavy, incompressible fluid from Euler's equations together with the continuity equation. We first introduce a coordinate system to use for the derivation. Let the *XY*-plane coincide with the unperturbed surface of the water, and the *Z*-axis be directed down. Then our two equations are as follows:

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho(\vec{v}\nabla)\vec{v} = -\nabla p + \rho \vec{g}, \tag{A.43}$$

$$\mathrm{div}\vec{v} = 0, \tag{A.44}$$

where ρ is the density of water, p is the pressure, $\vec{v}$ is the particle velocity, and $\vec{g}$ is the vector of gravitational acceleration. Assuming that the unperturbed density is $\rho_0(z)$ and that the corresponding equilibrium pressure distribution is $\rho_0(z)$, so that $\frac{\partial p_0}{\partial z} = \rho_0 g$, we get

$$\rho_0 \frac{\partial \vec{v}}{\partial t} + \rho_0(\vec{v}\nabla)\vec{v} = -\nabla p + \rho \vec{g}. \tag{A.45}$$

We now let $p,\ \rho$ be redefined by including small additions (perturbations) to the equilibrium values of pressure and density: $\rho_0(z) + \rho(x, y, z, t)$, $p_0(z) + p(x, y, z, t)$. In the linear approximation, we then have

$$\rho_0 \frac{\partial \vec{v}}{\partial t} = -\nabla p + \rho \vec{g}, \tag{A.46}$$

$$\mathrm{div}\vec{v} = 0. \tag{A.47}$$

To solve this set of equations, we introduce the horizontal and vertical components of velocity $\vec{v} = (\vec{u}, w)$ and the nabla operators in the horizontal plane $\nabla_- = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$,

$$\nabla = \left(\nabla_-, \frac{\partial}{\partial z}\right). \tag{A.48}$$

Equations (A.43) and (A.44) then take the form

$$\begin{aligned} \rho_0 \frac{\partial \vec{u}}{\partial t} &= -\nabla_- p, \\ \rho_0 \frac{\partial w}{\partial t} &= -\frac{\partial p}{\partial z}, \\ \nabla_- \vec{u} + \frac{\partial w}{\partial z} &= 0. \end{aligned} \tag{A.49}$$

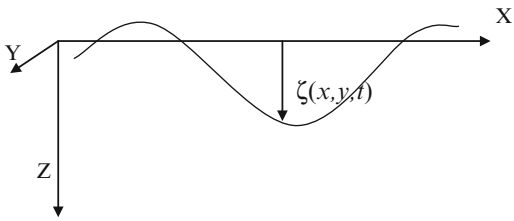


Fig. A.1 Surface waves

Replacing $\vec{u},\ p$ in favor of $w(\vec{r},z,t)$, we get

$$\frac{\partial}{\partial t}\Delta w = 0. \tag{A.50}$$

Let us denote the vertical deviation of the water surface from its equilibrium position (in the XY-plane) as $\zeta(\vec{r},t)$, as per Fig. A.1.

The boundary conditions for the water surface then have the form

$$\frac{d\zeta(\vec{r},t)}{dt} = \frac{\partial\zeta}{\partial t} + \nabla_{-}\zeta\cdot\vec{u} = w\big|_{z=\zeta(\vec{r},t)} \tag{A.51}$$

which in the linear approximation becomes

$$\frac{\partial\zeta(\vec{r},t)}{\partial t} = w|_{z=0}. \tag{A.52}$$

We can also write the so-called dynamic boundary condition, which corresponds to the assumption that the pressure force applied to the surface from both sides should be equal, i.e.,

$$p_{z=\zeta} = 0 \tag{A.53}$$

or

$$p|_{z=0} + \rho_0 g\zeta = 0. \tag{A.54}$$

Eliminating $\vec{u}$ from the first and third equations of (A.49), we get

$$\nabla_{-}^2 p = \rho_0 \frac{\partial^2 w}{\partial z\partial t}. \tag{A.55}$$

In this case, at the boundary

$$\left[\nabla_{-}^2 p + \rho_0 g\nabla_{-}^2\zeta\right]_{z=0} = 0 \tag{A.56}$$

or

$$\left[\frac{\partial^2 w}{\partial z \partial t} + g\nabla_-^2 \zeta\right]_{z=0} = 0. \tag{A.57}$$

After an additional differentiation over t, we finally obtain a second boundary condition [in addition to (A.54)]

$$\left[\frac{\partial^3 w}{\partial z \partial t^2} + g\nabla_-^2 w\right]_{z=0} = 0. \tag{A.58}$$

We also have the obvious "no penetration" boundary condition at the bottom

$$w|_{z=H} = 0. \tag{A.59}$$

A solution of (A.50) with the above boundary conditions and for water depth Hcan be found in the form:

$$w = -i\Omega\Phi(z)\, \mathrm{e}^{i(\tilde{\vec{q}}\vec{r} - \Omega t)}, \tag{A.60}$$

where $\tilde{\vec{q}}$ is the wavevector corresponding to propagation in the horizontal direction and $\Phi(z)$ describes the depth dependence. The factor $-i\Omega$ is added for convenience. Correspondingly, we have

$$\zeta = \Phi(0)\, \mathrm{e}^{i(\tilde{\vec{q}}\vec{r} - \Omega t)}.$$

After the substitution of (A.60) into (A.50) and (A.58), we get

$$\begin{gathered} \Phi'' - \tilde{q}^2\Phi = 0, \\ \left[\Phi + \frac{\Omega^2}{\tilde{q}^2 g}\Phi'\right]_{z=0} = 0, \\ \Phi(H) = 0. \end{gathered} \tag{A.61}$$

We can construct the solution to (A.61) as

$$\Phi(z) = A \sinh\left[\tilde{q}(z - H)\right] \tag{A.62}$$

with the dispersion relation

$$\Omega^2 = \tilde{q} g \tanh \tilde{q}H. \tag{A.63}$$

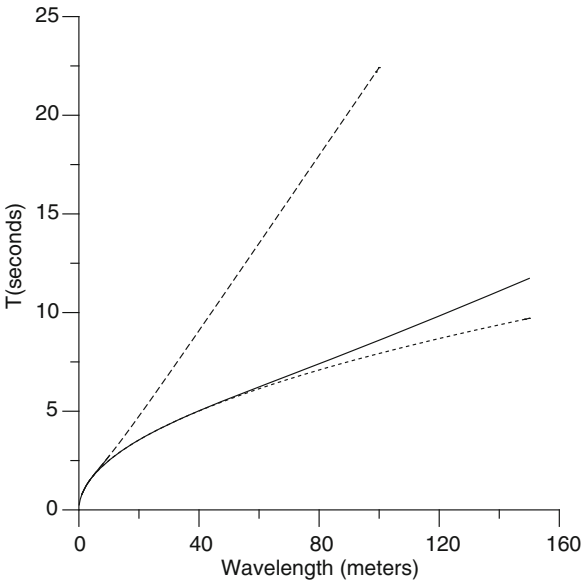


Fig. A.2 Dispersion of surface waves

From this equation, we can obtain the well-known dispersion relations for both deep and shallow water. For $qH \ll 1$, we have shallow water (surface wavelength greater than the depth of the water) and thus $\Omega^2 = \tilde{q}^2 gH$ or $\Omega = \tilde{q}\sqrt{gH}$ which gives an absence of dispersion (i.e., the group velocity is equal to the phase velocity $\tilde{c}^{\,\mathrm{ph}} = \tilde{c}^{\,\mathrm{gr}} = \sqrt{gH}$).

For deep water ($qH \gg 1$), we get the dispersion law

$$\Omega^2 = \tilde{q}g \tag{A.64}$$

and see that the phase velocity is two times greater than the group velocity.

In Fig. A.2 below, we show the dependencies of periods of surface waves as a function of their wave length, corresponding to dispersion curves $\Omega(\tilde{q})$ for several typical parameter values.

For the vertical component of velocity, it is possible to get an expression in the form of a traveling plane wave:

$$w(\vec{r}, z, t) = -i\Omega A \sinh\left[\tilde{q}(z - H)\right] \mathrm{e}^{i(\tilde{\vec{q}}\vec{r} - \Omega t)}. \tag{A.65}$$

The wavevector in the horizontal plane and the frequency are connected by the dispersion (A.63).

For the vertical displacement of the water for a wave propagating in the direction $\tilde{\vec{q}}$

$$\zeta_{\tilde{\vec{q}}\Omega}(\vec{r}, t) = \zeta_{\tilde{\vec{q}}} \mathrm{e}^{i(\tilde{\vec{q}}\vec{r} - \Omega t)}, \tag{A.66}$$

where we have introduced the complex amplitude of the surface wave $\zeta_{\tilde{\vec{q}}} = A \sinh(\tilde{q}H)$.

The second subindex Ω is now omitted because it can be expressed through the wavevector using the dispersion law. In this case, we have for the vertical component of the surface particle velocity

$$w(\vec{r},z,t) = -i\frac{\zeta_{\tilde{\vec{q}}}\Omega}{\sinh(\tilde{q}H)}\sinh\left[\tilde{q}(z-H)\right]\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)}. \tag{A.67}$$

The horizontal component of the particle velocity can also be found as

$$\vec{u}(\vec{r},z,t) = \frac{\tilde{\vec{q}}\zeta_{\tilde{\vec{q}}}\Omega}{\tilde{q}\sinh(\tilde{q}H)}\cosh\left[\tilde{q}(z-H)\right]\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)}. \tag{A.68}$$

The surface density of kinetic energy (averaged over a period of the wave in its direction of propagation in the horizontal plane) is defined by the expression

$$E_{\mathrm{kin}} = \int_0^H \frac{\rho|\vec{v}|^2}{4}dz = \frac{\rho g\left|\zeta_{\tilde{\vec{q}}}\right|^2}{4}. \tag{A.69}$$

The surface density of potential energy is the same, so the total energy density is

$$E = \frac{\rho g\left|\zeta_{\tilde{\vec{q}}}\right|^2}{2}. \tag{A.70}$$

The above theory is for monochromatic waves. For a random set of waves propagating in different directions, as in the real ocean, we need to introduce the surface spectral density. Let a random sea surface be described by the real function$\zeta(\vec{r},t)$. For a fixed time, we can represent it as a decomposition of solutions:

$$\zeta(\vec{r},t) = \int \zeta_{\tilde{\vec{q}}}\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)}d\tilde{\vec{q}}. \tag{A.71}$$

We note that when we integrate over $\tilde{\vec{q}}$, we should use the dispersion law for Ω. We also note that due to $\zeta(\vec{r},t)$ being a real function we have the relationship

$$\zeta_{\tilde{\vec{q}}}\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)} = \zeta^*_{\tilde{\vec{q}}}\mathrm{e}^{-i(\tilde{\vec{q}}\vec{r}-\Omega t)}. \tag{A.72}$$

Let us now examine the correlation function

$$\begin{aligned}\langle\zeta(\vec{r},t)\zeta(\vec{r}',t')\rangle &= \int\int\left\langle\zeta_{\tilde{\vec{q}}}\zeta_{\tilde{\vec{q}}}\right\rangle\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)}\mathrm{e}^{i(\tilde{\vec{q}}'\vec{r}'-\Omega' t')}d\tilde{\vec{q}}d\tilde{\vec{q}}' \\ &= \int\int\left\langle\zeta_{\tilde{\vec{q}}}\zeta^*_{\tilde{\vec{q}}'}\right\rangle\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)}\mathrm{e}^{-i(\tilde{\vec{q}}'\vec{r}'-\dot{\Omega}' t')}d\tilde{\vec{q}}d\tilde{\vec{q}}'\end{aligned} \tag{A.73}$$

Using the standard δ correlation ansatz, i.e. that $\left\langle \zeta_{\vec{\tilde{q}}} \zeta^*_{\vec{\tilde{q}}'} \right\rangle = G(\vec{\tilde{q}})\delta(\vec{\tilde{q}} - \vec{\tilde{q}}')$, we obtain

$$B_\zeta(\vec{r} - \vec{r}', t - t') = \langle \zeta(\vec{r}, t)\zeta(\vec{r}', t') \rangle = \int G(\vec{\tilde{q}})\, \mathrm{e}^{i[\vec{\tilde{q}}(\vec{r}-\vec{r}') - \Omega(t-t')]} d\vec{\tilde{q}}. \tag{A.74}$$

The surface energy density for given set of waves is thus

$$E = \frac{\rho g \langle \zeta^2 \rangle}{2} = \int \frac{\rho g}{2} G(\vec{\tilde{q}}) d\vec{\tilde{q}} = \int_0^\infty \tilde{q} d\tilde{q} \int_{-\pi}^{\pi} d\theta \frac{\rho g}{2} G(\tilde{q}, \theta). \tag{A.75}$$

The value $\frac{\rho g}{2} G(\vec{q})$ can be interpreted as the spectral energy density of the surface waves. The root mean square height of the waves can be found as

$$\bar{\zeta} = \sqrt{\langle \zeta^2 \rangle} = \left[\int_0^\infty \tilde{q} d\tilde{q} \int_{-\pi}^{\pi} d\theta G(\tilde{q}, \theta) \right]^{1/2}. \tag{A.76}$$

To get a simple linear equation describing the propagation of the surface wave (an equation for the function $\zeta(\vec{r}, t)$), a power series decomposition of the dispersion law can be used. This expansion corresponds to a two-dimensional wave propagating along the horizontal plane. If we expand (A.63) in powers of $\tilde{q}H$, we get

$$\Omega = \sqrt{gH}\tilde{q} - \frac{1}{6}\sqrt{gH}H^2\tilde{q}^3 + \cdots \tag{A.77}$$

The corresponding linear equation in one space coordinate (denoted as x) can be obtained using formal substitution $\Omega \to i\frac{\partial}{\partial t}$, $\tilde{q} \to -i\frac{\partial}{\partial x}$, so we have

$$\begin{aligned} &\frac{\partial \zeta}{\partial t} + \tilde{c}^{\,\mathrm{ph}} \frac{\partial \zeta}{\partial x} + \beta \frac{\partial^3 \zeta}{\partial x^3} = 0 \\ &\tilde{c}^{\,\mathrm{ph}} = \sqrt{gH}, \beta = \frac{1}{6}\sqrt{gH}H^2, \end{aligned} \tag{A.78}$$

which is the linear Korteveg de Vries (KdV) equation. It describes propagation of surface waves (and also other types of waves with the same dispersion law) in the x direction and is commonly seen in wave theory.

Nonlinear Corrections for Surface Waves

To get nonlinear corrections and nonlinear equations for surface waves, we will employ some qualitative assertions, similar to those presented in Konyaev and Sabinin (1992). For waves of a small amplitude (the linear approximation) we have

expressions (A.63) and (A.66) for the displacement of the surface (as a complex function of amplitude). The nonlinearity parameter, which determines the accuracy of the linear approximation, is the ratio of the wave particle velocity (the orbital velocity or its horizontal component) to the phase velocity of the surface wave

$$\mu = \frac{u}{\tilde{c}^{\,\mathrm{ph}}}, \tag{A.79}$$

where $u = |\vec{u}(\vec{r}, 0, t)|$. The following relations define the orbital velocity on the surface

$$w(\vec{r}, 0, t) = -i\Omega\zeta_{\vec{\tilde{q}}}(\vec{r}, t), \tag{A.80}$$

$$\vec{u}(\vec{r}, 0, t) = \frac{\vec{\tilde{q}}\Omega}{\tilde{q}\tanh \tilde{q}H}\zeta_{\vec{\tilde{q}}}(\vec{r}, t). \tag{A.81}$$

Nonlinear corrections arise due to taking into account the finite amplitude of oscillations of the surface and the finite orbital velocity. In order to understand this situation better, let us consider the variation of the previously discussed parameters with the depth.

We first consider the example of shallow water waves (long waves). In this case $\tilde{c}^{\,\mathrm{ph}} = \sqrt{gH}$.

Let us now make the following qualitative assertions. Consider that during wave propagation, a wave of finite amplitude changes the water depth, so H is now not a constant, but a variable. Additionally, due to the motion of the water particles, the effective velocity differs from the phase velocity. So we now have a displacement ζ of the water surface, which is changing depth and velocity (remember that in our coordinates, the Z-axis is directed downward and so positive values of ζ decrease the depth). Thus, we can write the variation of velocity as

$$\tilde{c}_{\zeta u} = \sqrt{g(H - \zeta)} - u. \tag{A.82}$$

From (A.81) we can get the approximation

$$u \approx \tilde{c}^{\,\mathrm{ph}}\frac{\zeta}{H}. \tag{A.83}$$

Therefore, for the phase velocity with nonlinear corrections we have

$$\tilde{c} \approx \tilde{c}^{\,\mathrm{ph}}\left(1 - \frac{3\zeta}{2H}\right). \tag{A.84}$$

From (A.84) we can see, for example, that for negative ζ (at the crests) the velocity is greater than for positive values (in the trough). This means that the

nonlinear corrections lead to acceleration of the crests and retardation of the troughs, and will eventually result in the breaking of the wave. The difference in the velocities of the crest and the trough is

$$\Delta\tilde{c} = -3\tilde{c}^{\,\mathrm{ph}}\zeta/H. \tag{A.85}$$

Using this equation, it is possible to estimate when nonlinear corrections will be significant, which can be characterized as when the crest will reach the trough in one period:

$$t\Delta\tilde{c} \sim \lambda/2 = \tilde{c}^{\,\mathrm{ph}}T/2,$$

where T is the period of the wave.

Substituting $\Delta\tilde{c}$, we get

$$t \sim \frac{HT}{6\zeta}. \tag{A.86}$$

If take into account the one-dimensional propagation equation for $\zeta(x,t)$:

$$\frac{\partial\zeta}{\partial t} + \tilde{c}\frac{\partial\zeta}{\partial x} = 0, \tag{A.87}$$

then after substituting the velocity from (A.84), we get

$$\frac{\partial\zeta}{\partial t} + \tilde{c}^{\,\mathrm{ph}}\frac{\partial\zeta}{\partial x} - \frac{3\tilde{c}^{\,\mathrm{ph}}}{2H}\zeta\frac{\partial\zeta}{\partial x} = 0. \tag{A.88}$$

Equation (A.88) is a well-known nonlinear equation, and its solution can be given by the Riemann expression (Brekhovskikh and Goncharov 1994).

We next consider dispersion corrections to the nonlinear equation. In the shallow water approximation, surface waves are nondispersive. We can take into account dispersion corrections using (A.63) and (A.84) and combine these with the nonlinear corrections. In this case, instead of (A.84) for phase velocity we have

$$\tilde{c}_{\tilde{q}\zeta} = \sqrt{\frac{g}{\tilde{q}}\tanh\left[\tilde{q}(H-\zeta)\right]}\left(1-\frac{\zeta}{H}\right) \approx \tilde{c}^{\,\mathrm{ph}}\left(1-\frac{3\zeta}{2H}-\frac{\tilde{q}^2H^2}{6}\right). \tag{A.89}$$

Previously, we examined nonlinear corrections due to the variation of depth. Now we also have a correction to the phase velocity due to the wave number (the last term) providing dispersion. Here again we should examine the variation of the local wave number at the crest and trough of the wave. Due to the dispersion law we have as a first approximation

$$\frac{\partial \zeta}{\partial x} = i\tilde{q}\zeta, \quad \frac{\partial^3 \zeta}{\partial x^3} = -i\tilde{q}^3\zeta, \tag{A.90}$$

from which we can get an estimate for the local wave number

$$\tilde{q} \approx \sqrt{-\frac{\partial^3 \zeta}{\partial x^3} \Big/ \frac{\partial \zeta}{\partial x}}. \tag{A.91}$$

If we use the expression in (A.89) for the phase velocity, then for the wave equation, we get

$$\frac{\partial \zeta}{\partial t} + \tilde{c}^{\,\mathrm{ph}} \frac{\partial \zeta}{\partial x} + \frac{3\tilde{c}^{\,\mathrm{ph}}}{2H} \zeta \frac{\partial \zeta}{\partial x} - \frac{\tilde{c}^{\,\mathrm{ph}} H^2}{6} \frac{\partial^3 \zeta}{\partial x^3} = 0. \tag{A.92}$$

This equation is a well-known nonlinear KdV equation.

A well-known solution to this equation is the so-called soliton or solitary wave (KdV soliton)

$$\zeta(x,t) = -\frac{2H}{3\tilde{c}^{\,\mathrm{ph}}} \frac{a}{\cosh^2\left(\frac{x-\tilde{c}t}{\Lambda}\right)}, \tag{A.93}$$

where $\tilde{c} = \tilde{c}^{\,\mathrm{ph}} + a/3$, $\Lambda = H\sqrt{2\tilde{c}^{\,\mathrm{ph}}/a}$.

In accordance with our choice of the Z-direction, the amplitude of the soliton is positive, which means that a surface wave soliton looks like a hill, propagating with a speed a little greater than the normal phase speed of surface waves c_{p}.

Internal Waves

Internal waves are one of the most important factors influencing sound propagation in the ocean. Their existence is supported by the depth dependence of the water column properties (temperature, salinity, and density), and caused by a large number of ocean dynamic processes. Here we will consider an elementary theoretical description of internal waves in shallow water. Again, the basic equations are Euler's equations, augmented by the continuity equation and the equation of state. To simplify things, we assume that water is incompressible and we also neglect the rotation of the Earth. The horizontal plane is described by the X- and Y-axes, and the Z-axis is directed down. We then have:

$$\rho\left[\frac{\partial \vec{v}}{\partial t} + (\vec{v}\nabla)\vec{v}\right] = -\nabla p + \rho g \nabla z, \tag{A.94}$$

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{v}) = 0, \tag{A.95}$$

$$\nabla \vec{v} = 0. \tag{A.96}$$

For consistency, we use the same notation as for surface waves.

Equilibrium pressure (unperturbed) values are determined by the equation

$$\frac{dp_0}{dz} = \rho_0(z) g, \tag{A.97}$$

where the density is an increasing function of depth for stable stratification.

Imposing the linear approximation and dividing all values into their horizontal and vertical components, we have

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\nabla_{\_} p, \tag{A.98}$$

$$\rho_0 \frac{\partial w}{\partial t} = -\frac{\partial p}{\partial z} + \rho g, \tag{A.99}$$

where $\nabla_{\_} = (\partial/\partial x, \partial/\partial y)$ is the gradient in the horizontal plane. Combining the continuity equation

$$\frac{\partial \rho}{\partial t} + \rho_0 \nabla_{\_} \vec{u} + \rho_0 \frac{\partial w}{\partial z} + w \frac{\partial \rho_0}{\partial z} = 0, \tag{A.100}$$

and the incompressibility equation

$$\frac{\partial \rho}{\partial t} + w \frac{\partial \rho_0}{\partial z} = 0, \tag{A.101}$$

we see that

$$\frac{\partial w}{\partial z} = -\nabla_{\_} \vec{u}. \tag{A.102}$$

If we exclude the horizontal components using (A.100) and (A.102), we have

$$\rho_0 \frac{\partial^2 w}{\partial t \partial z} = \nabla^2_{\_} p. \tag{A.103}$$

The preceding leads to an equation for the function $w(\vec{r}, z, t)$:

$$\frac{\partial^2}{\partial t^2} \nabla^2 w + \tilde{N}^2 \nabla^2_{\_} w = 0, \tag{A.104}$$

where $\tilde{N} = \sqrt{\frac{g}{\rho_0}\frac{d\rho_0}{dz}}$ is the so-called Brunt–Vaisala frequency (or buoyancy frequency). This quantity, which describes the frequency at which a parcel of water displaced from its equilibrium position would oscillate about that position, is one of the most important characteristics of stratified water. A typical value for real coastal conditions is several cycles per hour (as is discussed in more detail later). The boundary condition on the sea surface is determined by the equation for $\zeta(x, y, 0, t)$ and looks the same as for the surface waves, (A.57)

$$\left[\frac{\partial^3 w}{\partial^2 t \partial z} + g\nabla_-^2 w\right]_{z=0} = 0. \tag{A.105}$$

The boundary condition at the bottom is again the simple no-penetration condition

$$w|_{z=H} = 0. \tag{A.106}$$

Our equation for w is sufficient for our present purposes, since the remaining variables of interest (the horizontal component of velocity, the density, the vertical displacement, etc.) can be expressed through w.

Now let us consider a wave with harmonic dependence on the time and horizontal coordinates similar to (A.60):

$$w = -i\Omega\Phi(z)\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)}. \tag{A.107}$$

So for the function $\Phi(z)$, we get the equation

$$\frac{d^2\Phi}{dz^2} + \tilde{\vec{q}}^2\left(\frac{\tilde{N}^2}{\Omega^2} - 1\right)\Phi = 0 \tag{A.108}$$

with boundary conditions

$$\Phi|_{z=H} = 0 \tag{A.109}$$

$$\left[\frac{d\Phi}{dz} + \frac{g\tilde{\vec{q}}^2}{\Omega^2}\Phi\right]_{z=0} = 0. \tag{A.110}$$

Equation (A.108) with boundary conditions (A.109) and (A.110) constitutes a standard eigenvalue problem. The functions $\Phi_{l\vec{\kappa}}(z)$ and the corresponding eigenvalues (which depend on frequency) $\tilde{\vec{q}}_l(\Omega)$ and $\Omega_l(\vec{q})$ describe the so-called gravitational modes of a waveguide with internal waves (see Fig. A.3). Depending upon the situation, we will use or omit sub-indexes of the function Φ. From (A.108), we can also see that the character of the function Φ depends upon the relation

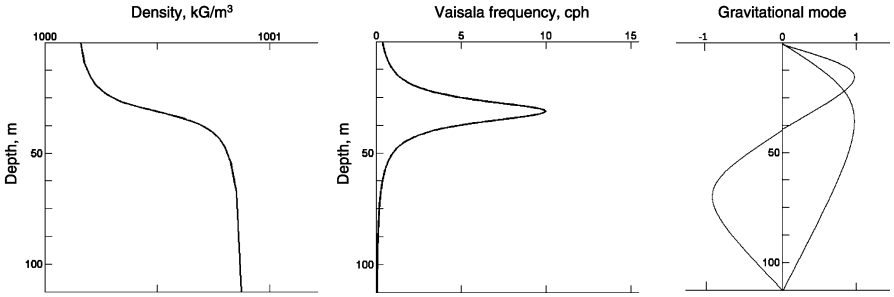


Fig. A.3 Example of typical depth dependence of density (*left panel*), the corresponding Brunt–Vaisala frequency (*middle panel*) and the two lowest gravitational modes (*right panel*) for internal waves

between the frequency of oscillation Ω and the Vaisala frequency $\tilde{N}(z)$; specifically, in regime where $\Omega > \tilde{N}$, we observe an oscillating behavior of $\Phi(z)$, whereas in the regime where $\Omega < \tilde{N}$, $\Phi(z)$ changes monotonically.

It is easy to demonstrate the orthogonality condition

$$-\Phi_m(0)\Phi_l(0) + \frac{1}{g}\int_0^H \tilde{N}^2(z)\Phi_m(z)\Phi_l(z)dz = 0 \quad \text{for } m \neq l \tag{A.111}$$

which can be normalized to unity[1]:

$$-\Phi_m(0)\Phi_l(0) + \frac{1}{g}\int_0^H \tilde{N}^2(z)\Phi_m(z)\Phi_l(z)dz = \delta_{ml}. \tag{A.112}$$

Using the so-called rigid lid approximation ($\Phi(0) = 0$), we get the more usual and simple normalization condition

$$\frac{1}{g}\int_0^H \tilde{N}^2(z)\Phi_m(z)\Phi_l(z)dz = \delta_{ml}. \tag{A.113}$$

The normalized functions above are dimensionless. An example of these mode functions for a given frequency, buoyancy frequency, and density profile is shown in Fig. A.3. It is also possible to introduce another normalization, where the maximum value of any eigenfunction is taken as unity. Again, all values of

[1] In this case, the relation between the gravitational modes and velocity differs from (A.107).

interesting physical quantities can be expressed through the vertical component of the velocity, i.e.,

$$\begin{aligned}
\zeta(\vec{r},z,t) &= \Phi(z)\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)},\\
\rho(\vec{r},z,t) &= -\frac{\rho_0}{g}N^2\Phi(z)\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)},\\
\vec{u}(\vec{r},z,t) &= \frac{\Omega\tilde{\vec{q}}}{\tilde{q}^2}\frac{d\Phi}{dz}\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)},\\
p(\vec{r},z,t) &= \frac{\Omega^2\rho_0}{\tilde{q}^2}\frac{d\Phi}{dz}\mathrm{e}^{i(\tilde{\vec{q}}\vec{r}-\Omega t)}.
\end{aligned} \tag{A.114}$$

Next, let us find the energy carried by internal waves. The kinetic energy, or more exactly the surface density of the energy per unit area over a "squared wavelength" (along x, $\lambda_x = \frac{2\pi}{\tilde{q}_x}$, and along y, $\lambda_y = \frac{2\pi}{\tilde{q}_y}$) is determined by the expression

$$E_k = \frac{\rho_0}{4}\int_0^H |\vec{v}|^2 dz. \tag{A.115}$$

In terms of modes, this is

$$E_k = \frac{\rho_0}{4}\int_0^H \left(\Phi^2 + \frac{1}{\tilde{q}^2}(\Phi')^2\right)dz. \tag{A.116}$$

After integrating by parts and using the boundary conditions, we get

$$E_k = \frac{\rho_0}{4\Omega^2}\int_0^H \tilde{N}^2\Phi^2 dz. \tag{A.117}$$

The potential energy in a narrow depth layer is determined by the work needed to displace a parcel of water from level $z-\zeta$ to level z:

$$dE_\mathrm{p} = g\int_{z-\zeta}^{z}[\rho_0(\xi)-\rho_0(z-\zeta)]d\xi = g\int_0^\zeta \frac{\mathrm{d}\rho_0}{dz}\alpha d\alpha = \rho_0\tilde{N}^2\frac{\zeta^2}{2}. \tag{A.118}$$

Correspondingly, using complex amplitudes gives us

$$dE_{\mathrm{p}} = \frac{\rho_0 \tilde{N}^2}{4} \Phi^2(z). \tag{A.119}$$

The total potential energy of the water column (or more exactly the surface density of the potential energy) in the rigid lid approximation is

$$E_{\mathrm{p}} = \int_0^H \frac{\rho_0 \tilde{N}^2}{4} \Phi^2(z) dz. \tag{A.120}$$

The first (lowest mode) term corresponds to the energy density of surface waves. The total energy density in the rigid lid approximation has the form

$$E_{\mathrm{t}} = \frac{\rho_0}{2} \int_0^H \tilde{N}^2 \Phi^2 dz \tag{A.121}$$

or using the displacement of an isopycnal surface

$$E_{\mathrm{t}} = \frac{\rho_0}{2} \int_0^H \tilde{N}^2 |\zeta(\vec{r}, z, t)|^2 dz. \tag{A.122}$$

If the displacement function under the integral is in the form of a plane wave with frequency Ω, then this expression can be interpreted as the spectral density of energy (or energy spectrum).

We will now look at two classic, simple cases for internal waves in a shallow water waveguide: the constant Brunt–Vaisala frequency case and the two-layered system.

Case of Constant Brunt–Vaisala Frequency

This case corresponds to an exponential depth dependence of the water density $\rho_0(z) = \rho_0 \exp(\alpha z)$. A realistic value of alpha is $\alpha \sim 10^{-5}$ m^{-1}, which corresponds to a buoyancy frequency of $N \sim 5$ cycles/h. Let the depth of the waveguide be a typical shallow water value of 100 m. For the simple boundary conditions $\Phi(0) = \Phi(H) = 0$, the gravitational modes are easily found to be

$$\Phi_l(z) = \zeta_0 \sin \frac{l\pi}{H} z, \tag{A.123}$$

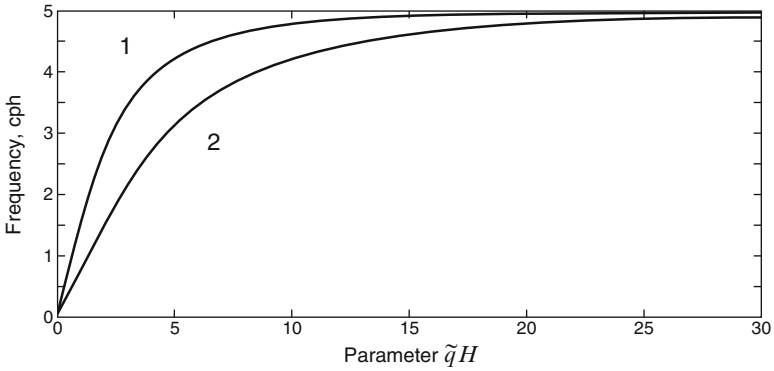


Fig. A.4 Dispersion curves for the two lowest modes (numbers in the figure) of internal waves in a layer with constant Vaisala frequency $\tilde{N} = 5\,\text{cph}$

$$\zeta(x,z,t) = \zeta_0 \sin\left(\frac{l\pi}{H}z\right) \exp[\mathrm{i}(\tilde{q}x - \Omega t)]. \tag{A.124}$$

For a frequency normalized amplitude of the internal wave $\zeta_0 = \frac{A}{\omega}$, the dispersion law is found to be:

$$\Omega_l(\tilde{q}) = \frac{N}{\sqrt{1 + \left(\frac{l\pi}{\tilde{q}H}\right)^2}}. \tag{A.125}$$

In Fig. A.4, we show this dispersion law, specifically the phase and group velocities as a function of the nondimensional parameter $\tilde{q}H$.

Two-Layered Water Column

A physically common situation in shallow water is to have the water column consist (to a good first approximation of the three-layered model with a sharp thermocline) of two layers with different temperature (corresponding to different densities and sound speeds). For simplicity, we consider only one coordinate in the horizontal plane. As a result of the internal waves, the interface between the two layers oscillates, and the displacement from the equilibrium state (plane) is the function $\zeta(x,t)$. Let us consider that the two layers have thickness and densities H_1, ρ_1, H_2, $\rho_2 (\rho_2 > \rho_1)$, as is shown in the Fig. A.5. To construct internal waves in this medium, we should solve (A.108) in both layers and find $\Phi(z)$ where

$$\Phi = \begin{cases} \Phi_1, & 0 < z < H_1, \\ \Phi_2, & H_1 < z < H_1 + H_2. \end{cases} \tag{A.126}$$

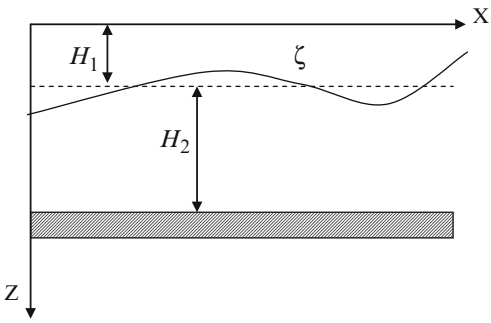


Fig. A.5 Internal waves in two-layered liquid

and then glue these solutions together at the interface. To do this, we use the continuity of pressure and the vertical components of velocity. Analogous to the condition from (A.110) on the surface, we get

$$\rho_1\left[\frac{d\Phi_1}{dz}+\frac{g\tilde{q}^2}{\Omega^2}\Phi_1\right]_{z=H_1}=\rho_2\left[\frac{d\Phi_2}{dz}+\frac{g\tilde{q}^2}{\Omega^2}\Phi_2\right]_{z=H_1}. \tag{A.127}$$

For the upper layer solution, satisfying the boundary condition at the surface gives us the form

$$\Phi_1=A\left(\cosh\,\tilde{q}z-\frac{g\tilde{q}}{\Omega^2}\sinh\,\tilde{q}z\right). \tag{A.128}$$

In the second layer, satisfying the bottom boundary condition gives

$$\Phi_2=B\sinh\,\tilde{q}(z-H_1-H_2). \tag{A.129}$$

The continuity conditions then give us the equation

$$B=A\frac{\cosh\tilde{q}H_1}{\sinh\,\tilde{q}H_2}\left(1-\frac{g\tilde{q}}{\Omega^2}\tanh\,\tilde{q}H_1\right) \tag{A.130}$$

and correspondingly we get the dispersion equation

$$\frac{\rho_1}{\rho_2}\tanh\tilde{q}H_2\left(\frac{g\tilde{q}}{\Omega^2}-\tanh\tilde{q}H_1\right)=\left(1-\frac{g\tilde{q}}{\Omega^2}\tanh\tilde{q}H_1\right)\left(1-\frac{g\tilde{q}\Delta\rho}{\Omega^2\rho_2}\tanh\tilde{q}H_2\right), \tag{A.131}$$

where $\Delta\rho=\rho_2-\rho_1$. It can be seen that only one vertical (gravitational) mode exists in this waveguide. This model can be applied for typical shallow water conditions where $\Delta\rho\sim 1-3\,\mathrm{kg/m^3}$, so $\Delta\rho/\rho\ll 1$. Taking into account that $g\tilde{q}/\Omega^2\sim 10^3-10^4\gg 1$, we have

$$B = -A\frac{g\tilde{q}}{\Omega^2}\frac{H_1}{H_2}.$$

It is now possible to write the dispersion equation in the simpler form

$$\tanh\tilde{q}H_2 + \tanh\tilde{q}H_1\left(1 - \frac{g\tilde{q}\Delta\rho}{\Omega^2\rho_2}\tanh\tilde{q}H_2\right) = 0. \tag{A.132}$$

Introducing the "reduced" gravitational acceleration $g' = \frac{g\Delta\rho}{\rho_2}$ for the density jump, $g' \sim 1 - 3 \times 10^{-2}\,\mathrm{m/s^2}$. Using this notation, we express the phase velocity of the internal wave as

$$c_{\tilde{q}} = \sqrt{\frac{g'\tanh\tilde{q}H_1\tanh\tilde{q}H_2}{\tilde{q}(\tanh\tilde{q}H_2 + \tanh\tilde{q}H_1)}}. \tag{A.133}$$

Let us consider long waves where $\tilde{q}H_{1,2} \ll 1$, so that for the phase velocity we have

$$c_0 = \sqrt{\frac{g'H_1H_2}{H_1 + H_2}}. \tag{A.134}$$

For realistic conditions, this gives $c_0 \sim 0.5 - 0.7\,\mathrm{m/s}$.

As was mentioned previously, for internal waves nonlinear corrections can often be significant. To get equations and solutions in the nonlinear case, we can use a semiqualitative approach, following the approach presented in the book (Konyaev and Sabinin 1992) (similar to the derivation of nonlinear corrections for the surface waves). This means that we will use (A.133) and include in it corrections to the particle velocities and displacements, which are added to the unperturbed values in the dispersion equations. In other words, we should add (and subtract) displacement of interface ζ to values H_1 (and H_2) and add corrections (orbital velocities) to phase velocity due to the same displacement. The amplitude of the "orbital" velocity is

$$U = \frac{\Omega}{\tilde{q}}\frac{d\Phi}{dz} = \tilde{c}^{\,\mathrm{ph}}\frac{d\zeta}{dz}. \tag{A.135}$$

We know that both water layers give a contribution to the corrections to phase velocity, and they have different signs. If we take roughly that $d\zeta/dz \sim \zeta/H$, then we thus can write

$$U_{1,2} = \frac{\Omega}{\tilde{q}}\frac{d\Phi_{1,2}}{dz} \sim \tilde{c}^{\,\mathrm{ph}}\frac{\zeta}{H_{1,2}}. \tag{A.136}$$

Let us now take into account the nonlinear corrections in the dispersion equation.

Using this approach, we get a KdV equation for the displacement function for the case of one spatial coordinate. Previously, we introduced the function $\zeta = \zeta(\vec{r}, z, t)$. Let us take the displacement of a water layer at a fixed depth, for example where the function $\Phi(z)$has a maximum $\Phi(z_0) = \zeta_0$. Next, suppose that the internal waves propagate in the X-direction, having wave front parallel to the Y-axis. In this case we have displacement function $\zeta = \zeta(x, t)$ for which we will derive the corresponding equation.

We additionally introduce the idea of a local wavevector $\tilde{q}^2 = -\frac{\partial^3\zeta/\partial x^3}{\partial\zeta/\partial x}$. Now let us write the expression for the horizontal velocity of the displacement function taking into account the corrections, which are due to the additional thicknesses of the layers due to displacement and the additional velocity due to the changed particle velocities in both layers. If we put these corrections into the dispersion law, we get:

$$\begin{aligned}\tilde{c}_{\tilde{q}} &= \sqrt{\frac{g' \tanh\tilde{q}(H_1+\zeta)\tanh\tilde{q}(H_2-\zeta)}{\tilde{q}[\tanh\tilde{q}(H_1+\zeta)+\tanh\tilde{q}(H_2-\zeta)]}} + \tilde{c}^{\,\mathrm{ph}}\frac{\zeta}{H_1} - \tilde{c}^{\,\mathrm{ph}}\frac{\zeta}{H_2} \\ &\approx \tilde{c}^{\,\mathrm{ph}}\left(1 + \zeta\frac{3\Delta H}{2H_1H_2} - \tilde{q}^2\frac{H_1H_2}{6}\right),\end{aligned} \tag{A.137}$$

where $\Delta H = H_2 - H_1$.

Using that

$$\frac{d\zeta}{dt} = \frac{\partial\zeta}{\partial t} + \tilde{c}_{\tilde{q}}\frac{\partial\zeta}{\partial x}, \tag{A.138}$$

we get a nonlinear KdV equation for the two water layers

$$\frac{\partial\zeta}{\partial t} + \tilde{c}^{\,\mathrm{ph}}\frac{\partial\zeta}{\partial x} + \tilde{c}^{\,\mathrm{ph}}\frac{3\Delta H}{2H_1H_2}\zeta\frac{\partial\zeta}{\partial x} + \tilde{c}^{\,\mathrm{ph}}\frac{H_1H_2}{6}\frac{\partial^3\zeta}{\partial x^3} = 0. \tag{A.139}$$

This equation has solutions in the form of both solitary waves [solitons, see (A.140)] and cnoidal waves (Apel et al. 1995).

$$\zeta(x,t) = \tilde{c}^{\,\mathrm{ph}}\frac{3\Delta H}{2H_1H_2}\frac{a}{\cosh^2\left(\frac{x-\tilde{c}t}{\Lambda}\right)}, \tag{A.140}$$

where $\tilde{c} = \tilde{c}^{\,\mathrm{ph}}\left(1 + \frac{\Delta H\zeta_0}{2H_1H_2}\right)$ is the velocity of soliton and $\zeta_0 = \frac{3\Delta Ha}{2H_1H_2}\tilde{c}^{\,\mathrm{ph}}$ is the amplitude of soliton. We see that these values are connected with each other.

This solution has an interesting dependence on the sign of the term $\zeta\Delta H$. In particular, for solitons where $\zeta\Delta H>0$, the maximum is directed downward (left panel in Fig. A.6), whereas for $\zeta\Delta H<0$, the maximum is directed upward (right panel in Fig. A.6).

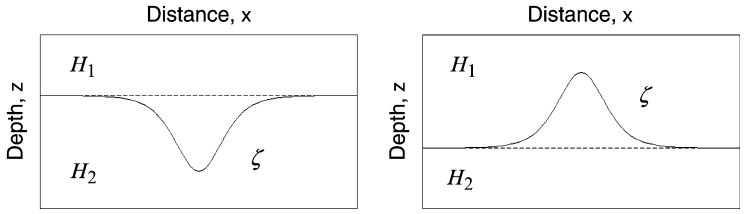


Fig. A.6 Displacement of thermocline layer for the KdV soliton for different depths of thermocline. *Dashed lines* denote position of unperturbed thermocline

Internal Wave Spectra

Like surface waves, the real internal wave field is generally constituted of a random mixture of waves propagating in different directions. For a coastal soliton field, one often sees very directional nonlinear waves superposed on a background of (often less directional) linear internal waves (waves of comparatively small amplitude). Mathematically, this situation can be described by a complex function of the displacement of the isopycnal surface:

$$\zeta(\vec{r}, z, t) = \sum_l \int d\tilde{\vec{q}} S_l(\tilde{\vec{q}}) \Phi_{l\vec{\kappa}}(z) \mathrm{e}^{i(\tilde{\vec{q}}\vec{r} - \Omega t)}, \tag{A.141}$$

where $S_l(\tilde{\vec{q}})$, the amplitudes of internal waves, also depend on frequency through dispersion law $\Omega(\tilde{\vec{q}})$. For a stationary homogeneous field

$$\left\langle S_l(\tilde{\vec{q}}) S^*_{l'}(\tilde{\vec{q}}') \right\rangle = G^{\mathrm{s}}_l(\tilde{\vec{q}}) \delta(\tilde{\vec{q}} - \tilde{\vec{q}}') \delta_{ll'}. \tag{A.142}$$

If we consider the internal waves in the "frozen approximation" (very useful for acoustics), we can also exclude time from the expression for the internal waves.

The value $G_l(\tilde{\vec{q}})$ is generally called the spectral density of the internal waves or simply the spectrum of the internal waves.

Acoustic Waves

In ocean acoustics, the prime variable of interest is the sound pressure, considered as a function of the spatial coordinates and time. In our work, the real component of sound pressure is denoted as $p(\vec{R}, t)$. The equation for acoustic waves, or more precisely for the sound pressure as a function of the space–time coordinates, can be obtained by using Euler's equations and taking into account the compressibility of media. In the linear approximation, these equations have the form

$$\rho_0 \frac{\partial \vec{v}}{\partial t} = -\nabla p, \tag{A.143}$$

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{v}) = 0, \tag{A.144}$$

$$\frac{\partial \rho}{\partial t} = \frac{1}{c^2} \frac{\partial p}{\partial t}. \tag{A.145}$$

The last equation is the equation of state, where the value of $c^2 = \frac{\partial p}{\partial \rho}$ determines the sound speed. If we now divide the sound pressure and density in the ordinary way into a sum of equilibrium and perturbation parts, and then take into account that the unperturbed density can depend on the coordinates, we obtain the wave equation for inhomogeneous media

$$\nabla^2 p - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} p - \frac{1}{\rho_0} \nabla \rho_0 \nabla p = 0. \tag{A.146}$$

The last term as a rule is small and can be ignored, and so for sound pressure we obtain the ordinary form of the wave equation.

The next thing we need to add to our description is a source of sound. Mathematically, this is included by inserting a source function on the right side of equation. The sound field created by the source then satisfies the inhomogenous wave equation

$$\left[\nabla^2 - \frac{1}{c^2} \frac{\partial^2}{\partial t^2}\right] p(\vec{R}, t) = -q(\vec{R}, t). \tag{A.147}$$

The function $q(\vec{R}, t)$ on the right side of the equation describing the source, according to dimensional arguments, should be called the real volume density of the radiated signal. On the left-hand side, the sound speed is usually a function of the spatial coordinates and time, $c(\vec{R}, t)$. The sound speed in water is practically nondispersive; i.e., it does not depend on frequency to a very good first approximation. The dispersion that is observed in our case is mainly due to the boundary conditions creating a finite waveguide, and is called "waveguide dispersion." As a rule, we will consider so-called "point sources" in our work. Mathematically, this means sources with sizes much smaller than a wavelength, which are placed at the point $\vec{R}_0$. Physically, this comes from the fact that at low frequencies, a wavelength is on the order of several meters, and in practice making sources this size or larger is both very expensive and unwieldy. For the point source case, we can write $q(\vec{R}, t) = f(t)\delta(\vec{R} - \vec{R}_0)$. We will call the function $f(t)$ the real radiated signal. As a simple example, consider a pulsating sphere, characterized by volume $V(t)$, in a homogeneous medium with density ρ. For this case, the real radiated signal is

$f(t) = \rho \partial V/\partial t$. We call this type of source a monopole source, and the maximum value (amplitude) of the volume velocity $\partial V/\partial t$ is called the strength of the source.

More computational power can (as usual) be gained by extending our analysis to the complex domain. In a homogeneous medium, we can write an equation for the analytic pressure field using an analytic function of the volume density of the radiated signal $Q(\vec{R}, t)$

$$\left[\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right] P(\vec{R}, t) = -Q(\vec{R}, t). \tag{A.148}$$

Solutions of (A.148) can be constructed using a space–time Green function, which satisfies the equation

$$\left[\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right] \Psi(\vec{R}, \vec{R}'; t, t') = -\delta(\vec{R} - \vec{R}')\delta(t - t'). \tag{A.149}$$

The explicit expression for the Green function in a homogeneous medium is of the form

$$\Psi(\vec{R}, \vec{R}'; t, t') = \frac{1}{4\pi\left|\vec{R} - \vec{R}'\right|}\delta\left(t - t' - \frac{\left|\vec{R} - \vec{R}'\right|}{c}\right). \tag{A.150}$$

The solution of (A.148) thus has the form

$$P(\vec{R}, t) = \int d\vec{R}' dt' \Psi(\vec{R}, \vec{R}'; t, t') Q(\vec{R}', t') = \int d\vec{R}' \frac{Q\left(\vec{R}', t - \frac{\left|\vec{R} - \vec{R}'\right|}{c}\right)}{4\pi\left|\vec{R} - \vec{R}'\right|}. \tag{A.151}$$

In particular, for a monopole source placed at the point $\vec{R}_0$ and oscillating in some arbitrary form

$$Q(\vec{R}, t) = F(t)\delta(\vec{R} - \vec{R}_0) \tag{A.152}$$

the complex pressure field is

$$P(\vec{R}, t) = \frac{F\left(t - \frac{\left|\vec{R} - \vec{R}_0\right|}{c}\right)}{4\pi\left|\vec{R} - \vec{R}_0\right|}. \tag{A.153}$$

If $F(t) = \mathrm{e}^{-i\omega t}$, then $P(\vec{R}, t) = \dfrac{\mathrm{e}^{i(k\left|\vec{R} - \vec{R}_0\right| - \omega t)}}{4\pi\left|\vec{R} - \vec{R}_0\right|}$. In a similar way we can consider a dipole source (for example a pair of pulsating spheres, placed at some distance from

each other) with dipole moment $\vec{M}(t)$, and placed at the point $\vec{R}_0$. For this source, the volume density of the radiated signal is

$$Q(\vec{R},t) = \rho \frac{\partial \vec{M}(t)}{\partial t} \nabla \delta(\vec{R} - \vec{R}_0). \tag{A.154}$$

Correspondingly, the complex sound field has the form

$$P(\vec{R},t) = \rho \nabla \frac{\frac{\partial}{\partial t}\vec{M}\left(t - \frac{|\vec{R}-\vec{R}_0|}{c}\right)}{4\pi|\vec{R} - \vec{R}_0|}. \tag{A.155}$$

As an example, if the dipole moment is directed along the z axis and depends upon time harmonically, i.e., $M(t) = M_z(t) = M\mathrm{e}^{-i\omega t}$, then

$$Q(\vec{R},t) = -i\omega\rho M \mathrm{e}^{-i\omega t} \frac{\partial}{\partial z} \delta(\vec{R} - \vec{R}_0). \tag{A.156}$$

Thus in free space we have

$$P(\vec{R},t) = -i\omega\rho M \mathrm{e}^{-i\omega t} \frac{\partial}{\partial z} \frac{\mathrm{e}^{ik|\vec{R}-\vec{R}_0|}}{4\pi|\vec{R} - \vec{R}_0|}. \tag{A.157}$$

In inhomogeneous media, we have that $c = c(\vec{R}), \rho = \rho(\vec{R})$. As before, the sound field of the source is determined by (A.148) or (A.149). The wave equation with the corresponding boundary conditions now cannot be solved simply for an arbitrary source. To deal with this case, we should first of all examine a spectral component of the sound field, i.e., the sound field for a sinusoidal time dependence of the source function. In other words, we introduce a spectral function $S(\vec{R},\omega)$ and an analytic field $P(\vec{R},t)$ depending on the space coordinates

$$S(\vec{R},\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} p(\vec{R},t)\mathrm{e}^{i\omega t} dt, \tag{A.158}$$

$$P(\vec{R},t) = 2 \int_{0}^{\infty} S(\vec{R},\omega)\mathrm{e}^{-i\omega t} d\omega, \tag{A.159}$$

$$p(\vec{R},t) = \mathrm{Re}\, P(\vec{R},t). \tag{A.160}$$

The equation for $P(\vec{R},t)$ has the form (A.137) with a complex radiated signal $F(t)$ on the right-hand side. For the spectral component $S(\vec{R},\omega)$ as function of space coordinates, we get

$$\left[\nabla^2+\frac{\omega^2}{c^2}\right]S(\vec{R},\omega)=-S(\omega)\delta(\vec{R}-\vec{R}_0). \tag{A.161}$$

If we define the Green function (transfer function) of the medium $\Psi(\vec{R},\vec{R}_0;\omega)$ as the solution of the equation

$$\left[\nabla^2+\frac{\omega^2}{c^2}\right]\Psi(\vec{R},\vec{R}_0;\omega)=-\delta(\vec{R}-\vec{R}_0) \tag{A.162}$$

then the spectral function and the complex sound field are derived from the equations

$$S(\vec{R},\omega)=\Psi(\vec{R},\vec{R}_0;\omega)S(\omega), \tag{A.163}$$

$$P(\vec{R},t)=2\int_0^\infty\Psi(\vec{R},\vec{R}_0;\omega)S(\omega)\mathrm{e}^{-i\omega t}d\omega. \tag{A.164}$$

The real field is then obtained as

$$p(\vec{R},t)=\int_{-\infty}^{\infty}\Psi(\vec{R},\vec{R}_0;\omega)S(\omega)\mathrm{e}^{-i\omega t}d\omega. \tag{A.165}$$

For a complex sound field depending in an arbitrary manner on the spatial coordinates (and on the radiated signal), we introduce the spatial wavevector spectrum or simply the spatial spectrum via the equation

$$\begin{aligned}P(\vec{R})&=\int S(\vec{k})\mathrm{e}^{i\vec{k}\vec{R}}d\vec{k},\\ S(\vec{k})&=\frac{1}{(2\pi)^3}\int P(\vec{R})\mathrm{e}^{-i\vec{k}\vec{R}}d\vec{R}.\end{aligned} \tag{A.166}$$

We can also introduce the spatial spectrum in the horizontal plane

$$\begin{aligned}P(\vec{R})=P(\vec{r},z)&=\int S(\vec{\xi},z)\mathrm{e}^{i\vec{\xi}\vec{r}}d\vec{\xi},\\ S(\vec{\xi},z)&=\frac{1}{(2\pi)^2}\int P(\vec{r},z)\mathrm{e}^{-i\vec{\xi}\vec{r}}d\vec{r},\end{aligned} \tag{A.167}$$

which unlike the former spectrum will be denoted by a vector in the horizontal plane, $\vec{\xi}$.

Finally we combine results and introduce the space–time spectrum

$$P(\vec{R},t) = 2\int_0^\infty d\omega \int d\vec{k} S(\vec{k},\omega) \mathrm{e}^{i(\vec{k}\vec{R}-\omega t)},$$

$$S(\vec{k},\omega) = \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} dt \int d\vec{R} \mathrm{Re}[P(\vec{R},t)] \mathrm{e}^{-i(\vec{k}\vec{R}-\omega t)}. \tag{A.168}$$

As we know from the cases of monochromatic and locally quasiplane fields, to compute energy-related quantities (and in general those quantities quadratic in the field), one can use complex signals, without transferring to their real parts. In particular, the energy density and intensity of the sound field are determined by

$$E = \frac{1}{\rho c^2}\left\langle p^2 \right\rangle = \frac{1}{2\rho c^2}|P|^2, \tag{A.169}$$

$$\vec{I} = \langle p\vec{v} \rangle = \frac{1}{4}\mathrm{Re}(P\vec{V}^*), \tag{A.170}$$

where the angle brackets denote averaging over time and $\vec{V}$ is the complex velocity.

We can express intensity only through sound pressure if use (A.143) which for monochromatic sound or for separate Fourier components gives

$$\vec{I} = \frac{i}{4\rho\omega}(P\nabla P^* - P^*\nabla P). \tag{A.171}$$

As was mentioned, for a locally plane wave (where $|\nabla P| \sim k|P|$), the expression (A.160) can be simplified. Thus for the value of intensity we get

$$I = \frac{1}{2\rho c}|P|^2. \tag{A.172}$$

For a broadband sound field where P is given by expression (A.164), we can obtain the intensity for a fixed frequency (spectral intensity) of the source with the spectrum$S(\omega)$:

$$\vec{I}_\omega = \frac{2\pi i}{\rho\omega}|S(\omega)|^2(\Psi\nabla\Psi^* - \Psi^*\nabla\Psi). \tag{A.173}$$

The main energy-related characteristic of a source is the radiated power, which is the flux density of power (intensity) generated by the field $p(\vec{R},t)$. To examine this

quantity, we integrate over a (generally spherical) surface surrounding the source and also, in the case of a quasimonochromatic emitted signal, average over a time greater than the period of the sound field. This gives[2]

$$W_0 = \frac{\langle p^2 \rangle}{\rho c} 4\pi \left| \vec{R} - \vec{R}_0 \right|^2 = \frac{\langle f^2(t) \rangle}{4\pi \rho c}. \tag{A.174}$$

For quasimonochromatic signals, this approximately corresponds to averaging over a whole pulse length $T = 2\pi/\Delta\omega$, which gives

$$\langle f^2(t) \rangle \approx \frac{1}{T} \int_{-\infty}^{\infty} f^2(t) dt = \Delta\omega \int_{-\infty}^{\infty} |S(\omega)|^2 d\omega = 2\Delta\omega \int_{0}^{\infty} |S(\omega)|^2 d\omega. \tag{A.175}$$

If we deal with short pulses (broadband signals), then instead of averaging over time we can integrate the instantaneous intensity over the total duration of the pulse. So instead of the intensity of the emitted (or received) signal, which is measured in energy/per unit of square meter per unit of time, we have the total flux of the pulse (energy per unit of square meter):

$$\vec{F} = \int_{-\infty}^{\infty} p \vec{v} dt. \tag{A.176}$$

Correspondingly for the value of the total flux per pulse, we have an expression for the locally plane wave sound pressure

$$F = \int_{-\infty}^{\infty} \frac{1}{\rho c} p^2(t) dt. \tag{A.177}$$

The sound source is characterized by the total energy radiated per pulse:

$$E_0 = \frac{1}{4\pi \rho c} \int_{-\infty}^{\infty} f^2(t) dt = \frac{1}{2\pi \rho c} \int_{0}^{\infty} |S(\omega)|^2 d\omega. \tag{A.178}$$

It is also possible to introduce the spectral energy radiated per pulse as

$$E_{0\omega} = \frac{|S(\omega)|^2}{2\rho c}. \tag{A.179}$$

[2] The power of sound normally transmitted for low-frequency acoustic shallow water acoustics is on the order of 10^1–10^3 W.

Appendix B
Modal Decomposition of the Sound Field from a Point Source (Green Function)

Here we construct the Green function of the waveguide, determined by the general (A.162). This field can be expressed as the solution of the inhomogeneous equation (omitting argument $\vec{R}_0$ and ω)

$$[\nabla^2 + k^2(z)]\Psi(\vec{R}) = -\delta(\vec{R} - \vec{R}_0), \tag{B.1}$$

satisfying the radiation condition at infinity. Here $\vec{R}_0(0, 0, z_0)$ is the radius vector of the source, and $k(z) = \frac{2\pi f}{c(z)}$. We can also consider this problem when the right-hand side of (B.1) is equal to zero. In this case, i.e., the homogeneous equation for the function $\Psi(\vec{R})$ as $\vec{R} \to \vec{R}_0$, we have the limit:

$$\Psi(\vec{R}) \to \Psi_0(\vec{R}) = \frac{\exp(ik(z_0)\left|\vec{R} - \vec{R}_0\right|)}{4\pi\left|\vec{R} - \vec{R}_0\right|}. \tag{B.2}$$

Both problems are part of the to construction of the Green function for the wave equation, corresponding to an outgoing wave. This problem additionally has the boundary conditions

$$\Psi|_{z=0} = 0 \tag{B.3}$$

determined by the pressure-release surface, and the conditions of continuity of the pressure and of the vertical component of the particle velocity across the transition from the water layer to the bottom

$$\Psi|_{z\to H-} = \Psi|_{z\to H+} \frac{1}{\rho}\frac{\partial\Psi}{\partial z}|_{z\to H-} = \frac{1}{\rho}\frac{\partial\Psi}{\partial z}|_{z\to H+}, \tag{B.4a}$$

where the notation $z \to H-$ $(z \to H+)$ denotes the limit as z tends to the boundary from the water and from the bottom layer, respectively. Sometimes,

this condition is written as the continuity of the "logarithmic derivative" $\frac{1}{\rho\Psi}\frac{\partial\Psi}{\partial z}$ on transition through the boundary, i.e.,

$$\frac{1}{\rho\Psi}\frac{\partial\Psi}{\partial z}|_{z\to H-} = \frac{1}{\rho\Psi}\frac{\partial\Psi}{\partial z}|_{z\to H+}. \tag{B.4b}$$

Given cylindrical symmetry, (B.1) can be rewritten in cylindrical coordinates independent of the azimuthal angle ϑ. In this case, we obtain for $\Psi(r,z)$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\Psi}{\partial r}\right)+\frac{\partial^2\Psi}{\partial z^2}+k^2(z)\Psi = -\delta(z-z_0)\frac{\delta(r)}{2\pi r}. \tag{B.5}$$

The right-hand side of (B.5) can be easily obtained from the δ-function on the right-hand side of (B.1) as follows:

$$\begin{aligned}\delta(\vec{R}-\vec{R}_0) &= \delta(z-z_0)\frac{1}{(2\pi)^2}\int_{-\infty}^{\infty}dk_x\int_{-\infty}^{\infty}dk_y\exp(ik_x x+ik_y y)\\ &= \delta(z-z_0)\frac{1}{(2\pi)^2}\int_0^{\infty}\xi d\xi\int_0^{2\pi}d\vartheta'\exp[i\xi r\cos(\vartheta'-\vartheta)]\\ &= \delta(z-z_0)\frac{1}{2\pi} = \int_0^{\infty}J_0(\xi r)\xi d\xi = \delta(z-z_0)\frac{\delta(r)}{2\pi r},\end{aligned} \tag{B.6}$$

where $\xi=\sqrt{k_x^2+k_y^2}$ and $J_0(\xi r)$ is the cylindrical Bessel function of order O. No angle is involved, since $\vec{r}_0=0$. Let us now construct a solution of (B.5) by using the Hankel transform. For the Hankel transform, $\bar{\Psi}(\xi,z)$ is related to the original function $\Psi(r,z)$ by the equations:

$$\bar{\Psi}(\xi,z) = \frac{1}{2\pi}\int_0^{\infty}J_0(\xi r)\Psi(r,z)rdr, \tag{B.7a}$$

$$\Psi(r,z) = 2\pi\int_0^{\infty}J_0(\xi r)\bar{\Psi}(\xi,z)\xi d\xi. \tag{B.7b}$$

Multiplying both sides of (B.5) by $(2\pi)^{-1}J_0(\xi r)r$ and integrating over r, we obtain

$$2\pi\int_0^{\infty}\frac{\partial}{\partial r}\left(r\frac{\partial\Psi(r,z)}{\partial r}\right)J_0(\xi r)dr+\frac{\partial^2\bar{\Psi}(\xi,z)}{\partial z^2}+k^2(z)\bar{\Psi}(\xi,z) = -\frac{1}{(2\pi)^2}\delta(z-z_0). \tag{B.8}$$

Evaluating the integral by parts, we obtain[3]

$$\frac{d^2\bar{\Psi}(\xi,z)}{dz^2}+[k^2(z)-\xi^2]\bar{\Psi}(\xi,z)=-\frac{1}{4\pi^2}\delta(z-z_0) \tag{B.9}$$

with boundary condition on the surface:

$$\bar{\Psi}(\xi,z)|_{z=0}=0. \tag{B.10}$$

We shall define the bottom boundary condition via (B.4a) based on properties of the bottom and the fluid, homogeneous, absorptive half-space. The solution of (B.9) for $z>H$

$$\frac{d^2\bar{\Psi}(\xi,z)}{dz^2}+[k_1^2-\xi^2]\bar{\Psi}(\xi,z)=0 \tag{B.11}$$

satisfying the radiation condition at infinity, and modulo a constant factor, has the form

$$\bar{\Psi}(\xi,z)=\mathrm{e}^{\mathrm{i}\sqrt{k_1^2-\xi^2}z}. \tag{B.12}$$

Using (B.4b), to obtain the boundary condition for the function $\bar{\Psi}(\xi,z)$, we rewrite it in the form

$$\left[\bar{\Psi}+g(\xi)\frac{d\bar{\Psi}}{dz}\right]_{z=H}=0, \tag{B.13a}$$

$$g(\xi)=\frac{m_1}{\sqrt{\xi^2-k_1^2}}, \tag{B.13b}$$

where $m_1=\frac{\rho_1}{\rho}$. The function $g(\xi)$ is determined by the properties of the bottom[4] and can be computed even for complex models of the bottom, e.g., taking into account shear waves, multilayered structure, etc. Thus, we come down to constructing the solution, over the range $0\le z\le H$, of the problem:

$$\frac{d^2\bar{\Psi}(\xi,z)}{dz^2}+[k^2(z)-\xi^2]\bar{\Psi}(\xi,z)=-\frac{1}{4\pi^2}\delta(z-z_0). \tag{B.14a}$$

$$\bar{\Psi}(\xi,z)|_{z=0}=0, \tag{B.14b}$$

[3] Generally speaking, we should write $\bar{\Psi}(\xi,z,z_0)$; however, when it is not necessary we shall omit the second argument.

[4] The function $g(\xi)$ is related to the impedance of the bottom by the equation $g(\xi)=\frac{i}{\omega\rho}Z_1$, where the impedance is defined according to Brekhovskikh (1980); the change in sign from Brekhovskikh (1980) is due to the direction of the z-axis.

$$\left[\bar{\Psi} + g(\xi)\frac{d\bar{\Psi}}{dz}\right]_{z=H} = 0. \tag{B.14c}$$

One standard way of constructing this solution (the one-dimensional Green function) involves finding two linearly independent solutions of the homogeneous equation corresponding to (B.14a–c), $V(\xi, z)$ and $U(\xi, z)$, satisfying, respectively, the first and the second boundary conditions:

$$\frac{d^2U(\xi,z)}{dz^2} + [k^2(z) - \xi^2]U(\xi,z) = 0, \tag{B.15a}$$

$$\left[U + g(\xi)\frac{\mathrm{d}U}{dz}\right]_{z=H} = 0, \tag{B.15b}$$

and

$$\frac{\mathrm{d}^2V(\xi,z)}{dz^2} + [k^2(z) - \xi^2]V(\xi,z) = 0, \tag{B.16a}$$

$$V(\xi,z)|_{z=0}. \tag{B.16b}$$

For this case, the solution of the problem (B.14a–c) has the form:

$$\bar{\Psi}(\xi,z) = \frac{U(\xi,z_>)V(\xi,z_<)}{4\pi^2 W(\xi)}, \tag{B.17}$$

where $z_< = \min(z, z_0), z_> = \max(z, z_0)$, and $W(\xi)$ is the Wronskian of the system of solutions

$$W(\xi) = \begin{vmatrix} U(\xi,z) & V(\xi,z) \\ U'(\xi,z) & V'(\xi,z) \end{vmatrix}, \tag{B.18}$$

which does not depend on z and therefore can be computed at any point. For our development, it will be convenient to compute it at the point $z = H$:

$$W(\xi) = -U'(\xi,H)[V(\xi,H) + g(\xi)V'(\xi,H)]. \tag{B.19}$$

Here, the prime denotes the derivative with respect to z. An expression for the sound field at the observation point can be obtained by substitution of (B.17) into (B.7b) only using a Hankel function instead of the Bessel function:

$$\begin{aligned} J_0(\xi r) &= \frac{1}{2}\left[H_0^{(1)}(\xi r) + H_0^{(2)}(\xi r)\right], \\ H_0^{(2)}(\xi r) &= -H_0^{(1)}(\mathrm{e}^{\mathrm{i}\pi}\xi r). \end{aligned} \tag{B.20}$$

Doing this we obtain

$$\Psi(r,z) = \frac{1}{4\pi} \int\limits_{-\infty}^{\infty} \frac{U(\xi, z_>)V(\xi, z_<)}{W(\xi)} H_0^{(1)}(\xi r)\xi d\xi. \tag{B.21}$$

The integral is computed as a path integral in the complex plane, with a proper circuit a round the poles and branch points of the integrand.

The poles ξ_l are simply determined by the zeros of the Wronskian (B.19); in other words, by the expression:

$$V(\xi_l, H) + g(\xi)V'(\xi_l, H) = 0. \tag{B.22}$$

In addition to the contribution of the poles to the integral (B.21), we must take into account branch points of the integrand, due to the presence of the root in the expression for $g(\xi)$ and, correspondingly, in the boundary conditions for the solutions. These branch points are at $\xi = \pm k_1$. Since the lower medium is assumed to be absorptive, the branch points lie in the complex plane, in the first and third quadrants, respectively, near the real axis. When the contour is closed in the upper half-plane, we must remain on the "physical" sheet of the Riemann surface, determined by forcing the decay of the waves in the lower half-space. Let us consider the properties of the function $\kappa(\xi) = \sqrt{k_1^2 - \xi^2}$ in the complex plane in more detail, using the assumption that the value of k_1 is complex ((3.1)). Additionally, let us define two angles (arguments) for the quantities $k_1 + \xi = |k_1 + \xi|e^{i\beta}$, and $k_1 - \xi = |k_1 - \xi|e^{i\gamma}$ (see Fig. B.1). Positive values of the arguments β and γ are taken to be in the anticlockwise direction with respect to the horizontal axes, beginning at the points $-k_1$ and k_1, respectively. For the angle $\beta = 0$ the direction of the horizontal axis coincides with the positive x-axis, while for the angle γ the corresponding direction coincides with the negative x-axis. The physical sheet will be determined by the direction of the coordinate axes and the condition that the field attenuates at infinity: therefore we draw the section with the lines $\mathrm{Im}\sqrt{k_1^2 - \xi^2} = 0$, which gives the equation $\mathrm{Re}\,\xi \cdot \mathrm{Im}\,\xi = \frac{\alpha k^2 n_1^2}{2}$, defining part of a hyperbola, beginning at the branch points $\xi = \pm k_1 = \pm k n_1\left(1 + i\frac{\alpha}{2}\right)$. These parts of the branches (shown in Fig. B.1 by dots) are found to the left of the point $\xi = k_1$ and to the right of the point $\xi = -k_1$. In the remaining part of the hyperbola (shown by dashes) $\mathrm{Re}\sqrt{k_1^2 - \xi^2} = 0$. In order to determine the value of the function in the upper half-plane on the right- and left-hand portions of the section, let us indicate the values of the angles β and γ in this region. Namely, on the right-hand part of this region (where the point ξ_1 is shown) $0<\beta<\pi/2$, $\pi<\gamma<2\pi$, i.e., $\pi/2<(\beta+\gamma)/2<5\pi/4$, which gives negative values for the real part of the function $\kappa(\xi)$ such that $\mathrm{Re}\sqrt{k_1^2 - \xi^2} = \left|k_1^2 - \xi^2\right|^{1/2} \cos\frac{\beta+\gamma}{2} < 0$. This means that for integration directly along the section line, where the function $\kappa(\xi)$ is real, on the right-hand

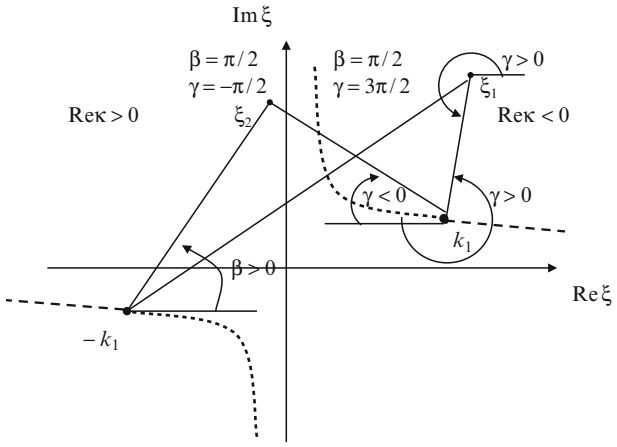


Fig. B.1 Complex plane for integration

side of the section, we must take the root with the minus sign. Correspondingly, on the left-hand side of the section (where the point ξ_2 is shown), we have values $0<\beta<\pi/2$, $-\pi/2<\gamma<0$ and $\mathrm{Re}\sqrt{k_1^2-\xi^2} = \left|k_1^2-\xi^2\right|^{1/2}\cos\frac{\beta+\gamma}{2}>0$, which implies a positive sign for the root.

Thus, for the sound field, we obtain an expression in the form of a sum of contributions of residues at the poles and an integral along the section line (branch line B), which we shall not yet make explicit, i.e.,

$$\Psi(r,z) = \frac{i}{2}\sum_l \frac{U(\xi_l,z_>)V(\xi_l,z_<)}{W'(\xi_l)}H_0^{(1)}(\xi_l r)\xi_l - \frac{1}{4\pi}\int_B \frac{U(\xi,z_>)V(\xi,z_<)}{W(\xi)}H_0^{(1)}(\xi r)\xi d\xi.$$

The contribution of the broader line integral corresponds to the so-called continuous spectrum of the field of a point source. Its role at comparatively large distances is small, and in our subsequent analysis we shall often neglect it. However, in some situations, the contribution of the continuous spectrum must still be taken into account; we refer the reader to a few of the many papers on this (Brekhovskikh and Godin 1992; Gazarian 1958; Stickler and Ammicht 1980). We shall touch upon the continuous spectrum once again seen in considering the Pekeris model, and also later on in considering a field of distributed sources (noise).

Let us now consider the discrete spectrum of the field in more detail. Though we are neglecting the continuous spectrum, we will still retain the same notation Ψ for the field, but now corresponding to the discrete spectrum. Calculating the residues at the poles, we obtain

$$\Psi(r,z) = \frac{i}{2}\sum_l \frac{U(\xi_l,z_>)V(\xi_l,z_<)}{W'(\xi_l)}H_0^{(1)}(\xi_l r)\xi_l. \tag{B.23}$$

Expression (B.23) can be transformed somewhat, taking into account the fact that when $\xi = \xi_l$ the functions $U(\xi,z)$ and $V(\xi,z)$ become linearly dependent, since

the Wronskian becomes zero. Let us suppose that $U(\xi_l, z) = AV(\xi_l, z)$. The equation for the functions U, V then becomes an equation in the eigenvalues

$$\frac{d^2\Psi_l(z)}{dz^2} + \left[k^2(z) - \xi_l^2\right]\Psi_l(z) = 0, \tag{B.24a}$$

$$\Psi_l(z)|_{z=0} = 0, \tag{B.24b}$$

$$\left[\Psi_l(z) + g(\xi_l)\frac{d\Psi_l(z)}{dz}\right]_{z=H} = 0, \tag{B.24c}$$

where we have denoted $\Psi_l(z) = V(\xi_l, z)$. Equation (B.24a) with boundary conditions (B.24b) and (B.24c) determines a Sturm–Liouville type equation problem, where $\Psi_l(z)$ and ξ_l are the eigenfunctions and eigenvalues, respectively. Using (B.15a, b) and (B.24a–c), we can obtain a simple equation for an arbitrary Wronskian $W'(\xi_l)$. Specifically, we multiply (B.15a) and (B.24a), respectively, by $\Psi_l(z)$ and $U(\xi, z)$; then, subtracting one equation from the other, we integrate both sides of the equation from zero to H, taking into account the gradient conditions, and then obtain

$$U(\xi, H)\Psi_l(H)\left[\frac{1}{g(\xi_l)} - \frac{1}{g(\xi)}\right] + U(\xi, 0)\Psi_l'(0) = (\xi^2 - \xi_l^2)\int_0^H U(\xi, z)\Psi_l(z)dz. \tag{B.25}$$

Using the expression for the Wronskian at the point $z = 0$:

$$W(\xi) = U(\xi, 0)V'(\xi, 0) \tag{B.26}$$

we obtain

$$\frac{W(\xi)}{\xi - \xi_l} = \frac{V'(\xi, 0)}{\Psi'_l(0)}\left[(\xi + \xi_l)\int_0^H U(\xi, z)\Psi_l(z)dz + U(\xi, H)\Psi_l(H)\frac{g^{-1}(\xi) - g^{-1}(\xi_l)}{\xi - \xi_l}\right] \tag{B.27}$$

In the limit $\xi \to \xi_l$ we obtain derivatives on the left-hand side of the equation and in the last term on the right-hand side. Computing the derivative of the variable $g^{-1}(\xi)$ at the point $\xi = \xi_l$, we obtain

$$W'(\xi_l) = 2A\xi_l\left[\int_0^H \Psi_l^2(z)dz + \frac{{\Psi_l}^2(H)}{2\xi_l}\left(\frac{dg^{-1}(\xi)}{d\xi}\right)_{\xi=\xi_l}\right]$$

$$= 2A\xi_l \left[\int_0^H \Psi_l^2(z)dz + \frac{\Psi_l^2(H)}{2m\sqrt{\xi_l^2 - k_1^2}} \right]. \tag{B.28}$$

Substituting this expression in the equation for the sound field (B.23), we finally obtain

$$\Psi(r,z) = \frac{i}{4} \sum_l \frac{\Psi_l(z_0)\Psi_l(z)}{\int_0^H \Psi_l^2(z)dz + \frac{\Psi_l^2(H)}{2m\sqrt{\xi_l^2 - k_1^2}}} H_0^{(1)}(\xi_l r). \tag{B.29}$$

If we invoke the normalization of the eigenfunctions of the Sturm–Liouville problem (B.24a–c), i.e.,

$$\int_0^H \Psi_l^2(z)dz + \frac{\Psi_l^2(H)}{2m\sqrt{\xi_l^2 - k_1^2}} = 1, \tag{B.30}$$

then for the field of a point source we have the decomposition:

$$\Psi(r,z) = \frac{i}{4} \sum_l \Psi_l(z_0)\Psi_l(z) H_0^{(1)}(\xi_l r). \tag{B.31}$$

The meaning of the second term on the left-hand side of (B.30) is easy to understand, if one includes the half-space $z>H$ in the search for the eigenfunctions. In this case, the solution in this region, satisfying the boundary conditions, will have the form $\Psi_l(z) = \Psi_l(H)\mathrm{e}^{-\sqrt{\xi_l^2 - k_1^2}(z-H)}$, and the normalization condition (B.31) becomes the condition:

$$\int_0^H \Psi_l^2(z)dz + \frac{\rho}{\rho_1} \int_H^\infty \Psi_l^2(z)dz = 1. \tag{B.32}$$

Similarly, we obtain the orthogonality condition for the functions $\Psi_l(z)$: if $m \neq l$ then

$$\int_0^H \Psi_m(z)\Psi_l(z)dz + \int_H^\infty \frac{\rho}{\rho_1} \Psi_m(z)\Psi_l(z)dz = 0 \tag{B.33}$$

or

$$\int_0^H \Psi_m(z)\Psi_l(z)dz = -\frac{\Psi_m(H)\Psi_l(H)}{m(\sqrt{\xi_m^2 - k_1^2} + \sqrt{\xi_l^2 - k_1^2})}. \tag{B.34}$$

Appendix C
Mode Coupling Equations

In this appendix, we derive the mode coupling equations presented in Chap. 3. For this goal, we should solve (3.102):

$$\frac{\partial^2\phi}{\partial z^2}+\frac{\partial^2\phi}{\partial r^2}+k^2(r,z)\phi=0 \tag{C.1}$$

containing perturbations of the waveguide parameters on the background of a regular waveguide: surface $s(r)$, depth $H+h(r)$, sound speed profile $c(r,z)=c(z)+\delta c(r,z)$, and correspondingly $k^2(r,z)=k^2(z)+\delta k^2(r,z)$, where $|h(r)|,|s(r)|\ll H,|\delta c|\ll c$, and $|h'|\ll 1$.

We also have the boundary conditions

$$\phi\big|_{z=s(r)}=0, \tag{C.2}$$

$$\phi|_{z=H+h-}=\phi|_{z=H+h+}, \tag{C.3}$$

$$\frac{1}{\rho}\frac{\partial\phi}{\partial n}\big|_{z=H+h-}=\frac{1}{\rho_1}\frac{\partial\phi}{\partial n}\big|_{z=H+h+}, \tag{C.4}$$

where $\vec{n}$ is normal to the bottom surface (see Fig. 3.13). Solution of (C.1) can be found in the form of a series over the eigenfunctions of the regular waveguide $\Psi_n(z)$, or in the region $0<z<H$

$$\phi(r,z)=\sum_n a_n(r)\Psi_n(z), \tag{C.5}$$

where

$$\frac{d^2\Psi_n}{dz^2}+[k^2(z)-\xi_n^2]\Psi_n=0 \tag{C.6}$$

with boundary conditions

$$\Psi_n(0) = 0, \tag{C.7}$$

$$\Psi_n^- = \Psi_n^+,$$
$$\frac{1}{\rho}\frac{\partial \Psi_n^-}{\partial z} = \frac{1}{\rho_1}\frac{\partial \Psi_n^+}{\partial z},$$

where we use the notation $F^{\pm} = F|_{z=H\pm}$.

It should be noted that for the functions Ψ_n and for the reduced sound pressure ϕ we have different boundary conditions at the bottom – in (C.7) we have a derivative with respect to z, whereas the boundary conditions (C.7) contain derivatives with respect to the normal to the bottom.

The eigenfunctions are normalized as:

$$\int_0^{\infty} \frac{\rho}{\rho_1}\Psi_n(z)\Psi_m(z)dz = \delta_{nm}. \tag{C.8}$$

For consistency, we should connect the normal derivative with the z-derivative. We can write for normal derivative

$$\frac{\partial}{\partial n} = n_r\frac{\partial}{\partial r} + n_z\frac{\partial}{\partial z} \approx h'\frac{\partial}{\partial r} + \frac{\partial}{\partial z}. \tag{C.9}$$

Next, to take into account the small variations of depth, we should write the boundary conditions for the unperturbed value of depth:

$$\phi(H+) + \frac{\partial \phi}{\partial z}(H+)h = \phi(H-) + \frac{\partial \phi}{\partial z}(H-)h \tag{C.10}$$

or

$$\phi^+ + \frac{\partial \phi^+}{\partial z}h = \phi^- + \frac{\partial \phi^-}{\partial z}h \tag{C.11}$$

or

$$\phi^- - \phi^+ = h\left(\frac{\partial \phi^+}{\partial z} - \frac{\partial \phi^-}{\partial z}\right). \tag{C.12}$$

The second boundary condition has the form

$$\frac{1}{\rho}\left[h'\frac{\partial\phi^-}{\partial r}+\frac{\partial\phi^-}{\partial z}\right]+\frac{1}{\rho}\left[h'\frac{\partial^2\phi^-}{\partial z\partial r}+\frac{\partial^2\phi^-}{\partial z^2}\right]h$$
$$=\frac{1}{\rho_1}\left[h'\frac{\partial\phi^+}{\partial r}+\frac{\partial\phi^+}{\partial z}\right]+\frac{1}{\rho_1}\left[h'\frac{\partial^2\phi^+}{\partial z\partial r}+\frac{\partial^2\phi^+}{\partial z^2}\right]h. \tag{C.13}$$

Taking into account small parameter h,

$$\frac{1}{\rho}\left[h'\frac{\partial\varphi^-}{\partial r}+\frac{\partial\varphi^-}{\partial z}+\frac{\partial^2\varphi^-}{\partial z^2}h\right]=\frac{1}{\rho_1}\left[h'\frac{\partial\varphi^+}{\partial r}+\frac{\partial\varphi^+}{\partial z}+\frac{\partial^2\varphi^+}{\partial z^2}h\right] \tag{C.14}$$

or

$$\frac{\partial\phi^-}{\partial z}-\eta\frac{\partial\phi^+}{\partial z}=-h'\left(\frac{\partial\phi^-}{\partial r}-\eta\frac{\partial\phi^+}{\partial r}\right)-h\left(\frac{\partial^2\phi^-}{\partial z^2}-\eta\frac{\partial^2\phi^+}{\partial z^2}\right). \tag{C.15}$$

So we now have new boundary conditions, (C.12) and (C.15).

Also, we should transform the boundary condition on the upper surface. For this case

$$\phi\big|_{z=s(r)}=\phi\Big|_{z=0}+\frac{\partial\phi}{\partial z}|_{z=0}s(r)=\phi|_{z=0}+\frac{\partial\phi^0}{\partial z}s(r), \tag{C.16}$$

where $\phi^0=\phi|_{z=0}$. So the boundary condition has the form

$$\phi|_{z=0}=-\frac{\partial\phi^0}{\partial z}s(r). \tag{C.17}$$

We can now consider sound propagation in the presence of combined small perturbation: surface and bottom roughness and volume inhomogeneities. They are described by functions $s(r),h(r)\ll H,\delta k\ll k$. We now assume that $H=\text{const}$, $k=k(z)$, so that

$$\frac{\partial^2\phi}{\partial z^2}+\frac{\partial^2\phi}{\partial r^2}+(k^2+\delta k^2)\phi=0 \tag{C.18}$$

with the boundary conditions (C.2–C.4). These better can be further reduced to the surfaces $z=0$ and $z=H$.

Let us now multiply equation (C.1) by $\frac{\rho}{\rho(z)}\Psi_m$ and integrate over z.

In this way, we get a system of integral equations for the reduced sound field $\phi(r,z)$ and also a system of coupled equations for the modal amolitudes

$$\int_0^\infty \frac{\rho}{\rho(z)}\Psi_m\frac{\partial^2\phi}{\partial z^2}dz = \int_0^H \Psi_m\frac{\partial^2\phi}{\partial z^2}dz + \int_H^\infty \frac{\rho}{\rho_1}\Psi_m\frac{\partial^2\phi}{\partial z^2}dz$$

$$= \Psi_m^-\frac{\partial\varphi}{\partial z}\Big|_0^H - \int_0^H \frac{\partial\Psi_m}{\partial z}\frac{\partial\phi}{\partial z}dz + \frac{\rho}{\rho_1}\Psi_m^+\frac{\partial\phi}{\partial z}\Big|_H^\infty - \int_H^\infty \frac{\rho}{\rho_1}\frac{\partial\Psi_m}{\partial z}\frac{\partial\phi}{\partial z}dz$$

$$= \Psi_m^-\left[\frac{\partial\phi^-}{\partial z} - \frac{\rho}{\rho_1}\frac{\partial\phi^+}{\partial z}\right] - \frac{\partial\Psi_m}{\partial z}\phi\Big|_0^H + \int_0^H \frac{\partial^2\Psi_m}{\partial z^2}\phi dz$$

$$-\frac{\rho}{\rho_1}\frac{\partial\Psi_m^+}{\partial z}\phi\Big|_H^\infty + \int_H^\infty \frac{\rho}{\rho_1}\frac{\partial^2\Psi_m}{\partial z^2}\phi dz$$

$$= \Psi_m^-\left[\frac{\partial\phi^-}{\partial z} - \frac{\rho}{\rho_1}\frac{\partial\phi^+}{\partial z}\right] - \frac{\partial\Psi_m^-}{\partial z}\phi^- + \frac{\partial\Psi_m^0}{\partial z}\phi^0 + \frac{\rho}{\rho_1}\frac{\partial\Psi_m^+}{\partial z}\phi^+$$

$$+\int_0^H [\xi_m^2 - k^2(r,z)]\Psi_m(z)\phi dz + \int_H^\infty \frac{\rho}{\rho_1}[\xi_m^2 - k_1^2]\Psi_m(z)\phi dz$$

$$= \Psi_m^-\left[\frac{\partial\phi^-}{\partial z} - \frac{\rho}{\rho_1}\frac{\partial\phi^+}{\partial z}\right] - \frac{\partial\Psi_m^-}{\partial z}(\phi^- - \phi^+) - \frac{\partial\Psi_m^0}{\partial z}\frac{\partial\phi^0}{\partial z}s$$

$$+\int_0^\infty \frac{\rho}{\rho(z)}[\xi_m^2 - k^2(r,z)]\Psi_m(z)\phi dz$$

$$= \left\{h'\left[\frac{\rho}{\rho_1}\frac{\partial\phi^+}{\partial z} - \frac{\partial\phi^-}{\partial z}\right] + h\left[\frac{\rho}{\rho_1}\frac{\partial^2\phi^+}{\partial z^2} - \frac{\partial^2\phi^-}{\partial z^2}\right]\right\}\Psi_m^-$$

$$+h\frac{\partial\Psi_m^-}{\partial z}\left[\frac{\partial\phi^+}{\partial r} - \frac{\partial\phi^-}{\partial r}\right] - \frac{\partial\Psi_m^0}{\partial z}\frac{\partial\phi^0}{\partial z}s$$

$$+\int_0^\infty \frac{\rho}{\rho(z)}[\xi_m^2 - k^2(r,z)]\Psi_m(z)\phi dz.$$

The wave equation (C.1) gives (C.19)

$$\left\{h'\left[\eta\frac{\partial\phi^+}{\partial r} - \frac{\partial\phi^-}{\partial r}\right] + h\left[\eta\frac{\partial^2\phi^+}{\partial z^2} - \frac{\partial^2\phi^-}{\partial z^2}\right]\right\}\Psi_m^-$$

$$+h\frac{\partial\Psi_m^-}{\partial z}\left[\frac{\partial\phi^+}{\partial z} - \frac{\partial\phi^-}{\partial z}\right] - \frac{\partial\Psi_m^0}{\partial z}\frac{\partial\phi^0}{\partial z}s$$

$$+\int_0^\infty \frac{\rho}{\rho(z)}\delta k^2(r,z)\phi\Psi_m dz \int_0^\infty \frac{\rho}{\rho(z)}[\xi_m^2\phi + \frac{\partial^2\phi}{\partial r^2}]\Psi_m(z)dz = 0. \quad \text{(C.20)}$$

Equation (C.20) is system of integral equations, equivalent to our equation with boundary conditions (C.2–C.4). We will find the solution of this equation as a series of functions Ψ_m. However, there are still some problems to overcome. Specifically, if we a take series $\sum_n a_n(r)\Psi_n(z)$ then value of this expansion at the point $z=0$ is equal to zero and does not satisfy the equations [after (C.15)]; the same can be said for the value at the point $z=H$. This means that, strictly speaking, we should take for the solution of equation (C.1) to provide continuity of ϕ:

$$\phi(r,z)=\sum_n a_n(r)\Psi_n(z)+\phi_1(r,z)+\phi_2(r,z), \tag{C.21}$$

where $\phi_1(r,z)=0$ if $z>0$, and correspondingly $\phi_2(r,z)=0$ if $z<H$. $\phi_{1,2}$ have values corresponding to $\sim s$ and $\sim h$ with respect to the function ϕ. These functions provide additional terms , proportional to $s,\ h,\ h'$ in (C.20). For example, within the framework of our accuracy generally speaking if we do not require continuity for ϕ at the point $z=0$ we do not need in ϕ_1 and ϕ_2. Summation of eigen functions can give discontinuity at $z=0$ (Cibbs phenomenon)

$$\phi_1(r,0)=\frac{\partial\phi_0(r,0)}{\partial z}s=\sum_m a_m\frac{d\Psi_m^0}{dz}s. \tag{C.22}$$

So the values of the functions at the boundaries are determined by the functions $\phi_{1,2}$ and their derivatives. In this case, we can differentiate the series (C.5) up to terms of order s and h to get

$$\frac{\partial^2\phi^+}{\partial z^2}=\sum_n a_n\frac{\partial^2\Psi_n^+}{\partial z^2}=\sum_n a_n\Psi_n^+(\xi_n^2-k_1^2), \tag{C.23}$$

$$\frac{\partial^2\phi^-}{\partial z^2}=\sum_n a_n\frac{\partial^2\Psi_n^-}{\partial z^2}=\sum_n a_n[\xi_n^2-k^2(H)]\Psi_n^-, \tag{C.24}$$

$$\left[\frac{\partial\phi^+}{\partial z}-\frac{\partial\phi^-}{\partial z}\right]=\sum_n a_n\left(\frac{\partial\Psi_n^+}{\partial z}-\frac{\partial\Psi_n^-}{\partial z}\right)=\left(\frac{1}{\eta}-1\right)\sum_n a_n\frac{\partial\Psi_n^-}{\partial z}, \tag{C.25}$$

$$\begin{aligned}\left[\eta\frac{\partial\phi^+}{\partial r}-\frac{\partial\phi^-}{\partial r}\right]&=\sum_n a'_n(\eta\Psi_n^+-\Psi_n^-)+\sum_n a_n\left(\eta\frac{\partial\Psi_n^+}{\partial r}-\frac{\partial\Psi_n^-}{\partial r}\right)\\&=(\eta-1)\sum_n a'_n\Psi_n^-+\sum_n a_n\left(\eta\frac{\partial\Psi_n^+}{\partial r}-\frac{\partial\Psi_n^-}{\partial r}\right),\end{aligned} \tag{C.26}$$

$$\frac{\partial\phi^0}{\partial z}=\sum_n a_n\frac{\partial\Psi_n^0}{\partial z}. \tag{C.27}$$

Consider the following integral

$$\int_0^\infty \frac{\rho}{\rho(z)} \Psi_m \frac{\partial^2 \phi}{\partial r^2} dz = \sum_n \int_0^\infty \frac{\rho}{\rho(z)} \Psi_m \frac{d^2 a_n}{dr^2} \Psi_n dz = a''_m. \tag{C.28}$$

Let us substitute this into (C.20). We get

$$a''_m + \xi_m^2 a_m + \sum_n a'_n h'(\eta - 1)\Psi_m^- \Psi_n^- + \sum_n a_n \int_0^\infty \frac{\rho}{\rho(z)} \Psi_m \delta k^2 \Psi_n dz - \sum_n a_n \frac{\partial \Psi_n^0}{\partial z} \frac{\partial \Psi_m^0}{\partial z} s$$
$$+ \sum_n a_n h \left\{ \left(\frac{1}{\eta} - 1 \right) \frac{\partial \Psi_n^-}{\partial z} \frac{\partial \Psi_m^-}{\partial z} + \Psi_m^- \Psi_n^- \left[\eta(\xi_n^2 - k_1^2) - (\xi_n^2 - k_H^2) \right] \right\} = 0. \tag{C.29}$$

This equation describes mode coupling due to surface, bottom, and volume inhomogeneities. It is often used in the theory of random inhomogeneities.

Appendix D
Empirical Orthogonal Functions

When measuring the ocean sound speed, we generally use XBTs, CTDs, or moored thermistor strings to provide input, as these give well-sampled $c(z)$ profiles at their given (x, y) geographical coordinates. However, for a reasonable number of such "casts" (order of say 10–100), each of which may sample the ocean sound speed every meter in depth, we can quickly get on the order of 10^3 to 10^5 data points. A reasonable question to ask is: "do we really *need* all these data points to represent the ocean sound speed field properly?" Moreover, the (x, y) sampling of the ocean is generally *not* done on a perfectly evenly spaced grid due to experimental difficulty and expense. More likely, the data collected are irregularly spaced and gappy, as well as somewhat noisy. In doing acoustics calculations, whether 2D or 3D, we will want to interpolate such data onto a convenient grid. So the next question becomes: "how do we interpolate such data efficiently?" We will look at one answer to these questions in the context of "empirical orthogonal functions" (EOF's), which have become a rather commonly used tool in ocean acoustics at this point in time.

To begin with, we express the fully 3D sound speed field as a vertical average profile and a 3D perturbation to that, i.e.,

$$c(x, y, z) = \bar{c}(z) + \Delta c(x, y, z). \tag{D.1}$$

More specifically, we break up the vertical average into N layers, i.e.,

$$\bar{c}(z_k) = \frac{1}{M} \sum_{j=1}^{M} c_j(x_j, y_j; z_k), \tag{D.2}$$

where$k = 1, 2, \ldots N$, and M is the number of "casts" at individual (x, y) geographical coordinates (horizontal plane grid points) that we have. Thus, we first find this $\bar{c}(z_k)$ at the number of levels we deem adequate to represent the vertical structure seen in the sound speed field over the entire region considered, we then subtract it from the true profiles as in the first equation to get $\Delta c(x, y, z)$. So far, this is just reshuffling the variability. Next, however, we make the leap needed by postulating that the

variability can be expressed as a sum of modes – specifically, the EOF modes. We write:

$$\Delta c(x, y, z) = \sum_{k=1}^{N} a_k(x, y) f_k(z), \tag{D.3}$$

where N is at most the number of vertical layers (and thus vertical modes) in the ocean that we are using.

Remind that function $\Delta c(x, y, z)$ is given in discrete set of points and its dependence on x, y, z can be replaced by discrete dependence and considered as a matrix: $\mathbf{\Delta C}$ with elements $\Delta C_{ij} = \Delta c(x_i, y_i, z_j)$ here $i = 1, 2 \ldots M$, $j = 1, 2 \ldots N$, where M is the number of points in horizontal plane (correspondingly number of strings in matrix $\mathbf{\Delta C}$), and N is the number of layers in vertical directions (or numbers of columns in $\mathbf{\Delta C}$, determined for example by the number of thermistors if we have M points in horizontal plane for the using thermistor strings for measurement of the sound speed profiles). As usually $M \gg N$.

In matrix form (D.3) can be written as

$$\mathbf{\Delta C} = \mathbf{AF}, \tag{D.4}$$

where $\mathbf{A}$ is the rectangular matrix $M \times N$, having M strings and N columns, $\mathbf{F}$ is squared, orthogonal matrix $N \times N$, $\mathbf{F}^{\mathrm{T}}\mathbf{F} = \mathbf{I}$, and columns of this matrix are orthogonal vectors $\mathbf{f}_k$ with the dimension $N \times 1$; these vectors are the so-called empirical orthogonal functions with discrete form of depth dependence.

The orthogonality of the modes implies that we can get the mode coefficients $a_k(x_i, y_i)$via the usual projection procedure, i.e.,

$$\mathbf{A} = \mathbf{\Delta C F}^{\mathrm{T}}. \tag{D.5}$$

Already we can see some advantage of the decomposition, in that the *c(x,y,z)* field is now represented by functions of (x, y) multiplied by only a comparatively few functions of z (the modes). More advantages will become apparent soon.

To obtain the EOF modes, we find the eigenvalues and eigenvectors of the covariance matrix, i.e., solve the matrix equation

$$\mathbf{\Delta C}^{\mathrm{T}}\mathbf{\Delta C f}_k = \lambda_k \mathbf{f}_k, \tag{D.6}$$

where the λ_i are the modal eigenvalues. This formula is concerned with the so-called singular value decomposition (SVD). In moredetail, matrix $\mathbf{\Delta C}$ can be decomposed in the form:

$$\mathbf{\Delta C} = \mathbf{USF}, \tag{D.7}$$

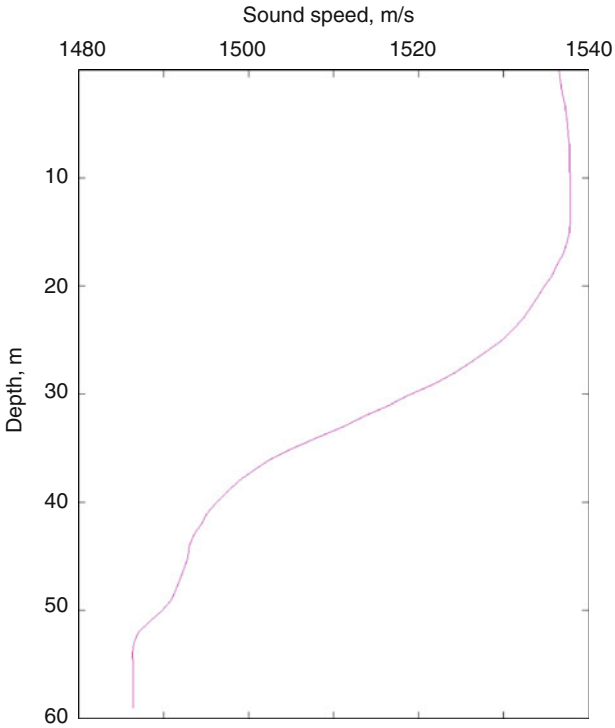


Fig. D.1 Sound speed profile in the SWARM'95 experiment

where **U** is the orthogonal matrix $M \times M$, and **S** is the diagonal matrix of the size $M \times N$. Elements at the diagonal are $s_{ii} = \sqrt{\lambda_i}$.

We will look at an example of this shortly, but first, we should talk about the further advantages of this representation.

Perhaps the biggest advantage of the EOF modes is that they contain the most energy possible in the lowest modes; i.e., they present the reddest spectrum of the ocean variability possible. This is an advantage for two reasons. First, the ocean variability spectrum is inherently red; i.e., it has more energy in the larger scale processes. Thus the EOF modes will tend to match up well with the dynamic modes of the ocean. Second, and perhaps more importantly, the "reddest" representation of the ocean variability allows one to truncate the modal expansion optimally, i.e., limit a truncated expansion to the fewest terms.

As an example, we look at some SWARM'95 experiment CTD data that were taken at a high repetition rate in order to sample coastal internal waves. These data are shown in Fig. D.1 below.

From this set of CTD casts, we can create the profile mean and deviations shown below in Figs. D.2 and D.3, and thus the EOF is shown in Fig. D.4. At the risk of triviality, we would warn the beginning user of these EOFs that they represent the variability of the quantity at hand, and nothing more. The modes one sees in

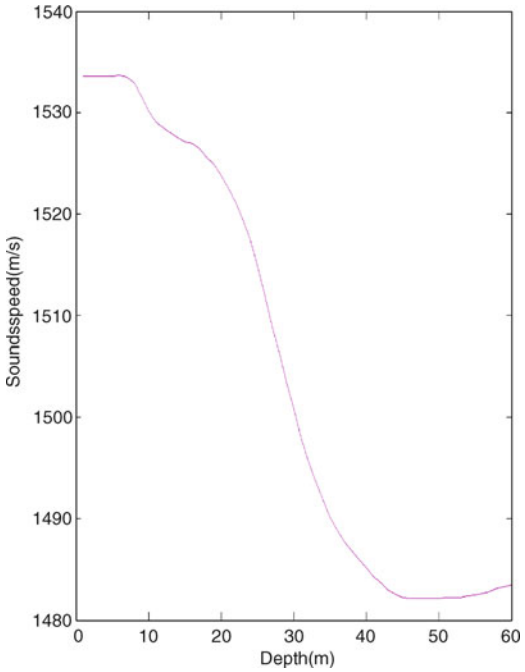


Fig. D.2 Mean of the sound speed profile from the SWARM'95 experiment

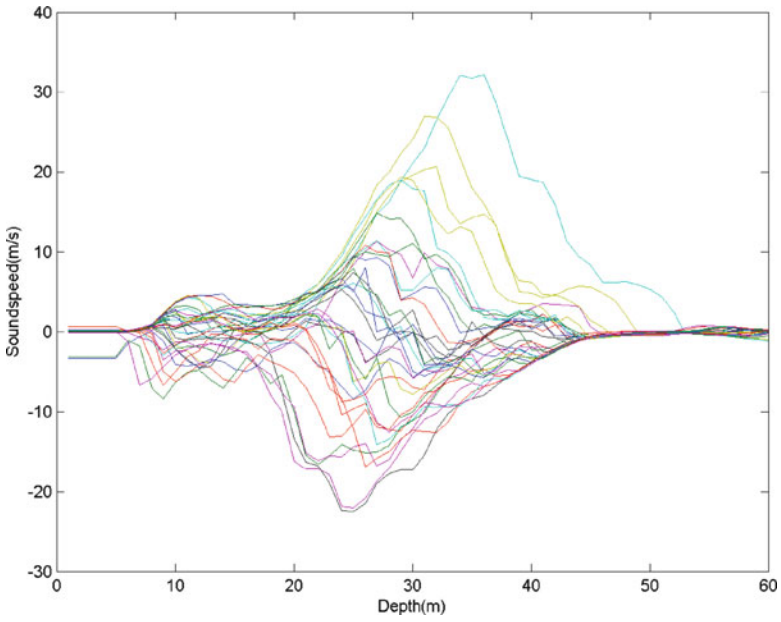


Fig. D.3 Deviation from mean of the sound speed profile from the SWARM'95 experiment

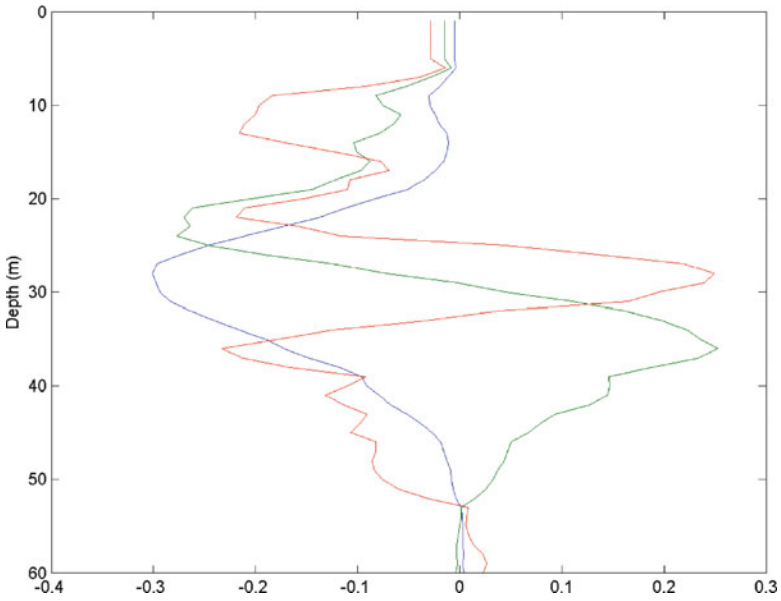


Fig. D.4 The first three lowest Empirical orthogonal functions

Fig. D.4 do have some similarities to the acoustic normal modes of the sound speed, and to the buoyancy frequency modes of the ocean density, but they are not exactly the same and should not be treated as equal or even proportional.

Added to the economy of representation and correspondence to the ocean modes that the EOFs display, there is yet another significant advantage to their use – in particular, one can do a very good job of 3D interpolation of the sound speed (and other ocean variable) field using the EOF representation. Specifically, if one wants the ocean profile at some particular (x,y) position where a profile has not been taken, a interpolation of the EOF coefficients for the energetically significant modes will provide a good answer. Nowadays, one can easily find software to do 2D linear interpolation between points (e.g., in MATLAB); however, there are options available that are truer to the ocean dynamics, and we will discuss one such here – Gauss–Markov interpolation. (In this style of interpolation, one accounts more precisely for the horizontal correlation scales of the ocean features, which are not considered in simpler interpolations.)

Let us assume, following Newhall et al. (1990), that a measurement (the EOF coefficients, $a_n(x, y)$ that we infer from the EOF analysis of the raw data) φ_r at an (x, y) position r consists of a true value θ_r plus some noise ε_r (computed from the error/noise in the $c(x, y, z)$measurements), i.e., $\varphi_r = \theta_r + \varepsilon_r$. If the error is spatially uncorrelated and also is not correlated to the true value, i.e.,

$$E[\varepsilon_r \varepsilon_s] = \varepsilon^2 \delta_{rs},$$

$$E[\varepsilon_r \theta_r] = 0, \tag{D.8}$$

where E is an average or expectation value, and $\varepsilon^2$is the error variance, then the least squares optimal estimator of θat any point t is given by

$$\hat{\theta}_t = \sum_{r=1}^{n} B_{tr} \left[\sum_{s=1}^{n} A_{rs}^{-1} \varphi_s \right]. \tag{D.9}$$

In the above equation, n is the number of measurements (data points) one uses in forming the estimate and $A_{rs}$and $B_{tr}$are matrices given by

$$A_{rs} = E[\varphi_r \varphi_s] = C(\vec{x}_r - \vec{x}_s) + \varepsilon^2 \delta_{rs},$$

$$B_{tr} = C(\vec{x}_t - \vec{x}_r), \tag{D.10}$$

where C is the spatial (horizontal) correlation function. In general, this correlation function should be constructed from a data–data correlation matrix, and will have different correlation scales in the x and y directions. However, we can consider a simpler, yet very useful case here, in which the medium is isotropic horizontally, and the correlation function is a simple Gaussian form with a radial length scale dictated by the dynamics of the medium. This scale can often be taken as the local baroclinic Rossby radius (discussed previously) for shallow water. Data points within this scale will affect the interpolation to the desired point, whereas points outside will have very little effect. Moreover, we merely need to use only a few (perhaps 6–10) of the points that lie within a correlation radius of our desired point to get a good estimate, thus saving computational difficulty. This type of interpolator will revert to the average profile when there are no data within a correlation length of the desired point, which again makes physical sense. For very large correlation lengths, this type of interpolation gives a "smoothest" result, and starts to look more like a standard linear interpolation of the three nearest (noncolinear) neighbors,

Moreover, we note that this interpolation scheme can also deal very easily with irregular bathymetry, e.g., underwater hills and valleys. Specifically, one takes the vertical levels to span the deepest depth encountered, and then extends these levels through the solid bottom as well. When a vertical level is in the bottom, it reverts to the ocean background (mean) profile for that depth, and is not used for interpolating water column properties at other locations.

For readers wishing to see some examples of this type of interpolation for acoustics purposes, we would refer them to Newhall et al. (1990), Emery and Thomson (1997).

Appendix E
Scattering at Localized Inhomogeneities in the Waveguide. Approximate Description

The mode theory described in this book can be used also for calculation of sound field scattered at localized inhomogeneities in shallow water waveguide. For this purpose, it is suggested (Kravtsov et al. 1983; Makris and Ratilal 2001; Katsnelson and Petnikov 2002) to use the so-called scattering amplitude $\mathrm{E}(\chi^{\mathrm{in}}, \vartheta^{\mathrm{in}}, \chi^{\mathrm{sc}}, \vartheta^{\mathrm{sc}})$ assignable for free space and depending on four angles in the general case. Here ϑ^{in} and ϑ^{sc} are polar angles defining the vertical planes, in which the wavevector of the incident and scattered sound waves lie, respectively $\left(\vartheta^{\mathrm{in}} - \vartheta^{\mathrm{sc}} = \Delta\vartheta^{\mathrm{sc}}\right)$. The amplitude $\mathrm{E}(\chi^{\mathrm{in}}, \vartheta^{\mathrm{in}}, \chi^{\mathrm{sc}}, \vartheta^{\mathrm{sc}})$ allows one to calculate scattered sound field including the case when the source and receiver are placed at different points and the angle $(\Delta\vartheta^{\mathrm{sc}} \neq 180^{\circ})$.

Let us examine this approach in more detail. To determine the scattering amplitudes, we use the well-known Helmholtz–Kirchhoff integral equation (Pierce 1991), which allows calculation of the scattered field $P^{\mathrm{sc}}(\vec{R})$ at a distant point removed from the scattering circle. Integrating over the scatterer surface A_{sc} defined by the radius vector $\vec{R}'$:

$$P^{\mathrm{sc}}(\vec{R}) = -\int\int_{A_{sc}} \left[P^{\mathrm{in}}(\vec{R}') + P^{\mathrm{sc}}(\vec{R}')\right] \frac{\partial \Psi(\vec{R}, \vec{R}')}{\partial n_{\mathrm{sc}}} - \Psi(\vec{R}, \vec{R}_{\mathrm{sc}}) \frac{\partial}{\partial n_t} \times \left[P^{\mathrm{in}}(\vec{R}') + P^{\mathrm{sc}}(\vec{R}')\right] dA_{\mathrm{sc}}. \tag{E.1}$$

For free space, assuming that a plane wave with amplitude A_0 is incident on the scatterer, the incident sound field $P^{\mathrm{in}}(\vec{R}')$ can be written as

$$P^{\mathrm{in}}(\vec{R}') = A_0 \exp(i\vec{k}\vec{R}') = A_0 \exp(i\vec{k}\vec{R}'_0 + i\vec{k}\vec{R}'') = A_0 \exp(ik(R'_0\eta_0(\chi^{\mathrm{in}}, \vartheta^{\mathrm{in}}) + R''\eta(\chi^{\mathrm{in}}, \vartheta^{\mathrm{in}}))), \tag{E.2}$$

where $\vec{R}'_0$ is the radius vector of the scatterer center and $\vec{R}''$ is the radius vector from the scatterer center to a point on its surface, $\vec{R}' = \vec{R}'_0 + \vec{R}''$. Further, let us assume that scatterer sizes are small and $\left|\vec{R}''\right| \ll \left|\vec{R}'_0\right|$, giving

$$\eta_0(\chi^{\text{in}},\vartheta^{\text{in}}) = \cos\chi^{\text{in}}\cos\chi_0^t + \sin\chi^{\text{in}}\sin\chi_0^t\cos(\vartheta^{\text{in}}-\vartheta_0^t),$$

$$\eta(\chi^{\text{in}},\vartheta^{\text{in}}) = \cos\chi^{\text{in}}\cos\chi^t + \sin\chi^{\text{in}}\sin\chi^t\cos(\vartheta^{\text{in}}-\vartheta^t), \tag{E.3}$$

where χ^t, χ_0^t, ϑ^t, and ϑ_0^t are angular coordinates of the point on the scatterer surface and at its center, respectively.

Then the following approximate equation for the Green's function (3.3) is valid for small scatterer sizes:

$$\begin{aligned}\Psi(\vec{R},\vec{R}') &= \frac{1}{4\pi}\frac{\exp\left(ik\left|\vec{R}-\vec{R}'\right|\right)}{\left|\vec{R}-\vec{R}'\right|} = \frac{1}{4\pi}\frac{\exp\left(ik\left|\vec{R}-\vec{R}'_0-\vec{R}''\right|\right)}{\left|\vec{R}-\vec{R}'_0-\vec{R}''\right|}\\ &\approx \frac{1}{4\pi}\frac{\exp(ik(|v|-R''\eta(\chi^{\text{sc}},\vartheta^{\text{sc}}))}{|\vec{R}-\vec{R}'_0|},\end{aligned} \tag{E.4}$$

where $\left|\vec{R}''\right| \ll \left|\vec{R}-\vec{R}'_0\right|$. Under the above assumptions, (E.1) can be rewritten as

$$\begin{aligned}P^{\text{sc}}(\vec{R}) = {} & -A_0\exp\left(ikR'_0\eta'(\chi^{\text{in}},\vartheta^{\text{in}})\right)*\\ & \left\{\frac{1}{4\pi}\int\!\!\int_{A_{\text{sc}}}\left[\exp\left(ikR''\eta(\chi^{\text{in}},\vartheta^{\text{in}})\right)+\frac{P^{\text{sc}}(\vec{R}')}{A_0\exp\left(ikR'_0\eta'(\chi^{\text{in}},\vartheta^{\text{in}})\right)}\right]\frac{\partial}{\partial n_{\text{sc}}}\exp(-ikR''\eta(\chi^{\text{sc}},\vartheta^{\text{sc}})\right.\\ & \left.-\exp(-ikR''(\eta(\chi^{\text{sc}},\vartheta^{\text{sc}}))\frac{\partial}{\partial n_{\text{sc}}}\left[\exp\left(ikR''\eta(\chi^{\text{in}},\vartheta^{\text{in}}\right)+\frac{P^{\text{sc}}(\vec{R}')}{A_0\exp\left(ikR'_0\eta'(\chi^{\text{in}},\vartheta^{\text{in}})\right)}\right]dA_{\text{sc}}\right\}\\ & \frac{\exp(ik\left|\vec{R}-\vec{R}'_0\right|)}{\left|\vec{R}-\vec{R}'_0\right|}.\end{aligned} \tag{E.5}$$

The braced expression in formula (6.27) is the scattering amplitude in free space $\mathrm{E}(\chi^{\text{in}},\vartheta^{\text{in}},\chi^{\text{sc}},\vartheta^{\text{sc}})$. This function is well known in plane wave diffraction and scattering problems, where the scattered field is written as the product of the plane incident wave, the scattering amplitude, and a spherical by divergent wave,

$$P^{\text{sc}}(\vec{R}) = A_0\exp(i\vec{k}\vec{R}'_0)E(\chi^{\text{in}},\vartheta^{\text{in}},\chi^{\text{sc}},\vartheta^{\text{sc}})\frac{\exp\left(ik\left|\vec{R}-\vec{R}'_0\right|\right)}{\left|\vec{R}-\vec{R}'_0\right|}, \tag{E.6}$$

where

$$\begin{aligned}E(\chi^{\text{in}},\vartheta^{\text{in}},\chi^{\text{sc}},\vartheta^{\text{sc}}) = {} & -\frac{1}{4\pi}\int\!\!\int_{A_{\text{sc}}}\left[\exp\left(ikR''\eta(\chi^{\text{in}},\vartheta^{\text{in}})\right)+\frac{P^{\text{sc}}(\vec{R}')}{A_0\exp\left(ikR'_0\eta'(\chi^{\text{in}},\vartheta^{\text{in}})\right)}\right]\\ & \frac{\partial}{\partial n_{\text{sc}}}\exp(-ikR''\eta(\chi^{\text{sc}},\vartheta^{\text{sc}})-\exp(-ikR''(\eta(\chi^{\text{sc}},\vartheta^{\text{sc}}))\\ & \frac{\partial}{\partial n_{\text{sc}}}\left[\exp\left(ikR''\eta(\chi^{\text{in}},\vartheta^{\text{in}}\right)+\frac{P^{\text{sc}}(\vec{R}')}{A_0\exp\left(ikR'_0\eta'(\chi^{\text{in}},\vartheta^{\text{in}})\right)}\right]dA_{\text{sc}}\end{aligned} \tag{E.7}$$

Here it is pertinent to make several notes. First, expressions (E.6) and (E.7) are valid in the far-field diffraction pattern, i.e., where the condition $|R - R'_0| \gg L^2/\lambda$, is satisfied, where L is the characteristic scatterer size.

Second, the scattering amplitude is usually measured or calculated in a coordinate system related to the scatterer. (The scatterer center is considered as the coordinate origin.) This should be taken into account when determining the angles $\chi^{\text{in}}, \vartheta^{\text{in}}, \chi^{\text{sc}}, \vartheta^{\text{sc}}$ in (E.6), where another coordinate system is used.

Third, the function $\mathrm{E}(\chi^{\text{in}}, \vartheta^{\text{in}}, \chi^{\text{sc}}, \vartheta^{\text{sc}})$ can be formally determined in a similar way for scattering from a bounded rough surface area or for scattering from localized inhomogeneities. In this case, the scattering amplitude can also be written in terms of the bulk density of secondary fictitious sources [see e.g., Katsnelson and Petnikov (2002)].

Fourth, for some specific symmetric inhomogeneities, i.e. structurally homogeneous bodies of revolution like the sphere and spheroid, the integral in (E.7) can be calculated. The scattering amplitude for these bodies is known. For bodies with more complex shape and structures, this characteristic is experimentally measured in free space.

Fifth, for a randomly rough surface, i.e., the sea-bottom, the scattering amplitude is an essentially random value. In this case, the above-mentioned angular dependences of the backscattering coefficient, obeying the Lambert and Lommel–Seeliger laws, are in essence the simplest and most typical angular dependences of the scattering amplitude variance. One of the possible methods for modeling sound scattering by such a bottom is considered in Sect. 6.5. See also Makris and Ratilal (2001).

Using the amplitude $\mathrm{E}(\chi^{\text{in}}, \vartheta^{\text{in}}, \chi^{\text{sc}}, \vartheta^{\text{sc}})$, scattering in the waveguide can also be described in a slightly different way. In particular, in the ray representation of the scattered wave in a smoothly inhomogeneous medium with refractive index $n(\vec{R})$ (such that $L|\nabla n| \ll n$), the expression (E.6) for the scattered sound field can be rewritten as[5] (Kravtsov et al. 1983)

$$P^{\text{sc}}(\vec{R}) = A_0 \exp(i\vec{k}\vec{R}'_0) E(\chi^{\text{in}}, \vartheta^{\text{in}}, \chi^{\text{sc}}, \vartheta^{\text{sc}}) \Theta(R'_0, R) \tag{E.8}$$

where $\Theta(R'_0, R)$ is the non-directional quasi-spherical wave from a point source in a smoothly inhomogeneous medium,

$$\Theta(R'_0, R) = \sqrt{\frac{d\Omega}{ds} \frac{n(R'_0)}{n(R)}} \exp\left(ik \int_{R'_0}^{R} n(l) dl \right) \tag{E.9}$$

where $d\Omega$ and ds are the solid angle and cross-section of the tube of rays leaving the point R'_0, and the angles $\chi^{\text{in}}, \vartheta^{\text{in}}, \chi^{\text{sc}}, \vartheta^{\text{sc}}$ define the direction of incident and scattered sound waves at the point $\vec{R}'_0$.

If more than one ray comes to the observation point, a sum of the corresponding fields should be taken,

$$P^{\mathrm{sc}}(\vec{R}) = A_0 \exp(i\vec{k}\vec{R}_0') \sum_{v} E(\chi^{\mathrm{in}}, \vartheta^{\mathrm{in}}, \chi_v^{\mathrm{sc}}, \vartheta_v^{\mathrm{sc}}) \Theta_v(R'_0, R). \tag{E.10}$$

Thus expression (E.10) defines the scattered field in a smoothly inhomogeneous medium, including the acoustic waveguide. However, it should be emphasized that this expression is valid only under the condition of the smallness of the secondarily scattered fields, i.e., the fields scattered again at the body after reflection from the waveguide boundaries.

To go to the mode description of the scattered field, mainly used in this monograph, let us expand the scattered field (E.10) in the eigenfunctions of the waveguide system:

$$P^{\mathrm{sc}}(R) = \sum_{\mu} b_\mu(r) \Psi_\mu(z) \tag{E.11}$$

where the coefficients $b_\mu(r)$ are determined by the equation:

$$b_\mu(r) = \int_0^H P^{\mathrm{sc}}(R) \Psi_\mu(z) dz \tag{E.12}$$

Substituting the ray field (E.10) into (E.12), we obtain

$$b_\mu(r) = A_0 \exp(i\vec{k}\vec{R}_0') \sum_{v} \int_0^H E(\chi^{\mathrm{in}}, \vartheta^{\mathrm{in}}, \chi_v^{\mathrm{sc}}, \vartheta_v^{\mathrm{sc}}) \Theta_v(R'_0, R) \Psi_\mu(z) dz. \tag{E.13}$$

The integrand in (E.13) includes two rapidly oscillating functions $\Psi_\mu(z)$ and $\Theta_v(R'_0, R)$; therefore, it is convenient to calculate the integral using the stationary point method.

Further, let us consider scattering in a Pekeris waveguide (see Sect. 3.2) with ideally reflecting boundaries. The upper boundary $z = 0$ is taken as soft, the lower boundary $z = H$ is taken as hard. For this case, if $P_0^{\mathrm{sc}}(R)$ is the scattered field in free space, the field scattered inside the waveguide is written in the ray approximation as

$$P^{\mathrm{sc}}(R) = \frac{1}{4\pi} \sum_{v=-\infty}^{\infty} (-1)^v \left[P_0^{\mathrm{sc}}(R_v) - P_0^{\mathrm{sc}}(R'''_v) \right], \tag{E.14}$$

where $\vec{R}_v = \vec{R} + 2H\vec{i}_z v$, $\vec{R'''}_v = \vec{R'''} + 2H\vec{i}_z v$, and $R''' = (x, y, -z)$ define mirror images of the observation point $R = (x, y, z)$ in the walls and $\vec{i}_z$ is the unit vector along the z axis. [The representation of the sound field from a point source in the waveguide in terms of its images is given in more detail in Keller and Papadakis (1977)]. For the waveguide under consideration, taking into account (3.61), and (3.62), the sound field incident on the scatterer can be written as sums of plane waves.

More precisely, each waveguide mode can be written as a sum of two plane waves, hence, the incident field is written as

$$P^{\text{in}}(\vec{R}') \cong \sum_m \frac{C_m^0 \exp(-\gamma_m r'_0/2)}{\sqrt{\xi_m r'_0}} \left(\exp i\vec{k}_m^+ \vec{R}' + \exp i\vec{k}_m^- \vec{R}'\right), \tag{E.15}$$

where r'_0 is the scatterer center coordinate in the cylindrical coordinate system, $C_m^0 = \frac{1}{2\sqrt{i}}\sqrt{\rho_0 c_0 W} N_m \sin\sigma_m z_s$, $\vec{k}_m^+ = q_m \vec{i}_r + \sigma_m \vec{i}_z$, $\vec{k}_m^- = q_m \vec{i}_r - \sigma_m \vec{i}_z$ and $\vec{i}_r$ is the horizontal unit vector toward the scatterer. Here it is assumed that the sound source is at the point with cylindrical coordinates $(0, z_s)$. For such an incident wave, according to (E.6), the scattered field is written as

$$P_0^{\text{sc}}(R) = \sum_m \frac{C_m^0 \exp(-\gamma_m r'_0/2)}{\sqrt{\xi_m r'_0}} \exp(\text{i} q_m r'_0) * (E(\chi_m^{\text{in}}, \vartheta^{\text{in}}, \chi^{\text{sc}}, \vartheta^{\text{sc}}) \exp(\text{i}\sigma_m z'_0)$$
$$+ E(\pi - \chi_m^{\text{in}}, \vartheta^{\text{in}}, \chi^{\text{sc}}, \vartheta^{\text{sc}}) \exp(-\text{i}\sigma_m z'_0)) \frac{\exp(\text{i}k|R - R'_0|)}{|R - R'_0|}. \tag{E.16}$$

Substituting (E.16) into (E.14), we obtain the ray representation of the scattered field such as (E.10). For the mode representation of the scattered field, it is sufficient to calculate the coefficients b_μ by relation (E.13) using the stationary point method. Paraphrasing the detailed technique for calculating the integral (E.13), we simply note that a single stationary point z_{st} for any summand $P_0^{\text{sc}}(R_v)$ and a single point for one of the summands $P_0^{sc}(R_v''')$ appears in the interval $(0<z<H)$ at a chosen value of μ; in this case, the π-vicinity of the stationary point (i.e., the vicinity in which the phase of the integrand differs from the stationary value by no more than π) is equal to $\Delta z_\pi = \sqrt{\lambda r/\cos\chi_\mu}$. As the condition

$$|r - r_0| \gg (L^2/\lambda)\cos\chi_\mu \tag{E.17}$$

is satisfied, scattering amplitudes are almost constant within the π-vicinity $(z_{\text{st}} - \Delta z_\pi/2,\ z_{\text{st}} + \Delta z_\pi/2)$, and they can be factored outside from the integral sign at the indicated values of angles,

$$\left\{ \begin{array}{cc} E(\chi_m^{\text{in}}, \vartheta^{\text{in}}, \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) & E(\pi - \chi_m^{\text{in}}, \vartheta^{\text{in}}, \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}), \\ E(\chi_m^{\text{in}}, \vartheta^{\text{in}}, \pi - \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) & E(\pi - \chi_m^{\text{in}}, \vartheta^{\text{in}}, \pi - \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) \end{array} \right\}. \tag{E.18}$$

Finally, we come to the following result:

$$b_\mu = \sqrt{\frac{i}{8\pi}} \frac{\exp(i\xi_\mu|\vec{r} - \vec{r}'_0|)}{\sqrt{\xi_\mu|\vec{r} - \vec{r}'_0|}} \sum_m \frac{C_m^0 \exp(-\gamma_m r'_0/2)}{\sqrt{\xi_m r'_0}} T_{m\mu}, \tag{E.19}$$

where the scattering matrix $T_{m\mu}$ is given by

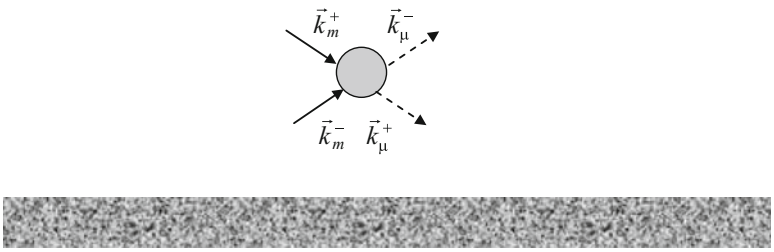


Fig. E.1 Sound scattering at localized inhomogeneity in the waveguide. $\vec{k}_m^+$ and $\vec{k}_m^-$ wave vectors of the plane waves corresponding to the mode of the incident field with number m. $\vec{k}_\mu^+$ and $\vec{k}_\mu^-$ a similar wave vectors corresponding to the mode of the scattered field with number μ

$$T_{m\mu} = \frac{2\pi}{i} N_m N_\mu \Big(E(\chi_m^{\text{in}}, \vartheta^{\text{in}}, \pi - \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) \exp(\mathrm{i}(\sigma_m + \sigma_\mu) z'_0) - E(\chi_m^{\text{in}}, \vartheta^{\text{in}}, \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) \exp(i(\sigma_m - \sigma_\mu) z'_0) - E(\pi - \chi_m^{\text{in}}, \vartheta^{\text{in}}, \pi - \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) \exp(i(\sigma_\mu - \sigma_m) z'_0) + E(\pi - \chi_m^{\text{in}}, \vartheta^{\text{in}}, \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) \exp(-\mathrm{i}(\sigma_m + \sigma_\mu) z'_0) \Big) \tag{E.20}$$

Thus, the derived relations (E.11), (E.19), and (E.20) describe sound field scattering in the waveguide in terms of modes. This process can be pictorially presented as the transformation of each pair of plane waves, corresponding to an individual mode of the incident sound field, to all possible pairs of similar waves, corresponding to scattered field modes (see Fig. E.1).

In the case where the sound speed depends on z sufficiently smoothly, we can use the WKB method for eigenfunctions (3.64); then expression (E.20) becomes

$$T_{m\mu} = \frac{2\pi}{i\sqrt{\sigma_m(z'_0)}} N_m N_\mu \Big(E(\chi_m^{\text{in}}, \vartheta^{\text{in}}, \pi - \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) \exp(i(\varphi_m + \varphi_\mu)) - E(\chi_m^{\text{in}}, \vartheta^{\text{in}}, \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) \exp(i(\varphi_m - \varphi_\mu)) - E(\pi - \chi_m^{\text{in}}, \vartheta^{\text{in}}, \pi - \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) \exp(i(\varphi_\mu - \varphi_m)) + E(\pi - \chi_m^{\text{in}}, \vartheta^{\text{in}}, \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) \exp(-i(\varphi_m + \varphi_\mu)) \Big), \tag{E.21}$$

where $\varphi_{m,\mu} = \int_0^{z'_0} \sigma_{m,\mu}(z) dz$.

In conclusion, we note one more representation for the scattering matrix $T_{m\mu}$ for the inhomogeneous waveguide, which was derived in Katsnelson and Petnikov (2002),

$$T_{m\mu} = 4\pi \Big(E(\chi_m^{\text{in}}, \vartheta^{\text{in}}, \pi - \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) a_m^+ a_\mu^+ + E(\chi_m^{\text{in}}, \vartheta^{\text{in}}, \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) a_m^+ a_\mu^- + E(\pi - \chi_m^{\text{in}}, \vartheta^{\text{in}}, \pi - \chi_\mu^{\text{sc}}, \vartheta^{\text{sc}}) a_m^- a_\mu^+ + E(\pi - \chi_m^{\text{in}}, \vartheta^{\text{sc}}, \chi_\mu^{\text{sc}}) a_m^- a_\mu^- \Big), \tag{E.22}$$

where $a_{m,\mu}^\pm = \frac{1}{2} \left[\Psi_{m,\mu}(z) \pm \frac{d\Psi_{m,\mu}/dz}{i\sigma_{m,\mu}(z)} \right]_{z=z'_0}$.

A specific example of scattered sound field, calculated by using approach described above can be found in this book (Katsnelson and Petnikov 2002) as well.

Appendix F
Reflection of a Plane Wave from Half-Space

One of the most important relationships, used in all areas of wave propagation is reflection and transmission coefficients of plane waves from a plane interface between homogeneous half-spaces. This situation is illustrated in Fig. F.1.

Let us consider a two-dimensional situation, where the X-axis is the horizontal coordinate, and the Z-axis is the vertical axis, directed downward. The wavevector of plane waves in the upper medium is $\vec{k} = (q, \sigma)$, and correspondingly the complex amplitude of the sound field is $P(x, z, t) = A\mathrm{e}^{\mathrm{i}(qx+\sigma z-\omega t)}$. In the lower medium components of wavevector will be labeled by index 1. If we have interface between two homogeneous media (Fig. F.1) then the time independent complex amplitudes of the sound field in both media have the form:

$$P(x,z) = \begin{cases} \exp[i(qx+\sigma z)] + V\exp[i(qx-\sigma z)], & z<0 \\ W\exp[i(q_1 x+\sigma_1 z)], & z>0, \end{cases} \tag{F.1}$$

where we have introduced the reflection coefficient V and the transmission coefficient W. To get these values, we use the continuity conditions

$$\begin{aligned} P(x,z)\big|_{z\to 0-} &= P(x,z)\big|_{z\to 0+}, \\ \frac{1}{\rho}\frac{\partial P(x,z)}{\partial z}\bigg|_{z\to 0-} &= \frac{1}{\rho_1}\frac{\partial P(x,z)}{\partial z}\bigg|_{z\to 0+}, \end{aligned} \tag{F.2}$$

where notation $z \to 0\pm$ denotes that z tends to zero being positive/negative

Substituting (F.2) into (F.1), we get

$$q = q_1 = k\cos\chi. \tag{F.3}$$

Or correspondingly using the values of the horizontal angles, we get

$$\frac{\cos\chi}{c} = \frac{\cos\chi_1}{c_1}. \tag{F.4}$$

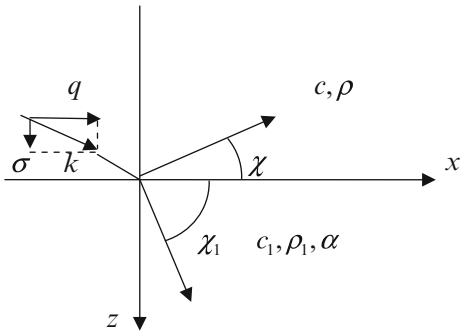


Fig. F.1 Wave vectors and angles for reflection of plane wave

This is simply Snell's law, and we find

$$\begin{aligned} 1+V &= W, \\ \frac{\sigma}{\rho}(1-V) &= \frac{\sigma_1}{\rho_1}W. \end{aligned} \tag{F.5}$$

Let us now denote

$$n_1 = \frac{c}{c_1},\ m_1 = \frac{\rho_1}{\rho}. \tag{F.6}$$

And solve the equations

$$\begin{aligned} V &= \frac{m_1\sqrt{k^2-q^2}-\sqrt{n_1^2k^2-q^2}}{m_1\sqrt{k^2-q^2}+\sqrt{n_1^2k^2-q^2}}, \\ W &= \frac{2m_1\sqrt{k^2-q^2}}{m_1\sqrt{k^2-q^2}+\sqrt{n_1^2k^2-q^2}}. \end{aligned} \tag{F.7}$$

We can also write another form of the reflection/transmission coefficients using the grazing angle χ:

$$\begin{aligned} V &= \frac{m_1\sin\chi-\sqrt{n_1^2-\cos^2\chi}}{m_1\sin\chi+\sqrt{n_1^2-\cos^2\chi}}, \\ W &= \frac{2m_1\sin\chi}{m_1\sin\chi+\sqrt{n_1^2-\cos^2\chi}}. \end{aligned} \tag{F.8}$$

The expressions (F.7 and F.8) are called the Fresnel coefficients. In underwater acoustics as usually values of grazing angle are comparatively small, not greater than about 0.3 rad for typical conditions for densities and sound speed in bottom. The grazing angle, corresponding to the value $\cos\chi_0 = n_1$is called the total reflection angle, which produces $V = 1$, $W = 2$.

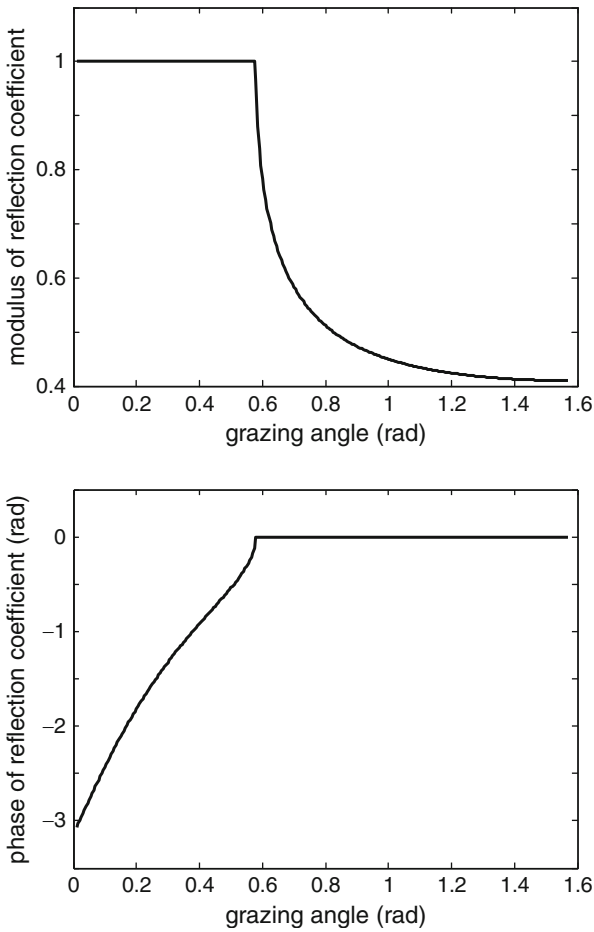


Fig. F.2 Modulus and phase of reflection coefficient

If $\cos\chi > n$ then the square root becomes imaginary and the complex reflection coefficient has a modulus equal to one:

$$V = \exp(i\varphi_V), \quad \varphi_V = -2\arctan\frac{\sqrt{\cos^2\chi - n_1^2}}{m_1 \sin\chi}. \tag{F.9}$$

Here the phase shift, determined by (F.9) gives phase difference between the incident and reflected waves. Also it determines displacement of a narrow beam by reflection (Brekhovskikh 1980).

If the lower medium is absorbing, characterized by coefficient α [see (3.1)] then the reflection coefficient becomes complex for any grazing angle

$$V = |V| \exp(i\varphi_V), \tag{F.10}$$

where the phase function has more a complex form than (F.9). The angle dependence of the modulus and the phase of the refraction coefficient are shown in the Fig. F.2.

In more general case we can introduce the reflection coefficient of a plane wave from the plane interface with arbitrary medium, if boundary (continuity) conditions have "local" character and can be described by equation

$$\left[P(x,z) + g(q)\frac{\partial P(x,z)}{\partial z}\right]_{z=0} = 0, \tag{F.11}$$

where value $g(q)$ is function depending on the lower medium. For example for the previous case $g(q) = \frac{m_1}{\sqrt{q^2 - k_1^2}}$, which in the case of absorbing medium will be complex. Value $g(q)$ can be constructed for different sorts of medium – layered, poro-elastic, etc.

Using this value reflection coefficient has the form:

$$V = \frac{\sigma g + i}{\sigma g - i}. \tag{F.12}$$

Main Provisional Notation

$\langle\rangle$	Point brackets denote averaging over a statistical ensemble of realization
a	Diffusion coefficient
a_l	Modal amplitude in the sound field decomposition
A_l	Amplitude of horizontal ray
A_{sc}	Scattering surface area
b_{s}	Soliton half width
$B(\vec{R}, \vec{R}')$	Sound field spatial correlation function
$B_C(\tau)$	Envelope of the joint correlation function of the received and emitted signals
$B_B(\tau)$	Calculated
$B_{\mathrm{c}}(\vec{R}, \vec{R}')$	Correlation function for the sound speed
$B_d(z, z')$	Correlation coefficient in the vertical direction
$B_{\mathrm{s}}(\vec{R}, \vec{R}')$	Correlation function for surface disturbance
$B_q(\vec{r}, \omega; \vec{r}', \omega')$	Joint spectral correlation function for surface noise sources
$c(z)$	Sound speed in water
$c(z) = c_s(1 - \varepsilon z)$	Vertical linear profile of the sound speed
c_0	Sound speed at the source
c_{s}	Sound speed at the surface
c_H	Sound speed on the bottom
C_l	Modal amplitude in the sound field decomposition in the forward scattering approximation
c_n	Sound speed in a multilayered bottom
c_m^{ph}	Phase velocity of mth mode
c_l^{gr}	Group velocity of lth mode
$E(\chi^{\mathrm{in}}, \vartheta^{\mathrm{in}}, \chi^{\mathrm{sc}}, \vartheta^{\mathrm{sc}})$	Scattering amplitude for free space
f	Cycle frequency
f_l	Cutoff frequency of lth mode
Δf_m	Doppler frequency shift for mth mode
δf	Width of spectral line

f_0	Position of first maximum in the spectrum of an explosive signal
$F(\vec{k}_i, \vec{k}_s)$, $F(\chi^{\text{in}}, \chi^{\text{sc}})$	Scattering amplitude
$g(\xi) = \frac{i}{\omega\rho} Z_1(\xi,), Z_1(\xi)$	Impedance of the bottom
$g(\vartheta)$	Directivity function for sources in the plane
g_{sc}	Constant factor determined scattering properties of sea bottom
$G(\omega)$	Temporal power-density spectrum
$G_{\text{a}}(\omega)$	Temporal power-density spectrum of the amplitude fluctuations
$G(\vec{k}, \omega)$	Spatial-temporal power-density spectrum of the sound field
$G_l(\tilde{\vec{q}}, \boldsymbol{\Omega})$	Spatial-temporal density of the internal wave field
$G^{\text{t}}(\boldsymbol{\Omega})$	Temporal power-density spectrum of the internal wave field
$G_l^{\text{s}}(\tilde{\vec{q}})$	Spatial power-density spectrum of the internal wave field
G_{ml}	Mode interaction coefficient
$h(r)$	Depth variations
H	Depth of sea
I	Sound intensity
$\bar{I}$	Sound intensity averaged over space
i	Imaginary unit
i_{r}	Number of array receiver, $(i_{\text{r}} = 1, 2, \ldots, I_{\text{r}})$
j	Number of array source $(j = 1, 2, \ldots, J)$
J_{r}	Ray invariant
$\vec{k}$	Wavevector of the sound field
$\vec{k}$	Wavevector of surface and internal waves
k	Value of wavevector
$\vec{k}_{\text{i}}$	Incident wave wavevector
$\vec{k}_{\text{s}}$	Scattered wave wavevector
k_{s}	Wavevector on sea surface
K	Value of the numerical criterion for comparing proximity
K_0	Coefficient of proportionality, depending on the type of the explosive material
l, m	Mode indices
L	Length of acoustic antenna
$\tilde{L}$	Characteristic scale for variations of waveguide parameters
$\tilde{L}_H = \frac{H}{H_y}$	Characteristic scale for variations of depth $(H_y = \frac{\partial H}{\partial y})$
$\tilde{L}_c = \frac{c}{c_y}$	Characteristic scale for variations of sound speed $(c_y = \frac{\partial c}{\partial y})$
M	Maximum number of propagating modes

$\vec{M}(t)$	Vector of time-varying dipole moment
$m_1 = \frac{\rho_1}{\rho_0}$	Density ratio
m_b	Bottom backscattering coefficient
$n(z) = c_\mathrm{s}/c(z)$	Refractive index in waveguide
$n_1 = c_H/c_1$	Refractive index in bottom
n	Index of layer in multilayered bottom
N	Number of layers in bottom
N_l	Normalization coefficient
$\hat{N}_m = \sum_j^J \Psi_m^2(z_j)$	Gain of vertical chain of hydrophones
$\tilde{N}$	Buoyancy (Brunt–Vaisala) frequency
p	Continuous parameter corresponding to the mode index
p_0	Rms sound pressure at a distance of 1 m from the source
$p_0(r,z)$	Unperturbed sound pressure
$p(r,z,t)$	Perturbation of real (measured) sound pressure
p_st	The standard pressure ($p_\mathrm{st} = 1\ \mu$Pa)
P_m	Pressure amplitude for mth mode
$P(r,z,t)$	Complex sound pressure
$\lvert P(r,z.t)\rvert$	Sound pressure amplitude
P^in	Complex sound pressure in incident wave
P^sc	Complex sound pressure in scattered wave
d	Exponent determining the frequency dependence of sound absorption in the bottom
q_l	Horizontal propagation constant of the lth mode
$\vec{q}$	Horizontal wavevector of internal or surface waves
$Q(\vec{R},t)$	Volume density of emitted signal
$Q(\vec{r},t)$	Surface density of emitted signal
$\tilde{Q}$	Coefficient, depending on the physical properties of the water
r,ϑ,z	Cylindrical coordinate system
R,ϑ,ϕ	Spherical coordinate system
$\hat{r}$	Distance, for which the comparison criterion has a maximum
r_b	Distance beyond which dispersion distortions become marked
r_bm	Distance, beyond which sound signals corresponding to two adjacent modes m and $m+1$ are distinguishable according to their propagation time in the channel
r_im	Distance, beyond which intramode signal distortions are marked
r_0	reference distance taken as 1 m
r_0^*	Reference distance
r_1	Transitional distance, beyond which spherical divergence of a wavefront ends and the "three by two" law for attenuation is valid

r^*	Distance beyond which the "three by two" law ceases to be satisfied
Δr_{d}	Discreteness of measurements of the sound field in terms of distance
Δr_j	Displacement of the jth hydrophone
$\vec{r}$	Radius vector in the plane
$\vec{R}$	Radius vector in space
$R_m = r/\cos\chi_m$	Length of the mth mode
$R_{\parallel}$	Longitudinal radius of correlation of inhomogeneities
$R_{\perp}$	Transverse radius of correlation of inhomogeneities
s	Sound loss coefficient in the bottom
$s(\vec{r}, t)$	Deviation of surface due to surface waves
S_{b}	Bottom scattering strength
$S(\omega)$	Emitted signal spectrum
$S(\vec{k})$	Spatial spectrum of sound field
$S(\vec{q})$	Spatial spectrum of internal wave field
$S(k, \omega)$	Space–frequency spectrum
$S(\vec{R}, \omega)$	Signal spectrum in waveguide in the point $(\vec{R})$
$S_{\mu\mathrm{m}}$	Scattering matrix
t Time	
t_m	Signal propagation time corresponding to mth mode
T	Geotime
$\hat{T}_l$	Time of propagation along ray cycle
T_{d}	Duration of emitted signal
$T_{m\mu}$	Mode transformation matrix for sound back scattering
$\hat{T}$	Accumulation (measurement) time
T_1	Period of first pulsation of gas bubble
T°	Temperature
u	Phase velocity of an internal wave
v	Speed of ship
$u(t)$	Arbitrary real signal
$U(t) = u(t) + iv(t)$	Analytic signal
$\vec{u}$	Horizontal component of particles velocity
$\vec{v}$	Velocity of particles in water (surface and internal waves)
V	Reflection coefficient from bottom
$V(t)$	Volume speed of pulsed exchange
V_ω	Spectral function of volume speed
w	Vertical component of particles velocity
w_{o}	Charge mass
$\hat{w}(\vartheta)$	Probability density of random process ϑ
W	Wronskian of a system of equations
W_0	Source power
W_l	Coefficient determining the proportion of sound field energy associated with the lth mode

x, y, z	Cartesian coordinate system
z_0	Depth of point sound source
z_j	Depth of source rank among sources of vertical radiating array
z_{i_r}	Depth of hydrophone rank among sources of vertical receiving array
z_s	Depth of plane of noise sources
α	Parameter determining the absorption coefficient in a model of a fluid bottom
β	Attenuation coefficient for sound in a waveguide, in the presence of a number of energybearing modes with the same attenuation coefficient
β_w	Sound attenuation coefficient in water
β_0	Interference invariant of sound field
β_1	Attenuation coefficient for sound in a waveguide, where the "three by two" law applies
δ	Relative sound field intensity in a waveguide
δ_r	Horizontal angle of refraction
Δ	Horizontal displacement of a ray on reflection from the bottom
$\zeta(\vec{r}, z, t)$	Vertical displacement of a surface of constant density
$\zeta_s(\vec{r}, t)$	Envelope of internal solitons
$\Theta(\varphi)$	Directivity function of internal wave field
κ	Propagation constant in the bottom for modes of the continuous spectrum
κ_l	Coefficient of attenuation in the bottom for the lth mode
λ	Length of a sound wave
$\Lambda_{ml} = \frac{2\pi}{\lvert q_m - q_l \rvert}$	Period of intermode beating $\left(\Lambda_l = \frac{2\pi}{\lvert q_l - q_{l+1} \rvert}\right)$
$\mu(r, z, t)$	Random component of the square of the refractive index in a waveguide
$\vartheta^{in}, \vartheta^{sc}$	Polar angles, defining the vertical planes, in which the wavevector of the incident and scattered sound waves lies, respectively
$\underset{\sim}{\xi}_m = q_m + i\frac{\gamma_m}{2}$	Eigenvalue of the mth mode
$\vec{\xi}$	Wavevector in the horizontal plane
ρ	Water density
ρ_1	Density in the bottom
σ_h	Root mean square of depth variation
$\tilde{\sigma}$	Value of the numerical criterion for comparing proximity
σ_l	Vertical wavenumber for the lth mode
$\vec{\sigma}$	Wavevector in the vertical plane
$\hat{\sigma}$	Parameter of Rayleigh distribution, describing random sound scattering by sea bottom

$\tau = 2\pi/\Delta\omega$	Pulse duration at the output of an optimal receiver (correlation detector) near sound source
τ_1	Pulse duration at the output of an optimal receiver (correlation detector) in the receiving point
$\hat{\tau}_l$	Time delay of sound pulse on bottom reflection
$\varphi(r, z, t)$	Signal phase
φ_m	Phase of mth mode
$\widetilde{\varphi}$	Angle between propagation directions of sound and internal waves
$\delta\varphi(t)$	Variations of signal phase
$\Phi_m(z)$	Internal wave mode with index m
χ_m	Grazing angle of a Brillouin ray corresponding to the mth mode
$\Psi_m(z)$	Sound waveguide mode with index m
$\Psi(\vec{R}, t; \vec{R}', t')$	Time depending Green function
$\Psi(r, z; \omega)$	Field of an acoustic point source of frequency ω, Green's function, waveguide transfer function
$\Psi_c(r, z)$	Field of a point source, corresponding to the continuous spectrum
ω	Cycle sound frequency $(\omega = 2\pi f)$
$\Delta\omega,\ \Delta f$	Bandwidth for signal frequency $(\Delta\omega = 2\pi\Delta f)$.
$\mathbf{\Omega}$	Cycle frequency of an internal and surface waves
$\mathbf{\Omega}_i$	Inertial frequency

References

Abbot P., Dyer I. (2002) "Sonar performance prediction incorporating environmental variability" in "Impact of littoral environmental variability on acoustic predictions and sonar performance." Kluwer Academic. Pages 611–618.

Abbot P., Dyer I, and Emerson C, (2006) Acoustic propagation uncertainty in the shallow East China Sea. IEEE, J. of Oceanic Engineering, 31, 368–383.

Abdullaev S.S., Niyazov B.A. Khabibullaev P.K. (1986) Spatial correlation of surface noise in an acoustic waveguide [DAN SSSR, 288, p. 223–225] (in Russian)

Ageeva N.S. (1974) Noise field of a lumpy source in the ocean In: Ocean acoustics (ed) L.M. Brekhovskikh. [Moscow, Nauka, p. 163–229] (in Russian)

Ageeva N.S., Krupin V.D. (1979) Structure of an infrasonic field in a shallow sea. [Sov. Phys. Acoust., Vol. 25(3), pp. 192–195]

Ageeva N.S. Krupin V.D. (1980) Effect of the bottom on the formation of the sound field in shallow water [Akusticheskii zhurnal, 26(2), p. 161–166] (in Russian)

Ageeva N.S., Krupin V.D. (1981) Frequency response curves for normal modes in a shallow sea with a layered absorbing bottom [Sov. Phys. Acoust., Vol. 27(5), pp. 372–376]

Ageeva N.S., Krupin V.D., Perelygin V.P. and Studenichnik N.V. (1994) Development of a geoacoustic model of shallow sea floor [Acoust. Phys., Vol. 40(2), pp. 159–165]

Aikal T. (1972) The relationship between the physical properties of underwater sediments that affect bottom reflection. Marine Geology, 13, 251–266.

Akulichev V. A., Bezotvetnykh V. V., Kamenev S. I., Kuz'min E. V., Morgunov Yu. N., and Nuzhdenko A. V. (2002) Acoustic Tomography of Dynamic Processes in a Sea Shelf Zone with the Use of Complex Signals [Acoust. Phys., Vol. 48(1), pp. 1–7]

Allen B., Stokey R., Austin T., Forrester N., Goldsborough R, Purcell M., von Alt C. (1997) REMUS: A small low cost AUV; System description field trails and performance results [Proceedings Oceans'97, Halifax, Canada]

Andreev M.Yu. Katsnel'son B.G., Kulapin L.G. and Petnikov V.G. (1996) Influence of hydrodynamic variations in a shallow sea on sound field phase [Acoust. Phys., Vol. 42(4), pp. 402–406]

Apel J.R., Badiey M., Berson J. et al. (1997) An overview of the 1995 SWARM shallow-water internal wave acoustic scattering experiment. IEEE J. Oceanic Eng. V.22(3), pp. 465–500.

Apel, J., Ostrovsky, L. and Stepanyants, Y. (1995) "Internal solitons in the ocean" APL/JHU Technical Report MERCJR0605.

Arase E.M, Arase T. (1967) Ambient sea noise in the deep and shallow water [Journal of the Acoust. Soc. Am., Vol. 42(2) pp. 73–77]

Aredov A.A. (1988), Anisotropy characteristics and spatial correlation of the noise field in the ocean [Sov. Phys. Acoust., Vol. 34(1), pp. 56–59]

Aredov A.A. and Furduev A.V. (1994) Angular and frequency dependencies of the bottom reflection coefficient from the anisotropic characteristics of a noise field [Acoust. Phys., Vol. 40(2), pp. 176–180]

Arons A.B. (1948), Secondary pressure pulses due to gas globe oscillation in underwater explosions: II. Selection of adiabatic parameters in the theory of oscillation. [J. Acoust. Soc. Am., Vol. 20, p. 277]

Artel'nyi V.V., Raevskii M.A. (1984) Statistical characteristics if normal modes in a waveguide with volume random inhomogeneities. [Izv. vuzov. Radiofizika 37(9), p. 1142– 1150] (in Russian)

Artel'nyi V.V., Raevskii M.A. (1987a) Multiple scattering of low-frequency sound waves by ocean turbulence [Sov. Phys. Acoust., Vol. 33(1), pp. 6–12]

Artel'nyi V.V., Raevskii M.A. (1987b) Multiple scattering of waves on anisotropic fluctuations of the refractive index in an infinite medium and refraction waveguides. [Izv. vuzov. Radiofizika 30(7), p. 866–874] (in Russian)

Ashley C.E., Siegman W.L., Jacobson M.J. (1987) Random-bottom structured and topographical effects on sound transmission in shallow channel [J. Acoustic Soc. Am., Vol. 81(3) p. 650–659]

Aster R.C., Borchers B., and Thurber C., (2005) "Parameter estimation and inverse problems", Elsevier, New York, 301 pages.

Avseth, P., Mukuvji, T., and Mavko, G., (2005), Quantitative Seismology Interpretation, Cambridge, University Press, 359 p.

Avilov K.V. (1995) Pseudodifferential parabolic equations of sound propagation in the slowly range-dependent ocean and their numerical solutions [Acoust. Phys., Vol. 41(1), pp. 1–7]

Avilov K.V. (2004) Modern techniques for practical computation of underwater sound fields [Proceedings of the 10th L.M.Brekhovskikh M.: GEOS , pp. 17–22 (in Russian)]

Babilov E.P., Kanevskii V.A. (1988) Sound scattering by a gas-filled spheroidal swim [Sov. Phys. Acoust., Vol. 34(1), pp. 11–13]

Babii V.V. (1983) Fine-scale structure of the sound speed field in the ocean. Leningrad, Gidrometeoizdat, 200 p. (in Russian)

Bachman, R.T. (1989), "Parameterization of geoacoustic properties" San Diego, CA: Scripps Institution of Oceanography.

Badiey, M., Katsnelson, B., Lynch, J., Pereselkov S., Siegmann, W., (2005) Measurement and modeling of 3-D sound intensity variations due to shallow water internal waves, J. Acoust. Soc. Am., Vol. 117, No. 2, pp. 613–625,

Badiey M., Katsnelson B. Lynch J., Pereselkov S., (2007) Frequency dependence and intensity fluctuations due to shallow water internal waves, J. Acoust. Soc. Am. Vol. 122, No.2, pp. 747–760

Badiey M., Bongiovanni K.P, Siegmann W.L. (1997) Analysis and model/data comparison of broadband acoustic propagation at the Atlantic Generating Station (AGS) site. [J. Acoustic Soc. Am., Vol. 101(4), pp. 1921–1935]

Badiey M., Jaya I. (1994) Shallow-water acoustic/geoacoustic/experiments at the New Jersey Atlantic Generating Station site [J. Acoustic Soc. Am. Vol. 96(6), pp. 3593–3604]

Baggeroer A.B., Kuperman W.A. (1992) Matched field processing in ocean acoustics In: Acoustic signal processing for ocean exploration , Ed Josa M.F. Moura and Isabel M.G. [Lourtie Kluwer Academic Publishers, Dordrecht–Boston–London, pp. 79– 114]

Baggeroer A.B., Kuperman W.A., Mikhalevsky P.N. (1993) An overview of matched field methods in ocean acoustics. IEEE Ocean Eng. Vol. 18, N4, 401–425

Barger J. E., Hamblen W. R. (1980) The air gun impulsive underwater transducer. [J. Acoust. Soc. Am., Vol. 68 (4), pp. 1038–1045]

Battle, D.J., Gerstoft, P., Kuperman, W.A., Hodgkiss, W.S., and Siderius, M., (2003) "Geoacoustic inversion of tow-ship noise via near field matched field processing", IEEE J. Oceanic Eng'g, 28(3), pp. 454–467.

Beardsley R.C. Brink K.H. Caruso M.J. et al. (1997) Shelfbreak PRIMER – an integrated acoustic and oceanographic field study in the Middle Atlantic Bight [Proceed. of the International conference on shallow-water acoustics. Beijing]

Beebe J.N. and McDanial S.T. (1980) Geoacoustic models of the seabed to support range-Scotian Shelf In: Bottom-interacting ocean acoustics, Ed. W.A. Kuperman and F.B. Jensen [Plenum Press, New York and London, pp. 507–523]

Beebe J. H., McDaniel S. T., and Rubano L. A., (1982) "Shallow-water transmission loss prediction using the Biot sediment model," [The Journal of the Acoustical Society of America 71(6), 1417–1426].

Belov A.I. and Komarov A.G. (1996), Mode analysis of the sound field in weakly irregular waveguides [Acoust. Phys., Vol. 42(20, pp. 232–233]

Belov A.I., Serebryany A.N., Zhuravlev V.A. (1997) Observation of internal wave effects on acoustic signals in a shallow sea with a weak thermocline [Proceed. of the International conference on shallow-water acoustics. Beijing]

Bel'kovich V.M., Yablokov A.V. (1972) Whales and dolphins [Moscow: Nauka 472p] (in Russian)

Bendat J.S., Piersol A.G (1986) Random data measurement procedures. John Wiley and Sons, 566 p.

Beran M.J. Whitman A.M. (1978) Scattering in the depth direction from a quadratic sound channel [J. Acoust. Soc. Am., Vol. 64(2), p. 658–664]

Beran M., Frankenthal S. (1996) Combined volume and surface scattering in a channel using a modal formulation. [J. Acoust. Soc. Am. **100** (3), 1463–1472]

Beran M., Frankenthal S. (1992) Volume scattering in a shallow channel. [J. Acoust. Soc. Am. **91**(6), 3203–3211]

Bergmann L. (1954) Ultrasound and its application in science and technology [Zurich] (in German)

Biot M., (1956) Theory of propagation of elastic waves in a fluid saturated porous solid. I. low frequency range.," [The Journal of the Acoustical Society of America 28(2),168–178.]

Biot M. (1956) Theory of propagation of elastic waves in a fluid-saturated porous solid II. High frequency range [J. Acoust. Soc. Am., Vol. 28(2), pp. 179–191]

Blakemore B., Sandell C. (2006), Seafood Supply Could Disappear in 42 Years, Effects of Overfishing and Other Factors Called Dire but Not Irreversible. ABC news, 11.03.2006

Bogolyubov B. N., Ostrovsky L. A., Okomelkov A. A., Slavinsky M. M. (1986) On resonant acoustic monopoles interaction. [Sov. Phys. Acoust. Vol. 32(1), pp. 15–21]

Bogolyubov B. N., Burlakova I. B., Dubovoy Yu. A., Marysev A. P., Farfel V. A. (2000) Low-frequency radiating complexes for hydro-acoustical monitoring of the ocean. [Proceeding of the U.S. – Russia workshop on experimental underwater acoustics. Nizhny Novgorod, pp. 48–60]

Box G., Jenkins G. (1970) Time series analysis. Forecasting and control [Holden-Day]

Bondar' L.F., Borisov S.V., Gritsenko A.V., Zakharov V.A., Il'ichev V.I., Kovzel'skiy D.G., Morgunov Yu.N. and Rutenko A.N. (1994) Intensity and phase fluctuations of low frequency acoustic signals along stationary tracks in shelf regions of the Sea of Japan [Acoust. Phys., Vol. 40(4), pp. 497–504]

Borodin V.V., Zhuravlev V.A., Kobozev I.K. and Kravtsov Yu.A. (1992) Average characteristics of sound fields in ocean waveguides [Sov. Phys.-Acoust., Vol. 38(4), pp. 332–336]

Bostrom A. (1984) Transmission and reflection of acoustic waves by an obstacle in a waveguide [Wave motion, V. 12(2), pp. 167–184]

Boutin P., Kemp J., Liberatore S., Lynch J., Witzell N., Metzger K., Webb D. (1989) Results of the Lake Seneca directivity source level and pulse response tests on the MIT 400 Hz Webb tomography sources [Technical Memorandum No. 1-89, Woods Hole Oceanographic Institution (WHOI-1-89)]

Bowden, (1983) "Physical oceanography of coastal waters" Ellis Horwood, Ltd., New York, 302 pages.

Boyles C.A. (1983) Coupled mode solution for a cylindrically symmetric oceanic waveguide with a range and depth dependent refractive index and a time varying rough sea surface [J. Acoust. Soc. Amer. V.73, pp. 800–805]

Brekhovskikh L.M. (1965) The average field in an underwater sound channel [Sov. Phys.-Acoust., Vol. 11(2), pp. 126–134]

Brekhovskikh L.M. (1980) Waves in layered media, 2nd Ed. [Academic Press, New York]

Brekhovskikh L.M. and Lysanov Yu. P. (1991) Fundamentals of Ocean Acoustics [Springer-Verlag, 270 p.]

Brekhovskikh L.M. and Godin O.A. (1992) Acoustics of layered medium [Springer- Verlag]

Brekhovskikh L. and Goncharov V. (1994) Mechanics of continued and wave dynamics [Springer-Verlag, 342 p.]

Buck J.R., Preisig J.C., Johnson M., Catipovic J. (1997) "Single-mode excitation in the shallow-water acoustic channel using feedback control". IEEE Journal of Oceanic Engineering, Volume: 22 Issue:2 , pp. 281–291.

Buckingham M.J. (1980) A theoretical model of ambient noise in a low loss, shallow water channel [J. Acoust. Soc. Am., Vol. 67(4), pp. 1186–1192]

Buckingham M.J. (1997) "Theory of acoustic attenuation, dispersion, and pulse propagation in unconsolidated granular materials including marine sediments." [*J. Acoust. Soc. Am.*, 102(5), pp. 2579–2596].

Buckingham M.J., Jones S.A. (1987) A new shallow ocean technique for determining the critical angle of the seabed from the vertical directionality of the ambient noise in the water column [J. Acoust. Soc. Am., Vol. 81(4), p. 938–946]

Bucker H.P. Morris H.E. (1968) Normal-mode reverberation in channels or ducts. [*J. Acoust. Soc. Am.* **44** (3), 827–828].

Bulatov M. G., Kravtsov Yu. A., Lavrova O. Yu., Litovchenko K. Ts., Mityagina M. I., Raev M. D., Sabinin K. D., Trohimovskii Yu. G., Churyumov A. N., Shugan I. V. (2003) Physical mechanisms of the aerospace radar imaging of the ocean [Physics-Uspekhi, Vol. 46(1), pp 63–79]

Bunkin F.V., Katsnel'son B.G., Kravtsov Yu.A., Kulapin L.G., Petnikov V.G., Rivelis E.A., Reznikov V.M, Sabirov O.I. and Sidenko A.V. (1989a) Average estimates of sound absorption in shallow ocean waveguides [Sov. Phys.-Acoust., Vol. 35(1), pp. 1–4]

Bunkin F.V., Gindler I.V., Kozel'skii A.R., Kravtsov Yu.A. and Petnikov V.G. (1989b) Dispersion distortion of complex acoustic signals in shallow water ocean waveguides [Sov. Phys. Acoust., Vol. 35(5), pp. 461–464]

Bunkin F.V., Vavilin A.V., Zhuravlev V.A., Kozel'skii A.R., Kravtsov Yu.A., Kuz'kin V.M., Kulapin L.G., Lyubchenko A.Yu., Omel'chenko N.N., Petnikov V.G., Sabirov O.I., Slavinskii M.M., Khudiev L.T. (1984) Investigation of bottom propagation of acoustic signals with highly stable [Izvestyia, Atmospheric and Ocean Physics, Vol. 20(2), pp. 159– 161]

Buntchuk A.V., Zhitkovskii Yu.Yu. (1980) Scattering of sound by sea in shallow region. *Sov. Phys. Acoust.* **26**(5), 363–370.

Burridge R., Weinberg G. (1977) Horizontal rays and vertical modes In: Wave propagation and underwater acoustics, ed J.B. Keller and J.S. Papadakis [Springer-Verlag, pp. 86–150]

Burlakova I,B., Saichev A.I. Slavinskii M.M. (1987) Spectrum of a Signal with the Elimination of Frequency Wandering [Sov. Phys, Acoust,, Vol. 33(3), pp. 236–241]

Burov V.A., Sergeev S.N. (1991) Processing of acoustic fields in oceanic waveguides with compensation of an unknown distortion of the profile of the vertical receiving antenna In Formation of acoustic fields in waveguides [Nizhnii Novgorod: Izd-vo IPF AN SSSR, p. 214–218] (in Russian)

Burov V.A., Sergeev S.N. (1992) Modern methods of perturbation theory in calculation of hydroacoustic fields [Vestn. Mosk. un-ta Seriya 3, Vol. 33(2), pp. 49–56] (in Russian)

Buryakovskii L.A., Dzhafarov I.S., Kerimov V.B. (1991) Search for and exploration of sea sites for petroleum and gas [Moscow, Nedra] (in Russian)

Bjorno L. (1981) Inhomogeneities and instabilities in underwater sound propagation In: Underwater acoustic signal processing ed. L.Bjorno [D. Reidel Publishing Company, Holland, pp. 29–40]

Cable P.G., Frech K.D., O'Connor J.C., Steel J.M. (1997) Reverberation-derived shallow-water bottom scattering strength. *IEEE J. Oceanic. Eng.,* **22**(3), 534–540.

Caille G.W., Dahl P.H. Zijun Gan et al. (1997) Overview of the joint China–U.S. 'Yellow Sea '96' experiment [Proceed. of the International conference on shallow-water acoustics. Beijing]

Candy J.V., Sallivan E.J. (1993) Sound Velocity profile estimation: A system theoretic approach [IEEE Journal of Oceanic Engineering, Vol. 18(3), pp. 240–252]

Carbone N.M., Deane G.B., Buckingham M.J. (1998) Estimating the compressional and shear wave speeds from the vertical coherence of ambient noise in the water column [J. Acoust. Soc. Am. Vol. 103(2), pp. 801–813]

Carey W. and Evans R. (1998) "Frequency dependence of sediment attenuation in two low-frequency shallow-water acoustic experimental data sets," [J. Oceanic Eng. 23(4), 439–447].

Carter, G., Knapp, C., Nuttall, A. (1973). "Estimation of the magnitude-squared coherence function via overlapped fast Fourier transform processing". Audio and Electroacoustics, IEEE Transactions on . Vol: 21 Issue:4 , page(s): 337–344.

Capon J. (1969) "High resolution frequency-wavenumber spectrum analysis" [Proc. IEEE, Vol. 57, P. 1408–1418]

Carey W., Doutt J., Evans R., and Dillman L. (1995), "Shallow-water sound transmission measurements on the New Jersey continental shelf," [J. Oceanic Eng. 20(4), 321–336].

Carey, W. M., Lynch, J. F., Siegmann, W. L., Rozenfeld, I., and Sperry, B. J. (2006), Sound transmission and spatial coherence in selected shallow-water areas: measurement and theory, Journal of Computational Acoustics, V14, No. 2, pp. 265–298.

Gasparovic R.F., Etkin V.S. (1994) An overview of the Joint US/Russia Internal wave Remote Sensing Experiment. IGARSS'94. Digest, pp. 741–743.

Cavanagh R.G., Renner W.W. (1980) Vertical directionality and depth dependence of averaged acoustic signals and noise [J. Acoust. Soc. Am. Vol. 68(5), pp. 1467–1474]

Chapman, N.R., Dizaji, R.M., and Kirlin, L. (2006) "Inversion of geoacoustic model parameters using ship radiated noise." In "Acoustic sensing techniques for the shallow water environment," A. Caiti, N.R. Chapman, J.P. Hermand, and S. Jesus (eds), pp. 289–302.

Chen C.T., Millero F.J. (1977) Sound speed of seawater at high pressures. [J. Acoust. Soc. Am., Vol. 62(3), pp. 1129–1135]

Chernov L.A. (1975) Wave propagation in a medium with random inhomogeneities [Moscow, Nauka, 172 p.] (in Russian)

Chuprov S.S. (1982) Interference structure of the sound field in a stratified ocean In: Ocean acoustics. Current status (ed) L.M. Brekhovskikh [Moscow, Nauka, p. 71–91] (in Russian)

Chuprov S.D. and Shvachko R.F. (1974) in *Acoustics of the Ocean*, Ed. by L. M. Brekhovskikh , Nauka, Moscow, pp. 559–614.

Chuprov, S. D.(1982) "Interference structure of sound field in a layered ocean," in *Akustika okeana: Sovremennoe sostoyanie,* edited by L. M. Brehovskikh, Moscow: Nauka, pp. 71–91.

Clay C.S., and Medwin H. (1977) Acoustical oceanography: principles and applications [John Wiley & Sons, Inc.]

Cole B., (1965) "Marine sediment attenuation and ocean-bottom-reflected sound," [The Journal of the Acoustical Society of America 38(2), 291–297].

Collins M, Ching-bin S. (1990) Three-dimensional parabolic equation that includes rough boundaries. JASA 87, 1104–1109

Collins M.D. (1993) A split-step Pade solution for parabolic equations method, [J. Acoust Soc. Am. V.93(4.1), pp. 1736–1742]

Collins M.D. (1993) The adiabatic mode parabolic equation [J. Acoust. Soc. Amer. V.]

Collins M., Kuperman W., and Schmidt H., "Nonlinear inversion for ocean bottom properties", J. Acoust. Soc. Am., 92(5), pp. 2770–2783 (1992).

Collins M.D. Cederberg R.J., King D.B., Ching-Bing S.A. (1996) Comparison of algorithms for solving parabolic wave equations [J. Acoust Soc. Am. V.100(1), pp. 178–182]

Colosi J.A., Baggeroer, A.B. (2004) On the kinematics of broadband multipath scintillation and the approach to saturation. Acoust. Soc. Am. V. 116, N5, pp. 3515–3522

Colosi J.A.,. Lynch J.F,. Beardsley R.C, Gawarkiewicz G., Scotti A., and.Chiu C.S, (2001) "Observations of nonlinear internal waves on the New England continental shelf during summer shelfbreak PRIMER." J. Geophys. Res. v. 106, C5, pp 9587–9601.

Colosi J. and Brown M., (1998) Efficient numerical simulation of stochastic internal-wave-induced sound-speed perturbation fields, J. Acoust. Soc. Am. Volume 103, Issue 4, pp. 2232–2235

Cooley J. W., Tukey J.W. (1965) An algorithm for the machine calculation of complex Fourier series, *Math. Comput.* **19**: 297–301.

Creamer D. (1996) Scintillating shallow-water waveguides. [J.Acoust.Soc.Am. **99**(5), 2825–2838]

Cron B. and Sherman C H. (1962) Spatial-correlation functions for various noise models [J. Acoust. Soc. Am., Vol. 34(11), pp. 1732–1736]

Crouch W.W., Burt P.J. (1972) The logarithmic dependence of surface generated ambient shelf noise spectrum level on wind-speed [J. Acoust. Soc. Am., Vol. 51(3), pp. 1066–1072]

Cushman-Roisin B., (1994) "Introduction to geophysical fluid dynamics", Prentice Hall, Englewood Cliffs, 320 pages.

Davydov A.S. (1965) Quantum mechanics [Translated, edited and with additions by D. ter Haar, 2nd ed. Oxford; New York: Pergamon Press]

Dacol D.K. (1997) Coupled-amplitude approach to solving the Helmholtz equation [J. Acoust. Soc. Am., Vol. 101(5), pp. 2566–2570]

Dahl P., Zhang R., Miller J., et al., (2004) "Guest editorial: Overview of results from the Asian Seas International Acoustics Experiment in the East China Sea," [J. Oceanic Eng. 29(4), 920–928]

Dargeiko M.M., Kravtsov Yu.A., Petnikov V.G., Petrosyan A.S., Samoilenko Yu.I., Slavinskii M. M. (1984) Unique features of radiation field focusing in multimode waveguide channels. [*Radiophysics and Quantum Electronics* **27**(6), pp. 516–522].

Daugherty J.R., Lynch J.F. (1990) Surface wave, internal wave, and source motion effects on matched field processing in a shallow water waveguide [J. Acoust. Soc. Am., Vol. 87(6), pp. 2503–2526]

Dawson T.W., Fawcett J. (1990) A boundary integral equation method for acoustic scattering in waveguide with nonplanar surfaces [J. Acoust. Soc. Am. Vol. 87(3), pp. 1110– 1125]

Denham R.N. (1986) Intensity decay laws for near-surface sound sources in the ocean [J. Acoust. Soc. Am., Vol. 79(1), pp. 60–63]

Densmore C. Dana, (1995) "The A Boat",. The Village Printer, Falmouth, MA. 195 pages.

Derevnin V.A., Kur'yanov B.F., Lukashev V.D. et al. (1986) Acoustic bottom stations for studying signals and noise in a deep ocean [Deep-water systems and complexes, Cherkassy, Proc. Seminar 'Deep-water systems and complexes', p. 52–55] (in Russian)

Derevyagina E.I., Katsnel'son B.G. and Lyubchenko A.Yu. (1994) Vertical structure of the intensity of low-frequency noise in shallow water [Acoust. Phys., Vol. 40(3), pp. 343–346]

Derevyagina E.I. and Katsnel'son B.G. (1995) The Effect of random inhomogeneities on the vertical directionality of surface noise in shallow water [Acoust. Phys., Vol. 41(2), pp. 205–209]

Derevyagina E.I. Katsnel'son B.G. (1988a) Influence of an absorbing bottom on the frequency spectrum of the noise field in a shallow sea [Sov. Phys.-Acoust., Vol. 34(1), pp. 95–97]

Derevyagina E., Katsnel'son B.G. (1988b) Effect of the shape of the region of distribution of surface noise sources on the spectrum of the noise field in shallow water [Sudostroitel'naya promyshlennoct'. Seriya 'Akustika' 3, p. 50–53] (in Russian)

DeRuiter S., Tyack, P., Lin, Y.T., Newhall, A.E., and Miller, J.O. (2006) , "Modeling acoustic propagation of airgun array pulses recorded on tagged sperm whales (*Physeter macrocephalus*)". *J. Acoust. Soc. Am.* Volume 120, Issue 6, pp. 4100–4114

DeSanto J.A. (ed) (1979) Ocean Acoustics [Springer-Verlag]
Diachok O.I., Smith P., Wales S. (2004) Classification of bioacoustic absorption lines at low frequencies: the Van Holiday connection [J. Acoust. Soc. Am. Vol. 115, pp. 2521]
Dickey T.D. (1991) The emergence of concurrent high-resolution physical and biological measurements in the upper ocean and their application [Rev. Geophys. Res., vol. 29(3), pp. 383–414]
Donelan M., Hamilton, J, and Hui, W.H. (1985) The directional spectra of wind generated waves. *Phil. Trans. R. Soc. Lond.* **A-315,** pp. 509–582.
Donskoy D. M., Blue J. E. (1994) A new concept of a low-frequency underwater sound source [J. Acoust. Soc. Am., Vol. 95(4), pp. 1977–1982]
Doolittle R., Tolstoy A., Buckingham M. (1988) Experimental confirmation of horizontal refraction of cw acoustic radiation from a point source in wedge-shaped ocean environment [J. Acoust. Soc. Am., Vol. 83(6), pp. 2117–2125]
Doolittle R.D. et al. (ed) (1993) Special issue on detection and estimation in matched-field processing, IEEE Journal of Oc. Eng, Vol. 18, N8,
Dosso S.E (2002) "Quantifying uncertainty in geoacoustic inversion. I. A fast Gibbs sampler", J. Acoust. Soc. Am. 111(1), pp. 129–142 (2002)
Dosso E.D., Yeremy M.L., Ozard J.M., Chapman N.R. (1993) Estimation of ocean bottom properties by matched field inversion of acoustic field data [IEEE Journal of Oceanic Engineering, Vol. 18(3), pp. 232–239]
Dosso S.E. and Nielsen P.L. (2002) "Quantifying uncertainty in geoacoustic inversion. II. Application to broadband, shallow-water data." *J. Acoust. Soc. Am.* Vol. 111, Issue 1, pp. 143–159.
Dowling D.R. (1993) Phase-conjugate array focusing in a moving random media. [J. Acoust. Soc. Am. **94**, pp. 1716–1718].
Dowling D.R. (1994) Acoustic pulse compression using passive phase-conjugate processing. *J. Acoust. Soc. Am.* **95**, pp. 1450–1458.
Dozier L.B., Tappert F.D. (1978) Statistics of normal mode amplitudes in a random ocean. 1. Theory [J. Acoust. Soc. Am., Vol. 63(2), pp. 353–365]
Duda T.F., Collis J.M., Lin Y.-T., Newhall A.E., Lynch J.F., and DeFerrari H.A. (2011). "Horizontal coherence of low-frequency fixed-path sound in a continental shelf region with internal-wave activity," in press *J. Acoust. Soc. Am.*
Duda T.F. and Farmer, D. (1999) "The 1998 WHOI/IOS/ONR internal solitary wave workshop: contributed papers." WHOI Tech Report WHOI-99-07.
Dungan M.R. and Dowling D.R. (2000) "Computed narrow-band time-reversing array retrofocusing in a dynamic shallow ocean." *J. Acoust. Soc. Am.* 107(6), pp. 3101–3112.
Dyer I. (1970) Statistics of sound propagation in the ocean. J.Acoust.Soc.Am. v.48, N1B, pp 337–345.
Eliseevnin V.A. (1981) Operation of a vertical line array in a water layer [Sov. Phys. Acoust., Vol. 27(2), pp. 125–127
Eliseevnin, V. A.; Tuzhilkin, Yu. I. (2001) Acoustic Power Flux in a Waveguide. Acoustical Physics V. 47 N6, pp 688–694
Ellis D.D. (1995) A shallow-water normal-mode reverberation model [*J. Acoust. Soc. Am.* **97** (5), 2804–2814].
Emery W., Thomson R. (1997) Data analysis methods in physical oceanography. Elsevier, Amsterdam, Netherlands, 634 pp
Essen H.H., Schirmer F., Sirkes S. (1987) Acoustic remote sensing of internal waves in shallow water [Int. J. Remote Sensing, Vol. 4(4), pp. 33–47]
Essen H.H. Schirmer F., Schmalfeldt B. (1983) Influence of tides on acoustic continuous wave (CW) propagation in shallow water [Acustica, Vol. 48(5), pp. 285–292]
Ewing W. M., Jardetzky W. S., Press F. (1957), Elastic Waves in Layered Media,. McGraw-Hill, 380 p

Farmer D. and Vagle, S (1989). "Waveguide propagation of ambient sound in the ocean-surface bubble layer," *J. Acoust. Soc. Am.* Volume 86, Issue 5, pp. 1897–1908

Fedorov K.N. (1976) Fine thermohaline structure of oceanic water Leningrad Gidrometeoizdat 184p. (in Russian)

Fedorov K.N. (1978) Fine structure of hydrophysical field in the ocean Oceanic physics Moscow: Nauka 1 Gidrofizika pp. 113–147 (in Russian)

Felsen L., Marcuvitz N. (1973) Radiation and scattering of waves. Wiley-IEEE, 888 p.

Field R., Mire C., Chandler H., Broadhead M. (1997) The SEASAME Experiments – the effects of internal solitons on the Marine shelf [Proceed. of the International conference on shallow-water acoustics. Beijing]

Fink M. (1995) Time Reversal Mirrors. In *Acoustical Imaging,* Vol. 21, edited by J.P. Jones (Plenum, New York), pp. 1–15.

Fizell R.G. (1987) Application of high-resolution processing to range and depth estimation using ambiguity function methods. [Journal of the Acoust.Soc. Am. Vol. 82, N2, P.606–613]

Flatte S.M. (ed) (1979) Sound transmission through a fluctuating ocean [Cambridge University Press]

Fock V.A. (1965) *Electromagnetic Diffraction and Propagation Problems* . Pergamon, New York, pp. 213–234.

Fokin V. N., Fokina M. S. (2003) Estimation of sea bottom parameters from angle-frequency resonances of the reflection coefficient. [Acoust. Phys., Vol. 49(5), pp. 584–591]

Forsithe G., Malcolm M., Moler C. (1977) Computer methods for mathematical computations [Prentice-Hall, Inc.]

Focke K, Weston D. (1978) Problem of the caustic in range averaged ocean sound channels [J. Acoust. Soc. Am. Suppl., Vol. 23 (1) p. 64]

Frankenthal S. (1998) Propagation in random stratified waveguides – A modal spectral treatment. J. Acoust. Soc. Am. **104**(6), 3282–3295

Fredericks, A., Colosi, J.A., Lynch, J.F., Gawarkiewicz, G., Chiu, C.-S., and Abbott, P. (2005) "Analysis of multipath scintillations from long range acoustic transmission on the New England continental slope and shelf," J. Acoust. Soc. Am., V.117, N3, pp.1038-1057,

Frisk G.V. (1988) Inverse methods in ocean bottom acoustics [WHOI, Woods Hole, MA, USA, 32 p]

Frisk G.V., (1994) "Ocean and Seabed Acoustics", Prentice Hall, Saddle River, N.J., 299 pp

Frisk, G.V., Talk at ASA meeting (2005)

Frisk G. and Lynch J., "Shallow Water Waveguide Characterization using the Hankel Transform," J. Acoust. Soc. Am., 76(1), 205–216, July (1984).

Furduev A V. (1966) Undersurface cavitation as source of noise in the ocean [Izvestyia Atmospheric and Oceanic Physics, V. 2(5) pp. 314–320]

Furduev A.V. (1974) Oceanic Noise. In Oceanic acoustics (ed) L.M. Brekhovskikh Moscow, Nauka, p. 615–691 (in Russian)

Gabor D. (1946), "Theory of Communications," Trans. Inst. Electr. Eng., vol. 3, pp. 429–456.

Galkin O. P., Popov R. Yu., and Simakina E. V. (2004) Spatial Correlation of Sound Fields from Underwater Explosions in the Barents Sea [Acoust. Phys., Vol. 50(1), pp. 30–36]

Garrett C., Munk W.H. (1975) Space–time scales of internal waves: a progress report [J. Geophys. res. Vol. 80, p. 291]

Gasparovic R.F., Etkin V.S. (1994a) An overview of the Joint US/Russia Internal wave Remote Sensing Experiment. IGARSS'94. Digest, pp. 741–743. (Repetition of 86)

Gasparovic R.V., Etkin V.S. (1994b) An overview of the Joint U.S./Russia Internal Wave Remote Sensing Experiment [Proceed. of the IGARSS'94 Pasadena pp. 744–746]

Gawarkiewicz G., Bahr F., Brink K. . . Beardsley R.C, Caruso M., Lynch J, and Chiu C.S., (2004) "A large amplitude meander of the shelfbreak front during summer south of New England: Observations from the shelfbreak PRIMER experiment." , J. Geophys. Res.109.

Gawarkiewicz G. and Chapman, D. (1992) "The role of stratification in the formation and maintenance of shelf-break fronts." *J. Phys. Oceanogr.* **22(7),** pp. 753–772.

Gazarian Yu.L. (1958) The sound field generated by a point source in a lying on a half-space [Sov. Phys. Acoust., Vol. 4(3), pp. 237–242]

Gershfeld D.A., Eller A.L. (1985) Geometric considerations in determining the optimum frequency of acoustic propagation in a shallow water waveguide [J. Acoust. Soc. Am., vol. 78(2), pp. 632–641]

Gindler I.V. (1990) Perturbation theory for plane-layered open acoustic waveguides [Sov. Phys. Acoust., vol. 36(4), pp. 349–352]

Godin O.AS. (1998) A note on differential equations of coupled-mode propagation in fluids J. Acoust. Soc. Am., vol. 103(1), pp. 159–169

Golubev V.N., Lazarev V.A., Sharonov G.A. (2007) Frequency dependence of spatial correlation of sound field in inhomogeneous shallow water waveguide [Proceedings of the 19th meeting of RAS M.: GEOS, Vol. 2, pp. 246–249 (in Russian)]

Gonorovskii I.S. (1977) Radio-engineering chains and signals [Moscow, Sov. Radio] (in Russian)

Gordienko V.A. Krasnopistsev N.V., Nasedkin A. V. Nekrasov V. N. (2007) Estimation of the detection range of a hydroacoustic system based on the acoustic power flux receiver [Acoust. Phys. **53**(6), pp. 721–729.

Gorskaya N.V., Nikolaev G.N., Rychova T.A., Salin B.M. (1981) Spectral analysis in the investigation of fields of harmonic sources in acoustic waveguides [Sov. Phys. Acoust., Vol. 27(2), pp. 110–112]

Gorskaya N.S., Raevskii M.A. (1984) Influence of a Random Internal-Wave field on sound propagation in the ocean [Sov. Phys. Acoust., Vol. 30(2), pp. 108–113]

Gorskaya N.V, Gorskii S.M., Zverev V.A., Nikolaev G.N., Kurin V.V., Khil'ko A.I. (1998) Short-wave diffraction in a multimode layered waveguide [Sov. Phys. Acoust. vol. 34(4), pp. 29–32]

Gorskaya N.S., Raevskii M.A. (1990) Low-frequency sound field in an ocean waveguide with an irregular bottom [Sov. Phys. Acoust., Vol. 36(3), pp. 235–237]

Grachev G.A. (1983) Specific characteristics of signal attenuation in a shallow sea [Sov. Phys. Acoust., Vol. 29 (2), pp. 160–161]

Grachev G. A. (1993) Theory of acoustic field invariants in layered waveguides. *Acoust. Phys.* **39**(1), 33–35.

Graves R.D., Nagl A., Uberall H. (1975) Range dependent normal modes in underwater sound propagation [J. Acoust. Soc. Am., vol. 58(4), pp. 1171–1177]

Grigor'ev V.A., Katsnelson B.G., Petnikov V.G. (1996) Frequency dependence of the effective bottom sound absorption in the Barents Sea [Acoust. Phys., Vol. 42(5), pp. 627– 629]

Grigor'ev V.A., Katsnel'son B.G., Kuz'kin V.M., Petnikov V.G., (2001) Characteristics of the diffraction of acoustic waves in stratified sound channels. Acoust. Phys. **47**(1), 35–41.

Grigor'ev V.A., Kuz'kin V.M., Petnikov V.G., (2004) Low-frequency bottom reverberation in shallow water ocean regions. *Acoust. Phys.* **50**(1), 37–45.

Grigor'ev V. A, Katsnel'son B. G., and Petnikov V. G. (2001) Determination of the absorbing and scattering properties of the sea floor in a shallow water environment by the spectra of wide-band signals. Acoust. Phys. **47**(3), 277–281.

Guthrie K.M., Tindle C.T. (1976) Ray effects in the normal mode approach to underwater acoustics. J. Sound and Vibr., v.47, pp. 403–413

Gurevich A.V., Tsedilina E.E. (1979) Ultra long range propagation of short waves in the ionosphere [Moscow, Nauka, 243 p.] (in Russian)

Gorskii S.M. Zverev V.A. Matveev A.L., Mityugov V.V. (1995) Incoherent accumulation of acoustic diffraction signals [Acoust. Phys. vol. 41(2), pp. 190–197]

Goff, J. A., Jordan, T. H. (1988). Stochastic modeling of seafloor morphology: Inversion of sea beam data for second-order statistics. Journal of Geophysical Research-Solid Earth and Planets. Vol. 93 (B11), pp. 13589–13608.

Hackman R.H., Sammelmann G.S. (1988) Multiple-scattering analysis for a target in an oceanic waveguide [J. Acoust. Soc. Am., Vol. 84(5), pp. 1813–1825]

Hall M.V. (1995) Depth variation of acoustic horizontal refraction theory a cold-core ocean eddy. In: Full field method in ocean and seismo acoustics ed. Diachok O., Kluwer [Ac. Publ. pp. 267–272]

Hall J., Watt J. (ed) (1976) Modern numerical methods for ordinary differential equations [Clarendon Press Oxford]

Hamilton E.L. (1985) Sound velocity as a function of depth in marine sediment [J. Acoust. Soc. Am., Vol. 78(4), pp. 1348–1355]

Hamilton E. et al., (1956) "Acoustic and other physical properties of shallow-water sediments of San Diego," The Journal of the Acoustical Society of America 28(1), 1–15

Hamilton E. L., (1976) Sound attenuation as a function of depth in the sea floor. J. Acoust. Soc. Am. **59**(3), 528–535.

Hamilton E.L. (1980) Geoacoustic modeling of the sea floor J. Acoust. Soc. Am. V.68, N5, pp. 1313–1340

Hampton L.D. (ed) Physics of sound in marine sediments [Plenum Press, New York, (1973)]

Hamson R.M. (1985) The theoretical response of vertical and horizontal line arrays to wind induced noise in shallow water [J. Acoust. Soc. Am. Vol. 78(5),pp. 1702–1709]

Hamson R.M., Heitmeyer R.M. (1989), Environmental and system effects on source localization in shallow water by the matched-field processing of a vertical array. Journal of the Acoust.Soc. Am. Vol. 85, N5, P.1950–1959

Harrison C.H., Harrison J.H. (1995) A simple relationship between frequency and range averaging for broadband sonar. J.Acoust.Soc. Am. V.97, pp. 1314–1317

Harrison, C.H. and Simons, D.G. (2002) "Geoacoustic inversion of ambient noise: a simple method." J. Acoust. Soc. Am. 112(4), pp. 1377–1389.

Hasselmann, D.E., Dunckel, M., and Ewing, J. (1980) Directional spectra observed during JONSWAP 1973. *J. Phys. Oceanogr.* **10,** pp. 569–573.

Headrick R.H., Lynch J.F., Kemp J.K., Newhall A.E., v.d. Heydt K., Apel J., Badiey M., Ching-sang Chiu Finette S., Orr M., Pasewark B., Turgot A., Wolf S., Tielbuerger D. (2000) Acoustic normal mode fluctuation statistics in the 1995 SWARM internal wave scattering experiment [J. Acoust. Soc. Am., Vol. 107(1), pp. 201–220], Modeling mode arrivals in the 1995 SWARM experiment acoustic transmissions [J. Acoust. Soc. Am., Vol. 107(1), pp. 221–236]

Heaney, K.D. and Cox, H., (2002) "Rapid geoacoustic characterization for limiting environmental uncertainty for sonar system performance prediction." In "Impact of littoral environmental variability on acoustic predictions and sonar performance," N. Pace and F. Jensen (eds.), pp. 163–170, Kluwer

Heaney, K.D., (2004) Rapid geoacoustic characterization: applied to range dependent environments." IEEE J. Oceanic Eng'g. 29(1), pp. 43–50

Henriques T.A., Young A.M. (1980) The Helmholtz resonator as high-power deep-submergence source for frequencies below 500 Hz. [J. Acoust. Soc. Am., Vol. 67(5), pp. 2330–2342]

Hinish M.J. (1973) Maximum-likelihood signal processing for vertical array. [Journal of the Acoust. Soc. Am. Vol. 54, N2, P. 499–503]

Hodgkiss W.S., Song H.C., Kuperman W.A., Akal T., Feria C. and Jackson D.R. (1999) "A long-range and variable focus phase-conjugation experiment in shallow water." *J. Acoust. Soc. Am.* 105(3), pp. 1597–1604.

How-Kin Wong, Chesterman W.D. (1968) Bottom backscattering near grazing incidence in shallow water. *J. Acoust. Soc. Am.* **44** (6), 1713–1718.

Ingenito F., Wolf S.N. (1976) Acoustic propagation in shallow water overlying a consolidated bottom [J. Acoust. Soc. Am., Vol. 60(3), pp. 611–617]

Ingenito F. (1987) Scattering from an object in a stratified medium [J. Acoust. Soc. Am., Vol. 82 (6), pp. 2051–2059]

Ingenito F., (1973) "Measurements of mode attenuation coefficients in shallow water," [The Journal of the Acoustical Society of America 53(3), 858–863.]

Isakovich M.A. (1973) General acoustics, Moscow, Nauka, (in Russian), 496 p.

Ishimaru A. (1978) Wave propagation and scattering in random media [Academic Press, New York]

Ivakin A.N. and Lysanov Yu. P., (1985) Backscattering of sound from inhomogeneous bottom at small grazing angles. Sov. Phys. Acoust. **31**(3), 236–237.

Iwase R., Tsuchiya T., Kikuchi T., Mizutani K. (2006) A method of fix point displacement measuring on seafloor by using phase conjugate underwater acoustic wave [Proceeding of the Eighth European conference on underwater acoustics, Carvoeiro, Portugal, P. 645–650]

Jackson D.R., Briggs K.B. (1992) High-frequency bottom backscattering: Roughness versus sediment volume scattering. [J. Acoust. Soc. Am. **92** (2), 962–977].

Jackson D.R., Dowling D.R. (1991) Phase conjugation in underwater acoustics. [J. Acoust. Soc. Am. **89**, pp. 171–181.]

Jackson D.R., Dowling D.R. (1992) Narrow-band performance of phase-conjugate arrays in dynamic random media. [J. Acoust. Soc. Am. **91**, pp. 3257–3277].

Jackson, C and Apel, J. (2002) "An atlas of internal solitary-like waves and their properties", Global Ocean Associates report.

Jackson, C. (2004) "An atlas of internal solitary-like waves and their properties (second edition)", Global Ocean Associates report.

James, I.D. (1977) A model of the annual cycle of temperature in a frontal region of the Celtic Sea. Estuar. Coastal Marine Sci. **5,** pp. 339–353.

Jason D. Holmes, William M. Carey, Allan D. Pierce, James F. Lynch. (2005) "Attenuation Characteristics of Sandy Sediments- A simplified Biot approach." Proceedings of the boundary influences in high frequency, shallow water acoustics meeting. University of Bath, UK.

Jensen F.B. (1981) Sound propagation in shallow water [J. Acoust. Soc. Am. Vol. 70(5), pp. 1397–1406]

Jensen F.B., Ferla M.C. (1979) SNAP: the SACLANTSEN Normal-Mode Acoustic Propagation Model, SACLANTSEN Memorandum SM-121, La Specia

Jensen F.B., Kuperman W.A. (1983) Optimum frequency by sound propagation in shallow water [J. Acoust. Soc. Am., Vol. 73(3), pp. 813–819]

Jensen F.B., Kuperman W. Sound propagation in a wedge-shaped ocean with penetrable bottom. JASA, 67, 1564–1566, 1980

Jensen F., Kuperman W., Porter M., and Schmidt H. (1994). "Computational Ocean Acoustics", AIP Press, 612 pages

Jin G., Lynch J.F., Pawlowicz R., Wahdhams P. (1993) Effects of the sea cover on acoustic ray travel times, with applications to the Greenland Sea tomography experiment. JASA, V.94(2) pp. 1044–1057.

Jin G.L., Zhang R.H. (1990) The numerical simulation of the average reverberation intensities in shallow water. *Acta Acust*. **9**, 36–44.

Jin G., Lynch, J.F., Chiu, C.S. and Miller, J.H. (1996) "A theoretical and simulation study of acoustic normal mode coupling effects due to the Barents Sea Polar Front, with applications to acoustic tomography and matched-field processing." *J. Acoust. Soc. Am.* Volume 100, Issue 1, pp. 193–205 .

Karnovskii A.M., Loginov K.A., Rivelis E.A., Freiman M.E. (1981) A model of the noise field in a waveguide [Sov. Phys. Acoust., Vol. 27 (5), pp. 417–420]

Katsenelenbaum B.Z. (1961) On the theory of irregular acoustic waveguides [Sov. Phys. Acoust., v. 7, pp. 159–164]

Katsenelenbaum B.Z. (1960) Theory of irregular waveguides with slowly varying parameters [Moscow Izd. AN SSSR, 216 p.] (in Russian)

Katsnelson B.G. , Pereselkov S.A. (1996) Shallow water internal wave field model for underwater acoustics [Proceedings of the Int. Conference OCEANS'96 MTS/IEEE, Florida, USA, pp. 27–31]

Katsnelson B.G., Kulapin L.G., (1984) Average Sound-Intensity decay law in an irregular underwater acoustic waveguide [Sov. Phys. Acoust., Vol. 30(5), pp. 379–382]

Katsnelson B.G., Kravtsov, Yu. A., Kulapin, L.G., Petnikov, V.G. and Sabirov O.I. (1985) Average decay law of sound in an irregular near-bottom sound channel [Sov. Phys. Acoust., Vol. (4), pp. 321–322]

Katsnelson B.G., Kulapin L.G., Migulin A.A., Petnikov V.G. (1992) Influence of hydrodynamic fluctuations on the vertical interference structure of a sound field in a waveguide [Sov. Phys. Acoust., Vol. 38(2), pp. 164–168]

Katsnelson B.G., Kuz'kin V.M., Pereselkov S.A., Petnikov V.G. (1998) Sound wave attenuation in shallow water with rough boundaries [ICA/ASA Proceedings, Seattle pp. 2713–2714]

Katsnel'son B.G., Kravtsov, Yu. A., Kuz'kin V M, Kulapin L.G., Petnikov V.G. (1984) Simplified theory of sound propagation in shallow water [Trudy FIAN, 156, Issledovaniya po gidrofizike, p. 41–55] (in Russian)

Katsnel'son B.G. Sidenko A.V. (1988) Decay in radiation intensity in a multimode waveguide with random inhomogeneities and absorbing boundaries Radiofizika 31(4) pp. 433–438 (in Russian)

Katsnel'son B.G., Kravtsov Yu.A., Petnikov V.G. The Primary methods of acoustic propagation theory in stratified horizontally inhomogeneous absorbing boundary waveguides In: Oceanic Remote Sensing, Proceeding of the Institute of General Physics Academy of the Sciences of the USSR [Nova Science Publisher, New York, (1987) Vol. 1, pp. 176–217]

Katsnelson B.G., Pereselkov (2000) Horizontal refraction of low frequency sound field due to soliton packets in shallow water. Acoustical Physics,

Katsnelson B.G., Pereselkov S.A., Petnikov V.G. (2004a) On the feasibility of normal wave selection in a shallow-water waveguide. Acoust. Phys. V.50(5), pp. 646–636.

Katsnelson B.G., Pereselkov S.A., Petnikov V.G., Sabinin K.D. (2004b) Effect of background internal waves on sound propagation on ocean shelf. Proceedings of the X^{-th} L.M.Brekhovskikh's conference "Ocean Acoustics", 106–109.

Katsnelson B.G., Petnikov V.G. (2002) Shallow water acoustics. Springer-Verlag.

Katsnelson B., Lynch J., and Tsoidze A. (2007) "Space-frequency distribution of sound field intensity in the vicinity of a temperature front in shallow water", [Acoustical Physics, 53(5), 611–617].

Keller J.B. and Papadakis J.S. (1977) Wave propagation and underwater acoustics [Springer-Verlag, 287 p.]

Kennett J. (1982) "Marine Geology", Prentice Hall, Englewood, NJ. 813 p

Kibblewhite A.C. (1989) "Attenuation of sound in marine sediments: A review with emphasis on new low-frequency data," The Journal of the Acoustical Society of America 86(2), 716–738

Kibblewhite A.C., (1989) "Attenuation of sound in marine sediments: A review with emphasis on new low-frequency data," The Journal of the Acoustical Society of America 86(2), 716–738.

Kibblewhite A. and Denham C. (1966), Experiment on Sound Propagation in Shallow Water under Isovelocity Conditions, *J. Acoust. Soc. Am.* Volume 40, Issue 6, pp. 1337–1344

Kim S, Kuperman W. A., Hodgkiss W. S., Song H. C., Edelmann G., Akai T. (2004) Echo-to-reverberation enhancement using a time reversal mirror [J. Acoust. Soc. Am. **115** (4), 1525–1531].

Klyachin B.I, Kur'yanov B.F. (1974) Effect of sound scattering on the depth dependence of low-frequency oceanic noise [DAN SSSR Vol. 260(4) pp. 1009–1012] (in Russian)

Knobles D.P., Mitchell S.K. (1994) Broadband localization by matched fields in range and bearing in shallow water [J. Acoust. Soc. Am. Vol. 95(3), p. 1813–1820]

Knobles D. P., Koch R. A, Miller J. H., and Potty G. R., (2004) "Evidence for nonlinear frequency dependence of attenuation in an East China Sea environment," The Journal of the Acoustical Society of America 115(5), 2551–2551.

Knudsen V.O., Alford R.S. and Emling J.W. (1948) Underwater ambient noise [J. Marine Res., 7, pp. 410–429]

Koch R., Vidmar P., Lindfray J. (1983) Normal mode identification for impedance boundary condition [J. Acoust. Soc. Am., Vol. 73(5), pp. 1567–1570]

Koch, R.A. and Knobles, D.P. (2004) "Geoacoustic inversion with ships as sources". J. Acoust. Soc. Am. 117(2), pp. 626–637.

Kobozev I.K, KravtsovYu.A., Ogurtsov A.V. (1988) Perturbation theory for calculating the characteristics of normal modes in near-bottom waveguides [Sov. Phys. Acoust., vol. 34(1), pp. 58–61]

Kohler W. and G. Papanikolau (1977) Wave propagation in a randomly inhomogeneous ocean In: Wave propagation and underwater acoustics (ed) J.B. Keller and J.S. Papadakis [Springer-Verlag, pp. 153–223]

Konyaev K.V, Sabinin K.D. (1992) Waves within the ocean. St Petersburg Gidrometeoizdat p. 271 (in Russian)

Koniaev K.V., Sabinin K.D. (1973) New data concerning internal waves in the sea obtained using distributed temperature sensors. [Transaction (Doklady) of the USSR Academy of Sciences. Vol. 209(1), pp. 86–89.

Korotin P.I., Salin B.M., Chashchin A.S., Tsiberev A.V. (2000) Marine autonomous system for acoustic measurement. [Proceeding of the U.S. – Russia workshop on experimental underwater acoustics. Nizhny Novgorod, pp. 114–154]

Kozubskaya G. I., Konyaev G. V., Pludeman A., Sabinin K. D. (1999) Internal waves at the slope of Bear island from the data of the Barents sea polar front experiment (BSPF-92) *Oceanology*, **39**(2), 147–154.

Kravtsov Yu.A., Kuz'kin V.M., Petnikov V.G. (1984) Perturbation calculation of the horizontal refraction of sound waves in a shallow sea [Sov. Phys. Acoust., Vol. 30(1), pp. 45–47]

Kravtsov Yu.A., Kuz'kin C.M., Petnikov V.G. (1983) Approximate approach to the problem of wave diffraction in multimode waveguides with smoothly varying parameters [Izv. vuzov. Radiofizika 26(4), p. 440–446 (in Russian)].

Kravtsov Yu.A., Kuz'kin C.M., Petnikov V.G. (1984) Wave diffraction by regular scatterers in multimode waveguides [Sov. Phys. Acoust., Vol. 30(3), pp. 199–202]

Kravtsov Yu.A., Kuz'kin V.M. and Petnikov V.G. (1988) Resolvability of rays and modes in an ideal waveguide [Sov. Phys. Acoust., vol. 34(4), pp. 387–390]

Kravtsov Yu.A., Orlov Yu.I. (1990) Geometrical optics of inhomogeneous media [Springer-Verlag]

Kravtsov Yu.A., Petnikov V.G. (1986) Feasibility of phase ocean tomography by using normal waves [Izvestia, Atmospheric and Ocean Physics, V22. N9, pp. 992–994

Kryazhev F.I., Kudryashov V.M. (1984) Effect of scattering on the boundary on the sound field in a waveguide In: Problems of oceanic acoustics [Moscow Nauka pp. 57–69] (in Russian)

Krupin V.D. (1972) A special effect in a waveguide having a negative sound velocity gradient [Sov. Phys. Acoust., Vol. 18(4), pp. 556–564]

Kvyatovskii S.O. (1988) Diffraction of sound waves by a scatterer in a waveguide [Sov. Phys. Acoust., Vol. 34(4), pp. 430–431]

Kudryashov V.M., (1999) Low-frequency reverberation in shallow water arctic seas. Acoust. Phys. **45**(3), 320–326.

Kuperman W.A., Jensen F.B. (ed) (1980) Bottom-interacting ocean acoustics [Plenum Press, New York and London, 717p.]

Kuperman W.A., Ferla M.C. (1985) A shallow water experiment to determine the source spectrum level of wind generated noise [J. Acoust. Soc. Am.., Vol. 77(6), pp. 2067– 2073]

Kuperman W.A., Ingenito F. (1980) Spatial correlation of surface generated noise in a stratified ocean [J. Acoust. Soc. Am., Vol. 67(6), pp. 1988–1996]

Kuperman W.A., Hodgkiss W.S., Hee Chun Song, Akal T, Feria C. (1998) Phase conjugation in the ocean: Experimental demonstration of an acoustic time-reversal mirror. J. Acoust. Soc. Am. **103** (1), 25–40.

Kuperman and J. Lynch, (2004) "Shallow water acoustics". Physics Today, pp. 55–61.

Kur'yanov B.F. (1963) Spatial correlation of fields emitted by random sources in a plane [Akust. zhurn., vol 9(4),pp. 441–448] (in Russian)

Kur'yanov B.M., Klyachin B.I. (1974) Application of the theory of radiation transport to problems of oceanic noise propagation In: Problems of oceanic acoustics [Moscow, Nauka, p. 16–36] (in Russian)

Kutschale, H.W. (1973) "Rapid computation by wave theory of propagation loss in the Arctic ocean. Rep CU.8-73, Columbia university, Palisades, NY

Kuz'kin V.M., Ogurtsov A.V., Petnikov V.G. (1998) The effect of hydrodynamic variability on frequency shifts of the interference pattern of a sound field in a shallow sea [Acoust. Phys. Vol. 44(1), pp. 77–82]

Kuz'kin V.M. (1996) Horizontal refraction of acoustic waves at internal waves in shallow sea [Acoust. Phys., Vol. 42(3), pp. 390–392]

Kuz'kin V.M. (1997) Characteristics of diffracted acoustic signals in an oceanic waveguide [Acoust, Phys., Vol. 43(4), pp. 440–445]

Kuzkin V.M., Lavrova O.Yu., Pereselkov S.A., Petnikov V.G., Sabinin K.D. (2006) Anisotropic field of the background internal wave at sea shelf. *Acoust. Phys.* 52(1), pp. 65–76.

Kuz'kin V.M. (2001) Radiation and Scattering of Sound Waves in Oceanic Waveguides [Acoust. Phys. Vol. 47(5), pp. 591–597].

Kuznetsova E.P. (1988) Sound field of surface noise sources in a randomly inhomogeneous waveguide [Sov. Phys. Acoust., Vol. 34(5), pp. 511–514]

Kuznetsova E.P. (1989) Calculation of the directivity of noise in a shallow sea [Sov. Phys. Acoust., Vol. 35(6), pp. 627–630]

Konyukhova N.B., Pak T.V. (1985) Diffraction of a plane sound wave on a rigid prolate spheroid [Moscow, Izd-vo VTs AN SSSR. 61p.] (in Russian)

Komarov N.V., Ponomarev L.I., Slavyanov S.Yu. (1976) Spheroidal and Coulomb spheroidal functions [Moscow, Nauka, 319p.] (in Russian)

Laville F., Abbott G.D., Miller M.J. (1991) Underwater sound generation by rainfall [J. Acoust. Soc. Am., Vol. 89(2), pp. 715–721]

Lebedev O.V. Pronchatov-Rubtsov N.V. and Simdyankin S.I. (1996) Calculation of the sound field in a waveguide with layered absorbing bottom [Acoust. Phys., Vol. 42(1), pp. 65–70]

Lee J. B. (1974) Low-frequency resonant-tube projector for underwater sound. [IEEE International conference on engineering in the ocean environment. Halifax, Vol. 2, pp. 10–15]

Leon-Garcia, (1994) "Probability and random processes for electrical engineering", Addison-Wesley, Menlo Park, 596 pages.

Leontovich M.A., and Fock V.A., (1946) "Solution of the problem of propagation of electromagnetic waves along the earth's surface by the method of parabolic equation," J. Exp. Theor. Phys. **16**, pp 557–573

Lermusiaux, P., C.-S. Chiu, G. Gawarkiewicz, P. Abbot, A. Robinson, R. Miller, P. Haley, W. Leslie, S. Majumdar, A. Pang, and F. Lekien (2006), Quantifying uncertainties in ocean predictions, Oceanography, **19**, pp 90–103,.

Leslie C.B. Sorensen N.R. (1961) Integral solution of the shallow water sound field [J. Acoust. Soc. Am. Vol. 33(3), pp. 323–329]

Li F., Lin J., Zhang R. (2003). Low-frequency shallow water reverberation and bottom scattering model. *8-th Western Pacific Acoustics Conference, Melbourne, Australia.*

Li Xiaofeng, Clement-Colon P., Friedman K.S. (2000) Estimating Oceanic Mixed-Layer Depth from Internal Wave Evolution Observed from Radarsat-1 SAR. Johns Hopkins APL Technical Digest, V. 21, N.1, pp 130–135

Lin Y.T., Chen C.F., and Lynch J.F. (2004). "An Equivalent Transform Method for Evaluating the Effect of Water Column Mismatch on Geoacoustic Inversion", *IEEE J. Oceanic Eng'g.* Special Issue on Effects of Environmental Uncertainty, 31(2), pp. 284–298.

Linder, C., Gawarkiewicz G., and Taylor M., (2006). Climatological estimation of environmental uncertainty over the Middle Atlantic Bight shelf and slope, IEEE, J. of Oceanic Engineering, 31, 308–324.

Lindsay C.E., Chapman N.R. (1993) Matched field inversion method for geoacoustic model parameters using adaptive simulated annealing [IEEE Journal of oceanic engineering, Vol. 189(3), pp. 224–231]

Luchinin A. G., Khil'ko A. I. (2005) Low-Frequency Acoustic Tomography of a Shallow Sea by Low-Mode Pulses [Acoust. Phys., Vol. 51(2), pp. 182–194]

Lukin A.N., Migulin A.A., Moiseev S.N., Petnikov V.G., Suslikov O.B. (1993) Model of fluctuations of the envelope of a harmonic signal in a shallow sea [Acoust. Phys., Vol. 39(4), pp. 363–366]

Lunkov A. A., Petnikov V. G. (2010) Effect of random hydrodynamic inhomogeneities on low frequency sound propagation loss in shallow water. Acoustical Physics V. 56, N3, pp. 328–335

Lynch J., (1983). "On the use of focused horizontal arrays as mode separation and source location devices in ocean acoustics. Part I Theory. [J. Acoust. Soc. Am. 74(5), pp. 1406–1417]

Lynch J., (1985). "On the use of focused horizontal arrays as mode separation and source location devices in ocean acoustics. Part II. Theoretical and numerical modeling results. [J. Acoust. Soc. Am. 78(2), pp. 575–586]

Lynch J., Tang D.J. (2008) Overview of Shallow Water 2006, Special issue paper, JASA, V.124, EL63–EL65

Lynch J.F, Rajan S.D., and. Frisk G.V, (1991). "A Comparison of Broadband and Narrowband Modal Inversions for Bottom Geoacoustic Properties at a Site near Corpus Christi, Texas," [J. Acoust. Soc. Am., 89(2), 648–665]

Lynch J.L., Jin G., Pawlowich R., Ray D., Plueddemann A.J., Chlu C., Miller J.H., Bourke R.H., Parsons A.R., Muench R. (1996) Acoustic travel-time perturbations due to shallow water internal waves and tides in the Barents Sea polar front: theory and experiment [J. Acoust. Soc. Am., Vol. 99(2), pp. 803–821]

Lynch J., Frye D., Peal K., Liberatore S., Kery S., Hobart E., Newhal A., Smith S. (June 1992) Real-time tomography mooring [Technical report,Woods Hole Oceanographic Institution (WHOI092- 29)]

Lynch J., Lin Y.T, Duda T.F., and Newhall A.E., (2009), "Acoustic ducting, shadowing, refraction, and dispersion by curved nonlinear internal waves in shallow water", IEEE J. Oceanic Eng'g. accepted with revisions

Lynch J. Colosi J, Gawarkiewicz G., Duda T., Pierce A., Badiey M., Katznelson B., Miller J., Siegmann W.,.Chiu C, and Newhall A., (2004), "Consideration of fine-scale coastal oceanography and 3-D acoustics for the ESME sound exposure model," [IEEE J. Oceanic. Eng'g., 31(1), 33–48.]

Lynch J., Lin Y.T, Duda T.F., and. Newhall A.E, (2010) "Acoustic ducting, reflection, refraction, and dispersion by curved nonlinear internal waves in shallow water", IEEE J. Oceanic Eng'g. 35(1), pp. 12–27

Lysanov Yu. P. (1980) Geoacoustic model of the upper sedimentary layer in a shallow sea. *Dokl. Akad. Nauk. SSSR.* **251**(3), 714–716.

Lyubchenko A.Yu., Petnikov V.G. (1991) Experimental investigations of the vertical distribution of the intensity of low-frequency noise in a shallow sea [Sov. Phys. Acoust., Vol. 37(3), pp. 303–304]

Macpherson I.D., Daintith M.J. (1967) Practical model of shallow water acoustic propagation [J. Acoust. Soc. Am., Vol. 41(4), pp. 850–854]

Makris N. C. (1996) The effect of saturated transmission scintillation on ocean acoustic intensity measurements. J. Acoust. Soc. Am. V. 100, N2, pp. 769–783

Makris N.C, Ratilal, P., Symonds, D.T., Srinivasan, J., Lee, S., and Nero, R.W. (2006) "Fish population and behavior revealed by instantaneous continental shelf imaging." *Science*, February issue, pp. 660–663.

McKinney C. Mc., Anderson C.D. (1964) Measurements of backscattering of sound from the ocean bottom. *J. Acoust. Soc. Am.***36** (1), 158–163.

Makris N. C., Ratital P. (2001) A unified model for reverberation and submerged object scattering in a stratified ocean waveguide [J. Acoust. Soc. Am., Vol. 109(3), pp. 909–941]
Mal'tsev N.E. (1970) Modifications of the method of cross sections [Sov. Phys. Acoust., Vol. (1), pp. 81–85]
Marcuse D. (1972) Light transmission optics [Van Nostrand Reinhold, Princeton, New Jersey]
Matthews J., Bucca P., and Geddes W., (1985) "Preliminary environmental assessment of the Project GEMINI site – Corpus Christi, Texas," Naval Ocean Research and Development Activity, Rep. No. 120 (June).
Mavko G., Mukerji T., Dvorkin J., (2003) The Rock Physics Handbook, Cambridge University Press, 511 p.
Max J. (1981) Méthodes et techniaues de traitement du signal et applications aux mesures, physiques [Masson, in French]
McDaniel S.T. (1978) Calculation of mode conversion rates [J. Acoust. Soc. Am., Vol. 63(3), pp. 739–749]
McDaniel S.T., McCammon D.F. (1987) Mode-coupling and the environmental sensitivity of shallow water propagation loss prediction [J. Acoust. Soc. Am., Vol. 82(1), pp. 217–223]
Medwin H et al. (2005), "Sounds in the Sea: From Ocean Acoustics to Acoustical Oceanography." Cambridge University Press, New York, NY. 643 pages
Mendus V.I., Postnov G.A. (1993) Angular distribution of the intensity of high frequency dynamic noise in the ocean [Acoust. Phys., Vol. 39 (6), pp. 582–586]
MendusV.I., Postnov G.A. (1993) Intensity of sound radiated by an air bubble [Acoust. Phys., Vol. 39(5), pp. 460–463]
Menke W., (1989). "Geophysical data analysis: discrete inverse theory (revised edition)", Academic Press, London, 289 pages
Michelson, E., Dozier, L., Stokes, A., Green, R. (1981) "Bottom Loss Upgrade", SAIC Technical Report, ADA099478, May.
Milder D. M., (1969) "Ray and wave invariants for SOFAR channel propagation," J. Acoust. Soc. Am. **46**, 1259–1263
Miller J.H., Ching-Sang Chiu, Lynch J.F. (1993) Signal processing for coastal acoustic tomography [Proceedings of the International Conference on Theoretical and Computational Acoustics, Mustique]
Miller J., Bartek L., Potty G., et al., (2004) "Sediments in the East China Sea," J. Oceanic Eng. 29(4), 940–951
Miller C.B. (2004), "Biological Oceanography," Blackwell Publishing, Malden, MA. 402
Millero F.J., Xu Li. (1994) Comments on "On equations for the speed of sound in seawater" [J. Acoust. Soc. Am. 93,255–275 (1993)]. [J. Acoust. Soc. Am., Vol. 95(5), pp. 2757–2759]
Mitchell S., Focke K. (1983) The role of seabottom attenuation profile in shallow water acoustic propagation [J. Acoust. Soc. Am., Vol. 73(1), pp. 153–162]
Morozov A.K., Webb D.C. (2003) A sound projector for acoustic tomography and global ocean monitoring. [IEEE J. Oceanic Eng., Vol. 28(2), pp. 174–185].
Morris G.B. (1978) Depth dependence of ambient noise in the northeastern Pacific Ocean [J. Acoust. Soc. Am., Vol. 64(2), pp. 581–590]
Munk W.H. (1979) Ocean acoustic tomography: a scheme for large-scale monitoring [Deep-Sea Res. Vol. 26(2A), pp. 123–161]
Munk, W., and C. Wunsch (1983), Ocean Acoustic Tomography: Rays and Modes, *Rev*. Geophys., 21(4), 777–793.
Munk W.H., Spindel R.C., Baggeroer A., Birdsall T.G. (1994). The Heard Island feasibility test [J. Acoust. Soc. Am., Vol. 96(4), pp. 2330–2342]
Murphy E.L., Wasilieff A., Jensen F.B. (1976) Frequency-dependent influence of the sea bottom on the near-surface sound field in shallow sea [J. Acoust. Soc. Am., Vol. 59(4), pp. 839–845]

Nechaev A.G., Khil'ko A.I. (1987) Reconstruction of oceanic inhomogeneities along and acoustic track by the differential diagnostics method [Institute of Applied Physics of the USSR Academy of Sciences, Preprint no. 178] (in Russian)

Nagl A., Uberall H., Hang A., Zarur G. (1978) Adiabatic mode theory of underwater sound propagation in range-dependent environment [J. Acoust. Soc. Am., Vol. 63(3), pp. 739–749]

Nechaev A.G., Khilko A.I. (1987) Reconstruction of oceanic inhomogeneities along an acoustic track by the differential diagnostics method. [Preprint No. 178, Institute of Applied Physics of the USSR Academy of Sciences (in Russian)]

Newhall A.E., Lynch J.F., Chiu C.S., and Daugherty J.R., (1990). "Improvements in Three Dimensional Raytracing Codes for Underwater Acoustics," [Computational Acoustics, Vol. I, 169–185]

Newhall A, von der Heydt K., et al. (1998). Preliminary acoustic and oceanographic observation from the Winter PRIMER experiment. Technical report, WHOI-98-19

Newhall A., Costello L, Duda T. et al. (2001) Preliminary acoustic and oceanographic observations from the ASIAEX 2001 South China sea experiment. Technical report, WHOI 2001-12

Newhall A, Duda T. et al. (2007) Acoustic and oceanographic observations and configuration information for the WHOI moorings form the sw06 experiment. Technical report. WHOI-2007-04

Nozdrin Yu.V., Plakhin Yu.A. (1983) On the fine thermohaline structure of the waters of the central portion of the Red Sea [Izvestyia, Atmospheric and Oceanic Physics, vol. 19(3), pp. 219–224]

Oguz H.N. (1994) A theoretical study of low-frequency oceanic ambient noise. [J. Acoust. Soc. Am., Vol. 95 (4), pp. 1895–1912]

Ocean acoustics (ed) L.M. Brekhovskikh. (1974) Moscow, Nauka, in Russian

Ocean acoustics. (2002) Proceedings of the 9th L.M. Brekhovskikh's conference. Moscow, 580 p.

Ocean Noise and Marine Mammals (2003)., National Research Council Ocean Studies Board, National Academy Press, Washington D.C. 192 pages

"Oceanography and Acoustics: Prediction and Propagation Models" (Robinson and Lee, Eds.; 1994)

Oppenheim, A. and A. Willsky, "Signals and Systems" (second edition), Prentice Hall, Upper Saddle River, N.J. (1997).

Parker R.L., (1994). "Geophysical inverse theory," Princeton University Press, Princeton, N.J., 386 pages

Parker S.P. (ed) (1980) Ocean and Atmosphere [McGraw-Hill Encyclopedia]

Parsons A.R., Bourke R.H., Muench R.D., et al. (1996) The Barents Sea polar front in Summer [J. of Geophysical res. Vol. 10 (C6), pp. 14,201–14,221]

Parvulescu A. (1995) Matched-signal ("MESS") processing by the ocean. [J. Acoust. Soc. Am. V. 98(2). p.943–960].

Parvelescu A. and Clay C., (1965) "Reproducibility of signal transmissions in the ocean", [Radio Electron. Eng. 29, pp. 223–228]

Pasewark, S. N. Wolf, M. H. Orr, and J. F. Lynch, "Acoustic intensity variability in a shallow water environment," in *Impact of Littoral Environmental Variability on Acoustic Predictions and Sonar Performance*, edited by N. Pace and F. Jensen ~Kluwer Academic, Dordrecht, 2002, pp. 11–18

Pekeris C.L. (1948) Theory of propagation of explosive sound in shallow water [Geol. Soc. Am. Mem., Vol. 27, pp. 1–117]

Peng Z, Zhou J, Dahl P, and Zhang R, (2004). "Sea-bed acoustic parameters from dispersion analysis and transmission loss in the East China Sea," [J. Oceanic Eng. 29(4), 1038–1045]

Penkin S. I. (2001) Development of subsonic radiators and their use in technical systems for ocean acoustic tomography. [Marine technology. Ed. Acad. M.D.Ageev. Iss.4. Vladivostok: Dalnauka, pp. 149–166.] (in Russian)

Perrone A.J. (1969) Infrasonic and low frequency ambient noise measurements on the Grand Banks [J. Acoust. Soc. Am., Vol. 47(3), pp. 762–770]

Petnikov V.G., Katsnelson B.G., Pereselkov S.A, Sabinin K.D. (1997) Numerical modeling of sound field perturbation caused by internal waves in shallow water [Proceed. of the International conference on shallow-water acoustics, Beijing]

Petnikov V.G., Sabinin K.D., Katsnelson B.G., Kuzkin V.M., Lavrova O.Yu., Pereselkov S.A. (2004) Specific features of internal waves and related peculiarities of acoustic fields at the oceanic continental margins. First fina

Phillips O.M. (1977) The dynamics of upper ocean (2nd edn) [Cambridge University Press]

Physics of Sound in the Sea, NDRC Red Book,(1945); reprinted by Peninsula Publishing, Los Altos, CA

Pierce A., (1991) "Acoustics. An introduction to its physical principles and applications". McGrow-Hill, NY, 642 p.

Pierce A.D. (1965) Extension of the method of normal modes to sound propagation in almost-stratified medium [J. Acoust. Soc. Am., Vol. 37, pp. 19–27]

Pierce A.D. (1982) Guided mode disappearance during upslope propagation in variable depth shallow water overlying a fluid bottom [J. Acoust. Soc. Am., Vol. 72(2), pp. 523–531]

Pierce A.D., (1988) "Acoustics: An Introduction to Its Physical Principles and Applications." McGraw Hill, New York, NY. 642 pages,

Price J.F, Weller R.A., Pinkel R (1986). "Observations and Models of the Upper Ocean Response to Diurnal Heating, Cooling, and Wind Mixing." [J. Geophys. Res, Vol. 91, No. C7, pages 8411–8427, July 15

Pierce A.D,. Carey W. M,. Lynch J. F, and Zampolli M., (2004) "Approximate single wave equation derived from Biot's porous media equations," [The Journal of the Acoustical Society of America 115(5), 2400–2400]

Pierson W.J., Moskowitz L.A. (1964) A proposed spectral form for fully developed wind seas based on the similarity theory of S.A.Kitaigorodsky [J. Geophys. Res., v. 69(24), pp. 5181–5190]

Piggott G.L. (1965) Ambient sea noise at low frequencies in shallow water of Scotia Shelf [J. Acoust. Soc. Am., Vol. 36, pp. 2152–2163]

Pond S.and. Pickard G, (1983) "Introductory dynamic oceanography", Pergamon, New York, 329 pages.

Poole, T.L., Frisk, G.V., Lynch, J.F. and Pierce, A.D. (2008) "Geoacoustic inversion by mode amplitude perturbation.", *J. Acoust. Soc. Am.* Volume 123, Issue 2, pp. 667–678.

Porter M., Reiss E.L. (1984) A numerical method for ocean-acoustic normal modes. [J. Acoust. Soc. Am., Vol. 76(1), pp. 244–252]

Porter M., Reiss E.L. (1985) A numeral method for bottom interaction in ocean acoustic normal modes [J. Acoust. Soc. Am., Vol. 77(5), pp. 1760–1767]

Porter M.B. (1992) The KRAKEN normal mode program [Report NRL/MR/5120-92- 6920, Washington, 194 p.]

Porter M.B, Jensen F.B, Ferla C.M. Problem of energy conservation in one-way models. [J. Acoust Soc of America, **89,** 1991, 1058–1067]

Potty G. R,. Miller J. H, and Lynch J. F., (2003) "Inversion for sediment geoacoustic properties at the New England bight," [J. Acoustical Society of America **114**(4), 1874–1887]

Potty G., Miller J., et al. (2004) , "Geoacoustic inversion results from the ASIAEX East China Sea Experiment," [J. Oceanic Eng. 29(4), 1000–1010]

Rajan S. D, (1992) "Waveform Inversion to Obtain the Geoacoustic Properties of Sediments in Deep Water", Journal of the Acoustical Society of America, 91(6), 3228–3241,

Rajan S.D., Lynch J.F., Frisk G.V. (1987) Perturbative inversion methods for obtaining bottom geoacoustic parameters in shallow water [J. Acoust. Soc. Am., Vol. 82(3), pp. 998– 1017]

Rajan S.D and Frisk G. V., (1992) "Seasonal Variations of the Sediment Compressional Wave Speed Profile in the Gulf of Mexico", [Journal of the Acoustical Society of America, **91**(1), 127–135],

Richardson W.J., Greene, C.R., Malme, C.I., and Thomson, D.H., (1995) "Marine Mammals and Noise." Academic Press, New York, NY. 576 p.

Rimsky-Korsakov A. V., Yamshikov V. S., Zhulin V. I., Rehtman V. I. (1984) Subsea acoustic low frequency sources. [Sudostroenie] (in Russian)

Robinson I.S. (1994) Satellite Oceanography [Wiley-Praxis Series in Remote Sensing, Chichester, 455 p]

Rogers P., Zhou J., Zhang X. Z.,. Li F, et al., (2000) "Seabottom acoustic parameters from inversion of Yellow Sea experimental data," in Experimental Acoustic In-JASA version Methods for Exploration of the Shallow Water Environment, edited by A. Caiti et al. (Kluwer, Norwell, MA, pp. 219–234.

Ross D. (1976) Mechanics of underwater noise [Pergamon Press, 375 p]

Rouseff D. (2001) Effect of shallow water internal waves on ocean acoustic striation patterns. Waves in Random Media. V.11. P.377–393.

Rouseff D. and Spindel R.C. (2002) Modeling the waveguide invariant as a distribution. In: Ocean acoustic interference phenomena and signal processing ed. W.A. Kuperman and G L d'Spain. New York: AIP pp. 137–148

Rouseff D. (2001) Effect of shallow water internal waves on ocean acoustic striation patterns, Waves in Random Media. V.11. P.377–393.

Rozenfeld I., Carey W., Cable P., and Siegmann W., (2001) "Modeling and analysis of sound transmission in the Strait of Korea," [J. Oceanic Eng. 26(4), 809–820].

Rubano.L. A., (1980) "Acoustic propagation in shallow water over a low-velocity bottom," The Journal of the Acoustical Society of America 67(5), 1608–1613

Rutherford S.R. (1979) An examination of multipath processes in a range dependent ocean environment within the context of adiabatic mode theory [J. Acoust. Soc. Am., Vol. 66(5), pp. 1482–1486]

Rutenko A.N. (1997) Seasonal variability of intensity and phase fluctuations of low frequency underwater sound signals at the shelf of the Sea of Japan [Acoust. Phys., Vol. 43(1), pp. 84–91]

Rutenko A. N. (1998) The Vertical Hydroacoustic Array "Mollyusk-97" [Instruments and Experimental Techniques, Vol. 41(5), pp. 723–726.]

Rutenko A.N. (2000) Experimental Study of the Effect of Internal Waves on the Frequency Interference Structure of the Sound Field in Shallow Sea [Acoust. Phys., Vol. 46(2), pp. 207–213]

Rutenko A.N. (2003) Effect of Internal Waves on the Intensity and Interference Structure of the Sound Field in a Shelf Zone [Acoust. Phys., Vol. 49(4), pp. 449–454]

Rutenko A.N. (2006) Spectral Features of the Water Temperature and Sound Intensity Variations Measured on the Shelf of the Sea of Japan [Acoust. Phys., Vol. 52(4), pp. 455–461]

Rytov S.M. Kravtsov Yu.A., Tatarskii V.I. (1989) Principles of statistical radiophysics [Springer-Verlag, Vol. 4]

Sarkissian A. (1994) Method of superposition applied to scattering from a target in a bounded medium [J. Acoust. Soc. Am., Vol. 95(5), pp. 2340–2345]

Sarkissian A. (1997) Extraction of a target scattering response from measurements made over long ranges in shallow water [J. Acoust. Soc. Am., Vol. 102(2), pp. 825–832]

Sarkissian A. (1994), Method of superposition applied to scattering from a target in a bounded medium [J. Acoust. Soc. Am., vol. 95(5), pp. 2340–2345]

Schiff L., (1968) "Quantum Mechanics." McGraw-Hill, New York NY. 544 pages.

Schmidt H., Kuperman W.A. (1986) Estimation of surface noise source level from low-frequency seismoacoustic noise measurements [J. Acoust. Soc. Am., Vol. 84(6), pp. 2153–2161]

Schmidt H., (1988) SAFARI: Seismo-Acoustic Fast field Algorithm for Range-Independent Environments, SACLANTCEN REPORT,SR-113.

Schumacher J.D, Kinder, T., Pashinski, T. and Charnell, R.L. (1979) a structural front over the conitinental shelf of the Eastern Bering Sea. [J. Phys. Oceanogr. **9,** pp. 79–87].

Seibold E.and Berger, W.H. (1996) "The Sea Floor: An Introduction to Marine Geology." Springer, New York. 356 p.

Serebryany A. N. (2003) Experience of study of processes in coastal zone of the sea by means of acoustic Doppler current profiler. Annular report of Scientific School of prof. S.A. Rybak, Moscow, pp. 155–165

Shang E.C., Anderson V.C. (1986) Surface generated noise under low-wind speed at kilohertz frequencies [J. Acoust. Soc. Am., Vol. 79(4), pp. 964–971]

Shang E.C. Chang R.H. (1975) Theory on long-range reverberation in shallow water. *Acta Phys. Sinica* 24, 260–267.

Shmelerv A.Yu. (1990) Ph.D. thesis, General Physics Institute of the USSR Academy of Sciences (in Russian)

Shenderov E.L. (1989) Emission and scattering of sound [Leningrad; Sudostroenie p. 304] (in Russian)

Sherman C. H., Butler J. L. (2007) Transducers and Arrays for underwater Sound [Springer, 2007, 610p.]

Shifrin Y.S. (1971) Statistical antenna theory [Golem, Boulder, CO]

Shmelerv A.Yu., Migylin A.A., Petnikov V.G. (1992) Horizontal refraction of low frequency acoustic waves in the Barents Sea stationary acoustic track experiment [J. Acoust. Soc. Am., Vol. 92(2), pp. 10003–1007]

Shmelerv A.Yu. (1990) Dissertation. Moscow General Physics Institute of the USSR Academy of Sciences (in Russian)

Shvachko R.F. (1982) Fluctuations of the sound field on random inhomogeneities on the ocean / Ocean acoustics. Modern state. Moscow, Nauka, pp. 132–141. (in Russian)

Shvachko R.F. (1984) Fluctuations of sound on inhomogeneities in a deep ocean in: Oceanic acoustics. Current status [Moscow, Nauka, p. 132–140] (in Russian)

Sidenko A.V. (1989) Wave propagation in randomly inhomogeneous dissipative waveguides [Dissertation, Voronezh, p. 120] (in Russian)

Siderius, M., Harrison, C.H., and Porter, M. (2006) "A passive fathometer technique for imaging seabed layering using ambient noise." J. Acoust. Soc. Am. 120(3), pp 1315–1323.

Sims C. C. (1960) Bubble transducer for radiating high-power low-frequency sound in water. [J. Acoust. Soc. Am., Vol. 32(10), pp. 1305–1308]

Smith D. T., (1974) In *Physics of Sound in Marine Sediments*, Ed. by L. Hampton Plenum, New York.

Smith G.B., Feullade C., Del Balzo D.R., Byrne C.L. (1989) A nonlinear matched-field processor for detection and localization of quiet source in a noisy shallow-water environment [J. Acoust. Soc. Am., Vol. 85(3), pp. 1158–1166]

Smith P.W. (1980) Averaged sound transmission in range dependent channels [J. Acoust. Soc. Am., Vol. 68(1), pp. 269–281]

Smith P.W. (1986) Low-frequency roll off in the response of shallow water channels [J. Acoust. Soc. Am., Vol. 79(1), pp. 71–75]

Spindel R.C. (1985) Signal processing in ocean tomography. In: Adaptive methods in underwater acoustics. (ed) H.G. Urban [D.Reidel, Boston, pp. 687–710]

Stickler D.C, Ammicht E. (1980) Uniform asymptotic evaluation of the continuous spectrum contribution for the Pekeris model [J. Acoust. Soc. Am., Vol. 67(6), p. 2018– 2024]

Sperry B., Lynch, J., Gawarkiewicz G., Chiu C., and Newhall A., (2003). "Characteristics of acoustic propagation to the eastern vertical line receiver during the summer 1996 New England shelfbreak PRIMER experiment." [J. Oceanic Eng'g.28 (4), pp. 729–749].

Stoll R., (1989) Sediment Acoustics (Springer Verlag, New York, NY,

Stoll R., (1977) "Acoustic waves in ocean sediments," Geophysics 42(4), 715–725

Tang Y.W. (1989) Average intensity of long range surface reverberations in shallow water with positive sound velocity gradient. *Acta Geophys. Sinica* **32**, 667–674.

Tappert F.D. (1977) "The parabolic approximation method," in *Wave Propagation and Underwater Acoustics*, edited by J. B. Keller and J. S. Papadakis, Lecture Notes in Physics 70 Springer-Verlag, New York

Tappert F. D. and Yamamoto T., (1993) An issue (sediment volume fluctuations) and a nonissue (shear wave propagation) in shallow water acoustics, J. Acoust. Soc. Am. 93, 2268.
Tarantola A., (2005). "Inverse problem theory and methods for model parameter estimation, SIAM, Philadelphia, PA, 342 pages
Tatersall J. M., Chizhik D., Cole B. F. and DiNapoli F. R. (1993) "The effect of frequency-dependent bottom reflectivity on transmission loss in shallow water over a sandy bottom," [The Journal of the Acoustical Society of America 93(4), 2395–2395].
Taylor J.R. (1972) Scattering theory. The quantum theory on nonrelativistic collisions [John Wiley & Sons]
Thode A. (2004)., "The derivative of a waveguide acoustic field with respect to an arbitrary sound speed perturbation", J. Acous. Soc. Am. 115(6), 2824–2833
Tindle C.T., Bold G. (1981) Improved ray calculations in shallow water. JASA, V.70, pp. 813–819
Tindle C.T., Guthrie K.M. (1974) Rays and intensity modeling in underwater acoustics. J. Sound and Vibr. V.34, pp. 291–295
Tindle C.T., Hobaeck H., Muir T.G. (1987) Downslope propagation of the normal modes in a shallow water wedge [J. Acoust. Soc. Am., vol. 81(2), pp. 275–286]
Tindle C.T., Dean G.B. (1985) Sound propagation over a sloping bottom using rays with beam displacement [J. Acoust. Soc. Am., Vol. 78(4), pp. 1366–1374]
Tindle C.T., Stamp, A.P, and Guthrie, K.M. (1976), "Virtual modes and the surface boundary condition in underwater acoustics." [Journal of Sound and Vibration, Volume 49, Issue 2, p. 231–240].
Tindle C.T., Weston D.E. (1980) Connection of acoustic beam displacement cycle distance and attenuation for rays and normal modes [J. Acoust. Soc. Am., Vol. 67(5), pp. 1614–1622]
Tindle C.T and Zhang, Z.Y. (1992) "An equivalent fluid approximation for a low shear speed ocean bottom", *J. Acoust. Soc. Am.* Volume 91, Issue 6, pp. 3248–3256
Titchmarsh E.C. (1958) Eigenfunction expansion associated with second-order differential equations [Oxford, Clarendon Press]
Tolstoy A,. Chapman N. R,. Brooke G. H (1998) Benchmarking geoacoustic inversion methods. [J. Comput. Acoust. V.6, pp. 1–28].
Tolstoy A., Diachok O., Frazer L.N., (1991) Acoustic tomography via matched field processing [J. Acoust. Soc. Am., Vol. 89(3), pp. 1119–1127]
Tolstoi I., Clay C.S. (1966) Ocean Acoustics, Theory and Experiment in Underwater Acoustics, McGraw-Hill Book Company, New-York – St. Louis – San Francisco – Toronto – London –Sydney
Tonoyan I.P. (1991) Propagation of acoustic noise in the ocean [Izvestyia, Atmospheric and Oceanic Physics, V. 27(5), pp. 570–577]
Triumph of the Nerds, DVD, (1996)
Trowbridge J.H., Nowell R.M. (Ed) (1994), Sediment transport events in shelves and slopes, STRESS, 1289 p
Turbiner A.V. (1984) The eigenvalue spectrum in quantum mechanics and the nonlinearisation procedure [Sov. Phys. Uspekhi, Vol. 27(9), pp. 668–694]
Turgut, A., Orr, M, and Rouseff, D.(2010) "Broadband source localization using horizontal beam acoustic intensity striations." J. Acoust. Soc. Am. 127(1), pp. 73–84 (2010).
Tuzhilkin Yu.I., Eliseevnin V.A. (1995) Diffraction of a sound field at a plane rectangular vertical screen in a waveguide [Acoust. Phys., Vol. 41(2), pp. 214–218]
Ugincius P. (1969) Intensity equation in ray acoustics [J. Acoust. Soc. Am., Vol. 45(1), pp. 193–209]
Urick R.J. (1975) Principles of underwater sound [New York, McGraw-Hill, 384p]
Urick R.J. (1954) The backscattering of sound from a harbor bottom. *J. Acoust. Soc. Am.* **26** (2), 231–235.
Uzunoglu N.K., Fikioris J.K. (1982) Scattering from an inhomogeneity inside a dielectric-slab waveguide [J. Opt. Soc. Am., Vol. 72(5), pp. 628–637]

Vadov R.A. (1994) Characteristics of underwater explosions sound sources characteristics [Acoust. Phys. Vol. 40(4), pp. 600–602]
Vadov R. A. (2001) Long-Range Sound Propagation in the Central Region of the Baltic Sea [Acoust. Phys., Vol. 47(2), pp. 150–159]
Vadov R. A. (2002) Long-Range Sound Propagation in the Shallow-Water Part of the Sea of Okhotsk [Acoust. Phys., Vol. 48(2), pp. 142–146]
Vadov R. A. (2003) Long-Range Sound Propagation in the Central Part of the Barents Sea [Acoust. Phys., Vol. 49(6), pp. 638–646]
Van Trees H.C. (1968) Detection, estimation and modulation theory. (Wiley, New York, Pt.1)
Vavilin A.V., Kozel'skii A.R., Petnikov V.G., Reznikov V.M., Rivelis E.A. (1987) Characteristics of the dispersion distortions of pulse signals in acoustic waveguides with absorption [Sov. Phys. Acoust., Vol. 33(5), pp. 830–833]
Vianna M.L., Soares-Filho W. (1986) Broadband noise propagation in a Pekeris waveguide [J. Acoustic Soc. Am., Vol. 79(1), pp. 76–83]
Voitovich N.N. and Shatrov A.D., (1972) Normal mode expansion of the field in an underwater sound channel [Sov. Phys. Acoust., Vol. 18(4), pp. 434–438]
Volovov V.I, Ivakin A,N. (1980) Reflection of sound from a bottom with sound speed and density gradients [Akust. zhurn, vol. 26(2), pp. 194–199] (in Russian)
Volovov V.I. Zhitkovskii Yu.Yu. (1977) Scattering and reflection of sound by the ocean bottom (review) Vopr. Sudostr. (in Russian) No. 8
Voronovich A.G., Goncharov V.V. and Chepurin Yu.A. (1991) Measurement of the mode spectrum of a low-frequency acoustic field in the deep sea [Doklady, Earth Science Sections, Vol. 317, pp. 250–254]
Voronovich A.G., Goncharov V.V., Nikol'tsev A.Yu, Chepurin Yu.A. (1992) Comparative analysis of methods for the normal mode decomposition of a sound field in a waveguide: numerical simulation and full-scale experiment [Sov. Phys. Acoust. Vol. 38(4), pp. 365–370]
Webb S.and Cox C., (1986) "Observations and modeling of seafloor microseism," [J. Geophys. Res. 91, 7343–7358]
Webb D.C., Simonetti P.J., Jones C.P. (2001) SLOCUM: An underwater glider propelled by environmental energy. [IEEE J. Oceanic Eng., V.26(4), pp. 447–452]
Wenz G.M. (1962) Acoustic ambient noise in the ocean: spectra and sources [J. Acoust. Soc. Am., vol. 34(12), pp. 1936–1956]
Weston D.E. (1963) Underwater explosions as acoustic sources. [In: V.M. Albers (ed) Underwater Acoustics, Proc. of a NATO-sponsored Institute, 31 July to 11 August 1961. New York: Plenum Press.]
Weston D.E. (1965) Explosive sound sources. In: Underwater Acoustics (translated from the English, (ed) L.M. Brekhovskikh Moscow Mir] (in Russian)
Weston D.E. (1980a) Acoustic flux formulas for range dependent ocean ducts [J. Acoust Soc. Am., vol. 68(1), pp. 269–281]
Weston D.E. (1980b) Wave-theory peaks in range dependent channels of uniform sound velocity [J. Acoust. Soc. Am., Vol. 68(1), pp. 282–286]
Weston D.E. (1980c) Acoustic flux method for oceanic guided waves [J. Acoust. Soc. Am., Vol. 68 (1), pp. 287–296]
Weston D.E. (1980d) Ambient noise depth-dependence models and their relation to low-frequency attenuation [J. Acoust. Soc. Am., Vol. 67(2), pp. 530–537]
Weston D.E. (1976) Propagation in water with uniform sound velocity but variable depth lossy bottom [J. Sound and Vibr. Vol. 47, pp. 473–483]
Weston D.E. (1994) Wave shifts, beam shifts, and their role in modal and adiabatic propagation [J. Acoust. Soc. Am., Vol. 96(1), pp. 406–416]
Weston D. E., Hocking P. D. (1990) Interference pattern in shallow-water reverberation [J. Acoust. Soc. Am., Vol. 87(2), pp. 639–651
Whitman A.M., Beran M.J. (1978) Numerical calculation of the intensity distribution in sound channels using coherence theory [J. Acoust. Soc. Am., vol. 63(6), p. 1727–1732]

Williams A.O., (1970). "Normal mode methods in propagation of underwater sound" in *Underwater Acoustics,* edited by. R.W.B. Stephens (Wiley Interscience, New York), pp. 23–56
Williams A. (1980) Mode interaction in an isovelocity ocean of uniformly varying depth [J. Acoust. Soc. Am., Vol. 67(1), pp. 177–185]
Williams K. L. and Jackson D. R. (1998) Bistatic bottom scattering: Model, experiments and model/data comparison. *J. Acoust. Soc. Am.* **103**(1), 169–181.
Wille P.C., Geyer D. (1984) Measurements of the origin of wind dependent ambient noise variability in shallow water [J. Acoust. Soc. Am., Vol. 75(1), pp. 173–185]
Wilson J.H. (1979) Very low-frequency (VLF) wind generated noise produced by turbulent pressure fluctuation in the atmosphere near the ocean surface [J. Acoust. Soc. Am., Vol. 66, pp. 1499–1507]
Wolf S.N., Ingenito F. (1982) Site dependence of wind-generated ambient noise in shallow water. In: Underwater acoustic noise (ed) R.A.Wagstaff and O.Z.Bluy [rep. CP-32 Saclant ASW Research Center, La Spezia, Italy]
Wood A. and Weston D. (1964) "The propagation of sound in mud," Acustica 14, 156:162
Wunsch C., "The ocean circulation inverse problem", Cambridge University Press, Cambridge, U.K., 442 pages (1996).
Yamamoto T. (1996) Acoustic scattering in the ocean from velocity and density fluctuations in the sediments. *J. Acoust. Soc. Am.* **99**(2), 866–879.
Yang, T.C. (2003) "Temporal resolutions of time-reversal and passive-phase conjugation for underwater acoustic communications". IEEE Journal of Oceanic Engineering, 28 (2), pp. 229–245.
Yang T.C., Kwang Yoo (1997) Modeling the environment influence on the vertical directionality of ambient noise in shallow water [J. Acoust. Soc. Am., Vol. 101(5), pp. 2541–2554]
Yang T.C and Yoo, K., (1999) "Internal wave spectrum in shallow water: measurement and comparison with the Garrett-Munk model. *IEEE J. Oceanic Eng'g.* **24(3),** pp. 333–345.
Young-chung Cho. (1980) Reciprocity principle in duct acoustics *J. Acoust. Soc. Amer.* **67**(5). P.1421–1426.
Zaitsev V.Yu., Nechaev A.G., Ostrovskiy L.A. (1989) Feasibility of mode acoustic tomography [Acoustics of ocean medium ed. by L.M. Brekhovskikh и I.B. Andreeva. M: Naika.. pp. 135–147. (in Russian)]
Zel'dovich B.Y., Pelipetsky N.F., Shkunov V.V. (1985) *Principles of Phase Conjugation.* Springer-Verlag, Berlin,
Zhang Z.Y. Tindle C.T. (1995) Improved equivalent fluid approximation for a low shear speed ocean bottom [J. Acoust. Soc. Am., Vol. 98(6), pp. 3391–3396]
Zhang R.H. and Jin G.L. (1987) Normal-mode theory of the average reverberation intensity in shallow water. *J. Sound Vib.* **119** , 215–223.
Zhuravlev V.A., Kobozev I.K., Kravtsov Yu.A. (1987) Power radiated by an acoustic source in an ocean waveguide [Sov. Phys. Acoust., Vol. 33(6), pp. 611–614]
Zhuravlev V.A., Zarubin V.N., Sochnev O.Ya, (1986) Frequency dependence of the attenuation coefficient of a sound field in a shallow sea [Sov. Phys. Acoust., Vol. 32(1), pp. 70–71]
Zhuravlev V.A., Kobozev I.K, Migulin A.A., Petnikov V.G., Shmelev A.Yu. (1991) Phase fluctuations of low-frequency acoustic waves propagating in an underwater sound channel near the bottom [Sov. Phys. Acoust., Vol. 37(6), pp. 635–636]
Zhuravlev V.A., Kobozev I.K, Kravtsov Yu.A. (1989) Dislocations of a phase front in an ocean waveguide and their manifestation in acoustical measurements [Sov. Phys. Acoust., Vol. 35(2), pp. 156–159]
Zavadskii V. Yu. (1972) Computation of wavefields in open regions and waveguides [Moscow Nauka p. 558] (in Russian)
Zhang R., Zhou J, (ed) (1997) Shallow water acoustics, Proceedings, (21–25 Apr, Beijing, China)
Zhou J., Zhang X. et al., (2004) "Reverberation vertical coherence and sea-bottom geoacoustic inversion in shallow water," J. Oceanic Eng. 29(4), 988–999.

Zhou J. xun, (1985) "Normal mode measurements and remote sensing of sea-bottom sound velocity and attenuation in shallow water," The Journal of the Acoustical Society of America 78(3), 1003–1009

Zhou J., Zhang X., Rogers P.H. (1987) Effect of frequency dependence of sea-bottom attenuation on the optimum frequency for acoustic propagation in shallow water [J. Acoust. Soc., Am., Vol. 82(1), pp. 287–292]

Zhou J., Zhang X., Rogers P.H. (1991) Resonant interaction of soundwave with internal solitons on coastal zone [J. Acoust. Soc. Am., Vol. 90(4), pp. 2042–2054]

Zhou J., Zhang X., Rogers P.H. (1996) Modal characteristics of acoustic signal fluctuations induced by shallow water internal waves [Proceedings of the Int. Conference OCEANS'96 MTS/IEEE, Florida, USA, pp. 1–8]

Zhou J. xun,. Zhang X,. Rogers P. H, and Jarzynski J., (1987) "Geoacoustic parameters in a stratified sea bottom from shallow-water acoustic propagation," [The Journal of the Acoustical Society of America 82(6), 2068–2074]

Zel'dovich B.Y., Pelipetsky N.F., Shkunov V.V. (1985) [*Principles of Phase Conjugation*. Springer-Verlag, Berlin]

Zverev V.A., Orlov E.F. Ed. (1984) [*Interferentsia shirokopolosnogo zvuka v okeane* (Interference on a wide-band sound in the ocean), Gorkii. Inst. Prikl. Fiz., in Russian]

Zverev B.A., Stromkov A.A., Hilko A.I. (2006) Selection of Modes in a Shallow Sea by Using a Vertical Antenna Array [Acoust. Phys., Vol. 52(6), pp. 676–682]

Zverev V.A., Korotin P.I., Stromkov A.A. (2008) Numerical time reversal of waves [Acoust. Phys., Vol. 54(1), pp. 58–64]

Abbreviations

ADC	Analog to digital converter
ADCP	Acoustical Doppler current profiler
ARGOS	Satellite tracking system, named for son of Zeus and Niobe
ASG	Array signal gain
ASW	Antisubmarine warfare
ATOC	Acoustic thermometry of ocean climate program
AUV	Autonomous underwater vehicle
ASIAEX	Asian Seas International Experiment
AVHRR	Advanced Very High Resolution Radiometer
BLUG	Bottom Loss Upgrade program
CODAR	Coastal radar (for surface current measurements)
CTD	Conductivity, temperature, depth – probe
CW	Continuous wave (signal of a single frequency)
EOF	Empirical orthogonal functions
FFT	Fast Fourier transform
FOR3D	A specific 3D parabolic equation model
GEMINI	Shallow water bottom inversion experiment off Corpus Christi, Texas
GM	Garrett–Munk internal wave spectrum
HLA	Horizontal line array
IOOS	International Ocean Observing System
JUSREX	Joint US Russian Experiment
JONSWAP	Joint North Sea Wave Observation Project
KdV	Korteveg de Vries (equation or solution)
KRAKEN	Computer code for calculation of eigenfunctions and eigenvalues of waveguide
LAN	Local area network
MFP	Matched field processing
NDRC	National Defense Research Council
NL	Noise level
OASIS	Ocean Acoustic Services and Information Systems

PDF	Probability density function
PE	Parabolic equation
PPD	Predictive probability distribution
PRIMER	Name of a mid-Atlantic Bight shallow water experiment
ppt	Parts per thousand
psi	Pounds per square inch
REMUS	Remote Environmental Measuring Unit
RMS	Root mean square
SRA	Source/receiver array
SWARM'95	1995 Shallow Water Acoustic Random Medium experiment
SW06	Experiment Shallow Water 2006, taking place in Atlantic shelf of USA
SVD	Singular value decomposition
SVP	Sound velocity profile
SNAP	Acoustic normal mode code
SACLANTCEN	Supreme Allied Commander Europe Atlantic Undersea Research Center
SNR	Signal to noise ratio
SOSUS	Sound surveillance system
SAFARI/OASIS	Ocean Acoustic and Seismic Exploration Synthesis computer code
SAR	Synthetic aperture radar
SL	Source level
SRA	Source/receiver array
TL	Transmission loss
VLA	Vertical line array
XBT	Expendable bathythermogram

Index

A

Absorption coefficient. *See* Attenuation coefficient
Acoustic
  monitoring, 239, 397–398, 413–423
  sources
    airgun, 406, 407
    bubble pulse, 409, 410
    efficiency, 399, 401, 403
    Helmholtz resonator, 403, 404
    monopole, 411
    piezo electric transducer, 399
    radiated power, 403–405, 407, 411
    shock wave, 408–410
  tomography, 109, 178, 272, 334, 398, 417
Acoustic navigation of sensors, 420
Adiabatic approximation, 9, 120–123, 127, 184–186
Ambiguity function, 314, 316–323
Analytic signal, 439–446
Angle
  grazing, 15, 103, 111, 124, 135, 145, 149–156, 158, 174, 175, 183–186, 188–190, 211–213, 218, 240–242, 244, 246, 248, 253, 264, 299, 300, 338, 340, 382, 386, 388–390, 500, 501
  horizontal refraction, 133, 135, 138
  total reflection, 151, 158, 340, 390, 500
Array
  focused, 258, 338, 344, 345
  horizontal, 7, 8, 112, 230, 299, 300, 305, 326, 329, 334, 338–341, 344
  mode filtration, 334–345
  nearfield/farfield, 300, 342
  shading, 334, 335
  signal gain, 324–333
  steered, 75, 304, 338–339, 341–343, 345
  towed, 5, 7, 8, 74, 299, 339, 341, 342, 431
  vertical, 8, 111, 112, 256, 299, 304, 321, 334, 335, 337, 338, 348, 350, 352, 355, 385, 412, 413, 416, 420, 434
Attenuation coefficient
  in bottom, 92, 153, 160, 172, 184, 190, 199, 398
  frequency dependence, 93, 158, 163, 172, 299
  modal, 84, 158, 163
Autonomous underwater vehicles (AUVs), 7, 11–13, 35, 75, 299, 341, 342, 416, 418, 419, 428, 429, 435–437
AUVs. *See* Autonomous underwater vehicles (AUVs)
Average acoustic field
  decay law, 209–212, 214, 216, 397
  description, 104, 181–191, 196–198, 244, 360

B

Bartlett beamformer (processor), 316–317
Beam displacement, 9, 107, 109, 145, 156
Biot–Stoll theory
  attenuation dispersion, 68
  sound speed dispersion, 68
Bohr quantization condition, 101
Brillouin angle/rays, 104, 124, 149, 340, 385
Brunt–Vaisala (BV) frequency. *See* Buoyancy frequency
Buoyancy frequency, 38, 48, 98, 194, 457, 458, 460, 491

C
Cauchy principal value, 91, 440
Coefficient
  Fresnel, 500
  mode-coupling, 114, 116, 117, 126
  normalization, 69, 92, 102, 213, 394
  reflection, 66, 67, 82, 107, 109, 149, 157–159, 188, 300, 302, 304, 386, 499, 501, 502
Coherence, 7, 8, 44, 45, 216, 221, 305, 324–333, 363, 364
Complex
  field, 372
  radiated signal, 469
Condition
  boundary, 2, 6, 21, 32, 81–83, 94, 101, 102, 114, 115, 118, 119, 123, 138, 142, 146, 148, 149, 157, 161, 199, 200, 211, 220, 274, 366, 367, 388, 391, 448, 449, 457, 459, 460, 462, 466, 468, 473, 475–477, 479–483, 485
  normalization, 34, 82, 95, 148, 274, 458, 480
Continental shelf
  active margins, 52
  broad shelf, 52
  narrow shelf, 52
  passive margins, 52
Correlation function
  bottom roughness, 63–65, 195
  cross, 443
  thermohaline structure, 194
  vertical, 206, 374, 387

D
Dipole
  moment, 468
  source, 373, 379, 388, 390, 467
Dispersion
  intramode, 171, 180, 245
  modal, 4, 269
Distributions. *See* Statistics

E
Eddy diffusion coefficient, 21, 23
Eigenfunction, 49, 81–85, 87, 89, 90, 94–96, 100, 102, 115, 118, 146–148, 172, 182, 200, 202, 205, 220, 260, 273, 293, 334, 335, 345, 346, 349, 379, 393, 412, 413, 420, 458, 479–482, 485, 496, 498
Eigenvalue, 35, 63, 81, 83, 87, 90, 94–102, 109, 134, 135, 142, 145, 147–149, 155, 163, 166, 182, 200, 203, 205, 210, 213, 220, 268, 269, 271–274, 280–282, 284, 293, 321, 335, 340, 350, 373, 379, 389, 394, 420, 426, 457, 479, 488
Empirical orthogonal functions (EOF's), 60–62, 268, 487–492
Equation of state for seawater
  adiabatic compressibility, 19
  thermal expansion coefficient, 37
Equations
  Airy functions, 95, 101, 102
  average modes amplitude, 203
  coupled intensities, 198
  diffusion, 21, 209–212, 216, 217, 387
  eikonal for horizontal rays, 130
  Fresnel, 151
  Helmholtz, 104, 140
  Riccati, 217
  Schrodinger, 213
  Sturm-Liouville, 113
  transfer, 206, 209
  transport for horizontal rays, 124–144
  wave, 2, 38, 39, 42, 81, 84, 95, 104, 138, 139, 141, 267, 270, 369, 426, 443–445, 455, 466, 468, 473, 484
Ergodic hypothesis, 358
Experiments
  ASIAEX experiment, 7, 8, 59–61, 153, 385, 428
  Barents Sea Polar Front experiment, 7
  GEMINI experiment, 7, 278–284
  PRIMER experiment, 7, 26, 28, 29, 233–235, 383, 384, 428
  SWARM experiment, 7, 98, 142, 144, 153, 176, 232, 345, 416, 489, 490
  Yellow Sea experiment, 6, 172

F
Fast Fourier transform (FFT), 4, 5
FFT. *See* Fast Fourier transform (FFT)
Field calculation models
  fast field programs, 5, 280
  Hankel transform, 5, 9
  parabolic equation, 5, 9, 142
  Weinberg-Burridge ray/mode, 426, 430
Fish schools, 13
Frequency
  angular, 79, 269
  buoyancy, 38, 48, 98, 194, 457, 458, 460, 491

cut-off, 31, 89, 94
shift, 296
Function
Bessel, 63, 373, 389, 474, 476
Green, 9, 80, 81, 84, 87, 90, 102, 105, 116, 125, 138, 140, 176, 280, 350, 366, 372, 373, 375, 387, 444, 467, 469, 473–480, 494
Hankel, 84, 105, 373, 476
Fuzzy rays, 106, 107

G
Gaussian beam, 107, 140, 141
Green function
of medium, 469
modal decomposition, 87, 373, 473–480

H
Hankel transform, 5, 9, 58, 272, 343, 372, 474
Hidden depths, 62
Hilbert transform, 440
Hooke's law, 65
Horizontal refraction, 3, 124–138, 144, 355

I
Impedance, 55, 82, 83, 91, 300, 475
Index of refraction, 127, 130, 131, 135, 142, 426
Intensity
average, 183, 186, 191, 197, 214, 219, 230, 247, 250–252, 254, 265, 387, 392
coherent component, 204, 207, 212, 215, 219
incoherent component, 204, 207
interference between modes, 181, 426
local, 182
noise, 325, 373, 376, 390–395
soliton-like,
of sound field, 143, 215, 221, 286, 431
spatial scale, 257
Internal waves–linear and nonlinear
buoyancy frequency, 38, 48, 194
Garrett–Munk internal wave spectrum, 228
inertial waves, 38, 47
Korteweg-deVries (KdV) equation, 38, 42
soliton trains, 45, 47, 144, 428
SUNTANS model, 425
Taylor–Goldstein equation, 39
Inverse techniques
average medium properties, 267, 268
Bayesian, 267, 268, 283, 284, 360, 433
cost function, 283
equivalent transform methods, 275–277
"forward problem", 267
genetic algorithem, 268, 283, 433
"inverse problem", 100, 267–304, 423, 425, 431–433
kernel, 268, 269, 272, 278, 317
Monte Carlo methods, 267, 283
Moore Penrose pseudoinverse, 278
a priori information, 270, 271
proximity coefficient, 286
simulated annealing, 268, 283, 433
singular value decomposition, 277, 278

J
Jacobian (ray divergence), 128, 136

L
Lloyd's mirror, 299, 426, 427
Logarithmic velocity profiles, 22

M
Marine mammals
right whale, 76
Matched field processing (MFP), 300, 305, 313–324, 352, 434, 436, 508
Maximum likelihood method (MLM), 319–324
Mid-frequency acoustics, 12
Mirror
phase conjugate, 255
time reversal, 255, 305, 348–357, 434, 435
Modal method
cycle distance, 145, 231
EJP branch cut, 90
equipartition, 231
evanescent modes, 35
generalized, 279
intensity, 85, 208, 387
leaky modes, 83
selection, 335, 413
transport equation, 103, 127, 128, 198, 199
trapped, 232
WKB, 94–102, 146, 148, 149, 166, 183, 391, 498
Mode coupling
coefficient, 114, 116, 117, 120–122, 124, 126, 178, 198, 199, 203, 222
diffusion approximation, 198
equations, 5, 9, 113, 114, 116, 118, 125–127, 142, 203, 262, 481, 483–486
nearest neighbor, 492

"Mode forming"(modal beamforming), 228, 245, 246, 248, 253–255, 337
Moore's law, 10, 11

N
Near-bottom sound channel, 181, 187–191, 224, 259, 391
Noise
  ambient (natural), 3, 31, 75–77, 299–304, 364, 436
  anthropogenic, 1, 76, 299
  Cron-Sherman model, 381
  intensity, angular distribution, 325, 364, 373, 376, 382, 385–388, 390–395
  notch effect, 340, 382–386, 432
  spectral intensity, 371, 377–380
Normalization coefficient
  Gaussian, 190, 214, 217
Nyquist frequency, 307, 337

O
Ocean regions studied (shallow)
  Barents Sea, 7, 8, 53, 64, 65, 75, 130, 153, 155, 172, 176, 178, 196, 223, 252, 260, 264, 285, 286, 291, 294, 297, 298, 334, 392
  East China Sea, 59–61
  Gulf of Mexico, 527
  Mid-Atlantic Bight, 28, 29, 225
  South China Sea, 8, 153, 385
Ocean structures/processes
  buoyant plume front, 24, 26
  Coriolis force, 24, 27
  eddies, 13, 29, 35
  Ekman transport, 24
  filaments/streamers, 13, 29
  fine structure, 14, 62
  foot of the front, 29
  fronts, 13, 29, 35
  internal waves, 13, 35, 36, 432
  Kelvin wave, 32
  Rossby radius, 492
  storm surge, 23, 32
  surface mixed layer, 28, 435
  surface (gravity) wave, 32
  thermocline, 22, 23, 25, 38, 43, 45, 98, 228, 333, 337, 350, 355, 461, 465
  tidal mixing front, 24
  upwelling, 6, 24, 25
Oceanographic instrumentation
  ADCP, 417, 421
  CTD, 28, 416, 417, 419, 435
  current velocity meters, 421
  drifting, 392, 416
  gliders, 7, 11–13, 416, 417, 419, 421, 435, 436
  REMUS AUV, 419
  thermistor string/chain, 417
Optimum frequency, 161–164, 178

P
Parabolic equation (PE), 5, 8, 9, 93, 138–144, 270, 430
Pekeris model, 2, 4, 86–94, 98, 99, 109, 110, 146, 148, 151, 153, 156, 163, 167, 170, 174, 175, 178, 180, 182, 190, 212, 216, 274, 302, 350, 376, 389, 478
Perturbation theory, 94–102, 131, 135, 136, 147, 268, 269, 271, 273, 282
Phase function, 127, 502
Plane wave reflection coefficient, 66, 304
Porosity, 54, 291, 292, 295–298
Power radiated by source, 250, 254, 256, 260, 403–405, 407, 411, 470
Pulse compression, 351

Q
Quadrature components, 289, 308, 311, 312

R
Random inhomogenieties, 179, 193–237, 260, 285, 332, 333, 355, 381–390
Ray
  cycle distance, 106, 122, 145, 149–152, 156, 175, 181, 184–186, 187, 213, 216
  displacement, 9, 107–109, 131, 145, 146, 150, 156, 174
  energy-carrying, 155
  horizontal, 5, 8, 9, 127–137, 142–144, 344, 426, 430
Ray-mode picture connection, 9
Ray-mode resolvability, 110, 111
Reduced gravity, 25
Relief
  bottom, 53, 241, 243, 264, 292, 422
  general, 195
  meso, 195
Reverberation
  backscattering coefficient, 239–242, 246, 248, 255, 259, 264, 265

Lambert's law, 242, 244, 251, 252, 262, 264
Lommel-Seeliger law, 242–244, 252, 264

S
Satellite oceanography
altimeter, 5
ARGOS, 418
AVHRR, 5, 30
SAR, 5
visual, 5, 6
Scattering
amplitude, 249, 493–495, 497
geometric, 72, 390
matrix, 246, 261, 497, 498, 505
Rayleigh, 72, 73, 193, 233, 262, 507
saturated, 230, 233, 431
volume, 69, 293, 296, 297, 382
weak, 230, 232, 233, 431
Scintillation index, 198, 230, 232
Seabed
bathymetry, 2, 3, 13, 14, 27, 29, 34, 53, 54, 63, 362, 426, 431, 432, 492
geoacoustic model, 15, 54, 56, 59
liquid equivalent model, 160
microtopography, 53
poro-elastic media, 67, 429
roughness, 267, 429
shear wave, 65, 66, 69, 82, 157–160, 163, 386, 428, 475
stratigraphy, 54–56
Sediment
consolidated, 57, 62
semi-consolidated, 57
unconsolidated, 57, 285
Shallow water
definition, 1, 52
Shannon theorem, 307
Shelf-break, 1, 3, 6, 7, 13, 14, 23, 24, 26–29, 35, 52, 53, 233, 360–362, 383, 430
Signal
complex, 176, 350, 365, 470
radiated, 176, 254, 260, 287, 288, 305, 345, 354, 355, 405, 466–471
real, 9, 31, 69, 71, 223, 313, 314, 316, 334, 338, 341, 342, 351, 360, 362, 415, 439–441, 443, 451, 457, 465, 466, 469, 470, 505
spectrum, 180, 294, 307, 345, 346, 348, 350–352, 410, 440, 441
Slowness
group, 167, 282
phase, 203
Sound field
amplitude, 142, 168, 197, 223, 225, 226, 227, 295
incident, 496
scattered, 493, 498
Sound speed profile
summer, 18, 181, 386
winter, 18, 162, 386
Source
dipole, 373, 379, 388, 390, 467
distributed, 363, 366, 387, 478
explosive, 240, 406, 408, 409
spectrum level, 365, 366
wideband, 181
Spectrum (acoustic)
continuous, 83, 89–93, 374–377, 478
discrete, 89, 90, 92, 373–377, 387, 478
of radiated signal, 345
Stationary acoustic track, 222, 225, 226, 284
Statistics
Chi-square, 233, 312
Gaussian (normal), 29, 39, 50, 63, 107, 140, 141, 190, 214, 217, 218, 230, 233, 278, 319, 332, 492
log-normal, 233, 235, 236
Rayleigh, 233, 262, 312
student, 311, 312
Sturm-Liouville equation, 113, 479
Surface wave buoys, 415, 435
Surface wave spectrum
Neumann–Pierson, 33
Pierson–Moscowitz, 33, 34, 196, 358

T
Thermocline
thermohaline structure, 194
transmission loss, 14, 15, 57, 70, 145–157, 163, 164, 168, 186, 190, 191, 231, 236, 237, 249, 282, 303, 360, 361, 397, 398
Three-over-two law, 190, 397, 398
Tides
baroclinic, 34–36, 425
barotropic, 34–36, 425
critical latitudes, 35
diurnal, 35
internal, 10, 14, 35, 36, 38, 49, 359, 425
semi deirual (M2), 35
Time-bandwidth product, 110, 231, 236
Transmission loss (TL), 14, 15, 57, 70, 145–157, 163, 164, 168, 186, 187, 190, 191, 231, 236, 237, 249, 282, 303, 360–362, 397, 398, 509

U
Uncertainty, 12, 13, 358–362, 436

V
Variable coefficient likelihood method, 320
Velocity
  group, 41, 100, 110, 128, 171, 172, 174–176, 179, 282, 283, 445, 450
  phase, 45, 174, 175, 445, 450, 453–455, 463
  volume, 467

W
Waveguide
  in horizontal plane, 85, 98, 124–126, 129, 131, 133–135, 138, 142, 144, 183, 195, 196, 340, 366
  invariant, 165, 167, 168, 170, 171, 183, 188, 217, 222, 227, 228, 229, 302, 432
  irregular, 84, 85, 198, 216–219
Wavenumber integration methods
  fast field program, 5
  hankel transform, 5, 9
  SAFARI/OASIS, 5
Wavevector, 39, 80, 83, 90, 95–98, 127, 128, 139, 149, 158, 195, 211, 334, 338, 372, 445, 449–451, 464, 469, 493, 498–500, 503, 504, 506, 507
Wiener–Khinchin theorem, 442
WKB solutions, 105, 120
Wronskian, 87, 90, 476, 477, 479, 506

ACOUSTICS *and* AUDIO TECHNOLOGY

Third Edition



Mendel Kleiner

ACOUSTICS *and* AUDIO TECHNOLOGY

Third Edition

Mendel Kleiner

J.ROSS PUBLISHING

Copyright © 2012 by J. Ross Publishing

ISBN: 978-1-60427-052-5

Printed and bound in the U.S.A. Printed on acid-free paper.

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Kleiner, Mendel, 1946-
Acoustics and audio technology / by Mendel Kleiner. -- 3rd ed.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-60427-052-5 (pbk. : alk. paper)
1. Architectural acoustics. 2. Sound--Recording and reproducing--Equipment and supplies. 3. Hearing. I. Title.
NA2800.K559 2011
620.2--dc22

2011002859

This publication contains information obtained from authentic and highly regarded sources. Reprinted material is used with permission, and sources are indicated. Reasonable effort has been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

All rights reserved. Neither this publication nor any part thereof may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher.

Cover photo courtesy of Ferguson Hill Studios, Ltd.

The copyright owner's consent does not extend to copying for general distribution for promotion, for creating new works, or for resale. Specific permission must be obtained from J. Ross Publishing for such purposes.

Direct all inquiries to J. Ross Publishing, Inc., 5765 N. Andrews Way, Fort Lauderdale, FL 33309.

Phone: (954) 727-9333
Fax: (561) 892-0700
Web: www.jrosspub.com

CONTENTS

Preface ix
Introduction xiii
About the Author xvii
List of Symbols xix
Web Added Value™ xxii

Chapter 1 Sound 1
1.1 Introduction 1
1.2 The Wave Equation 1
1.3 Solutions to the Wave Equation 5
1.4 Impedance 8
1.5 Sound Intensity 10
1.6 Sound Power 11
1.7 Propagation Losses 12
1.8 Reflection and Transmission at Boundaries 13
1.9 Acoustical Components and Circuits 18
1.10 Sound Propagation in Inhomogeneous Media 23
1.11 Dipoles and Quadrupoles 26
1.12 Problems 28

Chapter 2 Audio Signals 31
2.1 Spectrum and Time History 31
2.2 Signals and the $j\omega$-Method 31
2.3 Sine-Wave Signals 34
2.4 Noise 34
2.5 The Level Concept 35

2.6 Filters and Frequency Bands....36
2.7 Effective Value and Measurement Uncertainty....40
2.8 Reference Levels....40
2.9 Addition of Level Contributions....42
2.10 Weighted Sound Pressure Levels....43
2.11 Equivalent Level....43
2.12 Problems....44

Chapter 3 Hearing and Voice....47
3.1 Introduction....47
3.2 The Components of the Ear....47
3.3 Psychoacoustic Dimensions....54
3.4 Effects of Excessive Sound Exposure....58
3.5 Masking and Critical Bands....64
3.6 Distortion, Linearity, and Harmonics....66
3.7 Binaural Hearing....67
3.8 Voice and Speech....70
3.9 Problems....79

Chapter 4 Basic Room Acoustics....81
4.1 Introduction....81
4.2 Geometrical Room Acoustics....82
4.3 Statistical Room Acoustics....87
4.4 Wave Theoretical Approach....99
4.5 Problems....107

Chapter 5 Spatial Sound Perception....111
5.1 Introduction....111
5.2 Subjective Impression of Sound Field Components....112
5.3 Metrics for Room Acoustics....128
5.4 Problems....132

Chapter 6 Room Acoustics Planning and Design....135
6.1 Introduction....135
6.2 Basic Requirements for Good Room Acoustics....136
6.3 Fundamentals of Room Acoustic Planning....140
6.4 Tools for Prediction of Room Acoustic Response....153
6.5 Electronic Architecture....158
6.6 Auralization....159
6.7 Problems....162

Chapter 7 Absorbers, Reflectors, and Diffusers....165
7.1 Introduction....165
7.2 Absorption Coefficient....165

7.3 Porous Absorbers....168
7.4 Resonance Absorbers....177
7.5 Variable Absorbers....182
7.6 Audience Absorption....182
7.7 Reflectors....184
7.8 Barriers....186
7.9 Diffusers....188
7.10 Absorption Data Examples....190
7.11 Problems....194

Chapter 8 Waves in Solids....197
8.1 Introduction....197
8.2 Wave Types in Infinite Media....197
8.3 Wave Types in Media of Limited Extension....200
8.4 Structure-Borne Sound in Lossy Sheets....208
8.5 Damping by Viscoelastic Layers....209
8.6 Damping by Sand and Other Lossy Materials....213
8.7 Problems....215

Chapter 9 Sound Radiation and Generation....217
9.1 Introduction....217
9.2 Common Metrics for Sound Radiation....217
9.3 Sound Radiation by Vibrating Surfaces....220
9.4 Sound Radiation by Vibrating Pistons and Membranes....220
9.5 Sound Radiation by Vibrating Sheets Carrying Bending Waves....221
9.6 Sound Generation by Flow....226
9.7 Problems....229

Chapter 10 Sound Isolation....231
10.1 Introduction....231
10.2 Insulation against Airborne Sound....231
10.3 Sound Transmission Loss....232
10.4 Sound Reduction Index....234
10.5 Transmission Loss of Single-Panel Partitions....234
10.6 Transmission Loss of Multiple-Panel Partitions....238
10.7 Composite Transmission Loss....244
10.8 Cracks and Other Leaks....244
10.9 Flanking Transmission....245
10.10 Balanced Spectrum Design....247
10.11 Insulation against Impact Sound....248
10.12 Insulation against Structure-Borne Sound....252
10.13 Problems....253

Chapter 11 Vibration Isolation 257
11.1 Introduction 257
11.2 Classical Vibration Isolation Theory 257
11.3 Impedance and Mobility 259
11.4 Some Metrics for Vibration Isolation 260
11.5 Linear Single-Degree-of-Freedom Systems 260
11.6 Vibration Isolation Theory Using Electromechanical Analogies 263
11.7 Real Systems 266
11.8 Problems 268

Chapter 12 Microphones 269
12.1 Introduction 269
12.2 Dynamic Range, Frequency Response, Noise, and Distortion 269
12.3 Sensitivity 271
12.4 Electroacoustical Conversion Principles 271
12.5 Sound Field Sensing Principles 273
12.6 Directivity 273
12.7 Diaphragm Mechanical Properties 277
12.8 Resistance Microphones 278
12.9 Piezoelectric Microphones 279
12.10 Capacitive Microphones 280
12.11 Electrodynamic Microphones 283
12.12 Super-Directional Microphones 286
12.13 Wind Noise 289
12.14 Problems 290

Chapter 13 Phonograph Systems 293
13.1 Introduction 293
13.2 Disc Cutting 293
13.3 The Playback System 296
13.4 Cartridges 299
13.5 Amplification 301

Chapter 14 Loudspeakers 303
14.1 Introduction 303
14.2 Radiation and Directivity 304
14.3 Efficiency 308
14.4 Frequency Response 309
14.5 Electrodynamic Drivers 310
14.6 Loudspeaker Enclosures 319
14.7 Horn Loudspeakers 324
14.8 Multiple Direct Radiator Loudspeaker Systems 325

14.9 Array Loudspeakers 326
14.10 Room Effects 328
14.11 Transient Response 329
14.12 Nonlinear Distortion 330
14.13 Electronic Compensation of Nonlinearities 331
14.14 Electrostatic Loudspeakers 332
14.15 Problems 335

Chapter 15 Headphones and Earphones 339
15.1 Introduction 339
15.2 Headphones/Earphones vs. Loudspeakers 339
15.3 The Acoustic Environment 341
15.4 Electromagnetic Headphones 342
15.5 Electrodynamic Headphones 346
15.6 Piezoelectric Headphones 348
15.7 Electrostatic Headphones 348
15.8 Noise-Canceling Headphones 349
15.9 Problems 350

Chapter 16 Digital Representation of Sound 353
16.1 Introduction 353
16.2 Sampling and Digitization 353
16.3 Quantization 357
16.4 Additional Problems in A/D and D/A Conversion 365
16.5 Codecs, Uniform, and Nonuniform Quantization 366
16.6 Lossy Compression and Perceptual Coding 370

Chapter 17 Audio Systems and Measurement 377
17.1 Introduction 377
17.2 Audio Channels and Listening Modes 378
17.3 Monophonic and Monaural 378
17.4 Stereophony and Phantom Sources 378
17.5 Stereo Using Headphones 382
17.6 Binaural Sound Reproduction 384
17.7 Loudspeaker-Headphone Compatibility 385
17.8 Multichannel Sound Reproduction 387
17.9 Metrics For Audio Characteristics 389
17.10 Physical Audio Metrics 390
17.11 Noise Level and Signal-To-Noise Ratio 391
17.12 Amplitude Frequency Response 391
17.13 Phase Response and Group Delay 393
17.14 Nonlinear Distortions 393

17.15 Harmonic Distortion....396
17.16 Difference Frequency and Intermodulation Distortion....397
17.17 Multitone Distortion Measurement....398
17.18 FM Distortion....398
17.19 Measurements Using Special Chambers....399
17.20 Impulse Response Analysis....402
17.21 Frequency Response, Spectrogram, and Wavelet Analysis....403
17.22 Directivity....405
17.23 Sensitivity....406
17.24 Compression....407
17.25 Impedance....407
17.26 Audio Sound Character and Quality....408
17.27 Listening Tests....409
17.28 Common Listening Test Types....410
17.29 Some Common Listening Tests....410
17.30 Selecting and Training of Listeners....412
17.31 Expert Panels....413
17.32 Planning and Preparation of a Listening Test....413
17.33 The Test Session....414
17.34 Information to be Reported....414
17.35 Problems....415

References....419

Answers to Problems....431

Index....441

PREFACE

The purpose of this book, *Acoustics and Audio Engineering*, is to help you learn:

- The physical background to and mathematical treatment of sound propagation. These topics are treated in chapters on the fundamentals of acoustics and theoretical room acoustics.
- The properties of human hearing. This topic is discussed in chapters on hearing and speech and on the spatial aspects of room acoustics.
- The generation and radiation of sound as well as noise control. These topics are treated in chapters on sound and vibration isolation.
- The technologies used for pickup, recording, and reproduction of sound in various environments. These topics are treated in chapters on microphones, loudspeakers, and headphones.

In Chapter 1, the theory of sound propagation in fluids such as gases is discussed for the linear case. You will study the wave equation and its solutions as well as the concepts of sound pressure, particle velocity, and impedance. The physical background to *acoustical components* is discussed and the concept of acoustical circuits. The chapter also deals with sound propagation in layered media that is of interest in the cases of sound propagation outdoors and in water.

Chapter 2 discusses the $j\omega$-method for solving acoustic problems, various measures of sound, filter methods used to separate spectral components, and the concept of level.

Human hearing is one of the most important reasons to study audio and acoustics. Chapter 3 discusses the hearing mechanism and the properties of human hearing such as its sensitivity, bandwidth, and so on. The chapter also includes material on the effect of noise on hearing, hearing damage, and hearing related measures of sound. The concept of binaural hearing is introduced; binaural hearing is of great importance in everyday situations. Finally, the chapter is concluded by a section on the properties of voice and speech. Many communication systems are based on transmission of speech, and it is, therefore, important to be familiar with its properties.

Room acoustics is the subject of interest in Chapter 4. Alternative approaches to the study of room acoustics are introduced: geometrical acoustics, energy balance or statistical acoustics, and physical acoustics. Geometrical acoustics is of great importance for the practical treatment of time related properties in the propagation of transient signals. In practice, geometrical acoustics is being used in the forms of ray tracing and image source modeling. Statistical acoustics is also a *high-frequency* approach to solving room acoustics problems but looks at the conditions in the room for the steady state case and assumes that the signal emitted into the room has sufficient frequency bandwidth and that the room is sufficiently reverberant. In contrast to geometrical acoustics, it cannot be used for analysis of sound propagation outdoors. Physical acoustics uses the approach of finding a solution to the wave equation for the boundary conditions of the room for some sound source and reveals some of the shortcomings of the previously mentioned methods. The practical use of physical acoustics is limited except for understanding the basic principles of sound propagation and resonance unless one uses numerical methods to solve the equations. Such methods may be applied in both the time and frequency domains.

The way that human hearing analyzes the spatial properties of sound fields is the topic of Chapter 5. To be able to work effectively with room acoustical planning, it is necessary to understand the behavior of human hearing regarding how hearing analyzes the complex sound fields that are set up by direct sound and the many reflections off the room boundaries and how we perceive these sound fields. The chapter also discusses some useful measures for describing room acoustical quality.

The application of room acoustic planning is discussed in Chapter 6 and is focused on ways to make sure that a design results in good acoustical conditions in rooms for music performance, theaters, auditoria, and studios. Good room acoustical conditions include low noise levels so that listening and recording can be done without disturbing noise. Some noise criteria are discussed as well as the influence of noise on speech intelligibility.

Chapter 7 includes material on sound absorbers, diffusers, and reflectors that are tools used on the plain room surfaces to achieve the desired room acoustical conditions. Sound absorption is discussed with regard to sound-absorptive materials and discrete sound absorbers, such as membrane and other resonance absorbers. The subject of sound barriers is discussed as well since sound barriers are often used in offices along with sound-absorptive treatment. Sound barriers are of great importance in reducing outdoor and indoor noise levels in urban areas.

Wave propagation in solids is characterized by the possibility of having vibration propagation by many wave types besides longitudinal waves. The reason for this is the presence of shear in solids. Most of the discussion in Chapter 8 is devoted to one particular wave type—bending waves. Bending waves in plates are of great practical importance because they couple easily to waves in the surrounding air. Consequently, the attenuation of bending waves by various techniques is of great importance, such as in the construction of loudspeaker units, loudspeaker boxes, household appliances, cars, and many other devices.

The radiation of waves from vibrating structures is the subject of Chapter 9. Sound generation by turbulence is also included in the material in this chapter because its sound is generated by fast air flows, for example, wind around vehicles, microphones, and in loudspeaker ports.

To enjoy musical performance and high fidelity sound reproduction, it is necessary to reduce background noise levels so that the noise is almost inaudible. Since noise affects human ability to sleep and

concentrate, it is necessary to have low noise levels in work places and homes, for example. Chapter 10 deals with the subject of sound insulation. The properties of various sound insulating constructions are discussed, as well as the influence of leaks and flanking transmission. The impact noise properties of floors are also discussed.

The concept of vibration isolation is discussed in Chapter 11. Classical one-dimensional vibration isolation theory is introduced as is the concept of insertion loss to allow calculation of the vibrational properties of simple dynamical systems. Mechanical mobility and impedance considerations allow a more wide frequency range treatment of the subject and are also treated.

Audio technology starts being treated in Chapter 12. This chapter is devoted to an overview of various microphone types, their construction, electromechanical and acoustical properties. Interest is primarily focused on electrodynamic and electrostatic microphones since these are the most common types for professional recording of speech and music. There are various ways microphones can be designed for desired directional properties. Since microphones are so important in the measurement of sound, this is a very important chapter.

Gramophone cartridges are covered in Chapter 13. Different types of pickups and their dynamic properties are treated as are the fundamentals of gramophone recording technology.

Chapter 14 is devoted to the subject of loudspeakers. Most loudspeakers are composed of a driver unit and a loudspeaker box. The loudspeaker box can be designed in many ways and has a large influence on the characteristics of the complete loudspeaker. Various designs of boxes and how these designs are interrelated is discussed. The chapter focuses mainly on electrodynamic motor units since most commercial loudspeakers use such units. The interaction between the loudspeaker and the room as well as the influence of the listening environment are also a part of this chapter.

Headphones and earphones are personal listening devices; these are treated in Chapter 15. They can be thought of as small loudspeakers that are placed close to or inside the ear canal. Various types of constructions are discussed such as electrodynamic, piezoelectric, and electrostatic headphones. Other differences between headphones and loudspeakers are also pointed out.

Digital sound reproduction is the subject of Chapter 16. Many errors can be introduced in the digitizing the representation of sound. The influence of sampling frequency, amplitude resolution and other factors on the reproduced sound is discussed. Examples are given of digital signal processing in audio and measurement. Such digital signal processing is important in the design of various types of codecs used, for example, for MP3, AAC, and other digital audio data storage and transmission systems.

The final chapter of the book deals with sound reproduction system aspects—mono, stereo, surround sound, binaural sound pickup, and reproduction. The chapter also gives an introduction to basic audio measurements using instruments and to subjective testing of audio using listening tests.

This book is intended as an Acoustics 101 for students of Sound and Vibration as well as students of electrical and electronic engineering, civil and mechanical engineering, computer science, signals and systems, and engineering physics. The course requires knowledge of basic engineering mathematics and physics. A solutions manual accompanies the book.

The author thanks all his colleagues at the Division of Applied Acoustics for their constant encouragement and interest, as well as views on the contents. The author especially wants to thank past

students, particularly MSc. Jonah Sacks and MSc. Jennifer Martin, as well as Dr. Peter Svensson, Dr. Bengt-Inge Dalenbäck, Dr. Per Sjösten, and Dr. Rendell R. Torres for making the author observe and correct a number of omissions, errors, and other shortcomings. Thanks are also due to Samuel Kleiner for solving many Mathematica related issues.

Gothenburg, Sweden, October 2010.
Mendel Kleiner

INTRODUCTION

ABOUT ACOUSTICS AND AUDIO TECHNOLOGY

The subject areas of acoustics and audio technology span a wide range of subareas and are of interest for many technical systems. Professionals, such as architects, town planners, civil engineers, doctors, biologists, oceanographers, electronics engineers, and computer scientists, as well as specialists in digital signal processing, are in contact with audio and acoustics in their work. The interest in the subjects is self-evident for radio, television, and other media professionals. Acoustical and audio engineering touch on many other areas of knowledge as indicated in Figure i.1.

AUDIO ENGINEERING

Knowledge of audio engineering technology is useful in dealing with recording, storage, and playback of sound such as speech, music, and other signals. The design of musical instruments, samplers, and synthesizers can be done better and with more success with acoustics theory, technology, design, and measurement. Computer games and virtual environments of many kinds can be rendered more efficiently and with better results, for enjoyment and presence, by using proper acoustics theory and audio technology.

Audio and acoustic engineering has one challenge, however. The final determination of quality is determined by listeners whose ability to hear features varies considerably between listeners and for a particular listener also with time, previous exposure to sound, general conditions, memory, among others.

Since audio engineering is a commercially important field, it is natural to think that over the years there would have been a development of adequate measurement techniques. In a sense, that is correct. By measurement of traditionally evaluated audio equipment properties, such as frequency response,

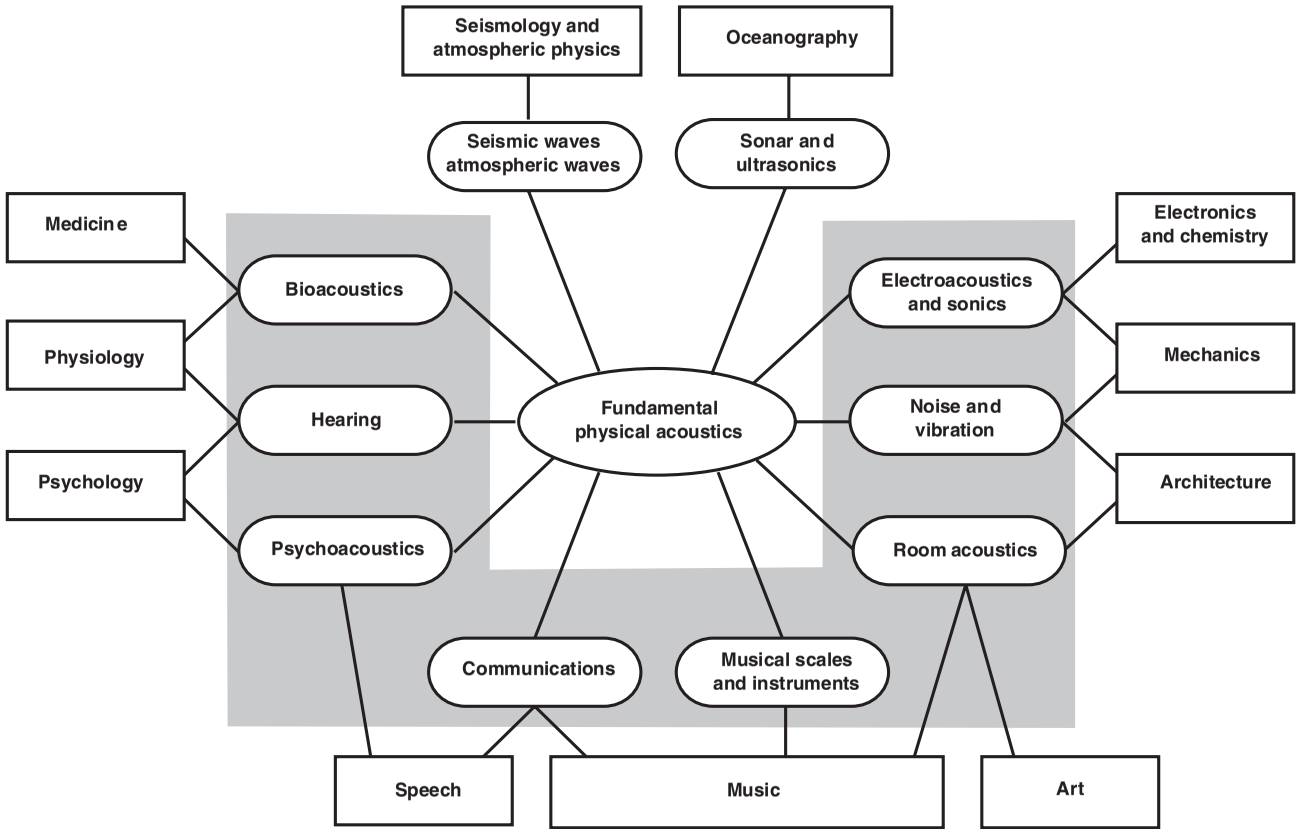


Figure i.1 The interdisciplinary world of acoustics: areas on gray background are of particular interest to the audio and acoustics professional (developed from Lindsay's "Wheel of Acoustics", Ref. i.22)

signal-to-noise ratio, harmonic and other forms of nonlinear distortion, much progress has been made. For high-end audio, however, the situation is quite unsatisfactory; it is difficult to separate marketing hype and blind faith from reality. One problem is that many audio equipment properties only become apparent after prolonged listening over weeks or months. This makes listening tests such as pair comparison tests unsatisfactory.

At the same time, these circumstances make audio engineering a quite fascinating field because there is always the possibility that your design is truly better than anyone else's; you just have to prove it. As Beranek writes, "It has been remarked that if one selects his own components, builds his own enclosure, and is convinced that he has made a wise choice of design, then his own loudspeaker sounds better to him than does anyone else's loudspeaker." (See Reference i.1.)

This book can only give a short introduction to the field; the reader is recommended to go on by reading the references. The books by Beranek and Olson (see References i.1-2) are classics in basic audio engineering; they are unique in their coverage and proper engineering treatment of the field. Recent books, such as the ones by Benson and Ballou (see References i.3-4), contain summaries by many specialists in the field. More specialized books on loudspeakers are the ones by Geddes, Borwick, and Colloms, ranging from science by way of engineering to subjectivism (see References i.5-7). Studio microphones are well covered in Gayford's book (see Reference i.8). An area not discussed at all in this book is audio electronics; the reader is advised to start by reading Linsley Hood's interesting book (see Reference i.9). Studio engineering and sound reinforcement engineering are fields that combine architectural acoustics with audio engineering; the books by Woram and Ahnert & Steffen cover these areas (see References i.10-11). The digitization of audio has introduced many new possibilities to manipulate sounds as well as many new types of effects that may or may not be audible; the book by Watkinson is an interesting starting guide to this field (see Reference i.12) as is the one by Pohlmann (see Reference i.23).

The Audio Engineering Society (http://www.aes.org) publishes the *Journal of Audio Engineering* and arranges many meetings and conferences that usually are documented by proceedings. It is the premier organization for audio engineering knowledge dissemination.

ACOUSTICS

With knowledge in acoustics, we wish to be able to design indoor and outdoor venues for public performance that are ideal for their intended purpose, for example, for speech, drama theater and musical performance. We want to design the venues so that they are excellent for listeners, speakers, singers and musicians, and others. We also want to help achieve a sustainable society, where noise exposure is voluntary and the acoustic climate agreeable to us.

It is virtually impossible to compare the acoustical* quality of rooms in an objective way, since human memory is short, and there are many variables outside the control of the scientist. It is a field where subjective judgment by single individuals, such as music critics, plays an inordinate role. Both

*Acoustic is used to modify terms that designate an object, or physical characteristics, associated with sound waves.

*Acoustical is used when the term being qualified does not designate explicitly something that has such properties, dimensions, or physical characteristics.

orchestras and audiences are influenced by the visual appearance of the hall, prior reports, and reputation, etc. Binaural sound rendering offers an avenue toward the goal of objective comparison but is not yet sufficiently perfected. Little objective research has been devoted to small room acoustics, such as that of home listening environments, cars, and even studios.

This book focuses on architectural acoustics, while discussing building acoustics and vibration isolation engineering on a small scale. The book by Raichel (see Reference i.13) covers many areas of engineering acoustics while still at an easy undergraduate level. A classic introduction to engineering acoustics is the book by Kinsler & Frey (see Reference i.14). A book discussing acoustics from the viewpoint of noise control is the one by Beranek (see Reference i.15). In modern product sound quality thinking, quiet is, however, not synonymous with quality; mechanical products are supposed to use appropriate sound to enhance their quality appearance—the car door slam or closing sound is but one example.

Architectural acoustics is another area where quality is difficult to assess. Many texts have been written on architectural acoustics, mostly from the viewpoint of the design of large auditoria, and then mostly with the audience in mind. The standard books are the ones by Kuttruff, Ando, Beranek, Barron, Egan, and Long (see References i.16-19, i.24). The books by Doelle and Cavenaugh & Wilkes look at architectural acoustics from the viewpoint of the architect (see References i.20-21).

The Acoustical Society of America (http://asa.aip.org) and the European Acoustics Association (http://www.eaa-fenestra.org) offer much information on acoustics through their journals and their conferences and meetings. For noise control, the International Institute of Noise Control Engineering is the premier information source (http://www.i-ince.org).

ABOUT THE AUTHOR

Mendel Kleiner is professor of acoustics at Chalmers University of Technology, Gothenburg, Sweden, in charge of the Chalmers Room Acoustics Group since 1989. Dr. Kleiner obtained his Ph.D. in architectural acoustics in 1978. He was professor of architectural acoustics at Rensselaer Polytechnic Institute, Troy, New York from 2003 to 2005. Kleiner is responsible for teaching room acoustics, audio technology, electroacoustics, and ultrasonics in the Chalmers master program on Sound and Vibration (http://www.ta.chalmers.se/intro/index.php). Professor Kleiner has brought his Chalmers group to the international research front of predictive room acoustics calculation, audible simulation, and 3-D sound. He returned to Chalmers in 2005 to continue leading the Chalmers Room Acoustics Group. He has more than 50 publications, has presented more than 110 papers and keynote lectures, has led courses at international conferences on acoustics and noise control, and organized an international conference on acoustics. Kleiner's main research areas are computer simulation of room acoustics, electro-acoustic reverberation enhancement systems, room acoustics of auditoria, sound and vibration measurement technology, product sound quality, and psychoacoustics. He is the author of *Audio Technology and Acoustics* published by the Division of Applied Acoustics, Chalmers and coauthor and editor of *Worship Space Acoustics* published by J. Ross Publishing, Ft. Lauderdale, FL. Kleiner is a Fellow of the Acoustical Society of America, Chair for the Audio Engineering Society's Technical Committee on Acoustics and Sound Reinforcement and its Standards Committee on Acoustics.

List of Symbols

The SI (metric) system of units is used in the book.

Upper case letters

A equivalent sound absorption area (m^2S metric sabin)
B bandwidth (Hz), magnetic field strength (Wb/m^2)
B' bending stiffness for sheets (Nm)
C capacitance (F)
C_A acoustic capacitance, acoustic compliance (m^3/Pa)
D damping, attenuation (dB), directivity index (dB), directivity factor
E modulus of elasticity (Pa), energy (J)
F force (N), directivity function
G shear modulus (N/m^2), power spectrum density (W/m^2)
I intensity (W/m^2), moment of inertia (m^4)
K_c coincidence number (m/s)
L level (dB)
L_I sound intensity level (dB). Normal reference level $1 \cdot 10^{-12}$ W/m^2
L_p sound pressure level (dB). Normal reference level $2 \cdot 10^{-5}$ Pa
L_u velocity level (dB). Normal reference levels are $1 \cdot 10^{-9}$m/s or $5 \cdot 10^{-8}$ m/s
L_W sound power level (dB). Normal reference level $1 \cdot 10^{-12}$ W
M mass, moment (Nm), molecular weight (kg)
M_A acoustic mass (kg/m^4)
N number of modes
P energy density (J/m^3)
Q volume flow (m^3)

R transmission loss (dB), viscous damping, resistance (Ns/m) (resistive part of impedance), gas constant, normalized autocorrelation
R_A acoustic resistance (Ns/m^5)
S surface area (m^2)
T reverberation time (s), absolute temperature (K), period (s)
U volume velocity (m^3/s)
V volume (m^3), potential energy (J)
W power (W)
X reactance (reactive part of impedance)
Z impedance (sound field impedance, Pa · s/m, acoustical impedance, Pa · s/m^3; mechanical impedance, Ns/m)

Lower case letters

a acceleration (m/s^2)
c speed of sound (m/s)
f frequency (Hz)
j imaginary unit, $j = \sqrt{-1}$
k wave number (m^{-1}), spring stiffness constant (N/m), Boltzmann's constant
m mass (kg), molecular damping coefficient (1/m)
m'' mass per unit area (kg/m^2)
n modal density (1/Hz)
p sound pressure (Pa)
q volume velocity (m^3/s)
r reflection coefficient, radius (m)
s radiation factor
t time (s)
u particle velocity (m/s)
u fluid velocity (m/s)
v velocity (m/s)

Greek letters

α absorption coefficient
β angle
γ standing wave ratio
δ damping constant (m^{-1})
ζ z-component of displacement (m), normalized impedance
η y-component of displacement (m), loss factor, efficiency
κ ratio between the specific heat at constant pressure and constant volume, $\kappa = C_P/C_V$
λ wavelength (m)
ν Poisson's ratio
ε extension

ξ x-component of displacement (m), relative standard deviation
ρ density (kg/m^3)
σ tension (Pa)
τ shear stress (Pa), transmission factor
φ angle
θ angle, temperature (°C)
ψ phase angle
ω angular frequency, $2\pi f$ (radians/s)
Δ difference
Λ constant

General symbols

$\hat{x}$ maximum value or peak value
$\bar{x}$ average of x over time
$\langle x \rangle$ average of x over space
$\tilde{x}$ rms value of x
$\underline{x}$ underline indicates that x is a complex quantity

Certain indices

$'$ per unit length
$''$ per unit area
0 normal condition, resonance, perpendicular to
b refraction, bending
c coincidence, critical
d diffuse
g limit
i incident
m average
r reflected
t transmitted
A acoustic, A-weighting
B bending wave, B-weighting
C C-weighting
L longitudinal
M mechanical, receiver
S transmitter
T transversal

Upper case letters are usually used to indicate amplitude (usually *A*, *B*) and number (usually *N*, *M*, *Q*).

Lower case letters are used for length and distance (usually *a*, *b*, *c*, *h*, *l*, *r*, *s*, *t*) and number (usually *l*, *m*, *n*, *q*).



This book has free material available for download from the
Web Added Value™ resource center at *www.jrosspub.com*

At J. Ross Publishing we are committed to providing today's professional with practical, hands-on tools that enhance the learning experience and give readers an opportunity to apply what they have learned. That is why we offer free ancillary materials available for download on this book and all participating Web Added Value™ publications. These online resources may include interactive versions of material that appears in the book or supplemental templates, worksheets, models, plans, case studies, proposals, spreadsheets, and assessment tools, among other things. Whenever you see the WAV™ symbol in any of our publications, it means bonus materials accompany the book and are available from the Web Added Value™ Download Resource Center at www.jrosspub.com.

Downloads for *Acoustics and Audio Technology, Third Edition*, consist of a solutions manual for instructors to end-of-chapter problems.

Sound 1

1.1 INTRODUCTION

We are familiar with the fact that a mechanical system consisting of an inter-coupled mass and spring can store mechanical energy in the forms of potential and kinetic energy. If the system is excited by an outside force, the system will respond by motion. When the force is eliminated, the energy stored will cause the system to oscillate—the system will move in a periodic way. For the simple system mentioned, the free motion will occur at one frequency only; we call this the resonance frequency. We say that the system is resonating, and unless there is damping in the system, the system will go on oscillating with the same amplitude forever. Damping always results when energy is removed from the oscillating system.

Friction and radiation are two processes by which energy can be removed from the vibrating system. Friction, also internal, leads to energy being transformed from motion into heat. Radiation of sound waves is an important way in which a mechanical system may lose energy. However, energy may also be lost by the mechanical system generating movement of air in close vicinity to the system if there are damping mechanisms that affect the air motion.

1.2 THE WAVE EQUATION

We use the expression *sound waves* to characterize wave motion in gases and fluids, such as air or water. Such waves are *longitudinal*, since fluids cannot exhibit shear motion. Longitudinal waves are characterized by the oscillation of the wave motion being in the direction of the *wave propagation*. The direction of the wave propagation is determined by the geometrical and vibrational properties of the radiating mechanical system as well as by the surrounding fluid and its boundaries.

Sound waves have *spatial extension*; the period of distance until the wave repeats its pattern of vibration is called the *wavelength* of sound. The wavelength depends both on the *speed of sound* in the medium and the *frequency of oscillation*. The basic governing equations lead to a differential equation, the *wave equation*. The wave equation describes the physics of wave motion, it defines the relationship between the spatial and temporal behavior of sound. The speed depends on the physical properties of the medium, such as the gas mixture. The wave equation also describes the propagation properties of sound waves.

The wave equation is derived from using three equations that each describes a particular property of the fluid. These equations are the *equation of motion*, the *equation of continuity*, and an equation describing the *thermodynamic properties* of the medium.

In the derivation of the wave equation, we assume the gas to be *elastic* and *continuous*. The small amount of gas considered in the derivation of the wave equation may be regarded as a *volume element* large enough to contain many molecules but still small enough for the *acoustical quantities* of interest to be constant throughout the element. Of course, in reality, molecules move in and out of the element, but the assumption is valid from a statistical point of view if the element is large enough.

Acoustic waves are characterized by quantities such as:

- *Sound pressure* is the dynamic excess pressure over the atmospheric static pressure in a small volume.
- *Particle velocity* is the average velocity of the gas molecules in a small volume.

Our derivation of the wave equation requires several things:

- The medium is considered to be *at rest*.
- The sound pressure must be small enough compared to the static pressure for the medium to be considered *linear*. In practice, the maximum sound pressure is assumed to be less than 0.001 times the static pressure.
- There can be no heat exchange within the medium; that is, no heat flows into or away from the volume element. The process is said to be *adiabatic*.
- There should be no losses in the medium. The medium does not exhibit *viscosity* or other phenomena leading to damping.
- The medium is *homogenous*—the effects of gravitation are not considered.

In many practical cases, it is necessary to take both losses, movement, and inhomogeneity, into account, such as when considering wave propagation over long distances, indoors, or outdoors. *Wind* and *temperature variations* cause the wave to change direction. *Internal losses* in the medium result in attenuation of the wave.

Plane Waves

Mathematically, it is relatively easy to derive the wave equation for wave motion in a one-dimensional medium. Let us consider nonviscous wave propagation in a gas, or in a gas mixture such as air, in a narrow, rigid tube at frequencies low enough for the wave propagation in the tube to be an approximation to one-dimensional wave propagation.

The Equation of Continuity

The *equation of continuity* expresses the relationship between density changes and volume changes in a volume element. The mass of the volume is considered constant. Assume that the volume element is V_0, which is trapped between the two planes at x and $x + dx$ in a rectangular canal having a cross section area of $dydz$ as shown in Figure 1.1, that is, $V_0 = dxdydz$. The particle is assumed to be displaced for some reason.

The volume element's displacement at x is ξ, and its displacement at $x + dx$ is $\xi + (\partial\xi/\partial x)dx$. The derivative of x with respect to time, $\partial\xi/\partial t$, is called the *particle velocity* and is denoted u_x. Note that the

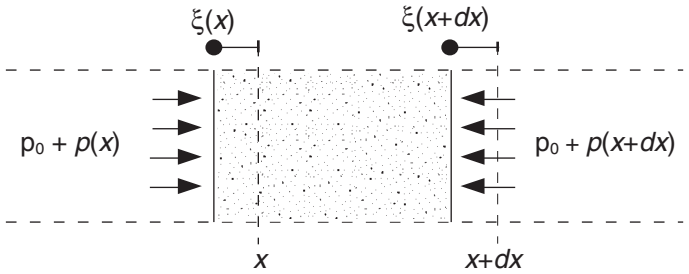


Figure 1.1 The small volume element under consideration. p_0 is the static pressure.

particle velocity is a *vectorial* quantity. The difference in displacement multiplied by the cross-sectional area *dydz* will be the change of volume ∂V, of the original volume V_0:

$$\partial V = \frac{\partial \xi}{\partial x} dx\, dy\, dz = \frac{\partial \xi}{\partial x} V_0 \tag{1.1}$$

Since the mass M_0 of the volume element will be unchanged, its density $\rho = M_0/V$ must change, leading to:

$$\frac{\partial \rho}{\partial V} = \frac{-M_0}{V^2} \qquad \rightarrow \left[\frac{\partial \rho}{\partial V}\right]_{V=V_0} = \frac{-M_0}{V_0^2} \tag{1.2}$$

When we combine Equations 1.1 and 1.2, we find that the change in the element's density is inversely proportional to its relative elongation, expressed as:

$$\frac{\partial \rho}{\rho_0} = -\frac{\partial \xi}{\partial x} \tag{1.3}$$

When we calculate the derivative of this relationship with respect to time, we obtain the equation of continuity:

$$\frac{\partial \rho}{\partial t} = -\rho_0 \frac{\partial u_x}{\partial x} \tag{1.4}$$

For the three-dimensional case, we write the equation of continuity:

$$\frac{\partial \rho}{\partial t} = -\rho_0 \mathrm{div}\, \vec{u} \tag{1.5}$$

where $\mathrm{div}\, \vec{u} = \partial u_x / \partial x$ for a one-dimensional problem.

The Equation of Motion

Newton's equation of motion states that the force *F* needed to accelerate a mass *m* at an acceleration of *a* is *F* = *ma*. Of course, this applies also to the mass of the gas in the volume element under consideration in our case. Again, consider the situation shown in Figure 1.1. We denote the static pressure by p_0 and the sound pressure by *p*.

The force needed to accelerate the volume element depends on the difference in pressure between the planes x and $x + dx$ respectively:

$$\left[p(x) - p(x+dx)\right]dy\,dz = -\frac{\partial p}{\partial x}dx\,dy\,dz \tag{1.6}$$

Since the acceleration of the volume element is written $\partial\xi^2/\partial t^2$, the equation of motion can be written as:

$$\rho_0 \frac{\partial^2 \xi}{\partial t^2} = -\frac{\partial p}{\partial x} \tag{1.7}$$

Usually it is more practical to express the movement of the volume element by its velocity, $\partial\xi/\partial t = u_x$, instead of by its acceleration, which allows us to express the equation of motion as:

$$\rho_0 \frac{\partial u_x}{\partial t} = -\frac{\partial p}{\partial x} \tag{1.8}$$

For the three-dimensional case, it is expressed:

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\operatorname{grad} p \tag{1.9}$$

where $\operatorname{grad} p = \partial u_x / \partial x$ for a one-dimensional problem.

Thermodynamic Properties

To analyze the dynamic properties of the gas, we also need to study the relationship between instantaneous density changes and sound pressure at a *temperature T* in the medium. If the system is adiabatic, the relationship between sound pressure and instantaneous volume is described by Poisson's equation:

$$(p_0 + p)V^{\kappa} = \text{constant} \tag{1.10}$$

Here κ is the ratio of specific heats at constant pressure and volume respectively; that is, $\kappa = C_p/C_v$. We also know that since the mass of the volume element is unchanged:

$$V(\rho_0 + \rho) = \text{constant} \tag{1.11}$$

Combining Equations 1.10 and 1.11, we obtain the partial derivatives as:

$$\frac{\partial p}{\partial V} = -\kappa\frac{p_0}{\rho_0} \quad \text{and} \quad \frac{\partial \rho}{\partial V} = -\frac{\rho_0}{V_0} \quad \text{i.e.,} \tag{1.12}$$

$$\frac{\partial p}{\partial \rho} = \kappa\frac{p_0}{\rho_0} \tag{1.13}$$

The *equation of state* for a gas is:

$$\frac{p_0}{\rho_0} = \frac{RT}{M} \tag{1.14}$$

where T is the temperature, M is the molecular weight of the gas or gas mixture, and R is the universal gas constant. Insertion of Equation 1.12 into Equation 1.11 gives the desired result:

$$\frac{\partial p}{\partial \rho} = \frac{\kappa RT}{M} \tag{1.15}$$

The Wave Equation

Combining Equations 1.4, 1.8, and 1.15, we obtain the wave equation for plane waves as expressed in sound pressure p:

$$\frac{\partial^2 p}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{1.16}$$

The wave equation may, of course, be expressed equally well in particle velocity, u. The quantity c depends on the thermodynamic properties of the gas:

$$c^2 = \frac{\kappa RT}{M} = \kappa\frac{p_0}{\rho_0} \tag{1.17}$$

For the three-dimensional case, the wave equation is written as:

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{1.18}$$

where $\nabla^2 p = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2}$ in a Cartesian coordiante system.

1.3 SOLUTIONS TO THE WAVE EQUATION

One way of finding solutions to an equation is to try various solutions to see if they satisfy the equation. If we use this approach in investigating the possible solutions to the wave equation (see Reference 1.16), a reasonable guess to mathematically describe waves is a general pair of functions such as:

$$p(x,t) = f_1(tc - x) + f_2(tc + x) \tag{1.19}$$

where f_1 and f_2 are arbitrary functions, required to have continuous derivatives of the first and second order. Testing this solution, we find that it satisfies the wave equation. Studying the properties of the solution, we see that the shape of the functions remains unchanged for variations of the variables x and t, but they are displaced in x according to the value of t. The relationship between the time variable t and the space variable x is $x = tc$. The shape of the wave moves with the speed c, which we call the *propagation speed of sound*.

We note from Equation 1.17 that the propagation speed in a gas depends only on temperature. Under normal conditions—for example, in air—close to a temperature of 20°C, the propagation speed is:

$$c_0 = \sqrt{\kappa\frac{p_0}{\rho_0}} \approx 331.29 + 0.606t_{^\circ\mathrm{C}} \quad [m/s] \tag{1.20}$$

where $t_{^\circ\mathrm{C}}$ is the temperature of the air expressed in degrees Celsius. At a temperature of 20°C, the speed of sound is $c_0 \approx 343.4$ m/s.

Since the shape of the wave does not change as it moves, we understand from Fourier theory that all frequency components, building up the shape of the wave, propagate at the same velocity. Sound propagation in gases is fairly independent of frequency, but frequency-dependent phenomena such as damping is discussed later. The term *nondispersive* is used to describe the fact that the propagation speed is frequency independent.

Stationary, One-dimensional Solutions for Cartesian Coordinates

In the rest of the discussion of the physical principles of sound and vibration, we will only study *continuous sinusoidal sound and vibration*. We know that according to the Fourier theorem, any physical wave shape may be obtained by suitable superposition of sines and cosines. It is practical to use the transform method called the *jω-method* to describe time variation as well as various properties of sound and media. The variables of sound pressure and particle velocity, for example, then become complex variables, which greatly simplifies the mathematics. The physical variables of instantaneous sound pressure and particle velocity are given by the real part of the complex quantities—designated by an underscore—for example $\underline{p}$, in equations. Our use of the *j*ω-method is discussed in Chapter 2, and more information in general on the method can be found in Reference 1.3.

Using the *j*ω-method, we can rewrite the wave equation as written in Equation 1.16 to:

$$\frac{\partial^2 \underline{p}}{\partial x^2} + k^2 \underline{p} = 0 \tag{1.21}$$

Here k is the *wave number*, sometimes called the *propagation constant*. The relationships between the wave number, the *speed* of sound c, *frequency* f, and *wavelength* λ, are given by $k = \omega/c = 2\pi f/c = 2\pi/\lambda$.

Using the *j*ω-method, we can write, as a special case, a possible solution to the wave equation as written in Equation 1.21 to:

$$p(x, \omega, t) = \hat{p}_+ e^{j(\omega t - kx)} \tag{1.22}$$

This equation describes a one-dimensional sinusoidal wave propagating in the direction of positive x. The magnitude of the sound pressure $p(x,t)$ is denoted $\hat{p}_+$. Generally, we do not explicitly write the time variation when we use this transform method, instead we write $\underline{p}(x)$. The general solution to the wave equation in the one-dimensional case also has to feature a wave propagating in the direction of negative x and is written as:

$$p(x, k) = \hat{p}_+ e^{-j(kx + \beta_+)} + \hat{p}_- e^{j(kx + \beta_-)} \tag{1.23}$$

where β_- and β_+ are case-dependent phase constants.

Stationary, One-dimensional Solutions for Spherical Coordinates

Using the formulation of the wave Equation 1.18, one can write the general solutions for the case of *spherical symmetry* as:

$$\underline{p}(r, k) = \frac{A_+ e^{-j(kr + \beta_+)}}{r} + \frac{A_- e^{j(kr + \beta_-)}}{r} \tag{1.24}$$

where A_+ and A_- are the wave amplitudes at 1 m distance, for the outward and inward moving waves respectively, and r is the radius from the center of origin. The angles β_- and β_+ are case-dependent constants.

For the case of spherical symmetry, we are generally only interested in the wave that moves outward, in the direction of positive r. Using the equation of motion, we can show that the particle velocity of such an expanding wave is given by:

$$\underline{u}(r, k) = \frac{A_+}{\rho_0 c_0 r}\left(1 + \frac{1}{jkr}\right) e^{-jkr} \tag{1.25}$$

We now have an additional term $1/jkr$. This implies that there will be a phase difference between $\underline{p}$ and $\underline{u}$ that will be dependent on the value of kr; that is, on frequency and distance to origin. The phase difference will be largest when $kr \ll 1$; that is, when the frequency is low and/or the distance to the center is small.

For the *near-field region*, where $kr \ll 1$, the radial particle velocity is described by:

$$\underline{u}(r,k) \approx \frac{A_+}{j\rho_0 c_0 kr^2} e^{-jkr} \tag{1.26}$$

We see that there is a phase difference of 90° between particle velocity and sound pressure; the particle velocity lags behind the sound pressure. Another observation of importance is that the amplitude of the particle velocity u_r increases proportionally to $1/r^2$ as the distance to the origin is diminished.

For the *far-field region*, where $kr \gg 1$, the radial particle velocity is described by:

$$\underline{u}(r,k) \approx \frac{A_+}{\rho_0 c_0 r} e^{-jkr} \tag{1.27}$$

In this outer region, particle velocity and sound pressure are in-phase. The amplitude of both quantities also depends on distance in the same way in the far-field region.

To radiate an ideal spherically symmetrical wave, a radiator would have to feature a spherical surface moving at the same radial velocity at all points. To be a *point source*, such a radiator would be infinitely small. We realize that it is not physically possible to have such a radiator. However, we use the terms *point source* and *monopole* to label radiators that, from our point of observation, behave as if they were ideally small and radially radiating. One way of approximating such radiators, under certain conditions, is to use small loudspeakers.

A *loudspeaker* usually consists of a *loudspeaker box* and a *loudspeaker driver* (motor). If we have a loudspeaker box designed in such a way that only one side of the loudspeaker driver's diaphragm is facing the exterior of the box, and if the *dimensions* of the loudspeaker box, d, are much smaller than the wavelength of sound being generated—that is, $kd < 0.1$—then the sound field radiated by the loudspeaker will be a fairly good approximation to a spherical sound field at a sufficiently large distance.

It is important to realize that close to the loudspeaker the sound field will not have the characteristics of a true, spherical sound source. Figure 1.2 shows a loudspeaker box using a dodecahedron arrangement of loudspeaker drivers to approximately achieve omnidirectional characteristics. Typically, the deviations start to become large when $ka > 3$, where a is the *radius* of the dodecahedron.

One might think that it would be easy to simulate a spherical source by using a large number of small loudspeakers mounted on a sphere. However, because of the finite difference in distance between the loudspeaker diaphragms as well as the fact that these usually have a conical shape and are not a continuous part of the spherical surface, we find that the radiation pattern is not ideally spherical at frequencies where the distance between loudspeakers or the depth of the cones are larger than a small fraction of the wavelength.

A different way of studying the properties of a radiator is to study its *impulse response* in various directions. If the surface of the spherical radiator could move with infinite acceleration for a short time, the radiated sound pressure would be a *Dirac pulse*, since sound pressure is proportional to the volume acceleration of the sound source. Only if the source has the radiation properties of an ideal monopole can we have a Dirac-type sound pressure response in all directions.

Notice that only a truly spherical source can radiate having this ideal *geometrical* impulse response. The geometrical impulse response of a dodecahedron-shaped loudspeaker, for example, having ideal



Figure 1.2 A loudspeaker using a dodecahedron arrangement of drivers to approximate the omnidirectional sound radiation characteristics of a monopole. (Photo by Mendel Kleiner.)

loudspeaker drivers—all in-phase, on all its flat surfaces—will feature impulse response contributions due to edge diffraction. Correspondingly, the frequency response of the loudspeaker will feature frequency response irregularities.

1.4 IMPEDANCE

The ratio between two interdependent quantities—such as force and linear velocity—at some point in a system is called *impedance*, $\underline{Z}$. The inverse of impedance is usually called mobility, $\underline{Y}$.

Both impedance and mobility are generally frequency-dependent, complex quantities, characterized by a real and an imaginary part, or, if we prefer, by magnitude and phase. The real part of impedance is usually called *resistance*, and its imaginary part is called *reactance*.

Definitions of Impedance Used in Acoustics

Because ratios between various sound field quantities often play an important role in problem-solving processes in acoustics, some impedances have been defined in various ways. Note that each impedance is characterized by its own unit or combination of units. A summary of the impedance relationships is shown in Table 1.1.

Characteristic impedance, $\underline{Z}_0$, is the ratio of sound pressure to particle velocity in an infinite plane wave. The unit of characteristic impedance is Rayl; that is, Ns/m^3. The characteristic impedance is usu-

Table 1.1 Summary of the impedance relationships.

Multiply by / to obtain	Z_M	Z_S	Z_A
Z_M	1	S	S^2
Z_S	$1/S$	1	S
Z_A	$1/S^2$	$1/S$	1

ally resistive if there are no propagation losses in the medium, and equal to the product of the density of the medium and the propagation speed of sound in the medium; that is, $Z_0 = \rho_0 c_0$.

Sound field impedance, $\underline{Z}_S$, is the ratio of sound pressure to particle velocity in the reference direction at a chosen point in the medium. Sometimes the term *specific acoustical impedance* is used for this quantity. The unit of sound field impedance is Ns/m^3.

Acoustical impedance, $\underline{Z}_A$, is the ratio of sound pressure to volume velocity in the normal direction to a reference surface in an acoustic system. Volume velocity is the product of the normal direction of the particle velocity and the surface area under consideration. The particle velocity is then considered constant regarding amplitude and phase over the surface. The unit of acoustical impedance is the *acoustical ohm*; that is, Ns/m^5.

Mechanical impedance, $\underline{Z}_M$, is the ratio of force to velocity at a chosen point in a mechanical system. The unit of mechanical impedance is the *mechanical ohm*; that is, Ns/m.

Characteristic and Sound Field Impedances in a Plane Wave

Using Equation 1.8, we find the sound field impedance for a plane wave propagating in the direction of positive x:

$$\frac{\underline{p}_+}{\underline{u}_{x+}} = \rho_0 c_0 = Z_0 \tag{1.28}$$

We see in this case that the sound field impedance is the same as the characteristic impedance of the medium. Obviously, the sound pressure and the particle velocity are *in-phase*.

In the same way, we obtain the sound field impedance for a plane wave propagating in the direction of negative x:

$$\frac{\underline{p}_-}{\underline{u}_{x-}} = -\rho_0 c_0 = -Z_0 \tag{1.29}$$

The minus sign in Equation 1.29 is a result of our choice of always regarding positive particle velocity as being in the direction of positive x, even for the wave that propagates in the negative x direction.

Sound Field Impedance in a Spherical Wave

The sound field impedance in a spherical wave varies according to the distance to the origin. Using Equation 1.25, we can show that the sound field impedance of a spherical, expanding wave varies with distance to the origin r as:

$$\underline{Z}_S(r,k) = \frac{\underline{p}(r,k)}{\underline{u}_r(r,k)} = Z_0 \frac{jkr}{1+jkr} = Z_0 \frac{1}{\frac{1}{jkr}+1} \tag{1.30}$$

In the *near-field* region ($kr \ll 1$), for low frequencies and/or small distances to the origin, the sound field impedance of the spherical wave is nearly pure reactance and has a *mass-type* character. That means that the reactance is positive and increases proportionally to frequency:

$$\underline{Z}_S \approx jkrZ_0 = j\frac{\omega}{c_0} r\rho_0 c_0 = j\omega\rho_0 r \quad ; \quad kr \ll 1 \tag{1.31}$$

If the sound field impedance is primarily reactance, as it is close to a small sound source, the sound field is said to be a *reactive near-field*.

In the *far-field* region ($kr \gg 1$), that is, for high frequencies and/or large distances to the origin, the sound field impedance of the spherical wave will be almost real and equal to the characteristic impedance of the medium:

$$\underline{Z}_S \approx Z_0 \quad ; \quad kr \gg 1 \tag{1.32}$$

Note that, in the far-field region, the sound field impedance of an expanding spherical wave is the same as that for an infinite plane wave.

1.5 SOUND INTENSITY

Wave propagation is characterized by transport of energy in the direction of the wave. The energy is transported at a velocity called the *group velocity* c_g. If the medium is *dispersive*, that is, the group velocity varies by frequency, the group velocity will be different from the *phase velocity* c_{ph}—that is, the velocity needed to always see the same phase in the wave. For sound waves in air, the difference between group velocity and phase velocity is usually negligible.

Sound intensity is a measure of the rate of energy transport per unit area in the wave. The sound intensity is a vectorial quantity and is directed along the direction of particle velocity:

$$\vec{I} = \frac{1}{2}\mathrm{Re}\left[\underline{p}\vec{\underline{u}}\right] \tag{1.33}$$

We know for an infinite plane wave in the positive x-direction that sound pressure and particle velocity will be in-phase, which results in:

$$I_x = \tilde{p}\tilde{u}_x = \frac{\tilde{p}^2}{Z_0} = Z_0\tilde{u}^2 \tag{1.34}$$

In this text, the ~ sign is used on top of letters to denote that the *root of the mean square value*—often called the *rms value*—is intended.

The sound intensity in a spherical wave is:

$$I_r(r) = \mathrm{Re}\left[\underline{Z}_S\right]\tilde{u}_r^2 \tag{1.35}$$

where

$$\mathrm{Re}\left[\underline{Z}_S\right] = Z_0\frac{(kr)^2}{1+(kr)^2} \tag{1.36}$$

We note that, according to Equation 1.35, the intensity in the sound field depends on the variables k and r. For a spherical sound source, the sound power radiated, for a given value of surface normal velocity u_0 and wave number k, will be much less for a small source than for a large source.

1.6 SOUND POWER

We can calculate the *sound power* being transported by a sound field by integrating the component of sound intensity normal to the surface over the surface area in question.

Consider a plane wave of intensity I [W/m^2] perpendicularly incident on a surface of area S. The intensity of the plane wave is given by Equation 1.34:

$$W = IS = \tilde{p}\tilde{u}S = \frac{\tilde{p}^2}{Z_0}S = Z_0\tilde{u}^2S \tag{1.37}$$

Under a short time dt, the energy $Wdt = Sc_0t$ will be traveling through the surface. This energy is confined to a volume $Sdx = Sc_0dt$. Consequently, the energy density P in this volume must be:

$$P = \frac{\text{energy}}{\text{volume}} = \frac{I}{c_0} = \frac{\tilde{p}^2}{\rho_0 c_0^2} \tag{1.38}$$

Equation 1.32 shows us that since the sound field impedances for a plane wave and a spherical wave under the condition $kr \gg 1$ are nearly the same, this expression can be used to calculate the sound power in a spherical wave at significant distances and/or high frequencies.

The radial sound intensity at a distance r from the origin of a spherical wave depends on the radiated sound power W as:

$$I_r(r) = \frac{W}{4\pi r^2} \tag{1.39}$$

The intensity of sound in a spherical wave diminishes as $1/r^2$ and the sound pressure as $1/r$ as a function of distance from the point source. We call this the *distance law* for the geometrical sound attenuation of a point source.

Using Equations 1.35–1.36, we can determine the radiated power of a small spherical sound source having a radius of a:

$$W = Z_0 \frac{(ka)^2}{1+(ka)^2} \frac{\tilde{U}^2}{4\pi a^2} \tag{1.40}$$

Here, of course, the *volume velocity* is $\tilde{U} = 4\pi a^2 \tilde{u}$. That is:

$$W \approx \frac{\pi \rho_0}{c_0} f^2 \tilde{U}^2 \quad \text{if } ka \ll 1 \tag{1.41}$$

1.7 PROPAGATION LOSSES

During the propagation of sound away from a small sound source, the amplitude and intensity of the wave are reduced not only because power is spread over a larger area—*geometrical attenuation*—but also because of various loss mechanisms that turn the sound energy in the sound wave into heat. These losses are known as excess absorption. The excess absorption is usually accounted for by simply introducing a damping term in the expression for the intensity as a function of propagation distance x:

$$I(x) \propto e^{-mx} \tag{1.42}$$

that is:

$$\tilde{p}(x) \propto e^{-mx/2} \tag{1.43}$$

where m is the attenuation coefficient.

The losses in sound propagation in air are due to *heat conduction* (not perfectly adiabatic processes), *viscous losses*, and *relaxation* phenomena. The relaxation phenomena are responsible for the major part of the losses at normal conditions. The attenuation depends both on *temperature* and *relative humidity*.

The attenuation due to heat conduction and viscous losses increases proportionally to frequency squared. The relaxation causes an attenuation that peaks at certain frequencies. The relaxation process involves excitation of the gas molecules to other energy states, for example, from translational energy to vibrational/rotational energy. The relaxation time depends on the time that the molecule remains in the excited state. Maximum attenuation will occur when the period of oscillation in the sound wave is approximately the same as the relaxation time.

Figure 1.3 shows curves of the frequency-dependent attenuation of sound propagation in air at a temperature of 20°C for some values of the relative humidity (see also Figure 4.9). Using these curves, it is possible to calculate corresponding values for the attenuation coefficient m. Notice that the gas mixture of air involves several relaxation processes.

The standard ISO 9613-1:1993 (E) can be used to calculate the attenuation of sound in air for general combinations of temperature and relative humidity.

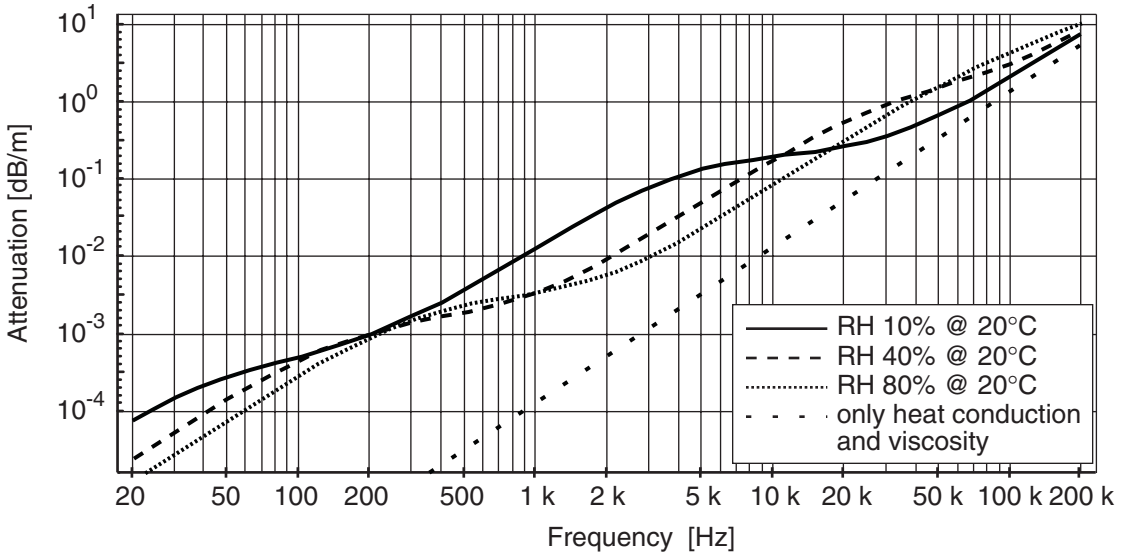


Figure 1.3 Attenuation in dB/m for sound propagation in air as a function of frequency at a temperature of 20°C and with relative humidity, RH, as a parameter. (After Reference 1.2) The graph can be used to calculate the attenuation coefficient m since the attenuation L in dB over a distance x is $L \approx 4.3\,mx$.

1.8 REFLECTION AND TRANSMISSION AT BOUNDARIES

When a sound wave is incident on the boundary between two media, or on a boundary where the characteristic impedance of a medium changes, part of the power in the incident wave may be reflected and some transmitted. The ratios of reflected and transmitted powers to that incident are determined by the characteristic impedances Z_{01} and Z_{02} at each side of the boundary as well as on the angles of incidence and transmission relative to the boundary.

If the boundary is not plane, but has a surface structure that has random variations in height, a major part of the reflection and transmission will be diffuse if the rms value of the height variations are of a size similar to the wavelength of the incident sound. By using special surface structures, one can obtain nearly perfect diffuse reflection (see Section 7.4). We will study the case of plane boundaries in this chapter.

Because of the *interference* between the incident sound wave and the reflected sound wave, note the *standing-wave phenomena* as the waves add up, in- or out-of-phase.

Normal Sound Incidence

Assume a plane wave incident on the plane impedance boundary at $x = 0$. Also, assume the angle of incidence to be 0° (*normal incidence*). The case of 90° incidence is called *grazing incidence*.

Because of the *linearity of the media* and the *coherence between the incident and reflected waves*, the resulting sound pressures and particle velocities are obtained by simple superposition of the incident and reflected sound pressures $\underline{p}_i$ and $\underline{p}_r$ and particle velocities $\underline{u}$ and $\underline{u}_r$, respectively.

Along the plane, the *boundary conditions* to be fulfilled are *continuity of pressure* and *continuity of particle velocity*, that is:

$$\left.\begin{aligned} \underline{p}_1 &= \underline{p}_2 \\ \underline{u}_1 &= \underline{u}_2 \end{aligned}\right\} \quad \text{at } x = 0 \tag{1.44}$$

where $\underline{p}_1$, $\underline{p}_2$ and $\underline{u}_1$, $\underline{u}_2$ are the sound pressures and particle velocities on the respective sides of the boundary, $x > 0$ and $x < 0$:

$$\underline{p}_1 = \left(\underline{p}_i + \underline{p}_r\right) \tag{1.45}$$

where

$$\underline{p}_i = \hat{p}_i e^{-jk_1 x} \tag{1.46}$$

$$\underline{p}_r = \hat{p}_r e^{jk_1 x} \tag{1.47}$$

$$\underline{p}_2 = \underline{p}_t = \hat{p}_t e^{-jk_2 x} \tag{1.48}$$

Because of the definition of sound field impedance, we find:

$$\underline{u}_1 = \left(\underline{u}_i + \underline{u}_r\right) = \frac{\underline{p}_i - \underline{p}_r}{\underline{Z}_{01}} \tag{1.49}$$

$$\underline{u}_2 = \underline{u}_t = \frac{\underline{p}_t}{\underline{Z}_{02}} \tag{1.50}$$

where index t means transmitted wave. If $\underline{u}_1$, $\underline{u}_2$, $\underline{p}_1$, and $\underline{p}_2$ are eliminated from these equations, we further find:

$$\underline{Z}_{01} \frac{\underline{p}_i + \underline{p}_r}{\underline{p}_i - \underline{p}_r} = \underline{Z}_{02} \qquad \text{at } x = 0 \tag{1.51}$$

The relationships between the sound pressures, in the respective waves, are usually expressed by a *pressure transmission coefficient* $\underline{t}$ and a *pressure reflection coefficient* $\underline{r}$ as:

$$\underline{t} = \frac{\underline{p}_t}{\underline{p}_i} = \frac{2\underline{Z}_{02}}{\underline{Z}_{02} + \underline{Z}_{01}} \tag{1.52}$$

$$\underline{r} = \frac{\underline{p}_r}{\underline{p}_i} = \frac{\underline{Z}_{02} - \underline{Z}_{01}}{\underline{Z}_{02} + \underline{Z}_{01}} \tag{1.53}$$

where index i means incident wave. These transmission and reflection coefficients are usually complex quantities. It is only when the incident wave is plane and the boundary plane is infinite that the reflection and transmission can be formulated in this simple way.

Associated with these reflection coefficients, we also define the *intensity transmission coefficient* τ and the *intensity reflection coefficient* γ:

$$\tau = \frac{I_t}{I_i} \tag{1.54}$$

$$\gamma = \frac{I_r}{I_i} \tag{1.55}$$

If there are no losses at the boundaries, the incident, reflected, and transmitted sound powers (W_i, W_r, and W_t) must be related as:

$$W_i = W_r + W_t \tag{1.56}$$

that is:

$$\gamma + \tau = 1 \tag{1.57}$$

From the side of wave incidence, it seems as if the energy of the transmitted wave is absorbed. One usually calls the quantity $(1 - \gamma)$ the sound-absorption coefficient, α; that is:

$$\alpha = 1 - |\underline{r}|^2 \tag{1.58}$$

Quasi-plane waves can be generated in long, straight tubes if the tube walls are hard and the tubes are wide enough, so that the influence of the viscosity of the medium can be neglected. The tubes must not be so wide as to allow crosswise sound propagation. Such tubes can be used as acoustical components to build various types of acoustical circuits, as described in Section 1.7.

Oblique Sound Incidence

Oblique incidence means that the angle of incidence of the sound is larger than zero, relative to the normal of the boundary. At oblique incidence, the reflection and transmission coefficients will be dependent on the angles of incidence and transmission. If the phase velocities in the two media are different, the transmitted wave will be *refracted*. Two cases will be studied: *normal reaction* and *Rayleigh type of reflection*.

Normal reaction (sometimes called *local reaction*) means that the sound wave transmitted into medium 2 propagates at right angles to the boundary. Normal reaction is accomplished when the sound—for example, due to a guiding structure—is forced to propagate in a particular direction. This is similar to when sound is transmitted into a porous absorber.

Rayleigh type of reflection (also known as *extended reaction*) means that the sound wave transmitted into medium 2 propagates at an angle to the boundary, determined by Snell's law (see Equation 1.6). Extended reaction typically occurs at boundaries between media in which free wave propagation is possible, for example, a large surface between water and air.

We will now study the conditions for a plane wave incident at an oblique angle as shown in Figure 1.4. Besides the requirement for continuity of sound pressure and particle velocity, according to Equation 1.44, it is also necessary for the three sound waves on either side of the boundary to propagate at the same phase velocity along the boundary. This means that the angle of reflection must be the same as the angle of incidence.

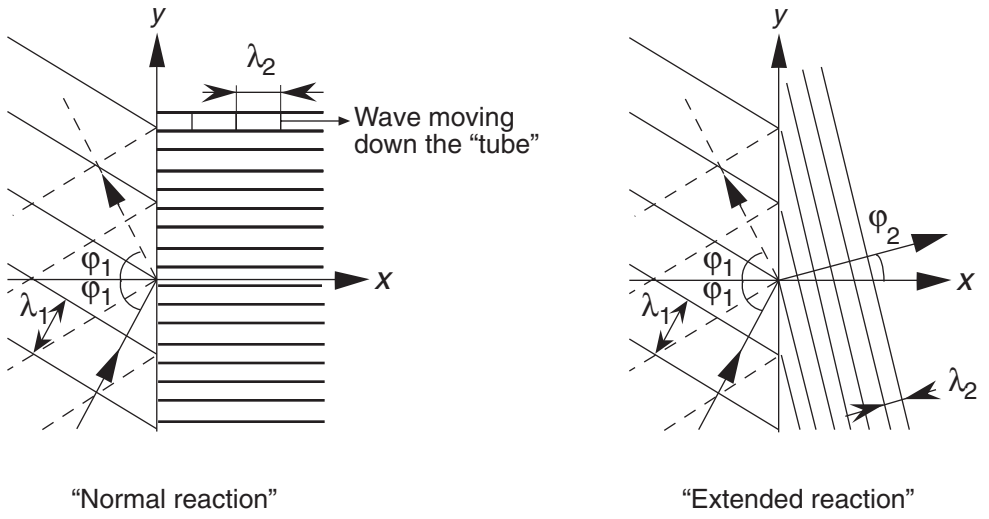


Figure 1.4 Oblique sound incidence on a plane impedance boundary.

The wave vector components in the y-direction must be the same for all three waves, since the wave numbers are based on the frequency of sound and the phase velocity. The wave number, which we have studied so far, is just the magnitude of the wave vector. In the present case, it is practical to regard the wave numbers for waves traveling at oblique angles as components of wave vectors in the x- and y-directions.

The angle of reflection is given by the equality of the wave vector components on either side in the y-direction, $k_{1y} = k_{2y}$; that is:

$$k_1 \sin(\varphi_1) = k_2 \sin(\varphi_2) \tag{1.59}$$

It can be reformulated to:

$$\frac{c_1}{\sin(\varphi_1)} = \frac{c_2}{\sin(\varphi_2)} \tag{1.60}$$

where c_1 and c_2 are the speed of sound in the two regions. This condition is known as *Snell's law*.

Normal Reaction

The obliquely incident sound wave can be formulated as:

$$\underline{p}_1 = \hat{p}_1 e^{-jk_1(x\cos(\varphi_1)+y\sin(\varphi_1))} \tag{1.61}$$

Since the reflection leads to the wave vector component in the x-direction of the reflected wave changing sign, one can write the resulting sound field in medium 1, using superposition, as:

$$\underline{p}_1 = \underline{p}_i + \underline{p}_r = \hat{p}_i e^{-jk_{1y}y}\left(e^{-jk_{1x}x} + \underline{r}e^{jk_{1x}x}\right) \tag{1.62}$$

where

$$k_{1x} = k_1 \cos(\varphi_1)$$
$$k_{1y} = k_1 \sin(\varphi_1) \tag{1.63}$$

The particle velocity can now be obtained using the definition of sound field impedance for a wave moving in the negative x-direction (see Equation 1.29):

$$\underline{u}_{1x} = \cos(\varphi_1) \frac{\hat{p}_i}{\underline{Z}_{01}} e^{-jk_{1y}y} \left(e^{-jk_{1x}x} - \underline{r} e^{jk_{1x}x} \right) \tag{1.64}$$

Let us now study the conditions on the side of medium 1. Remember that in the case of normal reaction studied now, the particle velocity in medium 2 is only in the x-direction; there is no propagation in the y-direction. The local complex sound field impedance on the side of medium 2 at the boundary $x = 0$ is written as:

$$\underline{Z}_{02} = \frac{\underline{p}_2}{\underline{u}_2} \tag{1.65}$$

Consequently, the boundary conditions for the particle velocity will be fulfilled if:

$$\frac{\underline{p}_1(0)}{\underline{u}_{1x}(0)} = \underline{Z}_{02} \tag{1.66}$$

This results in the pressure reflection coefficient $\underline{r}$:

$$\underline{r} = \frac{\underline{Z}_{02} \cos(\varphi_1) - \underline{Z}_{01}}{\underline{Z}_{02} \cos(\varphi_1) + \underline{Z}_{01}} \tag{1.67}$$

One notes that the reflection coefficient is dependent not only on the characteristic impedances of the two media but also on the angle of incidence (see Section 1.8).

Rayleigh Type of Reflection

In this case, the boundary condition with regard to the propagation direction is given by Snell's law. Of course, the regular requirements for sound pressure and particle velocity continuity across the boundary still apply.

Utilizing the same approach previously used, we obtain the reflection coefficient in the case of extended reaction as:

$$\underline{r} = \frac{\underline{Z}_{02} \cos(\varphi_1) - \underline{Z}_{01} \cos(\varphi_2)}{\underline{Z}_{02} \cos(\varphi_1) + \underline{Z}_{01} \cos(\varphi_2)} \tag{1.68}$$

The propagation velocities of sound in the two media influence the pressure reflection and transmission coefficients as given by Snell's law.

Note that:

$$\cos(\varphi_2) = \sqrt{1 - \left(\frac{c_2}{c_1} \sin(\varphi_1) \right)^2} \tag{1.69}$$

In some particularly interesting acoustical cases (for example, in some porous sound absorbers), the speed of sound in medium 2 is considerably smaller than in medium 1; that is, $c_2 \ll c_1$.

This means that $\cos(\varphi_2) \approx 1$, that is, $\varphi_2 \approx 0$ and, on the whole, independent of the angle of incidence φ_1. The propagation in medium 2 is then approximately equal to the case of normal reaction and the propagation in the *y*-direction is neglected.

When the angle of incidence φ_1 becomes close to 90°—grazing incidence—the pressure reflection coefficient $\underline{r}$ will become −1 if medium 2 has a finite characteristic impedance. This means that all energy will be reflected, and that the phase of the reflected wave will be inverted. In many situations, this causes the phase inverted wave canceling sound that is not reflected, leading to substantial cancellation of the sound pressure close to the sound-absorbing surface. Two examples of this cancellation at shallow angles of incidence are when sound from vehicles propagates nearly parallel to grassy ground and when sound from an orchestra propagates almost parallel to the audience seating area in an auditorium.

1.9 ACOUSTICAL COMPONENTS AND CIRCUITS

In many technical applications, one needs acoustic frequency response functions, or impedances, which vary in a predetermined way. Such applications can be found in acoustic measurement and excitation systems, and in transducer design, such as in the design of loudspeakers and microphones.

To obtain the desired response or impedance, it is practical to use electroacoustical analogies where the desired response is obtained by combination of discrete *acoustical* inductors, capacitors, and resistors. The circuits can then be analyzed either from the electrical or acoustical viewpoint (see Reference 1.1).

In this section, we discuss how, as long as the component size is small compared to the acoustic wavelength, this approach can be used to obtain acoustical equivalents of such electrical components. It is much easier, of course, to build discrete electrical components than acoustical components, because the speed of electromagnetic waves is approximately one million times greater than that of sound in air, making the wavelengths about one million times longer.

For the design of acoustical components to be practical, by the simple theory shown here, it is necessary to assume plane waves of sound. As discussed previously, it is possible to obtain approximations to plane waves in tubes over a certain frequency range. The frequency range is limited upward due to the possibility of oblique waves in the tubes at high frequencies. We will study hard-walled tubes.

In tubes having a rectangular cross section, the maximum distance between two parallel sides must be smaller than 0.5 λ. For tubes with a circular cross section, the diameter must be smaller than approximately 0.59 λ. In practice, it is wise to choose maximum cross section values less than 0.5 λ. The frequency range is limited downward because of the viscous layer at the tube walls. The layer becomes more important at low frequencies, and it prevents the particle velocity from being constant over the tube cross section. A commonly used limit is that the minimum dimension should be larger than 0.003 λ.

It is best to study the components using the idea of acoustical impedance, since the components can then be interconnected in the acoustical circuit as if they were electrical components. At the junctions, there will be continuity of pressure and volume velocity. In the acoustical circuits, the volume velocity is analogous to electrical current and pressure to electrical voltage. The pressure drop in an acoustical circuit is equivalent to the voltage drop in an electrical circuit.

A sound wave traveling down an infinite tube, with cross-sectional area S, sees a real acoustical impedance (*acoustical resistance*), which is $\rho c/S$. Obviously, making acoustical resistors by using infinite tubes is impractical and other ways have to be used. The design of pure acoustical resistance is difficult, however, and not studied here. In practice, one uses fine meshes or open-pore foam plastics and sintered materials (see Reference 1.1). In such acoustic resistors, one uses viscous losses to obtain the resistance.

If the tube is finite, the wave propagating in the tube will see changes to the acoustical impedance as the cross-sectional area is changed or if the tube end is closed or open. This will result in reflection of the sound wave and, consequently, standing waves in the tube. The local acoustical impedance corresponding to the sound field impedance discussed previously will then depend on the position in the tube.

By using abruptly close- and open-ended tubes, one can obtain *acoustical compliance components* and *acoustical mass components*, *acoustic inductors*, and *acoustic capacitors*. The compliance components will have impedance—*reactance*—proportional to $1/j\omega$, and the mass components will have impedances—*reactance*—proportional to $j\omega$.

Acoustical Compliance Components

Figure 1.5 shows a cut through a cylindrical tube, having a cross section area $S = \pi b^2$ and maximum and minimum dimensions corresponding to the limits mentioned previously.

Assume a plane wave propagating in the positive x-direction:

$$\underline{p}(x,k) = \hat{p}_+ e^{-jkx} \tag{1.70}$$

If the tube is truncated by a hard wall at $x = 0$, the particle velocity perpendicular to the wall must be zero, and the incident wave will be fully reflected by the discontinuity. The reflection coefficient will be 1 because of the hard wall, and the reflected wave is subject to the same impedance as the incoming wave (except for sign). Thus, the pressure amplitudes of the incoming and reflected waves must be equal.

Using superposition, the sound pressure in the tube at negative x coordinates can be written:

$$\underline{p}(x,k) = \hat{p}_+ \left(e^{-jkx} + e^{jkx}\right) \tag{1.71}$$

The particle velocity can similarly be written as:

$$\underline{u}_x(x,k) = \frac{\hat{p}_+}{Z_0}\left(e^{-jkx} - e^{jkx}\right) \tag{1.72}$$

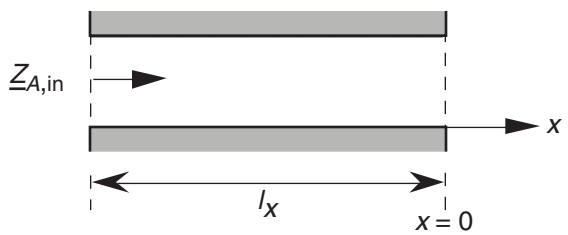


Figure 1.5 A short tube, open at both ends, to be used as an acoustical component.

By definition, volume velocity is the product of particle velocity by cross section area, $\underline{U} = \underline{u}S$. The acoustical impedance seen by the wave for $x < 0$ can then be written as:

$$\underline{Z}_A(x,k) = \frac{\underline{p}(x,k)}{S\underline{u}_x(x,k)} = \frac{Z_0}{S}\frac{e^{-jkx} + e^{jkx}}{e^{-jkx} - e^{jkx}} \tag{1.73}$$

It can also be written as:

$$\underline{Z}_A(x,k) = -\frac{Z_0}{jS}\cot(kx) = \frac{jZ_0}{S}\cot(kx) \tag{1.74}$$

This means that a wave that propagates along a tube, closed at the far end, at $x = -l_x$ sees an acoustical impedance, that is:

$$\underline{Z}_A(k)\big|_{x=-l_x} = -j\frac{Z_0}{S}\cot(kl_x) \tag{1.75}$$

One notes that the acoustical impedance is *fully reactive* (no real part) so the sound pressure and particle velocity are 90° out-of-phase. Depending on the frequency of the sound and on the length of the tube l_x, the impedance will be either positive or negative.

We say that the tube is resonant for frequencies where its acoustical input impedance is zero. Resonance requires $\cot(kl_x) = 0$, which gives the conditions for resonance as $l_x = \lambda/4$, $3\lambda/4$, and so forth.

For the tube entrance to act as a discrete acoustical component, the tube length l_x must be short. A commonly used limit is $l_x < \lambda/8$—that is, $kl_x \ll 1$—that corresponds to a tube that is short compared to the wavelength. We can then write:

$$\underline{Z}_A(\omega)\big|_{x=-l_x} \approx \frac{1}{j\omega C_A} \tag{1.76}$$

where C_A, the *acoustical compliance* (capacitance), is:

$$C_A = \frac{V}{\rho_0 c_0^2} \tag{1.77}$$

when $kl_x \ll 1$, $kb \ll 1$, and where $V = Sl_x$.

As long as the tube diameter and tube length are small compared to the wavelength of sound, it usually does not matter what the exact shape is; it is the confined volume V that is important. This property of C_A is used when designing loudspeaker boxes because the box can be given many shapes, as long as its dimensions are small compared to the maximum wavelength of sound.

Acoustical Mass Components

Again, consider the tube discussed previously and shown in Figure 1.5. Assume an incoming wave that can be formulated as in Equation 1.70.

Now assume the far end, at $x = 0$, to be open. The boundary condition will now be that there is pressure release; that is, the pressure at the far end is $p = 0$. This is not strictly correct since there is some reaction from the air outside the tube, but this is neglected here since we have assumed that $kb \ll 1$.

The reflection coefficient will, in this case, be −1, which is a result of a phase change in the reflected wave by 180°.

The sound pressure in the resulting sound field in the tube can then, in analogy to Equation 1.71, be written as:

$$\underline{p}_x(x,k) = \hat{p}_+\left(e^{-jkx} - e^{jkx}\right) \tag{1.78}$$

Correspondingly, the particle velocity can be written:

$$\underline{u}_x(x,k) = \frac{\hat{p}_+}{Z_0}\left(e^{-jkx} + e^{jkx}\right) \tag{1.79}$$

The acoustical impedance seen by the incoming wave at the tube entrance will then be:

$$\underline{Z}_A(x,k) = \frac{\underline{p}(x,k)}{S\underline{u}_x(x,k)} = \frac{Z_0}{S}\frac{e^{-jkx} - e^{jkx}}{e^{-jkx} + e^{jkx}} \tag{1.80}$$

In turn, it can be rewritten as:

$$\underline{Z}_A(x,k) = -\frac{jZ_0}{S}\tan(kx) \tag{1.81}$$

The acoustical input impedance Z_A for a narrow tube of length l_x can thus be written:

$$\underline{Z}_A(\omega)\big|_{x=-l_x} = \frac{jZ_0}{S}\tan(kl_x) \tag{1.82}$$

As previously stated, we say that the tube is resonant for frequencies where its acoustical input impedance is zero. The impedance is zero when $\tan(kl_x) = 0$; that is, when the tube length is such that $l_x = n\lambda/2$ where $n = 1, 2, 3 \ldots$

Equation 1.82 can be simplified for the special case of narrow and short tubes. Using a series expansion of $\tan(kl_x)$, we can write the acoustical input impedance Z_A as:

$$\underline{Z}_A(\omega)\big|_{x=-l_x} \approx \frac{jZ_0kl_x}{S} = j\omega\frac{\rho_0 l_x}{S} = j\omega\frac{\rho_0 l_x S}{S^2} \tag{1.83}$$

Since the magnitude of the impedance increases proportionally to frequency, and since the impedance has a positive reactance, it is called an acoustical mass impedance or acoustical inductance. The acoustical mass impedance M_A is:

$$M_A = \frac{\rho_0 l_x}{S} \tag{1.84}$$

So the acoustical input impedance at the entrance of the tube is:

$$\underline{Z}_A(\omega)\big|_{x=-l_x} \approx j\omega M_A \tag{1.85}$$

when $kl_x \ll 1$ and $kb \ll 1$.

Since the air outside the far end of the tube is moved by the sound field, there must be some energy radiated. The energy radiation becomes smaller as the tube diameter is reduced. The air at the far end also behaves as an extra added mass, virtually extending the tube slightly. Both of these effects can be taken into account, approximately, by including the radiation impedance at the end of the tube in the calculation instead of putting the value of the impedance at the tube far end to zero when calculating the reflection coefficient. The acoustical mass component is usually not as *ideal* as the acoustical compliance component.

The extra length is handled by using an *end correction* when calculating the effective length of the tube. The end correction is primarily important when dealing with short tubes such as small holes in thin plates and sheets. One can show that the end correction for a circular opening, having a radius b, is approximately $0.78b$ long, resulting in an added acoustical mass of:

$$M_{A,\text{end corr}} \approx \frac{\rho_0}{4b} \tag{1.86}$$

If the circular tube is a hole through a plate, the acoustical mass of the tube (open at both ends) can be written as approximately:

$$M_{A,\text{end corr}} \approx \frac{\rho_0 l_x}{S} + \frac{\rho_0}{2b} \tag{1.87}$$

As in the case of the acoustical compliance, it is primarily the volume of the enclosed air that is important. The exact shape of the tube is unimportant, as long as its maximum cross-sectional dimensions are much smaller than a wavelength.

Acoustical Circuits

If one attaches two acoustical impedance elements to one another in a series as shown in Figure 1.6, the volume velocity $\underline{U}$ in both elements will be the same, and the total pressure over the added components will be $\underline{p} = \underline{p}_1 + \underline{p}_2$.

The total impedance $\underline{Z}_A$, obtained when series coupling acoustical impedance components, is written:

$$\underline{Z}_A = \sum_i \underline{Z}_{Ai} \tag{1.88}$$

When attaching two acoustical impedance components in parallel, as in Figure 1.7, the resulting impedance will be:

$$\frac{1}{\underline{Z}_A} = \sum_i \frac{1}{\underline{Z}_{Ai}} \tag{1.89}$$

The pressure drop will be the same over each element, but the volume velocity in each component will depend on the impedances of both components. It is also possible to introduce acoustic transformers and sources, for example, (see Reference 1.1).

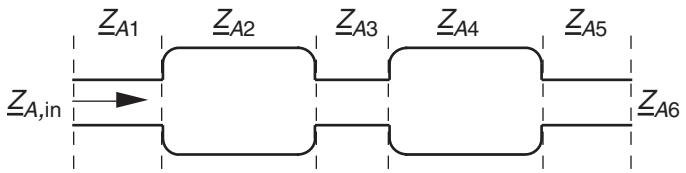


Figure 1.6 Some acoustical impedance elements in series.

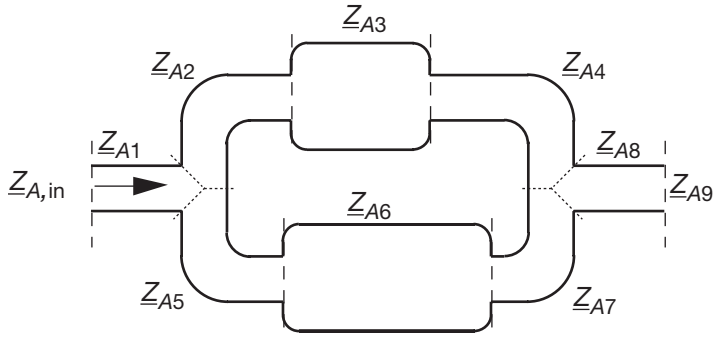


Figure 1.7 Acoustical impedance elements in parallel.

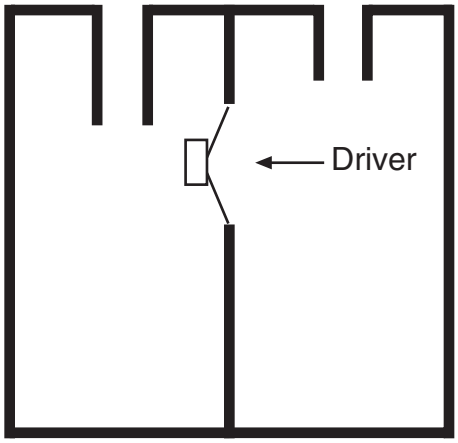


Figure 1.8 A bandpass loudspeaker box design often used for sub-bass loudspeakers and that has four acoustical impedance elements to obtain the desired acoustical filter characteristics.

Figure 1.8 shows a loudspeaker box design often used for sub-bass loudspeakers. It uses two acoustically resonant circuits on either side of the loudspeaker driver to improve the sound radiation from the driver. The calculation of the resonance frequencies uses the previously discussed compliance and mass components.

1.10 SOUND PROPAGATION IN INHOMOGENEOUS MEDIA

In many practical cases sound propagation in inhomogeneous media is characterized by varying temperature, density, and composition. Such problems are found when dealing with sound propagation outdoors and under water. Even sound propagation over long distances in rooms can be affected by air currents and temperature variations.

Outdoors most of the variation will be found close to the ground. As long as the air is at rest (no wind or turbulence) and the temperature does not vary with height, high-frequency sound will propagate linearly, like rays of light. When there are wind or temperature variations, the ray paths will become nonlinear. For some types of weather (inversion), the sound rays can be effectively reflected by a layer of air having a positive temperature gradient.

Temperature Variation

As shown by Equation 1.17, the speed of sound will vary with temperature. Usually the air temperature will vary with height but not along the ground. Two temperature profiles are shown in Figure 1.9.

Normally, the temperature of air at rest decreases with height as shown in Figure 1.9b. One calls the case of negative temperature gradient of −0.01°C/m the *normal temperature profile*. In the case of negative temperature gradients, the speed of sound also decreases with height, resulting in ray paths that bend upward. This is typical for sound propagation outdoors over grassy land on sunny summer afternoons, resulting in *sound shadow*.

When instead, temperature increases with height, the speed of sound will also increase with height, and the ray paths will be bent downward as indicated in Figure 1.9a. This case is called *inversion* and is typical of clear winter nights and mornings.

Sound shadow and the acoustic effects of inversion can be important for the resulting sound level at a far distance from the sound source. When measuring the attenuation of sound at low frequencies by distance outdoors, one typically finds that measurement results for the distance law show a level decrease of approximately −7 dB per distance doubling, rather than the theoretical decrease of −6 dB per distance doubling for free propagation. This difference is due to turbulence. In practice, there will also be additional interference due to ground reflection and scattering by various objects.

Wind

The (lateral) wind velocity u will usually vary with height due to air viscosity, ground roughness, and vegetation. Typically, the velocity varies with height z over the ground as approximately $u(z) \sim ln(z)$.

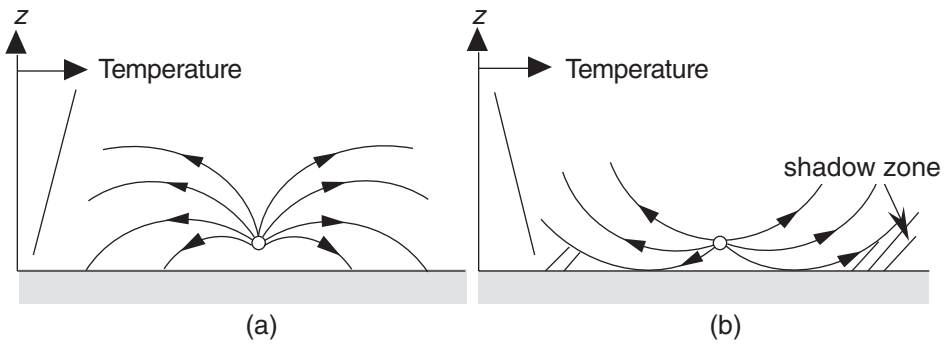


Figure 1.9 Sound propagation paths in the presence of temperature gradients above ground.

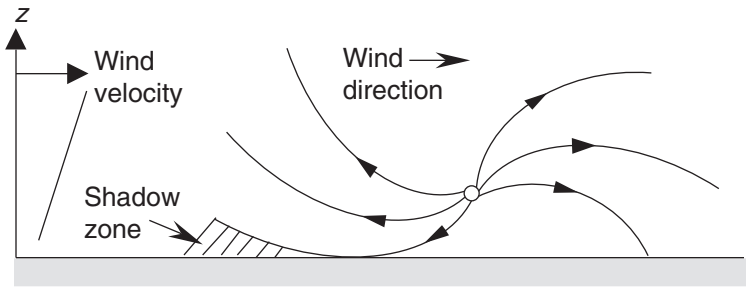


Figure 1.10 Influence of wind on the sound propagation paths.

In the case of tailwind, the sound ray paths will bend upward, whereas, in the case of headwind, they will bend downward as shown in Figure 1.10. The influence of wind and temperature gradients must be added when calculating the ray paths. Note that shadow zones also can occur in the case of tailwind, also shown in Figure 1.10.

One must also remember that the wind seldom is homogeneous—wind creates turbulence, resulting in vortices that scatter sound. The sky is *acoustically blue* because of the scattering of sound by such turbulence. In practice, the sound shadow in the case of negative temperature gradient seldom results in a level decrease of more than 20 dB, because of the scattering of sound from higher air layers due to turbulence.

The turbulence also introduces amplitude and frequency modulation to the sound, making it garbled and difficult to understand. It is important to consider this effect in the design of speech and music sound reinforcement systems to be used outdoors over long distance.

Radius of Curvature

One can show that the resulting ray paths, due to wind and temperature gradients, are approximately circular, having a certain radius of curvature. This applies also when both wind and temperature gradients are present at the same time. One can show that the radius of curvature for sound propagating close to the ground is given by:

$$r = \frac{20\sqrt{T}}{\frac{10}{\sqrt{T}}\frac{\partial T}{\partial z} + \frac{\partial u}{\partial z}} \tag{1.90}$$

In this formula, the speed of sound has been approximated by $c \approx 20\sqrt{T}$, and the wind speed has been assumed to be small compared to the speed of sound.

Figure 1.11 shows a high-intensity, high-directivity voice warning system used in a civil-defense application. Temperature and wind gradient effects can be quite noticeable in these applications, such as in the use of similar loudspeaker array technology for outdoor concert applications.



Figure 1.11 A high-directivity voice warning system using an array of toroidal pattern loudspeakers in a civil-defense application. Temperature and wind gradient effects can be noticeable in such applications involving communication outdoors over long distances. (Photo by Mendel Kleiner.)

1.11 DIPOLES AND QUADRUPOLES

A *dipole* is an elementary source that can be thought of as consisting of two point sources, monopoles, at a small distance b from one another (see Figure 1.12), oscillating at the same frequency but 180° out-of-phase. In the figure this property is indicated by plus and minus signs.

The sound pressure at point A at a distance r will be superposition of the pressures from each monopole, since the sources are assumed not to influence each other's radiation. Because of both positive and negative interference, the total sound pressure will vary, not only with distance but also with angle as in Fig. 1.12.

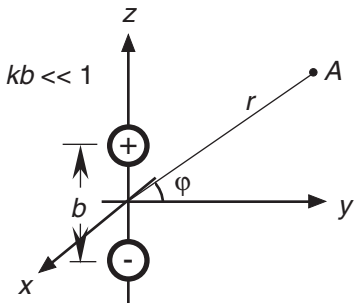


Figure 1.12 The coordinate system used in the analysis of the dipole. The two monopole sources (+/–) are on the *z* axis.

Assume that the sources are much closer than a wavelength and sum up the sound pressures. One then finds that the ratio between the sound pressure at point A from just the *plus* monopole, $\underline{p}_m$, and the sound pressure from the dipole, $\underline{p}_d$, can be written as:

$$\frac{\underline{p}_d}{\underline{p}_m} = \left(jkb + \frac{b}{r} \right) \sin(\varphi) \tag{1.91}$$

The sound power radiated by the dipole will be different from that radiated by a single monopole. Assume that each source acting as a free monopole, by itself, could radiate a power W_m. Calculating the sound power radiated by the dipole, one finds that the ratio between W_d and W_m is:

$$\frac{W_d}{W_m} = \frac{(kb)^2}{3} \tag{1.92}$$

where the *radiated power by a monopole* is:

$$W_m = \frac{\pi \rho_0}{c_0} f^2 \tilde{U}^2 \tag{1.93}$$

As noted, the dipole radiates much less energy than a monopole. The dipole cancellation is an important asset in many cases of noise control engineering. The cancellation is sometimes called *aerodynamic short-circuit* or *aerodynamic cancellation*. Some interesting situations when this phenomenon occurs are:

- The reduction of low frequency response when using a free suspended loudspeaker instead of using the same loudspeaker so that its rear side radiates into a closed box
- The reduced sound radiation from perforated metal sheets compared to solid metal sheets
- The weak sound radiation by small oscillating bodies, such as oscillating machine parts and rods
- Some cases of active noise control by interference

Quadrupoles are somewhat more complicated elementary sources than dipoles. One type of quadrupole is shown in Figure 1.13. This particular quadrupole consists of four monopoles, two in-phase and two out-of-phase at distances b_x and b_y in the $z = 0$ plane.

As in the case of the dipole, one can show that the effective sound power radiation for a quadrupole, W_q, is much smaller than for either of the free sources separately, W_m. One can show that the ratio W_q/W_m becomes:

$$\frac{W_q}{W_m} = \frac{(kb_x)^2 (kb_y)^2}{15} \tag{1.94}$$

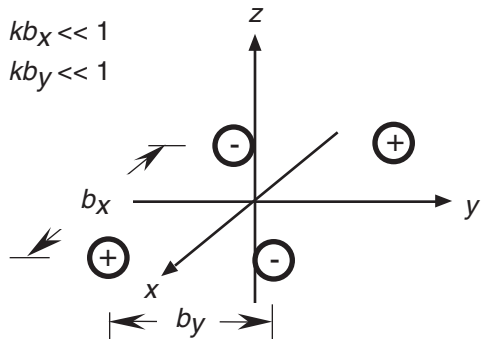


Figure 1.13 An example of a possible quadrupole configuration; all of the monopole sources are in the $z = 0$ plane.

The sound power radiation for a quadrupole is consequently a factor $(kb)^2/5$ less than that of the dipole studied earlier. The radiation pattern of the quadrupole will also be more complicated. Examples of quadrupole-type sound radiation can be found in the radiation of sound from plates and membranes, as well as in the sound generation mechanisms of turbulent flow.

1.12 PROBLEMS

(Note: Related problems can be found in Chapter 2)

1.1 A spherical, sinusoidal wave propagates away from a point source in air. The frequency is 1 kHz. The acoustic pressure amplitude at 1 m from the point source is 0.1 Pa. For air $\rho_0 = 1.20$ kg/m^3, $c_0 = 343$ m/s.

Tasks:

a. Plot the acoustic pressure amplitude and the particle velocity amplitude as functions of the distance from the source.
b. Plot the phase angle between acoustic pressure and particle velocity as a function of the distance from the source.
c. Graph the functions as functions of kr—that is, the product between wave number and distance.

1.2 The sound pressure in front of a wall varies with distance from the wall due to interference.

Task: Calculate the sound pressure increase, relative to the free field pressure, that one obtains when measuring the sound pressure level at a distance of 0.1 m in front of a flat, hard wall. The sound wave is perpendicularly incident to the wall.

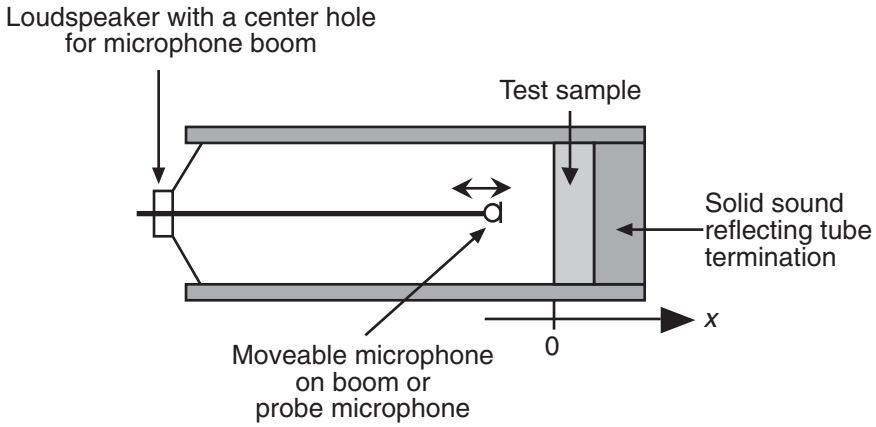
1.3 The surface of a sphere, whose radius is 0.05 m, vibrates radially, and its volume velocity is frequency-independent. The amplitude of the surface vibration is 0.001 m at 100 Hz.

Task: Calculate the radiated sound power as a function of frequency.

1.4 A Kundt's tube (a.k.a., plane wave tube) can be used to measure the complex reflection coefficient of a material at normal incidence. The standing wave ratio γ is defined as:

$$\gamma = \frac{\tilde{p}_{\max}}{\tilde{p}_{\min}}$$

Due to exterior noise, it is usually difficult to determine γ accurately when $\tilde{p}_{\min}$ is small. Consequently, it is also difficult to determine the sound-absorption coefficient α. By measuring the distance from the surface of the test sample facing the tube to the point where the first sound pressure level minimum is obtained, it is possible to obtain data for calculating the phase angle of the reflection coefficient.

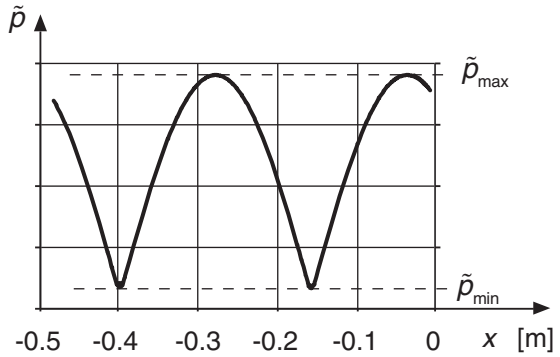


P1.4a The principle of the Kundt's tube equipment for measurement of the sound pressure reflection coefficient.

Tasks:

a. Derive an expression for the sound-absorption coefficient α of the material as a function of the standing wave ratio γ. Calculate the sound-absorption coefficient for γ = 10 ± 1.

b. Derive an expression for how the phase angle of the reflection coefficient can be calculated.



P1.4b Example of the measured rms sound pressure of a standing wave in a Kundt's tube. Note the locations of the peaks and dips of the sound pressure relative to $x = 0$.

Audio Signals 2

2.1 SPECTRUM AND TIME HISTORY

We need to study the properties of sound both from the technical and from the subjective viewpoints. Hearing acts as a complicated time-frequency analyzer that can also analyze patterns in sound and use cognition to determine the source identity, among other things. Technical analysis at this time cannot imitate most of these advanced properties. While time-frequency analysis, such as short-time Fourier analysis and wavelet analysis, can simulate some of the analysis features of hearing (see Chapter 17), we, in practice, primarily study the spectra of signals—that is, their power content as a function of frequency—using Fourier analysis. The signals, music, and noise are continuous or quasiperiodic from an analysis viewpoint (their spectra do not change much over the time of the analysis window). For transient signals or impulse responses, we may use the already mentioned short-time Fourier analysis with good results. References 2.1–2.4 are good introductions to the analysis and measurement of various sounds.

Measurement of audio system characteristics and qualities is discussed in Chapter 17. Chapters 3, 5, and 6 also discuss measurement aspects of room acoustics, while Chapter 10 discusses measurement in building acoustics.

2.2 SIGNALS AND THE *jω*-METHOD

The acoustic waves that we are interested in from the viewpoint of sound, such as noise, voice, and music, can be regarded as signals in time and space. We are accustomed to writing waves in this form:

$$p(x,t) = A\cos(2\pi ft - 2\pi x/\lambda) + B\sin(2\pi ft - 2\pi x/\lambda) \tag{2.1}$$

describing the shape of the signal at points in time and space (see Figure 2.1).

The signals picked up by the microphones apply to the particular measurement point in space and give a signal that is a function of time $s(t)$. A practical audio signal is usually finite and quasiperiodic. Assuming the period of the signal to be T_0, the signals can be represented by a sum (possibly infinite) of simple signals such as sine and/or cosine signals:

$$\begin{aligned} s(t) &= \sum_{n=0}^{\infty} A_n \cos(n\omega_0 t) + \sum_{n=0}^{\infty} B_n \sin(n\omega_0 t) \\ &= A_0 + \sum_{n=1}^{\infty} A_n \cos(n\omega_0 t) + \sum_{n=1}^{\infty} B_n \sin(n\omega_0 t) \end{aligned} \tag{2.2}$$

where

$$\omega_0 = \frac{2\pi}{T_0} \tag{2.3}$$

Since there is no steady state offset to be accounted for in audio, the coefficient $A_0 = 0$. The coefficients A_n and B_n are obtained by a form of correlation:

$$A_n = \frac{2}{T_0} \int_{-T_0/2}^{T_0/2} s(t) \cos(n\omega_0 t)\, dt \tag{2.4}$$

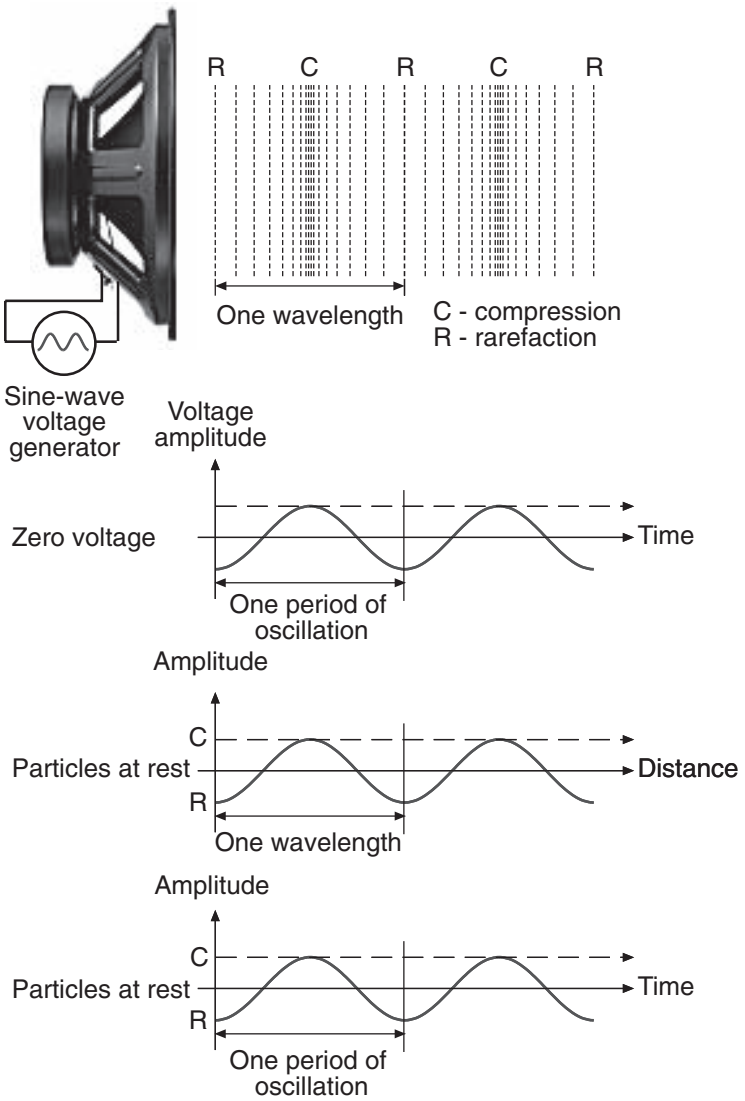


Figure 2.1 A wave represents a signal in time and space.

$$B_n = \frac{2}{T_0}\int_{-T_0/2}^{T_0/2} s(t)\sin(n\omega_0 t)\,dt \tag{2.5}$$

It is more convenient to use only cosines and a phase angle than to use the sum of sine and cosine because of the simpler mathematical handling:

$$\begin{aligned} s(t) &= \sum_{n=1}^{\infty} C_n \cos(n\omega_0 t - \phi_n) \\ C_n &= \sqrt{A_n^2 + B_n^2} \\ \phi_n &= \arctan\left(\frac{B_n}{A_n}\right) \end{aligned} \tag{2.6}$$

Using Euler's formula:

$$e^{j\omega t} = \cos(\omega t) + j\sin(\omega t) \tag{2.7}$$

we can make use of the simpler math afforded us. We use complex notation in the mathematical handling of the wave equation and its solutions but consider the real part of the exponential the actual real-world signal.

The coefficients C_n and angles φ_n are called the frequency domain representation of the time signal $s(t)$ or its spectrum. Phase is often of secondary interest in signal analysis, but the phase information can be used to analyze the delay of the signal in a transmission path. The reason for neglecting phase information is that, from the viewpoint of noise abatement, phase is of little interest because it is the energy exposure of the ear that is important for hearing loss.

In audio signal characterization and listening, phase is important. A transient sound and a noise sound may have the same energy but will appear entirely different to our hearing. It is sufficient to time mirror (reverse) a musical signal or speech and listen to it, to be convinced that phase is important; both the source and the reversed source signals have the same spectra.

The *jω-method* (pronounced the j-omega method), sometimes called the complex method, is the practical application of Fourier transform theory to the analysis of signals and linear systems. Since audio signals can be expanded into complex exponential series, each series term (frequency), can be analyzed separately as to how a transmission system affects it. Superposition (addition of the time history) of all relevant frequency terms with amplitude and phase then gives the time history of the output signal. Because the time function is always of the form:

$$e^{j\omega t} \tag{2.8}$$

It is not necessary to include the function in each equation. Instead, in this book, we use a shorthand notation and underline the variable to mark that it is has the time dependence given by Equation 2.8. We also use the underline to mark out other complex quantities and variables.

The ratio between two complex variables, such as pressure and particle velocity, in general will be complex as well but will not have the time component:

$$\underline{Z}_n = \frac{\underline{p}_n}{\underline{u}_n} = \frac{(A_n + jB_n)e^{j\omega t}}{(C_n + jD_n)e^{j\omega t}} = \frac{A_n + jB_n}{C_n + jD_n} \tag{2.9}$$

One of the most useful properties of the *jω*-method is that it represents differentation with respect to time by multiplication with *jω* and integration with respect to time by division with *jω*. We have already used this property several times in Chapter 1.

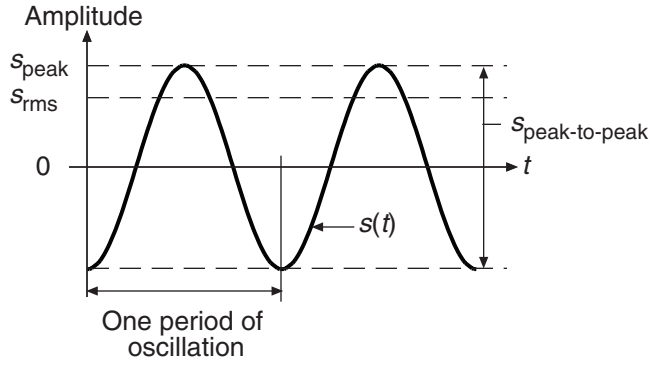


Figure 2.2 Characteristics of a sine-wave signal.

2.3 SINE-WAVE SIGNALS

Signals that can be described by just one frequency are called sine waves. Some terms used to characterize sine-wave signals are shown in Figure 2.2. A sinusoidal signal having a *peak value* of A has an *mean square value* (effective value) of $A^2/2$ and *root-mean-square* (*rms*) *value* of $A/\sqrt{2}$. The effective value is defined in Equation 2.21. The rms value of a sine-wave is approximately 71% of the peak value. The magnitude of the peak-to-rms ratio of a signal, sometimes called the *crest factor*, is an important quantity in the characterization of signals.

In this book, we write the rms value of variables using a tilde sign such as $\tilde{p}$ for rms sound pressure. The peak value is written $\hat{p}$.

2.4 NOISE

The term "noise" in colloquial language stands for a loud, unpleasant, undesired, or disturbing sound. In acoustics, the use of the term is more limited. While many use the term to describe the sound along a freeway, airport, or from a neighbor, the use of the word should really be restricted to signals (such as sound and voltage) that do not carry meaning. Thermal noise, the result of random physical processes, is such a signal. Figure 2.3 shows an example of such a noise signal. The peak value is nearly infinite but seldom occurs. In classical music, the crest factor is typically less than 10 and in speech less than 4.

The autocorrelation of a signal can be used to describe its quality of randomness. The normalized *autocorrelation function* for a signal $s(t)$ is defined by:

$$R_{ss}(\tau) = \frac{\int_{-\infty}^{\infty} s(t+\tau)s(t)\,dt}{\int_{-\infty}^{\infty} s(t)s(t)\,dt} \tag{2.10}$$

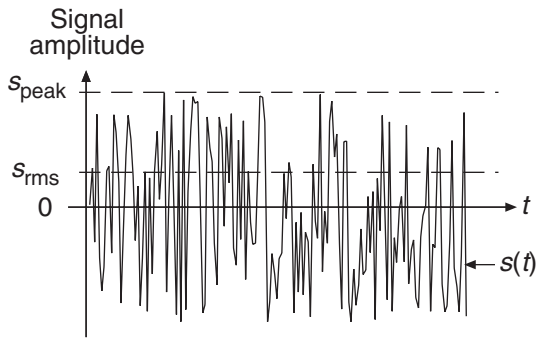


Figure 2.3 A noise signal.

The normalized autocorrelation $R(\tau)$ of thermal noise, often called *white noise*, is zero except for $\tau = 0$ (i.e., when the signal coincides identically with itself when $R(0) = 1$). Another characteristic of a white noise signal is that it has the same *spectral density* at any frequency. Spectral density is the power per Hz of the signal. Many other signals are characterized by an autocorrelation function like the one of white noise, for example, pink and red noise. These noise signals have the following power spectra G(ω):

$$\begin{aligned} G_{\text{white}}(\omega) &\propto 1 \\ G_{\text{pink}}(\omega) &\propto \frac{1}{\omega} \\ G_{\text{red}}(\omega) &\propto \frac{1}{\omega^2} \end{aligned} \tag{2.11}$$

The power spectrum describes the power for unit frequency. Long time segments of signals, such as most music and speech, also have properties that result in their normalized autocorrelation being almost zero for $\tau \neq 0$.

2.5 THE LEVEL CONCEPT

The psychophysical sensitivity of hearing is such that each increase of intensity by a certain ratio, over some intensity limit, is perceived approximately equally strong subjectively. This is a good reason to use logarithmic ratios to describe the subjective response.

Levels can be used both to describe differences between quantities and magnitudes of quantities. It is practical to use the internationally agreed reference values, when applicable, in calculating levels.

The concept of *level*, in acoustics, implies that one uses the logarithm between two values of a power related sound field property to describe their difference. Power-related quantities are sound intensity, sound power, energy, and energy density. Since sound pressure and particle velocity are related to sound intensity (and voltage and electric current to electric power), the use of levels can be applied to them as well. As an example, a power density level is written:

$$L(\omega) = 10 \log\left(\frac{G_1(\omega)}{G_0(\omega)}\right) \tag{2.12}$$

where $G_0(\omega)$ is a reference value, for example, 1 W/Hz. As another example, the difference between two power density levels $G_2(\omega)$ and $G_1(\omega)$ is written:

$$\Delta L(\omega) = 10 \log\left(\frac{G_2(\omega)}{G_1(\omega)}\right) \tag{2.13}$$

Note also that it is practical to extend the use of levels to impedance and its inverse mobility as well—for example, to more easily describe the impedance variation over the frequency range. Sound field impedance is often referenced to the specific impedance of air, mechanical impedance to 1 Ns/m.

Levels are denoted by L and an index, such as L_p, for sound pressure level. Levels are expressed in units of decibel [dB] irrespective of the quantities used. Sound power levels are, at times, expressed in units of Bel. (One Bel is, of course, ten decibel.) This should be avoided.

As an application of the level idea, consider the use of level to describe the properties of attenuation of a plane wave by air or by an electric filter.

2.6 FILTERS AND FREQUENCY BANDS

In audio technology, as in acoustics, the primary frequency range of interest is usually that of normal hearing (20 Hz–20 kHz). In many cases, we need to know the property of the sound in narrower frequency ranges—for example, when analyzing noise or speech.

To remove uninteresting sound components, we use *analog* or *digital* filters. While analog filters could, in principle, be acoustical, we usually use electronic filters for analog filtering since this is more practical. Digital filtering can be done both by direct simulation of analog filters and by digital Fourier transform techniques. Digital filters have the advantage of higher stability but have the disadvantage of signal artifacts being created in the analog-to-digital and digital-to-analog conversion processes. Most digital filters are also not true *real-time filters*, in contrast to analog filters. This has important implications in many signal processing applications.

The *frequency response* of a filter is generally considered to be the output level minus the input level over the frequency range. The frequency response is obtained from the *transfer function* as $20 \log(|H(\omega)|)$:

$$\textit{Frequency response} = 20 \log(|\underline{H}(f)|) + C \tag{2.14}$$

where C is a constant. In audio and acoustics, the constant is usually chosen so that the frequency response is zero dB at 1 kHz.

The transfer function operates on the input signal $\underline{S}_{\text{in}}(\omega)$ so that $\underline{S}_{\text{in}}(\omega)$ is transformed by the filter's transfer function $H(\omega)$ so that the output signal $\underline{S}_{\text{out}}(\omega)$ is:

$$\underline{S}_{\text{out}}(\omega) = \underline{H}(\omega)\underline{S}_{\text{in}}(\omega) \tag{2.15}$$

Filters are typically characterized by *pass bands* and *stop bands*, as shown in Figure 2.4. For each band, the filter will have a certain *attenuation* as a function of frequency. Typically, one does not want any attenuation in the pass band. The attenuation in the pass band is, however, usually characterized by *ripple*—small variations in attenuation. The pass band and the stop bands are characterized by their

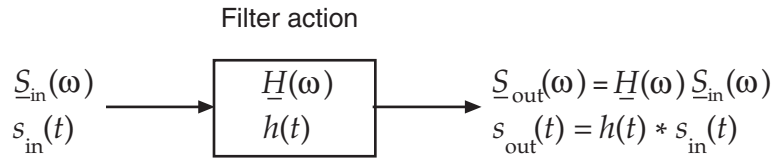


Figure 2.4 The filter operates on the signal $s_{in}(t)$ having a spectrum $S_{in}(\omega)$. The resulting signal spectrum is obtained by multiplication—at each frequency—of the input spectrum and the complex transfer function. The output time function is obtained by convolution of the input waveform by the filter's impulse response (see Equations 2.16 & 2.17).

start and *stop frequencies*, generally called *cutoff frequencies*, usually taken to be at the *−3 dB points of the filter* (power) *response* relative to the pass band average, or the filter's center frequency. Simple filters are *high-* and *low-pass filters*, so called because of their frequency rejection characteristics.

Practical filters do not have the sharp band edges shown in Figure 2.5. The frequency response of a practical analog commercial bandpass filter is illustrated in Figure 2.6. We note that the band edges have a finite *slope* (dB/Hz). Because of the finite slope, the filter will also let some signal through outside the pass band. This can have undesired influence on the accurate measurement of noise spectra, as an example.

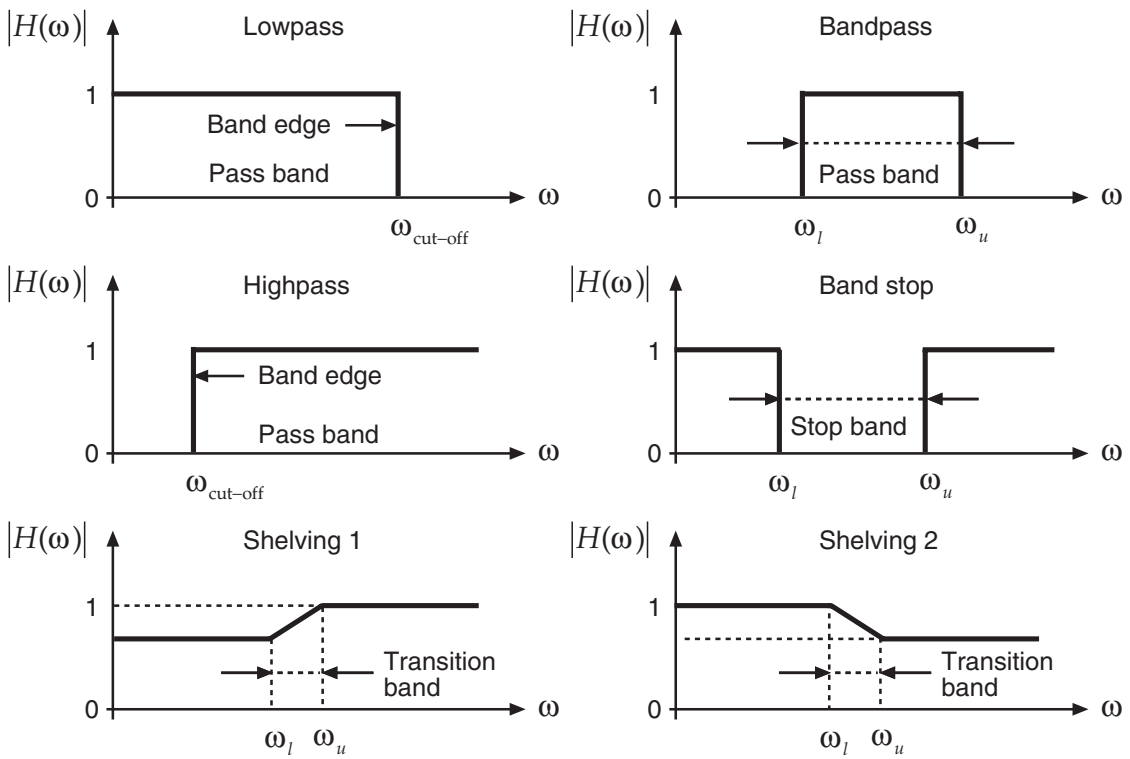


Figure 2.5 Some common filter types and their nominal frequency responses.

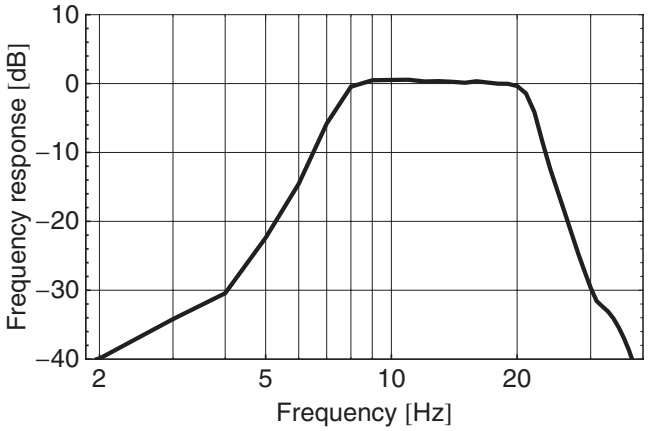


Figure 2.6 An example of the frequency response of a commercial analog octave band filter centered around 12.5 Hz.

It is also important to note that a filter's transfer function is related to its impulse response $h(t)$ by the Fourier transform:

$$\underline{H}(\omega) = \int_{-\infty}^{\infty} h(t) e^{-j\omega t}\, dt \tag{2.16}$$

And its inverse:

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \underline{H}(\omega) e^{j\omega t}\, d\omega \tag{2.17}$$

Figure 2.7 shows the impulse response of the filter that has the frequency response illustrated in Figure 2.6.

The analysis of the frequency content of signals is called *spectral analysis*. One can subdivide the many forms of spectral analysis into *narrow-* and *wide-band* spectral analysis, into *constant relative bandwidth*, and *constant bandwidth* spectral analysis. In spectral analysis, *bandpass filters* are used. Such filters pass signals with frequencies with a certain frequency band, the *pass band*. Frequencies that are *out-of-band* are rejected. The bandwidth of a bandpass filter is:

$$B = \Delta f = f_u - f_l \tag{2.18}$$

That is, it is the difference between the upper cutoff frequency, f_u, and lower cutoff frequency, f_1. The geometrical mean of the upper and lower cutoff frequencies, the *geometrical mean frequency*, usually called the *center frequency*, is used to denote the filter. The ratio between the bandwidth and the center frequency, f_m, is always constant for any particular type of constant relative bandwidth filter.

Two common types of bandpass filters used in wide-band, relative bandwidth spectral analysis, are *third octave* and *octave* band filters. For octave band filters the following applies:

$$\frac{B}{f_m} = \frac{\Delta f}{f_m} = \frac{f_u - f_l}{f_m} = \sqrt{2} - \frac{1}{\sqrt{2}} \approx 0.71 \tag{2.19}$$

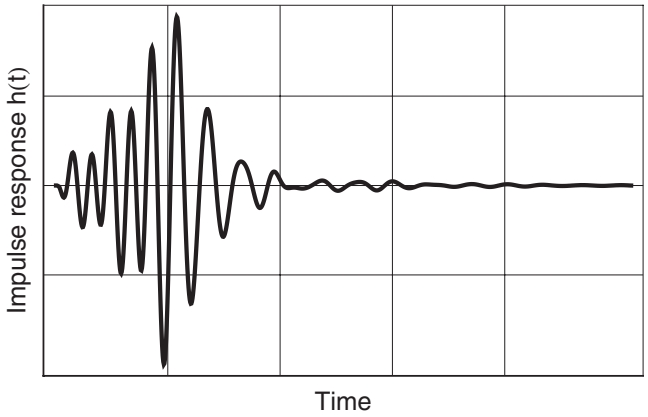


Figure 2.7 The impulse response of the octave band filter discussed in Figure 2.5. Note that low frequencies are more delayed by this filter than high frequencies.

Similarly, for third octave band filters:

$$\frac{B}{f_m} = \frac{\Delta f}{f_m} = \frac{f_u - f_l}{f_m} = \sqrt[6]{2} - \frac{1}{\sqrt[6]{2}} \approx 0.23 \tag{2.20}$$

The center frequencies (in Hz) of octave- and third octave band filters have been internationally standardized as:

$$\leftarrow \ldots 160\ 200\ \mathbf{250}\ 315\ 400\ \mathbf{500}\ 630\ 800\ \mathbf{1000}\ 1250\ 1600 \ldots \rightarrow$$

The numbers in bold style are the internationally agreed center frequencies of octave band filters (see the IEC 651 and ANSI S1.16 standards).

When analyzing sounds for information about tonal components, it is often practical to use filters having *constant bandwidth*. Such narrow-band, constant bandwidth filters for the audio frequency range are characterized by bandwidths as small as 0.3 Hz, in the case of analog filters, and by bandwidths almost arbitrarily small in the case of digital filters.

The level of a band-limited signal $L_{\Delta f}(f)$ having a bandwidth Δf is related to its spectrum level $L(f)$ as:

$$L_{\Delta f}(f) = L(f) + 10\log(\Delta f) \tag{2.21}$$

The constant bandwidth filters will give a constant value when applied to a white noise signal. Since the bandwidths of the octave- and third octave band filters increase in proportion to frequency, the octave- and third octave band levels of white noise will also increase as shown for the octave band levels in Figure 2.8. Pink and red noise signals will have 3 dB/octave and 6 dB/octave lower-level slopes as shown in the figure. In the figure, the three noises were arbitrarily adjusted to have the same level in the 1 kHz octave band.

It should be noted that all real audio signals are low-pass filtered because of the high-frequency attenuation properties of air, microphones, loudspeakers, and amplifiers, among others.

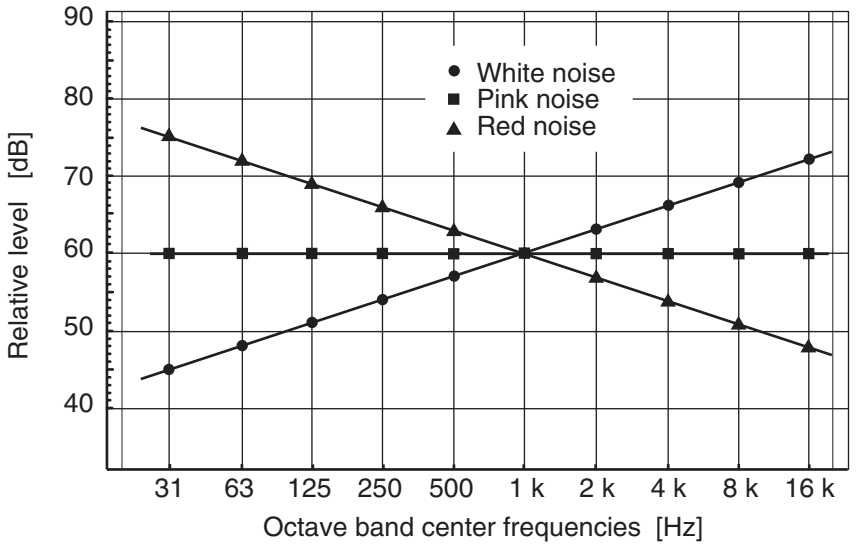


Figure 2.8 The octave band level value characteristics of white, pink, and red noise.

2.7 EFFECTIVE VALUE AND MEASUREMENT UNCERTAINTY

Any physical signal carries energy. The effective value is used to describe the power of a signal. The effective value or mean-square of a signal $\tilde{s}(t)$ is estimated as:

$$\tilde{s}^2 = \frac{1}{T}\int_0^T s^2(t)\,dt \tag{2.22}$$

where T is the integration time; generally a better estimate is obtained the longer the integration time.

A white noise signal has an amplitude that varies at random over time. One can show that the estimate of the rms value of bandpass filtered white noise has a *relative standard deviation* that is:

$$\xi_{ms} \approx \frac{1}{2\sqrt{BT}} \tag{2.23}$$

where B is the bandwidth in Hz of the noise and T is the time of integration for large vales of BT. Over small bandwidths, pink and red noises, as well as other noise-like signals, follow the same rule.

Since the variance determines the *confidence interval* of the estimated value, clearly to make good noise measurements, one must either use wide-band filters or long averaging times to obtain precise values. Figure 2.9 shows the 95% confidence intervals for a band-limited white noise signal as a function of the variable BT.

2.8 REFERENCE LEVELS

Hearing senses sound pressures in a range of approximately $.5{\cdot}10^{-5}$ Pa to 100 Pa. It is practical to use a *temperature* similar scale to describe the strength and relative strength of sounds since, as mentioned

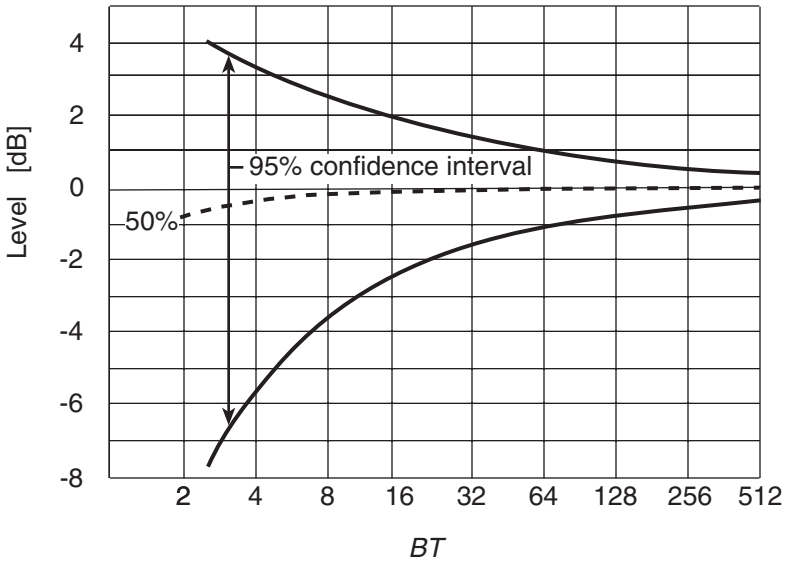


Figure 2.9 The confidence interval for the level of band-limited white noise as a function of the *BT*-product. (After Ref. 2.1)

previously, hearing perceives an increase of intensity by a certain ratio, over some intensity limit, as approximately equally strong subjectively over its sensitivity range.

For any scale to be practical, it needs a reference point or value. In acoustics, the level reference values for intensity and sound pressure are chosen so that related levels have the same size and the number 0 dB corresponds closely to the hearing threshold. When there is any risk of misunderstanding, one should always give the reference value. (Note, for example, that the reference values for sound pressure are different for sound in air and sound in water.)

Some common reference values for airborne sound are:

- Sound pressure (air) $p_{ref} = 2 \cdot 10^{-5}$ Pa
- Particle velocity (air) $u_{ref} = 10^{-9}$ m/s (sometimes $u_{ref} = 5 \cdot 10^{-8}$ m/s)
- Intensity $I_{ref} = 10^{-12}$ W/m^2
- Power $W_{ref} = 10^{-12}$ W

These values give the respective levels:

Sound pressure level

$$L_p = 20 \log\left(\frac{\tilde{p}}{2 \cdot 10^{-5}}\right) \quad \text{dB} \tag{2.24}$$

Particle velocity level

$$L_u = 20 \log\left(\frac{\tilde{u}}{10^{-9}}\right) \quad \text{dB} \tag{2.25}$$

Sound intensity level

$$L_I = 10\log\left(\frac{I}{10^{-12}}\right) \quad \text{dB} \tag{2.26}$$

Sound power level

$$L_W = 10\log\left(\frac{W}{10^{-12}}\right) \quad \text{dB} \tag{2.27}$$

With these reference values, the various levels will typically have values in the 0–140 dB range for airborne sound of practical interest. Note particularly that the ratios, in the case of sound pressure and particle velocity, are based on the *rms* (*root-mean-square*) *values of the quantities*.

In American usage, sound pressure level (L_p) is often denoted SPL, L_I is SIL, and L_W PWL. This convention is not used in this book.

2.9 ADDITION OF LEVEL CONTRIBUTIONS

It is important to note that one cannot just add the respective levels to obtain the resulting level of many sound sources acting at the same time. One must return to the *true* values of the respective sound field quantities and add those according to the rules applicable to each sound field quantity.

For sound power, one adds the power values before taking the logarithm. For sound intensity one adds the intensity values, but since intensity is a vectorial quantity, one must be careful with signs and directions.

For sound pressures and particle velocities, things are somewhat more complicated, as illustrated by the following example. It is practical to separate between two different cases, those of *correlated* and *uncorrelated* sounds. If the sounds are correlated—as, for example, the sound from the front and the back of a vibrating loudspeaker diaphragm, one must add the sound pressures directly before obtaining the root-mean-square value and dividing by the reference value.

Uncorrelated sounds are sounds for which the time weighted integral of the product of the sound pressures tends toward zero for sufficiently long integration times. Examples of uncorrelated sounds are the noises from two jets, two automobiles, and two voices. The sounds in different frequency bands of noise, music, and speech also tend to be uncorrelated and can be treated as such when calculating a sum over frequency bands.

For uncorrelated sounds, a simpler method is used. One can add the *effective value* (defined by Equation 2.22), of the sound pressure of each contributing sound source. Since the root mean square value, the *rms* value, is the square root of the effective value, we have:

$$\tilde{p}^2_{\text{tot}} = \tilde{p}^2_1 + \tilde{p}^2_2 = p^2_{\text{ref}}\left(10^{L_{p1}/10} + 10^{L_{p2}/10}\right) \tag{2.28}$$

That is:

$$L_{p,\text{tot}} = 10\log\left(10^{L_{p1}/10} + 10^{L_{p2}/10}\right) \tag{2.29}$$

The resulting level, due to more than two sources, is obtained by simply repeating this procedure or by directly adding up all the effective values before taking the logarithm.

2.10 WEIGHTED SOUND PRESSURE LEVELS

It is often necessary to characterize sound in a way similar to how it is subjectively perceived. It is tradition to analyze the strength of signals by using so-called *A*, *B*, and *C filters* in the electronic filtering process before obtaining the effective values of the signals.

These filters are characterized by frequency dependence that is somewhat similar to that of human hearing in certain sound pressure ranges (see Section 3.3). Figure 3.13 shows the frequency response characteristics of these filters, which are defined by the IEC651 standard.

The levels measured using the filters are called *weighted levels*. The sound-pressure level measured using an *A filter* is expressed in units of dB and is usually called *sound level* and denoted L_{pA}. However, many still write the A-weighted sound level as L_p in units of dBA for extra clarity (as sometimes in this book). Corresponding indices are used when measuring using the B and C filters. Sometimes one writes $L_{p\text{lin}}$ to emphasize that no filter has been used in a measurement—the measurement value is said to be *linear* or *unweighted*.

2.11 EQUIVALENT LEVEL

Sometimes neither the sound level nor the mean sound level is sufficient to describe a sound having a time varying sound level. One might want to describe how the sound level changes over long time intervals. It is then possible to use the statistical distribution of sound level over a certain time interval—for example, 10:00 p.m. to 6:00 a.m. for night time values—and the averaged integrated energy value of the sound level over a certain time, the *equivalent level.*

The *energy-equivalent A-weighted sound pressure level*, in decibels, often referred to as simply the equivalent level $L_{Aeq,T}$, is defined as:

$$L_{Aeq,T} = 10\log\left(\frac{1}{T}\int_0^T \frac{\tilde{p}^2}{p_{ref}^2}dt\right) = 10\log\left(\frac{1}{T}\int_0^T 10^{L_p/10}\,dt\right) \tag{2.30}$$

where T is the measurement period (usually in hours), $p(t)$ is the short time averaged A-weighted rms sound pressure and $L_p\,(t)$ is the short time averaged A-weighted sound level.

Typical integration times are 8 and 24 hours for many types of environmental noise measurements. The equivalent level can also be estimated approximately from a statistical distribution of the A-weighted sound pressure level:

$$L_{Aeq,T} = 10\log\left(\frac{1}{T}\sum_i t_i \cdot 10^{L_{p,i}/10}\right) \tag{2.31}$$

where t_i is the time that the sound level has fallen between the limits of class i, and $L_{p,i}$ is the A-weighted sound level corresponding to the class-midpoint of class i. The level classes are usually 5 dB wide.

2.12 PROBLEMS

2.1 To calculate the total (unweighted) sound pressure level using *band-limited* data for noise, one usually assumes that the octave band signal components are uncorrelated. A certain spectrum has the following octave band sound pressure level data:

Octave band	63	125	250	500	1 k	2 k	4 k	8k	[Hz]
L_p	50	58	64	59	51	46	41	35	[dB] re 20 μPa

Task: Calculate, assuming the signal components uncorrelated, the linear sound pressure level.

2.2 Consider the octave band spectrum. One can often estimate the A-weighted sound pressure level value from the octave band that has the highest level.

Tasks:

a. Calculate the total A-weighted sound pressure level for the signal spectrum given in Problem 2.2.
b. Which octave bands dominate in calculating the resulting value?

2.3 Three sources of sound are radiating inside a room. The rms value of the sound pressure was measured for each source radiating alone, giving the following set of results:

Source l: $p_{1,\mathrm{rms}} = 0.63$ Pa
Source 2: $p_{2,\mathrm{rms}} = 0.11$ Pa
Source 3: $p_{3,\mathrm{rms}} = 0.20$ Pa

Tasks:

a. Determine the sound pressure level at the measuring point when all three sources are active. Assume that the sources are uncorrelated.
b. Repeat Part a but assume that Source 1 has been modified so that $\tilde{p}^2_{1,rms}$ is reduced to half its value.

2.4

Task: Calculate the equivalent level over eight hours for a noise having the following sound level (L_{pA}) distribution over time: 60 dB for one hour, 80 dB for six hours, and 90 dB for one hour.

2.5 Noise measurements are often done using third octave band filters.

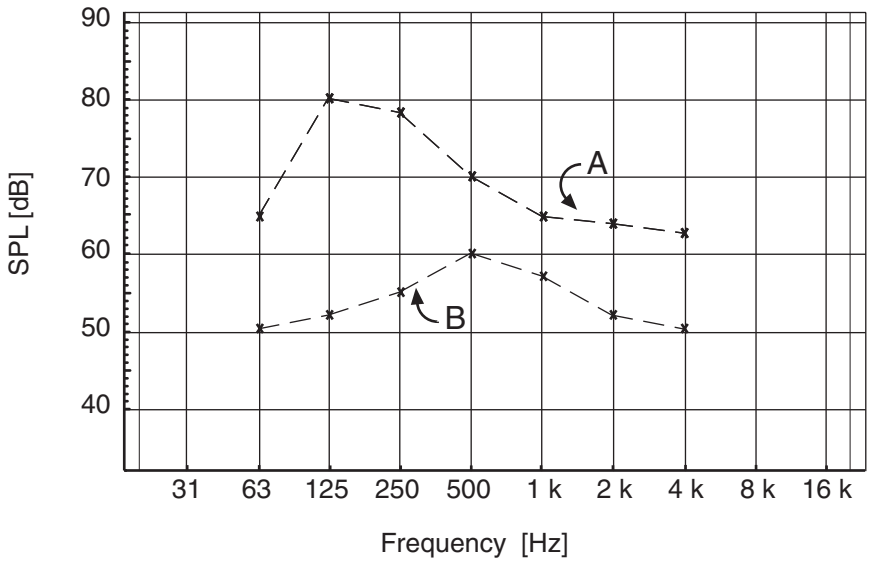
Task: In which third octave frequency bands do you expect a 60 Hz tone and its accompanying overtones to appear?

2.6 The diagram on the next page shows octave band spectra for two noise signals.

Tasks:

Calculate and plot (in the diagram):

a. The linear and total A-weighted sound pressure levels for the signals.
b. The sum of the spectra A and B, assuming that they are uncorrelated. Also calculate the total A-weighted sound pressure level for the sum of spectra A and B.
c. The A-weighted spectrum of the sound pressure level for signal A.



2.7 The graph in Figure 3.29 shows an example of pressure spectrum level (PSL), of speech at 5 cm distance from the lips. The PSL is the sound pressure level measured using an ideal bandpass filter 1 Hz wide.

Task: Calculate, using the graph, the approximate octave- and third octave band levels for speech at a distance of 1 m from the speaker's mouth.

2.8

Task: While measuring a particular noise in frequency bands 100 Hz wide, it is desired that the normalized variance should not exceed 0.3% (± 0.03 dB). Over what time is it necessary to integrate to obtain the effective value with this precision?

2.9 The sound pressure in front of a wall varies with distance from the wall due to interference.

Task: Plot the increase in sound pressure level as a function of frequency and calculate the values at the nominal octave band filter frequencies in the range of 63 Hz to 8 kHz.

2.10 The sphere surface vibrates radially and its volume velocity is frequency-independent. The sphere has a radius of 0.05 m. The amplitude of the surface vibration is 0.001 m at 100 Hz.

Task: Plot the sound power level as a function of frequency at the nominal octave band filter frequencies in the range of 125 Hz to 2 kHz.

2.11 You need to determine the sound pressure levels caused by a sound source outdoors. The sound pressure levels have been measured both with and without the active sound source. However, the values measured for the case when the sound source is active have been contaminated by the background noise.

Task: Calculate the *real* sound pressure level due to the sound source if the noise due to the sound source and the background noise are uncorrelated.

Octave band	125	250	500	1 k	2 k	4 k	[Hz]
$L_{p,\text{background only}}$	34	31	30	29	29	28	[dB] re 20 μPa
$L_{p,\text{with source}}$	37	37	37	38	40	40	[dB] re 20 μPa

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Hearing and Voice 3

3.1 INTRODUCTION

Human hearing is an extremely sensitive mechanism for sensing acoustic signals. As we *listen*, we use a set of subsystems: *ears*, *auditory nerves*, and *brain*. Humans may be able to detect rms sound pressures as low as 5 μPa and to endure pressures as high as 100 Pa, a significant dynamic range for any transducer system. The frequency range extends from 20 Hz to 20 kHz for a *normal hearing* person.

The high frequencies have short wavelengths that allow good directional hearing. The directional hearing properties are further improved by *binaural hearing* and the associated correlation of the instantaneous signals from the two ears. Hearing also acts as an adaptive signal processor that gives us the possibility of detecting signal patterns and weak signals, even in noisy environments.

Although the physiological functions of the ear are well known, little is known about the signal processing properties of the auditory nerve and the brain. Using *psychoacoustic tests*, it is possible to check various hypotheses for the function of these systems. Such experiments and results are discussed later in this chapter.

Figure 3.1 shows a schematic representation of a psychoacoustic experiment. In listening, we perceive internally the effect of the *stimulus* on our ears, the *perception* formed depends on emotional status, prior experience, and so forth. When we need to output our experience, we have to use language—writing or speaking—or some help device such as a pointer or a musical instrument. All of these factors influence the response for any given stimulus. This makes it difficult to obtain exactly the same response from various listeners. Listener response will depend on such matters as the experimental layout, hearing threshold, experience, and personal ability to verbalize what is being perceived. This means that any test involving listening must be performed under well-thought-through, specified, and controlled circumstances to eliminate *bias error*. The test must also involve many listeners to reduce *random error*. In psychoacoustic tests involving spatial discrimination, it is often practical to use the *head-related coordinate system* shown in Figure 3.2.

3.2 THE COMPONENTS OF THE EAR

Functionally, the ear can be regarded as consisting of three parts: (1) *outer ear*, (2) *middle ear*, and (3) *inner ear*. Together these form a system for the conversion of sound pressure oscillations to associated nerve signals in the auditory nerve. A drawing showing the anatomy of the ear is shown in Figure 3.3.

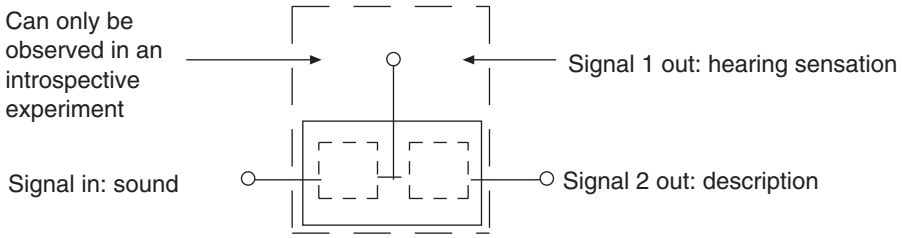


Figure 3.1 The psychophysical system.

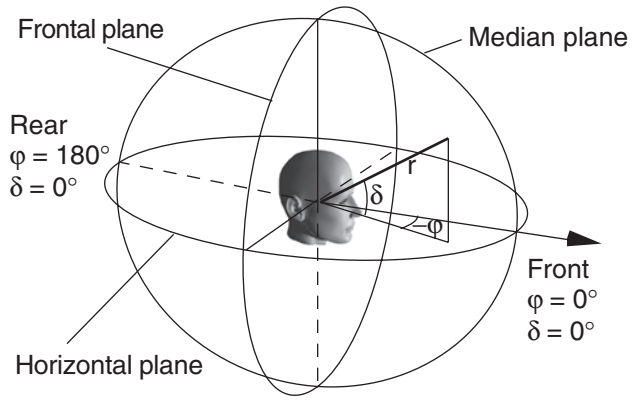


Figure 3.2 The head-related coordinate system.

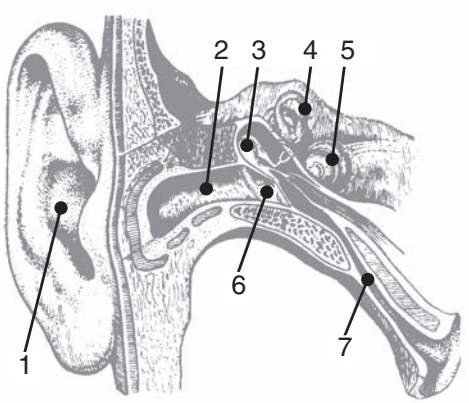


Figure 3.3 Anatomy of the ear: (1) pinna, (2) ear canal, (3) ossicles, (4) semicircular canals, (5) cochlea, (6) eardrum, and (7) eustachian tube.

The functional principles of the three parts are shown in Figure 3.4. The outer ear may be regarded as a signal collector or antenna, the middle ear as an impedance converter and the inner ear as a transducer from mechanical vibration to electrical signal.

The Outer Ear

The outer ear consists of the *eardrum*, the *ear canal*, and the outer, visible part of the ear, the *pinna*. The pinna contributes to many of the directional properties of hearing mentioned previously, but the head and the torso also influence the directional characteristics of hearing. A trained listener can quite easily discern from where a signal is coming, using only one ear. This is due to the changes in the *spectral content* of the signal that in turn are a result of the reflection and diffraction of sound around the head. These phenomena are angle-dependent and affect the sound primarily at frequencies above approximately 1 kHz. The curves shown in Figure 3.5 are examples of the frequency response in a pinna for various angles of sound incidence in the horizontal plane (sometimes called lateral angles). The curves show the magnitude of the *head-related transfer functions* (HRTFs). They are complex and relate to the sound pressure at the place of the head center. The HTRFs are quite individual, and human heads have small asymmetries between the left and right ears, influencing the localization ability.

The shape of the pinna is of great importance for our directional hearing. We can differentiate between sounds incident from the rear and front directions because of the shadow action of the pinna. At frequencies above 2 kHz, the level difference in sound from rear and front directions can exceed 10 dB.

Various *resonances* in the pinna are excited at frequencies typically above 3 kHz. The *modes* of these resonances are important for the directional hearing outside the horizontal plane, particularly at angles of incidence close to the median plane. Some examples of the behavior of these modes are shown in Figure 3.6. The excitation level of the various resonances depends on the angle of incidence of the sound.

Accurate aural localization of sound is important in many cases, such as hearing warning signals or optimizing speech intelligibility in noisy surroundings. Accurate localization is also important when

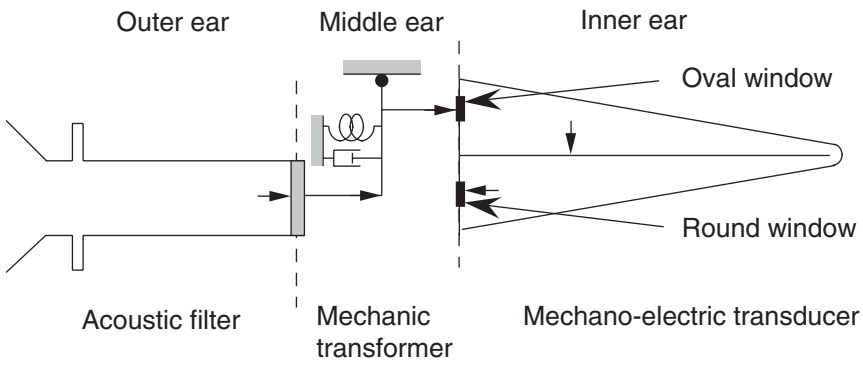


Figure 3.4 The functional components of the ear.

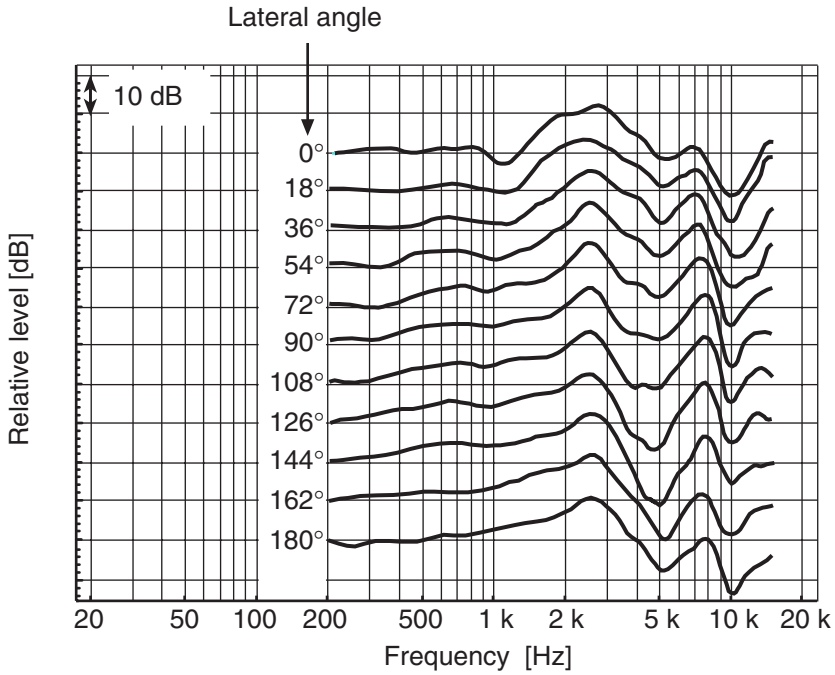


Figure 3.5 Frequency response for the head-related transfer functions from free field to the entrance of the ear canal as a function of the lateral angle φ in Figure 3.2. (After Ref. 3.2)

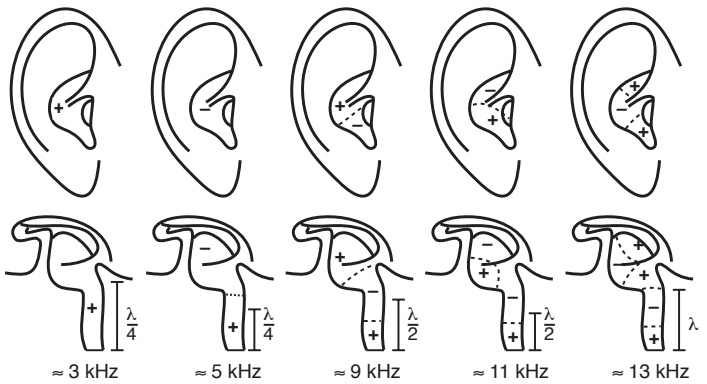


Figure 3.6 Acoustic modes in the pinna. (After Ref. 3.1)

simulating 3-D sound for games, virtual reality, and architecture. It is difficult to recognize the angle of incidence for sound arriving in the median plane. This is due to the lack of binaural cues as described later in this chapter. In the special case of sound incident in the median plane, hearing has to use memory and spectral cues to localize from which angle the sound is coming. The excitation of the resonances in the outer ear is important under such conditions.

As an example, the mode corresponding to the resonance at the frequency f_{03} in Figure 3.6 will be difficult to excite by sound from the frontal direction but will be easy to excite using sound coming from above. Modes at higher frequencies will behave analogously for various angles. The low frequency resonance f_{01} (at approximately 3 kHz) is of great importance for the impedance matching of the ear to the incident sound. The acoustic sensitivity of the ear is usually highest at this frequency. When the ear canal length is equal to $(1 + 2n)\lambda/4$ where λ is the wavelength of the sound, and n stands for a natural number, the ear canal will act as a quarter-wave transformer, and the impedance of the eardrum will appear small to the outside sound field.

The ear canal is essentially a tube, having an average length of 25 mm and an average diameter of 7 mm. The walls of the ear canal are acoustically hard. An example of the frequency response function of an ear canal is shown in Figure 3.7. The input impedance of the ear canal is of great importance in the design of earphones and headphones. The frequency response sensitivity increase of 10–20 dB also results in increased risk of hearing damage for sounds in this frequency range.

The eardrum is an elastic membrane separating the outer and middle ear sections. Most of the acoustic power at the eardrum will be transferred to the middle ear at medium frequencies.

The Middle Ear

The middle ear contains the three *middle ear bones*. These are ossicles and are the malleus, the incus, and the stapes. Normally, the middle ear is acoustically sealed from the outside except for the sound that enters through the vibrations of the eardrum (see Figure 3.8).

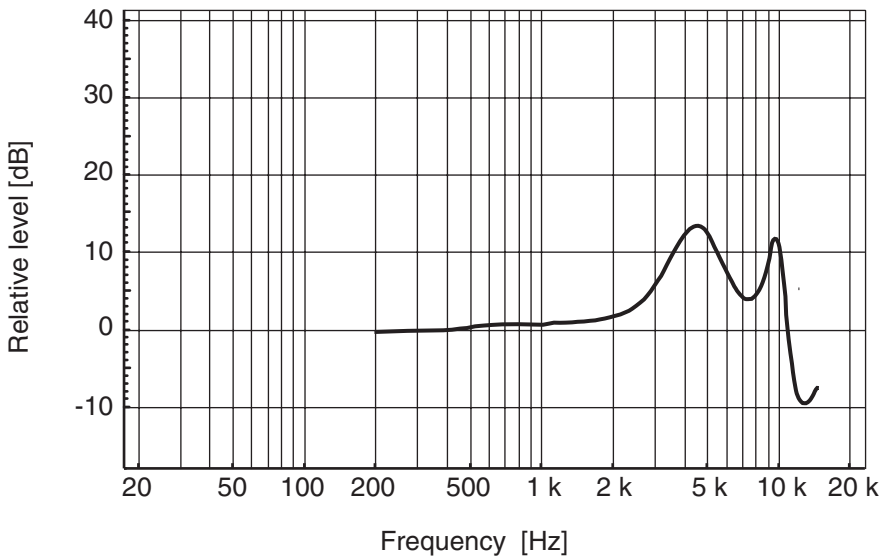


Figure 3.7 An example of the measured level difference between sound pressure at the eardrum and the entrance of the ear canal. (After Ref. 3.2)

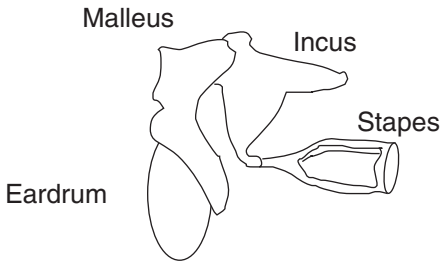


Figure 3.8 The ossicles attach to the eardrum and to the oval window of the cochlea.

To make it possible to equalize the static pressure, so that the eardrum will not be pre-tensioned, there is an opening, the Eustachian tube, leading to the mouth. The ossicles act as a mechanical impedance transformer to transfer the power from the eardrum to the inner ear in an optimal way. The muscles holding the ossicles can act as a form of ear protection. At sound pressure levels above approximately 70 dB, the *stapedius muscle* can change its mechanical properties and give approximately 20 dB of reduction in sound transmission. This is called the *acoustic reflex*. However, because of the reaction time required, more than 0.1 s, there is little protection against *transient sounds*.

Because of the mechanical transformation between large and small movement due to the lever action of the ossicles, and because of the area change, approximately 30:1, between the eardrum and the oval window (where the stapes attaches), one will have an impedance transformation between the high internal impedance of the inner ear and the low impedance at the eardrum. The impedance change reduces what would otherwise be approximately 30 dB attenuation to nearly zero. Measurements of hearing loss in people having had their ossicles removed show results of about 20 to 30 dB of hearing loss.

The Inner Ear

The inner ear consists of the cochlea and the semicircular canals. The latter are essential for our ability to keep balance but are normally not included when we speak of the inner ear or cochlea. The cochlea borders the middle ear with two membrane covered openings—the oval window and the round window. The oval window is attached to the stapes. As shown in Figure 3.3, the cochlea is spiral-shaped. The bone surrounding the cochlea is the hardest bone in the body. The length of the cochlea spiral is approximately 35 mm. It has a volume of approximately 50 mm^3 and a mean diameter of approximately 1.5 mm.

The physiology of the cochlea is shown in the section in Figure 3.9. The cochlea contains three canals filled with watery liquids. The canals of the cochlea are the vestibular canal, the cochlear duct, and the tympanic canals. The canals are joined at the helicotrema, the top end of the cochlea.

The basilar membrane, which separates the tympanic canal from the other canals, holds the inner and outer hair cells. There are approximately 12,000 outer hair cells, each holding some 100–200 stereocilia (hairs). The inner hair cells number about 3,500 and typically hold 50 stereocilia each. The movement of the stereocilia triggers the inner hair cells, which then send their signals to the brain down the auditory nerve. The outer hair cells play an important role as effector cells that are mechanically active and can change their length by acoustic stimulus. This results in an active feedback mechanism that amplifies the response of the ear, particularly close to the threshold of hearing. The system acts as

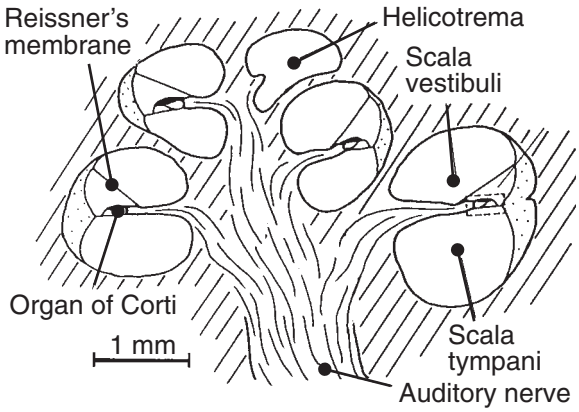


Figure 3.9 Physiology of the cochlea. The small rectangular area at the right is shown magnified in Figure 3.10.

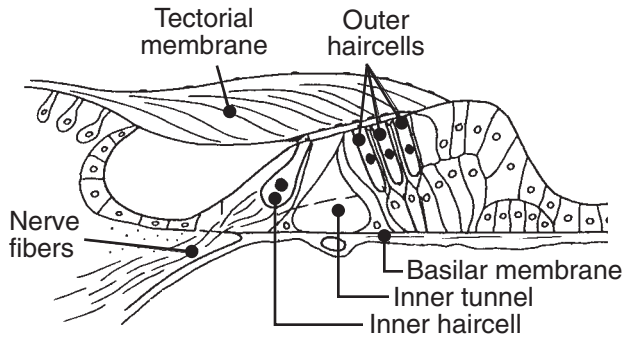


Figure 3.10 The organ of Corti.

a cochlear amplifier. One can note the active process by the phenomenon of otoacoustic emissions. If the ear is stimulated by a click, it will quickly respond by sending out sound. This forms the basis of a simple noninvasive test for hearing defects. The ear may also sometimes send out sound in the absence of exterior stimuli.

Figure 3.10 shows a section through the organ of Corti, indicated in Figure 3.9. One can see the arrangement of the basilar membrane, the hair cells, and the cilia. The upper part of the cilia attach to the tectorial membrane, which gives the arrangement extra moving leverage.

The movement of the basilar membrane is determined by the vibrations induced at the oval window by the sound incident on the ear. Since the speed of sound in the fluid is high, and the cochlea is short, the sound pressure at all points in the fluid will be approximately equal inside both the vestibular and tympanic canals respectively. The basilar membrane can be thought of as built of resonant sections that are highly damped. The sound pressure difference between the two canals will set the membrane in motion. The motion extends over a large area as illustrated in Figure 3.11, even with single frequency excitation. The motion of the basilar membrane is, however, not frequency selective enough to explain the frequency resolution of human hearing. Further signal processing takes place as the nerve signals propagate to the brain and in the brain.

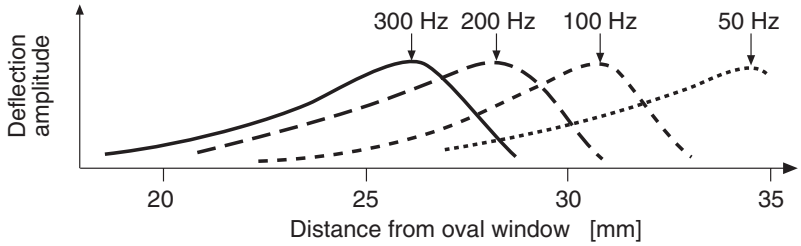


Figure 3.11 The envelope of the vibration amplitude of the basilar membrane for some frequencies. (After Ref. 3.4)

3.3 PSYCHOACOUSTIC DIMENSIONS

While it is possible to describe the physical properties of sound by using such terms as sound pressure, phase, frequency, and *duration*, it is not possible to describe the auditory experience similarly, since the various dimensions of the auditory experience are interrelated. In addition, the auditory experience of one listener is likely to be different from another. Even within one listener, it may change with time. The two most obvious psychoacoustic dimensions are *loudness* and *pitch*, but there are also others such as *roughness*, *sharpness*, and *timbre*. As one is exposed to an increase in sound pressure level, the loudness of the sound increases. The relationship is quite nonlinear; for example, it varies according to frequency and duration.

Loudness and Loudness Level

The *threshold of audibility* (hearing threshold) is the lowest sound pressure level at which one can perceive sounds. The threshold of audibility is usually measured over a frequency range extending from 20 Hz to 20 kHz for a plane-wave incident in the frontal direction of the listener. It is difficult to measure the hearing threshold correctly at low frequencies because of the difficulty of insulating the test chamber against other interfering low frequency sounds and loudspeaker or headphone nonlinear distortion. Different frequency response curves will be obtained for the threshold of audibility for different persons, but even different measurement procedures give different results. The threshold of audibility is more rigorously defined as the minimum perceptible free-field sound pressure level that can be detected at any frequency over the frequency range of hearing.

The threshold for a particular test person is in practice determined by subjecting the person to test tones at known sound pressure levels in an *anechoic chamber*. The sound field should be a plane-wave over a sufficiently large area. The background noise in the chamber has to be low enough so as not to interfere with the measurement. Since the threshold of audibility varies with signal duration, the test tone time has to be more than 0.3 s. A duration of 1 s is typical.

The threshold of audibility (the dotted curve marked MAF [minimum audible field]) is shown in Figure 3.12 and is frequency-dependent. The curves in the figure are from the ISO 226:2003 standard and are the measurement results obtained with the listeners facing a pure tone plane-wave in an anechoic chamber. The sound levels were measured at the listeners' head position without a listener present. The listeners were otologically normal young adults, ages 18 to 25 years.

The *Loudness Level* (LL) of a sound is the sound pressure level of a reference tone at 1 kHz that is perceived as equally loud as the test sound. The LL is expressed in the unit *phon*. The variation in sound

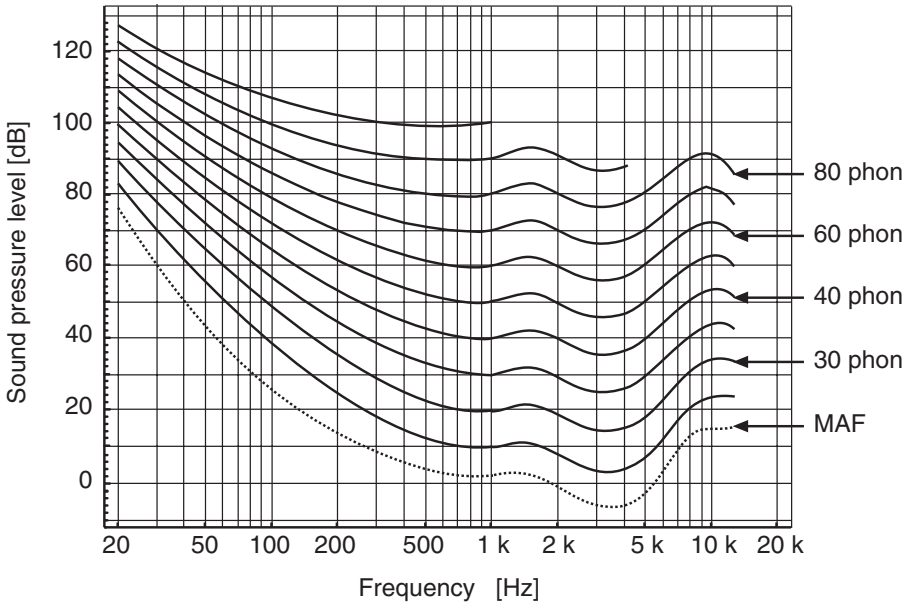


Figure 3.12 Equal loudness contours. The curve marked MAF illustrates the minimum audible sound field. (after ISO 226:2003)

pressure level for equal LL is described by *equal loudness contours*. These are measured using test persons in the same way as when determining the threshold of audibility, but in this case, the persons are asked to determine when the test tone at some frequency is equally loud as the reference tone. The standard deviations obtained in these types of psychoacoustic experiments are quite large, and the curves shown in Figure 3.12 represent averages of the equal loudness contours of many persons determined to have normal, undamaged hearing.

The equal loudness curves are very frequency-dependent at below 30 phon, particularly at low frequencies and low sound pressure levels. This is of great significance in the design of audio equipment since it makes it virtually impossible to have correct timbre in sound reproduction unless one listens at the same LL as was used in the actual recording. Only by using an adaptive frequency response equalization system that senses the LL (and has knowledge about the original level) can one have approximately the right timbre in sound reproduction. It is impossible to achieve this control by using what is known as *loudness compensation* in conventional amplifiers. A sound reproduction system playing louder than another does usually appears *better* in comparison, having seemingly stronger base and treble response. Because of this, it is always important to use the correct presentation sound levels in listening tests.

A trained listener has a threshold for *just noticeable differences* in sound pressure level of approximately 1 dB for pure tones and may sense down to about 0.3 dB deviations in octave bands in broadband spectra. At LL above 120 phon, the sound will be so strong that one can *feel* the sound, i.e., *threshold of feeling*. In the same way, there is a *threshold of pain* at approximately 140 phon.

It is useful to be able to measure the LL by instruments. Various methods have been derived to make this possible, both analog and digital. Today, digital methods based on various types of the spectral

analysis are the most common. The two most common metrics are *Steven's phon* and *Zwicker's phon*, defined by the ISO 532 standard. Tracking of the dynamic behavior of loudness level can be difficult though. It is also important to realize that in reality, the loudness of a sound will depend on the angle of incidence and the duration of the sound, properties that are not taken into account in the standard. Conventional acoustic noise measurements use omnidirectional microphones, sensing the sound equally independent of the angle of incidence of the sound.

One usually prefers to characterize the approximate LL of a sound by measuring the sound pressure level with a frequency filter correction. Such filters, often called *weighting filters*, may, if used appropriately, yield measurements with good correlation to subjective loudness determination. Common weighting filters are so called A, B, and C filters. When used in measurement, these give the *sound level* reading expressed in dBA, dBB, or dBC depending on the filter used or as L_{pA}, L_{pB}, or L_{pC} [dB] as discussed in Chapter 2. The frequency response curves of these filters are shown in Figure 3.13.

The filter characteristics of the A, B, and C filters were chosen to somewhat resemble the sensitivity characteristics of the equal loudness contours in the intervals 20–55 phon, 55–85 phon, and 85–140 phon respectively. Because of the availability of data and considerable experience, the sound level expressed in dBA has become the most commonly used, particularly for the determination of the risk of noise-induced hearing loss.

As most sensory experiences, loudness is related *logarithmically* to *excitation strength*, in this case sound pressure, over some level. For example, a doubling of sound pressure at 1 kHz corresponds to an increase in LL of about 6 phon. An alternate way of measuring loudness is to measure using a metric, which has the property that an increase in loudness corresponds to a proportional increase in the metric value. Such a metric is *Loudness* (*N*), which is expressed in units of *sone*. By definition, 1 sone corresponds to a Loudness Level of a 40 phon, 1 kHz, reference tone. The Loudness value is equal to how many times stronger, or weaker, a sound is compared to this reference tone. Loudness can be measured

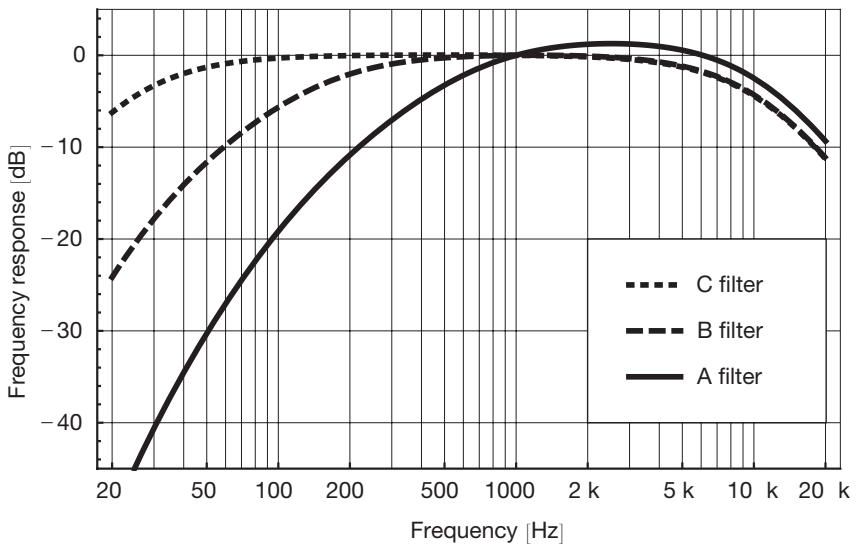


Figure 3.13 Attenuation characteristics for A, B, and C filters.

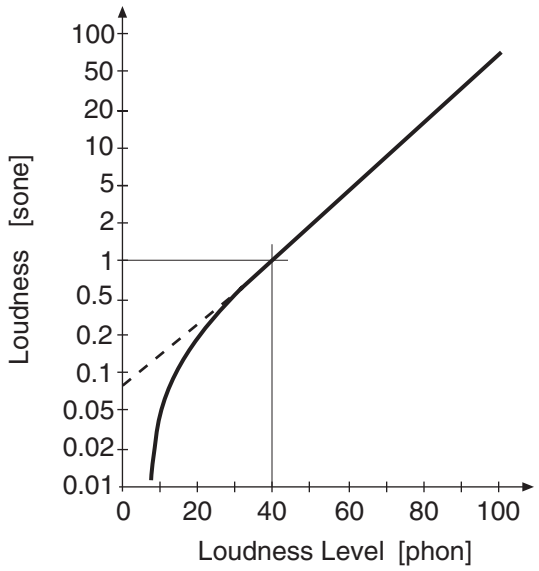


Figure 3.14 The nominal relationship between Loudness and Loudness Level at a frequency of 1 kHz. (After Ref. 3.3)

subjectively using a panel of listeners judging the loudness or by analog or digital means as with LL. The approximate relationship between Loudness and Loudness Level at 1 kHz is shown by the curve in Figure 3.14.

Pitch and Timbre

There are many subjective dimensions of sound other than loudness. They include *timbre*, the subjective perception of spectral content (frequency and balance between various parts in the spectrum) and *pitch*, the subjectively perceived *frequency* of a tone, the subjective *duration* of a sound, and the perception of the variation in time of sounds, such as *modulation* and *fluctuation*.

Timbre is defined as the attribute of sound that makes two sounds, having the same loudness and pitch, dissimilar. The timbre of a complex sound depends on its spectral characteristics. In audio equipment, various types of frequency selective filters are used to change the timbre of sounds, such as bass and treble filters.

Pitch is influenced by not only the frequency of the sound but also the intensity, the duration, and the spectral content. The presence of other harmonically related tones also influence pitch. Pitch is expressed in units of *mel*. A reference sine wave, having a frequency of 1 kHz, is defined as having a pitch of 1000 mel. One measures pitch by subjective comparison to this tone. Pitch does not vary linearly with frequency, as illustrated by the curve in Figure 3.15. The curve shows (ratio) pitch for various frequency tones heard at a loudness level of 60 phon.

Humans can sense static frequency deviations of 3 Hz or more at frequencies below approximately 0.5 kHz and of about 0.2–0.3 percent above, if the method of beats is not used. In the method of beats, one listens to two tones having frequencies f_1 and f_2 simultaneously, and, because of the limited frequency

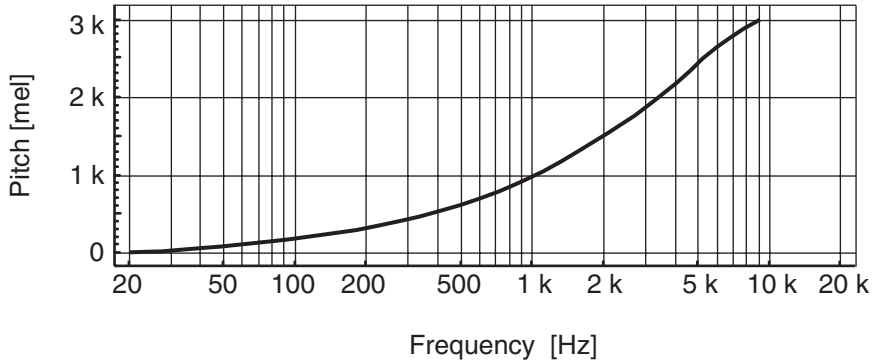


Figure 3.15 Pitch as a function of frequency (loudness level 60 phon). (After Ref. 3.3)

resolution of hearing, the two tones will subjectively combine into one tone having a frequency $(f_1 + f_2)/2$, modulated by the difference frequency between the two tones $|f_1 - f_2|$ if $|f_1 - f_2| < 10$ Hz.

Hearing is quite sensitive to various forms of repetitive changes, such as the amplitude or frequency of a tone. These changes are called *modulation*. The maximum sensitivity is found to modulation frequencies of about 4 Hz. Slow modulations having frequencies below 20 Hz are called *fluctuations*. For frequencies above approximately 20 Hz, the modulation subjectively changes character, and the modulation is perceived as *roughness*. The roughness sensitivity depends on the critical bandwidth. At 1 kHz the maximum roughness is heard for modulation frequencies of about 70 Hz (when the tone components are at the frequency band edges of the critical band).

An interesting phenomenon concerning pitch is virtual pitch. It is connected to a phenomenon called the *missing fundamental*. If one listens to a sound composed of a harmonic series of tones—for example, 200 Hz plus its first ten harmonics—the elimination of the fundamental (200 Hz) does not make much difference to the pitch. This is of great importance in audio engineering, since it allows even a loudspeaker with poor low frequency reproduction to have an *acceptable* low frequency response for many sounds.

3.4 EFFECTS OF EXCESSIVE SOUND EXPOSURE

Temporary and Permanent Threshold Shift

Excess sound exposure can damage various parts of the hearing mechanism, but even quite normal excitation can influence the sensitivity of hearing. We have noted earlier that the Stapedius muscle can become tensioned and in that way provide some hearing protection by reducing the vibration transmission by the ossicles. Short-time exposures, even below those that cause *noise-induced hearing loss*, may lead to so called *temporary threshold shift* (TTS). The recovery time depends on the sound pressure level and on the duration of the sound as illustrated in Figure 3.16. With exposure to a sufficiently high sound pressure level and long exposure times, hearing will not recover; the sound will have caused *permanent threshold shift* (PTS).

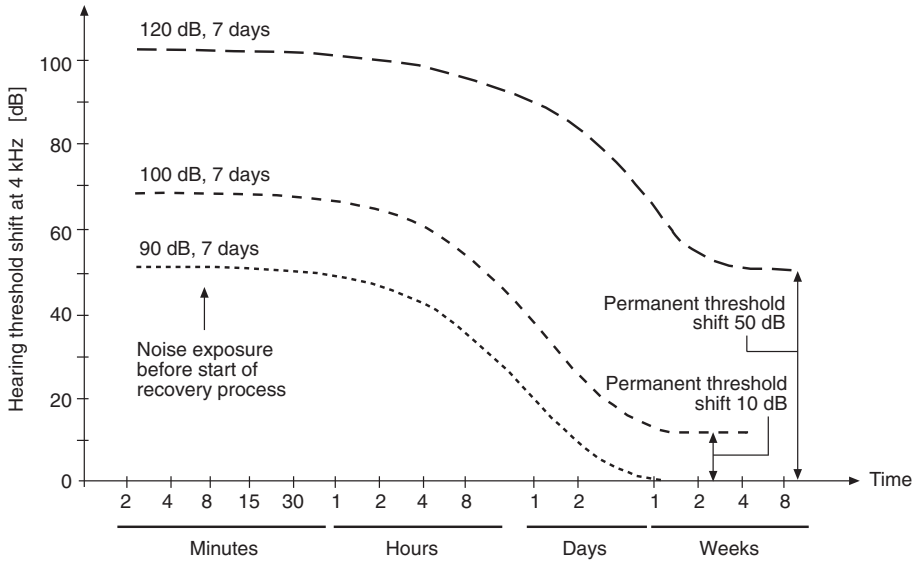


Figure 3.16 Examples of time dependence of recovery from TTS. (After Ref. 3.5)

Both TTS and PTS relate to phenomena in the inner ear, but PTS means that the sound exposure has led to permanent physical damage of the hair cells. The hair cells die, to some extent because the noise reduces their blood supply but also because of actual physical damage to the cilia. Such damage cannot be repaired, and the use of hearing aids can only restore some of the hearing abilities.

Hearing Impairment and Hearing Loss

The two main categories of hearing loss are *sensorineural hearing loss* that affects the cochlea or the auditory nerve and *conductive hearing loss* that affects the ear canal, tympanic membrane, or ossicles. In addition, there are cases of mixtures of these two categories, *mixed hearing loss*.

Sensorineural hearing loss is usually permanent and irreversible. Beside noise, it can be caused by poor blood supply or ototoxic chemicals. Since conductive hearing loss is of mechanical origin, it can often be compensated for by surgery in which the ossicles are replaced by a wire coupling the eardrum and the oval window. Conductive hearing loss is often the result of middle ear infections.

Prevention of noise-induced hearing loss is usually based on eliminating noise exposure, including less noisy equipment, workplace changes, use of ear defenders, or minimization of exposure time. Many studies indicate that hearing loss is linked to the *energy exposure*, that is, sound intensity and exposure time. This is the rationale for the use of the energy equivalent noise level $L_{\mathrm{Aeq},T}$ mentioned in Chapter 2.

Audiometry and Audiogram

In many countries, there is a requirement for yearly monitoring of an individual's hearing status using *audiometry* in an *audiometric testing program*. The purpose of such testing is to protect hearing sensitive

employees so that their hearing is minimally impaired. Usually hearing is then checked at a number of specified frequencies, a method known as *tone audiometry*.

For each ear, the *pure tone audiogram* indicates the difference between a person's actual hearing ability and that accepted as normal, undamaged hearing according to ANSI and ISO standards, often expressed in dB HL (hearing level). Since occupational hearing loss (and most other types of noise-induced hearing loss) is caused by a variety of sounds having quite broad spectra, and because of the sound pressure amplification by the ear canal in the 3–4 kHz frequency range many people exhibit large hearing loss in this frequency region and above.

Tone audiometry average values for the frequencies .5, 1, 2, and 3 kHz are often used to define hearing loss. For adults mild hearing loss is defined as when the hearing level is 27–40 dB HL. A hearing aid is necessary when the the value exceeds 40 dBHL. Moderate hearing sensitivity loss is defined as 41–55 dB HL.

Figure 3.17 shows examples of hearing loss curves. The reason hearing is not completely damaged in the frequency range where it is most severely damaged is that the basilar membrane moves over a large area as discussed previously, even at frequencies of sound that correspond to those of the damaged hair cells. Impaired hearing particularly influences our ability to understand speech and to enjoy music. The consonants that consist of mainly high frequency sound suffer the most, whereas the fundamentals and harmonics of vocal sounds are less affected by hearing loss. However, in most western languages, the consonants are the major carriers of speech information.

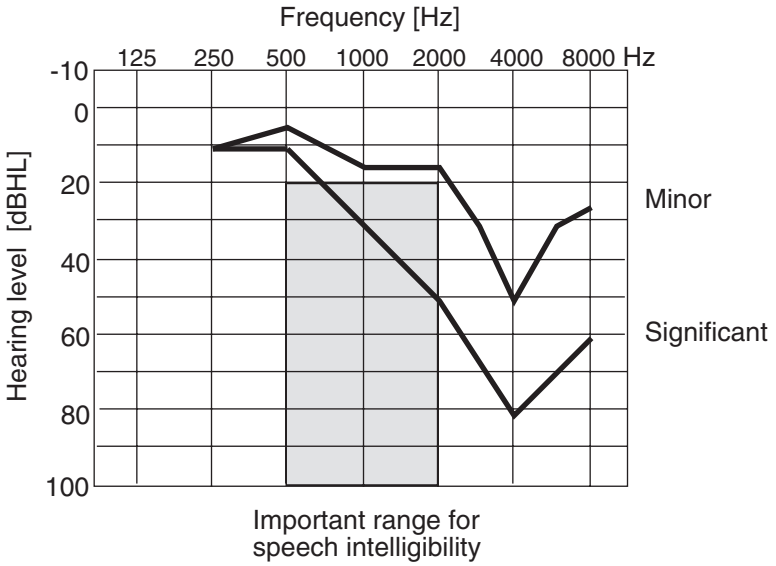


Figure 3.17 Tone audiogram showing examples of minor and significant hearing loss. Significant hearing loss is said to have occurred when the hearing loss curve reaches the shaded area. Compare to Figure 3.30 to see important frequency ranges for voiced and unvoiced sounds.

This text is an attempt at indicating the problem:

- Vowels only—••e •o•• o• •o••e••a•io• •o• •ea•i•• •o•• i• •i••.
- Consonants only—Th• c•st •f c•mp•ns•t••n f•r h••r•ng l•ss •s h•gh.

An additional type of audiometry is *speech audiometry*, where the difficulty in discriminating speech in the presence of noise is assessed.

Many people suffer from *age-related hearing loss* (ARHL) that usually takes the form of high-frequency hearing loss and is associated with a loss of hair cells, sensitive at high frequencies. This loss in the elderly is often called *presbycusis*. ARHL is a result of general wear-and-tear due to noise exposure in society. Health-related conditions such as high cholesterol levels, and habits, such as smoking, that affect the cardiovascular system, seem to influence ARHL. Some typical curves showing hearing loss due to ARHL are shown in Figure 3.18.

Because of changing working habits, it is likely that there will be a smaller gender difference regarding ARHL in the future. It is also important to realize that there is a difference between *hearing disability* and hearing impairment.

Tinnitus

Tinnitus is when a person perceives sounds, even though there is no sound exposure. Tinnitus is often the result of excessive noise exposure to either continuous or transient noise. Even a single exposure of sufficiently strong impulsive noise may cause hearing damage and associated tinnitus. Tinnitus may cause sound impressions that can be sporadic or steady, noisy or tonal. Sometimes the perceived sound can be so strong as to cause annoyance or prevent sleep, among other things. Often, tinnitus is a side result of hearing damage that has resulted in permanent hearing loss.

Tinnitus is often the result of sporadic hearing cell activity. One can think of the tinnitus-induced sound as being caused by hearing amplifying in spite of not having anything to amplify (because of the hearing loss). As in other regenerative systems, excess amplification can lead to instability and oscillation. Usually the tinnitus phenomenon starts as noise that after some time converts into a steady tonal oscillation. Once tinnitus has occurred, rest, absence of stress, and blocking sounds, such as low-level music, can help minimize the subjective discomfort.

Criteria for Excessive Noise

In 1990, the European Community instituted a noise legislation that sets a limit so that action is taken when the daily personal exposure of a worker to noise is likely to exceed 85 dBA. (The EC Directive on Noise 2003/10/EG is the most recent version at the time of writing.) One then has to take appropriate action as specified in the directive:

- Measure noise levels
- Provide ear protectors and train workers in their use
- Inform workers of the risk
- Make hearing checks available

In case the sound level exceeds 90 dBA, further action should be taken.

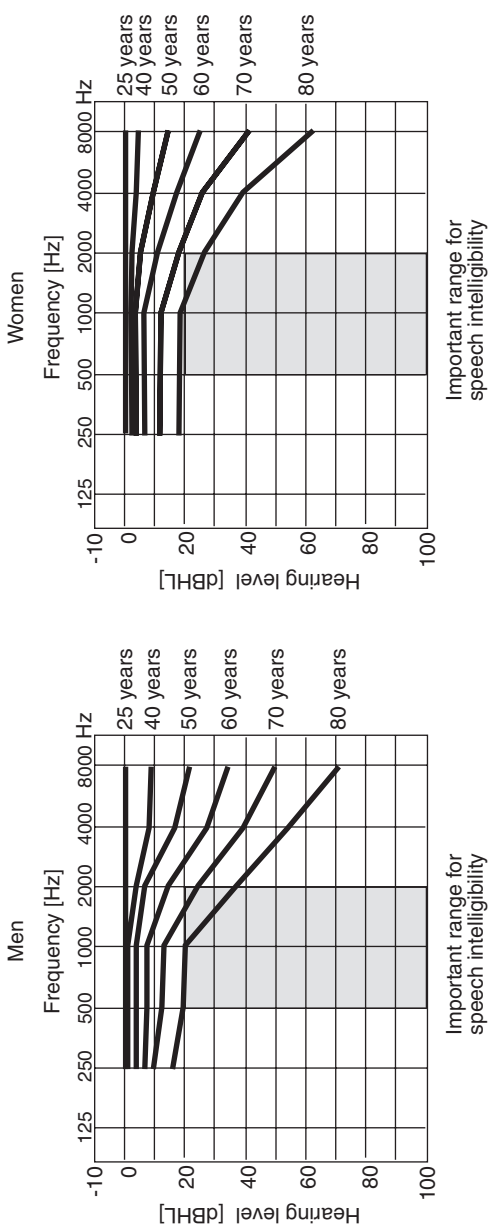


Figure 3.18 Hearing threshold shift due to age-related hearing loss for various age groups (for men and women respectively). (After Ref. 3.6)

The limiting value of 85 dBA does not imply that hearing loss cannot occur at sound levels below this value. For negligible risk, the sound levels have to be below 70 to 75 dBA in conjunction with hearing rest outside working hours.

In many other countries there is an additional national standard for the determination of risk for hearing impairment and hearing damage by excessive sound.

Impulsive Noise

Impulsive noise is particularly troublesome as a cause for noise-induced hearing loss. The reason is that hearing does need a certain amount of time to perceive sounds. Since much of the damage that can be caused by excessive noise is due to mechanical overload in the cochlea, the damage can be fairly instantaneous. Gun noise and other explosion-type noise can result in single exposure damage. The Stapedius reflex needs some time to react, which is not possible in the case of isolated transients.

Even repetitive tone bursts are not perceived as loudly as the steady-state tone having the same amplitude, shown by the data in Figure 3.18. The curves show the length that a burst of sound needs to be perceived as loudly as the steady-state tone. One sees that a duration of approximately 0.2 s is needed for this to occur. Figure 3.19 illustrates the sound level meter characteristics, *IEC impulse*, *IEC fast*, and *IEC slow*. One notes that even the impulse setting does not allow measurement of the peak values of short, transient sounds. Repetitive impulsive noise often takes on the properties of continuous noise.

Ear Defenders

The two most common types of ear defenders are circumaural defenders and earplugs: earmuffs that cover the external ear and insert earplugs that seal the ear from sound, respectively. Some examples of *real ear attenuation* for some protectors are shown in Figure 3.20.

The efficiency of earmuffs and earplugs in providing protection from noise depends on how the devices fit the person. Earplugs and earmuffs can be equally good protectors. The fact that ear muffs are larger does not mean that they necessarily protect better. Earmuffs often suffer from poor protection at low frequencies

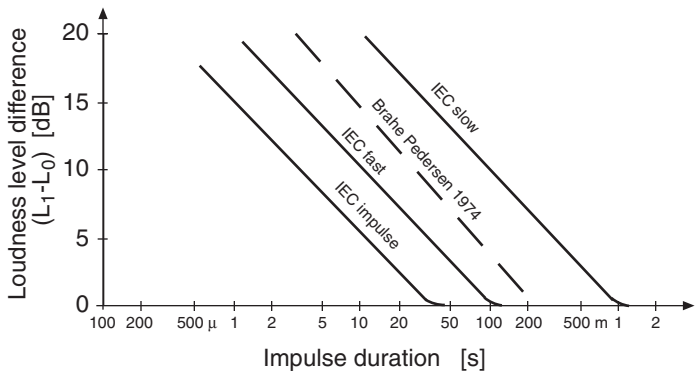


Figure 3.19 Measured loudness level differences between 1 second exposure and tone burst exposure for different burst lengths (at 1 kHz). The curves marked IEC slow, IEC fast, and IEC impulse correspond to the integration characteristics of sound level meters. (After Ref. 3.7)

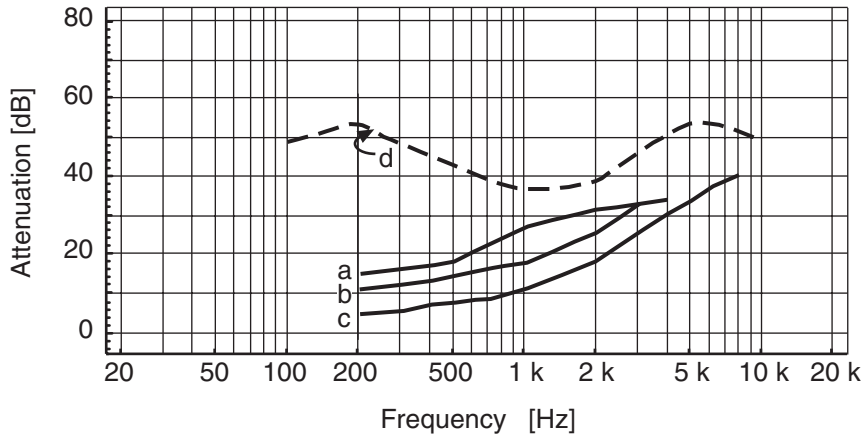


Figure 3.20 Measurement results for the insertion loss by some ear defenders: (a) neoprene insert earplug, (b) earmuff with negligible leakage, (c) earmuff with normal leakage, and (d) bone conduction limit.

because of the resonance due to earmuff mass and head-band compliance. People have different preferences. Many people prefer earplugs since they are easier to carry, lightweight, disposable, and do not show. Earplugs are often preferred in humid climates.

There can be significant differences in attenuation as measured using a manikin or simulator versus the attenuation measured using the response from people. The reason is that it is difficult to take *leakage*, *skin compliance*, and *bone conduction* into account when measuring ear defender insertion loss using a manikin. The skull is set into vibration by surrounding sound. This vibration reaches the cochlea by bone conduction and limits the maximum protection of ear defenders to around 40 dB.

The availability of ear protectors using *active noise cancellation* means that it is, in principle, possible to tailor the insert loss of ear protectors to the spectrum of the unwanted sound.

3.5 MASKING AND CRITICAL BANDS

Masking is the phenomenon when a sound, the *masker* blocks the perception of another sound, the *maskee*. Masking can take place both in the time and frequency domains. The *masking level* is the level by which one sound needs to be increased to still be heard in the presence of a masker. Simplified, one can regard the masking level as the shift of hearing threshold in the presence of another sound, such as a tone or a noise.

Figure 3.21 shows an example of spectral masking. It shows the shifted hearing thresholds of the maskee measured with a tone masker having a frequency of 1 kHz at different levels L_M. The masking is largest when the masker and maskee are close to one another in frequency. The masker's effect is mostly active upwards in frequency so from the viewpoint of speech intelligibility, low frequency masking sounds will also be disturbing, since they will mask speech.

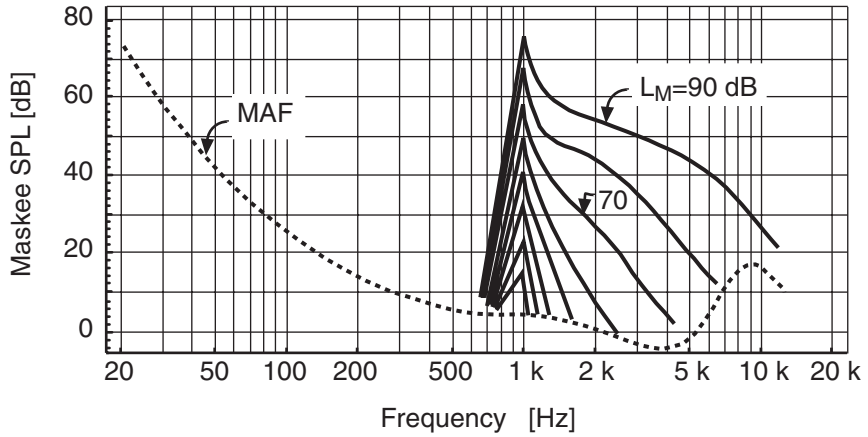


Figure 3.21 Hearing threshold shift of a test tone (maskee) due to masking. The masker is a 1 kHz tone at different sound pressure levels L_M. Solid curves show the resulting hearing threshold of the maskee in the presence of the masker. The dashed line shows the threshold of hearing for tones (MAF). (After Ref. 3.3)

Hearing is characterized by frequency analysis some of which is accomplished already by the basilar membrane. The stimulus is analyzed equivalent to an adaptive filter bank with a large number of parallel frequency selective filters. These are similar to those of bandpass filters used in signal processing by electronic equipment and are called critical band filters. The audio frequency range can be subdivided into 32 critical bands, each having its own "critical band-width".

The critical bandwidth depends on the frequency as shown in Figure 3.22. Up to about 0.5 kHz the critical bandwidth is fairly constant, but at higher frequencies the bandwidth is about that of a third octave band wide bandpass filter. At a frequency of 1 kHz the critical bandwidth is about 160 Hz. There are several ways of measuring the critical bandwidth, these give slightly different results. A common way to measure the critical bandwidth is to use noise having a constant spectral density, increasing the bandwidth of the noise only and to investigate the masking of an added pure tone. Reaching a certain bandwidth, the masking no longer increases as the bandwidth is increased. This bandwidth is the critical bandwidth.

Psychoacoustic experiments have shown that our ability to hear a sound, for example a tone, in the presence of noise centered on the frequency of the tone depends on the bandwidth of the noise. In order for a tone to be heard its power, approximately, has to exceed that of the noise within the critical bandwidth.

In most practical situations, the masker is a noise-like signal with some bandwidth. Wide-band noise will have energy over a bandwidth much wider than the critical bandwidth. It is then necessary to take the masking of several bands into account, since masking is asymmetric and primarily acts upward

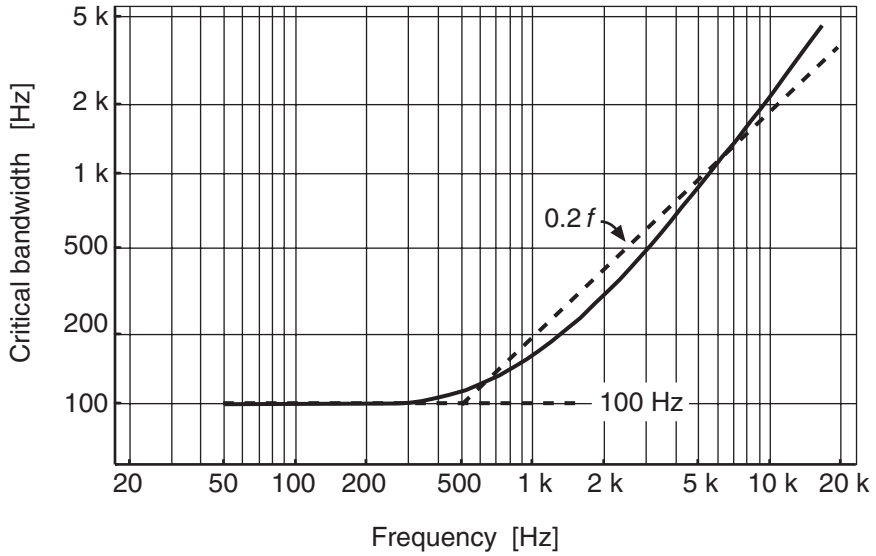


Figure 3.22 The solid curve shows the critical bandwidth as a function of frequency. The broken line marked 0.2*f* shows the third octave bandwidth as a function of center frequency. (After Ref. 3.3)

in frequency. This is done in the determination of loudness and loudness level according to the ISO 532 standard.

3.6 DISTORTION, LINEARITY, AND HARMONICS

Changes to a signal's properties are called *distortion*. *Linear distortion* is characterized by simple frequency-dependent amplitude changes to various parts of the spectrum. *Nonlinear distortion* is characterized by the generation of new sounds at frequencies that are multiples of the frequency of the original sounds, *harmonics*, and is called *harmonic distortion*. This type of the distortion is called *intermodulation distortion* (see Chapter 17). Some nonlinear processes create *subharmonics*. When the signal contains many frequencies, the distortion will take on a noise-like quality; this is of importance in digital signal processing. Sometimes nonlinear resonant systems may show *chaotic* behavior.

Hearing has its own nonlinear distortion; one can easily note this distortion when listening *monaurally* or *binaurally* to the reproduction of two tones or other sounds. One will note a difference between when the sum of the sounds is presented to both ears and when the two sounds are presented to each ear separately. The nonlinearities that can be detected in this way are a property of the cochlea.

Subjectively, low pass filtered sound reproduction may be superior to full frequency range sound reproduction, if the sound reproduction system is characterized by nonlinearities. Digital signal processing may be used to generate tones that are not present originally in the signal but which hearing expects. This technique is sometimes used to extend the frequency range synthetically, at both low and high frequencies.

Consonance is an example of a nonlinearity that takes place after the cochlea. Consonance is characterized by the property that tones having frequency relationships of the forms 2/1, 3/2, 4/3, and so on are perceived as pleasing.

3.7 BINAURAL HEARING

Binaural hearing is listening to sound using both ears. Binaural hearing leads to a clarity and subjective perception of auditory space that cannot be had when listening monaurally, such as when using only one ear. The reasons are three phenomena: (1) localization, (2) noise suppression, and (3) decreased masking.

One can better discriminate between sounds coming from different directions because of the better localization ability that is the result of the ability of hearing to perform cross-correlation analysis of the signals from the two cochleae. This is sometimes called the cocktail party effect because it allows better discrimination when listening to many sounds coming from various angles to the median plane. The cross-correlation analysis is successful because of the level and time differences between the signals from the cochleae when sounds are incident away from the median plane. The differences are a result of the reflection and shadow action of the head.

Localization Using Interaural Time and Level Differences

In studying binaural hearing, it is convenient to model the head as a sphere, having a radius r of 8 cm, with the ears positioned diametrically, as shown in Figure 3.23. The measured interaural time delays suggest that the equivalent diameter of the average head is approximately 16 cm.

The additional travel distance difference for signals reaching the far ear is $r(\theta) + \sin(\theta)$. For a sound arriving from an angle of $\theta = 0.96$ radians, the difference in arrival time will be approximately 0.45 ms. In the case of a tone having a frequency of 0.1 kHz, this corresponds to a phase difference of 0.045 periods, at 1 kHz a difference of 0.45 periods, and at 10 kHz a difference of 4.5 periods.

Experiments show that at frequencies below approximately 1 kHz hearing primarily uses phase differences for localization of sounds. At higher frequencies, it is primarily the time differences in the signal envelope that are important for localization.

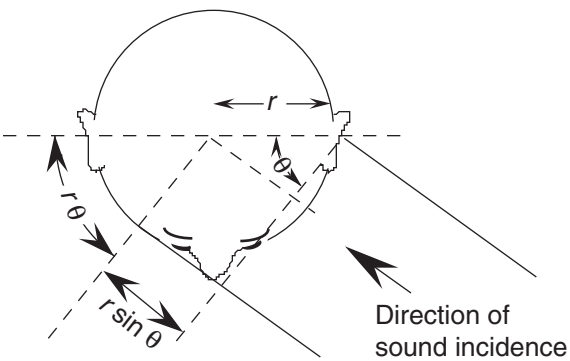


Figure 3.23 A simplified model of the human head for the calculation of the interaural path length difference.

Hearing uses these *inter-ear time differences* (ITDs) in the arrival of sound to the ears for determination of the angle of incidence relative to the median plane. The head causes *inter-ear level differences* (ILDs) of as much as 5 dB at frequencies as low as 1 kHz; at frequencies above 3–4 kHz, the ILD is sufficient for localization.

In case of localization difficulty, small head movements are necessary to decide on the arrival direction. Figure 3.24 shows the localization error using binaural hearing for sounds incident in the horizontal plane (*lateralization error*).

The noise suppression is due to reduced masking. Some undesired sounds, such as noise or excessive reverberation, often arrive simultaneously from many directions, being diffusely distributed in the room. The desired sound can then often be heard more clearly by turning one's head so that one ear faces the sound source. The sound reflecting properties of the head and the correlation ability help us hear the desired sound. The cross-correlation process improves the clarity of the direct sound over the noise and the reverberation.

These properties are of great use to us in daily listening. Binaural sound reproduction uses these advantages. In this technique, sound is picked up by a manikin head and torso, using two microphones mounted in the conchae of pinna replicas. An example of a commercial binaural recording manikin (dummy head) is shown in Figure 5.24. The left and right microphone signals are recorded and then played back to the listener binaurally, using two headphones or using loudspeakers and a crosstalk cancellation filter.

Localization Using Only Spectral and Level Changes

For sounds arriving in the median plane, both ear signals are essentially the same since human ears are fairly symmetric. In this case, the localization ability is poor, since median plane localization to some extent will depend on memory cues.

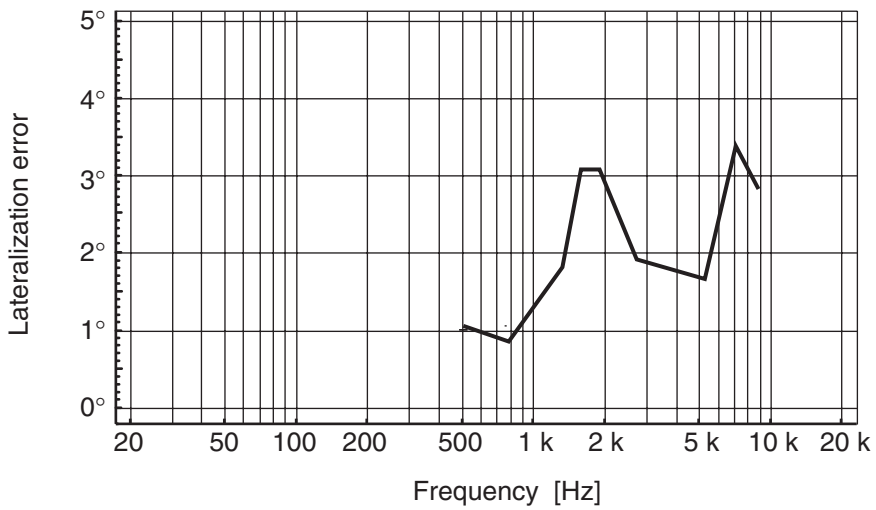


Figure 3.24 Localization uncertainty for tonal sounds incident in the horizontal plane for frontal incidence. (After Ref. 3.3)

Some examples of the spectral differences as a function of angle of arrival in the median plane are shown in Figure 3.25. Tests have shown that one can change the subjectively perceived angle of sound incidence in the median plane by spectral changes to the sound. The result of such an experiment is shown in Figure 3.26. The sound, in this case, was always coming from the same place, but the perceived location in the median plane of a narrowband noise changed, as the center frequency of the noise changed. The localization was in the direction where the HRTFs amplified the respective frequencies.

The poor localization in the median and horizontal plane when using monaural hearing are related; in both cases, the direction finding ability is limited to what is provided by memorized sound cues. Figures 3.27a and b show results from such experiments.

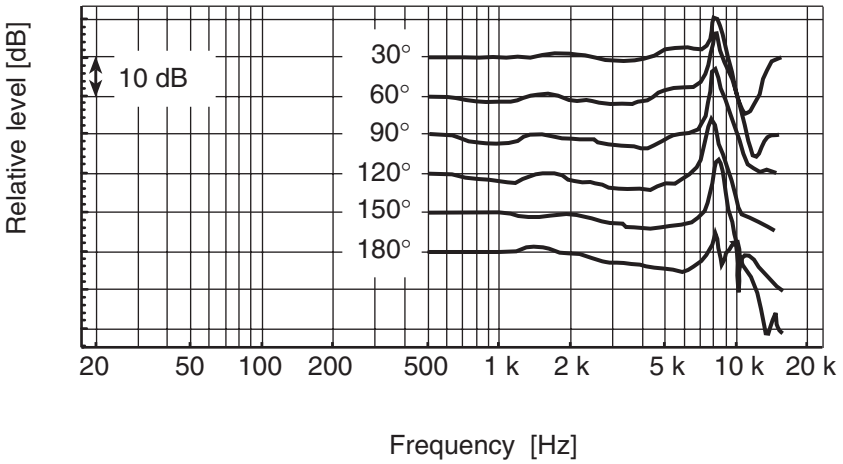


Figure 3.25 Relative frequency response curves for 6 different angles of sound incidence in the median plane ($\varphi = 0$). Each curve is normalized to the response at 0.5 kHz for the respective angle. The median plane angle is δ in Figure 3.2. (After Ref. 3.8)

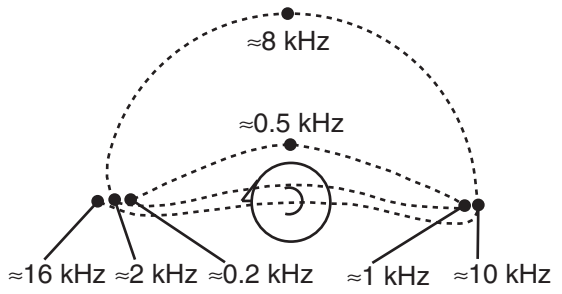


Figure 3.26 Result of experiment showing virtual median plane source positions for various narrow-band noise signals. (After Ref. 3.1)

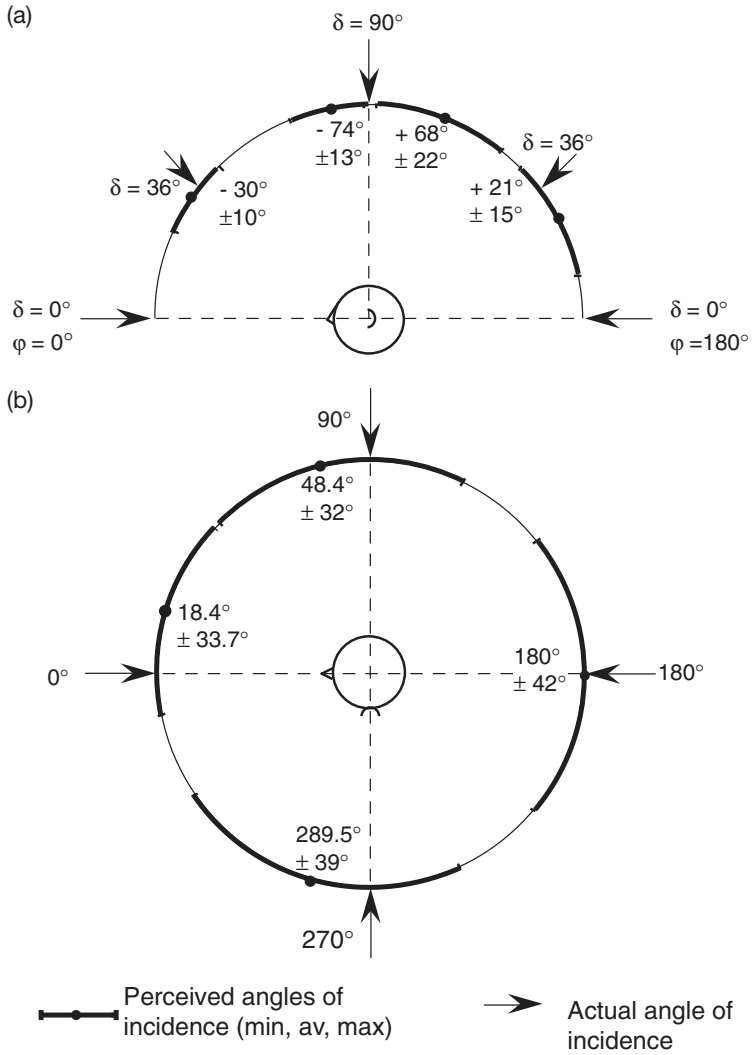


Figure 3.27 Localization and localization uncertainty in the median plane (a). Localization and localization uncertainty in the horizontal plane using one ear (b). (After Ref. 3.1)

3.8 VOICE AND SPEECH

The Voice Mechanism

The fundamental components of the voice mechanism are the lungs, the *vocal cords*, the tongue, the mouth, and the nose, as shown in Figure 3.28. The voice mechanism is powered by air that is exhaled from the lungs.

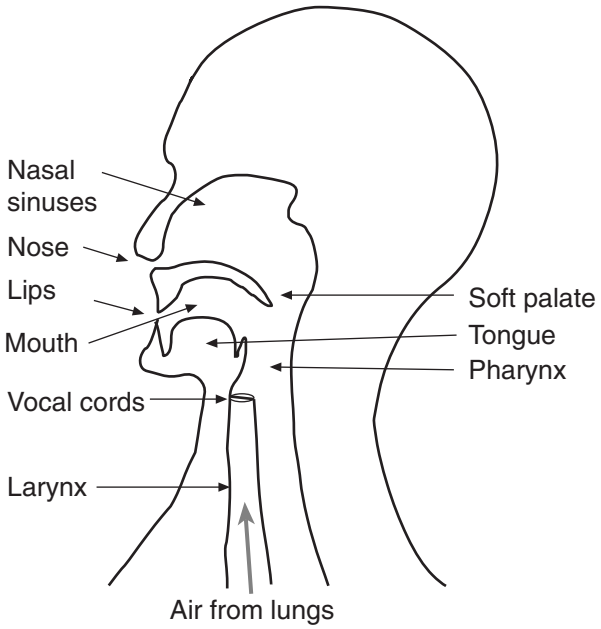


Figure 3.28 Important components in the voice generation system.

There are two basic voice sound components: *voiced* and *unvoiced* sounds. Voiced sounds are a result of the modulation of the air stream by the oscillation of the vocal cords in the larynx. The length of the vocal cord opening and the tension of the vocal cords determine the fundamental frequency of voiced sounds, which differs in men and women. The modulation of the air stream is sawtooth-shaped and is characterized by many harmonics. The acoustical filter, created by the volumes of the throat, the mouth, and the nose, has many resonances that amplify and attenuate the various frequency components in the air stream.

Unvoiced sounds, fricatives such as f and s, and stop consonants such as p and t, are produced by modulation of the air stream using the tongue and lips. The unvoiced sounds have a wide frequency range because of their noise like or transient character.

Spectral Characteristics of Speech

Speech has a wide frequency range, with frequency components from approximately 80 Hz up to 10 kHz. An example of the longtime-averaged speech pressure spectrum level, averaged for men and women, is shown in Figure 3.29; its sound pressure level is approximately 60 dB at a distance of 1 m in front of the talker. The sound power level of the voice is approximately 70 dB. A trained speaker can change the power output of the voice over approximately a 20 dB range. A whisper, entirely unvoiced speech, has an approximate sound power level of 30 dB.

By changing the shape of the volumes—for example, by moving the tongue—one can change the filter parameters and by that the spectra of vocal sound. The peaks in the spectra of vocal sounds are called formants. Figure 3.30 illustrates approximate spectral positions for some speech sounds.

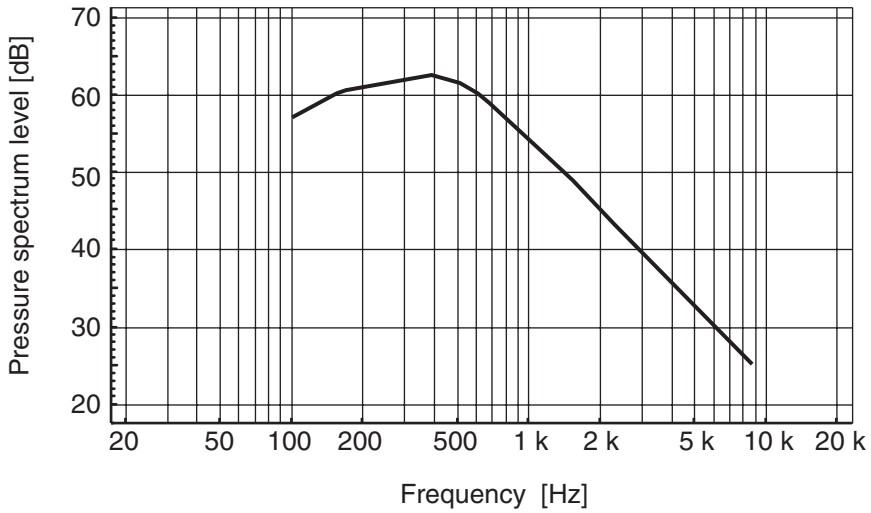


Figure 3.29 An example of longtime averaged speech pressure spectrum level in frontal direction at 0.05 m distance from mouth. (After Ref. 3.9)

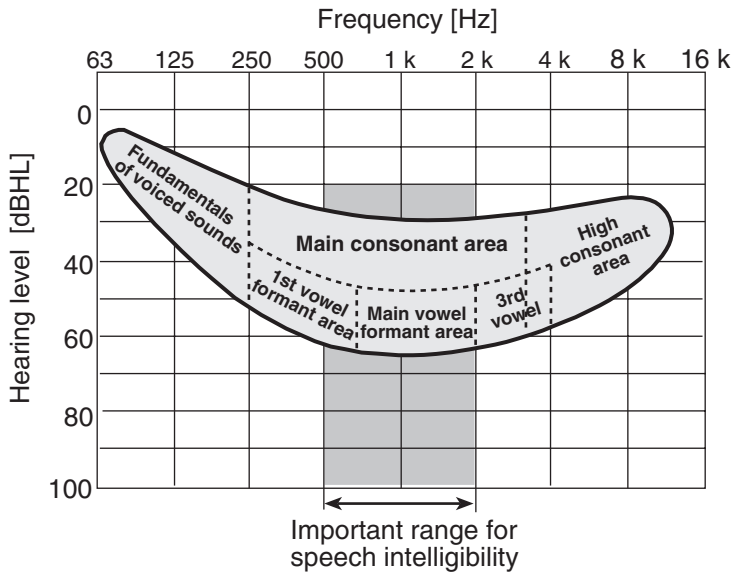


Figure 3.30 Sound pressure levels for voiced and unvoiced speech sound components drawn using the audiogram approach, that is, relative to the threshold of hearing (MAF) shown in Figure 3.12. (After Ref. 3.9)

Figure 3.29 shows further that most of the energy is concentrated at relatively low frequencies due to the voiced sounds. The unvoiced sounds have predominantly high frequency spectral content and are relatively more important for speech intelligibility.

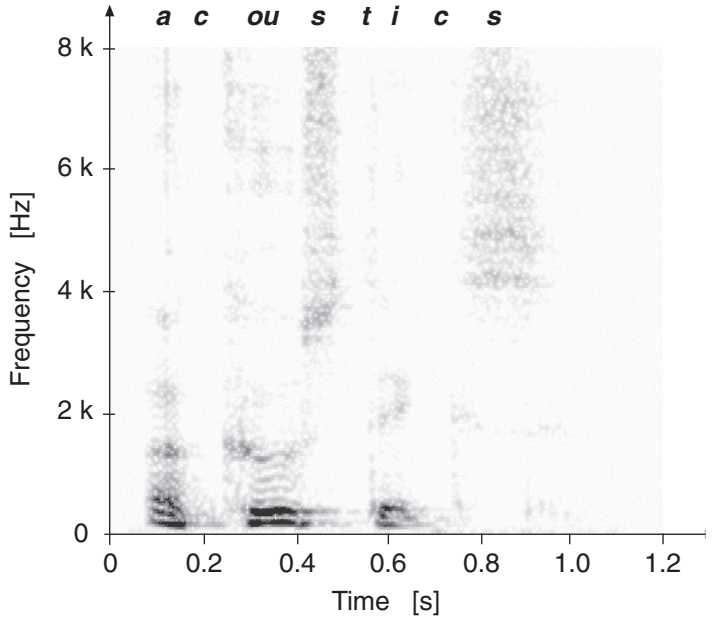


Figure 3.31 A time-frequency analysis for the spoken word acoustics.

It is convenient to show the spectral content of speech using *spectrograms* (time-frequency plots of spectral content). An example of such a spectrogram is given in Figure 3.31. One clearly sees how speech is modulated in various frequency bands. The fundamental modulation frequency of the speech *envelope* is approximately 7 Hz.

Voice Directivity

Speech is primarily radiated by the mouth and nose openings; only at the lowest frequencies is there radiation by the chest. The sound shadow caused by the head will cause some directionality though, as shown by the curves in Figure 3.32.

The directivity is quite substantial at frequencies over 1 kHz; there is a more than 5 dB sound pressure level difference between front and back radiation. Since most of the information in speech is at high frequencies, this means that one should strive to speak in the direction of the listener.

It is clear from this data that any testing of room acoustic quality for speech or song that does not include this directivity will be subject to a systematic error. Fortunately, it is possible to simulate reasonably well the directivity of the human voice by using a glass fiber baffle with a small rectangular hole in front of the loudspeaker driver of a small loudspeaker box as illustrated in Figure 3.33.

Speech Intelligibility

It is of great interest to understand how the relative attenuation of various speech spectrum components affects the *perception of speech*. Many situations critically involve correctly understood speech, such as

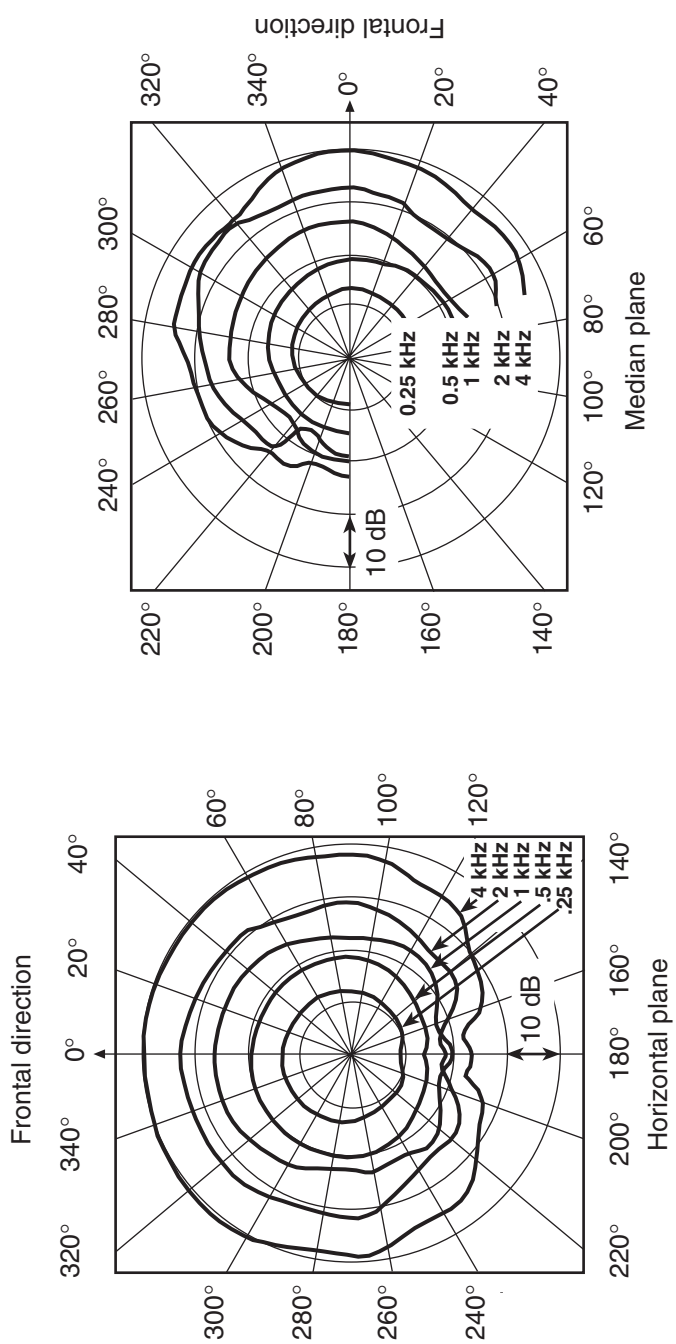


Figure 3.32 Polar plots for voice directivity in the horizontal and median planes. (After Ref. 3.10)



Figure 3.33 Using a commercial 2–way loudspeaker having similar dimensions as the human head (left), it was possible to achieve suitable directivity by attaching a baffle with a 30 by 10 mm hole in front of the low frequency unit (right). The baffle was made of 10 mm thick hard-pressed glass wool. The directivity that resulted was similar to that of the human voice.

classroom teaching, warning systems, or flight control systems, among others. The curves in Figure 3.34 show the relative influence of the various spectral components of speech on speech perception. One sees that most of the speech information is contained in the frequency range between 0.4 kHz and 6 kHz. Many transmission systems decrease this bandwidth even further; for example, the GSM system for mobile telephony uses a 4 kHz upper frequency limit.

It must be noted that speech intelligibility is not the same as speech (sound) quality. Speech quality is related to the result of linear and nonlinear distortion and the way the speech sounds.

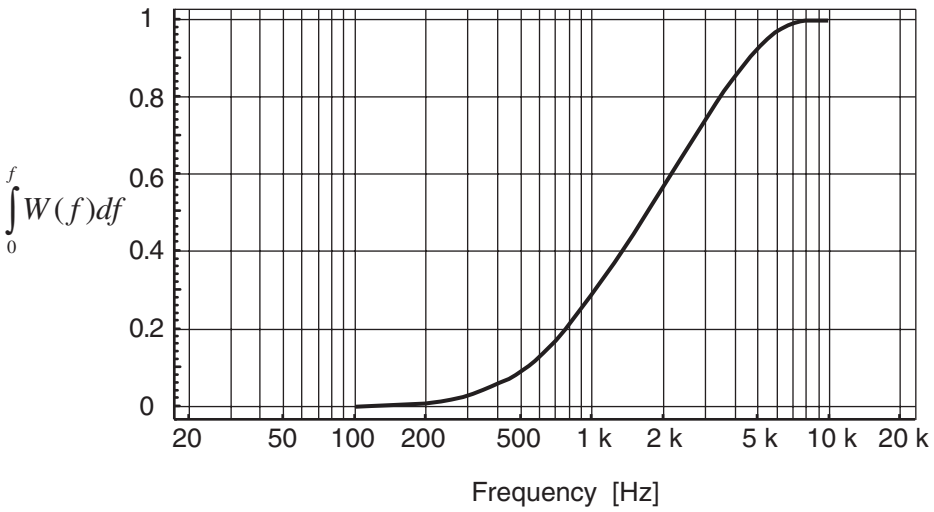


Figure 3.34 The integrated weight function for estimation of the importance of various parts of the spectrum on speech intelligibility. The region of largest slope is the most important for intelligibility. (After Ref. 3.11)

Depending on how *speech intelligibility* is defined and measured, one will obtain varying results. The intelligibility can be estimated by the percentage of correctly perceived phonemes, single syllable nonsense words, single syllable phonetically balanced words, or sentences. The choice of word lists, listener ability (hearing threshold, intellectual capacity), and presentation techniques all influence the intelligibility test results. In addition, the talker's voice spectrum, pronunciation, for example, will influence the test results. Figure 3.35 shows how the percentage of correctly understood syllables, words, or sentences varies with signal-to-noise ratio (S/N) for some different test items.

The percentage at which items are correctly understood is often called *articulation*, and it is usually necessary to specify which type of word material or sounds were used in the test—for example, by specifying sentence articulation. To assess the speech intelligibility it is necessary to use listeners, consequently the results will show large statistical uncertainty.

Objective Speech Intelligibility Assessment

Because of the difficulties, uncertainties, inconveniences, and costs involved in subjective speech intelligibility assessment, many metrics have been devised to estimate the influence of various changes to the speech spectrum and to the modulation of components of the speech spectrum. It is important to remember in using such metrics that they are only approximations and that the results obtained are usually specific for a certain test situation. Most of the metrics also do not even take the directionality of speech into account. One should always be cautious when using speech intelligibility metrics, particu-

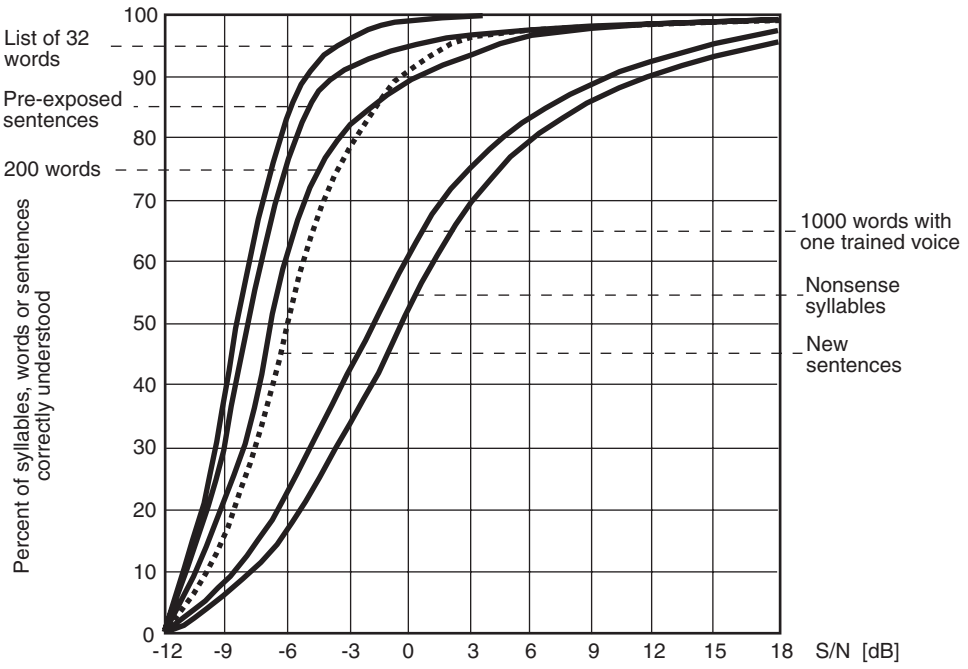


Figure 3.35 The influence of S/N ratio on speech intelligibility. (After Ref. 3.12)

larly within the field of room acoustics. Many metrics, however, function quite well for single channel, analog telephone communication situations.

The *articulation index* (AI) is a metric designed in the 1940s to estimate the quality of radio telephony and was intended to be used for single channel voice communication in the presence of noise. It can be estimated through listening tests, using the number of correctly understood voiced and unvoiced sounds according to a special formula. One can also use a table or nomogram by which the AI can be estimated from the S/N ratio. There is now a more modern and similar index, the *speech intelligibility index* (SII), defined by an ANSI standard. The SII is a simple way of roughly estimating the influence of noise on speech intelligibility.

Articulation loss of consonants (%ALcons) is still a popular metric among designers of speech reinforcement systems for stadiums and other sports arenas. It is calculated from knowledge of the level relationships between direct and reverberant sound and reverberation time.

Modulation damping (MD) is a simple, little used, but good metric based on the modulation characteristics of speech. The test signal consists of white noise that has been frequency weighted using an A filter and then chopped into *noise pulses*. The noise pulses consist of 70 ms of noise and 140 ms of silence. Because of this property, it can be easily generated and used to estimate the influence of both reverberation and noise on the quality of sound transmission. One simply measures the modulation index of the noise before transmission and after reception, then subtracts. The results using the modulation index show quite good agreement with those obtained by subjective speech intelligibility determination.

Speech transmission index (STI) uses the same principle but a more advanced approach than that of MD. STI is based on the transmission of a noise having a speech-like spectrum and a wide modulation spectrum. Essentially the modulation is checked for *98 carrier and modulation frequency combinations* as shown in Table 3.1. The result is then obtained by a weighed summation of the differences incurred by the transmission channel or room. As with MD, STI can be used to investigate the influence of both speech and reverberation. Because of the extensive subjective testing that accompanied the development of STI, it must be regarded as the best metric yet derived for speech intelligibility. Using a computer, the STI value can be determined from measurement of the impulse response of a system and the knowledge of the background noise spectrum. RASTI is a simplified metric similar to STI.

The relationships between STI and various types of speech intelligibility tests are shown in Figure 3.36. One problem with STI as a metric for the acoustical quality of auditoria, is that it (along with the other metrics discussed in this section) does not include the influence of some properties of binaural hearing. Comparing listening to natural speech in auditoria to STI measurement results, one quickly finds that very different speech quality situations can have approximately the same STI values. STI does not measure the sound quality aspects of speech, such as distortion, timbre, and such.

Finally, it is important to note that the metrics described have been developed primarily for testing of linear systems. Transmission systems using compression, perceptual coders (codecs), and the like, should also be tested with other appropriate test systems (see Chapter 16).

Table 3.1 The 96 combinations of carrier and modulation frequencies used in the calculation of STI. The gray squares show the combinations used for RASTI. Each combination is weighted using an empirically found coefficient.

Octave band carrier frequency [Hz]	Modulation frequency [Hz]													
	0.63	0.80	1.00	1.25	1.60	2.00	2.50	3.15	4.00	5.00	6.30	8.00	10.0	12.5
125														
250														
500														
1														
2														
4														
8														

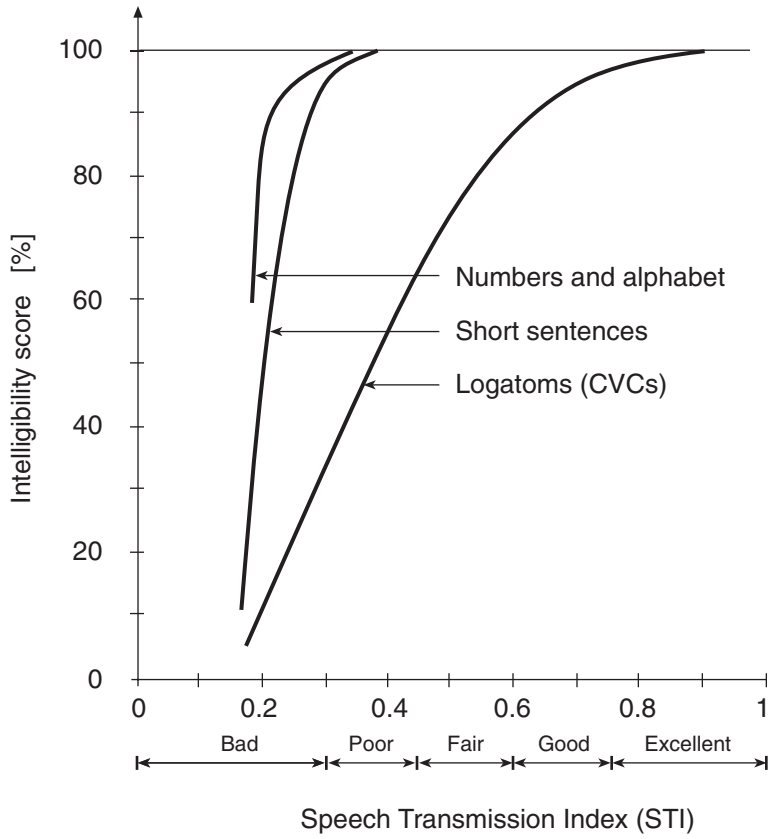


Figure 3.36 Comparison between STI results and the results of various speech intelligibility tests. (After Ref. 3.13)

3.9 PROBLEMS

3.1 You are listening to a 1 kHz Morse code sine signal that is contaminated by white noise.

Task: Will the use of a 100–Hz-wide bandpass filter centered on 1 kHz improve the audibility of the tone?

3.2 Figure 3.20 shows the hearing threshold shift due to masking.

Task: Determine, using Figure 3.20, how much the audibility of a 2 kHz tone is reduced when an additional 1 kHz masker tone having a sound pressure level L_p = 80 dB is added.

3.3

Task: Determine the sound pressure level that a 1 kHz tone must have to be audible in background noise having octave band levels given in the table. Assume that audibility requires that the tone have at

least the same level as the noise inside the critical band at 1 kHz. Use the power spectrum level, PSL, discussed in Problem 2.10.

Octave band	250	500	1k	2k	4k	8k	[Hz]
SPL of noise	75	80	85	80	75	70	[dB]

3.4 Because the sensitivity of hearing is different depending on sound frequency, the effective frequency response of hearing will vary with loudness level and spectral content.

Task: Determine, using the curves for equal loudness level in Figure 3.12, the perceived change in frequency response if we listen to a tone at a loudness level of 40 phon and raise the sound pressure level by 30 dB.

3.5 The diagram in Figure 3.21 shows the widths of the critical bands at various frequencies.

Tasks:

a. Calculate the widths of the critical bands.
b. Do the widths of the critical bands correspond to octave bands or third octave bands?
c. Carry through the calculations for the following center frequencies; 0.1, 0.2, 0.4, 1, 2, 4, and 10 kHz.

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Basic Room Acoustics 4

4.1 INTRODUCTION

Room acoustics involves a number of issues. Room acoustics is related to architecture, art, music, physics, engineering, and psychology, for example. It is common to subdivide room acoustics into two branches: (1) *technical room acoustics* and (2) *psychological room acoustics*.

Technical room acoustics can be further subdivided into *geometrical*, *statistical*, and *wave theory* room acoustics, which are three different ways of analyzing wave propagation in a confined space. Finally, there is always the possibility of using numerical approaches, such as those offered by the *finite element method* and various *finite time difference methods*.

Geometrical acoustics is similar to ray optics and is consequently a high-frequency approximation using concepts, such as mirrors and rays. For the approach used in geometrical acoustics to be applicable, surfaces should be much larger than the wavelengths of sound. Geometrical acoustics is the basis for many software models of sound propagation in rooms as well as outdoors.

Using *wave theory based room acoustics*, one can obtain exact mathematical solutions to problems in room acoustics, provided one can describe the room and its boundary conditions in a mathematically relevant way and that the problem is simple enough for analytical solutions to be feasible. This is seldom the case when dealing with practical problems in room acoustics. Therefore, the wave theory approach is primarily used to study the fundamental principles of behavior of sound waves in the low-frequency region of room acoustics for which the behavior of individual modes can be studied. It is used for studies of simple idealized room shapes, such as rectangular, cylindrical, and spherical shapes that approximate those of the real room and can be used as a point of departure for a further analysis.

Statistical room acoustics is based on the analysis of how the energy carried by the sound field is distributed in or between rooms and how the energy is absorbed by the sound-absorbing surfaces and objects of the room. Statistical room acoustics can only be used to study the average, approximately stationary behavior of sound under static or quasi-static conditions.

Because of these limitations, the solutions approach offered by statistical room acoustics is particularly useful to study sound due to continuous, wide-frequency range noise sources, such as ventilation, air conditioning, engines, and machinery. For quasistationary sound types, such as speech and music, statistical room acoustics must be used with great care to avoid drawing unfounded conclusions. It is also important to understand the limitations of the statistical approach and that the sound fields in real rooms are usually not ideal from the viewpoint of statistical room acoustics. Statistical room acoustics is a branch of *statistical energy analysis*, and it creates a possibility to study the interaction between sound in panels and rooms.

The purpose of *psychological room acoustics* is to document and explain how humans hear, perceive, and judge sound fields, both simple sound fields in the laboratory and the complex sound fields found in common rooms for speech and music. One major part of psychological room acoustics deals with the mapping of the perceived and judged room acoustics on physically measurable room properties and associated metrics. Psychological room acoustics is the topic of Chapter 5.

All of these approaches need to be understood to be able to successfully design and optimize the acoustic conditions in rooms for speech and music, ranging from small studios to large auditoria. The principles of adjusting and optimizing the room acoustic conditions are sometimes called *applied room acoustics*. In applied room acoustics *physical modeling* and numerical methods, such as the *ray tracing*, *mirror image*, and *finite element* methods are used to obtain better approximate solutions to room acoustic problems. Excellent references are 4.1 and 4.2. Modeling and numerical methods are discussed in Chapter 6.

4.2 GEOMETRICAL ROOM ACOUSTICS

In geometrical room acoustics, we use the same modeling principles as in geometrical optics—the ray tracing and the mirror images methods—to study the propagation of sound. These methods are then often extended by further acoustically relevant approximations. Geometrical room acoustics is useful to study the behavior of sound both indoors and outdoors.

The following starting approximations and assumptions are made in geometrical acoustics:

- Sources and receivers are infinitesimal and omnidirectional
- Sound propagation velocity is constant in the volume
- Surfaces have dimensions much larger than the wavelengths of sound considered
- Surfaces are smooth and nonscattering
- Surfaces have locally reacting reflection properties
- Surfaces have no resonant absorbers (real wall impedance)
- Sources radiate insensitive to surroundings
- Source signal is assumed to be a delta pulse
- Interference effects are neglected
- The phase part of the reflection coefficients is neglected
- Power summation of mirror image or ray contributions is used
- If diffraction and scattering are included in the model they are handled by angularly dependent reflection coefficients.

The Wavelength Criterion

A major problem in dealing with room acoustics is the large wavelength range. The lowest audible frequencies at 20 Hz have wavelengths of about 17 meters, while the highest at 20 kHz have wavelengths of approximately 0.017 meters.

This means there are few surfaces in buildings that can be considered perfectly reflecting in the sense of geometrical acoustics, at least in the high-frequency range. At low frequencies, the room dimensions may simply be too small for the room surfaces to be large compared to wavelength, whereas

at high frequencies, the irregularities caused by poor workmanship and unavoidable lack of precision are so large as to make the surfaces more scattering than mirroring.

In practice, it is customary to consider surfaces with random peak irregularities smaller than $\frac{1}{16}$ of a wavelength as flat and thus acting as ideal mirrors. It is also reasonable to consider surfaces having dimensions at least three wavelengths long as large compared to wavelength.

The essential frequency range for many speech applications is 0.5 to 4 kHz—that is, a wavelength range of 0.7 to 0.1 m. In dealing with music, one has to be concerned with a greater frequency range, and then one normally has to do a geometrical acoustics analysis separately for various frequency bands in the frequency range of interest, for example, in octave-wide frequency bands.

Even though geometrical acoustics is incorrect in many respects, it is still a useful approximation as it applies an intuitive time/space approach, yielding results that are easily understood and the consequences of which can be easily demonstrated to nonprofessionals. With experience, it is usually possible to know what kind of errors to expect and to consider the errors in design.

Fresnel Zones

Huygen's principle states that each point on a wavefront may be regarded as a source point in calculating the next wavefront. Studying the results of the experiment shown in Figure 4.1, one finds that this is indeed the case. In this experiment, the source is omnidirectional over the angle subtended by the circular, concentric *zone* discs. The microphone senses the sound being reflected by the discs. Varying

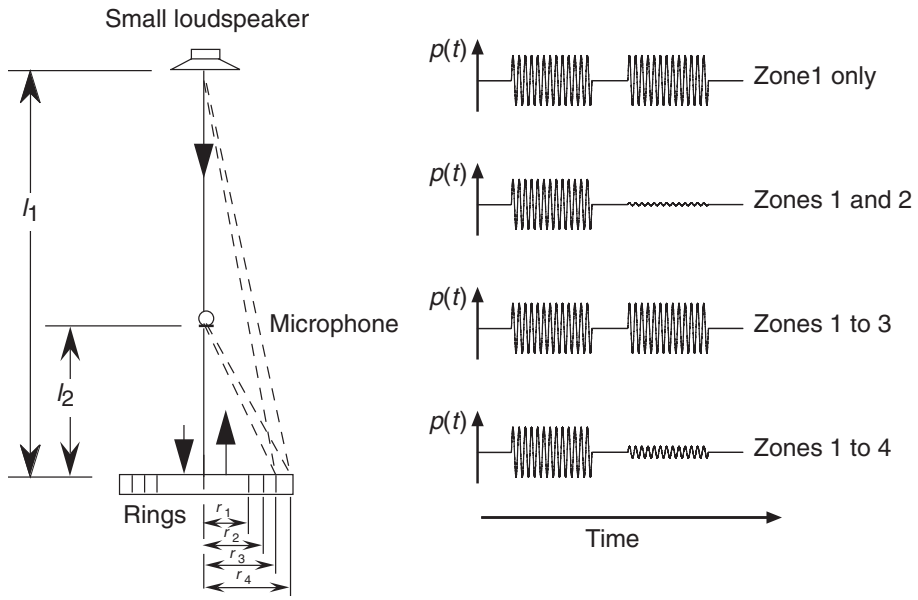


Figure 4.1 Demonstration of the influence of Fresnel zones on reflection by circular disks. The experimental setup consisting of a loudspeaker driver, a microphone, and a Fresnel zone plate using concentric reflecting rings (left). Response from the microphone for a tone burst for various numbers of reflecting Fresnel zones (right). (After Ref. 4.1)

the diameter of the disc assembly as r_1, r_2, and so on, one finds by measurement that the tone bursts will behave as shown in this figure.

The resulting sound pressure at the microphone is due to constructive and destructive interference between the sound pressure contributions by the various discs. The interference can be analyzed by studying the phase and amplitude of the reflected sound contributions by phasor analysis. It is easy to analyze the conditions for out-of-phase and in-phase reflection; that is, when the distance differences between the loudspeaker driver and the microphone—via points (or rather circles) on the disc—are multiples n of one half of the wavelength λ:

$$\Delta l = \sqrt{l_1^2 + r_n^2} + \sqrt{l_2^2 + r_n^2} - l_1 - l_2 = n\frac{\lambda}{2} \quad (n = 1, 2, 3, ...) \tag{4.1}$$

If one assumes l_1, $l_2 \gg r_n$, the radii r_n are obtained as:

$$r_n^2 \approx \frac{n\lambda}{\frac{1}{l_1} + \frac{1}{l_2}} \tag{4.2}$$

The zones created between the circles of these radii are called Fresnel zones. As the disc grows larger, the interference between the outermost zones becomes weaker. A useful approach is consequently to only consider the reflection by the innermost zone. The contributions by the outer rim sources can be considered approximately as the diffraction by the edges of the reflecting surface. This means that by this principle a surface will reflect, or mirror, irrespective of its size, once the size is more than some set limit and the *reflection point* is sufficiently within the surface area.

If we study an example of the distribution of the zone areas over a rectangular surface, such as that shown in Figure 4.2, we find that the zones 1b to 6 quickly cancel each other. The higher the frequency, the smaller the center zone has to be to reflect sound efficiently.

We can formulate a condition for how small the surface can be to function as an acceptable reflector of a plane-wave of sound (that is, l_1 very large) for a certain wavelength and receiver distance. It is reasonable to assume that the surface must contain at least the inner half zone. This results in the approximate condition:

$$l_{\text{limit}} \approx \frac{d^2}{2\lambda} \tag{4.3}$$

where d is the smallest surface dimension and l_{limit} is the distance to the observation point.

This condition is sometimes used when one needs to design sound reflectors for auditoria. Of course, a reflector also has to have sufficient surface mass (mass per unit area) so that it remains in place and is not moved by the sound field. A small object can act as a diffuser if it is small compared to the wavelength; the Fresnel zone criterion implies that a small object cannot mirror sound.

Ray Tracing and Mirroring

Assume a monopole sound source emitting sound energy in the form of rays. (Other related possibilities are emission of sound particles, beams, and cones.) Assign a certain fraction of the power radiated by the sound source to each of the rays, inversely proportional to the number of rays.

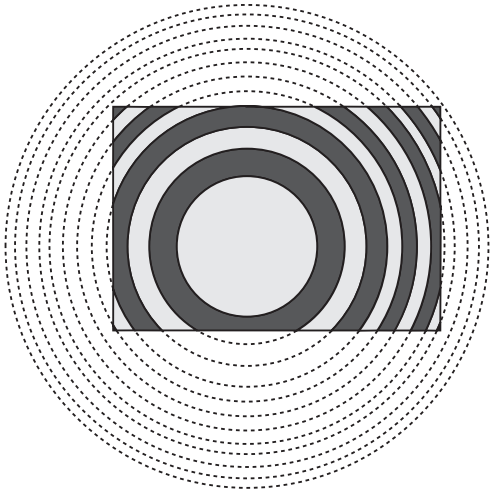


Figure 4.2 Illustration of the principle of Fresnel zones on a rectangular plate when the geometrical reflection point is offset from the center. The dark and light patches reflect out-of-phase with one another. When reflection and incidence occurs at an angle, the circular zone pattern will become elliptical. (After Ref. 4.1)

The path of each ray is determined by *Fermat's principle* that states that rays follow the fastest path, usually a straight line. During its progress, a ray may be reflected as well as bent, depending on the presence of sound-reflecting objects or changes in propagation conditions, such as wind or temperature and impedance gradients as mentioned in Chapter 2. Mirroring implies that the incident and reflecting rays are in the same plane as the normal to the surface and that the angles of incidence and reflection are equal. In the case of a gradual impedance change, the angles of incidence and transmission are related by Snell's law, mentioned in Chapter 2.

To find the rays, we can either send out rays and follow each ray along its path, or we can find the mirror images and then draw rays from the mirror images of the source to the observation point. The case of two opposing mirrors in Figure 4.3 is elementary. If one has several surfaces that are not parallel, one must take more complicated mirroring into account, as shown in Figure 4.4. In this case, we have four different mirror image sources since the ray has been reflected twice (note that two of the image sources have identical locations). The number of times a reflection has been mirrored, or a ray path reflected, is called the mirror image or ray *reflection order*. Figure 4.5 shows an example of the distribution of the mirror images of a two-dimensional rectangular room.

Geometrical acoustics is primarily useful for the study of the behavior of the early sound distribution in rooms. In practical rooms, the unevenness of the walls, the inaccuracies of wall positions, and the sheer multitude of rays or images, make it difficult to analyze the conditions by hand after the first or second order of reflection. Higher reflection orders have to be handled numerically by computerized calculation, which will be further discussed in Chapter 6.

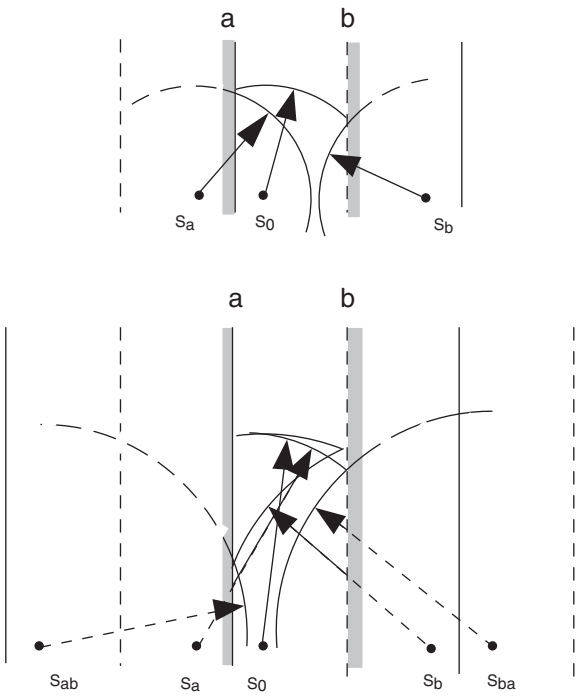


Figure 4.3 Mirroring by two parallel surfaces—a and b. The upper figure shows the sound source and the first order mirror image sources. The lower figure shows conditions after more time and the additional second order mirror images necessary to study the wavefronts. For each time instant, the radius of curvature is determined by the speed of sound and the time from the start of the sound.

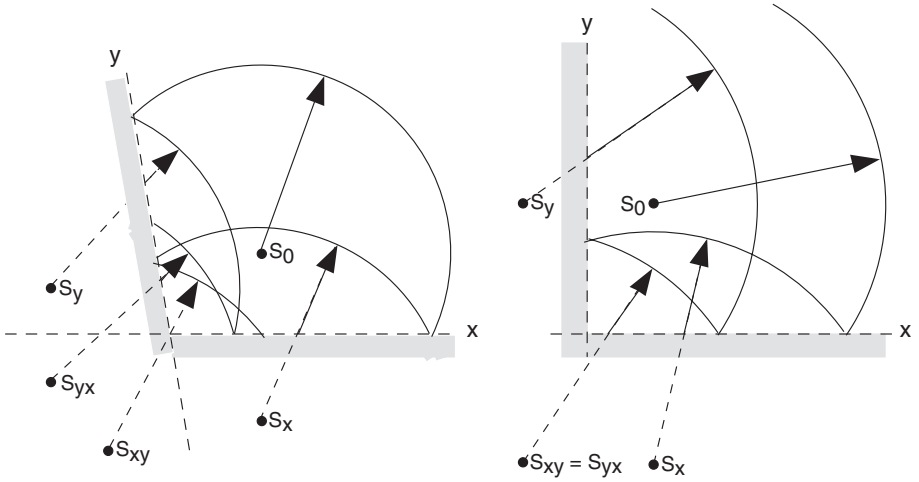


Figure 4.4 Mirroring of sound at the corners of intersecting surfaces, showing first and second order image sources.

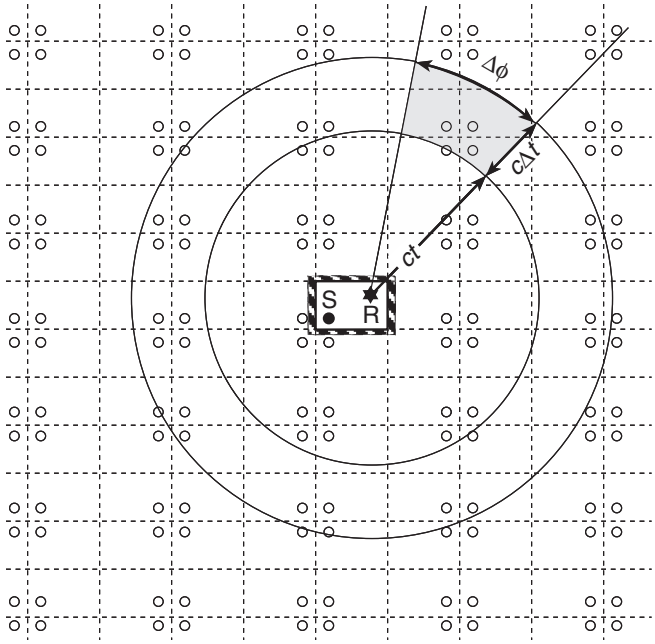


Figure 4.5 Mirroring by the walls of a rectangular room. The sound source is marked by the filled circle at S and the receiver is positioned at the star at R. The figure shows the location of mirror sources of various orders (empty circles). Only the sources in the grey zone contribute to the impulse response from the angle segment $\Delta\phi$ during the time segment Δt.

When surfaces are curved, ray tracing is intuitively simpler to use than mirror imaging. For a curved surface, the surface may be subdivided into smaller surface patches and analyzed as previously, or one can use the mirroring laws of optics.

In practice, room impulse responses are not as simple as one might be led to believe by simple geometrical optics. As shown by the example in Figure 4.6, the measured impulse response of a real room is complicated. This is due to the presence of scattering, diffraction, complex surface impedance, source and receiver impulse responses, for example. The energy behavior, however, can be quite well simulated. This is usually done by calculating the time integral of effective value of the sound pressure.

4.3 STATISTICAL ROOM ACOUSTICS

Diffuse Sound Field

In using statistical room acoustics, we idealize the sound field in a room so that we can limit ourselves to the study of the attack, steady state, and decay of sound energy. A prerequisite for this approach is that the sound has a wide spectrum and that we are not interested in the conditions at a specific frequency but rather in a frequency band wide enough to contain many frequencies.

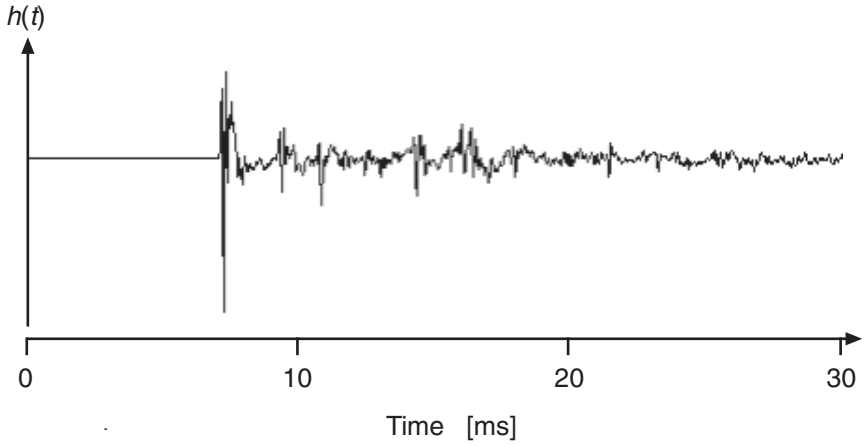


Figure 4.6 Example of the early part of a typical measured impulse response $h(t)$ of a living room (including the measurement equipment). The time before the initial transient at about $t = 7$ ms corresponds to the time necessary for the sound to propagate from the sound source to the microphone.

Geometrical acoustics shows that all the contributions of the various reflections, after some time, set up a complicated sound field in the room. In this sound field—provided some simple conditions are met—the sound waves will come from many directions; will have bounced off many walls and objects; and will, on the average, be equally strong from all directions. The sound field will be *diffuse*.

An ideally diffuse sound field is characterized by the following physical properties:

- Energy density is equal at all places.
- All directions of incidence are equally probable.
- Incident sound has random phase.

When the walls of the room under study have low average sound absorption, the sound absorption is well distributed over the different walls, and the room does not have an *extreme* shape, the sound field in the room will have properties similar to those of an ideally *diffuse sound field*.

In reality, there are no ideally diffuse sound fields. When studying the sound fields of rooms, one can always find places where the sound is not diffuse, for example, close to walls where the reflected sound is well correlated to the incident sound. If there are major wall areas that have markedly higher sound absorption and size than others, the conditions are also not met. In practical engineering, however, one can often assume the sound field diffuse at positions being more than one-half wavelength away from wall surfaces.

For steady-state conditions, the sound field is best considered as a superposition of resonant room modes. Another way of formulating the sound field properties necessary for diffusitivity is then to say that the sound field under study has to have a sufficiently large frequency range for it to contain many modes, that all modes are approximately equally damped, and that the room geometry is not extreme. We will study this way of expressing the requirements in the section on the wave theoretical approach to room acoustics, later in this chapter.

Attack and Rise Time

When a sound source starts to emit sound in a quiet room, the energy in the room will increase. The process of increase of energy density is known as the *sound attack*. The energy density, which is related to the effective value of sound pressure, is determined by the energy in the room and the room volume.

One can show that the rate of energy increase in the room depends on both the power radiated by the sound source and on the sound that is absorbed—by the surfaces of the room and by the objects present in the room.

The sound absorption of a surface was discussed in Chapter 1 (see Equation 1.58) and is specified by mean sound-absorption coefficient, α, characteristic for the surface at some frequency. The sound-absorption coefficient is related to the incident sound power, W_{inc}, and absorbed power, W_{abs}, as:

$$\alpha = \frac{W_{abs}}{W_{inc}} \tag{4.4}$$

It is common to find the value of α expressed in percentage in various tables of material data. The absorption area of a surface, A, is obtained by multiplying the sound absorption coefficient by the surface area and is expressed in units of m^2S metric sabin. (Note that the unit sabin refers to the imperial system). If there is a sound source in the room that is activated, then, during the attack, the energy in the room increases as:

$$E = E_0\left(1 - e^{-\frac{cA}{4V}t}\right) \tag{4.5}$$

where E_0 is the steady-state energy density that will ultimately be reached.

Later we will note that the attack process is complementary to the decay process. We will also derive an expression for the steady-state energy density as a function of room volume and sound absorption.

Reverberation Time

The rate at which the energy density diminishes after the sound source has been switched off is known as the *rate of decay*. It is often essential to know the rate of decay for sound in a room as a function of the acoustic properties of the room. The rate is sometimes described by a decay constant, but it is more common to express the rate indirectly by specifying the reverberation time of the room.

The *reverberation time, T* (also T_{60} or RT), is defined as the time it takes for the *sound energy density* to drop to one-millionth of its previous value. This corresponds to a sound-pressure level decrease by 60 dB. By using knowledge regarding the room volume, the sound-absorption coefficients of the room surfaces, and their respective sizes, it is possible to calculate the reverberation time.

By using geometrical acoustics, it is possible to trace the ray paths or calculate the mirror image distribution and study the decay process at any position in the room more in detail and also calculate the *local* reverberation time (within the limits of the method). Since the sound field is never ideally diffuse, the reverberation time will differ from one location in the room to the next. Even quite small differences in reverberation time are perceptible. One usually tries to calculate reverberation time to an accuracy of 0.05 s in room acoustics design. Note that it is difficult to measure the reverberation time over a 60 dB span (see Figure 5.16) in practice, particularly at low frequencies.

Examples of the determination of T are shown in Figure 4.7. Generally, one is limited to estimating the decay over a 30 to 40 dB span from which the reverberation time is then extrapolated. Because of this, it is common to give the span information by providing the start and stop levels, for example, for a span extending from −5 dB to −35 dB below start level as $T_{-5,-35}$ or $RT_{-5,-35}$.

In some cases the reverberation curves may be *dual-slope* or *multiple-slope*; that is, show two or more nearly linear slope regions. This behavior can be the result of different groups of modes having different decay constants or the sound field at the observation point being composed of sound coming from different volumes. When listening to reproduced sound, the effective reverberation curves will usually be dual-slope because of the presence of listening room reverberation along with the recorded reverberation.

Typically, such multislope reverberation decay curves can be found for rooms that have one large surface that is more highly absorptive than the others. Dual-slope reverberation processes can also be found when two rooms are coupled to one another through an opening. An example of where such conditions can be met is in a proscenium theatre where the stage and the auditorium are acoustically coupled through the stage opening.

Sabine's Formula

In this section, we will derive a simple expression for the decay rate expressed by reverberation time, using the room volume and the sound-absorption area as variables. Assume that the sound-absorption properties of the walls are the same for all walls, given by a mean sound-absorption coefficient α. Let a wave *packet* radiate at time $t = 0$. Each time the wave is reflected by a wall surface, it will lose a part α of its energy as heat or by transmission to some other acoustic system, and only a part $(1 - \alpha)$ will

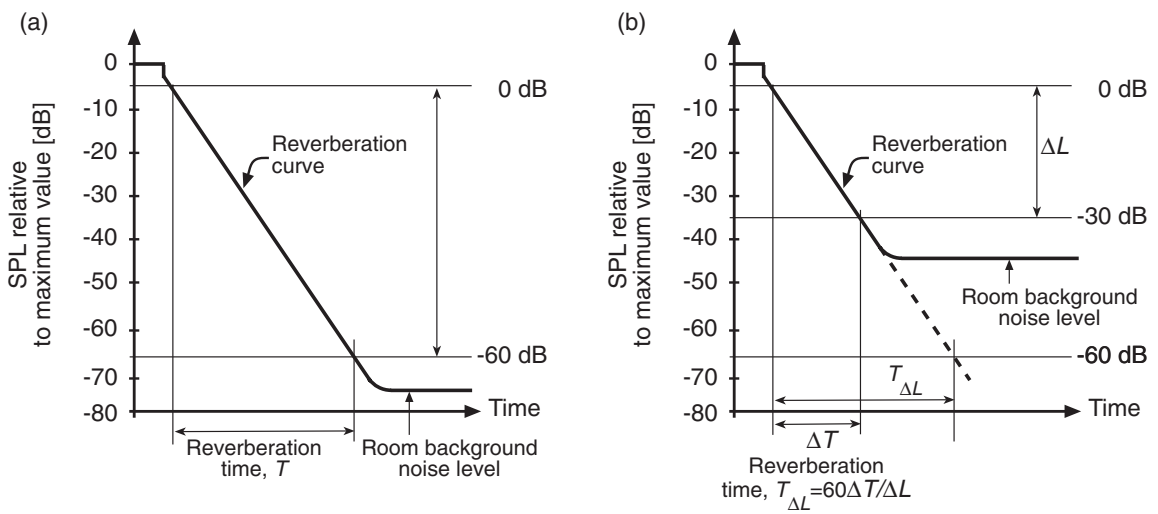


Figure 4.7 Examples of the determination of reverberation time (see also Figure 5.16). To ensure that the reverberant sound is diffuse the start of the measurement is usually 5 dB below the steady-state value (a). Determination of T when room background noise level is high (b). Real reverberation curves will not be characterized by smooth, constant slope, particularly in narrow, low-frequency octave and narrower bands.

remain. Another power loss mechanism is the attenuation of sound by propagation losses in air, which was studied in Chapter 1.

The length of the path between two successive ray-wall collisions can be represented by a mean length, usually called the *mean-free-path length* l_m, which depends on the properties of the room, such as its geometry, the diffusitivity of the walls, and the presence of sound diffusing objects. One can show that the mean-free-path length l_m in a volume V, having a total surface area S, is on the average given by:

$$l_m \approx \frac{4V}{S} \tag{4.6}$$

The distribution of path lengths will depend on the room geometry as shown in Figure 4.8. An even distribution of path lengths is typical of rooms that do not have an *extreme* shape or distribution of sound-absorptive materials over their surfaces.

It is convenient to characterize the decay rate of the energy by a real-valued decay constant, δ, which we can associate with the reverberation time T. The energy decay follows $\mathrm{E}(t) = \mathrm{E}_0 \mathrm{e}^{-2\delta t}$. Since the wave loses a part $(1 - \alpha)$ of its energy each time it is reflected, one can write the remaining energy at each time instant as:

$$E_{\text{wave}} \propto (1-\alpha)^{\frac{ct}{l_m}} e^{-mct} = e^{-2\delta t} \tag{4.7}$$

where the previously mentioned propagation losses in air have been included using the constant m (see Figure 4.9 or Figure 1.3).

Taking the logarithm of both sides in Equation 4.7 one finds:

$$2\delta = mc - \frac{c}{l_m}\ln(1-\alpha) \tag{4.8}$$

The definition of reverberation time gives the relationship between T and δ as:

$$T = \frac{3\ln(10)}{\delta} \approx \frac{6.91}{\delta} \tag{4.9}$$

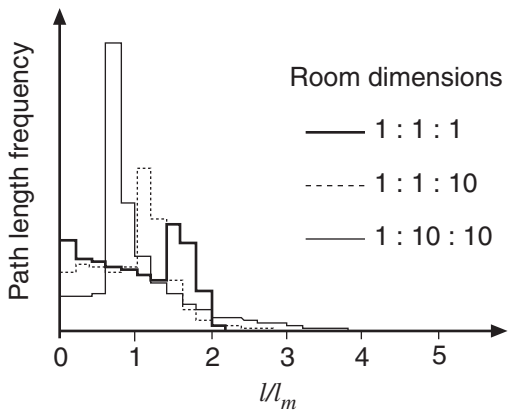


Figure 4.8 Some distributions of the path lengths l in rectangular rooms of different proportions and the respective mean-free-path lengths l_m. Note that the long paths are infrequent. (After Ref. 4.2)

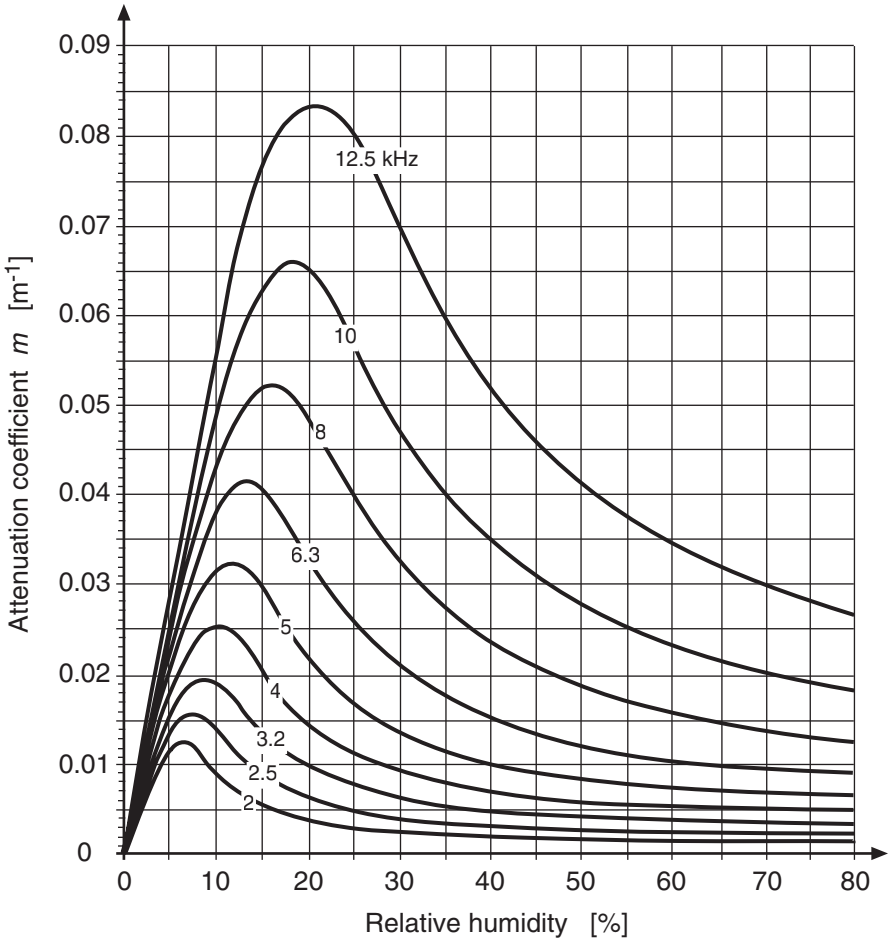


Figure 4.9 Values of the attenuation coefficient *m* for air at a temperature of 20°C, for various values of relative humidity and frequency (in kHz). (After Ref. 4.3)

Dependence of reverberation time on room volume and the sound-absorption coefficient is then obtained as:

$$T = \frac{24\ln(10)}{c_0}\frac{V}{4mV - S\ln(1-\alpha)} \tag{4.10}$$

It is now appropriate to introduce the concepts of *absorption area*, *equivalent absorption area*, *Sabine absorption,* and *room absorption*.

The *absorption area* is simply the product of absorption coefficient α and physical area *S*:

$$A = \alpha S \tag{4.11}$$

Note that this expression assumes that a surface absorbs sound in the same way no matter what size and placement.

One can show, by both theory and experiment, that this is not the case (see Chapter 6). Not only are certain dimensions relative to wavelength characterized by higher sound absorption, but splitting a large sound-absorbing area into small patches usually increases the diffusitivity of the sound field. In this way, it increases the effective sound absorption. This is the result of the mean-free-path length becoming shorter due to the more frequent reflections.

The *equivalent absorption area* is a concept used to characterize the sound absorption of irregularly shaped objects for which it is difficult to specify the area or the absorption coefficient. Examples of such objects are humans and chairs. The equivalent absorption area is defined as the absorption area that would absorb the same sound power as the object in a room with a diffuse sound field, such as a *reverberation chamber* (see Chapter 17).

The *room absorption* is the sum of the absorption by all room surfaces and objects and the sound absorption of the air (see Figure 4.9):

$$A_S = 4mV - S\ln(1-\alpha) + \sum_j A_j \approx 4mV + A \tag{4.12}$$

The room absorption A_S represents the total sound absorption of the room and is expressed in units of m^2S. The unit metric sabin is sometimes used.

The total surface area of the room S_t, including wall surfaces and the surface areas of other large objects, such as sound-absorptive screens and audience area, is obtained as:

$$S_t = \sum_i S_i \tag{4.13}$$

It is practical to use a mean sound-absorption coefficient α_m defined as:

$$\alpha_m = \frac{\sum_i \alpha_i S_i + \sum_j A_j}{\sum_i S_i} \tag{4.14}$$

Using this definition, we can the calculate the reverberation time using *Sabine's formula*:

$$T = \frac{24\ln(10)}{c_0} \frac{V}{4mV + \alpha_m S_t} \tag{4.15}$$

Equation 4.12 is named after the Harvard scientist W. C. Sabine who was the first to formulate it. The slightly more exact formula in Equation 4.10 is called *Eyring's formula*. The Sabine absorption is the absorption area given by Sabine's formula, by means of the measurement of the reverberation time of the room.

In the case of small rooms and low frequencies, the sound absorption by air can usually be neglected. In large spaces, such as concert halls and factory spaces, having volumes more than 10,000 m^3, the sound absorption by air is usually the dominant sound-absorption mechanism at high frequencies.

Assuming that the losses incurred by air sound absorption are much smaller than the losses at the walls, $m \ll \alpha/l_m$, one obtains a common approximation to the decay constant as:

$$\delta \approx \frac{cS\alpha}{8V} \tag{4.16}$$

Sabine's formula is commonly expressed in an approximate way, using SI units, as:

$$T \approx 0.161 \frac{V}{A_S} \tag{4.17}$$

One can usually use Sabine's formula under the following approximate conditions for diffuse sound fields:

- The room averaged sound-absorption coefficient α is less than 0.3.
- The sound-absorbing surfaces are approximately evenly distributed over the room.
- The shape of the room is not extreme.

Reverberation Curves

The most common metric for characterizing the suitability or quality of a room for a certain speech or music purpose is the reverberation time.

Central to the determination of the reverberation time is the reverberation curve, as discussed earlier. For practical reasons, the reverberation curve is usually studied as the decay of sound pressure level versus time, as shown in Figure 4.7. In the case of a single, exponentially damped oscillation, the curve would be a straight line, the lower part of which would be obscured by noise.

If the decaying signal is noise or otherwise contains a number of frequency components, the reverberation curve becomes wiggly simply because of the summation of the different tones at each instant. In practice, the different frequencies excite different room resonances that typically have unequal damping and reverberation times. These effects make the reverberation curve appear as a *noisy* line, but the latter condition leads to the reverberation curve having double or more slopes. In the low-modal density and low-frequency region, the reverberation curves may show effects of beats because of interference between strongly excited modes close in frequency.

A better way to measure the reverberation time than by direct approximation to the decay curve to evaluate the reverberation process using a method called *backward integration* or Schroeder's method. This method makes the determination of reverberation time slightly more precise. An example of such a curve is shown in Figure 5.14.

In any case, the determination of reverberation time is prone to vary depending on the method used. The ISO 3382 standard gives details on how the reverberation time of an auditorium should be measured. The ISO 354 standard details the measurement of sound absorption in a reverberation chamber. The measurement of sound power in a reverberation chamber s detailed in the ISO 3740 standard.

Sound Pressure Level and Sound Power Level

The sound field in a room acts as a store of acoustic energy. Measurement of the sound power radiated by a sound source is commonly made by measuring the sound pressure level and the room absorption.

As discussed previously, the energy in a room will asymptotically reach a steady-state value, if the sound source is allowed to act indefinitely. We have already noted that the sound pressure level in the

ideal diffuse field is constant in a room. Close to the sound source and walls, the sound field is not diffuse, and the sound pressure level will be higher.

We can consider the resulting sound pressure at a point in a room with a noise sound source to be composed of two uncorrelated parts, one due to the direct field and one due to the reverberant, diffuse field. If the two parts are sufficiently uncorrelated (sufficiently wide-band noise) the effective values of each can be added to give the total effective value.

If we assume a monopole sound source in free field, the effective value of sound pressure due to the sound source as a function of distance r must be:

$$\tilde{p}_{\text{dir}}^2 = Z_0 \frac{W_+}{4\pi r^2} \tag{4.18}$$

If the sound source has been active a long time, at least as long as the reverberation time, quasi-static conditions will have been reached. Under static conditions, the power radiated by the sound source will increase the energy in the room until it has reached a level where there will be an equilibrium between the power loss due to room absorption W_-, and the power supplied by the sound source, W_+.

When the sound source is switched off, the energy in the sound field will drop as a result of the sound absorption by the surfaces and objects in the room. Assume the sound source to be switched off at time $t = 0$. The energy of the sound field will then diminish as:

$$E(t) = E_0 e^{-2\delta t} \tag{4.19}$$

It will lose power W_- according to:

$$W_- = -\frac{dE}{dt} = 2\delta E_0 e^{-2\delta t} \tag{4.20}$$

Using our previously defined decay constant, we find that:

$$2\delta = \frac{cA_S}{4V} \tag{4.21}$$

where we have used Equations 4.6 and 4.8 and assumed $\alpha \ll 1$.

Before the start of the decay, at time $t = 0$, there must have been a power balance; that is:

$$W_+ = W_- \tag{4.22}$$

This gives the static energy E_0 as a function of the input power as:

$$W_+ = 2\delta E_0 = \frac{cA_S}{4V} E_0 = \frac{cA_S}{4} \frac{E_0}{V} \tag{4.23}$$

The relationship between energy density P and intensity I in a sound field can be shown to be:

$$I = Pc \tag{4.24}$$

where the energy density in this case simply is the stored acoustic energy divided by room volume:

$$P = \frac{E}{V} \tag{4.25}$$

The relationship between the intensity and the mean square value of the sound pressure in a plane wave is given by Equation 1.37, and the corresponding energy density by Equation 1.38. The energy density relationship, derived for the plane wave, also holds in a diffuse sound field, since we at any point can assume the sound field to be composed of plane-waves with random angles of incidence. The relationship between the mean square sound pressure and the energy density is:

$$\frac{<\tilde{p}_{\text{diff}}^2>}{Z_0} = \frac{E_0}{V}c \tag{4.26}$$

The brackets < > indicate that we mean the *spatial average of the mean square sound pressure,* that is, averaged over room volume.

Using Equation 4.23, we then find:

$$W_+ = \frac{c_0 A_S}{4V}\frac{V<\tilde{p}_{\text{diff}}^2>}{\rho_0 c_0^2} \tag{4.27}$$

Finally, we obtain the effective value of the static sound pressure in the diffuse field as a function of sound source input power as:

$$<\tilde{p}_{\text{diff}}^2> = Z_0\frac{W_+}{\left(A_S/4\right)} \tag{4.28}$$

Adding the two effective values for the direct field and the diffuse field, we obtain the total effective value as:

$$\tilde{p}_{\text{tot}}^2 = W_+ Z_0\left(\frac{1}{4\pi r^2}+\frac{4}{A_S}\right) \tag{4.29}$$

As usual, it is more convenient to express the relationship by levels. Using the expressions 2.24 and 2.26 we obtain the desired and useful formula:

$$L_p = L_W + 10\log\left(\frac{1}{4\pi r^2}+\frac{4}{A_S}\right) \tag{4.30}$$

Equation 4.30 is the common formula to calculate the sound pressure level due to a sound source in the reverberant field in a room. This expression is depicted graphically in Figure 4.10 that shows how the sound pressure level drops by 6 dB per distance doubling close to the sound source. Far from the source, in the diffuse field, the sound pressure level will theoretically be constant. The distance at which the direct field and the diffuse field are equally strong is called the *reverberation radius*.

When dealing with a directional sound source, we need to take the directivity of the sound source into account in describing the sound pressure level distribution in the room. The *directivity factor* Γ is related to the directivity and the directivity function as:

$$\Gamma(\theta,\varphi) = D\cdot F^2(\theta,\varphi) \tag{4.31}$$

The directivity factor tells us how strong the radiated intensity compared to that of an omnidirectional source radiating the same acoustic power. In the main lobe of a directional sound source, such as a violin or a large loudspeaker, the intensity (and sound pressure level) will be much higher than in other directions. In the direction of high directivity, the sound from the loudspeaker will reach farther, which is useful in rooms with long reverberation times or in noisy spaces.

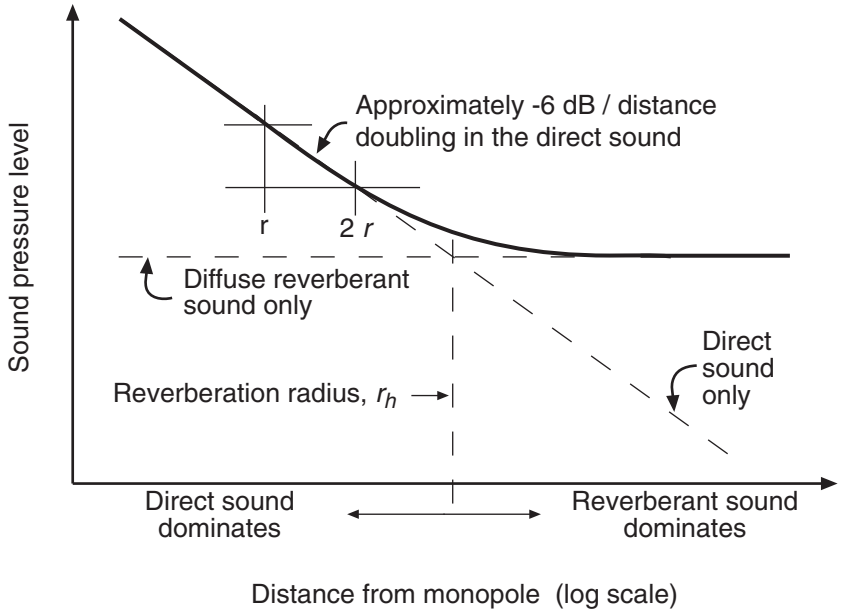


Figure 4.10 The theoretical behavior of the steady-state sound pressure level as a function of distance from a monopole source in a room with a broadband diffuse sound field, according to expression 4.24. The contributions of the direct field and the reverberant field will be equally strong at the reverberation radius r_h .Compare to the discussion and Figure 17.16 in Chapter 17.

One can show that an improved expression for the relationship between sound level as a function of angle in a reverberant space, $L_p(\theta, \varphi)$, and the sound power level of the source L_W, taking the directivity of the sound source and the diffusitivity of the sound field into account is:

$$L_p(\theta,\varphi) = L_W + 10\log\left(\frac{\Gamma(\theta,\varphi)}{4\pi r^2} + \frac{4\left(1-\alpha_m\right)}{A_S}\right) \tag{4.32}$$

The quantity RC = $A_S/(1-\alpha_m)$ is called the *room constant*.

As mentioned previously, there are no ideally diffuse sound fields. Close to walls and other sound-reflecting surfaces and objects, the sound pressure level can increase by up to 6 dB due to interference by the phase coherent incident and reflected waves. In corners, the increase can be even higher.

Power Incident on a Surface

The sound field is never perfectly diffuse. In a diffuse sound field, at any given time, one-half of the energy is moving in the opposite direction of the other half. Obviously, this cannot be the case close to a sound-absorbing surface.

The irradiation strength on a surface is defined as the incident acoustic power per unit area and is denoted B [W/m^2]. Because the surface area seen by a wave incident at an angle θ to a surface S is $S\cos(\theta)$, we need to average over all angles of incidence from 0 to π/2 to obtain the irradiation by a diffuse sound field. We find:

$$B = \frac{1}{2\pi}\int_0^{\pi/2}\int_0^{2\pi} I_{\text{inc}}\cos(\theta)\sin(\varphi)\,d\varphi\;d\theta = \frac{I_{\text{inc}}}{2} = \frac{I_{\text{diff}}}{4} = \frac{Pc_0}{4} \tag{4.33}$$

since the intensity seen by the surface I_{inc} is only one half of the diffuse field sound intensity I_{diff}.

Rooms Having Extreme Shape and/or Absorption Distribution

Because the sound field cannot be diffuse if the room has an extreme shape and/or uneven sound-absorption distribution over surfaces, Equations 4.30 and 4.32 are not applicable to rooms, such as open plan offices or large factory halls. In such spaces, the drop of sound pressure level as a function of distance from a (small) sound source typically behaves as shown by the curve in Figure 4.11.

Equations 4.30 and 4.32 indicate that it is possible to reduce the sound pressure level in the diffuse field of a room by increasing the room absorption. The noise reduction (NR) will be NR = 10 log(RC_2/RC_1) (dB). The level decrease will be large only if the initial room absorption is small. However, one often finds that the subjective improvement of room acoustics, due to added room absorption, is large even if the sound pressure level decrease is small. This subjective impression is due to the reduced reverberation time and the longer reverberation radius.

A room having little room absorption and correspondingly long reverberation time is often called a *live* or *hard* room. A test room for sound-absorption measurement is characterized by very rigid, glossy surfaces and is called a *reverberation chamber*.

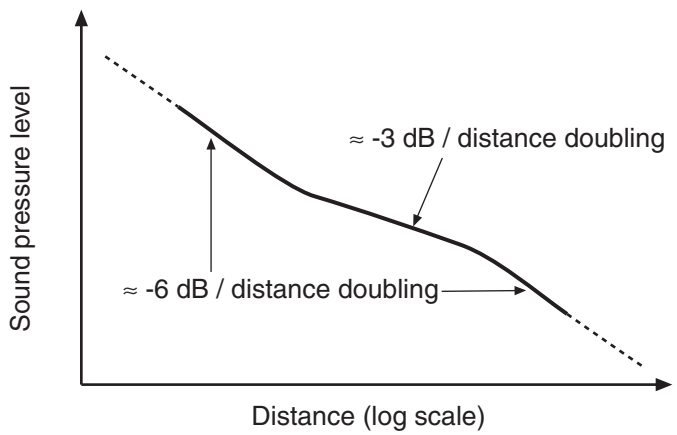


Figure 4.11 The reduction of sound pressure level by distance in a non-Sabine room can follow different distance laws. The type of sound level behavior shown here is typical of that from a small sound source in an open plan office or large factory space having low ceiling height.

A room that is characterized by short reverberation time is said to be *dead* or *dry*. For example, a test chamber to be used for electroacoustic measurements needs its walls to have a reflection coefficient $|\underline{r}| \leq 0.1$, an absorption coefficient $\alpha \geq 0.99$. Such a room is usually called an *anechoic chamber*.

Special purpose test chambers for sound power measurement of machinery may need a rigid floor surface. Such chambers that have only a half space of sound absorption are called *hemi-anechoic*.

4.4 WAVE THEORETICAL APPROACH

Using wave theory, we can find the distribution of the sound pressure in a room even at frequencies so low that the sound field is not diffuse. As an example of this approach, we will study the sound pressure distribution in an idealized rectangular room, but other geometries can also be analyzed using the same approach.

The room under study is shown, with the chosen coordinate system, in Figure 4.12a. It is assumed to have hard, immovable walls. The air in the room is assumed not to absorb any sound power, and the sound source is assumed to be a point source.

The wave equation in Cartesian coordinates is:

$$\frac{\partial^2 \underline{p}}{\partial x^2} + \frac{\partial^2 \underline{p}}{\partial y^2} + \frac{\partial^2 \underline{p}}{\partial z^2} + k^2 \underline{p} = 0 \tag{4.34}$$

The spatial distribution of the sound pressure over the room is a function of the room coordinates. The spatial distribution of particle velocity is linked to the pressure distribution. Since the wall surfaces are hard and immovable, the particle velocity component perpendicular to the respective walls must be zero:

$$\underline{u}_x\big|_{x=0,l_x} = 0 \quad ; \quad \underline{u}_y\big|_{y=0,l_y} = 0 \quad ; \quad \underline{u}_z\big|_{z=0,l_z} = 0 \tag{4.35}$$

As usual, when dealing with partial differential equations, we guess a reasonable form of solution. We then test the solution's validity by checking if the solution satisfies the differential equation. We expect

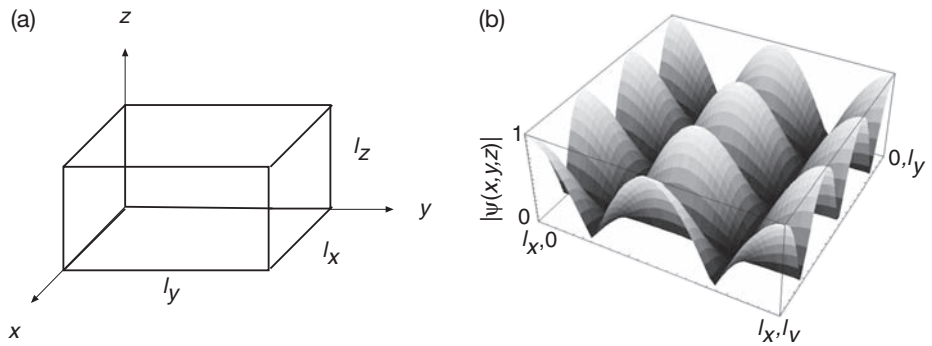


Figure 4.12 The rectangular room under study and its coordinate system (a). The normalized magnitude of the eigenfunction $|\Psi(x, y, z)|$ for the mode q_x: q_y: q_z = 3: 2: 0 (b).

the sound pressure to be high at the walls in analogy with the standing wave in front of a hard wall. In this case, a reasonable guess then is a solution of the form:

$$\underline{p}(x,y,z) = \hat{p}\,\Psi(x,y,z)\,e^{j\omega t} \tag{4.36}$$

where the spatial distribution of sound pressure is given by the function $\psi(x,y,z)$:

$$\Psi(x,y,z) = \cos(k_x x)\cos(k_y y)\cos(k_z z) \tag{4.37}$$

The second order derivative of p with respect to x is:

$$\frac{\partial^2 \underline{p}}{\partial x^2} = -k_x^2\,\hat{p}\cos(k_x x)\cos(k_y y)\cos(k_z z)\,e^{j\omega t} = -k_x^2\,\underline{p} \tag{4.38}$$

We find the second order derivatives with respect to the y and z directions in the same way. By inserting the second order derivatives with respect to x, y, and z into the differential equation, we obtain the condition:

$$k_x^2 + k_y^2 + k_z^2 = k^2 \tag{4.39}$$

where k is the wave number, $k = \omega/c$, which depends on the frequency of the sound. For the solution (which we guessed) to be a valid one, the condition 4.39 must be met.

The wave number k has the components k_x, k_y, and k_z. These components can be regarded as the wave number components of waves moving around inside the rectangular room, bouncing off the walls. The waves will interfere and lead to standing wave patterns in the same way as in the one-dimensional case of Kundt's tube.

The room will be resonant for the combinations of k_x, k_y, and k_z that fulfill the condition 4.39. At resonance, the energy in the room will build up infinitely if there are no losses. This is called an oscillation mode of the room, an *eigenmode,* or simply *mode*.

The function ψ, which describes the spatial variation of the sound pressure at resonance frequency of a particular eigenmode, is called an *eigenfunction*. Figure 4.12b shows an example of the normalized magnitude of the eigenfunction for such an eigenmode.

To determine the eigenfunctions, we need to find values for the wave number components k_x, k_y, and k_z. We use the condition that the particle velocity perpendicularly to the walls, $\underline{u}_x$, $\underline{u}_y$, and $\underline{u}_z$, must be zero.

We start by studying the conditions in the x direction—that is, at the walls at $x = 0$ and $x = l_x$. The equation of motion, Equation 2.7, gives us an expression for the particle velocity $\underline{u}_x(x,y,z)$ as a function of $\underline{p}(x,y,z)$. Insertion of the solution 4.36 for the differential equation into this expression gives us the particle velocity $\underline{u}_x(x,y,z)$:

$$\underline{u}_x = -\frac{1}{\rho_0}\int \frac{\partial \underline{p}}{\partial x}\,dt = -\frac{1}{\rho_0}\frac{1}{j\omega}\frac{\partial \underline{p}}{\partial x} \tag{4.40}$$

$$\underline{u}_x = \hat{p}\,\frac{k_x}{\rho_0}\frac{1}{j\omega}\sin(k_x x)\cos(k_y y)\cos(k_z z) \tag{4.41}$$

The conditions for the particle velocity at the wall at $x = 0$ is obviously fulfilled. Analogously, we find that the solution 4.28 also fulfills the conditions for the particle velocity at the walls at $y = 0$ and $z = 0$.

According to the expression given in Equation 4.40, for zero particle velocity at right angles to the wall at $x = l_x$, the following condition must be met:

$$\sin(k_x l_x) = 0 \tag{4.42}$$

The sine will only be zero when the wave number component k_x is:

$$k_x = \frac{q_x \pi}{l_x} \tag{4.43}$$

where q_x is a natural number 0, 1, 2, . . . In the same way, we find that requirements for the wave number components k_y and k_z are:

$$k_y = \frac{q_y \pi}{l_y} \tag{4.44}$$

$$k_z = \frac{q_z \pi}{l_z} \tag{4.45}$$

Inserting the values for the wave number components into Equation 4.39, we find that the condition for the wave equation to be satisfied (with the specified geometry and boundaries) is:

$$k^2 = \frac{\omega^2}{c_0^2} = \left(\frac{q_x \pi}{l_x}\right)^2 + \left(\frac{q_y \pi}{l_y}\right)^2 + \left(\frac{q_z \pi}{l_z}\right)^2 \tag{4.46}$$

Obviously, there are an infinite number of combinations of q_x, q_y, and q_z. Each of these combinations will identify a possible eigenmode. The *eigenfrequencies* of these modes are determined by the speed of sound and the room dimensions. The combination (q_x: q_y: q_z) is often called the *mode index*.

Eigenfrequencies and Eigenmodes

If we rewrite Equation 4.46, we find that the *k*-values correspond to frequencies according to the formula:

$$f_{q_x, q_y, q_z} = \frac{c_0}{2} \sqrt{\left(\frac{q_x}{l_x}\right)^2 + \left(\frac{q_y}{l_y}\right)^2 + \left(\frac{q_z}{l_z}\right)^2} \tag{4.47}$$

These frequencies are often called the eigenfrequencies or *resonance frequencies of the room*. For each eigenfrequency, there is at least one eigenmode. For certain room dimensions, many eigenmodes may have identical eigenfrequencies. In this discussion, we have limited ourselves to the case of hard and immovable walls. If the walls can move, have a finite impedance, the resonance frequencies will change.

Each mode in the rectangular room is characterized by a spatial distribution of sound pressure of the form:

$$\underline{p}_{q_x, q_y, q_z}(x, y, z) = \hat{p}_{q_x, q_y, q_z} \cos\left(\frac{q_x \pi x}{l_x}\right) \cos\left(\frac{q_y \pi y}{l_y}\right) \cos\left(\frac{q_z \pi z}{l_z}\right) \tag{4.48}$$

The sound field in a practical room with losses is built up by a superposition of pressure contributions from an infinite number of modes since the losses are accompanied by the resonances having finite bandwidths.

One can show that for a small sound source, having a volume velocity $\underline{U}$, placed at location, r_0, the expression for the total sound pressure at another point, r, due to all the modes, is given by an infinite sum of the pressure contributions of the various modes according to:

$$\underline{p}_{tot}(\omega, \mathrm{r}, \mathrm{r}_0) = \frac{j\omega\rho_0 c_0^2}{V}\underline{U}\sum_{q_x}^{\infty}\sum_{q_y}^{\infty}\sum_{q_z}^{\infty}\frac{\Psi_{q_x,q_y,q_z}(\mathrm{r})\Psi_{q_x,q_y,q_z}(\mathrm{r}_0)}{\Lambda_{q_x,q_y,q_z}\left[\left(\omega_{q_x,q_y,q_z}^2-\omega^2\right)+2j\omega_{q_x,q_y,q_z}\delta_{q_x,q_y,q_z}\right]} \tag{4.49}$$

In this expression, δ is the decay constant discussed previously, and the expression is only valid for small values of δ. The expression is also not valid close to the monopole at r_0. The normalizing factor Λ for a mode depends on if the mode is created by waves moving in one ($\Lambda = 1/2$), two ($\Lambda = 1/4$), or three ($\Lambda = 1/8$) dimensions.

An interesting property of Equation 4.49 is that the source and observation points r and r_0 can be exchanged. This property is an example of *reciprocity*.

The steady state frequency is determined by the sound source. How *strongly* any particular mode will be excited depends on the following:

- How close its resonance frequency is to the excitation frequency
- How damped the particular mode is
- Where the source is located in the room

We note from Equation 4.37, that all eigenfunctions always have a large magnitude at the corners of the room in the simple case studied here. We also note that the ratio between the sound pressure and the particle velocity is highest at the corners; that is, the sound field impedance is highest at the corners. This means that a constant volume velocity-type source will excite the room modes best when placed in the corner of a room. Many technical sound sources, such as machines and loudspeakers, are constant volume velocity-type sources.

Using a loudspeaker and a sine-wave generator, we can study the properties of the room modes. The summed sound pressure of several excited room modes is going to depend on how the sound pressure contributions of each mode add up with respect to magnitude and phase. This in turn depends on the particular modes that contribute at a particular location as shown by Equation 4.49. The sound pressure must therefore vary from one position to the next. At high frequencies where many modes leak into one another, the sound pressure level will vary considerably from one position to the next as shown by the experimental curve in Figure 4.13. The sharp minima and maxima are due to interference between many modes.

When the sound source is switched off, the modes will start to oscillate at their natural resonance frequencies (given by Equation 4.47) if the modes are weakly damped and uncoupled. Since the modes having resonance frequencies closest to the excitation frequency are likely to have been excited most strongly, they will tend to dominate the decay process.

Modal Density

If we study Equation 4.47 for the determination of the mode frequencies, we note that it describes the length of the diagonal in *frequency space*, extending from the origin out to a point determined by q_x, q_y, and q_z as shown in Figure 4.14. Note that all modes having lower resonance frequencies than f will have shorter diagonals—in other words, they will be contained within a radius of f Hz.

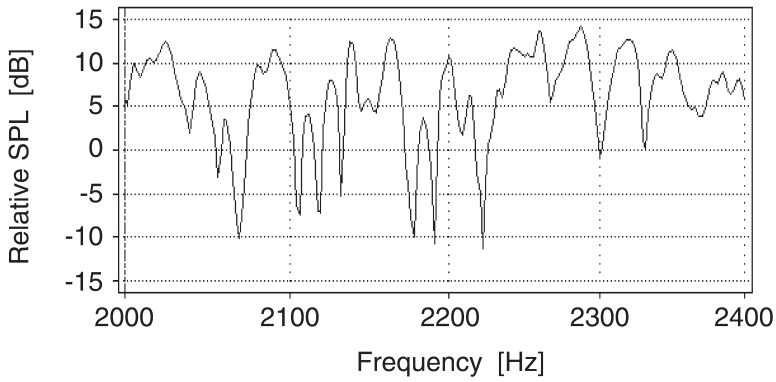


Figure 4.13 An example of the variation of the frequency response in the diffuse field of a living room as a function of frequency obtained from the room impulse response shown in Figure 4.6.

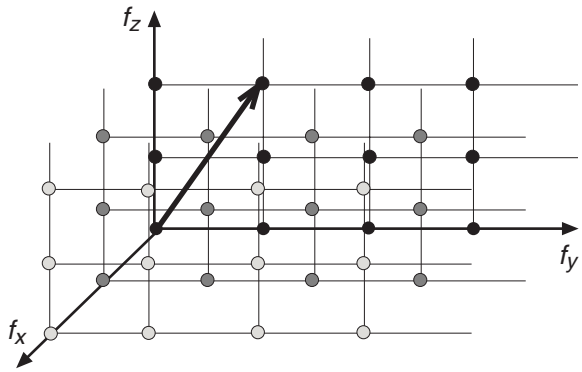


Figure 4.14 Resonance frequency lattice in the *frequency space* for a rectangular room. The arrow marks the lattice point corresponding to a mode index (q_x,q_y,q_z) = (0,1,2). The lattice points are assigned according to Equation 4.47.

The total number of modes below the frequency described by q_x, q_y, and q_z, and l_x, l_y, and l_z, can then be calculated by dividing the volume of 1/8 of the sphere by the volume of each frequency box. The reason we do not use the full sphere volume is that the numbers q_x, q_y, and q_z have to be natural numbers—that is, 0, 1, 2, 3, . . . Negative numbers would result in counting each frequency more than one time. Each frequency box has a volume, which is:

$$\Delta V = \frac{c_0^2}{8 l_x l_y l_z} \tag{4.50}$$

This now gives us the total number of modes N below some frequency f in a volume V as:

$$N \approx \frac{4\pi V}{3c_0^3} f^3 \tag{4.51}$$

The number of modes in the *dense mode region* in a room with an arbitrary (but not extreme) shape can be estimated rather well using this equation. However, in deriving this equation, we have not included waves that travel parallel to one or more walls. Such waves have one or two q-numbers that are zero. One can show that if one includes these modes, a more exact formula for the number of modes below a frequency f in a rectangular room is:

$$N \approx \frac{4\pi V}{3c_0^3} f^3 + \frac{\pi S}{4c_0^2} f^2 + \frac{\pi L}{8c_0} f \tag{4.52}$$

where V is room volume ($l_x\, l_y\, l_z$), S is total wall surface area 2 ($l_x\, l_y + l_y\, l_z + l_x\, l_z$), and L is total edge length 4 ($l_x + l_y + l_z$).

By differentiating N in Equation 4.51 with respect to f, one can obtain an expression for the number of modes ΔN in a narrow frequency band Δf. One finds:

$$\Delta N \approx \frac{4\pi V}{c_0^3} f^2 \Delta f \tag{4.53}$$

The *modal density* $\Delta N/\Delta f$ is dependent on both frequency and room volume. This is one of the reasons that we can, by ear, decide whether a room sounds small or large. Small rooms are unsuitable for music reproduction since they cannot support all the low-frequency sounds by appropriate (and suitably damped) room modes.

The presence of enough room modes is also necessary when measuring the spatially averaged effective value of the sound pressure. The number of active modes is then dependent on both the modal density of the room and the bandwidth of the noise. This applies equally well to the measurement of reverberation time using noise. Typically, one requires at least 10 modes in a frequency band in the room under study to assume reasonable diffusitivity.

The number of modes in a third octave-wide frequency band is given by:

$$\Delta N \approx 0.23V \frac{f^3}{c^3} \tag{4.54}$$

We note that there will only be a small number of modes strongly excited at low frequencies in small rooms, leading to poor averaging and measurement results. This corresponds to the condition of poor sound field diffusitivity, as discussed in the section on statistical room acoustics.

Complex Eigenfrequencies

Acoustical power losses are always present in real rooms. To simplify the discussion in this section, we assume all losses to be a part of the propagation through air so that the losses are continuous.

The energy of each eigenmode must decay exponentially in a linear system. The sound pressure of a mode can therefore be written:

$$\underline{p}_q = p_q(x, y, z) e^{j\omega_q t} e^{-\delta_q t} \tag{4.55}$$

where $q = [q_x, q_y, q_z]$.

Here δ_q is the mode-specific, real-valued decay constant introduced previously. We can thus write the exponential decay as:

$$e^{j\omega_q t} e^{-\delta_q t} = e^{j\underline{\omega}_q t} \tag{4.56}$$

where

$$\underline{\omega}_q = \omega_q + j\delta_q \tag{4.57}$$

is the complex eigenfrequency.

The resonance curves for the effective sound pressure of a particular mode (as a function of frequency) will have a certain width because of the presence of δ_q, as indicated by Equation 4.49. The resonance curves for the various modes will overlap considerably even at quite low frequencies.

The Q-Value

A metric to describe the *peakiness* of a resonance curve is the *Q-value*. The Q-value is calculated from the effective value of the sound pressure as a function of frequency as shown in Figure 4.15. The Q-value is defined as the resonance frequency divided by frequency width of the resonance curve at the –3dB points and is consequently a dimensionless number.

Low and high Q-values correspond to strongly and weakly damped resonances respectively. The Q-value concept is used in many other applications when dealing with resonances, for example, in conjunction with vibration isolation. Note that a response curve, such as the one shown in Figure 4.13, is the result of the addition of a large number of resonances and that the Q-value of a peak, in this case, is not the same as the Q-value of an individual resonance.

Loss Factor and Reverberation Time

We have already studied the relationship between the decay constant and the reverberation time. One can derive a similar expression for the relationship between the loss factor of an oscillating system and its reverberation time.

The spatial average of the effective value of the sound pressure is proportional to the acoustic energy in the room:

$$E \propto < \tilde{p}^2 > \tag{4.58}$$

The decay of the mode will be a function of time:

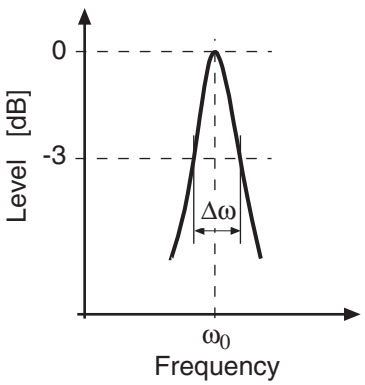


Figure 4.15 The response of a simple first order resonant system. The Q-value is defined as $Q = (\omega_0/\Delta\omega)$ where $\Delta\omega$ is the frequency at which the resonance curve is 3 dB below the peak value at the resonance frequency ω_0.

$$E \propto e^{-2\delta_q t} \tag{4.59}$$

It is often practical to express the decay of the room modes by a *loss factor*, η. The loss factor is usually defined as the relative loss of energy per radian:

$$\eta = \frac{W_+}{E_0}\frac{1}{\omega} \tag{4.60}$$

Using the previously derived expressions for the decay constant one can also write the loss factor as:

$$\eta_q = \frac{2\delta_q}{\omega_q} \tag{4.61}$$

We note also that the loss factor is a dimensionless number.

It is useful to know that the Q-value is the inverse of the loss factor:

$$Q_q = \frac{1}{\eta_q} \tag{4.62}$$

Assume that a room is excited by a sound source that is switched off at time $t = 0$. The power radiated by the source is W_+. The steady-state value of the energy in the room is then determined by the fact that the power loss W_- is equal to the power injected by the sound source W_+. If we use Equations 4.59 and 4.61, we find:

$$W_- = -\frac{dE}{dt} = \omega\eta E_0 e^{-\omega\eta t} \tag{4.63}$$

At time $t = 0$, the energy in the room is given by:

$$W_+ = \omega\eta E_0 \tag{4.64}$$

Using Equations 4.9, 4.61, and 4.64, we obtain the relationships between reverberation time, decay constant, and loss factor as:

$$T = \frac{3\ln(10)}{\delta} = \frac{6\ln(10)}{\omega\eta} \approx \frac{2.2}{f\eta} \tag{4.65}$$

and

$$T = \frac{3\ln(10)}{\delta} = \frac{6\ln(10)Q}{\omega} \approx \frac{2.2Q}{f} \tag{4.66}$$

Wave Theory and Diffuse Sound Fields

The wave theory approach to room acoustics tends to be complicated when we want to solve practical problems characterized by:

- High- and wide-frequency ranges
- Non-rectangular and irregularly shaped rooms
- Rooms that have uneven distribution of sound absorption

It is more practical then to use the approach offered by statistical room acoustics. The relationship between the wave theory for high frequencies and the statistical theory is fairly straightforward.

If one has a large number of modes excited by a wide bandwidth noise, the room does not have an extreme shape, and the modes have approximately the same damping, one can regard the resulting sound field as diffuse. The reason is that each mode is associated with a certain wave vector and consequently sound propagation direction in the room.

If there are many modes, there will be several directions of incidence. If, in addition, the modes are equally damped, the intensity of the sound from each direction will be approximately the same. The requirement for random phase will be satisfied if we excite the room by a wide bandwidth noise signal.

When the damping of the modes is so high that we have wide resonance curves and large *mode overlap*, it is difficult to keep track of the individual modes. An upper-frequency limit, for the application of wave theory, is commonly taken as the frequency at which the Q-value of the resonance curves is less than 3. This can be shown to correspond to mode overlaps of more than 3; that is, at least 3 modes within the half-value widths of the resonance curves of the effective value of the sound pressure.

This limit is the *Schroeder frequency*, f_S, and is approximately given by the expression:

$$f_S \approx 2000\sqrt{\frac{T}{V}} \tag{4.67}$$

Here T is the reverberation time and V the volume of the room. In practice, the calculation of the Schroeder frequency is somewhat iterative.

4.5 PROBLEMS

4.1

Task: Which eigenmodes belong to the octave and the third octave bands centered on 50 Hz, in a rectangular room having the dimensions $L \times W \times H = 10 \times 8 \times 5$ m^3?

4.2 The number of eigenmodes per frequency band is of importance both for the sound quality, when the room is used for performance/listening, and for the choice of acoustic measurement methods.

Task: Calculate the number of eigenmodes in each of the third octave bands from 100 to 500 Hz for a room having the dimensions $L \times W \times H = 4.5 \times 4.0 \times 2.6$ m^3.

4.3 In a room having the dimensions $L \times W \times H = 10 \times 7 \times 3.5$ m^3, the reverberation time, at frequencies high enough for diffuse sound field conditions, was measured as 1.5 s.

Tasks:

a. Determine the absorption area A_S and the mean sound-absorption coefficient α_m using Sabine's formula. Neglect the sound absorption due to air. Is $\alpha_m < 0.3$?

b. Repeat *a* but include sound absorption due to air by using the attenuation coefficient $m = 0.002\ \text{m}^{-1}$.
c. Determine the mean sound-absorption coefficient using Eyring's formula. Neglect the sound absorption due to air.

4.4 The reverberation times in the third octave bands in the table were measured in a room having the dimensions $L \times W \times H = 10 \times 7 \times 3.5\ \text{m}^3$.

f	50	63	80	100	125	160	[Hz]
T	2.5	2.5	2.0	1.7	1.4	1.4	[s]
f	200	250	315	400	500	630	[Hz]
T	1.3	1.3	1.3	1.3	1.3	1.2	[s]

Task: Determine from which frequency one can use the concept of statistical room acoustics using the Schroeder frequency as the criterion.

4.5 Assume an omnidirectional source emitting a wide-frequency range signal with the sound power in a room of 0.01 W is being used. The room dimensions are $L \times W \times H = 10 \times 15 \times 5.5\ \text{m}^3$, and the absorption area is 50 m^2S. The frequency range considered is above the Schroeder frequency.

Tasks:

a. Calculate the rms sound pressure as a function of distance to the sound source for distances 0.5, 1, 2, 4, and 8 m.
b. Calculate the corresponding sound pressure levels.

4.6 Consider the following three rooms:

a. A living room of volume $V = 80\ \text{m}^3$ and having a reverberation time $T = 0.5$ s
b. A theatre of volume $V = 4{,}000\ \text{m}^3$ and having a reverberation time $T = 1.1$ s
c. A concert hall of volume $V = 15{,}000\ \text{m}^3$ and having a reverberation time $T = 1.7$ s

Task: Assuming an omnidirectional sound source, calculate the reverberation radii for the sound fields in the respective rooms.

4.7 A general purpose studio is to be adjusted for recording of speech, which requires low reverberation times in the octave bands 0.5, 1, and 2 kHz. The table lists the present and the desired reverberation times. The volume of the studio is 60 m^3, and its total surface area is 94 m^2.

Octave band	0.5	1	2	[kHz]
T_{current}	0.43	0.38	0.38	[s]
T_{desired}	0.3 ± 0.02	0.3 ± 0.02	0.3 ± 0.02	[s]

Tasks:

a. Calculate the acceptable interval for the absorption area for each octave band. Sound-absorbing panels having the area 1.2 × 1.2 m^2 are available. The sound-absorption coefficient for these panels are:

Octave band	0.5	1	2	[kHz]
α	87	94	90	[%]

b. Calculate how many panels are required to achieve the desired reverberation times. Assume that the panels are mounted on top of surfaces having the current mean sound-absorption coefficient. Use Sabine's formula and neglect sound absorption due to the medium.

After the adjustments have been done, one finds that the mean sound-absorption coefficients are so high that Eyring's formula should have been used instead of Sabine's formula. In the Eyring formula, the expression $A_S = 4mV - S\ln(1 - \alpha)$ is used to calculate the total absorption of the room A_S.

c. Calculate the resulting reverberation times using Eyring's formula instead of Sabine's formula.

4.8 A small loudspeaker is radiating sound into a rectangular room where the dimensions are $L \times W \times H = 7 \times 5 \times 3$ m^3. The loudspeaker is positioned at the coordinate $(x_0, y_0, z_0) = (0, 0, 0)$. The following signal types are used:

1. A sinusoidal tone having a frequency of 66 Hz
2. Band-limited noise having a center frequency of 2 kHz and a bandwidth of 300 Hz

In both cases a sound pressure level of 75 dB was obtained at the coordinate $(x_1, y_1, z_1) = (2, 3, 1)$.

Task: Calculate the sound pressure level at the coordinate $(x_2, y_2, z_2) = (4, 1, 2)$ for the two cases using an appropriate theory.

4.9 A classroom having the dimensions $L \times W \times H = 12 \times 7 \times 3.5$ m^3 is to be acoustically adjusted so that the reverberation times are 0.8 ± 0.1 s in all octave bands in the range 125–4000 Hz.

The floor is covered by a soft carpet on top of concrete. One wall has 5 windows with a total area of 18 m^2 and has 10 m^2 covered by drapes. The rest of the walls, as well as the ceiling, are made of concrete covered by plaster.

The sound-absorption coefficients of the various surfaces are:

Octave band	125	250	500	1k	2k	4k	[Hz]
Plaster on concrete	2	2	3	2	5	5	[%]
Windows	40	30	20	17	15	10	[%]
Drapes	12	20	42	53	64	62	[%]
Carpet on concrete	9	8	21	26	27	37	[%]
Alternative (a) above	63	42	35	12	8	8	[%]
Alternative (b) above	57	97	68	63	59	40	[%]

Neglect the sound absorption due to doors, persons, and chairs.

Two different sound-absorbing treatments are available:

a. Using 0.006 m thick particle board on 0.05 m high battens on top of the concrete

and/or

b. Using 0.004 m thick perforated particle board on 0.15 m high battens on top of the concrete but with 0.03 m thick glass wool in the air space

Task: Determine the required area of each treatment to achieve the desired design goal. Motivate your choice.

4.10 A person living in an old apartment has a bedroom immediately above the boiler room. The noise from the boiler is disturbing, and its fundamental at 30 Hz corresponds to the lowest eigenfrequency of the bedroom. The eigenmode at this frequency completely dominates the sound field in the bedroom.

The bedroom pillow is located at coordinate (x_1, y_1, z_1)= (0.3, 0.3, 0.3). When the local health inspector measures the sound level this is done using a microphone approximately at the center of the room, in this case at the coordinate (x_2, y_2, z_2) = (2.5, 1.5, 1.5). The sound level L_{pA} at this point was measured as 35 dB, which is considered acceptable.

Task: Calculate the actual sound level at the point of the pillow, if the mode theory for an undamped rectangular room can be applied to the problem.

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Spatial Sound Perception 5

5.1 INTRODUCTION

The listener's auditory spatial perception is influenced by many sound field properties. Some of those properties are energy-related, such as reverberation time, the relationships between the levels of direct and reverberant sound, and the spectra of the incoming sound components. Some are spatially-related, for example, the spatial distribution of the incoming sound field components. Still others are dependent on arrival time, such as echoes and other strong sound reflections.

The subjective perception of spatial characteristics often does not map well onto the physically measured dimensions of the spatial sound field properties. As an example, one can mention that the perceived reverberation times in auditoria often do not correspond to those obtained by using instrumentation.

It is therefore necessary to be quite careful when using results from classical physical measurement, such as reverberation time, in judging the quality of an auditorium, such as a theater or concert hall. A careful study of the fine structure of the sound field is usually necessary for expressing oneself knowledgeably about the room acoustic quality. Unfortunately, considerable knowledge about human spatial sound perception seems to be lacking at this stage, even though great progress has been made in recent years.

To study the influence of various sound field parameters in a well controlled and detailed way, it is necessary to move from the auditorium to the laboratory. In *psychoacoustic tests* in the laboratory, it is convenient to use a technique known as *sound field synthesis*. This technique involves the variation of different sound field components, sound source radiation patterns, reflections, noise, and so forth, thus studying the parameters that are believed to influence spatial sound perception.

To use sound field synthesis, one needs to use either binaural sound reproduction or sound reproduction over loudspeakers in an *anechoic chamber*. This is necessary for complete control of the sound field to which the test person is exposed. In a regular room, the test person would be subject to uncontrolled sound field components due to sound reflections from walls, for example. Another possible problem would be background noise.

An anechoic chamber is characterized by low background noise levels, L_{pA} less than 15–20 dB, and by walls that are nearly nonreflective over some limiting frequency, for example 100 Hz. Usually the walls have a sound-absorption coefficient larger than 0.99. A classic system for sound field synthesis in an anechoic chamber uses a large loudspeaker array, such as that shown in Figure 5.1 (see Reference 5.15).

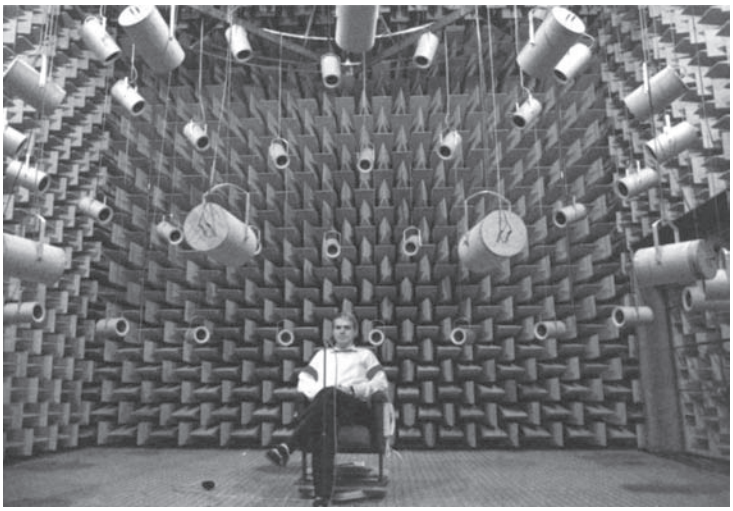


Figure 5.1 An anechoic chamber equipped for experiments in psychoacoustics and sound field synthesis using a large hemispheric loudspeaker array. (Photo by Jakub Kirszenstein).

In addition to the loudspeakers, the equipment includes *reverberation units*, *delay units,* among others, so that one can create an *illusion of presence*, similar to that of being in the sound field of an actual room.

More modern types of sound field synthesis use loudspeaker equivalent *binaural sound reproduction,* using only two loudspeakers and a technique known as *crosstalk cancellation*. In using crosstalk cancellation, the signals through the loudspeakers are tailored by digital filtering so that they, at the listener, correspond to those signals that would be obtained by conventional binaural listening using headphones.

Of course, one can use direct binaural sound synthesis to achieve simulation of the room sound field components.

The various techniques for achieving audible room sound field simulation are often called *auralization*, in analogy with visualization by computer aided techniques as used in architectural design (see Reference 5.16 and the discussion in Chapter 6).

5.2 SUBJECTIVE IMPRESSION OF SOUND FIELD COMPONENTS

When listening to sound under normal circumstances, the auditory perception is determined not only by what one hears but also by what one sees. In trying to find the fundamental properties of hearing it is necessary, in some way, to remove or neutralize other sensory information so that the listener is not affected, at least by visual sensory input (although visual information related in some way to the test in question may be added for experimental purposes). From this perspective as well, using an anechoic room may have both positive and negative influence on the *test*, the *test person*, and the *test results*.

Since 1950, many psychoacoustic tests have been made to study the behavior of hearing when exposed to sound fields of both small and large complexity (see Reference 5.14).

The following sections relate some of the important results of this work. One must remember that many of these tests were done with a fairly small and select number of test persons, that they were

seldom made with the same audio test material, and that the listening conditions varied between laboratories. Critical reading is advised.

Simple Sound Fields

The real or simulated sound field properties are determined by many factors, such as:

- Sound pressure level L_p of the direct sound
- Levels, angles, and arrival times of delayed sound components relative to the direct sound
- Spectra of delayed sound components

There are primarily four phenomena that are observed: (1) *masking*, (2) *localization*, (3) *echo*, and (4) *coloration*.

In the psychoacoustic tests, the test person listens to sounds having various angles of incidence in the horizontal and median planes. In addition, the delay time between the various signals Δt is varied. Such tests allow the researcher to investigate the importance of both time delay between direct sound and (simulated) sound reflections by room walls. Unless otherwise noted, all angles used in this chapter refer to the head-related coordinate system shown in Figure 3.2. Also, all direct sound field components arrive at an angle of (0°, 0°) unless otherwise indicated.

Figures 5.2 and 5.3 show the results of some tests using simple sound fields where recorded speech was used as audio material. Figure 5.2 shows the sensitivity of hearing to a single additional sound field component (that is, besides the direct sound field component), incident at (0°, 0°), as a function of its relative delay time Δt.

In a corresponding way, Figure 5.3 shows how the sensitivity varies as a function of angle of incidence (θ°, 0°) for a delay of Δt of 50 ms. It is noteworthy that the sensitivity increases as delay time is

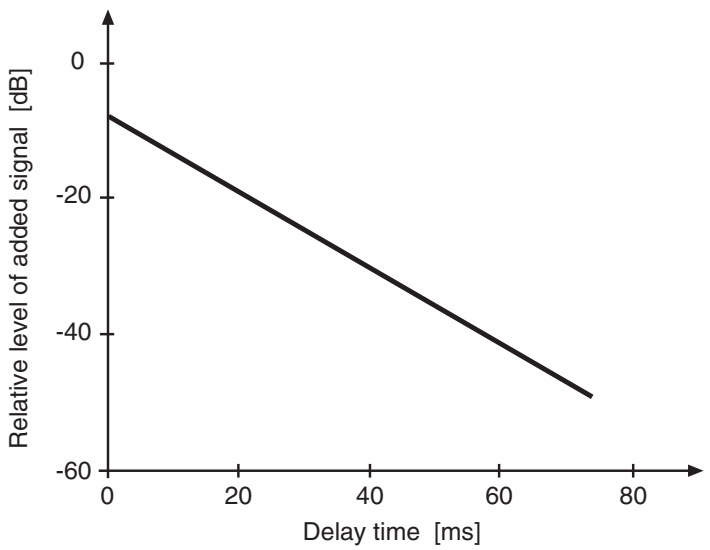


Figure 5.2 Regression line for the relative threshold for noticing a delayed signal added to a direct signal in the case of frontal sound incidence (speech at L_p = 70 dB). (After Ref. 5.1)

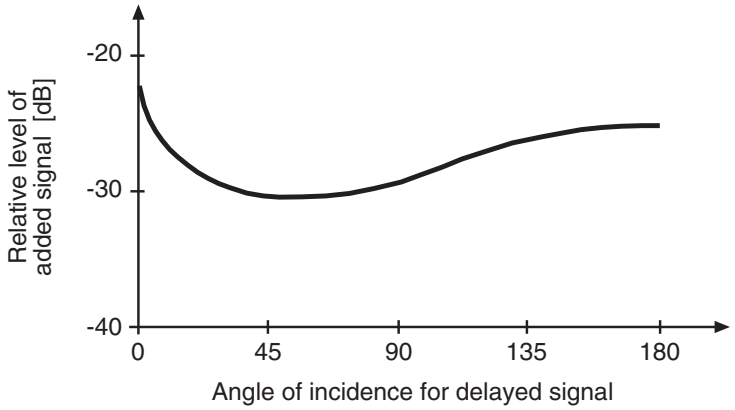


Figure 5.3 Relative threshold for noticing a signal delayed by 50 ms relative to a direct speech signal as a function of horizontal angle for the delayed signal. Both sound components incident in the horizontal plane, direct sound has frontal incidence. (After Ref. 5.2)

increased, indicating reduced masking in Figure 5.2, and that the sensitivity seems to be at a maximum at an incidence angle of (45°, 0°), as shown by the curve in Figure 5.3.

Tests using music as a test signal give somewhat different results, although it is seldom possible to do direct comparisons between tests. Music has a wider range of modulation frequencies than speech and often contains more low frequency modulation. Music is also characteristic in its wide spectrum, which makes it easier to hear interference effects, often as *coloration*—that is, usually unwanted changes in timbre. Another important difference is that music usually comes from many source locations, so it is less relevant to talk about one direction of incidence for the direct sound. It is also worth noting that it is quite difficult to find anechoically recorded music, usually stereo or mono recordings of various anechoic quality have been used for tests. Figures 5.4–5.6 show some important results from

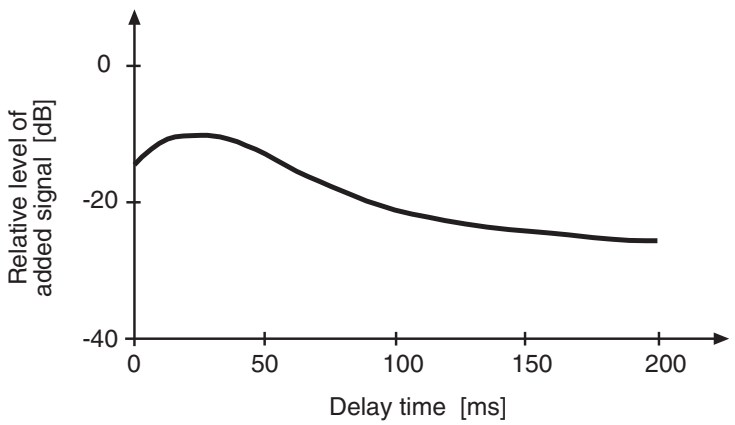


Figure 5.4 Relative for noticing a delayed signal incident at $\varphi = 90°$ added to direct sound incident at $\varphi = 0°$ (mean for six musical works). (After Ref. 5.3)

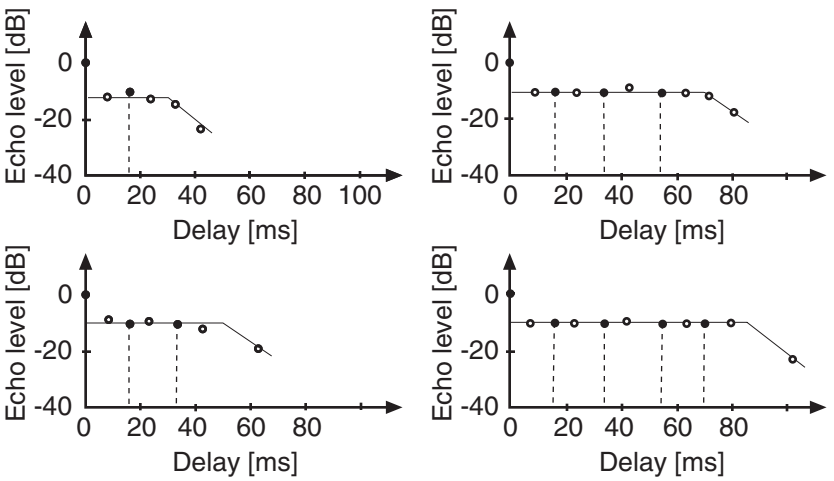


Figure 5.5 Relative threshold for noticing a delayed signal (open circles), added to a sound field of increasing complexity obtained by adding 1, 2, 3, and finally 4 sound field components with varying time delay and level (solid circles at 0 and −10 dB respectively) (speech, $\varphi = 0°$). (After Ref. 5.2)

psychoacoustic tests using recorded music, usually classical music. Figure 5.4 shows that the increased frequency range of music make us more sensitive to early reflections, as compared to the results shown in Figure 5.1 that were obtained for speech.

Usually there are no single sound field reflections in a room; reflections arrive in greater density as time goes. Results of simple tests using multiple sound reflections are shown in Figure 5.5. The graph shows the sensitivity of the test persons to various numbers of added sound field components all having arrival angles of (0°, 0°) and a level of −10 dB relative to the direct sound. Although this sound field structure is not really possible in a normal room, the results show that in the case of several sound field components, the components can mask one another. The results also show that the sensitivity increases only after some 20 ms of time have passed after the arrival of the last component.

Once the additional sound field components have risen above the threshold of perceptibility, they will influence the way the sound field is perceived in many ways; phenomena such as echo and coloration, for example, will be clearly noticeable.

Figure 5.6 shows how various levels and time delays of only one single additional sound field component (90°, 0°) will sound. It is often desired to have good spaciousness; low coloration and absence of directional shift are undesired. These problems, created by early reflected sound, are well known in dealing with studio acoustics.

In summary, it can be noted that hearing is extremely sensitive to added sound components, particularly if they are not spatially displaced and if the time delays involved are of the order of 5 ms or smaller.

The Precedence Effect

This effect, also called the *law of the first wave front,* implies that it is the arrival direction of the first of several coherent sound contributions (temporally and spatially displaced) that determines the subjectively

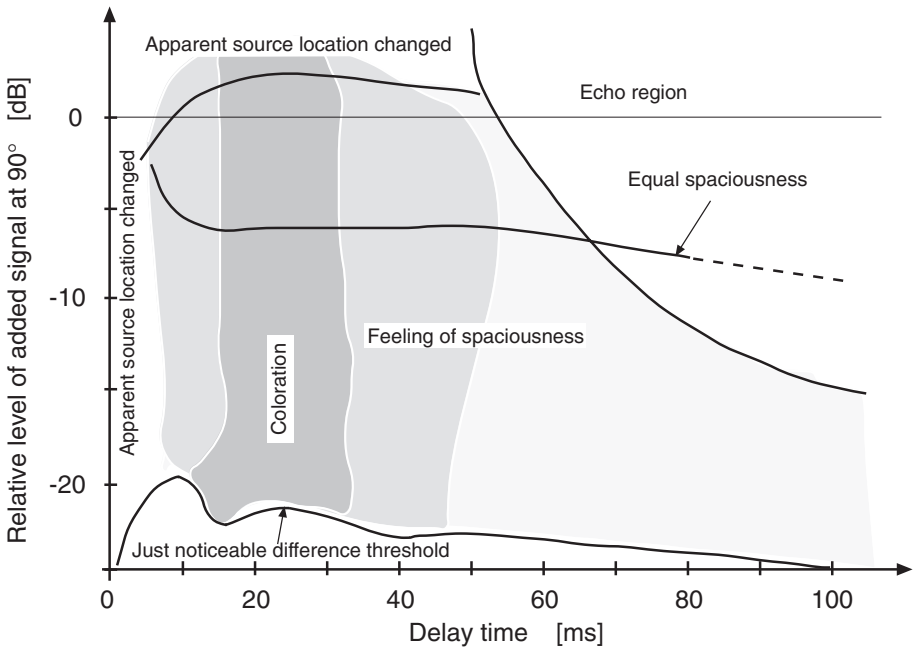


Figure 5.6 Various subjective effects of an added sound field component at 90°, for various delays and levels relative to the direct sound which has frontal incidence (classical music). (After Ref. 5.4)

perceived direction of the sound source. The law of the first wave front usually works quite well in rooms or auditoria, for example, but in some cases, such as when the sound source is obscured or when there are focusing surfaces, the directional impression can shift in an undesirable way. The limits of the principle are therefore of interest.

Psychoacoustic experiments, both regarding stereo sound reproduction and the usefulness of the principle, have given interesting results.

In one experiment, the test subjects were subject to direct sound arriving from the direction (0°, 0°) and the same sound, but variably delayed, arriving from (90°, 0°). The results showed that for the perceived sound source location to be shifted to (45°, 0°), that is, halfway away from the direction of the first wave front, the component arriving at (90°, 0°) had to be approximately 10 dB stronger than the direct sound component, as shown in Figure 5.7.

These results verify the observations discussed initially. The law of the first wave front can be put to good use in many sound reinforcement applications where amplification of sound, without directional shift, is desired. By introducing a small time delay into the amplified signal, the directional impression of the sound source will remain unchanged, while the level of the direct signal is perceived to have increased.

Delay versus Level Differences

In stereo sound reproduction, two loudspeakers and sound reproduction channels are used. The line between the loudspeakers is called the baseline. The listener is situated at a right angle to the baseline

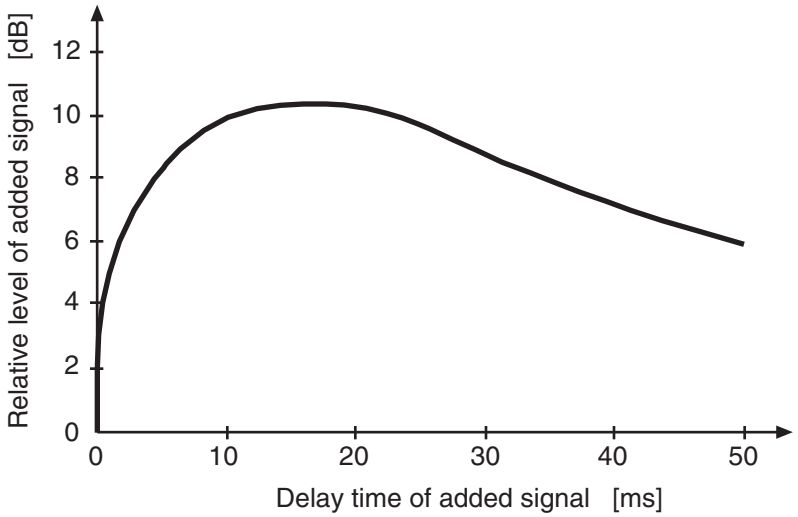


Figure 5.7 The level difference between the added delayed signal and the direct signal that gives both signals the same loudness, as a function of delay time (speech). (After Ref. 5.5)

symmetrically with respect to the loudspeakers. An equilateral triangle setup is often used. Stereophonic sound reproduction is based on the perception of virtual sound sources, sometimes called *phantom sources* (see Reference 5.6).

To render a phantom source on the baseline of the stereo loudspeaker pair, it is necessary that the signals radiated by the two loudspeakers be suitably correlated. Using magnitude and arrival time differences between the sounds from the two loudspeakers, one can move the phantom sources along the baseline between the two loudspeakers, as indicated by Figures 5.8 and 5.9. These figures show the perceived

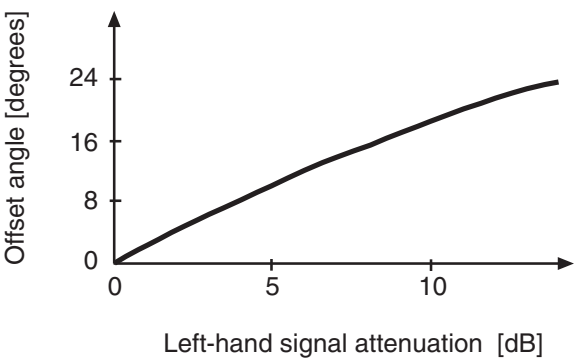


Figure 5.8 Lateral angle position of the phantom source for a mono sound source reproduced by two loudspeakers at $\varphi = \pm 30°$ as function of the reduced level of the left-hand loudspeaker. (After Ref. 5.6)

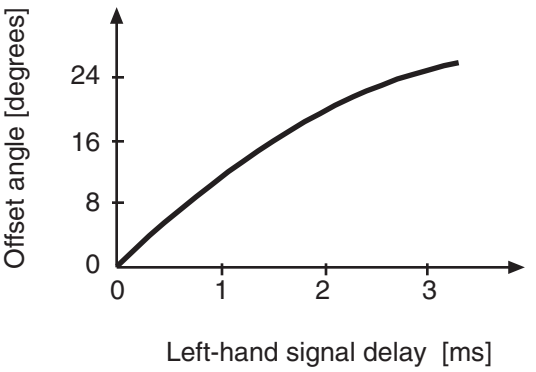


Figure 5.9 Lateral angle position of the phantom source for a mono sound source reproduced by two loudspeakers at $\varphi = \pm 30°$ as function of the extra time delay of the left-hand loudspeaker. (After Ref. 5.6)

location (receiving angle) of a single channel sound radiated by a stereo loudspeaker pair, but where the signals to the respective loudspeakers have been treated to obtain various level and time delays.

Figure 5.8 shows how the perceived angle of incidence depends on the relative level difference between the loudspeaker sounds. Analogously, Figure 5.9 shows how the perceived angle of incidence depends on the relative time difference.

A further discussion of the importance of the balance between level and time differences for stereo sound systems can be found in Chapter 17.

Echo

If the time delay between two coherent signals is long enough, one will hear *echo*. By using the same approach as in the experiments described in conjunction with Figures 5.8 and 5.9 but with both signals arriving from (0°, 0°), one has obtained the thresholds of echo perception shown in Figures 5.10 and 5.11. These results were obtained using speech as a test signal. The results indicate that the sensitivity to echo depends on speech rate, relative delay time, relative sound level, and reverberation time. The most important result shown is that there is a clear *echo threshold* for relative delays larger than around 50 ms that correspond to path length differences larger than approximately 17 m. The risk of echo makes it important to avoid concave, focusing surfaces in auditoria.

The results shown in Figure 5.10 also indicate that if the delay time is short, less than 20 ms, it will be possible to increase the intensity of the delayed signal relative to the direct signal without echo. This result is a further reminder of the temporal masking properties of hearing discussed earlier.

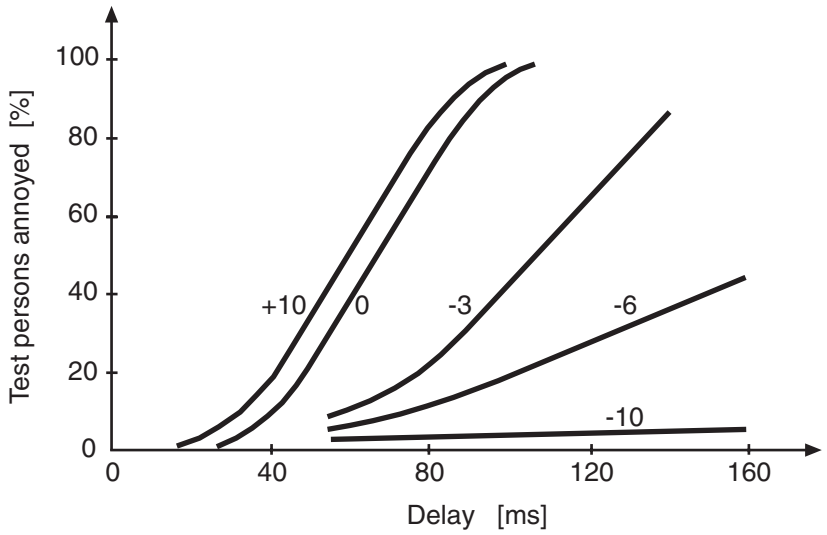


Figure 5.10 Percent of test persons who stated that they were annoyed by echo in listening to a direct sound in combination with a delayed sound ($\varphi = 0^{o}$) with the relative level of the echo to the direct sound as parameter. (After Ref. 5.7)

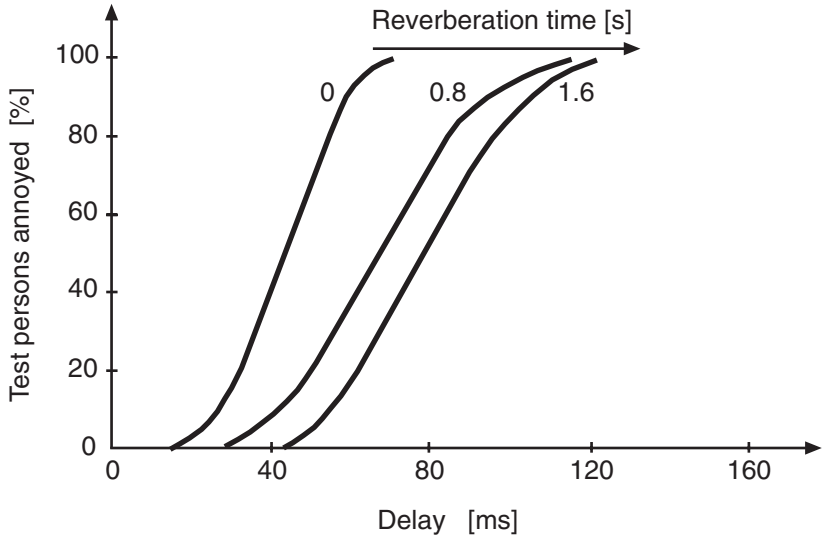


Figure 5.11 Percent of test persons who stated that they were annoyed by echo in listening to a direct sound and reverberation in combination with a delayed sound ($\varphi = 0°$), with reverberation time T as parameter. (After Ref. 5.7)

One will quickly notice that echoes are more annoying at high frequencies than at low frequencies. At high frequencies, there is simply more time available (more periods) to form an echo perception even with short time delays.

Coloration

When listening to music, *coloration* (timbre change) is usually more of a problem than echo. This is probably due to the typically slow modulation of classical music, although there are musical instruments that, when played in concert, are used to generate short sounds. At one extreme, one has slow organ tones, at the other, short trumpet passages.

Coloration particularly is a problem when binaural hearing cannot be used to separate the directions of incoming sounds, for example, when one is listening to a monaural reproduction of a stereo recording (particularly in AB-type recordings, see Chapter 17), or in auditoria where there are strong sound reflections from the ceiling or from the audience area (see Reference 5.8).

Figure 5.12 shows how the repetitive frequency response variations are introduced by various types of repetitive finite- and infinite-impulse response filters. For example, finite-impulse responses are found in the case of sound field contributions that are correlated, somewhat time delayed, and coming from the same direction. Repetitive infinite-impulse responses are obtained when one has two highly reflective parallel walls dominating the sound field in a room. The acoustic effect is, in this case, called a *flutter echo* or *comb filter effect*.

Comb filter coloration is not heard when the sound field is diffuse. Binaural hearing may resolve monaural ambiguities leading to comb filter effects. Figure 5.13 shows the critical values of the gain factor g for audibility of coloration for two types of filters as a function of the delay time t_0.

Another form of coloration is that which is generated by the sound field in small rooms having sparse, weakly damped room resonances at low frequency. In such rooms, there will be frequency shifts in the reverberation since the few modes seldom will have natural frequencies that correspond exactly to those of the audio from the loudspeakers.

Sequences of Reflected Sound

In a real room, having more than one sound-reflecting surface, there will be many sound field components—natural room reverberation is an infinite process. The impulse response at some point in the room can be regarded as the relationship between a short sound pulse at 1 m distance from an omnidirectional source (electroacoustic transducer or spark, for example) and the actual pressure response at the measurement point. By using directional microphones, other types of impulse responses may be measured and used for room acoustic analysis.

Of course, it is the binaural impulse response pair—*binaural spatial impulse response* (BSIR)—rather than the omnidirectional response, that is of real interest. However, at this time there is little knowledge regarding how such a measured BSIR can be analyzed, so currently analysis is usually limited to the measured omnidirectional *spatial impulse response* (SIR) (sometimes called the *room impulse response*).

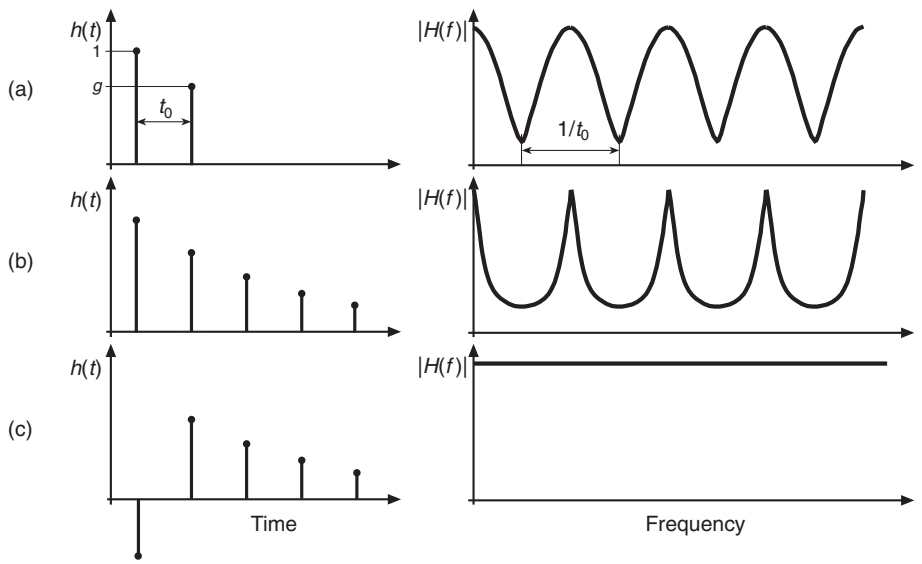


Figure 5.12 Examples of impulse responses and their associated transfer function magnitudes for some comb filters. Delay between impulses is t_0 in all cases. Gain setting is g = 0.7: (a) simple delay, (b) infinite impulse response, repetition with gain g^n, or (c) similar to b but with inverse start transient for all-pass frequency response.

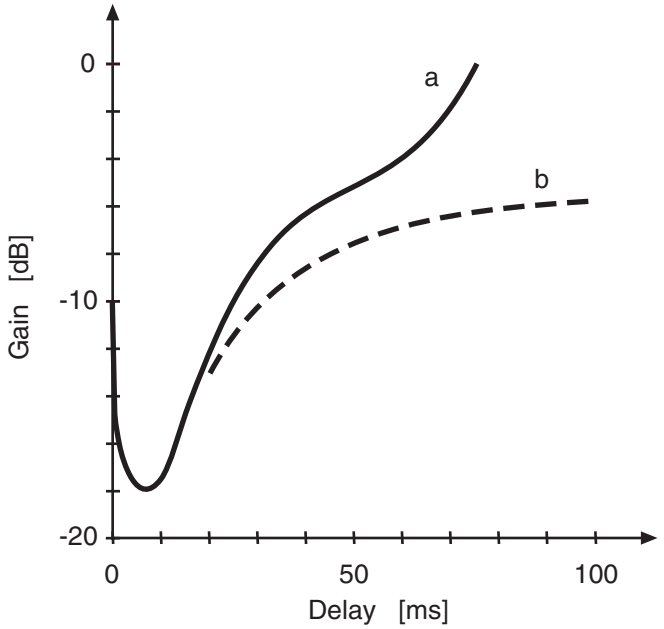


Figure 5.13 Gain factors *g* which give just perceptible coloration of white noise: (a) impulse response $h(t) = \delta(t) - g\delta(t-t_o)$; (b) impulse response $h(t) = \Sigma g^n \delta(t - nt_o)$ (n = 0, 1, 2, 3, 4, . . .). (After Ref. 5.8)

One can show that it is possible to obtain the true reverberation curve by *backward integration* of the SIR. True, here, means the reverberation curve that would be obtained by averaging an infinite number of (squared) impulse response measurement results for the room. The backward integration process is currently the common way to measure reverberation characteristics (see Reference 5.17).

An example of a measured impulse response for a large room is shown in Figure 5.14. The expected discrete reflections are not found in the graph. This is due to the multitude of reflections, the limited resolution of the graph, and primarily because the loudspeaker and sound-reflecting surfaces have a frequency-dependent phase and amplitude properties. Since the SIR behavior is so complex, it is current practice to use various types of energy-based metrics to describe the relevant properties of the room (see Section 5.3).

If one removes or alters sections of the SIR or BSIR and convolves the resulting impulse responses with audio, it is possible to listen to changes in the properties of the room. This is an attractive technique to study the importance of various parts of the impulse responses.

Reverberation Time and Subjective Preference

The reverberation time is the most useful of the room acoustic metrics to judge the acoustical properties of a room and its suitability for a certain use. Experience has shown that the reverberation time that is suitable for various uses and sizes of rooms can be approximated as shown in the graph in Figure 6.9.

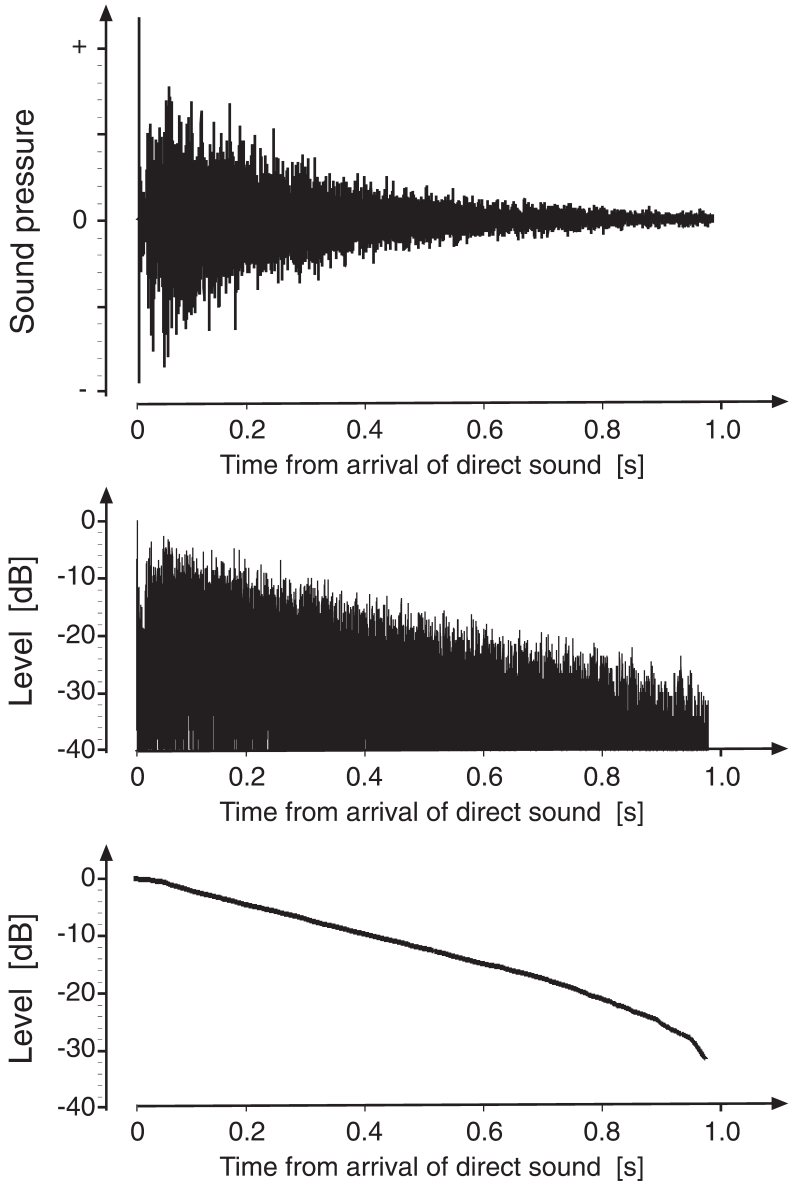


Figure 5.14 Three ways of representing the same measured sound decay characteristics. The top curve shows the sound pressure drop (impulse response); the middle curve shows the SPL drop (echogram); and the bottom curve shows the reverberation curve obtained by reverse summation of energy in the top curve.

Human ability to discriminate between different reverberation times is shown by the curve in Figure 5.15, which shows the average ability to sense a certain change in reverberation time ΔT for various reverberation times T. One can see that for most of the range, a reverberation time change of less than 5% can be heard.

In practice, one finds that people prefer particular ranges of reverberation times for various audio. Speech usually needs short reverberation times, whereas music can need reverberation times from nearly zero to several seconds, depending on its type. Since the reverberation of a room contributes to the loudness of sound, some reverberation is usually required for an optimum speech intelligibility. The reverberation times in rooms for speech should generally not be lower than approximately 0.6 s at the center frequencies; if the reverberation time is shorter, the rooms will sound dry. Without reverberation, the sound quality of speech and music would be dependent on the directivity of the sound source.

The ability of hearing to perceive changes in reverberation time is surprisingly good. A reverberation time change of less than 5% can be noticed (around 1–3 kHz) (see Reference 5.9). Usually one strives to achieve control of the reverberation time to an interval of ± 0.1 s at midrange frequencies.

The reverberation time in rooms for classical music (without amplification) should increase somewhat at low frequencies from 250 Hz and downward; the reverberation time in the 63 Hz octave band may be around 50% higher than in the 1 kHz octave band.

Rooms for speech and rooms used for amplified sound (such as speech or music) should generally not have any reverberation time increase at low frequencies. For these types of rooms, one should strive

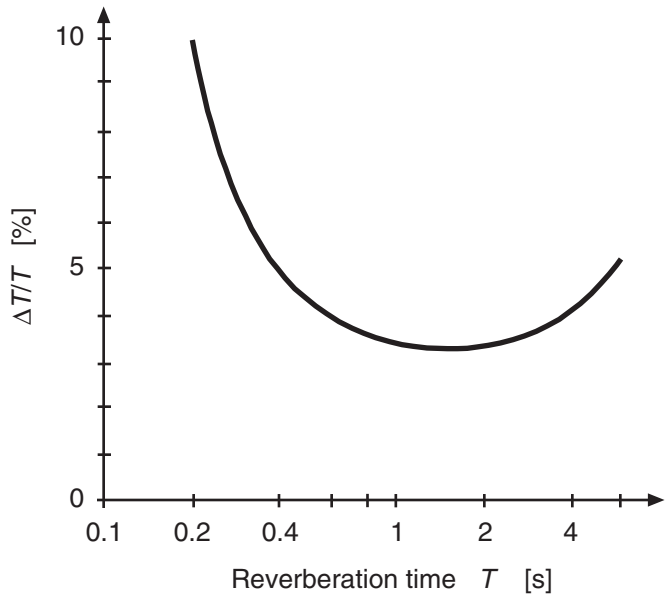


Figure 5.15 Barely noticeable difference in percentage for various reverberation times. (After Ref. 5.9)

for a flat, frequency independent, reverberation time curve. These rooms otherwise have a tendency to sound *boomy*.

One must also be aware of the difficulties inherent in trying to measure reverberation time with audience and performers in a room. The measurements take some time and the audience expects to be entertained rather than be part of an experiment. Considerable experience is required to expertly judge the difference in reverberation characteristics between when a room is occupied and when it is unoccupied.

Early and Late Reverberation

Reverberation time is defined as the time it takes for the energy density in a room to reach a level of −60 dB below that of the start of a sound decay, after a switching of the sound source. Since it is difficult to measure the energy density, the reverberation curve is measured using the sound pressure level decay characteristic, and the results obtained at various locations are averaged.

Psychoacoustic experiments indicate that hearing is primarily focused on approximately the first 0.15 s of the reverberation process in *running* music or speech. This can be understood from the importance of masking (see Figure 5.16). It is only at the end of a piece of music that one will hear the entire reverberation and how the reverberation recedes into the background noise of the room. Usually, at least

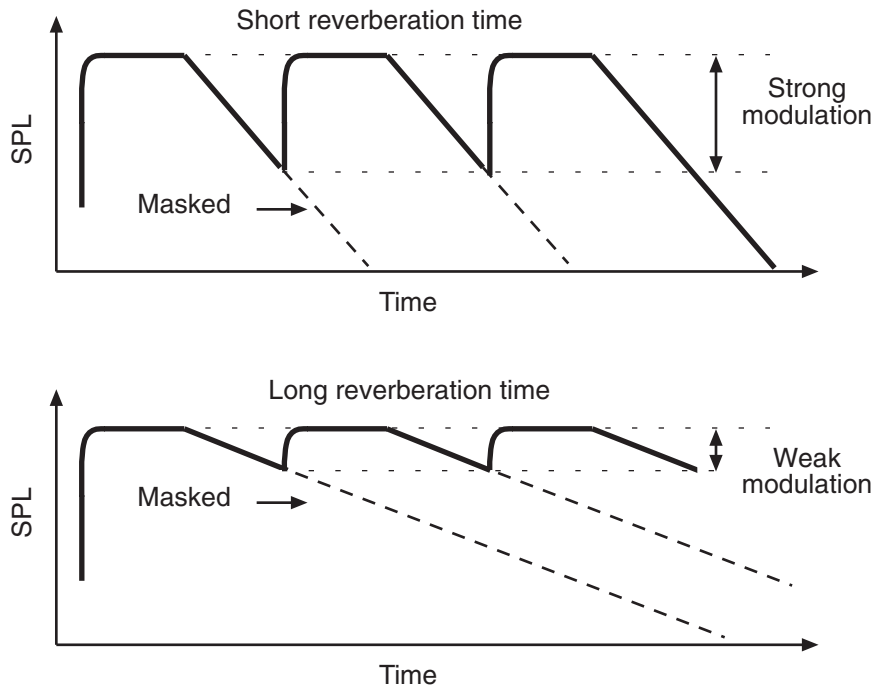


Figure 5.16 Reverberation by previous sound masks later sound. The modulation depth decreases as a result and less detail is heard.

for classical music, this is an important part of the music, so both the *early* and the *late* reverberation characteristics are important when designing high-quality auditoria.

Figures 5.17 and 5.18 show the results of experiments regarding the judgment of perceived reverberation time versus reverberation time measured in various ways. One sees that the early reverberation time is a reasonably good measure of the perceived reverberation time, particularly if judged over the interval of the first 0.16 s of the reverberation process instead of over the first 15 dB of the process.

These results underline the importance of a suitable acoustic design of the room geometry, since it will be the room geometry and the acoustic properties of the walls by which the sound is initially reflected that are important to the early part reverberation process.

Electroacoustic equipment can be used to change the properties of the reverberation process. This is, in effect, what is done when using sound reinforcement systems (see References 5.18 and 5.19).

Dual-slope Reverberation Curves

When doing reverberation time measurements, one quickly notices that the sound level decay curve is not a linear function of time, particularly during the time in which sound propagates the first few mean-free-path lengths. Quite often, one also finds *bends* in the slope of the reverberation curve, even when the sound in the room has become quite diffuse (see Figure 4.7).

Such reverberation curves were originally referred to as *broken* but now the term multi-slope is used. In the case of only one bend, the term is *dual-slope curve*. The reason for the dual-slope is usually that the characteristics that affect such things as the decay, absorption, diffusion, mean-free-path length, and volume are different in various subspaces of the volume. This is particularly the case for rooms in which several smaller subspaces make up one larger space. Churches and cathedrals often have such coupled subspaces.

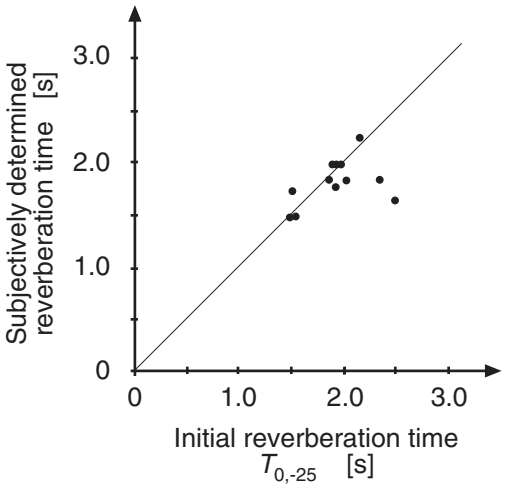


Figure 5.17 Measurement results for the subjectively perceived reverberation time as a function of the reverberation time of the first 25 dB of the reverberation process. (After Ref. 5.10)

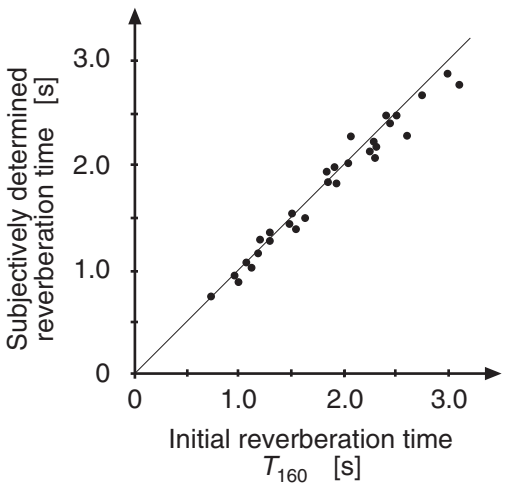


Figure 5.18 Measurement results for the subjectively perceived reverberation time as a function of the reverberation time of the first 0.16 s of the reverberation process. (After Ref. 5.10)

Many people have come to like the acoustics characterized by dual reverberation slopes. The reason for this is probably that when listening to music in the home, over stereo or multiloudspeaker systems, the reverberation in the recording is mixed with the reverberation of the listening space. Since these are seldom the same, the resulting reverberation curves will be characterized by many slopes. Dual-slope decays also make it possible to create listening situations that have extremely good clarity coupled with long reverberation times and much ambience. Some modern concert halls are expressly designed to have acoustically coupled subspaces that have much longer reverberation times than the main auditorium. One prerequisite for the audibility of such long, final slope, reverberation in auditoria, such as concert halls, is that the auditorium features low background noise.

Subjectively Diffuse Sound Fields

It is important to realize that sound fields may appear subjectively diffuse without being diffuse in the physical sense (discussed in Chapter 4). Even a sound field composed of only two components can be perceived as diffuse, for example, with coherent noise from two diametrically placed loudspeakers, at angles (90°, 0°) and (270°, 0°), when loudspeaker two is fed loudspeaker one's signal slightly delayed (approximately 1 ms or more). It is therefore best to be careful in the use of the term diffuse and make it clear whether it is meant in the physical or subjective sense.

The degree of *coherence* between the signals at a person's ears is important for the perception of the subjective diffuseness of a sound field. Of course, a sound field in which there is much frontal direct sound will have higher coherence than a sound field primarily consisting of reverberant sound.

Experiments have shown that the degree of spaciousness experienced when listening to a sound field is related both to the coherence of the signals at the ears and to the *level difference between direct sound and reverberant sound* (German: Hallabstand), H, as defined by:

$$H = L_{p,\text{direct}} - L_{p,\text{reverberation}} \tag{5.1}$$

Figure 5.19 shows some experimental results concerning how the *interaural cross-correlation* (IACC) influences the subjectively perceived spatial extension, as the reported *solid angle* distribution of sound in the upper half space when listening to a band-limited noise signal (see Reference 5.11). One can notice that the spatial extension, which here can be regarded as a measure of the degree of subjectively perceived diffuseness, increases as the value of IACC decreases.

Figure 5.20 shows the result of experiments in which the level difference between direct sound and reverberant sound was changed and how this affects the spaciousness of the sound field. One unit of change on the spaciousness scale corresponds to the *just noticeable difference*—that is, where 50% of the time the test persons were able to discern a difference.

The importance of the early part of the reverberation in relationship to H is also shown by the curves in Figure 5.21. The curves show the subjectively judged reverberation time T_{subj} as a function of H, with reverberation time T as parameter. It confirms the possibility of reducing the subjectively perceived reverberation time by addition of strong early reflections.

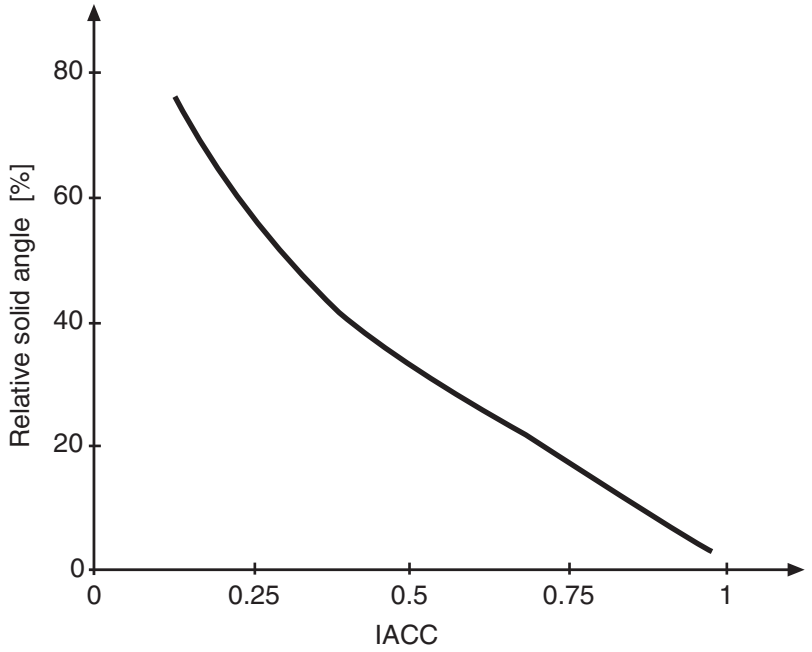


Figure 5.19 Perceived solid angle of the sound field as a function of the interaural cross-correlation. (After Ref. 5.11)

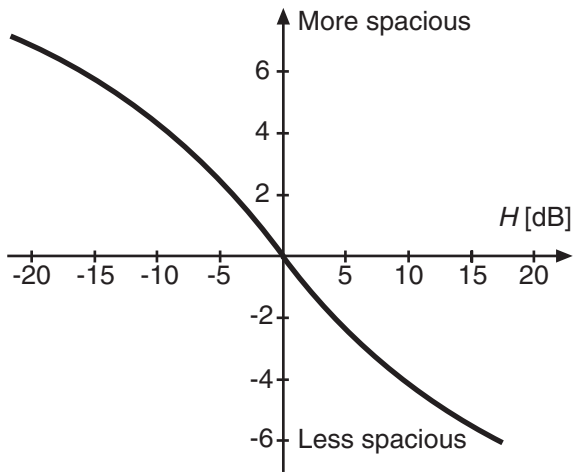


Figure 5.20 Subjectively perceived steps of spaciousness as a function of the level difference *H* between direct and reverberant sound. Note that high spaciousness is obtained for large negative values of *H*. (After Ref. 5.12)

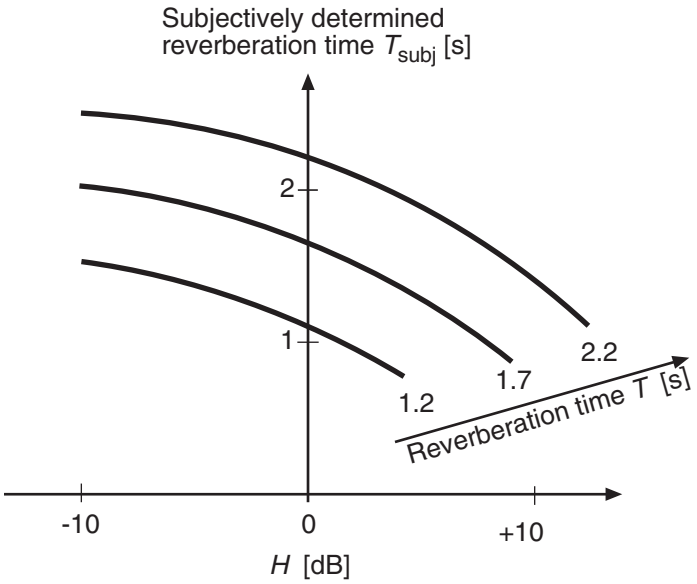


Figure 5.21 Subjectively perceived reverberation time T_{subj} as a function of H with reverberation time T as parameter. (After Ref. 5.12)

5.3 METRICS FOR ROOM ACOUSTICS

It would be advantageous if it was possible to specify the acoustic quality of a room by some simple numbers or indicators, for example, when writing building specifications or for quality control. It turns out that this is difficult because of the lack of sufficient knowledge about the spatial properties of human hearing. A room may show large numerical differences between the current quality metrics measured at various positions in the room while the subjective experience of listening in the room does not bear out these differences. It is particularly the fine structure of the reverberation that is difficult to measure. In experiments done in the laboratory, the influence of visual cues can be removed, but the sound field must be approximated. When doing experiments in real auditoria, it is difficult to remove the visual cues and other cognitive factors.

Nearly all of the currently used metrics for room acoustics are based on the measurement of SIR. The measured SIR, h, is a convolution of the impulse responses of the loudspeaker, microphone, and other parts of the measurement system (see Chapter 2).

$$h(t) = h_{\text{lsp}}(t) * h_{\text{room}}(t) * h_{\text{mic}}(t) * h_{\text{other}}(t) \tag{5.2}$$

where h_{lsp}, h_{room}, h_{mic}, and h_{other} are the impulse responses of the loudspeaker, room, microphone, and other equipment (such as electronic filters and sound cards) involved in the measurement process.

The influence of these other, undesired impulse responses is generally disregarded when considering the room acoustics metrics. For simplicity, the room impulse response is often set as the raw measured impulse response or filtered using a suitable bandpass filter, such as an octave band filter. It is important to note, that such filters will have group delay that must be compensated. Using a process called deconvolution, it is possible to eliminate some of the problems due to the electroacoustic equipment.

Along with the music room acoustics metrics listed here, there are also various speech-related metrics discussed in Chapter 3 as well as various metrics related to stage acoustics. References 5.13 and 5.14 are useful for their discussion of the various metrics mentioned in upcoming text.

Strength Index

The most important factor in determining acoustic quality is loudness. A metric designed to estimate the relative loudness of sound in a room is the *strength index G*, sometimes called the room gain. The strength index is based on the measured SIR $h(t)$ and is defined in the following way:

$$G = 10\log\left(\frac{\int_{0\text{ms}}^{\infty} h^2(t)\,dt}{\int_{0\text{ms}}^{5\text{ms}} h_A^2(t)\,dt}\right) \quad [\text{dB}] \tag{5.3}$$

where $h_A(t)$ is the measured impulse response for the direct sound component only, at 10 m distance in anechoic space. Since it is difficult to find such large anechoic spaces, the measurement is usually done at a shorter, convenient distance and then scaled to compensate for the geometric strength difference due to the distance difference, assuming that the level drops by −6dB per distance doubling.

Reverberation Time

From the viewpoint of subjective importance, after loudness, reverberation time is generally the property of a room that is the most important for the subjective impression of room acoustics.

According to the ISO 3382 standard the reverberation time is obtained by extrapolation to 60 dB of a line fitted to the reverberation curve from −5 dB to, for example, −35 dB of decay. The reverberation time is then often written as $T_{-5,-35}$. It is difficult to obtain a good dynamic range when measuring reverberation time in large venues. Unless otherwise noted, the room impulse responses are to be measured using an omnidirectional microphone. The standard also requires the measurements to be done using at least two sending positions and five receiving positions, spaced in a specified way on the stage and in the audience seating area.

Definition

Definition (*D*) (German: Deutlichkeit) is a metric used when judging the suitability of rooms for speech. The Definition metric is based on the assumption that the sound arriving during the first 50 ms of the reverberation process is useful (beneficial to speech intelligibility), whereas the rest of the sound is detrimental. Definition has been shown to correlate reasonably well to speech intelligibility. Definition is defined as:

$$D = 100\frac{\int_{0\text{ms}}^{50\text{ms}} h^2(t)\,dt}{\int_{0\text{ms}}^{\infty} h^2(t)\,dt} \quad [\%] \tag{5.4}$$

There are many other similar metrics, such as Clarity. Since Definition does not take the signal properties of sound into account, it is not as good a metric as modulation damping and speech transmission index. It also does not take the influence of the background noise of the room into account.

Clarity

Clarity C_{80} is a metric similar to Definition but adapted to evaluation of rooms for music. It is defined as:

$$C_{80} = 10\log\left(\frac{\int_{0\text{ms}}^{80\text{ms}} h^2(t)\,dt}{\int_{80\text{ms}}^{\infty} h^2(t)\,dt}\right) \quad [\text{dB}] \tag{5.5}$$

The index *80* marks the split between the early and late parts of the reverberation process, in this case taken as 80 ms.

Sometimes the split is taken at 50 ms and Clarity C_{50}, is calculated, and used in place of Definition. The two main differences between C_{50} and Definition are the use of different integration times and the use, for C_{50}, of the log of the ratio of energies. The reason for using C_{50} instead of C_{80} is agreement with the time constant used for *D*.

Interaural Cross-Correlation

The IACC is a metric describing the dissimilarity between the left and right ear parts of the binaural room impulse response. One is usually interested in the reverberation sounding diffuse. Since reverberation diffuseness is closely connected to correlation, it is reasonable to assume that it is advantageous that the late reverberation part of the SIR has as low an IACC as possible.

The IACC is defined as:

$$\text{IACC} = \max_{|t|<1\text{ms}} \frac{\int_{0\text{ms}}^{\infty} h_{LE}(t)h_{RE}(t+\tau)\,dt}{\sqrt{\int_{0\text{ms}}^{\infty} h_{LE}^2(t)\,dt \int_{0\text{ms}}^{\infty} h_{RE}^2(t)dt}} \tag{5.6}$$

where $h_{LE}(t)$ and $h_{RE}(t)$ are the left ear and right ear measured room impulse responses respectively. These are measured using microphones mounted in the ears of a manikin (see Figure 5.22 and 17.6).

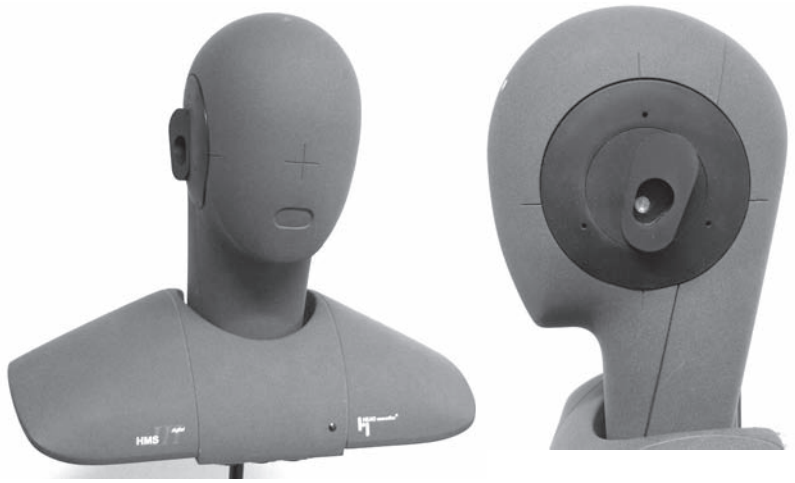


Figure 5.22 A manikin for binaural recording and measurement, and its ear replica, showing the microphone at the bottom of the imitated concha cavity. (Photo by Mendel Kleiner.)

It is important to note that IACC can be used with many different types of test signals and with bandpass-filtered impulse responses. Care is advised when using IACC data and specifying and measuring IACC. Also note that sometimes the IACC is measured over parts of the SIR, for example 0–80 ms and 80–∞ ms.

Originally, IACC was intended for use with binaural recording of running music so that the metric described what was happening in an actual listening situation.

Recently the use of the quantity 1–IACC (in octave bands) has become common (see Reference 5.13).

Lateral Energy Fraction

As discussed previously, it is much easier for hearing to separate between signals if they differ in angle of incidence relative to the median plane. A metric designed to separate between sound arriving in and out of the median plane is the *Lateral Energy Fraction* (L_f) defined as follows:

$$L_f = \frac{\int_{5\text{ms}}^{80\text{ms}} h_L^2(t)\,dt}{\int_{0\text{ms}}^{80\text{ms}} h_T^2(t)\,dt} \tag{5.7}$$

where $h_L(t)$ is the room impulse response measured using a bidirectional microphone and $h_T(t)$ is the room impulse response measured using an omnidirectional microphone.

The exact angular weighting factor for the lateral energy fraction is specified as $h^2 \propto \sin(\varphi)$, where φ is defined as shown in Figure 3.2. In practice, the weighting function $h^2 \propto \sin^2(\varphi)$ is usually used since it is difficult to find suitable microphones otherwise.

Center-of-Gravity Time

A problem common to both definition and clarity is sensitivity to abrupt SIR changes at around 50 and 80 ms respectively. The metric *center-of-gravity time*, t_s, is designed to circumvent this problem by not using any fixed *transition time*. The definition is:

$$t_S = 1000\frac{\int_{0\text{ms}}^{\infty} t \cdot h^2(t)\,dt}{\int_{0\text{ms}}^{\infty} h^2(t)\,dt} \quad [\text{ms}] \tag{5.8}$$

The center-of-gravity time (German: Schwerpunktzeit) has not had any success in replacing definition and clarity but is used alongside the two older metrics.

Early Decay Time

The importance of the early part of the reverberation process has been stressed several times. An alternate definition of what constitutes the early decay is used for the metric *early decay time* (EDT) which is based on the extrapolation to 60 dB of the reverberation time during the first 10 dB of reverberation decay. The

results reported in Figure 5.19 speak against the use of EDT as a metric for evaluation of reverberation time. The EDT values for some octave bands are used in the definition of the *brilliance* metric.

Warmth

Warmth is an important requirement for halls for classical music. It is dependent on the relative energy between reverberant sound at low and medium frequencies respectively. Since the relative energy is in turn dependent on the relative sound absorption, it will also depend on the reverberation time. *Bass ratio* (BR) is a way of expressing this relationship.

$$BR = \frac{T_{125\text{ob}} + T_{250\text{ob}}}{T_{500\text{ob}} + T_{1000\text{ob}}} \tag{5.9}$$

Here $T_{125\text{ob}}$ is the reverberation time T in the 125 Hz octave band, and so on.

Typical expected *BR* values are from 1.1 to 1.25 for rooms with $T > 2.2$ s and from 1.1 to 1.45 for rooms with $T < 1.8$ s.

Brilliance

A lack of *brilliance* can be observed in rooms where the high frequency early decay times are much shorter than the midfrequency early decay times. Two metrics have been suggested for brilliance, $B_{2\text{kHz}}$ and $B_{4\text{kHz}}$ for the 2 kHz and 4 kHz frequency bands respectively (see Reference 5.13).

$$B_{2kHz} = \frac{2EDT_{2000\text{ob}}}{EDT_{500\text{ob}} + EDT_{1000\text{ob}}} \tag{5.10}$$

$$B_{4kHz} = \frac{2EDT_{4000\text{ob}}}{EDT_{500\text{ob}} + EDT_{1000\text{ob}}} \tag{5.11}$$

Here the EDT indices refer to the octave band frequencies for which they were measured. The recommended minimum values for $B_{2\text{kHz}}$ and $B_{4\text{kHz}}$ are 0.9 and 0.8 respectively for concert halls.

5.4 Problems

5.1 The function $h(t) = Ae^{-\delta t}$ is sometimes used to approximate the envelope of an ideal SIR.

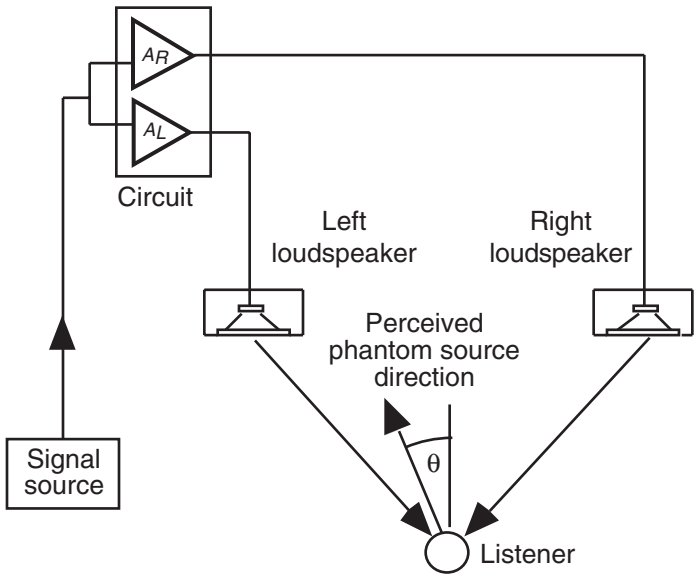
Task: Express the damping constant δ as a function of reverberation time T using the definition of reverberation time.

5.2

Task: Using the function $h(t) = Ae^{-\delta t}$ as the approximation of an ideal SIR envelope, determine how clarity and definition can be expressed as functions of reverberation time T.

5.3 A recording studio uses an omnidirectional microphone for recording and then uses a signal processing mixing console to position the virtual sound source at the desired location between

the listener's loudspeakers. This is done using time and level differences between the right and left channels. The phantom sound source can be placed between the loudspeakers in the sketch below, using different amplification factors A_R and A_L for the two channels, according to Figure 5.9.



There are two restrictions to the circuit:

1. The circuit is only allowed to give angular change—no amplitude change is allowed.
2. For $\theta = 0°$ the amplification factors must be unity.

Hint: Use the fact that the loudspeaker signals are correlated and have the same time-of-flight.

Task: Calculate the correct amplification factors A_R and A_L, to obtain directions θ from 0 to 24 degrees in 4 degree increments (to the phantom source).

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Room Acoustics Planning and Design 6

6.1 INTRODUCTION

The purpose of this chapter is to give examples of the kind of deliberations and concerns that are involved in the planning and design of rooms. Many of the physical and psychoacoustic aspects of room acoustics have already been presented and discussed in Chapters 4 and 5.

The purpose of room acoustics planning and design is usually to provide rooms that allow good communication between people—for example, by providing good acoustics for speech and music. This requires not only specific acoustic properties regarding the way the room guides the sound from the speaker or performer to the audience and other orchestra members, but also that competing sounds, such as noise are suitably controlled.

The requirements for the way in which sound is distributed in the room are not only expressed by sound level and sound level distributions but are also expressed with timbre and temporal characteristics. The temporal characteristics are a result of the room size and shape, as well as the way the audience and other sound-absorbing and scattering areas are distributed over the room's boundaries. It is important to note that there are substantial interindividual preference differences between people, and that the psychoacoustics data presented in Chapter 5 is not exact. Much room acoustic data still needs to be collected and compared to personal preference for improving the mapping between the physical domain of room design and the subjective domain of acoustics appreciation.

Performers and audiences may have conflicting requirements regarding suitable acoustics. Solo performers, such as speakers and singers will usually want to sense the acoustic response of the auditorium, whereas musicians playing in groups or orchestras will usually put more emphasis on the way the room allows them to interact.

Background noise levels must also be sufficiently low to allow good communication. In rooms for music, low levels of background noise are needed not only to keep the noise from masking the *running* music, but also so that people will be able to hear the full dynamic range and spatial qualities of the reverberation processes. Background noise can further influence and predispose people's reactions to the acoustic quality of the room. The sound quality of the noise will be different depending on whether the noise is due to heating, ventilation, and air conditioning noise (usually called HVAC noise), to other activities in the building, or to traffic, neighboring industry, and other exterior noise. This leads to different noise criteria (NC) being necessary depending on the noise and the use of the room.

For many types of rooms, the room acoustical conditions are so important that acoustic planning and design must be introduced in the early sketching stage. Cooperation between architect and acoustician is necessary for excellent acoustics. The possibilities of changing the general room acoustics of a room are limited once the room exists physically, without building a new interior shell or implementing other drastic changes.

Electronic architecture—that is, active sound field control, using microphones and loudspeakers in the room in conjunction with electronic signal processing—may sometimes be a way to improve acoustical conditions so that the room may be useful for some purpose other than the ones allowed by *natural* or *passive* acoustics. *Sound reinforcement systems* are often used to provide better conditions for listening to speech in acoustically difficult environments, such as those characterized by excessive distance or reverberation, or inadequate sound reflections.

It is important to stress that the quality of room acoustics is difficult to measure. Chapter 5 discussed some metrics for room acoustics quality. These metrics often require considerable experience and expertise to be applied correctly. Often, listening and subjective judgment by experts is the only way to assess the room acoustics quality. Hearing is an important measurement device.

6.2 BASIC REQUIREMENTS FOR GOOD ROOM ACOUSTICS

Quiet: Acceptable Noise Levels and Noise Sound Quality

The requirements of noise that does not damage hearing were discussed in Chapter 3. In designing the acoustics of rooms for homes, offices, industry, and entertainment, additional NC must be met. What is appropriate will depend on the use of the room and on the requirement for visual and auditory impressions to be matching. We expect a concert hall to be quiet, whereas we will not be surprised to find an industrial workplace somewhat noisy. Sometimes the noise is added intentionally as *acoustic perfume*, such as Muzak-type background music, broadband noise having an unobtrusive spectral character, or even intentional HVAC noise.

Typically, noisy environments may be found in schools, offices, industry, and restaurants. From the viewpoint of the acoustician, these environments require special consideration: limiting noise so that it does not cause hearing loss and is acceptable for the use of the room and designing sound reinforcement and other sound amplification equipment for maximum speech intelligibility and other program enjoyment.

Noise level requirements will vary between rooms and room uses. Different noise criteria typically must be applied to office equipment for use in small offices and in office landscapes. The maximum noise is often specified by the maximum allowable A-weighted sound pressure level, without considering the spatial characteristics of the sound field relative to the person being exposed to the sound. The sound pressure at the entrance of the ear canal will typically vary by at least ± 10 dB at frequencies above 2 kHz due to the angle of sound incidence.

Not only does a 10 dB sound pressure level increase in the midfrequency range typically correspond to more than a doubling of loudness, the change may also considerably increase the risk of noise-induced hearing loss and influence *task performance*. It is also important to note that an increase in A-weighted sound pressure level does not necessarily increase *annoyance*. The temporal and other char-

acteristics of the noise are important in this respect. One often uses various metrics of *Product Sound Quality* to further describe the properties of noise, see Chapter 17. The noise may also be described using special terminology usually specific to a certain engineering field; for example, in describing the noise of air handling equipment, words such as hiss, rumble, and roar are used.

It is important to stress the need for low background noise. Most noise criteria have been set with persons having normal hearing in mind. In societies with much sound-generating equipment (mechanical as well as electroacoustical), people are subject to extra noise-induced hearing impairments. Persons having hearing impairments or hearing loss generally have difficulty in understanding speech buried in noise and are often more annoyed by noise than persons having normal hearing.

Noisy Environments

Much of the negative influence of heating, ventilation, and air conditioning noise on speech is typically due to masking. Frequency domain masking is primarily active upward in frequency as discussed in Chapter 3, where metrics such as signal-to-noise (S/N) ratio, articulation index, speech transmission index, among others, were introduced. As indicated by Figure 3.34, it is reasonable to expect that the S/N ratio (assuming typical speech and HVAC noise spectra) must be at least 15 to 25 dB for speech intelligibility to be unaffected by noise in the speech frequency range.

The Speech Interference Level (SIL) can be used to estimate the seriousness of noise affecting speech. SIL is defined as the mean of the 0.5, 1, and 2 kHz octave band noise levels (see Reference 6.1). Using the data shown in Figure 6.1, one can roughly determine the interference by a particular noise on

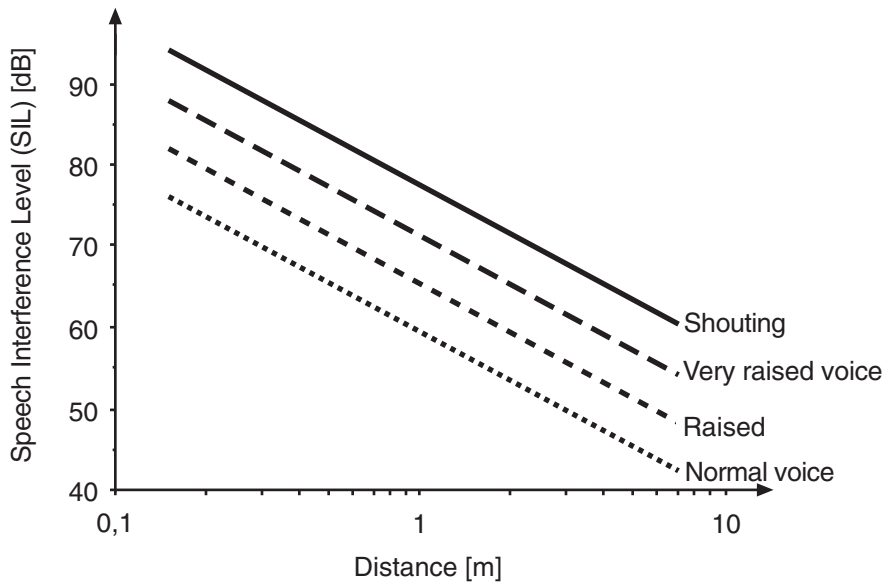


Figure 6.1 The Speech Interference Level (SIL) is defined as the mean of the sound pressure level values for noise spectrum in the 0.5, 1, and 2 kHz octave bands. The graph shows SIL for speech at various distances and for different voice modes (after Ref. 6.1).

speech, not only for offices and other workplaces but also for telephone transmission and similar cases. The data can also be used to determine the necessary sound pressure levels for sound reinforcement equipment.

Quiet Environments

Some environments are characterized by communication activities that are particularly sensitive to noise. This is particularly true for audio and video recording studios, radio and television stations, concert halls, operas, theaters, and similar venues. The background noise limits are usually set during consultation between director, fundraiser, architect, and acoustician.

In rooms for music in particular, the noise levels tolerated are very low. Curves, such as those shown in Figures 6.2 and 6.3, are often used in determining whether a noise spectrum is acceptable, or not. Such curves, specifying maximum octave band sound pressure levels from 31 Hz to 16 kHz, are useful because they, to some degree, take the character of noise into account.

In Europe, it is common to specify the maximum allowable noise level with dBA values. This is inferior compared to the U.S. way of specifying the maximum allowable noise as an NC or RC value. The noise criterion (NC) curves, shown in Figure 6.2, and the A filter curve are based on auditory threshold data. Noise shaped along the NC curves has a tendency to sound both *rumbly* and *hissy*. The NC curves were not shaped to have the best spectrum shape but rather to permit satisfactory speech communication without the noise being annoying (see Reference 6.2).

The room criterion (RC) curves shown in Figure 6.3, on the other hand, have been shaped to be perceptually neutral—they are straight lines with a slope of −5 dB/octave. The RC method involves the determination of both an RC rating and a *spectrum quality descriptor* that determines if the spectrum is rumbly or hissy (see Reference 6.2).

In some cases, one has to be content with a value for sound level L_{pA}. Concert halls usually require sound levels well below 20 dBA. The noise of microphones and HVAC equipment typically have very

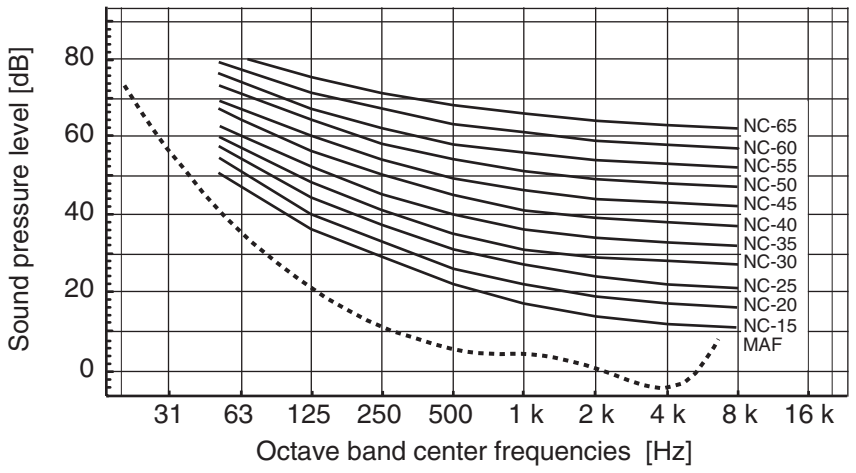


Figure 6.2 These NC curves apply to octave band-filtered noise levels and are often used in room acoustics (after Ref. 6.2).

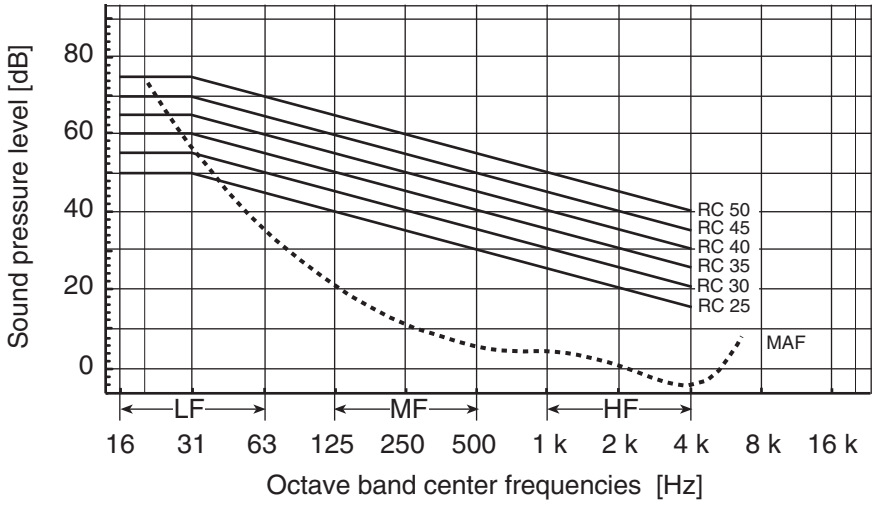


Figure 6.3 The RC curves also apply to octave band-filtered noise levels. Note that there are several editions of these curves—the ones shown are the Mark II curves (after Ref. 6.2).

different spectra. Comparing A-weighted sound levels for different spectra shapes is not relevant from the viewpoint of sound quality, as in the following example.

The acoustic background noise in a sound recording studio and the equivalent acoustic-electronic-background noise of microphones usually have different spectral shapes. Although one can say that the A-weighted sound level of microphone noise should be below the acoustic noise level in the recording room, the fact is that the noises will be perceived differently due to their different sound quality characteristics. The acoustic noise (for example, HVAC and traffic) will generally have spectra dominated by low-frequency noise, whereas microphone noise (due to its electronic generation mechanism) tends to have a *white spectrum*—that is, the octave band levels increase by 3 dB per octave. This gives the microphone noise subjectively a hissy character. Therefore, a microphone having an equivalent 20 dBA sound level internal noise level will probably sound hissy to the recording engineer when used for recording in a concert hall having an acoustic background sound level of 20 dBA. This may be a considerable problem when recording music in large concert halls, such as the one shown in Figure 6.4.

The recognizable properties of noise are usually important in the environments under discussion here. It may sometimes be easier to tolerate a noise having a random character than a noise that is transient, has a specific time pattern, or carries information content. The spatial distribution of the noise is also important. A diffuse sounding noise is often less disturbing to some people than a noise that can be localized in space (although in the latter case, one can perhaps more often find it possible to eliminate the noise source).

It is also important to avoid audible pure tones in the noise spectrum—the audibility of such tones was discussed in Chapter 3. It is reasonable to require the tones to be inaudible, which requires their sound pressure level to be below the critical band level of the other noise in the band. In practice, this results in the need for such tones to be at least 5 dB below the octave band level of the rest of the noise in the band.



Figure 6.4 In a modern concert hall, such as Symphony Hall, Birmingham, United Kingdom, the background noise sound level L_{pA} may be as low as 11 dB. (Photo by Mendel Kleiner.)

6.3 FUNDAMENTALS OF ROOM ACOUSTIC PLANNING

Rooms for Speech

The speech intelligibility properties of rooms intended for verbal communication are important, but it is essential to remember that good speech intelligibility is not the same as naturally sounding speech. One can increase the intelligibility of amplified speech, particularly in noisy surroundings, by various types of signal processing, such as spectral shaping and compression.

For our purpose here one can regard speech as a modulated signal with a complex tonal or noisy spectrum, primarily covering the frequency range 0.25 to 4 kHz that is modulated by frequencies in the 0.2 to 8 Hz range.

Reverberation and noise will decrease the *modulation strength* (also known as the *modulation depth*). A reduction of modulation depth, corresponding to the attenuation of modulation strength (sometimes called the *modulation damping*), results in a reduction of speech intelligibility.

For speech modulation at the listener to not be affected by reverberation, the reverberation time must be short, and the ratio between direct and early reflected sound to late reverberant sound must be high.

There is, however, a practical limit to shortening the reverberation time. Besides the expense of adding sound-absorbent treatment to the room, there is also another aspect. People expect rooms to have a certain reverberation time and level, determined by tradition and visual aspects (e.g., room purpose, shape, and volume).

Speech intelligibility can be shown to increase with increasing reverberation time for very short reverberation times in the range of up to 0.5 s. The reason for this speech intelligibility dependency on reverberation is that reduced sound absorption in the room leads to both an increased reverberation time and an increased sound pressure level. In addition, early sound reflections by more reflective walls

help eliminate the influence of speaker sound radiation directivity on the sound pressure. The directivity index of the human voice is approximately 6 to 10 dB in the frequency range that is most important for speech intelligibility.

These are some good rules for auditoria used primarily for the spoken word:

- Keep the travel paths of direct sound and important early reflections short. In practice, it is difficult for an untrained speaker to reach audiences with sufficient speech intelligibility at distances over 20 m without the use of amplification.
- There must be enough early reflections to provide sufficient sound level while at the same time not feeding sound energy into the late part of the reverberant sound field. It is advantageous to strive for early sound reflections to arrive close to the horizontal plane, rather than from above, to have good speech sound quality. Sometimes for practical reasons, however, it is necessary to make use of overhead sound reflectors as shown in Figure 6.5.
- Mirror-like (specular) reflections must not exceed the level of the reverberant sound if they arrive more than 30 ms after the arrival of the direct sound, such reflections may be perceived as echoes. This means that sound being focused by concave surfaces must have short propagation. The focusing effect may be remedied using alternate focal distances and sound-absorptive or scattering treatments in the travel paths as shown in Figure 6.6.



Figure 6.5 A modern lecture hall equipped with sound-reflecting panels over the sending end of the room in Chalmers University of Technology, Gothenburg, Sweden. (Photo by Mendel Kleiner.)

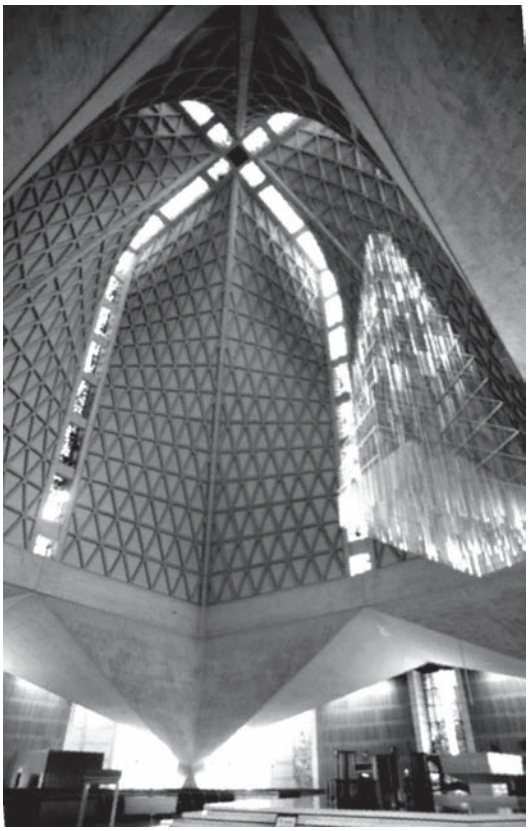


Figure 6.6 When the sound absorption of the walls is high, a room having high concave ceilings is susceptible to echoes. The ceiling in this church has been made extremely irregular to scatter sound and avoid echo. (Photo by Mendel Kleiner.)

- Take the directivity of the human voice into account. In small rooms, such as classrooms and small auditoria having relatively hard walls, the speaker will usually be quite close to some sound-reflecting surfaces. This leads to strong early reflections that will add to the direct sound in such a way as to increase the speech intelligibility. In large auditoria, such as theaters or assembly and lecture halls, it is important to place the seating area within an approximately 120 degree arc in front of the speaker. The effective speech intelligibility will increase when the listeners can see the mouth movements of the talker.
- The reverberation time should be in the range of 0.6 to 1.0 seconds, increasing with the size of the room, see Figure 6.9. The reverberation time should also be constant to within 10% over the 125 Hz to 8 kHz octave bands, for small- and medium-sized rooms (see Figure 5.17). It is advantageous if added sound-absorbent material is placed so that it does not interfere with the propagation of important early sound reflections. This means that one should leave the central part of the ceiling nonabsorptive in auditoria, classrooms, and other verbal communication rooms.
- For small studios and control rooms, such as those used at radio stations (see Figure 6.7), having volumes smaller than 25 m^3, it is common to reduce the reverberation time to the



Figure 6.7 Control rooms usually need to have short reverberation times. It is often necessary to remove undesirable reflections by sound absorption or sound scattering. School of Music, Gothenburg, Sweden. (Photo by Mendel Kleiner.)

range 0.2 to 0.4 s. It is also usually necessary to control the resonant modes at low frequencies using various types of porous or resonant sound absorbers.

Rooms for Music

Over the ages, the acoustic properties of the performance space have influenced the way music is written and performed. Composers of music have adapted their works to the reverberation characteristics of the spaces that were familiar to them as likely places for the performance of the particular music. Classical secular music from the baroque period was generally intended for performance in small spaces, having short reflection paths and highly reflective surfaces. Much of the classical music written during the nineteenth century, however, was written for the much larger rooms built in that era that were associated with considerably more reverberant characteristics and nearly twice as long reverberation times (see Reference 6.3).

Today it is common for musical performance spaces to be built acoustically flexible, so that many types of music can be performed successfully—one such flexible venue is shown in Figure 6.8.

Much of the popular music of today does not require a dedicated acoustically reflective environment. Using loudspeakers on and off stage, one can simulate or create the desired acoustical characteristics for the space. *Sound reinforcement systems* and *electronic reverberation enhancements systems*—particularly those outdoors—can be used effectively to simulate many types of spaces.

Some general recommendations for rooms intended for the performance and enjoyment of classical music are (see Reference 6.4):

- The *background noise levels* must be low.
- The *sound level* must be optimized—it should be neither too loud nor too low. It is defined by the *room gain*.



Figure 6.8 In this computer-controlled auditorium at IRCAM, Paris, the room acoustic conditions could be changed drastically using rotating wall sections. The sections were designed to be used as reflectors, scatterers, or absorbers (photo by Leif Rydén).

- There must be suitable *clarity*, and the *spatial* and *temporal distribution of the early reflections* must be good.
- The *reverberation time* must be appropriate for the music expected to be played.

The sound level in a real room will generally not behave exactly in the way described by the simple room acoustic theory presented in Chapter 4. The reason for this is the nondiffuseness of the sound field, sound absorption is not evenly distributed over the room surfaces. In the quasidiffuse sound field in a real room, the level will stay reasonably constant but will drop off with increasing distance. This is particularly noticeable in the audience area of, for instance, a concert hall. Far back on the main floor the sound level will also drop off due to negative interference from the sound reflected off of the seating in front, as discussed in Chapter 1.

The optimal *reverberation time* depends on the size of the room and on the type of performance that is expected to dominate the use of the room (see Figure 6.9). It is common to provide for several auditoria within one venue, which allows for the design of rooms that can complement one another regarding acoustical characteristics.

If it turns out to be necessary to build only one main auditorium, the auditorium can possibly be designed to provide for *variable acoustics* by passive or by active means. *Passive acoustic variation* can be achieved using variable absorption or volume. *Active acoustic variation* can be achieved by electronic sound field simulation or reverberation enhancement using microphones, digital signal processing, amplifiers, and loudspeakers (see Figure 6.10).

The reverberation process can often be considered as subdivided into several parts from the viewpoint of hearing, at least regarding large rooms having reverberation times between 1 and 3 seconds. Usually one considers *direct sound, early,* and *late* reverberation.

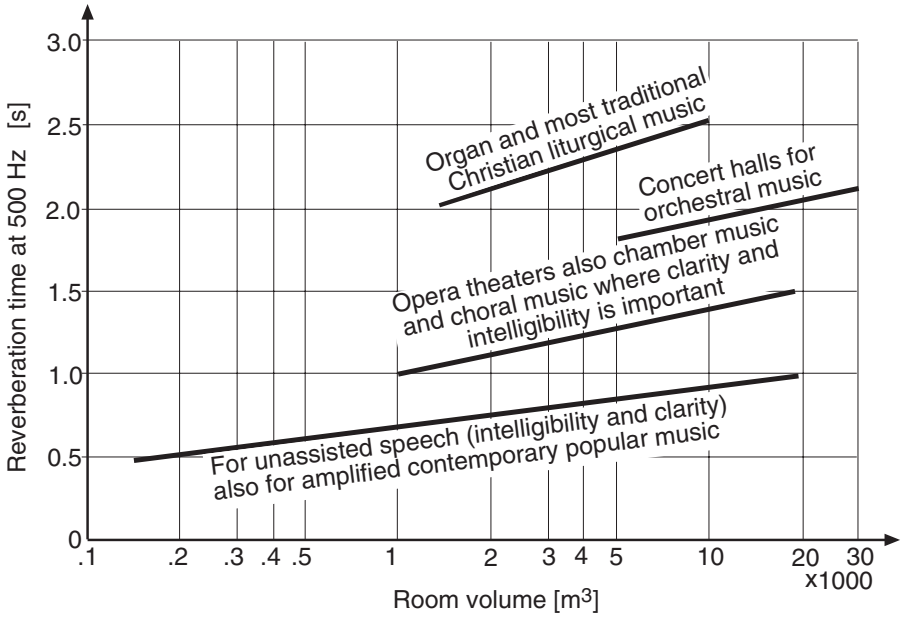


Figure 6.9 Typical recommended 0.5 kHz reverberation times for some common types of auditoria.



Figure 6.10 Using an electronic reverberation enhancement system one can achieve high flexibility in room acoustic conditions. The photo shows an active system for control of acoustics on a stage using loudspeakers integrated into a stage shell to achieve a proper blend of early reflections and local reverberation. Installation in Motala Community Center, Sweden. (Photo by Mendel Kleiner.)

The direct sound is the sound that arrives to the listener by way of the shortest path. The path may, at times, be interrupted by corners and balcony fronts, for example. It is also common to regard some reflections that follow immediately after the direct sound (having a time delay relative to the direct sound of less than approximately 5 ms) as a part of the direct sound. This is particularly the case if they are in the same vertical plane of the listener, such as from the audience area or the stage. From the listener's perceptual viewpoint, these early reflections are *fused* with the direct sound.

The early part of the reverberation process that follows the direct sound in the time range of 5 to 80 ms (5 to 50 ms for speech) is often called *early reflections*. Figure 6.11 shows reflectors suspended over the stage at the Concert Hall, Singletary Center for the Arts, University of Kentucky to enhance the early reflections. The hall was designed using results from sound field simulation, "Auditorium Synthesis", and by using physical scale modeling (see Reference 6.17). The acoustics of the hall combines high clarity with long reverberation time.

The *late reverberation* is the reverberation that follows the early reflections—that is, more than 80 ms after the arrival of the direct sound. The *very late reverberation* that follows more than about 160 ms after

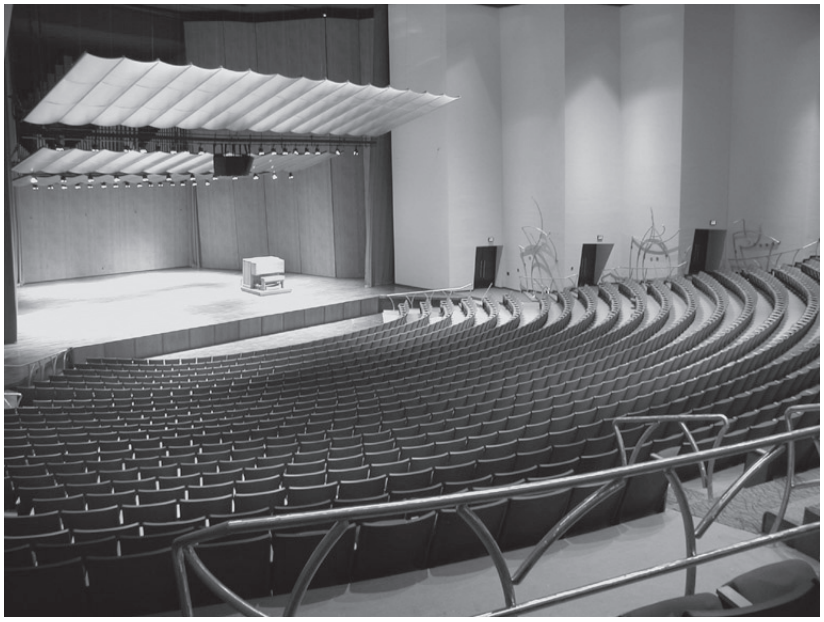


Figure 6.11 This photo shows the Concert Hall, Singletary Center for the Arts, University of Kentucky, Lexington, KY. Using sound reflectors suspended over the stage, one can add early reflections at the listener. In wide auditoria, overhead sound reflectors are usually necessary to provide both good stage acoustics and sufficient early reflected sound for the audience. It is important that the reflectors are large, have suitable scattering properties, and have enough mass so that they will provide strong and even sound coverage. The shape, number, placement, and adjustment of the reflectors are all factors that will influence the results. "Continental seating" is used in this hall. Note the sound reinforcement system's large central cluster loudspeaker array that is used to amplify speech (photo by Dwight Newton, courtesy of University of Kentucky College of Fine Arts).

the direct sound is usually heard only at the end of a musical piece. It could be argued that this very late reverberation is extremely important since it is the last part of any piece of music heard in a concert hall.

When measuring the reverberation time we usually obtain a decay curve that shows both the early and late parts of the reverberation process. A physical reverberation process, at least when one looks at a wide frequency range response, is not characterized by the exponential decay of the theoretical process discussed in Chapter 4. The real wide bandwidth decay in the frequency range above the Schroeder frequency has many irregularities caused by the approximately discrete reflection sequence of sound in the room. The process will usually approximate that of one or more exponential processes. The part of the reverberation process that is from the start of the process out to a time limit of 0.16 s after the start is particularly important for the subjective determination of reverberation time with running music.

Note that the reverberation process will depend on the location of the sound source and the microphone position in any given room. The ISO 3382 standard gives advice on good practice for the measurement of reverberation time of performance venues.

Some target values for the reverberation time T for some different types of rooms can be found in Figure 6.9. Large rooms for classical music will typically require a volume of around 12 m^3 per square meter of audience area to obtain a sufficiently long reverberation time at 1 kHz where the sound absorption of the audience and the chairs is the highest. One often strives for a frequency-independent reverberation time within ± 5% in the octave bands from 250 Hz to 4 kHz.

In the case of large rooms, such as concert halls (volumes more than approximately 5,000 m^3), the reverberation time will drop off in the 4 kHz octave band and above due to the sound absorption of air, as discussed in Chapter 1. It is common practice to leave surfaces for final adjustment of the reverberation characteristics of a room after completion.

Generally, the sound absorption by common building constructions and surfaces is fairly low in the octave bands at 125 and below. It is common to allow an increase of reverberation time down to the 63 Hz octave band of approximately 30% over the value at 0.5 kHz. This practice results in good bass response and a warm sounding reverberation. Sometimes surfaces, such as the stage floor, may act as secondary sound radiators at low frequencies for cellos and double basses that further enhance the *warmth*.

A room having a short reverberation time is usually considered *dry* and acoustically uninteresting. The desire for high clarity however makes reverberation times in the range 0.2 to 0.4 s desirable for control rooms, such as the one shown in Figure 6.7.

Another main factor influencing room acoustic quality is the spatial and temporal distribution of the early reflections.

The *time gaps* between the direct sound and major early reflections, are important and should be irregular and not arrive later than approximately 30 ms. The early reflections should be wide-band (not have a limited frequency range). One should avoid strong overhead reflections appearing in the same vertical plane as the direct sound. Since hearing cannot differentiate between signals having the same interaural delay time, such overhead reflections will cause comb filter effects unless masked by other reflections coming from the sides. A diffuse or curved ceiling will redirect these reflections to the walls.

The spatial distribution of the early reflected sound is important, particularly for creating the impression of *auditory source width*. One should strive for sound reflection angles close to the horizontal plane to maximize the advantages of our binaural hearing. It is advantageous if the directional distribution of the early reflections are such that the sound field appears to be *symmetrical* when listening. One should also strive for an even angular distribution of the reverberant sound to enhance the feeling of *envelopment* and diffuseness in the sound field.

It is important to realize that subjective diffusitivity and physical diffusitivity are not well correlated. One can have good subjective sound field diffusitivity without the sound field being diffuse in a physical sense. Good physical diffusitivity does not guarantee good listening conditions in a room.

The side walls should be somewhat scattering; for example, by using shelves, balconies, unevenly placed sound-absorbing patches or objects having sizes in the range of 0.5 to 0.05 m. Figures 6.12–6.14 show various ways of achieving high scattering by wall and ceiling surfaces. *Flutter echo* is the term given to repetitive short path sound reflections, usually appearing when reflections between two close and parallel walls dominate the sound field. It can be removed by splaying the walls, absorption or scattering.

One phenomenon of particular importance in long concert halls is that of cancellation on the main floor of direct sound by the first reflection of sound off the audience in the frequency range between 125 Hz and 1 kHz. This reflection will be at a small angle leading to phase reversal of the sound in the frequency range mentioned (that is, a reflection coefficient of $R = -1$). Since the sounds cannot be separated because of the short time difference between the direct and reflected sounds (typically less than 5 ms), there will be cancellation. The cancellation will result in *coloration*—timbral change in the sound quality of the *direct* sound.

According to the *law of the first wave front*, the direct sound determines our perception of the timbre of the sound source. Consequently, it is important to design the seating area in such a way that the coloration is minimized. This requires that:

- The chairs are designed appropriately.
- The seating area has a sufficient slope.
- The stage is sufficiently high.
- There are good sight lines to all seats.

A room having good properties regarding background noise, early reflections, reverberation time, and scattered sound will be close to ideal from the audience's acoustical point of view.

In addition, it is important that the acoustical conditions on stage give musicians good possibilities for hearing one another and give support for their own playing. The stage and its surrounding walls and ceiling must be designed so that the sound is redirected appropriately to all parts of the orchestra as well as to the audience. The stage and the auditorium must, in addition, be free of severe acoustical defects, such as echo and nonlinear distortion (for instance, a rattle).

Recording Studios

The acoustical requirements for music recording rooms are similar to those that apply to rooms for listening. Because of the different ways in which a listener interprets the sound by direct listening and by listening to sound picked up by microphones, there will be both difficulties (absence of binaural signal



Figure 6.12 One can scatter the reflections from surfaces using diffusors to soften the early reflections. This photo shows testing of diffusors to create good ensemble characteristics for a concert hall stage in Baltimore's Joseph Meyerhoff Symphony Hall. (Photo by Mendel Kleiner.)



Figure 6.13 Formerly, high sound field diffusitivity was achieved by uneven surfaces, niches, and rich ornamentation, as in the Wiener Musikverein Concert Hall, Vienna shown here. (Photo by Mendel Kleiner.)



Figure 6.14 Currently, the surface unevenness is achieved by simple means and is primarily introduced to provide desired diffuse sound reflection as in the San Francisco Symphony, San Francisco, CA. (Photo by Mendel Kleiner.)

processing) and advantages (close-up recording, directional microphones, and various forms of signal enhancement). (See Reference 6.5.)

By using various microphone techniques, one can adapt the recording to the room acoustical conditions. Since the visual impression is not as important, more effort can be put into optimization of the room acoustics.

It is common for studios to have various types of sound reflecting, scattering, and absorbing panels, both freestanding and wall mounted. A recording studio venue will also usually contain many small rooms, or subspaces, for particular instruments. These will have splayed surfaces to avoid flutter echo.

Any recording studio must be quiet, since the listener will likely have to listen to the same recording several times. The background noise sound level in a studio for recording of acoustical instruments should preferably be below 15 dBA, as mentioned previously. It is important to observe that the noises from traffic and HVAC have noise spectra that are different from that of typical microphones.

Television studios generally pose acoustical difficulties, from the viewpoint of both noise and room acoustics. Close-up microphone placement (a wearable broadcasting microphone, for example) is often necessary to allow visually unobtrusive sound pickup (see Reference 6.6).

Rooms for Sound Reproduction

Rooms for sound reproduction are typically much smaller than those for concerts or classical music recording—typically having volumes in the range of 25 to 75 m^3. This leads to problems at both low and high frequencies.

In sound reproduction the acoustical characteristics of the transfer paths between loudspeaker and listener are overlaid on the recorded signals.

It is necessary to differentiate between listening rooms in studios (control rooms of various kinds) and rooms in dwellings (living rooms, for instance). The studio environment is usually subject to a more critical evaluation and listening than the dwelling environment. Also, there is likely to be more interest and money available for creating a good listening environment in a studio than in a home. The freedom to design the room with acoustics in mind will also be larger since fewer compromises have to be made because of other uses.

Both environments require fairly short reverberation times. As a rough approximation, one should strive for a listening room reverberation time no longer than approximately half that of the recording venue if the recording has been made so that the reverberation is clearly audible and dominant (for example, using omnidirectional microphones far beyond the room's reverberation radius). Typically, a reverberation time in the interval between 0.3–0.7 s may be desired over the entire audio frequency range.

An important issue in playback rooms is to avoid early reflected sound incident in the median plane, since such sound components will cause comb filter effects as discussed previously. The components can be eliminated by using absorptive or scattering treatment (or simple redirection) of the surfaces responsible for the undesired reflections. Sometimes it is also useful to diffuse the wall patches that are responsible for the first side wall reflections. A specification of listening room characteristics is made in the IEC 268–13 standard that strives to formulate the acoustical characteristics of a typical listening room. A room that fulfills the standard is shown in Figure 6.16.



Figure 6.15 Wenger's small commercial portable cabin for music rehearsal featuring electronically variable reverberation. (Photo by Mendel Kleiner.)

In one technique often found in control rooms one half of the room (where the loudspeakers are) is treated to be highly sound absorptive, and the other half (where one listens) is "live" and treated to be diffusely reflecting. This classic approach allows a distinct and clear sound around the listener in combination with a feeling of space. More modern approaches are creating a "reflection free" zone around the listener or making walls and the ceiling diffusely reflecting.

Small rooms usually have problems in accurate music sound reproduction because of the low modal density and poor damping of modes at low frequencies as mentioned previously. This makes it virtually impossible to reproduce music accurately at low frequencies unless these low frequency modes are well damped. Using various types of signal correction by means of digital signal processing, it is possible to circumvent this problem to some extent. The use of various types of low-frequency sound absorption is in any case a must. Slit and Helmholtz resonators, as well as membrane-type sound absorbers, are suitable for this task. Figure 6.17 shows a combination of a diffuser and low-frequency sound absorber.

Correct positioning of the loudspeakers, relative to the room modes and the listening position, will ensure the least variation in frequency response. It is practical to use the reciprocity technique for finding the optimal positions.



Figure 6.16 The IEC listening room at the Section of Acoustics at the Aalborg University is a good example of an advanced IEC listening room (photo courtesy of AAU, Denmark).



Figure 6.17 An example of a combination of slit absorber and diffuser for broadband control of the acoustical conditions in a recording studio in the Gothenburg Opera, Sweden. (Photo by Mendel Kleiner.)

6.4 TOOLS FOR PREDICTION OF ROOM ACOUSTIC RESPONSE

The acoustic properties of a room can be predicted in many different ways using *computer aided design* (CAD) with varying results.

For *low frequencies*, calculations using software based on the *finite element method, boundary element method*, or *finite-difference time-domain method* are, in principle, quite exact, but comparatively slow. For the modeling, one needs the impedance data for the boundaries—without which the results may be quite erroneous, particularly regarding the location of resonance frequencies, mode shapes, and damping. The necessary impedance data however is seldom available. The generation of a suitable mesh is also a problem since few current mesh generators are suited to quickly and effectively meshing the geometries common in room acoustics tasks (see References 6.11 to 6.17).

For problems involving *medium and high frequencies*, geometrical acoustics is usually used. Initial design considerations usually include manual ray tracing and reverberation time calculation using Sabine's or *Fitzroy's* formula (see Reference 4.1). Sabine's formula essentially requires the sound field in the room to be diffuse, which is seldom the case. Fitzroy's formula is particularly useful in predicting the reverberation time for those rectangular rooms where sound absorption is unevenly distributed between the walls. For building acoustics and some uses in architectural acoustics software based on statistical energy analysis

(SEA), modeling is also used (see Chapter 10). This approach is based on the study of energy flow between resonant systems.

Ray tracing and mirror image method software is now common in most acoustics consultants' offices. Using CAD, it is possible to predict not only reverberation time, clarity, lateral fraction, and other room acoustics metrics but also to predict the impulse response. Many other modeling methods are, however, also possible (see Reference 6.8).

The following methods are (or have been) used in room acoustics modeling:

- Optical ray tracing using laser
- Optical intensity modeling using lamps
- Ultrasonic scale modeling by the impulse method
- Ultrasonic scale modeling by the intensity method
- Ultrasonic scale modeling by the transmission method
- Physical acoustics modeling using Finite Element (FEM) or Finite Difference Time Domain (FDTD) methods software for low frequency, small room modeling not discussed in this book
- Geometrical acoustics modeling using ray tracing or mirror image modeling software
- SEA modeling using specialized software

The optical methods must be considered obsolete except for some special uses. Optical ray tracing is based on the use of physical scale models (typical scale values are 1:10 to 1:100). The sound source is simulated using a laser or a small light source surrounded by a grill so that light is emitted in sectors. By using a scale model where reflecting surfaces are modeled by mirrors, it is possible to study the distribution of early reflected sound—sound that has been reflected only once or twice against surfaces. The advantage of the method is its simplicity and low cost. It can often be used directly in the architect's scale model. A major disadvantage of the method is that it does not take sound diffraction by edges or scattering by surface irregularities into account.

Optical intensity modeling is based on the same type of physical scale models as those used in optical ray tracing, but in this case, the static light distribution is studied as an approximation to the diffuse field sound intensity.

Ultrasonic scale modeling typically uses models in scales from 1:4 to 1:16, although some researchers have shown useful results even with scales as small as 1:50. (Outdoor sound propagation is typically modeled ultrasonically using a scale of 1:100.) Using spark source excitation, one can study the impulse response of the room. Such a scale model is shown in Figure 6.18.

The static sound distribution can be calculated from the impulse response measurements or directly studied using, for example, jet noise sources.

The following rules apply to ultrasonic scale modeling at 1:m scale:

- The frequency scale is transposed by a factor of m:1.
- The time scale is transposed by a factor of 1:m.
- The sound attenuation by air is reduced by a factor of m:1.
- The acoustic impedances of the materials used in the scale model must be similar to those in the real room (after appropriate frequency scaling).

One of the major problems in ultrasonic scale modeling is the correct scaling of the attenuation of sound by air. Another major problem can be found in the difficulties in designing sound sources and



Figure 6.18 Model of a church built for ultrasonic scale modeling. Note scale model of manikin for binaural sound pickup next to the hand. (Photo by Mendel Kleiner.)

microphones that can cover the frequency range of interest (typically 200 Hz to 200 kHz) with adequate directivity and sensitivity. One can approximate the correct scaling of sound attenuation by air, using dry air (typically having a relative humidity of less than 2%) or using a pure gas, such as nitrogen in the scale model. These methods have been shown to give good results not only regarding the simulation of reverberation time and levels but also regarding impulse response. Research has shown that it is possible to digitally compensate impulse response measurements made using scale models without these precautions, and still obtain quite good simulations.

A large advantage of physical ultrasonic scale modeling is that it is easy to simulate the acoustic conditions in irregularly shaped rooms, and in rooms with uneven sound absorption. In addition, ultrasonic scale modeling allows for inclusion of diffraction and scattering as well as the frequency-dependent sound absorption by various surfaces.

Using computer software, it is possible to predict the acoustic properties of a room in much more detail than by manual methods. Once the geometrical and acoustical data for the room have been entered into the computational model, one can calculate the way sound from one or several sources is incident at various listening positions. Usually one is interested in the spatial impulse response expressed as $p(t)$ relative to the sound pressure at 1 m distance from an omnidirectional source. If one is interested in auralization of the room, it is necessary to know the angles of incidence (φ, δ) for the various sound components contributing to the room impulse response. Using the room impulse response, it is possible to calculate the various room acoustic quality indices, as previously mentioned.

Most software for the prediction of room acoustics is based on sound propagation modeling using geometrical acoustics and, therefore, can only take diffraction, scattering, and real acoustic surface impedance into account in an approximate way. Some software will approximate the influence of these acoustic

phenomena on the propagation of sound and thus yield improved modeling. Most such software is based on ray tracing or mirror image modeling or possibly a hybrid between the two.

Modeling using the *ray tracing method* typically assumes a large number of rays (1000 to 1,000,000) being sent out from a sound source. The sound source can be assumed to be omnidirectional or directional. In the latter case, one can either assign different intensities to the rays or adjust the ray density to simulate the source directivity. One then follows each ray on its path and studies how the ray is reflected by the surfaces of the room and the objects it contains. Curved surfaces are generally simulated by piecewise flat surfaces. By studying the number of rays passing through a test surface or test volume (such as a sphere) at the listening position of interest, one can calculate the room impulse response at that position.

In most ray tracing models, the sound absorption is modeled by adjusting the ray intensity appropriately after calculation of the room impulse response. Scattering can be simulated by assigning various randomization rules to the angle of reflection. *Pyramid tracing* and *cone tracing* are variations on simple ray tracing.

Modeling of sound propagation using the *mirror image method* assumes that sound reflected by a flat, hard surface originates from a mirror source behind the surface. One needs to calculate the various mirror images created by the (flat) surfaces of the room and its objects. For a room having a complex

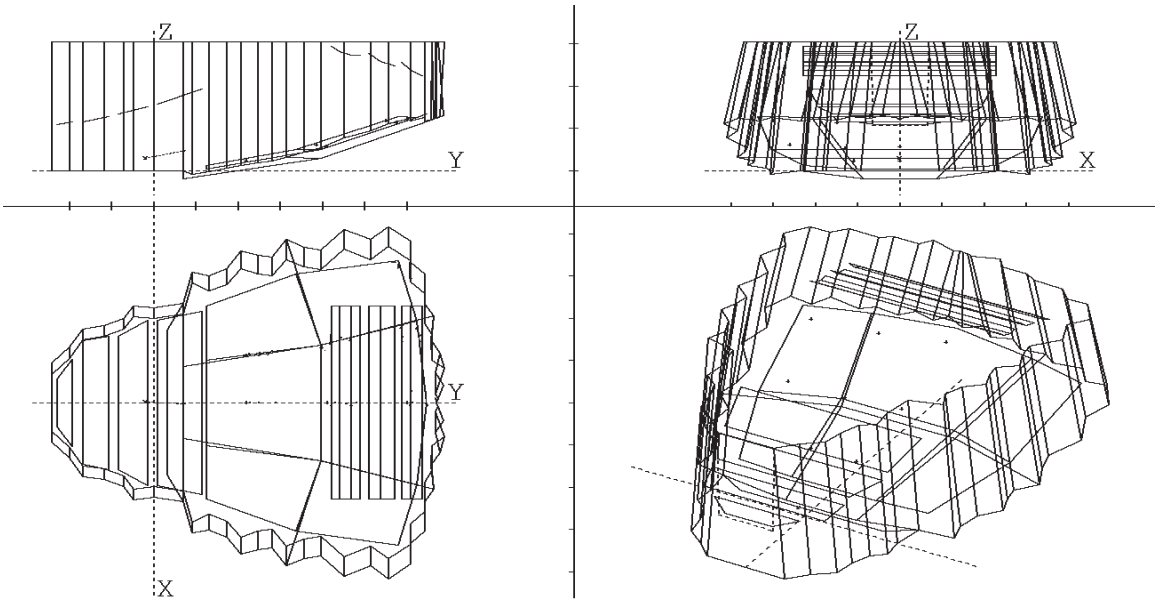


Figure 6.19 CAD using ray tracing or mirror image method-based software can be used to predict the acoustics (see also Figure 6.20). This model of the room shown in Figure 6.11 was done by Julie Byrne, MSc.

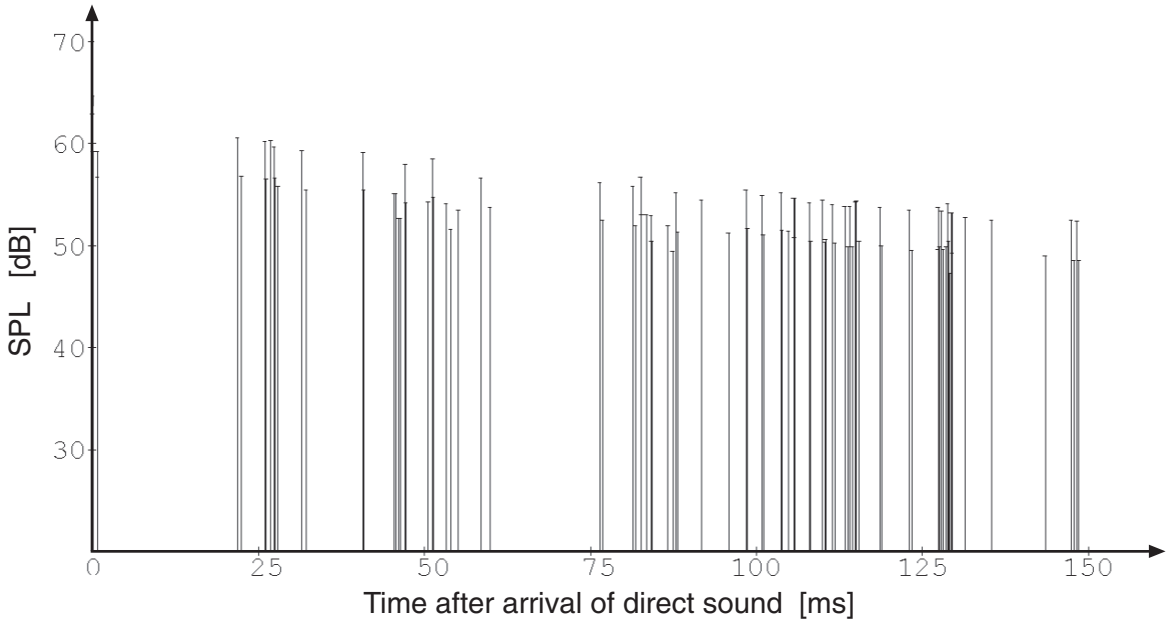


Figure 6.20 A calculated echogram for mid-stage to last row on center in the room model shown in Figure 6.19 (arbitrary reference level).

geometry, it may be necessary to calculate many millions of mirror sources, as sound is reflected multiple times by the various surfaces. The relevant sources must then be sorted out using rules for visibility. A disadvantage of the method is that it does not allow for easy inclusion of scattering and diffraction. Figure 6.20 shows the calculated echogram for the room shown in Figure 6.19 (set for not taking frequency-dependent sound absorption, scattering, or diffraction into account). One can see that the response consists of *discrete* reflections.

Hybrid models are typically based on the use of both ray tracing and mirror image modeling. Ray tracing is used initially to find the possible sound paths and then mirror image modeling is applied to find the exact sound paths. As with ray tracing, a disadvantage is the need for modeling the propagation of a large number of rays.

Both ray tracing and mirror image modeling are similarly computationally intensive. Neither method allows for exact room acoustic modeling since geometrical acoustics is used. Good modeling of absorption, diffraction, and scattering remains a problem. In any case the prediction will be limited by the experience and skill of the user. The late part of the room impulse response corresponding to the late part of the reverberation is usually too computationally intensive to be handled by a direct approach, and various approximations are generally used. Some of these may be based on the use of various statistical assumptions and random generated numbers. Such methods may lead to unevenness in the calculated full room impulse responses.

6.5 ELECTRONIC ARCHITECTURE

It is advantageous to be able to change the room acoustic properties of performance spaces to suit the type of performance, for example, various types of orchestras, music, and voice. This can be done using two approaches, *reverberation enhancement* and *sound field synthesis,* sometimes called *electronic architecture* (see References 6.4, 6.9, and 6.10).

Reverberation enhancement is typically used when the room acoustic conditions are reasonably good, but where reverberation time or level may need to be increased. It is often possible to increase the reverberation time by up to 50% with good results.

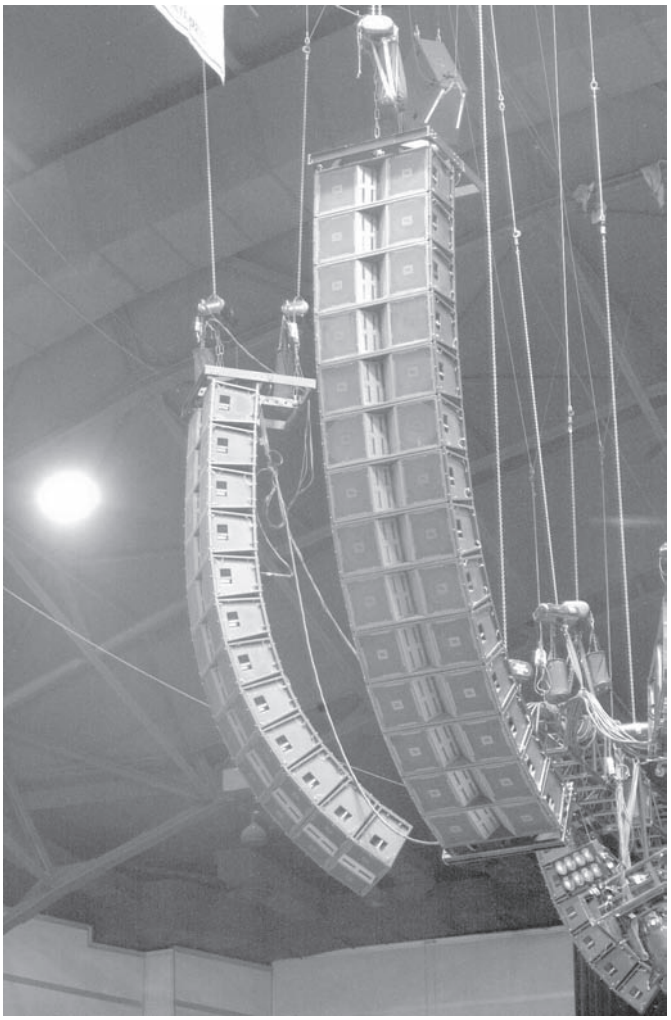


Figure 6.21 Sound reinforcement systems may use conventional direct-radiating or horn-coupled loudspeaker array elements to make up an array-type loudspeaker column to achieve the desired directivity (image courtesy JBL/Harman).

Sound field synthesis may include reverberation enhancement, but will usually include also the generation of specific early sound field components corresponding to early reflections by (imaginary) surfaces. Sound field synthesis is sometimes used to reduce the subjectively perceived reverberation time T_{subj} by reducing the relative level of the late reverberant sound. The conventional *public address system* is an example of primitive use of *sound reinforcement* to provide this effect, for example, in churches and other public meeting places (see Figure 6.21).

6.6 AURALIZATION

Audible sound field simulation is usually called *auralization*. Auralization of an environment (e.g., for speech and music) can be considered as analogous to visualization of that environment. The ambition is to make it possible to listen to the sounds in the environment without being in the actual environment (see Reference 6.8).

Computer auralization of an environment will start by entering data for the geometry of the room and the objects in the room into a computer database to be used by a mathematical model based on ray tracing or mirror image modeling. Once the software has calculated the response, it will generate a file containing a list of information about which sound components are incident at the desired listening positions:

- Levels relative to the direct sound for reflections
- Time delays relative to the direct sound for reflections
- Propagation path lengths

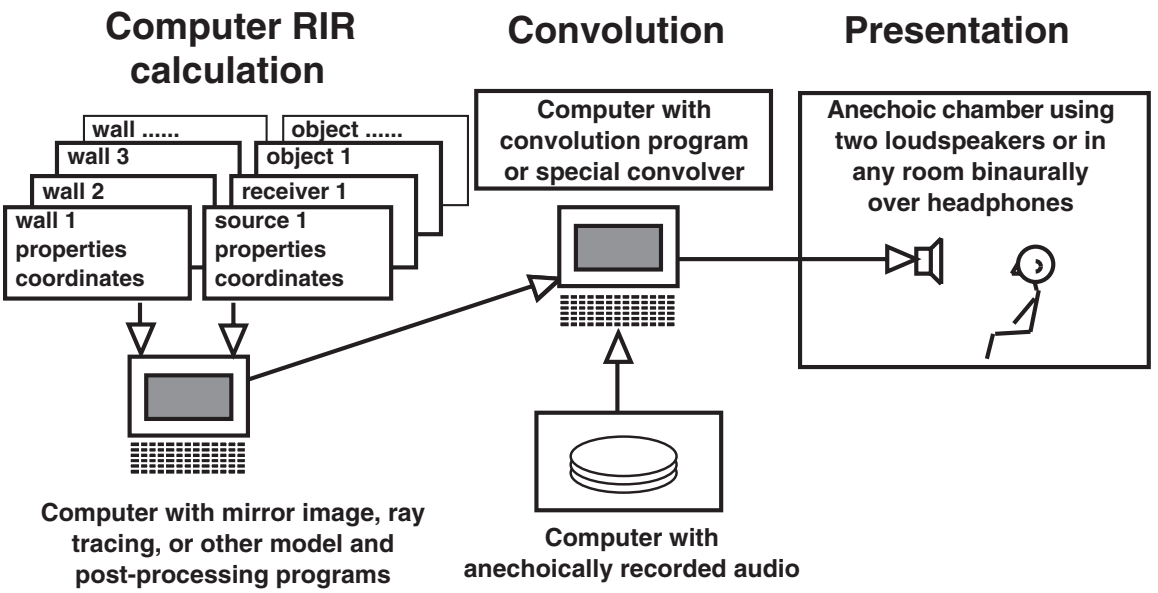


Figure 6.22 Functional diagram for fully computed auralization (After Ref. 6.8).

- Angles of radiation
- Angles of incidence

Using knowledge about the sound reflecting and scattering properties of the various surfaces, one can then add this information to the initially generated *generic* room impulse response (see Figure 6.22).

There are two main presentation methods used for auralization—binaural playback and playback by surround sound systems, such as *Ambisonics*. Binaural playback can be done either directly using headphones or by a stereo sound system using *crosstalk cancellation* (XTC). Surround sound systems need at least four loudspeakers—seven to ten loudspeakers are typically used.

In practice, XTC requires a quiet room in which the reverberation time is short, such as a semi-anechoic room. The aim is to provide the listener or listeners with sound signals at the ears corresponding to those that would be obtained when listening directly using headphones. This requires that the leakage of sound between the ears due to the loudspeaker presentation is cancelled using cancellation signals generated using knowledge about the transfer paths between the loudspeakers and the ears. Presentation using crosstalk cancellation generally gives better frontal localization than ordinary headphone presentation.

For auralization to be presented binaurally, information about the way the head influences the incoming sound waves must be added. Because of the way sound is reflected off the human body, the sound pressures at the ears are going to depend on the angles of incidence and on the frequency. At low frequencies, below 100 Hz, the effects will be negligible but already at approximately 250 Hz, the torso will contribute to the sound pressure. At frequencies above 5 kHz, there will be major effects as shown in Chapter 3 because of the reflection and diffraction of sound by the head. The frequency responses of the sound pressure at the left and right ears relative to the pressure in absence of the body are called the *head-related transfer functions* (HRTFs). Once the head-related impulse responses (HRIR) have been obtained, they can be converted into time domain signals as *head-related impulse responses*. Free libraries of HRTFs and HRIRs may be found on the Internet for various angular resolutions.



Figure 6.23 The Bose Auditioner® system is a commercial system for auralization using CAD. The binaural room impulse responses are convolved with music or speech so that it is possible to listen to a simulation of what it would be like to listen in the actual room being simulated. Crosstalk cancellation is used to render the binaural signals in this setup (photo courtesy of Bose Corporation).

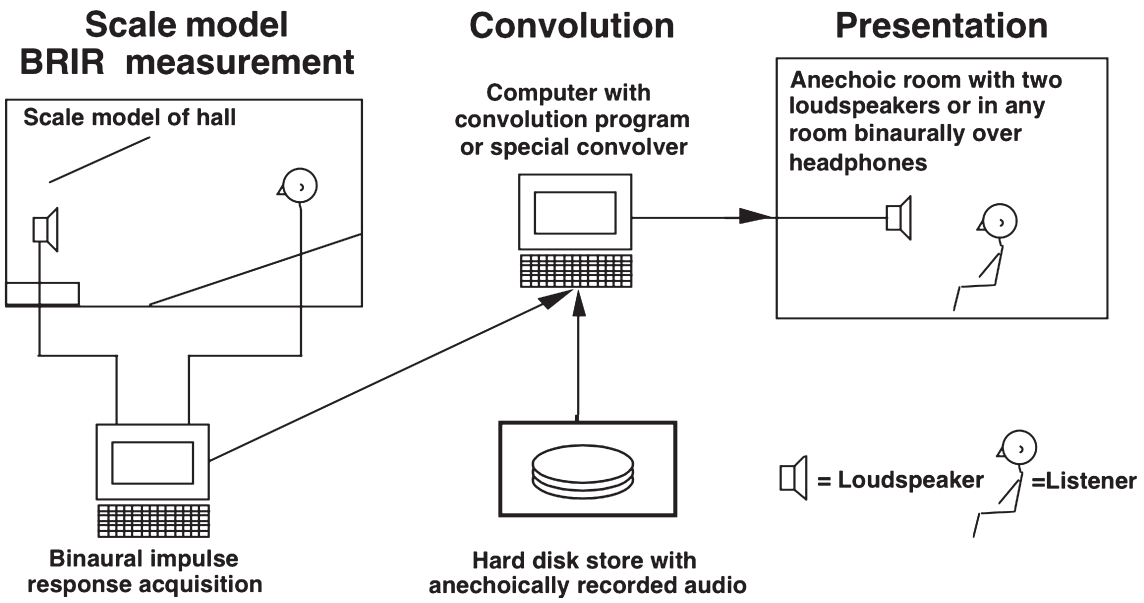


Figure 6.24 Functional diagram for auralization using the indirect ultrasonic scale modeling method (After Ref. 6.8).

Once the *generic spatial impulse responses* for the positions of the source and receiver in the room have been found, these can be appropriately *convolved* with the HRIRs and with the impulse responses of the sound-reflecting surfaces—the impulse responses of the various reflection coefficients described in Chapter 2. This yields the final binaural room impulse responses.

The binaural room impulse responses can then be convolved with *anechoically recorded* speech and music so that one can listen to what the sound of the room would be. The convolution can be done using software in a general purpose computer or by a dedicated *hardware convolver* (digital filter).

The convolved binaural signals can then be played back over headphones or further processed by a crosstalk cancellation network to allow for loudspeaker presentation. (See Figure 6.23.)

Auralization can also be done using a scale modeling approach. Direct modeling where the audio signals are played back as well as recorded in the model is quite problematic and an indirect approach where the binaural room impulse response is measured digitally and is subject to suitable digital signal processing is favored. The principle of this approach is shown in Figure 6.24. The main reasons for this is the possibility for a better S/N ratio and lower nonlinear distortion than using the direct method. In the indirect approach, the main difficulties lie in obtaining a suitably scaled manikin with torso, head, and ears, all correctly modeled to scale. Once the binaural impulse responses have been obtained, these are then convolved with the desired anechoic audio signal as in the case of fully computerized auralization described initially.

The advantages of auralization are many. By listening to auralization of rooms, costly mistakes in planning and building can be avoided, and different wall treatments and seating arrangements can be investigated. Auralization can also be used to study the properties of sound reinforcement systems, to reduce

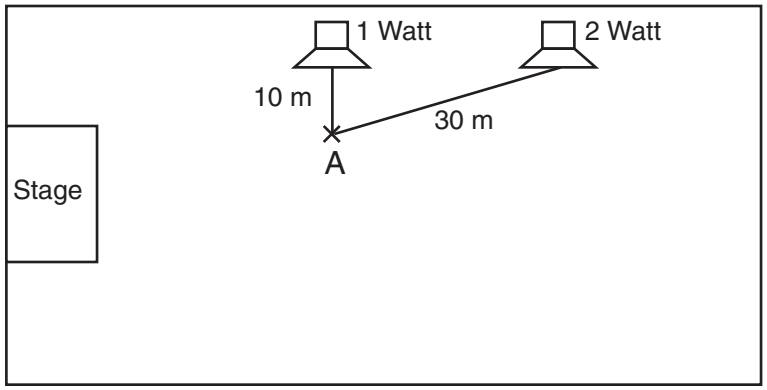
tuning time at building completion. Along with its use in architecture, auralization is used in various forms of virtual reality rendering, games, information systems, among others.

6.7 PROBLEMS

6.1 Two loudspeakers in an auditorium are connected to the same signal source but are radiating different sound power.

Task: Determine, using the figure on page 162 and Figure 5.11, if the sound at listening position A will be considered as good or poor, since one of the sounds will arrive delayed relative to the other.

Hint: Only consider the direct sound components from the loudspeakers and assume that the loudspeakers are omnidirectional.

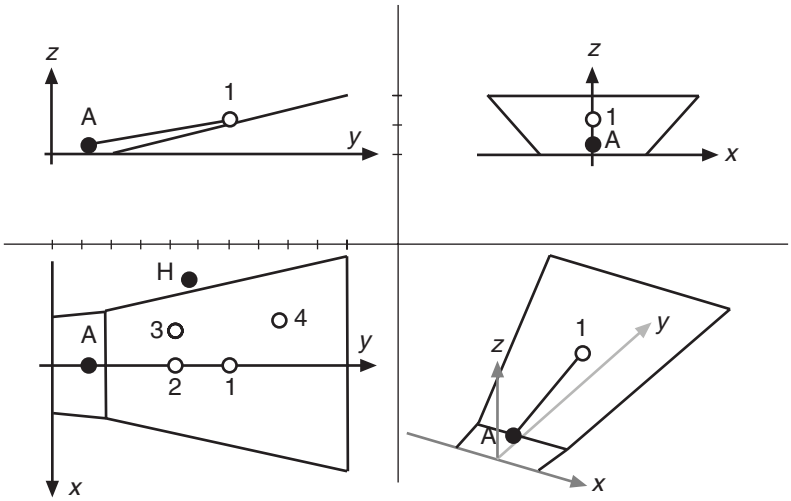


6.2 One wants to use a loudspeaker to improve the acoustical conditions in an outdoor theatre. Assume that the sound source is omnidirectional and all surfaces are sound absorptive. Use the figure to determine the required loudspeaker playback levels. The distances from loudspeaker H and actor A to the respective listening positions are:

	Pos. 1	Pos. 2	Pos. 3	Pos. 4	
Distance to actor	24	14	16	33	[m]
Distance to loudspeaker	15	14	9	16	[m]

The distance between the markers in the figure is 5 m.

Task: Calculate the necessary delay times for the four positions in the figure to avoid echoes.



6.3 One wants to use a loudspeaker to improve the listening conditions in a concert hall and mounts a loudspeaker above the stage. The direct sound level of the loudspeaker is adjusted to be 6 dB above that of the speaker in the direction of interest. The volume of the hall is 40,000 m^3 and the reverberation time T is 1.7 s. Assume the sound field in the reverberant field to be diffuse and that sound field components can be added on an intensity basis. Further, assume that the delay between the direct sound components from the speaker and loudspeaker is less than 0.010 s and that the directivity index of the loudspeaker in the direction of interest is 6 dB.

The directivity D is defined as the sound level at a certain distance from a directional sound source relative to the sound level that would be obtained from an omnidirectional sound source radiating the same power. The directivity index (DI) is defined as $DI = 10 \log(D)$.

Task: Calculate the level difference between direct sound and reverberant sound $H = L_{p,\,\text{direct}} - L_{p,\,\text{reverb}}$. Find the approximate subjective reverberation time T_{subj} for various seats at distances of 10, 20, and 30 m respectively from the speaker or loudspeaker with and without the loudspeaker active. Use Figures 5.10, 5.11 and 5.21 in making your decision.

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Absorbers, Reflectors, and Diffusers 7

7.1 INTRODUCTION

Sound absorbers are used to reduce the sound levels in noisy rooms and to optimize the acoustic conditions in rooms for voice and music. Absorbers are also used to reduce sound transmission through ducts and silencers. The sound energy is converted into heat by the absorber.

From an architect's viewpoint, absorbers can be grouped into natural absorbers and added absorbers. *Natural absorbers* are the audience and the performers, as well as such sound-absorptive surfaces, such as drapes, upholstered chairs and other furniture, walls, ceilings, carpets, and floors. In most cases, the desired acoustic properties of the room cannot be achieved by the raw surfaces and objects in the room. Materials used for walls, ceilings, and floors seldom have the necessary acoustic properties. *Added absorbers* are used to control reverberation time and other acoustic properties of the room further. Obviously, the division is not clear—an additional absorber can become a "natural" absorber once it is integrated into the building.

A more clear subdivision can be made between resonant and nonresonant absorbers. *Nonresonant absorbers* are usually made of porous materials and often simply called *porous absorbers*. This group includes mineral or glass wool, textiles, and carpets. The *resonant absorbers* are made using resonant acoustic or vibratory systems, such as air columns, volumes, membranes, and plates.

Most materials have quite frequency-dependent sound absorption—porous materials are much more sound absorbing at high frequencies than at low frequencies. By combining porous materials and resonant systems one can provide frequency uniform sound-absorption.

7.2 ABSORPTION COEFFICIENT

The Characterization of Sound-Absorption Properties

Materials are characterized by their sound-absorption coefficient, α. The sound-absorption coefficient α is defined as:

$$\alpha = \frac{W_{\text{abs}}}{W_{\text{inc}}} \tag{7.1}$$

where W_{inc} is the power of the incident sound and W_{abs} is the power removed from the sound field by the absorber, including power that is transmitted through the absorber, away from the acoustical system under consideration.

In most cases, one can neglect the sound power transmitted through the absorber. There are, however, two special cases that need consideration—open windows and thin sound-transparent drapes, screens, and such. In the first case, one does not have absorption in the sense that sound power is escaping from the room, rather than being converted to heat. At high frequencies, the open window will have an absorption coefficient close to unity. In the second case, if there is an absorber behind the drape or screen, the sound is transported through it (with some losses) and absorbed. If the surface behind the screen or drape is hard, a standing wave field is set up due to the interference with the reflected wave. If the drape or screen is at the particle velocity maxima of this standing wave, the sound absorption may be very high, depending on the flow resistance of the material. Note that the sound-absorption characteristic will be frequency-dependent in this case unless the drapes are in deep folds.

In some cases, it is more relevant to express the absorption as absorption area rather than absorption coefficient. This applies to, for example, resonant absorbers and to absorbers where the geometrical surface can be hard to define. This is typical of audience and performers, as well as some other types of natural absorbers.

The symbol α is used to indicate the absorption coefficient, but it is sometimes necessary to subdivide the use of α into the following categories:

Theoretical values:

α_0 perpendicular incidence (plane wave, infinite surface)
$\alpha_0(\varphi)$ a particular angle φ (plane wave, infinite surface)
α_d diffuse incidence

Measured values:

α_r using the impedance tube method (semiplanar)
α_s using the reverberation chamber method (semidiffuse)

Measurement Using an Impedance Tube

The *impedance tube* (the Kundt's tube or plane wave tube) can be used in two ways, either to measure the standing wave ratio along the tube or to measure the transfer function between two points in the tube (not discussed here). The standing wave method is based on the knowledge of the sound pressure amplitude maxima and minima and their locations along a tube in which a wave is incident on a test sample and was discussed in Chapter 1. The applicable standard is ISO 10534–1. The reflection coefficient measured this way would correspond to that of an ideal plane wave. Because waves can propagate in the tube also in other modes than the plane wave (usually called the 0:th mode), it is necessary to make sure that only plane waves are measured. The cutoff frequency is defined as the frequency at which there are also nonplanar waves that can propagate in the tube. A commercial circular Kundt's tube usually has an internal diameter in the range of 0.02 to 0.1 m. A large tube has a low cutoff frequency, as discussed in Chapter 1.

There are two ways out of this dilemma. The simplest is to limit one's measurements to frequencies below which the modes do not occur. In a rectangular tube, the modes will be similar to those discussed regarding the sound field in the rectangular room. For a rectangular tube having the crosswise dimensions d_1 and d_2 and where $d_1 < d_2$, the measurements should only be done at frequencies lower than:

$$d_2 < 0.5\lambda \tag{7.2}$$

In a circular tube, there will be waves having circular and radial propagation components. For a circular tube having a rigid wall, the corresponding condition for the diametered can be shown to be:

$$d < 0.59\lambda \tag{7.3}$$

A second way out of the measurement dilemma is to use a microphone array over the cross section of the tube. If this array senses the in- and out-of-phase components of the sound field in the tube equally, the summed microphone array signal will be that of the plane wave. These arrays are usually made using probe tubes leading to one microphone and only applied to measurement tubes having a rectangular cross section. Note that close to the loudspeaker, there may be near-field pressure asymmetries; few loudspeakers have planar diaphragms.

From the standing wave ratio γ measured using the tube, the plane wave sound-absorption coefficient is calculated as (see Problem 1.4):

$$\alpha = \frac{4}{2 + \gamma + \frac{1}{\gamma}} \tag{7.4}$$

Measurement of *A* Using a Reverberation Chamber

The *reverberation chamber method* is based on the measurement of reverberation time in a chamber having hard walls, with and without the test sample. The conditions and problems that apply to the reverberation chamber method are described in detail in Chapter 4. It should be noted that the reverberation chamber can also be used to measure the scattering properties of objects.

Because the sound field properties of the reverberation chamber will influence the angular distribution of sound incidence on the absorption sample under measurement, different measurement results will be obtained with different chambers. Since high absorption coefficient values are preferred by manufacturers, it is natural to strive for room designs that give such values within the bounds of typical standards documents, such as ISO 354.

Using Absorption Area

In some cases, it is more relevant to characterize the sound absorption by an *absorption area* rather than by an absorption coefficient. This applies to resonant absorbers and to absorbers where the geometrical surface can be hard to define. This is typical of audience and performers, as well as some other types of natural absorbers.

For a large surface, the absorption area is the product of the surface area and the absorption coefficient. Note that the absorption area may be given in either *metric sabins*, or in *sabins* that are given in square ft. (For example, 10 m^2 of a material having $\alpha = 0.9$ has an absorption area of 9 metric sabins.) In this book, we only use metric sabins.

7.3 POROUS ABSORBERS

The conversion of acoustic to thermal energy in porous absorbers is due to the viscous behavior of air flowing in the canals, pores, and air pockets created by the skeleton of the material. Porous materials that do not have open cells or pores cannot function as porous absorbers—an example of such a material is expanded polystyrene. If the flow is blocked by, for example, paint or other surface coverings, the absorption coefficient will usually drop considerably. Because of the elasticity of the material, there may still be some absorption, however. It is important to differ between the sound absorption and the heat insulation properties of porous materials. Porous materials may be good thermal insulators and still have poor sound-absorption properties. Expanded polystyrene and many other porous plastic materials that have closed pores are poor sound absorbers.

One can model the surface of a porous material as having the structure shown in Figure 7.1. The material is regarded as a skeleton that forms narrow air passages. A practical model is to regard the passages as parallel and extremely narrow tubes. The tubes make it unnecessary to be concerned by propagation of sound sideways. Because of the propagation losses in the material, this is a reasonable assumption. Such a material is said to be *locally reacting*—that is, its acoustic behavior is similar to that described in Section 1.7.

Assume that the open area of a tube has a surface area S_h and that each tube requires a surface area S. Since the volume velocity must be the same on both sides of the boundary, the particle velocity outside and inside the tube must have the ratio S/S_h, which results in an impedance transformation having the same ratio. Insertion of the impedance ratio into the expressions for reflection coefficient r and the absorption coefficient α_0 gives:

$$\alpha = \frac{4SS_h}{\left(S + S_h\right)^2} \tag{7.5}$$

One notes that even a small surface ratio of 0.5 yields an absorption coefficient of 0.9. In the model, the surface ratio is also a measure of the *porosity* of the material σ:

$$\sigma = \frac{V_{\text{pore}}}{V_{\text{total}}} \tag{7.6}$$

where V_{pore} is the pore volume and V_{total} the total volume required to hold the material.

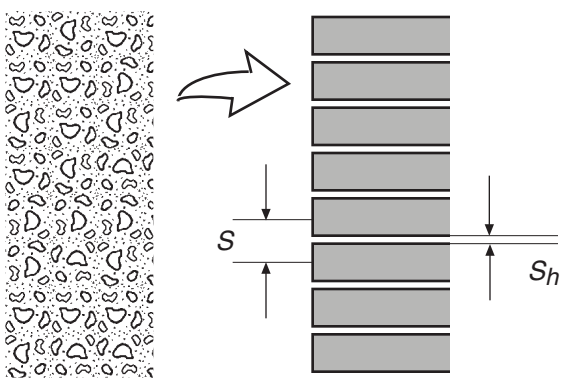


Figure 7.1 In a simplified model of a porous sound absorber one can replace the random pores by thin tubes extending into the material.

This approach is only valid at high frequencies, and the absorption coefficient will approach the value given by Equation 7.1 asymptotically. The porosity is decisive for the absorption properties. Commercial products for sound absorption, such as glass wool, may have a porosity larger than 0.95.

Because of the losses induced by the viscosity of air, the acoustic energy carried in the wave will be converted into heat in the air in the canals. This heat is then transferred to the skeleton of the material. If the canals are narrow, the viscous losses will be considerable. The energy conversion results in isothermal wave propagation and an associated wave propagation speed of down to about 280 m/s. The losses can be modeled by having a complex wave number.

The losses are due to the material's *flow resistance* (the ratio between the pressure needed for the flow, and the flow speed, Ns/m^3). The absorption coefficient of the material α_d will increase as the frequency increases. The absorption coefficient will also vary as a function of properties such as the thickness, extension, placement, and form of the material. The impedance of most materials will be characterized by a small mass component in series with the resistive part.

The most typical placement of sound-absorptive materials is against a hard surface. In this case, the wave propagating inside the material will be reflected as it meets the surface. When the absorber is suspended from a ceiling, of course, both sides will contribute to the sound absorption. The propagation loss of many glass and mineral wool sheets used for sound absorption is about 1 dB per centimeter. Two extreme cases are of special interest:

- *High flow resistance*: In this case, the wave will be rapidly attenuated inside the material, and there will be little reflection at the back side of the material. However, the wave incident on the material will meet a surface having a relatively high impedance and the absorption will be low. This applies to the case of various compressed mineral and glass fiber boards.
- *Low flow resistance*: Since the losses during the wave propagation inside the material will be fairly low in this case, the wave will be quite intense as it is reflected by the hard surface, unless the material has the sufficient thickness. When the wave propagates back toward the outer surface of the material it will be reflected again, and so on. Because of the repeated reflections, one may notice a frequency-dependent sound-absorption coefficient. However, the wave will not be particularly reflected at the outer surface, so the sound-absorption coefficient can be high if the thickness is large.

One can note these types of behaviors in the computed data shown in Figure 7.3, that show how the sound-absorption coefficient varies as a function of the flow resistance of the material. In this and following graphs, unless otherwise indicated, the absorber is assumed to be mounted directly against a hard surface. The computed data shown in Figure 7.2 clearly indicates the influence of material thickness.

The curves in Figure 7.3 show how the absorption at low frequencies increases as the flow resistance is increased, with the penalty of decreased sound absorption at high frequencies. It is clear from these examples that any porous absorber should be optimized with regard to its required sound-absorption properties by proper choice or design of its mounting, thickness, and flow resistance.

Since the losses are associated with flow resistance and material thickness, it is not optimal to position a material characterized by a low flow resistance flush to a hard surface. The particle velocity will be low close to such a surface, because of the interfering, reflected wave. The particle velocity, however, is high in front of the surface at distances of $\lambda/4$ ($+ n\lambda/2$ where $n = 1, 2, 3, \ldots$), so the effective sound

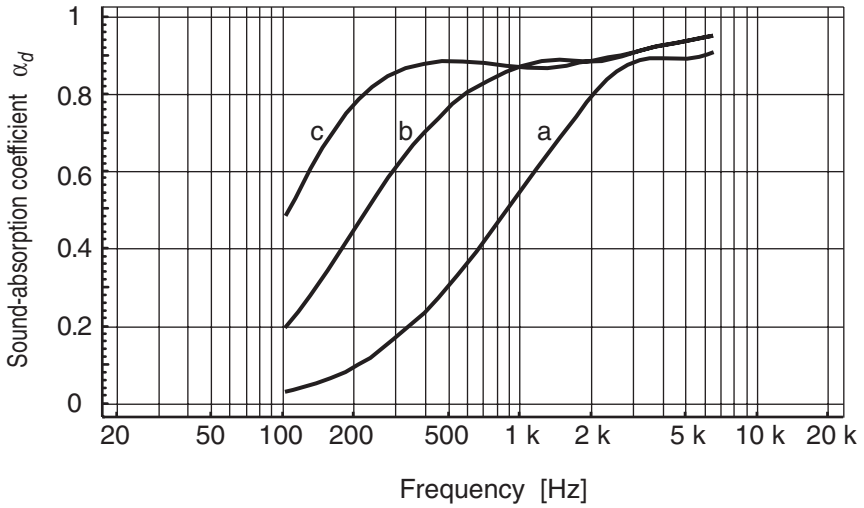


Figure 7.2 Calculated sound-absorption coefficients of a porous sound-absorbing material of different thickness, as a function of frequency. The absorber is assumed to be mounted directly against a hard surface. The specific flow resistance was set to 10^4 kg/m^{3}s and the absorber area was set to 10^2 m^2 to avoid diffraction effects. Curves based on third octave band values: (a) 0.05 m, (b) 0.1 m, and (c) 0.15 m. (After Ref. 7.1)

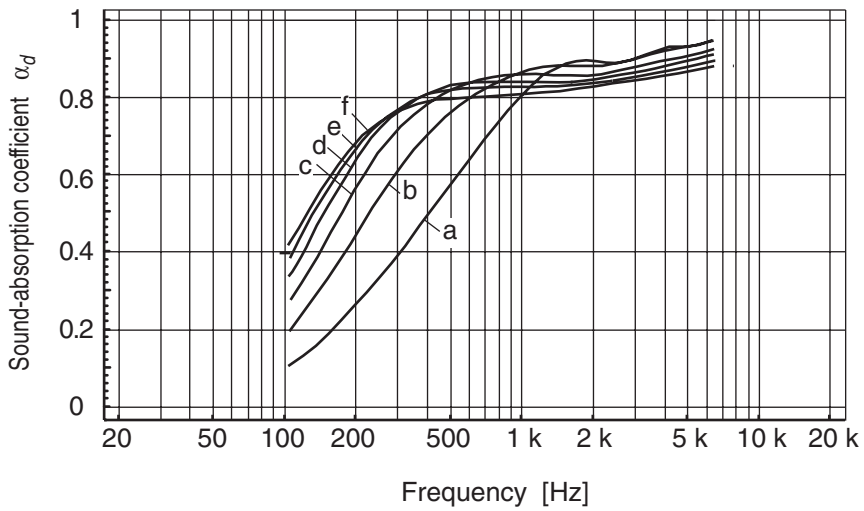


Figure 7.3 Calculated sound-absorption coefficients of porous sound-absorbing materials having different flow resistance as a function of frequency. The absorbers are assumed to be mounted directly against a hard surface. The absorber thickness was set to 0.1 m and the absorber area was set to 10^2 m^2 to avoid diffraction effects. Curves based on third octave band values: (a) $5 \cdot 10^3$ kg/m^{3}s, (b) $1 \cdot 10^4$ kg/m^{3}s, (c) $1.5 \cdot 10^4$ kg/m^{3}s, (d) $2 \cdot 10^4$ kg/m^{3}s, (e) $2.5 \cdot 10^4$ kg/m^{3}s, and (f) $3 \cdot 10^4$ kg/m^{3}s. (After Ref. 7.1)

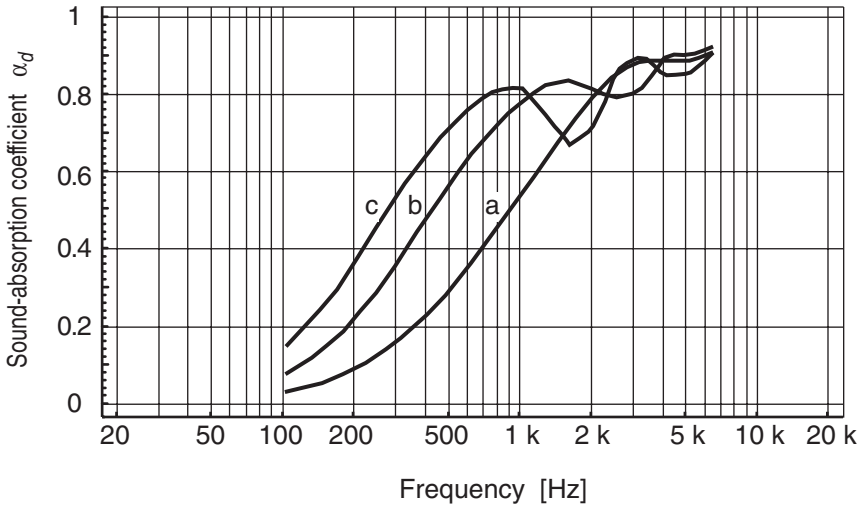


Figure 7.4 Calculated sound-absorption coefficients of a 0.05 m thick porous sound-absorbing material mounted at different distances *d* from a rigid surface as a function of frequency. The specific flow resistance was set to 10^4 kg/m^{3}s and the absorber area was set to 10^2 m^2 to avoid diffraction effects. Curves based on third octave band values: (a) no air space, material mounted directly on surface, (b) d = 0.05 m, and (c) d = 0.15 m. (After Ref. 7.1)

absorption is going to be high if such a material is positioned at these particle velocity maxima. The data in Figure 7.5 also shows this behavior; the absorption peaks of curve b are at frequencies corresponding to air space depths of λ/4, 3λ/4, etc. This can also be noted in curve c of Figure 7.4, so the phenomenon exists even with materials having considerable thickness.

Since the absorption maxima are so frequency specific, thin, flow resistive materials (such as curtains) have to be draped to achieve reasonable frequency-independent sound absorption.

To increase the sound absorption at low frequencies, one should use a suitable combination of material flow resistance, thickness, and air space distance. The data shown in Figure 7.4 exemplify the beneficial influence of such air space.

The decreased speed of sound in porous sound-absorptive materials (typically about 280 m/s) is due to non-adiabatic sound propagation. Because of the shorter wavelengths a somewhat thinner material can be used to achieve a desired sound absorption than if the wavelength were that of free propagation. A suitably designed porous absorber, mounted against a hard surface and having a thickness of 0.15 m, will have good sound-absorption properties down to a frequency of approximately 400 Hz.

Porous absorbers are easy to use and offer good sound-absorbing properties but have a number of disadvantages and problems. They are:

- Sensitive to mechanical damage
- Dust collectors—particularly close to air exhausts and intakes
- Likely to release fibers hazardous to health
- Difficult to paint since the pore openings will be clogged
- Cannot be washed because pores will be clogged by water due to capillarity
- Prone to condensation because of lack of a diffusion barrier

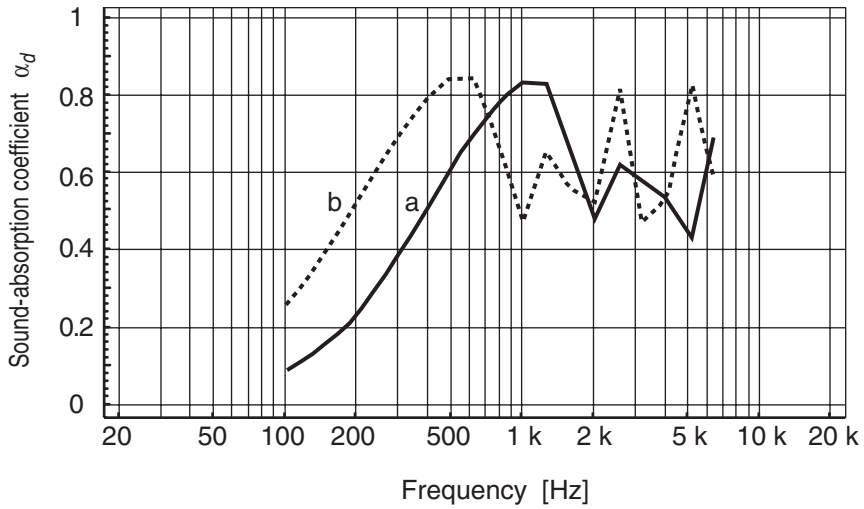


Figure 7.5 Calculated sound-absorption coefficients of a 0.005 m thick porous sound-absorbing material mounted at different distances from a hard surface as a function of frequency. The specific flow resistance was set to 10^5 kg/m^{3}s and the absorber area was set to 10^2 m^2 to avoid diffraction effects. Curves based on third octave band values. (a) 0.1 m and (b) 0.2 m. (After Ref. 7.1)

To avoid some or all of these disadvantages, it is possible to cover the surface of a porous material by various sheets made of glass fiber cloth or plastic and metal foils, as well as various boards made of gypsum or wood, for example. Such covers may have different degrees of limpness, mass, and compliance. Such protective sheets will also help avoid material fibers from contaminating the surrounding air, which may be a health hazard. The apparent surface mass of the limp sheet will increase if the sheet touches the absorber.

As an example, assume a free, limp sheet, characterized by mass only, on which a plane wave impinges perpendicularly, in front of an otherwise perfect sound absorber, having an impedance of Z_0, that is, $\alpha = 1$, as shown in Figure 7.6. We use the impedance analogy concept discussed in Section 1.8 because it is intuitive.

Assume further that the sheet has an area S and a mass M. The mass per unit area then is M/S, and the specific acoustic impedance seen by the incident wave will be:

$$\underline{Z}_2 = \frac{j\omega M}{S} + Z_0 \tag{7.7}$$

Inserting Equation 7.7 into the expression for the reflection coefficient Equation 1.53, we find:

$$\underline{r} = \frac{\dfrac{j\omega M}{S} + Z_0 - Z_0}{\dfrac{j\omega M}{S} + Z_0 + Z_0} = \frac{j\omega M}{j\omega M + 2Z_0 S} \tag{7.8}$$

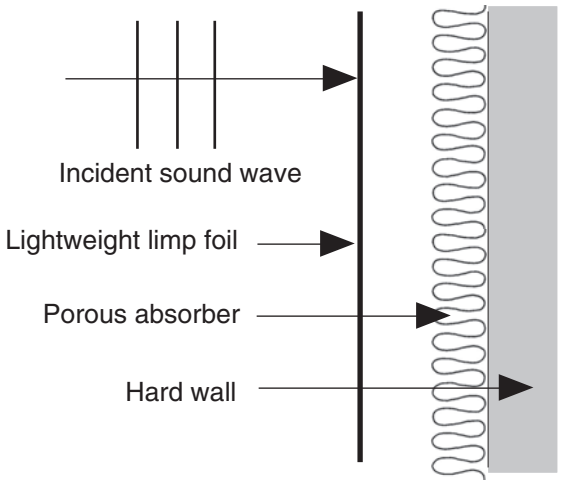


Figure 7.6 A porous sound absorber mounted against a hard wall and covered by a thin, lightweight, limp protective foil.

In practice, it is better to have the sound-absorption coefficient expressed using the concept of surface mass—that is, the *mass per unit area* m'' [kg/m^2]. Using Equation 7.4, we obtain the absorption coefficient as:

$$\alpha_0 = \frac{1}{1+\left(\frac{\omega m''}{2Z_0}\right)^2} \tag{7.9}$$

One notes that the sound absorption decreases as the frequency increases, as a result of the inertia of the sheet.

If an absorption coefficient larger than 0.8 is considered necessary for good sound absorption, the mass per unit area must be lower than:

$$m'' = \frac{Z_0}{\omega} \approx \frac{65}{f} \tag{7.10}$$

This, in turn, means that any protective sheet must be extremely lightweight, as also indicated by the data shown in Figure 7.7.

For the limp foil to have low effective surface mass, it must also be mounted so that it does not touch the porous absorber, since otherwise its effective mass will increase. Most surface protected commercial sound-absorptive materials have the foil attached to the skeleton of the material. This requires the foil to have a low compliance, so that it can move over the pore entrances in spite of the contact.

A constriction to the air (e.g., a perforated or slit sheet used to cover the absorber) will also behave as an acoustic mass as discussed in Chapter 1. The presence of this acoustic mass in front of the absorber, due to the air in the holes, will result in loss of high frequency sound absorption (assuming a wavelength much larger than the inter-hole distance).

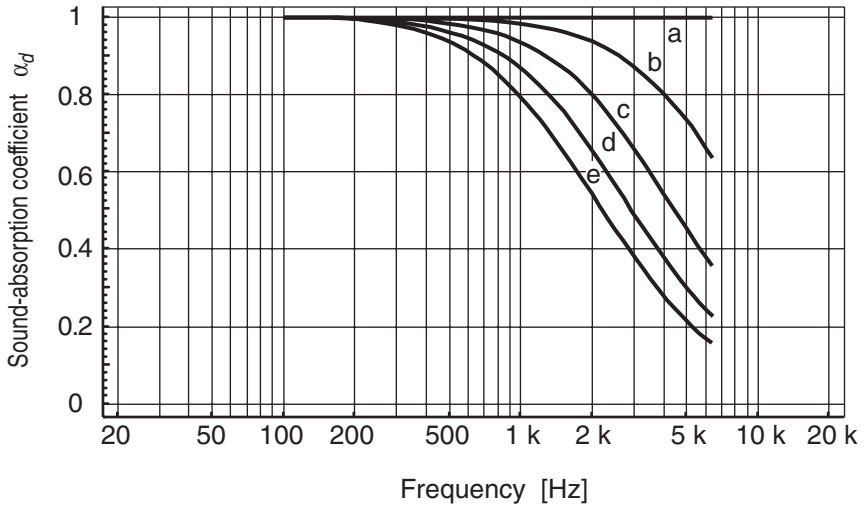


Figure 7.7 Calculated sound-absorption coefficients of an ideal porous sound-absorbing material protected by limp foils having different surface mass as a function of frequency. Curves based on third octave band values: (a) No foil, (b) 0.025 kg/m^2, (c) 0.05 kg/m^2, (d) 0.075 kg/m^2, and (e) 0.1 kg/m^2. (After Ref. 7.1)

As an example, consider a thin, free, perforated sheet in front of an ideal absorber. The sheet has a quadratic hole pattern, with an inter-hole distance of b, and circular holes of diameter $2a$.

The acoustic mass associated with each such hole in a thin sheet is M_A. The resulting expression for the sound-absorption coefficient can be shown to be:

$$\alpha_0 = \frac{1}{1+\left(\frac{\omega M_A S_{cc}}{2Z_0}\right)^2} \tag{7.11}$$

where

$$S_{cc} = b^2 \tag{7.12}$$

and

$$M_A = \frac{\rho_0}{2a} \tag{7.13}$$

One obtains an expression, corresponding to expression 7.10, for the upper limiting frequency for $\alpha_0 > 0.8$ as:

$$\omega < \frac{2ac_0}{b^2} \tag{7.14}$$

If the inter-hole distance is of the order of one half wavelength or larger, one will have appreciable sound reflection from the surface between the holes.

For the mass effect by holes of air in a surface to be negligible, the perforation rate—hole area divided by surface area—has to be larger than 0.3. Of course, increasing the thickness of the perforated sheet will lead to more stringent requirements.

Until now, sound incidence has been assumed as perpendicular to the surface of the absorbers. Extending the calculations to the case of nonperpendicular incidence, one finds that a locally reacting surface, having a real impedance of Z_2, has an angle-dependent sound-absorption coefficient $\alpha_0(\varphi)$ that varies by the incidence angle φ as:

$$\alpha_0(\varphi) \approx \frac{1}{\frac{1}{2} + \frac{1}{4}\left(\zeta\cos(\varphi) + \frac{1}{\zeta\cos(\varphi)}\right)} \tag{7.15}$$

where

$$\zeta = \frac{Z_2}{Z_0} \tag{7.16}$$

Figure 7.8 shows how the angle-dependent sound-absorption coefficient behaves as a function of φ.

The sound-absorption coefficient for diffuse sound incidence α_d is usually used for room acoustics since it represents an average of $\alpha_0(\varphi)$ for all angles. In a diffuse sound field, sound waves having oblique incidence carry most of the energy, so usually $\alpha_d > \alpha_0$.

Ideally, one would want to know the sound-absorption coefficient for the particular sound field that one is dealing with. Since the absorber influences the sound field, there is not an immediate solution. Typically, one has to be satisfied by two types of absorption-coefficient data, α_s and α_r, which are measured approximations to α_d and α_0.

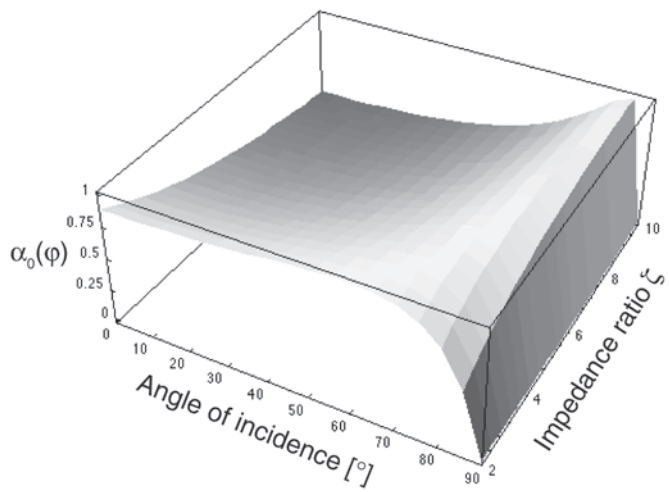


Figure 7.8 The sound-absorption coefficient of a porous sound absorber as a function of the impedance ratio $\zeta = Z_2/Z_0$ and the angle of incidence of the sound wave.

The measurement of α_S is done using reverberation chambers, usually according to the ISO 354 standard, and requires test sample areas of at least 10 m^2. The measured absorption coefficient will vary between laboratories as a result of different laboratory room geometry, diffuser, and sample placement.

When developing new absorption materials and combinations, it is impractical and expensive to use such large samples. One instead uses the standing wave tube method to measure α_r. There is usually considerable difference between the α_r and α_S values so care should be exercised in the use of measurement results.

Until now, the absorber surface has been assumed to be infinite or at least much larger than the wavelength of interest. Because of diffraction, however, one finds that the sound absorption of any surface varies with the global surface geometry. Figure 7.9 shows calculated α_d data for different sizes of a square sound-absorptive panel mounted in a hard surface.

One notes that the absorption coefficient varies with frequency and that maximum absorption is obtained when the length of the square panel is about twice the wavelength. This shows that it may be advantageous to use many smaller panels, mounted at sufficient distance from one another, if one wants to maximize the sound absorption of a certain amount of material. Such use also leads to increased sound field diffusitivity in a room, since the absorbing patches will act as diffusers. For patches that are small compared to the wavelength of sound, the sound absorption will be independent of the direction of sound incidence (see Reference 7.2).

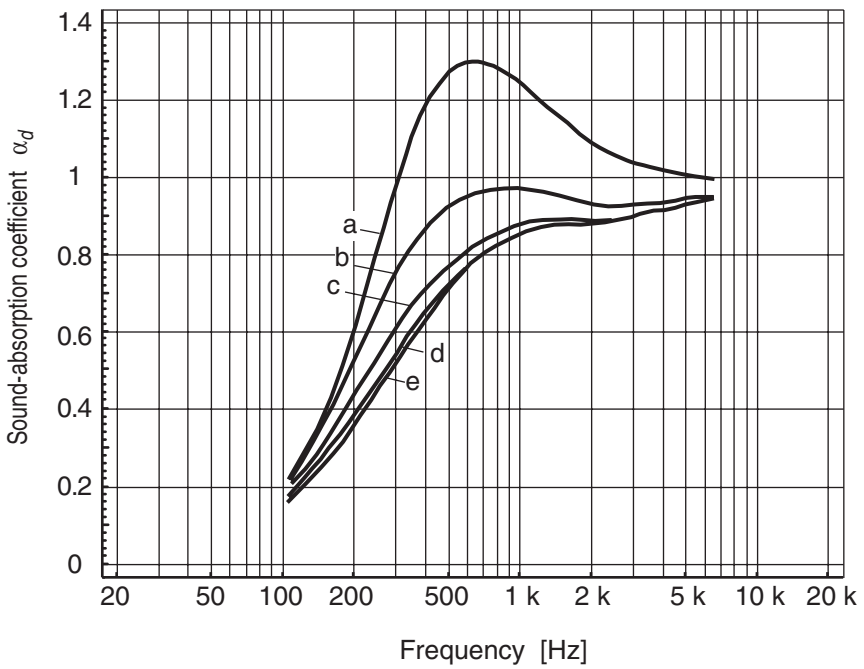


Figure 7.9 Calculated sound-absorption coefficients of a square patch of porous sound-absorbing material of different areas on a rigid surface, as a function of frequency. The specific flow resistance was set to 10^4 kg/m^{3}s and the absorber thickness was set to 0.1 m. Curves based on third octave band values: (a) 1 m^2, (b) 10 m^2, (c) 100 m^2, (d) 1,000 m^2, and (e) 10,000 m^2. (After Ref. 7.1)

7.4 RESONANCE ABSORBERS

The sound absorption by resonance absorbers relies on the losses in their acoustical and mechanical constructions that are set in motion by sound. The most common types of resonance absorbers are *Helmholtz*, *membrane*, *panel* and *slit absorbers.* Thin panels, walls, and windows can also act as resonant sound absorbers. One can also combine porous and resonant absorber action in a material. Porous materials can be designed to hold resonant elements and vice versa.

Resonance absorbers are primarily used for absorbing sound having frequencies below 400 Hz. In isolated cases, one can design resonance absorbers to cover a wide frequency range, including high frequencies. One example of such a construction, having reasonably smooth, wide range sound absorption, is a wall of *air-bricks* behind which is an airspace with a porous absorber (see Figure 7.13).

A section of a membrane absorber, shown in Figure 7.10, is a mechano-acoustical resonator where the mass of a flexible membrane—for example, impervious cloth or a plastic foil—is mounted so that a closed airspace is formed between the membrane and the underlying wall. This air space acts as a compliance, as explained in Chapter 1. Typical membrane absorbers are windows, wooden floors on joists, and wooden wall paneling on battens.

The losses consist of the mechanical losses in the membrane and the acoustical losses in the air space that can be filled with a porous absorber. Note that the addition of a porous absorber will affect the resonance frequency and the bandwidth of the absorption peak.

Because the resonant elements, the membrane, and the airspace behind it are small compared to wavelength (as it will turn out), it is convenient to use the concept of acoustical impedance to find the condition for resonance.

Using the ideas introduced in Chapter 1, we can write the acoustical impedance for a membrane absorber as:

$$\underline{Z}_{A,\text{tot}} = j\omega M_A + R_A + \frac{1}{j\omega C_A} \tag{7.17}$$

That is, it consists of mass, compliance, and loss. Remembering the relationship between mechanical and acoustical impedance, one can write the components of the acoustic impedance as:

$$M_A = \frac{M}{S^2} = \frac{m''}{S} \tag{7.18}$$

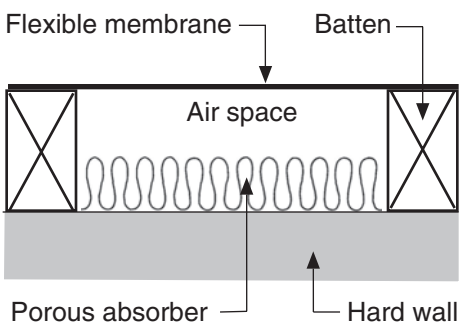


Figure 7.10 The basic design of a membrane absorber. The porous absorber may be placed in different ways in the air space to achieve different resonance and sound-absorption characteristics of the design.

and

$$C_A = \frac{Sd}{Z_0 c_0} \tag{7.19}$$

where m is the mass per unit area of the membrane, S the surface area, and d the depth of the air space.

It is difficult to estimate R_A in advance. The losses are associated with mechanical losses in the membrane, losses at the edges of the membrane, reradiation loss, and losses in the air space. The losses are best found by measurement of the bandwidth of the resonance. The concept of a Q-value to characterize the damping is useful here also. Membrane absorbers made of wood panels typically show low Q-values, on the order of 0.5 to 1.

The resonance frequency f_0 can be obtained by setting the imaginary part of Equation 7.17 to zero that gives (for perpendicular sound incidence):

$$f_0 = \frac{c_0}{2\pi}\sqrt{\frac{\rho_0}{m'' d}} \tag{7.20}$$

For diffuse field incidence, the resonance frequency is about 30% higher.

Typical values for f_0 are in the 50 to 400 Hz range when one uses panels having a thickness of about a centimeter and air spaces with a depth of a few centimeters. In these cases, one usually finds that α_S is 0.8 or lower.

If one wants high absorption over a small frequency range, one must use a thin panel and a large airspace distance. Thin panels often exhibit multiple absorption peaks due to air space room resonances. Membrane absorbers and resonator panels are discussed in detail in Reference 7.11.

A Helmholtz resonator is a single resonator where both mass and compliance are due to acoustical components. An open bottle is an example of a simple Helmholtz resonator. A Helmholtz resonator can also be integrated into a wall as shown in Figure 7.11.

The neck constriction is the acoustical mass component and the enclosed airspace the acoustical compliance. Since the Helmholtz resonator has intrinsically low losses, the reradiation of energy to the surroundings is an important factor in design. The resonator can be placed free or inserted in a wall or other surface; the absorption maximum will depend on placement (including placement relative to room corners).

Using the expressions for the acoustic components found in Chapter 2, one finds that the resonance frequency f_0 is determined by:

$$f_0 \approx \frac{c_0}{2\pi}\sqrt{\frac{\pi a^2}{V\left(l_n + \frac{\pi}{2}a\right)}} \tag{7.21}$$

where a is the neck radius, l_n the neck length, and V the enclosed air volume. Note that the end correction for an open tube has been incorporated when writing the effective neck length. Since the end correction will depend on the acoustic environment of the mouth opening, it is best to tune the resonator at its place of operation.

One can adjust the losses of a Helmholtz absorber by inserting a flow resistance in the throat or by adding a porous absorber inside the compliant air volume. The highest absorption will be found when the losses in the acoustic system of the bottle match those of the reradiation. The absorption peak is

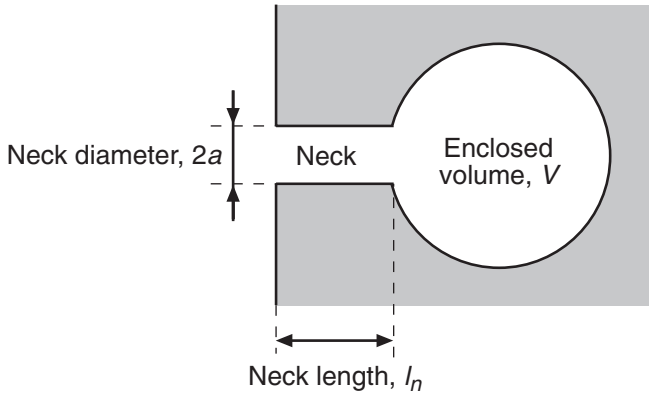


Figure 7.11 Section showing a Helmholtz resonator. The Helmholtz resonator consists of an air volume coupled to the outside air by an open tube. Here the resonator is shown integrated into a wall.

then somewhat narrow. By using a number of resonators having different resonance frequencies, one can achieve more wide band absorption than by using a single resonator.

The absorption of a Helmholtz absorber is best described by the idea of an absorption area, since the absorption area can be much larger than the area of the neck. One can show that the maximum absorption area A_{max} that can be obtained for a Helmholtz absorber inserted in a wall is:

$$A_{max} = \frac{\lambda_0^2}{2\pi} \tag{7.22}$$

where λ_0 is the wavelength of sound at the resonance frequency. The A_{max} absorption area is only obtained at the resonance frequency and the absorption peak is then quite narrow in frequency. Figure 7.12 shows measurement results that illustrate how the absorber draws in sound.

A wide bandwidth requires a large enclosed volume. Note that the shape of the volume is unimportant as long as its airspace has small dimensions compared to the wavelength.

If one listens close to a low loss Helmholtz resonator, it is sometimes possible to hear the ringing sound of the reradiation. Examples of such resonators are some musical instruments, for example guitars and violins.

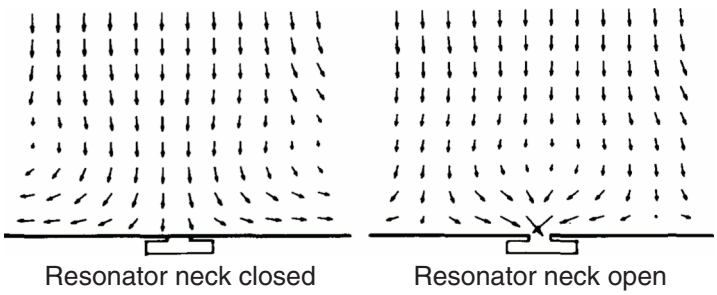


Figure 7.12 Measurement results showing power flow (intensity vectors) over a hard surface with an embedded Helmholtz resonator for normal sound incidence. Note how the open neck resonator draws in energy over a wide area. (From Ref. 7.12)

An advantage of Helmholtz resonators in room acoustics is that they can be used to control the decay of individual low frequency room resonances. Helmholtz resonators and other tuned low frequency sound absorbers are sometimes called *bass traps*.

Finally it must be mentioned that, in low loss resonators, the flow velocity u of the air in the resonator neck may become so high that turbulence occurs (typically for u >7 m/s). The turbulence is a nonlinearity that causes disturbing noise. An example of such effects can be found in bass reflex loudspeakers using open ports, as discussed in Chapter 8. By streamlining the port openings, it is possible to reduce the turbulence. In these cases, it may be difficult to calculate the neck length of the port, and it is usually necessary to use numerical simulation or measurement of the effective port length.

An important type of air-air resonator is a *resonator panel* that consists of panels having perforations, such as slits or circular holes. One can regard resonator panels as a large number of individual Helmholtz resonators next to one another as indicated in Figure 7.13.

Because of symmetry, it is possible to remove the internal side walls of the air volumes. The resonance frequency of such a panel can be determined in the same way as for a single resonator; the enclosed air volume will be:

$$V = S_{cc} d \tag{7.23}$$

where d is the depth of the airspace, $S_{cc} = b^2$, and b is the distance between hole centers, assuming a square hole pattern. If the perforations or holes occupy more than 30% of the surface area, the panel will be quite transparent to sound, and there will be no resonance. Note that the panel will act as a reflector at high frequencies.

As mentioned, the holes do not need to be cylindric—slits and other hole shapes can be used as well, as long as the maximum crosswise dimensions are much smaller than the wavelength. By damping the resonance using flow resistive materials, such as cloth or mineral wool, one can obtain wide bandwidth, typically at least one third octave band. The sound-absorption coefficient at the resonance frequency is typically high; α_s may be close to unity.

It is often easier to manufacture a resonant sound-absorbing panel by using slits rather than circular holes, for example, by using strips with a small airspace between the strips or by using a saw to make slits in a gypsum or particle board sheet. The resonance frequency of such a slit resonator panel having thin slits can be calculated using:

$$f_0 \approx \frac{c_0}{2\pi}\sqrt{\frac{S_{\text{slit}}}{V l_{\text{eff}}}} \tag{7.24}$$

where the effective neck length l_{eff} is given by:

$$l_{\text{eff}} = l_n + b\left(\frac{1}{2} + \frac{2}{\pi}\ln\left(\frac{c_0}{\pi b f_0}\right)\right) \tag{7.25}$$

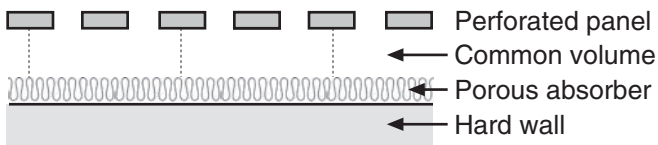


Figure 7.13 The resonator panel uses the acoustic mass of the air in the panel holes to resonate with the acoustic compliance of the trapped air. Porus absorber sheet added to broaden the resonance.

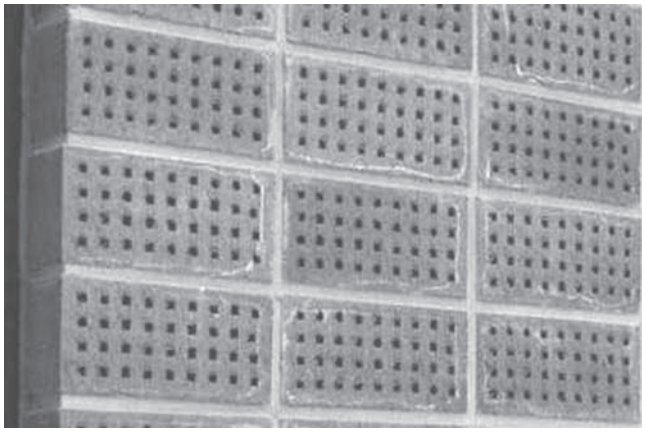


Figure 7.14 The air-brick absorber can be thought of as a panel absorber having a thick panel or as a Helmholtz absorber with a subdivided hole. (Photo by Mendel Kleiner.)

where S_{slit} is the slit area per unit (of panel), l_n the thickness of the panel, b the slit width, and V the enclosed volume per unit area. The expression requires that the slit width be much smaller than the wavelength at the resonance frequency. An iterative approach to the determination of the resonance frequency is needed since Equation 7.25 includes the resonance frequency inside the logarithm. Slit absorbers are often characterized by a fairly high sound absorption over a comparatively wide frequency range, that is they have a low Q-value.

Air-brick sound absorbers with a closed airspace behind the bricks (shown in Figure 7.14) function as Helmholtz absorbers but will have a wider frequency range of useful sound absorption because of additional high frequency sound absorption at frequencies where the length of the holes are a multiple of half-wavelengths.

Translucent Absorbers

Most materials and constructions used for absorbers are opaque and will impede the flow of light. Using the mechanical absorber idea discussed in the previous paragraph, it is possible to make *translucent sound absorbers*, using either glass or plastics. Microperforation can be achieved using lasers or mechanical means. The holes generated will exhibit comparatively high flow resistance. Such microperforated panels and foils may be used with enclosed airspace to design various membrane and resonator panel sound absorbers. One such design using micro-perforated plastic foil sails is shown in Figure 7.15.

Absorbers Using Mechanical Resonance

It must be noted that any surface that can be put into motion by sound waves is likely to act as a sound absorber, because of its transmission or its internal losses due to friction. Some mechanical sound absorbers are mechanically similar loudspeakers (without drive mechanism) and will act as resonance absorbers, as will loudspeakers.



Figure 7.15 Microperforated foils can be successfully employed as translucent sound absorbers. (Microsorber photo courtesy of KAEFER, Germany)

7.5 VARIABLE ABSORBERS

Ideally one wants to be able to adjust the sound absorption of the additional absorbing areas in a room to that required in the specific case, since the natural absorption in a room varies (e.g., as a function of the number of people present). As mentioned previously, various uses require different reverberation times. Using *variable absorbers*, it may be possible to adjust the reverberation time to the desired value irrespective of the number of people in the audience. Some examples of the construction of such variable absorbers are shown in Figure 7.16.

7.6 AUDIENCE ABSORPTION

The natural sound absorption by the audience is often the dominating sound-absorption component at mid frequencies in enclosed performance spaces, such as theaters and concert halls. At high frequencies, usually above 4 kHz for large rooms, the sound absorption induced by lossy sound propagation in the air will dominate as discussed in Chapter 1.

If one wants a certain reverberation time (determined by the air volume of the room), a certain volume per audience area is needed, depending on seating. The audience sound-absorption coefficient depends on the number of persons per unit area and the type of upholstered chairs. The effective audience sound-absorption coefficient also depends on the diffusitivity of the sound field so the design of

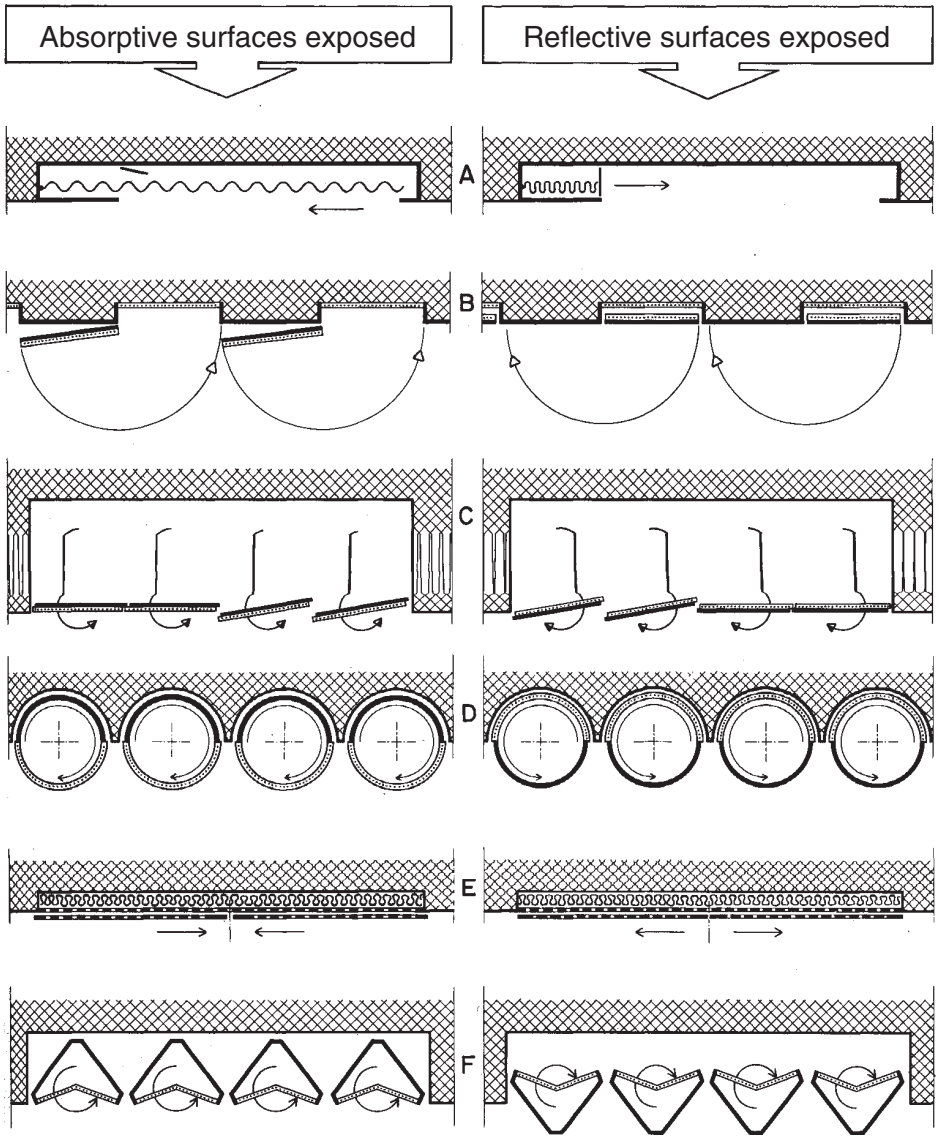


Figure 7.16 Examples of constructions of variable absorbers. (Adapted from Ref. 7.4)

large spaces to obtain a particular, suitable reverberation time may be difficult. Some examples of the sound-absorption coefficient of concert hall seats are shown in Figure 7.17.

Because of the sound field properties at the edges of the seating area, it is necessary to compensate for the sound that enters from the sides. The full *acoustical* audience area is approximately the seating block floor area plus areas of strips 0.5 m wide around the audience seating blocks, except for sides at balcony rails or walls.

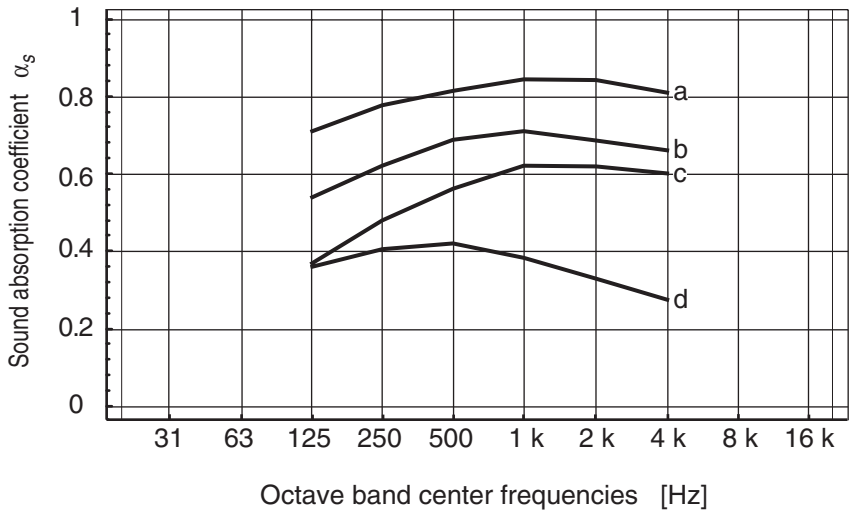


Figure 7.17 Unoccupied chair sound-absorption coefficients from four different groups of concert halls, according to degree of seat upholstering: (a) heavily upholstered seats, (b) medium upholstered seats, (c) lightly upholstered seats, and (d) extra lightly upholstered seats. (Adapted from Ref. 7.5)

The design of chairs present a special problem when they have the same sound-absorption. Such chair design has to be somewhat empirical and can be quite expensive, since at least some 16 to 25 chairs are usually necessary for measurement in a reverberation chamber to have a reasonable approximation to the sound field around the audience. The effective audience sound absorption also depends on edge effects, since one typically must add the sound absorption by the perimeter of the seating areas/sections. Typically, one adds an extra width of about 0.5 m to the sides of the floor area covered by seating.

The sound absorption by individuals (persons standing or sitting far apart, such as a few musicians on stage) will still be on a per-person basis and will depend on dress. Some values are shown in Table 7.2 (see Reference 7.8). Such absorption is hard to measure correctly, since it will depend on the incident sound field in the particular case. In most cases, the additional sound absorption by such persons will be small and negligible. Note however that audience and individuals contribute effectively to the scattering of sound.

7.7 REFLECTORS

Reflectors are usually used to distribute the sound from a speaker or an orchestra to the audience or to improve the acoustical conditions for musicians on stage. The room impulse response can sometimes be improved by careful use of reflectors. Reflectors can be planar or curved. Sometimes curved reflectors are used as diffusers.

For a reflector to be efficient, its surface must have dimensions much larger than the wavelength of sound to be reflected and sufficient mass to prevent it from being set into motion by the incoming sound. The mass per unit area m should be such that $|r| > 0.9$ in the frequency range of interest.

Note that the angle of incidence affects the efficiency of the reflector and that groups of reflectors will show different behavior than single reflectors. In addition, the reflection properties of reflectors will depend on their surface structure.

Figure 7.18 shows an example of calculated reflection coefficient magnitude, on the normal, for sound perpendicularly incident on a free, plane, immovable reflecting strip. One notes that the reflection coefficient is small at low frequencies but at high frequencies it will be close to unity (see the discussion of Fresnel zones in Chapter 4).

At low frequencies, and for large angles of incidence, the reflection by strips and other types of planar reflectors will tend to become diffuse. Because of interference, the reflections by groups of such reflectors will tend to be directional. Their reflection properties are best investigated using physical scale modeling.

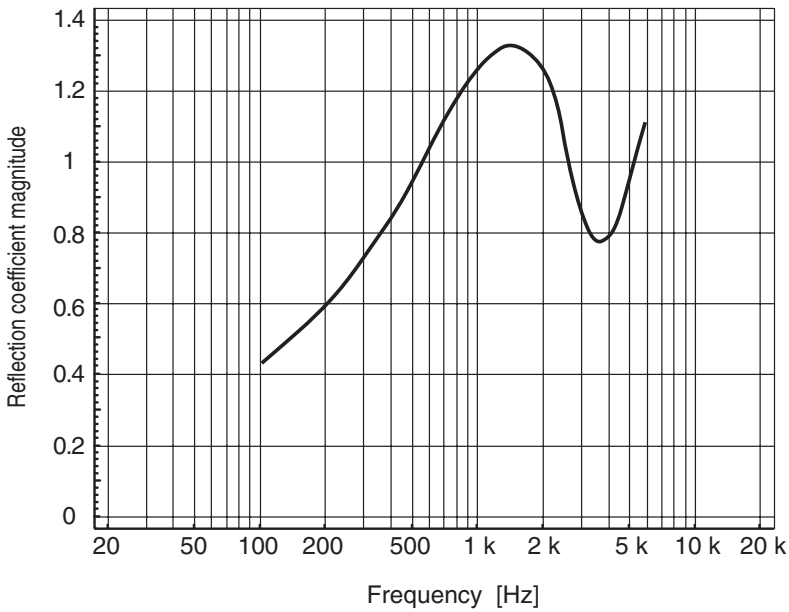


Figure 7.18 Example of the reflection coefficient for a free rectangular panel strip, 2.1 m wide, for perpendicular sound incidence. (Adapted from Ref. 7.6)

7.8 BARRIERS

In many cases, something is needed to reduce the sound level of some activity in a room, without building a wall to separate the noise-generating part of the room from the rest. A partial wall or a *barrier* can then be used to achieve a sound shadow. (Note that there are cases in which items such as pillars may cause an unwanted shadow.)

The effectiveness of a barrier is determined by the following factors:

- Wavelength of sound
- Angle of sound incidence relative to the barrier
- Acoustic and mechanical properties of the barrier

Often the frequency range of interest is 200 Hz to 4 kHz, which corresponds to a wavelength range of 0.08 to 1.7 m. This means that the barrier height is of the order of the longest wavelength. The result is considerable diffraction around the barrier and poor shadow action. Figure 7.19 shows the geometry used in the following discussion.

One can show that the insertion loss by the barrier, ΔL [dB], depends on the barrier geometry and the locations of the source and receiver. The shorter the wavelength and the larger the difference between the path via the edge and the path in absence of the barrier, the larger insertion loss will be. The parameter X_0 is used to characterize this geometrical condition and is determined by:

$$X_0 = \sqrt{2\pi \frac{R_1 - R}{\lambda}} \tag{7.26}$$

where R is the direct distance through the barrier between the source and the receiver points, R_1 the distance between the source and the receiver points for the closest path over the barrier edge, and λ the wavelength.

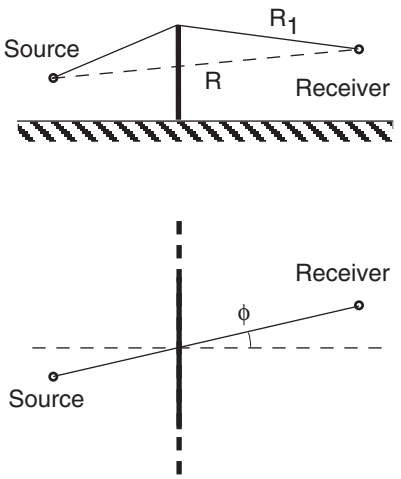


Figure 7.19 Geometry for calculation of the insertion loss of an infinitely long barrier. R is the direct distance through the barrier. R_1 is the distance of the closest path over the barrier edge.

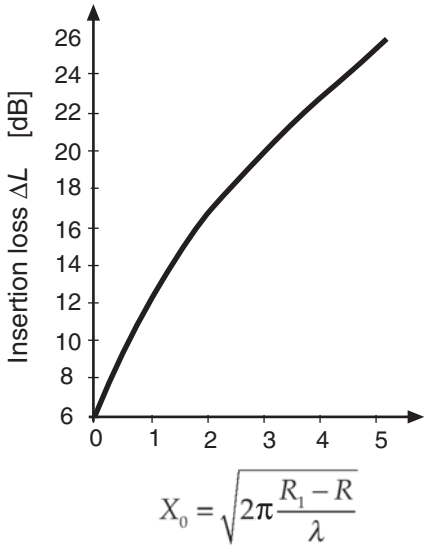


Figure 7.20 Insertion loss of a barrier in a free field as a function of the parameter X. Note the modification of $X(\varphi)$ needed in case $\varphi > 0$. (Adapted from Ref. 7.7)

For a point source near an infinitely wide barrier, made of a totally sound-absorptive material, the insertion loss ΔL achieved by using the barrier is approximately:

$$\Delta L \approx 11 + 20\log(X_0) \tag{7.27}$$

for $X_0 >> 1$.

On the line of sight, the loss is always 6 dB. In the case that X_0 is not much larger than unity, the curve shown in Figure 7.20 may be used to estimate the insertion loss. One notes that the insertion loss increases by 3 dB/octave for small values of X_0.

If the angle φ between the direct propagation path and the barrier is smaller than 90°, the parameter X_0 needs to be replaced by the slightly modified value X_φ:

$$X_\varphi = X_0\sqrt{\cos(\varphi)} \tag{7.28}$$

The difference in insertion loss between a sound-reflective and sound-absorptive barrier is small. A thick barrier will usually result in better insertion loss than a thin barrier, since sound has to propagate over two edges.

It is important that the barrier has sufficient sound insulation, so that sound is not transmitted through the barrier. This typically requires a mass per unit area of more than 10 kg/m^2 for barriers of the type common in open plan offices. It is also necessary to consider the sound reflection from the ceiling above the barrier.

7.9 DIFFUSERS

A common problem in room acoustics is to diffuse the sound reflection by planar surfaces. In most rooms for music and speech, a suitable amount of diffuse reflection will remove unwanted behavior such as flutter echo and improve the smoothness of the audio. A *diffuser* is designed to primarily scatter and diffuse sound so that the reflection is not planar. For the diffuser to work properly, it needs surface irregularities a quarter wavelength or larger at the frequency of interest, but even small surface unevenness is advantageous. Niches 0.3 to 0.4 m deep are usually sufficient for good results. The irregularities should be randomized for optimum effect, but this is often not possible for architectural reasons. Figure 7.21 shows an example of a commercially available diffuser for use in recording studios and home theaters.

Efficient scattering can be obtained using surfaces having specially designed depth profiles that have low spatial autocorrelation. An example of such a depth profile is shown in Figure 7.22. These diffusers operate on the principle that the phase of the reflected wave from the surface should be random for diffuse reflection (see Reference 7.9). Assume a plane wave is incident at an angle ϑ on a plane surface consisting of strips that generate phase jumps because of the distance the incoming wave has to travel down and up through the slots. If all slots reflect identically, one obtains the pressure at far distance:

$$\underline{p}(k, d, \vartheta) \propto \sum_n R_n e^{jkd\sin(\vartheta)} \tag{7.29}$$

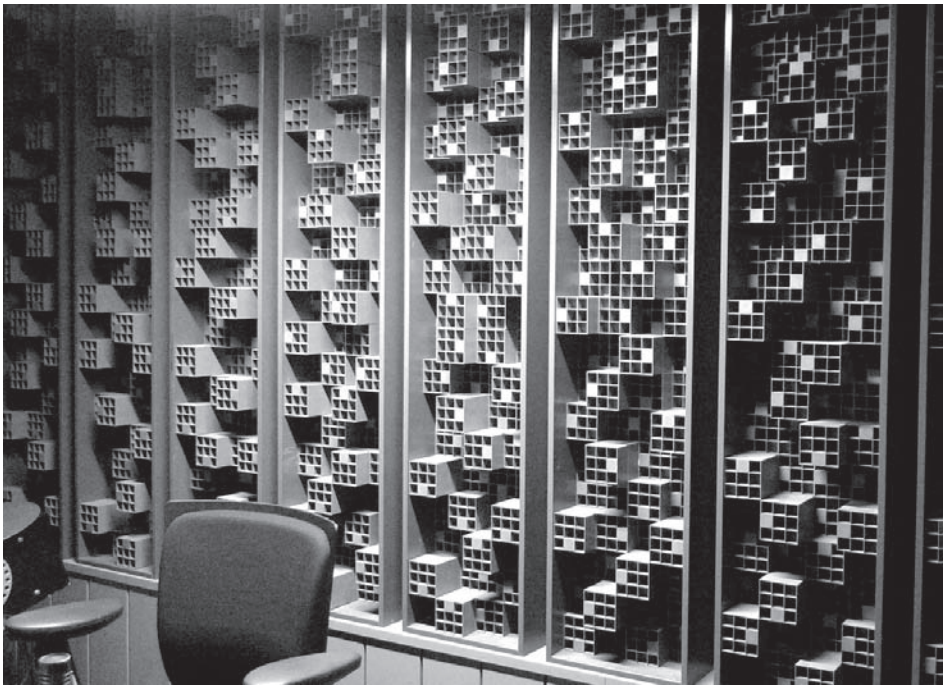


Figure 7.21 An example of a commercially available two-dimensional diffuser for use in recording studios and home theaters. (Photo by Mendel Kleiner.)

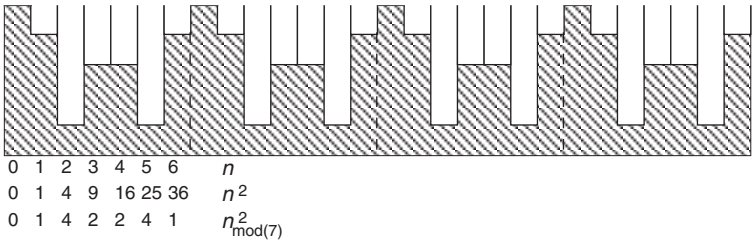


Figure 7.22 An example of the depth profile of a sound diffuser based on the use of a quadratic residue sequence techniques (period = 7 slots). Note that the slit dividers are necessary for proper function.

where n indicates the strip number, R_n the reflection coefficient of the strip, and the slot is d wide. If there are different phase jumps ψ_n at each respective slit, one instead finds:

$$\underline{p}(k, d, \vartheta) \propto \sum_n R_n e^{jkd\sin(\vartheta)+\psi_n} \tag{7.30}$$

Assume that the phase jumps are generated by slits having depths described by a quadratic residue sequence (QRD). A key property of QRD sequences is that they, over the length of the sequence, behave as white noise, that is, as a surface that has random *height*. This means that the phase jumps will be random as the slits reflect the incoming wave. The phase jumps will have the form:

$$h_n \propto \psi_n \tag{7.31}$$

where

$$\psi_n = 2\pi\frac{n^2}{p} \tag{7.32}$$

where $n = 2$ and p is a prime. Since the phase repeats at multiples of 2π, one can replace s^n by the remainder s^n that is obtained from:

$$s_n = n^2_{\mathrm{mod}(p)} \tag{7.33}$$

Using $p = 7$ and $n = 2$, one obtains the sequence $s_n = 0, 1, 4, 2, 2, 4, 1$ after which the sequence repeats periodically.

Since the wave traveling down the slit reflects at the bottom of the slit with a reflection coefficient of 1, the wave returns to the top with a total phase change $2kh_n$. The slit depth turns out to be:

$$h_n = s_n\frac{\lambda_l}{4\pi} = \pi\frac{s_n}{pk_l} \tag{7.34}$$

where k_l is the wave number at the lowest frequency. The lowest nominal working frequency of such a diffuser is the frequency at which the maximum slot depth is approximately $\lambda/4$. The upper operating frequency f_u is given by:

$$f_u = \frac{f_l}{p-1} \tag{7.35}$$

The slots should have widths smaller than $\lambda/2$ at the highest operating frequency, and the slot walls should be thin. It is clear that these diffusers are primarily mid-frequency range devices for practical manufacturing reasons. When sound is incident parallel to the slot walls there will be little scattering; it is only when sound is incident at right angles to the slot that the diffuser works optimally. Also note that the theory presented above assumes incident plane waves.

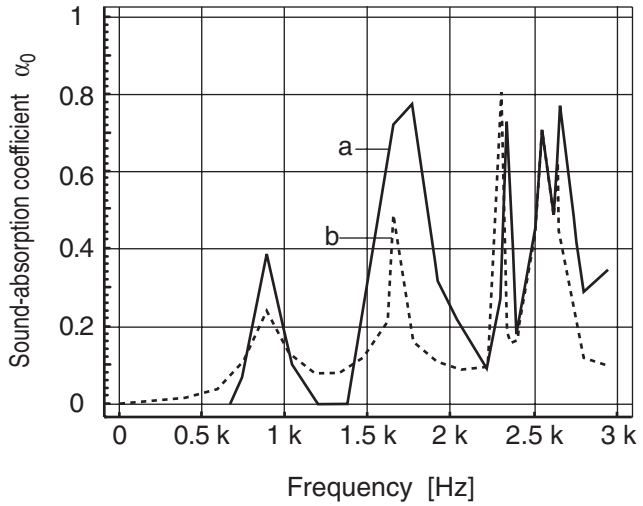


Figure 7.23 Sound-absorption coefficients of a one-dimensional diffuser similar in design to that shown in Figure 7.21. Note the use of a linear frequency axis: (a) measurement and (b) prediction. (Adapted from Ref. 7.10)

While the mathematically defined diffusers discussed so far have had one-dimensional shape variation, they can be calculated for two-dimensional depth shapes as well. Improved properties such as reduced sound absorption can be obtained by low-pass filtering of the surface shape.

Since diffusers of this type are based on acoustic resonance, the slits are similar to quarter-wave resonators. It follows that there is likely to be considerable sound absorption, due to not only the losses at the bottom and along the well walls, but also because of the high particle velocities immediately above the wells, between one well and another. Another problem in the use of these types of diffusers is the tendency of users to set up arrays of diffusers, leading to coloration because of periodicity in the reflected sound. For these reasons, it is much better to use truly randomized diffusers consisting, for example, of an assortment of spherical scatterers of different diameters than to use arrays of identical hemicylindrical or QRD-type diffusers. As the data in Figure 7.23 shows, resonant diffusers similar to the one in Figure 7.22 are characterized by peaky, uneven sound absorption.

Other types of efficient scattering can be obtained using other depth profiles having low spatial autocorrelation (see Reference 7.9). Primitive root based diffusors for example reject specular reflection.

Free objects in the room will act as "volume" diffusers. Their scattering properties will depend on their impedance and their size relative to the wavelength. Orchestra members form effective scatters and barriers on the stage at medium and high frequencies.

7.10 ABSORPTION DATA EXAMPLES

Indicative examples of typical sound-absorption properties of many materials and constructions may be found in Table 7.1. Table 7.2 shows examples of measured data for the absorption area for individuals (see Reference 7.8). References 7.13 and 7.4 contain much data.

Table 7.1 Examples of typical measured sound-absorption coefficients for various materials and constructions

	Octave band center frequencies					
	125 Hz	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz
Walls						
Concrete block, grout-filled, and unpainted	0.01	0.02	0.03	0.04	0.05	0.07
Concrete block, grout-filled, and painted	0.01	0.01	0.02	0.02	0.02	0.03
Poured concrete, unpainted	0.01	0.02	0.04	0.06	0.08	0.10
Normal, unpainted concrete block	0.36	0.44	0.31	0.29	0.34	0.21
Painted concrete block	0.10	0.05	0.07	0.09	0.08	0.05
6 mm heavy plate glass	0.18	0.06	0.04	0.03	0.02	0.02
2.4 mm regular window glass	0.36	0.25	0.18	0.12	0.07	0.04
12 mm plasterboard, wood studs spaced 0.4 m	0.29	0.10	0.05	0.04	0.12	0.09
Plasterboard, 1.6 × 0.8 panels, 62 mm apart, studs 0.4 m (voids filled with glass fiber)	0.55	0.14	0.08	0.04	0.10	0.10
Same, but panels 128 mm apart	0.28	0.12	0.10	0.07	0.10	0.10
Marble and glazed brick	0.01	0.01	0.01	0.01	0.02	0.02
Painted plaster on concrete	0.01	0.01	0.01	0.03	0.04	0.05
Plaster on concrete block or 25 mm on metal studs	0.12	0.09	0.07	0.05	0.05	0.04
Plaster 16 mm on metal studs	0.14	0.10	0.06	0.05	0.04	0.03
6 mm wood over air space	0.42	0.21	0.10	0.08	0.06	0.06
9–10 mm wood over air space	0.28	0.22	0.17	0.09	0.08	0.06
25 mm wood over air space	0.19	0.14	0.09	0.06	0.06	0.05
Venetian blinds, open	0.06	0.05	0.07	0.15	0.13	0.17
Velour drapes, 280 g/m^2	0.03	0.04	0.11	0.17	0.24	0.35
Fabric curtains, 400 /m^2, hung 1/2 length	0.07	0.31	0.49	0.75	0.70	0.60
Same, but 720 g/m^2	0.14	0.68	0.35	0.83	0.49	0.76
Shredded wood panels, 50 mm, on concrete	0.15	0.26	0.62	0.94	0.64	0.92
12 mm wood, holes 5 mm o.c., 11% open, over void	0.37	0.41	0.63	0.85	0.96	0.92
Same with 50 mm glass fiber in void	0.40	0.90	0.80	0.50	0.40	0.30

Continues

Table 7.1 *Continued*

	Octave band center frequencies					
	125 Hz	**250 Hz**	**500 Hz**	**1 kHz**	**2 kHz**	**4 kHz**
Floors						
Terrazo	0.01	0.01	0.02	0.02	0.02	0.02
Glazed marble	0.01	0.01	0.01	0.01	0.02	0.02
Linoleum, vinyl, or neoprene tile on concrete	0.02	0.03	0.03	0.03	0.03	0.02
Wood on joists	0.10	0.07	0.06	0.06	0.06	0.06
Wood on concrete	0.04	0.07	0.06	0.06	0.06	0.05
Carpet on concrete	0.02	0.06	0.14	0.37	0.60	0.65
Carpet on expanded (sponge) neoprene	0.08	0.24	0.57	0.69	0.71	0.73
Carpet on felt under pad	0.08	0.27	0.39	0.34	0.48	0.63
Indoor–outdoor thin carpet	0.10	0.05	0.10	0.20	0.45	0.65
Ceilings (also check floors and walls for similar material)						
Suspended 12 mm plasterboard	0.29	0.10	0.05	0.04	0.07	0.09
Same with steel suspension system frame	0.15	0.10	0.05	0.04	0.07	0.09
Plaster on lath suspended with framing	0.14	0.10	0.06	0.05	0.04	0.03
Glass fiber acoustic tile suspended	0.76	0.93	0.83	0.99	0.99	0.94
Wood fiber acoustic tile suspended	0.59	0.51	0.53	0.71	0.88	0.74
Acoustic tile, 50 mm on drywall, suspended	0.08	0.29	0.75	0.98	0.93	0.76
Same, air space between tile and drywall	0.38	0.60	0.78	0.80	0.70	0.75
Glass fiber, 360 g/m^2, suspended	0.65	0.71	0.82	0.86	0.76	0.62
Same, but 1120 g/m^2	0.38	0.23	0.17	0.15	0.09	0.06
Luminous ceiling, typical with openings	0.07	0.11	0.20	0.32	0.60	0.85
Hanging absorbers, 450 mm, 450 mm on centers	0.07	0.20	0.40	0.52	0.60	0.67
Same, but 159 mm on centers	0.10	0.29	0.62	0.72	0.93	0.98
Seats and people (area basis, not per person)						
Unoccupied seats, light upholstery, 20 mm	0.36	0.47	0.57	0.62	0.62	0.60
Same, occupied	0.51	0.64	0.75	0.80	0.82	0.83

Unoccupied seats, medium upholstery, 50–100 mm	0.54	0.62	0.68	0.70	0.68	0.66
Same, occupied	0.62	0.72	0.8	0.83	0.84	0.85
Unoccupied seats, heavy upholstery, 200 mm+	0.70	0.76	0.81	0.84	0.84	0.81
Same, occupied	0.72	0.80	0.86	0.89	0.90	0.90
Wood benches and pews, occupied	0.57	0.61	0.75	0.86	0.90	0.86
Same, unoccupied	0.15	0.19	0.22	0.39	0.38	0.30
Students in tablet-arm seats	0.30	0.41	0.49	0.84	0.87	0.84
Special material						
Organ at Boston Symphony Hall, total sabins measured by covering chamber opening with heavy, hard, sound-reflecting material and comparing reverberation times	0.11	0.11	0.15	0.11	0.26	0.41
Organ at Tokyo Opera City, total sabins measured by comparing reverberation times before and after free-standing organ installation	0.65	0.44	0.35	0.33	0.32	0.31
Surfaces outdoors						
Lawn grass, 50 mm high	0.11	0.26	0.60	0.69	0.92	0.99
Fresh snow	0.45	0.75	0.90	0.95	0.95	0.95
Plowed soil	0.15	0.25	0.40	0.55	0.60	0.60
Spruce trees 2.7 meters tall	0.03	0.06	0.11	0.17	0.27	0.31
Water in a swimming pool	0.01	0.01	0.01	0.02	0.02	0.03
Openings						
Open entrances off lobbies and corridors: 0.50–1.00						
Air supply and return grilles: 0.15–0.50						
Stage-house proscenium opening: 0.25–0.75						
Residual-absorption coefficient	0.14	0.12	0.10	0.09	0.08	0.07

Table 7.2 Sound absorption of individuals (From Ref. 7.13)

All values are in metric sabins	Octave band center frequencies					
	125 Hz	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz
Individuals						
Man in suit, standing	0.15	0.25	0.60	0.95	1.15	1.15
Man in suit, sitting	0.15	0.25	0.55	0.80	0.90	0.90
Woman in summer dress, standing	0.05	0.10	0. 25	0.40	0.60	0.75
Woman in summer dress, sitting	0.05	0.10	0.10	0.35	0.45	0.60

7.11 PROBLEMS

7.1 A commercial porous absorber (mineral wool) has been measured using the transfer function method to have a normalized sound field impedance $\underline{Z}_2/Z_0$, as shown in the table, for perpendicular sound incidence. Other measurements using the standing wave method have given the α_r values shown in the table.

f [Hz]	$\underline{Z}_2/Z_0$	α_r
100	3.7-j 8.5	0.17
125	2.8-j 7.0	0.20
160	2.9-j 5.5	0.28
200	3.3-j 4.6	0.33
250	2.6-j 3.3	0.44
315	1.7-j 2.6	0.49

Task: Determine the sound-absorption coefficient from normalized sound field impedance and compare these values to the measured α_r values.

7.2 A perfectly absorbing porous absorber is covered by a thin plastic film having a mass per unit area of 0.18 kg/m^2 that is mounted immediately in front of the absorber. The plastic film does not touch the absorber.

Task: Calculate the resulting sound-absorption coefficient at 1 kHz.

Hint: The total impedance of the foil and absorber is obtained as the sum of their respective impedance components (mass and resistance).

7.3 A porous sound-absorbing panel has the absorption coefficient 0.8 at a frequency of 1 kHz. For hygienic reasons, it is desired to cover the panel with a thin plastic foil. This will result in a reduced sound-absorption coefficient due to sound reflection. The transmission loss R_0 of the foil is:

$$R_0 = 20\log\left|1+\frac{j\omega m''}{2Z_0}\right|$$

Task: Calculate the maximum acceptable mass per unit area of the foil that will not result in an absorption coefficient α less than 0.5. Assume perpendicular sound incidence.

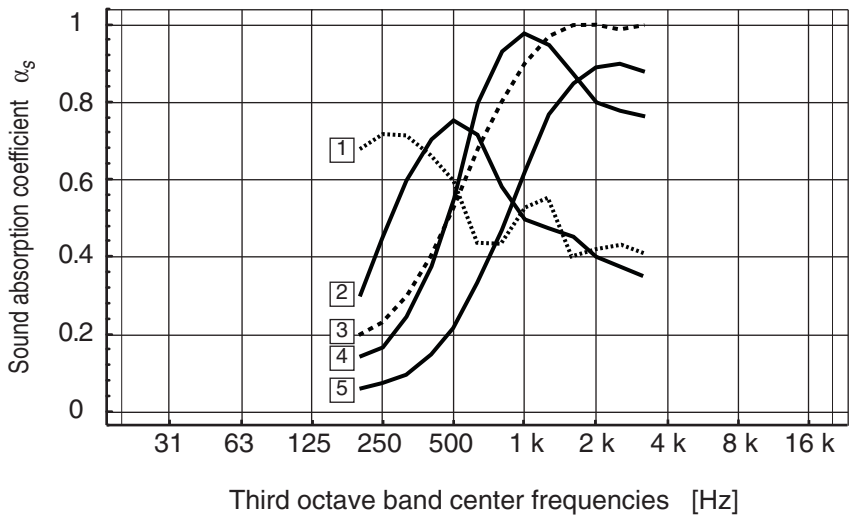
Hint: Only consider the direct reflection and the first transmitted reflection from inside the panel.

7.4 A sound absorber having significant absorption at 100 Hz is desired. The available space (depth) is limited, so the maximum thickness of the absorber must be 0.05 m or less. Materials at your disposal are battens of various thicknesses (2, 5, or 10 cm), 0.009 m gypsum sheets having a mass/unit area of 10 kg/m^2, 0.009 m perforated gypsum sheets with holes of 0.004 m diameter at 0.10 m spacing, and mineral wool sheets having a density of 50 kg/m^3 and various thicknesses.

Task: Which types of absorbers can be designed with these materials? Which one is to be preferred? Explain your choice.

7.5 *Task*: Match the sound-absorption curves in the graph with the absorbers listed. Assume the absorbers to be mounted in front of a hard wall. Explain your answer.

a. 0.050 m thick mineral wool board
b. 0.025 m thick mineral wool board
c. 0.025 m thick mineral wool board immediately behind a perforated tile having a relative hole area of 20%.
d. As in Case c but a relative hole area of 2.7%.
e. As in Case d but with a 0.14 m thick air space between the perforated panel and the mineral wool board that is mounted on the wall.



7.6 On measuring the sound-absorption coefficient for a sound absorber made using air-bricks, sound-absorption maxima were obtained at the following frequencies; 315 Hz, 2.5 kHz and 5 kHz. The brick holes were facing the room, with a 0.05 m air space in front of the wall. Each brick is 0.26 m wide, 0.13 m high and 0.065 m thick, and has 78 holes, each having a diameter of 0.01 m.

Task: Try to find the sound-absorption mechanisms in the construction that can explain the measured sound-absorption coefficient values.

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Waves in Solids 8

8.1 INTRODUCTION

Many acoustic problems involve wave propagation in solid media and structures. Examples of such problems are resonant vibration in loudspeaker diaphragms, sound transmission through walls and cabinets, and mounting of electronics in cars and other vibrating objects. An example of a resonant high frequency vibrational pattern of a poor loudspeaker diaphragm is shown in Figure 8.1.

In contrast to the case of sound propagation in gases, sound propagation in solids can involve many types of waves. Some of the most important and common wave types will be discussed in this chapter. A characteristic of sound transmission in solid structures is conversion between wave types at boundaries and edges. The wave conversion also distributes energy between different modal systems.

Gases oppose volume changes, resulting in longitudinal waves as a response to a disturbance in a gas volume. Solid media not only exhibit elasticity but also shear. This makes it possible for transverse waves and many other types of waves to exist in solid media. Infinite solid media can carry both longitudinal and transverse waves. Finite solid media can also carry many other wave types, particularly *bending*, *transverse*, *torsional*, *quasilongitudinal,* and various types of *surface waves*. Torsional waves, for example, were often used in spring reverberator units such as the one shown in Figure 8.2.

8.2 WAVE TYPES IN INFINITE MEDIA

Longitudinal Waves

The *longitudinal wave* is the basic form of wave motion in many materials. A longitudinal wave in air is characterized by sound pressure and by particle motion in the propagation direction of the wave. Analogously, due to the symmetry, in the case of solids the longitudinal wave is characterized by tension as well as by particle motion in the direction of the wave. The motional pattern for longitudinal waves is shown in Figure 8.3.

The wave equation for one-dimensional longitudinal waves in solids is:

$$\frac{\partial^2}{\partial x^2}\left(\underline{\sigma}_x, \underline{u}_x\right) + k_L^2\left(\underline{\sigma}_x, \underline{u}_x\right) = 0 \tag{8.1}$$

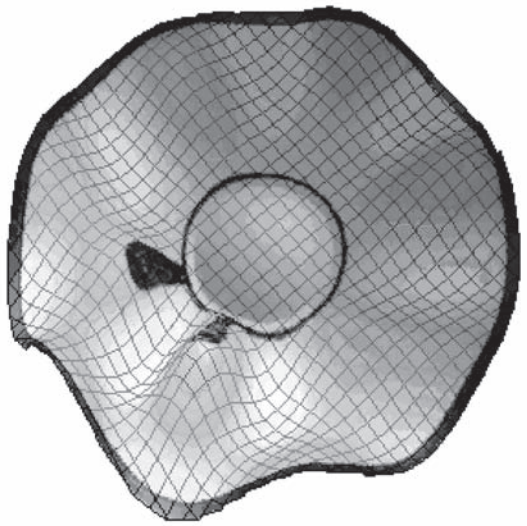


Figure 8.1 Deformation pattern of a conical loudspeaker diaphragm, showing "breakup" at one of the diaphragm's bending wave resonances. Yamaha NS10 loudspeaker paper cone diaphragm at 1050 Hz. (Photo by Mendel Kleiner.)



Figure 8.2 Reverberation units such as this one may use the low propagation velocity of longitudinal and torsional waves in metal spring coils to generate the desired delays and reflection patterns needed to synthesize reverberation for guitar amplifier applications. (Photo by Mendel Kleiner.)

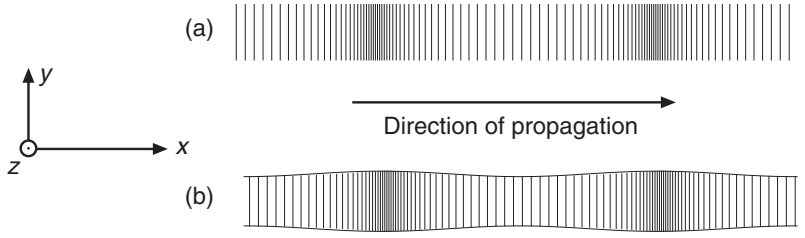


Figure 8.3 An example of the instantaneous deformation pattern (a) of a longitudinal wave, and (b) of a quasilongitudinal wave in a plate.

where $\underline{u}_x$ is the particle velocity in the x-direction, $\underline{\sigma}_x$ the tension in the x-direction, and k_L the wave number ($k_L = \omega/c_L$). The wave propagates with a velocity c_L [m/s], which depends on the elastic constants of the solid:

$$c_L = \sqrt{\frac{D}{\rho}} \tag{8.2}$$

where D is a constant representing the *longitudinal stiffness* of the material (expressed in units of Pascal) and ρ its density.

One notes that the one-dimensional longitudinal wave equation for solids has the same form as the wave equation for gases. This means that all waves propagate with the same speed; such propagation is said to be nondispersive. The longitudinal velocity is the fastest propagation velocity for waves in solids.

Young's modulus, E [Pa], is related to the longitudinal stiffness, D, in the following way:

$$E = D\left(1 - \frac{2\upsilon^2}{1-\upsilon}\right) \tag{8.3}$$

where υ is *Poisson's ratio* for the material. Poisson's ratio is a measure of the cross-contraction of a material as it is elongated. Poisson's ratio is sometimes called the cross-contraction number and must be below 0.5. For many materials, υ is found to be approximately 0.3, resulting in $D \approx 1.35\,E$.

Ideal longitudinal waves can exist only in infinite solids. In reality, because of cross-contraction, one usually finds quasilongitudinal waves, which will be discussed later.

Transverse Waves

The *transverse wave* is characterized by shear forces and particle velocity perpendicular to the propagation direction of the wave. The motional pattern of transverse waves is shown in Figure 8.4.

The wave equation for transverse waves is written in the following way for the case of wave propagation in the x-direction and particle velocity in the y-direction:

$$\frac{\partial^2}{\partial x^2}\left(\underline{\tau}_{xy}, \underline{u}_y\right) + k_T^2\left(\underline{\tau}_{xy}, \underline{u}_y\right) = 0 \tag{8.4}$$

where $\underline{u}_y$ is the particle velocity in the y-direction, $\underline{\tau}_{xy}$ the shear force, and k_T the wave number ($k_T = \omega/c_T$). The wave propagates with a velocity c_T [m/s], which depends on the elastic constants of the solid as:

$$c_T = \sqrt{\frac{G}{\rho}} \tag{8.5}$$

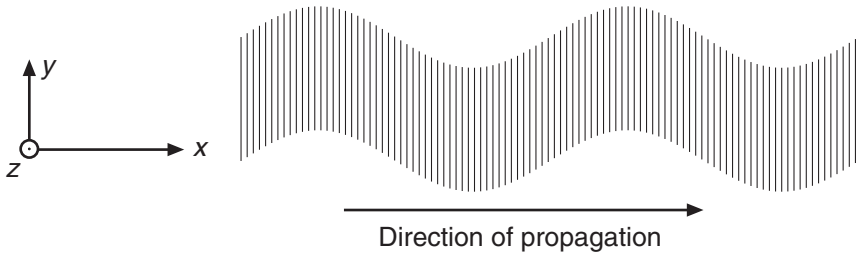


Figure 8.4 An example of the instantaneous deformation pattern of a transverse wave.

where G is the *shear modulus* and ρ the density of the solid. The shear modulus is given by:

$$G = \frac{E}{2(1+\upsilon)} \tag{8.6}$$

Assuming a Poisson's ratio of 0.3, one obtains the ratio between the transverse and longitudinal velocities as $c_T/c_L \approx 0.5$.

Ideal transverse waves can only be found in infinite bodies. In a limited material, such as a large sheet, there is some cross-contraction. This results in quasitransverse waves in the plane of the sheet; such wave motion has limited radiation capability because of the small particle velocity perpendicular to the plane of the sheet.

8.3 WAVE TYPES IN MEDIA OF LIMITED EXTENSION

Quasilongitudinal Waves in Sheets

The cross-contraction phenomenon of solids, characterized by Poisson's ratio, makes it impossible to have ideal longitudinal waves. It is possible though to have nearly longitudinal waves inside a material with an extension that is large compared to the wavelength. Due to the cross-contraction in a thin sheet, the longitudinal wave will be replaced by a quasilongitudinal wave with the motional pattern shown in Figure 8.3.

The quasilongitudinal wave will move at reduced speed because of the cross-contraction. The velocity c_{QL} of a quasilongitudinal wave in a sheet (in the plane of the sheet) is given by an equation similar to Equation 8.1 but with c_L and D replaced by c_{QL} and E_Q where:

$$c_{QL} = \sqrt{\frac{E_Q}{\rho}} \tag{8.7}$$

and

$$E_Q = \frac{E}{1-\upsilon^2} \tag{8.8}$$



Figure 8.5 Loudspeakers using pyramidal, homogeneous plastic foam diaphragms such as this KEF B139 will tend to have their frequency response affected by quasilongitudinal wave resonances in the diaphragm. (Photo by Mendel Kleiner.)

The wave impedance of such a quasilongitudinal wave is given by:

$$Z_{QL} = \sqrt{E_Q \rho} = \rho c_{QL} \tag{8.9}$$

Note also that the quasilongitudinal wave is characterized by poor sound radiation capability since the out-of-plane movement resulting from the cross-contraction is small.

Membrane Vibration

A *membrane* is a thin sheet in which the restoring force due to stiffness is small compared to that by membrane tension. The tension is produced by in-plane forces at the membrane edges. Typical examples of membranes in audio are those of condenser microphones and some electrostatic loudspeakers. Because membranes are often coupled to the air in cavities, the resonance frequencies of their modes will be different from free membranes. For example, this applies to the membrane of condenser microphones (see Reference 8.3). Membrane vibration in electrostatic loudspeakers is usually well damped by the radiation load of air except for the lowest frequency modes, as shown by Figure 14.25.

Membranes are often avoided in loudspeaker engineering as the preferred technology is for nonresonant pistonic motion in transducer radiating surfaces. In microphones, however, the small size, simple manufacturing, and relatively low impedance of membranes are attractive from an engineering viewpoint.

Transverse Waves in Sheets

Two types of *transverse waves* can occur in sheets: *in-plane waves* and *out-of-plane waves*. The sound radiation by in-plane waves is usually not a problem from the viewpoint of noise because the cross-contraction is small.

The out-of-plane waves can be excited using a sound field, vibrator, or other force or pressure generator. Their radiation depends on the mode index. Transverse waves are important in sandwich-type materials where a honeycomb core is bounded by thin foils such as the one shown in Figure 8.6 (see Reference 8.4). The transverse waves in sandwich materials have a velocity that is determined by the sheet materials and the core. The phase velocity of the waves will vary with wave type and frequency as shown in Figure 8.7.

The mode index, sheet geometry, and the speed of the transverse waves will determine the modal pattern on the sheet. If the distance between the antinodes (the points where the amplitude of the standing wave pattern is a maximum) is small compared to wavelength, the radiation from the antinodes will cancel out, as from dipoles and quadrupoles (described in Chapter 2). Each mode will have its own particular radiation pattern and radiation efficiency (see Reference 8.1).

Bending Waves in Sheets

Waves in solids are of particular acoustical interest since, under certain conditions, they may couple well to the surrounding air; that is, the wave motion in the solid can radiate energy easily.

The most important of such waves in sheets is the *bending wave*, in which motion is due to interaction between tension, shear, and mass. Figure 8.8 shows the motional pattern of a bending wave. The bending wave field variables are: rotation, transverse particle velocity, bending moment, and shear forces. The rotation is around the z axis in the figure.

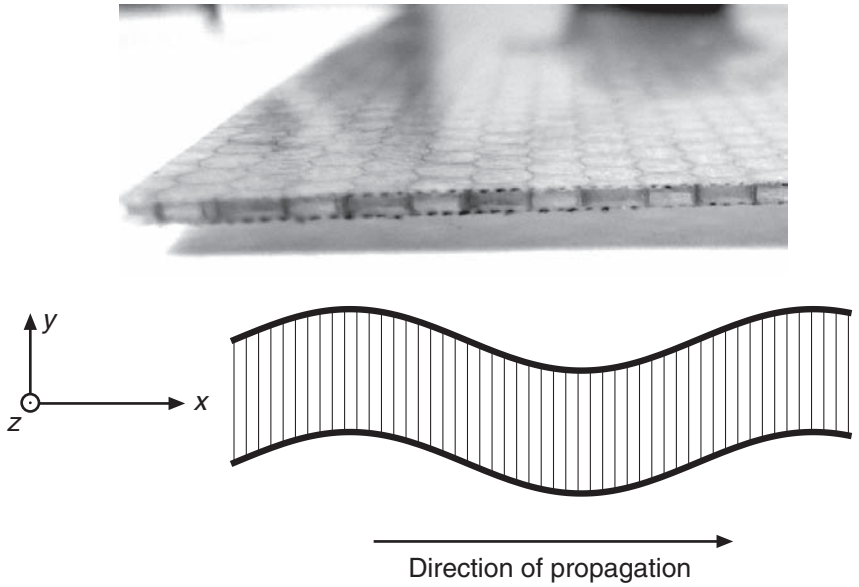


Figure 8.6 An example of a thin sandwich sheet and of an instantaneous deformation pattern of a shear wave in such a sheet. (Photo by Mendel Kleiner.)

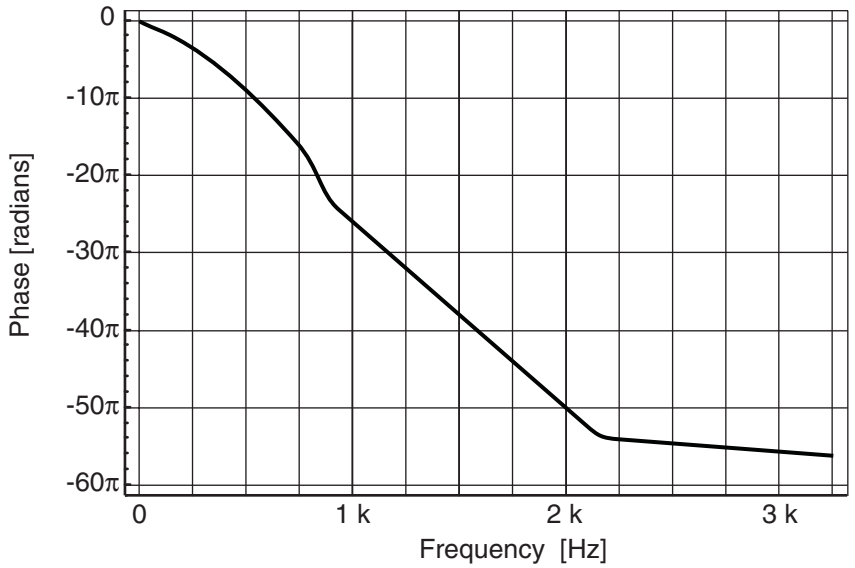


Figure 8.7 Measured phase delay over a 3 m long sandwich beam. Curve shows unfolded phase in radians as a function of frequency (after Ref. 8.5). Three different wave types are present, having different wave propagation speeds.

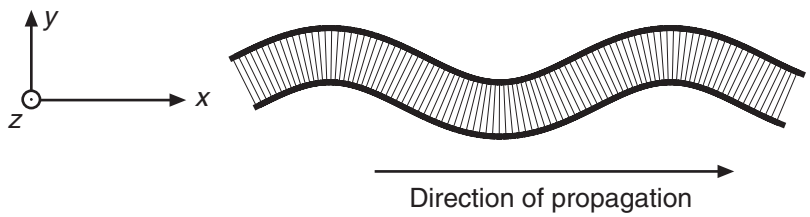


Figure 8.8 An example of an instantaneous deformation pattern of a bending wave in a thin sheet.

Because of the four bending field variables, the bending wave equation will have a somewhat different form from the wave equations studied earlier. Expressed for particle velocity u_y at right angles to the sheet, the equation is written:

$$\frac{\partial^4 \underline{u}_y}{\partial x^4} - \frac{\omega^2 m''}{B'} \underline{u}_y = 0 \tag{8.10}$$

where B' is the *bending stiffness per unit length* [Nm] and m'' the mass per unit area of the sheet. The bending stiffness B' for a thin sheet is given by:

$$B' = \frac{Eh^3}{12\left(1-\upsilon^2\right)} \tag{8.11}$$

where E is the Young's modulus and h the thickness of the sheet.

The sheet can be considered thin as long as its thickness h is much smaller than the bending wave's wavelength λ_B, expressed by the approximate condition:

$$\lambda_B \geq 6h \tag{8.12}$$

One notes that the wave equation of the bending wave is quite different from previously studied wave equations, in that it is a fourth order equation. The more complex wave motion also results in quite different wave propagation velocity behavior. The solution to a wave equation—in this case expressed for particle velocity at right angles to the sheet—is usually assumed to be of the form:

$$\underline{u}_y = Ae^{\pm jk_B x} \tag{8.13}$$

where k_B is the wave number for the bending wave. Inserting this solution into Equation 8.10 gives the wave number as:

$$k_B = \sqrt[4]{\frac{\omega^2 m''}{B'}} \tag{8.14}$$

However, because of the fourth order wave equation, one finds that solutions of the type:

$$\underline{u}_y = Ae^{\pm k_B x} \tag{8.15}$$

are also valid solutions to the bending wave equation. These solutions are the mathematical formulation of nonpropagating waves. One finds such waves close to discontinuities—such as driving points, edges, and so on. Both solutions give the same expression for the wave number.

The wave number for the bending wave is not proportional to frequency but rather to the square root of frequency, which results in frequency-dependent propagation velocity. The bending wave is an example of a *dispersive wave*. Such waves are characterized by different, frequency-dependent *phase* and *group velocities*.

The phase velocity is defined as the velocity by which the instantaneous phase of the wave motion propagates:

$$c_B = \frac{\omega}{k_B} \tag{8.16}$$

The group velocity, however, is defined as the velocity by which the envelope of the wave, which "carries" the energy of the wave, propagates—and is written c_{gB}. The group velocity is given by:

$$c_{gB} = \frac{1}{dk_B / d\omega} \tag{8.17}$$

Using these definitions, one obtains the group and phase velocities for the bending wave in a sheet as:

$$c_B = \frac{\omega}{k_B} = \sqrt[4]{\frac{\omega^2 B'}{m''}} \tag{8.18}$$

The group velocity is:

$$c_{gB} = \frac{1}{dk_B / d\omega} = 2c_B \tag{8.19}$$

Note that both of these velocities are much smaller than the quasilongitudinal wave speed for the sheet.

One should also note that since the velocities increase proportionally to the square root of ω, one cannot use traditional correlation and impulse measurement methods to determine the propagation paths of bending waves.

It is instructive to draw the relationship described by Equation 8.18 in a log-log diagram, as shown in Figure 8.9, together with the line describing the wave number as a function of frequency for sound in air. The bending wave phase velocity is equal to that of sound waves in the surrounding fluid at the so-called *critical frequency*, f_c. The critical frequency is sometimes called the *coincidence frequency*.

A bending wave having a frequency below f_c is called a slow wave, whereas a bending wave having a frequency above f_c is called a fast wave.

Using the condition $c_B = c$, one finds the expression for the critical frequency as:

$$f_c = \frac{c_0^2}{2\pi}\sqrt{\frac{m''}{B'}} \tag{8.20}$$

The bending stiffness is decisive for f_c—the stiffer the sheet, the lower the critical frequency. When using material data, it is convenient to use the *coincidence number* K_c [m/s], defined as:

$$K_c = hf_c \tag{8.21}$$

Table 8.1 shows the coincidence number for some materials that are commonly needed. Typically, the coincidence number is in the 10 to 40 range. It is noteworthy that, for common construction material thicknesses, the critical frequency of the bending waves falls inside the audio frequency range. This also applies to typical loudspeaker and microphone construction materials, such as various plastics and papers.

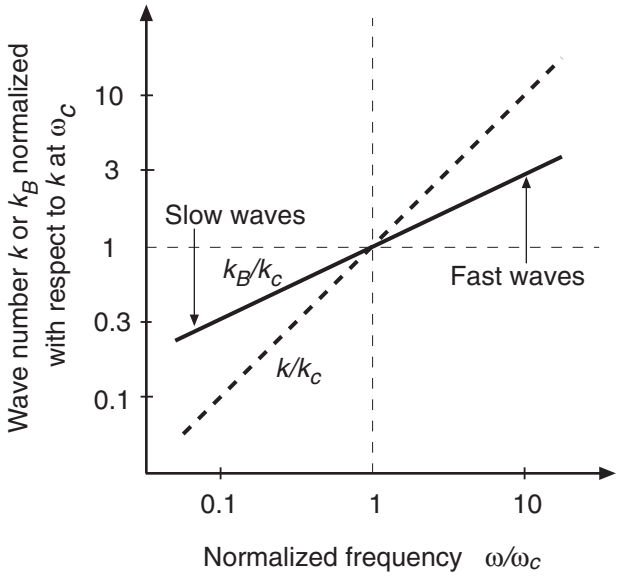


Figure 8.9 Wave number characteristics as functions of frequency. At the critical frequency $\omega = \omega_c$ and the wave numbers are $k = k_B = k_c$. Note: k_B is the wave number for bending waves in a sheet and k is the wave number for sound waves in air.

Bending waves can be excited along a line or at a point on the sheet. Bending waves that propagate at a speed given by the bending wave equation (Equation 8.10) are called *free bending waves*. Free bending waves propagate with the phase velocity, c_B. The wavelength of a free bending wave is consequently determined by the coincidence number of the material and the thickness of the sheet. Such waves are commonly generated by mechanical excitation—for example, on a shop floor. Far away from the point of excitation, circular bending waves will also resemble linear bending waves.

Since the sheet is compliant (because it can bend), sheet motion can be excited by an incident sound wave, as shown in Figure 8.10. This is what commonly happens when a sound wave is incident on a wall in a building. The figure shows the instantaneous extension of a sheet on which a sound wave is incident at an angle φ. The sound wave forces its pressure pattern onto the sheet.

Table 8.1 The coincidence number and the critical frequency for some common construction materials and typical thicknesses.

Material	Coincidence number	Typical thickness	Critical frequency
Concrete	18 m/s	$1.6 \cdot 10^{-1}$ m	0.1 kHz
Lightweight concrete	36 m/s	$7 \cdot 10^{-2}$ m	0.5 kHz
Gypsum	32 m/s	$1 \cdot 10^{-2}$ m	2.5 kHz
Steel	12 m/s	$1 \cdot 10^{-3}$ m	11.7 kHz
Glass	14 m/s	$3 \cdot 10^{-3}$ m	3.5 kHz

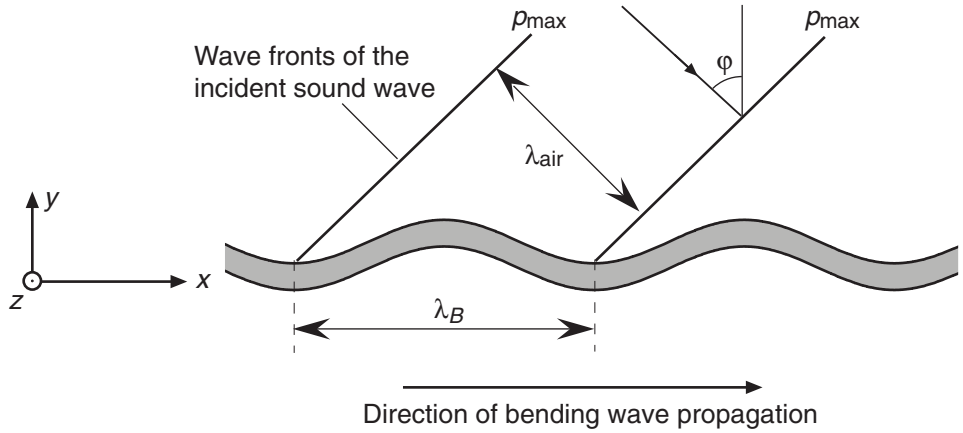


Figure 8.10 The case of sound waves exciting bending waves in a sheet. The wavelength of the bending wave is determined by the condition $\lambda_B \sin(\varphi) = \lambda_{air}$. The sheet radiates waves from the second side (not shown).

The wavelength of such a *forced bending wave* is determined by the angle of incidence and the wavelength of sound in the fluid. When the forced wave impinges on a discontinuity, such as an edge, there will be a wave conversion and free bending waves will be excited together with nonpropagating motion of the type $e^{\pm k_B x}$.

Wave conversion will take place when a wave (bending, quasitransverse, quasilongitudinal wave, or whatever) in a sheet is incident on a discontinuity. A sheet will typically have bending, quasitransverse, and quasilongitudinal waves present at the same time, if there is vibrational energy in the sheet. The vibrational energy is distributed between the modes of the various wave subsystems, which can be regarded as *coupled rooms*. This makes it difficult to measure the quasilongitudinal and quasitransverse waves, since their wave motion (out of the sheet plane) is small compared to that of the bending waves.

Since bending waves are typical for limp sheets, their coupling to waves in air is of considerable interest. The coupling is important from both the viewpoint of reception and radiation. The waves also have a low *mechanical input impedance*. The mechanical input impedance $\underline{Z}_{MB}$, when exciting an infinite sheet at a point, is:

$$\underline{Z}_{MB} = 8\sqrt{m'' B'} \tag{8.22}$$

For a finite sheet, the input impedance will oscillate between mass and compliance behavior, just as the acoustic input impedance into a tube or a room. Because of losses, the mechanical input impedance of a sheet will tend toward the value given by expression 8.22 even for a finite sheet, at least for high frequencies. Taking frequency averages over suitably wide bands will further increase this tendency.

The *modal densities* of the wave fields in a limited sheet will also be affected by the different phase velocities of the wave types. The different wave types will each set up their own modal system. These will be coupled due to edge effects, etc. The modal density is low, however, because of the high propagation velocities, since the waves only propagate in a two-dimensional space. The quasilongitudinal and quasitransverse modal systems will have modal densities that are proportional to frequency, because of the two-dimensional vibrational systems.

The bending wave modal density turns out to be constant, irrespective of frequency, because of the frequency dependent phase velocity of the bending waves. The bending wave resonance frequencies of a rectangular sheet, freely suspended at its edges, are given by:

$$f_{q_x,q_y} = \frac{\pi}{2}\sqrt{\frac{B'}{m''}}\left(\left(\frac{q_x}{l_x}\right)^2 + \left(\frac{q_y}{l_y}\right)^2\right) \tag{8.23}$$

where l_x and l_y are the lengths of the sides of the sheet and q_x and q_y are natural numbers (that is, 0, 1, 2, 3, . . .).

At high frequencies, the average bending wave mode density $\Delta N/\Delta f$ in a sheet having a surface area S is given by:

$$\frac{\Delta N}{\Delta f} \approx \frac{S}{2}\sqrt{\frac{m''}{B'}} \tag{8.24}$$

A simple calculation using the data in Table 8.1 shows that a 0.15 m thick and 20 m^2 large concrete slab has a distance between modes of approximately 15 Hz. In comparison, a 0.001 m thick and 2 m^2 large steel sheet—similar to the one used for the reverberator shown in Figure 8.11—has a distance between modes of approximately 6 Hz.

Good averaging over modes in acoustic and vibration measurement requires diffuse sound fields, typically at least 10 modes present in the frequency range of interest. For building acoustics measurements (125 Hz to 4 kHz) that typically are accomplished using octave band frequency resolution, the modal density will usually be satisfactory except in the lowest octave bands.

The modal density also influences the way that the reverberation in a sheet sounds. This gave the reverberation sheets that were used for simulation of room reverberation, their special sound quality. The impulse response of such plates are sometimes implemented in modern digital signal processing units and software.

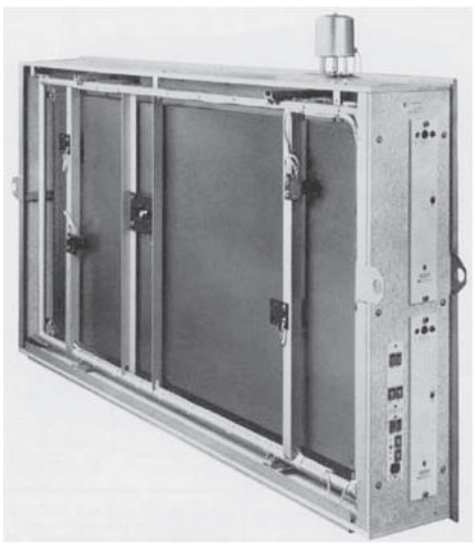


Figure 8.11 The discontinued EMT-140 plate reverberation unit was an advanced example of the use of the slow bending waves for simulation of room reverberation. The later EMT model 240 used gold foil to increase the mode density (photo courtesy of EMT Studiotechnik GmbH, Germany).

8.4 STRUCTURE-BORNE SOUND IN LOSSY SHEETS

All real sound propagation is affected by losses. In the case of sound propagation in solids, one usually regards the radiation of vibrational energy to the surroundings as energy loss. As in the case of sound waves in air, it is convenient to take the losses into account in calculating the propagation by including a complex wave number. Whereas sound wave losses are introduced *discretely* by sound absorbers at walls—internal losses being small—this is not practical in the case of wave motion in solids.

The approach is similar to that used when handling losses in the case of sound propagation in air in rooms. The wave motion in sheets is damped using some form of continuous damping along the surface or in a layer inside the sheet—similar to the wall sound absorption and the damping by air in the case of sound in rooms.

In the case of sound propagation in sheets, the losses are usually expressed by using a damping coefficient instead of the equivalent of the absorption area used in room acoustics. The loss factor is included by making the mechanical constants complex, for example, the Young's modulus.

The phase difference between extension $\underline{\varepsilon}$ and tension $\underline{\sigma}$ along a sheet is given by Hooke's law:

$$\underline{\sigma} = \underline{E}\underline{\varepsilon} \tag{8.25}$$

where $\underline{E}$ is the complex modulus of elasticity given by:

$$\underline{E} = E_0\left(1 + j\eta\right) \tag{8.26}$$

The loss factor η is typically in the range from 10^{-3} to 10^{-2} for many construction materials but can be increased to approximate unity by applying some kind of treatment to the sheet absorbing the energy of the wave fields.

Since the propagation velocities of quasilongitudinal and quasitransverse waves, c_L and c_T, are proportional to the square root of E, one finds:

$$\underline{c}^2 = c^2\left(1 + j\eta\right) \tag{8.27}$$

with c either being c_L or c_T. This gives the respective wave number as:

$$\underline{k}^2 = \frac{\omega^2}{\underline{c}^2} = k^2\left(1 - j\eta\right) \tag{8.28}$$

Provided that the loss factor η is much smaller than unity, one obtains the complex wave number for quasilongitudinal and quasitransverse waves as:

$$\underline{k} \approx k\left(1 - j\frac{\eta}{2}\right) \tag{8.29}$$

for $\eta \ll 1$. In analogy, since c_B is proportional to $E^{1/4}$, one finds the complex wave number for bending waves in sheets as:

$$\underline{k}_B = k_B\left(1 + j\eta\right)^{-1/4} \approx k_B\left(1 - j\frac{\eta}{4}\right) \tag{8.30}$$

The imaginary part of the wave number describes how quickly the wave decays along the propagation path. For quasilongitudinal and quasitransverse waves, one finds that the particle velocity, at right angles to the plane of the sheet, diminishes as:

$$\tilde{u}_y \propto e^{-k\eta x/2} = e^{-\pi\eta x/\lambda} \tag{8.31}$$

It is often practical to express the extra attenuation (on top of the geometrical attenuation) as a level difference per unit length, just as in the case of sound in air.

For quasilongitudinal and quasitransverse waves, the damping is given by:

$$\Delta L_{T,L} \approx \frac{27.2\eta}{\lambda_{T,L}} \quad \left[\mathrm{dB/m}\right] \tag{8.32}$$

For free bending waves the damping is given by:

$$\Delta L_B \approx \frac{13.6\eta}{\lambda_B} \quad \left[\mathrm{dB/m}\right] \tag{8.33}$$

Since bending waves typically have much shorter wavelengths than quasilongitudinal and quasitransverse waves, their damping is generally much larger per unit length.

Using various forms of damping layers, one can usually increase the losses considerably. Extra losses can also be added by air pumping, dry friction, and magnetic hysteresis. In an otherwise *undamped* sheet, such losses will be the dominant loss mechanisms.

8.5 DAMPING BY VISCOELASTIC LAYERS

As previously mentioned, it is often desirable to damp waves in sheets—metal sheets and other sheets, such as bending waves in building structures as well as the boxes and diaphragms of loudspeakers and other electroacoustic transducers—to reduce sound radiation and vibration propagation.

A simple way to increase the damping of bending waves is to add a layer of a *viscoelastic material* to the sheet, either externally or internally. Another way is to couple the sheet to another structure that has high damping.

A viscoelastic layer exhibits both viscosity and elasticity. Polymers have long molecular chains and allow many combinations of elastic and viscous properties through choice of chain lengths and forks. Chewing gum is a typical example of a viscoelastic material—other examples are most polymers, such as rubber and plastics. Usually the viscous properties are extremely temperature-dependent. The viscosity leads to energy conversion from vibration to heat.

The success of the application of viscoelastic damping depends on several factors: the temperature, the type of excitation, the wave types involved, the coupling to other structures, and the frequency of vibration (since this determines the wavelengths of the different waves). The addition of a viscoelastic layer also changes the mass per unit area and the bending stiffness per unit length. Consequently, it is not sufficient just to study the loss factor.

It is advantageous to use the viscoelastic layers as close to the source of vibration as possible because the application is only needed over a smaller area and because the bending waves usually dominate the wave type spectrum close to the source.

Finally, it must be emphasized that increasing the loss factor primarily affects the resonant vibration. The forced vibrations are usually not affected, except by the addition of mass or stiffness due to the layer.

Viscoelastic Materials

A feature of viscoelastic materials is that the loss factor is much higher, and the shear modulus much lower, than for typical construction materials. Most viscoelastic materials exhibit large temperature and frequency dependence in the Young's modulus, the shear modulus, and loss factor. An example of typical behavior is shown in Figure 8.12.

The shear modulus of a viscoelastic material has a real part $G_{ve:re}$ and an imaginary part $G_{ve:im}$. The parts have different frequency dependencies. The imaginary part of the shear modulus increases quite rapidly in a specific range as temperature is reduced or frequency increased. A rule of thumb is that a change in temperature from 5 to 7°C corresponds to a change in frequency by a factor of 10. The damping is fairly broadband.

Viscoelastic materials are generally not suited as construction materials. Because of their cost, they are typically used in thin layers, either on or in the sheet—possibly together with various distance layers.

Free Layers

By attaching one or more viscoelastic layers to the surface, or surfaces, of a sheet, one obtains both extension and bending in the *free layer*. Generally, one obtains the best use of the material if most of the total elastic energy is stored in the viscoelastic layer.

The loss factor for bending waves in a sheet with a single-sided free, soft viscoelastic layer is given by:

$$\eta_B = \eta_{ve} \frac{h_{ve} h_n^2 \,\mathrm{Re}\left[\underline{E}_{ve}\right]}{B'} + \eta_{pl} \tag{8.34}$$

where η_B is the combined total loss factor, η_{ve} loss factor of the viscoelastic layer, η_{pl} loss factor of the sheet, $\mathrm{Re}[\underline{E}_{ve}]$ real part of the complex Young's modulus of the viscoelastic layer, h_{pl} thickness of the sheet, h_{ve} thickness of the viscoelastic layer, h_n distance between the neutral planes of the two layers. For soft layers $h_n = (h_{ve} + h_{pl})/2$, and B' bending stiffness of the combined construction.

The equation shows that the viscoelastic layer should be thick and have as high a Young's modulus as possible. Often Young's modulus for viscoelastic layers is approximately 100 times smaller than that of the sheet on which it is applied. Free viscoelastic layers are therefore primarily used to damp thin sheets and sheet metal. One also observes that the loss factor of the combination has approximately the same frequency dependence as that of the viscoelastic layer.

Application of a free viscoelastic layer is often used as an after-treatment to resonant sheets. An example of such treatment to a loudspeaker diaphragm is shown in Figure 8.13. While free layers can be used in this way, their use is at a disadvantage because of the mass of the viscoelastic layer that needs to be thick. This limits the use of free viscoelastic layers for loudspeaker diaphragm applications since a loudspeaker's diaphragm mass needs to be small.

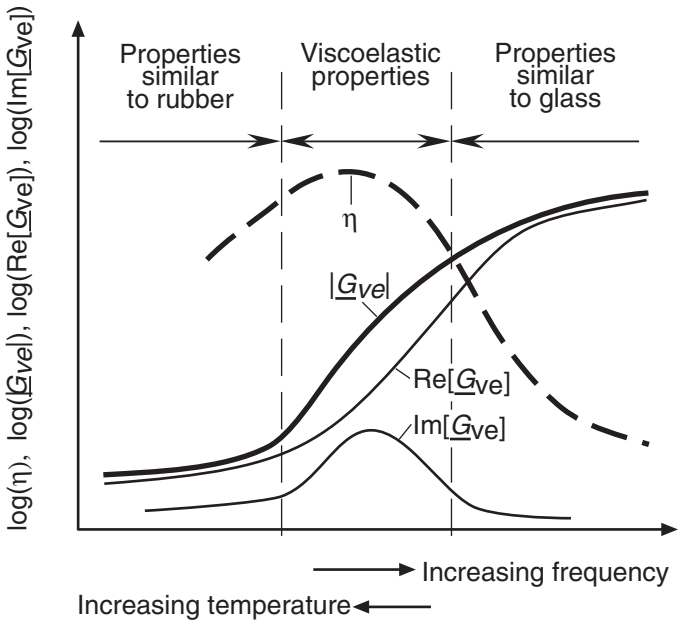


Figure 8.12 A graph of the typical properties of a viscoelastic material as a function of frequency and temperature.



Figure 8.13 Early loudspeakers having a diaphragm made of plastic material, such as this KEF B110, needed to have their diaphragm surfaces treated with viscoelastic compounds to damp the bending wave resonances since the plastics used had small loss factors. (Photo by Mendel Kleiner.)

Is there any advantage to using a free layer on both sides of a sheet? If the layers are thick, it does not matter if the layer is added by one-half to each side. If the layer is thin, it will contribute better to the combined loss factor if it is applied to one side only.

To increase the losses, one can add a flexible structure known as a *spacing layer*. An example is a honeycomb structure between the sheet and the viscoelastic layer. This layer should not store any energy ideally and should be very rigid to shear forces.

Constrained Layers

A *constrained viscoelastic layer* is sandwiched between two sheets so that it is in contact with both sheets. Its damping is achieved by its resistance to shear forces between the two attached sheets (see Reference 8.1). The bending wave loss factor for such a sandwich construction can be estimated using the diagram in Figure 8.13 where:

$$\eta_B = \frac{\eta_{ve} XY}{1+(2+Y)X+(1+Y)\left(1+\eta_{ve}^2\right)X^2} \tag{8.35}$$

with

$$X = \frac{\mathrm{Re}\left[\underline{G}_{ve}\right]}{k_p^2 h_{ve}}\left[\frac{1}{E_{p,1}h_{p,1}}+\frac{1}{E_{p,2}h_{p,2}}\right] \tag{8.36}$$

and

$$\frac{1}{Y} = \frac{E_{p,1}h_{p,1}^3 + E_{p,2}h_{p,2}^3}{12h_n}\left[\frac{1}{E_{p,1}h_{p,1}}+\frac{1}{E_{p,2}h_{p,2}}\right] \tag{8.37}$$

One usually denotes X as the shear parameter and Y as the stiffness or geometric parameter.

If both sheets are of the same material and have the same thickness, the Y parameter will depend only on the geometry. In Equations 8.35–8.37, the following symbols are used:

η_B	loss factor of the sandwich construction
η_{ve}	loss factor of the viscoelastic layer
$\mathrm{Re}[\underline{G}_{ve}]$	real part of the complex shear modulus of the viscoelastic layer
$E_{p,1}$; $E_{p,2}$	Young's modulus of sheet 1 and 2 respectively
$h_{p,1}$; $h_{p,2}$	thickness of sheet 1 and 2 respectively
h_{ve}	thickness of the viscoelastic layer
h_n	distance between the neutral planes of the two layers; $h_n = h_n + (h_{p,1} + h_{p,2})/2$
k_p	wave number of the sandwich panel at the frequency of interest
λ	bending wave wavelength for sandwich panel at the frequency of interest

Equation 8.37 shows that the shear parameter X is proportional to the square of the wavelength of the bending wave. The curve for the loss factor will move upward in frequency when the viscoelastic layer's thickness is reduced and when the real part of the shear modulus $\mathrm{Re}[\underline{G}_{ve}]$ increases, as shown in Figure 8.14.

Note that the loss factor curves show wide bandwidths, and that one can have considerable damping even with a thin viscoelastic layer. These are the main advantages of the constrained layer technique over that of free layers.

If both sheets have the same thickness and are of the same material, the maximum combined loss factor is approximately one-third of the loss factor of the viscoelastic layer. It is not necessary to add the viscoelastic layer to the entire sheet. One can use partial treatment by adding the viscoelastic material in strips or patches. The optimum working frequency will then be adjusted by a factor proportional to the ratio of the untreated and treated surface area. This applies also to free viscoelastic layers. One must note though that it is possible to create new modal systems in this way.

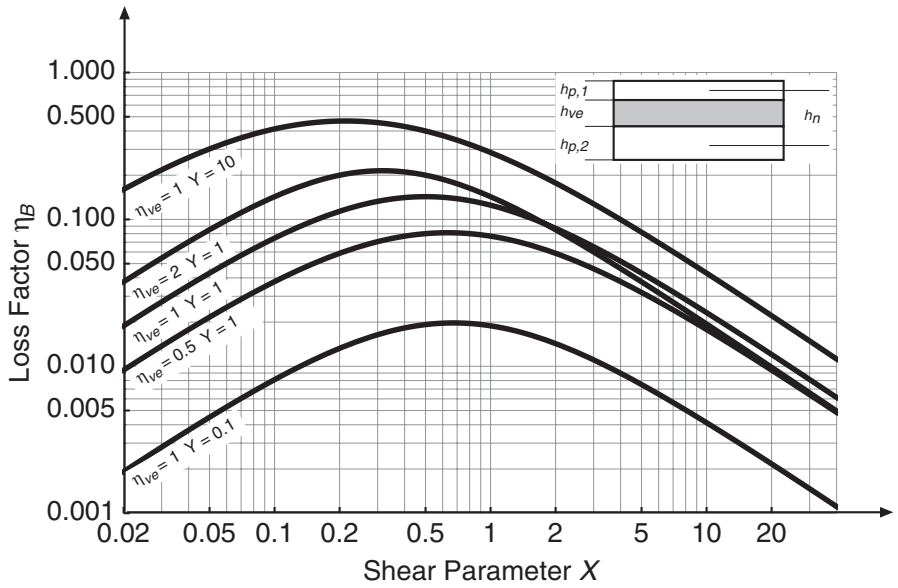


Figure 8.14 The resulting loss factor η_B for a sheet having a constrained viscoelastic layer as a function of the shear parameter X for some values of the geometry/stiffness parameter Y and layer loss factor η_{VE} (adapted from Ref. 8.1).

Figure 8.15 shows a loudspeaker diaphragm that is provided with a system of viscoelastic wedges to absorb radially propagating bending waves in the diaphragm, analogous to the way sound-absorbing wedges are used in anechoic chambers. Constrained layers are ideal for loudspeaker diaphragm applications since the added mass of the viscoelastic layer can be made small.

8.6 DAMPING BY SAND AND OTHER LOSSY MATERIALS

Loudspeaker cabinets and walls can often be well-damped using sand, gravel, and similar materials. Adding damping, by a damping compound such as damping glue, to a sheet does not necessarily lead to less sound radiation by bending wave vibration in the sheet, as discussed in Chapter 9. Because of the vibration generation mechanisms, however, it may often be advantageous to add mass to a structure affected by vibration generating forces. The added mass will increase the input impedance of the structure considerably without any acoustical adverse effects, since the sand has such high damping that the resonances will generally be unimportant. Heavy bracing of the structure, on the other hand, may lead to increased input impedance, but the structure's resonances will remain and primarily only be shifted upward in frequency. Figure 8.16 shows a loudspeaker cabinet where the outside walls are dual-sheet panels resulting in airspace that can be filled with sand. This kind of construction is also used in other applications where high sound insulation over a wide frequency range is desired, such as in concert hall walls.

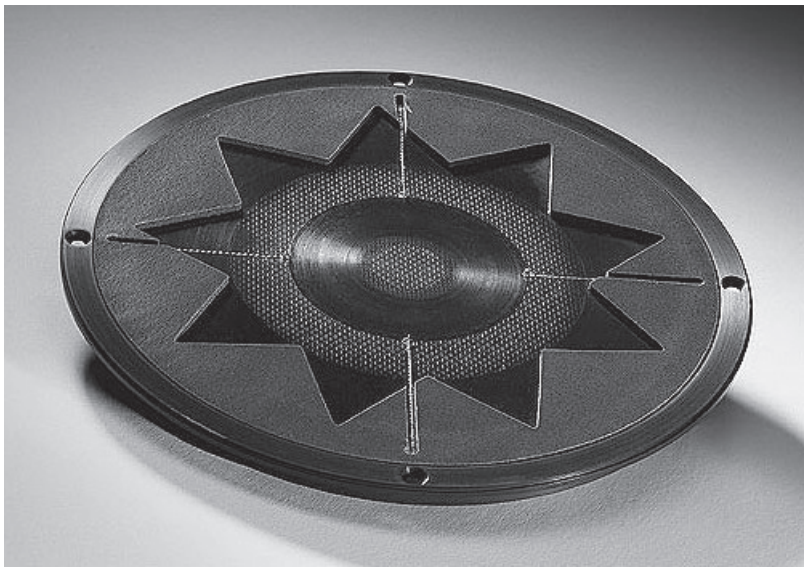


Figure 8.15 This Manager MSW loudspeaker, uses damped bending waves excited from an electrodynamically excited rod attached to the center of the spherical diaphragm made of a rubber-like material. Note the star-shaped damping arrangement intended to act as an anechoic termination for the bending waves expanding from the center. The intention is to prevent radial bending wave resonances (photo courtesy of Manger Products, Germany).

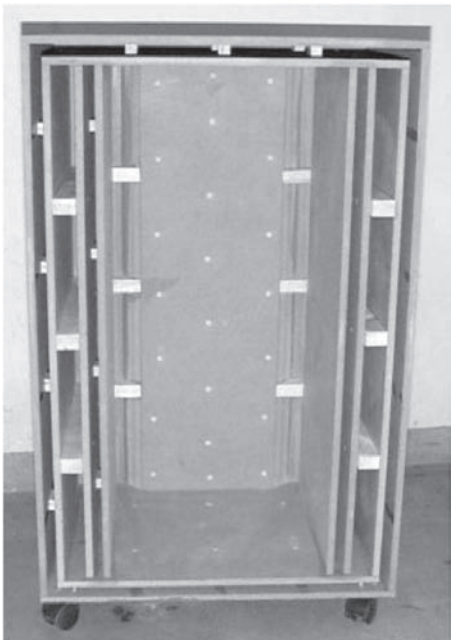


Figure 8.16 A loudspeaker enclosure featuring outer walls with cavities to be filled with sand (photo by Claudio Bonavolta).

8.7 PROBLEMS

8.1 A certain steel sheet has thickness $h = 0.001$ m, area $S = 2$ m^2, modulus of elasticity $E = 2 \cdot 10^{11}$ N/m, density $\rho = 7{,}800$ kg/m^3, and Poisson's ratio $\upsilon = 0.3$.

Task: Calculate the number of modes in the third octave bands 100 Hz and 800 Hz respectively for the freely supported steel sheet and draw the modal density as function of frequency.

8.2 A vibration-sensitive apparatus for the manufacture of integrated circuits is to be mounted on a wall. The wall consists of 0.07 m thick lightweight concrete having density $\rho = 500$ kg/m^3, critical number 38 m/s, and loss factor $\eta = 0.02$. Due to other machines in the room, there is a sound pressure level of 70 dB in the 1 kHz octave band. The apparatus is particularly sensitive to vibration in the 1 kHz octave band, and the maximum acceleration on its supports is 0.1 m/s^2 rms at l kHz.

Task: Calculate the resulting acceleration of the wall due to the sound field.

Hint: The sound power W_{abs} absorbed by the wall is:

$$W_{abs} = \frac{S\alpha_S \left\langle \tilde{p}^2 \right\rangle}{4Z_0}$$

where S is the surface area of the wall, and $\langle p^2 \rangle$ is the spatially averaged squared rms sound pressure over the wall. The absorption coefficient of the wall can be written:

$$\alpha_S = \frac{2\pi Z_0}{\omega m''} s \frac{f_c}{f}$$

where m'' is mass per unit area, s is the radiation factor (see Chapter 9), and f_c is the critical frequency.

The spatially averaged mean square vibration velocity $\langle \tilde{u}^2 \rangle$ of the wall is obtained from:

$$\left\langle \tilde{u}^2 \right\rangle = W_{abs} \frac{1}{\omega \eta S m''}$$

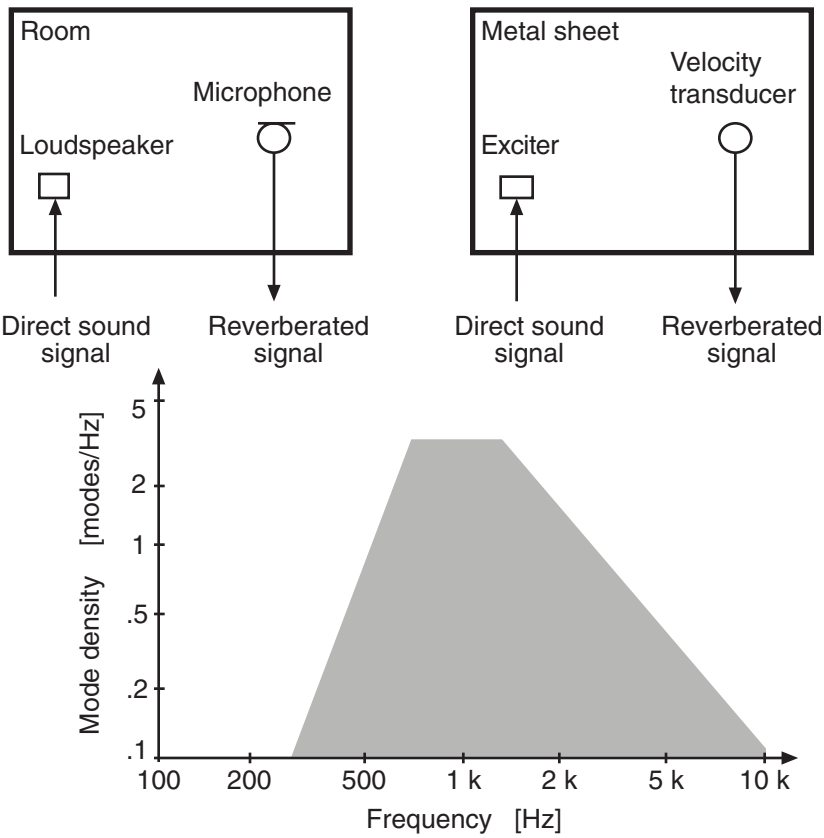
where η is the loss factor of the wall.

8.3 Using a room or a sheet to act as a reverberator are two separate ways of obtaining *artificial reverberation*. In the case of the sheet, a bending wave field is set up. The resulting sound (room) or vibration (sheet) is sensed, amplified, and added to speech or music as artificial reverberation. If the respective modal density is too low, the reverberation is perceived as *colored* and cannot be used. The shaded area in the diagram on page 216 shows in which modal density range the reverberation is perceived as colored.

Task: Determine which one of the possible alternatives—*a*, *b*, or *c*—can be used satisfactorily.

a. A room of volume $3 \times 4 \times 5$ m^3.
b. A steel sheet having thickness $h = 0.5$ mm and surface dimensions $= 1.2 \times 1.8$ m^2.
c. A gold foil having thickness $h = 20$ μm and surface dimensions $= 0.3 \times 0.4$ m^2.

Material	ρ [kg/m^3]	E [N/m^2]	ν
Steel	$7.7 \cdot 10^3$	$20 \cdot 10^{11}$	0.3
Gold	$19.0 \cdot 10^3$	$8 \cdot 10^{10}$	0.3



Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Sound Radiation and Generation

9

9.1 INTRODUCTION

Common to both sound radiation and sound generation is wave motion caused by disturbing the gaseous medium. The disturbance can have many causes and take many forms as shown by Figure 9.1.

The most common type of technical sound source is probably *the vibrating surface*, for example, the vibrating membrane or sheet, such as metal sheets on machine covers or parts, floors, walls, and ceilings in buildings, loudspeaker box walls, and loudspeaker diaphragms. For many types of plates, bending waves are the most important form of vibration from the viewpoint of sound radiation.

Sound can be generated in gases and other fluids by several mechanisms such as *pulsating flow*, *instability*, *turbulence*, and in fluids also by *cavitation*.

Heating of the medium by ionization is yet another type of sound source occurring in *plasma loudspeakers* (see Figure 9.2), *lightning*, *sparks*, and *random electric discharge* (for example, around high-tension power lines).

This chapter focuses on two types of sound generation—sound from surfaces carrying bending waves and sound by turbulence in gases. The important form of sound radiation by piston-like diaphragms, such as ideal loudspeaker diaphragms, is the topic of Chapter 14.

The measurable quantities for sound radiation are the total radiated sound power, intensity, sound pressure, particle vibration, surface vibration velocity, area, and geometry of the radiating surface. Using these quantities, one can define various metrics to describe the *coupling* between the vibration of the radiating surface and the power radiated to the surrounding medium.

9.2 COMMON METRICS FOR SOUND RADIATION

Some common metrics to describe the sound radiation properties of panels and membranes are *radiation resistance, radiation ratio*, and *radiation loss factor*. In analogy with electrical circuits, one can regard the radiation of acoustic power from an acoustic system as a loss in a resistive component. This is often a convenient approach when dealing with electroacoustic systems. In building acoustics, however, the radiation ratio and the radiation loss factor are typically used to calculate the power radiated by vibrating panels.

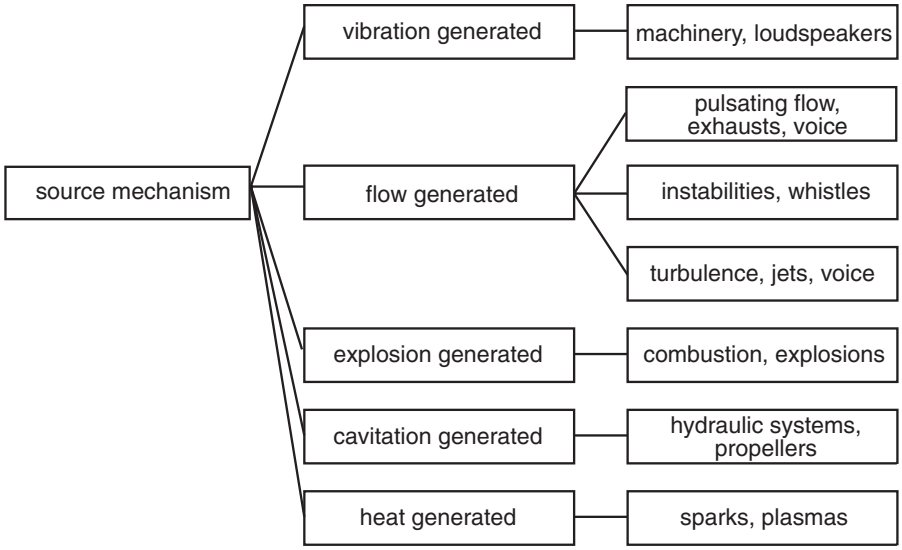


Figure 9.1 A schematic classification of various sound generation mechanisms.



Figure 9.2 A modern plasma loudspeaker using *hot* plasma generated by a high-frequency electric field (photo courtesy of Ulrich Haumann).

Radiation Resistance

The *radiation resistance*—the real part of the radiation impedance—can be defined for both acoustic and mechanic systems. The *mechanical radiation resistance* R_{MR} is related to the acoustical radiated power W_{rad} as:

$$W_{\text{rad}} = \tilde{u}^2 R_{MR} \tag{9.1}$$

where $\tilde{u}$ is the root-mean-square value of the vibration velocity, assumed to be perpendicular to the surface and equal over the entire area.

The complex *mechanical radiation impedance*:

$$\underline{Z}_{MR} = R_{MR} + jX_{MR} \tag{9.2}$$

is often used when working with electroacoustical and electromechanical analogies to characterize the radiation in a way analogous to that used for electrical circuits. The radiation impedance has real and imaginary parts—the real part represents the sound radiation while the imaginary part represents the stored acoustic energy usually in the near field of the radiator.

Radiation impedance is typically used when studying sound radiation by plane surfaces that move with the same velocity and phase over their entire sound radiating area—although, the theory is generally also applied to the various types of diaphragms and membranes used for loudspeakers and microphones that tend to have conical and hemispherical shapes. As long as the depth of the cone or dome is much smaller than the wavelength under study, the theory for planar surfaces holds reasonably well. The radiation impedance is usually considered for each side separately. Horns (such as those shown in Figure 9.3) are often used to increase the radiation impedance seen by the loudspeaker's diaphragm.

Radiation Ratio

The *radiation ratio* was addressed in Chapter 8 and is defined as:

$$s = \frac{W_{\text{rad}}}{Z_0 S \left\langle \tilde{u}^2 \right\rangle} \tag{9.3}$$

where W_{rad} is the radiated power by *one side* of the surface, S the radiating surface area, and $\langle \tilde{u}^2 \rangle$ the mean square of the vibration velocity in the normal of the surface, averaged over the surface.



Figure 9.3 Horns are sometimes used to increase the radiation impedance sensed by the loudspeaker diaphragm (photo courtesy of Avantgarde Acoustic GmbH, Germany).

A radiation ratio of unity corresponds to the radiation that will be obtained from one side of an infinitely large plane piston vibrating in the direction of its normal, equally at all points. The radiation ratio is used extensively to characterize the radiation ability of sheets carrying bending waves.

The radiation ratio varies due to a number of factors. If one has some knowledge about the sound radiation characteristics of a panel, for example, by prior measurement data, one can estimate the sound power radiated by a surface by measurement of its vibration. In using this approach, one can determine the sound radiation of various surfaces of a vibrating object—for example, the walls of a room. The radiation ratio, consequently, is an important number. A library of such data is useful to the noise and vibration consultant. Radiation ratio data are usually shown in logarithmic form as $10 \log(s)$.

Radiation Loss Factor

The sound *radiation loss factor*, η_{rad}, is sometimes used to characterize the effect of its sound radiation on the damping of a vibrating surface. This loss factor is defined in analogy to the definition of η in Chapter 2 as:

$$\eta_{rad} = \frac{W_{rad}}{E_k \omega} \tag{9.4}$$

where E_k is the mean of the total kinetic energy carried by the surface; ω, the angular frequency; and W_{rad}, the total power lost by sound radiation.

It is often difficult, by measurement, to separate the losses due to radiation from those caused by edge losses and internal damping.

9.3 SOUND RADIATION BY VIBRATING SURFACES

Sound radiation can be studied using analytical and numerical techniques. While analytical techniques are physically instructive, numerical techniques have the advantage of allowing the sound radiation of nearly any surface structure or vibration pattern and surrounding medium to be analyzed. In this book, we will only use analytical techniques.

Piston radiation and bending wave sound radiation are the two most interesting types of radiation, but other sound radiation—such as that by pistons coupled to horns and radiation by membranes—is also of great practical interest.

9.4 SOUND RADIATION BY VIBRATING PISTONS AND MEMBRANES

We will first study the sound radiation for the case of a monopole situated in free space and on a hard surface. Using Equations 1.24 and 1.25, we find that for a small free sound source having the volume velocity of $\underline{U}_D$, the sound pressure at a distance r will be:

$$\underline{p}(r) = j\omega 4\pi a^2 \underline{u} \rho_0 \frac{e^{-jkr}}{4\pi r} = j\omega \underline{U}_D \rho_0 \frac{e^{-jkr}}{4\pi r} \tag{9.5}$$

When the monopole is on top of a hard surface, the resulting sound at the observation point will be the linear sum of the sound from the monopole and its mirror image, which is also a monopole.

Since the monopoles are assumed infinitely small, they will acoustically be at nearly the same position. The resulting sound pressure will then be twice that of one of the monopoles in free space.

A sound radiating piston or membrane must have a velocity component normal to its surface. To simplify the mathematics and physics, the piston in our case is assumed to be *mounted in an infinite baffle*. The vibrating piston can be considered as a surface covered by small monopoles, where each monopole represents the volume velocity generated by a small patch of the piston surface. When the piston is vibrating, each element on the piston surface area (on both the front and back sides of the piston) can be thought of as an individual, in-phase monopole, generating volume velocity. Because the baffle is assumed, infinite radiation from each side of the piston can be considered independently.

Because of linearity, we can again use the principle of superposition of pressure to determine total sound pressure at the observation point.

Each monopole will contribute to the total sound pressure at a distance as if it has a part of the total volume velocity $d\underline{U}_D = \underline{u}_D dS$, due to the vibrational velocity of the diaphragm $\underline{u}_D$ and the size of the surface element of each surface element dS. The total pressure at the point of observation at a distance r can then be obtained by summing the sound pressure contributions $d\underline{p}$ from each surface element dS of the entire piston surface:

$$\underline{p}(r) = j\omega\rho_0 \int_S \frac{\underline{u}_D e^{-jkr'}}{2\pi r'} dS \tag{9.6}$$

In Chapter 14, we will use this technique to study the sound radiation from circular loudspeaker diaphragms.

If the piston is not moving in phase over its area or if the piston vibration velocity changes over its area, the same technique to find the sound pressure can still be used with appropriate summation of the various sound pressure contributions from the different piston patches.

It must be noted that we have assumed the volume velocity of the diaphragm to be unaffected by the impedance of the surrounding medium. While this is often the case for massive loudspeaker diaphragms, such as of those of electrodynamic loudspeakers, the vibrations of the thin low-mass membranes, for example electrostatic loudspeakers, are considerably damped by the surrounding air.

The radiated acoustic power can be estimated by integration of the sound intensity for the far field of the piston or membrane, where particle velocity and sound pressure are in phase. One prerequisite for far field conditions for a pistonic source is that the sum pressure diminishes with distance as $1/r$.

9.5 SOUND RADIATION BY VIBRATING SHEETS CARRYING BENDING WAVES

When studying the sound radiation of a wall or other large structure, it is best considered as a sheet having a vibration velocity that varies over the sheet. The sound radiation depends, along with the vibratory characteristics of the structure-borne sound, on the acoustic properties of the surrounding medium at

the specific location, the sheet geometry, and other factors. For sound radiation, the sheet must have a vibration component perpendicular to its surface.

Bending Waves in an Infinite Sheet

Assume an infinite sheet carrying a single frequency bending wave with a linear wave front and constant amplitude. One of the boundary conditions at the surface of the sheet requires that the phase velocity of the induced sound field along the sheet be the same as that of the vibration of the sheet. Another boundary condition requires that the particle velocity component of the induced sound field at right angles to the sheet be the same as the particle velocity of the sheet at right angles to the sheet.

Using these conditions, one can show that the radiation ratio must be:

$$s = \mathrm{Re}\left[\frac{1}{\sqrt{1-\left(\frac{k_B}{k}\right)^2}}\right] \tag{9.7}$$

The expression shows that there are three different cases:

1. $\lambda_B < \lambda_{\text{air}}$ low frequency $f < f_c$
2. $\lambda_B = \lambda_{\text{air}}$ critical frequency $f = f_c$
3. $\lambda_B > \lambda_{\text{air}}$ high frequency $f > f_c$

At frequencies below the critical frequency there cannot be any radiation of sound power to the far field. This can be understood intuitively by studying Figure 9.4. Since the distance between out of phase areas is smaller than one-half wavelength of sound in the medium surrounding the sheet, destructive interference will prevent sound radiation. One sometimes calls this condition an *aerodynamic short circuit*. We studied this type of cancellation for dipoles and quadrupoles in Chapter 1. In consequence, with the absence of radiated sound power, the radiation ratio must be zero.

Note, however, that close to the surface there will be audible sound pressure and particle velocity. This sound field carries reactive power and is sometimes called a *reactive near-field*. The sound pressure in the near field drops quickly with distance. It is important to remember though that the flow velocities in the near field may be very high.

The bending wave vibration of a panel can be damped by a porous sound absorber placed nearby due to the acoustic losses introduced by the flow resistance of the porous absorber. This property is used

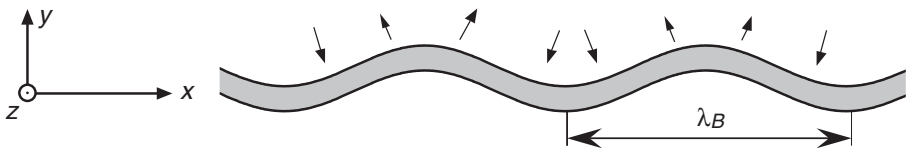


Figure 9.4 Volume flow in the near field at one side of a sheet carrying a bending wave in the x direction when $\lambda_B < \lambda_{\text{air}}$, that is at low frequencies $f < f_c$, (only one side shown).

in the design of the EMT reverberation plate shown in Figure 8.10. The reverberation time is adjusted by moving a porous sound absorber various distances from the steel sheet.

At the critical frequency, the sound radiation will be as a wave that is *radiated parallel to the sheet*. At frequencies above the critical frequency—under the assumed ideal conditions—the sheet will radiate plane waves off both its sides. (Remember that the radiation ratio only describes radiation to one side.) The plane waves radiate at an oblique angle φ as shown in Figure 9.5.

The radiation angle φ is obtained from the expression:

$$\lambda_B \sin(\varphi) = \lambda \tag{9.8}$$

The radiation will depend on the matching of the sound field in the sheet to that in the surrounding medium.

Figure 9.6 shows, in summary, the basic frequency dependence of the radiation ratio for an unattenuated plane bending wave in an infinite plane sheet. If the sheet is finite, the wave field in the medium surrounding the sheet will lose some of the symmetry that is necessary for the cancellation of sound radiation at frequencies below the critical frequency. The radiation will be primarily from the areas close to the discontinuity.

Bending Waves in Finite Sheets

The larger the dimensions of the sheet compared to the wavelength of sound in the medium, the smaller the sound radiation since the relative influence of the sound radiation by edges and corners will diminish due to their relatively smaller areas. The radiation ratios for some such cases are shown in Figure 9.7. The data shown in this figure applies to the case of a strip of material, on two edges, carrying an (unattenuated) bending wave that radiates from the ends of the strip (at the edges).

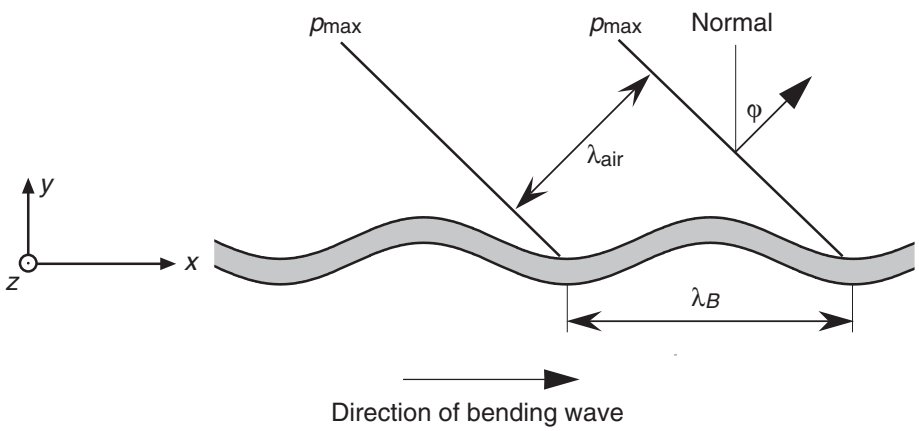


Figure 9.5 Sound radiation by bending waves at one side of a sheet at high frequencies, that is, $f > f_c$ will be at an angle ϕ to the sheet. The wavelength of the bending wave is longer than that of the longitudinal wave in air $\lambda_B > \lambda_{air}$. (only one side shown).

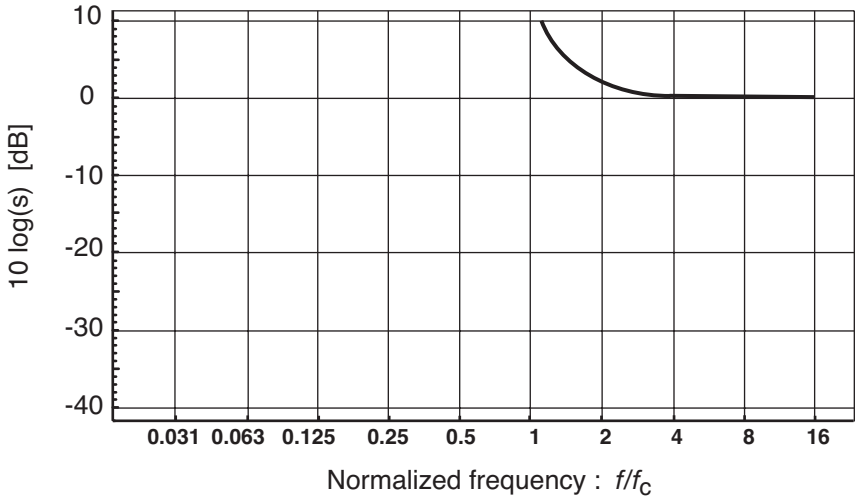


Figure 9.6 The radiation ratio, shown as 10 log(s), as a function of frequency for an undamped plane bending wave in an infinite sheet.

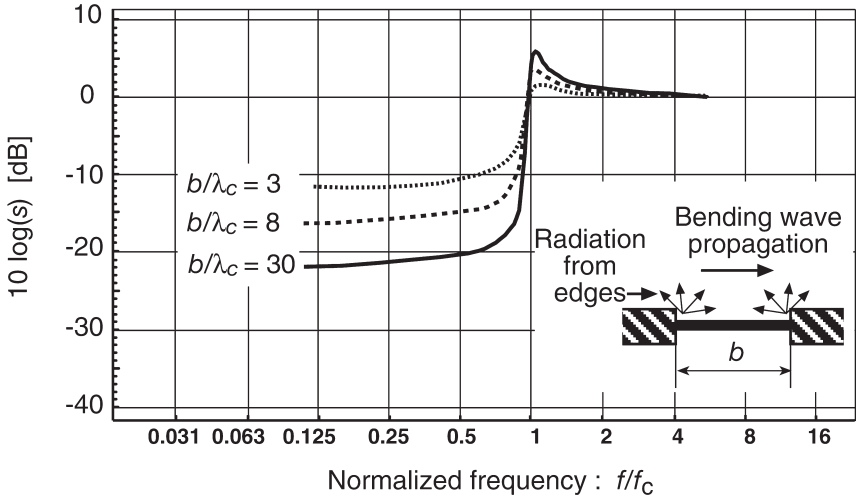


Figure 9.7 The radiation ratio, shown as 10 log(s), as a function of frequency for an undamped plane bending wave in a finite sheet of width b for some values of b/λ_c. The wavelength at the critical frequency of the sheet is λ_c.

Note that the symmetry leading to aerodynamic sound cancellation is also disturbed when a sheet is stiffened—for example, by beams, battens or point attachments. This results in sound radiation at frequencies below the critical frequency. Increasing the stiffness of a sheet does not necessarily lead to lower sound radiation by bending waves. The flat panel loudspeaker shown in Figure 9.8 has a suitable bending stiffness to mass ratio.

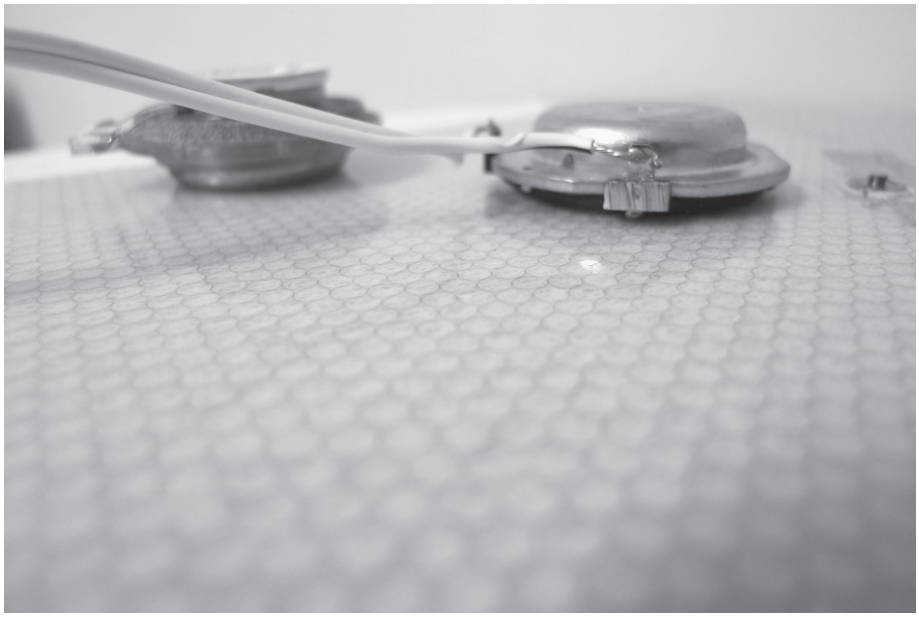


Figure 9.8 The NXT distributed mode loudspeaker uses the special properties of a sandwich sheet to achieve a suitable distribution of modes for a flat power radiation response. The sheet is extremely resonant and is primarily damped by edge and radiation losses. Note the two electrodynamic shakers. (Photo by Mendel Kleiner.)

Bending Waves in Damped Sheets

Another case of reduced symmetry is that of a sheet in which the bending wave is damped perhaps by internal losses. Again, the reduced symmetry results in sound radiation below the critical frequency, as shown by the radiation ratio curves shown in Figure 9.9. Increased damping is accompanied by more radiation, thus damping a sheet does not necessarily lead to less sound radiation.

Reduction of Radiation from Bending Waves in Sheets

As Equation 9.2 indicates, there are only a few fundamental ways of limiting the sound radiation by bending waves in sheets.

One can reduce the coupling to the medium by using finely perforated sheets, since this leads to an aerodynamic short circuit between the out-of-phase front and back sides of the sheet. Another way is to use framework constructions instead of sheet constructions. It is usually advisable to strive for high critical frequencies, if possible. Sheets should be attached where the vibration velocity levels are small. It is often advantageous to attach the sheet at a number of points instead of along a line, since the attachment point or line will cause a disruption in the symmetry of the field.

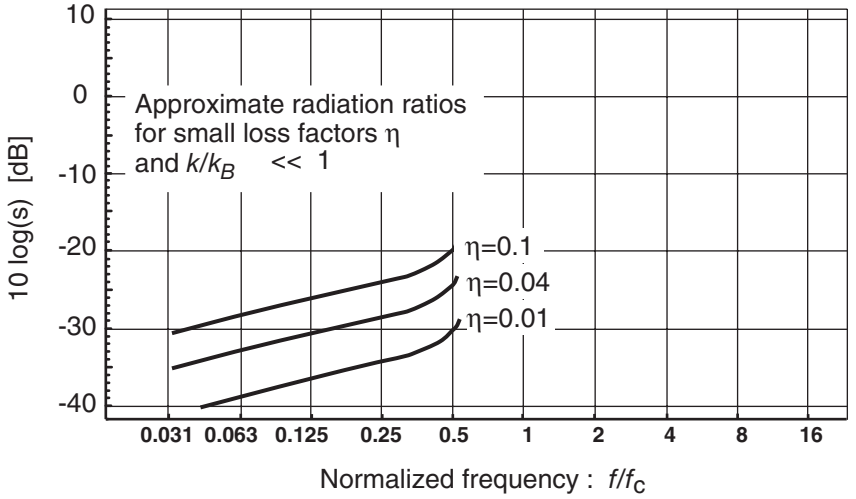


Figure 9.9 The radiation ratio, shown as 10 log(s), as a function of frequency for a plane bending wave in a sheet for some cases of damping at frequencies below the critical frequency.

9.6 SOUND GENERATION BY FLOW

The flow of gas or fluid can also result in sound generation, due to pulsating flow, vortices, turbulence, and cavitation. The elementary sound sources can be of *monopole*, *dipole*, or *quadrupole* types. The term *turbulence* is generally used to describe the property of the flow when the sound generated has a wide noise-like spectrum.

The pulsating flow from an exhaust tube or a siren are examples of *monopole* type aerodynamic sources. When a gas flows past a body, such as a rod, vortices are generated—diametrically in case of a rod—creating a *dipole* type aerodynamic source. An example of this type of sound is the whine from telephone wires on a windy day. Note that the dipole force on the rod can cause large movement if the resonant frequency is the same as the frequency of the dipole sources. *Quadrupole* sources are easily created around the flow from orifices because of viscous stress within the flow. The core flow will be surrounded by vortices, having a quadrupole property. This type of noise source is characteristic of jet aircraft engines and other high-speed gas exhausts.

Noise due to turbulent flow makes it easy to detect the rotating propellers of submarines and other ships. The noise from landing aircraft is dominated by noise due to turbulence around the fuselage. In audio engineering, it is sometimes necessary to design the bass-reflex ports of loudspeaker cabinets so that turbulence is not created.

The noise at close distance to the turbulence or vortices can be disturbing. Typically, the flow in air becomes turbulent at flow speeds above 7 m/s. Turbulent flow outside is one of the main sources of noise in the interior of an aircraft or car. Even when biking, the turbulence generated by the air flow around the head is easily audible.

Turbulence-generated noise due to wind can make it difficult to measure sound outdoors. The turbulence created by the microphone capsule and body is close to the sound sensing membrane of the

microphone. This results in high sound pressure at the membrane and associated poor signal-to-noise ratio in the measurement.

One can reduce the problem by using various forms of *wind screens*. These work by removing the region of turbulence from the membrane. Foam plastics or fine mesh cloth on a wire grid may be used for this purpose. In spite of the larger volume of turbulence, the noise sensed by the microphone is diminished since the attenuation of sound by the increase in geometrical distance is so strong. An example of this effect is given in Chapter 12.

Acoustic Efficiency

Air flow at orifices is typically characterized by presence of all the three source types mentioned. The sound power W generated is proportional to the mechanical power in the flow W_{mec}. One writes:

$$W = \eta W_{\text{mec}} \tag{9.9}$$

where η is the *acoustic efficiency*.

In the case of a streaming gas jet out of an orifice in a sheet, the mechanical stream power of the jet is given by:

$$W_{\text{mec}} = \Delta p u_{\text{flow}} S = \frac{\rho_0 u_{\text{flow}}^3 S}{2} \tag{9.10}$$

where ρ_0 is the density of the gas stream, u_{flow} the flow speed of the gas at the orifice, and S the area of the orifice. The pressure drop across the orifice Δp determines the speed of the gas flow at the orifice.

The acoustic efficiency is determined by the flow speed of the gas u_{flow} and by the speed of sound in the surrounding medium.

The acoustic efficiency of the noise generation by flow is:

$$\eta = c_m \frac{u_{\text{flow}}}{c_0} + c_d \left(\frac{u_{\text{flow}}}{c_0} \right)^3 + c_q \left(\frac{u_{\text{flow}}}{c_0} \right)^5 \tag{9.11}$$

where c_m, c_d, and c_q are the efficiency coefficients of monopole, dipole, and quadrupole sound generation respectively. Since the mechanical stream power is proportional to the flow speed u_{flow}^3, the contributions to the noise by the three source types will depend on flow speed as u_{flow}^4 for monopole, u_{flow}^6 for dipole, and u_{flow}^8 for quadrupole sources (see Reference 9.1).

The balance between volume flow and flow speed is important. If one, for example, wants to feed air to a room through a vent opening having a grille, dipole sound sources turn out to be the most important at typical flow speeds. For a certain volume flow, one can reduce the flow speed u_{flow} and increase the vent area S, or vice versa. Equations 9.10 and 9.11 show that to minimize flow noise it is advantageous to limit the flow speed by increasing the flow area. This is a general property of flows. Turbulence noise may be disturbing in low frequency sound reproduction by ported loudspeaker boxes, such as the one shown in Figure 9.10.

The spectrum of the turbulence-generated sound will typically be broadband (as shown by the curve in Figure 9.11), unless the associated mechanical structure is resonant. In air, the peak in the spectrum will be at a frequency f_{peak} given by:

$$f_{\text{peak}} \approx 0.2 \frac{u_{\text{flow}}}{d} \tag{9.12}$$



Figure 9.10 Note the flare of the three bass-reflex port openings of this SVSound™ PB12-Plus subwoofer. Many ported loudspeaker enclosures use a flared port to reduce the *swooshing* noise associated with turbulence generated by turbulence in air at high flow velocities. An application of this technology can be seen in the loudspeaker shown here (photo courtesy of SVSound™ [all rights reserved]).

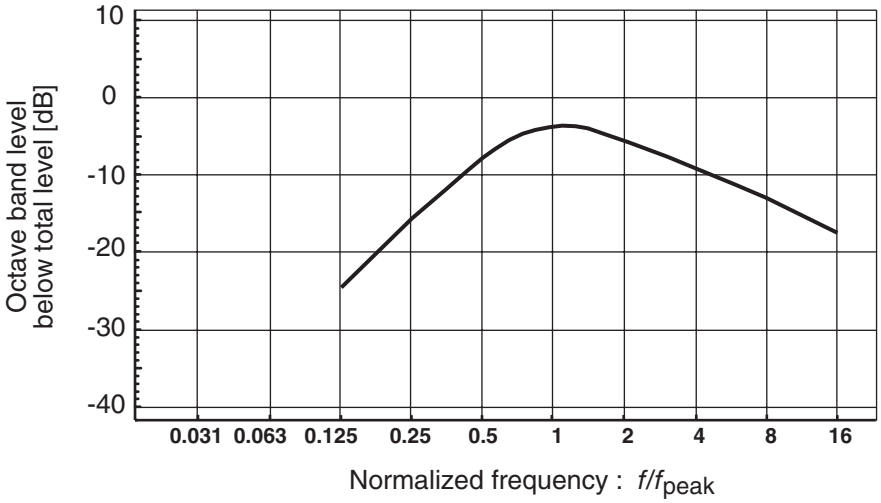


Figure 9.11 Typical noise power spectrum for turbulence generated by air blowing through a grille.

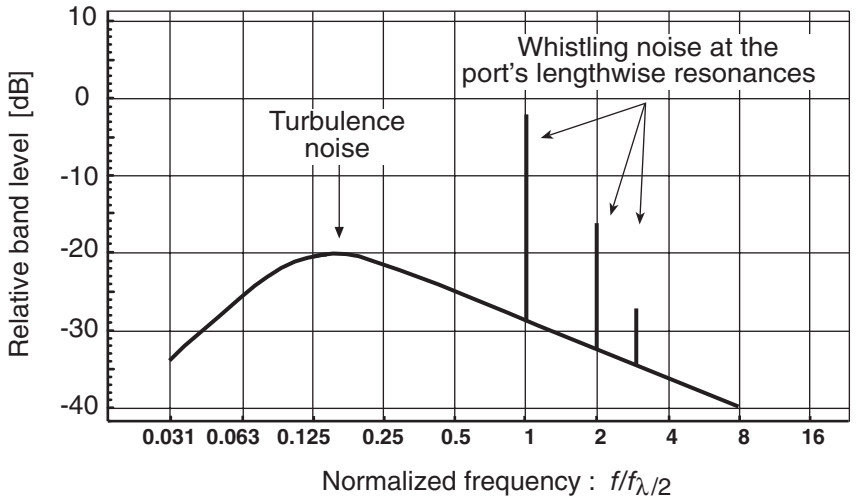


Figure 9.12 Typical noise power spectrum generated by high speed air blowing through a bass-reflex port at the Helmholtz resonance frequency. Whistling noise frequency is normalized to the half-wavelength resonance frequency of the port tube $f_{\lambda/2}$.

where d is the cross-sectional width of the obstruction and u the flow speed. The constant 0.2 is the so-called Strouhal's number for this case (see Reference 9.1).

Flow-generated noise may be a problem in loudspeaker design. In ported loudspeaker enclosures, such as bass-reflex and bandpass loudspeaker boxes and the one shown in Figure 9.10, the particle (flow) velocity may become large in the Helmholtz resonance frequency region. There are two sound effects involved in shaping the spectrum characteristic shown in Figure 9.12.

One is the pulsating tonal sound effect due to vortex excitation of the port lengthwise resonance. The other is a wide band turbulence-noise component generated by vortex shedding immediately in front of the port openings. Whistling noise will be modulated at double the frequency of the excitation, particularly in the frequency region close to the port's Helmholtz resonance frequency, since the flow speed is high at both negative and positive flow speed peaks (see Reference 9.3). These types of noises may also occur in conventional Helmholtz resonators used for sound absorption.

To reduce these noises, it is important to apply the rule of keeping the particle (flow) velocity low by using a wide port and to avoid sharp edges at the port openings.

9.7 PROBLEMS

9.1 Two identical loudspeakers in an outdoor arena are connected to the same signal source and radiating sound to reach the audience. The audience is seated on a plane field that is grass covered and has high sound absorption. Assume the loudspeakers are mounted at a height of 6 m above the audience's head level, and at a distance 10 m apart.

Task: Calculate numerically the approximate loci of the negative interference dips at a frequency of 340 Hz.

9.2 Assume that a circular loudspeaker mounted in a large baffle, its diaphragm has a radius a and is vibrating perpendicular to its surface with a velocity $\underline{u}$. Because of the closed loudspeaker box, there is no radiation from the back side of the loudspeaker. For frequencies where the loudspeaker's dimensions are small compared to wavelength, the real part radiation impedance seen by one side of the loudspeaker diaphragm is approximately given by:

$$R_{MR} = \frac{\pi a^2 Z_0}{2}\left[1 - \frac{J_1(2ka)}{ka}\right]$$

Task: Compare the power radiated by the loudspeaker to that radiated by a spherical, monopole loudspeaker having the same surface area and vibration velocity.

9.3 A piano lid is excited by a shaker and is vibrating with the octave band velocities shown in the following table. The lid's area is 2 m^2, the room volume 50 m^3, and the reverberation time 0.8 s. The diffuse field sound levels in the room are also given by the table.

f	0.5	1	2	4	[kHz]
L_u	135	122	113	112	[dB] re 10^{-9} m/s
L_p	89.9	84.9	80.9	79.9	[dB] re $2 \cdot 10^{-5}$ Pa

Task: Calculate the radiation ratio of the piano lid.

9.4 A small loudspeaker having a radius of 0.15 m is mounted at the top, in the side of a tall rectangular box that is 1.2 m high and has a square cross-section with sides that are 0.4 m. During measurement of the acceleration levels of the loudspeaker cone and the box walls, it was found that the vibration levels of the side walls were on the average 10 dB below that of the cone at a frequency of 90 Hz.

Task: Calculate the approximate level difference between the radiated sound power from the loudspeaker box walls and the cone in an anechoic environment.

9.5 A ported box loudspeaker similar to the one shown in Figure 9.10, but with only one port, has its port resonance tuned to 34 Hz. The circular port area is 0.01 m^2. During a test of the loudspeaker, it was fed a sine-wave signal at the port resonance frequency. A certain drive level turbulence-generated noise was beginning to be heard from the port opening. Experience has shown that when port air velocities reach about 7 m/s, turbulence becomes noticeable.

Task: Estimate the acoustic power radiated by the port. Estimate the spectrum of the noise, assuming $c_m \approx 10^{-8}$.

Sound Isolation 10

10.1 INTRODUCTION

Building acoustics is the branch of architectural acoustics dealing with the transmission of sound through partitions, enclosures, and openings. Building acoustics is often discussed in terms of sound transmission including airborne, structure-borne, and impact. In most cases, the noise transmitted into a room is due to many sources and comes by many ways.

Sound may be transmitted from room to room as shown in Figure 10.1. The transmission can be: (1) directly through the wall—direct transmission, (2) by an interconnecting structure as vibration—flanking transmission, or (3) through open-air paths.

Airborne sound generally means sound transmission through walls, windows, doors, ceilings and floors, provided that the structure is initially excited by sound in air. It is the sound that is incident on a building partition, moves the partition, and is reradiated in an adjacent space by the partition's vibration. Typical examples of transmitted airborne sounds are voice, sound from loudspeakers, noise from handheld tools, and noise from traffic. The *transmission loss* of a construction is a measure of the sound insulating properties of a wall or other partition.

Impact sound is the sound that is generated in neighboring rooms in a building when such things as floors, walls, and stairs are subject to impacts by footsteps, hammers, and the like. *Impact sound pressure* is a measure of the ability of the building to transmit and radiate sound that is transmitted mechanically to the structure. Note the difference between *airborne* and *structure-borne* sound.

Structure-borne sound generally means sound excited mechanically that travels from the source to the receiver mainly by way of mechanical paths. Structure-borne sound can come from steps in staircases, noise from elevators and their machinery, noise from the closing of doors, noise due to heating, ventilation, and air-conditioning machinery (HVAC), ducts, and outlets.

10.2 INSULATION AGAINST AIRBORNE SOUND

Assume two adjacent rooms separated by a wall with a sound source in one of the rooms that is known as the *sending room*. The walls of the sending room will vibrate due to the sound pressure of the incident sound. Normally, the wall is constructed in such a way that the side of the wall exposed to the *receiving room* will vibrate as well, due to the sound in the sending room.

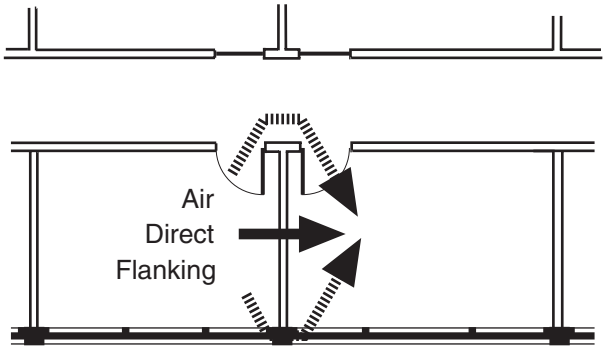


Figure 10.1 Sound can move from one room to the next by three types of room-to-room paths: (1) direct, (2) air, and (3) flanking transmission.

In practice, it is the sound level difference between the adjacent rooms that is of interest. The sound level difference is influenced not only by the sound transmission properties of the acoustically separating wall, or *direct transmission*, but also by adjacent walls and air ducts. This added component is called *flanking transmission* and is discussed later.

10.3 SOUND TRANSMISSION LOSS

Denote the sound power incident on the wall in the sending room, W_{inc}, and the sound power radiated into the receiving room, W_{trans}.

The *transmission coefficient* τ is defined as:

$$\tau = \frac{W_{\text{trans}}}{W_{\text{inc}}} \tag{10.1}$$

Transmission loss, TL, also called the sound reduction index, R, is given in dB and defined as:

$$R = -10\log(\tau) \tag{10.2}$$

The transmission loss is angle dependent. In the case of an inside wall, the sound field is likely to be diffuse, so there will be an average of the wall's angle-dependent transmission loss, which will be the *effective* diffuse field transmission loss generally referred to.

The difference between the mean sound pressure level due to an unspecified sound source in the sending room L_{pS} and the mean sound pressure level in the receiving room L_{pR} is called the *noise reduction* (NR). The NR will be determined by a few factors. If the sound fields in the rooms are diffuse, the NR will depend on the area of the partition separating the rooms S_p, its transmission loss R_d, and the ability of the receiving room to absorb sound, that is, its sound-absorption area A_R.The sound absorption in the sending room will affect the sound level generated by the source in that room.

The sound transmission through a partition between rooms will lead to an energy exchange between the rooms and the partition, which make up three subsystems. In the frequency range above

the Schroeder frequency, the sound fields are diffuse, and we can then use *statistical energy analysis* to study the acoustical interaction between the rooms. If the partition is assumed to be nonresonant (as when the wall behaves as a mass) one can simplify the situation to only involve the resonant energy in the two rooms, as shown in Figure 10.2. Using Equations 4.28, 4.33, and 10.1, the power balance equations can then be written:

$$W_+ + \tau\frac{P_R c S_p}{4} = \frac{P_S c A_S}{4} + \tau\frac{P_S c S_p}{4} \qquad \tau\frac{P_S c S_p}{4} = \frac{P_R c A_R}{4} + \tau\frac{P_R c S_p}{4} \tag{10.3}$$

If the partition is resonant (as it is above the critical frequency), a third system needs to be included between the two rooms in the figure. If one assumes that the transmission loss of the wall is larger than 10 dB, it is not necessary to take the sound being returned through the partition to the sending room from the receiving room into account. This will be the general case in most buildings in the temperate zones. The power through the partition from room *S* is simply absorbed by the walls of room *R*. We then obtain the following equation for the power balance between the reverberant sound fields of the sending and receiving rooms:

$$\tau\frac{P_S c S_p}{4} = \frac{P_R c A_R}{4} \tag{10.4}$$

This gives the ratio between room averaged squared sound pressures as:

$$\frac{<\tilde{p}_S^2>}{<\tilde{p}_R^2>} = \tau\frac{A_R}{S_p} \tag{10.5}$$

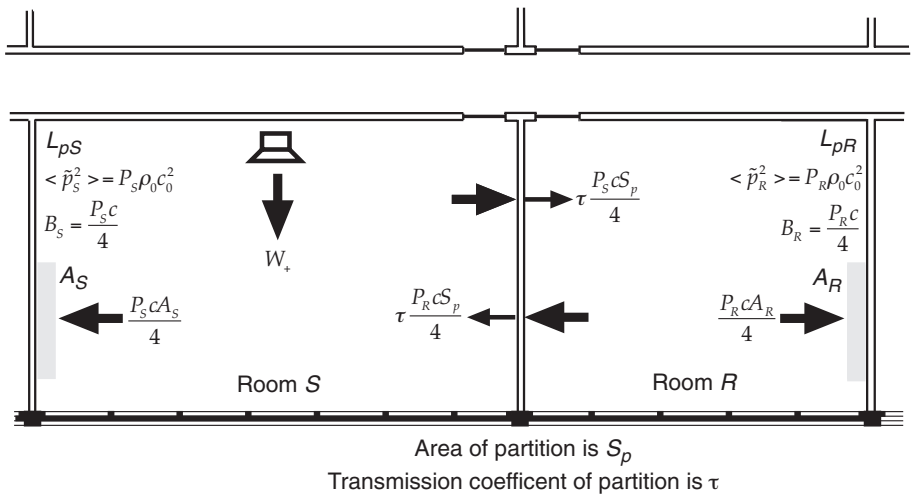


Figure 10.2 The arrows show the power flows from loudspeaker to sound absorbers in the sending and receiving rooms.

Typically, we are interested in the noise reduction (NR) between rooms, that is, the difference in sound pressure level that is obtained for a certain partition and receiving room. Notice that NR is not the same as R; the sound level difference is affected by the *room correction*. Our hearing ultimately senses NR, not the transmission loss. Taking the logarithm of Equation 10.5, we obtain the much-used Equation 10.6:

$$NR = L_{pS} - L_{pR} = R + 10\log\left(\frac{A_R}{S_p}\right) \tag{10.6}$$

Equation 10.6 is generally used in the experimental determination of transmission loss as well as in determination of the necessary sound insulation in buildings.

10.4 SOUND REDUCTION INDEX

Usually the transmission loss is averaged over octave or third octave bands, from at least the 125 Hz octave band (possibly also down to the 50 Hz third octave band) up to the 4 kHz octave band. The reason for not measuring at lower frequencies is the lack of measurement accuracy at low frequencies, where the room sound fields have little diffusitivity because of the low number of modes. However, after specifying the measurement conditions, there is nothing that prevents one from measuring NR—realizing that the data cannot be used with Equation 10.6. Lightweight constructions, such as double-panel drywalls and lightweight concrete block walls, may have poor transmission loss, particularly at low- and mid-frequencies.

It is common to use a sound reduction index R_w to grade the sound transmission loss properties of walls. The calculation of R_w is briefly discussed in the caption to Figure 10.3. The definition of the calculation of the sound reduction index is given in ISO standards. The ISO has published many standards relating to sound insulation and its measurement. The reader is referred to these standards as appropriate. The definition is usually modified nationally, so one should check the national standards applicable. The *Sound Transmission Class* (STC) rating used in the United States to characterize sound attenuation by building elements is calculated in a similar way as R_w.

It is common to differ between results from laboratory measurements R_w and those from measurements in the field R'_w (in the United States, Field STC (FSTC) is used instead). The results from field measurements may often be several decibels lower than those obtained from the laboratory measurements of the same wall type. Sound transmission loss and sound reduction index data may be found at the various websites of building materials manufacturers and national and private building research and testing organizations and institutes.

10.5 TRANSMISSION LOSS OF SINGLE-PANEL PARTITIONS

A single-panel (single-leaf) partition is an impervious, homogeneous building part made of one plane panel (sheet) of material. It is the mass per unit area, the bending stiffness, and the loss factor that determines the transmission loss and its frequency characteristic.

A finite-sized single panel partition must be supported in some way at its edges. The type of support depends on the building construction. For a lightweight partition, the transmission loss behavior varies at different frequencies. At low frequencies, it will behave as a spring (the *stiffness region*) and then show

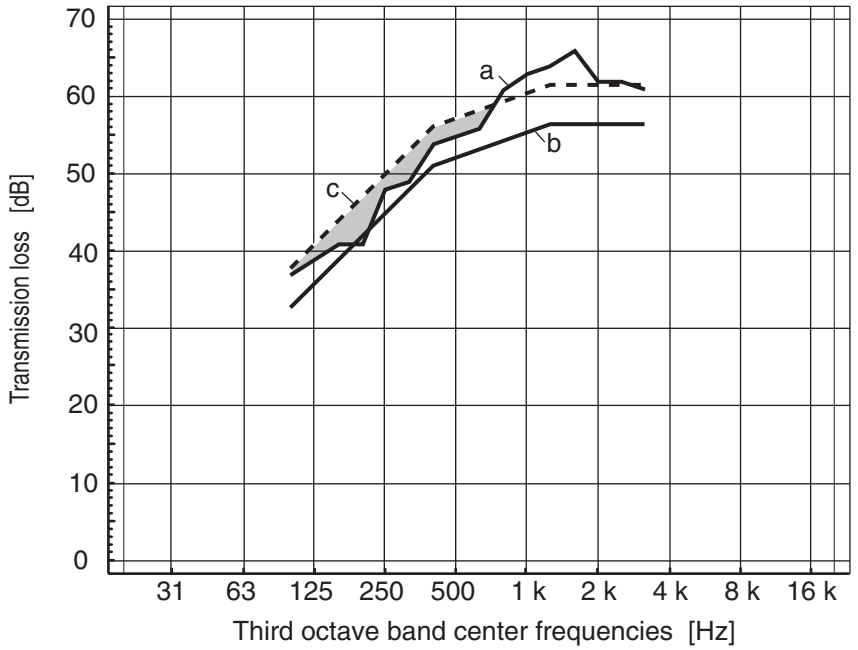


Figure 10.3 Example of calculation of the sound reduction index, R_w: (a) measured sound insulation in third octave bands, (b) reference curve, and (c) reference curve shifted to allow sum of differences above curve a but below the reference curve to be ≤ 32 dB (at the third octave band frequencies from 125 Hz to 3.2 kHz). The R_w value is then determined as 52 + the number of decibels that curve *b* has been shifted upward to fulfill the criterion. (Downward shift subtracts decibels.)

resonant behavior at the low resonance frequencies (the *resonance region*). It will then behave as a mass at higher frequencies (the *mass law region*) because of damping at the edges and in the material. Finally, at extremely high frequencies (the *coincidence region*) and above, there is another type of resonance phenomenon, already discussed in Chapter 9, where the partition may become approximately sound transparent. The transmission loss curve has the basic shape shown in Figure 10.4.

In practice, all of these characteristics are seen only in building acoustics range measurement data for simple, single-pane windows. For most other building partitions, the resonance region is below 100 Hz; it is then that transmission loss in the mass law, the coincidence, and the frequency regions above the critical frequency are usually measured. Single-panel partitions are usually made of poured concrete, concrete building blocks, or bricks. Such heavy, stiff walls have their resonance and critical frequency regions below 100 Hz that results in good sound insulation properties also at low frequencies.

The transmission loss depends on the diffusitivity of the incident sound field. Equation 10.6 was derived under the assumption that the room sound fields are diffuse. The sound field due to sound incident on the facade of a building will not be diffuse—in that case, a different approach is needed.

In Chapter 7, we studied the influence of a limp panel on the sound absorption of an ideal sound-absorbing material (for perpendicular sound incidence). We can study transmission loss of a panel in a similar way for perpendicular sound incidence. The sound power absorbed will be sound power

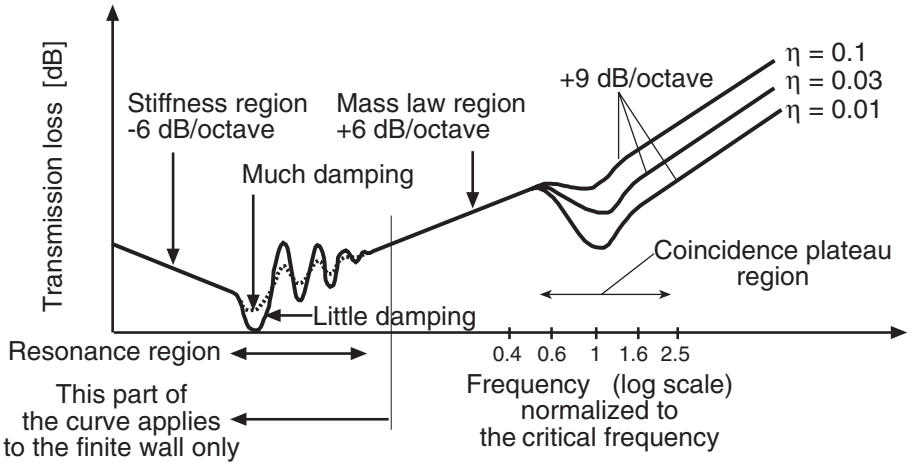


Figure 10.4 The transmission loss of the single-leaf partition shown normalized to the critical frequency for some values of the loss factor η of the material of the partition.

transmitted through the panel. Equation 7.9 gives the ratio of sound absorbed to the sound power incident, that is, α. Since no power is absorbed by the panel in the present case, the absorbed sound is equal to the sound power transmitted to the sound power incident. The transmission coefficient τ_0 will be:

$$\tau_0 = \frac{1}{1+\left(\frac{\omega m''}{2Z_0}\right)^2} \tag{10.7}$$

where m'' is the mass per unit area. From Equation 10.7, we find that the transmission loss for a plane wave, incident perpendicularly on a limp, single panel partition in the mass law region, is approximately given by:

$$R_0 \approx 20\log(m''f) + 42 \tag{10.8}$$

In the common case of a diffusely incident sound field, the transmission loss is slightly reduced:

$$R_d \approx R_0 - 7 \tag{10.9}$$

Lightweight, single-panel partitions are usually found only in window constructions and some indoor and old outdoor wall types. The transmission loss in the stiffness region will not be discussed here, but is important in the case of loudspeaker boxes and other types of enclosures.

The critical frequency of a panel can be calculated using Equation 8.20. Table 8.1 lists the critical frequencies f_c of some typical building materials.

The transmission loss will drop considerably in the coincidence region, particularly noted if measured with good frequency resolution. The coincidence phenomenon occurs first for obliquely incident waves that propagate parallel to the partition. As the frequency of sound increases, the coincidence phenomenon will occur at smaller angles of incidence. Around each angle of incidence, there will, how-

ever, be only a few modes with similar incidence angles. Since the energy in the sound field is divided between the modes, this means that only a part of the sound field energy is available for transmission at each frequency. This prevents the transmission loss from becoming zero. It is customary to approximate the resulting *coincidence plateau* by a constant transmission loss in the frequency range between 0.5 to 2 times the critical frequency. For the coincidence plateau region, the transmission loss will be approximately:

$$R_{d,f_c} \approx R_d + 10\log(\eta) + 8 \tag{10.10}$$

Over the coincidence plateau frequency region, the transmission loss can be approximated by the expression:

$$R_{d,f>f_c} \approx R_0 + 10\log\left(\frac{f}{f_c} - 1\right) + 10\log(\eta) - 2 \tag{10.11}$$

In summary, one can note that the transmission loss theoretically increases at a rate of 6 dB per octave up to the coincidence plateau and after the end of the plateau by 9 dB per octave. Note that both in the plateau region and above, the transmission loss depends on the loss factor of the panel; this is a result of the resonant behavior of the coincidence phenomenon.

Transmission loss data for some typical single-panel partitions are shown in Figure 10.5. We notice that high transmission loss is provided by massive, heavy materials such as concrete. Concrete and brick have better sound isolation properties than steel, glass, plywood, or gypsum panels. Porous sound-absorbing materials (such as expanded polystyrene and glass wool) have extremely poor sound

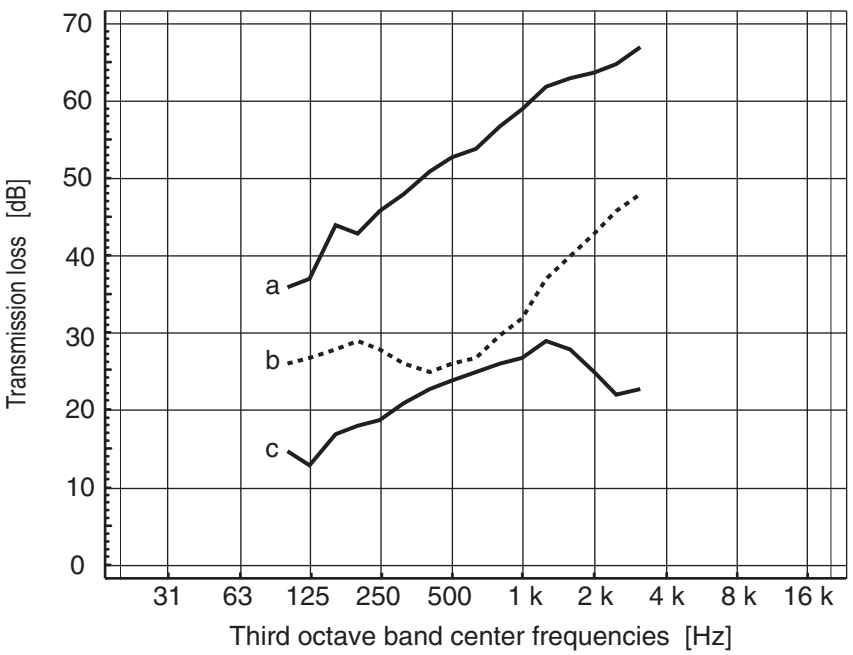


Figure 10.5 Examples of some measured transmission loss curves for single-panel walls: (a) 160 mm concrete, $f_c \approx 100$ Hz, (b) 70 mm lightweight porous concrete, $f_c \approx 400$ Hz, and (c) 13 mm drywall panel, $f_c \approx 2.0$ kHz.

isolation because of the open pores (glass wool) and the low mass of the porous sound-absorbing panel (both). Sound-isolating materials must not transmit air (must be impervious) and must either be rigid or have high mass per unit area. It is important to note that heat-insulating materials are generally poor sound isolators. One should not confuse the NR provided by acoustic absorbers within a room with NR between rooms.

10.6 TRANSMISSION LOSS OF MULTIPLE-PANEL PARTITIONS

The unsatisfactory transmission loss of single-panel partitions can often be increased considerably by the use of multiple panels. If the total panel mass/unit area is small, the transmission loss will still be poor, particularly at low frequencies.

The simplest form of multiple partition uses two panels with a mechanically separating airspace. In practice, it is necessary to use metal or wooden studs to provide a mechanical frame to which the panels can be attached. Attachment is usually done using nails or screws. Because of the mechanical attachment, a practical multiple panel partition has a lower transmission loss than theoretically predicted. Examples of such double-panel partitions are dual- or multiple-pane windows and drywall partitions using gypsum, plywood, or chipboard. Sometimes, when high transmission loss is required, double-panel partitions using concrete or bricks are necessary to ensure sufficient noise rejection.

Studs must be well separated to prevent fastenings from causing too high a stiffness. Steel studs will have lower stiffness than wooden studs that allow the individual panels to behave more according to the mass law. Even better results can be obtained by separating the two panels by using staggered studs (separate studs for each panel).

For the discussion to follow, it is assumed that the mass law applies, that there are no mechanical connections between the panels, and that the critical frequencies of the panels are much higher than any frequency of interest.

The masses of the two panels, in combination with the compliance of the air trapped between the panels, will form a resonant system. The compliance effectively mechanically decouples the panels. The resonance is called the *double-wall fundamental resonance*, and the frequency at which it occurs is the fundamental resonance frequency. Note that this resonance is of a different form than that of the single-panel discussed earlier. (The coincidence resonance phenomenon will be discussed in more detail later, but the higher order resonances will not be discussed here.)

The theoretical transmission loss, as a function of frequency, for an infinitely large double-panel partition with only airspace separating the panels is shown in Figure 10.6. The transmission loss at the resonance frequency depends on the damping of the system. If the panels of the partition are free, the fundamental resonance frequency f_0 of the double-wall for perpendicularly incident waves, is:

$$f_0 = \frac{c_0}{2\pi}\sqrt{\frac{\rho_0}{d}\left(\frac{1}{m_1''} + \frac{1}{m_2''}\right)} \tag{10.12}$$

where m_1'' and m_2'' are the respective masses per unit area of the panels, and d is the depth of the airspace. The fundamental resonance frequency of the double-panel partition is determined by the masses and the air compliance, as if the system consists of two masses joined by a spring. Figure 10.7 shows a double-panel gypsum wall under construction.

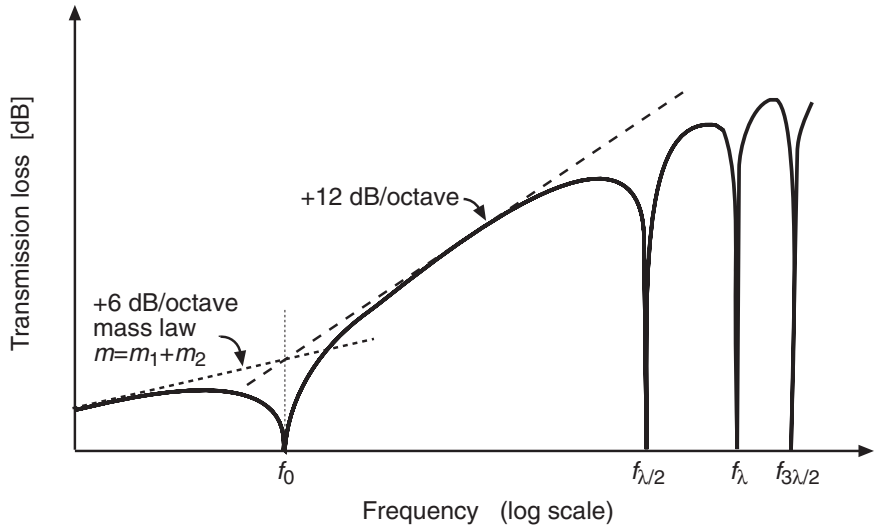


Figure 10.6 The transmission loss, as a function of frequency for perpendicular sound incidence, for an infinitely large double-panel partition (solid line) with only airspace separating the panels. The transmission loss curve for a single-panel partition (dotted line) having the same total mass as the double wall is shown for comparison. Dashed line is the asymptotic line for a double wall not having standing waves in the airspace. The effect of such standing waves is shown at the frequencies where the airspace distance between the two panels is a multiple of one-half wavelength. Curves shown for $f_0 << f_c$.



Figure 10.7 A double-panel drywall under construction showing the back of one panel and the steel studs, waiting for glass wool filling and the second panel to be installed.

At frequencies below the fundamental resonance, both panels vibrate in phase as one joined mass, and the double-panel partition behaves as a single-panel partition according to the mass law but with an equivalent mass per unit area equal to the sum of the mass per unit area for the two panels:

$$m'' = m_1'' + m_2'' \tag{10.13}$$

At resonance, the two panels oscillate in opposite phase. Above resonance, the compliance of the air trapped between the panels will act as a shunt and the transmission through the airspace will diminish as frequency increases. In this mass law frequency range, the transmission loss of the double-panel partition increases at a much faster rate, 12 dB/octave, than that of a single-panel partition, 6 dB/octave.

Sometimes one finds that the measured transmission loss curve of a double-panel partition exhibits sharp dips at high frequencies, which are unrelated to the critical frequencies of the panels. These dips are due to standing waves in the airspace acoustically coupling the two panels tightly. The standing waves occur at the frequencies where the airspace distance between the two panels is a multiple of one-half wavelength. If the airspace is wide, the double-wall resonance frequency will be correspondingly low, but at the same time, the dips will also occur at comparatively low frequencies. The resonances and dips are easily removed by partially or fully filling the airspace with a porous sound-absorptive material, such as a mineral or glass wool isolation blanket. Added advantages of such a porous blanket will be better damping of the fundamental resonance, a drop in the fundamental double-panel resonance frequency, and increased sound transmission loss in this frequency range, as shown in Figure 10.8.

Air space resonances may also occur parallel to the panels and will reduce the transmission loss. Such resonances can be damped by using sound absorption at the edges of the airspace for example in double glazing.

The influence of the coincidence phenomenon will be noted as a leveling of the transmission loss curve. For most multiple-panel partitions, however, this range is at such high frequencies as not to be a problem. If the coincidence effect is a problem, it can often be circumvented by substituting several thinner panels for a thick panel. This technique is often used in drywall construction with good results as shown by the *b* and *d* curves in Figure 10.8. Another application of this technique can be found in the design of high performance sound-insulating windows.

Special types of multiple-panel partitions are the *suspended ceiling* and the *resilient skin*. In both of these cases, the aim is to improve the sound insulation of a heavy and stiff partition, such as a concrete floor slab or wall, or a brick wall. These partitions have low critical frequencies, but, in spite of their mass, the transmission loss is sometimes still not as high as desired. In this case, the addition of an extra layer, effectively building a double-panel partition, is useful to improve the sound insulation, even if the extra layer has low mass. Usually the extra layer takes the form of a panel of gypsum or similar material having a high critical frequency.

In the case of the suspended ceiling, the airspace is usually large, a depth of some 200 mm is typical. In either case, partially filling the airspace by a mineral or glass wool blanket further increases the transmission loss, often by approximately 10 to 20 dB.

The resilient skin is typically an addition of a panel having a high critical frequency on steel or wood studs attached to an existing construction. Fire regulation often prevents the use of wood studs. The air space is usually quite thin (only 2 to 5 cm) and filled with a glass wool blanket. Figure 10.9 shows an example of the effect of such a radiation reducing resilient skin layer to a brick wall.

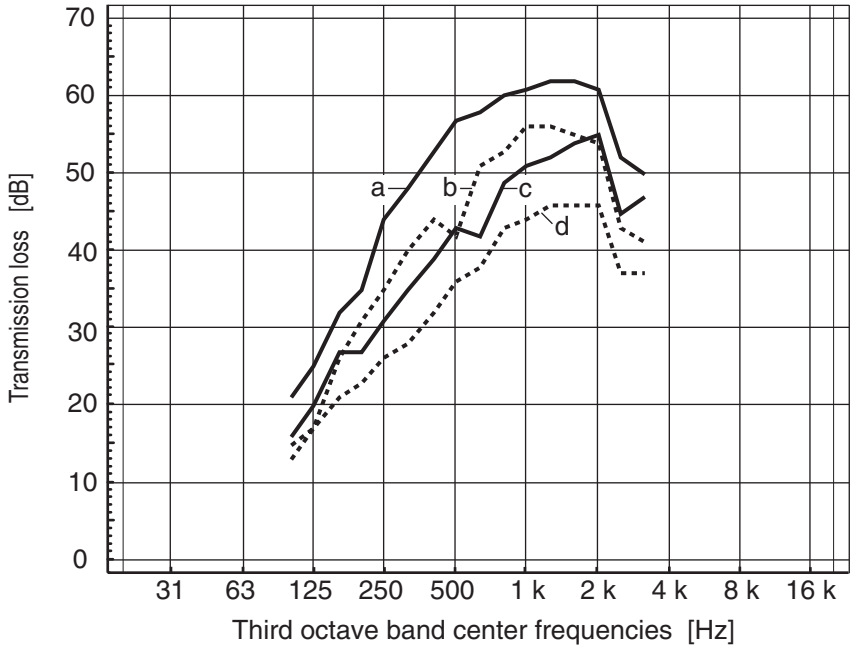


Figure 10.8 Results of some laboratory measurements of drywalls: (a) dual layer double-panel drywall with 45 mm steel studs at 0.6 m distance with 30 mm mineral wool blanket, surface mass approximately 50 kg/m^2, (b) single-layer, double-panel drywall with 70 mm steel studs at 0.6 m distance with 30 mm mineral wool blanket, surface mass approximately 50 kg/m^2, (c) dual-layer, double-panel drywall with 45 mm steel studs at 0.6 m distance, no mineral wool blanket, and (d) single-layer, double-panel drywall with 70 mm steel studs at 0.6 m distance, no mineral wool blanket.

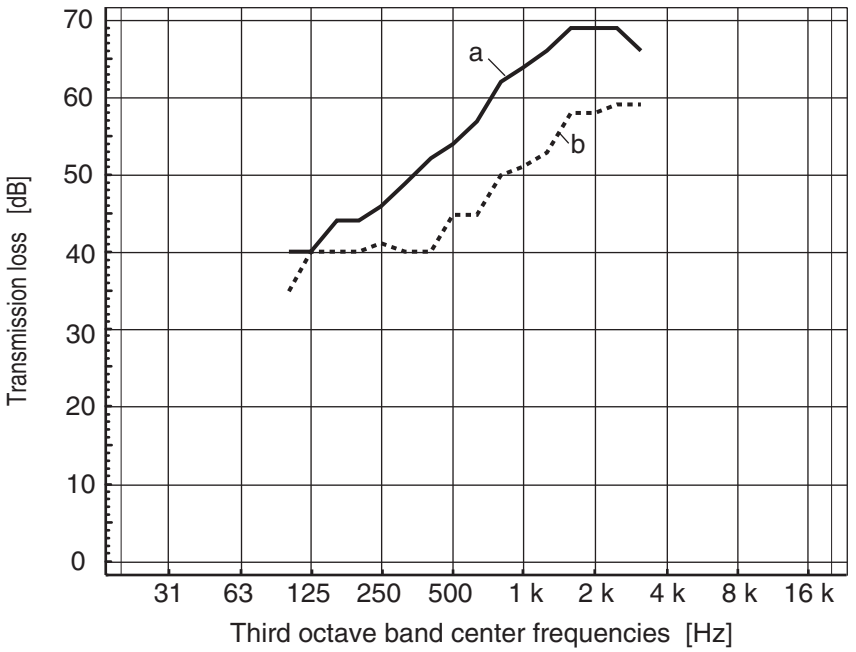


Figure 10.9 Example of the transmission loss of a 200 mm thick brick wall, with (a) and without (b) a resilient skin. The resilient skin was a 13 mm thick gypsum panel on 50 wood mm studs. The air space held a 25 mm thick mineral wool blanket.

The technique is particularly useful since it can be used to improve noise control in existing buildings having concrete or brick partitions, in an easy and inexpensive manner. Figure 10.10 shows a room where a resilient skin is being installed on the room walls.

Doors and windows are often problematic from the viewpoint of sound transmission. The panels or panes are lightweight and the separating distance small. The transmission loss curves of double-glazing often show dips due to half-wave resonances in the airspace. Such resonance dips can be avoided if the panes are angled relative to one another and sound-absorptive materials used at the edges of the air space. These techniques are commonly used in the design of high sound insulation windows—for example, in sound recording studios. Figure 10.11 shows an example of such a studio window.

As mentioned previously, it is advantageous to split the panels into thinner panels if there is danger of the critical frequency of the panels becoming too low. This is often done in window constructions. Because of the low mass per unit area of window glass panes, the fundamental resonance frequency often becomes high as shown by the curve in Figure 10.12.

Doors are usually constructed using lightweight panels, which are joined by a lightweight paper honeycomb construction that fills the airspace to provide sufficient structural rigidity. This type of door construction usually gives poor sound insulation. For good soundproofing, the doors must be heavy, resulting in handling problems. Due to these problems, it is often better to use two doors with an intermediary sluice or short corridor, if space permits. The sluice should be at least 0.2 to 0.5 m long and have sound-absorptive walls and ceiling.



Figure 10.10 A resilient layer being added to a room to improve the sound isolation (photo by Alex Hill, Soundservice, UK).



Figure 10.11 Example of a window between a recording studio and a control room that uses offset (non-parallel) glass panes and sound-absorptive material at the interior air space edges. (Photo by Mendel Kleiner.)

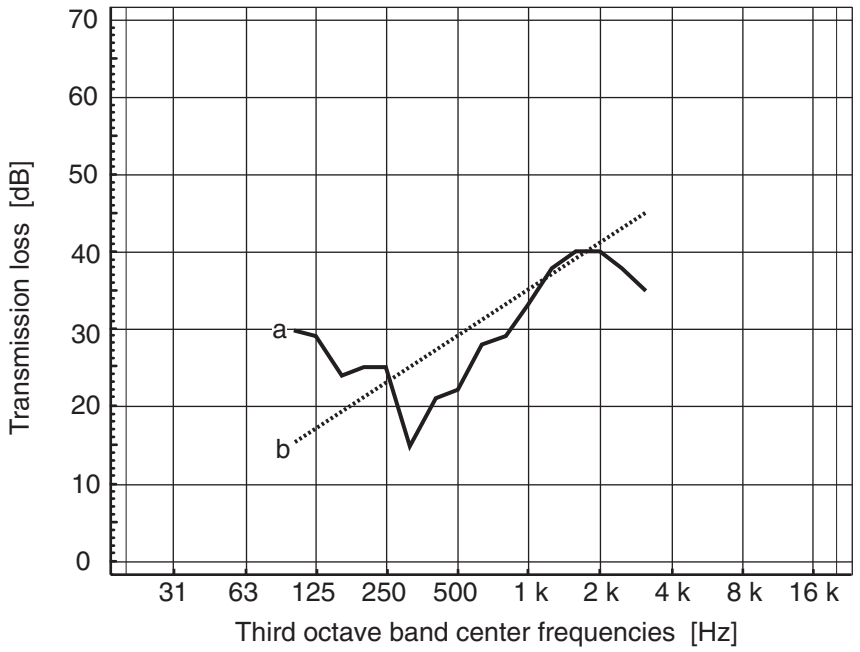


Figure 10.12 Example of the transmission loss of double glazing: (a) glass thickness 3 mm, air gap 10 mm and (b) shows the theoretical mass law curve for single 6 mm glazing having the same total mass.

Another advantage of the latter construction is that most doors and windows have leaks around their perimeters—even good seals will age—and by using a sluice, the impact of poor sound transmission due to leaks is minimized.

Folding doors and walls are particularly difficult to seal. For this reason, two folding units with a rather wide airspace in between are typically used.

10.7 COMPOSITE TRANSMISSION LOSS

Real partitions usually consist of several different parts—wall sections, doors, windows, and cracks or other openings. If one knows the transmission loss of the various parts, the *composite transmission loss* can be calculated using the expression:

$$R_{\text{comp}} = 10\log\left(\frac{\sum_i S_i}{\sum_i S_i \tau_i}\right) = 10\log\left(\frac{\sum_i S_i}{\sum_i S_i 10^{-R_1/10}}\right) \tag{10.14}$$

where S_i are the respective areas of the subelements and R_i are the respective transmission loss values in the frequency band.

Of course, close to a partition subelement having low sound transmission loss, the sound through that particular subelement will dominate, and the local NR will correspondingly be very low. Close to a keyhole, it is easier to hear what is going on in a neighboring room. The hole acts as a probe tube to the sending room and has little sound reduction.

10.8 CRACKS AND OTHER LEAKS

The transmission loss of a crack 1 mm wide or more in a hard panel is nearly 0 dB. One usually considers the crack opening as a separate partition part. By measuring the surface area of a fissure or crack, and using Equation 10.14, one can estimate the influence of the sound leaking in through the small opening on the composite partition.

Long holes tend to have smaller transmission loss than round holes for the same total surface area. Figure 10.13 shows an example of a large crack visually hidden by trim that caused a serious transmission loss problem between two offices.

Sound leakage by fissures around poorly installed drywalls is a common problem since it is easy to visually cover cracks by trim. The trim seldom makes an airtight seal, thus, sound will leak through, even though the fissure no longer attracts visual attention. The slit often used under doors for ventilation purposes is another example of an opening that seriously deteriorates the maximum sound insulation that can be obtained by a door. There are special sound-absorptive ventilation openings available that should be used if ventilation is desired in a similar way. One must also note that fissures are usually at the corners between walls.

As will be discussed in Chapter 14, the position of a sound source determines the coupling of sound from the source to the sound field in the room, since the radiation impedance changes between locations in rooms. A sound source that is located in the corner between three walls usually radiates sound



Figure 10.13 A major leak, between a lightweight concrete wall and the floor, that was hidden by wood trim. (Photo by Mendel Kleiner.)

better than one that is located at the corner between two walls, which in turn radiates more than one in the middle of a wall.

10.9 FLANKING TRANSMISSION

In most buildings, the vibration caused by sound incident on the walls, ceilings, and floors in a room is transmitted through a number of paths to other rooms as shown by Figure 10.14. The resulting transmission loss is determined by areas and acousto-mechanical properties of the various surfaces, and of course the mechanical properties of the joints. The transmission of sound by airborne or structure-borne paths, other than the direct path through the joining wall or floor, is called *flanking transmission*.

The importance of limiting the flanking transmission increases as the direct path transmission loss increases. The presence of flanking transmission is one important reason for the transmission loss of partitions in actual buildings being lower than that measured in the laboratory, where flanking transmission is carefully avoided.

The apparent transmission loss R_{field} may be calculated from:

$$R_{\text{field}} = L_{pS} - L_{pR} - 10\log\left(\frac{A_R}{S_p}\right) \tag{10.15}$$

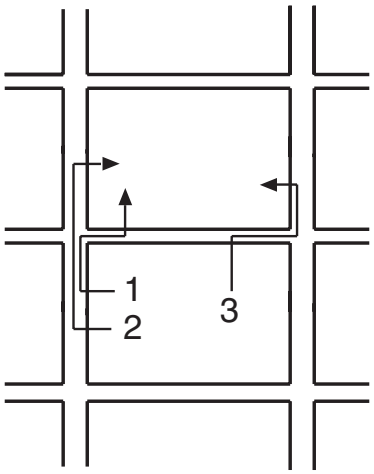


Figure 10.14 Paths 1–3 are the most important flanking sound transmission paths, thus, with four walls there will typically be at least 12 flanking sound transmission paths between two adjacent rooms in a building.

where L_{pS} is the sound level in the diffuse field in the sound source room, L_{pR} is the sound level in the diffuse field in the receiving room, A_R is the sound-absorption area in the receiving room, and S_p is the joint partition area between the two rooms, provided that it is larger than 10 m^2. If the partition area is smaller than 10 m^2, or if there is no joint partition area, this area is set to 10 m^2. Note that one refers to the same partition area irrespective of the path of the sound transport.

If the sound transmission between two rooms is over many paths, and one knows the apparent partial transmission loss for each path, one can calculate the composite transmission loss $R_{\text{field,comp}}$ using:

$$R_{\text{field,comp}} = -10\log\left(\sum_i 10^{-R_{\text{field,i}}/10}\right) \tag{10.16}$$

Since the same transmission area is referred to, this expression is just a simplification of Equation 10.4.

A special form of flanking transmission is the transmission over airborne paths, such as various leaks and crevices, as well as by way of air ducts for ventilation and heating. Occasionally, sound may be transmitted by the water pipes of heating systems.

Flanking transmission is often a problem in buildings using moveable lightweight walls when there are floating floors and suspended ceilings as indicated in Figure 10.15. Because of the mounting method, the structural vibrations in the floating floor will be subject to little *corner* transmission loss.

Flanking sound transmission through the airspace above the suspended ceiling is often considered by adding sound absorption to the suspended ceiling. The best way to limit the transmission of airborne sound is by extending the partition to include the volume over the suspended ceiling. However, even in this case, there may be flanking transmission due to sound pickup by the ventilation duct openings. Once the sound is inside the duct, there will be little attenuation of the sound unless silencers (in the canals or at the vents) are used to prevent transmission through the duct system. Even ventilation ducts without openings into a room can cause flanking transmission since the sound waves will transmit through the lightweight duct walls.

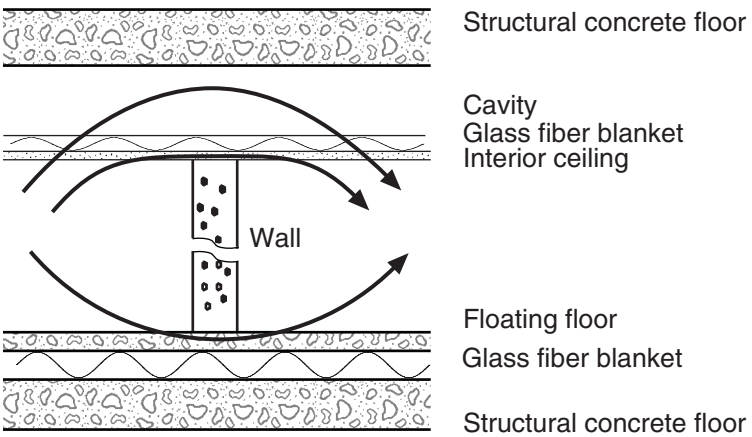


Figure 10.15 Flanking transmission paths for a floating floor and a resiliently suspended ceiling.

A particular problem of this type of flanking transmission is that the sound leaks reduce the privacy. The sound pressure level will be much higher close to a vent or a fissure than in the diffuse field of the room, making it easy to eavesdrop on conversations.

10.10 BALANCED SPECTRUM DESIGN

Any room will have background noise contributions from many sources—HVAC, traffic, neighboring rooms, and facilities. The *balanced spectrum design* approach calls for the contribution of the noise transmitted through a certain partition to contribute negligibly to the level in the receiving room. For continuous, random process noise, such as aerodynamic noise, this is a useful design approach. Figure 10.16 shows the design curves used in the discussion in the following paragraph.

Once a decision has been made on the allowable background noise levels in the receiving room—for example, using the octave band levels specified in Chapter 6—the design can start taking the properties of the sending room noise into account. In the figure, the sound level in the sending room is L_{pS}, and the sound level of the background noise in the receiving room is L_{pB}. The spectral shape of the masking noise will depend on its source; traffic noise (also as transmitted through windows), and ventilation noise (from, for example, air conditioning) have different spectral shapes.

Both noise reduction and transmission loss values typically grow with increasing frequency. The sound level of the background masking noise L_{pB} typically decreases with increasing frequency. However, many noise sources—such as radios, televisions, and musical instruments—have spectra that are not flat. They also have easily recognizable sound features making them irritating to listen to, even when they barely *breakthrough*. For such noise sources, it is advantageous that the transmitted sound level in the reverberant field is at least 10 dB below the masking noise, so as not to be audible.

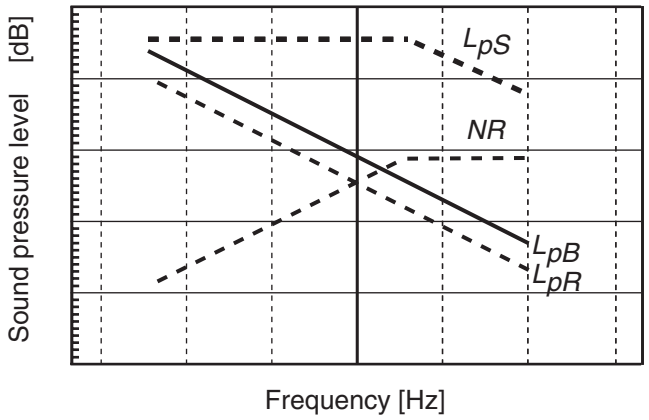


Figure 10.16 Spectrally optimized design approach to privacy by balanced spectrum design. The sound level of the transmitted noise should be below that of the masking noise at all octave band frequencies.

The design curve for the sound level due to the noise through the partition will then be for a sound level curve 10 dB lower, according to L_{pR} in the figure. The difference between curves L_{pS} and L_{pR} will then be the noise reduction target shown by the NR curve.

10.11 INSULATION AGAINST IMPACT SOUND

Often sound-radiating vibration is caused by forces, on the partitions of a building, other than sound. Forces on floors or ceilings are due to footsteps, rolling wheels, machinery, and such. However, heavy-duty, wall-mounted electrical relays, such as those used in elevator systems, and phenomena such as water hammer, may also cause impacts that are disturbing to dwellers.

A floor is usually constructed using several different layers in addition to the layer carrying the structural load. Typical extra layers are floating floors and carpets on the upper side, and sound-absorbing or isolating materials, either ceiling-mounted or resiliently suspended from the ceiling, on the lower side. All of these layers influence both the transmission loss and the sound radiation to the rooms, not only above and below the floor, but also to other rooms by the principle of flanking transmission.

The impact sound transmission is estimated by measuring the *impact sound pressure level* created in an adjoining lower room when using a tapping machine on the floor, as specified by the ISO 140-6 standard. The standardized tapping machine uses five weights, which consecutively drop on the floor 10 times per second, creating vibration in the floor and, of course, considerable sound levels in the upper room. The vibrations on the upper side of the floor are transmitted to the lower side of the floor and radiate sound into the lower, receiving room.

The *normalized impact sound pressure level* is adjusted for the sound absorption of the receiving room and is measured in the third octave frequency bands from 100 Hz to 3.15 kHz since the noise created by the tapping machine varies with frequency depending on the floor and its covering. A high impact sound pressure level indicates an acoustically poor floor construction.

The impact sound pressure level and spectrum created this way may not be relevant to the impact sound mechanisms in some cases, so care should be used when considering sufficient impact sound insulation. This particularly applies to the rating of the impact sound properties of lightweight floors, since the tapping machine uses masses that are much smaller than those of human feet and legs. The method also does not take into account the nonlinearities at the point-of-contact of the impact, which may be due to the elastic properties of, for example, the treatment layer.

Similar to the use of the weighted sound reduction quantity for airborne sound transmission, one uses a single-number quantity to characterize the impact sound transmission properties of floors and additional floor and ceiling layers. The calculation of these quantities, the *weighted normalized impact sound pressure level* $L_{n,w}$ [dB], and the *weighted impact sound improvement index* ΔL_w [dB] are described in the ISO 717–2 standard.

It is important to remember that a high value of $L_{n,w}$ corresponds to a poor construction. Values for $L_{n,w}$ and ΔL_w of typical floor constructions may be found at the various websites of materials manufacturers, research institutions, and testing organizations.

Most floor constructions, whether lightweight or heavyweight, have poor impact noise properties unless acoustically treated in some way. Concrete floor slabs have nearly a frequency-independent normalized impact sound pressure spectra, while lightweight timber and other floor constructions are plagued by high-normalized impact sound pressure at low frequencies (thumping noise).

Attempting to reduce the mass of floors by using concrete floors with cavities or by using precast φ-shaped or similar concrete elements, invariably leads to higher $L_{n,w}$ values than those of massive concrete slabs. Figure 10.17 shows data for a concrete slab and two types of surface treatments.

The low frequency dominant sound spectra obtained for impacts on lightweight floors are difficult to modify without adding more mass to the floor. Figure 10.18 shows an example of the measured normalized impact sound spectra of treated and untreated wood floors. One notes that lightweight wood floors tend to have severe low frequency noise problems unless treated with heavy ballast.

The poor impact sound properties of most floors makes it necessary to use various treatments, such as added layers on top or underneath of the floor, as mentioned previously. In the case of a concrete floor, common treatments consist of adding a resilient layer, such as thermoplastic tiles or carpets or other types of impact-reducing soft layers. More advanced treatments consist of cork or glass wool immediately on top of the concrete slab, covered by a sub-floor made of plywood, parquet, or wood strips. Such a construction is often called a *floating floor*. Since a floating floor is a double-panel partition, it exhibits a fundamental resonance, just as a double wall does. The resonance frequency is typically determined by the surface mass of the sub-floor and the compliance of the resilient layer or trapped air. Good floating floor installations require high-grade workmanship similar to that required in the case of avoiding flanking transmission.

Figure 10.19 shows an example of a multilayer floor characterized by significant high transmission and impact loss. Mechanical short-circuits, vibration transmission paths, as generated by water pipes, for example, must be eliminated for the floating floor to be effective in reducing the impact noise.

The fundamental resonance frequency of a floating floor can be estimated by Equation 10.17. When the frequency is well above the fundamental resonance frequency, the normalized impact sound pressure level improvement is approximately impact sound pressure level:

$$\Delta L = 40 \log\left(\frac{f}{f_0}\right) \qquad f_0 \ll f \ll f_c \tag{10.17}$$

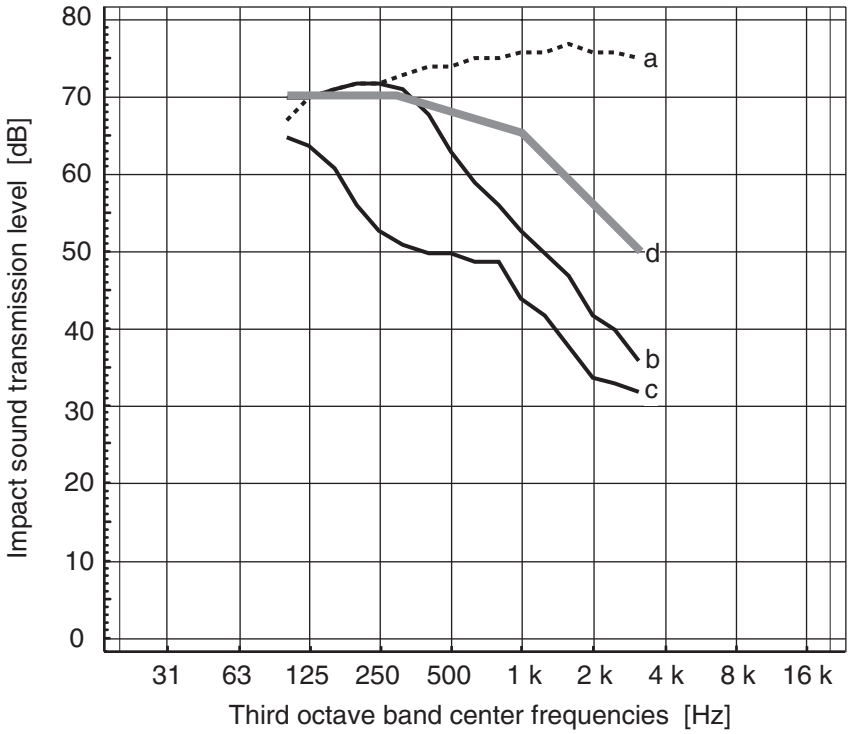


Figure 10.17 Examples of measured impact sound transmission levels for a 140 mm thick concrete slab and the influence of adding a 35 mm floating floor layer of cement on top of an elastic layer: (a) level for bare 140 mm thick concrete slab, (b) level using a 10 mm thick hard plastic elastic layer, (c) level using a 10 mm mineral wool elastic layer, and (d) reference curve used in calculating the weighted normalized impact sound pressure level. The reference curve is shifted in steps of 1 dB to allow sum of differences below measured curve (for example *a*) but above the reference curve to be ≤ 32 dB (at the third octave band frequencies from 125 Hz to 3.2 kHz). The $L_{n,w}$ value is then determined as 68 plus the number of decibels the curve has been shifted upward. (Downward shift subtracts decibels.)

above the *double-wall resonance* frequency given by Equation 10.18:

$$f_0 = \frac{\Lambda c_0}{2\pi}\sqrt{\frac{\rho_0}{m'' d}} \tag{10.18}$$

Here d is the thickness of the mineral wool layer and Λ a constant smaller than unity, which is due to the nonadiabatic propagation conditions for the sound waves inside the layer.

It is difficult to obtain sufficiently good impact sound insulation in houses using light floor constructions without adding mass, such as sand, stone, or some other type of ballast. Another common impact sound insulation problem is resonances in the stair landings. These must be isolated from the rest of the building using elastic supports.

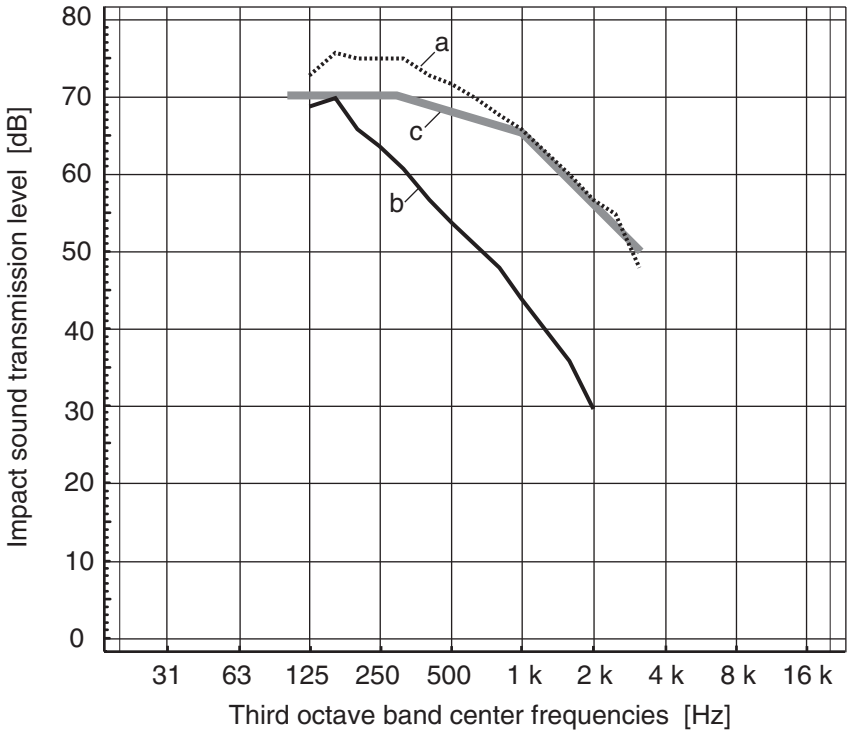


Figure 10.18 Examples of measurement results for impact sound transmission levels of a wood floor for two different treatments: (a) level without treatment, (b) level with treatment, using a particle board floating floor on sand, on top of a mineral wool blanket, and (c) reference curve used in calculating the weighted normalized impact sound pressure level.

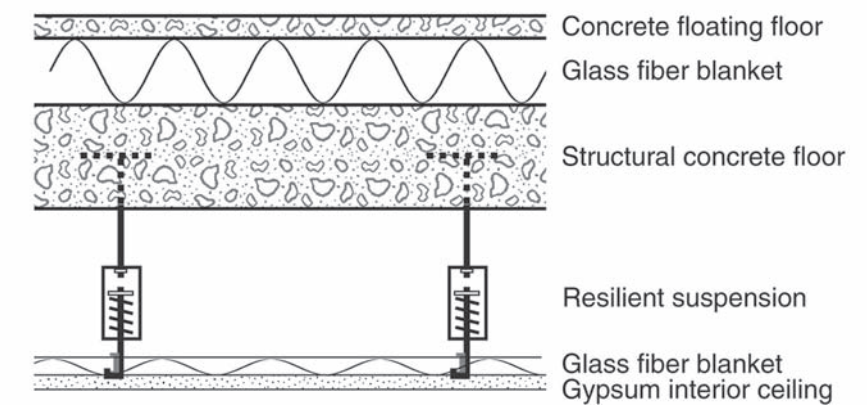


Figure 10.19 Example of a multilayer floor characterized by high transmission loss.

10.12 INSULATION AGAINST STRUCTURE-BORNE SOUND

As mentioned in Chapter 8, the loss factors of most construction materials are very low. This is a natural consequence of the desire for sturdy and stiff constructions. Commonly, the loss factors are below 10^{-2}. One often finds that the damping that exists in buildings is determined by edge losses, energy moving to another part of the structure, friction, and air pumping. This means that the vibrations injected by a vibrating machine into the building structure can travel far along the structure without being attenuated. The structure-borne sound can thus radiate disturbing noise in many places in a building. A well-known example is the noise in concrete buildings when someone is trying to drill holes using a hammer drill. Figure 10.20 shows the transmission and radiation paths of a typical installation.

Looking at the figure, one realizes that there are the following ways to reduce structure-borne sound and improve the sound insulation:

- Vibration insulation of the machine support so that injection of vibratory force into the floor is prevented
- Use of viscoelastic material in the partitions to attenuate the propagation of vibrations
- Use of vibration isolating mountings at corners to reduce transmission from one partition to another
- Radiation reduction layers used on top of partitions and the possible use of a floating floor construction
- Sound-absorptive materials utilized to reduce the level of the sound that has been radiated into the receiving room

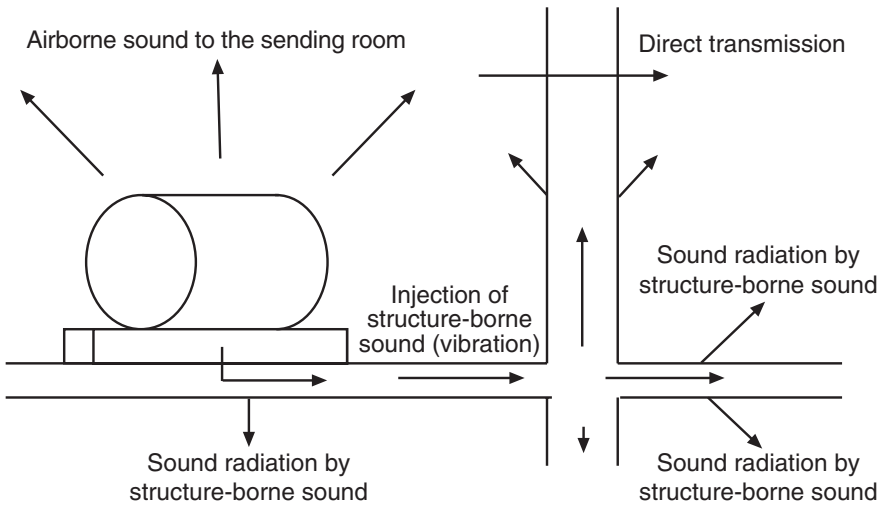


Figure 10.20 Excitation, transmission, and radiation of structure-borne sound.

Usually the sound is radiated by a bending wave field in one or more partitions making up the room. The bending waves have low mechanical impedance and are consequently easy to generate. The prerequisites for using viscoelastic layers to attenuate the propagation of structure-borne sound were discussed in Chapter 8.

It is necessary to differ between resonant and nonresonant waves in the reverberant vibration field in a panel or slab. The nonresonant (direct) wave is primarily attenuated due to geometrical spreading of energy. The vibration level will, in the two-dimensional case, diminish with distance r according to the distance law as $10\log(1/r)$—that is, at a lower rate than in a three-dimensional case. As shown in Chapter 8, the attenuation due to internal losses is approximately -13η dB/λ_B, which should be added to the geometrical attenuation. One notes that the loss factor η must be very large to have any appreciable influence on the propagation of the direct wave.

The resonant bending wave field is dependent on the loss factor. Analogous to the case of airborne sound, one finds that the level of the resonant vibration field is proportional to $10\log(1/\eta)$ and to the total area of the panel. An increase in loss factor η from 10^{-2} to 10^{-1} will therefore result in much lower resonant bending wave vibration levels over the entire panel, except in the direct field of the vibration source.

In most buildings, it is quite impractical to use single-sided application of viscoelastic materials. Constrained viscoelastic layers are generally more practical than single-sided layers in buildings (see Chapter 8). They can be applied in both concrete and in lightweight fiberboard and gypsum panel constructions. It is often not necessary to cover the entire panel area with the viscoelastic compound or glue.

The best approach is to use the damping layer, or other damping measure, close to the place where the vibrations are induced in the construction. Not only does this result in lower cost, it also reduces the wave conversion at edges and the generation of wave types that are harder to attenuate than bending waves.

In some cases, it is possible to use suspended ceilings or floating floors mounted on special compliant isolators. This is equivalent to the use of a double wall or a resilient skin.

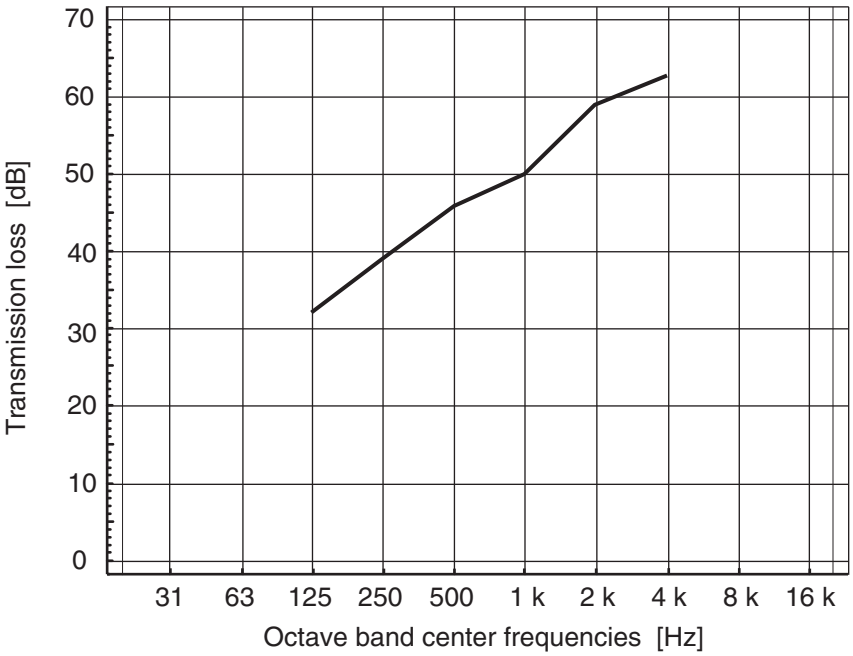
10.13 PROBLEMS

10.1 A wall has an area $S = 15$ m^2, a thickness $h = 0.16$ m, and is made of concrete. There is a door having an area $S_d = 2.2$ m^2 in the wall. The transmission losses of the concrete wall and of the door are 53 dB and 40 dB respectively at 500 Hz.

Task: Calculate the resulting transmission loss of the wall and door combined.

10.2 A television set is radiating noise into a room in the frequency range 125 Hz to 4 kHz. Assume the sound pressure level to be $L_p = 80$ dB in all the octave bands in this range on page 254. The wall to an adjoining bedroom has the transmission loss characteristic shown in the graph. The dimensions of the rectangular bedroom are $3.5 \times 3.5 \times 2.6$ m^3; its reverberation time is $T = 0.5$ s in this frequency range.

Task: Calculate the sound level in dBA in the bedroom due to the television set if there are no other noise sources. Only consider direct transmission.



10.3 An apartment consists of two adjoining rooms. The larger room has a volume of 75 m^3 and a reverberation time of 0.5 s. The smaller room has a volume of 24 m^3 and a reverberation time of 0.5 s. One person, *A*, wants to be undisturbed by the noise from a television set that the other members in the family insist on watching. Assume all persons to be in the reverberant field.

Task: Which solution is best from a noise viewpoint?

a. To put the TV in the smaller room and *A* in the larger room.
b. To put the TV in the larger room and *A* in the smaller room.

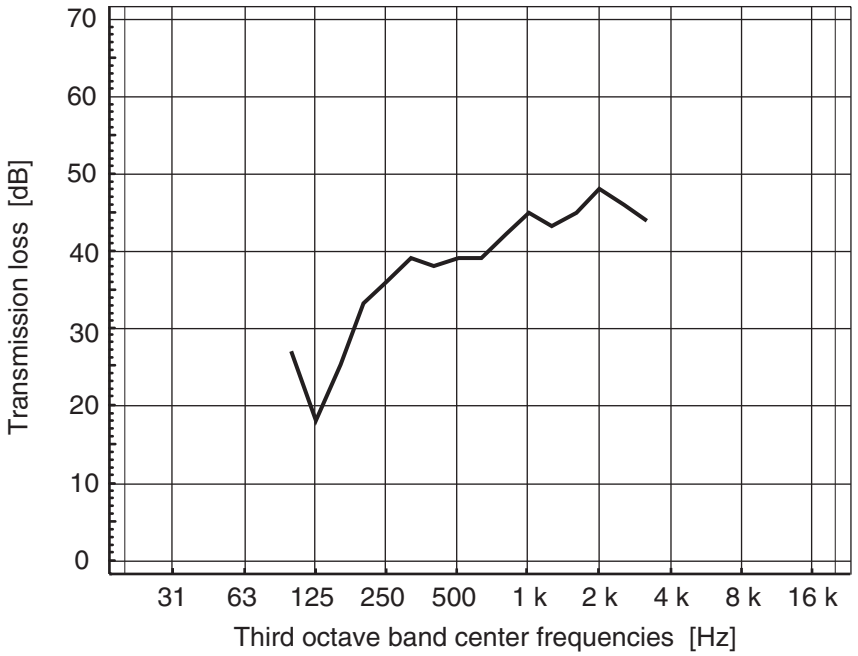
Assume that:

1. The TV set's sound power output is the same in both cases 1 and 2 (same volume setting).
2. The TV viewers use the same sound pressure level irrespectively of where they are sitting.

10.4 The transmission loss of a particular drywall is given by the curve in the graph on the next page. The wall is a double-panel construction using 9 mm and 13 mm gypsum panels. The panels are separated by 75 mm wood studs at 0.6 m intervals. The air gap is filled with glass wool. The panels have the following data: ρ_{gypsum} = 840 kg/m^3, $K_{c,gypsum}$ = 32–40 m/s.

Tasks:

a. Calculate the fundamental resonance frequency and the critical frequency of the construction and explain the appearance of the measured curve
b. Suggest suitable changes to the wall construction to improve its transmission loss.



10.5 Installing office equipment often results in the need for cooling. A high volume copier is to be installed in a room (Room 1) neighboring an adjoining room (Room 2). Rooms 1 and 2 have the following data: The reverberation times are 1.5 s and 0.5 s respectively and the volumes are 100 m^3 and 40 m^3 respectively. The area of the wall between the rooms is 10 m^2.

The sound power level of the cooling equipment installed in room 1 as well as the transmission loss of the wall, are listed in the table.

Octave band	125	250	500	1 k	2 k	4 k	[Hz]
L_W	74	76	84	86	89	80	[dB] re $1 \cdot 10^{-12}$ W
R'	18	35	40	45	50	40	[dB]

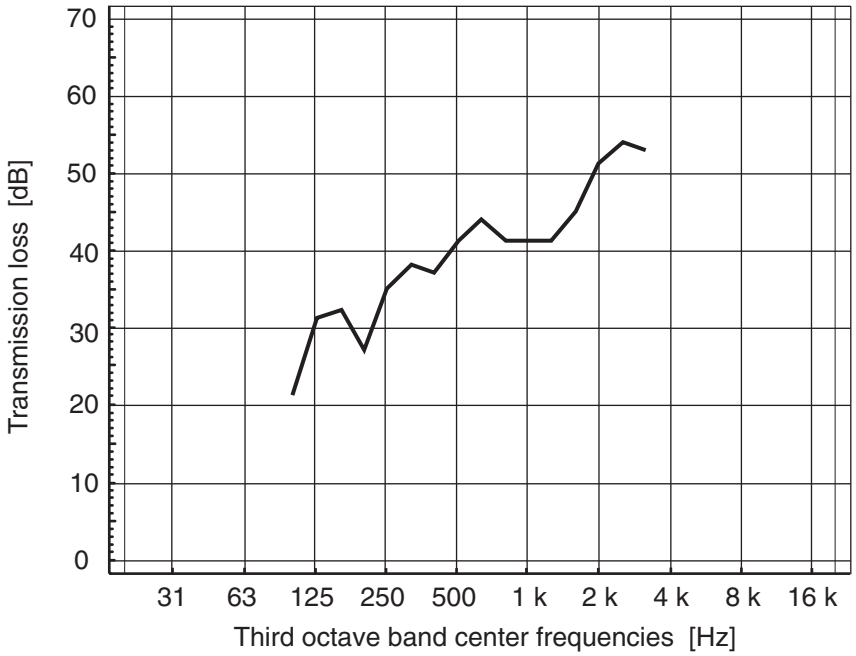
Task: The maximum allowable sound level in the adjoining room is 45 dBA. Calculate the sound level in Room 2 and decide whether this requirement has been fulfilled.

10.6 It is often difficult to avoid resonances that reduce the transmission loss when designing double pane window constructions. The transmission loss curve in the graph on page 256 shows several transmission loss dips at 100, 200, and 400 Hz.

The window has:

Glass pane thicknesses: 10 mm and 4 mm
Glass pane surface dimensions: 0.83 × 1.80 m^2
Glass density: 2,500 kg/m^3
Glass coincidence number: 12 m/s
Air gap: 56 mm

Task: Investigate the possible reasons for these dips by calculating the transmission loss of the window.



Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Vibration Isolation 11

11.1 INTRODUCTION

Vibration isolation is the name given to a number of techniques for reducing the transmission of vibration between generating and receiving mechanical structures. In audio engineering, we are interested primarily in two tasks—isolation of force and velocity:

1. *Force isolation* means that we try to reduce the force by which a device, such as a machine or loudspeaker, is acting on its support structure.
2. *Velocity isolation* means that we try to prevent a machine from shaking, in spite of its being mounted on a vibrating base. An example of velocity isolation is mounting hard disk memory or other electronics parts on a vibrating structure, such as a car or floor.

The devices used for achieving isolation are called *vibration isolators*. Vibration isolators need to have high mobility to be effective. Typically, isolators are soft springs of steel or brass, glass fiber pads, or air-filled cushions (for example, air cavities or sheets of glass wool). A simple vibration isolation system used in a phonograph turntable is shown in Figure 11.1.

The metal cross subchassis that will be holding the platter and tone arm is a steel structure suspended by three steel springs and some rubber bushings. The three spring systems are placed on the foundation (for example, a table). The aim is to prevent floor or wall vibration due to footsteps, as well as vibration induced by loudspeaker sound, from being injected into the steel structure.

There are also *alternate ways* of reducing vibrations, such as the *addition of mass* or *stiffness*, as well as the use of *tuned mechanical resonators*, also known as *dynamic absorbers*. These will not be dealt with here.

11.2 CLASSICAL VIBRATION ISOLATION THEORY

Vibration isolation is usually achieved by *creating a mismatch* between the mechanical impedances at the connections of the two structures. The impedances can be those of both translational and rotational motion. The generating and receiving structures generally have a *high mechanical impedance* (low mechanical mobility).

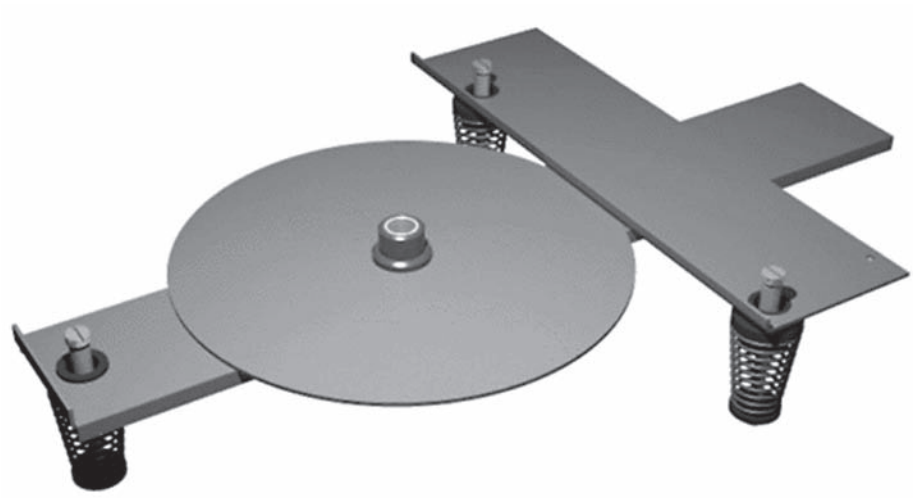


Figure 11.1 A subchassis suspended on steel springs can frequently be used to achieve good vibration isolation for a phonograph at medium and high frequencies.

The impedances of the two structures are often of the same magnitude, which may lead to a strong exchange of vibratory energy between the structures. The exchange can be reduced by inserting an impedance mismatch that will lead to vibration reduction.

For reasons of static stability, most equipment is fastened to its base at several points at some distance apart. There will then be an interaction between the vibratory motion at the various points. In *classical vibration isolation theory*, the influence of such interaction is neglected. Classical theory assumes that there is a discrete impedance, for example, due to a mass attached to a foundation by one or more discrete springs and dampers under the following conditions:

- The motion of the machine is one-dimensional, translational, and perpendicular to the base.
- The foundation has infinitely high mechanical impedance.
- Frequencies are low enough for the source of vibration to be considered a discrete element.
- Frequencies are low enough for the spring elements to be considered as discrete massless springs and for mass elements to be considered inelastic.

These conditions are often approximately fulfilled at low frequencies, and when dealing with small vibratory sources. The theory can then be used to estimate the fundamental resonance frequency of the single degree of freedom vibration isolation system.

It is quite feasible to extend this vibration isolation theory to handle all six degrees of motional freedom (three translational and three rotational directions of motion). This text will, however, only give a short introduction to the topic primarily as it can be applied to audio equipment and associated electronics and will be limited to a single degree of freedom systems.

The theories and methods described here further assume the systems have linear characteristics—that is, only relatively small vibrations are studied. It is common for vibration isolation systems to have *nonlinear properties*. An extreme form of nonlinearity is that used to limit vibration displacement. Many

isolation devices, such as rubber pads, have nonlinear characteristics also in their working range. There are also *active vibration isolators* that use electrically variable components to control vibration. These can be designed to interfere with the vibration or to simply change the dynamic properties of the devices in the system, such as effective mass or damping.

11.3 IMPEDANCE AND MOBILITY

For medium and high frequencies, the mechanical systems (machine, isolator, and base) are best analyzed using the *mobility theory*. The *mechanical mobility* $\underline{Y}_M$ is defined as the inverse of the mechanical impedance $\underline{Z}_M$, which in turn was defined in Chapter 2. The theory assumes that the mobilities can be regarded as complex. One can then use the methods of *electromechanical analogies* and electrical circuit theory to find engineering solutions to complicated mechanical systems.

At low frequencies, many devices such as hi-fi/stereo equipment can be regarded as discrete *masses*. This means that the internal mobility can be approximated by $\underline{Y}_M = 1/j\omega M$ where M is the mass of the unit. However, even at a few hundred Hz, internal resonances will lead to the internal mobility having approximate frequency independent characteristics at the equipment supports. These characteristics are similar to those of the foundations.

Springs are the most common isolator component. The mobility of an ideal spring depends on the spring constant k_I that is given by the ratio of applied force $\underline{F}$ to net length change $\underline{x}$ is $k_I = \underline{F}/\underline{x}$. Using the jω-method, we find that the spring's mobility is $\underline{Y}_M = j\omega/k_I$, since velocity u is related to displacement as $\underline{u} = j\omega\underline{x}$. Any actual spring will be characterized by internal losses. Steel springs have small internal losses whereas rubber, cork, and glass wool springs are characterized by greater losses. Because of the small losses, steel springs have internal resonances that, unless damped, will limit usefulness. At high frequencies, there may even be wave motion in the spring. Figure 8.3 illustrates how such wave motion can be used for special purposes. In many practical cases, it will be necessary to increase losses by adding extra damping to systems using steel springs (for example, the system shown in Figure 11.1). Some spring isolators are made of rubber, which is used in a shear mode, to combine softness and losses.

Components mainly characterized by internal losses are called *dampers*. Viscous dampers, such as air dampers, are common, as are oil-based dampers, but dry friction dampers made of felt are also used. The mobility of dampers can be written as $\underline{Y}_I = r_I$ where r_I is the mechanical admittance of the damper.

In using the classical vibration isolation theory, we often assume that the foundation supporting the system to be vibration isolated has zero mobility. Any real support will have some local compliance at the support points and at medium and high frequencies, the internal mobility will be real. At frequencies, well above the bending wave resonances of a limited size sheet having bending stiffness, the point mobility of the sheet will approach that of an infinite sheet—that is, $\underline{Y}_F = 1/(8\sqrt{(m''B')})$ as discussed in Chapter 8. Equipment to be vibration isolated should be placed over major floor beams or on a concrete slab or solid ground. Lightweight wood floors are problematic from the viewpoint of vibration isolation. In some cases, extra mass, such as sand or stone, can be added to such floors to reduce their mobility. Sometimes stone or cement blocks can be used successfully on the furniture or floor, holding the equipment to be isolated, to reduce the apparent foundation mobility.

11.4 SOME METRICS FOR VIBRATION ISOLATION

Since there are different aims for vibration isolation, several different metrics have been developed to estimate the efficiency of the techniques used. Here we will only study metrics used in applications where linear translational movement is involved.

Two sets of indices will be used, one to denote the structures and one to denote the case. Index *0* is used to denote the free velocity or unblocked force of the source and index *2* is used to indicate the receiving structure. Index *u* is used to denote the case of no vibration isolation (stiff mounting), and index *i* is the case of a vibration isolation technique or device being used.

One common vibration isolation metric is the ratio of transmitted force to the force acting on the vibrating device to be isolated, called the *transmissibility T* which is defined as:

$$T = \left|\frac{\underline{u}_2}{\underline{u}_0}\right| = \left|\frac{\underline{F}_2}{\underline{F}_0}\right| \tag{11.1}$$

Another metric is *efficiency E*, defined as:

$$E = \left|\frac{\underline{u}_{2u}}{\underline{u}_{2i}}\right| = \left|\frac{\underline{F}_{2u}}{\underline{F}_{2i}}\right| \tag{11.2}$$

An important metric for the performance of the vibration isolation is *insertion loss* in dB, ΔL, or *IL*, defined as:

$$\Delta L = 20\log(E) \tag{11.3}$$

Typically, we are more interested in the insertion loss than in the transmissibility since the insertion loss number gives us practical useful information about the success of our vibration isolation efforts. Both efficiency and insertion loss are related to the vibration in the practical case, taking into account the vibratory load on the structure.

11.5 LINEAR SINGLE-DEGREE-OF-FREEDOM SYSTEMS

The Simple Mass-spring System

Figure 11.2 shows a very simple system that idealizes a situation often found in practice. A loudspeaker is mounted in a car door. The loudspeaker consists of a diaphragm and voice coil combination, magnet, and basket (see Chapter 14). In practice, we can consider the diaphragm and voice coil combination as having little mass compared to the basket and magnet; it can be neglected in the further analysis here.

Let us study a device that is to be isolated. It is considered to be a rigid body (mass) *M* that is acted upon by a force. The vibrating device is connected to the foundation (the floor) by an isolator. Figure 11.3 shows an example of a commonly used practical steel spring isolator having a spring constant k_i. (Note the rubber pad, which increases insertion loss at high frequencies.) The isolator may contain compliance and damping but is assumed to be without mass. The foundation is also assumed to be rigid and to have infinite mass. The system is assumed only to vibrate in one direction. Such a system is called a linear *single-degree-of-freedom system* (often written SDOF system).

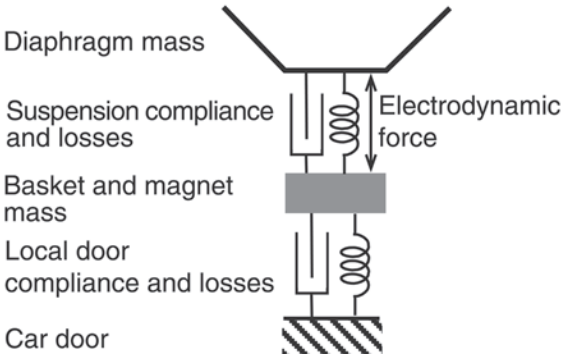


Figure 11.2 An electrodynamic loudspeaker mounted in a car door can be thought of as an example of a simple one-dimensional vibratory system. The diaphragm mass is small compared to the basket mass. The lower three components in the drawing can be considered parts of a single-degree-of-freedom system.

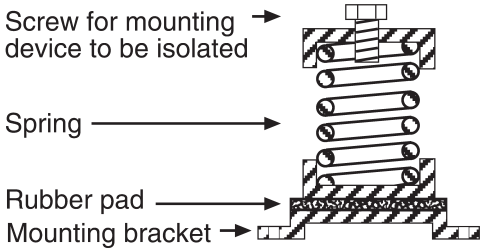


Figure 11.3 A metal-spring machinery isolator combining a steel spring with a rubber pad.

A transient force (or a momentary system displacement) will result in a damped oscillation by the mass as shown in Figure 11.4.

The natural frequency of the resonance for the undamped system will be:

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{k_i}{M}} \tag{11.4}$$

The transmissibility of a simple SDOF system is shown graphically in Figure 11.5—for some different values of viscous damping. (Other types of damping, such as friction, will give different results.) At critical damping, there is no oscillation as the mechanical system loses energy, as shown in Figure 11.4. Note, however, that the transmissibility curve, shown in Figure 11.5, still has some overshoot at the resonant frequency.

The curves in Figure 11.5 are misleading, however. Their simplicity has been achieved by unrealistic simplification. The vibrating device cannot usually be assumed to be a rigid body and thus has internal resonances. The spring has distributed mass, the damper has mass, and the receiving foundation is not solid, but has many resonances.

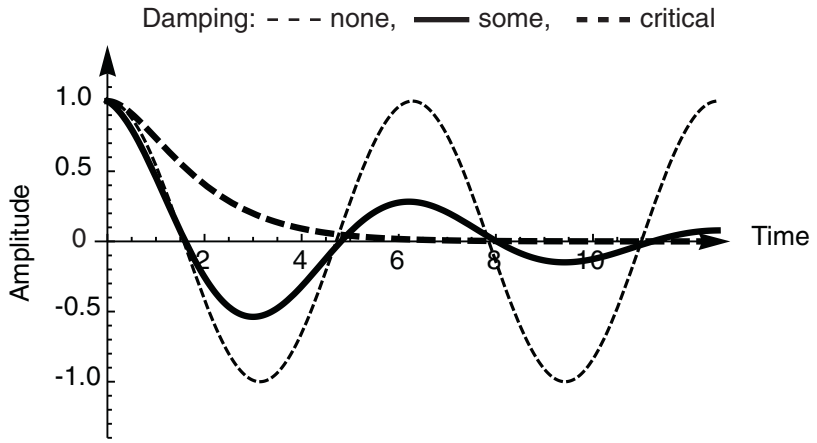


Figure 11.4 Natural vibration of a single-degree-of-freedom system deflected and released without velocity for different cases of damping.

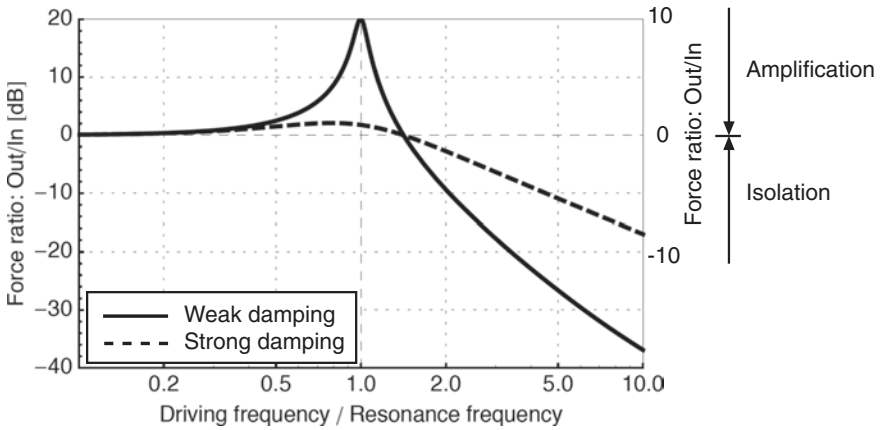


Figure 11.5 Example of the theoretical force transmissibility for a viscous damped single-degree-of-freedom system for two cases of damping.

The attraction of the theory, in spite of these serious problems, lies in that it is intuitive and gives reasonable results for many systems under a certain combination of conditions: a lightweight vibrating body, a relatively high mass foundation (compared to the mass of the vibrating body), and low frequencies, close to and below, the resonance frequency.

Using a more advanced theory, which allows representation of the more exact mechanical properties, we can estimate the response of the system at higher frequencies and for more realistic situations. It also allows us to understand why our vibration isolation efforts seldom lead to more than typically 15 to 20 dB

of insertion loss. The more advanced approach uses the theory of electromechanical analogies and can be expanded to multiple-degree-of-freedom systems.

11.6 VIBRATION ISOLATION THEORY USING ELECTROMECHANICAL ANALOGIES

Mechanical and Acoustical Analogies

Electromechanical analogy uses the similarity between the differential equations of electrical circuits and mechanical systems to enable the use of electrical symbols for mechanical components. Applying this approach, one can use the techniques of electrical circuit theory to study such things as transfer functions, impedances, and use circuit theorems, etc., to solve for the desired behavior of the mechanical system. One can also extend the method to include acoustical components. There are two types of commonly used analogies—impedance type and mobility type.

Acoustical analogies are usually of the impedance type and *mechanical analogies* of the mobility type. Techniques exist for the conversion between the two types and for interconnecting electrical, mechanical, and acoustical circuits. Reference 11.6 gives a summary of the principles of electromechanical-acoustical circuit analogies. Table 11.1 shows how the two types of analogies model mechanical systems.

Preventing Force from Acting on a Foundation

We will briefly return to the simple mechanical linear single-degree-of-freedom system discussed previously, but now use the analogy approach.

Start by studying a vibratory system without any vibration isolation, as shown in Figure 11.6. The rigid, vibrating body is represented by a mass M, initially mounted directly on a foundation having high internal impedance $\underline{Z}_f$. We also assume that the mechanical impedance of the mass, over the entire frequency range modeled, is much smaller than that of the foundation, that is, $\omega M << |\underline{Z}_f|$. This can also be expressed by saying that the mechanical mobility of the rigid body is much higher than that of the foundation. There is an internal force $\underline{F}_0$ acting on the mass.

Table 11.1 Substitution of mechanical variables by electrical variables for impedance and mobility systems

Impedance type analogy quantity	Mechanical system quantity	Mobility type analogy quantity
Electrical voltage	Mechanical force	Electrical current
Electrical current	Mechanical velocity	Electrical voltage

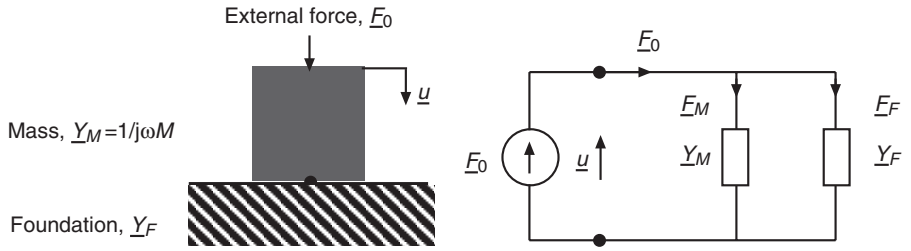


Figure 11.6 A mechanical system consisting of a mass mounted rigidly on a foundation. On the right, the system's electrical circuit analogy using the mobility approach.

Since the mass is mounted directly on the foundation, one realizes that both vibrate with the same velocity. According to Newton's first law, much more force is needed to move the foundation than the mass, since $\omega M \ll |\underline{Z}_f|$. As the mobility of the rigid body is much higher than that of the foundation, the foundation will pick up nearly all of the force internally applied to the rigid body—that is, $\underline{F}_{2u} \approx \underline{F}_{0.}$

We now mount an isolator, in this case just an idealized spring and damper combination, between the rigid body and the foundation as shown in Figure 11.7. One can then draw an electromechanical mobility type analogy as shown.

Now assume that we do not know what the impedances or mobilities are of the rigid body, the isolator, or the foundation. We can still use the same symbolic approach previously used and calculate the resulting force acting on the foundation since the *force currents* will split according to Kirchoff's first law:

$$\underline{F}_{2i} = \frac{\underline{Y}_M}{\underline{Y}_M + \underline{Y}_F + \underline{Y}_I} \underline{F}_0 \tag{11.5}$$

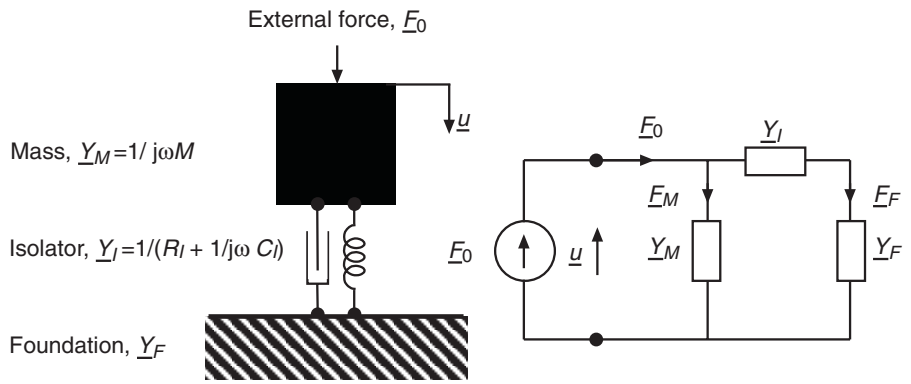


Figure 11.7 A mechanical system with a vibration isolation device inserted between source and foundation. On the right, the system's electrical circuit analogy using the mobility approach. The resulting vibration velocity of the mass is $\underline{u}$.

Using the electrical circuit diagram, we can now calculate the insertion loss as:

$$\Delta L = 20\log\left(\left|\frac{\underline{F}_{2u}}{\underline{F}_{2i}}\right|\right) = 20\log\left(\left|\frac{\underline{Y}_M + \underline{Y}_F + \underline{Y}_I}{\underline{Y}_M + \underline{Y}_F}\right|\right) \tag{11.6}$$

The isolator impedance is $\underline{Z}_I$ and its mobility $\underline{Y}_I$. The mechanical compliance of the spring is $C_I = 1/k_I$, where k_I is the stiffness of the spring:

$$\underline{Y}_I = \frac{1}{\underline{Z}_I} = \frac{1}{R_I + \dfrac{1}{j\omega C_I}} \tag{11.7}$$

One realizes that if the mobility of the isolator is high in comparison to the mobilities of the vibrating body and the foundation, the insertion loss will be:

$$\Delta L \approx 20\log\left(\left|\frac{\underline{Y}_I}{\underline{Y}_M + \underline{Y}_F}\right|\right) \tag{11.8}$$

We also immediately notice from Figure 11.7 that the prerequisite for obtaining a substantial insertion loss is that the mobility of the isolator is high compared to the mobility of the foundation. The resulting force (mobility type analogy current) will then be low.

We further note that another prerequisite for high insertion loss is that the mobility of the foundation be low compared to the mobility of the vibrating body. This is often difficult to achieve. In many real systems, the mobilities of the vibrating body and its foundation are of the same order of magnitude.

Now let us assume that the foundation is rigid (that is, $\underline{Y}_F = 0$), which means that $\underline{Z}_F$ is infinitely high. The equation for the insertion loss 11.6 then becomes:

$$\Delta L = 20\log\left(\left|\frac{\underline{Y}_M + \underline{Y}_I}{\underline{Y}_M}\right|\right) \tag{11.9}$$

We realize that, even in this case, it is necessary for the mobility of the vibrating body to be much lower than that of the isolator to achieve high insertion loss by applying vibration isolation.

The system is at resonance when the imaginary part of the mobility of the vibrating body is the same as that of the isolator but having the opposite sign—that is, if $\mathrm{im}[\underline{Y}_M] = -\mathrm{im}[\underline{Y}_I]$. At resonance, the system can become *matched*, and the insertion loss can, in fact, become negative.

If the real parts of the component's mobilities are low, the system will be poorly damped, resulting in an increased response close to the resonance frequency as indicated in Figure 11.3. High damping results in a poorer insertion loss at frequencies well above the resonance frequency than low damping does.

For a system with a force spectrum containing frequencies both around and over the resonance frequency, the choice of damping will depend on the force spectrum's shape and on other limits, such as the maximum allowable displacement of the vibrating body. For situations involving a force that is swept slowly in frequency across the region of resonance, the choice of damping will be particularly important.

Preventing Body Vibration Due to Vibration of the Foundation

Many practical cases of vibration isolation involve preventing a body mounted on the foundation from vibrating. For instance keeping a turntable for playing vinyl records, a computer hard disk, or

a CD/DVD player that is positioned on furniture from picking up floor vibrations due to steps or vibration due to loudspeaker excitation. This is sometimes called velocity isolation in contrast to the force isolation discussed previously.

We now regard the vibration velocity or displacement as given, and we are primarily interested in the resulting velocity or displacement of the attached body. As previously mentioned, we draw the mechanical mobility analogy as shown in Figure 11.8.

We see that the velocity insertion loss achieved by using the isolator will be:

$$\Delta L = 20\log\left(\left|\frac{\underline{u}_{2u}}{\underline{u}_{2i}}\right|\right) = 20\log\left(\left|\frac{\underline{Y}_M + \underline{Y}_I}{\underline{Y}_M}\right|\right) \tag{11.10}$$

We see that in this case the expression for the insertion loss will be the same as that obtained for force isolation with a rigid foundation.

11.7 REAL SYSTEMS

Using measured mobilities for the body, isolator, and foundation, we are in a better position to estimate the practically obtainable insertion loss that can be obtained by vibration isolation of a particular system. In reality, all three components are likely to be multiresonant systems with complex and very frequency-dependent mobilities.

The systems studied until now have been very idealized, also, because they have involved linear one-dimensional motion only. In a real system, there will be vibration, both linear and rotational, in each of the three directions as well as coupling between the attachment points. This makes most vibration isolation tasks difficult if high insertion loss is desired. Some possibilities for wave motion of a phonograph subchassis and the foundation on which it is placed are shown in Figure 11.9. It is important to not overreact to the high loss data shown in Figure 11.3, which applies to the idealized SDOF system case only.

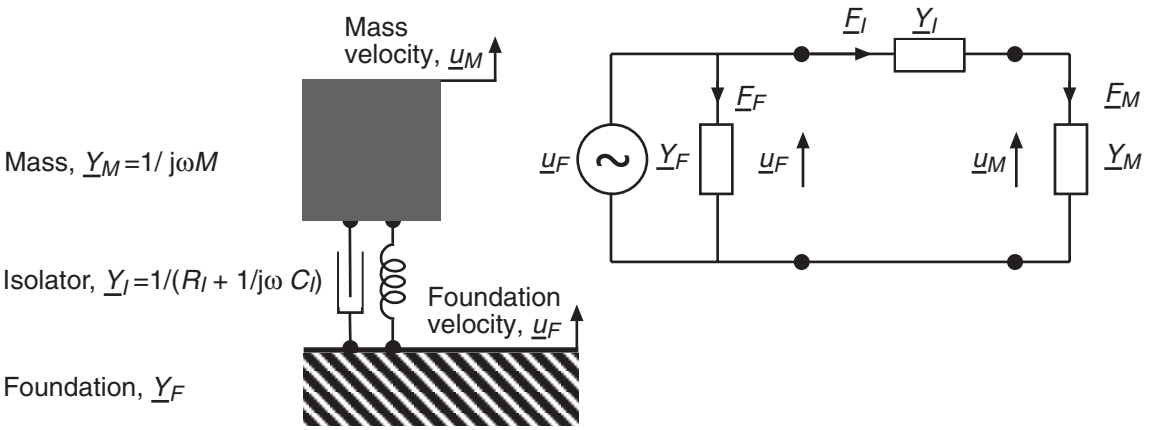


Figure 11.8 *Velocity* isolation and the electrical circuit analogy of the system using the mobility approach.

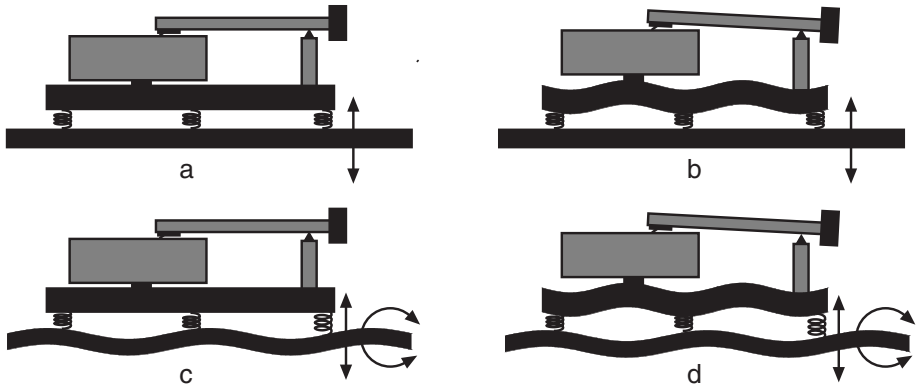


Figure 11.9 Some possibilities for vibration in a foundation and in an attached phonograph subchassis: (a) solid foundation and solid subchassis, very low frequencies, unidirectional translational movement in both, (b) solid foundation but bending wave motion in subchassis, medium frequencies, (c) bending wave motion in foundation but solid subchassis, medium frequencies, and (d) bending wave motion in both foundation and subchassis, high frequencies.

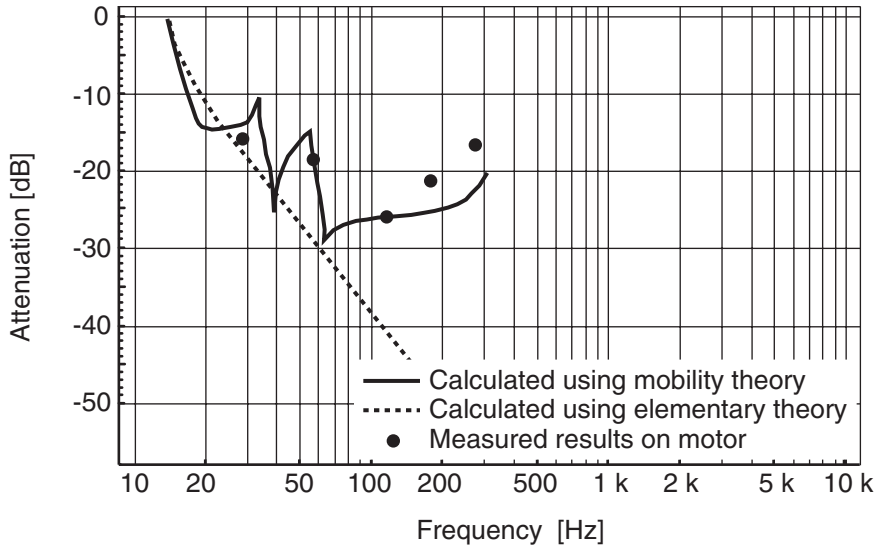


Figure 11.10 Predicted and measured attenuation by an isolator. (Adapted from Ref. 11.2)

At frequencies where the isolator is large compared to the wavelength in the isolator, there will be a possibility for half-wave transmission. This will reduce the insertion loss of the isolator.

Figure 11.10 shows predicted and measured attenuation of a SDOF system using both the simple theory initially described and mobility theory and data. The measurement data shows that the mobility approach gives a much better estimate of the attenuation (see reference 11.2).

11.8 PROBLEMS

11.1 Show that for a vertical linear spring carrying a mass load, with system movement purely vertical, the resonance frequency can be written as a function of the gravity induced static deflection d_{stat}.

11.2 A phonograph is mounted on an inertia base of stone for vibration isolation to give the phonograph added mass. The rectangular stone base (0.5×1 m^2) is resting on a sheet of mineral wool. The air trapped inside the mineral wool has stiffness that may exceed the static stiffness of the mineral wool itself. The stone base has a thickness of 2.5 cm as does the glass wool sheet before the stone base is laid on top. The static stiffness of the glass wool sheet is $1 \cdot 10^4$ N/m and the stone has a density of $2.5 \cdot 10^3$ kg/m^3.

Task: Calculate the resonance frequency for the stone base and the glass wool mass-spring SDOF system, assuming the foundation has low internal mobility.

11.3 The phonograph of Problem 2 is now mounted on a concrete floated inertia base that is supported by three steel springs. Each steel spring has a static stiffness of $12 \cdot 10^3$ N/m and a dynamic stiffness that is 90% of the static stiffness. The springs are placed so that they have the same static load.

Task: Calculate the resonance frequency of the mass-spring system if the springs compress by 10 mm when the concrete base is applied on them.

11.4 It is important that any velocity vibration isolation is designed to have its resonance frequency well below that of the maximum sensitivity of the supported structure—for example, a CD player. The CD-player contains a servo system to control the motion of the lens focusing the laser used to read information off the CD. The CD rotates at most at 10,000 rpm. Estimate the highest allowable vibration isolation resonance frequency.

11.5 In some vibration isolation systems, it may not be possible to control the Q-value of the resonant system because damping also reduces the insertion loss above the resonance frequency. The force transmission at resonance can then be reduced by the use of a parasitic resonant system, a *dynamic absorber*. Discuss the dynamic similarities between this and the ported box loudspeaker system shown in Figure 14.16.

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Microphones 12

12.1 INTRODUCTION

The ideal microphone is an *electroacoustic transducer* that can convert the waveform of sound pressure or sound pressure gradient to an electrical signal having the identical waveform.

Because of physical and engineering limitations, such conversion is not possible. A practical microphone will feature nonlinearities and self-noise. Because any practical microphone is also a mechanical device, microphones will be sensitive to vibration, which will give electrical output as well. Many microphones also contain electronic signal amplification and other signal enhancement circuits. These also contribute to the nonlinearities and the background noise of the microphone. Digital microphones use analog microphone capsules but incorporate a built-in analog-to-digital converter (ADC) so that the output signal is digitized and coded before being transmitted from the microphone. Such converters can, of course, add all the errors and noise types present in any system (see Chapter 16).

12.2 DYNAMIC RANGE, FREQUENCY RESPONSE, NOISE, AND DISTORTION

Any physical device has a limited *dynamic range*. A microphone's dynamic range is the ratio between the strongest and the weakest sounds that the microphone can faithfully reproduce. This ratio is usually expressed as a level difference in units of dB. It follows that the figure for the dynamic range is somewhat arbitrary; it depends on the background noise and maximum nonlinearities considered acceptable.

All microphone systems are characterized by *distortion*. We differ between two types of distortion, linear distortion and nonlinear distortion. *Linear distortion* is any distortion of the waveform that does not create new frequency components. The frequency-dependent amplification of an amplifier and the frequency-dependent attenuation by an acoustic or electric filter are examples of linear distortion. We usually expect a microphone to have *linear frequency response*, which is the absence of linear distortion in the audio frequency range—between 20 Hz and 20 kHz—often called a *flat* response. Depending on the nature of the acoustic signal and the engineering principle of the microphone, it may at times be necessary to add compensating linear distortion, for example, in the microphone amplifier or mixing console. In addition, the microphone response will be directional, influenced by diffraction around the microphone diaphragm and housing, acoustic resonances in the air behind grilles and such. The irregularities may not be much of a problem in dedicated applications, such as telephone systems, where

the frequency response of the microphone can and has to be optimized for criteria such as speech intelligibility or signal-to-noise (S/N) ratio.

Nonlinear distortion is distortion that creates new frequency components being added to the signal. Examples of such distortion are the distortion due to limited movement of the microphone or loudspeaker diaphragm suspension, the distortion due to electric and or magnetic field unevenness in transducers, and the acoustic distortion due to the nonlinearities of air at high sound levels. The distortion that is created by nonlinearities creates harmonics to the waveform—overtones to the fundamental tone. This is called *harmonic distortion*. In addition, the nonlinearities cause *intermodulation distortion*, which is characterized by added tones that have frequencies that are related to the sum and difference frequencies of signal components.

Because of nonlinear distortion, the linear distortion may also change character between high and low sound levels. The frequency response will be different depending on signal level. The system may become saturated.

Most microphones are dependent on the use of preamplifiers. The noise from the preamplifier will sometimes dominate over the microphone's own electrical noise. It is advantageous to express the noisiness of a microphone and preamplifier system by an equivalent room noise level, for example, as an equivalent sound level in dBA. Typically, condenser microphones will have background noise levels in the range of 15–35 dBA equivalent room noise. It is important to note that room noise and microphone noise generally have different character so even at the same dBA level number they will feature very different noise quality. The typical noise spectra for room noise and the electrical noise from microphones are shown in Figure 12.1.

Microphone *self* noise can be due to many effects. One inevitable but negligible noise source is the molecular motion of the air in which the diaphragm exists. This noise is called the microphone *thermal agitation* noise (represented by the real part of the radiation impedance seen by the microphone diagram). A second noise source is the *thermal noise due to the resistance* of the electric circuits. A third noise source

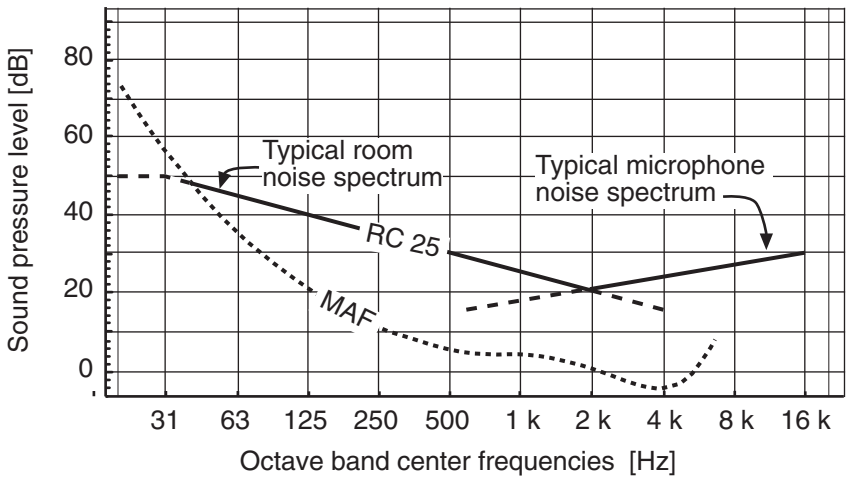


Figure 12.1 Typical spectra for room noise and microphone noise.

is the noise and *hum* induced in the electric circuits of the microphones due to *stray electromagnetic fields* in the microphone environment. A fourth noise source is the noise due to power supplies necessary for microphone electronic circuit operation. Digital microphones will also exhibit *quantization* noise characteristic of analog-to-digital (A/D) conversion systems. Other sources of noise exist as well. Microphone noise may also be due to *mechanical movement of the cables* and connectors by which the microphones are transmitting their signals.

12.3 SENSITIVITY

The *sensitivity* of a microphone is an important property in the choice of a microphone, along with the electric output impedance, and in some cases the electric power supply principle. The sensitivity of microphones usually relates to the output voltage to a certain sound pressure, often 1 Pa. The electric voltage obtained for that sound pressure is then compared to some reference voltage typical of the electric system to which the microphone is connected. Typically, the voltage at the output of an unamplified microphone is about 5–50 millivolts at a sound pressure of 1 Pa.

Many passive, self-powered microphones (microphones not using internal electric amplification) have fairly low *electric output impedance*, typically in the range of 10–200 ohm. Active microphones usually have very low output impedance, except in the case of electret microphone capsules that may need further impedance reducing electric circuits.

Active microphones, such as condenser and digital microphones, need electrical power to operate. The power is usually supplied through the audio output cables using a form of multiplexing. Since the power signal is DC, or very high frequency, it can coexist with the audio signal on the cable. The supply of electric power can be *balanced* or *unbalanced*. The most common type of balanced power supply, *phantom power*, uses an electrically shielded cable with two conductors to feed the microphone. In this way, the electric signal and its noise can be cancelled both at the microphone and at the microphone amplifier input stage. This can be done using electric *balancing transformers* or special signal balancing circuits.

Radio microphones transmit their output signal by radio to a radio receiver connected to the mixing console. Such microphones are commonly used in theatre and musical applications and can be conveniently hidden in the performers' hair or clothing.

Digital microphones often use Bluetooth transmission to connect to a computer or mobile phone. Such microphones still need an A/D conversion process to operate. Studio quality digital microphones will use PCM-based conversion and lossless transmission (see Chapter 16).

12.4 ELECTROACOUSTICAL CONVERSION PRINCIPLES

To obtain an electric signal, an electric, magnetic, or capacitive circuit must be disturbed by a mechanic or acoustic action, for example. Any microphone will also by its presence disturb the sound field in some way.

The sound pressure or the sound pressure gradient act on the microphone diaphragm(s) that is set in motion by the field. As the diaphragm moves in the magnetic or electrostatic field, the mechanical and electrical systems try to block the movement. It is the electrical system that reacts to the diaphragm movement by generating the output signal.

Many microphones work on a principle involving magnetism and inductors (coils) or electrets and capacitors. The reason is that it is practical to obtain the electric signal directly without the need for external power as for resistive microphones.

Because of flexibility, economy, and other advantages, many microphones are designed so that the transducer is housed by itself in a *microphone capsule* that is then immediately coupled, both mechanically and electrically, to a *microphone preamplifier*. The term microphone is then used to describe the entire assembly.

Most microphone capsules are designed to directly give an electric output voltage proportional to the sound. Alternate technologies are available; however, microphones may use a high frequency voltage whose frequency or amplitude is changed by the sound. In such cases, the microphone will need some form of demodulating circuit to yield a signal suitable for further audio signal amplification.

Recently microphones based on the cooling of a heated thin wire have become available. Another type of microphone recently introduced is based on micromachined sensors built directly on silicon blocks. Such microphones can be made very small, but practical microphones will need to be made up of many individual microphone elements to achieve a reasonable S/N ratio.

Microphones are often categorized by their engineering design principle:

- Electrodynamic
- Resistive
- Electromagnetic
- Condenser (electrostatic, capacitive)
- Electret
- Micromachined capacitive
- Piezoelectric

A summary of some advantages and disadvantages of some common microphone types is shown in Table 12.1.

Table 12.1 Some important properties of common audio microphone systems

Type	Advantages	Disadvantages
Condenser (external bias)	Stability High sensitivity Wide frequency range Low vibration sensitivity	Requires external electronics and bias voltage supply Relatively mechanically sensitive Some types sensitive to humidity
Condenser (electret)	High sensitivity Very low cost Small size Small vibration sensitivity	Limited temperature range Relatively high noise level Relatively high nonlinear distortion
Piezoelectric	Self-powered Simple electronics	Sensitive to vibration Limited frequency range
Electrodynamic	Self-powered No electronics needed	Sensitive to dynamic magnetic fields Large physical size Limited frequency range Sensitive to vibration
Variable resistance	High sensitivity Low cost	Limited frequency response High noise level

12.5 SOUND FIELD SENSING PRINCIPLES

A microphone which outputs an electric signal having a waveform proportional to the sound pressure is called a *pressure (sensitive) microphone*. Analogously, microphones where the waveform of the electric signal is proportional to the acoustic pressure gradient or the particle velocity are called *pressure gradient microphones* and *velocity microphones* respectively. The latter two terms are used somewhat interchangeably.

It is also common that microphones are classified by their directional sensitivity; an ideal pressure microphone has an omnidirectional sensitivity pattern and is called an *omnidirectional microphone* or sometimes just *omni* for short. An ideal pressure gradient microphone will have a bidirectional sensitivity pattern and is called a *figure-of-eight type microphone*, or sometimes just *gradient*. The *unidirectional microphone* can be engineered by using a combination of the principles of gradient and pressure microphones.

12.6 DIRECTIVITY

The *directivity* of a microphone describes the microphone's sensitivity to sounds from various directions. The importance and need for various directivity properties was discussed already in Chapter 3 where various alternatives of microphone directivities were considered for stereo recording and reproduction. Since all microphones have a finite size, all microphones will become directional at sufficiently high frequencies.

Directivity can be obtained in two different ways. One type of microphone attains its directivity by long or wide sound pickup areas. Examples of such microphones are array microphones and microphones that use parabolic reflectors. An array microphone uses sound pickup at multiple points; there are passive and active array microphones. Both of these types rely on the size and geometry of the pickup area to give the desired directivity.

A second way of obtaining directivity is using multipole techniques. The pressure gradient microphone has a sharp null in its directivity characteristic, the same as that of a dipole. Quadrupole and higher multipole patterns can also be realized, both passively and actively. The advantage of multipole techniques is that the microphone can be small. The disadvantage is low sensitivity and associated high background noise.

The directivity of a microphone is usually described by a *directivity function*. The directivity function $F(\theta,\varphi)$ is the relationship between the microphone output signal at angles (θ, φ) (see Figure 12.2) compared to its output in the *reference direction*, usually $\theta = 0$ at which $|F(\theta, \varphi)| = 1$. The directivity function is usually frequency-dependent and its magnitude ≤ 1.

Directivity plots are used to graphically display the directional properties of the microphone, for example, in the case of measurement of the directivity characteristics of real microphones. The directivity plots are usually based on $20\log|F(\theta, \varphi)|$.

The directivity function is usually defined with relation to some *reference direction* of the microphone. Usually the reference direction is chosen to be the normal to the diaphragm surface. Since most microphone diaphragms are circular, the center of the coordinate system is chosen as the center of the diaphragm. The directivity function and the directivity plot are usually defined or measured for plane

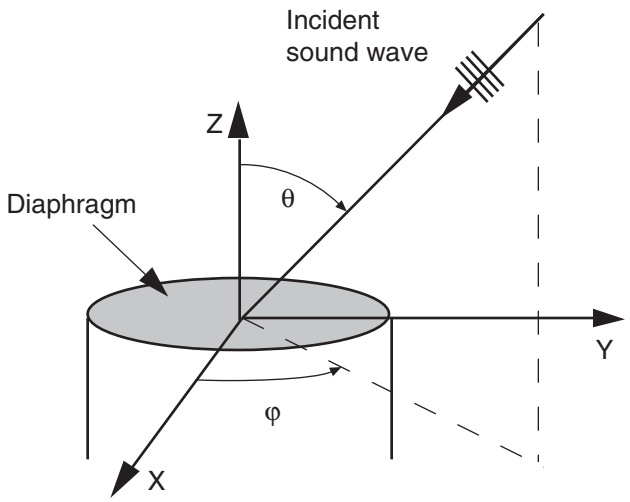


Figure 12.2 Definition of the coordinate system for the discussion of microphone directivity characteristics. The microphone diaphragm is in the plane $z = 0$. The z-axis is in the center of the microphone diaphragm.

incident waves. The direction of incidence is, in principle, specified by two angles, θ and φ. However, since so many microphones essentially have cylindrical symmetry, using θ only is usually sufficient for definition of the directivity function $F(\theta)$.

Since an *omnidirectional microphone* will have equal sensitivity to sound from any direction, its directivity function will be $F(\theta) = 1$.

A *bidirectional microphone* is characterized by a directivity function:

$$F(\theta) = \cos^n(\theta) \tag{12.1}$$

The number n specifies the *gradient order*. A first order gradient microphone has $n = 1$ and is sometimes said to have a figure-of-eight directivity since graphically the directivity pattern is a simple cosine.

There are several *unidirectional* microphone directivities. The most common unidirectional microphone is the *cardioid microphone*, which usually has the directivity function:

$$F(\theta) = \frac{1 + \cos(\theta)}{2} \tag{12.2}$$

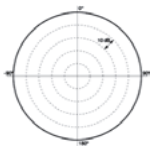
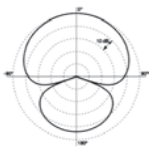
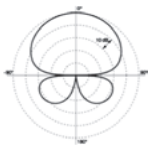
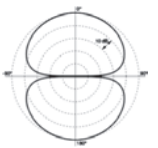
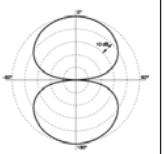
A similar type is the *hypercardioid* directivity function:

$$F(\theta) = \frac{1 + 3\cos(\theta)}{4} \tag{12.3}$$

By adding the outputs from coincident microphones (microphones that are essentially at the same point in space and that have different gradient orders and angles), one can achieve desired directivities. Some common audio microphone directivities are shown in Table 12.2.

One sometimes needs sharper directivity patterns than those that can be obtained by using gradient or multipole microphone techniques. This can be done using various forms of array microphones as mentioned previously.

Table 12.2 Some important audio microphone directivity characteristics

Characteristic	Omnidirectional	Cardioid	Hyper-cardioid	Super-cardioid	Second order cardioid	Bidirectional	Second order bidirectional
Polar response pattern							
F(θ)	1	(1 + cos(θ))/2	(1 + 3 cos(θ))/4	0.37 + 0.63 cos(θ)	(cos(θ) + cos²(θ))/2	cos(θ)	cos²(θ)
–3 dB pickup arc	360°	131°	105°	115°	77°	90°	60°
Random energy efficiency [dB]	0	–4.77	–6.02	–5.70	–8.70	–4.77	–6.99
Distance factor	1	1.73	2	1.93	2.74	1.73	2.24

One usually thinks of microphones as small devices; a typical electret microphone has a cylindrical shape with length and diameter typically about 6 mm. In spite of this, over frequencies of a few kHz, most microphones will exhibit directivity properties that are a function of their finite, although small, size.

The *directivity factor D* is a power-related metric that describes the ratio between the microphone power output for sound incident in its most sensitive direction and the mean power output over all angles. The directivity factor is defined as:

$$D = \frac{4\pi}{\int_0^{2\pi}\int_0^{\pi} |F(\theta,\varphi)|^2 \sin(\theta)\, d\theta\, d\varphi} \tag{12.4}$$

A common metric of the microphone's directivity is the directivity index (DI), that is expressed in units of dB and is defined as:

$$DI = 10\log(D) \tag{12.5}$$

The DI tells us how much more dB sensitive the microphone will be in its nominal 0°-direction than an omnidirectional microphone, for example, when recording sound in a reverberant and noisy room.

A more practical metric is that of the *random energy efficiency*, which gives the relative output of the microphone in a diffuse sound field (for example, random incidence noise or reverberation) relative to an omnidirectional microphone having the same sensitivity in its nominal 0°-direction. The reverberation energy efficiency is by definition equal to the negative number of the DI.

The *distance factor* shown in Table 12.2 shows the increase in reverberation radius that is the result of the suppression of random incidence reverberation. Figure 12.3 shows theoretical and measured data for the frequency response of a cylindrical microphone to an incident plane wave characterized by flat frequency response. We see that the output voltage level varies by more than 10 dB in the high frequency region because of the diffraction effects due to the finite microphone size. The microphone's directivity will become very directional at these frequencies as well.

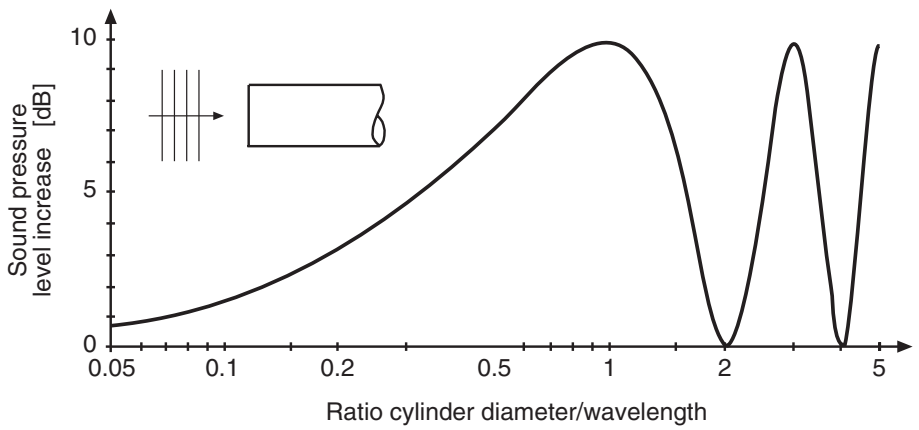


Figure 12.3 Influence of the finite dimensions of a cylindrical microphone. Theoretical and measured values for the sound pressure increase at the diaphragm end of the cylinder relative to the free field sound pressure.

12.7 DIAPHRAGM MECHANICAL PROPERTIES

The diaphragm and its suspension form the basic dynamic system of the microphone. The equation of motion for the mechanical system can approximately be written as follows:

$$\underline{F}_A = M_M \underline{a}_D + R_M \underline{u}_D + k_M \underline{\xi}_D = \left(j\omega M_M + R_M + \frac{1}{C_M} \right) \underline{u}_D \tag{12.6}$$

where

$\underline{F}_A$ = force
M_M = diaphragm mass
R_M = mechanical resistance
$C_M = 1/k_M$ diaphragm suspension compliance that is the inverse of the spring constant k_M
$\underline{a}_D$, $\underline{u}_D$, $\underline{\xi}_D$ = diaphragm acceleration, velocity, displacement.

The equation of motion shows that there will be a resonance at a frequency f_0.

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{1}{M_M C_M}} \tag{12.7}$$

The microphone diaphragm's resonance frequency can be designed to be in the desired part of the audio spectrum. The choice of frequency will depend on the properties of the electromechanical transduction mechanism chosen for the microphone as well as on the choice of pressure or pressure-gradient acoustic sensitivity desired. Pressure sensitive microphones based on electrodynamic transduction will need to have their vibrating system controlled mainly by mechanical resistance and have the resonance frequency in the center of the audio frequency range—*resistance control*. Those pressure sensitive microphones that are based on capacitive sensing of diaphragm motion need to have their vibrating system controlled by the system's mechanical compliance—*compliance control*. Pressure gradient sensitive microphones need to have their vibrating system controlled by the system's mass—*mass control*. Figure 12.4 shows the basic frequency response of three alternatives available.

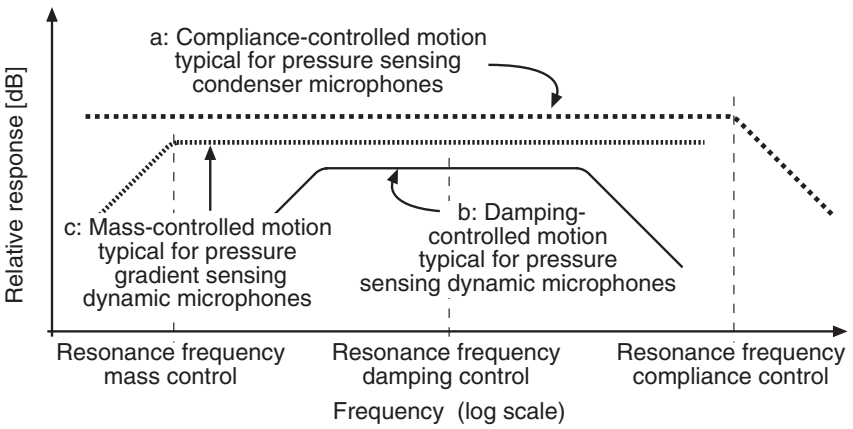


Figure 12.4 Relative frequency response for microphones using various diaphragm motion control: (a) dashed line pressure sensitive condenser microphone (compliance-controlled), (b) solid line pressure sensitive electrodynamic microphone (resistance-controlled), and (c) dotted line pressure gradient sensitive electrodynamic microphone (mass-controlled). Curves displaced to improve legibility.

By designing the mechanical system in such a way that its resonance frequency is much higher than the highest frequency of interest f_{max}, it is possible to give the microphone capsule a flat frequency response up to frequencies slightly below f_{max}. This requires the mechanical system to have its mechanical impedance $\underline{Z}_M$ dominated by the compliance term. We say that the system is *compliance-controlled*.

A *mass-controlled* diaphragm will have its resonance frequency below the audio frequency range. Typically, this design is only used with pressure gradient sensing microphones.

12.8 RESISTANCE MICROPHONES

The classical variable *resistance carbon microphone* was long the dominant microphone in telephone systems. The function of a carbon microphone is based on dynamic compacting of electrically conductive granules between two electrodes in a cavity, as shown in Figure 12.5. Since the granules are only loosely packed, variations in the degree of compaction will cause a variable resistance and thus a varying DC current in a simple electric, battery-operated circuit.

Like all resistive transducers, the resistive microphone needs an external power supply to operate that contributes toward its high sensitivity. Of course, the compaction of the granules is a nonlinear process, which means that the nonlinear distortion of the carbon microphone is high, so the carbon microphone was only used for speech purposes.

The electrical output of the microphone $\underline{e}_{out}$ is proportional to the compacting of the carbon granules and consequently to diaphragm displacement, that is:

$$\underline{e}_{out} = K\ \underline{\xi}_D \tag{12.8}$$

where K is a constant determined by the microphone design, and $\underline{\xi}_D$ is the displacement of the diaphragm from the rest. In this discussion the stiffness of the air in the cavity is neglected compared to that of the carbon granules.

This means that a pressure sensitive resistance type microphone must have its diaphragm displacement proportional to the sound pressure. The force $\underline{F}_A$ on the diaphragm is:

$$\underline{F}_A = S_D \underline{p}_D \tag{12.9}$$

where $\underline{p}_D$ is net pressure on the outside of the diaphragm, and S_D is the diaphragm area.

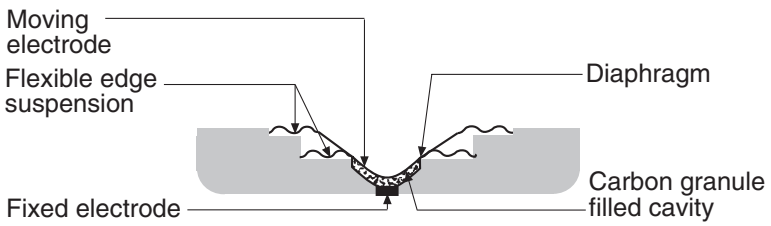


Figure 12.5 A microphone operating, using a sound modulated, carbon granule resistance cartridge.

The pressure sensitive, variable resistance microphone must use a compliance-controlled diaphragm unless frequency response equalization takes place in an associated filter or amplifier. The output voltage is then proportional to sound pressure since:

$$\underline{e}_{\text{out}} \propto \underline{\xi}_D \propto \frac{\underline{u}_D}{j\omega} \propto \underline{p}_A \tag{12.10}$$

Modern variations on the variable resistance microphone exist as micro-machined piezoresistive microphones. Since these are usually built on a monocrystalline substrate, the amplifier filter may be easily integrated into the function. Such microphones are now often found in communications equipment such as mobile phones.

12.9 PIEZOELECTRIC MICROPHONES

Piezoelectric microphones (also sometimes referred to as crystal or ceramic microphones) are based on the force of the diaphragm acting on a piezoelectric element. A piezoelectric element will become electrically charged if acted upon by an external force. Examples of materials that exhibit piezoelectric properties are crystals, such as quartz and Rochelle salt, as well as ceramic materials, such as lead-titanium-zirconate and polymers like polyvinyldifluoride (PVDF). The design of a typical piezoelectric *bender* microphone is shown in Figure 12.6.

The electric charge buildup causes a voltage over the piezoelectric element electrodes. The piezoelectric element's output voltage is given by:

$$\underline{e}_{\text{out}} = K\ \underline{\xi}_D \tag{12.11}$$

where $\underline{e}_{\text{out}}$ is the output voltage without electrical load, $\underline{\xi}_D$ the deformation, and K a constant depending on the piezoelectric material and the engineering design.

The piezoelectric element is an electric charge generator, and its electrical characteristic is typically that of a fairly low capacitance, typically a few nanofarads. Unless used with a microphone preamplifier, usually an *impedance converter*, to lower its output impedance, the microphone output impedance is very high, making its output voltage sensitive to cable capacitance and leakage resistance.

Since the electric output is proportional to displacement, the mechanical system of the piezoelectric microphone must be designed to the same principles as the resistive microphone. The microphone will typically have a flat frequency response, limited at the high frequency end by the mechanical resonance of the mechanical system, and at the low frequency end by the design of the electric circuit. Within this

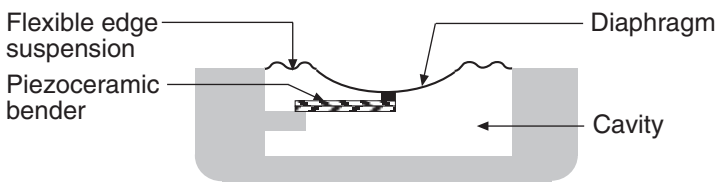


Figure 12.6 A piezoelectric microphone capsule operating, using a ceramic bimorph element.

frequency range, its frequency response can be made very flat, making the microphone suitable also for acoustic measurements.

12.10 CAPACITIVE MICROPHONES

Capacitive microphones are also known as electrostatic microphones or, more often, *condenser microphones*. The primary advantage of condenser microphones over other microphone types is their low linear and nonlinear distortion. Condenser microphones, because of their very low output voltage and their high capacitive internal electric impedance, always need a microphone preamplifier, typically an impedance converter. The internal capacitance is typically in the range of 5–50 pF.

Condenser microphones are available as two types, with or without electrically prepolarized diaphragms. The mechanical design of both types follows the same rule. The advantage of the prepolarized type is primarily a simpler power supply. The design principle of the condenser microphone is shown in Figure 12.7. The microphone capsule, the transducer, is a simple plate capacitor in which the distance between the plates varies due to sound pressure. The capacitance C_E between two parallel, equally large, metal plates placed close to one other, is:

$$C_E = \frac{\varepsilon\varepsilon_0 S}{d} \tag{12.12}$$

where ε, ε_0 is dielectricity constants, S is plate area, and d is distance between the opposing plates.

If the distance between the plates varies, the capacitance will vary as well. This capacitance variation can be used in several ways to produce an electric output signal proportional to the sound pressure.

The traditional, passive condenser microphone uses a flexible, electrically conductive diaphragm and a *back electrode* to form the two electrodes. In measurement and studio microphones, the capacitor is usually housed in a microphone capsule. Passive condenser microphones have to be supplied with a *polarization (bias) voltage* to generate a static charge between the plates.

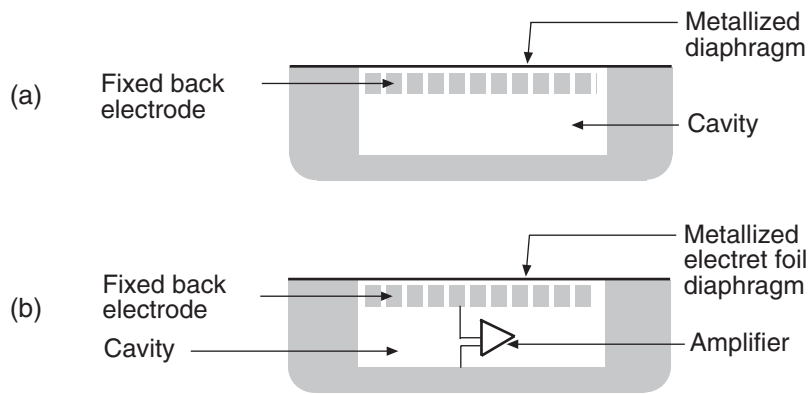


Figure 12.7 Two varieties of condenser microphone capsules: (a) traditional high voltage charged diaphragm, (b) precharged electret foil diaphragm.

The bias voltage is supplied to the condenser microphone capsule by way of a large electric resistance, typically 200 MOhms or higher. If the so-called time constant of the microphone, the product of the resistance and the capacitance of the microphone capsule, is large enough, the charge Q_E over the capacitor will be approximately constant over more than one cycle of sound pressure variation. Usually the time constant is chosen to be larger than 1 s. Fast variations in capacitance, such as those due to audio sound pressure variations, will, because of the constant electric capacitor charge, lead to voltage variations that follow the sound pressure variations.

For a charge Q_E, the voltage U_E over a capacitor is related to the capacitance C_E as:

$$U_E = \frac{Q_E}{C_E} \tag{12.13}$$

This means that the audio voltage $\underline{e}_{\text{out}}$ over the constant charge capacitor will be related to the diaphragm displacement $\underline{\xi}_D$ as:

$$\underline{e}_{\text{out}} = K\ \underline{\xi}_D \tag{12.14}$$

where K is a constant that depends on the microphone design and polarization voltage. Since the audio voltage is proportional to the diaphragm displacement, this microphone is also used in its compliance-controlled mode.

The low capsule capacitance and the low audio output voltage make the microphone sensitive to electrical interference. In practice, the microphone capsule has to be close to the preamplifier.

Because of the low capsule capacitance, the microphone circuit is sensitive both to *leakage resistance* and to *stray capacitance*. Leakage resistance is often due to the finite parallel resistance that appears between the electrode connectors due to dirt and fungus growth in a humid atmosphere. Even small leakage currents will usually lead to noise at the microphone output.

The stray capacitance is the capacitance that appears between the electrical conductors on the signal's path to the preamplifier. The preamplifier will also contribute parallel resistance and capacitance. Stray capacitance will lead to reduced microphone sensitivity.

In addition, the leakage of sound to the back side of the diaphragm will result in a further drop of response at low frequencies. Since it is usually desired for the microphone to have flat frequency response, the mechanical circuit of the microphone capsule has to be compliance controlled, that is, have its resonance frequency much over the highest audio frequency of interest.

Most condenser microphones manufactured today are *electret microphones*. The electret microphone uses an electrostatically charged, prepolarized device, an *electret*, usually a plastic film, to permanently charge the microphone capsule so that no bias voltage is necessary. PVDF is a plastic that can be electrostatically charged so that it can be used in microphones. The film can be used both as a diaphragm, after metallization, and as a layer on the back electrode.

By using the precharged electret film, there is no need for a bias voltage to be applied to the microphone capsule to provide the charge necessary to produce an audio output voltage. Together with a preamplifier using a field effect transistor, the electret microphone can be made very small. It can be used directly where probe microphones would otherwise be necessary or where it is important not to disturb the sound field. It also has the advantage that it can be manufactured at low cost.

Another way of using the capacitance variations produced by the sound pressure on the diaphragm is to employ the capsule capacitance as an element in an electric resonance circuit. The shift in resonance frequency can be detected in many ways. An advantage of this method is that there is no noise due to leakage currents.

The diaphragm is usually made of thin nickel or plastic film. Plastic films have to be made electrically conductive, typically by application of a thin gold or aluminum layer on the film by sputtering or electrochemical deposition.

The use of plastic films for condenser microphone capsules is now widely accepted even for high quality measurement microphones for laboratory use. Although plastic films do not have as good a resistance to high temperature and humidity as nickel film, accelerated tests have shown the sensitivity of, for example, electret films to be virtually unchanged over many years. One advantage of nickel films is that the diaphragm can be tensioned much more tautly, which results in a higher resonance frequency that gives a wider frequency range response. Of course, the sensitivity of the microphone capsule to sound suffers from this and the S/N ratio of the microphone will be lower. One important advantage of the low mass diaphragm of any capacitive microphone is that the microphone has little sensitivity to vibration.

The condenser microphone can easily be made so that the surface area of the diaphragm covers the entire end of a cylinder. This design results in the microphone having little influence on the sound field, and having comparatively low directivity. As shown in Figure 12.2, there is still considerable pressure buildup in front of the microphone capsule in such designs. The pressure buildup can be compensated for by strong damping of the microphone resonance. This damping is equivalent to a frequency response filtering by −6 dB over some limiting frequency. The microphone will then have fairly flat frequency response for a plane wave incident on the axis of symmetry of the microphone. Such a microphone will not, however, have flat frequency response in a diffuse sound field.

Micro-electric machined systems (MEMS) microphones form a recent addition to the available range of microphones. Capacitive MEMS microphones may be made by etching a grill, with a cavity below it, on a monocrystalline silicon substrate (that may already carry a CMOS electronics region). The grill is then covered by a polymer coating. The cavity will need to carry a gas—for example, air—at the same pressure as the surrounding air. This is accomplished by venting the cavity through a (meandering) channel etched into the substrate. There also are MEMS microphones based on piezoelectric and piezoresistive technologies. Yet another MEMS microphone technology uses a flexible diaphragm that works as the moving gate of the field effect transistor.

In practice, any microphone capsule needs a protective grille. This grille can be designed to minimize but not remove the frequency response difference between frontal incident sound and diffusely incident sound. Most laboratory measurement condenser microphones are designed as such pressure-sensitive omnidirectional microphones.

The bidirectional response can be obtained by using the diaphragm together with an acoustically transparent back electrode and no rear cavity.

The principles used in the design of capacitive microphones also make it relatively easy to design microphones having switchable omni, bidirectional, and unidirectional directivity as shown in Figure 12.8.

The unidirectional response is achieved by giving the back electrode a suitable acoustic resistance and making the rear side of the back electrode open to the sound. The bidirectional characteristic is achieved with making the capsule symmetric by having two diaphragms and combining the output voltages of both.

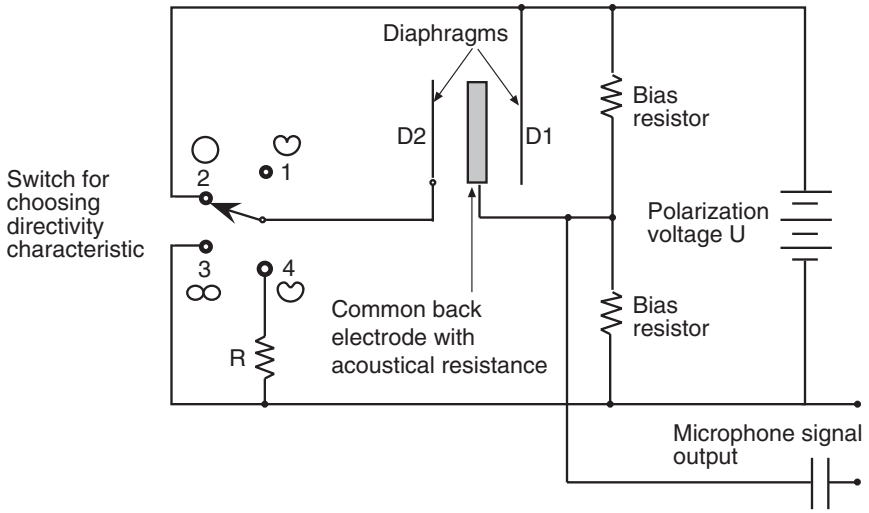


Figure 12.8 Electrical circuit for a capacitive microphone with variable directivity. Switch positions: 1) cardioid, 2) omnidirectional, 3) bidirectional, and 4) hypercardioid directivity. Note that the back electrode has to have a special flow resistance for the system to operate as intended. Microphone electrical connection is between back electrode and ground.

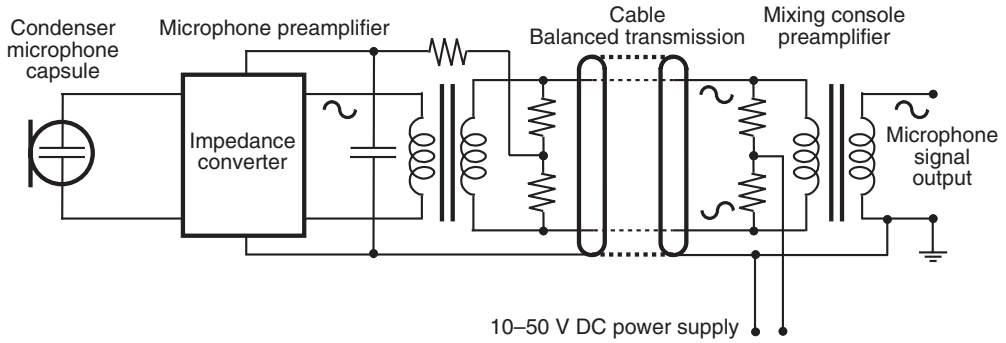


Figure 12.9 The phantom feed principle often used for capacitive microphones.

Professional condenser microphones are often powered using the phantom feed principle. This means that the audio and the supply current are multiplexed on the same wires, as shown in Figure 12.9.

12.11 ELECTRODYNAMIC MICROPHONES

The *dynamic microphone* works on the electrodynamic principle—a voltage will be induced in an electrical conductor moving in a permanent magnetic field. Since the magnetic field can store much energy, the microphone type can be made fairly sensitive. Because of its design, it is typically self-powered and does not require an impedance converter. The low impedance makes it possible to

typically connect it to the preamplifier over long, balanced lines. Since the microphone works by the electrodynamic principle, it is very sensitive to stray electromagnetic fields (hum), although a compensation coil can be used to reduce the sensitivity to such fields.

It is difficult to design a dynamic microphone that has a frequency response as flat as that of a condenser microphone. This is due to the engineering compromises that must be made in its design. An electrodynamic microphone needs to be damping-controlled to have a flat frequency voltage response to sound pressure. For the frequency response to be flat over a wide frequency range, the microphone will then need to be damped, which will result in a low output voltage. Instead, the frequency response of dynamic microphones is extended using acoustic resonant elements at low and high frequencies in front of and behind the diaphragm. This results in good sensitivity and sufficiently good frequency response for many applications.

A section of a simple electrodynamic microphone is shown in Figure 12.10. The incident sound pressure works primarily on the external side of the diaphragm. The diaphragm is firmly attached to a paper or plastic cylinder carrying a coil of electrically conductive wire that is placed in a magnetic field. The diaphragm is usually only suspended by a mechanically compliant structure at its edge. The resonance, which is caused by the mass and compliance, can be effectively damped both acoustically and mechanically.

The movement of the diaphragm and voice coil can be described by the same equation as previously, Equation 12.7. The voltage generated by the moving coil is:

$$\underline{e}_{\text{out}} = Bl\underline{u}_D \tag{12.15}$$

where $\underline{e}_{\text{out}}$ is the output voltage of the unloaded microphone, B is the magnetic field strength, l is voice coil length, and $\underline{u}_D$ is the velocity of the coil.

For the microphone to have a flat frequency response, the mechanical circuit must be strongly damped, and the moving system must be *resistance-controlled*. Since the damping leading to flat frequency response reduces the sensitivity of the microphone, large damping is unattractive because it reduces the sensitivity of the microphone and makes it susceptible to noise. To retain good sensitivity and still have a reasonably flat frequency response, one needs to extend the frequency response both upward and downward by using resonant acoustic circuits. This results in less than optimal reproduction of transient sounds.

While Figure 12.10 shows the design of an omnidirectional electrodynamic microphone, dynamic microphones can also be designed for a bidirectional or unidirectional directivity characteristic.

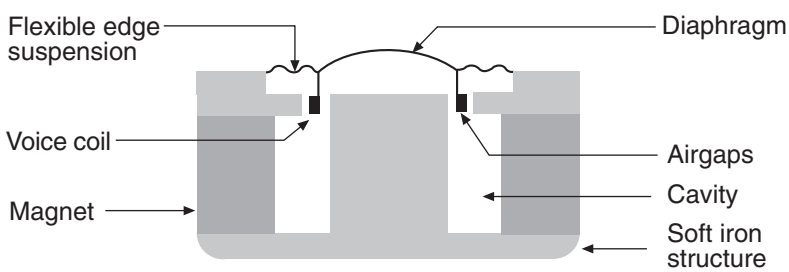


Figure 12.10 The electrodynamic omnidirectional microphone.

Such an alternate electrodynamic microphone is the *ribbon microphone* shown in Figure 12.11. The ribbon microphone is best designed as a pressure gradient microphone and usually has a bidirectional response because of this front/back symmetry. On occasion, the bidirectional response is desired for the balance between direct sound and reverberation, as well as noise reduction.

The ribbon microphone uses a lightweight electrically conductive ribbon in a magnetic field, as shown in the figure. The pressure difference between the two sides of the ribbon causes the ribbon to move. Because of the magnetic field in which the ribbon moves, a voltage will be generated across the ends of the ribbon. The voltage is low but is transformed to conventional microphone output voltages by a transformer.

The pressure sensitive ribbon microphone senses the sound field in the following way. Assume that the microphone has its ribbon in the $z = 0$ plane. Further assume the sound wave to be incident at an angle relative to the $z = 0$ plane, as shown in Figure 12.1. The resulting net force on the diaphragm $\underline{F}_A$ will be:

$$\underline{F}_A = S_D\left[\left(\underline{p}(z) - \frac{d\underline{p}}{dz}\Delta l\cos(\theta)\right) - \underline{p}(z)\right] = S_D\frac{d\underline{p}}{dz}\Delta l\cos(\theta) \tag{12.16}$$

where $\underline{p}(z)$ is sound pressure at a point z, $\cos(\theta)$ is angle of incidence relative to the z-axis, Δl is effective path length between the microphone's front and back sides, and S_D is the area of one side of the diaphragm.

The diaphragm is assumed to be very thin. Since the pressure gradient is proportional to the particle velocity in a plane wave, the net force can also be written as:

$$\underline{F}_A \propto S_D\,\omega\,\underline{u}_D\,\Delta l\cos(\theta) \tag{12.17}$$

Because of this property, the pressure gradient sensing microphone type is also sometimes called a *velocity microphone*.

The mechanical action of ribbon can be considered a mass-spring system that is damped by the radiation impedance and the viscosity of the air in the air gaps between the edges of the ribbon and the sides of the magnet's pole pieces. The ribbon behaves somewhat like a string and is characterized by many resonances; the higher order resonances are very damped, however.

We can use the same equation of motion as for the previously studied microphones because of the simple mass-spring analogy. Since the force is proportional to the pressure gradient $\partial\underline{p}/\partial z$, which

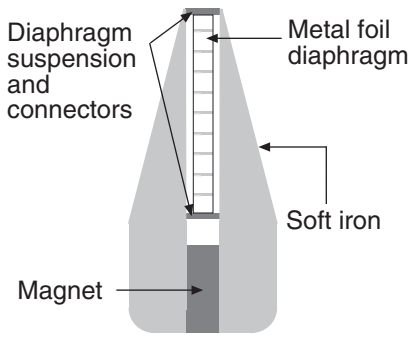


Figure 12.11 A pressure gradient ribbon microphone using a limp metal foil diaphragm. The soft iron magnet pole pieces are shaped so that the magnetic field in the air gap will be uniform.

becomes stronger as the frequency increases, the microphone must be made so that the mechanical impedance of the mass compensates for this frequency increase.

The microphone diaphragm must be *mass-controlled*—the M_M term must dominate in the equation of motion, Equation 12.7, to obtain a flat frequency response. This means that the fundamental resonance frequency of the mechanical system must be very low, well below the lowest audio frequency of interest.

The voltage response of the ribbon microphone to an incident plane wave can then be written as:

$$\underline{e}_{\text{out}} = K\,\underline{p}_A \cos(\theta) \tag{12.18}$$

where K is a constant that depends on the microphone design.

Note that the frequency response of the microphone is essentially flat over the resonance frequency, but that the microphone has a cosine-shaped directivity. At high frequencies, the microphone will have an even sharper bidirectional response than that of a cosine. This is due to the microphone diaphragm becoming large compared to the wavelength. The magnet's poles' pieces will also cause slight acoustic resonances leading to increased directivity and amplification.

Since the microphone is mass-controlled, it will also act as a vibration sensor; the ribbon microphone must be softly suspended to reduce the vibration sensitivity. This makes the microphone impractical to use.

Since the magnet of a ribbon microphone is usually made having large magnetic flux, a ribbon microphone can have about the same sensitivity as a conventional dynamic microphone. The need for a large magnet, a transformer, and high mechanical precision makes the ribbon microphone expensive to manufacture.

12.12 SUPER-DIRECTIONAL MICROPHONES

As mentioned previously, directional microphones can be designed according to two principles, multipole (gradient) designs and array designs.

Gradient designs are usually relatively small whereas *array microphones* are large since their function requires the microphone baseline to be large compared to the wavelength of the sound. Gradient designs can have directivities that are fairly frequency-independent over at least two decades, whereas array type microphones have directivities that are frequency-dependent. Most directional microphones used for studio recording are gradient designs. Array type microphones are sometimes used for measurement purposes and for specialized purposes, such as voice and birdsong pickup at large distances.

Since the directivity of an array microphone is also dependent on the microphone capsules used, one can use gradient microphones as elements in the array to achieve some low frequency directivity along with the directivity offered by the array action.

Noise Canceling Gradient Microphones

Gradient microphones base their function on the pressure gradient of the incident wave. In some cases, one can use this property to make microphones that have an improved S/N ratio.

The sound field close to a small sound source is different from that of a plane wave as studied in Chapter 1. The output signal of a velocity microphone placed close to a monopole sound source—that is, in a sound field that has spherical symmetry—will increase toward low frequencies in a sound field that has a flat frequency sound pressure response. If the gradient sensing microphone is designed so that it will compensate for this frequency increase, the microphone will correspondingly reject incoming plane waves (besides the rejection due to directivity). Many types of noise—for example, in office environments—are typically characterized by random incidence because of reverberation, and this noise canceling property will then function very well.

This type of microphone is common in various types of helmets and headsets for close pickup of sound; sometimes the microphone type is called *close-talking microphones*.

Array Microphones

Examples of popular array microphones are the parabolic reflector microphone, the passive or active line microphone, and the slit microphone. The directivity of these microphone types is based on the constructive and destructive interference between added sound components from various directions because of phase differences. More advanced arrays may use inhibition digital signal processing circuits, such as those used in cocktail-party processors, to reduce side lobes in the directivity pattern.

The directivity of an array microphone can be predicted from simple addition of its location dependent output signals. By using signal processing one can turn the sensitivity of the array microphone to the desired direction by manipulation of the phase (delay) and amplitude of the microphone signals, as indicated in Figure 12.12.

Figure 12.13 shows the design of a simple passive line array microphone. It functions by the passive addition of the sound at various tube openings. The sound is transmitted from the opening to the sound collecting cavity in front of the microphone diaphragm by tubes that act as *delay lines*. Active line and surface arrays give much better results since the addition can be optimized to give approximately frequency independent *main lobe* directivity or *side lobe* free directivity. The direction of the main lobe can also be easily adjusted.

Array microphones that have linear extension can only achieve toroidal or shotgun type directional patterns. To obtain receiving patterns that are better controlled, one can use two-dimensional arrays or

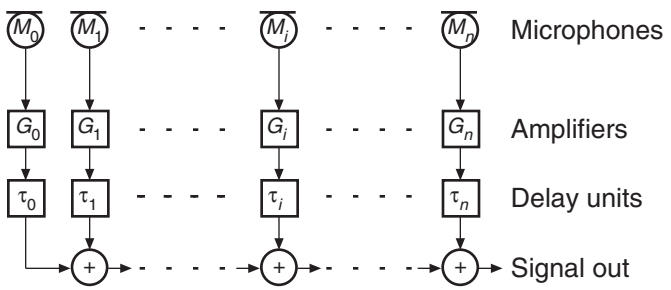


Figure 12.12 The basic operating principle of array microphones using beam-forming electronic control of pickup directivity pattern.

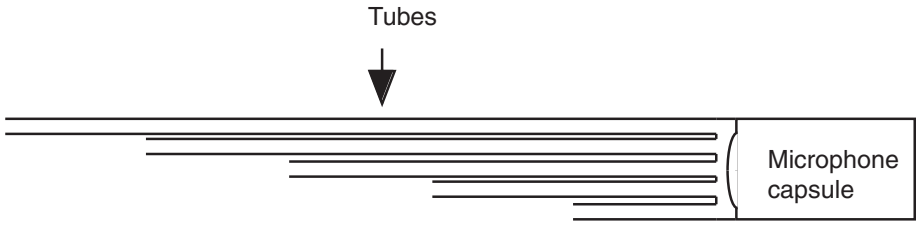


Figure 12.13 An old-style passive array microphone design using acoustic delay lines to feed the signals from the pickup points to the microphone.

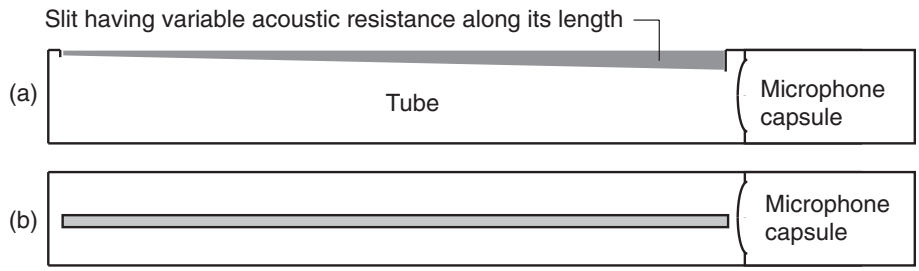


Figure 12.14 A slit microphone made using a cylindrical tube: (a) section through the microphone tube, and (b) top view of the microphone tube.

a parabolic reflector microphone, which essentially behaves as a passive two-dimensional continuous array.

The two most common types of line arrays are the *broadside array* and the *endfire array*. If a number of microphones are positioned along a line, and their output voltages added, the resulting directivity pattern will take the form of a toroid with the line at its symmetry center; this is the broadside pattern. If instead all microphone signals are added up by means of appropriate delay lines so that all signals sum in phase for a sound wave in the direction of the line, the pattern becomes directed toward one end of the line; this is the endfire pattern. The design shown in Figure 12.13 is a passive array using an endfire design. The acoustic signals add up at the microphone capsule.

The slit microphone is an endfire array, where all the pickup points have been replaced by a slit (Figure 12.14). The tube in front of the microphone can be thought of as a continuous distribution of pickup points. The slit is covered or contains an acoustic resistance that varies along the length of the slit. The sound that travels longest in the tube is most attenuated so the resistance to the incoming wave has to be highest close to the microphone.

The parabolic reflector microphone uses the circular pickup area of its parabolic reflector as a *continuous surface array*. The sound is picked up by a pressure or uni-directional microphone at the focal point of the parabola. The directivity can be changed by modifying the parabola, offsetting the microphone from the focus, and by choice of microphone directivity. For the parabolic reflector to work efficiently, the reflector diameter must be very large compared to the wavelength of sound, typically at least three times the wavelength. The *amplification* of the parabolic reflector is frequency-dependent

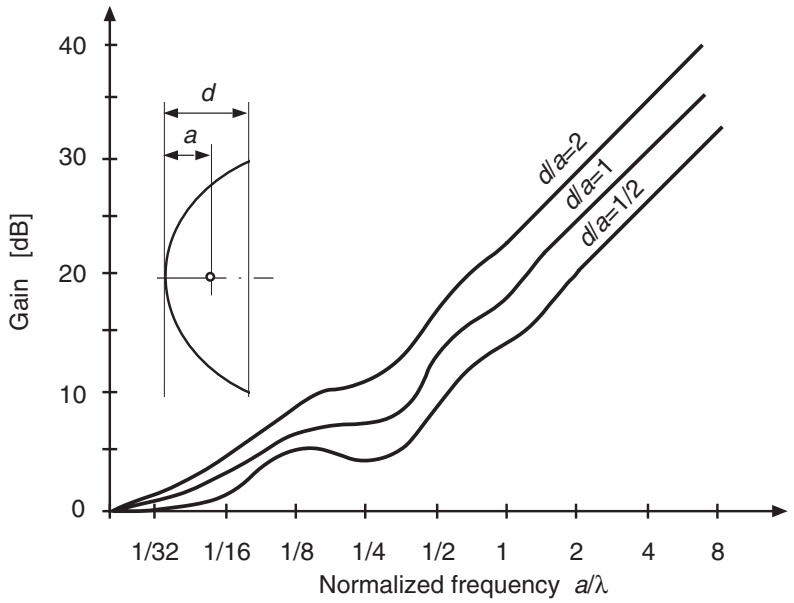


Figure 12.15 The pressure amplification characteristics of a parabolic reflector for some different choices of design variables (adapted from Reference 12.3).

so parabolic reflector microphones should be provided with a frequency response correcting filter to have a flat frequency response. The frequency response is irregular in the frequency region where the reflector diameter is of about the same size as the wavelength. Figure 12.15 shows the amplification of sound pressure at the focus of a theoretical parabolic mirror as function of frequency for varying reflector diameters.

12.13 WIND NOISE

The turbulence around microphones in wind causes local pressure fluctuations that are picked up by the microphone as noise. Such *wind noise* is a common problem when measuring and recording sound outdoors. The turbulence occurs close to the microphone, which disturbs the flow of air.

Wind noise can also occur when recording voice, due to breath. When voice is a problem, a *pop screen* is used to reduce the pops, which typically occur when plosives such as *p* and *b* are spoken or sung. The microphone may also be overloaded (buffeted) electrically and/or acoustically by the low-frequency wind gusts.

By moving the region of turbulence away from the microphone, one can reduce the noise picked up because of the strong decrease of pressure with distance. Typically, wind shields take the form of an acoustically transparent layer of cloth or plastic foam on a wire sphere surrounding the microphone at some distance. Even though the area of turbulence increases in this way, the increased distance between the microphone and the zone of turbulence more than compensates for the increased volume of turbulence.

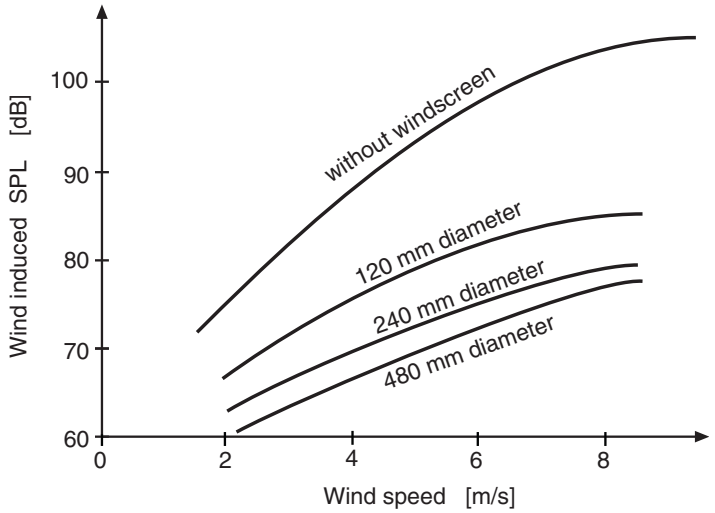


Figure 12.16 Wind induced noise levels at some wind speeds for a 25 mm diameter cylindrical microphone at the center of spherical windscreens of different diameter. The wind is incident in the $z = 0$ plane according to Figure 12.1. (Adapted from Ref. 12.1)

Figure 12.16 shows some results for the level of noise picked up by a microphone in wind, with and without windscreens of different diameters.

12.14 PROBLEMS

12.1 For a microphone, small compared to the wavelength, one may assume that the same force is acting over all of the microphone diaphragm's surface area; that is, the force is the product of sound pressure and diaphragm surface area.

Tasks:

a. Draw a simple sketch showing which parts of the microphone diaphragm correspond to mass, spring, and damper in a simple mechanical model. Draw this mechanical model and derive the equation of motion for the membrane, the mechanical impedance of the system, and its resonance frequency.
b. Show how one can obtain a constant ratio between rms sound pressure and rms output voltage above, below, or at the resonance frequency of the moving system of a microphone in which the output voltage is proportional to the diaphragm's.

   i. Excursion (which type of microphone?)
   ii. Velocity (which type of microphone?)
   iii. Acceleration

12.2 A dynamic microphone having the following characteristics is placed in a sound field:

Bl-product $5 \cdot 10^{-2}$ N/A
Diaphragm mass $1 \cdot 10^{-3}$ kg
Diaphragm resonance frequency $3 \cdot 10^{2}$ Hz
Diaphragm surface area $2 \cdot 10^{-4}$ m^2

Tasks:

a. Calculate the microphone's output voltage as a function of sound pressure level and the *Q*-value of the mechanical system. Assume the simplest possible mechanical model for the microphone. Hint: Q_M is defined as $Q_M = \omega_0 M_M / R_M$.
b. Draw the frequency response curve for the following values of Q_M: 1, 0.1 and 0.01. Plot the output voltage level as a function of frequency using a logarithmic frequency scale. Which value will the output voltage have at 1 kHz for a sound pressure level of 94 dB?

12.3 A dual-diaphragm condenser microphone is assumed to have the following directivity function:

$$F(\theta) = \frac{Q_1}{2}(1 + \cos(\theta)) + \frac{Q_1}{2}(1 + \cos(\theta + \pi))$$

where Q_1 and Q_2 are constants proportional to the charge on the two diaphragms respectively.

Task: Calculate, using this expression, how the polarization voltage should be chosen to obtain the following directivity characteristics:

a. Omnidirectional, $F = C$
b. Cardioid, $F = C_1 + C_2 \cos(\theta)$
c. Bi-directional, $F = C \cos(\theta)$

12.4 A bi-directional microphone has the directivity function $F(\theta) = \cos(\theta)$.

Task: Calculate the microphone's DI.

12.5 A dynamic, pressure sensing microphone is positioned close to the mouth of a speaker. The microphone has the following characteristics at 1 kHz:

Bl-product 1 N/A
Diaphragm mass $1 \cdot 10^{-2}$ kg
Diaphragm suspension mechanical resistance $6.8 \cdot 10^{3}$ Ns/m
Diaphragm surface area $3 \cdot 10^{-4}$ m^2

Task: Calculate the output voltage of the microphone for sound having sound pressure levels shown in the table.

Frequency	500	1k	2k	[Hz]
SPL	85	88	80	[dB] re 20 μPa

12.6 An intercom uses an electrodynamic loudspeaker unit both as loudspeaker and microphone. The microphone/loudspeaker has the following characteristics:

Bl-product 1 N/A
Effective loudspeaker unit diameter $3 \cdot 10^{-2}$ m
Diaphragm mass $1 \cdot 10^{-3}$ kg
Voice coil resistance 5 Ω

Task: Calculate the unit's S/N ratio when used as a microphone above its resonance frequency in the following case. The sound pressure level L_p at the membrane is 60 dB and the frequency of the sound is 1 kHz. The rms noise voltage in a resistor is given by $e = (4\,RkT\Delta f)^{1/2}$, where R is the resistance [Ω], k is Boltzmann's constant, T is the absolute temperature [K], and Δf is the noise bandwidth [Hz]. Assume the system bandwidth to be 4 kHz.

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Phonograph Systems 13

13.1 INTRODUCTION

A phonograph system uses electromechanical means to engrave and subsequently track a groove containing signal information onto a disc.

The principles of recording and playback systems are shown in Figure 13.1. A *cutting lathe* is used to hold the original *lacquer covered aluminum disc*. The audio signal is amplified and fed to an electrodynamic *cutting head*, with a *stylus* moving in proportion to the applied current. The heated stylus is used to cut the groove in the thin lacquer layer on the surface of the rotating aluminum disc. The signal to the cutting head will modulate the *stylus tip* position both *laterally* and *vertically,* creating a *modulated groove*. The groove modulation is tracked by the pickup cartridge stylus on playback and converted to an electric signal. The cutting head travels on a radially positioned bar so that the cut groove is tangential to the cutting head at the point of contact. Using an electroplating process, the engraved disc is used in a process for making a number of stampers used for the pressing of vinyl disc copies. These copies are then played back using a *turntable*.

In addition to the turntable, the playback system has two main components:

- A tone arm that holds the pickup cartridge, making it possible to track the groove but that does not move at audio frequencies.
- A pickup cartridge with a stylus arm to track the audio frequency signal. The cartridge also has an electromechanical system to convert the stylus arm movement to an electrical signal.

It should be emphasized that while this section only discusses cutting heads and cartridges, the resulting sound quality is determined to a large extent also by the other components in the chain, such as amplifiers, *tone arm*, phonograph motor, *turntable*, and bearings. Noise from turntable bearing, motor vibration, dirt, and dust particles in the groove will result in annoying noise. Feedback due to vibrations induced in the system supporting the playback turntable may also be annoying.

13.2 DISC CUTTING

Since the groove position contains the recorded information, it is necessary to also discuss the groove-cutting process. The disc cutter is based on the use of a turntable rotating at constant rotational speed, typically 45 or 33.33 rpm, although discs made using *half-speed cutting* is also commercially available.

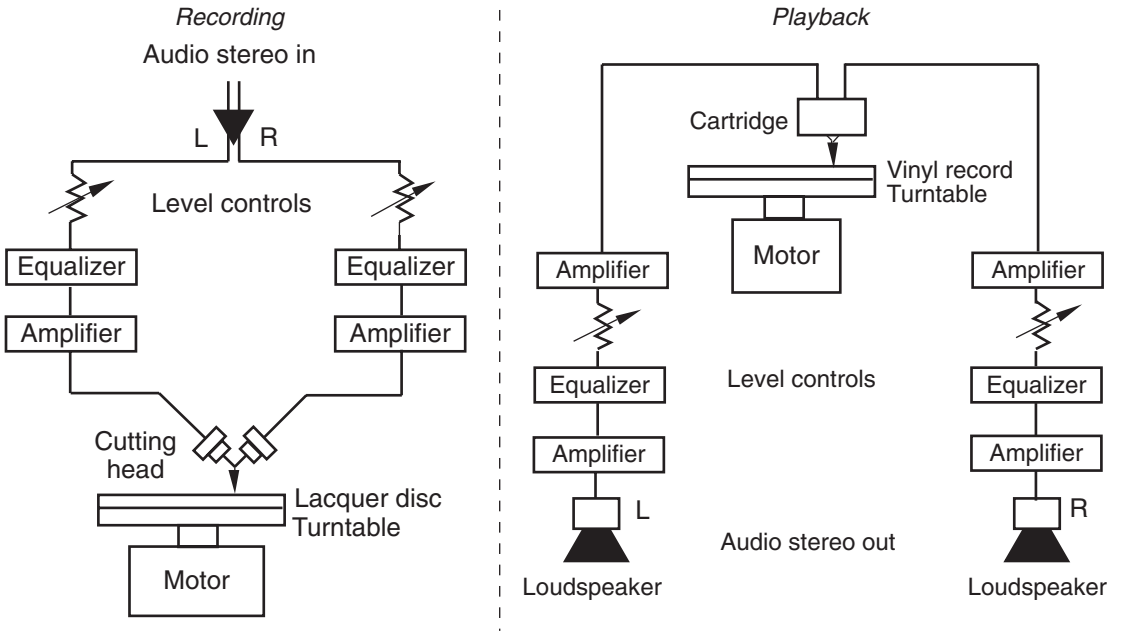


Figure 13.1 The phonograph sound recording and playback systems (after Ref. 13.1).

Guides and a screw carry the cutting head radially across the disc while the disc is spinning, cutting a spiral track in the lacquer on the original disc. The radial speed of the cutting head is dynamically changed according to the need for high excursions by the cut groove. This is done by sensing the signal to be recorded a short instance in advance to know the need for cutting head movement. A conventional stereophonic disc is cut using both lateral and vertical cutter stylus tip movement.

Cutting Heads

The basic design of a stereo disc cutting head is shown in Figure 13.2. The electromechanical system typically consists of two electrodynamic drivers that move the stylus. The drivers are mounted at right angles to one another and generate stylus tip movement in a plane almost perpendicular to the groove. Using the so-called 45/45 system, grooves can be cut in combinations of both lateral and vertical movements. The left and right channel information is carried by the groove walls at ± 45° angles to the disc surface.

The cutting head is usually tilted so that the movement plane of the cutting stylus is at an angle of 15° relative to the vertical axis. In this way, the cartridge will be able to track the groove with little nonlinear distortion.

The cutting head's motional system, in fact, consists of two first-order resonant systems, one for each transducer channel shown in Figure 13.2. These have their resonances in the middle of the audio frequency range.

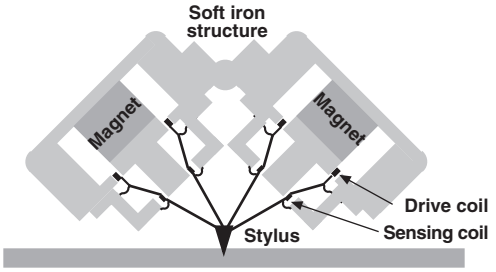


Figure 13.2 An example of the construction of a stereo cutter head.

Each electrodynamic transducer carries two coils. One coil is the main cutting coil and the second coil is the sensing coil. The output of the sensing coil is a voltage proportional to the tip movement. This voltage is used as feedback to adjust the driving current to the main coil so that the cutting head stylus moves according to the input voltage. The feedback system makes the frequency response correct and reduces the nonlinear distortion.

Examples of typical frequency response curves, with and without feedback applied, are shown in Figure 13.3. Special recording pre-emphasis filters are used that, along with filters in the playback preamplifier, achieve the goals of low noise, low distortion, and flat frequency response in the overall system. The standardized RIAA playback response curve is shown in Figure 13.5.

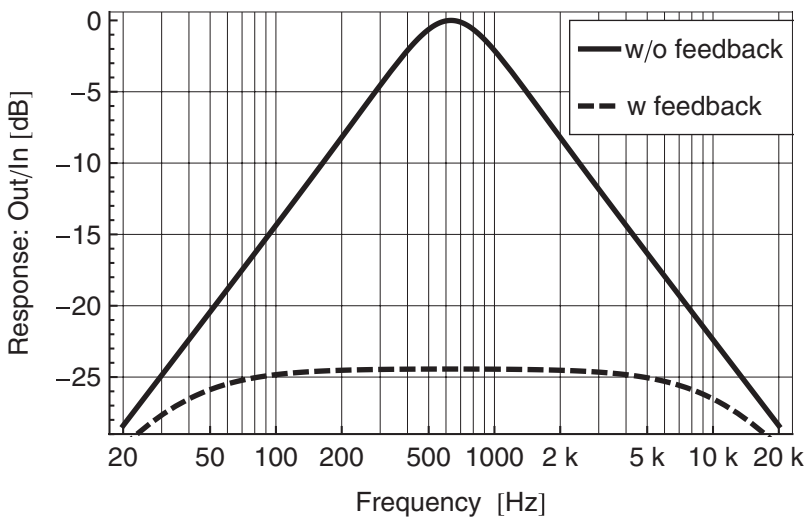


Figure 13.3 An example of the typical frequency response characteristics of a cutter head without A and with B feedback.

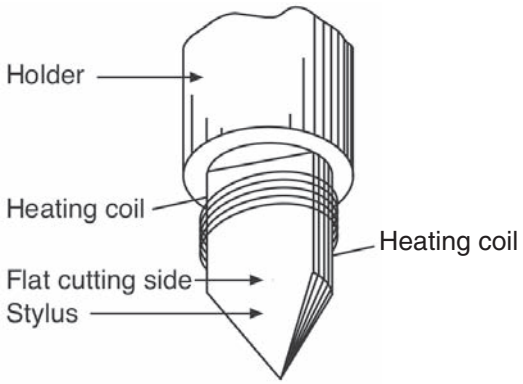


Figure 13.4 An example of a cutter head stylus equipped with a heating coil.

The amplifier driving the cutting head must typically be able to deliver at least 100 W of electrical power to each cutting head coil. The main cutting coil and the cutting head both become very hot in the cutting process. Cutting heads often use cooling systems to remove the heat using fluidic systems.

Cutting Styli

The cutting head's stylus tip is made of diamond, ruby, or sapphire and is shaped as shown in Figure 13.4. Since the stylus tip is acting on lacquer, a softer but still hard medium, there is friction, resulting in noise and distortion unless the stylus tip is heated. The stylus tip is generally heated to melt the lacquer to reduce this friction. Direct cutting to metal disks is also possible.

Disk Manufacture

The original lacquer disk needs to be mass-produced for commercial purposes. The engraved disk can be played back using a phonograph cartridge as described later. The approved disk is electroplated and a metal backed "negative" made. The negative, ridged copy of the cut disk ("master") is then used to manufacture a number of metal copies of the cut lacquer disk. These are used for making metal backed negatives, "stampers", that are used for double-sided hydraulic pressing of a thermoplastic material. The stampers wear out after a number of pressings. The resulting double-sided plastic ("vinyl") copies of the two lacquer disks are then ready for commercial distribution.

13.3 THE PLAYBACK SYSTEM

The playback turntable must be suspended by quiet bearings and have high mass so that any speed irregularities induced by the playback system and the turntable motor drive system (direct, pulley, or belt drive) are eliminated. The vinyl record should rest snugly on the turntable top mat so that any resonances in the plastic disc are damped by the turntable and its mat.

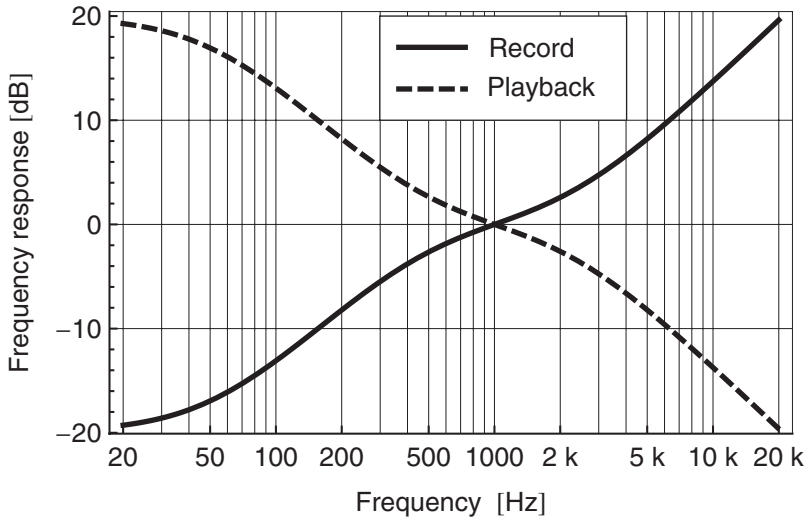


Figure 13.5 The curves shows the RIAA velocity frequency response recording characteristics used in the production of commercial vinyl recordings.

Since the playback equipment is generally placed in the same room as the loudspeakers, the sound pressure levels at the record playback equipment may become very large. The sound will induce vibrations both in the playback equipment and furniture carrying the equipment, as well as in floors and walls. These vibrations will arrive somewhat later than the direct sound from the cartridge and effectively add coloration to the sound from the record. It is important that the tone arm and turntable are constructed in such a way as to reduce such vibrations to a minimum. A rigid stand for the turntable system is the best.

Tone Arms

There are essentially two types of tone arms, those that have the cartridge to travel a curved path across the record, and those that allow the cartridge to travel a radial path, mimicking the way that the cutting head moves across the original cut. Most tone arms are the first type, as shown in Figure 13.6. Such pivoted tone arms must be shaped so that the *angular error* between the groove cut by the recording equipment and the cartridge needle system is minimized. The angular error will depend on tone arm position, length, and shape. It will also depend on cartridge mounting and the geometry of the stylus arm and tip. The linearly tracking tone arm is a modified pivoted arm that rests on a moveable base that can move along the side of the turntable so that the cartridge will move radially acoss the record. Such arms need a servo system to control the position of the moveable base so that the tracking error is kept within some small range.

Any tone arm must have bearings, allowing the arm free movement in two planes without skew or friction. Some tone arms do not use conventional bearings but rest on a *needle tip*, so called unipivot arms. The friction in the tone arm suspension and the compliance in the signal cables between the pick up and the tone arm base, which usually runs through the tone arm, introduces further forces on the cartridge's mechanical tracking system. Since the tone arm cannot be made ideally stiff, its mechanical resonances must be well-damped and preferably above the audio frequency range.

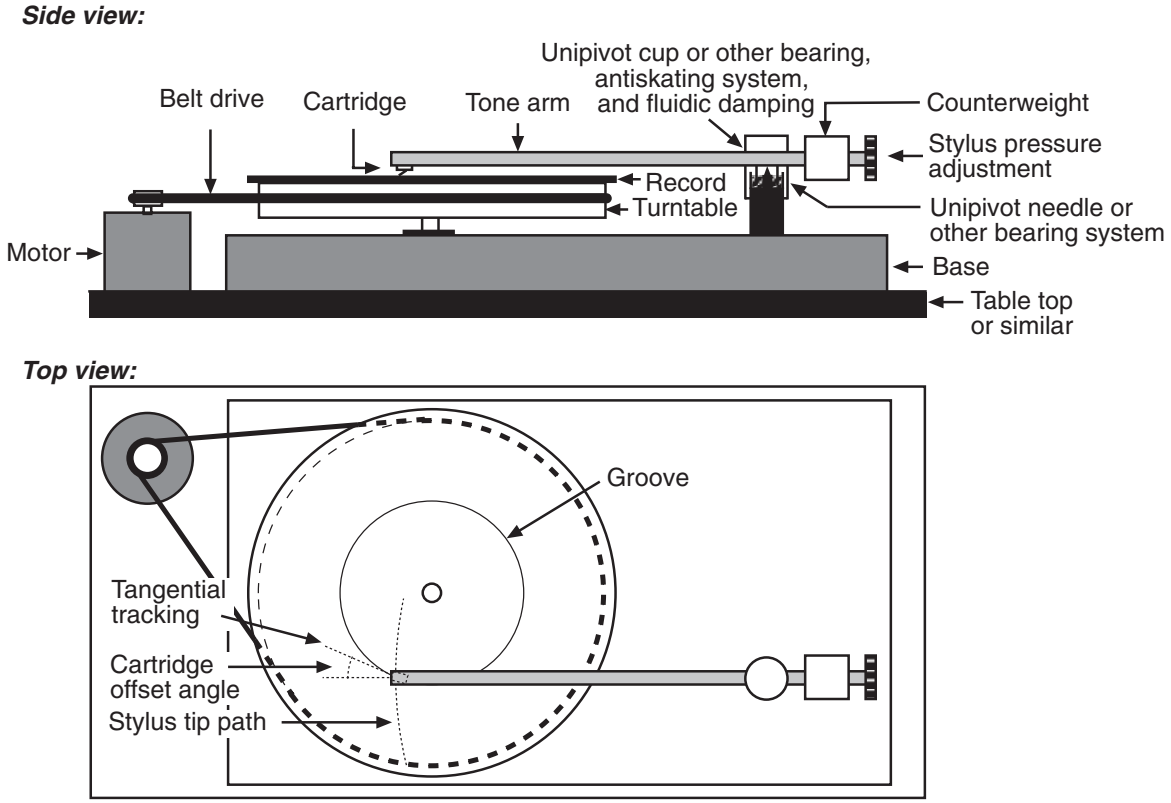


Figure 13.6 Outline of a modern belt-driven turntable with external motor. Special tone arm geometry is needed to reduce the tracking angle error for a pivoted arm since the recording is done with a tangentially mounted cutter head. Since the cartridge is offset by an angle to reduce the tracking angle error, there is an additional *bias* force acting to draw the cartridge toward the turntable center as the disk is rotating. Most arms will have an arrangement for compensation of this bias force. In the case shown, the counterforce is assumed to be achieved magnetically.

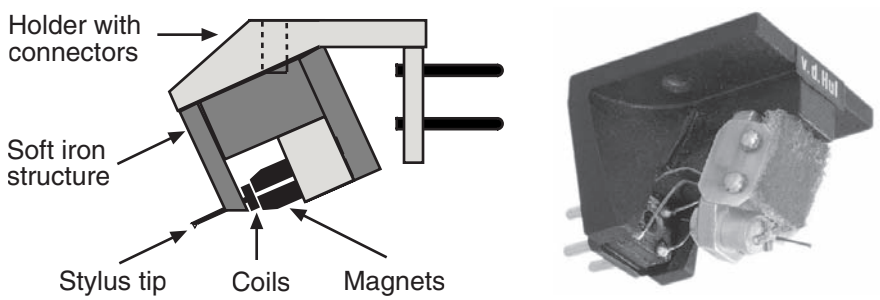


Figure 13.7 Basic components of a moving-coil cartridge (left) and a practical implementation by Van der Hul, Netherlands (right) (photo courtesy of A.J. van den Hul b.v., The Netherlands).

13.4 CARTRIDGES

The playback quality depends on the mechanical properties of the record groove, the stylus tip and arm, the transduction mechanism, the cartridge body, and the tone arm, along with the factors mentioned previously. A high-quality cartridge is shown in Figure 13.7.

At low frequencies, the response of the cartridge mounted in the tone arm will depend on the compliance of the stylus arm suspension and the inertia of the tone arm forming a second order high pass filter. The resonance due to these two components generally has a frequency in the range of 5–20 Hz. It is advantageous to have the resonance at approximately 15 Hz since this will reduce the stylus arm movement due to the vinyl record not being ideally flat and centered. Typically, the disc rotates at a speed of 33 rpm, which means that the lowest frequency induced by record unevennesses will be about 0.5 Hz. The tone arm/stylus resonance may be damped by possible viscous tone arm damping but more often, the resonance is damped because of the mechanical damping in the stylus arm suspension. It is advantageous to keep the Q-value of this resonance low.

The high frequency limit is given by the filter characteristics of the resonant system, consisting of the stylus tip mass and the compliance of the vinyl groove walls, forming a mechanical low-pass filter.

Stylus Tips

To reduce friction and associated disk wear, the *stylus tip* is highly polished. It also needs to have low mass so that the groove wall will not be deformed by repeated playback. The walls of the vinyl groove are compliant but not ideally elastic. There is a pressure limit above which the stylus groove wall will permanently yield and not recover its former shape.

The technical requirements regarding the stylus tip are conflicting. On one hand, the tip needs to have a small radius of curvature to follow the groove well at high frequencies. The cutting stylus is diamond-shaped, so the cartridge stylus needs to have a shape closely approximating the cutting stylus. On the other hand, the cartridge stylus tip must not apply so much pressure on the groove wall while accelerating that the wall yields. The wall will only endure a limited number of playback cycles until it deforms unacceptably. The stylus tip can have a conical, elliptical or other cross-sections. A low pressure requires a spherical tip that has a large radius of curvature. Most current stylus tips have an elliptical lateral cross-section to better track the groove at high frequencies.

The compliance of the groove walls together with the mass of the stylus tip form a resonant system, having a low-pass filter characteristic that removes high frequencies. Typically, this resonance occurs at 20–25 kHz.

Stylus Arms

The *stylus arm* (cantilever) is an elastic beam, typically hinged in a rubber collar. It has the stylus tip at one end and a magnet or coils at the opposite end. The cantilever will have bending wave resonances at various frequencies, preferably only at frequencies above the audio range. Hollow tubes made of synthetic semiprecious materials, such as ruby, make good stylus arms.

The movement of the beam at each resonance frequency will be characterized by a mode shape. The mode shape and mechanical properties of the stylus arm, along with the stylus tip mass, will determine

the *effective mechanical impedance* at the stylus tip. The modal movement of the stylus arm may result in an uneven frequency response. The mode shapes will depend on the shape of the stylus arm and the way it is suspended, the mechanical impedances at the stylus tip (including the groove wall impedances), and the magnets or coils attached to the arm.

The effective stylus mass (given by the effective mechanical impedance) can be regarded as a primary quality metric for cartridges since this mass will be directly proportional to the forces that act on the groove walls. These forces determine the abrasion and other groove deformation processes that occur on playback of the vinyl record. The higher the effective stylus mass, the more difficult it will be for the stylus tip to track the groove. In the extreme case, the tip will jump out of the groove. This can be circumvented by higher static pressure, again resulting in more groove wear.

Electroacoustic Systems

The main problem in designing the electroacoustic system is retaining linearity over a wide range of static stylus tip positions. Different users will use the cartridge at different stylus pressures, which means that the static position of the stylus arm will vary quite a lot. Another problem is obtaining sufficient output voltage so that the preamplifier noise will be negligible.

Some Common Cartridge Types

Most cartridges are *moving-coil* and *moving-magnet* designs, although other designs such as piezoelectric mechanisms were very common some decades ago.

Moving-coil cartridges work according to the electrodynamic principle, having electrically conductive coils moving in a static magnetic field. An outline drawing of a moving-coil cartridge is shown in Figure 13.7. The two coils are attached orthogonally to the rear end of the stylus arm. This results in comparatively low output voltage since there can only be a few coil windings. The mass of moving-coil cartridges is generally higher than that of moving-magnet cartridges, resulting in a lower tone arm resonance frequency, making the system more sensitive to record unevenness.

In a moving-magnet cartridge design, a magnet, positioned at the rear end of the stylus arm, creates time-varying magnetic fields as the arm moves (see Figure 13.8). The magnetic field induces voltages in

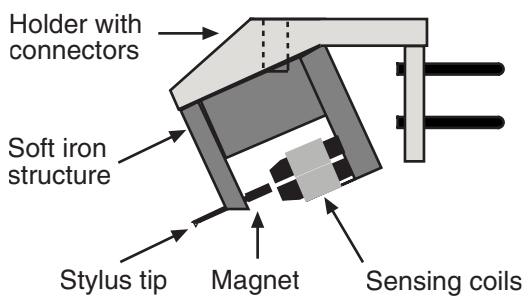


Figure 13.8 The basic design of a moving magnet cartridge.

two coils of electrically conductive wire. The design is quite rugged and has several advantages. Since the static coils can have many turns, the sensitivity can be high, and the output impedance relatively high. The design allows the user of the cartridge to replace the stylus, which is practical since styli need to be replaced after some thousand hours of use.

13.5 AMPLIFICATION

Since the output power of the cartridge is very low, the playback system needs a preamplifier to make the signal level high enough for the audio power amplifier that drives the loudspeakers. The preamplifier/equalizer must have very low noise level. The preamplifier will contain the frequency response correction network discussed previously.

Typically, a preamplifier for moving coil type cartridges needs to be preceded by a step-up transformer that may have a 1:40 winding ratio to increase the output voltage of the cartridge by a factor of 40. This results in an increase in signal-to-noise ratio by about 30 dB. Because of the transformer, the amplifier will also see the cartridge as having a different impedance. Most moving coil cartridges have an internal impedance that is primarily resistive and in the range from 2 Ohms to 50 Ohms. Since the impedance transformation ratio is the square of the voltage ratio, the resulting impedances will be in the range of about 3 kOhm to 80 kOhm. Most audio preamplifiers have an input impedance of 100 kOhm or more in parallel with some small capacitance. With the resulting high impedances, the frequency response of the cartridge may be affected by the filter action of the resistance and capacitance combination, particularly since the transformer will also feature some inductance.

For magnetic cartridges, the noise problem is smaller. On the other hand, the high inductance of the cartridge in combination with the cable capacitance also leads to low pass filter behavior, the damping of which is primarily determined by the preamplifier input resistance. It is always advisable to have short cables between the cartridge and the preamplifier for minimum capacitance and also some possibility for adjusting the load impedance. Since electromagnetically-induced noise *hum* is always present, it is advantageous to use a preamplifier having balanced signal input so that the hum is cancelled.

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Loudspeakers 14

14.1 INTRODUCTION

A *loudspeaker* is an electroacoustic device for converting an electric signal into an acoustic pressure signal retaining the waveform. The loudspeaker's diaphragm moves the surrounding air driven by a force from an electromechanical transducer.

A loudspeaker generally consists of a *loudspeaker driver* (sometimes also known as *motor*) and a *loudspeaker box*. The term loudspeaker is popularly used to describe both the loudspeaker driver and the complete loudspeaker, including the loudspeaker box. The term driver is sometimes used to specifically describe the type of loudspeaker motor used for medium and upper frequency range speakers, known as *horn loudspeakers*.

It is convenient to group loudspeakers according to how the (front) side of the loudspeaker diaphragm radiates sound into the surrounding environment:

- *Direct radiator loudspeakers* are loudspeakers in which at least one side of the diaphragm of the loudspeaker radiates directly (sometimes called direct firing).
- *Horn loudspeakers* are loudspeakers in which the diaphragm radiates through an acoustic filter in the form of an *acoustic transformer*, a *horn*.
- *Nondirect radiating loudspeakers* are loudspeakers in which the diaphragm radiates by way of an *acoustic filter*. These are sometimes called *bandpass* box designs.

The radiation from the second side (rear side in direct radiating loudspeakers) is usually, but not always, prevented by a *box*, *baffle*, or similar construction to prevent an aerodynamic short-circuit of the volume velocity generated by the diaphragm.

The shape and acoustic design of the loudspeaker box contribute to how the loudspeaker driver radiates sound. In addition, there is usually radiation of sound by the vibrations of the box walls. These can be excited both by the loudspeaker driver and the sound field inside the box.

There are many types of drive principles that can be used to generate the force needed to move the diaphragm. The most common designs use electrodynamic, piezoelectric, or electrostatic transduction principles. Loudspeaker drivers have been constructed using modulated ionization of air as well as modulated release of pressurized air. The latter are used for various industrial purposes.

In-depth discussions of the many issues involved in loudspeaker design and use may be found in References 14.1 and 14.2, and in the general references for this chapter.

14.2 RADIATION AND DIRECTIVITY

A simple and instructive way to analyze the radiation by a loudspeaker diaphragm is to use an approximation, where the diaphragm is assumed to be a *vibrating plane piston*. This approximation is relevant for *electrodynamic loudspeakers* in the frequency region above the *fundamental resonance frequency*, due to the *diaphragm mass* and *suspension compliance* of the loudspeaker. For *electrostatic loudspeakers*, it is valid well above the fundamental resonance frequency of the membrane. Most electrodynamic loudspeakers use dome- or cone-shaped diaphragms. The following analysis is reasonably valid also for such diaphragm shapes, provided the dome or cone is reasonably shallow (cone depth much smaller than λ).

In Chapter 9, we studied the sound radiation for the case of a monopole in free space and on a hard surface. In Chapter 1, we found that the sound pressure at a distance r from a monopole having a volume velocity $\underline{U}_D$ is:

$$\underline{p}(r) = j\omega \underline{U}_D \rho_0 \frac{e^{-jkr}}{4\pi r} \tag{14.1}$$

The sound at some distance will be the linear sum of the sound from the monopole and its mirror image, and will be twice that of one free monopole.

Now assume that the loudspeaker diaphragm is a circular piston, which, besides being stiff, is mounted, airtight, in an infinite, hard, immovable plane baffle at $z = 0$, as shown in Figure 14.1. The piston is assumed circular, in the plane $z = 0$, with its center at the origin.

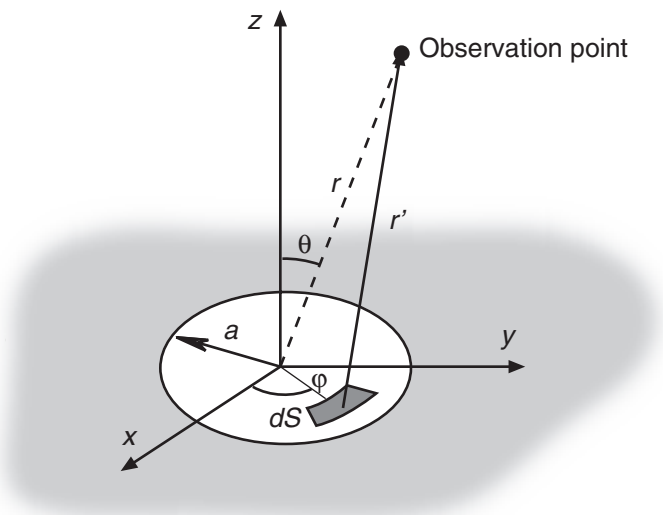


Figure 14.1 The coordinate system for calculation of the sound radiation by a circular piston in an infinite baffle, both in the $z = 0$ plane.

The volume velocity is generated by the piston vibrating in the *z*-direction. When the piston is vibrating, each element on the piston surface area (on both the + and –*z* sides of the piston) can be thought of as an individual, in-phase monopole, generating volume velocity.

Because of linearity, we can again use the principle of superposition of pressure to determine the total sound pressure at the observation point. Since each monopole is on a hard surface, the principle of doubling of pressure due to mirror images also applies.

Each monopole will contribute to the total sound pressure as if it has a part of the total volume velocity $d\underline{U}_D = \underline{u}_D dS$, due to the vibrational velocity of the diaphragm $\underline{u}_D$ and the size of the surface element of each surface element dS. The total pressure can then be obtained by summing the sound pressure contributions $d\underline{p}$ from each surface element dS of the entire piston surface:

$$d\underline{p}(r') = j\omega\left(\underline{u}_D dS\right)\rho_0 \frac{e^{-jkr'}}{2\pi r'} \tag{14.2}$$

where

$$r' = \sqrt{r^2 + \sigma^2 - 2r\sigma\sin(\varphi)\cos(\theta)} \tag{14.3}$$

For points far away from the piston, the integration results in:

$$\underline{p}(r,\theta) = j\omega\left(\underline{u}_D \pi a^2\right)\rho_0 \left[\frac{2\mathrm{J}_1(ka\sin(\theta))}{ka\sin(\theta)}\right]\frac{e^{-jkr}}{2\pi r} \tag{14.4}$$

where J_1 is a Bessel function (due to the cylindrical symmetry). Replacing $\underline{u}_D$ in this expression with $\underline{U}_D/\pi a^2$, one can see that the expression for the total sound pressure at r is the same as that for a monopole on a hard surface, having the volume velocity $\underline{U}_D$, except for a directivity term:

$$\frac{2\mathrm{J}_1(x)}{x} \approx 1 - \frac{x^2}{8} + \tag{14.5}$$

One finds:

$$\underline{p}(r,\theta) = j\omega\underline{U}_D\rho_0 \left[\frac{2\mathrm{J}_1(ka\sin(\theta))}{ka\sin(\theta)}\right]\frac{e^{-jkr}}{2\pi r} \tag{14.6}$$

As mentioned in Chapter 9, the radiated acoustic power can be estimated by integration of the sound intensity for the far-field of the piston, where particle velocity and sound pressure are in phase. One incident for far-field conditions is that the sum pressure diminishes with distance as $1/r$, which is the case at long distance.

The function $F(\theta, \varphi)$:

$$F(\theta,\varphi) = \frac{2\mathrm{J}_1(ka\sin(\theta))}{ka\sin(\theta)} \tag{14.7}$$

is a *directivity function* describing the directional sound radiation in the far-field of the loudspeaker. Because of the symmetry, there is no variation due to the angle φ, for a point source $F(\theta, \varphi) = 1$. (Note that the coordinate system in Figure 14.1 corresponds to that of the microphone studied in Chapter 12.)

The sound pressure close to the surface of the piston has a complicated structure for frequencies over those in which the piston has a size comparable to that of the wavelength. Figure 14.2 shows isobars near a piston for a case where $ka = 10$; that is, the piston diameter is about 3 λ.

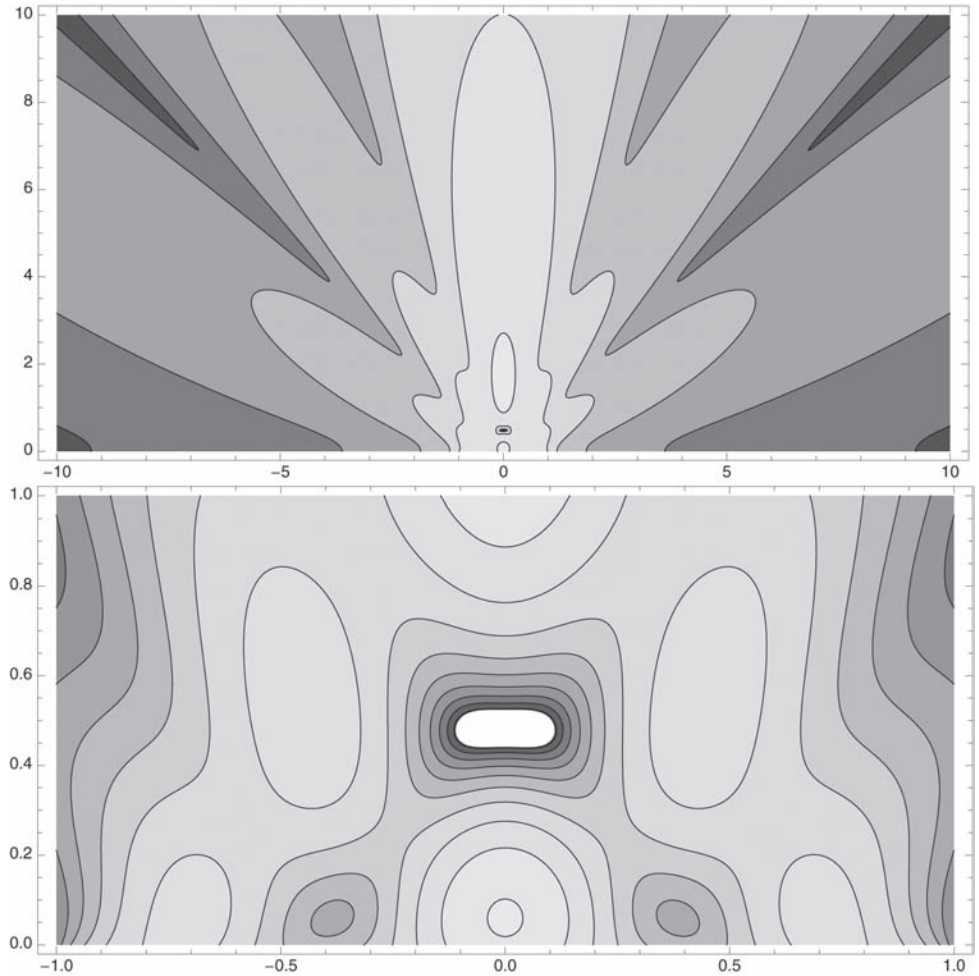


Figure 14.2 Isobars for the pressure magnitude of the sound radiated by a circular piston in a hard baffle for $ka = 10$. The wave number is k, and a is the piston radius. Near- and far-field regions for z from 0 to 10 and a from −1 to 1 (top) and near-field region for z from 0 to 1 and a from −1 to 1 (lower).

The sound pressure on the normal of the circular piston is obtained as:

$$\underline{p}(r,0) = -Z_0 \underline{u}_D \left(e^{-jk\sqrt{r^2+a^2}} - e^{-jkr} \right) \tag{14.8}$$

A commonly used criterion to estimate the shortest far-field distance $r(\lambda)$ is:

$$r(\lambda) > 4\frac{a^2}{\lambda} \tag{14.9}$$

where a is the piston radius and λ the wavelength of sound.

The *directivity D* describes the intensity of the sound in the direction of maximum radiation relative to the intensity at the same distance from a monopole having the same radiated sound power. This is directly analogous to the case of the microphone. The directivity is given by the ratio:

$$D = \frac{4\pi}{\int_0^{2\pi}\int_0^{\pi} |F(\theta,\varphi)|^2 \sin(\theta)\, d\theta\, d\varphi} \tag{14.10}$$

Also note that it is more common and practical to use the *directivity index* (DI) expressed in dB units as a metric for the directionality of the loudspeaker's radiation rather than the directivity. The directivity index is defined as:

$$DI = 10\log(D) \tag{14.11}$$

The radiation pattern from a loudspeaker having a shallow circular cone, mounted in the large flat surface of a loudspeaker box, is approximately described by the directivity function for a circular flat piston in an infinite, hard baffle given in Equation 14.7.

Some directivity patterns for a vibrating plane piston in a hard baffle, given by Equation 14.7, are shown in Figure 14.3 for some values of *ka*. The patterns correspond well to those of practical conical loudspeaker diaphragms having the same effective diameter. It is important to note that different cone profiles have different directivity patterns. The directivity patterns of nonplanar diaphragms are best obtained numerically, for example, by the boundary element method, or more approximately, by numerical integration as outlined initially in this section.

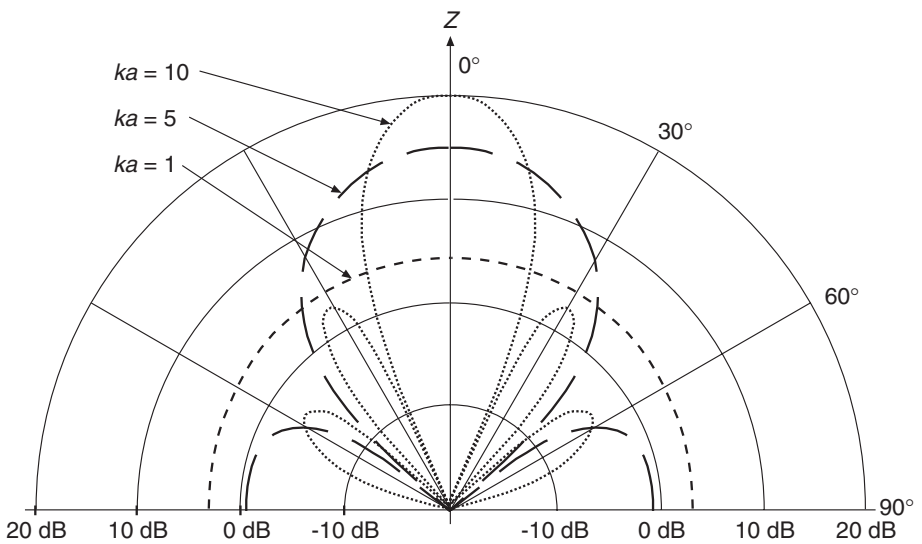


Figure 14.3 The sound radiation by a vibrating circular plane piston is highly directional when the piston diameter becomes large compared to wavelength. This graph shows the far-field directivity patterns for a piston having radius *a* in a baffle for three values of *ka* as a function of the angle θ relative to the *z*-axis in Figure 14.1. The curves show gain in dB over that of a free monopole having the same radiated power as the piston.

The directional behavior of the loudspeaker determines the effective area of good listening, the *sweet spot* in stereo and surround sound reproduction. The directivity of the loudspeaker is in practice affected by how the loudspeaker is mounted, how many loudspeaker drivers are being used, how the signal is distributed between the loudspeaker drivers, and additional factors. The directivity of a loudspeaker is also influenced by the diffraction of sound by the edges of the loudspeaker box.

The directivity is particularly difficult to control in the case of loudspeakers where the entire frequency range is handled by one driver only. Control of directivity by using several loudspeaker drivers is also problematic. For domestic loudspeakers, one usually wants to have a gradual change of directivity over the frequency range, from omnidirectional to more directive. The more reverberant the space, the more important the control of loudspeaker directivity becomes. In extremely reverberant spaces, such as churches and sports arenas, the directivity of the loudspeaker must be very high in the speech frequency range. It can then be necessary to eliminate frequencies where the loudspeaker has insufficient directivity by using suitable filters.

Horn loudspeakers are particularly advantageous when there is a need for high directivity. Horn loudspeakers and horn drivers use comparatively small diaphragms with more piston-like behavior than conventional direct radiating electrodynamic loudspeakers. To have the same radiating area and directivity as a horn loudspeaker, one must use many direct radiating loudspeakers. It is, however, difficult to obtain drivers that are uniform enough to obtain the desired array properties.

Most electrodynamic loudspeakers have very erratic phase and amplitude performance at medium and high frequencies. This is due to the diaphragm no longer vibrating with the same amplitude and phase over its area. At medium and high frequencies, the wave propagation in loudspeaker diaphragms will be as bending and torsional waves, the diaphragm will no longer vibrate as a rigid piston. The bending and torsional waves will set up resonant vibration fields with a modal behavior, popularly known as *breakup*.

In case of breakup, the frequency response of the loudspeaker will be very peaky, characterized by peaks and sharp dips due to the positive and negative interference of sound from the different parts of the diaphragm. The frequency response will also vary when the listening position is changed since the directivity pattern will be poorly controlled. The directivity pattern will often be characterized by many lobes and sharp nulls.

One can use the directivity of loudspeakers for special purposes. All electrodynamic, piezoelectric, and electrostatic electroacoustic transducers are reciprocal—they can function both as transmitters and receivers. Also, since the directivity is a result of interference, the transmit and receive directional responses are reciprocal. The directivity function will be the same, whether a loudspeaker is used to transmit or pick up sound. This property of electrodynamic loudspeakers is often used in intercommunication devices, where the transducer has both roles.

14.3 EFFICIENCY

The sound power required from loudspeakers varies considerably. Typically, the maximum acoustic power available from conventional electrodynamic loudspeakers is about 1 W. Higher radiated powers can be obtained by using a horn or other transformer-coupler, typically 10 W. Loudspeaker efficiency—acoustics power for a certain electrical input power—may be an important parameter in some applications.

Loudspeaker sensitivity is usually given as sound pressure at some distance from the loudspeaker for a certain input voltage, usually as sound pressure level on-axis at 1 m distance for an input voltage of 2.83 V in an anechoic environment.

There are several ways to express the sensitivity and efficiency of a loudspeaker system. One common way of writing the electroacoustic efficiency η is to use the ratio between acoustic output power and electric input power as:

$$\eta = \frac{W_{\text{acoustic}}}{W_{\text{electric}}} \tag{14.12}$$

The sensitivity can be expressed by the particular sound pressure level that can be obtained in a certain direction for a given input voltage or power.

Most direct radiating electrodynamic loudspeakers have efficiencies in the range of 0.1 to 5%. Using horns or other impedance transformers, one can increase the efficiency substantially. Horn loudspeakers often have efficiencies in the range of 5 to 15% in commercial designs.

Electrostatic loudspeakers have nominal efficiencies close to 100%, but the losses in transformers and amplifiers are substantial because of the reactive capacitive electrical load of the loudspeaker.

14.4 FREQUENCY RESPONSE

The acoustic load on a small sound source by sound field impedance, the *radiation load*, is primarily reactive and has a mass-type character for distances from the origin where $kr \ll 1$, as shown by Equation 2.27. For distances such that $kr \gg 1$, the sound field impedance is primarily resistive. Since the radiation load by the air influences the mechanical system (it adds to the mass and resistance in the equation of motion), and since the radiated power depends on both the vibrational velocity of the loudspeaker diaphragm and the real part of the acoustic load on the diaphragm, the loudspeaker driver must be designed with this in mind.

Most high-quality electrodynamic drivers are designed to operate mainly in the region where the diaphragm radius, a, is much smaller than the wavelength, so that $ka \ll 1$—that is, the radiation load is primarily reactive and its resistive component increases proportionally to frequency squared. As long as the diaphragm is small compared to wavelength, the mass load due to radiation will be constant.

The mass component of the radiation load may be as large as the mass of the loudspeaker diaphragm itself. This puts a limit to how thin and weak the diaphragm can be. If the diaphragm is weak, it may buckle under the radiation load. This, of course, leads to considerable nonlinear distortion.

In the $ka \ll 1$ frequency region, the sound intensity, as a function of frequency, in the radial direction in the far-field will be:

$$I = \frac{p^2}{Z_0} = \frac{\rho_0 a^4}{8c_0}\left|F(\theta,\varphi)\right|^2 \frac{\omega^2}{r^2} u_D^2 \tag{14.13}$$

As long as the magnitude of the directivity function $F(\theta, \varphi)$ is close to unity, the vibrational velocity must diminish proportionally to the frequency for the sound intensity at the listening point to be frequency-independent.

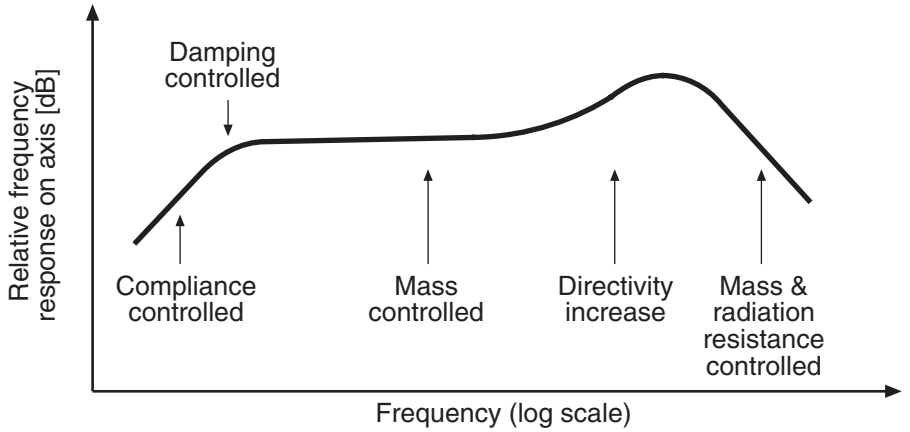


Figure 14.4 Idealized frequency response curve for an electrodynamic loudspeaker.

This, in turn, means that the vibrational velocity must be determined by the vibrating mass, which is the sum of the diaphragm mass and the mass caused by the radiation load. *Mass-controlled* operating conditions can only be achieved if the resonance frequency of the mechanical system is well below the lowest frequency at which the loudspeaker is designed to operate.

At high frequencies, the intensity will diminish because the radiation load no longer compensates for the decrease in vibration velocity with increasing frequency.

The directivity, however, will start increasing already for frequencies at which the diameter of the diaphragm is smaller than the wavelength, which will result in an increase of sound intensity on the symmetry axis of the loudspeaker.

The different factors will influence the frequency response of the loudspeaker driver so that its typical frequency response will be that shown in Figure 14.4 (see also Problem 14.4). Various ways of adjusting the frequency response to be more linear will be discussed later in this chapter.

14.5 ELECTRODYNAMIC DRIVERS

The electrodynamic loudspeaker driver can be given a robust, simple, nondangerous, and inexpensive design. These advantageous properties, over other designs, such as piezoelectric and electrostatic designs, in combination with sufficient acoustic power output, reasonably good frequency response, and good directivity, have contributed to the almost total market dominance by electrodynamic drivers.

An *electrodynamic driver* uses the force generated by an electric current in a wire placed in a magnetic field. The design is usually circularly symmetric, as shown in Figure 14.5, using a coil of metal wire, the *voice coil*, in the magnetic field provided by a *permanent magnet*.

Besides the coil, the driver also consists of a *diaphragm* firmly attached to the coil, both mounted in a stiff *basket* (usually made of stamped sheet metal), which also holds the magnet. The diaphragm is attached to the basket using two concentric compliance elements, the *suspension*.

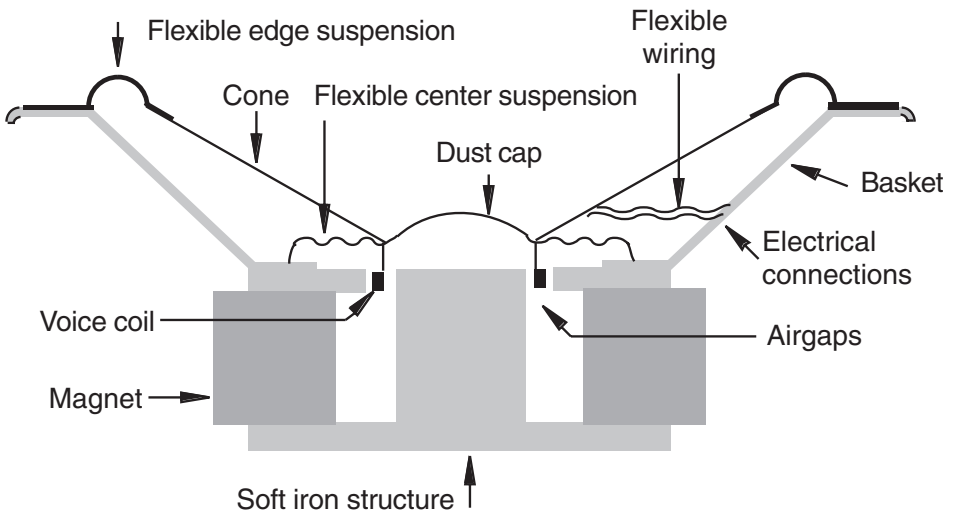


Figure 14.5 A cross-section through a typical electrodynamic driver having a concave conical diaphragm.

The force $\underline{F}_{vc}$, due to the interaction between the dynamic magnetic field of the current $\underline{i}$ through the voice coil wire and the permanent magnetic field is:

$$\underline{F}_{vc} = Bl\,\underline{i} \tag{14.14}$$

where B is the flux density of the magnetic field and l the length of the voice coil wire. The Bl-product is an important property.

The electric current through the coil is determined by the voltage applied to the coil, the electric resistances, inductances, and capacitances in the wiring and filters connected to the loudspeaker terminals and by the voltage induced in the voice coil due to its motion in the magnetic field. The field tries to keep the coil still.

This force is used to set the mechanical and acoustical system in motion. If the acoustic system is included in the mechanical system, the force equation is:

$$\underline{F}_{vc} = M_M \underline{a}_D + R_M \underline{u}_D + \frac{\underline{\xi}_D}{C_M} = \left(j\omega M_M + R_M + \frac{1}{j\omega C_M} \right) \underline{u}_D \tag{14.15}$$

where $\underline{a}_D$, $\underline{u}_D$, $\underline{\xi}_D$ are the acceleration, velocity, and displacement of the voice coil. Further, M_M is the total mass of the voice coil, the diaphragm, and the air load (the mass part making up the reactive part of the radiation impedance). The losses due to radiation are included in the mechanical resistance R_M along with the viscous losses in the air gaps and losses at the loudspeaker suspension. The compliance C_M includes compliance of the suspension and air inside the loudspeaker box. The compliance C_M is the inverse of the system's spring stiffness k_M:

$$C_M = \frac{1}{k_M} \tag{14.16}$$

Knowing the force $\underline{F}_{vc}$, we can calculate the resulting velocity $\underline{u}_D$ of the coil and diaphragm assembly. Since we know the real part of radiation impedance, we can then calculate the power radiated, the intensity at some distance, and the sound pressure $\underline{p}(r, \theta, \varphi)$.

Diaphragm Shape

The diaphragm and the magnet assembly are the two components that have the greatest influence on the sound quality of the loudspeaker.

The vibration and radiation properties of the diaphragm are affected by the following factors:

- Material thickness, density, Young's modulus
- Geometry and inhomogeneities
- Diaphragm profile (flat, straight cone, hyperbolic cone, corrugated circular sections, and such)
- Voice coil mass and attachment
- Voice coil placement
- Suspension mass, stiffness, losses and geometry
- Air load on both sides of the diaphragm

The shape of the diaphragm, its geometry, density, and elasticity influence the sound reproduction properties so much that many loudspeakers are similar except regarding the diaphragm. It is also important to note that inner and outer suspension rings of conventional electrodynamic loudspeakers also radiate sound.

Some high quality loudspeakers have diaphragms that are ideally stiff over most of the desired frequency range, so the diaphragm will move in phase over its entire surface. These loudspeakers tend to become very directional as the wavelength of sound is comparable to, or larger than, the diaphragm radius. The diaphragms of such loudspeakers are usually shaped as deep cones. Because of the depth, there will be considerable interference between the radiation from the inner, center, and the outer parts of the cone, resulting in frequency response dips already at quite low frequencies.

There are also high-quality loudspeakers where the diaphragm is designed to vibrate as a membrane with well-damped bending wave motion (see Figure 8.14). By controlling the diaphragm modes and their damping, one can achieve a fairly uniform frequency response and directivity, particularly if one is only concerned with the response in the reverberant sound field of a room. Loudspeaker manufacturers using this technique usually design diaphragms that are shallow cones or have flat circular or rectangular shapes.

Flexible cones have long been popular in the construction of high quality loudspeakers. Using flexible diaphragm materials (featuring suitably damped bending wave motion), one can achieve nonresonant behavior, and the effective cone area will shrink as the frequency is increased. This simplifies the use of the loudspeaker in multi-loudspeaker systems, where different loudspeakers are used for different frequency ranges. To control the directivity properly, one wants the loudspeakers to have about the same effective area in the crossover frequency region.

Some common cone shapes and profiles are shown in Figure 14.6. The use of elliptical cones makes it possible to increase the cone area in some applications where a circular cone would not use the available area optimally. The elliptical shape can also be used to give the loudspeaker different directivity

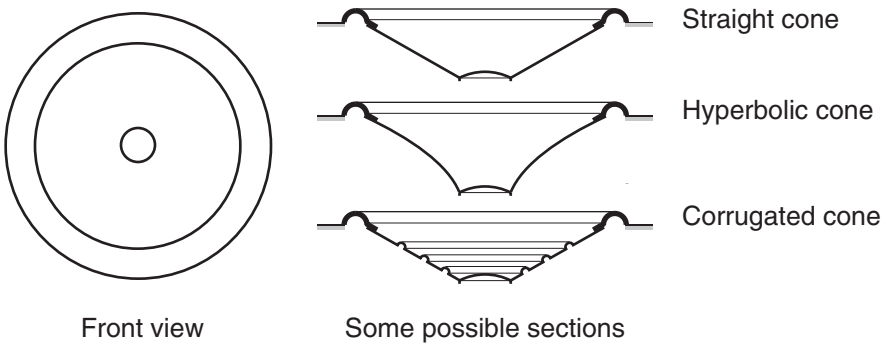


Figure 14.6 Some examples of typical diaphragm profiles.

characteristics in the horizontal and vertical planes. If the loudspeaker diaphragm moves as one unit, its directivity characteristics will be similar to those of the plane piston discussed in Section 14.1. A conical diaphragm will show similar but smaller directivity than the piston. The theory in Section 14.3 can be used to calculate the directivity of cones and other diaphragm shapes that are not too deep. For deep cone shapes and when the exact motion and directivity pattern is desired, one can use finite element or boundary element modeling to design the diaphragm and the associated directivity pattern. Finite element modeling can be used also to model possible buckling and other nonlinearities.

The choice between convex or concave diaphragms is determined to some part by the maximum allowable thickness of the loudspeaker. Concave diaphragms give less directional radiation than do convex ones, but convex diaphragms can sometimes be used to house and hide the magnet in shallow loudspeakers.

Many midrange and high frequency loudspeakers use dome shaped diaphragms. Looking at Figure 14.7, one can note that the directivity is much less dependent on dome depth or height for concave domes than for convex domes (that bulge outwards).

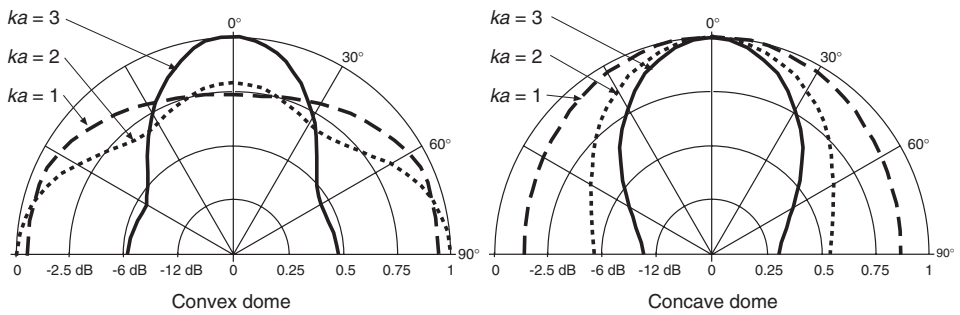


Figure 14.7 Directivity patterns, $F(\theta)$, for the sound radiation by vibrating convex and concave dome diaphragms for various values of ka. The wave number is k, and a is the dome radius. For the case shown, the dome radius and the dome height/depth ratios were constant. The dome vibration was in the z-direction shown in Figure 14.1. Note the linear scale. (Adapted from Ref. 14.4)

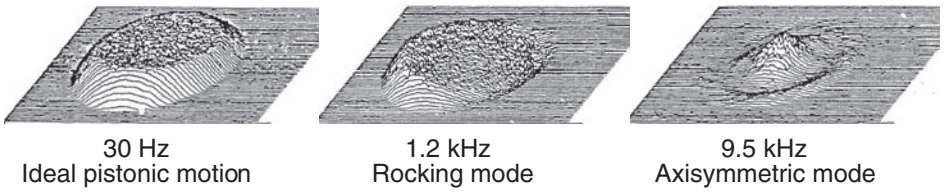


Figure 14.8 Laser scans of the vibration displacement in a 25 mm diameter convex soft-dome diaphragm showing pistonic motion and mode shapes for two of its bending wave resonances. (from Ref 8.2)

Domes, as well as cones, display resonant modal behavior. Domes have to be damped appropriately, the same way as conical diaphragms. Examples of mode shapes at some resonances of a loudspeaker dome diaphragm are shown in the laser vibrometer scans in Figure 14.8.

The mechanical compensation of improper loudspeaker diaphragm behavior is the economically most attractive and can take many forms, such as:

- Progressive vibration isolation of the cone
- Use of small, concentric extra cones in combination with vibration isolation of the main cone
- Use of soft, flexible cones with high damping

All of these actions have the goal of reducing the vibrating and radiating area at medium and high frequencies.

Diaphragm Materials

Most simple loudspeakers use conical diaphragms made of thermoformed plastic out of sheet stock or paper pulp that has been dried to conical shape. Such loudspeaker cones have a poor rigidity and yield poor frequency response.

Lightweight, thin cones cannot be stiff unless they are shaped as deep cones. The original cone shape was due to the cones being made from rolled paper. Today, paper cones are usually made by shaping and drying paper pulp with additives, such as animal hair, glue, and softeners. This technology makes it possible to design cones in various geometries with curved or flexible sections. The main advantage of using cones made of paper and felt mixtures is that the damping can be made much higher than in homogenous plastic cones. Because of the reasonably high Young's modulus of paper, one can make sandwich constructions of two sheets of paper with a constrained viscoelastic layer.

A somewhat different technique is applied when making diaphragms of woven materials. Traditionally, resin-impregnated cloth was used to cover the center of the cone so that dirt would not enter the voice coil gap. Dome diaphragms have long been made of impregnated cloth since it was possible to make soft domes in that way. Coarsely woven Kevlar cloth, impregnated using soft plastics, has been used successfully in diaphragm design. The coarse cloth can be made to use the viscous damping properties of the plastic in a good way.

An advantage of plastics is that the material can be formed by injection molding or thermoforming processes, which are much faster and simpler than the paper pulp process. Since plastics are more

homogeneous between batches than paper, one can have more similar properties and higher production yield. Typical plastics are polystyrene and polypropylene. The disadvantage of these plastics is their low internal damping that cannot be increased much by added external damping compounds. A better plastic for loudspeaker diaphragms is talc or mica-filled polypropylene. The addition of solid compounds helps in keeping the damping high and constant over a wide temperature range. Conventional plastic-based viscoelastic materials only function over a narrow temperature range and are not suitable for loudspeakers for automotive use where the temperature range is typically −40 to 80°C.

Over some frequency, the poorly damped bending wave motion of the cone will lead to resonant behavior due to the reflection of bending waves from the edges where the diaphragm meets the suspension. This leads to uncontrolled resonances and modal behavior—*breakup*. Many diaphragm constructions are also characterized by noncircular modes excited by asymmetrically mounted wire leads, and such (see Figure 8.1).

The resonant behavior can be minimized by using various damping compounds on the diaphragm or by using diaphragms made of materials with high inner damping.

Diaphragms made of sandwich-type materials can also be used for loudspeakers. A typical sandwich panel is made of two outer layers of stiff material, such as plastic, with an intermediary honeycomb structure. The honeycomb structure has low bending stiffness. Such materials have long been used in ship and aircraft design to build stiff, lightweight walls and structures.

If the diaphragm material is correctly designed, the wave speed in sandwich materials can be tailored to follow a different rule than for solid plates. This makes it possible to design sandwich panels having an increased modal density compared to plates having bending waves.

Diaphragm Supports and Surrounds

The diaphragms of most electrodynamic low- and midfrequency range drivers need to make large displacements to radiate sufficient low-frequency energy. Maximum diaphragm displacements of large long-throw driver designs are approximately ± 1 cm. For closed loudspeaker boxes, the maximum diaphragm displacement can be calculated from knowledge of the diaphragm area and the desired sound level at the lowest frequencies the loudspeaker is supposed to reproduce. Various types of tricks can be used to reduce the need for long diaphragm travel, such as the use of horn loading and acoustically resonant boxes (for example, bass-reflex loudspeaker boxes).

High-frequency driver diaphragms move much less than those of low- and mid-frequency range drivers. This makes two different types of suspension designs necessary. In long throw designs, the diaphragms are usually quite heavy; diaphragm mass of about 10 to 100 grams is common.

It must be possible to ship and handle commercial loudspeakers. For conventional drivers using conical diaphragms, there will be an outer suspension and an inner suspension (see Figure 14.5). Unless the diaphragm is supported by two concentric suspensions, it will not be possible to design the loudspeaker driver in such a way as to avoid the vibrating voice coil being displaced off-center in the magnetic field. In the worst case, the voice coil will be touching the walls of the magnet in the air gap. If the voice coil touches the gap walls, there will be nonlinear friction, since the voice coil walls are usually quite roughly machined. The effects of such nonlinearity are heard as *buzz*, not as regular harmonic distortion.

Typically, short-throw designs are found in treble drivers, mobile phone loudspeakers, and in earphone drivers. Such drivers are characterized by low-mass, dome-type diaphragms and are designed to have resonances in the medium to high-frequency range. This makes it possible to use single-ring suspensions that are stiff enough to keep the voice coil from being catastrophically statically displaced in the magnetic field.

The compliance characteristics of the suspension must be both linear and nonlinear. In the linear operating range of the suspension, it must have the correct compliance to give the desired resonance frequency and not show any hysteresis. The suspensions must also prevent the voice coil from leaving the magnet air gap; this is usually the task of the inner suspension. It is difficult to design loudspeaker suspensions with just the right amount of nonlinearity so that the voice coil movement will be linear in the main operating range.

The inner suspension is usually made of impregnated cloth so that the air trapped between the inner suspension and the magnet assembly is free to escape. The inner suspension must have high radial static stiffness so that the voice coil is centered in the air gap. In some experimental designs, the inner suspension has been made of metal rods, which makes it possible to achieve higher linearity and less hysteresis. Some disadvantages of this type of approach are that dirt can enter the air gap and that the rods need to be provided with viscoelastic damping to remove bending wave resonances in the rods themselves.

The outer suspension, the *surround*, has two tasks. It should keep the cone in place and introduce sufficient damping to the cone rim so that resonances due to radial and circular bending wave propagation in the cone are prevented.

In guitar loudspeaker drivers, the outer rim is usually made as corrugated continuation of the cone. This type of surround has comparatively high stiffness and low losses, which causes severe bending wave reflections at the rim. It is also a common suspension design in single suspension earphone and treble drivers.

High-quality, low-frequency and midrange drivers using cone diaphragms usually have the type of outer suspension shown in Figure 14.9, often called *roll surround*. Roll surround allows much larger and more linear movement than the corrugated surround. The roll surround is usually made of nitrile rubber, impregnated cloth, foam plastic, or foam rubber. The choice of material is critical for the durability of the loudspeaker. Surround *rot* is a common cause of loudspeaker failure and can be caused by ozone and exposure to sunlight.

An advantage of the roll surround is that it can be combined with vibration damping at the outer rim of the cone. A suitable flat section of the surround is allowed to cover the outer end of the cone. This section, if designed properly, can be made to act as an anechoic termination for the radial bending waves

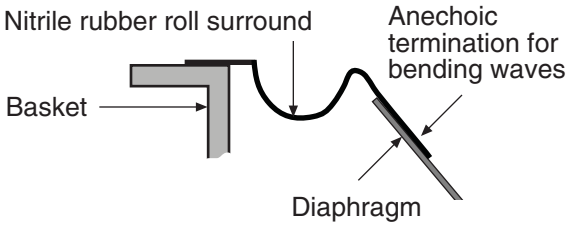


Figure 14.9 An example of a *roll surround*-type suspension designed to act as a vibration absorber.

in the cone (see Figures 8.12 and 8.14). This will prevent the formation of resonances due to reflection of bending waves at the rim.

Voice Coil and Magnet Air Gap

The magnetic system in an electrodynamic loudspeaker consists of the dynamic magnetic field induced by the voice coil and the static magnetic field in the magnet's air gap, which interact to give the force that moves the voice coil. Some modern loudspeaker drivers use multiple voice coils and air gaps to improve the force linearity.

The *voice coil* is the center of the loudspeaker driver. A stiff *voice coil former* is attached to the diaphragm and carries the voice coil wire windings. It is important that the voice coil former and leads are attached in a particular way to the diaphragm so that they do not excite unwanted bending waves in the diaphragm. Specially formed copper, silver, or aluminum wire is usually used for the voice coil. Typically, the wire cross-section is hexagonal or rectangular so that the electrical resistance of the loudspeaker can be reduced. The target is to have a voice coil with low mass and low electrical resistance that uses little volume.

The former must contribute to the cooling of the windings, thus, metal voice coil formers are preferred for high-power applications. In high-power, low-frequency drivers, the voice coil is often made of aluminum or copper and is cooled by incorporating special perforations in the voice coil former to increase turbulence and air movement. A common problem in loudspeakers is that the voice coil overheats, which makes the voice coil lacquer soften and lose its grip on the voice coil wire. This results in rub and buzz and ultimately makes the loudspeaker fail.

The voice coil winding must have a height that is suited to the design of the magnetic field. In principle, it is possible to compensate for magnetic field and diaphragm suspension nonlinearities by special voice coil designs, as shown in Figure 14.10. Typically, the voice coil has a height that is slightly less than that of the air gap. Low-frequency drivers sometimes have long voice coils, as shown in Figure 14.10c, since these drivers are less sensitive to high voice coil mass and need to be able to generate greater volume velocity.

The voice coil resistance is determined by the coil winding. An electrical resistance in the range of 3 to 7 ohm is typical. The resistance must be chosen so that the power losses in the wires leading to the loudspeaker are small compared to the losses in the voice coil winding, which encourages the use of high resistance circuits. At the same time, a high voice coil resistance is a disadvantage from the viewpoint of amplifier design, particularly in automotive applications.

The voice coil electrical impedance is a result of both the voice coil resistance and the inductance of the coil and the magnetic circuit, and typically behaves as shown in Figure 14.11. The inductance will

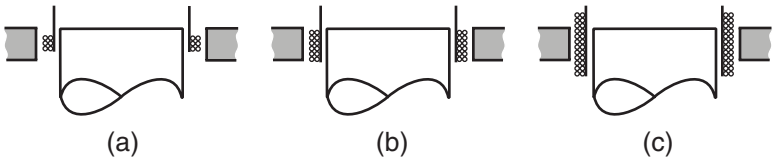


Figure 14.10 Some examples of short, medium, and long voice coil designs.

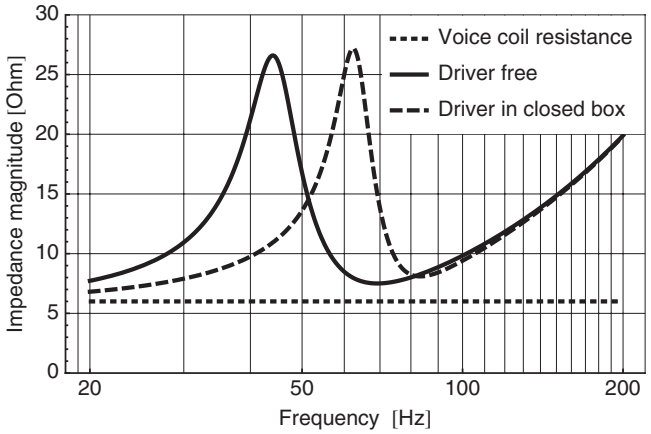


Figure 14.11 Typical behavior of the modulus of the electric input impedance Z_{el} of an electrodynamic loudspeaker.

contribute to a rise of the impedance at high frequencies. At low frequencies, the back electromotoric force (emf) of the voice coil movement will result in one or several impedance peaks. The frequencies and levels of these peaks can be used to trim the acoustic design of the loudspeaker box.

The static magnetic field in the air gap of the electrodynamic loudspeaker driver is provided by a magnet and a soft iron magnetic circuit. Figure 4.12 shows some magnet circuit designs. The desired magnetic field is determined by the mechanical system, the electrical system, and the loudspeaker box. All of these factors influence the loudspeaker sensitivity, the resonance frequency, the damping at resonance, and the suitability of the loudspeaker for a particular loudspeaker box construction. The magnetic circuit of high quality drivers must be designed so that the field in the air gap can be kept within small tolerances.

To have sufficient magnetic flux density in the air gap, the gap has to be narrow. Typically, the voice coil is cooled by the magnet structure, and a narrow air gap is necessary to be able to cool the voice coil properly. If the air gap is too narrow, the unavoidable sideways movement of the voice coil will lead to contact between the air gap walls and the voice coil resulting in rub and buzz sounds. This is a common problem in loudspeakers for mobile phones. A wide air gap will result in a weak magnetic field and low loudspeaker sensitivity. By adding a magnetic fluid consisting of a suspension of ferrite particles in a viscous *ferrofluid*, one can have a somewhat wider air gap and still reasonable flux density. This is a com-

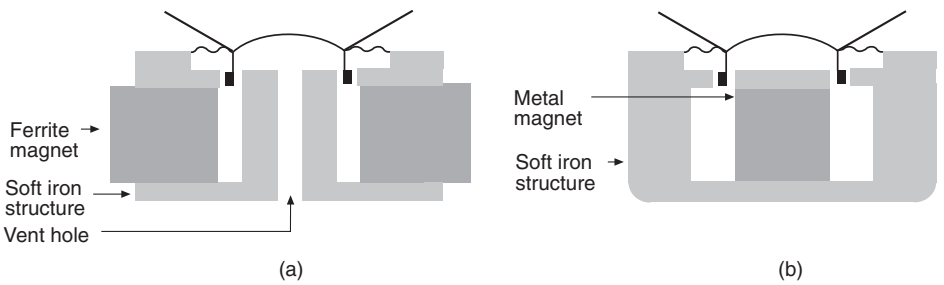


Figure 14.12 Two different magnet systems for electrodynamic loudspeakers: (a) ferrite magnet typical for low-frequency drivers and (b) metal magnet (for example, Neodymium) typical for high-frequency drivers.

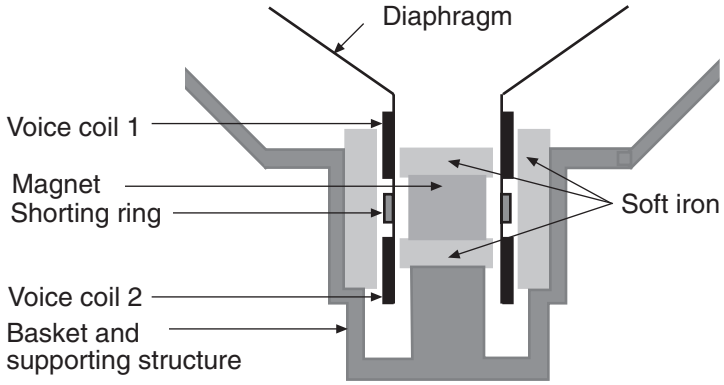


Figure 14.13 The basic design of a *dual drive* driver combining the action of two voice coils to reduce nonlinear distortion due to magnet field asymmetry. (After Ref. 14.7)

mon approach used in midrange and treble drivers. The method is unsuitable for low-frequency drivers because of the viscosity of the fluid. The ferrofluid also contributes to better cooling of driver voice coils. A disadvantage of this approach is that the oil emulsion-based magnetic fluid has a tendency to dry up.

An important addition to the magnet system is the shorting ring that acts as a secondary transformer winding, and reduces the magnetic field modulation in the magnet system induced by the voice coil's magnetic field. The position of this shorting ring is critical for reduction of nonlinear distortion in the loudspeaker. The shorting ring also reduces the inductance of the voice coil.

Another approach to reducing the nonlinear distortion is to use dual magnetic drives. The aim in using dual drives is to provide drivers with less mass, reduced stray magnetic fields, and that are more linear—that is, have less nonlinear distortion. Figure 14.13 shows the basic construction of such a driver. Note the shorting ring shown in the figure.

14.6 LOUDSPEAKER ENCLOSURES

Any finite vibrating plate will have a frequency below which the diffraction of sound from the rear side of the panel will interfere to reduce the sound pressure at a distance. This phenomenon is known as *aerodynamic short circuit*. Since it is impractical to build electrodynamic loudspeaker drivers with diameters larger than 30 cm, the low-frequency limit would become unacceptably high. This makes some form of acoustic isolation between the front and back side of the diaphragm necessary.

The isolation can be obtained in several ways, depending on design compromises. Three basic approaches exist: the use of *baffles*, *boxes*, and *horns*.

Baffles

The purpose of a *baffle* is simply to prolong the path that the rear radiated wave has to travel. For a baffle to be effective, it must be about as large as the wavelength at the lowest frequency one wishes to reproduce. The frequency response of a loudspeaker mounted in two different positions in a square baffle is shown in Figure 14.14.

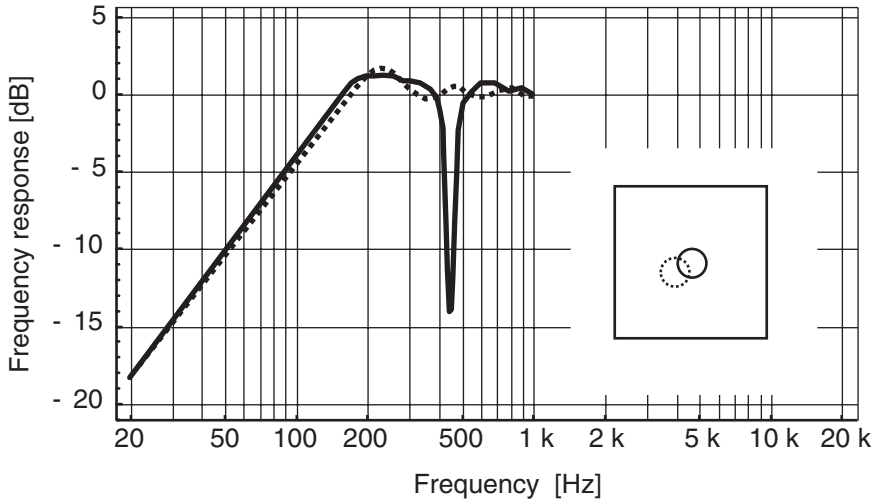


Figure 14.14 Two examples of frequency response curves for a loudspeaker mounted on center (solid line) and asymmetrically (dashed line) in a square baffle having 1.2 m long sides. (After Ref. 14.3)

Since one is usually listening at a distance of a few meters in homes, a large baffle will, in addition, contribute toward making the rear wave more attenuated than the front wave due to the longer travel distance.

The diffracted wave can be attenuated to some extent by providing a sound-absorptive layer on the rear side of the baffle. A sound-absorptive layer on the front side of the baffle would somewhat reduce the sound pressure at distance because of less pressure doubling due to the acoustically softer surface. Sound-absorptive treatments of this type are common to reduce the front side diffraction on loudspeaker boxes.

The baffle must have sufficient mass and rigidity so that it is not excited by the loudspeaker. This requires quite heavy baffles. Baffle resonance will lead to reduced sound insulation between the front and rear side of the baffle and to baffle sound radiation that might interfere with the direct sound from the loudspeaker.

An advantage of the baffle approach is the absence of acoustical resonance in the air. This is a particularly useful advantage when using loudspeakers that have light membrane-type diaphragms and thus lack sound insulation. Electrostatic loudspeaker panels, which use low mass, flexible membranes, are often mounted in baffles for this reason.

Loudspeaker baffles require particular attention when positioned in rooms due to sound cancellation by the rear radiation. In contrast to conventional loudspeaker boxes, which usually obtain the best low-frequency coupling to the sound field in the room when mounted in corners, baffle loudspeakers are best mounted with the transducer close to a side wall.

Loudspeaker Boxes

Loudspeaker boxes are designed to contain or control the rear radiation of the loudspeaker, so that it does not interfere, but couples optimally, with the front side radiation of the loudspeaker driver. Some common box designs are shown in Figure 14.15.

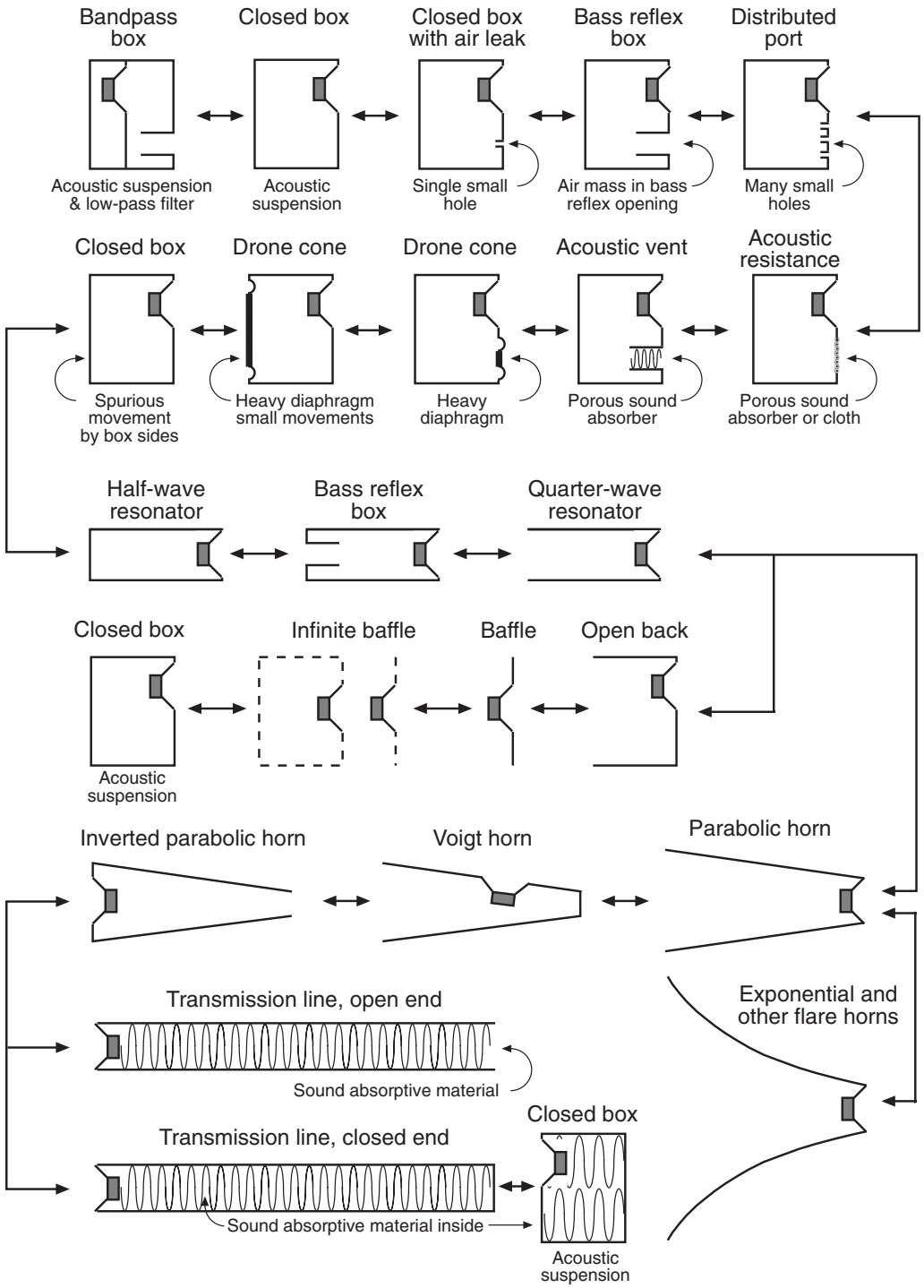


Figure 14.15 Some common loudspeaker box designs and their interrelation. (After Ref. 14.5).

A loudspeaker mounted in a bounding wall of a room that has short reverberation time effectively sees the room as an infinite box, well above the lowest resonance frequencies of the room. If the room is suitably damped, the rear radiated sound will be so weak that it will not be heard through the loudspeaker. If the loudspeaker is mounted in a wall separating two large rooms, the wall can be considered as an approximation to an *infinite baffle*. The resonance frequency of the mounted driver will not differ much from that of the unmounted driver. Sometimes a closet or something similar can be used as an approximation to a second, large room.

In practice, however, it is usually easiest to mount the driver in a box, since the loudspeaker can then be moved with respect to room modes, decor, and the like. Such a box can be optimized in various ways. The ideal loudspeaker box has stiff walls and is an immovable support for the driver. In addition, it does not give off any diffraction from its edges, it does not reflect any high-frequency sound, and it provides a known, well-defined acoustical impedance at low frequencies.

Real loudspeaker boxes can only behave in this desirable way to some extent. The loudspeaker box also has limited mechanical impedance; it can be set in vibration not only by the sound inside the box, but also by the driver itself. At some frequencies, the walls of the loudspeaker will be essentially transparent to sound. The loudspeaker exterior will diffract sound from its edges, unless it has a suitable sound-absorptive front.

On the other hand, it is possible to use the loudspeaker box in a constructive way, particularly for low-frequency loudspeakers. One can use acoustical circuits, such as resonant elements, in the paths of the rear and front side radiation. Such circuits can passively attenuate the radiation to reduce nonlinear distortion and tailor the frequency response. Another advantage of such passive acoustical impedances is that they can be used to improve the radiation in certain frequency ranges so that the cone displacement can be reduced, again reducing the nonlinear distortion.

The simplest box design is the *closed box*. The resonant behavior of closed volumes is known to us from room acoustics. The loudspeaker box is a small room and, depending on the loudspeaker design tradeoffs, the interior volume may be either damped or undamped. The resonance frequencies of an undamped rectangular room can be easily calculated. At very low frequencies, well under the first resonance frequency, any closed loudspeaker box will appear as an acoustical compliance element, irrespective of its shape. The acoustical compliance is determined by the volume of enclosed air given by Equation 1.77. In order to estimate the influence of the enclosed air on the behavior of the mechanical system, it is useful to convert the compliance into an equivalent spring stiffness.

We must convert the acoustical compliance $C_{A\text{box}}$ into mechanical compliance $C_{M\text{box}}$. Since the mechanical compliance is the ratio between displacement and force and acoustical compliance is the ratio between air volume displaced and pressure, we find:

$$C_{M\text{box}} = \frac{C_{A\text{box}}}{S_D^2} \tag{14.17}$$

Since the total stiffness k_M is the sum of the stiffness of the diaphragm suspension k_S and the stiffness of the volume V of air inside the loudspeaker box $k_{A\text{box}}$, we find:

$$k_M = k_{M\text{box}} + k_S \rightarrow \frac{1}{C_M} = \frac{1}{C_{M\text{box}}} + \frac{1}{C_S} \tag{14.18}$$

where k_S and C_S are the stiffness and the compliance of the diaphragm suspension respectively.

Thus, the mechanical compliance C_M of the loudspeaker and closed box is:

$$C_M = \frac{1}{\frac{1}{C_S} + \frac{\rho_0 c_0^2 S_D^2}{V}} \tag{14.19}$$

Sometimes manufacturers publish the driver compliance in the form of its equivalent acoustical volume. This is useful information since, if the driver is used in a closed box having that volume, the resulting driver resonance frequency will be about 1.4 times that of the unmounted driver. This volume is typically the minimum recommended for a closed box loudspeaker design.

Filling the interior of the loudspeaker box with sound-absorptive material will increase the compliance. This is due to the nonadiabatic behavior of sound as it moves through a medium that has heat exchange with a secondary structure—in this case, the fibers of the sound-absorptive material. The heat exchange can lower the speed of sound to about 80% of its value in free air. A consequence of the reduced propagation speed is an apparent increase of box volume. If the volume is entirely filled by the material, the increase in apparent volume can be up to about 50%.

There are several approaches to closed box design. The choice of approach depends on the type of loudspeaker used and the desired behavior at the loudspeaker's resonance frequency. Since the mass moves against the compliance of both the air trapped in the box and the compliance of the loudspeaker suspension, the *mounted resonance* frequency is going to be higher than the *free resonance frequency* of the loudspeaker.

The traditional approach was to let the mounted resonance frequency only be minimally higher than the free resonance frequency. This is a reasonable choice for a loudspeaker that has a stiff suspension. The typical applications are loudspeakers mounted in quasi-baffle-type enclosures, that is, large enclosures. This approach requires loudspeaker drivers having strong magnets or mechanical resistance high enough to give reasonable damping. There is otherwise a risk that the voice coil may travel so far at low-frequency transients that it moves out of the air gap, which is likely to damage the voice coil.

A modern approach is to make the box so small that its compliance is much smaller than that of the loudspeaker. This *acoustic suspension* design results in a marked increase of the mounted resonance frequency, compared to the free resonance frequency. The addition of the compliance of the trapped air results in better distortion characteristics for loudspeaker drivers that have poor suspension linearity. In such a case, the air may be more linear than the mechanical diaphragm suspension. Acoustic suspension designs need loudspeakers having rigid cones and large diameter voice coils.

The closed box design always results in an increased resonance frequency, unless a suitable sound-absorptive material is used in the box as discussed previously. This, in turn, leads to an increase in the Q-value of the mechanical system leading to a more resonant character of the high pass properties of the transfer function of the loudspeaker system.

To have optimal properties, it is necessary to adjust the properties of the box to those of the loudspeaker driver. At high frequencies, where the inner dimensions of the box start to become close in size to one-half wavelength of the sound, the box will become resonant. This means that it will be necessary to apply sound-absorptive materials to the inner box surfaces. A small air leak is necessary to compensate for changing atmospheric pressure.

The *bass-reflex loudspeaker* uses an extra resonant circuit to decrease the lower operating frequency of the loudspeaker. Below this cutoff frequency, the loudspeaker will have a faster drop in frequency

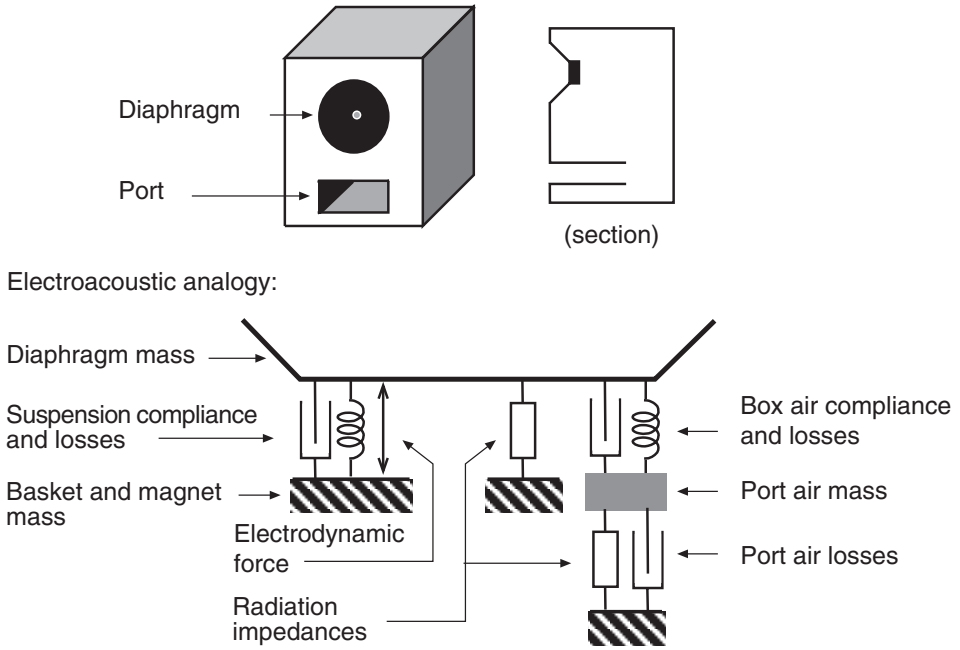


Figure 14.16 A ported box—bass-reflex—loudspeaker and the mechanical analogy of its acoustical system.

response than will the closed box loudspeaker. The mechanical system that is analogous to that of the acoustical system in the bass-reflex loudspeaker is shown in Figure 14.16. It is easy to design the system for various types of response if one knows all the necessary electrical, acoustical, and mechanical properties of the loudspeaker and the bass-reflex or closed box in question. The most common approach is to give the system a *Butterworth-type response*, in analogy to that of electrical filters. Such a response is sometimes also called a *maximally flat* response.

14.7 HORN LOUDSPEAKERS

By using a horn in front of the loudspeaker, it is possible to dramatically increase the efficiency and the sound power output of the loudspeaker. This is not due to confinement of the radiation but because the horn acts as an acoustic impedance transformer. A practical finite horn, however, introduces resonances and dispersion. Only an infinitely long horn can be resonance free. Bends and constrictions in the horn also cause resonances.

The horn is characterized by an expansion constant known as the *flare rate*. The flare rate determines the lowest frequency at which it is possible for sound waves to propagate through the horn. The propagation is often assumed to be as a plane wave, but this is a theoretical simplification. The waves inside the horn are seldom plane.

The resonant characteristics of the horn are due to sound being reflected at the mouth of the horn. Reasonably reflection-free conditions are obtained when the diameter of the mouth of the horn is larger than approximately 0.3 times the wavelength at the cutoff frequency.

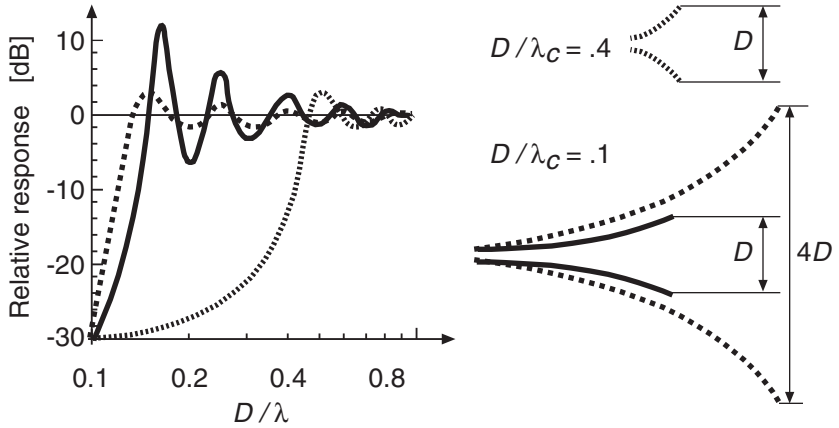


Figure 14.17 Typical frequency response of some exponential horn loudspeakers with different flare (expansion) rates, mouth diameters (D), and cutoff frequencies ($\lambda_c = c_0/f_{cutoff}$). (After Ref. 14.3)

Clearly, this limits the use of large exponential horns to places such as outdoor arenas, theaters, and cinemas, although designs for domestic use exist as well—for example, those shown on the cover of this book and in Figure 9.3. Short horns are common for midrange and treble loudspeakers since for these frequency ranges the horn can be made large compared to the wavelength in the working range. Figure 14.17 shows the response characteristics for some exponential horns having different flare rates and lengths.

14.8 MULTIPLE DIRECT RADIATOR LOUDSPEAKER SYSTEMS

The frequency response problems of electrodynamic loudspeakers have been outlined in Section 14.3. These problems result in a *high fidelity* loudspeaker driver often being usable only over a frequency range of 3 to 4 octaves. To circumvent some of these problems, one tends to subdivide the frequency range so that different loudspeaker drivers are responsible for different frequency ranges.

These designs are called two-way and three-way designs (or more if appropriate), according to how many loudspeaker drivers that are used in a single loudspeaker box to cover the desired frequency range of the loudspeaker. These designs require different types of electrical low-pass and high-pass filters, usually called *crossover filters*. The filters may be passive, using only resistors, inductors, and capacitors to divide the frequencies of signal from the amplifier to reach the appropriate loudspeaker driver. Figure 14.18 shows how *crossover networks* can operate on both low and high level signals. Crossover networks (and most other things concerning loudspeakers) are discussed in detail in Reference 14.7.

It is difficult to design loudspeakers using multiple drivers. Some of the difficulties concern the frequency and phase responses, the power handling, the directivity, and the balance between the frequency response in the main lobe direction and that in other directions.

There are now many types of computer software available designed to model the response of multiple driver direct radiator loudspeakers. Without such software, the development time for new loudspeaker designs would be much longer and slower. An example of the frequency response characteristics of a

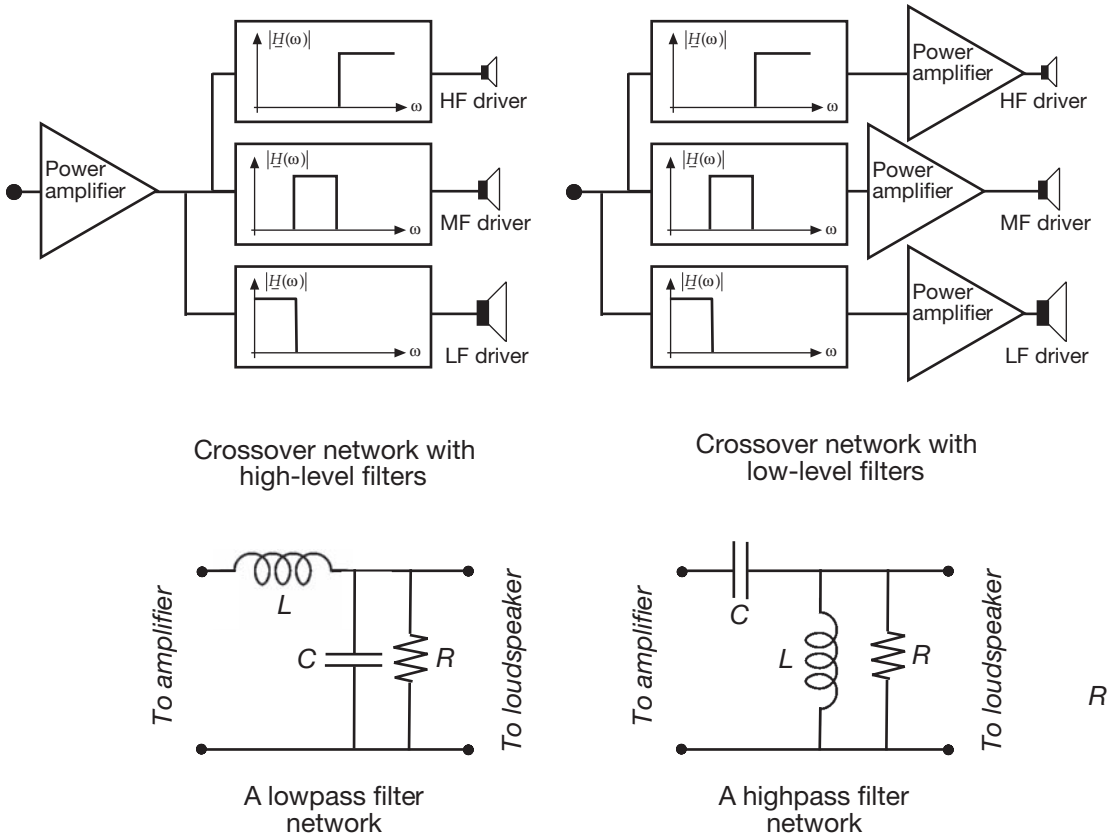


Figure 14.18 Two types of crossover networks. The lower part of the figure shows two simple electrical circuits similar to those used in practical crossover networks. For example, a bandpass network can be accomplished by connecting a lowpass and highpass filter in series.

commercial two-way system is shown in Figure 14.19. Obtaining flat frequency response in the crossover region may be difficult since the sound pressure and the sound power characteristics are different. The direct sound from the loudspeaker will usually not have a flat frequency response off the nominal axis of the loudspeaker system.

14.9 ARRAY LOUDSPEAKERS

Many sound reinforcement applications—such as churches, theaters, cinemas, and sports arenas—require loudspeaker systems that have a high- and frequency-independent directivity.

A high directivity requires that the sound radiator have large dimensions compared to wavelength. As previously noted, it is impractical to design loudspeaker diaphragms having diameters larger than wavelength. If high directivity is needed, it is necessary to replace the large diaphragm loudspeaker with an array of smaller diaphragms or a horn, as mentioned earlier.

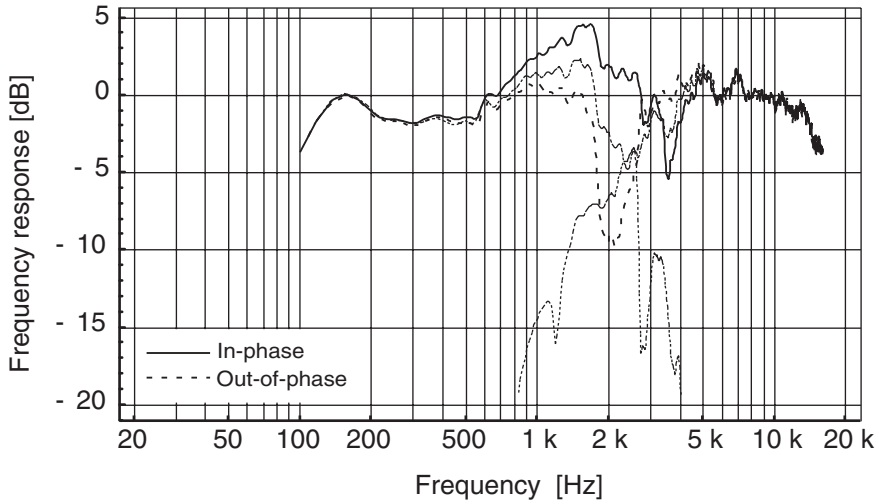


Figure 14.19 An example of the on-axis frequency responses of the individual drivers of a two-way Yamaha NS10 loudspeaker system (thin dotted lines) and the resulting responses for correct and incorrect (out-of-phase) interconnection (anechoic measurements).

Arrays of loudspeakers are often used in place of single large loudspeakers. A single one-dimensional *array loudspeaker* is usually called a *column loudspeaker*. Its directivity will depend on the directivity of the individual drivers, the distance between the drivers, the length of the array, and the way that the signals are distributed between the loudspeaker drivers. Since there are always unavoidable production tolerance differences between the drivers, the results are seldom as good as predicted by simple theory. Horn loudspeakers often have better directivity performance than arrays.

The basic directivity pattern of array loudspeakers can be designed approximately by adding the contributions of the individual driver elements at the receiving points. As an example, consider a linear array of four (small) drivers that has the following directivity function:

$$F(\theta) = \frac{\sin\left(2kb\sin(\theta)\right)}{\sin\left(\frac{kb}{2}\sin(\theta)\right)} \tag{14.20}$$

Some examples of directivity patterns for this array are shown in Figure 14.20. One notes that the directivity function is frequency-dependent. This problem can be solved for large arrays by *tapering* the response of the individual drivers.

It is also important to note that one-dimensional arrays only give high directivity in one plane. Sometimes two-dimensional arrays are necessary to control the directivity. Such two-dimensional arrays are occasionally used to confine the noisy area in discotheques and nightclubs.

Note that the far-field where the level will drop at −6 dB per distance doubling does not start until a considerable distance from a large array. At close distance to the array, the level will vary rapidly with location as discussed for large loudspeaker diaphragms earlier in this chapter.

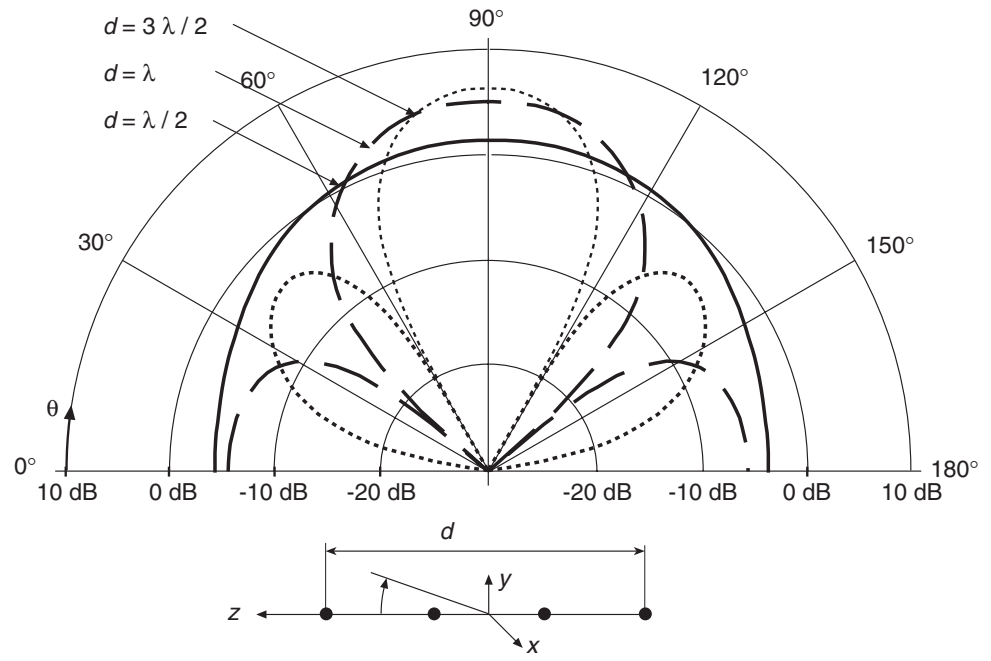


Figure 14.20 Directivity patterns of a simple, linear, four monopole array for three ratios of interelement distance *d* and wavelength λ. The curves show gain in dB over that of a free monopole having the same radiated power as the piston. Note that the array is along the *z*-axis shown in Figure 14.1 and that only one half of the toroidal pattern around the *z*-axis is shown.

Commercial loudspeaker systems for sound reinforcement often make use of several arrays to shape the directivity function over their working frequency range. Ribbon-type dynamic loudspeakers for domestic use have directivity characteristics similar to those of array loudspeakers.

14.10 ROOM EFFECTS

The listening room influences the response of the loudspeaker system in many ways. It is convenient to separate the low- and high-frequency effects. The low-frequency effects typically extend up to around 200 Hz in rooms of up to about 100 m^3 of volume. The high-frequency effects have already been discussed in Chapter 5. The discussion here will be confined to the low-frequency range, under the Schroeder frequency given by Equation 4.67.

The radiated power by a loudspeaker is influenced by the radiation impedance, which changes as the loudspeaker is moved close to sound reflecting surfaces. The effect will vary depending on such things as the properties of the loudspeakers, the position of the drivers, the surface reflection coefficients, and the number of (intersecting) reflecting surfaces.

The simplest case to handle is that of loudspeakers having a high internal impedance compared to the radiation impedance, which is often the case at low frequencies. For a constant volume velocity

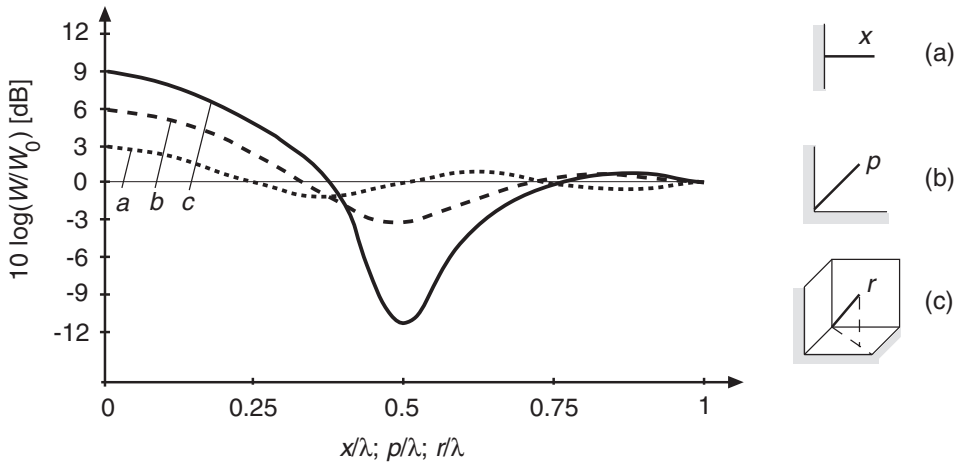


Figure 14.21 The sound power W radiated by a monopole close to rigid reflecting surfaces, relative to free field W_0, as a function of the ratio between distance and wavelength. Three cases shown: (a) plane, (b) on the diagonal between two orthogonal planes, and (c) on the space diagonal between three orthogonal planes. (After Ref. 14.6)

loudspeaker sound output, the power output of the loudspeaker will vary with frequency as shown in Figure 14.21. The sound pressure response will behave similarly.

For the loudspeaker to behave as if under free field conditions, the loudspeaker has to be at least a quarter to half a wavelength away from the sound reflecting surfaces. In practice, this often corresponds to a few meters. Note that this discussion concerns omnidirectional loudspeakers with constant volume velocity. Most loudspeakers are omnidirectional at the frequencies of interest in this discussion.

In a resonant sound field, such as that obtained in a rectangular room (discussed previously in Chapter 4), the position of the loudspeaker will influence the radiation impedance felt by the loudspeaker and, in this way, the radiated sound power. It is therefore important to test various positions for the loudspeaker, to find which position gives the best response in a particular room. This testing can be considerably simplified by using the reciprocity between source and receiver position indicated by Equation 4.38.

The principle of reciprocity can be used by putting the loudspeaker at the desired listening position and then moving an omnidirectional microphone between alternative loudspeaker positions to find the one that has the smoothest response. Figure 14.22 shows the result of such a pair of measurements in a rectangular room having a volume of approximately 25 m^3. The differences between the curves are small compared to the peaks and dips due to the modes of the room.

14.11 TRANSIENT RESPONSE

Human hearing functions as a complex time and frequency analyzer. At high frequencies, the time constant of hearing is only a few milliseconds.

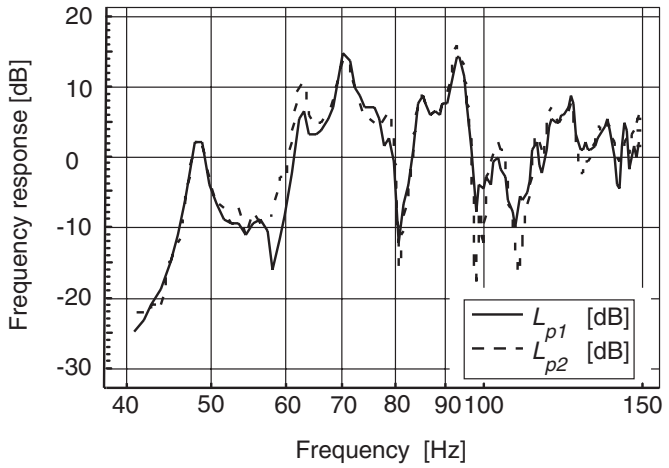


Figure 14.22 Two frequency response curves for a pair of points in a rectangular listening room having a volume of 90 m^3 obtained by switching microphone and loudspeaker positions. Both microphone and loudspeaker were small and omnidirectional in the frequency range shown. The effects of reciprocity are clearly visible, up to a frequency of 150 Hz.

The transient response of a loudspeaker is determined by the electrical circuit, the mechanical properties of the diaphragm and suspension, and by the geometrical extension of the loudspeaker. It can be very complex, showing considerable ringing, particularly at the loudspeaker resonance frequencies and in the crossover frequency regions in multiple driver direct radiating loudspeaker systems.

An additional complication is that a seemingly almost flat frequency response can be the result of many closely spaced resonances.

14.12 NONLINEAR DISTORTION

The first noticeable result of the presence of nonlinear distortion in loudspeaker systems is the generation of frequency components in the reproduced sound that were not present in the electrical signal supplied to the loudspeaker driver. The nonlinear distortion can come from several sources.

Tonal distortion components are created by the nonlinearities in the magnetic field and in the diaphragm's mechanical system, particularly the suspension. Tonal distortion is also created by the nonlinearity of the trapped air in the loudspeaker box and when sound propagates long distances at high intensity in horn-type loudspeakers. Distortion, such as turbulence noise, is created when air moves at high speed around the voice coil and in acoustic resonator components such as the port opening in bass-reflex loudspeaker boxes, as discussed in Chapter 9. Another source of nonlinear distortion is the buckling that can occur when driving the loudspeaker diaphragm with high force. An example of finite element modeling of such buckling is shown in Figure 14.23.

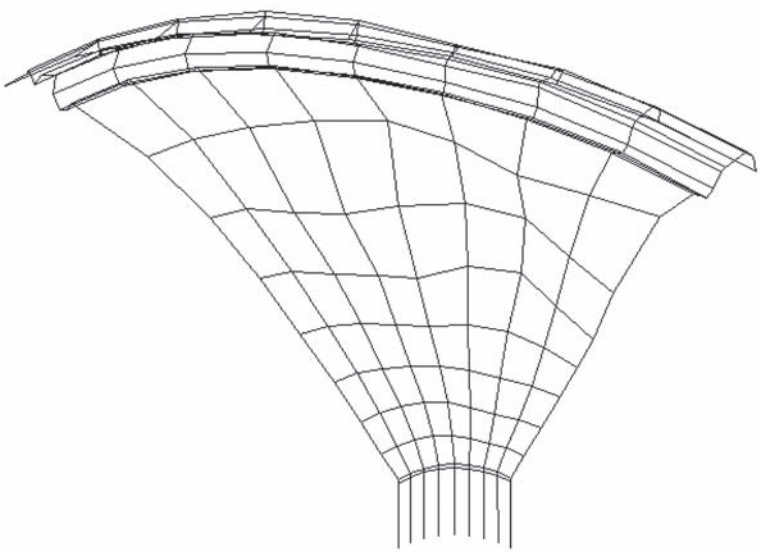


Figure 14.23 Buckling of a loudspeaker diaphragm calculated using finite element modeling.

Some of these types of nonlinear distortion can be minimized by using balanced designs, for example, in the magnetic field and in the diaphragm suspension. Nonlinear distortion can also be reduced by replacing mechanical components by simulation using various types of electronic feedback circuits. Other types of distortions, such as the distortion due to air nonlinearity and turbulence, can only be minimized by reducing sound pressure and particle velocity. A particular type of tonal distortion is the Doppler distortion that is the result of movement by the diaphragm at several frequencies at the same time, particularly high displacement movement at low frequencies.

Reduction of nonlinear distortion can also be done using both analog and digital signal processing if an accurate model of the distortion generation mechanism is available or if the nonlinearity can be sensed in some way.

14.13 ELECTRONIC COMPENSATION OF NONLINEARITIES

Since loudspeaker design is still mostly focused on loudspeakers to be used with amplifiers having linear frequency response, one must correct for the frequency and directivity responses in electrical, mechanical, or acoustical ways. This can only be done in an incomplete way by passive methods, such as those described in this chapter.

To achieve good loudspeaker system properties, one must regard the loudspeaker as a part of a complete sound reproducing system that includes amplification and signal processing of various kinds. This systems approach gives much larger design freedom. The signal processing can include compensation for both linear and nonlinear distortion, voice coil power, and diaphragm displacement overload protection, and, in the case of array loudspeakers, frequency-dependent adjustment of the directivity characteristics.

14.14 ELECTROSTATIC LOUDSPEAKERS

The *electrostatic loudspeaker* uses the force of a dynamic electric field on electrical charge to move a diaphragm. The charge is usually introduced by using a bias voltage supply e_{bias}, as shown in Figure 14.24. The figure shows the basic operating principle of the push-pull balanced drive electrostatic loudspeaker (and headphone). The fixed electrodes are usually made of plastic that has been provided an electrically conductive layer. The electrodes must be well insulated, usually by applying a special paint or insulator.

The diaphragm is typically made of a thin plastic foil, generally only 10 μm thick. The diaphragm should have low conductivity, sufficient to be charged over a day but must not lose its charge in the case of arcing—local ionization of the air inside the loudspeaker.

Modern electrostatic loudspeaker systems operate using a diaphragm having high surface resistivity, so that each part of the diaphragm can be considered to hold a certain permanent charge. The charged foil is placed between the electrodes, so that it is subject to an electrical field $\underline{E}_{audio}$. The field strength between two planes, closely placed, large electrodes depends on the audio voltage applied to the electrodes $\underline{e}_{audio}$ and on the distance between the electrodes h.

$$\underline{E}_{audio} = \frac{\underline{e}_{audio}}{h} \tag{14.21}$$

The electrodes need to be perforated so that sound can escape the loudspeaker. This will cause the field to be somewhat uneven because the foil has some mass and stiffness. The electrostatic force will tend to average over the surface of the foil. The force per unit area $\underline{F}''$ is ideally:

$$\underline{F}'' = q''\underline{E}_{audio} \tag{14.22}$$

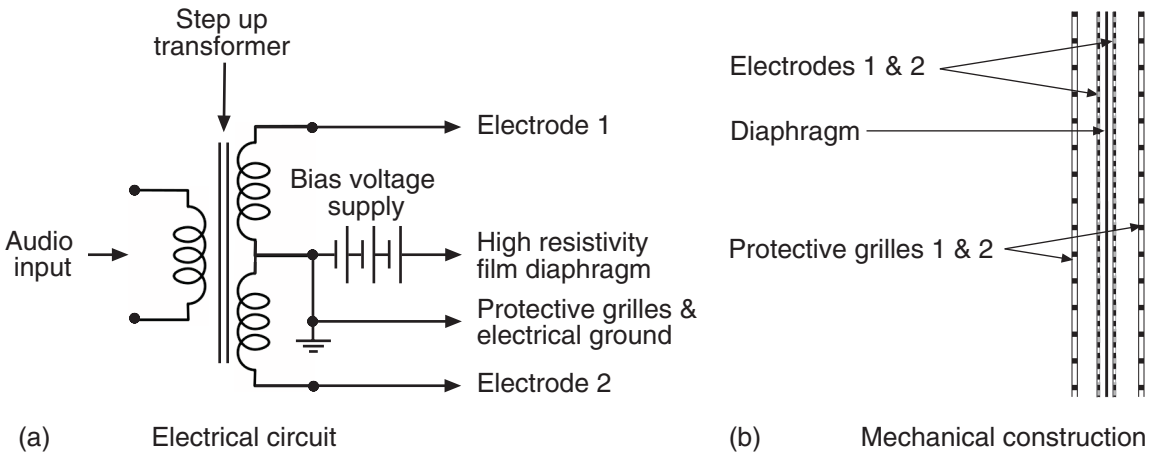


Figure 14.24 The electrical circuit for a *push-pull* balanced drive electrostatic loudspeaker. The electrical input impedance of the electrostatic loudspeaker is primarily capacitive making it a difficult load for power amplifiers (a). The electrostatic loudspeaker uses the force of an electric field on electric charges on a thin polymer film diaphragm to move the diaphragm. The electrodes are fixed and have little acoustic resistance but need to be isolated to withstand the maximum voltages between the electrodes, ground, and film. The film must have high electrical resistivity so that charges remain evenly distributed over the film as it moves and if there is arcing to an electrode (b).

where q'' is the charge per unit area on the foil. The charge is ideally constant as a function of time on each patch of the foil since the foil has such high surface resistivity. The charge is determined by the area of the foil S_D, the voltage to which the foil is charged e_{bias}, and the capacitance C_E between each side of the foil and the corresponding electrode, the same as with the condenser microphone (see Equations 12.11 and 12.12).

$$q'' S_D = e_{\text{bias}} C_E \tag{14.23}$$

In contrast to magnetic fields, electrostatic fields may ionize the air if the field strength is large. On ionization the air forms a conductive path between electrodes and foil, this is called arcing. Arcing can burn holes in the foil and in the electrode insulation layer causing the loudspeaker to cease operating. Typically air will become ionized if the field strength exceeds 3 to 6 kV/mm, depending on air humidity. The field strength inside the loudspeaker depends both on the bias voltage applied to the foil and on the audio voltage applied to the electrodes. In practice, the applied audio voltage is limited to a few kV. Once the air is ionized, it will burn a hole in the plastic foil and carbonize a part of the foil. This will reduce the acoustic output of the loudspeaker. High resistivity foils are less susceptible to this type of damage than foils that have been metalized and have high conductivity.

Because the time constant of the charge circuit cannot be infinitely large, the charge will start to flow to and from the bias supply for very low frequencies. The position of the foil becomes unstable, and the foil may even be attracted to, and stuck to, an electrode. For this not to happen, the foil needs to be mechanically tensioned so that most of the displacement available can take place without the foil going into one of the two unstable regions close to the electrodes.

The foil tension, together with the mass and Young's modulus of the foil, determine the lowest resonance frequency of the electrostatic loudspeaker. The frequency should be at the low end of the spectrum, since the frequency response of the loudspeaker is essentially frequency-independent above this resonance frequency and up to the Helmholtz resonance frequency determined by the compliance of the air between the foil and the electrode, as well as by the air mass in the holes of the electrodes.

Electrostatic loudspeakers are used as dipoles without boxes but could be mounted in a baffle. The reason is that the loudspeaker diaphragm area needs to be large to generate sufficient volume velocity to radiate much sound power and achieve high sound pressure at the listening position. The diaphragm movement is controlled by the radiation resistance. Because the diaphragm has such low mass per unit area, this leads to most diaphragm vibration modes, except the lowest frequency ones, being extremely well damped.

The frequency response of an experimental full frequency range electrostatic loudspeaker strip under anechoic conditions is shown in Figure 14.25. The electrostatic loudspeaker strip (0.15×1.8 m^2) was mounted in a rigid baffle and the measurement point was at 1.5 m distance from the strip, on-axis, and symmetrically positioned. Low-frequency diaphragm resonances are clearly visible as is the low-frequency sound pressure cancellation by the back side radiation and also the frequency response peak at about 18 kHz of the Helmholtz resonance of the air between the diaphragm and the electrodes and the holes in the electrodes.

Since the sound radiating surface is typically large, even small diaphragm movements lead to satisfactory sound pressure in home settings. The large diaphragm results in high directivity that reduces the influence of the room on the reproduced sound. The small diaphragm movement and the balanced drive push-pull operation results in electrostatic loudspeakers being characterized by low nonlinear distortion.

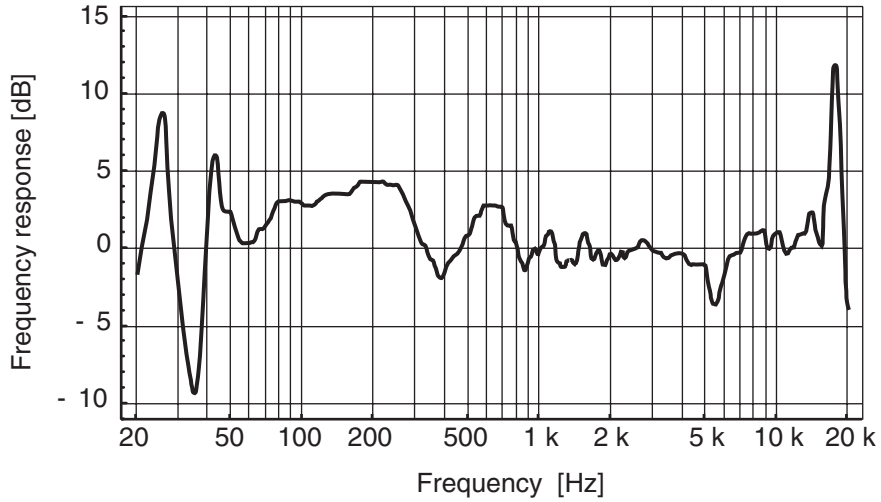


Figure 14.25 Anechoic measurement of the on-axis frequency response at 1.5 m distance of an electrostatic loudspeaker strip (0.15 m by 1.8 m) mounted in a rigid baffle 1.2 m by 2.2 m. Low-frequency membrane resonances are clearly visible, as is the sound pressure cancellation by the back side radiation (at about 300 Hz), and also the frequency response peak of the Helmholtz resonance of the air between the diaphragm and the electrodes and the holes in the electrodes (at about 18 kHz).

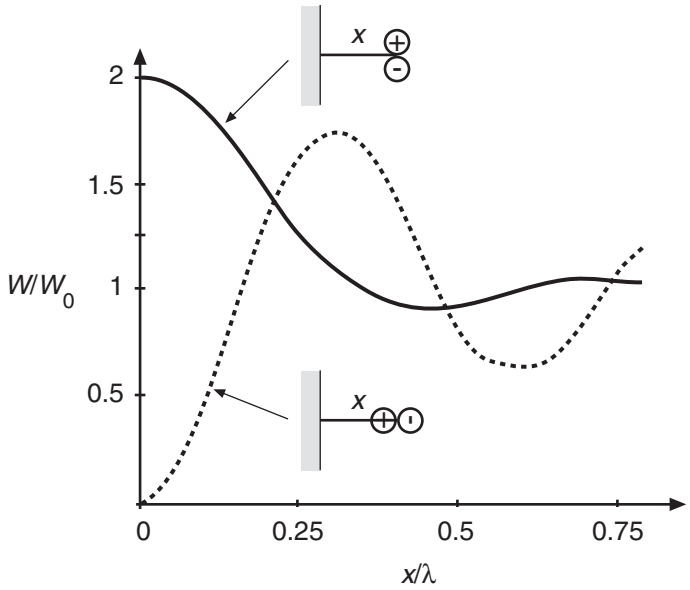


Figure 14.26 The power radiated by a dipole close to a rigid plane depends on its direction, relative to the plane. This graph shows the power W radiated by a dipole for two different directional alignments of the dipole, relative to the free field power radiated W_0. The center of the dipole is at a distance x from the reflecting plane. (After Ref. 14.6)

An interesting psychoacoustic effect in using full frequency range electrostatic vertical strip loudspeakers is that the sound in listening appears to come from the part of the loudspeaker at the height of one's head.

In contrast to loudspeaker box designs, electrostatic and other open baffle mounted loudspeakers are best placed away from the walls of the listening room. Because the dipole is effectively two different sources, the alignment of these sources relative to the reflecting surface is important, as indicated by the power radiation curves for dipoles shown in Figure 14.26. For good low-frequency sound power generation, the dipole operating open baffle loudspeakers should be positioned so that their dipole action is parallel relative to the close reflecting wall.

14.15 PROBLEMS

14.1 An electrodynamic transducer has the following electrical and mechanical data:

Diaphragm mass	$M = 10^{-2}$ kg
Suspension stiffness	$k = 2{\cdot}10^{3}$ N/m
Suspension loss resistance	$R_M = 2$ kg/s
Air gap flux density	$B = 1$ N/A
Voice coil winding length	$l = 7.5$ m
Voice coil electrical resistance	$R_E = 10\ \Omega$
Diaphragm diameter	$d = 0.1$ m

Tasks:

a. Calculate the *free* resonance frequency f_0.
b. Calculate the rms value of the cone displacement and velocity as a function of frequency.
c. Draw a curve showing the diaphragm displacement and velocity as a function of frequency for a rms drive current of 2A.

Use a logarithmic frequency axis and express diaphragm displacement and velocity as levels, using 1 mm and 1 m/s, respectively, as reference levels.

14.2 Assume that the electrodynamic transducer given in Problem 14.1 is mounted in a *closed box* arrangement. The box volume (after removing the volume occupied by the loudspeaker) is $2 \cdot 10^{-2}$ m^3.

Task: Calculate the resulting resonance frequency f_{mount} for the loudspeaker when mounted in the box.

14.3 The acoustic power radiated by the electrodynamic transducer vibrating with the velocity $\tilde{u}$ and box described in Problems 14.1 and 14.2 can be written as:

$$W \approx sSZ_0\tilde{u}^2$$

Here, S is the radiating area, and s is the radiation ratio.

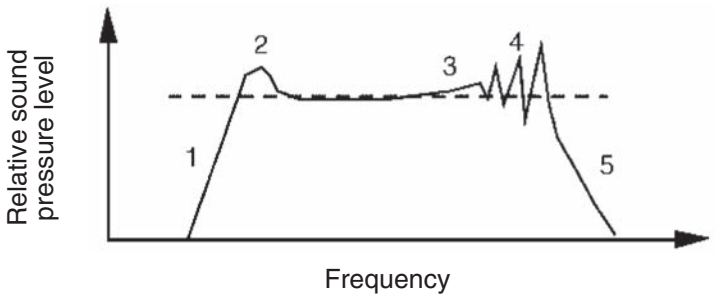
When the dimensions of the radiating surface are much smaller than the wavelength of sound, the radiation ratio can be approximated as:
Here, a is the loudspeaker diaphragm radius

$$s \approx \left(\frac{\omega a}{c_0}\right)^2$$

Tasks:

a. Calculate the radiated sound power at the following frequencies: $1/4\, f_{\text{mount}}$, f_{mount}, and $4\, f_{\text{mount}}$ for a rms electrical drive current of 2A.
b. Calculate the resulting sound power level as a function of frequency using a logarithmic frequency axis.

14.4 An electrodynamic transducer mounted in a *closed box* was measured in free field. The frequency response given in the graph was measured *on axis*.



Tasks:

a. Explain the reasons for the deviations 1 to 5 in the curve, relative to *flat* frequency response.
b. Explain how the frequency response problems may be changed/altered/removed by modifying the construction of the loudspeaker drive unit and box.

14.5 An electrodynamic transducer, having a free resonance frequency of 50 Hz, is mounted against the end of a tube closed at the opposite end. The length l of the tube is 2 m, and its cross-sectional area S is $3 \cdot 10^{-2}$ m^2. The mass of the loudspeaker's diaphragm M_M is $1 \cdot 10^{-2}$ kg.

Task: Determine the resulting resonance frequency using a graphic or numerical solution if the loudspeaker's effective cone area is equal to the cross-sectional area of the tube.

14.6 One can use a loudspeaker having an electrodynamic transducer as a resonance absorber. Determine an expression for the resonance frequency expressed as a function of a loudspeaker driver's diaphragm area, diaphragm mass, suspension stiffness, suspension losses, and effective box volume. Also, derive an expression for the absorption of the loudspeaker.

14.7 An electrodynamic transducer having the following specifications is mounted in a closed box arrangement. The maximum displacement of the diaphragm results in a volume change of $3 \cdot 10^{-4}$ m^3. The sound power output is also limited by the maximum power dissipation allowed in the loudspeaker drive unit's voice coil.

The drive unit's specifications are:

Resonance frequency (fitted)	50 Hz
Effective diaphragm area (circular)	$8 \cdot 10^{-2}$ m^2
Efficiency ($W_{\text{acoustical,out}}/W_{\text{electrical,in}}$)	3%
Maximum voice coil power dissipation	20 W

Assume that the loudspeaker radiates as if the diaphragm were an ideal sphere having a radius equal to one-half of the diaphragm's radius.

Task: Calculate the maximum sound power available from the loudspeaker as a function of frequency for frequencies from 0.5 times the resonance frequency up to the frequency corresponding to *ka* equal to 0.5 where *k* is the wave number and *a* is the radius of the cone.

14.8 We wish to construct an intercom using an electrodynamic transducer as both loudspeaker and microphone (see Problem 12.6). The electrodynamic transducer is mounted in a *closed box*, having an effective volume of $5 \cdot 10^{-4}$ m^3.

The transducer's specifications are:

Resonance frequency (free)	250 Hz
Effective diaphragm diameter	$3 \cdot 10^{-2}$ m
Diaphragm mass	$1 \cdot 10^{-3}$ kg
Bl-product	1 N/A
Voice coil resistance	5 Ω

Task: Calculate the sound pressure level, L_p, at a distance of 1 m from the transducer at a frequency of 1 kHz for a voice coil driving voltage of 1 V.

Hint: Assume that the diameter of the transducer is much smaller than the wavelength of sound and that the fitted transducer radiates as a monopole having a volume velocity of $\underline{U}_{\text{diaphragm}} = \underline{u}_{\text{diaphragm}} S_{\text{diaphragm}}$.

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Headphones and Earphones 15

15.1 INTRODUCTION

Headphones are small personal loudspeakers that radiate sound close to the ear canal opening. The distance between the ear canal opening and the headphone driver is about 5 cm or less. *Earphones* are smaller devices designed to be inserted into the ear canal entrance, typically forming a sealed cavity.

Headphones and earphones enjoy wide usage, extending into areas far beyond general high fidelity audio sound reproduction. Many forms of military and civilian telecommunications rely on the use of headphones, for example, telephones, protective gear, or diver and pilot helmets.

Since headphones and earphones are used in mobile applications and are often handled carelessly, they are subject to stringent mechanical requirements. They must also fulfill demanding hygienic requirements, since they are close to the body, become dirty, and are exposed to comparatively high temperatures and humidities.

Many earphones are used in hearing aids. A hearing aid contains the microphone, filters, (digital) signal processor and compressor, amplifier, and headphone in one single small unit, possibly insertable into the ear canal and nearly filling the concha. Sometimes it is advantageous to use bone conduction transducers when the middle ear is damaged or when the ear canal entrance cannot be closed.

The most common headphone and earphone designs for high-quality sound reproduction use are electrodynamic, electromagnetic, and piezoelectric drivers.

15.2 HEADPHONES/EARPHONES VS. LOUDSPEAKERS

The audio requirements for headphones are different from those of loudspeakers. Headphone and earphone transducers need only generate small volume velocities, thus, they may operate in a more linear range and have less nonlinear distortion. Since the listener's room is eliminated from the audio chain, the response characteristics of the room are eliminated.

Do headphones and earphones need *flat* frequency response? When we listen to sound reproduced over loudspeakers in rooms, the aural impression is formed by the total response of the direct sound, early reflected, and reverberant sound. Since most living rooms have short reverberation times, direct and early reflected sound will be dominant. The sound pressures at the listener's ears will depend on such things as the loudspeakers, their direct sound frequency response, the response in other directions,

the reflection characteristics of the room surfaces, and scattered sound from objects. But the sound pressure will also depend on the reflection of sound by the listener's body and head.

We have already discussed the concept of the head-related transfer function (HRTF) in Chapter 6. Figure 15.1 shows an example of the frequency response of the HRTF for frontal sound incidence, as well as for diffuse field incidence, for one ear on a manikin head. Clearly, the frequency responses for both cases are far from linear.

Examples of frequency response curves, measured using a simple coupler, for some high-quality headphones are shown in Figure 15.2. One notices that even high-quality headphones have non-flat frequency responses. The response curves for the semi-open supra-aural headphone and the insert earphone show relatively resonance free behavior. The response curve of the circumaural phone shows the influence of the modes inside the cavity formed by the headphone. Note, however, that our hearing is used to these types of resonances, which, in any case, are associated with the external ear.

The peak/rms relationships in sound pressure between headphone, earphone and loudspeaker listening are also going to be different. Significant musical peak levels can have a duration as short as a few milliseconds, particularly with popular amplified instruments, such as electric guitars. Looking at the dashed curve in Figure 3.18, which shows the reported level difference between sine-wave tone bursts and continuous sine waves at 1 kHz, we note that for a 5 ms burst, the perceived level *deficiency* is approximately 20 dB. When listening to headphone sound, which, of course, has not been subject to the room impulse response, the peak/rms ratios are thus likely to be 10 to 20 dB higher than in loudspeaker listening.

Since it is likely that listeners adjust the average reproduced sound level to about the same loudness for both types of listening, it is likely that people will listen to peaks that are relatively much stronger than while listening to loudspeaker reproduced sound. This will increase the risk of hearing damage.

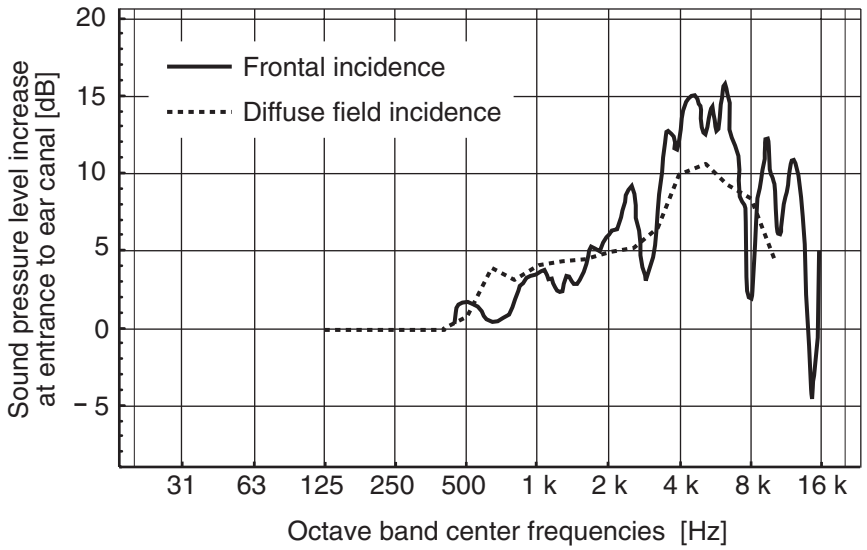


Figure 15.1 The log magnitude (frequency response) of the HRTF for an ear on a manikin head measured in an anechoic chamber for frontal sound field incidence and in a reverberation chamber for diffuse field incidence. No torso was used with the head for this measurement—if used, it would have increased levels in the 250 Hz range.

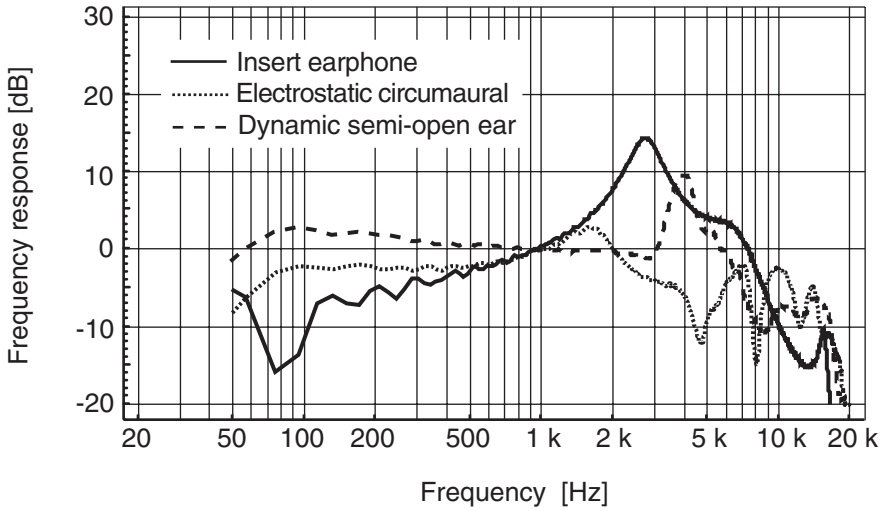


Figure 15.2 The frequency response of some high quality head- and earphones, measured using a simple small volume coupler.

Note that this also is a serious problem in mobile phone use, where the phone often is also used for music reproduction over headphone or earphones. Additionally, people generally adjust monaural sound reproduction to be louder than binaural sound reproduction.

By including a convolver in the amplifier, headphone or earphone electronics, having impulses responses of (or representative of) a real or fictive room, the convolution of the recorded audio and the room impulse responses will eliminate this risk.

15.3 THE ACOUSTIC ENVIRONMENT

The *acoustic environment* in which the headphone operates is very different from that of general loudspeakers. Some are made to operate directly coupled to the ear while others are designed to be used at some small distance from the head so that there is an air path or leak between headphone and head. The use of a correctly designed coupler is essential for the measurement of the response of headphones and earphones. Many couplers, however, are designed for telephony and will not adequately model the acoustic properties of the ear to be useful for full frequency range audio. Such a coupler typically has a volume of about a few cubic centimeters. The anthropometric head of the manikin shown in Figure 17.6a, however, has soft rubber pinnae, as well as a concha, ear canal, and eardrum simulator. Its design is shown to the left in Figure 15.3. On the right is a simple coupler that still allows the concha and ear canal modes to be similar to those in actual head (see Reference 15.4).

The direct coupling headphones can use an ear-surrounding tight cushion (in the case of *circumaural headphones*) or an *ear bud* inserted into the ear canal with little leakage (insert earphones).

Those headphones that are used away from the ear are designed so that they are primarily affected by the acoustic impedance of the opening between the headphone and the head, but not by the ear and the ear canal. These are often called *supra-aural headphones.*

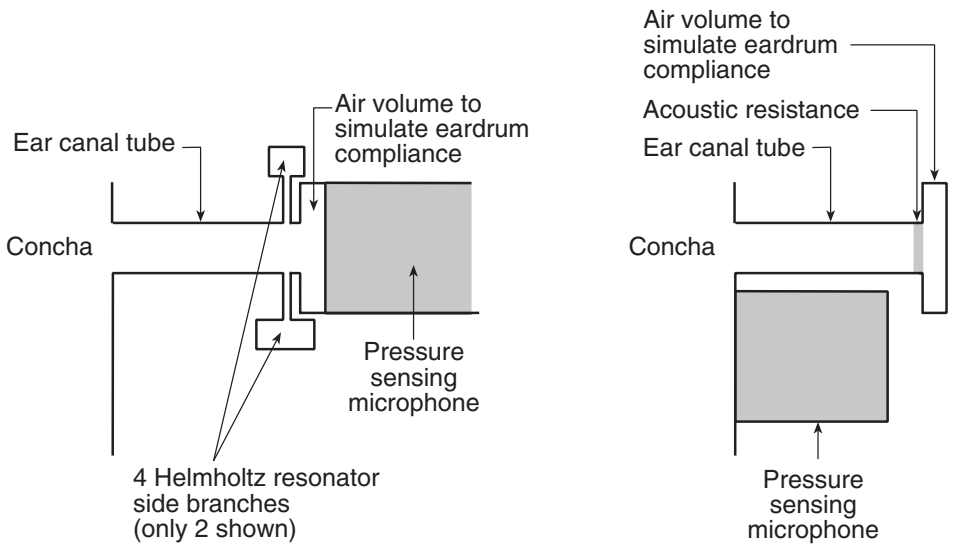


Figure 15.3 Two types of couplers to be used in headphone/earphone measurement and calibration as well as binaural sound recording. To the left is shown the Zwislocki coupler design. On the right is shown a simple coupler. Both allow the concha and ear canal modes to be relatively similar to those in a real head. The latter type is easier to equalize for use in binaural sound recording.

Usually headphones are much smaller than the wavelengths of sound over much of the audio frequency range. This means that the headphone (depending on construction) will act as a nearly ideal monopole or dipole sound source. The radiation impedance at frequencies up to 0.5 kHz can be described by the expressions given in Chapter 1. Since the headphone operates close to the head, the sound reflection by the head will considerably influence the impedance at higher frequencies. Any headphone design will need to be optimized regarding the operating distance between head and headphone. Figure 15.4 shows the large influence of leakage on the resulting frequency response of a headphone.

One must also note that the contact pressure will influence the compression, and, thus, the flow resistance, of any foam plastic pads used to seal the leak between head and headphone. Such variations will also influence the resulting frequency response, as shown by Figure 15.5.

Just as in the design of loudspeaker systems, it is necessary to have the intended use of the transducer clearly focused in its design. Since headphones are generally used to simulate the experience obtained when listening to loudspeakers, it is important to equalize the frequency response of the headphones so that it is similar to that of loudspeakers, at the ear canal entrance. Simulation of the diffuse and direct field loudspeaker frequency responses will be different.

15.4 ELECTROMAGNETIC HEADPHONES

There are several different electroacoustical operating principles used in headphone design. Electromagnetic and piezoelectric headphones are used in telephones and hearing aids. Electrodynamic and electrostatic headphones are used for high-quality audio sound reproduction.

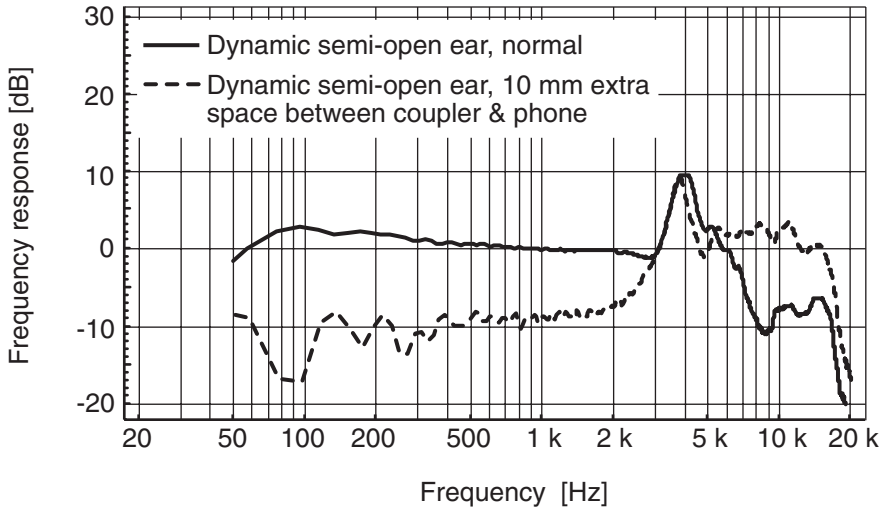


Figure 15.4 The leakage between the headphone and the body/ear influences the frequency response drastically. The solid line shows the frequency response of a particular headphone with little leakage and the dashed line shows the frequency response with large leakage.

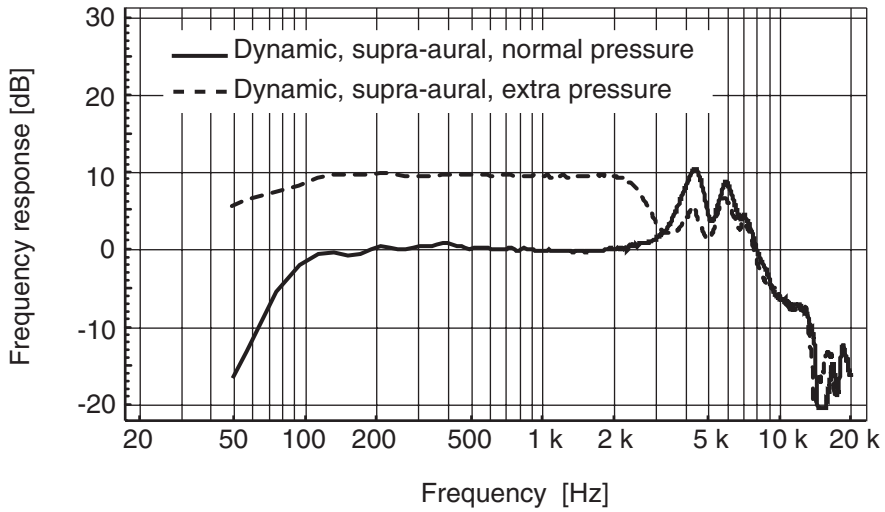


Figure 15.5 An example of the influence of leakage due to different contact pressures on the frequency response of a supra-aural headphone using a foam plastic seal.

The *electromagnetic headphone* (one design shown in Figure 15.6) has high sensitivity, a simple and rugged construction making use of a permanent magnet, a small air gap, a magnetically conductive diaphragm, and a coil of electrically conductive wire to enable modulation of the magnetic field. When audio voltage is applied to the coil, the magnetic field strength will vary, and so will the attraction

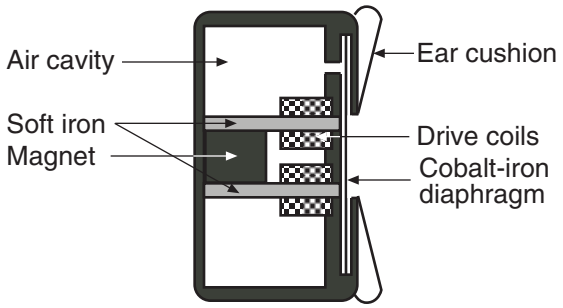


Figure 15.6 A simple electromagnetic headphone.

between the magnet parts. Since the diaphragm is mechanically compliant, it will move and change the air gap width.

Because the diaphragm resonances have high *Q*-values, the frequency response will be jagged for frequencies at and above the first diaphragm resonance. If the headphone has a tight seal against the ear, the acoustic impedance will be primarily capacitive at frequencies up to approximately 2 kHz. This means that the diaphragm stiffness and the stiffness of the trapped air will act as two series-coupled springs. This requires the diaphragm stiffness to be low, to generate sufficient sound levels, which, in turn, leads to a requirement for low diaphragm mass to have a reasonably high first diaphragm resonance frequency.

At low frequencies, the response will be determined primarily by the acoustic leakage around the seal of the headphone against the ear. If the seal is not tight, its impedance will dominate that of the trapped air at low frequencies. This results in a poor, low frequency response.

At higher frequencies, resonances in the trapped air, various acoustic circuits, and such will affect the frequency response. Above the first diaphragm resonance frequency, the frequency response of the electromagnetic headphone will drop off, as shown by the curves in Figure 15.7. Due to the asymmetry of the force-extension relationship of the air gap and the rest of the magnetic circuit, the electromagnetic headphone is inherently nonlinear, although symmetric, balanced designs are possible.

Lightweight and occluding earphones are important for listening to audio while exercising and traveling. The low mass is necessary so that the earphone does not move around in the ear canal entrance, causing it to fall out or create noise. The occlusion is desirable to have good low-frequency response and to keep environmental and transportation noise out so that the best signal-to-noise ratio is obtained. An advantage of having low background noise is, of course, that it removes the necessity for listening at sound levels that may lead to hearing damage.

Different technologies are in use for insert earphones. One technology that lends itself to miniaturization, and that has been in use a long time in hearing aids, is the balanced armature design shown in Figure 15.8.

The frequency response of such an earphone in a small coupler is shown in Figure 15.2. Since an insert earphone only needs to produce sound inside the ear canal, a cavity that is small, minimal volume velocities need to be generated by the diaphragm for high sound pressure at the eardrum. The small volume also results in a high acoustic impedance seen by the diaphragm, so the diaphragm suspension

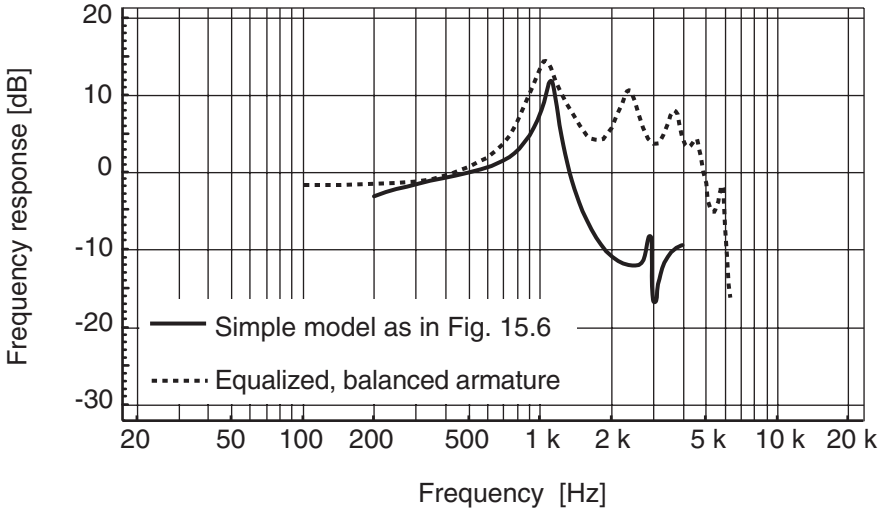


Figure 15.7 Frequency response curves for some electromagnetic headphones. Data from Knowles BK (balanced armature) series. Unknown couplers.

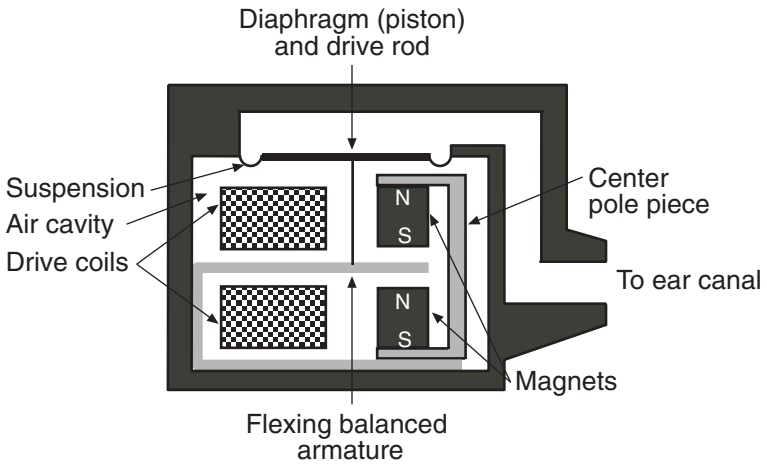


Figure 15.8 A simple electromagnetic bridge-type insert headphone using a flexible balanced armature that drives a small piston. Center pole piece and armature are magnetically conductive.

may be quite stiff. As with other electromagnetic transducers, the response is essentially stiffness controlled, leading to a drop-off of −12 dB/octave for frequencies above the resonance. This, in turn, results in the need for several transducers to cover the entire audio range with high sensitivity. The frequency response curve for the insert earphone shown in Figure 15.7, clearly shows the resonances and the associated high frequency drop.

15.5 ELECTRODYNAMIC HEADPHONES

Electrodynamic headphones are nearly as rugged as the electromagnetic designs. They are also suitable for those applications where the transducer must also be used as a microphone.

The relatively symmetrical design results in low nonlinear distortion. This means that the headphone can also be gainfully used as a high-fidelity audio sound reproducer. It is also used in other applications where there is a need for high-quality sound reproduction at high sound pressure levels, for example, in industrial and military applications.

Electrodynamic headphones are essentially built like miniature electrodynamic loudspeakers. Two types are common: (1) those that have an open back (dipole action) and (2) those that have a closed back (monopole action).

A basic closed box headphone design is shown in Figure 15.9. The internally trapped air will act in series with the air trapped between the diaphragm and the ear. Using a correctly chosen resonance frequency and adequate damping, the frequency response can be made quite smooth. For even better frequency response, one can use various types of resonant acoustical compensating circuits. This typically results in an extension of the effective frequency range by one or two octaves.

Any headphone having an *open back* will suffer from aerodynamic short circuit—that is, dipole cancellation. This leads to poor low frequency response and sensitivity unless the headphone is worn close to the ear. The response can be improved by a careful choice of diaphragm resonance frequency and the use of a suitable plastic foam pad between the headphone and the ear. The pad will introduce an acoustic resistance into the acoustic short circuit and minimize its effects.

One can also design electrodynamic headphones that use a resonance circuit similar to that of ported box loudspeaker systems as shown in Figure 15.10. If correctly designed, such headphones do not require a seal against the ear. Many types of mobile telephones use this approach to headphone design.

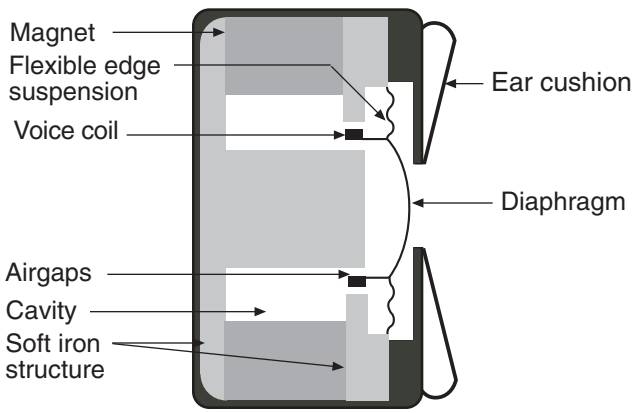


Figure 15.9 A simple electrodynamic headphone

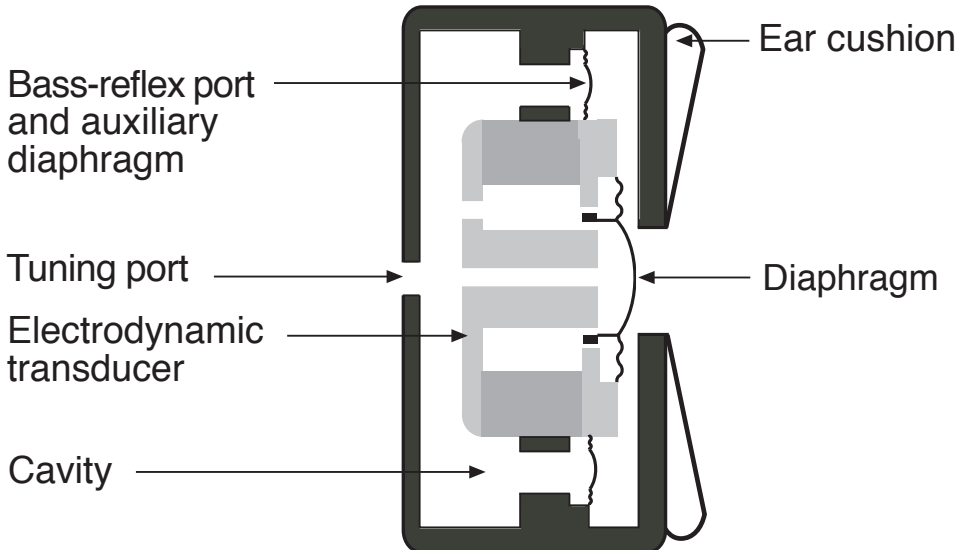


Figure 15.10 An example of a modern electrodynamic headphone using a bass reflex system ("ported box design").

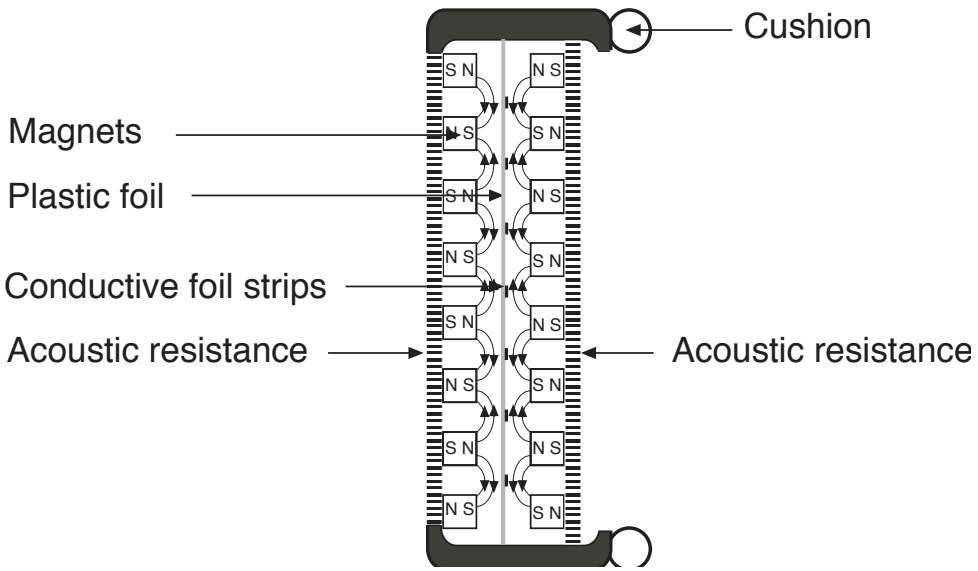


Figure 15.11 An isodynamic headphone using a balanced magnetic field design.

A modern version of the electrodynamic headphone is the *isodynamic* headphone, a dipole design, shown in Figure 15.11. Its design is reminiscent of that of electrostatic headphones, in that the entire surface area of the diaphragm is driven with about the same force per unit area. This is approximately achieved by applying a meander of an electrically conductive foil path onto a thin plastic film that is

placed in an appropriate magnetic field. The magnetic field is obtained using essentially a multitude of small magnets, polarized so that the film will move in the same direction over its entire area when electric current is fed through the conductor. The magnets can be made using a plastic mixed with a ferrite powder.

The advantage of this design is that one does not need to worry about diaphragm resonances other than the fundamental resonance, such as *rocking mode* resonances. Since the film has low mass, any high frequency resonances will be damped by the radiation impedance. By using a magnetic material with high flux, one can design headphones combining high sensitivity, low nonlinear distortion, as well as a wide and even frequency response.

15.6 PIEZOELECTRIC HEADPHONES

Piezoelectric headphones are based on achieving diaphragm motion using piezoelectric materials. Such materials are subject to shape changes when an electric field is applied to the material. While the piezoelectric effect can be obtained from naturally piezoelectric crystals, such as quartz and some salts, piezoelectric headphones typically are based on the use of piezoelectric ceramics and plastics (see Chapter 13).

Piezoelectric headphones can be made small, robust, and sensitive. The design is often used in hearing aids and in inexpensive portable devices. The main drawback of the design is the high electric impedance, which is mainly capacitive.

Since the mechanical system is mainly stiffness-controlled, it is important that there be a tight seal between the headphone and the ear. This is usually the case in the applications mentioned. If the fundamental resonant frequency of the headphone diaphragm is high enough, the effective frequency response will be determined by the resonances of the ear canal. Figure 15.12 shows an example of a piezoelectric headphone design using a conical diaphragm driven by a piezoelectric ceramic bender. The ceramic bender is assembled from two pieces of piezoelectric material (length mode operation) that

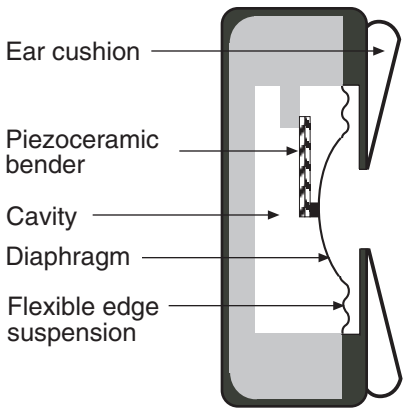


Figure 15.12 A piezoelectric headphone using a ceramic bender.

are combined in a *bimetal* fashion to achieve large excursions. Insert earphones may use a pile of thin piezoelectric disks operating in their thickness mode.

As mentioned, some modern piezoelectric headphones operate using piezoelectric plastic films. Superficially similar to electrostatic headphones, their mode of operation is, however, more similar to that of the bender design described previously.

15.7 ELECTROSTATIC HEADPHONES

Electrostatic headphones are generally used when the highest possible sound quality reproduction is desired. Most modern implementations feature an open-back design, as shown in Figure 15.13. Electrostatic headphones operate on the same principles as the electrostatic loudspeakers discussed in Chapter 14. The electrostatic headphone is, in principle, designed as an electrostatic loudspeaker with a comparatively large diaphragm. The problem of achieving sufficient loudness, often encountered when using electrostatic loudspeakers, is not as serious when using electrostatic headphones, since the latter are placed close to the ear.

There are two types of electrostatic headphones—those that use a bias voltage supply to provide charge for the membrane suspended between the electrodes and those that use *precharged* electret film to provide the charge that the audio voltage field acts on.

The mechanical system of the electrostatic headphone consists only of the film, which is mounted between two electrodes for push-pull operation. The film has low mass and is tuned for a low resonance frequency. The motion of the film is mass-controlled in the case of an open-back headphone. The resonances of the pinna and the ear canal will remain unaltered by the presence of the headphone since the film has such low impedance. If a sealing pad is used between the headphone and the head, the resulting cavity will exhibit acoustic resonances detrimental to the audio quality.

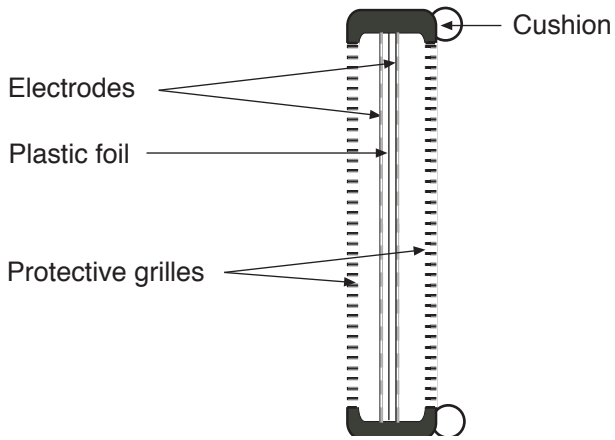


Figure 15.13 A circumaural headphone using a push-pull electrostatic transducer.

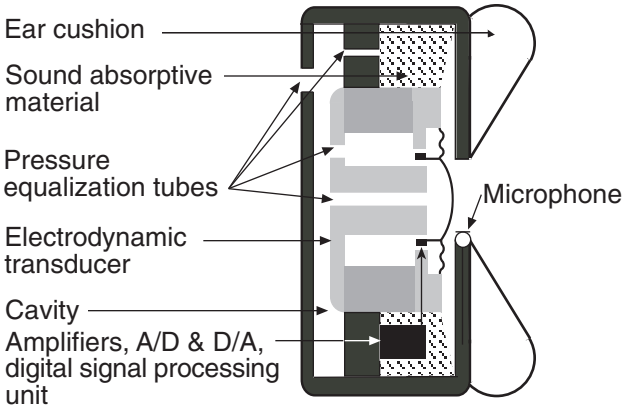


Figure 15.14 A typical headphone design using active control for noise reduction over a limited bandwidth.

As with most open-back headphones, sound insulation is nearly nonexistent; music listening, using these headphones, is best done in quiet surroundings.

15.8 NOISE-CANCELING HEADPHONES

Headphones are also used as noise-cancellation devices. The sound insulation of traditional headphones is mainly due to the mass of the headphone cups, their seal against the head, and the stiffness of the headband. Current signal processing techniques allow the use of active noise cancellation to reduce the environmental noise leaking in through these paths.

Two types of active control are provided by analog or digital signal processing. The noise-cancellation technique can be of the feed-forward or feedback type. Typically, attenuation of 10 to 20 dB over that of passive devices can be achieved in the frequency range of 100 Hz to 2 kHz.

To be effective, the electronic controller of the noise cancellation circuit must use an adaptive approach and an error microphone, as indicated in Figure 15.14.

The microphone is generally positioned as close to the ear canal entrance as possible. In the design shown, it is necessary to use pressure equalization tubes since the ear cushion seal must be effective to block out sound leaking in between the head and the cushion. The active circuit is usually effective in the range below 1 to 2 kHz—above this frequency, the noise cancellation generally relies on passive attenuation.

15.9 PROBLEMS

15.1 An electrodynamic headphone transducer of a hearing protector is coupled to a small sound coupler. Assume that the transducer current is determined by the resistance of its voice coil.

Task: What will the frequency response be of the sound pressure in the coupler (below any acoustic mode)?

15.2 An electrodynamic circumaural headphone that has an open back is to be used in reverse as a microphone when it is not being used for transmission. The *Bl*-product of the transducer is 0.01 N/A. The resonance frequency of the transducer is 2 kHz when it is attached to the head. For an rms input voltage of 1 V at 250 Hz, the headphone generates a sound pressure level of 94 dB inside the air cavity that has a volume of $3 \cdot 10^{-6}$ m^3. The voice coil resistance is 1000 Ohms.

Task: Calculate the output voltage of the headphone for a surrounding sound pressure of 94 dB at 250 Hz.

15.3 An electrodynamic supra-aural headphone is designed to give a flat frequency response at the head. The transducer can be regarded as a small dipole system.

Task: What will the approximate frequency response characteristic be of the sound pressure at far distance in an anechoic chamber?

15.4 An insert earphone is designed using a balanced armature stiffness controlled transducer that generates a volume velocity of $2 \cdot 10^{-8}$ m^3/s for an input voltage of 1 V at 400 Hz. Assume the mechanical system to have its resonance frequency at 8 kHz.

Task: Calculate the approximate frequency response at the eardrum for frequencies below 2 kHz.

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Digital Representation of Sound

16

16.1 INTRODUCTION

Digital representation of sound is commonplace. Digitization of sound, digital signal processing, storing, transmission, and conversion to analog signals are all used both for measurement and for communication purposes.

Most communication uses of digitized sound signals incorporate some type of signal information reduction, usually called *coding*, to reduce the transmission and storage requirements. Coding can be further categorized into *lossless and lossy coding*. Lossy coding may be optimized for the properties of human hearing; such coding is called *perceptually optimized*. Transmission technologies such as the *pulse code modulation* (PCM) used in personal computing, *compact disc* (CD), and *Digital Audio Tape* (DAT) use various forms of lossless coding. Most other modern sound distribution media and formats all use some form of *perceptual coding*.

This chapter only discusses the fundamental principles of some digitization and lossy coding techniques. More in-depth discussion at a fundamental level can be found in References 16.1–16.5, in journals, such as the *Journal of the Audio Engineering Society*, in standards and recommendations issued by organizations, such as ISO, IEC, ITU, EBU, as well as at many websites.

16.2 SAMPLING AND DIGITIZATION

Most digital audio systems will use the processing blocks shown in Figure 16.1. The incoming analog signal is sampled in the time and amplitude domains to convert its history into a *digital format*. Generally, sampling is done at regular intervals in time determined by the *sampling frequency*. The sampling frequency needs to be well above any audio signal frequency. In most basic consumer oriented analog-to-digital conversion systems for audio, the sampling frequency is 44.1 kHz or higher. A number of standard audio sampling frequencies exist—common are 48, 96, and 192 kHz. For speech and low quality music, lower sampling frequencies such as 22.05, 11.025 kHz, or lower, are also used.

To digitize the amplitude of the incoming audio signal, it is first band-limited by low-pass filtering to remove unnecessary, and possibly detrimental, high-frequency sound information as discussed later.

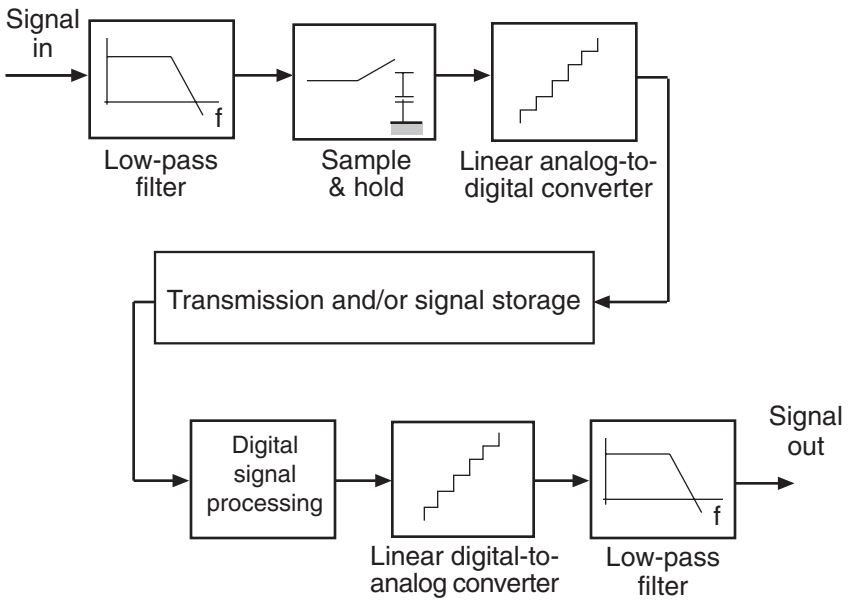


Figure 16.1 Basic digital signal sound reproduction system blocks.

Analog-to-digital Conversion

To provide time for the analog-to-digital converter (ADC) to convert the analog signal's voltage to a number, the voltage at the sampling moment is *remembered* by a sample-and-hold (S/H) network until the next sampling instance. The digitized momentary voltage can then be stored, delayed, coded, transmitted and reassembled into an analog electrical signal by a *digital-to-analog converter* (DAC, D/A) and a low-pass filter. This signal must then possibly be decoded, amplified, and fed to an electroacoustic transducer to become listenable. If there is no lossy coding and all technical components are perfect, the only signal degradation will be the low-pass filtering and the digitization processes at the input and output of the system. Any quantization system will introduce bandwidth-limiting errors, round-off errors, and clipping (or overload) errors.

Let us study an analog signal varying in time as shown in Figure 16.2a. If one samples the signal at equal time intervals, as indicated by the drop lines in Figure 16.2b, one will obtain momentary voltage spikes having the values indicated by the filled dots.

Such signals are difficult to work with; time is needed for the *analog-to-digital converter* to convert the voltage at the sampling instance into an equivalent number, so the voltage is kept constant to give the ADC the time necessary for the conversion as shown in Figure 16.2c. The building blocks of an ADC operation using a common summation and comparison method is shown in Figure 16.4. Note the DAC that is necessary for the operation and is shown in more detail in Figure 16.3.

The time between sample moments t_s gives the *sampling frequency* f_s as $f_s = 1/t_s$. When sampling at this rate, the signal spectrum will be uniquely determined up to the *Nyquist frequency* f_N, which is half the sampling frequency—$f_N = f_s/2$. One can show that if one filters the signal shown in Figure 16.2b by a

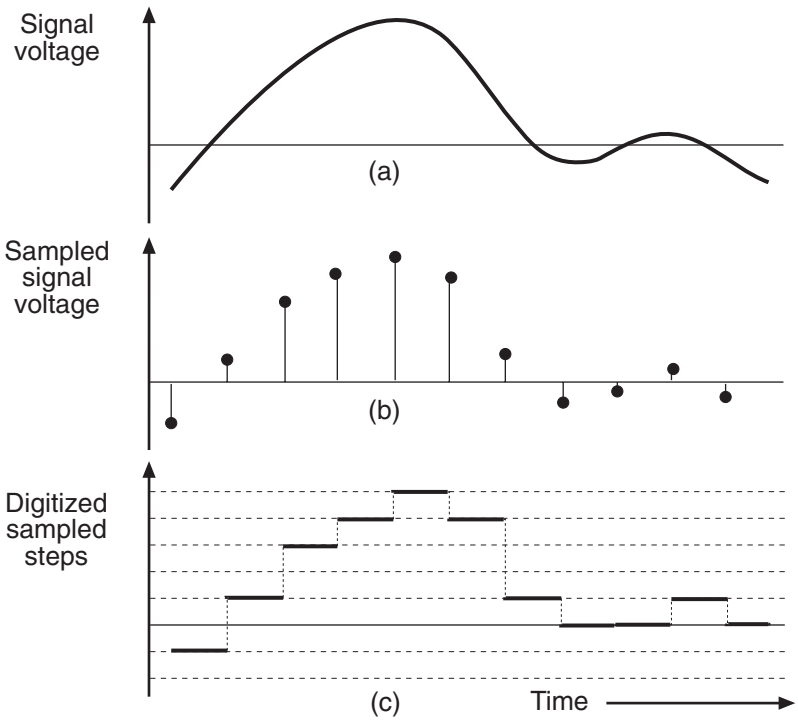


Figure 16.2 Principle of equal time interval sampling: (a) audio signal, (b) instantaneous samples, and (c) step samples.

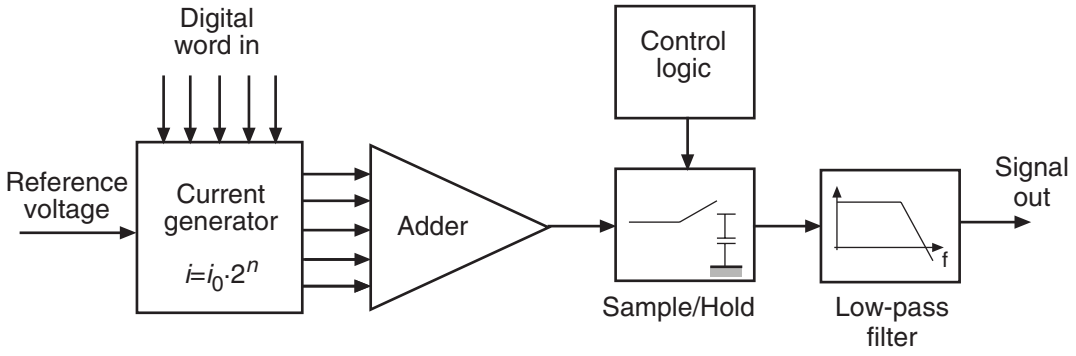


Figure 16.3 The basic building blocks of a digital-to-analog converter.

phase linear low-pass filter (such a filter has the same time delay at all pass band frequencies), one will obtain the signal in Figure 16.2a. Working on the signal shown in Figure 16.2c, there will be a slight drop in high-frequency content in the filter output that can be compensated for by digital or analog signal processing. The recovered signal will not have any added noise, linear or nonlinear distortion.

The sampling has not introduced any errors, provided the time instances of sampling are exact, the filter is linear, and there are no frequency components above f_N in the spectrum of the analog signal.

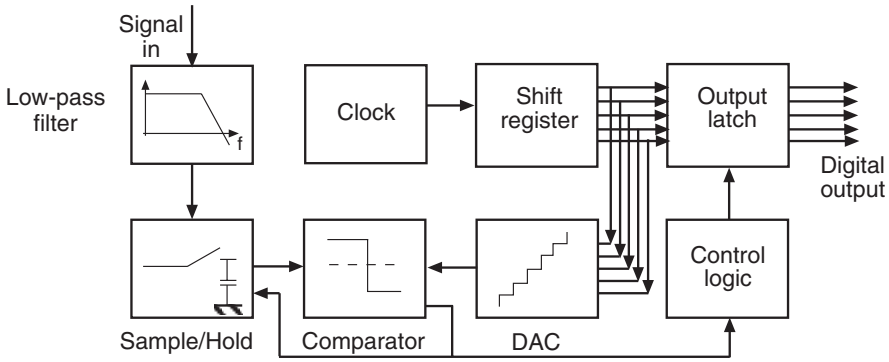


Figure 16.4 The basic building blocks of a successive approximation register analog-to-digital converter.

If, however, the input signal contains spectral components at frequencies above f_N, such components will be repeatedly mirrored around the Nyquist frequency f_N and its odd multiples, as shown in Figure 16.5. This phenomenon is called *aliasing* and will occur with any sound component having a frequency higher than f_N.

The audible result of aliasing is that the reconstituted signal at the output of the system will contain noise and whistles due to the folding of the spectrum around f_N and its multiples. This means in practice that there must be a low-pass filter at the system input (an *anti-aliasing filter*). Even if the microphones have poor response at frequencies above 20 kHz, there will be such frequency components in the signal at the digitizing system's input due to noise, nonlinear distortion, residuals from FM-stereo receivers and other sampling systems, etc. These signals must be attenuated using the anti-aliasing filter so that their aliased components are below the noise level in the audible frequency range.

There are two ways of solving the low-pass filtering requirement. One way is to use analog low-pass filters having steep slopes, the other to use "gentle" analog filters and oversampling. To avoid aliasing the filters need have 40 to 60 dB attenuation per octave above the *low-pass cutoff frequency* f_{LP}, typically 15–20 kHz. (Some filter action is provided already by the microphone and mixer.) A low sampling frequency will require corresponding low-pass filters having low cutoff frequencies and steep slopes. By necessity, there needs to be a guard-band between the low-pass filter's cutoff frequency and the Nyquist frequency. To minimize the guard-band one can use, for example, so-called elliptical filters; these can be designed to also have low ripple in the pass band. Typically, the ripple in the pass band will need to be less than 0.1 dB. Such filters will need to be *high order filters*, which, if analog, will need many reactive components such as capacitors and possibly inductors. Any high order filters will have resonances having high *Q*-values. Such analog filters will be costly, have poor aging, and poor transient response properties. The low-pass slope of the anti-aliasing filter (the *roll-off rate* in dB/octave) should be chosen according to the sampling frequency and the signal's spectral content.

Because of the mentioned drawbacks, another way of solving the low-pass filtering requirement is modern digital sound systems that use so-called *oversampling*—sampling at a frequency much higher than any anticipated signal frequency. The analog anti-aliasing filter can be of low order and inexpensive, while the *digital filter* used after sampling may use steep slopes to remove all frequency components

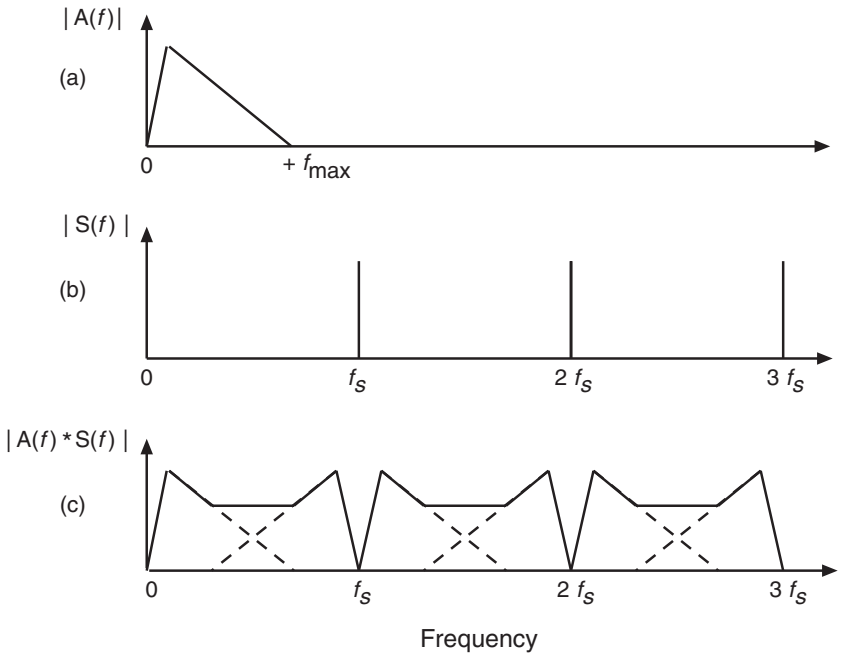


Figure 16.5 Principle of aliasing ($f_{max} > f_s/2$): (a) audio signal spectrum, (b) sampling signal spectrum, and (c) resulting sampled signal spectrum showing aliasing.

above the audio range. The digital filters do not change with age and are of course easily reproducible. The oversampling frequency is then reduced to the desired operating sampling frequency, for example 44.1 kHz, by *downsampling*. The downsampling is done using an interpolation process. The audio quality of downsampling routines can differ.

16.3 QUANTIZATION

Resolution and Quantization Noise

The instantaneous values of the amplitude of the analog signal at the times of sampling are digitized by the ADC in a quantization process. The accuracy of the numbers is determined by the sample and hold circuit, the ADC, and the sampling clock. Because of the limited resolution of any number scheme, there will always be an error, the *quantization error*—the difference between the true value and the number obtained in the quantization process. Figure 16.6 shows an analog signal's instantaneous value and its digitized sampled amplitude values; the gray bars show the difference between these values. One should note that the amplitudes and signs of the errors differ between samples. Figure 16.6 shows how the difference between the analog and digital signal results in quantization errors while the signal is being sampled.

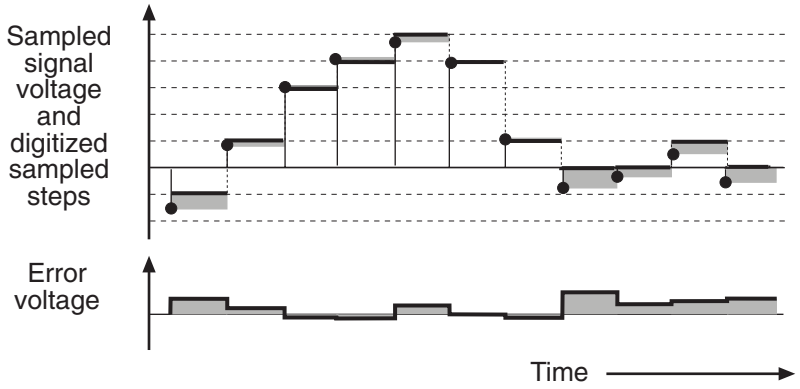


Figure 16.6 Influence of finite quantization step values on a time sampled signal showing the quantization errors at different time instances.

A binary digital number of n bits can have 2^n values. Since the analog signal can have both negative and positive values, the maximum signal amplitude will have the number $\delta \times 2^{n-1}$, where δ is the quantization step size, that is, the *least significant bit* (LSB). If the quantized signal has a wide spectrum of frequencies, and if its amplitude is large compared to the LSB, the error signal will have a random character (statistically independent between samples), and all its amplitudes will have equal probability over the interval ± 0.5 LSB. The error signal is sometimes inappropriately called *quantization noise* because of its near random character, in the case of high amplitude, wideband quantized signals. The RMS value of the error signal (quantization noise) will then be (see Reference 16.1):

$$\tilde{e}_{\text{noise}} = \frac{\delta}{2\sqrt{3}} \tag{16.1}$$

If the analog signal is a sinusoid having a peak value of $\delta \times 2^{n-1}$, the error signal will have a RMS value of $\delta \times 2^{n-1.5}$. The signal-to-quantization noise ratio will then have a maximum value of:

$$\left.\frac{S}{N}\right|_{dB} = 20\log\left(\tilde{e}_{\text{maximum}}\right) - 20\log\left(\tilde{e}_{\text{noise}}\right) = 20\log\left(\sqrt{\frac{3}{2}}2^n\right) \approx 2 + 6n \quad \text{dB} \tag{16.2}$$

From this expression, one could incorrectly infer that a perfect 16-bit linear quantizing system has a signal-to-noise (S/N) ratio of approximately 98 dB.

In reality, however, the audio signal being digitized usually needs about 20 dB of *headroom*; that is, the difference between the peak and long-term RMS values of the audio signal, corresponding to approximately 3 bits. In addition, the music audio quality becomes poor if less than 10 bits are used. This means that the dynamic range, in a sense, can be said to be only 3 bits—that is, 24 dB. This explains the need to move from 16-bit to 24-bit PCM for high-quality music recording and reproduction. The reasoning is illustrated in Figure 16.7 that shows the relationship between minimum digital resolution and dynamic range for three types of audio.

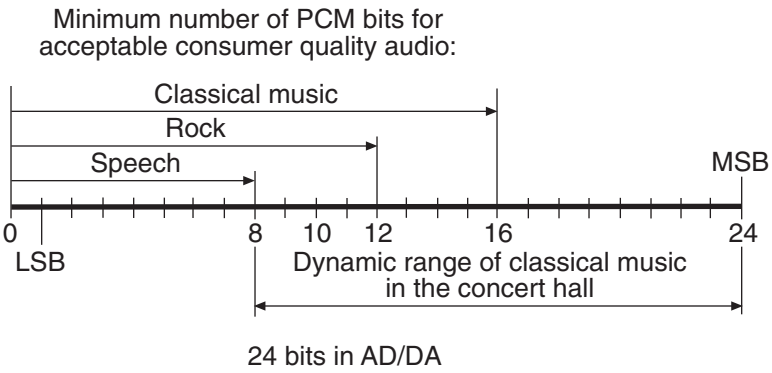


Figure 16.7 The graph shows the minimum number of bits needed for acceptable transmission of various audio materials using linear PCM. Below a resolution of 8 bits there is serious degradation of the signal due to quantization noise and distortion. These do not seriously affect speech intelligibility at resolutions above 4 bits, however.

This explains why the main advantages of the 16-bit CD format over the analog vinyl format for the reproduction of classical music are the absence of pops and clicks in the playback and the ease of copying and handling, rather than audio quality.

Some of the power of the quantization error signal is outside the audio spectrum $f < f_{LP}$. The power spectrum of the quantization error signal, under the assumptions made earlier, can be shown to be (see Reference 16.1):

$$G_{\text{noise}} = \tilde{e}^2_{\text{noise}} t_S \frac{\sin^2\left(\omega t_S/2\right)}{\left(\omega t_S/2\right)^2} \tag{16.3}$$

The power spectral density can be seen to drop toward higher frequencies; at the Nyquist frequency f_N, it is nearly 4 dB below the low-frequency value. Its behavior is, therefore, nearly the same as for white noise, which has a spectral density independent of frequency.

Since the frequency dependence of the power density spectrum of the sample-and-hold type pulse is the same as that of quantization error signal, frequency equalization of the sampled signal will lead to the quantization error signal also having a frequency-independent power spectrum. The quantization error will then sound like white noise. The level of the quantization noise within the audio band can be reduced by using oversampling. In oversampling, the sampling frequency is increased from the marginal values of 44.1 kHz or 48 kHz for 20 kHz bandwidth audio to, for example, 96 kHz or 192 kHz. The quantization noise will then be spread out over a bandwidth two or four times larger in these later cases. Because of the almost white noise character of the quantization noise, the level of the noise in the audio band will drop by 3 or 6 dB, respectively.

A traditional analog way of achieving perceptual noise reduction in audio systems is to use spectral pre- and de-emphasis. This means using a high-frequency boost at the system input and a high-frequency cut at the system output, usually as simple shelving filters as shown in Figure 2.4. Provided the input signal has a power spectrum that drops toward high frequencies, there will be a *transmission margin*; that is, the high frequencies in the input signal are not likely to force an overload on the input.

This margin can then be reduced by preemphasis of the high frequency part of the spectrum before quantization. Similar methods have been used for many years in gramophone recording, analog tape recording, television audio, and FM stereo radio transmission and are generally considered transparent from the viewpoint of audio quality.

Dither

Most real voice and music signals will have spectra very different from the wide-band signal assumed in this discussion. Often the signal will be composed of a small number of harmonically related tones. The quantization error signal will then have a much more disturbing characteristic. This can be understood from the following discussion.

Assume that the analog input signal to the system has an amplitude smaller than 0.5 LSB. Two extreme cases are shown in Figure 16.8. If the input signal is centered over one quantization step, the output will be a square wave as shown to the left in Figure 16.8. If, on the other hand, the analog signal is offset relative the quantization step, there can be a case of only a static output signal, a steady-state constant voltage, as shown to the right in Figure 16.8.

In either case, one must consider the signal as seriously distorted. A linear ADC and DAC transmission system has its highest S/N ratio when the (wide bandwidth) input signal has maximum rms value. In practice, however, one needs to keep the long term average of the signal at least 2 to 40 dB below its peak value, and the properties of a linear 16-bit PCM system are then similar to those of a high-quality analog system. The main advantage of the digital system is its robustness and the possibility of perceptual coding to reduce its transmission bandwidth and storage requirements.

The system performance can be subjectively improved by adding an extra random noise signal, called a *dither* signal, to the signal being digitized. Figure 16.9 shows the influence of one type of dither on the case shown in Figure 16.8. One notes that in both cases the introduction of dither results in a noisy pulse-width-modulated signal. If this signal is synchronously integrated over several time segments, the result will be that of the sinusoid input signal but with noise added as shown in Figure 16.10 for different numbers of synchronous averaging. Hearing does this type of integration, and dither can then subjectively improve the system performance, making it possible to hear tones having amplitudes

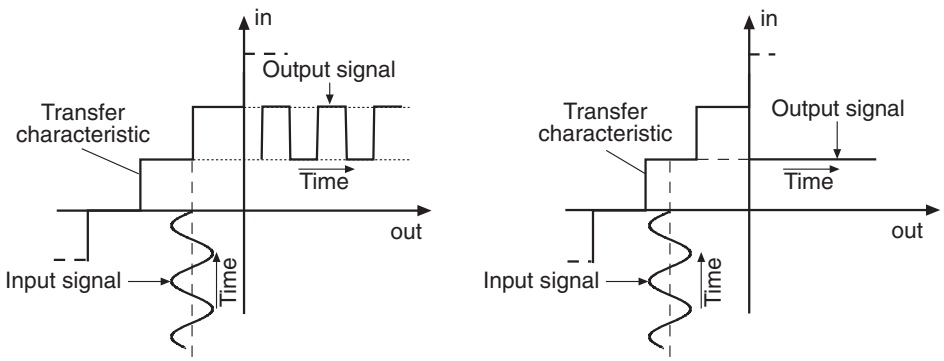


Figure 16.8 Two cases of digitization of an audio signal having amplitude of less than 1 LSB.

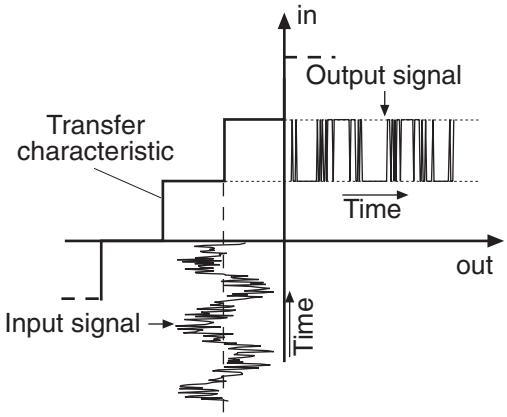


Figure 16.9 Influence of dither on the transfer characteristics shown in Figure 16.8. With dither noise the square wave becomes modulated.

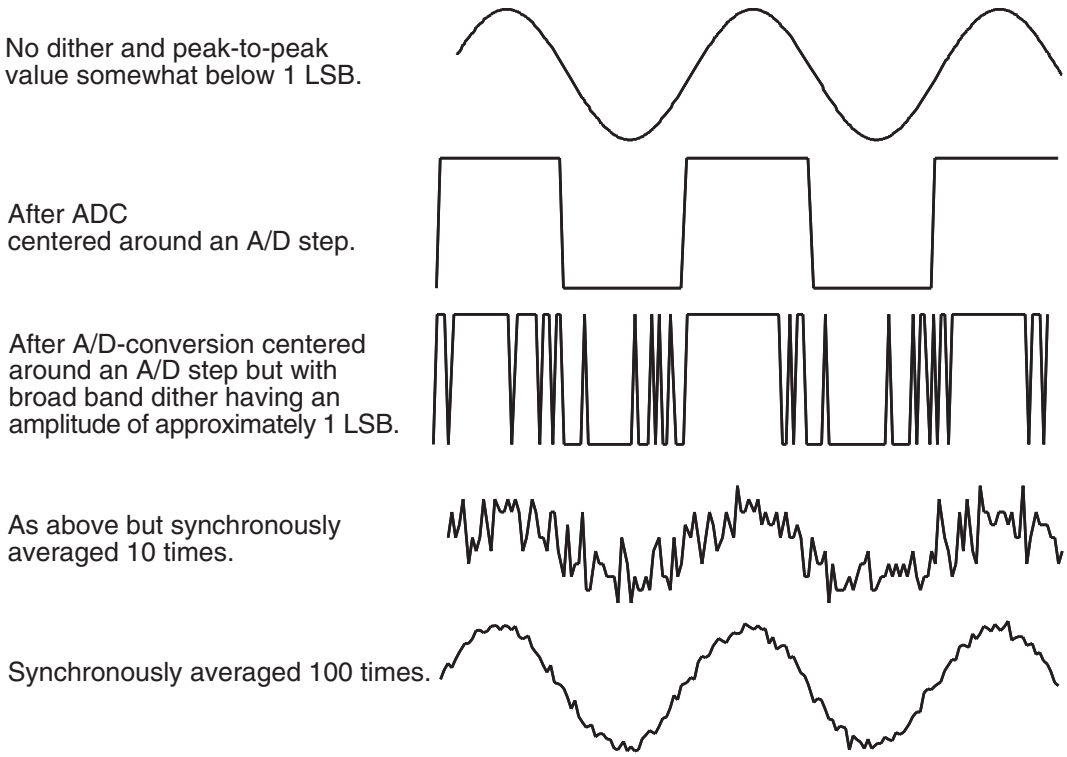


Figure 16.10 Influence of dither on a sine-wave signal when the signal has been D/A-converted and synchronously averaged over a number of periods.

smaller than the LSB. The audibility of the sine wave can be determined using the discussion of critical bands found in Chapter 3. One can show that in some cases, it is possible to perceive sines having a power of 10 to 15 dB below that of the noise in a third octave-wide frequency band (see Reference 16.12).

Dither also results in reduced nonlinear distortion. A sine-wave signal having a peak-to-peak value of 1 LSB will quantize into a square wave having the signal spectrum shown in the upper graph in Figure 16.11. Using dither noise, the harmonics of the sine may be reduced in amplitude as shown in the lower graph in Figure 16.11.

For example, a simple dither noise can have equal probability over a ±0.5 LSB interval. This is difficult to obtain using linear analog techniques but can be generated digitally using a DAC. High-

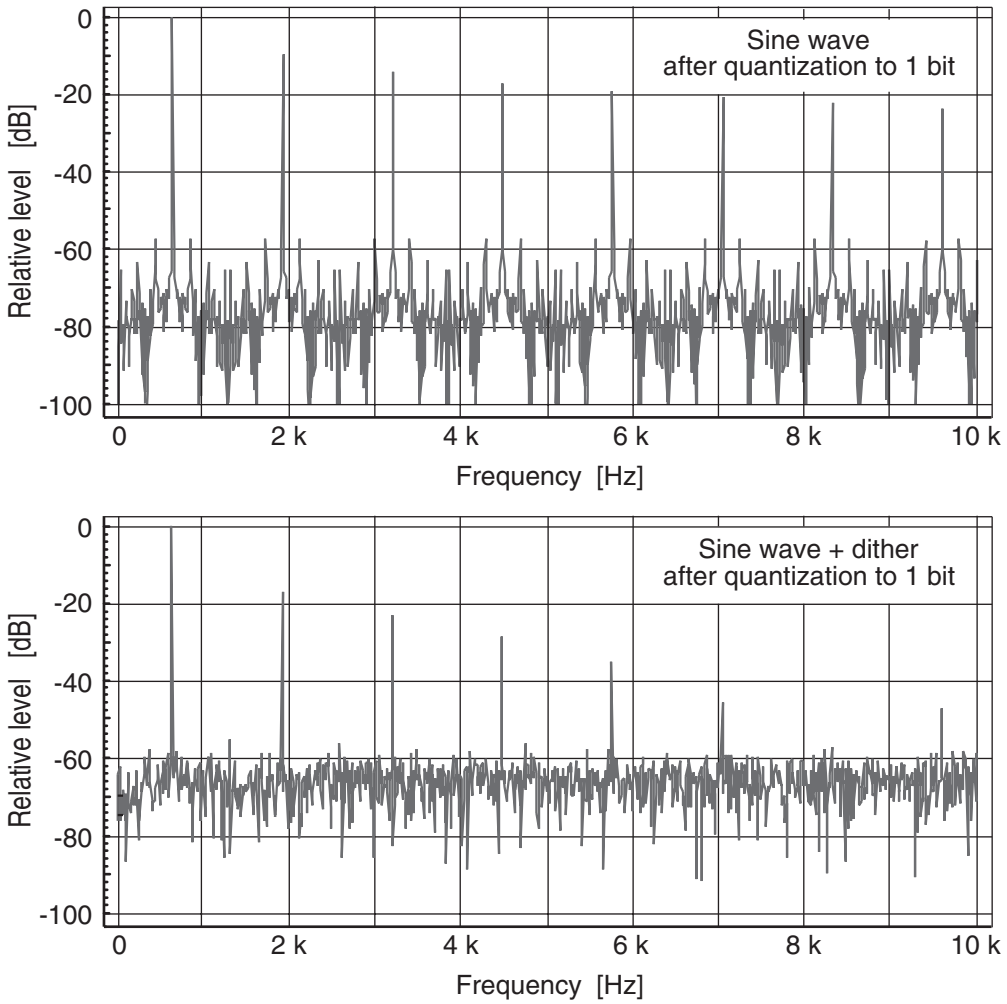


Figure 16.11 The spectrum of a sine-wave after quantization to 1 bit, and of a sine wave + dither signal after the same quantization.

frequency noise present in the input signal is often advantageous since it acts as dither. Different audio signals need different optimum dither noise characteristics. In many cases, however, a dither noise having a triangular probability density over the ±0.5 LSB interval turns out to be a good choice.

Quantization Methods

There are many ways to achieve the task of converting the analog signal value at the sampling instance to a number. It is reasonable to expect a faster conversion to be more coarse than a slower conversion, simply because there is not enough time for the electronic digital circuitry to convert with higher resolution. The power consumption by the ADC is also an important issue.

There are two extreme cases. The first is to convert to the desired resolution within the time window t_S accorded to the *sample-and-hold circuit* (SHC). Usually t_S is set as $1/f_S$; this is the method used in PCM converters using the *successive approximation register* (SAR) approach, as shown previously in Figure 16.4. Such a converter has inside it a precision DAC that outputs the instantaneous value of the conversion process (see Figure 16.3). The digital input to the DAC is numerically ramped in a positive or negative direction until the DAC output is the same as the analog input voltage within the resolution of 1 LSB. At this point, the digital input value is output to the buffer at the ADC output for further processing. Typically, such ADCs output a binary code, 6, 8, 12, 16, or 24-bits wide.

At the time of writing, most personal computers have a built-in sound card having a 16-bit ADC. Such a converter, in the worst case, has to step through all possible 2^{16} values to reach its settling point (where the output of the internal D/A converter is the same as the input voltage). Usually, however, the final value is reached in a series of binary comparison steps.

In conventional audio CDs, the audio data is represented as two channels of 16-bit wide data words at a sampling frequency of 44.1 kHz, whereas the old DAT format uses a 48 kHz sampling frequency. Modern ADCs use 24-bit resolution at a sampling rate of 96 or 192 kHz.

The sampling rates can be converted between one another by interpolation or decimation processes, as required. Such sampling frequency conversions may generate errors that can be detrimental to the subjectively determined sound quality.

A completely different approach is that used in *delta modulation* (DM) and *sigma-delta modulation* (SDM) converters. These operate at very high speed but with low resolution; in the extreme case the resolution is only just one bit. The principles and basic operation of 1 bit DM and SDM converters are shown in Figures 16.12 and 16.13. The output of the one-bit conversion process is integrated, and the integrated value is continuously compared to the analog input value. The bit sign is then changed when the appropriate value has been reached. The number of bits at the required output sampling frequency (for example, 48 kHz) is accumulated into a corresponding 16-bit word. Because the converter must, in the worst case, pass through all 2^{16} values, the 1-bit sampling frequency must be 2^{16} times higher—that is, 3.1 GHz. Such a high sampling frequency is currently impractical, and various ways are used to reduce the sampling frequency.

SDM converters are similar to DM converters but use the integrator block before the quantizer. An obvious advantage of the DM and SDM converters is that only a simple integration circuit is necessary to retrieve the analog signal. This integrator, in a sense, does the same task as the DAC used in SAR converters.

As discussed previously, the quantization depth—that is, the number of bits used, determines the quantization noise. The noise of the DM or SDM converter is much higher than that of the PCM con-

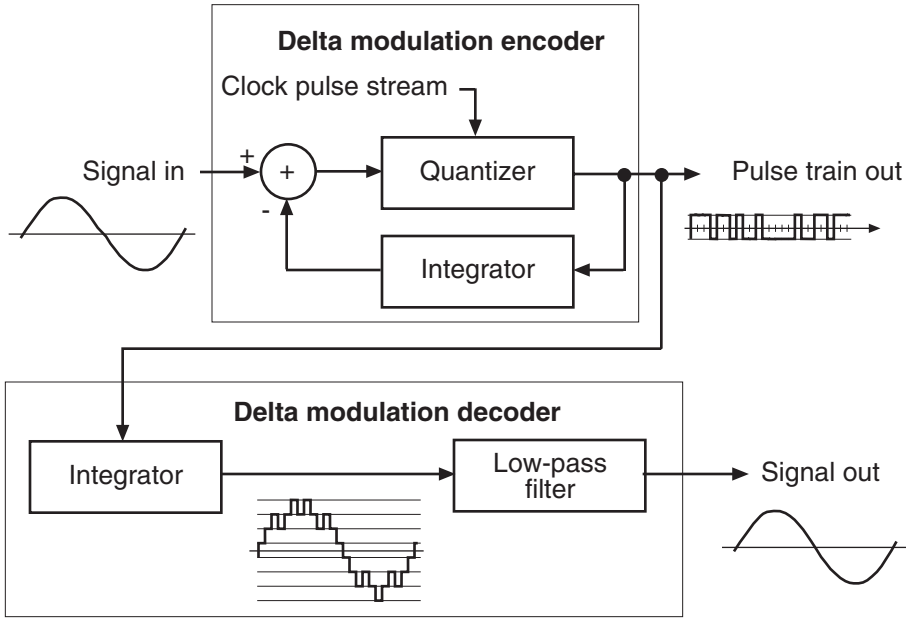


Figure 16.12 Basic operating principles of DM encoders and decoders.

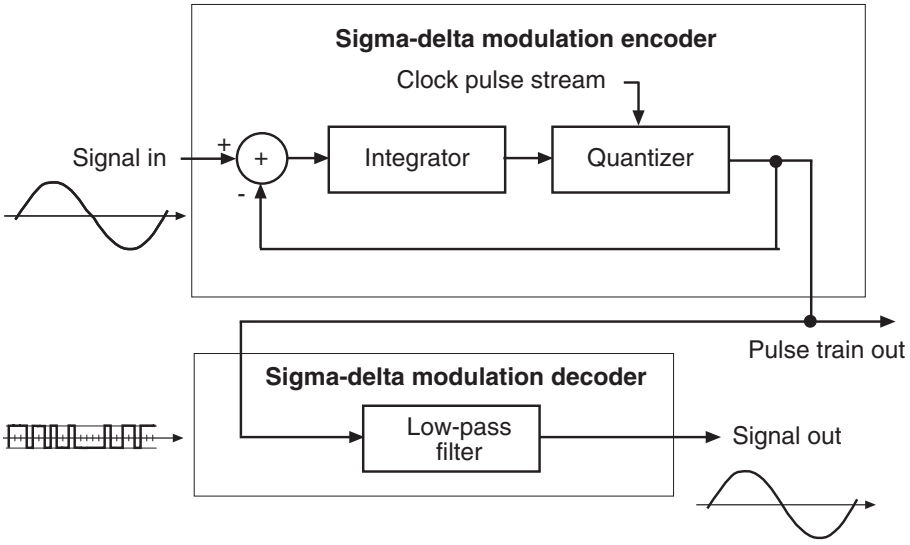


Figure 16.13 Basic operating principles of SDM encoders and decoders.

verter because of the small number of bits. Because of the high sampling frequency, however, the quantization noise power is spread out over a much wider frequency range than is typically the case in PCM converters, so the noise can still be low at audio frequencies. A particular problem with DM and SDM converters is the *idling noise*; that is, the noise when the input signal is close to zero. This idling noise

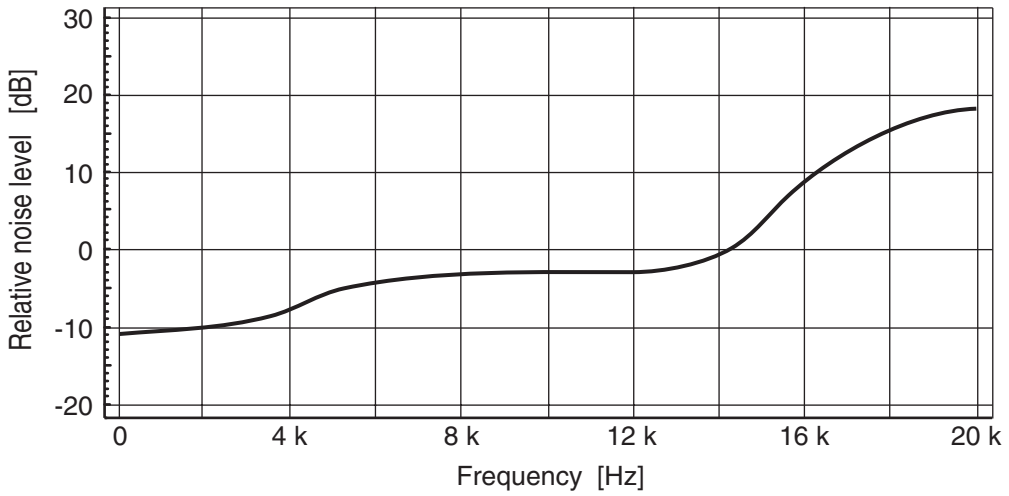


Figure 16.14 An example of a noise-shaping contour that moves noise power from the 1 to 5 kHz frequency region where the ear is sensitive to the higher frequency region of the audio spectrum. (Adapted from Ref. 16.5)

tends to take the form of tones. Consequently, there is a need for dither noise addition in these applications as well. The noise of the SDM is further characterized by high correlation from sample to sample.

The quantization noise can be regarded as an essentially white noise signal that is injected at the quantizer. This noise is affected by the feedback loop of the converter and appears as a high-pass filtered noise having a power spectrum that increases with frequency. The term *noise-shaping* is often used to describe the action of the system. Even more radical reduction of the audio band noise of the SDM converter can be achieved by using so-called higher order noise-shaping. The noise-shaping can take the masking characteristics of hearing into account; this is so called *perceptually optimized noise-shaping*. In this case, one finds that the oversampling should be at least six times the highest audio frequency of interest. It is perceptually advantageous to remove noise power from the 1–5 kHz frequency region and put the noise in the frequency region above 15 kHz where it is less noticeable. An example of such a noise-shaping frequency response is shown in Figure 16.14.

Other types of quantization schemes are *pulse-width modulation* and *pulse-density modulation*. These quantization schemes are often used in the DAC section of audio units. *Flash converters* are characterized by extremely short conversion times but feature comparatively fewer bits, that is, coarser resolution.

16.4 ADDITIONAL PROBLEMS IN A/D AND D/A CONVERSION

Sample-and-hold Circuits

It was mentioned previously that some time is required for the ADC to calculate the value of the waveform at the sampling instant. This time is provided by the S/H capacitor that ideally holds the sampled voltage of one sampling instance constant until the next sampling instant. If the S/H capaci-

tor is not able to keep the voltage constant during the conversion process, there will be an error in the assigned value of the sampled signal. Besides having low leakage, low hysteresis, and dielectric absorption, the S/H capacitor must be able to be quickly charged by the amplifier feeding it. This amplifier must have high *slew-rate* (voltage time derivative), even in the presence of a capacitive load.

Nonlinearities in A/D and D/A Converters

From the viewpoint of audio signal quality, equal quantization step size is of larger interest than amplitude accuracy.

For an 8-bit converter in which the signal is close to the middle of the conversion region, the binary value must switch from 10000000 to 01111111. If the error in the first bit is large, this will result in large errors in the middle of the conversion region, which is a problem in converting audio signals since they do not carry a DC voltage component. To avoid this so-called modulation noise the accuracy of the most significant bits must be 1 to 2 bits better than one LSB. Typically a DC voltage of about 10% of the full scale maximum value is injected to reduce the problem of modulation noise, which only results in a reduction of the AC signal full range level by around 1 dB. A different approach is that zero input voltage is in the middle of a quantization interval so that no input will result in essentially no active bits.

In any digitizing system, it is important that the input signal be suitably amplitude-limited before, within, and after the anti-aliasing filter. Any limiting in or after the filter will result in high-frequency harmonics that will be mirrored around the Nyquist frequency and that will fall into the audio range. It is also important to consider the transient properties of the anti-aliasing filter so that even a square wave input does not cause clipping (and associated harmonics) due to ringing in the filter. This is particularly a problem in active analog filters. Hum and small variations—noise—in the timing of the sampling (so-called *jitter*) may also cause nonlinearities in the sampling of high-frequency audio signals.

To improve DAC properties, a SHC is often used at the output of the converter. The purpose of this SHC is to sample the converter output voltage after it has settled to its proper value. The SHC is followed by a low-pass filter to remove high-frequency components that may interfere with recording equipment, causing aliasing. The *slew-rate* properties of the low-pass filter influences the output as shown in Figure 16.15. A low slew-rate capability will cause distortion in high-amplitude, high-frequency sounds.

16.5 CODECS, UNIFORM, AND NONUNIFORM QUANTIZATION

Most commercial high-quality digital audio recordings are at the time of writing issued on CDs using 16-bit PCM with a sampling frequency of 44.1 kHz. Digital tape recording on DAT uses 48 kHz. If recordings are to be edited and manipulated using digital signal processing, PCM with a resolution of 24 bits and a sampling frequency of at least 96 kHz must be used to avoid audible artifacts once the signals have been downsampled to CD characteristics. A typical audio CD will contain approximately 45 minutes of stereo audio that requires about 650 Mb of data. For transmission by radio, television, or the Internet, the CD data rate is too high and must be reduced. This is done by using various processes called compression.

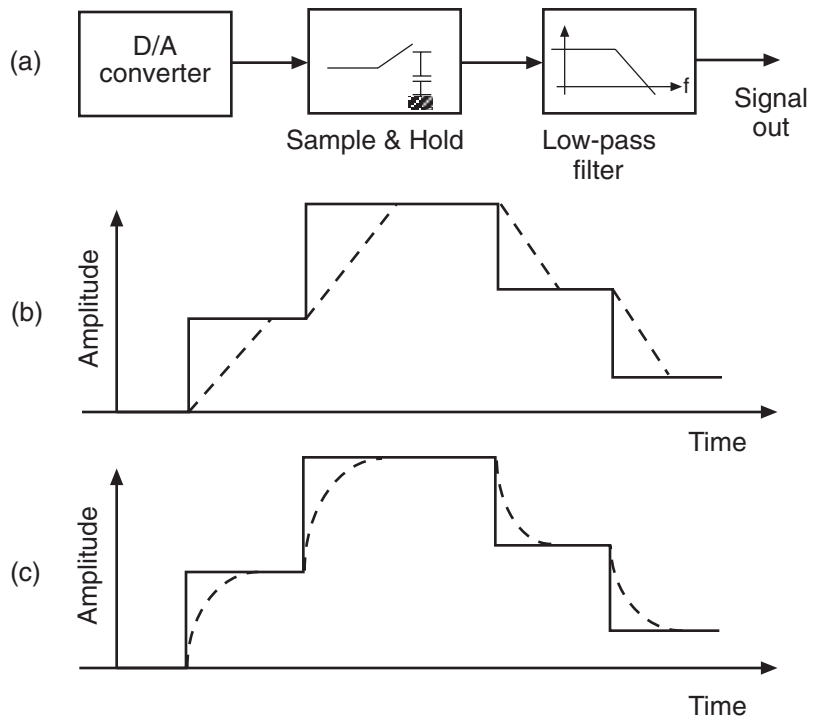


Figure 16.15 (a) Basic building blocks of the output section of a D/A-system, (b) slew-rate limited slope between settling levels, and (c) exponential slope between settling levels.

One usually distinguishes between *lossless* and *lossy compression*. Lossless compression is such compression that results in smaller data storage requirements but that allows exact data recovery. Lossy compression allows much smaller data storage requirements than lossless compression, but data cannot be recovered exactly. This, however, may not be a problem if the receiving system, such as human hearing, cannot detect the loss. This type of compression is called *perceptual coding*.

The file format MPEG II layer III, often just called MP3, which is used for music storage and transmission, is based on the use of lossy compression. The same applies to the PCM technologies used in digitizing voice in telephone applications such as GSM; here the quantization is typically done to 8 bits and at a sampling frequency of 13 kHz (see Reference 16.7). One does not need to be a trained listener to hear the difference between ADC and DAC conversions at 16-bit 48 kHz and 24-bit 96 kHz; it is quite easy to hear the differences at low audio signal levels. The differences are particularly easy to hear on long, reverberant piano tones.

Compression schemes of various types are consequently needed to reduce the amount of space necessary to transmit and archive sound data recorded at high bit rates. Some of the important issues in compression are fidelity to the original, the speed of compression and decompression (data rate and delay), and complexity. Audio data compression is a discipline in its own, and the reader is referred to the references for this chapter as well as to timely internet resources for further information (see References 16.5, 16.6, 16.8).

Lossless Compression

Lossless compression does not have any audio quality issues. In lossless compression, patterns in the data are targeted, and repeating sequence patterns are replaced by shorter symbol sequences, thus achieving a compression that will depend on the recurrence of certain patterns and their respective lengths. This type of compression can often achieve considerable data file size reductions and is comparatively fast, both to compress and to decompress, for audio codecs. A file size compression of typically 50% can be achieved. The main advantage is that the file is stored without any loss of information. This is used to compress software and data files.

Nonuniform Coding

The use of nonuniformly coding ADC and DAC systems allows a larger dynamic range, since the S/N ratio with signal will be less dependent on level. Nonuniform quantization is often used in speech applications since it is simple and inexpensive to implement. One can regard nonuniform quantization as a linear quantization but with pre- and post-distortion, as shown in Figure 16.16.

The relative height differences are reduced before quantization (companding) and increased after the DAC (expanding). Two types of companding are common: power-law and logarithmic companding. In practice, it is often difficult to match the pre- and post-distortion characteristics. In addition, the methods result in the generation of harmonics of the waveform, which means that the sampling frequency must be increased or a further low-pass filter used. It is of course also possible to use a conventional linear PCM quantizer and then just retain bits according to a nonlinear characteristic to reduce data storage and transmission requirements. In this case, the sampling frequency can remain the same.

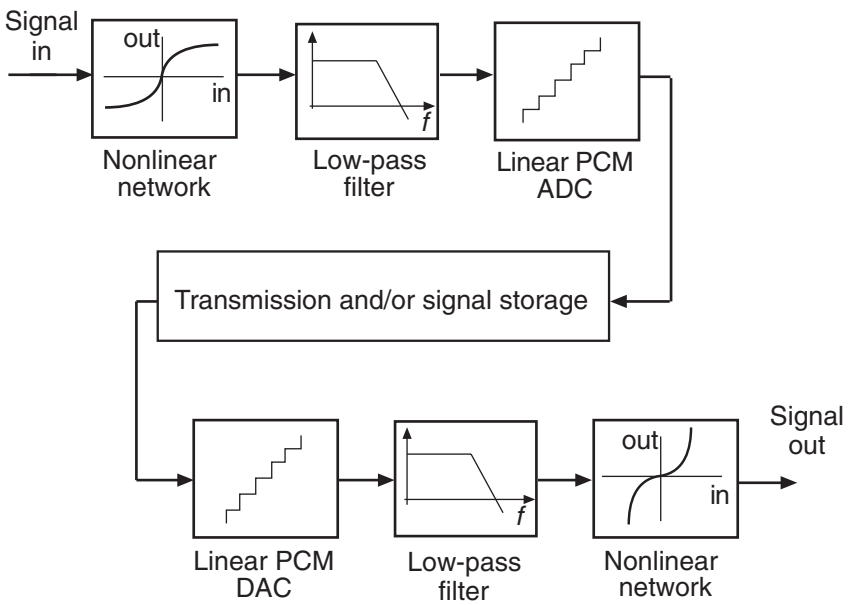


Figure 16.16 A PCM A/D and D/A-system implementation of nonuniform coding using analog nonlinear circuits.

Floating-Point Quantization

Human hearing features a wide dynamic range and *gain-riding*. Unless a signal is extremely quiet, loud, or contrasted, we usually do not consider its loudness. Gain-riding in this context means that hearing continually adjusts the perceived level to be similar; the hearing mechanism effectively reduces the apparent dynamic range. To reproduce sounds as we hear them in classical music, a dynamic range of about 100 dB is needed, for voice about 50 dB (the level difference between whispers and shouts). Over a short time, however, the dynamic range does not need to be as large as 100 dB; it takes some time for hearing to recover its sensitivity after it has been exposed to louder sounds.

In floating-point quantization, one tries to give the same resolution to both weak and strong signals. The aim is to reduce the number of bits used for storing and transmission, compared to linear PCM, but with little subjectively perceived audio quality reduction. This is possible if the bit scheme is chosen appropriately. The basic layout of a floating-point quantizer is shown in Figure 16.17.

One forfeits accuracy of strong signals by giving them the same number of bits as weaker signals and instead describes the *strength* by a scale factor. The sample value is quantized into a word having two parts, the mantissa and exponent. The mantissa and exponent are used to describe the basic quantization and gain adjustment factor respectively. A system using 4 bits for the exponent and 12 bits for the mantissa is denoted E4M12.

When dealing with floating-point quantizers, it is particularly important to separate between dynamic range and S/N ratio.

Block Floating-Point Quantization

In *block floating-point quantization*, one looks at the average strength of the signal over a number of samples making up one block of data. The principles of a block floating-point quantizer are shown in Figure 16.18. The choice of block length is critical and will depend on the audio source.

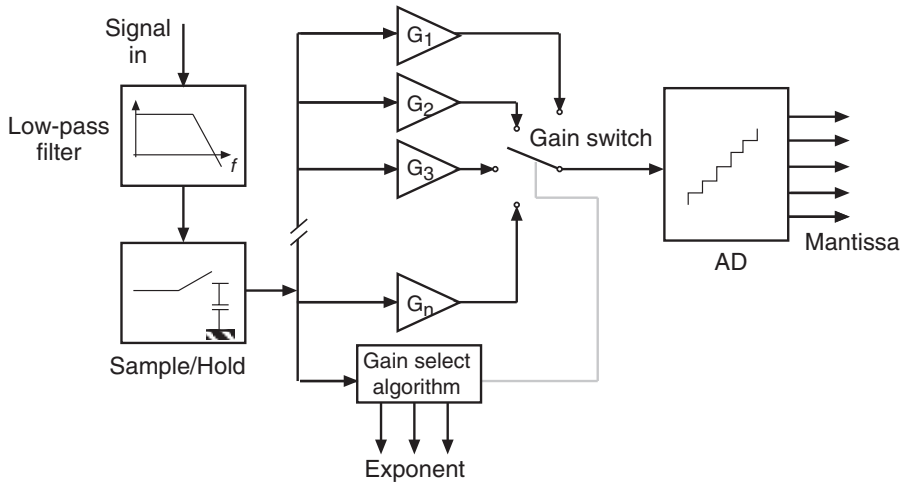


Figure 16.17 Operating principles of a floating-point ADC.

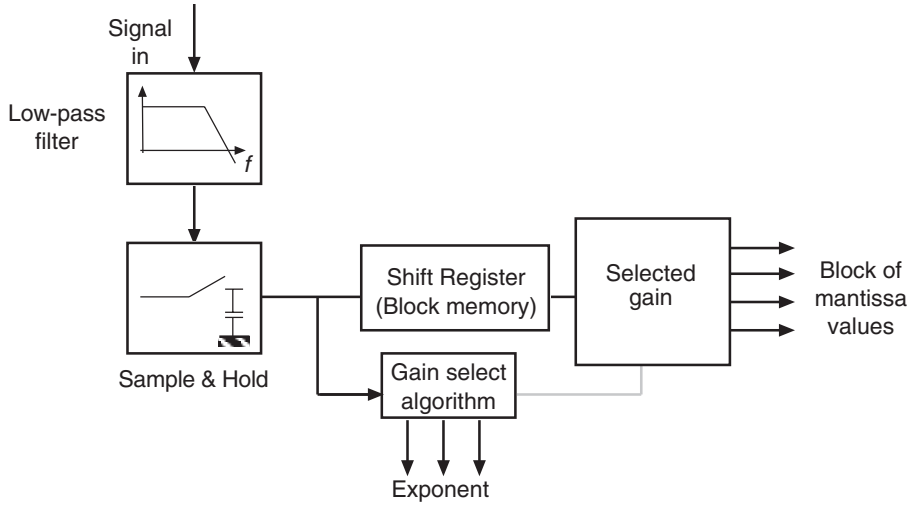


Figure 16.18 Operating principles of a block floating-point A/D converter using a linear PCM A/D converter and subsequent calculation of exponent and mantissa to set block gain.

The masking of pure tones by noise was subjectively determined and results show that a 15-bit mantissa was sufficient (see Reference 16.10). Further experiments using music have shown that in using a block size of 50 ms (approximately 2,200 samples at a sampling frequency of 44 kHz), one could use E3M13 format with exponent gain steps of 6 dB giving a range of 66 dB + 48 dB. The maximum ratio between signal and noise without signal (S/N_{ns}) will be approximately 114 dB if 1-bit dither is used, but the maximum ratio between signal and noise with signal (S/N_{ws}) will be only 66 dB (see Reference 16.9).

If S/N_{ws} is too low, or if the block length is unsuitable to the audio content, one can experience *modulation noise* or gain-riding artifacts. One advantage of the system is that the exponent bits can be neglected if needed, and one then has an effective companding system that may be useful if one wants to avoid hearing tiredness and hearing damage.

16.6 LOSSY COMPRESSION AND PERCEPTUAL CODING

At the time of writing most commercial ADC and DAC systems for high quality sound reproduction use the CD quality standard of 16-bit PCM at a sampling frequency of 44.1 kHz. Sometimes it is necessary to reduce the quality requirements to be able to transfer the information with less bandwidth or memory requirements. Using PCM coding for speech, one can show that 6–8 bits and a sampling frequency of 10–12 kHz is sufficient for good quality speech transmission from the viewpoint of speech intelligibility. Often, however, even this reduction of transmitted data is not sufficient, for example, in multichannel applications and mobile telephony.

Time-to-Frequency Mapping

Because of the perceptual properties of human hearing, it is often advantageous to transfer the signal from being represented in the time domain—the way we normally hear it—to being represented by spectra of suitably long blocks. This allows signal manipulation taking into account frequency masking, for example. It also allows efficient coding techniques, particularly regarding speech. Two alternatives for time-to-frequency mapping are filter bank methods and transform methods.

Filter Bank Methods

In using *filter bank methods*, one applies digital filters to the signal. These subdivide the audio frequency range into a number of frequency sub-bands, as shown in Figure 16.19. The output of each sub-band filter is then resampled at the optimum bit resolution for quantization noise in that band to be inaudible. The available bits can then be distributed to the various sub-bands by, for example, perceptually relevant methods. At playback, the different bands are reconverted from the digital to analog domain and recombined to become a reasonable replica of the original signal. The method is difficult to use as the number of frequency bands is increased. The principle of frequency band subdivision of the audio signal is, in a sense, similar to that used in the Dolby A analog compander system.

Transform Methods

The operating principles of a transform coder and decoder are shown in Figure 16.20. *Transform methods* are similar to the filter bank methods; it is mainly the approach to filtering that is different. The transform methods may use Fourier, wavelet, and other transforms. Fourier transforms are often used since they are easy to make relatively computationally efficient.

The Fourier transform is applied to a short block of audio data with a spectrum that should not change much over the time interval. As discussed in Chapter 3, speech can be considered as noise in

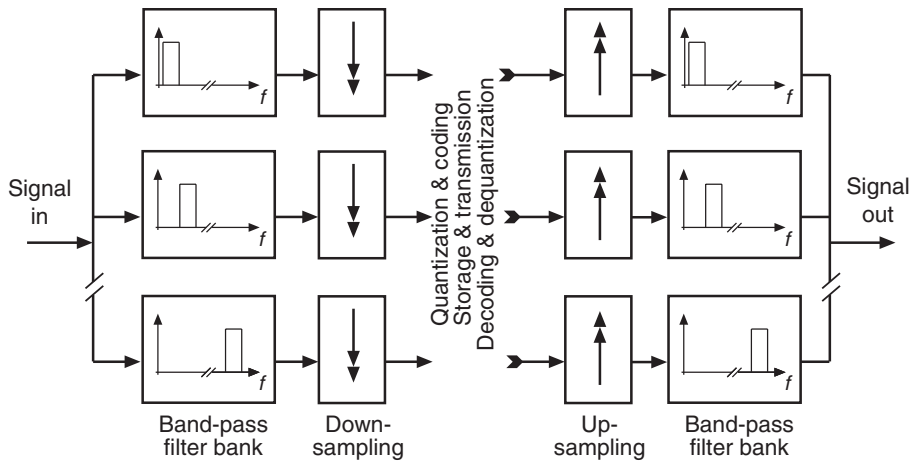


Figure 16.19 Operating principles of a coder & decoder using the filter bank method.

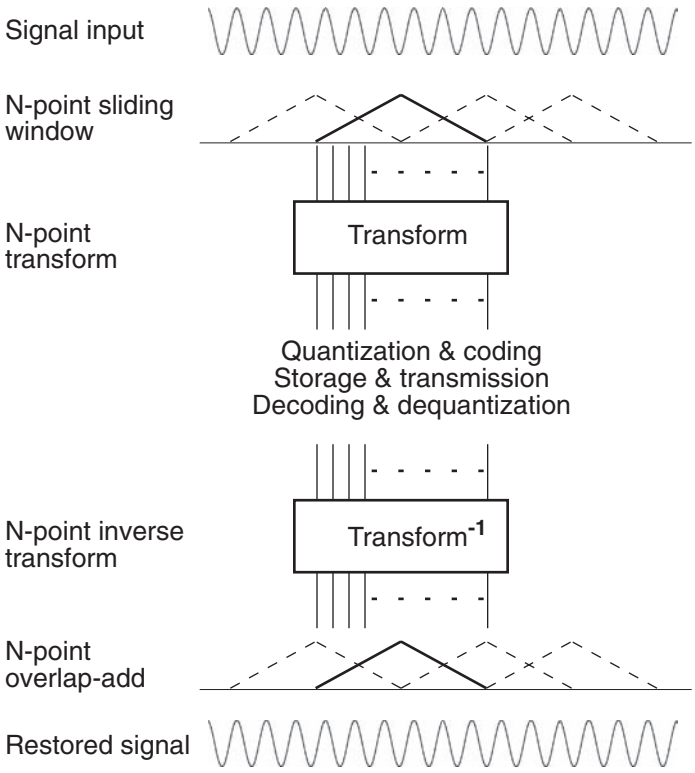


Figure 16.20 Operating principles of a transform coder & decoder using windowing and overlap-add methods.

a number of frequency bands (octave or third octave), modulated at different rates (compare to the speech transmission index as discussed in Ch. 3). The basic modulation frequency is approximately 7 Hz. Audio blocks 10 to 20 ms long are, therefore, comparatively short and suitable for Fourier analysis. The Fourier analysis can be performed either in hardware or by software in a computer.

The Fourier analysis effectively provides a number of filtered outputs according to the number of samples in the frame or block analyzed. These filter outputs can then be used much in the same way as the filter outputs described previously. The Fourier transform filters will have side lobes—energy from the main lobe will spread to a wide range of frequencies both above and below the main (lobe) filter frequency. This will affect the calculation of masking. For optimal results, different window functions are required for different audio signals. The need to switch window functions *on the fly* further complicates the coding and may create audible artifacts between blocks of data.

One drawback of this method is the latency due to the need for an entire N-point block of data (typically at least 1,024 samples) to be collected before the transform can be executed; however, any frequency filter will exhibit signal delay. Some coders use a combination of the transform method with the filter bank method. Because an entire block of data is processed simultaneously, the quantization noise of the process is spread out over the entire block. If the block length is too long, this may result in audible noise, as shown in Figure 16.21.

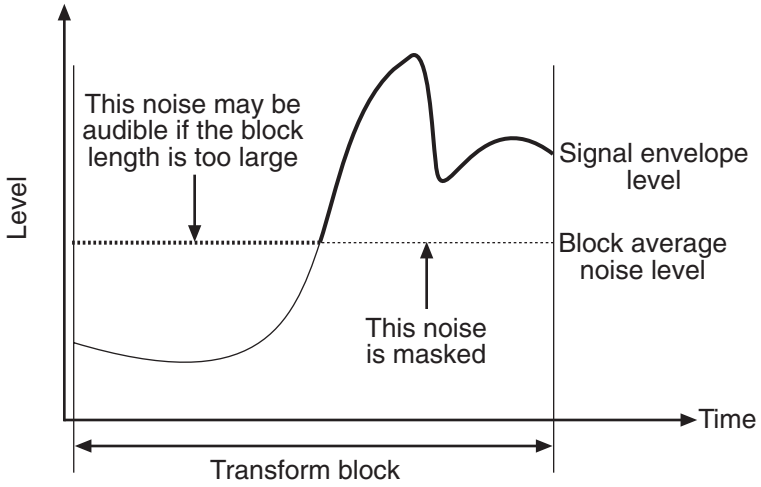


Figure 16.21 Quantization noise can become audible if the block length is excessively large. (After Ref. 16.4)

Perceptual Coding

Perceptual coding uses filtered versions of the input signal, sampled at high resolution and high sampling frequency. The filtered signals are then analyzed from the viewpoint of masking. It is well known that masking takes place both in the time domain and in the frequency domain. However, most perceptual coding schemes use only the principles of masking in the frequency domain, also known as simultaneous masking (see Figure 3.20). The masking will differ between individuals and signal levels. It turns out, however, that effective use of simultaneous masking can be done even with simple approximations to the actual masking properties of human hearing.

It is important to remember that most investigations of masking curve properties have been done with simple maskers and maskees, such as pure tones and narrow-band noise. In perceptual coding, one assumes that these measured characteristics are applicable to general audio signals and that tones and narrow-band noise have the same masking characteristics.

The excitation patterns of narrow-band noise signals, at the same sound level, are fairly similar, irrespective of frequency, if shown on a bark band frequency scale. The patterns will vary with sound level; lower masker levels lead to less prominent masking. Many coding schemes assume this effect to be negligible and use a triangular *spreading function* for calculation of masking effects. An example of spreading functions is shown in Figure 16.22. A further complication is given by the fact that audio signals, typically, will contain many maskers and maskees. This leads to the problem of how to account for the overlap of spreading functions.

If a sound would be inaudible in the recording situation, should it then be coded? The hearing threshold is defined by tones or band-limited noise measured at certain sound pressure levels. Microphone, mixing console, and other noise sources at recording will usually have noise spectra stronger at most frequencies than the hearing threshold of a normally hearing person. The standard assumption for 16-bit PCM is to assume that its maximum digital value corresponds to a sine having a sound pressure level of 96 dB. It is important to remember, though, that one does not know the playback sound pressure level of the decoded signal.

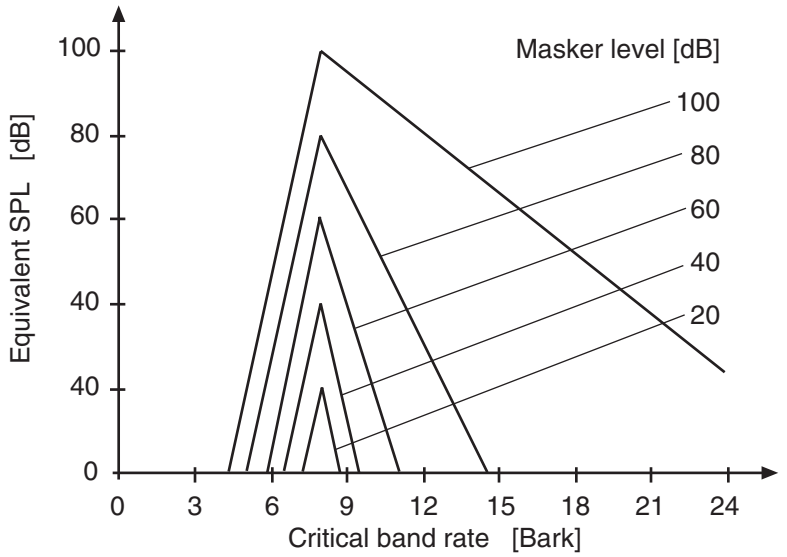


Figure 16.22 Examples of slope-spreading functions approximating the masking curves shown in Figure 3.20.

After sampling the signal, a block of its data will be analyzed from the viewpoint of masking. This can be done in various ways. A straightforward way is to look at the level of each Fourier analysis frequency and calculate its contribution to the total spreading function. This looping method is slow since it requires that all (typically thousands) frequencies be analyzed. Another way is to bundle the contribution from all frequencies into appropriate critical bands. Still another method is to find the *tonality* of a signal and only take into account the dominant tones.

Having decided on the influence of masking, the final step in coding is to decide on how the available bits should be allocated; strong frequency components will be represented by many bits and those that are masked by no bits. The quantization noise can be controlled for each frequency band and shaped so that it will be inaudible. Note, however, that there now is a quantization of the signal in the frequency domain as well.

In summary, one can say that different audio signals will perform differently in various coding/decoding schemes, and further that, even within a standard, different software implementations of the standard can sound differently.

Stereo and Multichannel Coding

The previous discussion of perceptual coding strictly concerned only a single channel of audio data. In practice, most audio is represented as stereo (2 channels) or multichannel (5.1 surround or higher, such as in DVD film and on Super Audio CDs). Holographic techniques such as wave field synthesis (WFS) using hundreds of channels may become common in the future. Can one use redundancies between channel data to further reduce the bit rate so that it will not be directly proportional to the number of channels?

If one looks at the time data, there is generally little correlation between channels. There is more correlation in the frequency spectra of the different channels. This can be used to reduce bit rates.

Human binaural hearing behaves differently above and below 2 kHz, as discussed earlier. Below 2 kHz, phase differences between signals at the ears account for the direction finding, whereas above this frequency, the direction finding is based essentially on the time of arrival and intensity cues. The cocktail party effect allows us to separate sound sources in complex sound fields.

Simply coding the stereo as two single channels may result in unmasking of, for example, quantization noise since the perceived stereo sound field becomes more *transparent* to us. If two sounds sum up to zero because of phase differences, quantization noise may become audible, requiring an increase in resolution.

A different approach to stereo coding uses finding the direction of incidence of each sound component. One can then code the maximum direction signal together with a direction indication.

Audio Quality of Perceptual Coding/Decoding

The perceived transmission quality of any audio coder/decoder system is determined not only by the coder/decoder but also by the electroacoustic quality of the entire audio transmission chain, particularly the loudspeakers. These are the final link in the audio reproduction chain, and the better the loudspeakers, the more likely are listeners to hear whatever artifacts are introduced by the digital sampling and processing. Table 16.1 shows the specifications of the loudspeakers and listening room as required by the ITU-R BS.1116 recommendation for listening tests.

Typical unpleasant characteristics of codecs are quantization noise, pre-echo (spreading of quantization noise to before the arrival of a transient), aliasing (because of the side lobes of the filters), chirps (high-frequency spectral content suddenly appearing and disappearing), speech reverberation (where there is none), and multichannel artifacts (such as loss or shift of the phantom source direction).

Listening tests using trained listeners are an essential component in any development of digital audio systems, if perceptual coding is used. It is important to choose appropriate test routines, test audio, and test subjects. The statistical spread in any routine using human listeners as measurement instruments is likely to be higher than using physical instruments. Different psychometric methods will have different sensitivity to various faults and shortcomings. In addition, prolonged listening, over weeks and months, may show sensitivity to characteristics missed in the conventional laboratory listening test where the test subject is listening to audio files that are only 10 to 20 s long. Some problems in subjective testing are discussed in Chapter 17.

Table 16.1 The specifications of the loudspeakers and listening room as required by the ITU-R B5.1116 recommendation for listening tests. [Ref 17.20]

Parameter	Specifications
Reference loudspeaker monitors amplitude vs. frequency response	40 Hz to 16 kHz ± 2 dB (noise in 1/3 octave bands, free-field) ± 10° frontal axis ± 3 dB re 0° ± 30° frontal axis ± 4 dB re 0°
Reference loudspeaker monitors' directivity index	6 dB ≤ *DI* ≤ 12 dB 0.5 kHz to 10 kHz
Reference loudspeaker monitors nonlinear distortion at 90 dB SPL	< –30 dB (3%) for f < 250 Hz < –40 dB (1%) for $f \geq 250$ Hz
Reference monitors' time delay difference	< 100 μs between channels
Height and orientation of loudspeakers	1.10 m above floor reference axis at listener's ears, directed towards the listener
Loudspeaker configuration	Distance between loudspeakers 2 to 3 m Angle to loudspeakers 0°, ±30°, ± 110°, Distance from walls > 1 m
Room dimensions and proportions	20 to 60 m^2 area for mono/stereophonic reproduction 30 to 70 m^2 for multichannel reproduction 1.1 $w/h \leq l/h \leq (4.5\ w/h\text{-}4)$ $l/h < 3\ w/h < 3$ where: l is length, w is width, h is height
Room reverberation time	$T = 0.25(V/100)^{1/3}$ for 200 Hz ≤ f ≤ 4 kHz where: V = volume of the room The following limits apply: + 0.3 s @ 63 Hz to + 0.05 s @ 200 Hz ± 0.05 s @ 200 Hz to 4 kHz ± 0.1 s @ 4 kHz to 8 kHz
Room early reflections	< –10 dB for t ≤ 15 ms
Operational room response	≤ + 3 dB, – 7 dB @ 50 Hz to ≤ + 3 dB @ 250 Hz ≤ ± 3 dB, @ 250 Hz – 2 kHz ≤ ± 3 dB @ 2 kHz to ≤ + 3 dB, – 7.5 dB @ 16 kHz
Background noise (equipment & HVAC on)	≤ NR15 in the listening area ≤ NR10 in the listening area recommended



This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Audio Systems and Measurement

17

17.1 INTRODUCTION

In audio engineering, the term *high-fidelity* means the faithful reproduction of sounds of voice and classical nonelectronic musical instruments—sounds that are recorded by way of microphones and then played back using loudspeakers, headphones, earphones, or even shakers. Sounds of music and voice can also be generated electronically either by way of special synthesizers or by computer software. Such sounds can have waveforms and frequencies beyond those common to conventional musical instruments.

The properties of hearing were discussed in Chapter 3. When we listen to sounds, hearing converts the acoustic pressure history in the sound field that surrounds us to generate the percept of sound in our brain. Audio systems may be designed for reproduction sounds for maximum intelligibility (with or without noise), for maximum enjoyment or subjectively perceived quality, or for maximum similarity to what would have been heard if the listener had been present at the recording.

Four commonly accepted conditions for high-fidelity sound reproduction are (see Reference 17.1):

1. Full frequency range reproduction with negligible level and time delay differences between various parts of the audio spectrum
2. Noise and distortion free sound reproduction at all sound pressure levels
3. Reverberation characteristics of the original sound field must be retained
4. Spatial properties of the original sound field must be retained

In practice, the complete fulfillment of these conditions remains an unattainable goal. Engineering compromises are always necessary, and the listener must accept that, without the cues offered by vision and other modalities, the experience will, in any case, not be that of attending the live event. In many cases there is no "original" sound event or sound field, and for artistic reasons the recorded sound may have been processed in various ways. This chapter will give an introduction to some common sound recording and playback techniques, an introduction to physical metrics for audio quality, and finally a short discussion of measurement techniques using humans as measurement tools.

17.2 AUDIO CHANNELS AND LISTENING MODES

Telephones use a single channel pickup and sound reproduction system. Most other sound carriers (except for telephones), however, are designed to record and reproduce audio in formats using more than one channel. For stereo, at least two channels are usually used for recording and playback to render the spatial properties of sound in a better way. The audio contained in the two channels is usually reproduced by way of two headphones or loudspeakers but may be recorded using many microphones or pickup devices. Other carriers, such as radio and TV transmissions, CDs, DVDs, Internet radio (and other streaming audio), and others may be configured from one to several signal channels, two and five channels being the most popular. Because of the many ways that audio signals can be recorded, transferred, stored, and reproduced, it is important to understand how audio systems are characterized and how they differ.

17. 3 MONOPHONIC AND MONAURAL

The word ending *-phonic* is used to describe the number of channels that are used in the sound carrier. Old gramophone records and most AM broadcasts that are single channel carriers are said to be *monophonic*. In spite of telephone systems being essentially monophonic, they are seldom referred to in that way. Many people use headsets to communicate using telephone systems and the headsets usually use one transducer for each ear. Presenting the same channel to both ears is called dichotic presentation.

The number of ears used in listening is referred to using the word ending *-aural*. We can listen monaurally or binaurally (using one or two ears). The principles of mono- and stereophonic audio systems are shown in Figures 17.1 and 17.2.

17.4 STEREOPHONY AND PHANTOM SOURCES

Vinyl records, albums, CDs, DVDs, and Internet radio are generally transmitting audio using two carrier channels and are called *stereophonic*, see Figure 17.1b. In stereophonic sound reproduction, the listener and the two loudspeakers are assumed to be at the corners of triangle as indicated in Figure 17.1b.

Typically, the loudspeakers need to be close so that the triangle will have a smaller apex angle ϕ at the listener—40° to 60°. The line between the loudspeakers is called the baseline. If the recording is correctly done, the baseline will be filled with appropriate *phantom sources*. Listening off-center, in principle, results in the phantom sources tending to collect in the loudspeaker closest to the listener. By using loudspeakers that have a controlled directivity pattern, it is possible to have phantom sources between the loudspeakers also for off-center listening positions. Of course the presence of room generated early reflections and reverberation will affect the perception of the phantom sources. Using crosstalk cancellation techniques discussed later in this chapter one can also have phantom sources outside the baseline.

Figures 5.9 and 5.10 show how the perceived angle of sound incidence depends on the relative time difference. Because recordings can use microphones with different directivity patterns and because of the way microphones may be positioned and directed relative to the sound source, different time-level patterns will occur in the reproduced audio. Figure 17.3 shows the effects of combinations of such recording induced time and level differences.

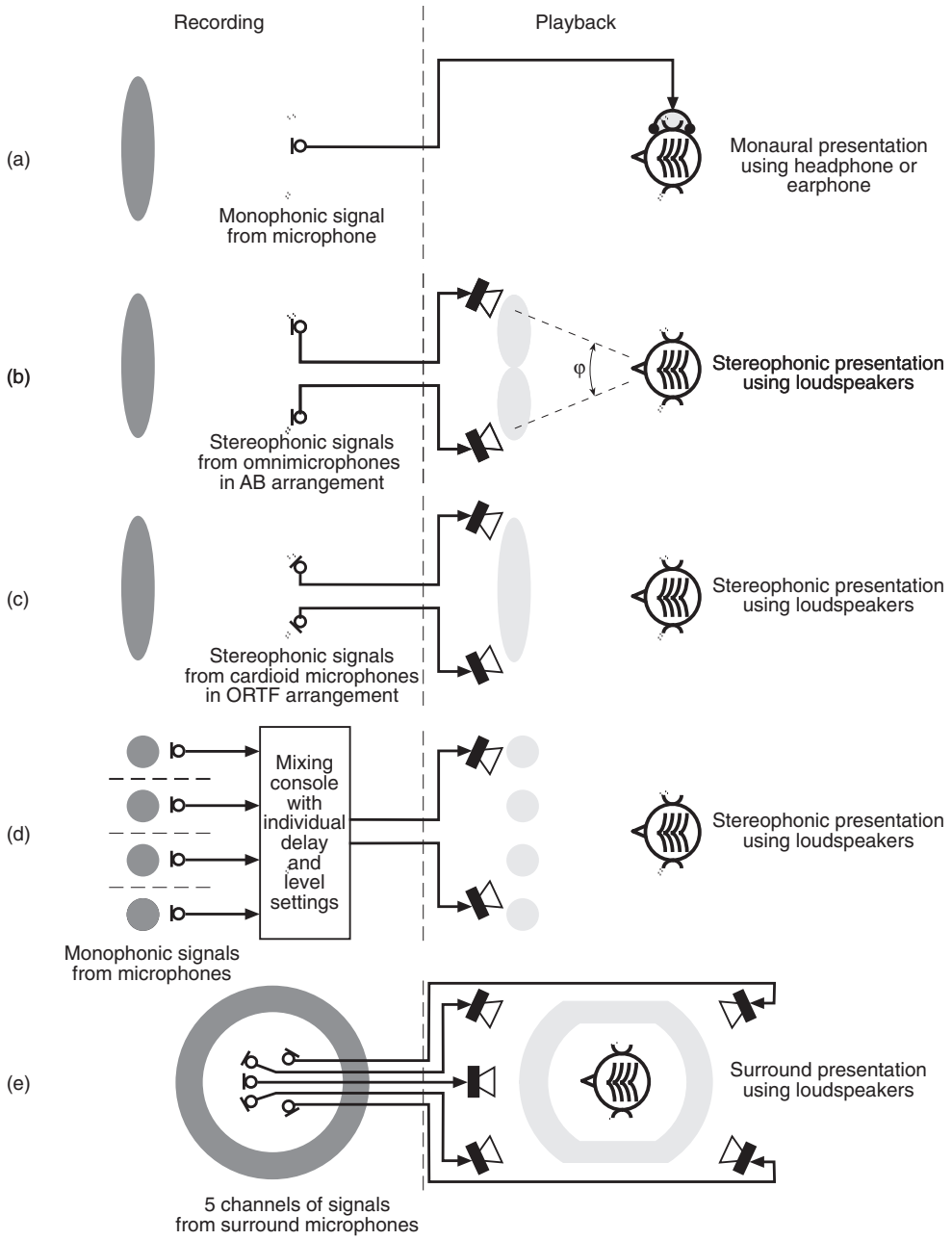


Figure 17.1 Some sound recording/playback systems: (a) monaural using a monophonic source, (b) stereophonic using AB microphones, (c) stereophonic using a ORTF microphone arrangement, (d) stereophonic using a multimicrophone technique, and (e) surround sound recording & playback. Many more are commonly used. Amplifiers, filters, delays, and other signal processing equipment are not shown.

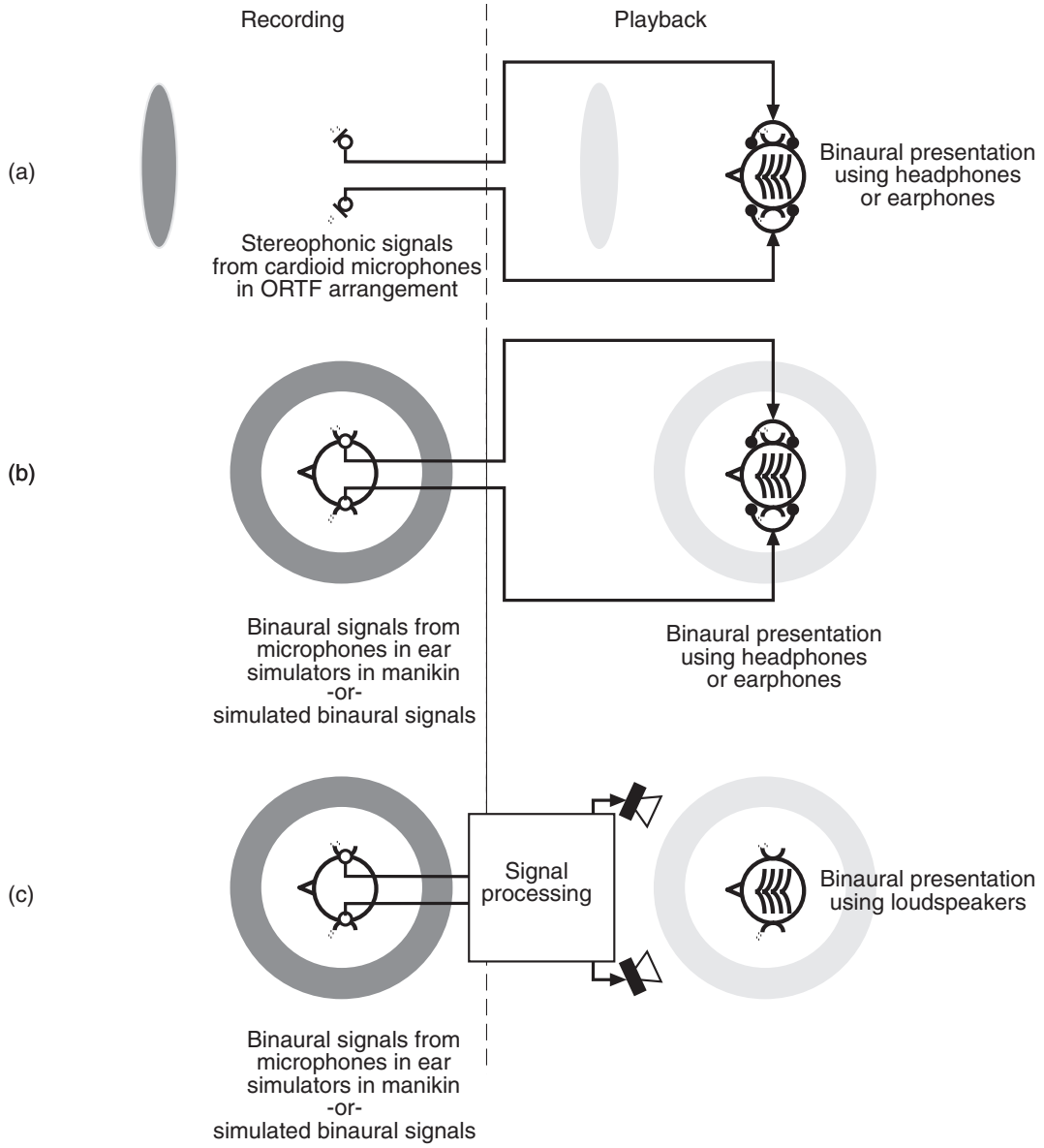


Figure 17.2 Some sound recording/playback systems for binaural playback using: (a) spaced cardioid microphones, (b) a recording manikin, and (c) cross-talk cancellation and other signal processing (see Figure 17.7).

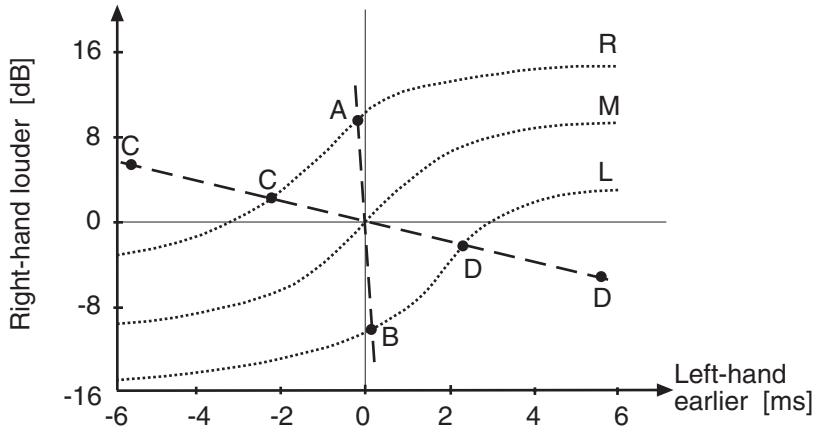


Figure 17.3 Lateral position of the phantom source in the case of mono sound source reproduced by two loudspeakers at $\varphi = \pm 30°$ with simultaneous variation of delay and level. Dashed lines explained in text. (L = phantom source in left-hand loudspeaker; M = phantom source in the middle between loudspeakers; R = phantom source in right-hand loudspeaker. (After Ref 5.6)

Figure 17.4 shows some popular ways of arranging the microphones for stereo and binaural sound recording. Originally, AB recording was used to record stereo sound. In AB recording, two spaced omnidirectional microphones provide the sound signals that are transferred to the stereo loudspeakers. In AB recordings, there will be both time and level differences due to the way music signals reach the microphone. Sometimes support microphones are used to provide more clarity for certain instruments or instrument groups.

Later, XY technology was developed in which two-coincident cardioid or bidirectional microphones pick up the stereo sound signals. In MS stereophony, coincident cardioid and bidirectional microphones are used, and their signals added and subtracted to obtain signals similar to those obtained by XY stereophony. In both XY and MS recording, there are mainly level differences between the stereo signals; these methods are called intensity stereophony.

Characteristic for all multichannel playback methods is the creation of phantom sources in listening. The curves marked R, H, and L in Figure 17.3 were obtained in listening tests and mark combinations that result in phantom source placement at the right hand loudspeaker, between the loudspeakers, and at the left hand loudspeaker respectively. Figure 17.3 shows two lines that indicate how AB and XY microphone techniques differ in stereo rendering capability. Line AB shows the typical behavior of MS recording technique signals, whereas line C′ C D D′ shows rendering using poor AB microphone placement. In the latter case, most of the phantom sources will be located in either loudspeaker. The effect is sometimes called *hole in the middle*, no sound sources are convincingly perceived as being in between the loudspeakers. The ORTF technique shown in Figure 17.4d has good loudspeaker/headphone compatibility.

The perceived depth localization is mainly due to level and spectral differences in the recorded reverberant and direct sound.

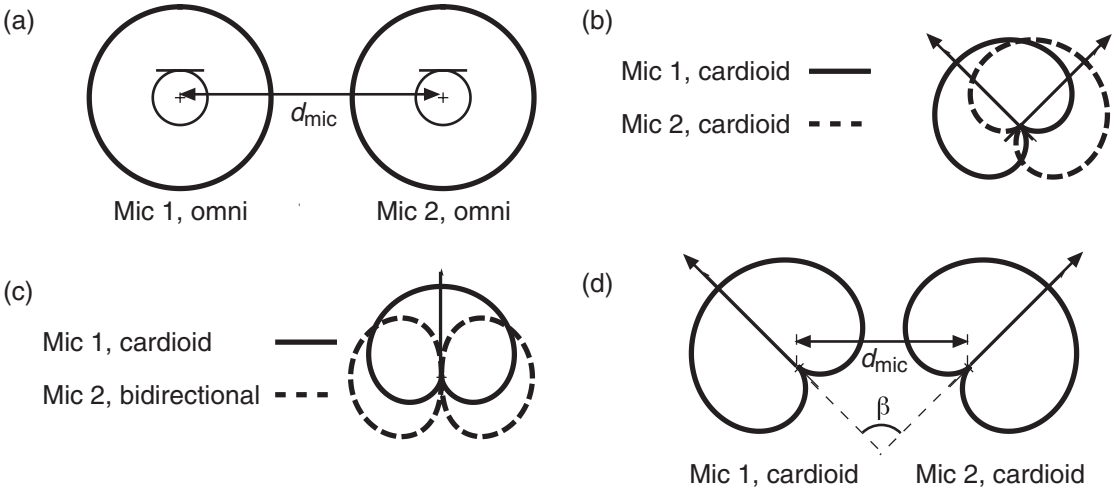


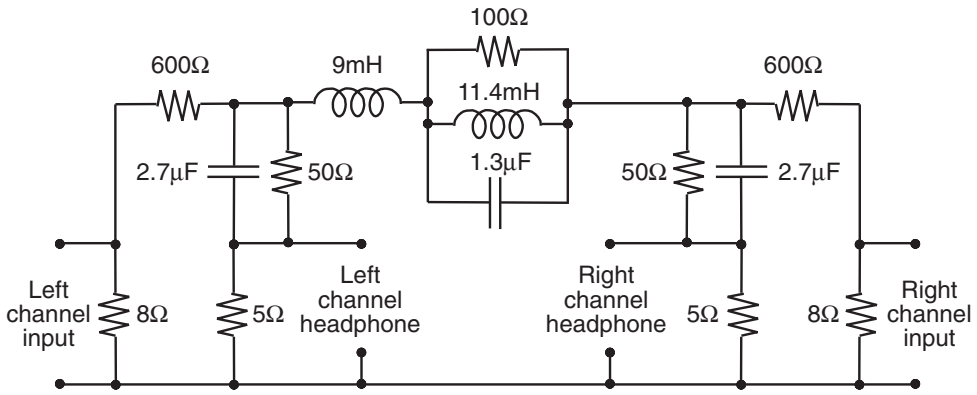
Figure 17.4 Some microphone arrangements for stereophonic recording of sound. Microphones 1 and 2 feed left- and righthand channels unless otherwise noted: (a) AB (time delay-based) the distance d_{mic} between the microphones is usually quite large, (b) XY (level-based), (c) MS (level-based) note that the microphone signals must be summed and subtracted to give left- and right-hand signals, and (d) ORTF microphone pair where the distance d_{mic} and the angle β between the two microphones are adjusted for optimum stereo sound reproduction with minimal sound stage distortion. Often d ≈ 0.2 m and β ≈ 110°.

In multimicrophone recordings, one can position the sound source more freely; there are now commercially available sound mixing desks that allow direct positioning of the phantom sources anywhere around the listener. These effects are often used in surround sound reproduction and to some extent, also in film sound. By adding simulated reverberation and early reflections, the phantom sources can be positioned inside a virtual space as well. This auralization-based technique is often used for recreating reverberation and *ambience*. The reverberation is usually generated by digital signal processing using either *finite* or *infinite impulse response filters* that can simulate the acoustic properties of real and imaginary environments.

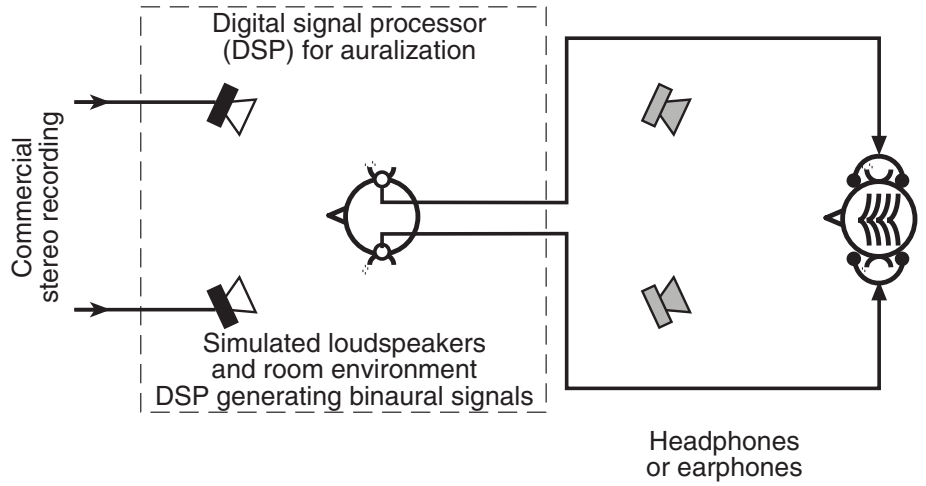
17.5 STEREO USING HEADPHONES

Many listen to stereo recordings using headphones rather than loudspeakers. Since the two-channel recording does not contain the correct binaural cues (time and frequency response differences between the ear signals), at least two phenomena occur. One is that the sound recorded to appear mainly toward the right and left loudspeakers now appears in each headphone and is not externalized; the other is that the sound recorded to appear between the loudspeakers now appears inside the head, so called *in-head localization*. Several methods have been proposed to eliminate these problems. Commonly, binaural cues (time and level differences) are introduced to try to fool hearing that the sounds are actually emanating far from the head.

Figure 17.5a shows a hardware solution where the stereo loudspeaker signals are fed to an electronic network that introduces approximately correct cross-talk signals mimicking those that would be obtained in free-field listening, so that the stereo recording is better suited for headphone listening. A problem



(a)



(b)

Figure 17.5 By introducing appropriate transfer functions and cross feed of signals one can simulate stereo and multi-channel listening using headphones. Analog (a) or digital (b) signal processing techniques may be used. The circuit in (a) was proposed by B. B. Bauer. Using digital techniques as in (b) the room environment for stereo listening may be simulated as well.

with this approach, leading to an unsatisfactory sound, is that people listen to loudspeakers in reverberant rooms, not in anechoic chambers. The audio will not sound natural.

A software solution now available is using real-time filtering to introduce also the room cues into the audio signals as shown in Figure 17.5b. This is an application of auralization, as discussed in Chapter 6. Such systems are in common use in headphone sound reproduction systems used for aircraft passengers.

Since humans never hold their heads completely still, real-life signals will continuously vary in amplitude and time. Because the two methods described do not introduce these minute cues, they usually fail in achieving convincing natural sounding frontal and out-of-head localization in headphone listening.

Loudspeakers generally have inconsistent directional sound radiation characteristics, as discussed in Chapter 14, so clearly angling loudspeakers toward the listener differently will result in different frequency response at the listener. In addition, the listener's sensitivity to sound varies with the angle of incidence of the sound. The head-related transfer functions described in Chapter 3 show that these differences can be considerable. To some extent, the frequency response differences induced by the head-related transfer functions can be compensated by the frequency response of the loudspeakers.

In most real listening situations, there will be early reflections and reverberation that will tend to average out the differences in head-related transfer functions and loudspeaker frequency responses. The precedence effect will determine the localization of the phantom sources, but the room's sound reflections will affect the timbre of the sound.

17.6 BINAURAL SOUND REPRODUCTION

Binaural sound reproduction eliminates many of these effects. The importance of binaural hearing for our impression of the spatial properties of the sound field was pointed out in Chapter 3. The use of microphones in spheres and manikins are similar approaches to sensing the sound field the way a human does. In the best binaural sound recordings, an anthropomorphic manikin with a microphone in each ear, such as the two shown in Figure 17.6, is used to record the two signals to be reproduced. Hair has virtually no influence on the signal characteristics and is usually not used on recording manikins other than to make them less conspicuous in some recording situations.

Headphones or earphones that are correctly frequency response equalized are essential for good listening results in term of phantom source placement. Typically, such equalization can only be achieved using digital signal processing.

It is important to note that binaural recordings are difficult to use for sound level measurements. The microphones no longer sense the sound pressure of the sound field without the manikin but rather sound that has been amplified by the reflections of the head and torso. To some extent, these effects can be removed by equalization if the sound field is diffuse.

Because of the problems in equalizing sound signals that have been picked up at the concha or at the eardrum positions in a manikin, some prefer to use a hard sphere with diametrically positioned pressure sensing microphones at the surface of the sphere. Note that the reflection of the sound field at the surface of the sphere causes frequency response irregularities in any case.

Initially, binaural recording may seem straightforward and problem free. However, we are used to listening to our own ears, head, and torso and the specific time/frequency response patterns that those create in the cochlea. When using someone else's ears, these patterns are no longer the ones we are used to, and the binaural illusion falls apart, typically positioning frontal sound sources to the back or in the worst case causing in-head localization of the phantom sources. The absence of the minute response cues due to head movement discussed earlier contributes seriously to these effects. In experiments where the listener's head movements have been transferred to the manikin's head using a servo system,

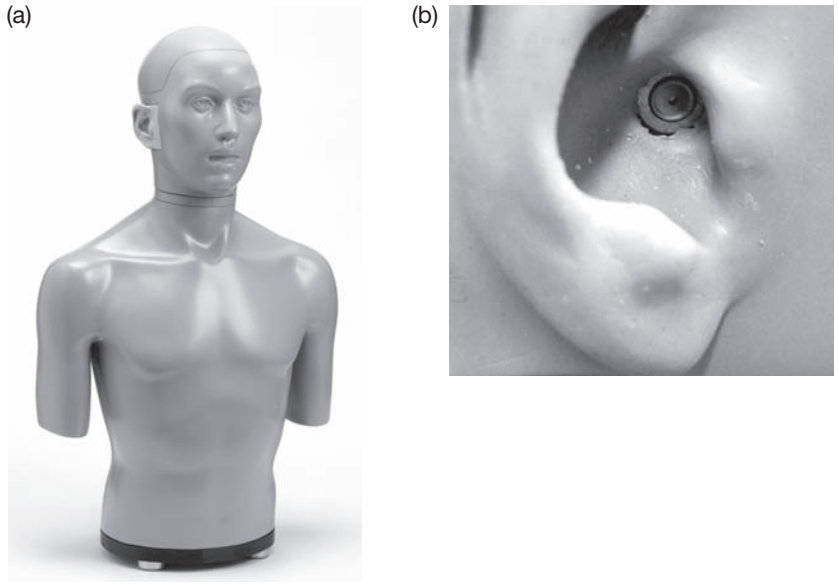


Figure 17.6 (a) The anthropometric head and torso simulator KEMAR manikin for binaural sound recording and measurement is based on worldwide average human male and female head and torso dimensions (Photo courtesy of G.R.A.S., Denmark). (b) Microphone mounted at the simulated ear canal entrance. (Photo by Mendel Kleiner.) (Compare to Fig 5.22)

it has been shown that the presence of these movement cues can largely restore the correct placement of phantom sources in front of the listener.

By using loudspeakers to reproduce the binaural signals, sometimes called *transaural sound reproduction* or *cross-talk compensation* (XTC), one can reduce the problems mentioned in the previous paragraph. Because the loudspeaker signals are affected by the listener's head-related transfer function, they need to be filtered (equalized) to remove this influence. The filtering also needs to include cross-talk cancellation to remove the left and right signals reaching the right and left ears respectively. The principle of cross-talk cancellation for binaural systems is shown in Figure 17.7.

17.7 LOUDSPEAKER-HEADPHONE COMPATIBILITY

Another disadvantage of binaural recording is its lack of compatibility with conventional stereo listening using loudspeakers. It is important for radio, television, and film that recordings sound good when listening using headphones or loudspeakers.

The ORTF recording technique addresses this goal by using a microphone arrangement according to Figure 17.4e. In the ORTF approach, two cardioid microphones are used. These are placed about

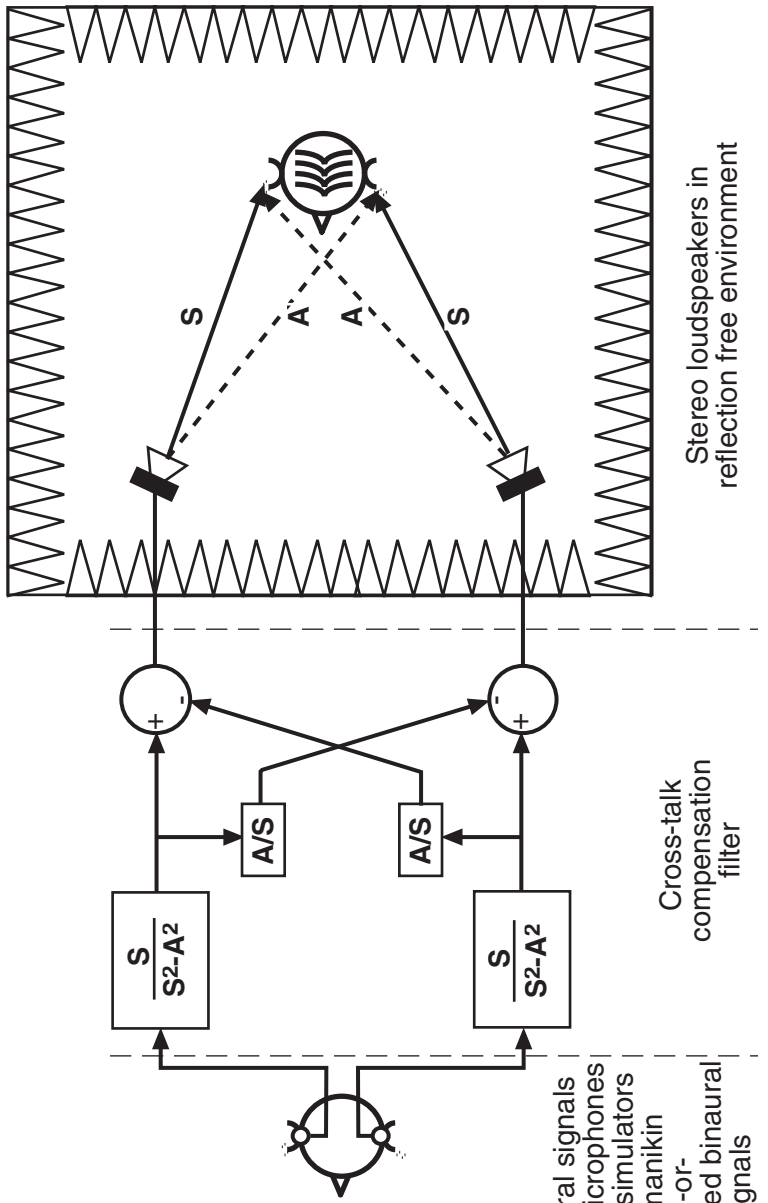


Figure 17.7 The principle of cross-talk compensation used for listening to binaural sound recordings using a loudspeaker pair for stereo. Note that off-center listening and lateral sound reflections from room surfaces seriously degrade the listening experience.

17 cm apart and angled away from one another so that their pickup angle covers the distribution of musical instruments. Figure 5.3 shows that maximum sensitivity of hearing to reflections is for sound that arrives at an angle of 45° to the median plane. If the ORTF pair microphones have their sensitivity lobes at right angles, their sound pickup will approximate this sensitivity characteristic.

The cardioid directivity pattern ensures effective loudspeaker function and the angling and intermicrophone distance ensures good binaural properties. The distance between the microphones results in approximately the same time delays as those that appear at human ears for different angles of incidence relative to the median plane.

17.8 MULTICHANNEL SOUND REPRODUCTION

By introducing more sound reproduction channels, one can obtain a more ideal sound reproduction involving 1) better phantom source localization, 2) phantom sources out of the horizontal plane, and 3) phantom sources that remain in place even when moving around in the sound field. Extra support microphones may be used as well as mentioned earlier.

So-called *5.1 surround sound* recording addresses the need for better phantom source localization. Surround recordings are intended for loudspeakers placed in the horizontal plane around the listener as shown in Figure 17.8 and aim at reproducing both low-frequency sound effects (the .1 channel), better front phantom source localization (the three front channels), and better effects and reverberation (the two rear channels).

Ambisonics is a surround sound recording technique that uses a tetrahedron of cardioid microphones (sound field microphone), such as the one shown in Figure 17.9, to pick up sound in four primary directions all separated by a 120° angle.

The tetrahedron microphone technique addresses the need to achieve phantom sources out of the horizontal plane and scalability. The signals picked up by the cardioid microphones are converted to four signals that represent those that would have been obtained if one had used one omnidirectional and three orthogonal bidirectional (cosine pattern) microphones. These four signals can then again be rematrixed to represent those that would be obtained for any other cardioid microphone direction. In this way, loudspeakers can be placed around the listener, and their position measured and entered into the software matrix so that there is a virtual microphone in each of these directions.

Wave field synthesis (WFS) is an approach to sound-field generation putting Huygen's principle to use and addresses the need for phantom sources that remain in place even when moving around in the sound field. Usually used for virtual sound sources, the signals are processed so that they appear at the loudspeakers with the time and amplitude that would have been obtained for a real sound field at the corresponding position. The distance between the loudspeakers should be smaller than one-half of the highest wavelength to be reproduced to avoid serious lattice effects, but this is currently not practical. Ideally, of course, a surface array of loudspeakers should be used. Since it is fairly easy to fool hearing into believing that there are out-of-plane signals, this is not a large limitation. A typical WFS array uses about 250 loudspeakers surrounding the listener in a plane. Figure 17.10 shows an example of a large experimental WFS loudspeaker array. An experimental cinema installation has shown that the principle can be used successfully for cinema sound reproduction.

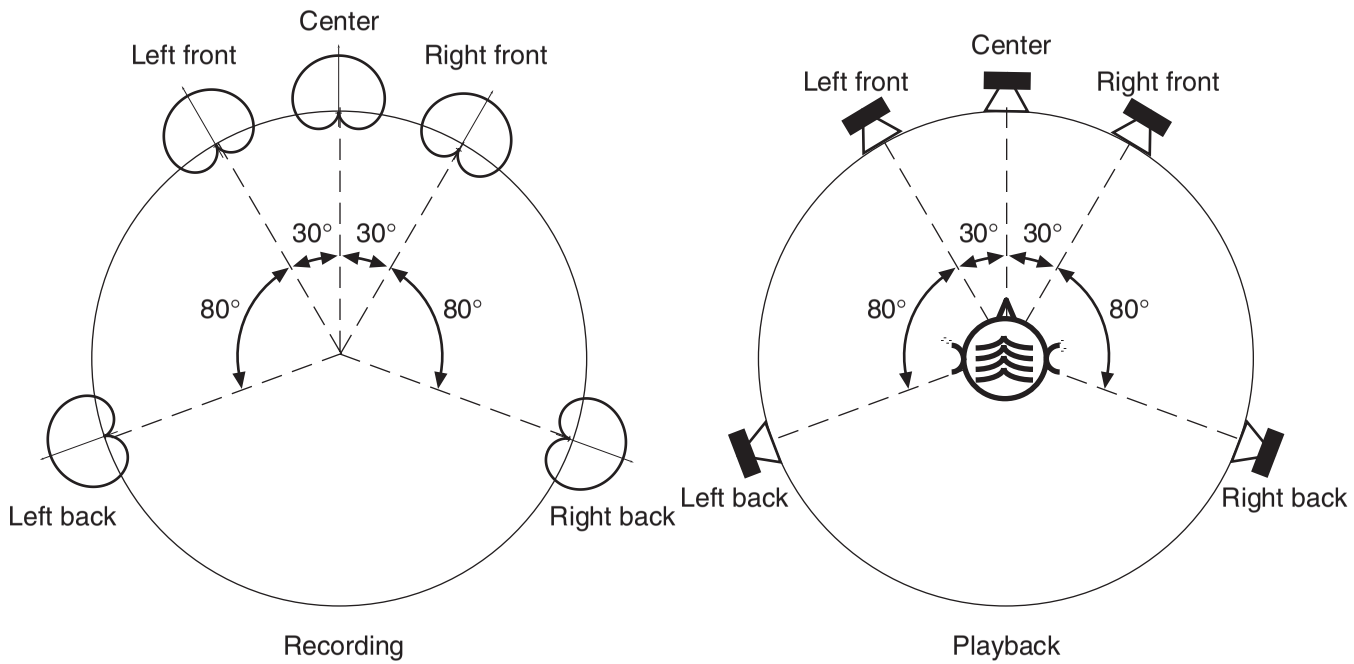


Figure 17.8 The principle of surround sound recording playback. There are many suggested recording setups, the one shown here is similar to the Wide Cardioid Surround Array suggested by DPA Microphones, Denmark. Five cardioid microphones directed radially are used at the various locations on a circle having a radius of 0.60 to 1 m. The array configuration shown here tends to give a wider "sweet spot" in listening than does the coincident array of cardioid microphones. For playback according to the ITU-R BS.775 recommendation five identical full frequency range loudspeakers are used. The positioning of the reference listening point is at the center of an imaginary circle having a radius between 2 m and 4 m (minimum and maximum radii defined by the ITU-R BS.1116–1 recommendation). Loudspeakers should be at a height of the listener's ears. A low-frequency effects channel and loudspeaker is usually added, hence the "5.1" designation.



Figure 17.9 A microphone using four cardioid microphones in a tetrahedron arrangement for Ambisonics recording. (Photo by Mendel Kleiner.)



Figure 17.10 An experimental rectangular array for WFS sound reproduction using 256 loudspeakers (photo by Mendel Kleiner.

17.9 METRICS FOR AUDIO CHARACTERISTICS

Audio and acoustic measurements are used to describe sound, typically with numerical values, expressed in metrics, determined using ad-hoc or predefined methods. Measuring audio quality is similar to the measurement of room acoustic quality in that it is necessary to measure transmission system properties that can truly only be described by listeners. Because of the complexity of audio signals and systems, one usually finds a large variation in the way listeners describe the system properties.

An audio transmission system starts in the recording venue, a concert hall, a stage outdoors, or recording studio, for example. At the recording venue, the venue's background noise, its acoustic properties, the microphones and their placement and directional characteristics, as well as the recording system (analog-to-digital [A/D] converters, sample word length, sampling frequency, codec, etc.), all influence the information that is uploaded to be stored or transmitted to the listener.

The system is reversed at the receiving end of the transmission system. Here the signals are decoded, amplified, reradiated by loudspeakers, and subject to the acoustics of the listening room. The frequency-dependent directivity of the loudspeakers and the sound absorption and scattering properties of the room will make each listening system have its own individual characteristics that are difficult to replicate and that will, in any case, be quite different from those of the recording venue.

One way of regarding the situation is to say that the best that can be expected is that the sound field at the listener is that assumed and strived for by the recording engineer or Tonmeister. A different way is to say that the best that can be expected is that the listener's audio expectations are fulfilled. In either case, one can say that the system has high audio quality. In practice, it is the latter view of what constitutes high audio quality that is the common one.

17.10 PHYSICAL AUDIO METRICS

The audio sound character descriptors can, to some limited extent, be mapped onto a set of audio metrics such as noise level, loudness, amplitude frequency response (usually called just frequency response), group delay, nonlinear distortion, modulation damping, directivity, and interaural cross-correlation, for example. Electronic devices such as amplifiers will require a set of metrics somewhat different from those suitable for characterization of electroacoustic devices such as loudspeakers. Note that the various audio sound character ratings are interdependent, so it is not possible to achieve a 1:1 mapping, of distortion onto roughness, for example. No audio metrics exist to characterize the spatial agreement between recording and playback sound.

Derived metrics, such as dynamic range, are also important in many practical situations. Dynamic range is usually taken to be the level difference between overload and background noise, so it will depend on the criteria used for determination of noise level and nonlinear distortion. The dynamic range will be different in different parts of the frequency spectrum.

Since audio metrics have been of great interest for a long time, many metrics have been introduced along with those discussed. Audio equipment manufacturers typically need to measure many more properties of their products as part of design, manufacture, and quality control. Many standards are available for example those issued by the International Electrical Commission (IEC) and the Audio Engineering Society (AES). The AES also issues *AES Recommended Practice* documents. Note that many metrics need the equipment to be properly calibrated to a reference. The ITU-R BS.1116 recommendation is often used to specify desirable playback conditions. An abbreviated version showing the main points can be found in Table 16.1. Other relevant recommendations are ITU-R BS.1283, 1284, 1285 and 1286 (see Reference 17.21).

17.11 NOISE LEVEL AND SIGNAL-TO-NOISE RATIO

Various metrics for noise were discussed in Chapters 2 and 6. Metrics such as octave and third octave band levels, as well as A-, B-, and C-curve frequency-weighted levels, were introduced in Chapter 2 and noise criteria (NC) usually used in room acoustics, such as NC curves, in Chapter 6. Additionally, there are many other weighting curves recommended by various standardization bodies.

It is important to remember that many types of noise have a recognizable character (for example, *popcorn* noise) and such noise should be rated differently from thermal (Gaussian) noise, (the previously mentioned weighting curves are intended for the evaluation of thermal noise). It is easy to convince oneself of the importance of noise *intelligibility* by listening to a time-inverted music sound file.

17.12 AMPLITUDE FREQUENCY RESPONSE

The transfer function H of a system is the complex ratio of the spectrum of the output signal S_0 divided by the spectrum of the input signal S_i:

$$\underline{H}(f) = \frac{\underline{S}_0(f)}{\underline{S}_i(f)} \tag{17.1}$$

The level of the transfer function L_H can be written:

$$L_H(f) = 20\log\left(\left|\underline{H}(f)\right|\right) \tag{17.2}$$

In audio engineering, the amplitude frequency response is usually taken relative to its level at 1 kHz. Other reference frequencies may be chosen, for example, when describing the frequency response of filters, low- and high-frequency loudspeakers, and other devices that may not have or be intended to have any useful response at 1 kHz. In such cases, the response reference frequency is typically taken as approximately the geometric mean of the low and high pass frequencies f_L and f_U where the transfer function level is –3 dB down relative to the level at the reference frequency f_{ref}. Some of the terms used in characterizing frequency response curves are shown in Figure 17.11.

$$f_{\text{ref}} \approx \sqrt{f_L f_U} \tag{17.3}$$

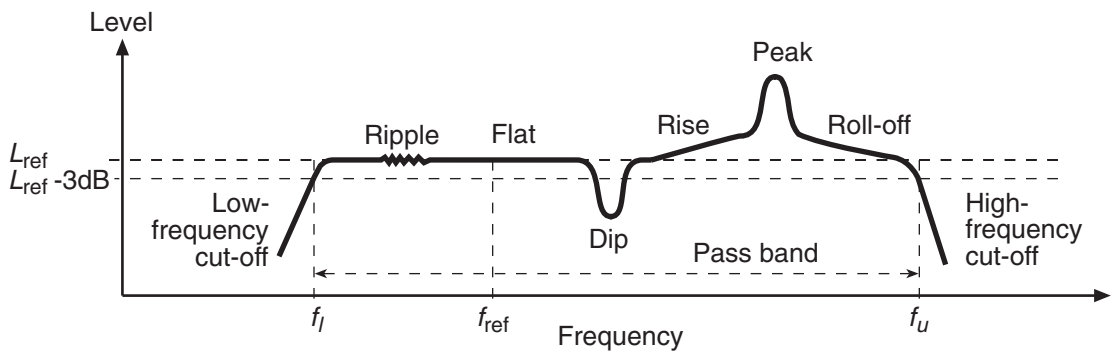


Figure 17.11 Various characteristics of the FR.

The amplitude frequency response function *FR* (usually just called the *frequency response* (FR), or simply the response) can be written:

$$FR(f) = \frac{L_H(f)}{L_H(1000)} \tag{17.4}$$

The *band limits* are usually taken as the frequencies where the response is –3 dB down relative to the level at some reference frequency, often 1 kHz. Small variations in the FR within the passband are usually called *ripple*. In most cases, one will want amplifiers, loudspeakers or microphones to have a FR that is as flat as possible—that have a minimum of ripple in the pass band. One also usually wants the FR to extend far beyond the pass band with slow decay so that there are only small deviations in phase in the transfer function. This is discussed further in the following section.

Strictly, the measurement of the FR should be done one frequency at a time, but a comprehensive measurement using this technique requires considerable time, so two other techniques are usually used. Using analog or digital techniques, the response can be measured using a sweep sine-wave generator set for slow sweep speed and a level recorder. In these measurements, the sweep speed (frequency increment per time unit, Hz/s) usually is set to increase proportionally to the signal frequency. Typically, the time necessary for a sweep measurement depends on the accuracy by which one wants to measure peaks or dips in the response. Assuming the resonant system to have a quality factor *Q* (or the filter used for the analysis) and that the resonance frequency is estimated as f_0, one can write the maximum sweep speed:

$$S_{\max} < 0.9\left(\frac{f_0}{Q}\right)^2 \qquad \text{Hz/s} \tag{17.5}$$

for linear sweeps (constant sweep rate in Hz per second). For logarithmic sweep rates (see Reference 17.1):

$$S_{\max} < 1.3\frac{f_0}{Q^2} \qquad \text{Octaves/s} \tag{17.6}$$

It is common to see commercial equipment measured at much too high sweep rates that make the equipment seem to have a *flatter* FR than it really has.

Fast sweep sine-wave signals are usually called *chirp* signals because of their birdsong-like sonic character. Note that such signals are unsuitable for analog measurement, but may be used successfully with digital fast Fourier transform (FFT) based measurement.

For digital frequency analysis using the FFT, similar relationships hold. The FFT window length must be large enough for any transient to decay to some sufficiently small value within the analysis time. The window function should be chosen appropriately for the signal type. Further, the sampling frequency must be high enough for aliasing not to occur, and the number of samples within the time window must also be large enough so that peaks (or dips) in the frequency spectrum can be adequately resolved. This typically requires at least five spectrum lines within one-half power bandwidth of a resonance frequency peak. Dips caused by interference may be difficult to resolve since they are often very sharp.

17.13 PHASE RESPONSE AND GROUP DELAY

Any audio system will be characterized by unavoidable time delays such as the delay of sound due to its travel from instruments to microphones and from loudspeakers to listeners. Characteristic for these time delays (for point sources and receivers) is that the delays are frequency-independent. Only at large distances (such as out-of-doors) will the influence of humidity on the velocity of sound be of interest.

Because of the resonant characteristics of audio transducers that cause frequency roll-off at spectrum extremes and peaks and dips in the FR, there will be additional delays because of the time it takes for energy to be stored and released by the resonant systems, no matter whether these are electrical, mechanical, or acoustical. Many physical filters are characterized by a minimum-phase response. For minimum-phase filter systems, the phase response is linked to the magnitude of $\underline{H}(f)$ via the Hilbert transform.

The rate of phase shift in the transfer function (at some frequency) as a function of frequency can be shown to be coupled to the group delay of the signal. The group delay is:

$$\tau_g = -\frac{1}{360}\frac{d\varphi}{df} \tag{17.7}$$

where $d\varphi$ is given in degrees and df in Hz.

The group delay is a function of frequency and is of considerable interest in the design of loudspeaker systems. The more resonant the system, the larger will be its group delay deviations from the purely geometrically caused delay because of the travel time for sound. The use of loudspeaker system crossover filters results in large group delays that need to be controlled between loudspeakers in a stereo or multichannel audio playback system.

A system having *constant group delay* has *linear phase*; that is, the phase of the transfer function increases proportionally to frequency. This preserves the shape of the wave. Mathematically, it means that all frequency components are time-synchronized exactly as they were in the original signal.

In practice, group delay is best measured using digital techniques—for example, using maximum-length sequence analyzers, such as MLSSA, or by FFT-analyzers using chirp signals. In these systems, the phase response is unwrapped and the phase versus frequency slope calculated to find the group delay.

17.14 NONLINEAR DISTORTIONS

One characteristic of nonlinear signal distortion is that not only the shape of the signal is changed but also that energy is shifted away from the excitation frequency to another frequency. Some types of nonlinear distortion characteristics are shown in Figures 17.12 and 17.13.

Figure 17.12 shows some important distortion generation mechanisms: hard clipping, soft clipping, hysteresis, frequency/amplitude modulation, and slew rate limiting. All of these distortion generating mechanisms may be present both in electronic and electroacoustic audio equipment and are typical for electrodynamic loudspeaker drivers. Some of the mechanisms will be acting the same way on both steady-state and transient signals; the action of others, such as hysteresis, may need to be analyzed separately for the two signal types. The distortion effects of resolution limiting in digital systems was discussed in Chapter 16.

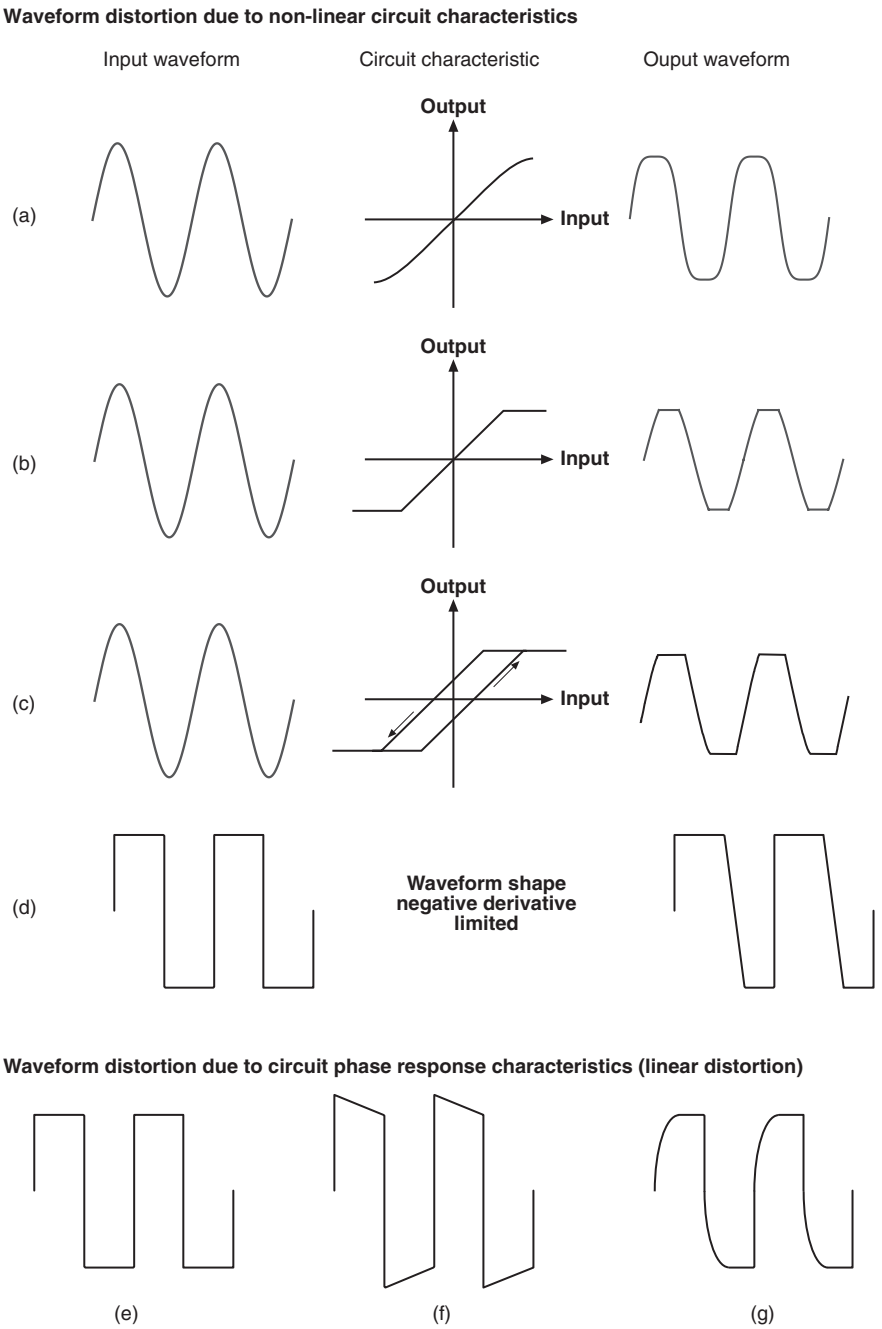


Figure 17.12 Waveform distortion characteristics for sine waves by: (a) soft clipping, (b) hard clipping (limiting), and (c) hysteresis. For square waves due to (d) slew rate limiting. Original square wave shown left (e). Square wave distortion shown for (f) phase distortion at low frequencies, and (g) *gentle* low-pass filter at high frequencies. Many other waveform distortions exist.

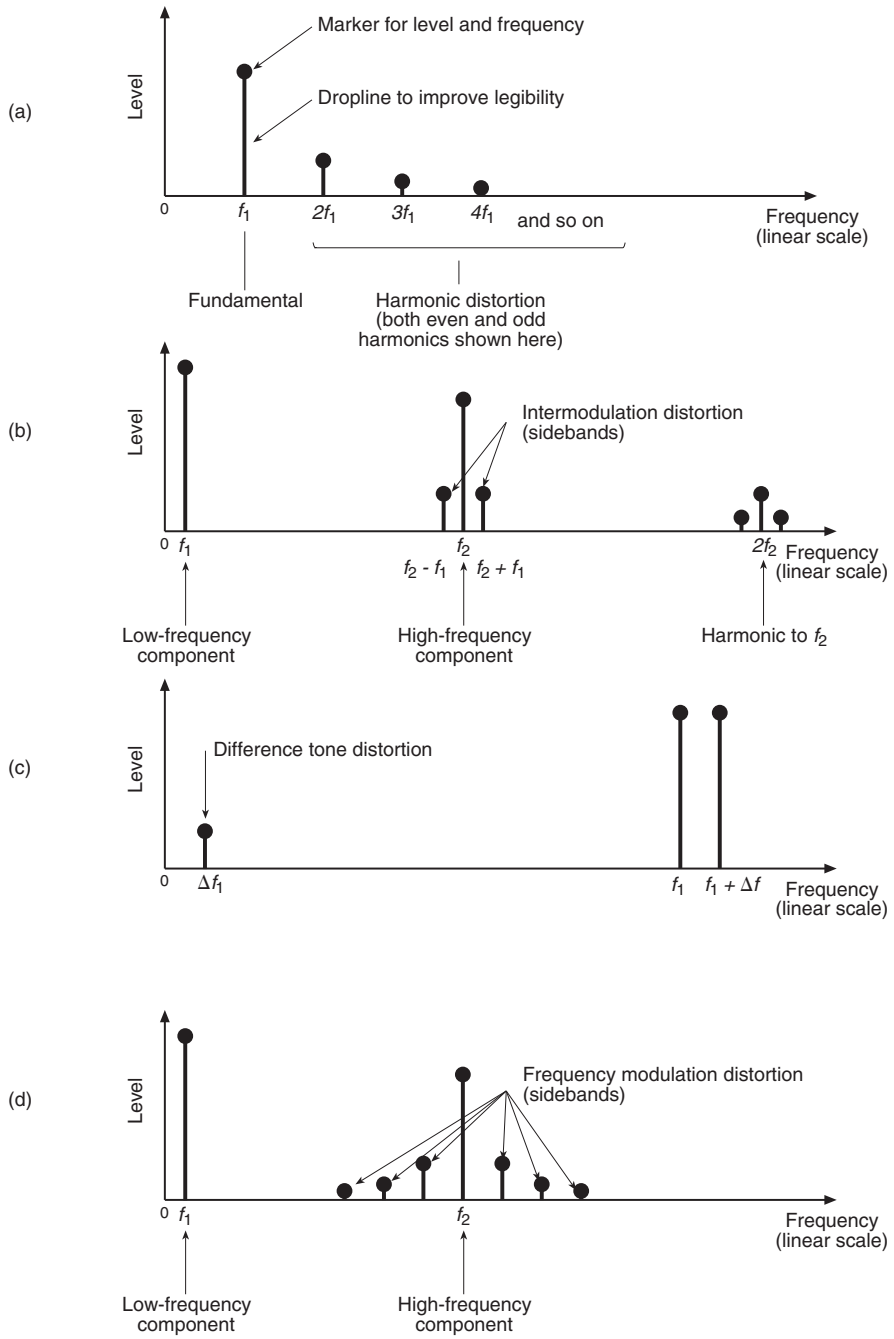


Figure 17.13 Examples of new spectral components generated by nonlinear distortion: (a) harmonic distortion (both odd and even harmonics shown in this example), (b) IMD, and (c) difference tone distortion, and (d) frequency modulation distortion.

Ideally, one should differentiate between distortions in the time and frequency domains. Most metrics for distortion apply to frequency domain measurements. Strictly, one could argue that the changes in waveform of transient signals due to, for example, highpass and lowpass filtering, are distortion. Moderate changes (that do not incur major oscillation in the time signals) due to FR roll-off beyond the audio frequency limits at 20 Hz and 20 kHz are usually not considered distortion.

17.15 HARMONIC DISTORTION

A system that causes a change in the waveform of an ideal sinusoidal signal generates distortion in the form of overtones (harmonics) and undertones (subharmonics). Subharmonics are generated by systems that feature nonsymmetric distortion generation—the positive and negative parts of the signal are distorted differently. Typically, only the harmonics of the signal are measured as a metric of nonlinearities in audio systems.

The *total harmonic distortion* (THD) in percentage is usually defined:

$$THD = 100\frac{\sqrt{\sum_{2}^{n}\tilde{e}_n^2}}{\tilde{e}} \tag{17.8}$$

Here $\tilde{e}$ is the rms voltage of the signal with distortion, and $\tilde{e}_n$ is the rms voltage of the n:th harmonic ($\tilde{e}_2$ is the first overtone to the fundamental) (see Reference 17.2). The relative level of the harmonics to the fundamental may be measured by analog or digital equipment. In either method, noise may be a problem when measuring small amounts of harmonic distortion.

In analog measurement of harmonic distortion, a *notch* filter is used. This filter rejects frequency components in a narrow frequency range. Sometimes the sound of the residual (harmonic distortion) components can yield insight into the distortion mechanism. Studying the residuals versus the signal on an X-Y oscilloscope is also useful (see Reference 17.6). The measurement can also be done using a narrow band bandpass filter that is swept over the frequency range. The signal spectrum is then best plotted showing relative level in dB.

When using digital equipment, it is important to use sufficiently high resolution A/D converters as discussed in Chapter 16. For the best results in measurement of harmonic distortion using FFT, the sampling frequency and window length should be chosen so that spectral line broadening due to frequency domain *leakage* does not occur. This is best done by having an integer number of cycles of the sine wave signal within the FFT analysis window (see References 17.4 and 17.5). Otherwise, a modern window weighting function such as the Blackman-Harris window should be carefully used. It must also be stressed that the sampling frequency must be much higher than any important harmonics, at least twice as high. The reason is that high harmonics may otherwise be mirrored into the audible frequency range due to aliasing. The test equipment must have suitable anti-aliasing filters. Built in sound cards, particularly in laptop computers, seldom have such filters and will also pick up noise from the digital circuits and power supply feeds of the computer. A high resolution, high sampling frequency outboard sampling unit is recommended.

Small amounts of harmonic distortion may be obscured by noise in the system. It may then be necessary to analyze the audibility of the distortion frequency components concerning their relative power in the critical bands where they occur.

It is difficult typically to see the presence of small amounts of low order harmonic distortion, less than 5%, in the waveform of a sine-wave signal. Due to the influence of frequency, domain masking in hearing the harmonic distortion may or may not be a suitable metric for nonlinearities. The audibility of the generation of signal harmonics will depend on the signal complexity. If the signal has multiple frequencies, intermodulation will occur, creating new nonharmonically related frequencies as discussed.

An important effect in the audibility of harmonic distortion is that the overtones of most musical instruments and voice are not exactly harmonically related to their fundamentals. When major harmonics are created by the nonlinearities of the audio system (so that they are about the same amplitude as those of instruments and voice), hearing will perceive the interaction between the two as the result of modulation. For modulation frequencies below 150 Hz, such modulation is called *roughness* (see Reference 17.3).

The audibility of harmonic distortion may depend on the frequency filter characteristic of the audio system component studied. Harmonic distortion generated by low- and mid-frequency range loudspeakers is likely to be more audible than that generated by high-frequency loudspeakers.

Here, only steady-state or quasi-steady-state measurements were considered. Recent advances in signal processing have made it possible to use sweep signals to considerably reduce the time needed for full frequency range measurement of harmonic distortion (see Reference 17.7).

17.16 DIFFERENCE FREQUENCY AND INTERMODULATION DISTORTION

Any nonlinear system will cause intermodulation between two or more frequency components. Difference tones will be created along with harmonics as shown in Figure 17.13. Two types of test procedures are common, the SMPTE and CCIF methods, named after the respective standardization bodies. In the SMPTE method, *intermodulation distortion* (IMD) is measured using a system input consisting of a mix of a low (70 Hz) and a high-frequency (6 kHz) signal where the voltage of the 70 Hz signal is 4 times that of the 6 kHz signal (Figure 17.13b). In the CCIF method, *difference frequency distortion* (DFD) is measured in which a mix of two high-frequency signals 10 and 11 kHz of equal voltage are supplied to the system under test (Figure 17.13c). These signals are used to test wide range and high-frequency systems respectively. Since the purpose of distortion should be testing the system in a way that reflects its behavior with *real* audio signals, the two frequencies should be chosen with this in mind.

The IMD and DFD distortions in percentage are typically defined in analogy to harmonic distortion as, for example:

$$IMD = 100\frac{\sqrt{\sum_{1}^{n}\tilde{e}_{IM,n}^{2}}}{\tilde{e}} \tag{17.9}$$

Here $\tilde{e}$ is the rms voltage of the modulated signal and $\tilde{e}_{\text{dist}}$ are the rms voltages of the respective sideband or DFD signals (see Reference 17.2).

Results of IMD measurement are sometimes claimed to be more accurate than harmonic distortion in corresponding to the subjective perception of nonlinearities in audio sound reproduction. For wide band electronic systems that have fairly similar distortion characteristics over the audio range, there will be little

difference between the two in characterization of the leveled system's distortion properties. For frequency limited systems, such as loudspeakers, properly chosen mixed frequencies that generate IMD with suitable frequencies may be more relevant. Note that there will also be harmonic distortion generated by the mixed frequencies, so it is important to define which distortion frequency components are measured.

Even more than in the measurement of harmonic distortion, it is important in IMD measurement to choose analysis window length, sampling frequency, and number of samples within the analysis window wisely. The IMD and DFD test frequencies should be chosen appropriately to the systems bandwidth and other relevant characteristics. Since the measurement is not particularly sensitive to the exact frequencies used, it is clearly advisable to chose IMD and DFD test frequencies. These frequencies and the distortion frequency components will then correspond to the FFT analysis frequencies so that frequency domain leakage is minimized, unless measurement standards require differently.

Another factor that one might want to take into account is the loudness of the distortion frequency components. These might be masked and an evaluation procedure similar to that used in the determination of Zwicker's Loudness could then be appropriate see Chapter 3 and (Reference 17.3).

17.17 MULTITONE DISTORTION MEASUREMENT

As briefly mentioned in the previous section, it might be advantageous to choose test frequencies that are similar in behavior to *real* audio signals. Real audio signals are characterized by their statistical properties, such as positive and negative means, peak values, peak to rms values, and their modulation properties. The signal generated for Speech Transmission Index testing (see Chapter 3) is an example of a signal generated to simulate speech in audio testing. Similar signals have been created for music simulation, *Music Transmission Index*.

In practice, however, it might be wiser to chose signals having line spectra rather than noise signals since the frequency components can then be adjusted concerning phase as well so that a suitable peak to rms ratio can be obtained. A suitable input signal, without frequency domain leakage, can be obtained by using 5 low- and 5 high-frequency components to simulate popular music audio (see Reference 17.4 and general references for this chapter).

Since the signal consists of many different tones, the resulting distortion components will be spread over many frequencies, generating a distortion *noise*. Because leakage was eliminated by the choice of signal and analysis scheme, it is then straightforward to measure this distortion noise by simple elimination of the input signal frequency components. One can then define a frequency-dependent distortion density metric that can further be analyzed by taking the frequency domain masking of hearing into account.

17.18 FM DISTORTION

Because the signals radiated by a loudspeaker are from superimposed vibrations on the loudspeaker's diaphragm there will be intermodulation. This type of distortion is usually called frequency modulation (FM) distortion. The sound power radiated by a loudspeaker is proportional to the volume velocity generated by the diaphragm's motion. The volume velocity is proportional to the diaphragm excursion times the angular frequency. A small loudspeaker will thus need larger excursions at low frequencies

to radiate the same power as a loudspeaker having a large diaphragm. This means that the smaller the loudspeaker the more FM-distortion will be generated. The FM distortion of the mix between a low- and a high-frequency tone will have similar characteristics to IM distortion but with many sidebands around the higher frequency as shown in Figure 17.13d.

17.19 MEASUREMENTS USING SPECIAL CHAMBERS

Two specialized types of rooms are used in audio testing, *anechoic chambers* and *reverberation chambers*. A special type of measurement room intended to simulate a living room environment is the *IEC 268-13 listening room* of which an example is shown in Figure 6.17 (see Reference 17.8).

Reverberation chambers are typically used to determine the power output (and sensitivity) of loudspeakers. The measurement is usually done using third octave band analysis with a random noise signal using equal energy per percentage bandwidth (that is having constant power in each third octave frequency band). Such a signal is often called a pink noise signal to differentiate it from a signal that has thermal noise characteristics. Since many loudspeakers such as studio monitoring loudspeakers are designed to have a flat, frequency-independent, on-axis FR the power response of the loudspeaker will typically drop with increasing frequency since the directivity of the loudspeaker typically increases with frequency. Figure 17.14 shows a reverberation chamber. It is important to

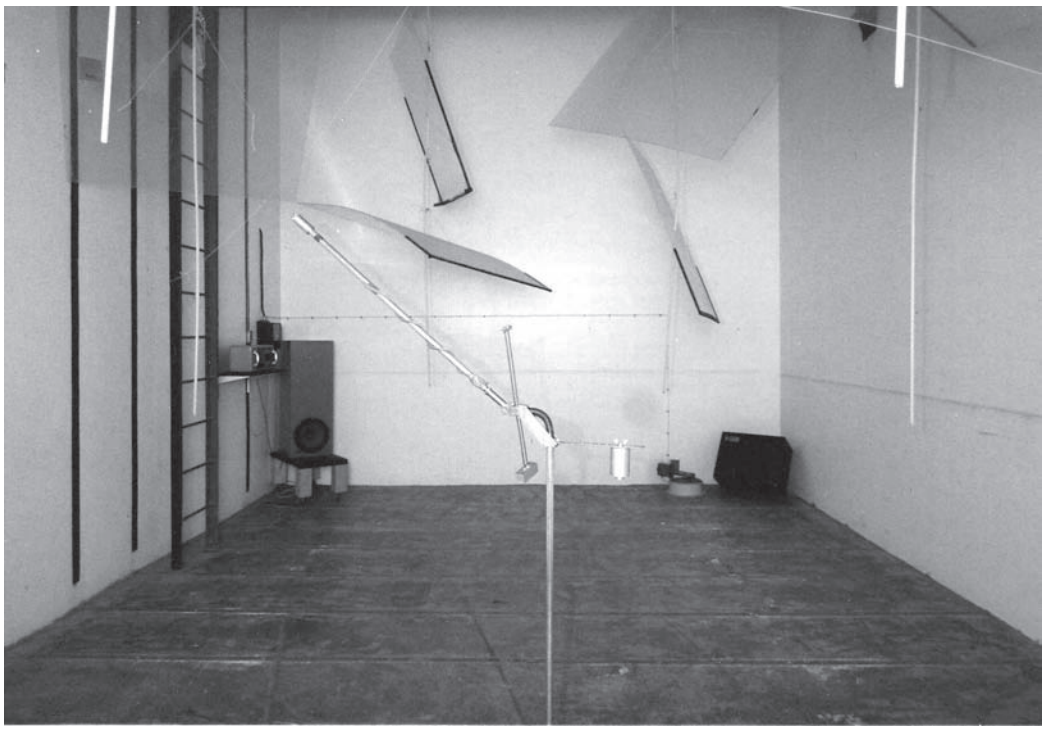


Figure 17.14 A typical reverberation chamber with diffusors. (Photo by Mendel Kleiner.)

note that reverberation chambers need to be large to support enough room modes for the sound field to be diffuse at low frequencies.

When measuring the power response in the reverberation chamber, it is important to position the loudspeaker relative to the chamber corner surfaces so that the placement corresponds to the recommended usage (see Chapter 4). Some loudspeakers are designed to be listened to in a situation where the direct sound of the loudspeaker is small compared to the reverberant sound; for such loudspeakers the reverberation chamber is ideal for measurement.

The rationale for using anechoic measurement chambers is to provide a sound reflection free measurement environment similar to out-of-doors conditions. Because of practicality, noise, weather, confidentiality, and such, it is more practical to use an anechoic chamber than to measure outdoors. Figure 17.15 shows such a room. For low frequency measurement an outdoor facility will be better since the lower frequency limit of anechoic chambers is determined by the sound absorption of the surface material and room size. Measurement of high directivity devices needs large rooms in order to be in the far-field of the device.

It is important to remember that an anechoic chamber still is a room with sound-reflecting walls and objects that create a diffuse sound field like that in a reverberation chamber. The relative level difference between direct sound and reverberant sound is much higher and the reverberation radius considerably larger.

The influence of the reverberant field in anechoic chambers is usually investigated by measuring the systematic and random deviation from the geometric attenuation, −6 dB per distance doubling, for a point source. In practice, it is difficult to build anechoic chambers having less than 1 dB standard deviation at 1 m distance.

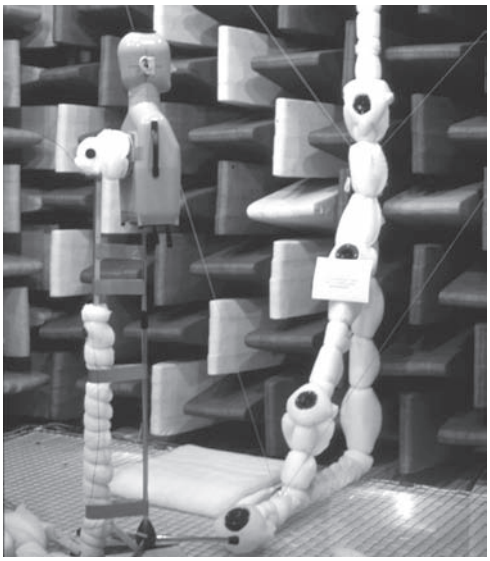


Figure 17.15 An anechoic chamber having sound-absorptive wedges on the walls in use for HRTF measurements. The floor is made of sound transparent netting under which there are sound-absorptive wedges as on the walls. (Photo by Mendel Kleiner.)

Figure 17.16 shows the 67% confidence interval for pure tones levels in a room (see Reference 17.9). In addition to the distance between sound source and receiver, the confidence intervals are determined by the directivity of the sound source and microphone as well as by the reflection coefficient of the walls. Figure 17.17 shows the measured deviation from the distance law at 1 m distance for a small point source in the anechoic chamber shown in Figure 5.1.

The inside surfaces of anechoic chambers typically are covered by porous, wedge-shaped sound-absorptive wall elements. These quickly lose their sound-absorptive properties when their length becomes shorter than one quarter of the wavelength of sound. The modal behavior of the room is easily noticeable below the chamber's lower limiting frequency.

To obtain negligible reverberation and low-frequency modes at the low-frequency end of the audio range, the chamber and its wedges must be very large as indicated previously. Since building

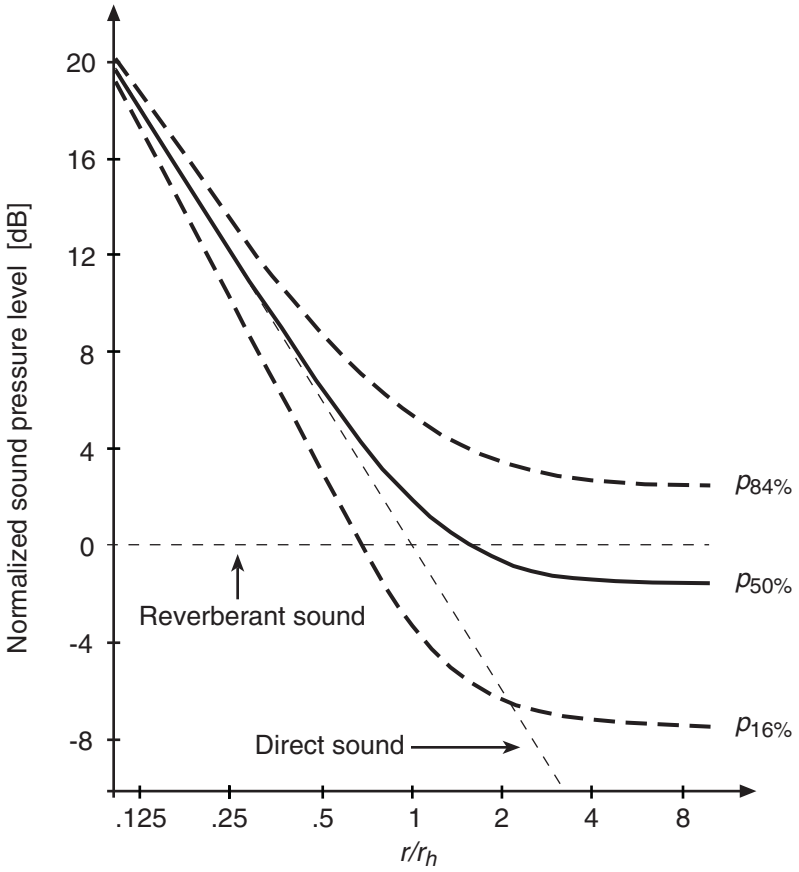


Figure 17.16 Expected spread in measurement of sound pressure level in a room. Sound pressure level distribution: 16, 50, and 84 percentiles shown. (After Ref. 17.9)

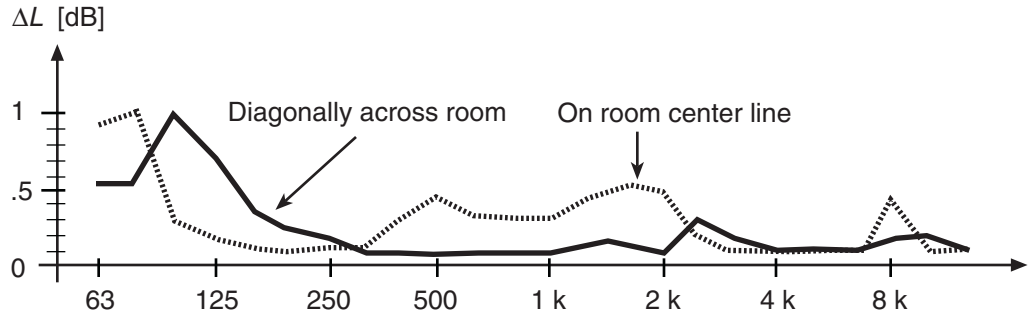


Figure 17.17 Measured deviation ΔL from the geometrical distance attenuation from a small sound source at 1 m distance in the anechoic chamber shown in Figure 5.1. Measured without the loudspeaker array shown in that figure present.

costs are approximately proportional to built volume, this makes anechoic chambers for low-frequency measurement prohibitively expensive.

Most anechoic chambers are suspended on spring-type vibration isolators to minimize sound leaking into the chamber by way of vibrations in the building structure in which the chamber is housed. One major advantage of a well-built anechoic chamber is its quiet. Background noise spectra below the threshold of hearing can be achieved.

17.20 IMPULSE RESPONSE ANALYSIS

The transfer function is coupled to the impulse response via the Fourier transform. The length (number of samples) of the impulse response and the sampling frequency determine the number of frequency points in the transfer function. The impulse response is only valid as a measurement for linear systems but is typically valid for the study of audio systems that are well below the amplitude limit set by clipping.

Analysis of the impulse response was discussed in conjunction with room acoustics in Chapter 5. Typically, the impulse response of a modern laboratory condenser microphone has such good properties that it does not play a role in the analysis of the performance of loudspeakers and rooms.

One way of avoiding the influence of wall reflections and subsequent reverberation on the measurement of loudspeakers in anechoic chambers (and to some extent also in other rooms) is to use impulse response measurement and apply time-gating to the measurement. Figure 17.18 shows the principle of time gating. If the room is large, the reflection-free time available between the direct sound from the source and the arrival of the scattered sound from the walls will be long. Time-gating corresponds to having a perfect anechoic chamber within the time slot under analysis.

Since the high-frequency components in the impulse response of electrodynamic loudspeakers and microphones generally have a length of interest of a few milliseconds, it is often advantageous to use time gating techniques and to remove the part of the impulse response that occurs after 10 ms after the

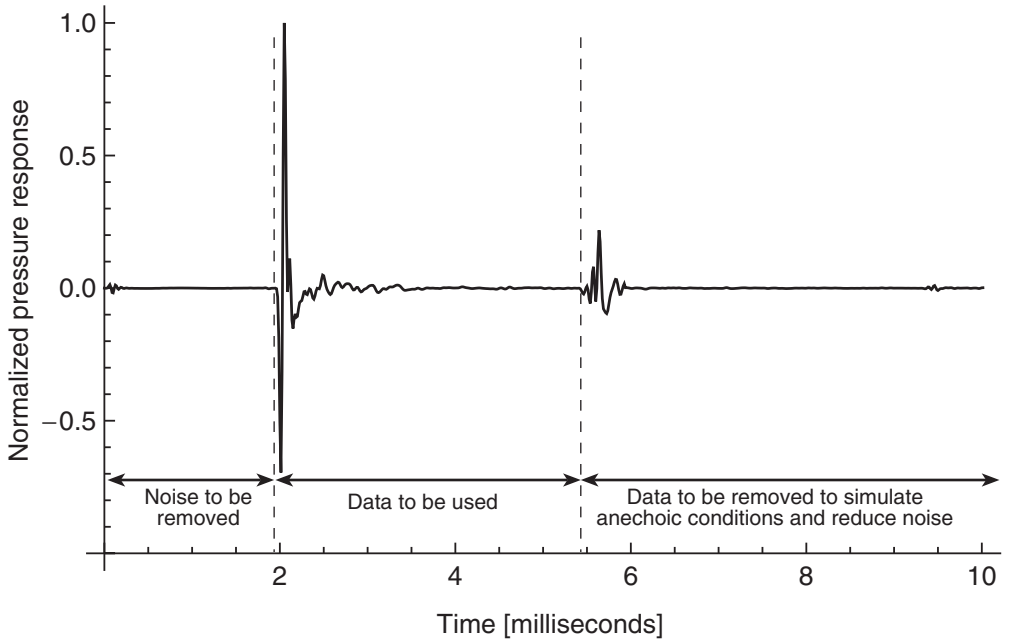


Figure 17.18 Application of time-gating to remove undesired reflection and noise from the measurement of the impulse response of a loudspeaker.

arrival of direct sound from the analysis. This corresponds to having the walls of the anechoic chamber more than 2 m away. This creates a virtually reflection-free acoustic environment. Some audio measurement software has built-in analysis routines for frequency-dependent time-gating.

The time-gating approach works quite well at medium and high-audio frequencies but gives problems in measuring the low-frequency properties of the loudspeaker unless one uses *zero padding* of the impulse response tail, extending its length. This approach causes errors in the measurement of the low FR, so it still makes sense to use a large chamber. One must remember that zero padding effectively assumes that the response of the loudspeaker is negligible after the zero gate time, so resonant systems characterized by strong long time extended oscillations will be incorrectly measured. Studying the sound pressure level envelope decay characteristics of the impulse response may give valuable guidance to the appropriateness of time gating.

17.21 FREQUENCY RESPONSE, SPECTROGRAM, AND WAVELET ANALYSIS

Once the impulse response has been trimmed by time-gating it can be analyzed to find interesting FR features of the loudspeaker or other device under test. Fourier transformation of the impulse response is trivial. Typically, the FR of the product such as a loudspeaker is one item of interest and is obtained using Equation 17.4.

Since hearing acts as a time/frequency analyzer, it is useful to have methods to study the information contained in the impulse response in a similar way as time-frequency histories. Two such methods are *short-time Fourier transform* (STFT) analysis for which the results are graphically presented as spectrograms and wavelet analysis for which the results are graphically presented as *scalograms*.

The basic principle of spectrogram analysis is to look at the FR magnitude in *time snippets*—gliding local analyzing time windows of the impulse response data. By using windowing techniques and moving the local analyzing time window along the time axis of the impulse response, one can graphically show how the frequency component develops over time in the response of the loudspeaker. Spectrogram analysis gives the same time and frequency resolution over the whole time/frequency region under study.

An alternative to the use of short-time Fourier transform-based spectrogram is to use wavelet analysis. While linear FR analysis is used in Fourier transforms, wavelet transforms are studied on a logarithmic frequency basis. The resulting scalograms offer different time and frequency resolution in different parts of the joint time/frequency region under study and correspond to constant relative bandwidth frequency analysis. The principle of the analysis by the two methods is shown in Figure 17.19.

The scalogram graphs are typically slightly easier to study than conventional STFT spectrograms. Wavelet analysis involves convolution of time snippets of the impulse response by suitably chosen short impulses, wavelets that are scaled in length depending on their base frequency. A commonly used wavelet is the modulated Gaussian wavelet, also called Morlet wavelet, shown in Figure 17.20. The energy in the resulting impulse response snippet is then graphically rendered as a function of time and frequency.

The advantage of wavelet analysis is clear when analyzing nonstationary signals, since it allows better detection and localization of transients than spectrograms. The compromise between good time or frequency resolution in the short-time Fourier transform analysis is eliminated in wavelet analysis because of the automatic rescaling of the resolution during the analysis of different frequency bands.

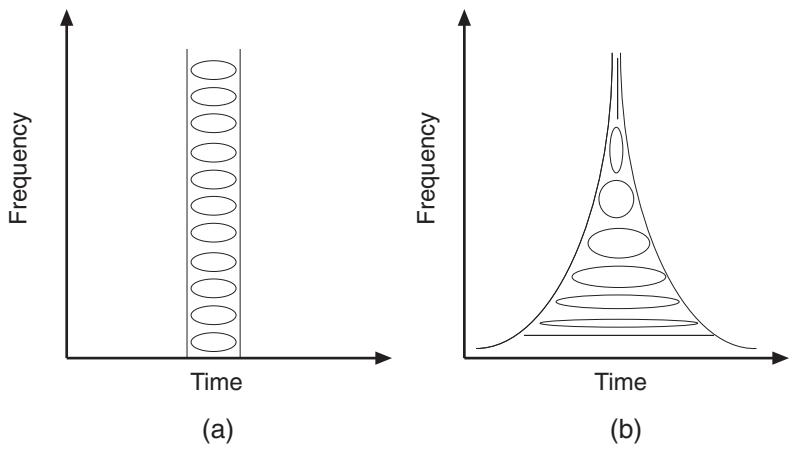


Figure 17.19 Time-frequency relationships in different octaves for: (a) the STFT-based spectrogram and (b) the wavelet transform-based scalogram). The spectrogram consists of a number of Fourier transforms of a signal multiplied by a *local analyzing window*, gradually shifted over the signal.

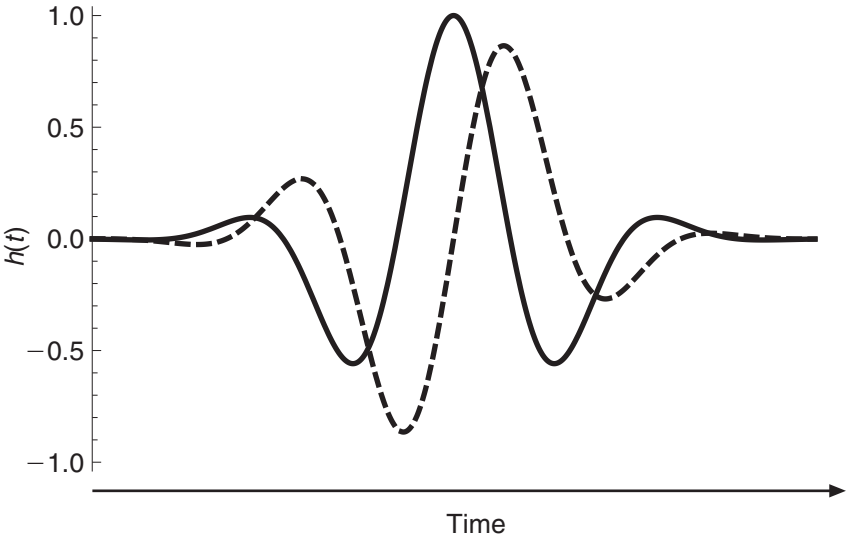


Figure 17.20 An example of the real (solid) and imaginary (dashed) parts of a Gaussian-type wavelet having the impulse response $h(t) = e^{-j\omega t/a} e^{-t^2 a^2/2}$. The parameter *a* is called the scale factor and determines the time and frequency resolutions.

Since the critical bands of hearing are approximately one-third octave band wide (over 500 Hz) it is common to use at least 3 wavelets per octave (sometimes this is referred to as 3 voices per octave in wavelet literature). The use of ten wavelets (voices) per octave often gives a graphically pleasing representation of the signal in wavelet analysis as shown in Figure 17.21.

17.22 DIRECTIVITY

The directivity of loudspeakers and microphones are important properties for their correct use and was discussed in Chapters 12 and 14. The measurement of directivity for out of plane angles is quite difficult, because a *directivity balloon* will need a few hundred measurement points. Directivity measurements in the plane are comparatively easy to perform using a computer operated turntable in an anechoic chamber. Directivity measurement results are shown in either polar or linear formats. The latter has some advantages but is less intuitive. Directivity measurements are essentially the same as frequency or impulse response measurements except that the loudspeaker or microphone is turned or moved to a set of angles relative to the measurement device.

With highly directional transducers, the response will be weak at angles away from the main lobe. The reverberation of the room will then cause considerable random variation in the response unless time gating is used. In some types of directivity measurement, narrow band noise (such as 1/3, 1/6, or even 1/12 octave noise) or multifrequency signals (such as a frequency modulated sinewave) are used instead of pure sinewaves to achieve a frequency average. Note that the noise signals should have pink noise characteristics.

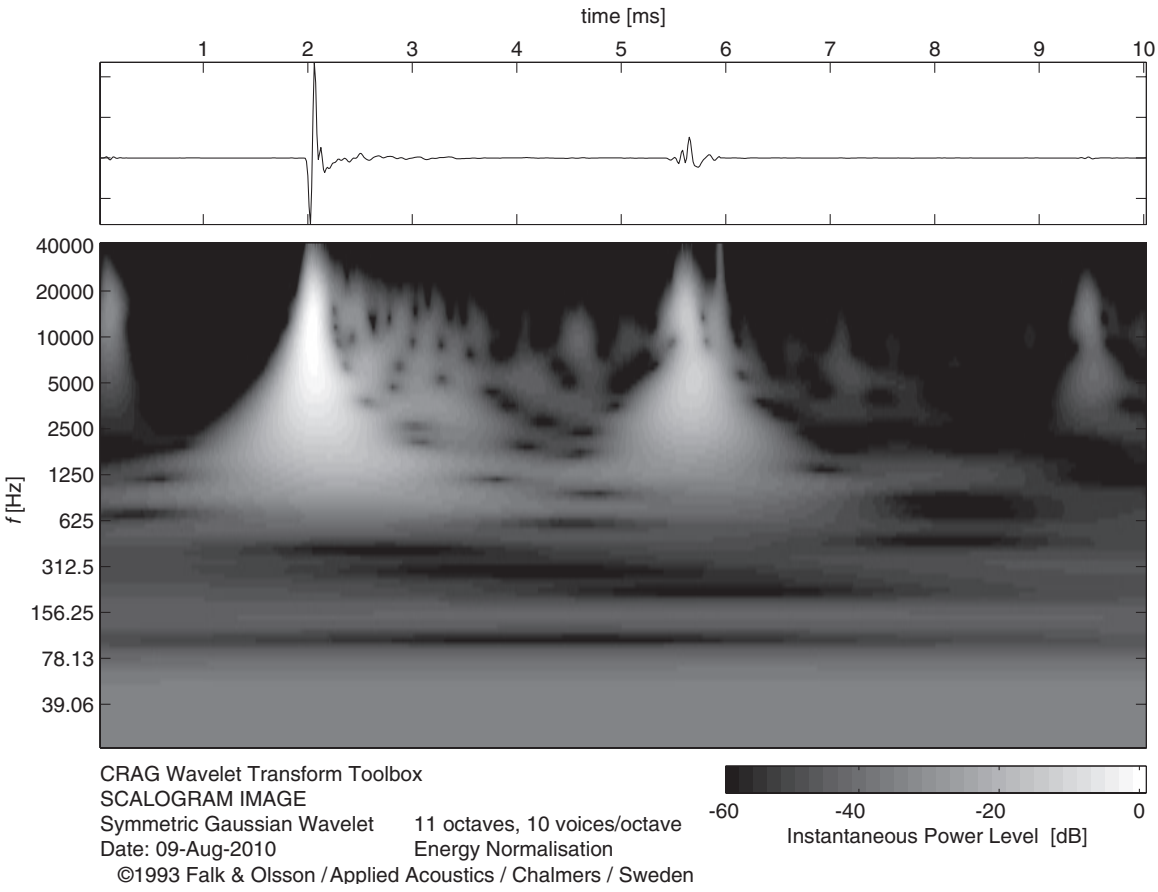


Figure 17.21 An example of the application of a wavelet transform-based scalogram to the loudspeaker impulse response shown in Figure 17.20.

A different way of measuring the directivity is to use a holographic approach, measuring the sound field at many points on a plane close to the diaphragm and then mathematically propagate the sound field to that at a large distance away from the loudspeaker.

17.23 SENSITIVITY

The sensitivity of loudspeakers and microphones (and gain of amplifiers) is comparatively easy to measure. Proper calibration of the measurement microphone is important. For loudspeakers, sensitivity is usually specified in sound pressure level for a certain input voltage, usually 2.83 volts (corresponding to 1 watt power input into a resistance of 8 ohms). For microphones the sensitivity is measured as the rms output voltage for the microphone placed in a plane wave sound field where the sound pressure level is 94 dB, that is, 1 Pa.

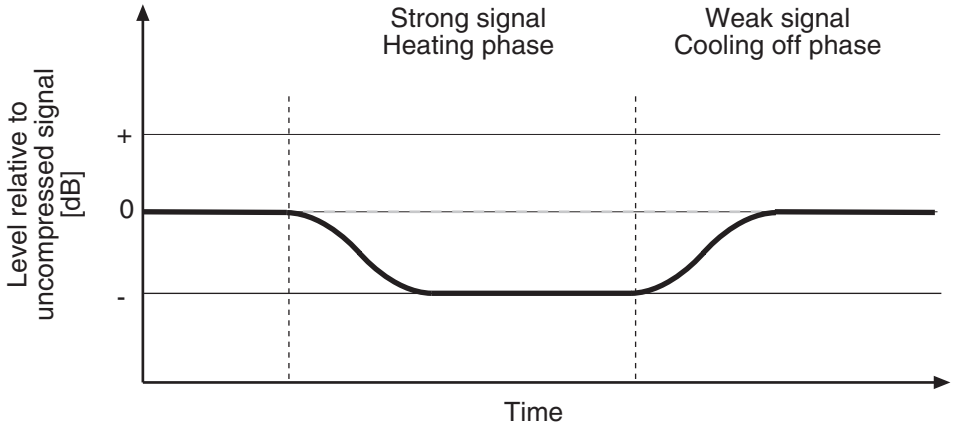


Figure 17.22 An example of the influence of compression effects on a loudspeaker's output.

17.24 COMPRESSION

A real audio signal such as voice and many types of music will be characterized by weak and strong parts—low-frequency modulation with modulation frequencies below 0.01 Hz. Since the voice coil of electrodynamic loudspeakers has limited thermal capacity, the voice coil will heat and cool with the signal envelope—the audio signal's low-frequency modulation. Since the electrical resistance of the voice coil is related to its temperature, the sensitivity of the loudspeaker will change causing a compression effect, resulting in a change in the *temporal envelope* of the radiated sound. This compression effect is important and is best measured using suitable gated/modulated tones or tone combinations or even music signals. An example of the influence of compression effects on the loudspeaker's output power is shown in Figure 17.22.

17.25 IMPEDANCE

The electrical input and output impedances of microphones, amplifiers, and loudspeakers are important properties. Typically, these are measured using small-signal methods, comparing the impedance of the measurement object to a reference using some form of electric series or bridge circuit. For loudspeakers, the FR of the electrical input impedance forms an important tool in tuning the loudspeaker enclosure to the loudspeaker driver. An example of the electrical input impedance characteristics of an electrodynamic loudspeaker was shown in Figure 14.11.

The traditional method of measuring loudspeaker impedance using a high value resistor in series with the loudspeaker gives erroneous results and should be replaced by a measurement using a low value series resistor in series instead. By measuring the voltage over this resistor, one can measure the current through the loudspeaker as it is driven by the amplifier output. This will let the loudspeaker operate as it is used in practice and give correct impedance values (although one is really measuring admittance that has to be inverted to yield impedance). This is particularly important if one is measuring a multidriver loudspeaker

having a crossover network that may otherwise have undesirable resonances or a loudspeaker driver that has little internal mechanical damping.

17.26 AUDIO SOUND CHARACTER AND QUALITY

Many different types of audio products such as microphones, amplifiers, and loudspeakers, and also related products such as rooms and cars, need to be evaluated using listening tests, that is, using humans as measurement instruments since the knowledge about human hearing, its signal processing, and signal evaluation is weak. The results of listening tests usually supplement physical measurements of audio quality and may possibly be used for finding the relevant physical measure (metric) for the audio sound character by correlating the results of the listening tests with various physical metrics or combinations of metrics, as illustrated in Figure 17.23.

In audio product evaluation, we differentiate between *audio sound character* and *audio sound quality* (see References 17.10 and 17.11). Character refers to the sensory properties of the product's sound whereas quality refers to the adequacy of its sound to satisfy the listener's expectations. Figure 17.24 illustrates the differences between audio character and quality.

The term sound quality is often used in audio to describe what really should be called audio sound character. Audio sound character is the set of psychoacoustic dimensions of the sound at the listener such as the noisiness, loudness, FR, timbre, roughness, dynamic compression, and spatiality (for example auditory source width and envelopment) of the sound at a particular setting of the product. The terms are used here similarly to their use in product sound engineering and in room acoustic design. The sound character can be measured using listeners that are asked to rate the sound on semantic differential scales, magnitude scales, or by pair comparisons. It is obvious that the use of words introduces considerable possibilities for bias in such measurements. It is important to instruct and describe to the listeners how they should listen without bias (see References 17.12–17.14).

Perceived audio sound quality is a collection of features that confer the product's ability to satisfy stated or implied needs. Perceived audio sound quality determination is the evaluation of preference or goodness and refers to the adequacy of the product's sonic characteristics in evaluating the total auditory

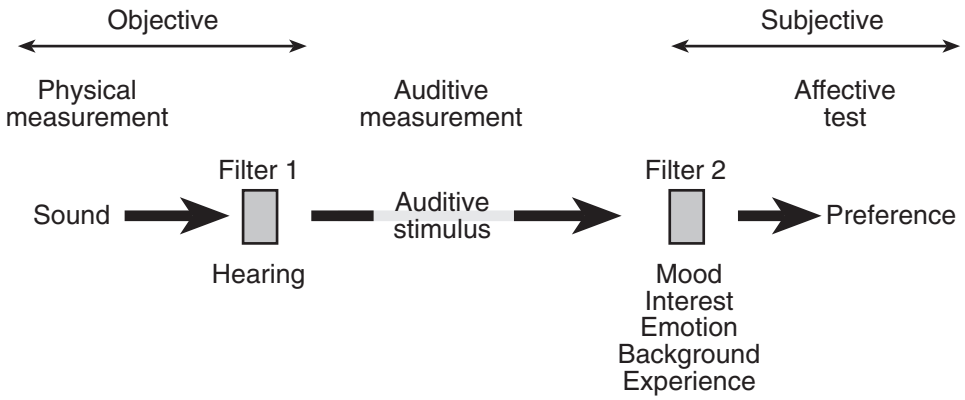


Figure 17.23 A simplified illustration of human sound perception and judgment.

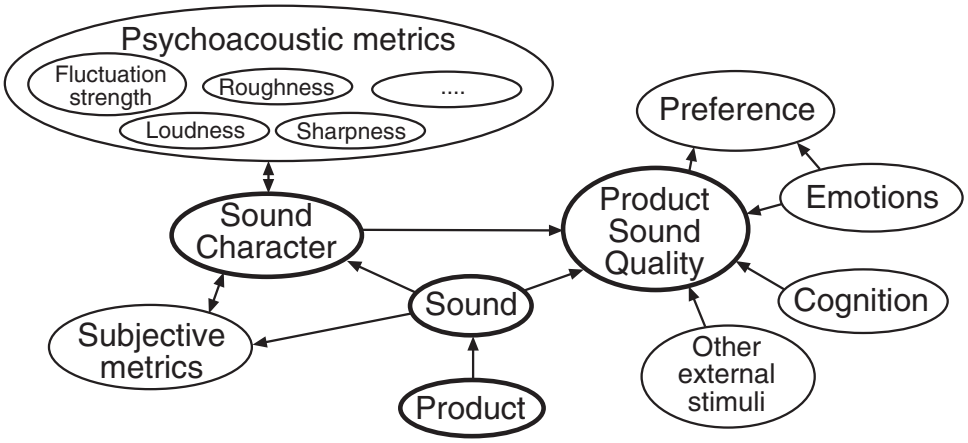


Figure 17.24 The difference between the audio sound character and sound quality concepts (inspired by Ref. 17.10).

characteristics of the sound, with reference to the set of desirable product features that are apparent in the user's cognitive and emotional situation. Any such evaluation is multidimensional and may be extremely dependent on distracters such as individual *taste* preferences, *cognitive factors* such as expectations (for example due to pretest information), *haptic* and *cognitive factors*, as well as *emotional* factors such as previous brand-experience and the audio program (sad or joyous music, for example).

Clearly, it is neither possible nor desirable to include these *distracters* into the set of audio metrics, but they need to be *neutralized* so that listeners/customers are able to account for their own biases and make informed decisions, for example, on recommendations for purchase. In other words, audio sound quality is not an inherent quality of the product but is a result of the perceptive and mental processes when a listener is exposed to the audio rendered by a product. Audio sound quality is not an absolute property but applies to a certain product and its customers. Audio sound quality tests must take acoustics, psychoacoustics, and psychology into account. By using proper test techniques and skilled listeners, the influence of distracters may be reduced.

17.27 LISTENING TESTS

Listening tests are *sensory tests* used in audio for the comparison of different products, situations and/or listening conditions (see Reference 17.15). In audio listening tests, one or more persons, *assessors*, are presented *audio samples* in a systematic way and asked to give their evaluations or other response in a prescribed manner. Listening tests may be either *objective* (perceptive—what do the assessors hear) or subjective (affective—what do the assessors prefer or dislike). The assessors may be naïve (no particular criterion) or initiated (have already participated in a sensory test). Typically, a group of assessors, a jury, is chosen to participate in a listening test. The test jury may be composed of engineers, marketing people, users, and/or buyers. The test jury is typically asked to judge not only the audio sound character but also the acceptability of a product in terms of its attributes, acceptance and quality (finding that a

product answers satisfactorily to the assessors expectations), and preference (finding one product better than one or several others). (See Reference 17.16.)

Because of time and economic constraints, it is often chosen to perform the listening test, using *initiated assessors* (*expert listeners*, sometimes also called *golden ears*) having specialized competence, knowledge or experience. Two types of initiated *expert* assessors are used, *expert assessors* and *specialized expert assessors*.

An expert assessor is a person with a high degree of sensitivity and experience of listening who is able to make consistent and repeatable assessments of various audio products. The specialized expert assessor has additional experience as a specialist in the product and/or process and/or marketing and is able to listen and evaluate or predict the effects of variations relating to the product. A group of experts who participate in a listening test is called an *expert panel.* A central problem in any listening test is that some audio properties are only discernible after very prolonged listening (and training).

17.28 COMMON LISTENING TEST TYPES

Perceptive tests are objective tests where humans are used as measuring instruments. The characteristics of the perceived stimulus are *measured* in objective terms without asking the test persons for any preferences. The tests are usually made using a panel of expert assessors who have been trained for the tests and trained to express their sensory perception in terms that have been well defined in advance. Such an expert panel may be trained for a specific purpose. The tests may be perceptive or affective.

Perceptive listening tests are objective audio sound character measurements with speech and music as stimuli where the assessors express their perception of the sound characteristics in terms that describe the sound as, for example, *loud*, *soft*, *distorted*, *sharp*, *smooth*, *rough*, *muddled*.

Affective listening tests are audio sound quality measurements, normally performed with a group of naive assessors who are representative of the relevant user group, *a consumer jury*—for example, car audio buyers. As they may use other words than audio assessors or *experts* for the attributes of the product they hear, the relevant words for expression of the heard sound often have to be *found* before the answering forms for the listening tests can be made. This can be done by interviews or focus group discussions. The words may be *exclusive*, *efficient*, *powerful*, *acceptable*, for example. The immediate response of the persons' judgments is usually desired, so naïve listeners are preferred for these types of tests. Affective tests are used when the preferred characteristics or the *dislikes* of a product are sought (for example, in preference tests). The main purpose is to give information about *the product in relation to listeners in a given context*.

17.29 SOME COMMON LISTENING TESTS

In audio sound character determination, the two most common tests are the *paired comparison test* and the *semantic differential test*.

The semantic differential test gives an assessment with words for specific isolated sound characteristics. Listeners are asked to rate their impression on a scale (see Reference 17.16). Two typical graphical scales are shown in Figure 17.25. Such a scale does not restrict the test person to specific points and has the advantage of absence of any numerical value associated with the response plus the limited use



Figure 17.25 Examples of bipolar (a) and unipolar (b) scales for semantic differential tests. The upper scale has open ends, the lower scale has one closed end. Typically two anchors help determine the scale.

of words to minimize word bias. After the test, the answers are measured for example in mm from the left end of the line.

The test scale in Figure 17.25a is bipolar in the sense that the words at the ends describe opposite characteristics. The scales may also be unipolar as in Figure 17.25b and describe the intensity of a characteristic.

Numeric scales may also be used—for example, as seven- or nine-point scales. They are fast to read after the test but restrict the assessors to choose numbers between the fixed points.

Many other scales exist such as graphical and verbal scales. The verbal nine-point *liking* scale shown in Figure 17.26 is often used for audio sound quality measurements of product acceptance-preference. The words used for questions and/or to scale the responses must be familiar, easily understood, and unambiguous to the subjects. The words must be readily related to the product and the task, and they must make sense to the subjects as to how they are applied in the test.

Words that have specific and useful meaning to the person requiring the test and to the professional audio engineer but may be much less meaningful to subjects, especially if they are not trained in understanding the properties referred to. Words used for the scales may be found using focus groups (a guided discussion involving a small group of persons with small differences in background and age). Interviews are another method to obtain words that describe the audio characteristics and quality of the product. Words used for the characterization reproduced sound may be found in Section 17.8.

In *paired comparison tests*, the sounds are presented in pairs for comparison using defined attributes. Each sound is successively compared to all other sounds. At least two comparisons should be made for each pair (*A, B*): a comparison with sound *A* first and another comparison with sound *B* first. The number of comparisons will be *N* where *N* is the number of sounds. The paired comparison test gives the listeners the ability to distinguish between a specified characteristic of different sounds. The result will typically be of the type *higher* or *lower* without information about the absolute placement (in the *high* end or the *low* end) of the sounds. Pauses between sounds to be compared must be short, typically less than one second long. Analysis of preference using Thurstone's method is a convenient and quick way of presenting the data on a scale of subjective preference as shown in Figure 17.27 (see References 17.18 and 17.19).

- Like extremely
- Like very much
- Like moderately
- Like slightly
- Neither like nor dislike
- Dislike slightly
- Dislike moderately
- Dislike very much
- Dislike extremely

Figure 17.26 An example of a verbal nine-point *liking* scale.

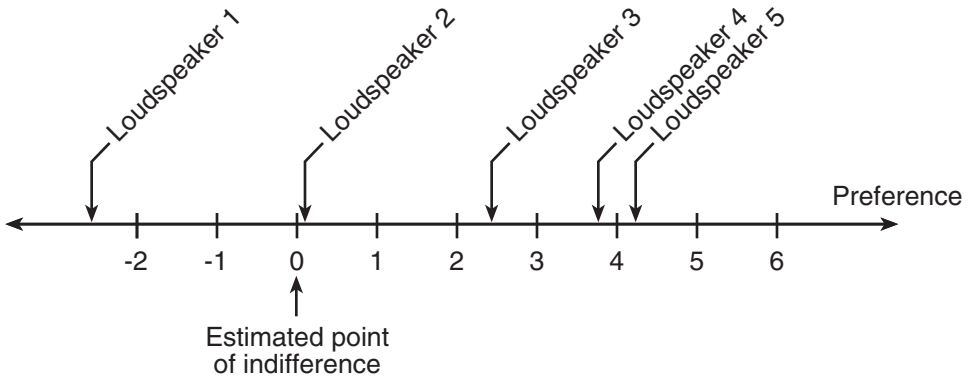


Figure 17.27 An example of a scale of subjective preference. The scale is given in units of normalized variance.

17.30 SELECTING AND TRAINING OF LISTENERS

The assessors for affective tests are normally a representative group of potential buyers, users, neighbors or other persons relevant for the actual situation. The persons should be qualified based on typical demographics (gender, age, education, cultural and ethnic background, income, etc.) physiological criteria (for example, hearing ability) and usage criteria.

In affective tests, one should in general avoid persons involved with the product, such as sales persons, technicians, designers, and employees at the company producing the product under test. If such persons are used, their responses should be compared with data from other groups to determine that the subjects are comparable, at least in their preferences toward a specific product or sound characteristic.

The *training* of the assessors for a test jury should be limited to training in the test procedure and to pre-tests with the purpose of ensuring that the words or other descriptors for the sounds presented are familiar and unambiguous to the subjects. The training will depend on the type of product and the accuracy and representability of the test in relation to the relevant market.

The *number of assessors* should be in the range from ten to several hundred. It is usually necessary to recruit two or three times the number of persons actually constituting the final panel. Successful tests are seen with 16 to 32 persons. The variance of the test results will give guidance for the reliability of the test.

17.31 EXPERT PANELS

The assessors in an expert panel are trained listeners. Expert listeners may be recruited from the office, plant, or laboratory staff. Still, as noted, one should avoid persons who are too personally involved with the products being examined, as they may cause bias. If such persons participate, their assessments should be compared with data from other groups in the panel to determine that the results are comparable for the specific sound characteristic under assessment. The expert listeners may also be recruited outside the organization by advertisement, among persons visiting the organization or among friends. The criteria for selection may be audiometric tests, reliability of judgments of pre-test sessions, and criteria for listening experience.

The main reason for training the expert assessors for a certain type of judgment is to make certain that different aspects of performance have reached an asymptotic level before the listening tests start. This will ensure that the assessments are precise as well as cost and time effective. The variability between selected and trained assessors is less than that between naïve listeners.

Ideally, it is not desirable to work with a panel of fewer than 10 selected expert assessors. For survey measurements, acceptable results may be obtained with fewer than 10 experts.

17.32 PLANNING AND PREPARATION OF A LISTENING TEST

It is difficult to avoid sources of bias in audio listening tests. Normally we want the listening test to be robust, unaffected by external variables. Planning, removal of nonacoustic sources of variance, selection of suitable assessors, randomizing the order of sample presentation can be used to minimize *bias*. However, in some tests regarding audio sound quality, knowledge of the product and the context are necessary to obtain relevant assessments. Multimodal sensory inputs are normal and must be taken into account during the planning of the test. Some modes of presentation are inherently difficult; for example, it is difficult to hide the tactile influence of headphone listening.

Bias is also a result from the choice of *listening level*. The listening level can, for example, be chosen as the natural level, loudness equalized level, or sound pressure equalized level. The equipment must be *calibrated* so that the playback level of the sound samples is known.

The presentation mode also may induce bias. Neither experimenter nor assessor should control the order of stimuli. So called *double-blind testing* is preferred—that is, randomized testing where the presentation order is out of the control of the test engineer.

Further bias may result from the *surroundings* and *context*. Are these real world conditions, laboratory, or simulated conditions? It is particularly difficult in an amateur listening test to avoid bias due to the *visual influence* of the listening room. Its room-acoustic properties, the placement of loudspeakers, and the

background noise are difficult to hide. In most cases, one will want to avoid influence from nonacoustical parameters, such as visual and tactile cues, or artist recognizability.

If possible and practical, it is recommended that listening tests are performed with *custom recordings* of natural (live) sound sources. Special care should be taken to ensure that the audio is presented at calibrated sound pressure levels.

Audible noise caused by electric noise in the presentation set-up and other unwanted background noise from ventilation equipment and transportation should be avoided.

Announcements are usually necessary to be given in connection with presentation of the sound samples (announcement of sample numbers). These should be recorded at relevant levels and in similar acoustic surroundings as the sound samples, in order to avoid confusion or other effects of different acoustics for sound samples and announcements. The announcements should also be presented at a natural and calibrated level.

17.33 THE TEST SESSION

The listening test should start with a presentation of test organization and test leader, purpose, listening conditions, safety, the expected duration and pauses and other practical information. The assessors are then provided with aural and written instructions that describe the test situation and conditions, sounds and scenario (imagine that you are . . .), and their task.

It is advantageous to provide sample rating sheets, to perform a pretest, and to provide a questions and answers session ahead of the core listening test. After the test, there should be a debriefing session in which one discusses the assessors' observations and experience of the listening test. The test should take into account normal engineering ethics and the personal safety of the assessors. This will put a limit on the loudness levels that can be used in loudspeaker and headphone testing.

It is difficult to concentrate on a test for more than 20 minutes. After this time, a break for some 10 to 15 minutes is necessary. Products containing sugar, such as soft drinks and candy, may be used liberally to ensure that the blood sugar levels of the assessors remain high. Additionally, no listener should be engaged for more than two hours of listening per day and must be given regular breaks.

17.34 INFORMATION TO BE REPORTED

When reporting on listening tests, it is important that the type of equipment, its brand, serial number and other relevant data used is stated in the measurement report along with the technical data. The technical data should be given referring to the IEC 268 standards series or the ITU-R BS.1116 recommendation mentioned previously as applicable. The relevant acoustical and other characteristics of test environment should be specified and includes:

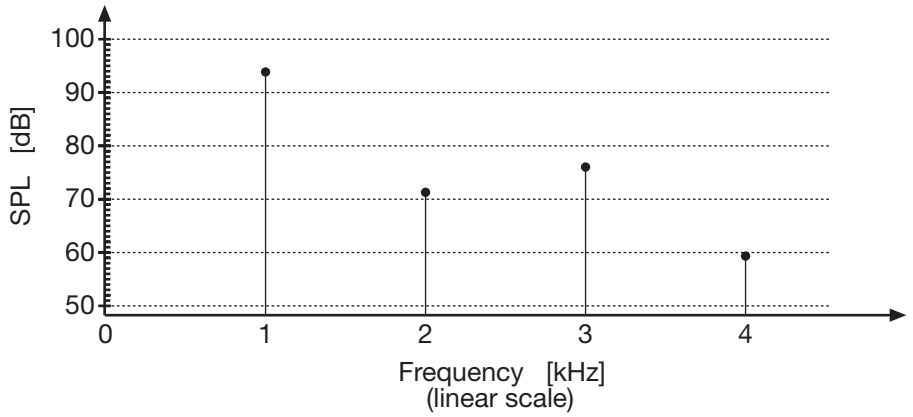
- Purpose of the measurements and involved parties
- Type of listening test (perceptive or affective)
- The test environment
- Descriptions of measuring objects/sound sources and their normal use
- The working conditions and stability of the source

- Recording, playback, and analysis equipment; type, make and model
- Description of the sound samples (recording technique, editing, announcements, length and pauses
- Criteria for selection of test persons, number of persons
- The method of the listening test, instructions to test persons, amount of information given to the test persons about the sound sources/situations of use or duration of the listening sessions, for example.
- The test conditions (surroundings, nonacoustical cues, individual or group test, etc).
- Results of listening tests, validation of results and test persons
- Graphical summary of data
- Mean standard deviations and confidence intervals of the results
- Statistical methods used
- Evaluation of listeners' qualifications from test results
- Generalization of test results
- Results of physical measurements
- Any correlation between physical measurement results and the results of the listening test, other types of statistical data processing
- Uncertainty and significance of the main results. Representativity of the results in relation to *non-test* situations. How representative are the tests?

17.35 PROBLEMS

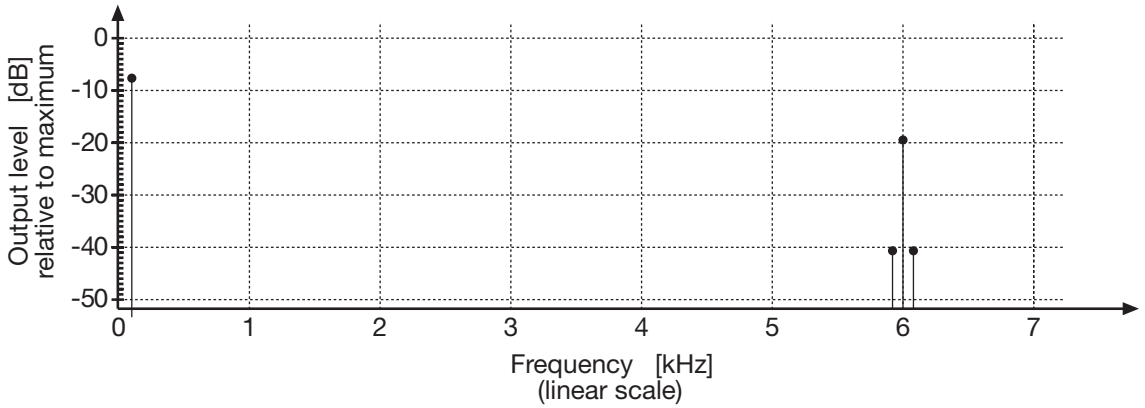
17.1 The graph shows the measured output spectrum for a loudspeaker being fed a sine wave at 1 kHz.

Task: Calculate the THD.



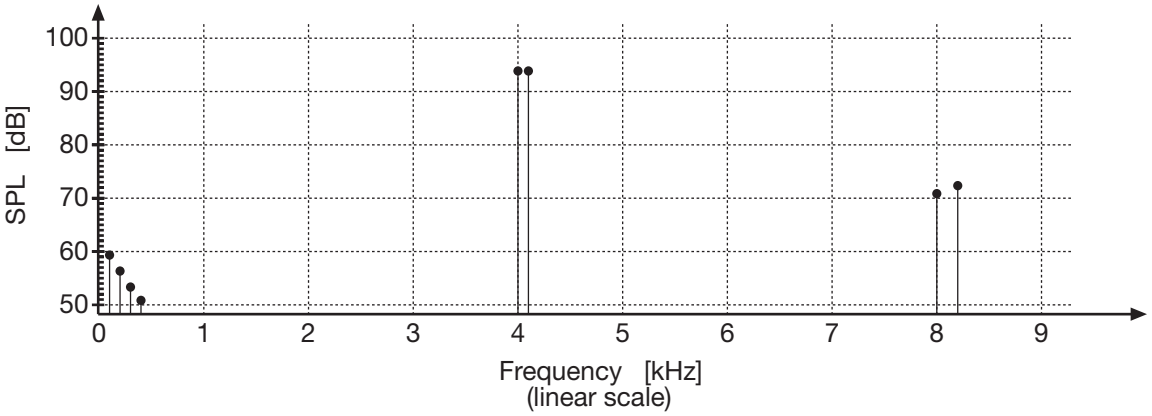
17.2 The graph shows the measured output spectrum of an amplifier for a signal consisting of 70 Hz and 6 kHz at a 4:1 ratio.

Task: Calculate the total IMD.

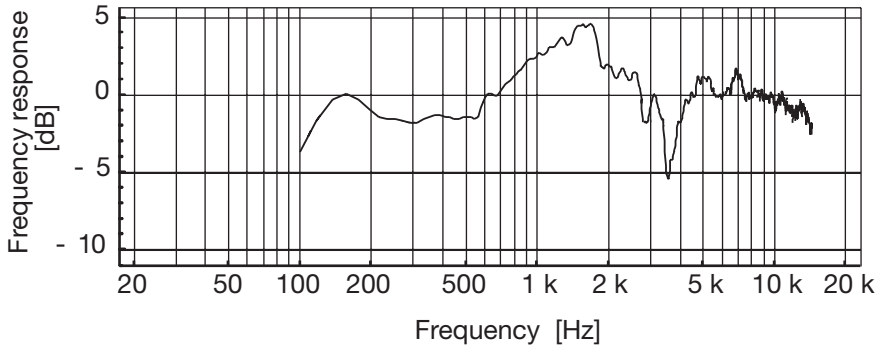


17.3 A loudspeaker was tested in an anechoic chamber using a two tone signal consisting of 4 kHz and 4.1 kHz sine waves. The test result is shown in the graph.

Task: Determine the harmonic and difference tone distortions. How can the test be improved to show greater clarity?



17.4 A loudspeaker was tested in an anechoic chamber with the FR result shown in the graph. How would you go about *improving* the measurement results (for example by redoing the test or by using signal processing) so that the loudspeaker becomes more attractive in the marketplace?



17.5 A listening test is to determine the relative timbre of 5 loudspeakers. How would you plan, design, and practically arrange the test and which difficulties do you foresee? Is the test sufficiently well specified?

Web Added Value™

This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

References

Introduction

i.1 Beranek, L. L., *Acoustics*. New York: McGraw-Hill (1954), reprinted by the Acoustical Society of America (1986). ISBN-13: 978-0883184943.

i.2 Olson, H. F., *Acoustical Engineering*, 3rd ed. Princeton, NJ: D. Van Nostrand (1957). ASIN: B001HDVCMO.

i.3 Benson, B. K., *Audio Engineering Handbook*. New York: McGraw-Hill (1988). ISBN-13: 978-0070047778.

i.4 Ballou, G. M., *Handbook for Sound Engineers*, 2nd ed. Carmel, IN: Sams (1991).

i.5 Geddes, E., *Audio Transducers*. Northville, MI: Gedlee (2002). ISBN-13: 978-0972208505.

i.6 Borwick, J., *Loudspeaker and Headphone Handbook*, 3rd ed. Oxford: Focal Press (2001). ISBN-13: 978-0240515786.

i.7 Colloms, M., *High Performance Loudspeakers*, 6th ed. Chichester, UK: J. Wiley & Sons (2005). ISBN-13: 978-0470094303.

i.8 Gayford, M. L., *Electroacoustics: Microphones, Earphones and Loudspeakers*. London: Newnes-Butterworth (1970). ISBN-13: 978-0408000260.

i.9 Hood, J. H., *Audio Electronics*, 2nd ed. Oxford, UK: Newnes (1999). ISBN-13: 978-0750643320.

i.10 Woram, J. M., *The Recording Studio Handbook*. Plainview, NJ: Elar Publishing (1982). ISBN-13: 978-0914130017.

i.11 Ahnert, W., and Steffen, F., *Sound Reinforcement Engineering*. London: E & FN Spon (1999). ISBN-13: 978-0415238700.

i.12 Watkinson, J., *The Art of Digital Audio*, 3rd ed. Oxford, UK: Focal Press (2001). ISBN-13: 978-0240515878.

i.13 Raichel, D. R., *The Science and Applications of Acoustics*. New York: Springer Verlag (2000). ISBN-13: 978-0387989075.

i.14 Kinsler, L. E., et al., *Fundamentals of Acoustics*. New York: J. Wiley & Sons (1999). ISBN-13: 978-0471847892.

i.15 Beranek, L. L., ed., *Noise and Vibration Control*. New York: McGraw-Hill (1971). ISBN-13: 978-0070048416.

i.16 Kuttruff, H., *Room Acoustics*, 5th ed. London: Spon Press (2009). ISBN-13: 978-0415480215.

i.17 Ando, Y., *Concert Hall Acoustics*. New York: Springer Verlag (1985). ISBN-13: 978-3540135050.

i.18 Beranek, L. L., *Concert Halls and Opera Houses*, 2nd ed. New York: Springer Verlag (2004). ISBN-13: 978-1441930385.

i.19 Barron, M., *Auditorium Acoustics and Architectural Design*. London: E & FN Spon (2003). ISBN-13: 978-0419245100.

i.20 Doelle, L. L., *Environmental Acoustics*. New York: McGraw-Hill (1972). ISBN-13: 978-0070173422.

i.21 Cavanaugh, W. J., et al., *Architectural Acoustics: Principles and Practice*, 2nd ed. New York: J. Wiley & Sons (2009). ISBN-13: 978-0470190524.

i.22 Lindsay, R. B., "The Story of Acoustics." *Journal of the Acoustical Society of America*, vol. 39, pp. 629–644 (1966).

i.23 Pohlmann, K. C., *Principles of Digital Audio*, 6th ed. New York: McGraw-Hill (2010). ISBN-13: 978-0071663465.

Chapter 1 Sound

1.1 Beranek, L. L., *Acoustics*. New York: American Institute of Physics (1986). ISBN-13: 978-0883184943.

1.2 Kuttruff, H., *Physik und Technik des Ultraschalls*. Stuttgart, S. Hirzel Verlag (1988). ISBN-13: 978-3777604275.

1.3 Skilling, H. H., *Electrical Engineering Circuits*. New York: Wiley (1966). ASIN: B000V0M8OU.

General references

Kinsler, L. E., et al., *Fundamentals of Acoustics*. New York: J. Wiley & Sons (1999). ISBN-13: 978-0471847892.

Reynolds, D. R., *Engineering Principles of Acoustics*. Boston: Allyn and Bacon (1981). ISBN-13: 978-0205072712.

Seto, W. W., *Acoustics*. McGraw-Hill (1971). ISBN-13: 978-0070563285.

Chapter 2 Audio Signals

2.1 General Radio, *Handbook of Noise Measurement*. General Radio Division (1972). ASIN: B00314JQUW.

General references

Broch, J. T., *Principles of Experimental Frequency Analysis*. London: Elsevier Applied Science (1990). ISBN-13: 978-1851665549.

Hassall, J. R., and Zaveri, K., *Acoustics Noise Measurements*. Copenhagen, DK: Brüel & Kjær (1988). ISBN-13: 978-8787355216.

Randall, R. B., *Frequency Analysis*. Copenhagen, DK: Brüel & Kjær (1988). ISBN-13: 978-8787355070.

Chapter 3 Hearing and Voice

3.1 Blauert, J., *Spatial Hearing*. Cambridge, MA: The MIT Press (1996). ISBN-13: 978-0262024136.

3.2 Ando, Y., *Concert Hall Acoustics*. Berlin: Springer Verlag (1985). ISBN-13: 978-3540135050.

3.3 Zwicker, E., and Fastl, H., *Psychoacoustics*. Berlin: Springer Verlag (1990). ISBN-13: 978-3540526001.

3.4 von Békésy, G., *Experiments in Hearing*. New York McGraw-Hill (1960). ISBN-13: 978-0070043244

3.5 Möller, A. R., et al., *Man and Noise* (in Swedish). Lund, SE: Studentlitteratur (1978).

3.6 Spoor, A., "Presbyacusis Values in Relation to Noise-Induced Hearing Loss." *Audiology*, vol. 6, p. 48 (1967).

3.7 Brüel & Kjær, *Acoustic Noise Measurement*. Nærum, DK: Brüel & Kjær (1988).

3.8 Kleiner, M., "Problems in the Design and Use of Dummy-Heads." *Acustica*, vol. 41, pp. 183–193, (1978).

3.9 Fant, O., *Speech Sounds and Features*. Cambridge, MA: The MIT Press (1973). ISBN-13: 978-0262060516.

3.10 Flanagan, J. L., "Analog Measurements of Sound Radiation from the Mouth." *Journal of the Acoustical Society of America*, vol. 32, p. 1616 (1960).

3.11 Fletcher, H., *The ASA Edition of Speech and Hearing in Communication*. Acoustical Society of America (1995 reprint of 1961 edition). ISBN: 978-1563963933.

3.12 Mapp, P., "Measuring Intelligibility." *Sound and Communications*, pp. 56–68 (April 2002).

3.13 Houtgast, T., et al., "Predicting Speech Intelligibility in Rooms from the Modulation Transfer Function. I. General Room Acoustics." *Acustica*, vol. 46, pp. 60–72 (1980).

General references

Green, D. M., *An Introduction to Hearing*. Hillsdale, NJ: Lawrence Erlbaum Associates Publishers (1976). ISBN-13: 978-0470151884.

Kinsler, F. et al., *Fundamentals of Acoustics*, 2nd ed. New York: J. Wiley & Sons (1962). ASIN: B0012NMF3O.

Guyton, A. C., *Textbook of Medical Physiology*, 11th ed. London: Saunders (2005). ISBN-13: 978-0721602400.

Probst, R., et al., A Review of Otoacoustic Emissions. *Journal of the Acoustical Society of America*, vol. 89, pp. 2027–2067 (1991).

Chapter 4 Basic Room Acoustics

4.1 Cremer, L., et al., *Principles and Applications of Room Acoustics*. New York: Applied Science Publishers (1982). ISBN-13: 978-0853341130.

4.2 Kuttruff, H., *Room Acoustics*, 5th ed. London: Spon Press (2009). ISBN-13: 978-0415480215.

4.3 Harris, C. M., "Absorption of Sound in Air vs. Humidity and Temperature." *Journal of the Acoustical Society of America*, vol. 40, pp. 148–159 (1966).

Chapter 5 Spatial Sound Perception

5.1 Burgtorf, W., et al., "Untersuchungen zur Wahrnehmbarkeit verzögerter Schallsignale." *Acustica*, vol. 11, p. 97 (1961).

5.2 Seraphim, H. P., "Über die Wahrnehmbarkeit mehrerer Rückwürfe von Sprachschall." *Acustica*, vol. 11, p. 80 (1961).

5.3 Schubert, P., "Wahrnehmbarkeit von Einzelrückwürfen bei Musik." *Zeitschrift für Hochfrequenztechnik u. Elektroakustik*, vol. 78, p. 230 (1969).

5.4 Barron, M., "The Subjective Effects of First Reflections in Concert Halls." *J. Sound Vib*., vol. 15, p. 4 (1971).

5.5 Meyer, E., and Schodder, G. R., "Über den Einfluss von Schallrückwürfen auf Richtungslokalisation und Lautstärke bei Sprache." *Nachr. Akadem. Wissenschaften*, Göttingen, Math.-Phys. Nr 6, p. 31 (1962).

5.6 Franssen, N. V., *Stereophony*. Eindhoven: Philips (1964). ISBN-13: 978-0333038444.

5.7 Haas, H., "The Influence of a Single Echo on the Audibility of Speech." *Journal of the Audio Engineering Society*, vol. 20:2, pp. 146–159 (1972).

5.8 Atal, B. S., et al., "Perception of Coloration in Filtered Gaussian Noise; Short Time Spectral Analysis by the Ear." *Proc. Int. Cong. Acoust*., Copenhagen paper H3 (1962).

5.9 Cremer, L., et al., *Principles and Applications of Room Acoustics*. New York: Applied Science Publishers (1982). ISBN-13: 978-0853341130.

5.10 Atal, B. S., et al., "Subjective Reverberation Time and its Relation to Sound Decay." *Proc. 5th. Int. Cong. Acoust*, Liège, paper G32 (1965).

5.11 Damaske, P., "Subjektive Untersuchung von Schallfeldern." *Acustica*, vol. 19, p. 199 (1967–1968).

5.12 Reichardt, W., et al., "Raumakustische Nachbildungen mit elektroakustischen Hilfsmittein." *Acustica*, vol. 17, p. 75 (1966).

5.13 Beranek, L. L., *Concert Halls and Opera Houses*, 2nd ed. New York: Springer Verlag (2004). ISBN-13: 978-1441930385.

5.14 Kuttruff, H., *Room Acoustics*, 5th ed. London: Spon Press (2009). ISBN-13: 978-0415480215.

5.15 Kleiner, M., "Speech Intelligibility in Real and Simulated Sound Fields." *Acustica* (1978).

5.16 Kleiner, M., Dalenbäck, B-I., and Svensson, P., "Auralization—an Overview." *J. Audio Engineering Soc.*, vol. 41:11, pp. 861–875, (1993).

5.17 Schroeder, M. R., "New Method of Measuring Reverberation Time." *J. Audio Engineering Soc.*, vol. 35, pp. 299–305 (1987).

5.18 Kleiner, M., "Subjective Perception of Sound Field Simulation and Variable Acoustics by Some Passive and Electroacoustic Systems." *Applied Acoustics*, vol. 31, pp. 197–205 (1990).

5.19 Kleiner, M., and Svensson, P., "A Review of Active Systems in Room Acoustics and Electroacoustics." *Int. Symp. on Active Control of Sound and Vibration*. Newport Beach, CA (July 1995).

General references

Ando, Y., *Concert Hall Acoustics*. New York: Springer Verlag (1985). ISBN-13: 978-3540135050.

Burgtorf, W., et al., "Untersuchungen über die Richtungsabhängige Wahrnehmbarkeit verzögerte Schallsignale." *Acustica*, vol. 14, p. 254 (1964).

Mehta, M., Johnston, J., and Rocafort, J., *Architectural Acoustics*. Columbus, OH: Merrill Prentice Hall (1999). ISBN-13: 978-0137937950.

Meyer, E., and Schodder, G. R., "Über den Einfluss von Schallrückwürfen auf Richtungslokalisation und Lautstärke bei Sprache." *Nachr. Akadem. Wissenschaften*, Göttingen, Math.-Phys. no. 6, p. 31 (1962).

Chapter 6 Room Acoustics Planning and Design

6.1 Beranek, L. L. ed., *Noise and Vibration Control*. New York: McGraw-Hill (1971). ISBN-13: 978-0070048416.

6.2 Tocci, G. C., "Room Noise Criteria—The State of Art in the Year 2000." *Noise News International*, vol. 8, p. 3 (2000).

6.3 Beranek, L. L., *Concert Halls and Opera Houses*, 2nd ed. New York: Springer Verlag (2004). ISBN-13: 978-1441930385.

6.4 Kleiner, M., Klepper, D. L., and Torres, R. R., *Worship Space Acoustics*. Ft. Lauderdale, FL: J. Ross Publishing (2010). ISBN-13: 978-1604270372.

6.5 Newell, P., *Recording Studio Design*. Focal Press (2007). ISBN-13: 978-0240520865.

6.6 Rettinger, M., *Handbook of Architectural Acoustics and Noise Control: a Manual for Architects and Engineers*. New York: McGraw-Hill (1988). ISBN-13: 978-0830626861.

6.7 Toole, F., *Sound Reproduction: The Acoustics and Psychoacoustics of Loudspeakers and Rooms*. Focal Press (2008). ISBN-13: 978-0240520094.

6.8 Kleiner, M., Dalenbäck, B-I., and Svensson, P., "Auralization—An Overview." *J. Audio Engineering Soc.*, vol. 41:11, pp. 861–875 (1993).

6.9 Kleiner, M., "Sound Field Enhancement." *McGraw-Hill Encyclopedia of Science and Technology*, New York, (2001).

6.10 Kleiner, M., "Sound Reinforcement." *McGraw-Hill Encyclopedia of Science and Technology*. New York, NY: (2001).

6.11 http://www.comsol.com/products/multiphysics/.

6.12 Pryor, R. W., *Multiphysics Modeling Using COMSOL: A First Principles Approach*. Sudburg, MA, Jones & Bartlett Publishers (2009). ISBN-13: 978-0763779993.

6.13 SoundEasy and BoxCAB software. Information at http://www.interdomain.net.au/~bodzio/.

6.14 ABEC2. Information at http://www.randteam.de.

6.15 Pietrzyk, P., and Kleiner, M., "The Application of the Finite Element Method to the Prediction of Sound Fields of Small Rooms at Low Frequencies." paper 4423, *The 102nd Audio Engineering Society Convention* (1997).

6.16 Granier, E., et al. "Experimental Auralization of Car Audio Installations." *Journal of the Audio Engineering Society*, vol. 44, Issue 10, pp. 835–849 October 1996.

6.17 Veneklasen, P.S., *Design Considerations from the Viewpoint of the Professional Consultant*. ICA 1974, Architectural Acoustics, Edinburgh, UK (1974).General references

Ando, Y., *Concert Hall Acoustics*. Berlin: Springer Verlag (1985). ISBN-13: 978-3540135050.

Doelle, L. L., *Environmental Acoustics*. New York: McGraw-Hill (1972). ISBN-13: 978-0070173422.

Egan, M. D., *Architectural Acoustics*. Ft. Lauderdale, FL: J. Ross Publishing (2007). ISBN-13: 978-1932159783.

Klepper, D. L., ed., "Sound Reinforcement," *Journal of the Audio Engineering Society, Vol. 1*. (1980). ASIN: B0015O6RAM.

Klepper, D. L., ed., "Sound Reinforcement", *Journal of the Audio Engineering Society, Vol. 2*. (1980). ISBN-13: 978-0937803301.

Long, M., *Architectural Acoustics*. New York, NY: Academic Press (2005). ISBN-13: 978-0124555518.

Mehta, M., Johnston, J., and Rocafort, J., *Architectural Acoustics*. Columbus, OH: Merrill Prentice Hall (1999). ISBN-13: 978-0137937950.

Chapter 7 Absorbers, Reflectors, and Diffusers

7.1 Lindblad, S., and Gudmundson, S., *Compendium of Acoustics vol. V* (in Swedish). Lund University of Technology: Dept. of Applied Acoustics (1981).

7.2 Kuttruff, H., *Room Acoustics*, 5th ed. London: Spon Press (2009). ISBN-13: 978-0415480215.

7.3 Wack, R., and Fuchs, H., "On the Use of Micro-Perforated Sails in Assembly Rooms." *Proc. 30. Deutsche Jahrestagung für Akustik,* Strasbourg (March 2004).

7.4 Doelle, L. L., *Environmental Acoustics*. New York, McGraw-Hill: (1972). ISBN-13: 978-0070173422.

7.5 Beranek, L. L., and Hidaka, T., "Sound Absorption in Concert Halls by Seats, Occupied and Unoccupied, and by the Hall's Interior Surfaces." *Journal of the Acoustical Society of America*, vol. 104, pp. 3169–3177 (1998).

7.6 Maekawa, Z., "Problems of Sound Reflection in Rooms." *Auditorium Acoustics*, ed. Mackenzie, R., London: Applied Science Publishers, pp. 181–196 (1975).

7.7 Maekawa, Z., "Noise Reduction by Screens." *Applied Acoustics*, vol. 1, pp. 157–173 (1968).

7.8 Kleiner, M., Klepper, D. L., and Torres, R. R., *Worship Space Acoustics*. Ft. Lauderdale, FL: J. Ross Publishing (2010). ISBN-13: 978-1604270372.

7.9 Schroeder, M., *Number Theory in Science and Communication: with Applications in Cryptography, Physics, Digital Information, Computing, and Self-Similarity*. Springer (2009). ISBN-13: 978-3642099014.

7.10 Wu, T., Cox, T. J., and Lam, Y. W., "From a Profiled Diffuser to an Optimized Absorber." *Journal of the Acoustical Society of America*, vol. 108:2 (2000).

7.11 Cremer, L., et al., *Principles and Applications of Room Acoustics*. New York: Applied Science Publishers (1982). ISBN-13: 978-0853341130.

7.12 Pattersen, O.KR.Ø., Sound Intensity Measurements for Describing Acoustic Power Flow, *Applied Acoustics*, vol. 14, pp. 387–397 (1981).

7.13 Fasold, W., et al., *Bau-und Raumakustik*. (in German) VEB Verlag für Bauwesen, Berlin. (reprinted 1987 by Verlag Rudolf Müller as ASIN-B002TUURJS)

Chapter 8 Waves in Solids and Plates

8.1 Beranek, L. L. ed., *Noise and Vibration Control*. New York: McGraw-Hill (1971). ISBN-13: 978-0070048416.

8.2 Cremer, L., et al., *Structure-Borne Sound: Structural Vibrations and Sound Radiation at Audio Frequencies*, 3rd ed. Berlin: Springer Verlag (2005). ISBN-13: 978-3540226963.

8.3 Kinsler, L. E., et al., *Fundamentals of Acoustics*, 4th ed. New York: J. Wiley & Sons (1999). ISBN-13: 978-0471847892.

8.4 Kurtze, G., *Physik und Technik der Lärmbekämpfung*. Karlsruhe, GE: Verlag G. Braun (1964). ASIN: B0000BKML2.

8.5 Granhäll, A., Mechanical impedance and vibration measurement. Report F81/05 Applied, Acoustics, Chalmers Gothenburg (1981) ISSN 0348-288x.

Chapter 9 Sound Radiation and Generation

9.1 Beranek, L. L. ed., *Noise and Vibration Control*. New York: McGraw-Hill (1971). ISBN-13: 978-0070048416.

9.2 Gösele, K., "Schallabstrahlung von Platten die zu Biegeschwingungen angeregt sind." *Acustica*, vol. 3, p. 243 (1953).

9.3 Roozen, N. B., "Reduction of Bass-Reflex Port Nonlinearities by Optimizing the Port Geometry." *Proc. Audio Eng. Soc. Conv.*, vol. 104:4661 (1998).

Chapter 10 Sound Isolation

10.1 Brandt, O., *Acoustic Planning* (in Swedish). Statens Nämnd för Byggnadsforskning. Stockholm (1959).

10.2 Kleiner, M., Klepper, D. L., and Torres, R. R., *Worship Space Acoustics*. Ft. Lauderdale, FL: J. Ross Publishing (2010). ISBN-13: 978-1604270372.

10.3 Kurtze, G., *Physik und Technik der Lärmbekämpfung*. Karlsruhe, Verlag G. Braun (1964). ASIN: B0000BKML2.

General references

Vigran, T. E., *Building Acoustics*. London: Spon Press. (2008). ISBN-13: 978-0415428538.

Hopkins, C., *Sound Insulation*. Maryland Heights, MO, USA Butterworth-Heinemann (2007). ISBN-13: 978-0750665261.

Chapter 11 Vibration Isolation

11.1 TD150 Subchassis Concepts. www.theanalogdept.com (sampled 2011).

11.2 Muster, D., and Plunkett, R., "Isolation of Vibrations." In Beranek, L. L., ed., *Noise and Vibration Control*. Washington: Inst. of Noise Control Engineering (1988). ISBN-13: 978-0070048416.

General references

Harris, C. M., "Vibration Control Principles." In Harris, C. M., ed., *Handbook of Acoustical Measurements and Control*. Reprinted by the *Acoust. Soc. Amer.*, Woodbury, NY (1998). ISBN-13: 978-0070268685.

Racca, R. H., "Types and Characteristics of Vibration Isolators." In Harris, C. M., ed., *Handbook of Acoustical Measurements and Control*. Reprinted by the *Acoust. Soc. Amer*. Woodbury, NY (1998). ISBN-13: 978-0070268685.

Rivin, E. I., *Passive Vibration Isolation*, *American Society of Mechanical Engineers* (2003). ISBN-13: 978-0791801871.

Chapter 12 Microphones

12.1 Brüel & Kjær: *Measuring Microphones*. Nærum, Denmark (1971). ASIN: B000YT5TFI.

12.2 Dickreiter, M., *Mikrofonaufnahmetechnik*. Leipzig, Hirzel Verlag (2003). ISBN-13: 978-3777611990.

12.3 Wahlström, S., "The Parabolic Reflector as an Acoustical Amplifier." *Journal of the Audio Engineering Society*, vol. 33:6, pp. 418–429 (1985).

12.4 Olson, H. F., *Acoustical Engineering*, 3rd ed. Princeton, NJ: D. van Nostrand (1957). ASIN: B001HDVCMO.

General references

Abbagnaro, L. A., ed., "Microphones." *Journal of the Audio Engineering Society* (1980).

Ballou, G., ed., *Electroacoustic Devices: Microphones and Loudspeakers*. Focal Press Oxford, UK: (2009). ISBN-13: 978-0240812670.

Gayford, M. L., *Electroacoustics: Microphones, Earphones, and Loudspeakers*. London: Newnes-Butterworth (1970). ISBN-13: 978-0408000260.

Rossi, M., and Roe, P. R. W., *Acoustics and Electroacoustics*. Artech House Acoustics Library (1988). ISBN-13: 978-0890062555.

Woram, J. M., and Kefauver, A. P., *The New Recording Studio Handbook*. New York: Elar Publishing (1989). ISBN-13: 978-0914130048.

Chapter 13 Phonograph Systems

13.1 Olson, H. F., *Modern Sound Reproduction*. New York: Van Nostrand Reinhold. (1972). ISBN-13: 978-0442262808.

General references

Anderson, C. Roger, "Fifty Years of Stereo Phonograph Pickups: a Capsule History." *Journal of the Acoustical Society of America*, vol. 77:4, pp. 1320–1326 (April 1985).

Gravereaux, D. W., "Engineering Highlights of the LP Record. *Journal of the Acoustical Society of America*, vol. 77:4, pp. 1327–1331 (April 1985).

Roys, H. E., "Reminiscing—the Stereophonic Record." *Journal of the Acoustical Society of America*, vol. 77:4, pp. 1332–1334 (April 1985).

Stephani, O., "Geschichte von Schallplattentechnik." *Frequenz*, vol. 15, pp. 44–53 (1961).

Temmer, S. F., ed., "Disk Recording vol. 1: Groove Geometry and the Recording Process. *Journal of the Audio Engineering Society* (1980).

Temmer, S. F., ed., "Disk Recording vol. 2: Disk Playback and Testing, *Journal of the Audio Engineering Society* (1980).

Chapter 14 Loudspeakers

14.1 Beranek L. L., *Acoustics*. New York: McGraw-Hill (1954). ISBN-13: 978-0883184943.

14.2 Colloms, M., *High Performance Loudspeakers*, 6th ed. London: Pentech Press (2005). ISBN-13: 978-0470094303.

14.3 Olson, H. F., *Acoustical Engineering*. Princeton, NJ: Van Nostrand (1957). ASIN: B001HDVCMO.

14.4 Suzuki, H., and Tichy, J., "Radiation and Diffraction Effects of Convex and Concave Domes in an Infinite Baffle." Preprint 1675, *67th Convention Audio Engineering Society* (1980).

14.5 Tyrland S., *Construction of a Monitor Loudspeaker* (in Swedish). Chalmers University of Technology, Gothenburg: Dept. of Applied Acoustics (1974).

14.6 Waterhouse R. V., "Output of a Sound Source in a Reverberation Chamber and Other Reflecting Environments." *Journal of the Acoustical Society of America*, vol. 30, p. 1 (1958).

14.7 Button, D. J., and Gander, M. R., "The Dual Coil Drive Loudspeaker." *Proceedings of the 1st AES UK Conference* (1998).

14.8 Borwick, J., *Loudspeaker and Headphone Handbook*, 3rd ed. *Focal Press* Oxford, UK: (2001). ISBN-13: 978-0240515786.

General references

Ballou, G., ed., *Electroacoustic Devices: Microphones and Loudspeakers*. Focal Press (2009). ISBN-13: 978-0240812670.

Gayford, M. L., *Electroacoustics: Microphones, Earphones, and Loudspeakers*. London: Newnes-Butterworth (1970). ISBN-13: 978-0408000260.

Rossi, M., and Roe, P. R. W., *Acoustics and Electroacoustics*. Artech House Acoustics Library (1988). ISBN-13: 978-0890062555.

Cooke, R. E., ed., "Loudspeakers, vol. 1." *Audio Engineering Society* (1978). ASIN: B001ABN690.

Cooke, R. E., ed., "Loudspeakers, vol. 2." *Audio Engineering Society* (1984). ASIN: B001ABNACS.

Gander, M. R., ed., "Loudspeakers, vol. 3." *Audio Engineering Society* (1996). ISBN-13: 978-0937803288.

Gander, M. R., ed., "Loudspeakers, vol. 4." *Audio Engineering Society* (1996). ASIN: B001ABNDTI.

Chapter 15 Headphones and Earphones

15.1 Burkhard, M. D., and Sachs, R. M., "Anthropometric Manikin for Acoustic Research." *Journal of the Acoustical Society of America*, vol. 58, p. 1 (1975).

15.2 Kleiner, M., "Problems in the Design and Use of Dummy-Heads." *Acustica*, vol. 41, p. 3 (1978).

15.3 Gayford, M. L., *Electroacoustics: Microphones, Earphones and Loudspeakers*. London: Newnes-Butterworth (1970). ISBN-13: 978-0408000260.

General references

Ballou, G., ed., *Electroacoustic Devices: Microphones and Loudspeakers*. Oxford, UK: Focal Press (2009). ISBN-13: 978-0240812670.

Borwick, J., *Loudspeaker and Headphone Handbook*, 3rd ed.. Oxford, UK: Focal Press (2001). ISBN-13: 978-0240515786.

Chapter 16 Digital Representation of Sound

16.1 Blesser, B. A., "The Digitization of Audio." *Journal of the Audio Engineering Society*, vol. 26:10 (1978).

16.2 Blesser, B. A., et al., "Digital Processing in Audio Signals." In Oppenheimer ed., New York, NY: *Applications of Digital Signal Processing*. New Jersey: Prentice-Hall. (1978).

16.3 Blesser, B., et al., ed., "Digital Audio." *Proceedings of the AES Premier Conference* (1982).

16.4 Watkinson, J., *The Art of Digital Audio*, 3rd ed. Oxford, UK: Focal Press (2001). ISBN-13: 978-0240515878.

16.5 Pohlmann, K. C., *Principles of Digital Audio*, 6th ed. New York: McGraw-Hill (2010). ISBN-13: 978-0071663465.

16.6 Bosi, M., and Goldberg, R. E., *Introduction to Digital Audio Coding and Standards*. Boston: Kluwer Academic Publishers (2003). ISBN-13: 978-1402073.

16.7 Lehtonen, K., "T-61.246 Digital Signal Processing and Filtering: GSM Code." www.cis.hut.fi/Opinnot/T-61.246/Kutri2003/lehtonen_gsmdoc.pdf (sampled 2006).

16.8 http://www.iis.fraunhofer.de/amm/download/index.html (sampled 2006).

16.9 Kriz, J. S., "A 16-Bit A-D-A Conversion System for High-Fidelity Audio Research." *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 23, p. 1 (1975).

16.10 Lee, F. F., et al., "Floating-Point Encoding for Transcription of High-Fidelity Audio Signals." *Journal of the Audio Engineering Society*, vol. 25, p. 5 (1977).

16.11 Finger, R. A., "On the Use of Computer Generated Dithered Test Signals." *Journal of the Audio Engineering Society*, vol. 35, p. 6 (1987).

16.12 Vanderkooy, J., et al., "Resolution Below the Least Significant Bit in Digital Systems with Dither." *Journal of the Audio Engineering Society*, vol. 32, p. 3 (1984).

Chapter 17 Audio Systems and Measurements

17.1 Olson, H. F., *Modern Sound Reproduktion*. New York, NY: Van Nostrand Reinhold (1972). ISBN-13: 978-0442262808.

17.2 IEC 268-3 *Sound System Equipment. Amplifiers*.

17.3 Zwicker, E., and Fastl, H., *Psychoacoustics*. Berlin: Springer Verlag (1990). ISBN-13: 978-3540526001.

17.4 Kleiner, M., and Lindgren, C., *Objective Characterization of Audio Sound Fields in Automotive Spaces*. Paper 15-007. Audio Eng. Soc. 15th International Conference: Audio, Acoustics & Small Spaces (October 1998).

17.5 Metzler, R. E., *Audio Measurement Handbook*. Portland, OR: Audio Precision (1993). ISBN-13: 978-9994569885.

17.6 Cabot, R. C., "Fundamentals of Modern Audio Measurement." *Journal of the Audio Engineering Society*, vol. 47:9, pp. 738–744, 746-762 (1999).

17.7 Farina, A., "Simultaneous Measurement of Impulse Response and Distortion with a Swept Sine Technique." *Proceedings 110 AES Convention*, Paris (2000).

17.8 IEC 268-13 *Sound System Equipment, Listening Test on Loudspeakers*.

17.9 Diestel, H. G., "Zur Schallausbreitung in reflexionsarmen Räumen." *Acustica*, vol. 12, pp. 113–118 (1962).

17.10 Sköld, A., *Integrative Analyses of Perception and Reaction to Information and Warning Sounds in Vehicles*. Dissertation 2784, Chalmers University of Technology Dept. of Applied Acoustics, Gothenburg, Sweden (2008). ISBN 978-91-7385-103-9.

17.11 Blauert, J., and Jekosch, U., *Sound Quality Evaluation—A Multilayered Problem*. EEA-Tutorium Antwerp, Belgium: (1996).

17.12 Bech, S., "Selection and Training of Subjects for Listening Tests on Sound-Reproducing Equipment." *Journal of the Audio Engineering Society*, vol. 40:7/8 (1992).

17.13 Bech, S., and Zacharov, N., *Perceptual Audio Evaluation—Theory, Method and Application*. Wiley. ISBN-13: 978-0470869239. New York (2006)

17.14 Meilgaard, M. C., et al., *Sensory Evaluation Techniques*, 4th ed. CRC Press (2006). ISBN-13: 978-0849338397.

17.15 ISO 5492 *Sensory Analysis—Vocabulary*

17.16 Nordtest recommendation: *Measurements and Judgments of Sound in Relation to Human Sound Perception*. Copenhagen, Denmark (2001).

17.17 ISO 4121 *Sensory Analysis—Methodology—Evaluation of Food by Methods Using Scales*.

17.18 ISO 4121 *Sensory Analysis—Methodology—Paired Comparison Test*.

17.19 Guilford, J. P., *Psychometric Methods*. McGraw-Hill, ISBN-13: 978-0070993396. New York (1956).

17.20 ITU-R BS.1116 Recommendation: *Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems* (1997).

17.21 ITU-R BS.1283 Recommendation: *Subjective assessment of sound quality—A guide to existing Recommendations (1997)*.

General references

Scott, H. H., "Intermodulation Measurements," *J. Audio Eng. Soc.*, Volume 1, Issue 1, pp. 56–61; January 1953.

Czerwinski, E. et al., "Multitone Testing of Sound System Components. Some Results and Conclusion, Part 1: History and Theory." *J. Audio Eng. Soc.*, Volume 49, Issue 11, pp. 1011–1048; November 2001.

Cabot, R. C., "Comparison of Nonlinear Distortion Measurement Methods" in *Proc. of the 11th International Conference: Test & Measurement*, Audio Engineering Society, New York (May 1992).

Bohn, D., "Audio Specifications." RaneNote 145. *Rane Corporation*, Mukilteo, WA 98275-5000. (also at *http://www.rane.com/note145.html* [sampled May 2011])



This book has free material available for download from the Web Added Value™ resource center at *www.jrosspub.com*

Answers to Problems

1.1

r [m]	$\lvert p\rvert$ [Pa]	$\lvert u\rvert$[mm/s]	ϕ [°]
0.1	1	2.4	29
0.2	0.5	1.2	15
0.4	0.25	0.61	7.8
0.8	0.125	0.30	3.9

1.2

f_{mid} [Hz]	ΔL [dB]
63	6.0
125	5.8
250	5.1
500	1.6
1k	−5.2
2k	4.6
4k	−1.1
8k	1.8

1.3

f_{mid} [Hz]	L_W [dB] re 10^{-12} W
125	105
250	111
500	117
1k	121
2k	123
4k	124

1.4 a) $\alpha = 0{,}34 \pm 0{,}06$
b) Reflection coefficient phase angle $\phi = 2\,k\,x_{min} - \pi$

2.1 $L_{p,tot} = 66$ dB

2.2 $L_{pA,tot}$ = 60 dB. The energy in the 250 and 500 Hz octave bands contributes most to the sound level.

2.3 a) $p_{rms,tot}$ = 0.67 Pa
b) $p_{rms,tot}$ = 0.50 Pa
c) $L_{p,a}$ = 90.5 dB and $L_{p,b}$ = 88.0 dB

2.4 $L_{Aeq,\,T}$ (8 hours) = 83 dB

2.5 Fundamental + harmonics (60, 120, 180, 240, . . .) will be in the 63, 125, 200, 250 Hz, ... third-octave bands.

2.6 a) $L_{p,A}$ = 83 dB, $L_{pA,A}$ = 74 dB, $L_{p,B}$ = 64 dB, $L_{pA,B}$ = 62 dB.
b) $L_{p,A+B}$ = 83 dB, $L_{pA,A+B}$ = 74 dB

2.7

f_{mid} [Hz]	Third-octave band SPL	Octave band SPL
100	72 dB	
125	74 dB	79 dB
160	76 dB	
200	78 dB	
250	80 dB	85 dB
315	81 dB	
400	83 dB	

2.8 T = 3.3 s

2.9

Octave band [Hz]	$L_{p,signal}$ [dB]
125	34
250	36
500	36
1k	37
2k	40
4k	40

3.1 A filter having a 100 Hz bandwidth will not improve the audibility of a signal at 1 kHz since the critical bandwidth is approximately 60–80 Hz.

3.2 The threshold will be raised by 20 dB (Diagram B)

3.3 The 1 kHz-tone must have a sound pressure level higher than 72 dB to be audible.

3.4

f [Hz]	Perceived change [phon]
100	+ 38
400	+ 32
1 k	+ 30
5 k	+ 31
10 k	+ 32

3.5

f_{center} [Hz]	Δf_{cb} [Hz]	Comment
100	≈ 100	≫ 1/3 octave band
200	≈ 100	> 1/3 octave band
400	≈ 110	≈ 1/3 octave band
1 k	≈ 170	≈ 1/4 octave band
2 k	≈ 320	≈ 1/4 octave band
4 k	≈ 650	≈ 1/4 octave band
10 k	≈ 2500	> 1/3 octave band

4.1 The modes (3 0 0), (3 1 0), (1 2 0), (2 2 0), (2 0 1), (2 1 1), (0 2 1) will fit in the 44.5 Hz–56.1 Hz frequency band.

4.2 $$\Delta N = \frac{4\pi V f^2}{c^3}\Delta f$$

Third-octave band [Hz]	Number of modes, ΔN
100	3
125	7
160	14
200	27
250	53
315	≈ 110
400	≈ 220
500	≈ 420
630	≈ 840
800	≈ 1,700
1k	≈ 3,400

4.3 a) $\alpha = 0.10$ b) $\alpha = 0.09$ c) $\alpha = 0.09$

4.4 According to Schroeder's formula (Eq. 4.67), one can use the statistical room acoustics approach from a frequency of 160 Hz and upwards.

4.5

r [m]	$p_{rms,tot}$ [Pa]	Lp [dB]
0.5	1.3	96
1.0	0.81	92
2.0	0.64	90
4.0	0.59	89
8.0	0.58	89

4.6 a) $r = 0.7$ m b) $r = 3.4$ m c) $r = 5.3$ m

4.7 a)

Octave band	500	1k	2k	[Hz]
$A_{S,added}$	7.7–12.0	4.7–9.0	4.7–9.0	[m²S]

b) To reach $A_{S,added}$ 9 sheets of absorber are necessary.

c) Octave band 500 1k 2k [Hz]
T_{Eyring} 0.28 0.25 0.26 [s]

4.8 To obtain the desired added room absorption one can, for example, treat the rear wall by adding 24.5 m² of perforated board. In addition 40 m² of the membrane absorber will be needed. Using these extra absorbers one will reach the $A_{S,added}$ listed below.

Desired		Achieved
Octave band [Hz]	$A_{S,added}$ [m²S]	$A_{S,added}$ [m²S]
125	32.5–47.4	37.9
250	34.4–49.3	39.3
500	21.2–36.1	28.7
1 k	18.3–33.2	18.9
2 k	11.0–25.9	14.4
4 k	3.7–18.6	9.8

4.9 a) L_p at coordinate (4,1,2) m = 83.5 dB.
b) In a diffuse field L_p is the same at all positions, i.e. L_p = 75 dB also at coordinate (4, 1, 2) m.

4.10 The sound pressure level will be 15 dB higher in the corner position than in the middle of the room, i.e L_{pA} = 50 dBA at the corner.

5.1 $\delta = 6.91 / T$

5.2 $C_{80} = 10 \cdot \log\left(e^{\frac{1.1}{T}} - 1\right)$

$D_{50} = 100\left(1 - e^{\frac{-0.69}{T}}\right)$

θ [°]	Δ [dB]	A_V [dB]	A_H [dB]
24	15	4.6	−10.4
20	11	3.8	−7.2
16	8.5	3.2	−5.3
12	6	2.5	−3.5
8	4	1.8	−2.2
4	2	0.9	−1.1
0	0	0	0

6.1 The level difference between the direct sounds from source 2 and source 1 is −6.5 dB and the difference in arrival time is 59 ms. The number of people annoyed by the echo should then be less than 10%.

6.2 To achieve an arrival time difference of less than 20 ms, the time delay Δt must be within the intervals listed below:

Position	Δt [ms]
1	26–46
2	0–20
3	21–41
4	50–70

6.3

r [m]	$H_{without}$ [dB]	$T'_{without}$ [s]	H_{with} [dB]	T'_{with} [s]
10	−1.5	1.8	2.5	1.5
20	−7.5	2.0	−3.5	1.9
30	−11	2.1	−7.0	2.0

7.1

f [Hz]	$\alpha_{0,estimated}$
100	0.16
125	0.18
160	0.26
200	0.33
250	0.44
315	0.48

7.2 $$\alpha = \frac{1}{1+\left(\frac{\omega m''}{2Z_0}\right)^2}$$

7.3 The mass per unit area, $m'' \leq 0.12$ kg/m^2

7.4 Both a membrane absorber at a distance of 36 mm from the wall and the perforated board at a distance of 30 mm from the wall can give a 100 Hz resonance frequency. The perforated board is more suitable in this application because of its sharper and higher absorption peak.

7.5 3 – a, 4 – c, 5 – b, 2 – d, 1 – e

7.6 The perforated brick will act as a resonance absorber, peaking at 382 Hz. In addition there will be resonance peaks at 2.3 kHz, 4.7 kHz, . . . i.e. the frequencies at which the brick thickness is a multiple of half wavelengths. At these frequencies the brick is acoustically transparent because of impedance matching between the two sides of the brick if there is some absorption inside the air space. (Losses at the walls of the brick holes also help because of the half wave resonance.)

8.1 The modal density is $dN/df \approx 0.65$ Hz^{-1} which leads to $\Delta N \approx 15$ in the 100 Hz and $\Delta N \approx 120$ in the 800 Hz third octave bands respectively.

8.2 The rms value of the acceleration is approximately 0.012 m/s^2 at a sound pressure level of 70 dB.

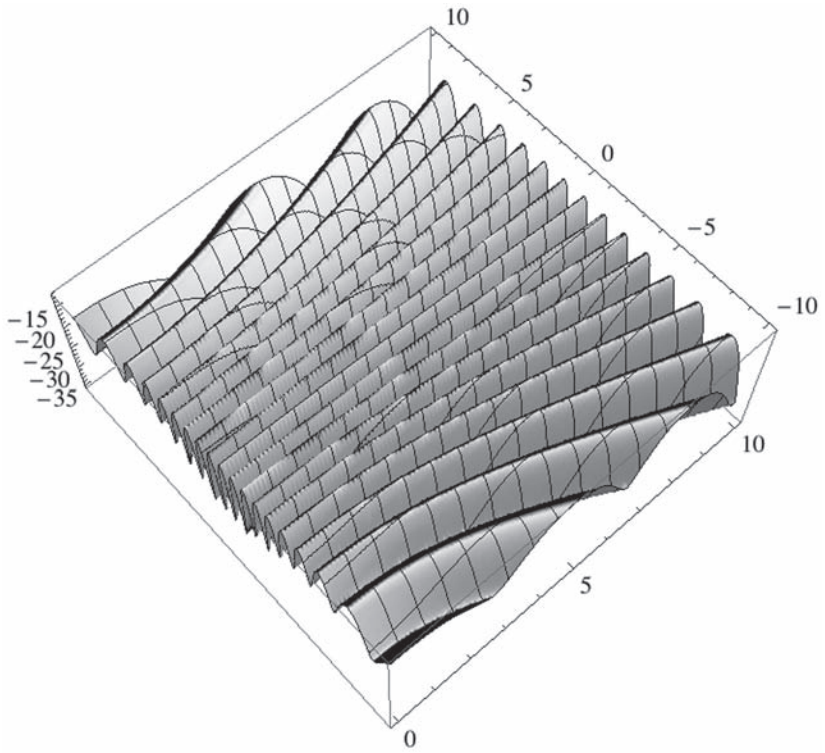
8.3 a) The modal density is approximately given by

$$\frac{dN}{df} \approx \frac{4\pi V f^2}{c^3} + \frac{\pi S f}{2c^2} + \frac{L}{8c} \qquad [\text{Hz}^{-1}]$$

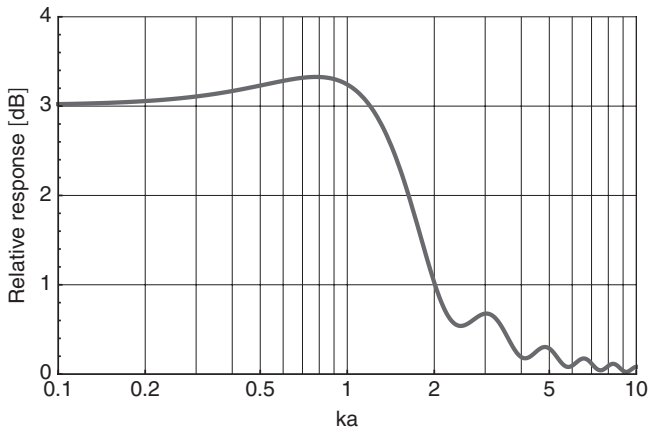
f [Hz]	dN/df [Hz^{-1}]	Comment
100	0.34	Not audible
200	1.0	Not audible
500	5.4	Not audible
700	10	Not audible

b & c) The modal densities are $dN/df \approx 1.4$ Hz^{-1} for the sheet of steel (audible coloration) and $dN/df \approx 4.9$ Hz^{-1} for the gold foil (no audible coloration).

9.1


10
5
0
−5
−10
−15
−20
−25
−30
−35
10
5
0

9.2


Relative response [dB]
4
3
2
1
0
0.1
0.2
0.5
1
2
5
10
ka

9.3

Octave band	0.5 k	1k	2k	4k	[Hz]
Radiation ratio	0.05	0.3	1	1	

9.4 $\Delta L_W \approx -20$ dB

9.5 Port power level ≈105 dB at 34 Hz and port flow noise level is ≈26 dB peaking at about 12 Hz.

10.1 $R_{tot} = 48$ dB

10.2 $L_{pA} = 27$ dB

10.3 a) Same L_p in the room without the TV set in both cases.
b) lower L_p in the room without the TV set if the TV set is in the small room.

10.4 The fundamental resonance frequency of the wall is 106 Hz. The critical frequencies for the two sheets are in the intervals 3.6–4.4 kHz (9 mm sheet) and 2.5–3.1 kHz (13 mm sheet). To increase the transmission loss one must lower the fundamental resonance frequency, for example by separating the two sheets. In addition it is advisable to have separate battens for each wall so that the wall will act more like a theoretical double wall.

10.5 The sound level in room 2 is $L_{pA} = 42$ dB.

10.6 The fundamental resonance frequency is 94 Hz. The critical frequencies are 1.2 kHz and 3 kHz respectively. The modes of the cavity are at the frequencies 95, 190, 200, 222, 275 Hz etc.

11.2 $f_{res} \approx 48$ Hz

11.3 $f_{res} \approx 4.7$ Hz

11.4 $f_{res} \approx 18$ Hz

12.1 a) The equation of motion: $F(x) = M\frac{d^2x}{dt^2} + R\frac{dx}{dt} + \frac{1}{C_m}x$

The mechanical impedance: $\underline{Z}_M(\omega) = j\omega M + R + \frac{1}{j\omega C_M}$

The resonance frequency: $f_{res} = \frac{1}{2\pi\sqrt{MC_M}}$

b) The output voltage is proportional to the sound pressure:
1) below f_0 (for example carbon and condenser microphones)
2) at f_0 (for example dynamic microphones)
3) over f_0

12.2 a) $\tilde{e} = \dfrac{BlS}{M\omega_0\sqrt{\left(\dfrac{\omega_0}{Q}\right)^2 + \left(\dfrac{\omega}{\omega_0} - \dfrac{\omega_0}{\omega}\right)^2}}\tilde{p}$

b)

f [Hz]	$\tilde{e}_{rms}$ [mV]		
	$Q = 1$	$Q = 0.1$	$Q = 0.01$
75	1.4	0.52	0.055
150	3.1	0.55	0.055
300	5.5	0.55	0.055
600	3.1	0.55	0.055
1.2 k	1.4	0.52	0.055
2.4 k	0.7	0.43	0.055

12.3 a) $Q_1 = Q_2$

b) $Q_1 = -k\,Q_2$; k should be larger than 0 but smaller than 1.

c) $Q_1 = -Q_2$

12.4 Directivity index $DI = 4.8$ dB

12.5

f [Hz]	$\tilde{e}_{rms}$ [mV]
0.5 k	0.016
1 k	0.022
2 k	0.009

12.6 The signal-to-noise ratio is 42 dB.

14.1 a) The resonance frequency is 71 Hz.

b) The diaphragm velocity is $\tilde{u} = \dfrac{Bl}{\sqrt{R^2 + \dfrac{M}{C_M}\left(\dfrac{\omega}{\omega_0} - \dfrac{\omega_0}{\omega}\right)^2}}\tilde{i}$ and the displacement is $\tilde{x} = \dfrac{\tilde{u}}{\omega}$

c)

f [Hz]	$\tilde{u}_{rms}$ [m/s]	$\tilde{x}_{rms}$ [mm]
17	0.89	8.0
36	2.1	9.4
71	7.5	16.7
142	2.1	2.3
285	0.89	0.49

14.2 The resonance frequency is 78 Hz.

14.3 a)

f [Hz]	W [mW]
17	0.6
71	940
285	160

14.4 a) Region I—below f_0 the response is compliance controlled, L_p increases by 12 dB/octave.
Region II—the peak at f_0 is determined by the damping of the mass-compliance resonance.
Region III—above f_0 the response is controlled by the diaphragm mass and L_p will be constant except that, because of the increased directivity at higher frequencies, there will be a slight increase with frequency.
Region IV—interferences between the contributions from various parts of the diaphragm because of bending wave resonance mode patterns.
Region V—radiation resistance no longer increases with frequency (remains constant) so the radiated power decreases and thus also L_p.

b) Regions I–III—choice of diaphragm mass, suspension mechanical compliance and loudspeaker box acoustic compliance, and losses will determine the frequency and Q-value of the fundamental resonance.
IV—damping bending wave resonance for example by using a diaphragm having a viscoelastic layer.
V—a smaller diaphragm will lead to a higher radiation resistance cut-off.

14.5 The resonance frequency is 44 Hz.

14.6 The resonance frequency is $f_0 = \frac{1}{2\pi}\sqrt{\frac{\frac{1}{C_M} + \frac{\rho c^2}{V}S^2}{M}}$

The absorption coefficient is $\alpha = \frac{4a}{(a+1)^2 + b^2}$

where $a = \frac{R_M}{SZ_0}$

and $b = \frac{1}{SZ_0\left(\omega M_M - \frac{1}{\omega}\left(\frac{1}{C_M} + \frac{\rho c^2}{V}S^2\right)\right)}$

14.7 At frequencies above approximately 74 Hz: W_{max} = 600 mW

At frequencies below approximately 74 Hz : $W_{max} \approx 0.23\frac{\omega^4}{\omega^2 + 1.8\cdot 10^7}$

14.8 The sound pressure level is 63 dB.

15.1 $\underline{p} = \frac{S\rho c^2}{j\omega V R_{VC}} \frac{Bl}{j\omega M_M + \frac{1}{j\omega C_M} + R_m}\underline{e}$

15.2 The output voltage will be about 0.05 mV.

15.3 The frequency response will have a +6dB/octave characteristic if the near field pressure close to one side of the headphone was adjusted for flat frequency response.

15.4 The frequency response at the eardrum will be flat up to about 2 kHz.

17.1 THD ≈ 14%

17.2 THD ≈ 13%

17.3 THD ≈ 7.5% and DFD ≈ 1.8%

Index

A

Absorbers, 165–196
  absorption coefficient, 165–167
  absorption data examples, 190–194
  audience absorption, 182, 183–184
  diffusers and, 188–190
  generally, 165
  porous. *See* Porous absorbers
  problems to solve, 194–196
  reflectors and, 185
  resonance, 177–182
  variable, 182, 183
Absorption coefficient, 165–167
  absorption area, using, 167
  impedance tube, measurement using an, 166–167
  reverberation chamber, measurement of *A* using, 167
  sound-absorption properties, characterization of, 165–166
Absorption data examples, 190–194
Acoustics. *See* Room acoustics
Acoustic efficiency, flow and, 227–229
Acoustic environment, 341–342
Acoustic perfume, 136
Acoustic reflex, 52
Acoustic uncertainty principle, 119
Acoustical audience area, 183
Acoustical components and circuits, 18–23
  circuits, acoustical, 22–23
  compliance components, acoustical, 19–20
  mass components, acoustical, 20–22
Acoustical impedance, 9
Acoustically blue sky, 25
Active acoustic variation, 144
Active noise cancellation, 64
A/D and D/A conversion, additional problems in, 365–366
ADC. *See* Analog-to-digital converter
Aerodynamic short circuit, 222, 319
Aerodynamic shortcircuit/aerodynamic cancellation, 27
AES. *See* Audio Engineering Society
Age-related hearing loss (ARHL), 61, 62
AI. *See* Articulation index
Airborne sound, insulation against, 231, 232
Aliasing, 356
Ambisonics, 159, 387
Amplification, phonograph systems, 301
Amplitude frequency response, 391–392
Analog-to-digital conversion, 354–357
Analog-to-digital converter (ADC), 269
Anatomy of the ear. *See* Ear, components of
Anechoic chamber, 54, 99, 111–112, 399, 400, 402
Anechoically recorded sound, 160
Angles of incidence, 155
Annoyance, 136
Anti-aliasing filter, 356
Applied room acoustics, 82
ARHL. *See* Age-related hearing loss
Array loudspeakers, 326–328
Array microphones, 287–289
Articulation, 76
Articulation index (AI), 77, 137
Articulation loss of consonants (%ALcons), 77

Attack and rise time, 89
Attenuation due to interference, 184
Audience absorption, 182, 183–184
Audio channels and listening modes, 378
Audio characteristics, metrics for, 389–390
Audio Engineering Society (AES), 390
Audio signals, 31–46
  effective value/measurement uncertainty, 40
  equivalent level, 43
  filters/frequency bands, 36–40
  level concept, 35–36
  level contributions, addition of, 42
  noise, 34–35
  problems to solve, 44–46
  reference levels, 40–42
  signals and the jω-method, 31–33
  sine-wave signals, 34
  spectrum and time history, 31
  weighted sound pressure levels, 43
Audio sound character and quality, 408–409
Audio systems and measurement, 377–417
  amplitude frequency response, 391–392
  audio channels/listening modes, 378
  audio sound character and quality, 408–409
  binaural sound reproduction, 384–385, 386
  compression, 407
  difference frequency and intermodulation distortion, 397–398
  directivity, 405, 406
  expert panels, 413
  FM distortion, 398–399
  frequency response, spectogram, and wavelet analysis, 403–405, 406
  harmonic distortion, 396–397
  impedance, 407–408
  impulse response analysis, 402–403
  listening tests/listeners, 409–413, 413–415
  loudspeaker-headphone compatibility, 385, 387
  metrics for audio characteristics, 389–390
  monophonic and monaural, 378
  multichannel sound reproduction, 387–389
  multitone distortion measurement, 398
  noise level and signal-to-noise ratio, 391
  nonlinear distortions, 393–396
  phase response and group delay, 393
  physical audio metrics, 390
  problems to solve, 415–417
  sensitivity, 406
  special chambers, measurements using, 399–402
  stereo using headphones, 382–384
  stereophony and phantom sources, 378–382
Audiometry and audiogram, 59–61
Auralization, 112, 159–161
Autocorrelation function, normalized, 34–34

B

Background noise, 137
Baffles, 319–320
Balanced spectrum design, 247–248
Band-limited data for noise, 44
Bandpass loudspeaker box design, 23, 303
Bandwidth, critical, 65
Barriers, 186–187
Basilar membrane, inner ear, 53–54
Bass ratio (BR), 132
Bass traps, 180
Bass-reflex loudspeaker, 323–324
Bending waves in sheets, 201, 203–207
  critical frequency/coincidence frequency, 204
  dispersive waves, 204
  forced bending wave, 206
  free bending waves, 205
  incident sound wave, 205, 206
  mechanical input impedance, 206
  modal densities of wave fields, 206–207
  phase and group velocities, 204
  slow wave/fast wave, 204
  wave conversion, 206
Bias error, 47
Binaural hearing, 47, 67–70
  localization using interaural time and level differences, 67–68

localization using only spectral and level changes, 68–70
Binaural sound reproduction, 384–385, 386
Binaural spatial impulse response (BSIR), 120
Block floating-point quantization, 369–370
BR. *See* Bass ratio
Breakup, 308
Brilliance, 132
BSIR. *See* Binaural spatial impulse response
Butterworth-type response, 324

C

CAD. *See* Computer aided design
Capacitive microphones, 280–283
bias voltage, 280–281
directivity and, 282, 283
electret microphones, 271, 272, 276, 281, 282
micromachined microphones (MEMS), 282
phantom feed principle, 283
variations in capacitance, 282–283
Cardioid microphone, 274
Cartridges, playback quality and, 299–301
cartridge types, common, 300–301
electroacoustic systems, 300
stylus arms, 299–300
stylus tips, 299
CD. *See* Compact disc
Center-of-gravity time, 131
Ceramic microphones. *See* Piezoelectric microphones
Characteristic impedance, 8–9
Circumaural headphones, 341, 349
Clarity, 130
Cochlea, inner ear, 52, 53, 63, 66
Codecs, uniform, and nonuniform quantization, 366–368, 366–370
block floating-point quantization, 369–370
compression, 366–368
floating-point quantization, 369
lossless compression, 367, 368
lossy compression, 367
nonuniform coding, 368
Coding, nonuniform, 368
Coincidence frequency, 204
Coloration, 114, 119–120, 148
Column loudspeaker, 327
Comb filter effect, 119–120
Compact disc (CD), 353
Compliance-controlled diaphragm, 277, 278
Compression, 366–368, 407
Computer aided design (CAD), 153, 154, 155
Condenser microphones, 201, 280
Conductive hearing loss, 59
Confidence interval, variance and, 40, 41
Consonance, 67
Constant group delay, 393
Constrained viscoelastic layers, 212–213, 214
Corner transmission loss, 246
Correlated/uncorrelated sounds, 42
Cracks and other leaks, 244–245
Critical band filters, 65
Critical bands, masking and, 64–66
Critical bandwidth, 65
Critical frequency, 204, 233
Cross-contraction number. *See* Poisson's ratio
Cross-contraction phenomena of solids, 200
Crossover networks, 325, 326
Cross-talk
cancellation, 112, 159
compensation, 385
Crystal microphones. *See* Piezoelectric microphones

D

DAC. *See* Digital-to-analog converter
Dampers, 259
Damping by sand and other lossy materials, 213, 214
Damping by viscoelastic layers, 209–213
constrained layers, 212–213, 214
free layers, 210, 211
viscoelastic materials, 210
DAT. *See* Digital audio tape
dBHL (hearing level), 60

Delay
group, 393
vs. level differences, 116, 117–118
lines, 287
units, 112
Delta modulation (DM) converters, 363, 364
Dense mode region in a room, 104
Derived metrics, 390
DFD. *See* Difference frequency distortion
Diaphragm
materials, loudspeakers, 314–315
mechanical properties, microphones, 277–278
shape, loudspeakers, 312–314
supports/surrounds, loudspeakers, 315–317
Difference frequency distortion (DFD), 397–398
Diffuse sound field(s), 87–88
wave theory and, 106–107
Diffusers, 149, 188–190
Digital audio tape (DAT), 353
Digital representation of sound, 353–376
sampling and digitization, 353–357
Digital-to-analog converter (DAC, D/A), 354
Dipole sound sources, 26–28, 226
Dirac pulse, 7
Direct radiator loudspeakers, 303
Direct sound, 148
Direct transmission, 232
Directivity, 405, 406
balloon, 405
loudspeakers and, 304–308
Directivity of microphones, 273–276
directivity characteristics, 275
directivity function, 273, 274
distance factor and, 276
random energy efficiency metric, 276
Disc cutting, 293–296
cutting heads, 293–295, 296
cutting styll, 296
disk manufacture, 296
Distance law, 11
Distortion(s), 269–270
FM, 398–399
harmonic, 270, 396–397
intermodulation, 270
linear, 269–270
linearity and harmonics, 66–67
multitone, measurement of, 398
nonlinear, 270, 330–331, 393–396
Dither, 360–363
DM. *See* Delta modulation (DM) converters
Doors and windows, 242, 243–244
Double-wall fundamental resonance, 238–240, 241
Downsampling, 357
Dual-slope reverberation curves, 125–126
Dynamic absorbers, 257
Dynamic microphone, 283–284
Dynamic range, 269, 390

E

Ear, components of, 47–54
anatomy of the ear, 48
inner ear, 52–54
middle ear, 51–52
outer ear, 49–51
Ear bud, 341
Ear defenders, 63–64
Early decay time (EDT), 131, 147
Early reflections, 146, 149
Earphones. *See* Headphones and earphones
EC Directive on Noise, 61, 63
Echo, 118–119
EDT. *See* Early decay time
Effective value
level contributions and, 42
measurement uncertainty and, 40, 41
Eigenfrequencies
complex, 104–105
Eigenmodes and, 101–102, 103
Electret microphones, 282
Electric output impedance, 271
Electrodynamic headphones, 346–347

Electrodynamic loudspeakers, 308
Electrodynamic microphones, 283–286
  dynamic microphones, 283–284
  mass-controlled microphone diaphragm, 286
  resistance-controlled moving system, 284
  ribbon microphones, 285
  velocity microphones, 285
Electromagnetic drivers, loudspeakers, 310–319
  diaphragm materials, 314–315
  diaphragm shape, 312–314
  diaphragm supports/surrounds, 315–317
  voice coil/magnet air gap, 317–319
Electromagnetic headphones, 342–345
Electromechanical analogies, 259
Electromechanical analogies, vibration isolation theory and, 263–266
  mechanical and acoustical analogies, 263
  preventing body vibration due to vibration of foundation, 263–265
  preventing force from acting on foundation, 263–265
Electronic architecture, 136, 157–158
Electronic reverberation enhancements systems, 143
Electrostatic headphones, 348–349
Electrostatic loudspeakers, 201, 304, 332–335
Electrostatic microphones. *See* Capacitive microphones
Energy exposure, hearing loss and, 59
Energy-equivalent A-weighted sound pressure level, 43
Equal loudness contours, 55
Equation
  of continuity, 2–3
  of motion, 3–4
  wave equation/solutions to, 5–8
Equivalent level, 43
Euler's formula, 33
Excessive noise, criteria for, 61, 63
Excessive sound exposure, effects of, 58–64
  audiometry and audiogram, 59–61, 62
  ear defenders, 63–64
  excessive noise, criteria for, 61, 63
  hearing impairment/hearing loss, 59
  impulsive noise, 63
  threshold shift, temporary and permanent, 58–59
  tinnitus, 61
Excitation strength, loudness and, 56
Expert panels, 410, 413
Explosion-type noise, 63
Eyring's formula, 93

F

Fast Fourier transform (FFT), 392
Fermat's principle, 85
FFT. *See* Fast Fourier transform
Field STC (FSTC), 234
Filter(s)
  critical band, 65
  frequency bands and, 36–40
  parallel, 65
  sound pressure levels and, 43
  weighting, 56
Filter bank methods, 371
Finite impulse response filters, 382
Fitzroy's formula, 153
Flanking transmission, 232
Flare rate, 324
Flash converters, 365
Floating floor, 249
Floating-point quantization, 369
Flow
  resistance, 169
  sound generation by, 226–229
Fluctuation(s), 57, 58
Flutter echo, 119, 148
FM. *See* Frequency modulation
FM distortion, 398–399
Force isolation, 257
Forced bending wave, 206
Fourier theory, 5, 6
Fourier transforms, 371, 372, 402
FR. *See* Frequency response

Free bending waves, 205
Free resonance frequency, 323
Frequency bands, 36–40
Frequency modulation (FM), 398
Frequency response (FR), 392, 403–405
Frequency space, 102
Fresnel zones, 83–84, 85
Fundamental resonance frequency, 304

G

Gaussian-type wavelet, 405
Geographical attenuation, 12
Geometrical room acoustics, 81, 82–87
  Fresnel zones and, 83–84, 85
  ray tracing and mirroring, 84–87
  wavelength criterion, 82–83
Gradient microphones, 286–287
Group delay, 393

H

Hard room, 98
Hardware convolver, 160
Harmonic(s)
  distortion, 270, 396–397
  linearity and, 66–67
Headphones and earphones, 339–351. *See also* Loudspeaker-headphone compatibility; Stereo using headphones
  acoustic environment, 341–342
  electrodynamic headphones, 346–347
  electromagnetic headphones, 342–345
  electrostatic headphones, 348–349
  vs. loudspeakers, 339–341
  noise-canceling headphones, 349–350
  piezoelectric headphones, 348
  problems to solve, 350–351
Head-related coordinate system, 47, 48
Head-related impulse responses (HRIR), 160
Head-related transfer functions (HRTFs), 49, 50, 69, 160
Headroom, 358
Hearing, 47–70. *See also* Ear, components of
  binaural, 67–70
  disability, 61
  distortion, linearity, and harmonics, 66–67
  excessive sound exposure and, 58–64
  generally, 47
  loss, 59, 60, 63
  masking and critical bands, 64–66
  problems to solve, 79–80
  psychophysical sensitivity of, 35
  threshold, 54
Hearing, dimensions of, 54–58
  loudness/loudness level, 54–57
  pitch and timbre, 57–58
Hearing level (dBHL), 60
Heat conduction, 12
Helmholtz absorbers/resonators, 152, 153, 177, 178–180
Helmholtz resonance frequency, 333
Hemi-anechoic chamber, 99
High fidelity loudspeaker driver, 325
High order filters, 356
High-directivity voice warning system, 26
High-fidelity, 377
Hole in the middle effect, 381
Hooke's law, 208
Horn loudspeakers, 303, 308, 324–325
HRIR. *See* Head-related impulse responses
HRTFs. *See* Head-related transfer functions
Huygen's principle, 83
HVAC noise, 135, 137, 231
Hybrid models, room acoustics modeling, 156, 157

I

IACC. *See* Interaural cross-correlation
Idling noise, 364–365
IEC. *See* International Electrical Commission
IED 268-13 listening room, 399
ILDs. *See* Inter-ear level differences
IMD. *See* Intermodulation distortion
Impact sound
  insulation against, 248–251

pressure, 231
pressure level, 248, 249
Impaired hearing, 59, 60, 61, 63
Impedance, 8–10, 407–408
characteristic and sound field impedances in a plane wave, 9–10
converter, 279
definitions of, 8–9
electric output, microphones and, 271
mobility and, 259
relationships, summary of, 9
sound field impedance in a spherical wave, 10
tube, measurement using, 166–167
Impulse response, 7
Impulse response analysis, 402–403
Impulsive noise, 63
Incidence
normal sound, 13–15
oblique sound, 15–16
Infinite impulse response filters, 382
Infinite media. *See* Wave types in infinite media
Inhomogenous media. *See* Sound propagation in inhomogeneus media
Insertion loss, 260
Insulation
against airborne sound, 231, 232
against impact sound, 248–251
against structure-borne sound, 252–253
Interaural cross-correlation (IACC), 126, 130–131
Inter-ear level differences (ILDs), 68
Inter-ear time differences (ITDs), 68
Intermodulation distortion (IMD), 270, 397–398
International Electrical Commission (IEC), 390
Inversion, 24
Isodynamic headphone, 346, 347
ITDs. *See* Inter-ear time differences

J

Journal of the Audio Engineering Society, 353
jω-method, 6, 31–33

K

Kundt's tube, 29, 166

L

Late reverberation, 147
Lateral energy fraction, 131
Lateralization error, 68
Law of the first wave front, 149. *See also* Precedence effect
Leakage, 396
Leaks. *See* Cracks and other leaks
Least significant bit (LSB), 358
LEDE. *See* Live End Dead End (LEDE) technique
Level
concept, 35–36
contributions, addition of, 42
Linear distortion, 269–270
Linear single-degree-of-freedom systems, 260–263, 266, 267
Linearity and harmonics, distortion and, 66–67
Listening modes and audio channels, 378
Listening room, 399
Listening tests, 409–415
common types of, 410–412
expert panels, 413
information to be reported, 414–415
listeners, selecting, 412–413
planning/preparation of, 413–414
test session, 414
Live End Dead End (LEDE) technique, 152
Live room, 98
Localization, 67–70
using interaural time and level differences, 67–68
using only spectral and level changes, 68–70
Longitudinal waves, infinite media and, 197–199
Loss factor, reverberation time and, 105–106
Lossless compression, 368
Lossy compression and perceptual coding, 370–376

audio quality of perceptual coding/decoding, 375–376
filter bank methods, 371
percepual coding, 373–374
stereo and multichannel coding, 374–375
time-to frequency mapping, 371
transform methods, 371–373
Loudness compensation, 55
Loudness/loudness level, hearing and, 54–57
Loudspeaker(s), 303–337
array, 326–328
box design, bandpass, 23
components of, 7
early, viscoelastic compounds and, 211
efficiency of, 308–309
electrodynamic drivers, 310–319
electrostatic, 201, 332–335
enclosures for, 319–324
frequency response of, 309–310
-headphone compatibility, 385, 387
vs. headphones and earphones, 339–341
horn, 303, 324–325
multiple direct radiator systems, 325–326, 327
nonlinear distortion, 330–331
nonlinearities, electronic compensation of, 331
omnidirectional characteristics, 7, 8
plasma, 217, 218
problems to solve, 335–337
radiation and directivity, 304–308
room effects, 328–329, 330
transient response, 329, 330
Loudspeaker enclosures, 319–324
baffles, 319–320
loudspeaker boxes, 320–324
Low-pass cutoff frequency, 356
LSB. *See* Least significant bit

M

Masking and critical bands, 64–66
Mass controlled diaphragm, 277, 278
Mass-spring system, simple, 260–263
Maximally flat response, 324
MD. *See* Modulation damping
Measurement. *See* specific subject
Measurement uncertainty, effective value and, 40
Mechanical compliance, 277
Mechanical impedance, 9
Mechanical mobility, 259
Mechanical radiation
impedance, 219
resistance, 218
Mel, units of, 57
Membrane
absorber, 177
vibration, 201
MEMS. *See* Micromachined microphones
Metrics. *See* specific subject
Micromachined microphones (MEMS), 282
Microphones, 269–292
bidirectional, 274
capacitive, 280–283
cardioid, 274
diaphragm mechanical properties, 277–278
directivity of, 273–276
dynamic range, 269
electroacoustical conversion principles, 271–272
electrodynamic, 283–286
frequency response, noise, and distortion, 269–271
omnidirectional, 274
piezoelectric, 279–280
preamplifiers for, 270, 272
problems to solve, 290–292
resistance, 278–279
self noise from, 270–271
sensitivity of, 271
sound field sensing principles, 273
super-directional, 286–289
wind noise and, 289–290
Middle ear, 51–52
Mirror image method, 155, 156

Mirroring. *See* Ray tracing and mirroring
Missing fundamental, the, 58
Mobility theory, 259
Modal density, 102, 103–104
Modulation, 57, 58, 140
Modulation damping (MD), 77, 140
Monochromatic sound source, 102
Monophonic and monaural audio systems/ measurement
Monopole, 7, 8, 304
Monopole sound sources, 226
Mounted resonance frequency, 323
Multichannel sound reproduction, 387–389
Multilayer floor, 249, 251
Multiple direct radiator loudspeaker systems, 325–326, 327
Multiple-panel partitions, transmission loss of, 238–244
  doors and windows, 242, 243–244
  double-wall fundamental resonance, 238–240, 241
  resilient skin, 240, 241–242
  suspended ceiling, 240
Multitone distortion measurement, 398
Music, rooms for, 143–150
Music Transmission Index, 398

N

NC. *See* Noise criteria
Newton's equation of motion, 3–4
Noise, 34–35. *See also* White noise; Wind noise, microphones and
  -canceling gradient microphones, 286–287
  cancellation, active, 64
  -induced hearing loss, 58
  legislation, 61, 63
  level, 391
  pulses, 77
Noise criteria (NC), 135, 138
Noise reduction (NR), 232, 234
Noise-canceling headphones, 349–350
Noise-shaping, 365
Noisy environments, 137–138
Nondirect radiating loudspeakers, 303
Nonlinear distortions, 66, 270, 393–396
Nonlinearities
  in A/D and D/A converters, 366
  electronic compensation of, 331–332
Nonresonant absorbers. *See* Porous absorbers
Nonuniform coding, 368
Normal sound incidence, 13–15
Normal temperature profile, 25
Normalized autocorrelation function, 34
Normalized impact sound pressure level, 248
Notch filter, 396
NR. *See* Noise reduction
Nyquist frequency, 354

O

Oblique sound incidence, 15–16
Occupational hearing loss, 60
Octave band signal components, 44
Open-back headphone, 346, 349
Optical intensity modeling, 154
Optical ray tracing, 154
Organ of Corti, inner ear, 53
ORTF recording technique, 385, 387
Ossicles, middle ear, 51, 52
Outer ear, 49–51
Oversampling, 356

P

Paired comparison test, 410, 411–412
Parallel analyzer, 65
Parallel filters, 65
Particle velocity, 2–3
Passive acoustic variation, 144
PCM. *See* Pulse code modulation
Peakiness of a resonance curve, 105
Perception of speech. *See* Speech intelligibility
Perceptual coding, 367, 373–374. *See also* Lossy compression and perceptual coding
Perceptual coding/decoding, audio quality of, 375–376

Perceptually optimized noise-shaping, 365
Permanent threshold shift (PTS), 58
Phantom feed principle, 283
Phantom power, 271
Phantom sources, 117
  stereophony and, 378–382
Phase response, 393
Phonographic systems, 293–301
  amplification, 301
  cartridges, 299–301
  disc cutting, 293–296
  playback system, 296–298
  tone arms, 297–298
Physical acoustics modeling, 154
Physical audio metrics, 390
Piezoelectric headphones, 348
Piezoelectric microphones, 279–280
Pitch, timbre and, 57–58
Plane wave(s), 2
  characteristic and sound field impedances in, 9–10
  tube, 29, 166
Plasma loudspeakers, 217, 218
Point source, 7
Poisson's ratio, 199, 200
Porous absorbers, 168–176
  constriction to air and, 173, 174–175
  disadvantages and problems, 171, 172
  flow resistance and, 169–170
  sound-absorption coefficients for, 175–176
Power density level, 35–36
Precedence effect, 115–116
Presbycusis, 61
Problems to solve
  absorbers/reflectors/diffusers, 194–196
  audio signals, 44–46
  audio systems/measurement, 415–417
  headphones/earphones, 350–351
  hearing and voice, 79–80
  loudspeakers, 335–337
  microphones, 290–292
  room acoustics, basic, 107–110
  room acoustics planning/design, 161–163
  sound, 28–29
  sound isolation, 253–256
  sound radiation/generation, 229–230
  spatial sound perception, 132–133
  vibration isolation, 268
  waves in solids, 215–216
Product Sound Quality, 137
Propagation
  constant, 6
  losses, 12–13
  speed of sound, 5
Psychoacoustic experiment, schematic representation, 47, 48
Psychoacoustic tests, 47, 111
Psychological room acoustics, 81, 82
Psychophysical sensitivity of hearing, 35
PTS. *See* Permanent threshold shift
Public address system, 158
Pulse code modulation (PCM), 353, 360, 363, 364
Pulse-density modulation, 365
Pulse-width modulation, 365

Q

QRD. *See* Quadratic residue sequence
Quadratic residue sequence (QRD), 189
Quadropole sound sources, 26–28, 226
Quantization, 357–365
  block floating-point, 369–370
  dither, 360–363
  error, 357
  floating-point, 369
  methods, 363–365
  noise, 271, 358
  resolution and quantization noise, 357–360
Quasilongitudinal waves in sheets, 200–201
Quiet environments, 137–140
Q-value, 105

R

Radiation

impedance, 328
load, 309
loss factor, 220
loudspeakers and, 304–308
ratio, 219–220
Radius of curvature, 25
Random error, 47
Rate of decay, 89
Ray tracing and mirroring, 84–87
corners of intersecting surfaces, 86
method, 155, 156
opposing mirrors, 86
ray reflection order, 85
in rectangular room, 87
Rayleigh type of reflection, 17–18
Reactance, impedance and, 8
Reactive near-field, 222
Real ear attenuation, 63, 64
Real systems, vibration isolation and, 266–267
Receiving room, 231
Reciprocity, 102
Recording studios, 150
Reference levels, 40–42
Reflected sound, sequences or, 120–121, 122
Reflection and transmission at boundaries, 13–18
normal reaction, 16–17
normal sound incidence, 13–15
oblique sound incidence, 15–16
Rayleigh type of reflection, 17–18
Reflection point, 84
Reflectors, 185
Relaxation phenomena, 12
Resilient skin, 240, 241–242
Resistance
control, 277
impedance and, 8
microphones, 278–279
Resonance
frequencies of room, 101
fundamental resonance frequency, 304
Resonance absorbers, 177–182
air-brick, 177, 181
Helmholtz, 177, 178–180
slit, 177
translucent, 181, 182
using mechanical resonance, 181
Resonator panel, 180
Reverberation
chamber, 98, 167, 399
curves, 94
enhancement, 157–158
radius, 97
time, 89–90, 144
units, 112
Reverberation time and subjective preference, 121, 123–126
dual-slope reverberation curves, 125–126
early/late reverberation, 124–125
Ribbon microphone, 285
Ribbon-type dynamic loudspeakers, 328
Ripple, 392
Rocking mode resonances, 347
Roll-off rate, 356
Room(s)
correction, 234
gain, 129
impulse response, 120
for music, 143–150
Room acoustic response, tools for prediction, 153–157
methods used, 154
models used, 155–156
Room acoustics, basic, 81–110
geometrical. *See* Geometrical room acoustics
problems to solve, 107–110
statistical room acoustics, 87–99
wave theoretical approach, 99–107
Room acoustics, metrics for, 128–132
brilliance, 132
center-of-gravity time, 131
clarity, 130
definition, 129

early decay time (EDT), 131
interaural cross-correlation (IACC), 130–131
lateral energy fraction, 131
reverberation time, 129
strength index, 129
warmth, 131
Room acoustics, planning and design, 135–163
basic requirements, 136–140
fundamentals of planning, 140–153
problems to solve, 161–163
Roughness, perception of, 58, 397
Running music, 135

S

Sabine, W. C., 93
Sabine's formula, 90–94, 153
absorption area, 92–93
equivalent absorption area, 93
expression of, 94
mean-free-path length, 91
room absorption, 93
Sabine absorption, 93
Sample-and-hold circuit (SHC), 363, 365–366
Sampling frequency, 353, 354, 392
SAR. *See* Successive approximation register
Scalograms, 404
Schroeder frequency, 107, 147, 233
SDM. *See* Sigma-delta modulation (SDM) converters
SDOF systems, 260–263, 266, 267
Semantic differential test, 410–411
Sending room, 231
Sensitivity, 405
Sensorineural hearing loss, 59
Sequences of reflected sound, 120–121, 122
SHC. *See* Sample-and-hold circuit
Shear modulus, 200
Short-time Fourier transform (STFT), 404
Sigma-delta modulation (SDM) converters, 363, 364
Signals and the jω-method, 31–33
Signal-to-noise ratio (S/N), 76, 137, 270, 391
SII. *See* Speech intelligibility index
SIL. *See* Speech Interference Level
Sine-wave
signals, 34
voltage generator, 32
Single-panel partitions, transmission loss of, 234–238
SIR. *See* Spatial impulse response
Slew-rate, 366
Slit absorbers/resonators, 152, 153, 177
Snell's law, 17
Solutions to wave equation, 5–8
cartesian coordinates—stationary, one-dimensional solutions for, 6
spherical coordinates—stationary, one-dimensional solutions for, 6–8
Sone, units of, 56
Sound
absorbers. *See* Absorbers
attack, 89
direct, 148
intensity, 10–11
level, 43, 56
perception/judgment, 408–409
power, 11–12
problems to solve, 28–29
propagation losses, 12–13
reflection/transmission. *See* Reflection and transmission at boundaries
reinforcement, 136, 143, 158
speed of, 1
transient, 52
voiced/unvoiced, 71
Sound energy density, 90
Sound exposure, effect of excessive, 58–64
audiometry and audiogram, 59, 62
ear defenders, 63–64
excessive noise, criteria for, 61, 63
hearing impairment/hearing loss, 59
impulsive noise, 63
threshold shift, temporary and permanent, 58–59

tinnitus, 61
Sound field
impedance, 9
synthesis, 111, 157, 158
Sound field components, subjective impression of, 112–128
coloration, 119–120
delay vs. level differences, 116, 117–118
echo, 118–119
precedence effect, 115–116, 117
reverberation time/subjective preference, 121, 123–126
sequences or reflected sound, 120–121, 122
simple sound fields, 113–115
subjectively diffuse sound fields, 126–128
Sound generation mechanisms, 218
Sound isolation, 231–256
airborne sound, insulation against, 231, 232
balanced spectrum design, 247–248
composite transmission loss, 244
cracks and other leaks, 244–245
flanking transmission, 245–247
impact sound, insulation against, 248–251
multiple-panel partitions, transmission loss of, 238–244
problems to solve, 253–256
single-panel partitions, transmission loss of, 234–238
sound reduction index, 234
sound transmission loss, 232–234
structure-borne sound, insulation against, 252–253
Sound perception. *See* Spatial sound perception
Sound pressure level(s), 42
sound power level and, 94–97
weighted, 43
Sound propagation in inhomogeneus media, 23–26
temperature variation, 24
wind, 24–26
Sound radiation and generation, 217–230
flow, sound generation by, 226–229
metrics for sound radiation, common, 217–220
problems to solve, 229–230
vibrating pistons/membranes, 220–221
vibrating sheets carrying bending waves, 221–226
vibrating surfaces, 220
Sound radiation, common metrics for, 217–220
radiation loss factor, 220
radiation ratio, 219–220
radiation resistance, 218–219
sound generation mechanisms, 218
Sound reduction index, 234
Sound reproduction
binaural, 384–385, 386
multichannel, 387–389
rooms for, 151–153
Sound Transmission Class (STC), 234
Sound transmission loss, 232–234
Spacing layer, 211
Spaciousness, 148
Spatial impulse response (SIR), 120
Spatial sound perception, 111–133
metrics for room acoustics, 128–132
problems to solve, 132–133
sound field components, subjective impression of, 112–128
Special chambers, measurements using, 399–402
Spectral characteristics of speech, 71–73
Spectral density, 35
Spectrograms, 73, 403–405
Spectrum
quality descriptor, 138
and time history, 31
Speech. *See also* Voice and speech
audiometry, 61
envelope, 73
spectral characteristics of, 71–73
Speech intelligibility, 73, 75–79
Speech intelligibility index (SII), 77
Speech interference level (SIL), 137–138
Speech transmission index (STI), 77, 137

Speed of sound, 1
Spherical symmetry, 6
Spherical wave, sound field impedance in, 10
Spreading function, triangular, 373
Square meters Sabine, 89
Standing-wave phenomena, 13
Stapedius reflex, 63
Statistical energy analysis (SEA), 81, 153, 154, 233
Statistical room acoustics, 81, 87–99
  attack and rise time, 89
  diffuse sound field, 87–88
  extreme shape and/or absorption distribution, 98–99
  power received by a surface, 97–98
  reverberation curves, 94
  reverberation time, 89–90
  Sabine's formula, 90–94
  sound pressure level/sound power level, 94–97
STC. *See* Sound Transmission Class
Stereo using headphones, 382–384. *See also* Headphones and earphones
Stereophony and phantom sources, 378–382
Steven's phon, 56
STFT. *See* Short-time Fourier transform
STI. *See* Speech transmission index
Strength index, 129
Structure-borne sound, 231
  insulation against, 252–253
  in lossy sheets, 208–209
Stylus arms, phonograph systems and, 299–300
Stylus tips, phonograph systems and, 299
Subharmonics, 66
Successive approximation register (SAR), 363
Super-directional microphones, 286–289
  array microphones, 287–289
  noise canceling gradient microphones, 286–287
Supra-aural headphones, 341
Surround sound, 387, 388
Suspended ceiling, 240
Sweet spot, sound reproduction and, 308

T

Task performance, 136
Technical room acoustics, 81
Temperature variation, speed of sound and, 24
Temporal envelope, 407
Temporary threshold shift (TTS), 58, 59
Test session. *See* Listening tests
Tetrahedron microphone technique, 387, 389
THD. *See* Total harmonic distortion
Thermodynamic properties, 4
Threshold
  of audibility, 54
  of feeling, 55
  of pain, 55
Thurstone's method, 411
Timbre, pitch and, 57–58
Time gaps, 148
Time-to frequency mapping, 371
Tinnitus, 62
TL. *See* Transmission loss
Tone arms, phonograph systems and, 297–298
Tone audiometry/tone audiogram, 60
Total harmonic distortion (THD), 396
Transaural sound reproduction, 385
Transform methods, 371–373
Transient sounds, 52
Translucent absorbers, 181
Transmissibility, 260
Transmission loss (TL), 231, 232
  composite, 244
  of multiple-panel partitions, 238–244
  of single-panel partitions, 234–238
Transmission margin, 359
Transverse waves
  infinite media and, 199–200
  in sheets, 201, 202
TTS. *See* Temporary threshold shift
Tuned mechanical resonators, 257
Turbulence, 226
Turntable. *See* Phonographic systems

U
Ultrasonic scale modeling, 154, 155
Unvoiced sounds, 71

V
Variable absorbers, 182, 183
Velocity
  isolation, 257
  microphone, 285
Vibrating pistons/membranes, sound radiation by, 220–221
Vibrating sheets carrying bending waves, sound generation by, 221–226
  bending waves in an infinite sheet, 222–223
  bending waves in damped sheets, 225, 226
  bending waves in finite sheets, 223, 224–225
  reduction of radiation from bending waves in sheets, 225
Vibrating surfaces, sound radiation by, 220
Vibration isolation, 257–268
  impedance and mobility, 259
  linear single-degree-of-freedom systems, 260–263
  metrics for, 260
  problems to solve, 268
  real systems, 266–267
  theory, classical, 257–259
  theory, using electromechanical analogies, 263–266
Virtual pitch, 58
Viscoelastic materials, 209, 210
Viscous losses, 12
Vocal cords, 70, 71
Voice and speech, 70–79. *See also* Speech
  directivity of voice, 73, 74, 75
  problems to solve, 79–80
  spectral characteristics of speech, 71–73
  speech intelligibility, 73, 75–76
  voice directivity, 73, 74, 75
  voice mechanism, 70–71
  warning system, high-directivity, 26
Voiced sounds, 71

W
Walls. *See* Barriers
Warmth, 131, 147
Wave
  number, 6
  propagation, 1
Wave equation, 1–8
  combined equations and, 5
  equation of continuity, 2–3
  equation of motion, 3–4
  plane waves, 2
  solutions to, 5–8
  thermodynamic properties, 4
Wave field synthesis (WFS), 374, 387, 389
Wave theoretical approach, 99–107
  Eigenfrequencies, complex, 104–105
  Eigenfrequencies and Eigenmodes, 101–102
  loss factor and reverberation time, 105–106
  modal density, 102, 103–104
  Q-value, 105
  wave theory and diffuse sound fields, 106–107
Wave theory
  based room acoustics, 81
  diffuse sound levels and, 106–107
Wave types in infinite media, 197–200
  longitudinal waves, 197–199
  transverse waves, 199–200
Wave types of limited extension, 200–207
  bending waves in sheets, 201, 203–207
  membrane vibration, 201
  quasilongitudinal waves in sheets, 200–201
  transverse waves in sheets, 201, 202
Wavelength criterion, room acoustics and, 82–83
Wavelet analysis, 403–405, 406
Waves in solids, 197–216
  damping by viscoelastic layers, 209–213
  problems to solve, 215–216
  structure-borne sound in lossy sheets, 208–209
  types of waves, 197
  wave types in infinite media, 197–200

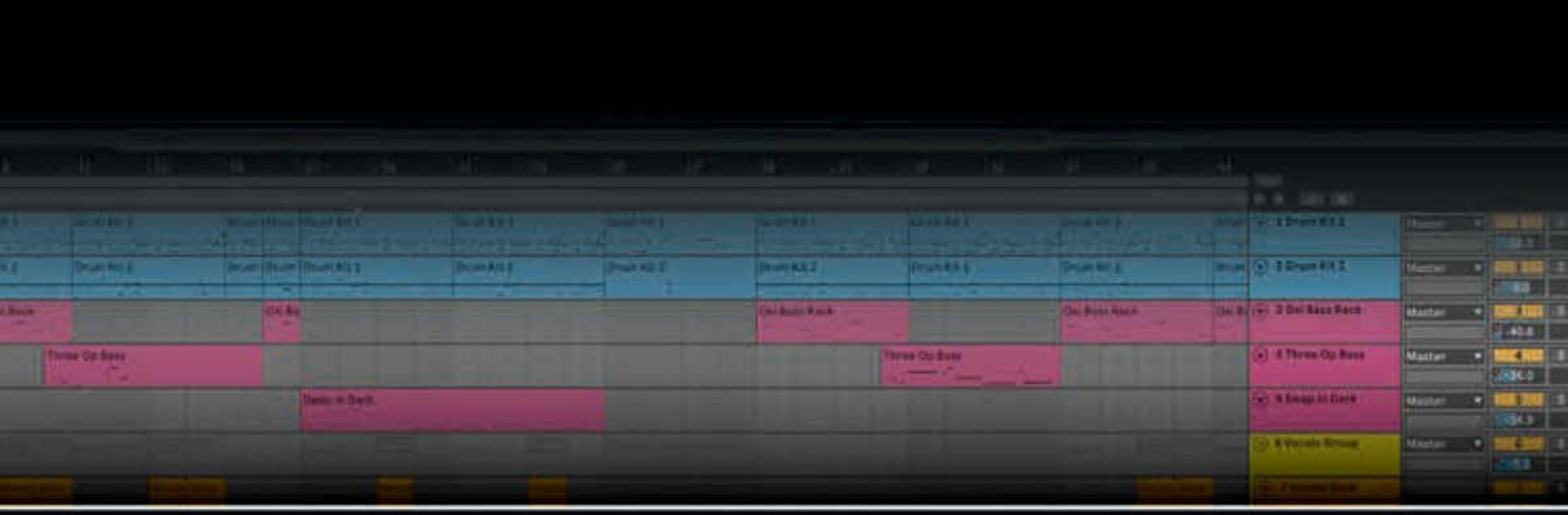
wave types of limited extension, 200–207
Weighted impact sound improvement index, 249
Weighted normalized impact sound pressure level, 249
Weighted sound pressure levels, 43
Weighting filters, 56
WFS. *See* Wave field synthesis
White noise, 35
White spectrum, 139
Wind, sound propagation and, 24–26
Wind noise, microphones and, 289–290
Windows. *See* Doors and windows

Y

Young's modulus, 199, 208, 210

Z

Zero padding, 403
Zwicker's Loudness, 398
Zwicker's phon, 56



ABLETON LIVE 101

AN INTRODUCTION TO ABLETON LIVE 10



Eric Kuehnl and Andrew Haak

ONLINE MEDIA INCLUDED

Ableton Live 101

An Introduction to Ableton Live 10

Ableton Live 101
An Introduction to Ableton Live 10

Eric Kuehnl
Andrew Haak

with contributions by

Frank D. Cook

NextPoint Training, Inc.

Ableton Live 10

Rowman & Littlefield

Lanham • Boulder • New York • London

Published by Rowman & Littlefield

An imprint of The Rowman & Littlefield Publishing Group, Inc.

4501 Forbes Boulevard, Suite 200, Lanham, Maryland 20706

www.rowman.com

6 Tinworth Street, London SE11 5AL, United Kingdom

Copyright © 2019 by NextPoint Training, Inc.

Book design by NextPoint Training, Inc.

All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without written permission from the publisher, except by a reviewer who may quote passages in a review.

Ableton Live and all related product names and logos are registered trademarks of Ableton AG in Germany and/or other countries. All other trademarks are the property of their respective owners.

The media provided with this book and accompanying course material is to be used only to complete the exercises contained herein or for other valid educational purposes. No rights are granted to use the media in any commercial or noncommercial production.

All product features and specifications are subject to change without notice.

Library of Congress Cataloging-in-Publication Data available

ISBN 978-1-5400-4686-4 (paperback)

ISBN 978-1-5381-3470-2 (eBook)



The paper used in this publication meets the minimum requirements of American National Standard for Information Sciences—Permanence of Paper for Printed Library Materials, ANSI/NISO Z39.48-1992.

Contents

Cover

Half Title

Title

Copyright

Contents

Acknowledgments

Introduction. Welcome to Ableton Live

Chapter 1. Background Information

- The Ableton Live Digital Audio Workstation
  - Audio Processing
  - MIDI Production
  - Mixing and Automation
  - Audio for Video Post-Production
- Ableton Live History and Evolution
  - Max for Live
  - Control Surfaces
- Audio Basics: Waveform, Frequency, and Amplitude
  - Waveform
  - Frequency
  - Amplitude
  - Recording and Playing Back Analog Audio

Converting Audio to Digital Format
The Importance of Sample Rate
The Importance of Bit Depth
Sample Rate, Bit Depth, and File Size
Recording in Digital Format
Ableton Live Configurations
Software Editions
Host-Based Systems Versus DSP-Accelerated Systems
Audio Interface Options
Cross-Platform Issues
Keyboard Commands
File-Naming Conventions
Review/Discussion Questions

Exercise 1. Selecting Your Audio Production Gear

Chapter 2. Getting Started with Ableton Live

Target Systems
Registering Your Software
Software Installation and Operation
Included Devices and Packs
Installing Additional Packs
Ableton Edition Comparison
Ableton Live File Structure
File Organization
Project Components
Starting Ableton Live
Powering Up Your Hardware
Launching Ableton Live
Authorizing Live
Accessing Connected Audio Devices

Optimizing Ableton Live Performance
The Ableton Live Menu Structure
File Menu
Edit Menu
Create Menu
View Menu
Options Menu
Help Menu
Ableton Live Views and Sections
Session View
Arrangement View
The Browser
Detail View
Control Bar
Overview (Arrangement View only)
Rulers and Scrub Area (Arrangement View only)
Adjusting the Display of Views and Sections
Review/Discussion Questions

Exercise 2. Identifying Ableton Live Views

Chapter 3. Ableton Live Basic Controls

Playback in Arrangement View
Clip Editing in Arrangement View
Selecting a Clip
Moving a Clip
Resizing a Clip
Creating Fades
Selecting an Area for Playback or Editing
Zooming in Arrangement View
Zooming with the Magnifying Glass Icon

- Zooming with the Arrangement Overview
- Zooming with Keyboard Shortcuts

Grid Editing in Arrangement View
- Moving and Placing Clips
- Editing Clips
- Edit Grid Modes
- Setting the Edit Grid Mode and Value

Time Scales and Rulers in Arrangement View
- Arrangement Position
- Loop and Punch-In Start and Length
- Rulers

The Scrub Area in Arrangement View
- Adjusting the Loop Start/Punch-In Point and Length
- Changing the Time Signature
- Using Locators

Performance Control Features
- Tap Tempo Button
- Tempo Field
- Meter Display
- Metronome Button
- Metronome Settings Selector
- Quantization Menu

Modifying Devices and Clips
- Accessing Devices in Device View
- Accessing Clips in Clip View
- Edit Mode Versus Draw Mode

Using the Info View

Review/Discussion Questions

Exercise 3. Identifying Primary Tools and Controls

Chapter 4. Creating Your First Live Set

- Creating and Configuring a Live Set
  - Configuring Live's Audio Preferences
  - Audio Device and Sample Rate Settings
  - File Type and Bit Depth Settings
  - File Size Considerations
- Working with Tracks
  - Supported Track Types
  - Adding Tracks to Ableton Live
  - Naming Tracks
  - Deleting Tracks
  - Adding Audio to Tracks
  - Adding MIDI to Tracks
- Controlling Playback
  - Playback in Arrangement View
  - Playback in Session View
- Saving, Locating, and Opening Live Sets and Projects
  - Live Sets and Live Projects
  - Saving a Live Set
  - Locating and Opening a Live Set
- Review/Discussion Questions

Exercise 4. Creating a Live Set

Chapter 5. Importing and Working with Media

- Considerations Prior to Import
  - File Characteristics
- Import Methods
  - Live's Built-in Browser
  - Importing Audio from the Browser

Audio, MIDI, and Tempo Changes
Other Methods of Importing Audio Files
Importing with a Menu Command
Importing from an External File Browser
Importing Video
Video Track Display
Working with Video Tracks
Review/Discussion Questions

Exercise 5. Importing Audio

Chapter 6. Working with MIDI

MIDI Basics
MIDI in Ableton Live
Creating MIDI Tracks
MIDI Versus Audio Clip Operation
Setting the Tempo and Time Signature
Setting the Base Tempo
Automating Tempo Changes with Breakpoints
Setting the Base Time Signature
Preparing to Record MIDI
Enabling the Metronome
Configuring Record Quantization
Connecting a MIDI Device
Selecting MIDI Inputs
Record-Arming MIDI Tracks
Using MIDI Arrangement Overdub Mode
Using Virtual Instruments
Inserting a Virtual Instrument on a MIDI Track
Using Drum Rack
Using Simpler

Using Wavetable (Ableton Live 10 Suite only)
Recording a MIDI Performance
Session View
Arrangement View
Creating a MIDI Performance Manually
Creating and Editing MIDI Performances
The MIDI Note Editor
Creating MIDI Notes
Selecting MIDI Notes
Moving a Selection to a Different Note
Editing MIDI Notes
Editing Velocity
Additional Editing Features
MIDI Quantization
Navigating the MIDI Note Editor
Review/Discussion Questions

Exercise 6. Working with MIDI

Chapter 7. Making Your First Audio Recording

Before Recording
Audio Storage Requirements
Determining Available Disk Space
Preparing to Record
Enabling the Metronome
Customizing the Metronome
Checking Hardware Connections
Selecting and Routing the Sound Source
Arming (Record-Enabling) Tracks
Monitoring Modes
Controlling Input Level and Pan

Recording and Managing Audio
Recording in Arrangement View
Recording in Session View
Organizing after Recording
Disarm the Record Tracks and Adjust Playback Settings
Organize Audio Files and Clips
Removing Audio Clips and Deleting Audio Files
Review/Discussion Questions

Exercise 7. Recording Audio

Chapter 8. Selection, Navigation, and Configuration

Using the Arrangement Overview
The Overview Outline (Arrangement View only)
Zooming and Scrolling with the Overview (Arrangement View Only)
Making Selections in the Arrangement View
Selecting Clips and Time
Zooming and Scrolling the Arrangement View
Zooming In and Out
Zooming In and Out on Selections
Zooming to Fit the Screen
Scrolling
Adjusting Your Live Set
Changing the Track Order
Folding and Unfolding Tracks
Adjusting Track Size
Working with Locators (Arrangement View Only)
Adding Locators
Editing Locators
Recalling and Playing from Locators

Using Locators for Loops and Selections
Deleting Locators
Navigating the Arrangement
Navigating Time
Navigating Tracks
Navigating Through Clips and Locators
Making Selections while Navigating
Review/Discussion Questions

Exercise 8. Recording, Configuring, and Adding Locators to an Arrangement

Chapter 9. Editing Your Media

Selecting Playback Options
Follow Behavior
Arrangement Loop
Using the Edit Grid
Configuring the Edit Grid Settings
Snap to Grid
Disabling Snap to Grid
Editing Clips
Basic Editing Commands
Moving and Resizing Clips in the Arrangement View
Moving Clips
Resizing Clips (Arrangement View))
Looping Clips
Nudging Clips and Selections
Undoing Your Work
Using Multi-Level Undo
Restoring from Backup Sets
Review/Discussion Questions

Exercise 9. Editing Audio

Chapter 10. Mixing and Creating a Bounce

- Basic Mixer Terminology
  - Inputs
  - Outputs
  - Inserts (Devices)
  - Sends and Returns
- Mixing in the Session View
  - Configuring the Session View for Mixing
- Automation
  - Arrangement Automation Workflow
  - Recording Arrangement Automation
  - Viewing Arrangement Automation
  - Drawing Automation Envelopes
  - Editing Automation Breakpoints
  - Locking Envelopes
  - Overriding Automation
  - Deleting Automation
- Using Devices for Mixing
  - Devices Provided with Ableton Live
- Backing Up Your Session
  - Considerations for Backup and Archive Copies
  - Saving an Archive Copy
- Creating a Stereo Mixdown
  - Mixdown Options
  - Considerations for Exporting a Mix
  - Exporting Audio
  - Uploading a Mix to SoundCloud
- Review/Discussion Questions

Exercise 10. Automating Shared Effects and Exporting a Stereo Mix

Final Project. Creating a Musical Arrangement

- Powering Up
- Getting Started
  - Adjust the Live Set Display
  - Connect Monitoring Devices
- Creating New Tracks
  - Create, Name, and Color Tracks
  - Save Your Live Set
- Working with MIDI Data
  - Add the Drum Rack to the Snare Drum Track
  - Add Simpler to the Bass Instrument Track
  - Create a Snare Drum MIDI Clip
  - Duplicate the Snare Drum MIDI Clip
  - Create a Bass Instrument MIDI Clip
  - Save Work in Progress
- Working with Audio Data
  - Import an Audio File to a Track
  - Save Work in Progress
- Recording from Session to Arrangement View
  - Prepare for Recording into Arrangement View
  - Record into Arrangement View
  - Track Status Indicators
  - Verify Your Results
  - Clean up the Recorded Arrangement
  - Save Work in Progress
- Finishing Your Work
  - Add a Limiter
  - Export the Song

Archive Your Work

Appendix A. Ableton Live Audio and MIDI Effects

Appendix B. Ableton Live Software Instruments

About the Authors

Acknowledgments

The authors would like to extend special thanks to the following individuals who have provided support, feedback, technical information, editorial input, and other contributions for this version of the book: Mark Anderson, John Cerullo, Andy Cook, Frank D. Cook, Matt Donner, Mark Garvey, Greg Gordon, Bruce Tambling, Michael Tanamachi, Simon Pennington.

Eric would like to thank his wife and children for their support and understanding: Amanda Goodroe, Thom, and Freddie.

Andrew would like to thank Casey, Eric Kuehnl, Steve Heithecker, Matt Donner, the Pyramind community at large, and Football the cat.

Introduction

Welcome to Ableton Live

Congratulations on beginning work under the NextPoint Training Digital Media Production program. Within this program, you have the opportunity to pursue certification in Ableton Live and other digital media applications.

Ableton Live's award-winning technology is embraced by recording artists and audio production professionals around the world. The material in this course book will help you to join their ranks, unleashing the power and productivity of your own Ableton Live system.

This book represents the first step on a journey toward mastering Ableton Live software. The information, exercises, and projects you will find here are written for Ableton Live systems running version 10.1 software. However, the vast majority of the book applies equally to earlier software versions, including Ableton Live 9.x. Whether you are interested only in self-study or you would like to pursue formal training through a NextPoint Training Certification Partner, this book will develop your core skills and introduce you to the awe-inspiring power of Ableton Live.

About This Book

This book provides a course outline for the audio enthusiast with relatively little Ableton Live experience. This text and the associated Ableton Live 101 instructor-led training course were developed by NextPoint Training, Inc. (NPT) as part of our Digital Media Production program and certification offerings. While this course book can be completed through self-study, we recommend the hands-on experience available through an instructor-led class with an NPT Certification Partner.

For more information on the classes offered through the NextPoint Training Digital Media Production program, please visit http://NextPointTraining.com/certification.

Ableton Live 10 Edition

This edition of the Ableton Live 101 course has been written to address recent software changes up through the Ableton Live 10.1 release. The material is focused on the principles that you need to understand to complete an Ableton Live project, from initial setup to final export. Whether your project involves recording audio, preparing MIDI sequences for virtual instruments, or editing and mixing audio files, Ableton Live 101 will teach you the steps required to succeed.

The Download Media

This book makes use of media files for the included exercises and the final project. The media files can be accessed by visiting www.halleonard.com/mylibrary and entering your access code, as printed on the opening page of this book. From there, you can download the Media Files 2019 folder for this course. Instructions for downloading the media files are provided with the exercises and final project, as needed.

The downloadable Media Files 2019 folder includes three subfolders for use with this book.

Sample Files for Exercises

The 01 Completed Exercises folder within the Media Files 2019 folder provides sample Set files of each completed Ableton Live exercise (Exercises 4 through 10). These are provided primarily for reference purposes. Sample files 4 though 9 can also be used as starting points for subsequent work in Exercises 5 through 10, as needed.

Exercise Media

The 02 Exercise Media folder provides the audio and MIDI files you will import to complete the work in Exercises 5 through 10, following the instructions included in this book.

Files for the Final Project

The 03 Final Project Media folder provides content for the final project that you will complete in the second part of this book. This folder includes a Set

file and associated media required to complete the project, following the step-by-step instructions in the Final Project section of the book.

Course Prerequisites

Most Ableton Live enthusiasts today have at least a passing familiarity with operating a computer. If you consider yourself a computer novice, however, you should review some basics before beginning this course. You will need to know how to complete such tasks as:

- Starting up your computer
- Using the mouse to navigate through folders and select files
- Using standard menus and menu commands
- Using common keyboard commands for standard operating system operations
- Locating, moving, and renaming files and folders
- Using standard cut, copy, and paste commands
- Opening, saving, and closing files

This course focuses on using Ableton Live in a digital audio recording and production environment. The work requires a basic understanding of MIDI and audio recording techniques, processes, and equipment, such as the following:

- MIDI devices
- Microphones and miking techniques
- Audio interfaces
- Mixer signal flow

- Audio monitoring equipment

If you are a beginner in the field of audio production, you can supplement this text with independently available literature or courses on audio recording tools and techniques.

Course Organization and Sequence

This course has been designed to familiarize you with the practices and processes you will use to complete a recording, editing, and mixing project.

Chapters and Exercises

The first part of the book includes ten primary chapters and ten associated short exercises. The first three chapters provide information on Ableton Live's background and history, installation and startup procedures, and the current user interface. The subsequent chapters present specific processes and techniques, in the general order that you will use them to complete a project —from creating a new Set, to recording and editing, and on through to mixing and completing a final export.

Each of the chapters is followed by a brief exercise that gives you practice applying what you have learned.

Final Project

The second part of this book is the final project. This section includes instructions for completing an unfinished project included in the Media Files 2019 folder. The project in this section can be completed at any point as you progress through the ten primary chapters.

Appendices

This edition of the book includes two appendices. Appendix A provides information on the audio and MIDI Effects devices that are included with Ableton Live. Appendix B provides details on the Software Instrument devices that are included with Ableton Live.

The NPT Digital Media Production Program

This Ableton Live 101 book has been written as a textbook for teaching and learning Ableton Live software. In addition to being an off-the-shelf guide for consumers, this book is also the official text for NextPoint Training's professional certification in Ableton Live. By completing the coursework in this text, you are taking an important step toward certification. And consider this: Having a certification from NextPoint Training just might help you land that next gig, find others with similar skills and interests, or even obtain your dream job in the industry.

To become certified in Ableton Live, you must register at ElementsED.com, where you can complete additional study in Ableton Live, review coursework (as desired), take practice quizzes, and take the Ableton Live Certified Professional Exam. Detailed information on current requirements is available from the NextPoint Training website at http://NextPointTraining.com/certification.

Curriculum and Certification Levels

NextPoint Training currently offers certification in Ableton Live software, Pro Tools|First software, and Steinberg Cubase software. (Visit NextPointTraining.com/certification for the latest information.) The certification credential associated with the Ableton Live 101 course is Ableton Live Certified Professional I.

The Certified Professional program prepares individuals to operate an Ableton Live system in an independent production environment. This certification requires successful completion of the Ableton Live 101 instructor-led course at an NPT Certification Partner school and a passing score on the associated Ableton Live certification exam.

Other Courses Offered in the Digital Media Production Program

NextPoint Training offers several additional certification courses to help you become proficient using a wide range of audio production tools.

- ***Audio Production Basics with Pro Tools | First*** teaches the basics of recording, editing, mixing, and processing audio using Pro Tools | First software. It offers an introduction to digital audio for students aspiring to work in music or video production, audio engineering, broadcast, or new media.

- ***Steinberg Cubase 101*** provides the foundational skills needed to operate a Steinberg Cubase system in an independent production environment. The goal of this course is to help individuals start working effectively on their own projects in Cubase Artist or Cubase Pro software.

NextPoint Training Course Configuration

NextPoint Training uses a version–specific approach to course design, enabling students and educators to access classes based on the products and software versions that meet their particular needs and training environments.

Audio Curriculum

The NextPoint Training audio coursework includes programs supporting certification in several areas, including Ableton Live, Avid Pro Tools, Steinberg Cubase, and more. The Ableton Live certification path is described above. Course components are designed to be completed as individual product-focused classes; however, the same content may be made available through different class configurations.

Other Media Courses

Details on of each of the courses offered through the NextPoint Training Digital Media Production program are available in the NPT Certification Program Guide.

How Can I Learn More?

Additional resources are available to help you explore the topics covered in this book, review key points in each chapter, and test your knowledge of the material. The **Ableton Live 101 Study Guide** module available through the Elements|ED online learning platform (ElementsED.com) allows learners to

review key concepts and information from this course through visual examples and to assess their learning using practice quizzes. This module is particularly useful to help students prepare for the Ableton Live Certified Professional exam.

Conventions and Symbols Used in This Book

Following are some of the conventions and symbols used in this book, and throughout the books in the NextPoint Training Series.

Keyboard Shortcuts and Modifiers

Menu choices and keyboard commands are typically capitalized and written in bold text. Hierarchy is shown using the greater than symbol (>), keystroke combinations are indicated using the plus sign (+), and mouse-click operations are indicated by hyphenated strings, where needed. Brackets ([ ]) are used to indicate key presses on the numeric keypad.

Convention	Action
File > Save Set	Choose Save Set from the File menu.
Ctrl+N	Hold down the Ctrl key and press the N key.
Command-click (Mac)	Hold down the Command key and click the mouse button.
Right-click	Click with the right mouse button.
Press [1]	Press 1 on the numeric keypad.

Icons

The following icons are used in this book to call attention to tips, shortcuts, listening suggestions, warnings, and reference sources.

Tips provide helpful hints and suggestions, background



information, or details on related operations or concepts.



Shortcuts provide useful keyboard, mouse, or modifier-based shortcuts that can help you work more efficiently.



Warnings caution you against conditions that may affect audio playback, impact system performance, alter data files, or interrupt hardware connections.



Cross-References alert you to another section, book, or resource that provides additional information on the current topic.



Online References provide links to online resources and downloads related to the current topic.

Chapter 1

Background Information

This chapter introduces you to Ableton Live's capabilities and uses for audio production, MIDI workflows, mixing, and video post-production. You will learn about the evolution of Ableton Live and get an introduction to the characteristics of analog and digital audio. You will also get an overview of the latest developments in Ableton Live 10 software and learn about the different Ableton Live configurations available today.

Learning Targets for This Chapter

- Identify common industry uses for Ableton Live software
- Recognize the contributions of historical developments in sampling and sound editing, MIDI technology, computer I/O, and recording technology to today's digital audio workstation
- Understand the relationship between sample rate and frequency response in digital audio
- Understand the relationship between bit depth and dynamic range in digital audio
- Recognize components and features of various Ableton Live configurations

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform. Sign up for free at ElementsED.com.

The Ableton Live Digital Audio Workstation

Ableton Live is one of the most widely used applications for music production in the world today, integrating capabilities in audio and MIDI recording, editing, mixing, and mastering, as well as support for desktop video. As such, Ableton Live empowers a wide range of producers—from hobbyist to professional—to easily achieve all of their production tasks within an intuitive yet powerful environment.

At its core, Ableton Live is a multi-track software-based music production system. When the first version of Ableton Live was introduced back in 2001, it's unique approach to audio and MIDI composition revolutionized the music production world.

Today's Ableton Live offers audio recording, graphical audio editing, MIDI sequencing, digital signal processing, and mixing through a single, integrated system. With the ability to incorporate QuickTime video files, Ableton Live has also established itself as viable option for composing for visual media such as film and television.

Audio Processing

Ableton Live works with audio that is stored electronically in digital format. The software records audio performances and saves them as files on a storage drive. Like a digital photograph that is comprised of a collection of discrete pixels, the audio files created by Ableton Live are made up of a collection of discrete *samples*. Ableton Live supports audio formats with resolutions up to 32-bit floating point and sample rates up to 192 kHz.

Just as you can use an image editor to modify, enhance, and otherwise alter your digital photographs in creative ways, you can use Ableton Live to edit your digital audio. Working in the digital realm makes it easy to copy, paste, move, delete, process, and otherwise manipulate parts of your recordings. Ableton Live lets you resize and split waveforms, convert audio data to MIDI data, change the timing and feel of performances, rearrange song sections, and much, much more.

MIDI Production

Ableton Live's built-in sequencing technology enables you to record and edit MIDI data along with your audio recordings. MIDI recordings differ from their digital audio counterparts in that they capture performance event data rather than sound samples. You can record MIDI signals from an external controller (such as Ableton Push) through a MIDI interface or USB port and then edit the data using Live's Clip View.

Ableton Live's MIDI features include MIDI tracks, MIDI effects, grid and groove quantize functions, velocity editing, and more. Furthermore, some editions of Ableton Live come bundled with several great-sounding virtual instrument plug-ins.

Mixing and Automation

Beyond recording, editing, and arranging, Ableton Live offers a software-based mixing environment that provides control over signal routing, effects processing, signal levels, panning, and more. The mixing operations in Ableton Live can be automated and stored with your Set file, enabling you to recall, edit, and refine your mixes over time. When you save an Ableton Live Set, all routing, automation, mixing, and effects settings are stored in their present state and will recall in that state whenever the Set is reopened.

Additionally, Ableton Live can be combined with hardware from third-party manufacturers in various configurations to provide multiple channels of simultaneous input and output for your projects. Large Sets including dozens of simultaneous audio and MIDI tracks can be managed without audio degradation. Ableton Live systems can range from very simple to extremely advanced and powerful.

Audio for Video Post-Production

Ableton Live also provides a powerful audio platform for video post-production tasks. You can import QuickTime movies and use the Ableton Live Video window for quick visual reference or full-screen display as you compose a music score, edit dialog, and/or create sound effects. When

completed, your finished movie file can be exported with the final audio mix embedded.

Ableton Live History and Evolution

In the 1990s, two musicians named Gerhard Behles and Robert Henke spent much of their time performing minimal techno around Berlin under the moniker *Monolake*. Out of necessity, the two developed custom performance software using the Max music programming language to trigger audio loops and samples in their shows. By the end of the 1990s, Behles and Henke made the decision to convert their messy Max code into a streamlined commercial application that could be sold to other musicians. After several years of development, the first version of Ableton Live was released in 2001. The software was an immediate success thanks to its intuitive interface and innovative sample manipulation and playback features.

Over the years since, Ableton has released a steady stream of updates that have each improved the functionality of Ableton Live. One of the most important early updates was Ableton Live 4 (2004), which introduced MIDI sequencing and VST plug-in support. Ableton Live 5 (2005) introduced track freeze and MP3 support. Ableton Live 6 (2006) introduced QuickTime video support and allowed users to produce music for film and video from within Live for the first time.

Max for Live

In 2007, Ableton announced that they were working with Cycling ’74 to integrate the Max music programming language directly into Ableton Live. In 2009, the two companies released Max for Live software, which made it possible to use Max patches directly inside of Ableton Live. Ableton acquired Cycling ’74 in 2017, and beginning with Ableton Live Suite 10, the Max for Live application is now integrated into Ableton Live at no additional cost.

Control Surfaces

In 2009, Ableton worked with Akai Professional to release the first dedicated

Ableton Live controllers. The result was the APC40, which closely approximated the Ableton Live software interface in an intuitive hardware format. The APC functionality included the ability to launch clips and scenes and adjust mixer and device settings from a hardware controller.

In 2013, Ableton again teamed up with Akai Professional to create the Push hardware instrument. Push took the APC concept to the next level and offered access to almost every aspect of Ableton Live from the controller, including note entry, step sequencing, and much more.

In 2017, Ableton designed and built Push 2 completely in-house. Push 2 expanded on the capabilities of the original Push by adding a larger display and higher quality pads.



Figure 1.1 Ableton Push 2 hardware instrument

Getting the Most from Your System

To make the most of your experience with Ableton Live 10 and to optimize your results using the platform, it is helpful to understand some audio basics and to familiarize yourself with the software options available with Ableton

Live. The remainder of this chapter focuses on these topics.

Audio Basics: Waveform, Frequency, and Amplitude

To work effectively with sound, it is helpful to understand what sound is and what gives a sound its character. When we hear a sound, what we actually experience is a variation in the air pressure around us. This variation results from vibrations in material objects—whether a knock on a tabletop, a running car engine, or a plucked guitar string. When a vibrating object moves through a back-and-forth motion, the variation in air pressure that it produces becomes an auditory event. If the object is vibrating at a frequency that falls within the range of human hearing, we perceive the varying air pressure as a sound. The nature of the sound we hear is determined by the waveform, frequency, and amplitude of the vibration.

Waveform

The waveform of the sound pressure variations that reaches our ears creates our perception of the sound's source, be it a tabletop, a car engine, or a guitar string. The waveform is the "shape" of the sound—or, more accurately, the shape of the vibration that produced the sound. As a vibrating object moves through its back-and-forth motions, its path is not smooth and continuous. Instead, the cycles of vibration are typically complex and jagged, creating a sound that is influenced by the physical material that the object is composed of and the resonance induced by the object's surroundings. Each object vibrates differently; the waveform of the vibration gives the sound its unique character and tone.

Frequency

The frequency of the sound pressure variations that reaches our ears creates our perception of the pitch of the sound. We measure this frequency in *cycles per second* (CPS), also commonly denoted as *Hertz* (Hz). These two terms are synonymous—15,000 CPS is the same as 15,000 Hz. Multiples of 1,000 Hz are often denoted as kilohertz (kHz). Therefore, 15,000 Hz is also written

as 15 kHz.

As the frequency of vibration increases, the pitch of the sound goes up—numerically higher frequencies produce higher pitches, while numerically lower frequencies produce lower pitches. Each time the frequency doubles, the pitch rises by one octave.

By way of example, the open A string on a guitar vibrates at 110 Hz in standard tuning. Playing the A note on the 12th fret produces vibrations at 220 Hz (one octave higher).



The range of human hearing is between 20 and 20,000 cycles per second, or stated another way, from 20 Hz to 20 kHz.

Amplitude

The intensity or amplitude of the sound pressure variations that reaches our ears creates our perception of the loudness of the sound. We measure amplitude in *decibels* (dB). The decibel is a logarithmic unit that is used to describe a ratio of sound pressure; as such, it does not have a linear relation to our perception of loudness.

As the amplitude of pressure variations increases, the sound becomes louder. Doubling the intensity of sound-pressure variations creates a gain of 3 dB; however, we do not perceive this change as doubling the sound's loudness. An increase of approximately 10 dB is required to produce a perceived doubling of loudness. By way of example, the amplitude of ordinary conversation is around 60 dB. Increasing the amplitude to 70 dB would essentially double the loudness; increasing amplitude to 80 dB would double it again, quadrupling the original loudness.

Recording and Playing Back Analog Audio

The task of a recording microphone is to respond to changes in air pressure—the waveforms, frequencies, and amplitudes that make up a sound—and translate them into an electronic output that can be captured or recorded. A

microphone functions as a *transducer*, converting acoustic energy into an electrical current.

The continuous electrical signal produced by a microphone is an alternating current with a waveform, frequency, and amplitude that directly corresponds to, or is analogous to, the original acoustic information. This electrical signal is thus considered to represent *analog audio*.

If this continuous analog signal is captured on traditional recording media, such as magnetic tape, it can be played back by directly translating the electrical waveform, frequency, and amplitude back into analogous variations in air pressure through the means of an amplifier and a loudspeaker.

Converting Audio to Digital Format

Before you can record or edit with Ableton Live, the analog audio (electrical signal) relayed by a microphone, guitar pickup, or other device must be translated into digital numeric information (binary data). This conversion is necessary so that the signal can be stored, read, and subsequently manipulated by a computer.

The process of translating electrical signals to binary data is referred to as *analog-to-digital conversion*, commonly abbreviated as *A/Dconversion*. Two essential factors affect the A/D conversion process: *sample rate* and *bit depth*.

The Importance of Sample Rate

Sampling is the process of taking discrete measurements of an electrical signal at various moments in time. Each measurement, or sample, is a digital representation of the signal voltage at that instant. Played back in succession, these samples approximate the original signal, much like a series of photographs played back in succession approximates movement in a film or video.

The sample rate is the frequency with which these digital measurements are collected.

Nyquist Theorem

A fundamental law of analog-to-digital conversion, commonly referred to as the Nyquist Theorem, drives the sample rate required for digital audio.

The Nyquist Theorem states that in order to produce an accurate representation of a given frequency of sound, each cycle of the sound's vibration must be sampled a minimum of two times. If the sample rate is any lower, the system will read the incoming frequencies inaccurately and produce the wrong tones. (In concept, this is much like the effect seen in early motion pictures, where a wagon wheel will appear to rotate backward due to the low frame rates being used.) In digital audio, the false tones produced by this type of frequency distortion are known as *aliastones*.

Minimum Sample Rate

Because the range of human hearing is generally accepted to be 20 Hz to 20 kHz, the Nyquist Theorem tells us that a sampling rate of at least 40 kHz (twice the upper range of human hearing) is required to capture full-frequency audio.

Most professional digital recording devices today offer sample rates of at least 44.1 kHz (often as much as 96 kHz or higher). The digital information on an audio CD is stored at a standard sample rate of 44.1 kHz.

The Importance of Bit Depth

Computers use binary digits called *bits* (0s or 1s) to represent each sample measurement that is collected. The number of bits used for each sample is referred to as the *binary word length*, or bit depth.

The more binary digits included in the bit depth, the greater the accuracy of each sample measurement. The relative amplitude (or loudness) of each sample is *quantized*, or rounded to the closest available whole-number value within the word length.

The range of numeric values available for each sample at a

given bit depth is equal to 2 to the *n*th power (2^n), where *n* is the number of bits in the binary word.

By way of example, consider a 4-bit binary word. This word length can represent only 16 different amplitude levels (2^4). As such, a 4-bit binary word would record audio using only 16 discrete amplitude levels. By contrast, a 16-bit digital word could represent 65,536 discrete amplitude levels (2^{16}), creating a much more continuous dynamic response. A 24-bit digital word could define more than 16 million discrete amplitude levels (2^{24}).

Larger binary words are able to quantify variations in amplitude with much greater accuracy. Therefore, a 24-bit audio file will always more accurately reflect the dynamic range of the original sound than its 16-bit counterpart.

Thirty-two-bit floating-point files represent discrete amplitude levels in the same way as 24-bit files. The 8 additional bits provide exponent biasing and allow for headroom above full-scale 24-bit audio.

Calculating Dynamic Range

A very general rule of thumb can be used to calculate the dynamic range capability of an A/D system. By multiplying the word size by six, you can estimate the useful dynamic range of a fixed-point system. For example, a system with an **8-bit** binary word would produce a dynamic range of about 48 dB (8 × 6), while a **16-bit** system would accommodate a 96-dB dynamic range (16 × 6). A **24-bit** system would have a theoretical dynamic range of 144 dB (24 × 6).

In theoretical terms, the dynamic range (or signal-to-quantization noise ratio) increases by approximately 6 dB for each bit added to the binary word length.

Minimum Bit Depth

The useful dynamic range of speech and music is generally considered to be

from 40 to 105 dB. To capture this range, an A/D converter must be able to accurately represent differences in amplitude of at least 65 dB; stated another way, it must have a minimum 65-dB dynamic range. This would require at least 11 bits in the binary word.

To provide an adequate dynamic range to minimize the impact of the noise floor at the low end of the spectrum, and to allow a healthy amount of headroom at the high end, Ableton Live uses a minimum word length of 16 bits. Greater bit depths can be used to increase precision and accommodate a wider dynamic range.

Sample Rate, Bit Depth, and File Size

A consequence of files with higher sample rates and greater bit depths is the higher storage capacity required to record them. Each minute of 16-bit/44.1-kHz mono audio occupies about 5 MB of storage space. Higher sample rates increase the storage requirement. At 96 kHz, each minute of 16-bit mono audio requires about 11 MB of storage space.

Increasing the bit depth has the same effect. For example, one minute of 24-bit/44.1-kHz mono audio occupies about 7.5 MB of hard-drive storage space, while the same audio in a 32-bit float file (at 44.1 kHz) will require about 10 MB of hard-drive space per minute.



Stereo files require twice as much storage space as mono files, since each file includes two channels of audio (left and right).

Recording in Digital Format

When you are recording into Ableton Live using audio that is already in a digital form (from DAT or CD, for example), you don’t need to translate the audio before bringing it into the system. The process of converting from digital to analog and back to digital can introduce distortion and degrade the original signal.

Digital Transfers

To prevent audio degradation, unnecessary conversions should be avoided. Keeping audio information in the digital domain while transferring between machines or devices will retain its sonic integrity with no discernible signal degradation.

Digital Audio Connections

On the rear panel of many audio interfaces are connections for accomplishing digital transfers. Common digital connections include *S/PDIF*, which uses RCA jacks (sometimes called *coaxial jacks*), and *AES/EBU*, which uses XLR-type connectors. S/PDIF is the *Sony/Philips Digital Interface* standard, a consumer format, and AES/EBU is the *Audio Engineering Society/European Broadcast Union* digital interface standard, a professional format.

Although the formats are nearly identical in audio quality, if given the choice, you should use the AES/EBU format over S/PDIF. As a professional format, AES/EBU is technically more stable, and it filters out any copy protection encoded in the digital audio stream.

Ableton Live Configurations

The requirements for your digital audio recording projects will determine the edition of Ableton Live that you will need to use. Ableton Live 10 is available in three editions: Ableton Live Intro, Ableton Live Standard, and Ableton Live Suite.

Software Editions

Ableton Live Intro (and the similar Ableton Live Lite) is a basic version of Ableton Live, with limitations on editing features, track count, scene count, and input and output (I/O) capabilities.

Ableton Live Standard provides the full editing feature set, unlimited tracks and scenes, and up to 256 channels of I/O. This book focuses on Ableton Live Standard software features.

Ableton Live Suite offers all of the Ableton Live Standard features, and

includes additional virtual instruments, audio and MIDI effects, and Max for Live.

For more information about the various editions of Ableton Live, see "Ableton Live Edition Comparison" in Chapter 2 of this book.

All Ableton Live editions are available for both Mac and Windows operating systems.

Ableton Live Intro vs. Ableton Live Standard vs. Ableton Live Suite

Throughout this book, we use the term to refer generically to all available software **Ableton Live** editions and the terms, and, to to refer **Intro Ableton Live Ableton Live Suite Ableton Live Standard** specifice ditions of Ableton Live software, where needed for differentiation.

Host-Based Systems Versus DSP-Accelerated Systems

Ableton Live systems can either be host-based, meaning they rely solely on the processing power of the host computer for routing, mixing, and processing of audio signals, or DSP-accelerated systems, meaning that the computer's processor is supplemented by additional processor chips dedicated to digital signal processing (DSP) for mixing and real-time processing. DSP-accelerated Ableton Live systems can be constructed using third-party products that feature DSP processing. The most common examples are the Apollo interfaces and UAD Satellite units by Universal Audio.

Audio Interface Options

An audio interface provides the analog-to-digital conversion required for recording to Ableton Live, as well as the digital-to-analog conversion

required for playback from Ableton Live to your analog speakers or headphones. Ableton Live can be run without an audio interface, using your computer's built-in speakers or headphone jack for playback; however, recording options may be limited or unavailable.

Examples of Audio Interfaces

All versions of Ableton Live support a range of audio hardware:

- **Basic Interfaces**—Ableton Live supports a variety of hardware I/O options on both Mac and Windows computers. These may include your computer's built-in audio hardware or an external interface that connects via USB, FireWire, or Thunderbolt. Many of these peripherals can be powered by the computer's FireWire or USB bus, enabling portability for laptop systems.



Figure 1.2 Focusrite's Scarlett 2i4 is an excellent entry-level interface

When considering the purchase of an external interface, it's important to remember that Ableton Live Intro supports a maximum of eight input and output channels. Although interfaces with more inputs and outputs can be used with Ableton Live Intro, you'll need to select a maximum of eight inputs and outputs at one time.

- **Mid-Range Interfaces**—Ableton Live Standard and Ableton Live Suite support up to a maximum of 256 channels of input and output, so much larger interfaces (or combinations of interfaces) can be used. You should consider a mid-range interface if you need more than just a couple of channels of input and output. You should also consider stepping up to a

mid-range interface if you're looking for better quality microphone preamps and analog-to-digital and digital-to-analog convertors.



Figure 1.3 MOTU's UltraLite Mk3 is an affordable, mid-range interface

- **High-End Interfaces** – If you're looking for an interface with a large number of inputs and outputs, you'll most likely want to consider a high-end interface from manufacturers such as Antelope Audio, Apogee, Focusrite, or Universal Audio. These interfaces offer as many as 64 channels of input and output to accommodate the largest recording configurations. You may also want to step up to a high-end interface if you're looking for the absolute best quality in terms of microphone preamps and convertors.



Figure 1.4 Antelope Audio's Orion Studio is high-end interface with 32 channels of I/O

Cross-Platform Issues

Ableton Live editions are available for both Mac and Windows systems. Most Ableton Live controls, tools, procedures, and menus are similar on both systems. There are, however, some differences in keyboard commands and file-naming conventions that can impact your work when moving between different platforms.

Keyboard Commands

Many keyboard commands in Ableton Live use *modifier keys*, which are keys pressed in combination with other keys or with a mouse action. Modifiers and other equivalent keys can have different names on each platform. Table 1.1 below summarizes the modifier keys that are commonly used in shortcut operations on Mac and Windows computers.

Table 1.1 Table 1.1 Modifier keys on Mac and Windows systems

Mac	Windows
Command key	Ctrl (Control) key
Option key	Alt key
Control key	Start (Win) key
Return key	Enter key on main (not numeric) keypad
Delete key	Backspace key

File-Naming Conventions

A few differences exist in the way files are named and recognized by Mac and Windows.

File Name Extensions

For cross-platform compatibility, all Ableton Live files in a project must have a three-letter file extension added to the file name. Ableton Live Set files use the *.als* extension. WAV files have the *.wav* file extension, and AIFF files have the *.aif* file extension.

Incompatible ASCII Characters

Ableton Live file names cannot use ASCII characters that are incompatible with a supported operating system. The following characters should be avoided in order to maintain cross-platform compatibility:

/ (slash)

\ (backslash)

: (colon)

* (asterisk)

? (question mark)

“ (quotation marks)

’ (apostrophe)

< (less-than symbol)

> (greater-than symbol)

| (vertical line or pipe)

You should also avoid any special characters typed with a modifier key on Mac or Windows systems.

Review/Discussion Questions

1. Name and describe four types of production tasks that Ableton Live can be used for. (See “The Ableton Live Digital Audio Workstation” beginning on page 8.)

2. What is the frequency range of human hearing? (See “Audio Basics: Waveform, Frequency, and Amplitude” beginning on page 10.)

3. What does the frequency of a sound wave affect in terms of how we perceive the sound? How is frequency measured? (See “Audio Basics: Waveform, Frequency, and Amplitude” beginning on page 10.)

4. What does the amplitude of the sound wave affect? How is amplitude measured? (See “Audio Basics: Waveform, Frequency, and Amplitude” beginning on page 10.)

5. How does the sample rate of a system relate to the frequency of audio it can capture? What is the name of the law that specifies the relationship between sample rate and audio frequency? (See “The Importance of Sample Rate” beginning on page 12.)

6. How does the bit depth of a system relate to the dynamic range of audio it can capture? How can you estimate the dynamic range of a system? (See “The Importance of Bit Depth” beginning on page 13.)

7. What are some common digital connections available on audio interfaces? What type of connector jack does each use? (See “Recording in Digital Format” on page 14.)

8. Name some audio interfaces that are available for use with Ableton Live. (See “Audio Interface Options” beginning on page 16.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 1

Selecting Your Audio Production Gear

Activity

In this exercise, you will define your audio production needs and select components for a home studio based around Ableton Live software. By balancing your wants and needs against a defined budget, you will be able to determine which hardware and software options make sense for you.

Duration

This exercise should take approximately 10 minutes to complete.

Goals/Targets

- Identify a budget for your home studio
- Explore microphone options
- Explore audio interface options
- Explore speakers/monitoring options
- Consider other expenses
- Identify appropriate components to complete your system

Getting Started

To get started, you will create a list of needs and define an overall budget for your home studio setup. This budget should be sufficient to cover all aspects

of your initial needs for basic audio production work. At the same time, you'll want to be careful to keep your budget realistic so that you can afford the upfront investment. Keep in mind that you can add to your basic setup over time to increase your capabilities. Use the table below to outline your basic requirements and to serve as a guide when you begin shopping for options. Place an **X** in the appropriate column for your expected needs in each row.

Do not include MIDI gear in this table, as we will address that separately.



This exercise assumes that you own a compatible computer with built-in speakers for playback. Do not include a host computer in this table.

Function or Component	Not Required	Minimum Configuration	Maximum Configuration for Ableton Live Intro
Audio Interface: Output Channels (Playback)	-	1–2 Inputs	8 Inputs
Audio Interface: Innut Channels (Recording)	-	2 Outputs (for stereo playback)	8 Outputs (for stereo playback and output to external gear)
Output Device(s)	Built-in Computer Speakers	Headphones	Stereo Monitor Speakers
Input Device(s) (Microphones)	-	USB Microphone	XLR Microphone(s) (specify type and number)

Accessories and Other (List or Describe) (Stands, Soundproofing, Software Add-Ons, etc.)			

Identifying Prices

Your next step is to begin identifying prices for equipment that will meet your needs. Using the requirements you identified above as a guide, do some Internet research at an online music retailer of your choice to identify appropriate options for each of the items listed in the table below. You may also want to browse some manufacturers' websites for more information.

Component	Manufacturer and Model	Unit Cost
Audio Interface		
Microphones		
Headphones		
Monitor Speakers		
Accessories/Other		
TOTAL		

Finishing Up

To finalize your purchase decisions, compare the total in the table above to the budget you allocated. If you find that your budget is not sufficient to cover the total cost, you will need to determine which purchase items you can postpone or consider bundle options. On the other hand, if you have money left over in your budget, you can consider upgrade options.

Chapter 2

Getting Started with Ableton Live

This chapter covers the basic requirements to get up and running with Ableton Live. It reviews the Ableton Live software components and installers, as well as the file structure used for Ableton Live Sets. It also introduces the basic user interface for the software and reviews menu operations. The second half of this lesson provides an overview of the main views in Ableton Live.

Learning Targets for This Chapter

- Understand the components included with an Ableton Live system and how to access installers
- Recognize the basic Ableton Live project file structure
- Understand basic Ableton Live software and plug-in options
- Power up an Ableton Live system
- Navigate the Ableton Live menu system to locate common commands
- Recognize and work in the main Ableton Live views

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform. Sign up for free at ElementsED.com.

This chapter presents an overview of basic Ableton Live operations and functions. You will be introduced to the filing structure that Ableton Live uses for its Sets and backups, the steps required to start up an Ableton Live system, and the primary elements of the main Ableton Live views.

Target Systems

Although most of the concepts discussed in this book are applicable to all Ableton Live systems, the book is specifically written for Ableton Live 10.1 *Standard* software. While any Ableton Live system can be used with this book, certain menus, commands, and functions may differ from one configuration to another. Additionally, Ableton Live Suite users will have access to various features and devices that may not represented in this book.

Because many users will be new to Ableton Live 10, new features introduced in Ableton Live 10 are generally identified as such in the text. All descriptions are based on the user interface and functionality in Ableton Live 10.1, unless otherwise noted.



You can try Ableton Live 10 free for 30 days by visiting the Ableton website at https://www.ableton.com/trial/.

Registering Your Software

Before you can install Ableton Live, you'll need to create an account on Ableton.com and enter your product registration code. If you purchased a boxed version of Ableton Live or Push, your registration code will be included in the box. If you purchased a digital download of Ableton Live from a reseller, your registration code can be found in your confirmation email.



If you purchased Live through Ableton.com, the software will already be registered to your account.

If you've never registered an Ableton product, do the following:

1. Go to Ableton.com and create an account. You will be taken directly to the product registration page shown in Figure 2.1.

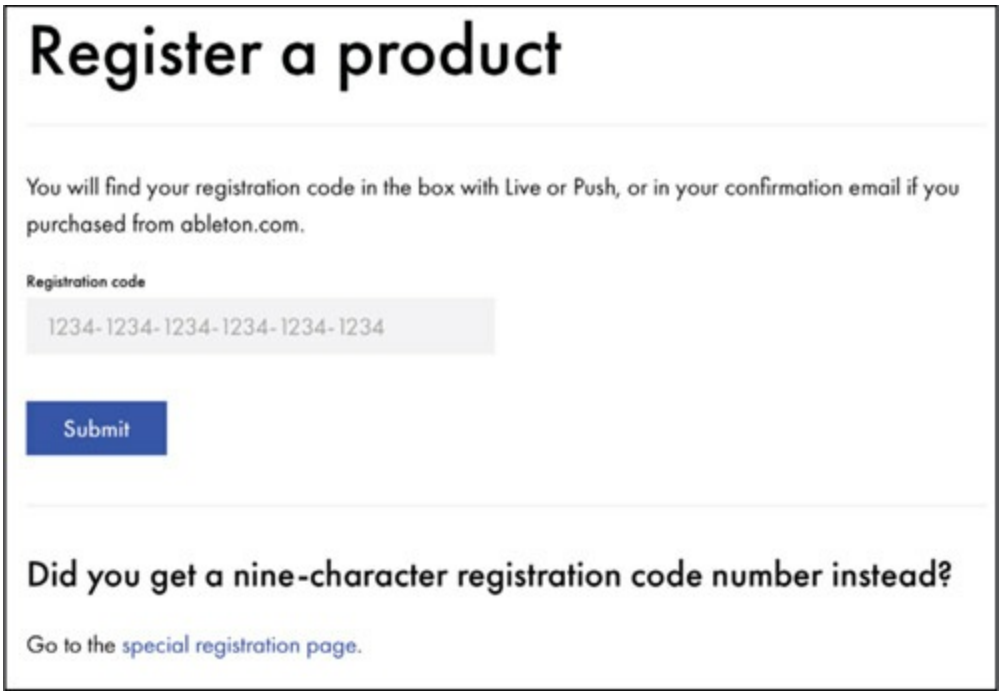


Figure 2.1 The product registration page on Ableton.com

2. Enter your registration code and click the **SUBMIT** button.

If you've previously registered an Ableton product, do the following:

1. Go to Ableton.com and log in to your account.

2. Click on the **ACCOUNT** link in the upper right corner of the webpage. The Account page will appear, displaying a Licenses pop-up near the top.



Figure 2.2 The Licenses pop-up menu in the user's Account page at Ableton.com

3. Click on the pop-up menu next to Licenses and choose **REGISTER A PRODUCT**. You'll be taken to the registration page (shown in Figure 2.1) where you can register your new or upgraded product.

Software Installation and Operation

Ableton uses separate installers for each version and edition of Ableton Live. Each Ableton Live installer also installs a variety of included software devices providing additional functionality. Supplemental devices and content (known as “packs”) may be included as well, but these will require separate installation through the Ableton website or directly within the Ableton Live application (Ableton Live 10 and later only).



Ableton uses the term *version* to differentiate numbered versions of the software such as Live 9 or Live 10. Ableton uses the term *edition* to differentiate between the different flavors of Live, such as Intro, Standard, and Suite.

To access the installers for Ableton Live, do the following:

1. Go to Ableton.com and log in to your account.

2. Click on the **ACCOUNT** link in the upper right corner of the webpage. The Account page will appear.

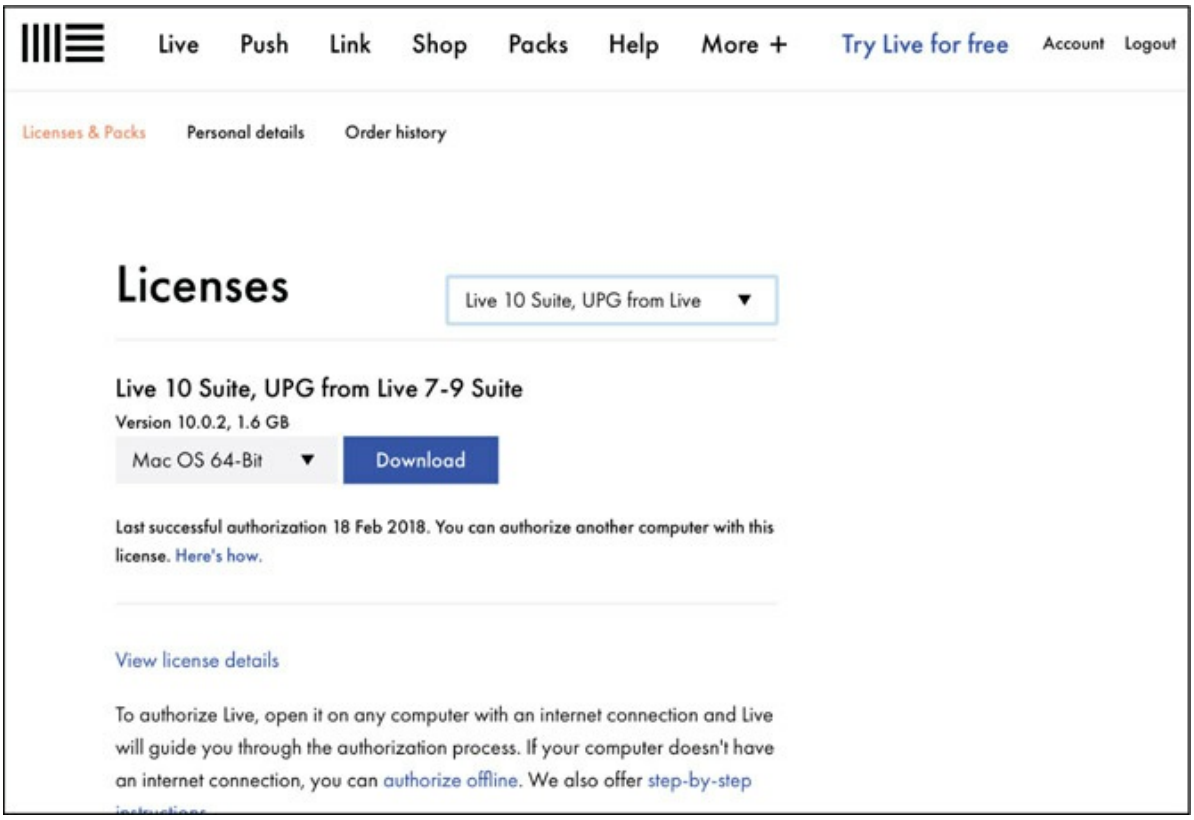


Figure 2.3 The Account page on Ableton.com

3. Generally, the most recent Ableton Live installer is shown by default. If necessary, you can click on the pop-up menu next to Licenses and choose an alternate version of Ableton Live.

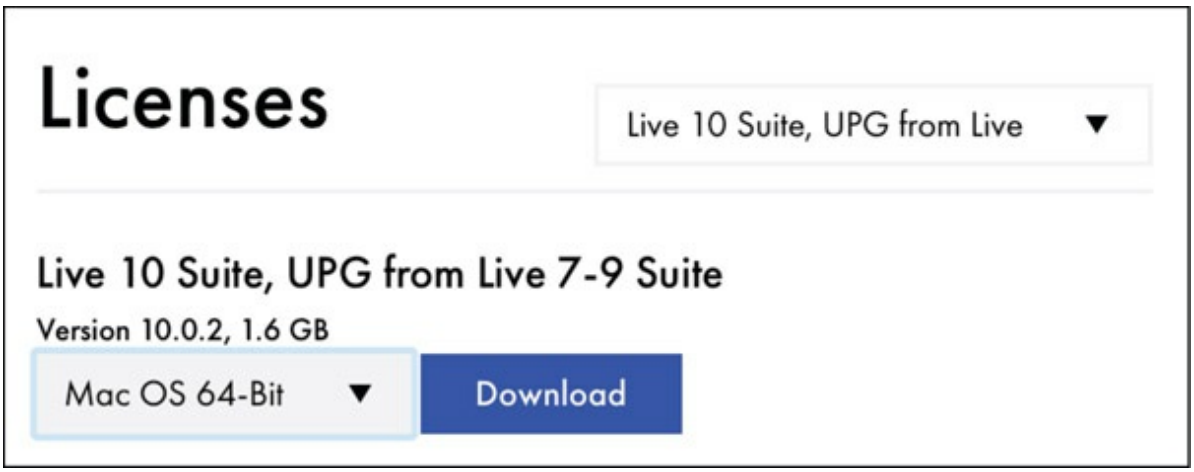


Figure 2.4 Viewing the available installers in your account

4. Choose the appropriate operating system for your computer (Mac or Windows), and click on the **DOWNLOAD** button to download the installer.

Included Devices and Packs

Devices are special-purpose software components that provide additional signal processing and other functionality within Ableton Live. Ableton devices come in three varieties: instruments, audio effects, and MIDI effects.



Other digital audio workstation software such as Pro Tools, Logic Pro X, and Cubase use the term *plug-in* for the processors used on a track. Ableton uses the term *device* to describe the built-in processors included with Live, and the term *plug-in* to describe optional third-party processors in VST or AU format.

Ableton Live Intro comes with a small set of Ableton devices. Ableton Live Standard adds an instrument and a variety of audio effects. Ableton Live Suite comes with the full complement of Ableton devices and also includes Max for Live.

Audio & MIDI Effects Devices

Ableton audio effects devices include a standard assortment of EQ, dynamics, reverb, delay, modulation, and harmonic effects, and many more specialized devices that can be used to create unique and exotic results. While Live Intro offers just the essentials, more than 46 audio effects are included in Ableton Suite!

Ableton MIDI effects devices are also quite varied and include arpeggiation, chord creation, velocity and pitch manipulation, and several more. The audio and MIDI effects devices installed with Ableton Live are described in Appendix A, “Ableton Live Audio and MIDI Effects.”

Instrument Devices

Ableton instrument devices are a collection of virtual instruments that use a MIDI input to create sound. Live Intro offers a basic set of instruments, including percussion devices (Drum Rack and Impulse), a basic sampler (Simpler), and a device for organizing and controlling instruments (Instrument Rack). Live Standard offers the same set and adds the External Instrument device that can be used to integrate hardware synthesizers or drum

machines into your setup.

Live Suite includes all of Ableton's instrument devices. In addition to those mentioned above, this set includes an analog synth (Analog), an FM and additive synth (Operator), a wavetable synth (Wavetable), a full-featured sampler (Sampler), and many more.

A description of the included instrument devices is provided in Appendix B, "Ableton Live Software Instruments."

Installing Additional Packs

Depending on the edition of Ableton Live that you have purchased, you may be entitled to additional devices and content that are not automatically installed. Ableton builds these additional items into "Packs" that can be installed manually or through the Ableton Live application (Ableton Live 10 and later only).

Installing Packs Manually

For all versions of Ableton Live, you can go to Ableton.com and download packs to manually install.

To access the installers for Ableton Live packs, do the following:

1. Go to Ableton.com and log in to your account.
2. Click on the **ACCOUNT** link in the upper right corner of the webpage.
3. Scroll down past the Ableton Live application installers until you see Packs. Click the link to the right to see all available packs (**ALL**) or only the packs that you have not previously downloaded (**NOT YET DOWNLOADED**).

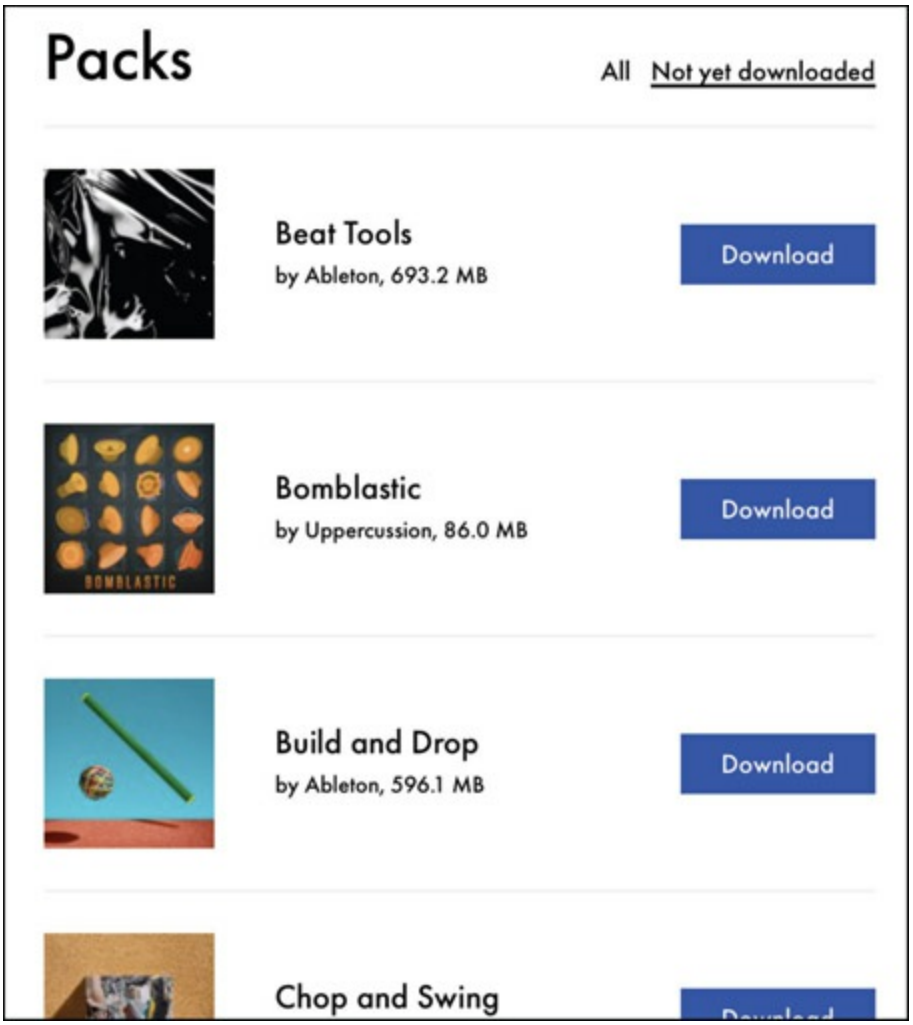


Figure 2.5 Viewing available packs in an Ableton user account

4. Click on the **DOWNLOAD** link to download a specific pack.

5. Once the pack has downloaded, double-click on the installer. The Ableton Live application will automatically install the pack on your computer.

Installing Packs in the Ableton Live Application

Users running Ableton Live 10 and later have the ability to download and install packs directly from inside of the Ableton Live application.

To download and install packs inside the Ableton Live application:

1. Verify that the Show Downloadable Packs preference is enabled:

- Choose **LIVE** > **PREFERENCES** (Mac) or **OPTIONS** > **PREFERENCES** (Windows) to open the Preferences window.
- Click on the **LIBRARY** tab on the left side of the window.
- Set Show Downloadable Packs to On by clicking on the toggle display.

2. Close the Preferences window.
3. Make sure that the Ableton Live Browser is displayed (**VIEW** > **SHOW BROWSER**).

The Browser is discussed in detail in Chapter 5 of this book.

4. Click on **PACKS** under the Places section in the Browser sidebar. The display will switch to show the packs that are available to download and install.
5. Click the triangle next to Available Packs at the top of the content pane (the **NAME** column) to see the packs that are available in your account. (See Figure 2.6 on page 31.)
6. Click the download arrow next to the desired pack. The pack installer will begin downloading to your computer. Once the pack has downloaded, the **INSTALL** button will appear.
7. Click **INSTALL** to automatically install the pack on your computer.

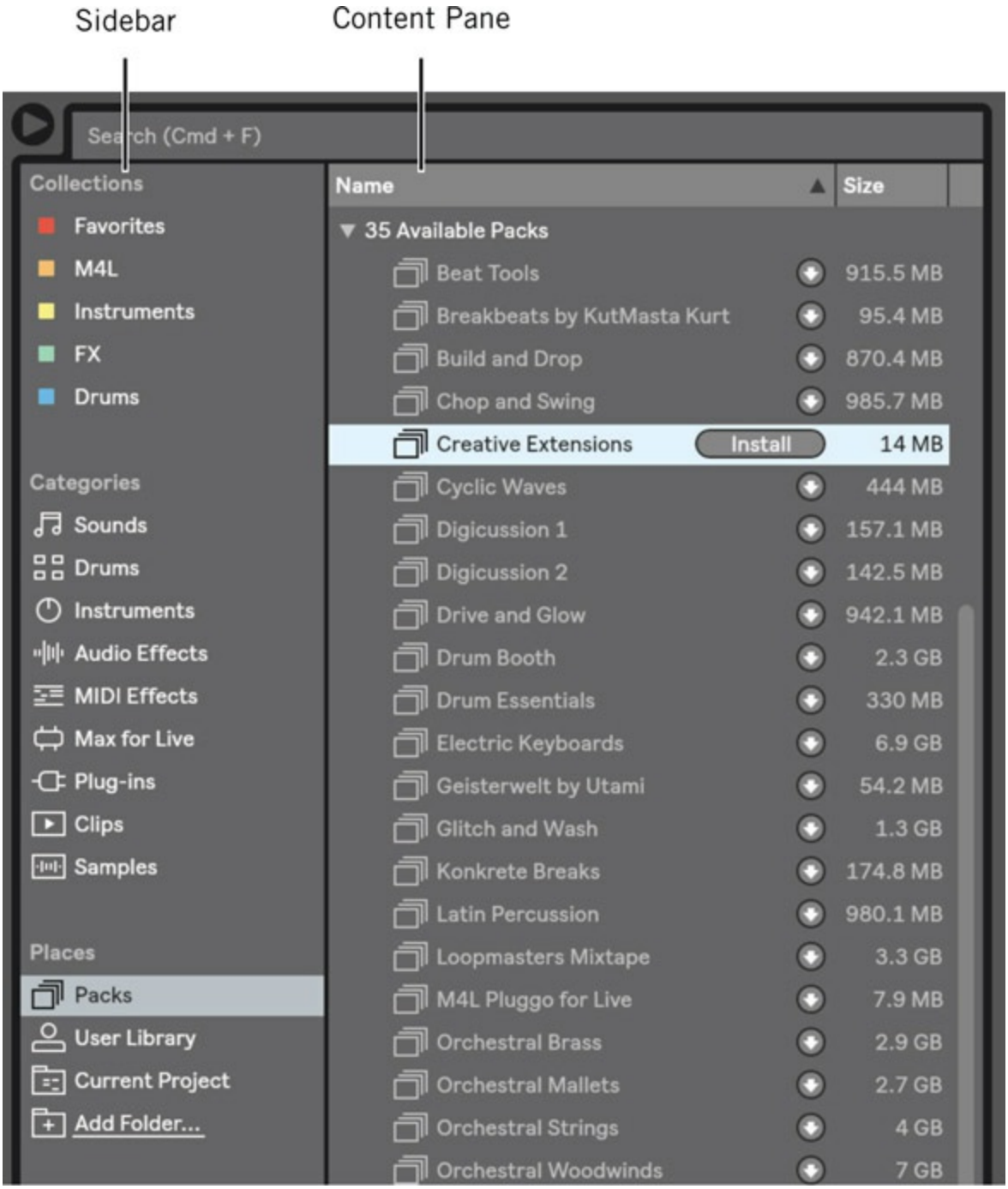


Figure 2.6 Viewing available packs in the Ableton Live application

Ableton Edition Comparison

As mentioned in Chapter 1, Ableton provides three separate *editions* of the Ableton Live application: Live Intro/Lite, Live Standard, and Live Suite. Each edition offers a different feature set, with the Intro edition offering a limited feature set, the Standard edition offering the full set of key features

but a limited number of devices, and the Suite edition adding the full collection of Ableton devices, including Max for Live.

For most users, the major limitation of Ableton Live Intro will be its limit of 16 audio and MIDI tracks. The smaller collection of devices in Intro can also be a limiting factor, but many users will opt to purchase third-party plug-ins from manufacturers such as Native Instruments and others, so they may not require the Ableton devices. Note that Ableton Live Intro does support video import and export.

Both Ableton Live Standard and Ableton Live Suite support an unlimited number of audio and MIDI tracks. The major difference here is the number of included devices and access to Max for Live. The impact these additional features will have on your personal production workflow will depend on your collection of third-party plug-ins. Additionally, Ableton Live Standard and Suite can import movies in Apple QuickTime format (.mov) to be used as video clips, making these editions ideal for scoring and sound design for picture. Be sure to consider your long-term goals before committing to a particular edition of Ableton Live.

Table 2.1 below summarizes the features included in each edition of Live.

Table 2.1 Ableton Live Editions

Edition:	Intro/Lite	Standard	Suite
Key Features	Limited	Full	Full
Audio & MIDI Tracks	16	Unlimited	Unlimited
Send & Return Tracks	2	12	12
Software Instruments	4	5	15
Audio Effects	21 (lntro)/12 (Lite)	34	46
MIDI Effects	8	8	16

Mono Audio Inputs	8	256	256
Mono Audio Outputs	8	256	256
Sounds	5+GB	10+GB	70+GB
Scenes	8	Unlimited	Unlimited
Video Support	No	Yes	Yes

Ableton Live File Structure

With all of the software components properly installed, you will likely be anxious to launch Ableton Live and get to work. But before you create your first Ableton Live document, or *Set*, it is helpful to understand how the software interacts with the various files that are related to the Set. Rather than storing a Set as a single file, Ableton Live stores various Set components separately and maintains a roadmap to the files it uses in a Set file. All of the files used for a project are grouped together in a *project folder*.



Additional information on creating Ableton Live Sets is provided in Chapter 4.

File Organization

When you create an Ableton Live Set, the system sets up a standard hierarchy for the Set and its associated files by automatically creating a top-level project folder. This folder contains the Ableton Live Set file (*filename.als*) as well as subfolders for various supplemental files used for the Set. (See Figure 2.7.) When you record, convert, import, or edit material, specific files will appear in each of these subfolders.

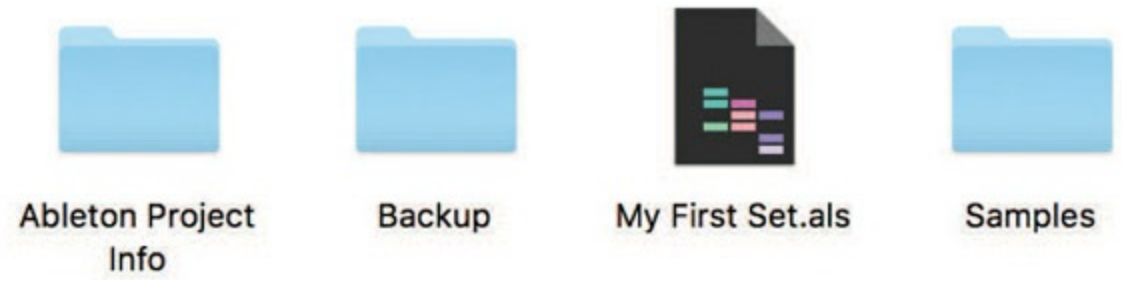


Figure 2.7 Ableton Live project folder hierarchy

Ableton Live keeps related files together in this hierarchy to facilitate backups of Sets and transfers between computers.

Project Components

The types of files that Ableton Live generates and stores in each folder in the hierarchy are described in the following sections. Ableton Live creates many of these files automatically as you work on a project, although some are generated by export operations only.

Ableton Live Set

A *Set* is the document that Ableton Live creates when you start a new project. Ableton Live creates this file (along with various subfolders) inside a project folder of the same name. Set files created in Ableton Live are recognizable by their *.als* extensions.

The Set file contains a map of all the tracks, audio files, video files, settings, and edits associated with your project. Set documents can be saved and reopened, recalling a given project with all of its edit information, input/output assignments, and links to the associated media files. You can easily copy and rename Set documents, allowing you to save alternate versions of a project without changing the source audio.

Ableton Project Info Folder

Each Ableton Live Set you create will also have an associated *Ableton Project Info* folder created inside the project folder. This folder houses a configuration file that contains important project information and uses a *.cfg* extension. You will not need to modify this file when working with Ableton

Live.

Backup Folder

Ableton Live will automatically create a Backup folder when you first save a Set. This folder will contain the 10 most recent prior saves of the Set.



Backups are enabled in Ableton Live by default. There is no way to disable this feature or change the frequency or number of backups.

Samples Folder

The Samples folder is where imported and recorded audio clips are saved while working with Ableton Live. You may not see the Samples folder when you first create a new Set. This folder is created only when you record audio or you collect files for export.

When you record audio into an unsaved Ableton Live Set, each recording is initially stored in a Temp Project folder. Once the Set is saved, the files are moved from the temporary location to a folder inside the Samples folder titled Recorded.



You can change the location of the Temp folder using the File/Folder tab of the Live Preferences window.

Ableton Live natively supports audio files in either the WAV or AIFF format. For compatibility purposes, WAV is typically the best choice on both Mac and Windows systems.



Audio that you record in Ableton Live is saved *only* in the Samples/Recorded folder; it is not saved in the Set file (.als). When transferring projects between systems, be sure to copy over the entire top-level project folder in order to include all associated audio files and other material needed for the project.

When you import audio or video files into Ableton Live, the imported files are not automatically copied to the Samples folder. Instead, Ableton Live will

simply play the files from their original location on your computer's hard drive. If you use the **File > Collect All and Save** command, Ableton Live will create a folder inside the Samples folder titled Imported and will copy the imported files to that location.



It's a good idea to use Collect All and Save when you finish an Ableton Live project to ensure that all of the associated media files are in the project folder. This makes it easy to archive the project and open it again at a later date.

MIDI Files

MIDI data is normally stored within the Ableton Live Set; as such, no MIDI files will exist outside of the Set document. However, MIDI files can be exported from Ableton Live using the **File > Export MIDI Clip** command. Exported MIDI files can be recognized by their *.mid* extensions.

Starting Ableton Live

Because Ableton Live systems are often comprised of both hardware and software, preparing your system for use might involve more than simply turning on your computer and launching the Ableton Live application. The larger the system, the more important it becomes to follow a specific startup sequence.

Powering Up Your Hardware

When starting a computer–based audio system, it's important to power up the system components in the proper order. Starting components out of sequence could cause a component to not be recognized, prevent the software from launching, or cause unexpected behavior.

The recommended sequence for starting any computer–based audio system is as follows:

1. Start with all your equipment (including your computer) powered off.

2. Turn on any external hard drives that require external power and wait

about 10 seconds for them to spin up to speed.

3. Turn on any MIDI interfaces and MIDI devices that require external power (including any MIDI control surfaces) as well as any synchronization peripherals, if used.

4. Turn on your audio interface (if not bus powered). Wait at least 15 seconds for the audio interface to initialize.

5. Start your computer and launch Ableton Live.

6. Turn on your audio monitoring system, if applicable.

Many audio interfaces get their power from the computer (via a USB port or other connection); these interfaces do not need to be powered up in advance.

Launching Ableton Live

Ableton Live software can be launched by double-clicking on the application icon on the system's internal drive or by double-clicking on a shortcut to the application. On Windows systems, the Ableton Live application is typically installed under C:\Program Files\Ableton\Ableton Live\AbletonLive10.exe, and a shortcut is placed on the desktop. On the Mac, Ableton Live is typically placed under Applications\Ableton Live 10.app, and an application shortcut is placed in the dock.



Figure 2.8 The Ableton Live application icon

On Windows systems, Ableton Live may also be available from the Start menu at the lower-left corner of the display.

When you first launch Ableton Live, the application will prompt you for authorization.

Authorizing Live

Ableton Live software uses copy protection, as do many other software products and plug-ins. The Ableton Live license is intended for a single user. However, Ableton permit you to install the Live application on up to two computers belonging to the registered user.

The first time you launch Ableton Live, an authorization dialog box will appear (Figure 2.9). At this point, you must choose one of three options:

- Authorize with ableton.com
- No Internet on this computer
- Authorize later



Figure 2.9 Live's Authorization dialog box

Authorize with Ableton.com

If your computer has Internet access, you can quickly authorize Ableton Live with Ableton.com.

To authorize Ableton Live with Ableton.com, do the following:

1. Choose **AUTHORIZE WITH ABLETON.COM** from the authorization dialog box; your computer will automatically open a browser and take you to the Ableton website.

2. Log into your Ableton.com account. You'll be asked to select the license that you would like to authorize.

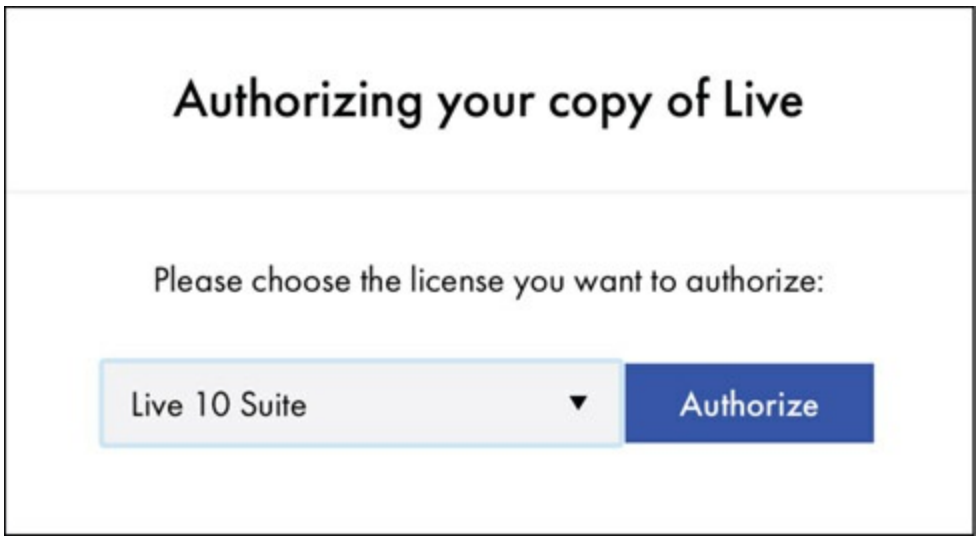


Figure 2.10 Selecting the license to authorize from your Ableton.com account

3. Assuming you have already registered Ableton Live using a registration code (see "Registering Your Software" earlier in this chapter), choose the appropriate license from the Licenses list and click the **AUTHORIZE** button. Ableton.com will send an authorization to your computer.

   This may take few seconds, during which time you will see a progress bar near the bottom of the authorization dialog box.



Figure 2.11 The progress bar in the Authorization dialog box

4. Once the authorization is complete, you will get a confirmation message. (See Figure 2.12.) Click **OK** to begin using Ableton Live!



Figure 2.12 The authorization confirmation message

Authorizing Offline

If you have no Internet access on your computer, or have difficulty authoring using the Authorize with albeton.com option, you can authorize offline.

To authorize Ableton Live offline, do the following:

1. Choose **No Internet on this Computer** in the authorization dialog box. The offline authorization dialog box will appear.



Figure 2.13 The offline authorization dialog box

2. Note the hardware code or save it to a text file by clicking the **Save** button.

3. On a computer with Internet access, log into your Ableton.com account.

4. Click the **Account** link in the upper right corner of the website.

5. Select the appropriate license from the Licenses list and then click the **Authorize Offline** link. This will take you to the Authorize Offline page.

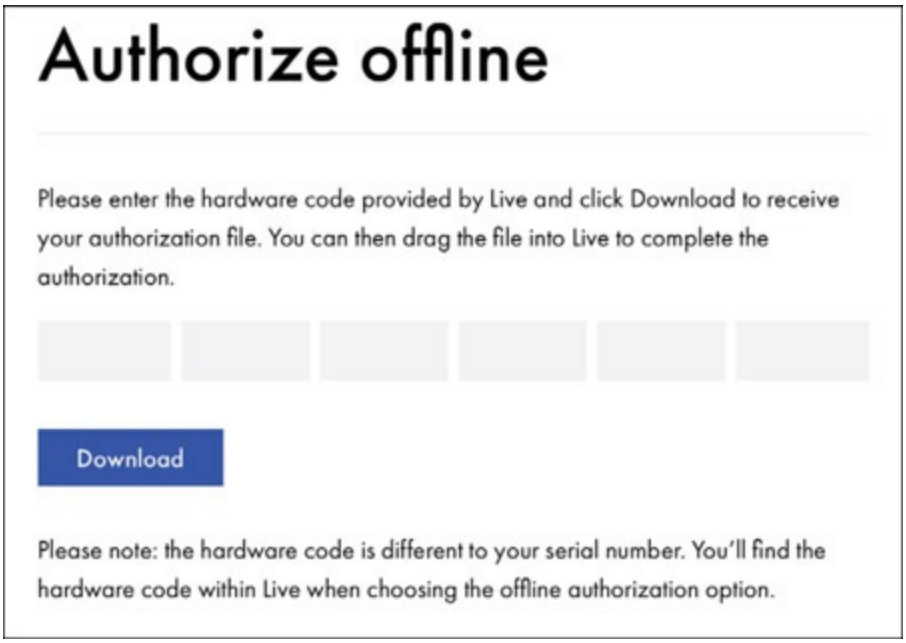


Figure 2.14 The Authorize Offline page on Ableton.com

6. Enter your hardware code and click the **DOWNLOAD** button. Ableton.com will generate an authorization file (.auz), and the file will download to your computer.

7. Transfer the .auz file to the computer that you wish to authorize.

8. Double-click the .auz file (or drag and drop it onto the Authorization dialog box) to complete the authorization process.

Authorize Later

If you choose the **AUTHORIZE LATER** option from the Authorization dialog box, Ableton Live will still launch, but saving and exporting will be disabled until you complete the authorization process.

Accessing Connected Audio Devices

When Ableton Live launches, it will generally be configured to use the default audio device on your computer. If no supported audio device is found, Ableton Live will launch with the audio engine turned off. You can immediately recognize that the audio engine is off because the **CPU LOAD METER** at the far right of the toolbar will turn red and display the word OFF.



Figure 2.15 The CPU Load Meter when the Audio Engine is off

To configure Ableton Live to use a specific audio device, such as a connected USB microphone or a separate audio interface, you will need to go to the **AUDIO** tab on the left side of the Live Preferences window (**LIVE** > **PREFERENCES** [Mac] or **OPTIONS** > **PREFERENCES** [Windows]).

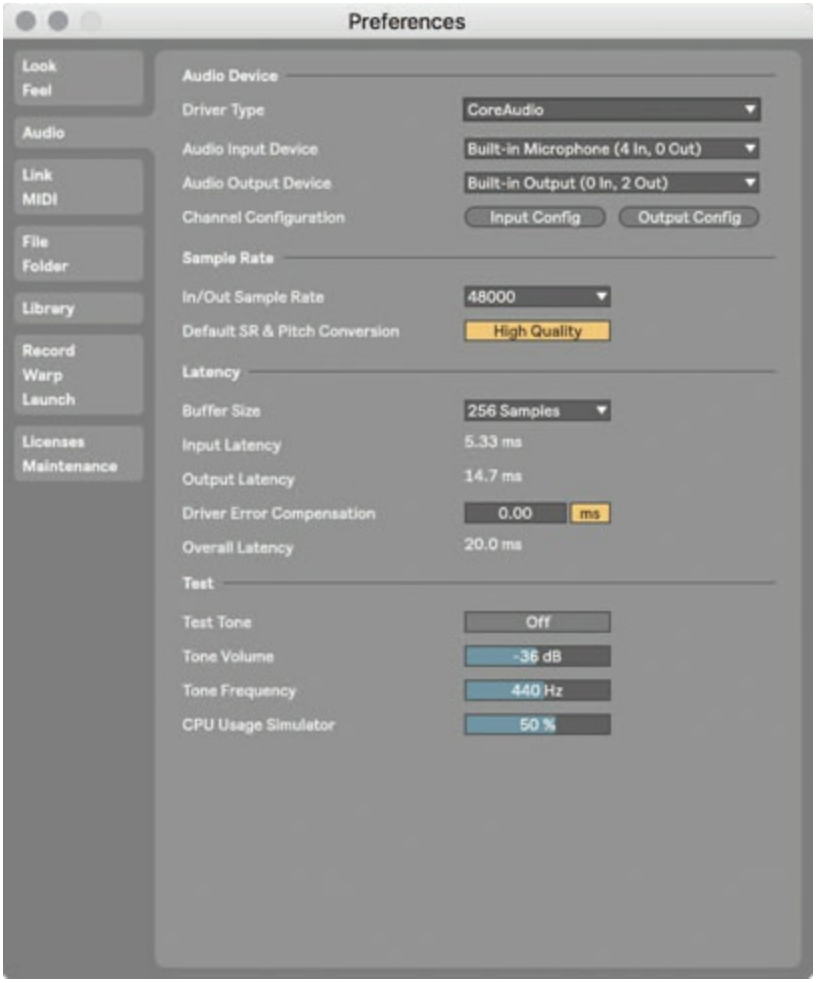


Figure 2.16 The Audio tab of the Live preferences

If your supported audio interface is connected but not recognized by Ableton Live, you may need to install or update the device drivers. Check the manufacturer's website for the latest drivers for your interface.

Optimizing Ableton Live Performance

The **AUDIO** tab of the Live Preferences window is also used to optimize the performance of the software. Like most audio software, Ableton Live utilizes the computer's processing capacity to carry out operations such as recording, playback, mixing, and plug-in processing (virtual instruments and effects). You can monitor the current CPU usage at any time using the CPU Load Meter in the upper right corner of the toolbar.



Figure 2.16 The CPU Load Meter showing 7% usage

While the default system settings are adequate in some scenarios, Ableton Live lets you adjust a system's performance using the Sample Rate and Latency settings in the Audio tab of the Preferences window.

Sample Rate Settings

The Sample Rate setting will affect the quality of the audio that is going into or out of your system through a connected audio interface. The Sample Rate setting will also determine the sample rate of audio files that result from recording on an audio track.

A higher sample rate requires more processing power than a lower sample rate, and will reduce the number of tracks and plug-ins you can run on a particular computer. For this reason, many producers prefer to work with sample rates of 44,100 and 48,000. While high-definition sample rates (such as 88,200 and 96,000) will result in better fidelity in the studio, they don't necessarily translate to better quality in final mixes that are distributed as MP3 files or streamed on services such as SoundCloud or Spotify.

Refer to the "Audio Basics: Waveform, Frequency, and Amplitude" section in Chapter 1 for details on how the sample rate affects audio quality.

Latency Settings

The Buffer Size setting in the Audio tab of the Preferences dialog box controls the size of the hardware buffer. This buffer handles processing tasks such as plug-in processing.

- Lower Buffer Size settings reduce monitoring latency when you are recording or monitoring an active input.
- Higher Buffer Size settings provide more processing power at the expense of greater monitoring latency.

As a general rule, the Buffer Size should be set as low as your project will allow, in order to minimize latency when monitoring an active input. You may need to change the setting later as your project becomes more complex.

Reducing Latency on Tracks with an Active Input

Ableton Live offers an additional option for reducing latency on tracks with an active input. By activating **Options > Reduced Latency when Monitoring**, any track with an input set to **In** or **Auto** will have the smallest possible amount of latency given the current configuration of the Audio Engine. However these tracks may be out of sync with other tracks in your set, such as Return tracks.



Enabling this setting turns off delay compensation for input-monitored tracks.

Modifying Performance Settings

Adjustments to the Sample Rate and Latency settings can be made in the in the Audio tab of the Preferences window, as follows:

1. Choose **Live > Preferences** (Mac) or **Options > Preferences** (Windows).
2. Select the **Audio** tab.
3. From the Sample Rate section, click the **In/Out Sample Rate** pop-up menu and choose the desired sample rate.

4. From the Latency section, click the **Buffer Size** pop-up menu and select the buffer size in samples—lower the setting to reduce latency; raise it to increase processing power for plug-ins.

5. Click **OK**.

The Ableton Live Menu Structure

Before beginning to work on an Ableton Live Set, you should have some basic familiarity with the software interface, including the menu structure and views.

Among the first things you see upon launching Ableton Live is the menu system across the top of the screen. Learning how the menus are organized will save you a lot of time when you are trying to find a specific Ableton Live function. Following is a brief description of each menu.

File Menu

File menu commands let you create and maintain Ableton Live Sets and perform other file-based commands. The File menu includes options for opening, creating, and saving Sets; exporting audio, video, and MIDI; installing packs; and managing files.

Edit Menu

Edit menu commands allow you to edit and manipulate the media in your current selection. The Edit menu includes options for cutting, copying, and pasting; duplicating, deleting, and renaming; splitting, consolidating, freezing, and quantizing; and performing similar operations.

Create Menu

Create menu commands generally result in the creation of new tracks in Ableton Live. The Create menu includes options for inserting Audio, MIDI, and Return tracks; processing audio and MIDI data to new tracks; inserting scenes; and importing audio files.

View Menu

View menu commands affect the display within Ableton Live windows and tracks. Most View menu commands show or hide parts of the main Ableton Live windows. Selecting a command will display a component part of a window, and deselecting the command will hide it. Zoom commands are also included in the View menu.

Options Menu

The Options menu commands let you toggle several editing, recording, monitoring, playback, and display options on/off. From this menu, you can define MIDI and key mapping, enable delay compensation, adjust launch quantization, change the solo mode, set the time ruler format, and make other similar choices.



The Options menu displays independent functions that toggle on or off. Menu items with a check mark next to them are currently on, or enabled; items without a check mark are off, or disabled. Selecting an item toggles its state on/off.

Help Menu

The Help menu allows you to show or hide the Help View and provides links to important documentation and online resources, including the Ableton Live Manual, the Ableton User Forum, and your Abeton.com account. You can also check for Ableton Live updates from this menu.

Ableton Live Views and Sections

Ableton Live software displays either one or two main windows, depending on your settings (**VIEW > SECOND WINDOW**). Within these windows are a number of available “views” and “sections” that are used to accomplish various tasks.

The primary components that you will need to be familiar with to begin working with Ableton Live include the following:

- The Control Bar at the top of the window(s)
- The two main views: the Session View and the Arrangement View
- The two optional displays: the Browser display and the Detail View.

In addition, the Detail View features two separate view options: the Device View and the Clip View.

The Browser display and Detail View can be shown or hidden separately in the Session View and the Arrangement View.



Figure 2.18 The Control Bar (top), Browser (left), Session View (right), and Detail View (bottom; Device View shown)

Session View

The Session View is the view that sets Ableton Live apart from most audio software. When the first version of Ableton Live was released in 2001, it was the Session View that caught the attention of music producers around the world. There was nothing else like it!

The Session View provides a number of controls grouped into sections. Several of these sections, such as the Clips Slots section and the Scenes section, are unique to the Session View. Other sections, such as the In/Out,

Sends, and Mixer sections, can also be found in the Arrangement View.

Clip Slots Section

The Clips Slots section of the Session View offers a collection of slots into which audio and MIDI clips can be placed. When a clip is played (or "launched") from a Clip Slot, the signal from the audio clip or MIDI instrument plays through the signal path comprised of the track sections that follow it (see below).

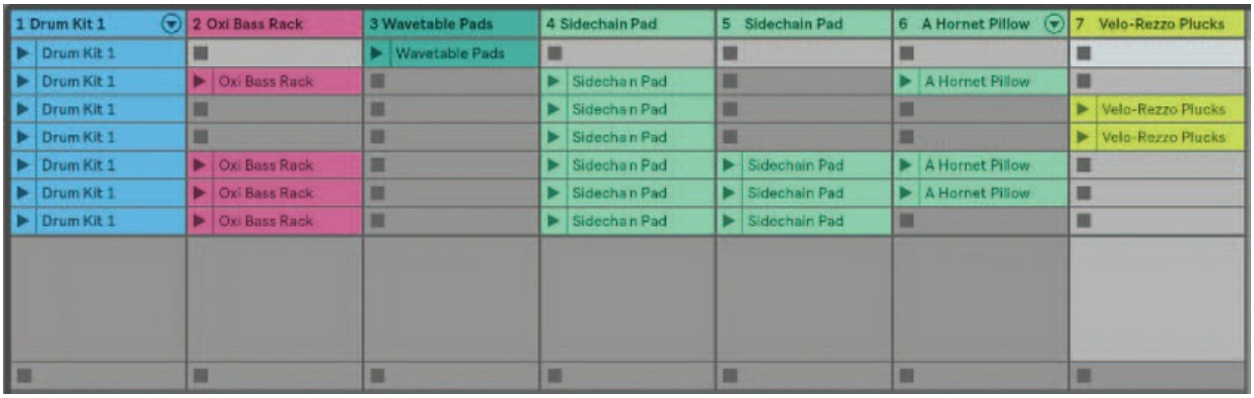


Figure 2.19 Clip Slots section of the Session View

Scenes Section

The Scenes section of the Session View is displayed on the Master track and allows you to launch an entire row of clips (a "scene") all at once. Scenes are typically used to navigate through sections of a single song (verse, chorus, etc.) during production, or to move between complete songs during performance.



Figure 2.20 Scenes section of the Session View

In/Out Section

The top portion of each channel strip in the Session View provides controls for routing signals for the track. Depending on the type of track, these controls may include Audio Input and Output selectors, MIDI Input and Output selectors, and Monitor controls.

Audio Input and Output selectors are used to route input and output signals from your audio interface for recording or playback. MIDI Input and Output selectors are used to route MIDI data from MIDI controllers to either instrument plug-ins or external MIDI hardware. The Monitor controls determine whether you will hear the live track input or playback of an existing clip on a given track. The Monitor controls can generally be left on the **AUTO** setting, which will automatically switch from monitoring live input when a track is recording (or record-armed in stop) to monitoring clip playback when the track is in play.



Figure 2.21 The In/Out section of the Session View

Mixer Section

The Mixer section of the Session View provides a mixer-like environment for working with tracks. In the Mixer section, tracks appear as mixer strips (also called *channel strips*). Each track displayed in the Mixer section has controls for panning and volume. The Mixer section also provides buttons for enabling record, activating/deactivating tracks, and toggling solo on and off.



Figure 2.22 The Mixer section of the Session View

The Pan controls can be used to position the output of a track within a stereo field. The Track Activator, Solo, and Arm Recording buttons can be used for mute, solo, and record arm functions, respectively, during record and playback. The Volume slider can be used to adjust the playback/monitoring

level of a track.

The Volume slider in the Mixer Section does not affect the input gain (record level) of a signal being recorded. The signal level must be set appropriately at the source or adjusted using a preamp or gain-equipped audio interface.

Additional Session View Sections

Other Session View sections include the Sends Section, Returns Section, Delay Section, and X-Fade Section. These are discussed in more detail in later chapters.

Arrangement View

The Arrangement View provides a timeline display of audio, MIDI data, video, and mixer automation for recording, editing, and arranging tracks. This view displays waveforms for audio clips and MIDI note data for MIDI clips.

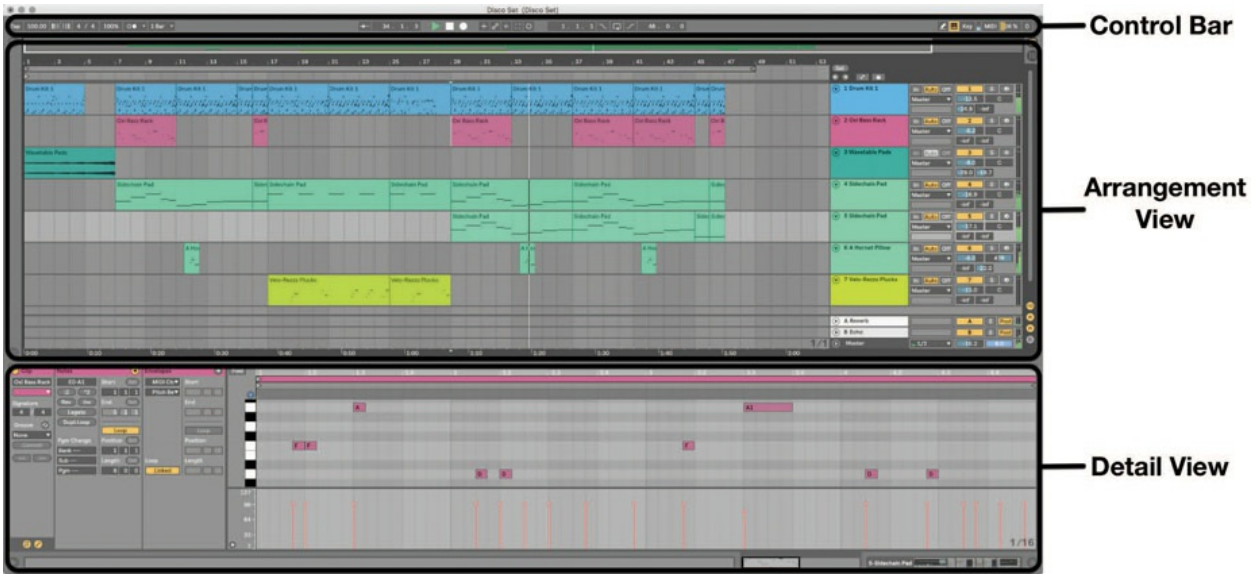


Figure 2.23 The Control Bar, Arrangement View, and Detail View (showing Clip View)

Each Audio and MIDI track in the Arrangement View features some of the same track sections as the Session View, including the In/Out section and Mixer section.



You can toggle between the Session View and Arrangement View at any time by pressing the TAB key.

The Browser

The Browser can be displayed on the left–hand side of the main window. It can be accessed when either the Session View or Arrangement View is active.

The Browser is used to locate clips, samples, instruments, audio and MIDI effects, plug-ins, and Max for Live devices. The Browser is also used to access and install Ableton Packs (in Ableton Live 10 or later), to browse through user settings and current project files, and to quickly access folders located anywhere on your computer.



To display the Browser, choose VIEW > SHOW BROWSER in either the Session View or the Arrangement View.

The Browser display is divided into two sections, with the sidebar on the left and the content pane on the right. You can resize the sections by dragging the divider line horizontally.

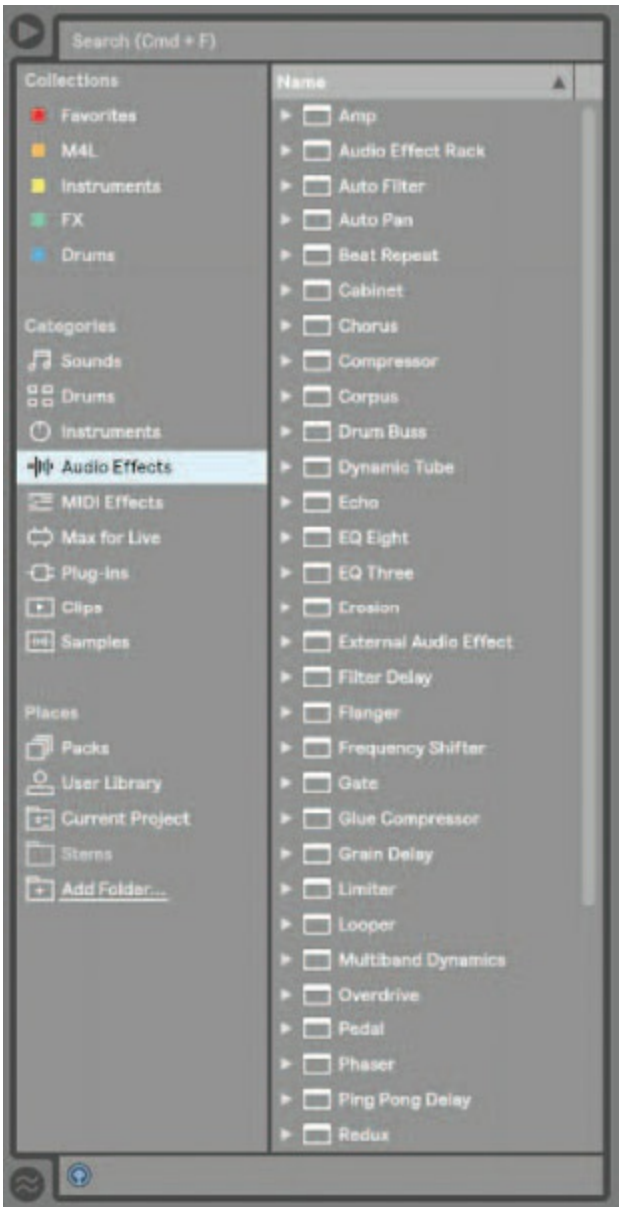


Figure 2.24 The Ableton Live Browser display showing the sidebar (left) and content pane (right)

Detail View

The Detail View section can be displayed at the bottom of the main window. Like the Browser, the Detail View is accessible regardless of whether the Session View or Arrangement View is currently active. The Detail View section is always operating in one of two modes: Device View or Clip View.

To toggle the Detail View section between Device View and Clip

View, press SHIFT+TAB.

Device View

When the Detail View section is set to Device View (Figure 2.25), it will display any devices inserted onto the currently selected track. These devices can include instruments, audio effects and MIDI effects, as well as Max for Live devices.



Figure 2.25 The Device View

Clip View

When the Detail View section is set to Clip View, it will display the currently selected audio clip's waveform in the Sample Editor, or the currently selected MIDI clip's data in the MIDI Note Editor. The Clip View also features several property boxes that can be used to modify various aspects of a clip.

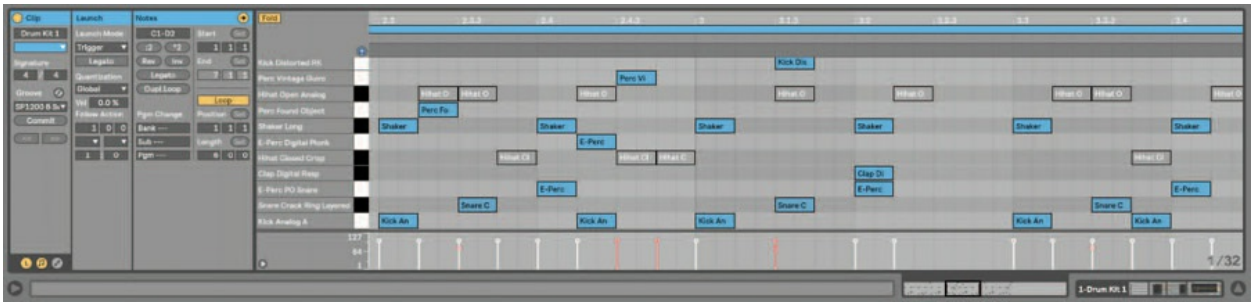


Figure 2.26 The Clip View

Info View

The Info View is an additional display that is available as part of the Detail View. When active, the Info View appears at the far left side of the Detail View. The Info View provides a brief description of any element that you position the mouse over within the Ableton Live user interface.



Figure 2.27 The Info View

Control Bar

Ableton Live provides a variety of controls in the Control Bar area at the top of the main window. These controls cover a wide range including tempo, time signature, and quantization settings; transport and looping controls; edit settings; keyboard and MIDI mapping; and performance indicators.


Figure 2.28 The Control Bar

Overview (Arrangement View only)

The Overview section appears just below the Control Bar in the Arrangement View. It appears as a miniature representation of all of the clips that are currently on tracks in the Arrangement View. You can click in the Overview to quickly navigate to a new position in the Set. You can also click and drag within the Overview to change the horizontal or vertical zoom level.


Figure 2.30 The Overview Arrangement View

For details on zooming with the Overview, see “Zooming in



Arrangement View" in Chapter 3 of this book.

Rulers and Scrub Area (Arrangement View only)

Rulers are horizontal strips that appear in the Arrangement View, just above and below the Track Display area. Rulers provide measurement indicators to help you identify specific locations in your Set's timeline.

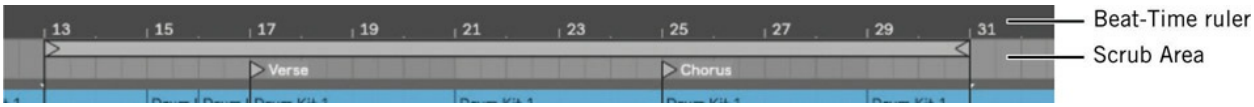


Figure 2.30 The beat-time ruler and Scrub Area in the Arrangement View

- **Beat-Time Ruler.** This ruler appears above the Track Display. It will display bar numbers when zoomed out, but will also show beats and sub-beats as you zoom in.

- **Scrub Area.** The Scrub Area sits just below the beat-time ruler and performs a number of functions. You can click in the Scrub Area to start playback from a specific location. You can also set loop start and end points by dragging the left or right side of the loop brace (or by entering values into the Loop Start and Loop Length fields in the Control Bar). In addition, you can right-click in the Scrub Area to insert a time signature change or to add a Locator.

- **Time Ruler.** The Time Ruler is a secondary ruler that appears at the bottom of the Arrangement View. The user can configure this ruler to show Time (minutes:seconds), or film and video frame positions at a variety of frame rates, including 24 fps (film), 25 fps (PAL), and 29.97 fps (NTSC). This can be helpful when working with imported video clips.

Adjusting the Display of Views and Sections

Ableton Live lets you customize the display of the windows to accommodate your needs at any given point in your project.

Either the Session View or Arrangement View must be visible at all times.

Showing/Hiding Views

You can toggle the display between Session View and Arrangement View at any time by pressing the **TAB** key. If you are displaying a second main window (**VIEW > SECOND WINDOW**), pressing Tab will swap the views in the two windows.

The secondary Ableton Live views (the Browser, Detail View, and Info View) can be shown or hidden in one of two ways.

To show or hide views, do one of the following:

- Go to the **VIEW** menu and select a view to show or and hide.

  Or

- Click the triangular Show/Hide button corresponding to the view you wish to show or hide.

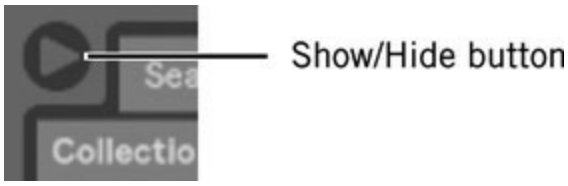


Figure 2.31 The triangular Show/Hide Button for the Browser display

Showing/Hiding Sections

Sections in both Session View and Arrangement View (such as the In/Out Section and the Mixer Section) can be shown or hidden in one of two ways.

To show or hide sections, do one of the following:

- Go to the **VIEW** menu and select the section to show or hide.

- Click the Show/Hide section buttons located in the bottom right corner of the Session View and Arrangement View.

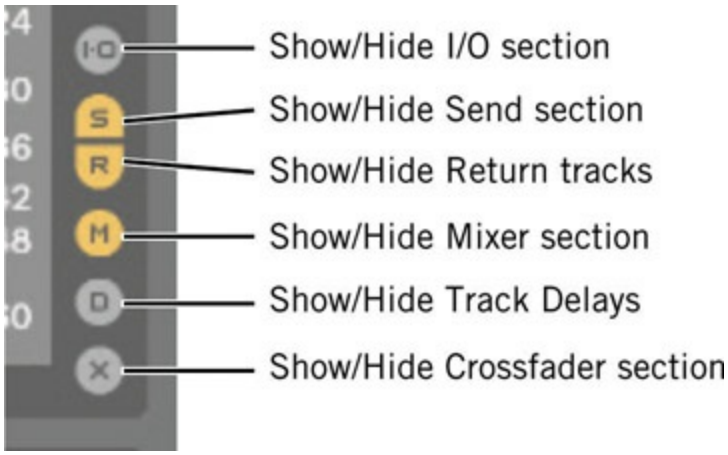


Figure 2.32 The Show/Hide Buttons for the sections in the Session View

Resizing Views

Several of the Ableton Live views can be resized to accommodate a particular task. These include the Browser display and the Detail View section.

To adjust the width or height of a view, follow these steps:

1. Position the mouse over the view separator where the cursor changes into a double-headed arrow.

2. Click and drag on the view separator to adjust its position as needed.



Figure 2.33 Clicking on the view separator to resize the Detail View section at the bottom of the Arrangement view

Review/Discussion Questions

1. What are some of the differences between Ableton Live editions? Which editions offer support for video playback? (See “Ableton Edition Comparison” beginning on page 30.)

2. Name some of the folders and files that Ableton Live creates as part of the project hierarchy. Where is the Live Set file (.als) stored? (See “Project Components” beginning on page 32.)

3. Where are imported and recorded audio files stored in the project hierarchy? (See “Project Components” beginning on page 32.)

4. Where are Ableton Live’s MIDI files normally stored? (See “Project Components” beginning on page 32.)

5. Which component should you turn on first when powering up an Ableton Live system? Which component should you turn on last? (See “Powering Up Your Hardware” beginning on page 34.)

6. What type of processing does the Buffer Size preference affect? (See “Latency Settings” beginning on page 40.)

7. What Buffer Size setting should you choose when recording audio or MIDI? (See “Latency Settings” beginning on page 40.)

8. What kinds of commands can be found under the Ableton Live Create menu? (See “The Ableton Live Menu Structure” beginning on page 41.)

9. What kinds of commands can be found under the Ableton Live Options menu? (See “The Ableton Live Menu Structure” beginning on page 41.)

10. Which Ableton Live View provides access to the Clips Slots section for launching clips? (See “Ableton Live Views and Sections” beginning on page 42.)

1. Which main Ableton Live View provides a timeline display of audio waveforms and MIDI data on horizontal tracks? (See “Ableton Live Views and Sections” beginning on page 42.)

2. Is it possible to see Pan and Volume controls for each track in both the Session View and the Arrangement View? (See “Ableton Live Views and Sections” beginning on page 42.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 2

Identifying Ableton Live Views

Activity

In this exercise worksheet, you will identify the main views in Ableton Live and their component sections. The information referenced in these questions is covered in Chapter 2.

Duration

This exercise should take approximately 10 minutes to complete.

Goals/Targets

- Identify the main views in Ableton Live
- Recognize the component sections of the main Views
- Identify the two options for viewing clips in the Detail view

Instructions

Answer the questions in this exercise by filling in the blanks on the following pages or by listing your answers on a separate sheet of paper or worksheet. Be sure to use the terminology referenced in Chapter 2 as you complete this exercise.

Components of the View that Shows Clip Slots

Refer to Figure 2.34 when answering the questions below. Refer to the section on "The Ableton Live Software Interface" in Chapter 2 for assistance.

Questions 1 through 4 refer to Figure 2.34.

1. The view shown in Figure 2.34 is called the _____ view.

2. The area labeled **A** across the top of the window is called the _____.

3. The area labeled **B** on the left side of the window is called the _____.

4. The area labeled **C** is currently showing the _____ view.



Figure 2.34 The Ableton Live View that displays the Clip Slots Section

Components of the View that Shows the Timeline

Refer to Figure 2.35 when answering the questions below. Refer to the section on "The Ableton Live Software Interface" in Chapter 2 for assistance.

Questions 5 through 10 refer to Figure 2.35.

5. The view shown in Figure 2.35 is called the _____ view.

6. The area labeled **A** at the top of the window is called the _____.

7. The area labeled **B** is currently showing the _____ view.

8. The clip displayed in the area labeled B is a/an _____ clip.

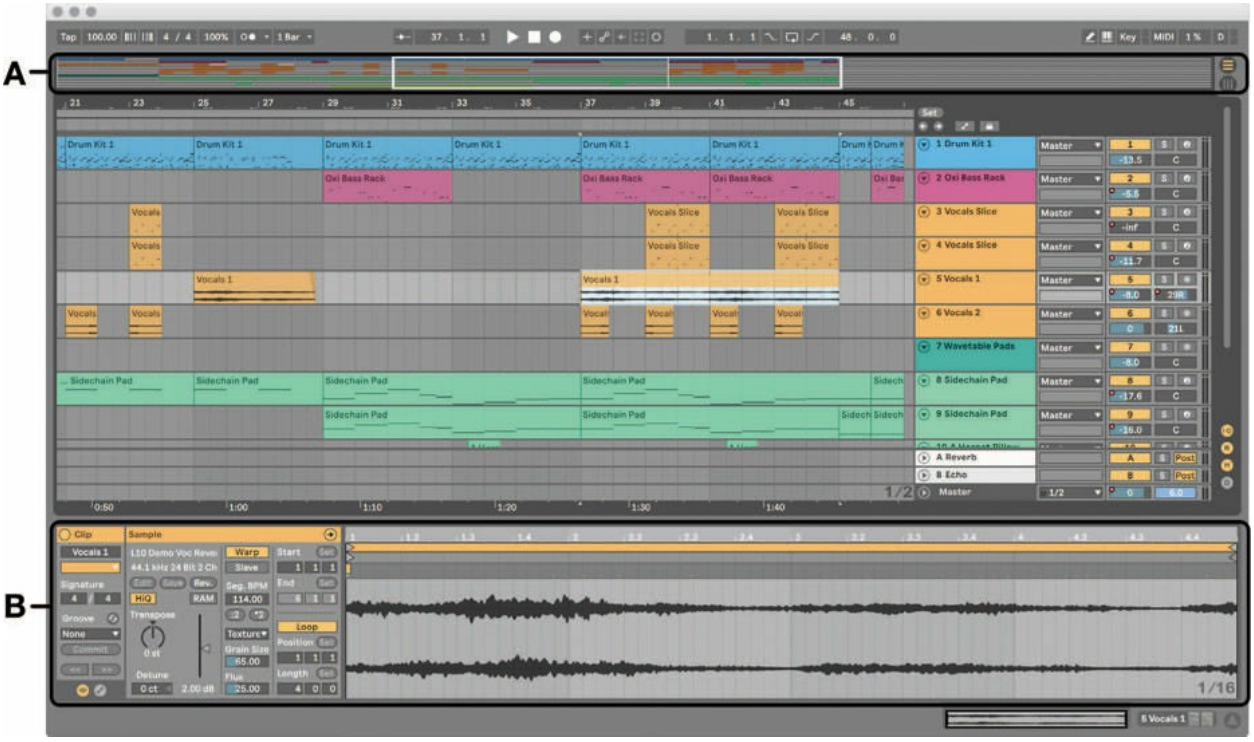


Figure 2.35 The Ableton Live View that displays clips on a timeline

Chapter 3

Ableton Live Basic Controls

This chapter provides an overview of the edit functions, zoom options, and MIDI controls used in Ableton Live's Arrangement View. It also discusses the Edit Grid available in Ableton Live, how to set the Edit Grid Value, and how to work with Rulers.

Learning Targets for This Chapter

- Understand how to activate playback in the Arrangement View
- Perform typical clip editing and zooming functions in the Arrangement View
- Configure the Edit Grid
- Work with Rulers and the Scrub Area
- Recognize the available performance controls
- Understand the difference between Edit Mode and Draw Mode in the MIDI Note Editor

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform. Sign up for free at ElementsED.com.

This chapter provides an overview of the playback functions, editing controls, display options, and rulers available in the Ableton Live Arrangement View. These primary controls are used day-in and day-out in Ableton Live for recording, playback, and basic editing of audio and MIDI material. Establishing some familiarity with the available tools, operational modes, and performance control functions early on will help you in all aspects of the work you do in Ableton Live.

Playback in Arrangement View

When you first create an Ableton Live set, the Play function in the Control Bar will serve to activate playback of the state stored in the Arrangement View. However, if you activate playback of a Clip Slot or Scene in Session View for the Set, the Play function will generally switch to activate playback of the current state of the Session View instead. This may cause the media on tracks in the Arrangement View to appear dimmed (grayed out) and to not play back as expected.

When this happens, the **BACK TO ARRANGEMENT** button will light in red to indicate that one or more tracks are currently not playing the Arrangement, but are playing a clip from the Session View instead. This button is available in both the Session View and the Arrangement View.



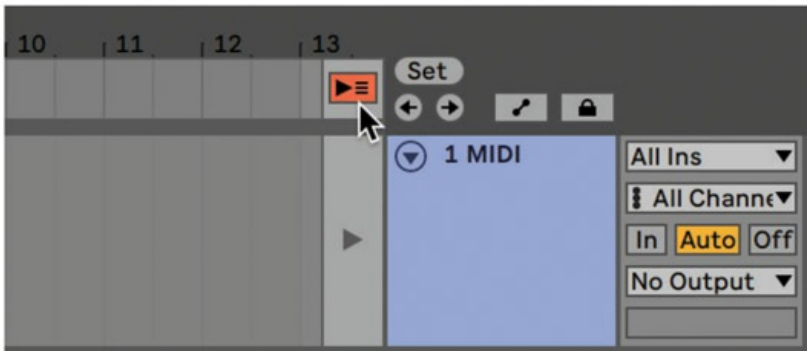


Figure 3.1 The Back to Arrangement button in the Session View (left) and in the Arrangement View (right)

To reactivate playback for the Arrangement View, click the lit **BACK TO ARRANGEMENT** button in either the Session View or the Arrangement View so that it becomes unlit (Session View) or disappears (Arrangement View).



You can also press function key F10 to activate the Back to Arrangement function for the Transport.

Clip Editing in Arrangement View

Unlike most other audio applications that require users to change tools to perform clip-editing tasks, Ableton Live has a single tool that is used for clip editing. This unified tool provides instant access for selecting, moving, and resizing clips. This tool can also be used to create fade-ins, fade-outs, and crossfades.



Clip editing in the Session View is discussed in Chapter 6 (for MIDI clips) and Chapter 7 (for Audio clips).

Selecting a Clip

Ableton Live allows you to select an entire clip or just a portion of a clip without changing tools.

To select an entire clip:

- Click once in the upper half of the clip, where the hand icon appears.

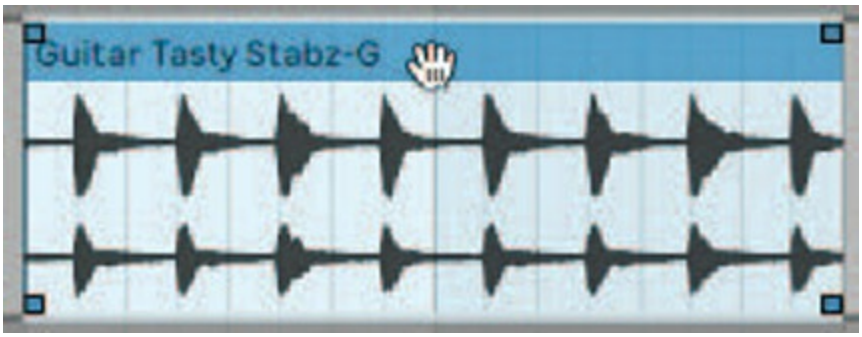


Figure 3.2 Selecting an entire clip by clicking in the upper half

To select time within a clip:

- Click and drag (forwards or backwards) in the lower half of the clip, where the pointer icon appears.

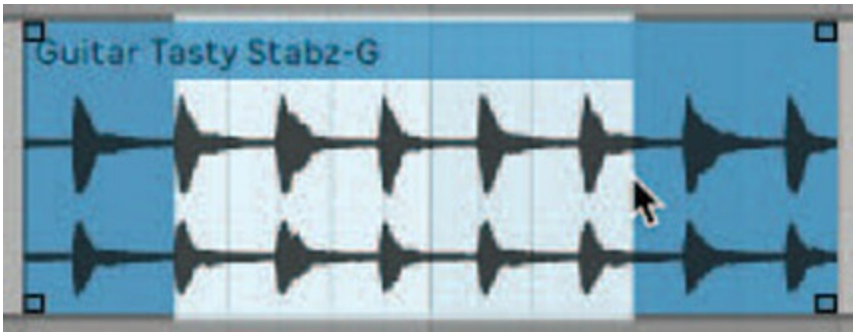


Figure 3.3 Selecting time within a clip by clicking and dragging in the lower half

Moving a Clip

As with selecting, Ableton Live allows you to move an entire clip or a selected portion of a clip without changing tools.

To move an entire clip:

- Click in the upper half of the clip and, without releasing the mouse button, drag the clip to the left or right.

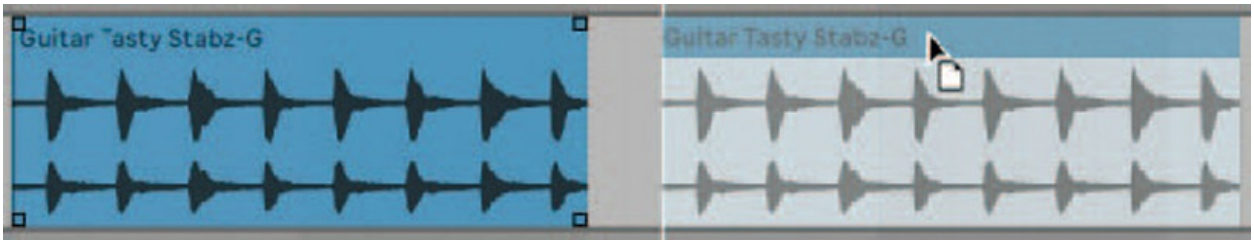


Figure 3.4 Moving an entire clip by clicking and dragging in the upper half

To move a time selection within a clip:

1. Make a time selection by clicking and dragging (forwards or backwards) in the lower half of the clip.

2. Release the mouse button.

3. Click in the upper half of the clip within the existing time selection and, without releasing the mouse button, drag the selected portion of the clip to the left or right. The selected audio will be removed from the original clip, creating a new, independent clip.

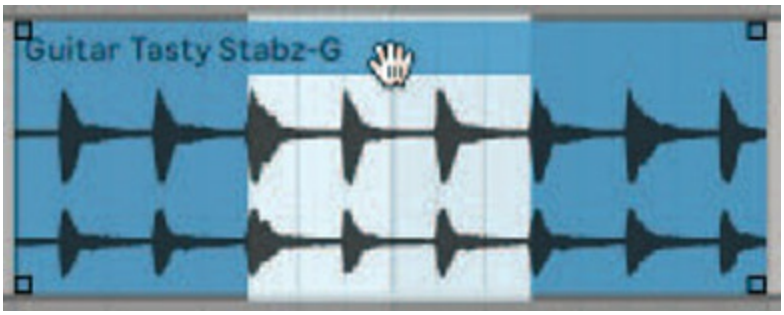


Figure 3.5 Clicking in the upper half of an existing time selection within the clip

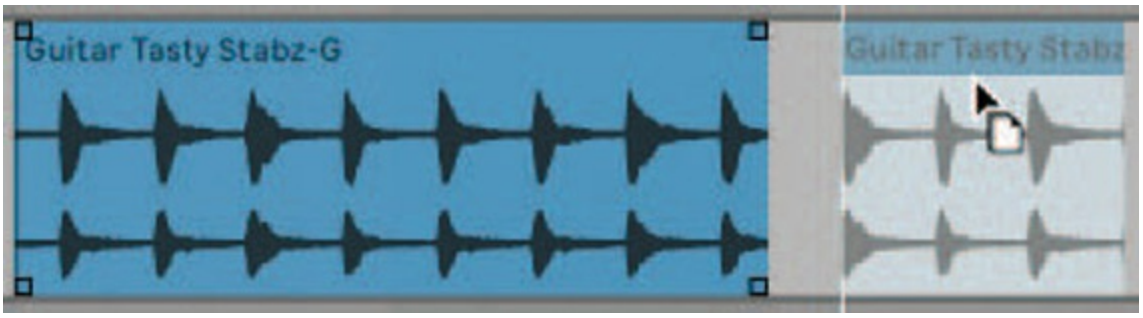


Figure 3.6 Moving the time selection



Figure 3.7 The result of moving the time selection after releasing the mouse

Dragging a clip vertically (up or down) will move the clip to another compatible track in your Set.

Clip movement in the Arrangement View may be constrained, depending on the current grid setting. An overview of grid settings is provided in the section titled "Grid Editing in Arrangement View" later in this chapter.

Resizing a Clip

Ableton Live offers a clip–resizing feature that can be used to adjust the clip boundaries to hide or expose underlying material. This feature can be used on audio, MIDI, and video clips. Like all clip editing in Ableton Live, resizing a clip is *nondestructive*, meaning it will leave the underlying source media file unchanged.

To resize a clip:

- Click and drag in the upper half of the clip near the clip boundary (beginning or end), where the bracket icon appears.

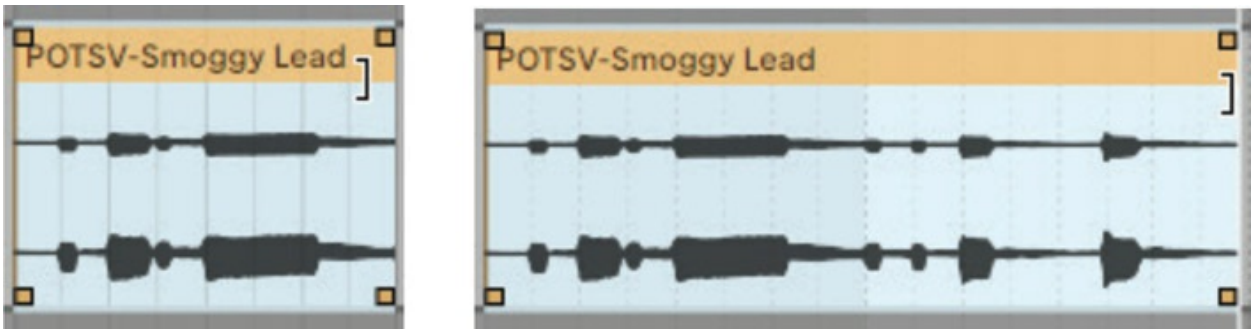


Figure 3.8 Extending the end of a clip to expose underlying material from the source file: before (left) and after (right)

Note that extending the length of a clip earlier or later can create two different outcomes:

- If the clip has underlying material available in the source media file, the

clip will extend normally.

- If there is not sufficient additional material available, and the Loop switch is active for the clip, Ableton Live will automatically loop the clip to create the desired extension.



The Loop switch for a clip is available in the Sample box in Clip View, as discussed in Chapters 4 and 9 of this book.

Creating Fades

A *fade* is a steady volume ramp that you create on a clip boundary. Fades have many different applications, from smoothing out an edit, to creating seamless clip overlaps, to building volume fade-ins and fade-outs for music and sound effects. This section covers the process of creating simple fade-ins, fades-outs, and crossfades.

It's very easy to add fades to audio clips in Ableton Live. When you hover the cursor over an audio clip, fade controls will appear as handles in all four corners of the clip. These handles can be used to create a fade in, fade out, or crossfade.



Fade handles may not appear at small track heights. If needed, you can adjust a track's height by clicking and dragging on its lower boundary.

As an alternative, you can also apply a fade based on an existing selection.

Applying Fades using Fade Handles

A quick way to apply fades is to use the fade handles that appear in the upper corners of audio clips.

To create a fade in or fade out using fade handles:

- Click and drag inward on the fade handle in the upper corner of the clip. Use the fade handle at the start of the clip to create a fade-in; use the

fade handle at the end of the clip to create a fade-out.

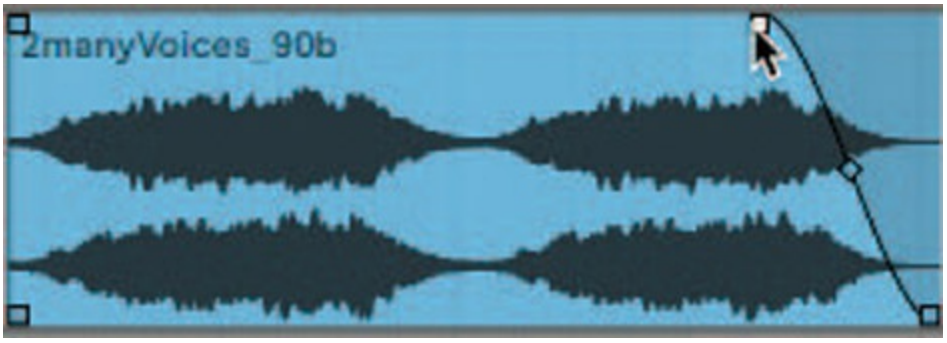


Figure 3.9 Adding a fade out to an audio clip

Applying Fades using Selections

Fade-in and fade-out effects can also be created at the beginning or ending of any audio clip using a selection that touches or crosses the clip boundary.

To create a fade from a selection, do the following:

1. Select across the beginning or ending of a clip.

2. Choose **CREATE > CREATE FADE IN/OUT** or press **OPTION+COMMAND+F** (Mac) or **CTRL+ALT+F** (Windows).

   The corresponding fade-in or fade-out will be created and will appear at the head or tail of the clip.

Following are some basic guidelines for creating fades using selections:

- Make your selection to match the desired fade length.

- Make a selection that touches or crosses an open clip boundary (not adjacent to another clip); selecting across adjacent clips will create a crossfade. (See “Applying Crossfades” below.)

- To create a fade-in, select from the start of the clip or before the start of the clip; to create a fade-out, select up to or across the end of the clip.

- Fade-ins always begin at the head boundary, and fade-outs always end at

the tail boundary. Extending a selection into the blank area beyond a clip's boundaries will not change the clip length when the fade is applied.

Applying Crossfades

Ableton Live allows you to create crossfades between any two adjacent audio clips that have sufficient underlying audio in their parent audio files. *Crossfading* is essentially the process of overlapping two audio sources and fading out the first source while simultaneously fading in the second source. Ableton Live achieves this effect using the underlying audio on either side of the boundary between adjacent trimmed clips.



If insufficient audio exists in the underlying clips, and the Loop switch is active for one or both clips, Ableton Live will loop the clips to extend their audio as needed to accommodate the crossfade.

To create a crossfade, do one of the following:

- Click a fade in or fade out handle and drag it across an adjacent clip's edge.
- Select a range of time that includes the boundary between two adjacent clips and choose **Create > Create Crossfade** or press **Option+Command+F** (Mac) or **Ctrl+Alt+F** (Windows).

Fade Settings

The curve shapes determine how the amplitude of the clip's audio will change over the course of the fade.

Once a fade has been created, you can adjust the slope of the fade.

To adjust the slope of an existing fade:

- Click and drag the slope handle that appears in the middle of the fade.

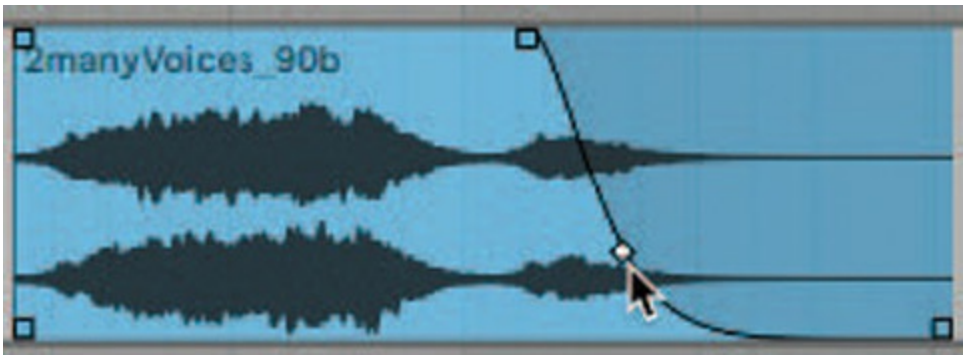


Figure 3.10 Adjusting the slope of a fade

Selecting an Area for Playback or Editing

You can use the mouse to position the insert marker or to select an area on a track for playback or editing.

To position the insert marker, click at the point where you want playback to begin. To select an area for playback or editing, click and drag across any area on one or more tracks.

You can select both horizontally and vertically, allowing you to create selections across multiple tracks in a single operation. Selected areas are represented with a white highlight in the Arrangement View. The selection range is also indicated by two small markers along the bottom of the Scrub Area, just above the Arrangement View.

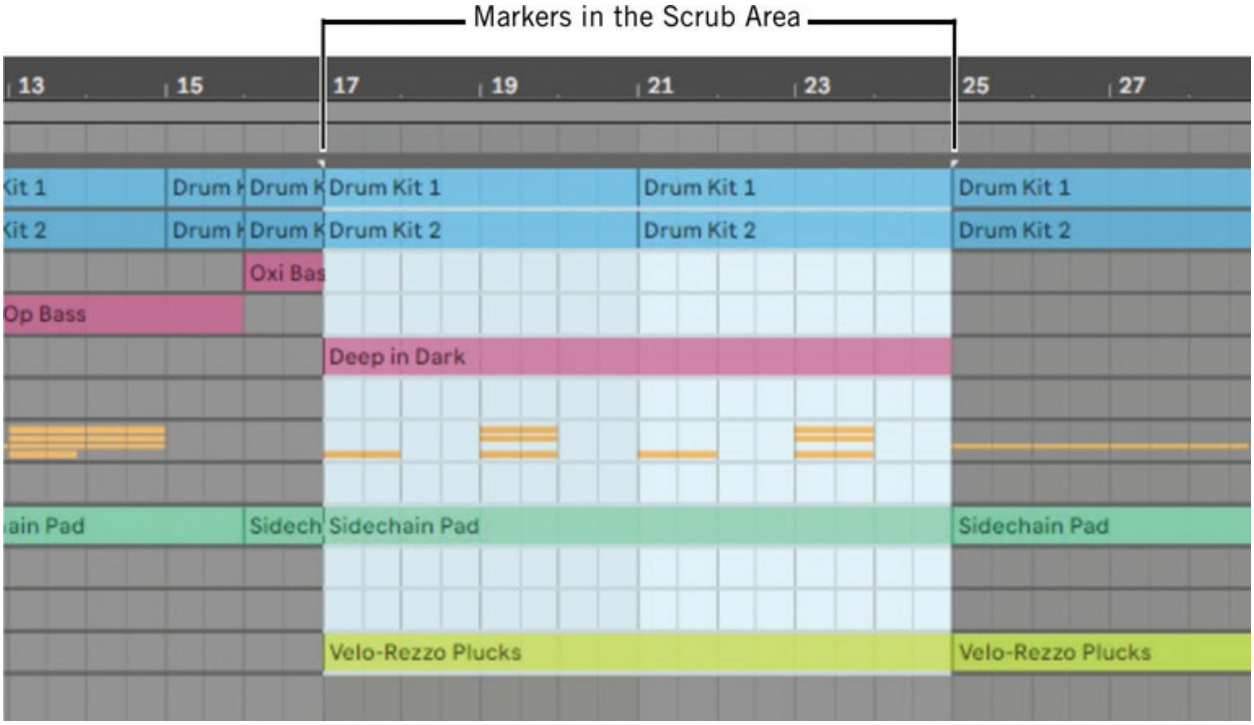


Figure 3.11 Selection across multiple tracks

You can use the mouse with the Shift key to quickly make a selection, as follows:

1. Click to position the insert marker where you want the selection to start.
2. Scroll to the desired endpoint.
3. Shift-click at the desired endpoint to complete the selection.

Zooming in Arrangement View

You can use Ableton Live's zoom features to zoom into and out of a particular area within the set. Zooming in is often helpful when you need to examine a clip or waveform closely.

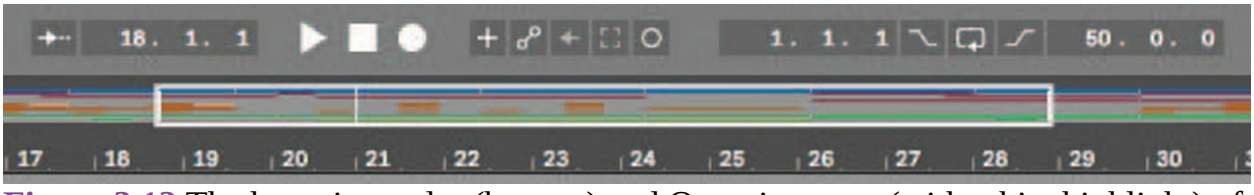


Figure 3.12 The beat-time ruler (bottom) and Overview area (with white highlight) of

the Arrangement View

The Arrangement View provides several different ways to zoom, including using the magnifying glass icon, using the Arrangement Overview, and using keyboard shortcuts.

Zooming with the Magnifying Glass Icon

The most common technique for zooming in the Arrangement View is to use the magnifying glass icon in the beat-time ruler or the Arrangement Overview.

To zoom using the magnifying glass icon:

1. Position the mouse cursor over the beat-time ruler or the Arrangement Overview, where the magnifying glass icon appears.

2. Click and drag vertically to change the zoom level: drag down to zoom in; drag up to zoom out.



Figure 3.13 Zooming with the magnifying glass icon in the beat-time ruler

Zooming with the Arrangement Overview

The Arrangement Overview is like a “bird’s eye” view of your entire set. It always shows the complete song in a miniature form. The current zoom level is indicated by a black or white box in the Arrangement Overview (depending on your chosen color theme).

To change the zoom level using the Arrangement Overview, do the following:

1. Position the mouse cursor along the left or right border of the Arrangement Overview's rectangular outline, where the bracket icon appears.

2. Click and drag the border of the outline to extend or shrink the currently visible area.



Figure 3.14 Zooming with the Arrangement Overview by dragging the start of the outline with the bracket icon

Zooming with Keyboard Shortcuts

More experienced users generally prefer to use keyboard shortcuts to zoom in the Arrangement View. Each key press zooms all tracks in by one level, with the Arrangement View centered on the insert marker.

To zoom using keyboard shortcuts:

- Press the + (plus) or − (minus) keys on your computer's keyboard.

Press Z to zoom the current Arrangement Time Selection to fill the screen. Press X to return to the previous zoom; press X a second time to zoom all the way out and see the complete Arrangement.

Grid Editing in Arrangement View

Ableton Live's Arrangement View features a flexible Edit Grid to assist in a number of common editing tasks. With the Edit Grid enabled, the cursor will snap to grid lines that represent a subdivision of the current song tempo. Selections, clip movements, and clip resizing operations are all constrained to the grid.

Moving and Placing Clips

Clips and MIDI notes that are moved with the Edit Grid enabled will snap to the nearest time increment using the current Edit Grid Value. The current Edit Grid Value is displayed in the lower right corner of the Arrangement View.

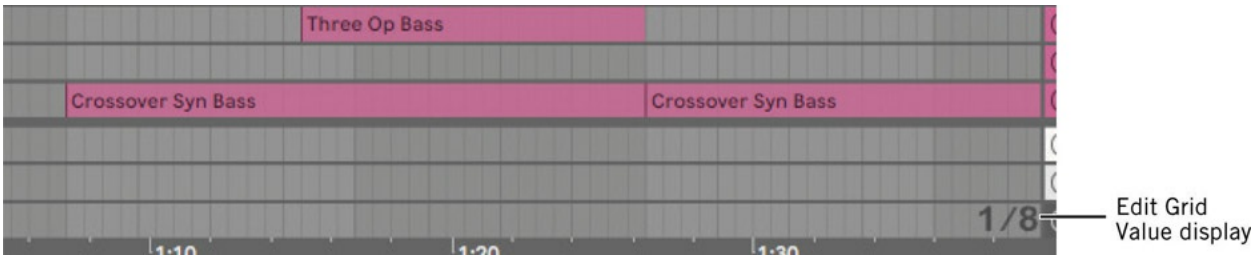


Figure 3.15 The current edit grid value (1/8 note) is visible in the bottom right corner of the Arrangement View

Regardless of the Edit Grid Value, clips can be placed anywhere on a track, so long as they start on a grid line. Clips can be positioned to overlap or cover other clips. Any underlying clips will be resized or obscured by overlapping clips.

Editing Clips

Editing Clips in the Arrangement View does not affect the timing of other clips on the same track. Deleting a clip or partial clip will leave empty space at the edit location. Pasted or duplicated material will overlap any other clips within the edit range.

Resizing a clip will trim the start or end of the clip to the nearest grid line or increment.

You can use the Edit Grid for making precise edits and aligning clips and selections using precise time intervals based on the Edit Grid Value. This is especially useful when using the Edit Grid for editing musical material that is aligned to the Set tempo.

Edit Grid Modes

Ableton Live's editing grid can be set to one of two modes: fixed grid and zoom-adaptive grid.

Fixed Grid

In fixed grid mode, selections, clip movements, and resizing operations are constrained to a specific grid value that does not change regardless of the zoom level.

Zoom-Adaptive Grid

In zoom-adaptive grid mode, selections, clip movements, and resizing operations are constrained to a grid value that changes based on the zoom level. Zooming in causes the Edit Grid to switch to progressively smaller values.

Setting the Edit Grid Mode and Value

To switch between edit grid modes and values, do one of the following:

- With no clip selected, **RIGHT-CLICK** in the Arrangement View and select the desired edit grid mode and value from the context menu:
  - Select an Adaptive Grid option, from Widest to Narrowest, to enable zoom-adaptive grid mode.
  - The Widest setting uses grid values starting a 16 bars when fully zoomed out; the Narrowest setting uses grid values starting at 1/2 note when fully zoomed out.
  - Select a Fixed Grid option, from 8 Bars to 1/32 note, to enable fixed grid mode.

- Select Off under Fixed Grid to disable the Edit Grid.

Disabling the Edit Grid allows you to position clips freely anywhere within the Arrangement view, without being constrained to grid lines.



Figure 3.16 The Arrangement View context menu showing Edit Grid options

- Press **COMMAND+5** (Mac) or **CTRL+5** (Windows) to toggle between fixed and zoom-adaptive grid modes.

Edit Grid settings can also be modified under the Options menu.

With the Edit Grid active, you can hold COMMAND (Mac) or ALT (Windows) when performing editing tasks to temporarily suspend snapping to grid lines.

Time Scales and Rulers in Arrangement View

Both the Session View and Arrangement View always display the current playback position and loop/punch start and length values in dedicated fields in the Control Bar. These values are always displayed in a bars.beats.sixteenths format.

Arrangement Position

The current playback position is displayed in the Arrangement Position field. Arrangement Position locations are represented in bars, beats, and sub-beats relative to the start point of the Set.



Figure 3.17 The Arrangement Position display in the Control Bar (highlighted in red)

Loop and Punch-In Start and Length

If you're planning to loop a section of a song or to punch in for a record pass, you'll need to enter the starting position in the **Loop Start/Punch-In Point** field. You then have the option to set the loop length or the length of the punched record take using the **Loop/Punch-In Region Length** field.



Figure 3.18 The Loop/Punch Start display (left) and Length display (right) in the Control Bar

Rulers

Every Ableton Live Set features two rulers: the beat-time ruler and the time ruler.

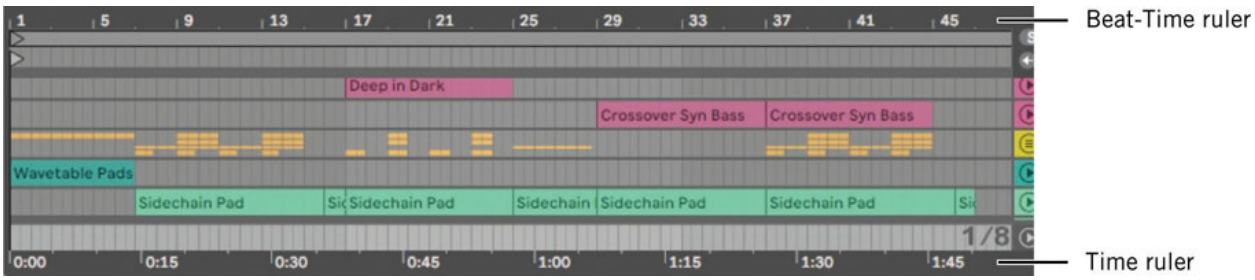


Figure 3.19 The beat-time ruler (top) and the time ruler (bottom) in the Arrangement View

Beat-Time Ruler

The beat-time ruler appears at the top of the Arrangement View and displays in a bars.beats.sixteenths format (depending on the current zoom level). This ruler is used for all transport functions including Arrangement Position, Loop and Punch-In Start and Length, and Edit Grid values.

Time Ruler

The time ruler provides additional timing reference and appears at the bottom of the Arrangement View. The time ruler can display one of several time formats:

To choose the time ruler format:

1. Right-click on the time ruler.

2. Choose the desired format from the context menu. (See Figure 3.20.)

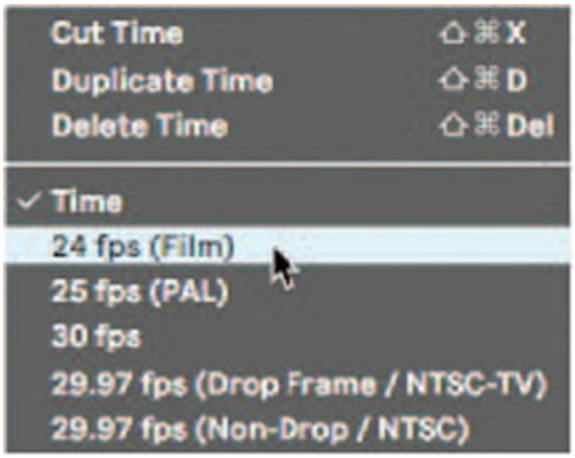


Figure 3.20 Setting the format from the time ruler's context menu

Time Ruler Display Options

Unlike the beat-time ruler, which always displays bars and beats, the time ruler format can be changed and set as needed. This can be useful for measuring the Set length in minutes and seconds, and also for referencing frame locations when producing music for film or TV. Ableton Live provides

options for common film and video frame rates in frames per second (fps).

The available time ruler formats include the following:

- Time (minutes:seconds:milliseconds)
- 24 fps (Film)
- 25 fps (PAL)
- 30 fps
- 29.97 fps (Drop Frame/NTSC TV)
- 29.97 (Non-Drop/NTSC)



A discussion of frame rates is beyond the scope of this book. However, the most important consideration here is to select the frame rate that matches the video you are using in your Set.

The Scrub Area in Arrangement View

The Scrub Area appears just below the beat-time ruler in the Arrangement View. Clicking in the Scrub Area will make playback jump to that specific location in the Set. The playback point will be quantized based on the global launch quantization value. To continuously scrub through the Set, set the global launch quantization value to **None** in the Quantization Menu (see "Quantization Menu" later in this chapter).

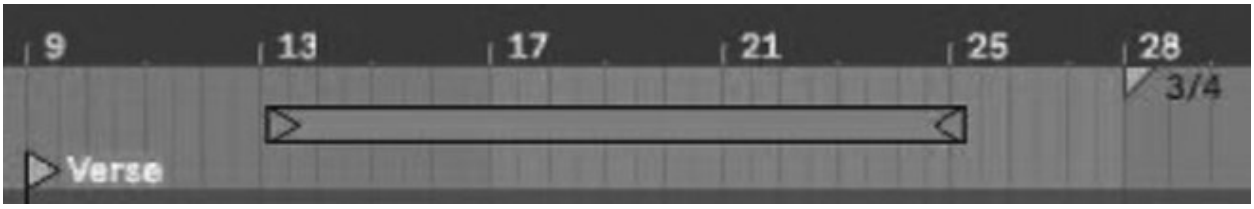


Figure 3.21 The Scrub Area with locator (left), loop brace (center), and time signature change (right).

Although the Scrub Area appears by default, it can be disabled

using the Permanent Scrub Area option in the LOOK/FEEL tab of the Ableton Live Preferences window.

You can also jump to a playback position by holding the SHIFT key and clicking on the beat-time ruler.

Adjusting the Loop Start/Punch-In Point and Length

The Scrub Area displays a loop brace that can be used to view and modify the Loop Start/Punch-In Point and Length values.

To move the loop to a new location:

- Click and drag the center of the loop brace, where the double-headed arrow icon appears.

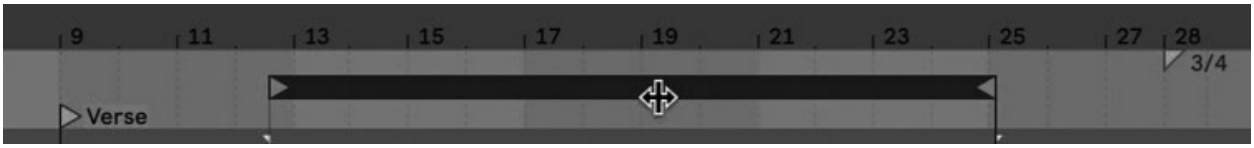


Figure 3.22 Moving the loop brace

To adjust the loop/punch start/end and length:

- Click and drag on the left or right edge of the loop brace, where the bracket icon appears.

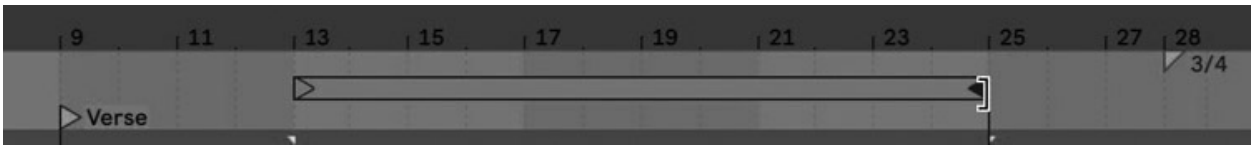


Figure 3.23 Adjusting the length of the loop brace

Changing the Time Signature

The Scrub Area can also be used to add and edit Time Signature changes.

To create a Time Signature Change:

1. Right-click in the Scrub Area and select **INSERT TIME SIGNATURE**

CHANGE from the context menu.

2. Type in a value for the time signature, such as 3/4 or 7/8.

To delete a Time Signature Change, do one of the following:

- Right-click anywhere in the Scrub Area and select **DELETE ALL TIME SIGNATURE CHANGES** from the context menu.
- Right-click directly on a Time Signature Marker and select **DELETE** from the context menu.

To reposition a Time Signature Change:

- Click directly on a Time Signature Marker and drag it to a new position.

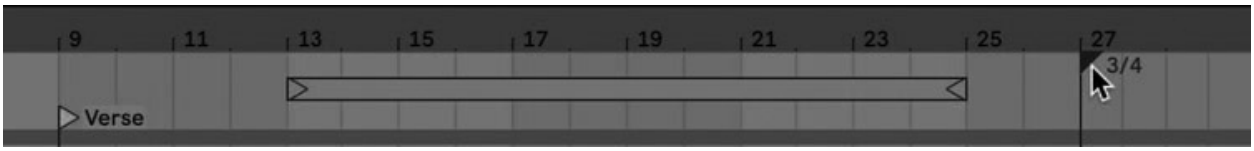


Figure 3.24 Moving a Time Signature Marker

Using Locators

Locators provide a great way to navigate to specific locations in your set. Locators are created in the Scrub Area of the Arrangement View.

To create a locator:

1. Do one of the following:

   - Right-click in the Scrub Area and select **ADD LOCATOR** from the context menu.
   - Position the insert marker on a track at the desired location and choose **CREATE > ADD LOCATOR**.

   A new locator icon will appear in the Scrub Area, with its name field active.

2. Type a name of the locator and press **RETURN** (Mac) or **ENTER** (Windows).

To begin playback using a locator:

- Double-click on the locator with the transport stopped.

To rename a locator:

1. Right-click on the locator and select **RENAME** from the context menu.
2. Type a new name and press **RETURN** (Mac) or **ENTER** (Windows).

To delete a locator, do one of the following:

- Click on the locator and press the **DELETE** key.
- Right-click on the locator and select **DELETE** from the context menu.

To reposition a locator:

- Click on the locator and drag it to a new position.



For information on triggering locators using the Next and Previous functions, see Chapter 8 in this book.

Performance Control Features

The Ableton Live toolbar provides access to various additional controls and options that affect playback and recording operations for the performance in your Set. Some of the available functions include the Tap Tempo button, the Tempo field, the Meter display, the Metronome button, the Metronome Settings selector, and the Quantization Menu.

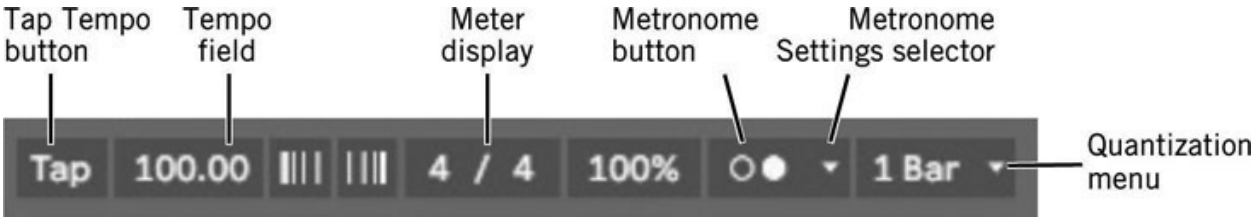


Figure 3.25 The performance controls on the left side of the toolbar

Tap Tempo Button

To set the tempo by tapping along with a performance, you can use the Tap Tempo button (labeled **TAP** in the Playback Controls). Click the mouse using successive clicks at the desired tempo. Ableton Live will display the tempo (BPM value) in the adjacent field (see below) based on your tap speed.

Tempo Field

The Tempo field displays the Set's current tempo based on the play location. You can set this value by using Tap Tempo or by entering a BPM value directly.

The tempo of the Set affects playback speed of audio and MIDI performances as well as the spacing of bars and beats on the beat-time ruler and edit grid.

See the section on "Setting the Tempo and Time Signature" in Chapter 6 for more information on adding tempo changes to a Set.

Meter Display

The Meter display field indicates the Set's current meter based on the play location. You can enter a meter change value directly into this field to change the Time Signature for the current section (starting at any existing Time Signature Change within the section and ending at the next Time Signature Change). This will not affect Time Signature Changes that you've added before or after the current section.

The meter determines how many beats are included in each measure on the beat-time ruler. It also affects the playback of the click sound—see

“Metronome Button” below. (The click places emphasis on the first beat of every measure.)

Metronome Button

The Metronome button controls whether or not the metronome click will be audible. When the Metronome button is enabled (highlighted in orange), a metronome click will sound during playback and recording, as specified by the metronome settings.

Metronome Settings Selector

Next to the Metronome button is a down-pointing arrow button called the Metronome Settings selector. You can use this selector to set the Count-In duration and other attributes that affect the metronome click.

To modify the metronome settings:

- Click on the Metronome Settings selector and choose the desired settings from the Metronome menu.

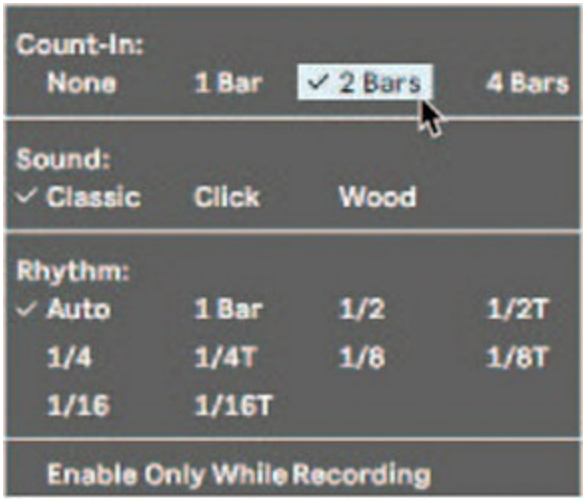


Figure 3.26 The Metronome menu

The Count-In setting determines if a click will play before playback or recording begins. You can also specify the number of bars that will be used for the count-in.

Quantization Menu

The Quantization menu to the right of the metronome controls on the toolbar is used to choose the global launch quantization setting for your Set. This sets the interval used to begin playback at a new position, such as when clicking in the Scrub Area or triggering a locator during playback.

To set the global launch quantization, do the following:

1. Click on the Quantization menu.

2. Select a quantization value from the pop-up menu, or select **NONE** from the top of the menu to disable launch quantization.

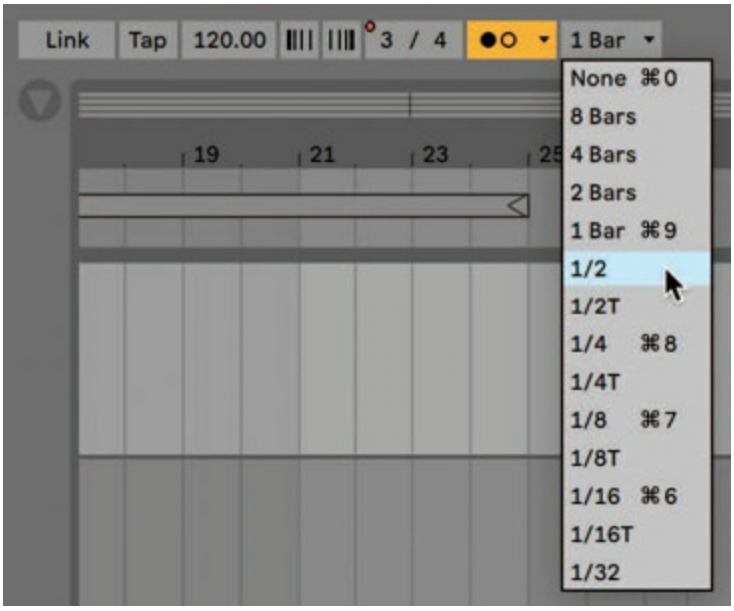


Figure 3.27 Options available in the Quantization menu

Modifying Devices and Clips

As mentioned in Chapter 2, the Detail View area is an optional display that can be shown at the bottom of the Session or Arrangement View. This area features two separate display options: the Device View and the Clip View.

Accessing Devices in Device View

When Device View is active in the Detail View area, clicking on a track's

title bar will display the track's device chain at the bottom of the Session View or Arrangement View. Devices displayed here can include Live's built-in audio effects, MIDI effects, and instruments, as well as third-party plug-in devices in VST and AU format.

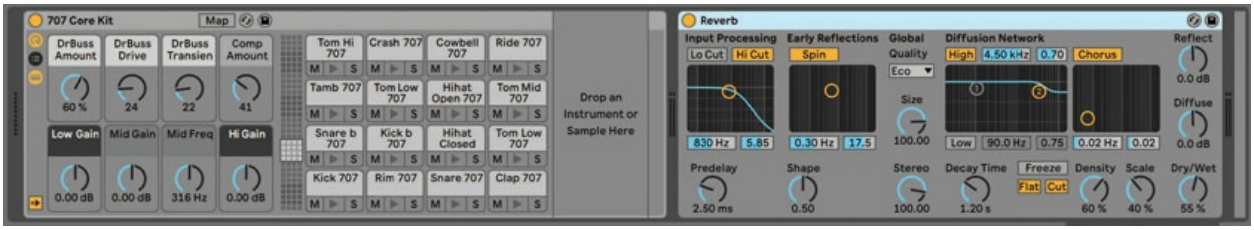


Figure 3.28 Device View shown in the Detail View area

If Device View is not active, double-click on a track's title bar to display the Device View with the track's device chain shown.

Ableton Live's devices and third-party plug-ins are available from the Browser and can be added to tracks by dragging them from the Browser into the Device View. The Device View allows you to add, view, and delete devices and plug-ins for the selected track. This is also where you can see and adjust parameters for the devices and plug-ins used on a track.

Additional details on devices and plug-ins are provided in Chapter 10 of this book.

Accessing Clips in Clip View

When Clip View is active in the Detail View area, selecting a clip on a track in Arrangement View or in a Clip Slot in Session View will display the clip's waveform or MIDI note arrangement in the Clip View area. The Clip View also displays the properties of the selected clip and allows you to modify clip properties.



Figure 3.29 Clip View shown in the Detail View area (with an audio clip selected)

If Clip View is not active, double-click on a clip to display the Clip View with the clip's waveform or MIDI performance shown.

MIDI clips and audio clips in Live have somewhat different Clip View controls. However, the Clip View has the following items in common for both audio and MIDI clips:

- The Clip box, which is used to adjust basic clip settings.
- The Launch box, which is used to control clip launch behavior in the Session View (available for Session View clips only).
- The Envelopes box, which is used to modulate clip, mixer, and device parameters within each clip.



Figure 3.30 The Clip box (left), Launch box (middle), and Envelopes box (right) in the Clip View

Audio clips have some unique Clip View controls, including the following:

- The Sample box, which is used to modify the sample playback, looping, and warping behavior.
- The Sample Editor, which shows an overview of the sample and allows you to adjust Transient and Warp Marker positions.

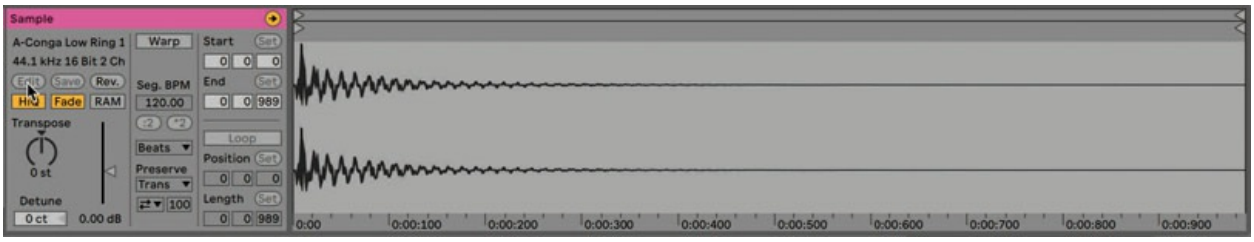


Figure 3.31 The Sample box (left) and Sample Editor (right) in the Clip View

MIDI clips also have some unique Clip View controls, including the following:

- The Notes box, which is used to modify the note arrangement and playback behavior for the clip.
- The MIDI Note Editor, which is used to edit individual MIDI notes within the clip.

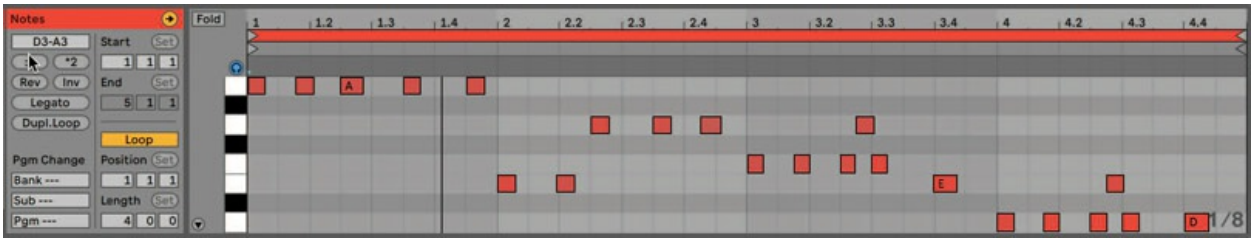


Figure 3.32 The Clip box (left) and the MIDI Note Editor (right) in the Clip View

Edit Mode Versus Draw Mode

The Ableton Live Clip View operates in Edit Mode by default; however, you can switch to Draw Mode at any time. Edit Mode is used to adjust Transient and Warp Marker positions for audio clips and to edit notes by clicking and dragging for MIDI clips. In this mode, MIDI notes can also be added or deleted by double-clicking in the MIDI Note Editor.

Draw Mode offers alternate functionality for editing MIDI notes and automation data. With Draw Mode active, you can “draw” notes in the MIDI Note Editor by clicking with the Pencil tool; you can also click and drag to draw multiple notes in succession. When working with automation, Draw Mode allows you to click and drag to draw envelope curves.

To toggle Draw Mode on/off, do one of the following:

- Click the **DRAW MODE** button in the Control Bar.
- Press the **B** key.

The Draw Mode button will be highlighted in yellow when enabled.



Figure 3.33 The Draw Mode button (yellow) in the Control Bar

Using Draw Mode to create MIDI notes or automation will create items that adhere to the current Edit Grid value.

Using the Info View

As discussed briefly in Chapter 2, the Info View is an additional display that is available within the Detail View area. It is visible in both the Session View and Arrangement View. When displayed, the Info View appears at the far left side of the Detail View. When you position the mouse pointer above an element of the Ableton Live user interface, the Detail View will display a short description of that element.

The Info View is extremely helpful for learning various aspects of the Ableton Live interface. It not only helps you to understand what a particular knob or button can do, but also helps you to learn the correct terminology for the software.

To show or hide the Info View, do one of the following:

- Click the **SHOW/HIDE INFO VIEW** button (triangle actuator) in the bottom left corner of the Session View or Arrangement View.
- Select **VIEW > INFO** from the main menus.
- Press **SHIFT+?** (question mark) on your computer keyboard.

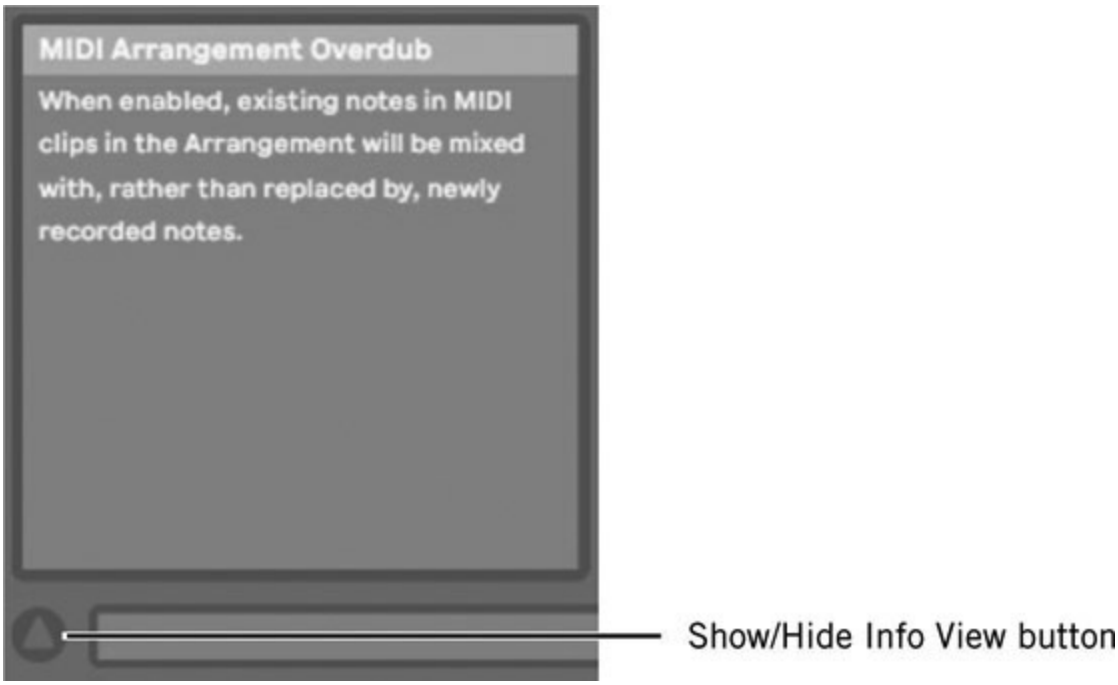


Figure 3.34 The Info View with the Show/Hide button visible at the bottom left

Review/Discussion Questions

1. Which part of a clip must be clicked to select the entire clip? Which part of a clip must be clicked to select a portion of a clip? (See “Selecting a Clip” beginning on page 59.)

2. How can you move a portion of a clip, separating it from the underlying clip? (See “Moving a Clip” beginning on page 59.)

3. Where must you click in a clip to resize the clip beginning or end? (See “Resizing a Clip” beginning on page 61.)

4. What are the two ways that fades can be applied at the beginning or end of a clip? (See “Creating Fades” beginning on page 62.)

5. How can you adjust the slope of a fade or crossfade? (See “Fade Settings” beginning on page 63.)

6. Which key on the computer keyboard can be used to quickly make a lengthy selection for playback or editing? (See “Selecting an Area for Playback or Editing” beginning on page 64.)

7. What are some ways that you can zoom in the Arrangement View? (See “Zooming in Arrangement View” beginning on page 65.)

8. Will editing clips on a track in the Arrangement View affect the other clips on the track? (See “Editing Clips” beginning on page 67.)

9. What are the two Edit Grid Modes in Ableton Live? How can you set the Edit Grid Mode and Value? (See “Edit Grid Modes” beginning on page 67.)

10. What two rulers are available in the Arrangement View? Is it possible to change the time format for both rulers? (See “Rulers” beginning on page 69.)

1. What happens when you click in the Scrub Area at the top of the Arrangement View? (See “The Scrub Area in Arrangement” beginning on page 71.)

2. What are some ways to set the Tempo of your Set? (See “Performance Control Features” beginning on page 73.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 3

Identifying Primary Tools and Controls

Activity

In this exercise worksheet, you will identify various Ableton Live controls. The information referenced in these questions is covered in Chapter 3.

Duration

This exercise should take approximately 10 minutes to complete.

Goals/Targets

- Identify components of the Control Bar
- Id entify proper names for performance control options

Instructions

Answer the questions in this exercise by filling in the blanks on the following pages or by listing your answers on a separate sheet of paper or worksheet. Be sure to use the terminology referenced in Chapter 3 as you complete this exercise.

Control Bar Fields

See the section on “Time Scales and Rulers in Arrangement View” in Chapter 3 for assistance with the questions below.

Question 1 refers to Figure 3.35.

1. The item labeled **A** in Figure 3.35 is called the _____ field.



Figure 3.35 Fields available in the Control Bar

Questions 2 and 3 refer to Figure 3.36.

2. The item labeled **A** in Figure 3.36 is called the _____ field.

3. The item labeled **B** in Figure 3.36 is called the _____ field.



Figure 3.36 The Looping Controls in the Control Bar

Performance Controls

Refer to the section on "Performance Control Features" in Chapter 3 for assistance with the questions below.

Questions 4 through 7 refer to Figure 3.37.

4. The item labeled **A** in Figure 3.37 is called the _____ button.

5. The item labeled **B** in Figure 3.37 is called the _____ field.

6. The item labeled **C** in Figure 3.37 is called the _____ field.

7. The item labeled **D** in Figure 3.37 is called the _____ button.

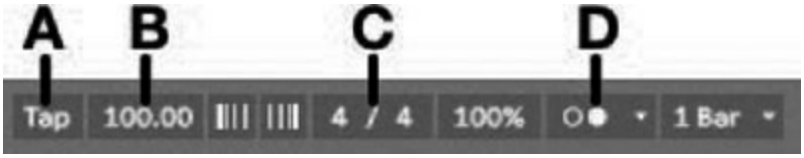


Figure 3.37 The performance controls in the Control Bar

Chapter 4

Creating Your First Live Set

This chapter covers the basics of working with Ableton Live. It introduces configuration options, playback and navigation controls, and project saving and opening operations.

Learning Targets for This Chapter

- Choose appropriate preferences for Live
- Create a new Live Set
- Create and name Audio and MIDI tracks
- Navigate your Live Set for playback and editing
- Save, locate, and open Live Sets
- Recognize the difference between a Live Set and a Live project

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform. Sign up for free at ElementsED.com.

Before you can begin working with audio or MIDI in Ableton Live, you need to have a Live Set open. This chapter covers the basics of creating Live Sets and Live projects, adding tracks to your Live Set, using basic navigation for your tracks, and saving and re-opening your work.

Creating and Configuring a Live Set

The first time you open Ableton Live, you are greeted with a pre-created Live Set that demonstrates some of the software's capabilities and features. While you are welcome to explore and edit the demo project, this chapter will guide you through creating a new blank Live Set, which will serve as a fresh starting point for expressing your creativity.

To create a new Live Set, do one of the following:

- Launch Ableton Live, if it's not already running. Live will open a new untitled Set whenever the software launches.
- Choose **File > New Live Set**, if Ableton Live is already running with another Set open.

Configuring Live's Audio Preferences

Before you start creating in your new Live Set, it is recommended that you configure several important Ableton Live preference settings. These settings include selecting the audio device you will use with Live and choosing the sample rate, file format, and bit depth for your audio. Audio preferences apply to Ableton Live as a whole: once you have configured your audio preferences, Live will use them for any and all projects you create or open in the future. If your audio needs or hardware devices change, you can modify your preference settings at that point.

To view Ableton Live preferences, do one of the following:

- Choose **Live > Preferences** (Mac) or **Options > Preferences** (Windows).

■ Press **COMMAND+**, (comma) on Mac or **CTRL+**, (comma) on Windows.

The Live Preferences window will display. (See Figure 4.1.)



Figure 4.1 The Live Preferences window, with the Audio tab selected

Audio Device and Sample Rate Settings

The **AUDIO** tab in the Preferences window allows you to select an audio device, whether it's a dedicated hardware device or your computer's built-in audio driver, to use in conjunction with Live. You can also use this tab to set the sample rate to use for your Live Sets.

Selecting an Audio Device

To get started recording and playing back through an audio interface, you'll need to select a **Driver Type**, which allows Ableton Live to recognize and communicate with the interface. The options available to you will depend on the audio hardware drivers installed on your system. On Mac-based systems, you can select **Core Audio** to use the built-in audio capabilities of your computer. Many audio interface options from third-party manufacturers are also Core Audio-compliant. Other interfaces may require that you download and install a specific driver.

Consult your audio device manual or the manufacturer's website for information on the required drivers.

If you have a compatible audio interface connected to your computer, you can select it from the **Audio Input Device** and **Audio Output Device** dropdown menus. Once you've selected your device, you can use the **Input Config** and **Output Config** buttons to choose the hardware inputs and outputs that will be available for use in Ableton Live tracks. Click the **Hardware Setup** button, if available, to launch the native configuration software for your connected audio device.

Use the Ableton Live Help View

If you're not familiar with audio devices and their configuration, you can access onscreen help from within the application to assist you during setup. Ableton Live features a "Setting up Audio I/O" help lesson, which walks you through the set up and optimization process.

To access the help lesson, click **Help > Help View** from the menu bar, and then click **Audio I/O**.

Setting the Sample Rate

After selecting an audio device, you can choose from its available sample rates. Live supports sample rates up to 192 kHz, when used with a compatible audio interface. To optimize the file sizes in your Set, choose the lowest

sample rate that meets the needs of your project.

A sample rate of 44.1 kHz is often adequate for home– and project–studio recordings. Higher sample rates can be chosen for demanding projects, to capture a greater frequency response from the source audio, and to minimize sound degradation throughout the project lifecycle. However, with higher sample rates come greater disk space requirements for your Set. (See Table 4.1 in the "File Size Considerations" section later in this chapter.)



You can change the sample rate in Ableton Live at any time. Audio clips with different sample rates can be used together in a Live Set because Live performs real-time sample rate conversion during playback.



For more details on sample rates, see "The Importance of Sample Rate" in Chapter 1 of this book.

File Type and Bit Depth Settings

Within the **PREFERENCES** window, the **RECORD/WARP/LAUNCH** tab allows you to select the audio file format and the bit depth to use with Ableton Live. While you can mix and match audio files of varying formats and bit depths in your Live Set, these preferences determine the format and bit depth of audio that you record or generate (through Freezing or Resampling, for example) within your Live Set.

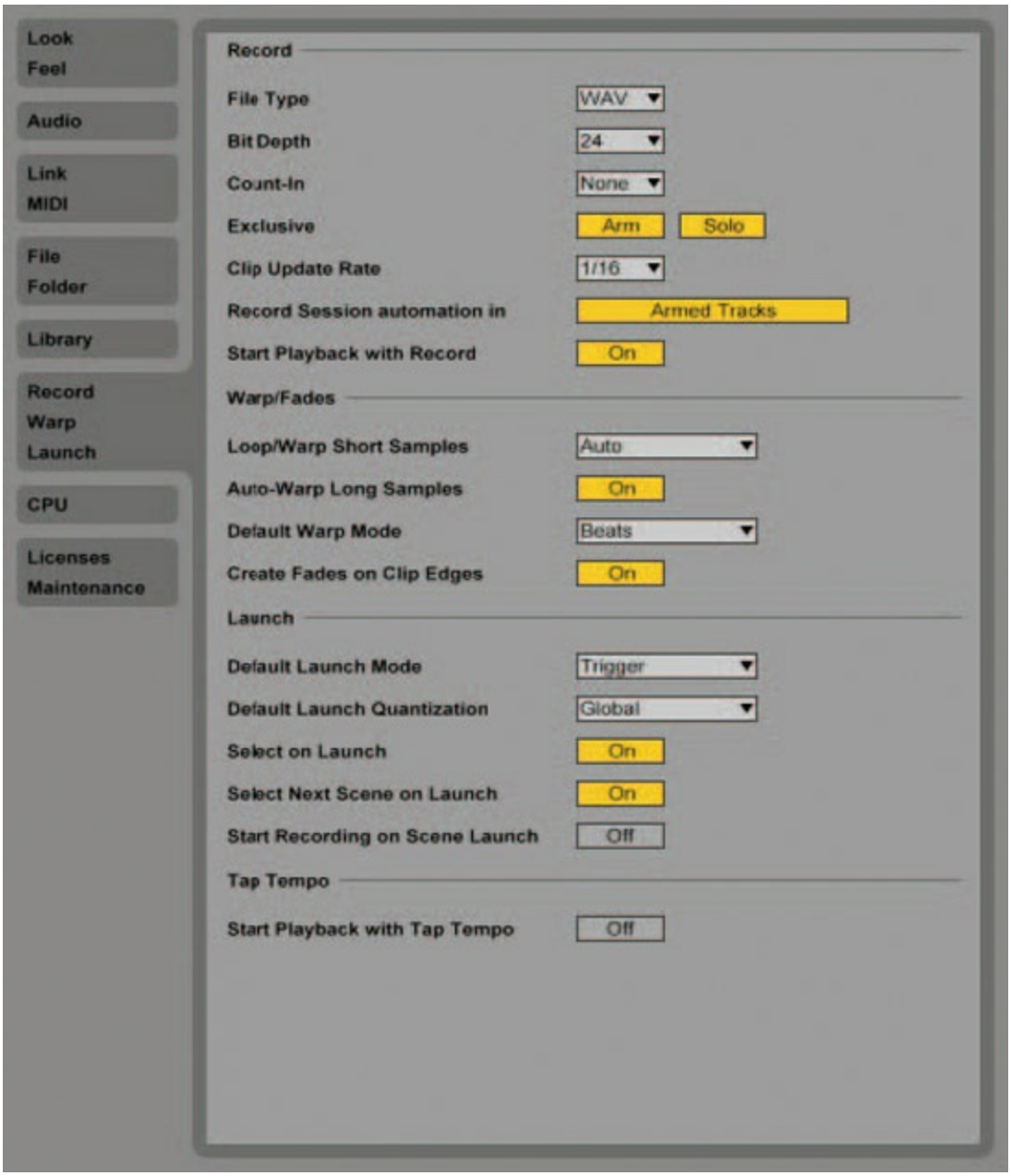


Figure 4.2 The Live Preferences window, with the Record/Warp/Launch tab selected

File Type

Live stores audio as WAV or AIFF files. WAV is the default file type on Windows computers, while AIFF is the default on Macs. Use the default format unless you intend to use your files for another purpose that requires the alternative format.

Bit Depth

Live works with files in 16-bit, 24-bit, or 32-bit floating-point audio resolution. The 16-bit option generates smaller files and is typically adequate for basic recording projects. The 24-bit and 32-bit options provide greater dynamic range in your recorded audio and also help lower the noise floor.



For more details on bit depth, see "The Importance of Bit Depth" in Chapter 1 of this book.

Higher bit-depth options should be used for high-end recordings, especially recordings that include very quiet passages (such as a classical orchestra), recordings that require intensive processing, and recordings intended for media that support high resolution audio, such as DVD and Blu-ray disc.



For the highest quality audio, record at 24-bit or 32-bit floating point and properly dither down, if needed, during the final mix. Dithering is covered in Lesson 10.

File Size Considerations

A tradeoff of choosing higher sample rates and bit depths is an increase in the amount of disk space required to store your audio files and the amount of processing power required to manipulate them. Table 4.1 shows the relationship between sample rate, bit depth, and disk space consumption for the standard configurations supported in Ableton Live.

Table 4.1 Audio Recording Storage Requirements (Approximate)

Sample Rate	Bit Depth	Megabytes/Track Minute (Mono)	11 Megabytes/Track Minute (Stereo)
44.1 kHz	16-bit	5 MB	10 MB
44.1 kHz	24-bit	7.5 MB	15 MB
44.1 kHz	32-bit	10 MB	20 MB
48 kHz	16-bit	5.5 MB	11 MB

48 kHz	24-bit	8.2 MB	16.4 MB
48 kHz	32-bit	11 MB	22 MB
88.2 kHz	16-bit	10 MB	20 MB
88.2 kHz	24-bit	15 MB	30 MB
88.2 kHz	32-bit	20 MB	40 MB
96 kHz	16-bit	11 MB	22 MB
96 kHz	24-bit	16.5 MB	33 MB
96 kHz	32-bit	22 MB	44 MB
176.4 kHz	16-bit	20 MB	40 MB
176.4 kHz	24-bit	30 MB	60 MB
176.4 kHz	32-bit	40 MB	80 MB
192 kHz	16-bit	22 MB	44 MB
192 kHz	24-bit	33 MB	66 MB
192 kHz	32-bit	44 MB	88 MB

Working with Tracks

Once you've created a new Live Set, you will have a project with two MIDI tracks and two audio tracks, along with two return tracks and a Master track (provided you haven't changed the default Live Set template).

In Ableton Live, tracks are where audio, MIDI, and automation data are recorded and edited. Audio and MIDI data is stored in clips, which you can

edit, arrange, and trigger in any order you choose to create patterns and songs.

Supported Track Types

Like most audio applications, Ableton Live supports a variety of track types that can be used for different purposes.

The basic track types supported in Live include the following:

- Audio tracks
- MIDI tracks
- Return tracks
- Master tracks

Ableton Live also includes a feature known as a Group Track. While not a track type in the traditional sense, a Group Track provides a summing container for submixing audio or MIDI tracks, or for nesting other existing Group Tracks.

The Function of Group Tracks

Group Tracks are unique in that they do not contain any audio or MIDI clips. Instead, they function as a *container* for other audio and MIDI tracks. They have mix controls and can host audio effects, making them a useful way of creating submixes within your Live Set.

To create a Group Track, select the tracks you'd like to include in the group and choose **Edit > Group Tracks**. (You can also you can press **Command+G** [Mac] or **Ctrl+G** [Windows] to create a new Group Track.)

Once created, you can add and remove audio and MIDI tracks from a Group Track by dragging them in and out of the Group.

Any combination of track types can be added to your Live Set, with the exception of the Master track, which is limited to one and is included in all Live Sets by default.

Audio Tracks

Audio tracks allow you to import, record, and edit audio signals as waveforms within clips. Live supports mono (one-channel) or stereo (two-channel) audio.

There is no limit to the number of audio tracks you can insert into a Live Set if you're running the Standard or Suite edition of Ableton Live. (Ableton Live Intro is capped at 16 audio and/or MIDI tracks.) However, the Set sample rate, the system hardware, and the continuity of audio all impact how many tracks can actually play back and record simultaneously.



Live uses the computer's CPU to mix and process audio tracks. Computers with faster clock speeds and greater processor counts will support more tracks and audio effects than computers with slower CPUs or fewer processors.

MIDI Tracks

MIDI tracks store MIDI note and controller data within MIDI clips. Live's MIDI sequencer lets you import, record, and edit MIDI data in much the same way that you work with audio. However, MIDI on its own does not generate sound; it is data that is sent to software instruments, which output sound in response. MIDI and audio tracks can coexist in one Live Set.

As is the case with audio tracks, there is no limit to the number of MIDI tracks you can add to a Live Set (when running the Standard or Suite edition of Ableton Live). The capabilities of your system and your Ableton Live preference settings will determine how many MIDI tracks can play back or record simultaneously.

Return Tracks

Return tracks do not contain any audio or MIDI clips; instead, they host

audio effects. A return track can process audio sent to it from any number of other tracks within the Live Set.

This is particularly useful for applying the same audio effect to multiple tracks. For instance, you could set up a delay effect on a return track and send signals to it from multiple audio and MIDI tracks, thereby saving system resources and setup time.

Master Track

The Master track serves to control the overall level of the audio output for your Set. A Master track is included by default in all Live Sets. It can host audio effects, and its Master Volume slider will control the level of Ableton Live's final output.

Adding Tracks to Ableton Live

To add audio or MIDI tracks to your Set, choose **Create > Insert Audio Track** or **Create > Insert MIDI Track** from the menu bar. You can also add new return tracks under the **Create** menu.

Shortcuts for Creating Trac

Shortcuts exist to speed up the process of creating new MIDI, audio, and return tracks.

Use **Command+T** (Mac) or **Ctrl+T** (Windows) to create a new audio track.

Use **Command+Shif+T** (Mac) or **Ctrl+Shift+T** (Windows) to create a new MIDI track.

Use **Command+Option+T** (Mac) or **Ctrl+Alt+T** (Windows) to insert a new return track.

Regardless of the method you use to create a new track, new

MIDI and audio tracks will be inserted below or to the right of the currently selected track (in Arrangement View and Session View, respectively). To rearrange MIDI and audio tracks, click on the track name and drag the track to a new position.

Naming Tracks

When you create tracks in Ableton Live, they are added to the Live Set using generic names, such as 1 Audio, 2 Audio, 1 MIDI, and so on. Fortunately, you can easily change the generic track names to something more meaningful.

To change a track name, do one of the following:

- Right-click the track name and select **RENAME** from the pop-up menu.
- Click the track name and press **COMMAND+R** (Mac) or **CTRL+R** (Windows).

To retain automatic track numbering at the start of a track name, include the "#" symbol as the first character. To number tracks using multiple digits (such as 01, 02, 03, etc.), use multiple "#" characters when renaming.

Press TAB or SHIFT+TAB when renaming a track to cycle forward or backward through your tracks and rename each without leaving the keyboard.

Deleting Tracks

You can delete a track from your Live Set at any time. When you delete tracks, the audio or MIDI clips associated with the track will be lost. If you inadvertently delete a track or tracks, you can undo the operation by choosing **EDIT > UNDO DELETE TRACK(S)**.

To delete a single track:

1. Click the track name to select the track.

2. Choose **EDIT > DELETE** or press the **DELETE** key on the computer keyboard.

To delete a continuous range of tracks simultaneously:

1. Click the track name of the first track you'd like to delete.
2. S HIFT - CLICK on the track name of the last track you'd like to delete.
3. Choose **EDIT > DELETE** or press the **DELETE** key on the computer keyboard.

To delete a discontinuous collection of tracks simultaneously:

1. Click the track name of the first track you'd like to delete.
2. **COMMAND-CLICK** (Mac) or **CTRL-CLICK** (Windows) on the track name of each additional track you'd like to delete.
3. Choose **EDIT > DELETE** or press the **DELETE** key on the computer keyboard.



You can also right-click on a track name and choose DELETE from the pop-up menu to remove all selected tracks.

Adding Audio to Tracks

Once you have created one or more audio tracks in your Set, you can begin adding audio clips, either by recording to your track(s) or by importing existing audio files (or *samples*) from a hard drive. Audio clips can be added to timeline locations on tracks in the Arrangement View (where they can be played sequentially) or to Clip Slots in the Session View (where they can be launched in non-sequential order).



Importing audio files is covered in Chapter 5; audio recording is covered in Chapter 7.

Adding MIDI to Tracks

Once you have created one or more MIDI tracks in your Set, you can create MIDI clips and write MIDI data with the pencil tool by activating Draw Mode. You can also add MIDI to a track by recording a performance from a MIDI controller or by importing a MIDI file from a hard drive. Like audio, MIDI clips can be added to timeline locations on tracks in Arrangement View or to Clip Slots in Session View.



The various methods of importing, recording, editing, and working with MIDI data are covered in Chapters 5 and 6 of this book.

Controlling Playback

How you go about playing your Live Set is dependent on the current view mode you are using. You can choose between the Arrangement View and the Session View.

Playback in Arrangement View

When viewing a Live Set in the Arrangement View, the main Track Display window shows a horizontal timeline of the clips in your project. Playback starts from the location of the Arrangement Insert Marker (or playhead), which is a cursor indicated by a solid, blinking line that runs through one or more tracks, and an arrow that appears both in the lane above the Track Display (below the beat-time ruler) and in the area below the Track Display (in the time ruler). The playhead's location is reflected in the Arrangement Location counter at the top of the Live Set.

Click the **PLAY** button (or press the **SPACEBAR**) to begin playback based on the location of the playhead.

Selections

You can also click and drag within the Track Display window to select a period of time. The selection is indicated by a highlighted zone that can span one or more tracks, with markers on each end shown in the lane above the

Track Display and in the time ruler below the Track Display. By making a selection on a track, you define an area for Ableton Live to perform a desired editing task, such as deleting, duplicating, or muting, among others.

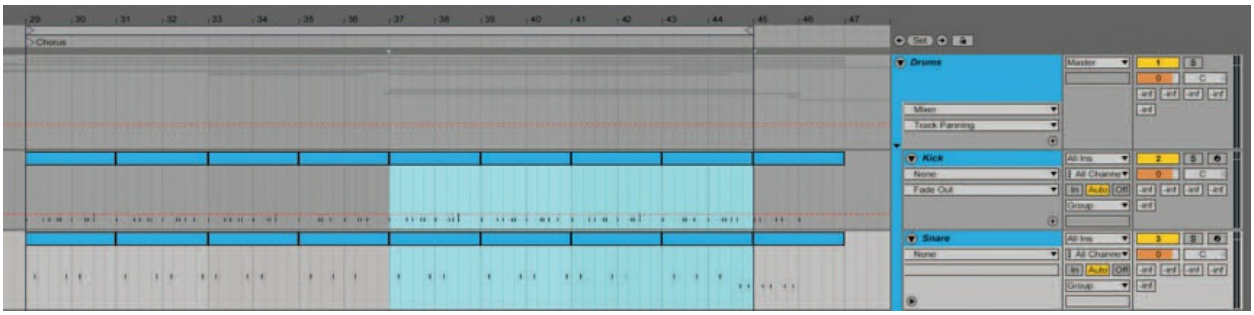


Figure 4.3 Selection from Bar 37 to Bar 45

Click the **PLAY** button (or press the **SPACEBAR**) to begin playback based on the start of the selection. Playback will continue until you click the **STOP** button (or press the **SPACEBAR** a second time).

Press OPTION+SPACEBAR (Mac) or ALT+SPACEBAR (Windows) to play only the selection, stopping at the selection end.

Be aware that you cannot make a selection by clicking and dragging an audio or MIDI clip, as that will move the clip. To make a selection, click and drag across empty space or, with a track expanded, click and drag in the bottom part of a clip to avoid selecting the clip itself.

Alternatively, you can click on an Audio or MIDI clip to create a selection. With the clip selected, playback will begin from the clip start.

The Arrangement Loop

As an alternative to the linear playback that is typical of the Arrangement View, you can activate the Arrangement loop, causing playback to loop repeatedly whenever playback starts within or crosses into the loop area. The loop span is demarcated in the Scrub Area by the Arrangement loop brace: a horizontal strip with triangle markers at either end.

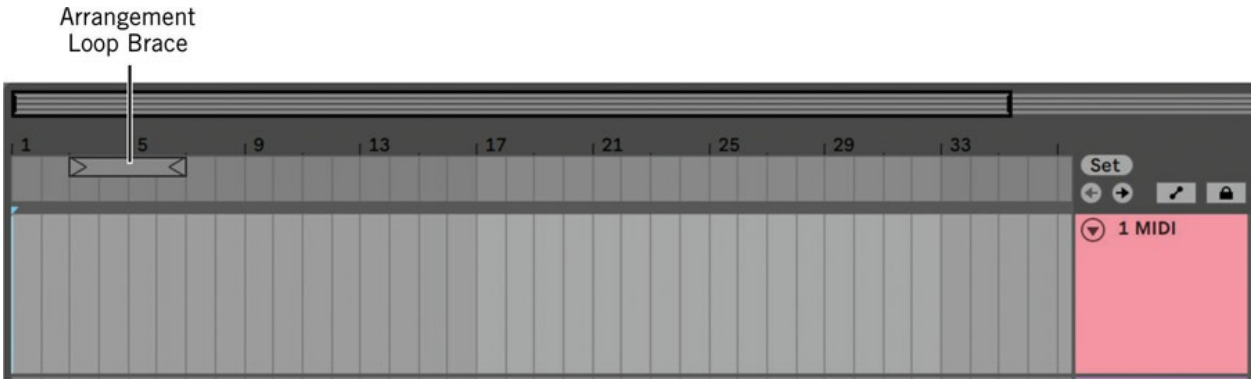


Figure 4.4 Arrangement loop brace in a new Set (shown inactive)

To activate the Arrangement loop, do one of the following:

- Click the **LOOP** switch in the Control Bar so that it becomes highlighted.

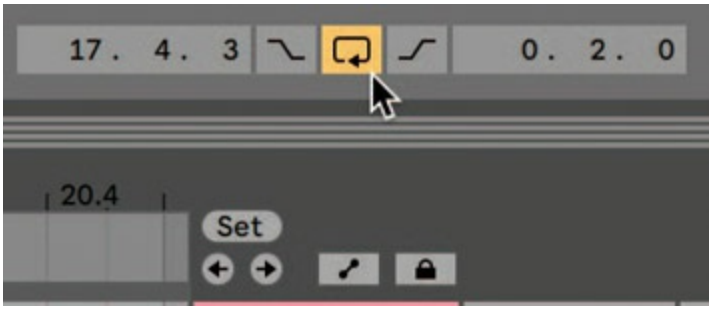


Figure 4.5 The Loop switch enabled in the Control Bar

- Choose **EDIT > ACTIVATE LOOP**.
- Press **COMMAND+L** (Mac) or **CTRL+L** (Windows).

When active, the Arrangement loop brace will become highlighted in the Scrub Area.

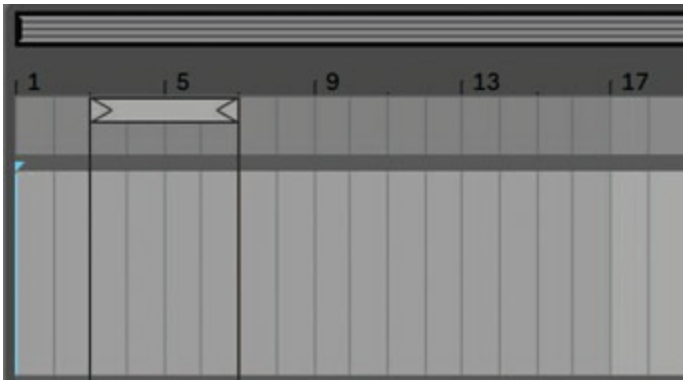


Figure 4.6 The Arrangement loop brace when the loop is active



For more information on the Arrangement loop, see Chapter 9 in this book.

Starting and Stopping Playback

You can set the playback point by clicking directly on a track to position the Arrangement Insert Marker. You can then start and stop playback by clicking the Transport controls in the Control Bar or by pressing the **SPACEBAR**.

To play back a portion of your Live Set, follow these steps:

1. Click anywhere within an existing track (avoiding clicking in the top of a clip).

2. Press the **SPACEBAR** to begin playback from this point.

3. To stop playback, press the **SPACEBAR** again.

To move to a different playback point in the track, click a new position and press the **SPACEBAR** again.

Toggling the Scrolling Mode

Live includes a scrolling option called **FOLLOW** mode that causes the display to scroll during playback, keeping the playhead visible in the Track Display window. As the Live Set plays, the Track Display will scroll, either page-by-page or continuously, depending on the Follow behavior that is set in Preferences.

To enable/disable Follow mode, do one of the following:

- Choose **OPTIONS** > **FOLLOW**.

- Press **COMMAND+SHIFT+F** (Mac) or **CTRL+SHIFT+F** (Windows).

- Click the **FOLLOW** button in the Control Bar. The button will be highlighted in a yellow/orange color when Follow mode is active.



Figure 4.7 The Follow button in the Control Bar (shown active)

Follow mode will pause if you make an edit, scroll horizontally, or zoom during playback. Follow will resume when you stop or restart playback. You can also re-engage Follow mode on-the-fly using any of the above methods.

For detailed information on Follow behavior and the associated settings in Preferences, see Chapter 9 of this book.

Locating the Playhead

At times, the playhead might be off screen. For example, when **FOLLOW** mode is disabled as described above, the cursor will move off screen after reaching the edge of the Track Display window in the Arrangement View.

Ableton Live displays the playhead cursor as a single vertical line in the Clip Overview at the top of the window in Arrangement View. To jump to the playhead, you can click on the cursor line in the Overview, using the magnifying glass icon. The Overview will scroll to that area of the Set.

Alternatively, you can zoom out by pressing the – (minus) key until the playhead comes into view.

Playback in Session View

While the Arrangement View displays your Live Set along a fixed timeline, the Session View displays a grid of tracks and clips that you can play at any time and in any order. The Session View is designed for improvisation and dynamic, non-sequential playback, so the layout of tracks and clips does not predetermine their playback order.

Getting to Know Session View

The Session View is the feature that truly sets Ableton Live apart from the linear production focus of most DAWs. In the Ses sion View, clips can be placed in a track's clip slots and triggered to play or stop independently. Each clip on a given track plays through the devices inserted on that track, so that the same instrument and/or signal processing is used by all clips on the track. You can also trigger an entire row of clips across all tracks simult aneously by launching a Scene.

Ableton Live Intro permits up to eight cl ip slots per track and scenes. Ableton Live Standard and Suite support an unlimited number of cli ps slots per track and scenes. Because Session View features non-sequential playback, it can be useful for sketching out songs and performing them into the Arrangement View. This is a classic Ableton Li ve studio production workflow. The Session View is also a powerful live performance tool. It can be used to play backing tracks for a live band, to perform DJ sets that are free-flowing yet tightly synchronized, or to fire off music cues and sound effects for a theatrical performance.

Starting and Stopping Playback

Each clip that you load in the Session View has a triangular **CLIP LAUNCH** button (Play button) at its left edge. Clicking this button will launch playback for the clip at any time. Launching a clip will also activate playback of the Arrangement, if it is not already in progress.



Figure 4.8 The triangular Clip Launch button

To stop a clip from playing, click the **CLIP STOP** button on any Clip Slot in the track or in the Track Status Display.

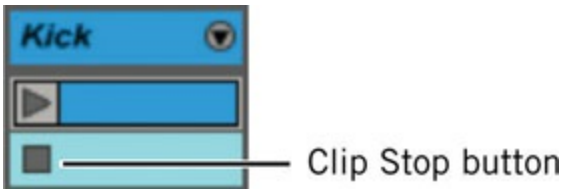


Figure 4.9 Click the square Clip Stop Button below an active clip to stop it from playing.

When you stop playback for a specific clip in the Session View, the Play button in the Control Bar will remain engaged and the Arrangement Position fields will continue running. These fields maintain a continuous flow of musical time so that you always know your position in song-time during a live performance or while recording clips into an arrangement.

You can stop playback by pressing the **STOP** button in the Control Bar or by pressing the **SPACEBAR**. Pressing the **STOP** button a second time (or anytime while playback is stopped) will return your Arrangement Insert Marker to the start of the entire Live Set.

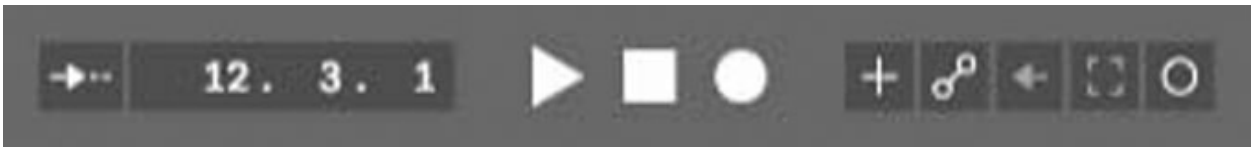


Figure 4.10 The Control Bar, including Play and Stop buttons, as well as the Arrangement Position counter

Enabling Clip Looping

In addition to Ableton Live's Arrangement Loop feature (discussed earlier in this chapter), Live also provides the ability to enable looping for individual clips. Enabling Loop functionality for a clip will allow that clip to loop indefinitely when launched in the Session View. When used the Arrangement View, a Loop-enabled clip can be resized beyond the length of its original media, with Ableton Live automatically adding successive iterations (repetitions) of the clip contents to achieve the desired length.

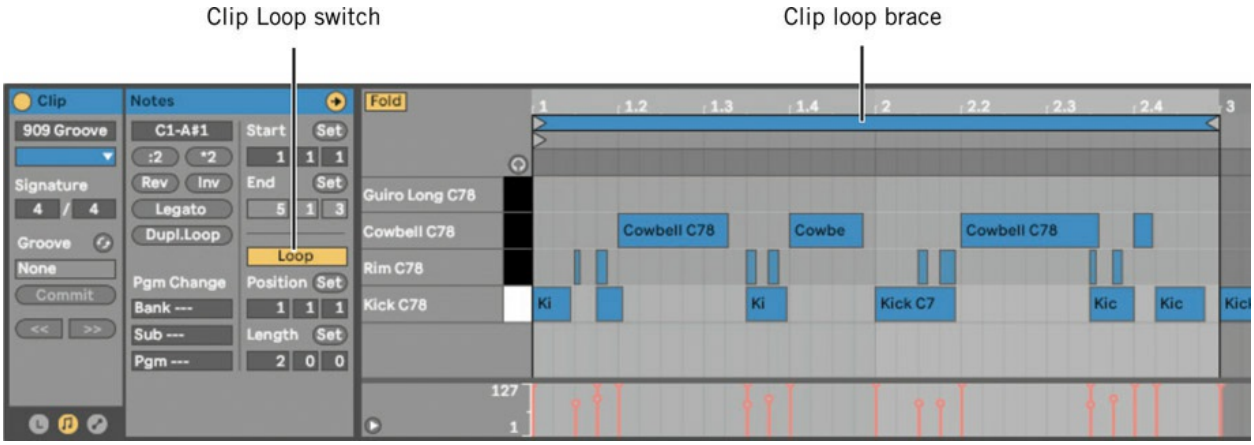


Figure 4.11 The Clip View, showing the Loop switch and loop brace for a MIDI clip

The Warp switch must be activated to loop an audio clip. Unwarped audio clips cannot be looped.

To enable clip looping:

1. Double-click on the target clip to open it in the Clip View.
2. If necessary, enable warping for the clip (required for audio clips).

For information on enabling and using Warping for audio clips, see "Audio, MIDI, and Tempo Changes" in Chapter 5.

3. Click the **Loop** switch in the Notes box (for MIDI clips) or in the Sample box (for audio clips).

To modify the length of the loop, do one of the following:

- Click and drag the triangle markers at either end of the loop brace to change its position and length.
- Type in exact position and length values in the Notes box (for MIDI clips) or the Sample box (for audio clips).

Saving, Locating, and Opening Live Sets and

Projects

As with most software applications, Live provides commands for saving and opening your files under the **FILE** menu. The following sections describe the options for saving, locating, and opening your Live Sets and other files related to your Live Projects.

Live Sets and Live Projects

The type of document that you create and edit in Ableton Live is called a Live Set. Live Sets are contained within a Live Project, which is the top-level folder containing Ableton Live-related files, such as audio files that belong to the Live Set.

When you save an existing Live Set under a new name and folder location, Ableton Live will create a new project folder and store the Live Set within that folder.

Saving a Live Set

While working in Ableton Live, it is important to save your work often. Note that when you save your progress in Live, you are saving only the Live Set document, not its associated files. (Audio clips in a Set either reference existing files on disk or are recorded directly to disk within the project folder, so you don't have to save them independently.) This means even very large Live Sets can be saved quickly.

Save Live Set Command

Saving can be done manually by choosing the **SAVE LIVE SET** command from the **FILE** menu. This saves the changes you have made since the last time you saved and writes the Live Set in its current form over the old version. You cannot undo the Save command.



If you need to return to an earlier state after completing a Save command, you can use the Undo command to work backwards through previous actions.

Save Live Set As Command

The **SAVE LIVE SET AS** command is useful for saving a copy of a Set under a different name or in a different drive location. Because this command leaves the original Set unchanged and allows you to continue working on the renamed copy, it is particularly useful for experimenting and saving successive stages of your work. You can save each stage under a different name, such as "Edit Session-Day 1," "Edit Session-Day 2," and so on. By working this way, you can always retrace your steps if you should want to go back to an earlier stage of the project.

To use the Save Live Set As feature, follow these steps:

1. From within an open Set, choose **FILE > SAVE LIVE SET AS**. The associated dialog box will appear.
2. Type a new name for the Set in the dialog box.
3. Optionally, navigate to a new drive location.
4. Click **SAVE**.

The renamed, newly saved Set will remain open for you to continue your work.



The Save Live Set As command should not be used to transfer a Live project to a portable drive. Doing so will copy the Live Set document file only, without including the associated audio files.

Save a Copy Command

The **SAVE A COPY** command simply saves a copy of the current Live Set with a different name or within a different directory location without affecting the currently open Set. After saving a copy, the original version of the project—not the copy—will remain open for you to continue your work.

To use the Save a Copy feature, follow these steps:

1. From within an open project, choose **FILE > SAVE A COPY**. The Save a

Copy dialog box will appear.

2. Type a new name for the project version in the dialog box.

3. Optionally, navigate to a new drive location.

4. Click **OK**.

Collect All and Save Command

The **COLLECT ALL AND SAVE** command simultaneously saves your current Live Set and packages all of the associated audio files into the Live Project folder structure. This is useful when you are saving a project for an archive or for use on another computer, or when you are sending the Live Set to a collaborator who doesn't already have all of the source audio files.

To use the Collect All and Save feature, follow these steps:

1. From within an open project, choose **FILE > COLLECT ALL AND SAVE**. The Collect All and Save dialog box will appear.

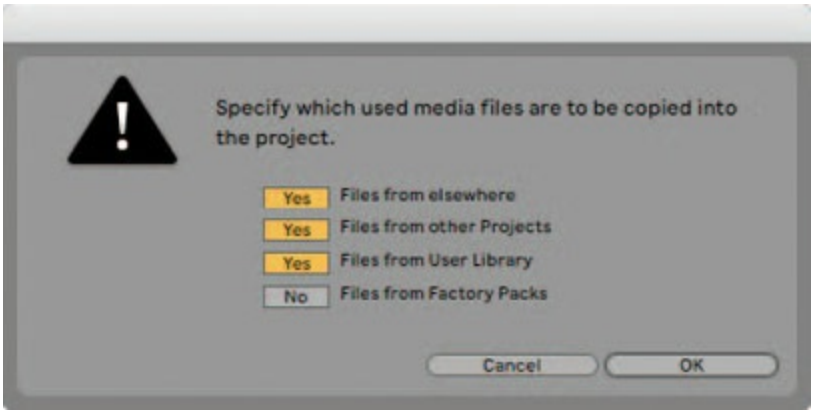


Figure 4.12 The Collect All and Save dialog box

2. Choose the types of audio files you want to save into the Live Project folder.

3. Click **OK** to complete the save. If you haven't previously saved the Live project, you'll be prompted to provide a new name for the project in a Save Live Set As dialog box.

The Collect All and Save command can be used when first saving a Live Set, or at any time while working on an existing Live project. If you use the Collect All and Save feature multiple times, Live will only collect audio files that have been added since the previous Collect All and Save command.



For information on using Collect All and Save to create a self-contained backup or archive of a Set, see Chapter 10 in this book.

Locating and Opening a Live Set

If you know the location of a Live Set file you want to open, you can open it directly from the File menu. In addition, Ableton Live makes it very easy to open a recently used Live Set regardless of the location.

To open a Live Set, do one the following:

- Choose **File > Open Live Set**.
- Press **Command+O** (Mac) or **Ctrl+O** (Windows).

To open a recently used Live Set, do the following:

- Choose **File > Open Recent Set** and select the file that you wish to open from the submenu.



You can also open Live Sets from the computer's hard drive by double-clicking on the Live Set file in an Explorer or Finder window.

Regardless of which method you choose, the Live Set will open with all tracks and clips appearing exactly as they were when last saved.



Ableton Live can have only one Live Set open at a time. If you attempt to open a Live Set while another is open, Live will prompt you to save the current Live Set and close it before opening the new one.

Review/Discussion Questions

1. What is the maximum sample rate supported in Live? (See “Audio Device and Sample Rate Settings” beginning on page 87.)

2. What audio file types are supported in Ableton Live? What is the default file type on Mac and Windows? (See “File Type and Bit Depth Settings” beginning on page 88.)

3. What is the maximum bit depth supported in Live? (See “File Type and Bit Depth Settings” beginning on page 88.)

4. What are the four basic track types supported in Live? What are Group Tracks? (See “Adding Tracks to Ableton Live” beginning on page 92.)

5. What menu command lets you add audio tracks to your Live Set? What menu command lets you add MIDI tracks to your Live Set? (See “Adding Tracks to Ableton Live” beginning on page 92.)

6. Why might Audio and MIDI tracks respond differently to changes in the tempo of a Live Set? What setting affects this behavior? (See “Audio, MIDI, and Tempo Changes” beginning on page 123.)

7. What happens to the audio and MIDI clips on a track when the track gets deleted from your Live Set? Can the delete command be undone? (See “Deleting Tracks” beginning on page 93.)

8. What are the two different views you can use to interact with a Live Set? (See “Controlling Playback” beginning on page 95.)

9. In Arrangement View, what is the Arrangement Insert Marker? How does it appear in the Live Set? (See “Playback in Arrangement View” beginning on page 95.)

10. How does playback differ between the Arrangement and Session views?

(See “Playback in Session View” beginning on page 98.)

1. What is the purpose of the Save Live Set As command? Which Set will be open after completing the Save Live Set As command—the original or the renamed copy? (See “Save Live Set As Command” beginning on page 101.)

2. How does the Collect All and Save command differ from the normal Save or Save Live Set As commands? (See “Collect All and Save” beginning on page 102.)

3. What are some ways you can open a Set from within Live? (See “Locating and Opening a Live Set” beginning on page 103.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 4

Creating a Live Set

Activity

In this exercise tutorial, you will create a new Live Set, configure your Ableton Live preferences, add and remove tracks for the Live Set, name each of the tracks you are keeping, and save the Live Set as an Ableton Live project for use in subsequent exercises.

Duration

This exercise should take approximately 10 to 15 minutes to complete.

Goals/Targets

- Create a new Live Set
- Configure audio device and sample rate
- Choose an appropriate file type and bit depth
- Add and rename Audio and MIDI Tracks

Downloading the Media Files

To complete the exercises in this book, you will need to use various media files included in the Media Files 2019-Live101 folder. If you

haven’t done so already, you can download the media files now. Be sure to save the files to a location th at you will have ongoing access to as you complete the exercises and projects in this book.

To download the media files, point your browser to www.halleonard.com/mylibrary and enter your access code (printed on the opening page of this book). Next, click the Download link for the Media Files 2019-Live101 listing in your My Library page. The Media Files folder will begin transferring to your Downloads folder.

Getting Started

To get started, you will need to open Ableton Live, create a new Live Set, and configure its audio settings (audio device, sample rate, bit depth, and file format) to suit your needs.

Launch Live and create a new Live Set:

1. Power up your computer and any connected hardware.
2. Do one of the following to launch Ableton Live:
   - Double-click on the Ableton Live shortcut icon on the desktop.
   - Click the Ableton Live icon in the Dock (Mac).
   - Click **START > ABLETON LIVE** (Windows).
3. Once Ableton Live has launched, click FILE > NEW LIVE SET from the menu bar. A new untitled Set will be created.

Configure Live’s audio device and Sample Rate:

1. Open the Preferences window by selecting **LIVE > PREFERENCES** (Mac) or **OPTIONS > PREFERENCES** (Windows).

2. Under the **AUDIO** tab, choose an appropriate Driver Type to allow Live to communicate with your available audio hardware.

3. Choose your Audio Device.

   If you do not have a dedicated Audio Device, you can choose your computer's built-in Audio Device.

4. Select **44100** for your Sample Rate.

Configure Live's File Type and Bit Depth settings:

1. Navigate to the **RECORD/WARP/LAUNCH** tab of Live's Preferences window.

2. Set the audio File Type to **AIFF** (Mac) or **WAV** (Windows).

3. Choose **24** for the Bit Depth.

4. When finished, close the Preferences window.

Creating and Naming Tracks

For this part of the exercise, you can work in either the Session View or the Arrangement View. You can switch between views by pressing the **TAB** key. Here, you will add and remove tracks as needed and give your tracks meaningful names.

Adding Tracks

By default, a new Live Sets contain two empty audio tracks and two empty MIDI tracks. In the next series of steps, you will add some new tracks, delete unneeded tracks, rename the tracks you will use in the Set, and reorder the tracks.

Create additional tracks for the Live Set:

1. Choose **CREATE > INSERT AUDIO TRACK** to insert a new audio track.



Try using the keyboard shortcut for creating an audio track: COMMAND+T (Mac) or CTRL+T (Windows).

2. Choose **CREATE > INSERT MIDI TRACK** to insert a new MIDI track.



Try using the keyboard shortcut for creating a MIDI track: COMMAND+SHIFT+T (Mac) or CTRL+SHIFT+T (Windows).

This should give you a total of three audio tracks and three MIDI tracks, along with the default Reverb and Delay return tracks and the Master track.



The number of tracks included in a new Live Set can vary, depending on the Default Template being used by Ableton Live. If the Default Template has not been changed, your Set will include two audio tracks, two MIDI tracks, and two return tracks.

Deleting Tracks

For this project, we only need two audio tracks and two MIDI tracks, so let's go ahead and remove any extra audio and MIDI tracks we have. To simplify the Set, we'll also remove the return tracks for now.

The following four tracks should be deleted from the Set:

- The third audio track
- The third MIDI track
- The Reverb return track
- The Delay return track

Delete each of the above tracks using one of the following techniques:

- Right-click on the track name and choose **DELETE** from the pop-up menu.

■ Select the track, then press the **DELETE** key on the computer keyboard.

Now you'll want to rename the remaining tracks so that you'll be ready to starting producing music.

Name the four remaining tracks:

1. Right-click on the track name of the first track and select **RENAME** from the pop-up menu.
2. Change the names for the audio tracks as follows:
   - Name the first audio track Chords.
   - Name the second audio track Bass.



Try using the Tab key to automatically move the Rename cursor to the next track.

3. Rename the MIDI tracks as follows:
   - Name the first MIDI track Drums.
   - Name the second MIDI track Melody.
4. Press **RETURN** or **ENTER** on the last track to finish renaming.

The last step is to make sure the tracks are in the correct order for the next exercise.

Move the Drums track:

■ Click the on track name for the Drums track and drag it to top position in the Arrangement View or the left-most position in the Session View.

The final track order should be:

- Drums

- Chords
- Bass
- Melody

Finishing Up

To complete this exercise, you will need to save your work under a new name and close the Live Set. Note that you will be reusing this project in Exercise 5, so it is important to save the work you've done.

Finish your work:

1. Choose **File > Save Live Set As** to save the Live Set and create a Live Project folder structure containing the associated Live project files.
2. In the Save Live Set As dialog box, rename the Set as *Exercise04-XXX* (where *XXX* is your initials), and click **Save**.
3. Press **Command+Q** (Mac) or **Ctrl+Q** (Windows) to quit Live and close the project.

Chapter 5

Importing and Working with Media

This chapter introduces various processes for importing audio and video files into a Live Set. We describe file formats and types that can be imported and cover how to use Ableton Live's Browser to locate and preview audio files. We go on to discuss methods for importing audio and video files in both the Arrangement View and Session View.

Learning Targets for This Chapter

- Understand the file resolutions and formats that can be imported into Live
- Locate supported audio and video files using Live's Browser
- Use both the Arrangement View and Session View to import audio files as clips on tracks
- Import video files and perform basic video editing tasks

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform. Sign up for free at ElementsED.com.

As a music producer, composer, sound designer, or engineer, you'll encounter many situations in which a project will start with some kind of existing media. You might find inspiration in a pre-created musical loop, such as a drum loop or a chord progression, and want to use that as a starting point for a new song. A collaborator might send you tracks they created so you can continue to develop or mix the idea in your own studio. You might receive video from a director whose film needs a score. Perhaps you just want to browse and use your own samples and sound effects.

In these situations and many others, you will be working with existing media files—and you will need a way to get them into Live. In this chapter, you will learn the types of files you can import into a Live Set, as well as how to import them.

Considerations Prior to Import

Ableton Live allows you to add audio, MIDI, and video files from any available storage location to your Live Set. There are multiple ways to get media files onto tracks in your Live Set, including dragging and dropping files from a file browser and locating files through Live's built-in Browser.

Prior to attempting to add a file to your Live Set, you should understand whether it is compatible with Live.

File Characteristics

Live supports audio, MIDI, and video files of various resolutions, types, and formats. In order for a media file to be recognized in a Live Set, it must meet certain criteria.

Supported Audio Bit Depth and Sample Rate

Live supports mono and stereo audio files with bit depths of 8-bit, 16-bit, 24-bit, or 32-bit, and sample rates ranging from 44.1 kHz to 192 kHz. You can mix and match audio files of different bit depths and sample rates in a single Live Set without needing to convert them to match the Set's resolution.

Supported Audio File Formats

Ableton Live features native support for two audio file formats: Audio Interchange File Format (AIFF) and Waveform Audio File Format (WAV). WAV and AIFF are the only audio file formats that Live can load without conversion.

Live can also play Sound Designer II files, as well as compressed formats such as MP3, AAC, Ogg Vorbis, and FLAC files. These formats are converted into WAV when imported. The converted file is stored in Live's Decoding Cache:

- Windows: System Drive (C:) > Users > [username] > AppData > Roaming > Ableton > Cache > Cache > Decoding
- Mac: Macintosh HD > Users > [username] > Library > Caches > Ableton > Cache > Decoding



When compressed or encoded audio files such as MP3s are added to a Live Set, Ableton Live automatically decodes them for use and stores the decoded audio files in the Decoding Cache. You can set the maximum size and location of the Decoding Cache in Live's Preferences window, under the FILE/FOLDER section.

Here is an overview of the types of audio files that can be imported in Ableton Live:

- **Audio Interchange File Format (AIFF).** This file format is used primarily on Macintosh systems, although Live supports it natively on both Mac and Windows. AIFF files can be recorded directly or imported into a Live Set.
- **Audio Interchange File Compressed (AIFC).** A variant of the AIFF audio file standard, AIFC provides audio compression for AIFF files.
- **Waveform Audio File Format (WAV).** Live supports any standard WAV (WAVE) files. WAV files can be recorded directly and imported on either Mac or Windows systems.

- **Sound Designer II (SD II).** SD II is a mono or stereo file format supported on Macintosh systems only. This format supports sample rates up to 48 kHz.

- **MP3 (MPEG-1 Layer-3).** MP3 files are supported on all common computer platforms and employ file compression of up to 10:1, while still maintaining reasonable audio quality. Because of their small size and cross-platform support, they have traditionally been popular for streaming, social media, and file exchanges.

- **Advanced Audio Coding (AAC).** AAC files were created as a successor to MP3. This format generally achieves better sound quality than an MP3 at the same bit rate. AAC is the standard audio format for platforms such as YouTube and iOS devices.

- **OGG Vorbis (OGG).** Vorbis is an open-source compressed audio format, which is often used in conjunction with the OGG container format.

- **Free Lossless Audio Codec (FLAC)**. FLAC is an audio format for lossless audio compression. FLAC compression typically reduces audio files to between 50 and 70 percent of their original size.

Supported MIDI Files

Live can import Standard MIDI Files (SMFs) in both common formats—type 0 and type 1—as well as the much less common type 2 format. Supported MIDI files are recognizable by their **.mid* or **.smf* file extensions.

Type 0 Standard MIDI Files (SMF0)

In type 0 SMFs, all MIDI data is stored in a single track, separated by the MIDI channel. When imported into Live, all MIDI data will appear in one track.

Type 1 Standard MIDI Files (SMF1)

In type 1 SMFs, individual parts are saved on different tracks within the MIDI sequence. When imported into Live, each part will appear on its own

track.

Up until Live 9, SMF1 formatted files appeared as a folder in the Browser. From there, the folder could be opened to import selected tracks, if necessary.

Type 2 Standard MIDI Files (SMF2) (Rare)

In type 2 SMFs, the MIDI file can contain multiple tracks, and each track can contain a different and complete sequence (essentially a collection of type 0 sequences in a single MIDI file). When imported into Live, each sequence will appear on its own track.

Supported Video Files

Live supports video files in Apple QuickTime format (MOV), allowing you to perform sound design and scoring to video within a Live Set. Supported video files appear in Live's built-in Browser and can be imported by dragging them into a Live Set when working in the Arrangement View.

Import Methods

Ableton Live provides several methods for bringing audio and MIDI files into your Live Set: you can import from Live's built-in Browser, use the **Create > Import Audio/MIDI File** command, or drag and drop files from an external file browser.

In Live 10.1 and later, you can also import MIDI to your Set by choosing Create > Import MIDI File.

The following sections provide details for importing audio, starting with a focus on the built-in Browser, as it offers useful search and organizational features.

Live's Built-in Browser

Live's Browser makes it easy to find and audition supported media files, and

ultimately to insert them as clips on tracks. While the Browser provides access to much more than just audio files, by default it displays the core media included with Live and allows you to add additional folders of media from any available storage location on your system.

Showing the Browser

The Browser is located on the left-hand side of Live's interface. The Browser display is divided vertically, with the left-hand column displaying categories of tools, sounds, and media files, and the right-hand column listing the contents of each category.

To show and hide the Browser, select **View > Show Browser** or press **Option+Command+B** (Mac) or **Ctrl+Alt+B** (Windows). Alternatively, you can toggle the browser using the Show/Hide triangle in the upper left-hand corner of the Ableton Live interface.

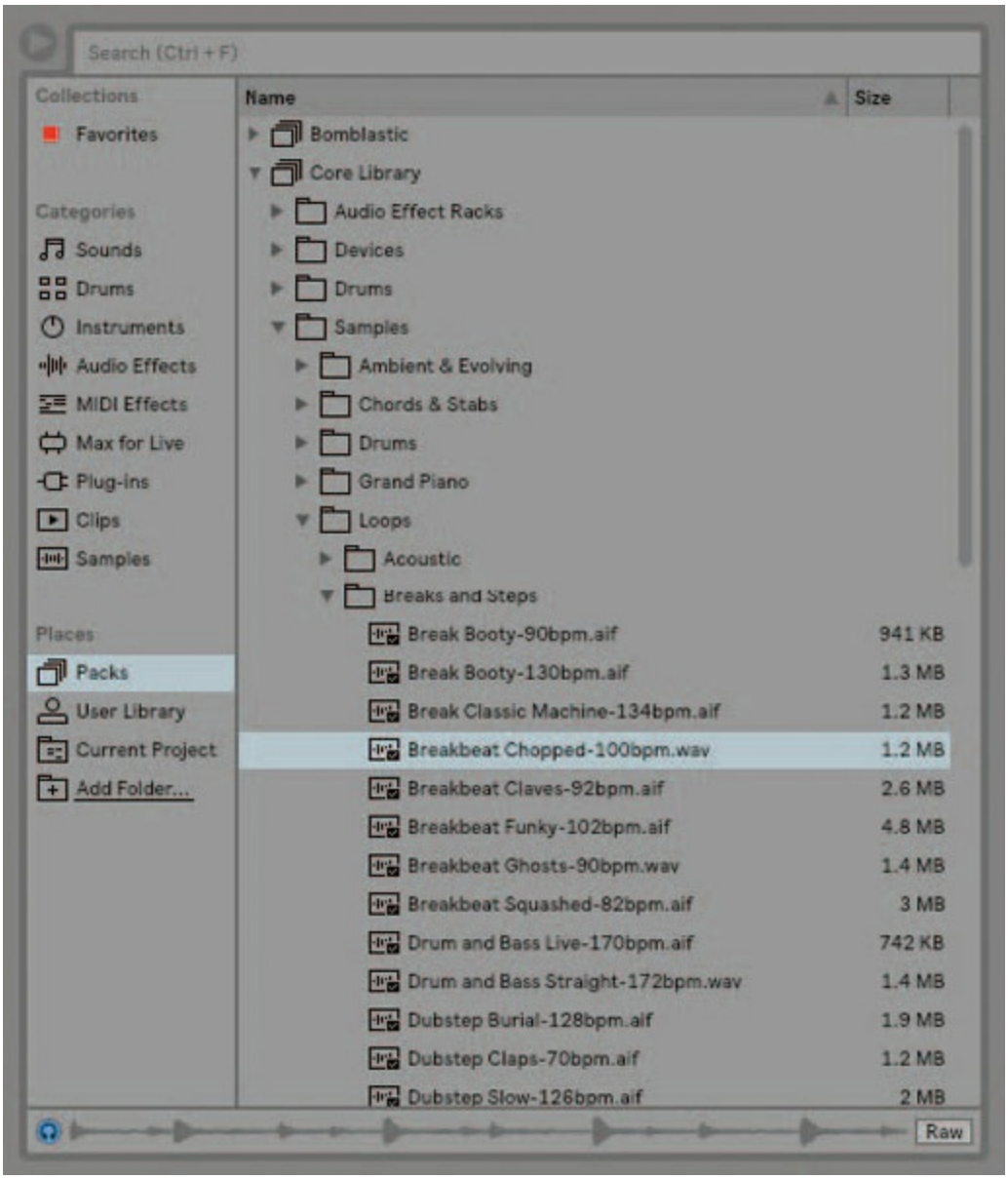


Figure 5.1 The Browser showing audio files included in Live's Core Library Pack.

When it comes to importing audio files, the Browser provides a few useful starting points:

- ***Samples***: This Browser category lists all of the audio samples available to Ableton Live. If you've accumulated a large volume of samples, this list can become very long, making the search function at the top of the browser very important.

- ***Packs***: This category displays any official Ableton Packs you have

installed and any Packs that you have available to install (Live 10 and later). Some Packs contain a large volume of individual audio files.

- ***User Library***: This is the default storage location for the various components of a Live Set, including your own samples and other audio files. When you add folders and audio files to the User Library, Ableton Live will automatically display them in this section of the Browser.

Notably, you can choose the **ADD FOLDER** option under Places in the Browser to select additional storage locations and make them available for browsing.

Navigating the Browser

Once you have selected a category, you will be able to navigate that section in the right-hand column. If the category has additional folders, you can expand and collapse them using the disclosure triangles next to each folder name. Alternatively, you can use the right and left arrows on your keyboard to expand and collapse folders, and the up and down arrows to move your selection through the list of folders and files.

Searching the Browser

It doesn't take long to accumulate a large volume of audio files in your library. The larger your audio library becomes, the more important Live's search and collections features become.

To search the browser and filter the files it displays:

1. Do one of the following:
   - Click in the Search field at the top of the browser.
   - Press **COMMAND+F** (Mac) or **CTRL+F** (Windows) to place your cursor in the Search field.
2. Enter a search term to browse for matching files.

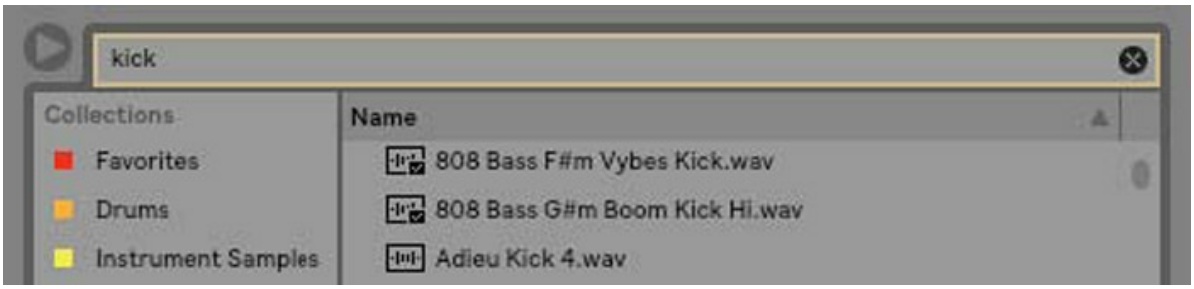


Figure 5.2 Searching the Browser for audio files with “kick” in the name

As you enter a search term, Ableton Live will automatically begin filtering the files it displays. For instance, searching for kick will cause Ableton Live to only show files with the text “kick” in their file names.

Using Collections

Collections offer an additional opportunity to organize your browser contents and ultimately find what you’re seeking faster.

Ableton provides seven Collection labels that you can use to tag items in the Browser. Each label has its own unique color and name, which you can customize as desired. You might use one Collection label to mark your favorite drum samples, and another to delineate your favorite instrument samples. You can use the Collection labels to tag anything you want, including audio effects, instruments, plug-ins, and clips.

To add or remove a Collection label tag for an audio file:

1. Locate the target audio file in the Browser.
2. Do one of the following:
    - Right-click the audio file and select the Collection label you’d like to assign or unassign.
    - Press keys **1-7** to assign or unassign the associated Collection label.

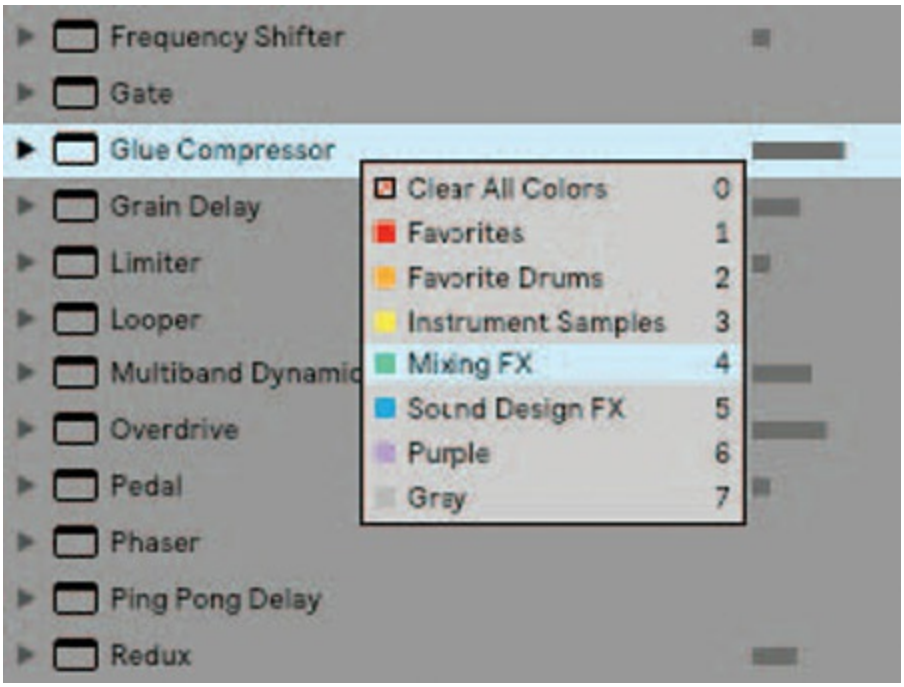


Figure 5.3 Right-click any item in the browser to assign or unassign a Collection label.

To view the contents of a Collection:

1. Click the name of the Collection in the upper, left-hand corner of the browser. The browser will display a combined list of everything contained in the Collection.

2. Use the search feature in conjunction with viewing a Collection to narrow your results even further.

3. De-select the Collection to return to normal browsing.

Previewing Audio Files in the Browser

By default, when you click on or select an audio file name, Live automatically previews the audio. You can turn off the automatic preview by clicking the PREVIEW button in the lower left-hand corner of the browser. (See Figure 5.4.)

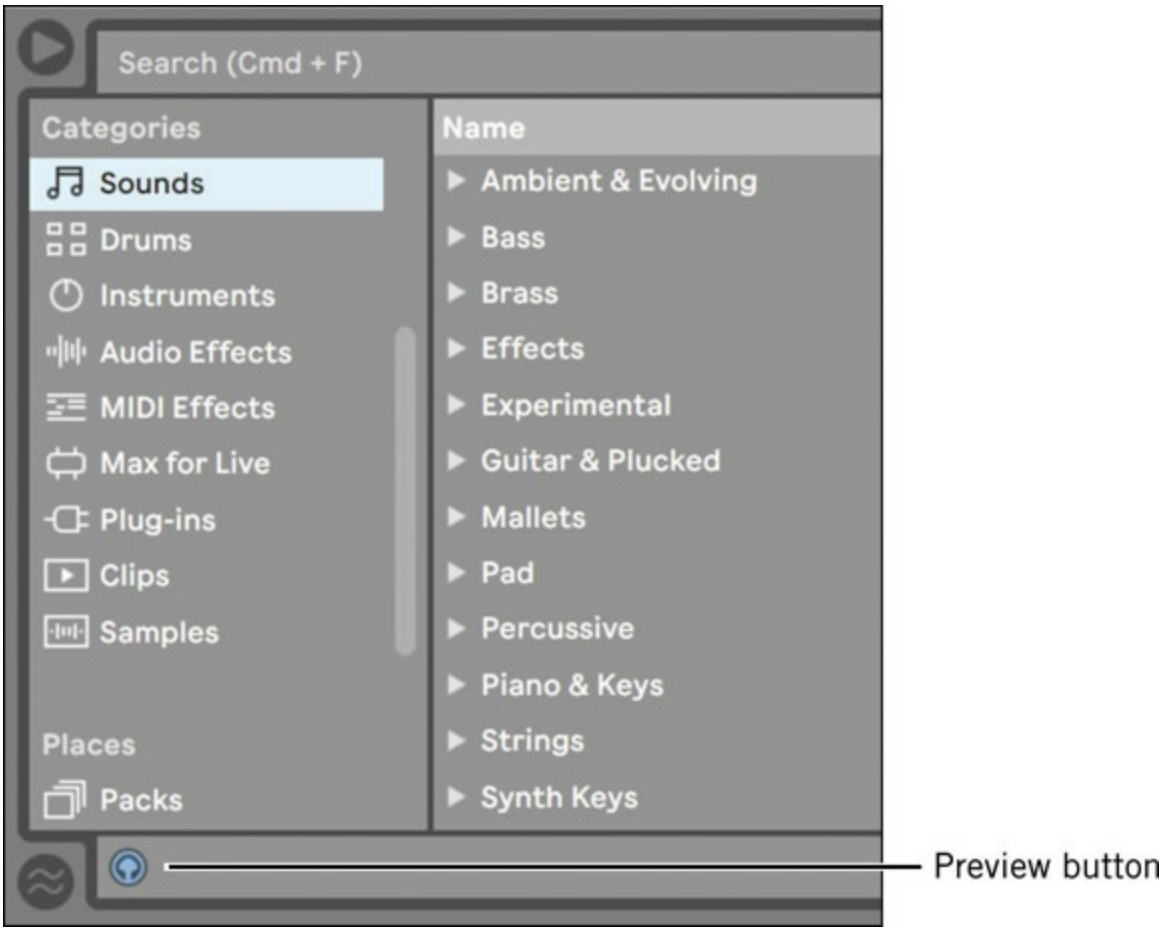


Figure 5.4 The Preview button (highlighted blue) in the lower left-hand corner of the browser.

After selecting an audio file, you will also see its waveform display at the bottom of the browser window, next to the Preview button. If the Preview button is enabled (highlighted in blue), a playhead will move across the waveform to indicate the current playback position of the sample. You can also click anywhere in a sample waveform to start playback from that specific location.

To stop the preview, you can follow any of these options:

- Press the **SPACEBAR**.
- Select another audio file or folder in the browser.
- Press the **PREVIEW** button to turn off audio previews altogether.

You can trigger Preview for a selected audio file at any time by pressing the right arrow key or Shift+Return / Shift+Enter with an audio file selected in the browser. This works even with the Preview button turned off.

When applicable, Live attempts to preview any type of file in the browser in sync with the current Live Set's tempo, so you can preview it in context. To prevent this behavior, click the **RAW** button to enable it.

Previews play at the current tempo only while the Set is in playback and the Raw button is not engaged.

With the Raw button engaged (highlighted in blue), files will preview at their original tempo and will not loop during previews.

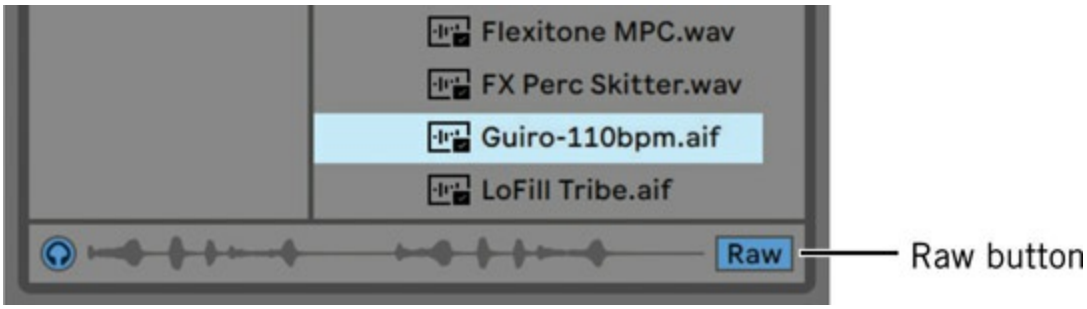


Figure 5.5 The Raw button engaged (highlighted) in the Preview display

You can adjust the volume of audio previews using the **PREVIEW/CUE VOLUME** knob on the Master track in the Session View.

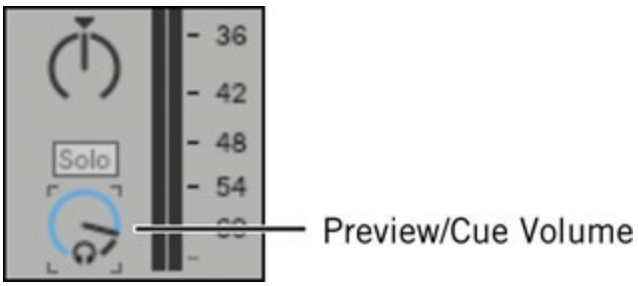


Figure 5.6 Use the Preview/Cue Volume knob on the Master track to adjust Live's preview volume.

Importing Audio from the Browser

After you have located an audio file (or files) in the Browser that you would like to use in your Live Set, your next step is to get the audio to an audio track. You can import files to existing tracks or create new tracks for the audio as you import files.

Importing Audio to Existing Tracks

You can import to existing tracks in either the Arrangement View or Session View; either way, Ableton Live will create an audio clip that you can edit however you see fit.

Import to Tracks in Arrangement View

When you're working in the Arrangement View, you'll see a linear, timeline-based view of your production.

To place audio from the browser on an audio track in the Arrangement View:

1. Select the audio file in the browser.

2. Drag the audio file onto an existing audio track. You can drag the clip anywhere in the arrangement.

Use the time rulers along the top and bottom of the Track Display window to locate the desired insert position, such as Bar 9 or the 0:30 second mark.

Import to Tracks in Session View

When you're working in the Session View, you will have access to a series of scenes and clips. You can launch clips and scenes in any order to generate performances without being tied to a linear timeline.

While the Session View environment is very different from that of the Arrangement View, the process of placing audio files as clips on tracks is very similar.

To place audio from the browser on an audio track in the Session View:

1. Select the audio file in the browser.

2. Click and drag the audio file to a clip slot on an audio track. The slot will update to display a play button and the associated clip name.

With the clip added to a clip slot, you can launch the audio file using either the clip launch button or the associated scene launch button.

Import Directly to a New Audio Track

You can skip the step of manually creating a destination track for an audio file, if desired.

To automatically place an audio file on a *new* audio track, drag the file from the browser to any position *below* the last track (Arrangement View) or *to the right of* the last track (Session View) in your Live Set. Live will create a new audio track at that position containing the clip.

Batch Importing Audio to Tracks

At times you might find yourself in a situation where you'd like to import multiple audio files simultaneously. In these cases, you can quickly send a set of selected files to one or more tracks.

Batch Import in Arrangement View

The Arrangement View offers two methods for importing multiple audio files to tracks. You can import the audio as a contiguous string of clips on a single track, or you can place each audio file on its own track at the same start position.

To place the audio files as contiguous clips on a single audio track:

1. Select the audio files in the browser:

    - Hold **SHIFT** to select a contiguous range of files, or

    - Hold **COMMAND** (Mac) or **CTRL** (Windows) to select non-contiguous files

2. Click and drag the audio files, placing them on the target audio track.

To place each audio file on its own audio track:

1. Select the audio files in the browser using the methods described above.

2. While holding **COMMAND** (Mac) or **CTRL** (Windows), drag the audio files to the audio track where you would like the first file to be placed. Subsequent files will automatically be placed on tracks below.

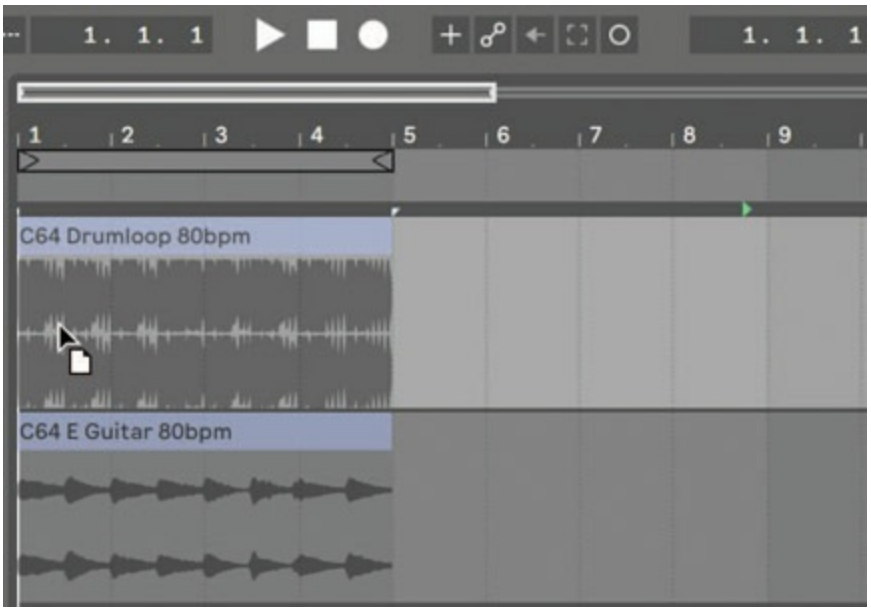


Figure 5.7 Simultaneously dragging multiple audio files to different tracks in the Arrangement View with the Command/Ctrl modifier

Batch Import in Session View

The Session View also offers two methods for importing multiple audio files to tracks. You can import the audio into subsequent clip slots on a single track, or you can place each audio file on a separate track in the same clip slot position.

To place the audio files into subsequent clip slots on the same audio track:

1. Select the audio files in the browser using the methods described above.

2. Click and drag the audio files to a clip slot on an audio track. The clips will be added to successive slots on the track. (See Figure 5.8.)



Figure 5.8 Simultaneously dragging multiple audio files to the same track in the Session View

To place the audio files into clip slots on separate tracks:

1. Select the audio files in the browser using the methods described above.

2. While holding **Command** (Mac) or **Ctrl** (Windows), drag the audio files onto the desired clip slot position on an audio track. The clips will be added to successive audio tracks at the same slot position.

Batch Import Directly to New Audio Tracks

You can skip the step of manually creating destination tracks for the audio files you plan to import, if desired, and instead create the tracks automatically upon import.

To place audio files on one new audio track:

- Drag one or more files from the browser to any position below the last track (Arrangement View) or to the right of the last track (Session View) in your Live Set.

  A new track will be added at that position, and the audio file(s) will be added to the track.

To place each selected file on its own new audio track:

- Hold **COMMAND** (Mac) or **CTRL** (Windows) while dragging multiple files from the browser to any position below the last track (Arrangement View) or to the right of the last track (Session View) in your Live Set. New tracks will be added at that position as needed.



You can easily to recognize the new track drop area by the displayed text label, "Drop Files and Devices Here."

Audio, MIDI, and Tempo Changes

Although audio and MIDI tracks can coexist in Live Sets, they can respond differently to changes in tempo because their source material is different – audio tracks contain *samples*, while MIDI tracks contain *data*. MIDI clips easily conform to changes in your Live Set tempo, as the MIDI data is flexible. Audio clips, on the other hand, can be absolute, meaning Ableton Live will continue to play them back at their original speed regardless of the project tempo. This can be desirable for non-rhythmic audio, such as spoken words or ambient textures, but it can cause rhythmic audio to fall out of sync in the context of a song.

With Warp algorithms, Ableton Live is capable of time-warping audio clips while streaming them from disk so as to synchronize them with the Live Set's tempo. Tempo matching is done without affecting the pitch, which can be changed independently. As such, even if you mix and match audio files from various sources with MIDI data, your Live Set remains flexible to tempo changes.

To enable time-warping on audio tracks, do the following:

1. Select one or more audio clips.

2. If Clip View is not already enabled, show the Detail View as needed (**VIEW > SHOW/HIDE DETAIL VIEW**) and then switch to Clip View as needed (**VIEW > TOGGLE CLIP/DEVICE VIEW**).

3. Click the **WARP** button from the Sample box in the Clip View. (See Figure 5.9.)

   The selected Audio clip(s) will now automatically follow tempo changes in your Set and conform to the project tempo map.

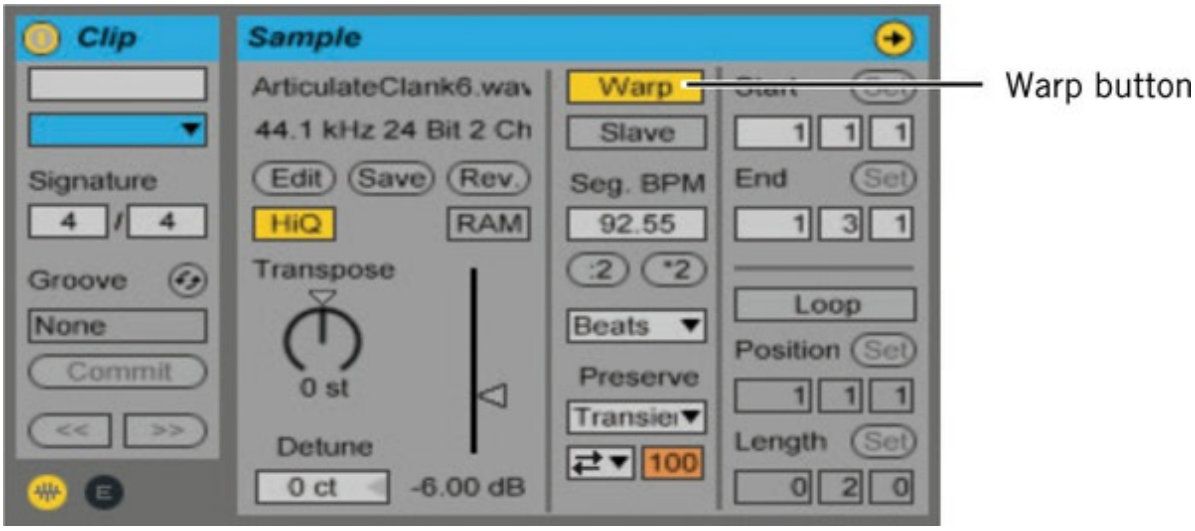


Figure 5.9 The Warp button when activate (highlighted yellow) in the Sample box within the Clip View area

Warping audio, especially when done to extremes, can create undesired audio artifacts. Always check the results and use the Edit > Undo command if you are not happy with the outcome.

Other Methods of Importing Audio Files

Audio files can be imported using menu commands or using a file browser from your computer's operating system, rather than using the built-in Browser in Ableton Live.

Importing with a Menu Command

Ableton Live gives you the option of locating and importing audio from the **CREATE** menu. This option can be useful if you know the disk location of the audio files you want to import, such as when you've downloaded files from an online resource or when you are importing shared files from a USB drive.

To import audio using the Create menu:

1. Select the audio track on which you'd like to insert the audio file.
    - In Arrangement View, position the Insert Arrangement Marker where you want to insert the audio clip.
    - In Session View, select the clip slot in which you would like to place the audio clip.
2. Select **CREATE > IMPORT AUDIO FILE**.
3. Navigate the file browser to the location of the audio file.
4. Double-click on the file to import it.

Importing from an External File Browser

You can also import audio files by drag-and-drop from an external file browser, such as a Finder window on Mac or a File Explorer window on Windows.

To import files to a single track:

1. Locate and select the files using a file browser on your operating system.
2. Drag the files from the file browser into the Ableton Live window. The files will be added to a single track at the location where they are dropped (on an audio track, on a new track, or in a clip slot).



Hold COMMAND (Mac) or CTRL (Windows) when importing multiple files to place them on separate tracks.

Importing Video

When you are scoring and/or performing sound design for any sort of visual media, it is often best to bring the video directly into your Live Set. This way, you can not only perform basic editing—such as trimming the start and end of the clip and breaking the clip into smaller segments—but also configure your Live Set so your composition aligns with edits and key "hit points" in the video.



Video will display in in the Arrangement View only. If you import a QuickTime file into a clip slot in the Session View, only the audio will import for playback.

To import a video into your Live Set:

1. Select the video file (MOV) from the Live browser or an external file browser.

2. Drag the file into the Track Display window in the Arrangement View. A video track will be created in the Set and will display the audio waveform from the QuickTime movie.



Unlike some other DAWs, Ableton Live displays an audio waveform on its video tracks instead of thumbnail images of frames from the video file.

Live supports video files in Apple QuickTime (MOV) format only. You will need to have QuickTime installed on your computer in order for Live to be able to decode video files for playback in a Live Set.

Video Track Display

Regardless of whether you drop the video file on a MIDI track, an audio track, or an empty space in the Arrangement View, the track will become a video track with the video clip (waveform display) inserted in the timeline.

Ableton Live will also display a floating Video Window, reflecting the current position of the Insert Arrangement Marker (i.e., the playback position).

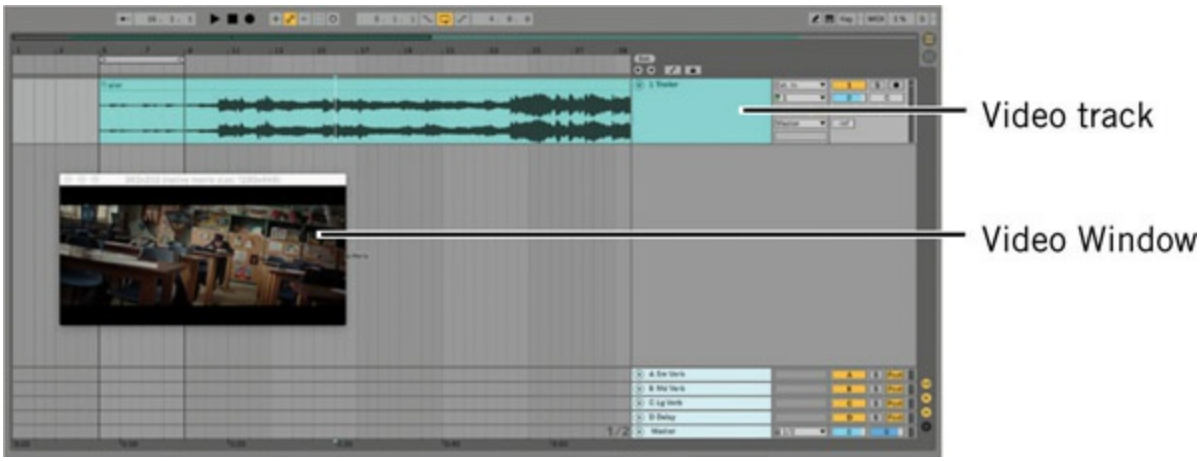


Figure 5.10 An imported video clip and associated video window in the Arrangement View

Ableton Live Intro and Lite editions do not support video import and playback.

The Video Window can be hidden at any time to reduce on-screen clutter and shown again later as needed.

To show or hide the Video Window, do one of the following:

- Choose **View > Video Window**.
- Press **Command+Option+V** (Mac) or **Ctrl+Alt+V** (Windows).

The Video Window can also be resized as needed to fit comfortably within the available display space on your screen.

To resize the Video Window, do one of the following:

- Click and drag the lower right-hand corner of the Video Window to resize it.
- Double-click the Video Window to display it in full-screen mode.

- **OPTION-CLICK** (Mac) or **ALT-CLICK** (Windows) anywhere on the Video Window to restore it to its original size.

Working with Video Tracks

Aside from only functioning in the Arrangement View, video clips look and function similarly to audio clips. You can move a video clip around the timeline, make separations in the clip by using the **EDIT > SPLIT** command (or by pressing **COMMAND+E** [Mac] or **CTRL+E** [Windows]) with the Arrangement Insert Marker positioned within the clip, and trim the start and end points of the clip.

If you attempt to perform more complex editing functions on the video clip, however, it will be converted to audio only. The Consolidate, Reverse, and Crop functions will all cause this conversion to occur.

Configuring the Start of a Video Clip

If you're scoring a piece of video, it can be helpful to reposition the start marker of your video clip so that it's right in line with the first frame of action.

To reposition the start marker within a video clip:

1. Double-click the video clip in the Arrangement View to display the Clip View for the video.

2. Click and drag the Start Marker (lower arrow in the upper left-hand corner) to the right.

3. Position the Start Marker where the first frame of the video appears in the Video Window.



The Start Marker in the Clip View determines where the clip will start playing when it is launched or encountered on a track.



Figure 5.11 Adjusting the video Start Marker location

You can now align the start of the video with the start of your composition.

Review/Discussion Questions

1. What two audio file formats are natively supported by Live? (See “Supported Audio File Formats” beginning on page 112.)

2. Name some of the other common audio file formats that Ableton Live supports. (See “Supported Audio File Formats” beginning on page 112.)

3. What automatically happens when you import an encoded or compressed audio file into a Live Set? (See “Supported Audio File Formats” beginning on page 112.)

4. What video file format does Ableton Live support? (See “Supported Video Files” beginning on page 114.)

5. What are the three methods you can use to import an audio or video file into a Live Set? (See “Import Methods” beginning on page 114.)

6. How can you show and hide the Ableton Live browser? (See “Showing the Browser” beginning on page 114.)

7. Once you’ve located an audio file in the browser, how can you import it into your Live Set? (See “Importing Audio to Existing Tracks” beginning on page 119.)

8. How can you simultaneously import multiple audio files with each file being placed on its own new audio track? (See “Batch Import Directly to New Audio Tracks” beginning on page 122.)

9. How do you insert a video file that you’ve located in the browser as a clip on a Video track? (See “Importing Video” beginning on page 125.)

10. How can you show and hide the floating video window once you have inserted a video file on a track? How can you re-size it? (See “Video Track Display” beginning on page 125.)

1. What kinds of editing functions are available on video clips? What editing functions are *not* applicable? (See “Working with Video Tracks” beginning on page 126.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 5

Importing Audio

Activity

In this exercise, you will use the Browser to import audio files into the Live Set you created in the previous exercise. You will import two fragments of a musical idea: a chord progression and a bass line, both created with synths. You can choose to place these audio files on tracks using either the Arrangement View or Session View, depending on how you prefer to work.

Duration

This exercise should take approximately 10 to 15 minutes to complete.

Goals/Targets

- Add a File Folder to the Browser
- Import Audio to Tracks in Arrangement or Session View

Media Files

To complete this exercise, you will need to use various audio files included in the Media Files 2019-Live101. You should have downloaded the media files in Exercise 4.

If needed, you can re-download the media files by going to

www.halleonard.com/mylibrary and entering your access code (printed on the opening page of this book). From there, click the Download link for the Media Files 2019-Live101 listing in your My Library page. The Media Files folder will begin transferring to your Downloads folder.

Getting Started

You will start by opening the Live Set you saved at the end of Exercise 4. If that file is not available, you can use the Ex04 Sample file in the 01. Completed Exercises folder within the Media Files 2019-Live101 folder.

Open the Set and save it as Exercise 5:

1. Open the Set file that you created in Exercise 4 (Storage Drive/Folder > Live101-XXX > Exercise04-XXX.als).

If your Exercise 4 Set is not available, you can use the Exercise 4 Sample file instead (Media Files 2019-Live 101 > 01. Completed Projects > Ex04 Sample.als).

2. Choose **File > Save Live Set As** and name the Set *Exercise05-XXX*, keeping the Live Set inside the original Project folder. (Move the Set into your Live101-XXX folder if working from the sample file.)

The Live Set will open with the four tracks—two Audio and two MIDI—that you created in Exercise 4.

Add the Audio File Folder to the Browser

As discussed in Chapter 5, there are multiple ways of importing audio files into a Live Set. For this exercise, you will use Live's built-in Browser, as its functionality makes it well worth learning. To get started, you will add the Exercise Media folder to the Browser.

Add the sample media folder to the browser:

1. Show the Live browser by clicking **View > Show Browser** or pressing **Option+Command+B** (Mac) or **Ctrl+Alt+B** (Windows).

2. Click the **Add Folder** button at the bottom of the browser's left-hand column.

3. Navigate to the Media Files 2019-Live101 > 02. Exercise Media location.

4. Click **Select Folder**. The 02. Exercise Media folder will display in the Live Browser.

5. Click on the 02. Exercise Media folder in the Browser to display the audio files it contains.

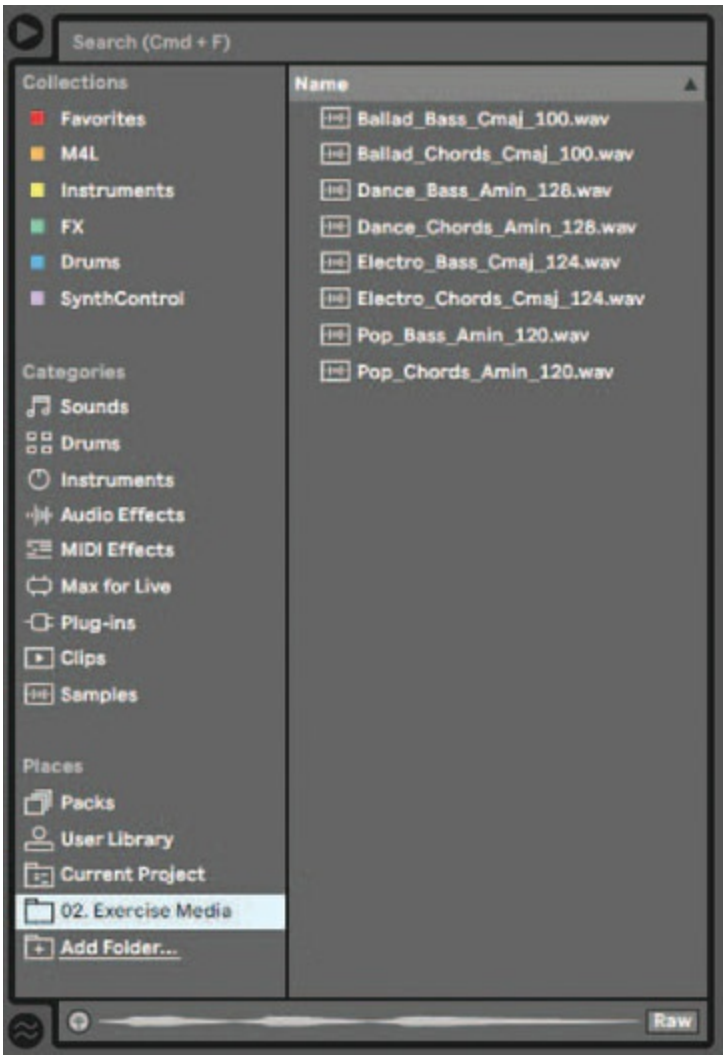


Figure 5.12 The Exercise Media folder after being added to the Live browser.

The audio files in this folder are all in WAV format. Notice that the file names include a loose genre identifier, a descriptor (Bass or Chords), a key (C major or A minor), and a tempo. In the next section, you will select a chord progression and the matching bass part to use for this exercise. For example, if you select the pop chord progression in C major at 120 BPM, you should also use the pop bass in C major at 120 BPM. This way, you will combine two distinct instruments that work together.

Preview the chord progression and bass audio files:

1. Click through the various audio files to listen to the different options for the Live Set. As discussed in Chapter 5, you can click an audio file to preview it (assuming the Preview button is enabled).

2. Make note of the chord progression and corresponding bass part that you like the most. In the next section, you will place the files on audio tracks and start assembling a simple musical pattern in your Live Set.

Importing Audio to Tracks

With the audio files for this exercise available through the Browser, importing them into audio tracks will be quick and easy. This is a good time to decide whether you want to work in the Arrangement View or Session View. If you prefer working in a traditional, timeline-based fashion, use the Arrangement View. If you'd like to use Live's unique features for auditioning different combinations of sounds and creating flexible arrangements, you can try the Session View instead.

Arrangement View (Option 1)

Complete this section if you'd like to work in the Arrangement View. Otherwise, skip forward to Option 2 below.

Import the audio files to tracks in the Arrangement View:

1. Set the Live Set's tempo so that it matches the tempo of the audio files you will be using.

2. Drag and drop the chord progression file you would like to use to Bar 5 of the audio track labeled Chords.

3. Rename the audio clip to something simpler, such as Chords.

4. Select the Chords clip and choose **EDIT > DUPLICATE** to create a second copy of the clip on the track, immediately after the original.

5. Repeat the command two more times, for a total of four identical clips, to create a longer, 16-bar song segment.

6. Drag and drop the corresponding bass part to the Bass track, this time starting at Bar 13.

7. Right-click on the top of the clip with the hand icon and choose **RENAME** from the pop-up menu.

8. Rename the clip to something simpler, such as Bass.

9. Duplicate the Bass clip once, so it plays from the middle to the end of the 16-bar segment.

You've now created a 16-bar segment, with a chord progression spanning the whole time and the bass part starting after 8 bars.

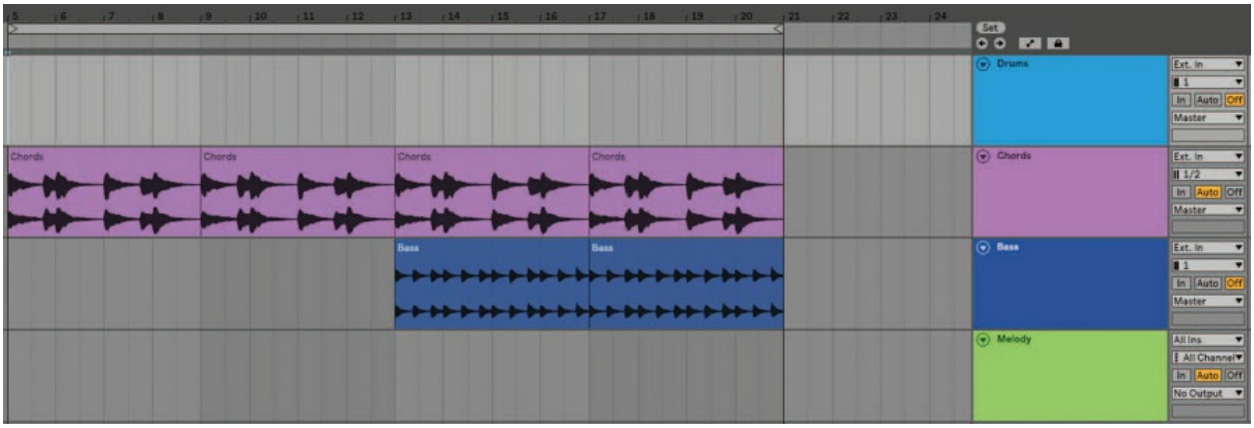


Figure 5.13 A Set with chord and bass parts completed in the Arrangement View

Review your work:

1. Position the Arrangement Insert Marker at the start of the clips.
2. Press the **SPACEBAR** to play back the Live Set and confirm your results.
3. Press the **SPACEBAR** a second time when finished.

Session View (Option 2)

Complete this section if you'd like to work in the Session View. (If you completed Option 1 above, skip forward to the next section.)

To import the audio files to tracks in the Session View:

1. Set the Live Set's tempo so that it matches the tempo of the audio files you will be using.
2. Drag and drop the chord progression file you would like to use onto the first and second clip slots of the audio track labeled Chords. This will place the clip on both Scene 1 and Scene 2.



Instead of dragging the audio file from the Browser twice, you can press and hold OPTION (Mac) or CTRL (Windows) while a dragging the clip from clip slot 1 to duplicate it on clip slot 2.

3. Right-click on the clip on each clip slot and choose **RENAME** from the pop-up menu.
4. Rename the clip to something simpler, such as Chords.
5. Drag and drop the bass file onto the second clip slot of the Bass audio track, so the clip is only on Scene 2.
6. Right-click the clip and rename it to something simpler, such as Bass.

At this point, you will have a basic, two-scene Live Set. You can alternate playback between the chord progression in isolation (Scene 1) and the combination of the chord progression and the corresponding bass part (Scene 2).

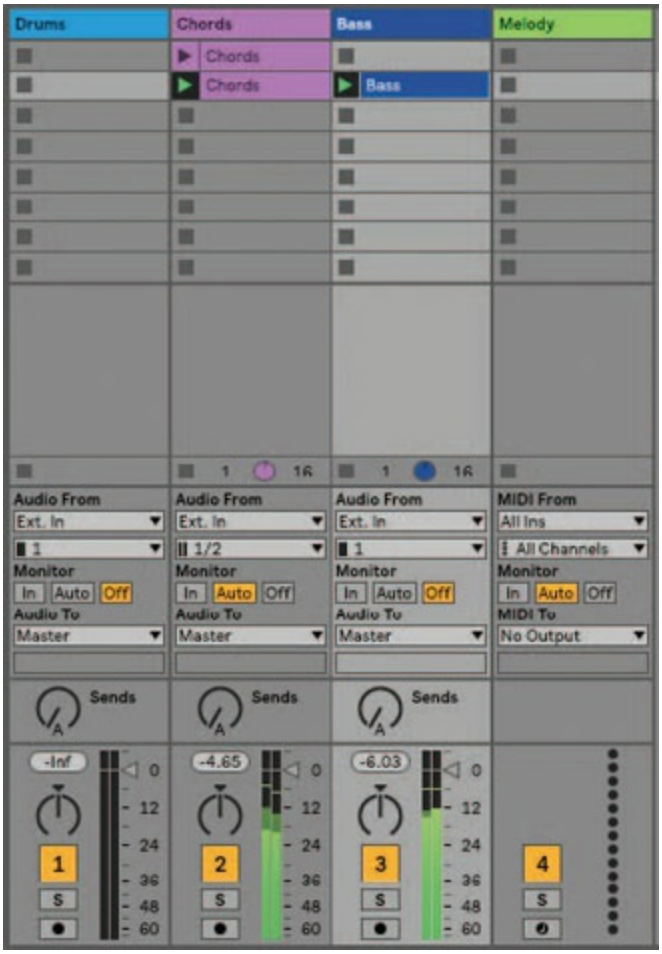


Figure 5.14 A Set with chord and bass parts added in the Session View

Review and your work:

1. Click the **SCENE LAUNCH** button on Scene 1 of the Master track. The chord clip will launch and begin playing. Left unchanged, the clip will repeat when it reaches the end.

2. When the clip is just a couple beats away from repeating, press the **SCENE LAUNCH** button on Scene 2 to queue up the next scene. When the first Scene ends, the second Scene will launch, containing both the chord progression and the corresponding bass part.

3. When finished, press the **SPACEBAR** or the **STOP ALL CLIPS** button on the Master track to stop playback.

Finishing Up

To complete this exercise, you will save your work and close the Live Set. You will be re-using this set in the next exercise, so it is important to save the work you've done.

Save your work and close the Live Set:

1. Choose **FILE > SAVE LIVE SET**.

2. Press **COMMAND+Q** (Mac) or **CTRL+Q** (Windows) to quit Live and close the project.

Chapter 6

Working with MIDI

This lesson covers the basics of recording, programming, and editing MIDI data in Ableton Live. It describes what MIDI is, how to record or manually create a MIDI performance, how to send MIDI data to a virtual instrument, and how to use many of the editing features available in Live's MIDI Note Editor.

Learning Targets for This Chapter

- Understand the basics of MIDI
- Recognize the different behaviors of MIDI clips and audio clips
- Set up a virtual instrument to generate sound using MIDI data
- Record a performance to a MIDI track using a MIDI controller
- Manually program a MIDI performance using a mouse and keyboard
- Edit and fine-tune MIDI data using the MIDI Note Editor

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform. Sign up for free at ElementsED.com.

MIDI data offers music producers, sound designers, and audio engineers a tremendous amount of flexibility when carrying out their work. MIDI provides a simple and efficient way to program or record musical performances, which can then be sent to various software– or hardware–based instruments—such as synthesizers, samplers, and drum machines—to produce almost any type of sound. This chapter introduces Live's MIDI recording, programming, and editing features, as well as other general considerations to keep in mind whenever you are working with MIDI data.

MIDI Basics

MIDI, or *Musical Instrument Digital Interface*, is a protocol for connecting electronic instruments, performance controllers, and computers so they can communicate with one another. MIDI data is fundamentally different from data stored in an audio file in that MIDI data does not represent sound waves or produce any sound on its own; instead, it represents information about a performance, such as the pitch, duration, and intensity of the notes.

A *MIDI sequencer* allows you to create, edit, and play back MIDI information that can be used to control MIDI-compatible devices, such as synthesizers, samplers, drum machines, and sound modules. In modern-day music production, these devices are often virtual instrument plug-ins (i.e., software-based instruments). Virtual instruments can be added directly to MIDI tracks within Live.

MIDI data is transmitted via MIDI messages. These messages are composed of 8-bit numbers (or *bytes*) and include information such as the *note* or *pitch number* (indicating an individual note in a scale) and the *velocity* value (typically affecting an individual note's volume or intensity).

MIDI data can be transmitted in multiple ways, such as by routing to virtual instrument plug-ins within Live or by connecting to hardware MIDI devices via MIDI cable. Up to 16 separate channels of MIDI information can be sent over a single MIDI cable, allowing a single cable path to control multiple MIDI devices or to control multiple sound sources within a single device that

is capable of multi-channel (or *multi-timbral*) operation.

Many other kinds of information can be conveyed via MIDI messages, such as pan and general MIDI volume information for instruments that support these, as well as program change events, or commands that tell MIDI instruments which of their available sounds, or *patches*, to use.

The Format of a MIDI Message

The most significant bit (first digit) in a MIDI message byte is reserved to distinguish between status bytes and data bytes. The remaining seven bits represent the unique data of the message byte, encompassing a range of values from 0 to 127.

The maximum length for a standard MIDI message is three bytes, consisting of one status byte and one or more data bytes. The format of each byte is as follows:

Status Byte	**Data Byte 1**	**Data Byte 2**
lttt nnnn	**Oxxxxxxx**	**Oxxxxxxx**

Where:

The leading 0 signifies a data byte
The leading 1 signifies a status byte
t is used to specify the type of status message being sent
n is used to specify the associated MIDI Channel Number
x is used to specify the associated data value, such as a note number (pitch) or velocity value

MIDI in Ableton Live

Ableton Live includes an integrated MIDI sequencer (or editor) that lets you program, record, import, and edit MIDI in much the same way that you work

with imported audio (see Chapter 5) or recorded audio (see Chapter 7). MIDI data is stored in MIDI clips and can be edited in the Clip View regardless of whether you're working in the Session View or Arrangement View.

Press Shift+Tab to toggle between the Clip View and Device View.

The Clip View is where you will find Live's visual MIDI Note Editor, which you can use to view, write, and edit MIDI data.

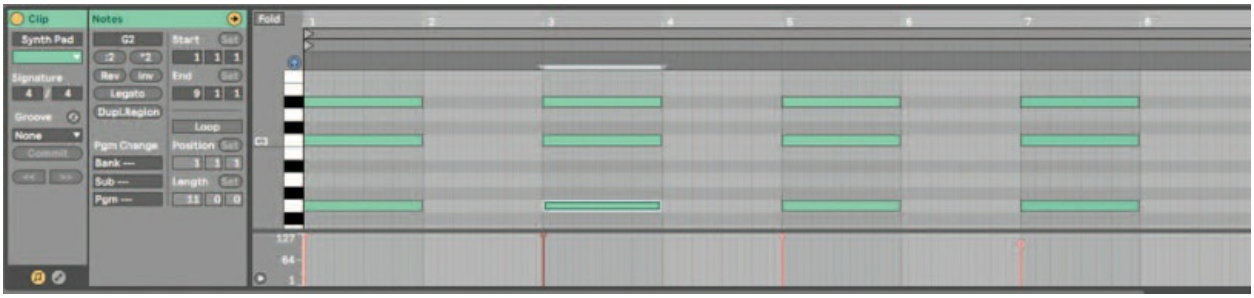


Figure 6.1 Live's MIDI Note Editor with a MIDI clip selected

As you may recall from Chapter 4, MIDI tracks can co-exist with audio tracks in a Live Set; clips on both track types reference the same timeline. Similarly, both track types feature similar mixer controls, such as volume, pan, input and output routing options, and more.

Creating MIDI Tracks

Before you can start working with MIDI, you'll need at least one MIDI track in your Live Set. To add a MIDI track, select **Create > Insert MIDI Track** or press **Command+Shift+T** (Mac) or **Ctrl+Shift+T** (Windows).

You can create MIDI clips by recording a performance to a MIDI track or manually programming in the MIDI notes. Both methods are covered later in this chapter.

For a detailed breakdown of track types and associated keyboard shortcuts, see Chapter 4 of this book.

MIDI Versus Audio Clip Operation

Audio and MIDI clips can and often do co-exist in a Live Set. However, they're very different, and it is worth understanding the differences in how audio and MIDI clips behave. We've already established one key difference in that MIDI clips don't inherently produce sound; the other key difference is how MIDI clips behave when you make changes to the tempo of your Live Set. Audio and MIDI clips respond differently to changes in tempo because their source material is different: audio clips contain *samples* whereas MIDI clips contain *performancedata*.

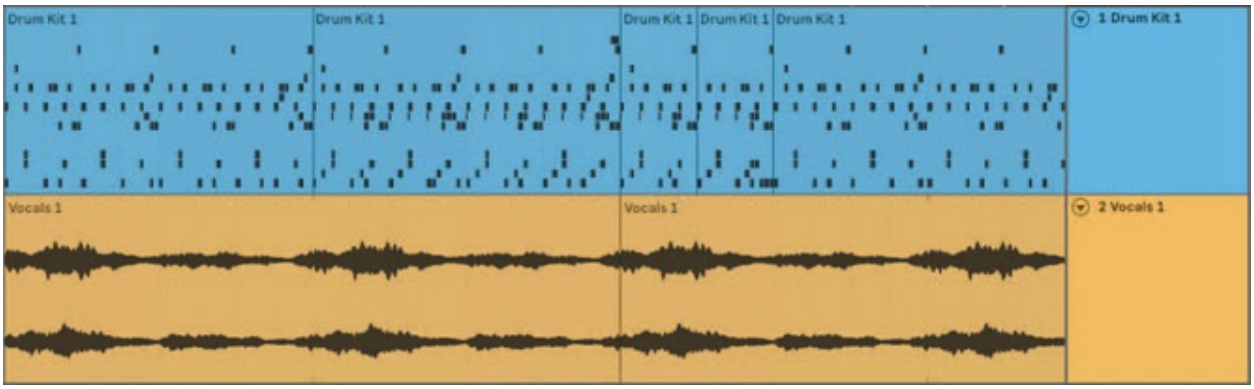


Figure 6.2 A blend of audio and MIDI clips in the Arrangement View Track Display

MIDI Clip Operation

MIDI clips easily conform to changes in your Live Set tempo, as the MIDI data is flexible. MIDI events are recorded relative to particular bar and beat locations (such as Bar 16, Beat 1), and their locations in time adjust based on the project tempo: if the tempo increases, the MIDI data will play back faster, and individual events will occur earlier in time; if the tempo decreases, the MIDI data will play back more slowly, and the same events will occur later in time.



Figure 6.3 A MIDI clip in the Arrangement View Track Display

Audio Clip Operation

Audio clips, on the other hand, are generally absolute, meaning they will continue to play back at their original speed and location after changing the project tempo—though the audio clips' positions relative to the Set's bars and beats will change. This behavior can be desirable for non-rhythmic audio, such as spoken words or ambient textures, but it can cause rhythmic audio like drum loops to fall out of sync in the context of a song.



Figure 6.4 An audio clip in the Arrangement View Track Display

Live's Warping feature allows audio clips to automatically speed up or slow down to conform to the Live Set's tempo.

Audio Warping

With Warp algorithms, Live is capable of time-warping audio clips while streaming them from disk so as to synchronize them with the Live Set's tempo. This happens without affecting the pitch, which can be changed independently. As such, even if you mix and match audio files from various sources with MIDI data, your Live Set can remain flexible with respect to tempo changes.

To enable time warping-audio, select an audio clip (or clips) and click the **WARP** button in the Sample box within the Clip View (see Figure 6.5). The selected audio clips will stretch or compress as needed to automatical ly follow tempo changes and conform to the Set's tempo.

Live provides a variety of Warp modes, as illustrated in Figure 6.5. The different modes change the way warping is applied, allowing you to select the best option for different program material.

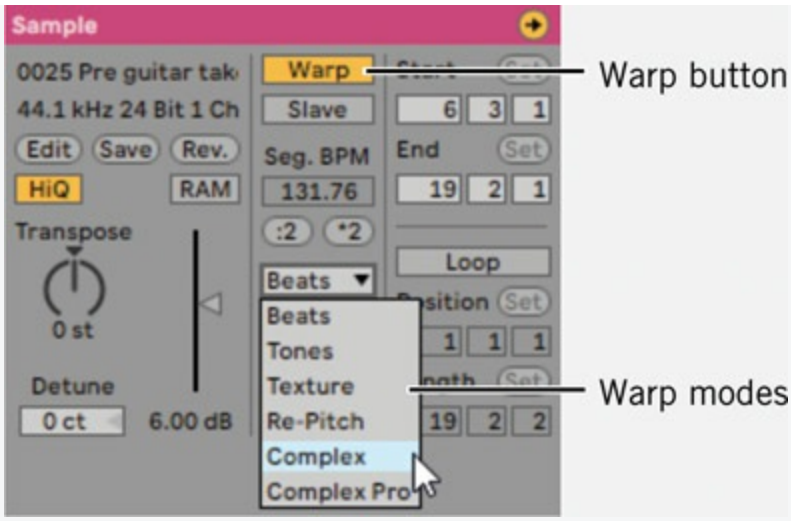


Figure 6.5 The Warp but ton and available Warp modes for a selected audio clip

Warping audio, especially when done to extremes, can create undesired audio artifacts. Always check the results and use the Edit > Undo command if you are not happy with the outcome.

Setting the Tempo and Time Signature

Before you begin recording MIDI data, you should determine the required tempo and time signature for the performance. The following sections describe how to configure these options for your composition.

Setting the Base Tempo

When you create a new Live Set, the tempo defaults to 120 beats per minute (BPM). If you need to work at a different tempo—and especially if you are planning to record a performance with the metronome—make sure to set the tempo in Live accordingly.

Entering a Tempo Value

If you know the tempo that you wish to use, you can enter it directly into the Control Bar.

To configure the Live Set's tempo directly, do the following:

1. Click the **TEMPO** field in the Control Bar.

2. Drag up andb down to adjust the value, or type a value in BPM. This will change the tempo for the entire Live Set in real time.



Figure 6.6 The Tempo field in the Control Bar (highlighted in yellow)

Tap Tempo

If you have a tempo in mind but you're not sure of the exact value, click the **TAP** button in the Control Bar once per beat; the Live Set's tempo will follow your tapping. For example, to set the tempo based on a performance you will be recording, have the musician(s) begin playing through the piece while you tap along using a mouse or trackpad with the cursor on the Tap button.



Figure 6.7 The Tap button in the Control Bar (highlighted in yellow)

You can use something other than a mouse or trackpad to tap a tempo. To do so, click on the KEY switch in the Control Bar to enable the Key Map mode. Click the TAP button and then press the target key to use for tapping. Press the KEY switch again to exit the Key Map mode and use the newly assigned key.

Automating Tempo Changes with Breakpoints

Your Live Set does not have to stick to one static tempo throughout its duration. You can use an automation envelope to change the tempo at any time in your Live Set.

Adding tempo changes with automation requires that you work in the Arrangement View.

To add tempo change breakpoints to your arrangement, do the following:

1. If necessary, switch to the Arrangement View.

2. Right-click the **TEMPO** field in the Control Bar and select **SHOW AUTOMATION**. This will show the Song Tempo automation envelope on the Master track.

   The automation envelope will display as a dotted orange line on the Master track in Arrangement View.

The automation envelop in Ableton Live is also sometimes referred to as the automation graph.

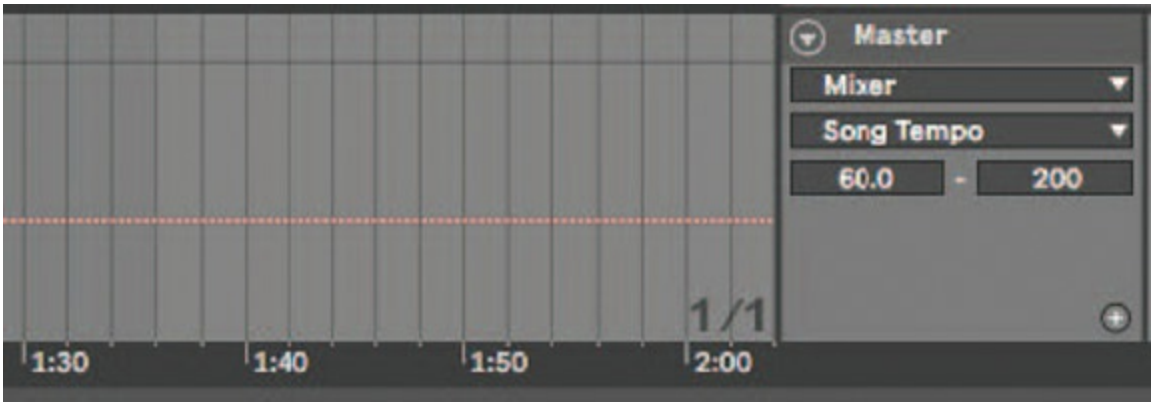


Figure 6.8 The automation envelope on the Master track before adding breakpoints

3. Disable the **DRAW MODE** switch in the Control Bar, if necessary.

4. Click anywhere on the dotted line to add a breakpoint at the existing tempo. A small circular node will display on the automation envelope, and the envelope graph will change to a solid orange line.

5. Click again on the line to add a second breakpoint.

6. Drag breakpoints up or down to set the desired tempo; drag breakpoints forward or backward to set the position for the tempo change and adjust the tempo ramp.

Tempo changes can either take effect instantly or ramp smoothly over time (see Figure 6.9).

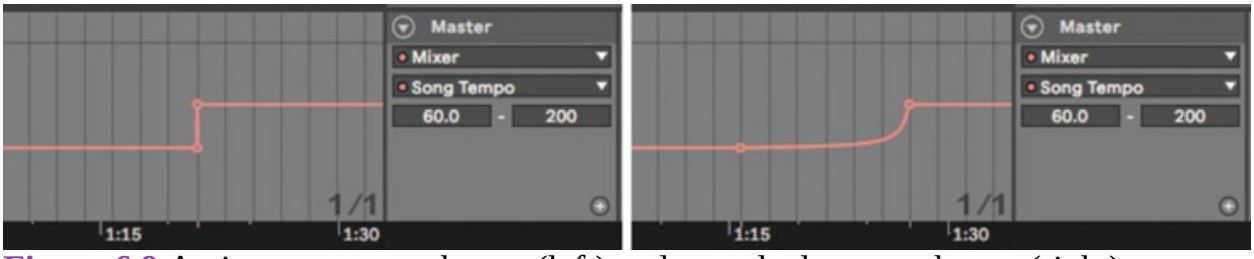


Figure 6.9 An instant tempo change (left) and a gradual tempo change (right)

Automation is covered in more detail in Chapter 10 of this book.

Setting the Base Time Signature

When you create a new Live Set, the time signature defaults to 4/4. If you intend to record and work in another time signature, you'll need to set it accordingly. You can either alter the time signature for the entire Live Set or have the time signature change within an arrangement.

To set the time signature for the entire Live Set, do the following:

1. Click the **Time Signature Numerator** in the Control Bar.
2. Drag up or down, or type a numeric value.
3. If necessary, click the **Time Signature Denominator** in the Control Bar.
4. Drag up or down, or type a numeric value.



Figure 6.10 The Time Signature field in the Control Bar

Adding Time Signature Changes

Your Live Set does not have to stick to one time signature for the entire

piece. You can change the meter at any point in your Live Set using time signature markers.

Adding time signature changes requires that you work in the Arrangement View.

To add time signature markers to your arrangement, do the following:

1. If necessary, switch to the Arrangement View.
2. Place the Arrangement Insert Marker where you want a time signature change to occur.
3. Do one of the following:
   - Select **Create > Insert Time Signature Change**.
   - Right-click at the desired location in the scrub area of the beat-time ruler and select **Insert Time Signature Change** from the pop-up menu.

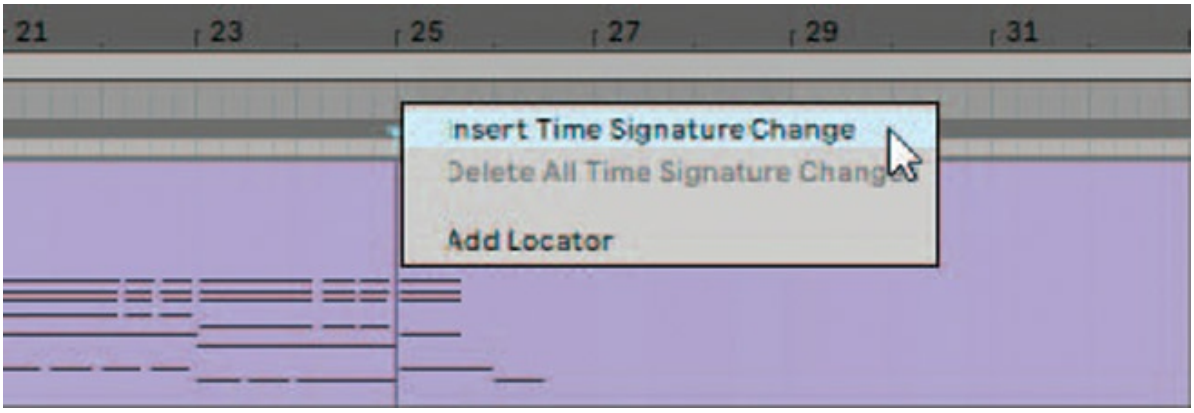


Figure 6.11 Right-click in the scrub area below the beat-time ruler to insert a time signature change.

4. Enter the desired time signature. For instance, you can type 4/4, 3/4, or some other valid time signature.
5. Press **Return or Enter**.

This will create a marker beneath the beat-time ruler where all time signature changes appear, as shown in Figure 6.12. A red dot will also appear on the **TIME SIGNATURE** field in the Control Bar, indicating that it has been automated (see Figure 6.13).

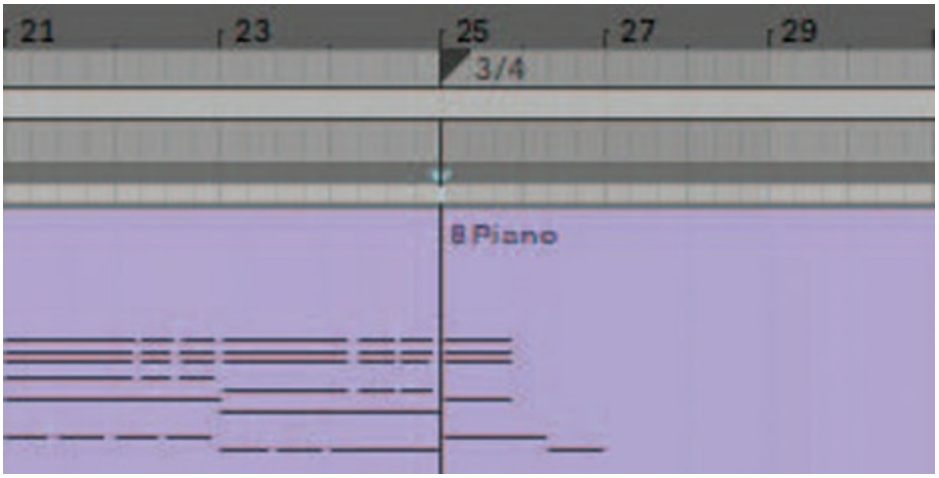


Figure 6.12 A time signature change marker in the Arrangement View



Figure 6.13 The Time Signature field with the automation indicator displayed (red dot)

You can click and drag time signature markers to move them along the timeline. Right-click a time signature marker and select **EDIT TIME SIGNATURE** to make changes to it or select **DELETE** to remove it.

Preparing to Record MIDI

With at least one MIDI track added to your Set and the tempo and time signature configured as desired, you will next need to prepare your Live Set for recording.

The general processes you will use to prepare for recording MIDI are as follows:

1. Configure the metronome to aid in making a well-timed recording, if

desired.

2. Configure your Record Quantization settings to automatically correct timing errors during the performance, if desired.

3. Connect a MIDI device, if available.

In the absence of a dedicated MIDI device, you can record MIDI using your computer keyboard or manually program MIDI using your mouse and keyboard.

4. Check the track inputs/outputs.

5. Record-enable the track(s).

Enabling the Metronome

When you're recording parts for a song or other composition that is based on a specific tempo (or tempos) with established bar and beat divisions, you might want to perform along with a metronome-based tick. The metronome may help you synchronize your performance to the composition when you're recording – whether it's MIDI data or audio – and ultimately align the recorded parts to musical divisions. Having your recorded parts in alignment with the Live Set's tempo makes it easy for you to make selections according to the musical grid, and perform such tasks as copying and pasting entire song sections as you work on your arrangements.

To enable or disable the metronome, click the **METRONOME** switch in the Control Bar along the top of the Ableton Live interface. When you press the **PLAY** button in the Control Bar, launch a clip from the Session View, or begin recording, the metronome will play a tick at each beat.

Customizing the Metronome

Ableton Live provides several metronome options, which you can access by clicking the menu triangle on the **METRONOME** switch.



Figure 6.14 The Metronome switch (yellow) and Metronome settings menu

Options available in the Metronome settings menu include the following:

- **Count-In**: determines the number of bars the metronome will tick before recording commences.
- **Sound**: allows you to choose between three styles of metronome ticks (Classic, Click, and Wood).
- **Rhythm**: determines the musical divisions that will be represented by the metronome.
- **Enable Only While Recording**: causes the metronome to sound during recording only – not during playback.

You can control the volume level of the metronome by adjusting the Preview/Cue Volume on the Master track (see Figure 6.15).

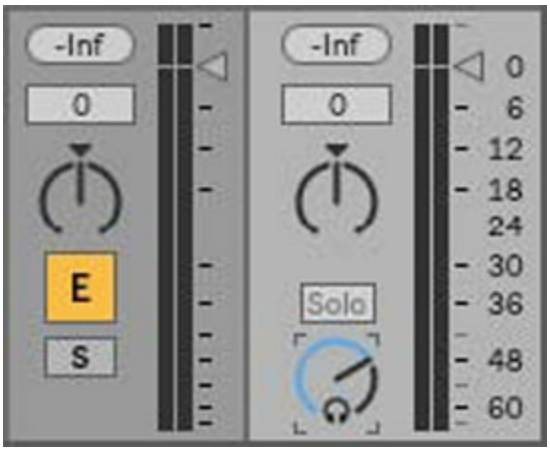


Figure 6.15 Use the Preview/Cue Volume knob (blue) on the Master track to adjust the metronome volume relative to the level of the Live Set

Configuring Record Quantization

You can set up Live so that it automatically aligns, or quantizes, MIDI notes during a recording.

To set up Record Quantization, do the following:

1. Select **EDIT > RECORD QUANTIZATION**.

2. From the list of options, choose the musical divisions to which you would like to align your notes. Larger divisions will result in a more noticeable quantization.

If you want to retain the original feel of the performance, do not use quantization during the recording. You can always quantize your performance after the recording. Quantization is covered in the last section of this chapter, "Creating and Editing MIDI Performances."

Connecting a MIDI Device

Recording MIDI data often involves connecting a keyboard, a drum machine, or another MIDI device as an input to your system. Before starting to record, you should verify that the MIDI device you will use for input (also called a *MIDI controller*) is connected to your system through an input on your MIDI interface or a USB port on your computer.

You can view connected MIDI devices in the **PREFERENCES** window (choose LIVE > PREFERENCES on Mac or OPTIONS > PREFERENCES on Windows) by selecting the **LINK/MIDI** tab (see Figure 6.16).

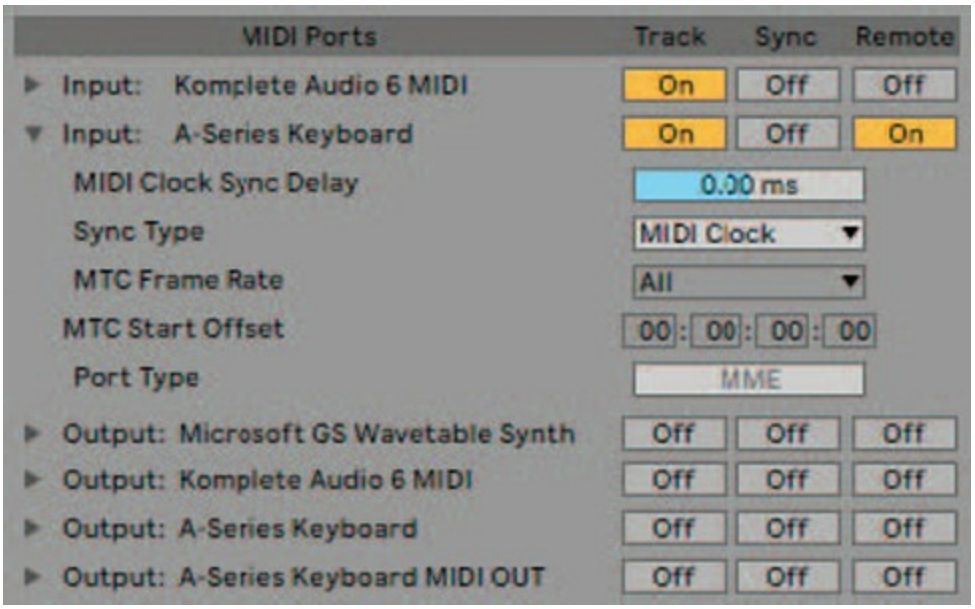


Figure 6.16 MIDI devices, as listed in the Live Preferences menu

Using the Computer Keyboard for MIDI

To use your computer keyboard as a MIDI device, press the M key or click the **COMPUTER MIDI KEYBOARD** switch in the Control Bar. This will make the computer keyboard available to function as a MIDI device, as described below.



Figure 6.17 The Computer MIDI Keyboard switch (yellow) in the Control Bar

When active, the Computer MIDI Keyboard uses the top and middle rows of alphabetic keys on the QWERTY keypad to play the black and white notes on a piano, as indicated in Figure 6.18. The octave targeted by the computer keys can be adjusted up or down using the Z and X keys.

You can adjust the velocity that will be used for the played notes using the C and V keys. Each key press will shift the velocity value up or down in 20 unit intervals.



Figure 6.18 MIDI mapping for the Computer MIDI Keyboard

Selecting MIDI Inputs

Once your MIDI device is connected, you will need to route its signal into the input of your MIDI track.

Before moving forward, make sure that the In/Out controls are visible by clicking **View > In/Out** or COMMAND+OPTION+I (Mac) or **Ctrl+Alt+I** (Windows). This will reveal input and output options for all the tracks in your Live Set in both the Session View and Arrangement View.

Input Type and Input Channel

Each MIDI track has an *Input Type chooser* and an *Input Channel chooser* at the top of the track's I/O section. These input options allow you to choose the MIDI device you would like to use with the track.

To select your MIDI device, do the following:

1. Locate the In/Out section of the track on which you will record.
2. Use the **Input Type** dropdown menu to select the MIDI controller you will use to record a MIDI performance.
3. Select an **Input Channel**. For now, All Channels will suffice.

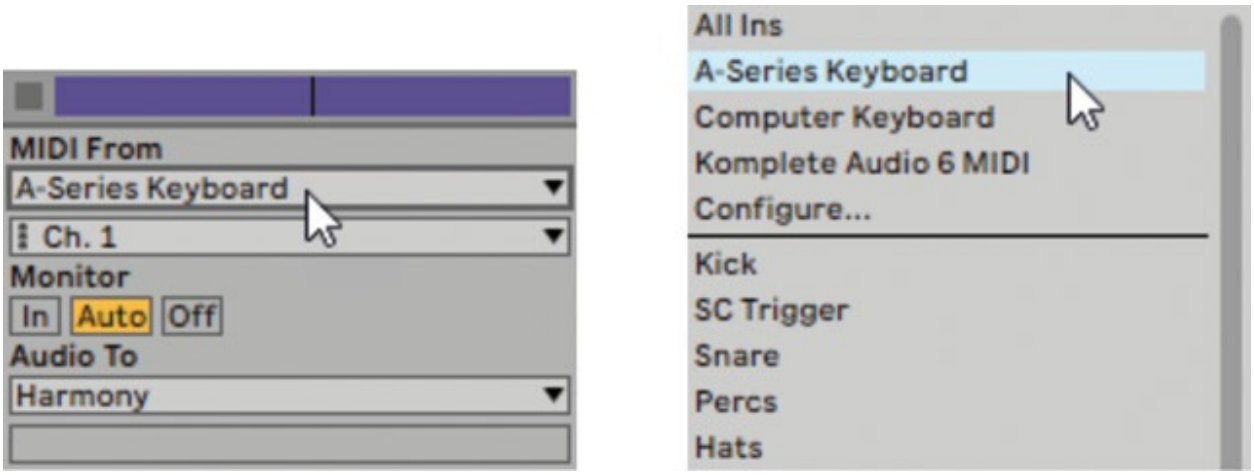


Figure 6.19 The MIDI In/Out options (left) and Input Type chooser menu (right)

Record-Arming MIDI Tracks

To enable recording on a MIDI track, click the track's **ARM** button in either the Session View or Arrangement View. The **ARM** button turns red when the track is record-ready.

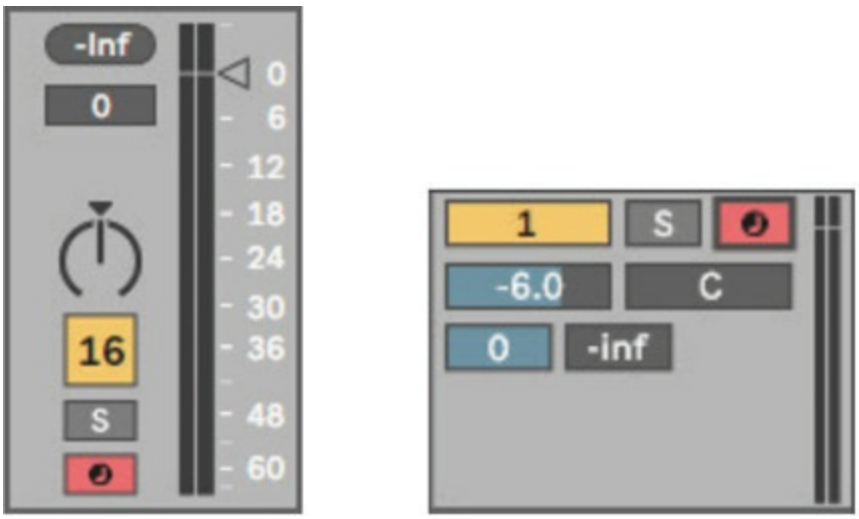


Figure 6.20 The track Arm button in the Session View (left) and Arrangement View (right)

Using MIDI Arrangement Overdub Mode

In certain MIDI recording workflows, you may wish to enable MIDI Arrangement Overdub mode. This feature allows you to record additional MIDI notes into a clip without overwriting existing notes. This recording method can be useful for many purposes, such as adding layers to an

instrumental performance or recording layers of a drum kit in separate passes.

When the **MIDI Arrangement Overdub** button is active, you can record a MIDI performance while merging it in with existing MIDI data on the track, overdubbing the track. When this button is not active, recorded MIDI data will replace existing track material.

To enable MIDI Arrangement Overdub do one of the following:

- Select **Options > MIDI Arrangement Overdub**.
- Click the **MIDI Arrangement Overdub** button (plus sign) in the Control Bar.



Figure 6.21 The MIDI Arrangement Overdub button (yellow) in the Control Bar

Before starting a MIDI recording on a track that already contains MIDI data, be sure to verify the status of the MIDI Arrangement Overdub mode to ensure you get the intended results.

Using Virtual Instruments

While the MIDI track is now ready to record, at this stage it will not produce any sound. In order for the MIDI signal to generate sound, it needs to be sent to a *virtual instrument*. Virtual instruments are the software equivalents of outboard synthesizers, samplers, or sound modules. Ableton Live Suite includes a selection of native virtual instruments and also supports third-party virtual instruments as plug-ins.

Ableton Live Suite ships with an assortment of popular software-based virtual instruments, including Wavetable, Operator, and Sampler. Ableton Live Standard includes a smaller number of sample-based instruments. In order to experience MIDI recording and editing, you will need at least one software-based instrument.

Inserting a Virtual Instrument on a MIDI Track

You can add a native or third-party virtual instrument to a MIDI track at any time.

To add a virtual instrument to a MIDI track, do the following:

1. Show the Browser by choosing **View > Show Browser** or by pressing **Command+Option+B** (Mac) or **Ctrl+Alt+B** (Windows).

2. Do one of the following:

   - Select the **Instruments** category to view the available Ableton Live virtual instruments.

   - Select the Plug-Ins category to view any third-party plug-ins you've installed (including virtual instruments).

3. Expand the instrument as needed to navigate the available contents, using the disclosure triangles in the Browser content pane.

4. With the destination MIDI track selected, double-click the virtual instrument device you want to insert. Your MIDI signal will now be routed to the virtual instrument.

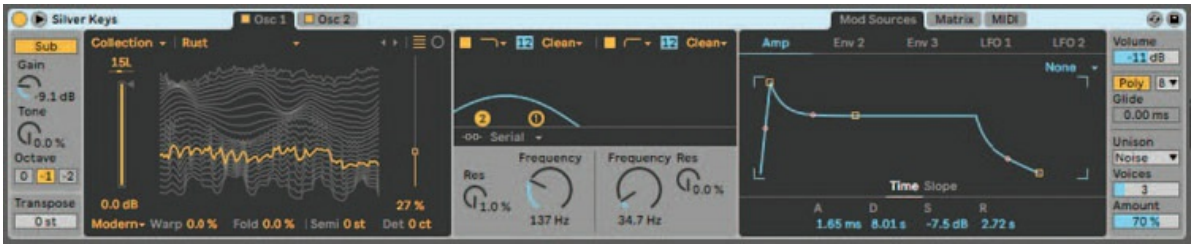


Figure 6.22 The Wavetable virtual instrument device on a MIDI track

5. Set the track's **Volume** slider to the desired output level.

6. Record-enable the track and play some notes on your MIDI controller. The meters on the MIDI track will register the instrument's audio output.

Deactivating Virtual Instruments

You can deactivate a virtual instrument without removing it from a track. Deactivating the instrument will free up the CPU resources that the instrument was using, while retaining the instrument settings.

To deactivate a virtual instrument:

- Click the round Device Activator in the upper left-hand corner of the device.

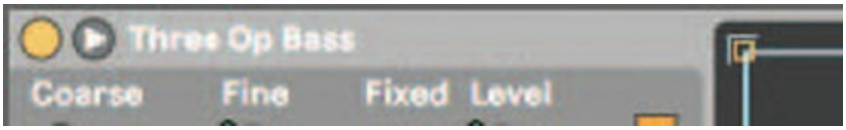


Figure 6.23 The Device Activator (yellow) on a virtual instrument

Using Drum Rack

The Drum Rack is a simple but powerful drum sampler that is bundled with Ableton Live Standard and Suite editions. It has a grid-based layout that is reminiscent of many drum machines, allowing you to create your own drum kits and trigger the drum sounds with MIDI signals. Each cell in the Drum Rack corresponds to a single a MIDI note, such as C1, C#1, D1, and so on. When you drag and drop an audio file to a cell, the MIDI note assigned to it will trigger playback of the sound, making the drum kit playable by using a MIDI device or by manually programming MIDI notes.

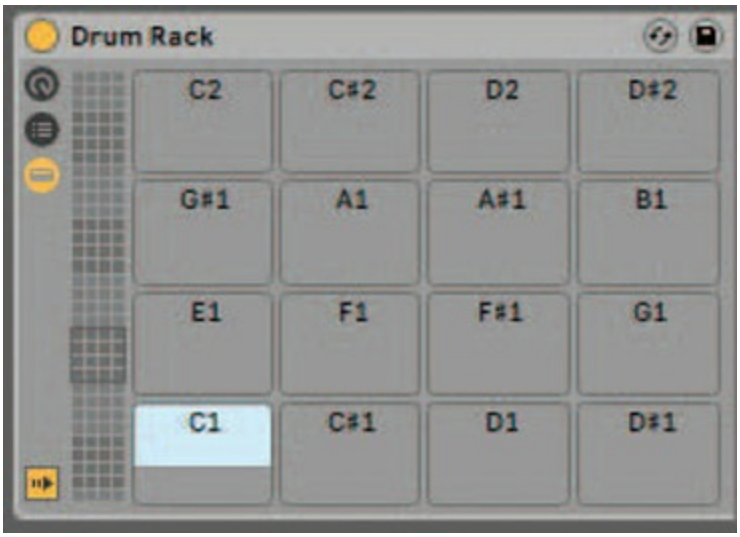


Figure 6.24 The grid-based layout of the Drum Rack instrument

The flexibility of this instrument comes from the amount of control you have over each sample you load into the Drum Rack. For instance, you can configure basic mix controls, such as volume and pan, as well as filter, LFO (or low-frequency oscillator), and sample length parameters for each individual sample.



Figure 6.25 A Drum Rack preset with drum samples that can be triggered with MIDI data

If you are not feeling ready to construct your own kit using the Drum Rack, you can browse pre-created kits in the Browser. Click the Drums category in the browser to audition available drum kits. Double-click a drum kit that you'd like to add to a MIDI track.

Using Simpler

Simpler is an instrument that allows you to trigger any piece of recorded audio—instrument sounds, vocal recordings, drum and synth one-shots, and more—using MIDI data. In order to use this sample-based instrument, you must have an audio file to load into the Simpler.

Simpler is a useful tool for making playable instruments out of any source material, as well as performing realistic recreations of acoustic instruments through MIDI. Sampling is a deep and complex art, so to start to understand how it works, you might want to explore some of Ableton's Simpler presets.



Figure 6.26 A Simpler-based preset called “BBass” loaded on a MIDI track

To browse Simpler presets, click the disclosure triangle next to Simpler in the Browser. You can explore different categories of sounds and audition them by clicking their name. Once you have found one you like, double-click it to insert it on the currently selected MIDI track. Be aware that this will overwrite any existing instrument device for the track.

Using Wavetable (Ableton Live 10 Suite only)

Wavetable is a wavetable synthesizer that was introduced with Ableton Live 10 Suite. It is particularly useful for creating dynamic, morphing sounds through wavetable modulation. However, synthesis is also a complex topic, so at this time you might consider browsing for a preset, or a pre-created sound, rather than creating your own from scratch in Wavetable.

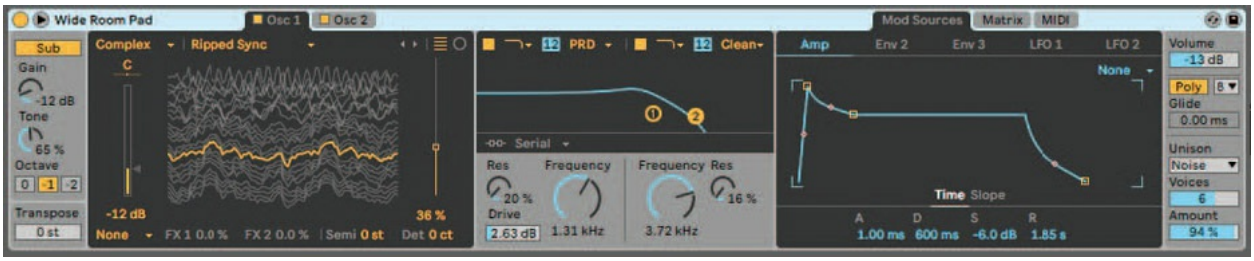


Figure 6.27 A Wavetable preset called “Wide Room Pad” loaded on a MIDI track

To browse Wavetable presets, click the disclosure triangle next to Wavetable in the Browser. You can explore different categories of sounds and audition them by clicking their names. Once you have found one you like, double-click it to insert it on the currently selected MIDI track. Be aware that this will overwrite any existing instrument device on the track.

For more information on navigating the Browser and using its



controls, see Chapter 5 in this book.

Recording a MIDI Performance

With a MIDI controller selected, a MIDI signal feeding a virtual instrument device, and the track armed for recording, you are ready to begin recording.

Session View

Recording in Session View allows you to create individual clips and launch other clips while recording.

To begin recording MIDI in Session View, do the following:

1. Record-arm at least one MIDI track as described earlier in this chapter. Once the track is armed, each clip slot in the track will display a round Record button in place of its square Clip Stop button.

2. Click the **RECORD** button on one of the clip slots to record a new clip to that slot.

3. The record button will flash red and the metronome count-in will begin, if enabled.

4. Once recording has commenced, the record button will turn into a red Clip Launch button.

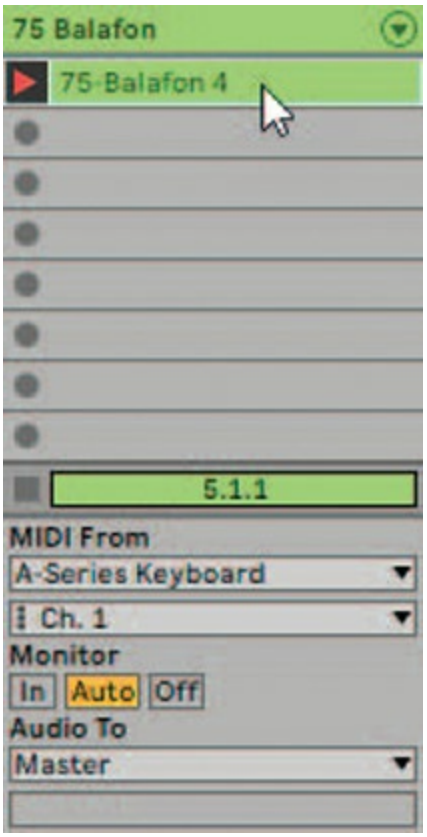


Figure 6.28 Recording MIDI to a clip slot in the Session View

5. When you have finished your record take, click the **STOP** button in the Control Bar (or press the SPACEBAR).

Alternatively, you can press the red Clip Launch button at the end of a record pass to continue playback of the current scene, including the new MIDI recording.

Recording to Multiple Tracks

If you have multiple tracks armed for recording, you can click the **SESSION RECORD** button(hollow circle) in the Control Bar to record onto all armed tracks on a selected scene.



Figure 6.29 The Session Record button (highlighted) in the Control Bar

Customizing Session View Recording

When recording in the Session View, you can optionally select a value from the Quantization Menu. When the value is set to anything other than None, Live will automatically cut the boundaries of your recorded clip so it aligns with your Live Set. This will aid in keeping clips in rhythmic alignment.



Figure 6.30 The Quantization Menu with a value of 2 Bars selected

The Session View can also be configured so that recording begins on all armed tracks upon launching a given scene. To do this, you need to enable the Start Recording on Scene Launch option in the Preferences window. This option is located under the **RECORD/WARP/LAUNCH** tab.

Arrangement View

The Arrangement View gives you a linear, timeline-based view of your recording progress. It is best suited for recording to multiple tracks simultaneously.

To begin recording MIDI in Arrangement View, do the following:

1. Arm at least one MIDI track for recording.
2. Position the Arrangement Insert Marker where you want the recording to begin.
3. Click the **ARRANGEMENT RECORD** button (solid circle) in the Control Bar.



Figure 6.31 The Arrangement Record button (red) in the Control Bar

4. The Arrangement Record button will turn red and recording will begin immediately, starting with the metronome count-in (if enabled). The

track(s) to which you are recording will be highlighted red during the recording.

5. When you have finished your record take, click the **STOP** button in the Control Bar (or press the **SPACEBAR**).

Customizing the Arrangement Record Button

You can change the behavior of the Arrangement Record button by toggling the **START PLAYBACK WITH RECORD** option in the Ableton Live Preferences window. This option is located under the **RECORD/WARP/LAUNCH** tab. When disabled, recording does not start until the Arrangement Record button is triggered **and** the **PLAY** button (or the SPACEBAR) is pressed.



You can use function key F9 as a shortcut to activate the Arrangement Record button.



You can also hold Shift while activating the Arrangement Record button to override the current Start Playback with Record setting, saving you from having to open the Preferences window and manually switch modes.

Creating a MIDI Performance Manually

Sometimes, you might want or need to create a MIDI performance manually (by programming notes using the mouse and keyboard). This requires that you create a MIDI clip and draw notes in the MIDI Note Editor.

First, you'll need to create an empty MIDI clip; there are multiple ways of doing this in both Session View and Arrangement View.

To create a MIDI clip in Session View:

- Double-click an empty clip slot on a MIDI track.
- Select an empty clip slot on a MIDI track and choose **CREATE** > **INSERT MIDI CLIP** (S).

To create a MIDI clip in Arrangement View:

- Double-click on a MIDI track in the Track Display window.
- Select an empty range of time on a MIDI track in the Track Display window and choose **CREATE > INSERT MIDI CLIP** (S), or press **COMMAND+SHIFT+M** (Mac) or **CTRL+SHIFT+M** (Windows).

The next section covers the MIDI Note Editor in detail, including its features for creating and editing notes.

Creating and Editing MIDI Performances

Live's MIDI Note Editor allows you to both create new MIDI performances and edit existing ones. To access Live's MIDI Note Editor, select a MIDI clip (if Clip View is active) or double-click a MIDI clip (if Clip View is not active). This will display the clip contents in Clip View, within the MIDI Note Editor.

Click and drag the divider between the Clip View and Tr ack Display to expand the MIDI Note Editor, if necessary.

The MIDI Note Editor

The MIDI Note Editor shows individual MIDI notes in a piano-roll format, with note pitch shown on the vertical axis, and note position and duration in time shown on the horizontal axis. MIDI notes are represented as colored bars of different brightness, depending on their respective velocities. A dedicated velocity editor is located along the bottom of the MIDI Note Editor, with the current grid spacing value indicated in the lower right-hand corner (see Figure 6.32).

If the velocity editor is not visible, click the triangle icon at the bottom of the note ruler area in the MIDI Note Editor (bottom left corner).

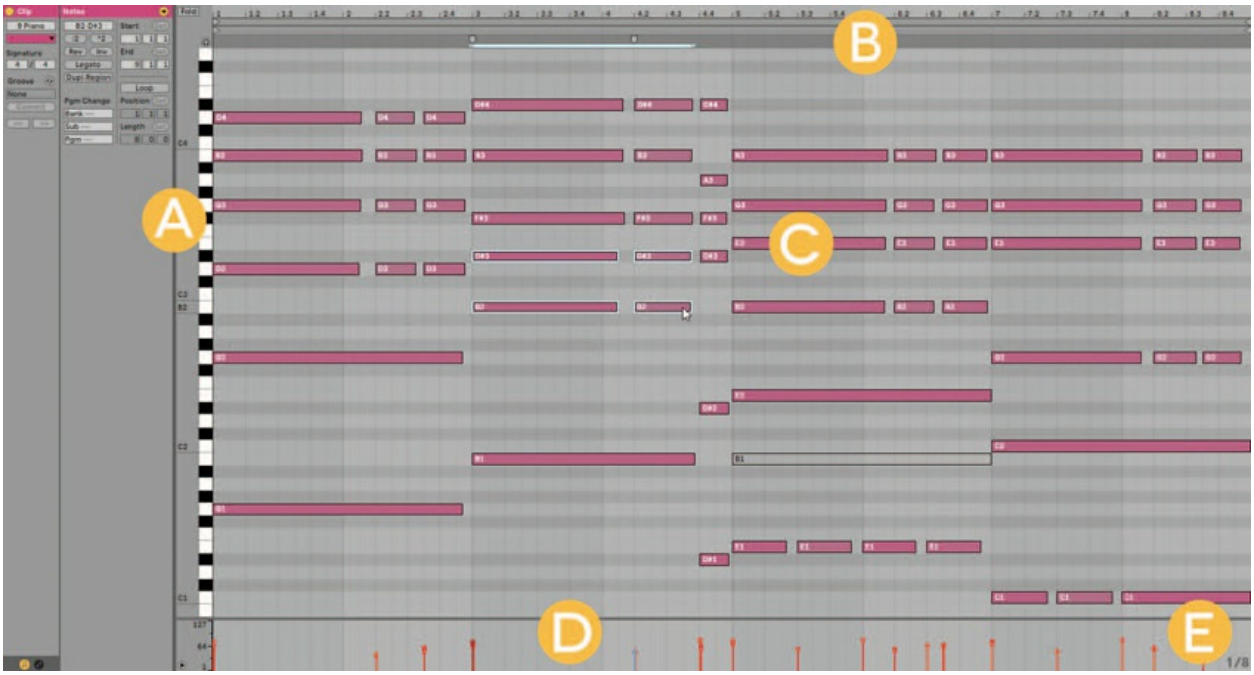


Figure 6.32 The MIDI Note Editor: (A) note ruler (B) time ruler (C) MIDI notes (D) velocity editor (E) grid spacing

The visual nature of the MIDI Note Editor makes it easy to see musical performances as well as create, select, and edit notes. This section covers editing functions that will allow you to create the exact performance you desire.

Creating MIDI Notes

During the production process, you may want to enter MIDI notes by hand. For many producers, this is a faster way of entering MIDI data than playing a keyboard or other controller.

There are two methods of creating MIDI notes by hand:

- Double-click in the MIDI Note Editor at the desired note and position in time. Before releasing the mouse, drag to the right to extend the duration of the note. (You can also trim note durations after adding them by dragging on a note edge with the trim icon.)

- Press the B key or the **Draw Mode** button in the Control Bar to enable Draw Mode. This enables the pencil tool, allowing you to “draw” MIDI

notes in the editor. Note lengths will match the current grid size of the MIDI Note Editor.



Figure 6.33 The Draw Mode button (yellow) in the Control Bar

When adding notes using the above methods, note start and end points will snap to the current grid lines. To change the grid size, right-click in the MIDI Note Editor and choose a different Fixed Grid setting. You can also press and hold **COMMAND** (Mac) or **ALT** (Windows) to remove grid constraints when adding new notes.

With the cursor in the MIDI Note Editor, press COMMAND+1 and COMMAND+2 (Mac) or CTRL+1 and CTRL+2 (Windows) to narrow or widen the grid, respectively.

Clicking on an existing note in Draw Mode (or double-clicking in Edit Mode) will delete the note.

Selecting MIDI Notes

Whether you have recorded or manually created MIDI notes, you can make selections for editing in multiple ways.

To make a selection of MIDI notes, do one of the following:

- Click an individual note in Edit Mode.
- Starting from an empty space, click and drag within the MIDI Note Editor in Edit Mode to draw a marquee around multiple notes.
- **SHIFT+CLICK** a note in Edit Mode to add a note to a selection or remove a note from a selection.
- Click and drag within the MIDI Note Editor in Edit Mode *without* selecting any notes to select a timespan, or click and drag within the

MIDI Note Editor Scratch Area in either mode to create a timespan. Then press **RETURN or ENTER** to select all notes that start within the timespan.

- With the cursor located in the MIDI Note Editor or any note selected, press **COMMAND+A (Mac) or CTRL+A** (Windows) to select all notes in the clip.

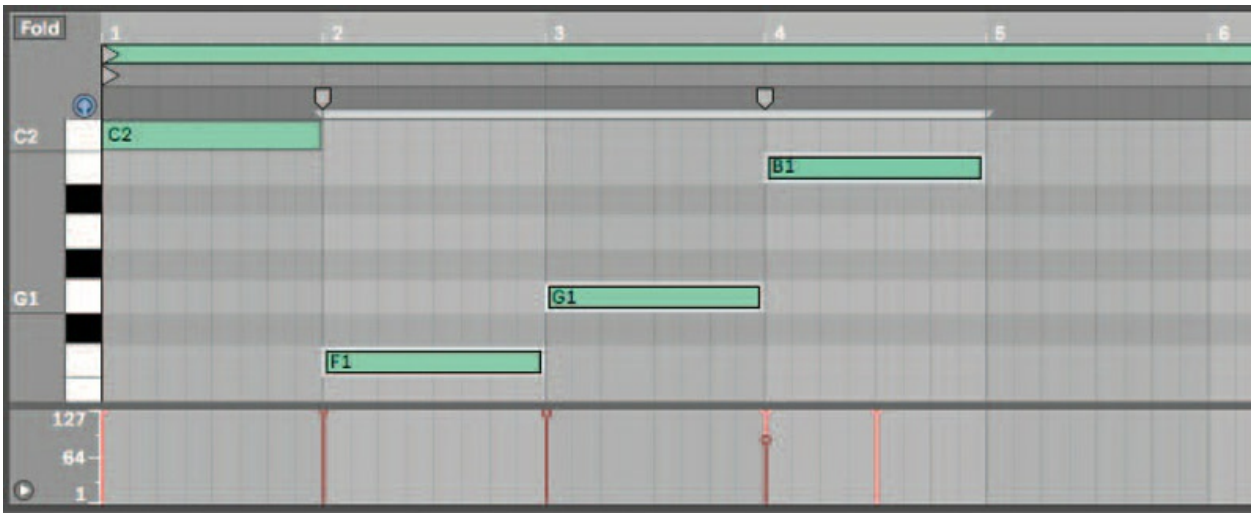


Figure 6.34 A multi-note selection in the MIDI Note Editor

Moving a Selection to a Different Note

Once you've made a selection of one or more MIDI notes, you can easily move the selection to a different note.

To move a selection from note to note using the keyboard, do one of the following:

- With at least one note selected, you can cycle between the previous and next note by pressing **OPTION+UP/DOWN ARROW** (Mac) or **CTRL+UP/DOWN ARROW** (Windows).

- With at least one note selected, you can cycle between the previous and next note *of the same pitch* by pressing **OPTION+LEFT/RIGHT ARROW** (Mac) or **CTRL+LEFT/RIGHT ARROW** (Windows).

Editing MIDI Notes

The MIDI Note Editor provides many ways to edit notes, allowing you to develop your own workflows. It is possible to edit MIDI notes using a mouse or trackpad, and also directly from the computer keyboard.

Editing MIDI Notes Using a Mouse or Trackpad

You can quickly change the pitch, position, and length of MIDI notes using the mouse.

To edit MIDI notes using a mouse or trackpad:

- Click and drag a note in Edit Mode to move it to another pitch and position in time.
- Click and drag the edges at the start or end of a note to trim its duration in either direction.
- Right-click and select **Deactivate Note(s)** to prevent a note from playing without deleting it.
- Right-click and select **Duplicate** to duplicate a note or a selection of notes.
- While holding **Option** (Mac) or **Ctrl** (Windows), click and drag a note or selection of notes to duplicate it.

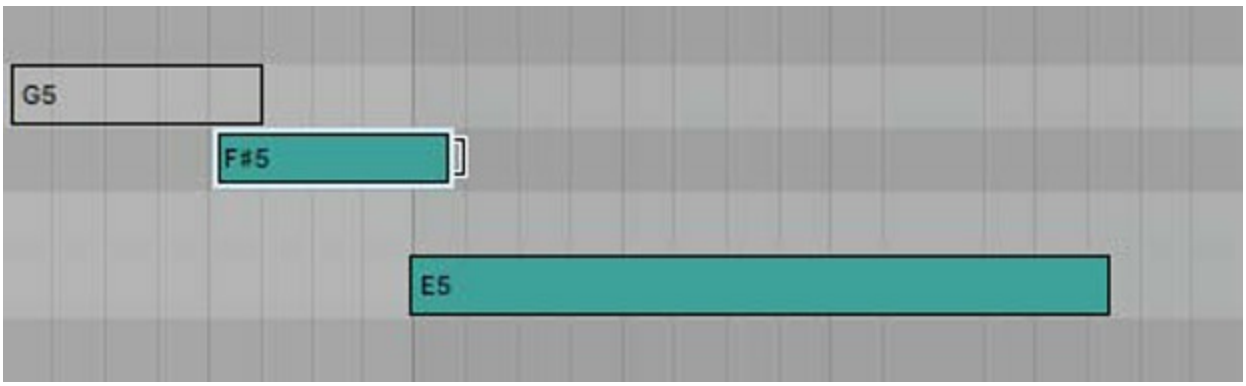


Figure 6.35 A deactivated MIDI note (G5) and a note being trimmed with the mouse (F#5)

Editing MIDI Notes with the Computer Keyboard

You can also perform some of the same editing tasks described above using the computer keyboard.

To edit MIDI notes using the computer keyboard:

- With a note selected, press any directional arrow to move it to adjacent pitches and positions in time.

Hold SHIFT while pressing the UP or DOWN ARROW to move a note or notes by an entire octave.

- To change the duration of a selected note (or notes), press **SHIFT+LEFT/RIGHT ARROW**.

- Press the 0 key to activate or deactivate a note; deactivating a note will prevent it from playing without deleting it.

- Press **COMMAND+D** (Mac) or **CTRL+D** (Windows) to duplicate a selected note or notes.

To toggle notes snapping to the grid, select OPTIONS > SNAP TO GRID or press COMMAND+4 (Mac) or CTRL+4 (Windows).

To suspend the grid and prevent snapping, hold COMMAND (Mac) or ALT (Windows) when adding, moving, or trimming a note.

Editing Velocity

Velocity is MIDI data that indicates how hard a key or pad was pressed when recording from a MIDI controller. This data is stored within the MIDI performance. Velocity information can help you create more lifelike MIDI performances, as different velocity values can trigger different audio samples and different volume levels from virtual instruments. Even if you don't

perform MIDI using a velocity-sensitive MIDI controller, you can manually edit velocity data for each note in a clip.

Velocity is represented both by the brightness of a note and by the velocity stalk at the bottom of the MIDI Note Editor. Higher velocity notes are brighter and more vibrant in color. Their velocity stalk in the velocity editor is also taller than lower-velocity notes.

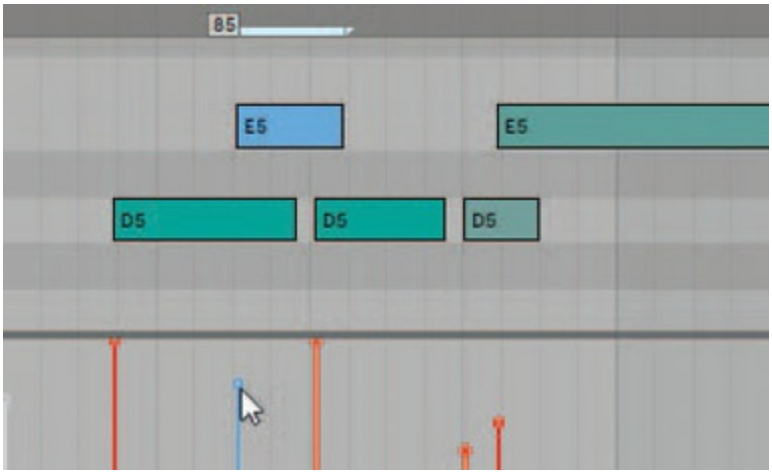


Figure 6.36 Editing velocity by clicking and dragging a velocity stalk

To quickly adjust the velocity for one or more notes:

1. Select one or more MIDI notes.
2. Do one of the following:
    - Click and drag a velocity stalk up and down in the velocity editor.
    - While holding **COMMAND** (Mac) or **ALT** (Windows), click and drag up and down on a MIDI note.

The velocity value will display at the top of the MIDI Note Editor window, using a range spanning from 1 to 127. When multiple notes are selected, the velocity of all notes will change relative to one another.

Additional Editing Features

Just to the left of the MIDI Note Editor display is the Notes section of the

Clip View. Several handy MIDI editing features are located here:

- ***Transpose***: Allows you to move all selected notes up and down by semitones by typing in a positive or negative value.
- ***Play at half or double tempo***: Halves or doubles the duration of all selected notes and the spacing between notes.
- ***Reverse Notes***: Reverses the order of a selection of notes.
- ***Invert Notes***: Inverts the pitch of a selection of notes, while retaining the intervals between notes.
- ***Legato***: Shortens or lengthens selected notes so that the end of each note reaches the start of the next note without overlapping.

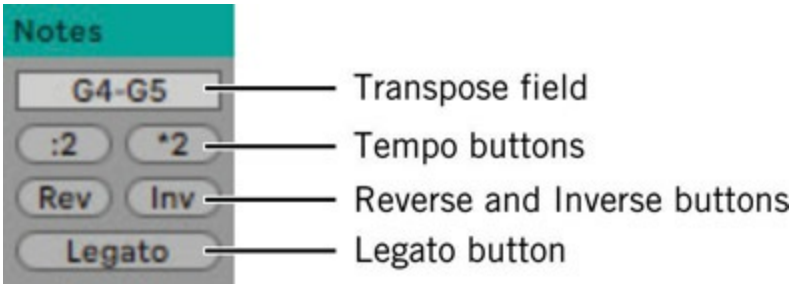


Figure 6.37 Additional MIDI editing features in the Notes section

MIDI Quantization

Quantization allows you to align, or quantize, MIDI notes to a specified timing grid. This creates a style of recording similar to working with a hardware sequencer or drum machine. While you might want that sort of strict timing for your performance in some cases, Live's quantization feature also allows you to improve the accuracy of your MIDI performance without sacrificing its natural human quality.

To quantize a MIDI note or multiple notes:

1. Select one or more notes using any of the methods described earlier in this chapter.

2. Select **EDIT** > **QUANTIZE SETTINGS**, or Right-click a selected note in the MIDI Note Editor and select **QUANTIZE SETTINGS**. The Quantize Settings dialog box will display.



Figure 6.38 The Quantize Settings dialog box

3. In the dialog box, select the desired quantization grid, whether you want to quantize the start or end of the notes (or both), and the intensity of the quantization.

4. Click OK to apply the change.

By using a quantization value of less than 100%, you can improve the timing of the performance without losing its human touch.

To skip the Quantization Settings dialog box and apply the previous quantization settings, select EDIT > QUANTIZE or press COMMAND+U (Mac) or CTRL+U (Windows).

Navigating the MIDI Note Editor

Depending on the complexity of your MIDI performance, it may be very useful to zoom the MIDI Note Editor to increase the visual resolution of the recorded notes. You can navigate and zoom the vertical and horizontal axes

independently, allowing you to isolate specific notes and make fine-tuning adjustments, or show all of the MIDI data in a clip for broad edits.

Scrolling and Zooming with Time Ruler (Horizontal Axis)

The horizontal axis, or time ruler, represents the position and duration of MIDI notes. There are multiple ways to zoom the time ruler for specific editing tasks.

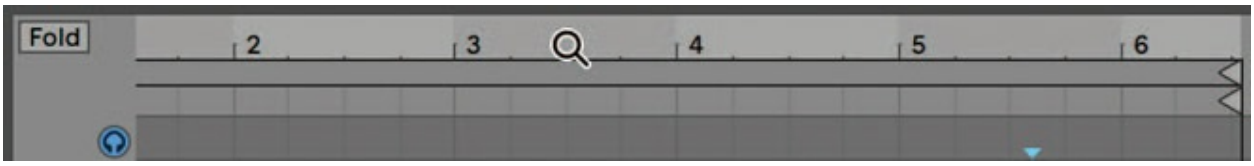


Figure 6.39 The cursor will display a magnifying glass when positioned over the time ruler

To zoom horizontally using the timer ruler:

- Click and drag up and down on the time ruler to zoom the MIDI Note Editor horizontally.
- Double-click the time ruler to show the contents of the entire selected MIDI clip.
- With one or more notes selected, double-click the time ruler to fit the selection horizontally to the MIDI Note Editor window.

With the cursor placed in the MIDI Note Editor (or with a note selected), you can also press the PLUS (+) and MINUS (–) keys on the keyboard to zoom horizontally.

Once you have zoomed in, you can click and drag left and right on the time ruler to scroll the MIDI Note Editor horizontally. You can also scroll left and right with the keyboard by pressing **COMMAND+PAGE UP/PAGE DOWN** (Mac) or **CTRL+PAGE UP/PAGE DOWN** (Windows).

Scrolling and Zooming with the Note Ruler (Vertical Axis)

The vertical axis, or note ruler, represents all the notes or pitches that you can send to a MIDI instrument. This display format is known as a piano roll.

Ableton Live provides multiple ways to zoom using the note ruler for specific editing tasks.



Figure 6.40 The cursor will display a magnifying glass when positioned over the note ruler

To zoom vertically using the note ruler:

- Click and drag left and right on the note ruler to zoom the MIDI Note Editor vertically.
- Double-click the note ruler to show the entire range of notes used in the selected clip.
- With multiple notes selected, double click the note ruler to fit the selection vertically to the MIDI Note Editor window.

When zooming in vertically on selected notes, the range will scroll to begin at the top of the MIDI Note Editor. If the range of notes does not fill at maximum vertical zoom, additional pitches will be displayed below the selected range.

Once you have zoomed in, you can click and drag up and down on the note ruler to scroll vertically through the octaves of available notes. You can also scroll up and down from the keyboard using the **PAGE UP and PAGE DOWN** keys.

Click the FOLD button in the upper left-hand corner of the MIDI Note Editor to hide any MIDI pitches that are unused in the selected clip.

Review/Discussion Questions

1. What does the term MIDI stand for? How is MIDI data different from the data stored in an audio file? (See “MIDI Basics” beginning on page 138.)

2. What are two key differences between the behavior of MIDI clips and audio clips? (See “MIDI Versus Audio Clip Operation” beginning on page 140.)

3. What is the default tempo in a Live Set? How would you go about changing the tempo? (See “Setting the Base Tempo” beginning on page 142.)

4. What is the default time signature in a Live Set? How would you go about changing the time signature in the middle of an arrangement? (See “Setting the Base Time Signature” beginning on page 144.)

5. How would you go about locating and placing a virtual instrument device on a MIDI track? (See “Using Virtual Instruments” beginning on page 151.)

6. What steps would you take to record a MIDI performance to a new clip in the Session View? (See “Recording a MIDI Performance” beginning on page 155.)

7. What steps would you take to record a MIDI performance to a new clip in the Arrangement View? (See “Recording a MIDI Performance” beginning on page 155.)

8. What are two ways you can manually create MIDI notes using a mouse and keyboard in the MIDI Note Editor? (See “Creating a MIDI Performance Manually” beginning on page 158.)

9. How do you display Live’s MIDI Note Editor? What are the main components of the MIDI Note Editor? (See “The MIDI Note Editor”

beginning on page 158.)

10. What is velocity in the context of MIDI data? What are two ways you can go about editing the velocity values of MIDI notes? (See “Editing Velocity” beginning on page 162.)

11. What is MIDI quantization? How would you go about using quantization to improve the timing of a MIDI performance without sacrificing its “performed” feeling? (See “MIDI Quantization” beginning on page 164.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 6

Working with MIDI

Activity

In this exercise, you will continue work on the Set that you started in Exercises 4 and 5. You will add character to the project by creating both a drum part and a melody part to accompany the chord progression and bass part.

This exercise provides three approaches: Option 1 lets you record MIDI performances from a MIDI controller; Option 2 lets you manually create MIDI performances using the MIDI Note Editor; and Option 3 lets you import existing MIDI clips to use for the new parts. Each of the three options is covered separately.

Duration

This exercise should take approximately 10 to 15 minutes to complete.

Goals/Targets

- Configure the Set
- Assign Instruments to MIDI tracks
- Prepare the Set for recording
- Record or manually create a MIDI performance

Media Files

To complete this exercise, you may need to use various media files included in the Media Files 2019-Live101 folder. You should have downloaded the media files in Exercise 4.

If needed, you can re-download the media files by going to www.halleonard.com/mylibrary and entering your access code (printed on the opening page of this book). From there, click the Download link for the Media Files 2019-Live101 listing in your My Library page. The Media Files folder will begin transferring to your Downloads folder.

Getting Started

You will start by opening the Live Set you saved at the end of Exercise 5. If that file is not available, you can use the Ex05 Sample file in the 01. Completed Exercises folder within the Media Files 2019-Live101 folder.

Open the Set and save it as Exercise 6:

1. Do one of the following:

   - Open the Set file that you created in Exercise 5 (Storage Drive/Folder > Live101-XXX > Exercise05-XXX.als).

   - Alternatively, you can use the Ex05 Sample file (Media Files 2019-Live101 > 01. Completed Projects > Ex05 Sample.als).

2. Choose **File** > **Save Live Set As** and name the Set *Exercise06-XXX*, keeping the Live Set inside the original Project folder. (Move the Set into your Live101-XXX folder if working from the sample file.)

In Exercise 5, you created the foundation of a song segment using imported audio files. You had the choice of four bass and chord progression options, spanning several styles of music and several tempos. To review the genre and

BPM you chose, click any of the audio clips. Then press **SHIFT+TAB**, as needed, to display the Clip View, where you can see the file name of the referenced audio. This will include the genre, key, and tempo of the file.



Figure 6.41 Audio file information in the Clip View

For the full name and location of an audio file referenced in a clip, hover over the clip on a track or clip slot and see the Status Bar at the bottom of the Live window.

From here, you will augment this song segment using the two MIDI tracks: Drums and Melody.

Configuring the Set

Before continuing, you will configure the Live Set to ensure that the tempo is correct and that Record Quantization is enabled.

Configure the Set:

1. Make sure the tempo of your Live Set matches the tempo of your audio clips.

As noted above, the tempo is listed on the file names of the audio files you selected in Exercise 5.

2. Select **EDIT** > **RECORD QUANTIZATION** and choose a value. Higher quantization values, such as Quarter-Note Quantization, will have a more dramatic effect on your MIDI performance when you record from a MIDI controller.

Alternatively, you can leave Record Quantization disabled and quantize your MIDI performance after recording using the MIDI Note Editor.

Assigning Instruments to MIDI Tracks

You will now assign instrument devices to both of the MIDI tracks in your project so that they can produce sound. The Drums track will use the Drum Rack instrument, and the Melody track will use the Simpler instrument.

Assign a virtual instrument to each MIDI track:

1. Show the Browser by selecting **VIEW** > **SHOW BROWSER** or by pressing **OPTION+COMMAND+B** (Mac) or **CTRL+ALT+B** (Windows).

2. Select the Drums MIDI track.

3. In the browser, choose the **DRUMS** category.

4. Audition the pre-created drum kits by clicking on each one to select one for use in your Live Set.

5. With the Drums track still selected, double-click the name of the your selected drum kit to insert a Drum Rack device.



Figure 6.42 A Drum Rack preset on the Drums MIDI track

6. Select the Melody MIDI track.

7. In the browser, select **INSTRUMENTS**.

8. Click the disclosure triangle next to Simpler to browse the different synth options, then expand the categories and click on the sounds to audition them. Some good places to look for the purposes of this exercise are Pianos & Keys and Synth Lead.

9. Once you've found a sound you like that fits with your existing audio clips, double-click its name to add the Simpler instrument to the Melody track.

Getting Ready to Record

This section will help you prepare to record a MIDI performance using a MIDI controller device or the computer keyboard. If you are not using a MIDI controller, you can skip this section.

Select a MIDI controller to use on each MIDI track:

1. Make sure that the In/Out controls are visible by selecting **VIEW** > **IN/OUT** or pressing **OPTION+COMMAND+I** (Mac) or **CTRL+ALT+I** (Windows).

2. Locate the In/Out section of the Drums MIDI track.

3. Use the **INPUT TYPE** dropdown menu to select the MIDI controller you will use to record your performance.

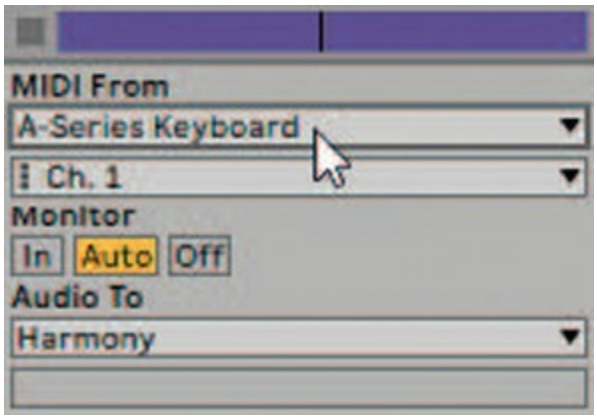


Figure 6.43 In this example, an A-Series Keyboard is selected as the MIDI input device

4. Repeat this process for the Melody MIDI track.

If you are not using a MIDI controller to record a performance, you can either import MIDI data from the Exercise Media folder or manually create a MIDI performance. Both options are described later in this exercise.

Enable the metronome and count-in to aid in recording:

1. Click the **METRONOME** switch in the Control Bar to enable the metronome.

2. Click the down arrow on the **METRONOME** switch to enable the count-in from the pop-up menu.

3. Select a count-in value of 1 or 2 Bars.

You can also customize the sound of the metronome tick, if desired, as described in Chapter 6.

Recording a MIDI Performance (Option 1)

In this section, you will record a MIDI performance for both the Drums track and the Melody track. If you prefer not to perform live recording passes, skip this section and continue with Option 2 or Option 3 later in this exercise.

Recording in Session View

To get started, you will record MIDI clips into Clip Slots in the Session View.

Record the MIDI drum and melody performance:

1. Press **TAB** as needed to display the Session View.

2. **ARM** the Drums MIDI track for recording.

3. Click the **RECORD** button in the second clip slot of the Drums track. This will initiate the count-in.

4. After the count-in, begin recording your drum pattern.

   A good length is 4 bars, matching the audio clip length in the Live Set. However, you can record a clip of any length, and Live will continue to loop the clip as you play Scene 2 in Session View.

5. Once you have recorded your drum pattern, press **SPACEBAR** to stop the recording.

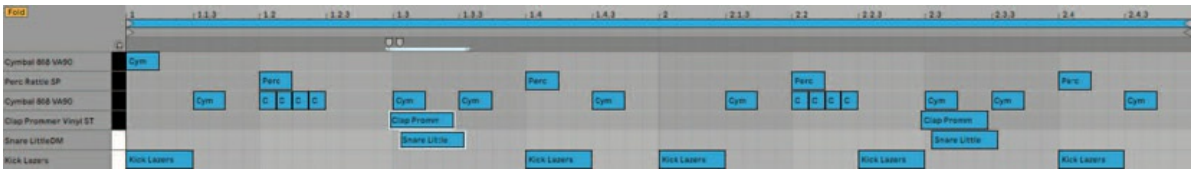


Figure 6.44 A 2-bar MIDI drum performance with Fold mode enabled

6. Launch Scene 2 by clicking the Scene Launch button on the Master track to listen to your MIDI performance along with the existing clips.

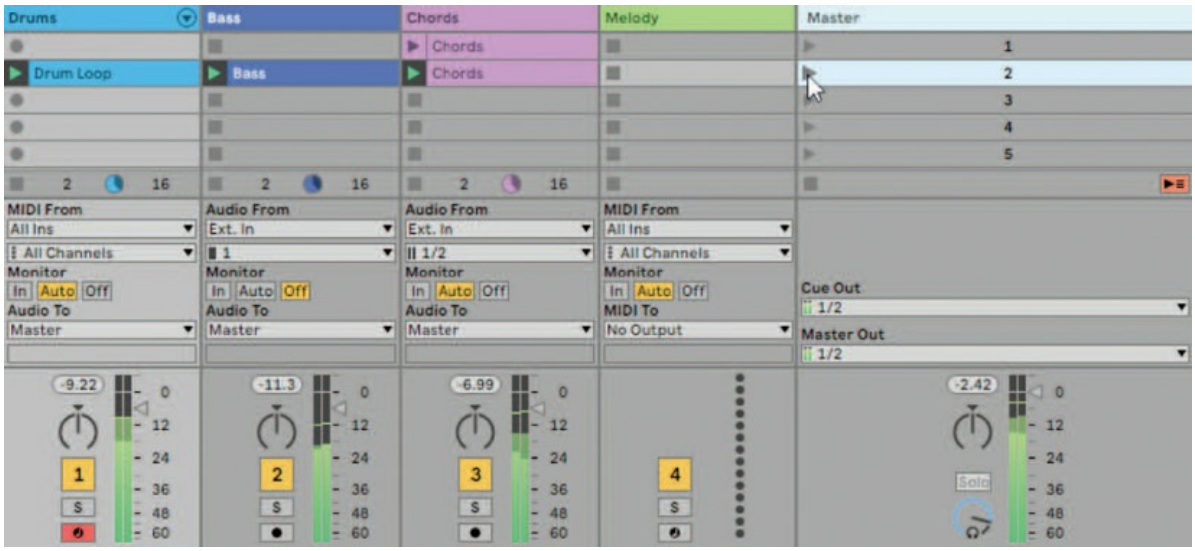


Figure 6.45 Listening to Scene 2 to audition the drum recording along with the bass and chord clips

7. If you're not satisfied with the results, you can either delete the clip and try again, or edit and quantize the performance to improve the timing. You will have an opportunity to edit the MIDI performance later in this exercise.

   You can also use the MIDI Arrangement Overdub feature at this point, if desired, to layer additional MIDI notes over your initial performance.

To add notes without overwriting your performance, click the MIDI ARRANGEMENT OVERDUB button, then begin recording by pressing the SESSION RECORD button (hollow circle) in the Control Bar.

8. Once you are satisfied with your recording, copy the clip to the first clip slot of the Drums track by holding **OPTION** (Mac) or **ALT** (Windows) while dragging the clip to the top slot position.

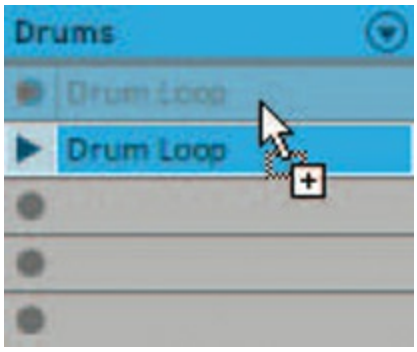


Figure 6.46 Duplicating a clip to another slot on the Drums track by Option-/Alt-dragging (Mac/Windows)

Alternatively, you could choose to leave the drums only on Scene 2, allowing you to introduce the drums along with the bass.

9. Repeat the above process on the Melody MIDI track. This time, you will record a simple melody to accompany the drums, bass, and chords.

All of the provided audio clips for this exercise are in the key of either C major or A minor. Try using the notes C, D, E, F, G, and A to construct your melody.

Recording in Arrangement View

In this section, you will record MIDI onto tracks using Arrangement View.

Record the MIDI drum and melody performance:

1. Press **TAB** to toggle the display to the Arrangement View.

2. Click the red **BACK TO ARRANGEMENT** button in the Arrangement View to reactivate playback for the Arrangement View, as needed.



Figure 6.47 Clicking the Back to Arrangement button

For details on the Back to Arrangement button, see Chapter 3 in this book.

3. A RM the Drums MIDI track for recording.

4. Place the Arrangement Insert Marker at the start of the arrangement.

Clicking the STOP button in the Control Bar at anytime while playback is stopped will return your Arrangement Insert Marker to the start of the Live Set.

5. Click the **ARRANGEMENT RECORD** button (solid circle) in the Control Bar. This will initiate the count-in.

6. After the count-in completes, begin recording for a 16-bar drum pattern.

   You can record an entire 16-bar clip, or record a shorter clip—such as 4 or 8 bars—and duplicate it to fill the required length.

7. Once you have completed the recording, press **SPACEBAR** to stop the recording.

8. Duplicate the clip as needed to span 16 bars. (Select the clip and choose **EDIT > DUPLICATE**.)

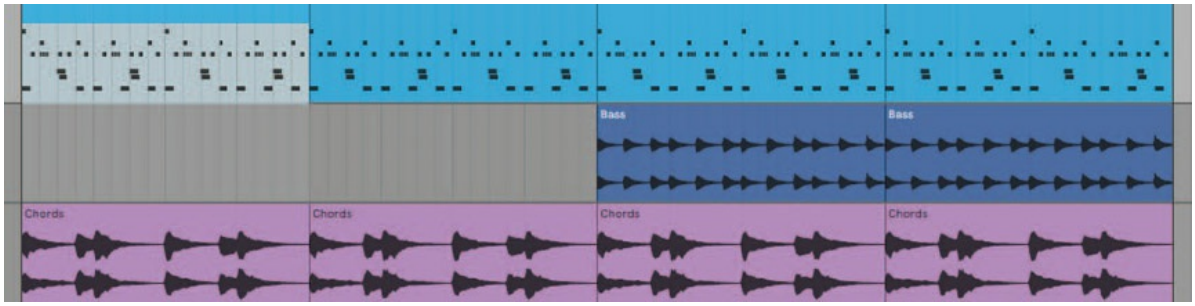


Figure 6.48 A 4-bar MIDI drum performance (light blue), duplicated to span the entire arrangement

9. Place the Arrangement Insert Marker at the start of the arrangement again and press **SPACEBAR** to listen to your work.

- If you're not satisfied with the results, you can either delete the performance and try again, or edit and quantize the performance to improve the timing.

- You can also use the MIDI Arrangement Overdub feature to layer additional MIDI notes over your base performance. Click the MIDI **ARRANGEMENT OVERDUB** button; then repeat the recording process above to add additional notes without overwriting your original performance.

10. Repeat the above process to record a part on the Melody MIDI track. On this track, compose a simple melody to accompany the drums, bass, and chords.

All of the provided audio clips for this exercise are in the key of either C major or A minor. Try using the notes C, D, E, F, G, and A to construct your melody.

Edit Your MIDI Performances

In this section, you will have the opportunity to fine-tune your MIDI performances as needed.

Tweak your MIDI performances using the MIDI Note Editor:

1. Double-click a MIDI clip you that would like to edit in the MIDI Note Editor.

2. Using editing features such as moving, trimming, and quantizing, tweak the MIDI performance to your liking.

To quantize your performance, select all of the notes in the MIDI Note Editor by pressing COMMAND+A (Mac) or CTRL+A (Windows), and then select EDIT > QUANTIZE SETTINGS to configure the quantize behavior.

Manually Creating a MIDI Performance (Option 2)

If you are unable to record a MIDI performance using a MIDI controller or your computer keyboard, you can instead create a MIDI performance manually using the MIDI Note Editor. (If you have completed the recording process in Option 1, you can skip this section.)

Creating a Performance in Session View

In this section, you will create MIDI clips for Clip Slots in the Session View.

Manually create a MIDI drum and melody performance in the Session View:

1. Double-click the second clip slot on the Drums track to create an empty clip.

2. Under the **NOTES** section in the Clip View, set the clip Length to 4 bars (4 . 0 . 0)



Figure 6.49 Clip length set to 4 bars

3. Create notes for different drum sounds at different points in time. For instance, you might start by inserting a kick drum note on beats 1 and 3 of each measure and then add snare hits on beats 2 and 4. Then you can add hi-hats and other drums and experiment with timing patterns from there.

Double-click at the desired location in the MIDI Note Editor to add a new note in Edit Mode (or single-click with the Pencil tool in Draw Mode).

Press the B key at any time to toggle between Edit Mode and Draw Mode in the MIDI Note Editor.

4. Once you are satisfied with your performance, copy the clip to the first clip slot of the Drums track by holding **OPTION** (Mac) or **ALT** (Windows) while dragging the clip to the top slot position.

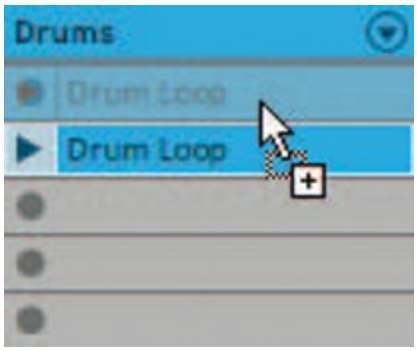


Figure 6.50 Duplicating a clip to another slot on the Drums track by holding Option (Mac) or Alt (Windows) while dragging

Alternatively, you could choose to leave the drums only on Scene 2, allowing you to introduce the drums along with the bass.

5. Repeat the above process to create a part for the Melody MIDI track. On this track, compose a simple melody to accompany the drums, bass, and chords.

All of the provided audio clips for this exercise are in the key of either C major or A minor. Try using the notes C, D, E, F, G, and A to construct your melody.

Creating a Performance in Arrangement View

In this section, you will create a MIDI performance for each track using the Arrangement View.

Manually create a MIDI drum and melody performance in the Arrangement View:

1. Press **TAB** to toggle the display to the Arrangement View.

2. Click the red **BACK TO ARRANGEMENT** button in the Arrangement View to reactivate playback for the Arrangement View, as needed.

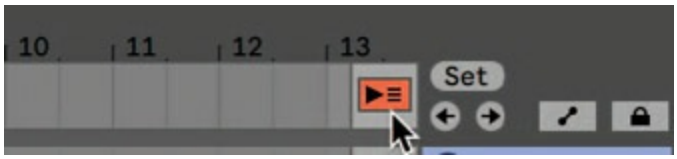


Figure 6.51 Clicking the Back to Arrangement button

For details on the Back to Arrangement button, see Chapter 3 in this book.

3. Click and drag on the Drums MIDI track in the Track Display window, selecting from the start of the Arrangement (at Bar 1) up to the start of Bar 5 (a 4-bar timespan).

4. Select **CREATE > INSERT MIDI CLIP** (S), or press **COMMAND+SHIFT+M** (Mac) or **CTRL+SHIFT+M** (Windows), to create an empty 4-bar clip.

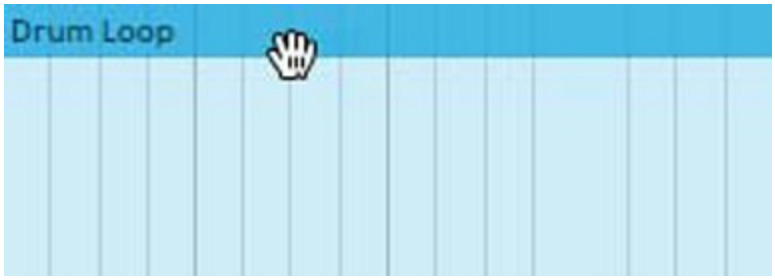


Figure 6.52 An empty MIDI clip

5. Double-click the empty MIDI clip to open it in the MIDI Note Editor.

6. As you did when creating the drum clip in Session View, create notes for different drum sounds at different points in time. Again, you could start by inserting kick drum notes on beats 1 and 3 of each measure, then add snare hits on beats 2 and 4, and then add hi-hats, other drums, and pattern variations from there.

Double-click at the desired location in the MIDI Note Editor to add a new note in Edit Mode (or single-click with the Pencil tool in Draw Mode).

Press the B key at any time to toggle between Edit Mode and Draw Mode in the MIDI Note Editor.

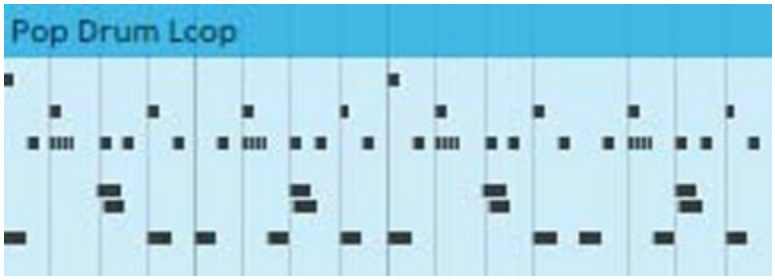


Figure 6.53 A clip including MIDI note data

7. Once you have created a drum pattern you like, you can move and duplicate the clip in the Track Display window so that it plays at the times you desire.

   To duplicate the clip, select it and choose **EDIT** > **DUPLICATE**.

8. Repeat the above process on the Melody MIDI track. On this track, compose a simple melody to accompany the drums, bass, and chords.

All of the provided audio clips for this exercise are in the key of either C major or A minor. Try using the notes C, D, E, F, G, and A to construct your melody.

Importing MIDI Clips (Option 3)

Finally, if you would prefer to use pre-created MIDI clips, you can simply import MIDI data into your Live Set. If you are satisfied with the MIDI performances you recorded or created in Option 1 or Option 2, you can skip this section; otherwise you can follow the steps below to import MIDI in lieu of completing Option 1 or 2 or as a replacement for the MIDI performances you created under Option 1 or 2.

Importing MIDI in Session View

In this section, you will import MIDI clips into the Session View.

Import pre-created MIDI files in the Session View:

1. Press **TAB** as needed to toggle to the Session View.
2. Select the second clip slot of the Drums MIDI track.
3. Select **CREATE** > **IMPORT MIDI FILE**.
4. In the Import dialog box, navigate to the Media Files 2019-Live101 folder; then select an appropriate MIDI drum pattern to match your audio from the 02 Exercise Media > [MIDI] folder.
5. Click **OPEN**. The selected clip will be added to the clip slot in the Session View.
6. Right-click the clip and select **DUPLICATE** to copy it to the second clip slot on the Drums track.
7. Edit the MIDI performance as desired by double-clicking the clips and using the MIDI Note Editor.
8. Repeat this process for the Melody MIDI track, selecting an appropriate melody clip for your Set.

Importing MIDI in Arrangement View

In this section, you will import MIDI clips into the Arrangement View.

Import pre-created MIDI files in the Arrangement View:

1. Press **TAB** to toggle to the Arrangement View.
2. Click the red **BACK TO ARRANGEMENT** button in the Arrangement View to reactivate playback for the Arrangement View, as needed.



Figure 6.54 Clicking the Back to Arrangement button

For details on the Back to Arrangement button, see Chapter 3 in this book.

3. Place the Arrangement Insert Marker at the start of the arrangement.

Clicking the STOP button in the Control Bar at anytime while playback is stopped will return your Arrangement Insert Marker to the start of the Live Set.

4. Select **CREATE** > **IMPORT MIDI FILE**.

5. In the Import dialog box, navigate to the Media Files 2019-Live101 folder; then select an appropriate MIDI drum pattern to match your audio from the 02 Exercise Media > [MIDI] folder.

6. Click **OPEN**. The selected clip will be added to the track in Arrangement View, and a dialog box will appear prompting you to select an option for importing tempo and time signature information.

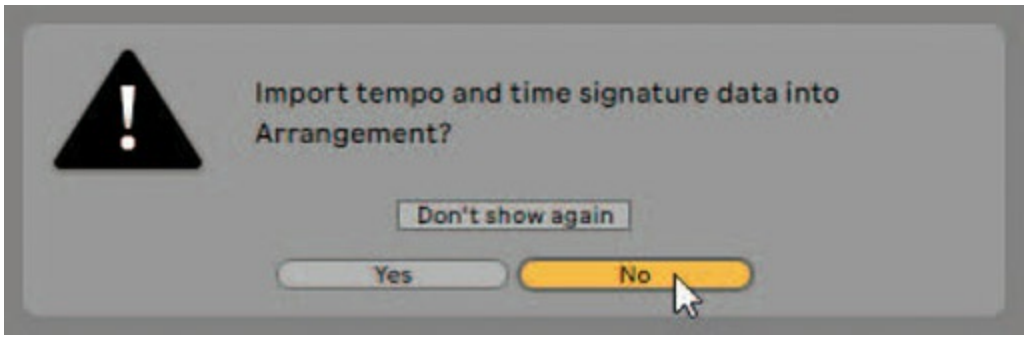


Figure 6.55 MIDI import options in the Arrangement View

7. Select **NO** in the dialog box, as you have already set the tempo of the Live Set.

8. Duplicate the drum clip across the arrangement by selecting it and

pressing **COMMAND+D** (Mac) or **CTRL+D** (Windows).

9. Edit the MIDI performance as desired by double-clicking the clips and using the MIDI Note Editor.

10. Repeat this process for the Melody MIDI track, selecting an appropriate melody clip for your Set.

Finishing Up

To complete this exercise, you will need to save your work and close the Live Set. You will be reusing this project in Exercise 7, so it is important to save the work you've done.

Review and your work in the Session View:

1. Press **TAB** to toggle the display to the Session View.

2. Click the **PLAY** button on Scene 1 of the Master track to play back the clips in Scene 1.

3. Click the **PLAY** button on Scene 2 of the Master track to play back the clips in Scene 2.

4. Repeat for any other Scenes that include clips.

5. Press the SPACEBAR or the **STOP ALL CLIPS** button on the Master track to stop playback.



Figure 6.56 Example of the Session View after completing Exercise 6

Review your work in the Arrangement View:

1. Press Tab to toggle the display to Arrangement View.

2. Click the red **BACK TO ARRANGEMENT** button to reactivate playback for the Arrangement View.

3. Click the Stop button in the Control Bar to position the Arrangement Insert Marker at the start of the clips.

4. Press the SPACEBAR to play back the Live Set and confirm your results.

5. Press the SPACEBAR a second time when finished.

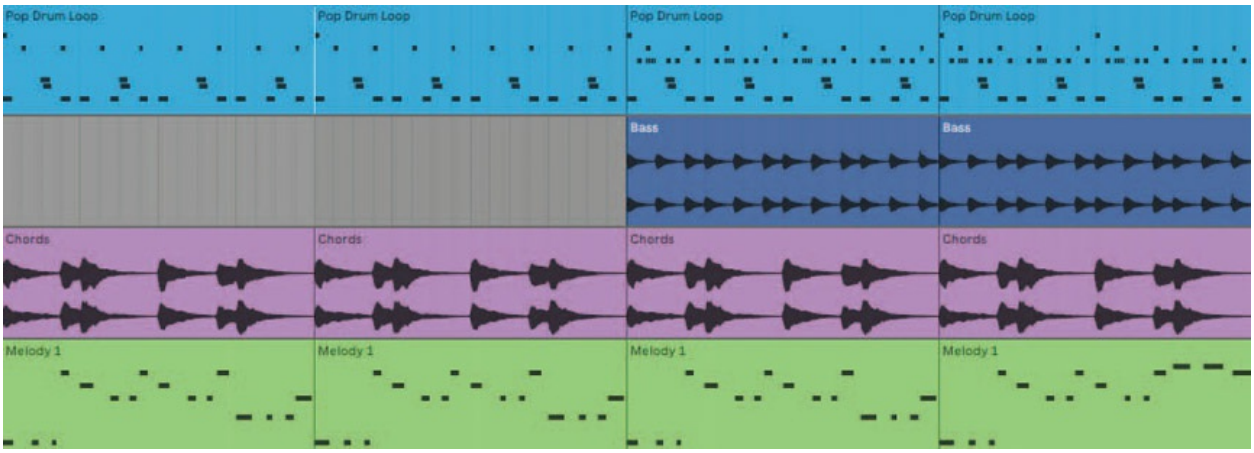


Figure 6.57 Example of the Arrangement View after completing Exercise 6

Save your work and close the Live Set:

1. Choose **File > Save Live Set**.

2. Press **Command+Q (Mac) or Ctrl+Q** (Windows) to quit Live and close the project.

Chapter 7

Making Your First Audio Recording

This chapter covers the steps required to begin recording audio into a Live Set. It also describes the types of audio files your project will include and covers processes for keeping your audio files organized.

Learning Targets for This Chapter

- Configure the metronome to aid in recording audio
- Record audio onto tracks in your Live Set from external and internal sources
- Record in both the Arrangement View and Session View
- Recognize the difference between audio clips and audio files
- Manage audio files after recording to minimize clutter and optimize your Live Set

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform. Sign up for free at ElementsED.com.

Many Ableton Live productions require some amount of audio recording, whether the source is live instruments and voices or other tracks within the Live Set. While the recording controls and processes in Ableton Live are intuitive in many respects, getting optimal results can be challenging if you are new to recording audio.

This lesson provides the background information on recording that you'll need to get started on the right foot. Whether your audio endeavors involve a simple setup in a home studio or an elaborate system in a professional environment, this information will help you take the first steps toward capturing quality audio recordings in Ableton Live.

Before Recording

Before you begin recording in a Live Set, you should ensure that your system has enough storage space for the parts you plan to record. The amount of storage space consumed by audio files will vary, depending on the bit depth and sample rate of the project. (See the "Converting Audio to Digital Format" section in Chapter 1 for a detailed discussion of bit depth and sample rate.)

Audio Storage Requirements

Ableton Live records all audio at sample rates ranging from 44.1 kHz to 192 kHz – the upper limit depends on the capabilities of your audio hardware, as discussed in Chapter 4 – and with bit depths between 16-bit and 32-bit floating point. At a sample rate of 44.1 kHz, each track consumes approximately 5 megabytes (MB) of disk space per minute for 16-bit audio (mono), 7.5 MB per minute for 24-bit audio (mono), and

10 MB per minute for 32-bit floating-point audio. With increasing bit depth and sample rates, hard disk space consumption increases correspondingly; recording at a sample rate of 88.2 kHz, therefore, consumes twice as much space as recording at 44.1 kHz. Similarly, recording in stereo consumes twice the space of recording in mono.

Table 4.1 in Lesson 4 shows approximate storage consumption at the

different data rates supported by Ableton Live.



Ableton Live creates a separate analysis file (.asd) for each audio file loaded into a Live Set or previewed in the Browser. These files reference their respective audio files, and contain data that helps Live optimize stretching quality, speed up the waveform display, and automatically detect the tempo of long samples.



Clip settings files consume a small but variable amount of additional disk space, depending on the size of the audio file they each reference.

Calculating File Sizes

The sample rate and bit depth of a recorded audio file are directly related to the resulting file size. In fact, you can calculate file sizes using these two parameters with the following equations:

Sample Rate x Bit Depth = Bits per Second

Sample Rate x Bit Depth x 60 = Bits per Minute

In the binary world of computers, 8 bits make a byte, 1,024 bytes make a kilobyte (KB), and 1,024 KB make a megabyte (MB). Therefore, the file size equation can be restated as follows:

(Sample Rate x Bit Depth x 60) / (8 bits per byte x 1,024 bytes per kilobyte x 1,024 kilobytes per megabyte) = Megabytes (MB) per Minute

Reducing terms gives us the following:

Sample Rate x Bit Depth / 139,810 = MB per Minute

For example, recording audio at a sample rate of 44,100 samples per se cond with a bit depth of 24 bits per sample would generate files that

consume space at the following rate:

44,100 x 24 / 139,810 = 7.57 MB per Minute

Determining Available Disk Space

Now that you have an understanding of how much disk space is required to store recorded audio files at different sample rates and bit depths, you can estimate whether your system or storage drive has enough space available to allow you to complete your project. There are many ways to check the amount of available space on the hard drive where Ableton Live saves your audio files, but Ableton Live does not directly provide such information. Consult your operating system manual for more information.

Preparing to Record

Once you have created a Live Set, added an audio track (or tracks) on which to record, and verified that you have adequate disk space available, you will need to prepare your hardware and Live Set for recording. You might also want to enable and configure the metronome (sometimes called the *click* or *tick*) to use as a tempo reference while recording. Whether or not you use the metronome, the general processes you will use to prepare for recording audio are as follows.

To prepare to record audio:

1. Check the hardware connections, if you're using external instruments.
2. Route the signal to be recorded to an audio track.
3. Record-arm the audio track on which you'd like to record.
4. Set the input level of the signal to be recorded.

Enabling the Metronome

As discussed in Chapter 6, when you're recording for a song that is based on

a specific tempo (or tempos) with established bar and beat divisions, it can help to perform along with a metronome. The metronome will help performers synchronize to the composition while tracking and will ultimately help align the recorded parts to the bars and beats in the Set. Having your recorded parts in alignment with the tempo will help you to make selections according to the musical grid, and can simplify tasks such as copying and pasting an entire song section as you work on your arrangements.

To enable or disable the metronome, click the **METRONOME** switch in the Control Bar. When enabled, the metronome will begin ticking to the beat whenever you press the **PLAY** button in the Control Bar, launch a clip from the Session View, or start a record pass.

Customizing the Metronome

As described previously in Chapter 6, Ableton Live includes several metronome options, which are available by clicking the down arrow icon on the Metronome switch.

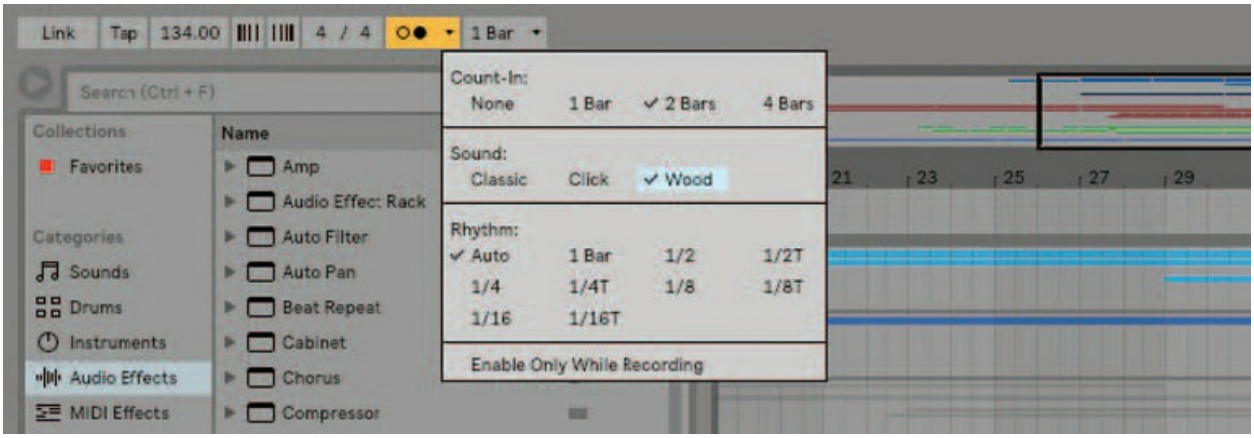


Figure 7.1 The Metronome switch (yellow) and associated settings dropdown menu

By way of recap, the available options for the Metronome include:

- **Count-In:** determines the number of bars the metronome will tick before recording commences
- **Sound:** allows you to choose between three styles of metronome ticks: Classic, Click, and Wood

- **Rhythm:** determines the musical divisions that will be represented by the metronome

- **Enable Only While Recording:** causes the Metronome to sound during recording only, and not during playback

You can control the volume level of the metronome by adjusting the Preview/Cue Volume on the Master track.



Figure 7.2 Use the Preview/Cue Volume knob (blue) on the Master track to adjust the metronome volume relative to the level of the Live Set

Checking Hardware Connections

When recording audio from an external source, such as an instrument, microphone, or other sound source, you first need to connect the source to the system that is running Ableton Live. This is generally done through an audio interface, which is capable of converting incoming signals to digital audio and typically also lets you amplify the incoming signal using a preamp or gain control.

Most audio interfaces have inputs designated for different sound sources and input types. Before starting to record, you should verify that your sound source is connected to an appropriate input on the audio interface, and that the signal is being passed through the interface correctly. For basic recording, it is simplest to use the lowest available inputs on your audio interface (for example, Input 1 for a mono source or Inputs 1 and 2 for a stereo pair).

If necessary, check the configuration of your audio interface and/or the

AUDIO tab of Ableton Live's Preferences window to ensure that your audio device's hardware inputs will be available on audio tracks.



For details on configuring Preferences in Ableton Live, see Chapter 4.

Selecting and Routing the Sound Source

With your sound source connected to the inputs of your audio interface, you are now ready to configure an audio track to receive a signal from your source and to pass the signal through the system for recording and monitoring purposes.

Displaying In/Out Controls

Before moving forward, make sure that the In/Out controls are visible by selecting **VIEW** > **IN/OUT** or pressing **OPTION+COMMAND+I** (Mac) or **CTRL+ALT+I** (Windows). This will reveal input and output choosers for all the tracks in your Live Set in both the Arrangement View and Session View.



Figure 7.3 In/Out controls in the Arrangement View

Selecting Input Type and Input Channel

Each audio track has an *Input Type* chooser and an *Input Channel* chooser at the top of the track's In/Out section. This selector allows you to route a signal from an input on your interface to the track for recording.

To set the incoming signal in both the Arrangement View and Session View, do the following:

1. Locate the In/Out section of the track on which you will record.

2. Using the top chooser, select the **INPUT TYPE** (if using an external audio interface or the computer's built-in audio inputs, select Ext. In).

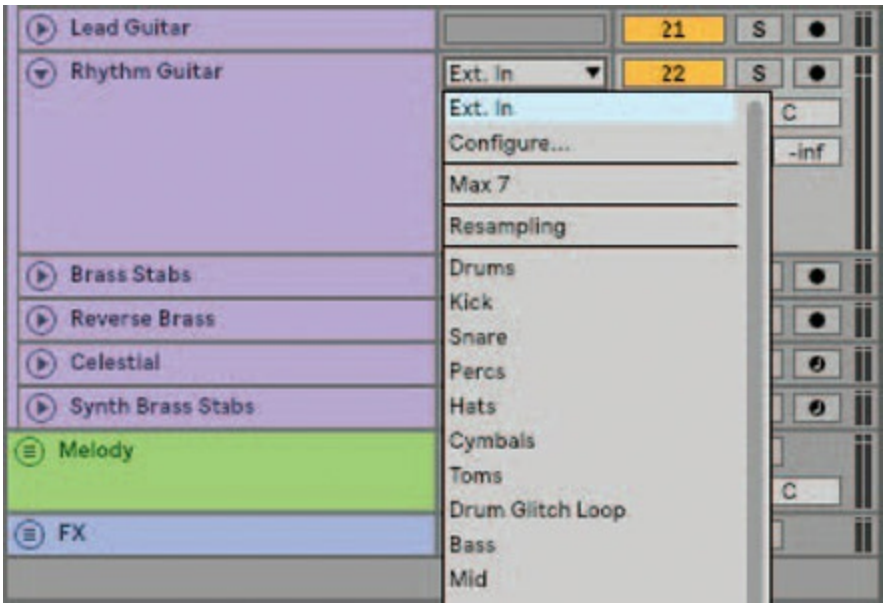


Figure 7.4 The Input Type chooser on the Rhythm Guitar audio track in Arrangement View

3. Using the second chooser (underneath the Input Type), select the appropriate **INPUT CHANNEL**. Select a single channel (such as 1 or 2) to record a mono sound source or a channel pair (such as 1/2 or 3/4) to record a stereo sound source. (See Figure 7.5.)

Make sure that the selection displayed on the Input Channel chooser matches the input number(s) that your sound source is plugged into on your audio interface. Otherwise you may record the wrong signal (or no signal) to the track.

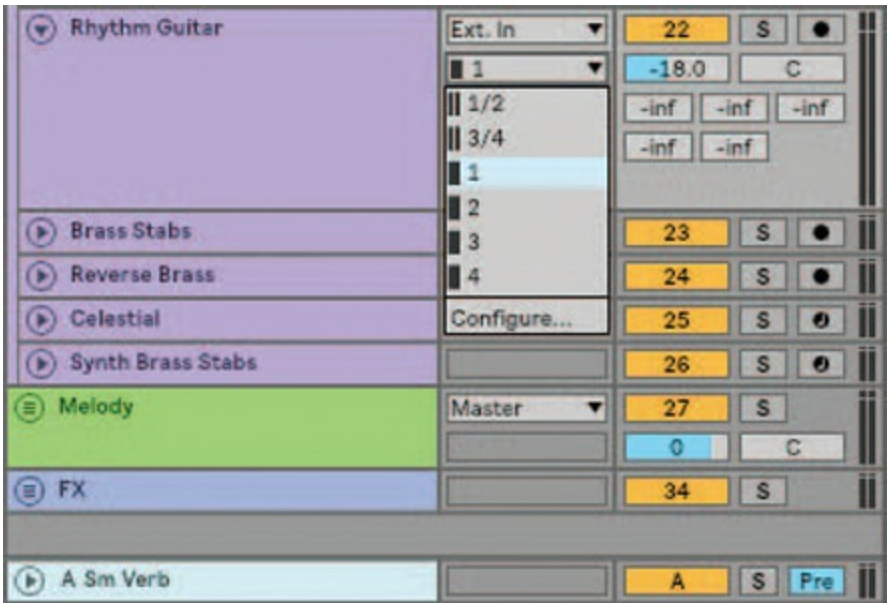


Figure 7.5 Selecting a mono source from the Input Channel chooser on an audio track in Arrangement View

Setting Input Channels for Recording on Multiple Tracks

When recording to multiple tracks simultaneously, each track will need to have a unique input channel. For example, suppose you are recording a vocalist on Track 1 and an acoustic guitar accompaniment on Track 2 of your Live Set. You might have the vocal microphone connec ted into Input 1 of your audio interface, and the guitar micro phone connected into Input 2.

In the Live Set window, you would set the Input Channel for your first track to 1 (or the corresponding input channel name, such as Analog 1). Similarly, you would set the Input Channel for your second track to 2 (or its corresponding input channel name).

This setup will allow you to record and process the vocal signal on the first track separate from the guitar signal on the second track.

Recording an Existing Track to a New Audio Track

Many composers who work in Ableton Live commonly record the output of a track already existing within their Live Set to a new audio track. This workflow offers many creative editing possibilities, such as Warping, stretching, re-pitching, and reversing the resulting audio, while preserving the original source material. Additionally, repeating this process offers an opportunity to process sounds in subsequent layers, which can result in full sound transformations. For example, you could print an existing track to a new audio file, process it with creative effects, record it again to a new audio track, and process it again with additional effects. You can repeat this process again and again to create interesting sounds.

To record an existing track to a new audio track, simply change the Input Type of the new audio track from an external input (Ext. In) to the name of the existing track you would like to record. When you initiate recording on the new audio track, the output of the existing track will be captured in a new audio file.

Arming (Record-Enabling) Tracks

To set up an audio track for recording, click the track's **ARM** button (the button with a solid circle icon) in either the Arrangement View or Session View. The button will turn red when the track is armed and ready to record.

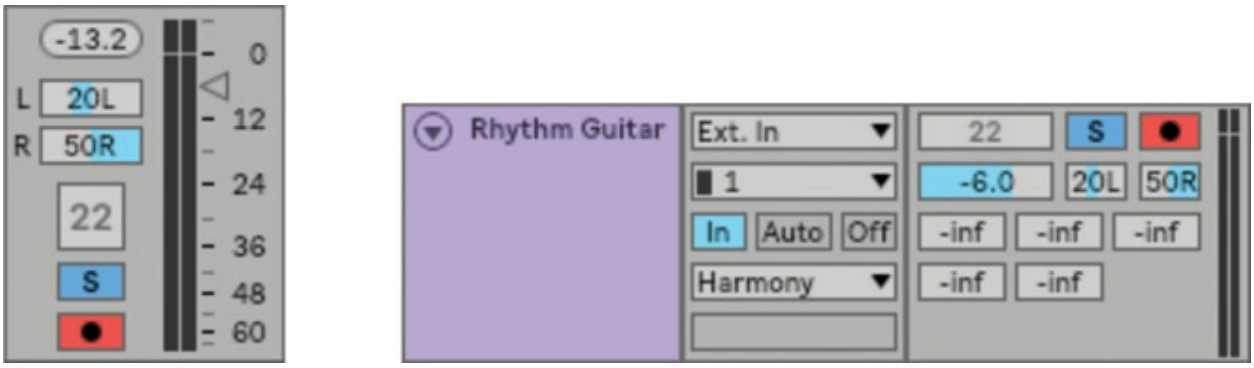


Figure 7.6 The track Arm button in the Session View (left) and in the Arrangement View (right)

To arm multiple tracks for recording, COMMAND - CLICK (Mac) or

CTRL-CLICK (Windows) on the ARM buttons on additional tracks.

Monitoring Modes

In order to hear your sound source through Ableton Live, you need to choose a Monitoring mode. The three modes are listed in the In/Out section below the Input Type and Input Channel choosers.

- **Off:** In this mode, monitoring is turned off, meaning you will not hear the sound source through Ableton Live. This might be useful if you are recording an acoustic instrument and want to exclusively hear the sound of the instrument, rather than the recorded sound.

- **Auto:** In general, this Monitoring mode will be best for most recording applications. As long as the track is armed for recording, your input signal will be passed through any processing to the output so you can hear what is being recorded onto the track. The only time you *won't* hear your sound source is during normal playback, making it easy to listen to what you've recorded.

- **In:** With this Monitoring mode, you will always be monitoring your input signal – regardless of whether the track is armed for recording or the Live Set is playing. This mode can be useful to allow the talent to rehearse a part while listening to playback from other tracks in the set.

Controlling Input Level and Pan

Adjusting input levels and panning is essential to a smooth recording process.

Input Level

As a general rule, input levels should be adjusted to obtain a strong, clean signal while avoiding clipping. You do not, however, need to record at the highest possible level. Recording too hot can leave little room for subsequent gain-based processing (such as EQ) and can lead to digital clipping, which is detrimental to audio quality.

For best results, aim for an average peak input level around –6 dBFS or lower, and ensure that the track meter never turns red. To do this, adjust the level of your analog source while monitoring the volume meter of your track within the Live Set.



The amplitude of digital audio is represented in decibels relative to full-scale audio (dBFS), with full scale indicating the loudest signal that can be recorded at a given bit depth. Digital clipping occurs whenever a signal exceeds 0 dBFS at an input or output.



The Ableton Live mixer includes a Peak Level indicator button to the left of the track meter. Use this indicator to help set record levels. When the Peak Level meter exceeds 0.00, the button will highlight in orange.

Adjusting the input level will typically require you to change the source volume, adjust the microphone placement, or modify the incoming signal strength using a mixer or preamplifier. Note that although a track's Track Volume slider can be used to increase or decrease playback levels, the Track Volume slider *does not* affect record levels.

Many audio interfaces provide preamplifier gain controls for their inputs. For all other I/O devices, record levels are set entirely from the source or pre-I/O signal processing.

Pan Position

Setting the pan affects the stereo placement of a signal for monitoring and playback purposes only; it has no effect on how the audio files are recorded.

Ableton Live provides a graphical Track Pan control (pan knob) for audio tracks in the Session View by default and a Track Pan value display for audio tracks in the Arrangement View.

Track Pan in Session View

To set the pan position of the signal in the Session View, change the position of the **STEREO PAN** knob for the track in the track mixer section.

To control panning independently for the left and right channels of a stereo source, right-click the Stereo Pan knob and click SELECT SPLIT STEREO PAN MODE. The pan knob will be replaced by Left and Right pan value displays.

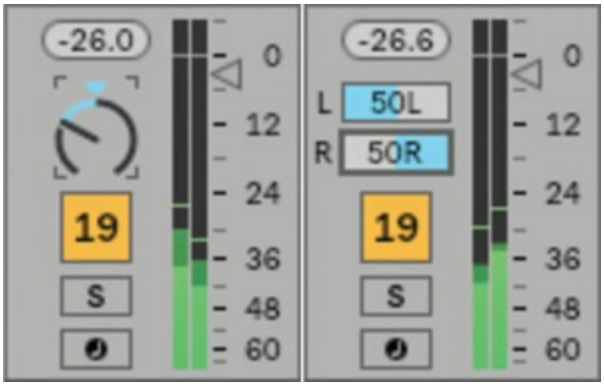


Figure 7.7 Stereo Pan knob display (left) and Split Stereo Pan display (right) in Session View

Arrangement View

To set the pan position of the signal in the Arrangement View, click and drag in the **TRACK PAN** value display in the track mixer section. The pan value is initially set to C, signifying the center of the stereo field. Pan settings range from 50L (hard left) to 50R (hard right).

To control panning independently for the left and right channels in Arrangement View, right-click on the Track Pan value display and choose SELECT SPLIT STEREO PAN MODE.

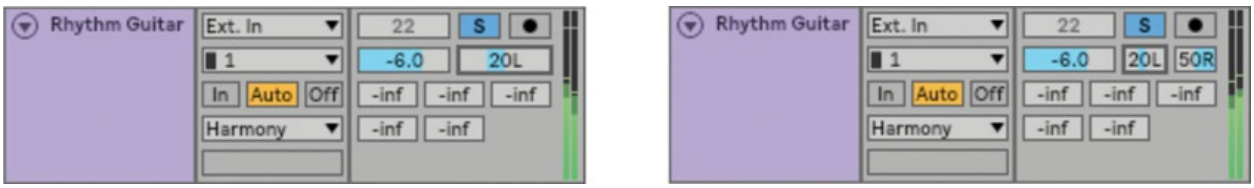


Figure 7.8 Stereo Pan display (left) and Split Stereo Pan display (right) in Arrangement View

Recording and Managing Audio

With your sound source routed to one or more tracks and the desired tracks armed for recording, you are ready to begin recording audio.

Depending on whether you are working in the Arrangement View or Session View, the recording process will vary.

Recording in Arrangement View

Arrangement View gives you a linear, timeline-based view of your recording progress. It is best suited for recording to multiple tracks simultaneously.

To begin recording audio in Arrangement View, do the following:

1. Arm at least one audio track for recording.

2. Position the Arrangement Insert Marker where you want the recording to begin.

3. Click the **ARRANGEMENT RECORD** button (solid circle) in the Control Bar. The Arrangement Record button will turn red when enabled.



Figure 7.9 The Arrangement Record button armed (red) in the Control Bar

Press function key F9 as a shortcut to activate the Arrangement Record button.

4. Depending on your Preferences settings, recording may begin immediately when you enable the Arrangement Record button. If not, click the **PLAY** button (triangle) or press the SPACEBAR to begin the record pass, starting with the metronome count-off (if enabled).

The Preferences setting under the RECORD /WARP /LAUNCH tab called START PLAYBACK WITH RECORD can be enabled to automatically start a record pass whenever the Arrangement Record button is clicked.

To override the current Preferences setting for START PLAYBACK WITH RECORD, using the opposite behavior, SHIFT-CLICK the

Arrangement Record button when starting a record pass.

5. When you have finished your record take, click the **STOP** button (solid square) in the Control Bar, or press the SPACEBAR.

Recording in Session View

Recording in Session View allows you to create individual clips and launch other clips while recording.

To begin recording audio in Session View, do the following:

1. Arm at least one audio track for recording. Once armed, each clip slot in the track will have its own record button.

2. Click the **RECORD** button on one of the clip slots to record a new clip to that slot.

If you have multiple tracks armed for recording, you can click the Session Record button (hollow circle) in the Control Bar to record onto all armed tracks on a selected scene. (See Figure 7.10 below.)



Figure 7.10 The Session Record Button (highlighted) in the Control Bar

3. The clip slot record button will flash red, and the metronome count-off will begin (if enabled).

4. Once recording has commenced, the record button will turn into a red Clip Launch button.

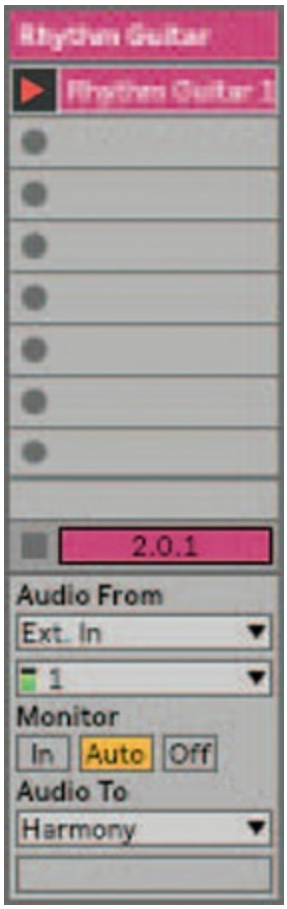


Figure 7.11 Recording audio to a clip slot in the Session View

5. When you have finished your record take, click the **STOP** button in the Control Bar (or press the **SPACEBAR**).

   Alternatively, you can press the red Clip Launch button to continue the playback of the current scene, including the new audio recording.

Session View Recording Quantization

When recording in the Session View, you might want to select a value from the Quantization Menu. When the value is set to anything other than None, Ableton Live will automatically cut the clip boundaries of your recorded audio so they align with your Live Set. This will aid in keeping clips in rhythmic alignment.



Figure 7.12 The Quantization Menu (right) with a value of 2 Bars selected

Recording on Scene Launch

The Session View can also be configured so that recording begins on all armed tracks upon launching a given scene. To do this, you need to enable the **START RECORDING ON SCENE LAUNCH** option in the Ableton Live Preferences window. This option is located under the **RECORD/WARP/LAUNCH** tab.

Organizing after Recording

Once you've completed a record pass, it is recommended that you complete a few steps to preserve and organize the audio file(s) that were created in the process.

Disarm the Record Tracks and Adjust Playback Settings

First, disarm the track(s) to which you recorded to prevent inadvertently recording further audio and overwriting clips. Once you've done that, you can also dial in basic mixing controls, such as the track volume and panning.

To return a track to playback mode and adjust the playback settings, do the following:

1. Click the **ARM** button on the audio track to take it out of record-enable mode.

2. Click **PLAY** in the Control Bar (Arrangement View) or on the recorded clip slot (Session View) to hear your performance.

3. Adjust the playback level and panning as necessary.

Organize Audio Files and Clips

Each time you record audio into Ableton Live, you create a single audio file. This single audio file is the original, unedited, continuous audio recording. It is referenced by a clip (or clips) within the Live Set, but is stored externally. Organizing audio files involves maintaining information both within the Live Set and in the Live Project folder.

When you record audio into a Live Set, the audio files are stored in the project's SAMPLES > RECORDED folder by default.

As you begin to edit your recording within the Live Set, you will likely end up creating multiple clips that reference the original audio file. These clips are simply electronic pointers within the Live Set to the audio file—they do not store audio information directly, but instead are used to display, edit, and play back audio information contained within the referenced file. Clips can range in duration from a single sample to many minutes.

Working with Clips

Audio clips represent pieces of audio data that can be moved or edited within the Live Set. They are block-like containers that can be positioned along the horizontal timeline in the Arrangement View and in dedicated slots in the Session View. Clips are created during normal editing and can refer to any type of supported audio file.

For instance, after a file has been recorded to disk, a corresponding clip will display in the Arrangement that points to the recorded audio file. You can copy, paste, split, trim, duplicate, fade, and perform other editing functions on the clip within the Live Set without affecting the original audio file.

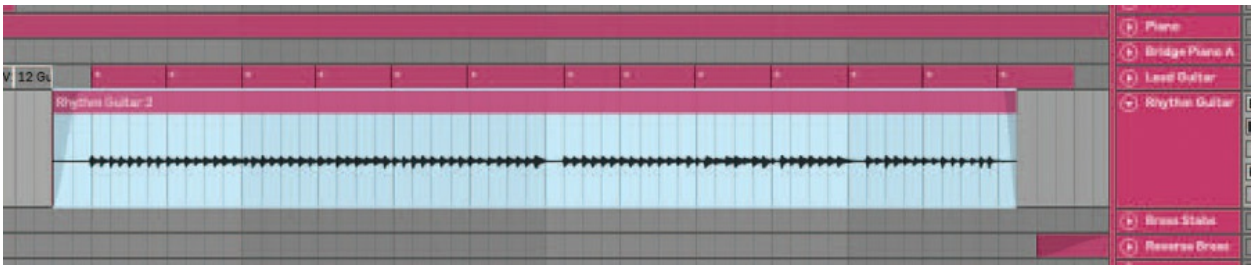


Figure 7.13 A clip referencing recorded audio, selected in Arrangement View

Naming Audio Files and Clips

During recording, Ableton Live automatically assigns names to audio files and clips. You can change these names from within the Live Set, if needed, to make them more meaningful or to help with organization.

Default Naming Conventions

When you record audio on a track, Ableton Live names the resulting file according to the track name. It also appends a take ID—a sequential number based on the number of times you've recorded on that track—followed by the exact date and time of recording in brackets.

Here is an example of the file name generated after recording on a new audio track for the first time:

1-Audio 0001 [2018-01-01 120000]	Where 1-Audio is the track name, 0001 is the take number, and 2018-01-01 120000 is the date and time of recording

Similarly, Ableton Live automatically assigns names to clips that have been created within a Live Set after recording. Clips are named by default according to the track name, with a sequential number appended for each *new clip* that is created. When copying or duplicating existing clips, they retain their original name, meaning there can be many clips with the same name.

Here is an example of the type of name Ableton Live will automatically generate for a clip:

1-Audio 3	Where 1-Audio is the name of the audio track on which the clip was created, and 3 is the clip number

Changing Clip Names

You can change the default names that Ableton Live has applied to clips at any time.

To rename a clip, select a clip in the Arrangement View or Session View and do one of the following:

- Right-click on the clip and select **RENAME** from the pop-up menu; then type the new name for the clip and press **ENTER**.
- Press **COMMAND+R** (Mac) or **CTRL+R** (Windows), enter a new name, and press **ENTER**.

■ In the Clip View, click in the **CLIP NAME** field and enter a new name.

You can apply the same name to multiple clips by first selecting all the clips you'd like to rename simultaneously.

Changing File Names

You can change the default names that Ableton Live has applied to audio files at any time, assuming the files aren't in use by another process.

To rename an audio file, use one of the following processes:

■ Process 1:

- Select a clip that references the audio file you'd like to rename.
- In the Clip View, under the Sample section, click the file name referenced by the clip to select the audio file in the Browser on the left-hand side. (See Figure 7.14.)
- Right-click the file name in the Browser and select **RENAME** from the pop-up menu. After renaming, all clips pointing to the audio file within the Live Set will now point to the newly renamed version.

■ Process 2:

- In the Ableton Live Browser, click **CURRENT PROJECT** to list all of the audio files that belong to the current Live Set.
- Right-click the file name you wish to change and select **RENAME** from the pop-up menu.

  After renaming, all clips pointing to the audio file within the Live Set will now point to the newly renamed version.

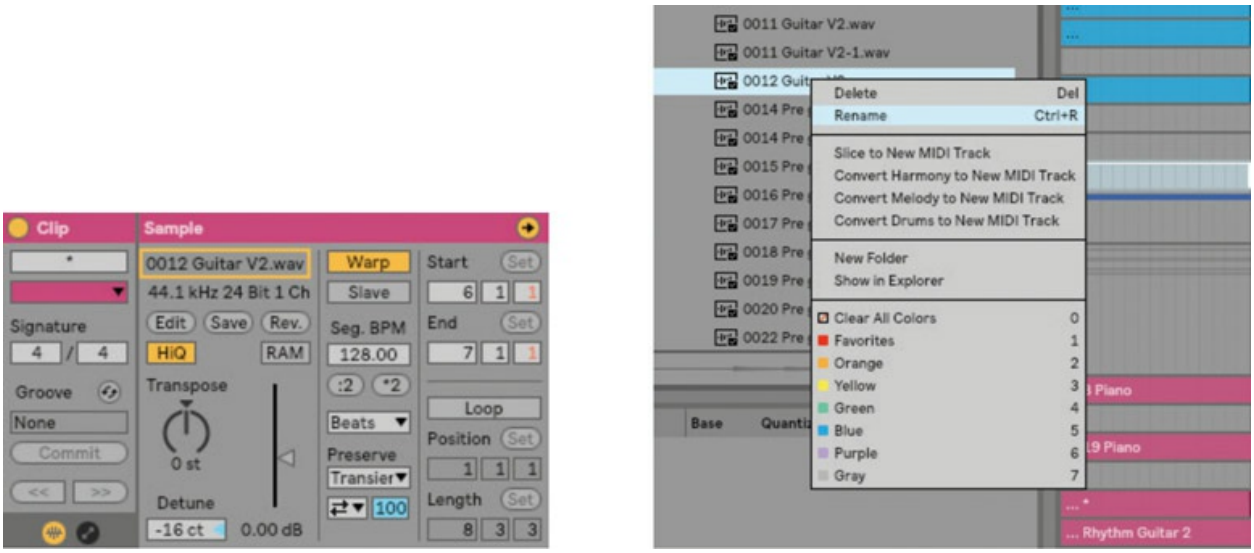


Figure 7.14 Clicking an audio file name in the Clip View (left) selects it in the Browser (right), where it can quickly be renamed

Renaming an audio file may prevent other projects or programs that reference the file from being able to locate it. Use this feature with caution for files in loop libraries and sample collections.

Removing Audio Clips and Deleting Audio Files

Ableton Live makes an important distinction between removing clips from a Set and deleting files from disk:

- When you remove a clip from a Live Set, the parent audio file remains on the storage drive and can be used in other clips elsewhere in the Live Set or in other Live projects.
- When you delete an audio file from disk, clips referring to that file will no longer be able to locate it and will not produce any sound.

Removing Audio Clips

To remove audio clips from your Live Set:

1. Select one or more clips on tracks in Arrangement View or in Clip Slots in Session View.
2. Press the **DELETE** key, or right-click the selection and choose **DELETE**

from the pop-up menu.

Removing audio clips from a set does not delete the audio files, so this action will have essentially no effect on the drive space used for the project.

Deleting Audio Files

As you work on your Live Set, you may accumulate unwanted audio files from test recordings or unusable takes. You might want to delete these unneeded audio files to clean up your Live project and free up drive space.



Deleting files from disk will permanently remove them from your system. This can affect other Live Sets that reference the same audio files.

To remove or delete audio files associated with the current Live Set from your drive, do the following:

1. In the Ableton Live Browser, click **CURRENT PROJECT**.
2. Select the audio file(s) you wish to remove from the Live Set and your storage drive.
3. Right-click the selection and choose **DELETE**. A confirmation dialog box will appear.
4. Select **YES** in the dialog box to send the selected audio file(s) to the trash.

Finding Unused Audio Files

Another option is to use Ableton Live's File Manager to isolate the audio files that are contained within the current Live project but not used by the current Live Set.



Using this process doesn't guarantee that the identified files are not used by a *different* Live Set. Use caution when deleting files with this method.

To search for unused audio files, do the following:

1. Select **FILE > MANAGE FILES** to display Live's File Manager on the right side of both the Session View and Arrangement View.

   You can also display the File Manager by choosing VIEW > FILE MANAGER.

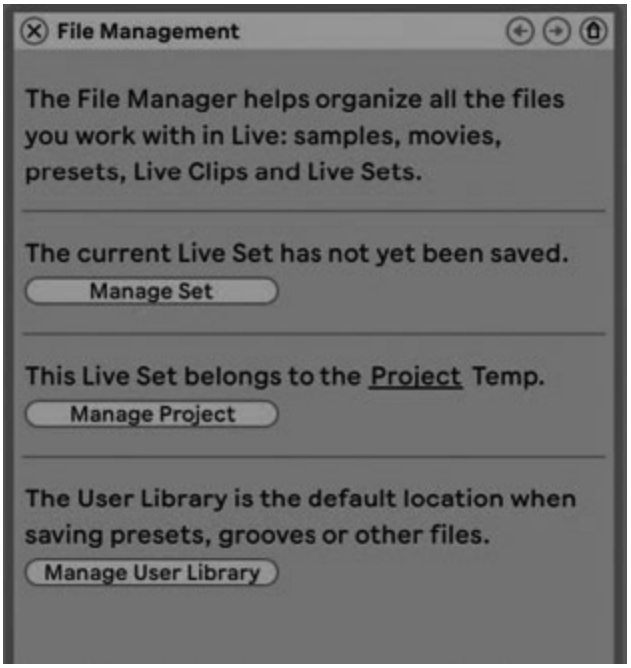


Figure 7.15 The File Manager display in Ableton Live

2. Click the **MANAGE PROJECT** button in the File Manager. The display will change to show information and tools for managing the current project.

3. In the Unused Files section, you'll see a report of the number of audio files that are not currently in use in the current Live Set, as well as the amount of disk space they consume.

   If needed, click the disclosure triangle next to Unused Files to expand the list.

4. Click the **SHOW** button for the relevant type of audio file—in this case, recordings. The unused files will display in the Browser on the left-hand

side of the screen.

5. Select the audio file(s) you wish to remove from the Live Set and your storage drive.

6. Right-click the selection and choose **DELETE**. A confirmation dialog box will appear.

7. Select **YES** in the dialog box to send the selected audio file(s) to the trash.

Review/Discussion Questions

1. Aside from the length of the recording and the number of channels (mono or stereo) being recorded, what two factors ultimately determine the amount of disk space consumed by an audio recording? (See “Audio Storage Requirements” beginning on page 186.)

2. How can you enable the metronome in a Live Set? What are some of the metronome settings you can customize? (See “Enabling the Metronome” beginning on page 188.)

3. How do you route an external sound source into the input of an audio track in a Live Set? (See “Selecting Input Type and Input Channel” beginning on page 190.)

4. How can you adjust the input level of a sound source on an armed track? Can you use the Track Volume slider to affect the strength of the incoming signal? (See “Input Level” beginning on page 193.)

5. How do you arm an audio track for recording in the Session View and Arrangement View? (See “Arming (Record-Enabling) Tracks” beginning on page 192.)

6. What function key can be used as a shortcut for triggering the Arrangement Record button? (See “Recording and Managing Audio” beginning on page 194.)

7. Where are recorded audio files stored within the Live project? (See “Organize Audio Files and Clips” beginning on page 197.)

8. How do audio clips differ from audio files? (See “Working with Clips” beginning on page 198.)

9. How can you rename a recorded audio file from within Ableton Live? (See “Changing File Names” beginning on page 199.)

10. How would you go about removing audio files that are associated with your project but unused in the current Live Set? (See “Deleting Audio Files” beginning on page 200.)

11. Which Ableton Live view can be used to locate unused audio files? (See “Finding Unused Audio Files” beginning on page 201.)

12. Why should you use caution when deleting unused audio files from a project? (See “Finding Unused Audio Files” beginning on page 201.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 7

Recording Audio

Activity

In this exercise, you will be recording a new part to accompany the tracks you have set up in previous exercises. You have two options available to complete this exercise.

Option 1: Live Recording. For this option, you will first need to connect an external sound source, such as a microphone, guitar, or other instrument, to your audio interface. If you do not have an audio interface, you may be able to connect a microphone directly to a computer input or USB port, or use the computer's built-in microphone, if available. Don't worry too much about the quality of the recording—at this time, the goal is to understand how to route a signal into an audio track and capture a recording.

This exercise does not include instructions for routing audio from a USB microphone into Ableton Live.

Refer to the section in Chapter 2 called "Accessing Connected Audio Devices" for basic setup information for audio interfaces and USB microphones. Consult the documentation that came with your device for additional details.

Option 2: Internal Recording. If the above live recording options are not available or are impractical, you can instead record from an existing track in your Live Set to a new audio track.

Duration

This exercise should take approximately 10 to 15 minutes to complete.

Goals/Targets

- Configure a Set for audio recording
- Record an external audio source or record internally

Getting Started

You will start by opening the Live Set you completed in Exercise 6. If that file is not available, you can use the Ex06 Sample file in the 01. Completed Exercises folder within the Media Files 2019-Live101 folder.

To re-download the media files, go to www.halleonard.com/mylibrary and enter the access code printed on the opening page of this book. Then click the DOWNLOAD link for Media Files 2019-Live101 listing in your My Library page.

Open the set and save it as Exercise 7:

1. Do one of the following:
   - Open the set file that you created in Exercise 6 (Storage Drive/Folder > Live101-XXX > Exercise06-XXX.als).
   - Alternatively, you can use the Ex06 Sample file (Media Files 2019-Live101 > 01. Completed Projects > Ex06 Sample.als).
2. Choose **FILE > SAVE LIVE SET AS** and name the Set *Exercise07-XXX*, keeping the Live Set inside the original Project folder. (Move the Set into your Live101-XXX folder if working from the sample file.)
3. Create a new audio track to use as a record track. Choose **CREATE > INSERT AUDIO TRACK**, or use the keyboard shortcut **COMMAND+T** (Mac) or **CTRL +T (Windows).**

Live Recording (Option 1)

Before getting too much further, think about what you plan to record. Since we have already created the foundation of a song segment, you can either record another layer to accompany the four existing tracks, or record a new part to replace one of the existing tracks. Remember that this musical idea is rooted in the key of C Major, so the notes C, D, E, F, G, A, and B will sound most natural.

Configure the Set

If you plan to capture a live recording from an external sound source, you will need to route its signal to the input of your new audio track.

Route the external signal to the input of the new audio track:

1. Locate the **IN/OUT** section of the new track you created.
2. Select the **INPUT TYPE** (in the case of an external audio interface, select Ext. In).
3. Select the appropriate **INPUT CHANNEL**.
4. Select the **AUTO** Monitoring mode for the audio track. (See Figure 7.16.)

In Auto mode, the incoming signal will pass through to the output while tracking so you can hear what is being recorded onto the track in your headphones or monitor speakers. During playback, monitoring will switch to play the track content, making it easy to listen to what you've recorded.



Figure 7.16 The In/Out settings for recording an external sound source

If you are using your computer's built-in audio for recording and playback, enabling the Auto Monitoring mode may lead to feedback. In this case, use Off mode instead.

5. Verify that the input displayed on the **INPUT CHANNEL** chooser matches the input that your sound source is plugged into on your audio interface.

6. If necessary, click the **INPUT CHANNEL** chooser to make changes, selecting the correct input channel from the drop-down menu. (Note: Stereo sources will have a pair of inputs routed to the track.)

Check Levels

Once you have your sound source routed to an audio track, it's time to check the level of the signal to ensure you're receiving a strong signal, but not exceeding 0 dBFS and introducing digital clipping.

Monitor the incoming signal and check the level:

1. Arm the audio track for recording.

2. Test the input level from your external sound source by playing the connected instrument, speaking or singing into a connected microphone, or otherwise activating the incoming signal.

3. Adjust the strength of the incoming signal, as needed.

Remember, the Track Volume slider does not affect the strength of the input signal. If your input signal is too strong or too soft, you need to adjust the instrument volume, microphone position, or audio interface gain.

Be careful not to push the meter past 0 dBFS and into the red zone, as this could lead to irreversible clipping in the recorded audio file.

Record Your Performance

Once you've tweaked your sound source and arrived at a good input level, it's time to try recording the part to your audio track. If you plan to replace one of the existing tracks with your recording, you might want to disable/mute the track before recording.

To mute a track, click the Track Activator button below the pan control in the Session View or left of the Solo button in the Arrangement View.

Record in Arrangement View

Record the performance to an audio track in Arrangement View:

1. First, name the record track appropriately; for example, Guitar or Vocal.
2. Click the **METRONOME** switch to enable the metronome.
3. Click the down arrow next to the Metronome switch and select a 2-bar count-in (or more, if needed).



Figure 7.17 Setting the Metronome Count-In to 2 Bars

4. Position your Arrangement Insert Marker where you would like the

recording to begin.

5. Click the **ARRANGEMENT RECORD** button or press function key **F9** to initiate recording.

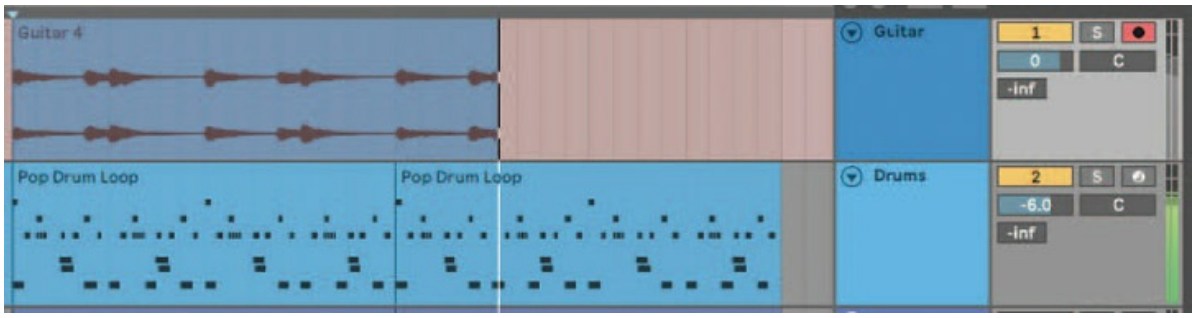


Figure 7.18 Recording audio in the Arrangement View

6. After completing the record pass, press **SPACEBAR** to stop the recording.

Repeat this process, if necessary, until you are satisfied with your results. If you don't want to delete or overwrite a recorded take but want to perform the part again, simply move the clip to a later point in the arrangement.

Record in Session View

Record the performance to an audio track in Session View:

1. First, verify that the record track has been named appropriately; for example, Guitar or Vocal.

2. Click the **METRONOME** switch to enable the metronome.

3. Click the down arrow next to the Metronome switch and select a 2-bar count-in (or more, if needed).

4. From the Quantization Menu, select a value of 1 Bar.



Figure 7.19 Setting the Quantization value to 1 Bar

5. Click the record button on an empty clip slot. If you want to play along with other clips in the Live Set, you can manually launch them by pressing their Clip Launch buttons before performing the recorded part.

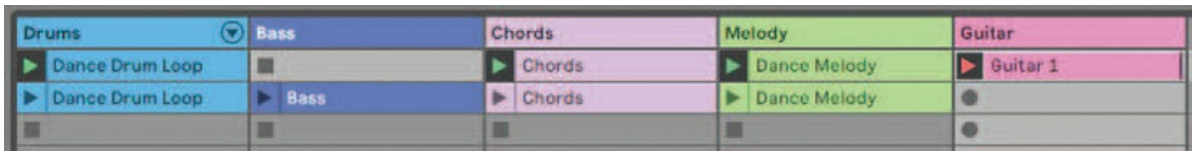


Figure 7.20 A red Clip Launch Button indicates that a track is currently recording

6. As the recording begins the fourth bar, click the **PLAY** button on Scene 1. This will launch the scene, and conclude the record pass. The quantization setting of 1 bar will guarantee that the length of the recording is an exact amount in full bars (such as 4 bars).

Repeat this process, if necessary, until you are satisfied with your results. You can either delete a record take from the selected clip slot, or move on to another slot for subsequent recordings. When you settle on a take, move that clip to the scene containing the other elements of the song.

Internal Recording (Option 2)

If you plan to record internally, you will need to route the output of an existing track to the input of your new audio track. To complete this part of the exercise, you will use the Melody track that you created Exercise 6 as the source of the recording.

Configure the Set

Route the existing track to the input of the new audio track:

1. Name the audio track Melody Recording.
2. Locate the **IN/OUT** section of the Melody Recording audio track.
3. From the **INPUT TYPE** chooser, select the Melody MIDI track. This routes the output of the existing Melody track to the new audio track.
4. Set the Monitor Mode of the Melody Recording track to **AUTO**.



Figure 7.21 The In/Out settings and Monitor Mode for internal recording

5. Arm the Melody Recording audio track for recording.

Record the Audio to the New Track

Now that you've routed the existing track to your new audio track, you're ready to record.

Record in Arrangement View

Record the existing track to your new audio track in Arrangement View:

1. In Arrangement View, position the Arrangement Insert Marker at the location where you want to begin recording from the Melody track.

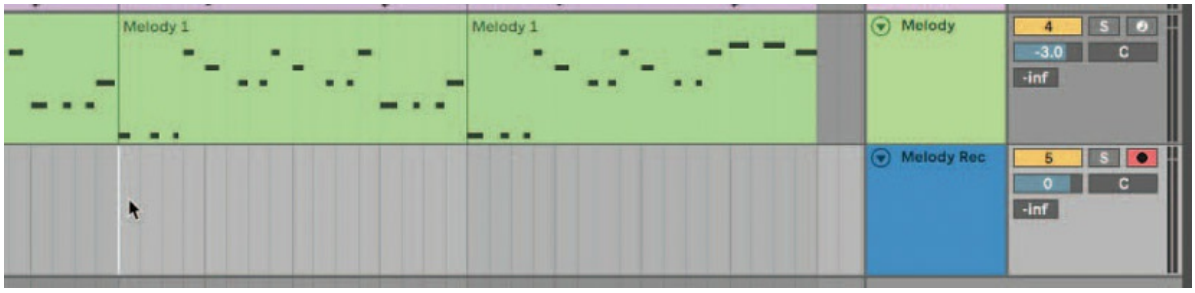


Figure 7.22 Placing the Arrangement Insert Marker

2. Click the **ARRANGEMENT RECORD** button or press function key F9.

3. After completing the record pass, press **SPACEBAR** to stop the recording.

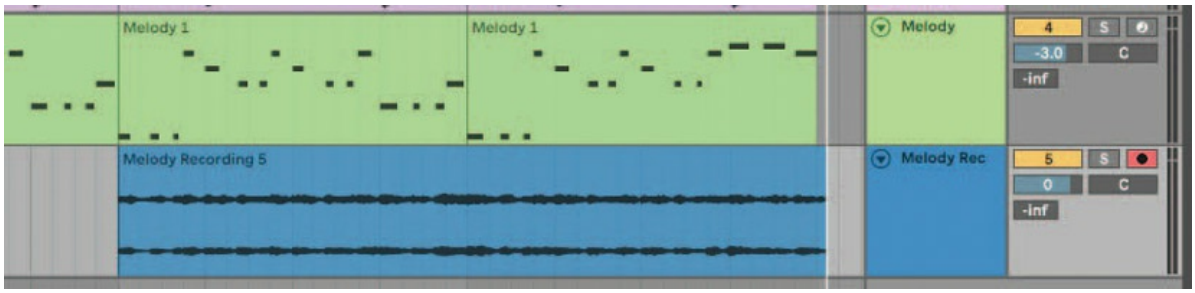


Figure 7.23 The completed record pass in Arrangement View

Record in Session View

Record the existing track to your new audio track in Session View:

1. From the Quantization Menu, select a value of **1 BAR**.

2. Open the Ableton Live Preferences window by selecting **LIVE** > **PREFERENCES** (Mac) or **OPTIONS** > **PREFERENCES** (Windows).

3. Under the **RECORD/WARP/LAUNCH** tab, enable the **START RECORDING ON SCENE LAUNCH** option.

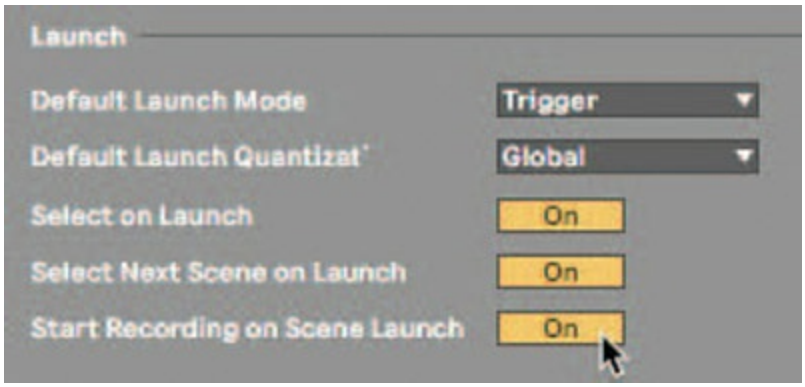


Figure 7.24 Enabling the Start Recording on Scene Launch preference

4. Close the Preferences window when finished.

5. Click the **PLAY** button on Scene 1 of the Master track. This will simultaneously launch the clips on the scene and initiate recording on the Melody Recording track.

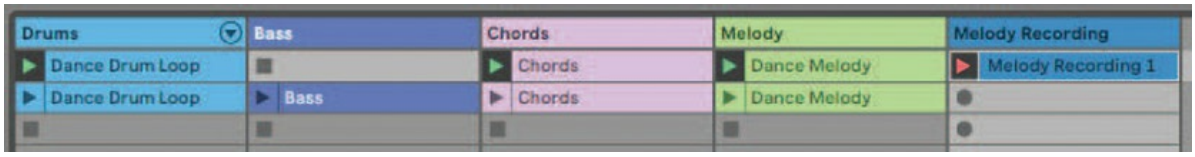


Figure 7.25 A red Clip Launch Button indicates that a track is currently recording.

6. As the recording begins the fourth bar, click the **PLAY** button on Scene 1 again. This will re-launch the scene, and conclude the record pass. The quantization setting of 1 bar will ensure that the length of the recording is an exact multiple of full bars.

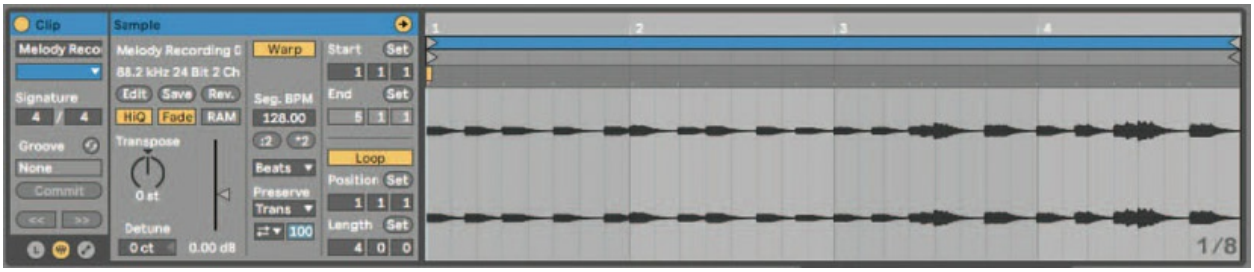


Figure 7.26 The recorded clip is exactly four bars long.

After completing the internal recording, you can deactivate/mute the Melody MIDI track, as it is essentially a duplicate of the audio recording you just created.

To mute a track, click the Track Activator button below the pan control in the Session View or left of the Solo button in the Arrangement View.

Finishing Up

To complete this exercise, disarm your record track(s) so that you don't overwrite any of your takes. After that, you can review your work and save the Live Set.

Review your work in the Arrangement View:

1. Position the Arrangement Insert Marker at the start of the clips.
2. Press the **SPACEBAR** to play back the Live Set and confirm your results.
3. Press the **SPACEBAR** a second time when finished.

Review your work in the Session View:

1. Click the **PLAY** button on Scene 1 of the Master track to play back the Live Set and confirm your results.
2. Press the **SPACEBAR** or the **STOP ALL CLIPS** button on the Master track to stop playback when finished.

Save your work and close the Live Set when finshed:

1. Choose **FILE > SAVE LIVE SET**.
2. Press **COMMAND+Q** (Mac) or **CTRL+Q** (Windows) to quit Live and close the project.

Chapter 8

Selection, Navigation, and Configuration

This chapter covers various selection and navigation techniques that can add efficiency to your workflow in Ableton Live. It includes descriptions of how to quickly navigate even complex Live Sets, make different types of selections on tracks in the Arrangement View, customize the order and size of tracks, and work with locators to organize and add notes to an arrangement.

Learning Targets for This Chapter

- Navigate an arrangement with the Overview
- Make selections in tracks based on clips and time
- Quickly zoom and scroll through an arrangement
- Customize the order and size of tracks
- Add, delete, and work with locators

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform. Sign up for free at ElementsED.com.

Understanding how to navigate, select content in, and customize the display of a Live Set can dramatically speed up your production workflow. Whether you need to make broad edits or zoom in and isolate a specific moment, being able to quickly locate and select material is the key. In this lesson, you will learn to streamline your creative process with methods for quickly navigating, selecting content, and customizing the layout of your Live Set.

How you navigate a Live Set will vary, depending on whether you are working in the Session View or Arrangement View. Whereas the Session View displays tracks and clips on a non-linear grid, the Arrangement View displays tracks and clips along a linear timeline.

Using the Arrangement Overview

The Arrangement View is meant to display an entire—and perhaps finished—piece of music, and thus offers a series of unique navigation features and techniques. One of those features is the Overview, which is essentially a bird's-eye view of an arrangement.

The Overview is located at the top of the Arrangement View or Session View and shows all MIDI, audio, and video material on tracks in the Arrangement. Clips are represented by narrow, horizontal lines, in the same color as the clips on the tracks in Arrangement View.

To toggle the display of the Overview, do one of the following:

- Choose **VIEW > OVERVIEW**.
- Press **OPTION+COMMAND+O** (Mac) or **CTRL+ALT+O** (Windows).

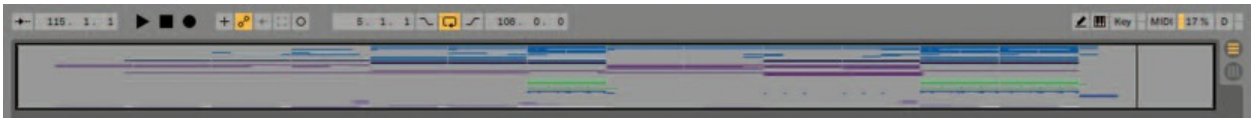


Figure 8.1 The Overview situated at the top of the Arrangement View

Ableton Live 10.1 introduced the ability to change the height of the Overview by clicking and dragging at the bottom of the

Overview area.

The Overview Outline (Arrangement View only)

When working in the Arrangement View, you can use the Overview as one method to navigate and zoom an arrangement. This can be very useful when you are working with complex or lengthy pieces of music. The Overview displays a black outline (rectangle) surrounding the entire Overview area (when zoomed out completely) or framing a portion of the Overview (when zoomed in). This outline represents the part of the arrangement that is currently displayed in the Arrangement View. If you change the display of the Arrangement View by zooming or scrolling, the framed outline area in the Overview will relocate and resize accordingly. During playback, if the Follow behavior is set to scroll, the outline area in the Overview will also scroll.

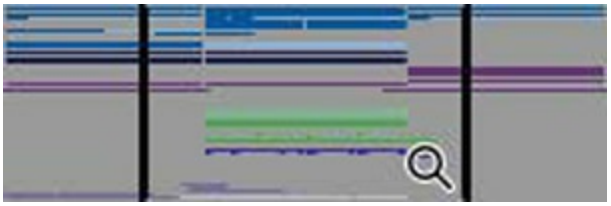


Figure 8.2 The Overview outline with the Arrangement View zoomed in on a specific song segment

Zooming and Scrolling with the Overview (Arrangement View Only)

The Overview offers a simple way to zoom and scroll a Live Set.

To zoom and scroll using the Overview, do the following:

1. Click anywhere in the Overview area to move the framed outline to the selected location, scrolling the Arrangement View to that location.

2. Do one of the following:

    - To zoom in or out on an arrangement, click within the black Outline

area in the Overview and drag up or down (or click and drag either edge to resize the rectangle).

- To scroll left or right in an arrangement, click within the black Outline area and drag left or right.

- To zoom out completely, double-click anywhere within the Overview Outline.

You can also zoom by holding COMMAND (Mac) or CTRL (Windows) and scrolling with the mouse wheel or trackpad while hovering over the Overview.

Making Selections in the Arrangement View

Making selections allows you to quickly edit material on audio, MIDI, and video tracks, as well as the contents of Group Tracks. Being able to quickly make selections will allow you to perform editing tasks with ease and efficiency. Most of the track-based selections techniques take place within the Arrangement View.

This chapter covers selections of clips and time on any kind of track. Selections within clips and automation selections are covered in greater detail in later chapters.

Selecting Clips and Time

There are two distinct methods of making selections when you are working in the Arrangement View. You can select *clips* or you can select spans of *time*. Once you have made a selection of either type, you can perform any action available under the **EDIT** menu.

Selecting Clips

Making selections often starts with selecting audio, MIDI, or video clips. To select a clip, click within the top half of the clip, where the hand icon appears (see Figure 8.3).

When a track display is folded (collapsed), you can click anywhere on a clip to select it.

Information on folding and unfolding tracks is provided in the section on "Adjusting Your Live Set" later in this Chapter.

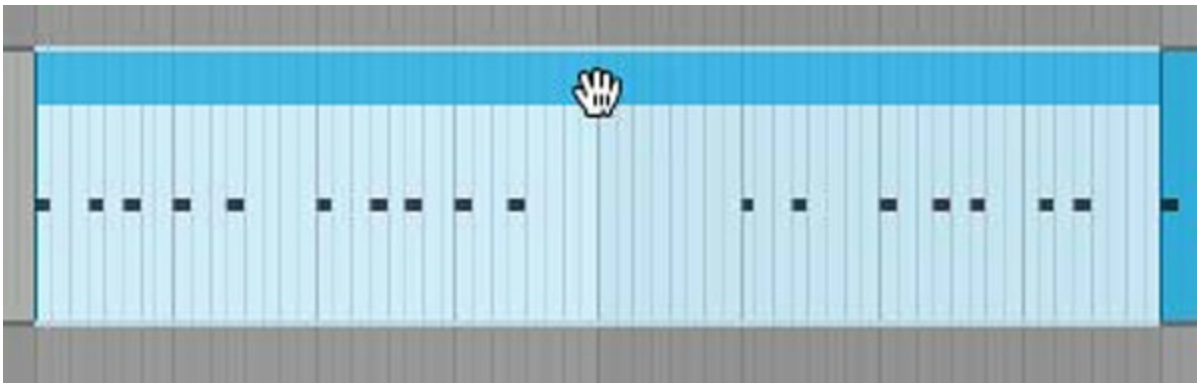


Figure 8.3 Selecting a single clip on an unfolded track using the solid colored bar at the top of the clip

To add clips to an existing selection, hold **SHIFT** and click on another clip with the hand icon. This will select everything between the existing and new selections, even across multiple tracks.

Selecting Time

Selecting time allows you to perform broad editing tasks, such as duplicating a song segment, as well as making selections *within* a clip's boundaries. This can be accomplished by clicking and dragging within the bottom half of a clip with the arrow pointer icon or clicking and dragging from anywhere on a track outside of a clip.

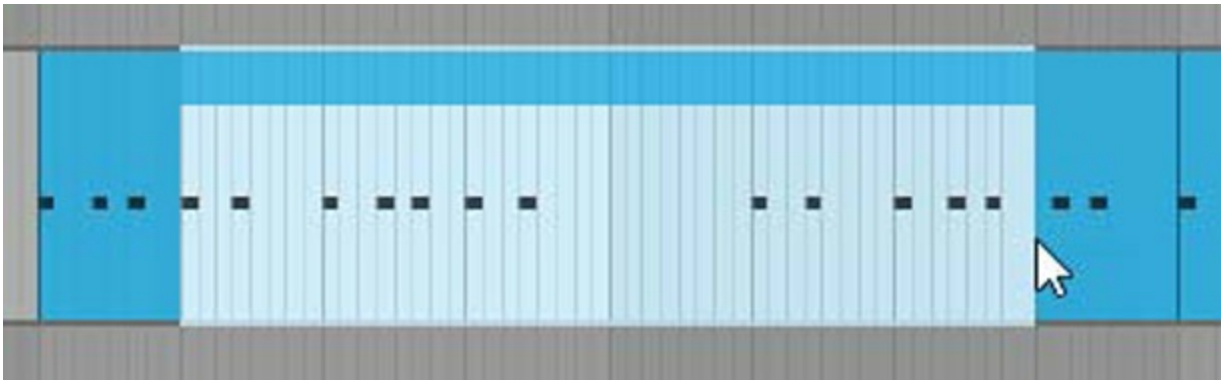


Figure 8.4 Selecting time within clip boundaries

To add time to an existing selection, hold **SHIFT** and click at another point in time with the arrow pointer icon. This will extend the selection to include everything between the existing and new selections, even across multiple tracks.

When you hover your mouse cursor over a selection, Live will display the start and end locations, as well as the selection length, in the Status Bar at the bottom of the Live interface.

Selections On and Off the Grid

By default, when you click and drag within the Track Display, your selection start and end will snap to the current grid. This is often desired behavior, but at times you may want to make a more precise selection. To prevent a selection from snapping to the grid, hold **COMMAND** (Mac) or **ALT** (Windows) while clicking and dragging a selection.

Multi-Track Selections

You can make selections across any number of available tracks in your Live Set. Ableton Live provides several methods of doing this.

To make a selection across adjacent tracks, do one of the following:

- Click and drag horizontally and vertically across multiple tracks. (Do not click and drag a clip, as that will move the clip to another location and track.)
- Make a selection on one track, hold **SHIFT**, and press the **UP** or **DOWN ARROW** keys to extend the selection across additional tracks.
- Make a selection on one track, hold **SHIFT**, and click at the end location on another track to make a contiguous selection across all tracks in between.

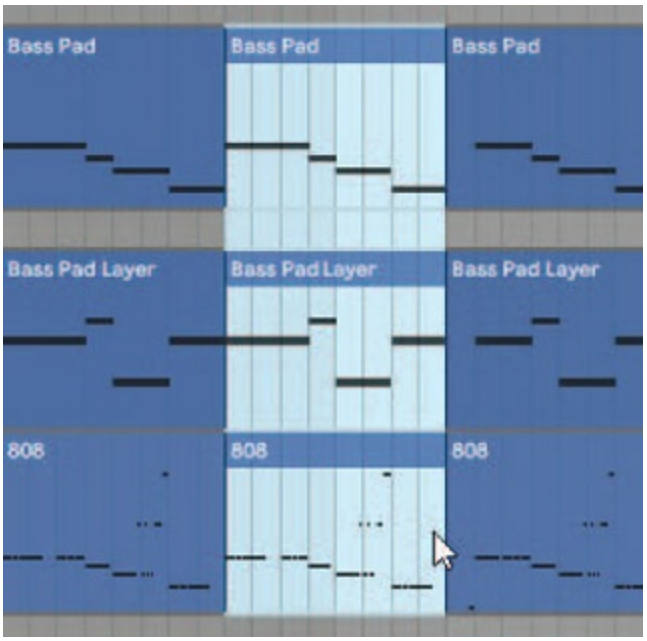


Figure 8.5 A selection across adjacent tracks

To make a selection across non-adjacent tracks, do the following:

1. Make a selection on one track.

2. While holding **COMMAND** (Mac) or **CTRL** (Windows), click on another non-adjacent track with the arrow pointer icon. The selection will be added to the second track non-contiguously.

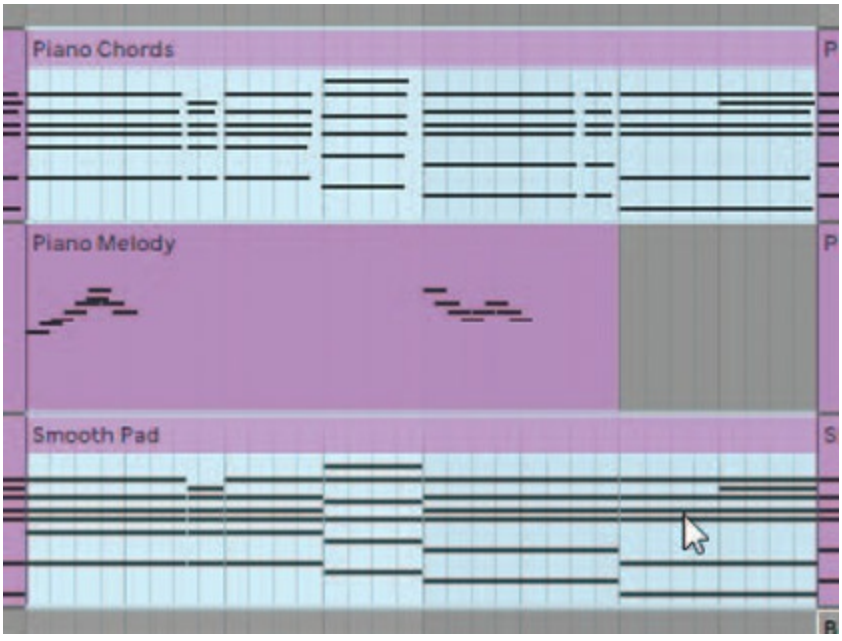


Figure 8.6 A selection across non-adjacent tracks

For details on making selections within an Automation envelope, see the "Automation" section in Chapter 10 of this book.

Zooming and Scrolling the Arrangement View

The Arrangement Overview, discussed at the start of this chapter, is a useful tool for quickly determining your position within a more complex arrangement, as well as locating and zooming into other areas. However, there are many other methods of navigating and zooming in the Arrangement View, and they may provide additional efficiency for your production process.

Zooming In and Out

One of the quickest methods of zooming into a specific location is done using the beat-time ruler.

To zoom using the beat-time ruler, do the following:

1. Hover your pointer over the beat-time ruler. The cursor will change into a magnifying glass.

2. Click the location you would like to zoom into and drag up or down to zoom out or in, respectively.

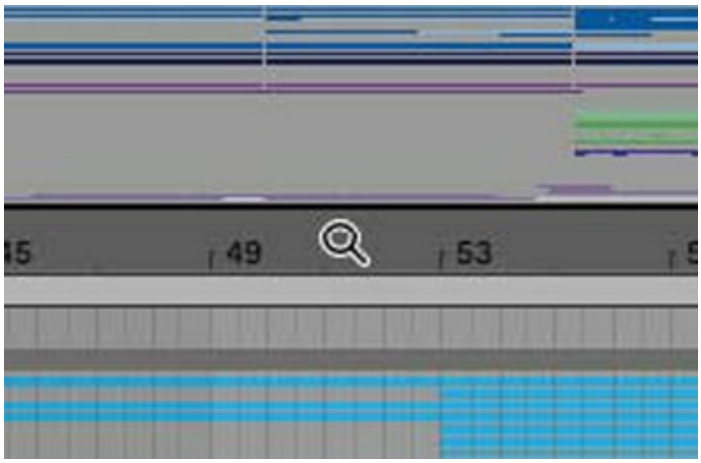


Figure 8.7 Zooming using the beat-time ruler at the top of the Arrangement View

You can also zoom on your current mouse cursor location by scrolling the mousewheel or trackpad up and down while holding COMMAND (Mac) or CTRL (Windows).

Zooming In and Out on Selections

Whether you have a clip selected or simply have the Insert Marker positioned in a specific location, you can zoom in on the location using the keyboard.

To zoom using the keyboard, do the following:

1. Place the insert marker on a track or make a selection.

2. Press the **PLUS** (+) or **MINUS** (–) keys to zoom in or out on the location or selection.

You can zoom to fit the current selection to the Arrangement View by double-clicking the beat-time ruler. If nothing is selected, this action will zoom out to show the entire length of the arrangement.

Zooming to Fit the Screen

At times, you may need to zoom a selection such that it fills the screen horizontally. One way to do this is to select **VIEW > ZOOM TO ARRANGEMENT TIME SELECTION**, or press the **Z** key.

To zoom back out from this view, press **X** or select **VIEW > ZOOM BACK FROM ARRANGEMENT TIME SELECTION**. Once you are zoomed out again, you can also use the X shortcut to show the entirety of an arrangement.

Ableton Live 10.1 introduced the ability to use the W key to zoom out to show the entirety of an arrangement, and the H key to zoom track heights to fit the screen.

The above shortcuts will not work when the Computer MIDI Keyboard switch is active in the Arrangement View.

To toggle the Computer MIDI Keyboard on/off, press the M key.

You can also zoom a selection to fit the screen by double-clicking the beat-time ruler or by Right-clicking within a selection and choosing Zoom to Arrangement Time Selection from the pop-up context menu.



Figure 8.8 Zooming a selection to fit the screen using the right-click menu

Scrolling

Ableton Live provides multiple methods of scrolling the Arrangement View to the left and right.

To scroll the Arrangement View, do one the following:

- Click and drag the beat-time ruler (at the top of the Track Display) to the left or right.

- Click and drag the Time Ruler (at the bottom of the Track Display) to the left or right.

- Hold **OPTION+COMMAND (Mac) or CTRL+ALT** (Windows), click anywhere in the Track Display, and drag left or right.

Adjusting Your Live Set

Live offers several methods of organizing your project and customizing the display of individual tracks. While these are subtle optimizations to your Live Set, they have the potential to make your production and editing processes faster.

Changing the Track Order

Ableton Live allows you to change the order of tracks in your Live Set at any time. Arranging tracks in a logical order can speed up your workflow and project navigation. You might consider arranging the tracks in your Live Set such that related tracks are displayed together, instruments are displayed in a logical order, or commonly used tracks are located at the left (Session View) or the top (Arrangement View).



Any changes you make to the track order in the Session View will also affect the display in the Arrangement View, and vice versa.

To change the track order, do the following:

- In the Session View, click the name of a track you wish to move and drag it left or right. The destination location will be represented by a vertical black line as you drag.

- In the Arrangement View, click the name of the track and drag it up or down. The destination location will be represented by a horizontal black line as you drag.

To move multiple tracks simultaneously, select the target tracks

prior to dragging.

Hold the SHIFT key while clicking to select multiple adjacent tracks or hold COMMAND (Mac) or CTRL (Windows) while clicking to select non-adjacent tracks.

Folding and Unfolding Tracks

Each track in the Arrangement View includes a **FOLD/UNFOLD TRACK** button to the left of the track name. Clicking this button will fold the track display in and out.



Figure 8.9 The Fold/Unfold Track button on an audio track (shown unfolded)

Folding a track collapses the track display to its minimum size, allowing you to fit more tracks in the Arrangement View simultaneously.

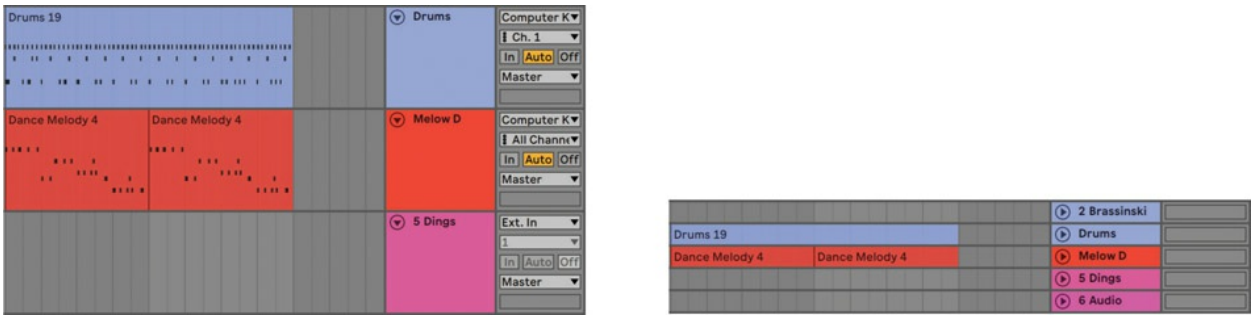


Figure 8.10 Tracks shown unfolded (left) and folded (right)

To toggle a track display between a folded and unfolded state, do one of the following:

- Click the triangular **FOLD/UNFOLD TRACK** button next to the track name.

- Select the track and press the **Left and Right Arrow** keys on your computer keyboard to fold and unfold the track, respectively.



Hold Option (Mac) or Alt (Windows) while clicking on the Fold/Unfold Track button for any track to fold/unfold all tracks in the Set simultaneously.

Adjusting Track Size

Live allows you to customize the size of each track in both the Session View and Arrangement View. Larger track sizes are particularly useful for precision editing because they show more detail. Smaller track sizes are useful for conserving screen space in large projects.

Adjusting Track Width in Session View

In the Session View, tracks are displayed as vertical strips. In this view, you can change the width of individual tracks as needed.

To expand a track in the Session View, do the following:

- Click and drag the track border to the right of the track name. In the Session View, making tracks wider can be useful to view long clip names on Clip Slots; making tracks narrower is useful for reducing track size and freeing up screen real estate.

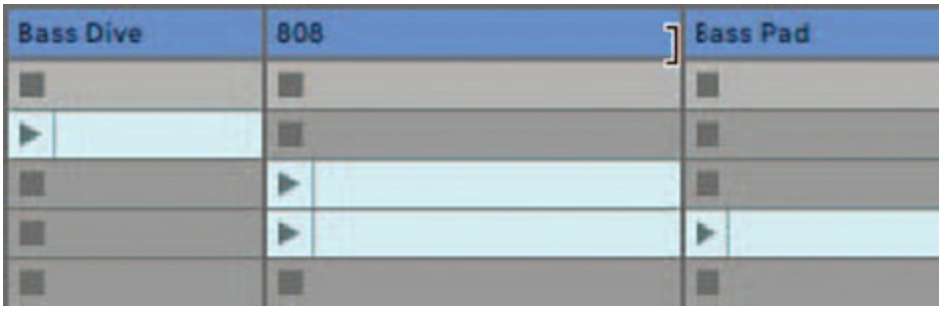


Figure 8.11 When changing the track size in the Session View, the cursor turns into a bracket icon

Adjusting Track Height in Arrangement View

In the Arrangement View, tracks are displayed as horizontal strips along the timeline.

To expand a track in the Arrangement View, do the following:

1. Position your pointer over the lower boundary of any track in the Mixer area; the cursor will change into a double-headed arrow.

2. Click on the track boundary and drag up or down. The track height will change as you drag.



Figure 8.12 Changing track height in the Arrangement View

To change the height of all tracks simultaneously, hold OPTION (Mac) or ALT (Windows) while dragging the track boundary of any track.

To change track height using a mousewheel or trackpad, select the target track(s) and hold OPTION (Mac) or ALT (Windows) while scrolling up or down with the mouse cursor anywhere in the Track Display.

You can also resize all tracks in a Live Set by selecting **VIEW > SHOW ALL TRACKS** or by pressing the **H** key. This will fit as many tracks as possible into the Track Display by folding them into their most condensed form.

The H key shortcut will not work when the Computer MIDI Keyboard switch is active in the Arrangement View.

Working with Locators (Arrangement View Only)

Locators can be used to bookmark important locations in your arrangement, making it easy to organize song sections, create notes on important hit points when scoring or sound designing with video, or simply leave reminders of ideas to keep your creative process flowing. The following sections describe how to add, edit, recall, and delete locators, and how to create selections and loops using locators.

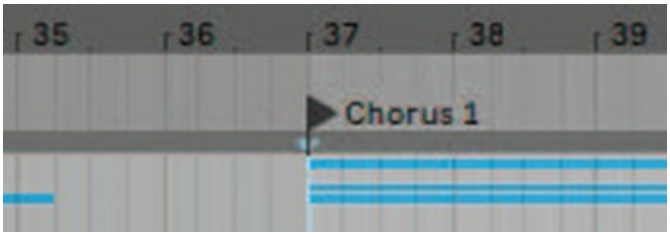


Figure 8.13 A “Chorus” locator at bar 37 of an arrangement

Adding Locators

Locators can be added at any point in an arrangement.

To add a locator, do one the following:

- Click the **SET LOCATOR** button next to the beat-time ruler. This can be done in real-time during recording or playback, or while the transport is stopped. A new locator will be added at the playhead location, Arrangement Insert Marker location, or selection start location.

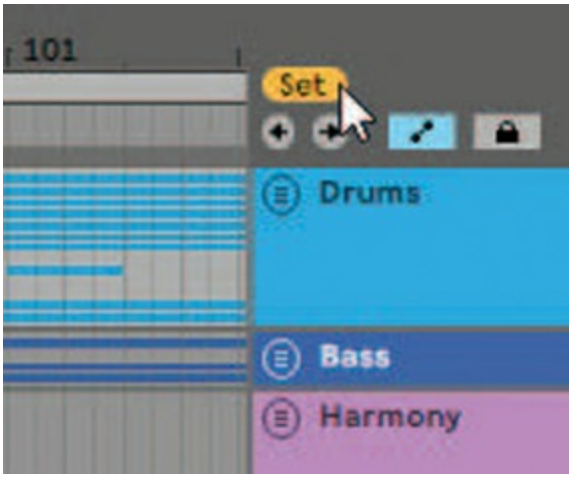


Figure 8.14 Adding a locator at the currently selected location using the Set

button

- Right-click in the scrub area and choose **ADD LOCATOR** to add a locator at the selected location.

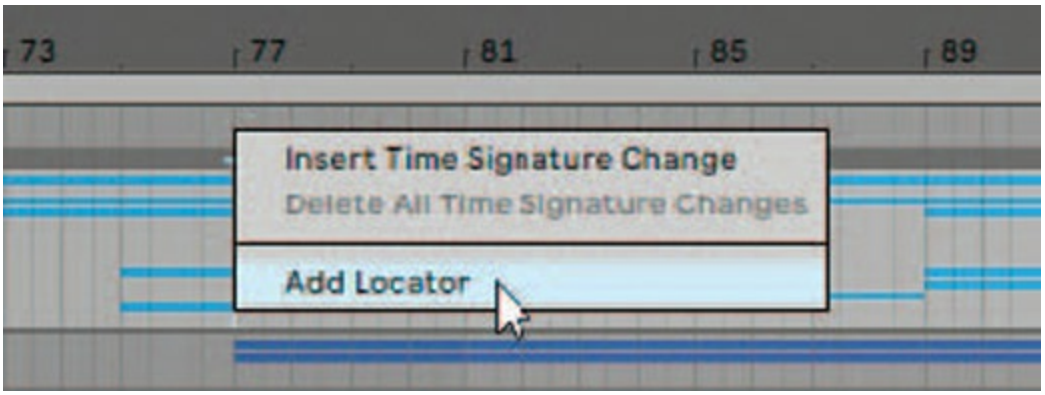


Figure 8.15 Adding a locator by right-clicking in the scrub area below the beat-time ruler

- Select **CREATE > ADD LOCATOR** to add a locator at the Insert Marker position.

When you add a locator, it will be quantized according to the Quantization setting specified in the Control Bar. This is done to keep locators on musical divisions; however, you can freely edit the position of the locator after adding it, as described in the next section.

With a locator selected or the Insert Marker placed at the position of a locator, the Set Locator button becomes a **DELETE** button for quick removal of the locator.

Editing Locators

Once you've created a locator, you have a few options to move, rename, and add text information to it.

Moving Locators

Locators can be moved by clicking and dragging on the locator arrow in the Scrub Area, or by selecting the locator arrow and pressing the left and right

arrow keys on your computer keyboard.



As is the case with other editing operations in Live, you can hold COMMAND (Mac) or ALT (Windows) while dragging a locator to prevent it from snapping to the grid.

Renaming Locators

Ableton Live provides a few different ways that you can name or rename an exiting locator.

To rename a locator, do one of the following:

- Right-click the locator arrow and select **RENAME** from the pop-up menu.
- Click on a locator arrow to select it and then do one of the following:
  - Choose **EDIT > RENAME**.
  - Press **COMMAND+R** (Mac) or **CTRL +R** (Windows).

Adding Info Text to Locators

In addition to naming your locators, you also have the opportunity to add informational text to each locator in your Set.

To add info text to a locator:

1. Right-click on a locator arrow and select **EDIT INFO TEXT** or, with a locator arrow selected, choose **EDIT > EDIT INFO TEXT**.

   This action will place your cursor in the Info View section of the Live interface.

2. Type additional notes about the locator in the Info View section.

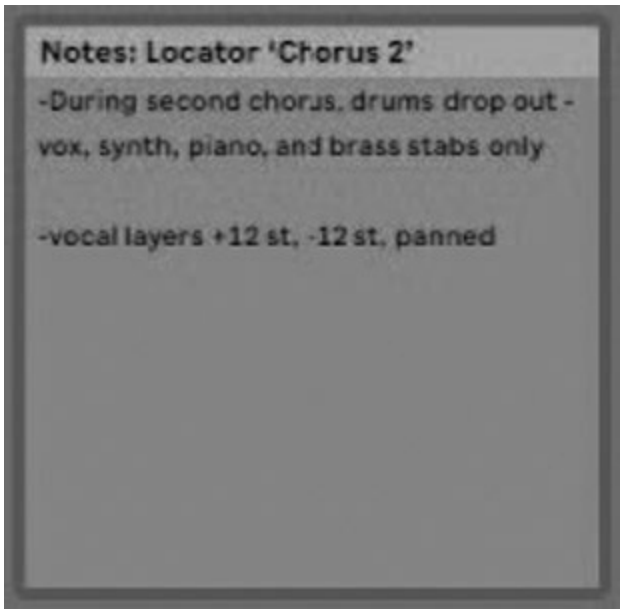


Figure 8.16 You can add notes to a locator using the Edit Info Text option

Recalling and Playing from Locators

Once you have set locators in your Live Set, you can recall the locator positions and begin playback from those locations for quick access to different song sections or components of your project.

To recall a locator position, do one of the following:

- Click on a locator to select it.
- Click the **Previous Locator** or **Next Locator** buttons below the **Set Locator** button to cycle through locator positions sequentially.

Double-click on a locator arrow to begin playback from that position.

Using Locators for Loops and Selections

If you have at least two locators in your Live Set, you can use them to create selections and loop points based on the locator positions.

To create a selection between two locators:

1. Right-click on the first of the two locator arrows where you'd like to

create a selection.

2. Choose **SELECT TO NEXT LOCATOR** from the pop-up menu.

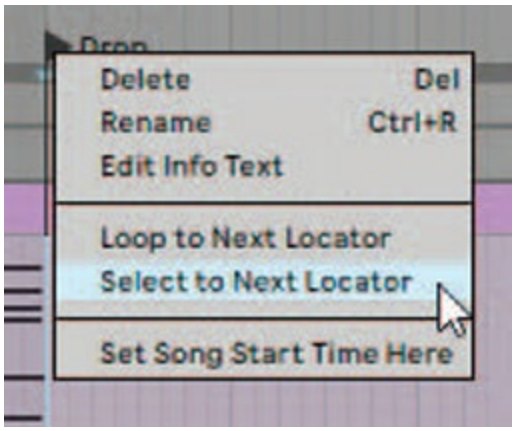


Figure 8.17 Making a selection or loop between two locators

To create loop points between two locators:

1. Right-click the first of the two locator arrows where you'd like to create a loop area.

2. Choose **LOOP TO NEXT LOCATOR**. The loop brace will be added to the Scrub Area, spanning between the two locators.

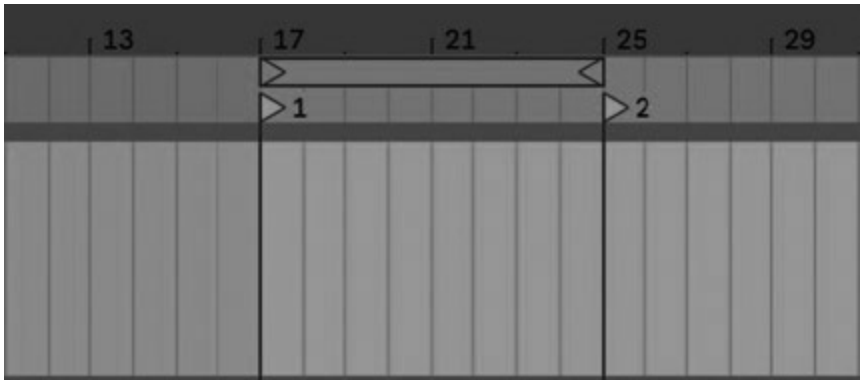


Figure 8.18 The loop brace after creating loop points based on locator positions 1 and 2

Deleting Locators

You can remove locators from an arrangement at any time using one of a variety of commands.

To delete a locator, do one of the following:

- With a locator selected or the Insert Marker at the position of a locator, press the **DELETE LOCATOR** button next to the beat-time ruler.
- Right-click a locator and select **DELETE**.
- With a locator selected, press **DELETE** or **BACKSPACE** on your computer keyboard.
- With a locator selected, choose **CREATE > DELETE LOCATOR**.

Navigating the Arrangement

Clicking in the Track Display area in the Arrangement View selects a point in time, represented by the flashing insert marker. The insert marker can be placed on any track and at any position by clicking at the desired location. The insert marker position determines where playback will begin in Arrangement View.

The insert marker can also be used to navigate within the arrangement using the arrow keys on your computer keyboard.

Navigating Time

The insert marker can be moved forward or backward in time from the keyboard, based on the current Edit Grid settings. This can be useful for repositioning the playback point or edit location without having to reach for the mouse.

To move the insert marker in time:

- Press the **LEFT** and **RIGHT ARROW** keys. Each key press will move the insert marker to the next or previous Edit Grid line.

Navigating Tracks

The insert marker can also be moved vertically between tracks from the

keyboard. This can be useful for moving the edit point up or down through your track. The same technique can be used to move a selection up or down between tracks.

To move the insert marker or selection to an adjacent track:

- Press the **UP** and **DOWN ARROW** keys. Each key press will move the insert marker or selection to the next or previous track in the Arrangement View.

Navigating Through Clips and Locators

At times, you may want to move the insert marker forward to the start of next clip, backward to the end of the previous clip, or to another clip boundary or marked location. Using modifiers with the arrow keys, you can easily navigate horizontally between the clips on a track and any locator positions in a Set.

To move though clip boundaries and locators:

1. Position the insert marker on the track where you wish to navigate.

2. Press **OPTION+LEFT/RIGHT ARROW** (Mac) or **CTRL+LEFT/RIGHT ARROW** (Windows).

   Each key press will move the insert marker to the next or previous clip boundary on the track; the insert marker will also stop at each locator position in the Set.

Making Selections while Navigating

Navigating from the keyboard can also be useful for making selections on a track. You can select horizontally and vertically from the current insert marker location or existing selection by holding **SHIFT** while navigating using any of the above techniques.

To select horizontally using the keyboard:

- Hold **Shift** while pressing the **Left** or **Right Arrow** keys to select forward or backward in Edit Grid increments.

- Hold **Option+Shift** (Mac) or **Ctrl+Shift** (Windows) to select forward or backward to the next/previous clip boundary or locator position.

- Hold **Shift** while pressing the **P** or **Down Arrow** keys to extend the insert marker or existing selection to adjacent tracks.

Review/Discussion Questions

1. What is the purpose of the Overview? How can you use this feature to zoom and scroll an arrangement? (See “Using the Arrangement Overview” beginning on page 216.)

2. How can you make a selection based on a range of time (rather than a selection based on a clip’s boundaries)? (See “Selecting Clips and Time” beginning on page 218.)

3. How can you prevent a selection from snapping to the grid when clicking and dragging? (See “Selections On and Off the Grid” beginning on page 219.)

4. What key modifier allows you to extend a selection to additional, *non-adjacent* tracks? (See “Multi-Track Selections” beginning on page 219.)

5. What are some of the ways you can zoom in and out in the Arrangement View? (See “Zooming In and Out” beginning on page 221.)

6. How do you change the order of tracks in the Session View and Arrangement View? What are some reasons you might want to re-order tracks? (See “Changing the Track Order” beginning on page 223.)

7. What are locators, and why might you use them in an arrangement? (See “Working with Locators” beginning on page 226.)

8. What are some ways to add locators to a Live Set? Can you add locators in real time during recording or playback? (See “Adding Locators” beginning on page 226.)

9. How can you quickly start playback from a locator position? (See “Recalling and Playing from Locators” beginning on page 228.)

10. How can you use locators to make a selection or to designate a loop area?

(See “Using Locators for Loops and Selections” beginning on page 228.)

1. What keys can you use to move the insert marker forward or backward in time from the keyboard? (See “Navigating Time” beginning on page 230.)

2. What keys can you use to move the insert marker forward or backward on a track through clip boundaries? (See “Navigating Through Clips and Locators” beginning on page 230.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 8

Recording, Configuring, and Adding Locators to an Arrangement

Activity

In this exercise, you will extend and configure an arrangement using the techniques discussed in Chapter 8. You will make basic selections, duplicate a segment of an arrangement, and add locators that indicate changes in the arrangement.

Duration

This exercise should take approximately 10 to 15 minutes to complete.

Goals/Targets

- Record from Session View into an Arrangement
- Edit an Arrangement
- Extend an Arrangement
- Configure your Live Set
- Use Locators

Media Files

To complete this exercise, you may need to use various media files included in the Media Files 2019-Live101 folder. You should have downloaded the media files in Exercise 4.

If needed, you can re-download the media files by going to www.halleonard.com/mylibrary and entering your access code (printed on the opening page of this book). From there, click the Download link for the Media Files 2019-Live101 listing in your My Library page. The Media Files folder will begin transferring to your Downloads folder.

Getting Started

You will start by opening the Live Set you completed in Exercise 7. If that Set is not available, you can use the Ex07 Sample file in the 01. Completed Exercises folder within the Media Files 2019-Live101 folder.

Open the Live Set and save it as Exercise 8:

1. Do one of the following:

   - Open the Set file that you created in Exercise 7 (Storage Drive/Folder > Live101-XXX > Exercise07-XXX.als).

   - Alternatively, you can use the Ex07 Sample file (Media Files 2019-Live101 > 01. Completed Projects > Ex07 Sample.als).

2. Choose **File > Save Live Set As** and name the Set *Exercise08-XXX*, keeping the Live Set inside the original Project folder. (Move the Set into your Live101-XXX folder if working from the sample file.)

Recording from Session View into an Arrangement

Up until this point, you've had the option of completing exercises in either

the Session View or the Arrangement View. For this exercise, you will need to be working in the Arrangement View. If you have been working in the Session View, you will first record your work into the Arrangement View. Live makes this process of translating your Session View scenes into an arrangement fairly simple.

If you have been working exclusively in the Arrangement View thus far, you can simply review these steps to familiarize yourself with the process, as it may be useful for you in the future.

Complete a Record Pass

Record from the Session View to the Arrangement View:

1. Click the **STOP** button in the Control Bar twice to place the Arrangement Position at 1 . 1 . 1—the very beginning of bar 1. This will ensure that your recording begins at the start of the arrangement.

2. Click the **ARRANGEMENT RECORD** button (solid circle) in the Control Bar. The button will turn red, preparing the Live Set for recording into an arrangement. However, recording will not begin until you launch a clip or scene.



Figure 8.19 The Arrangement Record button (red) in the Control Bar

3. Launch Scene 1 by clicking its **PLAY** button on the Master track.

4. Allow the scene to repeat twice.

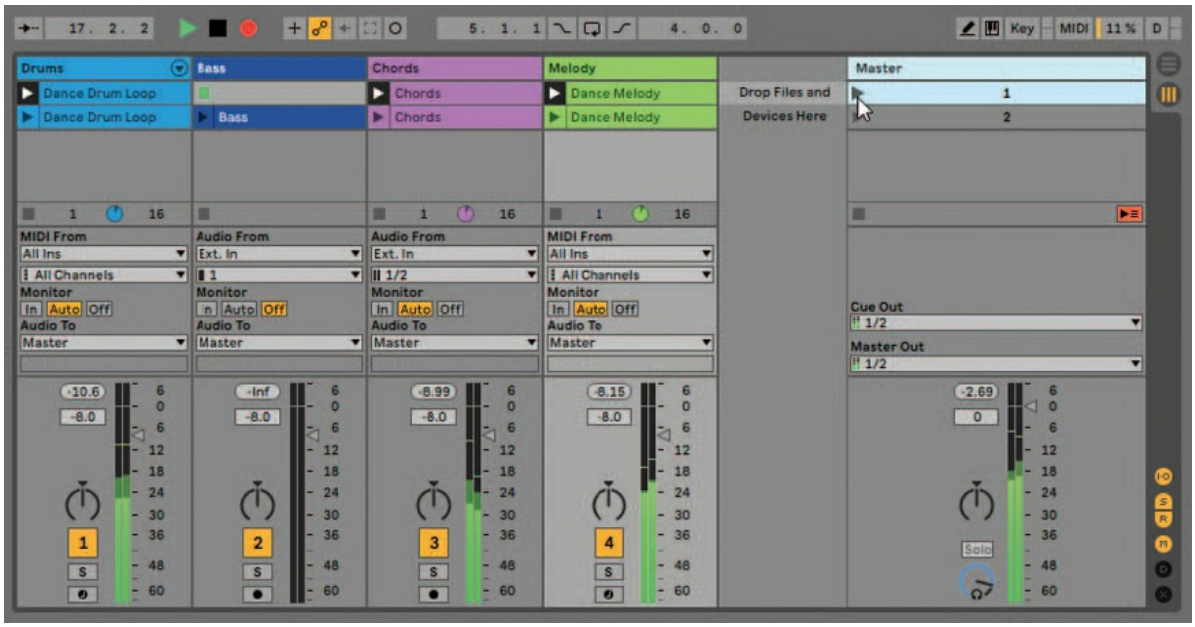


Figure 8.20 An active recording of Scene 1

5. Roughly two beats before the second repetition ends, launch Scene 2 from the Master track.

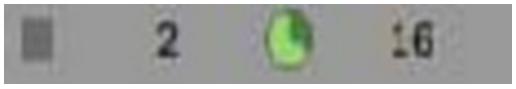


Figure 8.21 You can use the track status indicator in the Session View to see the progress of clip playback

6. Allow the scene to repeat twice.

7. After the second repetition of Scene 2, click the **STOP** button in the Control Bar or press the **SPACEBAR** to end the recording.

You've now recorded your scenes to an arrangement.

Clean Up the Recording

Next, you will verify your recorded results and clean up the ending of the arrangement, if necessary.

Switch to the Arrangement View and verify your results:

1. Press **TAB** to switch to the Arrangement View and see your recorded

scenes. The recorded material is displayed on tracks along a horizontal timeline.

You can also choose VIEW > TOGGLE ARRANGEMENT/SESSION VIEW to swap to the Arrangement View or click the ARRANGEMENT VIEW SELECTOR button.

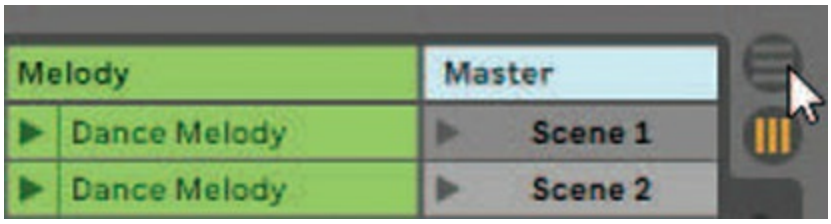


Figure 8.22 The Arrangement View Selector button

2. Click the **BACK TO ARRANGEMENT** button in the upper right-hand corner of the Arrangement View to activate the tracks for playback in the arrangement.

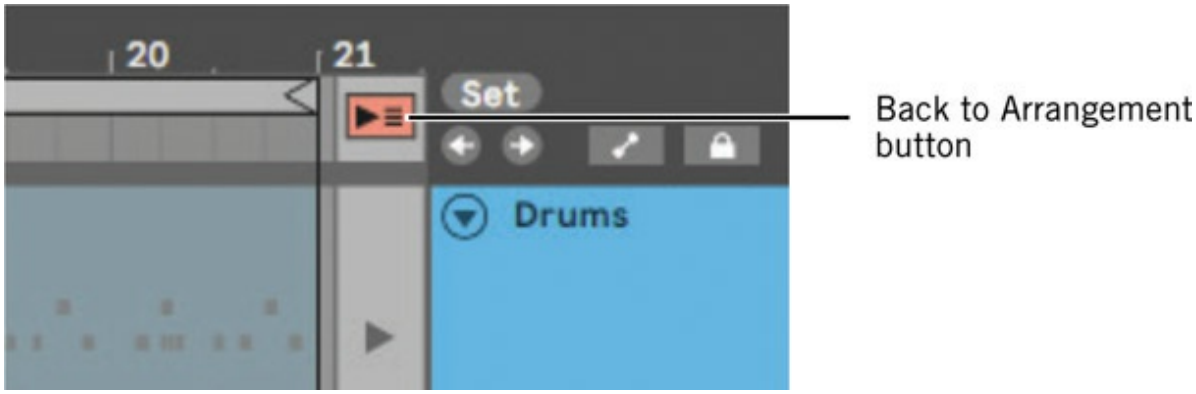


Figure 8.23 The Back to Arrangement button (red)

3. Trim the ends of clips as necessary. Your recording should be 16 bars in length; you can trim any clips that extend slightly past 16 bars so you have a clean ending to the arrangement.

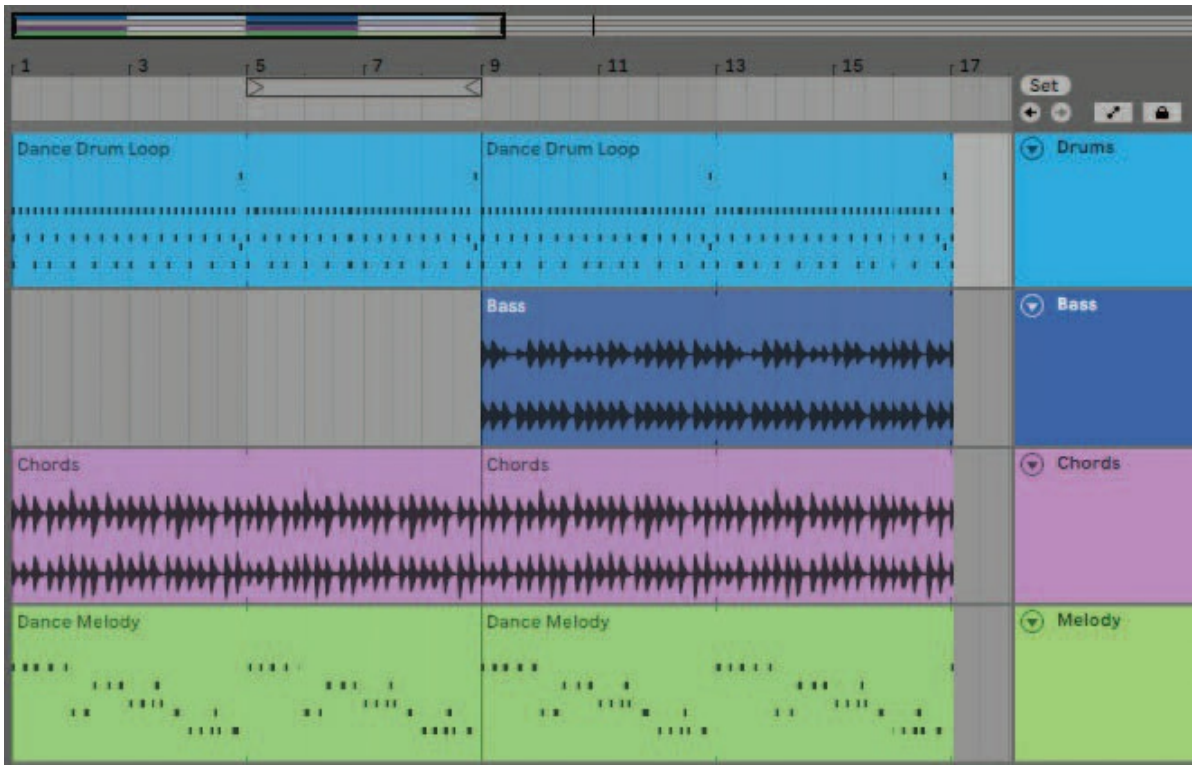


Figure 8.24 The Set after recording two scenes into the Arrangement View and cleaning up the tracks

Extending the Arrangement

In this part of the exercise, you will make a basic selection and use it to extend the duration of your existing 16-bar song segment.

Extend the arrangement by selecting and duplicating time:

1. Make a selection across the first eight bars of the composition. (See Figure 8.25.)

   You can make your selection by selecting time, by selecting an 8-bar clip, or by using any of the other selection techniques you've learned.

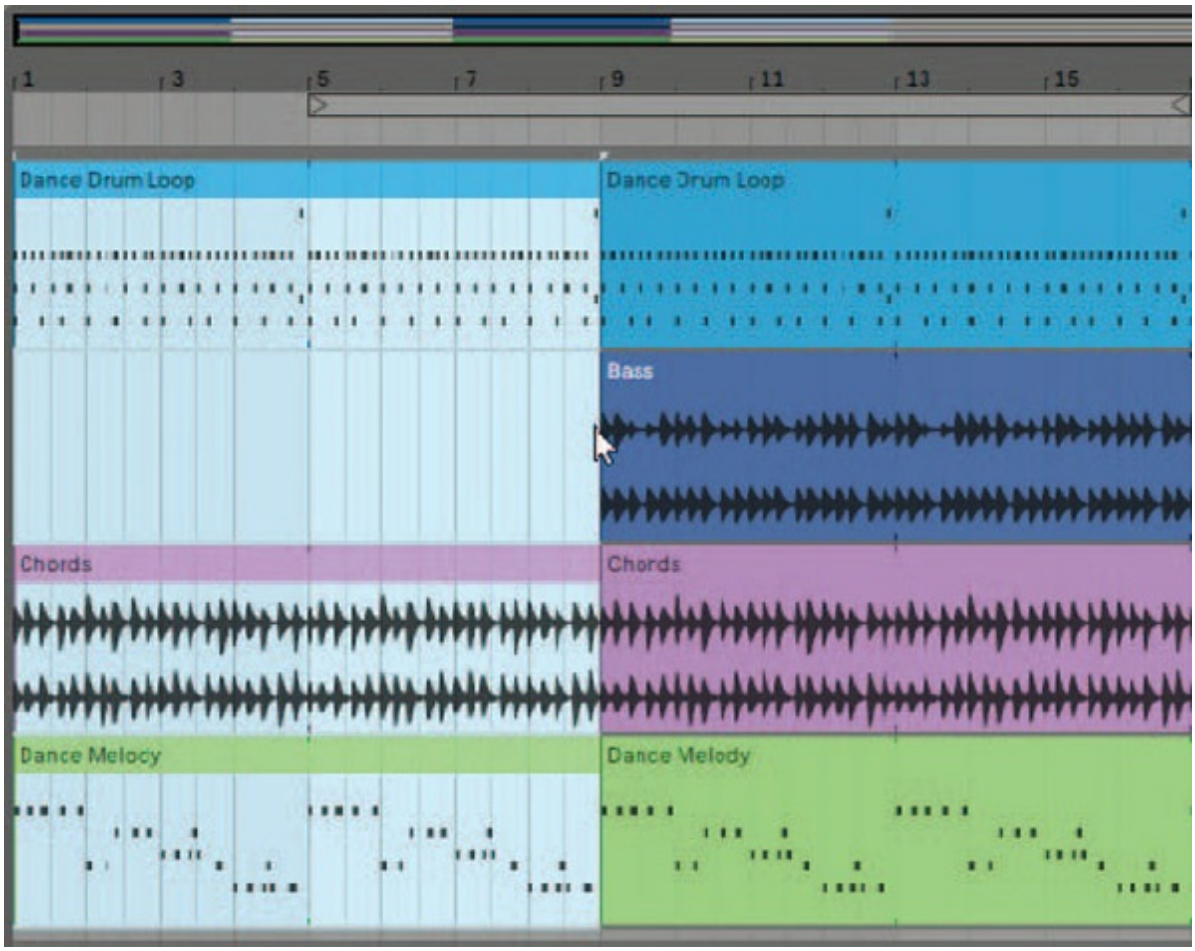


Figure 8.25 An 8-bar selection

2. From the menu bar, select **EDIT > DUPLICATE TIME**. This will duplicate all the content existing in tracks across the selected time range, extending your 16-bar arrangement to 24 bars.

3. Select the first eight bars on the Drums track.

4. Deactivate the selected drum clip(s) by pressing the **O** (**ZERO**) key or by selecting **EDIT > DEACTIVATE CLIP(S)**. This will prevent the clip(s) from playing.



Figure 8.26 Deactivating clips allows you to effectively mute clips without deleting them

You'll now have three eight-bar segments, each with different levels of activity.

Configuring the Live Set

In this part of the exercise, you will adjust the track order and height to your liking.

Change the track order according to your preferences:

- Click the name of any track that you wish to change and drag it up or down. The destination location will be represented by a horizontal black line.

Change the track height to a size of your choosing:

1. Position your pointer over the lower boundary of any track you wish to resize in the Mixer area; the cursor will change into a double-headed arrow.

2. Click on the track boundary and drag up or down. The track height will change in increments.

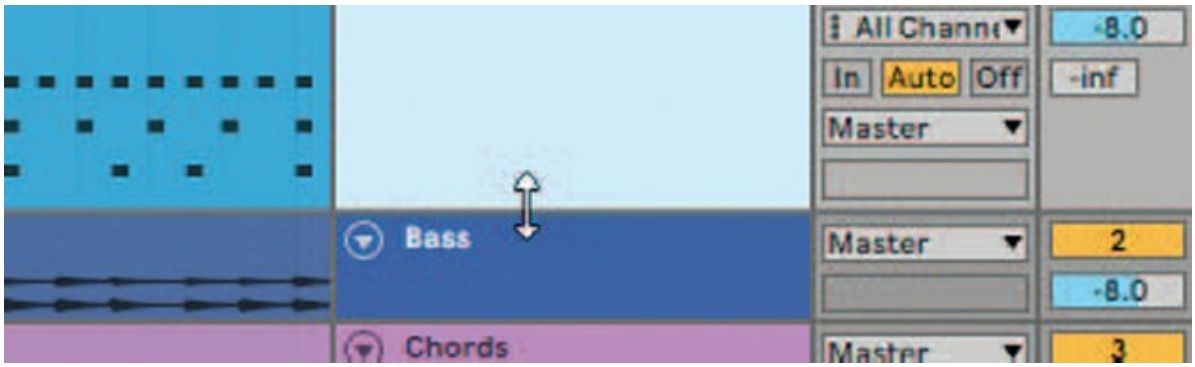


Figure 8.27 Changing track height in the Arrangement View

Adding Locators

To finish configuring the arrangement, you will add three locators to your Live Set. Here you will use locators to mark where changes occur in the arrangement.

Add locators at each change:

1. Place the Insert Marker at the start of your arrangement.
2. Right-click in the Scrub Area and choose **ADD LOCATOR**, or select **CREATE > ADD LOCATOR** from the menu bar.
3. Name the locator Intro and press **RETURN** (Mac) or **ENTER** (Windows) to accept the name.
4. Place the Insert Marker at the start of the bar 9 of your arrangement.
5. Add a second locator and name it Drums.
6. Place the Insert Marker at the start of the bar 17 of your arrangement.
7. Add a third locator and name it Bass.

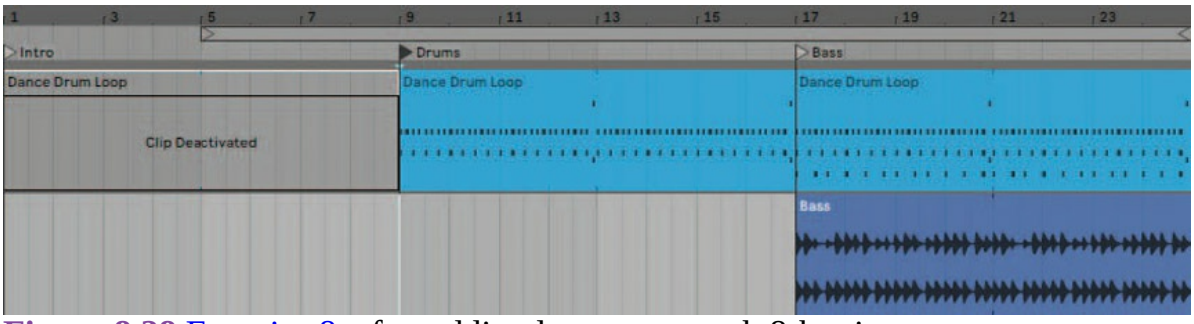


Figure 8.28 Exercise 8, after adding locators at each 8-bar increment

Finishing Up

Using the knowledge you have gained thus far, you can continue to edit and extend this basic musical idea to your liking, if desired. For now, to complete this exercise, you will need to save your work and close the Live Set.

Review your work:

1. Place the Insert Marker at the start of your arrangement.
2. Press the SPACEBAR to play back the Live Set and confirm your results.
3. Press the SPACEBAR a second time when finished.

Save your work and close the Live Set:

1. Choose **File > Save Live Set**.
2. Press **Command+Q** (Mac) or **Ctrl+Q** (Windows) to quit Live and close the project.

Chapter 9

Editing Your Media

This chapter covers the basics of editing audio and MIDI data in Ableton Live. It provides details on playback options, Edit Grid options, edit commands, and moving and resizing operations. It also introduces techniques for creating fades and for undoing edit actions.

Learning Targets for This Chapter

- Set options for scrolling (follow behavior) and looping during playback
- Recognize the difference between Fixed Grid mode and Zoom-Adaptive Grid mode
- Configure Grid values
- Apply standard editing commands to clips
- Under stand the effects of the Edit Grid on moving and trimming operations
- Create fade-in, fade-out, and crossfade effects on your tracks

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform. Sign up for free at ElementsED.com.

Any time you add audio or MIDI data to the tracks in your set, you are likely to need to do some editing. Whether you need to adjust timing, smooth out a transition, or improve a performance, editing techniques will play a large part in transforming a project from a basic recording to a polished final product. The processes described in the following sections will help you make that transformation, enabling your recordings and compositions to sound their best.

Selecting Playback Options

To simplify your navigation and workflow, Ableton Live provides several playback options to choose from while working on a Set. Two configurations you will likely want to explore include the *follow* behavior and the loop playback option.

Follow Behavior

Ableton Live offers three different scrolling options or "follow" behaviors. The current settings determine how the contents of the Arrangement View and Clip View are displayed during playback and recording.

Available options include no scrolling (follow behavior off), follow by page, and follow with a continuous scroll of the track display:

- **Follow: Off (No Scrolling).** This option prevents the screen from following the current play position. Use this option to keep the display located on an area that you are editing while playing back an area that starts or ends off screen.

- **Follow: Page.** This option scrolls on a page-by-page basis whenever the play position reaches right edge of the screen or jumps to a new location.

- **Follow: Scroll**. This option scrolls the timeline continuously, keeping the current play position centered on the screen. This option works best at lower zoom settings.

Toggling Follow On/Off

To toggle the follow behavior on or off, do one of the following:

- Click the **FOLLOW** switch in the toolbar.



Figure 9.1 The Follow switch enabled in the toolbar (yellow highlight)

- Choose **OPTIONS** > **FOLLOW**.

- Press **COMMAND**+**SHIFT**+**F** (Mac) or **CTRL**+**SHIFT**+**F** (Windows).

Setting the Follow Behavior in Preferences

To select the desired follow behavior, do the following:

1. Open the Ableton Live Preferences window by selecting **LIVE** > **PREFERENCES** (Mac) or **OPTIONS** > **PREFERENCES** (Windows).

2. Click on the **LOOK/FEEL** tab.

3. Click the indicator box to the right of the Follow Behavior label to toggle between the Page and Scroll settings.



Figure 9.2 The Follow Behavior setting in the Look/Feel tab of Ableton Live's preferences

Arrangement Loop

During editing, you will often want to listen to a selection repeatedly. Enabling Ableton Live's Arrangement Loop feature allows you to repeat your selection continuously, looping from the end of the selection back to the

start without interruption. This allows you to easily review the continuity of an edit or transition point.

To loop a selection in the Arrangement View, do the following:

1. Make the desired selection in the Arrangement View.
2. Choose **EDIT > LOOP SELECTION**. This will accomplish two tasks at once:
   - The Loop Brace will move to match the current selection.
   - The Loop switch in the Control Bar will become highlighted to show that Arrangement loop is enabled.
3. Click **PLAY** or press the SPACEBAR to start continuous looped playback.



Figure 9.3 The Control Bar with Arrangement Loop active (yellow highlight)

You can adjust the start and end po ints of the Arrangement loop during playback by modifying the Loop Brace. By dragging the corresponding triangle markers in the Loop Brace (start and end arrows), you can resize the loop area.

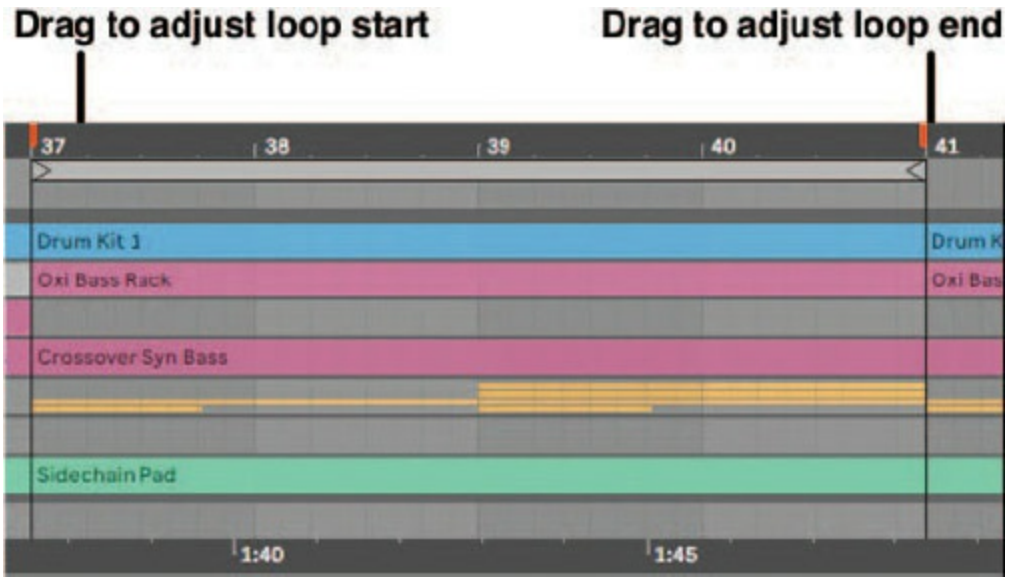


Figure 9.4 Adjusting the Loop length using the triangular start and end arrows in the Loop Brace

The Arrangement loop requires a selection of at least one quarter-note (0. 1. 0) in duration. You will not be able to set the Loop Brace to a smaller size, regardless of the Edit Grid setting.

Using the Edit Grid

As you learned in Chapter 3, Ableton Live provides two Edit Grid modes: fixed grid and zoom-adaptive grid. The Edit Grid mode can affect the size of the grid increments at different zoom settings, which in turn affects selections and edits you make to audio and MIDI clips. Both Edit Grid modes allow you to place a clip on a track without affecting the placement of other clips, leaving space between clips or overlapping clips as desired.

- **Fixed Grid.** In fixed grid mode, selections, clip movements, and resizing operations are constrained to a specific grid value that does not change based on the zoom level.

- **Zoom-Adaptive Grid.** In zoom-adaptive grid mode, selections, clip movements, and trim operations are constrained to a grid value that changes based on the zoom level. As you zoom in, the grid increments get successively smaller, allowing you to edit with greater precision.

Configuring the Edit Grid Settings

Ableton Live allows you to set the Edit Grid value based on an interval of your choosing. This helps maintain the timing of clips, notes, events, and selections as you edit your track material. The Edit Grid value affects edit operations whenever the Snap to Grid function is enabled (see “Snap to Grid” below).

The Edit Grid can also be used for display purposes, allowing it to serve as a visual reference whether or not Snap to Grid is enabled.

In Ableton Live, the Edit Grid value is always based on bars and

beats.

To set the Edit Grid value to a Fixed Grid increment, do the following:

1. Right-click in the Arrangement View or Clip View.

2. From the pop-up menu, choose a value under the Fixed Grid heading (such as 1/8). The menu will close after you make a selection.



Figure 9.5 Selecting a Fixed Grid value

To set the edit grid value to a Zoom-Adaptive Grid increment, do the following:

1. Right-click in the Arrangement View or Clip View.

2. From the context menu, choose an option under the Adaptive Grid heading (such as Medium). The menu will close after you make a selection.



Figure 9.6 Selecting a Zoom-Adaptive Grid value

To enable a Triplet Grid, value first select a Fixed Grid value, then open the menu a second time and select Triplet Grid at the bottom.

Once you've set the Edit Grid to an appropriate value, the corresponding grid lines will appear in the Arrangement View to serve as a visual reference.

Snap to Grid

Ableton Live's Snap to Grid function is enabled by default when the Edit Grid value is set to something other than **NONE**. With Snap to Grid enabled, you can make edits based on the timing interval defined by the grid. (See "Configuring the Edit Grid Settings" above.) Selections snap to grid intervals, which in turn affects cut, copy, and paste operations. Move and resize operations either align to the grid or move in grid increments relative to their origination point.

- If a clip is already positioned on a grid line, moving the clip snaps the clip start to an exact grid line. In this scenario, resizing a clip snaps the clip edge to an exact grid line as well.

- If a clip's start point falls between grid lines, moving the clip will snap

the clip start two separate incremental locations as you drag:

- The clip start will snap to exact grid lines, aligning with absolute grid locations.
- The clip start will also snap to a secondary interval that shares the same offset from the grid line as the original clip start point. This alternate snap position allows you to move clips in grid increments while preserving the clip's relative offset from the grid.

■ Similarly, if a clip's start or end point falls between grid lines, trimming the clip will snaps the associated clip edge both to successive grid lines and to the offset grid interval, preserving the relative starting point or ending point location.

Disabling Snap to Grid

Disabling Snap to Grid allows you to move or arrange the parts of your set freely and to place material anywhere on the timeline.



When Snap to Grid is disabled, the grid lines will retain their previous value but will appear greyed out.

To toggle Snap to Grid on/off, do one of the following:

■ Choose **Options > Snap to Grid**.

■ Press **Command+4** (Mac) or **Ctrl+4** (Windows).



You can temporarily suspend Snap to Grid by holding down the Command key (Mac) or Ctrl key (Windows). The same modifiers work to temporarily enable Snap to Grid when it has been toggled off.



Setting the Edit Grid Value to Off will automatically disable Snap to Grid.

Editing Clips

Ableton Live offers a variety of common editing commands—such as Copy and Paste—as well as application-specific commands—such as Split and Consolidate—that affect clips.

As we've seen previously, Ableton Live supports both audio clips and MIDI Clips. The editing techniques described in this section apply to both types of clips, unless otherwise specified.

Basic Editing Commands

Like most commercial applications, Ableton Live offers standard Cut, Copy, Paste, and Delete commands. The Ableton Live Duplicate command also offers functionality similar to that found in other media applications. Ableton Live performs each of these editing functions nondestructively, meaning that the operations do not alter the original media files on disk.

Each of these commands can be performed on a single track or on multiple tracks simultaneously, depending on the selection or the Clipboard contents. Edits can apply to the following material:

- Part of a clip or parts of multiple clips (selected by clicking in the lower half of clips)
- A whole clip or multiple whole clips (selected by clicking in the upper half of clips)

Selections can cross multiple clip boundaries, can include entire clips or partial clips, and can even include silence, if desired.

The Cut Command

Using the Cut command, you can remove selected material from its current position and place it on the Clipboard (in the computer's RAM) to be used elsewhere via the Paste command.

To cut a selection and place the material on the computer's Clipboard,

do the following:

1. Make a selection of any length on a single track or multiple tracks.

2. Choose **EDIT > CUT**, or press **COMMAND+X** (Mac) or **CTRL+X** (Windows). The selected audio and/or MIDI data will be removed from the original location and copied to the Clipboard.

The Copy Command

The Copy command is much like the Cut command, but instead of removing the selected range, it leaves the original in place and creates a copy of it on the Clipboard so that you can paste it elsewhere.

To copy a selection, do the following:

1. Make a selection of any length on a single track or multiple tracks.

2. Choose **EDIT > COPY**, or press **COMMAND+C** (Mac) or **CTRL+C** (Windows). The selected audio and/or MIDI data will be copied to the Clipboard.



When you place a selection on the computer's Clipboard using a Cut or Copy command, you replace any material previously stored on the Clipboard.

The Paste Command

Using the Paste command, you can insert the contents of the Clipboard at a location that you have selected. You can paste data only after something has been cut or copied to the Clipboard.

To use the Paste command, do the following:

1. Select the desired paste destination using one of the following methods:

    - Place the insert marker on the desired destination track at the location where you want the start of the paste to occur.

- Make a selection of any length on the desired destination track, with the beginning of the selection at the location where you want the start of the paste to occur.

To paste a multi-track selection from the clipboard, place the insert marker on the topmost track where you would like to paste the clipboard contents.

2. Choose **EDIT > PASTE**, or press **COMMAND+V** (Mac) or **CTRL+V** (Windows). The material on the Clipboard will be pasted in, beginning at the insert marker or selection start point.

If the Clipboard contains material from multiple tracks, the data will be pasted to tracks starting where the insert marker resides, filling subsequent tracks fr om top to bottom, provided the destination track types match the source tracks.

To paste data immediately after a clip, move the insert marker to the ending clip boundary by pressing Option+Right Arrow (Mac) or Alt+Right Arrow (Windows) before using the Paste command.

The Delete Command

The Delete command allows you to remove any selected material without placing it on the Clipboard.

To delete a selection, do the following:

1. Make a selection of any length on a single track or multiple tracks.

2. Choose **EDIT > DELETE**. This command has the same result as pressing the **DELETE** key on the keyboard.

The Duplicate Command

The Duplicate command makes a copy of any selected material and places it immediately after the end of the selection. This command provides a quick

way to repeat a selection, extend a sound, or create a simple looping effect—it is faster and more convenient than copying and pasting data to achieve the same result.

To duplicate audio or MIDI data, do the following:

1. Make a selection of any length and content on one or more tracks.

2. (Optional) Play the selection using Arrangement Loop to ensure that it plays smoothly in succession. If the selection loops smoothly, you can duplicate it without creating an audible edit point.

3. (Optional) Adjust the selection as needed to create a smooth loop transition.

4. Choose **EDIT > DUPLICATE** or press **COMMAND+D** (Mac) or **CTRL+D** (Windows). The selection will be duplicated and pasted at the end of the selected area or clip.

The Split Command

Splitting a clip is the process of breaking a clip in two or separating a selection as a new, independent clip.

You can split a clip for one of several purposes:

- To divide a source clip into two new clips on a track. (See Figure 9.7.)

- To separate a selection from a parent clip or from the material on either side. Use this process to create a clip from a selection, creating new byproduct clips on either side. (See Figure 9.8.)

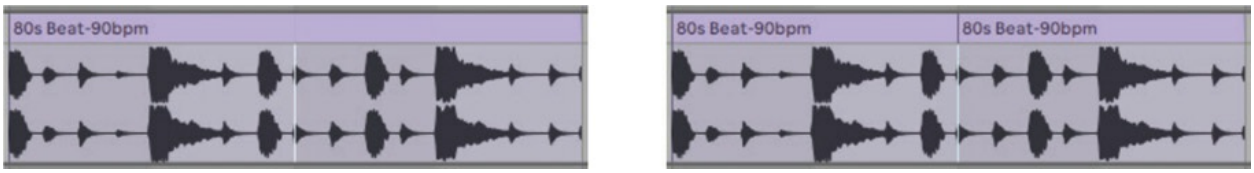


Figure 9.7 Splitting a clip at the insert marker: before (left) and after (right)

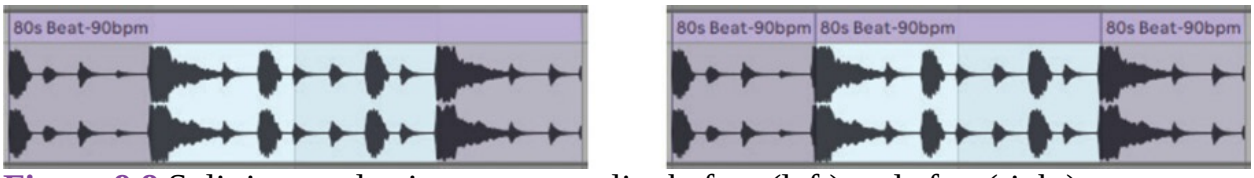


Figure 9.8 Splitting a selection as a new clip: before (left) and after (right)

When you split clips, you create byproduct clips from the material on either side of the selection.

To split a clip, do the following:

1. Place the insert marker at the location where you want a split to occur, or make a selection of any length within a clip or across multiple clips.

2. Do one of the following:

   - Choose **EDIT > SPLIT**.
   - P ress **COMMAND+E** (Mac) or **CTRL+E** (Windows).
   - Right-click within the selection and choose **SPLIT** from the pop-up menu.

Ableton Live will create new clips based on the insert marker or the selection start and end points.

Ableton Live does not automatically rename the clips that result from using the Split command, so you will have two different clips that share the same name.

Moving and Resizing Clips in the Arrangement View

The following sections describe techniques for moving and resizing clips and discuss the effects of the Edit Grid and the clip Loop option on these operations. The nudge function is also introduced.

Moving Clips

In Ableton Live, you can easily move clips within the Arrangement View. You can drag a clip to a different location within the same track or to a different track. Regardless of the Snap to Grid setting, you can always place clips so that they overlap or so that they have space between them on a track. During playback, you will hear silence in any open areas.

Moving Clips with Snap to Grid Enabled

When editing clips with Snap to Grid enabled, moving and dragging operations are constrained by the current Edit Grid value as configured in the Arrangement View.

To move a clip with Snap to Grid enabled, do the following:

1. Verify that the Edit Grid value has been set as desired. (See “Configuring the Edit Grid Setting” earlier in this chapter.)

2. Position the mouse cursor over the top half of the clip that you’d like to move. The cursor will change into a hand icon.

3. Click and drag the clip to the desired destination. A preview of the clip will appear on the track, snapping to each successive grid line as you drag.

4. Release the mouse to position the clip at the desired location.

Moving Clips with Snap to Grid Disabled

With Snap to Grid disabled, you can move clips freely.

To move a clip with Snap to Grid disabled, do the following:

1. Position the mouse cursor over the top half of the clip that you’d like to move. The cursor will change into a hand icon.

2. Click on the clip and drag it to the desired destination. A preview of the clip will appear as you drag, moving smoothly across the Track Display

area.

3. Release the mouse to position the clip at the desired location.

As you drag a clip, the Status Bar at the bottom left of the Arrangement View will dynamically update to show you the precise insert position.

Resizing Clips (Arrangement View))

The resize function in Ableton Live allows you to easily trim clips within the Arrangement View. You can shorten or lengthen clips as desired by trimming their heads or tails at any time.

Resizing a Clip

By resizing the head or tail of a clip, you can eliminate unwanted material or reveal additional material that precedes or follows the currently visible data. If the clip Loop switch is engaged, you can also loop a clip or clip segment by resizing.

To resize a clip, do the following:

1. If needed, zoom in on the clip you want to resize in the Arrangement View.

2. Position the mouse cursor at the top of the clip near the start or end. The cursor will change to a left or right bracket shape. (See Figure 9.9.)

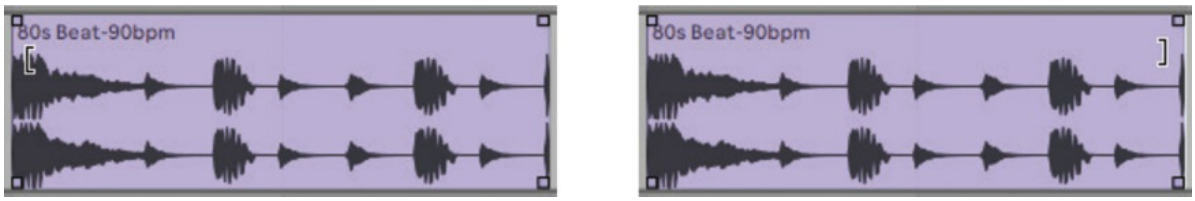


Figure 9.9 The resize cursor, as displayed near the start (left) or end (right) of the clip

3. Click and drag to resize the clip in either direction.

4. Release the mouse button at the desired location to complete the trim.

Looping Clips

The resize behavior in Live will vary, depending on whether the target clip has its Loop switch engaged. The Loop switch can be found in the Notes box (for MIDI clips) or Sample box (for audio clips). Enabling Loop functionality for a clip will allow that clip to loop indefinitely when resized.

Audio clips that are not Loop-enabled can only be resized to the up to the outer boundaries of the original audio sample or audio file recorded on disk.

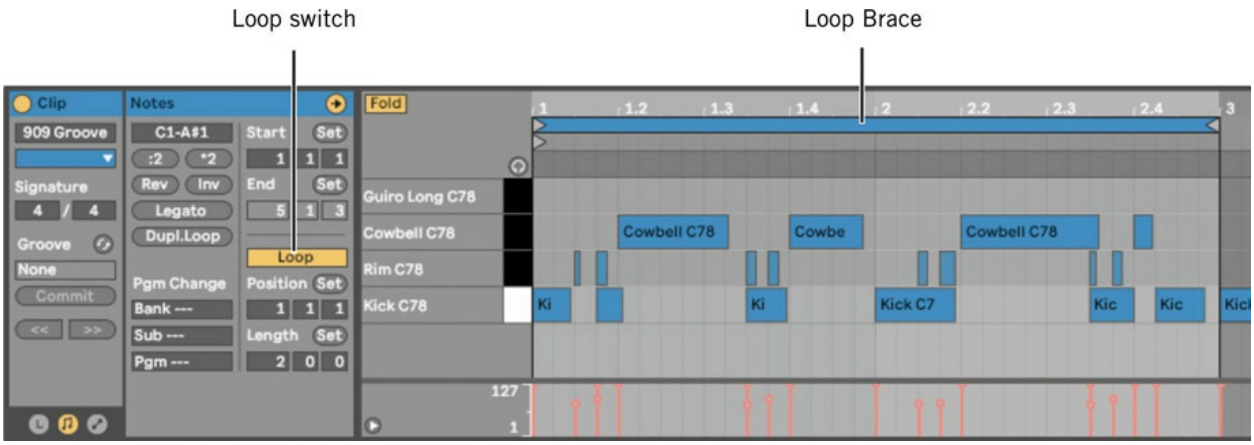


Figure 9.10 The Clip View for a MIDI clip, showing the Loop switch (Notes box) and loop brace (MIDI Note Editor)

The Warp switch must be activated prior to enabling the Loop switch for an audio clip. Unwarped audio clips cannot be looped.

To activate warping for an audio clip, do the following:

- Click the **WARP** button from the Sample box in the Clip View. (See Figure 9.11.) When active, the audio clip will automatically conform to tempo changes in your Set and the project tempo map.

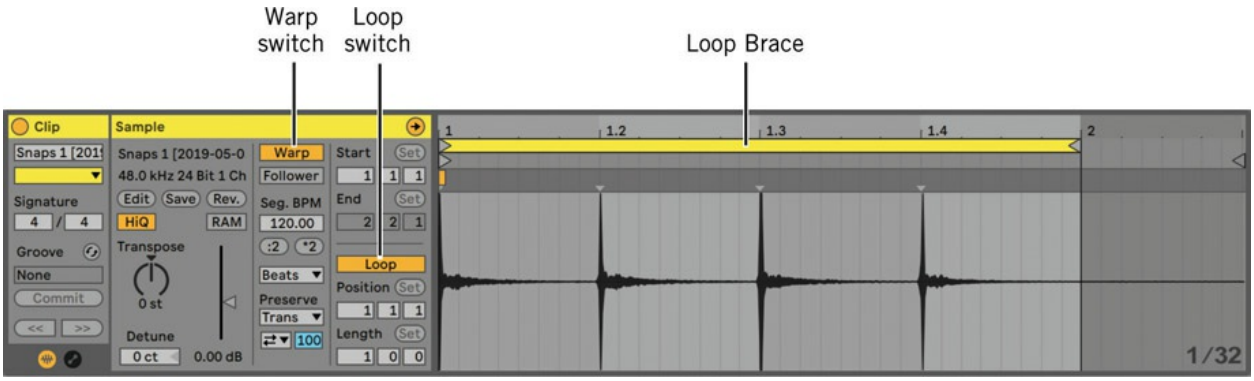


Figure 9.11 The Clip View for an audio clip showing the Warp and Loop switches in the Sample box and the loop brace in the Sample Editor

In the Arrangement View, any Loop-enabled clip can be resized beyond the length of its original media, with Ableton Live automatically adding successive iterations (repetitions) of the clip contents to achieve the desired length. The looped area of the clip will correspond to the loop position and length, as demarcated by the loop brace in the Sample Editor.

To enable clip looping:

1. Double-click on the target clip to open it in the Clip View.

2. If necessary, enable warping for the clip as described above (required for audio clips).

3. Click the **LOOP** switch in the Notes box (for MIDI clips) or in the Sample box (for audio clips).

You can drag the triangle markers (start and end arrows) in the loop brace to change length of the loop brace, or you can type exact values into the Loop Length and Position fields. You can also use various keyboard commands to adjust the loop length and position.

To modify the loop, do one of the following:

- Click and drag a triangle marker at either end of the loop brace to change its position.

- Type in exact position and length values in the Notes box (for MIDI

clips) or the Sample box (for audio clips).

■ Click on the loop brace to select it, and do one of the following:

- Press the **LEFT** or **RIGHT ARROW** keys to nudge the loop brace to the left/right by the Grid value.
- Hold **COMMAND** (Mac) or **CTRL** (Windows) while pressing the **LEFT** or **RIGHT ARROW** keys to shorten or lengthen the loop brace by the Grid value.



Regardless of the position of the loop brace, clip playback will begin at the clip start marker. This means you can configure a clip to run into a loop by setting the clip start location earlier than the loop brace start location.

Nudging Clips and Selections

Ableton Live allows you to adjust the placement of clips and selections by nudging them with the keyboard. With Snap to Grid enabled, nudging moves clips and selections by the Edit Grid Value. With Snap to Grid disabled, nudging moves in very small increments and adapts to the zoom level.

Nudging Clips

To nudge a single clip or multiple clips by the Edit Grid Value, do the following:

1. Verify that that Snap to Grid is enabled.
2. Select the clip or clips you want to nudge.
3. Press the **LEFT ARROW** key to move the clip(s) earlier in the track or the **RIGHT ARROW** key to move the clip(s) later in the track. The clip(s) will move incrementally by the Edit Grid Value.

To nudge a single clip or multiple clips by a small, zoom-adaptive value, do the following:

1. Do one of the following to disable Snap to Grid:
   - Choose **Options** > **Snap to Grid**.
   - Press **Command+4** (Mac) or **Ctrl+4** (Windows)
   - Set the Edit Grid Value to **Off**.
2. Select the clip or clips you want to nudge.
3. Press the **Left Arrow** key to move the clip(s) earlier in the track or the **Right Arrow** key to move the clip(s) later in the track. The clip(s) will move incrementally by a small, zoom-adaptive value.



Hold the Command key (Mac) or Alt key (Windows) while nudging to flip the nudge behavior so that clips nudge by the Edit Grid Value when the Edit Grid is disabled or by a small zoom-adaptive value when the Edit Grid is enabled.



You can snap clips to the next or previous Locator by holding down the Option key (Mac) or Ctrl key (Windows) while nudging.

Nudging Selections

In addition to nudging clips, you can also nudge selections using the **Left Arrow** and **Right Arrow** keys on the computer keyboard.

To nudge a selection, do the following:

1. Make a selection on any track or tracks.
2. Press the **Left Arrow** key to move the selection earlier in the track or the **Right Arrow** key to move the selection later in the track. The selection and its contents will move earlier or later following the nudge behavior described above relative to the Snap to Grid status.

To change the length of a selection by nudging:

- Hold the **Shift** key while pressing the **Left Arrow** key or the **Right Arrow** key on the computer keyboard. The selection will lengthen or shorten incrementally following the nudge behavior relative to the Snap to Grid status.

Nudging to Clip Boundaries and Locators

You can perform macro-level nudging to move to successive clip boundaries and locators, by holding modifier keys.

To nudge a clip or selection to the next or previous clip boundary or locator:

- Hold **Option** (Mac) or **Ctrl** (Windows) while pressing the **Left Arrow** key or **Right Arrow** key on the computer keyboard. The start of the clip or selection will snap to successive start and end boundaries for other clips on the track and any locators in the Set.

You can also use macro-level nudging to extend selections to successive clip boundaries and locators.

To extend a selection to the next or previous clip boundary or locator:

- Hold **Option+Shift** (Mac) or **Ctrl+Shift** (Windows) while pressing the **Left Arrow** key or **Right Arrow** key on the computer keyboard. The selection will lengthen or shorten by snapping to the next or previous clip boundary or locator.

Undoing Your Work

Often your editing tasks will involve performing a series of related steps to achieve a desired effect. Along the way, you might at times need to revert to an earlier point, either to start over or to do a before-and-after comparison. Fortunately, Ableton Live provides rich undo options that give you the flexibility to work without constraints.

Using Multi-Level Undo

Multi-level undo operations make it possible to return to earlier stages of work during the editing process. This lets you work and experiment with confidence, knowing that you can back out of changes if you are not satisfied with the results.

Ableton Live provides an unlimited number of undo operations. All commands that are undoable are stored sequentially in an undo queue. However, a small number of commands in Live cause changes that are not undoable.

Some actions that *cannot* be undone include the following:

- Closing a Set and/or quitting Ableton Live (Clears the Undo Queue)
- Deleting Clips from the Current Project Folder in the Browser

Using the Undo and Redo Commands

To access the Undo command, choose **Edit > Undo** or press **Command+Z** (Mac) or **Ctrl+Z** (Windows). The Undo command in the Edit menu lists the action to be undone along with the command name.



Figure 9.12 The Undo command in the Edit menu, showing the action that will be

undone

To perform multiple undo operations, repeat the above process as needed.

If you undo an action that you want to keep, you can reinstate the action using the Redo command. To access the Redo command, choose **Edit > Redo** or press **Command+Shift+Z** (Mac) or **Ctrl+Shift+Z** (Windows). Like the Undo command, the Redo command lists the action that it will affect.

Restoring from Backup Sets

Ableton Live retains the undo queue as long as the Set is running. However, at times you may need to go back far enough that using the undo command hundreds of times simply isn't practical. In these cases, restoring from a Backup Set may be a better option.

The ten most recently saved versions of the current Set are automatically saved in a Backup folder that Ableton Live creates inside the project folder.



Ableton Live automatically deletes the oldest backup Set once the number of backups exceeds 10 files.

Opening a Backup Set is similar to opening an earlier saved copy of the Set. However, you can recognize a Backup Set because it will have the date it was saved appended to the Set name.

To restore your Set from a backup, do the following:

1. Choose **File > Open Live Set** and navigate to the Backup folder within your project folder.

2. Select a Backup Set based on the modification date and time.

3. Once the Set opens, verify that the backup file represents the desired stage of the Set.

4. Select **File > Save Live Set As** to save the restored Set with an appropriate name.



You can also open Backup Sets using the Browser by navigating to Current Project > Backup and double-clicking one of the Backup Sets.

Review/Discussion Questions

1. What three scrolling behaviors are available in Ableton Live? (See “Follow Behavior” beginning on page 244.)

2. What does the Arrangement Loop option do? What is the minimum selection length required for an Arrangement Loop? (See “Arrangement Loop” beginning on page 245.)

3. What two Edit Grid Modes are provided in Ableton Live? (See “Using the Edit Grid” beginning on page 246.)

4. How can you configure the Edit Grid value? (See “Configuring the Edit Grid Settings” beginning on page 246.)

5. What happens when you move a clip whose start point falls between Grid lines with the Snap To Grid function enabled? What happens when you move a clip with Snap to Grid mode disabled? (See “Snap to Grid” beginning on page 248.)

6. Name some basic editing commands provided in Ableton Live. (See “Basic Editing Commands” beginning on page 249.)

7. What are some operations that the Split command can be used for? (See “The Split Command” beginning on page 251.)

8. What is the purpose of resizing a clip? How can you resize a clip in the Arrangement View? (See “Resizing a Clip” beginning on page 253.)

9. How can you enable a clip to repeat its contents when resized, creating a looping pattern? Where are the controls for this functionality found for MIDI clips? Where are they found for audio clips? (See “Looping Clips” beginning on page 254.)

10. What additional switch needs to be activated to allow looping for an

audio clip? (See “Looping Clips” beginning on page 254.)

11. What keys are used to nudge a clip or selection earlier or later on a track? (See “Nudging Clips” beginning on page 256.)

12. What is the maximum number of operations can you undo in Ableton Live? What are some operations that cannot be undone? (See “Using Multi-Level Undo” beginning on page 258.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 9

Editing Audio

Activity

In this exercise, you will use the Locators you created in Exercise 8 to add a noise effect. You will also shorten the music tracks to fit within a 45-second target length.

Duration

This exercise should take approximately 10 to 15 minutes to complete.

Goals/Targets

- Use Locators to add a noise sweep
- Extend the end of song to a specific length

Media Files

To complete this exercise, you may need to use various media files included in the Media Files 2019-Live101 folder. You should have downloaded the media files in Exercise 4.

If needed, you can re-download the media files by going to www.halleonard.com/mylibrary and entering your access code (printed on the opening page of this book). From there, click the Download link for the Media Files 2019–Live101 listing in your My Library page. The Media Files folder will begin transferring to your Downloads folder.

Getting Started

You will start by opening the Live Set you completed in Exercise 8. If that Set is not available, you can use the Ex08 Sample file in the 01 Completed Exercises folder within the Media Files 2019-Live101 folder.

Open the Live Set and save it as Exercise 9:

1. Do one of the following:

   - Open the Set file that you created in Exercise 8 (Storage Drive/Folder > Live101-XXX > Exercise08-XXX.als).

   - Alternatively, you can use the Ex 08 Sample file (Media Files 2019-Live101 > 01. Completed Projects > Ex08 Sample.als).

2. Choose **File > Save Live Set As** and name the Set *Exercise09-XXX*, keeping the Live Set inside the original Project folder. (Move the Set into your Live101-XXX folder if working from the sample file.)

Adding a Noise Sweep

In this part of the exercise, you will use the Locators you added earlier to create a transition using a noise sweep effect.

Configure the Edit Grid Value:

- Right-click in the Arrangement View and set the Edit Grid value to **Fixed Grid**: 1/8 (1/8 note), if not already selected.



Figure 9.13 Setting the Edit Grid Value to 1/8 notes

Place the noise effect:

1. In the Browser, navigate to **PLACES > 02 EXERCISE MEDIA** and locate the Noise_Sweep_120 clip.

2. Drag the clip onto the Arrangement View so that the start of the clip lines up with the **DRUMS** Locator. Note that the clip placement does not need to be exact.

Be sure to drag the clip to the "Drop Files and Devices Here" area so that clip will be place on a new track.

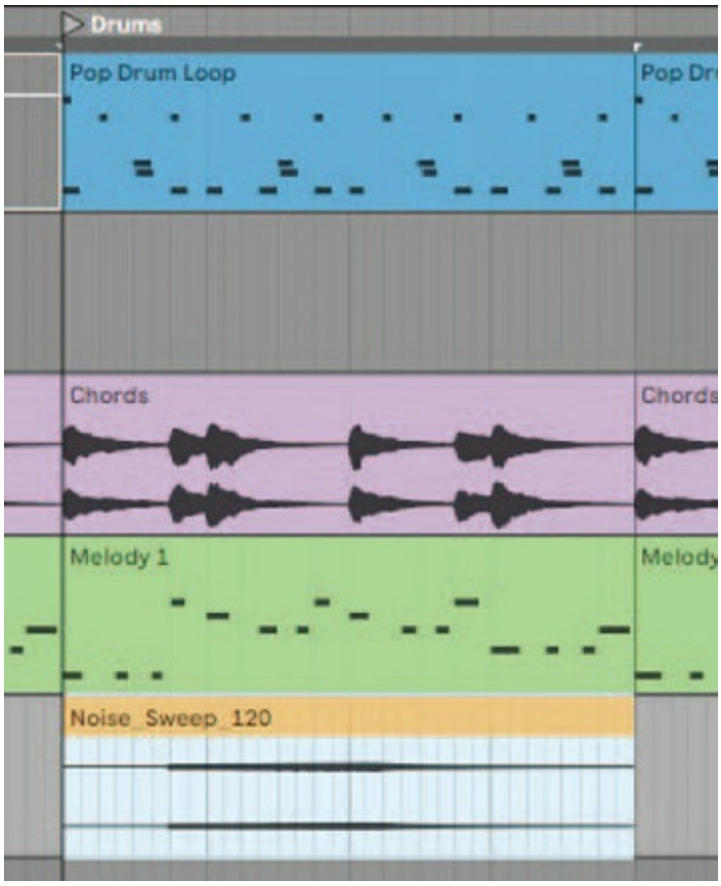


Figure 9.14 The Noise Sweep clip placed at the Drums Locator position

Nudge the noise sweep:

- With the Noise_Sweep_120 clip selected, use the **LEFT** and **RIGHT ARROW** keys to nudge the clip into position, so that it peaks near the Drums locator position.

Nudge the clip by Edit Grid increments until the middle of the clip is visually aligned with the Drums locator. Continue to nudge with Snap to Grid turned off, auditioning periodically, to align the clip by ear at the best location.

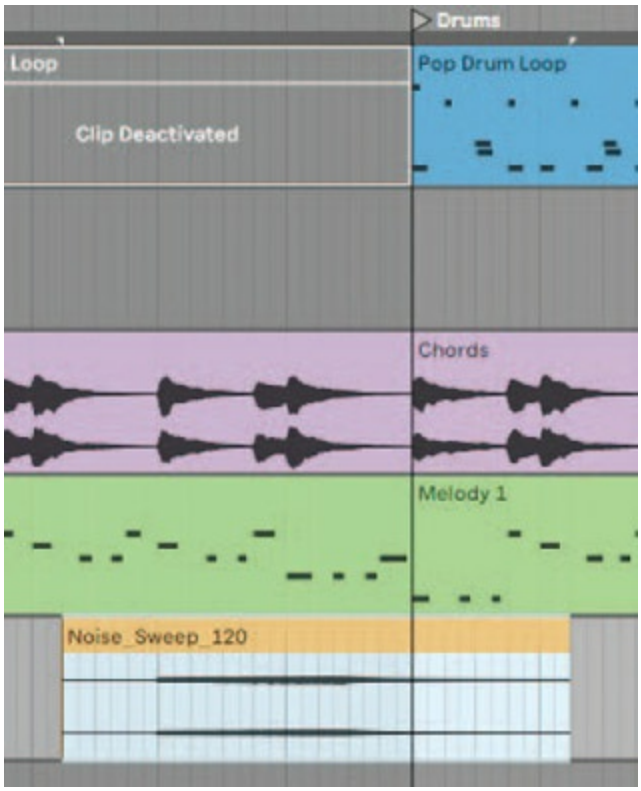


Figure 9.15 The Noise Sweep peaking near the Drums Locator

Extending the Song

In this section of the exercise, you will extend the Drums, Bass, Chords, and Melody tracks so that the song is around 60 seconds in length.

Duplicate the clips:

1. Locate the final clips at the end of the song.
2. Select any one of the clips.
3. Choose **EDIT > DUPLICATE TIME** or press **COMMAND+SHIFT+D** (Mac) or **CTRL+SHIFT+D** (Windows) to duplicate the clips.

If you have a song that you need to shorten, you can select a clip and use the Edit > Delete Time command to remove content, reducing the total duration.

Resize the Chords clip:

1. Place the cursor near the right edge of the final Chords clip. The cursor will display a bracket icon.

2. While referring to the Time Ruler at the bottom of the Arrangement View, click and drag to resize the end of the Chords clip to around 1:00.

Hold Command (Mac) or Alt (Windows) as needed to trim with the Edit Grid temporarily inactive.

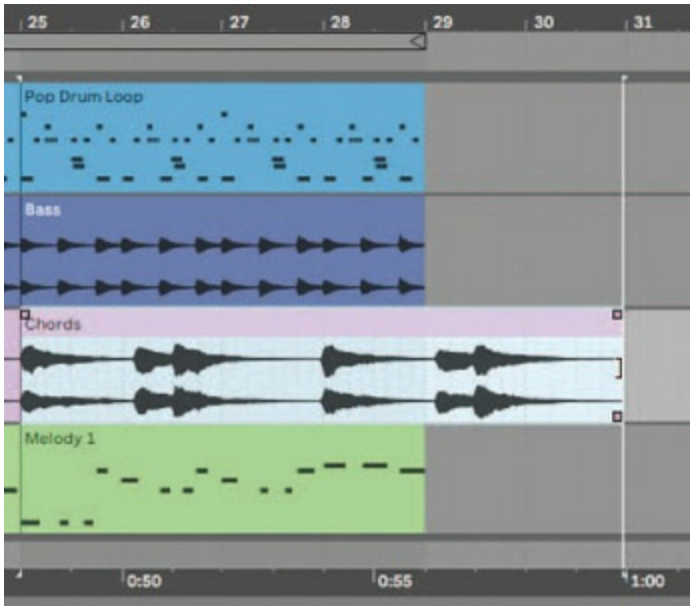


Figure 9.16 The Chords clip resized to end around 1:00

Add a fade out to the end of the Chords clip:

1. Place the cursor near the upper right edge of the clip. The Fade Our Start Handle will appear.

2. Click and drag to the left to create a fade out. Try creating a fade out over approximately one bar.

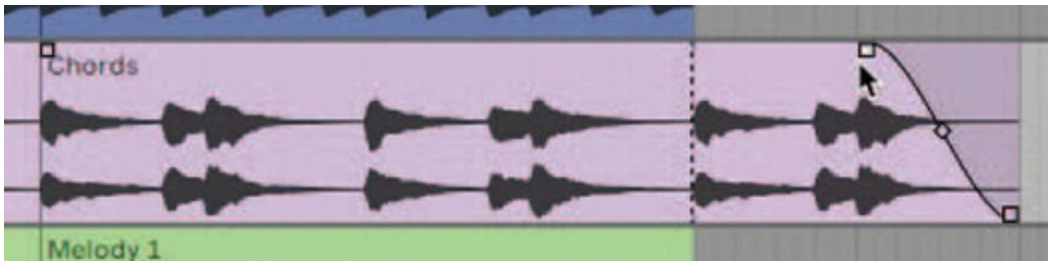


Figure 9.17 The Chords with a fade out

Finishing Up

Using the knowledge you have gained thus far, you can continue to edit and fine-tune this basic musical idea to your liking, if desired. To complete this exercise, you will need to save your work and close the Set.

Review your work:

1. Place the Insert Marker at the start of your arrangement.
2. Press the SPACEBAR to play back the Live Set and confirm your results.
3. Press the SPACEBAR a second time when finished.

Save your work and close the Live Set:

1. Choose **File > Save Live Set**.
2. Press **Command+Q** (Mac) or **Ctrl+Q** (Windows) to quit Live and close the project.

Chapter 10

Mixing and Creating a Bounce

This chapter covers basic mixing techniques and processes as they are performed in Ableton Live. It includes discussions of mixer terminology, mixer configuration (including configuring devices, sends, and returns), basic automation, and real-time processing. The chapter also covers techniques for finishing your work, including creating a project back-up/archive and a final mixdown your project as a stereo file.

Learning Targets for This Chapter

- Recognize common mixer terminology
- Understand how devices and send-and-return paths are used to add signal processing to your tracks
- Configure the devices and sends in the mixer
- Record and edit basic automation for your mix
- Add devices to your tracks for internal effects processing and sound shaping
- Understand the purpose of the Collect All and Save command and recognize situations in which you should use it
- Select appropriate options for your stereo mix when using Export Audio/Video

Key topics from this chapter are illustrated in the Ableton Live 101 Study Guide module available through the Elements|ED

online learning platform. Sign up for free at ElementsED.com.

With all of your tracks recorded and edited, it's time to start thinking about how to create a good balance between the audio elements, adding emphasis where it's needed and creating the subtle touches required for a professional–sounding result. Setting levels, adding signal processing, and creating dynamic automation are all parts of this process.

Once the mix sounds the way you want it, you will be ready to output the finished product as a stereo file. This chapter outlines the essential concepts and techniques required to complete all of these mixing tasks.

Basic Mixer Terminology

The fundamental job of any audio mixer is to route incoming and outgoing audio via the mixer's inputs and outputs. Additional signal routing and processing can be achieved using the mixer's inserts and send and return functions. These terms are defined in this section as they apply to general audio mixing; specific Ableton Live applications of these concepts are described under "Mixing in the Session View" later in this chapter.

Inputs

The term *input* refers to an audio signal traveling into an audio hardware device, such as a mixer or an audio interface. The inputs available in Ableton Live vary depending on the system and the audio interface(s) in use.

Outputs

The term *output* refers to an audio signal traveling out of an audio hardware device. The outputs available in Ableton Live also vary depending on the system and the audio hardware in use. In a new Set, all tracks will route to the main stereo outputs (left and right) by default.

Inserts (Devices)

Most mixers have a feature known as a *channel insert*. An insert is an audio patch point that allows a signal processor to be placed directly into the signal path of the audio channel. Ableton Live uses the term *device* to describe

channel inserts.

Ableton Live provides an unlimited number insert positions per track, allowing you to process a track's signal through multiple software plug-ins and/or external effects loops in series.

Sends and Returns

The term *send* refers to a signal path carrying a mix output of one or more channels (or tracks) routed for parallel processing. The send may route to an external receiving device, such as a hardware effects unit, or to an internal processor, such as a software plug-in.

Pre-Fader versus Post-Fader Sends

Sends can be *pre-fader*, meaning the send level is independent of the source track's Volume slider level, or *post-fader*, meaning the send level will be affected by changes you make to the Volume slider on the source track. In Ableton Live, sends are post-fader by default; however, you can set a send to pre-fader at any time, as needed.

Send and Return Processing

When using a send for external processing, the signal is routed out of the mixer, through an external device where some type of processing is added, and then returned to the mixer through an *auxiliary return* channel. When using a send for internal processing in Ableton Live, effects are added using a plug-in applied to the returned signal on the destination track (known as a Return track).

The return channel in the mixer provides level and pan controls, allowing precise control over how the reintroduced signal combines with other audio in the system.

Mixing in the Session View

Although all of Ableton Live's mixing controls can be viewed in the Arrangement View, mixing operations and functions are typically performed

using the Session View. The Session View is similar to a standard mixing console. This view offers a variety of display options, many of which can also be customized.

The Session View can be swapped with the Arrangement View as needed. To switch to the Session View, choose **VIEW > TOGGLE ARRANGEMENT/SESSION VIEW**.



You can also press the TAB key to toggle between the Session and Arrangement Views at any time.

Configuring the Session View for Mixing

The Session View includes several component parts and controls. Among the controls that you will use to create your mix are Volume controls, Pan controls, I/O choosers, and Send controls.



The In and Out section in Live includes choosers for Audio Input Type and Channel and Audio Output Type and Channel, as well as MIDI Input Type and Channel and MIDI Output Type and Channel (MIDI tracks only).

To create a mix, you will set the Volume control for each track to achieve an appropriate blend of audio levels, and set the Pan controls to achieve the desired positioning of sounds within the stereo field. You can make changes in real time during playback, either by manually adjusting the controls or by using automation. (See “Using Basic Automation” later in this chapter.)

The Input and Output choosers are used to route signals to and from your tracks. Often the basic I/O routing for your mix will already be in place, based on the work you have done up to this point. However, if needed, you can use these selectors to configure different signal routing for your mix.

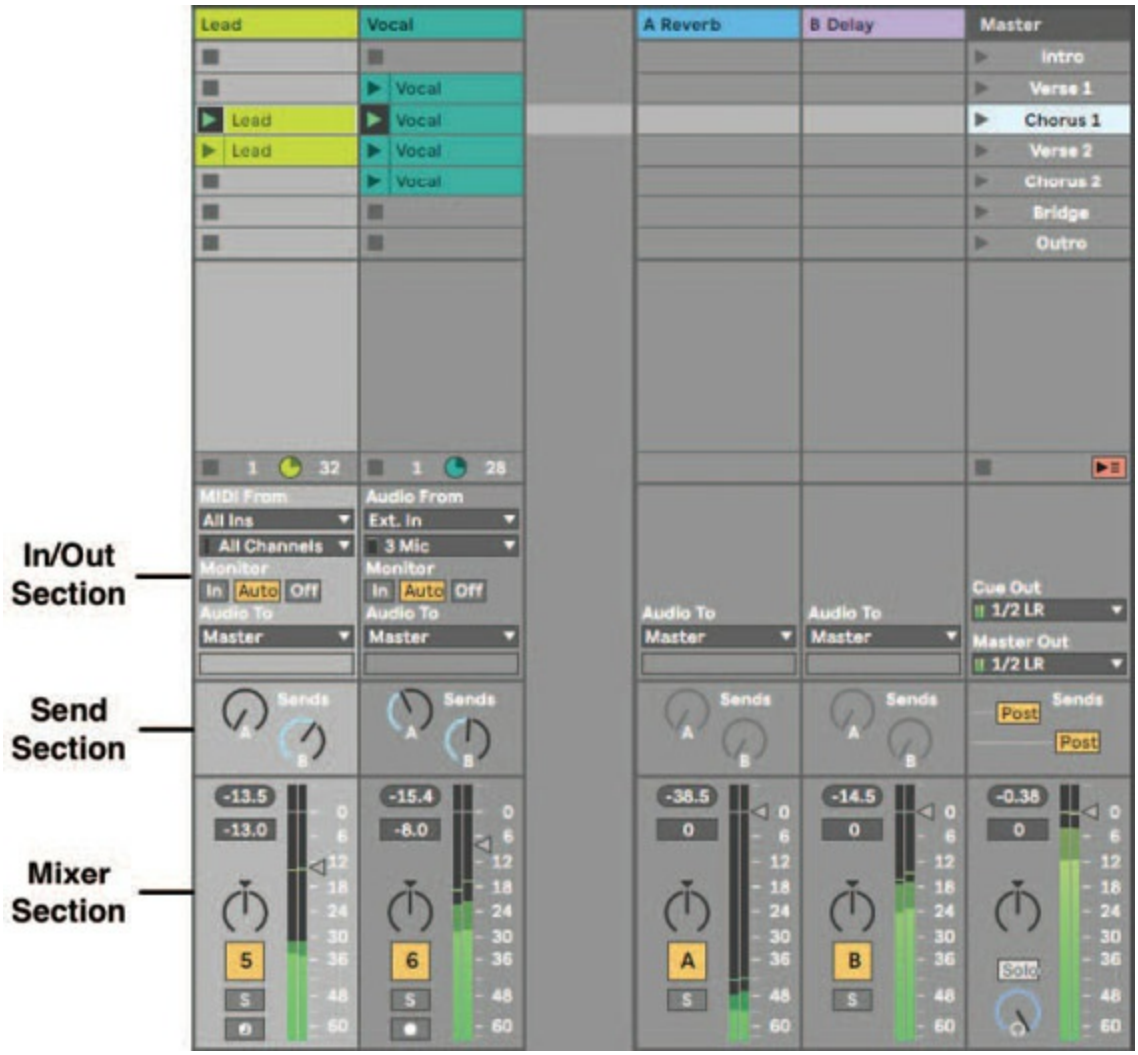


Figure 10.1 Mixing controls in the Session View

Mixing also often involves using devices and sends to add various types of signal processing to the audio on a project's tracks. You can use the Send controls and the Device View in the Session View to achieve these operations.

Showing/Hiding Sections

You can show or hide sections of the Session View using the Show/Hide buttons at the bottom right corner of the Session View. Click a button to toggle the show/hide status of the associated section.



Figure 10.2 The Session View Show/Hide buttons for the In/Out section

The Show/Hide buttons shown in Figure 10.2 are as follows, from top to bottom:

- **I•O –** In/Out Section
- **S –** Sends Section
- **R –** Return tracks
- **M –** Mixer Section
- **D –** Track Delays Section
- **X –** Crossfader Section

Input and Output Choosers

The Session View displays the main Input and Output choosers for each track in the project by default. Though much of your signal routing might have been set up during the recording and editing stages of your project, it is always a good idea to double-check the I/O settings when you begin mixing.

Input Choosers

For tracks that are playing back material already on the track, no input routing is necessary. Tracks that are receiving live input from other sources will need to have their inputs set accordingly.

Output Choosers

Ableton Live enables you to route the output of each track to any available hardware output. For the purposes of creating a stereo mix, you will generally use the main stereo outputs of your audio interface (default).

To set up a basic stereo mix, verify that the Audio Output choosers for the tracks in your Set are routed to the main outputs of your audio interface, as appropriate, so that the audio from each track is included in the stereo playback. This is typically done by assigning each track to the output type labeled Master. This setting routes the track output to the Master track, where the physical output can be assigned using the Master Out chooser.

Ableton Live 10 introduced the ability to label inputs and outputs with custom names. To do this, go to the Audio tab in Preferences. After selecting the Driver Type and Audio Devices, click the INPUT CONFIG and OUTPUT CONFIG buttons to add labels for the available mono and stereo paths.

If needed, use the Audio Output Type and Channel choosers to select the desired output for your tracks.

Sends Section

The Session View has a dedicated section for the track send controls known as the Send Section. This area can be shown or hidden in the Session View (see “Showing/Hiding Sections” above).



Figure 10.3 Sends Section in the Session View

The Sends Section allows you to access and view up to 12 Send Controls (A – L) in Ableton Live Standard and Suite editions. The default Ableton Live Set is pre-configured with two sends (A and B). Additional send controls are added as you create additional return tracks.

The Sends Section is part of the Mixer Section in the Arrangement View.

Configuring Devices

You can add one or more signal processing devices to any track in Ableton Live. Audio Effects and Plug-ins provide software-based signal processing, and the External Audio Effect device can be used to create an effects loop for an external hardware device.

To add a device to a track:

1. Locate the device using the Audio Effects or Plug-ins Categories in the Browser.

2. Do one of the following:

   - Drag and drop the device onto the desired track.

   - Select the destination track and then double-click on the device in the Browser.

Software Devices

Audio Effects and Plug-ins route audio through a software add-on inserted onto a track. Using the Browser, you can choose from the Audio Effects or Plug-ins Categories to add a software signal processor, such as an EQ plug-in, directly into the signal path of the channel.



Figure 10.4 EQ Eight and Compressor inserted onto a track

External Audio Effect

The External Audio Effect device routes audio through an external hardware device using parallel inputs and outputs of an audio interface. This option requires an audio interface with sufficient I/O capacity to support hardware inserts in addition to stereo playback through the main left–right outputs.

Configuring Sends and Returns

Sends are used to route a track's signal to a secondary path for parallel processing (internal or external) without interrupting the signal flow through the originating track. To add the processed signal back into the mix, it is returned through a channel strip known as a return track in Ableton Live.

To create a Send to route to an internal processor, such as an Audio Effect or Plug-in, simply create a new return track. A Send to that return track will automatically be created across all of the tracks in the Set.

To create a new return track do one of the following:

- Choose **Create > Insert Return Track**.
- Press **Option+Command+T** (Mac) or **Ctrl+Alt+T** (Windows).

To route a send to an external device:

1. Create a return track.
2. Insert the External Audio Effect device on the return track.
3. Set the Audio To chooser and the Audio From chooser in the External Audio Effects device to the appropriate connections on your audio interface.

Automation

For a simple mix, you can often set the Volume and Pan controls as desired and leave them unchanged from the start of the mix to the end. For a more complex mix that requires dynamic changes during the course of playback, you can use Arrangement automation to change control settings throughout a

song. Ableton Live allows you to record any real-time changes you make as automation on your tracks. Automatable controls include volume, pan, speaker on/off (mute), send levels, and more.

Arrangement Automation Workflow

In a typical production workflow, the ultimate destination for all automation is the Arrangement View. Aggregating all of the automation in a unified timeline enables a producer to fine tune automation while considering the needs of the entire song.

Automation can be recorded to the Arrangement View in two ways:

- Clip automation created in the Session View can be recorded to the Arrangement using the Arrangement Record function.
- Manual parameter changes can be recorded directly into the Arrangement.

Recording Arrangement Automation

Recording automation in the Arrangement View is quite simple. When the Automation Arm button is enabled, any track control that is moved during an Arrangement Record pass will become Arrangement Automation, regardless of the status of the individual track's Arm button.

To record Arrangement automation:

1. Enable the Automation Arm button in the Control Bar.



Figure 10.5 The Automation Arm button (yellow)

2. Click the Arrangement Record button (solid circle) in the Control Bar.
3. If needed, click the Play button to start the transport.

If the START PLAYBACK WITH RECORD preference is enabled,

clicking the Arrangement Record button will immediately start playback.

4. Perform any desired control changes in real time during the record pass.

   If Automation Mode is currently enabled (see "Viewing Arrangement Automation" below), you will see the track's main automation lane update in real time. Otherwise, the track will turn red to show that automation is being written.

5. Stop playback when finished.

Automation LEDs

Any control that has automation data on it will display a red automation indicator (or "LED"). The location of this LED can vary depending on the parameter type and the current view. In the Session View, automation LEDs will appear to the upper left corner of most controls, and inside Volume sliders. In the Arrangement View, automation LEDs will appear in the upper left corner of the control.

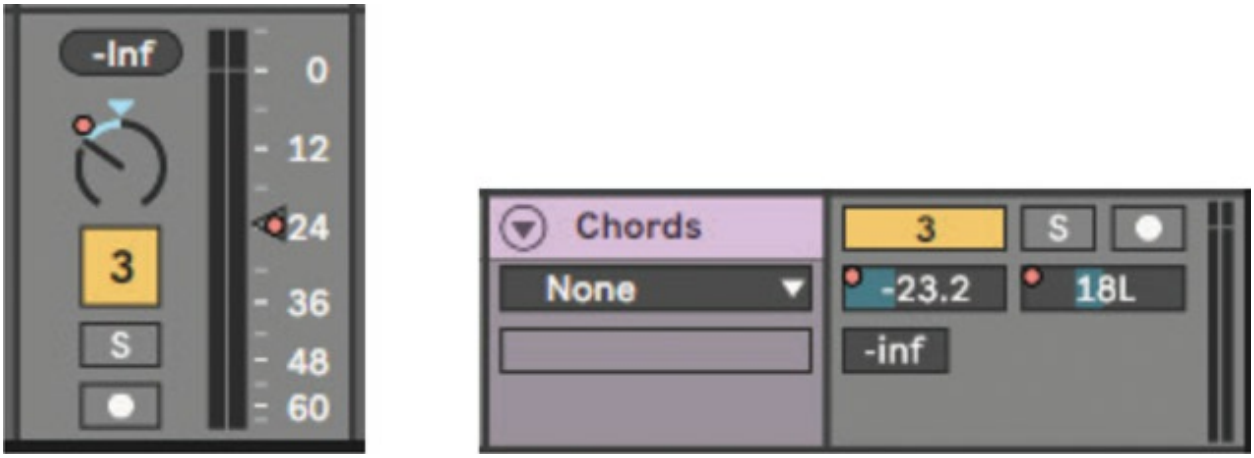


Figure 10.6 Pan and Volume Automation LEDs visible in the Session View (left) and the Arrangement View (right)

Viewing Arrangement Automation

By default, tracks in the Arrangement View show track content and not automation data. To view automation data you must enable Automation

Mode.

Toggling Automation Mode

To toggle Automation Mode on or off, do one of the following:

- Click the **AUTOMATION MODE** button

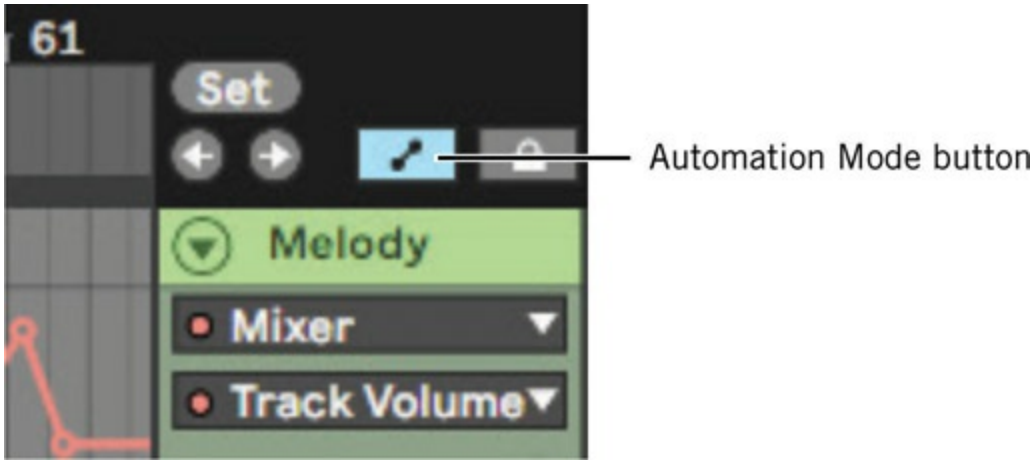


Figure 10.7 The Automation Mode button (blue)

- Choose **VIEW > AUTOMATION MODE.**

- Press the A key on the computer keyboard (with the Computer MIDI Keyboard disabled).

Viewing Automation Envelopes

With Automation Mode enabled, th ere are a number of ways to display automation envelopes.

To view the automation envelope for a visible control:

- Click on any mixer or device control to display the control's envelope on the track. The automation envelope will appear in the track's main automation lane and will be superimposed on top of any audio or MIDI data

To use the choosers to display an automation envelope:

1. Use the **DEVICE** chooser to select the track Mixer or a Device on the track.

2. Use the **AUTOMATION CONTROL** chooser to select the particular control that you would like to view.

3. The automation envelope for the selected parameter will appear in the track's main automation lane and will be superimposed on top of any audio or MIDI data on the track.

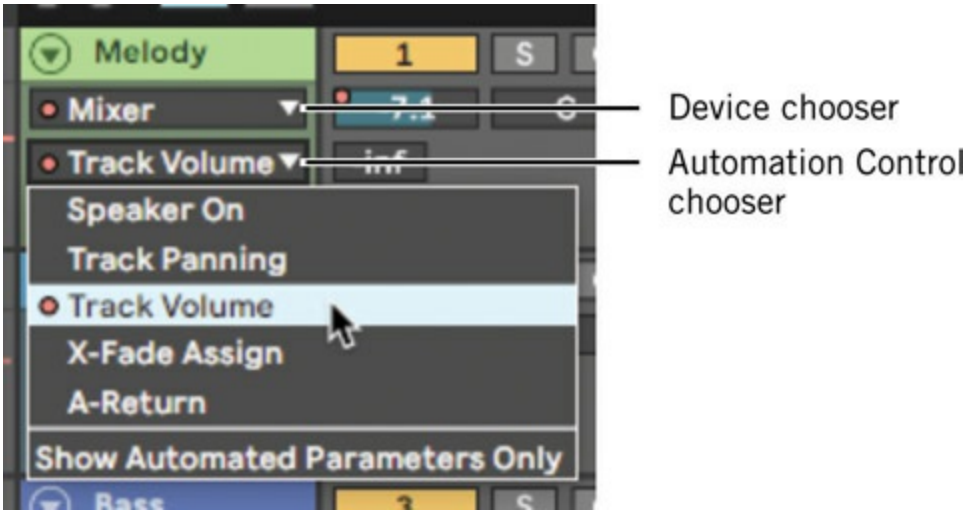


Figure 10.8 Selecting Track Volume from the Automation Control chooser

Oftentimes devices will have dozens of automatable parameters, making the menu in the Device chooser cumbersome to navigate. To simplify the navigation, you can hide all unused parameters in the list, showing automated parameters only.

To limit the Device chooser to devices with automated parameters only:

- Click the Device chooser and select **SHOW AUTOMATED PARAMETERS ONLY** from the bottom of the menu.

You can also hide all automation lanes on a track to prevent accidental changes to an automation graph while working in Automation Mode on other tracks.

To hide automation envelopes on the track's main automation lane:

- Click the Device chooser and select **NONE** from the top of the menu.

Using Automation Lanes

In addition to displaying automation envelopes in the track's main automation lane, you can create additional automation lanes below the track.

To display an automation envelope using an automation lane, do the following:

- Click the **ADD AUTOMATION LANE** button in the bottom corner of the track Title section.

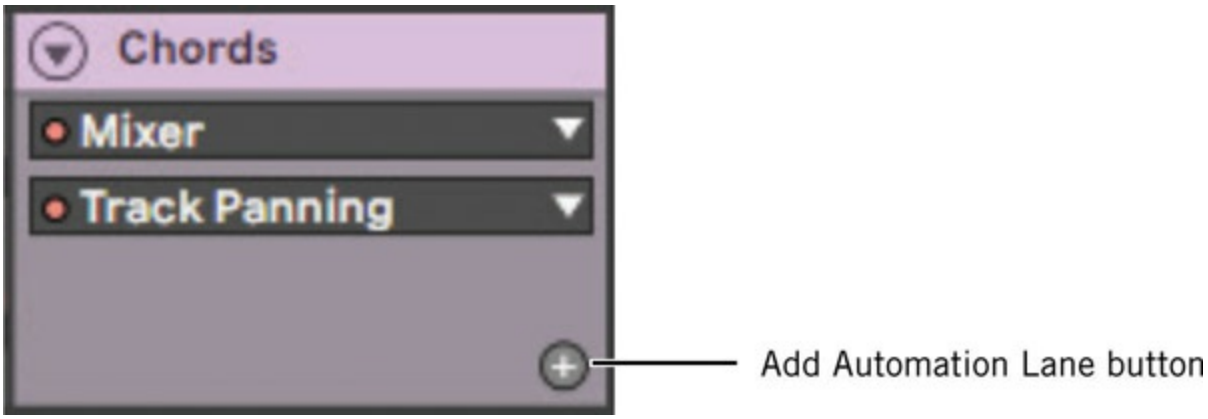


Figure 10.9 The Add Automation Lane button ("+" symbol in the bottom right)

If the Device chooser is set to None, the ADD AUTOMATION LANE button will not be visible.

To display all automation envelopes in their own automation lanes, do one of the following:

- Hold **COMMAND** (Mac) or **CTRL** (Windows) and click the **ADD AUTOMATION LANE** button.

- Right-click on an existing automation lane header and choose **ADD LANE FOR EACH AUTOMATED ENVELOPE**.

To hide an automation lane, do one of the following:

- Click the **REMOVE AUTOMATION LANE** button ("–" symbol to the right of an automation lane header).
- Right-click the automation lane header and choose **REMOVE AUTOMATION LANE**.

To hide an automation lane and all subsequent lanes on the same track, do one of the following:

- Hold **OPTION** (Mac) or **ALT** (Windows) and click the **REMOVE AUTOMATION LANE** button. The current lane and all lanes below it will be hidden.

Drawing Automation Envelopes

In some scenarios, it may be necessary to manually draw an automation envelope in order to achieve a specific result. Ableton Live offers Draw Mode for these situations, which activates a pencil tool for drawing automation on screen.

To enable Draw Mode do one of the following:

- Click the Draw Mode switch in the Control Bar.
- Press the B key on the computer keyboard.



Figure 10.10 The Draw Mode switch (yellow) with Draw Mode active

To momentarily toggle Draw Mode:

- Press and hold the B key on the computer keyboard; release to return to Edit Mode.

With Draw Mode enabled, you can simply click and drag in a track's Envelope Editor to create new automation data.

Drawing with Snap To Grid Enabled

Drawing with the Snap To Grid enabled will always result in "steps" that are the width of the current Edit Grid Value.

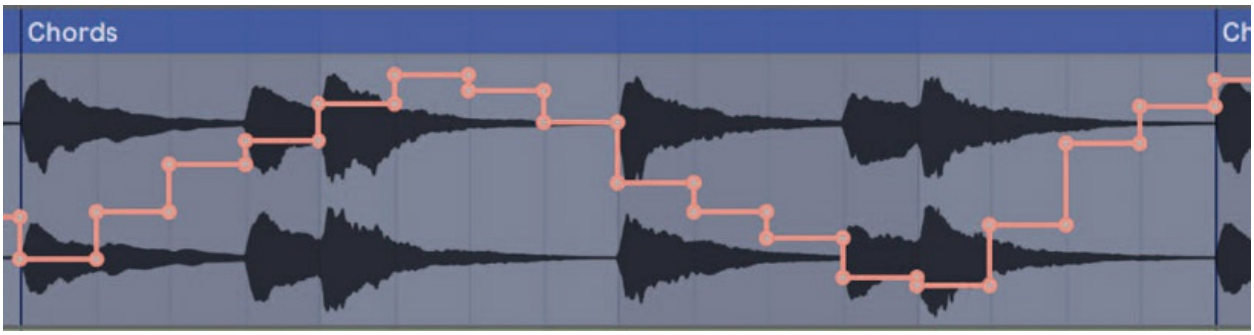


Figure 10.11 "Stepped" automation drawn with Snap to Grid enabled

To draw automation with a finer resolution:

- Hold **SHIFT** and then click and drag vertically on a segment.

Drawing with Snap To Grid Suspended

With Snap to Grid disabled, you can draw a "freehand" automation curve that is not affected by the current Edit Grid value.

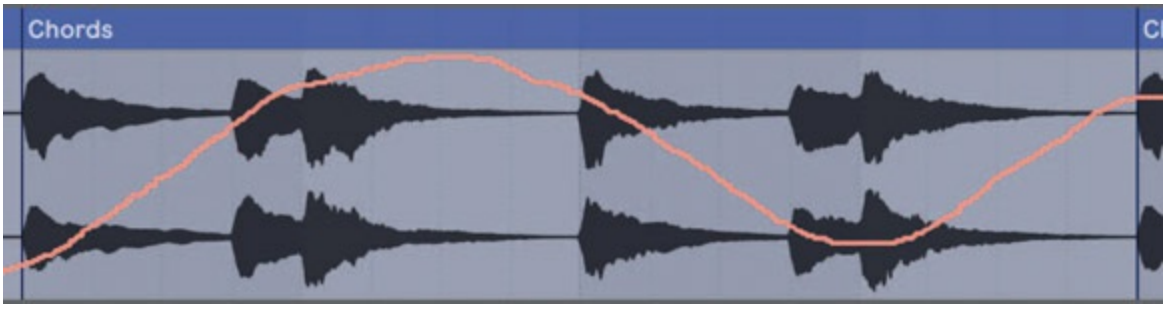


Figure 10.12 "Freehand" automation drawn with Snap to Grid disabled

To disable Snap to Grid, do one of the following:

- Press **COMMAND+4** (Mac) or **CTRL+4** (Windows).
- Right-click in the Arrangement View and set the Edit Grid Value to **OFF**.

To temporarily disable Snap to Grid:

- Hold **COMMAND** (Mac) or **ALT** (Windows) while drawing.

Editing Automation Breakpoints

With Draw Mode off, it becomes possible to use individual breakpoints to create and modify an automation envelope. When creating or editing simple automation moves, it can be convenient to edit the breakpoints directly. As you move a breakpoint, a thin line will appear to show you the position of the breakpoint relative to the beat-time ruler, and a small box will appear to show you the vertical position of the breakpoint.

To add a new breakpoint, do one of the following:

- Click anywhere on a line segment. A new breakpoint will be added at the clicked location.

- Double-click anywhere in the envelope display. A new breakpoint will be added at the clicked location and the automation envelope will conform to include the new breakpoint location.

To delete an existing breakpoint:

- Click on an existing breakpoint. The breakpoint will be deleted and the automation envelope will adjust accordingly.

To move a breakpoint:

- Click and drag on an existing breakpoint. The breakpoint position will move and the automation envelope will adjust as you drag.

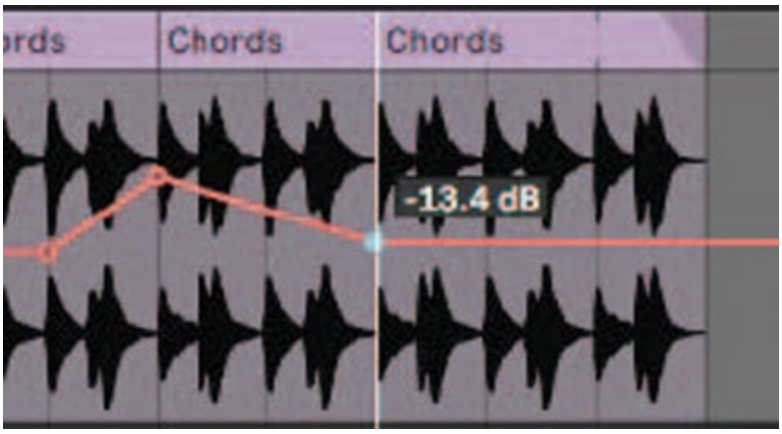


Figure 10.13 Moving a breakpoint

Moving a breakpoint past a neighboring breakpoint will delete that breakpoint from the envelope. Moving the breakpoint back prior to releasing it will restore the neighboring breakpoint.

To restrict the horizontal or vertical movement of a breakpoint:

- Hold the **SHIFT** key when dragging the breakpoint horizontally or vertically.

To adjust the breakpoint value with a finer resolution:

- Hold the **SHIFT** key while dragging the breakpoint vertically.

To suspend grid snapping when moving breakpoints:

- Hold **COMMAND** (Mac) or **ALT** (Windows) while dragging

To move multiple breakpoints simultaneously:

1. Click and drag in the background of the automation lane, not on the automation envelop, to make a time selection that includes the target breakpoints.

2. Click and the drag on any selected breakpoint to move all of the selected breakpoints.

You can also nudge selected breakpoints using the Left and Right Arrow keys.

Making Selections in Automation Mode

When Automation Mode is enabled, you can select an automation range using the same methods you would to select time. You can then carry out many editing tasks on the selected automation range, such as duplicating, copying, and moving the automation selection earlier or later in time.

Locking Envelopes

In the default mode, moving a clip to a new position in the Arrangement Timeline will also move all of the automation associated with the clip. However, it is also possible to lock the automation envelopes so that you can move a clip and have its automation remain in its original position in the Arrangement Timeline.

To enable the Lock Envelopes option, do one of the following:

- Choose **Options > Lock Envelopes**.
- Click the Lock Envelopes switch (lock icon) in the Arrangement View

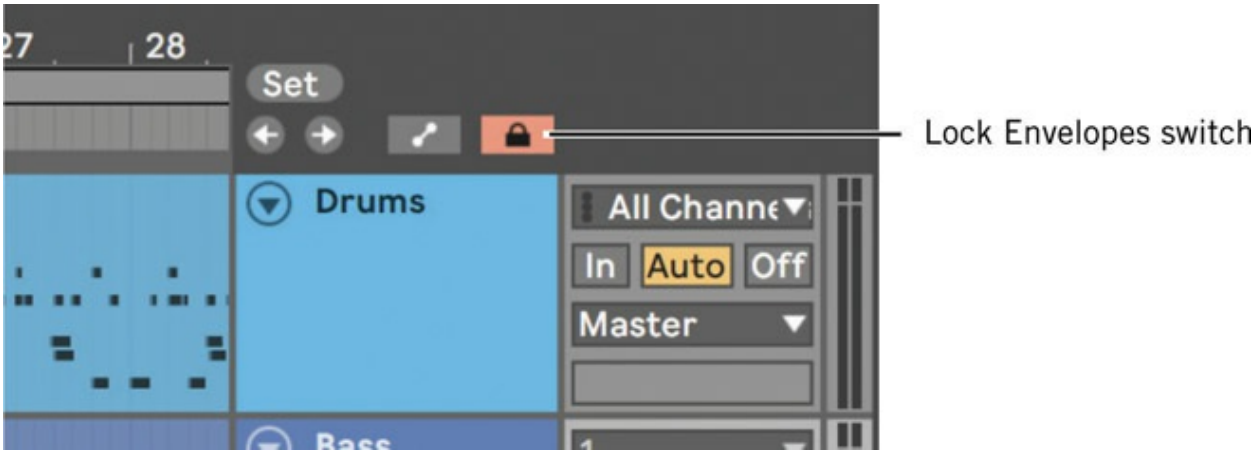


Figure 10.14 The Lock Envelopes switch (red) when enabled

Overriding Automation

In both the Session View and Arrangement View, it is possible to temporarily override automation for a control without modifying the automation data. This can be done by moving a control while not recording automation. The control will be overridden, and its automation LED will turn white. When one or more automation envelopes has been overridden, the **RE-ENABLE AUTOMATION** button (left-pointing arrow) in the Control Bar will light red. Controls will stay overridden until you re-enable automation.



Figure 10.15 The Re-Enable Automation button (red)

To re-enable automation for the entire Live Set:

- Click the **RE-ENABLE AUTOMATION** button.

To re-enable automation for a single control:

- Right-click a control displaying a white LED and choose **RE-ENABLE AUTOMATION** from the context menu.

Deleting Automation

Sometimes you may simply want to delete all of the automation data for a particular control, device, or track. Delete operations are available for all of these situations.

To delete all automation data for a control, do one of the following:

- Right-click on the automated control and select **DELETE AUTOMATION** from the context menu.
- Right-click on an automation lane header and choose **CLEAR ENVELOPE**.
- Select an automated control and press **COMMAND+DELETE** (Mac) or **CTRL+DELETE** (Windows).

Once a control's automation data has been deleted, the automation LED will disappear and the control's value will return to a static value for the duration of the Arrangement timeline.

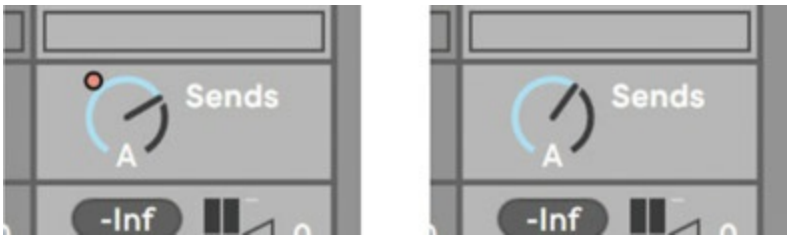


Figure 10.16 A control before deleting automation data (left) and after deleting automation (right)

To delete all automation data for a device:

- Right-click on an automation lane header and choose **CLEAR ALL ENVELOPES OF <DEVICE NAME>.**

To delete all automation data for a track:

- Right-click on an automation lane header and choose **CLEAR ALL ENVELOPES.**

Using Devices for Mixing

As you learned in earlier chapters, Ableton's devices include Live's instruments and effects; third-party plug-ins are also available in the VST or AU format. Devices and plug-ins are both types of software add-ons for Ableton Live that provide essential added functionality, such as effects processing and virtual instrument sound sources for tracks.

Devices Provided with Ableton Live

Ableton Live comes bundled with a variety of device options. Among these devices you will find dynamics processors, EQs, reverbs, delays, flangers, choruses, and more. The number of included devices differs significantly depending on the edition of Ableton Live that you are using. Two commonly used types of processing for audio and MIDI tracks are equalizers (including

graphic and parametric EQs) and dynamics processors (such as compressors and gates). Both Ableton Live Standard and Ableton Live Suite feature the following:

- EQ Three
- EQ Eight
- Compressor
- Limiter

EQ Three

EQ Three is a simple, DJ-style equalizer for adjusting the frequency spectrum of audio material in Ableton Live.

To add the EQ Three device to an Ableton Live track, first locate the device in the Browser under **Audio Effects > EQ Three**. Then drag and drop the device onto a track (or double-click the device if the destination track is currently selected). The EQ Three device window will appear in the Detail View.



Figure 10.17 The EQ Three equalizer device

EQ Eight

EQ Eight is an advanced equalizer for adjusting the frequency spectrum of audio material in Ableton Live.

To add the EQ Eight device to an Ableton Live track, first locate the device in the Browser under **Audio Effects > EQ Eight**. Then drag and drop the device onto a track (or double-click the device if the destination track is currently selected). The EQ Eight device window will appear in the Detail View.

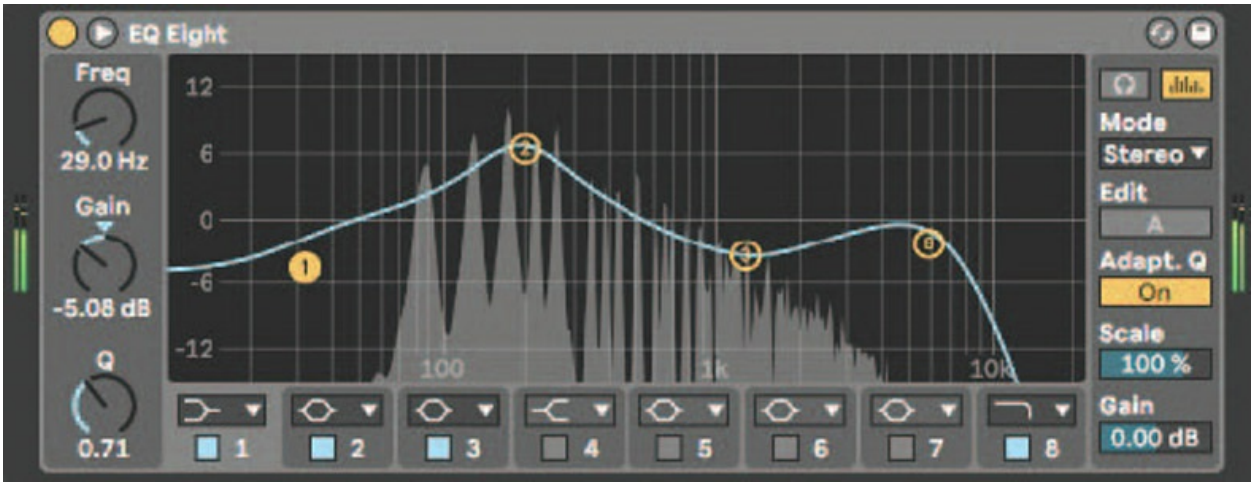


Figure 10.18 The EQ Eight equalizer device

Compressor

The Compressor device can be used to control dynamic levels, using standard attack, release, threshold, and ratio controls.

To add the Compressor device to an Ableton Live track, first locate the device in the Browser under **Audio Effects > Compressor**. Then drag and drop the device onto a track (or double-click the device if the destination track is currently selected). The Compressor device window will appear in the Detail View.

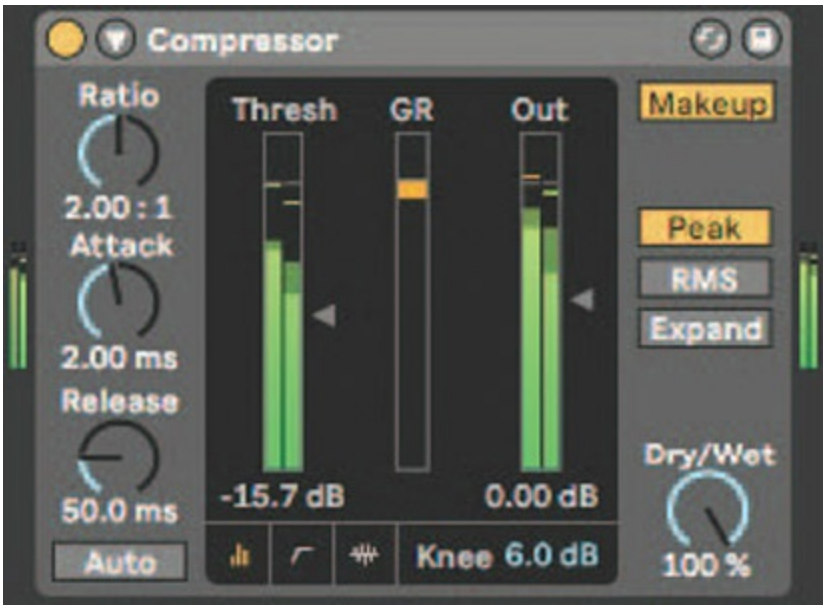


Figure 10.19 The Compressor device window

Limiter

The Limiter device is similar to a compressor, but completely prevents level from going above the ceiling value. It is typically used on complete mixes, but can also be used on individual audio and MIDI tracks.

To add the Limiter device to an Ableton Live track, first locate the device in the Browser under **AUDIO EFFECTS > LIMITER**. Then drag and drop the device onto a track (or double-click the device if the destination track is currently selected). The Limiter device window will appear in the Detail View.



Figure 10.20 The Limiter device window

Backing Up Your Session

Once you've finished crafting your mix with automation and device processing, you can safeguard your work by creating a backup copy. Creating backups is critical for archival and disaster-recovery purposes. Because Ableton Live projects are stored electronically, it is possible to accidentally lose your work by deleting or overwriting files, having a file become corrupt, encountering a virus, or having a drive fail.

Considerations for Backup and Archive Copies

Some of the best protection measures include creating multiple copies of your files, using a separate drive for backup copies, and storing copies in the cloud to protect against disasters such as fire or flood. The more valuable your projects, the more robust your backup plans should be. At a minimum, you should create a backup project upon completing any work that would be difficult or time-consuming to re-create, especially if the recording has significant value or importance to you or your clients.

Whenever creating a backup or archive copy, be sure to include any media files (audio and video) that are referenced by the Set. Keep in mind that these files may not be included in the original project folder.

Saving an Archive Copy

One method you can use to create a copy of your Set with all related files is to follow a two-step process. First, use the **Save a Copy** command, which creates a copy of the Set in a new project folder. Then open the copy and use the **Collect All and Save** command. This command allows you to copy all project media into the current project folder, creating a self-contained archive of the project. This archive project folder can then be stored separately or transferred to a different location, such as on an external drive.

Step 1: Apply the Save a Copy Command

By the time you've completed work on a project, your Live Set will likely include samples that are referenced from many different locations on your system. While this generally does not present a problem for the original project, it does mean that moving the project (or making a copy of it on a separate drive) can inadvertently separate the Set from the associated samples and other referenced media.

To reduce the likelihood of losing files, Ableton Live can copy all of a project's external samples directly into the project folder, using the **Collect All and Save** command (see below). However, this process will duplicate any referenced media, and can greatly increase the size of the project folder.

To prevent the needless duplication of files on your system, it is best to create a new project folder for your archive prior to using Collect All and Save. This is where the **Save a Copy** command comes in.



Although you could also use the Save Live Set As command for the workflow described here, you would not be able to specify the name of the project folder in the process.

To create a copy of your Set in a new project folder for archiving, do the following:

1. Open the Set that you want to archive.

2. Choose **File > Save a Copy** to create a new copy of the Set within a new

project folder.

3. In the **SAVE A COPY** dialog box, navigate to an appropriate location for your archive, such as an external drive.

4. Provide a meaningful name for the archive copy at the top of the dialog box.



By default, Ableton Live appends the word "Copy" to the end of the Set name for use as the new project name. It is good practice to replace this default name, using a meaningful naming convention of your own, such as appending the word "Archive" to the name and including the archive date.

5. Click **SAVE** to complete the process.

Ableton Live will create a new Set file, inside of a new project folder, using the name and location that you've specified. Note that you will still be working in the original Set after completing this command.



Any changes you make to a Set after completing the Save a Copy command will not be included in the copy. To modify the copy, you will need to close the original set and then open the copy you've just created.

Step 2: Apply the Collect All and Save Command

As discussed in Chapter 4, the Collect All and Save command simultaneously saves your current Live Set and packages all of the associated audio files into the project folder. The Collect All and Save dialog box includes a variety options that you can use to specify which media files to copy.

To use the Collect All and Save command for your archive copy, do the following:

1. From within your original Set file, choose **FILE > OPEN LIVE SET**.

2. In the **OPEN DOCUMENT** dialog box, navigate to the location where you

saved the archive copy of the Set (within the new project folder of the same name).

3. Select and open the archive Set file (*filename.als*).

You may be prompted to save changes to the original Set, if you've made any unsaved changes to the file. Be sure to keep any important changes you may have made to the original prior to creating the archive copy.

4. With the archive copy open, choose **File > Collect All and Save**. The Collect All and Save dialog box will display.

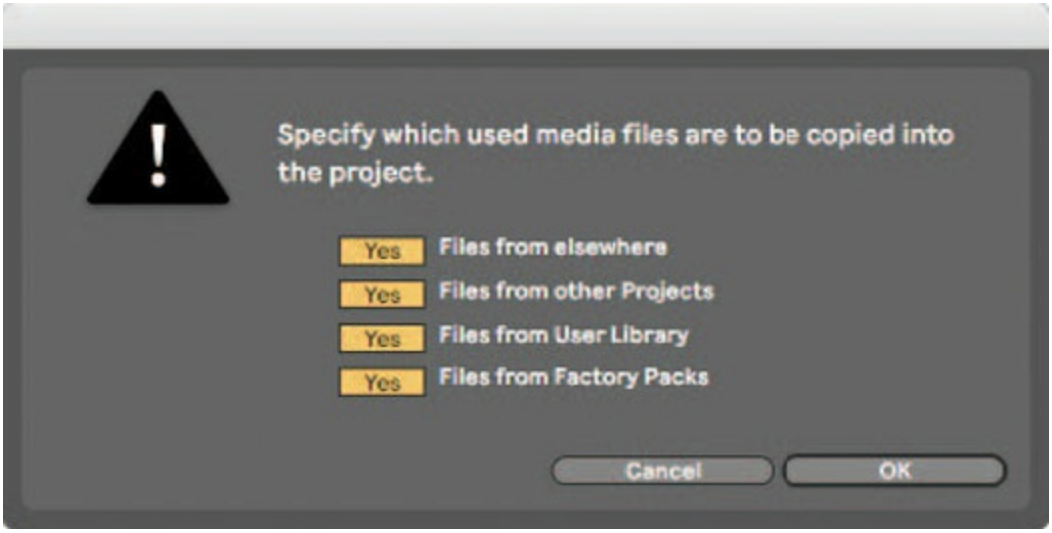


Figure 10.21 The Collect All and Save dialog box

5. Configure the file collection options as desired. (See "File Collection Options for Collect All and Save" below for detailed descriptions of each.)

6. Click OK when you are finished configuring the options as desired. Any external samples from the selected locations will be copied directly into the archive project folder.

File Collection Options for Collect All and Save

When using the Collect All and Save command, a number of options will be available that allow you to consolidate external samples into the project

folder. The available file collection options are displayed in the Collect All and Save dialog box and include the following:

- **Files from elsewhere.** This option selects media files used in the current project from any locations besides Factory Packs, the User Library, and other projects and then copies them into the current project folder. This is useful to collect any media files you may have imported from an external source (not including samples imported from libraries installed with Ableton Live).

- **Files from other Projects.** This option selects media files used in the current project from any other projects and copies them into the current project folder. This is useful to collect any samples you may have recorded into a Set prior to using the **SAVE A COPY** command.

- **Files from User Library.** This option selects media files used in the current project from your User Library and copies them into the current project folder. This is useful to collect any media files (samples or clips) you've imported from items you've previously saved in your User Library.

- **Files from Factory Packs.** This option selects media files used in the current project from the Factory Packs installed with Ablton Live and copies them into the current project folder. This can be done to protect imported samples against accidental loss or corruption in their default locations.



Multisampled instruments and libraries can be extremely large. Copying them will greatly increase the size of your project folder. Avoid making duplicate copies of files that are installed with Ableton Live, if possible, as this is generally not necessary for a Set to play back properly from another system.



After collecting the files for your archive, you can reduce the amount of disk space it requires by creating a Pack. This copies the contents of the project folder into a single compressed file.

Creating a Stereo Mixdown

Mixing down is the process of recording the output from multiple tracks to a stereo file. This process is also commonly referred to as *bouncing* the project or *exporting a mix*. Mixing down is often the last phase of music production, although you can export your audio at any time to create a complete mix as a stereo file.

Mixdown Options

The most commonly used mixdown option in Ableton Live is the **File > Export Audio/Video** command. This option is a fast and easy way to create a bounce, requiring little to no setup.

Advanced users also frequently use the *bounce to tracks* technique. Bouncing to tracks is the process of recording the output of any or all of the tracks to a new audio track within the project. This process is also commonly referred to as creating an internal bounce or internal layback.

For the purposes of this course, we will focus on the Export Audio/Video option.

Considerations for Exporting a Mix

When exporting a mix with Ableton Live, it is important to recognize that the exported mix will capture all audible information in your Set just as you hear it during playback.

The following principles apply to exporting a mix:

- **The mix will include only audible tracks.** What you hear during playback is exactly what will be included in the mix. Any tracks that are deactivated will not be included. Conversely, any tracks that are soloed will be the *only* tracks represented in the mix.

- **The created file(s) will be based on the Rendered Track option selected in the Export Audio dialog box.** This chooser in the dialog box lets you select which track or tracks to render. When creating a

mixdown, the Rendered Track should be set to Master.

- **The mix file will be a "printed" version of your Set.** Devices, sends, and external effects will be applied permanently in the mix. Listen closely to your entire Set prior to completing a bounce to ensure that everything sounds as it should. Pay close attention to levels, being sure to avoid clipping.

- **The mix will be based on the Timeline selection.** If you have an active selection, Ableton Live will bounce the length of the selection automatically. If no selection is present, the Export command will create a mix from the start of the first clip in the arrangement to the end of the longest track.

Exporting Audio

The Export Audio/Video command allows you to mix your entire Set directly to a stereo file on disk. The corresponding dialog box lets you set the bit depth, file format, and sample rate for the exported file. The export will automatically be rendered offline for faster-than-real-time mixdown.

Supported File Types

The file types that you can create from the Bounce to Disk command include the following:

- **WAV.** This is the default file format for Windows- and Mac-based audio production.

- **AIFF.** This file format is primarily used on Mac systems. Use the AIFF format if you plan to import the bounced audio into Mac applications that do not support the WAV format.

- **FLAC.** This is a lossless file format that can create smaller audio files without a perceptible reduction in sound quality.

- **MP3.** This file format can create smaller audio files for Internet streaming and portable devices. Use this file format to balance file size

against audio quality.

Using the Export Audio/Video Command

The Export Audio/Video command combines the outputs of all currently audible tracks routed to the Master (or other selected option) to create a new audio file on your hard drive.

To export a mix of all currently audible tracks, do the following:

1. Adjust track output levels and finalize any automation. Any devices or effects settings that are active on your tracks will be permanently written to the exported media.

2. Make sure that all of the tracks you want to include in the mix are audible. To mix down all tracks in your Set, verify that no tracks are soloed or deactivated.

3. Verify that the output of each track is routed to the same destination (typically the Master track). Use the **Output Type Chooser** as needed to set the track outputs.

4. Choose **File > Export Audio/Video**. The Export Audio/Video dialog box will appear.

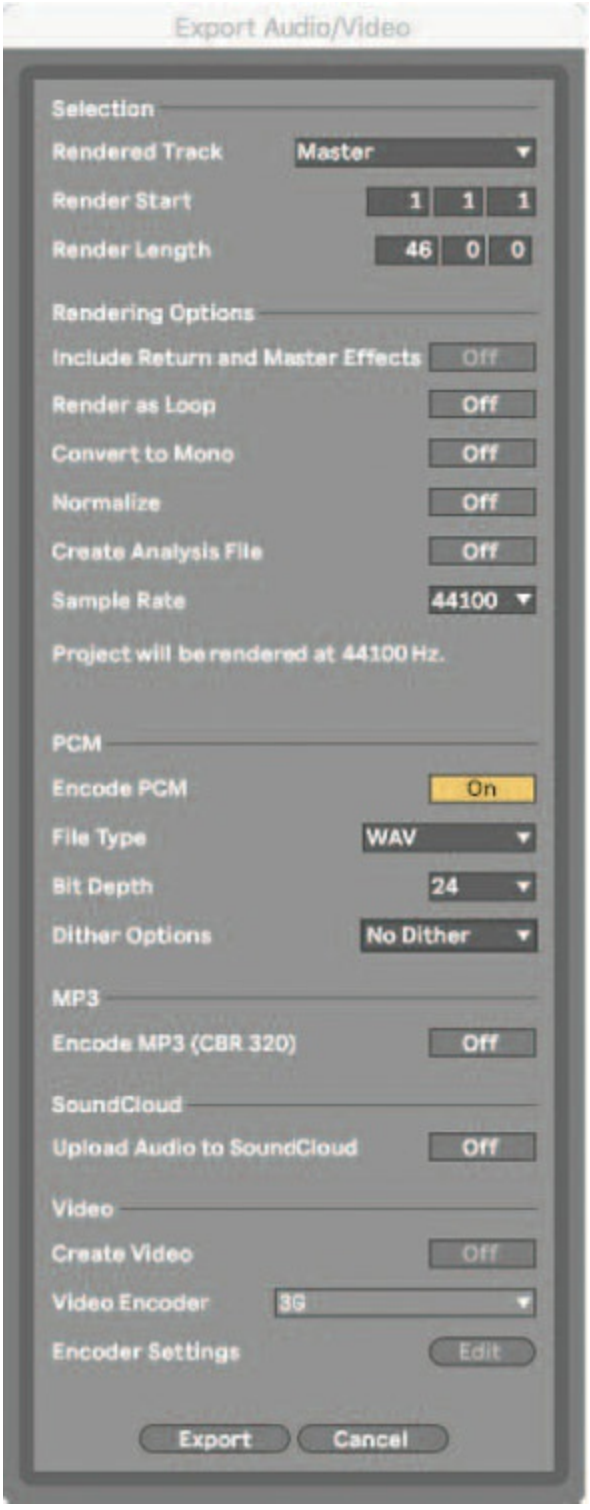


Figure 10.22 The Export Audio/Video dialog box

5. Verify that the **RENDERED TRACK** is set to Master (the default).

6. The **RENDER START** and **RENDER LENGTH** fields will already be

populated by the current timeline selection (if one exists), or the start of the first clip in the arrangement until the end of the longest track (if no selection is present). You can also manually edit these fields to change the start and length of the exported mix.

7. Choose the desired sample rate for the exported file from the **SAMPLE RATE** pop-up menu. Higher sampling rates will provide better audio fidelity but will also increase the size of the resulting file(s).



If you plan to burn your mix directly to CD without further processing, choose 44.1 kHz as the sample rate for the bounce.

8. Enable the **ENCODE PCM** option and choose the desired file type for your exported file from the **FILE TYPE** pop-up menu. Available options include WAV, AIFF, and FLAC.



When Encode PCM is on, a lossless audio file is created. WAV, AIFF, and FLAC formats are available for PCM export.



You can also export to a WAV, AIFF, or FLAC file and simultaneously create an MP3 file. To do so, enable both the Encode PCM option and the Encode MP3 (CBR 320) option.

9. Choose the desired bit depth for the exported file from the **BIT DEPTH** pop-up menu.

    - Choose 16 if you plan to burn your mix to CD without further processing.
    - Choose 24 when you want to create a final mix that will be mastered separately.
    - Choose 32 for ultra-high-resolution files that will undergo further processing or editing.

The standard bit depth resolution for compact discs is 16 bits.

10. Select other options as desired.

11. After confirming your settings, click the **EXPORT** button. A Save dialog box will appear.

12. Specify a file name and save location for your exported file(s). By default, the exported file(s) will be named after the Ableton Live Set.

When performing an export, Ableton Live processes the export without audio playback. A progress window will appear, displaying the percentage of the export that has been completed.

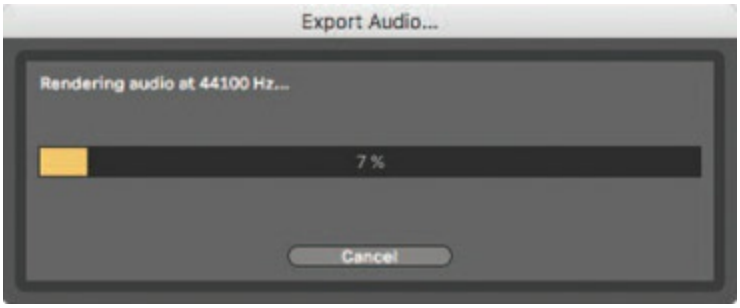


Figure 10.23 The Export Audio progress window

Creating a CD-Compatible Mixdown

Creating a CD-ready mixdown from an Ableton Live Set will require a bit-depth reduction during the bounce, to create a 16-bit file. To maintain the highest possible audio quality when reducing bit depth, it is necessary to apply *dither* to your audio. Dither is a form of randomized noise used to minimize quantization artifacts in digital audio systems. Quantization artifacts are most audible in the lowest parts of a signal's dynamic range, such as during a quiet passage or fade-out.

Introducing dither to a signal helps to reduce quantization artifacts and preserve dynamic range by adding very low-level noise. Dither provides a trade-off between signal-to-noise performance and quantization artifacts. Proper use of dither can create better subjective performance out of 16 bits

compared to audio without dither.

Adding Dither to the Mixdown

Dither processing can be added in Ableton Live during the export process when creating a mix at 16 or 24 bits. In these cases, the 32-bit floating-point output from Live will be reduced to create an audio file on disk at a lower bit depth. This in turn will reduce the dynamic range in the file on disk. Adding dither processing will help preserve some of the dynamic range that would otherwise be lost.

To add dither during a mixdown:

1. Choose **File > Export Audio/Video**. The Export Audio/Video dialog box will appear.

2. In the PCM section of the dialog box, set the **Bit Depth** chooser to 16 or 24.

For a CD-compatible mixdown, set the Bit Depth to 16.

3. Use the Dither chooser to select the desired dither option.



Figure 10.24 Dither options in the Export Audio/Video dialog box

4. Set other options in the dialog box as desired and click **EXPORT** to complete the mixdown.

Types of Dither

When the dither is active in the Export Audio/Video dialog box, the Triangular option is selected by default. This mode is useful if there is a possibility that additional processing will be applied to the mix file, as it adds very little background noise. Rectangular mode introduces an even less background noise, but at the expense of additional quantization error.

The three Pow-r dither modes are generally the most effective at preserving dynamic range, offering successively higher amounts of dithering, but with the noise pushed above the audible range.



If you will be sending your mixed file to a mastering house, you'll want to keep it at 32-bit floating-point resolution to retain the highest quality for the mastering process. In this case, you should NOT apply dither.

Uploading a Mix to SoundCloud

Ableton Live lets you upload your exported mix directly to your SoundCloud account during the export process.

To upload your mix to SoundCloud, do the following:

1. Choose **FILE > EXPORT AUDIO/VIDEO**.

2. Enable the **UPLOAD AUDIO TO SOUND CLOUD** option near the bottom of the dialog box.

3. Click **EXPORT**. The export process will proceed as described above, after which a **SHARE ON SOUND CLOUD** dialog box will appear.

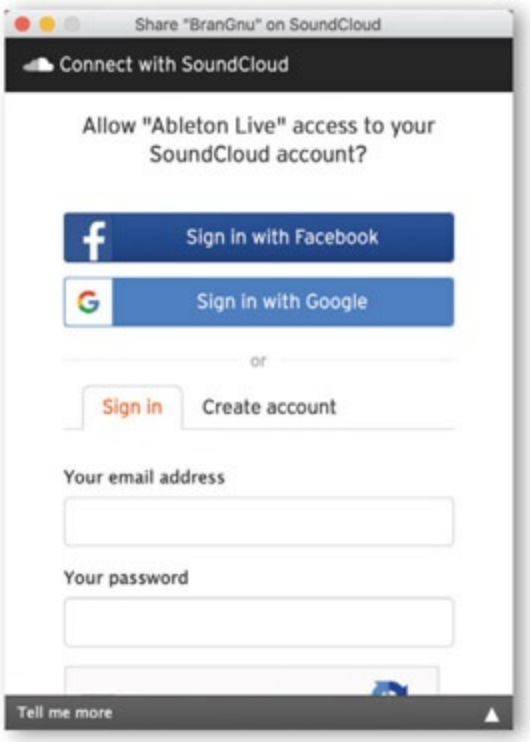


Figure 10.25 The Share on SoundCloud dialog box

4. Log in to your SoundCloud account and click **SIGN IN AND CONNECT** at the bottom of the dialog box. A copy of your mix file will automatically be uploaded to SoundCloud.

Review/Discussion Questions

1. What term is traditionally used to describe an audio patch point that allows a signal processor to be inserted directly into the signal path on a track? What term does Ableton Live use for this feature? How many of these does Ableton Live support on each track? (See “Basic Mixer Terminology” beginning on page 268.)

2. What term is used to describe a signal path carrying a mix output of one or more tracks routed for parallel processing? How can this signal be returned to the sending device? (See “Basic Mixer Terminology” beginning on page 268.)

3. What menu would you use to display or hide the Arrangement View window? What keyboard shortcut can you use to toggle between the Session View and Arrangement View? (See “Mixing in the Session View” beginning on page 269.)

4. What menu command can you use to display or hide the Sends Section in the Session View? (See “Showing/Hiding Sections” beginning on page 270.)

5. Which Ableton Live button can be enabled to allow track controls to record automation in real time when playing a Set? (See “Recording Arrangement Automation” beginning on page 274.)

6. Can you record automation in Ableton Live when the transport is not in Arrangement Record mode? Can you record automation to a track that is not Armed for recording? (See “Recording Arrangement Automation” beginning on page 274.)

7. What are some ways that you can display automation envelopes on a track? Which mode must be enabled before automation can be displayed? (See “Viewing Arrangement Automation” beginning on page 275.)

8. What are some ways that you can add, move, or delete automation breakpoints? (See “Editing Automation Breakpoints” beginning on page 279.)

9. What are some Ableton device options that are available for EQ and dynamics processing? (See “Using Devices for Mixing” beginning on page 282.)

10. Why is it important to back up your Ableton Live projects? What are some ways in which your Ableton Live work can be lost accidentally? (See “Backing Up Your Session” beginning on page 285.)

11. What is the recommended process for creating a project archive? How is the Collect All and Save command different from other save options, in terms of the files that are saved? (See “Saving an Archive Copy” beginning on page 285.)

12. What are some considerations for creating a stereo mixdown in Ableton Live? How is the mixdown file affected by soloed or muted tracks? How is it affected by a timeline selection? (See “Considerations for Exporting a Mix” beginning on page 288.)

13. What command lets you mix your entire Ableton Live Set directly to a stereo file? What file types are supported for audio files created with this command? (See “Exporting Audio” beginning on page 289.)

14. How can you upload a mix directly to SoundCloud from Ableton Live? (See “Uploading a Mix to SoundCloud” beginning on page 293.)



To review additional material from this chapter and prepare for certification, see the Ableton Live 101 Study Guide module available through the Elements|ED online learning platform at ElementsED.com.

Exercise 10

Automating Shared Effects and Exporting a Stereo Mix

Activity

In this exercise, you will finalize your project. You will start by creating a shared effect using a delay plug-in on a return track. Then you will automate the delay effect using send automation on the Noise Sweep track. Next you will set basic levels for your tracks, and you'll finish by exporting the mix to a stereo file.

Duration

This exercise should take approximately 15 to 20 minutes to complete.

Goals/Targets

- Add a return track to create a delay effect
- Use automation to set levels for your mix
- Use a limiter to maximize output levels
- Export your final mix

Media Files

To complete this exercise, you may need to use various media files included in the Media Files 2019-Live101 folder. You should have

downloaded the media files in Exercise 4.

If needed, you can re-download the media files by going to www.halleonard.com/mylibrary and entering your access code (printed on the opening page of this book). From there, click the Download link for the Media Files 2019–Live101 listing in your My Library page. The Media Files folder will begin transferring to your Downloads folder.

Getting Started

You will start by opening the Live Set you completed in Exercise 9. If that Set is not available, you can use the Ex09 Sample file in the 01 Completed Exercises folder within the Media Files 2019-Live101 folder.

Open the Live Set and save it as Exercise 10:

1. Do one of the following:

   - Open the Set file that you created in Exercise 9 (Storage Drive/Folder > Live101-XXX > Exercise09-XXX.als).

   - Alternatively, you can use the Ex09 Sample file (Media Files 2019-Live101 > 01. Completed Projects > Ex09 Sample.als).

2. Choose **File > Save Live Set As** and name the Set *Exercise10-XXX*, keeping the Live Set inside the original Project folder. (Move the Set into your Live101-XXX folder if working from the sample file.)

Creating a Delay Effect

In this part of the exercise, you will create a return track to function as the delay return. Then you will assign a delay device on the track. Finally, you will use the send controls on the **Chords** and **Noise** tracks to create a delay effect for the audio from those tracks.

Adding the Return Track

Use the following steps to add a new return track and add a delay device to the track.

Create a delay return track:

1. If necessary, press **TAB** to switch to the Session View.

2. Create a new return track by pressing **OPTION+COMMAND+T** (Mac) or **CTRL+ALT+T** (Windows). A new return track labeled A Return will appear to the left of the Master track.

3. Click the new return track's Title Bar to select the track.

4. Press **COMMAND+R** (Mac) or **CTRL+R** (Windows) to rename the track. Change the track name to Delay and press **RETURN** or **ENTER** to commit the change.

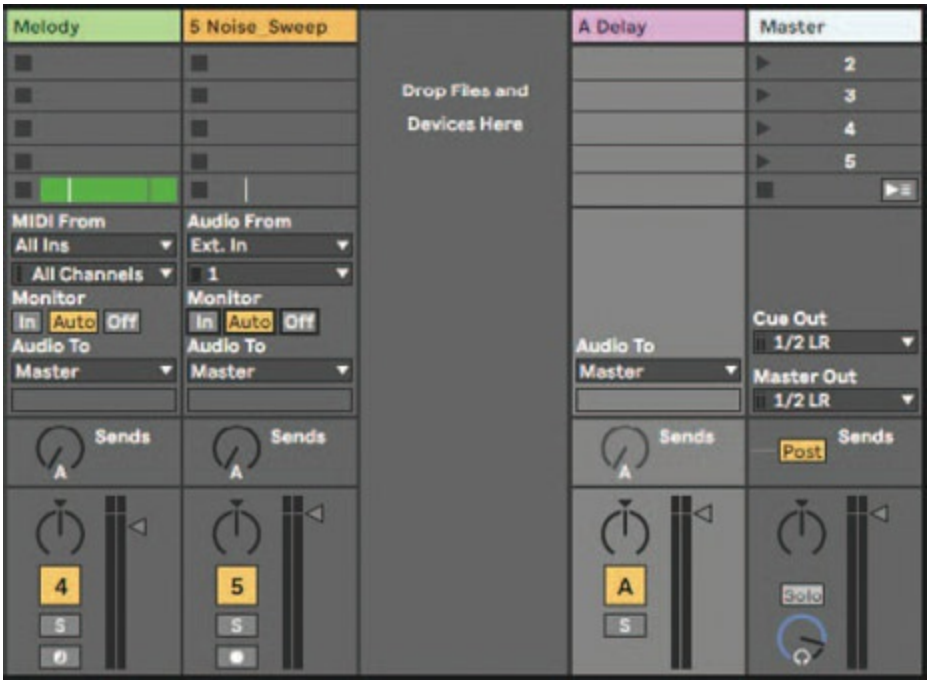


Figure 10.26 The Delay Return track to the left of the Master track

Configure the delay device on the track:

1. Locate the Delay device in the Browser under the **AUDIO EFFECTS** category.

A quick way to find a specific device is to press COMMAND+F (Mac) or CTRL+F (Windows) and type the first few letters of the device name into the search field.

2. Click the disclosure triangle next to the Delay device in the Browser Content pane to reveal the available options; then expand the Clean Delay category to see the available presets.

3. Double-click the **DOTTED EIGHTH NOTE** preset to add the device to the Delay Return track.

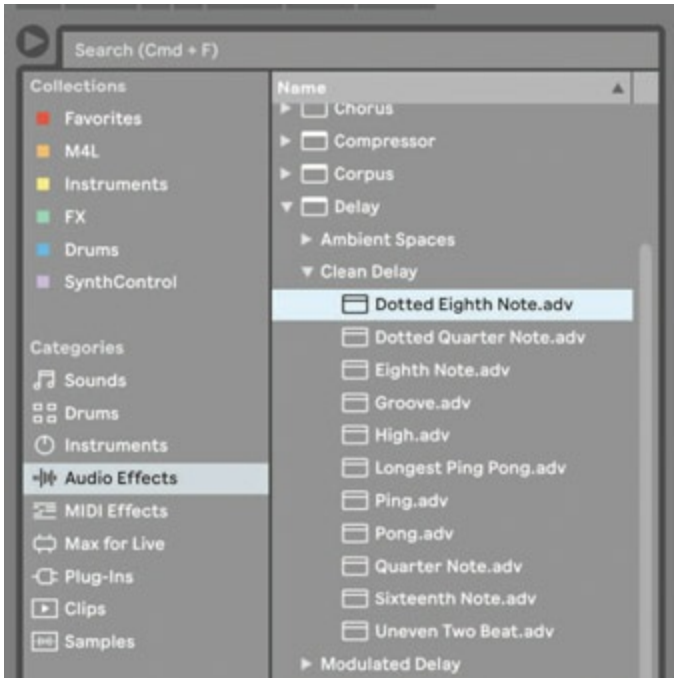


Figure 10.27 Selecting the Dotted Eight Note preset for the Delay device in the Browser

The Delay device window will appear in the Detail View, configured with settings from the Dotted Eighth Note preset.

4. Click on the **FEEDBACK** control and reduce the Feedback amount to 30%.

5. Click on the **DRY/WET** control and increase the Dry/Wet ratio to 100%.

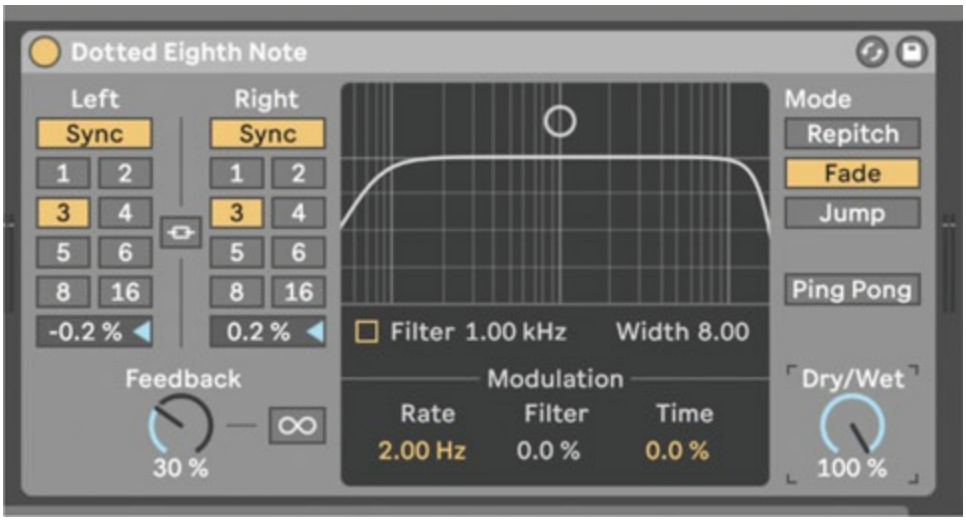


Figure 10.28 The Delay device with the Feedback and Dry/Wet parameters set

Set the Send Levels

Use the following steps to set appropriate send levels for the target tracks in your project.

Raise the send levels on the Chords and Noise tracks:

1. Locate the **SEND A** control for the Chords track in the Session View.
2. Click on the send control for the track and raise the level to around −10.0 dB.

In Session View, the current Send Level is displayed in the Status Bar area at the bottom of the screen when your mouse cursor is positioned over a Send control.

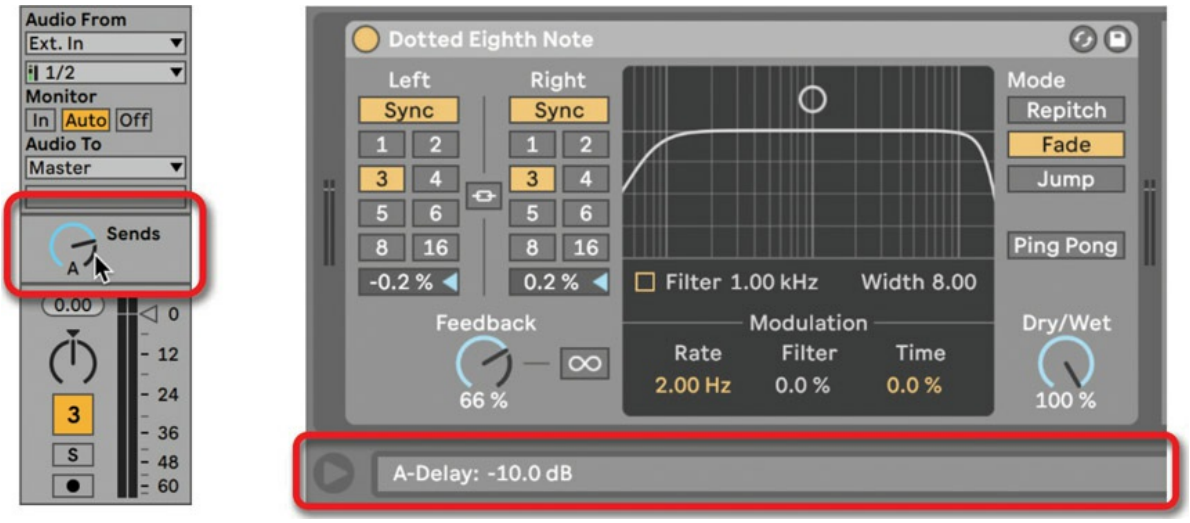


Figure 10.29 Send control set to −10.0 dB (left); Status Bar display (right)

3. Press **TAB** to switch to the **ARRANGEMENT VIEW**.

4. If the Mixer Section is not showing, click the Show/Hide Mixer Section button (M) at the bottom right of the Arrangement View to show it.

5. Locate the **SEND** A control for the Noise track. This will be the bottom field in the Mixer section of the track and will be labeled –inf by default.

6. Click on the field and drag up to set the Send level to around −3.0 dB.



Figure 10.30 Adjusting the Send level in the Arrangement View

Recording Automation and Setting Levels

In this part of the exercise, you will record automation during playback. Then you will set basic levels from the Volume sliders for all the tracks in your mix.

Display automation envelopes in the Arrangement View:

1. If necessary, press the M key to disable the Computer MIDI Keyboard.

2. Press the A key on the computer keyboard to enable Automation Mode.

3. Click to position the Insert Marker in the last few bars of the Arrangement (just before 1:00) where the final chord clip plays.

Record Automation

In the next series of steps, you will be automating the track volume of the Delay return track so that the delay echoes do not extend past the end of the song. You may need to practice the timing a few times to get the desired results.

To practice the timing without recording Arrangement Automation, you can simply move the control while playing through the Set. After each practice pass, double-click the Track Volume slider to reset the value to 0.

Record volume automation for the Delay return:

1. Click the **UNFOLD TRACK** button (triangle) next to the track name on the Delay Return track to expand the track height for better access to the controls in the Mixer section.



Figure 10.31 The Unfold Track button on the Delay Return track (shown folded in)

2. Verify that the Delay return's Track Volume is currently set to 0. This is displayed in the middle field in the Mixer section.

3. Click the **AUTOMATION ARM** button in the Control Bar to arm automation recording.

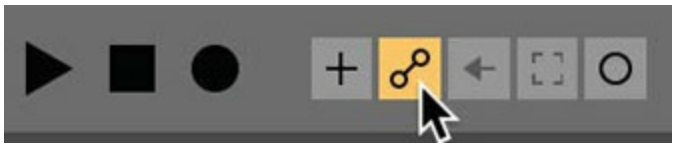


Figure 10.32 Automation Arm button in the Control Bar

4. Click a few bars before the end of the song in any track to set the arrangement position.

5. Click the **ARRANGEMENT RECORD** button (solid circle) in the Control Bar and prepare to record your automation.

6. As the playback position approaches the end of the song, fade out the Delay return track by gradually reducing the Volume slider control (click and drag downward on the Volume display field) so that the delay echoes do not extend past 1:00 (one minute).



Figure 10.33 Track Volume automated on the Delay return track.

7. Continue recording automation for a few seconds with the track volume pulled down to -inf.

8. Press the **SPACEBAR** when finished to stop playback.

9. Press the **SPACEBAR** again to audition the automation that you recorded. If you are not satisfied with the results, you can click the Arrangement Record button and try again.

Set Relative Levels for the Tracks

In the next series of steps, you will adjust the levels for individual tracks

using the tracks' Volume sliders.

Set levels for your mix:

1. Press **TAB** to return to the Session View.

2. Begin playback and adjust the levels of each track to achieve a clear, intelligible mix with appropriate levels from the audio tracks, MIDI tracks, and delay return. Following are some suggestions:

   - Start by lowering all of the Volume sliders to around −6 dB. This will give you room to increase the levels on tracks that are too quiet.



Any easy way to change the levels of all tracks in the Set is to select them all and then move the Volume slider on any one track.



You will not be able to affect the volume on the Delay track, since it has been automated. The Volume slider will return to the automated levels on playback.

   - The drums may be a bit too prominent in the mix. Try lowering the Volume slider on the Drums track by an additional 3 dB.

   - The Noise sound comes in a bit too quiet. Try raising the Volume slider on the noise track by 4 to 6 dB.

Exporting the Mix

In this part of the exercise, you will maximize the output levels for the mix using a limiter plug-in on the Master track. Then you will use the Export Audio function to create a stereo file from your mix.

Add a Limiter

Use the following steps to add a limiter to the Master track.

Apply a limiter to the mix:

1. Locate the Limiter device in the Browser under **AUDIO EFFECTS** > **LIMITER**.

The Limiter device will not be available when using Ableton Live Lite. In this case, you can use the Multiband & Limiter option instead, found in Audio Effect Rack > Mixing & Mastering.

2. Drag and drop the Limiter device onto the Master track. The Limiter device window will appear in the Detail View.

3. Press the SPACEBAR to begin playback.

4. In the Limiter device window, set the **CEILING** slider to around –0.10 dB. This sets the maximum allowable level for the limiter's output to prevent clipping.



Figure 10.34 The Limiter device with Ceiling set to −0.10 dB

If using the Multiband & Limiter option, start by setting the Bass, Mid, High, and Rack Volume controls to 0 dB. Then set the Limiter Ceiling control to –0.10 dB.

5. While listening to the mix, adjust the **GAIN** slider to achieve a healthy output level without artificially compressing the dynamic range. A setting of around 3 to 6 dB will likely be adequate for this mix.

In the Multiband & Limiter device, use the Multiband Macro Control to set the Master Output level.

Create a Stereo Mixdown

Use the following steps to export a stereo mixdown file for your project.

Export the mix to a stereo file:

1. Press **TAB** to return to the Arrangement View.
2. Make a selection on any track, beginning at the start of the song and extending to slightly past 1:00.
3. Choose **FILE > EXPORT AUDIO/VIDEO** to open the Export Audio dialog box.
4. Select the following options for your export:
   - File Type: WAV
   - Bit Depth: 16 Bit
   - Sample Rate: 44.1 kHz
5. Leave the other options set to their defaults (or as directed by your instructor) and click the **EXPORT** button. The Save dialog box will appear.
6. Name your file **YOUR NAME-FINAL MIX**.
7. Navigate to an appropriate save location, making note of the drive location and file path; then click **SAVE** to export your mix.

Finishing Up

Congratulations! Over the course of these exercises, you have created an Ableton Live Set from scratch, imported and edited audio, recorded audio and MIDI, added effects, recorded automation, and exported the result to a stereo file.

1. When your export completes, save your Set and quit Ableton Live.

2. Navigate to the location of your exported file and verify that it plays back as intended.

   In classroom settings, your instructor may require that you copy this file and/or your Project Folder to a learning management system, flash drive, or shared network location to submit it for a grade.

For more hands-on experience, see the Final Project section of this book.

Final Project

Creating a Musical Arrangement

Activity

In this hands-on project, you will complete a Live Set featuring a roughly 90-second musical arrangement. To complete the project, you will create audio and MIDI tracks, import audio clips, create playable instruments using the Drum Rack and Simpler instruments, record from the Session View into the Arrangement View, and export an audio file from the Live Set

Duration

This project should take approximately 90 minutes to complete.

Goals/Targets

- Customize the appearance of your Live Set and your tracks
- Work with the Drum Rack and Simpler devices to create MIDI-based drum and bass parts
- Import and work with audio files
- Record from the Session View into the Arrangement View
- Add a Limiter to enhance your mix
- Export your final mix and archive your finished project

The media files for this project are provided courtesy of the artists Haak and Siddhant Bhosle:

- **Song**: The Fall (excerpt)
- **Performed by**: Haak and Siddhant Bhosle
- **Written By**: Haak and Siddhant Bhosle
- **Produced By**: Haak and Siddhant Bhosle ©2018



The audio files provided for this project are strictly for use to complete the exercises contained herein. No rights are granted to use the files or any portion thereof in any commercial or non-commercial production or performance.

Powering Up

To get started on this project, you will need to power up your system. It is important to power up properly to avoid problems that could possibly damage your equipment.

When using audio equipment, you should power up components in the order that the audio signal will flow through them.

The general process for starting up an Ableton Live Set is as follows:

1. Power up external hard drives, if used.
2. Verify connections and power up any audio and/or MIDI interfaces.
3. Start the computer.
4. Launch Ableton Live.
5. Power up your monitoring system, if applicable.

Refer to Chapter 2 for more details on powering up your system.

Getting Started

You will start by opening the Final Project Set file. The Set file is located in the Final Project Media folder within the Media Files 2019-Live101 folder.

Accessing the Media Files for this Project

To complete this project, you will use various files included in the Media Files 2019-Live101 folder. You can download the media files by going to www.halleonard.com/mylibrary and entering your access code (printed on the opening page of this book). From there, click the Download link for the Media Files 2019-Live101 listing in your My Library page.

The Media Files folder will be downloaded and placed in your Downloads folder by default.

Open the Final Project:

1. Do one of the following:
   - Choose **File > Open Live Set**.
   - Press **Command+O** (Mac) or **Ctrl+O** (Windows).
2. Locate and open the Final Project file (Storage Drive/Folder > Media Files 2019-Live101 > 03 Final Project Media > FinalProject-Starter.als).
3. Choose **File > Save Live Set As** and name the Set *FinalProject-XXX*, where *XXX* is your initials. Save the Set into your Live101-XXX folder or other location of your choosing.

Refer to Lesson 4 for additional information on locating and opening Live

Sets.

Adjust the Live Set Display

When the Live Set opens, you will see the starter template in the Session View. This is where you will start, but eventually you will record your work from the Session View into the Arrangement View. You can switch between the two views at any time by pressing **TAB**.



Figure 11.1 The starter template, as seen in the Session View

You may want to adjust the Session View track widths before continuing.

Change track width in the Session View:

1. Hover your mouse cursor over the right-hand edge of a track's title (for instance, Drums or Bass).

2. When the pointer turns into a bracket, click and drag left and right to contract or expand a track, respectively. (See Figure 11.2.)

3. Adjust the track widths so each track fits comfortably on your screen.

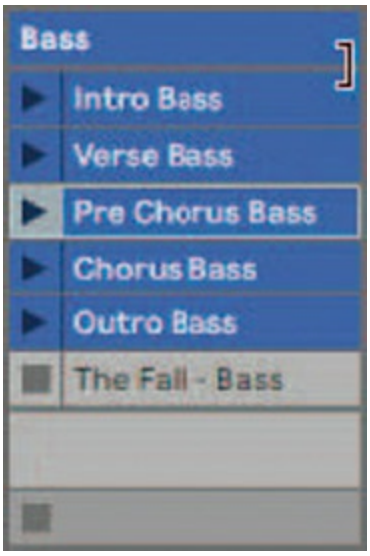


Figure 11.2 Adjusting the width of the Bass track in Session View

You may also want to expand or contract the Mixer section of your tracks. This is a vertical adjustment that can reveal additional Mixer features, as well as add resolution to each track's volume meter.

Expand or contract the Mixer section in the Session View:

1. Hover your mouse cursor over the horizontal divider that separates the In/Out section from the Mixer section.

2. Click and drag vertically to expand or contract the Mixer section, respectively.



Figure 11.3 Adjusting the Mixer height in Session View

Connect Monitoring Devices

If you have a monitoring system connected to the left and right outputs of your audio interface, you will use that to listen to the Live Set playback. If you do not have a monitoring system, you can listen to the playback using headphones on a compatible interface or on your computer (if using onboard sound output). Plug in your headphones to an available headphone jack and test the output and playback level.

Creating New Tracks

In this section of the project, you will create the new tracks needed for the Live Set. Additional details on the commands and processes used in this section can be found in Chapter 4.

Create, Name, and Color Tracks

You will need to create two new tracks for the Live Set. Both tracks will play back MIDI information using virtual instruments, so both will be MIDI tracks.

The first track you will create will be a Snare Drum track to accompany the

existing drums and percussion. The second will be a Bass Instrument track, which will be played during the Intro and Verse scenes.

Create new MIDI tracks:

1. Select the Drums track by clicking on its name.

2. Select **Create > Insert MIDI Track** or use the keyboard shortcut **Command+Shift+T** (Mac) or **Ctrl+Shift+T** (Windows). This will insert a MIDI track to the right of the currently selected track.

3. Select the Bass track by clicking on its name.

4. Press **Command+Shift+T** (Mac) or **Ctrl+Shift+T** (Windows) to insert a second MIDI track, placed to the right of the currently selected track.

Live will give the tracks you inserted generic names and colors. This is a good time to customize both qualities of the tracks you have created to help keep your Live Set organized.

Name the new MIDI tracks:

1. Do one of the following to rename the tracks:

   - Right-click the name of the first MIDI track you created and select **Rename**. (See Figure 11.4.)

   - Select the track and press **Command+R** (Mac) or **Ctrl+R** (Windows).

2. Type Snare Drum as the new track name.

3. Press **Enter** to commit the change.

4. Repeat this process to rename the second MIDI track you created. Name this track Bass Instrument.

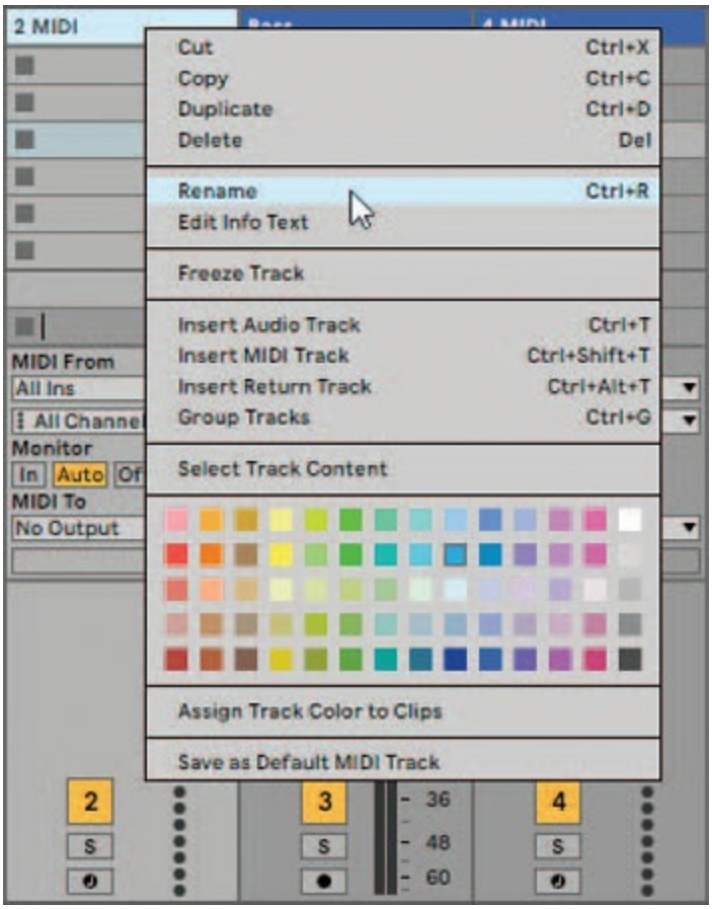


Figure 11.4 Right-click menu for renaming and color-coding tracks in Session View

Color-code the new MIDI tracks:

- Right-click each track and select a track color of your choice from the pop-up menu.

Save Your Live Set

After making any significant changes to a Live Set, it's a good idea to save your work. Doing so will minimize any rework that you might have to do in the event of a disruption.

Save your progress up to this point by doing one of the following:

- Choose **File > Save Live Set**.
- Press **Command+S** (Mac) or **Ctrl+S** (Windows).

Working with MIDI Data

For this section of the project, you will add the Drum Rack and Simpler instruments to your MIDI tracks, add audio samples to each instrument to make them playable, and program or record MIDI data using Live's MIDI Note Editor.

Add the Drum Rack to the Snare Drum Track

In the previous section, you created a MIDI track and named it Snare Drum. Now, you will add a Drum Rack instrument to this track. This will allow the MIDI track to play back audio drum samples that are triggered by MIDI data.

Insert Drum Rack on the Snare Drum track:

1. Show the Browser (if it isn't already visible) by selecting **VIEW > SHOW BROWSER** or by pressing **OPTION+COMMAND+B** (Mac) or **CTRL+ALT+B** (Windows).

2. Select the **INSTRUMENTS** category from the Browser.

3. Select the Snare Drum track by clicking on track name.

4. Double-click the Drum Rack instrument in the Browser, or click and drag to place it on the Snare Drum track.

By default, the Drum Rack does not contain any drum samples, so you will need to add a snare drum sample to it in order for MIDI information to trigger sound. This project includes the original snare drum samples used in the song.

Add a snare drum sample to the Drum Rack:

1. In the Browser, select **CURRENT PROJECT**.

2. Click the disclosure triangle to expand the Instrument Samples folder.

3. Click the snare drum samples (Snare Drum 1.wav and Snare Drum

2.wav) to audition them.

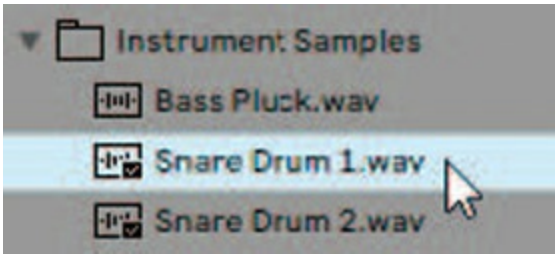


Figure 11.5 Auditioning snare drum samples in the Browser

If clicking on a sample does not trigger playback in the Browser, make sure the Preview button at the bottom of the Browser (next to the waveform display) is enabled and highlighted blue.

4. Click and drag the snare drum sample of your choice to a cell in the Drum Rack. (See Figure 11.6.) Note that each cell is triggered by a specific MIDI note, so pay attention to the note corresponding to the cell where you drop the sample.



Figure 11.6 Snare Drum 1.wav mapped to MIDI note C1 in a Drum Rack

You can also put each of the two snare samples in its own cell and trigger them separately from their respective MIDI notes.

You are now ready to play the snare drum sound of your choice using MIDI information. You can experiment with this by enabling the **ARM** button on the Snare Drum track and playing the associated note with a MIDI controller. You can also trigger MIDI from your computer keyboard by activating the Computer MIDI Keyboard switch in the Control Bar.

See Chapter 6 for details on using the Computer MIDI Keyboard function.

Add Simpler to the Bass Instrument Track

While you're in the mindset of loading audio samples into MIDI-based instruments, it is a good time to insert the Simpler instrument on the Bass Instrument track and find an audio sample to use for it.

Insert Simpler on the Bass Instrument track:

1. Show the Browser (if not already visible) by selecting **VIEW** > **SHOW BROWSER** or by pressing **OPTION+COMMAND+B** (Mac) or **CTRL+ALT+B** (Windows).

2. Select the **INSTRUMENTS** category from the Browser.

3. Select the Bass Instrument track by clicking on its name.

4. Double-click the **SIMPLER** instrument in the Browser, or click on it and drag it to the Bass Instrument track.

By default, Simpler does not contain any samples, so you will need to add a sample to the device in order for MIDI information to trigger a sound. This project includes the original synthesized bass instrument sample used in the song.

Add a bass sample to Simpler:

1. In the Browser, select **CURRENT PROJECT**.

2. Click the disclosure triangle to expand the Instrument Samples folder.

3. Click Bass Pluck.wav to audition the sound.

> **This sample may be difficult to hear at when played using the built-in outputs on a computer or laptop. You may need to temporarily increase your computer's speaker volume to**

adequately hear the low-frequency audio.

4. Click and drag Bass Pluck.wav to the **DROP SAMPLE HERE** indicator within Simpler. Because the original sample is tuned to a C, you do not need to worry about the tuning of the sample.

In many other cases, you will need to determine the pitch of the original sample and adjust its transposition in the sampler as necessary.

5. Select 1-S HOT mode using the Playback Mode buttons along the left side of Simpler. Now, even very short MIDI note durations will cause the entire sample to play back, which will work best with this sort of a short, percussive bass sound.

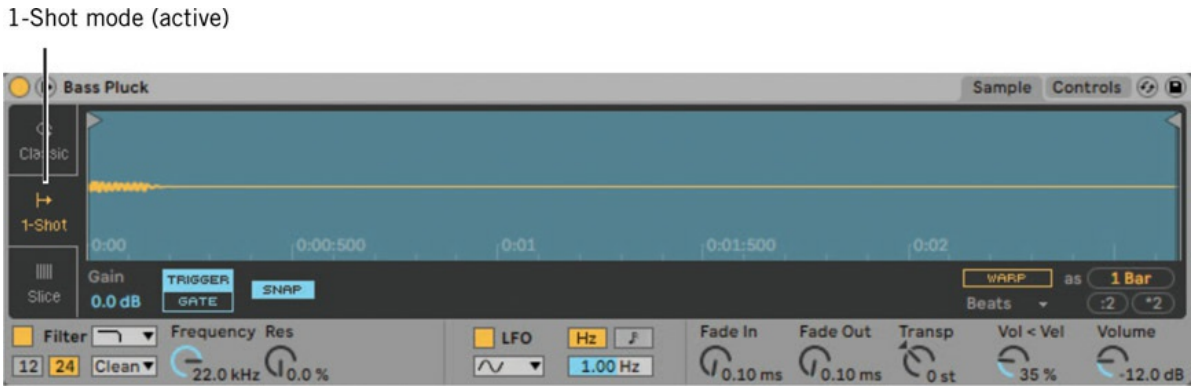


Figure 11.7 An instance of Simpler with the provided Bass Pluck.wav sample loaded in 1-Shot mode

You are now ready to play back the bass sound of your choice using MIDI information. You can experiment with this by pressing the **ARM** button on the Bass Instrument track and playing the sound with a MIDI controller or your computer keyboard with the Computer MIDI Keyboard function enabled.

See Chapter 6 for details on using the Computer MIDI Keyboard function.

Create a Snare Drum MIDI Clip

Now that your two instruments are created and capable of playing sounds, you can use them to round out this piece of music. You will start by creating a pattern of MIDI notes for the Snare Drum track.

Create a MIDI clip on the Verse scene:

1. Find the clip slot on the Snare Drum track that corresponds to the Verse scene.

2. Do one of the following:

   - Right-click the empty clip slot and select **INSERT MIDI CLIP** (S).

   - Double-click the empty clip slot.



Figure 11.8 Inserting a MIDI clip on the second scene of the Snare Drum track

3. Rename the new MIDI clip as Snare Drum.

   Your new Snare Drum clip's parameters and MIDI data should automatically become visible in the Clip View.

4. If not already displayed, double-click the Snare Drum clip to show it in the Clip View.

5. In the Notes section of the Clip View, set the clip's **LOOP LENGTH** to 8 bars (8 . 0 . 0).

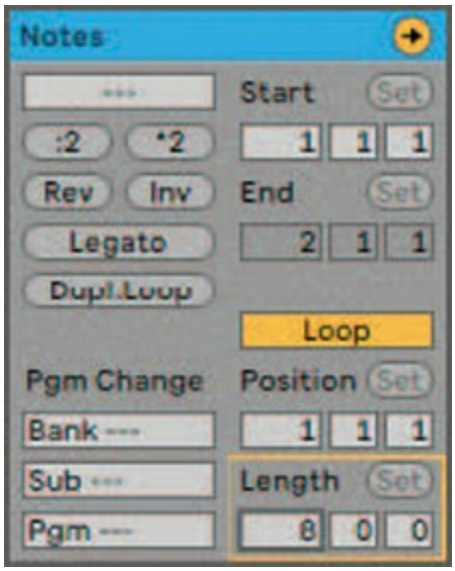


Figure 11.9 Setting the Loop Length parameter to 8 bars in the Clip View

6. Add MIDI notes using the MIDI Note Editor. Each note will cause the Drum Rack to trigger the snare drum sound. For a simple, straightforward snare drum presence, add MIDI notes at the third beat of each of the 8 bars, as seen in Figure 11.10.

There are multiple ways to manually create MIDI notes in the MIDI Note Editor. You can pencil in notes using Draw Mode (activate by pressing the B key or clicking the Draw Mode switch), or double-click at a note location in Edit Mode.

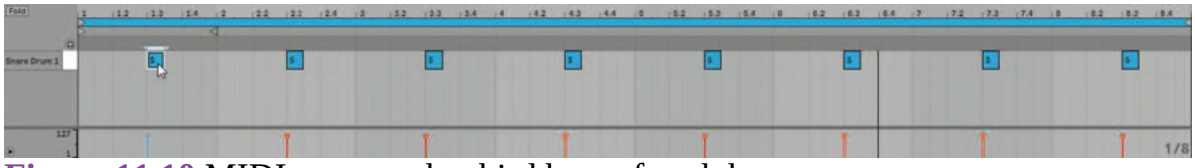


Figure 11.10 MIDI notes at the third beat of each bar

7. Press the **SCENE LAUNCH** button on the Verse scene to audition your Snare Drum track in context with the arrangement.

8. If necessary, adjust the placement and velocity of the MIDI notes to your liking.

Duplicate the Snare Drum MIDI Clip

To save some time, you can copy your snare drum MIDI clip to both the Pre-Chorus and Chorus scenes of the Live Set. That will allow your snare drum pattern to play across all of the applicable parts of the song.

Use one of the following methods to duplicate the Snare Drum clip to additional scenes:

- Right-click the clip and select **DUPLICATE** to add it to the subsequent scene.
- Right-click the clip and select **COPY**, then right-click another clip slot and select **PASTE**.
- While holding **OPTION** (Mac) or **ALT** (Windows), click and drag the clip to another clip slot.



Rather than simply duplicating the Snare Drum clip, feel free to create unique snare drum patterns for different parts of the song in this project, as time allows.

Create a Bass Instrument MIDI Clip

Next, you will create a MIDI sequence for your Bass Instrument track by importing MIDI clips that contain the song's original bass line.

Import existing MIDI clips for the bass line:

1. Show the Browser (if needed) by selecting **VIEW > SHOW BROWSER** or by pressing **OPTION+COMMAND+B** (Mac) or **CONTROL+ALT+B** (Windows).
2. In the Browser, select **CURRENT PROJECT**.
3. Click the disclosure triangle to expand the MIDI Clips folder.
4. Click and drag the Intro Bass Clip.mid file to the empty clip slot that corresponds to the Intro scene of the Bass Instrument track.
5. Click and drag the Verse Bass Clip.mid file to the empty clip slot that

corresponds to the Verse scene of the Bass Instrument track.

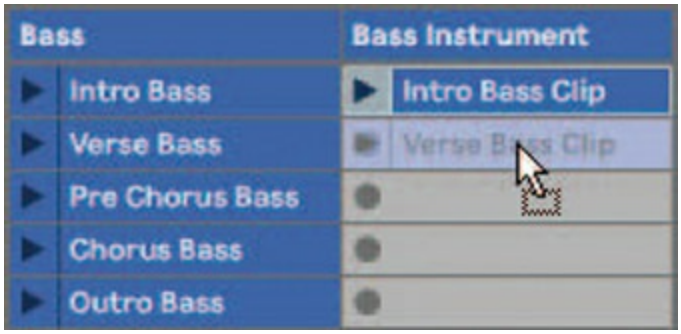


Figure 11.11 Dragging MIDI clips from the Browser to empty clip slots

Alternatively, you can program your own bass line (using the MIDI Note Editor) or record a bass performance (using a MIDI controller). Be sure to work in the key of C minor and stick to the notes of the scale: C, D, E-flat, F, G, A-flat, and B-flat.

For details on creating MIDI performances, see Chapter 6 in this book.

6. Click the **SCENE LAUNCH** button on the Intro scene to audition your Bass Instrument track in the context of the arrangement. Repeat this process on the Verse scene.

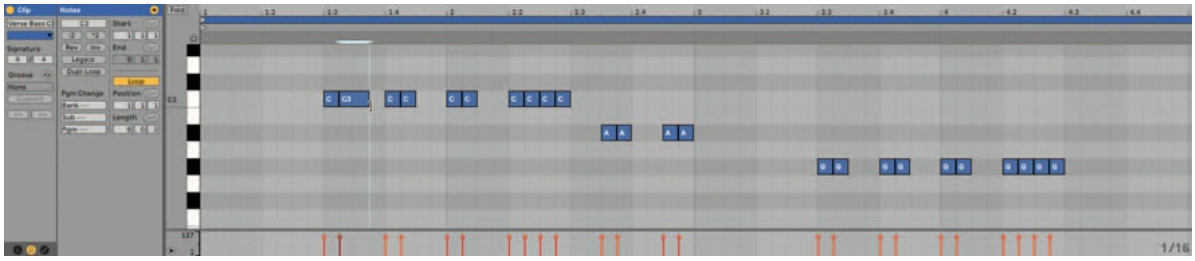


Figure 11.12 The provided MIDI notes, after being imported and displayed in the MIDI Note Editor

7. If necessary, edit the MIDI notes to your liking.

If you find that your new bass line is too quiet, turn up the Volume of Simpler. The default volume is set at −12.0dB. Try increasing that value to −6.0dB (or higher).

Save Work in Progress

As you complete each main portion of the project, you should save your work in progress. This will protect your work while the Live Set is open.

To save your progress up to this point, do one of the following:

- Choose **File > Save Live Set**.
- Press **Command+S** (Mac) or **Ctrl+S** (Windows).

Working with Audio Data

In the next section of the project, you will use the Browser to import several audio clips to an existing audio track.

Import an Audio File to a Track

As you've already seen, Live's Browser is a great feature for locating, auditioning, and importing all sorts of instruments and audio files. For this part of the project, you will use the Browser to import audio files containing the vocals of the song.

Locate and import the audio files:

1. Show the Browser as needed by pressing **Option+C ommand+B** (Mac) or **Ctrl+Alt+B** (Windows).
2. In the Browser, select **Current Project**.
3. Click the disclosure triangle to expand the folder called The Fall Stems.
4. Click the disclosure triangle to expand the enclosed Vocals folder. In this folder, you'll find the vocals, separated into clips for the verse, pre-chorus, chorus, and outro.

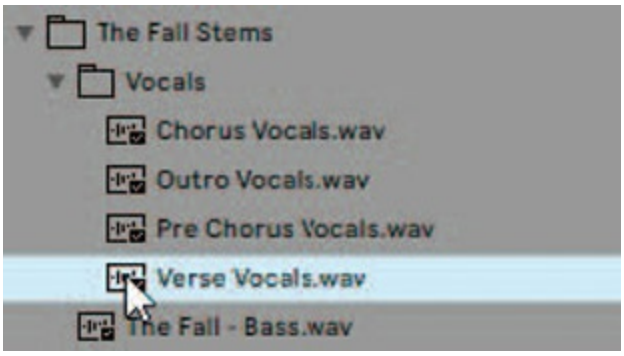


Figure 11.13 The vocals are separated into four audio clips, as seen in the Live Browser.

5. Click and drag the Verse Vocals.wav file to the empty clip slot on the Vocals track that corresponds to the Verse scene. Repeat this process for the remaining audio files: Pre-Chorus Vocals.wav, Chorus Vocals.wav, and Outro Vocals.wav.



Figure 11.14 After importing the audio, you will have clips on the Verse, Pre Chorus, Chorus, and Outro scenes.

6. Audition each scene to confirm that the vocal clips are playing back as desired.

When importing audio files, Live may automatically warp the audio clips it creates. Before continuing, you should check on the warp status and loop status of the clips and make any needed changes.

Verify warping and looping status for imported clips:

1. Click on the Verse Vocals clip in the Clip Slot on the Vocals track so that its sample information displays in the Clip View.

2. In the Sample box in Clip View, verify that the **WARP** switch is enabled.

3. Below the Warp switch, ensure the **SEG**. BPM value is set to 110 to match the tempo of the original song. If needed, you can override the current Seg. BPM value by clicking in the field and typing a new value.

4. On the right of the Sample box, verify that the **LOOP** switch is enabled.

5. Repeat the above process for the Pre-Chorus Vocals clip and the Chorus Vocals clip.

6. Next click on the Outro Vocals clip to display its sample information. Since the Outro clips occur at the end of the performance, the clips in this scene should not be set to loop.

7. Verify that the **WARP** and **SEG**. BPM controls are set as above.

8. Verify that the **LOOP** switch is disabled.

You can also experiment with disabling the Warp switch on the Outro Vocals clip. Since this clip has no rhythmic material, it may not need to match the tempo.

Save Work in Progress

You have now imported the audio files needed for the Live Set. Take this opportunity to save your work.

Save your work by doing one of the following:

- Choose **FILE > SAVE LIVE SET**.
- Press **COMMAND+S** (Mac) or **CTRL+S** (Windows).

Recording from Session to Arrangement View

While the Session View is a useful environment for live performances and sketching out song ideas, you will often want to record what you have created

in the Session View into the Arrangement View as a linear piece of music. In this section of the project, you will complete this process.

Prepare for Recording into Arrangement View

Before recording the five scenes—Intro, Verse, Pre-Chorus, Chorus, and Outro—as a linear arrangement, take a moment to play through the scenes. When each scene is about to end, press the play button on the following scene. This is the process you'll use when recording to the Arrangement View, so it will help to get some practice in advance.



When launching clips and scenes in the Session View, your timing doesn't have to be perfect. The Global Launch Quantization value ensures that clips only launch at the specified increment. This will keep your clips and scenes in rhythm with one another by aligning their launch and stop points to a global grid.



You can change the Global Launch Quantization value in the Quantization menu on the left side of the Control Bar (next to the Metronome switch).

If you want to get creative, you can make changes to the scenes or experiment with having certain scenes play for longer periods of time. For instance, you might want a 16-bar intro rather than an 8-bar intro. If you want to take things further, you can add tracks and instruments to create your own version of the song. Working in Live means it's never too late to make changes; this is a good time to experiment with additional creative choices.

Record into Arrangement View

Use the following general steps for recording from the Session View to the Arrangement View. Although the steps will largely be the same even if you've chosen to modify the Live Set, you can vary the process as needed or desired for your own composition.

Record from Session View to Arrangement View:

1. Before starting a record pass, press the **STOP** button in the Control Bar twice to place the Arrangement Position back to 1 . 1 . 1—the very beginning of bar 1. This will ensure that your recording begins at the start of the arrangement.

2. Before recording, set the **GLOBAL LAUNCH QUANTIZATION** value to 1 Bar (or another value of your choice), as seen in Figure 11.15. This will help avoid rhythmic error when recording from the Session View to the Arrangement View.



Figure 11.15 Setting the Global Launch Quantization value to 1 Bar

3. Click the **ARRANGEMENT RECORD** button (solid circle) in the Control Bar. This will turn the **ARRANGEMENT RECORD** button red and prime the Live Set for recording, but will not initiate recording just yet.

4. Click the **SCENE LAUNCH** button on scene 1, Intro. This will trigger the recording and begin playback of the Intro clips.

5. During the final measure of the Intro scene, click the **SCENE LAUNCH** button on scene 2, Verse. To gauge the timing, watch the Track Status Display indicators (see “Track Status Indicators” below).

6. During the final measure of the Verse scene, click the **SCENE LAUNCH**

button on scene 3, Pre Chorus.

7. During the final measure of the Pre Chorus scene, click the **SCENE LAUNCH** button on scene 4, Chorus.

8. During the final measure of the Chorus scene, click the **SCENE LAUNCH** button on scene 5, Outro.

9. At the end of the Outro scene, click the **STOP** button in the Control Bar or press **SPACEBAR** to conclude the recording.

Track Status Indicators

When playing in and recording from the Session View, it can be very useful to watch the Track Status Display indicators (as shown in Figure 11.16) for the currently playing clips. This will help guide you through switching scenes, as you can see how much time remains in a given clip. Looping clips are represented by pie charts, while non-looping clips display a simple countdown timer.



Figure 11.16 Track Status indicators (highlighted with yellow box)

Verify Your Results

Having completed the recording, it's time to switch to the Arrangement View and listen to your results.

Verify your recording in the Arrangement View:

1. Press **TAB** to switch from the Session View to the Arrangement View.

2. Click the red **BACK TO ARRANGEMENT** button at the top-right corner of the Arrangement display. This will cause Live to play back from the arrangement, rather than from the Session View clips.

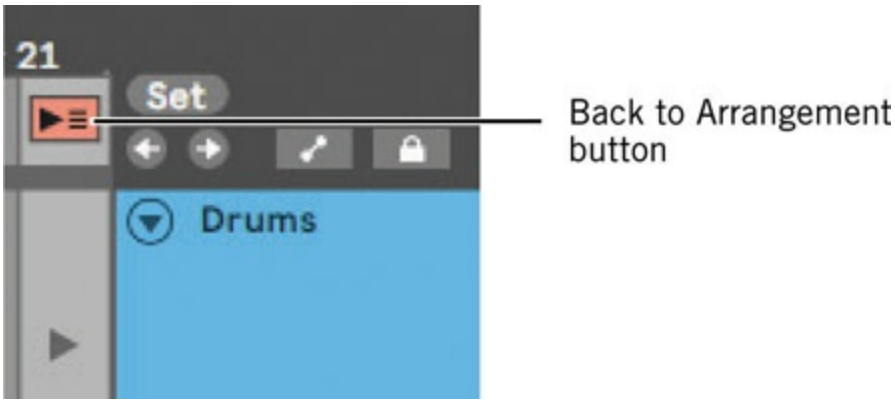


Figure 11.17 The Back to Arrangement button in the Arrangement View

3. Press the **STOP** button in the Control Bar twice (or press **HOME** on the computer keyboard) to position the cursor at the start of the Live Set.

4. Press the **PLAY** button in the Control Bar or press **SPACEBAR** to begin playback.

If you are not satisfied with your recorded arrangement, you can use **EDIT** > **UNDO** to undo the recording, or you can delete the clips. Then repeat the recording from the Session View as described above.

Clean up the Recorded Arrangement

You may want to fine-tune your arrangement once you've recorded it from the Session View. For instance, if you've extended a section of the song, you might need to smooth out the transition as the section repeats.

Here are some techniques you can use to clean up the arrangement you've created.

- With the insert marker at the start of the arrangement, choose **Create** > **Insert Silence** or press **Command+I** (Mac) or **Ctrl+I** (Windows). Insert 4 bars of silence on the front end of the arrangement.

- Delete, deactivate, or edit certain clips so that they play back only when you want them to.

- Trim the end of the arrangement of any unnecessary tails on your audio and MIDI clips.

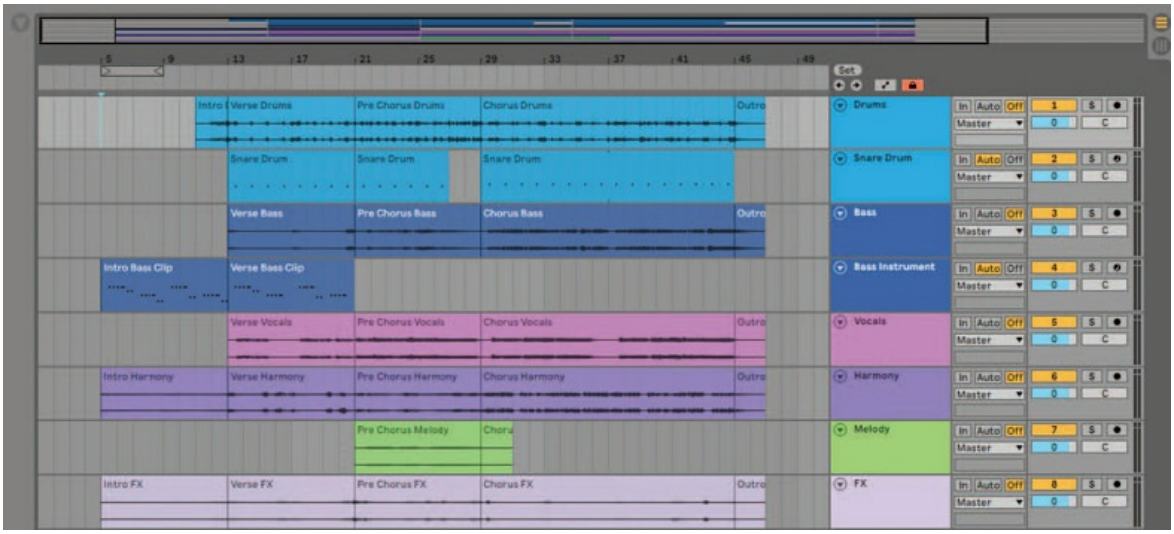


Figure 11.18 An example of an arrangement that has been edited and simplified after recording

Save Work in Progress

You have now created an arrangement for your project. Take this opportunity to save your work.

To save your work, do one of the following:

- Choose **File** > **Save Live Set**.

- Press **Command+S** (Mac) or **Ctrl+S** (Windows).

Finishing Your Work

To finish the project, you will add final processing on the Master track, export your arrangement to a stereo audio file, and archive your work. The archive process will allow you to create a backup of your work without consuming excess disk space.

Add a Limiter

First, you will add a Limiter audio effect to the Master track. A limiter is meant to maximize the overall level of a mix, while simultaneously limiting peaks to prevent clipping. Ideally, the end result is a mix that sounds louder while preserving its overall quality.



Live's Limiter device performs "look-ahead" analysis, anticipating peaks in audio material and preserving attack transients during reduction. This helps makes the results more transparent and maintains the character of the original audio signal without clipping or distortion.

Insert the Limiter effect on the Master track:

1. Show the Browser as needed by pressing **Option+Command+B** (Mac) or **Ctrl+Alt+B** (Windows).

2. Click **Audio Effects** in the Browser.

3. Select the **Master** track by clicking on its name.

4. Double-click the **Limiter** effect in the Browser, or click and drag it onto the Master track.



The Limiter device will not be available when using Ableton Live Lite.



See Exercise 10 for suggestions on using the Multiband & Limiter device as an alternative in Ableton Live Lite.

5. Set the **Ceiling** parameter in the Limiter to −0.30 dB. This will ensure

that your output will never exceed −0.30 dB and will safeguard the mix against clipping.

6. Increase the **GAIN** parameter to around 6.00 dB. During louder parts of the song, you should see some light to moderate activity on the Gain Reduction meter. This indicates the Limiter is working to prevent the signal from exceeding the −0.30 dB ceiling and protecting against clipping, while raising the overall level of the Live Set by 6.0 dB.



Figure 11.19 The Limiter audio effect on the Master track

Be careful not to increase the Gain too much, as this can introduce distortion.

Export the Song

You can now proceed to export your song as a single stereo audio file.

Export the song:

1. Select the export range by doing one of the following:

   - Make a selection spanning the entire length of the arrangement on at least one track.

   - Create and select a loop region spanning the length of the arrangement, using the loop brace, as illustrated in Figure 11.20.

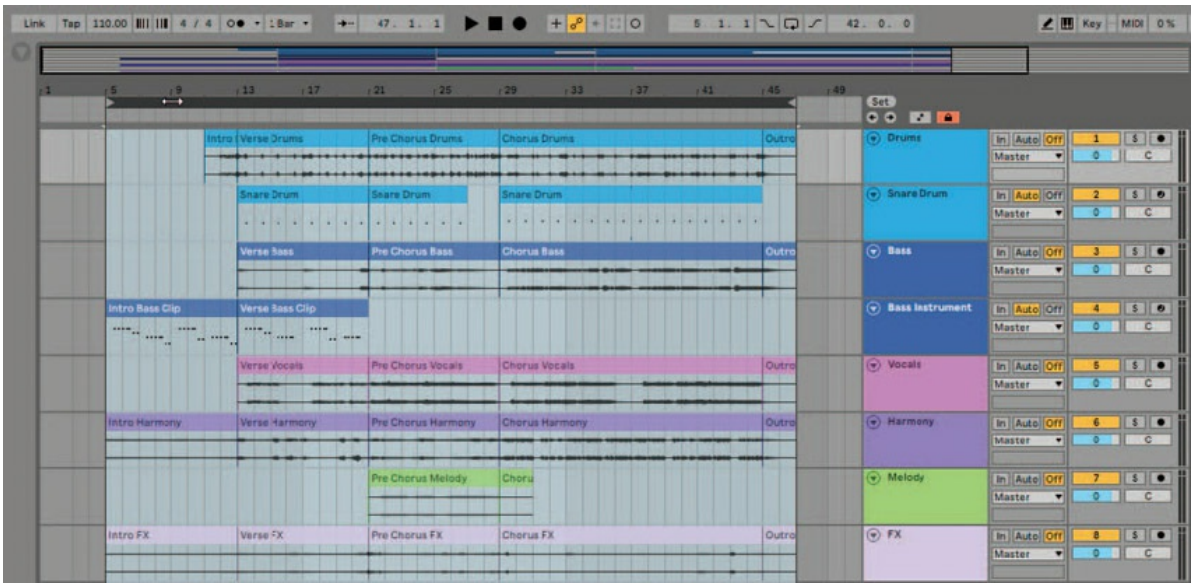


Figure 11.20 A selection of the arrangement made by selecting a loop region

2. Select **FILE** > **EXPORT** or press **COMMAND+SHIFT+R** (Mac) or **CTRL+SHIFT+R** (Windows) to open the **EXPORT AUDIO /VIDEO** dialog window.

3. Verify that the **RENDERED TRACK** is set to Master, and that the **RENDER START** and **RENDER LENGTH** values are an accurate representation of the arrangement start and length. This will help you avoid exporting silence and/or abruptly cutting off audio in the exported file.

4. Choose the following settings in the Export dialog box:

    - Sample Rate: 44100
    - File Type: WAV
    - Bit Depth: 16
    - Dither Options: POW-r 1

As a rule of thumb, you should always add dither when bouncing to a lower bit depth.

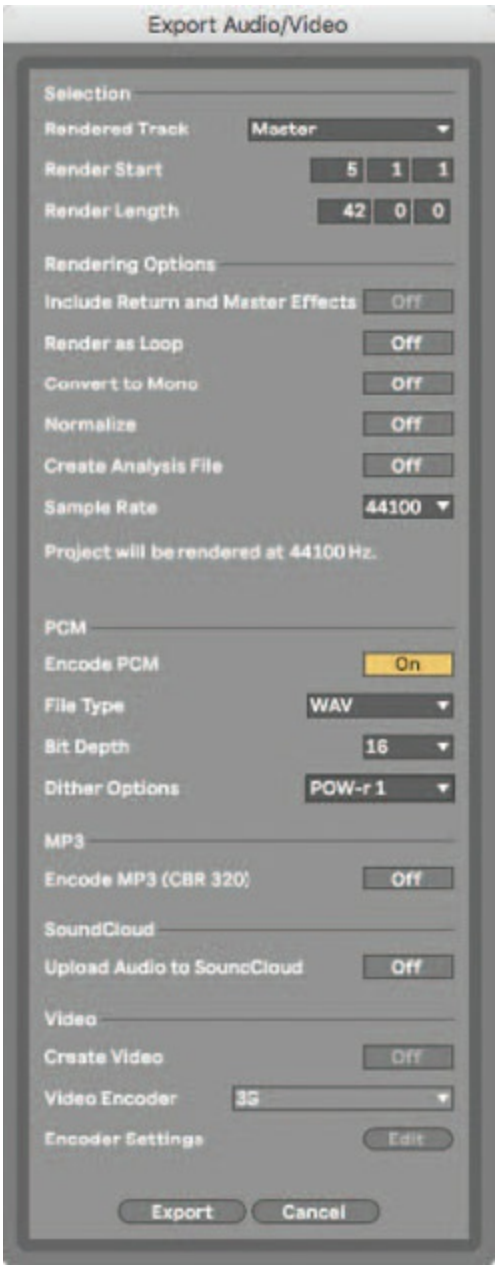


Figure 11.21 The Export Audio/Video dialog window

5. Click the **EXPORT** button.

6. Navigate to the Final Project directory where your Live Set is saved.

7. Name your file and click **SAVE**. The mixdown will be saved as a .WAV file in your project directory.

Archive Your Work

Now that your project is complete, you have the option to back it up for storage. On a real-world project, you might also need to deliver the work to a client. If you are completing this project in an academic classroom environment, you may be required to submit your work to your instructor for grading.

Because there are often many files associated with a Live Set, something could get lost if the archival process isn't completed properly. In this section, you will use the **COLLECT ALL AND SAVE** command to collect all of your Live Set's media files into the current Live Project directory. This will ensure that your archive includes all of the files you need for the Live Set.

Archive your project:

1. Choose **FILE > COLLECT ALL AND SAVE**.

2. When prompted to specify the files that will be included in the project, choose the following settings:

   - Files from elsewhere: Yes
   - Files from other Projects: Yes
   - Files from User Library: Yes
   - Files from Factory Packs: No

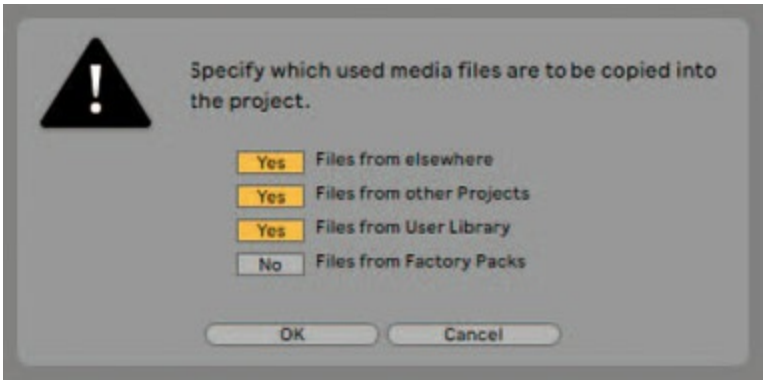


Figure 11.22 The Collect All and Save dialog box

3. Click the OK button. This will save your Live Set and copy the audio files used in the project into the current project directory.

This concludes the project. If you are com pleting this work in an academic environment, please check with your instructor for submittal requirements; be sure to include the exported audio mixdown file with your Live Set archive, as appropriate.

Appendix A

Ableton Live Audio and MIDI Effects

The following Audio and MIDI Effects devices are installed with Ableton Live 10 Suite. Ableton Live Standard and Ableton Live Intro feature a reduced set of these devices.

Audio Effects

- Amp – Features models of seven classic guitar amplifiers.
- Audio Effect Rack – Can be used to combine multiple audio effects into a single device with a shared set of Macro controls.
- Auto Filter – Models an analog multi-mode filter with an envelope follower and LFO.
- Auto Pan – Automatic panning device featuring an LFO.
- Beat Repeat – Repeats portions of drum beats to create new and interesting patterns.
- Cabinet – Features models of five classic guitar cabinets.
- Chorus – Thickens incoming audio using two parallel modulated delays.
- Color Limiter (New in Live 10) – A limiter inspired by vintage hardware limiters.
- Compressor – Workhorse dynamics processor that helps control and maximize levels.
- Convolution Reverb – Ultra-realistic reverb that uses samples of real physical spaces.

- Corpus – Physically models seven types of resonant objects.
- Dynamic Tube – Models analog tube saturation effects.
- Drum Buss (New in Live 10) – A drum-processing device that can add warmth, distortion, drive, compression, transient shaping, and low-frequency enhancement.
- Echo (New in Live 10) – Combines classic analog and digital delay into a hybrid device.
- EQ Eight – Eight multi-mode EQ bands with comprehensive control.
- EQ Three – Classic DJ-style EQ featuring three frequency bands with kill switches.
- Erosion – Can modulate an incoming signal with mono or stereo noise, or a sine wave.
- Flanger – Emulates classic tape flanging effects using time-modulated delays.
- Gate – Can be used to reduce the volume of quieter portions of a single for purposes of eliminating unwanted noises or for rhythmic or processing effects.
- Grain Delay – Uses small “grains” of signal to create delay effects with optional pitch randomization.
- External Audio Effect – Connects external hardware effects into Ableton Live.
- Filter Delay – A delay device featuring three independent delay lines with independent controls for delay time, feedback, pan, and volume.
- Frequency Shifter – Frequency shifting and ring modulation device.
- Gated Delay (New in Live 10) – A delay fed by a gated sequencer that

be set to rhythmic patterns.

- Glue Compressor – Bus compressor modeled after the classic SSL bus compressor.

- Limiter – Limits incoming signals to a specific level while maximizing output volume.

- Looper – Emulates classic pedal looping devices and adds modern features such as quantization and tempo control.

- Multiband Dynamics – Features compression and expansion in up to three separate frequency bands.

- Overdrive – Distortion effect based on classic guitar distortion pedals.

- Pedal (New in Live 10) – Features models of classic overdrive, distortion, and fuzz stompbox effects.

- Phaser – Creates a sweeping sound by modulating the frequency and feedback of a multi-pole filter.

- Ping Pong Delay – Delay that bounces from left to right and features a band-pass filter.

- Pitch Hack (New in Live 10) – A delay line with transposition controls. Additional features include reversing audio, randomizing transposition, and recycling signal back into the delay line.

- Redux – Bit-depth and sample-rate reduction for low-fi effects.

- Re-Enveloper (New in Live 10) – A multiband envelope processor that divides the signal into three adjustable frequency bands.

- Resonators – Five tunable resonators that can be applied to any input signal.

- Reverb – Synthetic reverb based on classic analog and digital hardware

devices.

- Saturator – A distortion effect featuring waveshaping to add warmth or dirt to a signal.
- Spectral Blur (New in Live 10) – Creates reverb-like sounds and textures within a set frequency range.
- Spectrum – Offers spectral analysis of audio signals.
- Surround Panner (New in Live 10) – Surround panning device for mixing with multi-channel speaker configurations.
- Simple Delay – Basic delay device with feedback and tempo sync.
- Tuner – Simple tuner for tuning real or virtual instruments.
- Utility (Updated for Live 10) – Simple yet useful device for adjusting phase, stereo width, gain, balance, and collapsing signals to mono.
- Vinyl Distortion – Models the noise, distortion, and crackle of vinyl records.
- Vocoder – Offers classic robotic voice effects processed through a bank of bandpass filters.

MIDI Effects

- Arpeggiator – Creates rhythmical patterns from one or more held MIDI notes.
- Chord – Turns MIDI notes into chords with up to six notes.
- Envelope – Can be used to modulate device parameters with incoming MIDI notes.
- Envelope Follower – Can be used to modulate device parameters with any audio signal.

- Expression Control – Can be used to modulate any parameter in your Live Set from a range of MIDI data such as velocity, modwheel, and pitchbend.

- LFO (New in Live 10) – Can be used to modulate up to eight device parameters with one device.

- Melodic Steps (New in Live 10) – **AMIDI** step sequencer designed to generate evolving patterns.

- MIDI Effect Rack – Can be used to combine multiple MIDI effects into a single device with a shared set of Macro controls.

- MIDI Monitor – Lets users monitor MIDI data including notes, velocity, and continuous controllers.

- Note Echo – Like a delay line for MIDI notes with delay and feedback control.

- Note Length – Can modify the length of MIDI notes or trigger new notes via Note Off.

- Pitch – Can transpose MIDI notes by a specified interval within a user-definable range.

- Random – Can randomly transpose MIDI notes based on chance.

- Scale – Transposes incoming MIDI notes to different outgoing notes based on user-specified scale.

- Shaper (New in Live 10) – An LFO-driven envelope control that can be used to create unique modulation changes.

- Velocity – Can randomize or control the dynamic range of MIDI note velocity.

Appendix B

Ableton Live Software Instruments

The following Software Instruments are installed with Ableton Live 10 Suite. Ableton Live Standard and Ableton Live Intro feature a reduced set of these instruments.

Software Instruments

- Analog – Physically models the sound of vintage analog synthesizers.
- Bass – A monophonic virtual analog synthesizer that emulates a range of vintage and modern sounds.
- Collision – Physically models the sound of mallet instruments.
- Drum Rack – Akai MPC-style 16-pad grid for playing drums, percussion, and other samples.
- DrumSynths – A collection of devices that use synthesis to create drum and percussion sounds.
- External Instrument – Allows you to integrate your external hardware synthesizers into Ableton Live.
- Electric – Physically models the sound of classic electric pianos.
- Impulse – A minimal sampler optimized for playing drum and percussion samples.
- Instrument Rack – Can be used to combine multiple instruments into single device with a shared set of Macro controls.
- Operator – A synthesizer that combines Yamaha-style frequency modulation (FM) synthesis with analog subtractive synthesis.

- Poli – A polyphonic virtual analog synth that emulates the classic poly-synths of the 1970s and 80s.

- Sampler – A comprehensive sampling instrument with multi-sample support.

- Simpler – An intuitive single-sound sample player with deceptively powerful sound manipulation capabilities.

- Tension – Physically models the sound of stringed instruments.

- Wavetable (New in Live 10) – A wavetable synth that combines powerful wavetable oscillators with an elegant and intuitive user interface.

About the Authors

This book and associated coursework has been developed by **Eric Kuehnl** and **Andrew Haak**, and is designed to prepare students for Ableton Live certification under the NextPoint Training Digital Media Production program.

Eric Kuehnl is a composer, sound designer, and educator. He is the Director of Partner Programs for NextPoint Training, and the Co-Director of the Music Technology Program at Foothill College. He also teaches game audio and audio post-production courses at Pyramind in San Francisco. Previously, Eric was an Audio Training Strategist in the Avid Education Department, and a Sound Designer for Sony Computer Entertainment America. He holds a Master's degree from California Institute of the Arts, a Bachelor's degree from Oberlin Conservatory, and studied composition at the Centre Iannis Xenakis in Paris.

Andrew Haak is a San Francisco-based producer, sound designer, and engineer who is passionate about the infinite possibilities of digital audio. A graduate of Pyramind's Sound for Pictures and Games program, he's currently pursuing his artist career as an electronic music producer and helping fellow musicians and producers express their creativity as a sound design instructor. Formerly a full-time communications and PR professional, Andrew took a sudden turn into the realm of music after discovering the potential for self-expression and learning that modern music DAWs enable — and hasn't looked back.

1. Because it can help you make better programming decisions when working with a singular language, or to help you bridge your understanding from one lang to another. Example: knowing how optimizers improve your code for you, so you know what good code looks like, as well as what the compiler will do for you.

2. fortran

3. cobol

4. c (and in-line asm) also Linux started adopting Rust in parts

5. You wind up with c++! Just joking... Having multiple ways to accomplish the same thing leads the programmer to debate best practice in their head, slowing them down. It also aids in readability to only have one way to do things, unifying codebases with multiple developers by force.

6. I'm assuming this is asking "what is an example of Java having multiple ways to accomplish the same thing", then interfaces and inheritance. If you have an interface with (I forget the exact syntax so I'm going to pseudocode it out)

interface dog:
ears, tail, nose

... and a class with the same features

class dog:
ears, tail, nose

then inheriting from either of these will provide the same results.

7. Ortogonality is a quasi-measure of how universal and independent lang features are. For example, an array in c can be of any type, from primitives to custom types to structs. Arrays are "orthogonal" because a ton of possibility opens up for data structures.

8. I think you could argue either way. I would say that orthogonal features are in general good and keep the language simple. In C, it makes sense to have the highly orthogonal struct as a cornerstone of the language. Some would call c

complex, but in my opinion it's one of the simplest languages -- it literally has the least number of features of most of the modern languages.

9. oh, cool, this dismisses my last answer so let's go for it.

1. pointer decay is annoying and represents a lack of orthogonality.

```
int* x()
{
   int z = 2;
   int y* = &z;
   return y;
}
```

when I dereference y, it will be an error or just a random memory address... however, in my opinion this makes perfect sense because y* points to nothing after that 2 goes out of scope. But let's just say that I don't believe that... there's my answer!

2. Structs and pointer decay

This is actually a really weak point of c. Building on my first answer:

```
Struct a
{
   struct* b structpointer;
};

struct b
{
   int x;
};
```

struct a contains one reference to a struct b. But see what happens when you try to reference b.

```
foo()
{
   struct a Alpha;
   struct b Beta;
```

```
    Alpha.structpointer = &Beta;
    return Alpha;
}

// then after you return Alpha, it's pointer decays to nothing but Alpha stays
and is passed as a copy.
// again, you're to use malloc here, but this gets pretty dicey pretty fast...
// now then, I argue that you should always store malloc'd pointers in a data
structure that you can easily loop through, calling free on each element, but
not everyone can be as cool as me I guess
```

3. I'm having a really hard time here. Void pointers are really not orthogonal. For example:

```
int main()
{
    void* x;
    int y;
    char* z = "hello world";
    char* compare = "goodbye, cruel world!";

    // make my void pointer point to my integer
    x = &y;

    // then if I go
    strcmp(compare, x);
    // I can't do that because my void pointer doesn't point to a string

    // likewise if my void pointer points to my string
    x = z;
    // then I can't do things that you can do to ints with them
    &x += 2;
}

// but again, I'd argue that void pointers are MUCH cleaner than the very
annoying type generics that you get in java. I'd argue that java's way is far
less orthogonal, but that's just my opinion I guess.
```

10. Type checking prevents what I just discussed in part 3 of the last question. You get more robust language server processing, syntax checking and

completion, etc. Without static typing, the compiler can't tell you before runtime if my strcmp(compare, x) is valid.

11. a, java

12. optimization

13. True

14. T

15. F, I think? In the class you're building overloaders for it probably leads to reduced readability, for the benefit of that class' implementation being far more readable. Don't want to be building a strcat() function in every file and such, haha. But also it can reduce readability if you have some weird class and have no idea what + or < might actually do. like:

```
class dog:
   ears, nose

foo:
   dog 1, dog 2;
   if dog 1 < dog 2:
      return true
```

How could dog 1 be less than dog 2? its weight? its height? its dollar value? how cute it is?? haha

16. It's hard to find a non-reliable programming language in the langs we typically write on a day to day basis (except for javascript, haha!). Reliability tells us how deterministic its outputs are. So, the most reliable languages have strong type systems and stability. Python programs can be extremely unpredictable on both of these fronts: they can run the same code in wildly different lengths of time, and the weird typeing can make it hard to predict why bugs occur. For example:

```
def pythonfunction():
   # if this fails, there's nothing preventing bugs from occuring
   x = someclass()
```

```
# also there's no knowing if this will work properly:
# the lsp can't predict this being a possible bug
x += 2
```

17. Exception handling is dealing with edge cases in your code. Java throws an exception when you, for example, divide by zero. So you set up a try catch block, and in the event of a divide by zero, you can have code that deals with that situation.

18. Generality is how universal its problem solving capabilities are. A programming language without the ability to perform addition would not be very general.

Portability means "write once, debug everywhere!". Just joking, portability is a measure of how reusable the same code is on multiple platforms. An example is c on ARM vs x86, with or without an OS, certain features of the language compile and run the exact same.

19. I have no idea, I think it'd be a pretty neat feature, except that most programmers can't use pointers cleanly! Though because java has method variables, pointers are less necessary. Since classes have access to their method variables and the ones they inherit... well, this basically acts like a pointer and solves the problem of having to copy memory all around the place. You get enough memory cleanliness this way.

20. To turn code into machine code.

21. To store variables and other identifiers.

Modern Acoustics and Signal Processing

Ning Xiang
Gerhard M. Sessler *Editors*

Acoustics, Information, and Communication

Memorial Volume in Honor of Manfred R. Schroeder



ASA Press



Modern Acoustics and Signal Processing



Modern Acoustics and Signal Processing

Editor-in-Chief

William M. Hartmann, East Lansing, USA

Editorial Board

Yoichi Ando, Kobe, Japan
Whitlow W. L. Au, Kane'ohe, USA
Arthur B. Baggeroer, Cambridge, USA
Neville H. Fletcher, Canberra, Australia
Christopher R. Fuller, Blacksburg, USA
William A. Kuperman, La Jolla, USA
Joanne L. Miller, Boston, USA
Manfred R. Schroeder, Göttingen, Germany
Alexandra I. Tolstoy, McLean, USA

For further volumes:
http://www.springer.com/series/3754

The ASA Press

The ASA Press imprint represents a collaboration between the Acoustical Society of America and Springer dedicated to encouraging the publication of important new books in acoustics. Published titles are intended to reflect the full range of research in acoustics. ASA Press books can include all types of books published by Springer and may appear in any appropriate Springer book series.

Editorial Board

James Cottingham (Chair), Coe College
Diana Deutsch, University of California, San Diego
Mark Hamilton, University of Texas at Austin
William Hartmann, Michigan State University
Philip Marston, Washington State University
Allan Pierce, Boston University
Arthur Popper, University of Maryland
Erica Ryherd, Georgia Tech
Andrea Simmons, Brown University
Martin Siderius, Portland State University
Ning Xiang, Rensselaer Polytechnic Institute
William Yost, Arizona State University



ASA Press

Ning Xiang • Gerhard M. Sessler
Editors

Acoustics, Information, and Communication

Memorial Volume in Honor of Manfred R. Schroeder



ASA Press

Springer

Editors
Ning Xiang
The School of Architecture
Rensselaer Polytechnic Institute
Troy, NY, USA

Gerhard M. Sessler
Darmstadt University of Technology
Darmstadt, Germany

ISBN 978-3-319-05659-3 ISBN 978-3-319-05660-9 (eBook)
DOI 10.1007/978-3-319-05660-9
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014941939

© Springer International Publishing Switzerland 2015
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Acoustical Society of America

The mission of the **Acoustical Society of America** (www.acousticalsociety.org) is to increase and diffuse the knowledge of acoustics and promote its practical applications. The ASA is recognized as the world's premier international scientific society in acoustics, and counts among its more than 7,000 members, professionals in the fields of bioacoustics, engineering, architecture, speech, music, oceanography, signal processing, sound and vibration, and noise control.

Since its first meeting in 1929, The Acoustical Society of America has enjoyed a healthy growth in membership and in stature. The present membership of approximately 7,500 includes leaders in acoustics in the United States of America and other countries. The Society has attracted members from various fields related to sound including engineering, physics, oceanography, life sciences, noise and noise control, architectural acoustics; psychological and physiological acoustics; applied acoustics; music and musical instruments; speech communication; ultrasonics, radiation, and scattering; mechanical vibrations and shock; underwater sound; aeroacoustics; macrosonics; acoustical signal processing; bioacoustics; and many more topics.

To assure adequate attention to these separate fields and to new ones that may develop, the Society establishes technical committees and technical groups charged with keeping abreast of developments and needs of the membership in their specialized fields. This diversity and the opportunity it provides for interchange of knowledge and points of view has become one of the strengths of the Society.

The Society's publishing program has historically included the *Journal of the Acoustical Society of America*, the magazine *Acoustics Today*, a newsletter, and various books authored by its members across the many topical areas of acoustics. In addition, ASA members are involved in the development of acoustical standards concerned with terminology, measurement procedures, and criteria for determining the effects of noise and vibration.

Series Preface for Modern Acoustics and Signal Processing

In the popular mind, the term "acoustics" refers to the properties of a room or other environment—the acoustics of a room are good or the acoustics are bad. But as understood in the professional acoustical societies of the world, such as the highly influential Acoustical Society of America, the concept of acoustics is much broader. Of course, it is concerned with the acoustical properties of concert halls, classrooms, offices, and factories—a topic generally known as architectural acoustics, but it is also concerned with vibrations and waves too high or too low to be audible. Acousticians employ ultrasound in probing the properties of materials, or in medicine for imaging, diagnosis, therapy, and surgery. Acoustics includes infrasound—the wind-driven motions of skyscrapers, the vibrations of the earth, and the macroscopic dynamics of the sun.

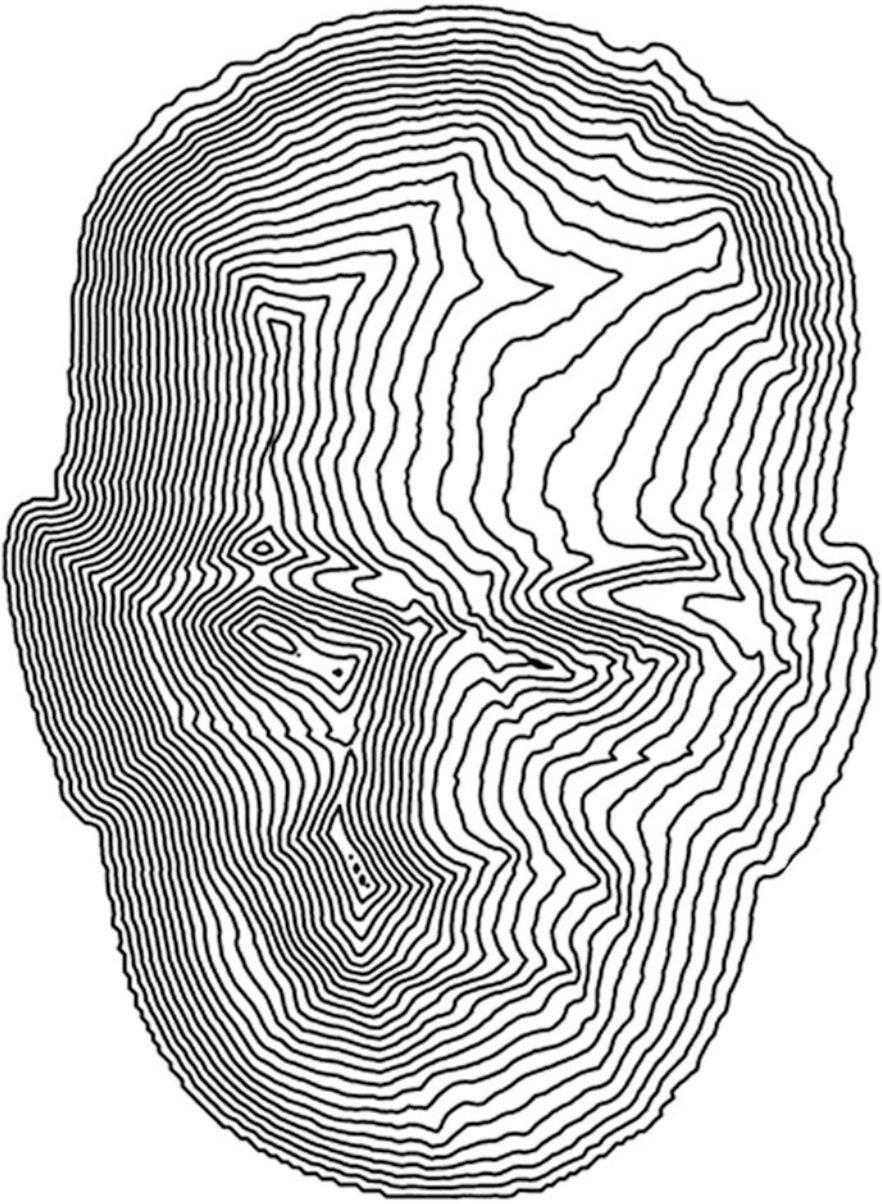
Acoustics studies the interaction of waves with structures, from the detection of submarines in the sea to the buffeting of spacecraft. The scope of acoustics ranges from the electronic recording of rock and roll and the control of noise in our environments to the inhomogeneous distribution of matter in the cosmos.

Acoustics extends to the production and reception of speech and to the songs of humans and animals. It is in music, from the generation of sounds by musical instruments to the emotional response of listeners. Along this path, acoustics encounters the complex processing in the auditory nervous system, its anatomy, genetics, and physiology—perception and behavior of living things.

Acoustics is a practical science, and modern acoustics is so tightly coupled to digital signal processing that the two fields have become inseparable. Signal processing is not only an indispensable tool for synthesis and analysis but it also informs many of our most fundamental models about how acoustical communication systems work.

Given the importance of acoustics to modern science, industry, and human welfare Springer presents this series of scientific literature, entitled Modern Acoustics and Signal Processing. This series of monographs and reference books is intended to cover all areas of today's acoustics as an interdisciplinary field. We expect that scientists, engineers, and graduate students will find the books in this series useful in their research, teaching, and studies.

July 2012 William M. Hartmann



Manfred R. Schroeder's Portrait – Solution of Eikonal equation, programed by Wolfgang Moeller (see also Chapter 20, Fig. 20.11, see also at the beginning of Part II, Fig. 28 in photo collection on page 293) (Wolfgang Moeller is one of Schroeder's former students)

Preface

Manfred Schroeder was born in Ahlen, Westphalia, Germany. In Manfred's youth, his father, a mining engineer, and his mother encouraged an early interest in the beauty and utility of mathematics. His mathematical talent was already evident in his secondary schooling. Like many science-oriented youths of the time, he was attracted to radio technology: He built short-wave receivers and transmitters. He never considered himself a legitimate "radio amateur" ("ham") because of the difficulty of obtaining a government license, and the (not unusual) additional obstacle of Morse code. But, in all other respects, he was a "ham."

Also, like many other youths of his time, Manfred's continued education was interrupted by the hostilities of World War II. At age 16, he was drafted into the German air force, serving first in Poland and later in Holland. Because of his talent for electronics, he was trained in the early uses of radar for detecting range and direction. He thereby became part of an intensive technology race, an experience that no doubt contributed to his later remarkable zeal in acoustics research.

After the conclusion of hostilities, he returned home, rejoicing that his family was intact and unharmed in this turbulent time. Manfred continued his education and entered the University of Göttingen to study physics and mathematics. In due course, as his university studies advanced, he fell under the tutelage of Professor Erwin Meyer, director of the Drittes Physikalisches Institut (Third Physics Institute) and an internationally recognized authority in acoustics. Working towards his doctoral degree, Manfred conducted fundamental research on the distribution of microwave and acoustic normal modes in enclosures, relations that characterize the frequency response and spatial spread of electromagnetic and sound energy in structures such as waveguides, rooms, and auditoria. He received his Dr. rer. nat. degree as a student of Professor Meyer in 1954. Later that year he joined Bell Laboratories in Murray Hill, New Jersey. Manfred went on to be appointed Head of the Acoustics Research Department in 1958 and Director of the Acoustics and Speech Research Laboratory in 1963. Soon thereafter, he assumed responsibility for all areas of acoustics, speech, and mechanics research at Bell Labs. In 1969, while still maintaining some of his responsibilities at Bell, he was appointed Professor of Physics and Director of the Drittes Physikalisches Institut at the

University of Göttingen. After his retirements from Bell Labs and the University in 1987 and 1991, respectively, Manfred continued his activities as a scientist and teacher.

Manfred's scientific contributions are so numerous that it would require much more space than we have in this Preface to provide a complete introduction. One field about which Manfred was always enthusiastic and to which he made many fundamental and far-reaching contributions is room acoustics. As mentioned above, this area fascinated him as early as his thesis work in the early 1950s in Göttingen, where he developed his brilliant statistical theory of frequency responses of enclosures. Manfred built vacuum-tube-based devices for electromagnetic waves to validate his hypothesis. His "radio amateur" experience and early use of radar along with the dynamic research environment at Erwin Meyer's Institute prepared Manfred to be an excellent experimentalist. In a series of publications Manfred described the relations between the eigenmodes and the frequency response of a room. He explained the random character of the response above a certain frequency (today known as the Schroeder frequency). Other ingenious contributions followed, such as the use of frequency shifting to suppress feedback in public address systems and the integrated-tone-burst method for measuring sound energy decay functions of enclosures (today known as Schroeder curves via Schroeder's integration or backwards integration). His seminal paper in 1965 on backwards integration has led to numerous research activities in room acoustics (see Chap. 3 by Xiang). Manfred introduced binary maximum-length sequences in architectural acoustics in 1979—the so-called fast M-sequence transform for room impulse response measurements (see Chap. 6 by Xiang, Xie & Cox). Another outstanding achievement is Manfred's invention of pseudo-random surface structures based on number-theoretical schemes (Schroeder diffusers, see Chap. 6 by Xiang et al. and Chap. 9 by D'Antonio). Such devices are used to scatter sound waves in all directions, and hence improve the acoustics of concert halls by providing lateral sound reflections.

Of much public interest was Manfred's involvement in the activities to improve the acoustics of Philharmonic Hall in New York City. Soon after the opening of the Hall in 1962, grave deficiencies in its acoustics were detected, and Manfred together with a few acoustic consultants was invited to analyze and remedy the situation. He was asked to make an objective evaluation of the acoustics of the Hall and decided to use a novel measuring method based on digital signal generation and processing. It was in this period that Manfred first began to develop his backwards integration method. Preparation, performance, and evaluation of these experiments turned into a scientifically challenging project for Manfred and his coworkers. Later on, while at Göttingen, Manfred and his students evaluated 20 of the major concert halls of the world by subjective auditory comparison, thereby gaining significant information about subjective parameters of room acoustics, such as lateral reflections. In addition to these new digital measuring techniques, between 1968 and 1969 Manfred pioneered a ray-tracing method of computer simulation of large halls. This was shortly followed by active research followed in room-acoustics simulations, leading to many refined methods (see Chap. 2 by Krokstad et al.). These methods have gained great practical importance for the planning of new auditoria.

Manfred Schroeder was the first to recognize the importance of digital signal processing, not only in room acoustics but in many other areas as well, and also the first to actually employ these methods. Examples are the generation of artificial reverberation (Schroeder's all-pass filter) and artificial stereophony as well as digital signal processing applications in ultrasonics, speech, and electroacoustics. Manfred is also one of the pioneers of computer graphics: For his application of concepts from mathematics and physics to the creation of artistic works he was awarded the First Prize at the 1969 International Computer Art Competition.

Another area where Manfred made fundamental contributions is speech and hearing acoustics. Among his outstanding achievements in this field are the invention of the voice-excited vocoder in 1960 and the introduction of linear predictive coding (LPC) in 1967 during the exciting days of the digital revolution at Bell Labs. LPC remains an important technique for analysis and synthesis of speech and music signals. Also in 1967 Manfred used acoustic flectrometry to determine the human vocal tract. He also worked on early development of the concept of surround sound. His mathematical model of the inner ear has inspired much of the following subsequent work in auditory research.

Manfred's interest in mathematics and especially in number theory motivated him in 1984 to write his very successful treatise "Number Theory in Science and Communication," now in its fifth edition. This book differs from other texts on the topic by its intuitive approach and by emphasizing the applications of number theory to such diverse fields as cryptography, physics, computing, and self-similarity. Many amazing relationships are uncovered in this text which is not only stimulating to read but also a lot of fun to muse about. Similarly, Manfred's 1992 book "Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise" illustrates the far-reaching influence of the concept of self-similarity on science, music, and the visual arts. Both books are marvellous examples of how to present difficult subjects transparently and in an amusing style. Manfred's third book, "Computer Speech," first published in 1999, spans an arch from speech recognition and synthesis to monaural and binaural hearing and modern signal analysis. This text summarizes Manfred's long involvement with speech and hearing research at Bell Labs. Apart from these monographs, Manfred Schroeder published more than 350 papers, many of them seminal communications, and holds 45 US patents.

As a professor and teacher at Göttingen University, Manfred was able to attract excellent students due to his inspiring lectures on such diverse topics as physics, acoustics, mathematics, and signal processing. Most of the over 45 Ph.D. candidates whom he supervised successfully continued their careers either in academia or in industry, many of them as professors, heads of companies, or chief scientists.

Manfred received numerous accolades and professional recognitions throughout his long and varied career. He was elected a member of the U.S. National Academy of Engineering and the Göttingen Academy of Science, and Fellow of the American Academy of Arts and Sciences and of the New York Academy of Sciences. As mentioned above, in 1969 he was awarded First Prize at the International Computer Art Competition for his application of concepts from mathematics and physics to the creation of artistic works. He was a founding member of the Institute de

Recherche Acoustique Musique of (IRCAM) in Paris. Manfred has also been recognized for his scientific achievements as the recipient of Gold Medals from the Audio Engineering Society in 1972 and the Acoustical Society of America in 1991. In addition, he was presented with the Rayleigh Medal from the British Institute of Acoustics in 1987, the Lower Saxonion State price in 1992, the Helmholtz-Medal of the German Acoustical Society in 1995, and the Technology prize of the Eduard Rhein-Foundation in 2004.

This introduction to Manfred Schroeder would be incomplete without a few words about Manfred's personality. Everybody who has met Manfred knows his kindness, his humor and cheerfulness, his ease of communication. Many of these attributes were demonstrated in his farewell lecture, presented at the occasion of his official "retirement," to an overflow audience in Göttingen's largest lecture hall: For 90 min, Manfred fascinated his listeners with a brilliant, lucid, and amusing demonstration of the most beautiful experiments from the fields of acoustics, optics, and chaos physics.

In December 2006, the Acoustical Society of America, at its Joint-Meeting with the Acoustical Society of Japan in Honolulu (ASA), organized an honorific session for Manfred, chaired by Ning Xiang, Juergen Schroeter, and Akiro Omoto. The ASA also organized a memorial session for Manfred, chaired by the present authors, at its Seattle meeting in May 2011. Among the invited speakers of these two Sessions were his former colleagues and students (in alphabetic order).

Joshua Atkins, Bishnu Atal, Peter Cariani (coauthor with Yoichi Ando), Peter D'Antonio, James Flanagan, Hiroya Fujisaki, Fumitada Itakura, Birger Kollmeier, Armin Kohlrausch, Roland Kruse, Heinrich Kuttruff, Max Mathews (deceased), Volker Mellert, Jean-Dominque Polack, Gerhard Sessler, Michael Vorlaender, James West, and Ning Xiang.

The first part of this book contains 13 chapters, written by colleagues and former students of Manfred, on a variety of topics. These range from architectural acoustics (chapters by James West and Joshua Atkins, Ning Xiang, Guillaume Defrance & Jean-Dominique Polack, Asbjørn Krokstad, Peter Svenson & Svein Strøm, Jean-Dominique Polack, Peter D'Antonio), psychological acoustics (Yoichi Ando & Peter Cariani, Armin Kohlrausch & Steven van de Par), and speech (Bishnu Atal) audio, acoustics measurements to electro acoustics (Roland Kruse & Volker Mellert, Gerhard Sessler, and Ning Xiang, Bosun Xie & Trevor J. Cox) and underwater acoustics (Dieter Guicking). The style of the chapters ranges from truly scientific to absolutely colloquial. Many chapters refer not only to Manfred's original work but also to new studies that originated from Manfred's studies. In all these papers, the relationship of the problems discussed to Manfred's own fields of interest is, in general, evident.

The second part of the book consists of memoirs which Manfred wrote over the last decade of his life. We are grateful to Manfred's wife Anny Schroeder and to Manfred's children Marion, Julian, and Alexander Schroeder for granting us permission to include the memoirs in this book. These recollections shed light on many aspects not only of Manfred's life but also on that of many of his colleagues, friends, and contemporaries. They also portray political, social, and scientific

events during a time span that extends from the pre-war period to the present. These memoirs, written in an inimitable and witty style, are full of information, entertaining, and fun to read. We trust that the reader will find this volume useful and stimulating.

Troy, NY — Ning Xiang
Darmstadt, Germany — Gerhard M. Sessler

Contents

Part I Scientific Chapters

1 Spatial Audio and Room Acoustics 3
Joshua Atkins and James E. West

2 The Early History of Ray Tracing in Acoustics 15
Asbjørn Krokstad, U. Peter Svensson, and Svein Strøm

3 Advanced Room-Acoustics Decay Analysis 33
Ning Xiang

4 Estimating the Crossover Time Within Room Impulse Responses . . . 57
Guillaume Defrance and Jean-Dominique Polack

5 Are Impulse Responses Gaussian Noises? 77
Jean-Dominique Polack

6 Recent Applications of Number-Theoretic Sequences in Audio and Acoustics 93
Ning Xiang, Bosun Xie, and Trevor J. Cox

7 Where Mathematics and Hearing Science Meet: Low Peak Factor Signals and Their Role in Hearing Research 113
Armin Kohlrausch and Steven van de Par

8 Neurally Based Acoustic and Visual Design 145
Yoichi Ando and Peter Cariani

9 Manfred R. Schroeder: A Personal Memoir, Optimizing the Reflection Phase Grating 177
Peter D'Antonio

10 Manfred R. Schroeder: Challenge the Present and There Is a Better Way 199
Bishnu S. Atal

11 Transducer Research at Bell Laboratories Under Manfred Schroeder 213
Gerhard M. Sessler

12 Manfred Schroeder and Acoustical Impedance 229
Roland Kruse and Volker Mellert

13 Research on Underwater Acoustics in Göttingen 241
Dieter Guicking

Part II Manfred Schroeder's Memoirs

14 Right Turns 299
Manfred R. Schroeder

15 School Years 303
Manfred R. Schroeder

16 July 1943 325
Manfred R. Schroeder

17 Radar 329
Manfred R. Schroeder

18 Lucky Breaks 337
Manfred R. Schroeder

19 The University of Göttingen 343
Manfred R. Schroeder

20 Bell Laboratories 363
Manfred R. Schroeder

21 The KGB 401
Manfred R. Schroeder

22 Professor of Physics 407
Manfred R. Schroeder

23 The German Language 423
Manfred R. Schroeder

24 Retirement and Beyond 431
Manfred R. Schroeder

25 The Year 2006: One of Our Best (So Far) 437
Manfred R. Schroeder

Bibliography of Manfred R. Schroeder's Publications 441

Index 457

List of Contributors

Joshua Atkins Beats Electronics, LLC, Santa Monica, CA, USA

Yoichi Ando Graduate School of Science and Technology, Kobe University, Kobe, Japan

Bishnu Atal Department of Electrical Engineering, University of Washington, Seattle, WA, USA

Peter Cariani Harvard Medical School, Boston, MA, USA

Boston University, Boston, MA, USA

Trevor J. Cox Acoustics Research Centre, University of Salford, Salford, Greater Manchester, UK

Peter D'Antonio RPG Diffusor Systems, Inc., Upper Marlboro, MD, USA

Guillaume Defrance Acoustics and Lighting Dpt., CSTB, Saint-Martin-d'Heres, France

Dieter Guicking Drittes Physikalisches Institut der Universität Göttingen, Göttingen, Germany

Armin Kohlrausch Smart Sensing & Analysis Group, Philips Research Europe, Eindhoven, The Netherlands

Human-Technology Interaction, Eindhoven University of Technology, Eindhoven, The Netherlands

Asbjørn Krokstad Department of Electronics and Telecommunications, Norwegian University of Science and Technology, Trondheim, Norway

Roland Kruse Acoustics Group, Institute of Physics, Carl von Ossietzky Universität, Oldenburg, Germany

Volker Mellert Acoustics Group, Institute of Physics, Carl von Ossietzky Universität, Oldenburg, Germany

Steven van de Par Acoustics Group, Institute of Physics, Carl von Ossietzky Universität, Oldenburg, Germany

Jean-Dominique Polack Université Pierre et Marie Curie—Institut Jean Le Rond d'Alembert, M° Dupleix, France

Gerhard Sessler Department of Communications Technology, Technische Universität Darmstadt, Darmstadt, Germany

Svein Strøm COWI A/S, Trondheim, Norway

U. Peter Svensson Department of Electronics and Telecommunications, Norwegian University of Science and Technology, Trondheim, Norway

James E. West Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, USA

Ning Xiang Graduate Program in Architectural Acoustics, School of Architecture, Rensselaer Polytechnic Institute, Troy, NY, USA

Bosun Xie Physics, South China University of Technology, Guangzhou, China

Part I
Scientific Chapters

Chapter 1
Spatial Audio and Room Acoustics

Joshua Atkins and James E. West

1.1 Introduction

Manfred Schroeder's pioneering work in spatial hearing, speech coding, and room acoustics during his career at Bell Labs and the University of Göttingen paved the way for many teleconferencing advances that are still being researched today. Some of the current challenges, such as multichannel acoustic echo cancellation, spatial audio capture and reconstruction, noise reduction, and coding and compression, were originally studied by Dr. Schroeder. In this chapter we will discuss his contributions to the field and how current research continues to explore methods for building immersive communication systems.

1.2 Room Acoustics

Manfred Schroeder's interest in the behavior of sound in enclosures began with his Ph.D. thesis at the University of Göttingen under Professor Meyer and continued at Bell Labs. Work at Bell Labs was done mainly to support methods to improve communication between groups of people connected through the telecommunications network, i.e. teleconferencing. In a quote from the IEEE's oral history [1], he states:

> The third one was room acoustics, my old field from my Ph.D. thesis. That was reactivated in connection with work on Philharmonic Hall in 1963 at Lincoln Center for the Performing Arts in New York City. The acoustics there were severely criticized, and the management of Lincoln Center turned to AT&T. AT&T turned to Bell Labs, and then Bill Baker asked me to join a committee of four experts to look at the acoustics situation at Philharmonic Hall. The other three were consultants, but I, by the consent decree between Western

J. Atkins (✉)
Beats Electronics, LLC, 8600 Hayden Place Culver City, CA 90232, USA
e-mail: josh.atkins@beatsbydre.com

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_1,
© Springer International Publishing Switzerland 2015



Fig. 1.1 Manfred Schroeder (standing) and James West in Bell Labs anechoic chamber at Murray Hill, New Jersey, measuring the impulse response of two walls using a starter pistol

Electric and the government, was not allowed to do consulting. But I was permitted to make measurements and assess the situation.

Figure 1.1 shows Manfred Schroeder in the Philharmonic Hall at Lincoln Center (later renamed Avery Fisher Hall). At the time, tools for measuring properties of enclosures were mainly analog, and consequently inaccurate with respect to the properties that define clarity and good articulation of voices and instruments. Initial efforts using existing equipment and techniques were performed in the Bell Labs anechoic chamber. Walls were constructed to measure reflections and to determine the impulse response from just two perpendicular surfaces using a starter pistol as a sound source (shown in Fig. 1.2). This method, even with just two walls, presented many problems; poor signal-to-noise over a broad frequency range and variations in spectrum and energy from shot to shot were among the worst. So, we knew this would be a serious problem in larger enclosures. References to balloons, clappers, and other devices for generating an acoustical impulse were all lacking in some way, but the lack of energy presented the major problem. To solve the problem, we borrowed a small canon from Rutgers University that was used to celebrate home team football touchdowns. Outdoor measurements of the impulse from the canon were encouraging in energy, length of the burst, and repeatability. With the canon placed on the stage in Philharmonic Hall and microphones distributed throughout the audience, we fired the canon. The loud burst of energy sent sound waves throughout the hall that were picked up by all microphones. The loud sound caught the interest of the hall's manager who entered immediately and panicked because visibility was poor due to the smoke from the blast. The manager forbid any more shots and immediately began to try to remove the smoke and smell from the canon before the evening performance (Fig. 1.3).

So began the use of digital signal processing to more accurately determine the acoustical properties of enclosures [2, 3]. First, the proper stimulus was created in the form of a tone burst composed digitally and played back in the enclosure over a nearly omnidirectional loudspeaker. The response of the enclosure was picked up



Fig. 1.2 Manfred Schroeder on stage at Philharmonic Hall in 1963 at Lincoln Center

by strategically placed microphones and recorded for evaluation using a computer. Such measurements were repeated for different loudspeaker and microphone positions. This new method provided a much more accurate value for the reverberation time, derived from the integrated tone-burst decay. The digital method also allowed refined measurements that resolved the seat attenuation effect which explained why certain frequency bands were missing as sound propagated over the regularly spaced seats [4]. Hall raking, adding a gradual incline in seat height from the stage, can reduce this "seat effect."

The early reverberation decay, the first 50 ms, could be defined, as well as reverberation decay in other time domains. The digital method allowed measurement of other room properties that contributed to better understanding of how sound propagates and how it effects the impression of listeners [5]. One New York Times music critic found that the subjective impression in the hall differed greatly; the best seats were said to be in the upper balcony to the right of the stage. Standard analog measurements showed little difference between locations, but the digital

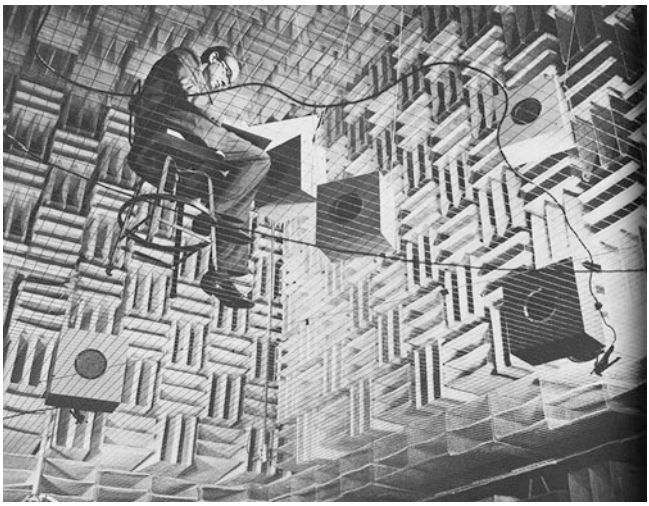


Fig. 1.3 Manfred Schroeder in the Bell Labs anechoic chamber surrounded by loudspeakers placed to represent specific echo response measured in Philharmonic Hall (now Avery Fisher Hall). The research group wanted to independently test the accuracy of the measurements by playing back the results obtained in Philharmonic Hall to see if the same subjective impression could be obtained just using early reflections

measurements of the impulse response showed many early reflections that supported better acoustics in the upper balcony.

Continuing along this line of work, a method for calculation of the sound energy decay from a room impulse response was developed, now called the Schroeder backward integration method [6]. This allowed for accurate estimation of the reverberation time. A method, based on number theory, was also created to measure the impulse response of a room without the need of an impulsive signal [7]. Based on maximum length sequences (MLS), the method relied on correlation of the input signal with the recorded signal and resulted in high SNR and the ability to take measurements of a concert hall during a performance to better understand the effect of occupied seats.

Today, room and concert hall measurements continue to be developed with similar techniques. It has been found that the MLS technique suffers from issues when the measured system has nonlinear behavior and a modified, correlation-based method based on a sine sweep is now widely used [8]. New microphone array systems are also enabling high resolution capture of the acoustic sound field as it propagates through a space [9–11]. Spherical arrays, treated as "acoustical cameras," can plot the sound energy decay in a room as a function of angle of incidence, and are leading to new methods for analysis of the complex sound field in large enclosures [12].

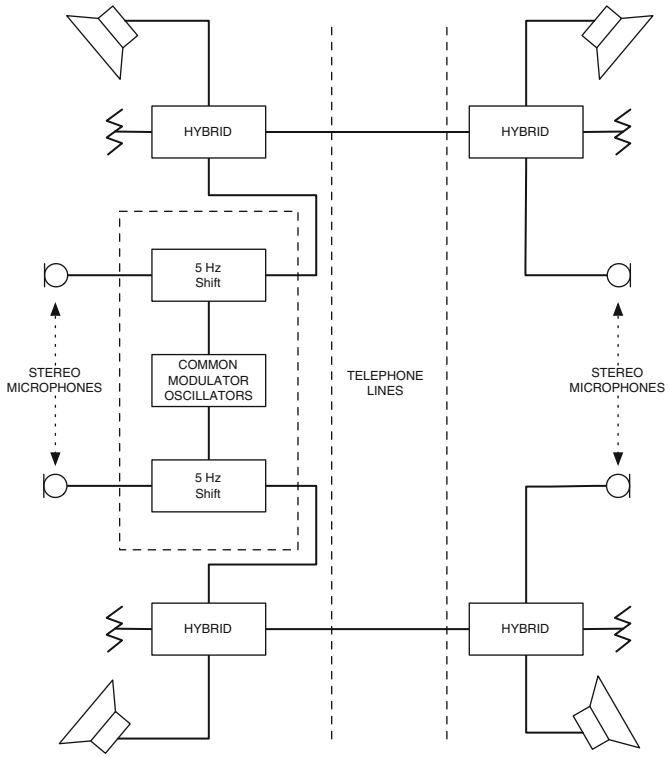


Fig. 1.4 The stereo teleconferencing system from [13]. The system shows a 5 Hz time-varying analog pitch shift that was used to minimize feedback on the far-end

1.3 Hands-Free Telephony

In the early 1960s, Bell Labs began pursuing research on conference telephony and speakerphones, which continues to be an active area of study even 50 years later. Early experiments in teleconferencing revealed the serious problem that half-duplex systems imposed on users; one-way communication simply did not work well, so efforts to reduce the echo return path began [13]. A solution was proposed using a simple frequency shifting circuit that altered the pitch of transmitted speech by 5 Hz to make the feedback path nonlinear [14]. Figure 1.4 shows a schematic of the system. This shift was nearly imperceptible on speech, but allowed for an additional 3 dB of gain in the system without "singing" artifacts.

It was later found that the single-channel acoustic echo cancellation problem can be solved efficiently using a digital system with an adaptive filter to estimate the room impulse response. However, as hands-free teleconferencing systems developed in the 1990s to include stereo and multichannel audio, a problem of correlation between the loudspeaker channels became apparent [15]. Here,

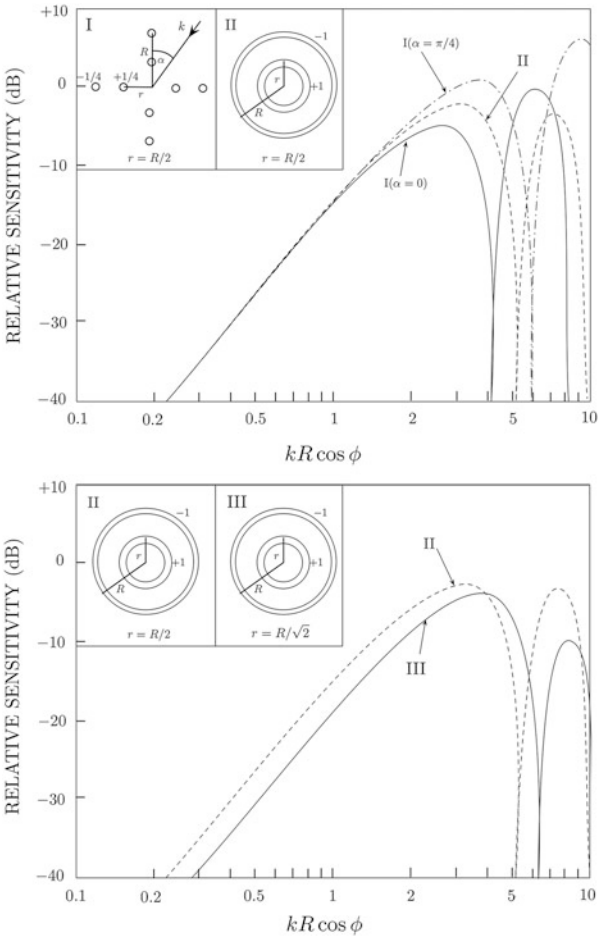


Fig. 1.5 Sensitivities of three toroidal microphones (point, line, and area sensors) from [18]. The specific microphone patterns were used to attenuate sound from a loudspeaker mounted above the microphone (in a null of the response pattern) while still capturing the voices of the teleconference participants

Schroeder's work on the problem became relevant again; in order to decorrelate the two signals, the method of adding a small frequency shift (with some slow time variation) to the speakers in a multichannel setup allows for the adaptive filters in the system to converge quickly [16]. Other methods of decorrelation, through nonlinearities or transform domain methods, have also been helpful in solving the problem [17].

The tools developed for concert hall measurements were also applied to problems incurred in small rooms for teleconferencing with considerable success. This leads to the need for microphones with special directional characteristics. The electret microphone opened the possibility to obtain small, inexpensive transducers with similar frequency and phase characteristics. This allows sensor arrays to form directional patterns to fit specific needs for reducing pickup of unwanted sounds such as noise.

Early work in this direction by Schroeder and others at Bell Labs led to the development of a differential microphone array with a constant toroidal pattern [18]. This allowed for the selective capture of speech from conference participants

seated around an array and rejection of the audio from a loudspeaker mounted above the table. Figure 1.5 shows the measured sensitivities of three toroidal microphones constructed at Bell Labs. Further research at Bell Labs led to the development of other fixed directivity microphones with different polar patterns [19, 20].

At the time there was no possibility of digitally adjusting the multiple microphone signals, but today similar microphone array systems have evolved to be adaptable in real time to track multiple moving talkers and noise sources [21]. Through the use of beamforming, modern microphone arrays can remove noise without degrading speech quality, a problem in single-channel, statistical techniques.

Taking the differential array concept to its full extent, modern cylindrical and spherical microphone arrays are now under development [9]. Spherical arrays, such as the Eigenmike, can digitally control the microphone gains to create any beam shape (toroid, cardioid, hypercardioid, etc.) and steer the pattern in any direction [9]. Current implementations suffer from SNR issues at low frequencies due to mismatch between the array radius and wavelength, and aliasing issues at high frequencies due to microphone spacing larger than the wavelength [9, 22, 23].

1.4 Spatial Audio and Perception

Manfred Schroeder's interest in spatial audio and binaural perception began shortly after he joined Bell Labs in 1954. His work started with a series of papers examining a pseudo-stereophonic effect that could increase the spatial presence of a monaural signal by using a pair of comb filters [24]. This research later led to the development of a "colorless" artificial reverberation method still widely used in digital reverberation systems today [25].

Manfred's work on binaural perception continued with a research study on the subjective effects of transmission errors in two-channel stereo [26]. Shortly after this, Manfred and Bishnu Atal developed the crosstalk canceler, first described in a patent filed in 1962 [27]. A crosstalk canceler, shown in Fig. 1.6, is used to present binaural recordings (e.g., from a manikin head) for auralization over loudspeakers. Its aim is to transmit two independent signals from a set of loudspeakers to the two ears by applying filters that compensate for the acoustic crosstalk that happens when one loudspeaker's output is heard by the ear on the opposite side [28]. The crosstalk canceler was used in research systems through the 1970s. The most notable being Schroeder's comparative study of European concert halls at the University of Göttingen, where subjective preferences of many halls were correlated with objective measurements [29]. The result of this work showed that concert halls should be designed to provide dissimilar signals to each ear at every seat, necessitating diffusive surfaces on the ceiling in modern, wide halls.

Work on crosstalk cancellation systems continued at other research institutions with improvements for playback in non-anechoic spaces and extensions to systems with more than two loudspeakers [30–33]. Today, many home sound systems have this type of processing as a standard feature. While, binaural methods focus on presenting spatial sound for a single listener with a fixed head location, techniques

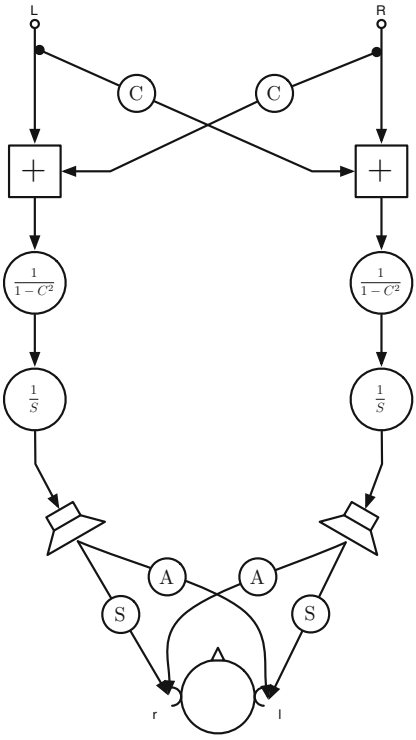


Fig. 1.6 The crosstalk canceler from [27]. The transfer functions A and S represent the ipsilateral and contralateral responses, respectively, and C represents the cancellation transfer function

for spatial sound reproduction for a group of listeners are also being developed under varying methodologies—Wave Field Synthesis [34], Ambisonics [35], and Vector Base Amplitude Panning [36]. Modern research in this area combines solutions to the wave equation in multiple dimensions with methods for solving inverse scattering problems. In many idealized cases the loudspeaker driving functions for an array can be solved analytically using simplifications to the Kirchhoff–Helmholtz integral [37] or a spatial harmonic solution to the wave equation [38]. However, whether solved numerically or analytically, attempts to recreate a given physical sound field for many listeners break down when the number of speakers is small. In many cases, only half of the system is being addressed; research on spatial hearing and psychoacoustics still has a large role to play in the future development of multichannel, multi-listener sound systems.

1.5 Summary

Today, we use many results from the early work on teleconferencing at Bell Labs in developing solutions for modern multichannel, immersive systems. Schroeder's work on modeling, simulating, and measuring room acoustics form the fundamentals of our understanding of sound behavior in rooms and how we develop and test new systems. His work on stereo reproduction and perception forms the basis for

our motivation to explore multichannel systems that can present spatial auditory cues to participants. Early work on frequency shifting in public address systems was a first attempt at a problem that research is still focused on, multichannel acoustic echo cancellation. Directional microphone arrays, which are now digitally steerable, allow for a new direction to combat noise and echo in conferencing systems. While there is much work left to be done until immersive teleconferencing systems are equivalent face-to-face meetings, the systems we have today owe a lot to the focused work of Manfred Schroeder and many others at Bell Labs.

References

1. Schroeder, M., Nebeker, F.: Oral-History: Manfred Schroeder. IEEE History Center, Hoboken, NJ (1994)
2. Schroeder, M.R.: Digital-computer method of measuring concert-hall acoustics. J. Acoust. Soc. Am. **36**(5), 1011 (1964)
3. Atal, B.S., Schroeder, M.R., Sessler, G.M., West, J.E.: Evaluation of acoustic properties of enclosures by means of digital computers. J. Acoust. Soc. Am. **40**, 428 (1966)
4. Sessler, G.M., West, J.E.: Sound transmission over theatre seats. J. Acoust. Soc. Am. **36**(9), 1725–1732 (1964)
5. Atal, B.S., Schroeder, M.R.: Study of sound decay using ray-tracing techniques on a digital computer. J. Acoust. Soc. Am. **41**(6), 1598 (1967)
6. Schroeder, M.R.: New method of measuring reverberation time. J. Acoust. Soc. Am. **37**(6), 1187–1188 (1965)
7. Schroeder, M.: Integrated-impulse method measuring sound decay without using impulses. J. Acoust. Soc. Am. **66**(2), 497 (1979)
8. Farina, A.: Simultaneous Measurement of Impulse Response and Distortion with a Swept-Sine Technique. AES Convention, New York, NY (2000)
9. Meyer, J., Elko, G.: A highly scalable spherical microphone array based on an orthonormal decomposition of the soundfield. ICASSP **2002**(2), 1781–1784 (2002)
10. Merimaa, J.: Applications of a 3-D microphone array. In: AES Convention 112 (2002)
11. Gover, B.N., Ryan, J.G., Stinson, M.R.: Measurements of directional properties of reverberant sound fields in rooms using a spherical microphone array. J. Acoust. Soc. Am. **116**(4), 2138–2148 (2004)
12. O'Donovan, A., Duraiswami, R., Zotkin, D.: Imaging concert hall acoustics using visual and audio cameras. In: ICASSP 2008, pp. 5284–5287 (2008)
13. Harvey, F.K., Cisek, J.G., MacLean, D.J., Schroeder, M.R.: Some aspects of stereophony applicable to conference use. J. Audio Eng. Soc. **11**(3), 212–217 (1963)
14. Schroeder, M.R.: Improvement of acoustic-feedback stability by frequency shifting. J. Acoust. Soc. Am. **36**(9), 1718–1724 (1964)
15. Benesty, J., Morgan, D.R., Sondhi, M.M.: A better understanding and an improved solution to the problems of stereophonic acoustic echo cancellation. ICASSP **1997**(1), 303–306 (1997)
16. Herre, J., Buchner, H., Kellermann, W.: Acoustic echo cancellation for surround sound using perceptually motivated convergence enhancement. ICASSP **2007**(1), I-17–I-20 (2007)
17. Atkins, J.: Spatial Acoustic Signal Processing for Immersive Communication. Johns Hopkins University, Baltimore, MD (2011)
18. Sessler, G., West, J.E., Schroeder, M.R.: Toroidal microphones. J. Acoust. Soc. Am. **46**, 28 (1969)
19. Sessler, G.M., West, J.E., Kubli, R.A.: Unidirectional, second-order gradient microphone. J. Acoust. Soc. Am. **86**(6), 2063–2066 (1989)

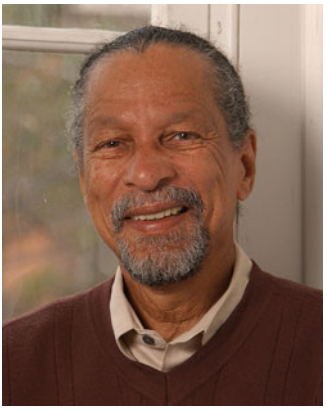
20. Sessler, G.M., West, J.E.: Second-order gradient unidirectional microphones utilizing an electret transducer. J. Acoust. Soc. Am. **58**(1), 273–278 (1975)
21. Flanagan, J.L., Berkley, D.A., Elko, G.W., West, J.E., Sondhi, M.M.: Autodirective microphone systems. Acustica **73**(2), 58–71 (1991)
22. Atkins, J.: Optimal spatial sampling for spherical loudspeaker arrays. In: ICASSP 2010, pp. 97–100 (2010)
23. Atkins, J.: Robust beamforming and steering of arbitrary beam patterns using spherical arrays. In: IEEE WASPAA 2011 (2011)
24. Schroeder, M.R.: An artificial stereophonic effect obtained from using a single signal. In: AES Convention 9 (1957)
25. Schroeder, M.R., Logan, B.F.: 'Colorless' artificial reverberation. J. Audio Eng. Soc. **9**(3), 192–197 (1961)
26. Harvey, F.K., Schroeder, M.R.: Subjective evaluation of factors affecting 2-channel stereophony. J. Acoust. Soc. Am. **32**(11), 1521 (1960)
27. Atal, B.S., Schroeder, M.R.: Apparent sound source translator. US Patent 3,236,949, 1966
28. Schroeder, M.R.: Models of hearing. Proc. IEEE **63**(9), 1332–1350 (1975)
29. Schroeder, M.R., Gottlob, D., Siebrasse, K.F.: Comparative study of European concert halls: correlation of subjective preference with geometric and acoustic parameters. J. Acoust. Soc. Am. **56**, 1195 (1974)
30. Bauck, J., Cooper, D. H.: Generalized transaural stereo. In: AES Convention 93 (1992)
31. Ward, D.B., Elko, G.W.: Optimum loudspeaker spacing for robust crosstalk cancellation. ICASSP 1998 **6**, 3541–3544 (1998)
32. Kirkeby, O., Nelson, P.A., Hamada, H.: Local sound field reproduction using two closely spaced loudspeakers. J. Acoust. Soc. Am. **104**(4), 1973–1981 (1998)
33. Gardner, W.G.: 3-D Audio Using Loudspeakers. Massachusettes Institute of Technology, Cambridge, MA (1997)
34. Berkhout, A.J., de Vries, D., Vogel, P.: Acoustic control by wave field synthesis. J. Acoust. Soc. Am. **93**, 2764 (1993)
35. Gerzon, M.: Psychoacoustic decoders for multispeaker stereo and surround sound. In: AES 93rd Convention (1992)
36. Pulkki, V.: Virtual sound source positioning using vector base amplitude panning. J. Audio Eng. Soc. **45**(6), 456–466 (1997)
37. Spors, S., Rabenstein, R., Ahrens, J.: The theory of wave field synthesis revisited. In: AES Convention 124 (2008)
38. Poletti, M.A.: Three-dimensional surround sound systems based on spherical harmonics. J. Audio Eng. Soc. **53**(11), 1004–1025 (2005)

***

Biography



Joshua Atkins received the B.S. degree in Electrical Engineering with a minor in Computer Science from Johns Hopkins University in 2005. He joined the Electro-optics group at the Johns Hopkins University Applied Physics Lab and completed his M.S. in Electrical Engineering in 2006 with a dissertation on the polarization characteristics of fluorescence from biological aerosols. In 2011, he received his Ph.D. from the Department of Electrical and Computer Engineering at Johns Hopkins University for dissertation work on immersive communication systems. Since then, he has been working at Beats Electronics where he leads the speech and audio processing research group. His research interests include microphone and loudspeaker array design and signal processing, statistical and adaptive signal processing, models of human auditory perception, and musical and architectural acoustics.



James E. West is currently Research Professor at Johns Hopkins University, Department of Electrical and Computer Engineering and the Department of Mechanical Engineering. He was formally a Bell Laboratories Fellow, at Lucent Technologies.

Professor West holds more than 50 U.S. and 200 foreign patents on various microphones and techniques for making polymer electrets and transducers. He was inducted into The National Inventors Hall of Fame in 1999 for the invention of the electret microphone. West is a member of the National Academy of Engineering, a Fellow of IEEE and the Acoustical Society of America He is the recipient of the Acoustical Society of America's Silver Medal in Engineering Acoustics (1995), an honorary Doctor of Science degree from New Jersey Institute of Technology (1997). West was awarded the Mexican Institute of Acoustics' John William Strutt 3rd Baron of Raleigh Award (2003), the Acoustical Society of America's Gold Medal (2006), an honorary Doctor of Engineering from Michigan State University (2006), the National Medal of Technology (2006) and the Franklin Medal in Engineering (2010), an honorary Doctors degree from University of Pennsylvania(2013). Professor West has been active in creating and supporting programs aimed at improving diversity in the Science, Technology, Engineering and Mathematics (STEM) fields.

Chapter 2
The Early History of Ray Tracing in Acoustics

Asbjørn Krokstad, U. Peter Svensson, and Svein Strøm

Abstract Manfred Schroeder and Bishnu Atal are behind two seminal conference presentations on room acoustics, one in 1962 and the other in 1967. In the first, computerized auralization was outlined remarkably early, long before the term auralization was introduced. In the second, a ray tracing technique in room acoustics was presented. Independently, Asbjørn Krokstad and colleagues in Norway worked on the ray tracing technique, which lead to a journal paper in 1968. Here we describe some developments that lead up to these two breakthroughs for the use of computers in room acoustics.

2.1 Introduction

Bishnu Atal and Manfred Schroeder presented a paper titled "Study of sound decay using ray-tracing techniques on a digital computer" at the 73rd Meeting of the Acoustical Society of America in New York, April 1967, but only an abstract was published for this presentation [1]. In 1968, a similarly titled paper, "Calculating acoustical room response by use of a ray tracing technique" was published by Asbjørn Krokstad, Svein Strøm, and Svein Sørsdal in the relatively recent Journal of Sound and Vibration [2]. The latter was submitted in November in 1967, but no reference to the Atal and Schroeder presentation was made—since neither of the authors went to the ASA meeting. At the same time, Krokstad remembers the great inspiration from another conference paper by Schroeder, Atal, and Bird at the ICA congress in Copenhagen in 1962 on "Digital computers in room acoustics" [3]. Schroeder did eventually, in 1970, publish a journal paper titled "Digital simulation of sound transmission in reverberant spaces" in the Journal of the

A. Krokstad (✉)
Department of Electronics and Telecommunications, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: asbjorn.krokstad@iet.ntnu.no

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_2,
© Springer International Publishing Switzerland 2015

Acoustical Society of America, submitted in February 1969, which described the ray-tracing technique but which did not cite the paper by Krokstad et al. [4]. Curious as this may seem to the researcher generation who has grown up with access to most publications right at the computer, in the times before efficient computer search engines it was quite a tedious work to look for all potentially relevant references.

Being the first published full paper, the paper by Krokstad et al. has become the standard work to cite for later studies on computerized prediction in room acoustics and has been a remarkably relevant paper to cite throughout more than 40 years. It was the first journal paper that presented the computerized ray-tracing technique for finding the impulse response, or rather echogram, in any three-dimensional model of a room. The reason that it has stayed relevant is that computerized prediction techniques still is an active research area, and such prediction still relies on the ray-tracing technique to a large extent. As an example, the two leading commercial softwares that are used by practitioners today, CATT Acoustic [5] and Odeon [6], use variants of the ray-tracing technique. In this paper, we will try to present some of the work in Trondheim and other places that lead up to the publishing of this paper. It will be a mix of a view from the inside (by the first and third author) and from the outside (by the second author). We will also see that the development was progressing in quite natural steps, yet at one point the development was perceived to have reached a mature enough stage that the "definitive" citation resulted. We will also see that in Trondheim there was an ongoing development of, and practical use for, the ray-tracing technique from the mid-1960s and onwards. The initial work by Atal and Schroeder, on the other hand, wasn't followed up to any larger degree, simply because there were several other highly significant developments going on, such as the linear prediction speech coder [7].

2.2 First Studies by Allred and Newhouse in 1958: Mean Free Path Calculations

In the late 1950s, researchers started to use computers quite widely for solving demanding numerical problems. The computers offered a new and more efficient tool to solve an existing problem, or to get more accurate results than had been possible before.

The field of room acoustics gives an interesting example of how the computers offered a straightforward improvement to existing numerical problems. Sabine's famous equation from 1898 states that the sound level in a room decays linearly with time, that is, an exponential decay for the sound pressure amplitude. The decay rate, or rather reverberation time, T_{60}, was determined by a very simple relationship, the first equality below,

$$T_{60} = 0.163\,\left[\mathrm{m}^{-1}\mathrm{s}\right]\frac{V}{\overline{\alpha}S} = 0.041\,\left[\mathrm{m}^{-1}\mathrm{s}\right]\frac{l_m}{\overline{\alpha}} \tag{2.1}$$

where V is the room volume (in m^3), $\overline{\alpha}$ is the average absorption coefficient, and S is the total surface area of the room (in m^2). Researchers have been kept busy ever

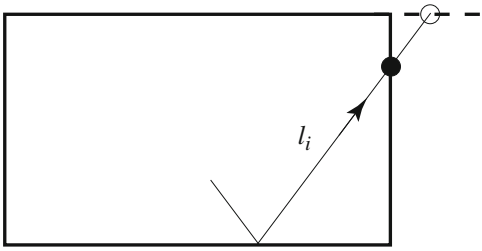


Fig. 2.1 Illustration of the Allred and Newhouse algorithm from 1958 [9]. Their paper studied three-dimensional shoe-box-shaped rooms; here a two-dimensional version is drawn. Each ray was reflected specularly and the next hit point was chosen as the closest of the three (two for the two-dimensional version) possible cross-points with other planes. The data sought were the lengths of the free paths, l_i

since studying to which degree this simple formula is true in various room geometry and absorption distribution cases. This relationship could also be written on a form which involves the so-called mean free path length in a room, that is, the average length that a sound wave can travel between wall hits, l_m (in m), and that is the second equality above.

This mean free path length had been established theoretically for some simple room shapes but novel approaches were used to find the mean free path in rooms of other shapes. In his book from 1932, *Architectural Acoustics*, Vern Knudsen described experiments where they used light rays in a scale model to figure out where sound waves would hit during several consecutive specular reflections [8]. By marking hit points on walls and measuring the corresponding lengths, he could collect data on these path lengths. Obviously, the accuracy was quite limited but he did find that quite different room shapes seemed to give the same average path length, or "mean free path." Such optical techniques in scale models continued to be in use, but also drawings, a pen and a ruler offered similar possibilities.

A first step towards computerized ray tracing was taken by collecting data on the path lengths and the collision probabilities of the walls using a computerized Monte Carlo technique by Allred and Newhouse in 1958 [9]. They studied rectangular/shoe-box-shaped rooms and Fig. 2.1 illustrates that the algorithm was quite straightforward. The algorithm generated rays in a number of randomized directions from one source position. For each ray, its path was followed through a succession of specular reflections. Data on path lengths and which reflecting walls that were hit were stored. Allred and Newhouse did not specify what kind of computer they used, or calculation times, but they could follow ten consecutive reflections for each ray, and they used 150 rays that were emitted in random directions.

This computerized calculation method gave accurate values for the mean free path in shoe-box rooms of different room dimension aspect ratios, and showed that there will be different reverberation times in rooms with different ratios. Two follow-up papers pointed out two errors in the original paper: the authors themselves found a programming error that gave around 10 % too long rev. times in the first paper [10]. Then Hunt, in 1964, showed that they had used a randomization of

the source emission directions which did not correspond to an isotropic sampling of the sound field [11].

Allred and Newhouse did outline an extension of their work in [9]:

> The Monte Carlo method of machine computation is shown to be applicable to the evaluation of room acoustics. Its extension to the study of reverberation time is straightforward, and air absorption, as well as frequency dependence of air and absorbers, can also easily be taken into account. .... It would be a simple matter also to vary the absorption coefficient as a function of position, either discretely or continuously, ...
>
> The method is also applicable to the study of coupled rooms and rooms of irregular shape, such as auditoria. The complications are geometrical, and are of degree rather than kind as referred to our parallelopiped calculations.

Apparently, they did not follow this path of development, and the reverberation time was their focus also for future directions of the research.

2.3 Schroeder at ICA in 1962: The Path Towards Auralization

A visionary conference paper by Schroeder, Atal, and Bird was presented at the ICA in Copenhagen in 1962 titled "Digital computers in room acoustics" [3], as mentioned above. That paper was remarkable in that it laid out the methodology for what much later was to be called auralization, that is, the technique for creating audible computerized simulations of the sound in a room [12]. The auralization process was still in the early 1990s a computationally very demanding process but Schroeder and colleagues had identified the necessary steps:

The use of computerized convolution between a (simplified) impulse response and anechoic music.

The cross-talk cancellation technique for the presentation of binaural sound over two loudspeakers in an anechoic room, which can give the impression of sound incidence from any direction.

The development of a natural sounding reverberation unit without needing to carry out a convolution with a huge number of discrete reflections.

Interestingly, Schroeder mentioned that discrete echoes can be handled by discrete delay units but otherwise did not describe the computation of a detailed impulse response or how these discrete echoes should be found. Rather more focus was given to the use of computers for implementing Schroeder's earlier (1954) seminal work on the statistics of frequency response functions [13].

2.4 Developments in Oslo 1965–1967: Diffuseness of Reverberation Rooms with Hanging Reflectors

At the University of Oslo, Department of Physics, there were two professors who were both interested in room acoustics, but using different approaches: Johan Holtsmark, who published a paper titled "Reverberation as a stochastic process"

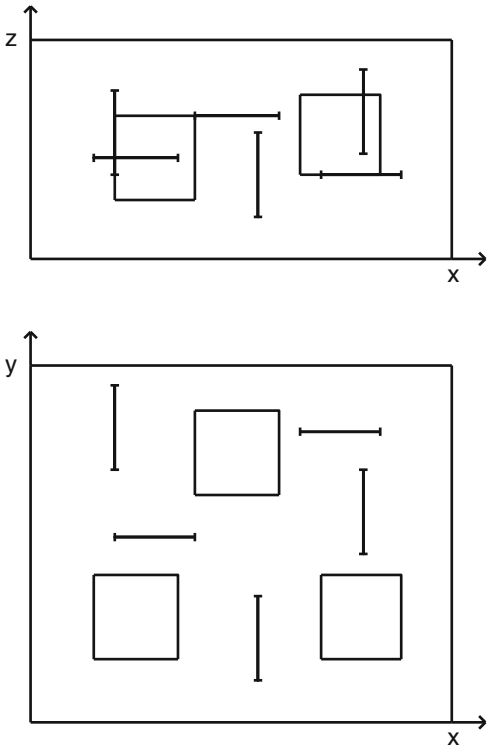


Fig. 2.2 Illustration of a shoe-box-shaped room with free-hanging reflectors. Reproduction from [15] where reflectors parallel to the walls were used

in 1966 [14]; his colleague Wilhelm Løchstøer was also interested in the reverberation process and supervised a Master's student Nikolai Stenseng who published his master thesis in January 1965 titled "The influence of diffusers on sound absorption measurements of a small volume" (translation of the Norwegian title) [15]. This thesis by Stenseng was a gradual development from the previous studies at that time in that it implemented a ray-tracing technique for a 3D room, albeit shoe-box-shaped, with free-hanging diffusors. These diffusers were always parallel to one of the walls, which made the computation of obstruction checks somewhat easier. Since reverberation rooms typically have smooth and flat walls and reflectors, only specular reflections were implemented and the need for something else was not discussed either. The program was written in the language Fortran IV for the computer UNIVAC 1107 (which had been delivered to the Norwegian Computing Center in Oslo in August 1963, after having been introduced in the USA in October 1962). An illustration from the thesis by Stenseng shows the parallel diffusers, see Fig. 2.2. Stenseng had spent a period at the NATO underwater research centre in La Spezia in Italy in the early 1960s, and remember being introduced to the power of the computers for the first time during that visit. When he returned to Oslo, he took one of the first programming courses offered, and when he started his master thesis, the idea to use a computer to simulate the propagation of light rays came quite naturally.

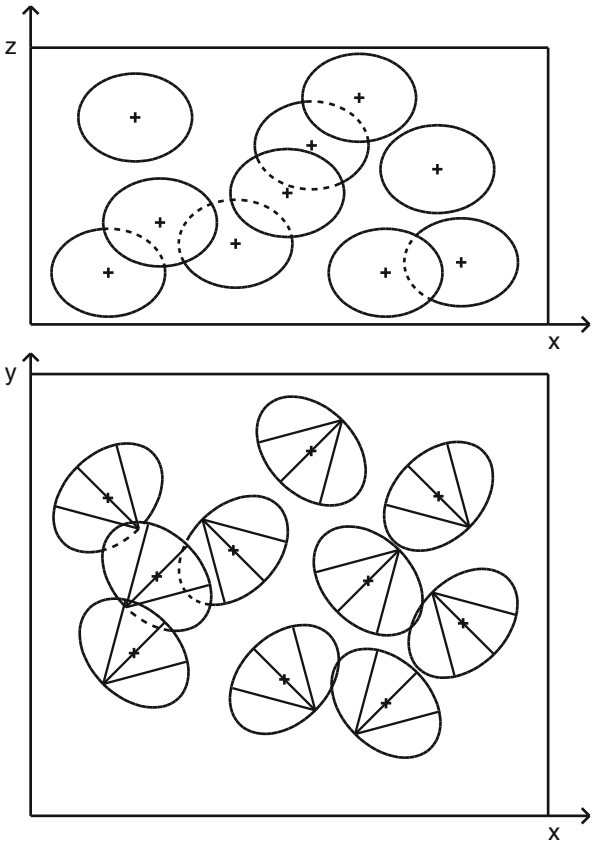


Fig. 2.3 Illustration of room with oblique reflectors (reproduced from [16])

An extension of this study by Stenseng was the master thesis by Svein Strøm in 1967, on "The diffusivity in small reverberation rooms with and without diffusers" (translation of the Norwegian title), also under Løchstøer in Oslo [16]. Strøm expanded Stenseng's study by implementing oblique reflectors, and also spherical reflectors. Furthermore, Strøm also computed actual decay curves, rather than collecting statistics data on path lengths. The purpose was still, however, to study the reverberation time. Figure 2.3 shows an example of a room with reflectors from [16].

2.5 Developments in Trondheim, Leading up to the Paper in 1968

Asbjørn Krokstad was hired as a research assistant at the Norwegian Technical Institute in Trondheim in 1956, and got the task to manage the specification of a new anechoic chamber, which was finished in 1958.[1] Soon after that, he got the further task to evaluate the performance of this new anechoic chamber. Since the chamber was rather small, compared to the recently finished one in Lyngby, Denmark, the classical free-field distance law criterion was difficult to apply, and Krokstad wanted to study the influence of the first-order reflections represented by image sources. The Institute got its first computer, a "GIER" from Denmark, in late 1962. Krokstad took the first course offered in Trondheim on the ALGOL programming language in 1963 (500 students and employees took this first course on computer programming!). During the course he started to use computer calculations to quantify the influence of reflections. The work on evaluating the anechoic chamber [17] later became part of his Ph.D. degree in 1963, which had Wilhelm Løchstøer and Gordon Flottorp, both from Oslo, as opponents (the supervisor of Krokstad was Reno Berg, associate professor in acoustics in Trondheim since 1935). As mentioned above Løchstøer was associate professor at the University of Oslo, Department of Physics, specializing on acoustics, whereas Flottorp was chief audiologist at the Oslo University Hospital. Curiously, this trio of supervisor and opponents were the three founders of the Acoustical Society of Norway in 1955.[2]

Krokstad got inspired by the successful application of the first-order image source method to the anechoic room and wanted to extend this to more general room shapes (and less absorbing ones too). Beranek had published his classical book "Music, acoustics & architecture" in 1962, and the book contained simplified plans for 100 of the most famous concert halls in the world [19]. This book gave the possibility to get input data for further studies, and furthermore, a possibly crucial event took place in Norway around this time: a new concert hall, "Grieghallen," was being planned in Bergen. An architectural bidding process was launched with a deadline in February 1965 and the winning design was made by the Danish architect Knud Munk. Krokstad naturally got very interested in studying the suggested design of the new hall.

The acoustics group in Trondheim, which had a brand new laboratory, opened in March 1965 [20], with Krokstad as director, hired Svein Sørsdal as a computer-savvy summer student, and the two of them tried to implement an image source

[1] The anechoic chamber in Trondheim still, in 2014, stands as it did in 1958 (but is rather ready for renovation). It has the quite unusual feature of graphite-covered wool wedges in order to improve the anechoic properties for electromagnetic waves.

[2] Reno Berg was also one of the founders of the *Nordic Acoustical Society*, founded on 19 June 1954 which happened to be the day of the 25th anniversary of the Acoustical Society of America. A telegram was sent by the "newly born son" to the "grandfather" ASA, as referred to in [18].

visibility test for general 3D geometries on the GIER computer that they both had experience with. This failed, however, since the visibility tests turned out to be too complex. They changed horses, so to say, aiming at using ray tracing instead and readily got in touch with Stenseng, and later Strøm, who were both in Oslo. At the same time, the Institute got their next type of computer, a UNIVAC 1107, in the end of 1965, after the GEIR computer had been running non-stop for a few years.

Strøm moved to Trondheim after he had finished his master thesis in early 1967 and joined the simulation project. His Fortran implementation of a 3D visibility test turned out to be exactly what was needed, and was combined with the ALGOL program developed earlier. Sørsdal implemented the main program structure, routines for reading geometry data, as well as for post-processing the result for each ray. A visualization technique was implemented where each ray's hit point in the audience area was plotted, with a little tail indicating the incidence angle onto a receiving surface. A Kingmatic plotter had been acquired by the Institute in 1966 and it made such visualizations possible.

Strøm carried out most of the actual calculations, and used the program as the Trondheim group were acoustics consultant in a sequence of projects: the "Hjertnes Kino og Kulturhus" in Sandefjord, opened in 1975; Grieghallen in Bergen, opened in 1978; and Oslo concert hall, opened in 1977. Prior to these projects, several of the halls in Beranek's book were studied and published in a report (in Norwegian) in 1971 [21]. The Grieghallen project had got delayed quite severely after the design competition had been settled in 1965. Locally in Bergen, Helmer Dahl, research director at the Christian Michelsen Research Institute, had got involved in the plans to build the new concert hall, and had tried to use the old lightbeam method of Vern Knudsen for studying different room shapes. These attempts had failed and Dahl therefore had contacted Krokstad who had the ray-tracing program up and running.

Svein Strøm describes that the small Trondheim group was hired as advisors for the Grieghallen project in Bergen [22]. The result was that the architect had to accept the suggestions from Trondheim for the orchestra enclosure, the ceiling shape, the sidewall reflectors, and for large sound reflecting objects that were hung from the ceiling along the sidewalls (Fig. 2.4). The exciting trial concert on the 12th of May, 1978, showed that the project was successful. Ten years later the local newspaper, *Bergens Tidende*, confirmed this in an article about the hall.

The journal paper was finished and submitted in November 1967. In order to develop a functioning engineering tool, they had taken several steps from the previous state of the art where ray length histories could be collected. In addition to the handling of arbitrary 3D geometries (admittedly with small numbers of polygons: up to 99 corners and 50 polygons per symmetric half of a room; and typically with around 2,000 emitted rays [21]), Krokstad et al. introduced some refinements [23]:

1. Some areas were marked as audience surfaces, and ray histories were collected for these. Also smaller sub-areas were introduced in order to study the sound field distribution over the audience area, see Fig. 2.5 (from [2]).



Fig. 2.4 Grieghallen in Bergen, one of the first concert halls where computerized ray tracing was used as a design tool

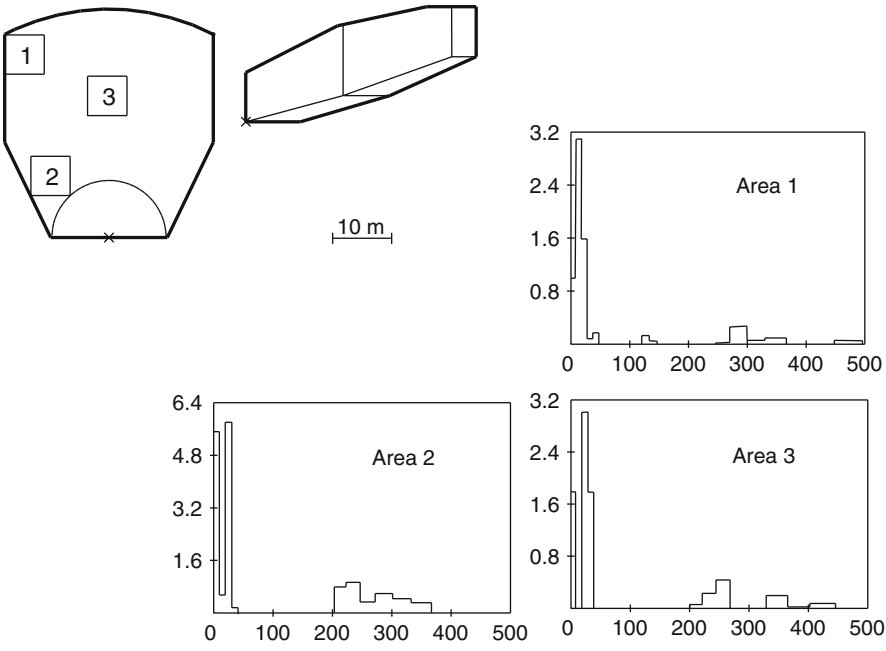


Fig. 2.5 Illustration of an irregularly shaped room, and computed echograms for three different receiver patches (reproduced from [2])

2. A kind of echogram was collected for each such audience surface, see Fig. 2.5. These resulted, however, simply from counting the number of hits per discrete time interval.
3. Different absorption coefficients were introduced—sort of. Surfaces were classified as totally reflecting, totally absorbing, or audience (which were also considered as totally absorbing). The motivation was that the room shape determined, to a large degree, the amount of early energy and its spatial distribution.

There were clearly some simplifications compared to what is common today, but some of these were commented by the authors in [2].

On the topic of specular reflections: "However, in the actual design of a hall, one may also wish to know the effects of changes in details, such as the introduction of diffusing wall elements. Even if such details can be included in the mathematical model of a hall without causing excessive computing time, the validity range of the results may be difficult to state." It was, however, not indicated how to handle diffuse reflections.

On the topic of absorption: "As mentioned in Sect. 2.2, the surfaces of the room are considered to be either totally reflective or totally absorbent. Other absorption characteristics of the surfaces, together with air damping, may readily be included in the computations. The authors believe that these factors are of minor interest in view of the building materials commonly used in concert halls."

2.6 Schroeder at the ASA Meeting in 1967: Computing Decays in Generally Shaped Convex Rooms

After the Allred and Newhouse study, and some subsequent theoretical discussion papers on mean free paths, other studies were looking into the computation of reverberation time (rather than mean free path and collision frequencies). At each reflection, the ray's energy was decreased by the wall reflection factor. Then, each ray gave a stepwisely decreasing contribution, and averaging across many of them gave a smooth exponentially decaying curve. This extension to the ray-tracing technique was presented by Atal and Schroeder at the ASA meeting paper in April 1967 [1], completely independently of the work in Norway. ASA meetings don't publish any papers, so all that is available is an abstract, plus some example results in two later papers by Schroeder in 1969 and 1970 [4, 24]. It did seem like Atal and Schroeder had a functioning program for general geometries, see Fig. 2.6, however, only for two dimensions.[3]

[3] In a paper by Wayman and Vanyo in 1977 [25], the quite straightforward extension of Schroeder's approach from two to three dimensions was presented, with no reference to the 1968 paper by Krokstad et al.

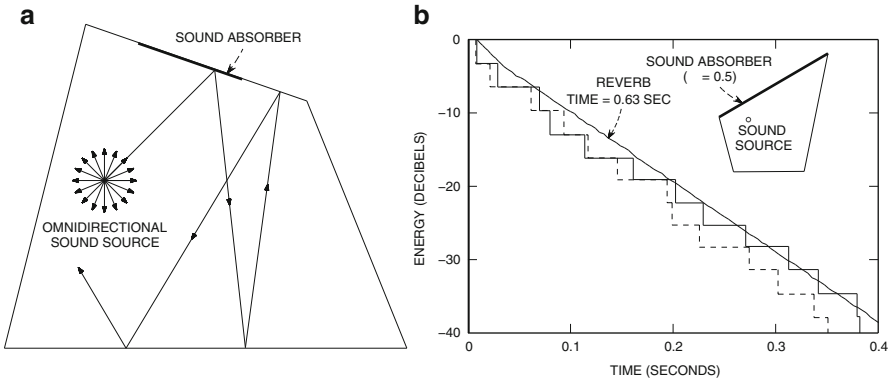


Fig. 2.6 Illustration of how the ray-tracing technique was used to gather individual ray-decay curves, which could be averaged to a smooth exponential decay (reproduced from [24])

The Atal and Schroeder algorithm was presented in the JASA paper in 1970 and their implementation clearly handled various degrees of absorption coefficients, as opposed to the approach by Krokstad et al. Furthermore, Atal and Schroeder specified the handling of diffuse reflections: "These rays are traced on the computer and are reflected from the walls either specularly or according to some specified random law." It seems to have been implemented as well, according to a further comment in the paper: "The discrepancies found for rooms with randomly reflecting walls and 'suspended' diffusing elements were generally, although not always, somewhat smaller."

On the other hand, Atal and Schroeder did not study the distribution of received sound across any audience surface; instead the global decay curve was studied, see Fig. 2.6. An interesting detail of the 1970 JASA paper by Schroeder is its reference number 16, to a paper by Atal and Schroeder in JASA which is marked as "To be published," but it never seems to have been published.

2.7 Later Developments of Ray Tracing in Room Acoustics

After the paper by Krokstad et al. was published in 1968, it appeared as if the paper was unknown for some time, at least to American researchers. In 1973, a paper published by Haviland and Thanedar outlined a method to compute the detailed response in one specific receiver location using ray tracing—but only for rectangular rooms [26]. This paper cited the Schroeder 1967 ASA abstract but not the 1968 paper by Krokstad et al. Another paper, published in 1977, that also seemed to be unaware of the Krokstad paper, has been mentioned above [25]. On the other hand, citations of the Krokstad paper appeared in 1971 [27], 1973 [28], and later. Krokstad et al. summarized the work in Trondheim in the paper titled "Fifteen years' experience with computerized ray tracing" [29].

Of special relevance is the paper by Kuttruff in 1971 [27]. That paper presented an integral equation formulation (later, independent versions of this became

known as the radiosity method), and a discrete Monte Carlo-type solution where narrow beams where traced until they hit a wall, and then they were reflected in a direction generated by two random numbers so as to simulate a Lambert-like reflection. Probably, the beam was represented by a single center ray. From the paper: "In der so festgelegten Richtung läuft das Teilchen also wieder in den Raum, wird erneut an einer Wand gestreut usw., bis es schliesslich einmal auf die absorbierende Wand trifft, die seiner Wanderung ein Ende setzt.", or (our translation) "The narrow beam thus propagates through the room in the determined direction, hits another wall where it is scattered, etc., until it finally hits the absorbing wall, which will end its propagation."

A lot of research has been done to develop the ray-tracing technique further. Much work has studied accuracy issues since ray tracing is inherently a stochastic process (as long as there is at least one partly diffusing surface area in a room). In addition, there are surface-sampling issues for the deterministic part of the process. The so-called cone tracing was presented by Van Maercke et al. in 1986, replacing thin rays (hitting a receiver sphere) by propagating cones (hitting a receiver point) [30]. An important development was presented by Vorländer in 1989 with the so-called hybrid technique: ray tracing was used to find possible reflection paths but in a subsequent phase, the ray-tracing-identified specular reflection paths were replaced by their image source equivalents. An advantage was that the image source method gives exact reflection paths, which can be used for improved accuracy in the early path of the impulse response [31]. This linking between ray tracing and the image source methods was also explored through the beam-tracing technique presented by Walsh already in 1980, where ray bundles were treated as coherent "beams" [32]. This beam tracing was shown by Stephenson to facilitate the inclusion of diffraction and developments to avoid its computation time explosion [33]. Also, Funkhouser et al. have demonstrated very efficient implementations of beam tracing via special geometry structures [34].

A step towards avoiding the "either specular or diffuse reflection" approach in ray tracing was taken by Dalenbäck where a method let reflections generate both specular and diffuse reflections while avoiding an exponential growth in path number tracing as function of reflection order [35]. The ray-tracing technique was combined with another algorithm, radiosity, by Lewers [36]. Later, the Odeon software employed components from radiosity [37] in their implementation. In fact, ray tracing in a room with only diffusely reflecting walls, is a discretized/Monte Carlo-type solution of the integral equation in radiosity, which was presented by Kuttruff in 1971 as described above [27].

2.8 Ray Tracing in Other Fields

So, was room acoustics the pioneer field for the computerized ray-tracing technique? Not really; as early as 1954 examples from optics were published. A paper titled "Ray tracing on the Manchester university electronic computing machine" by

Black was published in the British Proceedings of the Physics Society [38]. This was not so surprising—the term ray tracing does after all come from optics, and ray tracing in optics is typically used to study the refraction through optical lenses. Compared to room acoustics the "direct wave" is dominating, and studied in detail, while reflections might be ignored. On the other hand, the medium is refractive so that the rays do not travel along straight lines.

In underwater acoustics, early work was apparently done using ray tracing in the US Navy. The abstract in a 1956 report by Anderson and Peterson claims that "This report discusses: (1) an acoustic intensity program for estimating convergence zone propagation loss using ray theory" [39]. More publically, in an abstract from an ASA Meeting in November 1961, Norris claimed that "A ray-tracing program which has been developed for the IBM 704 and 7090 computers yields results which show excellent agreement with experimental data. The program permits ray computations for arbitrary source and receiver depths, bottom profiles, and velocity structures. The tabulated results give actual ray trajectories as well as the geometrical spreading associated with each path." [40].

In underwater acoustics, a central problem is that the medium is not homogeneous, which leads to that straight-line rays cannot be used (similar to in optics). On the other hand, geometries are practically always modeled as 2D, with a flat top surface (the sea surface) and a deterministically or stochastically shaped bottom profile. Multiple reflections are studied but with more restricted geometries and lower orders of reflection. Of curious interest is also early ray-tracing work on so-called analog computers—electronic circuits that performed calculations such as integrations and differentiations, by, e.g., Graber et al. in 1961 [41].

Ray tracing has more recently been developed heavily in computer graphics. A paper by Whitted in 1980 is considered as an important foundation of ray tracing in computer graphics [42], but an earlier paper from IBM in 1969 is considered as the original citation [43]. This kind of ray tracing was similar to the one in acoustics, except that rays were "shot" from a receiver's eye, through a grid of pixels that represent the display, reflected off surfaces, and finally reaching light sources. The more acoustics-like approach of emitting rays from (light) sources is usually called "global illumination" in graphics. Because of the generally large interest in computer graphics, many researchers are active in this field and the techniques are developed rapidly. In addition, the hardware for computer graphics generation (graphics processing units, or GPUs) has recently been developing at a faster pace than general CPUs for computers. This has lead to that so-called general-purpose GPUs (GPGPU) can be used as numerical coprocessors for many kinds of numerical calculations with processing power many times higher than CPUs [44]. Acoustical ray tracing has been demonstrated on such GPGPUs [45, 46].

A last field where ray tracing has become a common tool is the study of radio wave propagation, indoors as well as outdoors in city environments. This application applies specular reflections (as in room acoustics) as well as transmission, and of multiple orders. An early paper on this technique was published in 1991 [47]. It could be noted that diffraction over edges has been employed in radio propagation studies for a long time, either based on classical Fresnel diffraction, or based on the

high-frequency asymptotic geometrical theory of diffraction (GTD) that was presented by Keller in 1962 [48].

2.9 Concluding Remarks

The longevity of the ray-tracing technique, and the number of fields that it is applied to, indicates a very flexible method which can handle many phenomena. Ray tracing is based on geometrical acoustics, and equivalent approximations to true wave fields. Consequently, there will always be wave-related aspects that are more or less difficult to represent but the concept of edge diffraction has been implemented to some degree.

This overview has clearly had a focus on the development in Norway, which we know the best, and on the early work by Atal and Schroeder. Early attempts that have been published through more or less easily available channels might certainly have slipped our attention, and we humbly apologize if this is the case. We think that the development of the ray-tracing technique in room acoustics is a nice example of quite a gradual development process where several steps have been taken by different researchers, at different places, and thereby reached a kind of breakthrough.

Acknowledgements The authors are grateful to discussions with Svein Sørsdal, Nikolai Stenseng, and Prof. Bishnu Atal for sharing their memories from their early work on ray tracing.

References

1. Atal, B., Schroeder, M.R.: Study of sound decay using ray-tracing techniques on a digital computer. J. Acoust. Soc. Am. **41**, 1598 (1967)
2. Krokstad, A., Strøm, S., Sørsdal, S.: Calculating the acoustical room response by the use of a ray tracing technique. J. Sound Vib. **8**, 118–125 (1968)
3. Schroeder, M.R., Atal, B.S., Bird, C.: Digital computers in room acoustics. In Proceedings of the 4th International Congress on Acoustics, Copenhagen (1962)
4. Schroeder, M.R.: Digital simulation of sound transmission in reverberant spaces. J. Acoust. Soc. Am. **47**, 424–431 (1970)
5. CATT Acoustic. www.catt.se
6. Odeon. www.odeon.dk
7. Atal, B.S.: The history of linear prediction. IEEE Signal Processing Magazine, Match 2006, pp. 154–161 (2006)
8. Knudsen, V.: Architectural Acoustics, pp. 133–141. Wiley, New York, NY (1932)
9. Allred, J., Newhouse, A.: Applications of the Monte Carlo method to architectural acoustics. J. Acoust. Soc. Am. **30**, 1–3 (1958)
10. Allred, J., Newhouse, A.: Applications of the Monte Carlo method to architectural acoustics. II. J. Acoust. Soc. Am. **30**, 903–904 (1958)
11. Hunt, F.V.: Remarks on mean free path problem. J. Acoust. Soc. Am. **36**, 556–564 (1964)

12. Kleiner, M., Dalenbäck, B.-I., Svensson, P.: Auralization—an overview. J. Audio Eng. Soc. **41**, 861–875 (1993)
13. Schroeder, M.R.: Die Statistichen Parameter der Frequenz Kurven von Grossen Räumen. Acustica **4**, 594–600 (1954)
14. Holtsmark, J.: Reverberation as a stochastic process. Phys. Norv. **2**, 63–70 (1966)
15. Stenseng, N.: Diffusorers innflytelse på lydabsorbsjonsmålinger i et klangrom med lite volum. M.Sc. Thesis, University of Oslo, Department of Physics (in Norwegian) (1965)
16. Strøm, S.: Diffusitet i små klangrom med og uten diffusorer. M.Sc. Thesis, University of Oslo, Department of Physics (in Norwegian) (1967)
17. Krokstad, A.: Design and construction of an anechoic chamber. Report. Department of Electrical Engineering, Technical University of Norway, Trondheim, (1964)
18. Firestone, F.A.: The twenty-fifth anniversary celebration. J. Acoust. Soc. Am. 26, 879 (1954)
19. Beranek, L.L.: Music, Acoustics and Architecture. Wiley, New York, NY (1962)
20. Nordahl, O.: Akustisk Laboratorium ved NTH/SINTEF. Master Thesis, Department of History, NTNU, Trondheim (in Norwegian) (2005)
21. Strøm, S.: Fordeling av reflektert lyd i modeller av eksisterende konsertsaler, Report LBA 381. Akustisk Laboratorium, SINTEF, Trondheim (1971) (in Norwegian)
22. Strøm, S.: A look back on 40 years of cooperation with Asbjørn on room acoustics consulting for multi-purpose halls. In: Svensson, P. (ed.) Reflections on Sound. In Honour of Professor Emeritus Asbjørn Krokstad. Norwegian University of Science and Technology, Department of Electronic and Telecommunication, Trondheim (2008)
23. Krokstad, A.: The hundred years cycle in room acoustic research and design. In: Svensson, P. (ed.) Reflections on Sound. In honour of Professor Emeritus Asbjørn Krokstad. Norwegian University of Science and Technology, Department of Electronics and Telecommunication, Trondheim (2008)
24. Schroeder, M.R.: Computers in acoustics: symbiosis of an old science and a new tool. J. Acoust. Soc. Am. **45**, 1077–1088 (1969)
25. Wayman, J., Vanyo, J.: Three-dimensional computer simulation of reverberation in an enclosure. J. Acoust. Soc. Am. **62**, 213–215 (1977)
26. Haviland, J., Thanedar, B.: Monte Carlo applications to acoustical field solutions. J. Acoust. Soc. Am. **54**, 1442–1448 (1973)
27. Kuttruff, H.: Simulierte Nachhallkurven in Rechteckräumen mit diffusem Schallfeld. Acustica **25**, 333–342 (1971)
28. Juricic, H., Santon, F.: Images et rayons sonores dans le calcul numerique des echogrammes. Acustica **28**, 77–89 (1973)
29. Krokstad, A., Strøm, S., Sørsdal, S.: Fifteen years' experience with computerized ray tracing. Appl. Acoust. **16**, 291–312 (1983)
30. Van Maercke, D.: Simulation of sound fields in time and frequency domain using a geometrical model. In Proceedings of 12th International Congress on Acoustics, Toronto, paper E11-7 (1986)
31. Vorländer, M.: Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm. J. Acoust. Soc. Am. **86**, 172–178 (1989)
32. Walsh, J.P.: The design of Godot: a system for computer-aided room acoustics modeling and simulation: In: Proceedings of 10th International Congress on Acoustics, Sydney, Paper E15.3 (1980)
33. Stephenson, U.: Quantized pyramidal beam tracing—a new algorithm for room acoustics and noise immission prognosis. Acta Acustica united with Acustica **82**, 517–525 (1996)
34. Funkhouser, T., Tsingos, N., Carlbom, I., Elko, F., Sondhi, M., West, J.E., Pingali, G., Min, P., Ngan, A.: A beam tracing method for interactive architectural acoustics. J. Acoust. Soc. Am. **115**, 739–756 (2004)
35. Dalenbäck, B.-I.: Room acoustic prediction based on a unified treatment of diffuse and specular reflection. J. Acoust. Soc. Am. **100**, 899–909 (1996)

36. Lewers, T.: A combined beam tracing and radiant exchange computer model of room acoustics. Appl. Acoust. **38**, 161–178 (1993)
37. Naylor, G.: Treatment of early and late reflections in a hybrid computer model for room acoustics. J. Acoust. Soc. Am. **92**, 2345 (1992)
38. Black, G.: Ray tracing on the Manchester university electronic computing machine. Proc. Phys. Soc. Sect. B **67**, 569–574 (1954)
39. Anderson, E.R., Pedersen, M.A.: Application of Computer Techniques to the Prediction of Underwater Sound Propagation, Report ADA066760. Navy Electronics Lab, San Diego, CA (1956)
40. Norris, A.: Ray-tracing program for high speed digital computers. J. Acoust. Soc. Am. **33**, 1659 (1961)
41. Graber, G., Rubin, A.: Underwater acoustic ray analysis. J. Acoust. Soc. Am. **33**, 840 (1961)
42. Whitted, T.: An improved illumination model for shaded display. Commun. ACM **23**, 343–349 (1980)
43. Appel, A.: Some techniques for shading machine renderings of solids. In: Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference, AFIPS '68, pp. 37–45 (1969)
44. www.gpgpu.org
45. Jedrzejewski, M., Marasek, K.: Computation of room acoustics using programmable video hardware. In: Wojciechowski, K. (ed.) Computer Vision and Graphics, Proceedings of International Conference ICCVG 2004, Warsaw, Poland, September 2004. Springer Netherlands, pp. 587–592 (2006)
46. Tsingos, N., Dachsbacher, C., Lefebvre, S., Dellepiane, M.: Instant sound scattering. In Kautz, J., Pattanaik, S. (eds.) Proceedings of the 18th Eurographics Symposium on Rendering 2007 (2007)
47. McKown, J.W., Hamilton, R., Lee, J.: Ray tracing as a design tool for radio networks. IEEE Netw. Mag. **5**, 27–30 (1991)
48. Keller, J.B.: Geometrical theory of diffraction. J. Opt. Soc. Am. **52**, 116–130 (1962)

***

Biography



Dr. **Asbjørn Krokstad** is a professor emeritus at the Norwegian University of Science and Technology, Trondheim, Norway, since 1998. He has a Ph.D. from the Norwegian Institute of Technology, Trondheim, in 1962, and was professor from 1970. He did research on computational room acoustics, with the first paper on ray tracing in 1968, reverberation enhancement systems, and was behind the development of one of the world's first digital hearing aid prototypes in 1989, and holds several patents related to this. He was president of the Norwegian acoustical society 1994–1998, and the general chairman of the ICA in Trondheim 1995. He has been an active musician and conductor throughout his career.



Dr. **U. Peter Svensson** is a professor of electro-acoustics at the Norwegian University of Science and Technology, Trondheim, Norway, since 1999. He has a Ph.D. from Chalmers University of Technology, Gothenburg, Sweden, in 1994. His research has dealt with auralization, especially computational methods involving diffraction modeling and loudspeaker reproduction techniques. He has also worked on beamforming techniques for microphone arrays, measurement techniques, reverberation enhancement, and interaction over Internet/video conferencing. He has been on the boards of the acoustical societies of Sweden, Norway, and the European Acoustics Association.



Svein Strøm is a senior adviser at COWI AS, Trondheim, Norway. He has an M.Sc. in physics from the University of Oslo in 1967. He worked at the ELAB/SINTEF in Trondheim with room acoustics research together with Dr. Krokstad until 1994. He did his M.Sc. thesis on the ray-tracing technique, which paved the way for the paper by Krokstad, Strøm, and Sørsdal in 1968. Strøm has worked with room acoustical design throughout Norway, including many examples of passive and active variable acoustics installations. Together with Prof. Krokstad, he was the acoustical consultant for the concert hall in Trondheim, Olavshallen, which was opened in 1992.

Chapter 3
Advanced Room-Acoustics Decay Analysis

Ning Xiang

Abstract Schroeder's integration method broke the new ground in classical architectural acoustics. Schroeder's integration method yields sound energy decay functions from room impulse responses. For reverberation time estimation based on a parametric model, a nonlinear regression method has been proposed. The nonlinear regression method yields reverberation time estimates, insensitive to background noise and the upper limit of integration. Recent interest in acoustically coupled-volume systems has prompted new challenges in analyzing sound energy decay characteristics, which are more complicated than just single-rate decays. This chapter will demonstrate a suitable framework for this room-acoustics application using Bayesian inference and the Schroeder's integration as the foundation of the advanced model-based energy decay analysis. Based on the Schroeder's integration, two levels of inference, decay order selection and decay parameter estimation, are discussed in details.

3.1 Introduction

In 1965, Manfred R. Schroeder published his seminal paper [1] on a new method of measuring reverberation time. The paper presents a novel method of obtaining sound energy decays in terms of integration of energy room impulse responses, *Schroeder integration*. The Schroeder integration method broke new ground in classical room-acoustics, and has since received wide acceptance in the architectural acoustics community. The sound energy decay function obtained by Schroeder integration is termed *Schroeder decay function*, or simply, *Schroeder curve* when

N. Xiang (✉)
Graduate Program in Architectural Acoustics School of Architecture, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
e-mail: xiangn@rpi.edu

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_3,
© Springer International Publishing Switzerland 2015

presented graphically. This chapter discusses a parametric model derived from Schroeder integration, which has been proven to fully describe Schroeder decay functions. The parametric model plays a central role within the scope of advanced sound energy decay analysis since no characteristics can be inferred from experimental measurements for the relevant features without an appropriate model. Based on the Schroeder decay model, this chapter introduces a model-based Bayesian probabilistic inference in advanced room-acoustic energy decay analysis where both decay order (model) selection and decay parameter estimation problems are to be solved. A brief introduction to a model-based nonlinear regression approach for the specific case of reverberation time estimation is also given.

Sound energy decays in room-acoustics investigations are traditionally determined using random-noise excitations; once this signal has brought the sound field to its steady state, the noise excitation is interrupted. Normally sound pressures at receiver locations are recorded, and the squared pressures over time are termed steady-state *energy-time* functions. Schroeder considered a sound source-receiver arrangement in a room under investigation to be a linear time-invariant system. When the sound source is driven by the interrupted noise, the receiver receives a sound signal represented by convolution between the room impulse response and the interrupted noise. The steady-state energy decay functions can be derived by integration of the energy room impulse response, so-called Schroeder integration [1]. This method, from a single measurement of a room impulse response, produces an energy decay function which is theoretically equivalent to an ensemble average of a large number of steady-state energy decay functions when using interrupted noise. Following Schroeder, a number of investigations of this method [2–12] have substantiated its advantages. The Schroeder decay functions are not only free of unnecessary random fluctuations, but they contain significant single or multiple exponential processes as well.

3.2 Schroeder Decay and Model

A normalized stead-state sound energy decay function $d(t)$, via Schroeder integration [1, 13], can be expressed by its discrete version in digital domain as:

$$d(t_i) = \frac{1}{E}\sum_{\tau=t_i}^{t_K} h^2(\tau) = 1 - \frac{1}{E}\sum_{\tau=0}^{t_i} h^2(\tau), \quad 0 \le i < K, \tag{3.1}$$

and

$$E = \sum_{\tau=0}^{t_K} h^2(\tau), \tag{3.2}$$

where variable t_i represents discrete time, i, K are integers with K being the total number of data points, so t_K represents the total time record length of *room impulse response* $h(t)$ as shown in Fig. 3.1a. The room impulse response $h(t)$ is defined between a sound source and a receiver in the room under investigation. In the following t_K is also termed *the upper limit of the integration*. E is the total energy contained in $h(t)$. Schroeder integration essentially establishes the relationship between the *energy room impulse response* $h^2(t)$ and the *steady-state energy decay function* $d(t)$ [7]. The right-hand side of Eq. (3.1) is of both theoretical and practical relevance, it represents the complementary relation between the energy decay and build-up [13], at the same time it is beneficial for numerical implementation due to a straightforward accumulative summation. In architectural acoustics practice, logarithmic energy room impulse responses $10\log_{10}[h^2(t)]$ often represented graphically over time, termed *energy-time curve* (ETC), as illustrated in Fig. 3.1b.

A parametric model (*Schroeder decay model*) with s exponential decay terms [14–16]

$$\mathbf{H}_s(\mathbf{A}_s, \mathbf{T}_s) = A_0(t_K - t_i) + \sum_{j=1}^{s} A_j \left[\exp\left(\frac{-13.8 \cdot t_i}{T_j}\right) - \exp\left(\frac{-13.8 \cdot t_K}{T_j}\right)\right], \quad (3.3)$$

can fully describe Schroeder decay function $d(t)$. Vector $\mathbf{A}_s = [A_0, A_1, \cdots A_s]$ contains $s+1$ linear coefficients, while vector $\mathbf{T}_s = [T_0, T_1, \cdots T_s]$ contains s decay time parameters. The subscript s in the model in Eq. (3.3) is the decay order, denoting that the decay model contains s different exponential decay terms with s different decay times. The models in the present application are nested models—a higher order model embodies lower ones. Particularly, if the second-order model describes the data reasonably well, the first-order or the third-order models often also describe the data well. When $s = 1; 2; 3$, it is said that the sound energy decays are of single-, double- or triple-slope nature, respectively. For single-slope decays, the corresponding decay time is termed *reverberation time*, while in multiple-slope decay cases, multiple different *decay times* are expected. When t_K is selected large enough, the constant term $\exp(-13.8 \cdot t_K/T_j)$ in Eq. (3.3) is negligible.

There are two major issues reported in published literature since 1965; Schroeder integration yields energy decay functions from experimentally measured room impulse responses, which sensitively depend on the upper limit of the integration [2] and inevitable background noise [5]. The Schroeder decay model in Eq. (3.3) helps us understand the characteristic behavior of the Schroeder decay functions. A full understanding of this characteristic behavior, in turn, facilitates the discussion of the advanced analysis tools elaborated on later in this chapter. It further reveals that the two reported issues, associated with the upper limit of the integration and background noise, are actually redundant. Along with Fig. 3.1c we summarize the following characteristic features

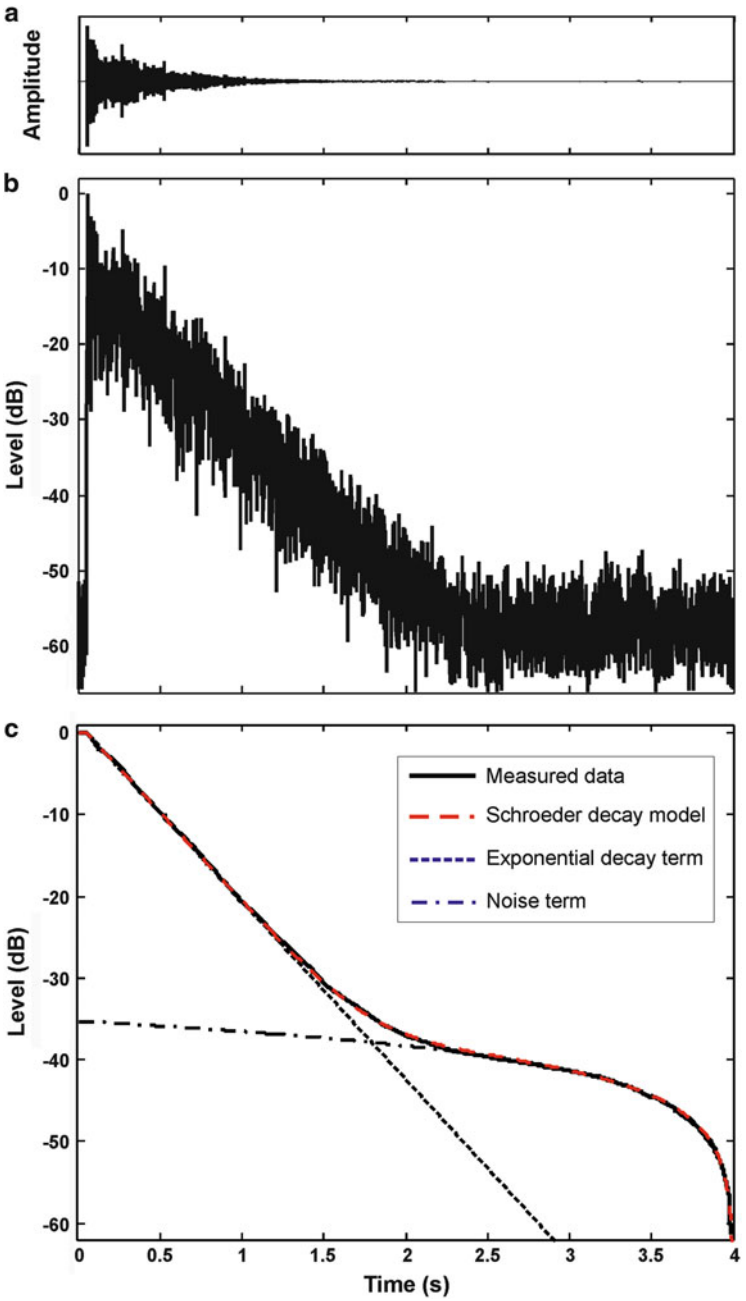


Fig. 3.1 Experimentally measured room impulse response, band-pass filtered at 1 kHz (Oct.), its corresponding energy-time curve and Schroeder decay curve. (**a**) Room impulse response. (**b**) Normalized energy-time curve (energy room impulse response) illustrated in logarithmic scale [level in (dB)], derived from the room impulse response shown in (**a**). (**c**) Normalized Schroeder decay curve illustrated in logarithmic scale [level in (dB)] derived from the energy room impulse response as shown in (**b**). The Schroeder decay model along with decomposed exponential decay term and the linear decay term is also illustrated for ease of comparison

1. The Schroeder decay curve will, generally (or very often), not exhibit a linearly decaying curve when graphically presented in a logarithmic scale [level in (dB)], even for the single-slope case. It can be characterized by three major portions.
   (a) *Beginning portion*: the decay curve decays within this portion with a slope or with slopes predominantly determined by room-acoustic properties, such as the reverberation time (decay times).
   (b) *Middle portion*: a curve often appears to decay with a somewhat slower slope in comparison to that in the beginning portion. Only when the upper limit of the integration is long enough this portion remains observable [see the curves between 1.5 and 2.5 s as shown in Fig. 3.1c], otherwise it can disappear so that the beginning portion transitions directly to the late portion.
   (c) *Late portion*: the decay curve ends up with a characteristic curvature towards the end of the time record of the room impulse response (the upper limit of the integration).
2. The linearly decreasing term $A_0(t_K - t_i)$ [see Eq. (3.3)] is predominantly responsible for the characteristic curvature of the late portion in the logarithmic scale, and it accounts for both the effect of the upper limit of the integration through parameter t_K [2] and the effect of background noise through parameter A_0; therefore, $A_0(t_K - t_i)$ is also termed the *noise term* in this chapter. It is actually a nuisance term, irrelevant to the room-acoustics problem at hand.
3. Schroeder decay functions become undefined at the upper limit of integration t_K [see Fig. 3.1c] when examined logarithmically since for $t_i \to t_K$, $d(t_i) \to 0$ and the parametric model accounts for this fact.

Luizard and Katz [6] recently proposed an alternative model to the envelope of the energy-time functions (energy room impulse responses)

$$\mathbf{H}_s(\boldsymbol{\alpha}_s, \mathbf{T}_s) = \alpha_0 + \sum_{j=1}^{s} \alpha_j \exp\left(\frac{-13.8 \cdot t_i}{T_j}\right). \tag{3.4}$$

The relationship between Schroeder decay functions and the energy time functions is already represented by Eq. (3.1) [1, 7]. Xiang et al. [8] also demonstrated the difference between the two types of decay functions using diffusion equation modeling results. The parameter estimates based on two different data models should not be directly compared without the conversion using Schroeder integration since Schroeder integration of the model in Eq. (3.4) between the time interval $[t_i, t_K]$ results in exactly the model in Eq. (3.3) with $A_j = \alpha_j T_j/13.8$, when neglecting the constant term $\exp(-13.8 \cdot t_K/T_j)$ in Eq. (3.3). The model-based analysis discussed in this chapter, including nonlinear regression and particularly two levels of Bayesian inference, can also be applied to the smoothed energy-time functions using the alternative model in Eq. (3.4). The following discussions in the context of model-based decay analysis, however, remain focused on the Schroeder decay functions.

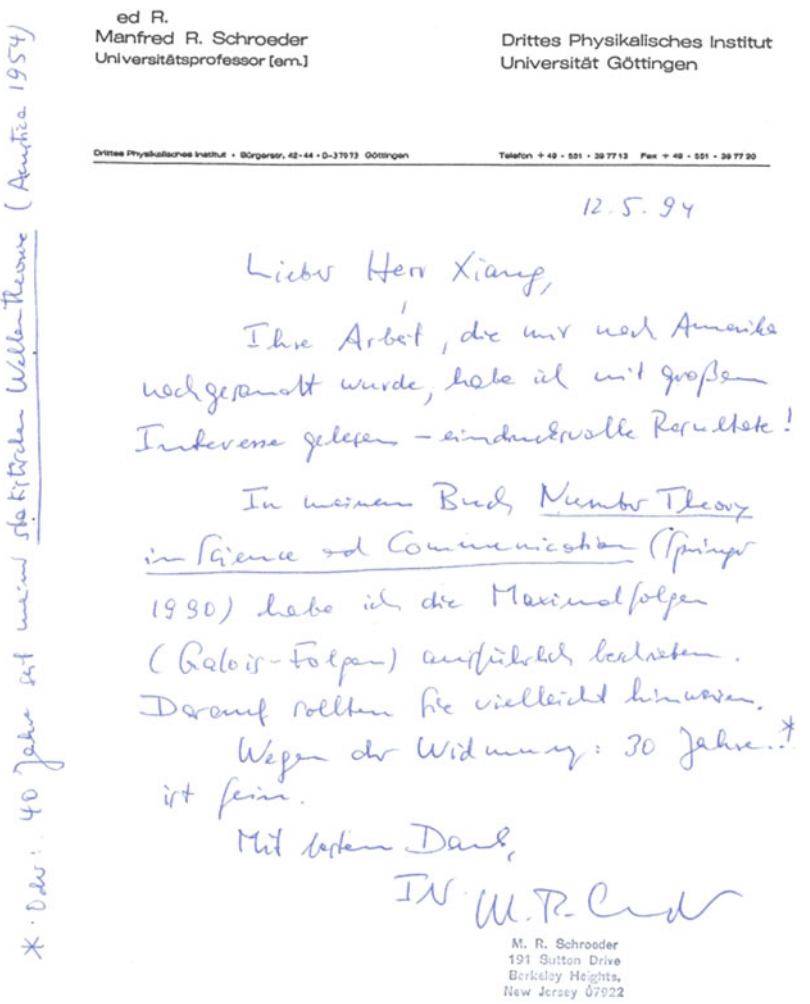
3.3 Linear and Nonlinear Regression

The characteristic features of Schroeder decay curves summarized above, along with the model in Eq. (3.3), help us understand the fact that the linear least-squares-fit approach to directly fit logarithmic Schroeder decay functions (curves) relies on an inappropriate model. A large number of investigations have been reported in major acoustics journals for decades after Schroeder published this method. The reported effort, among others, includes mitigation of the nonlinear decay curves [as shown in Fig. 3.1c] by careful selection of the upper limit of integration, the so-called truncation method [2, 9], or subtraction/elimination of background noise, the so-called subtraction method [5, 10, 11]. All these mitigation methods, however, employ linear (least-square) regression, implicitly relying on a linear decay model. The success of reverberation time estimation then depends critically on accurate estimation of the upper limit of the integration t_K, accurate estimation of A_0 associated with the background noise [14], or accurate elimination of the background noise [10]. In order to improve reverberation time estimates, iterative processes have also been proposed to accurately estimate the upper limit of the integration [12]. Separate multiple measurements at a single receiver location have also been proposed to reduce the effect of the background noise [10]. A close look at the model in Eq. (3.3) reveals that these methods depend critically on accurate estimations or removal of one or the other nuisance parameter (A_0, t_K).

In 1995, a nonlinear regression method [14] was introduced by this author[1] based on the nonlinear model in Eq. (3.3). For reverberation time estimates (single-slope case), the nonlinear regression removes the stringent requirement on accurate estimation of nuisance parameters, both the upper limit of the integration t_K or the parameter A_0 associated with the background noise. The nonlinear regression starts with a rough estimate of initial values of model parameters A_1, T_1, and A_0. It then iteratively improves the model parameters in terms of infinitesimal corrections of each individual parameter by solving a linear equation system. For the single-slope cases, it easily converges to the global minimum by properly estimating the initial values of A_1 and T_1, which can be straightforwardly estimated from a small segment at the beginning of a logarithmic Schroeder decay function. Accurate initial estimates of A_0 and t_K are uncritical, and t_K can also be set with ease to be somehow smaller or larger than T_1, such as one, two or even a multiple of the initial value of T_1 which has already been estimated [14], or simply set sufficiently large without any burden of accurate determination.

The nonlinear regression works efficiently for the reverberation time estimation (single-slope case) without the stringent requirement on parameters t_K and A_0 because it involves t_K and A_0 as adjustable model parameters in the regression process. The iterative process of the nonlinear regression will lead to convergence

[1] Dr. Wolfgang Ahnert was collaborative in the initial work [17, 18] by drawing this author's attention to a paper published in the *Computation Journal* in 1965 dealing with minimizing a sum of squares of nonlinear functions [14].



ed R.
Manfred R. Schroeder
Universitätsprofessor (em.)

Drittes Physikalisches Institut
Universität Göttingen

Drittes Physikalisches Institut • Bürgerstr. 42-44 • D-37073 Göttingen Telefon + 49 • 551 • 39 77 13 Fax + 49 • 551 • 39 77 20

12.5.94

Lieber Herr Xiang,

Ihre Arbeit, die mir nach Amerika nachgesandt wurde, habe ich mit großem Interesse gelesen – eindrucksvolle Resultate!

In meinem Buch Number Theory in Science and Communication (Springer 1990) habe ich die Maximalfolgen (Galois-Folgen) ausführlich beschrieben. Darauf sollten sie vielleicht hinweisen.

Wegen der Widmung: 30 Jahre..* ist fein.

Mit herzlichem Dank,

IN M. R. Schroeder

M. R. Schroeder
191 Sutton Drive
Berkeley Heights,
New Jersey 07922

* Das: 40 Jahre seit meiner statistischen Wellentheorie (Acustica 1954)

Fig. 3.2 Professor Manfred Schroeder's handwritten letter, shortly before publication of the nonlinear regression method [14] in the Journal of the Acoustical Society of America during the revision phase, the final draft was sent to Professor Manfred Schroeder for his critical comments. The first paragraph of the letter: "Your paper, relayed to me to America, I have read with great interest—impressive results!"

to the global minimum after only a few iterations even with rough estimates of initial values [14]. They require some added computational expenses, which are, on current PC-platforms, barely noticeable; on order of a few milliseconds. Professor Manfred Schroeder has shown a great interest in this work [see Fig. 3.2].

This author soon found out [19, 20], however, if the model order is higher than one (double- or multiple-slope cases), initial value estimation becomes too critical to allow convergence. For this reason, the model-based Bayesian method, elaborated in

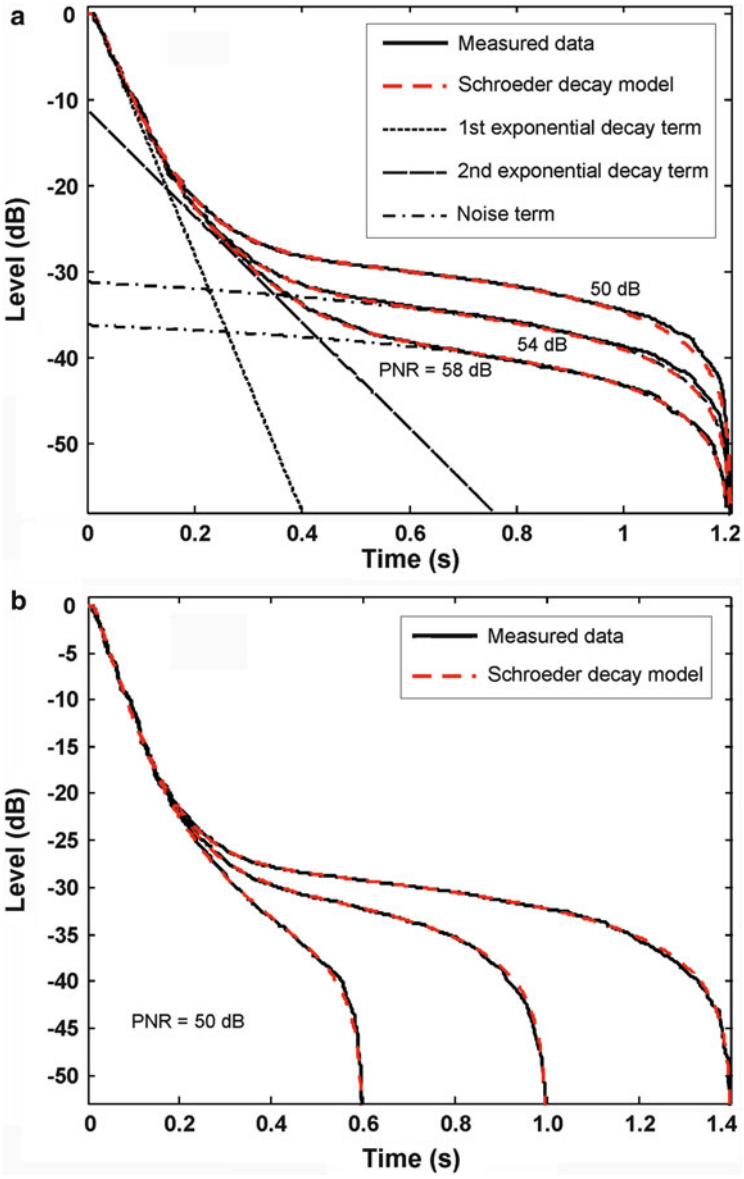


Fig. 3.3 Experimentally measured Schroeder decay curves at one specific source-receiver configuration. (**a**) Normalized Schroeder decay curves for a fixed upper limit of integration with its peak-to-noise ratio PNR = 50, 54, and 58 dB, respectively. Decomposed exponential terms and the noise term are also shown for ease of comparison. (**b**) Normalized Schroeder decay curves for a fixed peak-to-noise ratio (PNR = 50 dB) while changing the upper limit of integration $t_K = 0:6$ s; 1 : 0 s, and 1 : 4 s, respectively

the following, has emerged [20, 21]. Before describing Bayesian inference, a close look at experimentally measured data as illustrated in Fig. 3.3 motivates applications of advanced energy decay analysis methods. Figure 3.3a illustrates Schroeder decay

Table 3.1 Bayesian analysis of an experimentally measured Schroeder decay function in acoustic scale models as shown in Figs. 3.3 and 3.4, derived from a 1 kHz octave-band-pass-filtered room impulse response

	Single-slope	Double-slope	Triple-slope
BIC (neper)	−85.76	0.0	−81.08
A_0 (dB)	−62.40	(−58.5) −62.2 (−67.2)	−62.26
A_1 (dB)	−5.15	−5.15 (±3.4E-3)[a]	−5.26
T_1 (s)	0.483	0.411 (±9.3E-3)	0.39
A_2 (dB)	–	−14.91 (±1.3E-3)[a]	−14.24
T_2 (s)	–	0.765 (±9.7E-3)	0.768
A_3 (dB)	–	–	−20.43
T_3 (s)	–	–	1.22

Bayesian information criterion (BIC) along with decay parameters estimated using single-slope, double-slope, and triple-slope models are listed. The double-slope decay obtains the highest natural logarithmic BIC, by which the other BIC values are normalized. A_0 listed in column "Double-Slope" contains three different values corresponding to three different curves in Fig. 3.3a. The standard derivations dA_1, dT_1, dA_2, and dT_2 are also listed only for the double-slope case
[a]dA_1, dA_2 are listed linearly

curves, derived from a 1 kHz octave-band-pass-filtered room impulse response measured in a coupled-volume system at one specific receiver location, but with three different resulting peak-to-noise ratios (PNR = 50, 54, and 58 dB, approximately). The decay characteristics represented by these three curves are actually the same. According to Eq. (3.3), a model-based decay parameter estimation with $s = 2$ yields decay parameter values as listed in Table 3.1, (Column 3).

Using these estimated parameters, the decay model curves [see Eq. (3.3), plotted by dash lines in Fig. 3.3] agree well with the experimental data. The decay model curves consist of two exponential decay terms and one noise term [a linearly decaying term, $A_0(t_K - t_i)$]. The only parameter differing between the model curves is $10\log_{10}(A_0) = -58.5, -62.2,$ and -67.2 dB for the three cases, respectively. Figure 3.3a also illustrates the noise term for the two highest peak-to-noise ratios. Figure 3.3b illustrates the decay curves for three different upper limits of integration with a fixed peak-to-noise ratio of 50 dB. The Schroeder decay models with the exact five decay parameters listed in Table 3.1 (Column 3), but with different upper limits of integration $t_K = 0.6, 1.0,$ and 1.4 s, well describe the course of the experimental decay curves.

Some previous authors reported that Schroeder decay functions present difficulties when fitting the straight lines [2, 5, 9, 10, 12]. One may argue that corrections can be applied to make Schroeder decay curves look like linear decays by carefully selecting the upper limit of the integration or by subtraction/elimination of background noise, even resorting to iterative processes to optimize the corrections. The breaking mechanisms of the iterative process are conventionally set according to linearity of the resulting curves. From the viewpoint of the model-based analysis, these previous approaches have applied an inappropriate (linear) model for analyzing Schroeder decay data. As Fig. 3.3 demonstrates, however, if the data contain decay processes more complicated than a single-decay process, those correcting

measures, implicitly assuming a straight-line decay function, will significantly destroy inherent decay characteristics in the data. More profound challenges are due to the fact that one cannot know prior to data analysis how many decay processes the data at hand contain, since in single-space rooms, multiple-slope decays may occur, whereas in coupled-volume systems, single-slope decays can often be found. The following section elaborates on an advanced method based on Bayesian inference, which will first determine how many slopes are in the data before using an appropriate model to analyze the data.

3.4 Bayesian Decay Analysis

The fact that the decay models in Eq. (3.3) describe Schroeder decay functions reasonably well, constitutes basic background information for the following discussion. The background information I encapsulates the fact that a number of models are competing with each other to describe the energy decay data reasonably well. In particular, the background information I restricts the number of models M to be finite, the models in set $\Phi = \{\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_M\}$ are exhaustive and exclusive, and only one model is expected to be true (or useful) to explain the data. In practice, architectural acousticians are often expecting single-, double-, or triple-slope energy decays. All these propositions constitute the background information I which will be explicitly used in the subsequent formulation of model-based Bayesian decay analysis.

Professor Manfred Schroeder already found in his first computer room simulation based on ray tracing that non-exponential decays can be found if the simulated room contains one absorptive wall while all the others are highly reflective [22]. Multiple-slope decays are of auditory significance because they are believed to meet two competing auditory perceptual attributes simultaneously, both clarity and reverberance [20, 23]. Investigations on perceptual attributes, audibility of different decay profiles, or perceptual effects of multiple-slope decays over single-slope decays are beyond the scope of the current chapter, but the analysis methods described in this chapter will provide interested researchers with quantitative tools for the further studies.

3.4.1 Two Levels of Bayesian Inference

Bayesian probabilistic inference encompasses both the parameter estimation and the model selection problems by extensive use of Bayes' theorem. Bayesian inference applied to solving parameter estimation problems is referred to as the first level of inference, while solving model selection problems is referred to as the second level of inference [21, 24]. The following discussion begins with the second level of inference, model selection. This is also the scientifically logical approach;

one should determine which of the competing models is appropriate before the relevant model parameters are inferred.

3.4.1.1 Model Selection: The Second Level of Inference

Among a set of competing models, increased decay orders will always improve curve fitting but will often generalize poorly [24]. To avoid over-parameterization, Bayesian model selection applies Bayes' theorem to one of the competing models $\mathbf{H}_s$ given the background information I and the data $\mathbf{D}$, containing Schroeder decay function of K elements, as

$$p(\mathbf{H}_s|\mathbf{D}) = \frac{p(\mathbf{H}_s)p(\mathbf{D}|\mathbf{H}_s)}{p(\mathbf{D})}, \tag{3.5}$$

while pushing any interest in model parameter values into the background of the current problem. In Eq. (3.5), $p(\mathbf{H}_s)$ represents the *prior probability* of the model $\mathbf{H}_s$ given the background information I as defined previously at the beginning of this section (Sect. 3.4), but the conditional as "given the background information I" as noted by $p(\mathbf{H}_s|I)$ [or the other quantities, $p(\mathbf{D}|\mathbf{H}_s, I)$, $p(\mathbf{H}_s|\mathbf{D}, I)$], is omitted throughout this chapter for simplicity. Quantity $p(\mathbf{D}|\mathbf{H}_s)$ is termed *marginal likelihood*, also termed *evidence*. Quantity $p(\mathbf{D})$ is a constant, and $p(\mathbf{H}_s|\mathbf{D})$ represents the *posterior probability* of the model $\mathbf{H}_s$, given the data $\mathbf{D}$.

Model comparison between two different models $\{\mathbf{H}_j, \mathbf{H}_k\}$ relies on the posterior ratio,

$$\frac{p(\mathbf{H}_j|\mathbf{D})}{p(\mathbf{H}_k|\mathbf{D})} = \frac{p(\mathbf{H}_j)}{p(\mathbf{H}_k)} \frac{p(\mathbf{D}|\mathbf{H}_j)}{p(\mathbf{D}|\mathbf{H}_k)}, 1 \leq j, k \leq M; j \neq k. \tag{3.6}$$

While model selection exhaustively calculates every posterior probability $p(\mathbf{H}_s|\mathbf{D})$ among the finite model set, to determine which model leads to the maximum posterior probability, it can also sequentially evaluate Eq. (3.6) to rank the competing models.

The first ratio on the right-hand side in Eq. (3.6), termed *prior ratio*, represents the degree of one's initial belief in how much the model $\mathbf{H}_j$ would have been preferred over the model $\mathbf{H}_k$ before the data were acquired, while the second ratio on the right-hand side of Eq. (3.6), termed *likelihood ratio*, measures how much the data prefer the model $\mathbf{H}_j$ over $\mathbf{H}_k$. The prior ratio opens up the opportunity, if desirable, to incorporate a prior preference to $\mathbf{H}_j$, but it is discouraged. Assigning equal prior probability

$$p(\mathbf{H}_s) = \frac{1}{M}, \quad 1 \leq s \leq M, \tag{3.7}$$

implies no subjective preference to any of models, which results in

$$\frac{p(\mathbf{H}_j|\mathbf{D})}{p(\mathbf{H}_k|\mathbf{D})} = \frac{p(\mathbf{D}|\mathbf{H}_j)}{p(\mathbf{D}|\mathbf{H}_k)}. \tag{3.8}$$

This indicates that the marginal likelihood (or evidence) $p(\mathbf{D}|\mathbf{H}_s)$ plays a central role in Bayesian model selection.

3.4.1.2 Parameter Estimation: The First Level of Inference

For the purpose of parameter estimation, the first level of Bayesian inference applies Bayes' theorem

$$p(\boldsymbol{\Theta}_s|\mathbf{D}, \mathbf{H}_s) = \frac{p(\boldsymbol{\Theta}_s|\mathbf{H}_s)p(\mathbf{D}|\boldsymbol{\Theta}_s, \mathbf{H}_s)}{p(\mathbf{D}|\mathbf{H}_s)}, \tag{3.9}$$

to the parameters $\boldsymbol{\Theta}_s$ containing $2 \cdot s + 1$ parameters and the data $\mathbf{D}$ once a specific model $\mathbf{H}_s(\boldsymbol{\Theta}_s)$ is determined by the model selection, where $p(\boldsymbol{\Theta}_s|\mathbf{H}_s)$ is the prior probability of parameters $\boldsymbol{\Theta}_s$, $p(\mathbf{D}|\mathbf{H}_s)$ is exactly the marginal likelihood in Eq. (3.5), and $p(\mathbf{D}|\boldsymbol{\Theta}_s, \mathbf{H}_s)$ is the likelihood function of the parameters, which is the probability of the residual error $\mathbf{e} = \mathbf{D} - \mathbf{H}_s$. Application of the principle of maximum entropy leads to an assignment of the likelihood function [25],

$$L(\boldsymbol{\Theta}_s) = p(\mathbf{D}|\boldsymbol{\Theta}_s, \mathbf{H}_s) \approx \frac{(2\pi E)^{-K/2}}{2}, \tag{3.10}$$

with

$$E = \frac{\mathbf{e}^{\mathrm{T}}\mathbf{e}}{2}, \tag{3.11}$$

where $(\cdot)^{\mathrm{T}}$ denotes matrix transpose.

The probability $p(\boldsymbol{\Theta}_s|\mathbf{D}, \mathbf{H}_s)$ on the left-hand side of Eq. (3.9) is the *posterior probability* of the parameters $\boldsymbol{\Theta}_s$, representing the updated knowledge about the parameters once the data become available. The integration over the entire parameter space $\int_{\boldsymbol{\Theta}_s}$ on both sides of Eq. (3.9), combined with the simplified notations $\pi(\boldsymbol{\Theta}_s) = p(\boldsymbol{\Theta}_s|\mathbf{H}_s)$ and $L(\boldsymbol{\Theta}_s) = p(\mathbf{D}|\boldsymbol{\Theta}_s, \mathbf{H}_s)$ in Eq. (3.10), yields

$$p(\mathbf{D}|\mathbf{H}_s) = Z_s = \int_{\boldsymbol{\Theta}_s} L(\boldsymbol{\Theta}_s)\pi(\boldsymbol{\Theta}_s)d\boldsymbol{\Theta}_s. \tag{3.12}$$

The quantity $p(\mathbf{D}|\mathbf{H}_s)$ in the denominator on the right-hand side of Eq. (3.9) is independent of the parameters $\boldsymbol{\Theta}_s$, and can be taken out of the integral. $p(\mathbf{D}|\mathbf{H}_s)$ determined by Eq. (3.12) is exactly the same as the marginal likelihood in Eqs. (3.5) and (3.8). It plays a central role in the model selection. Equation (3.9) can then be expressed as

$$\begin{gathered} p\left(\boldsymbol{\Theta}_s \middle| \mathbf{D}, \mathbf{H}_s\right) \times Z_s = \pi\left(\boldsymbol{\Theta}_s\right) \times L\left(\boldsymbol{\Theta}_s\right). \\ \text{posterior} \times \text{evidence} = \text{prior} \times \text{likelihood} \end{gathered} \tag{3.13}$$

The quantity $Z_s = p(\mathbf{D}|\mathbf{H}_s)$ is often referred to as (Bayesian) *evidence* for model $\mathbf{H}_s$ [24]. Equation (3.13) states the logical relations among the quantities of Bayesian inference [28]: prior probability $\pi(\boldsymbol{\Theta}_s)$ and the likelihood function $L(\boldsymbol{\Theta}_s)$ are the inputs, while the posterior probability $p(\boldsymbol{\Theta}_s|\mathbf{D}, \mathbf{H}_s)$ and the evidence Z_s are the outputs. Particularly, the posterior probability is the output for the first level of inference, the parameter estimation, while the evidence, Z_s, is the output for the second level of inference, the model selection. Bayesian evidence encapsulates the principle of parsimony, and quantitatively implements Occam's razor: *when two competing theories explain the data equally, the simpler one is preferred*. Using an asymptotic approximation, the next section elaborates on the quantitative implementation of Occam's razor.

3.5 Bayesian Information Criterion

The Bayesian information criterion (BIC) asymptotically approximates Bayesian evidence if a (multi-dimensional) Gaussian distribution can approximate the posterior probability distribution within a vicinity around the global extreme of the likelihood. Experimental results have shown that this is often the case for the energy decay analysis [16]. In this application, the (natural logarithmic) BIC for ranking a set of decay models $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \cdots$ is given by

$$\text{BIC} \approx 2 \cdot \ln\left[\widehat{L}\left(\widehat{\boldsymbol{\Theta}}_s\right)\right] - N_D \ln(K) \quad \text{[Neper]}, \tag{3.14}$$

with $N_D = 2 \cdot s + 1$ being the dimensionality, or the number of parameters involved in model $\mathbf{H}_s$. The quantity $\widehat{L}$ is the peak value of the likelihood whose location in the parameter space is denoted by $\widehat{\boldsymbol{\Theta}}_s$. The first term in Eq. (3.14) represents the degree of the model fit to the data, while the second term represents the penalty of over-parameterization. When increasing the decay model order, the first term may also increase, indicating a better fit to the data, but the BIC will dramatically decline because of the second term, which penalizes over-parameterization. It will outweigh the first term [16]. In the scope of energy decay analysis among a set of decay models $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \cdots$, the model yielding a significantly larger BIC value in Eq. (3.14) is the most concise model providing the best fit to the decay function

data, while at the same time capturing the important exponentially decaying features evident in the data.

Taking the experimentally measured data (PNR = 54 dB, as shown in Fig. 3.3), Fig. 3.4 illustrates the Bayesian analysis results and shows the decomposed decay terms in the second- and third-order Schroeder decay models. The first-order decay model represents a strong misrepresentation of the data and is not shown here. Figure 3.4b illustrates the typical example of an over-parameterized model which may or may not improve the curve fitting significantly in comparison with that of the second-order model as shown in Fig. 3.4a, but Occam's razor, quantitatively implemented in Bayesian analysis, penalizes the over-parameterized model since it receives a significantly lower BIC value. Its BIC value is 81 nepers lower than that of the second-order decay model (see Table 3.1, last column).

3.6 Sampling Methods

The BIC critically relies on estimation of the global maximum of the posterior distribution Eq. (3.14); it does not actually calculate the integral of the likelihood volume/mass over the prior distribution Eq. (3.12). When the posterior distribution cannot be approximated by (multi-dimensional) Gaussian distributions within the vicinity at its global peak, the BIC will represent a poor approximation of the evidence. This section outlines calculation of the evidence, Z_s. Nested sampling, among other advanced sampling methods, most recently proposed by Skilling [26] provides a cohesive method for estimating the evidence Z_s as a primary quantity while obtaining the posterior samples $p(\boldsymbol{\Theta}_s|\mathbf{D}, \mathbf{H}_s)$ as subsidiary values useful for parameter estimations during the process of calculating Z_s. This becomes plausible by inspecting Eqs. (3.9) and (3.12) since Z_s is calculated via the integral of $L(\boldsymbol{\Theta}_s)\pi(\boldsymbol{\Theta}_s)$ over the entire parameter space $\widehat{\boldsymbol{\Theta}}_s$; at the same time, decay parameter estimation solely relies on calculating the posterior $p(\boldsymbol{\Theta}_s|\mathbf{D}, \mathbf{H}_s)$ through $L(\boldsymbol{\Theta}_s)\pi(\boldsymbol{\Theta}_s)/Z_s$ [see Eq. (3.9)]. The decay parameters can then be estimated in terms of probabilistic moment calculations [25] [see Eq. (3.23) in Sect. 3.6.3] or in terms of the maximum *a posteriori* (MAP) approach using these subsidiary posterior samples [25, 29].

3.6.1 Basic Formulation

This section outlines the nested sampling applied to both Bayesian model selection and parameter estimation tasks. More complete treatments can be found elsewhere [26–29]. The following discussion will use simplified notation so that Eq. (3.12) becomes

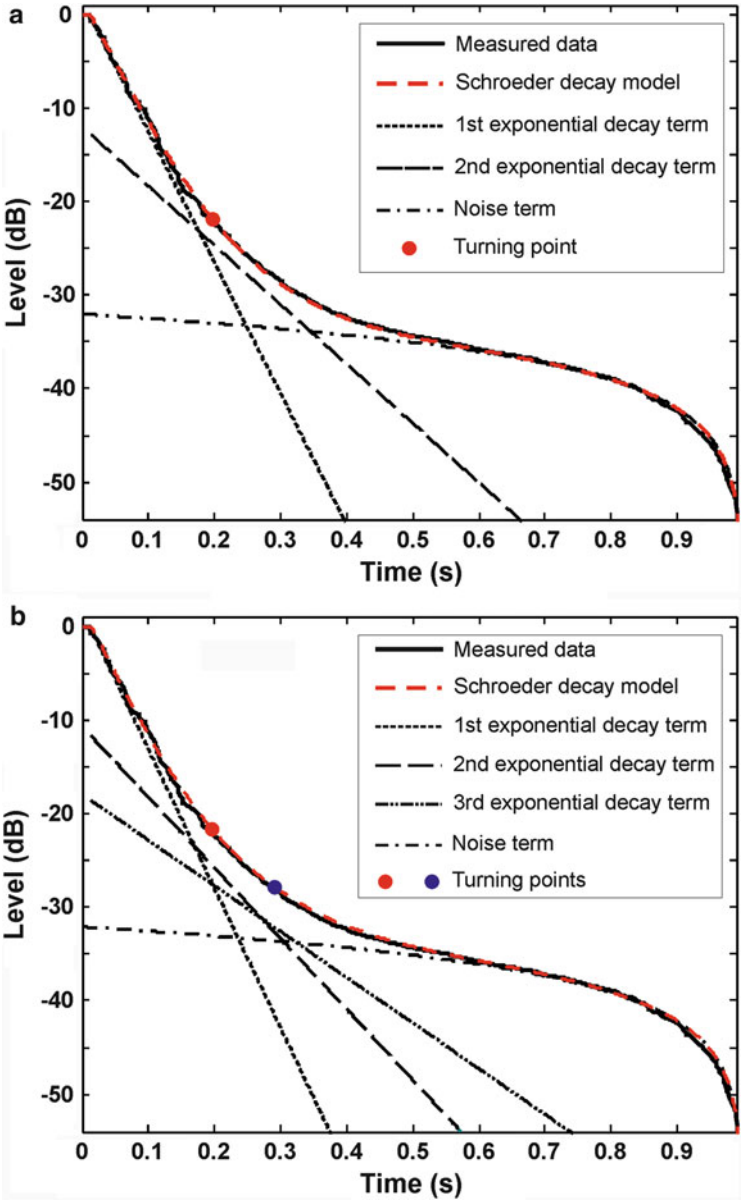


Fig. 3.4 Experimentally measured Schroeder decay curves as shown in Fig. 3.3 (the peak-to-noise ratio PNR = 54 dB). (**a**) Comparison with Bayesian model curve and decomposition using a double-slope model. (**b**) Comparison with Bayesian model curve and decomposition using a triple-slope model

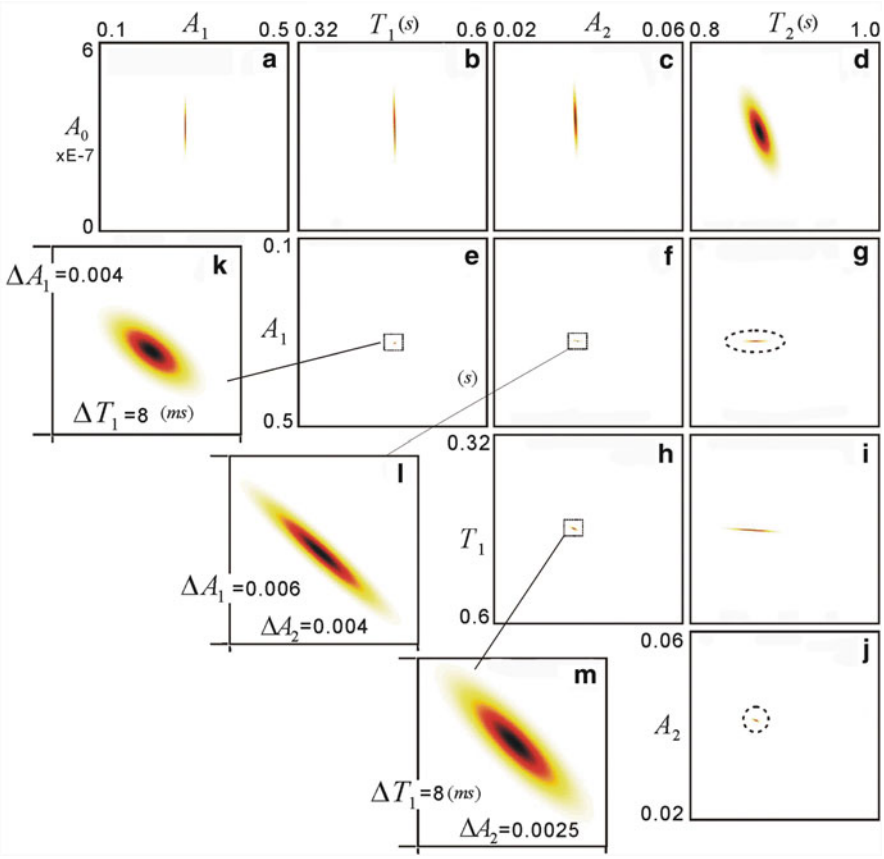


Fig. 3.5 Normalized (likelihood or) marginal posterior probability distributions (MPPDs) over two-dimensional (zoomed-in) parameter space from an experimentally measured data set in an acoustical scale model using the second-order decay model (double-slope model). (**a**) MPPD over $\{A_0, A_1\}$. (**b**) MPPD over $\{A_0, T_1\}$. (**c**) MPPD over $\{A_0, A_2\}$. (**d**) MPPD over $\{A_0, T_2\}$. (**e**) MPPD over $\{A_1, T_1\}$. (**f**) MPPD over $\{A_1, A_2\}$. (**g**) MPPD over $\{A_1, T_2\}$. (**h**) MPPD over $\{T_1, A_2\}$. (**i**) MPPD over $\{T_1, T_2\}$. (**j**) MPPD over $\{A_2, T_2\}$. A grid of 200×200 is used for (**a–d**), and (**k–m**), while a grid of 300×300 for (**e–j**)

$$Z = \int_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta})\pi(\boldsymbol{\Theta})d\boldsymbol{\Theta}. \tag{3.15}$$

The subscript s is dropped, but still bear in mind that the parameters $\boldsymbol{\Theta}$ and the evidence Z are associated with a specific model $\mathbf{H}_s$ given the data $\mathbf{D}$ and the background information I.

This integral over the parameter space will become intractable even over a small dimensionality of the parameter space. In order to highlight this challenge, Fig. 3.5 illustrates posterior probability distributions $[L(\boldsymbol{\Theta}_s)\pi(\boldsymbol{\Theta}_s)]$ over a five-dimensional

decay parameter space, often found in the decay analysis of room-acoustics practice. The figure shows one example experimentally measured in an acoustical scale model of coupled rooms. The measured room impulse response is first filtered at 1 kHz octave-band, then Schroeder-integrated to obtain the data vector **D**. In the following, the area/volume under the posterior probability distribution is termed *posterior mass*. The figure illustrates small two-dimensional subspaces of the five-dimensional parameter space while keeping the rest of other three parameters constant, being optimum-values determined from well-sampled estimates of the parameters. These two-dimensional distributions are referred to marginal posterior probability distributions (MPPDs). All the MPPDs as shown in Fig. 3.5 contain the concentrated posterior mass in order to show its shape and sharpness. In room-acoustics practice, prior knowledge of individual parameters is often vague. Without experimental data, decay times T_i can only be assumed to fall in a range of a few seconds (0.2 s–few seconds). Varied shapes and sizes of the posterior probability distribution, some extremely sharp, narrow, some very broad, already indicate that a direct integral over a few dimensions of relevant parameter space in terms of exhaustive sampling makes the problem hopelessly intractable.

The sampling method discussed in this section makes evaluating this integral a tractable task, Eq. (3.15) can be simply rewritten [26] as

$$Z = \int_0^1 L(\mu) d\mu, \tag{3.16}$$

where $d\mu(\boldsymbol{\Theta}) = \pi(\boldsymbol{\Theta}) d(\boldsymbol{\Theta})$ is the elementary hyper-volume under the prior probability distribution, with

$$\mu = \int \pi(\boldsymbol{\Theta}) d(\boldsymbol{\Theta}), \tag{3.17}$$

μ is termed *prior mass*. Figure 3.6 illustrates the concept of the likelihood function over the prior mass. Since it is a monotonic function, $L(\mu)$ is just the inverse of $\mu(L)$. A decreasing sequence of prior mass yields

$$1 = \mu_0 > \mu_1 > \cdots > \mu_i > \mu_{i+1} > \mu_\infty = 0, \tag{3.18}$$

when an increasing sequence of likelihood values

$$0 \approx L_{\min} = L_0 < L_1 < \cdots < L_i < L_{i+1} < \cdots < L_{\max} < \infty \tag{3.19}$$

can be determined. This is elaborated on in the following subsection. The elementary prior mass $d\mu$ is associated with the likelihood within $[L_i, L_{i+1}]$.

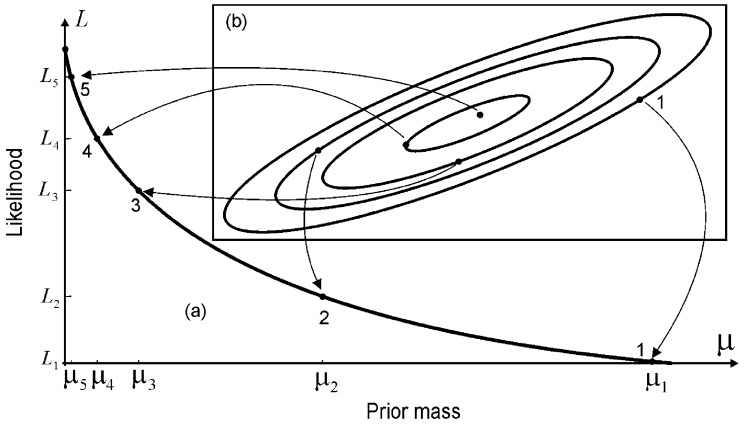


Fig. 3.6 Estimation of the Bayesian evidence by the nested sampling. (**a**) The area under the likelihood function curve over the prior mass between 0.0 and 1.0 represents the Bayesian evidence as defined in Eq. (3.16). A decreasing sequence of the prior mass along μ-axis corresponds to an increasing sequence of the likelihood values along L-axis. (b) The likelihood contours corresponding to different likelihood values

3.6.2 Numerical Implementation

In order to implement the above integration for Z in Eq. (3.16), one should start backwards from Eq. (3.19) to Eq. (3.15) by defining the prior mass sequence Eq. (3.18) based on constraints on likelihood values Eq. (3.19). The integral is processed sequentially by ordered elementary prior mass $d\mu$ confined to likelihood values within $[L_i, L_{i+1}]$. Each prior mass value μ_i corresponds to the likelihood value $L_i = L(\mu_i)$. In relating each L_i with its outward interval $[\mu_i, \mu_{i-1})$, the evidence can be approximated according to Eq. (3.16) by

$$Z \approx \sum_{i=1}^{\infty} L_i \Delta\mu_i, \quad \text{with} \quad \Delta\mu_i = \mu_{i-1} - \mu_i, \tag{3.20}$$

where $\mu_0 = 1$, $\mu_\infty = 0$, and $L_\infty = L_{\max} < \infty$.

Numerical implementation starts drawing a finite number N of random samples $[\boldsymbol{\Theta}^{(1)}, \cdots, \boldsymbol{\Theta}^{(N)}]$ from the prior probability $\pi(\boldsymbol{\Theta})$ with very low likelihood values. This is most easily achieved if outer bounds of the parameter ranges can be roughly specified *a priori*. The prior mass, according to Eq. (3.18), then corresponds to $\mu_0 \approx 1$. Among N samples as illustrated in Fig. 3.7, the next step is to identify and record the sample associated with the lowest likelihood value, denoted by $[\boldsymbol{\Theta}^{(1)}, L_1]$. The replacement of this lowest value by another new sample constrained by $L(\boldsymbol{\Theta}) > L_1$ keeps the total number of the samples to be N. At the ith iteration, a new sample associated with $[\boldsymbol{\Theta}^{(i)}, L_i]$ is recorded and replaced by the next new one constrained by $L(\boldsymbol{\Theta}) > L_i$.

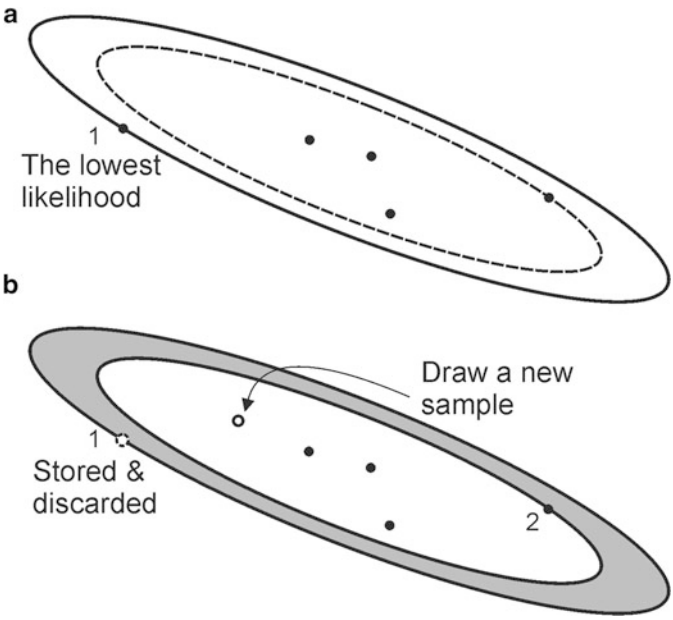


Fig. 3.7 Nested sampling algorithm using five random, uniformly distributed samples ($N = 5$) over the likelihood contours of a two-dimensional parameter space at each iteration. (**a**) The sample associated with the lowest likelihood value (L_1) is identified. (**b**) The sample (denoted by a *dashed circle*) associated with the lowest likelihood value is recorded and discarded. A new sample (denoted by a *circle*) is drawn constrained within the likelihood contour $> L_1$

Figure 3.7 conceptually illustrates this sampling procedure with $N = 5$. In Fig. 3.7a, among $N(=5)$ sampled likelihood values, the lowest one labeled by "1" is identified. In Fig. 3.7b this sample, labeled by a dotted circle, is recorded and discarded, upon which a new sample is drawn from the prior distribution constrained upon the condition $L_{\text{new}} > L_1$. After adding this new sample to the remaining $(N - 1)$ samples, another sample point is then identified to have the lowest likelihood value among the new N samples, labeled by "2" in Fig. 3.7b. In Fig. 3.7b the shaded area between two likelihood contours, when projecting onto the prior distribution, corresponds to the elementary prior mass $\Delta\mu_1$. It can be shown [26] that $\Delta\mu_i$ in Eq. (3.20) can be approximated statistically as

$$\Delta\mu_i \approx \left(\frac{1}{N+1}\right)\left(\frac{N}{N+1}\right)^{i-1}. \tag{3.21}$$

Accumulating Eq. (3.20) to infinity can be interrupted either by specifying termination after R iterations or by specifying a convergence criterion.

3.6.3 Posterior Samples for Parameter Estimation

While accumulating the evidence Z in Eq. (3.20), samples $\boldsymbol{\Theta}^{(i)}$ associated with the likelihood value L_i are recorded in R objects $[\boldsymbol{\Theta}^{(1)}, L_1]$, $[\boldsymbol{\Theta}^{(2)}, L_2], \ldots, [\boldsymbol{\Theta}^{(R)}, L_R]$. Posterior sample values are only of interest if the proper model preferred by the data has been selected via evidence estimation. According to Eqs. (3.13) and (3.21), the posterior samples are readily determined by

$$p_i\left(\boldsymbol{\Theta}\middle|\mathbf{D},\mathbf{H}\right) = \mu_i L_i/Z \approx \left(\frac{N}{N+1}\right)^i L_i/Z, \tag{3.22}$$

given the selected model without any further effort to sample them. For numerical convenience, Eq. (3.22) is evaluated logarithmically.

Probabilistic moment calculations [30] yield the mean values of the parameters $\langle\boldsymbol{\Theta}\rangle$ by

$$\langle\boldsymbol{\Theta}\rangle = \frac{1}{R}\sum_{i=1}^{R}\boldsymbol{\Theta}^{(i)}. \tag{3.23}$$

and bearing in mind that $\langle\boldsymbol{\Theta}\rangle$, given the model $\mathbf{H}_s$ of s slopes contains $2\cdot s+1$ parameters $\langle\boldsymbol{\Theta}\rangle = \{\langle\theta_1\rangle, \ldots, \langle\theta_{2s+1}\rangle\}$ [see Eq. (3.3) with $\mathbf{A}_s$ and $\mathbf{T}_s$]. The expected element at jth row, kth column of the covariance matrix $\langle\mathbf{C}\rangle = [\langle C_{jk}\rangle]$ can be estimated by

$$\langle C_{jk}\rangle = \frac{1}{R}\sum_{i=1}^{R}\left(\theta_j^{(i)} - \langle\theta_j\rangle\right)\left(\theta_k^{(i)} - \langle\theta_k\rangle\right), 1 \le j,k \le 2\cdot s+1, \tag{3.24}$$

respectively, where each $\langle\theta_k\rangle$ is estimated by Eq. (3.23). The covariance matrix $\langle\mathbf{C}\rangle$ is a square matrix of dimension $2\cdot s+1$. The individual variance σ_j^2 and corresponding standard deviation σ_j of jth decay parameter can be estimated from the diagonal elements of the co-variance matrix $\langle\mathbf{C}\rangle$ as discussed in [30]. The inter-relationship between decay parameters is measured by the cross-correlation coefficient

$$ccc(j,k) = \langle C_{jk}\rangle / \sqrt{\langle C_{jj}\rangle\langle C_{kk}\rangle}. \tag{3.25}$$

If the mean values of decay parameters cannot meet the required accuracy based on the available posterior samples, the mean values in Eq. (3.23) and the covariance matrix in Eq. (3.24) estimated so far can guide a further effort to refined sampling, using e.g., importance sampling [25], where the proposal distribution, essential to the importance sampling, can be more accurately assigned.

3.7 Discussion

As an analysis method for quantifying energy decay characteristics, the methods reviewed in this chapter are model-based methods, which critically depend on the model involved. No decay characteristics can be inferred without an appropriate parametric model. Note that linear least-squares fitting is essentially a model-based method as well. The reason that correcting measures must be applied in previous analysis [2, 5, 9, 12] is that the linear model misrepresents Schroeder decay functions. Even Bayesian decay analysis can also provide misleading estimates if incorrect models are used. In this specific application, the decay order (number of slopes) must be determined before using the estimated parameters to explain the data. Fortunately, the Bayesian framework also embodies decay model selection, and quantitatively implements Occam's razor [24]. One should apply Bayesian model selection to a proper model set to determine which model among the competing models is appropriate before interpreting the decay parameters.

The model-based Bayesian decay analysis discussed in this chapter can be applied to both reverberation time estimation and to quantification of multiple-slope decay characteristics. Efficient sampling methods within the framework of Bayesian inference, particularly dedicated to multiple-slope decays, are still the on-going research of the author's team. Perceptual aspects associated with multiple-slope decays represent another potentially promising field of future research. Efficient implementation and successful applications of the model-based Bayesian decay analysis discussed in this chapter can benefit the future investigations since the quantification of acoustical conditions is inevitable in systematic psychoacoustic investigations.

3.8 Conclusions

Sound energy decay analysis in room-acoustics is of fundamental relevance and Schroeder integration is at its foundation. This chapter is mainly dedicated to sound energy decay analysis using the model-based Bayesian inference method. The parametric model derived from Schroeder integration is used for the task, and at the same time, it is also helpful to understand the behavior of Schroeder decay functions. In room-acoustics practice, Schroeder decay models of different orders are often competing with each other to explain or predict the data. The calculation of Bayesian evidence is then of central importance for both decay model selection and decay parameter estimation problems, more specifically in room-acoustic applications where reverberation times or decay characteristics in a room under investigation need to be quantified. This chapter outlines how the nested sampling method can be applied cohesively within the Bayesian framework to obtain both energy decay order selection and the decay parameter estimation.

Acknowledgments The author would like to dedicate this work to the memory of Manfred Robert Schroeder whose great interest in the author's work on room-acoustic energy decay analysis has inspired him to continue this endeavor. The author is thankful to Tomislav Jasa and Jonathan Botts for their insightful discussions and for implementing the nested sampling for this work, to Jean-Dominique Polack, Dejan Ciric, and Brian Katz for their critical comments on an early version of this chapter, and to Cameron Fackler and Paul Luizard for proofreading the final version of the chapter.

References

1. Schroeder, M.R.: New method of measuring reverberation time. J. Acoust. Soc. Am. **37**, 409–412 (1965)
2. Kuerer, R., Kurze, U.: Integrationsverfahren zur Nachhallauswertung. Acustica **19**, 314–322 (1967/1968)
3. Kuttruff, H., Jusofie, M.: Messungen des Nachhallverlaufs in meheren Räumen ausgefuehrt nach dem Verfahren der integrieten Impulseantworten (Energy decay measurements in rooms using the integrated impusle-response method). Acustica **11**, 1–9 (1969)
4. Bodlund, K.: On the use of the integrated impulse response method for laboratory reverberation measurements. J. Sound Vib. **56**, 341–362 (1978)
5. Chu, W.T.: Comparison of reverberation measurements using Schroeder's impulse method and decay-curve averaging method. J. Acoust. Soc. Am. **63**, 1444–1450 (1978)
6. Luizard, P., Katz, B.F.: Coupled volume multi-slope room impulse responses: a quantitative analysis method. In: Proceedings of IoA, Dublin (2011)
7. Cremer, L., Müller, H.A., Schultz, T.J.: Principles and Applications of Room Acoustics. Applied Science Publishers, London (1982)
8. Xiang, N., Jing, Y., Bockman, A.: Investigation of acoustically coupled enclosures using a diffusion equation model. J. Acoust. Soc. Am. **126**, 1187–1198 (2009)
9. Vorländer, M., Bietz, H.: Comparison of methods for measuring reverberation time. Acustica **80**, 205–215 (1994)
10. Hirata, Y.A.: A method of eliminating noise in power responses. J. Sound Vib. **82**, 593–595 (1982)
11. Xiang, N.: Two approaches to evaluation of reverberation times from Schroeders decay curves. In: Proceedings of 15th ICA, Trondheim, IV 175–178 (1995)
12. Lundeby, A., Vigran, T.E., Bietz, H., Vorländer, M.: Uncertainties of measurements in room acoustics. Acustica **81**, 344–355 (1995)
13. Schroeder, M.R.: Complementarity of sound buildup and decay. J. Acoust. Soc. Am. **40**, 549–551 (1966)
14. Xiang, N.: Evaluation of reverberation times using a nonlinear regression approach. J. Acoust. Soc. Am. **98**, 2112–2121 (1995)
15. Faiget, L., Legros, C., Ruiz, R.: Optimaization of the impulse response length: application to noisy and highly reverberant rooms. J. Audio Eng. Soc. **46**, 741–750 (1998)
16. Xiang, N., Goggans, P., Jasa, T., Robinson, P.: Bayesian characterization of multipleslope sound energy decays in coupled-volume systems. J. Acoust. Soc. Am. **129**, 741–752 (2011)
17. Xiang, N., Ahnert, W., Feistel, R.: A nonlinear regression approach to the evaluation of reverberation times using Schroeder's integrated impulse response method (A). J. Acoust. Soc. Am. **94**, 1846 (1993)
18. Xiang, N., Ahnert, W., Feistel, R.: Nachhallauswertung mittels nicht-linearer Regression (Reverberation time evaluation using nonlinear regression). In: Proceedings of Fortschritte der Akustik, DAGA93, pp. 255–258 (1993)
19. Xiang, N., Vorländer, M.: Using iterative regression for estimating reverberation times in two coupled spaces (A). J. Acoust. Soc. Am. **104**, 1763 (1998)

20. Xiang, N., Goggans, P.M.: Evaluation of decay times in coupled spaces: Bayesian parameter estimation. J. Acoust. Soc. Am. **110**, 1415–1424 (2001)
21. Xiang, N., Goggans, P.M.: Evaluation of decay times in coupled spaces: Bayesian decay model selection. J. Acoust. Soc. Am. **113**, 2685–2697 (2003)
22. Schroeder, M.R.: Digital simulation of sound transmission in reverberant spaces. J. Acoust. Soc. Am. **47**, 424–431 (1970)
23. Jaffe, J.C.: Innovative approaches to the design of symphony halls. Acoust. Sci. Tech. **26**, 240–243 (2005)
24. MacKay, D.: Information Theory, Inference and Learning Algorithms, Chapters 28, 29. Cambridge University Press, Cambridge (2002)
25. Jasa, T., Xiang, N.: Efficient estimation of decay parameters in acoustically coupled spaces using slice sampling. J. Acoust. Soc. Am. **126**, 1269–1279 (2009)
26. Skilling, J.: Nested sampling. In: Fisher, R. et al. (eds.) Bayesian Inference and Maximum Entropy Methods in Science and Engineering, American Institute of Physics, vol. 735, pp. 395–405 (2004)
27. Jasa, T., Xiang, N.: Using nested sampling in the analysis of multi-rate sound energy decay in acoustically coupled rooms. In: Knutz, K. et al. (eds.) Bayesian Inference and Maximum Entropy Methods in Science and Engineering, American Institute of Physics, vol. 803, pp. 189–196 (2005)
28. Sivia, D., Skilling, J.: Bayesian Data Analysis—A Bayesian Tutorial, 2nd edn. Oxford University Press, New York (2006)
29. Jasa, T., Xiang, N.: Nested sampling applied in the Bayesian room-acoustics decay analysis. J. Acoust. Soc. Am. **132**, 3251–3262 (2012)
30. Xiang, N., Goggans, P.M., Jasa, T., Kleiner, M.: Evaluation of decay times in coupled spaces: reliability analysis of Bayeisan decay time estimation. J. Acoust. Soc. Am. **117**, 3707–3715 (2005)

***

Biography



Prof. Dr. **Ning Xiang**, Director, Graduate Program in Architectural Acoustics, Rensselaer Polytechnic Institute, New York, USA. He obtained his Ph.D. (Dr.- Ing) in 1990 at BochumRuhr-University, Germany. He is a Fellow of the Acoustical Society of America, a Fellow of the Institute of Acoustics, UK, Associate Editor of the Journal of the Acoustical Society of America (JASA), and a book-series Editor with J. Ross Publishing. Inspired by Professor Manfred Schroeder, he has been working on applications of number-theoretic sequences since 1988. Among his long list of journal publications, he and Professor Manfred Schroeder have coauthored one JASA paper for simultaneous dual source measurements in 2003, a measurement technique being critical in

outdoor sound propagation research. Since 1993 one of his research interests has been in room-acoustic energy decay analysis. Since 1999 he, along with his colleagues and students, has applied Bayesian inference in Schroeder's energy decay analysis. He is a recipient of Wallace Clement Sabine Medal from the Acoustical Society of America in 2014. [Photo: with Professor Manfred Schroeder in 2001, during the time he demonstrated initial success using Bayesian inference for advanced room-acoustics decay analysis.]

Chapter 4
Estimating the Crossover Time Within Room Impulse Responses

Guillaume Defrance and Jean-Dominique Polack

Abstract In 1954, M.R. Schroeder found that above a transition frequency, modes overlap and can even be superposed. This particular frequency has been called the Schroeder frequency. In the time domain, there also exists a time after which the density of sound rays, that have been emitted into a room, becomes homogeneous and isotropic. The purpose of the present chapter is to illustrate similar findings in the time domain, namely a domain where reflections are singled out, shortly after the direct sound, followed at longer times by superposed reflections that follow a different distribution in time. In previous publications, the transition time between the two regimes was called mixing time. Here, using two different approaches, it is experimentally shown that the estimated time is inconsistent with the theory of dynamical systems. Therefore, a more generic name is proposed for the transition from early reflections to late reverberation, the crossover time.

4.1 Introduction

In 1954, M.R. Schroeder experimentally investigated the distribution of eigenfrequencies measured in a room [1]. While, in 1912, Weyl had theoretically shown that this distribution is a quadratic function of the frequencies [2], Schroeder observed that above a certain frequency, the density of eigenmodes is apparently constant. In other words, he found that above a transition frequency, modes overlap and can even be superposed. This particular frequency has been called the Schroeder frequency. At low frequency, eigenmodes are discrete and are perceptively important since they participate to the sound coloration of the room. Above the Schroeder

G. Defrance (✉)
Laboratory of Signals and Systems, Dept. Electromagnetism, SUPELEC, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France
e-mail: defrance.all@gmail.com

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_4,
© Springer International Publishing Switzerland 2015

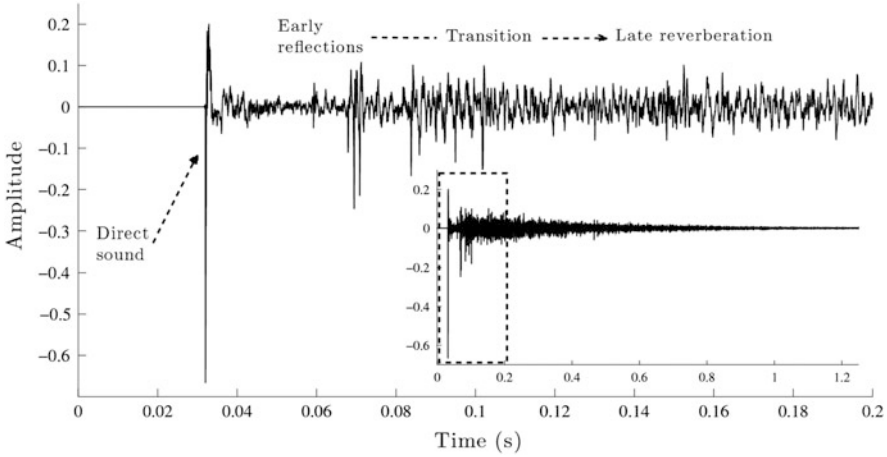


Fig. 4.1 Early times of an experimental RIR (encapsuled graphic), measured in Salle Pleyel

frequency, eigenmodes are statistically distributed [1]. This frequency depends on the reverberation time and the volume of the hall, and is given by:

$$f_{\mathrm{Sch}} = \sqrt{\frac{\pi c_0^3 T_{\mathrm{R}}}{4\pi V \ln(10)}} \tag{4.1}$$

$$\approx 2000\sqrt{\frac{T_{\mathrm{R}}}{V}} \tag{4.2}$$

where f_{Sch} is in Hz, T_{R} is the reverberation time of the room in s, c_0 is the speed of sound in m s^{-1}, and V is the volume of the hall in m^3.

In the time domain, there also exists a time after which the density of sound rays, that have been emitted into a room, becomes homogeneous and isotropic. We call that time the *crossover time*. This particular time stands for the transition between two regimes of room impulse responses: early reflections and late reverberation, also called the diffuse sound field. While at early times, reflections are discrete, the late reverberation is statistically distributed (i.e., superposition and overlap of sound rays) [3] (Fig. 4.1).

Therefore, the knowledge of the transition time can be particularly interesting when synthesizing room impulse responses (RIRs) since the computation of room impulse responses can be stopped at this time and synthesized by an exponentially decaying filtered Gaussian noise.

In the following, we shortly review statistical room acoustics in order to introduce the transition time in detail. Then, we show how to estimate the transition time within experimental RIRs using two different signal processing tools: Matching Pursuit and the eXtensible Fourier Transform. After comparing our results, we finally discuss the relationship between the transition time and dynamical properties

of large rooms. In particular, we focus on the first name given to the crossover time, that is, the *mixing time*. By reviewing the meaning and definition of mixing and ergodicity, we justify the name of crossover time and explain its difference with the mixing time.

4.2 A Brief Review of Statistical Room Acoustics

A standard way to document the acoustics of a room is to measure a set of RIRs. The RIR is built by the superposition of *arrivals* [4], that is, modified versions of an original pulse emitted by the source and reaching the receiver after travelling through the room. Sound emitted by the source undergoes scattering and absorption (i.e., filtering) when encountering boundaries of the room. The source pulse is therefore divided in many wavelets that each follows a different *trajectory* within the hall. When these trajectories are closed, that is, when sound rays stick to the same path, eigenmodes are created.[1] RIRs are composed of all these trajectories with due consideration for their respective amplitude. In room acoustics, the time distribution of these arrivals plays an important role since it is directly linked to the acoustical quality of the room, as is demonstrated in numerous books [3, 5].

Since Weyl's path-breaking paper on eigenfrequency density in bounded rooms [2], it is well known that this density increases with the square of the frequency. However, when experimentally measuring this density in a box, Schroeder found out in 1954 [1] that it is not the case at high frequencies: the finite width of the modes, created by the losses, leads to a superposition of many modes at any frequency and to a constant density of peaks. Schroeder carried out further his investigation in order to determine the transition frequency between the two regimes, and introduced what is now called the Schroeder frequency, linked to volume and decay time, i.e., to mode width.

Weyl's estimation of eigenfrequency density was based on the approximation of the general solution to the Helmholtz equation in a bounded space by the Green function of free space. Thus, the notion of trajectory was implicitly introduced in mode counting, albeit in a crude way as represented by the Green function that links a source position to a receiver position. In Weyl's free space approximation, no reflections on the boundaries were taken into account, but later refinements to his theory took reflections into account [6].

As a consequence, Weyl's theory and its successors naturally induce researchers to look at the temporal succession of arrivals from trajectories linking the source to the receiver and taking into account reflections at the boundaries.

At short times, arrival density is not constant, but increases with time. A simple theory, derived for rectangular rooms [7], teaches us that the density $D_e(t)$ Eq. (4.3) is proportional to the square of the time elapsed since the sound was emitted by the

[1] In the semi-classical approach, we talk about periodic orbits.

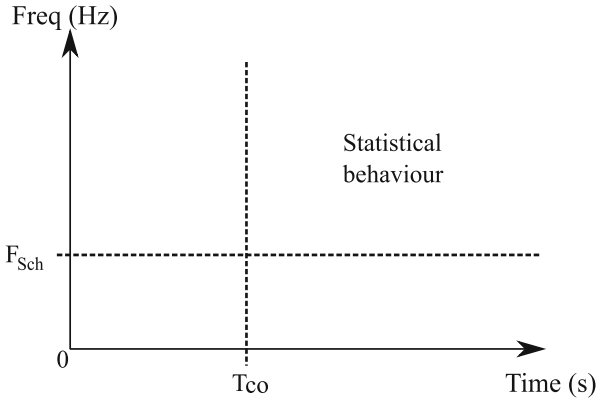


Fig. 4.2 Schematical representation of the Schroeder frequency and the crossover time

source (which is different from the time laps between the arrival and the direct sound):

$$D_e(t) = \frac{dN}{dt} = \frac{4\pi c_0^3}{V} t^2 \tag{4.3}$$

$$N(t) = \frac{4\pi V}{3c_0^3} t^3 \tag{4.4}$$

where $D_e(t)$ is in number of arrivals per second, N is the number of arrivals up to time t, c_0 is the speed of sound in m s^{-1}, and V is the volume of the room in m^3.

The present chapter investigates whether a transition time can experimentally be demonstrated between two domains: an initial time domain where many arrivals are discrete, and a late time domain where arrivals are superposed and therefore become statistically equidistributed, just as they are in the frequency domain (Fig. 4.2). Recall that the crossover time is measured from the time of emission of the impulse by the sound source.

4.3 Estimating the Crossover Time

4.3.1 Introduction

The first idea that comes to mind is to estimate the transition time by detecting the time at which the distribution of the RIR tends to be Gaussian. This idea arises from the report that if the spectrum of an RIR, measured in a large room, is Gaussian above the Schroeder frequency, then its inverse Fourier transform also provides a Gaussian distribution. However, the different statistical tests that one could use

(Kolmogorov–Smirnov, Lilliefors, etc.) are very dependent on the number of samples comprised within the analysis window, on the one hand, and on the threshold set for each test, on the other hand [8–10].

Therefore, estimating the crossover time must be carried out differently. Using the algorithm of Matching Pursuit (MP) [11], we show how the transition time can be estimated within RIRs by studying the temporal distribution of arrivals [12]. In the time–frequency domain, it is also possible to estimate the transition time as the time at which the Fourier coefficients of the spectrum of the RIR tends toward a Gaussian distribution. This is achieved using the eXtensible Fourier Transform [13].

4.3.2 Matching Pursuit

Assuming the hall to be a system of time invariant linear impulse responses, or a bunch of filters with delay, the source is expected to be filtered and translated in time along the RIR. Therefore, supposing a high correlation between the RIR and the pulses emitted by the source (i.e., the direct sound), with due consideration to the filtering of the room, MP appears to be well suited for this purpose. Matching Pursuit provides a sparse decomposition of a signal, that is, it decomposes any signal into a linear expansion of waveforms that belong to a dictionary. These waveforms are selected iteratively in order to best match the signal structures. Although Matching Pursuit is nonlinear [11], it maintains an energy conservation which guarantees its convergence. In practice the number of iterations must be finite, leading to only an approximate decomposition of the signal. In these experiments, the dictionary of waveforms (or atoms) is limited to the direct sound itself, translated in time. The determination of its exact temporal boundaries is of importance for a perfect match with the RIR, but is achieved within MP itself, as explored in [14].

Matching Pursuit can help understanding more deeply the architecture of an RIR since this algorithm introduced by [11] provides information, which can be seen as maxima of correlation Eq. (4.6) [15] between two signals, the RIR (x) and the direct sound (the mother-atom).

Matching Pursuit works as follows:

1. Initialization: $m = 0$, $x_m = x_0 = x$.
2. Computation of the correlations between the signal x_m and every atom γ of a dictionary φ using inner products:

$$\forall \gamma \in \varphi : \mathrm{CORR}(x_m, \gamma) = \left| \langle x_m, \gamma \rangle \right| \tag{4.5}$$

   The dictionary φ is a set of atoms γ, of the same length than x, constituted by the direct sound and translated in time, by step of one sample.
3. Search the most correlated atom, by searching for the maximum inner product:

$$\widetilde{\gamma}_m = \text{argmax}\{|\langle x_m, \gamma\rangle|\}_{\gamma\in\varphi} \tag{4.6}$$

4. Substracting the corresponding weighted atom $\widetilde{\alpha}_m\gamma_m$ from the signal x_m:

$$x_{m+1} = x_m - \alpha_m\widetilde{\gamma}_m \tag{4.7}$$

$$x_{\text{R}}^{(m)} = \sum\nolimits_{k\leq m}\alpha_k \cdot \widetilde{\gamma}_k \tag{4.8}$$

where $\alpha_m = \langle x_m, \widetilde{\gamma}_m\rangle$

5. (a) Stops if the desired level of accuracy is reached: $R = x_{m+1}$
   (b) Otherwise, re-iterate the pursuit to step 2:

$$m \leftarrow m + 1.$$

where x is the RIR, R the residual, γ the atom (here, the direct sound), φ the dictionary of atoms γ, and $x_{\text{R}}^{(m)}$ the reconstructed signal.

In theory, any signal x can be perfectly decomposed in a set of atoms for an infinity of iterations. In practice, this number must be finite and a stopping criterium has to be set. The authors propose to use the signal/residual ratio (*SRR*) in dB of the norm L2 of x over the norm L2 of the residual (R) defined as:

$$SRR = 10\log_{10}\left(\frac{\|x_2\|}{\|R_2\|}\right) \tag{4.9}$$

The quality of the decomposition of x in atoms depends on the value of *SRR*, that is, the stopping criterium. On the one hand, for a too low *SRR*, the residual has an energy level too high, and the rebuilt signal $x_{\text{R}}^{(m)}$ is an impoverished approximation of x. On the other hand, a too high *SRR* leads to a high number of iterations. In that case, it is not necessary to perform more iterations beyond a certain threshold of accuracy. In [12], we show that the best value of *SRR* is found when the mean differences of room acoustical indices (RT_{30}, RT_{20}, *EDT*, C_{80}, etc.) [16] between synthesized and original RIRs is below 5 %.

If the source impulse is unknown, it is possible to recover it from recorded RIRs running MP. The temporal boundaries of the direct sound that are thought to be the best correspond to the lowest number of iterations ran by MP onto experimental RIRs, with a stopping criterium of $SRR = 20$ dB to reach. For further details, please refer to [12, 14].

A key point of this method is the compensation of the energy decay of RIRs. Indeed, the probability to detect arrivals is directly linked to the local energy of the signal. As this latter decreases exponentially, one can expect the probability of detection to decrease exponentially too. Therefore, energy compensation, by making the signal stationary, ensures equal weight to all parts of the RIR, thus equiprobability of detecting arrivals. Note that the energy compensation is achieved

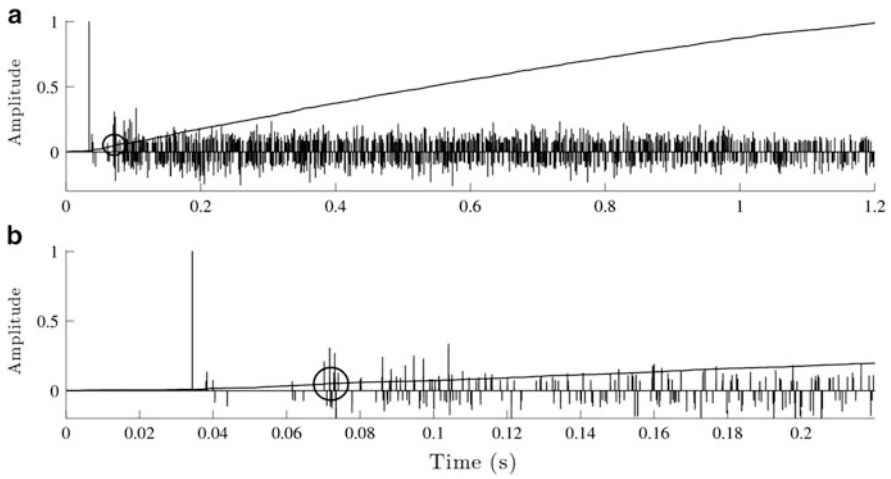


Fig. 4.3 (**a**) Crossover time (*circle*) detected on a *CDF* (*black bold line*) for an experimental RIR and the set of estimated arrivals (*grey*) (*SRR* = 5 dB). Plot (**b**) details the beginning of plot (**a**). Note the energy compensation applied to the signal

by applying an inverse exponential, whose argument is proportional to the reverberation time and to the mean absorption [3].

Matching Pursuit run on RIR provides a linear set of coefficients which can be seen as a temporal distribution of arrivals of the RIR, providing the knowledge of the cumulative distribution function (*CDF*) of arrivals (Fig. 4.3).

According to the theory of room acoustics, arrivals overlap for times greater than the transition time. We observe that MP detects only one arrival instead of two when the time delay between them is inferior or equal to the equivalent duration [17] of the source impulse (i.e., the direct sound). Therefore, the crossover time T_{co} is estimated by looking for:

$$T_{\mathrm{co}} = \arg\min\left(t_{i+1} - t_i \leq \widetilde{d}_{\mathrm{Source}}\right) \tag{4.10}$$

where t_i is the time of occurrence of the ith arrival and $\widetilde{d}_{\mathrm{source}}$ is the equivalent duration of the source impulse.

At early times ($t < T_{\mathrm{co}}$), a few arrivals are detected (Fig. 4.4). A cubic fit of this part of CDF curves permits to verify that the number of arrivals is a function of t^3, as seen in Eq. (4.4). The transition time marks the time at which CDF of estimated arrivals increases at a constant rate (linear part), which is coherent with the definition of late reverberation. In other words, for times greater than T_{co}, the RIR can be defined as behaving stochastically, rather than deterministically. In a large room like a concert halls, we observe that the evolution of the number of arrivals as a function of time (for $t \geq T_{\mathrm{co}}$) is almost constant all over the room [12].

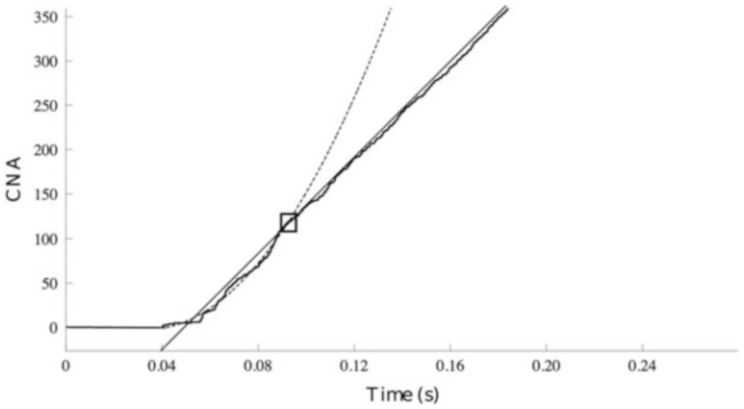


Fig. 4.4 Detail of a cubic ($t < T_{co}$) and a linear ($t > T_{co}$) fits made on a Cumulative Number of Arrivals (*CNA*). *Plain bold line*: *CNA*. *Dashed bold line*: cubic fit. *Dashed line*: linear fit. *Square*: crossover time

Remark If applying MP to RIRs is straightforward, this method presents disadvantages. The first outcome is the use of the dictionary of atoms φ, which is only constituted of the direct sound translated in time. The filtering of the room is thus not taken into account. Therefore, at each iteration of the algorithm, some spurious waveforms are created (due to Eq. (4.7)) that can be estimated as arrivals. This behavior of MP has been deeply studied by Sturm et al. [18].

4.3.3 eXtensible Fourier Transform

The eXtensible Fourier Transform [13], or XFT, is comparable to a memory process, that is, the time of occurrence of events (such as arrivals within RIRs) can be detected compared to what happened in the past (e.g., time of occurrence of the direct sound).

The XFT consists in computing the Discrete Fourier Transform of a signal using a window that grows from a given starting time (Fig. 4.5). For a discrete signal $x(n)$ of length N samples, its XFT is defined as:

$$X(p,f) = \int_{n=0}^{N-1} w\left(\frac{n}{N}\frac{M}{p}\right) x(n) e^{-2j\pi nf/N} \tag{4.11}$$

with $w(t)$ the analysis window, which is a continuous function of time:

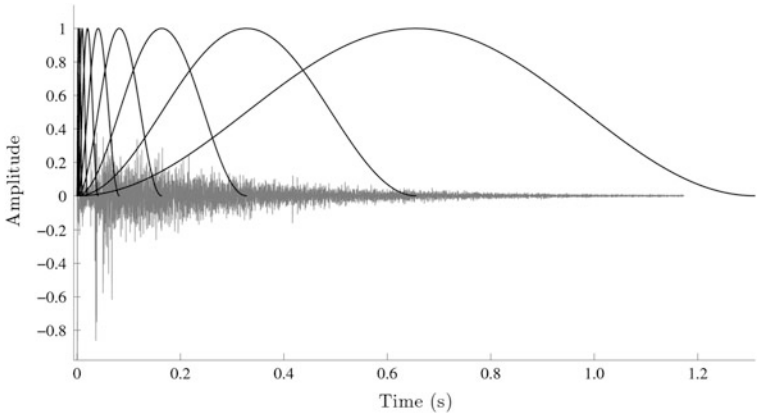


Fig. 4.5 Scheme of the principle of the XFT. For each window, a Discrete Fourier Transform of the signal x is computed. In that case, the window w is a Hann window, which starts from the time of arrival of the direct sound. The latter is detected as described in [19]

$$\mathrm{supp}(w(t)) = [0, 1], \tag{4.12}$$

where $X(p,f)$ is the complex matrix of the XFT, $x(n)$ a signal of N samples length, w the analysis window that grows with a constant rate while p increases (the first window has a width L_0), f is the discrete frequency in Hertz ($f \in Z$), M is the maximum number of windows w that the signal x can admit ($M = N/L_0$), p is the number of windows ($p = 1, 2, \ldots, M$), and $\mathrm{supp}(w(t))$ is the support of the function $w(t)$.

Equation (4.11) can be written differently:

$$X(p,f) = \int_{m=1}^{p} \int_{n=0}^{L_0-1} w\left(\frac{(m-1)L_0 + n}{pL_0}\right) x(n) e^{-2j\pi fn/N}. \tag{4.13}$$

Equation (4.13) emphasizes the fact that the XFT is a sum in the time domain of spectral components of the signal. Furthermore, the XFT can be seen as a DFT computed at different times of the signal but keeping the phase shift information.

The XFT is calculated for the same total number of samples (using zero padding) in order to keep the same nominal frequency resolution for each window. The number of samples (N_{DFT}) for computing each Discrete Fourier Transform is calculated as the highest power of 2: $N_{\mathrm{DFT}} = 2^{\lceil (\log(N)/\log(2)) \rceil}$, where $\lceil \cdot \rceil$ is the ceiling symbol and N is the number of samples of the signal x.

Interpreting the magnitude of the XFT is not straightforward, although it is related to Schroeder's work. Indeed, it permits to visualize the progressive establishment of modes within the room at a given position. Therefore, we concentrate on the phase of the XFT, noted $\Phi(p,f)$.

The time evolution of $\Phi(p,f)$ can be understood as follows. As long as $t_{k+1} - t_k \gg 0$ (i.e., p is small), the function $unwrap(\Phi(p,f))$ is approximately a linear function of frequency f, which is more or less robust. While for larger p ($\lim t_{k+1} - t_k \to 0$), if the number of arrivals is large enough, then the coefficients of the Fourier series expansion become independent and normally distributed [20, 21]. The XFT does not permit one to claim if a signal is normally distributed, but it does allow the visualization of the evolution of the system toward a random behavior. The time evolution of the XFT phase embodies the changing statistics of a signal. We propose calculating the sum of the differences between the unwrapped phases in the first window ($p = 1$) and successive ones ($p = 2, 3, 4, \ldots M$). The phase differences are calculated at each frequency f as follows:

$$\delta\Phi(p) = \int_{f=0}^{N-1} \text{unwrap}[< X(1,f)] - \text{unwrap}[< X(p,f)]. \tag{4.14}$$

Equation (4.14) can be seen as an estimate of the distance between the phases of $X(1,f)$ and $X(p,f)$. Therefore, the smaller the quantity $\delta\Phi(p)$, the more similar are the phase spectra. Overall, Eq. (4.14) is an estimate of the time of occurrence of arrivals, regarding the time origin given by the first window ($p = 1$).[2]

Figure 4.6 shows several graphs of the unwrapped phase of the XFT computed on an experimental RIR. As expected [3], the phase wraps faster between $\pm\pi$ with time, denoting a gradual changing of behavior in the signal, due for instance to an increasing number of events (such as arrivals and diffusion). This confirms findings from earlier investigations [12, 22, 23]. Estimating the crossover time is thus equivalent to estimating the time at which an important change of behavior occurs in the phases.

As seen in Figs. 4.7 and 4.8, $\delta\Phi(p)$ is an increasing function of time, up to a certain time, and comprises one main inflexion point. The latter is a breaking point in the statistics of the signal, and separates two states of behavior of the impulse response. The first state is deterministic, that is, arrivals can be discriminated, and their time of occurrence can be predicted (Sect. 4.2). The function that best fits this part of the curve $\delta\Phi(p)$ is a cubic function of time (Fig. 4.7), which confirms Eq. (4.4). The second state of behavior of the curve $\delta\Phi(p)$, which has a constant slope until a certain time, denotes a statistical behavior: arrivals are statistically distributed in time.

For a proper estimate of the crossover time, the first window must comprise the direct sound. The detection of the onset of RIRs and the estimation of the direct sound duration can be found in [14, 19]. Furthermore, as in Sect. 4.3.2, the energy decay of the RIR needs to be compensated in order to give the same weight to all arrivals within the signal, especially those at large times. In that case, the

[2] One may link this to [12, 22] and Sect. 4.2, in which the statistics of RIRs are investigated by looking at the cross-correlations in the time domain between the direct sound and the rest of the signal.

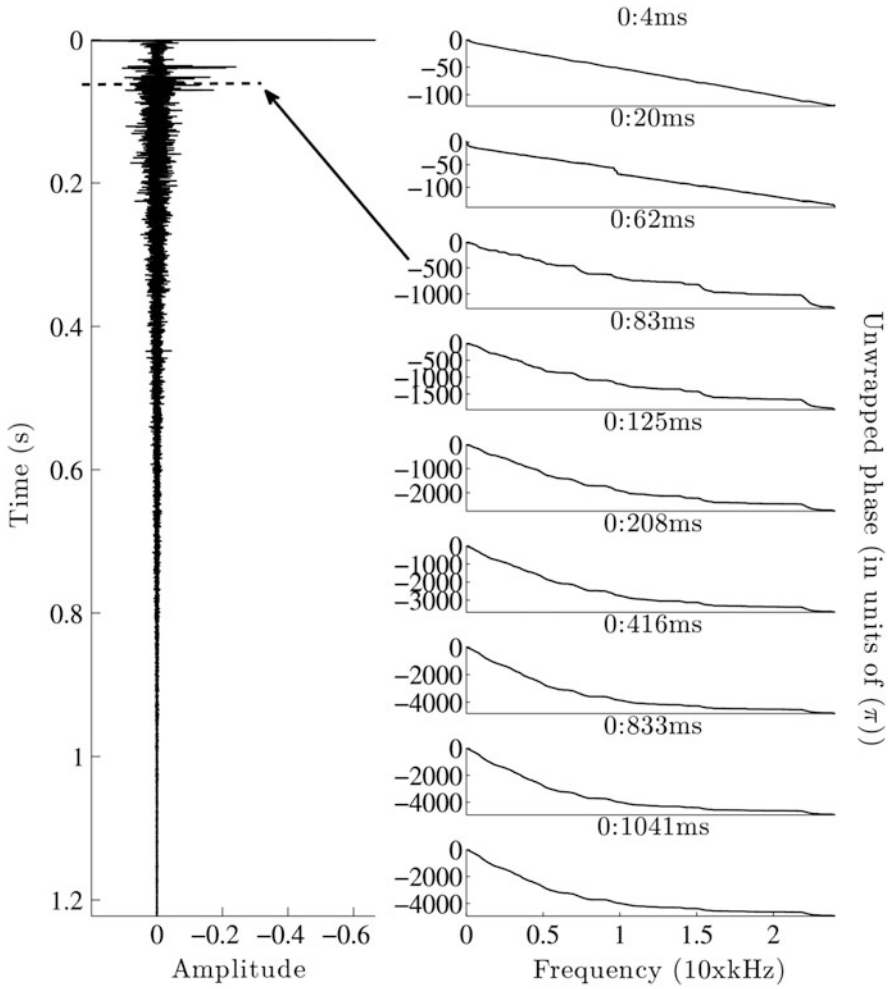


Fig. 4.6 Visualization of the unwrapped phases of the XFT computed on a RIR measured in Salle Pleyel (sound source: pistol shot). Note that the compensation of the energy decay of the RIR is applied, but not shown for a better readability. Furthermore, the time of propagation between the source and the receiver positions is not shown here. First window width: $L_0 = 128$ samples. The estimated crossover time is materialized by a *dashed line* superposed to the RIR

compensation of energy decay leads to use a rectangular window w Eq. (4.11) in order to avoid any other modifications of the amplitude of the signal x.

We propose to estimate the crossover time by searching the time of occurrence of the maximum of the curve $\delta^2\Phi(p)$, which is defined as:

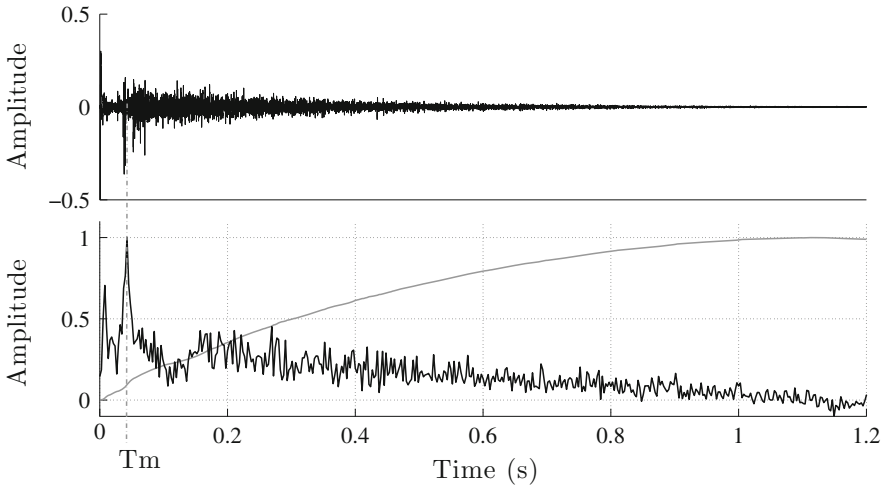


Fig. 4.7 *Grey*—experimental RIR (the energy decay compensation is applied, but not shown for a better readability). *Plain black line*—curve $\delta\Phi(p)$ Eq. (4.4) (normalized). *Black bold dashed line*—cubic fit. Note that $\delta\Phi$ is over sampled for a better readability. The graphic included shows the details of the selected region within the *dashed rectangle*

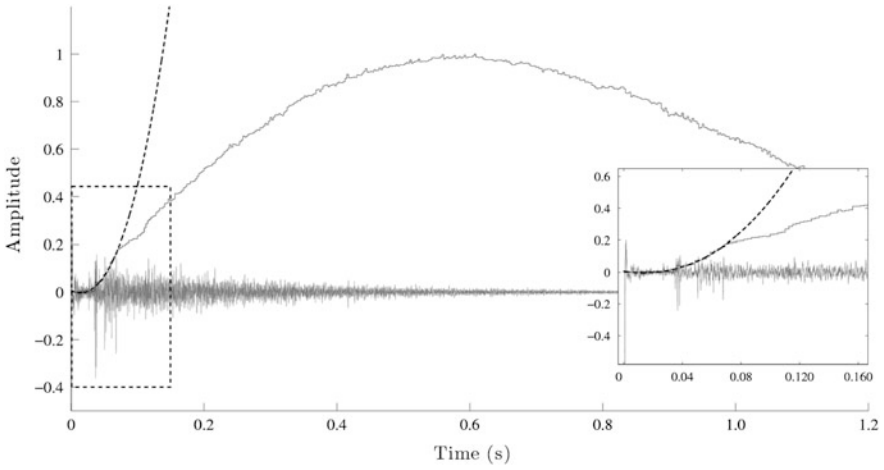


Fig. 4.8 *Top*—an experimental RIR (the energy decay compensation is applied, but not shown for a better readability). *Bottom*—Eq. (4.14) in *grey*; Eq. (4.15) in *black*. The breaking point (i.e., the crossover time) in the statistics of the RIR is materialized by a *dashed line*. Note that bottom graphic is over sampled for a better readability. Notice the other breaking points after the crossover time (other peaks). The time of propagation between the source and the receiver positions is not shown here. On *bottom graphic*, curves are normalized for an easier readability

$$\delta^2\Phi(p) = \delta\Phi(p+1) - \delta\Phi(p) \tag{4.15}$$

$$T_{\mathrm{co}} = \mathrm{argmax}_p\left(\delta^2\Phi(p)\right) * L_0/f_{\mathrm{S}} \tag{4.16}$$

where $p = 1, 2, \ldots, M-1$, L_0 is the width of the first window w of analysis in samples, f_{S} is the sampling frequency in Hz, and T_{co} is the crossover time in second.

4.3.4 Comparing MP and XFT

In this section, we present crossover times estimated using MP and XFT[3] on experimental RIRs measured in Salle Pleyel (Paris, France) [25]. Salle Pleyel is a concert hall of 19,000 m^3 and 1,900 seats. We carried out a set of 21 measurements using pistol shots as sound sources and an omnidirectional microphone. Figure 4.9 shows crossover times estimated using MP and XFT.

For both estimators, transition times are a function of the source/receiver distance: the slope of each linear regression is proportional to the speed of sound. If several values estimated with MP and the XFT are the same, most of them are different. These differences can be explained as follows.

When using MP, transition times are a function of the stopping criterion (Sect. 4.3.2). The higher the SRR, the larger the number of estimated arrivals, and thus the larger the probability to detect the transition times at early times. Moreover, as the dictionary of MP is only composed of the direct sound of each RIR, artifacts of estimation are created at each iteration. Therefore, the number of arrivals can be overestimated. Finally, for MP and the XFT, the estimation of the direct sound can be erroneous when it is immediately followed by an arrival. And when using the XFT, the estimation of transition times also depends on the window width of analysis.

An interesting result is the values of transition times when the source and the receiver positions are the same. In such a case, values returned by the XFT predict that the time needed in order to reach the diffuse sound field is 0.044 s, that is, the mean free path of Salle Pleyel ($\widetilde{\ell} = 14\,\mathrm{m}$). The linear regression performed on the values given by MP predicts approximately half of the mean free path. It is difficult to conclude about this report. However, it seems realistic to assume that the diffuse sound field can only be established once sound rays have travelled the mean distance between two reflections within the hall.

[3] Both estimators, MP and XFT, have been validated on thousands of synthetic RIRs [12, 13], created by a stochastic model [24].

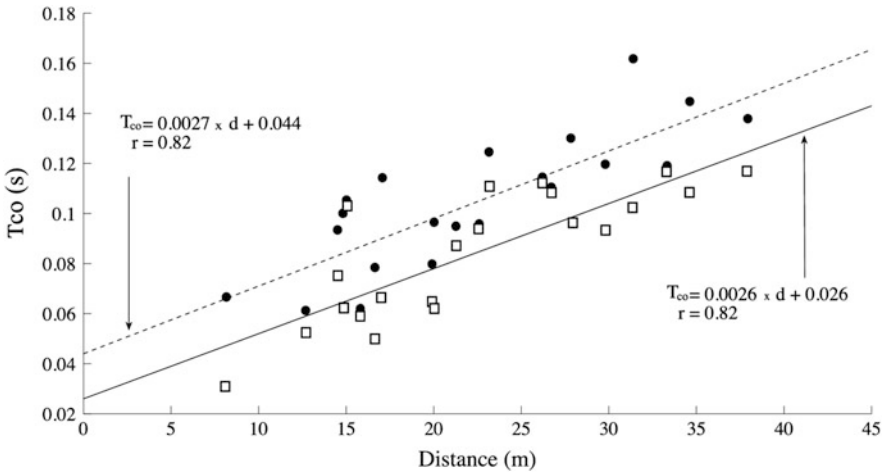


Fig. 4.9 Crossover times estimated within experimental RIRs. *Circles*: estimated with XFT. *Stars*: estimated with MP

4.4 Discussion Around the Crossover Time and the Mixing Time

The introduction of ergodic dynamical systems, made by Joyce [26], was able to justify Schroeder's spectral observations in the frequency domain. Joyce assumed large halls to be ergodic and mixing. In fact, he theoretically showed that the expression of the reverberation time given by Sabine is recovered if the room is mixing—thus, ergodic—and if absorption is weak and uniformly distributed within the space [26]. Since the publication of that article, mixing has been assimilated to late reverberation. In particular, the transition time from early reflections to the diffuse sound field was called the *mixing time* in the literature of room acoustics.

After briefly reviewing ergodicity and mixing, we discuss our methods and results regarding the theory of dynamical systems. We finally propose to use the general term *crossover time* instead of mixing time.

4.4.1 Ergodicity and Mixing

The hypothesis of ergodicity was first introduced by Boltzmann in 1871 for his kinetic theory of gases. But it is only in the 1930s that it has been theoretically proven [27]. The validity of the equidistribution of energy relies on the hypothesis of ergodicity. In fact, this hypothesis is validated a posteriori based on its relevant predictions, but it is still an hypothesis. Considering an ergodic system (at the thermodynamical equilibrium), all the states of the phase space have the same probability to be occupied. This means that the energy is equidistributed. In the

theory of room acoustics, when the energy is equidistributed, that is, after a certain number of reflections, the diffuse sound field is fully established within the hall.

Referring to Krylov's theory of mixing [28], we learn that most dynamical systems gradually loose memory of their history with time. Thus, trajectories become independent of their origin, so that the probability of reaching any phase point, at any time, along the trajectory becomes equivalent (ergodicity). In such a case, the arrival density, at any receiver position, becomes constant and independent of the receiver position. The mixing character of a room is associated to the (exponential) divergence of sound rays emitted by a sound source, that were initially adjacents when emitted. Divergence of trajectories are created by obstacles (seats, balconies, etc.) or even by asperities of walls. Once the system is mixed, sound rays density is homogeneous and isotropic.[4]

Shannon's theory of information states that if all the states of a system are equiprobables, then its entropy is maximum [29]. The entropy is related to the information contained within the system. When the entropy is the highest, the system has lost all its initial information, that is, there is no memory of its initial state. In room acoustics, the loss of information is created by the numerous filtering that sound rays undergo on their trajectory (reflections on obstacles, dissipation, etc.). According to Shannon's theory, as long as the entropy of a system is not maximum, the system keeps evolving until it reaches its final state. For instance, a mixing system provides a maximum entropy.

It is worthwhile to recall that ergodicity has been experimentally proven only for a few systems (like Sinai's billiard or the Bunimovich stadium [30, 31]) but never for large halls, except using numerical simulations [32–34]. This explains why some can doubt about the relevance of this hypothesis within the context of room acoustics [35].

In fact, the *mixing time* is a naive concept. Indeed, a mixing system tends progressively (i.e., asymptotically) toward equilibrium, but does not suddenly reach this state. The hypothesis of ergodic rooms is useful to room acousticians since it allows one to assume that the energy is uniformly distributed within the hall after a sufficient number of reflections [26, 35, 36]. If experimental observations show that the rate of decay of the sound energy is statistically the same within the hall [37, 38], none of the studies prove that the energy is uniformly distributed within the phase space. Moreover, this equipartition of energy, that we assume in room acoustics, is only valid up to a certain distance of the boundaries of the hall, that is, where the migration of energy cannot be observed.

Several definitions of the mixing time can be found in the literature. The first one comes from the theory of dynamical systems, the second one can be found in the room acoustics literature (i.e., that transition time between early reflections and the diffuse sound field), and the third one is related to psychoacoustics. The latter has been given by Polack [39] based on Cremer's work [3]. Polack proposed to define the mixing time such as the time at which ten arrivals overlap within a window of 24 ms. Therefore, using $N = 10$ and $\Delta t = 24$ ms in Eq. (4.4), we obtain a heuristical formulation of the mixing time that is now well known:

[4] Note that a mixing system is ergodic but the inverse is not necessarily true.

$$T_{\mathrm{M}} \approx \sqrt{V} \tag{4.17}$$

where T_{M} is expressed in ms and V is the volume in m^3.

The existence of these three definitions makes sometimes difficult the comprehension of the concept of mixing and mixing time. To estimate the mixing time according to its dynamical definition, one must experimentally observe the time evolution of the system in the phase space. While this approach requires to follow, within the room, the divergence of trajectories of sound particles, the two other definitions focus on a physical and a perceptual analysis of the system at a given position within the hall. It is thus obvious that the dynamical and the perceptual definitions are not related since mixing, in the perceptual sense, has to be understood as the inability to discriminate more than ten events in a given period of time. In this work, we do not discuss this alternative definition of mixing, especially because the comparison between the time distribution of arrivals estimated by MP and Eq. (4.17) is inconclusive [8].

4.4.2 Why the Crossover Time Cannot be Called the Mixing Time

Recent work have experimentally attempted to estimate the mixing time within room impulse responses [8–10, 12, 13]. While tests of Gaussianity, Matching Pursuit or the eXtensible Fourier Transform point out several phenomena such as the progressive apparition of diffusion, the growth of entropy, and the change of distribution of arrivals, the hypothesis of ergodicity and mixing is still not verified.

Running MP on RIRs measured in Salle Pleyel, we show that, after the crossover time, the total number of sound rays received (until the reverberation time) is not the same according to the receiver position in the hall. Nevertheless, the time evolution of this number is approximately constant in the hall (1,140 arrivals/s ± 7 %) [8, 12]. If this result suggests that the energy is uniformly distributed over the room, one cannot conclude that the system is mixing. But this result depends on the stopping criterion used for estimating arrivals within RIRs using MP (Sect. 4.3.2). Indeed, the higher the stopping criterion, the larger the number of estimated arrivals. Later measurements of RIRs carried out in Salle Pleyel with a B-format microphone have shown that there exists some privileged directions of propagation of sound rays at high frequencies. This is in contradiction with the assumed equipartition of energy.

On the other hand, the XFT and the phase difference are closer to the dynamical definition of mixing than MP. Indeed, the XFT is an estimate of the loss of memory of the system, that is, a progressive fall of correlation of the system regarding its initial state, in agreement with Krylov's and Shannon's theories. Moreover, as detailed in [13], the use of the XFT allows one to visualize when the distribution of a RIR tends to become Gaussian. This is to relate to Schroeder's work and his crossover frequency, but this has nothing to do with the dynamical definition of mixing.

Applying either MP or the XFT to experimental RIRs (Sect. 4.3.4), we find an obvious relationship between the transition time and the source/receiver distance, as well as with the sound source. However, the theory of dynamical systems clearly states the contrary [40]: mixing occurs approximately at the same time within the system, and is not a function of the source excitation. Moreover, as none of these methods do consider the trajectories of sound particles within the space, we cannot conclude to the equivalence between late reverberation and the dynamical definition of mixing. Therefore, the estimated transition time cannot be called the mixing time. We propose instead to call it the *crossover time*.

Proposition The crossover time defines the time above which arrivals are statistically distributed at a given position in the hall. In other words, the crossover time indicates when the sound field is locally diffuse, but does not mean it is diffuse elsewhere within the hall.[5]

Finally, in the hypothesis of a mixing hall, the mixing time should be reached later than the crossover time. In addition to develop a system for measuring the divergence of sound particles, the question is now to know whether RIRs measured in concert halls last enough time to estimate the dynamical properties of such systems.

References

1. Schroeder, M.R.: Statistical parameters of the frequency response curves of large rooms. J. Acoust. Eng. Soc. **35**(5), 307–316 (1987)
2. Weyl, H.: Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen. Math. Ann. **71**, 441–479 (1912)
3. Cremer, L., Muller, H.A., Schultz, T.J.: Principles and Applications of Room Acoustics, vol. 1. Applied Science Publishers Ltd., London (1982)
4. Polack, J.-D.: Modifying chambers to play billiards: the foundations of reverberation theory. Acustica **76**, 257–272 (1992)
5. Beranek, L.: Concert and Opera Halls How They Sound. Acoustical Society of America, New York (1996)
6. Balian, R., Bloch, C.: Distribution of eigenfrequencies for the wave equation in a finite domain: I. Three-dimensional problem with smooth boundary surfaces. Ann. Phys. **60**, 401–447 (1970)
7. Bolt, R.H., Doak, P.E., Westervelt, P.J.: Pulse statistics analysis of room acoustics. J. Acoust. Soc. Am. **22**(3), 328–340 (1950)
8. Defrance, G.: Caractérisation du mélange dans les réponses impulsionnelles de salles. Application à la détermination expérimentale du temps de mélange (Characterization of mixing within room impulse responses. Application to the experimental estimation of the mixing time). Ph.D. thesis, Université Pierre et Marie Curie (UPMC—Paris VI) (2009)
9. Stewart, R., Sandler, M.: Statistical measures of early reflections of room impulse responses. In: Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-07), Bordeaux, France, pp. 59–62 (2007)

[5] Here, the term *diffuse* means that sound rays come with the same probability from any region of the space.

10. Defrance, G., Polack, J.-D.: Measuring mixing time in auditoria. J. Acoust. Soc. Am. **123**(5), 3499 (2008)
11. Mallat, S., Zhang, Z.: Matching pursuit with time-frequency dictionnaries. IEEE Trans. Signal Process. **40**(12), 3397–3415 (1993)
12. Defrance, G., Daudet, L., Polack, J.-D.: Using matching pursuit for estimating mixing time within room impulse responses. Acta Acust. Acust. **95**(6), 1082–1092 (2009)
13. Defrance, G., Polack, J.-D.: Estimating the mixing time of concert halls using the eXtensible Fourier Transform. Appl. Acoust. **71**, 777–792 (2010)
14. Defrance, G., Daudetm, L., Polack, J.-D.: Characterizing sound sources for room acoustical measurements. In: Proceedings of the Institute of Acoustics, vol. 30 (2008)
15. Krstulovic, S., Gribonval, R.: Mptk: matching pursuit made tractable. In: ICASSP'06, Toulouse, France (2006)
16. ISO 3382: Acoustics-measurements of the reverberation time of rooms with reference to other acoustical parameters (1997)
17. Bendat, J.S., Piersol, A.G.: Random Data: Analysis and Measurements Procedures. Wiley, New York (1971)
18. Sturm, B.L., Shynk, J.J., Daudet, L., Roads, C.: Dark energy in sparse atomic estimations. IEEE Trans. Audio Speech Lang. Processing **16**(3), 671–676 (2008)
19. Defrance, G., Daudet, L., Polack, J.-D.: Finding the onset of a room impulse response: straightforward? J. Acoust. Soc. Am. **124**, EL248 (2008)
20. Hurwitz Jr., H., Kac, M.: Statistical analysis of certain types of random functions. Ann. Math. Stat. **15**(2), 173–181 (1944)
21. Rice, S.O.: Mathematical analysis of random noise. Bell Syst. Tech. J. **23**, 13–29 (1944)
22. Defrance, G., Daudet, L., Polack, J.-D.: Detecting arrivals within room impulse responses using matching pursuit. In: Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08), Espoo, Finland (2008)
23. Schodder, G.R.: Uber Die Verteilung Der Energiereicheren Schallruckwurfe In Salen. Acustica **6**, 445–465 (1956)
24. Polack, J.-D.: Playing billiards in the concert hall: the Mathematical Foundations of Geometrical Room Acoustics. Appl. Acoust. **38**, 235–244 (1993)
25. Defrance, G., Polack, J.-D., Katz, B.-F.G.: Measurements in the new Salle Pleyel. In: Proc. Int. Symp. Room Ac., Sevilla, Sept 2007
26. Joyce, W.B.: Sabine's reverberation time and ergodic auditoriums. J. Acoust. Soc. Am. **58**(3), 643–655 (1975)
27. Birkhoff, G.D.: Proof of the Ergodic theorem. Proc. Natl. Acad. Sci. **17**, 656–660 (1931)
28. Krylov, N.S.: Works on Foundations of Statistical Physics. Princeton University Press, Princeton, NJ (1979) (translated by A.B. Migdal and Ya. G. Sinai and Yu. L. Zeeman)
29. Shannon, C.E.: The mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 (1948) and 623–656
30. Sinai, Y.G.: Dynamical systems with elastic reflections. Russ. Math. Surv. **25**, 141 (1972)
31. Bunimovich, L.A.: Conditions of stochasticity of two-dimensional billiards. Chaos **1**(2), 187–193 (1991)
32. Mortessagne, F.: Dynamique et interférences géométriques dans les billards chaotiques. Ph.D. thesis, Univ. Paris 7, Denis Diderot (1994)
33. Kawabe, T., Aono, K., Shin-ya, M.: Acoustic ray chaos and billiard system in Hamiltonian formalism. J. Acoust. Soc. Am. **113**(2), 701–704 (2003)
34. Koyanagi, S., Nakano, T., Kawabe, T.: Application of Hamiltonian of ray motion to room acoustics. J. Acoust. Soc. Am. **124**(2), 719–722 (2008)
35. Lee, J.B.: Why concert halls are not ergodic? J. Acoust. Soc. Am. **85**(6), 2680–2681 (1989)
36. Sabine, W.C.: Collected Papers on Acoustics. Harvard University Press, Cambidge (1964) (reprinted by Dover, New York)
37. Kuttruff, H.: Room Acoustics, 3rd edn. Elsevier Applied Science, New York (1991)
38. Penna, P.: Some statistical properties of reverberation process in diffuse sound fields. M.S. thesis, The acoustics laboratory, Denmark tekniske hojskole (1993)

39. Polack, J.-D.: La transmission de l'énergie sonore dans les salles. Ph.D. thesis, Thèse de doctorat d'Etat, Université du Maine, Le Mans, France (1988)
40. Katok, A., Hasselblatt, B.: Introduction to the modern theory of dynamical systems. Cambridge University Press, Cambridge (1996)

***

Biography



Guillaume Defrance received the French equivalent of a B.S. degree in Physical Measurements and in Mechanical Engineering from Université du Maine, Le Mans in 2003 and 2004, respectively, and finally the M.S. in Acoustics and Physics and Ph.D. degrees (Electrical and Mechanical Engineering) at Université Pierre et Marie Curie (UPMC), Paris, France, in 2006 and 2009, respectively. Dr. Defrance specializes in statistical room acoustics, late reverberation and diffuse sound field, stochastic and random processes, using algorithms for sparse approximation and time–frequency signal processing tools. In 2010, Dr. Defrance was a Marie Curie Research Fellow at the University of Sheffield (UK) where he carried out a research project on the diffuseness of urban sound fields. In 2011, he worked on the spatial repartition of energy at low frequencies in reverberant chambers at UPMC. As a research fellow at Echosens (Paris), he currently works on the characterization of disordered media using ultrasounds. He is currently a research fellow in the department of electromagnetism of the Laboratory of Signals and Systems at SUPELEC.



Dr. **Jean-Dominique Polack** received his Ph.D. in 1882 in Göttingen. He joined CNRS as Research Fellow in 1983, then was elected Professor of Acoustics at University Pierre et Marie Curie in Paris in 1997. From 1999 to 2001, he was appointed Professor of Electroacoustics at the Technical University of Denmark. From 2002 to 2006, he was Head of the Laboratoire d'Acoustique Musicale, now merged into Institut d'Alembert. He is presently in charge of the Graduate School at UPMC. Dr. Polack's areas of expertise cover loudspeaker models, room-acoustic measurements, ergodic theory of sound fields, and sound quality of soundscapes.

Chapter 5
Are Impulse Responses Gaussian Noises?

Jean-Dominique Polack

Abstract Many years ago, Manfred Schroeder proved that transfer functions in rooms follow complex Gaussian distributions. This property was extended to impulse responses by the author, following a suggestion by Moorer for simulating impulse responses. More recently, several authors have checked again the later property with modern signal analysis tools. They obtained mixed success, with results that strongly depend on the length of the analysis window. In order to understand these unstable results, the present paper takes a closer look at the statistical distributions of both impulse responses and transfer functions. It shows that an accurate model of both the impulse response and the transfer function is necessary in order to test the distributions. Further, it presents several sources of artefacts that skew the distributions, and show that they can be ascribed to the signal processing methods used to extract the responses. Finally, the conditions for obtaining true Gaussian distributions are specified.

5.1 Introduction

In 1954, Manfred Schroeder published two seminal papers [1, 2] on sound distribution in rooms. In the first paper, in refutation of the accepted theory of the time, he was able to show that resonance frequency in rooms are random if one introduces in the room small objects of dimensions equivalent to the wave length of interest. Up to then, practitioners believed that big diffusing objects were necessary to

J.-D. Polack (✉)
Institut Jean Le Rond d'Alembert, Sorbonne Universités, UPMC Univ Paris 06, UMR 7190, Paris F-75005, France

Institut Jean Le Rond d'Alembert, CNRS, UMR 7190, Paris F-75005, France
e-mail: jean-dominique.polack@upmc.fr

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_5,
© Springer International Publishing Switzerland 2015

randomize the modal distribution. In the second paper, he carried on with statistical properties of such randomized sound field, using advanced statistics recently developed by Rice. Thus, he could show that, above a cut-off frequency dependent of volume and reverberation time, modes overlap and combine so that average distance between peaks, between troughs, and even the standard deviation, was predicted by the theory of random noise.

Several years later in 1962, he and his collaborators went one step further [3]. They simulated the random distribution of sound fields in rooms by means of a powerful tool of random process theory: Monte-Carlo simulation. In analogy with the hazard game of roulette, which made Monte-Carlo casino world famous, Monte-Carlo technique simply simulates complex processes by choosing values at random, but following a given probability distribution, and combining them according to the properties of the process to be simulated. Using this powerful technique, Schroeder and his collaborators were thus able to simulate the frequency response of a room, showing that it displayed the same characteristics as real measurements.

The assumptions upon which Schroeder based his simulation are now well known. They are twofold. Firstly, the resonance frequencies of the modes are distributed randomly, that is, they locally follow a uniform distribution. Secondly, the transfer functions of the modes at their resonant frequency follow exponential distributions with imaginary random arguments. Thus, the transfer function at any frequency can be obtained by superposition of the different modes that respond at the frequency. When modes overlap, the transfer function becomes Gauss distributed by virtue of the Central Limit Theorem of probability. Of course, this is only valid at high frequencies, when many modes overlap, but the approximation is considered satisfactory as soon as ten modes respond at a given frequency.

The present paper develop this idea and extend it to the time domain, as was suggested by Moorer [4] when he wrote that an impulse response is similar to white noise exponentially decreasing with time. Indeed, the properties of the Fourier transform ensure that the time domain response, that is, the impulse response of a room, also is Gauss distributed. Several researchers [5, 6] have independently tried to check this property on measured impulse responses, with mixed success. We shall review some attempts, and stress the reason for their failure: the absence of a proper model of impulse response. Indeed, statistics teaches us that it is almost impossible to prove statistical properties ex post. Only the existence of a model, that is, analysis ex ante, can prove the statistical properties of room response.

As a consequence, the next step is a review of Schroeder's model for the transfer function, and its translation in the time domain, generalizing Moorer's model. The paper then carries on with the analysis of a measured impulse response, arbitrarily selected among some 200 obtained during a recent campaign in Paris [7]. It stresses the necessity of compensating for the decay, in order to reconstruct a stationary signal, before carrying out a detailed statistical analysis in both the time and frequency domains. It concludes with the necessity of improving the deconvolution algorithms currently used in room acoustics to obtain impulse response from sine sweep.

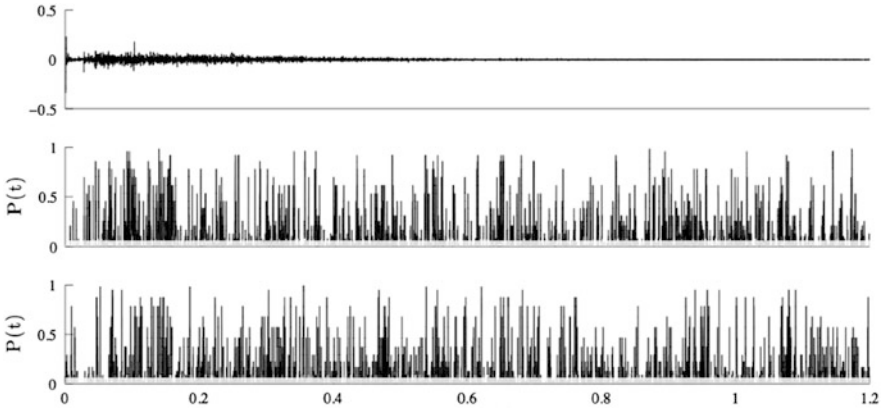


Fig. 5.1 Impulse response from Salle Pleyel (*top*) and the probability that samples are distributed according to Gaussian law of same variance and mean value (Kolmogorov-Smirnov test)—*middle curve*: using 200 samples; *bottom curve*: using 120 samples—gray values correspond to intervals where the test fails

5.2 Results from Previous Research

In 1989, the present author offered a generalization of Schroeder's model [8] that gave the theoretical framework of Jot's approach to digital reverberation filters [9]. As a consequence, several authors [5, 6] have tried recently to check the validity of the model. They used different statistical tests, but all of them rely on some statistical estimator: the ratio of the kurtosis to the variance [5], or Kolmogorov test [6]. We briefly present the results of Defrance [6].

Defrance used a set of impulse responses measured in Salle Pleyel with pistol shots (Fig. 5.1), and checked whether it was distributed according to a Gaussian distribution of same variance and mean value (null hypothesis). He used Kolmogorov-Smirnov test, that is, he compared the distribution of the experimental values to the empirical distribution of data obtained with a Gaussian random generator fed with the same mean value and the same variance. Most of the time, the impulse response passes the test (probability $P(t) > 0.05$, black values in Fig. 5.1), although the probability remains low, but every now and then the null hypothesis is rejected, meaning that the two distributions are not the same. A striking feature of Fig. 5.1 is the fact that the results of the test depend on the length of the window used (200 or 120 samples), with slightly better results for longer windows. Other tests give similar results [6], leading to the rejection of the hypothesis that impulse responses are distributed according to Gaussian laws (see also [5]).

In order to check the correctness of the procedure, Defrance also evaluated simulated impulse responses constructed by weighting Gaussian random noise with an exponential window. The results of Kolmogorov-Smirnov test are given in Fig. 5.2. This time, the impulse response passes the test most of the time, with

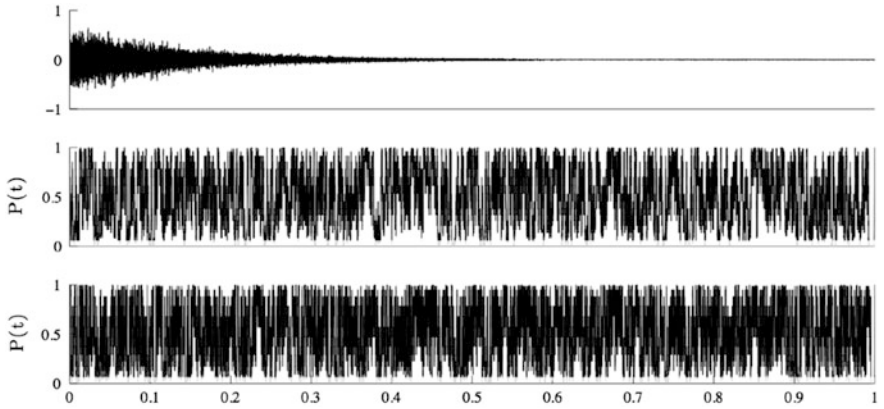


Fig. 5.2 Gaussian impulse response (*top*) and the probability that samples are distributed according to Gaussian law of same variance and mean value (Kolmogorov-Smirnov test)—*middle curve*: using 200 samples; *bottom curve*: using 120 samples—gray values correspond to intervals where the test fails

rather high probability. But every now and then, it fails the test. Once again, the results depend on the length of the analysis window used for the test.

The conclusion of this survey of previous research is that the methodology they use is not appropriate. Indeed, they are not constructed on a theoretical framework, like Schroeder's model, but only attempt to test a property: that impulse responses are Gaussian. In other words, they want to prove statistical properties ex post, whereas an analysis ex ante, making use of the properties of a model, is necessary to prove the statistical properties of room responses.

5.3 Generalizing Schroeder's Model to Time Domain

As stressed in the previous section, no proof of the statistical properties of impulse responses can be achieved without a proper model of impulse responses. This model relies on Schroeder assumptions:

- The resonance frequencies of the modes locally follow a uniform distribution.
- The transfer functions of the modes at their resonant frequency follow exponential distributions with imaginary random arguments.

At arbitrary frequencies, therefore, one must take into account the bandwidth of each mode, and superpose the modes accordingly (Fig. 5.3). The result is a random walk.

As shown in Fig. 5.3, it results in a complex transfer function, which is best decomposed into its real part $X(\omega)$ and its imaginary part $Y(\omega)$:

$$H(\omega) = X(\omega) + jY(\omega) \tag{5.1}$$

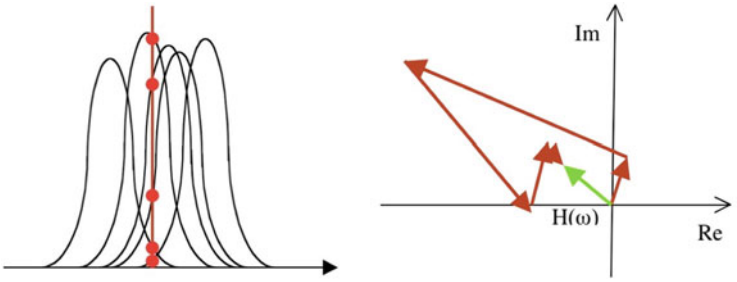


Fig. 5.3 Superposition of modes with random initial phases, taking into account their bandwidth (*right*)—it results in a random walk (*right*)

The theory of random walk then predicts that both the real and imaginary parts follow centered Gaussian distribution of same variance:

$$\begin{aligned} \langle X(\omega)\rangle &= \langle Y(\omega)\rangle = 0 \\ \langle X^2(\omega)\rangle &= \langle Y^2(\omega)\rangle \end{aligned} \tag{5.2}$$

In other words, the real and imaginary parts are equidistributed. Further, they are decorrelated [3]:

$$\langle X(\omega)Y(\omega)\rangle = 0 \tag{5.3}$$

Now, Schroeder model is defined in the frequency domain, and applies to modes taken individually, with due consideration of their bandwidth, that is, of their decaying nature. As a consequence, when translated into the time domain, it means that each mode will be exponentially decaying with random initial phase. By summing up all the modal contributions, one obtains an exponentially decaying random noise. However, since modes last the whole duration of the impulse responses, the random distribution of their superposition must be checked over the whole duration of the impulse response. Locally at some instant, nothing stops the impulse response from deviating from a Gaussian distribution.

In the following, we keep in mind the long-term characteristics of room impulse responses. As a consequence, we compensate for time variation, using the property of exponential decay. In a similar way, we must take into account the spectrum of the source when checking the Gaussian distribution of the transfer function; therefore, the next section is devoted to the presentation of room impulse responses.

5.4 Raw Analysis of Impulse Responses

For the purpose of illustrating the properties of room impulse responses, we arbitrarily chose an impulse response measured at Opéra Garnier in Paris during a recent campaign in 16 Parisian theaters and concert halls [7]. This response,

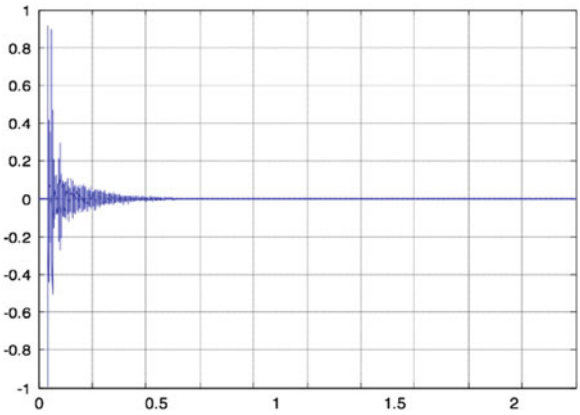


Fig. 5.4 Impulse response measured at Opéra Garnier in Paris

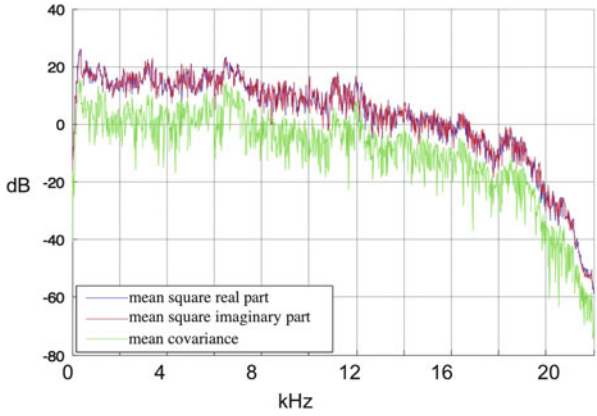


Fig. 5.5 Transfer function measured at Opéra Garnier. *Blue*: real part; *red*: imaginary part; *green*: cross-correlation

sampled at 44.1 kHz, is presented in Fig. 5.4. It was obtained by deconvolution of a logarithmic sine sweep of 30 s duration, using the Aurora suite developed by Farina [10].

From this response, we computed the transfer function by Fourier transformation, and estimated the running variances and cross-correlation according to Eqs. (5.2) and (5.3). These estimates were obtained by averaging 44 adjacent values of the squared real part, of the squared imaginary part, and of the product of the two. These estimates are presented in Fig. 5.5, where the real part is traced in blue, the imaginary part in red, and the cross-correlation in green. It is evident from Fig. 5.5 that the averaged squared real and imaginary parts coincide, since the blue curve is almost completely hidden behind the red one. On the other hand, the cross-correlation does not vanish, as predicted by Eq. (5.3). Proper estimation of the empirical process corresponding to averaging 44 adjacent values of the product of the real and imaginary parts leads to the prediction that the green curve should lay 16.5 dB below the two others [8], a value which agrees with Fig. 5.5.

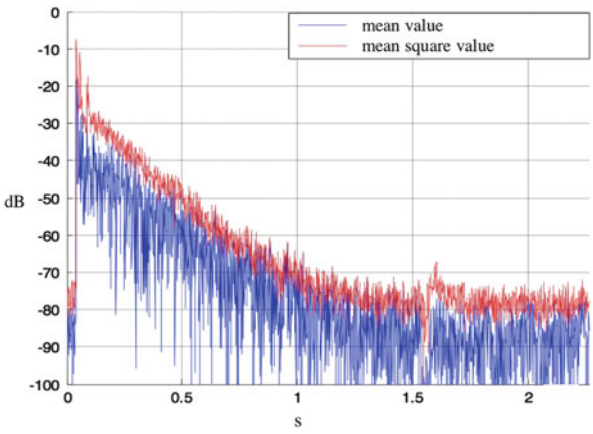


Fig. 5.6 Logarithmic plot of the impulse response. *Blue*: mean value; *red*: quadratic value

It is also instructive to have a look at the logarithmic display of the impulse response. Once again, we estimated the running variance and the running average on 44 adjacent values. They are displayed in Fig. 5.6, where the running average is traced in blue, and the running variance in red. This time, due to correlation between successive values, the difference between the two curves is much less than 16 dB [8].

It can be seen from Figs. 5.5 and 5.6 that Schroeder's model needs adaptation before any Gaussian distribution can be checked. First, the decay in time must be compensated, so that the impulse response approximates a stationary signal; then the spectrum must be compensated, to account for the spectrum of the source. These two issues are successively addressed in the next sections.

5.5 Compensating for Decay

The compensation procedure for the decay is illustrated in Fig. 5.7. Firstly, it is necessary to detect the background noise, or noise floor to which the impulse decays at larger times. This background noise is traced in dark blue for the mean square value, and in red for the mean value. Only the mean square value is of interest, and the noise floor is estimated by linear fit (green line in Fig. 5.7). By convention, any value within 10 dB of the noise floor is considered as background noise [11]. Therefore, the impulse response is reduced to the light blue part for its mean square, and the brown part for its mean.

The next step consists in estimating the decay by a linear fit of the mean square—magenta line in Fig. 5.7. Notice that the linear fit of the mean decay (yellow line) does not run parallel to the magenta line. This difference will come to light later on.

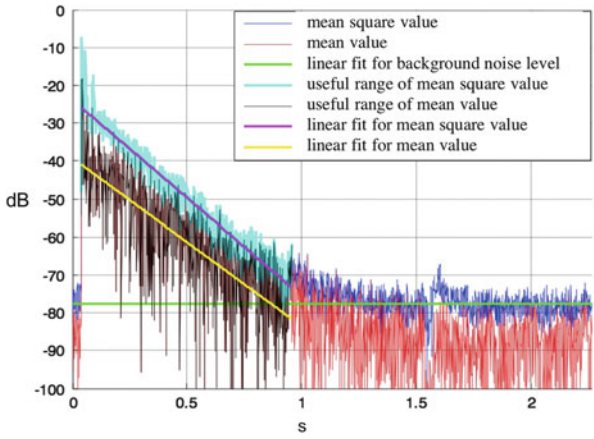


Fig. 5.7 The different linear fits used in the decay compensation process

Once the mean decay is obtained by linear fit, it is straightforward to compensate for the decay by multiplying the impulse response by the exponential of the logarithmic decay, while windowing it from 0.04 to 0.96 s. However, inspection of the compensated mean square (Fig. 5.8) reveals that the procedure is not sufficient. There remain abnormal values at the onset of the response, corresponding to coherent reflections. In a similar fashion, the mean square value increases toward the end of the window, revealing that some background noise is influencing the response. A shorter window must be used.

Figure 5.9 presents the mean compensated decay over the same window extending from 0.04 to 0.96 s. Now, the trend is different, with a positive mean slope of the curve. This corresponds to the fact that the yellow fit curve is not parallel to the magenta one in Fig. 5.7, having a smaller slope. In fact, it simply traduces the fact that computing mean values amount to low-pass filtering, albeit a very rough one, and that reverberation times at low frequencies usually are longer than at higher frequencies.

As a consequence, it is not sufficient to look at mean and mean square values to check for that compensated impulse responses follow Gaussian distribution, as was originally proposed in [8]. One must refine the analysis.

5.6 Detailed Analysis

The Artefact (cf. Oxford Dictionary) observed in the previous section led us to slightly amend the compensation procedure by selecting a more conservative time window in its last step. Figure 5.10 presents the thus selected portion of the impulse response used in this section.

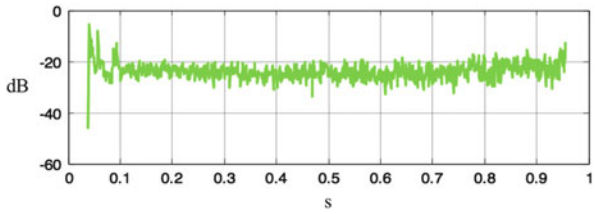


Fig. 5.8 Mean square of compensated impulse response

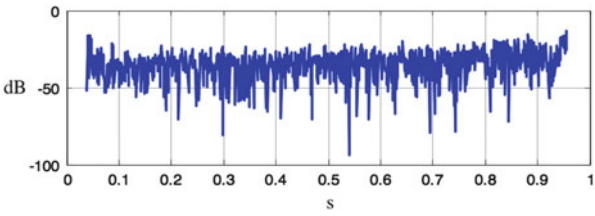


Fig. 5.9 Mean compensated impulse response

5.6.1 Impulse Response

Visual inspection of Fig. 5.10 reveals that the selected portion of the compensated impulse response looks indeed like white noise. It is therefore meaningful to compare its properties to the properties of white noise, for example by looking at the histogram of the values it takes (Fig. 5.11).

Figure 5.11 compares the histogram of the compensated impulse response, traced in red, to the theoretical histogram of a Gaussian distribution with zero mean and the same variance, traced in blue, computed for the same number of samples as contained in the impulse response. The two traces look similar. Therefore, we decided to check the distribution with statistical test. We used the Kolmogorov-Smirnov test, which compares the empirical distribution to a sample of the same number of random values that follow the theoretical distribution. We repeated the test several times with different simulated samples of the Gaussian distribution, and the compensated impulse response always passed Kolmogorov-Smirnov test.

However, a zoom on the histogram around the peak of the distribution reveals a slight skew of the impulse response. Indeed, its distribution does not top at zero value, as expected, but slightly below it (Fig. 5.12). We interpret it as a misalignment of the deconvolution filter that recovers the impulse response from the sine sweep measurement. Great care must be taken in the alignment of the filter in order to ensure Gaussian distribution, as well as the right amount of background noise.

5.6.2 Transfer Function

In order to check the distribution of the transfer function, it is also necessary to carry out some compensation of the source spectrum. Indeed, the transfer function

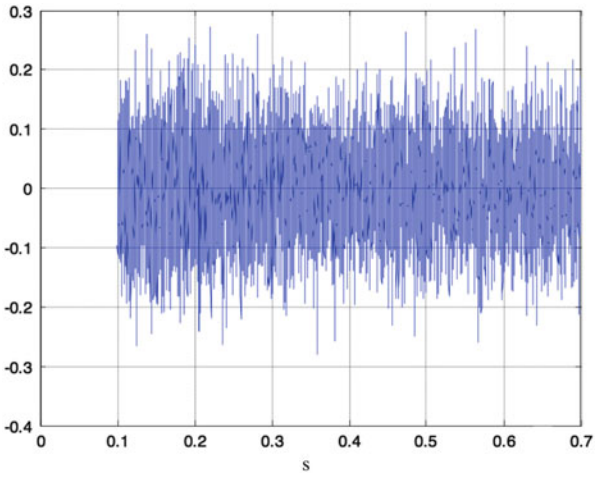


Fig. 5.10 Selected portion of compensated impulse response

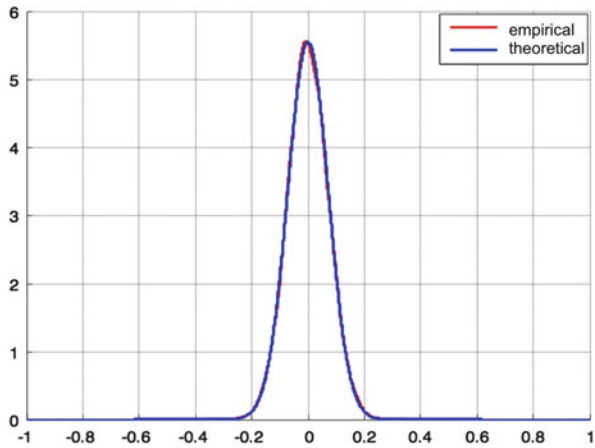


Fig. 5.11 Histogram of compensated impulse response

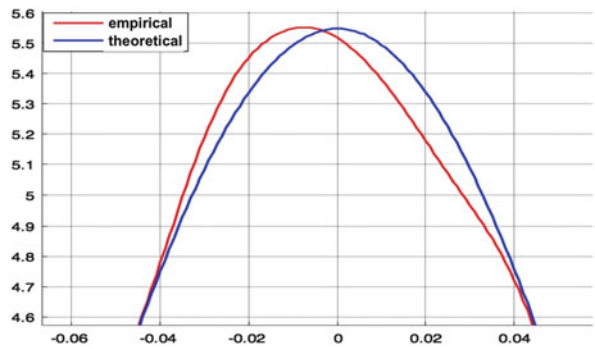


Fig. 5.12 Zoom on the histogram revealing misalignment of the deconvolution filter

computed from the impulse response by Fourier transformation is far from being flat, as shown in Fig. 5.5, but only the compensated part of the impulse response must be taken into account.

Figure 5.13 presents the raw spectrum of the compensated part of the impulse response (red). This estimate of the transfer function is far from being flat, and need smoothing before any compensation can be envisaged. Therefore, Fig. 5.13 also display several methods for smoothing the spectrum, from Welch spectrum computed with window lengths of 64 samples and 50 % overlap (green), to Burg spectral estimation based on 32nd order autoregressive process (light blue), and Yule-Walker spectral estimator with the order of the autoregressive model set to 16 (dark blue).

As little difference subsists between the Burg and Yule-Walker estimators, we decided to carry out the spectral compensation using the 16th order Yule-Walker estimator. In a similar way to the decay compensation of the impulse response, the complex spectrum computed from the compensated impulse response is then divided by the Yule-Walker estimator, yielding real and imaginary parts of the spectrum (Fig. 5.14) that look very similar after truncation of the central part of the spectrum—between 200 Hz and 14 kHz—and once again similar to white noise. Thus, proper distribution analysis can now be carried out.

Figure 5.15 presents the histogram of the real part of the compensated transfer function, traced in blue, and compares it to the theoretical histogram of a Gaussian distribution with zero mean and the same variance, traced in red, computed for the same number of samples as contained in the transfer function. The two traces look similar. Therefore, we decided to check the distribution with statistical test. Once again, we used the Kolmogorov-Smirnov test, and repeated the test several times with different simulated samples of the Gaussian distribution, and the compensated transfer function only passed Kolmogorov-Smirnov test some of the times.

This time, a zoom on the histogram around the peak of the distribution reveals a skew of both the empirical and theoretical distributions. None of them tops at zero value, as expected, but slightly below it for the empirical distribution, and slightly above it for the theoretical distribution (Fig. 5.16). But this time, an interpretation of this discrepancy is less evident, although it is obvious from Fig. 5.14 that the variance of the distribution slightly decreases with frequency, probably explaining why the Kolmogorov-Smirnov test sometimes fails.

Figure 5.17 presents the histogram of the imaginary part of the compensated transfer function, traced in blue, and compares it to the theoretical histogram of a Gaussian distribution with zero mean and the same variance, traced in red, computed for the same number of samples. The two traces look similar, except around the center of the distribution where the empirical distribution visibly has a higher peak than the theoretical one. As a consequence, several repetitions of the Kolmogorov-Smirnov test with different simulated samples of the Gaussian distribution always failed.

A zoom on the histogram around the peak of the distribution confirmed that the empirical distribution has a higher, and simultaneously narrower, peak than the theoretical one. Further, they both top at the same slightly negative value, but not at zero as expected (Fig. 5.18). This behavior points to a non-constant variance over

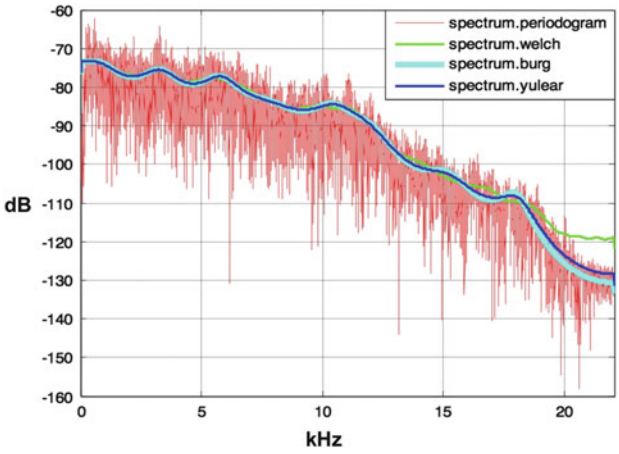


Fig. 5.13 Spectrum of compensated part of impulse response, and its estimation according to several procedures

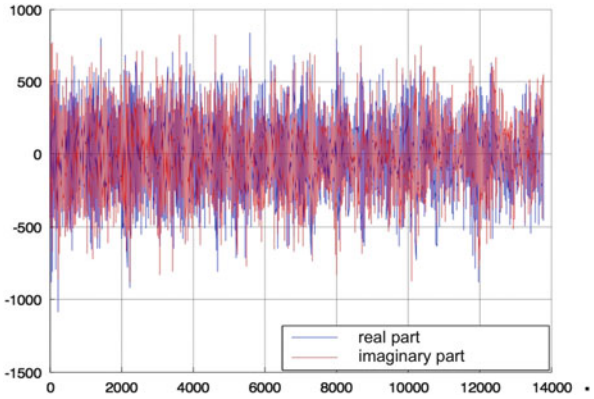


Fig. 5.14 Compensated complex spectrum. *Blue*: real part; *red*: imaginary part

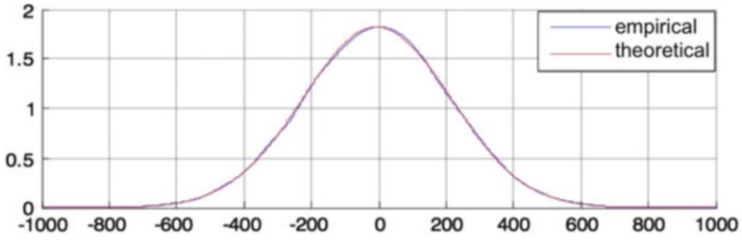


Fig. 5.15 Histogram of real part of compensated transfer function

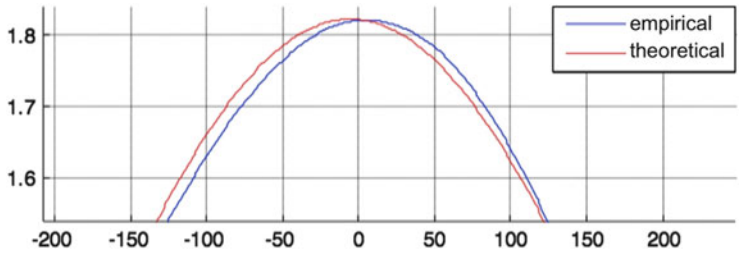


Fig. 5.16 Zoom on the histogram revealing abnormal values of both curves

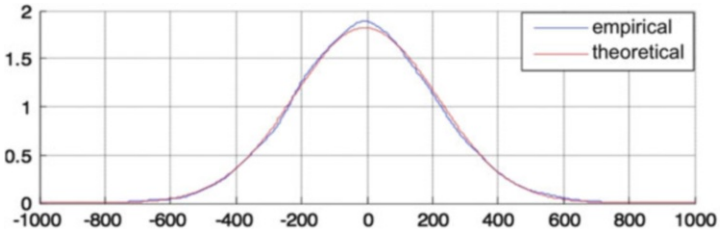


Fig. 5.17 Histogram of imaginary part of compensated transfer function

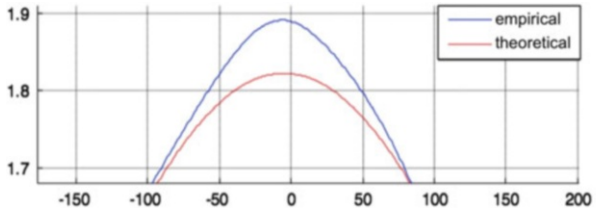


Fig. 5.18 Zoom on the histogram revealing abnormal peak of empirical distribution

the whole frequency range, as is visible in Fig. 5.14: the variance of the distribution decreases with frequency. This is why the Kolmogorov-Smirnov test always fails.

Indeed, complete analysis of the impulse response [8] reveals that the variance of the frequency response at a given frequency is proportional to the reverberation time at that frequency. Since reverberation times at high frequencies are always shorter, this is why the variance at high frequency is also smaller. Proper compensation of the spectrum should therefore take this property into account.

5.7 Conclusion

In this paper, we hope to have convinced the reader that, despite some unsuccessful previous attempts to prove it, impulse responses are Gaussian process, *provided* that global analysis is carried out on hand of a proper model of impulse responses. In

this process, three points are essential: discard the early part with strong reflections, and the late part which simply is background noise; accurately compensate for the decay; and compensate for the source spectrum. Further, we have shown that such an analysis can reveal the shortcomings of the measurement procedure, especially of the inverse filtering used to recover the impulse from the measurement signal: care must be taken that it accurately provides zero mean.

Moreover, the analysis has also revealed that Schroeder's simple model is not sufficient, especially in the frequency domain. There remains a frequency dependent variance which Schroeder's model does not account for. As a consequence, a more complex time-frequency compensation is needed to improve the model so that it passes the statistical tests. This sets the goal for further research in the domain.

Acknowledgement The author wants to thank Xi Zhang, who carried out all the computations used in this paper.

References

1. Schroeder, M.R.: Eigenfrequenzstatistik und Anregungsstatistik in Raümen. Acustica **4**, 456–465 (1954)
2. Schroeder, M.R.: Die statistischen Parameter der Frequenzkurven von grossen Raümen. Acustica **4**, 594–600 (1954); reprinted in M.R. Schroeder, "Statistical parameters of the frequency response curves of large rooms", J. Acoust. Eng. Soc. 35 (1987) 307–316
3. Schroeder, M.R., Kuttruff, H.: On frequency response curves in rooms. Comparison of experimental, theoretical and Monte Carlo results for the average frequency spacing between maxima. J. Acoust. Soc. Am. **34**, 76–80 (1962)
4. Moorer, J.A.: About this reverberation business. Comput. Music J. **3**, 13–18 (1979)
5. Stewart, R., Sandler, M.: Statistical measures of early reflections of room impulse responses. In: Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-07), Bordeaux, France, pp. 59–62 (2007)
6. Defrance, G., Polack, J.D.: Measuring mixing time in auditoria. J. Acoust. Soc. Am. **123**, 3499 (2008)
7. Figueiredo, F.L.: Indices acoustiques et leurs rapports statistiques: vérification objective et subjective pour un ensemble de salles de spectacles. Ph.D. thesis, Université Pierre et Marie Curie, Paris (2011)
8. Polack, J.D.: La transmission de l'énergie sonore dans les salles. Thèse d'Etat—Université du Maine, Le Mans (1988)
9. Jot, J.M.: Etude et realisation d'un spatialisateur de sons par modeles physiques et perceptifs. Ph.D. thesis, Telecom Paris (1992)
10. Farina, A.: Advancements in impulse response measurements by sine sweeps. In: AES 122nd Convention, Vienna, Austria (2007)
11. ISO 3382: Acoustics - Measurement of the reverberation time of rooms with reference to other acoustical parameters, (1997)

***

Biography



Dr. Jean-Dominique Polack received his Ph.D. in 1882 in Göttingen. He joined CNRS as Research Fellow in 1983, then was elected Professor of Acoustics at University Pierre et Marie Curie in Paris in 1997. From 1999 to 2001, he was appointed Professor of Electroacoustics at the Technical University of Denmark. From 2002 to 2006, he was head of the Laboratoire d'Acoustique Musicale, now merged into Institut d'Alembert. He is presently in charge of the Graduate School at UPMC. Dr. Polack's areas of expertise cover loudspeaker models, room-acoustic measurements, ergodic theory of sound fields, and sound quality of soundscapes.

Chapter 6
Recent Applications of Number-Theoretic Sequences in Audio and Acoustics

Ning Xiang, Bosun Xie, and Trevor J. Cox

Abstract Manfred R. Schroeder made influential contributions to acoustics by applying number theory. He introduced number-theory sequences in room impulse response measurements, applied number-theoretical sequences to shape wall surfaces for diffuse sound reflections. Another area Manfred Schroeder made broad impact is invention of an artificial reverberator using all-pass transfer-function properties to create colorless reverberation using a simple algorithm. Vast research activities and engineering developments over past decades have extended Schroeder's work involving artificial reverberators, artificial stereo and applications of maximum-length sequences. This chapter briefs a number of recent applications of maximum-length and quadratic-residue sequences, such as simultaneous dual/multiple sources measurements in acoustical tomography, in artificial pseudo-binaural reverberation, in decorrelation for spatial audio signals, and room acoustic diffuser design.

6.1 Introduction

In 1961, Manfred R. Schroeder, along with Ben Logan [1, 2] invented an artificial reverberator using all-pass transfer-function properties while having an exponentially decaying impulse response, thereby creating colorless reverberation using a simple algorithm. It has been widely accepted both in analog and digital audio devices for processing audio signals. One can now find colorless reverberators and artificial stereo, including simulations of virtual sound sources and auditory

N. Xiang (✉)
Graduate Program in Architectural Acoustics, School of Architecture, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
e-mail: xiangn@rpi.edu

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_6,
© Springer International Publishing Switzerland 2015

environments in various stereophonic and surround sound reproduction formats, in practically all electronic music instruments (See Schroeder's Memoirs in Part II).

In the 1970s, Manfred Schroeder made another influential contribution to acoustics with the introduction of number-theory sequences, applying bipolar, binary maximum-length sequences (MLSs) to applications such as shaping wall surfaces for diffuse sound reflections [3], and acoustics system identification to enable room impulse response measurements based on correlation techniques [4]. Extending the wall surface work to exploit nonbinary sequences, led to Schroeder devising the quadratic residue diffuser [5] and other designs widely used in studios and performance spaces to improve acoustics [6] (see also Chap. 9).

Vast research activities and engineering developments over past decades have extended Schroeder's work involving artificial reverberators, artificial stereo and applications of MLSs. This chapter is dedicated to the memory of Manfred Robert Schroeder in briefing a number of recent applications of maximum-length and quadratic residue sequences, such as simultaneous dual/multiple sources measurements in acoustical tomography, in artificial pseudo-binaural reverberation, in decorrelation for spatial audio signals, and room acoustic diffuser design.

6.2 Properties of Maximum-Length Related Sequences

MLSs, also termed pseudorandom sequences/noises, offer excellent pseudorandom properties. Therefore, they are used in communication, acoustics, and many other relevant applications. They possess properties similar to those of random noise, but are periodically deterministic and have a strict time structure within their periods. In the following, a brief review of some basic properties pertaining to acoustic applications is given. Detailed description and definitions can be found in [7–9]. In particular, Xiang [9] has most recently summarized these properties succinctly.

An n-stage linear shift-register device with proper feedback taps can generate periodic binary MLSs $\{a_i\}$ with $a_i \in (0, 1)$. Binary MLSs can also be generated numerically using a simple recurrence in the digital domain

$$a_i = \sum_{k=1}^{n} c_k a_{i-k}, \quad 0 \le i < L, \tag{6.1}$$

with $c_k \in (0, 1)$ and $c_0 = c_n = 1$, where the summation and indices are calculated, *modulo* 2 and the positive integer n is said to be the *degree* of the linear feedback registers or of the corresponding MLS. Both analog generation using linear feedback registers and numerical generation using the linear recurrence are subject to a nonzero *initial state* $\{a_{-n}, a_{-n+1}, a_{-n+2}, \ldots a_{-2}, a_{-1}\}$ denoted as $\{\overleftarrow{a}_i\} \neq \{0\}$. The proper feedbacks are expressed by feedback coefficients $c_k \in (0, 1)$; when resulting in a periodic sequence of length $L = 2^n - 1$, the sequence arrives at its maximum possible length, therefore, termed MLS. Under

this MLS condition, the proper set of feedback coefficients can also be expressed in polynomial form, $f(x) = \sum_{k=0}^{n} c_k x^k$, termed the *primitive polynomial*. One primitive polynomial given degree n corresponds uniquely to one MLS. The feedback coefficients c_k and the binary sequence element a_i are defined over Galois field GF(2) while the sequence $\{a_i\}$ is defined over Galois field $GF(2^n)$. The underlying mathematical calculus is a substantial subject of number theory [7].

Among a number of number-theoretic properties, the most important property of MLS is that the normalized periodic autocorrelation function of a bipolar MLS within one period is a two-valued function [8]

$$\phi(i) = \frac{L+1}{L}\delta(i) - \frac{1}{L}. \tag{6.2}$$

With large enough period length $L = 2^n - 1$, the periodic autocorrelation function approximates the *unit sample sequence* $\phi(i) \approx \delta(i)$. Therefore the power spectrum of any MLS is flat, expect for a "dip" at zero-frequency.

An MLS $\{b_i\}$ can always be derived from a given one $\{a_i\}$ in terms of a circular phase shift such that $b_i = a_{i+\tau}$ so that an invariant decimation $b_i = b_{2i}$ can be satisfied. Here all the indices are calculated modulo L. This specific MLS $\{b_i\}$ is designated as *characteristic* MLS and *self-similar* MLS [9]. This invariant decimation holds only for characteristic MLSs with decimation factors of 2^m with m being a positive integer [9]. There exists a unique characteristic MLS for any given primitive polynomial and a unique initial state $\{\overleftarrow{a}_i\}$ of the shift register to generate it. An algorithm deriving the required initial state $\{\overleftarrow{a}_i\}$ given the primitive polynomial for generating the characteristic MLS has recently been described by Xiang [10]. The characteristic MLSs with the invariant decimation are of practical significance, they are also relevant to the following applications.

Furthermore, *a variant decimation* is also of practical significance. When the decimation factor d is properly chosen, the decimation will lead to a pair of MLSs $(\{a_i\}, \{b_i\})$ with $\{b_i\} = \{a_{di}\}$, whose periodic cross-correlation function (PCCF) is of clearly lower value than the peak of the periodic autocorrelation function of either sequence. In this context, all the indices are calculated modulo L. One of the straightforward variant-decimation factors is $d = L - 1$. This decimation $\{b_i\} = \{a_{(L-1)i}\} = \{a_{-i}\}$ simply indicates that a time reversal of MLS $\{a_i\}$ always leads to another MLS $\{b_i\}$. This pair of MLSs $(\{a_i\}, \{b_i\})$ derived from decimating $\{a_i\}$ using factor $d = L - 1$ or equivalently from time reversing $\{a_i\}$ is termed reciprocal MLSs due to the fact that their corresponding primitive polynomials are reciprocal to each other [10]. The most relevant correlation properties pertaining to the applications described in this chapter are that one can find MLS pairs or sets of sequences such that the PCCFs between them are of lower values, and the absolute bounds of these low values are deterministically predictable. For example, the bound value of the reciprocal MLS pair of degree n has been reported [8, 11]

$$l_r(n) = \frac{2^{(n+2)/2} - 1}{2^n - 1}. \tag{6.3}$$

For MLSs whose degrees are not a multiple of 4, some decimation factors are of the form $d = 2^k + 1$ or $d = 2^{2k} - 2^k + 1$, where k is chosen such that $n/\gcd(n, k)$ is odd, with gcd(,) being the greatest common divider. A decimation using these factors leads to an MLS pair $(\{a_i\}, \{b_i\})$, with $\{b_i\} = \{a_{di}\}$. For MLSs whose degrees are a multiple of 4, a decimation factor $d = 2^{(n+2)/2} - 1$ leads to a MLS pair. The cross-correlation between the original MLS $\{a_i\}$, and the decimated one $\{b_i\}$ results in small values bounded [8] by

$$l_p(n) = \frac{2^{\lfloor (n+2)/2 \rfloor} - 1}{2^n - 1}, \tag{6.4}$$

where $\lfloor \alpha \rfloor$ denotes the integer part of the real number α.

Besides the reciprocal MLS pairs, this chapter will refer to MLS pairs derived from above-mentioned decimation factors with low-valued PCCFs as *preferred MLS pairs* (see Xiang [8] for a detailed summary). In addition, one can derive a large number of sequences by combining the preferred or reciprocal MLS pairs $(\{a_i\}, \{b_i\})$, as $G_\tau(i) = a_i \oplus b_{i+\tau}$ with $\oplus$ denoting addition modulo 2, so-called Gold sequences $\{G_\tau(i)\}$, whose correlation bound values between each other are the same as those of either preferred MLS pairs or reciprocal pairs [8]. Similarly a decimation from an MLS $\{a_i\}$ of even-numbered degree with a decimation factor $d = 2^{n/2} + 1$ leads to a pair $(\{a_i\}, \{e_i\})$, with $e_i = a_{di}$. The combination $K_\tau(i) = a_i \oplus e_{i+\tau}$ leads to a binary Kasami sequence $\{K_\tau(i)\}$, and their low-valued PCCFs represent even lower bounds,

$$l_K(n) = \frac{2^{n/2} + 1}{2^n - 1}, \tag{6.5}$$

approximately half of those of preferred pairs and Gold sequences.

Figure 6.1 illustrates the bounds of the PCCF of preferred pair, reciprocal pairs, and Kasami sequences. Gold sequences have the same bound values as those of preferred pairs and reciprocal pairs for even-numbered degrees. The preferred pairs possess approx. 3 dB lower bound values than those of reciprocal pairs for odd-numbered degrees. And the bound values of Kasami sequences, existing only for even-numbered degrees, are as $6 \sim$ dB lower than those of the others. Figure 6.2 shows auto- and cross-correlation functions of Kasami sequences of degree 14 to illustrate the excellent correlation properties of MLS-related sequences.

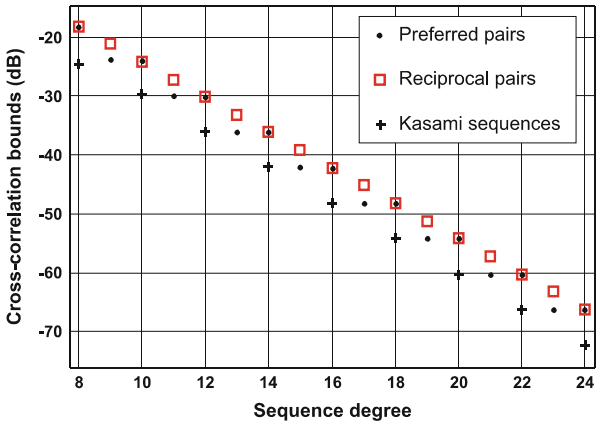


Fig. 6.1 Bound values of the cross-correlation functions of preferred, reciprocal MLS pairs and of Kasami sequences. The bound values are expressed in dB relative to the peak value of their normalized autocorrelation function [11]. [Reproduced from Xiang, N., Daigle, J. N., and Kleiner, M.: Simultaneous acoustic channel measurement via maximal-length-related sequences, *J. Acoust. Soc. Am.*, 117, 2005, pp. 1889–1894. Copyright 2005, Acoustical Society of America.]

6.3 Acoustical Measurements Using Simultaneous Sound Sources

Recent research in outdoor sound propagation for acoustic atmospheric tomography [12] calls for a critical measurement exploiting these correlation properties of MLSs and related coded signals. In tomographical applications, a number of sound sources have to be excited at the same time, within the same frequency range. This simultaneous multiple acoustic source measurement (SMASM) can be used in acoustic delay-time tomography to investigate temperature distributions and wind profiles near the ground surface in outdoor environments. With simultaneous excitations of multiple sound sources and one or multiple sound receivers, the SMASM considers the acoustic system under test as a linear time-invariant multiple-inputs and multiple-outputs (MIMO) system, at least approximately during the excitation period. Using the excellent correlation properties of the coded signals (sequences) briefly discussed above, a large number of coded signals can be straightforwardly derived for the SMASM technique.

Figure 6.3 illustrates a SMASM scheme, where the vector $\mathbf{s} = [s_1, \ldots s_n]^T$ stands for the multiple coded signals used as system excitations, and $\mathbf{r} = [r_1, \ldots r_p]^T$ denotes the system's responses to these excitations, with $[,]^T$ standing for matrix transpose; n is the number of simultaneous sources, while p is the number of receivers.

With properly designed excitations, the system identification task is to determine the impulse response matrix $\mathbf{h} = [h_{ij}]$ for $1 \leq i \leq p$, $1 \leq j \leq n$, determined by

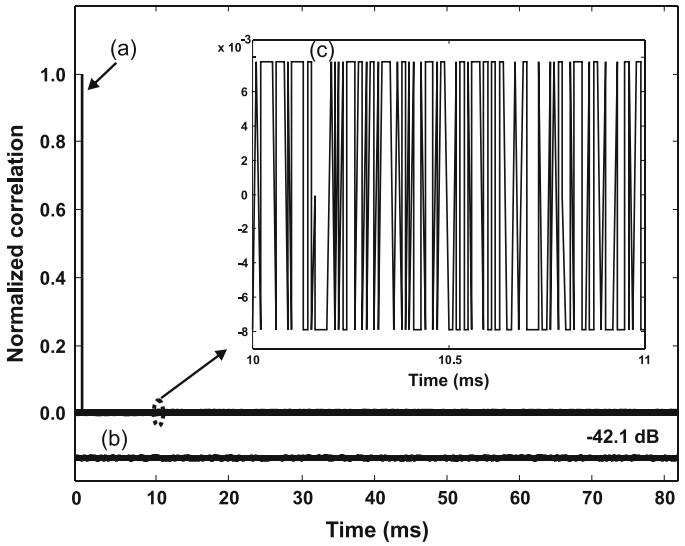


Fig. 6.2 Normalized correlation functions of 14-degree Kasami sequences [9]. (a) Periodic autocorrelation function (PACF) of individual sequences. (b) Periodic cross-correlation function (PCCF) between two Kasami sequences (shifted downwards beneath the autocorrelation function for a convenient comparison). The peak value of the PCCF is 42.1 dB lower than that of the delta-like PACF. (c) Magnified presentation of a segment from (a). Their peak values in the side-lobe of the PACF are the same as those of the PCCF in (b). [Reproduced from Xiang, N., Daigle, J. N., and Kleiner, M.: Simultaneous acoustic channel measurement via maximal-length-related sequences, *J. Acoust. Soc. Am.*, 117, 2005, pp. 1889–1894. Copyright 2005, Acoustical Society of America.]

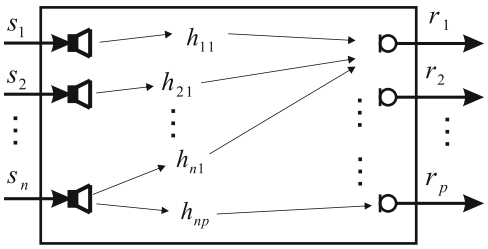


Fig. 6.3 An acoustical linear time-invariant system under investigation with n number of sound sources and p number of sound receivers. An $p \times n$ impulse response matrix $\mathbf{h}$ describes the entire system under investigation

$$\mathbf{h} \approx \mathbf{r} \otimes \mathbf{s}^T, \tag{6.6}$$

with $\otimes$ denoting periodic cross-correlation and the correlation properties discussed above leading to this concise form [11]. The physical meaning of Eq. (6.6) is that the cross-correlation between the receiver (column) vector $\mathbf{r} = [r_1, \ldots r_p]^T$ and the source (row) vector $\mathbf{s}^T = [s_1, \ldots s_n]$ approximately results in the channel impulse

response matrix **h**. The approximation is due to the fact that the cross-correlation functions among individual source signals in the form of coded sequences (Kasami sequences, preferred and reciprocal MLS pairs) are of finite low-values rather than zeros. In calculating the individual cross-correlations as expressed in Eq. (6.6), particularly for $n > 2$, a specialized algorithm [13] can be used for efficient calculations. In the case of $n = 2$, reciprocal MLS pairs can be exploited with a dedicated fast MLS transform algorithm recently published by Xiang and Schroeder [10].

6.4 Artificial Reverberations Using Reciprocal MLS Pairs and Related Sequences

Following Schroeder [1, 2], there have been many developments of all-pass filter type artificial reverberators. The most recent overviews on all-pass-filter artificial reverberators can be found in [14, 15]. Another line of development is reverberators using finite-impulse-response (FIR) filters, probably due to Moorer [16]. He proposed the use of exponentially decaying random noise for the late part of room impulse responses, so that when convolved with anechoic sound materials, artificial reverberation with a desirable degree of reverberance is created. With the rapid advent of digital audio processing on personal-computer and DSP platforms, FIR filtering via linear convolutions have reached multiples of real-time speed at low cost, so that even a large number of multiple audio channels can be filtered in real-time with sufficiently long FIR-filter coefficients (room impulse responses). FIR-filtering algorithms have also emerged that ensure sufficiently short latency for long room impulse responses [17].

There is a need for rendering the FIR-filter-based artificial reverberations binaurally, such as in binaural room-acoustic simulations [18], where the tails of binaural room impulse responses (see Fig. 6.4) are replaced by exponentially decaying random noise. The decay rates (reverberation times) are determined via statistical room-acoustic principles. Alternately, other room-acoustic information is first extracted from the early part of detailed room-acoustic simulations, such as ray-tracing, image-source approaches, or hybrid approaches [18]. Such schemes have recently been used for psychoacoustics studies investigating speech intelligibility, as well as perceptual aspects of acoustically coupled-volume systems [19].

In principle, an artificial reverberation for a binaural rendering can be achieved with a spatial envelopment, so-called *listening envelopment* when the two random-noise late reverberation tails are incoherent. This avoids using geometrical room-acoustic simulation (e.g., ray-tracing) to simulate the late reverberation tails, saving extremely time-consuming processing given the current technology in numerical simulations. Reciprocal MLS pairs have been used [19, 20] for this purpose, since the cross-correlation between pairs, given the MLS degree, is of low-value (see Fig. 6.2). The intriguing aspect of using MLS pairs, in contrast with other methods

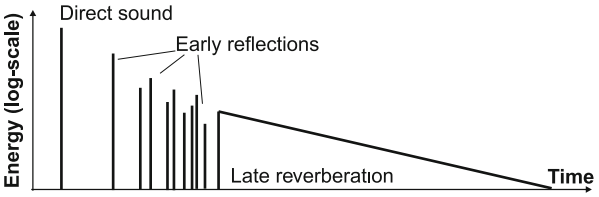


Fig. 6.4 Conceptual echogram containing the direct sound, early reflections, and the late reverberation tail

used in previous work [18] are that MLS pairs (and their related coded sequences) possess deterministically predictable, low values of cross-correlation. Hardly any other random noise signals can be found as low as these, Kasami sequences being the lowest. Lower values correspond to a high degree of enveloping spaciousness in the reverberance. In addition to the highest achievable degrees of spatial envelopment, one can use mix-networks [20] to control the cross-correlation values of the mixed MLS pairs for binaural channels, so that the degree of spatial envelopment can also be adjusted. This is of practical value, since different enclosure conditions will provide different spaciousness, in addition to the reverberance. As recent psychoacoustical investigations [21, 22] have demonstrated, simply shaping the decaying slopes of random noise in the late reverberation tails for achieving artificial reverberations using FIR-filtering technique is not sufficient in terms of naturalness and spatial envelopment of the artificial reverberation. Our recent work [21, 22] adjusts the decay slope (for controlled reverberance) in individual octave/third-octave bands. For each band, the target-reverberation time shapes the exponentially decaying envelope of the reciprocal MLS pair. In addition, interaural decorrelation coefficients which are complimentary to interaural cross-correlation coefficients (IACC) of the binaural reverberation tails

$$\text{IADC} = 1 - \text{IACC}, \tag{6.7}$$

in the individual bands also need to be adjusted accordingly in order to achieve targeted degrees of spatial envelopment. Adjusting only the decay slopes and the interaural decorrelation coefficients so that they are similar to those often observed in experimentally measured binaural room impulse responses creates natural sounding, artificial reverberance with the desired (or targeted) spatial envelopment. Figure 6.5 illustrates the procedure of the binaural artificial reverberation with controllable spatial envelopment and reverberance. A reciprocal MLS pair is first octave-band filtered. Two channels of band-pass filtered pseudorandom noise are mixed with an attenuation factor α_k with $0 \leq \alpha_k \leq 1$ and k being the octave-band index running from 63 Hz to 8 kHz at each octave-band step.

In Fig. 6.5 between "A" and "B" the two channels of the band-pass filtered, reciprocal MLS-pair are mixed to obtain the desired "interaural decorrelation." The band-pass filtered pseudorandom noise is then multiplied by an exponentially decaying function $E = \exp(-6.9 \cdot t/T_k)$, with the desired reverberation time T_k within octave band k. For broadband resulting "binaural" reverberation tails (late portion of room impulse responses)

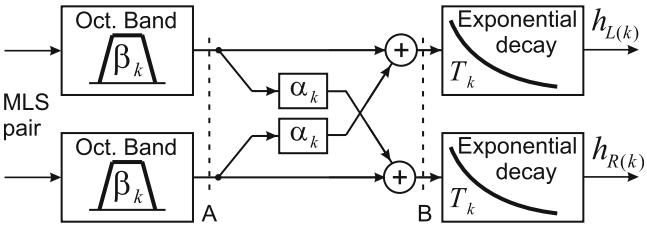


Fig. 6.5 Generation of single-band "binaural" reverberation tails with controllable interaural decorrelation coefficients and the reverberation times within octave band (k)

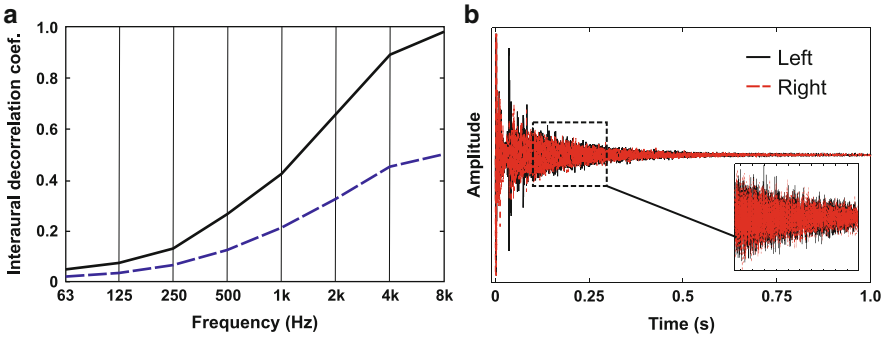


Fig. 6.6 Spatial profiles and the reverberation tails. (a) Two spatial profiles (large spaciousness, middle spaciousness) adjusted for two different degrees of listening envelopment. Interaural decorrelation is defined as 1 – IACC. (b) "Binaural" reverberation tails, properly scaled in terms of amplitude, it will be appended to the early portion of a pair of "binaural room impulse responses at time instance at 90 ms

$$h_{\mathrm{L,R}} = \sum_{k=1}^{M} h_{\mathrm{L,R}(k)}, \tag{6.8}$$

with $k = 1$ for octave band 63 Hz, $k = M$ for octave band 8 kHz, other k values correspond octave bands between 125 Hz and 4 kHz. In order to create naturally sounding, enveloping reverberance, in addition to the desired reverberation time profile, the interaural decorrelation profile also has to be adjusted. Figure 6.6a illustrates two different profiles of interaural decorrelation obtained using Fig. 6.5, inspired from analysis of experimentally measured binaural room impulse responses in existing real concert halls. Figure 6.6b illustrates one (binaural) pair of late-portion artificial "reverberation tails."

Advantages of this artificial enveloping reverberation scheme using reciprocal MLS-pairs are

- MLSs of degree 16–19 with a length between $2^{16} - 1$ and $2^{19} - 1$ are typical lengths used for the artificial reverberation application. At standard audio sampling frequency, the reciprocal MLS pairs are easily generated, and they

provide sufficient reverberation density when sampled using standard audio sampling frequency.
- Spectral flatness of each individual MLS ensures a colorless reverberance.
- Low cross-correlation values of reciprocal MLS pairs (see Fig. 6.2) ensure a high degree of listener envelopment. Reduced degrees of listener envelopment can be straightforwardly achieved using a mixing network [20].

6.5 Decorrelation of Audio Signals Using Reciprocal MLS Pairs [23]

Audio signal decorrelation is a technique that creates two or more replicas of an input signal, which have different waveforms but are perceived similarly to the original signals in most aspects except in some spatial auditory effects [24]. This is applicable to a variety of spatial audio processing effects, such as broadening the auditory source width, enhancing the listener envelopment in surround sound reproduction, and the externalization of auditory events in headphone representation, among others.

In order to create decorrelated audio signals without altering perceived timbre, decorrelation processing should change the signal waveform but leave the magnitude or power spectra of signals intact. A straightforward method is to filter the input signal with a pair of all-pass digital filters with unit magnitude and random phase responses ranging from $-180°$ to $180°$ at every discrete frequency. However, the degree of decorrelation resulting from this method is uncontrollable and unrepeatable.

As stated above, bipolar MLSs are pseudorandom sequences with deterministic and periodic structures, but possess characteristics similar to random noise. In particular, the favorable characteristics of nearly uniform power spectra and deterministic, low-valued cross-correlation functions between each reciprocal MLS pair make it an excellent candidate for the design of all-pass digital filters for audio signal decorrelation. In addition, taking advantage of the deterministic and periodic characteristics of MLS, the design of MLS-based decorrelation filters is controllable and repeatable. This is the advantage of MLS-based decorrelation filters over conventional all-pass filter with random phase. We can also use Kasami sequences or other coded pairs for the decorrelation algorithm, but reciprocal MLS pairs are easier to generate.

The design steps of reciprocal MLS decorrelation filters are outlined as follows:

1. Create a pair of n-degree or $L = 2^n - 1$ points reciprocal bipolar MLSs $\{a_i\} = \{a_0, a_1, \ldots a_{L-2}, a_{L-1}\}$ and $\{b_i\} = \{b_0, b_1, \ldots b_{L-2}, b_{L-1}\} = \{a_{L-1}, a_{L-2}, \ldots a_1, a_0\}$.
2. Sequences $\{a_i\}$ and $\{b_i\}$ are respectively used as the coefficients of a decorrelation FIR-filters pair.

As shown in Eq. (6.3) and Fig. 6.1, the bounds of the normalized periodic cross-correlation value of a reciprocal MLS pair, and thereby that of the reciprocal MLS-based filter coefficients, decreases with increasing degree or length of the MLS. A high degree or long length leads to nearly zero cross-correlation between the filter coefficients. The normalized cross-correlation between two filtered output signals is related to the normalized cross-correlation of filter coefficients and normalized autocorrelation $\Phi_0(n)$ of the input signal by:

$$\Phi_{\text{out}} = \Phi_{ab}(n) * \Phi_0(n), \tag{6.9}$$

where $\Phi_{ab}(n)$ is the normalized PCCF of the reciprocal MLS pair. Therefore, the lower the values of normalized cross-correlation between filter coefficients, the more obvious the decorrelation effect will be. For a better decorrelation effect, a pair of high degree or long length reciprocal MLS filters is preferred. This makes design of real-time devices complicated, however. In implementing real-time devices, the specialized convolution algorithm discussed in Daigle and Xiang is of highly practical significance [25]. Moreover, filters with excessive length may distort the transient properties of signals. In practice, the length of reciprocal MLS filters is selected as a compromise between decorrelation effect and simplicity.

Because the perceived effect of decorrelation is greater for frequencies below 1 kHz than for high frequency above 3 kHz, the perceived performance of decorrelation filters at a given length can be optimized or improved by properly adjusting the filter coefficients. In fact, a cyclic time shift of inverse-order MLS $\{a_{L-1}, a_{L-2}, \ldots a_1, a_0\}$ yields L new sequences $\{a_0, a_{L-1}, a_{L-2}, \ldots a_1)$, $\{a_1, a_0, a_{L-1}, a_{L-2}, \ldots a_2\}$, and so on. The cyclic time shift of an inverse-order MLS is equivalent to adjusting its phase characteristics. The original sequence $\{a_0, a_1, \ldots a_{L-2}, a_{L-1}\}$ and each of these time-shifted inverse-order MLS exhibit auto- and cross-correlation characters similar to those in Eq. (6.3) and Fig. 6.1. We can select the one from these time-shifted MLSs that makes the cross-correlation of filters output lowest for low-pass input signal (such as below 1.5 kHz).

Similar to the case of artificial reverberation, a pair of output signals with controllable but higher values of cross-correlation can be approximately obtained by an appropriate mixture of the reciprocal MLS pair.

The algorithm of reciprocal MLS-based audio signal decorrelation is applicable to broadening or controlling the auditory source width (ASW) in spatial sound reproduction. The ASW is an important attribute of hall spatial auditory perception and closely related to early lateral reflections in the hall [26]. The early IACC, which is derived from the interaural cross-correlation within the first 80 ms, can be used as an objective index to evaluate ASW. In stereophonic or multichannel surround reproduction, the cross-correlation between/among the loudspeaker signals can also be used to evaluate the degree of the decorrelation effect. Taking conventional stereophonic reproduction as an example, results from psychoacoustic experiments show that a pair of 2,047-point reciprocal MLS filters (at 44.1 kHz sampling frequency) yields a relatively natural decorrelation effect in terms of left-right symmetry of broadened auditory events as well as less timbre coloration.

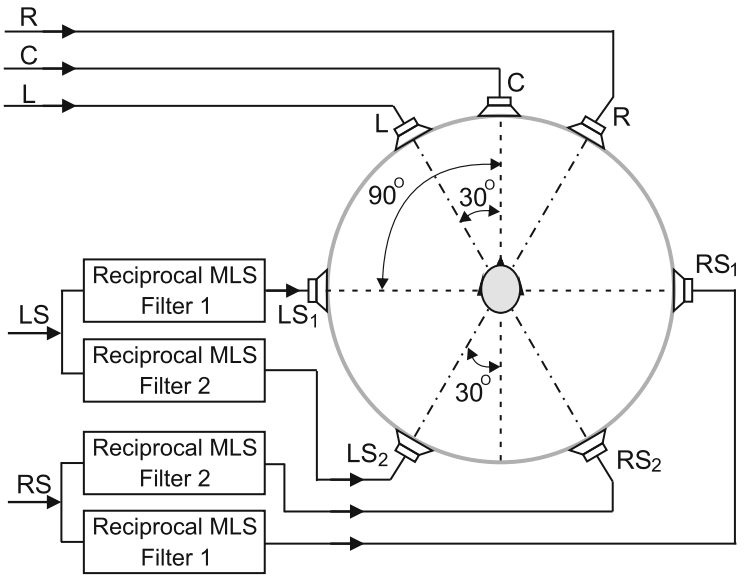


Fig. 6.7 Block diagram of converting 5.1 channel surround sound signals to 7.1 channel surround sound signals

The perceived performance of 511-point filters can be improved using the optimizing scheme mentioned above so that it is perceptually better than the original 1,023-point filters.

Another application of the algorithm of reciprocal MLS-based audio signal decorrelation is enhancing the listener envelopment in surround sound reproduction. It is well known that reproducing decorrelated signals via a series of surround loudspeakers results in the experience of better listener envelopment. An example is converting the 5.1 channel surround sound (music) signals to 7.1 channel reproduction. Figure 6.7 shows the block diagram of such an application. There are five independent full audio bandwidth channels in a 5.1-channel surround sound system, including left (L), center (C), right (R), left-surround (LS), and right-surround (RS), plus an optional low frequency effect channel (called .1 channel, which has been omitted in the figure). In the figure, the original L, C, and R signals are directly reproduced by three corresponding loudspeakers. The original LS and RS signals are filtered by two pairs of reciprocal MLS filters to yield four decorrelated surround signals LS_1, LS_2, LS_3, and LS_4, respectively, and then reproduced by four surround loudspeakers. The two reciprocal MLS pairs are derived from one pair of preferred MLSs. The length of reciprocal MLS filters is 511 points at a 48-kHz sampling frequency, and the optimizing scheme has been incorporated in the filter design. Here, a left-right symmetric decorrelation processing is adopted. That is, the reciprocal MLS filter 1 for signal LS is identical to that for signal RS, and so is the reciprocal MLS filter 2. For two pairs of resultant signals, LS_1 and RS_1 as well as LS_2 and RS_2, when the original signals LS and RS



Fig. 6.8 Gateway Mastering, Portland, ME showing a fractal diffusing rear wall (Diffractal®). (Photo courtesy of Gateway Mastering + DVD)

are decorrelated, the four signals are also decorrelated. While original LS and RS are correlated to each other, the signals within each pair are correlated, and the signals among different pairs are decorrelated. A preliminary subjective experiment shows that the algorithm of reciprocal MLS filter-based decorrelation improves listener envelopment in surround reproduction.

6.6 Diffuser Sequences

Figure 6.8 shows a diffuser applied to the rear wall of a studio, a fractal design which exploited the devices invented by Schroeder in the 1970s. Schroeder originally devised slatted wall surfaces based on MLSs [3], but these devices only operate across about an octave centered around the design frequency. Consequently, Schroeder sought out nonbinary number sequences such as the quadratic residue and primitive root sequences to form diffusers that had wells with many different depths that then operate over a wider bandwidth.

A quadratic residue sequence a_i is given by:

$$a_i = i^2 \bmod N; \quad 0 \leq i < N \tag{6.10}$$

where mod indicates the least nonnegative remainder; N is the number generator that must be an odd prime number and is also the number of wells per period. For example, one period of an $N = 7$ sequence is {0,1,4,2,2,4,1}.

For room diffusers, it is a complex exponential R_i incorporating the sequence that is crucial to how the structure reflects sound:

$$R_i = \exp\left(\frac{2\pi j a_i}{N}\right) \tag{6.11}$$

where j is $\sqrt{-1}$. Schroeder noted that R_i had an "astounding property" [5]. He was referring to the fact that the magnitude of the discrete Fourier Transform was constant and consequently the periodic autocorrelation function is the unit sample sequence.

The diffuser shown in Fig. 6.8 causes scattering in the horizontal plane. To diffuse sound vertically requires two-dimensional number sequences. To do this for a quadratic residue diffuser requires two number sequences, one for the horizontal (x-direction), one for the vertical (z-direction). Then the z-sequence is used to amplitude modulate the x sequence. For a quadratic residue sequence, this can be expressed as [5]:

$$a_{i,k} = \left(i^2 + k^2\right) \bmod N; \quad 0 \le i < N; \;\; 0 \le k < N \tag{6.12}$$

where i and k are the integers that index the sequence in the x and z direction respectively.

Another common method for making multidimensional phase grating diffusers is to use the Chinese remainder theorem [27]. This folds a one-dimensional sequence of length $N \cdot M$ into a two-dimensional array of size $N \times M$ while preserving the ideal autocorrelation and Fourier properties of the one-dimensional sequence. To use this method N and M must be co-prime. By co-prime, it is meant that the only common factor for the two numbers is 1. A quadratic residue sequence cannot be folded using the Chinese remainder theorem because it has a prime number of elements, and so an alternative sequence is needed, and the primitive root sequence is one possibility.

A primitive root sequence a_i is defined as:

$$a_i = r^i \bmod N; \quad 1 \le i < N \tag{6.13}$$

where N is an odd prime, r is the primitive root of N, and the sequence has $N-1$ elements per period. A primitive root is an integer that yields a sequence a_i for $i = 1$, 2, ... $N-1$ that are all unique. For example, $N = 13$ has a primitive root of 2, so $a_i = \{2,4,8,3,6,12,11,9,5,10,7,1\}$, which generates every integer from 1 to $N-1$. Primitive roots can be found by a process of trial and error, alternatively, tables can be found in texts such as [28]. The autocorrelation of the primitive root sequence placed in an exponential using Eq. (6.11) is a two-valued function.

Consider taking the length 12 primitive root sequence and wrapping it into a 3×4 array. The one-dimensional sequence is written down the diagonal of the array, and as it is periodic, every time the edge of the array is reached, the position is folded back into the base period. The process is illustrated in Fig. 6.9.

a_i mod 3 \ a_i mod 4				
	2			3
		4		
			8	

a_i mod 3 \ a_i mod 4				
	2	10	11	3
	6	4	7	9
	5	12	8	1

Fig. 6.9 *Top*: The first four elements of the length 12 primitive root sequence {2,4,8,3,6,12,11,9,5,10,7,1} placed in a 4×3 array using the Chinese Remainder Theorem. *Bottom*: the final complete table for one period

This sequence folding technique still maintains the good autocorrelation properties of the original one-dimensional sequence. In the case of the primitive root sequence, the two-dimensional autocorrelation will be two-valued and the power spectrum is flat, expect for a decrease at zero-frequency.

6.7 Diffusers

The sequences with good autocorrelation properties were turned into wall corrugations by Schroeder. When sound encounters a phase grating diffuser (see Fig. 6.10), provided that half a wavelength is larger than the width between the dividers, then plane waves will result within the wells. Plane waves propagate down each well, reflect from the bottom and return to the mouth of the diffuser. For the ith well with depth d_i, for sound with a wavelength of λ, the sound undergoes a phase change of $2jd_i(2\pi/\lambda)$ while propagating in the wells and the pressure reflection coefficient of the well at the front face of the diffuser is:

$$R_i = \exp\left(\frac{4\pi j d_i}{\lambda}\right). \tag{6.14}$$

Comparing Eqs. (6.14) and (6.11) shows that when:

$$\frac{2d_i}{\lambda} = \frac{a_i}{N}. \tag{6.15}$$

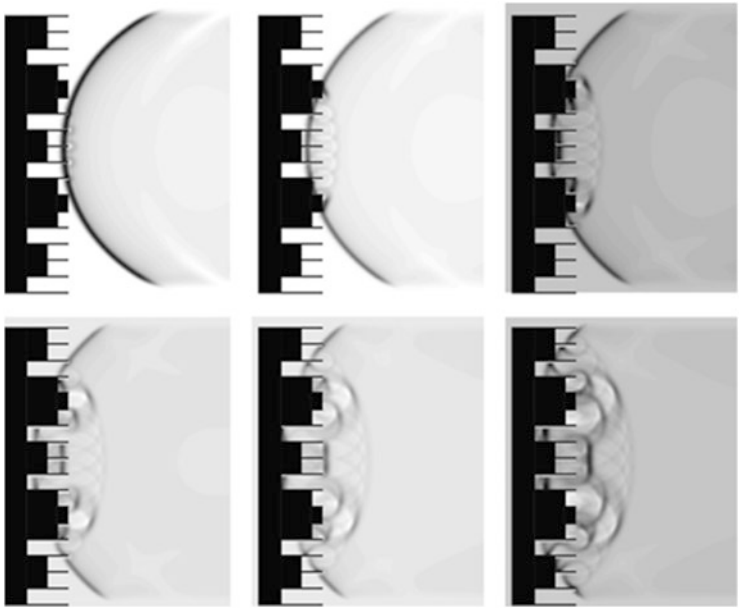


Fig. 6.10 Cylindrical wave reflected from a Schroeder diffuser calculated using a Finite Difference Time Domain (FDTD) model (after Cox and D'Antonio [6])

The sequence of sound waves reflected from the diffuser surface has the same astounding Fourier property that Schroeder identified. A rearranged version of Eq. (6.15) is therefore used to turn the number sequences into physical depths for a particular design wavelength.

What is heard some distance from the diffuser is a complicated interference pattern caused by the waves that emerged from each well before propagating to the listener. It is well-known from classical optics, that a Fourier transform of a wavefront passing through an aperture gives the far field diffraction pattern. Translating this rule for acoustic diffusers, it can be shown that the Fourier transform of the surface reflection coefficients gives the sound distribution in the far field. When there are many diffusers stacked side-by-side, the spatial periodicity causes energy to be preferentially reflected in certain directions creating so-called grating lobes. When a diffuser is formed from a quadratic residue sequence, at the design frequency these grating lobes all have the same energy. (See Fig. 9.11 in Chap. 9) This happens because the autocorrelation of the reflection coefficients is ideal.

In the 40 years since Schroeder pioneered modern diffusers for performance spaces, designs have been refined and improved as detailed in [6]. The autocorrelation of a quadratic residue sequence might be ideal at most multiples of the design frequency, but when translated into a real diffuser operating over many octaves, there are weaknesses in performance that allow room for further improvement. Researchers tried new number sequences, sequences were used in combination to

remove periodicity and fractal constructions were developed. All these designs are still, however, recognizable descendants of Schroeder's original diffusers.

One of the most ingenious features of Schroeder's work was the concept of using wells to create a set of known surface reflection coefficients. But this locked diffusers into a geometry that gave an appearance that was not liked by all architects. This drove researchers to come up with new shapes, like curves, designed using numerical optimization. While numerical optimization allows diffusers to be designed with a visual appearance to better match an architect's concept, the design method lacks the elegant simplicity of Schroeder's original ideas. Despite the complex underpinning of Schroeder's designs with number theory, the end design process was just a case of applying a few simple equations on a calculator.

References

1. Schroeder, M.R.: Improved quasi-stereophony and "Colorless" artificial reverberation. J. Acoust. Soc. Am. **33**, 1061–1064 (1961)
2. Schroeder, M.R.: Colorless artificial reverberation. J. Audio Eng. Soc. **9**, 192–197 (1961)
3. Schroeder, M.R.: Diffuse sound reflection by maximum-length sequences. J. Acoust. Soc. Am. **57**, 149–150 (1975)
4. Schroeder, M.R.: Integrated-impulse method measuring sound decay without using impulses. J. Acoust. Soc. Am. **66**, 497–500 (1979)
5. Schroeder, M.R.: Binaural dissimilarity and optimum ceilings for concert halls: more lateral sound diffusion. J. Acoust. Soc. Am. **65**, 958–963 (1979)
6. Cox, T.J., D'Antonio, P.: Acoustic Absorbers and Diffusers, 2nd edn. Taylor & Francis, London (2009)
7. Schroeder, M.R.: Number Theory in Science and Communication. Springer, Berlin (1991). 2nd enl. edn
8. Xiang, N.: Digital sequences. In: Havelock, D., Kuwano, S., Vorländer, M., et al. (eds.) Handbook of Signal Processing in Acoustics, pp. 87–106. Springer, Berlin (2008)
9. Xiang, N., Genuit, K.: Characteristic maximum-length sequences for the interleaved sampling method. Acustica **82**, 905–907 (1996)
10. Xiang, N., Schroeder, M.R.: Reciprocal maximum-length sequence pairs for acoustical dual source measurements. J. Acoust. Soc. Am. **113**, 2754–2761 (2003)
11. Xiang, N., Daigle, J.N., Kleiner, M.: Simultaneous acoustic channel measurement via maximal-length-related sequences. J. Acoust. Soc. Am. **117**, 1889–1894 (2005)
12. Ziemann, A., Arnold, K., Raabe, A.: Acoustic tomography as a method to identify small-scale land surface characteristics. Acta Acust. Acust. **87**, 731–737 (2001)
13. Xiang, N., Li, S.: A pseudo-inverse algorithm for simultaneous measurements using multiple acoustical sources (L). J. Acoust. Soc. Am. **121**, 1299–1302 (2007)
14. Gardner, W.C.: Reverberation Algorithms. Kluwer Academic Publishers, Dordrecht (1998)
15. Blesser, B.: An interdisciplinary synthesis of reverberation viewpoints. J. Audio Eng. Soc. **49**, 867–903 (2001)
16. Moorer, J.: About this reverberation business. Comput. Music J. **3**, 13–28 (1979)
17. Gardner, W.G.: Efficient convolution without input-output delay. J. Audio Eng. Soc. **43**, 127–136 (1995)
18. Van Maercke, D., Martin, J., Vian, J.-P.: Binaural simulation of concert halls: a new approach for the binaural reverberation process. J. Acoust. Soc. Am. **94**, 3255–3264 (1993)

19. Olson, S.: Understanding the perception of double-sloped energy decays: a multidimensional scaling approach. M.S. thesis, School of Architecture, Rensselaer Polytechnic Institute (2008)
20. Blauert, J.: Spatial Hearing. MIT Press, Cambridge, MA (2001)
21. Trivedi, U., Dieckman, E., Xiang, N.: Reciprocal maximum-length and related sequences for artificial spatial enveloping reverberation (A). J. Acoust. Soc. Am. **125**, 2735 (2009)
22. Trivedi, U., Xiang, N.: Utilizing reciprocal maximum length sequences within a multichannel context to generate a natural, spatial sounding reverberation (A). J. Acoust. Soc. Am. **126**, 2155 (2009)
23. Xie, B.-S., Shi, B., Xiang, N.: Audio signal decorrelation based on reciprocal maximal-length sequence filters and its applications to spatial sound. Preprint 8805, the 133rd convention of the Audio Eng. Soc., San Francisco, CA, 2012
24. Kendall, G.S.: The decorrelation of audio signals and its impact on spatial imagery. Comput. Music J. **19**, 71 (1995)
25. Daigle, J.N., Xiang, N.: A specialized cross-correlation algorithm for acoustical measurements using coded sequences. J. Acoust. Soc. Am. **119**, 330 (2006)
26. Beranek, L.: Concert Hall and Opera Houses, 2nd edn. Springer, New York (2004)
27. Schroeder, M.R.: Chaos, Power Laws: Minutes from an Infinite Paradise. W.H. Freeman & Co., San Francisco, CA (1991)
28. Fan, P., Darnell, M.: Sequence Design for Communications Applications. Wiley, New York (1996)

***

Biography



Prof. Ning Xiang earned his Ph.D. in 1990 from the Department of Electrical Engineering at the Ruhr-University Bochum (RUB) in Bochum, Germany. Since Aug 2003 he has been with the Department of Electrical, Computer, and Systems Engineering and the School of Architecture, Rensselaer Polytechnic Institute, Troy, New York, USA. Since 2005, Dr. Xiang has been the chair of the Graduate Program in Architectural Acoustics. Dr. Xiang is a Fellow of the Acoustical Society of America, a Fellow of the Institute of Acoustics (United Kingdom), and a Member of the German Acoustical Society since 1991, a Member of the Audio Engineering Society, a Member of INCE-USA, the Institute of Noise Control Engineering-USA. Dr. Xiang is currently an Associate Editor of the *Journal of the Acoustical Society of America* for the section Architectural Acoustics. In the Acoustical Society of America, he is now serving the Technical Committee Chair on Signal Processing in Acoustics. He is a recipient of Wallace Clement Sabine Medal from the Acoustical Society of America in 2014.



Prof. Bosun Xie received a Bachelor of Science degree in physics and a Master of Science degree in acoustics at South China University of Technology. In 1998, he received a Doctor of Science degree in acoustics at Tongji University. Since 1982, he has been working at South China University of Technology and is currently a professor at Acoustic Lab., School of Science. His research fields include binaural hearing, spatial sound, and sound signal processing. He has published a book named "head-related transfer function and virtual auditory display" and over 150 papers in these fields (most in Chinese). He is a vice-chairman of China Audio Engineering Society and a committee member of Acoustical Society of China.



Trevor Cox is Professor of Acoustic Engineering at the University of Salford, UK, a former Senior Media Fellow funded by the Engineering and Physical Sciences Research Council, and a past president of the UK's Institute of Acoustics (IOA). One major strand of his research is room acoustics for intelligible speech and quality music production and reproduction. Trevor's diffuser designs can be found in rooms around the world. He has co-authored a research book entitled "acoustic absorbers and diffusers." He was awarded the IOA's Tyndall Medal in 2004. Trevor has a long track record of communicating acoustic engineering to the public and has been involved in engagement projects worth over £1M. He was given the IOA award for promoting acoustics to the public in 2009. He has developed and presented science shows to 15,000 pupils including performing at the Royal Albert Hall, Purcell Rooms, and the Royal Institution. Trevor has presented 18 documentaries for BBC radio including: Life's soundtrack, Save our Sounds and Science vs. the Strad. He authored The Sound Book for W.W. Norton (entitled Sonic Wonderlands in the UK).

Chapter 7
Where Mathematics and Hearing Science Meet: Low Peak Factor Signals and Their Role in Hearing Research

Armin Kohlrausch and Steven van de Par

Abstract In his scientific work, Manfred Schroeder touched many different areas within acoustics. Two disciplines repeatedly show up when his contributions are characterized: his strong interest in mathematics and his interest in the perceptual side of acoustics. In this chapter, we focus on the latter. We will first give a compressed account of Schroeder's direct contributions to psychoacoustics, and emphasize the relation with other acoustics disciplines like speech processing and room acoustics. In the main part of the chapter we will then describe psychoacoustic work being based on or inspired by ideas from Manfred Schroeder. Due to Schroeder's success in securing a modern online computer for the Drittes Physikalisches Institut after returning to Göttingen in 1969, his research students had a head start in using digital signal processing in room acoustics for digital sound field synthesis and in introducing digital computers into experimental and theoretical hearing research. Since then, the freedom to construct and use specific acoustic stimuli in behavioral and also physiological research has grown steadily, making it possible to test many of Schroeder's early ideas in behavioral experiments and applications. In parallel, computer models of auditory perception allowed users to analyze and predict how specific properties of acoustic stimuli influence the perception of a listener. As in

This chapter is an adapted version of a chapter which the authors contributed to a book published on the occasion of the 60th anniversary of the Drittes Physikalisches Institut (DPI) at the Georg-August Universität Göttingen [37].

A. Kohlrausch (✉)
Behavior, Cognition, & Perception Group, Philips Research Europe, High Tech Campus 36 (MS WO 02), Eindhoven 5656AE, The Netherlands

Human-Technology Interaction, Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: armin.kohlrausch@philips.com

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_7,
© Springer International Publishing Switzerland 2015

other fields of physics, the close interplay between experimental tests and quantitative models has been shown to be essential in advancing our understanding of human hearing.

7.1 Introduction

One of the scientific areas that was close to the heart of Manfred Schroeder was psychoacoustics. During his first period in Göttingen, the focus of his work was on the physical and statistical side of acoustics. On moving to the Bell Laboratories in 1954, he came into an environment with a long history in hearing and speech research, started by Harvey Fletcher in the 1910s. In his autobiographic chapter, Schroeder describes a number of examples how he got interested in the perceptual side of acoustics, both from his research in room acoustics and in speech processing. During his time as professor at the University of Göttingen, the number of diploma and Ph.D. students focusing on hearing related studies increased continuously. This increased interest was supported by a rebuilding of the central space in the "Halle," in front of the "Reflexionsarmer Raum," where the control panels for the loudspeaker dome (well known to acousticians from a photo in Blauert's book on spatial hearing, see Fig. 3.50 in [2]) were dismantled and spaces for two listening booths were created—unofficially "owned" by the two research groups around Birger Kollmeier and the first author. In parallel with the acoustic spaces, the computer infrastructure for controlling listening experiments and generating acoustic stimuli also grew steadily. This experimental infrastructure and a growing group of young scientists were essential for the increasing level of sophistication of hearing research at the Drittes Physikalisches Institut (DPI).

In this chapter, we want to briefly summarize Schroeder's contributions to the psychoacoustic literature. In the major part of this chapter, we will describe how the particularly close link between hearing research and another of Manfred Schroeder's scientific interests, defining signals with very specific properties, like low peak factors, has influenced the psychoacoustic community. Some aspects of these developments are also closely interlinked with other facets of Schroeder's life, as will become clear by comparing the present text with the corresponding passages from his autobiographic chapter.

The advanced digital signal processing capabilities at the DPI, quite unusual for an institute of physics in the 1970s and strongly inspired by Schroeder's experience at the Bell Laboratories, allowed a high level of creativity in constructing signals with specific spectral and temporal properties in both listening experiments and model simulations. These innovative approaches, which started at the DPI, spread to other places like Eindhoven and Oldenburg, and later on to Lyngby/Kopenhagen, and have influenced research paradigms in hearing research groups all over the world.

7.2 Schroeder's Major Contributions to Hearing Research

Schroeder's interest in psychoacoustics had a number of roots. Through his work on efficient speech coding and the naturalness of synthesized speech, he developed an interest in the role of the signal envelope and fine structure on timbre. A specific manifestation was his work on "monaural phase effects." Furthermore, he used concepts of perceptual masking and perceptual distance measures in speech coding algorithms to improve the trade-off between perceptual quality and bit rate. Through his work on concert hall acoustics and digital reverberators, he recognized the great importance of spatial parameters such as the interaural cross-correlation, for good perceptual quality of a reverberant environment; this had led to work on binaural modeling. Actually, stimulated by his research in concert hall acoustics in the 1970s, binaural psychoacoustics evolved in the 1980s into one of the strongholds of his research group at the DPI in Göttingen. And finally, he had a genuine interest in developing and improving models of the auditory periphery—models that could help to understand specific perceptual phenomena, like the level dependence of difference tones. An excellent overview about how deeply he thought about these various aspects of hearing science can be found in his overview article from 1975: Models of Hearing [59].

Probably the psychoacoustic topic that interested Schroeder the earliest was the question of the extent to which the human hearing system could decode the phase spectrum of a signal. His first publication on this topic was an abstract from the 58th meeting, in 1959, of the Acoustical Society of America with the title: New results concerning monaural phase sensitivity [54]. Schroeder describes the perceptual consequences of changing the component phase in harmonic complex signals comprising up to 31 components. Aspects mentioned are the strong influence of the peak factor on the signal timbre, the absence of timbre changes for signals with identical envelopes, and the possibility to create strong and varying pitch phenomena allowing one to play melodies, just by changing phases of individual components.

The emphasis on the relation between timbre and waveform had a direct link to the work on vocoder quality, where, on the synthesis side, the excitation signal for voiced speech was composed of harmonic complex tones for which one could choose the relative phases freely. Schroeder and colleagues had observed that the excitation function strongly influenced the quality of vocoder speech (see, e.g., [55]). Schroeder described this early work, including waveform examples for different phase choices, in his 1975 IEEE paper [59] where he also referred to an earlier, less-known account by R. L. Pierce on this work in the American Scientist from 1960 [50]. In the course of his research on monaural phase effects, he also formulated closed solutions for generating periodic signals with low peak factors [57]. This short mathematical paper from 1970 remained relatively unknown for nearly 20 years, receiving only 29 citations until 1989. After this phase rule was introduced into hearing research in 1986 in [69] (see next section), the number of citations exploded and grew to a total of 312 (as of October 2013), making it,

together with the paper on measuring reverberation time, by far the most cited of his scientific publications.

Through his work on efficient speech coding, eventually resulting in code-excited linear prediction, Schroeder became acquainted with the phenomenon of masking and the great value of perceptually based distance metrics (see also the Chap. 10 in this book). The term "subjective error criteria" appeared first in the title of a conference contribution by Atal and Schroeder in 1978 [1]. As stated in the abstract of that paper, the human ear does not use simple RMS error measures when judging distortions introduced through coding. The new approach by Atal and Schroeder was to minimize the subjective distortion by shaping the spectrum of the resulting prediction error to be optimally masked by the speech spectrum. This approach does not reduce the total power of the error signal, which is determined by the quantizer, but redistributes its power along the frequency axis to have a more constant signal-to-noise ratio at all frequencies.

The concept of masking and its application in speech coding appeared in a follow-up article in 1979, jointly written with Atal and J.L. Hall [62]. The paper contains an extensive description of how to transform a short-term power spectrum into an excitation pattern, including transformation through outer and middle ear, calculation of critical-band densities, and transforming these into an excitation pattern from which the loudness was finally computed. The masked thresholds due to a masking signal, which in this application was the speech signal, were computed by multiplying the signal (speech) excitation function by a sensitivity function. In this way, the authors were able to predict whether, for a given short-term spectrum of the speech signal, a specific noise would fall below this masking function and was therefore inaudible. In addition to predicting the masked threshold, they derived, for suprathreshold noise levels, an objective degradation scale by relating the speech and the noise loudness (which, in modern terms, was calculated as partial noise loudness in the presence of the speech signal). This approach, in which state-of-the-art perception science was functionally integrated in a real-life signal-processing application, was later on also successful in the development of perceptual audio coding starting in the mid-1980s. An early account of this audio coding work, referring back to the earlier work by Schroeder et al. [62], is the paper by Johnston: Transform coding of audio signals using perceptual noise criteria [28].

An early account of spatial perceptual aspects in concert hall acoustics is given in the context of the acoustic measurements in the Philharmonic Hall [63]. One of the analyzed parameters which varied strongly between listener positions was the directional distribution of the early energy within 50 ms of the direct sound. After the acoustic changes in the hall had been finished, the variability of this parameter was much reduced. In their conclusion, the authors state, "It is therefore tentatively concluded that the directional distribution of early reflections is a significant contributing factor to acoustical quality" (p. 440 in [63]). This recognition of the role of spatial dissimilarity for good concert hall acoustics was further supported by the large comparative study of European concert halls, performed together with Gottlob and Siebrasse in Göttingen [64]. From their multidimensional analysis of

paired comparison data from many listeners, the interaural coherence turned out to be an independent dimension, being negatively correlated with subjective preference. This article addresses spatial hearing also from a different perspective. It describes the setup in the anechoic chamber in Göttingen, which permitted the reproduction of two-channel binaural recordings via two loudspeakers in an anechoic environment. The critical element in the reproduction chain was a cross talk cancellation computation, as it had been described and demonstrated 10 years earlier by Schroeder and Atal [61]. With cross talk cancellation the sound transmission from two loudspeakers could be directed to the two ears as otherwise only possible by headphone reproduction.

In his paper "New viewpoints in binaural interaction" at the fourth International Symposium on Hearing in Keele, 1977 [60], Schroeder made an original contribution to theories of binaural hearing, inspired by his work on basilar membrane characteristics. The core of his proposal used cochlear delays as a basis for the analysis of interaural delays. The computation of waveform delays between right and left ear would, instead of being enabled by interaural neural delay lines (as they are included in many types of binaural models), be realized by comparing interaurally the activities between different places on each basilar membrane. Such a comparison was possible and meaningful, because the basilar membrane activity has, for a given signal, a systematic relation between place and delay, and the *basilar membrane delay* values are of the magnitude necessary to analyze realistic *interaural delay* values. This proposal was later implemented and evaluated by Shamma and colleagues [67] who coined the term "stereausis" for this scheme. A consequence of this way of thinking about interaural delay, as pointed out by Joris and colleagues at the 13th International Symposium on Hearing in Dourdan, France [29], was that the traditional delay-line model using neural delay lines needed to have a very exact anatomical link between equivalent basilar membrane positions in the two cochleae. Only small spatial mismatches would introduce considerable offsets in the interaural delay values, particularly at low frequencies.

A final topic which we want to mention is Schroeder's research on basilar membrane modeling and its relation to specific psychoacoustic phenomena. Schroeder's interest in the relation between basilar membrane and hair cell properties on one side and perceptual observations on the other might have its roots in his attempts to relate cubic difference tone (CDT) data to the critical-band concept and basilar membrane mechanics. In his first paper from 1969 [56], he analyzed CDT data from Goldstein, and added basilar membrane simulation and additional CDT phase measurements performed at Bell Laboratories. As in similar earlier work by Zwicker, this established a close relation between CDT generation and auditory excitation. In addition, Schroeder concluded that a "purely mechanical model of the ear" was insufficient to explain the amplitude and phase characteristics of the $2f_1 - f_2$ difference tone. An additional amplitude nonlinearity was required, and Schroeder concluded, based on the sign of the resulting CDT, that it had to be an amplitude-*limiting* nonlinearity.

In the following years, Schroeder, often together with J.L. Hall, contributed papers on basilar membrane models [58] and a model for mechanical to neural transduction in the inner hair cell [65], both of which have been highly influential in the development of more realistic models for the auditory periphery. How useful fast and realistic time-domain basilar membrane models can be for the interpretation of psychoacoustics results will be demonstrated in the next section.

7.3 Harmonic Complex Tone Stimuli and the Role of Phase Spectra

A particular class of auditory stimuli are signals with a periodic waveform. They can be analyzed in terms of their temporal properties (e.g., the perceptual influence of a slight deviation from temporal regularity in an otherwise regular series of clicks, see [43]) or in the context of their spectral properties (consider, for example, the role of the vocal tract filter on the resulting vowel quality). Of course, from a mathematical point of view, time domain and spectral descriptions are fully equivalent, as long as the complex spectrum, including phase, is considered. Historically, however, temporal and spectral views were quite distinct, mainly because of the influence of signal analysis systems that represented the power spectrum but not the phase spectrum. Also, the psychoacoustic paradigm of critical bands and auditory filters, which for a long time were only defined in terms of their overall bandwidth and their amplitude characteristics (see, e.g., [19]), made it difficult to bring the temporal and spectral views closer together. Again, the increasing use of computer programs to generate acoustic stimuli and to perform time-domain modeling of perceptual processes emphasized the role of the phase spectrum on the perceptual quality of periodic signals (e.g., [16]). In addition, for us students in Göttingen, there was the strong interest of Schroeder in phase effects which prepared us more than other colleagues at that time to think in terms of signal waveforms, envelopes, and the time dimension (e.g., [66]). In the following, we will focus on one specific type of complex tones, tones with so-called Schroeder phases.

7.3.1 Schroeder-Phase Harmonic Complex Tones

7.3.1.1 Definition

The term “Schroeder phase” refers to a short paper by Schroeder from 1970 [57]. In this paper, he addressed the problem how the peak-to-peak amplitude of a periodic waveform can be minimized. His proposed solution lies in a phase choice which gives the signal an FM-like property. The general solution derives the individual phase values without any restriction on the amplitude spectrum. A specific case is

that of a harmonic complex with a flat power spectrum having a band-pass characteristic. The solution for the individual phase values of such a complex is as follows:

$$\phi_n = -\pi n(n-1)/N, \tag{7.1}$$

with N the total number of components in the complex. The important term in this equation is the quadratic relation between component number, n, and component phase, ϕ_n, which leads to an approximately linear increase in instantaneous frequency over time. The normalization with N creates a signal, for which the instantaneous frequency sweeps once per period from the frequency of the lowest to that of the highest component in the complex. In fact, the instantaneous frequency has a periodic sawtooth-like course for such Schroeder-phase signals.

It is obvious that reversing the initial sign in (7.1) has no influence on the peak factor of the resulting signal, but it will invert the direction of the linear frequency sweep. Because these two versions of a Schroeder-phase signal lead to substantially different percepts, a convention has been introduced to distinguish them. A *negative* Schroeder-phase signal is a signal were the phases of individual components are chosen as in (7.1). In contrast, a *positive* Schroeder-phase signal has phase values with a positive sign in front of the fraction. One can memorize this relation by using the fact that the sign of the phase is opposite to the direction of change in instantaneous frequency.

7.3.1.2 Acoustic Properties

By construction, Schroeder-phase stimuli have a relatively flat temporal envelope and the peak factor, defined as the ratio between envelope maximum and the RMS-value of the signal, is much lower than for other phase choices. This is demonstrated in Fig. 7.1, which compares the waveforms for harmonic complexes composed of 19 equal-amplitude harmonics of a 100-Hz fundamental from 200 to 2,000 Hz. There are three different choices of the component phases: positive Schroeder phase, negative Schroeder phase, and, in the top part, a zero-phase stimulus. For this latter stimulus, the energy is concentrated at very short instances within each period, leading to a much higher peak factor. The spectro-temporal properties of these signals can be seen more clearly in a short-time spectral representation.

Figure 7.2 shows the spectra of the three signals from Fig. 7.1 calculated using a moving 5-ms Hanning window. The sawtooth-like frequency modulation of the two Schroeder-phase complexes is pronounced in this representation. In addition, the plot for the zero-phase complex shows ridges at the spectral edges of 200 and 2,000 Hz. These relative spectral maxima can be perceived as pitch, superimposed on the 100-Hz virtual pitch of the complexes, and the presence of these pitch percepts has been used as a measure of the internal representation of such harmonic complexes [34, 35]. The visibility of the temporal structures in the short-time

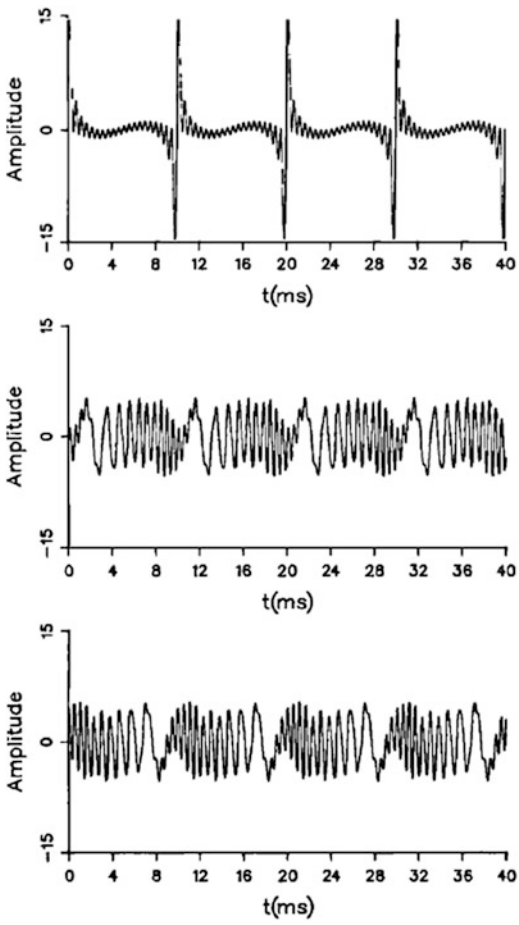


Fig. 7.1 Time functions of harmonic complexes for three different choices of the component's starting phases. *Top*: $\phi_n = 0$ zero-phase complex, *middle*: $\phi_n = -\pi n(n-1)/N$, negative Schroeder-phase complex, *bottom*: $\phi_n = +\pi n(n-1)/N$, positive Schroeder-phase complex. All complexes are composed of the equal-amplitude harmonics 2 to 20 of fundamental frequency 100 Hz. For this plot, the amplitude of an individual harmonic was set at 1. Reprinted from [36]. Copyright (1995) Acoustical Society of America

spectrum depends critically on the duration of the temporal window, relative to the period of the sound. The shorter the window, the more visible are temporal changes; the longer the window, the more the spectral properties are emphasized.

7.3.2 Role in Hearing Research and Perceptual Insights

The first paper in which the Schroeder-phase formula was used in hearing experiments was published by Mehrgardt and Schroeder in the proceedings of the 6th International Symposium on Hearing, 1983 [41]. In this paper, the quadratic phase formula from (7.1) was combined with an additional scaling factor, which permitted control of the spread of signal energy throughout the period. The spectrum of the harmonic complex was, however, not flat as in most later

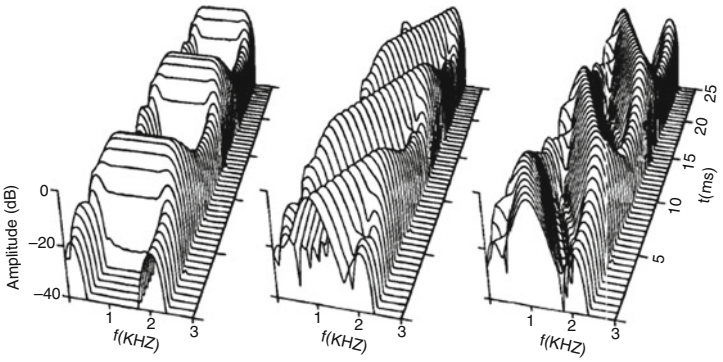


Fig. 7.2 Short-time spectral representation of the signals from Fig. 7.1 using a Hanning window with length 5 ms. The *left* panel shows the zero-phase signal, the *middle* panel the negative Schroeder phase and the *right* panel the positive Schroeder-phase complex. Reprinted from [36]. Copyright (1995) Acoustical Society of America

investigations, but the individual components had Hanning-weighted amplitudes. This paper emphasized the influence of the masker's temporal waveform on the observed masking behavior and showed how strongly the acoustic waveform and the resulting masked thresholds can vary by just varying the phase spectrum.

The great potential of Schroeder-phase signals to expose the phase characteristic of an auditory filter was discovered quite accidentally. During his master thesis research, Bennett Smith, traveling back and forth between Göttingen and Paris, where he continued to work as an engineer at the IRCAM, was interested in acoustic figure-ground phenomena, translating spatial orientation in the visual domain into linear frequency modulation in the auditory domain [68]. In the construction of his acoustic background stimuli, he made use of the Schroeder-phase formula. He did, unfortunately, not find any effect of acoustic figure-ground orientation on audibility, but made instead another observation: The audibility of a sinusoidal stimulus in a complex-tone masker depended strongly on the sign in the phase formula: When the background was constructed with a positive sign, the masked thresholds were lower by up to 20 dB, compared to the situation in which the phase sign was negative.

This large threshold difference formed a considerable scientific puzzle which kept the participants of Schroeder's weekly seminar busy for a prolonged period: Both masker versions had a similarly flat temporal envelope, so there was no reason to assume a difference in masking potential for simultaneously presented sinusoids. In addition, no temporal asymmetry, as in backward versus forward masking, could play a role given the simultaneous presentation of masker and test signal. We even considered the effects of FM to AM conversion of the masker waveform that should occur when the linear frequency modulation of the masker interacts with the shallow slopes of the auditory filter, without arriving at a satisfactory explanation. The effect was finally understood on the basis of computer simulations with a time-domain basilar membrane model which had been realized by Hans-Werner Strube

[70]. This model was, at that time, the first computer model with a realistic phase characteristic for the basilar membrane, which was able to compute the basilar membrane output in the time domain while being sufficiently time efficient. Strube observed that the two versions of the Schroeder-phase complex led to very different waveforms at the output of the various model elements that each represented a specific place on the basilar membrane. One waveform was clearly more modulated, and the existence of valleys in the masker waveform fitted nicely with the observation by Smith that this masker also led to lower masked thresholds: The target signal could be detected easier, i.e., at a lower level, within these masker valleys—a phenomenon for which the term "listening in the valleys" is used in the psychoacoustic literature. On the basis of the consistent explanation made possible by these simulations, we dared to summarize our experimental data and submit the manuscript to JASA. For me (AK), this was to become my first internationally peer-reviewed publication, and actually the only one jointly co-authored by Manfred Schroeder [69].

In the next years, a great many further experiments and model simulations were performed in Göttingen [30, 36], which led to the following insights: The clue to understanding the differences between positive and negative Schroeder-phase stimuli lies in the phase characteristic of the auditory filter, in combination with some general rules in masking of narrowband signals. If subjects have to detect a narrowband stimulus in a broadband masker, only the frequency region around the target frequency is of interest. In order to simulate the transformations in the auditory periphery for such an experiment, we have to compute the waveforms of the acoustic stimuli at the output of the auditory filter that is centered on the target frequency. For random noise maskers, only the amplitude characteristic of the filter is relevant, in line with the early "critical-band" concepts. For periodic signals as considered here, however, this filtered waveform, and its masking behavior, will depend critically on the amplitude and the *phase* characteristic of the auditory filter.

The most important conclusion was that, for the right choice of stimulus parameters, the phase characteristic of a Schroeder-phase stimulus matches quite closely the phase characteristic of the auditory filter, at least in the spectral region of maximum transfer of the filter. Since Schroeder-phase stimuli come in two flavors, one version, the negative Schroeder-phase stimulus, will have the same phase characteristic as the auditory filter, while the positive Schroeder-phase stimulus has a phase spectrum that is *opposite* to that of the filter. In the transfer through a filter, the input phase spectrum and the filter phase spectrum add to give the phase spectrum of the output signal. For the positive Schroeder phase, we thus have the situation of phase compensation, and the resulting filtered signal at the output of the auditory filter has a nearly constant phase of the strongest components, those with a frequency at the center of the corresponding basilar membrane filter, resulting in a highly peaked signal. Conceptually, this situation is quite similar to pulse compression through frequency modulation, as it is used in radar and sonar technology. In a way, the positive Schroeder-phase stimuli are matched in their phase characteristic to the auditory filters as they are realized mechanically in the inner ear.

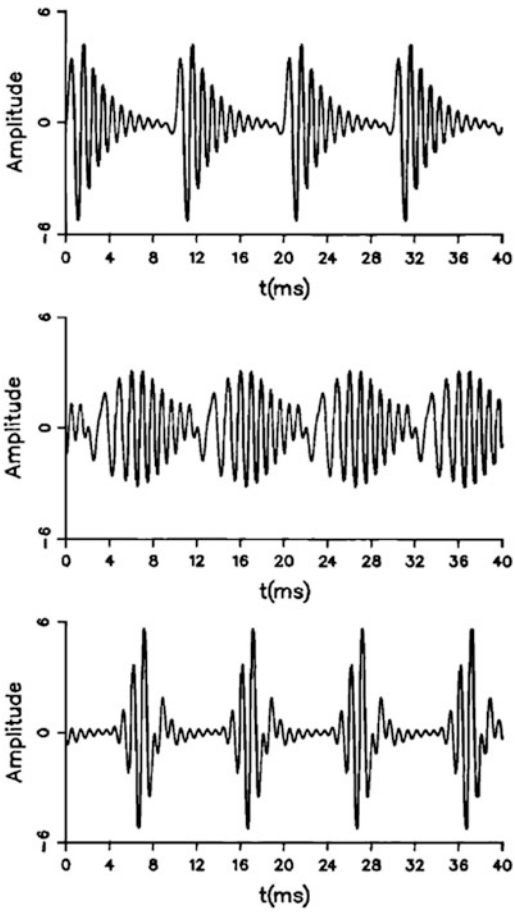


Fig. 7.3 Responses of a linear basilar membrane model at resonance frequency 1,100 Hz to the three harmonic maskers with fundamental frequency 100 Hz shown in Fig. 7.1. The *top* panel shows the zero-phase signal, the *middle* panel the negative Schroeder phase and the *bottom* panel the positive Schroeder-phase complex. Reprinted from [36]. Copyright (1995) Acoustical Society of America

This interpretation leads to some interesting, counterintuitive predictions: If a specific Schroeder-phase stimulus is optimally matched in its phase to the inner-ear filter at a certain frequency, then such a signal should have a higher peak-factor after filtering than a zero-phase input stimulus. This prediction is analyzed in Fig. 7.3, which shows waveforms of the three stimuli from Fig. 7.1 at the output of a linear basilar membrane filter tuned to 1,100 Hz. The two panels at the bottom show the waveforms of the two Schroeder-phase complexes. While the broadband input signals to the filter have a flat temporal envelope (cf. Fig. 7.1), both waveforms have a clear amplitude modulation after filtering, which follows the 10-ms periodicity of the stimulus. But the depth of modulation is quite different for the two signals, the positive Schroeder-phase stimulus at the bottom has a much higher peak factor and a longer period of small envelope values than the negative Schroeder-phase complex in the middle. This simulation reflects the initial observation made by Strube in [70].

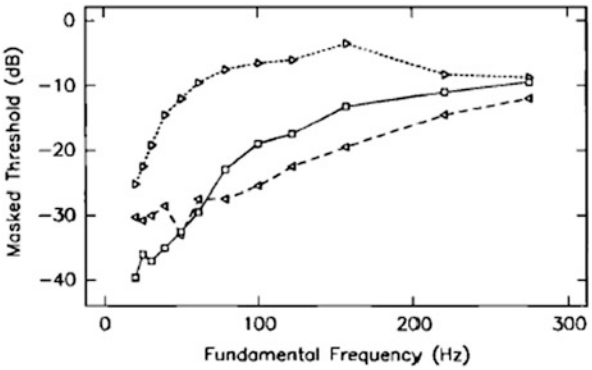


Fig. 7.4 Simultaneous masked thresholds of a 260-ms, 1,100-Hz signal as a function of the fundamental frequency f_0 of the harmonic complex masker. Thresholds are expressed relative to the level of a single masker component. The maskers were presented at a level of 75 dB SPL. *Squares*: zero-phase complex; *right-pointing triangles*: negative Schroeder-phase complex; *left-pointing triangles*: positive Schroeder-phase complex. Reprinted from [36]. Copyright (1995) Acoustical Society of America

The top panel shows the filtered version of the zero-phase complex, which has a highly peaked input waveform. The filtered waveform reflects, within each period, the impulse response of the auditory filter, because the zero-phase complex is similar to a periodic sequence of pulses. By careful analysis one can even see some properties of the filter's frequency-dependent group delay: around each peak, low-frequency components with a long waveform period occur first, followed later on by higher-frequency components. Comparing the top and the bottom panel reveals that, indeed, the positive Schroeder-phase complex has a somewhat higher peak factor than the zero-phase complex, and its energy is more concentrated in time, as expected based on the pulse-compression analogon.

The relation between zero-phase and positive-Schroeder-phase stimuli also formed the key to estimating the phase properties of a specific point on the basilar membrane. If we were able to determine the phase curvature, for which the match between stimulus and filter phase is "optimal," then this value would be an indication of the phase curvature (or the frequency-dependent group delay) of the filter. The clue to such an analysis is given by comparing perceptual thresholds for positive Schroeder-phase maskers with those for zero-phase maskers. For maskers, for which this difference is largest (and for which the Schroeder-phase stimulus as masker gives lower masked thresholds), the phase curvature at the signal frequency is an estimate of the filter phase. Figure 7.4 shows data from [36] which were used for such a computation.

In the region of f_0 values between 100 and 150 Hz, the differences in thresholds obtained with the positive Schroeder-phase complex (left-pointing triangles) and with the zero-phase complex (squares) are largest. For theses complexes, the second derivative of the phase-versus-frequency relation, which indicates the phase curvature has values between $1.05 \times 10^{-5}\pi/\text{Hz}^2$ and $0.74 \times 10^{-5}\pi/\text{Hz}^2$. We can

conclude that the curvature of the phase characteristic for the basilar membrane filter centered at 1,100 Hz should be in the range of these two values. A similar conclusion about the phase curvature can be derived from the parameters of those complexes, for which positive Schroeder-phase and zero-phase complexes lead to approximately the same threshold. In this case, the internal envelope modulation of the two complexes after filtering on the basilar membrane should be approximately equal. As explained in detail in [36], the phase curvature in the Schroeder-phase stimuli should be half the value of the filter curvature, and this is reached for fundamental frequencies of 50–75 Hz. And exactly in this region, the two lower curves in Fig. 7.4 cross each other.

This consideration allowed a first computation of the auditory filter phase for one frequency, 1,100 Hz. In [36], additional threshold measurements were included for frequencies 550, 2,200 and 4,400 Hz, thus covering a range of 3 octaves. It is often assumed that the auditory filter has a nearly constant quality factor across the range of audible frequencies. If this scaling relation were to hold also for the phase characteristic, then the results obtained at 1,100 Hz would allow a direct prediction for the phase characteristic in the range of 3 octaves around 1,100 Hz. The comparison with the results at 550 Hz indeed revealed the expected relation, while towards higher frequencies, the curvature changed somewhat less with center frequency than expected for a system in which the amplitude and phase characteristics of the filters remain constant on a logarithmic frequency scale.

One final important observation from these initial Schroeder-phase studies needs to be mentioned. In the 1980s, the view on the shape of auditory filters was strongly influenced by the work of Patterson, Moore and colleagues, who had used the notched-noise technique to estimate the amplitude characteristic of the auditory filter. The best characterization was possible with a so-called rounded exponential filter shape (see, e.g., [49]). A time-domain implementation of a filter with such an amplitude characteristic was possible based on so-called gamma-tone filters [27, 48]. Because of the many studies supporting this concept of auditory filters, we were interested to analyze the Schroeder-phase stimuli with such a filter.

Figure 7.5 presents, in a format similar to Figs. 7.1 and 7.3, four periods of the waveform for harmonic complexes with fundamental frequency 100 Hz. The analysis shows the output of the gamma-tone filter tuned to 1,100 Hz, and the three subpanels are for the three different phase choices. It is apparent that this filter does not lead to differences in the modulation depth between the three stimuli, and based on this simulation one would expect quite similar masking behavior of all three complex tones, in contrast to the experimental data. The major reason for the similar treatment of the two Schroeder-phase maskers by the gamma-tone filter is its antisymmetric phase characteristic close to its resonance frequency. The curvature of the filter phase changes its sign at the resonance frequency from negative to positive. A filter with such a phase characteristic can never flatten out the phase of a Schroeder-phase complex that changes over the full range of its passband, and should therefore be used with caution in experiments where signal phase matters. As has been shown later by Lentz and Leek [40], this shortcoming of

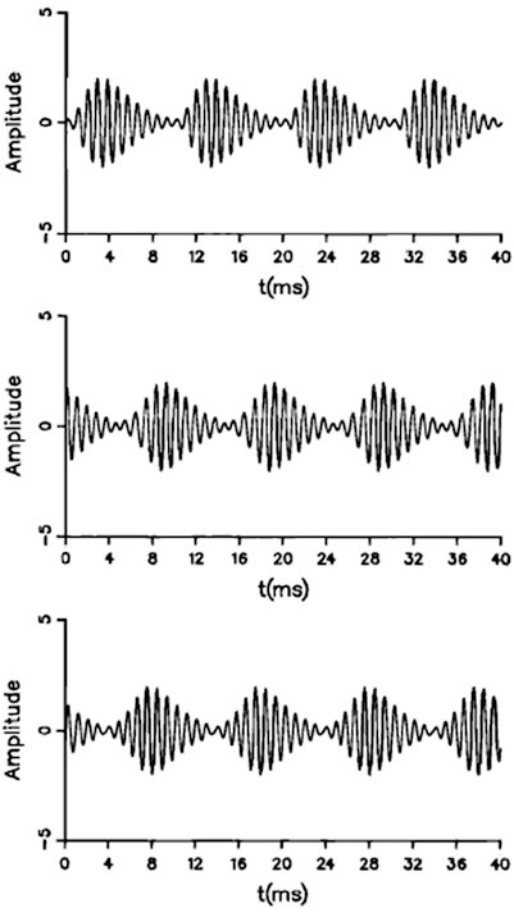


Fig. 7.5 Responses of a linear, fourth-order gamma-tone model at resonance frequency 1,100 Hz to the three harmonic maskers with fundamental frequency 100 Hz shown in Fig. 7.1. The *top* panel shows the zero-phase signal, the *middle* panel the negative Schroeder phase and the *bottom* panel the positive Schroeder-phase complex. Reprinted from [36]. Copyright (1995) Acoustical Society of America

the gamma-tone filter can be overcome by using a nonlinear extension of this filter, the gammachirp filter proposed by Irino and Patterson [26].

7.3.3 Later Developments

Although the first paper on Schroeder-phase stimuli was already published in 1986 ([69]), the paradigm was only widely adopted after publication of our second paper in 1995 [36]. The first papers that used the term "Schroeder phase" in their title were published in 1997 [9, 10]. Many authors related psychoacoustic findings with Schroeder-phase stimuli to the properties of the basilar membrane. Differences that were found between normal-hearing and hearing-impaired subjects and also influences of the overall presentation level indicated some role of active processes

in creating large differences between positive and negative Schroeder-phase stimuli [10, 71, 72]. Based on the results of these studies, Summers concluded: "The current results showed large differences in the effectiveness of positive and negative Schroeder-phase maskers under test conditions associated with nonlinear cochlear processing. The two maskers were more nearly equal in effectiveness for conditions associated with more linear processing (high levels, hearing-impaired listeners). A number of factors linked to the cochlear amplifier, including possible suppressive effects and level-dependent changes in the phase and magnitude response of effective filtering, may have contributed to these differences." [[71], p. 2316].

The analysis of the phase characteristics of auditory filters was further refined by Oxenham and Dau [44, 45]. They varied the phase curvature of Schroeder-phase complexes by using a scalar multiplier in front of (7.1), very similar to the use of the Schroeder-phase formula by [41]. They concluded that the scaling invariance of filter phase with filter center-frequency, as expected for a set of filters with constant quality factor, might hold for frequencies above 1 kHz, but not for lower frequencies.

Schroeder-phase stimuli have also been used in physiological experiments, which allowed a direct test of the basic hypothesis of the role of peripheral filtering on modulation depth of the waveform, as published in 1985 by Strube. In 2000, Recio and Rhode [52] measured the basilar membrane response in the chinchilla for positive and negative Schroeder-phase stimuli and also for clicks, thus using types of stimuli very similar to the early psychoacoustic studies. They concluded: "The behavior of BM responses to positive and negative Schroeder complexes is consistent with the theoretical analysis performed by Kohlrausch and Sander in 1995, in which the curvature, i.e., the second derivative of the phase versus frequency curve of the BM was used to account for the differences in the response to each of the two Schroeder phases... Hence, phase characteristics of basilar membrane responses to positive Schroeder-phase stimuli show reduced curvatures (relative to the stimulus), and, as a result, peaked waveforms (Kohlrausch and Sander, 1995)" [[52], p. 2296].

Given the relevance of signal phase in Schroeder's work and in hearing research, the first author chose this topic in his contribution to the session "Honoring Manfred R. Schroeder, his contributions and his life in acoustics" at the 4th joint meeting of the Acoustical Society of America and the Acoustical Society of Japan in Honolulu, Hawaii (2006) in his talk: "Schroeder's phase in psychoacoustics." This session was held in the period that the DPI building at the address Bürgerstr. 42–44 was being dismantled and rebuilt for non-academic use. In order to preserve some memories, I (AK) was able to physically remove and to hand over to Schroeder a sign from the institute's parking lot, indicating the place where he and his family used to park their car during their weekend strolls through downtown Göttingen (see Fig. 7.6). The plate had been signed by all speakers of this memorial session.



Fig. 7.6 Photo from the 4th joint ASA and ASJ meeting, Honolulu, 2006, showing the first author in local outfit handing over to Manfred Schroeder a sign that used to indicate the parking spot reserved for goods delivery from the parking lot of the Drittes Physikalische Institut in Göttingen. Photo courtesy of session co-chair Akiro Omoto

7.4 Noise Signals with Non-Gaussian Statistics

In the last part of our chapter we want to demonstrate that the use of the very traditional auditory stimulus known as "noise" has also been influenced by advanced signal processing possibilities. Again, the role of mathematics, in this case of amplitude and envelope statistics and of the amount of intrinsic fluctuations and their spectral composition, is evident and has enabled major steps forward in understanding working principles of the human auditory system. Although the meaning of the term noise is wider in daily use, in hearing sciences, it refers to signals that are inherently random. When we consider for example white Gaussian noise, samples taken from its temporal waveform are randomly distributed according to a Gaussian distribution, and samples taken at subsequent moments in time are uncorrelated. The frequency-domain representation of white Gaussian noise shows a complex spectrum where the real and imaginary parts are also normally distributed.

Noise signals are often subjected to some kind of spectral filtering. Although this influences the spectral envelope of the signal, the Gaussian distributions of the time-domain samples and the complex spectral components are not influenced. However, the correlation across samples taken at different moments in time is influenced. This is reflected in the auto-correlation function. For a white noise signal, the autocorrelation function has a peak at lag zero and is zero at all other lags, in line with the idea that samples are mutually uncorrelated. For a filtered noise, however, there will be correlation across samples which is reflected in the autocorrelation function being unequal to zero at non-zero time lags.

Noise signals have a long history in hearing research because they are relatively easy to generate by analog means. Also, the most often used modifications, spectral filtering and temporal windowing, could be realized using basic analog electronics. Noise stimuli have been used extensively to study auditory masking where noise often serves as a masker. For example, in the early experiments related to critical bandwidth by Hawkins and Stevens [23], white, Gaussian-noise maskers were used to measure the frequency dependence of masked thresholds of a tonal signal. The preference for using white noise signals can be attributed to properties such as uniform energy distribution across time and frequency and the fact that the signals can be described by using only a few parameters.

The inherently stochastic nature of noise has strong implications for its masking behavior. This was demonstrated in studies that employed reproducible noise for which the stochastic uncertainties in the noise are effectively removed. Generally, reproducible noise produces lower masked thresholds than running noise (e.g., [39, 77]).

The probability distribution of noise amplitudes and envelopes can also have an influence on the masking properties of a noise signal. As long as stimuli for hearing research were generated by analog means, noise signals typically had a Gaussian statistic. One early exception to this was so-called multiplied noise (also called multiplication, or regular zero-crossing noise), generated by direct multiplication of a sinusoid and a lowpass noise [21]. It was a very convenient way to generate band-pass noises with tunable center frequency, and these noises were therefore quite useful in spectral masking experiments in which noise maskers with variable center frequencies and steep spectral cutoffs were required (e.g., [46, 47]). The fact that the envelope statistics of this stimulus differed significantly from that of Gaussian noise was only recognized after some time, once it was shown that these properties influence the outcome of listening experiments (see, as an example, [76]). One particularly relevant detail, often overlooked, is that the addition of two multiplied noise signals, for which the two sinusoidal center components are uncorrelated, results in a noise with Gaussian statistics.

With the advance of digital computers, additional noise types have been developed, often with the goal to vary the amount and nature of envelope fluctuations in a controlled way. Transposed stimuli were constructed to realize in the envelope of a high-frequency sinusoidal carrier the same waveform that resulted from auditory filtering and half-wave rectification of noise signals with a low center frequency [73]. "Sparse noise" was generated to have a white noise with a controllable, high amount of envelope fluctuations [25]. The opposite in terms of envelope fluctuations was "low-noise noise," a term coined by Pumplin in 1985 [51]. Due to its resemblance with periodic low-peak signals, we will focus in the remaining part of the chapter on this stimulus.

7.4.1 Low-Noise Noise

7.4.1.1 Definition

In the previous section, we noted that a filtering operation on white Gaussian noise causes the auto-correlation function to change from a delta function at lag zero, to a pattern that reflects predictability of successive time samples of such noise. This predictability is reflected in a smooth development of the envelope of the time-domain noise waveform.[1] The rate of fluctuation in the temporal envelope is proportional to the bandwidth of the filtered noise signal. The spectrum of the envelope has a large DC component and a downward tilting slope that leaves very little spectral power beyond frequencies equal to the bandwidth of the band-pass noise. Interestingly, the *amount* of fluctuation, i.e., the ratio between the AC and the DC part of the envelope, is independent of the bandwidth for Gaussian noise, a property which is reflected in the Rayleigh distribution of the temporal envelope values. Thus, there is an inherently high degree of fluctuation in Gaussian noise.

The inherent fluctuations that are present in Gaussian noise have prompted the development of so-called low-noise noise [51]. This special type of noise has the same spectral envelope as Gaussian noise, but a much lower degree of inherent fluctuations in its temporal envelope, hence the name *low-noise* noise. This stimulus allowed the study of the contribution of envelope fluctuations to auditory masking phenomena by comparing the masking effect of Gaussian and low-noise noise. The first authors to pursue this idea were Hartmann and Pumplin [22].

7.4.1.2 Stimulus Generation

The original means of generating low-noise, such as promoted by [51] was via a special optimization algorithm. First, a band-pass noise was digitally generated in the frequency domain by setting amplitudes in a restricted spectral range to some specific values, e.g., one constant value, and randomizing the phases. Such a noise will approximate all the properties of a band-pass Gaussian noise when the product of duration and bandwidth is sufficiently large. Via a steepest-descent algorithm, the phase spectrum was modified step-by-step in the direction that made the temporal envelope more flat, according to some statistical measure of envelope fluctuation.[2] After a sequence of iterations, a low-noise noise waveform was obtained with a rather flat temporal envelope and the initial amplitude spectrum. Summarizing, the Pumplin's method obtained low-noise noise by modifying the phase spectrum in a special way.

[1] There are alternative ways to determine the envelope of a signal which lead to somewhat different envelopes. We will consider here the Hilbert envelope.

[2] In this case, the normalized fourth moment of the waveform.

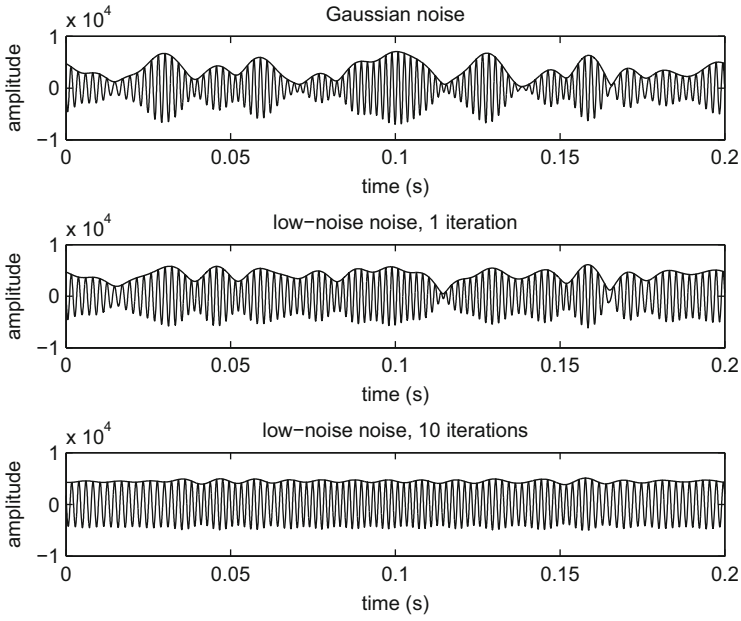


Fig. 7.7 Illustration of the low-noise noise generation. The *top* panel shows the time-domain Gaussian noise at the start of the iterative process, the *middle* panel the low-noise noise after one iteration, the *lower* panel the low-noise noise after 10 iterations. All waveforms are shown with their respective envelopes

Later on, in a publication dedicated to Manfred Schroeder on the occasion of his 70th birthday, several alternative methods of generating low-noise noise were proposed and evaluated by Kohlrausch et al. [33]. We will here describe the method that led to the lowest degree of fluctuation in the temporal envelope. The method consists of an iterative process that is initiated by generating a time-discrete Gaussian band-pass noise. The iterative process then consists of a sequence of straightforward steps.

First, the Hilbert envelope of the noise is calculated. Secondly, the noise waveform is divided by its Hilbert envelope on a sample-by-sample basis in the time domain. For the rare occasions that the Hilbert envelope is equal to zero, the resulting division is set to zero. In the third step, a band-pass filtering is applied to remove the new spectral components outside of the specified band-pass range that were introduced by the division operation in the previous step. By repeating these steps several times, a much flatter envelope is obtained.

After the first two steps, i.e., after calculating the Hilbert envelope and dividing the noise waveform by this envelope, the resulting waveform will have a flat temporal envelope. The spectrum, however, will also be modified considerably. The division by the Hilbert envelope can be seen as a time-domain multiplication by the reciprocal Hilbert envelope. In the frequency domain, this is equivalent to a convolution of the band-pass noise signal with the spectrum of the reciprocal Hilbert envelope. Due to the large DC component present in the Hilbert envelope,

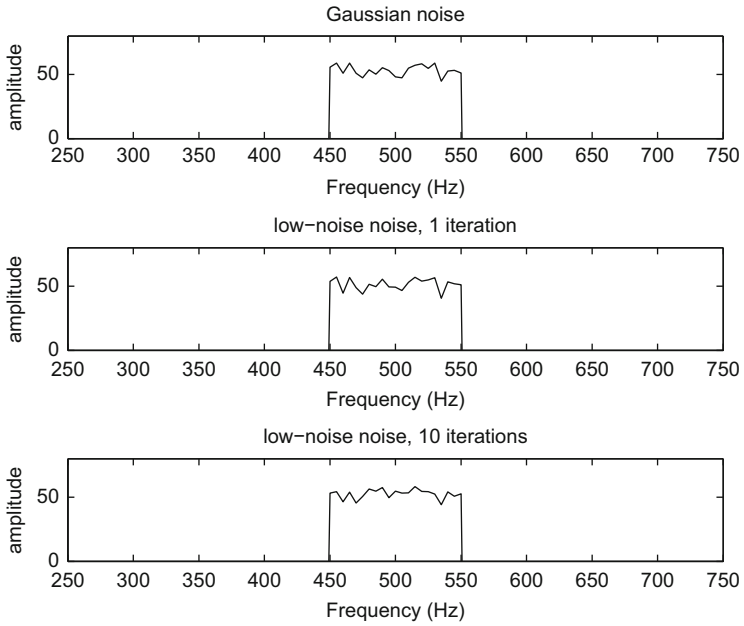


Fig. 7.8 Illustration of the low-noise noise generation. The *top* panel shows the power spectrum of the Gaussian noise at the start of the iterative process, the *middle* panel the spectrum of the low-noise noise after one iteration, the *lower* panel the spectrum of the low-noise noise after 10 iterations

the reciprocal Hilbert envelope will also have a large DC component. Thus, the convolution in the frequency domain will be dominated by this DC component and as a consequence, the spectrum of the band-pass noise will remain largely intact. However, there will be additional, new spectral components that are outside the band-pass range of the original band-pass noise.

Therefore, in the third step, band-pass filtering is applied to remove the new spectral components outside of the specified band-pass range that were introduced by the division operation in the previous step. Considering the argumentation given above, only a relatively small amount of signal power is removed by this operation. Nevertheless, after filtering, the temporal envelope will not be flat anymore. In Fig. 7.7, the temporal waveforms are shown for the original 100-Hz wide Gaussian band-pass noise centered at 500 Hz, that was input to the iterative process (top panel), and after the first iteration of our algorithm (middle panel). As can be seen, the degree of envelope fluctuation is reduced considerably. By repeating the iterative steps 10 times, a much flatter envelope is obtained (lower panel). Convergence is assumed to be obtained due to the DC component in the Hilbert envelope becoming more dominant over the higher spectral components after each iteration.

In Fig. 7.8, the same signals are shown, only now represented in the frequency domain. As can be seen, the original, band-pass Gaussian noise has a uniform spectral envelope. The spectrum of the low-noise noise signal is, even after

Table 7.1 Normalized fourth moment of low-noise noise as a function of the number of iterations

Number of iterations	Normalized fourth moment
0	3.030
1	1.845
2	1.701
4	1.591
6	1.552
8	1.535
10	1.526
Hartmann and Pumplin	1.580

The bottom row gives the value from [22] for comparison

10 iterations, quite similar to the spectrum of the Gaussian signal. There is, however, a tendency for the spectrum to have a somewhat lower level towards the edges of the band-pass range.

As a measure of envelope fluctuation, Table 7.1 shows the normalized fourth moment for different numbers of iterations of our algorithm. The value obtained by [22] is shown at the bottom of the table. As can be seen, already after 6 iterations, we obtain a lower degree of envelope fluctuation than the method of Hartmann and Pumplin. After 10 iterations, the normalized fourth moment is 1.526, close to the theoretical minimum of 1.5 for a sinusoidal signal.

In summary, the iterative method is able to create a low-noise noise by modifying both the phase and the amplitude spectrum. The specific ordering of spectral components in the passband causes the flat envelope that is seen in the lower panel of Fig. 7.7. Due to the very specific arrangement of phase and amplitude values throughout the noise spectrum, any modification of this spectral ordering will affect the flatness of the temporal envelope. In Fig. 7.9, the low-noise noise signal of Fig. 7.7, which was centered at 500 Hz, and had a bandwidth of 100 Hz, is shown after being filtered with a 78-Hz-wide gamma-tone filter centered at 500 Hz. As can be seen, the degree of envelope fluctuation has increased considerably. Since the gamma-tone filter used here is a reasonable first-order approximation of auditory peripheral filtering, this figure demonstrates that the properties that are present in the external stimulus should not be taken to be representative for the manner in which the stimulus is represented within the auditory system (see also Figs. 6 and 7 in [33]).

7.4.2 Role in Hearing Research and Perceptual Insights

As discussed before, Gaussian noise is frequently used as a stimulus in experiments investigating auditory masking. Early experiments by Fletcher [18] used noise signals of various bandwidths to determine detection thresholds of sinusoidal signals centered in the band-pass noise maskers. In these experiments, it was found that only the masker energy that was spectrally close to the sinusoidal target signal contributed to the masking effect of the noise. This led to the concept of the

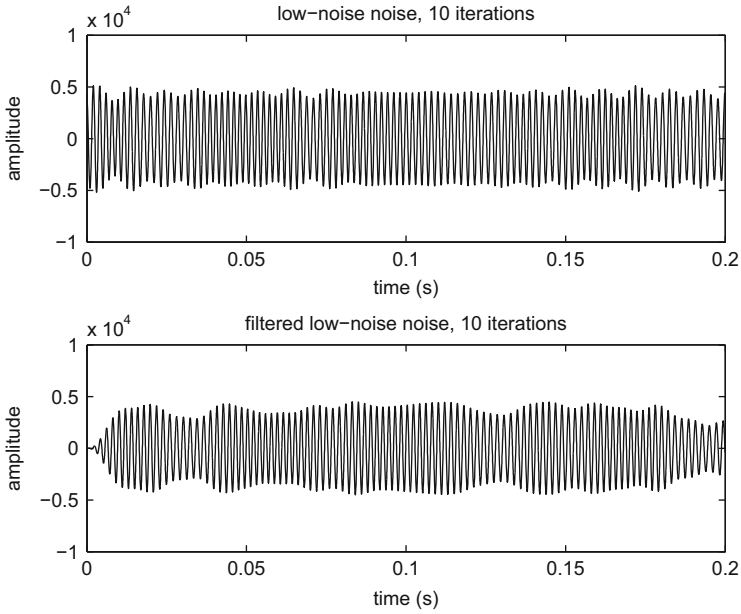


Fig. 7.9 Illustration of the low-noise noise generation. The *top* panel shows a low-noise noise with a bandwidth of 100 Hz after 10 iterations, the *lower* panel shows the same waveform after peripheral filtering with a gamma-tone filter of 78-Hz width

critical band, which indicates the spectral range that contributes to the masking effect on the sinusoidal signal. The integrated intensity of the masker within this range determines the masked threshold.

This purely intensity-based account of masking does not provide insights into the reasons for observing quite different masked thresholds when narrow-band noises or sinusoidal signals of equal level are used as a masker. When the bandwidth of the noise is smaller than the critical bandwidth, there is no difference in the masker intensity within that critical band for both masker types. Thus if overall masker intensity determines masking, thresholds should be the same. Typically, however, thresholds for tonal maskers are about 20 dB lower than for narrowband Gaussian-noise maskers (e.g., [42]).

One of the factors that is believed to contribute to the different masking strengths of these signals is the difference in the inherent envelope fluctuations. A tonal masker has no inherent envelope fluctuations, and the addition of the target tone will introduce a beating pattern that may be an effective cue for detecting the presence of the target. A noise masker, however, will have a high degree of fluctuation of its own. Addition of the sinusoidal signal does not alter the properties of the envelope fluctuations by a significant degree, and therefore, changes in the temporal envelope pattern may be a less salient cue for a noise masker.

Low-noise noise maskers provide an elegant stimulus to verify that the inherent fluctuations in Gaussian noise are an important factor contributing to its strong

masking effect. Such an experiment had been done by Hartmann and Pumplin [22], but the difference in masked thresholds that they found was only 5 dB. This difference is considerably smaller than the 20-dB difference in masking found for Gaussian-noise maskers and tonal signals. A complicating factor may be that the inherent fluctuations in the low-noise noise that was used by Pumplin and Hartmann were still strong enough to cause a significant masking effect. Furthermore, the bandwidth of their low-noise noise stimulus was 100 Hz around a center frequency of 500 Hz. Although such a bandwidth agrees approximately with the estimates of auditory filter bandwidth at this frequency, peripheral filtering may have caused a significant reduction of the flatness of their low-noise noise stimulus, as we demonstrated in Fig. 7.9, where a low-noise noise signal with their spectral properties was filtered with a 1-ERB wide filter.

A more recent experiment by Kohlrausch et al. [33] used low-noise noise created by the iterative method outlined in the previous section resulting in an even lower degree of inherent fluctuation. In addition, the experiment of Kohlrausch et al. measured masked thresholds as a function of center frequency of the low-noise noise masker while keeping the target tone always spectrally centered within the 100-Hz wide noise masker. The highest center frequency in their experiment was 10 kHz, a frequency where the peripheral filter bandwidth is considerably larger than the masker bandwidth. As a result it can be assumed that peripheral filtering will only have a marginal effect on the temporal envelope flatness of the low-noise noise. Thus in these conditions, the difference in masking thresholds between Gaussian noise and low-noise noise should be about the same size as the difference seen for Gaussian-noise and tonal maskers.

One of the results from [33] is shown in Fig. 7.10. As can be seen, the masked thresholds for Gaussian noise (triangles) are constant for center frequencies of 500 Hz and above. This is in line with the fact that auditory filtering does not reduce masker intensity for a 100-Hz wide masker in this frequency range, and that the degree of inherent envelope fluctuations does not vary as a function of center frequency. For low-noise noise (circles), however, we see a clear dependence of thresholds on the center frequency. Although low-noise noise thresholds were lower than Gaussian-noise thresholds already at a center frequency of 1 kHz, for 10 kHz we see a much larger difference of more than 15 dB, which is much more similar to the difference observed for sinusoidal and Gaussian-noise maskers. The higher thresholds for lower frequencies are well in line with the idea that peripheral filtering affects the temporal envelope flatness of low-noise noise.

A variant of the experiment by Kohlrausch et al. [33] investigated the effect of a frequency offset between masker and sinusoidal target signal with the target always higher in frequency than the masker [75]. The addition of the sinusoidal target to the masker band introduces modulations with a rate that is characterized by the frequency difference between target and masker. When the target is centered within the masker, the newly introduced modulations will have a rate comparable to those already present within the masker alone and will therefore be difficult to detect. When the target is sufficiently remote from the masker band, the modulations that will be introduced due to the addition of the target will be of considerably higher

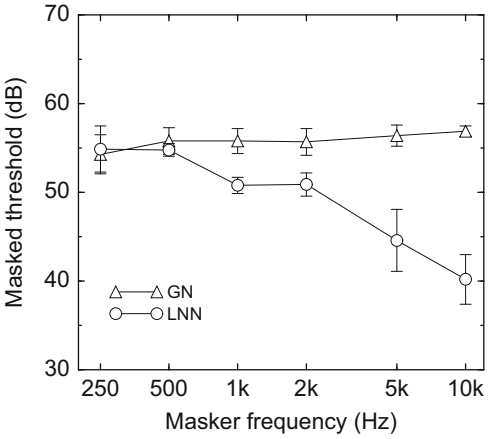


Fig. 7.10 Masked thresholds for 100-Hz wide Gaussian-noise (*triangles*) and low-noise noise maskers (*circles*) as a function center frequency. Reprinted from [33] with permission from Hirzel Verlag and the European Acoustics Association

rate than those already present within the masker and may be much easier to detect. Regarding the perceptual processing of stimuli with various modulation rates, there is evidence for frequency selectivity associated with the processing of temporal envelope fluctuations, which led to the modulation filterbank model proposed by [11]. Low-noise noise is a very suitable stimulus to test some of the non-intuitive consequences of this modulation filter bank concept.

In Fig. 7.11, results of the experiment in [75] are summarized. Masked thresholds are shown for Gaussian noise (dashed lines) and low-noise noise (solid lines) maskers centered at 10 kHz for bandwidths of 10 and 100 Hz, indicated by circles and squares, respectively. The abscissa gives the frequency of the sinusoidal target in terms of the frequency offset of the target relative to the masker center frequency. For an abscissa value of 0 Hz, the target is placed at the center frequency of the noise masker. The values at 0 Hz offset represent the most common on-frequency masking conditions. Note that for the two Gaussian-noise maskers (dashed curves), thresholds increase by about 5 dB, when the masker bandwidth is reduced by a factor of 10 from 100 to 10 Hz. This corresponds to a change of 1.5 dB/oct, exactly the value that is predicted by the notion that for this condition the detection process is dominated by an energy cue in combination with increasing variability of this cue for decreasing degrees of freedom in the masker [3, 20, 74].

The corresponding threshold values for low-noise noise are considerably lower, in line with the results from [31], replotted in Fig. 7.10. Again, thresholds for the 10-Hz masker are higher than those for 100 Hz. This can best be explained by the notion of frequency specific modulation perception. The total modulation energy is the same for the two masker bandwidths, but it is concentrated in a much smaller frequency range (basically, within a 10-Hz range of modulation rates) for the 10-Hz masker than for the 100-Hz masker. Thus, the amount of masker modulation within the lowest modulation filter, ranging up to 10 Hz, will be very high for the 10-Hz masker, making it difficult to detect the extra modulations introduced by adding the target (for this argument, see also Figs. 3 and 9 in [31]).

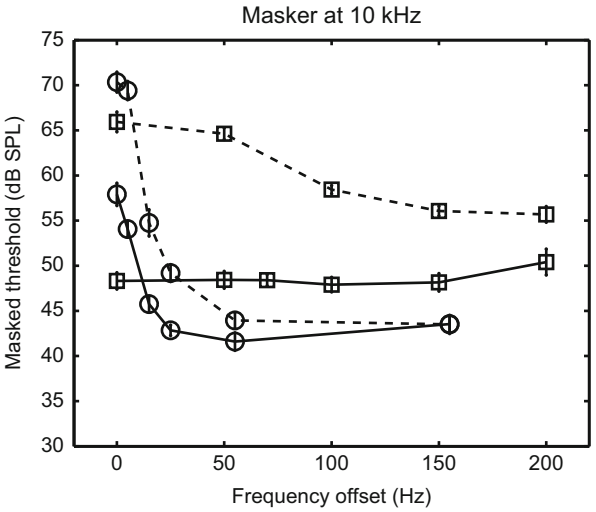


Fig. 7.11 Masked thresholds for Gaussian-noise (*dashed lines*) and low-noise noise maskers (*solid lines*) as a function of frequency offset between the masker and target. Squares show 100-Hz wide maskers and circles 10-Hz wide maskers both presented at 70 dB SPL

When the target frequency offset is increased, very different patterns are observed for the two different masker bandwidths, but also for the two maskers types that have very different spectra of their intrinsic envelope modulations. For 10-Hz maskers (circles), thresholds start to drop sharply once the offset exceeds 10 Hz. This drop is best explained by the introduction of masker-target modulations with a frequency range exceeding that of the masker modulations. These can be detected the easier, the further they are removed from the masker fluctuations. For the largest offset of 150 Hz, the two maskers lead to identical thresholds, an indication that detection is no longer limited by the intrinsic masker fluctuations ("external noise"), but by the hearing system's limited resolution ("internal noise"). For this condition, the target threshold is about 27 dB below the masker level. This value is in excellent agreement with data for a *sinusoidal* masker which amounted to a threshold about 29 dB below the masker level for a sinusoidal masker at 10 kHz and a sinusoidal target 150 Hz above the masker frequency (see Fig. 8 in [32]).

As can be seen from the squares indicating the 100-Hz-wide maskers, the low-noise noise thresholds (squares with solid lines) are roughly independent of frequency offset. They are about 5 dB higher than the values for the 10-Hz masker, so there is a small, but significant remaining influence of the 100-Hz masker. This can be understood by looking at the envelope spectrum. For a 100-Hz wide masker, the dominant rate of internal envelope fluctuations is around 100 Hz, and this corresponds exactly to the frequency difference between upper edge of the masker and the target for an offset of 150 Hz. Thus, even for frequency offsets of 150 Hz, the relatively small amount of masker envelope fluctuation prevents an optimal detection of the target based on temporal cues. In contrast, the Gaussian-noise thresholds for the same bandwidth (squares with dashed lines) show a clear dependence on the frequency offset, and thresholds are generally higher than the low-noise noise thresholds, in line with the idea that the inherent fluctuations in the

Gaussian masker prohibit the detection of the modulations introduced through the addition of the sinusoidal target signal. At larger frequency offsets, however, the modulations introduced by the target signal become higher in rate, and thresholds decrease systematically. This decrease reflects the decrease of the masker envelope spectrum for a Gaussian noise (cf. Fig. 8 in [15]).

One should note that the frequency offsets discussed here are considerably smaller than the peripheral filter bandwidth at the masker frequency of 10 kHz, which amounts to more than 1,000 Hz. Thus the patterns of thresholds observed in Fig. 7.11 cannot reflect peripheral *spectral* filtering. The dependence of thresholds on target frequency offset that is observed here is a reflection of the processing of temporal envelope fluctuations and not of spectral resolution.

7.4.3 Outlook

We have seen that the use of low-noise noise as masker does lead to different thresholds compared to Gaussian noise. This supports the idea that temporal fluctuations in Gaussian noise are a significant factor in auditory masking. Thus, low-noise noise may be an interesting stimulus also in the future to study the contribution of envelope fluctuations to masking.

Low-noise noise may also be of value for acoustical measurement techniques because it is a signal that couples a low crest factor with a continuous spectrum. Whereas in hearing experiments, the bandwidth of low-noise noise is usually limited to, at most, that of one critical bandwidth to prevent peripheral filtering from re-introducing fluctuations in the envelope, for physical measurements this restriction may not exist and wideband low-noise noise may be used to put maximum wideband power into a system that is somehow restricted in its maximum amplitude.

Various studies have applied low-noise noise for a number of different purposes. Dau et al. [15] have used low-noise noise, together with a number of different noise types to study the spectral processing of envelope fluctuations. Due to its flat temporal envelope, low-noise noise has a markedly different envelope spectrum as compared to Gaussian and multiplied noise. The envelope spectral content seems to govern the degree of masking that is observed. Buss et al. [8] have presented low-noise noise in their studies on comodulation masking release, where low-noise-noise provided a masker with little fluctuation but similar bandwidth as a Gaussian-noise masker. Low-noise noise has also been used to measure the minimal level to mask tinnitus [53] because it lacks excessive peaks and troughs. A final application for low-noise noise comes from speech perception research. Healy and Bacon synthesized artificial speech by amplitude modulating bands of low-noise noise and used this type of speech to measure the critical band for speech [24].

7.5 Conclusion

The topics in this chapter are naturally biased towards examples, to which we have ourselves contributed, and which were inspired by ideas and publications of Manfred Schroeder. In this way we hope to demonstrate how influential his way of thinking and his early emphasis on using computers in acoustics has been, using the example of psychoacoustics. The great potential of mathematically well-defined acoustic stimulus types lies in the possibility that they can equally well be used in behavioral as well as in physiological experiments, and that they can serve as input to those types of perception models which allow the processing of arbitrary signal waveforms (e.g., [5–7, 11–14]). This close interplay between psychoacoustics, physiology and modeling is one of the central themes of a conference series, the International Symposia on Hearing, which has been mentioned at many places in this chapter. This series was initiated in 1969 by, among others, Manfred R. Schroeder and Jan Schouten, the founding director of the Institute for Perception Research (IPO) in Eindhoven, where the present authors met each other some 20 years ago. The last of the symposia in which Schroeder participated was held 1988 in Paterswolde near Groningen, and Schroeder, with his love for the Dutch people, language, and country was happy to deliver the after dinner address at the conference social outing (see pp. vi–vii in [17]). This symposium never took place in Göttingen, but it can be seen as a late echo of the psychoacoustic research tradition at the DPI that two editions of these symposia, the 12th in 2000 [4] and the 14th in 2006 [38], were co-organized by scientists who received their initial academic training and were shaped in their scientific interests by Manfred Schroeder.

Acknowledgments The work described in this chapter reflects the close and fruitful cooperations that I (AK) had over the past more than 30 years with friends and colleagues at the DPI in Göttingen and later on in Eindhoven where I was joined at the IPO by SvdP in 1992. We both would like to thank all our colleagues for the creative atmosphere, never-ending curiosity and great fun, which helped to generate some interesting scientific insights. A part of the Göttingen atmosphere was in 1991 exported to Eindhoven, and as can be seen from the reference list, this has become a similarly fruitful period.

References

1. Atal, B.S., Schroeder, M.R.: Predictive coding of speech signals and subjective error criteria. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1978, pp. 573–576 (1978)
2. Blauert, J.: Spatial Hearing. The Psychophysics of Human Sound Localization. The MIT Press, Cambridge, MA (1997)
3. Bos, C.E., de Boer, E.: Masking and discrimination. J. Acoust. Soc. Am. **39**, 708–715 (1966)
4. Breebaart, J., Houtsma, A., Kohlrausch, A., Prijs, V., Schoonhoven, R.: Physiological and Psychophysical Bases of Auditory Function. Shaker Publishers, Maastricht (2001)

5. Breebaart, J., van de Par, S., Kohlrausch, A.: Binaural processing model based on contralateral inhibition. I. Model structure. J. Acoust. Soc. Am. **110**, 1074–1088 (2001)
6. Breebaart, J., van de Par, S., Kohlrausch, A.: Binaural processing model based on contralateral inhibition. II. Predictions as a function of spectral stimulus parameters. J. Acoust. Soc. Am. **110**, 1089–1104 (2001)
7. Breebaart, J., van de Par, S., Kohlrausch, A.: Binaural processing model based on contralateral inhibition. III. Predictions as a function of temporal stimulus parameters. J. Acoust. Soc. Am. **110**, 1105–1117 (2001)
8. Buss, E., Grose, J.H., Hall III, J.W.: Features of across-frequency envelope coherence critical for comodulation masking release. J. Acoust. Soc. Am. **126**, 2455–2466 (2009)
9. Carlyon, R.P., Datta, A.J.: Excitation produced by Schroeder-phase complexes: evidence for fast-acting compression in the auditory system. J. Acoust. Soc. Am. **101**, 3636–3647 (1997)
10. Carlyon, R.P., Datta, A.J.: Masking period patterns of Schroeder-phase complexes: effects of level, number of components, and phase of flanking components. J. Acoust. Soc. Am. **101**, 3648–3657 (1997)
11. Dau, T., Kollmeier, B., Kohlrausch, A.: Modeling auditory processing of amplitude modulation: I. Detection and masking with narrowband carriers. J. Acoust. Soc. Am. **102**, 2892–2905 (1997)
12. Dau, T., Kollmeier, B., Kohlrausch, A.: Modeling auditory processing of amplitude modulation: II. Spectral and temporal integration. J. Acoust. Soc. Am. **102**, 2906–2919 (1997)
13. Dau, T., Püschel, D., Kohlrausch, A.: A quantitative model of the 'effective' signal processing in the auditory system: I. Model structure. J. Acoust. Soc. Am. **99**, 3615–3622 (1996)
14. Dau, T., Püschel, D., Kohlrausch, A.: A quantitative model of the 'effective' signal processing in the auditory system: II. Simulations and measurements. J. Acoust. Soc. Am. **99**, 3623–3631 (1996)
15. Dau, T., Verhey, J., Kohlrausch, A.: Intrinsic envelope fluctuations and modulation-detection thresholds for narrowband noise carriers. J. Acoust. Soc. Am. **106**, 2752–2760 (1999)
16. Duifhuis, H.: Audibility of high harmonics in a periodic pulse. J. Acoust. Soc. Am. **48**, 888–893 (1970)
17. Duifhuis, H., Horst, J.W., Wit, H.P. (eds.): Basic Issues in Hearing. Academic Press, London (1988)
18. Fletcher, H.: Auditory patterns. Rev. Mod. Phys. **12**, 47–65 (1940)
19. Glasberg, B.R., Moore, B.C.J.: Derivation of auditory filter shapes from notched-noise data. Hear. Res. **47**, 103–138 (1990)
20. Green, D.M., Swets, J.A.: Signal Detection Theory and Psychophysics. Wiley, New York, NY (1974). reprinted by Krieger
21. Greenwood, D.D.: Auditory masking and the critical band. J. Acoust. Soc. Am. **33**, 484–502 (1961)
22. Hartmann, W.M., Pumplin, J.: Noise power fluctuation and the masking of sine signals. J. Acoust. Soc. Am. **83**, 2277–2289 (1988)
23. Hawkins, J.E.J., Stevens, S.S.: The masking of pure tones and of speech by white noise. J. Acoust. Soc. Am. **22**, 6–13 (1950)
24. Healy, E.W., Bacon, S.P.: Measuring the critical band for speech. J. Acoust. Soc. Am. **119**, 1083–1091 (2006)
25. Hübner, M., Wiegrebe, L.: The effect of temporal structure on rustling-sound detection in the gleaning bat, Megaderma lyra. J. Comp. Physiol. A **189**, 337–346 (2003)
26. Irino, T., Patterson, R.D.: A time-domain, level-dependent auditory filter: the gammachirp. J. Acoust. Soc. Am. **101**, 412–419 (1997)
27. Johannesma, P.I.M.: The pre-response stimulus ensemble of neurons in the cochlear nucleus. In: Cardozo, B.L., de Boer, E., Plomp, R. (eds.) Proceedings of the 2nd International Symposium on Hearing, pp. 58–69. Institute for Perception Research, Eindhoven (1972)
28. Johnston, J.D.: Transform coding of audio signals using perceptual noise criteria. IEEE J. Selected Areas Commun. **6**, 314–323 (1988)

29. Joris, P.X., van der Heijden, M., Louage, D.H., de Sande, B.V., Kerckhoven, C.V.: Dependence of binaural and cochlear "best delays" on characteristic frequency. In: Pressnitzer, D., de Cheveigné, A., McAdams, S., Collet, L. (eds.) Auditory Signal Processing: Physiology. Psychoacoustics, and Models, pp. 477–483. Springer, New York (2005)
30. Kohlrausch, A.: Masking patterns of harmonic complex tone maskers and the role of the inner ear transfer function. In: Duifhuis, H., Horst, J., Wit, H. (eds.) Basic Issues in Hearing, pp. 339–350. Academic Press, Harcourt Brace Jovanovich. Publ, London (1988)
31. Kohlrausch, A., Fassel, R.: Binaural masking level differences in nonsimultaneous masking. In: Gilkey, R.H., Anderson, T. (eds.) Binaural and Spatial Hearing in Real and Virtual Environments, Chapter 9, pp. 169–190. Lawrence Erlbaum Ass, London (1997)
32. Kohlrausch, A., Fassel, R., Dau, T.: The influence of carrier level and frequency on modulation and beat-detection thresholds for sinusoidal carriers. J. Acoust. Soc. Am. **108**, 723–734 (2000)
33. Kohlrausch, A., Fassel, R., van der Heijden, M., Kortekaas, R., van de Par, S., Oxenham, A.J., Püschel, D.: Detection of tones in low-noise noise: further evidence for the role of envelope fluctuations. Acustica United Acta Acustica **83**, 659–669 (1997)
34. Kohlrausch, A., Houtsma, A.J.M.: Edge pitch of harmonic complex tones. IPO Annual Progress Report **26**, 39–49 (1991)
35. Kohlrausch, A., Houtsma, A.J.M.: Pitch related to spectral edges of broadband signals. Phil. Trans. R. Soc. Lond. B **336**, 375–382 (1992)
36. Kohlrausch, A., Sander, A.: Phase effects in masking related to dispersion in the inner ear. II. Masking period patterns of short targets. J. Acoust. Soc. Am. **97**, 1817–1829 (1995)
37. Kohlrausch, A., van de Par, S.: On the use of specific signal types in hearing research. In: Kurtz, T., Parlitz, U., Kaatze, U. (eds.) Oscillations, Waves, and Interactions. Sixty Years Drittes Physikalisches Institut. A Festschrift, pp. 37–71. Universitätsverlag Göttingen, Göttingen (2007)
38. Kollmeier, A., Klump, G., Hohmann, V., Langemann, U., Mauermann, M., Verhey, J. (eds.): Hearing—From Sensory Processing to Perception. Proceedings of the 14th International Symposium on Hearing. Springer, Berlin (2007)
39. Langhans, A., Kohlrausch, A.: Differences in auditory performance between monaural and diotic conditions. I: Masked thresholds in frozen noise. J. Acoust. Soc. Am. **91**, 3456–3470 (1992)
40. Lentz, J.J., Leek, M.R.: Psychophysical estimates of cochlear phase response: masking by harmonic complexes. J. Assn. Res. Otolaryngol. **2**, 408–422 (2001)
41. Mehrgardt, S., Schroeder, M.R.: Monaural phase effects in masking with multicomponent signals. In: Klinke, R., Hartmann, R. (eds.) Hearing—Physiological Bases and Psychophysics, pp. 289–295. Springer, New York (1983)
42. Moore, C., Alcántara, J.I., Dau, T.: Masking patterns for sinusoidal and narrow-band noise maskers. J. Acoust. Soc. Am. **104**, 1023–1038 (1998)
43. Münkner, S., Kohlrausch, A., Püschel, D.: Influence of fine structure and envelope variability on gap-duration discrimination thresholds. J. Acoust. Soc. Am. **99**, 3126–3137 (1996)
44. Oxenham, A.J., Dau, T.: Reconciling frequency selectivity and phase effects in masking. J. Acoust. Soc. Am. **110**, 1525–1538 (2001)
45. Oxenham, J., Dau, T.: Towards a measure of auditory-filter phase response. J. Acoust. Soc. Am. **110**, 3169–3178 (2001)
46. Patterson, R.D.: Auditory filter shape. J. Acoust. Soc. Am. **55**, 802–809 (1974)
47. Patterson, R.D.: Auditory filter shapes derived with noise stimuli. J. Acoust. Soc. Am. **59**, 640–654 (1976)
48. Patterson, R.D., Allerhand, M.H., Giguère, C.: Time-domain modelling of peripheral auditory processing: a modular architecture and a software platform. J. Acoust. Soc. Am. **98**, 1890–1894 (1995)
49. Patterson, R.D., Nimmo-Smith, I., Weber, D.L., Milroy, R.: The deterioration of hearing with age: frequency selectivity, the critical ratio, the audiogram, and speech threshold. J. Acoust. Soc. Am. **72**, 1788–1803 (1982)

50. Pierce, J.R.: Some work on hearing. Am. Scientist **48**, 40–45 (1960)
51. Pumplin, J.: Low-noise noise. J. Acoust. Soc. Am. **78**, 100–104 (1985)
52. Recio, A., Rhode, W.S.: Basilar membrane responses to broadband stimuli. J. Acoust. Soc. Am. **108**, 2281–2298 (2000)
53. Schneider, P., Andermann, M., Wengenroth, M., Goebel, R., Flor, H., Rupp, A., Diesch, E.: Reduced volume of Heschl's gyrus in tinnitus. Neuroimage **45**, 927–927 (2009)
54. Schroeder, M.R.: New results concerning monaural phase sensitivity. J. Acoust. Soc. Am. **31**, 1579 (1959)
55. Schroeder, M.R.: Vocoders: analysis and synthesis speech. Proc. IEEE **54**, 720–734 (1966)
56. Schroeder, M.R.: Relation between critical bands in hearing and the phase characteristics of the cubic difference tones. J. Acoust. Soc. Am. **46**, 1488–1492 (1969)
57. Schroeder, M.R.: Synthesis of low-peak-factor signals and binary sequences with low autocorrelation. IEEE Transact. Inf. Theor. **16**, 85–89 (1970)
58. Schroeder, M.R.: An integrable model for the basilar membrane. J. Acoust. Soc. Am. **53**, 429–434 (1973)
59. Schroeder, M.R.: Models of hearing. Proc. IEEE **63**, 1332–1350 (1975)
60. Schroeder, M.R.: New viewpoints in binaural interaction. In: Evans, E.F., Wilson, J.P. (eds.) Psychophysics and Physiology of Hearing, pp. 455–456. Academic Press, New York (1977)
61. Schroeder, M.R., Atal, B.S.: Computer simulation of sound transmission in rooms. IEEE Int. Convention Record **11**, 150–155 (1963)
62. Schroeder, M.R., Atal, B.S., Hall, J.: Optimizing digital speech coders by exploiting masking properties of the human ear. J. Acoust. Soc. Am. **66**, 1647–1652 (1979)
63. Schroeder, M.R., Atal, B.S., Sessler, G.M., West, J.E.: Acoustical measurements in Philharmonic Hall (New York). J. Acoust. Soc. Am. **40**, 434–440 (1966)
64. Schroeder, M.R., Gottlob, D., Siebrasse, K.F.: Comparative study of European concert halls. J. Acoust. Soc. Am. **56**, 1195–1201 (1974)
65. Schroeder, M.R., Hall, J.L.: Model for mechanical to neural transduction in the auditory receptor. J. Acoust. Soc. Am. **55**, 1055–1060 (1974)
66. Schroeder, M.R., Strube, H.W.: Flat-spectrum speech. J. Acoust. Soc. Am. **79**, 1580–1583 (1986)
67. Shamma, S.A., Shen, N., Gopalaswamy, P.: Stereausis: Binaural processing without neural delays. J. Acoust. Soc. Am. **86**, 989–1006 (1989)
68. Smith. Der Einfluß des Vorzeichens des Momentanfrequenzverlaufes von Maskierern auf die Detektion von reinen Tönen (The influence of the sign of the change in the instantaneous frequency of a masker on the detectability of pure tones). Master's Thesis, Georg-August-Universität Göttingen (1984)
69. Smith, K., Sieben, U., Kohlrausch, A., Schroeder, M.R.: Phase effects in masking related to dispersion in the inner ear. J. Acoust. Soc. Am. **80**, 1631–1637 (1986)
70. Strube, H.W.: A computationally efficient basilar-membrane model. Acustica **58**, 207–214 (1985)
71. Summers, V.: Effects of hearing impairment and presentation level on masking period patterns for Schroeder-phase harmonic complexes. J. Acoust. Soc. Am. **108**, 2307–2317 (2000)
72. Summers, V., Leek, M.R.: Masking of tones and speech by Schroeder-phase harmonic complexes in normally hearing and hearing-impaired listeners. Hear. Res. **118**, 139–150 (1998)
73. van de Par, S., Kohlrausch, A.: A new approach to comparing binaural masking level differences at low and high frequencies. J. Acoust. Soc. Am. **101**, 1671–1680 (1997)
74. van de Par, S., Kohlrausch, A.: Diotic and dichotic detection using multiplied-noise maskers. J. Acoust. Soc. Am. **103**, 2100–2110 (1998)
75. van de Par, S., Kohlrausch, A.: The role of intrinsic masker fluctuations on the spectral spread of masking. In *Forum Acusticum 2005, Budapest*, pp. 1635–1640, European Acoustics Association (2005)

76. van der Heijden, M., Kohlrausch, A.: The role of envelope fluctuations in spectral masking. J. Acoust. Soc. Am. **97**, 1800–1807 (1995)
77. von Klitzing, R., Kohlrausch, A.: Effect of masker level on overshoot in running- and frozen-noise maskers. J. Acoust. Soc. Am. **95**, 2182–2201 (1994)

***

Biography



Armin Kohlrausch studied physics at the University of Göttingen (Germany) and specialized in acoustics. He received his master degree in 1980 and his Ph.D. in 1984, both on perceptual aspects of sound. From 1985 until 1990 he worked at the Drittes Physikalisches Institut at the University of Göttingen, being responsible for research and teaching in the fields psychoacoustics and room acoustics. In 1991 he joined the Philips Research Laboratories in Eindhoven and worked in the speech and hearing group of the Institute for Perception Research (IPO), a joint venture between Philips and the Eindhoven University of Technology (TU/e). Since 1998, he combines his work at Philips Research Laboratories with a Professor position for auditory and multisensory perception at the TU/e. In 2004 he was appointed research fellow of Philips Research. He is a member of a great number of scientific societies, both in Europe and in the USA. Since 1998 he is a Fellow of the Acoustical Society of America and has served as Associate Editor for the Journal of the Acoustical Society of America, for the Journal of the Association for Research in Otolaryngology and for Acta Acustica United with Acustica. His main scientific interest is the experimental study and modeling of auditory and multisensory perception in humans and the transfer of this knowledge to industrial applications.



Steven van de Par studied physics at the Eindhoven University of Technology, Eindhoven, The Netherlands, and received the Ph.D. degree in 1998 from the Eindhoven University of Technology, on a topic related to binaural hearing. As a Postdoctoral Researcher at the Eindhoven University of Technology, he studied auditory–visual interaction and was a Guest Researcher at the University of Connecticut Health Center. In early 2000, he joined Philips Research, Eindhoven, to do applied research in auditory and multisensory perception, low-bit-rate audio coding and music information retrieval. Since April 2010 he holds a professor position in acoustics at the University of Oldenburg, Germany, with a research

focus on the fundamentals of auditory perception and its application to virtual acoustics, vehicle acoustics, and digital signal processing. He has published various papers on binaural auditory perception, auditory–visual synchrony perception, audio coding, and computational auditory scene analysis.

Chapter 8
Neurally Based Acoustic and Visual Design

Yoichi Ando and Peter Cariani

Abstract Manfred Schroeder's ideas concerning acoustics, auditory percepts and preferences, and acoustic design had profound influences on the development of Yoichi Ando's theory of architectural acoustics. Building on Schroeder's theoretical frameworks, over subsequent decades Ando formulated a systematic theory of architectural acoustics design that incorporates acoustics (the structure of sound and how it propagates through an enclosed space), psychoacoustical models of auditory percepts and listener preferences (what we hear, which perceptual attributes are most important for the design of enclosed spaces, what we like to hear, which percepts are most important to our overall satisfaction, how individualized are these preferences), and strategies for optimal design (how a design process can harness psychoacoustical knowledge in order to optimize listener preferences).

In the last two decades, Ando's theory has taken "a neural turn" in which monaural and binaural percepts are grounded respectively in putative central auditory autocorrelation and cross-correlation representations. Experimentally, Ando and co-workers have identified some observable neural correlates of relevant auditory percepts and preferences (e.g., EEG and MEG response latency patterns, spatial extent and temporal persistence of alpha rhythms, hemispheric lateralizations). In theory, the identification of neurophysiological signs of listener satisfaction permits neurally driven design processes that optimize acoustics such that neural processes responsible for listener satisfaction are fulfilled. Finally, the same battery of psychophysical methods, correlation-based representations, and

Note from Peter Cariani: I stood in for Yoichi Ando at the Manfred R. Schroeder Memorial Session at the 116th ASA Meeting, Seattle, USA, May 2011 because, due to illness, he had to be hospitalized in Japan a few days before. I also served as Guest Editor of the book, *Auditory and Visual Sensations*, Springer-Verlag, NY, 2009, and I am honored to contribute here in a similar capacity. In reading this chapter, unless otherwise noted, all first-person perspectives, singular and plural, refer to Ando and his colleagues and not to me.

P. Cariani (✉)
Hearing Research Center, Boston University, 44 Cummington Mall, Boston, MA 02215, USA
e-mail: cariani@bu.edu

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_8,
© Springer International Publishing Switzerland 2015

neurophysiological experiments has been applied to problems of visual percepts and preferences. The visual results suggest deep similarities with auditory percepts. These many considerations lead to an integrative theory of spatial and temporal design.

8.1 Introduction

Manfred Schroeder had an early and profound influence on the development of my theory of architectural acoustics, and we have maintained our friendship and scientific exchange of ideas throughout our lives.

Our first approach to acoustic design was based on acoustics, psychophysics, and the psychology of preferences, and was inspired in part by Manfred's 1966 assessment of the field of architectural acoustics [1]. This early research program drew from sources outside the listener's nervous system, external acoustical measurements, cochlear biophysics, and overt psychoacoustical judgments, to model acoustics, percepts, and preferences.

Rational design of sensory experience is possible, given a theory of the stimulus, the percepts it produces, and listener preferences for particular stimulus parameters. In acoustics, adaptive methods can be used to optimize opera house and concert hall sound fields, musical instruments, performances, and real time speech recognitions. We employed genetic algorithms (GAs) to design concert halls by using acoustical–psychophysical models to predict what listeners would hear at each seat location, and then optimizing the shape of the hall to best satisfy listener preferences for what they would hear.

In more recent years, our theory has moved inside the listener, grounding acoustic design in terms of brain processes that subserve auditory percepts and preferences. Neural response correlates associated with auditory qualities related to central autocorrelation functions (ACFs) (temporal sensations, such as pitch, timbre, loudness) and those related to interaural cross-correlation functions (IACFs) (spatial sensations) were found to be dominant in one or another cerebral hemisphere. Features of these two central correlation functions describe primary auditory qualities well. The observable neural correlates of subjective preferences and annoyances involve the temporal persistence and spatial extent of electroencephalograms (EEG) and magnetoencephalograms (MEG) alpha rhythms in cortical regions.

Most recently the theory has been applied to visual sensations. Analogous percepts, preferences, and correlation-based representations have been explored for a number of visual patterns (flickers, oscillating objects, and textures). Subjective scale values of listener preferences for sounds, sound fields, and visual patterns all show the same cortical alpha rhythm persistence and extent.

Thus, the research program that was originally outlined by Manfred Schroeder has been greatly advanced, and even expanded in its scope to encompass brain processes and aspects of vision.

This chapter discusses in turn

- People (friendship and scientific collaborations of Manfred and Yoichi).
- Acoustics (how sound propagates through an enclosed space).
- Percepts (modeling of listener percepts, what we hear).
- Preferences (modeling of listener preferences for different percepts, what we like to hear).
- Design (combining acoustics, auditory signal-processing models, psychophysics, psychology of auditory preferences, and architectural constraints to formulate a rational theory of architectural acoustics design, what we need to know to design a building for the purpose of listening to sounds).
- Auditory signal processing (the modeling of signal processing in the auditory system that gives rise to percepts, how we hear, or how acoustics produces what we hear).
- Neural observables (the behavior of neural observables that are associated with percepts and preferences, how we can venture inside the listener's brain to bypass overt judgments to directly see signs of the underlying neural processes involved in the percepts and preferences).
- Vision (the application of the general theory and its methods to the visual modality, what we see and what we like to see).

By necessity, only the essentials of Ando's acoustical theory can be presented here. For the full methodological and empirical details, as well as the application of the theory to visual attributes and preferences, interested readers should consult his most recent book, *Auditory and Visual Sensations* [2] and the original papers in which they were reported.

8.2 People: Manfred R. Schroeder and Yoichi Ando

In his overview of the problem of architectural acoustics, Manfred Schroeder [1] cogently argued that persistent uncertainties in the acoustical design of concert halls resulted in highly uneven results, with both outstanding successes, such as La Grande Salle in Montreal and the Music Pavilion in Los Angeles, and deficiencies, such as London's Royal Festival Hall (1951), New York's Philharmonic Hall (1962), and Berlin's new Philharmonie (1963). Schroeder attributed this inconsistency in the acoustical quality of concert halls, especially large ones, to "an insufficient understanding of the important factors that make for good concert hall acoustics:" the physical (how sound is propagated to the listener), the psychoacoustical (what we hear), and aesthetic (what we prefer to hear).

These three sets of factors that describe sound, percepts, and preferences are the fundamental elements of a unified and comprehensive theory of architectural acoustics. Together with theories of materials and building construction, these form a rational basis for the design of the acoustics-related aspects of buildings.

The immediate problems at that time involved developing better acoustical and psychophysical measures associated with sound reverberation and diffusion.

I first visited Manfred Schroeder for one month in summer of 1971 at the Third Physics Institute, Göttingen, with the intent of investigating aspects of spatial hearing most relevant to the design of concert halls. There, Peter Damaske [3] had investigated subjective diffuseness (SD), and was working to explain the degree of SD of sound fields. He and I worked together to develop methods for measuring degree of diffusion by calculating the magnitude of the interaural cross-correlation function (IACC) [4]. Schroeder and co-workers [5] measured various physical, psychoacoustic, and aesthetic variables at a number of existing concert halls in Europe, and found the IACC to be most effective and consensus factor for sound field preference.

I returned again and again to Göttingen to work out different aspects of the whole theory. During the period of 1975–1977, as an Alexander von Humboldt Fellow [6], I worked with Schroeder's group in Göttingen on modeling subjective preference. In August 1979, we investigated effects of multiple early reflections of the sound field on subjective preference [7]. In 1983 on another visit, we developed a method to quantitatively estimate scaled subjective preference values for each seat of a prospective concert hall design [8]. Soon after, my first book, Concert Hall Acoustics (1985), with Manfred Schroeder's preface, was published.

During the period of 1979–2003, my wife Keiko and I stayed several times at Manfred and Anny Schroeder's house to prepare manuscripts of books [2, 9]. They visited us in Kobe for dinner together after he delivered his invited lectures at the Acoustical Society of Japan, Nagano, and Kobe University (Fig. 8.1). In designing Uhara Hall in Kobe [10], we applied his quadratic residue sequence to the design of the front part of the ceiling to decrease IACC in order to improve listener satisfaction. After the hall was constructed in 1992, Manfred gave a lecture on fractals and chaos in the hall's form and their consequences for the resulting acoustics.

In summer 1995, Manfred was invited to give the keynote lecture at the International Symposium on Music and Concert Hall Acoustics (MCHA), which was held at Kirishima International Music Hall (Miyama Conceru) [59]. Manfred and Anny Schroeder, Peter and Christiane Damaske and Keiko and I all climbed up the strato-volcano Ohachi, near Kirishma. The climb was made memorable by strong sudden winds we encountered on top, which meant that we had to help each other get down the mountain. Our friendship and collaboration continued again at the Third Institute of Physics Institute at Gottingen, and Manfred's many influences have stayed with me even up to the present time.

8.3 Acoustics, Percepts, and Preferences

For a theory of architectural acoustics, one needs to understand how sound is propagated within enclosures in order to predict the sound that will arrive at the listener's ears. Next, one needs to know what the listener hears given the sound presented to the ears, and also what the listener prefers to hear given all possible



Fig. 8.1 Manfred and Anny Schroeder at Zemtsuji-Temple in Nagano, Japan, 1979

choices of listening conditions. Thus theories of acoustics, models of auditory perception, and the psychology of listener preferences are all essential components for a rational theory of architectural acoustics. To these one must also add a theory of architectural design and construction.

8.3.1 Major Acoustical Factors for Concert Hall Design

There are four major acoustical factors, with their corresponding auditory perceptual attributes, that are most important in listener satisfaction in concert hall settings. These are the binaural listening level (LL), the delay time of the first sound reflections (Δt_1), the delay associated with later reverberations (T_{sub}), and the magnitude of the IACC function. The first three factors were appreciated relatively early on in architectual acoustics [11, 12], but the last, binaural factor was a more recent discovery [4, 5, 13–16].

These acoustical factors and perceptual attributes correspond to features in two general temporal correlation representations (Fig. 8.2): the monaural ACF and the binaural IACF. Many more, perhaps all, auditory percepts can be characterized in terms of these two functions, be they descriptions of the acoustical waveform or of central neural representations that subserve auditory percepts. The features and their corresponding auditory attributes are discussed more fully in Sect. 8.5.

In order to systematically characterize these acoustical factors and their perception, we used a sound simulation system in an anechoic enclosure with seven speakers arranged around the listener. Four separate, independent delays could be fed to subsets of the speakers [2, pp. 23–24, 1, 5, 17]. With such a system, we could

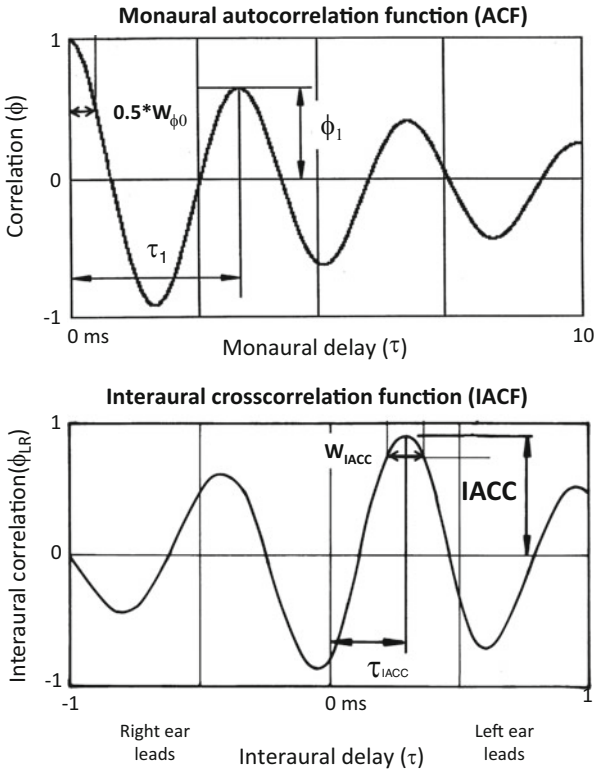


Fig. 8.2 Autocorrelation and cross-correlation functions and their respective features. *Top*. Normalized monaural short-time autocorrelation function (ACF), showing features (respective perceptual correlates in parentheses): time lag of ACF maximum τ_1 (pitch), height of ACF maximum ϕ_1 (pitch strength, regularity), and half-width of ACF zero-lag peak $W_{\phi 0}$ (spectral tilt). *Bottom*. Normalized interaural cross-correlation function (IACF), with respective binaural features: time lag (interaural time difference) of IACF maximum τ_{IACC} (ITD, azimuth), magnitude of the interaural correlation IACC (subjective diffuseness), and width of the IACC peak W_{IACC} (apparent source width ASW)

independently control the delays in the sound field that correspond to apparent right-left position and source width, first reflection time, and later reflection time. Listener preferences were assessed using paired comparison tests.

The four orthogonal and perceptually distinct factors of the sound field most important for concert hall design are:

1. *Binaural listening level* (LL). The LL is the averaged sound pressure level (SPL) at the two ear entrances, in dBA. The corresponding perceptual attribute is loudness. Most listeners prefer listening levels in the 70–90 dBA range, with about 60 % of listener preferences in the 80–85 dBA range [2].

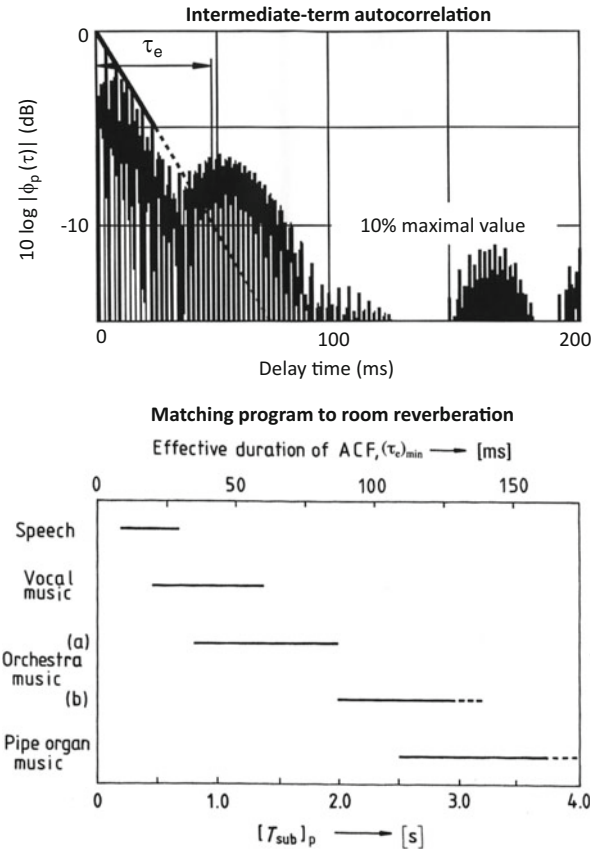


Fig. 8.3 Effective duration, τ_e. *Top*. The computation of effective duration from the extrapolated 90 % decay of the envelope of the intermediate-term ACF of a signal. *Bottom*. Ranges of minimum effective duration $(\tau_e)_{min}$, and the matched, optimal subsequent reverberation times, T_{sub}, for different types of program material

2. *Delay time of the first reflection* after the direct sound (Δt_1). Acoustically, this is the delay (in s) between the arrival of the direct sound and its arrival via the first reflection path at the listener's ears. The first reflection is the earliest and strongest reflection of the generated sound, usually from the stage and its immediate surrounds.

The preferred first reflection time varies with the effective duration of the program material. Effective duration (τ_e) is a measure of the temporal coherence of a sound, the persistence of its regularity, or how fast the sound pattern changes with time. Effective duration can be measured by how fast the ACF decays with increasing time lag (Fig. 8.3, top). Here, effective duration is defined as the delay at which the ACF of the signal has decayed to 10 % of its initial, maximal value.

For a piece of music or speech, the effective duration of the whole is determined by the rate of change of its fastest portion [minimum effective duration $(\tau_e)_{min}$, Fig. 8.3, bottom]. Speech changes its pattern rapidly and has short effective

durations (typically 10–15 ms), whereas slow music with sustained notes typically has much longer durations (typically 120 ms). Fast music lies between these two extremes. Listeners prefer the first reflection time to be roughly equal to the effective duration of the program material presented, when the total amplitude of reflections $A = 1.0$ [2, pp. 26–29, 6, 7]. When first reflection times are long compared to this effective duration, echoes may be perceived and the received sounds suffer in clarity. As a consequence, speech in halls with long first reflection times suffers in intelligibility. When first reflection times are short, the delays are perceived as a change in timbre (tone coloration), and when they are very short, they can result in spatial directional effects.

Thus, ideally one wants to match the program material to the specific concert hall in which it is to be played, and vice versa. In an architectural acoustics utopia, each concert hall would have an acoustician on call who could recommend which sound materials would sound best in that hall. How one performs a piece can also be adapted to the concert hall in which it is performed. We have proposed a method of controlling minimum effective duration $(\tau_e)_{min}$ in performance of vocal music that blends the sound source and a given concert hall [9]. If vibrato is introduced during singing, for example, it decreases this value, blending the sound field with a short reverberation time [18].

One can also design different types of concert halls with different kinds of music and speech in mind. Some concert halls are best suited for faster music, such as Mozart's Symphony No. 41, Stravinsky's Le Sacre du Printemps, and Arnold's Sinfonietta, whereas other halls with longer reverberation times would be better suited for slower, more resonant music, such as Brahms' Symphony No. 4, Buckner's Symphony No. 7, and Bach's works for pipe organ.

3. *Subsequent reverberation time* after the first reflection (T_{sub}). Subsequent reverberation time is the decay rate of later reverberations. Sabine's formula [19] estimated this time as a function of room dimensions (volume and surface area) and the average absorption coefficient of its interior surfaces. Late reverberations contribute to the perceived sense of envelopment and the liveliness of the hall. Late reverberations can also be quantified using the effective duration of signals recorded in the sound field. The preferred reverberation time, which is equivalent to that defined by Sabine, is expressed approximately by $[T_{sub}]_p \approx 23(\tau_e)_{min}$. For "Water Music" by Handel, which has a minimal effective duration $\tau_e \sim 63$ ms, listeners preferred subsequent reverberation times between 0 and 4 s, with 45 % preferring delays between 1 and 2 s [2]. A flat frequency response for reverberations is also preferred.
4. *Magnitude of IACC*. This is the magnitude of normalized IACF $\phi_{l,r}(\tau)$, and it indicates the degree of directional dispersion (correlation) of the sound field across the two ears. Highly directional compact sound sources produce maximal, unity-valued IACCs, whereas highly diffuse, spatially distributed sound sources produce minimal, zero-valued IACCs. Perceptually, the IACC covaries with

apparent source width (ASW) and contributes to the sense of envelopment. Listeners in concert hall contexts uniformly prefer minimal values for IACC. This contrasts with the other three parameters, in which there can be considerable variation in individual listener preferences, which follow a U-shaped function with a maximal, optimal value.

For a rational, well-specified theory of acoustics design, one needs to be able to predict listener percepts and preferences given the sound stimulus (program material) and its modification by a particular enclosure (sound field). In the spirit of Manfred Schroeder's research program [1, 20] we developed correlation-based psychoacoustic models for these and other auditory percepts that serve this purpose [2]. Each auditory percept is related to a specific feature of either an internal monaural autocorrelation or binaural cross-correlation representation. This systematic theory of auditory percepts, along with their presumed neural signal-processing operations and their observable neurophysiological correlates, is outlined below in Sect. 8.4. The observable neural correlates of preferences are discussed in Sect. 8.5.

8.3.2 The Centrality of Listener Preferences for Theories of Design

There were several reasons that we initially got into the investigation of subjective preference. Preferences are evaluative judgments made by the listener that reflect particular situations or conditions that the listener chooses over other alternatives. A listener's preferences thus guide his or her actions, and serve as indicators of the degree of satisfaction. Preferences are estimated quantitatively using systematic, paired comparison tests that, through the law of comparative judgment, permit numerical, metrical scale values to be obtained.

Subjective preferences are the most fundamental aspects of an acoustic theory of design because they embody the desires of the listener. In biological terms, preferences are the goal states of the organism, the most primitive response that steers behavior. In the nervous systems of every higher organism are such embedded goals and drives. Preferences guide the organism in the direction of maintaining life, toward taking actions that enhance self-preservation and long-term survival. As such, in every brain there are structures and neural processes that realize preferences by steering the behavior of the organism such that preferred conditions are more reliably achieved. Being an essential, fundamental part of the functional organization of the nervous system, it is possible for us to observe brain activities that are related directly to preferences and their influences on behavior.

Preferences are deeply related to an individual's aesthetic sense. More than any other aspect of ourselves, the set of all our preferences, what we prefer most, is what gives each of us our individual unique personalities. Because we are always striving

to realize our preferences, sometimes in new ways, our preferences provide a constant motivational source of new creations.

8.3.3 *Theory of Multidimensional Listener Preferences*

We have listed above the four major factors that matter most for concert hall listening (Figs. 8.3 and 8.4). Each is a separate perceptual dimension with its associated preference curves [5]. In order to optimize the design of an enclosure for listening, we need an overall preference measure that takes the individual preference dimensions and tradeoffs between them into account (see Manfred's discussion of the design problem in [21, pp. 74–79]). Such an integrative model of preference clearly leads to comprehensive criteria for achieving the optimal design of concert halls [8–10, 22].

The integrated preference model takes the form of weighted linear addition of the four normalized preference scale values (Fig. 8.4). The scale values were obtained using paired comparison tests rather than absolute preference magnitude estimation. Paired comparison tests have the advantage that they are simpler and allow relative judgments to be made by relatively inexperienced subjects, thus widening both the pool of people who can usefully participate and the range of acoustic parameters that can be studied.

We found that the four different acoustic factors are in fact independent and are not affected by changes in each other. In a given individual listener, there appear to be no natural groupings or interactions of preferences across these four dimensions [2]. Also, results of such a number of subjective preference tests indicate that the units of scale values (preferences for individual parameters S_i) are almost constant, so that we may add scale values to obtain the total scale value (overall preference S) for a sound field [8]. Thus $S = S_1 + S_2 + S_3 + S_4$.

One can transform and normalize the four preference scale values so that each is an inverted U-shaped curve with a zero value at the most preferred condition (designated by subscripted p). Data plots and curve fits can be found in [2, Chap. 3]. The individual normalized preference factors x_i are given in Table 8.1.

Table 8.1 Preference factors of individual auditory attributes and their relative weights in overall preference

Perceptual attribute on which the preference factor x_i is based	Normalized preference factor x_i[a]	Weighting[b]	
		α_i ($x_i < 0$)	α_i ($x_i > 0$)
Binaural listening level, LL	$x_1 = 20 \log P - 20 \log[P]_p$	0.07	0.04
First reflection time, Δt_1	$x_2 = \log(\Delta t_1/[\Delta t_1]_p)$	1.42	1.11
Subsequent reverberation time, T_{sub}	$x_3 = \log(T_{sub}/[T_{sub}]_p)$	$0.45 + 0.75A$	$2.36 - 0.42A$
Interaural correlation, IACC	$x_4 = \text{IACC}$	1.45	–

[a] P is sound pressure in dBA, p subscripts indicate listener-preferred values
[b] See text below

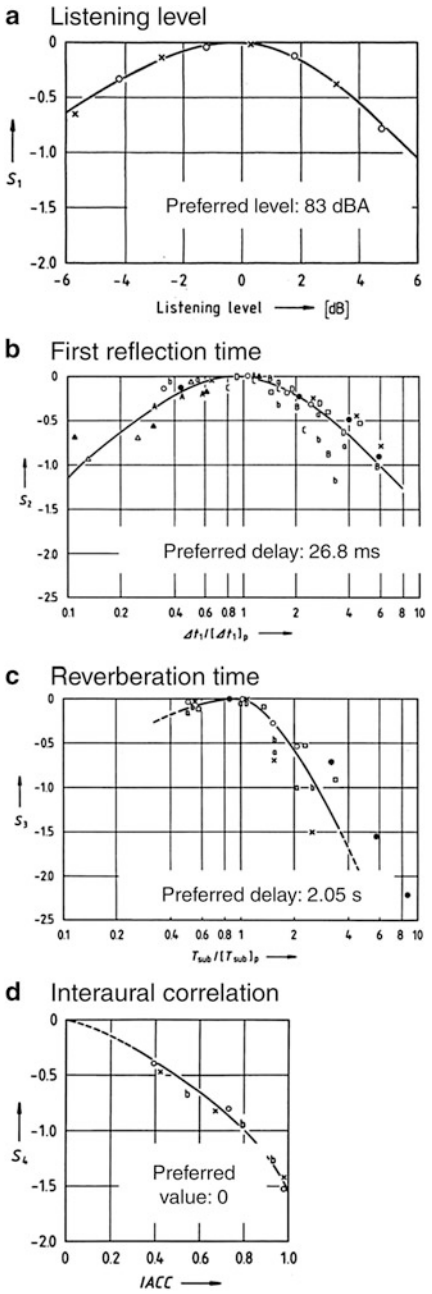


Fig. 8.4 Measured scale preference values of the four major acoustical factors. (**a**) Listening level (LL). (**b**) First reflection time (Δt_1). (**c**) Subsequent reverberation time (T_{sub}). (**d**) Interaural correlation (IACC). Estimated average preferred values were 83 dBA (LL), 26.8 ms (Δt_1), 2.05 s (T_{sub}), and 0 (IACC)

Overall preference S is then the weighted sum of preferences, x_1 through x_4, for the individual dimensions S_i (principle of superposition of preferences).

$$S = S_1 + S_2 + S_3 + S_4 = -\alpha_1 |x_1|^{3/2} - \alpha_2 |x_2|^{3/2} - \alpha_3 |x_3|^{3/2} - \alpha_4 |x_4|^{3/2}$$

where x_i are the normalized individual preference factors and α_i are the weighting coefficients for each factor x_i, below and above its optimal value (0), that indicate their relative importance in determining overall preference. The weighting of T_{sub}, α_3, depends in part on the total pressure amplitude of reflections in the room, parameter A in Table 8.1.

Thus, the scale values of preference have been formulated approximately in terms of the 3/2 powers of the normalized objective parameters, expressed as logarithms of parameters, x_1, x_2, and x_3. It is remarkable that the spatial binaural parameter of IACC, $x_4 = \text{IACC}$, is expressed in terms of the 3/2 power of its real value rather than its logarithm, which indicates a much greater relative contribution to overall preference than for the other parameters.

The scale values are not greatly changed in the neighborhood of the most preferred conditions, but decrease rapidly outside of this range. Because the experiments were conducted to find the optimal conditions, this theory holds in the range near the preferred conditions tested for the four factors.

8.4 Rational Theory of Acoustic Design

Once one has developed models for the acoustics of enclosed spaces, the percepts associated with sound fields, and listener preferences for particular values and combinations of those percepts, then it becomes possible to rationalize and automate the design of concert halls. Given particular hall dimensions and geometry, it then is possible to compute the sound fields present for every seat in the house, and subsequently, to compute the values for the four relevant acoustic parameters, and then given a set of listener preferences, to estimate the level of listener satisfaction that will be obtained.

Conversely, if one has a given concert hall, and one knows one's own preferences in quantitative terms, then it is possible to find the seat in the house that will provide the best listening experience, and one could even sell tickets on this basis. We introduced a special facility for seat selection at the Kirishima International Concert Hall in 1994 that tested each listener's own subjective preference and matched these with hall locations in order to make individualized seating recommendations.

8.4.1 Acoustic Design Using GAs

Even if one has fully developed acoustical, psychophysical, and psychological preference models, the optimal choice of concert hall size, shape, geometry, and composition is a complex problem. We used GAs to design two concert halls, the

Kirishima International Music Hall and the Tsuyama Music Cultural Hall, both in Japan. The GA allows optimization of listener preferences for the largest numbers of seats [2, pp. 148–158].

In the case of the Kirishima hall, we began with a shoebox shape and the GA iteratively evolved the shape of the hall, by independently moving the walls, to best satisfy listener preferences at 49 spatial locations distributed within the audience and performer areas [23, 24]. The front and rear walls were vertically bisected to obtain two faces, and each long wall along the side of the seating area was divided into four faces. The coordinates of the two bottom vertices of each surface were encoded on the chromosomes for the GA. Walls were kept vertical to examine the plan of the hall solely in terms of maximizing the average scale value of subjective preference obtained at the greatest number of seating positions, which meant reducing the IACC to its lowest possible value. We focused on this parameter first, because it is the most important in terms of overall preference and because everyone prefers the smallest value that can be attained (IACC $< \varepsilon$) [25]. This contrasts with individual listener preferences for other three parameters, such as LL, where there can be a wide range of preferred values.

The full GA strategy takes into account the listening preferences for both audience and performers (Fig. 8.5, top). Each wall was moved independently of the other walls to optimize the overall shape of the room (left, middle panel). In the acoustical simulation using image method, openings between walls were assumed not to reflect the sound. To minimize IACC, the rear wall of the stage and the rear wall of the audience area took on convex shapes because this avoids reflections from near the median plane.

The resulting shape of optimizing for minimal IACC is similar to the final leaf shape of the Kirishima International Music Hall, Japan (floor plans on the right). Contour lines of equal estimated IACC values are shown in the bottom left panel. To our knowledge, the Kirishima hall is the largest structure that has been designed thus far by GA optimization.

8.4.2 Rules of Thumb for Optimizing Hall Acoustics for Performances

From a historical viewpoint, for blending human life and the built environment, architects have been much more concerned with spatial criteria and visual perspective than temporal criteria and sound environments. On the other hand, from Sabine [19] onward, architectural acousticians have been mainly concerned with temporal criteria, represented primarily by reverberation time. Through most of the history of our practice, there has existed no comprehensive theory of design that included the spatial criterion represented by the IACC, so that discussions between acousticians and architects were rarely on the same subject. What we need is a general theory of physical environments in which deeply interrelated temporal and spatial, auditory and visual sensations are taken into account in a combined practice of acoustic and architectural design.

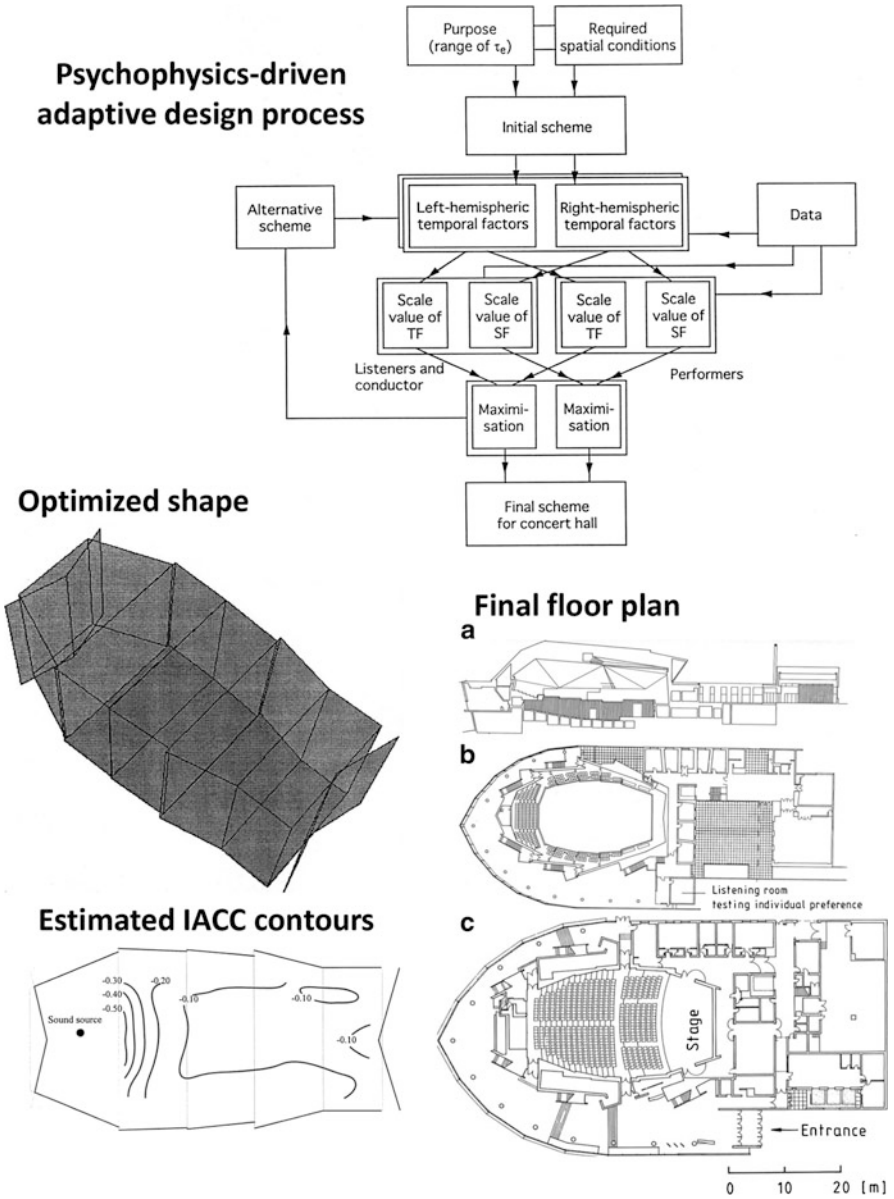


Fig. 8.5 Design of a concert hall using GAs for optimizing listener experience on the basis of acoustic theory, perceptual psychophysics, and the psychology of preferences. *Top*. Schematic of the GA-based iterative, evolutionary design process. *Lower right*. Final design of the Kirishima International Concert Hall. *Left, middle*. Final, optimized shape of the hall. *Lower left*. Predicted listener preference contours for IACC, which was the deciding spatial factor that effectively determined the hall's shape

In designing a concert hall, one must first have some idea of the type of program material that will be performed, so that the dimensions, shape, and composition of the hall can be optimized accordingly for those sounds. The nature of the music or speech material, specifically its minimal effective durations $(\tau_e)_{min}$, will determine the optimal first reflection and subsequent reverberation (T_{sub}) times for the hall. For pipe organ music, for example, values of τ_e about 200 ms or $T_{sub} \sim 4.6$ s are most appropriate, whereas for chamber music, these values would be more like 65 ms and 1.5 s, respectively.

For a given hall, the conductor or the sound coordinator should select music material with effective durations that best blend the music and the sound field within that space. Using the same methods as we used for individualized seat selection, one can also determine the best positions for performers to take by finding those locations that maximize listener preferences over the widest range of seats. The performance can be further optimized by locating the various instruments and performers such that their effective durations are matched to the first reflection times associated with their stage positions. For instance, the values of $(\tau_e)_{min}$ for violins is usually shorter than that of contrabasses that play mainly in lower frequency ranges, so it is advantageous to shift the position of violins closer to a stage sidewall (usually the left wall) and to position the contrabasses more to the center, increasing their distance to the sidewalls, thereby increasing their first reflection time.

In order to provide vertical reflections from places near the median plane of each of the performers on the stage, the back wall on the stage is carefully tilted [9, 26]. The tilted back wall consists of six sub-walls with angles adjusted to provide appropriate reflections within the median plane of the performers. It is worth noticing that tilted sidewalls onstage provide good reflections to the audience sitting close to the stage that reduce IACC. Also, the sidewall can provide useful reflections (arriving from a performer's back) for a piano soloist, for example.

To suppress normal-mode vibration of the stage floor and anomalous sound radiation from the stage floor during performances, arrange joists in triangular configurations, without any neighboring parallel structure. The floor should be thin, e.g., 27 mm, in order to radiate sound effectively through the coupled vibrations of instruments such as cellos and contrabasses. During rehearsal, music performers can control radiation power to some extent by adjusting their position or by inserting a rubber pad under their instrument.

IACC should be kept as small as possible for the audience by suppressing a strong reflection from the ceiling and by appropriate reflections from the left and right sidewall. If the source signal, like most music, contains mainly mid-frequency components, the reflections from the sidewalls should be optimized by orienting the walls at roughly 55° relative to each listener's front [9].

In order to obtain low IACCs for most listeners, we design ceilings using a number of triangular plates with adjusted angles, and the sidewalls are given a 10 % tilt with respect to the main-audience floor. In addition, diffusing elements are added to sidewalls to avoid image shifts of onstage sound sources caused by the strong high frequency (>2 kHz) reflections. These diffusers on the sidewalls,

for example, are designed by a deformation of the Schroeder diffuser [27] that eliminates thin partitions that separate the wells.

Under actual listening conditions in an existing hall, the perceived IACC depends on whether or not the amplitudes of reflection exceed hearing thresholds. Thus, a more diffuse sound field may be perceived with increasing power of the sound source. For example, Keet [13] first reported that ASW increases with increased LLs. When the source is weak enough, just above threshold, one hears only the direct sound, and one gets the opposite of a diffuse sound impression. In general, small values of IACC can be realized only by early strong reflections. If the sound source is located in the center line on the stage, then coherent signals arrive at the same time from both sidewalls to produce a high IACC. From this point of view, acoustical-asymmetric properties in shaping the hall may create further advantages.

For music performers on stage, temporal factors, such as pitch, timbre, and sound qualities related to reverberations, are much more critical than the spatial factors, such as LL and sound diffuseness. Since musicians perform over time durations, first reflections with delays commensurate with those of the sounds they produce are of particular importance. We recommend that the sound field for the conductor should be designed as that of a "listener" with appropriate reflections of sidewalls on the stage [28]. Some design iterations may be required to optimize acoustic conditions for performers, conductors, and listeners, before the final scheme of the concert hall is reached.

8.5 Correlation-Based Model of Auditory Percepts

In order to design structures for listening, an adequate theory of auditory function is indispensable. Manfred Schroeder in his 1975 review of the known structure and function of the auditory system concluded that the sense of hearing is a very active field that is fascinating and ultimately still mysterious [20]. In many respects, the situation is not so different almost four decades later.

Over many years, we developed a comprehensive theory of major auditory percepts that is based on two internal representations: the monaural ACF and the binaural IACF. Features of the unnormalized monaural autocorrelation representation are associated with auditory qualities we have called "temporal sensations," namely loudness, pitch, and timbre. Features of the binaural cross-correlation representation are associated with "spatial sensations," namely binaural LL, ASW, and SD. At the cortical level, we found observable neural response correlates associated with temporal sensations predominantly in the left hemisphere, whereas those associated with spatial sensations were found predominantly in the right hemisphere. The perceptual characteristics and neural correlates of all of these sensations are discussed in much greater depth elsewhere [2].

The various dependencies of the percepts on features of these two representations are shown in Fig. 8.6 and Table 8.2. The computation of these features from autocorrelation and cross-correlation functions is shown schematically in Fig. 8.2. We describe the various temporal and spatial perceptual attributes in turn.

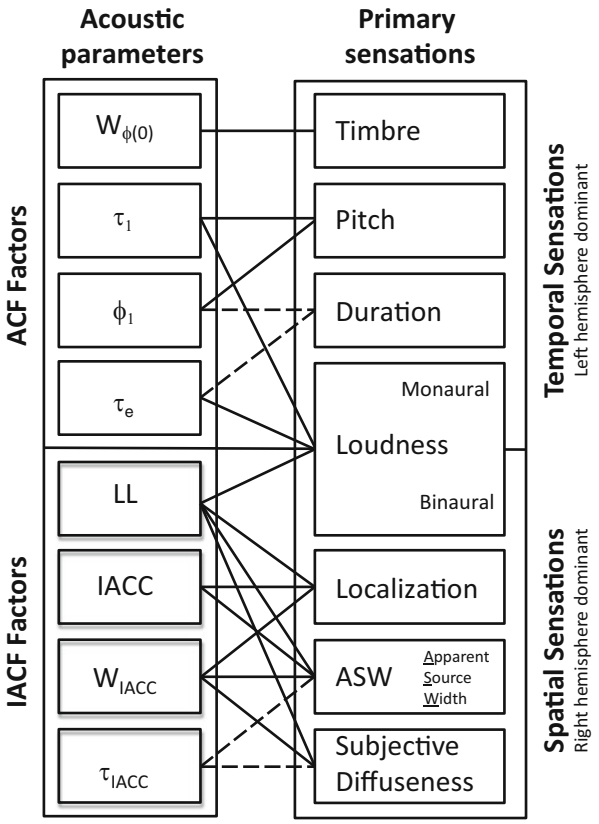


Fig. 8.6 Auditory acoustic features and related percepts. Relations between auditory percepts and features of monaural ACF and IACF. Temporal sensations are related to the temporal forms of signals, whereas spatial sensations are related to the perception of spatial aspects of sounds and the enclosing space. *Solid lines* indicate stronger relations, *dashed lines*, weaker determinants. Observable neural correlates of temporal and spatial sensations and their associated preferences are predominantly seen respectively in either left or right hemisphere

Table 8.2 Hemispheric specializations of temporal and spatial factors observed by analyses of SVR, EEG, and MEG in response to stimuli with changing acoustic parameters

Acoustic factors changed	SVR $A(P_1-N_1)$ latency difference	MEG N1m latency	EEG wave persistence or spatial extent*	MEG wave persistence
Temporal				
Δt_1	**L** > **R** (speech)		**L** > **R** (music)	**L** > **R** (speech)
T_{sub}			**L** > **R** (music)	
Spatial				
LL	**R** > **L** (speech)			
IACC	**R** > **L** (vowel /*a*/)	**R** > **L** (band noise)[a]	**R** > **L** (music)*	
	R > **L** (band noise)			
τ_{IACC}		**R** > **L** (band noise)[a]		
Head-related transfer function[b]		**R** > **L** (vowels)		

[a]Soeta and Nakagawa [29]
[b]Palomaki et al. [30]
*The asterisk indicates that spatial extent was the parameter, not persistence

8.5.1 Loudness

Loudness is a function of total SPL (dB SPL), periodicity (frequency, period) τ_1, normalized amplitude ϕ_1 (regularity), effective duration τ_e (bandwidth), and duration D.

In paired comparison tests of loudness, we found that loudness is both a function of frequency and effective duration. Loudness of a pure tone is greater than that of filtered noise centered on the pure tone, and loudness increased with increasing effective duration τ_e within the critical band [23, 24]. For example, we found that loudness of band pass noise produced by an extremely sharp filter (2068 dB/octave) with identical SPL is not constant within the critical band, quite a different result from previous studies [31]. Practically, this means that if environmental noise, such as aircraft landing noise, contains a pure-tone component or has long effective duration τ_e, then loudness increases and accordingly annoyance increases [32]. This is a typical example, why only measurements of SPL from a sound level meter fail to adequately predict subjective attributes. Just adding +10 dB to a noise with a coherent component is too simple, as loudness and annoyance increases with the value of the effective duration τ_e.

8.5.2 Duration

We studied the perception of duration for short complex tone and pure tones. Perceptually, the duration sensation depends most strongly on the physical signal duration, D, but it can also be influenced slightly by periodicity τ_1, which corresponds to perceived periodicity pitch. The apparent stimulus duration depends on signal periodicity τ_1, i.e., pure tone frequency (f_1) or the fundamental frequency (F_0) of a complex tone, rather than spectral center of gravity. For example, a 3,000 Hz pure tone of duration 150 ms has the same perceived duration as 140 ms duration pure and complex tones that evoke pitches at 500 Hz (a 500 Hz pure tone and a pure tone dyad of 3,000 and 3,500 Hz, $F_0 = 500$ Hz). Thus, when the sound signal evokes a lower periodicity pitch, perceived duration is slightly (~7 %) longer. Duration thus appears to be weakly linked with periodicity, which evokes sensations associated with periodicity pitch, rather than spectral center of gravity, which evokes sensations associated with pitch height.

8.5.3 Pitch

The periodicity pitch of pure and complex tones is associated with the time lag τ_1 and its multiples in the autocorrelation representation. We conducted several pitch-matching studies using stimuli with missing fundamentals and found the pitch

matches to be independent of phase (in-phase vs. random phase). The highest missing fundamental matched was ~1,200 Hz [33], which means that the highest frequency components in such a missing fundamental stimulus need to be less than about 5 kHz, a value that is comparable to the frequency limit of significant phase locking in auditory nerve fibers, see Sect. 8.6.1.

Pitch strength or salience, depends on the regularity of the signal, which is indicated by the amplitude ϕ_1 of the corresponding periodicity in the normalized ACF (see also [34, 35]). Maximum pitch strength is observed at $\phi_1 = 1.0$, and no pitch is perceived when $\phi_1 < \varepsilon$, ε being a small positive number associated with the threshold for pitch strength.

8.5.4 Timbre

Timbre includes all those aspects of sound quality that are independent of loudness, duration, pitch, and its spatial attributes, i.e., it encompasses those qualities of sound texture that distinguish two notes of equal pitch, loudness, duration, location, and apparent size that are played by different musical instruments. One of the dimensions of timbre is the relative brightness of a sound, which is often characterized in terms of spectral tilt. The corresponding autocorrelation feature is the factor $W_{\phi(0)}$, defined by width of the central (zero-lag) peak as measured by the delay time at the first 0.5 crossing of the normalized ACF.

8.5.5 Binaural LL

The remaining perceptual attributes are spatial sensations that are associated with features in the IACF. In the binaural listening condition, the LL is the mean amplitude given by $\mathrm{LL} = 10\ \log[\Phi_{\mathrm{ll}}(0)\Phi_{\mathrm{rr}}(0)]^{1/2}/\Phi_{\mathrm{reference}}(0)$ [dBA], with $\Phi_{\mathrm{reference}}(0)$ being the reference amplitude.

8.5.6 Sound Direction in the Horizontal Plane

Apparent sound direction in the horizontal plane depends partially on all of the spatial factors, some of which are features of the IACF. The most important factors are dominant interaural time delay (ITD), τ_{IACC}, and interaural intensity difference (IID), which is a function of the sound energies at the two ears, $\Phi_{\mathrm{ll}}(0)$ and $\Phi_{\mathrm{rr}}(0)$. Direction can be influenced more weakly by the broadness of the ITD peak W_{IACC} and the degree of IACC.

8.5.7 Apparent Source Width

ASW is the perceived spatial breadth of a sound, how much of the horizontal plane it seems to occupy. We developed a model for estimating the scale value for ASW as a function of IACC, ITD breadth W_{IACC}, and LL. ASW $\approx \alpha(\mathrm{IACC})^{3/2} + \beta(W_{\mathrm{IACC}})^{1/2} + \gamma(\mathrm{LL})^{3/2}$, where $\alpha \approx -1.64$, $\beta \approx 2.42$, $\gamma \approx 0.005$. Thus we found that apparent width increases with decreasing IACC and increasing W_{IACC} as well as LL [36].

8.5.8 Subjective Diffuseness

SD is related to the sense of envelopment of sound. We found the scale values for this quality observed in paired comparison tests to be inversely proportional to the IACC, so that $\mathrm{SD} \approx -\alpha(\mathrm{IACC})^{3/2}$, where $\alpha = 2.9$ [37].

8.6 Observable Neural Correlates of Auditory Percepts

The foregoing correlation-based model of auditory perception is based on the notion that the auditory system uses an autocorrelation-like representation for analyzing the internal structure of signals (periodicity, spectrum, duration, and various transients and modulations), and an interaural cross-correlation representation for orienting sounds in space. Like Manfred Schroeder, we have all along had deep interests in how the auditory system might realize these representations and operations, i.e., how the central auditory system works. We subsequently developed an auditory signal-processing model with these two modes of representation in mind, as shown in Fig. 8.7. It can be noted that early on both Licklider [38] and Cherry [39] also proposed such dual auto- and cross-correlation architectures for auditory neurocomputation.

8.6.1 Auditory Signal-Processing Model

This functional, signal-processing model embodies our assumptions about where in the auditory system these two representations might reside. A sound source produces a signal that propagates through space, interacting with surroundings, to reach the listener's two ears. Vibrations in the middle ear and cochlea are mechanically filtered, and in some cases amplified by cochlear hair cells. Mechanical vibrations are transduced into ionic currents that then produce pulsatile action

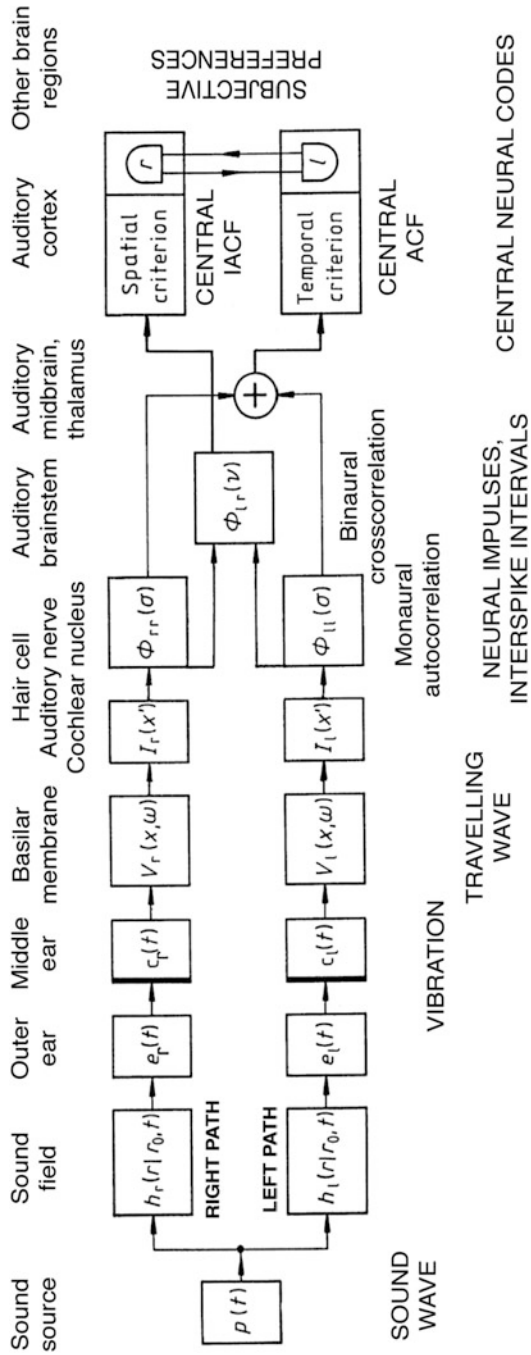


Fig. 8.7 Auditory signal processing model and central correlation representations. The neuroanatomical locations of central autocorrelation (IACF) and binaural cross-correlation (IACF) representations are hypothetical

potentials (spikes) in auditory nerve fibers that are phase locked to stimulus periodicities up to about 5 kHz.

The most direct correlate of the monaural ACF is found in patterns of spike timings in the auditory nerve [2, 34, 35, 40, Chap. 5]. The times between both consecutive and non-consecutive pairs of spikes, called all-order interspike intervals, form an autocorrelation-like, population-based neural representation of the stimulus in the auditory nerve. The representation, which combines the interspike intervals from the 30,000 axonal fibers that make up the human auditory nerve, is robust, precise, and effective up to the ~5 kHz limit of phase locking. The ensuing distribution resembles the positive portion of the stimulus ACF, such that there are direct correspondences between the monaural, temporal ACF parameters outlined in the preceding section, and features of this early temporal, neuronal representation.

The auditory nerve fibers synapse on neurons in the cochlear nucleus, and projections from these neurons in turn synapse on neurons elsewhere in the auditory brainstem and midbrain. Major pathways through the cochlear nucleus preserve spike fine timing and on their way to the midbrain send axon collaterals that synapse on neurons in binaural stations in the brain stem. The bipolar, binaural neurons in the auditory brainstem (medial superior olivary nucleus, MSO) act as coincidence detectors. Because the incoming spike trains are correlated, by virtue of phase locking, with the filtered stimulus at each ear, the population of neurons in this nucleus in effect carries out an interaural cross-correlation (IACF) operation. Thus there are direct neural correlates of the features of the IACF in the binaural stations in the auditory brainstem.

Unfortunately, despite considerable advances over the last several decades, the precise means by which these two temporal, correlation-based early auditory representations might be analyzed by the rest of the system are still poorly understood. As we outline below, we observed the central neural correlates of percepts associated with the monaural ACF, what we have called "temporal sensations" primarily in the left cerebral hemisphere. The observed correlates of percepts associated with the interaural correlation function (IACF), what we have called "spatial sensations," were all found in the right hemisphere. It is for this reason that in the schematic of Fig. 8.7 the combined monaural autocorrelation representations are connected to the left hemisphere, and the binaural cross-correlation representation goes to the right hemisphere. Beyond this, there are further neural-processing stages, cortical and subcortical (e.g., limbic), that mediate auditory preferences.

8.6.2 Observable Neural Correlates of Auditory Percepts and Preferences

The neuronal correlates of the monaural ACF can be observed in single unit responses at the level of the auditory nerve and cochlear nucleus. These also can

be seen in averaged gross electrical potentials, such as local field potentials and the frequency-following response (FFR). The neural correlates of the interaural correlation function can be found in single-unit spike train responses, local electrical field potentials, and averaged scalp responses (ABR) from sources in the auditory brainstem.

In a large series of neurophysiological experiments we have examined the neural correlates of auditory percepts and preferences at the level of the auditory brainstem, midbrain, and cortex. Whereas in the auditory nerve and cochlea, the fine timing patterns of neuronal response can be observed through a variety of means, as one ascends the pathway, overt signs of these progressively and dramatically decrease, and the observable neural correlates of auditory percepts then come mainly in the form of changes in neural response latency.

At the level of the auditory brainstem and midbrain, we used the auditory brainstem response (ABR, 0–10 ms latencies) to probe the nature of central auditory representations [41]; [2], pp. 40–48. Here systematic changes can be observed in the relative latencies of corresponding ABR peaks I–V from right and left pathways that covary with stimulus intensity (loudness) [42], sound direction (azimuth [41]), and IACC. These latencies of neuronal responses reflect the electrical effects of large numbers of spikes that are produced roughly synchronously in various neuronal nuclei as volleys of spikes make their way up the auditory pathway.

At the level of the cerebral cortex, we used slow vertex (SV) averaged late evoked electrical responses (SVR, 10–500 ms latencies), EEG, and MEG to probe observable neural correlates of both percepts and preferences [2, 9, 22]. These manifested themselves as changes in response latencies in the slow vertex responses and in the persistence and spatial extent of alpha waves in EEG and MEG recordings.

Slow vertex responses. Slow vertex responses reflect cortical neural responses with longer latencies than ABR, in the range of 170–320 ms. Using transitions between more-preferred to less-preferred stimulus conditions, we were able to observe changes in response latency that covaried with listener preferences. By changing sensation level (SL), first reflection time Δt_1 and IACC we consistently found that the latency increases with increasing preference (Fig. 8.8).

Alpha rhythms. Alpha rhythms can be observed both in running scalp electrical potentials (EEG) and in magnetic signals (MEG) that are generated by cortical populations. In both EEG and MEG, we consistently found that preferred conditions are associated with increased temporal persistence and spatial extent of alpha rhythms. Annoying stimuli, on the other hand, decreased the persistence and extent of the rhythms. Remarkably, these neural correlates of auditory preferences also were found to hold for visual preferences, as we will discuss in the next section.

The temporal persistence of the alpha rhythm is measured by computing the effective duration of the alpha-band EEG or MEG signal. The longer the effective duration, the more steady the rhythm over longer periods of time [43].

The spatial extent of the alpha rhythm was assessed in terms of how correlated were the alpha-band neural response signals across different electrode or sensor

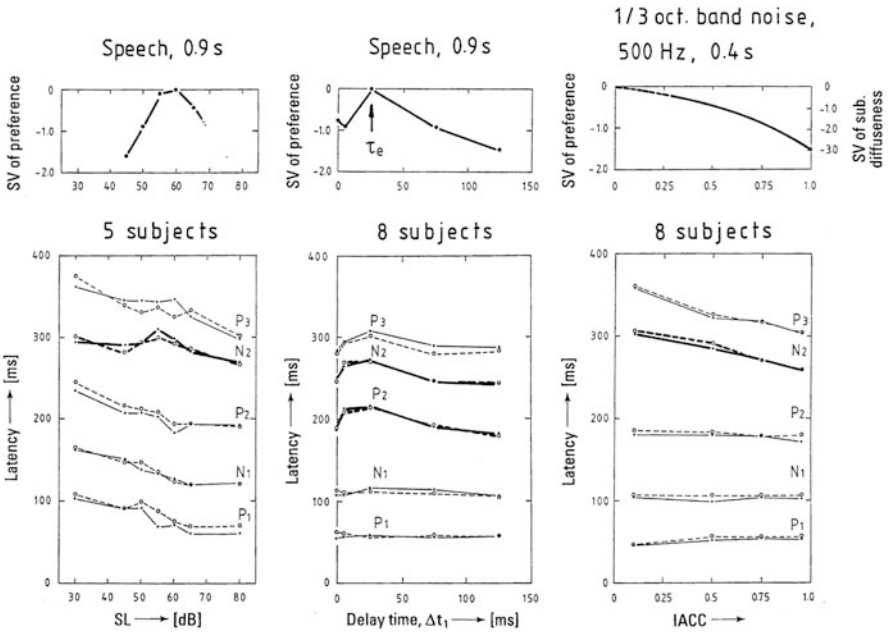


Fig. 8.8 Observable neural correlates of three acoustical factors in latencies of neural responses to two stimuli, speech and noise, as seen in averaged evoked auditory potentials (slow vertex responses). *Top plots*. Preference curves for the three factors, sound level (SL), first reflection time (Δt_1), and IACC. *Bottom plots*. Latencies of successive peaks and troughs in the slow vertex responses from left (*solid lines*) and right side (*dashed lines*) scalp recording sites, averaged over multiple subjects. Note correspondences between preferred values and longer latencies and interhemispheric relative latency differences [2, pp. 52–55]

sites. The higher the average correlation, the more similar are the alpha-band signals across different cortical locations, and the greater the spatial extent of the rhythm over the whole cortical surface. Interestingly, we found that the signal power in the alpha-band did not correlate with preference.

Hemispheric lateralization. In a series of experiments using SVR, EEG, and MEG in response to transitions between more-preferred and less-preferred stimulus conditions [44–47], we observed clear and consistent differences between neural responses associated with each of the two cerebral hemispheres (Table 8.2). In all cases, temporal sensations associated with first reflection time Δt_1 and subsequent reflection time T_{sub}, which correspond to features in the monaural ACF, were predominant at recording locations over the left hemisphere. Conversely, neural responses for spatial sensations that were related to changes in LL, IACC, predominant interaural delay (ITD), and head-related transfer function (HRTF) were observed in recording locations over the right hemisphere.

8.7 Visual Sensations

We believe that there are many deep parallels between auditory and visual perception because the two modalities may share common underlying neurally based correlational information-processing mechanisms. In the last decade we have applied our correlation theory and psychophysical methods to the study of visual sensations. Here we very briefly summarize some of the specific visual percepts and preferences. Details of the psychophysical and neurophysiological investigations are available in the primary published reports and are also presented in depth in book form [2].

As with audition, many visual percepts can be explained in terms of correlation-based representations. Temporal aspects of visual sensation, such as the perception of flickering lights, including a "missing fundamental" perception of flicker periodicity [48], can be described in terms of temporal ACFs (TACFs), whereas spatial aspects of visual sensation, such as the perception of texture can be described in terms of spatial ACFs (SACFs) [49].

Different basic aspects of visual perception and their corresponding features in temporal and spatial autocorrelation representations are illustrated in Fig. 8.9. For temporally modulated visual stimuli, the visual percepts of brightness, duration, flicker periodicity, and flicker frequency spectrum, correspond to auditory percepts of loudness, duration, pitch, and spectral aspects of timbre. Perceived visual contrast, coarseness, and regularity across spatial positions correspond roughly to auditory perceptual dimensions of loudness, low vs. high pitch, and low vs. high regularity (pitch salience, noise vs. tonal quality).

We also studied the preferences for these visual percepts as well as the observable neurophysiological signs that are associated with them [58]. There appear to be universal preferences for the rates and regularities of visual stimuli, such as flickering lights (e.g., twinkling stars, fires, water flows, sunlight playing on water), oscillating forms (leaves fluttering in the breeze), and visual textures (clouds, cloth patterns). One notices the pleasing correlated movements and the sound that leaves of bamboo trees make in the wind. As the classical Japanese poet Yakamochi Otomo (c. 715–785) observed "... A small bamboo clump of my house, we might listen to bracing sound moving in the wind of this evening" (poem *Manyoshu*, Y. Ando, translation).

As with auditory stimuli, there are inverted U-shaped preference curves for individual visual preferences—we like visual patterns that are neither too ordered nor too disordered. These also can be combined in a multifactor model to predict overall viewer satisfaction. In addition, we also found that preferred visual perceptual states produced coherent EEG and MEG alpha rhythm activity that persisted longer, i.e., with longer effective duration, and was more spatially widespread over the brain [43, 50–52].

Remarkably, as with auditory percepts and preferences, we also observed hemispheric lateralizations associated with visual sensations. The two cerebral hemispheres appear to play somewhat different roles, with the left cerebral hemisphere concerned more with linear, sequential modes of thinking, such as speech and

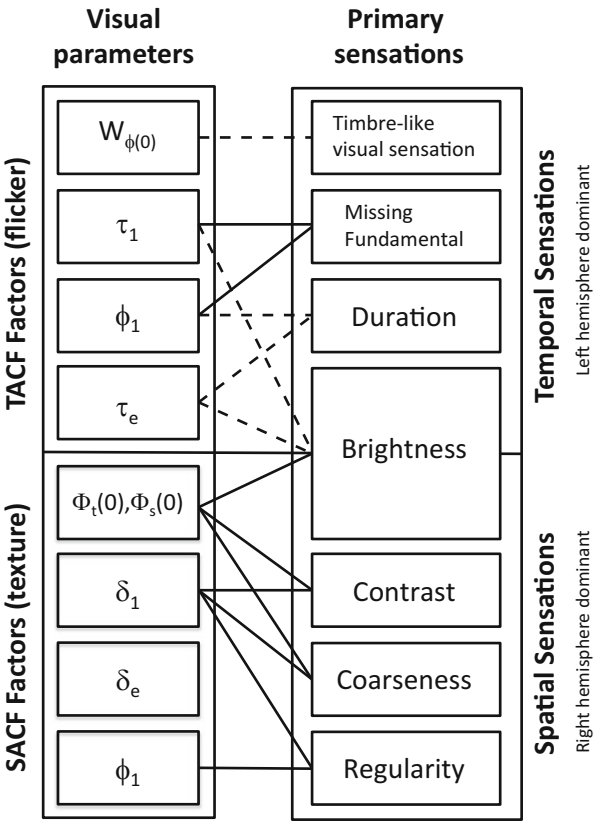


Fig. 8.9 Relations between visual percepts and features of visual temporal and spatial ACFs (TACF, SACF). Temporal sensations are related to the temporal forms of visual signals, such as flicker rate and duration, whereas spatial sensations are related to spatial vision, such as textural coarseness and regularity. Like their auditory counterparts, cf. Fig. 8.6, observable neural correlates of temporal and spatial visual sensations and their associated preferences are predominantly seen respectively in either left or right hemisphere

calculation, and with the right hemisphere more engaged with visual space in multidimensional and non-temporal terms [53–56].

In our investigations we found that the observable neural correlates of temporal sensations in both auditory and visual modalities were almost always found predominantly in the left hemisphere, whereas spatial sensations in both modalities were found predominantly in the right hemisphere [2]. Thus, there may exist a high degree of independence between the temporal and spatial factors for any subjective attribute, and this in turn suggests a basis for a theory of environmental design. Whenever possible, we should strive to design for both hemispheres in order to facilitate their complementary capabilities (Fig. 8.10).

8.8 Theory of Temporal and Spatial Environmental Design

We seek to develop a comprehensive human-centered theory of design that takes into account both the spatial and temporal dimensions of things. In architectural acoustics we need theories of sound and its propagation, models of human percepts

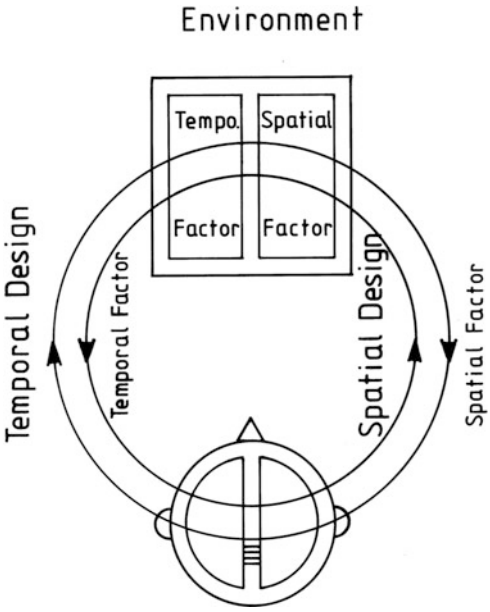


Fig. 8.10 Toward an integrated theory of temporal and spatial environmental design. The two cerebral hemispheres cooperate with each other to respond to and create spatial and temporal patterns in their environments

and preferences, techniques for optimizing the shapes and compositions of structures, and the means for building them. Our methods are rational to the extent that we attempt to make as clear as possible the basis of our design process, what human preferences we are attempting to optimize, and how we will go about that process.

In the general spirit of Manfred Schroeder's [1] influential *Science* article on architectural acoustics, a general theory of environmental design should include models of the physical world, models of how human perceive and interact with that world, as well as models of what human beings want [2, 57]. Such a theory needs to embrace and accommodate both individual tastes and group preferences. In taking the neural turn, in venturing inside the brain, we acquire more insights into the workings of those percepts and preferences, and consequently into the means by which both individual and group preferences can be better satisfied. We envision creative design practices that explicitly take into account temporal and spatial factors to best suit and utilize the respective needs and capabilities of the two cerebral hemispheres.

Philosophically, we hold great importance for the creative role of the individual. Personality is uniquely formed by both genetics and environmental influences on development. Temporal design is that part of an integrated theory of design that deals with the rhythms and temporal flows of life that occur over different time spans, from momentary awareness to daily, weekly, monthly, and yearly cycles, to even longer spans of phases of our lives (Fig. 8.11, top). Our theories of design should on a very fundamental level self-consciously take these different temporal regimes, each with its own special properties and possibilities, into account. Ultimately, we want to create design theories and practices that feed back upon

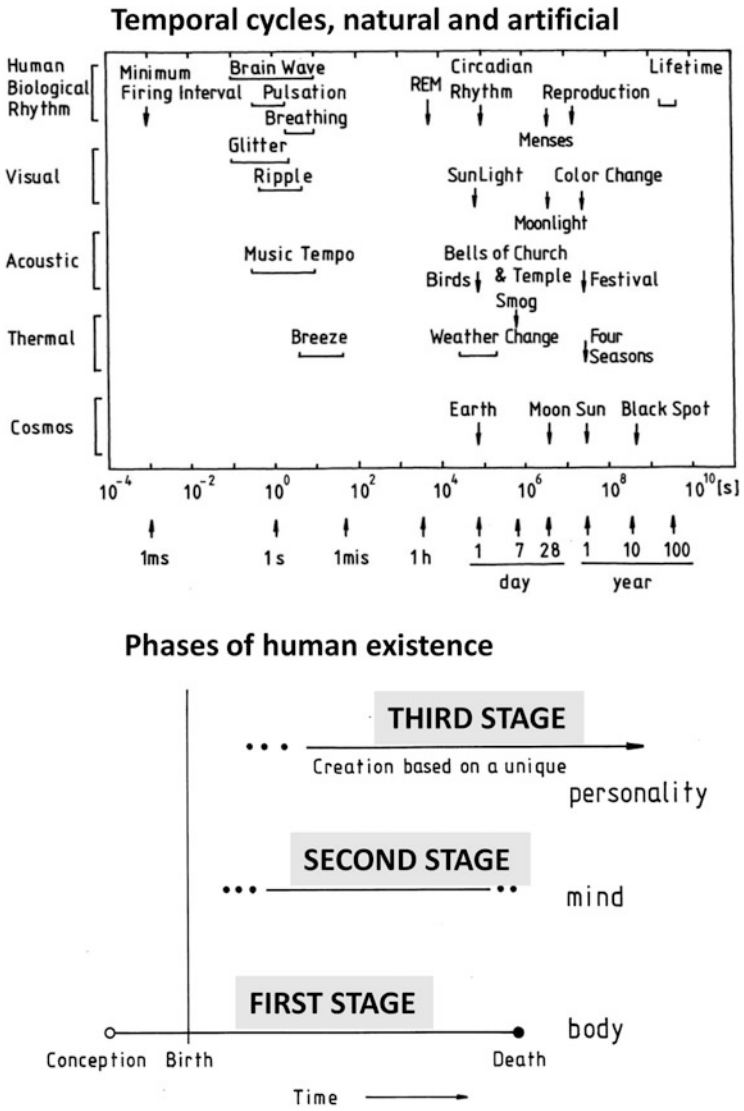


Fig. 8.11 Timescales of natural and artificial cycles and existential phases of human life. *Top*. The timescales of our world. *Bottom*. Phases of human life. The first stage is the lifespan of the mortal body. The second stage is the life of the individual mind. The third stage is the continuing effect of an individual's personality and ideas on the rest of humanity. While the first two stages of an individual are limited in time, the third stage can persist almost indefinitely into the future through the enduring memories of others

ourselves to enhance our own physical and mental development. When we close this final self-modifying loop, we enable an open-ended process of improved designs that further realize human potentials.

Over the arc of our lives, each of us inevitably passes through several overlapping phases (Fig. 8.11, bottom). The first stage is the life of the body. Our mortal bodies are conceived, born, mature, and die. The second stage is the life of the mind in which our minds develop, grow, realize their possibilities, and for some, eventually senesce. We share these first two phases with animals, which have bodies and minds that are basically like our own, the differences being only those of degrees. In some cases the capabilities of other animals exceed those of our own bodies and minds. The third stage of life is the life of personalities and ideas that are shared among human beings. Because of our social natures and the continual regeneration of our individual memories and collective cultures, personalities and ideas can reverberate long after the first two stages have ended.

Manfred Schroeder in the first and second stages of his life had many very positive and formative influences on my life and ideas. Manfred Schroeder is now continuing on in his third stage of life, and we believe and hope that his personality and ideas will continue to influence us and many others far into the future.

References

1. Schroeder, M.R.: Architectural acoustics. Science **151**, 1355–1359 (1966)
2. Ando, Y., Cariani, P.: Auditory and Visual Sensations. Springer, New York (2009)
3. Damaske, P.: Subjektive Untersuchungen von Schallfeldern. Acustica **19**, 199–213 (1967)
4. Damaske, P., Ando, Y.: Interaural cross-correlation for multichannel loudspeaker reproduction. Acustica **27**, 232–238 (1972)
5. Schroeder, M.R., Gottlob, D., Siebrasse, K.F.: Comparative study of European concert halls: correlation of subjective preference with geometric and acoustic parameters. J. Acoust. Soc. Am. **56**, 1195–1201 (1974)
6. Ando, Y.: Subjective preference in relation to objective parameters of music sound fields with a single echo. J. Acoust. Soc. Am. **62**, 1436–1441 (1977)
7. Ando, Y., Gottlob, D.: Effects of early multiple reflections on subjective preference judgments on music sound fields. J. Acoust. Soc. Am. **65**, 524–527 (1979)
8. Ando, Y.: Calculation of subjective preference at each seat in a concert hall. J. Acoust. Soc. Am. **74**, 873–887 (1983)
9. Ando, Y.: Architectural Acoustics, Blending Sound Sources, Sound Fields, and Listeners. AIP Press/Springer-Verlag, New York (1998)
10. Ando, Y.: Concert hall acoustics based on subjective preference theory. In: Rossing, T.D. (ed.) Springer Handbook of Acoustics. Springer, New York (2007)
11. Sabine, W.C.: Architectural Acoustics (J. Franklin Institute, January, 1915). In: Sabine, W.C. (ed.) Collected Papers on Acoustics, pp. 219–236. Dover reprint, New York (1915) (original edition Harvard, 1922)
12. Beranek, L.L.: Music, Acoustics and Architecture. Wiley, New York (1962)
13. Keet, M.V.: The influence of early lateral reflections on the spatial impression. In: Proceedings of the 6th International Congress of Acoustics, Tokyo, Paper E-2-4 (1968)
14. Barron, M., Marshall, A.H.: Spatial impression due to early lateral reflections in concert halls: the derivation of a physical measure. J. Sound Vib. **77**(2), 211–232 (1981)
15. Cremer, L., Müller, H.A., Schultz, T.J.: Principles and Applications of Room Acoustics, vols. 1 & 2. Applied Science Publishers, New York (1982)

16. Schroeder, M.: Concert halls: from magic to number theory. In: Deutsch, D. (ed.) The Psychology of Music, 2nd edn, pp. 25–46. Academic, Amsterdam (1999)
17. Schroeder, M.R.: Acoustics in human communications: room acoustics, music, and speech. J. Acoust. Soc. Am. **68**(1), 22–28 (1980)
18. Kato, K., Fujii, K., Hirawa, T., Kawai, K., Yano, T., Ando, Y.: Investigation of the relation between minimum effective duration of running autocorrelation function and operatic singing with different interpretation styles. Acta Acust. United Acust. **93**, 421–434 (2007)
19. Sabine, W.C.: Reverberation (The American Architect and the Engineering Record, 1900). In: Sabine, W.C. (ed.) Collected Papers on Acoustics, pp. 3–68. Dover reprint, New York (1900) (original edition Harvard, 1922)
20. Schroeder, M.R.: Models of hearing. Proc. IEEE **63**, 1332–1352 (1975)
21. Schroeder, M.R.: Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise. Freeman, New York (1991)
22. Ando, Y.: Concert Hall Acoustics. Springer, Heidelberg (1985)
23. Sato, S., Otori, K., Takizawa, A., Sakai, H., Ando, Y., Kawamura, H.: Applying genetic algorithms to the optimum design of a concert hall. J. Sound Vib. **258**, 517–526 (2002)
24. Sato, S., Kitamura, T., Ando, Y.: Loudness of sharply (2068 dB/Octave) filtered noises in relation to the factors extracted from the autocorrelation function. J. Sound Vib. **250**, 47–52 (2002)
25. Singh, P.K., Ando, Y., Kurihara, Y.: Individual subjective diffuseness responses of filtered noise sound fields. Acustica **80**, 471–477 (1994)
26. Nakayama, I.: Preferred time delay of a single reflection for performers. Acustica **54**, 217–221 (1984)
27. Schroeder, M.R.: Binaural dissimilarity and optimum ceilings for concert halls: more lateral sound diffusion. J. Acoust. Soc. Am. **65**, 58–963 (1979)
28. Meyer, J.: Influence of communication on stage on musical quality. In: Proceedings of the 15th International Congress on Acoustics, Trondheim, pp. 573–576. (1995)
29. Soeta, Y., Nakagawa, S.: Auditory evoked magnetic fields in relation to interaural time delay and interaural crosscorrelation. Hear. Res. **220**, 106–115 (2006)
30. Palomaki, K., Tiitinen, H., Makinen, V., May, P., Alku, P.: Cortical processing of speech sounds and their analogues in a spatial auditory environment. Cogn. Brain Res. **14**, 294–299 (2002)
31. Zwicker, E., Flottorp, G., Stevens, S.S.: Critical band width in loudness summation. J. Acoust. Soc. Am. **29**, 548–557 (1957)
32. Soeta, Y., Maruo, T., Ando, Y.: Annoyance of bandpass filtered noises in relation to the factor extracted from autocorrelation function. J. Acoust. Soc. Am **116**, 3275–3278 (2004)
33. Inoue, M., Ando, Y., Taguti, T.: The frequency range applicable to pitch identification based upon the auto-correlation function model. J. Sound Vib. **241**, 105–116 (2001)
34. Cariani, P.A., Delgutte, B.: Neural correlates of the pitch of complex tones. I. Pitch and pitch salience. II. Pitch shift, pitch ambiguity, phase-invariance, pitch circularity, and the dominance region for pitch. J. Neurophysiol. **76**, 1698–1734 (1996)
35. Cariani, P.: Temporal coding of periodicity pitch in the auditory system: an overview. Neural Plast. **6**, 147–172 (1999)
36. Sato, S., Ando, Y.: On the apparent source width (ASW) for bandpass noises related to the IACC and the width of the interaural cross-correlation function (WIACC). J. Acoust. Soc. Am. **105**, 1234 (1999)
37. Ando, Y., Kurihara, Y.: Nonlinear response in evaluating the subjective diffuseness of sound field. J. Acoust. Soc. Am. **80**, 833–836 (1986)
38. Licklider, J.C.R.: Three auditory theories. In: Koch, S. (ed.) Psychology: A Study of a Science. Study I. Conceptual and Systematic. vol. I, pp. 41–144. McGraw-Hill, New York (1959)
39. Cherry, C.: Two ears – but one world. In: Rosenblith, W.A. (ed.) Sensory Communication, pp. 99–117. MIT Press/John Wiley, New York (1961)

40. Secker-Walker, H.E., Searle, C.L.: Time domain analysis of auditory-nerve-fiber firing rates. J. Acoust. Soc. Am. **88**, 1427–1436 (1990)
41. Ando, Y., Yamamoto, K., Nagamastu, H., Kang, S.H.: Auditory brainstem response (ABR) in relation to the horizontal angle of sound incidence. Acoust. Lett. **15**, 57–64 (1991)
42. Hecox, K., Galambos, R.: Brain stem auditory evoked responses in human infants and adults. Arch. Otolaryngol. **99**, 30–33 (1974)
43. Soeta, Y., Okamoto, Y., Nakagawa, S., Tonoike, M., Ando, Y.: Autocorrelation analyses of MEG alpha waves in relation to subjective preference of a flickering light. Neuro Rep. **13**, 527–533 (2002)
44. Ando, Y., Kang, S.H., Morita, K.: On the relationship between auditory-evoked potential and subjective preference for sound field. J. Acoust. Soc. Jpn. (E) **8**, 197–204 (1987)
45. Ando, Y., Kang, S.H., Nagamatsu, H.: On the auditory-evoked potentials in relation to the IACC of sound field. J. Acoust. Soc. Jpn. (E) **8**, 183–190 (1987)
46. Ando, Y., Chen, C.: On the analysis of the autocorrelation function of α-waves on the left and right cerebral hemispheres in relation to the delay time of single sound reflection. J. Archit. Plan. Environ. Eng., AIJ **488**, 67–73 (1996)
47. Chen, C., Ando, Y.: On the relationship between the autocorrelation function of the α-waves on the left and right cerebral hemispheres and subjective preference for the reverberation time of music sound field. J. Archit. Plan. Environ. Eng., AIJ **489**, 73–80 (1996)
48. Fujii, K., Kita, S., Matsushima, T., Ando, Y.: The missing fundamental phenomenon in temporal vision. Psychol. Res. **64**, 149–154 (2000)
49. Fujii, K., Sugi, S., Ando, Y.: Textural properties corresponding to visual perception based on the correlation mechanism in the visual system. Psychol. Res. **67**, 197–208 (2003)
50. Soeta, Y., Nakagawa, S., Tonoike, M., Ando, Y.: Magnetoencephalographic responses corresponding to individual subjective preference of sound fields. J. Sound Vib. **258**, 419–428 (2002)
51. Soeta, Y., Uetani, S., Ando, Y.: Relationship between subjective preference and alpha wave activity in relation to temporal frequency and mean luminance of a flickering light. J. Opt. Soc. Am. A **19**, 289–294 (2002)
52. Soeta, Y., Uetani, S., Ando, Y.: Propagation of repetitive alpha waves over the scalp in relation to subjective preferences for a flickering light. Int. J. Psychophysiol. **46**, 41–52 (2002)
53. Sperry, R.W.: Lateral specialization in the surgically separated hemispheres. In: Schmitt, F.O., Worden, F.C. (eds.) The Neurosciences: Third Study Program. MIT Press, Cambridge (1974) (Chapter 1)
54. Davis, A.E., Wada, J.A.: Hemispheric asymmetry: frequency analysis of visual and auditory evoked responses to non-verbal stimuli. Electroencephalogr. Clin. Neurophysiol. **37**, 1–9 (1974)
55. Galin, D., Ellis, R.R.: Asymmetry in evoked potentials as an index of lateralized cognitive processes: relation to EEG alpha asymmetry. Neuropsychologia **13**, 45–50 (1975)
56. Levy, J., Trevarthen, C.: Metacontrol of hemispheric function in human split-brain patient. J. Exp. Psychol. Hum. Percept. Perform. **2**, 299–312 (1976)
57. Ando, Y.: Theory of temporal and spatial environmental design. In: McGraw-Hill Yearbook of Science & Technology 2009, pp. 384–389. McGraw-Hill, New York (2009)
58. Soeta, Y., Mizuma, K., Okamoto, Y., Ando, Y.: Effects of the degree of fluctuation on subjective preference for a 1 Hz flickering light. Perception **34**, 587–593 (2005)
59. Ando, Y., Noson, D.: Music and Concert Hall Acoustics. Academic, London (1997)

***

Biography



Yoichi Ando received his Ph.D. degree from Waseda University, 1975 (adviser Takeshi Itoh). Early in his career he was an Alexander-von-Humboldt Fellow at the Third Physics Institute, University of Goettingen, where he worked with Manfred Schroeder. He has held a number of positions at Kobe University (Associate Professor, Professor, and Associate Dean) and is currently Professor Emeritus there. He has authored over 130 papers on acoustics and has written several books on architectural acoustics (Concert Hall Acoustics. Springer-Verlag, Heidelberg, 1985; Architectural Acoustics, Blending Sound Sources, Sound Fields, and Listeners. AIP Press/Springer-Verlag, New York, 1998; Auditory and Visual Sensations, Springer-Verlag, New York, 2009). His honors include a "Laurea Honoris Causa" from the University of Ferrara and a member of the Academy of Sciences of the Institute of Bologna.



Peter Cariani's training and work has involved theoretical biology, biological cybernetics, and auditory neuroscience (B.S. 1978, MIT, biology; M.S. 1983, Ph.D. 1989, Binghamton University, systems science). His doctoral work explored epistemic implications of self-constructing adaptive systems (evolutionary robotics). Subsequent work at Eaton-Peabody Laboratory for Auditory Physiology investigated autocorrelation-like neural time codes for pitch, timbre, and consonance and proposed artificial neural timing nets for sound separation. In 2009 he guest-edited Yoichi Ando's book, *Auditory and Visual Sensations*. He is presently a Senior Research Scientist at the Hearing Research Center at Boston University and a Clinical Instructor in Otology and Laryngology at Harvard Medical School. He currently teaches courses at Harvard and MIT related to psychology of music.

Chapter 9
Manfred R. Schroeder: A Personal Memoir, Optimizing the Reflection Phase Grating

Peter D'Antonio

I did not have the privilege of working directly with Manfred Schroeder, but rather I was motivated and inspired by the development of the reflection phase grating in the late 1970s. This research was the spark that ignited my passion for acoustics and converted my avocation for music and recording into a second vocation and the birth of RPG Diffusor Systems, Inc.

This story begins in 1980, in the conference room of the Laboratory for the Structure of Matter at the Naval Research Laboratory (NRL) in Washington, DC, Fig. 9.1, where I was employed as a diffraction physicist. Knowing my interest in music, a colleague handed me the latest issue of Physics Today, Fig. 9.2, with a cover photo of Manfred Schroeder seated in an anechoic chamber. This was my first virtual meeting with Manfred Schroeder. The article suggested using number theoretic diffusors in concert halls to provide lateral reflections. While my interest at the time was not in concert halls, and in fact at this time my only link to the field of acoustics was a love of composing, recording and performing music, I became fascinated with the thought of using these diffusors in a renovation of Underground Sound, Fig. 9.3, a private recording studio I originally built in 1972 with Jerry Ressler, a colleague and fellow musician. The acoustic renovation utilized a new concept called Live End Dead End proposed by Don and Carolyn Davis, of Synergetic Audio Concepts ["The LEDE concept for the control of acoustic and psychoacoustic parameters in recording control rooms," J. Audio Eng. Soc., 28, 585–95 (1980)].

At NRL, I was examining the three-dimensional structure of matter in various phases, using electron, X-ray, and neutron diffraction techniques, Fig. 9.4. I shared the article with John Konnert, a colleague in my group, and it became apparent that the "reflection phase gratings" suggested by Schroeder were in effect,

P. D'Antonio (✉)
RPG Diffusor Systems, Inc., Upper Marlboro, MD 20774, USA
e-mail: pdantonio@rpginc.com

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_9,
© Springer International Publishing Switzerland 2015



Fig. 9.1 U.S. Naval Research Laboratory, Washington, DC

two-dimensional sonic crystals, which scatter sound in the same way that three-dimensional crystal lattices scatter electromagnetic waves, Fig. 9.4. Since the diffraction theory employed in X-ray crystallographic studies were applicable to reflection phase gratings, it was straightforward for us to model and design the reflection phase gratings. The central theoretical connection was the coherent diffraction equation developed by Sir Lawrence Bragg, shown in Fig. 9.4, where the product of two times the periodicity repeat, d, and the sin of the diffraction direction, θ, equaled the wavelength, λ, times and integer n.

Having scientific backgrounds, John Konnert and I approached acoustics as we did the field of diffraction physics and began researching and publishing findings in the scientific literature. The Audio Engineering Society and Syn-Aud-Con offered a unique forum and community for discussing the research. In October 1983, at the 74th AES Convention in New York, I presented our research on the Schroeder diffusor in the Studio Design Session C, shown in Fig. 9.5, with a bit of intimidation, because Manfred Schroeder was the lead-off invited speaker. As part of this presentation, I described an Apple II program, shown in Fig. 9.6, which allowed acousticians to design these phase gratings, as well as plot their diffraction patterns. Following the session, Manfred and I had our first personal meeting where he enlightened me on the use of the Chinese Remainder Theorem, which enabled the creation of a two-dimensional primitive root sequence from a longer one-dimensional sequence, maintaining the beneficial Fourier property of a flat power spectrum. In Fig. 9.7, I show his handwritten notes in the 74th Technical Meeting & Professional Exhibits AES Program Oct. 8–12, 1983, illustrating the diagonal filling process, making use of periodicity, for several primes, N, for which $N-1$ could be factored into two relative coprimes, i.e., it cannot be used for $N=5$.

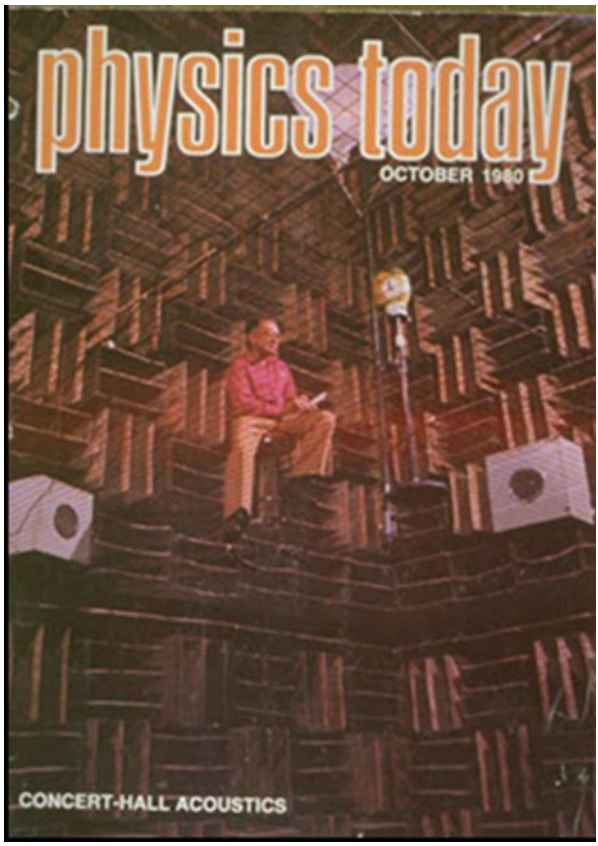


Fig. 9.2 Cover of Physics Today, October 1980, where I first was introduced to Manfred Schroeder

Two periods of a reflection phase grating, with $N = 17$ divided wells and well width W, are shown in Fig. 9.8. The angles of incidence and diffraction are α_i and α_d, respectively. There are three aspects of the phase grating that are important, namely the number of periods, the number of wells, and their relative depth with respect to a reference surface plane. Any periodic surface, a diffraction grating, a crystal or a reflection phase grating, scatters sound coherently when a certain condition is satisfied, as described in Fig. 9.9. Incident ray AB is reflected as BD. Incident ray EG is reflected as ray GH. When the difference in path length BC-FG is equal to an integral number of wavelengths, mλ, coherent scattering occurs in the diffraction direction α_d, according to Eq. 1, where N is an odd prime and W is the width of a well.

$$\sin \alpha_{\mathrm{d}} = \frac{\mathrm{m}\lambda}{\mathrm{NW}} - \sin \alpha_{\mathrm{i}} \tag{9.1}$$

In far field theory, sin α_i is assumed to be 0. The second and third aspects pertaining to the number of wells and their relative depths, is where Schroeder's



Fig. 9.3 Underground Sound Recording Studios, where my interest in diffusion was born

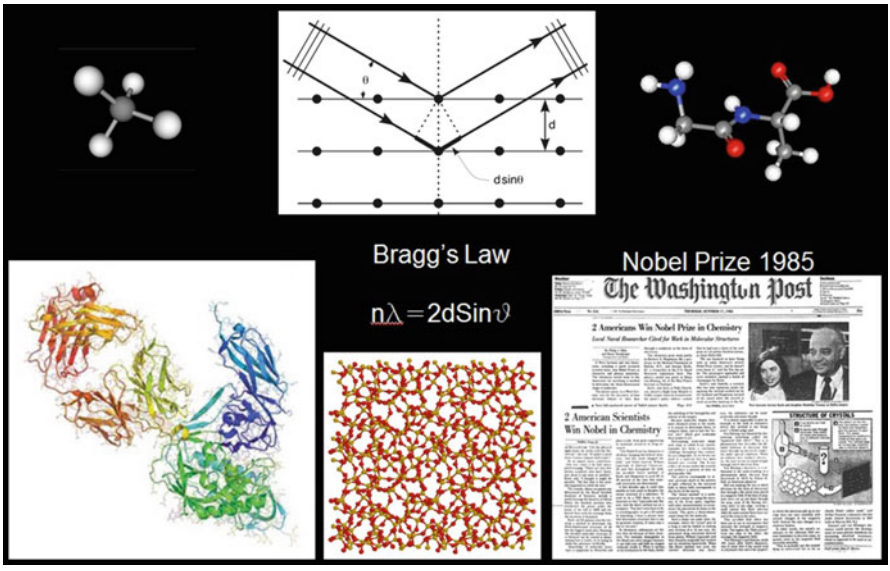
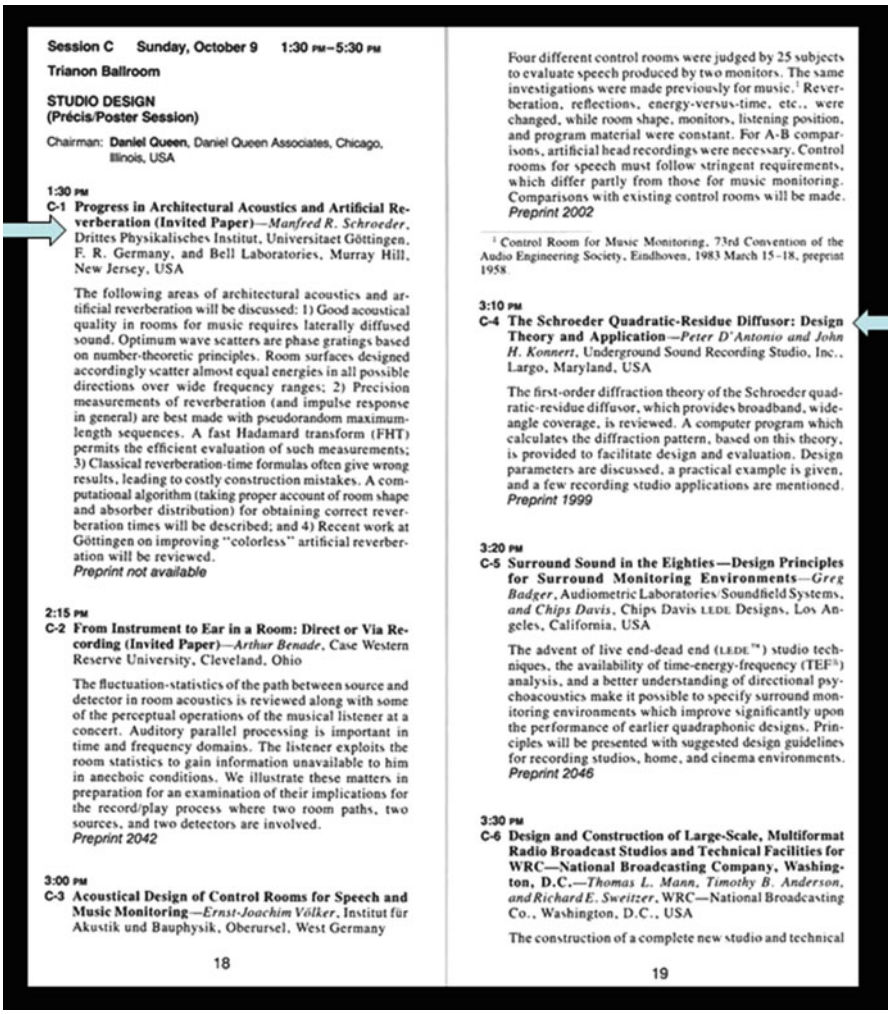


Fig. 9.4 A collage of matter in gas, crystalline, and amorphous phases that I was studying in my work as a diffraction physicist at NRL, in a group headed by Nobel Laureate Dr. Jerome Karle. Bragg's Law was directly transferrable from crystallography to the reflection phase grating

insight and genius came into play. Being a proficient mathematician, he was very aware of the power and "magic" of prime numbers. He often marveled at the unreasonable effectiveness of number theory in science and communication and described numerous applications in his books. Much of this early research was done by Carl Friedrich Gauss in the eighteenth century.



Session C Sunday, October 9 1:30 PM–5:30 PM

Trianon Ballroom

STUDIO DESIGN
(Précis/Poster Session)

Chairman: Daniel Queen, Daniel Queen Associates, Chicago, Illinois, USA

1:30 PM
C-1 Progress in Architectural Acoustics and Artificial Reverberation (Invited Paper)—*Manfred R. Schroeder*, Drittes Physikalisches Institut, Universitaet Göttingen, F. R. Germany, and Bell Laboratories, Murray Hill, New Jersey, USA

The following areas of architectural acoustics and artificial reverberation will be discussed: 1) Good acoustical quality in rooms for music requires laterally diffused sound. Optimum wave scatters are phase gratings based on number-theoretic principles. Room surfaces designed accordingly scatter almost equal energies in all possible directions over wide frequency ranges; 2) Precision measurements of reverberation (and impulse response in general) are best made with pseudorandom maximum-length sequences. A fast Hadamard transform (FHT) permits the efficient evaluation of such measurements; 3) Classical reverberation-time formulas often give wrong results, leading to costly construction mistakes. A computational algorithm (taking proper account of room shape and absorber distribution) for obtaining correct reverberation times will be described; and 4) Recent work at Göttingen on improving "colorless" artificial reverberation will be reviewed.
Preprint not available

2:15 PM
C-2 From Instrument to Ear in a Room: Direct or Via Recording (Invited Paper)—*Arthur Benade*, Case Western Reserve University, Cleveland, Ohio

The fluctuation-statistics of the path between source and detector in room acoustics is reviewed along with some of the perceptual operations of the musical listener at a concert. Auditory parallel processing is important in time and frequency domains. The listener exploits the room statistics to gain information unavailable to him in anechoic conditions. We illustrate these matters in preparation for an examination of their implications for the record/play process where two room paths, two sources, and two detectors are involved.
Preprint 2042

3:00 PM
C-3 Acoustical Design of Control Rooms for Speech and Music Monitoring—*Ernst-Joachim Völker*, Institut für Akustik und Bauphysik, Oberursel, West Germany

18

Four different control rooms were judged by 25 subjects to evaluate speech produced by two monitors. The same investigations were made previously for music.[1] Reverberation, reflections, energy-versus-time, etc., were changed, while room shape, monitors, listening position, and program material were constant. For A-B comparisons, artificial head recordings were necessary. Control rooms for speech must follow stringent requirements, which differ partly from those for music monitoring. Comparisons with existing control rooms will be made.
Preprint 2002

[1] Control Room for Music Monitoring, 73rd Convention of the Audio Engineering Society, Eindhoven, 1983 March 15–18, preprint 1958.

3:10 PM
C-4 The Schroeder Quadratic-Residue Diffusor: Design Theory and Application—*Peter D'Antonio and John H. Konnert*, Underground Sound Recording Studio, Inc., Largo, Maryland, USA

The first-order diffraction theory of the Schroeder quadratic-residue diffusor, which provides broadband, wide-angle coverage, is reviewed. A computer program which calculates the diffraction pattern, based on this theory, is provided to facilitate design and evaluation. Design parameters are discussed, a practical example is given, and a few recording studio applications are mentioned.
Preprint 1999

3:20 PM
C-5 Surround Sound in the Eighties—Design Principles for Surround Monitoring Environments—*Greg Badger*, Audiometric Laboratories/Soundfield Systems, *and Chips Davis*, Chips Davis LEDE Designs, Los Angeles, California, USA

The advent of live end-dead end (LEDE™) studio techniques, the availability of time-energy-frequency (TEF®) analysis, and a better understanding of directional psychoacoustics make it possible to specify surround monitoring environments which improve significantly upon the performance of earlier quadraphonic designs. Principles will be presented with suggested design guidelines for recording studios, home, and cinema environments.
Preprint 2046

3:30 PM
C-6 Design and Construction of Large-Scale, Multiformat Radio Broadcast Studios and Technical Facilities for WRC—National Broadcasting Company, Washington, D.C.—*Thomas L. Mann, Timothy B. Anderson, and Richard E. Sweitzer*, WRC—National Broadcasting Co., Washington, D.C., USA

The construction of a complete new studio and technical

19

Fig. 9.5 Program for the Studio Design Session C, at the 74th AES Convention in New York, where I first met Manfred Schroeder

I would like to digress a bit to present a statement made by the New York Times writer John Tierney in which he stated that "No matter how its practitioners of mathematics try to deliberately ignore the physical world, they consistently produce the best tools for understanding it." A few examples help to emphasize this idea. The Greeks decide to study a strange curve called an ellipse and 2000 years later astronomers discover that it describes the orbits of the planets. In 1854 Bernhard Riemann conjectured that it's not possible to draw two parallel lines ad infinitum and described curved space, which 60 years later Einstein announced as the shape of the universe. In the eighteenth century in Gottingen, Carl Friedrich Gauss

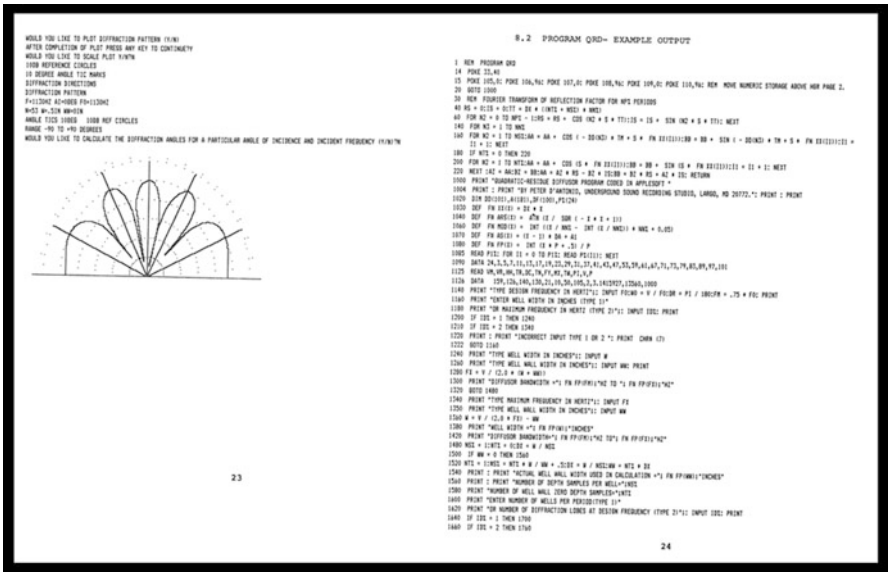
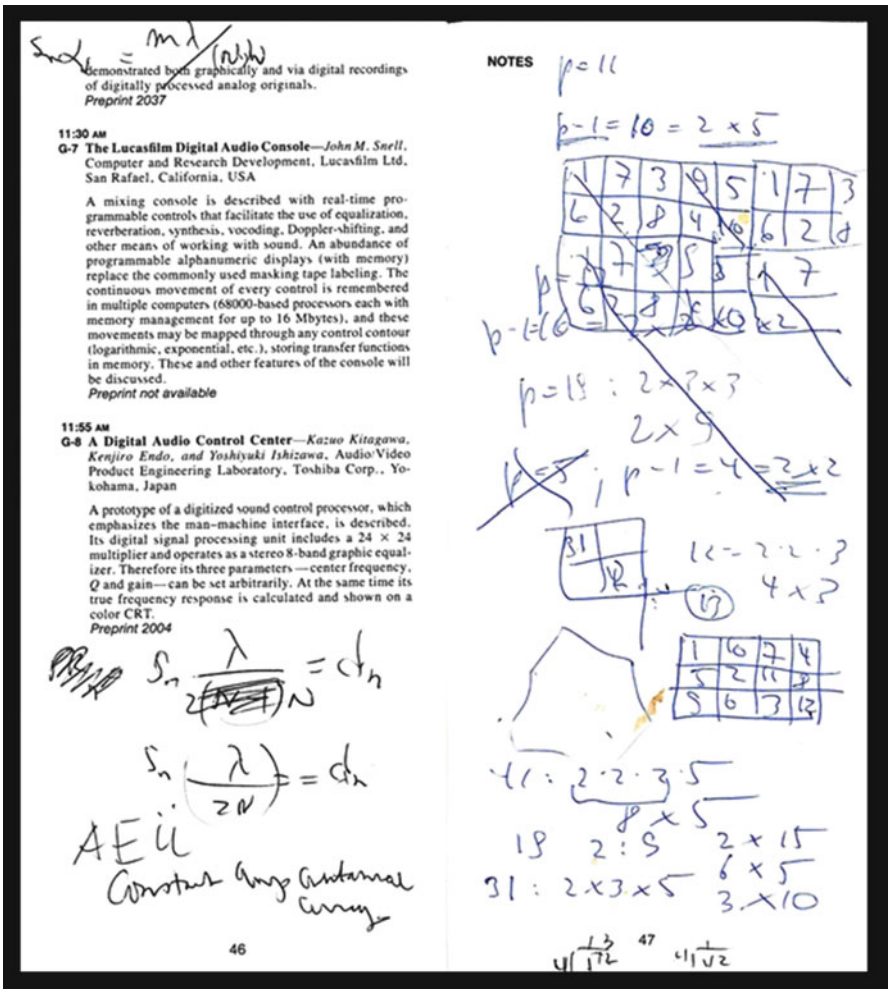


Fig. 9.6 Apple II program presented at the Studio Design Session C, which calculated the well depths and the diffraction pattern of a QRD

discovered quadratic residues, quadratic reciprocity and much more, with no application in mind. In 1975 Schroeder introduced number theory into the world of room acoustics from simple binary m-sequences to multivariate sequences, one of which being the quadratic residue sequence, with good autocorrelation properties and broader bandwidth. In his 1987 Rayleigh Lecture, the topic was "The Unreasonable Effectiveness of Number Theory in Science and Communication." Schroeder pointed out that in wave interference it is not the path differences that determine the interference pattern, but the residues after dividing by the wavelength.

The reflection phase grating has two fundamental properties, the incident sound is scattered into diffraction directions determined by the width of the period, NW, and the energy in the diffraction directions is equal, because the exponentiated well depths have a flat power spectrum. The first property of grating lobes from periodic surfaces is well established in optics, i.e., diffraction gratings. However, the second property, namely the uniformity of the energy in the diffraction lobes, is where Schroeder made one of his brilliant realizations. Flat surfaces reflect energy preferentially in one direction, the specular direction. Schroeder realized that one way to scatter sound uniformly into all of the diffraction lobes was to create a periodic scattering surface consisting of divided wells whose depths were based on the number theory sequences that Gauss developed, e.g., the quadratic residue sequence. These reflection phase gratings, based on quadratic residue sequences shown in Fig. 9.10, have the unique property that the energy in the seven diffraction


demonstrated both graphically and via digital recordings of digitally processed analog originals.
Preprint 2037

11:30 AM
G-7 The Lucasfilm Digital Audio Console—*John M. Snell*, Computer and Research Development, Lucasfilm Ltd., San Rafael, California, USA

A mixing console is described with real-time programmable controls that facilitate the use of equalization, reverberation, synthesis, vocoding, Doppler-shifting, and other means of working with sound. An abundance of programmable alphanumeric displays (with memory) replace the commonly used masking tape labeling. The continuous movement of every control is remembered in multiple computers (68000-based processors each with memory management for up to 16 Mbytes), and these movements may be mapped through any control contour (logarithmic, exponential, etc.), storing transfer functions in memory. These and other features of the console will be discussed.
Preprint not available

11:55 AM
G-8 A Digital Audio Control Center—*Kazuo Kitagawa, Kenjiro Endo, and Yoshiyuki Ishizawa*, Audio/Video Product Engineering Laboratory, Toshiba Corp., Yokohama, Japan

A prototype of a digitized sound control processor, which emphasizes the man–machine interface, is described. Its digital signal processing unit includes a 24 × 24 multiplier and operates as a stereo 8-band graphic equalizer. Therefore its three parameters—center frequency, *Q* and gain—can be set arbitrarily. At the same time its true frequency response is calculated and shown on a color CRT.
Preprint 2004

46

NOTES

47

Fig. 9.7 Schroeder's hand written notes in the back of the 74th Technical Meeting & Professional Exhibits Program describing the Chinese Remainder Theorem to me

directions shown is equal. In Fig. 9.11, d_n are the well depths based on the sequence values S_n, the wavelength λ and the prime N. k is the wavenumber, α_i and α_d are the angles of incidence and diffraction, $R(x)$ is the reflection factor, and the Fourier transform of the reflection factor yields a constant energy, $|p(k)|^2$ equal to $1/N$ for all of the diffraction orders. Another way to think of this is that the autocorrelation of the sequence values is zero except for the zero shift modulo N. And it is well known that the spectrum of a two valued autocorrelation is flat. In this case, the frequencies are spatial frequencies or directions.

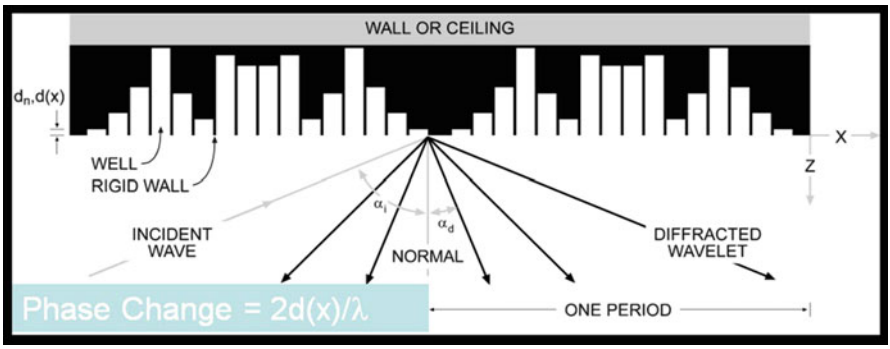


Fig. 9.8 Two periods of a QRD $N = 17$ diffusor showing incident and diffracted waves

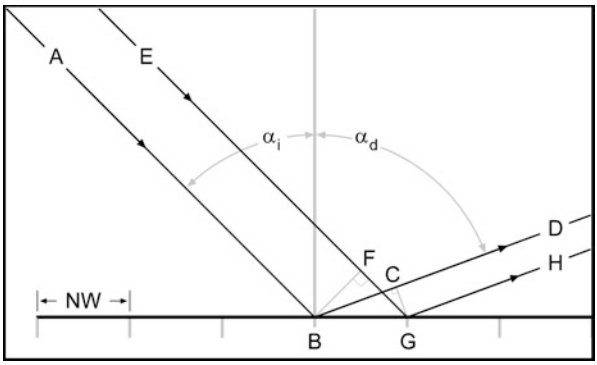


Fig. 9.9 Construction illustrating constructive interference condition in which BC-FG = mλ

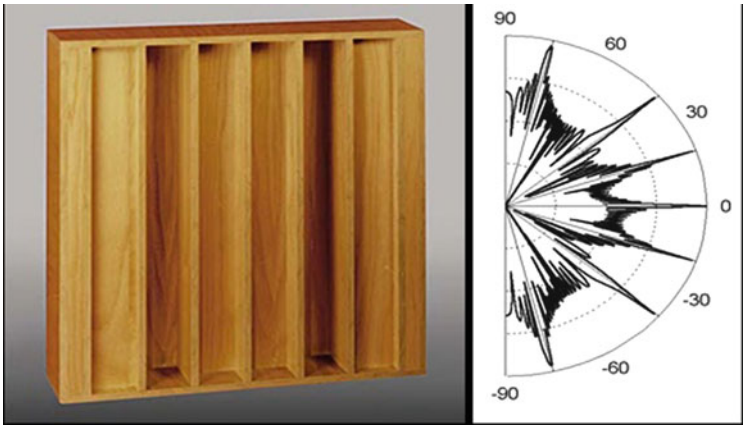


Fig. 9.10 *Left*: commercial QRD $N = 7$. *Right*: diffraction pattern at 3,000 Hz for 50 periods where the energy is concentrated in the diffraction directions (Taken from "*Acoustic Absorbers and Diffusers: Theory, Design and Application*", T.J. Cox and P. D'Antonio, Taylor & Francis, 2nd Edition (2009))

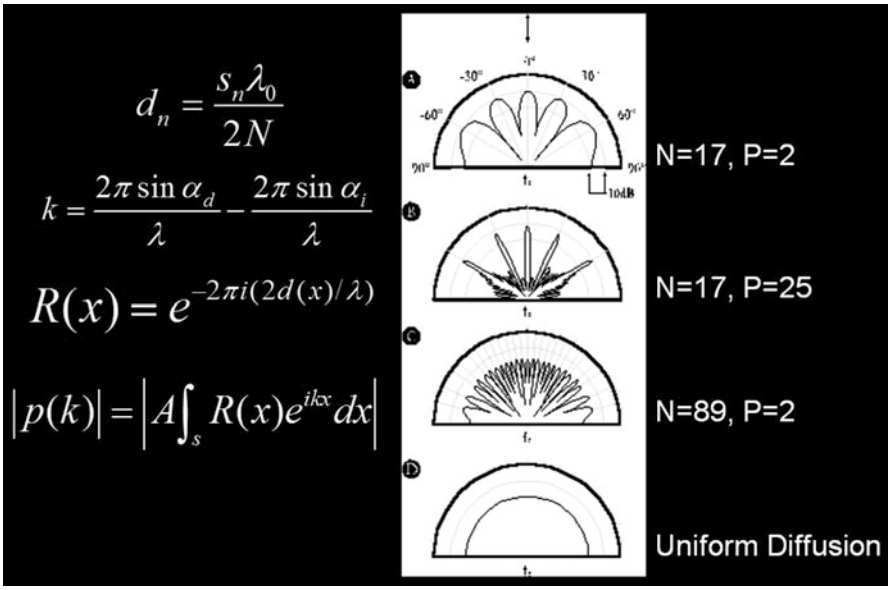


Fig. 9.11 Equations describing the well depths, d_n, wavenumber, k, reflection factor, $R(x)$, and scattered pressure, $p(k)$, along with the diffraction patterns for several prime numbers, N, and repeats, P

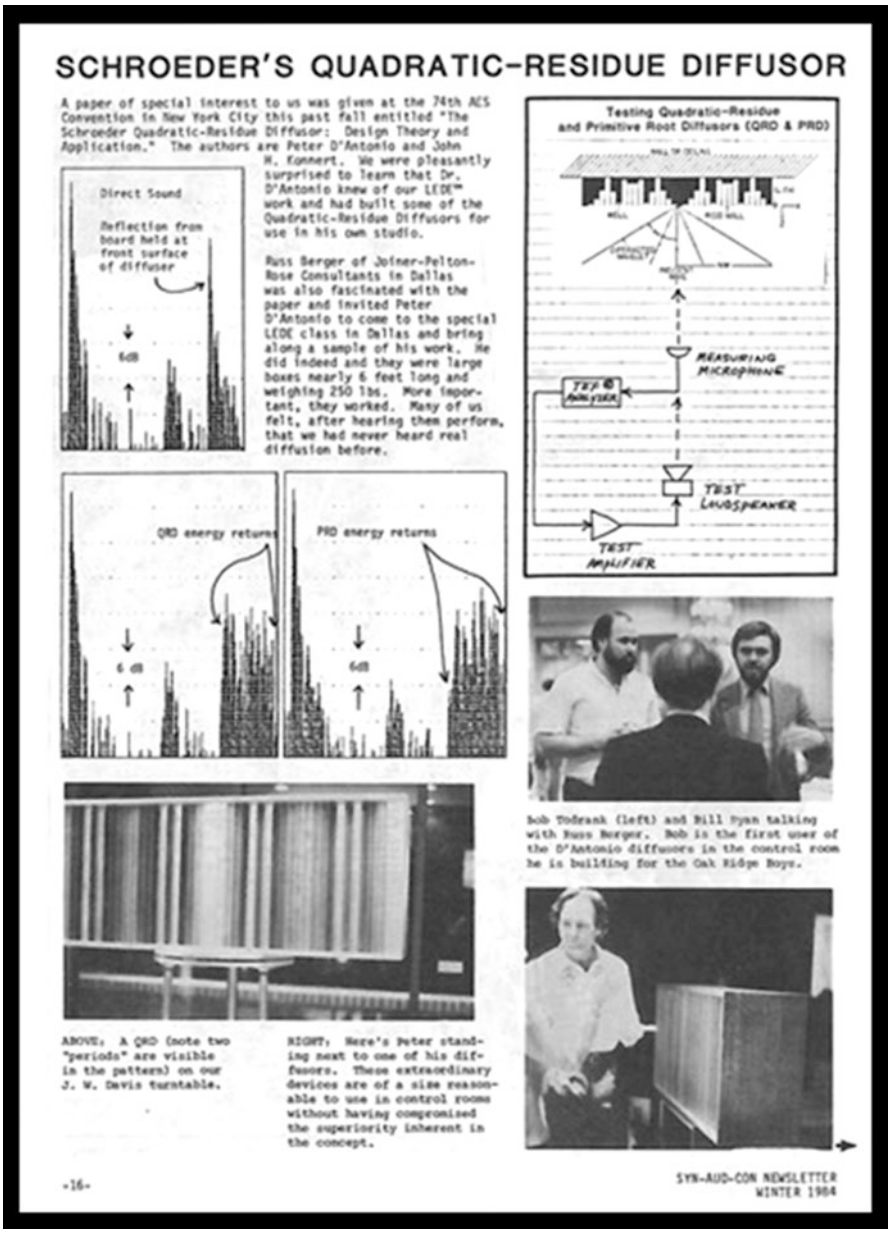
Following my presentation and meeting with Schroeder, I met Bob Todrank at an evening reception. Bob was designing a new studio for the Oak Ridge Boys in Hendersonville, TN and was interested in utilizing these new acoustical surfaces. The Oak Ridge Boy's Acorn Sound Recorders project, Fig. 9.12, was celebrated with a Syn-Aud-Con control room design workshop in 1984. This project was a resounding success and turned out to be a harbinger of many exciting things to come. It also led to many other projects and collaborations with a growing community of new studio designers. Use in recording studios soon led to broadcast studios, high end listening rooms, worship spaces, and eventually to performance spaces and schools.

In 1983, I carried out the first measurements of quadratic residue and primitive root diffusors with a TEF 10 analyzer at a Syn-Aud-Con seminar in Dallas, Texas, with the assistance of Don Eger of Techron, shown in Fig. 9.13. In 1984, an intensive measurement program was carried out using Richard Heyser's time delay spectrometry. Farrell Becker was very helpful in the initial evaluation of these exciting new surfaces. Not having access to an anechoic chamber, a boundary measurement technique was developed. These measurements were initially carried out at full scale in large spaces, like open fields and parking lots, eventually moving indoors to a sports arena, a motion picture sound stage, and a local high school gymnasium. The measurements enabled the theories to be validated.



Fig. 9.12 First application of commercial diffusors at Acorn Sound Recorders, Hendersonville, TN in 1984

It was clear that to properly evaluate these surfaces, a standard needed to be created. This turned out to be a 28-year process! The diffusion coefficient is now standardized as ISO 17497-2. The goal was to measure the scattered polar responses and extract from these data a diffusion coefficient, which was a measure of how uniformly these surfaces scattered sound versus frequency, as a complement to the absorption coefficient. In the early 1980s polar response measurements were made by measuring impulse responses one at a time, as shown in the left panel of Fig. 9.14, from a loudspeaker at a given angle of incidence to 37 microphones separated by 5°. This was an incredibly laborious process, but yielded polar responses that allowed evaluation of these early surfaces. As RPG began to grow from a cottage industry, it became necessary to make these measurements routinely, so a 1:5 scale boundary plane measurement goniometer was built, using a microphone switcher, shown in the right panel of Fig. 9.14. Under computer control, the TEF analyzer emitted 37 sequential MLS test signals and the switcher automatically switched to adjacent microphones. This was a great time savings and eliminated the need to constantly find large open spaces to make full scale measurements. As computer hardware evolved, it became possible in 2011 to measure all of the observation positions, for a given angle of incidence, simultaneously with one MLS test signal. This setup is shown in Fig. 9.15, using



SCHROEDER'S QUADRATIC-RESIDUE DIFFUSOR

A paper of special interest to us was given at the 74th AES Convention in New York City this past fall entitled "The Schroeder Quadratic-Residue Diffusor: Design Theory and Application." The authors are Peter D'Antonio and John H. Konnert. We were pleasantly surprised to learn that Dr. D'Antonio knew of our LEDE™ work and had built some of the Quadratic-Residue Diffusors for use in his own studio.

Russ Berger of Joiner-Pelton-Rose Consultants in Dallas was also fascinated with the paper and invited Peter D'Antonio to come to the special LEDE class in Dallas and bring along a sample of his work. He did indeed and they were large boxes nearly 6 feet long and weighing 250 lbs. More important, they worked. Many of us felt, after hearing them perform, that we had never heard real diffusion before.

Direct Sound

Reflection from board held at front surface of diffuser

6dB

Testing Quadratic-Residue and Primitive Root Diffusors (QRD & PRD)

MEASURING MICROPHONE

TEF® ANALYSER

TEST LOUDSPEAKER

TEST AMPLIFIER

QRD energy returns

PRD energy returns

6 dB

6dB

Bob Todrank (left) and Bill Ryan talking with Russ Berger. Bob is the first user of the D'Antonio diffusors in the control room he is building for the Oak Ridge Boys.

ABOVE: A QRD (note two "periods" are visible in the pattern) on our J. W. Davis turntable.

RIGHT: Here's Peter standing next to one of his diffusors. These extraordinary devices are of a size reasonable to use in control rooms without having compromised the superiority inherent in the concept.

-16-

SYN-AUD-CON NEWSLETTER
WINTER 1984

Fig. 9.13 First measurements of QRD and PRD commercial diffusors, using the TEF Analyzer in the Winter 1984 issue of the Syn-Aud-Con Newsletter

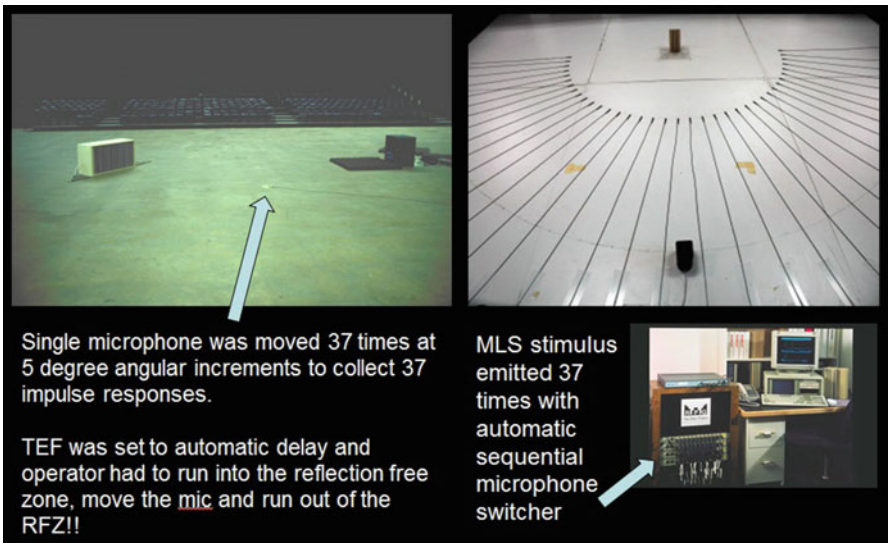


Fig. 9.14 *Left*: the first goniometer measurements were made full scale one at a time with a TEF analyzer with a microphone radius of 5 m and a speaker radius of 10 m. *Right*: in 1994 a 1:5 scale goniometer was built with a 1 m mic radius and a 2 m speaker radius

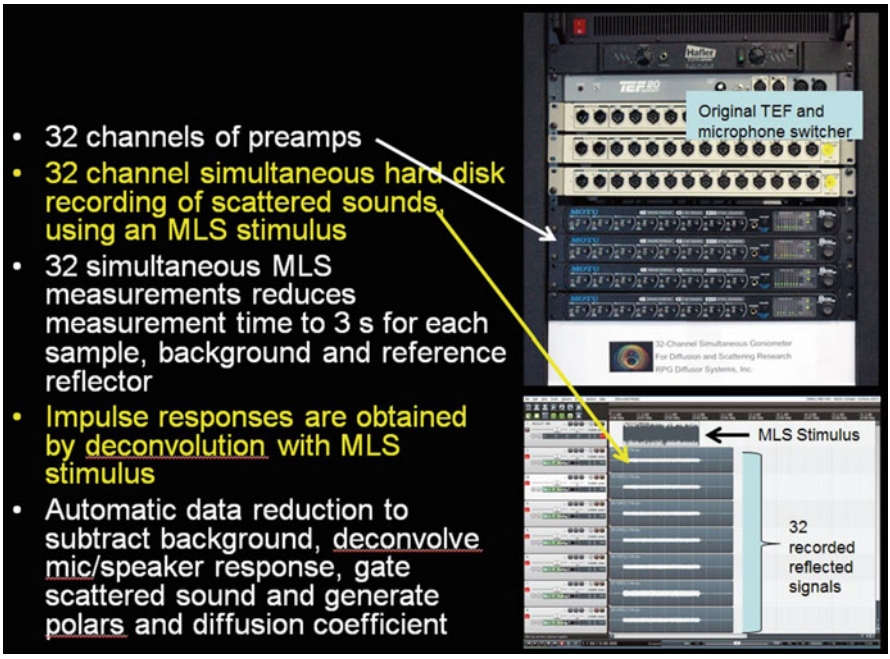


Fig. 9.15 *Top*: original TEF and microphone switcher followed by the 32 MOTU preamps; *Bottom*: Reaper screen illustrating the MLS Stimulus at the top and a few of the 32 recorded reflected signals below

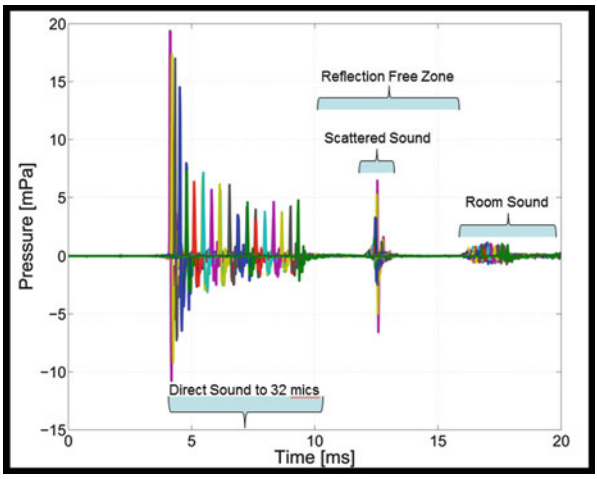


Fig. 9.16 Impulse responses for the 32 direct sounds and scattered reflections, along with the room sound interference

32 microphones. Each microphone was connected to a MOTU preamp and the Firewire output was sent to a computer, which recorded the scattered MLS signals on hard disk. The scattered signals were deconvolved, using the MLS test stimulus, to obtain the impulse responses. The impulse responses for the 32 direct sounds and scattered reflections are shown in Fig. 9.16, along with the room sound interference. The scattered sound was extracted via a multistep process illustrated in the left panel of Fig. 9.17, in which a background response, $h2(t)$, with no sample present, is subtracted from the full impulse response, $h1(t)$, to minimize the direct sound and interfering room reflections. $h2(t) - h1(t)$ is then deconvolved with the loudspeaker/microphone response, $h3(t)$, to yield $h4(t)$ and windowed to isolate the scattered impulse response. In the right panel of Fig. 9.17, we show the entire process leading to the diffusion coefficient. (A) the goniometer with a speaker at 150 degrees, (B) the total impulse response at one microphone position, with the scattered sound outlined, (C) the isolated impulse responses at all microphone positions, (D) 5 selected Fourier transforms of the scattered impulse responses and three selected 1/3rd octave polar responses, (E) the diffusion coefficient obtained from the circular autocorrelation of these polar responses, without and with normalization. To remove edge diffraction, the diffusion coefficient of the sample is normalized by the diffusion coefficient of the reference reflector. In Fig. 9.18, we show a photo of a test sample, in this case, three hemicylinders, the diffusion coefficient for the sample and reference reflector and the normalized diffusion coefficient for normal incidence. Below we show the 1/3rd octave polar responses for the sample (red) and the reference reflector (blue).

While the QRD was revolutionary, there were three aspects that we investigated to improve performance, shown in Fig. 9.19. These included extending the bandwidth, minimizing the effect of grating lobes, i.e., making the response uniform, and lastly

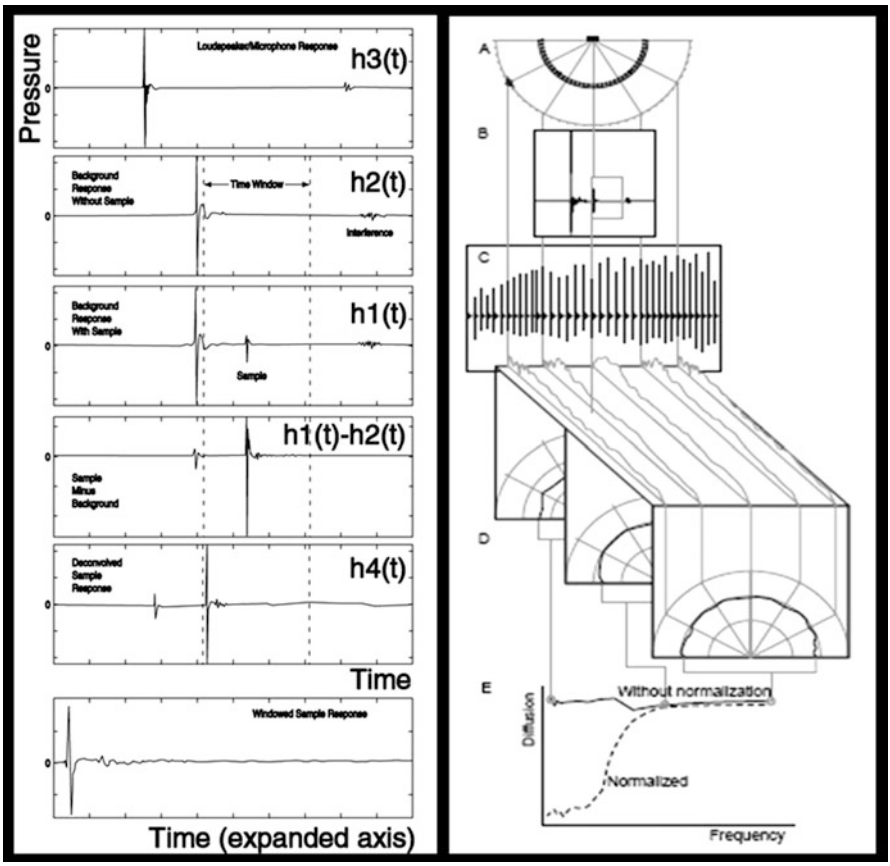


Fig. 9.17 *Left*: data reduction procedure to extract scattered sound at a given microphone position from a given angle of incidence; *Right*: complete process to determine the diffusion coefficient from the scattered impulse responses (Taken from "*Acoustic Absorbers and Diffusers: Theory, Design and Application*", T.J. Cox and P. D'Antonio, Taylor & Francis, 2nd Edition (2009))

eliminating the quantized well depth effect, which results in a specular reflection at the frequency where all wells scatter in phase. For a QRD, these frequencies occur at integer multiples of the prime multiplied by the design frequency.

When considering how to expand the bandwidth, I was intrigued by the idea of the self-similarity of fractals and proposed nesting, scaled versions of the QRD forming a self-similar design, in which each generation of nesting would cover different frequency ranges [P. D'Antonio, "A new 1 or 2-dimensional fractal sound diffusor," J. Acoust. Soc. Am., Suppl. 1, 87, S10]. On my way to an Acoustical Society meeting at Penn State, I accidentally met Manfred Schroeder in the Philadelphia airport and we flew together to the meeting. During the flight we discuss many things, including how his diffusors were being accepted, the success of RPG and the diffusing fractal, now called a Diffractal, which fascinated him. He then contacted Freeman, the publisher of his forthcoming book, *Fractals, Chaos,*

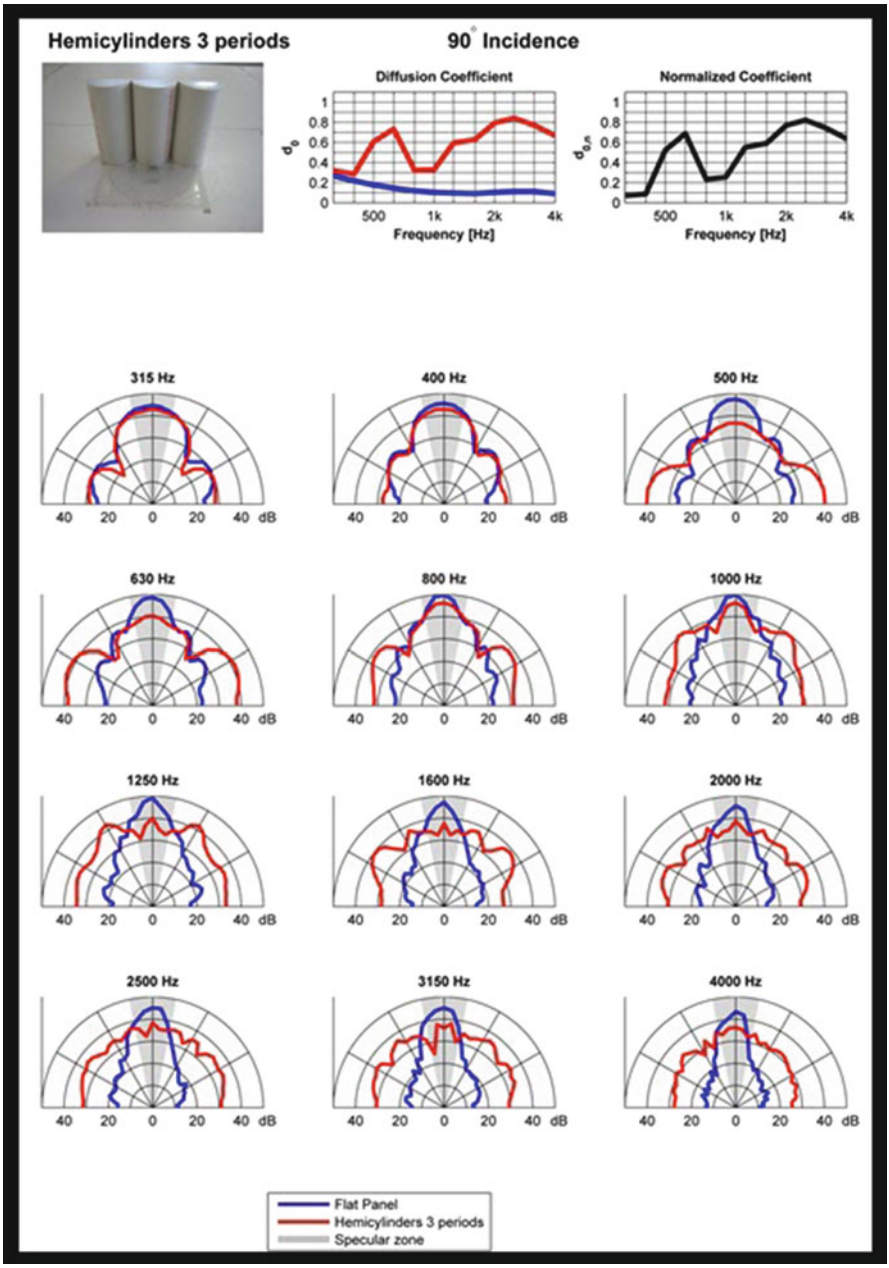


Fig. 9.18 *Top*: photo of 3 hemicylinders, diffusion coefficient of the sample (*red*) and reference reflector (*blue*), normalized diffusion coefficient of sample (*black*). *Bottom*: third octave polar responses of the sample (*red*) and reference reflector (*blue*)

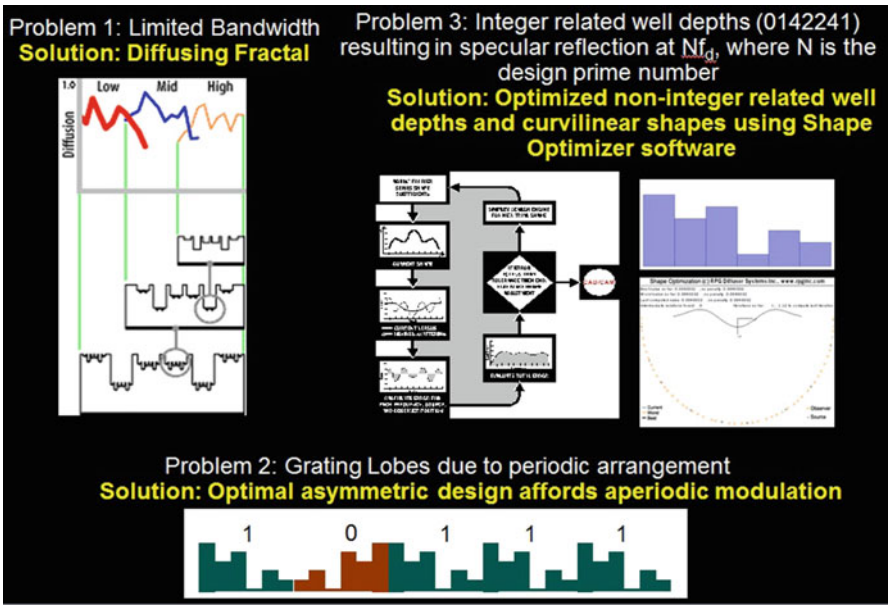


Fig. 9.19 Three problems that were mitigated to optimize the phase grating

Power Laws: Minutes from an Infinite Paradise, and asked them to include the statement at the bottom of Fig. 9.20 prior to publication. During a subsequent Audio Engineering Society Convention in NY, Manfred Schroeder visited me at the RPG booth and he can be seen pointing to a Diffractal with product literature in hand.

The second problem is associated with grating lobes and is very ironic, because the QRD is based on the concept of periodicity, using number theory sequences which insure equal energy in the diffraction directions. Yet to achieve uniform scattering in all directions, a way had to be found to minimize grating lobes! The scattered polar responses in Fig. 9.18 are dominated by grating lobes generated by the fact that the diffusors are periodic. The lobe energy may be constant, but there are large minima between the lobes, except at high frequencies when the number of lobes becomes very large. For this reason, significantly better performance can be obtained if the periodicity lobes can be removed by making the diffusor aperiodic or increasing the repeat distance. It seemed the QRD was cursed by periodicity. James Angus came up [J.A.S. Angus, "Large area diffusors using modulated phase reflection gratings," Proc. 98th Convention Audio Eng. Soc., Preprint 3954, D4 (1995)] with a solution in a series of papers outlining methods for using two phase grating base shapes in a modulation scheme to minimize periodicity. Another approach Trevor Cox and I developed is to form an asymmetric QRD sequence and instead of repeating it periodically, one would follow the prescription of an optimal binary sequence whose aperiodic Fourier transform is as flat as possible.

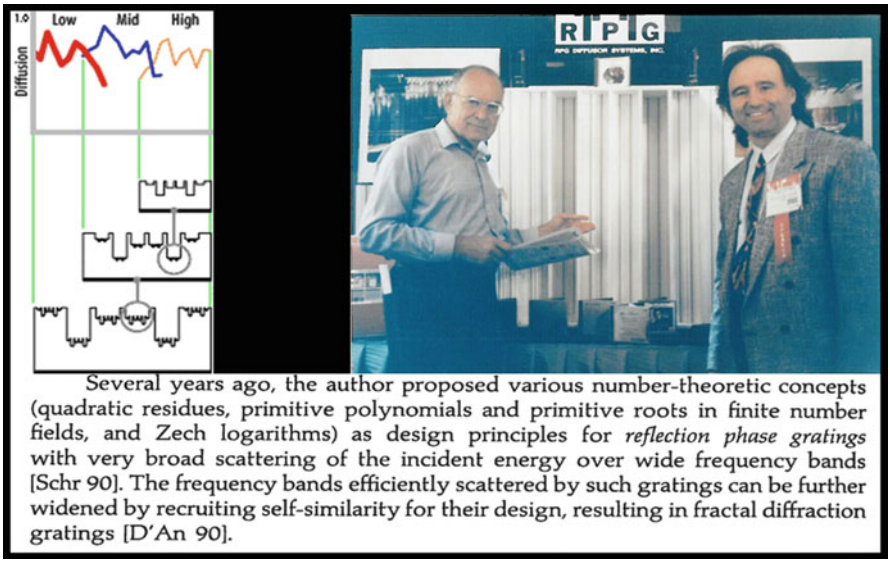


Fig. 9.20 Manfred Schroeder and Peter D'Antonio in the RPG booth at AES in New York in front of a Diffractal along with a mention in his new book with reference [D'An 90]

That is, if the binary sequence is a zero, the base shape is used, if the sequence value is one, the asymmetric QRD is flipped. In this way an aperiodic modulation is formed and grating lobes are minimized.

The last problem deals with specular scattering at a frequency where all of the wells scatter in phase. For the QRD, this frequency is equal to integer multiples of the prime times the design frequency. This occurs because the well depths are integer multiples of one another. To minimize these flat plate frequencies an optimization program was created, in collaboration with Trevor Cox, that combined boundary element prediction, multidimensional optimization techniques, and the diffusion coefficient. It is an iterative program which cycles until the shape produces a desired diffusion coefficient. When used for divided wells or nondivided steps, the goal is to find non-integer related wells or steps, thus avoiding the flat plate frequencies. It can also be used to define a wide range of curvilinear shapes which can complement contemporary architecture, shown in Fig. 9.21.

Thus by utilizing a variety of techniques including, boundary element prediction, multidimensional minimization, defining a diffusion coefficient, fractal geometry, optimal aperiodic modulation, etc. we have been able to optimize Schroeder's seminal idea of an acoustical reflection phase grating. This research has yielded essentially three types of diffusive surfaces, optimized and modulated reflection

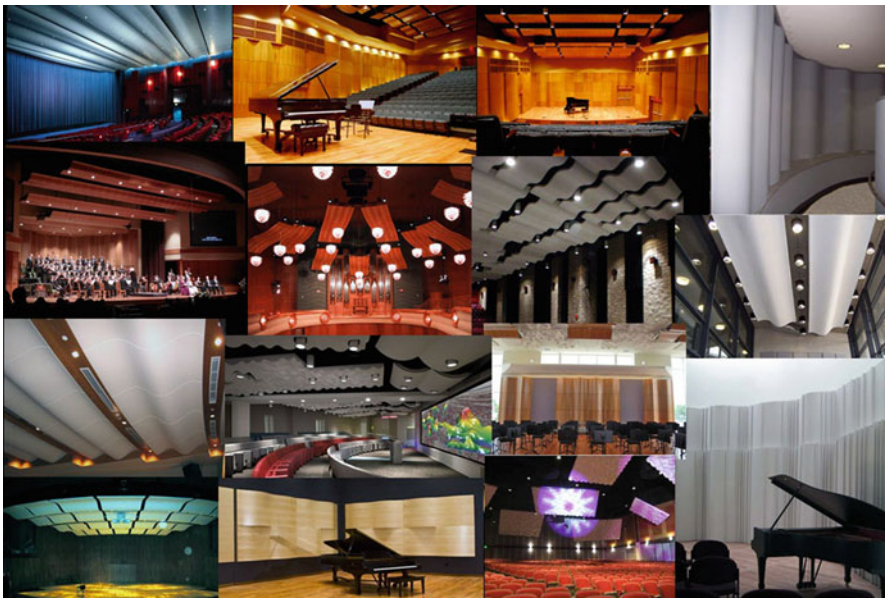


Fig. 9.21 A wide variety of curvilinear diffusive shapes obtained with the Shape Optimizer software

phase gratings, planar binary absorption-reflection amplitude gratings, and optimized curvilinear shapes, seen in Fig. 9.22.

It was a great pleasure to meet Manfred Schroeder a few additional times. One was in Rome at the ICA, where Michael Vorlander convened a special session on diffusors. It was a great privilege for me to present a paper outlining the progress we had made in optimizing his Schroeder diffusor. Unfortunately he was recovering from a stroke, but was still in good spirits. Our last meeting was at an ASA meeting in Paris, where we shared some drinks in the hotel lobby and met again briefly in the Louvre.

As a small way to thank Manfred Schroeder for inspiring us and launching our careers in acoustics, Trevor Cox and I dedicated our book *Acoustic Absorbers and Diffusers: Theory, design and application* to him. His thank you letter and a photo of the Second Edition are seen in Fig. 9.23.

Manfred Schroeder had many hobbies and cycling was one of them. For comic relief, I compiled a collage of Trevor Cox, James Angus, Manfred Schroeder, and myself entitled Diffusor Docs, in Fig. 9.24. As a fitting tribute to Manfred Schroeder, the RPG Diffusor was inducted into the music industry's Technology Hall of Fame, in 2013. My closing sentiments are expressed in Fig. 9.25.

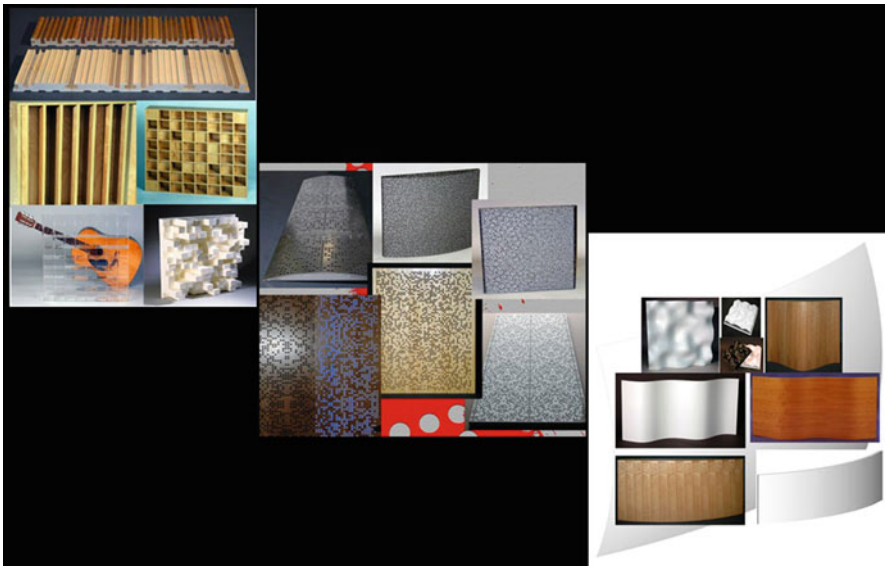


Fig. 9.22 Three types of diffusors that evolved from Schroeder's seminal suggestion. *Left* to *right*: a family of 1D and 2D reflection phase grating diffusors; a family of 2D binary amplitude diffusor-absorbers (diffsorbers); and a family of wood and glass reinforced gypsum curvilinear optimized diffusors

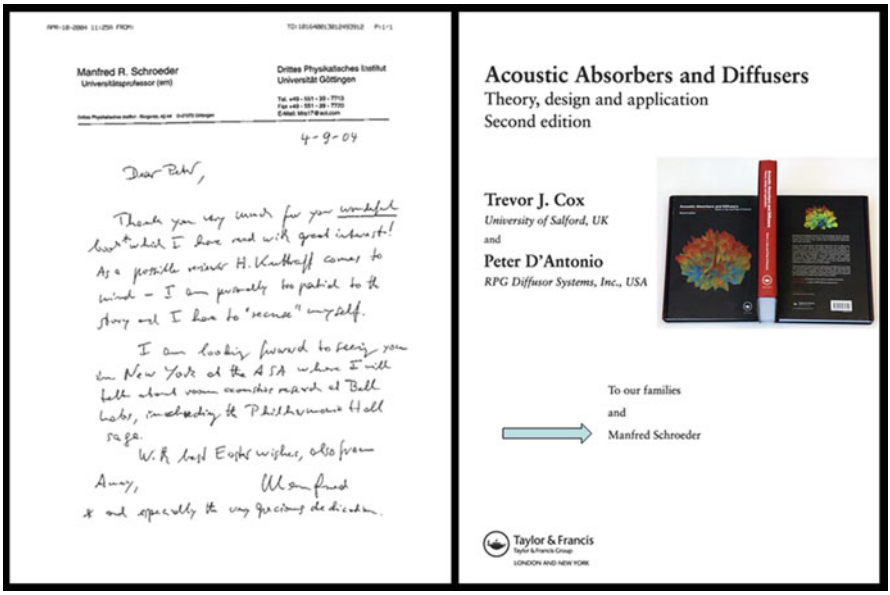


Fig. 9.23 A photo of a book written by Trevor J. Cox and Peter D'Antonio, with a dedication to Manfred Schroeder, along with his thank you letter



Fig. 9.24 Diffusor Docs: *Top left* clockwise, Manfred Schroeder, Trevor Cox, James Angus, and Peter D'Antonio

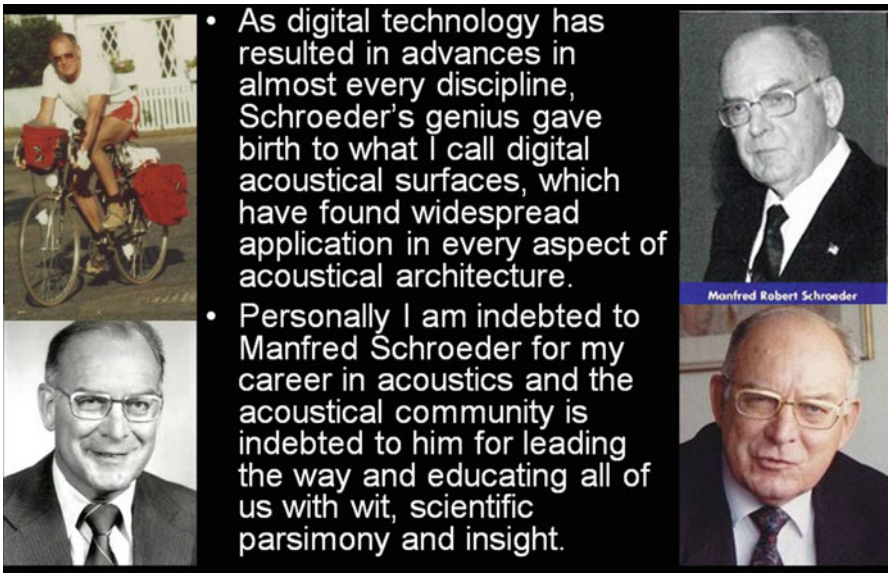


Fig. 9.25 Vielen Dank und Abschied

***

Biography



Dr. **Peter D'Antonio** was born in Brooklyn, NY in 1941 and is Founder/Chairman of RPG Diffusor Systems, established in 1983. He has co-authored *Acoustic Absorbers and Diffusers: Theory, Design and Application,* 2nd Edition, Taylor & Francis 2009 and contributed several chapters to the *Master Handbook of Acoustics,* 5th Edition, McGraw Hill 2009. Dr. D'Antonio served as Chairman of the AES SC-04-02 and is a member of ISO 17497-1 and ISO 17497-2. He is an adjunct professor of acoustics at the Cleveland Institute of Music, since 1991 and a Fellow of the Acoustical Society of America and the Audio Engineering Society.

Chapter 10
Manfred R. Schroeder: Challenge the Present and There Is a Better Way

Bishnu S. Atal

Abstract The chapter author worked with Manfred R. Schroeder for about 25 years at Bell Telephone Laboratories starting in May 1961. Manfred always inspired this author to look for things one has never seen before. Digital computers were emerging in the 1960s with great promise and we used computers to do many new things: simulation of acoustics of concert halls, studying sound decay in rooms using ray tracing, and producing high-quality speech at low bit rates. This chapter will highlight a few examples. Manfred had been deep in the area of vocoder research at that time, but we started on a new track leading us ultimately to code-excited linear prediction. This set the spark for expanding the use of cell phones worldwide. As a manager, Manfred Schroeder was uncompromising in pushing for scientific excellence resulting in success; Bell Labs had the reputation of being the best.

10.1 Introduction

I worked with Manfred Schroeder for about 25 years at Bell Telephone Laboratories starting in May 1961. Manfred always inspired me to look for things one has never seen before. Digital computers were emerging in the 1960s with great promise and we used computers to do many new things: simulation of acoustics of concert halls, studying sound decay in rooms using ray tracing, and producing high-quality speech at low bit rates. This chapter will highlight a few examples. Manfred had been deep in the area of vocoder research at that time, but we started on a new track leading us ultimately to code-excited linear prediction (CELP). This

B.S. Atal (✉)
Department of Electrical Engineering, University of Washington, Seattle, WA, USA
e-mail: bsatal@bishnu.net

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_10,
© Springer International Publishing Switzerland 2015

set the spark for expanding the use of cell phones worldwide. As a manager, Manfred Schroeder was uncompromising in pushing for scientific excellence resulting in success; Bell Labs had the reputation of being the best.

10.2 Computer Simulation of Acoustics of Concert Halls

In early 1960s, digital computers were primarily used for solving numerical problems. However, it was becoming increasingly obvious that computer simulation was a powerful tool for understanding complicated processes. Manfred Schroeder was a pioneer in using such tools. In one of the early applications of computer simulation, a digital computer (IBM 7099) was programmed to "simulate" sound transmission in rooms [1]. The idea was that one could evaluate the "acoustics" of a new concert hall even before the hall was built. The digital simulation could help to "preaudit" architectural designs before construction and to investigate subjective correlates of a wide variety of reverberation processes. If a design was found to be unsatisfactory, as a result of subjective evaluations of the simulation, modifications could be made in the architectural plans.

Thus, one could start with reverberation-free signals and program a digital computer to add echoes and reverberation with specified delays, spectral content, and decay characteristics. Several output signals could be generated on the computer; these signals when radiated from loudspeakers in an anechoic chamber could produce sound pressure waves at listener's ears resembling those in real halls. Since we have only two ears to listen with, it should be possible to accomplish the result by radiating from only two loudspeakers. If done properly, the acoustical cross-talk between two loudspeakers at the two ears can be eliminated.

Figure 10.1 illustrates the sequence of pulses to be emitted from the two loudspeakers to achieve the desired response from a source *S* on the right. The details of these operations are contained in [1]. Sound arrivals from about ten different directions were created on the computer and reproduced in the anechoic chamber at Bell Laboratories by means of two loudspeakers located several meters in front of the listener. Although the computations were based on an essentially fixed head position, the experiments showed that head movements up to $\pm 10^{\circ}$ did not affect the percept; "externalization" (perceiving the sound as originating outside one's head) was perfect. In fact, the synthesis of lateral arrival, including arrivals from $\pm 90^{\circ}$ was so perfect that listeners often turned their heads to see the nonexistent sound source.

10.3 Study of Sound Decay in Rooms Using Ray Tracing on Computers

Shortcomings of classical reverberation time formulas, named after Sabine, Eyring, and Millington, to predict reverberation time correctly or to calculate absorption coefficients accurately have been known for a long time. Occasionally, one

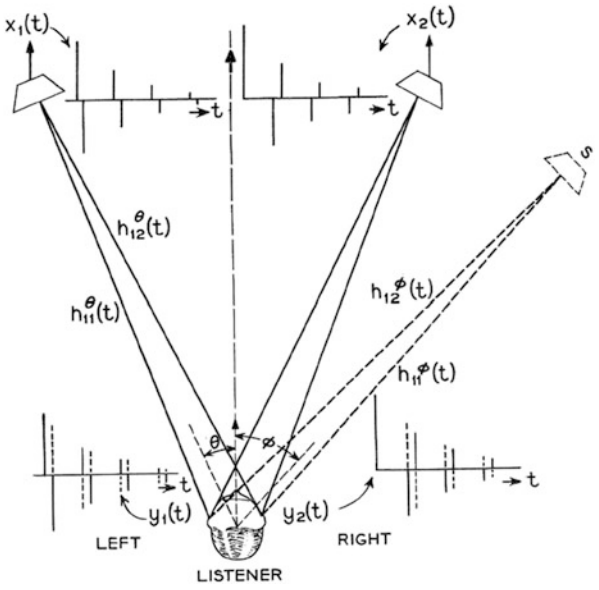


Fig. 10.1 Creation of virtual sound image at an arbitrary lateral direction by means of two loudspeakers in front of the listener. The signals $x_1(t)$, $x_2(t)$ from the two loudspeakers combine at the listener's ears to resemble those from a real lateral sound source S

obtained absorption coefficients greater than 1, a physically impossible result! Such problems were often ascribed to "lack of diffusion" in the reverberation chambers. Manfred and I felt that the basic assumptions that go into these formulas were incorrect.

Another argument raised by many people then was that the existing reverberation theories were based on ray approximation. Since, sound does not travel like rays, no wonder, theory and experimental results do not agree. It turned out that the existing reverberation theories were not correct even under the ray approximation [2, 3].

In order to develop an accurate theory of reverberation based on ray acoustics, I computed sound decays in two-dimensional "rooms" of different shapes and sizes by simulating the propagation of rays on a digital computer. The method of ray tracing on a digital computer is shown in the Fig. 10.2. It shows a two-dimensional enclosure with a patch of absorbing material on one of its walls and an omnidirectional sound source emitting 300 equal-energy rays spaced 1.2° apart (although any other directional characteristic could have been used). The computer program tracked the path of each of the 300 rays following the laws of geometrical acoustics and the energy of each of these rays as they traveled in the room. Each time a ray hit the wall, its energy was diminished by a factor $(1-\alpha)$, where α is the absorption coefficient. The computer program could follow a ray over any desired length of time. The total sound energy after the introduction of a short pulse of sound in the enclosure was obtained by adding the energies of all the component rays.

An example of the sound decay in a two-dimensional non-rectangular enclosure with one of the walls covered with an absorbing material with $\alpha = 0.8$ is shown in

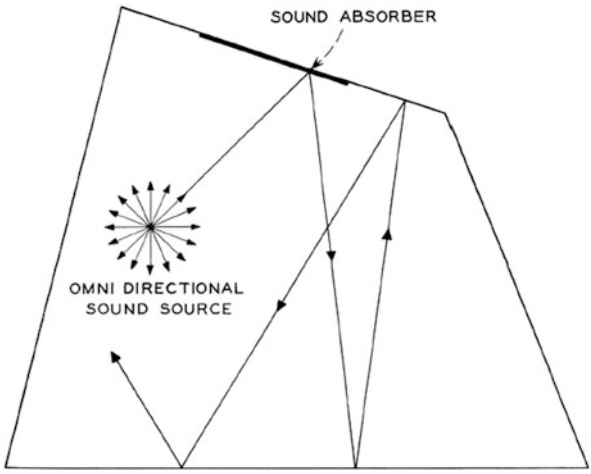


Fig. 10.2 The computer calculates the path of 300 rays from an omnidirectional source. Every time a ray hits an absorbing material, its energy is reduced depending on the absorption coefficient of the material. The computer keeps a running account of the remaining energy

Fig. 10.3. The figure shows a comparison between the decay obtained on the computer (reverberation time $T = 0.38$ s) and the decays predicted by the two-dimensional forms of the Sabine ($T = 0.63$ s), Eyring ($T = 0.56$ s), and Millington formulas ($T = 0.31$ s), respectively. The reverberation time predicted by the Sabine formula is 65 % too large. The Millington value is 17 % too small. The Eyring formula provides an intermediate value with an error of +45 %.

I made a modest beginning in developing a new theory of sound decay in rooms. I will give here a brief outline of steps that provide a more accurate representation of how sound decays in rooms. For simplicity, the case of absorbing material on a single wall is considered. The theory can easily be generalized to case when the absorbing material is spread over more than one wall. Let $\varepsilon(\alpha, t)$ represent the total sound energy in the room at a time t sec after the sound source is stopped. It is easily seen that $\varepsilon(\alpha, t)$ is written as

$$\varepsilon(\alpha, t) = \sum_{k=0}^{\infty} (1-\alpha)^k P(k, t) \tag{10.1}$$

where $P(k, t)$ is the probability of a ray hitting the sound absorber k times in t sec. The logarithmic decay $\ln \varepsilon(\alpha, t)$ can be expressed in terms of cumulants $q_n(t)$ of the probability distribution $P(k, t)$. Thus,

$$\ln \varepsilon(\alpha, t) = \overline{n}(t)\ln(1-\alpha) + \sum_{k=2}^{\infty} \frac{q_k(t)}{k!}[\ln(1-\alpha)]^k \tag{10.2}$$

where $\overline{n}(t) = \sum_{n=0}^{\infty} nP(n, t)$. All of the classical reverberation time formulas can be derived from Eq. (10.2) by substituting the cumulants appropriate to the probability distribution implicit in the formulas. For example, Sabine's formula is

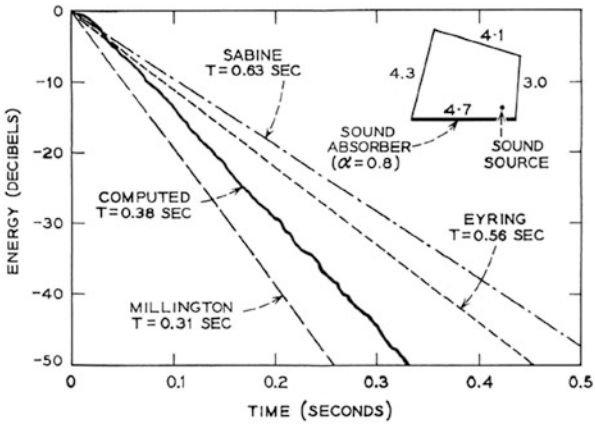


Fig. 10.3 Comparison of sound decay curves found by computer ray-tracing with decay rates predicted by the Sabine, Eyring, and Millington formulas

obtained by setting all the cumulants equal to $\overline{n}(t)$. The above equations can be easily generalized when more walls are covered with absorbing material. The probability distribution and cumulants are replaced by the joint distribution and joint cumulants.

Further analysis of Sabine and Eyring formulas has revealed that the underlying probability distributions implicit in these formulas are based on the assumption that the probability that a ray hits a given wall is independent of the wall the ray came from. This cannot be correct, since a ray cannot hit the same wall twice in succession. The probability of hitting a wall at any reflection depends upon the entire past history of the ray. One can thus regard the Eyring formula as equivalent to a zero-order Markov process. A natural way of improving the Eyring formula will be to consider the effect of taking the first-order Markov dependence into account in developing the theory.

10.3.1 Reverberation Theory Based on First-Order Markov Dependence

Here is a brief description of the new theory. Consider a room with K walls. Let α_i be the absorption coefficient of the absorbing material on wall i. Let p_{ij} be the probability that a ray starting from the wall i hits the wall j at the next reflection. It is reasonably straightforward to compute these probabilities from the geometry of the room. Consider now a matrix $\mathbf{P}$ with its term in the row i and column j given by p_{ij} and a diagonal matrix $\mathbf{A}$ with its i-th diagonal term given by $1 - \alpha_i$. Then, each element of the matrix $\mathbf{PA}$ is nonnegative and the matrix $\mathbf{PA}$ has an eigenvalue $\lambda_1(\alpha_1, \ldots, \alpha_K)$ which is greater than the absolute value of every other eigenvalue of the matrix $\mathbf{PA}$ [4]. The reverberation time is given by

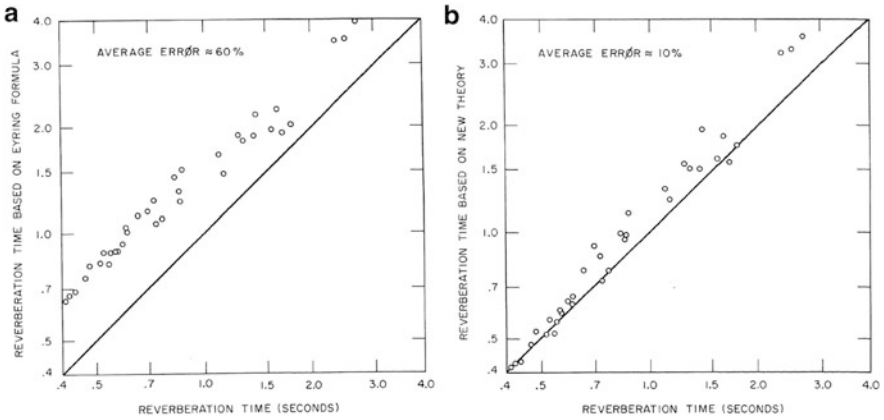


Fig. 10.4 (**a**) Comparison of reverberation times, obtained by computer ray-tracing (abscissa), with reverberation times computed by Eyring formula for 36 different shapes (ordinate) and (**b**) comparison of reverberation times, obtained by computer ray-tracing, with reverberation times computed using the new theory based on first-order Markov dependence for the same 36 shapes used in (**a**)

$$T(\alpha_1, \ldots, \alpha_K) = \frac{-13.82}{\overline{n}\ln[\lambda_1(\alpha_1, \ldots, \alpha_K)]} \tag{10.3}$$

where $\overline{n}$ is the mean collision frequency of the rays. Equation (10.3) reduces to Eyring formula if p_{ij} is independent of i (Fig. 10.4).

10.4 Early Predictive Coding Research

The research in speech coding (or speech bandwidth compression) originated about 75 years ago. At that time, the demand for telephones was growing rapidly. Telephone service was available coast-to-coast, with close to ten million telephones in service. The first under-the-ocean transatlantic telephone cable was still years away. Western Electric asked its research engineers at Bell Telephone Laboratories if voice signals could be transmitted over existing telegraph cables.

The bandwidth required for voice transmission is approximately 3,000 Hz, but the transatlantic telegraph cables had a bandwidth of only a few hundred Hz. Homer Dudley, who had joined Western Electric (parent of Bell Laboratories) in 1921, argued that the real information in speech was carried by telegraph-like low-frequency modulation signals corresponding to slow motion of the vocal organs. Dudley implemented his ideas in a device named "Vocoder," but the signal processing technology available then was primitive and it was very difficult to realize the full potential of his ideas. Although vocoders did not produce speech of quality good enough for telephone application, they were used during World War II by President

Roosevelt and Prime Minister Churchill for secure voice communication. Vocoders remained the central theme of speech coding research for about 35 years.

During the time speech scientists were occupied with vocoders, several major developments of fundamental importance were taking place outside speech research. A major contribution to the mathematical theory of communication of signals was made by Claude Shanon, represented in his published work in 1948 [5]. Shanon's work established a mathematical framework for coding and transmission of signals. In the 1940s, Norbert Wiener developed a mathematical theory for calculating the best filters and predictors for detecting signals hidden in noise.

Wiener worked during the Second World War on the problem of aiming anti-aircraft guns to shoot down an airplane. Since the plane moves, one must predict its position at the time the shell will reach the plane. Wiener's work appeared in his famous monograph [6], "Extrapolation, Interpolation and Smoothing of Time Series," published in 1949.

Linear Prediction is now a major tool in signal processing. Another development was Pulse Code Modulation (PCM), a method for sampling a continuous signal and quantizing each sample into binary digits. It enabled coding of speech with telephone bandwidth at a bit rate of 64 kb/s with negligible loss of quality. PCM marked the beginning of the digital age.

Following the work of Shanon and Wiener, Peter Elias published two papers in 1955 [7] on predictive coding. In predictive coding, both the transmitter and the receiver store the past values of the transmitted signal and from them predict the current value of the signal. The transmitter transmits, not the signal, but the prediction error, that is the quantized difference between the signal and its predicted value. At the receiver, this transmitted prediction error is added to the predicted value to reproduce the signal. For efficient coding, the successive terms of the prediction error should be uncorrelated and the entropy of its distribution should be as small as possible. In predictive coding, the transmitted prediction error is made as white as possible (Fig. 10.5) to achieve coding efficiency. Predictive coding is a remarkably simple concept.

In 1965, as part of my Ph.D. course work at the Polytechnic Institute of Brooklyn, New York, I had to review two papers of my choice and present the material in the class. One of them was Peter Elias's paper on predictive coding, published in 1955. Just a few months later, in 1966, I was one day in Manfred Schroeder's office at Bell Labs when John Pierce brought a tape showing a new speech time compression system. Schroeder was not impressed. After listening to the tape, he said that *there had to be a better way* of compressing speech and mentioned the work in image coding by Chape Cutler at Bell Labs based on differential pulse code modulation (DPCM) technique, which was a simplified version of predictive coding. Our discussions that afternoon kept me thinking. Schroeder's remarks at our meeting made a deep impression. Waiting at the subway station for a train to Brooklyn, I convinced myself that I should do some exploratory investigation to determine if predictive coding could work for speech signals. A first step was to find out if the first-order entropy of the distribution of prediction error signal is significantly smaller than the corresponding entropy of the speech signal; indeed, smaller entropy of the prediction error could produce a lower bit

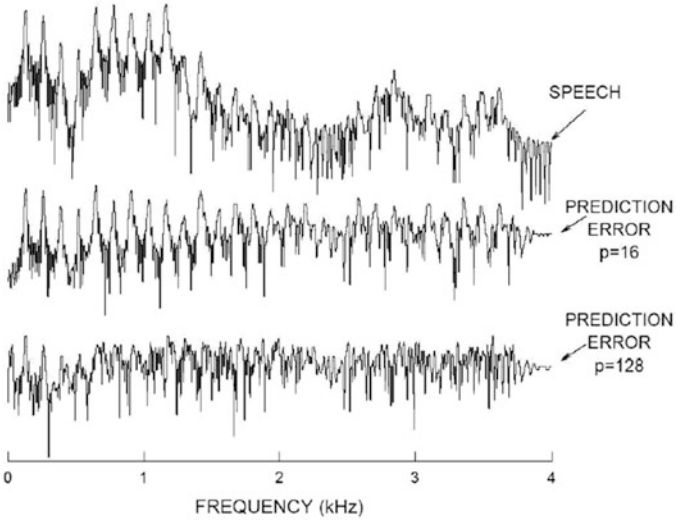


Fig. 10.5 Spectrum of the speech signal (first waveform), the spectrum of the prediction error after prediction with a linear predictor of order 16 (second waveform), and the spectrum of the prediction error after prediction with a linear predictor of order 128 (third waveform). The speech signal was sampled at 8 kHz. The 128th-order linear predictor was able to remove most of the pitch-related fine structure of the signal

rate. The results were encouraging. For speech sampled at 6.67 kHz, the first-order entropy of prediction error turned out to be 1.3 bits/sample as compared to 3.3 bits/sample for the speech signal.

Of course, for a signal that changes rapidly from one speech sound to the next the prediction has to be *adaptive*; hence the name adaptive predictive coding (APC). We demonstrated the first APC system and its speech quality at the IEEE International Conference on Speech Communication held in Boston in 1967 [8], using two-level encoding of the prediction error. The predictive coder produced natural-sounding speech and speech quality was good [9], except for the presence of a low-level crackling noise that could be heard with careful listening over headphones.

10.5 Exploiting Masking for Noise Reduction in Speech Coders

Digital coders invariably introduce errors in the coded speech signal. Our next advance came in 1978 when we started to introduce *perceptual error* criteria. In other words, instead of minimizing a mathematical root-mean-square error, we minimized the *subjective loudness* of the quantizing noise as perceived by the human ear in the presence of the speech signal [10, 11]. This required measurements of loudness and masking was performed in collaboration with J. L. Hall [10],



BISHNU ATAL, right, a member of the Speech and Communications Research Department, Murray Hill, and Manfred Schroeder, head, Hearing and Speech Synthesis Research Department, Murray Hill, have won the Senior Award of the IEEE Acoustics, Speech, and Signal Processing Society for their paper entitled, "Predictive Coding of Speech Signals and Subjective Error Criteria." The paper appeared in the June 1979 issue of the Transactions in Acoustics, Speech and Signal Processing *and describes their recent work on encoding speech efficiently for digital transmission. The award was presented April 10 at the International Conference on Acoustics, Speech, and Signal Processing in Denver.*

Fig. 10.6 Manfred and I were working hard to reduce noise in speech coders to compress speech at low bit rates

to educate us as to the precise causes of audible distortion in a speech signal coded at low bit rates. Our new directions were recognized by IEEE with an award in 1980 (Fig. 10.6). The importance of adjusting the spectrum of the error so as to minimize the subjective distortion is now well established.

10.6 Developments in Speech Coders Leading to CELP

The optimum noise spectrum for minimum subjective distortion is in general non-flat. Early predictive coders designed for minimum subjective distortion used instantaneous quantizers; but instantaneous quantizers are suboptimal. We therefore decided to investigate *non*-instantaneous quantizing. Ideally in speech coding, one is interested in finding a sequence of binary digits which after decoding produces speech with minimal audible distortion. We therefore considered the use of block codes using tree search procedures to determine the optimal code. We first

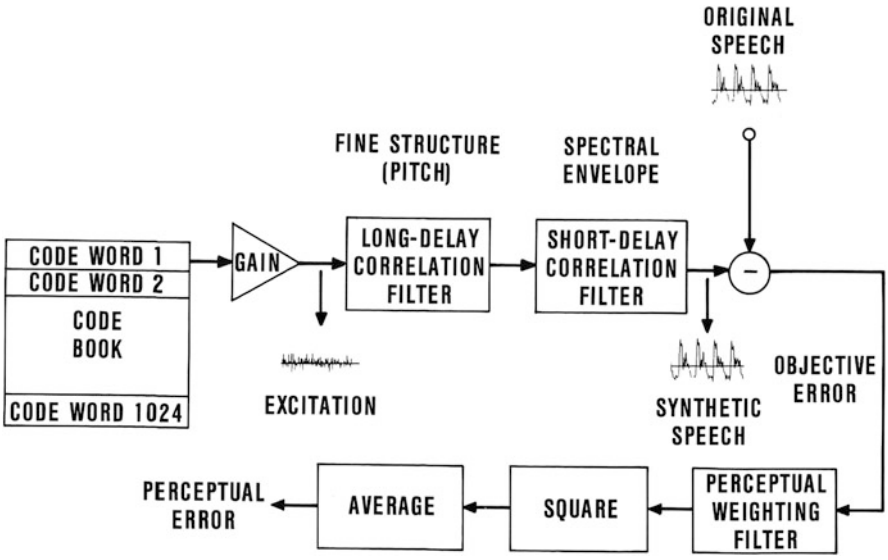
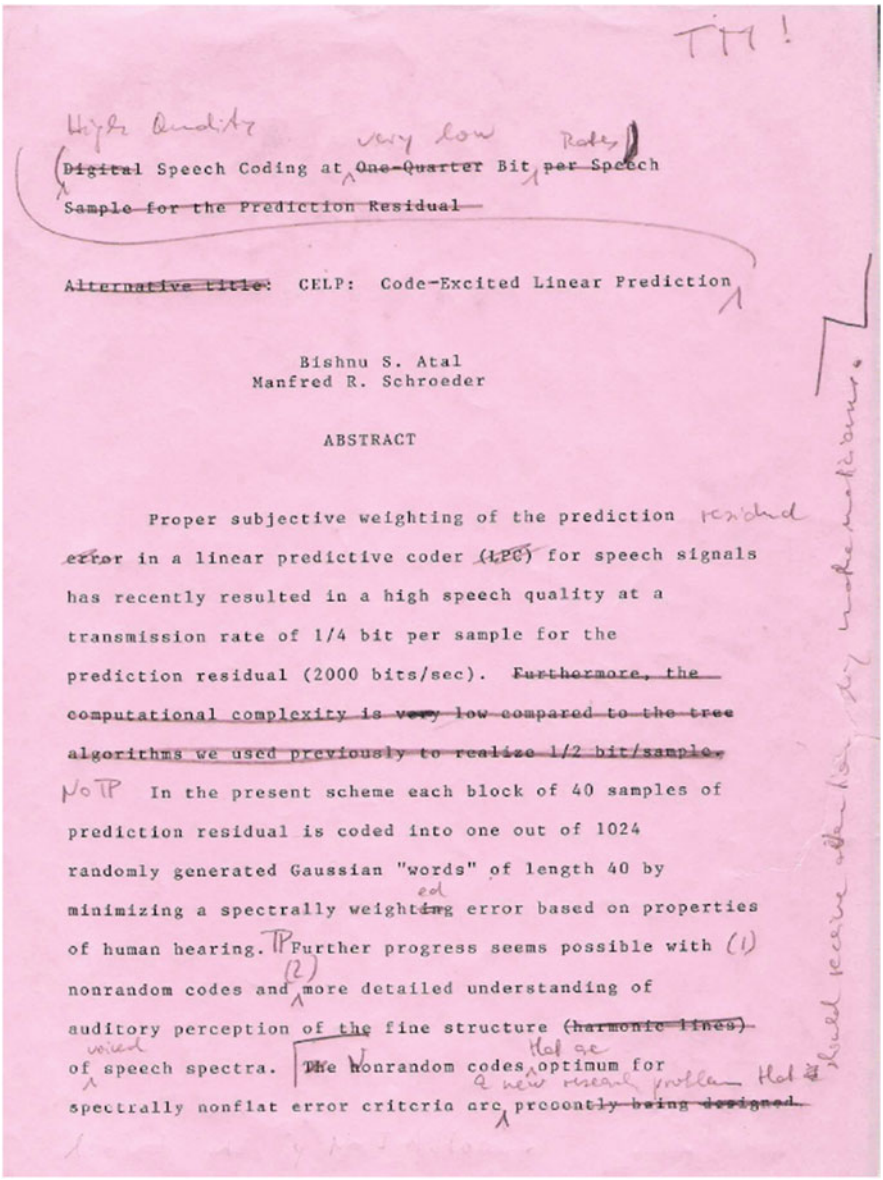


Fig. 10.7 Illustration of the search procedure in a code-excited linear prediction (CELP) coder

employed a binary tree search procedure for coding the excitation at 1 bit/sample which produced speech of high quality.

The use of block codes is necessary when only a fraction of a bit is available for coding each sample of the excitation. Very low bit rates for the excitation are necessary to bring the total bit rate for coding the speech signal down to 8 kb/s or even lower. In code-book coding, a set of possible sequences (may be random) is stored in a code book. For a given speech segment, the optimum sequence is selected by exhaustive search of the code book and an index specifying the optimum sequence is transmitted to the receiver. Code-book coding is often impractical due to the large size of code books. At very low bit rates, however, exhaustive search of the code book became practical, when the processing speed of digital processing chips was increasing three times every 2 years. The search procedure was still computationally very expensive; it took 125 s of Cray-1 CPU time to process 1 s of the speech signal. Here is a summary of various coding methods that we investigated before converging to code-excited linear prediction (CELP) (Fig. 10.7) [12].

[1978] Subjective error criteria
[1979] Exploiting masking properties of human ear
[1981] Tree coding based on rate distortion theory
[1982] Tree coding using block codes
[1982] Multipulse LPC
[1983] Stochastic coding of speech
[1984] Code-excited linear prediction



TM !

High Quality very low Rate

~~Digital~~ Speech Coding at ~~One-Quarter~~ Bit ~~per Speech Sample for the Prediction Residual~~

~~Alternative title:~~ CELP: Code-Excited Linear Prediction

Bishnu S. Atal
Manfred R. Schroeder

ABSTRACT

Proper subjective weighting of the prediction ~~error~~ residual in a linear predictive coder ~~(LPC)~~ for speech signals has recently resulted in a high speech quality at a transmission rate of 1/4 bit per sample for the prediction residual (2000 bits/sec). ~~Furthermore, the computational complexity is very low compared to the tree algorithms we used previously to realize 1/2 bit/sample.~~ NoTP In the present scheme each block of 40 samples of prediction residual is coded into one out of 1024 randomly generated Gaussian "words" of length 40 by minimizing a spectrally ~~weighting~~ ed error based on properties of human hearing. TP Further progress seems possible with (1) nonrandom codes and (2) more detailed understanding of auditory perception of the fine structure ~~(harmonic lines)~~ of voiced speech spectra. The nonrandom codes optimum for Hat ge a new research problem Hat spectrally nonflat error criteria are presently ~~being designed.~~

Fig. 10.8 Corrections from Manfred on the first manuscript of the code-excited linear prediction (CELP) coder

10.7 Conclusion

The entrance hall of the Bell Telephone Laboratories building at Murray Hill, New Jersey had an inscription attributed to Alexander Graham Bell that said "Leave the beaten track and dive into the woods. You will be certain to find something that you have never seen before." Working with Manfred was always looking for something new. Research was great fun. I presented a few examples in this chapter, but Manfred played around with many more exciting things. I hope that the other chapters will share some of this excitement (Fig. 10.8).

References

1. Schroeder, M. R., Atal, B. S., Bird, C.: Digital computers in room acoustics. In: Proc. Fourth Intern. Congr. Acoust, Copenhagen (Aug 1962)
2. Atal, B.S., Schroeder, M.R.: Study of sound decay using ray-tracing techniques. J. Acoust. Soc. Am. **41**, 1598 (1967)
3. Schroeder, M.R.: Digital simulation of sound transmission in reverberant spaces. J. Acoust. Soc. Am. **47**, 424 (1970)
4. Braner, A.: On the characteristics roots of non-negative matrices. In: Schnieder, H. (ed.) Recent Advances in Matrix Theory, pp. 3–38. University of Wisconsin Press, Madison (1964)
5. Shanon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(379–423), 623–656 (1948)
6. Wiener, N.: Extrapolation, Interpolation, and Smoothing of Stationary Time Series. The MIT Press, Cambridge, MA (1949)
7. Elias, P.: Predictive coding - I II. In: IRE Trans. Inform. Theory, IT-1, pp. 16–33 (Mar 1955)
8. Atal, B. S., Schroeder, M. R.: Predictive coding of speech signals. In: Proc. 1967 Conf. on Communication and Proc., pp. 360–361 (Nov 1967)
9. Atal, B.S., Schroeder, M.R.: Adaptive predictive coding of speech signals. Bell Syst. Tech. Jour. **49**, 1973–1986 (1970)
10. Schroeder, M.R., Atal, B.S., Hall, J.L.: Optimizing digital speech coders by exploiting masking properties of the human ear. J. Acoust. Soc. Am. **66**, 1647–1652 (Dec 1979)
11. Atal, B. S., Schroeder, M. R.: Predictive coding of speech signals and subjective error criteria. In: IEEE Trans. Acoust., Speech, Signal Proc., ASSP-27, pp. 247–254 (Jun 1979)
12. Schroeder, M. R., Atal, B. S.: Code-excited linear prediction (CELP): high-quality speech at very low bit rates.' In: Proc. Int. Conf. on Acoust., Speech and Signal Proc., vol. 10, paper no 25.1, pp. 937–940 (Apr 1985)

***

Biography



Bishnu S. Atal is an Affiliate Professor in the Electrical Engineering Department at the University of Washington, Seattle, WA. He retired in March 2002 after working for more than 40 years at Lucent Bell Labs, and AT&T Labs. He was made a Bell Laboratories Fellow in 1994 and an AT&T Fellow in 1997. He is internationally recognized for his many contributions to speech analysis, synthesis, and coding. His pioneering work in linear predictive coding of speech established linear prediction as one of the most important speech analysis technique leading to many applications in coding, recognition, and synthesis of speech. He was elected to the National Academy of Engineering in 1987, and to the National Academy of Sciences in 1993. He is a Fellow of the Acoustical Society of America and the IEEE. He is the recipient of the IEEE Centennial Medal in 1984 and the IEEE Morris N. Liebmann Memorial Field Award in 1986. He received the Thomas Edison Patent Award from the Research & Development Council of New Jersey in 1994 and the New Jersey Inventors Hall of Fame Inventor of the Year Award in 2000. He was awarded the Benjamin Franklin Medal in Electrical Engineering in 2003 and the IEEE Jack S. Kilby Signal Processing Medal in 2013.

Chapter 11
Transducer Research at Bell Laboratories Under Manfred Schroeder

Gerhard M. Sessler

Abstract In this chapter, the work on electroacoustic transducers and the corresponding materials research in Manfred Schroeder's Department at Bell Laboratories is reviewed, together with ensuing work at other laboratories. The emphasis is on studies of electret-type transducers and materials. This includes the description of the first polymer electret microphone, the first use of fluorocarbon electrets of high thermal stability soon thereafter, and the introduction of directional electret microphones. Other electret transducers designed during this period, such as microphone arrays, headphones, and touch actuators, are also discussed. Next, early materials research on polymer electrets at Bell Labs, utilizing isothermal and thermally stimulated methods as well as electron radiation experiments, is reviewed. Finally, the later studies in a number of other laboratories, based on the early work under Schroeder, are outlined. This includes, among others, research on digital electret headphones, two-dimensional arrays, silicon (MEMS) condenser microphones, and, extending into the present time, transducers based on piezoelectric cellular polymers.

11.1 Introduction

Bell Laboratories and its parent organization, the Bell Telephone Company, had a long and rich history of contributions to electroacoustic transducer research. Foremost, and right at the beginning of Bell Telephone, was the work by Alexander Graham Bell on transducers needed to operate a telephone. His first transducers were moving armature microphones and receivers, described in his 1876 patent on

G.M. Sessler (✉)
Department of Communications Technology, Technische Universität Darmstadt, Merckstrasse 25, 64283 Darmstadt, Germany
e-mail: g.sessler@nt.tu-darmstadt.de

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_11,
© Springer International Publishing Switzerland 2015

telephony [1]. Later achievements at Bell Telephone were the invention of the condenser microphone by E. C. Wente in 1917 [2], the use of the thermophone as a precision source of sound energy by Arnold and Crandall, also in 1917 [3], and in 1942 the development and commercialization of a condenser microphone which was the standard precision microphone for several decades, the legendary 640 AA [4] by Western Electric, the manufacturing branch of Bell.

Manfred Schroeder came to Bell Laboratories in 1954 and was appointed Head of the Acoustics Research Department in 1958 and Director of the Acoustics and Speech Research Laboratory in 1962, a position which he held until his departure from Bell Labs in 1969. One of the research topics in Manfred's area was the field of electroacoustic transducers, particularly transducers for the audio frequency range, suitable for use in telephony. The liberal scientific atmosphere at Bell Labs in those days made it possible to extend this work into the fields of infrasonic and ultrasonic transducers and to also include research work on materials of interest in transducer design. Work in these areas continued into the 1980s, long after Manfred's departure from Bell Labs.

In the chapter, the work on electroacoustic transducers and the corresponding materials research in Manfred Schroeder's Laboratory in the 1960s is reviewed. The emphasis is on the work on electret-type transducers and materials. Although this was not one of Manfred's own main activities, he did contribute many interesting ideas, particularly on directional transducers and he always had a vivid interest in the progress of this work and gave it the managerial support necessary for its successful execution.

11.2 Electret Microphones

The work on electret microphones in Schroeder's laboratory began around 1960. As opposed to much older Japanese work on electret microphones, in which the bias voltage in a condenser microphone was supplied by a thick wax electret with a dipolar polarization [5], the new studies were based on the use of thin polymer films carrying surface or space charges. Initial work showed that condenser microphones utilizing such "foil electrets" are of simple design and have high sensitivity, good frequency response, low distortion, and relatively high capacitance, with the latter being advantageous for the processing of the output signal [6]. The possible usefulness of these transducers in telephony was therefore soon recognized and the work on electret microphones intensified over the following years [7–9].

The first such microphone, built with a polyethyleneterephthalate electret (PETP, Trade name Mylar) film, is shown in Fig. 11.1. The essential parts of this transducer are the electret film, metalized on its outer surface, and a metal backplate with circular ridges which establish and control an air gap between electret and plate. If the membrane is deflected by an impinging sound wave, a voltage proportional to the sound pressure is generated between the metal layer of the film and the backplate. The air cavity behind the backplate is connected by holes to the

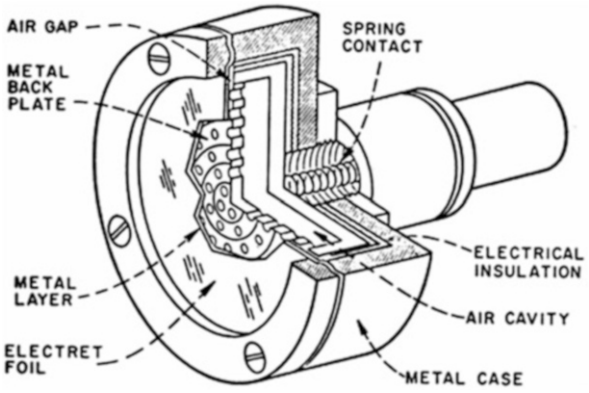


Fig. 11.1 First electret microphone with polymer film electret [6, 7]

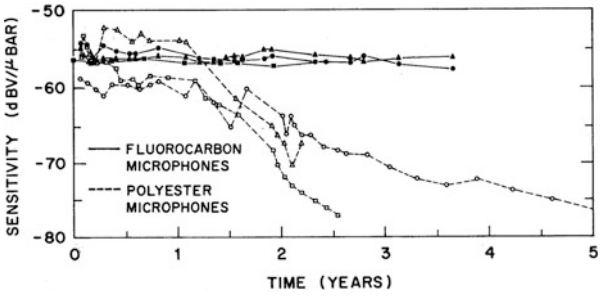


Fig. 11.2 Decay of sensitivity of early electret microphones [11]

air gap to soften the restoring force on the film and thus ensures larger vibration amplitudes and thus a larger sensitivity of the microphone.

It was soon recognized that the sensitivity of microphones with PETP electret under normal environmental conditions was constant for about a year but decreased thereafter. Typical results are shown in Fig. 11.2. The decrease was, of course, due to the decay of the electret charge. This made it necessary to look for electrets with improved charge stability. Experiments with a number of polymers eventually showed that electrets made of fluoroethylenepropylene (FEP, Trade name Teflon) had significantly longer time constants of the charge decay [10, 11]. It appeared that FEP and its homopolymer polytetrafluoroethylenepropylene (PTFE) were the best polymer electret materials available. Actually they were, and still are, the polymers of choice in almost all electret microphones.

The operation of electret microphones was analyzed by assuming a charge layer on the free surface of the electret film and induction charges on the film and back electrodes. The sensitivity of the device then follows from the density of the charge and the geometric dimensions of the system [12]. Sensitivities obtained from this analysis are, for the typical charge densities used on the electrets, of the order of 20 mV/Pa and thus in good agreement with experimental values. The calculations also showed that the electret charge corresponds to a bias voltage of about 100–200 V in conventional electrostatic microphones. Later numerical and experimental

work at Bell Labs was devoted to an analysis of the static deflection of an electret membrane by the electrostatic field of the charges and the resulting stability problem, and also to the dynamic deflections of the membrane, due to the sound pressure, for particular air gap geometries [13–15].

The development of prototypes of electret microphones, suitable for use in telephony, started in various laboratories around the world in the mid-1960s. At Bell Labs, the Development Department came up with several prototypes designed for use in telephony and denominated as EL-1 to EL-4. These transducers had initially external dimensions equal to those of the carbon microphones in use at that time, but were later on built with smaller dimensions. The microphones were equipped with a transistor preamplifier integrated directly into the microphone case. At Northern Electric in Canada a microphone prototype, also for use in telephony, was designed with a polycarbonate electret [16] and was later evaluated by field tests. In 1968 Sony in Japan brought the first commercial electret microphone developed for use in HiFi on the market. Soon other companies followed, production of electret microphones increased at a rapid pace and has now reached about three billion annually.

11.3 Directional Electret Microphones

It was recognized early that electret microphones, due to their simplicity in mechanical design, would be very suitable sensors for directional transducers. Particularly microphones based on the gradient principle could be realized easily with electret microphones since these are small and permit ready access to the rear cavity behind the membrane.

In a discussion about the application of the gradient principle to electret microphones, Manfred Schroeder suggested to build second-order microphones with toroidal directional characteristics by utilizing the gradient combination shown in Fig. 11.3. Toroidal microphones were of particular interest in conference telephony, where participants seated around a table have to be covered with equal sensitivity by a directional microphone. To implement such a microphone, the sound field has to be sampled in eight points as indicated in Fig. 11.3, left [17]. This sampling may be performed by the opening of tubes which guide the sound waves into two cylindrical cavities, as shown in the center part of the figure. Using two cavities, positioned on either side of the membrane of an electret microphone and each collecting the sound of four tubes, one obtains the opposite sensitivities designated by “plus” and “minus” in the left part of the figure. The tubes together with the cavities form a Helmholtz resonator while the tubes by themselves have a length resonance. The Helmholtz and length resonances are placed at the lower and upper corner frequencies of the telephone band, respectively. With such a design, a flat frequency response is obtained within this frequency band since the ω^2-dependence of the second-order gradient is compensated by the $1/\omega^2$-dependence of the Helmholtz resonator above its resonance.

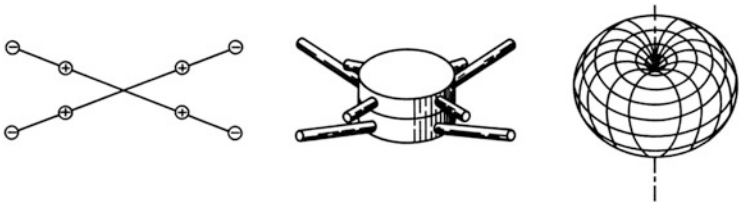


Fig. 11.3 Basic design and directional characteristics of the second-order toroidal microphone [17]. *Left*: schematic. *Center*: implementation. *Right*: directivity pattern

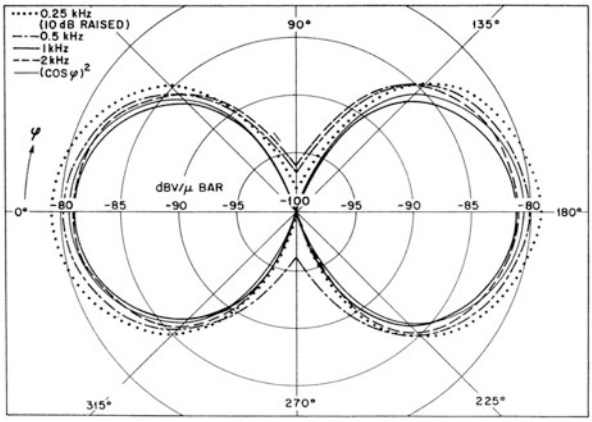


Fig. 11.4 Polar pattern of a toroidal microphone designed according to Fig. 11.3 [17]

The directivity pattern of a toroidal microphone of this kind is illustrated in the right-hand part of Fig. 11.3. The calculated and measured polar patterns are depicted in Fig. 11.4 [17]. Expected are cosine-square dependencies which are independent of frequency. The figure shows the relatively good agreement between the calculated directivity and the measured patterns for three frequencies. The directivity index of these microphones is 3 dB.

Other directional gradient-type microphones realized in Schroeder's laboratory were unidirectional and close-talking transducers. The unidirectional microphones were again second-order transducers. They were similarly designed with two tube-cavity systems, each containing two tubes, and thus had also constant sensitivity in the telephone band. The calculated and measured polar patterns are depicted in Fig. 11.5 [18, 19] and compare favorably in the direction of the main lobe but show some deviations in the other directions where the sensitivity is lower. The directivity index of such systems is about 9 dB. These unidirectional transducers, just as the toroidal transducers, are relatively simple systems. Nowadays they can be easily realized with spherical microphone arrays, as shown in Fig. 11.6, which have proper signal processing capabilities [20].

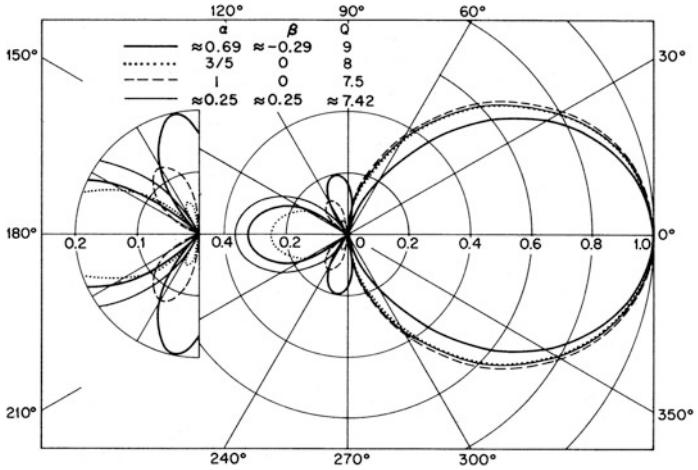


Fig. 11.5 Polar pattern of unidirectional microphone [18, 19]



Fig. 11.6 "Eigenmike" with adjustable directional characteristics (Meyer and Elko [20])

Close-talking microphones were designed with the first-order gradient arrangement in which the membrane of an electret microphone is accessible to the sound waves from both sides [21]. These microphones have, without further compensation, a flat frequency response for a close sound source but a response decreasing toward low frequencies for far sound sources. Thus, low-frequency noise generated at a large distance from the microphone is suppressed by up to 30 dB compared to speech or other sounds originating nearby.

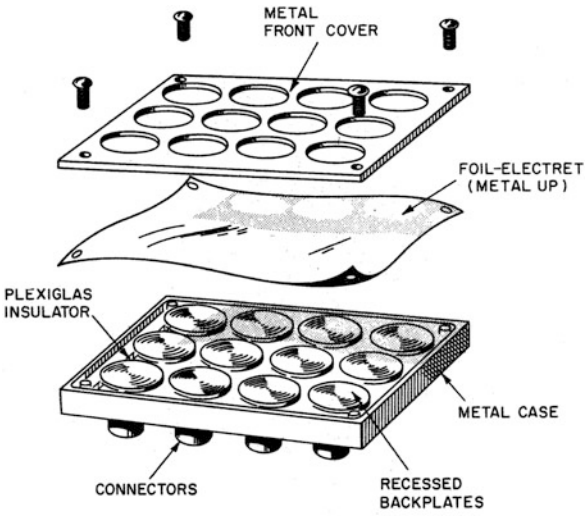


Fig. 11.7 Exploded view of foil-electret touch dial [23]

11.4 Other Electret Transducers

Apart from electret microphones, other electret transducers were realized in the 1960s and early 1970s in the Acoustics Research Laboratory in Murray Hill. Among these were headphones, touch-sensitive transducers, ultrasonic transducers, and holographic arrays.

Electret headphones were actually introduced in 1961 [22], prior to the electret microphones. The first transducers of this kind comprised a 6 μm thick PETP membrane which carried an electrostatic charge and therefore did not require an external DC-bias. Because of their high resonance frequency, such headphones had an excellent impulse response, a flat frequency response in the audio range and also low distortion. Electret headphones were brought on the market shortly after electret microphones and have since been in commercial use.

Touch-sensitive electret transducers were of interest in the 1960s for many applications, in particular as telephone dials. An exploded view of such a dial is shown in Fig. 11.7 [23]. It consists of a metalized foil electret and 12 backplates which are recessed in a dielectric plate. By touching the foil through one of the holes of the front cover, the film is deflected and an electric signal is generated. Such dials were attractive in those days because of their simplicity and reliability due to the fact that no metallic contacts are required.

Ultrasonic two-dimensional arrays allow one to sample sound fields in a large number of points, as required in acoustic holography. A possible approach to such a sampling device is a subdivided, large-area electret microphone [24, 25]. The design of a transducer of this kind is shown in Fig. 11.8, left. It consists of a film electret with n metal strips and a backplate with an equal number of metal strips. If film and backplate are arranged such that the strip patterns are perpendicular to each other, as shown in Fig. 11.8, right, a square array with n^2 microphone elements is

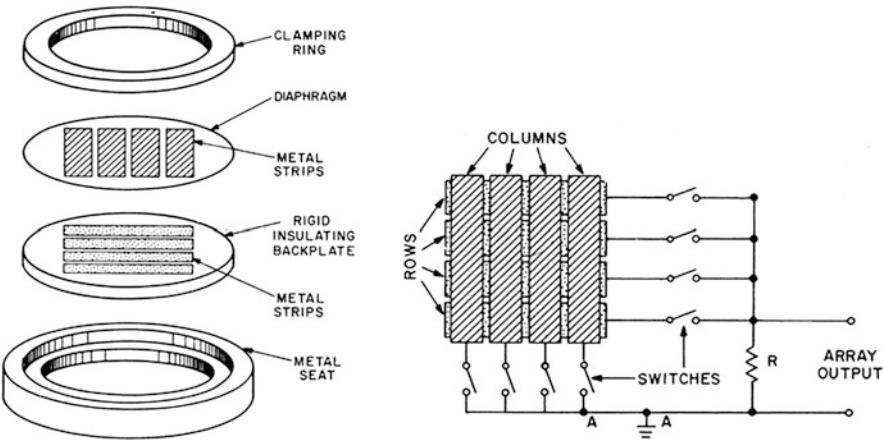


Fig. 11.8 Two-dimensional electret microphone array for acoustic holography [24, 25]. *Left part*: exploded view of the array. *Right part*: electrical connections

obtained. With switches connected to each film and backplate strip, the output of any particular microphone element is obtained by closing the corresponding column and row switches. The advantage of this array is that n^2 microphones are realized with only $2n$ components and that the gating requires only $2n$ gates. This is particularly important if the number of elements is large.

11.5 Research on Electret Materials for Use in Electroacoustic Transducers

Research on electret materials was carried out in the Acoustics Research Department in support of the transducer work. The electret research also started in the early 1960s and was initially directed toward a search for polymers with superior charge-storage capabilities. The electret material used initially was the above-mentioned PETP which was widely employed in condenser transducers with solid dielectric, the so-called Sell-transducers. It had been noticed before that PETP assumes a quasi-permanent charge if exposed to an electric field [26]. Systematic experiments revealed, however, that charge in this material is not sufficiently permanent for use in electret microphones (see Sect. 11.2). After some search and extensive testing, FEP was identified as an extremely stable electret material and then used in electret microphones [10, 11].

Among the experimental methods for the investigation of charge retention in polymer films were, for example, the study of isothermal charge decay at room temperature and at elevated temperatures, the measurement of thermally stimulated currents at linearly increasing temperatures, the analysis of charge deposition

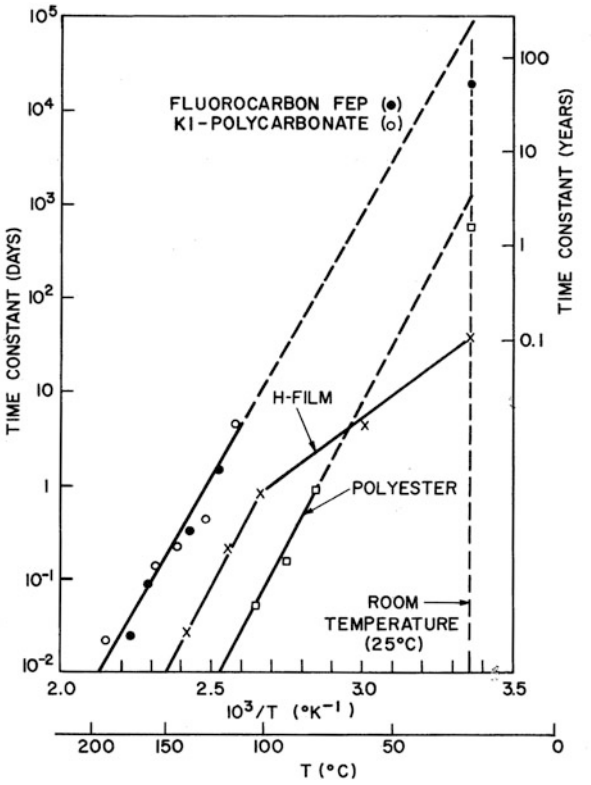


Fig. 11.9 Time constants of surface-charge decay of various 25 μm polymer films as function of inverse temperature (Arrhenius plots) [27]

by electron beams and of the accompanying radiation-induced effects, and the evaluation and extrapolation of time constants of the charge decay obtained at elevated temperatures. The latter method was particularly useful in view of the long time constants of the charge decay at room temperature expected for some of the polymers.

Results for the time constants τ of the charge decay for four different polymers at elevated temperatures and at room temperature are shown in Fig. 11.9 [27]. The log τ vs. $1/T$- plots usually yield straight lines (the so-called Arrhenius plots) which indicate that the activation energy in the temperature range considered is constant. The data show that the time constants evaluated directly at room temperature are in order-of-magnitude agreement but somewhat smaller than the constants obtained by the straight-line extrapolations of the high-temperature data. The deviations are not surprising since the isothermal decay is always non-exponential in the sense that it slows down with time, such that longer time constants than those obtained by extrapolation of relatively short-term RT-data have to be expected. The longest time constants are those for FEP and polycarbonate. It was found, however, that FEP is much better in humid atmospheres, which made it the material of choice for electret microphones (see Sect. 11.2). The polyester (PETP) electrets exhibit the

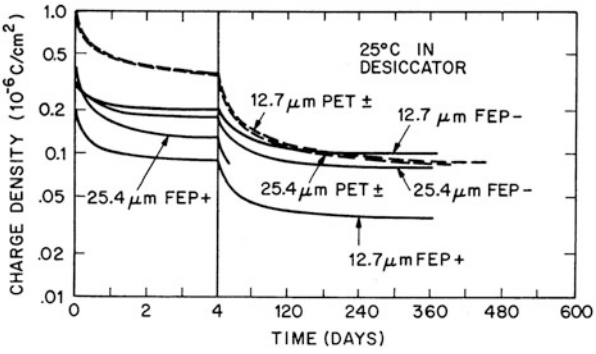


Fig. 11.10 Charge decay for high charge densities on various polymer films [29]

relatively short lifetimes that had already been found in the experimental results for the first electret microphones, as shown in Fig. 11.2. As opposed to these results, the polyimide (H-film) data have to be approximated by two straight lines, indicating a change in activation energy of the charge-activation process at about 100 °C.

In the late 1960s, first experiments with electron-beam charging were carried out [28]. The beam energy was chosen such that the electrons did not penetrate the entire thickness of the film but were trapped within the film volume. It was recognized that this method allows one to obtain stable electrets and that the charges deposited in the films had a very uniform lateral density distribution. These experiments paved the way for the later cooperation with Professor Bernhard Gross, who visited the Acoustics Research Department regularly in the 1970s and with whom many experiments on charge storage and charge dynamics in electrets irradiated with low-energy electrons were performed (see Sect. 11.6).

Around 1970, interest focused also on the potential of electrets for other applications. In this context, the maximum charge densities that can be achieved in various polymers were investigated. The limits for stored charge are determined by the dielectric strength of the materials and are as high as 10^{-2} C/m^2 for PETP and 0.5×10^{-2} C/m^2 for FEP, as shown in Fig. 11.10 [29]. Charge decays were always found to be non-exponential in the sense that time constant increase with time, as the charge density gets lower. For example, for charge densities below about 10^{-3} C/m^2, the charge decays slow down considerably and time constants reach about 10 years for FEP. This indicates the presence of traps of different energetic depth with the shallower traps discharging more quickly than the deeper traps. As Fig. 11.10 also shows, the superior charge retention properties of FEP are only observed for negative charging. Positive charge dissipates much more quickly. Later on it was found that the stability of positive charges in FEP could be improved significantly by corona-charging at elevated temperatures [30].

11.6 A Few Results of Later Work

The work on electroacoustic transducers and on electro-active materials, based on the results obtained in the 1960s, continued after Manfred's departure. The studies were carried out at Bell Labs and later on also at a number of other places where new research laboratories devoted to this kind of work were set up by people departing from the Acoustics Research Department in Murray Hill. Just a few examples, mostly related to the authors' own activities, will be discussed in the following.

Transducer work at Bell Labs in the 1970s was still concerned with directional microphones, particularly second-order gradients [19], but also with other electret devices. One such gadget was an earphone that converts digital PCM signals directly into an analog acoustic output signal by using a backplate subdivided such that areas scaled by powers of 2 are obtained [31]. By covering these areas with an electret film, one obtains an array of individual sound radiators whose backplate segments are addressed with the proper digital bit. Thus, an acoustic output corresponding to the digital input signal is obtained.—A few years later, in the early 1980s, transducer work intensified again. Topics were once more electret earphones and directional microphones, but major activities were also in signal processing for microphone arrays and in silicon or MEMS microphones.

The work on electret earphones was directed toward a better understanding of edge-supported systems. While a theoretical model described the static and dynamic deflections of the membrane, predicted the nonlinear response, and allowed one to maximize the earphone sensitivity, experimental results confirmed the predictions and offered additional insight into the importance of transducer parameters, such as ridge height and air loading [32]. Later on, these studies were extended to earphones with radially varying parameters, such as charge density or air gap thickness [33].

A directional microphone in the form of an acoustic antenna was realized with a long strip of electret foil placed over a metalized backplate [34]. By locally adjusting parameters such as the charge density on the electret along the length of the strip, a shading of the sensitivity may be achieved which results in a desired directivity pattern. Compared to a linear microphone array with the same directionality, the antenna offers some advantages such as easier construction and higher signal-to-noise ratio.

Two-dimensional arrays consisting of hundreds of electret microphones for capturing speech signals of persons seated in large auditoria were also designed in the 1980s. Signal processing was needed to discriminate speech from interfering noise and to automatically steer the arrays in the speaker direction by a microprocessor system [35]. Similar arrays, used for teleconferencing, included hands-free voice control [36]. Another design feature of such arrays was the control of the pattern of the beam, including a constant beam width, within a broad frequency range [37].

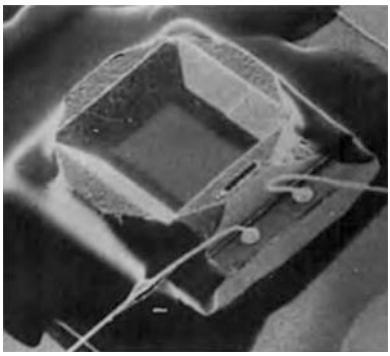


Fig. 11.11 First silicon condenser (MEMS) microphone [41, 42]

The work on silicon microphones was initiated independently at Bell Labs and at the Darmstadt University of Technology, where the author established a new laboratory in 1975. In both places, silicon *condenser* microphones were studied, although in the Darmstadt laboratory later on also other transduction principles, such as piezoelectric, FET-modulating, and opto-acoustic systems [38], were investigated. Several design concepts for the condenser microphones were published or patented between 1983 and 1985 [see [39, 40] and references therein]. They were mostly based on sacrificial-layer etching technology to obtain a single-chip transducer with the required air gap. As far as biasing is concerned, either an external voltage in the 10–28 V range or electret biasing by charging a silicon dioxide layer was suggested. Implementations of externally biased microphones, as shown in Fig. 11.11 and consisting of a backplate chip and a membrane chip, came in 1986 [41, 42]. These were actually the first operational silicon condenser microphones worldwide. Improved transducers, mostly consisting of a single chip, followed. These were designed in several other laboratories, including Darmstadt, in the 1990s [38] but it took until 2002 until commercial MEMS microphones were built.

Electret work in the 1970s shifted to a powerful diagnostic tool already developed before, namely electron-beam irradiation (see above) [28]. Starting in 1972, Bernhard Gross was invited annually to Bell Labs to cooperate on experiments with electron beams. He introduced the method of the split-Faraday cup to investigate charge deposition, charge distribution, and radiation-induced conductivity (RIC) in polymers [43]. From such measurements conclusions were drawn about charge dynamics, i.e., transport, recombination, and trapping of deposited charges in polymers used as electret materials. At Bell Labs, this work was supported by computer simulations of charge dynamics in electron-irradiated polymer films [44]. The calculations were performed by solving the charge-transport equations for open- and short-circuit conditions. Other electret work in the 1980s at Bell Labs concerned the experimental discrimination between surface and bulk traps in polymers [45] and the thermal stabilization of positive charges in FEP [30]. The collaboration with Bernhard Gross was also extended to the Darmstadt laboratory and continued into the 1990s [46].

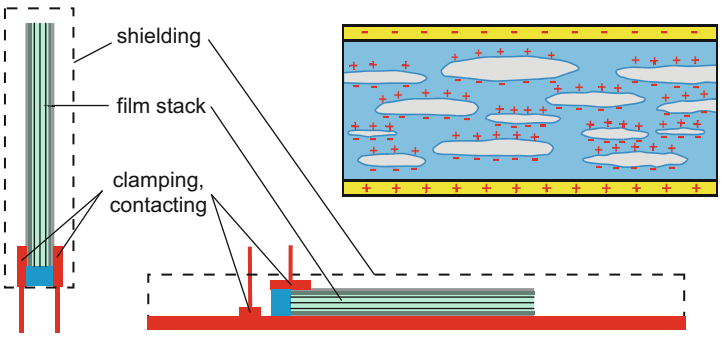


Fig. 11.12 Schematic cross section of piezoelectret (*right top*) and design of piezoelectret microphone (*left bottom*) [52, 53]

In the late 1990s, a new type of piezoelectric material became available [47, 48]. This material consists of a cellular polymer, e.g., cellular polypropylene, which is charged such that all the voids in the material carry positive charges on their upper surface and negative charges on their lower surface, see Fig. 11.12 [49]. Such films show a strong longitudinal piezoelectric effect with d_{33}-coefficients above 1,000 pC/N at 0.001 Hz and 500 pC/N in the audio frequency range [50]. These coefficients are more than an order of magnitude larger than those of poly(vinylidene fluoride), the best piezoelectric polymer known before. Further improvements up to 1,500 pC/N in the audio frequency range were obtained by applying an additional DC-bias to the films [51].

This large piezoelectric activity suggested the use of such films as microphones: A sound wave impinging on a film of this type will periodically modulate the film thickness and thus cause an electrical output signal proportional to the sound pressure as long as the film dimensions are small compared to the wavelength. The only additional components in such a microphone are suitable contacts, output leads, and electric shielding, as illustrated in the lower part of Fig. 11.12. First microphones of this type showed relatively flat frequency responses extending from the infrasonic into the ultrasonic region and sensitivities of about 0.5 mV/Pa [52]. Later transducers using stacked film designs and improved cellular electrets exhibited sensitivities of up to 15 mV/Pa [53].

The transduction mechanism of these microphones can, of course, be described either as a piezoelectric effect or as electret-biasing of a capacitive microphone. The former kind of view has been largely adopted nowadays since the electro-activity of the piezoelectrets is phenomenologically identical to that of a classical piezoelectric medium, including the hysteresis effects exhibited by the old and the new materials [54]. However, looking at the intrinsic causes of the piezoelectricity, the charges trapped at the walls of the voids in the material are responsible for the observed piezoelectric activity and thus the transducers are also of the electret type. Therefore, the term “piezoelectret” has been coined for these materials and the corresponding transducers.

This new transducer work and the supporting materials research are thus, in a sense, a continuation of the electret studies started under Manfred Schroeder in the 1960s: Polymers charged in a more intricate way, the piezoelectrets, are used in various ways in electroacoustic and in electromechanical transducers. These transducers are simpler than the older electret devices in the sense that the air gap is directly integrated into the film. Thus, while the electret replaces the external bias by a built-in charge, the piezoelectret additionally replaces the air gap by built-in air cavities. With the recent activities in the field of piezoelectret devices, we are witnessing another step in the evolution of electroacoustic transducers.

References

1. Bell, A. G.: U. S. Patent 174465, Improvement in Telegraphy (1876)
2. Wente, E.C.: A condenser transmitter as a uniformly sensitive instrument for the absolute measurement of sound intensity. Phys. Rev. Ser. II **10**, 39–63 (1917)
3. Arnold, H.D., Crandall, I.B.: The thermophone as a precision source of sound. Phys. Rev. **10**, 22–38 (1917)
4. Western Electric Improved 640 AA condenser microphone and RA-1095 Amplifier. Copyright: Western Electric Company, Incorporated (1946)
5. Nishikawa, S., Nukiyama, D.: A use of the electret in a condenser transmitter. Proc. Imp. Acad. Tokyo **4**, 290–291 (1928)
6. Sessler, G.M., West, J.E.: Self-biased condenser microphone with high capacitance. J. Acoust. Soc. Am. **34**, 1787–1788 (1962)
7. Sessler, G.M., West, J.E.: Electret transducers: a review. J. Acoust. Soc. Am. **53**, 1589–1600 (1973)
8. Elko, G.W., Harney, K.P.: A history of consumer microphones: the electret condenser microphone meets micro-electro-mechanical-systems. Acoustics Today **5**(2), 4–13 (2009)
9. Flanagan, J.L.: A singular advance in conversion of acoustic signals to electrical form: the electret microphone. IEEE Signal Process. Mag. **27**(5), 102–116 (2010)
10. Sessler, G.M., West, J.E.: Electrostatic microphones with foil electret. J. Acoust. Soc. Am. **37**, 1209–1210 (1965)
11. Sessler, G.M., West, J.E.: Foil-electret microphones. J. Acoust. Soc. Am. **40**, 1433–1440 (1966)
12. Sessler, G.M.: Electrostatic microphones with electret foil. J. Acoust. Soc. Am. **35**, 1354–1357 (1963)
13. Warren, J.E., Brzezinski, A.M., Hamilton, J.F.: Capacitance-microphone static membrane deflections. J. Acoust. Soc. Am. **52**, 711–719 (1972)
14. Warren, J.E.: Capacitance microphone static membrane deflections: comments and further results. J. Acoust. Soc. Am. **58**, 733–740 (1975)
15. Warren, J.E., Brzezinski, A.M., Hamilton, J.F.: Capacitance microphone dynamic membrane deflections. J. Acoust. Soc. Am. **54**, 1201–1213 (1973)
16. Reedyk, C.W.: An electret transmitter for the telephone set. Electrochem. Technol. **6**, 6–10 (1968)
17. Sessler, G.M., West, J.E., Schroeder, M.R.: Toroidal microphones. J. Acoust. Soc. Am. **46**, 28–36 (1969)
18. Sessler, G.M., West, J.E.: Directional transducers. IEEE Trans. Audio Electroacoust. **AU-19**, 19–23 (1971)
19. Sessler, G.M., West, J.E.: Second-order gradient unidirectional microphones utilizing electret transducers. J. Acoust. Soc. Am. **58**, 273–278 (1975)

20. Meyer, J., Elko, G. W.: Realization of a spherical microphone array. In: Proc. 19th Internat. Congr. Acoust., Paper ELE-02-007, Madrid (2007)
21. Sessler, G.M., West, J.E.: First-order gradient microphone based on the foil-electret principle: discrimination against air-borne and solid-borne noises. J. Acoust. Soc. Am. **46**, 1081–1086 (1969)
22. Sessler, G.M., West, J.E.: Condenser earphone with solid dielectric, Audio Engineering Society Preprint No. 187, (Audio Engineering Society, 1961). J. Audio Eng. Soc. **10**, 212–215 (1962)
23. Sessler, G.M., West, J.E., Wallace Jr., R.L.: New touch actuator based on the foil-electret principle. IEEE Trans. Commun. **COM-21**, 61–65 (1973)
24. Nigam, A.K., Taylor, K.J., Sessler, G.M.: Foil-electret transducer arrays for real-time acoustical holography. In: Wade, G. (ed.) Acoustical Holography, vol. 4, pp. 173–194. Plenum Publishing Corp, New York (1972)
25. Nigam, A.K., Sessler, G.M.: Condenser–transducer array for acoustical holography. Appl. Phys. Lett. **21**, 229–231 (1972)
26. Kuhl, W., Schodder, G.R., Schröder, F.-K.: Condenser transmitters and microphones with solid dielectric for airborne ultrasonics. Acustica **4**, 519–532 (1954)
27. Sessler, G.M., West, J.E.: Foil electrets and their use in condenser microphones. J. Electrochem. Soc. **115**, 836–841 (1968)
28. Sessler, G.M., West, J.E.: Charging of polymer foils with monoenergetic low-energy electron beams. Appl. Phys. Lett. **17**, 507–509 (1970)
29. Sessler, G.M., West, J.E.: Production of high quasipermanent charge densities on polymer foils by application of breakdown fields. J. Appl. Phys. **43**, 922–926 (1972)
30. von Seggern, H., West, J.E.: Stabilization of positive charge in fluorinated ethylene propylene copolymer. J. Appl. Phys. **55**, 2754–2757 (1984)
31. Flanagan, J.L.: Direct digital-to-analog conversion of acoustic signals. Bell Syst. Tech. J. **59**, 1693–1719 (1979)
32. Busch-Vishniac, I.J.: Response of an edge-supported circular membrane electret earphone. Parts I and II. J. Acoust. Soc. Am. **75**, 977–989 (1984), 990–995
33. Busch-Vishniac, I.J.: Response of an edge-supported, circular, foil-electret earphone with radially varying parameters. J. Acoust. Soc. Am. **83**, 796–804 (1988)
34. Busch-Vishniac, I.J., West, J.E., Wallace Jr., R.L.: A new approach to transducer design applied to a foil electret acoustic antenna. J. Acoust. Soc. Am. **76**, 1609–1616 (1984)
35. Flanagan, J.L., Johnston, J.D., Zahn, R., Elko, G.W.: Computer-steered microphone arrays for sound transduction in large rooms. J. Acoust. Soc. Am. **78**, 1508–1518 (1985)
36. Flanagan, J.L., Berkley, D.A., Elko, G.W., West, J.E., Sondhi, M.M.: Autodirective microphone systems. Acustica **73**, 58–71 (1991)
37. Elko, G.W.: Microphone array systems for hands-free telecommunication. Speech Commun. **20**, 229–240 (1996)
38. Sessler, G.M.: Silicon microphones. J. Audio Eng. Soc. **44**, 16–22 (1996)
39. Hohm, D., Sessler, G. M.: An integrated silicon-electret-condenser microphone. In: Proc. of the 11th Int. Congress on Acoustics, vol. 6, pp. 29–32. Paris (1983)
40. Poteat, T. L., Bush-Vishniac, I. J.: Silicon micromechanics: new technology for sensors. In: Proc. of the 12th Int. Congress on Acoustics, vol. 3, paper L 1–2. Toronto (1986)
41. Hohm, D.: Silicon sensor for airborne sound. In: Proc. of the 12th Int. Congress on Acoustics, vol. 3, paper L 3–3. Toronto (1986)
42. Hohm, D., Hess, G.: A subminiature condenser microphone with silicon nitride membrane and silicon backplate. J. Acoust. Soc. Am. **85**, 476–480 (1989)
43. Gross, B., Sessler, G.M., West, J.E.: Charge dynamics for electron-irradiated polymer-foil electrets. J. Appl. Phys. **45**, 2841–2851 (1974)
44. Berkley, D.A.: Computer simulation of charge dynamics in electron-irradiated polymer foils. J. Appl. Phys. **50**, 3447–3453 (1979)
45. von Seggern, H.: Detection of surface and bulk traps. J. Appl. Phys. **52**, 4086–4089 (1981)
46. Sessler, G.M.: Bernhard Gross and electret research: his contributions, our collaboration, and what followed. IEEE Trans. Dielectr. Electr. Insul. **13**, 942–952 (2006)

47. Lekkala, J., Poramo, R., Nyholm, K., Kaikkonen, T.: EMF force sensor—a flexible and sensitive electret film for physiological applications. Med. Biol. Eng. Comput. **34**, 67–69 (1996)
48. Sessler, G.M., Hillenbrand, J.: Electromechanical response of cellular electret films. Appl. Phys. Lett. **75**, 3405–3407 (1999)
49. Hillenbrand, J., Sessler, G.M.: High-sensitivity piezoelectric microphones based on stacked cellular polymer films. J. Acoust. Soc. Am. **116**, 3267–3270 (2004)
50. Zhang, X., Hillenbrand, J., Sessler, G.M.: Improvement of piezoelectric activity of cellular polymers using a double-expansion process. J. Phys. D. Appl. Phys. **37**, 2146–2150 (2004)
51. Hillenbrand, J., Sessler, G.M.: DC-biased ferroelectrets with large piezoelectric d_{33}-coefficients. J. Appl. Phys. **103**, 074103-1–074103-7 (2008)
52. Kressmann, R.: New piezoelectric polymer for air-borne and water-borne sound transducers. J. Acoust. Soc. Am. **109**, 1412–1416 (2001)
53. Hillenbrand, J., Sessler, G.M.: Stacked piezoelectret microphones of simple design and high sensitivity. IEEE Trans. Dielectr. Electr. Insul. **13**, 973–978 (2006)
54. Bauer, S., Gerhard-Multhaupt, R., Sessler, G.M.: Ferroelectrets: soft electroactive foams for transducers. Phys. Today **57**(2), 37–43 (2004)

***

Biography



Gerhard M. Sessler studied physics at the University of Goettingen in Germany and joined Bell Laboratories in 1959 as a member of the Acoustics Research Department in Murray Hill, N.J. He had responsibilities for the work on electroacoustic transducers, concert hall acoustics, and charge storage phenomena in solids. Together with James E. West he invented the first polymer electret microphone which was commercialized later on and still is the predominant microphone type. In 1975, he was appointed a professor of electroacoustics at the University of Technology in Darmstadt, Germany. He has since been involved in various fields of acoustics and solid-state physics, in particular in acoustic silicon transducers and acoustic signal processing. In 1983, he and his co-worker Dietmar Hohm designed the first MEMS condenser microphone based on silicon micromachining. Presently, he is an Emeritus Professor and he and his group are working on new piezoelectric polymers and their applications.

Chapter 12
Manfred Schroeder and Acoustical Impedance

Roland Kruse and Volker Mellert

Abstract Manfred Schroeder was a man with many interests. Between the years 1966 and 1967, he invented devices for measuring the surface impedance of materials as well as the vocal tract impedance. At the time, he would only consider normal incidence; however, he already anticipated that the behavior of materials at grazing incidence would have relevance to room acoustics, a field of research he is well known for.

To achieve this aim, an experimental set-up to determine the characteristic impedance of absorbers at grazing incidence has been developed. The sample is placed on the bottom of a rectangular impedance tube and the horizontal wave number above the sample is calculated from the pressure transfer function between two microphones. From this wave number, the absorber properties can be deduced. While the method works reasonably well for highly absorbing samples, the non-ideal sound field in the tube—as confirmed by finite element simulations—limits its usefulness in case of less absorbent materials. Improvements of the method are suggested.

12.1 Introduction

Manfred Schroeder was a man with many interests. Between the years 1966 and 1967, he invented devices for measuring the surface impedance of materials as well as the vocal tract impedance. At the time, he would only consider normal incidence; however, he already anticipated that the behavior of materials at grazing incidence would have relevance to room acoustics, a field of research he is well known for.

R. Kruse (✉)
Acoustics Group, Institute of Physics, Oldenburg University, 26111 Oldenburg, Germany
e-mail: r.kruse@tu-bs.de

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_12,
© Springer International Publishing Switzerland 2015

To achieve this aim, an experimental set-up to determine the characteristic impedance of absorbers at grazing incidence has been developed. The sample is placed on the bottom of a rectangular impedance tube and the horizontal wave number above the sample is calculated from the pressure transfer function between two microphones. From this wave number, the absorber properties can be deduced. While the method works reasonably well for highly absorbing samples, the non-ideal sound field in the tube—as confirmed by finite element simulations—limits its usefulness in case of less absorbent materials. Improvements of the method are suggested.

12.2 System for Determining Acoustic Reflection Coefficients

The first sign of Manfred Schroeder's research on measuring impedances is his patent number 3,346,067 [10], granted in October 1967 but filed already in March 1966, which deals with a "system for determining acoustic reflection coefficients" which equals, as we have plane wave propagation, a device for impedance determination. Before having a detailed look at this invention, let us have a look at how reflection/absorption coefficients of materials (at normal incidence) were measured at that time: using the Kundt's tube. This meant that for each frequency of interest, one had to find—by moving the microphone along the tube—the minimum and maximum of the sound pressure. In addition, if one was interested in the complex reflection coefficient, one had to locate the position of the first minimum with respect to the sample's surface. Obviously, that was a time-consuming task, and for materials with high absorption, like porous absorbers at high frequencies, it was difficult to locate the extrema of the sound pressure.

Manfred Schroeder sought to overcome this problem by using broadband signals and two microphones, relying on the definition of the reflection coefficient: "The reflection coefficient, an important parameter in determining the acoustic characteristics of a material, is defined as the ratio of the amplitude of a reflected pressure wave to the amplitude of an incident pressure wave, both amplitudes being measured at the face of the material" [10]. In other words, we try to separate incident and reflected wave and to predict the sound pressure at the sample's surface from measurements done at some distance to it.

Figure 12.1 shows a sketch of the system. It shows what would nowadays be called a two-microphone impedance tube, the only difference being that the sound source is not located opposite the sample but at the side of the tube.

From the measured pressure amplitudes at the two locations, first the amplitudes of the incident and reflected wave were derived with Eq. (12.1), then the (complex) reflection coefficient: Eq. (12.2). Analog electronics (delays, multipliers, etc.) were used to perform these operations, as this was the established solution for such a

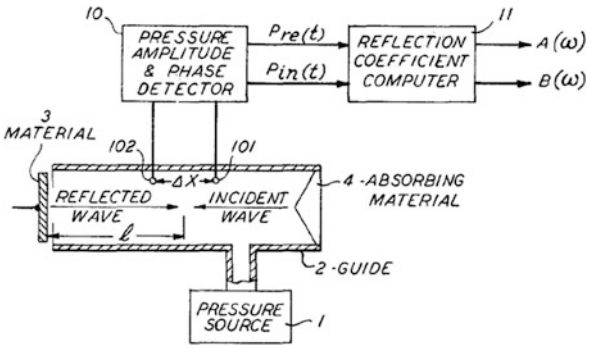


Fig. 12.1 Sketch of the impedance measurement system invented by M.Schroeder in 1966 [10]

problem at that time (the term "reflection coefficient computer" is rather misleading from today's point of view on digital signal processing). Broadband processing was achieved by using a bandpass filter bank and having multiple "computers" evaluating Eq. (12.2) simultaneously. Hence, one could obtain the frequency-dependent, complex reflection coefficient almost instantaneously.

$$\begin{aligned} P_{\text{in}}(\omega) &= \frac{1}{\mathrm{e}^{i\omega\Delta t} - \mathrm{e}^{-i\omega\Delta t}} \left[P_1(\omega)\mathrm{e}^{i\omega\Delta t/2} - P_2(\omega)\mathrm{e}^{-i\omega\Delta t/2}\right] \\ P_{\text{re}}(\omega) &= \frac{1}{\mathrm{e}^{i\omega\Delta t} - \mathrm{e}^{-i\omega\Delta t}} \left[P_2(\omega)\mathrm{e}^{i\omega\Delta t/2} - P_1(\omega)\mathrm{e}^{-i\omega\Delta t/2}\right] \end{aligned} \tag{12.1}$$

$$\Re(R) = \frac{\overline{P_{\text{re}}(t)P_{\text{in}}(\omega_j, t)}}{\overline{P_{\text{in}}^2(\omega_j, t)}} \quad \Im(R) = \frac{\overline{P_{\text{re}}(t)\left(P_{\text{in}}(\omega_j, t)\mathrm{e}^{i\pi/2}\right)}}{\overline{P_{\text{in}}^2(\omega_j, t)}} \tag{12.2}$$

Thinking about the work of Manfred Schroeder one wonders how this invention fits into his line of work, as it seems to be neither related to room acoustics (surprisingly, the patent does not refer to the use of reflection/absorption coefficients in this area) nor to speech processing or perception. A closer look reveals that it is most likely a spin-off from his research on speech (re-) production.

12.3 Determination of the Geometry of the Human Vocal Tract

In 1967, Manfred Schroeder submitted a manuscript [11] to JASA, published the next year, in which he describes a procedure involving, first, the determination of the frequency-dependent impedance of the vocal tract and, second, the derivation of the vocal tract geometry using an appropriate model. Reading the motivation behind this study, it becomes clear why Schroeder was interested in such a topic: "Knowledge of vocal-tract configuration (tongue and jaw position, lip rounding,

etc.) is basic to a better understanding of the physical and physiological processes involved in the human speech process—one of man's important means of communication. Present lack of knowledge in this area is epitomized by the variable quality and limited vocabulary exhibited by present speaking machines. . .." No, Manfred Schroeder was not interested in a mere measurement technique but had a much more visionary goal in mind, something that was also part of the zeitgeist with science fiction movies featuring talking robots and "computers".

There're several factors making this publication particularly interesting: on the one hand, the setup used to measure the impedance is very similar to the one in the patent, with the material sample replaced by the subject's mouth, and on the other hand, because of the extensive use of digital processing: the two microphone signals are digitized right away, the impedance is calculated, poles and zeros of the impedance curves (representing resonances in the vocal tract) are located and from them the geometry (cross-sectional area versus distance) is estimated by numerical inversion of Webster's horn equation, a process involving an iterative procedure which digital computers are well suited for. We may, consequently, guess that the system filed for patent is a scaled-down version of the system presented in this chapter, making use of established components.

This in not the first occasion demonstrating Schroeder's interest in digital computers, he also used them for measuring room-acoustical parameters [12] and for simulating the effect of such parameters, e.g., the frequency-dependent reverberation time, on sound perceived in a "virtual" room [13]. And all this despite the limited processing power available in the Sixties and the enormous price of digital computers. Obviously, someone believed in the future of these machines.

Now where do we stand today? The two-microphone impedance tube (with digital processing) is a standardized method (ISO 10534-2) for measuring the normal-incidence reflection coefficient. In addition, in situ methods are used to determine the impedance at arbitrary angles of incidence [14, 15]. What one may miss is a laboratory method for determining the impedance at grazing incidence.

Manfred Schroeder gives us a hint on what this can be useful for: in his paper on measurements in the Philharmonic Hall [16], he and his colleagues made an interesting observation, the seat effect: "This lack of low frequencies in the first overhead reflection revealed another low-frequency deficiency that ad hitherto gone unnoticed: A progressive attenuation of low frequencies in the direct sound as it grazes across the rows of seats. (This 'seat effect' must exist in many other halls, but it is usually masked by the presence of low-frequency components in the early overhead reflections)."

The acoustical properties of materials or structures can vary with the angle of incidence (non-local reaction), and for a correct simulation of the room acoustics data for other than normal incidence is needed.

Therefore, we shall now outline how such data can be obtained using a standing-wave tube.

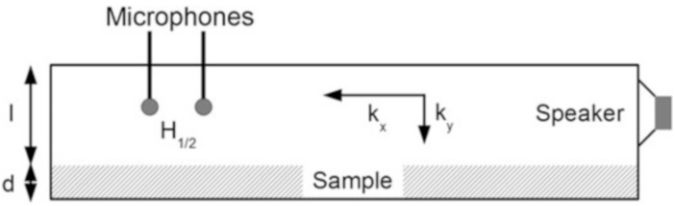


Fig. 12.2 A standing-wave tube with grazing sound incidence

12.4 Standing-Wave Tube at Grazing Incidence

Figure 12.2 shows an overview of the proposed setup. It is similar to a two-microphone impedance tube, but the sample is now placed at the bottom of the tube, which has a square cross-section of $10 \times 20\,\text{cm}^2$ and a length of 91 cm. It is made from 20 mm thick aluminum to guarantee a sufficient bending stiffness. The upper cover is height adjustable and allows the channel height to be increase by up to 8 cm. The loudspeaker at the right radiates broadband noise, which is recorded by two microphones having the same height above the sample's surface. The quantity to be measured is the transfer function $H_{1/2}$ between the microphones. To allow for differences between the sensitivities and phase responses of the microphones, the switch-microphones calibration technique [17] has been used.

This technique is an advancement of a technique described by [18] who used a Kundt's tube-like approach with a movable microphone with an otherwise identical setup.

The general effect of the (absorbing) sample at the bottom of the channel is that waves traveling above it are attenuated and do not remain plane waves. This can be exploited to find the acoustical properties of the material under test.

12.4.1 Theoretical Background

The sound field inside the tube is assumed to be two-dimensional as the depth of the channel is only 10 cm. The field can then be described by a horizontal wave number k_x and a vertical wave number k_y. In the first step, k_x is deduced from the measured transfer function $H_{1/2}$ and the distances x_1, x_2 of the microphones with respect to the hard wall at the end of the tube. We shall assume that this wall is perfectly reflecting. Equation (12.3) can then be used, keeping in mind that the microphones are located at the same height y above the sample; it can be solved numerically for k_x.

$$H_{1/2} = \frac{p_1}{p_2} = \frac{\mathrm{e}^{-ik_x x_1} + \mathrm{e}^{ik_x x_1}}{\mathrm{e}^{-ik_x x_2} + \mathrm{e}^{ik_x x_2}} = \frac{\cosh(ik_x x_1)}{\cosh(ik_x x_2)} \tag{12.3}$$

The wave number k_x is linked to the wave number k_a and characteristic impedance Z_a of the material and the geometry of the problem. The theory of Scott [19] as outlined by Eq. (12.4) has been employed, which basically states that the vertical wave components are to be continuous at the sample's surface. It is important to note that this theory (and the similar one by Mechel [20]) assumes an infinitely long channel; especially, it does not take any additional effects, i.e. the boundary condition of a source into account.

$$\sqrt{k^2 - k_x^2}\tan\left(\sqrt{k^2 - k_x^2}l\right) = \frac{1}{Z_a' k_a'}\sqrt{k_a^2 - k_x^2}\tan\left(\sqrt{k_a^2 - k_x^2}d\right) \tag{12.4}$$

Equation (12.4), too, can be solved numerically for the absorber properties. Because this would require deriving two complex quantities from a single one, the solution may be non-unique. In addition, just like in the case of in situ impedance deduction, pretests indicate that it is prone to small measurement errors [21]. Therefore, we will assume that the absorber's impedance and wave number can be described by, e.g., the Delany–Bazley model [22], so that only the single parameter "flow resistivity" has to be deduced. Other absorber models would serve as well.

12.4.2 Results and Discussion

To verify the functionality of setup and data processing, measurements were first done with an empty tube, so that the expected value for k_x is $(1 + 0i)$. Figure 12.3 shows the actual results. At low frequencies, the wave number deviates from the expected value, likely due to the (relative) proximity of the loudspeaker and hence the initially non-plane wave field. Unexpectedly, the wave number is also incorrect around 800 Hz (and at other, higher frequency ranges). In addition, within these frequency ranges, the coherency between the microphone signals (not shown) was reduced due to the sound pressure at one of the microphones being very low. Assuming plane waves, the lowest frequency a pressure minimum is expected to occur at a microphone 10 cm in front of a hard wall is 850 Hz but the minimum actually occurred at the microphone 2 cm in front of the wall.

A finite element simulation of the sound propagation inside a rigid channel with known geometry and position of the sound source was done to better understand how the sound field looks like. In Fig. 12.4, the pressure distribution inside the empty tube is shown for four frequencies. At 500 and 700 Hz, one sees the non-plane field close to the source, whereas at the left side (where the microphones are located), the wave fronts are mostly flat. On the other hand, at 850 Hz, there's a

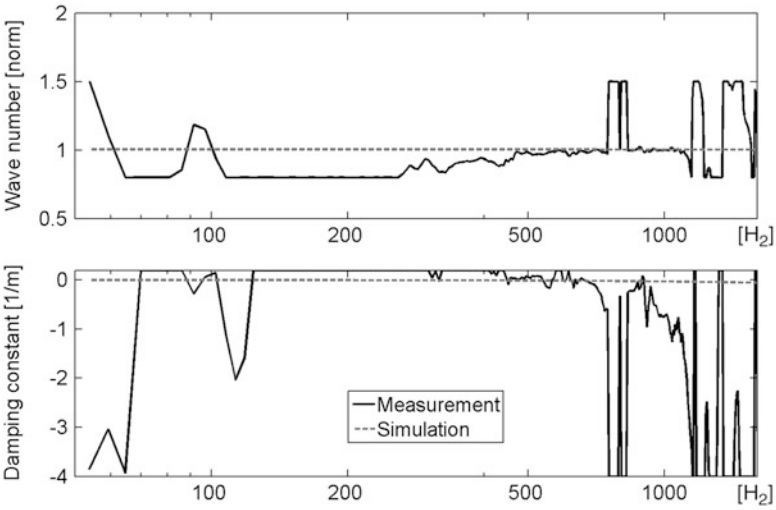


Fig. 12.3 Measured horizontal wave number in an empty tube with the microphones 2 and 10 cm in front of the hard backing. Expected value is $k_x = (1 + 0i)$

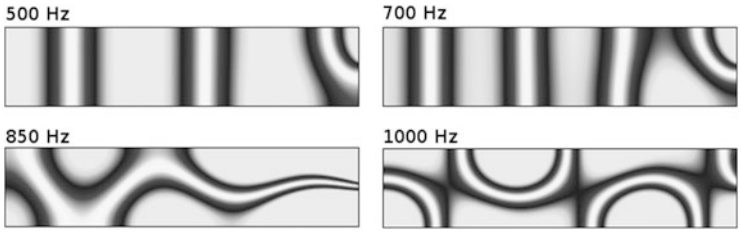


Fig. 12.4 Sound pressure distribution inside a channel of 20×91 cm^2 with rigid walls and a 8 cm loudspeaker at the *right side*. *White indicates* low, black high pressure

pressure minimum in the top left corner leading to the low coherency of the microphone signals. Though this is unexpected from a room acoustics point of view, where one would always expect pressure maxima in the corners, this is simply the result of the source not being small compared to the channel dimensions. By use of a probe microphone, the course of the pressure minimum in this corner has been verified to agree well with the FE simulation.

Further simulations show that this unwanted "source effect" can be reduced by placing a second loudspeaker at the right wall, thus making the excitation more symmetric, and by reducing the height of the channel above the sample.

With the given setup, another option was to place the microphones further away from the reflecting wall, as then the propagation loss due to the absorbing material would reduce the interference between incident and reflected wave, hence increasing the pressure in the mimima. One must be aware that this places the microphones closer to the source and might reduce the accuracy of the method at low frequencies.

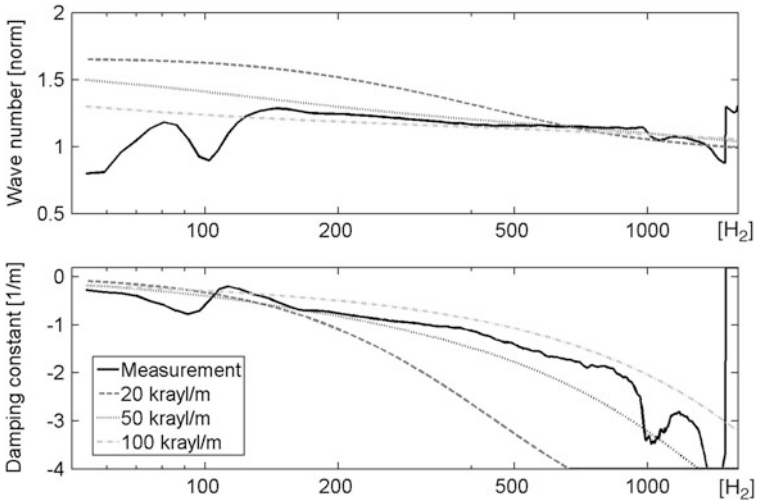


Fig. 12.5 Measured wave number above 9 cm polyurethane foam in comparison with the wave number predicted for a Delany–Bazley type absorber with three different flow resistivities

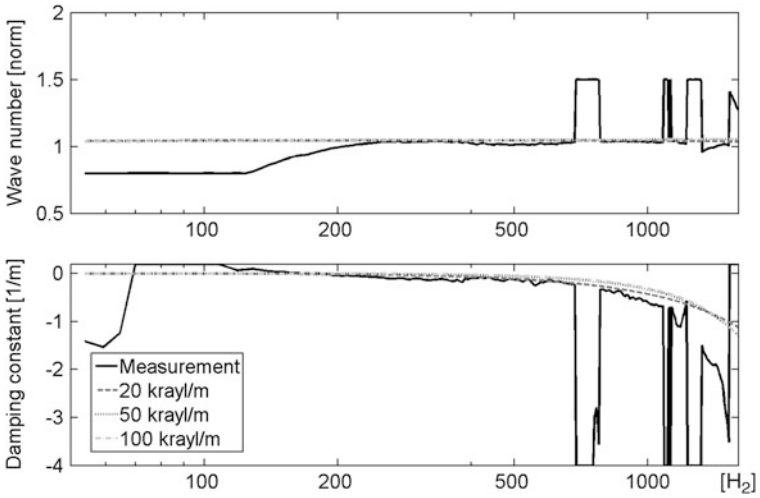


Fig. 12.6 Measured wave number above 1 cm polyurethane foam in comparison with the wave number predicted for a Delany–Bazley type absorber with three different flow resistivities

Figures 12.5 and 12.6 show the measured wave number in comparison with the wave number expected for a Delany–Bazley type absorber of same thickness. The microphones were now located 22 resp. 31.5 cm in front of the left wall. For the 9 cm thick foam, the (average) course of the wave number above 150 Hz agrees well with the one expected for a porous absorber with a flow resistivity between

50 and 100 krayl/m. Due to the high propagation loss, the sharp peaks as observed with an empty tube have disappeared.

The results for the otherwise comparable 1 cm thick foam do not prove satisfactory. On the one hand, due to the reduced absorption, the undesired extrema have reappeared. On the other hand, even the predicted courses of the wave number do hardly vary for flow resistivities of 20–100 krayl/m because a 1 cm thick layer simply does not produce a significant propagation loss within the investigated frequency range.

12.4.3 Summary and Conclusion

In 1966, Manfred Schroeder demonstrated his interest in measurement of the acoustic impedance through a publication in JASA, dealing with the impedance of the human vocal tract, and through a patent describing a two-microphone impedance tube. The knowledge of the (surface) impedance of materials and structures is required for room-acoustical simulations. The standard procedure ISO 10534-2 measures the impedance for normal incidence, though observations by Schroeder in relation to the seat effect indicate that a measurement at grazing incidence can be desirable.

In our study, we have advanced an existing but hardly known and inconvenient technique to gain a procedure which is similar to the ISO standard. The tested material is placed on the bottom of a rectangular impedance tube and the wave number in the air space above it is deduced from the sound pressure transfer function between two microphones. By use of a model for the sound propagation in such a situation, the absorber characteristics can be deduced. Measurements show that the new technique does still have its limitations: the sound field deviates from what the model predicts because the existence of the source of finite size is not taken into account. As a result, at low frequencies the incident field is not reasonably flat; additionally, pressure minima leading to poor coherency between the microphone signals occur at other locations than expected. Finite element simulations verify these findings. By careful choice of the microphone positions, credible results could be achieved for absorbent material samples. But for samples with low absorption, like a 1 cm thick open cell foam, the method is not sensitive enough to distinguish between materials with largely different flow resistivities, at least with the current measurement setup and geometry.

All in all, though improvements are still required, the new method has demonstrated its ability to determine the impedance at grazing incidence in a well-defined laboratory setting allowing for the comparison between different materials.

References

1. Schroeder, M.R.: Die statistischen Parameter der Frequenzkurve von großen Räumen. Acustica **4**(2), 594–600 (1954)
2. Schroeder, M.R.: New method of measuring reverberation time. J. Acoust. Soc. Am. **37**(3), 409–412 (1965)
3. Schroeder, M.R.: Diffuse sound reflection by maximum-length sequences. J. Acoust. Soc. Am. **57**(1), 149–150 (1975)
4. Schroeder, M.R.: An artificial stereophonic effect obtained from a single audio signal. J. Audio Eng. Soc. **6**(2), 74–79 (1958)
5. Harvey, F.K., Schroeder, M.R.: Subjective evaluation of factors affecting two-channel stereophony. J. Audio Eng. Soc. **9**(1), 19–28 (1961)
6. Schroeder, M.R.: Computer Speech: Recognition, Compression, Synthesis. Springer, New York (1999)
7. Schroeder, M.R.: Improvement of feedback stability of public address systems by frequency shifting. J. Audio Eng. Soc. **10**(2), 108–109 (1962)
8. Schroeder, M.R., Logan, B.F.: Colorless artificial reverberation. J. Audio Eng. Soc. **9**(3), 192 (1961)
9. Schroeder, M.R.: Natural sounding artificial reverberation. J. Audio Eng. Soc. **10**(3), 219–223 (1962)
10. United States Patent 3,346,067: System for determining acoustic reflection coefficients. Manfred R.Schroeder, Bell Telephone Laboratories (1967)
11. Schroeder, M.R.: Determination of the geometry of the human vocal tract by acoustic measurement. J. Acoust. Soc. Am. **41**(4), 1002–1010 (1967)
12. Atal, B.S., Schroeder, M.R., Sessler, G.M., West, J.E.: Evaluation of acoustic properties of enclosures by means of digital computers. J. Acoust. Soc. Am. **40**(2), 428–433 (1966)
13. Schroeder, M.R., Atal, B.S.: Computer simulation of sound transmission in rooms. Proc. IEEE **51**(3), 536–537 (1963)
14. Mommertz, E.: Angle-dependent in situ measurement of reflection coefficients using a subtraction technique. Appl. Acoust. **46**, 251–263 (1995)
15. ANSI/ASA S1.18: American National Standard Method for Determining the Acoustic Impedance of Ground Surfaces (2010)
16. Schroeder, M.R., Atal, B.S., Sessler, G.M., West, J.E.: Acoustical measurements in Philharmonic Hall (New York). J. Acoust. Soc. Am. **40**(2), 434–440 (1966)
17. Chung, J.Y., Blaser, D.A.: Transfer function method of measuring in-duct acoustic properties. I. Theory. J. Acoust. Soc. Am. **63**(3), 907–913 (1980)
18. Müller, W.: Messung von Absorptionseitenschaften von porösen Materialien bei streifendem Einfall (measurement of absorption properties of porous material at grazing incidence), Diplomarbeit, University of Oldenburg (in German) (1990)
19. Scott, R.A.: The propagation of sound between walls of porous material. Proc. Phys. Soc. **58**, 358–368 (1946)
20. Mechel, F.P.: Schallabsorber, vol. I, chapter 11.1: “Ebene Wellenleiter”. Hirzel Verlag, Stuttgart (in German) (1989)
21. Kruse, R., Mellert, V.: Effect and minimization of errors in in situ ground impedance measurements. Appl. Acoust. **69**(10), 884–890 (2008)
22. Delany, M.E., Bazley, E.N.: Acoustical properties of fibrous absorbent materials. Appl. Acoust. **3**, 105–116 (1970)

***

Biography



Dr. **Roland Kruse** received his Ph.D. degree in 2009 from the University of Oldenburg, under the supervision of Prof. Volker Mellert, for his work on the in situ measurement of the acoustical ground impedance. From 2009 to 2010 he continued his research at the Acoustics Group, The Open University, Milton Keynes studying the influence of the environmental conditions on source localization outdoors. After returning to the Acoustics Group at Oldenburg University, he became a lecturer at the Technical University of Braunschweig in 2011. His current research is aimed at finite element technologies and material parameter identification for biological and other composite materials.



Prof. **Volker Mellert** was head of the Acoustics Group at the Institute for Physics of Oldenburg University and retired in 2009. He received his Ph.D. in physics in Göttingen 1972 and became a Professor for Applied Physics in Oldenburg in 1976. He was President of the German Association for Acoustics (DEGA) and of the European Association for Acoustics (EAA). He served as board member of the International Commission of Acoustics and as Editor-in-Chief for ACUSTICA. He received the German Medal of Merit for his contributions in acoustics and the Helmholtz-Medal of the DEGA. He is Fellow of the Acoustical Society of America.

Chapter 13
Research on Underwater Acoustics in Göttingen

Dieter Guicking

Abstract An overview is given of the research work on hydroacoustics at the "Drittes Physikalisches Institut" of the University of Göttingen (Germany) from 1947 till 2002, under the leadership of Erwin Meyer, Manfred Schroeder, and Werner Lauterborn. Details are presented on absorbers of underwater sound, on the Institute's experimental facilities (most of them newly developed), on the statistical sound field theory, on the measurement of sound absorption and radiated power in a reverberation tank, and on natural vibrations of and sound radiation from flat plates and cylindrical shells submerged in water. Brief summaries are given on the investigation of viscometers, on self-imaging in dispersive hydroacoustic waveguides, on Schroeder diffusers for waterborne sound, on resonance scattering as applied to localize metallic objects at the sea floor, and on the construction of adjustable absorbers of underwater sound using electrorheological fluids. Much of the material has previously been published in German only. The bibliography comprises 64 entries.

13.1 Background

When Manfred Schroeder in 1968 returned to Göttingen as Director of the "Drittes Physikalisches Institut" in succession of the Institute founder Erwin Meyer, he took over a research contract with the British Navy by which investigations on certain aspects of underwater sound were sponsored. During WWII, Meyer and his team—then at the Heinrich-Hertz Institute for Vibration Research in Berlin—were forced by the Authorities to work on sound propagation and sound absorption in water, a consequence of the emerging sonar technology by which most of the German

D. Guicking (✉)
Drittes Physikalisches Institut der Universität Göttingen, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany
e-mail: Dieter.Guicking@phys.uni-goettingen.de

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_13,
© Springer International Publishing Switzerland 2015

submarines were detected and destroyed. Meyer, who disliked any war and other conflicts, was convinced that the Nazi regime would not persist very long and saved secretly copies of his classified reports on the hydroacoustics research (ignoring the mortal danger if discovered) and offered them after the war to the British and American occupying forces with the intention to get the material published for international access. The resulting extensive report [1] is still today estimated by researchers on hydroacoustics all over the world as a treasury of important fundamentals and detailed experimental results.

In 1947, Meyer was appointed Director of the newly founded third Physics Institute at the University of Göttingen, and in 1948 he succeeded in placing the research contract with the British Navy, initially for 1 year, but the contract was extended year after year for a record breaking period of 30 years until 1978—long beyond Meyer's death (March 1972). Main topics of research were the further development of absorbers of underwater sound, sound absorption in sea water and related liquids, development and improvement of experimental techniques, later on sound radiation from submerged objects, and, finally, (with another sponsor) resonance classification of metallic objects on the sea bottom. Manfred Schroeder did not involve himself in underwater acoustics very much, but he supported the hydroacoustics group and observed their achievements with interest. A close cooperation with him on statistical sound fields resulted from investigations on a "reverberation tank" (Sect. 13.3.7) and a "shallow water basin" (Sect. 13.3.6). Much of this material has been published in German only (except for the reports delivered to the Navy, which were—of course—not available to the public). A selection of the research projects is outlined here in English for the first time unless otherwise referenced.

Not included are two major hydroacoustic research fields of the Institute which have been documented elsewhere. Continuing the earlier work of Meyer's team in Berlin [1], the sound absorption and dispersion in aqueous solutions and other liquids were studied, now with emphasis on the molecular mechanisms. The extensive investigations with acoustic and microwave technologies covered a very wide frequency range. The team was headed by Reinhard Pottel and Udo Kaatze ([2], 13th and 14th Chapter, pp. 333–404). Also continuing the work described in [1], acoustic cavitation and sonoluminescence were studied in Göttingen from the Institute's founding until today, since 1968 by Werner Lauterborn and his team ([2], 6th and 7th Chapter, pp. 139–198).

13.2 Resonance Absorbers of Underwater Sound

During the war time, Meyer's team developed a thin-layer sound absorbing rubber coating for possible application to submarines. The absorber carried the cover name "Alberich" after a Germanic mythical dwarf who became invisible by a magic hat. The absorber was very effective, but to Meyer's knowledge not applied in combat until to the end of the war. However, in 1998 the wreck of the German submarine U-480 was detected by chance in the sediment of the British Channel. Divers saw

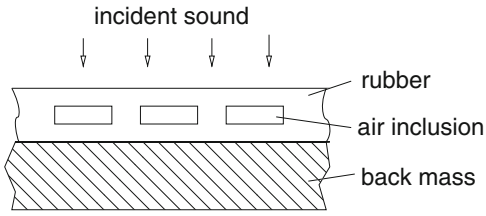


Fig. 13.1 "Alberich" absorber

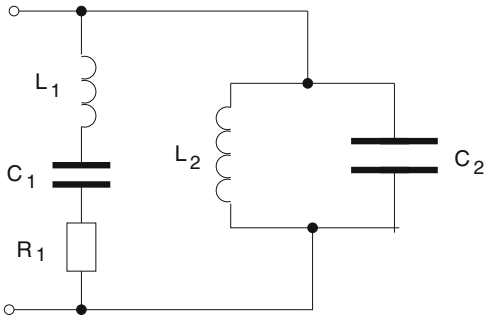


Fig. 13.2 Electrical equivalent of a mechanical two-circuit resonance absorber

the characteristic Alberich coating and realized that they had found the first "Stealth U-boat" ever built. The TV transmitter *arte* has produced an emotional film (in German) about U-480 and its history (successful operation in the Channel in August 1944, but destroyed by a mine on 24 February 1945). The film was broadcast on February 18, 2008 [3].

Figure 13.1 shows the principle of the absorber. The air inclusions in the rubber layer act as mechanical parallel resonance circuits. The sound pressure incident on the rubber surface causes compressions and depressions of the air, the vibrating rubber layer above the air cushions acts as a membrane with specific mass M_1, its bending stiffness and the air hole compression give the spring part F_1, and the loss factor of the bending vibrations determines the resistance W_1. Applying the force/voltage analogy, the electrical equivalent of a mechanical parallel circuit is a series circuit, $M_1 \Rightarrow L_1$, $F_1 \Rightarrow C_1$, $W_1 \Rightarrow R_1$. The back mass $M_2 \Rightarrow L_2$ (e.g., a submarine hull) and the resilient rubber layer with spring constant $F_2 \Rightarrow C_2$ form a mechanical series resonance circuit, i.e., an analog electrical parallel circuit, see Fig. 13.2. If both circuits are tuned to the same resonance frequency ω_0, the input impedance of the system varies rather slowly in the vicinity of ω_0, and the bandwidth of the absorber increases with the loss factor of the rubber. In the war-time development of "Alberich," the loss factor of the synthetic rubber ("Buna S") was enhanced by about 40 % carbon black filler ([1], p. 296).

After the war, more and more plastic materials with a wide variety of elastomeric properties became available, and Meyer hoped to improve the resonance absorbers by a suitable polymer instead of the rubber. It was no problem to find materials with very high loss factors, but it turned out soon that high loss is inevitably related to a strong temperature (and frequency) dispersion of the elastic modulus

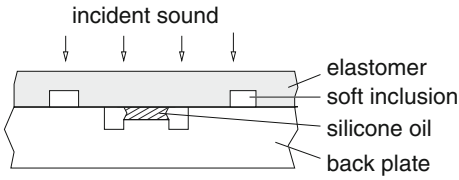


Fig. 13.3 Squeeze film absorber

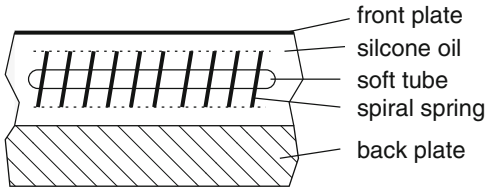


Fig. 13.4 Coil spring absorber

(quantitatively described by the Kramers–Kronig relations) so that highly effective absorbers of this type are useful only in a narrow temperature interval which prohibits practical application.

Searching for an alternative loss mechanism with only moderate temperature dependence, viscous flow of thin fluid layers was chosen. Figure 13.3 shows an early design where between the massive back plate and an elastic cover layer a thin film (0.25–0.5 mm) of viscous silicone oil is placed which serves as absorbing element. The incident sound wave deforms the elastic cover layer which acts like a piston squeezing the fluid into the ring slot and back, thereby causing viscous loss (the slots are filled with compressible material, e.g., closed-cell sponge rubber). The cover layer is a nearly lossless castable polymer (e.g., silicone rubber) which is adjusted to the specific impedance of water by soft inclusions of sponge rubber. Again, this mechanism forms a damped mechanical parallel circuit, and the cover layer with the back mass a rather lossless series circuit so that the electrical equivalent circuit of Fig. 13.2 is valid for this absorber type, too. Test samples of 5 cm diameter with a few such "flow elements" were tested in a hydroacoustic impedance tube ("Kundt's tube") and showed a frequency bandwidth (reflection coefficient below 20 %) of about 20 kHz in a temperature interval of more than 20 °C, which is better than achieved with pure relaxation loss [4–7].

This absorber construction was not suited for applications at variable depth of water because static pressure reduces the thickness of the silicone oil layer and thereby alters the flow resistance. Less sensitive to ambient pressure was the construction sketched in Fig. 13.4. Central part is a slightly stretched coil spring so that the surrounding silicone oil can oscillate through the slits between the coil turns. Inside the spring is an air-filled soft plastic tube to enable compression. Again, the setup is a two-circuit resonance absorber. With a slit width of 0.1 mm, a wire diameter of 0.4 mm, viscosity of the silicone oil of 1,000 cSt, and a back plate of 9 mm brass, the reflection coefficient stayed below 10 % from 4.5 through 13 kHz. With slightly modified dimensions, a really broadband absorber of 13.5 cm total thickness was developed, consisting of a 10.5 cm wedge-type

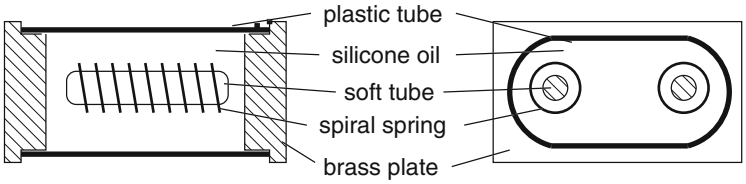


Fig. 13.5 Absorber element

absorber backed by a coil spring absorber. Its bandwidth extended from 5 kHz up to very high frequencies [8].

As a step towards commercial production, compact absorber elements were developed, consisting of an oval plastic tube with two coil spring sections inside (Fig. 13.5). The idea was to produce many such elements, fasten them on a structure to be coated, and then to embed the elements by a casting resin. The absorption coefficient of an optimized absorber of this type fell below 12 % in the frequency interval from 3.5 through 10 kHz.

The research contract with the British Navy ended in 1978 because they would no longer support unclassified research. Since also the respective German authorities insisted on classification of research results sponsored by them, Manfred Schroeder and myself decided not to continue work on absorbers of underwater sound because it was important for us to do research work within the scope of diploma and doctoral dissertations which have to be published. For a later resumption see Sect. 13.11.5.

13.3 Experimental Facilities

The hydroacoustics laboratory was equipped with conventional devices and several newly developed special appliances.

13.3.1 Deep Water Tank

A large anechoic water tank, 7 m long, 4 m wide, and 4 m deep, was installed beneath the basement of the institute [10]. The tank was lined with rib-type absorbers [11], the reflectivity of which is lower than 10 % between 7 and 70 kHz. Figure 13.6 shows a view into the empty tank, here with an installation for the measurement of sound scattering from sound-soft spheres [12].



Fig. 13.6 Deep water tank

13.3.2 Impedance Tube

Several thick-walled steel tubes (Kundt's tubes) with 5 cm inner diameter and up to 3 m length facilitated impedance measurements of absorbers by the pulse-echo method ([1], Chap. IX). A PZT transducer at the lower end of the upright standing tube transmits a sine-wave pulse of typically six periods and receives a sequence of echo pulses, reflected between the test object at the upper tube end and the transmitter. The first echo is compensated electronically. Amplitude and phase of the compensation signal are compared to the corresponding values obtained with a sound-soft reference reflector (or the free water surface) and allow the complex impedance of the test object to be calculated. Measurements were possible in the frequency range from about 3 kHz through 15 kHz. At lower frequencies the primary pulse and the echo would overlap, and at higher frequencies tube modes would be excited, violating the assumption of plane waves.

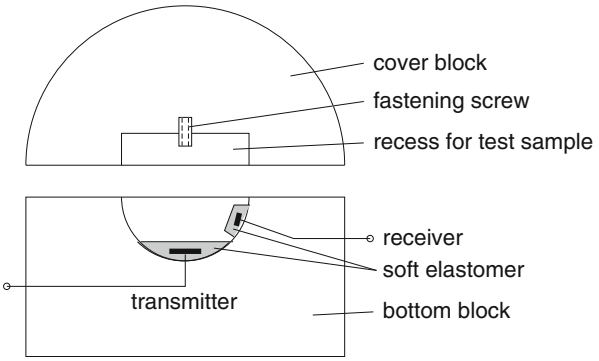


Fig. 13.7 Pressure chamber for impedance measurements

13.3.3 Pressure Chamber

In an attempt to extend impedance measurements to lower frequencies than possible with the impedance tube, a pressure chamber was developed for the frequency range from about 200 Hz to 5 kHz. After preliminary experiments the construction as shown in Fig. 13.7 turned out to be optimal. The bottom block (brass cylinder, diameter 140 mm, height 70 mm) contained a semispherical cavity (diameter 52 mm) with a transmitter (PZT or $BaTiO_4$ disk, diameter 20 mm) at the bottom and a receiver (diameter 5 mm), mounted about 3 cm apart. In order to avoid mechanical cross talk between the transducers via the chamber body, they were embedded in a very soft casting resin, the surface of which was silver coated to reduce electrical cross talk. The semispherical brass cover block contained a 15 mm deep recess of 52 mm diameter in which the test sample could be fastened by a headless screw. The two brass blocks were bolted together by six strong screws (not shown). In order to avoid air bubbles in the chamber, the two parts were assembled in a water bath. The cross talk attenuation exceeded 40 dB, which was sufficient for absorber measurements.

Assuming that the inner chamber dimensions are small compared to the wavelength, an a.c. voltage u_t applied to the transmitter creates a sound pressure p which is constant over the chamber volume. The complex compliance $\underline{C}_s$ of a sample introduced into the chamber is

$$\underline{C}_s = \frac{1}{C_0}\left[\left(\frac{u_t}{\underline{u}_r}\right)_s - \left(\frac{u_t}{\underline{u}_r}\right)_0\right] + \kappa V \tag{13.1}$$

where C_0 is the calibration constant of the chamber, to be determined with a sample of known compliance; $\left(u_t/\underline{u}_r\right)_s$ is the complex ratio of input and output voltage with the test sample in the chamber; $\left(u_t/\underline{u}_r\right)_0$ is the complex ratio of input and output voltage if the chamber is filled with water only; κ is the compressibility of the chamber fluid (here: degassed water); and V is the volume of the test sample.

Then the acoustic impedance $\underline{Z}_s$ of the test sample is

$$\underline{Z}_s = \frac{1}{j\omega \underline{C}_s}. \tag{13.2}$$

More details, including electrical equivalent circuits, calibration results, and test measurements with different fluids and resonance absorbers, have been published in [13].

13.3.4 Resiliently Lined Water Duct

The necessity of small test samples in impedance tubes or a pressure chamber is in general favorable; it can, however, be disadvantageous in case of inhomogeneous absorbers where a small sample does not represent the acoustic properties, or if it cannot be produced from constructional reasons. This is especially problematic at higher frequencies: the width of an impedance tube for measurements up to 80 kHz would be as small as 1 cm.

In order to enable hydroacoustic impedance measurements of "large area" specimens, a resiliently lined water duct of 2 m length and 34 cm width with open water surface at a maximum height of 34 cm was constructed (it is the acoustic analogue of a square waveguide in microwave technology). In the basic experimental design, a test sample of size 34 × 34 cm filled the cross-section at one end of the duct, an equally sized capacitive transmitter was placed at the other end. The measurements were made with a procedure known from the Kundt's tubes ([1], Chap. IX): a hydrophone is moved lengthwise through the duct, evaluated is the standing wave ratio and the shift of a pressure minimum in comparison to a sound-soft reference reflector (the modern two-microphone technique with standing-wave separation would also be applicable).

In contrast to hard-walled waveguides (which can hardly be realized for water-borne sound), the sound-soft edge conditions (acoustic "short circuit") cause dispersion and lead to a higher phase velocity in the duct than in the free field. If f is the frequency of a sinusoidal wave, the free-field wavelength is $\lambda_0 = c_0/f$ with the free-field sound velocity c_0 (1,484 m/s in water at room temperature). Sound propagation between sound-soft walls can be understood as zigzag reflection ([27], Sect. 2.1.3). The oblique lines in Fig. 13.8 indicate wave fronts of a wave propagating with free-field sound velocity c_0, incident on the test object at an angle $\varphi > 0°$. Zero sound pressure at $z = 0$ and $z = h$ requires $h = n\lambda_z/2 = n\lambda_0/2 \sin\varphi$, $n = 1, 2, \ldots$ The phase velocity in x-direction is $c_x = \lambda_x f = c_0/\cos\varphi$. For a water height h and square cross-section, a unique wave mode (the 1,1-mode, $n = 1$) exists if λ_z lies in the range $h < \lambda_z < 2h$. If the hydrophone slides exactly along the center line, the next duct mode ($h = \lambda_z$) is suppressed because it has a pressure node at the center line so that the usable frequency range is extended to $2h/3 < \lambda_z < 2h$. The angle of incidence is

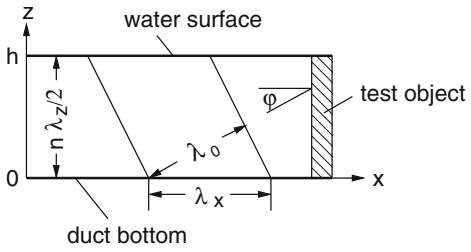


Fig. 13.8 Resiliently lined water duct

$$\varphi = \arccos\frac{c_0}{c_x} = \arccos\left(\sqrt{1-(\lambda_0/2h)^2}\right), \tag{13.3}$$

which means that it is frequency dependent at constant water height (and width in the square duct). To avoid this, measurements at different frequencies can be made with correspondingly adjusted duct cross-section, realized by a movable length-wall, and adjusted height of water. At 6.8 kHz the full cross-section of 34 × 34 cm is appropriate, and at 15 kHz: 15.5 × 15.5 cm. The angle of incidence in both cases is $\varphi \approx 18°$. Experimental results for various absorbers have been published in [8, 9].

13.3.5 Wide Impedance Tube

Since the aforementioned water duct applies to comparatively low frequencies only, we developed a device for impedance measurements in water up to about 200 kHz, employing a tube with diameter much larger than the wavelength. The problem of generating a plane wave over the whole tube cross-section was solved with a sound-soft paraboloid reflector, machined from polystyrene foam ("Styrofoam"), as sketched in Fig. 13.9. Two miniature PZT transducers were placed at the focal point. Their separation was much smaller than a wavelength at the highest frequency of interest. The usual pulse-echo method was applied, demanding for a minimum pulse length of six periods. Thus, in order to avoid overlap of signal and echo, the tube length must exceed three wavelengths at the lowest measuring frequency. Since, on the other hand, the tube width should be much larger than a wavelength, the tube had—in contrast to Kundt's tubes—not to be much longer than wide. The "screen" above the transducers (made from an elastomer filled with cork particles) served as a wedge-type absorber for the spherical direct sound wave which would interfere with the reflected pulse. The screen was fixed by three thin, stiff wires which did not disturb the sound field. The tube wall above the paraboloid was also lined resiliently in order to weaken the discontinuity at the bend which is source of a weak diffraction wave. An advantage of the parabolic reflector is that disturbance waves are not focused to the receiver and thus do not significantly contribute to the output signal. The PVC stop at the top served to fix the foam against buoyancy, and it was the support for the test samples resp. the reference

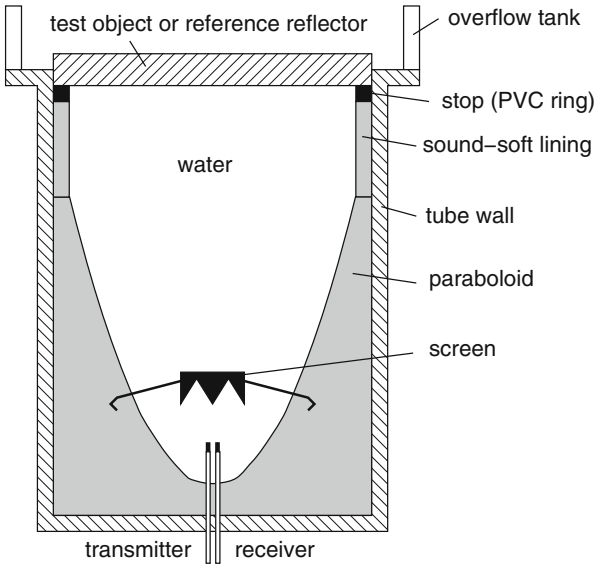


Fig. 13.9 Wide impedance tube (not to scale)

reflector. The "overflow tank" allowed to raise the water level so that the test objects could be inserted obliquely to remove air bubbles (which would strongly impair the experimental results).

Two tubes were built: Tube I with inner diameter 29 cm and tube length 60 cm, for frequencies above 10 kHz, and tube II with inner diameter 12.5 cm and length 30 cm, for frequencies above 30 kHz. With the sound-soft reference reflector, sequences of typically four echoes were observed. The impedance measurements were performed as with the standard impedance tube by compensating the first echo pulse and comparing amplitude and phase of the compensation signal with the corresponding values for the reference reflector.

More details about the experimental technique and measurements with water layers of varying height as well as various absorber samples have been published in [14].

13.3.6 Shallow Water Basin

A two-dimensional sound field is suited for certain experimental investigations, e.g., sound scattering, radiation, and absorption measurements, and also to check theoretical predictions for two-dimensional statistical sound fields, see Sect. 13.4.2.

To this end, a shallow water basin (4 m long and 3 m wide) with horizontally leveled chipboard bottom and 30 cm high sidewalls was constructed. Sheets of 3 mm thick polystyrene foam were laid on the chipboard and served as sound-soft

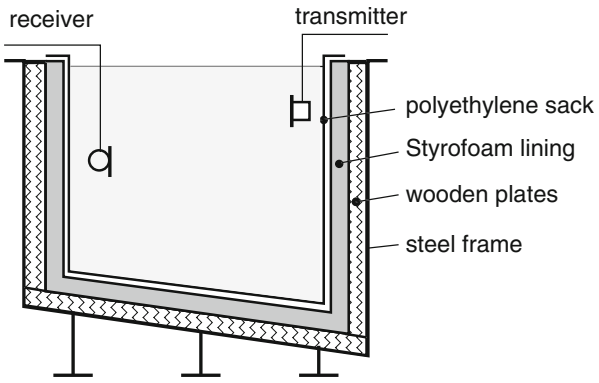


Fig. 13.10 Reverberation tank

underside. A watertight plastic foil covered the foam and was pulled up along the sides. The sidewalls could be lined with sound-soft reflector plates to create a two-dimensional reverberation basin, or by rib-type absorbers. The sound-field in the basin was scanned by a small hydrophone (Celesco LC-5) with motor-driven support. It could be moved either in two directions through the whole basin area with constant speed and at constant depth, or by a rotating cantilever arm along a semicircle to record sound radiation patterns of objects placed near the center of one long side. The speed of the hydrophone motion was slow enough to avoid generation of surface waves and eddies in the water.

13.3.7 Reverberation Tank

In analogy to the reverberation rooms in airborne acoustics for the measurement of sound absorption and sound power radiation, a reverberation tank for similar hydroacoustic measurements was constructed as sketched in Fig. 13.10. The tank had an irregular shape as common for reverberation rooms. The walls consisted of wooden boards which were supported by a steel frame. In contrast to former reverberation tanks with steel walls [15–17], this tank was lined with polystyrene foam ("Styrofoam" of Dow Chemical), glued to the wooden plates. The mean tank dimensions were $1.60 \times 1.43 \times 1.03$ m. The sound-soft lining covered an area of 8.7 m^2, the free water surface was 2.3 m^2, and the water volume 2.4 m^3. The water was softened to prevent lime precipitation, and it was degassed to avoid air bubbles which are known to absorb sound ([1], Chap. II).

The reverberation time T_{60} of the tank in this state varied between 80 ms and 160 ms in the frequency range from 4 kHz through 200 kHz. It was found in preliminary experiments that the reflectivity of the polystyrene foam could be enhanced drastically by coating it with a thin plastic foil (polyethylene, thickness

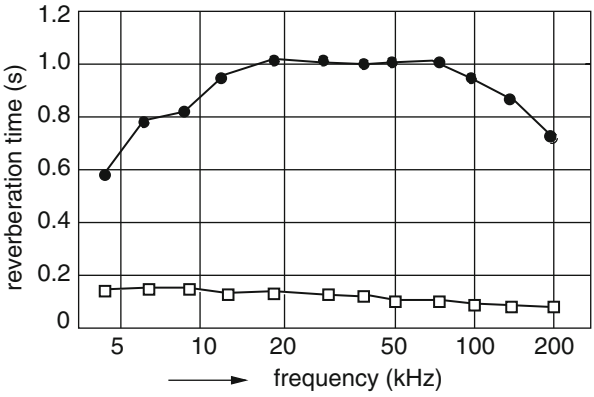


Fig. 13.11 Reverberation time of the water tank. *Lower curve*: without plastic foil, *upper curve*: with foil

150 µm): T_{60} exceeds 500 ms in the whole frequency range with a maximum of 1 s between 18 and 70 kHz [18], see Fig. 13.11. The reason for this effect is that the water is prevented from penetrating into the narrow surface pores of the foam where it forms tiny flow absorbers. The absorption coefficient of the tank walls was about 0.02, the Q-factor reached 1,400 at 5 kHz, 23,000 at 50 kHz, and 62,000 at 200 kHz.

13.4 Statistical Sound Field Theory

13.4.1 Three-Dimensional Sound Field

Manfred Schroeder, as a young student at Erwin Meyer's Institute, has developed the statistical theory of reverberant sound fields for airborne sound [20–23], but the formalism can be applied to water with slight modifications. Experiments with the reverberation tank were performed in the scope of a Ph.D. Thesis [19] and have not been published somewhere else. Therefore, they shall be described here in some detail. The measurement of absorption coefficients requires reliable values of the reverberation time. Sufficient accuracy can be achieved only by averaging over many independent measurements. This is also true for determining the total acoustic power radiated from vibrating objects. Strategies for getting independent measurements can be deduced from Schroeder's theory.

A reverberation room is a resonator, operated in the frequency range of overlapping modes. Manfred Schroeder has stated in [22] that the average spacing df of the resonance frequencies must be smaller than one-third of the halfwidth Δf of a resonance peak, which is related to the reverberation time T_{60} by

$$\Delta f = \frac{3\ln 10}{\pi T_{60}} \approx \frac{2.2}{T_{60}}. \tag{13.4}$$

The value of df can be obtained in the following way. The number N of modes in a room with volume V below a frequency limit f_g is

$$N = \frac{4\pi V}{3c^3} f_g^3 \tag{13.5}$$

where c is the sound velocity. (The usually added correction terms for tangential modes in reverberation rooms do not apply here because such modes do not exist on sound-soft walls.) The average interval df between two successive eigenfrequencies, as calculated from (13.5), is

$$\mathrm{d}f = \frac{c^3}{4\pi V f^2}, \tag{13.6}$$

provided $\mathrm{d}f \ll f$. The condition of $\Delta f/\mathrm{d}f > 3$ was fulfilled in the reverberation tank above 4 kHz.

A stationary sound field in the reverberation tank can be understood as the superposition of resonant modes which are excited by the transmitter signal (nearly 200 modes in the lowest third-octave band about 4.5 kHz; the number increases in proportion to f^3 due to (13.5)). An essential change of the sound field, i.e., a new mode configuration, is achieved by a frequency shift greater than

$$\delta f = \frac{\Delta f}{2} \approx \frac{1.1}{T_{60}} \tag{13.7}$$

if the modes overlap. An equivalent variation can be reached by a volume change δV, in the reverberation tank easily done by draining some water. According to (13.5), δV is related to δf by

$$\frac{\delta V}{V} = 3\frac{\delta f}{f}. \tag{13.8}$$

Thus, the volume change δV is equivalent to δf of (13.7) if

$$\delta V = 3\frac{V}{f} \cdot \frac{\Delta f}{2} \approx \frac{3.3V}{fT_{60}}. \tag{13.9}$$

Another way to alter the sound field is to change the room shape, e.g., by submerging an object in the tank. Obviously, the sound field at the hydrophone location is changed significantly if the object volume is of the order of δV as given by (13.9).

The sound field modification by changing the tank shape is also equivalent to changing the hydrophone position. For two points half a wavelength apart, the spatial cross-correlation coefficient of the sound energy vanishes, and it stays below 0.05 for larger distances. This was proved for excitation with a single frequency or narrow-band noise [24]; for excitation by third-octave noise the spatial decay of the cross-correlation function will be still faster. Thus, the sound fields at positions half an average wavelength apart (or more) are statistically independent.

Summarizing, there are four possibilities to obtain statistically independent measures of the sound field in the reverberation tank:

(a) Change of the excitation signal by frequency shift.
(b) Change of water volume.
(c) Submerging objects.
(d) Different hydrophone positions.

Assuming Gaussian processes, the four methods are theoretically equivalent, and their effects on measurable quantities (e.g., stationary sound pressure level and reverberation time) can be predicted theoretically. According to [23, 24], the standard deviation σ_L of the stationary sound pressure level L depends on the reverberation time T_{60}, the bandwidth B of the excitation signal, and the number n of independent measurements:

$$\sigma_L = \frac{5.57}{\sqrt{1 + 0.238 T_{60} B} \cdot \sqrt{n}} \text{ dB}, \tag{13.10}$$

assuming interference of at least ten excited modes.

From experimental reasons, the time interval $T_{15} = T_{60}/4$ for 15 dB decay was measured instead of T_{60}, in practice as the time interval between two instants at which the reverberation curve crossed for the first time two pre-adjusted levels $\Delta L = 15$ dB apart. The scatter of the switching instants is $\sigma_t = \sigma_L \cdot \Delta t/\Delta L$ where $\Delta t/\Delta L = T_{60}/60$ dB is the slope of the reverberation curve. The dependence of T_{15} on two such measurements leads to $\sigma_{T_{15}} = \sqrt{2}\sigma_t$ and $\sigma_{T_{60}} = 4\sigma_{T_{15}}$, thus with (13.10)

$$\frac{\sigma_{T_{60}}}{T_{60}} = \frac{\sqrt{2}}{15}\sigma_L = \frac{5.57 \cdot \sqrt{2}}{15\sqrt{1 + 0.238 T_{60} B}\sqrt{n}}. \tag{13.11}$$

The bandwidth $B = 0.2316 f$ of third-octave band noise yields

$$\frac{\sigma_{T_{60}}}{T_{60}} = \frac{0.525}{\sqrt{1 + 0.0551 T_{60} f} \cdot \sqrt{n}} \approx \frac{2.24}{\sqrt{T_{60} f n}} \tag{13.12}$$

if the term "1" under the root is neglected.

The scatter of reverberation time measurements was investigated experimentally. In a first series, the scatter was recorded for excitation with pulses of third-octave band random noise about 6.4 kHz. Without averaging ($n = 1$) and $T_{60} \approx 0.75$ s, (13.12) yields a value of 0.032 or 3.2 %. The finite 3 % resolution of the level

recorder used in the experiments has to be considered, too: $\sqrt{3.2^2 + 3^2} = 4.4\,\%$, which is a bit smaller than the measured standard deviation of 5.6 %. Averaging over 25 repetitive recordings should reduce the scatter to $1/\sqrt{n} = 1/5$ of the former value, i.e., 0.9 %, and this was found experimentally.

While in this test the sound field was kept constant, in subsequent experiments the sound field was modified by the above-mentioned methods (b) and (d). In one series, the tank was excited by third-octave band random noise about 6.4 kHz. The water volume was stepwise reduced 25 times by decrements of $\Delta V = \delta V/2 = 1.65V/(fT_{60}) = 1,400\ \mathrm{cm}^3$. For each of these 26 volumes, 25 repetitive decay processes were recorded. Single values for different tank volumes jumped statistically with a standard deviation of 7 %. Averaging over the 25 repetitions reduced this value to 2 %. Similar values were obtained from measurements with third-octave band noise around 12.7, 25.5, 50.9, and 101.9 kHz.

In a further series of scatter measurements, the water volume and the transmitter position were kept constant, but the receiver, a small hydrophone of the type Celesco LC-5, was shifted 25 times by $\Delta x = \lambda/4$ of the mean frequency, taking again 25 repetitive measurements for each hydrophone position. The displacement of the hydrophone support (a thin brass tube) modified the mode pattern a little, thereby further reducing the correlation of subsequent data. As excitation noise pulses, the same frequency bands were applied as before. The acoustically active PZT element of the hydrophone was a cylinder of 3.5 mm length and 2 mm diameter which is much smaller than the shortest wavelength used so that spatial averaging by the finite hydrophone size is insignificant. The measured scatter was almost the same as in the series with volume change, confirming the theoretical predictions.

13.4.2 Two-Dimensional Sound Field

A two-dimensional sound field can also be treated with statistical methods for higher frequencies where sufficiently many modes overlap. As in the resiliently lined water duct (Sect. 13.3.4), the basin wavelength λ_b depends on frequency $f = c_0/\lambda_0$ and water depth h:

$$\lambda_b = \frac{\lambda_0}{\sqrt{1 - \left(\frac{\lambda_0}{2h}\right)^2}} \tag{13.13}$$

(c_0 and λ_0 are the free-field sound velocity and wavelength). According to Cremer [25], the number of eigenfrequencies below a limit f_g is

$$N = \frac{\pi S f_g^2}{c_b^2} \tag{13.14}$$

where S is the surface area of the basin and $c_b = \lambda_b f$ the phase velocity of the waves in the basin. Other than in the 3D field, N increases with f_g^2 instead of f_g^3 (13.5). From (13.14) we get the mean spacing between adjacent eigenfrequencies in the vicinity of f_g:

$$\mathrm{d}f = \frac{c_b^2}{2\pi S f_g} \tag{13.15}$$

(instead of (13.6) for the 3D field). A unique sound pressure distribution in the vertical direction occurs for water depths h if

$$\frac{\lambda_0}{2} < h < \lambda_0. \tag{13.16}$$

For $h = \lambda_0$, (13.13) yields $\lambda_b = 1.155\lambda_0$ as lower limit of λ_b. With the condition of Schroeder's large room frequency f_{cr}: $\mathrm{d}f/\Delta f \geq 3$, Δf from (13.4), we get for the shallow water basin

$$f_{cr} = \frac{c_b^2}{4S} T_{60}. \tag{13.17}$$

Applying this to our basin (3.9×2.9 m within the resilient lining) with $T_{60} \approx 0.5$ s, $S = 11.4$ m^2, and a two-dimensional phase velocity of $c_b = 1,700$ m/s, we calculate $f_{cr} = 32$ kHz, or a maximum water depth of 5 cm for the statistical theory to be applicable.

Theoretical probability density functions of the complex sound pressure amplitudes and phases, of the sound pressure level, of spatial frequencies in an isotropic diffuse sound field and in a semi-diffuse sound field, statistical parameters of the spatial transfer functions, energy flux density and phase rotation and derivatives have been derived and published in [26], together with experimental results from single-frequency wave fields in the shallow water basin. The experimental data confirmed the theoretical predictions, except for excitation frequencies below f_{cr} if one eigenmode dominates. The semi-diffuse sound field was realized by lining part of the basin with wedge-type absorbers, and with maximum-length diffusers elsewhere. It turned out that the weak reflections from the absorber cause a small maximum in the power spectrum at negative spatial frequencies and modulate the spatial pattern of energy density so that the mean height of maxima is drastically reduced.

13.5 Sound Absorption Measurement

The reverberation tank was used for the measurement of sound absorption. The test object was a "large" (34×34 cm) coil spring absorber (of the type sketched in Fig. 13.4). The absorber was suspended closely below the water surface. The

absorption coefficient α can be evaluated from the reverberation times T_0 and T of the tank without and with absorber, respectively. Assuming Eyring's theory (e.g., [27], p.73) the sound energy E decays from an initial value E_0, after switching off the sound source, according to

$$E = E_0(1 - \overline{\alpha})^{(Act/4V)} \tag{13.18}$$

where V is the tank volume, A its total surface area, $c = 1{,}484$ m/s the sound velocity, and $\overline{\alpha} = \left(\sum_n \alpha_n A_n\right)/A$ the average absorption coefficient of all boundary surfaces. With $E/E_0 = 10^{-6}$ for the reverberation time, $t = T$, (13.18) yields

$$1 - \overline{\alpha} = \mathrm{e}^{-\frac{kV}{AT}} \tag{13.19}$$

where $k = (24 \ln 10)/c = 0.03724$ s/m. Let α_0 and T_0 be the quantities for the tank without absorber, $\overline{\alpha}$ and T with absorber, A_s the absorber surface, and α its absorption coefficient. $\overline{\alpha}$ is defined as

$$\overline{\alpha} = (\alpha_0(A - A_s) + \alpha A_s)/A, \tag{13.20}$$

and with

$$\alpha_0 = 1 - \mathrm{e}^{-\frac{kV}{AT_0}} \tag{13.21}$$

we get

$$\alpha = 1 - \mathrm{e}^{-\frac{kV}{AT_0}} + \frac{A}{A_s}\left(\mathrm{e}^{-\frac{kV}{AT_0}} - \mathrm{e}^{-\frac{kV}{AT}}\right). \tag{13.22}$$

Assuming small exponents and $A_s \ll A$, (13.22) is approximated by

$$\alpha \approx \frac{kV}{A_s}\left(\frac{1}{T} - \frac{1}{T_0}\right). \tag{13.23}$$

The influence of the scatter of the experimental reverberation times T_0 and T on the accuracy of α can be estimated. Let σ_T and σ_0 be the standard deviations of T and T_0, respectively. Then the standard deviation of α is

$$\sigma_\alpha = \sqrt{\left(\frac{\partial \alpha}{\partial T}\right)^2 \sigma_T^2 + \left(\frac{\partial \alpha}{\partial T_0}\right)^2 \sigma_0^2} = \frac{kV}{A_s}\sqrt{\frac{\sigma_T^2}{T^4} + \frac{\sigma_0^2}{T_0^4}} \tag{13.24}$$

and the relative error

$$\frac{\sigma_\alpha}{\alpha} = \sqrt{\frac{T_0^2\left(\frac{\sigma_T}{T}\right)^2 + T^2\left(\frac{\sigma_0}{T_0}\right)^2}{T_0 - T}} \tag{13.25}$$

or, assuming $\sigma_T/T \approx \sigma_0/T_0$,

$$\frac{\sigma_\alpha}{\alpha} = \frac{\sigma_T}{T}\,\frac{\sqrt{1 + T^2/T_0^2}}{1 - T/T_0}. \tag{13.26}$$

If, as in the measurements described in the following, $T = 0.7T_0$, we conclude from (13.26) that the error of α is about four times greater than that of T and T_0.

Measurements with the "large area" coil spring absorber were made with third-octave band random noise between 4.5 and 72 kHz and showed a rather wide scatter of single measurements, typically $\alpha = 0.7 \pm 0.3$ (about 40 %) at 4.5–15 kHz, shrinking to about 4 % at 72 kHz. Averaging over 13 hydrophone positions and either 10 repetitions reduces the scatter to 20 % at low and 2 % at high frequencies. Obviously, reliable values of the absorption coefficient at low frequencies demand for averaging over many more independent measurements.

An effect which has not been considered in this context is the occurrence of (cylindrical) diffraction waves originating from the edges of the test absorber. They propagate across the absorbing surface and cause excess absorption. This "edge effect" or "area effect" influences the experimental results. We showed in a later theoretical study that the edge effect is much stronger for absorbers on pressure-release boundaries than on hard walls [28]. This was confirmed by experiments [29].

13.6 Measurement of Power Radiation

The acoustical power P radiated by a vibrating object into a reverberation room is related to the average steady state energy density E_0 of the stationary sound field by

$$P = \frac{E_0 \overline{\alpha} A c}{4} \tag{13.27}$$

where $\overline{\alpha}$ is the average absorption coefficient and A the total surface area of the reverberation room [30]. According to (13.19) $\overline{\alpha}$ is given by the reverberation time T:

$$\overline{\alpha} = 1 - \mathrm{e}^{-\frac{kV}{AT}} \approx \frac{kV}{AT} = 24 \ln 10 \frac{V}{AcT} = 55.262 \frac{V}{AcT}, \tag{13.28}$$

and the stationary energy density

$$E_0 = \frac{\overline{p_{\text{eff}}^2}}{\rho c^2} \tag{13.29}$$

by the r.m.s. sound pressure p_{eff} (ρ is the mass density of the medium, and the bar denotes spatial averaging). Hence, the sound power P can be determined by measuring the sound pressure p and the reverberation time T:

$$P = 13.82 \frac{\overline{p_{\text{eff}}^2} V}{\rho c^2 T}. \tag{13.30}$$

The reverberation room technique for the measurement of power radiation is particularly useful for objects with pronounced directional characteristics since the room integrates over all angles of radiation. The sound pressure measurements have to be taken in the diffuse field, i.e., at sufficient distance r from the radiator. According to [22], the following condition has to be met:

$$r > \frac{1}{4}\sqrt{\frac{Q\overline{\alpha}A}{1-\overline{\alpha}}} \approx 0.048\sqrt{\frac{QV}{T}} \tag{13.31}$$

where Q is the directivity factor of the source (in the direction of the receiver). The approximation is obtained with $\overline{\alpha}$ from (13.28), $c = 1,484$ m/s, and $\overline{\alpha} \ll 1$.

When investigating vibrating plates or similar objects one often is not so much interested in the absolute power radiated but in the radiation efficiency s which is defined as the ratio of the actual power P radiated by the object to the power which would be radiated by an equally large area of an infinite piston vibrating with the same r.m.s. particle velocity v_{eff} [31]:

$$s = \frac{P}{\rho c S \overline{v_{\text{eff}}}^2} \tag{13.32}$$

where ρc is the characteristic impedance of the surrounding medium, S the radiating surface, and $\overline{v_{\text{eff}}}$ the spatial average of the r.m.s. velocity of the radiating surface. According to (13.30) and (13.32), the radiation efficiency can be determined in a reverberation room by measuring the radiated power P and the reverberation time T, or with (13.30)

$$s = \frac{13.82 \overline{p_{\text{eff}}^2} V}{\rho^2 c^3 T S \overline{v_{\text{eff}}}^2}. \tag{13.33}$$

13.7 Diffuseness in the Reverberation Tank

A sound field is diffuse in a point if the acoustic energy flux density through this point is equally distributed over all solid angles [32]. Diffuseness is assumed in the reverberation theories and has therefore to be guaranteed, e.g., for the measurement

of sound absorption and power radiation in reverberation rooms. Besides measurements with directional microphones [33] or deviations from the spatial $1/r^2$ decay of the energy density with distance r from a sound source [34], Manfred Schroeder has developed methods for the measurement of diffuseness based on the statistical theory of reverberant sound fields. We have supplemented them as outlined in the following [35].

13.7.1 Angular Distribution

A point source at the origin of a coordinate system may radiate a sinusoidal signal $\underline{a}_1(t) = \hat{a}_1 e^{j\omega t}$ and create a stationary sound field in a room. A microphone or hydrophone displaced in x-direction may record the output signal $\underline{a}_2(t) = \underline{\hat{a}}_2 e^{j\omega t}$. The ratio

$$\underline{H}(x) = \frac{\underline{a}_2(t)}{\underline{a}_1(t)} = R(x) + jX(x) = \left|\underline{H}(x)\right| e^{j\varphi(x)} \tag{13.34}$$

is the complex transfer function between transmitter and receiver. A plane wave propagating at an angle α against the x-direction has the wavenumber component $k_x = 2\pi(\cos\alpha)/\lambda$. The sound field $\underline{a}_2(t)$ and the function $\underline{H}(x)$ at the receiver position are composed of many plane waves incident under angles α_m:

$$\underline{H}(x) = \sum_m \underline{\hat{a}}_m e^{-2\pi j \frac{\cos\alpha_m}{\lambda} x}, \tag{13.35}$$

and equivalently for the other directions: $\underline{H}(y)$ with $\underline{\hat{b}}_m$ and β_m, $\underline{H}(z)$ with $\underline{\hat{c}}_m$ and γ_m. Angular equal distribution of the energy flux means

$$\overline{\cos\alpha} = \left(\int_0^\pi \left|\underline{\hat{a}}(\alpha)\right|^2 \cos\alpha \, d\alpha \right) / \left(\int_0^\pi \left|\underline{\hat{a}}(\alpha)\right|^2 d\alpha \right) = 0 \tag{13.36}$$

and correspondingly $\overline{\cos\beta} = \overline{\cos\gamma} = 0$. Since generally

$$\overline{\cos^2\alpha} + \overline{\cos^2\beta} + \overline{\cos^2\gamma} = 1, \tag{13.37}$$

in a diffuse sound field $\overline{\cos^2\alpha} = \overline{\cos^2\beta} = \overline{\cos^2\gamma} = 1/3$. $\overline{\cos\alpha}$ measures the mean energy flux, $\overline{\cos^2\alpha}$ the width of the energy flux distribution. Experimental data of $\overline{\cos\alpha}$, $\overline{\cos\beta}$, $\overline{\cos\gamma}$ and, e.g., $\overline{\cos^2\alpha}$, $\overline{\cos^2\beta}$ ($\overline{\cos^2\gamma}$ follows from (13.37)) provide information on the directional energy flux distribution. Omitting detailed derivations, the following comparatively simple experimental procedures can be deduced.

13.7.2 Spatial Distance of Pressure Maxima

Assuming that the first (y') and second (y'') spatial derivatives of $y = \left|\underline{H}(x)\right|^2$ are Gaussian distributed, S. O. Rice [36] has shown that the mean spatial distance of sound pressure maxima is given by

$$\overline{\Delta x_{\max}} = 2\pi\sqrt{\frac{\overline{y'^2}}{\overline{y'^2}}} \approx \lambda\sqrt{\frac{\overline{\cos^2\alpha} - \overline{\cos\alpha}^2}{\overline{\cos^4\alpha} - 4\,\overline{\cos^3\alpha}\cdot\overline{\cos\alpha} + 3\,\overline{\cos^2\alpha}^2}}. \tag{13.38}$$

For a diffuse three-dimensional sound field follows $\overline{\Delta x_{max}} \approx 0.79\lambda$. In a "semi-diffuse" sound field, e.g., in a reverberation room with one totally absorbing wall (perpendicular to the x-direction) is $\overline{\cos\alpha} = 1/2$ and further $\overline{\Delta x_{\max}} \approx 1.5\lambda$. One concludes that the mean distance of maxima in a monofrequent stationary sound field depends on the angular distributions.

13.7.3 Zeros of R(x)

The average spatial distance of the zeros of $R(x)$ (see (13.34)) is

$$\overline{\Delta x_0} = \frac{\lambda}{2\sqrt{\overline{\cos^2\alpha}}} \tag{13.39}$$

and correspondingly for the other directions. From (13.37) follows

$$\frac{1}{\overline{\Delta x_0}^2} + \frac{1}{\overline{\Delta y_0}^2} + \frac{1}{\overline{\Delta z_0}^2} = \frac{4}{\lambda^2}, \tag{13.40}$$

and for the three-dimensional diffuse sound field

$$\overline{\Delta x_0} = \overline{\Delta y_0} = \overline{\Delta z_0} = \lambda\frac{\sqrt{3}}{2}. \tag{13.41}$$

If deviations from this equality are found, they indicate lacking diffuseness but do not give information on the energy flux direction since only the quadratic moments are considered.

13.7.4 Spatial Phase Rotation

The average over x of the spatial phase rotation $\mathrm{d}\varphi/\mathrm{d}x$ (see (13.34)) must vanish. Generally holds

$$\overline{\left(\frac{\mathrm{d}\varphi}{\mathrm{d}x}\right)} = -\frac{2\pi}{\lambda}\ \overline{\cos\alpha}, \tag{13.42}$$

i.e., this quantity measures the mean energy transport.

13.7.5 Spatial Correlation Functions

The measurement of diffuseness by spatial correlation functions is known since long [37, 38]. A few supplements are given in the following. Let

$$c_{RR}(\Delta x) = \left(\int R(x)R(x+\Delta x)\mathrm{d}x\right) / \left(\int R^2(x)\mathrm{d}x\right) \tag{13.43}$$

and

$$c_{RX}(\Delta x) = \left(\int R(x)X(x+\Delta x)\mathrm{d}x\right) / \sqrt{\int R^2(x)\mathrm{d}x \int X^2\mathrm{d}x}. \tag{13.44}$$

If Δx_1 is the first zero of $c_{RR}(\Delta x)$, a good approximation is

$$\overline{\cos^2\alpha} \approx \left(\frac{\lambda}{3.6\Delta x_1}\right)^2 \tag{13.45}$$

(correspondingly for the y- and z-direction). The measurement can also be done with octave-band filtered noise; let λ be the geometric mean of the wavelengths at the band limits, then it follows for the diffuse sound field $\Delta x_1 = 0.49\lambda$ instead of 0.50λ for sinusoidal excitation. Well beyond the first zero, $c_{RR}(\Delta x)$ is almost equal for both cases.

While $c_{RR}(\Delta x)$ depends only on the width of the angular distribution, $c_{RX}(\Delta x)$ informs about the energy flux; assuming $\Delta x \ll \lambda/2\pi$, we find

$$c_{RX}(\Delta x) \approx \frac{4\pi}{\lambda}\Delta x\overline{\cos\alpha}. \tag{13.46}$$

If measurements with frequency mixtures are made, the higher frequency components are overestimated due to the factor $1/\lambda$. This can be avoided by temporal integration of one component. Let $p_1(t)$ and $p_2(t)$ be the receiver signals at two points $\Delta x \ll \lambda/2\pi$ apart. Then the mean energy flux is

$$I = \frac{1}{\rho\Delta x}\overline{\langle p_1\rangle p_2}^t \tag{13.47}$$

where ρ is the density of the medium and $\langle p_1\rangle = \int p_1(t)\mathrm{d}t$.

13.7.6 Height of Maxima in the Frequency Response Curve

Let the level difference h between a sound pressure maximum and the next minimum be called height of a maximum. Its mean value $\overline{h}$ in a diffuse sound field is 12.5 dB for spatial averaging at a fixed frequency, and 7.8 dB for spectral averaging at a fixed position (because of the different "spectra"). For the latter case the influence of direct sound can be estimated as

$$\overline{h} \approx \frac{6.3\,\mathrm{dB}}{\left(0.4 + d^4\right)^{1/4}} \tag{13.48}$$

where d is the amplitude ratio of direct and diffuse sound.

13.7.7 Experiments

Measurements in the resiliently lined reverberation tank revealed that for sinusoidal excitation the average distance of maxima in front of a 20 cm wide absorber strip is larger, and the mean height of maxima smaller than in a diffuse sound field. But at 60 cm distance from the absorber the sound field is apparently diffuse. Further experiments were planned, but abandoned because of the end of the research contract with the British Navy.

13.8 Vibration and Sound Radiation of Flat Plates

A representative of the British Navy visited us every summer to discuss our past and future work within the research contract. By the end of the 1960s, we were asked to do some fundamental investigations about "how does the sound get out of the ship into the water." Since a ship hull is a rather thick curved steel plate, we decided to study the vibrational behavior and sound radiation from thick-walled cylindrical shells. We realized soon that it would be helpful to study beforehand the vibration and sound radiation of flat plates in air and water. Both are dominated by natural frequencies of the flexural resonances which can be calculated from the wave equation and the boundary conditions.

The particle velocity v of *thin* plate flexural vibrations obeys the fourth order partial differential equation

$$\frac{D}{\rho h}\Delta^2 v + \frac{\partial^2 v}{\partial t^2} = 0 \tag{13.49}$$

where $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$ is the Laplacian operator, $D = Eh^3/(12(1-\mu^2))$ the flexural rigidity, E Young's modulus, μ Poisson's ratio, h the plate thickness, and

ρ the mass density. For *thick* plates additional terms account for rotary inertia and shear deformation:

$$\frac{D}{\rho h}\Delta^2 v + \frac{\partial^2 v}{\partial t^2} - \left(\frac{h^2}{12} + \frac{D\gamma}{hG}\right)\frac{\partial^2}{\partial t^2}\Delta v + \frac{\rho h^2 \gamma}{12G}\frac{\partial^4 v}{\partial t^4} = 0 \tag{13.50}$$

where G is the shear modulus and $\gamma \approx 1.2$ the shear distribution constant [39, 40]. Assuming harmonic vibration, $v(x, y, t) = V(x, y)\cos \omega t$, (13.49) yields for the mode shapes $V(x, y)$ of thin plates

$$\frac{D}{\rho h}\Delta^2 V - \omega^2 V = 0, \tag{13.51}$$

and (13.50) for thick plates:

$$\frac{D}{\rho h}\Delta^2 V - \omega^2 V + \left(\frac{h^2}{12} + \frac{D\gamma}{hG}\right)\omega^2 \Delta V + \frac{\rho h^2 \gamma}{12G}\omega^4 V = 0. \tag{13.52}$$

Equation (13.52) can be solved rigorously by a superposition of sin, cos, sinh and cosh terms in x and y, the weights being determined by the boundary conditions. In order to calculate the natural frequencies of a plate with free edges and dimensions a and b in x- and y-direction, it is convenient to assume

$$V(x, y) \approx \hat{V}\,(\cos k_x x)\big(\cos k_y y\big) \tag{13.53}$$

with

$$k_x = \left(m - \frac{1}{3}\right)\frac{\pi}{a}, \qquad k_y = \left(n - \frac{1}{3}\right)\frac{\pi}{b} \tag{13.54}$$

where m and n are the number of nodes in x- and y-direction. The terms $-1/3$ account for the fact that the exponential near-fields at the edges shift the outer node lines to $\lambda/6$ distance from the edges if λ is the flexural wavelength in the respective direction. For $m = 0$ holds $k_x = 0$, and for $n = 0$: $k_y = 0$. $m = 1$ or $n = 1$ would mean rotations without restoring forces so that for $a \geq b$ the lowest vibrational mode is $(m, n) = (2, 0)$. Substituting (13.53) in (13.51) yields the frequency equation for thin plate modes

$$\omega_{mn}^2 = \frac{D}{\rho h}\left(k_x^4 + k_y^4\right), \tag{13.55}$$

and in (13.52) for the thick plate modes:

$$\omega_{mn}^4 - \omega_{mn}^2 \frac{12G}{\rho h^2 \gamma}\left(1 + \left(\frac{h^2}{12} + \frac{D\gamma}{hG}\right)\left(k_x^2 + k_y^2\right)\right) + \frac{12GD}{\rho^2 h^3 \gamma}\left(k_x^4 + k_y^4\right) = 0. \quad (13.56)$$

The natural frequency of the thick plate flexural vibration is the lower one of the real-valued solutions of (13.56).

Our test object was a flat steel plate with dimensions $a = 36$ cm, $b = 12$ cm and $h = 2$ cm, excited at the center by a longitudinally vibrating piezoelectric tube (10 mm diameter) with a damped countermass—a sandwich of lead plates and plastic layers so that its resonances were sufficiently damped and a constant driving voltage excited the plate with nearly constant force in the frequency range of interest (up to 35 kHz). To realize free edge conditions on all sides, the plate and the countermass were suspended from thin plastic threads to hang freely in the water. With sliding sine wave excitation, the frequency response curves of the plate in air and water were measured with a small accelerometer, mounted by a magnetic clamp at the plate center (opposite to the exciter), and recorded on a level recorder. The response curve for the plate in air showed sharp resonances up to 35 kHz, but in water only up to about 12 kHz. Due to the mass load and greater radiation loss, the resonance frequencies are shifted to lower values as compared to the results in air, and at higher frequencies the damping by sound radiation into water is so strong that no resonance peaks occur.

The resonance shift caused by the added mass $m_w = \rho_w h_w$ of water per unit area can be estimated by modifying (13.55):

$$\omega_{mn}^2 = \frac{D}{\rho h + m_w}\left(k_x^4 + k_y^4\right). \quad (13.57)$$

According to [41], m_w for an infinite plate with one-sided load at low frequencies is the mass of a water layer with thickness $h_w \approx 1/6$ of the flexural wavelength of the plate. Our measurements of the resonance frequencies in air and water showed, however, that—on the average over many modes—the mass load amounts to a water layer of only 1/10 of the flexural wavelength on both sides of the submerged plate, presumably because of the acoustic short circuit around the free edges of the plate. Calculated flexural wavelengths for the $(m, 0)$ modes agreed exactly to our measured values in air and water.

Vibration patterns of the plate were scanned by the accelerometer which could be moved motor-driven along the x-direction sliding on a grease film at any height (the y-coordinate). The vibration patterns of some modes were also visualized by holographic interferometry. Both methods showed that the node lines never intersect ("node lines avoid each other").

The far-field sound radiation pattern was recorded in the shallow water basin (Sect. 13.3.6). A hydrophone could be moved along a semicircle of radius 1.5 m with the aid of a motor-driven turning arm. The far-field radiation pattern $p(\Theta, \phi)$ and the vibration pattern $V(x, y)$ are a pair of Fourier transforms. With Θ the

azimuth and ϕ the elevation angle, $\alpha = k \sin \Theta$ and $\beta = k \sin \phi$ (k the wavenumber in the medium) holds

$$p(\alpha, \beta) = \frac{j\omega\rho}{2\pi R} \mathrm{e}^{-jkR} \int_{-a/2}^{+a/2} \int_{-b/2}^{+b/2} V(x, y) \mathrm{e}^{j\alpha x} \mathrm{e}^{j\beta y} \mathrm{d}x \mathrm{d}y. \tag{13.58}$$

$\Theta = \phi = 0$ is the direction of the plate normal. Confining to the horizontal plane, we find

$$p(\Theta) \propto \int_{-a/2}^{+a/2} V(x) \mathrm{e}^{jk \sin \Theta} \mathrm{d}x. \tag{13.59}$$

With the approximation $V(x) \propto \cos k_x x$ follows for the sound pressure magnitude in the far-field

$$|p(\Theta)| \propto \left| \frac{\sin [(a/2)(k_x - k \sin \Theta)]}{k_x - k \sin \Theta} + \frac{\sin [a/2(k_x + k \sin \Theta)]}{k_x + k \sin \Theta} \right| \tag{13.60}$$

which compared well with our experimental results, except for $\Theta \approx \pm 90°$.

The radiation efficiency s of flexurally vibrating plates (see (13.32), (13.33)) is strongly frequency dependent because of the dispersion of bending waves. The flexural wave velocity is proportional to $\sqrt{\omega}$ so that at low frequencies the acoustic short circuit along the plate surface reduces the power radiation. At the critical frequency f_c, where the flexural wave velocity equals the sound velocity in the surrounding medium, s runs through a maximum $s(f_c) > 1$, and at higher frequencies s approaches unity (see also the next Section). Many of our experimental results on sound radiation from plates and their comparison with theoretical predictions have been published in [42, 43], including the influence of rib-stiffeners.

13.9 Critical Frequency of Plates in Dense Media

Ignoring the influence of the surrounding medium on the plate bending wave velocity, the critical frequency for thin plates is

$$\omega_{c0} = c_0^2 \sqrt{\frac{\rho h}{D}} \tag{13.61}$$

where c_0 is the sound velocity in the medium, ρ, h, and D as in (13.49). But the high mass and radiation load at f_c leads to a splitting of the dispersion curve, the frequency dependence of the plate bending wave velocity $c_b(\omega)$ [44] so that c_b

never equals c_0. The separation is small for plates vibrating in air, but it becomes important in water. A stability criterion allows to determine f_c. We could show that for steel plates in water $\omega_c = 1.34\omega_{c0}$ and for aluminum plates $\omega_c = 1.52\omega_{c0}$. The deduction and illustrations have been published in [45].

13.10 Vibration of Cylindrical Shells

While flexural vibrations of flat plates are governed by fourth order partial differential equations (Sect. 13.8) thick-walled cylindrical shells obey tenth order equations ([46], a), and even the neglection of rotary inertia and shear deformation (as for thin-walled shells) leads to still eighth order equations ([46], b). If the added mass of the surrounding medium, e.g., water, is not negligible, the rigorous treatment demands for solving integral equations [47]. In order to reduce the computational effort we tried to develop simplified equations for the natural frequencies of flexural vibrations of thick-walled, finite-length cylindrical shells.

The natural modes are described by (m, n) where m is the number of node circles and n the number of node lines parallel to the cylinder axis along half the circumference. Let l be the cylinder length, r_a the outer radius, and h the thickness of wall. Assuming cosine-shaped particle velocity distribution, the circumferential wavenumber is

$$k_u = \frac{2\pi}{\lambda_u} = \frac{n}{r_a} \tag{13.62}$$

(λ_u circumferential wavelength). The axial wavenumber can be approximated by

$$k_a = \left(m - \frac{1}{3}\right)\frac{\pi}{l} \tag{13.63}$$

for the free-free cylinder (see also (13.54) for the plate with free edges). It is suitable to normalize the calculated natural frequencies $\omega_{mn} = 2\pi f_{mn}$ with ω_r', the frequency of the infinitely long cylinder in the mode $(0, 0)$, a uniform radial vibration ("breathing cylinder"):

$$\omega_r' = 2\pi f_r' = \frac{c_L}{r_0} \tag{13.64}$$

where $r_0 = r_a - h/2$ is the mean cylinder radius and $c_L = \sqrt{E/\rho(1-\mu^2)}$ the longitudinal wave velocity, E Young's modulus, ρ mass density, and μ Poisson's ratio. f_r' is the frequency at which the longitudinal wavelength equals the cylinder perimeter. Slightly lower than ω_r' is the "ring frequency" $\omega_r = 2\pi f_r = c_D/r_0$ with $c_D = \sqrt{E/\rho}$. f_r is important for sound radiation since in its vicinity resonances with high radiation efficiencies are clustered. In the following the normalized frequency

$$\Omega_{mn} = \frac{\omega_{mn}}{\omega_r'} \tag{13.65}$$

will be considered.

The theory of elasticity leads to different eighth order partial differential equations for thin-walled cylindrical shells, depending on the way of deduction ([46], b). Assuming

$$v(\varphi, z, t) = \hat{v}\,(\cos n\varphi)(\cos k_a z)(\cos \omega t) \tag{13.66}$$

for the particle velocity (φ in circumferential, z in axial direction) yields cubic algebraic equations for Ω_{mn}^2 ([46], a). The three solutions correspond to radial, axial, and circumferential vibrations, of which the radial vibration is relevant for sound radiation. With the assumption $\Omega \ll k_a r_0$ the cubic equations reduce to quadratic equations, as M. Heckl [48] has shown for the "simplified Kennard equation," ending up with

$$\Omega^2 = \left(1-\mu^2\right)\frac{x^4}{\left(x^2+n^2\right)^2} + \delta A \tag{13.67}$$

where $x = k_a r_0$, $\delta = h^2/12r_0^2$, and $A = (x^2+n^2)^2 - [n^2(4-\mu) - 2 - \mu]/(2-2\mu)$. Other cubic equations for Ω^2 were deduced by Flügge and Arnold/Warburton:

$$\Omega^6 + p\Omega^4 + q\Omega^2 + r = 0 \tag{13.68}$$

with (for the Flügge equation):

$$p_F = -\left[\frac{3-\mu}{2}\left(x^2+n^2\right) + 1 + \delta\left(x^2+n^2\right)^2\right], \tag{13.69}$$

$$q_F = \frac{1-\mu}{2}\left[\left(x^2+n^2\right)^2 + (3-2\mu)x^2 + n^2\right] + \delta\frac{3-\mu}{2}\left(x^2+n^2\right)^3,$$

$$r_F = -\frac{1-\mu}{2}\left[\left(1-\mu^2\right)x^4 + \delta A_F\right],$$

$$A_F = \left(x^2+n^2\right)^4 - 2\left[\mu x^6 + 3x^4n^2 + (4-\mu)x^2n^4 - (2-\mu)x^2n^2 + n^6\right] + n^4,$$

and for the Arnold/Warburton equation:

$$p_{\mathrm{AW}} = p_F - \delta\left[n^2 + 2(1-\mu)x^2\right], \tag{13.70}$$

$$q_{\mathrm{AW}} = q_F + \delta\left[2x^2(1-\mu)\left(x^2+1\right) - \left(2-\mu^2\right)x^2n^2 - \frac{3+\mu}{2}n^4 + n^2\right],$$

$$r_{\mathrm{AW}} = r_F - (1-\mu)x^2\delta\left[\mu\left(x^4 - n^4 + n^2\right) + x^2\left(1-\mu^2\right)\left(2-n^2\right)\right].$$

Applying Heckl's approximation $\Omega \ll k_a r_0$ to the Flügge equation, one obtains

$$\Omega^2 = \frac{(1-\mu^2)x^4 + \delta A_F}{\left(x^2+n^2\right)^2 + n^2.} \tag{13.71}$$

In an attempt to account for the influence of rotary inertia and shear deformation, we had the idea to apply the relation of thick and thin plates. Imagine the cylinder cut open along one axial node line and developed into a plane, then you get the "equivalent plate" of the cylinder. Its edge conditions are "simply supported" at the new edges, and free at the former cylinder top and bottom. Let Ω_{C0} be the frequency of the thin-walled cylinder (calculated from one of the aforementioned theories), Ω_{P0} that of the thin plate, and $\Omega_{\mathrm{P}h}$ that of the thick plate, then we proposed to calculate the frequency $\Omega_{\mathrm{C}h}$ of the thick-walled cylinder by

$$\Omega_{\mathrm{C}h} = \Omega_{\mathrm{C0}} \frac{\Omega_{\mathrm{P}h}}{\Omega_{\mathrm{P0}}} \tag{13.72}$$

where Ω_{P0} and $\Omega_{\mathrm{P}h}$ are obtained from (13.55) and (13.56).

In order to check this suggestion and to compare the different theoretical approaches for the cylinder eigenfrequencies, we performed careful measurements in air and in the deep water tank with two cylinder shells: Cylinder I with $l = 40$ cm, $r_0 = 20$ cm, $h = 1.7$ cm, and Cylinder II with $l = 40$ cm, $r_0 = 11$ cm, $h = 2.4$ cm. Both cylinders had free edges. Cyl. II was excited by a point force (piezo transducer with countermass) at the upper edge from inside, Cyl. I at half height to suppress modes with odd m (otherwise the resonances would overlap too much). The resonance frequencies could be read from the maxima of the admittance frequency response, the mode shapes were determined by scanning the particle velocity along the outer surface with a motor-driven accelerometer. For the measurements in water, the end faces of the cylinders were sealed with plastic foils, so that the cylinders were air filled, and water loaded only at the outside.

We could identify in air 26 modes of Cyl. I and 15 of Cyl. II between 800 and 8,300 Hz. The reference frequencies f_r' were 4.3 kHz for Cyl. I and 7.8 kHz for Cyl. II. The measured resonance frequencies were compared to those calculated from the cubic equations without and with Heckl's approximation, and without and with the thickness correction (13.72). The best results were obtained with Heckl's approximation of the Flügge equation and the thickness correction ((13.71) and (13.72)): mean error 0.1 %, standard deviation ±1.9 %. This is better than could be expected. Probably some of the approximation errors cancel each other. The complete list of results, estimation of the added mass in water, diagrams of the natural frequencies as functions of m and n have been published in [49]. Theory and experiments on sound radiation from the two cylinders are described in [50], and for the cylinders with ring stiffeners in [51].

13.11 Miscellaneous

13.11.1 Viscometer

During the development of flow absorbers (Figs. 13.3, 13.4, and 13.5) it turned out to be advantageous to measure the viscosity of the fluids and the impedance of thin fluid layers. Our student Joachim Richter developed experimental facilities and the associated theory. A fluid volume of less than 10 mm^3 was brought between two circular PZT (or barium titanate) transducers of 8 mm diameter, the distance of which could be adjusted by a precision thread to an accuracy of about 1 µm. The apparatus was predominantly suited for Newtonian fluids (without elasticity), but also for viscoelastic fluids. The theory, construction and experimental results have been published in [52] for the viscometer and in [53] for the impedance measurements.

13.11.2 Self-Imaging

Self-imaging is a phenomenon occurring in dispersive waveguides, e.g., in optical fibers [54] and the deep sea SOFAR channel [55]. Because each mode in the waveguide propagates with its own phase velocity, the modes soon become dephased and an object in the entrance plane cannot be reconstructed, unless after a certain distance the accumulated phase differences between any two modes are approximately multiples of 2π and so produce a "self-image." Since self-imaging occurs at different distances for different frequencies, it is possible with monofrequent sound fields only.

Our student Gero Timann proved theoretically that self-imaging exists in any waveguide with discrete phase velocities of the guided modes and confirmed this experimentally in the resiliently lined water duct (Sect. 13.3.4). He proved also that self-imaging is possible over long distances. However, because a waveguide is a spatial low-pass filter, the resolution of the images is smaller than that of the original [56].

13.11.3 Schroeder Diffusers

One of Manfred Schroeder's great contributions to room acoustics was the development of diffusively reflecting wall structures, based on number theoretical principles. I remember the origin. When Manfred Schroeder returned to Göttingen he had to give a four-semester lecture course on the Institute's research topics, in the summer semester 1971 on the fundamentals "Physics of vibration." I was Erwin Meyer's co-author for a book on this subject [57], and Schroeder—as usual during the German university vacation in the USA—asked me to send him the manuscript.

He thanked me writing "this was exactly what the doctor has prescribed." He found a short chapter on Barker sequences and maximum length sequences which were new to him (both have "thumb tack" autocorrelation functions). Back in Göttingen, he asked me if I had thought about the spatial equivalent of these time sequences, and I had to confess that I did not. But he had immediately realized that such structures could solve the problem of diffuse sound reflection. He engaged students to start experiments, and, typical for him, asked himself how to optimize such sequences. He found the solution in number theory as is well known [58].

The success in room acoustics inspired our hydroacoustics group to apply Schroeder diffusers (or, as he wrote, diffusors) to underwater sound, too. We thought that diffuse reflectors might be of some interest for underwater structures, e.g., to prevent sonar localization. Since rigid reflectors are not readily available for underwater sound, we made experiments with sound-soft structures in the shallow water basin (Sect. 13.3.6), applying quadratic residue sequences. We encountered problems for oblique incidence since separations between adjacent diffuser elements are difficult to realize with sound-soft material, but for rather vertical incidence they showed a wide scatter in the directivity patterns [59].

13.11.4 Resonance Scattering

By the end of 1986, a representative of the German BWB: "Bundesamt für Wehrtechnik und Beschaffung" (Federal Office of Defense Technology and Procurement) offered us a research contract on sonar classification of submerged objects. We learnt that thousands of undiscovered mines are still buried in the North Sea and the Baltic Sea at unknown locations, presenting a potential danger for ships leaving the normal routes. Scanning the sea bottom with sonar yields on the sonar screen many "mine-like objects," the majority of which are stones. We learnt also that a few very experienced sonar operators listening to the acoustic echoes can discriminate metallic objects from stones with high reliability, but they are unable to tell how they do it. This shows that the information about the nature of a sound scattering object is contained in the echo, but in a not very obvious way. This encouraged us to apply artificial neural networks as discriminators.

In analogy to airborne acoustics, where a metallic object "sounds" different than a stone when being hit, a sonar impulse exciting eigenfrequencies causes a specific ringing of the object. In the same manner as atomic particles can be characterized by the frequencies and halfwidths of their scattering resonances, macroscopic underwater bodies should theoretically be identifiable from the line spectra of their backscattered echoes.

Experiments with the two steel cylinders introduced in Sect. 13.10, the flat steel plate of Sect. 13.8, a high-ρc ceramic rod, and several stones showed resonance backscatter by more or less pronounced ringing in the time domain, and resonance peaks in the spectrograms.

Initial experiments were made in our deep water tank. Test signals were rectangular or Gaussian pulses of a few ms duration at frequencies around 100 kHz. Later on, test facilities of the German Navy on a lake in Northern Germany could be used. The test objects and transducers were suspended from a raft to a depth of 10 m below the water surface (depth of the lake 40 m). It turned out soon that isolated resonances could be identified at low frequencies only, while at higher frequencies the resonances strongly overlap. In order to distinguish four different objects (big cylinder, small cylinder, stone, and "no object," i.e., the direct sound only) we applied an artificial neural network, a conventional perceptron with 16 input channels, one hidden layer with 3 neurons, and two output neurons. After training the network with echoes at 4 aspect angles, the four targets were reliably identified for other angles, and also new objects (cylinder/stone) were discriminated with 80–90 % reliability.

Computer simulations showed that a neural network can much better discriminate between slightly different signals if time series data are offered instead of power spectra. The reason is the loss of information on trans-spectral coherence by the transformation to the power spectrum.

More details on theory and experiments have been published in [60], and a resonance scattering theory for strongly overlapping resonances in [61].

13.11.5 Electrorheological Fluids

In the mid 1990s, the BWB (see Sect. 13.11.4) asked us to do some research on the application of electrorheological fluids (ERF) to absorbers for underwater sound. Suspensions of polarizable particles in insulating fluids show the effect that the application of an electric field changes the viscosity of the mixture from liquid towards solid or gel. The electric field polarizes the individual particles, and their electrostatic attraction forms particle chains between the electrodes, thus enhancing the flow resistance. Standard applications of ERF were clutches, low frequency vibration dampers, engine mounts, and electro-hydraulic devices. Acoustic applications were not yet known.

We started experiments to study the ERF properties at acoustic frequencies with an appropriately constructed impedance tube. Sound velocity and attenuation of ERFs were measured as functions of frequency, field strength, and temperature. Further, a laboratory model of a squeeze film absorber (Fig. 13.3) for underwater sound was built and tested. Within certain limits, the absorber impedance could be controlled by the electric field strength applied to the ERF. In another construction, the field strength and the gap width were adjusted under computer control. A genetic algorithm was applied to adaptively minimize the reflection coefficient. Convergence to $|\underline{r}| \approx 0.13$ was reached after about 20 "generations" of the genetic algorithm. Details and experimental results have been published in [62] and [63], including a theoretical model based on the Biot theory for fluid-saturated marine sediments.

13.12 Résumé

Summarizing, I have experienced a very fruitful period of acoustic research under the Institute's directors Erwin Meyer, Manfred Schroeder, and finally (since 1994) Werner Lauterborn. I enjoyed stimulating discussions with all of them and all other colleagues on topical problems, and I estimated also the freedom to select research projects of my own preference, particularly during the "Schroeder era." Our Institute was highly esteemed during Meyer's leadership, and even more under Manfred Schroeder's guidance. At an ASA meeting in the mid 1990s, the chairman announced my talk saying "Dieter comes from the *legendary* Third Physical Institute in Göttingen." The high reputation of our Institute made it also attractive to excellent students, and we benefitted from their skill in theoretical and computer work, in my research group not only on underwater acoustics, but also in the field of active noise and vibration control (ANVC) which we started upon Manfred Schroeder's suggestion when the British Navy contract ended in 1978. Our research group became known as the first one propagating active impedance control both for sound and vibration. I have given an overview of the whole field of ANVC in ([2], 5th Chapter, pp. 107–138). Readers may also be interested in my biography of Erwin Meyer [64].

(If any reader of this text is interested in an English translation of one of the originally German publications, please contact me at Dieter.Guicking@phys.uni-goettingen.de.)

References

1. Meyer, E. (ed), Kuhl, W., Oberst, H., Skudrzyk, E., Tamm, K.: Sound Absorption and Sound Absorbers in Water (Dynamic Properties of Rubber and Rubberlike Substances in the Acoustic Frequency Region). Translated by Charles E. Mongan Jr., Department of the Navy, Bureau of Ships, Washington, DC, Report NavShips 900,164, Vol. 1. Manuscript delivered 1946, published 1950 (XVIII + 519 pp.) (1950)
2. Kurz, T., Parlitz, U., Kaatze, U. (eds.): Oscillations, Waves, and Interactions. Sixty Years Drittes Physikalisches Institut. A Festschrift. Universitätsverlag Göttingen 2007. ISBN: 978-3-938616-96-3. Vii + 474 pp. http://www.webdoc.sub.gwdg.de/univerlag/2007/oscillations.pdf (2007)
3. TV film "Vom Jäger zum Gejagten. Das erste Stealth-U-Boot der Welt" ("From the hunter to the hunted. The first stealth U-boat of the world") transmitted by *arte* on 18 February 2008, 21:00–21:50. http://de.wikipedia.org/wiki/U_480, http://www.uboat.net/boats/U480.htm (2008)
4. Meyer, E., Brendel, K.: Reflexionsarme Dünnschichtabsorber für Wasserschall. In: Sixth International Congress on Acoustics (ICA), August 21–28, 1968. Proceedings: Paper L-3-1, pp. L-65–L-68. Tokyo, JP (1968)
5. Meyer, E., Brendel, K., Richter, J.: Absorption von Wasserschall durch erzwungene Wechselströmung von Flüssigkeiten. Acustica **19**, 8–20 (1967/1968)
6. Brendel, K., Leuschner, H.-W.: Wasserschall-Resonanzabsorber mit Reibungsverlusten in Flüssigkeitsschichten. Acustica **21**, 260–268 (1969)
7. Leuschner, H.-W.: Ölgedämpfter Resonanzabsorber für Wasserschall. Acustica **25**, 280–288 (1971)

8. Ahlswede, Th., Guicking, D.: Breitband-Wasserschallabsorber geringer Schichtdicke. In: Fortschritte der Akustik, DAGA'75, pp. 663–666. Physik-Verlag, Weinheim (1975)
9. Ahlswede, T.: Breitband-Wasserschallabsorber geringer Schichtdicke. Dissertation (Ph.D. Thesis) Göttingen (1974)
10. Meyer, E., Schilz, W., Tamm, K.: Über den Bau eines reflexionsfreien Wasserschall-Meßbeckens. Acustica **10**(AB 1), 281–287 (1960)
11. Meyer, E., Tamm, K.: Breitbandabsorber für Flüssigkeitsschall. Acustica **1/2**, AB 91–AB 104 (1951/1952)
12. Wille, P.: Experimentelle Untersuchungen zur Schallstreuung an schallweichen Objekten. Acustica **15**, 11–25 (1965)
13. Richter, J., Leuschner, H.-W., Ahlswede, T.: Druckkammerverfahren zur Messung von akustischen Impedanzen in Wasser. Acustica **28**, 90–97 (1973)
14. Guicking, D., Karcher, K.: Hydroacoustic impedance measurement of large-area samples in the frequency range up to 200 kHz. Acustica **54**, 200–208 (1984)
15. Lubman, D.: Reverberation measurements of narrow-band noise power in a water tank. J. Acoust. Soc. Am. **37**, 1209 (1965)
16. Watters, B.C.: A reverberant water tank for underwater measurements. J. Acoust. Soc. Am. **53**, 357 (1972)
17. Blake, W.K., Maga, L.J.: Chamber for reverberant acoustic power measurements in air and in water. J. Acoust. Soc. Am. **57**, 380–384 (1975)
18. Guicking, D., Wetterling, Th.: A Reverberation Tank for Hydroacoustic Measurements. In: Eighth International Congress on Acoustics (ICA). Proceedings: Contributed Papers, vol. II, p. 482. London (1974)
19. Wetterling, Th.: Hydroakustische Messungen in einem schallweich begrenzten Halltank. Dissertation (Ph.D. Thesis) Göttingen (July 1975)
20. Schroeder, M[anfred R.]: Eigenfrequenzstatistik und Anregungsstatistik in Räumen. Modellversuche mit elektrischen Wellen. Acustica 4(AB 1) (1954) 456–468. English translation: Normal Frequency and Excitation Statistics in Rooms: Model Experiments with Electric Waves. J. Audio Eng. Soc. **35**, 307–316 (1987)
21. Schroeder, M[anfred R.]: Die statistischen Parameter der Frequenzkurven von großen Räumen. Acustica 4(AB 2) (1954) 594–600. English translation: Statistical Parameters of the Frequency Response Curves of Large Rooms. J. Audio Eng. Soc. **35**, 299–306 (1987)
22. Schroeder, M.R.: Frequency-correlation functions of frequency responses in rooms. J. Acoust. Soc. Am. **34**, 1819–1823 (1962)
23. Schroeder, M.R.: Effect of frequency and space averaging on the transmission responses of multimode media. J. Acoust. Soc. Am. **46**, 277–283 (1969)
24. Lubman, D.: Spatial averaging in a diffuse sound field. J. Acoust. Soc. Am. **46**, 532–534 (1969)
25. Cremer, L., Müller, H.L.: Die wissenschaftlichen Grundlagen der Raumakustik. Band 2, 2. Aufl. S. Hirzel Verlag, Stuttgart (1976)
26. Rollwage, M., Ebeling, K.J., Guicking, D.: Experimental investigation of two-dimensional diffuse sound fields in a shallow water basin. Acustica **58**, 149–161 (1985)
27. Meyer, E., Neumann, E.-G.: Physical and Applied Acoustics. An Introduction. Academic, New York, London (1972)
28. Guicking, D., Voronovich, A.: Theoretical evaluation of the edge effect of an absorbing strip on a pressure-release boundary. Acustica **70**, 66–75 (1990)
29. Zogalla, P., Guicking, D.: Untersuchungen zum Absorber-Kanteneffekt im schallweich berandeten Halltank. In: Fortschritte der Akustik, DAGA'92, pp. 269–272. DPG GmbH, Bad Honnef (1992)
30. Meyer, E., Just, P.: Messung der Gesamtenergie von Schallquellen. Zeitschrift für technische Physik **10**, 309–316 (1929)
31. Gösele, W.: Schallabstrahlung von Platten, die zu Biegeschwingungen angeregt sind. Acustica **3**, 243–248 (1953)

32. Schroeder, M.R.: Methoden zur Messung der Diffusität in Hallräumen. Acustica **9**(AB 1), 256–264 (1959)
33. Furduev, V.V., Tung, C.: Measurement of diffuseness of the acoustic field in rooms by the directional microphone method. Sov. Phys. Acoust. **6**, 103–111 (1960)
34. Reichardt, W., Sarkov, N.: Definition und Messung von Hall-Diffusitätsgraden. Hochfrequenztechnik und Elektroakustik **80**, 80–85 (1971)
35. Guicking, D., Lutze, R., Schroeder, M.R.: Messungen zur Diffusität im Halltank. In: Fortschritte der Akustik, DAGA'78, pp. 309–312. VDE-Verlag, Berlin (1978)
36. Rice, S.O.: Mathematical analysis of random noise. In: Wax, N. (ed.) Selected Papers on Noise and Stochastic Processes. Dover, New York (1954)
37. Dämmig, P.: Zur Messung der Diffusität von Schallfeldern durch Korrelation. Acustica **7**, 387–398 (1957)
38. Koyasu, M., Yamashita, M.: Evaluation of the degree of diffuseness in reverberation rooms by cross correlation technique. Bull. Kobayasi Inst. Phys. Res. **18**, 1–10 (1976)
39. Mindlin, R.D.: Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates. J. Appl. Mech. **18**, 31–38 (1951)
40. Leissa, A.W.: Vibration of Plates. NASA SP-160, US Government Printing Office, Washington, DC (1969)
41. Morse, P.M., Ingard, K.U.: Linear acoustic theory. In: Flügge, S. (ed.) Handbuch der Physik, Vol. XI/1, Akustik 1. Springer, Berlin (1961)
42. Guicking, D., Warhonowicz, T., Buddruss, C.: Schallabstrahlung von ebenen, zu Biegeschwingungen angeregten Platten in Wasser. Acustica **23**, 131–141 (1970)
43. Teubner, V.: Wasserschallabstrahlung von Platten mit und ohne Versteifungen. Acustica **31**, 203–215 (1974)
44. Kurtze, G., Bolt, R.H.: On the interaction between plate bending waves and their radiation load. Acustica **9**(AB 1), 238–242 (1959)
45. Guicking, D., Boisch, R.: Zur Grenzfrequenz ebener Platten in dichten Medien. Acustica **44**, 41–45 (1980)
46. Leissa, A.W.: Vibration of Shells. NASA SP-288, US Government Printing Office. a) Chapter 3.5.2, b) Chapter 2.1, c) Chapter 2.3 (1973)
47. Sandman, B.E.: Fluid-loading influence coefficients for a finite cylindrical shell. J. Acoust. Soc. Am. **60**, 1256–1264 (1976)
48. Heckl, M.: Vibrations of point-driven cylindrical shells. J. Acoust. Soc. Am. **34**, 1553–1557 (1962)
49. Guicking, D., Boisch, R.: Vereinfachte Berechnung der Eigenfrequenzen dickwandiger Zylinder in Luft und Wasser. Acustica **42**, 89–97 (1979)
50. Boisch, R., Guicking, D.: Schallabstrahlung von dickwandigen Stahlzylindern in Wasser. Acustica **45**, 322–339 (1980)
51. Boisch, R., Guicking, D.: Schallabstrahlung von ringversteiften Stahlzylindern in Wasser. Acustica **51**, 29–43 (1982)
52. Richter, J.: Viskosimeter für sehr geringe Flüsssigkeitsmengen. Rheol. Acta **9**, 550–553 (1970)
53. Richter, J.: Mechanische Impedanzen dünner Flüssigkeitsschichten. Acustica **22**, 95–106 (1969/1970)
54. Friesem, A.A., Levy, U., Silberberg, Y.: Parallel transmission of images through single optical fibers. Proc. IEEE **71**, 208–221 (1983)
55. Semenov, A.T.: Transmission of acoustic images in natural ocean waveguides. Sov. Phys. Acoust. **27**, 171–175 (1981)
56. Timann, G., Guicking, D.: Acoustic Self-Imaging in Resiliently Lined Waveguides. In: Ermert, H., Harjes, H.-P. (eds.) Nineteenth International Symposium on Acoustical Imaging, Bochum, April 3–5, 1991. Proceedings: Acoustic Imaging, vol. 19, pp. 149–154. Plenum Press, New York (1992)
57. Meyer, E., Guicking, D.: Schwingungslehre. ISBN 3-528-08254-2. Friedr. Vieweg + Sohn, Braunschweig (1974)

58. Schroeder, M.: Number Theory in Science and Communication. With Applications in Cryptography, Physics, Digital Information, Computing, and Self-Similarity, 5th ed. ISBN 978-3-540-85297-1. Springer, Berlin (1984, 1986, 1997, 2006, 2009)
59. Wicker, K., Guicking, D.: Diffus reflektierende Oberflächen im Wasserschall. In: Fortschritte der Akustik, DAGA 94. Teil B, pp. 837–840. DPG GmbH, Bad Honnef (1994)
60. Guicking, D., Görk, K., Peine, H.: Classification of underwater objects using resonance scattering and neural networks. In: Weydert M. (ed.) Proceedings of the European Conference on Underwater Acoustics, September 14–18, pp. 567–570. Luxembourg (1992)
61. Peine, H., Guicking, D.: Acoustical resonance scattering theory for strongly overlapping resonances. Acta Acustica **3**, 233–241 (1995)
62. Guicking, D., Wicker, K., Eberius, C.: Electrorheological fluids as an electrically controllable medium: I. Experimental arrangement and application to an absorber of underwater sound. Acustica **88**, 886–895 (2002)
63. Wicker, K., Guicking, D.: Electrorheological fluids as an electrically controllable medium: II. Medium properties and description by the Biot theory. Acta Acustica **88**, 896–903 (2002)
64. Guicking, D.: Erwin Meyer—A Great German Acoustician. Biographical Notes. http://www.dpi.physik.uni-goettingen.de/guicking/Erwin-Meyer-Eng.pdf

***

Biography



Dieter Guicking, born 1935, studied physics in Göttingen, Germany, and joined the "Drittes Physikalisches Institut" in 1959, starting with experiments on dynamic properties of high polymers. Since 1966 staff member, 1971–1998 in tenured position. Research fields: underwater acoustics, models of biological circadian rhythms, and active control of sound and vibration.

Part II
Manfred Schroeder's Memoirs



Fig. 1 Baby, 1 year old, summer 1927



Fig. 2 1929 with mother Hertha



Fig. 3 As a child with father Karl



Fig. 4 With parents and sisters Ingrid (*left*) and Helga (*right*) at age 16 or 17

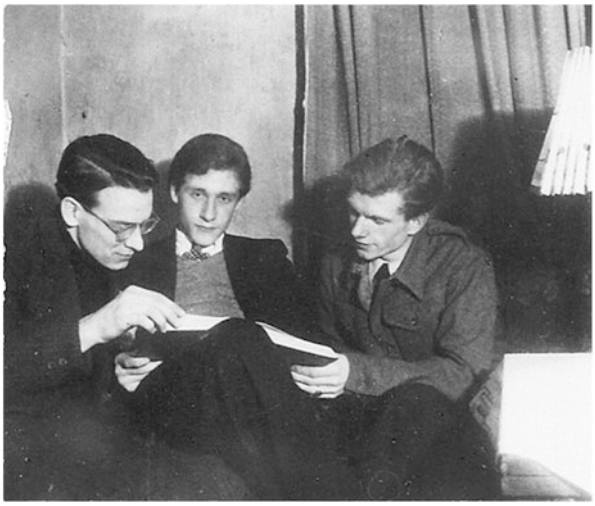


Fig. 5 As a student in Göttingen with Erhard Scheibe and Jobst von Behr



Fig. 6 Building a radio as a student

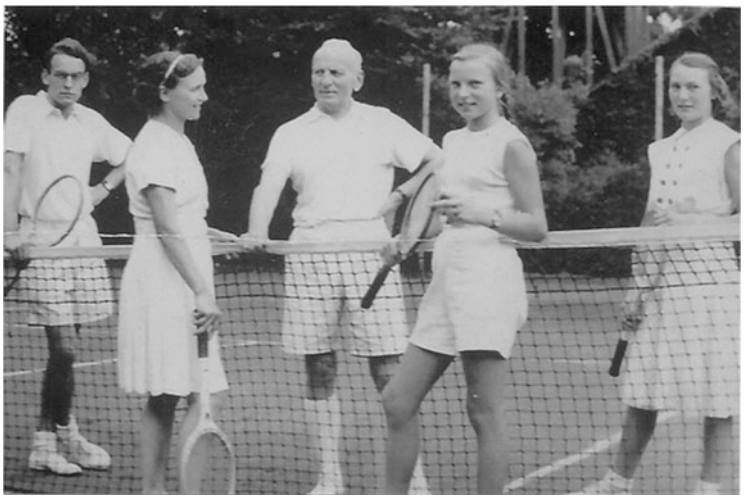


Fig. 7 With parents and Helga and Ingrid (sisters) as student visiting in Hamm



Fig. 8 Immigrating to America on the Andrea Doria, 1954



Fig. 9 First good camera, Leica, bought 1954



Fig. 10 First date with Anny, October 3rd 1954



Fig. 11 Wedding, February 25th 1956, Anny and Manfred



Fig. 12 First car 1957, Plymouth, purchased in 1954



Fig. 13 With Anny and children, Marion, Alexander, and Julian, 1964



Fig. 14 Gillette, New Jersey with wife and three kids: Alexander, Julian, and Marion, 1966



Fig. 15 Young Manfred



Fig. 16 Magazine cover, 1960

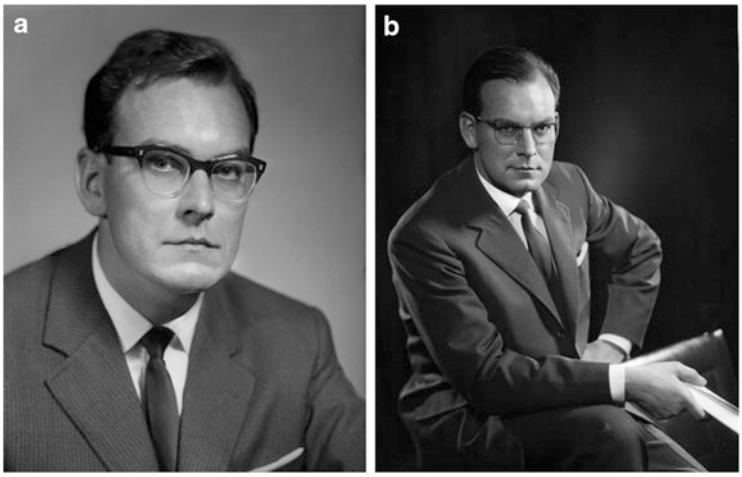


Fig. 17 (**a**, **b**) Manfred in Bell Labs



Fig. 18 Concert hall acoustics research

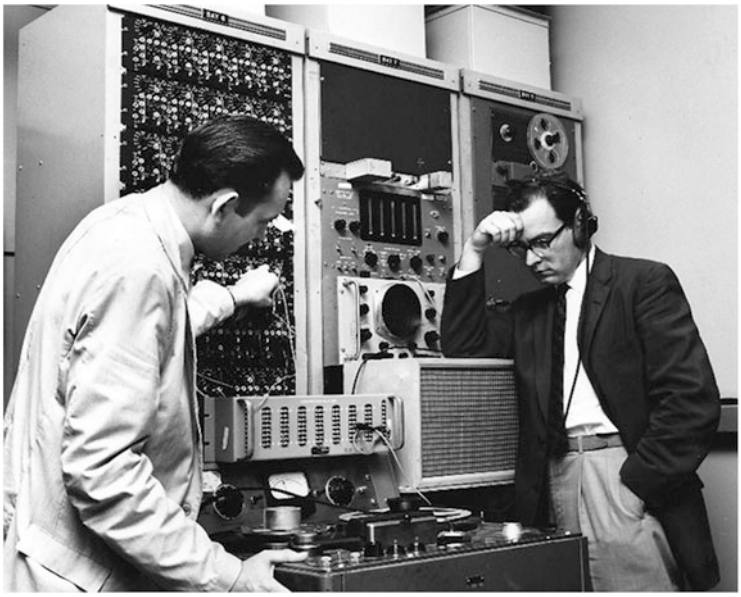


Fig. 19 Manfred with Ben Logan in Bell Labs (invention of the all-pass reverberator)



Fig. 20 Manfred with McLean in Bell Labs (invention of the anti-feedback system)

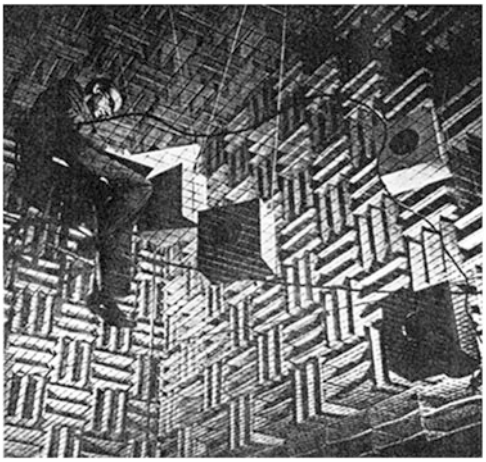


Fig. 21 Manfred in Bell Labs anechoic chamber (simulation of concert halls)



Fig. 22 Bicycling with Anny on Nantucket (1981)

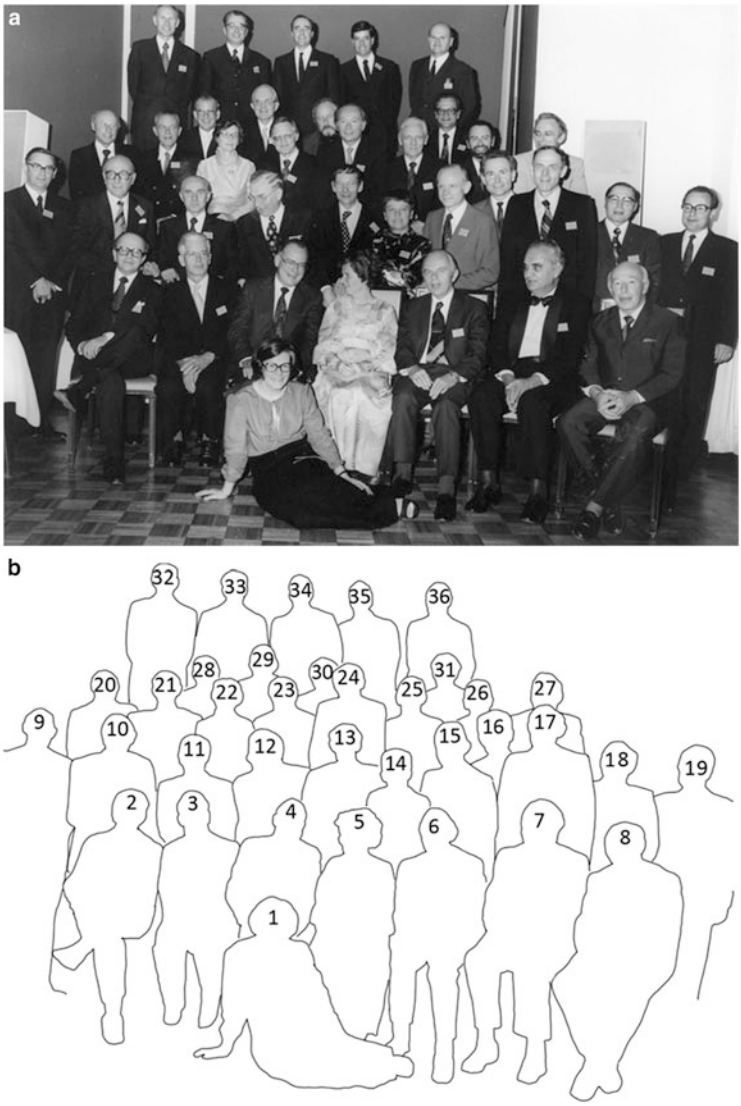


Fig. 23 (**a**) Göttingen 1977: 30th Anniversary of the Third Physics Institute. (**b**) *1*. Inge Schmid, geb. Laskowski, *2*. Horst Haeske, *3*. Horst-Gunther Diestel, *4*. Manfred R. Schroeder, *5*. Edith Kuhfuß, *6*. Günther Kurtze, *7*. Hans Severin, *8*. Konrad Tamm, *9*. Klaus Brendel, *10*. Clemens Starke, *11*. Peter Dämmig, *12*. Rolf Thiele, *13*. Georg-Renatus Schodder, *14*. Lucie Hackenbroich, geb. Bursian, *15*. Wolfgang Westphal, *16*. Helmut Siegert, *17*. Gerhard Sessler, *18*. Heinrich Henze, *19*. Reinhard Pottel, *20*. Karl Werner, *21*. Joachim Malzfeldt, *22*. Marie-Luise Beyer, geb. Exner, *23*. Klaus Helbig, *24*. Hans-Jürgen Schmitt, *25*. Peter-Paul Heusinger, *26*. Eberhard Mundry, *27*. Ernst-August Hampe, *28*. Wolfgang Burgtorf, *29*. Walter Kuhl, *30*. Wilhelm Ebrecht, *31*. Heinrich Kuttruff, *32*. Hans-Wilhelm Helberg, *33*. Wolfgang Eisenmenger, *34*. Walther Junius, *35*. Jörg Schmid, *36*. Frieder Eggers



Fig. 24 With secretary Mrs. Edith Kuhfuß in Göttingen, Drittes Physikalisches Institut

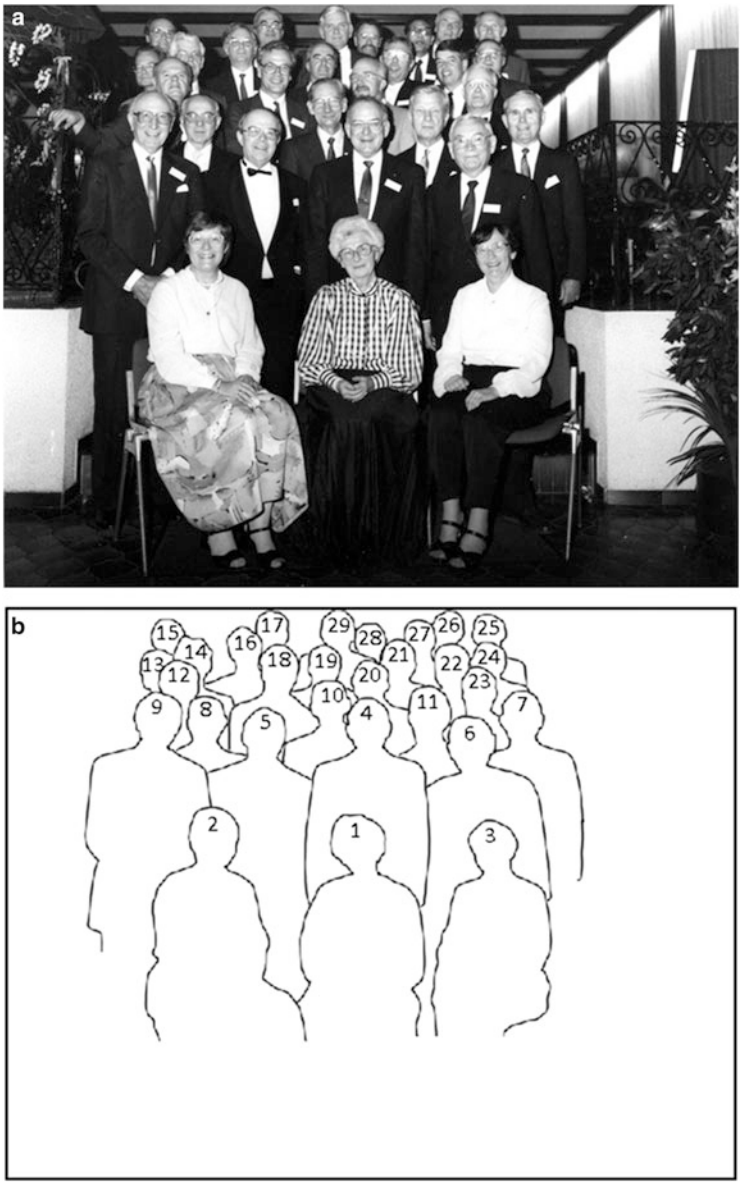


Fig. 25 (**a**) Göttingen 1987: 40th Anniversary of the Third Physics Institute with all people who joined the Institute in the first 10 years. (**b**) *1*. Edith Kuhfuß, *2*. Inge Schmid, geb. Laskowski, *3*. Marie-Luise Beyer, geb. Exner, *4*. Manfred R. Schroeder, *5*. Horst Haeske, *6*. Rolf Thiele, *10*. Georg-Renatus Schodder, *12*. Frieder Eggers, *13*. Reinhard Pottel, *14*. Horst-Gunther Diestel, *16*. Klaus Brendel, *17*. Wolfgang Eisenmenger, *18*. Heinrich Kuttruff, *22*. Jörg Schmid, *25*. Hans-Wilhelm Helberg, *26*. Gerhard Sessler. The following list was not very certain by the editors: *8*. Peter Dämmig, *9*. Clemens Starke, *19*. Walther Junius, *24*. Heinrich Henze *27*. Sanat Kumar Mukherjee, *28*. Eberhard Mundry, *29*. Friedrich Wiekhorst



Fig. 26 Göttingen, October 11th 1997: 50th Anniversary of the Third Physics Institute

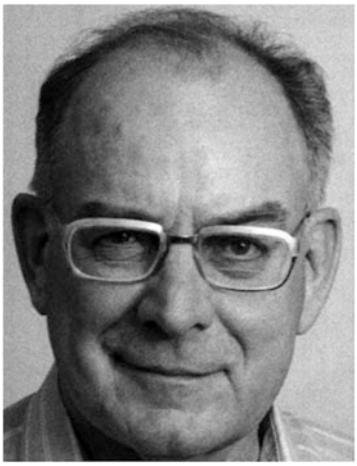


Fig. 27 *1991: The frontispiece in page vii is derived from this photo using Eikonal equation algorithm by Wolfgang Moeller*



Fig. 28 *ICA Congress 1983 Paris (from left: Anny, Manfred, Gerhard and Renate Sessler, Mildred and Jim Flanagan)*

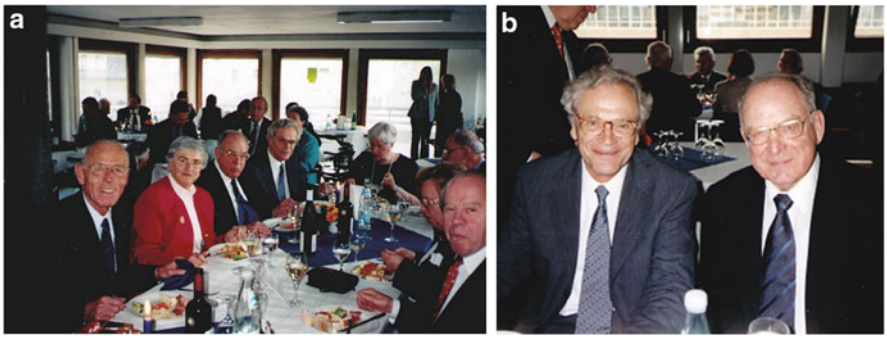


Fig. 29 (**a**, **b**) At *Darmstadt University of Technology 1998*



Fig. 30 With Ning Xiang and his daughter Charlotte at the 75th Anniversary of the Acoustical Society of America, New York City 2004



Fig. 31 *At the ASA Meeting Banquet, Honolulu 2006 (from left: Gerhard Sessler, Renate Sessler, Anny Schroeder, Ingrid Kuttruff, Manfred, Heinrich Kuttruff)*



Fig. 32 Family get together December 2007, from *left* to *right* standing: Nicola, Julian, Marion, Uwe, Alexander, Marion, Manfred, and Anny; kneeling: Nora, Lilly, Shadow, and Julia



Fig. 33 Manfred's hobby Photography (on the board: the Old Man from the Mountain, courtesy: A. Kohlrausch)



Fig. 34 With Heinrich Kuttruff



Fig. 35 With Armin Kohlrausch at the Honoring session for Manfred sponsored by the Acoustical Society of America Meeting, 2006 (Board: Freihalten für Anlieferungen)—courtesy by Omoto



Fig. 36 With Hiroya Fujisaki at the Honoring session for Manfred sponsored by the Acoustical Society of America Meeting, 2006—courtesy by Omoto



Fig. 37 Brief remarks at the end of the Honoring session for Manfred: with high appreciation and pleasure, sponsored by the Acoustical Society of America Meeting, 2006—courtesy by Omoto



Fig. 38 Honoring Manfred 2006 at the ASA Meeting Hawaii jointed with the Acoustical Society of Japan, with all invited speakers and chair (from *left*: Ning Xiang (chair), Bishnu Atal, James West, Gerhard Sessler, Volker Mellert, Armin Kohlrausch, Manfred Schroeder, Heinirch Kuttruff, Anny Schroeder, Jean-Dominique Polack, Michael Vorlaender, Fumitada Itakura, James Flanagan, Birger Kollmeier, Hiroya Fujisaki)—courtesy by Omoto



Fig. 39 In Memory of Manfred 2011 at the Acoustical Society of America meeting in Seattle, WA (from *left*: Birger Kollmeier, Joshua Atkins, Julian Schroeder, James West, Peter D'Antonio, Alexander Schroeder, Anny Schroeder, Jean-Dominique Polack, Ning Xiang, Roland Kruse, Gerhard Sessler, Peter Cariani, Bishnu Atal)—courtesy by the ASA

Chapter 14
Right Turns

Manfred R. Schroeder

In the course of my lifetime, I have made many bad decisions and a few good ones; some without further consequences, others with lasting results. Here I want to focus on the good decisions—the Right Turns—that have had major impacts on my life.

14.1 Pulling Little Roots

Even as a child I was never much interested in sports. When the neighborhood boys rang the doorbell for me to join them in *Schlagball* (Soft ball), my mother said "Manfred is in his room 'pulling little roots'" (*Würzelchen ziehen*), her metaphor for solving math problems. To this day, for better or worse, I have never been interested in participating in competitive sports. (But I do like to watch tennis, downhill skiing, skating, and ballroom dancing, activities that I actually practiced when I was younger.)

In my entire life, I rarely did things I didn't like and that, I think, is good advice for a happy life. For example, I abhorred chemistry and never studied it as I was actually required to do as a physics student. Later in life, as an institute director, I had to supervise many theses involving a heavy dose of (physical) chemistry. So I got hold of some text books and read up on the chemistry I needed to help other people in need.

Author was deceased at the time of publication.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_14,
© Springer International Publishing Switzerland 2015

14.2 Air Force to Navy

A potentially life-saving Right Turn I took in September 1944, 2 months after I had turned 18. Just before reporting to my Air Force radar unit in Detmold for boot camp, I had cut my right toe on a glass shard during my last civilian swim. I therefore missed boot camp. The Air Force, not wanting to bother with a half-invalid, packed me off to the Navy. (The Navy and the Air Force had an understanding to exchange technical personnel if and when the need arose.) I was not amused and my father, a lifelong flying enthusiast (and combat pilot in World War I) was incensed. He was about to call the Air Force personnel office in Berlin, where he had some old friends, to have my transfer rescinded. But I thought "Why bother—maybe I'll be better off in the Navy." I prevailed and on or about September 30th 1944, I reported to the corresponding Navy radar unit in Stralsund—which was just concluding its own boot camp. But with my lack of basic training, I was demoted to a (live!) hand grenade practicing target. Luckily the grenades that were lobbed at me were low-yield and I survived the demonstration unhurt (see my Chap. 17).

In early April 1945, when the war in Europe drew to a close, my classmates who had stayed with the Luftwaffe (Air Force) were pulled out of their classrooms at the *Luftnachrichtenschule* (Air force communications school) in Halle near Leipzig in central Germany and sent against the—by then—highly experienced American army. My buddies were of course totally unprepared for trench warfare, equipped only with ancient (Italian) rifles. About a third of my class perished on a *single* day, April 12th 1945, including my Austrian friend, Heinrich Brenner, on the banks of the Unstrut river, west of Halle.

By contrast, at the same time in the last weeks of the war, I was operating coastal radars on the North Sea coast in Holland, where we never saw enemy action after the ill-fated Market Garden offensive (17–25 September 1944) by British and Canadian paratroopers near Arnhem (commemorated in the 1977 film "A Bridge Too Far").

14.3 Göttingen to Murray Hill

My next Right Turn came when I decided to commence my studies of physics in Göttingen (rather than nearby Münster) because of the presence there of the "Father of Uncertainty," Werner Heisenberg. But, as so often in life, that main motivation proved irrelevant. I attended just a single lecture (on the shell model of the atomic nucleus) by the famous physicist and I decided I had enough of Heisenberg: I didn't understand a word ("spin-orbit coupling" and other strange concepts). But my interest continued to lie in theoretical physics; I was married to theory you might say. Yet even so, we were obliged to do a lab course in applied physics. One of the experiments in that lab course was to measure the speed of light (electromagnetic

waves) on a long cable coiled up in the basement of the Drittes Physikalisches Institut (DPI). The standard (and expected) method was to send a very short electric impulse into one end of the cable and measure the delay with which it appeared at the other end. Before starting out on this not exactly simple task, I thought it would be much easier to connect the output of the cable to its input via an audio amplifier. Once I cranked up the amplifier gain enough, this feedback set-up would become unstable and the cable would start to "sing." The easily measured frequency of this tone is the reciprocal delay on the cable. This simple method worked like a charm and the lab supervisor (Dr. Hans Severin) alerted the institute director (Prof. Erwin Meyer) who, on the spot, offered me a position as a graduate student at the DPI. Thus, while maintaining my connection with the theory group, I was now working on a thesis in applied physics (involving the statistics of microwave cavity resonances).

And it was precisely this combination of theoretical and experimental physics that propelled me on my trajectory from Germany to the United States. Through my friend the theorist Peter Hasen I learned that William Shockley, Bell Labs physicist and co-inventor of the transistor, was visiting Göttingen looking for good German students for Bell.[1] My thesis advisor, the experimentalist Meyer, was kind enough to write a letter on my behalf to one of Bell's research directors (Winston Kock) that resulted in my employment interview (at the Dorchester in London, no less) and subsequent employment offer (cum visa) to join Bell Labs.

I had initially planned to stay in the States for a year or two, but I soon got married (to Anny Menschik of Radio Free Europe in New York), raised a family (Marion, Julian, Alexander), got promoted (to department head and, later, to director of acoustics and speech research), became a US citizen (in April 1963) and now, 55 years later, I am still living in the States (in Berkeley Heights, N.J.) for 5 months every year.

All of this happened, I believe, as a result of a few Right Turns:

- The decision to study in Göttingen
- To study physics rather than mathematics (my first love)
- To measure the speed of light by an audio feedback loop
- To stay connected with the theory group in Göttingen
- To emigrate to America

14.4 Murray Hill to Göttingen

Ironically, my next Right Turn—after 15 years in America—was to return to Germany. In 1966, Wolfgang Eisenmanger, whom I had invited to spend a summer at Bell Labs, asked me—as charged by the faculty of physics—whether I would be

[1] Shockley's primary interest was in one-dimensional metals on which he had just published himself (*Phys. Rev.* **91**, 228 (1953)).

interested in a call to Göttingen, to succeed Erwin Meyer as a professor of physics and director of the DPI. I was very happy then with my work at Bell and never thought of leaving Murray Hill. But—out of politeness—I answered yes, I would be interested. (I mean, what could I possibly say? "I am not interested"?? That would have been very impolite.)

When the call actually came, in 1968, I was getting a bit bored by Bell and I gladly accepted (after a few details had been negotiated with the state government in Hanover, represented by Herr Ministerial-something Ulrich Hopfe). When Hopfe asked me what salary I expected, I showed him my last salary slip from Bell (for $39,000). Well, Hopfe said, "we can't match *that*," but he would give me all raises and extras until retirement—which amounted to barely half my American income. But, instead of quibbling, I said "fine" and accepted the offer (which included *seven* new permanent positions for the institute: one associate professor, two assistant professors, and four shop people—very generous indeed).

Although taking a 50 % salary cut seemed like a big gamble, but with the Deutschmark's (and the Euro's) unremitting rise against the dollar the income gap was soon bridged and, in any case, it wasn't for the money I had changed jobs. I always thought I would like teaching and, in fact, it turned out I *loved* it.

In addition, and perhaps most importantly, I became a full-fledged physicist again (at Bell my work involved a heavy dose of engineering—speech compression for cell phones and the future Internet). I also widened my horizon far beyond physics and engineering by weekly contacts with colleagues from the humanities (history, religion, philosophy, linguistics, art).

Chapter 15
School Years

Manfred R. Schroeder

15.1 Ahlen

I was born in the house of my grandparents, a large director's villa near the entrance to the coal mine where both my father and grandfather worked. There was a large orchard behind the house (with delicious red and black currents and goose berry bushes and a tall pear tree) and a small kennel for *Hector*, my grandpa's hunting dog.

When I was first born, I was given up for dead and received an emergency baptism (*Nottaufe*) in the kitchen sink because I didn't make any sounds. Later I didn't stop crying for 4 years and my parents had to string woollen blankets across my room to deaden the sound because grandpa was very sensitive to every little noise (*Höllenspektakel*, as he called it).

Not far from the house was a city bus stop where I befriended one of the drivers, Karl Jaxtin, who took me on extended bus tours around the city. I loved it. This was before 1933 but Jaxtin was already a Nazi, an *SA-Mann*, feared by the local communists among the miners for his combativeness. During the war he became a tank driver and perished in the early part of the attack on the Soviet Union.

My first "school" was a Roman Catholic kindergarten, St. Joseph's, in Ahlen in Westphalia (now part of the state of North-Rhine-Westphalia) not far from the coal mine *Gewerkschaft Westfalen* where my father worked as an engineer and my grandfather, Robert Kraemer, was director of below-ground operations. *Westfalen* was the north-eastern most mine of the Ruhr coal fields and they had to go to a depth of 1 km and more to find the black gold.

I remember my nickname from those preschool days: *Kohlenklau* (coal thief). I don't know how I acquired that moniker: I certainly never stole any coal as a child and I wasn't particularly dirty. But most of my friends were miners' children and,

Author was deceased at the time of publication.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_15,
© Springer International Publishing Switzerland 2015

during the depression, *Kohlenklau* was a meaningful concept among them. Sixty years later, one of my Internet passwords was based on this uncommon word. (At another time I chose as a password the non-word name I gave myself as a kid because I couldn't properly pronounce *Manfred*, containing as it does a cluster of three consonants: *nfr*.)

After kindergarten, I attended the parochial *Volksschule* (grade school) for 4 years. One of the few things I remember from those years were the canings (*Stockhiebe*) on the palms of your hand when you had dared to snicker during the singing of the national anthem or forgot a line in a poem. I soon developed calluses and hardly felt the whipping but my friend Eugen and others always howled with pain on such occasions.

It was also at St. Joseph's that I went to Confession and Holy Communion. Since I really didn't know how sin was defined, I confessed some innocent doings simply because they felt good—so, I figured, they must be sinful. And I remember the perennial question in the confessional: *Alleine oder mit Anderen*? (Alone or with others?).

One thing I remember from my early school days was in 1934 (it must have been early July) when I saw police patrols near our house on *Stapelstrasse*: the "Röhm Putsch" was in full swing with many "disloyal" *S.A.*-men being murdered by Hitler and his gang—Goering, Himmler, Heydrich and members of the loyal, elite *Schutzstaffel* (SS)—during and after the "Night of the Long Knives" as it became known. Although barely 8 years old, I sensed that something was very wrong with Germany's leadership.—But we, as children, still got a kick out of the rambling rantings of the *Führer*, sometimes lasting 3 h or more.[1]

On my 10th birthday, in 1936, I got my first bicycle which I rode with abandon, always pretending that I was flying a fighter plane. The bicycle was a gift from my aunt Else Schroeder, a missionary in China, who was on a rare leave from Szechuan where she taught English and Bible classes. As a young woman she had joined the "China Inland Mission" and, after basic training in England, had set out, in 1913, on the 6-week journey, via Berlin, Moscow, Vladivostok, Shanghai, and up the Yangtse river, to her mission near the Tibetan border. She stayed until 1952, 3 years after Mao Tse Dong had taken over and she and the other missionaries were expelled from China—after having "donated" their schools and hospitals to the state, the Christian crosses being replaced by Red Stars. But in those 39 years in China she had become so Chinese that she didn't feel comfortable living among non-Chinese any more. So

[1] Around 1920 Hitler became member # 7 of the *Deutsche Arbeiter Partei* (German Worker's Party) which he soon re-christened *National-Sozialistische Deutsche Arbeiter Partei* (abbreviated as NSDAP)—a clever name, seemingly uniting the opposing political poles: National and Socialist. Hitler also designed the Swastika flag, the swastika derived from the Germanic sun-wheel still rolled down the hillsides at summer solstice in Nordic countries. (The Hindu swastika has a different parentage, I believe.)

she moved to Taiwan to continue her missionary work. She also kept up a lively correspondence with several of her Christian converts from the mainland.[2]

Even before leaving Szechuan, but after the Communist take-over, she was asked by kids in the street where she was from. She answered "Germany." The next day the kids wanted to know "which Germany?" But aunt Else pretended not to understand and stuck to her previous answer: "Germany." Another day later, the kids came back (obviously primed by their parents) asking "who is your president?" And aunt Else answered "Theodor Heuss," whereupon the children scolded her: "*Heuss*—bad Germany. *Pieck* (the East German communist party boss)—good Germany!" The gospel of communism was obviously spreading to some distant corners of the world.

At age 10, I left the *Volksschule* and transferred to the local *Gymnasium* (high school emphasizing the humanities, Latin, etc.). The only thing I remember from that high school is that I was always late. Why couldn't I get to class, like everybody else, before 8 o'clock instead of habitually arriving around 8:05? The teacher was not amused and once turned full circle on his heel in despair. I also remember a few classmates: Hugo Kuhweide, who later perished in Russia, and a very handsome boy, Norbert Plate.

Another thing I remember from my *Gymnasium* days were two students, a boy and a girl, who were always walking to school in the opposite direction. My parents explained that they were Jewish children attending a different school. But one day I didn't see them any more. My parents said that the family had moved away. (Later it became clear to me that, since this was in 1936, the kids and their family had probably avoided a more sinister fate.)

15.2 Rotenburg

In the meantime my father, a World War I combat pilot, had joined the reconstituted German Luftwaffe and, after a brief stay in Göttingen, was assigned to the gigantic *Luftzeugamt* (air force supply center) Rotenburg on the Wümme, halfway between Bremen and Hamburg. So, in early 1937, the family moved to Rotenburg where I started to attend the Middle School. (The nearest high school was 50 km away, in Bremen.) In Rotenburg, the first foreign language was English and I needed private tutoring to catch up. I learned to pronounce the "English" *r* by saying, ever faster over and over again, "Pedinz von Pedeussen," until it became "Prinz von Preussen" (Prince of Prussia). But the result was more Italian than English. I found the whole English experience frustrating and once started crying and the teacher explained:

[2] In the eighteenth century, some Jesuits, who were sent by the Vatican to the Qing imperial court in Beijing to convert the Chinese to Christianity were, instead, themselves "infected" by Confucianism.

"You never had to study very hard; everything was easy for you. But now, for the first time in your life, you have to really exert yourself." I thought the teacher was exactly right and henceforth I started to study more diligently.

It was also at Rotenburg that I first heard the word Gestapo (*Geheime Staatspolizei*). For a time, we were living in a beautiful thatch-roof building that housed the airport architects, assorted secretaries and some officers' families. One day the building burnt down (luckily we had just moved out) and sabotage was immediately suspected. Soon the Gestapo took over, but they couldn't find any culprits. (The cause of the fire was probably one of the tenants, the bohemian kind, smoking in bed and falling asleep—for longer than intended.) The design of the building called *Bauleitung* (construction supervision)—all wood, and as mentioned, a thatch roof—was the brainchild of the commanding officer (Colonel Christensen) who was strong on aesthetic appeal but weak on fire safety. (It's amazing what whims people got away with even after 5 years of Nazi rule.) The Christensens had a beautiful daughter, Elfie, on whom I had cast an early (and distant) eye. But she was very closely watched by the family ever since she had been a victim of statutory rape.

15.3 Jungvolk

In 1937, after I turned 10, I had to join the *Jungvolk*, the junior branch of the *Hitlerjugend*. I had volunteered when I was 7 or 8 because I was envious of the boys marching in the street outside our house. This looked like great fun. I applied and was accepted. But, being the smallest kid, I always marched in the last row with "giant" strides and was still unable to keep up with the older boys. My uniform was of course much too large for me and my belt was sometimes slipping down near my knees. Everyone, my parents included, got a great laugh out of watching little Manfred parade by the house. Later, when I *had* to join, it was no longer such fun. I remember with particular dread the *Kriegsspiele* (war games) where we had to box and wrangle with each other.

A little later, I was appointed to the *Fanfarenzug* (fanfare platoon) but I never learned to play the *Fanfare* properly, and few of the other boy "musicians" did. But when we all played together the tune might have been recognizable—by the law of averages, I presume. To this day I remember some of the campfire songs like the melodious *Flamme empor* (Rise flame).

15.4 Rübeland im Harz

One of the worst experiences while a member of the *Jungvolk* was a summer camp in Rübeland in the Harz mountains in 1937. We had to bicycle through the Lüneburg Heath from Rotenburg to Celle, some 70 km—easy. The next day, we

biked 130 km (80 miles) to Bad Harzburg, including an extended tour of the old city of Braunschweig which lay enroute. One of the sights we saw and that I still remember was the *Alte Waage* (Old Weighing Station) near the center of the medieval town which miraculously survived the war (and which I revisited 50 years later). By the time we got to our youth hostel in Bad Harzburg we were really pooped. During an evening stroll I had lost a 2-Mark coin. It was pitch dark except under the street lights and, naturally, that's where I looked, but with no success.

The next day was really bad. It was only 40 km (25 miles) but all the way up, up, up to a place called Rübeland. Our bikes had of course only one gear and even some of the toughest boys from our class of 11-year olds including Hermann "Bully" Sander cried on his bike. If we could have only dismounted and pushed our bikes (burdened by extra clothing and gear). But this was after all Nazi Deutschland and we were being prepared (without our realizing it) for war.

This warlike spirit became even more pervasive in the camp itself. The *Zeltlager* (tent camp), situated in the midst of the woods, had of course no regular toilets and we had to do with a latrine—I for the first (and last!) time in my life. One of the exercises I remember was throwing (wooden) hand grenades. The food was abominable and the rations much too small.

One rainy day was scheduled for a 60-km bike trip to the Brocken, the highest peak in northern Germany (1,142 m, 3,747 ft). The first half was a strenuous pedalling uphill. Then we ditched our bikes and continued climbing on foot. When we finally reached the top, the fog was so dense that we could hardly make out the outlines of the tourist hotel and the new TV-tower, one of the first in Germany, used during the 1936 Berlin Olympics for demonstrating live country-wide television to gaping audiences—German and foreign. But we never stopped at the top and continued marching right down again. Eventually, the group leaders gave us a little rest and we could munch a sandwich in the roadside ditch. Talk about steeling German youths to the deprivations of war!

After 2 or 3 weeks my parents and grandparents came to visit. When they saw me and my classmates (including the formerly *dicke* (fat) *Peinemann*) reduced to thin, ghostlike states, they immediately invented some pretext to rescue me from my camp ordeal. Now liberated, we spent a wonderful week in the Harz, visiting the famous *Roßtrappe*, according to legend the imprint of a horse's hoof before leaping across a wide gorge. And at the top of the Brocken we actually went *into* the restaurant for delicious cakes and *Torten mit Schlagsahne*.

Before we checked into a hotel for the night my mother inspected the beds and believed to have found some bedbugs. But the owner brushed them away and denied their existence.

In 1938 the family took a vacation on the Baltic. The resort was then called Brunshaupten but later "rechristened" Kühlungsborn by the Nazis—a name, in a show of poor taste, kept by the East German regime after the war. (Kühlungsborn was a typical Nazi coinage reminiscent of Lebensborn, the SS "stud farms" to create more blond, blue-eyed babies for the *Führer*.)



Fig. 15.1 Ingrid, Herta, Helga, and Manfred in Brunshaupten in 1938

In the middle of our vacation my father was called back to his office because of the impending war over the *Sudetenland*. But I still have a nice snapshot from those wonderful days (Fig. 15.1).

One thing I remember was the superb banana ice cream served in a parlor between our modest apartment and the beach. My favorite toy was a small model sail boat that "always" turned around by itself when it hit the pier. But one day it didn't—no matter how long I waited. Where did it go? And where did it finally founder? Perhaps in some distant ocean.[3]

15.5 Opa and His Primus

I also remember, in fact I can still see, my grandfather haggling with a waiter who, he charged, had dared round off a half *Pfennig* to a full *Pfennig*.

A strange apparition I remember, in a park in Bad Harzburg, were two very pale (by German standards of robustness) English-speaking ladies made up with brilliant

[3] On my father's low salary we couldn't of course afford a *Pension*, let alone a hotel, right on the *Strandpromenade* ("board walk"). In fact, the whole trip was *KdF* sponsored.—The next year we took another *Kraft durch Freude* vacation in Strobl on the Wolfgangsee. We were lodged with a small-time farmer (where I caught Asthma from their moldy bedding). One day, at the end of August, I saw a long line of cars at a filling station. The reason? Impending war, I was told. In the meantime we enjoyed the lovely Wolfgangsee, the nearby mountains—especially a gorgeous view of the *Dachstein* glacier. And in Austria they were still serving strawberries and whipped cream (*Schlagsahne*, the real thing).

red lipstick—a sight I had never seen before. Lipstick! Maybe German women applied some on the sly in Berlin but not in the boondocks where we lived. (Hitler had practically outlawed lipstick as a sign of Western moral decay—but his paramour, Eva Braun, used some anyhow.)

At the end of the vacation, grandpa couldn't find his way out of the Harz mountains: "*Seesen*, *wo ist Seesen*?" he inquired perhaps a dozen times.

On another occasion, his car, an *Adler Primus*, wouldn't start moving. My parents knew of course the reason: Grandpa still had the handbrake applied. But they didn't say a thing, always afraid of the irascible Opa Kraemer. Then my little sister Ingrid blurted out "*Opa hat vielleicht die Bremse drin*" (Grandpa probably has the brakes applied). "*Verdammt*, *das Blag hat Recht*!" (Damn, the kid is right). Whereupon he released the handbrake and the car leaped forward and immediately stalled again. Yes, it was always great fun to ride with Opa in his beloved *Primus*—if it *was* running. Opa, now retired, liked to take his car apart but then always needed the help of an expert auto mechanic to put it back together again.

Once *Opa*, then about 63, took a vacation—"alone." Upon his return he always talked about "we" and "us." "Who 'we'?" my grandmother wanted to know. "Me and my Primus" was *Opa's* constant refrain. But *Oma* knew better. (And who took all the pictures of *Opa*? Always "someone who passed by," according to *Opa*.)

Going for an outing in *Opa's* car was always fun. But if there was another adult male passenger, *Opa* always made him crank the car "to save the battery." And days before an intended trip, even just to the center of the city, he would constantly consult the barometer. And if there were just a few drops of rain, the trip would be postponed for "less tempestuous" times.

I once went hunting with Opa but was afraid of shooting at a gaggle of pheasants alighting just in front of us so as not to waste any precious ammunition. Well, my grandfather didn't shoot anything either the whole day. Driving home, we stopped at a game store to buy some dead animals. But my grandmother always saw right through the ruse: the shot size in the purchased animals was never the same size as she had put into the shells in the garage. But Oma never let on. Opa's reputation as a "great hunter" was never questioned.

As a dedicated hunter, he loved dogs, but abhorred cats. One "pitch dark" night, returning from nearby Soest ("*stock-finstere Nacht*, *kein Weg kein Steg*"), he had encountered *eine dicke lange schwarze Katze* (a fat long black cat) crossing the road in the headlights of his car *from left to right*!—a particularly bad omen. Even so, by sheer willpower, I guess, he made it home (some 15 miles).

Of course, cats hated him right back, he was convinced. Once, with friends (the Diekmann's) on a trip to the *Sauerland*, he insisted on parking his car as far away from the street as possible so it wouldn't catch any dust. So poor Diekmann had to move his car (an *Opel* P4). The next morning *Opa's Primus* was covered in dust anyhow and showing (heaven forfend!) cat paws all over. (He was convinced *die Biester* (the beasts) knew exactly what they were doing and to whom because Diekmann's *Opel* was totally untouched.)

At one point Opa had bought a small tent that we pitched in his back yard. We wanted to spend one night in the new tent and so, in our nighties (Opa armed with a

pistol), we moved in at sundown, Opa—a big man—occupied about ¾ of the floor space. Of course, neither of us could fall asleep and after about one hour of tossing we went back into the house. I can still see Opa, in the bright light of the full moon, leading the way in his nightshirt—pistol at the ready.

15.6 Bremen

After 2 years at Rotenburg I transferred to the *Lettow-Vorbeck* high school in Bremen named after the "heroic" World War I defender of German East Africa (now Kenya). In front of the school there was (and still is) a huge brick elephant standing on top of the *Ost-Afrika Denkmal* (East Africa Memorial, now closed). The monument served as a favorite playground for me and my classmates: we could hide behind the elephant's legs after having launched slingshot attacks on the bowler (derby) hats of unsuspecting Bremen burgers passing by.

Since we continued to live in Rotenburg I became a *Fahrschüler* (commuting student), always leaving Bremen at 13:26 for the 45-min return trip home. Fifty years later, the train is still running, but it takes only 25 min now. The train ride was always fun. I remember once telling my buddies that I had observed a certain pretty girl (probably not older than 16 but "old" for us 13-year olds) disappear in the direction of the woods with a *Flieger* (airman): "I guess I don't have to tell you what they were up to." Whereupon said young girl appeared from the next compartment asking: "Tell me, please, what were we up to?" I think I got very red in the face and in future always made sure there was no acoustic leakage between adjacent train compartments.

One thing we particularly enjoyed in Rotenburg were the Wümme river and its many canals. In winter they offered endless opportunities for skating (and flirting) with our classmates. In particular, we boys were always on the lookout for any "fallen angels": we were only too eager to help them back on their feet, thereby getting a first-hand feel of their physical development (or lack thereof). (We were all around 12 at the time.)

While living in Rotenburg the *Autobahn* Bremen-Hamburg was completed and our family took our first trip to Bremen on the expressway. But Bremen wasn't getting any closer and the sun was on the "wrong" side of the car; before too long we were approaching Hamburg. My father had never driven on a clover-leaf ramp and now he couldn't just turn the car around: we had to go to the next exit near Buxtehude (famous for its mentally-handicapped asylum) to turn around.

Around the same time (1938) also came my only chance to see Hitler in the flesh. My father and I had travelled to Hamburg (by train, I believe) to witness the launching of the KdF cruise ship "Wilhelm Gustloff" (or perhaps her sister ship the "Robert Ley") at the *Bloom & Voss* shipyard. KdF stands for *Kraft durch Freude* (strength through joy), the Nazi attempt to assuage the German worker after the rape of their unions on 2 May 1933. These large ships (and several smaller ones) gave working-class Germans the opportunity to visit distant shores such as Madeira and the Norwegian fjords.

Another KdF operation was the *Volkswagen*, originally known also as *KdF-Wagen* and obtainable for a mere 1,000 Reichsmark. But most buyers never saw their prewar VW: it was "high-jacked" by the Nazis and served as a German jeep, especially in Rommel's *Afrikakorps*.—As to the "Gustloff" and the "Dr. Ley," they were used, in fact designed, as *Lazarettschiffe* (hospital ships) for the war. (The "Gustloff" was sunk on 30 January 1945 by a Soviet submarine with over 5,000 refugees from East Prussia aboard.)

One thing I remember from the "Gustloff" launching were the few and isolated shouts of *Heil* when the *Führer* appeared on the platform. I was amazed because on the radio we always heard the endless *Heils* when Hitler loomed. Were we being hoodwinked? Not necessarily, because, it was explained to me, the *Bloom & Voss* workers were mostly former communists.

The actual christening of the huge vessel was performed by the widow of Wilhelm Gustloff, a Swiss Nazi "martyr" who had been assassinated by opponents.

Incidentally, Rotenburg, too, had a large asylum for the mentally handicapped: some two thousand patients—a lot in a little town of six thousand (before the air force "invasion"). Many of the patients (kept as inmates) were allowed to stroll around town on weekends: always in twos, arm-in-arm. It was here, around 1939, that I first heard the expression "*verschickt und nicht angekommen*" (sent away but never arrived), a euphemism for the Nazi program (dated September 1, 1939) to "eliminate unworthy life." Indeed, after the beginning of the war, the patient population dwindled rapidly.

Soon after the war started, British bombers appeared over northern Germany and my parents decided to move to East Germany, to Liegnitz in Silesia, where my father had just been transferred. They didn't want to get separated again—and they were afraid of bombs. Of course, the real bombing campaign didn't start until several years later but my parents never regretted their decision.

Not far from the *Luftzeugamt* Rotenburg, a sizeable sham airport was built up by the German air force. But the British were not deluded: they dropped *wooden* bombs on the make-believe operation. This was a mistake because the Germans henceforth saved a lot of money by closing down their "fairy port."

The 1 year at the Lettow-Vorbeck school in Bremen was one of my best. I befriended Dietmar Risch, one row behind me. But it was kind of friendship at a distance because he lived in Bremen and I in Rotenburg. But we met some 60 years later—and recognized each other immediately. The home-room teacher lent me his (amateur) telescope and asked me to describe to the class what I had seen in the sky.

One event at the Lettow-Vorbeck school that I remember vividly is a Reichstag speech given by Hitler at the end of April 1939, which we all had to listen to, in which he ridiculed a recent demarche by Franklin Roosevelt demanding from the German government a pledge of nonaggression toward a long list of countries. Hitler assured Roosevelt that he had no intention of attacking—and there followed another long list of mostly irrelevant nations. And his listeners got a big laugh at the expense of FDR. One of the countries Roosevelt had listed was Syria and Hitler, to great general amusement, reminded Roosevelt that Syria was a *French* protectorate and that he should address his plea to Paris.

Significantly, and no one noticed at first, Hitler left out Poland, the only country that was really threatened at the moment. This propaganda debacle (for the democracies) could have been avoided if Roosevelt had concentrated on one or two countries, say Poland and Lithuania, in which case the omission by Hitler of Poland would have been glaringly obvious.

When it came to dealing with foreign nations, Roosevelt, the superb politician, was often "out of his depths." How could he call the mass-murderer Joseph Stalin "Uncle Joe"—a term of endearment (which, by the way, the Soviet dictator, not being familiar with American mores, took as an insult). And why did he call (in May 1943, in Casablanca) for Nazi Germany's *unconditional* surrender? Churchill was aghast. For Hitler and Goebbels this was, of course, grist on their mill, giving them a superb excuse for prolonging the war. (On the other hand, Roosevelt was one of the few Americans who, early on, saw the danger of Nazi Germany. While most of his countrymen were still dreaming of peace, he supported the British, stretching neutrality to its very limits: by the Lend-Lease legislation, the donation of 22 "mothballed" destroyers, and declaring the West Atlantic a war zone in which the U.S. Navy would sink any "unfriendly" ships (i.e., U-boats).)

In those days, Hitler was at the peak of his prestige in Germany: he had just conquered Czechoslovakia, "recovered" the *Memelland* from Lithuania and was flush with hubris after his big 50th-birthday celebration on 20 April 1939, attended by diplomats and military attaches from around the world.[4]

I saw the downside of the *Führer's* policies first-hand when a (distant) relative (a customs official) showed me around the Bremen docks and warehouses where I caught sight of a lot of furniture in storage. "Belongings of Jewish emigrants," my guide explained "but they cannot be shipped on during the blockade of the German ports during the war." The loss of the Jewish citizens after 1933 created a void in German intellectual life that, 70 years later, has still not been filled, I believe. Without that deprivation, Germany would be a much more vibrant country than it actually is.

One thing that made Bremen very attractive to me as a commuting student from the boondocks, was the wonderful smell of roasted coffee—speaking to me of distant lands—that pervaded the city. Bremen was famous for its coffee roasting. *Kaffee Hag*, the first decaffeinated coffee, had its home there.

15.7 Liegnitz

When we moved to Liegnitz I had to leave the school and its director wrote a spontaneous letter to my father that, to his great regret, the best student, both academically and *im Betragen* (general comportment) was leaving. I had been an

[4] The place where Hitler stood during the mammoth parade on the *Ost-West-Achse* (later *Strasse des 17. Juni*) is now occupied by a massive Soviet monument to their victory over Germany—constructed, fittingly, with marble slabs from Hitler's *Neue Reichskanzlei*.

inspiration to the whole school, he wrote. Wow! My parents were of course delighted but the letter went to my head and in Liegnitz I became one of the worst students (especially *im Betragen*) so that the director upon my leaving *that* school said: "We are glad you are leaving." And he didn't even know some of the worst things my classmates and I had perpetrated.

For example, we broke into the school at night with a pass-key that I had illegally duplicated. Among the items we stole was potassium metal that we made to "dance" on a lake in the city's park after midnight. (When the locksmith wanted to know for what purpose I wanted the key, I said it was for our apartment door. He said it looked more like a general key for a large building. But he made the key anyhow—from a soap impression.)

We also launched for-sale ads for dogs in two local papers that brought a barrage of potential buyers to our home-room teacher's home. My ad, which was for "my dearest little lap dog," never appeared because I had to giggle too much while placing the ad by phone and the paper double-checked with the named teacher. But the ad concocted by my friend Franz Prauser for an "exceptional hunting dog," did appear and several incensed farmers, who had come from afar (Jauer, now Jawor), wanted their travel expenses back and compensation for a lost day at work. Our music teacher acted as a spy to pry the secret from us as to who had the "ingenious" idea for the ads. But we kept mum—because the affair was taking on legally actionable overtones. Nobody ever unmasked the malfeasants even though it was clear to everybody that the culprits were from class 5c (*Fünfzeher*, i.e., five-toed creatures).

At home I was manufacturing explosive powders and to accelerate the mixing process of the different chemicals used, I dissolved them in water and then evaporated the solution over the kitchen stove—boom. A great mixing invention!

At another time I ate all the crumbs from a crumb cake my mother had baked for Easter and then "plowed" up the denuded surface of the cake to make it *look* like crumbs. Of course, nobody was fooled and I was forbidden to eat any of that sorry cake (which I didn't care to, anyhow).

In my free spare time, I built a radio transmitter that fitted into an unobtrusive little box (disguised as a small suitcase). It sported a high-quality microphone and was intended for listening in on the teachers' conversation in the teachers' room. It used a single vacuum tube that served both as an audio amplifier and a radio-frequency oscillator. It never worked too well but managed to give my parents a big scare when I was testing the transmitter from a neighbor's house and then rang our doorbell to check on the signal strength. My parents thought it was the Gestapo at the door. (It was, in fact, *very* dangerous to operate illegal transmitters in Nazi Germany.)

This was also the time (1940), that I contracted my first "serious" disease: *Scharlach* (scarlet fever) which left a little scar on my heart and made me near-sighted. Four years later I fell ill with rheumatic fever which left me practically immobile for several weeks. (The nurse had to put her arm through my pyjama sleeve to pull my arm through.)

In the hospital I furtively listened to the American Forces Network (Glenn Miller!) over the BBC. While reconvalescing the SS tried to draft me but, with the help of my father, I was able to escape.

In the summer of 1940 we vacationed in Schreiberhau in the *Riesengebirge*. I remember especially July 19th, when Hitler gave his victory (over France) speech in the *Reichstag*, announcing the promotion of several generals to field marshal and Herman Göring to the unique rank of *Reichsmarschal* (a title that goes back to the Holy Roman Empire). He offered Britain peace—guaranteeing the British Empire—if he could have a free hand in continental Europe. As is well known, the British declined the preposterous gesture. In fact, they didn't respond at all.

One date that sticks in my mind from my year at Liegnitz was Monday, 23 June 1941, the day after Hitler had attacked the Soviet Union, which, true to its pact with Nazi Germany, was still shipping wheat and gold and other items in short supply in Germany the night before the attack. That morning the first course was geography, taught by Dr. Pfeiffer—a Party member (always sporting a swastika on his lapel). He unrolled a large map of the Soviet Union and said "the USSR is a very large country." Prolonged pause. Then he said—and it was obvious that he didn't believe it himself—"but I am sure that the *Führer* knows what he is doing."

During my two summers in Liegnitz I worked in the cucumber "industry." I had to sort (on a conveyor belt) the freshly-harvested produce and carry heavy sacks (40 kg) of cucumbers—no mean feat for the pencil-legs person I was at age 14. But I earned 32 *Pfennig* an hour—enough for one section of track for my future gauge-00 model railroad that, however, never came into existence because by the time I had enough money (about 250 marks for one summer, working 10 h or more per day) there were no more model railroads available. (I remember one *D-Zug* locomotive patterned on the 03-series of the *Reichsbahn* that then cost 27.50 *Reichsmarks*—which, many years later, I bought for our children for many times more *Deutschmark*.)[5]

During the cucumber harvest I also encountered several French PoWs whom I asked (since I knew little French) "Parlez-vous brittanique?" No, they didn't. But they were interested in our cucumbers (for which they would "later" give us bananas).

Another diversion was building model airplanes from plywood, parchment paper and lots of glue. This was fun. And the planes actually flew.

Since I wasn't any good at regular *Hitlerjugend* activities, they made me a *Gebietsgeldverwalter* (regional accountant) for which I had to attend a special course in Striegau near Liegnitz. I don't remember any of the math from that course, but I do recall the mid-night alarms when we had to get up and, in our nightshirts, run barefoot around the snow-covered grounds. So much for the Nazi idea of "accounting."

[5] Earlier, in Rotenburg, I was "working on my railroad" by collecting *Blaubeeren* (blueberries) for 32 *Pfennig* a (metric) pound.

Incidentally, when we first arrived in Liegnitz, my parents wanted me to attend the *Ritterakademie* (Knight's Academy), a boarding school for Silesian nobility (*Landadel*). But the school also accepted a few local students. On my first day, I was asked by the homeroom teacher my name and whether I was an *Akademiker*, meaning, in this case, whether I was a boarding student. I didn't know what *Akademiker* meant, so, to be on the safe side (and because it sounded good), I answered "yes." Then the teacher said something that became a standing expression in the family: "*Nicht mal von*?" (Not even *von*?) What's behind the teacher's comment? Well, all the *Akademiker* were at least *von* something if not a Baron or an Earl. Also the *von* Schröders were not unknown in Germany. They were bankers; one of them, Baron von Schröder from Cologne, had helped bankroll Hitler's election campaign in the early 1930s. And, of course, there is the famous London bank "J. Henry Schroder." But I was just an ordinary Schröder without title (or money, for that matter).

In any case, I lasted only for three days at the *Ritterakademie*—not because I lacked any titles but their language courses didn't match mine from Bremen.

Later, I learned that the *Ritterakademie* (now *Academia Richerska*, in Legnica, Poland) harbored an anti-Nazi-group of students—a very rare circumstance in Germany during the Third Reich. (The only other known case is the *Weisse Rose* group lead by Sophie Scholl, her brother Hans, Christoph Probst, and Professor Huber at Munich University. On the 18th of February they were betrayed by a janitor who saw them distributing "subversive" leaflets, pointing out some of the horrors committed by the *Wehrmacht* (!) in the Soviet Union. They were unceremoniously condemned to death by the infamous Roland Freissler of *Volksgerichtshof* fame, and guillotined the same day on 22 February 1943.)

Long after the war (in 1992) Anny and I visited my old school in Liegnitz. A very kind superintendent showed us my old classroom (5c) and the *Zeichensaal* ("drawing room") where I had perpetrated many pranks—some not so innocent, as when I used my limited drawing talent to inscribe some "A's" for me in the teacher's records book. This *Urkundenfälschung* (falsification of documents) was thankfully never discovered.

During the same trip we also visited Breslau (now Wrocław), Auschwitz, Birkenau, the former Jewish ghetto in Cracow, and the Polish Catholic shrine in Czestochowa (home of the Black Madonna).

The final destination of this excursion was a conference on chaos in the Royal Castle in Rydzyna (Reisen before 1919). This magnificent structure had been meticulously restored by Polish artisans after having been torched in a bonfire by the departing Soviet troops in 1945. (It was destroyed in just a few hours but it took 45 years to rebuild!)

15.8 Ballenstedt

To avoid constant school changes, my parents sent me to a boarding school in Ballenstedt in the Harz mountains: modern buildings surrounded by forests and a beautiful little lake nearby on whose shore I spent many a dreamy moment. This was in September 1941. I had just turned 15, feeling very lonely, far from family and old friends.

Other than the lake, one of the few things I remember from Ballenstedt was an assembly of the entire school in the loudspeaker-equipped mess hall on 11 December 1941. Hitler was giving a speech to the *Reichstag* declaring war on the United States. He didn't actually say as much, but instead pointed out that Roosevelt had already turned the Atlantic into a war zone by declaring that the U.S. navy would sink, without warning, any ill-intentioned submarine. Hitler also stressed that gallant Germany would of course stand by its great ally, Japan, which was "fighting for its life" since December 7th. Hitler, naturally, didn't stress the fact that Japan had been the aggressor and that Germany wasn't actually treaty-bound to join Japan in the fight. Hitler believed that the U.S. would be absorbed by its war in the Pacific. But Roosevelt, as is well known, made the decision "Europe first" because that's where he saw the greater danger.

The question has often been debated why Hitler chose to declare war on America at a time when he had just suffered his first major military defeat (in front of Moscow). The reasons, I believe, are the following:

1. He knew, as Roosevelt did, that war between the USA and Nazi-Germany was inevitable. So, in order not to lose the initiative, he preempted FDR. Otherwise it might have looked as if war with the USA had been *forced* on him.
2. He believed that America would turn to the Pacific and Asia rather than Europe, because he underestimated how much of an ogre he had become to the world outside Germany.
3. He honestly believed that America's power was much overrated. In several of his speeches he stressed the "fact" that America had only 153 million people—and he had over 300 million. (He must have counted all the captive nations) Also, he didn't think much of America's industrial prowess and banished Henrietta von Schirach, who was part-American and wife of the *Reichsjugendführer* Baldur von Schirach, from his dinner table because she had dared say some good things about the USA and the capabilities of its industry. Hitler liked to surround himself with women (Leni Riefenstahl, the legendary film-maker, Hanna Reitsch, the daredevil pilot who had first flown a helicopter *inside* a (large) building, to name just two) but Henrietta had to go. This was a pity because she could have given her *Führer* much cogent advice.

Of course Hitler was wrong on points 2 and 3, which he had to learn the hard way.

Someone who did know of Japanese intentions was Joseph Stalin who had been informed by Richard Sorge, the Nazi Party member and Soviet master spy in Tokyo. This allowed "Uncle Joe" to withdraw his crack divisions from the Far East to face the Nazis in front of Moscow and deal them the first blow in early December 1941.

Like most, if not all, boarding schools in Germany during the Nazi period, Ballenstedt was infested by ideology. On my first day, my homeroom teacher asked me whether I went to church regularly and when I answered "yes," he said that I would be free continuing to do so, but I would be the only one in the entire school.

We were also "encouraged" to study Italian. But I dropped out after a short time because I didn't like being told what to study. Later in life, when I was not forced to learn Italian, I took a great interest in the language. In 1951, I went to Italy, for a summer course *La Lingua e Cultura Italiana*, which the University of Pisa held at the Collegio Colombo in the *Pineta* (pine forest) at Viareggio. I loved it!

After 4 months, I implored my parents to find me another school. The straw that broke the camel's back for me was the following incident: I was standing by a window of our dormitory clipping my fingernails. In doing this I was of course bent over my fingers, whereupon one of my *classmates* (not even a teacher) admonished me to stand straight, without a hunchback. How do you work on your hands while standing at attention? Well, I didn't ask, and on October 16, 1941,[6] my father came through with a new school for me: Neuzelle near the Oder river.

15.9 Neuzelle

Neuzelle, like Ballenstedt, was a *Nationalpolitische Erziehungsanstalt* (national political education school). But what a difference to Ballenstedt! The official motto of these schools, the old Prussian *Mehr sein als scheinen* (Be more than appear), was inverted by us to the ungrammatical *Mehr schein als seinen*.[7]

[6] The date was also marked by one of the more important events of the "Russian campaign" (as the Germans liked to call the war on the Eastern front): October 16th saw the most serious riots in the history of the Soviet Union. In the general mayhem, the cars of party-bureaucrats fleeing Moscow were overturned by frantic Russians fearing the arrival of the *Wehrmacht* within a few days. Government departments and foreign embassies had already been evacuated to Kuybechev and a train was readied for Stalin to leave Moscow. But the Supreme Leader, after inspecting the train, made one of the most fateful decisions of the war: he decided to stay in the city and if necessary, perish with it. Soon Russian forces rallied and, only 2 months later, dealt a devastating blow to the German army just west of Moscow—the first major Nazi defeat in the war.

[7] Neuzelle was descended from Frederick the Great's famous Potsdam *Kadettenanstalt*. (The king had once offered Giacomo Casanova a position as a teacher of French (and good manners!) in Potsdam, but the great lover declined when Frederick showed him around one of the dormitories—with unemptied chamberpots. Giacomo fled Prussia forthwith for the better smells of St. Petersburg.)

I soon discovered that some students even subscribed to communism—unheard of in Nazi Germany. In other words there was much more freedom of expression. And some teachers even had a sense of humor as when one of them wanted to discourage us from using too much toilet paper: "His parents swing from chandelier to chandelier just to conserve the parquetry and what does the *Jungmann* (as we were called) do? He uses two, three or more sheets of toilet paper for a single wiping!"

Another teacher, Studienrat Blanke (our physics teacher) was perhaps an anti-Nazi because he disappeared from one day to the next without any explanation. Was he sent to a concentration camp? We never heard from him again. He had honored me by appointing me his assistant for the physics lab course. But soon after he discharged me again. Was this a precaution on the teacher's part to spare me any involvement in his fate? Blanke also impressed me with his (very un-American) credo that it takes several generations for humans to rise in society from laborer to employee to professional status to high governmental office. (This may have been a veiled allusion to Hitler who had risen from corporal to commander-in-chief in a mere 18 years.)

In Neuzelle I also operated my "secret" transmitter, which I had originally built to listen into the teacher's conferences in Liegnitz. (As a back-up, we also strung electrical wires along the wooden floors to a bug in the teacher's room. But the plan never came to full fruition.) Once, in Neuzelle, I saw an army (or SS) radio direction finder parked in the courtyard. I turned off my illegal transmitter in a hurry and never turned it on again for a long time.

I remember one student (Plokarz), 2 years older than I, who wrote in his composition assignment "What I want to become and why" that he wanted to become a German army officer because uniforms, especially officers' uniforms, carried a lot of prestige then and guaranteed a successful "love" life. As far as money was concerned: his older brother would inherit the family's shoe factory in Beuthen (Bytom) in Upper Silesia and he would get half the profit. He later did become an army officer but he had one weakness (other than girls): kleptomania. Although "rich," he would steal from gift parcels of his own troops and was thrown into the brig—from which he escaped to desert to the advancing Soviet army where he might have identified himself as Polish. (The name is Polish.) As a result, he was condemned to death (*in absentia*). I don't know what happened to my friend Plokarz; I never heard from him again. (Was he killed while crossing the lines?) The only thing I heard was from the school's director (Burow) as he broke the news to me: "*Plokarz ist zum Tode verurteilt*."

Director Burow was one of the teachers with whom I had a close relationship: even though I was already serving as an auxiliary with an antiaircraft unit, we corresponded about mathematical problems. Once I wrote him that I could not integrate the function $\exp(x)/x$ and he wrote back that I had stumbled on an "insoluable" problem, that there was no solution in elementary functions—quite a shock to me then.

Another word about my friend Plokarz: when I first arrived at Neuzelle, shyly standing at the classroom door not knowing where to sit, a handsome, swarthy boy

with beautiful, wavy long hair[8] invited me to sit next to him. After that, his grades made a "quantum" leap upward: he was always looking over my shoulder and copying what I had written. In fact, the grades in mathematics of the entire class went up: everybody was copying my homework solutions. Director Burow was very pleased with his students but in the end he became suspicious of what was going on and he assigned 12 *different* homework problems, one for each student. But the grades stayed unshakeably high. The only difference was that I now had to solve *12* problems each time.

In spite of my top grades in math and everything else (including Latin and chemistry), I was in danger of not being promoted because of an *ungenügend* (failed) grade in sports. A top school administrator was dispatched from Berlin to witness this un-German *Schlappschwanz* (weakling) called Schroeder. At shot-putting he was standing next to me while the iron ball dropped not far from our feet. In spite of his imposing presence (in the uniform of an SS colonel, no less), I didn't mind how far my shot went. And the colonel uttered a great truth about my character: "It's not that you haven't enough *strength*. The problem is that you lack the *will* to succeed." Right—as far as shot-putting and most other competitive sports were concerned, I couldn't care less—nor was I eager to prove different.

During the summer of 1942 our class was shipped to German-occupied Poland, to a village then called Nussdorf (near Środa) to help with the harvest. I was assigned to a farmer from one of the Baltic states, a dour, ethnic German who was given a former Polish farm to run which, for all I knew, may have belonged to a German before 1919. We had to get up at 5 in the morning to harvest buckwheat overgrown by weeds, especially thistle. It was very painful work and as the sun rose during the day it became hot as a furnace. One of the few good things I remember was the creamy milk from "our" cows. (By slowly lowering a glass into the milk you could skim just the cream—delicious.)

Another of my duties involved taking pigs to their male counterparts on a nearby "stud" farm. I must say, I was amazed at the amount of liquid involved in the process. Some of it was still dripping from my pigs on the way home.

Watching the impregnation of one of our cows by an eager bull was another spectacle that was new to a city boy like me. (I had never seen such a long-and-thin thing in action before.)

Some of the farm hands were Polish and they were supposed to draw their caps before members of the "master race" like us. (I am still ashamed that I once berated a Polish man for not removing his cap before me.)

During the trip to Nussdorf I saw some people in prisoner's garb working on the railroad tracks. Were they criminals or perhaps *political* prisoners, i.e., enemies of the Third Reich? I don't know, but looking at those poor men gave me a fearful feeling. Later, in Poznan (Posen), I saw a large camp, a soccer stadium surrounded

[8] This getup was already highly unusual in Nazi Germany where a closely cropped chevellure was *de rigeur*.

by barbed wire. After the war, from reading books about the holocaust, I learned that this was an *Arbeitslager* ("work camp"), a transshipment point for forced labor.

In a speech welcoming us to the *Warthegau*, the Nazis' name (from the river Warthe) for the annexed part of Poland, a high party official boasted that there was another *Wartegau* (waiting province), namely Switzerland, which was just waiting to become a German province. A despicable pun!

On or around 19 August German mountain troops (*Gebirgsjäger*) planted the swastika flag on Mount Elbrus, the highest peak in the Caucasus *and* Europe (5,642 m—18,510 ft.). One should think that Hitler, a lover of mountains himself, was delighted but actually he was not at all amused: he expected his troops to go after the oil in and around Baku on the Caspian Sea instead of wasting time on mountain expeditions. In the event, his armies never reached Baku, running out of gas—irony of ironies—near Mozdok at a bridge over the Terek 60 miles from Grozny (Chechnya) but still 350 miles from Baku. One bridge too many! (August 19, 1942, was also the date of the Allied raid near Dieppe on the French Channel coast.)

The longitude of Mozdok ("Dense Forest") is almost 45° East, the farthest the *Wehrmacht* ever got to the Indus river in India (around 70° East), the agreed upon demarcation line with the Empire of Japan (Tokyo is at 140° East). I forget the "boundary" between Japan and Germany in North America. (Was it the Mississippi, the Continental Divide?)[9]

This brings to mind the trick question that my friend Tom Rossing asked me: "Which of the 50 states of the USA is, respectively, the most northern, western and eastern state in terms of latitudes and longitudes." The surprising answer: Alaska, Alaska, *Alaska*! I leave it to the interested reader to ascertain this incredible-sounding result. (The southernmost state is of course Hawaii: the southern tip of the "big island" has a record latitude of just 18° North.)

Another reason I remember 19 August was that it was the day my maternal grandfather, Robert Kraemer, died of cancer at the age of 68 in distant Hamm (Westphalia). So my patriotic *Opa* never had to witness a German defeat (other than the debacle near Moscow in December 1941, which of course was kept secret from

[9] The Indian nationalist Subhas Chandra Bose (and friend of Gandhi before they parted ways) offered his services to Hitler for "liberating" India. The plan, however, fell through because the Germans failed to reach India. Bose was then shipped by submarine to Japan, where he formed a "Free India" division which actually saw action in India but was defeated by the superior weapons of the British.—The other day I asked an Indian sales clerk in a supermarket whether he knew S.C. Bose and he answered without hesitation: "Yes—a great man!" In fact, S.C.B. is still held in high regard by many Indians, because he was ready to fight the British by the force of arms—as opposed to Gandhi's philosophy of nonviolent resistance.—Another famous Bose was Satyendra Nath Bose who invented Bose-Einstein statistics and for whom *bosons* are named, such as the particles of light, the photons. S.N. Bose was also a close family friend of S.C. Bose, the nationalist leader. (From Dwarka N. Bose, University of Calcutta, in *Physics Today*, June 2007, page 12.)—(My acquaintance Amar Bose of loudspeaker renown is apparently unrelated to the two more famous Boses.)

ordinary Germans). As long as he lived, things were moving "forward" (to use a current term).

Opa Kraemer was sacked, at age 60, by the Nazis from his job in a coal mine in 1933 to make room for new workers. This was part of their solution to solve the (severe) unemployment problem in Germany. Opa always liked to drink a little beer and *Schnaps* which, however, according to my grandmother Maria, he all sweated out in the (very hot) mine.

He was also the first "acoustician" in the family: He negotiated contracts with suppliers in his personal bathroom at the mine where he kept the water filling his tub running to mask anything that was said in the privacy of his "office."

Back to Neuzelle. Another thing I remember from that school were the letters of former students from Stalingrad which were read to us in January 1943. Some letters were written in the vast Tractor Plant, one of the most hotly contested battlefields inside the city of Stalingrad. Of course, we never heard from these woeful alumni again. The last German pocket of Stalingrad capitulated on 2 February 1943, three days after the tenth anniversary of the Nazi seizure of power, and Hitler, who, like many leaders, preferred upbeat news to bad tidings, left it to his deputy Göring to give the anniversary speech. Göring, in collusion with Goebbels, turned the debacle into an historic event likening it to Greek tragedy (Leonidas, Thermopylae and Sparta).[10]

15.10 Pölitz

Two weeks after the Stalingrad debacle, on or about 16 February 1943, although I was only 16 ½ years old, I was drafted as an antiaircraft auxiliary (*Flakhelfer*) to serve with a battery of six 88-mm guns which, together with 17 other batteries, was guarding the huge hydrogenation plant in Pölitz on the Oder river (now Police,

[10] One day before the German surrender, Hitler had promoted Colonel General von Paulus to Field Marshall, hoping that von Paulus would kill himself because no German field Marshall had ever capitulated. That way, Hitler could claim that everybody had fought the Bolsheviks until their last breath—a legend Nazi-propaganda spread anyhow. It was only after the war that it became widely known that the 100,000 surviving troops (out of 300,000) had actually surrendered. To fortify their despicable lie, the Nazis, in one of their most heinous crimes against their own people, *withheld* letters and postcards from the prisoners to their families. (They were discovered long after the war in East Germany.)—The Nazis offered to exchange Yakov Stalin, whom they had taken prisoner, for Marshal Paulus (to court-martial him?). But father Stalin refused the deal. Later Yakov was killed while trying to escape the German POW camp. Stalin is said to have been relieved when the news reached him. A prisoner of war was considered a traitor and many were, in fact, shot out-of-hand when the Western Allies shipped them back (often against their will) to the motherland after they were "liberated" at the end of the war. The shootings sometimes took place in the presence of the American officers and truck drivers who had delivered the poor souls (personal communication).—(Galina Dzhugashvili, Yakov's daughter, who died in August 2007, maintained during her entire life the fiction that her father fell in battle, in spite of massive evidence to the contrary (*New York Times*, CNN, August 28, 2007.)

Poland), which was said to produce 25 % of German gasoline made from Silesian coal. When the battery commander saw me unpack a book entitled *Funkmesstechnik* (radio measuring techniques) he assigned me to the battery's fire-control radar (*Würzburg*, 50 cm) because *Funkmesstechnik* also happened to be the German word for radar. I have detailed my life as a radar operator during and after the war in the chapter "Radar" of my memoirs. But besides the radar activity, mostly in the afternoon (training) and occasionally at night (enemy planes), we had "regular" (actually somewhat curtailed) high school instruction. I think English—of all languages—(but not Latin!) was dropped. Of course we were not on speaking terms with the English-speaking countries. But why keep Latin? Did the Nazis expect a revival of the Roman empire? I remember one of our teachers asking one day whether we realized that shooting at people could mean killing human beings. Well, I must confess, we had never thought of it. But afterwards I could never forget that soldiering meant *killing* people, no matter how "justified" by self-defense or "preemption."

In August or September 1943 I, along with many other electrically interested youngsters, received an "invitation" from Hermann Göring to register for a special course in radar. This sounded interesting and I applied and was accepted. So on Saturday 23 October I decamped for the *Reichsausbildungslager* (Reich Training Camp) "Prinz Eugen" in the forsaken Westerwald near the triple point where the German states of *Westfalen*, *Hessen*, and *Rheinland* meet.

15.11 Prinz Eugen

Prinz Eugen (named after an Austrian field marshall, Prince Eugen of Savoy) was a combination high school, electronics college, and boot camp (on weekends and, occasionally, at night). The schedule of duties was so demanding that a high air force officer, General Martini, upon seeing our emaciated looks, took pity on us and sent us all to the Austrian Alps, the *Schönleiten Hütte* in the *Pinzgau*, for 2 or 3 weeks of skiing. This was in March 1944.

In addition to skiing, we had ample time for reading. My friend Jobst von Behr reminded me recently that I was reading a biography of Leibniz. The book was part fiction, part fact and I wouldn't touch it today, but it made a deep impression on Jobst and me at the time, when we were both 17. I haven't read fiction for ages, other than the "The DaVinci Code," a real page turner. But I swore not to read another fiction book for a long time. Life, *real* life, is simply too interesting to waste time on fiction, good or bad. But in my youth, I read quite a bit of fiction and enjoyed it. Like most German kids I read many volumes of Karl May's classic (American) Indian exploits (Apache Chief Winnetou, his beautiful, blushing sister *Nscho Tschi* (Fair Day), paleface Old Shatterhand and the Sioux and Mohicans). It was marvellous and of course pure fiction. (It is said that Karl May never set foot on American soil before writing these books. But where did he get the enchanting name *Nscho Tschi*?—Gesundheit!)

Another book I read, or was read to, was "Der Struwwelpeter," the hilarious depiction of the ills that befell "bad boys" (not cutting their fingernails, not eating their soup and not combing their hair—*Struwwelpeter*, in fact, means "messy-hair Peter").

And there are probably few Germans, young or old, who were not enthusiastic readers of Wilhelm Busch's captioned cartoons of assorted mischief makers, especially Max and Moritz. (Some of the funniest exploits "happened"—or, in any case, were invented—in the Ebergötzen flour mill not far from where we live, when we live in Germany.)

Once I knew enough Dutch, I read a few books in Dutch. I particularly liked the Flemish satirist Willem Elsschott (Alfons de Ridder) whose *Kaas* ("Cheese") and *Lijmen* (literally "bamboozle" but translated and made into a movie as "Soft Soap"). I cannot imagine Elsschott being translated into any other language without much being lost in translation.

The first serialized crime story I read (at age 10!) was "*Dr. Crippen an Bord*" which was later made into an equally gripping movie. The story is retold in a recent book *Thunderstruck* by E. Larson, in which the Crippen murder case is intertwined with the story of Marconi's radio experiments. (Crippen and his paramour were uncovered by the ship's captain when they tried to flee from England to Canada—thanks to radio telegrams between Scotland Yard and the ship. According to Larson "the Crippen saga did more to accelerate the acceptance of wireless as a practical tool than anything the Marconi company previously had attempted.")

Another bunch of books I loved were Hans Dominick's *Zukunftsromane* (literally "future novels," i.e., science fiction). I remember particularly his *Wettflug der Nationen* (Air Race of the Nations), describing an air race around the globe in which, of course, Germany won, but the US was a close second. I forget how the Italians fared, but the Japanese were painted as real devils, mixing sand with the engine oil of their competitors. (This book must have been written before the *Antikomintern* (Anticommunist-International) pact that allied Germany, Japan, and Italy for any future fight with the Soviet Union.)

As to more serious reading, even in high school, in Neuzelle, I once had to give a talk on an exposition by Max Planck on "free will and physical determinism." I don't remember whether I understood much (or anything) of Planck's argument. Today, 65 years later, I still don't know the answer as to how in a world governed by the deterministic laws of physics something like personal free will can exist. Yet, we all know that it does and that we are responsible for our deeds and misdeeds. The fact that quantum mechanics introduces an element of randomness looks like an easy escape hatch out of this conundrum. But is it relevant to free will? (One thing I do remember from this talk in 1942 was my near-sightedness and that I was too shy to wear glasses in class. I therefore placed my notes for the talk on top of a high stack of (learned) books, so I could read my notes without glasses.)

Another serious book that I read (or rather was read *to* by our house maid) was Erich Maria Remarque's "*Im Westen nichts Neues*" (All Quiet on the Western Front). I think she (and we children) liked the scene best where a row of soldiers are peeing "liquid silver" in the moonlight outside their dugout. The book was turned

into an acclaimed movie that Goebbels and his Nazi thugs tried to boycott (because of its pacifist message) by releasing hundreds of white mice during the opening minutes of the movie. As a result, the gala premier had to be cancelled but the mice surely helped to publicize the film.

As I have said in another chapter, besides the prescribed school literature (Caesar, Goethe, Schiller, Shakespeare), I gulped English crime fiction, especially by James M(allahan) Cain (*The Postman Always Rings Twice*, *Double Indemnity*).

Chapter 16
July 1943

Manfred R. Schroeder

July 12th, 1943, was my 17th birthday. I was attending a gliding school in preparation of an intended "career" as a fighter pilot. This was ridiculous. I was already nearsighted and wearing glasses. But my father had been a combat pilot (*Kampfflieger*) in World War I so they made an exception. Of course, I never piloted a powered plane, much less a fighter. The *Luftwaffe* was literally running out of gas. But in the meantime I had great fun sailing quietly over the Pomeranian landscape. (The first time up I lost my way because the church spires I had memorized were no longer visible on the horizon: they were *below* me.)

One of our duties between flying exercises was guarding sheep. (Generally, German airfields kept the grass clipped by employing sheep.) I was given a large flock and guard dog—and promptly fell asleep in the shade under the wing of a parked airplane.[1] When I woke up, the sheep (cum dog) were gone! As far as the eye could see: no sheep. I panicked, running this way and that, all the while seeing my sheep already blocking the runway or worse. I was afraid of a court-martial. But I was lucky: after one or two false starts, I found my lost flock, all held in a tight circle by my good dog.

But in scampering across the air field, I had lost a precious birthday gift from my parents: my first "gold" (actually gold-plated) wrist watch. The entire kitchen personnel took pity on me and together we scanned the grass along the path I had run. And, lo and behold, one of the girls found my watch.

While I was amusing myself, blithely flying gliders, more momentous things happened around the world: the Allies had just begun their invasion of Sicily (July 10th) and in Russia Hitler's last offensive (July 5th to 12th) collapsed in a mammoth tank battle involving thousands of tanks (including the brand new but

Author was deceased at the time of publication.

[1] Is there a (1970) Broadway play *Sleep on the Runway* by Art Buchwald?

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_16,
© Springer International Publishing Switzerland 2015

untried *Tigers* and *Panthers*) outside the town of Prokhorovka near Kursk. According to latest Soviet sources, each side lost around 700 tanks in that single battle, which the USSR could quickly replace (thanks in part to American help) at a rate of up to 300 tanks *a day*.[2] Many gasoline-powered *Panthers* went up in flames because of ruptured fuel lines. Not a few tanks (on both sides) were lost to collisions because they couldn't see each other in the dirt storm kicked up by the unprecedented density of tanks on the battle field. The Germans had fielded 2,700 tanks against 7,000 Soviet tanks—including the legendary *T34*. (Another weapon of great renown was the German 88-mm gun, originally designed as a *flak* (*Flieger Abwehr Kanone*) but used most successfully as an armor-piercing gun.)

The German armed forces never recovered from this blow. Yet *before* the "Battle of Kursk," Hitler had declared: "*Zitadelle* (the code word for Kursk) would demonstrate to the entire world the enduring prowess of the *Wehrmacht*"—even after the "setback" in Stalingrad, which, after all, had happened in winter—as if the Russians didn't suffer from the cold as much as any other human being. (Most Soviet experts agree that their victory at Kursk was in fact more important than Stalingrad in winning the war.)

Once I had heard about Prokhorovka, I always wanted to visit the place to see how the Soviets commemorated this historical battlefield. Then I found out that I had, in fact, been in Prokhorovka before: in August 1959 on a night (actually 2 nights) train to Moscow from Ordzhonikidze in the Caucasus (now Vladikavkaz again as under the Czars).

From the same train I did see Izyum (south of Kharkov) which still showed the scars from the ferocious battles fought there 16 years earlier.

In August 1959, after a long drive through the Caucasus from Tblisi to Ordzhonikidze, we caught a glimpse of the beautiful Mount *Kazbek* (5,047 m). Our driver was kind enough to stop the car for about 20 min on the Georgian Military Highway when, suddenly, Kazbek was revealed, for just a few seconds, in all its towering grace.

In Ordzhonikidze (named after one of Stalin's henchmen, Sergo O.) I asked our German-speaking translator (most of our guides spoke English) whether the Germans had actually reached Ordzhonikidze. "No, but they came close." I later discovered that they had penetrated as far as Mozdok some 90 km to the north (on the Baku-Moscow rail line).

I also saw places like *Reichssender Gleiwitz* where the SS had staged a fake "Polish" attack on German soil on 31 August 1939 that gave Hitler, the next day in a speech at the Reichstag, the pretext for starting World War II: "*Seit 5 Uhr 45 wird zurückgeschossen*" (since 5:45 we are shooting back; actually it was 4:45 when the "return" fire was started—by the German battle cruiser *Schleswig Holstein*, shelling Hela near Danzig/Gdansk).

[2] When Hitler was told this number, he became very angry: "How could the Russians of all people produce that many tanks! But it was the truth.—At the same time, on the other side of the globe, in San Francisco and Sausalito, Henry Kaiser assembled one big *Liberty* ship every 24 h.

Another place I visited was the spot where the advance of the German armies was fought to standstill by the defenders of Leningrad. It had to be somewhere between Peterhof Palace and the outskirts of Leningrad. I asked my driver whether he knew and he proudly pronounced "I *fought* on the Leningrad front." He stopped at a little brook besides which there was a stone monument with a big red star and the inscription "At this point the fascist hordes drowned in their own blood" (or something to that effect).

Around July 19th I visited my father in his Blankenese apartment near his office (*Luftgaukommando* XI), just north of Hamburg, not knowing that I would never see old Hamburg again: only a week later much of Hamburg would be smoldering ruins. My father's prior assignment had been *Luftzeugamt* M (M for Moscow!). But they had to settle for Minsk, where my father was ordered to watch a mass-execution of "partisans." (This was Himmler's great idea to inure people to the "necessary measures" that had to be taken during the war.) My father was very upset. He complained to his commanding officer who answered "don't get excited, Schroeder, people here are as cheap as black berries." My father was subsequently transferred out of Russia and was never promoted again and, in fact, discharged from the *Luftwaffe* in the summer of 1944, at age 51. (One of the worst things I ever said in my life was to my father when, after the war, he complained about his low pension—a result of his not being promoted after 1941: "You lost a war and people who lose wars shouldn't get *any* pension." What a horrible thing to say, especially in view of the reason for his pension being so low!)

His next, and final, job was as chief of the Iráklion air base on Crete. There he befriended the monks of a nearby monastery, which he visited with my mother after the war.

He sometimes enriched our meagre diet by sending little boxes with raisins which however did not always arrive intact. I remember my mother opening one such box and encountering a layer of peat moss, below which another layer of peat moss appeared . . . and so forth until she was scraping the bottom: nothing but peat moss!—The theft was later exposed as a large-scale scam at Athens air base where boxes with peat moss were always waiting to be quickly switched for boxes with more valuable goods.

But Sicily and Kursk were not the only debacles that befell the Third Reich in July 1943. Beginning on July 24th parts of the great city of Hamburg were turned to ashes in a ferocious firestorm with a loss of only 17 (out of 792) British planes during the first night—thanks largely to the tinfoil strips ("chaff") tuned to the wavelength of the German fire-control radars (*Würzburg*, 50 cm) thereby completely blinding them.

Also on July 24th the Fascist Grand Council convened in Rome to depose Mussolini in the wee hours of the 25th. Undeterred, the ex-*Duce* appeared at his desk in the Palazzo Venezia the next morning, whereupon the King of Italy had him arrested and bundled off in an ambulance: destination unknown. Mussolini was located a few weeks later in a remote military reservation (*Campo Imperatore* named after Holy Roman Emperor Fredrick II (1194–1250)) near *Gran Sasso*, the highest peak in central Italy. On 12 September SS-daredevil Otto Skorzeny

"liberated" the *Duce* in one of the most spectacular commando operations of the war and delivered him, now a mere puppet, into the presence of his *Führer*. (I remember the date because the next day, when the news broke, I was in Berlin for another phase of my fighter pilot career: the "courage test," meaning we had to box until all contenders were sufficiently bloodied. But how do I remember that it was September 13th when I stood in front of that news stand near the *Tempelhof* airport with the big headline "Duce liberated"? As I said on another occasion, once certain numbers enter my brain, they get stuck there "forever." I never made any notes.)

Before the month was out, Palermo was liberated, as were, shortly thereafter, on August 5, Orel and Belgorod in Russia, igniting the first of many gigantic fireworks over Red Square in Moscow in anticipation of final victory.[3]

But the Nazis, long divorced from reality, even after this disastrous July, hung on for another year and 9 months. They "stayed the course" and continued a war that was irrevocably lost on September 3rd, 1939, when Britain and France declared war on Germany following the unprovoked invasion of Poland 2 days earlier. Even the two top Nazis had queasy feelings on that day: after chief translator Paul Schmidt had finished with the British ultimatum, there was a long silence. Finally Hitler, absent his habitual bombast, said "*Was nun*?" (What now?). This question—if not an expression of utter cluelessness—must have been directed at his foreign minister, Joachim von Ribbentrop, the former Nazi ambassador to London, who had "promised" his *Führer* that Britain would not budge. But Ribbentrop kept mum. According to Schmidt's memoirs, Göring, the only other person present, looking out the window, said in an eerily prescient mood "If we lose this war, God have mercy on us." Bravo, Marshall Göring! So why didn't his Commander-in-Chief declare a "regrettable misunderstanding," recall his troops from Poland and preserve the peace? He couldn't, because he would have lost face. So, instead of losing one *face*, the world lost *millions of lives*. But that is the way of human history, driven by the inability to admit error no matter what the ultimate cost—a prospect even more frightening now in the nuclear age.

Or, as Hobbes had it: *Hell is truth seen too late*.

[3] According to only recently released Kremlin archives, "Generalissimo" Stalin was reading a *History of Greece* at the time, whence he got the idea of fireworks to celebrate great victories. (I find it hard to believe that Stalin was engrossed in ancient history at the cusp of World War II, but that's what the records say.)

Chapter 17
Radar

Manfred R. Schroeder

In mid-February 1943, around the time of Goebbel's infamous "total war" speech, I was drafted, barely 16 years old, to serve, as an air force auxiliary (*Flakhelfer*), with an anti-aircraft battery near Stettin on the Baltic Sea. This was to free the regular crews for service on the Eastern front in a futile attempt to make up for the loss of 300,000 experienced men in the battle of Stalingrad.

During unpacking, the battery commander noticed a book entitled *Funkmesstechnik* (radio measuring techniques). But *Funkmess* was also the German word for radar, so I was assigned forthwith to the fire-control radar of our battery, comprising six of the renowned 88-mm guns. There were a total of 18 batteries guarding a huge "hydrogenation" plant that converted coal into gasoline. (It was said that this single plant produced some 25 % of the German fuel that was not derived from oil—a scarce commodity.) Of course, we were "thoroughly" trained—not on a realistic large cluster of aircraft but on single slow-moving planes (*Junkers W34*) pulling air bags as targets. Our radar range data was so accurate that, combined with optical data for azimuth and elevation, we sometimes hit the bag on the first try—and were given the rest of the day off. Once or twice a week there was an alarm: approaching enemy aircraft. But they never got very close, and, after an hour or so, we could go back to our bunks. This got to the point that I never even got up during alarms.

But one night, 21–22 April 1943, matters got serious. I pulled my pants over my pyjama and wrapped myself in a greatcoat. By the time I reached my position at the radar, my screen was filled with a forest of "blips," barely 10 km away. How to pick *one* target in this mess? I selected one of the bigger blips more or less at random. But by the time my two buddies (responsible for range and elevation) had zeroed in on the selected target, it had disappeared and I had to select a new one. The same

Author was deceased at the time of publication

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_17,
© Springer International Publishing Switzerland 2015

scenario repeated itself over and over again during the entire night. Where did the vanishing targets go; they were not shot down: they disappeared even before our guns could fire a single round.

Pretty soon it dawned on me that the big blips on my radar screen were not really individual planes but the result of random interference of the radar reflections from *many* planes that would add up or cancel each other depending on the (slightly changing) distances between the planes. Thus, we were shooting at "thin air," random blips, all night. But what to do. I couldn't possibly run to the battery commander and advise him of the problem (and risking a court-martial). So we moved our servos as we had learned from our single-plane exercises so at least the (analog) fire-control computer wouldn't go haywire with random input data. Almost needless to say, we didn't shoot down a single plane all night. After the enemy planes had dropped their bombs (on Stettin it turned out) and left for home, our battery commander jumped on the earthen wall surrounding our radar and, arms akimbo, announced "Boys, I have never seen such smooth data!" (Never mind that we hadn't hit a single thing.) Of course, we had learned to aim our radar and "properly" track a target even without any real planes present.

The next day was our day off. But the busses weren't running. They were needed to transport wounded people, it was explained to us. Staring at our radar scopes all night, the thought had never occurred to us that it was a night of great suffering for many.

The Pölitz (now Police, in Poland) hydrogenation plant, in spite of its strategic importance, was never bombed until late in 1943. "Bomber Harris" (General Sir Arthur T. Harris of the Royal Air Force) had simply prevailed with his strategy of carpet bombing of German cities against the many Allied experts who saw a much greater potential for bringing the war to a speedy end by bombing selected targets relating to fuel supplies, rail communications, manufacture of ball bearings, and other German bottlenecks.

In early 1943, an Allied plane was downed near Rotterdam without the (navigational) radar self-destructing so that it fell into German hands nearly intact. *Telefunken*, the German electronics giant, repaired the *Rotterdam Gerät*, as the captured radar was henceforth known, and installed it on top of the Berlin Zoo bunker. Hermann Göring was taken aback when he saw a precise map of Berlin, including its lakes and rivers, on the scope of the reconstituted enemy radar. He rescinded his 1940 order of dropping radar research then and there. In his earlier opinion, the war had already been won (sounds like 2003?). Anyhow, who needed those "boxes with coils inside," his moniker for radar. It was manly courage that decided wars.

Göring then issued an order, in his capacity as president of the *Reichsforschungsrat* (Reich Research Council), for all young Germans with a background in electronics to volunteer for special radar training to catch up with the Allies. I, together with some 400 other 17-year olds, followed the call and reported for duty at the camp "Prinz Eugen" in the *Westerwald* on Saturday, 23 October 1943. In the next 6 months, we learned all about Maxwell's laws, German radars and radar receivers (preferred on U-boats), and Allied radars as far as known. (After the war I could still converse with American radar experts about

the details of some of their "secret" radars, for example that the American high-μ tube 6AC7 was replaced by the similar German tube EF14.)

I once measured the wavelength, λ, of our fire-control radar with a makeshift *Lecher* setup and was immediately reprimanded for "exposing" one of the *Luftwaffe's* top secrets. This was a ridiculous charge because the Allies knew perfectly well that λ was about 50 cm—as evidenced by their tinfoil strips tuned to the "secret" wavelength.

Early during the *Westerwald* I met Jobst von Behr who became a lifelong friend. We were both interested in mathematics (and still exchange problems 64 years later) and despised military discipline. I remember chatting with Jobst during a roll call. We were standing in the back row oblivious to the goings on around us when his name was called by *Oberscharführer* Snita. Snita had to call my friend three times (Behr, *von* Behr, *Herr von* Behr) before Jobst responded with a meek "here." Whereupon the *Oberscharführer* launched into one of his favorite tirades about these "gentlemen" who won't respond until addressed with all their titles. Jobst had to step up to the front, fall down on his hands and do pushups.

Snita was also partial to wine. I remember, when we had to help him with the "tasting," we became somewhat "tipsy." In the subsequent class (*Deutsch*, if I recall correctly) my head was swaying. The teacher (Dr. Sieg) must have noticed this and he asked me something about our current reading (Hermann Löns: *Wanderer zwischen den Welten*). Being hardly able to speak, let alone give an intelligent answer, I replied in a slow voice: "*Das möchte ich gerne noch einmal lesen*." ("I would like to read the book again.") Just the right answer! The teacher was duly impressed.

After the *Westerwald* my next military assignments were with the Navy: first boot camp and then the Radar School on the Baltic island of Fehmarn and the Radar Observation Station in St. Peter-Ording on the North Sea. The German air force wanted to get rid of me (because I had missed boot camp as a result of a foot injury contracted by stepping on a glass shard on my last outing to a local lake with my parents and sisters before being drafted). As it happened, I had also missed the *navy* boot camp so on the day of inspection I was used as a *target* for hand grenade throwing exercises. I had to climb into a fox hole while everybody was throwing live practice grenades at me. One hit the earth wall behind me and rolled into my fox hole. I was practically sitting on the sizzling thing and jumped out as quickly as I could. Everybody, the admiral included, exploded in laughter: "This is just what the enemy wants you to do. You would be mowed down by machine gun fire by now!" I didn't care. All I knew was that I was still in one piece and a potential father.

In Fehmarn we learned about German counter measures against the tinfoil strips ("chaff") which had blinded the German radars ever since the attacks on Hamburg in July 1943. The new German radars exploited the Doppler effect to distinguish between slow moving strips and fast airplanes.

We also learned about the special radar problems for submarines. To preserve "radio silence," they would only use "passive" radar, i.e. radar *receivers*. This was a good idea until it was discovered that the French-made *Metox* receiver (named after

its inventor Metox Grandin) had a little wire smuggled into its circuitry that connected the local oscillator with the U-boat's search antenna. This was communicated by French resistance to the Allies which thereupon could switch off their own radars and just listen to the radiation from the German sub. This was a double blow to the subs: they were no longer forewarned by the approaching radar of Allied planes and had their own position unwittingly betrayed.[1]

Another trick, invented by the American physicist Luis Alvarez (later of dinosaur-extinction fame), exploited the relation between distance and radar intensity: The inverse square law that governs gravitation also describes the falloff of radar power with distance. This simple fact was exploited by German submarines during World War II. By measuring the increase in radar intensity, they could gauge the rate of approach of an enemy plane and dive undersea for safety before the plane could attack.

This tactic worked very well for Grand Admiral Karl Dönitz until Alvarez had a foxy vision, code-named *Vixen*. Alvarez suggested reducing the radar power so that it would be proportional to the *third* power of the range to the submarine. Thus, while the plane was approaching, the power incident on the unsuspecting U-boat was actually *decreasing*, giving the false impression that the radar plane was flying *away*. A grand idea indeed! (For the attacking plane, however, the received radar power reflected from the boat would still increase as it closed in.)

One *German* radar invention were the highly directive dielectric finger antennas mounted on a revolving turntable under a kind of plexiglass cheese bell. These antennas fed the U-boat's radar receiver but had the bad habit of breaking off in an Atlantic storm. Then a sailor discovered that the best way to repair (and protect) these fingers were condoms. Henceforth all German submarine crews had to carry an extra supply of condoms as a regulation wartime outfit.

At the end of the war, I found myself with coastal radar in Holland, huge installations that could pick up a plane over distant London once it was 200 m above ground. The Dutch navy officers, who had missed radar while in hiding during the war, were impressed and wanted us to repair them for their use in coastal shipping. But the Allies insisted on our dismantling them being afraid (in the summer of 1945) that the Soviet Army might advance to the coast opposite Britain and use them against their former allies.

Once we had a visit of some ten higher-up Dutch officers (we counted a total of 32 golden sleeve rings) and a dispute developed as to the distance of a target, a fishing trawler: our (freshly calibrated) radar showed 7 km, but one of the older officers, a former ship's captain, insisted that it was only 3 km. Neither side would

[1] According to other sources, the *Metox* saga was planted by the Allies to divert attention from the real reason for the increased U-boat losses, namely, the introduction, in March 1943, of a new (10-cm) search radar (H2S) which *Metox* couldn't detect.—The *Metox* was superseded by, among other radar receivers, the *Naxos* and the *Naxos ZM*, with an antenna which rotated at 1,300 rpm and with which I had a personal encounter in January 1945.

give in until one of the Dutch officers reminded the captain that he was looking at the sea from a high dune (Noordwijk aan Zee) rather than from a shipboard and what looked to him as only 3 km away could well be 5 km or more.

Toward the end of the war, we were planning to escape from Holland to neutral territory (Spain, Portugal?) by commandeering, if need be by force of arms, the launch of the Harbor Master of Ijmuiden. To be able to leave the port, we had to make some telephone calls to have the anti-submarine barriers (chains) lowered. It worked!—The chains were, in fact, removed. But someone in the Harbor Authority must have become suspicious and, upon double-checking, was able to expose our ruse and the chains went up again. End of dream of a free post-war "cruise"—a potentially very dangerous situation (the port exit channel was also protected by two medium-caliber cannons).

Of course, everybody was happy that the war was over. The only reason we youngsters were sorry that Germany had actually lost the war was our fear that there wouldn't be any more nice German *Tanzmusik* on the radio. But it didn't take us long to discover that the Allies were broadcasting even nicer jazz ballads and swing music. I remember how we fell in love with "Don't fence me in," "Sentimental Journey," "American Patrol" and many other then very popular tunes. In the same building as ours, with walls adjacent to our living room, some elderly officers were quartered. They congratulated us on how well-behaved we were—except, occasionally, all hell seemed to break loose in our quarters. Why? Did we tell them about the American jazz that drove us crazy? Maybe. After all, the war was over and it was no longer forbidden to listen to foreign radio stations. (But we certainly didn't tell the gentlemen from next door that one of us—a navy lieutenant, no less, in his nightshirt—actually jumped on top of a wardrobe and crowed like a rooster.)

Another circumstance that lightened our burden of having lost the war was that we were sitting on endless supplies of molasses (in oil barrels!). We soon discovered how to ferment the stuff, distil it and collect it in a beaker, drop by precious drop. Needless to say, the "distilled" nights—brewing was of course illegal, so we couldn't do it during the day—were among our happiest. And what we couldn't consume on the spot, we brought to a boil, poured it over ground coffee and produced the best *Kaffee-Liquör* I have ever tasted.

We also had beautiful carpets in our "camp" quarters. One day, after a new set of guards had taken over, one of their sergeants requisitioned one of our gorgeous "groundcovers" for himself. When he came back an hour later, wondering why we hadn't delivered the carpet as ordered, the "object of his desire" had disappeared. (We were hiding it in the attic.) And the sergeant, taking a very belligerent pose, proclaimed "*Als het tapijt niet onmiddelijk ter voorschijn kommt*, *werd u gearresteered*!" (If the carpet doesn't reappear immediately, you will be arrested!) I was just standing there smiling. (I liked to hear Dutch.) Whereupon the sergeant turned in my direction: "*En u ook*!!" (And you too (will be arrested)!!)—The *En u ook* later became a standing expression in my family.

17.1 Outside the Geneva Conventions

Incidentally, at the end of the hostilities in May 1945, we were classified as "capitulated personnel"—outside the Geneva Conventions.[2] But in our case, in Holland, it meant Canadian army rations and even home leave, which allowed me to ascertain that my parents and sisters had survived the war unhurt and our house in heavily-bombed Hamm was still standing.

Indeed, as I approached our house, after having traversed plains of rubble, I didn't dare ring the door bell. Who would answer? Were my parents still alive and had they managed to flee from distant Silesia? So I walked around the house and saw, through a basement window, a large store of apples that maybe only my mother could have scrounged up in those desperate days. I finally entered the house through a back door into the kitchen. And I didn't have to ask who was still alive. They were all there in that one room: my parents, my sisters and even my grandmother. I was so overjoyed, I started crying. But the rest of my family, while certainly glad to see me return from the war hale and healthy, didn't seem as emotional as I. Why? They had just received a postcard that I was on my way home.

The Canadians to whom we had capitulated were boys about our age (18–20 years) and they commiserated with us that, as radar people, we had never really fired a weapon. So they returned our machine guns (and even a bazooka!) to us for one afternoon so we could have our little private war in the dunes (with tracer ammunition and the rest). (This exercise in silliness, incidentally, was not without its dangers. I remember one bazooka going off right in front of us. Nobody was hurt but all of us were blackened like chimney sweeps.)

Understandably, the higher-up brass were not amused when they got wind of the happenings.

Since we were sitting on plenty of now useless ammunition, the lure of making our own rockets was always close. And, indeed, some of our contraptions, made from wooden sticks (broomsticks) and sheet metal (from army ration tins), flew 100 ft high in the air. The only bad thing about it was that we actually *ate* some of the spaghetti-like ammo: it was sweet and tasty. But too much gave you a headache. How silly can you get?

In April 1947 I was repatriated from Holland where I had remained voluntarily since the end of the war. Originally, they had picked six older guys who didn't know much about the inner workings of radar and who were eager to be reunited with their wives and children. But three of us, all below twenty, wanted to stay in Holland and our "petition" was granted. The six older men were of course jubilant and so were we. (We were afraid of the makeshift PoW camps on the banks of the Rhine, where many unfortunate prisoners perished as a result of hunger and poor

[2] At the end of the war, General Eisenhower suddenly found himself with some three million German captives on his hands and he felt he couldn't possibly treat them all under the Geneva Conventions. So they were placed outside the Conventions.

sanitary conditions entailing typhoid and other epidemics.) Best of all, as already mentioned, we were treated by the Allies as "capitulated personnel" outside the Geneva Conventions. I know it sounds unreal after 9/11 and Guantánamo but we enjoyed many benefits including excellent food, good accommodations, and even vacations (home leaves). And of course the work was interesting: being able to demonstrate to the Dutch naval officers who had stayed in Holland during the war how radar worked. In fact, they wanted to keep some of our radars intact to monitor coastal traffic. But, as already pointed out, the British were unrelenting: all radars had to be dismantled.

After the Canadians left, a company of Nederlandse Stoottroepen (Dutch Assault Troops) became our guardians. I befriended corporal Pieter Broeks with whom (as a "prisoner"!) I travelled to Haarlem and other places. In Haarlem I became close friends with Nico Bartels who owned the radio repair shop "*De Onderduiker*" (The Underground Diver). He taught me (and gave a living example of) democracy and friendship between nations. During the war he had been a member of the Dutch underground (called *onderduiker*). This was also the time I started to study *Nederlands* (Dutch).—I visited both Pieter and Nico and their families after I was released from my Ersatz-Guantánamo on 15 April 1947.

Our next guardians were from the Palestine Regiment of the British Army. We were probably among the few Germans inside barbed wire with the Jews on the outside. We had a very fair relationship, all in English—until the last day when the Jewish sergeant-major concluded the daily roll call with some words of appreciation in perfect German.—Once, when a wristwatch of one of my buddies was stolen, captain Neumann (originally from Berlin) had his troops lined up so that the German prisoner could identify the thief (who was duly disciplined). A fine example of fairness of victor toward the vanquished! Or as Churchill wrote in 1948 in his *The Second World War* (Moral of the Work. Vol. I, *the Gathering of the Storm*):

> In War: Resolution. In Defeat: Defiance.
> In Victory: *Magnanimity*. In Peace: Good Will.

Once the radar job was completed, we were assigned to ordnance dumping duty in the North Sea—a dangerous job because the guards liked to shoot at the (punctured) ammunition canisters before they sank and sometimes there was an explosion and one or a few prisoners were killed.

My main job, though, for which I had volunteered, was to build an electrocardiograph for the Dutch Navy which I felt I could do ("slam dunk"). But to my shame, I must admit that I never got the contraption to work properly—there was always too much noise (and hum) and the EKG records were hard to read. (Now I would use tubes heated by d.c., as opposed to a.c., for the heaters in the preamplifiers.)

Buying the proper cameras for recording the EKG required some travel to different parts of Holland and pretty soon, prisoner Schroeder travelled *solo* without accompanying guards. Once, on a "private" trip through Holland, I opened the door

to a railway compartment full of our camp guards. I quickly slammed the door shut before anybody could recognize me.

On another occasion I shared the compartment with several ladies who were wondering why the train wouldn't leave the station. "Had the train schedule changed?" I felt my Dutch was already so good that I could join their conversation and said: "Mij zijn helemaal geen wijzingen van de dienstregelingen bekent" (I haven't heard of any changes in the schedule). Whereupon they looked at me in great astonishment. Had they recognized my "Prins-Bernhard" accent or were they only surprised by my butting in? I never knew, but I kept my mouth shut from then on.

Another illegal activity indulged in by many camp mates was smuggling. I remember once hiding, under my greatcoat, a medium-size munitions box with ground coffee and scented soap (in the same box!) for my family in Germany. As soon as I had passed the camp guards, I panicked and started running whereupon one of the strings that held the box snapped and the box fell down. But it was caught at the last moment by a second string which almost strangled me because it went around my neck.

On my last day in camp I put all my belongings into my *Seesack* (sea bag) and had it taken to our home-bound vessel by an ambulance which some buddies were using to smuggle pots and pans and other goods that were scarce in Germany. They pretended that the patient in the ambulance was highly contagious—a ruse that worked. But when I came to the inspection point our captors were wondering why I had so few belongings. They immediately smelled a rat, arrested me on the spot and confined me to a small cell. What especially raised their suspicion were my mysterious sketches and calculations of multivibrator circuits that I had kept with me in my briefcase: they were convinced they had caught a spy. (Later these calculations came in handy when I was designing pulse generators to test early TV sets at *Grundig Radio*.)

The happenings during the next 10 min were almost beyond belief: I told one of the Dutch prison guards, whom I had befriended a bit before, to please run to the ship and warn my buddies of the impending disaster. I can still see the guard sprinting past the suspicious inspectors on their way to the boat for a thorough check. Thanks to my friend nothing was found and I was released (without apologies).

In May 1947, I was discharged from the German navy, receiving one pair of long johns and a very nice blanket for my services to the Third Reich. (The Nazis had of course promised everyone a farm in Kazakhstan, but I was never very fond of farming anyhow and the blanket came in very handy: my mother had it dyed a fashionable dark blue and turned it into a pretty overcoat for my older sister.)

Chapter 18
Lucky Breaks

Manfred R. Schroeder

18.1 The Düsseldorf Murderer

My first lucky break happened when I was about 4 years old. I was visiting my aunt Lydia in Düsseldorf and, after a downtown shopping trip, I wanted to stay behind to watch the traffic, directed by a policeman, at a busy intersection of the Grafenberger Allee. My aunt agreed if I promised not to follow the "Düsseldorf murderer," also known as the "Düsseldorf Vampire," who was then killing, mostly young boys and girls, in droves in the neighborhood. A few minutes after she had left me, a middle-aged man with a mustache, wearing a boater, approached me and invited me to go with him. (I forget where and what the enticement was-sweets, or an interesting sight?) I would have followed him because he seemed very nice, but remembering my aunt's warning I declined, telling him I was very intent on watching the traffic. Then, significantly, he turned around and walked back in the same direction from which he had approached me. He had hardly disappeared when I ran to my aunt's flat proclaiming: "I saw the Düsseldorf murderer!" But, of course, nobody believed me. Not much later, when he was caught and his picture appeared in the newspapers, I recognized him. His name was Peter Kürten. He confessed and was guillotined on 2 July 1931. In an interesting twist, he first confessed to his unsuspecting wife so that, on her reporting him to the police, she could collect the substantial reward.

When questioned by the police psychiatrist, Dr. Karl Berg, whether he had any regrets about his life of perpetual murder, Kürten's response came out almost like an epitaph. . . "I have no remorse. As to whether recollection of my deeds makes me feel ashamed, I will tell you. Thinking back to all the details is not at all unpleasant.

M.R. Schroeder
Author was deceased at the time of publication.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_18,
© Springer International Publishing Switzerland 2015

I rather enjoy it." Kürten expressed his last earthly desire as follows: "Tell me," he asked Dr. Berg, "after my head has been chopped off, will I still be able to hear, at least for a moment, the sound of my own blood gushing from the stump of my neck?" He savored this thought for a while, then added, "that would be the pleasure to end all pleasures." (From crimelibrary.com) A severed head is, in fact, said to remain conscious for about 1½ minutes.

18.2 An SS Draft Attempt

In July 1944, I was emerging from an extended hospital stay (rheumatic fever) and ordered to convalesce for the next 6 weeks. While walking the streets of our home town, Liegnitz, seemingly hale and healthy (and 18 years old), the local SS-recruitment office caught sight of me and sent me a draft notice. Now this was outright dangerous. The war was ending and the SS needed more cannon fodder. (At the beginning of the war the SS would only take volunteers, true-believers. But in 1944, things were becoming desperate.) My parents and I decided then and there to put me on a night train to my old radar group, so that the next day my father could write the SS that I was no longer in Liegnitz and had rejoined my old air force unit.

The train which I took for my precipitous escape was so crowded with soldiers on leave from the Eastern Front that my parents literally had to push me through the door and my mother cried: "We are pushing Manfred into the war, into his grave!" But the sudden escape probably saved my life.

18.3 The Loose Hand Grenade

Toward the end of the war, February 1945, some buddies and I were transferred from Hoek van Holland to another coastal radar at IJmuiden. All ground traffic was limited to the dark night hours because at daytime the sky was completely dominated by allied fighter planes (*Jagdbomber*). In addition, gasoline was so scarce that many German military trucks ran on wood (little wood blocks that were shoveled into a stove that produced a gaseous fuel). Every so often, the truck driver had to stop for more wood to be fed into the oven. At one stop, near Leiden, the driver was checking, with a flash light, whether the oven was completely filled. And to his surprise (and ours) he was fishing a live hand grenade from the oven. "Whose is this?" he wanted to know. Upon checking, I discovered that it was one of my grenades that had fallen out from my loosened belt and got mixed up with the woodblocks. And we had all been sitting with our backs to the stove to keep warm during the cold winter night! My third escape from almost certain death—not counting my nearly botched birth when all in attendance thought I was dead on arrival and received baptism *in extremis* in the nearest available sink.

18.4 A Leisurely Trip Through Holland

Another lucky event happened to me on my very first day in Holland on the way to my new radar unit. At the end of our training with German and (captured) Allied radars on the island of *Fehmarn* in the Baltic I was picked as the "leader" of a group of six classmates who were assigned to Navy coastal radars on the Dutch North Sea coast. The first big city we hit was Hamburg and those of us who had relatives or friends in or near Hamburg wanted a day off. I said "O.k., have fun." The same thing happened 2 days later at Bremen. By the time we arrived at the first Dutch city (Groningen) we were already 4 days behind schedule. We all found Groningen so charming that we all decided to spend an extra day there. The same urge for sightseeing gripped us in Apeldoorn and Amersfoort, where we even got accidentally separated from each other—a dangerous occurrence, given the ever present military police, always on the look-out for deserters. When we finally arrived in the Hague we were a full week behind schedule. And then I made a big mistake. Upon arrival at 8 a.m. (the trains were running only during dark), I called our new headquarters ("Kommando Scherer") to tell them we had safely arrived. But then we all decided to use the day to explore the Dutch capital (none of us had ever been abroad). Two friends and I wanted to take a close look at Dutch books. Although we were "sailors" (actually Navy cadets) we were carrying rifles: Holland was considered enemy territory, infested as it was by the Dutch underground. So before going up on the ladder to reach the upper shelves, I leaned my rifle against one of the book cases and later promptly forgot about it. (We had never carried rifles before.) Only half an hour later I discovered that everybody had a rifle—except me. I was scared to death: losing your weapon in "enemy territory" was the cause for the most severe court martial prosecution for sabotage! But I was rescued by the kind book dealer. When I entered his store again, he reached under the counter near the cash register and handed me back my sorely missed weapon. (He was quite terrified too, I think, because being discovered in possession of a lethal weapon was, well, lethal to the poor occupied people.)

When we finally reached our radar headquarter in Scheveningen, all hell broke loose: why did it take us a full day to walk from the Hague to Scheveningen? In the ensuing commotion, made outright dangerous by a machine-gun wielding, drunken captain, nobody discovered that we had just "wasted" a *full week* at the fatherland's expense.

18.5 My Pants Stuffed with Cash

Another lucky escape, though less deadly but fearsome nevertheless, happened while I was a student in Göttingen. I had mentioned to a friend at the lunch table (shared with a young woman unknown to us) in the *Mensa* on Wilhelmsplatz that I had seen law student W. with yet *another very* pretty girl. During the lunch I had

also mentioned the fact that I was in dire need of some extra cash, like 100 marks. When we departed from the table, the unknown woman said "Thanks for the compliment. That woman was me." A few days later, after a round of tennis, while dressing in the locker room, I found a handful of crumpled 10-mark bills in my pants pockets, a total of DM 100, just the amount I needed! But I immediately reported my find to the president of the club who happened to be in the club house. Not 10 min later a fast Porsche with W. at the wheel came roaring up proclaiming that "100 marks had been stolen from him in the locker room; could he have it searched?" The club president said that was unnecessary; Schroeder had already found the money—Can anybody imagine the situation I would have been in if they had found the money in my pockets? Expulsion from the university and worse!

18.6 An Invitation from Bell Labs

My next lucky break occurred in 1954, long after the war's end, shortly after my Ph.D. (in physics, at the University of Göttingen). I wanted to join the renowned Bell Laboratories in Murray Hill, New Jersey. But my professor, Erwin Meyer, told me in good faith that Bell wouldn't take foreigners. (He had one of his students turned down in 1938.)

But then, in March 1954, William Shockley, co-inventor of the transistor, came to Göttingen on some solid-state business and he let it be known that Bell was looking for qualified physics students. I immediately went to Meyer and the next day he wrote a letter of recommendation to one of Bell's research directors (Winston Kock). A few days later I was invited by Kock for an interview at the posh Dorchester Hotel in London. (The somewhat hilarious details of the interview are noted in my book *Computer Speech*.) The interview went well and after another 6 weeks I received an offer of permanent employment at $640 per month, which was five times as much as I could have made at Siemens in Munich (500 Deutsche Marks).

After finishing some ongoing projects at Göttingen, I traveled to Genoa to board the *Andrea Doria* (then still afloat) and arrived in New York on 30 September 1954 to begin a new life: a new country, a new language, a new job, my first car and, 3 days later, on my first foray into New York, to meet my future wife Anny, working at the Bulgarian desk of Radio Free Europe.

18.7 Kugelblitz

I was always afraid of lightning, so during a particularly violent thunderstorm in Berkeley Heights (on 26 July 1988) I repaired the basement of our home on Sutton Drive. The house had recently acquired aluminum siding, so I felt quite safe in my "Faraday Cage," especially in the basement. (As everybody knows, thunderbolts

come from on high.) Then one lightning landed in a tree standing next to our house and, I couldn't believe my eyes, a globular lightning (*Kugelblitz* in German), about 15 cm across, wandered between my head and the ceiling before it disappeared with a "poof" about 3 m from where I was hiding. Even as a physicist I had never believed in *Kugelblitze* and I was convinced the deadly ball of fire I had seen was a paranormal perception. But when, after the storm, I followed the path the ball had taken, I discovered that it had run along the wires connecting the door bell with a little transformer in the basement. And that transformer was completely burned black. So that much was for real. But where did the darned *Blitz* thing come from? Well, the doorbell was of course not covered by the aluminum siding and the lightning had discovered that little opening to visit me in the basement. If my head had been in its path, it might have resembled the ex-transformer. . . .

Chapter 19
The University of Göttingen

Manfred R. Schroeder

19.1 Student Days

After having been released from my captivity "outside the Geneva Conventions," I had to face the question of what to study and where. I was always strongly attracted to mathematics but my grandmother thought I should become a physicist. Fine, but *where* to study physics in those days? In Göttingen perhaps, where several great physicists were in residence, Nobelists Max Planck, Max von Laue, and Werner Heisenberg whose quantum mechanics we youngsters were all eager to understand. We were of course duly impressed by Albert Einstein's relativity but it contained no great mysteries for us. We had accepted that space was curved, simultaneity and the "ether" were exposed as figments of the imagination and mass and energy were shown by Albert to be equivalent: $E = mc^2$.

But quantum mechanics we *didn't* understand and so I attended, in one of my first semesters, a course by Heisenberg on the shell model of the atomic nucleus—and didn't understand a word: Spin-orbit coupling! What is *spin-orbit coupling*? It's basic to the shell model, proposed by the former Göttingen physicist Maria Goeppert Meyer and two colleagues in 1949, which explained the "magic numbers" of nuclear physics, the existence of extra stable nuclei. After a few lectures, I dropped out and decided to learn some basic physics first. (Goeppert Meyer was only the second woman, after Marie Curie, to win the Nobel Prize in Physics (in 1963). Lise Meitner, who, with her nephew Otto Frisch, had explained nuclear fission, was passed over in favor of her male collaborator, Otto Hahn, who collected the 1945 Prize in Chemistry.)

Later I met Heisenberg on a more personal basis. I was asked to guard his house on Merkelstraße and his in-laws (the Schumachers) while the family were taking their annual vacation in the Alps. The house had been broken into on more than one

Author was deceased at the time of publication.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_19,
© Springer International Publishing Switzerland 2015

occasion and valuable musical instruments had been stolen. (The Heisenbergs were addicted to music—see Anna Maria Heisenberg-Hirsch's collection of Heisenberg's letters to his parents *Liebe Eltern*. Their house concerts were sometimes attended by Max Planck who lived nearby and who had wanted to become a concert pianist before settling on physics.)

Another reason for my moving in with the Heisenbergs was that around 1946 suspicious characters had been observed casing the house. The border with the Soviet zone was only 20 km away and rather penetrable immediately after the war. The fear was that these guys were Russian agents preparing to capture Heisenberg and exploit his knowledge for the Soviet atomic bomb project. I never believed this cloak-and-dagger story but his son Martin, a biology professor in Würzburg, in a recent conversation, confirmed the story, except that the suspicious characters were from a *Western* secret service.

One evening, when I turned on the alarm, it immediately went off with a howling noise. In other words, one window wasn't properly closed. But I couldn't find the fault, so, to the Schumachers' dismay, I turned the alarm system off, to have it repaired the next day by the Max-Planck Institute. Needless to say, the poor Schumachers didn't sleep a wink all night, especially since their little dog had behaved in a funny manner all day (was he poisoned?). Anyway, the Schumachers were convinced that this night disaster would strike. But I slept like a log. This story of my alleged insouciance got around to Mrs. H. and she was not amused.

One benefit that we drew from our close association with the Heisenberg family were the CARE gift parcels from the USA that were accumulating in the attic of Merkelstrasse 18. Every so often, one of the Heisenberg maids (Marlene and Ortrud) would divert one of the parcels to us—a now scarcely imaginable joy: Hershey chocolate syrup, *white* flour, powdered eggs, and other goodies not available to the ordinary German in 1947. Erhard Scheibe, Jobst von Behr, and I baked *Topfkuchen* (pound cake) with which we could greatly impress our girl friends (Lotte Wengler among others).

On the occasion of Heisenberg's 100th birthday, I had been invited to give a talk (at the Lower Saxony State Library) reminiscing about my encounters with the great physicist (see my homepage [http://www.physik3.gwdg.de/~mrs/] for pictures of the family and the text of the talk).

Before one could even be admitted to the university, with so many young men returning from the war, one had to pass an entrance examination. I applied to the theoretical physicist Richard Becker and went to his office in early July 1947. But before I could see Becker, I was intercepted by his assistant Günther Leibfried who asked me: "What did you do during the war?" "I summed trigonometric series" (of course, he didn't want to hear any battle stories). He asked how I did it and I answered: "By using the addition rules for $\sin x$ and $\cos x$," which is of course very cumbersome. "So, you haven't heard of the complex exponential function? Using *it*, you would have obtained a geometric series that is very easy to sum." I knew enough mathematics to know he was right, but I had missed it. How stupid can you be! I concluded that my chances of being accepted were nil when Professor Becker

appeared in person. "What did you do during the war?" The exact same question, but I avoided mentioning summing series like the plague. Instead I answered "I analyzed electromagnetic circuits." "Go right ahead," Becker said and I delved into a long spiel involving, electrical fields, magnetic fields, radiation, absorption and more. After about 15 min Becker said: "Thank you. That's enough."

Scared as I was at that point, I thought he meant "that's enough nonsense." But I wanted to know for certain if I had flunked the entrance examination. So I asked, timidly, "Professor, what are my chances of being accepted?" My whole future was in the balance. And Becker answered with something that I will never forget: "If *you* are not accepted, nobody will be admitted." Thank you, Professor Becker! (Later, the high opinion in which Becker held me, went a little to my head and I became somewhat lazy.)

Before becoming fully accepted at Göttingen I had to hop over the hurdle: I couldn't produce a valid *Abitur* (high school leaving diploma)—a *sine qua non* for being registered at a German university. The administration gave me 2 months for producing some documentary evidence for my claim that I had indeed passed the *Abitur* but that the papers were lost in the final days of the war. (My family had fled from Silesia in February 1945 just before the onslaught of the Red Army.) Luckily, after a week or 2, I met an old classmate, Leo Klingen, who knew the address of our last homeroom teacher (and director) of my last high school, Dr. Sieg. He kindly provided me with a minute description of our Abitur, including grades and the names of the school district supervisor, etc. When I took his letter to the university administration, the official, after perusing the letter a long time, said: "That's enough." Again I was so timid that I asked what he meant. Enough of my baloney? No, enough to be fully accepted!

Best of all: all my buddies, who had passed the exact same type of *Not-Abitur* (wartime Abitur), had to go back to high school for a year to get a real Abitur. How lucky I was to have lost my documents (and that I had found my old teacher and that he was so kind)!

Another problem was finding a room (few dormitories then). My friend Jobst von Behr was offered a small room in Heisenberg's house, which was not under the control of the city housing department. Jobst made the room assigned to him by the city available to me and I gratefully accepted it "for the first two weeks." But, although it was very small and unheated, I kept it for the next 7 years, through my Ph.D. and emigration to America. (The rent was 12.50 marks/month, which I later "rounded off" to 15 marks.)

After just three semesters I felt I was ready for the *Vordiplom* (sort of B.S.), except for the required chemistry. So I went to Professor Becker and told him "I am ready for the Vordiplom, except that I didn't go to the chemistry lab." (It stank atrociously.) Becker said: "O.k., you can go ahead without the lab course, if you otherwise know enough chemistry." "But I have never been to a single chemistry lecture!" I countered: (I hated chemistry.) Becker: "Hmm, we never had a case like that before." "But, professor, I have attended several extra courses in mathematics," (which I liked). Becker: "O.k., why don't you do your Vordiplom in mathematics then?" "But, professor, I want to become a physicist!" "Well," Becker said, "on the

day of the examinations you are a mathematician and the next day you are a physicist again." How simple things used to be in the late 1940s (with the *right* people in the *right* places)!

Becker was also amenable to our complaints about the "lecturing style" of one of our professors. Becker actually sat in on his lectures and completely agreed with our criticisms. But what could he do? For better or worse: Freedom of Teaching—good or bad—reigned supreme!

After Becker retired, he was succeeded by Friedrich Hund, one of the pioneers of quantum mechanics and its application to molecules. On his first day in office Hund wanted to see the *Schriftverkehr* (correspondence files) of the institute. But Becker's old assistant Leibfried told him "There are no files; we threw everything into the waste basket." Hund, a stickler for formalities and forms (besides mathematical formulas) had the whole *Schriftverkehr* (spanning two decades!) resurrected from the university's central files. Talk about German thoroughness! But Hund was a great teacher and, even as a professor in my own right, I attended several of his fascinating lectures on the history of physics.

19.2 Ringvorlesungen

Once, while I was in the States, I got a call from Konnie (Kaufmann): would I participate with Hund in a *Ringvorlesung* ("circle of lectures") to be called *Die Grundlagen der Physik* (the foundations of physics). I said I would if it was called *Zu den Grundlagen der Physik* (*On* the foundations of physics). Hund would cover Newton, Hubert Goenner would talk about "Einstein and Relativity," Lorenz Krüger on "Statistical Physics." But who would cover quantum mechanics? Someone said: "Schroeder." But someone else objected "Schroeder is an acoustician and not a quantum mechanic." Whereupon the great Hund said "Don't worry. Schroeder knows everything!" Yes, and Friedrich Hund knows even more.

19.3 Quantum Mechanics

This early *Ringvorlesung* was an instant success. I learned a lot of quantum mechanics (QM) while preparing my part. It is a weird subject and gets weirder and weirder the more you know about it. As Niels Bohr, the father of the atom model (1913) said: "If you sometimes don't get dizzy when you think about QM, you have not really understood it." Or, as Richard Feynman had it: "I think it is safe to say that *nobody* understands quantum mechanics." The most infamous paradoxes of QM were constructed in 1935 by Einstein, Podolsky, and Rosen. It leads to the inescapable conclusion that either physical actions can spread instantly, "ghost-like," over large distances or, worse, *there is no physical reality* as we have always

thought of reality (like, say, beams in the attic on which you can bump your head or myriads of other objects that surround us).

The only way out of these fundamental difficulties that I see is the realization that our understanding of reality is the result of our daily experience with macroscopic, tangible things. For the survival of our species, it was never important to see electrons inside an atom and we should not be surprised when our images of reality fail when we try to apply them in the world of the atom.

This *Ringvorlesung* became the forerunner of many more *Ringvorlesungen* originally inspired by Kaufmann and Otto Creutzfeldt but later taken over by others. Speakers included the London art historian Sir Ernst Gombrich, Anton Zeilinger from Vienna, and Karl Friedrich von Weizsäcker ("not a bad physicist" according to a *public* pronouncement by Hund in the presence of W.).

One more word about Becker. When in April 1933 the Jewish professors were sacked except—initially—those who had distinguished themselves in the Great War (1914–1918) such as Nobelist James Franck. Franck resigned his directorship of the institute for atomic physics voluntarily, noting that he could not, in good conscience, continue to serve a state such as the Nazis were then erecting. How many German professors sympathized with Franck? In my research (for a talk on Lise Meitner and nuclear fission) I came across just one letter—by Richard Becker, expressing his sympathy with Franck's courageous stand. Nor did anybody call in by phone. (There is now a bronze plaque in the main building of the university with the words of Franck's secretary, who was hovering in vain over the phone all night ". . .aber niemand rief an.")

One prominent physicist who suffered from the Nazis was Lise Meitner. She was Austrian by birth but could work in Germany until Austria was annexed by the Third Reich in 1938 whereupon she became automatically a German citizen and subject to the Nazis anti-semitic policies. She fled overnight with the help of her friend Otto Hahn ("Hähnchen"—little cock—as she liked to call him). Shortly thereafter, in Berlin, Hahn (and Strassmann) discovered nuclear fission—a discovery that shook the world (and of which Hiroshima/Nagasaki are perhaps not last evil manifestations). In 1945, Hahn was awarded the Nobel Prize in Chemistry for this discovery while Lise Meitner was all but forgotten in her Stockholm laboratory. But it was she (and her nephew Otto Frisch) who first recognized what was going on and coined the term *Kernspaltung* (nuclear fission)—a realization of Einstein's famous formula $E = mc^2$. I am proud to have known Frisch personally and to have been able to pay tribute to Meitner in a talk I gave in the Berlin *Staatsbibliothek* (see my homepage http://www.physik3.gwdg.de/~mrs/).

Emmy Noether, a leading mathematician of the twentieth century, was another woman scientist who was expelled from Nazi Germany. She went to Bryn Mawr in Pennsylvania. Her brother Fritz went to Tomsk in Siberia as a professor, where he applied for Soviet citizenship in 1937. Soon after, he was arrested by the NKVD (forerunner of the KGB) as a German spy. Four years later, on 10 September 1941, he was shot at a camp in Orel as the German army was approaching the area on their way to (defeat before) Moscow.

Later, for my Ph.D., chemistry raised its ugly head again. Yet, with the help of another professor (Erwin Meyer), I was able to circumnavigate that cliff, too. Meyer cautioned me: "We have some *wandelnde Gewissen* (walking consciences) in the faculty and it may be difficult. But I will bring up your request at the end of a long faculty meeting when everybody wants to go home and I will also mention that you are leaving for the States soon." (Never to return?) Next week Meyer told me that it had worked as predicted but that it wasn't right to totally skip chemistry. I agreed.

Ironically, 15 years later, when I became a professor myself, quite a few of my students were working on problems that involved a lot of (physical) chemistry. I quickly got some freshman chemical texts and my handicap soon disappeared. As is well known, there is nothing like being properly motivated to learn something!

The Vordiplom was relatively easy, with chemistry substituted by mathematics. But even in math I had a deficit: I knew next to nothing about differential equations, so I asked the examiner (Dr. Lyra) to please skip differential equations. But during the actual oral examination he asked me for the solution of $dy/dt = y$. I said, "But, Dr. Lyra, that is a differential equation and we had agreed. . ." "But it's a very simple one" he said apologetically. I said: "O.k., the solution is $y = \exp(t)$." After that minor skirmish, we settled on topics more to my liking.

With the Vordiplom in hand, I registered for the obligatory advanced lab course in physics, which didn't really stoke my interest—except when I had a chance to *umfunktionieren* (turn upside down) one of the experiments. For example, we had to measure the speed of light on a long cable (curled up in the basement of the physics building). The idea was to send a short electrical pulse through the cable and measure the delay with which it emerged at the other end. I thought, if I just connected the two ends of the cable through a feedback amplifier and cranked up the gain, the cable would start "singing" at a frequency which is simply the reciprocal of the sought-after delay.

This new approach to measuring the speed of light on a cable, which I considered almost self-evident, was rated a "major breakthrough." The next day I was offered a position at the *Dritte Physikalische Institut* for doing a master's thesis. Thus, while I had started out in theory, I was now an experimental physicist at a renowned institute that, 4 years later, smoothed my path to the USA and the prestigious Bell Laboratories. Some students who were much smarter than I—one even got the Nobel Prize (Horst Krömer)—who stayed behind in theory never made it to Bell Labs, because their professor wasn't as well connected as my new boss, Erwin Meyer.

19.4 The Dritte Physikalische Institut: Historical Antecedents

Around the turn of the twentieth century, the well-known Göttingen mathematician Felix Klein proposed to promote the pursuit of applied sciences at German universities. He was further inspired in this effort by a visit to the 1893 Chicago World

Exhibition—as an official representative of Kaiser Wilhelm II—and subsequent visits to several American universities, which had a strong tradition of fostering applied sciences and engineering. But Klein's attempt to enlist the *Technische Hochschule Hannover*—let alone any university—in his endeavor failed miserably. Finally, he succeeded, with support by German industry (Böttinger), to establish an *Institut für Angewandte Mathematik und Mechanik* and an *Institut für Angewandte Elektrizität* (1905) at Göttingen.

These institutes, which had been home to such famous aerodynamicists as Ludwig Prandtl and Theodor von Karmann, morphed, in May 1947, into the *Dritte Physikalische Institut* under the leadership of Erwin Meyer, who had been professor at the *Charlottenburg Technische Hochschule* in Berlin. With Meyer, room acoustics, including concert hall acoustics, microwaves, underwater sound, and acoustics in general came to Göttingen.

The *Institut* became soon well known through several successful large projects: the *Jahrhunderthalle* in Höchst, the *Beethovensaal* in Bonn, the Hamburg *Staatsoper*, and the *Lower Saxony Landtag* (parliament) in Hanover. Especially noteworthy was the ingenious utilization of electroacoustic means utilizing the *Haas-Effekt*, related to the "precedence effect."

In fact, the *Institut* became well known internationally by the discovery, stimulated by Meyer, of the *Haas-Effekt* by Helmut Haas in 1950. This property of human auditory perception says that even an *amplified*—but suitably delayed—sound will not affect the perceived direction of the sound source. (The name Haas effect was suggested by Richard Bolt of the Massachusetts Institute of Technology.)

Around 1953, under the technical leadership of Günther Kurtze, a large *Reflexionsfreier Raum* ("free-space room") was constructed, which—unique in the world—was also designed to be nearly free from reflexions for *electromagnetic* centimeter waves to facilitate measurements with microwaves (antennas, etc.)

In general, there was a world-wide interest to supplement reverberation time by other physical parameters to characterize the acoustic quality of performance spaces. One of these criteria, favored in the early 1950s, was the so-called frequency-irregularity of the sound transmission between source (on the stage) and a listener's ears. Experimental results obtained by Heinrich Kuttruff and Rolf Thiele in the *Herkulessaal* in Munich (and other concert halls) showed that the number of maxima of the frequency response did *not* correspond to the number of normal modes (resonances)—as posited by a faulty theory by Bolt—but was actually more than a *thousand* times less. This astonishing result was explained by a statistical theory by me, then a postdoc at the institute. This theory showed that the frequency response (sound pressure and phase as functions of frequency) is, approximately, a complex Gaussian process. As a result, the average distance between maxima above a critical frequency ("Schroeder frequency") is fully determined by the (reciprocal) reverberation time—thus not affording the much sought-after new quality parameter.

Besides measurements in actual rooms, scale models were also studied, including electromagnetic models, using microwaves rather than sound waves. In my dissertation (1954), I could show that the distribution of the frequencies and



Fig. 19.1 Concert Hall Acoustics using Microwaves: Erwin Meyer explaining my microwave setup to Kultusminister Voigt

excitations of the normal modes in metallic cavities were highly irregular—even for very small deviations from the symmetry of a perfectly rectangular space, such as a cube. Thus, for all practical purposes in room acoustics, the normal modes (resonances) can be considered completely random (Fig. 19.1).

In the early 1970s, in a large study of concert hall quality, sponsored by the *Deutsche Forschungsgemeinschaft*, Dieter Gottlob, and Karl-Friedrich Siebrasse—on the basis of measurements in 22 concert halls in Europe and the USA—showed that the lack of early lateral reflections in many modern halls with low ceilings and wide (fan-shaped) ground plan was the main culprit. To counteract this deficiency, I developed sound-diffusing structures ("reflection phase gratings") based on number-theoretic principles (quadratic residues), which have found broad acceptance in room acoustics.

I also proposed a new method of accurately measuring reverberation times by "backward integration" of the (squared) impulse response of a room, using the so-called maximum-length sequences constructed from the theory of finite number fields.

Before being fully accepted at the institute, you had to design a new experiment for the institute's lab course, and I was asked to set up an exercise with 10-cm microwaves involving *round* waveguides. I said: "Professor, round guides means *Bessel* functions and I don't like Bessels." "Not to worry he said, all you have to know are the first few zeroes of those functions: 1.84, 2.40 and so forth." And Meyer was basically correct.

I soon discovered, though, that my round guides always had *two* resonances close to each other instead of just one. What was the reason? *Nobody could tell me*. I finally had to figure it out myself—drawing an analogy with the vibrations of a kettle drum. The drum in the relevant mode of oscillation always has two resonances, going with the two possible orientations at right angles to each other. Thus, my round wave guides showed what the physicist calls degenerate modes. For my master's thesis I elaborated this "discovery" into a precision tool for making very accurate measurements at microwave frequencies.[1]

Meyer was much impressed and felt I should go to the States, the microwave haven. So I applied for a 1-year Fulbright scholarship. But I flunked resoundingly—not because of my English and certainly not because of my academic credentials. The Fulbright commission felt that I was not sufficiently *politically* motivated (spreading the gospel of American democracy after my return to Europe). It was, alas, all too obvious that I was interested only in science, mathematics, and languages, to the exclusion of almost everything else: politics, sports (except tennis), etc. I remember when, in the summer of 1954 (I was just returning to the clubhouse from a few sets of tennis), the entire membership went wild, exclaiming: "We won, we won!" I thought they meant that our junior tennis team had beaten nearby Einbeck. No, Germany had just beaten Hungary (or was it Brazil?) in the finals of the soccer world championship. The whole country was in an uproar (as if Germany had belatedly won World War II) and I didn't even know that there was a championship on.

Other hobbies of mine were the theater (the famous *Deutsche Theater* under Heinz Hilpert) and movies. I fell in love with Hildegard Knef and her films (*Film ohne Titel, Die Sünderin*). I particularly liked *Les Enfant du Paradies* (with Jean-Louis Barrault), *Casablanca* (Bogart and Bergman), and *To Have and Have Not* (Lauren Bacall's first film, with Bogey) and, later, *Les Vacances de M. Hulot* (which I saw in four different languages) and other films by Jacques Tati. Another favorite of mine was Roman Polanski's *Tanz der Vampire* (*The Crazy Vampire Killers*) with his wife Sharon Tate and *A Man and a Woman* (with Jean Louis Tritignan and Anouk Aimée).

After finishing my work for the degree of *Diplomphysiker* (sort of M.S.) in just 4 years after coming to Göttingen, I had enough of university life and went into industry: *Grundig Radio*. In early 1952, they were just beginning to set up production of (black-and-white) TV sets. For their production line, they needed various testing devices and I joined that effort building a square-wave generator that could switch between its two states ("black" and "white") in just a few *nano*seconds—an unheard-of speed in the early TV manufacturing days. The price I paid was a very unusual electronic circuit that could give you painful shocks if you got too close.

[1] After I got my degree of *Diplomphysiker* on 19 December 1951, one of our co-eds appeared, out of nowhere so to speak, in my home town to spend a night with me. This kind of forwardness has happened to me only one more time—shortly after I got my Ph.D. degree. (I never realized that fresh academic degrees were that seductive. *Ius primae noctis*?)

When I told the Grundig managers that I had promised my Göttingen professor to return after 6 months in industry, they offered to increase my salary from 500 to 750 Deutschmarks and then even hinted I should name my price. But I made the right decision by honoring my word to Professor Meyer and start working on my Ph.D.

The fact that Max Grundig had said of acoustical physicist Helmut Haas (of Haas-effect fame) that all scientists should be kicked in the behind also facilitated my decision to return to Göttingen. Haas had been put in charge of developing a dictation machine and made a mess of it. It was known in the company as the *Jaulophon* (wowophone). But then Haas was not a mechanical engineer. Before this disaster, Haas was Grundig's fair-haired boy—driving around in the boss's Packard, one of the few American cars owned by Germans in Nuremberg. So perhaps I would suffer a similar fate.

When I said good-bye to my friends at the Grundig lab, the head of the storeroom called after me "rechts von den Trompeten" (to the right of the trumpets) and I knew exactly what he meant: when, on Judgement Day, I was looking for him, I could find him standing to the right of the trumpets—in other words our farewell would be for a very long time.

One of my co-workers (Gäbelein) at Grundig always came to my desk to chat in his local (Fürth) dialect—of which I understood very little. Once, he said, Herr Schroeder you look as if you were thinking of something else—which I was.

19.5 Ph.D.

I had always felt a great attraction to statistics and when I plunged my microwave antenna into a large discarded US army biscuit tin (after the last crumbs had been shaken out), I saw on my scope thousands of resonances, a veritable "forest" of spikes, all of different height and seemingly randomly distributed in frequency (around 10 GHz). This looked interesting, to say the least! But Meyer wanted me to do *acoustics* (his field) for my thesis. Yet I didn't want to part with my microwave statistics. So we compromised: I would investigate the random resonances of microwave cavities and he would interpret the results in terms of concert hall resonances. Thus, my published thesis bore the unlikely title "The distribution of resonances in large rooms: Experiments with microwaves."

I started working on my thesis in October of 1952 and was able to hand it in before the end of the following year—after just 15 months, which must have been a record for a Ph.D. thesis in experimental physics.

Our Institut, the *Drittes Physikalische Institut*, already had another Ph.D. student called Schroeder, so I became *Schroeder2*. But I soon got tired of always having to call myself *Schroeder2*, so one day, when I was asked which Schroeder I was, I answered, on a whim, *Schroeder17*—and the name stuck! For the next 50 years I remained *Schroeder17*. When the Internet came around, I chose as my e-mail address MRS17. (There is a double entendre here: some of my correspondents

were eager to meet the 17-year-old "Mrs." they thought was luring behind my screen name.)[2]

I had always been (and still am) an *eye*-person strongly attracted by things visual (landscapes, flowers, people, architecture) and now I had become, willy-nilly, an *ear*-man. From an early age on, I liked photography and, while I first worked with borrowed cameras, I already was the proud owner of a precision light meter (an *Ikophot* from Zeiss). Once, when I gave a slide show at the Eggers' house, the mother of my friend Frieder Egger exclaimed "Ganz (wholly like) Caspar David Friedrich!" And I didn't even know who Caspar David Friedrich was. But I think she was right: the pictures from my "romantic" period did have the feel of C.D. Friedrich. Highly educated in the humanities as she was, Frau Eggers lacked (or pretended to lack) the most basic understanding of the physical world. Once, during a walk near Göttingen, when it started to rain, I remarked casually "No wonder, with all the clouds above us." "Herr Schroeder, you mean the rain comes from the clouds?" "Yes, Frau Eggers, the rain comes from the clouds." (I still have trouble believing this story, but it happened just as recounted.)

Another hobby of mine was dancing. We had a nice student swing band, the *Music Mixers*, that played at least once a week from 8 p.m. to long after midnight. While I was often late for lectures and other events, I was always ready to go the moment the doors of the *Mensa* (student mess hall) swung open, dressed in loosely fitting clothes and plenty of *Dextro Energen* (dextrose sugar) to keep me going until the wee hours of the next day. Another favorite venue for me was the *Taberna Academia* on Wilhelmsplatz with nightly dancing and weekly *Bockbierfeste* (bock beer fests).

Once, in Amsterdam, I went to a public dance hall with an Indonesian friend (Rudy Tian Tik Tiang) who encouraged me to ask one of the ladies at a nearby table for a dance. (I wasn't sure you could simply invite a stranger.) She was young, beautiful, and raven haired. When I asked her where she hailed from, she said Sweden. (When she noticed my surprise, she assured me that not all Swedes are blond.) What was she doing in Amsterdam? "Oh, I am on my honeymoon." And where is your husband? "He doesn't like dancing, so he stayed behind in our tent in the campground"—while she took their VW into the city. Some honeymoon!

Back to acoustics: For the distribution of the resonances, I had expected a "Poisson" process, but what I found was completely different. And, timid as I was, tried to explain it away. Now the actual distribution, named after Eugene Wigner, is recognized as a tell-tale sign for chaotic systems. But how should I have known? Wigner had introduced his distribution to describe the energy level spacings of large atomic nuclei. (It also describes the spacings of the zeroes of the

[2] The prime number 17 is a so-called *Fermat Prime*. The only known Fermat Primes are 3, 5, 17, 257, and 65,537. Fermat thought that all numbers of the form $2^m + 1$, where m is a power of 2, are prime. But Euler proved him wrong, showing that for $m = 32$ the resulting number is divisible by 641 and therefore not prime. (Try it. A modest-size pocket calculator will do! But to have "guessed" that 641 was a divisor of 4,294,967,297 required a mathematical genius like Euler.)

famous Riemann zeta-function from number theory.) It took another 30 years for people to recognize this common thread in chaotic systems, atomic nuclei and number theory.

Before delving into the actual measurements, I asked a renowned mathematician, Franz Rellich, whether a mathematical theorem existed to cover my case of "complex eigenvalues." (Hermann Weyl, Hilbert's successor, had proven the case for *real* eigenvalues, corresponding, in physics, to resonances without losses.) Rellich searched the library for maybe half an hour but found nothing relating to my case. Then Rellich said something that I will long remember: "Do you realize that you are the first physicist since the 1930s who comes to ask a mathematician for advice." He was of course referring to the fact that before Hitler there was a very close relationship between physics and mathematics in Göttingen which, along with so many other things, had been destroyed by the Nazis (Max Born, Heisenberg, Ludwig Prandtl, Peter Debye, Einstein (on visits from Berlin), Felix Klein, Hilbert, Emmy Noether, Richard Courant, Hermann Minkowski (before 1909), John von Neumann, Gustav Herglotz, Theodor Kaluza, Bartels van der Waerden (from Leipzig), Hermann Weyl, Carl-Ludwig Siegel, Wilhelm Magnus).[3]

19.6 Friends

My main contacts during my student days were Erhard Scheibe and Jobst von Behr, whom I had first met during the war in an air force radar school when we were both 17 years old. Both are still my best friends, now living in Hamburg as retired professors (of the philosophy of science and mathematical physics, respectively.). But my main support, other than my parents, was *Tante Lina*, my landlady, who had lost her entire life's savings during the currency reform in 1948. She cared and cooked for me. Sometimes, when we were both out of cash, she would produce a tasty meal with a few vegetables that I had bought with the deposit money from empty bottles. (Once, I suggested to her, in jest, that we rob a bank but she admonished me "Aber, Herr Schroeder, das ist doch verboten!" I countered: "Look, you just have to stand at the street corner and whistle when the police comes. I'll take care of the bank myself." But, no, she stood firm. We had to continue scrounging on the 85 marks/month she got from welfare and the 60 marks/month donated by my parents (who had "lost everything" during the war).

[3] I once heard it said that mathematician Emmy Noether was like a Nobelist in physics or chemistry. But that is a gross understatement. In the twentieth century some 400 physicists and chemists were awarded a Nobel Prize but Emmy clearly ranked among the top ten mathematicians of the century. She had to flee Germany in 1933 and died 2 years later as a professor at Bryn Mawr. I recently (14 February 2007) succeeded in locating one of her surviving relatives, her nephew Herman Noether, in Summit, New Jersey, from whom I could retrieve some documents of his Aunt Emmy for the Göttingen Academy Commission charged with assembling the *Nachlässe* (scientific estates) of famous mathematicians.

Later I added to our income by repairing radios and even building one or two sets from scratch. One radio I named (in sterling silver letters!) after my favorite big-band leader *Kenton*. It boasted a "dynamic expander" that made the music sound more exciting (and more disturbing to the neighbors). Once, when my *Kenton* failed, a radio repairman said: "No wonder. It says so even on the cabinet *ken Ton*—no tone." Later I added to my "income" by translating German research reports into English (at 5 deutschmarks per page). When I look at these translations today I am amazed how good my English was even then. Where did I learn it? Some in high school, no doubt, but much by reading—not Shakespeare but James M. Cain. Cain's novels (*The Postman Always Rings Twice*, *Double Indemnity*) which were on sale in Göttingen in paperback for one mark. But, as I discovered much later, I didn't know the proper pronunciation of many words: *steak*, *catastrophe,* and even the simple-looking *noise*. (I once ordered a "banana split" in a restaurant when I thought a whole banana would be too much to stomach. Was I ever surprised at the gargantuan dish—with lots of faked whipped cream—set in front of me.)

This was also the time (July 1949) that I gained my first "carnal knowledge," with a person (Christa) who was likewise a novice. She was 19 but I was already 23 years old, my somewhat late initiation being occasioned by my being caught up in the boys-only war and the "monastic" PoW camps.

I also listened a lot to AFN (the American Forces Network), which I had first encountered on D-day: 6 June 1944. I was in a hospital then (with rheumatic fever) and I had a little radio on my nightstand with which I could listen to London (very risky in those days, but I had turned the volume way down, with my ear glued to the loudspeaker). And what did I hear in the afternoon (3 p.m.) of that historic day? "This is the American Forces Network, with the Army Air Force band under Major Glenn Miller, broadcasting from London, England. Hello, boys in France." Followed by the *American Patrol*. (I wonder how many "boys in France" had the leisure to listen to AFN on those bloody beaches. But those who did must have felt a tremendous uplift.)

In 1948, after the almost total devaluation of the *Reichsmark*, life became a bit difficult for me. I therefore applied for *Gebührenerlass* (release from tuition payments), which meant I had to pass a special little exam to prove my worthiness. I selected the beginners' lab course and saw Professor Kopfermann for the test. It looked as if I knew all the answers except one which baffled me. "But this was covered in the lectures that go with the lab work," Kopfermann said. I countered, "I only went to a single lecture and then I quit. The lecture was awful." "How can you say that? I drafted the lectures myself and Dr. Paul, who actually delivered the lectures, is only a *Schallplatte* (phonograph disc)!" "But then Dr. Paul (who later won a Nobel Prize) is a very bad disc," I ventured. Understandably, this was the end of the interview. Kopfermann rose and I left his office starting to think of other ways to pay for my tuition.

19.7 Italy

What happened next is almost beyond belief. A week after the disastrous interview I got a call from the *Akademisches Hilfswerk* (the academic assistance office): Professor Kopfermann had approved my tuition request but had asked why I hadn't also applied for a *Barbeihilfe* (monthly cash support). "Yes, that would be wonderful. I could really use it." It was only 100 marks a month but, with back payments, I received some 430 marks for my first installment.

What did I do with all that money? I took the money and ran: I ran to the railroad station and bought a ticket to Italy for a summer course *La Lingua e Cultura Italiana* given by University of Pisa in the *Collegio Colombo,* situated in the beautiful *Pineta* (pine woods) near Viareggio already admired by Rainer Maria Rilke.

I'm really glad I went, following my friend Erhard who had taken the course a year earlier. I learned a lot about Italy, its people; it was the beginning of a lifelong love affair with the language. Courses (by Professor Bolelli) were given in the morning. In the afternoon—after lunch and siesta—we were free to spend on the nearby beach (*Bagno Giorgio*). They had two beautiful daughters. Fourteen years later, when I wanted to show Anny my erstwhile models, I couldn't find Silvia. I finally approached a young matron with three or four kids at her feet. "Where is Silvia?" "Silvia son' io"—that's me. I was glad to see her again and to see that her beauty at 17 had not been vain (Fig. 19.2).

I continued my Italian studies at the University of Göttingen with the *Lektor* there and read "Don Camillo e Pepone" and (with difficulty) Manzoni's "I Promessi Sposi" (The Betrothed) and Boccaccio's "Decameron."

After the summer course ended, I went with a French student into the Appuane Alps where we climbed *Monte Cavallo* (1,889 m) and spent the night in the rifugio *Aronte*. On our way up from the Mediterranean coast near Massa (think white marble and Michelangelo), we encountered a peasant who proudly pointed to a tractor, proclaiming "that's what we call a *trattrice*," as if we had never seen a tractor before.

Of course, once in Italy, I didn't want to miss Rome, the "Eternal City." I was lucky to find a room in the student dormitory of the university ("La Sapienza"). My roommate was a young Englishman who one evening returned with his hands inked in blue. He had bought, for a very low price, a "genuine" Parker pen. Except it wasn't a PARKER pen; it was a P.ARKER pen and he hadn't looked closely enough to see the little dot.

My only other memory of that dorm was that one night I woke up with a terrible toothache. I ran around the deserted neighborhood yelling "Farmáccia, Farmáccia!" But, in spite of mispronouncing the word, I soon found a drug store with a night bell. Before long two eyes appeared behind a narrow slit in the door and before I could say anything (I was still holding my aching cheek), the voice belonging to those eyes said "I denti?" "Si, Si, i denti!" He gave me some pills which I swallowed on the spot. The pain went away and miraculously never returned.



Fig. 19.2 Silvia at 17

What I found most impressive in Rome was the ancient though starkly ruined *Foro Romano*. I never had felt *History* that intensely. Luckily, I had a friend in Rome (Maria, a native of Gubbio) who showed me around the city and its sights and introduced me to some of her friends. Unfortunately, I later lost contact with Maria. She married a British diplomat and was "gone." But some 40 years later, when Anny and I visited Gubbio and bought some ceramics, the dealer, an elderly professor, remembered her: "Oh yes, Maria Moreno! But I haven't seen her in a long time" (Fig. 19.3).

Before leaving Rome, I had to pay some bills (for laundry, the return ticket?) and I had to sell my trusted *Baldinette* camera which a friend had bought for me cheap in East Germany.

On the way home, I absolutely wanted to see Milan and its *Duomo*. I put my suitcase on the train to Switzerland and asked the lady in my compartment to "watch it." I would be back shortly after some sightseeing. I had enough time to walk once around the Duomo and then I made a big mistake on the way back to the



Fig. 19.3 With Maria Moreno in Rome

Stazione Centrale: I ordered an espresso. But the espresso was so hot I couldn't drink it right away and as a near "destitute" student I couldn't just abandon it. When it was finally cool enough to drink, I resumed my way back to the station and discovered that the *Stazione Centrale* is very large. As I finally got to the right platform, I saw the red tail lights of the departing train. When the porters saw me running after the train, all yelled "Forza! Forza!" But it was too late: Train and suitcase were gone.

The next 24 h were the stuff of comic opera. I went to station master's office and told him (in halting Italian) about my *valigia gialla semiaperta*, my yellow suitcase, half open (because my tennis racket wouldn't fit in). He advised to take the next *Accelerato* (actually a slow train) to Gallarate, the next stop of the train I had missed. There the station police told me that the train stopped for only 2 min—not enough time to look for my suitcase. So I took the next slow train to Domodossola near the Swiss border. There a policeman told me he had searched the whole train (*ho visto tutto il treno*) and in fact had seen a yellow half-open suitcase in the car that I had specified, but a lady told him that it belonged to a man she knew. End of story. *La valigia é andata via*—the suitcase is gone, as the Italian policeman volunteered. Couldn't he help me further? Yes, he would take me to the representative of the Swiss Railroads, whom I told my sad story: *una valigia gialla semiaperta*... Upon which he said "Je téléphonerai avec Berne." I said, wait a minute, let me explain the whole thing in French first. Then on the phone he said "Ich hab' hier einen Herrn, der seinen Koffer..." I asked him whether I could repeat everything in German, the only language I really knew well. Then he said something that was really the highpoint of the suitcase saga. "If your suitcase was on the train as it entered Switzerland, you will have it back tomorrow. The car you indicated will be cleaned in Olten (near Basel) tomorrow, starting at 7:30 in the morning. By 8:30 the cleaning crews will have found it. The next train to Bern (where I was staying overnight) gets there some time before noon. Thus, if you go to the lost-and-found office in Bern, they will hand you your suitcase back." I thought

the man was crazy. But that's exactly what happened. The next day I went to the Bern station and said (without even giving my name) "Haben Sie meinen Koffer?" The kind Swiss went backstage and returned—with my suitcase, still half open. That will be "Zwei Franken und dreißig Rappen." I hope I didn't forget to say "Thank you!"

The whole trip to Italy ended on a drole note: I lacked the two Deutschmarks to pay for the *Eilzugzuschlag* ("express" train supplemental fare). I naively thought the German railroad should take me home as quickly as possible without asking inconvenient questions. But I had evidently forgotten the pedantry of German officialdom. In fact, the conductor returned with an agent of the railroad police. One of the passengers, who had followed the whole mini-drama, quickly slipped me a two-mark piece. So when the conductor reappeared with the police in tow, I said: "What's the problem?" and gave him the extra fare.

Next year I went again to Italy with my friend Erhard and a new camera, the lovely Zeiss *Contessa*. We spent 3 weeks in a *hotel* in Malcesina on Lake Garda. Afterwards I went to Venice for another overwhelming experience. Italy had a strong mental hold on me now, and 2 years later, when emigrating to the USA, I went of course via Italy, on the *Andrea Doria* out of Genoa.

19.8 A Vacation on the Rhine

The year before my Italian adventures, I spent 3 weeks at the invitation of my maternal grandmother in Sinzig on the Rhine, between Koblenz and Bonn. I had just befriended a young French student Geneviève Hochapfel, who was spending the summer in Germany to learn German. As her name indicates, she was of Alsatian extraction. I was allowed to bring her along to a hillside cabin that my uncle Paul Dehne had purchased from his ample income as a *Heilpraktiker* (a health practitioner without a medical degree—very popular at the time). Uncle Paul was my favorite uncle, always ready with a quip or trick like pulling candy from behind his ears or money bills out of thin air or "swallowing daggers." (To add to the realism, he spent a long time in the bathroom the next day before he appeared with the "recovered" dagger.) During the war, he kept a slight wound from healing so they had to discharge him from the service.

In the summer of 1939 while most Germans were still hoping for peace and believed that Hitler (after Munich) would get away with anything, he said goodbye to his nephew with the prophetic words "Auf Wiedersehen im Massengrab" (see you in the mass grave). Actually, we both survived the war but he lost his only (step) son, Ernst-Adolf (about my age), who disappeared in the snows of Russia and was never found again. My only other male cousin, Karl-Heinz, also perished in the war, in the senseless defense of Italy (ca. 1944). But he died in a military hospital, so *his* place of burial is known. I visited my uncle and his family in Düsseldorf during my honeymoon from the USA.

One night in Sinzig, when I wanted Geneviève to stay a little longer in my room, he came to my aid by warning her that there were rattlesnakes in the stairwell. But Geneviève said she wasn't afraid of rattlesnakes and left. (But she *was* scared of "zee liddle animals"—her name for *spermatozoa*.[4])

During a walk in the woods, we came upon boar tracks and Uncle Paul explained in his English (our only common language) "wild pork has crossed here." (Did we realize that there was something wrong with the sentence? I don't remember.)

The Rhine vacation included interesting visits to the wine growing country and Rüdesheim and a few hours at a lab for dubbing movies—a "big" industry in Germany, where almost all foreign-language films are dubbed into German (with people of any description, including black people, Chinese and Eskimos, speaking perfect German). The high point was, not surprisingly, a boat trip down the Rhine from Bingen to Cologne past many of the fabled castles.

After our return to Göttingen, Geneviève helped me print my many snapshots in the darkroom of the physics institute (Fig. 19.4). We were promptly denounced by one of the professors who was jealous of his darkroom privileges. But Erwin Meyer, the director, believed *me* (rather than the accusing colleague) that the pictures we were printing were strictly personal (and not for profit, as had been alleged). I also assured Meyer that I had a "sturmfreie Bude" (undisturbed pad) and didn't need the institute's darkroom for any hanky panky. So Meyer allowed me continued use of the darkroom—alone.[5]

19.9 First Congress

It was also during my years as a graduate student that I attended my first international congress, in fact the First International Congress on Acoustics, in Delft, Holland. Since I was the only student from Göttingen who spoke Dutch, the task of introducing our group to the Dutch students who were permanent residents of our dormitory fell to me. But our attempt at politeness backfired when I concluded our visit with the words "Thanks for your hospitality, but we have to go now" and received the unexpected answer "Ja, dat is ook goed zo, want er is nog steeds hoogspanning." (Yes, that is good so, because there is still high tension). This was in 1953 (June 17th)—8 years after the end of the war. The German misdeeds during the war were long remembered, especially in Holland which had suffered disproportionately.

[4] Her previous boyfriend, she told me, likened protected congress to taking a shower in a rain coat.

[5] Meyer, ever the gentleman, once had every reason to be angry at me: I had "borrowed" the Institute's projector to show some slides at home. When I brought it back the next morning, he had already been waiting for it because he needed it for his lecture. Then, upon plugging the projector in, the light bulb exploded with a big bang. Meyer was obviously "boiling," but all he said was "Diese Lampe bezahlen Sie!" (You'll have to pay for *this* bulb.)



Fig. 19.4 Geneviève in Amsterdam

It was during this visit, incidentally, that we heard of the momentous events that were unfolding in East Germany on that Wednesday.[6]

While the Congress in Delft was in progress, a circus was in town and besides all the wild animals, one of the shows was a "flea circus" where "trained" fleas danced, pulled little carts, and performed other feats. During an intermission, while the flea master was outside his tent barking for new customers, my friends (Georg Schodder, Rolf Thiele) egged me on to take over the show. So I started out "Dames en Heeren, onze flooien zijn getraineerd om. . ." Soon the fleas started hopping and dancing and the onlookers mistook my sham show for the real thing. But it didn't last long: when the real flea master returned he was, you guessed it, hopping mad. (And that I was a German impersonating a Dutchman didn't help.)

[6] *Der 17. Juni* was soon declared a national holiday in West Germany, and called *Tag der deutschen Einheit* (Day of German Unity) implying that the East German workers had taken to the streets (and faced Russian tanks) not to protest for better working conditions but for *Wiedervereinigung* (Re-unification).—This was an obscene lie on the part of the West. If you voiced any doubt of this interpretation you were labelled unpatriotic (*Einheitsverleugner*).

At the end of the year I completed my Ph.D. thesis and on 26 February 1954 I passed the orals for my degree (*Dr.rer.nat.*, pronounced "rare nut"). Seven months later I was on my way to the USA to become a Member of the Technical Staff of what was then called Bell Telephone Laboratories.

During my thesis work, I encountered some number theory problems, which deepened my relation with the "Queen of Mathematics" (Gauss' moniker). I discovered that in my cube-shaped resonator certain resonances were missing, their absence being explained by the *3-Squares-Theorem*, which says that integers which leave a remainder of 7 upon division by 8 cannot be decomposed into the sum of three squares. For example, $14 = 3^2 + 2^2 + 1^2$, but 15 requires *four* squares: $15 = 3^2 + 2^2 + 1^2 + 1^2$. (Multiplying these "forbidden" numbers by some positive power of 4 leads to more impossible integers, e.g., 28, 60, 92, 112, ... On average every sixth number cannot be written as the sum of three squares. When I once mentioned this casually to mathematician Deuring (during a talk he gave at the Göttingen academy), he had to reflect a moment and then agreed: "Yes, asymptotically." Right!)

Much later, in 1985, I wrote a "mathematical bestseller" about number theory (*Number Theory in Science and Communication*), published by Springer and now in its fourth edition.[7]

Emboldened by this success I wrote another book that became very popular: *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*. The subtitle is an allusion to David Hilbert's dictum when mathematicians around the world were attacking Georg Cantor's *Mengenlehre* (set theory): "From his Paradise no one shall ever evict us." (The last word in the subtitle—Minutes—is of course a triple pun.)

[7] One avid reader, Kristina Lerman, named my book as one of the few she would take with her to a coffee shop to keep her company. Some company!

Chapter 20
Bell Laboratories

Manfred R. Schroeder

Bell Telephone Laboratories in Murray Hill, New Jersey, where the Transistor (1947), Information Theory (1948) and Error-correcting codes (1949) were invented was of course a dream place to start one's career as a mathematically inclined physicist. I am not implying that people at other times and other places cannot enjoy their work as much as we did. But there is a general perception that Bell Labs was something special—a "national resource." This did not come about by accident: Bell management decided early on that freedom to pursue one's own ideas and stable long-term funding were the best well-springs of innovation.—I do hope that industrial research at some future time will see the light again and not limit research to short-range goals and immediate profit. Some of the greatest advances in the past have come from taking the long-range view.

How did I get to Bell Labs from distant Germany in the first place more than 50 years ago? Let me explain: I was studying at the University of Göttingen when a noted linguist from Bonn, Werner Meyer-Eppler, visited us around Christmas in 1951 to give a talk at the General Physics Colloquium. I don't remember the details, but he talked about Shannon and Information Theory. Perhaps this was the first time I had heard the name Shannon and I was immediately fascinated. This sounded much more interesting than the nuts-and-bolts of experimental physics. But after the talk, during the Christmas party, people professed not to have "understood a word" of the talk. The chairman of the colloquium, Richard Becker, even seemed a little miffed when I dared pronounce the lecture "interesting." How dare I! How could I find a field that was devoid of *relativity* and the *quantum* to be interesting? But undeterred, I approached my professor, Erwin Meyer, whether he could recommend me for a job at Bell Labs. Unfortunately, the answer was "no" because

M.R. Schroeder
Author was deceased at the time of publication.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_20,
© Springer International Publishing Switzerland 2015

he remembered one of his students before the war (at the height of the Jew-baiting pogroms in Germany) who wanted to emigrate but Bell said they didn't hire any foreigners. End of dream to join Bell! Or so it seemed.

Two years later I learned that William Shockley, co-inventor of the transistor, was coming to Göttingen actually *looking* for bright students for Bell. Well, I flew back to my professor telling him "they *do* take foreigners . . . they are even *looking* for them." Fine he said, I am corresponding with one of the research directors of Bell and I will put in a few lines for you. (Later his secretary told me that he had "put in" a whole *page* singing my praise.)

When I said good-bye to Professor Becker, he asked me where I was going and I answered: "Murray Hill." And he knew exactly what "Murray Hill" meant. In fact he responded "Hmm, Murray Hill." Now of course the name Murray Hill doesn't ring any more bells (no pun). Indeed, a street sign in neighboring New Providence directs you to Murray Hills. *Sic transit gloria mundi* (or whatever they said in ancient Rome).

Two weeks later, I received an invitation for an employment interview by said research director. I was to meet him in the lobby of the Dorchester hotel in London, no less, on April 25, 1954, at 2 p.m. Well, the interview[1] apparently went well, and after 6 weeks I received an offer of permanent employment by Bell Labs for $640 a month—a princely sum then, especially for a perennially impecunious student. One alternative would have been for me to join Siemens in Munich for the equivalent of $125. Of course, I accepted the Bell offer—in fact I would have worked at Bell for nothing, but I didn't mind getting paid for what was for me a dream come true.

So on September 30, 1954, I arrived in New York on the Italian liner "Andrea Doria"—still afloat then. I was met at the pier by my future supervisor, Ralph LaRue Miller (who turned 100 on March 11, 2007), and the director, Winston Kock, with a long black chauffeur-driven limousine. But our first stop was not the Labs but an opulent restaurant (*The Newarker*), with a Menu in French that I couldn't read, except for the word *Bratwurst*. To this day, even after Iraq, French dominates many American menus. (I just returned from a luncheon at the *Grand Summit* in Summit,

[1] After explicating my thesis work, I asked the recruiter to tell me a bit about the Bell System. Well, there was the parent company, AT&T, Western Electric, the manufacturing arm, Bell Laboratories, and 23 operating companies: New York Telephone, New Jersey Bell, Southern Bell when, in the middle of this recitation, he stopped short and, with his eyes, followed an elegant young lady (an incognito countess?) traversing the long lobby. After a minute or two, without losing a beat, he continued yes, and there is Southwestern Bell, Pacific Telephone. . . Twenty-five years later, to the day, on April 25, 1979, I went back to the Dorchester. At the far end of the lobby there was a kind of hat-check counter with an elderly lady behind it. I went up to her and asked "Could it be that on this day, 25 years ago, on April 25, 1954, a Sunday, a young woman might have appeared from the door behind you—it was about 2 p.m.—crossed the lobby and then exited by the revolving door?" She must have thought I was from Scotland Yard or something. But, unflustered, she answered "Oh yes, of course, at 2 p.m. we had a change of shifts then. This entrance was for service personnel—chambermaids and so forth." I asked her, "How do you know this?" And she said, "I have been here for 30 years."

New Jersey, where I "savored" the *Soup du Jour*, followed by *Chateaubriand* and chocolate *Mousse*.[2])

The following three days were perhaps the most memorable in my entire life: My first regular job, a new country on a different continent and a new language (I used to pronounce "steak" *steek* then) and, yes, a new regular companion: on my third day in the USA. I travelled to New York and met a young woman from Bulgaria, Anny Menschik, who was working for Radio Free Europe and who soon became my wife.

Incidentally, on my first outing to New York I took the Lackawanna train to Hoboken and then the ferry across the Hudson. From the ferry landing near Christopher Street I wanted to take a cab to Anny's apartment on West 85th Street. But there was only one taxi and two people were already sitting in it. Therefore, having been in a taxi only once before, I hopped in and the taxi took off—for the *East*side. Nobody said anything. The cabbie collected the fare from the couple and then, without resetting the meter, doubled back to the Westside. The fare was $8.30 (more like $50 in today's dollars). But it was worth it!

In my application for employment at Bell I wrote "My confession is Roman Catholic," as was (is?) customary in Germany (*Mein Bekenntnis ist römisch-katholisch*). My Executive Director, Bill Doherty—a Catholic himself, treasured my letter for 37 years and then gave it back to me as a memento. He took an instant liking to me.

At the Labs, Win Kock encouraged me to continue my thesis work on concert hall acoustics (the discovery of a critical frequency, now called "Schroeder frequency"), but I figured—this being the telephone company—I better do something more germane to the telephone business. So I picked "speech" as my new research field—and speech it was for the next 50 years. Nobody objected to my total ignorance of this field. Electronic Engineering, especially, was an enigma to me, but I learned a lot by "osmosis" you might say. Bell Labs was an ideal place not only for doing research but for learning. The doors were wide open and people friendly and communicative. Nobody was guarding any "trade secrets." Not getting any monetary rewards for our inventions of course helped to foster this open atmosphere. I, for one, was quite happy with the $1. I received on my first day of

[2] Some New York restaurants are actually *bona fide* French. Anny and I often lunched with the **Einsteins** (Charlene and Ernst, of carpet fame) at such establishments, Charlene, a French native, usually making the reservations (*au nom d'Einstein*). When we happened to arrive first, asking for a *table pour quartre—au nom d'Einstein*, the maitre d' invited me, in a loud voice, *Suivez moi Monsieur Einstein*—and the whole restaurant would look up but was of course disappointed when they saw little me instead of the famous physicist. But it was fun—quite apart from the good food!—While we take French for granted in New York City, I was surprised when once, in Saarbrücken, only a few miles from the French border, nobody in a French-name restaurant could handle the language.

work for all my future inventions. (This amounts to a little over 2¢ per patent for the ca. 45 patents I received while at Bell.)[3]

Just one tiny but typical example: one day I was reading a technical paper bemoaning "the metallic twang" of existing artificial reverberators for electronic music. I thought what these audio types needed was a reverberator that—when seen as an electrical filter—has an all-pass frequency response. So I swivelled around in my chair and asked my office mate, electrical engineer and country musician Ben ("Tex") Logan, "are there any all-pass filters with an exponentially decaying impulse response?" Ben's answer was "yes" and *colorless artificial reverberation* was born. As far as I know, Bell Labs didn't make a penny on the patents. But you can now find colorless reverberators and artificial stereo (now called soundscape, surround sound, or virtual acoustic images) in practically all electronic music instruments, Yamaha and the rest.

Kock (whose hobby was breeding orchids) asked me to join the phono-vision research effort (known around the Labs as "phony-vision") but, reluctantly, I declined. Phono-vision never really worked; it was too far ahead of its time. Ten years later, in 1964, AT&T introduced the Picturephone™ but there was no sufficient market for it. The real breakthrough had to await *digital* image processing. Now, of course, phone and image are nearly inseparable—think cell phones and Skype™.

After giving the required "Declaration of Intent" (to become a US citizen) on June 26, 1956, I also got a *Top Secret* clearance. But I never did much work for *Jezebel*, the code name of Bell's Ocean Acoustics research. This was a pity, because the top secret work involved side trips to Eleuthera and Bermuda which, however, I did visit once or twice (in June 1966) to test one of my ideas (which I called "volume focussing") for detecting the presence of submarines in a deep ocean environment. Our official host in Bermuda was the Royal Navy.

No memories of the good days at Bell would be complete without remembering some of the great people I was privileged to work with like the late John Kelly, a true genius, and Jim Flanagan, who joined my department around 1958. After fathering innumerable advances in speech and acoustics, Jim went on to the highest honors: the Marconi Medal, bestowed by the Spanish king, and the National Medal of Science, a unique honor in our field, presented by President Clinton in a White House ceremony.

[3] A patent held by one of my collaborators, **Joseph L Hall**, recently figured in a suit brought by Alcatel Lucent against Microsoft for infringing on their MP3 patent. Lucent (the former Bell Labs) was awarded $1.52 *billion* (February 22, 2007). This patent was an outgrowth of our work on perceptual speech compression, applying it to music signals.—Of course, Microsoft will appeal the decision.

20.1 John Robinson Pierce

One of the greatest leaders at Bell was the unforgettable John Pierce of satellite communication fame and the father of *Telstar*, the first transatlantic space TV (July 4, 1963). In fact, Pierce (1910–2002) was the most inspiring boss and mentor I had between 1955 and 1971 when he left Bell Laboratories to join his old alma mater, the California Institute of Technology. By his writing he inspired my own writing—although when I once told him I might call my next book "Number Theory for Almost Everyone" he suggested instead, in his usual acerbic mode, "Number Theory for Almost No One." Other famous sayings by Pierce were "A job not worth doing isn't worth doing well" (to his boss W.O. Baker) or "Artificial Intelligence is mostly real stupidity," or "In the university, no one can tell a professor what to do. But, on the other hand, in any deep sense, nobody cares." What is the main purpose of the National Academy? According to John: "To keep the riffraff out."

Pierce also coined the word *transistor*, when co-inventor Walter Brattain asked him, over lunch, to come up with a good name. (Pierce's coinage is an amalgam of "*trans*conductance," a parameter measuring the sensitivity of vacuum tubes, and res*istor*.) When Pierce suggested *transistor*, Brattain exclaimed "That's it, Pierce, TRANSISTOR!" The name also fitted in nicely with other electronic devices such as *varistor* and *thermistor*.[4]

Pierce was also not averse to risqué jokes: As when, according to John, a young bride complained to her gynecologist of "painful congress" and the kind doctor, after having explored innumerable causes, finally asks the young lady "Do you smoke afterwards?" And blushing all over she admits "I don't know, Doctor—I haven't looked."

But while often blunt, John could be quite sweet too, as when, in 1957, he called my Voice Excited Vocoder "The first speaking machine that sounds human."

[4] In a piece on technical neologisms in the *New York Times Magazine* (in the Fall of 2006) James Gleick, who should have known better, claimed that the name transistor was the work of a committee. Yes, a committee of two! (Where did Gleick get the idea? From *Wikipedia*? No, *Wikipedia* actually got it right.) My Letter-to-the-Editor of the *Times*, a paper that prides itself on correctness, pointing out the error, was ignored.

In fairness, I should add that the *Times* and the *International Herald Tribune* have published all my previous letters: one concerning a story (about 1980) of a "singing computer" in Dresden, then part of communist East Germany. I pointed out to the *Trib* that the first singing computer ("Daisy, Daisy…") was demonstrated by J.L. Kelly and Carol Lochbaum at the Fourth International Congress on Acoustics in Copenhagen as early as 1962. Their reporter had simply been duped by communist propaganda.

Another of my printed letters concerned the use of a script L by Einstein in a paper on special relativity and his famous formula $E = mc^2$.

One of the most astounding admissions of "guilt" by the *Times* concerned a (very subtle) grammatical error they had committed in an *editorial*. I wrote them that I was a recent immigrant and English was my second language but something in the construction of one of their sentences seemed amiss to me—and the *Times* published my letter in full. Long live the *New York Times*!

(The reason why modern cell phones produce intelligible speech—and in fact would sound pretty good if connected to a high-quality loudspeaker—is that they incorporate the principle of voice-excitation.)

While John and I were watching the unfolding debacle of the avant-garde spectacle "9 Evenings: Experiments in Art and Technology" at the 69th Regiment Armory in New York City, I was concerned about the bad press our friend Billy Klüver (1927–2004) and *Experiments in Art and Technology* (EAT) might get. But John reassured me "They will be written up in the New York Times, and that's the main thing—never mind *what* they actually write."

John always impressed me by his honesty. Every time I had something unpleasant to report, he would immediately call *his* boss, Vice-president Baker, to relay the bad tidings. I was impressed because the culture in which I had been raised, honesty—volunteering the truth—could be deadly. Lack of forthrightness, I believe, was in fact one of the root causes for the failures of Kaiser Wilhelm's Germany and Hitler's regime—and could yet undo other governments.

When John was mistaken for the inventor of the Traveling Wave Tube, which happened not infrequently, he would reply "Rudy Kompfner invented it—I only discovered it." Meaning that when John went to England on a wartime mission to search for ideas for a new, more powerful microwave oscillator for radar, he found instead Austrian refugee architect Rudolph Kompfner, working on a microwave amplifier that unfortunately had a tendency to become unstable and oscillate. John exclaimed: "that's just what we are looking for, an oscillator!" This also led to the low-noise traveling-wave amplifier without which John's idea for an ocean-spanning satellite communications system would have been impractical. (Pierce was also the inventor of the reflex klystron which was ideal as a local oscillator in microwave receivers.—In my Göttingen Ph.D. thesis I used two such klystrons to investigate the resolved resonances of metallic cavities modeled after concert halls, something that could not be done with sound.)

20.2 Computer Music

John Pierce had a long-time interest in music. He studied the piano while a student at Caltech and later installed a pipe organ in his home near Bell Labs. John, Claude Shannon, and Shannon's wife, Betty, who was a pianist, carried out several ingenious experiments to estimate the information content of music.

John and Max Mathews attended a piano concert in 1957, which included pieces by Schoenberg played by Schnabel. They both felt that the Schoenberg was great and Schnabel was horrible. During the concert, John said to Mathews, "Max, with the right program your equipment should be able to synthesize better music than this. Take some time and write a music program." This sojourn into computer music was possible because to facilitate research on speech coding, Mathews with Ed David and H.S. McDonald had recently developed equipment to put digitized sound into a computer and to recover processed sound from a stream of numbers

generated by the computer. John's support and inspiration led Mathews to write a series of programs, "Music 1" through "Music 5," which started and set the course of present-day synthesized music.

AT&T administrators were not enthusiastic about music programs. They asked for an explanation as to the appropriateness of the work in a telephone company laboratory. With the strong support of both John and Bill Baker, Mathews was able to show them how music synthesis grew directly out of vital speech compression research and how music synthesis techniques fed back useful technology to speech synthesis. Without the support and encouragement from John and Bill Baker, computer music would not have begun when, where, and how it did.

About 1956, Max, with Ed David, also introduced Bermuda shorts at the Labs, in the days before Bell Labs was air conditioned—thereby exposing the least attractive part of the male anatomy, but to the great relief of all male members of staff. (But it took another decade or more before women wore slacks.)

20.3 Freedom of Research

Computer music is an apt example of the freedom of research that pervaded Bell Labs.

An example of freedom at Bell, one where it backfired, occurred with a Swedish linguist, Sven Ö., we wanted to hire in the mid-1960s. His future department head, **Peter Denes**, who was to take him out for dinner, asked me what he should answer if the candidate enquired about "freedom" at the Labs. I told Denes to reassure the candidate that he had total freedom—he could do whatever he wanted to do. The next thing we knew was that Sven's office and the nearby corridor were adorned with posters of Mao Zedong, Ho Chi Min and Che Guevara—just to test what we meant when we guaranteed him complete freedom. As you might imagine—this was the height of the Vietnam War—there was quite a commotion. People came to see me saying they couldn't continue with their classified, secret work. I had to reassure them that posters couldn't actually see. Then plant security got wind of the situation. They called me and I told them I would take charge, everything was "under control"—which of course it wasn't. But at least I got the security people off my back. Then I called my boss, John Pierce, and explained the whole thing to him. When I was through, he asked me *what* I was going to do and I said "Nothing, John." He replied with a single word: "Right." Of course, in less than 48 h the whole thing had blown by.

While I may have handled the Posters Case pretty well, I made quite a few horrible mistakes. Here is just one example: Once, ca. 1964, when returning from an ultrasonics meeting in Boston, one of the return flights, an Eastern Airlines shuttle from Boston to Newark collided over Connecticut with a big transcontinental Pan Am plane enroute from San Francisco to New York. Part of a wing of the Pan Am plane was severed and fell to the ground but the rest of the plane made it

safely to its destination. But the smaller plane, the shuttle, had to make an emergency landing in a potato field near Darien, Connecticut. Luckily, nobody was killed and nobody was seriously injured in the mid-air collision. But people didn't know this. They heard only on the radio about the collision and assumed the worst. One of our group from Bell who was still missing was H. (I had taken a flight an hour earlier.) Then I made the unpardonable mistake of calling his wife to see whether he was at home. He wasn't, and his poor wife started crying on the phone.—H. later returned safely to his family: he had taken a later flight.

20.4 William Oliver Baker

One person from whom I learned a lot about human relations and research administration was our Research Vicepresident Bill Baker (1915–2005). Bill would always think long and hard before making a decision—which, however, he often communicated in, to many, incomprehensible language. In other words, he made *you* think hard, too, and hopefully come to the same conclusion. This could be very frustrating but I, for one, preferred his indirect way of operating to the "methodology" of his successor, a Nobel Prize winner no less, who always seemed to act first, then talk and finally, as a last resort, start thinking.

Baker was a true "diplomat of science"—witness his many contributions as a humanist, his advocacy of the use of science for the benefit of mankind. He was a trusted personal advisor on matters dealing with science and technology and national security to Presidents Eisenhower, Kennedy, Johnson, Nixon, and Ford.—He received 27 honorary doctorates in recognition of his varied accomplishments.

Bill Baker was also instrumental in getting my main collaborator Bishnu Atal admitted to the USA. I had heard of Bishnu in the late 1950s as a very good student at the Bangalore Institute of Science in India. After a long telephone conversation I had with Bishnu, I was able to persuade our management to make an offer of employment. However, Bishnu didn't come—he couldn't get a visa. After waiting for 2 years, I asked Bill Baker to intervene, who had Bishnu declared as "crucial to our efforts in national defense." So Bishnu got a special dispensation from Washington forthwith, and he appeared at Murray Hill in no time—thanks to the excellent relations Baker had with various government departments.

Given Baker's standing, it is with great satisfaction that I learned of the high esteem in which he held me throughout my career at Bell.

Here is one of my favorite Baker quotes:

> The ideas of scientific discovery come one at a time from one person and one mind at a time. Sometimes two or three can aid each other. But scientific discovery cannot be collectivized, and it does not flourish in collectivized structures.

20.5 John Wilder Tukey

Another memorable character was John Tukey (1915–2000) who divided his time between Bell Labs and Princeton where he headed the statistics department. John, who seemed to thrive on six glasses of skim milk for lunch, is of course best known for his (re)invention (with Jim Cooley) of the Fast Fourier Transform which has revolutionized digital signal processing. (Gauss had it about 150 years earlier but, as was his wont, didn't publish it—to him it was self-evident.)

Incidentally, Tukey added some of the most important words to the English language: *bit* (ca. 1948), *software* (1958) and the less well-known *cepstrum*—and thereby hangs a story: In the late 1950s US and Soviet diplomats and scientists were negotiating a nuclear test ban treaty in Geneva, Switzerland. One crucial question was how to distinguish reliably between underground nuclear explosions and earthquakes. Tukey (with Bogert and Healy) suggested untangling the multiple reflections from the earth's mantle from the direct shock wave by the "cepstrum" method which meant taking the Fourier transform of the *logarithm* of the power spectrum of the signals received by various strategically placed geophones. (Earthquakes generate "shear waves" caused by the movements of tectonic plates. By contrast, explosions generate "compressional waves" which travel faster in the earth mantle, leading to measurably shorter delays between successive reflections.)

Being always close to Tukey, as a kindred (mathematical) spirit, I soon heard of the cepstrum idea and thought it was just what was needed for pitch detection for speech signals, namely to untangle the formant frequencies (the resonances of the human vocal tract) from the fundamental frequency, the pitch periods, the delays between successive flaps of the vocal chords. This was in the summer of 1962 and I had to attend some meetings in Europe, so I assigned the problem to one of our interns, A. Michael Noll. When I returned to Murray Hill a few weeks later, Mike had solved the whole pitch problem. The cepstrum has since found numerous other applications in speech coding and of course geophysics. Mike later became one of the pioneers in computer graphics and I was very glad to have kept him on—even without a Ph.D. degree, which was *de rigueur* under the guidance of Bill Baker. (Usually, interns are returned to the home departments after a year or so.)

20.6 Dave Slepian, Jessie MacWilliams, Ed Gilbert

Other mathematicians whose advice proved invaluable were Jessie MacWilliams, who taught me finite fields (which I needed for the design of diffusing surfaces in concert halls) and Dave Slepian, who along with his wife Jan became lifelong friends. Calculating reverberation times of acoustic enclosures (auditoria, opera houses, etc.) is notoriously difficult. According to Dave, one has to solve an "impossible" integral equation. But a day later, Dave brought mathematician Ed Gilbert along who had the brilliant idea of turning the integral equation into *two* equations which could then be solved iteratively on the computer. It worked like a

dream and David Hackman, a "cooperative student" from Columbia University, obtained some beautiful results showing the importance of absorber *placement* within an enclosure. (Dave Slepian, one of the pioneers of information theory, passed away on 28 November 2007—a great loss, also for Anny and me personally.)

20.7 Ron Graham, Neil Sloane, Roger Shepard

Another example of cross-discipline interaction at Bell was the work on Hadamard coding of two-dimensional images that I did with mathematician Neil Sloane (and which earned me the Erdös-number 3, meaning that Sloane had coauthored a paper with Ron Graham, who had written many papers with mathematical legend Paul Erdös).

Bell Labs provided, in fact, a sheer inexhaustible pool of helpful talent. How could I have done our work on concert hall quality without the scaling methods of psychologists Roger Shepard, Doug Carroll, and Joe Kruskal. George Sperling, whose field was human vision, became a lifelong friend.

20.7.1 Claude Shannon and Dick Hamming

The most famous of all the Bell "mathematicians" was (actually an electrical engineer) Claude Elwood Shannon (1916–2001), inventor of "information theory." (Shannon actually called it *communication* theory.) My contacts with him were relatively sparse but we were standing daily at lunch time in the same cafeteria chow-line which gave me, the novice, not a little moral uplift. Many years later, when I was a bit better known around the Labs, he occasionally consulted me on some acoustics problem or other. (I remember he once came to my office with a funny-looking little trumpet and asked me whether I could calculate its resonances from its shape.)

Another classical Bell person from whom I learned a lot was Richard Hamming (1915–1998), father of the Hamming distance and inventor of the Hamming error-correcting codes. Dick was also a great aphorist. Here is one of my favorites:

> In science, if you know what you are doing, you should not be doing it.In engineering, if you do not know what you are doing, you should not be doing it.

20.8 Bela Julesz

Bela Julesz (1928–2003), who invented the random-dot stereogram and used it to show that human binocular depth perception does not require recognizable shapes, became one of my closest personal friends at Bell.

Bela and his wife Margit had fled their native Hungary in November 1956 during the ill-fated uprising against the brutal Soviet occupation of their country. Already in December of 1956 we were able to welcome Bela to Bell Labs where he stayed and worked for the next 32 years. Bell gave Bela the freedom—and the means—to develop and test new theories of visual perception. Later, after having declined honorable offers from the Swiss Federal Institute of Technology, the University of Zurich and Rockefeller University, Bela became the State-of-New Jersey Professor of Psychology at Rutgers University.

Bela received numerous high honors, including the MacArthur Fellow Award. He was elected to membership in several prestigious Academies, including the Göttingen Academy of Sciences, intellectual home to Gauss, Hilbert, and Heisenberg. Bela was the quintessential Bell Labs researcher who inspired us all and made it such a joy to work there.

For his friends, Bela Julesz will live on as a great teller of intelligent jokes—always delivered with a grave demeanor and a charming Hungarian accent. One of my favorites concerns the Hungarian officer at the end of a pleasant night in a Vienna hotel who is about to leave without having paid his "dues." When the young lady of the night makes the gesture of paying, the great lover waves her off and, with great modesty, says "Ungarrischer Offezerr nimmt kain Gelt."

Once Bela invited us to an "exhibition" in a basement in Greenwich Village that featured a "dead admiral" lodged in a large aquarium. On closer inspection, the body was revealed as consisting of chicken and other nonhuman parts—a gruesome sight, nevertheless. There was also a live goat present and a New York policeman guarding the goat from being "violated," as the organizers had promised. This must have been in the late 1960s—unhinged times indeed!

I am forever thankful for the many good years I spent at Bell Labs and for the superb people I was privileged to know there. Of course I was too young to have much interaction with the old icons: physicists John Bardeen and Walter Brattain (co-inventor of the transistor) and the mathematicians Hendrik Bode and Serge Schellkunoff. (I only got to know Walter Brattain personally after his retirement at an *Institut de la Vie* meeting in Paris.) But I was once engaged in a joint project with Harry Nyquist, who discovered the important (for digital communication) sampling theorem. And I thoroughly enjoyed my friendships with the next generation: the physicists Sid Millman, Conyers Herring, Phil Anderson, G.H. Wannier, Bernd Matthias, Harry Suhl, George Feher, Warren Mason, Herb McSkimin, Bob Thurston, Albert Clogston, Phil Platzman, Pierre Hohenberg, Veit Elser, Chandra Varma, Günther Wertheimer, and Horst Störmer. (Matthias, Suhl and Feher later went to San Diego to form a "Bell Labs West.")

In 1966, I introduced Wolfgang Eisenmenger from Göttingen to Bob Dayem, who together launched the field of phonon spectroscopy.

I also had friendly relations with the electrical engineer Bob Lucky, an erudite author, and the gifted mathematicians Elwyn Berlekamp, Henry Landau, Larry Shepp, Andrew Odlyzko, Ingrid Daubechies, and Henry Pollak.

In the 1960s, I joined Larry Shepp and Conyers Herring for the weekly Russian Table, presided over by Astrid Werner, Bell Labs' chief Russian translator. Ingrid

Daubechies, a Belgian native, presided at the Flemish Table where I was able to dust up my Dutch. In the early 1960s, Dave Slepian and I founded an Italian Table. Thus, I sometimes spoke five different foreign languages at lunch on the five days of the week—Mondays: French, Tuesdays: Italian, Wednesdays: German (where I was one of the "experts"), Thursdays: Russian, and Fridays: Flemish. (At work we spoke English—or some approximation thereof.)

Sid Millman was Executive Director of Physics Research and Director of University Relations. After I became a professor in Göttingen, I had lunch with Sid every time I spent a stint at Bell Labs—a mutually instructive habit. According to Bell (now Stanford) physicist Ted Geballe, Sid, who was born in Russia in 1904, was only the second Jewish scientist to become a member of Bell Labs—hired in the late 1940s by James B. Fisk, a great gentleman of science who later became president of the Labs.[5] (Sid died on November 11, 2006, aged 102.)

Other people who were of great help to me were Floyd Harvey, who was delegated to teach me the ropes when I first began working at the Labs. Bill Kefauver, Harry Hart and Al Hirsch were my capable patent attorneys. Herb Hines, assisted me in building some of my more complicated electronic circuits. Jimmy Kronmeyer was a superb Old-World style instrument maker, as was Charlie Mattke, who invented an ingenious compact rotary dial to fit the new handheld *Princess* telephone.

One of the department heads who reported to me was Warren Mason, who held over 200 US patents and who invented the sharply tuned crystal filters—crucial for carrier telephony, i.e., the transmission of many voice channels over a single copper wire (or glass fiber). His invention of highly stable quartz-crystal oscillators are used in the ubiquitous quartz watches.

Robert L. Wallace headed another of my departments, developing highly directive microphones for conference telephony. In his early days at Bell, Bob invented the *tetrode transistor* which however was never used commercially. He and his brother owned a small telephone company in Texas which provided a sturdy extra income to his family. (The original capital came from the Roosevelt administration's endeavor to provide telephone service to outlying rural regions—at 2 % interest. Ma Bell did the billing for the small fry for free.)

Throughout my career as a research "executive" at Bell, I was greatly aided by a succession of very able secretaries: Ginny (in 1958—I forget her last name), Mary Porecca (when I became a director in 1963), who stood by me in difficult times, Dorothy Caivano, and finally (when I was commuting from Germany) Esperanza Plata, a Colombian native, who once called me the best boss she ever had.

[5] Richard Feynman, one of the greatest physicists of the twentieth century (and indefatigable pranksters), couldn't even get a *summer* job at Bell on his first try in the early 1940s. But AT&T, the "mother" of Bell Labs, was by no means alone in its anti-Semitic policies among major US corporations until after World War II.

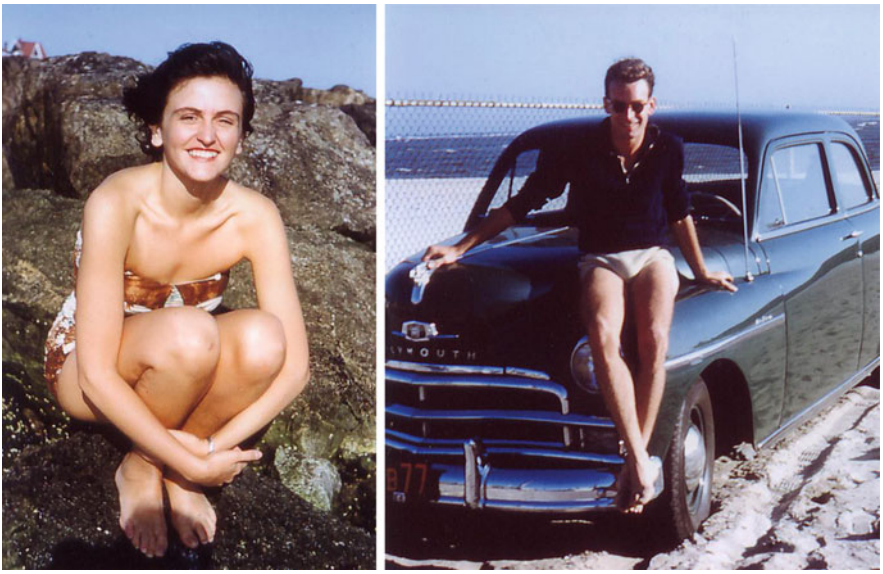


Fig. 20.1 Anny and Manfred (on his first car) (The Plymouth was manoeuvred onto the sand for this picture and got promptly stuck, but a helpful resident got his truck and pulled us out.) Long Beach, Long Island, Columbus Day 1954

My books were typed by professional typists from the Labs: Elvire Hung (*Number Theory*), Hildegard Franks (*Fractals, Chaos*), and Martina Sharp (*Computer Speech*).

Another friend was Floyd Becker. When I asked him on October 13, 1954 how he had spent Columbus Day (October 12), he said "painting my house." I got the same answer from Floyd Harvey and perhaps others and I decided never to ask again what people did on a warm, beautiful day off. Anny and I drove (in my "brand new" car) to Long Beach on Long Island for a swim and some sunbathing on an otherwise deserted beach—5 weeks after Labor Day, but still very summer-like to a recent immigrant from northern Europe (Fig. 20.1).

Besides New York State (the Hudson Valley), early excursions took us to New England for skiing and Florida for a Christmas vacation (Figs. 20.2 and 20.3).

20.9 My Early Days in the USA

Besides having to adjust to new technical fields, the social adjustments also were not always easy. I was 28 when I joined Bell and still unmarried. This was somewhat unusual at the time and my Executive Director, Bill Doherty, introduced me to one and all with "This is Dr. Schroeder. He just joined us from Germany—and he's a *bachelor*." Or, as my old math professor from Göttingen, Wilhelm



Fig. 20.2 Anny in Florida, December 1955

Magnus, then at New York University, put it: "This is not a country for bachelors. Either you will return to Europe within the year or you will get married." Well, he was right, I *did* get married 17 months after my arrival in the USA.[6] My Best Man was Ed David, who taught me how to grill steak properly (among many other American essentials), and, from 1972 to 1974, was ensconced in the White House as a presidential Science Advisor before Richard Nixon abolished the post. (Although the President had assured David of direct access to him, he actually never saw Nixon during his entire tenure—which may have been a blessing in disguise at the height of the *Watergate* scandal.)

[6] Win Kock asked me whether my bride was a "professional." I firmly swore *No*—with my still limited knowledge of English I thought *professional* was an allusion to the "worlds first profession."



Fig. 20.3 Cannon Mountain, New Hampshire, April 1955

20.10 Columbus Circle (About 1954)

Tuesday night, the regular session of the "Village Camera Club" was concluded. I left Bank Street (No. 673) to retrieve the old Plymouth from its street-side parking place: A few turns, West 4th Street (or was it 12th?), headed for Anny's place on West 85th Street. I had once made the run north as far as 72nd Street without a red light and I was determined to beat that record tonight. Would I ever reach 85th in one full swoop?

The lights on the Avenue were synchronized and the traffic not too heavy so that one could stay comfortably "in sync." But, oh my, the lights on Columbus Circle turned red just as I had crossed 58th Street, while beyond, Broadway, was all green. How could I reach these greener pastures which might carry me on to 85th? Obviously only by ignoring the ignominious red of the intervening Circle! Of course, one had to watch for other cars—who wants a mid-Circle collision?

Everything seemed to be going well with no apparent imminent collision when, suddenly, there was the traffic officer—whistling in the road. He approached the

(red-faced?) offender who was ready to receive the well-earned summons. But this was New York City—Mid-town Manhattan—not Summit, New Jersey, or Zanesville, Ohio, or what other explanation would there be for the ensuing pronouncement by the "arm of the law": "Mister, if you go on driving like that, you will—one day—end up with a ticket!"

Just then the Circle lights changed to green again and the speedy traveller was on his way again.

20.11 January Night 1955

Anny had to go to the Ladies' room once more before resuming our trip to Florida. It was Christmas Eve and we had "celebrated" the evening in a roadside EAT place in Bamberg, South Carolina. Anny had given me her handbag—containing all our cash and papers—to hold while she disappeared. I wanted to take a few more pictures of the neon signs and deposited the handbag on the car! bumper . . .

Half an hour and twenty miles later, Amy needed something from her bag. Where was it? On the bumper! But the bumper was unencumbered by handbags (or anything else) . . . Luckily, the Severins had enough cash to tidy us over our week in Florida.

Upon returning to New York City, Anny found a letter in her mail: ". . . when I was driving down U.S. 301 (?) with my girl friend on Christmas night, I saw a handbag lying in the middle of the road. It contained papers with your address. What shall I do?" Keep some of the cash and return post-haste!

In the meantime, I was driving without a license (or any other identification, for that matter). Of course, one hardly ever needs to show these papers—especially, if one drives carefully, obeys all traffic lights, etc.

One very cold Saturday, we were returning from the Bronx when, watching all the lights ahead of me, I went, smack, through one right in front of me. And, would you believe it at 2 a.m., the red light came with a real live blue cop.

"License, please!" "Yes," I said, hoping the inconvenient apparition would disappear again before my bluff was called.

"Your license, please." "yes, . . . I have no license." "Show me your registration." "Yes." "Your registration, please, mister." "Yes . . . I have no registration." "Show me anything!" "Yes. . . I have nothing . . . This is my girl friend." "I see." Then I saw the Bell Lab pass with my name and picture on the car's sun visor. "I have this, officer." "That's you all right, but it doesn't mean anything. For all I know, you stole this car." "Yes, officer, that could well be. But it is my car. I lost all my papers in Florida—in my girlfriend's handbag." "Good story . . . You know how cold it is outside?" "Yes, officer, about 20 below." "Do you know that I have been on these feet for 2 hours now and that I am frozen stiff?" "I can imagine." "You are the best thing that has happened to me tonight. We'll drive to the station house and I'll get a chance to warm up, while we book you." "You are very lucky, officer. I wish I was as lucky as you tonight."

Pause. Pause. And then the voice of the officer: "Go on. It's o.k. But watch those traffic lights. Good night!" "Good night, officer. And thank you!"

20.12 One-Way Street

> The show in the old Phoenix Theater was over (Had it been Marcel Marceau?) and Anny and I dashed across Second Avenue to retrieve our old Plymouth from an even older garage. It had started snowing and we were eager to reach our safe haven in New Jersey without a second to delay.
>
> As usual, it seemed like an eternity until the familiar green vehicle was disgorged by the cavernous storage establishment. We hopped in and off we went across the Avenue and into the next side street where, one third down the block a policeman emerged from below a lantern and stopped us. What was wrong? The snow was getting thicker and we were in a hurry! The officer approached my side of the car and lowered his head (if not his voice) to speak through the rolled-down window one short sentence which explained it all: "Sir, if you would switch on your headlights, you would see that this is a one-way street—in the other direction!" Whereupon the member of New York's finest marched toward Second Avenue to hold the traffic so I could back into the Avenue without harm. Thank you so much, officer!

20.13 Spy Stories

One of my more memorable encounters with the police took place not in New York but in Hoboken, New Jersey, in November 1954. It was night and the full moon was rising over Manhattan to the east. So, being an avid photographer, I took my tripod and a very long lens and was clambering around the abandoned piers when I was stopped by a bunch of longshoremen. A citizen arrest, if you will. They alerted the Hoboken police that they had caught a Russian spy with a very long lens and a thick accent. (The McCarthy days were still in full swing.) The police soon appeared on the scene and I explained to them that I was just a "shutter bug," and it so happened that I had two boxes with me, full of pictures I had taken: landscapes, flowers and—girls. Well, the police spent the next hour or so looking at my photographs and apparently completely forgot about the serious charge levelled against me. When they had finished their "examination" of my "art," they were convinced that I was indeed just a harmless amateur and they sent me home with their best wishes for my future work.

This did not seem to please my apprehenders; they felt I had "snowed" the police. So they reported me to their management, which alerted Bell Labs, where my executive director, Bill Doherty, vouched for me. Case closed—or so you might think. *Eight* years later, during my naturalization proceedings in Newark, the immigration agent asked me: "What were you doing during the night of November 10th in 1954 on the Hoboken piers?" Well, Bill Doherty had to rescue me again and I was sworn in as an American citizen in good standing on June 10, 1963—ready to

leave, on June 14th, for the Soviet Union as a US national. (But that's a different story; see my vignette on the KGB.)[7]

Lest I leave the reader with the impression that I always got away scot-free, here are a few counter-examples: I was once fined in New Providence, New Jersey, for supposedly not coming to a complete stop at a stop sign. And at another time I took an illegal left turn on 14th Street in New York City onto Park Avenue and was caught (by an unmarked police car). I felt the summons served me right because there was a large sign saying "NO LEFT TURNS."

Shortly after I arrived in the USA. I was caught speeding on the New York Thruway. I don't know how fast I was going—the needle of the speedometer was pegged at 110 miles per hour. But I am sure I wasn't going much over 100 miles; in fact, it didn't feel all that fast for a recent arrival from Europe. When a State Trooper caught up with me, he said it took him some time to catch up. But he only charged me with doing 95—the judge wouldn't believe him if he put in my actual speed. When I later appeared before a justice of the peace in New Paltz, New York, the kind man said: "The police must have been exaggerating. You couldn't have done more than 85. But I must warn you: "Speeding on the N.Y. Thruway is considered a *felony* not just a misdemeanor. And this may endanger your future application for US citizenship. So I'll put you on $5 bail and you consult with a lawyer in Summit, New Jersey, where I was living at the time, if you should plead guilty or not." But he forewarned me: "Our juries are wont to believe the police. If you plead guilty, I'll keep the $5 as a fine." My Summit lawyer (Hugh Hartlaub) felt that I should plead guilty and that there wouldn't be any long-range consequences. And there were none.

Lest some reader think that my friendly encounters with the police are exceptional, here is an excerpt from a *Dear Diary* letter by Noel Bacchus to the *New York Times* (Metropolitan Diary, October 2, 2006):

> A sunny Wednesday afternoon following a washout of tennis the previous day. We're heading for the Billie Jean King Tennis Center in Flushing Meadows-Corona Park, anticipating a day of competitive tennis. Three seniors: my brother visiting from Chicago, a friend from Mexico, and me.

[7] Here is what Doherty wrote to the Immigration and Naturalization Service:

As director of research in electrical communications, I was responsible for the selection of Dr. Schroeder in 1954 to work in our company as a result of strong recommendation by his professor at the University of Göttingen. My recollection of the incident in question is that I received a telephone call from the General Manager of our company, indicating that he had heard about Dr. Schroeder being observed taking pictures from the Hoboken waterfront. I immediately discussed the incident with Dr. Schroeder, and he showed me the pictures as soon as they could be developed. The pictures could not conceivably have been for anything except artistic purposes. I can well imagine that the persons who observed Dr. Schroeder might have been suspicious because of his telephoto lens. He has followed a long-time hobby of amateur photography, in which such a lens is increasingly common, and he has received awards in our company contests. I trust that this information may be of assistance in expediting Dr. Schroeder's petition. If it is possible to have this matter come up for final hearing on June 10, 1963, Dr. Schroeder would be able to commence his travel to Europe on government research business on June 14 as a United States citizen rather than a resident alien. This result would be in the public interest as well as in the interest of scientific research.



Fig. 20.4 Marion, Anny, Alexander, Manfred, and Julian in our first house in Gillette, New Jersey, September 1963

Crossing the boardwalk that extends from the No. 7 subway station to the entrance of the tennis center, we are frustrated trying to remember the words and tune to an old song, "Under the Boardwalk." Spotting a police officer—female, African-American, of a certain age—I impulsively address her.

"Good day, Officer, how are you?"

"Fine, thank you."

"Do you happen to know the song 'Under the Boardwalk,' by the Drifters?"

Looking a little surprised, she responds, "Of course!"

"Would you mind singing a bit of it for me?"

A big smile. "Are you serious?"

"Yes, Officer."

Without further ado, snapping her fingers, she sings quietly: "Under the boardwalk, out of the sun. Under the boardwalk, we'll be having some fun. . ."

20.14 Family Development

Not long after our wedding (in the Chapel of the Riverside Church near the Columbia Campus), Anny and I were blessed with three children: Marion Frances (born in 1956), Julian Ivan (1958), and Alexander Hilbert (1960). Marion's and Julian's middle names are based on the given names of Anny's parents (Franz and Ivanka Menschik). Alexander's middle name is that of the famous Göttingen mathematician David Hilbert (Fig. 20.4).

Here are two more snapshots from our years in Gillette (Figs. 20.5 and 20.6):

And here are our children in 1985, after we had moved to Germany (Figs. 20.7 and 20.8):



Fig. 20.5 Marion (age 7) Gillette, New Jersey

20.15 Stephen Oswald Rice

Another thing that surprised me was that people were not only very accessible but, I thought, quite humble like S.O. Rice of (mathematical) noise fame, one of my heroes even before joining Bell Labs. His office was at the old New York headquarters at 463 West Street. I had always imagined that such a renowned mathematician would reside in a palatial office with lesser people scurrying around him. But no, Steve was actually *sharing* an office with several other people who treated him like an ordinary human being. And then, a little later, we all went out together, on *foot*, for lunch in a little restaurant in Greenwich Village. I thought someone as famous as Rice would be carried by a company limousine . . .. Well this was just one of my European prejudices that took some time to subside.[8]

Incidentally, after lunch, when I returned to my car, which was parked illegally under the West Side Highway, I saw a cop writing a ticket. When he saw me, he apologized profusely but since he had already filled in my plate number he couldn't issue it to someone else. But he urged me to leave the car in the illegal spot for the

[8] There is now (2006) a *Stephen O. Rice Prize* by the IEEE Communications Society.



Fig. 20.6 Anny in her in first US Passport Photo, 1959

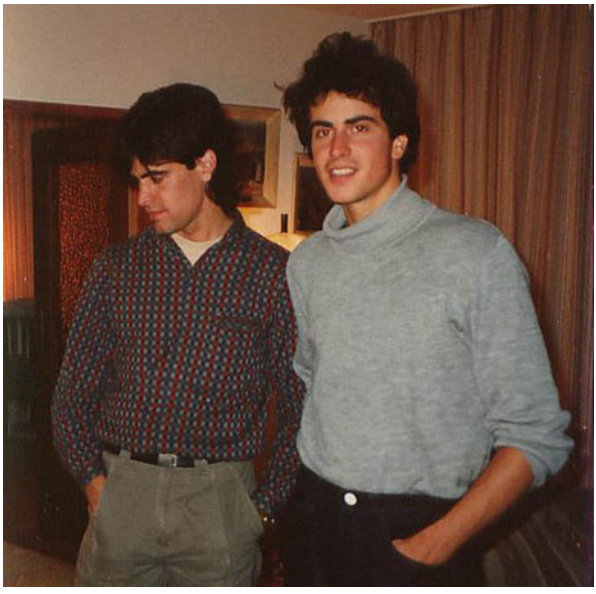


Fig. 20.7 Alexander (25) and Julian (27) in our Göttingen living room on a visit to Nikolausberg



Fig. 20.8 Marion (29) on a visit to Nikolausberg

whole day—the ticket was good for 24 hours! He was visibly disappointed when I told him I had to return to New Jersey right away.

As auspicious as my beginning at Bell may sound, it was, in retrospect, rather pedestrian. My first speech research job at the Labs, suggested by my supervisor Ralph Miller (who just turned 100 as I write this—March 10, 2007), was *inverse filtering*: you spoke a vowel sound into a microphone displayed it on an oscilloscope and adjusted tuneable filters until the formant "wiggles" disappeared and a relatively smooth waveform remained, which was a pretty good approximation of the glottal waveform. Neat! As I learned much later, as I was picking up more electrical engineering expertise, I was putting "zeroes" on top of the "poles" in the complex frequency plane. Of course, if the speech signal is the least bit noisy, inverse filtering will only lower the signal-to-noise ratio. (In order to *increase* the SNR, you have to use *matched* filters, just the opposite of inverse filters, piling poles on poles.) So inverse filtering is no good for pitch tracking for low-quality telephone signals, which was one thing we were trying to do. But it was an interesting introduction to speech signals.

My next (self-inflicted) project was a 6-channel vocoder ("voice coder"). Even though a complete novice, I felt that 10 or 16 channels for a vocoder was more than was really required, I figured that six channels would be enough. Indeed, I thought single-tuned filters would be sufficiently selective, if I added some "lateral inhibition" (as in the human eye) to my filter bank: subtracting from each filter a small fraction of the outputs of its two neighbors. It took me many months to build this

contraption because I had to wind my own coils. Imagine a Ph.D. in physics, winding coils and spending his days soldering sickly circuits—the technical assistance that I had been promised *presto* did not materialize until a full year later (the very capable Tony Prestigiacomo, who later shortened his name to Presti).

Be that as it may, after all the manual labor and mental gymnastics I had invested in my vocoder project I was convinced that it produced not only *intelligible* speech, but high-quality, human-sounding speech. Thus, I alerted my supervisor, who alerted our director, Win Kock, who—if you can believe it—alerted the president of Bell Labs (Mervin Kelly). They all filed into my lab and the demonstration began. . .. In retrospect, I am convinced they hardly understood a word of what my vocoder said. But everybody was very polite and all left with best wishes for the future of my project, an early attempt at speech compression, which later became crucial for the Internet and mobile telephony (cell phones).

Once, a cooperative student from MIT, Tom Crystal, played me *his* speech compressor in a noisy, fan-infested, lab. He thought it sounded perfect and so did I. So I suggested that we listen in a very quiet location and I put earphones on—and it sounded horrible. Recently, Fumitada Itakura reminded me that when I first visited him in Japan in 1968, I followed the same procedure. But Fumitada's "maximum-likelihood" compression still sounded pretty good, though not as good as at the International Congress on Acoustics in Tokyo, where our two papers, Fumitada's paper and mine (on *Adaptive Predictive Coding*) were presented back to back on 28 August 1968.[9] (Little did we realize then that our two methods were actually identical. Yes, Linear Prediction and Maximum-Likelihood (or Maximum-Entropy) exploit exactly the same redundancies in a speech signal.)

Another horrible contraption I built was an *analogue* Fourier-synthesizer capable of generating periodic waveforms with 31 harmonic frequencies. The gadget, built by the able Herb Hines, comprised almost 1,000 little trimmers (variable resistors) and a number of highly unreliable transistors (this was around 1956!). Nevertheless, I got some nice waveforms that helped me solve the problem of minimizing the peak-factor of periodic waveforms by choosing the proper ("Schroeder") phases and making synthetic speech sound more pleasing.

Before I had my analytic formula, I collaborated with Vic Vyssotsky (who later became Executive Director of our Division) on a computer sorting program to find optimal phase combinations. Vic was also a great hiker and I once went on the Appalachian Trail with Vic.[10]

Manipulating just the phase angles of a periodic signal while keeping its power spectrum fixed, I could elicit some astounding perceptual effects including *flat-spectrum* intelligible speech (published with H.W. Strube). Later Schroeder phases

[9] An earlier presentation of our work on predictive coding was at the first IEEE meeting on speech communication in Boston, 1967.

[10] Vic also fled to Brazil for a couple of years around 1960 because he was afraid the Northern Hemisphere would soon become unliveable, contaminated by the radiation fall-out of an impending nuclear war.

became important in hearing research—in studying the motions of the basilar membrane in the inner ear.

I also concocted cross-correlation and autocorrelation vocoders. The autocorrelation function squares the amplitude spectrum. To undo this spectrum squaring, I employed square-rooters, one for each formant range. Only much later did I learn that the proper way to do this is via taking the *inverse* of the correlation matrix (as is done in linear predictive coding of speech and present today in every cell phone).[11]

On a more fundamental level, I was intrigued by the relationship between vocal tract shape and its resonances. Of course, a point emphasized by the Swedish mathematician Borg, who later became president of the Royal Institute of Technology (KTH) in Stockholm, the problem is ambiguous: different geometric shapes, as the ventriloquist knows well, can produce the same acoustic output. (The ventriloquist can produce intelligible speech with his lips unmoving.) But by measuring the *impedance* of the vocal tract, as seen from the lips, I was able to obtain unique area functions from acoustic data.

20.16 Max Mathews

Incidentally, the loathsome analogue-circuit building days were soon over. In 1955 Max Mathews appeared on the scene coming from MIT. He taught us that every circuit we were building could be seen as performing some mathematical operation and as such, if it could be digitized and be done by computer. In fact, John Kelly, Carol Lochbaum, and Vic Vyssotsky wrote a block-diagram compiler, called BLODI-Compiler, that could be used by computer dummies like myself to simulate almost any electronic circuit. Carol also worked with Max Mathews on the design of digital fonts—a novelty at the time.

Although I had been raised in Germany under Hamming's motto "Don't compute—think!" I was soon one of the heaviest users of digital simulation including, with Ben Logan, of a "Harmonic Compressor" that comprised 400 narrow band-pass filters and could be used to slow down natural speech. We made this available to the American Foundation for the Blind which used it in their "recorded books" program.

[11] To test the intelligibility of my synthetic speech, I asked people walking by in the corridor outside my lab whether they could understand it. I said "One—two—three—four—five…" over the device and, yes, they understood every word: One—two—three—four—five… Of course, people could *guess* what was said just from the very *rhythm* of the utterance. But I honestly believed my correlation vocoder produced intelligent speech! (This is reminiscent of a similarly silly procedure adopted by prisoner-scientist Rudin in Alexander Solzhenitsyn's *First Circle of Hell*, when Rudin wanted to convince himself that frequency division by a factor of 32 would produce intelligible speech—an outrageous claim, as Solzhenitsyn, an engineer himself, no doubt knew. As test material Rudin chose the first paragraph of lead articles from *Pravda*, entirely predictable boilerplate, invariably beginning with the words: "The Presidium of the Central Committee of the Communist Party of the Union of Soviet Socialist Republics …").

Of course, the simulation didn't run in real time (this was in 1961) and some eyebrows were raised over my simulations that took several hundred times real time. Even earlier, in the late 1950s, I had started simulating frequency responses of concert halls: a complex Gaussian process in the frequency domain. To accumulate enough statistical data on some parameters that couldn't be calculated analytically, I let the computer run entire weekends. At $600 an hour, the cost was of course astronomical. But the only financial consequence to me was that my batch-processing budget was raised. Talk about "the good days" at Bell Labs! I never had to write a single research proposal in all my years at Bell—just do it and, if it made sense, the money would be provided, unless it was "additions to the plant," which were strictly budgeted. What helped of course was that, before "divestiture" (1984), AT&T—within reason—could write off Bell Labs as a business expense.

After the frequency-domain simulations, I began concert-hall simulations in the time domain, in other words: convolving the music signal with the hall's impulse response. The final output was of course reverberated music. When we first did this we had a little amplifier with a loudspeaker standing behind the digital-to-analog converter to monitor the computer output. When the people who ran the converters heard the music coming from their machines they thought some prankster was trying to fool them. But no, the music *did* come out of the computer—something entirely new in the 1950s.

Later I began simulating 16-channel vocoders, again in very "unreal" running times. Some people thought I was crazy. They had envisioned simulations for simple waveform coders, such as delta modulation or differential pulse-code modulation. But that was Bell Labs research in the years gone by: you could do "crazy" things if you so felt (and didn't mind a few raised eyebrows). A good case in point is Peter Denes's successful advocacy of *dedicated laboratory computers* for speech research—in defiance of considerable doubts by higher management who at the time (ca. 1964) were still wedded to "computing centers" with big machines.[12] In fact, we sometimes had to travel to IBM headquarters on Madison Avenue in New York City with a *trunk full* of punched cards representing a single sentence of speech (3 s). (One of the problems was the necessary double parking on 57th Street while we were unloading the digitized goods.)

The nearness to New York City also allowed us to attend many interesting meetings: the Audio Engineering Society, the Acoustical Society of America, various IEEE (Institute of Electrical and Electronics Engineers) sessions and many meetings of the American Physical Society.

I still remember with awe the "Woodstock of Physics" in March 1987 at the New York Hilton when high-temperature superconductors were first publicly presented. The session was attended by more than 2,000 people and lasted until 3:15 a.m. the next morning (I left before midnight). Many physicists followed the

[12] I remember Vice President Baker thanking me for having "insisted" on the need for small on-line computers when leaving our demonstration lab. ("Thanks for your insistence, Manny [*sic*]" etc.)

proceedings on video monitors set up around the hotel after the grand ballroom had been quickly filled. Drs. Alex Müller and J. Georg Bednorz later collected a Nobel prize for their pathbreaking work. (Müller and Bednorz had published their results first in a German physics journal to little fanfare before their paper was "discovered" by the rest of the world.)

On other occasions I saw and heard the great physicists Wolfgang Pauli ("Einstein's successor" at Princeton) and Robert Oppenheimer (leader of the wartime atomic bomb project) shortly before their untimely deaths (in 1958 and 1963, respectively).

20.17 Linear Predictive Coding

By 1966, I had become totally disenchanted by the speech quality of vocoder speech. I thought that instead of the "rigid" encoding of the speech signal practiced by vocoders, we should look for methods of speech coding that left "room for error." At this point I remembered the work of my friends in the picture-coding community, especially an engineer named Ernie Kretschmar, a refugee from Nazi-Germany—in fact the Münsterland, my native neck of the woods. The method of choice in video coding was of course prediction: point by point, line by line and frame by frame. I thought something like that should also work for speech signals, except that the predictor parameters should change with the speech sound: adapted to the different spectra of different speech sounds. That is why Bishnu Atal and I called our first paper *Adaptive Predictive Coding*, later called *linear predictive coding* (LPC[13]) and CELP (for code-excited linear prediction) which allowed the excitation signal to be transmitted by ¼ bit (!) per sample, or roughly 2 kb/s, an astonishing compression factor made possible by perceptual coding exploiting the masking properties of the human ear. J.L. Hall and I measured the masking of (quantizing) noise by the tone signals—the opposite of the usual paradigm in which masking of a signal by noise is studied. Perceptual coding is also used widely for bit-rate compression of music (MP3, etc.).

Code-excited linear prediction can also be considered a kind of *voice-excited* vocoder—hence its high speech quality. Apart from the technical breakthrough that LPC represented, its genesis is a beautiful example of the cross-fertilization that was possible, even encouraged, at Bell.

One of the more ubiquitous results of speech compression is the wide-spread use of cell phones and Internet telephony—unthinkable without compression.

[13] Correspondences with Bishnu Atal and J.L. Hall during the development of LPC are archived at the University Libray Archives of the University of Göttingen.

20.18 Computer Graphics

An illustration of the freedom we enjoyed at the Labs and the good communication between researchers is our computer art work. I had already used the computer in the late 1950s to produce artificial reverberation and concert hall simulations, and generally I was very interested in any non-numerical application of the digital computer. So I was immediately attracted by the possibility, suggested by Leon Harmon and Ken Knowlton, to use the computer cum plotter to generate graphic output that no human could ever hope to draw by hand, like their rendering of a nude Debbie Hay by electronic symbols.[14]

Leon became also widely known through his *block portraits*, which looked very different from different viewing positions: from close up, you see a set of squares of different shades of gray; from afar you see a human portrait (Lincoln in Leon's most memorable work). I was likewise much interested in the idea to produce graphic art that changed appearance depending on viewing distance. One of many interests that we shared was the recognition of human faces—by man or machine and the, probably age-old question: "What exactly makes a face *beautiful*?" (Leon, one of my closest soulmates, perished by his own hand in July 1982, several years after he had left Bell Laboratories.)

At about the same time, Mike Noll, Bela Julesz, and others made their own noteworthy contributions. Mike, who joined my research area as an intern in June 1962, became prominent in the computer art world, not least through his aesthetic analysis of Piet Mondrian's *Composition with Lines* by comparing it with several statistically generated Pseudo-Mondrians. His earliest computer art was, in fact, done in 1962. I vaguely remember approving, for internal Bell circulation, his August 1962 Technical Memorandum on using the computer to draw artistic line compositions. Of course, I had no idea of the historic significance Mike's memo would later acquire.

Mike also pioneered computer-generated stereographic movies and *tactile* computer interfaces, allowing geometric shapes defined by software to be physically felt.

Bela, who joined Bell research in late 1959, came to us from the field of visual perception. His thesis, in Budapest, was on optical camouflage. At Bell he became famous for his *random dot stereograms*. Some of his creations had, in fact, the feel of Op Art.

At one of the monthly research directors meetings, instead of presenting the latest progress in speech coding, I showed my new computer graphics—and nobody complained. In 1965 there was an exhibition of Computer and Op Art, including works by Noll and Julesz, at the Howard Wise Gallery in New York City. Computer Art had arrived—as it had in Germany at the same time: Curd Alvensleben

[14] The nude study of Debbie was published in the New York Times on October 11, 1967. Harmon's block portrait of Lincoln was incorporated (without permission or attribution) into one of Salvadore Dali's paintings.

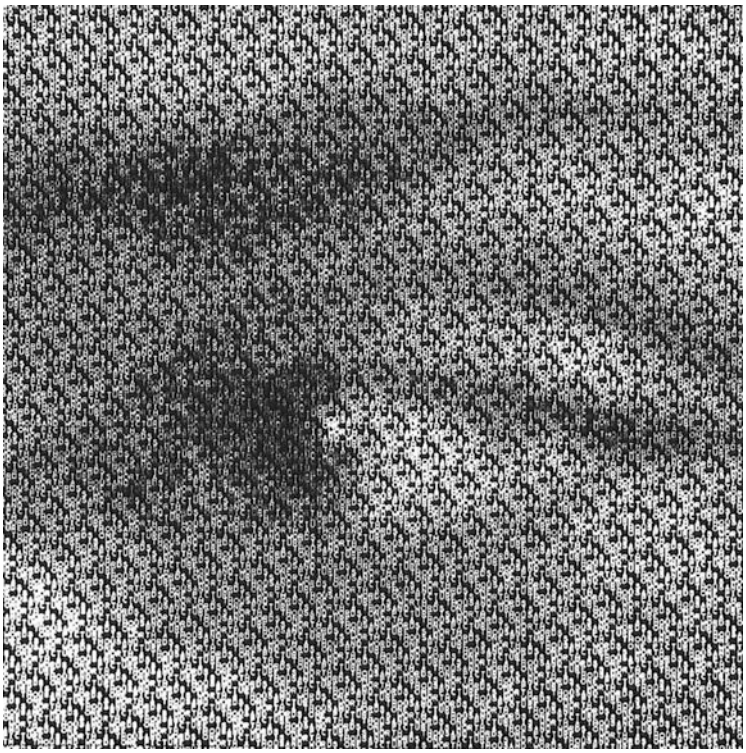


Fig. 20.9 One picture is worth a thousand words

(1962, using an analog plotter), Frieder Nake, Georg Nees, Herbert Franke and others. I even earned some money with my computer graphics—enough for a taxi ride and dinner for four in Zagreb, Croatia.

Here are a few examples of my creations. One design, programmed by Sue Hanauer, won First Prize at the Computer Art Salon in Las Vegas in 1969 (Fig. 20.9):

This computer graphic is a prime example of eliciting different perceptions depending on viewing distance: from close-up one can read the text: "One Picture is Worth a Thousand Words." From a still larger distance one perceives a kind of "weaving" pattern (caused by the periodic repetition of the sentence). And from afar one recognizes a human eye!

Another image, in the same vein, showed my initials (MRS) composed of 65,536 samples of the letter "S" with different luminosities.

Other shows at which my computer creations were exhibited included "On the Eve of Tomorrow," held in Hannover in late 1969 (organized by Käthe Schröder), "Some New Beginnings" in Brooklyn (also in 1969), "Cybernetic Serendipity: The Computer and the Arts" (Jashia Reichardt, London 1968).

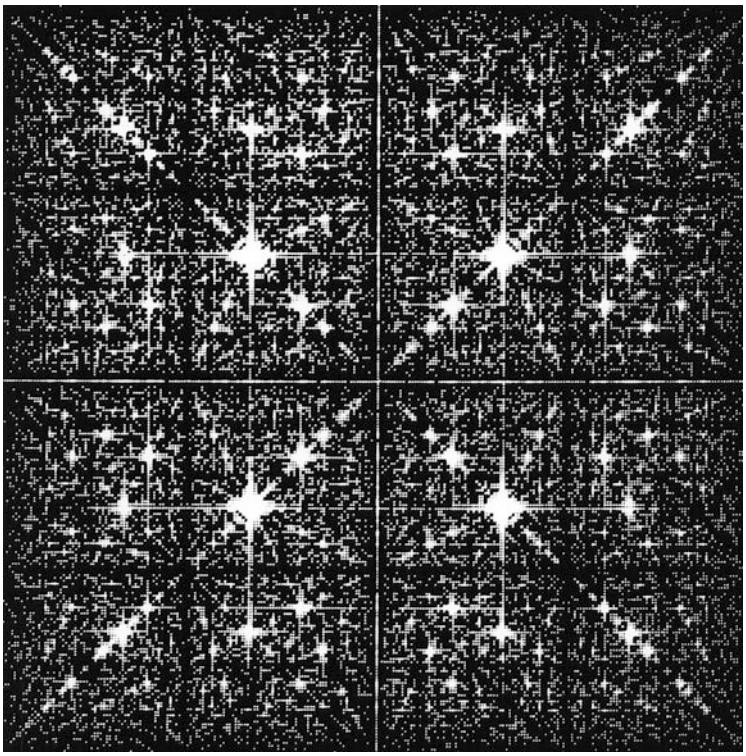


Fig. 20.10 Prime spectrum

Number theory was another inspiration for my computer-graphic endeavors. Here, as an example, is the Fourier transform of the distribution of pairs of relative-prime integers (Fig. 20.10):

Unbeknownst to me, computation of this image required shutting down the entire Bell computer center to marshal the necessary random access memory. (Fortunately, I was away on a trip at that time—or I might have prevented the effort.)

Here is another of my favorites, based on the eikonal equation of geometric optics that governs the propagation of light waves. The image (realized with the assistance of Hans Werner Strube and Wolfgang Möller) becomes recognizable as a human portrait at larger viewing distances[15] (Fig. 20.11):

With Gerhard Sessler and Mohan Sondhi I constructed an "Acoustic Camera," which was capable of reconstructing a physical object from its blurred acoustic shadow. With Mohan, I also did a study of the restoration of sharp images from

[15] I used the Eikonal Portrait as a cover of a book I edited (*Speech and Speaker Recognition*) where it was promptly misinterpreted (by G. Fant among others) as a contour spectrogram of speech.

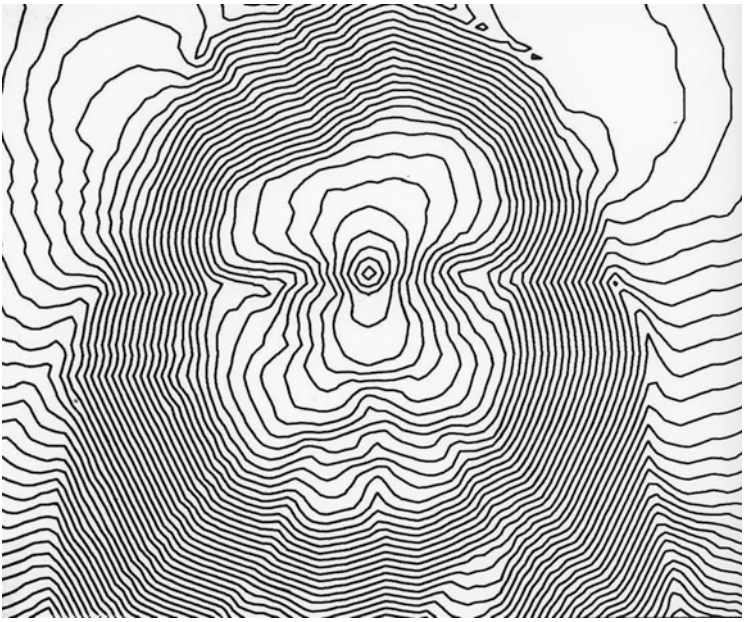


Fig. 20.11 Eikonal portrait

pictures degraded by linear-motion blur, and of computer ray tracing in the deep ocean. This was in connection with what I called "Volume Focussing" (suggested to me by the self-steering antenna arrays keeping communication satellites focussed on the proper earth station). Volume Focussing is capable of focussing on a specified volume in the ocean, important for the defence against submarines. The surface sound channel, called SOFAR, allows sound waves in the ocean to travel over thousands of miles with little attenuation. In a peacetime application, the SOFAR channel is used from Heard Island in the Indian Ocean to Canada, Australia, Antarctica and California to monitor ocean temperatures (and the effect of greenhouse gases) over vast distances.

Other early pioneers in computer graphics were Ed Zajac and Frank Sinden. Ed made computer *movies* that illustrated the tumbling of a satellite in orbit around the earth in connection with the efforts to stabilize the Telstar® communication satellite. Frank produced many educational computer-generated movies. I will never forget his demonstration of what would happen to a planet if the gravitational law was different from Newton's $1/r^2$. Fascinating!

20.19 Among the Avant-Garde

It was also around this time (1964) that I first met Lillian Schwartz at some (IEEE sponsored?) computer graphics affair in New York City. We immediately took to each other and soon she was working for Bell Laboratories and AT&T, not as a regular employee but as a consultant.

The idea then was to acquaint *bona fide* artists like Lillian with the intricacies of the digital computer. "The computer," we told them, "is just a tool—like a paint brush, albeit more powerful—that you have to learn to use."

This thought was especially propagated by Billy Klüver who went on to found EAT. His "9 Evenings" at a New York City armory is unforgettable and the exhibition at the Brooklyn museum "Some New Beginnings—Experiments in Art and Technology" was a great success. It was co-sponsored by Marion Javitz, wife of Senator Jacob Javitz. Both Lillian and I had several works on display. I was asked to design the poster for the exhibition, an endeavor in which I was assisted by Lorinda Landgraf. (The poster showed a photograph of the Brooklyn Museum composed of letters (of different luminosities) giving the title and venue of the exhibition, the dates and opening hours.)

In my involvement with modern artists, I attended regular meetings in Robert Rauschenberg's downtown loft (planning the "9 Evenings" among other avant-garde activities), where I met David Tudor and other artists. John Cage[16] sometimes came to Bell Labs, where I had the opportunity to demonstrate some of my electronic "music" to him (generated by acoustic feedback through a reverberant hall and a frequency shifter). The composer/conductor Jim Tenney was, in fact, employed by Bell by Max Mathews, as was the French composer Jean-Claude Risset, who became a lifelong friend.

On one occasion, I was privileged to "listen to" Cage's *4'33"* with the (motionless) composer on the stage.

The same performance was also graced by the topless Charlotte Moorman playing her cello. (Later, one of her performances ended with Charlotte burning her instrument.)

In the 1960s, during the height of the Vietnam war, I also had a brief collaboration with Steve Reich when he thought he could use my speech processing equipment for one of his eerie compositions (an endless repetition of the words "Let it all come out") decrying police brutality. (Like Cage, Reich was a pioneer of minimalism, although his music has increasingly deviated from a purely minimalist style. In any case, his compositions have significantly influenced contemporary music.)

New York City was also an ideal venue to see undubbed foreign-language movies—including German films (like *Das Leben der Anderen*—The Lives of Others) which I often found hard to understand without English subtitles.

[16] Cage was perhaps best known for his "composition" *4'33,"* i.e., 4 minutes and 33 seconds of silence (divided into three movements).

Apart from languages, photography —images—have been my premier passion. The pictures that I took as a student evoked admiration: "just like Caspar David Friedrich," the German romantic painter I had never even heard of. (Yes, I was truly uncultured—nekulturnij, as the Russians say.)

When, in the 1950s, I showed my slides at the *Village Camera Club* in Manhattan (attended by such luminaries as Roman Vishniac, Weegee, and Elliot Erwitt), I was applauded for my artistry. One reason, as Norman Rothschild (editor at Popular Photography), told me was that my pictures were "totally original." And the reason for that was that I didn't know of any fashion or trends in photography because I had never been a member of a club of shutterbugs and never studied any magazines that could have given me "fresh" ideas. In fact, I didn't even know that there was a literature on artful photography. So, before joining the New York club, I was completely on my own. No wonder then that my pictures were "totally original": I had no icons to follow. (When I first arrived in the States, I joined the Summit Camera Club. But half their time was taken up by reading minutes and other trivia, so I asked members John Talpey and Paul Kennedy whether there wasn't a more stimulating club in the area and they recommended the *Village Camera Club* on Bank Street in New York City.) One of my main interests was, and still is, *unposed* human portraits—snapshots of people of all ages and genders. At the monthly slide competitions of Bell Laboratories' Murray Hill Camera Club I sometimes won First Prize. But in spite of these successes I never took photography to the heights of my friend Dankwart Köhler who has created many images of breathtaking beauty, now supplemented by digital techniques (see his Internet postings at http://tripod.com/).

In fact, my professional life was largely preempted by research in sound: speech (Linear Predictive Coding), hearing (monaural phase sensitivity, perceptual coding of sound signals), concert hall acoustics (Philharmonic Hall, New York, number-theoretic diffusors), artificial reverberation (virtual sound, soundscapes) and acoustic signal processing in general.

20.20 Philharmonic Hall, New York

In September 1962 Philharmonic Hall at Lincoln Center for the Performing Arts (located on Upper Broadway) had opened to great fanfare with a gala concert under the baton of Leonard Bernstein in the presence of the First Lady of the Realm: Jacqueline Kennedy. But audiences, not least the New York Times music critic Harold Schoenberg, were less than enthusiastic about the new hall's acoustics. For example Schoenberg complained about a persistent echo. I got his seat number and when we looked into this, we discovered that there was indeed an echo. It was easily taken care of. (Ironically, most seats were free from echoes.)

In its predicament, Lincoln Center turned to a trusted friend on *Lower* Broadway: AT&T, which in turn turned to Bell Labs where, being in charge of acoustics research at the time, it ended up in my lap.



Fig. 20.12 "Old buddies": James E. West and George W. Bush (*right*)

A Committee of Four was formed under the chairmanship of Vern Knudsen, Chancellor of UCLA and one of the founders of the Acoustical Society of America. Under the so-called Consent Decree (1956) between the USA government and AT&T, Bell Labs was actually prohibited from acoustic consulting. So it was decided that I would limit myself to making the necessary measurements to analyze the hall. And analyze we did, using computer-generated Hamming-window tone-bursts as an excitation signal and matched filtering on the acoustic output from the hall. I was assisted in this work by Gerhard Sessler, Jim West,[17] Bishnu Atal, Mike Noll, and Carol Bird McClellan, who did the complex computer programming (Fig. 20.12).

[17] Jim West, who was of African and American Indian descent, once, during a scientific talk, warmed my heart by mentioning how considerate I always was about his feeling at home in my area (in which he was the only member "of color"). In 2007 Jim was awarded, in a White House ceremony, the National Medal of Technology for his invention (with Sessler) of electret transducers now ubiquitous in cell phones and numerous other applications. Here is an excerpt from Jim's acceptance speech:

The two billion Electret Microphones made each year have transformed the world of acoustics by providing a simple, inexpensive, very linear transducer for telephones, cell phones, microphone arrays, hearing aids, professional measurements, and outer space communications. The Electret Microphone is the only transducer in mass production that delivers very linear sound reproduction universally in all applications.

It is also important to acknowledge the importance of the electret's engine, the storage of real charge in inexpensive polymers that have been used in air filtration, radiation detection, wound healing, cell development in animal and cell cultures, and piezo-active polymers.

Mr. President, Mr. Secretary, and members of the Nomination Committee thank you for awarding me The National Medal of Technology.

Before starting the measurements, I asked the ushers, students of the Julliard School of Music, whether there was one good seat in the hall and they pointed to Seat A15 on the second balcony as really good. So I decided to include A15 in our measurements. The results showed that in the center of the orchestra floor, there was a loss of almost 30 dB between middle notes (750 Hz) and low notes (125 Hz). By contrast, the loss was only 4 dB at Seat A15! I later demonstrated this effect to Cleveland maestro George Szell who agreed that A15 was indeed superior. Then, after the intermission, the legitimate ticket holder for A15 appeared and Szell and I had to sit on the floor, where it sounded even better to me. But I didn't say anything. Then Szell turned around and said: "Down on the floor here it *really* sounds good!"

What is the *cause* of this lack of low frequencies (which made the celli in *tutti* passages nearly inaudible)? By time-gating the responses on the computer, we were able to isolate different reflections and the culprit was identified as the overhead reflecting panels.

In addition to the poor base response the main problem with Philharmonic Hall (now Avery Fisher Hall) was a feeling of "detachment" from the music. To get at this fundamental difficulty, I was able to persuade the German Science Foundation to underwrite a large-scale study of concert hall quality. My collaborators (Dieter Gottlob and Karl Friedrich Siebrasse) made recordings with a specially designed "dummy-head" in 22 major halls, which were subsequently reproduced in a large anechoic space by means of an electronic filtering method. This was an early example of the now ubiquitous "virtual acoustic images."[18]

The main finding of this study, involving thousands of paired comparison tests, was that for good acoustics, there should be strong early *lateral* reflections—a difficult goal, given that most modern halls are wide and have a low ceiling which favors sound arriving in the median plane of a forward-facing listener's head. To overcome this problem I proposed diffusing surfaces based on number-theoretic principles to be incorporated in a hall's design. The most widely used reflection phase gratings are based on *quadratic residues*. I got the idea during a talk by **André Weil** (brother of **Simone Weil**) on *Gauss sums and quadratic residues* that he gave in Göttingen in 1977 celebrating the 200th anniversary of the birth of the "Prince of Mathematicians."

The accurate measurement of reverberation time also benefited from our work on Philharmonic Hall by using reverse (backward in time or "Schroeder") integration.—Ideally, concert hall measurements should be made with music as an excitation signal. This can be done by measuring the modulation transfer functions, both on the stage and in the audience area and forming their ratio.

[18] Later I became a confidant of Avery Fisher, who had donated $10 million for correcting the acoustics of the hall. Fisher introduced me to the architect **Philip Johnson** who showed me the new design, as suggested by **Cyrill Harris** of Columbia University. (Of course, I couldn't give an impartial evaluation of the design—Cyrill was a good friend.)—Avery invited me once or twice for lunch in the Century Club in New York City. Anny and her sister, Hella, had to wait in the lobby. This was in the days (1975) when the Club was a purely male preserve.

With Atal I also studied sound decays by means of computer ray simulation. The results revealed gross inaccuracies of existing reverberation time formulas and the strong dependence of reverberation time on absorber *location*.

20.21 A Call from Göttingen

In 1969 I received a call from the University of Göttingen as a professor of physics and director of the *Dritte Physikalische Institut*. Of course, Bell Labs was loath to let me go, so—thanks to Max Mathews, John Pierce and Bill Baker—a part-time arrangement was agreed upon for me to spend several months every year at Bell. This agreement held until my eventual retirement on September 30, 1987, exactly 33 years after I had first set foot on Murray Hill soil.

20.22 New York to San Francisco: With Ten Minutes to Spare

In the fall of 1960 I had to give a talk at the Audio Engineering Society at the hotel *New Yorker* in Manhattan. Five or six days later I was scheduled to speak at the Acoustical Society of America in the *Sheraton Palace* hotel in downtown San Francisco. Instead of flying, we opted for driving with our "new" Ford which we had just bought for $100. To waste no time, Anny was parked outside the *New Yorker* hotel on 34th Street—with the engine running. After my talk I hopped into the car and off we went, more or less nonstop, heading west.

All went well until Lake Tahoe when the car wouldn't start. The starter seemed to jam—or something. Anny and I were lost; there was little time to lose to arrive at my meeting in SF on time. Then, seemingly out of nowhere, a kindly "native" appeared and suggested to us recent arrivals to "rock" the car. And sure enough, the starter unjammed and we were on our way for the last leg of our transcontinental trip. After crossing the Bay Bridge, I remember making one or two right turns and, before we knew it, we had reached our hotel—with 10 min to spare before my talk.

After the San Francisco meeting we visited some of the places we had missed because of the rush across the Continent. We went up Mount Tamalpais for a grand view of the Bay, including the Golden Gate Bridge, and we stopped briefly at the historic Point Reyes Lighthouse with a grand vista of the Pacific surf and the beaches.

Of course, in the city proper, we saw many of the "required" sites, the tramway, Market Street, Chinatown, Alcatraz (from afar) and the War Memorial Opera House, which was about to be converted to a concert hall—with totally different acoustic requirements. At the invitation of Joseph Kripps, the new music director, I attended the grand opening in October or November 1963. (My plane was late

because of a snowstorm on the East Coast and I had to change into black-tie getup in the men's room at the Chicago airport. Luckily my taxi driver in San Francisco knew the address, the program, the name of the new conductor and that it was Opening Night so I got there just in time—again with just 10 min to spare. How did the driver know all these things? Well, he just loved music and was building his own little organ in his basement.—Some taxi driver!)

On our way back, we stayed for a night or two at Yosemite National Park with its breathtaking views of mountains, waterfalls, rivers and trees. One morning, Anny, wrapped in just a bath towel, couldn't return to our cabin from the communal bathhouse because a big (brown) bear was blocking her way. But after a while the bear got tired of waiting for the lady and ambled away.

20.23 Baja California: A Vacation

On the West Coast, along the beautiful highway 101, we relished Monterrey, had breakfast on fisherman's wharf, drove the 17-Mile Drive, and enjoyed Mount Carmel until we arrived at our destination: Baja California, which we had chosen for our vacation because of its remoteness and low prices. After Tijuana and Ensenada (Bahia de los Todos Santos), we followed Mexico Federal Highway No. 1 toward La Paz and Cabo San Lucas, the southernmost tip of the Baja peninsula. But we didn't get very far. The "highway" became ever wider and more difficult to navigate with potholes that would arouse the envy of the moon's surface. Finally, at the end of the first day, we saw some lights in the distance: probably human beings that could help us. The lights turned out to be from a ramshackle motel with its own little generator, run by an elderly lady and her sons. Well, we spent the night there and, in fact—we liked the place so much—our entire vacation. This motel was right on the ocean and the sons turned out to be fishermen, specializing in langoustines. They were also good hunters and had an extra shotgun so that I could join them hunting rabbits, quails (*cordonices*) and other small fowl. The mother was a great cook so we persisted for the next 2 weeks on our own rich baggings. For sightseeing there was nearby Mount Kenton, which consisted entirely of volcanic cinder. We never reached the peak of this perfectly cone-shaped mount because two steps up always meant sliding back one or more steps. Since Mt. Kenton was close to the sea, it was easy to mine. The cinder, we were told, was ideal for US road construction.

The bay at which Mt. Kenton was situated was also a favored winter vacation spot for hundreds of whales arriving from farther north. Quite a spectacle: like a can of worms hugely magnified!

On our way back to the States, due to the rough road in Baja, the steering broke and the car preferred to go around in circles instead of heading north. Upon investigation, I discovered that one part of the steering mechanism was secured by three bolts, two of which had broken off. So when you turned the steering wheel, said part turned around the one remaining bolt but the front wheels didn't budge. So, although bereft of any automotive training, I inserted a little wooden peg

(fashioned from a broken-off branch from a nearby bush) into one of the empty holes and—lo and behold—the wheels became steerable and we were on our way again.

Two stops that I remember were the huge asteroid crater in Arizona and the Petrified Forest. From there we called our friends, Max and Marj Mathews and learned that Anny's mother, Wanja, who had stayed behind caring for our youngest son Alexander, had fallen ill. So Alexander was transferred to the Mathews.

Since we had "seen it all," Anny and I decided to call it a day and drive nonstop back to New Jersey—2,300 miles (3,700 km). While on our trek west we had counted the tankfuls of gas, we now went by the sun: sun down—sun up—sun down—sun up for 63 h. But in Indiana, near Terre Haute, we got so sleepy that we checked into a motel to grab 5 h of sleep, a shower and something substantial to eat.

Outside Zanesville, Ohio, we were stopped for speeding by a patrol car that was in contact with an overhead "spy" plane that monitored the speeds of cars as they went from a 70-mile/h zone on US 40 to some 50 miles/h—rather unfair. But the police officer was very kind: he let me listen to the radio traffic from the plane.

We were hauled before a judge in the Zanesville court house where I was fined $20. But the trouble was we didn't have that much cash left over after the long trip and the county cashier couldn't accept checks. What to do? The alternative was going to jail for a long weekend. We called my bank in Summit but they were closed.

Then I remembered *Ma Bell*, the phone company. Bell employees had been told that, if they ever needed help, including money, they should go to the nearest phone company office and show their Bell Laboratories pass. The police officer volunteered to drive me there but I asked him to park two blocks away: I didn't want to be seen arriving in a police cruiser.

My reception at the phone company was less than cordial. Small wonder: I was unshaven and had a big hole in my sweater and of course spoke with a foreign accent. After seemingly endless questioning the Bell bureaucrat finally agreed to *endorsing* my check for $20 (no cash changed hands). I can still hear the office workers snickering behind my back because their boss "had been had."

Unfortunately, after I had secured the $20, we were back to square one, by my asking the Bell person to endorse a $100 check (to buy food for the weekend) because that really convinced him that something was amiss. Then, after about 30 min, Anny appeared and said the waiting police officer was getting impatient and feared that I had fled by the rear door. I told Anny to placate the officer—that I was making progress and would return shortly.

Thirty-four years later, in 1994, during another safari across the country, we revisited Zanesville by briefly driving off Interstate 70. The town made a desolate appearance with many stores on the main street boarded up. The only remaining industry appeared to be ceramics (pots and vases). But the County Court House was still standing gloriously. And on a side street next to it was an Automatic Teller Machine (ATM) where I could have easily drawn a lot more cash than $20 (if I had needed it).

Chapter 21
The KGB

Manfred R. Schroeder

In early 1963, I received an invitation from the "State Committee for the Coordination of Scientific Research" of the Soviet Government to "see" their work in speech coding. Bell management felt leery about my going as a German national and strongly recommended that I become a US citizen first. Before leaving for the "evil empire" (before it became known as such) I was extensively briefed by the CIA about the potential pitfalls I might encounter. Thus, if confronted with photos showing me with another man in bed, I should not doubt the veracity of the images but respond cheerfully: "Oh, how lovely! Could I have an extra print for my wife and—come to think of it—one for my boss, too?" Of course, nothing of the kind happened.

However, one day, while waiting at a busy intersection in Moscow, with my escort, Victor Ryazantsev (a KGB major—but I didn't know it at the time), I noticed several extraordinarily beautiful women on the other side of the street. Victor saw me "noticing" and said, deadpan, "Would you like one of them? They are working for us." I politely declined.

Another useful advice the CIA gave me was to never close my hands on a package that someone handed me on the street but let it drop to the pavement. This is precisely how the KGB entrapped Harvard professor Frederick Barghorn in the summer of 1963. (The parcel contained Soviet state secrets.) When President Kennedy heard about this, he blew his top: "How on earth dare the CIA turn a good Harvard professor, a friend of mine, into one of their spies!" The CIA swore that they did no such thing. Whereupon Kennedy went for the "hot-line" and berated Nikita Khrushchev. Now it was Nikita's turn to get angry at his KGB chief: "Didn't you nitwits know that Barghorn was a friend of Kennedy's? Next time pick a better target for your dirty tricks." Barghorn was released from his

M.R. Schroeder
Author was deceased at the time of publication.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_21,
© Springer International Publishing Switzerland 2015

Moscow prison forthwith and, with some face-saving apologies, returned to the USA. He immediately went to Washington where he thanked President Kennedy personally for his intervention. (I believe the oval office encounter occurred in October 1963; a month later the president was dead. All this was covered *in extenso* in the New York Times at the time. But you can also find bits of the story by googling "professor barghorn" in the Internet.)

I was invited to give a lecture during this trip, with a translator standing next to me, translating my lecture into Russian. Thus I had to stop speaking every two or three sentences. During my introduction I stated: "I work for Bell Laboratories, a company of 14 thousand employees." The Russian translator went to work (in Russian): ". . .Bell, a company of 14 employees." Without hesitating, I spontaneously exclaimed *in Russian*: "Nyet! Nyet! *14 thousand*" ("Het! Het chetirnadtsat' tisyach"). The whole room broke out in roaring laughter. One of my hobbies was foreign languages and I was following every word of the translator with great interest and curiosity—to his surprise. The poor translator's face turned awfully red and he was replaced a few minutes later by another translator. (I was never quite sure if the size of Bell Labs was purposefully being down-played to the audience, but the spontaneous humor of the situation was priceless).

My 1963 visit was my second trip to the Soviet Union, after Anny and I had gone as tourists during an earlier visit. I recall waiting times of 2 h or more, and once we went without food for 28 h because we simply didn't have the necessary time it took even for a simple meal. When (after a superb puppet show) we finally got back to our hotel, the old, Victorian-style, *Astoria*, all restaurants were closed. Same luck at the other big hotel for foreigners, the *European*.

Tourists from the West were really a novelty in Russia before 1959, so people often gathered around us. Someone in the crowd outside the hotel pointed out a very beautiful, dark-haired woman to us. Could we believe that she was Russian? I said "yes" because that's what they wanted to hear, I think. (I believe she may have been Jewish or from one of the southern republics of the Soviet Union.) Someone else insisted that there was a mile-high skyscraper in the USA (he must have had a *design* by Frank Lloyd Wright in mind). But we were too hungry to argue. Food! Where is food? Yes, there was a snack bar on the fifth floor—but they were about to close. All they could serve at this late hour was caviar, white bread, butter, and champagne. Well, we didn't mind. It was a glorious finale to a very long day.

Now, instead of waiting hours for service at a restaurant, Victor always introduced me as a "guest of our government" when we arrived at a new location, and service thereafter was always near instantaneous.

One of my tasks in Moscow (other than ascertaining the state of Soviet speech research) was to advise the Russian acousticians on the sound system in the Kremlin's new *Palace of Congresses*, seating 6,000 delegates during Party congresses and being used to supplement the Bolshoi Theater for opera performances. The hall was relatively dead acoustically and the required reverberation for music was "manufactured" in the basement and piped into the main hall by a large number of (hidden) loudspeakers. I heard Glinka's "A Life for Tsar" (in Soviet times called "Ivan Susanin") there and liked it. Yet, for the final scene the audio engineers

couldn't contain their enthusiasm and turned up the audio volume to an ear-splitting level. So much for *artificial reverberation* (one of my specialties)!

Once Victor and I took the night train (*Red Arrow*) from Moscow to Leningrad and upon arrival, on a beautiful June morning, he told me that, me having been in Leningrad before, I sure knew my way around and didn't need to be escorted. Would it be alright with me if he disappeared for a couple of days? He had been a student in Leningrad and, though married in Moscow, he still had an old girl friend whom he had not seen in a while. Of course, I had no objection, and we parted company right then and there on Uprising Square in front of the main train station. (I mention the name of the square because I am thrilled by its sound in Russian, as I am thrilled by the sound of so many other Russian words and phrases.)

It has been pointed out to me that Victor's disappearance may have been a ruse: that I was closely followed and observed during my entire stay in Leningrad. This is quite possible.

Shortly after arriving in Moscow around 16 June 1963, I heard the voice of Valentina Tereshkova, the first woman cosmonaut speaking from her spacecraft to Nikita Krushchev: "Dorogoy Nikita Sergeyevich. . ."[1]

On the last day of my state-sponsored visit in 1963, Yevgeniy Levin, head of the Foreign Department of the State Committee for Coordination invited me to dinner at Moscow's famous *Aragvi* restaurant (named for a valley near Mount Kazbek in Georgia). (Victor came along too but had to excuse himself after a while—the Stolinichnaya vodka was flowing a bit too freely.) The *Aragvi*, in addition to its ancient décor, boasted venerable sterling silver cutlery. Levin noticed my acute interest in one of the finely chiselled forks. "Take one with you, as a souvenir of Old Russia" he said. Of course, I declined the offer. After all, I had to pass exit controls by the Soviet border police the next morning. "Don't worry," Levin assured me, "tell them that Yevgeniy Levin gave you permission." Later I learned that Levin held the rank of Lieutenant General. As a polite guest, although not knowing Levin's high rank then, I finally did obey orders and took the fork with me across the Iron Curtain back to the USA. For a long time after, the *Aragvi* fork was the only sterling silver we owned, a fact that does not seem to have escaped some discriminatory thieves: One day, after returning to New Jersey after a longish sojourn in Europe, the temporary tenants had left everything in our house in place—except, alas, the treasured fork.

The Russians are traditionally a very hospitable people and I had the pleasure of experiencing their warmth on several sallies into the Soviet Union. My surprise was therefore complete when, one cold February night, upon arrival in Leningrad, nobody was at the airport to meet me—no flowers, no friendly handshakes, no nothing: nobody I knew as far as the eye could see. After pacing up and down the

[1] I recently met a salesclerk at a supermarket in La Jolla, whose first name was Valentina. Had she heard of Valentina Tereshkova? "Sure, I was *named* after her.—My parents, although Serbs, were very proud of the first woman in space, coming from the Socialist Camp."

crowded arrival hall for a while not knowing *where* to turn (I didn't even know the name of my hotel or the venue of the meeting) I finally turned to the sole "familiar" character in the thick crowd: the man I thought might be from the KGB—although in mufti, easily identifiable to the frequent (and not-so-frequent) traveler behind the Iron Curtain by his mien and demeanor. I went up to this person and said in my minimal Russian: "I, Dr. Schroeder from the Federal Republic. To attend All-Union Conference on Acoustics. Please, which hotel?" Without blinking an eye the stranger, incredibly, answered "You, *Oktyabraskaya Gostinitsa*." Two minutes later I was in a taxi headed for the Hotel October where I encountered all my friends who had expected me on a flight 2 h later.[2]

Imagine arriving at Kennedy Airport in New York City and telling a policeman you forgot the name of your hotel and being told, without a moment's hesitation exactly where to go. Yes, it's difficult to get lost in the Evil Empire—but of course, it didn't last. And I still don't know why the KGB operative at the Leningrad airport blew his cover by answering my question (correctly, as it turned out).

It is quite possible that this story, given the KGB's passion for deception, has a more sinister background too, namely that my Leningrad friends were told (falsely) that I would arrive on a later plane so that the KGB could better observe me.

One thing the Soviet secret services were particularly good at was "word spotting," now subsumed under the ubiquitous "data mining." At some time in the 1960s, Russia had suffered another disastrous wheat harvest. They had to make up the shortfall from imports, mostly from Argentina and the USA, at the lowest possible price. For this purpose they tapped into and monitored zillions of telephone calls. Their excellent word spotting program was set to recognize the words "wheat" and "price." The thus vetted calls were then examined by human listeners for clues to the US pricing strategies. This stratagem, which was unmasked only much later, is said to have saved the Soviet Union (and cost the American farmer) many millions of dollars.

How did the Russians manage to tap, without being discovered, into that many US telephone calls? Easy! They bought a hillside lot for their new Embassy in Washington which "happened" to lie directly in the path of a major telephone microwave link. Hidden antennas in the embassy did the rest—without anyone being the wiser.

21.1 Shinjuku, Shinjuku: Lost in Tokyo

When Anny and I went to Japan for the first time in 1968, we were told that, as Americans, we shouldn't try to speak Japanese. So, for our first foray into Tokyo, we dutifully asked the hotel porter to prepare a little note for the taxi chauffeur telling him that we wanted to go to the Keyo Department Store in *Shinjuku*. The

[2] This story first saw the light of day in a Letter-to-the-Editor in the Travel Section of the *New York Times* on 8 April 1984.

driver nodded and went off in some direction. But before long he doubled back and, after some more criss-crossing, we passed our hotel again. A little later, he stopped at an intersection and asked a policeman for help, but to no avail; we were still lost. Everyone he asked pointed in a different direction. Finally, I thought, it couldn't get much worse if I intervened and uttered a single word: "Shinjuku!" The driver turned around and asked "Shinjuku?" And I repeated: "Shinjuku, Shinjuku, Keyo!" And in no time we reached our intended destination, the Keyo Department Store in Shinjuku.

How did this happen? Did the hotel porter not understand us? Or was he unable to write? Or was it that the cabbie couldn't read the porter's script (and was too polite to say so)? We don't know. But what we *do* know is that it is difficult to navigate the Tokyo streets. Tokyo isn't Manhattan: its streets are like a weird warren. This was nicely brought home to me on our second day in Tokyo. Peter Denes and I had been invited to visit the *Hitachi Central Laboratories* and they had sent one of their limousines to fetch us at the *Okura* hotel in the center of town (opposite the US embassy). On the way back to Hitachi, which was after all his home base, our chauffeur seemed less and less certain as to where he actually was. He had to stop several times to ask *children playing ball in the street* for directions. These must have been very smart kids: before too long we sailed into Hitachi headquarters, happy and relieved, to be greeted by our hosts.[3]

But Tokyo taxi drivers can also be very helpful. I once left my parka in a cab at Kichijoji train station on our way to the island of Shikoku. A friend dashed after the cab, but too late: the vehicle was gone. When, a few hours later, I arrived at my hotel in Shikoku, my "lost" parka was already waiting for me at the reception. What had happened? The cabbie, on noticing the forgotten garment, drove back to the house where he had picked us up. There he encountered a secretary who was waiting for the gasman to read the meter. It so happened that her boss, too, had to fly to Shikoku and he kindly took the parka with him and delivered it at our hotel.

The zenith of Japanese service happened to me during a semi-formal dinner in Tokyo: I had spilled some soup on my pants and was asked to step behind a screen to undress. I was given a kimono and rejoined the dinner. An hour or so later I was

[3] I just bought (September 2007) a GPS (Geostationary Positioning System) navigator that I love and which would have come in very handy in Tokyo (TomTom ONE for $199 at Amazon.com). These little machines are veritable marvels. They have to use Special *and* General Relativity—or their readings would be off the true location. For one, the GPS satellites are in a weaker gravitational field, making their (atomic) clocks run faster than on earth. (So much for Heisenberg's insistence that General Relativity was just a theory with no practical consequence.)

It takes only a few seconds to connect up with up to seven geostationary satellites in the sky. And in a fraction of a second the device will consider *millions* of different road combinations. (As a test, I once asked the machine (stuck to the windshield with a suction cup) for the fastest road from our house to the Rickshaw Garage in New York's Chinatown. Answer: 28.4 miles—34 min. Right! Then I asked for a *toll-free* connection and the machine, unfazed, found one—via Albany, 7 h and 58 min.—Very fast search algorithms are in fact one of the marvels of the digital age.)

again motioned behind the screen: my suit had just returned from the dry cleaner and I was ready to partake of the proceedings in proper western dress.

Once, in the *Okura*, we asked for "sweet butter." After 20 min the waiter returned. He had searched the whole hotel, he told us: "There was no *sugared* butter anywhere in the *Okura*."

Chapter 22
Professor of Physics

Manfred R. Schroeder

Around 1966 I received some "feelers" from a physicist friend (Wolfgang Eisenmenger) whom I had invited for a summer at Bell Labs, where he invented phonon spectroscopy (in collaboration with A.H. Dayem). Was I interested in a full professorship at the University of Göttingen? I was very happy at Bell Labs and didn't know how to say "no". So I said "yes, I am interested." (I mean, how can you possibly say "no"—it would be impolite.)

When the offer was actually made, in late 1968, I was glad to accept. I had spent 14 years at the Laboratories in New Jersey and my only way "up" was, well, up: less research and more management, which I detested. Thus, in March 1969 I made a side trip to Göttingen (during a skiing vacation in Kitzbühel, Austria, where I broke a leg) to discuss the situation. They had a long list of equipment and additional personnel: seven people, including one professor and several *Assistenten* (assistant professors) that I should ask for at the state government in Hanover. One item (for 200,000 marks) on the list was "*Prozessrechner*" and I said: "What's that?" "That's what you call an *on-line computer*" (for doing real-time experiments). But I thought such a computer would really cost over a million marks. Yet I didn't want the secretary (Frau Edith Kuhfuß) to type the whole page again so asked her to put a "1" in front of the 200,000. This figure of 1,200,000 marks was accepted in Hanover (actually, the *VW Foundation* footed the bill which, in the end, came to 2.3 million).

Then came the question of my salary. Instead of answering, I showed the kind official, Ulrich Hopfe, my last monthly pay slip from Bell showing 39,000 dollars per year—then about 150,000 marks. "Well", said Herr Hopfe, "we can't compete with *that*, but we can offer you our highest rate plus all the raises until retirement", which came to about half my American salary. I said "Fine!" and accepted and I am

Author was deceased at the time of publication.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_22,
© Springer International Publishing Switzerland 2015

still glad I did. Soon the dollar started dropping from 4 marks to the dollar to around 3.70 (September 1969) and it kept dropping as President Nixon cut the dollar loose from the gold standard.

Later, while already in Göttingen, I discovered that the new computer didn't fit into the space I had planned for it. Since there was no other free space available, we needed an annex to the present buildings. Of course, the *Kurator* (chief financial officer of the university) wanted to know whether I had the money for the extra building. And I lied and said (like a good groom) "yes, I do". I immediately called Ulrich Hopfe (whose telephone number I always kept on my desk) in Hanover and told him what had just happened and he said (I will never forget) "Don't worry, you *have* the money." Period. He rescued me from a very bad situation and the building was built on time.

Twenty-one years later I invited Hopfe to my farewell lecture on 5 July 1991: "*Die schönsten Experimente aus der Schwingungsphysik, vom Hören und Sehen und zur Zahlentheorie*" ("The most Beautiful Experiments from 'Swinging Physics', Hearing and Seeing, and Number Theory".) And Hopfe, long since retired, came—in a wheelchair.

Of course, Bell Labs didn't want me to leave so a part-time arrangement was reached with the state government in Hanover that allowed me to do research at Bell while I wasn't teaching in Göttingen: my transatlantic-commuter life began—still ongoing 38 years later!

Before becoming a German professor, I had to swear an oath of office. For that I went to the Rector's office to be greeted by a very friendly "*Magnifizenz*", Hans-Heinrich Voigt, an astrophysicist. When his secretary brought in the folder with the oath, he said "no, the *other* one". What was going on here? Voigt explained that, as a foreigner, I didn't have to swear allegiance to the German Federal Republic and that the "other" oath, the one I swore, made no reference to Germany. Thus, I simply promised that "I would obey the laws of the land".

There is an interesting bit of history behind these oaths. After the war, Germany decided that you could become a civil servant (like a professor) without becoming a German citizen (exceptions: federal judges, armed-forces officers, and President of the Republic). Einstein, after accepting his triple appointment in Berlin (professor of physics without teaching duties, member of the Prussian Academy of Sciences, and director of the (new) *Kaiser-Wilhelm-Institut für Physik*) had, to his dismay, become a German citizen again. (He was Swiss, after having been born German.) Hitler, before being able to run for president, had to become a German. (He lost his Austrian citizenship in 1914 when he joined a Bavarian infantry regiment to fight in World War I.) In 1932, a high-level Nazi in the Brunswick state government made him a *Landvermesser* (surveyor), which gave him Brunswick citizenship and, by the German citizenship law of 1913, German citizenship as well—with the well-known disastrous consequences.—There is an extant photo showing Hitler and two buddies

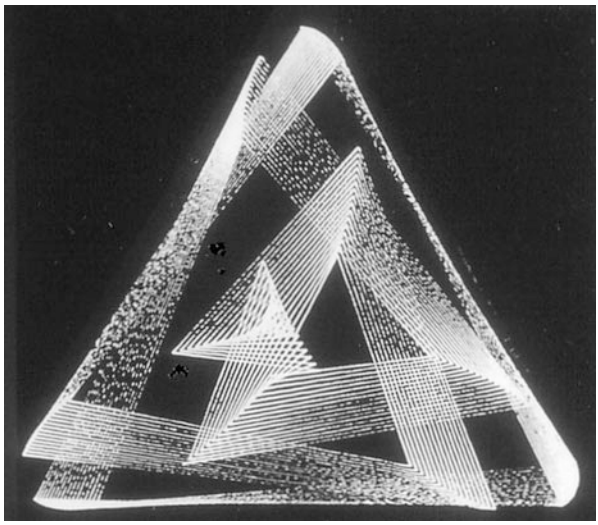


Fig. 22.1 Heinrich's "Broken-Symmetry" triangle

"on the way to the Brunswick legation in Berlin" to pick up his citizenship papers: He didn't even have to travel to *Braunschweig* to become a citizen of that state.[1]

More recently, Hitler and Einstein crossed paths again, posthumously, when *Time Magazine* ran a competition for *Man of the Century*. Einstein won; Hitler was second. (The question was "Who had affected the 20th century the most—for better or worse?") It is interesting to note that Einstein and Hitler were born within 150 miles of each other and both named Karl May (American Indian exploits) as one of their favorite authors as adolescents. (But I doubt that Hitler ever studied Baruch Spinoza, as Einstein did as a young man.)

My teaching duties at the *Institut für Schwingungsphysik* (literally "swinging physics") consisted of a weekly 2-h lecture with demonstrations on a great variety of subjects: acoustics (including speech, hearing, and concert halls), electronics, coherent optics, electromagnetics, computer graphics (one of my hobbies), and applications of number theory (my true love). I really liked preparing the numerous demonstrations. The only topic I didn't like was solid-state physics ("*squalid* state physics" as nobelist Wolfgang Pauli called it).

My lectures on computer graphics struck a particularly sympathetic chord with the students, some conceiving their own creations, like the "broken-symmetry" triangle by Gerhard Heinrich, which I find very appealing (Fig. 22.1):

Best of all, I never had to teach or test undergraduates. Our institute was what would be called a graduate school in the USA. The students were highly motivated,

[1] According to the *International Herald Tribune*, in a report dated 14 February 1932, Hitler was appointed a professor of practical pedagogy at the Technical Academy of Brunswick. But Hitler's likeness on a postage stamp of the *Reichspost* was once featured in a talk entitled *Berühmte Geometer* (Famous Geometers). Others so honored were Gauss and Eötvös, who became the prime minister of Hungary.

intelligent, and often smarter than the professor. And, of course, there were no "classes" or, perish the thought, "credits". If a student was interested in a topic, he would attend the relevant lectures. If not, he would drop out. I usually started with some 80 students at the beginning of the semester and ended up with about 40—par for the course (no pun).

When I first arrived, they gave me a little "torchlight" parade (actually lit candles): I was of course tickled pink. When my Russian friends in Leningrad (Ludmila Chistovich and Valery Kozhevnikov) asked me to characterize my new position in Germany I ventured "*Ya polubog*", which they immediately understood. (I knew the prefix "polu" from Russian champagne: *polusukhoe* = demisec. And I knew the word for god (*bog*) from Old Church Slavonic.) Yes, I had become a demigod. But while I used the term tongue-in-cheek, for some of my older colleagues it was gospel: they actually believed in it (and acted like demigods).

22.1 A Brief History of Speech Research: Early Speaking Machines

An early highlight of speech research was the work of the Hungarian count Wolfgang (Farkas) von Kempelen which resulted, in 1771, in a speaking machine that was widely exhibited at princely courts around Europe. Even Goethe in Weimar and Lichtenberg in Göttingen heard live demonstrations of von Kempelen's contraption. But while Goethe found some nice words to say about the synthetic voice (*die verschiedne kindische Worte und Töne ganz artig hervorbringt*), Lichtenberg was his usual acerbic self by pronouncing the machine incapable of saying much beyond *mama*, *papa*, and *Roma*.

The next advance came with Christian Gottlieb Kratzenstein from Wernigerode who, in 1779, won the First Prize from the Imperial Academy in St. Petersburg for designing (wooden) resonators capable of uttering the German vowels *a*, *e*, *i*, *o*, *u*. Thus, the resonance theory of vowel production was born, later elaborated by Hermann von Helmholtz and Lord Rayleigh.

One of the most useful practical applications growing out of this work was the invention of the telephone by the German high-school teacher Philipp Reis (1860) and the Scottish speech therapist Alexander Graham Bell (working in Canada and the USA). In America, the invention of the telephone led to the formation of the American Telephone and Telegraph Company (AT&T) in the late 19th century which in turn, in 1925, incorporated *Bell Telephone Laboratories* in New York City (463 West Street).

Apart from solid-state physics (transistors, solar cells) and information theory (Shannon's channel capacity, error correcting codes), Bell Laboratories was a strong supporter of fundamental research in hearing and speech, exemplified by

the invention of the *Voder* and *Vocoder* (from **Vo**ice **Coder**) by Homer Dudley[2] in 1928, first demonstrated at the New York World Fair in 1939. (The vocoder played a crucial role in World War II by enabling a secret telephone link between Churchill's War Room in London and Roosevelt's White House in Washington.)

The vocoder, by analyzing and synthesizing speech signals, could be made to sound like a pipe organ or a waterfall or a VW engine (groaning for more gas). In its most recent incarnation the vocoder (having morphed into *linear predictive coding*[3]) inhabits every cell phone and most Internet speech compression.

22.2 Speech and Hearing Research in Göttingen

Some of the speech research pursued at Bell was transplanted to Göttingen in 1969 when I succeeded Erwin Meyer as director of the *Dritte Physikalische Institut*. Much of our speech research at Göttingen involved the application of measuring methods from physics to the human speaking process.

In view of the importance of human hearing for the proper encoding of speech and music signals, extensive studies of perceptual masking of one sound by another were undertaken by Sönke Mehrgardt and many others. This research was led by Birger Kollmeier and Armin Kohlrausch. Some of the work was concerned with the design of better hearing aids and tests of speech intelligibility for hearing-impaired listeners.—An important portion of the ongoing work in speech is aimed at improving diagnostic tools for voice pathologies (in collaboration with professor E. Kruse of the Medical Department of the university). Jan Lessing, my most recent Ph.D. (July 2007), developed a sophisticated acoustic method to assist in the diagnosis of various malfunctions of the human "voice box".

In December 1970 I gave my inaugural lecture "Can computers be taught to speak?" At the end I had an (American) computer recite a German poem:

> Mit deinen blauen Augen
> siehst du mich lieblich an.
> Da wird mir so träumend zu Sinne,
> daß ich nicht sprechen kann.

The Bell computer, with the help of Noriko Umeda and Cecil Coker, had been trained to properly pronounce the German *r* and *ch*. But even so, nobody in the audience understood. Then, without mentioning it, I projected a slide with the text and repeated the computer speech. Now most listeners understood. Why? Because they had (unwittingly) *read* the text.—This is a famous trick to convince sceptic

[2] Homer was convinced that the State Department was infested by communists and later, in 1957, that the beep–beep–beep of *Sputnik* was nothing but electronic deception—the Russians couldn't possibly be the first in space. Of course Homer was not alone in these superstitions.

[3] Correspondences with Bishnu Atal and J.L. Hall during development of LPC are archived at the University Library Archives of the University of Göttingen.

listeners that your synthetic speech is actually intelligible. (Of course, I told the August audience afterwards what was going on.)

After the lecture Frau Eggers—mother of my friend Frieder—came up to the podium: "*Sagen Sie mal, Herr Schroeder, das war doch Heinrich Heine?*" "*Ja, Frau Eggers, das war Heinrich Heine*".

Naturally, as a new professor, I was immediately condemned to some onerous committee duty: the University Senate Planning Committee. One of our tasks, under the able chairmanship of Rector (university president) Eduard Lohse, was to plan the new physics department buildings. This was in 1970, but it took until 2005 for the move to the North Campus to be completed. (First money ran out, then some of my colleagues didn't want to move, etc.)

Much later I was elected to the *Kommission für Mathematiker Nachlässe*, concerned with the unpublished papers of famous mathematicians (Gauss, Riemann, Hilbert, Noether). Since little was known about the papers of Emmy Noether, one of the greatest mathematicians of the 20th century, I was glad to have succeeded, 72 years after her death in 1935, in locating her surviving nephew Hermann Noether in Summit, New Jersey.

Emmy was forced to flee Germany in 1933 because of her Jewish descent. She became a professor at Bryn Mawr. Her brother Fritz emigrated to Tomsk, Siberia, where he became a professor at the Institute for Mathematics and Mechanics. In 1937, during the Great Purge, he was arrested as a "German spy" and sentenced to 25 years imprisonment. In 1941, he was in a Soviet camp near Orel, where he was shot on September 10th for spreading "anti-Soviet-propaganda". Some 160 other prominent political prisoners were shot the next day on Stalin's orders in a forest outside Orel. (The true reason for these hasty executions may have been the approach of the German army, which, in fact, captured Orel on October 3rd.)[4]

Another commission I served on was the *Prüfungskommission* (examinations commission) that had to settle contested cases. I remember one unfortunate student who had flunked theoretical physics in the orals twice in a row and he was not eligible for a third try. I pleaded with my colleagues on the commission to make an exception (the candidate had obviously been nervous). "No," they said, "that would set a bad precedent." I didn't see how we could nullify a student's 4 years of labor by just one flunked test. But they wouldn't budge. So the case went to the top legal eagle of the university (Herr Bahlsen) for arbitration. Bahlsen was known to be very pedantic and I saw all hope for the student fade. But Bahlsen asked a single question which saved the case: "Was the required extra witness present when the test was repeated?" No! so the second test was declared invalid and the student could take it one more time. I made sure that the examining professor was replaced by a less threatening person and the student passed and got his degree (M.S.).

One of my older colleagues actually, in the middle of the oral examination, threw out a Ph.D. candidate whose answers he didn't like (Max Delbrück

[4] Orel (pronounced arYOL) is the Russian word for eagle. Orel was liberated by the Soviet army on August 5, 1943 (see Chap. 16 of my memoirs).

(1906–1981), who later created the field of molecular biology and won a Nobel Prize for it). Talk about demigods!

When I had to examine a student myself, my first question usually was "Are you right or left handed?" (If the student was left handed, I sat on his right so his arm wouldn't block my view of his jottings.) This was a simple question that no student had any problem with. My next question, taking a leaf from Richard Becker, was "What would you like to talk about?" which some students found difficult to answer because they were expecting the typical quiz. Then I let them talk freely, only occasionally interrupting with a question. This way it was more interesting for me, often learning something new. And it certainly was nicer for the student. Also, I could learn more about the student's abilities than by asking questions from a "catalogue", which the students were known to have memorized. Sometimes, at the very end of the 30 min, I would ask a pointed question or two, like "What is the spin of the photon, the electron, the graviton?"

My demonstration lectures, in which I was greatly assisted by Heinrich Henze, continued the tradition of Robert Pohl and Erwin Meyer. But my style of lecturing must have been somewhat unorthodox. Once an American student told me after a lecture that I reminded him more of an American professor than a German.

Some of my experimental demonstrations were based on number theory, such as the "2-squares theorem", which allowed me to create circularly *polarized* sound waves. This was a small sensation because sound waves are *longitudinal* and—in contrast to light—have no polarization.

A few of my demonstrations were deliberately misleading, for example when I "showed" that microwaves traveled through a hollow plastic hose while they actually cling to its outside (depending on the wavelength and the diameter of the hose). (Of course, in the end I allowed the truth to emerge by plugging the hose shut—and the waves continued propagating.)

One of the finest accolades I received from a student was from Joachim Neumann, who wrote on his webpage, next to a picture of me, "He taught me that physics is show business". (Some colleagues, I am sure, would be outraged by such a statement.)

One of my academic tasks other than giving lectures was selecting thesis topics. At any one time, there were on average some 20 theses, leading to the degrees of *Diplomphysiker* (M.S.) and *Dr. rer. nat.* (Ph.D.), that I had to supervise. I was much helped in this task by the untiring and conscientious Dr. Hans Werner Strube, as well as Birger Kollmeier and Armin Kohlrausch, who later became professors at prestigious institutions in their own right.

My "first professor" was Reiner T., who was in danger of not getting a job in Germany because he had a police record (participation in some leftist student demonstration). I advised him to emigrate to the USA and never mention his record of "unruliness". I wrote a warm recommendation and after a few years he had become a *professor*. Of course, he was very smart. Before taking him on as a graduate student, I asked him "What is the numerical value of i raised to the power i?" (i is the square root of -1.) After a moment's hesitation, he ventured "Oh, about

1/5"—which is the right answer! (I have asked some mathematics students the same question and they didn't get it.)

Shortly after I arrived in Göttingen in 1969, at the height of the Vietnam War and the worldwide student protests, the students called a *Vorlesungsboykott* (boycotting the lectures). However, the students still appeared at my lectures. So I asked them whether there wasn't a strike on. Yes, but they wanted to hear my lecture anyhow. These were graduate students eager to complete their education. Sitting on the front table, feet dangling (to signal that I didn't have any particular axe to grind), I said, "let's have a vote whether I should give the lecture". All but three students voted "yes". To the three strikers I said that I would give them a private tutorial to catch up. Talk about taking the wind out of a protest!

One of the three students, now a professor, admitted his "sin" during a recent encounter, hoping that I really wouldn't remember him. I said, "Yes, I do remember you, but I never held it against you". (His name was Gauss—no relation though to the great mathematician.) In some departments (philosophy, *Germanistik*) there were violent sit-ins and in one case the professor was locked in (or out—I forget which) of his office. Albrecht Schöne, the university's star *Germanist*, simply worked at home.

One of my Ph.D. students is Bennet Smith, an American, who later worked with Pierre Boulez at the Centre Pompidou in Paris. He didn't finish his thesis (on the peripheral auditory system) while in Göttingen, so I carried him as an external student. Whenever I visited Paris he asked me whether my offer that he could get a Ph.D. from Göttingen was still good. I advised him to closely observe the obituaries: as long as he didn't see my name there, he could still come to Göttingen and complete his thesis. (Twenty years later I am still waiting. But I guess he made his way without the degree.)—Bennet, being American, once, during colloquium, sat in the front row, traditionally reserved for professors. The second row is for assistant professors, etc. Students should not seat themselves in any of the three or four front rows—a despicable holdover from more authoritarian times. But I assured Bennet that he could sit wherever he liked.

Apart from Bennet Smith, I had several other external Ph.D. students: Jan Lessing, working at the Göttingen clinics (in phoniatrics—the subject of his thesis: telling from a recording of a voice what's wrong with the vocal chords), Olaf Schreiner working at DaimlerChrysler for a while (on automatic speech recognition in cars) and, until recently, Holger Quast working at Bosch (on spoken dialogue systems: airline reservations and the like). Holger passed the orals and got his degree on 26 January 2006. Of the five-member Ph.D. commission two colleagues had voted with me for "summa cum laude" and two had opted for "magna cum laude". So, the majority was for *summa* and I awarded Holger the "summa cum laude". (Later, one busybody colleague discovered that for *summa* the vote had to be unanimous. But I advised Quast that he could always say that his professor gave him a *summa*.)

Besides Holger, one of my star students was Tino Gramss. After his Ph.D., he could look forward to a distinguished academic career. But he was riven by self-doubt—not made better by jilted love—and chose to end his life at a very early age.

Some of my top students, like Sönke Mehrgardt, became "captains of industry". Others became professors or opened Software and Acoustics consultancies. Some rose to high academic positions, such as K.J. Ebeling, who became the president of Ulm University.

Most of my students were working in the fields of speech, hearing, architectural acoustics, and applications of number theory. Once a week we all met in a joint seminar. Some of the work in hearing was psychoacoustic (investigating special properties of human audition), but some involved animal physiology and animal hearing. The work with animals was done in a very fruitful collaboration with Otto Creutzfeldt at the Max Planck Institute for Biophysical Chemistry. I remember one Ph.D. thesis (by Rainer Koch) on guinea fowl (*Perlhühner*). Koch had electronically synthesized various "calls" by the fowl (danger, mate seeking, etc.). But his animals wouldn't budge; they only reacted to natural calls in spite of the fact that both natural and synthetic calls sounded the same to human listeners and looked exactly alike on the sound spectrum analyzer. What was going on? I thought maybe the fowl can perceive fine temporal detail that the human ear cannot perceive and the analyzer wouldn't show. After Koch modified the analysis to show the fine time-structure, he discovered that the latter was missing in his synthetic calls. After incorporating the appropriate temporal detail in his synthetic calls, the birds *would* budge as expected. Later Koch discovered that his birds were mostly sensitive to time rather than frequency detail. In other words, you could play the birds almost any melody, as long as the rhythm was right.

22.3 Underwater Sound

Research at our institute was underwritten mostly by the German Science Foundation (*DFG*) except for contracts we had with the Rome[5] Air Force Command and the British Admiralty for (unclassified) underwater sound research, supervised by Werner Lauterborn, Dirk Ronneberger and Dr. Dieter Guicking. Professor Erwin Meyer had insisted, back in the late 1940s, that we should be allowed to publish all our results, which of course precluded "secret" work. During my tenure, Leslie Kendrick was the Admiralty's representative and Leslie—and his wife Mary—became lifelong friends.

Besides the university and the Max Planck Society (of which I became a "Foreign Scientific Member" in 1972), I was elected, in 1974, to the Göttingen Academy of Sciences (founded by George II in 1751). This Academy holds *plenary* sessions every other week while the university is in session. *Plenary* means that both natural scientists, including mathematicians, and professors from the humanities attend and present short papers on, or related to, their work. Thus, someone attending regularly, as I do, gets a broad overview of much of contemporary

[5] Rome, N.Y.

scholarship. The Göttingen Academy has become a "second home" to me. (By contrast, many other academies have separate, specialized sessions in the humanities, sciences, etc.)

On one of my visits from Göttingen to New York I got to know Pierre Boulez, then the conductor of the New York Philharmonic. (His musical manner received mixed reviews. Harold Schoenberg (from the *Times*) once characterized his style as "the iceberg conducteth".) Pierre had been asked by the French President Pompidou, who wanted him back in Paris, whether he would like a new concert hall, but Pierre said he preferred an acoustic/music research center. The wish was granted and soon the foundation was laid for the *Institut de Recherche et Coordination Acoustique/Musique* (*IRCAM*) within the *Centre Pompidou* in Paris. During the planning, I traveled with Pierre to several existing acoustic/music research centers: to Peter Zinoviev's[6] laboratory in London and Max Mathews at Bell Labs. We also visited John Chowning at Stanford University as well as Stockholm and Göttingen (digital simulation of concert halls).

Once, during a meeting in Paris, someone pointed out that new laboratories usually run out of space after a few years—we should plan it twice as large. "No problem", said Monsieur Bordaz (Pompidou's representative), "we'll raze another street of the *Quartier Beaubourg* to make room for a bigger laboratory".

The institute's name was concocted during a lunch in the old neighborhood (then still awash with *filles de joie*). The original choice, *Institut de Recherche Acoustique/Musique* was voted down because its acronym would have been pronounced like *Iran*, where Ayatollah Khomeini had just taken over.

During my tenure I also became a "great" cyclist—both in the USA and around Göttingen. In America, Anny and I took many trips with Vermont Bicycle Touring, mostly in New England. In Germany we cycled a lot with friends (Edith and Wolfgang Schröter) and with my students (*Studienstiftler*) (Fig. 22.2).

For indoor fun, I studied (with Gerhard Sessler) the mechanics of the roulette wheel and visited several casinos in the process: Baden-Baden, Baden (near Vienna), Evian-les-Bains. Once we had ascertained that we could win considerable amounts of money (given enough starter capital), we lost interest and dropped the project—in contrast to the authors of the book *The Newtonian Casino* who pursued the idea but were of course barred from the casinos once their method became known. (But, in the meantime, they had lots of fun.)

[6] Peter was one of the pioneers of computer music. When I asked him whether he was related to the (in)famous Old Bolshevik Grigory Zinoviev, he said *No*, he was descended from a rich Russian family whose assets were confiscated after the 1917 revolution and that Grigory, whose birth name was Hirsch Apfelbaum, stole his family's name.—Grigory, who was on that sealed train through Germany with Lenin, to jump-start the revolution, first sided with Leon Trotsky and later allied himself with Stalin to denounce Trotsky (1924). Grigory himself paid the ultimate price when he became the first Old Bolshevik to be executed (August 1936) in Stalin's Great Purges. He was exonerated by Gorbachev in 1988. (For an extensive account of the murderous machinations of Stalin & Co. see Sebag Montefiori: *Stalin*.)



Fig. 22.2 Bicycling with Anny on Nantucket (1981). *Source*: In 'Sconcet (Siasconcet), Nantucket, 1981

My transatlantic commuting necessitated a permanent abode in the USA, so Anny and I decided to buy another house in New Jersey. We could no longer afford the house we had bought in 1968 in Mountainside for $60,000 and sold for $69,000 just 14 months later. Prices had gone through the roof in the meantime. However, we found a nice, smaller house in Berkeley Heights for $62,900. But we had only $700 in cash. So I went to my bank (then The Summit Trust Company) and said: "I need a personal loan for new furniture". How much? I said $20,000 would be fine. Then I went to the mortgage office (next door on the same floor) and said I had found a nice house that I wanted to buy and needed a mortgage loan. "O.k., how much cash do you have?" I said: "20,700 dollars". Fine, so they gave me a mortgage for 43,000 dollars. It was that simple (in the early 1970s) to buy a house in the USA! (Of course, the personal loan was secured by stocks as collateral—as required by law.)

22.4 The Family in Göttingen

By contrast, life in Germany was a lot more difficult. For one, there was no school bus in Nikolausberg, the suburb of Göttingen where we lived. The children had to get up very early to catch the 6:50 city bus to get to school on time. Then, not infrequently, an hour later they would return: classes had been canceled—the teacher was ill. This had never happened to us in the USA—there were always enough substitute teachers around in the New Jersey schools. (One reason may have been that in German, high-school teachers are civil servants—difficult to hire and near-impossible to fire.)

Language—the German language!—was another problem. The teachers paid little attention to the fact that our children had never been to a German-speaking school before. All three needed private tutoring in German at home—Alexander for 2 years—with many a tear shed over the abominable syntax and the orgies of compounding (see my Chap. 23). Now, of course, all our children are perfectly bilingual and, in fact, *tri*lingual because they stayed with French families during their vacations or studied at French universities (Grenoble, Julian).

Another problem was housing: in spite of spending considerable time on house hunting, we found nothing suitable. One house, owned by Percy Ernst Schramm, the keeper of the official war diaries (*Kriegstagebuch*, published in eight volumes) of the Supreme Command of the Armed Forces (*OKW*), was so run down it was disgusting just to visit. Another house, also located in the prestigious *Ostviertel* of Göttingen, was the sumptuous villa of the *Kurator* of the university. But, unfortunately, it was located near a busy intersection with the noise and fumes from accelerating cars and busses.

In our predicament the State of Lower Saxony came to our rescue by finishing two apartments ahead of schedule and combining them into one apartment with a common front door. This gave us 160 m^2, about half of our house in Mountainside, New Jersey. The bigger apartment (90 m^2) became our three children's abode while the smaller one (70 m^2) contained the master bedroom and my study. The second living room became our dining room and the second kitchen served as a nice walk-in closet. Since these apartments were subsidized for middle-level university people, our lease was limited to 4 years—ample time to build a house from scratch.

Our future neighbors (Leo and Clara deMaeyer) found some farm land (20,000 m^2) for sale at 22 marks/m^2 of which they and we took 4,000 m^2 each, the remaining acreage going to the Max Planck Society to build on.

Since we didn't want to live in a stone house (*Stein auf Stein*, huh) we looked at some 20 prefab-housing companies, of which we selected *Bien Haus* near Frankfurt because their workers drank Coca Cola (instead of the ubiquitous beer) during lunch breaks. They were also the lowest priced and the most accommodating to our wishes: we could choose our own design, as long as all measurements were multiples of 62.5 cm. In fact, we did not make use of an architect: the children and Anny designed the L-shaped layout based on our double apartment. Once the basement was finished, the house was erected in 48 h from the prefab walls that arrived on giant trucks. Curiously, the house didn't look prefab. Only our Göttingen contractor (for the basement) remarked in ill-disguised astonishment "*Herr Professor bauen fertig*?" (Mister Professor is building a *prefab* house?). "*Ja, Frau Fricke, wir bauen fertig*".

Incidentally, in all our 37 years at Göttingen, we never bought a German car. We first drove four cars imported from the USA, including an eight-cylinder Plymouth 1965 *Belvedere* with a 6-L (360 cubic inches) engine with a torque of 450 N m at 1,800 r.p.m. What a car! There was of course nothing comparable available on the German market. The car cost 2,900 dollars in 1965 (about 20,000 in 2007-dollars) and, at 14 L per 100 km (17 miles per gallon), was cheaper to run than a VW at today's gas prices.

Before we dared take the Plymouth to Europe, I asked the dealer whether we should take any spare parts. The dealer said "Take an extra fuel filter—that's all you'll need for the next couple of years". And he was right![7] After we retired the *Belvedere* (after 14 years of flawless service), its superb engine had a second life as a winch motor to pull up gliders for the local flying club to which I had donated it.

In 1983, we switched to Japanese cars—mostly Toyotas but recently some Hondas. If you want to know the reason, look at the repair statistics in *Consumer Reports* (or even the *ADAC*—the German AAA), which show Japanese cars consistently ahead of German makes.—Of course, if I was presented with a BMW as a gift, I would gladly accept it.

22.5 Talks and Trips

While a professor in Germany I continued to give talks—at various congresses and physics colloquia in Europe, the USA, Japan, and China—about applications of number theory, speech compression, hearing, and other topics. Once, at Cornell University in Ithaca, New York, I was greatly honored by the presence in the audience of Hans Bethe, one of the greatest physicists of the 20th century. On another occasion, also at Cornell, the famous former Soviet dissident, Yuri Orlov, approached me after the talk. He was honored, he said, to hear my lecture. No, I countered, *I* am honored by your presence. (Yuri was a cofounder of the Moscow Helsinki Watch Group and a close ally of Andrei Sakharov in their fight against Soviet repression. He spent 9 years in the Gulag, many weeks in solitary confinement. He was set free by Gorbachev in 1986.)

Earlier trips led me to Lisbon, from where I visited the Estoril Casino to study the predictability of the odds in roulette. I even made a little money—without any help from physics.

Four times I traveled to the Soviet Union: in 1959 (with Anny, as early tourists from America[8]), in 1963 (at the invitation of the Soviet government—see my Chap. 21), in 1971 to attend the All-Union Conference on Acoustics in Leningrad, and in 1987 to Tallin (the old Hanseatic city of Reval in Estonia) for a conference on phonetics.

One of my most momentous forays saw me in Argentina at the height of the Falkland crisis with Britain in 1982.[9] I had been invited to give a series of seminars on concert hall acoustics at the La Falda mission in the western mountains.

---

[7] In Orange, during a trip to the south of France, the car stalled in the middle of a busy intersection. What had gone wrong? I figured it must be the fuel filter—and the fuel filter it was.

[8] Apart from sightseeing in Leningrad and Moscow, we vacationed on the Black Sea (Sochi, Sukhumi) and visited Gagra and Lake Ritsa in the Caucasus (where Stalin maintained a summer home).

[9] Here is one of my favorite English headlines from those days (when Britain was ruled by Margret Thatcher): BRITISH LEFT WAFFLES ON FALKLANDS. Hmm, how very considerate!

Back in 1963 we started skiing in Switzerland, taking charter flights from New York to Zürich or Geneva (for $180 round trip). We continued this custom when we moved to Germany in 1969—except that the planes were replaced by cars or trains—until 2001. In those 32 seasons we skied all the major resorts in Switzerland (Zermatt, St. Moritz, Davos/Klosters, Arosa, Grindelwald, Verbier and the rest) and three Austrian places (St. Anton, Kitzbühel, Heiligenblut).

The triennial International Congress on Acoustics saw me in Delft, Holland (1953), Cambridge (1956), Stuttgart (1959), Copenhagen (1962), Liège (1965), Tokyo (1968), Budapest (1971), London (1974), Madrid (1977), Sidney (1980), Paris (1983), Toronto/Vancouver (1986), Beijing (1992),[10] Trondheim (1995), and Rome (2001).

It was in Tokyo in 1968, on August 28 (the same day that Warsaw Pact forces invaded Czechoslovakia to subdue Dubchek's "Prague Spring"), that I presented a joint paper with Bishnu Atal on "adaptive" predictive coding, which later morphed into Code-Excited Linear Prediction (CELP), which is at the basis of all speech compression for cell phones and the Internet.

In collaboration with Joseph L. Hall, Bishnu and I introduced *perceptual* coding that resulted in the required high quality of synthetic speech. (Yes, when you listen to a friend on a cell phone, it's not his or her original voice but a synthetic sound manufactured in real-time—with a negligible delay—inside your phone.) The reason cell phones sound so good (if connected to a high-quality loudspeaker) is that CELP is akin to "voice excitation", a principle I invented in 1957 and that, according to John Pierce, led to the first human-sounding speaking machine (see Chap. 20).

22.6 Export–Import

When we first moved to Germany, we missed many things we took for granted in the States. Of course, we couldn't do much about the (nonexistent) school bus service in Nikolausberg. (Our children had to catch a 6:50 city bus to make it to school on time!) But some things could be rectified by the "frequent flyer" in the family. Marion missed the right flour (*Bisquick*) for her pizza dough. And all of us missed the maple syrup for our waffles. Also the local lawnmowers were inadequate for our large lawn (4,000 m^2).

I once imported a Sears lawnmower from the USA and, when German customs wanted to know what I was going to use it for, I said, "Mow my lawn." Then they asked "And afterwards?" "You mean, after I mowed my lawn?" "Yes, yes." Their English was none too good and I sensed that they wanted to get off the hook, get rid of me. So I said, "After having mowed my lawn, I will take the mower back with me

[10] With side trips to Xian—the terra cotta army—and on the Li river with its sugar-cone mountains.

to the States." "Well, then it's o.k." (I can still hear them snickering behind my back about the crazy American who brings his own mower from the States.)

Another occasion, when my not speaking any German was helpful, happened when I was importing live lobsters in a cooler filled with ice cubes (replenished by a helpful flight attendant). Pointing at the cooler, the customs official said "*Was ist das*?" Of course, my not knowing German, he had to repeat in English: "Vot is in zär?" I answered truthfully "Maple syrup and lobster." Whereupon he starts poking around in the ice and first finds a bottle of maple syrup. "Vot is zis?" "Maple syrup." Sure enough, after some more poking he sees the back of a lobster. I see the front end of the animal looking at me through an ice cube (the kind with a hole in the middle). "Vot is zis?" Knowing that he wouldn't understand the English word (lobster is *Hummer* in German) I said "Oh, that's a lobster." Extended silence. Then the kind but overtaxed official said "*Ja, bitte, gehen Sie weiter*." Which, of course, I didn't understand. So he repeated "O.k., please go on."

Once, during a talk in London, on the mechanics of the inner ear, I wanted to show a film clip showing the (simulated) vibrations of the basilar membrane in the *cochlea*. But the projectionist said "The film cannot be shown". Why, this incompetent projectionist, I thought. But arriving at home in Göttingen, I discovered that some of my "smuggled" maple syrup had leaked into the film can. (What a mess it would have made of the projector! And how polite and with British understatement the projectionist had responded to the situation.)

Chapter 23
The German Language

Manfred R. Schroeder

Mark Twain has of course skewered the German language to his heart's delight (see *A Tramp Abroad*). I will comment on the language as a native speaker who is not bothered by the fact that girls in Germany are neutered (*das* Mädchen) or that a woman is not neutered but becomes *masculine* in (of all cases) the genitive form (*der* Frau). However, when you meet two or more girls they regain their femininity (*die* Mädchen). So when you encounter *Mädchen*, pay close attention to the article (no pun). Depending on *der*, *die* or *das*, *dem*, *des* or *den* you would know their number (hmm) and whether they are doing something or something is being done to them. Some language! (In Dutch, by the way, a girl is also neutered (*het* Meisje) but at least she keeps her other feminine assets intact (*haar* moeder—her mother; *haar* haar—her hair, etc.).

Why, by the way, is it *der* Euro but *die* Mark? (Of course, *der* Euro is worth about *two* Marks.)

I was once asked in chemistry class in middle school (in Rotenburg/Wümme) to elaborate on last week's topic (dimethyl-ether). So I started out "der Methyläther ist. . .." No, that was wrong, the teacher said. The boys behind my back were whispering "di, di, di!" So I tried again "*die* Methyläther. . ." Wrong again! So, as a last resort, I ventured "*das* Methyläther". Well, wrong again! So, in desperation, I said "*Der die das* Methyläther." By this time the teacher was quite angry and he called on another boy who calmly proclaimed "*der Dimethyläther*." I always knew that chemistry harbored dangerous substances but I never realized that the innocent-looking particle *di* was one of them.

But the article *die* can also be exploited to great advantage. Göttingen used to be known as "*Stadt der Wissenschaft*" (City of Science). But then someone must have remembered that Wissenschaft is actually female (*die* Wissenschaft) as it is in

Author was deceased at the time of publication.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_23,
© Springer International Publishing Switzerland 2015

Latin, Italian, etc. Furthermore, by separating Wissenschaft into its two components: Wissen (knowledge) and schaft (ship), and adding a comma and an *f*, he arrived at the new slogan for our city: "*Stadt, die Wissen schafft*" (City that creates science). An ingenious pun!

The *der-die-das* business has also led to some hilarious Pennsylvania Dutch expressions like "Throw the cow over the fence some hay." Poor cow! In the German original there is of course no doubt as to what is being thrown: "Wirf *der* Kuh etwas Heu über den Zaun."

The Dutch, incidentally, did away with the distinction between the grammatical male and female a long time ago. It's always *de* instead of *der* or *die* in German. And in the genitive case they abolished the awkward *der* (*der Frau*!). Instead they write *van de* (except for the Queen, in whose "honor" they keep the *der*: *der Koningin*.

The Dutch have also repeatedly simplified their orthography, making it almost completely phonetic. Compare this to the horrors of English spelling where writing and spelling are largely divorced from each other. (A German friend of mine (Inge Spelsberg), who worked as a secretary for the U.S. Army in the Nuremberg Palace of Justice, became the supreme arbiter of spelling for her Major (in English?) and many minor officials.)

Of course, for some English words spelling and sound are actually the same: orts (leftovers).

Apart from the gender mix-ups, I am also bothered by the relentless compounding of words into veritable tapeworms of letters. A few examples may suffice: The German *Stadtabgeordnetenversammlung* is nothing but a city councilors meeting. And *Vergangenheitsbewältigung*, as Günther Grass knows best, is coming to terms with one's own past. *Waffenstillstandsverhandlungen* are simply armistice negotiations. Even a native German is often at a loss how to segment a word monster like *Seeschiffahrtsassistenzunternehmen* (tug boat assistance enterprise) to extract meaning. This of course slows down his reading and comprehension.

And some words can only be segmented from the context. Is *neuromanisch* some neuro-maniac? No, I encountered the term in an article on architecture so it probably means "New Romanesque".

What on earth is the poor reader to make of *Waldohreulen*. Using within-word capitals would surely help: *WaldOhrEulen*, i.e. we are talking about "wood ear owls" (not barn owls or owls without ears, I suppose).

In a recent article on *Stammzellenforschung* (stem cell research) I encountered a Greek philosopher unknown to me: *Ethikrates*. Was he a follower of *Sokrates*? No! After some mental gymnastics I discovered that *Ethikrates* meant "ethics council's". If it had been spelled *EthikRates* there would have been no problem. Within-word capitalization has of course been used to great advantage in computer languages, advertising, product names, and even personal names: DeGeneres, MacDonald, McGurk, O'Neal, VanDoren, etc.

On the other hand, compounding in German entails a great advantage. It allows the construction of words, even neologisms, that take entire sentences in English and other languages that don't allow compounding. How best to translate

Neiddebatte (debate engendered by jealousy); *Prozessökonomie* (cessation of judicial proceedings)? *Vertragsverletzungsverfahren* are one (or more) judicial proceedings to investigate the alleged violation of a treaty—one word in German, a longish phrase in English.

But the worst sin, as everyone knows, is the German word order. Even having grown up with German, I often have to read a sentence several times before I get its sense (if there be any). As a result, I do most of my reading in English, including German authors. (I make a few exceptions for Thomas Mann and Grass and selected others who really sound best in the German original.) It was only in the USA, at Bell Labs, composing scientific papers, that I learned to express myself understandably in English—which I hope carried over to my German writing style. As Georg Christoph Lichtenberg said in one of his more charming aphorisms: "I really went to England to learn how to write German". (He met George III, who spoke no German, several times. George I, hailing from Hanover, was of course fluent in German and George II, patron of Handel and founder of Göttingen, Princeton, Columbia and Rutgers universities, could still handle it.)

As the head of a large university institute, a lot of German missives crossed my desk, but I let my secretaries tell me what was in it. (Mostly nothing I needed to know.) One of my worst correspondents was the university president, whose long sentences may have been difficult to decode even for himself. I fondly remember one instance: the *Ehrenpromotion* (awarding of a *Dr. honoris causa*) to mathematician Fritz Hirzebruch in the festive *Aula* (ceremonial hall). The president read the *laudatio* from a sheaf of typed pages. At one point, after having turned over another page, he discovers *toward the middle of the next page* that the sentence doesn't make any sense—the proper verb was still missing or something. Then he found out that he had accidentally skipped a page! Three cheers for German syntax!

As if German words weren't long enough already, the native speaker likes to festoon them with extra garlands. Thus, coffee isn't *Kaffee* but often *Bohnenkaffee* (because during the war, coffee was made mostly from barley). And honey isn't simply *Honig* but *Bienenhonig* because the German chemical industry had contrived an ersatz product called *Kunsthonig* (which we as children—to my mother's despair—actually preferred over the real thing).

Apple juice isn't just *Apfelsaft* but, on one of our containers, *Apfelfruchtsaftgetränk* (apple fruit juice beverage) and this, I am assured, for good legal reasons. Just give me a glass of apple juice, please, and forget about apples that are not fruit and juice that isn't a beverage, or whatever other possibilities exist. Of course milk isn't just *Milch* but always *Voll*milch. To top it all, I recently discovered a label on a box that said it contained *Alpenvollmilchschokolade* while the English part of the label simply said *milk chocolate*. The poor monolingual tongue, alas, never realizes that her milk chocolate was made from the whole(some) milk of mountain-climbing cows.

On the other hand, there are a few admirable and short German word creations: e.g. *Twens* (for the awkward twenty-somethings) and *Handy* (for cell phone).

Another German weakness is to pack several, often disparate, ideas into the same sentence. I leave the patient reader with just one example in the original:

Könnte man die Artus- und Gralsdichtung als Versuch werten, die als chaotisch erfahrene Realität durch einen ‚utopischen' Gegenentwurf ethisch und ästhetisch zu bewältigen, so schienen andere zeitgenössische Werke dieses optimistische Modell, das auf Ausgleich der Gegensätze zwischen geistlicher und weltlicher Macht sowie zwischen individuellen und gesellschaftlichen Ansprüchen abzielte, unterlaufen zu wollen. Get it?

Twain also makes fun of the myriad meanings of the German *Zug* for which he lists no fewer than 29 different meanings. This reminded me of a 30th meaning: During World War II a secret telegram from Field Marshal Rommel's *Afrikakorps* was decoded (in part) as "*sie kommen in einem Zug*". One of the meanings of *Zug* is train. But there are no trains in the desert. So what did *Zug* mean in this instance? The decoding station luckily had a native speaker of German who said *Zug* in this case meant *platoon*. In other words the enemy would arrive in platoon strength. (Similar problems arose in the decoding of Japanese military messages. Just cracking the code was not enough: you needed a native speaker *and* someone familiar with Japanese military usage.)

23.1 Military Terms

One aspect of German usage that I do not particularly like is the abundance of military terms—even after two catastrophic wars. Thus, my chief secretary, was often standing *Gewehr bei Fuß* (rifle at the ready) when she wanted to signal that she was ready to take dictation. *Schützenhilfe* (help; literally: cover given by a rifleman) is another favorite, as is *Schulterschluß* (presenting a united front). Other examples are *austreten* (step out of line to pee), *Pkw* (from *Personenkraftwagen*; literally person power wagon, i.e. car), *Lkw* (from *Lastkraftwagen*; i.e. truck) and, the really horrible but thankfully disappearing, *Krad* (pronounced crud), military abbreviation for *Kraftrad* (literally: power wheel, i.e. motor bike).

My friend and comrade-in-words Eike Christian Hirsch has published a collection of military expressions still in wide use in his hilarious books. Here is a sampler:

generalstabsmässig, geballte Ladung, volle Deckung, bombensicher, Bombenstimmung, auf Vordermann bringen, Tuchfühlung, Rückendeckung und Flankenschutz, Durchbruch, Brückenkopf, Volltreffer. Etc.

Hirsch is also famous for such hilarious German word plays as "*DIE HAUT DER ARME ENTZWEI*", meaning either "the skin of one's arms…" or "he hit the poor one to pieces".

Why are even modern Germans wedded to the military terms of bygone days? Does it reflect a persisting affinity to *Disziplin und Ordnung* (discipline and order)? In a similar vein, American English abounds with words from the business world. I just read that an Olympian, who had won a second gold medal, had "doubled her holdings in gold". Who was it again, who said "America's business is business!" Charlie Wilson, the head of GM? No, he became famous for his "What's good for

General Motors is good for America." In a recent editorial in the New York Times (18 August 2006) Paul Krugman of Princeton, in a critique of conservative tax policies, coined the paraphrase "what's good for the rich is good for America".

But back to German. How can a "refined" lady, wife of a professor no less (albeit of chemistry), stoop to such abominable words as *Latrinenparolen* (rumors exchanged while sitting and doing one's business on a latrine)? I am always amazed how little attention many people pay to the origin of the words they use. Of course, in Germany, *derbe Sprache* (coarse language) is considered an asset by many native speakers; it's considered a sign of honesty. (Compare this with the prudishness of English as when one suburban housewife complains about her neighbor: "Her dog always goes to the bathroom on my front lawn.")

A pinnacle of prudishness is reached when, in the men's locker room of our local (New Jersey) swim club, men shower with their swim trunks left on. This reminds me of my mother, who was educated in a catholic convent, where students were strictly forbidden to see their own bodies. Once a week, they took a cleaning shower dressed in a one-piece bathing suit, applying the soap to the fabric rather than the naked flesh.

I was a little astonished when, not too long ago, I saw big posters all over Switzerland ironically proclaiming (in an anti-AIDS campaign) "*Ich schlafe nur mit gesunden Menschen—ich brauche keine Condoms*." ("I sleep only with healthy people—I need no condoms."

The current German public pro-condom campaign is even more pregnant: It shows a large condom (in its virgin state) with the exhortation "*Mach's mit*" (literally: make it with, i.e. do it with condom). This is a fine pun on the frequent German plea "*Mach mit*" (come on, join in). These ads are probably more effective in preventing AIDS than calls for abstinence and "virginity pledges".

Prudishness also manifests itself when a toddler loses his or her diapers on the beach and people get very upset about such a "wardrobe malfunction", to use a current term. The original "wardrobe malfunction," displayed by Janet Jackson during intermission time of a football game at the 2004 Superbowl, of course wouldn't raise the tiniest eyebrows outside the USA, where dogs actually pee on front lawns (if not held back by a ubiquitous fence).

Another example that comes to mind is when, at the funeral of French President Mitterrand, his widow invited Mitterrand's paramour and their illegitimate child to walk behind the coffin.

23.2 Borrowed Words

Today, many sports terms in the UK and the USA, especially soccer, are German: *Eigentor* (own goal), *Abseits* (off side) and many others.

On the other hand, most languages, including German, have borrowed heavily from English (weekend, computer...). But not a few English words come from German: *Kindergarten*, *Blitzkrieg*, *Weltschmerz, Katzenjammer*, *Zeitgeist*,

Gemütlichkeit, *Wunderkind*, *Ersatz*, *Flak*, *Angst* and *Schadenfreude* come to mind. In fact, *über* (over) now appears in compounds such as *Übermensch* and überlobbyist. More German words entered English via Yiddish: *spiel*, *shtick*, *schmuck* and *schmo*, to name a few.

Some German words sound as if they had been taken from English but are actually German creations: *Handy* = cell phone; *Twen* = twenty-something person (a young man or woman between the ages of 20 and 29 years); *Smoking* = black-tie suit; *Beamer* = projector for digital images.

The language with perhaps the most words borrowed from German is Russian. Here are some of my favorites: *galstukh* (necktie), *bukhhalter* (bookkeeper), *shlagbaum* (road barrier), *butterbrot* (sandwich), *straf* (penalty), *strafbatalyon* (penal battalion), *feldweibel* (sergeant), *gefreiter* (corporal), *front* (army group), *pochtamt* (post office).

Bismarck, on his way to St. Petersburg as the new Prussian Minister, complained about the driver of his sled who always called the rider in front "*Verräter*" (traitor). Well, the Russian word for advance rider is the German *Vorreiter* but pronounced much like the German word for traitor (from W. Lehfeldt, Göttingen).

23.3 Gruntled

I once saw a cartoon in the *New Yorker* magazine showing two return windows in a department store, one window labeled "disgruntled" and the other one labeled "gruntled". What does "gruntled" mean? This reminded me of the many expressions in English (and German) in which only the negative form exists. *Ungeheuer* means monster. But there is no noun I know *Geheuer*.

Words like *Ungeheuer* are called "orphan negatives", of which English has quite a few other than disgruntled: inept and insipid come to mind. But actually, inept and insipid do have positive forms, albeit spelled differently: apt and sapid (having taste or intriguing, stimulating the mind) from the Latin *sapidus* (tasty).

Talking about negatives, I can't help thinking of the German prefix *un*, which can have the same meaning as the English *un* or *in*: *un*ending, *in*capable. But the German *un* has still another function besides negation, namely emphasis. Thus, *Untiefe* does not mean "undepth" (i.e. "shallow", as the innocent bather might discover to his chagrin) but great depth. (But, paradoxically, for a ship's captain looking at his charts, *Untiefe* does mean: "Attention! Shallow waters.")

Unkosten doesn't mean it's gratis, but that it costs extra. And *Unwetter* is not "unweather" but a heavy thunderstorm. An *Unfall* is an accident, not a "no-fall". I leave the reader with decoding *Ungezieferbekämpfungsmittel*, a typical German construction.

Other non-sequiturs of German are *bestimmt* (certainly) which often means "I am not sure". Or take *ganz* (wholly) which sometimes connotes "tolerable". Or look at *langsam* (slowly) which can mean "immediately" as in the phrase *Ich gehe langsam ins Bett*, i.e. I am going to bed right away (from L. Crawfords, Göttingen).

One idiosyncrasy of German, which makes it unnecessarily difficult for foreigners to comprehend, is its infatuation with the Latin neuter ending *um*, as in *Klinikum* (clinics), *Juridicum* (law school) and, perhaps worst (but perfectly legal), *Zentrum* (center). What is the lost foreign motorist, desperately seeking city center, to make of a road sign saying *Zentrum*? What kind of a tantrum is this, especially since around the world *Zen* is associated with Zen-Buddhism. So is *Zentrum* the place where Buddhists perform tantrums? Or what?

I was once berated by my colleagues in the Göttingen Academy that I had construed *Agenda* as a feminine singular in the title of a talk ("*Chaos: eine neue Agenda der Physik*"). I should have known they told me, that *Agenda* was the plural of *Agendum*, which is a neuter noun in Latin. Yes, I know. But what my friends, apparently disconnected from linguistic reality, didn't know was that the whole world, including Germany, has been using *agenda* as a singular for a long time.—The worst part of the episode was that one academician tried to console me that I didn't have to be ashamed for my linguistic lapse. Thank you, Professor W.

23.4 Du Versus Sie

Another difficulty of German (and other languages) is the strict adherence to the distinction between the familiar *Du* (thou) and the more formal *Sie* (you). Many German scientists call each other Helmut, Kurt, Karin, Erhard, Ingrid, etc. at international meetings where the *lingua franca* is English but revert to the formal *Herr Kollege*, *Frau Kollegin* when they speak German. A suitable compromise is to stay on a first-name basis in German but combine it with the formal *Sie*-form of address: Karin haben *Sie* (never Karin hast *Du*, which would carry a strong undertone of familiarity, if not intimacy).

23.5 English Versus German

I prefer English when it comes to reading (because of the simple syntax I read much faster in English than in German). However, the identity of nouns and verbs in English sometimes slows comprehension. Does "reports" mean (the) reports or (he) reports? Same with many other English noun/verb pairs. This is particularly problematic with newspaper headlines. What is meant by the headline "Ethics rules lag states." Are *rules* and *states* nouns or verbs? I had to read half the article to find out. (The proposed ethics rules in Congress lacked enough votes by the individual states to be adopted.)

During the Falkland war (before Margaret Thatcher put her foot down) the following headline in a British paper caught my attention: BRITISH LEFT WAFFLES ON FALKLANDS—hmm, how very considerate! (But my English friends never had any difficulty with this pronouncement.)

I prefer to speak German—after all my native language. And I can hear (understand) German speech better than English. In other words speaking and listening to speech were learned and programmed in the brain at a very early age. For the same reason, accents absorbed as a young child (before the age of 10 or 12) stay with us for the rest of our lives. (Audrey Hepburn, who grew up in Holland with Dutch, speaks English without much of her native accent—a rare exception to the general rule. Of course, she underwent extensive language coaching, as did Archibald Leach of Bristol while turning into Cary Grant in Hollywood.)

When it comes to writing, I do about equally well in either language—with a slight edge for English (again because of the simpler syntax, I guess).

Chapter 24
Retirement and Beyond

Manfred R. Schroeder

On September 30, 1987, I retired from Bell Laboratories 33 years after having been put on the Murray Hill payroll on September 30, 1954—the day I arrived in the USA on the *Andrea Doria*.

While my double appointment had thereby come to an end, my life as a professor of physics at the University of Göttingen continued unabated until September 30, 1991, when I became an emeritus.[1] Although *entpflichtet* (released) from many pleasant duties (lectures, chair General Physics Kolloquium, shepherding students from the *Studienstiftung*) and not-so-pleasant chores (committees, administration), my intellectual activities survived unshorn. I still offered (and offer) seminars on some of my favorite topics (speech, applications of number theory, fractals & chaos, and, lately, quantum cryptography) albeit to much smaller audiences (typically 5–15 students instead of 60–100 *Hörer* during my prime). In addition, I give the occasional talk on a variety of subjects (Bell Laboratories, Werner Heisenberg, Lise Meitner, Concert Hall Acoustics, Computer Graphics, Mathematics, and Music, The Development and Future of Physics). At the same time my attachment to the Göttingen Academy of Sciences, my second intellectual home, has grown stronger.

I also had more time to attend Manfred Eigen's *Winterseminare* in Klosters, Switzerland, where I met many famous scientists: John Wheeler, Lars Onsager, Murray Gell-Mann, Stephen Weinberg from the USA; Christopher Longuet-Higgins, Roger Penrose from the UK, and numerous Nobel laureates and near-Nobelists from around the world.

Author was deceased at the time of publication.

[1] I don't know why the date September 30 keeps cropping up in my life.—It was also the date I was kicked out of the *Luftwaffe* in 1944 (for having missed boot camp) and shoved off to the *Kriegsmarine*.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_24,
© Springer International Publishing Switzerland 2015

The *Winterseminar* also allowed Anny and me to continue our "tradition" of annual skiing vacations in Switzerland, beginning in 1963 in Zermatt (three times, the first time I broke my shoulder on the downhill run to Cervinia in the Val d'Aosta, Italy[2]) and continuing in Arosa, Grindelwald (from where we took the train to the *Jungfraujoch* to ski into a distant valley along the *Aletschgletscher*—source of both the Rhine and the Rhone rivers).

Other slopes we hit included Davos, St. Moritz, Diavolazza; Seiser Alm, Italy, and Lac de Tigne (Club Med) and *Mer de Glace* (above Chamonix, France, but reached from Entrèves in Italy), and Verbier (twice; once we took a helicopter to a high glacier (Glacier du Trient) near the French border. Another time we made a very strenuous hiking tour on skis among the surrounding mountains (Rosa Blanche). And once we witnessed a large avalanche plow into the *beginner's* slope. Miraculously, although several skiers, including children, were buried under snow, they were all dug out—nobody was lost.).

We also skied in Crans-Montana, Saas-Fee and finally in Klosters with its incomparable—and incomparably long—downhill runs.

Before Switzerland we skied in New England: Stowe, Franconia Notch (Cannon Mountain), Killington (at −40 °F (−40 °C), when the snow feels like *sand*), Canada (Mont Tremblant), New York (Belleayre and Lake Placid), Squaw Valley, California, and, for the faint of heart, *New Jersey* (McAffee—only an hour's drive from New York City).

With all this skiing, from Switzerland to California, we only hit the slopes twice in Austria: St. Anton am Arlberg (including Lech and Zürs) and Kitzbühel (where I broke my leg in 1969).—Austria somehow never seemed as attractive to us as Switzerland.

We also skied on Camel Back Mountain, Pennsylvania; Snowbird and Park City, Utah; Aspen, Vail and Copper Mountain (another Club Med) in Colorado; Jackson Hole, Wyoming (at a freezing −37 °C, corresponding to −34 °F).

I also skied several times at Alta, Utah, with its superb dry powder snow and only a $7 taxi ride (in the 1960s) from the Salt Lake City airport, where I often stopped over enroute from New York to California. (The only thing I didn't like about Alta was that it was also dry in the *liquor* sense: you had to bring your own bottle of Scotch or you were out of luck—as I was during my first visit.)

With Ed David and Ann, often accompanied by their dog Gretchen, in their Studebaker Starliner we picked up Anny near the George Washington Bridge in New York on Fridays at 4 p.m. Around midnight, we reached our destination in Vermont or New Hampshire, ready to ski the next day and a half. (Ed always made sure that Anny and I occupied separate bedrooms—this was in the "chaste old days" of 1955.) There were no Interstate Highways then and once we reached New York

[2] My Zermatt doctor was a descendent of the physician who accompanied the first, ill-fated, ascent to the *Matterhorn* on July 14, 1865. (On the way down, four of the climbers fell to their deaths.) He was also responsible for keeping the outbreak of typhoid fever in 1963 under wraps.—My broken shoulder he diagnosed as "twisted" until an X-ray in the USA revealed the true state of affairs.

as late as Monday morning at 8. The next 4 days we relaxed at work for another weekend on the slopes. (How we ever got any work done is beyond me!)

Nevertheless, although not remotely of Nobel calibre, I was honored with a number of Gold Medals: The Audio Engineering Society in New York (already in 1972), the Rayleigh Gold Medal of the British Institute of Acoustics in 1987, the Helmholtz-Medal of the German Acoustical Society in 1995 and the Gold Medal of the Acoustical Society of America in 1991 ("For theoretical and practical contributions to human communication through innovative application of mathematics to speech, hearing and concert hall acoustics"). Whew!

In 1993 I received the Lower Saxony State Prize "for outstanding scientific accomplishments" (signed by then Premier Gerhard Schröder[3]).

In 2004, I was awarded the Eduard Rhein Preis (for my "scientific life work in room acoustics, psycho-acoustics, speech coding and computer graphics, and especially for the invention of linear predictive coding, the most important foundation of speech coding and speech analysis." Good citation!).

Also in 2004 the International Speech Communication Society, in a meeting on Jeju Island, Korea, gave me their Medal "For significant scientific achievement in the field of Speech Communication Science." (While in Korea, we "inspected" one of the secret tunnels the North Koreans had dug under the Demilitarized Zone.)

Apart from these "metals," I received the greatest satisfaction from the enthusiastic reception of my books. (One Kristina Lerman even recommended taking my book on Number Theory to a coffee shop to keep her company.[4]) The enthusiastic review by Philip Morrison in *Scientific American* kept me awake all night—until I could call Springer the next morning to print an extra 2,000 copies.

When I returned to Göttingen (the book was written in Murray Hill) I was amazed to learn that there was no street named of Hermann Minkowski to whom I had dedicated the book. With considerable patience (and the help of *Oberbürgermeister* Rinck) this was eventually rectified: there is now a *Minkowskiweg* (near the Frank-Born-Ring in the north of the city). At the same time the city created an Emmy-Noether-Weg, a Hermann-Weyl-Stieg, a Maria-Göppert-Weg.

---

[3] In his presentation speech the future Chancellor of Germany mentioned that he had heard that I liked good wines. He therefore thought it fitting to nominate me for membership in the *Toskana Fraktion* (Tuscany Faction). Not following German internal politics, I didn't know what he meant. Friends had to tell me that *Toskana Fraktion* designated a (fictitious) club of well-to-do *SPD* (social democratic party) bigwigs that maintained private villas in Tuscany and generally preferred the "good" life. Thank you, Gerhard!

[4] On her webpage *All the Books I have Known* she wrote: I was never terribly interested in number theory, but I regret it now. It is a fascinating subject, and Schroeder's book is a wonderful place to start. He is highly readable, if prone to corny jokes, but the best thing is his ability to show how obscure statements of mathematics can have far-reaching effects, e.g., how fundamental theorem of arithmetic leads to the well-tempered scale, and other wonderments. His other book *Chaos, Fractals and Power Laws* is a similarly engaging look at the wide ranging manifestations of chaos, and a good non-technical, though intellectually demanding introduction to the field.



Fig. 24.1 Nikki (2) on a visit to Nikolausberg

When the city asked me about Fermi, I said, of course, Göttingen should have a street named after Fermi—post-doc in Göttingen, Nobel prize winner and creator of the first "atomic pile" (now called nuclear reactor) in December 1942—something Heisenberg and his collaborators never managed before the end of the war. "Aber der Kerl hat sich nicht abgemeldet!" (But the chap never deregistered with the city!) I persuaded the city to name a street after Fermi anyhow: And now Göttingen has an *Enrico-Fermi-Eck* (Fermi corner)—not quite a street.

Later the city called about Johannes Starck. I asked: "Where did you get that name?" "Well, he was a student in Göttingen and he won a Nobel prize in physics." "And which other sources did you consult?" "The *Brockhaus Enzyklopädie*." I said: "Go to the Encyclopedia Brittanica, where it is clearly stated that Starck was the top Nazi among German physicists."

When I told Friedrich Hund this story, he stopped in mid-track, exclaiming: "Ich ziehe weg von Göttingen!" (I will move away.) I said: "No need to move away, Herr Hund." I persuaded the city *not* to name a street after Starck after all—it would have been a disaster.



Fig. 24.2 Lilly and Julia (both 7) in Bremen

In July 2001 I suffered a minor stroke at our home in Berkeley Heights, New Jersey. Luckily, our daughter Marion, a neurologist, was staying with us at the time. She could provide a quick diagnosis and arrange a speedy admission to Emergency at Overlook Hospital in Summit. This may have saved me from more serious consequences. In any case, I was discharged after a mere 9 days (I had asked Dr. Michael Suhl, who became a good friend, whether my discharge was an "honorable discharge," which he confirmed). Thus I could join family and friends for my 75th birthday celebration in Manasquan on the Jersey shore. Dr. Suhl told me that I could no longer ski which I felt was an exaggerated precaution. But in actual fact I could no longer *schuss* because I lacked the necessary balance on my feet—the only persistent residue from my mishap. In the meantime life that doesn't require physical balancing acts continues. I am 80 now but still "going strong."

The time after my second retirement (as a professor, in 1991) was further brightened by the arrival of four granddaughters: Julia in La Jolla and Lilly in Bremen, both in early 1994. Next came Nora, also in Bremen, in 1997, and then Nikki in La Jolla in 1999 (Figs. 24.1, 24.2, and 24.3).

In 2003 all 12 of us got together for Anny's 75th birthday (and my 77th) in Riva del Sole near Castiglione della Pescáia on the coast of Tuscany. A memorable vacation with breakfast on a sun drenched patio near the sea—with another cup of cappuccino and the *Herald Tribune* at my side. (I always started the day with the cartoons and the anagram.) And at the end of the day, dinner *al fresco*, lasting from dusk into the night, under the beautiful umbrella pines so typical of Italy.—I rarely felt so European.

The children availed themselves of the opportunity for side trips to Lucca, Pisa, Elba (Napoleon's first exile), Florence, and Rome. In the evening there was dancing, even for the little ones.



Fig. 24.3 Nora (8) at Schloss Hardenberg and in Göttingen (the young Lauren Bacall?)

Insanely, one of the things I liked best about the whole trip was the 17-h train ride from Germany—or rather the night-train part across the Alps from Munich to Florence: I rarely slept so well—I just love trains. (The children from Germany drove and the Californians of course flew in.)

I am closing this memoir with a letter to friends touching on some of Anny's and my activities in 2006.

Chapter 25
The Year 2006: One of Our Best (So Far)

Manfred R. Schroeder

February 25 was the 50th anniversary of our wedding in the Chapel of the Riverside Church in Manhattan. We commemorated the day—just the two of us—at the Metropolitan Opera with a performance of *Samson and Delilah* (Fig. 25.1).

On May 1 my sister Helga celebrated her 70th birthday with a day of hiking, singing, and dancing at a country resort in the *Fränkische Schweiz* north of Nuremberg.

Later in the year, my New York nephew and godson David Harkness got engaged to Leslie Johnson of Kansas—the extended family is growing, still (Fig. 25.2).

Another memorable moment was my 80th birthday. During the weekend before the great day, our children, Marion from Bremen, Julian from San Diego, Alexander from San Francisco, and my sisters Ingrid from Hamm and Helga from Erlangen—together with their husbands, children, and grandchildren—arrived in Göttingen. The weather was beautiful if a bit too warm (as it had been all summer) so our first get-togethers took place on the terrace and lawn of our house in Nikolausberg. We particularly enjoyed seeing our four granddaughters, Lilly and Nora from Bremen and Julia and Nikki (Nicola Alexandra Jazzmin!) from California. And of course they had great fun seeing each other again—great pals that they are.

For the Sunday preceding the birthday we had invited 74 of our best friends, including my oldest friend, Hermann Precking (from a time when we were both 3 years old), to lunch in the Schloßhotel Hardenberg near Göttingen. We had counted on perhaps 50 guests but actually all but two (felled by the abominable heat that engulfed most of Europe) came. Our children and my sisters entertained us with amusing reminiscences. Professor Thomssen told a story involving the great

Author was deceased at the time of publication.

M.R. Schroeder (✉)
Drittes Physikalisches Institut, Georg-August-University Göttingen, Friedrich Hund-Platz 1, 37077 Göttingen, Germany

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9_25,
© Springer International Publishing Switzerland 2015



Fig. 25.1 At the Metropolitan Opera, New York, February 25, 2006

French mathematician Pierre de Fermat and Herbert Roesky, the President of the Göttingen Academy, gave a heart-warming speech about my close attachment to the Academy—My actual birthday, on July 12, was a comparatively modest affair on our terrace. But I was especially pleased by the presence of our *Ortsbürgermeisterin* Dr. Gabriele Funck who, in an earlier incarnation, had been our family physician.

Two weeks later we celebrated Anny's birthday with a most enjoyable breakfast attended by many of her friends at the patisserie *Cron & Lanz* (one of Europe's best, in our opinion).—Two days later we were off to New Jersey and our second home. Of course it was just as hot on the East Coast as in Europe, but we didn't mind because our house there is fully air-conditioned. Twice every day we went swimming, savoring a seemingly endless summer with occasional forays to the Big Apple.



Fig. 25.2 Leslie Johnson in Port Washington, New York, Easter 2007

25.1 Hawaii

On November 11 we left NJ to visit Alexander and our friends in San Francisco, where we celebrated the 80th birthday of Max Mathews with a brunch at the Sheraton Palace (where sailor Max and his future wife Marjorie had their first date some 58 years earlier). This was another chapter of our long friendship with the Mathews. Among our joint ventures were many sailing trips—from New York harbor up the East Coast to Maine, in the Caribbean and, most notably, in the Aegean among the Cycladic Islands. We visited Nauplion (Agamemnon's house) and anchored at Delos, Naxos, and Cap Sunion. I even took out a *Small Craft* license from the U.S. Coast Guard just in case the main skipper, Max, was incapacitated—which thankfully never happened.[1]

[1] In Delos (birthplace of Apollo), the aromatic smell of fresh potato pancakes wafted across the bay, emanating from a boat that flew the German flag. Starved for fresh food (after 2 weeks of canned stuff), I shouted across the bay "Haben Sie frische Kartoffeln?" "Nein." Then where does the wonderful smell come from? Answer: "Wir haben Kartoffelpuffer-Mix!" "Okay, could you spare some?" After asking permission to board our boat, which was granted, they brought a pound or two of the potato pancake mix—delicious! In Cap Sunion, one of us (Guy Mathews) had to dive into the bay to get enough cash to pay the restaurant. At another Island, I swam ashore with my clothes in a dry bucket. But when I asked (by sign language) for a private place to change, they sent me to the public area where half the village youths congregated to watch the proceedings. Since we had opted to sail without a dinghy we sometimes had to walk fully clothed into the water to reach our boat and onlookers were probably suspecting suicides.

After a delightful week in the Bay Area, we decamped for Honolulu and a cruise among the Hawaiian Islands. We had never wanted to do a cruise but a good friend, Thomas Rossing, persuaded us to join them on this—in retrospect—magnificent adventure. Our ship, the PRIDE OF HAWAII of the Norwegian Cruise Lines, was sailing under the US flag because its itinerary was completely in American waters. It was built in Germany, served by an American crew and owned by a Malaysian company—talk about globalization! (Not to mention the French, Italian, Chinese, and Japanese eateries aboard.)

After the cruise we participated in a Joint Meeting of the Acoustical Societies of America and Japan in Honolulu. The outstanding event of the meeting for us was a special session on my life in acoustics with speakers from Germany, the USA, Japan, France, and the Netherlands. I was particularly pleased to hear how some of the concepts that I had proposed a long time ago and named after me (Schroeder frequency, Schroeder integration, Schroeder phases, Schroeder diffusors) are still in current use.[2]

Also, during the meeting in Hawaii, the American Institute of Physics conducted a lengthy interview with me (conducted by Gerhard Sessler at the *Sheraton Waikiki* poolside) for their Oral History Projects. I was happy to learn that the AIP has added several photos of me and our family to their *Emilio Segrè Visual Archives*.

After the meeting (and last swim in the Pacific on December 1) we flew back to Germany via Tokyo which means you fly only during daytime without losing any sleep. (While flying over northern Siberia you see the sun, after having first set over China, rise again in the *West*.)

In Tokyo we stayed again at our favorite hotel, the International House of Japan. As on many previous occasions, Hiroya Fujisaki acted as our gracious host.

After more than a year of trying in vain, we were finally able to connect reliably to the broadband Internet so Anny could play Bridge online. I indulged two of *my* hobbies by giving a seminar at the university on *Applications Of Number Theory* and a talk in Jena on *The Early History of Computer Graphics at Bell Laboratories*.

[2] My friend Manfred Eigen (from my first years in Göttingen in the 1940s), who won the Nobel Prize in chemistry (actually *physical* chemistry) in 1968, once remarked to me about the several things that bear my name and that only *one* concept (from mathematical physics) was named after him: *Eigenfunktion*. This was a nice joke because eigenfunctions have nothing to do with Manfred Eigen; they are named after the German *Eigenfunktion* (proper function).

Bibliography of Manfred R. Schroeder's Publications

1. Schroeder, M.R.: Number Theory in Science and Communication: With Applications in Cryptography, Physics, Biology, Digital Information, and Computing. Springer, New York (1983)
2. Schroeder, M.R.: Computer Speech Recognition, Compression, Synthesis. Springer, Berlin (1985)
3. Schroeder, M.: Hundert Jahre Friedrich Hund: Ein Rückblick auf das Wirken eines bedeutenden Physikers. Vandenhoeck und Ruprecht, Göttingen (1996)
4. Schroeder, M.R.: Die statistischen Parameter der Frequenzkurven von großen Räumen. Acustica **4**, 594–600 (1954)
5. Schroeder, M.R.: Eigenfrequenzstatistik und Anregungsstatistik in Räumen. Modellversuche mit elektrischen Wellen. Acustica **4**, 456–468 (1954)
6. Schroeder, M.R.: Messung von Dielektrizitätskonstanten durch Aufspaltung entarteter Eigenfrequenzen. Z. Angew. Physik **7**(4), 169–174 (1955)
7. Schroeder, M.R.: On the separation and measurement of formant frequencies. J. Acoust. Soc. Am. **28**(1), 159 (A) I4 (1956)
8. Schroeder, M.R.: On the steady-state transmission in rooms. J. Acoust. Soc. Am. **29**(11), 1258(A) I5 (1957)
9. Schroeder, M.R.: New interpretation of the stereophonic effect in Lauridsen's and similar binaural experiments. J. Acoust. Soc. Am. **29**(6), 774 (A) K4 (1957)
10. Schroeder, M.R.: Spatial fluctuations of steady-state sound fields in rooms. J. Acoust. Soc. Am. **30**(7), 679 (A) I2 (1958)
11. Schroeder, M.R.: An artificial stereophonic effect obtained from a single audio signal. J. Audio Engrg. Soc. **6**(2), 74–79 (1958)
12. Schroeder, M.R.: Vocoders for military use. In: Proc. Seminar on Speech (1959)
13. Schroeder, M.R.: Methoden zur Messung der Diffusität in Hallräumen. Acustica **9**, 256–264 (1959)
14. Schroeder, M.R., Logan, B.F.: The sound of rain. Frequenz **13**, 229–234 (1959)
15. Schroeder, M.R.: New results concerning monaural phase sensitivity. J. Acoust. Soc. Am. **31** (11), 1579 (A) J5 (1959)
16. Schroeder, M.R.: Measurement of sound diffusion in reverberation chambers. J. Acoust. Soc. Am. **31**(11), 1407–1414 (1959)
17. Schroeder, M.R.: New approach to time domain analysis and synthesis of speech. J. Acoust. Soc. Am. **31**(5), 852 (A) R2 (1959)
18. Schroeder, M.R.: Improvement of acoustic feedback stability in public address systems. J. Acoust. Soc. Am. **31**(5), 851 (A) Q7 (1959)
19. Schroeder, M.R., David Jr., E.E.: Vocoder and its application to the transmission of high-quality speech over narrow-band channels. J. Acoust. Soc. Am. **31**(1), 113 (A) D5 (1959)

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9,

© Springer International Publishing Switzerland 2015

20. Schroeder, M.R.: Measurement of reverberation time by counting phase coincidences. In: 3rd International Congress on Acoustics (ICA 3), Stuttgart, vol. 2, pp. 897–901 (1959)
21. Schroeder, M.R.: Improvement of acoustic feedback stability in public address systems. In: 3rd International Congress on Acoustics (ICA 3), Stuttgart, vol. 2, pp. 771–775 (1959)
22. Schroeder, M.R.: Recent progress in speech coding at bell telephone laboratories. In: 3rd International Congress on Acoustics (ICA 3), Stuttgart, vol 1, pp. 202–210 (1959)
23. Harvey, F.K., Schroeder, M.R.: Subjective evaluation of factors affecting two-channel stereophony. J. Acoust. Soc. Am. **32**(11), 1521 (A) S8 (1960)
24. Schroeder, M.R., David Jr., E.E.: A vocoder for transmitting 10 kc/s speech over a 3.5 kc/s channel. Acustica **10**(1), 35–43 (1960)
25. Schroeder, M.R., Logan, B.F.: "Colorless" artificial reverberation. J. Acoust. Soc. Am. **32** (11), 1520 (A) S3 (1960)
26. Schroeder, M.R., Crystal, T.H.: Autocorrelation vocoder. J. Acoust. Soc. Am. **32**(7), 913 (A) B2 (1960)
27. Schroeder, M.R.: Stop feedback in public address systems. Radio Electronics **31**, 40–42 (1960)
28. David Jr., E.E., Schroeder, M.R., Logan, B.F., Prestigiacomo, A.J.: New applications of voice-excitation to vocoders. J. Acoust. Soc. Am. **33**(11), 1684 (A) SP6 (1961)
29. Harvey, F.K., Schroeder, M.R.: Subjective evaluation of factors affecting two-channel stereophony. J. Audio Engrg. Soc. **9**(1), 19–28 (1961)
30. Schroeder, M.R.: Vocoders. Encyclopedia of Electronics. Reinhold, New York (1961)
31. Schroeder, M.R.: Novel uses of digital computers in room acoustics. J. Acoust. Soc. Am. **33** (11), 1669 (A) N9 (1961)
32. Schroeder, M.R.: Improved quasi-stereophony and "colorless" artificial reverberation. J. Acoust. Soc. Am. **33**(8), 1061–1064 (1961)
33. Schroeder, M.R.: Further progress with colorless artificial reverberation. J. Acoust. Soc. Am. **33**(6), 852–853 (A) Q2 (1961)
34. Schroeder, M.R., Logan, B.F.: "Colorless" artificial reverberation. J. Audio Engrg. Soc. **9**, 192–197 (1961)
35. Sessler, G., Schroeder, M.R.: Sound velocity in slightly ionized gases. J. Acoust. Soc. Am. **33** (11), 1673 (A) P11 (1961)
36. Atal, B.S., Schroeder, M.R., Kuttruff, K.H.: Perception of coloration in filtered gaussian noise – Short-time spectral analysis by the ear. In: 4th International Congress on Acoustics (ICA 4), Kopenhagen, Vortrag H31 (1962)
37. David, E.E., Jr., Schroeder, M.R., Logan, B.F., Prestigiacomo, A.J.: Voice-excited vocoders for practical speech bandwidth reduction. In: Proceedings of International Symposium on Information Theory, Brüssel (1962)
38. Flanagan, J.L., Schroeder, M.R., Bird, C.M.: Single-channel speech interpolator for 2:1 bandwidth reduction. J. Acoust. Soc. Am. **34**(12), 2003 (A) X7 (1962)
39. Harvey, F.K., Cisek, J.G., Schroeder, M.R.: Initial characteristics of impulse response for real and artificial reverberation. J. Acoust. Soc. Am. **34**(5), 726 (A) K7 (1962)
40. Prestigiacomo, A.J., Schroeder, M.R.: Practical systems for quasi-stereophony. J. Acoust. Soc. Am. **34**(5), 737 (A) T3 (1962)
41. Schroeder, M.R., Logan, B.F.: A solution to problem of compatible single-sideband transmission. In: Proceedings of International Symposium on Information Theory, Brüssel (1962)
42. Schroeder, M.R., Logan, B.F., Prestigiacomo, A.J.: Harmonic compression – A new method for speech bandwidth compression. J. Acoust. Soc. Am. **34**(12), 2003 (A) X5 (1962)
43. Schroeder, M.R.: Frequency-correlation functions of frequency responses in rooms. J. Acoust. Soc. Am. **34**(12), 1819–1823 (1962)
44. Schroeder, M.R., Atal, B.S.: Generalized short-time power spectra and autocorrelation functions. J. Acoust. Soc. Am. **34**(11), 1679–1683 (1962)
45. Schroeder, M.R., Atal, B.S., Kuttruff, K.H.: Perception of coloration in filtered gaussian noise Short-time spectral analysis by the ear. J. Acoust. Soc. Am. **34**(5), 738 (A) U1 (1962)

46. Schroeder, M.R.: Frequency correlation of sound-transmission functions in rooms. J. Acoust. Soc. Am. **34**(5), 736 (A) S2 (1962)
47. Schroeder, M.R., Kuttruff, K.H.: On frequency response curves in rooms. Comparison of experimental, theoretical, and Monte Carlo results for the average frequency spacing between maxima. J. Acoust. Soc. Am. **34**(1), 76–80 (1962)
48. Schroeder, M.R.: Natural sounding artificial reverberation. J. Audio Engrg. Soc. **10**, 219–223 (1962)
49. Schroeder, M.R.: Correlation techniques for speech bandwidth compression. J. Audio Engrg. Soc. **10**(2), 163–166 (1962)
50. Schroeder, M.R.: Improvement of feedback stability of public address systems by frequency shifting. J. Audio Engrg. Soc. **10**(2), 108–109 (1962)
51. Schroeder, M.R., Logan, B.F., Prestigiacomo, A.J.: New Methods for Speech Analysis-Synthesis and Bandwidth Compression. Speech Communication Seminar R.I.T, Stockholm (1962)
52. Schroeder, M.R., Logan, B.F., Prestigiacomo, A.J.: New methods for speech analysis-synthesis and bandwidth compression. In: 4th International Congress on Acoustics (ICA 4), Kopenhagen, Vortrag G41 (1962)
53. Schroeder, M.R., Atal, B.S., Bird, C.M.: Digital computers in room acoustics. In: 4th International Congress on Acoustics (ICA 4), Kopenhagen, Vortrag M21 (1962)
54. Schroeder, M.R., Logan, B.F., Prestigiacomo, A.J.: New applications of voice-excitation to vocoders. In: Speech Communication Seminar R.I.T., Stockholm, vol. 2, Paper G–3 (1962)
55. Sessler, G., Schroeder, M.R.: Interaction of soundwaves with a gas discharge. In: 4th International Congress on Acoustics (ICA 4), Kopenhagen, Vortrag O25 (1962)
56. David Jr., E.E., Schroeder, M.R., Logan, B.F., Prestigiacomo, A.J.: New applications of voice-excitation to vocoders. J. Acoust. Soc. Am. **35**(7), 1116 (A) (1963)
57. Harvey, F.K., Cisek, J.G., MacLean, D.J., Schroeder, M.R.: Some aspects of stereophony applicable to conference use. J. Audio Eng. Soc. 11, 212–217 (1963)
58. Schroeder, M.R., Atal, B.S.: Computer simulation of sound transmission in rooms. IEEE Int. Conv. Rec. **11**(7), 150–155 (1963)
59. Schroeder, M.R., Logan, B.F., Prestigiacomo, A.J.: New methods for speech analysis-synthesis and bandwidth compression. J. Acoust. Soc. Am. **35**(7), 1116 (A) (1963)
60. Schroeder, M.R.: Bookreview: Ungeheuer, G. (Ed.), Elemente einer akustischen Theorie der Vokalartikulation (Elements of an Acoustic Theory of Vowel Articulation). J. Acoust. Soc. Am. **35**(5), 743 (1963)
61. Atal, B.S., Schroeder, M.R.: Subjective responses of nonexponential reverberation and delayed low frequencies in concert halls. J. Acoust. Soc. Am. **36**(5), 1011 (A) C8 (1964)
62. Noll, A.M., Schroeder, M.R.: Short-time 'Cepstrum' pitch detection. J. Acoust. Soc. Am. **36** (5), 1030 (A) N5 (1964)
63. Schroeder, M.R.: Improvement of acoustic-feedback stability by frequency shifting. J. Acoust. Soc. Am. **36**(9), 1718–1724 (1964)
64. Schroeder, M.R.: An acustical journey to the soviet union. J. Acoust. Soc. Am. **36**(7), 1403–1406 (AN) (1964)
65. Schroeder, M.R.: Digital-computer method of measuring concert-hall acoustics. J. Acoust. Soc. Am. **36**(5), 1011 (A) C6 (1964)
66. Schroeder, M.R.: Bookreview: Sapozhkov, M.A. (Ed.), Rechevoy Signal v Kibernetike i Svyazi (The Speech Signal in Cybernetics and Communications). J. Acoust. Soc. Am. **36**(1), 225 (1964)
67. Schroeder, M.R., Atal, B.S., Sessler, G., West, J.E.: Acoustical measurements in philharmonic hall (New York). In: Application of Digital Computer Techniques to the Evaluation of the Acoustics of Concert Halls, vol. 3. Akustische Konferenz, Budapest, pp. 380–384 (1964)
68. Schroeder, M.R., Sessler, G., Atal, B.S., West, J.E.: Application of digital computers in the design and evaluation of concert halls. In: Nerem Record, Northeast Electronics Research and Engineering Meeting, M–4.1 (1964)

69. Sessler, G., Atal, B.S., Schroeder, M.R., West, J.E.: Acoustical measurements in philharmonic hall (New York). J. Acoust. Soc. Am. **36**(5), 1011 (A) C7 (1964)
70. Atal, B.S., Schroeder, M.R., Sessler, G., West, J.E.: Reverberation times and early energies in philharmonic hall (New York). J. Acoust. Soc. Am. **37**(6), 1188 (A) J5 (1965)
71. Atal, B.S., Schroeder, M.R., Sessler, G.: Subjective reverberation time and its relation to sound decay. In: 5th International Congress on Acoustics (ICA 5), Lüttich, 1b, Vortrag G32 (1965)
72. Mermelstein, P., Schroeder, M.R.: Determination of smoothed cross-sectional area functions of the vocal tract from formant frequencies. J. Acoust. Soc. Am. **37**(6), 1186 (A) I7 (1965)
73. Mermelstein, P., Schroeder, M.R.: Determination of smoothed cross-sectional area functions of the vocal tract from formant frequencies. In: 5th International Congress on Acoustics (ICA 5), Lüttich, 1a, Vortrag A24 (1965)
74. Schroeder, M.R.: Response to Comments on 'new method of measuring reverberation time'. J. Acoust. Soc. Am. **38**(2), 359–361 (L) (1965)
75. Schroeder, M.R.: Artificial reverberation: review and future outlook. J. Acoust. Soc. Am. **37** (6), 1192 M1 (k. Ex.) (1965)
76. Schroeder, M.R.: New method of measuring reverberation time. J. Acoust. Soc. Am. **37**(6), 1187–1188 (A) J4 (1965)
77. Schroeder, M.R.: New method of measuring reverberation time. J. Acoust. Soc. Am. **37**(3), 409–412 (1965)
78. Schroeder, M.R.: Erwin Meyer (Receives Wallace Clement S. Award). J. Acoust. Soc. Am. **37**(1), 178–180 (AN) (1965)
79. Schroeder, M.R.: Some new results in reverberation theory and measurement methods. In: 5th International Congress on Acoustics (ICA 5), Lüttich, 1b, Vortrag G31 (1965)
80. Schroeder, M.R., Noll, A.M.: Recent studies in speech research at bell telephone laboratories (I). In: 5th International Congress on Acoustics (ICA 5), Lüttich, 1a, Vortrag A21 (1965)
81. Atal, B.S., Schroeder, M.R., Sessler, G., West, J.E.: Evaluation of acoustic properties of enclosures by means of digital computers. J. Acoust. Soc. Am. **40**(2), 428–433 (1966)
82. Schroeder, M.R.: Architectural acoustics. Science **151**(3716), 1355–1359 (1966)
83. Schroeder, M.R.: Simulation with a block diagram compiler. In: Symposium on "The Human Use of Computing Maschines", Bell Telephone Laboratories (1966)
84. Schroeder, M.R., Atal, B.S.: Nachahmung der Raumakustik durch den Elektronenrechner. Gravesaner Bl **27–28**, 124–131 (1966)
85. Schroeder, M.R., Hanauer, S.L.: Nonlinear time compression and time normalization of speech. J. Acoust. Soc. Am. **40**(5), 1243 (A) 1H9 (1966)
86. Schroeder, M.R., Flanagan, J.L., Lundry, E.A.: Speech band compression by analytic-signal rooting. J. Acoust. Soc. Am. **40**(5), 1242–1243 (A) 1H8 (1966)
87. Schroeder, M.R.: Complementarity of sound buildup and decay. J. Acoust. Soc. Am. **40**(3), 549–551 (1966)
88. Schroeder, M.R., Atal, B.S., Sessler, G., West, J.E.: Acoustical measurements in philharmonic hall (New York). J. Acoust. Soc. Am. **40**(2), 434–440 (1966)
89. Schroeder, M.R.: Residue pitch: a remaining paradox and a possible explanation. J. Acoust. Soc. Am. **40**(1), 79–81 (1966)
90. Schroeder, M.R.: Reverberation time: definition, theory, and measurement. J. Acoust. Soc. Am. **39**(6), 1230 (A) 3B2 (1966)
91. Schroeder, M.R.: Articulatory information from acoustic data. J. Acoust. Soc. Am. **39**(6), 1220–1221 (A) 2B1 (1966)
92. Schroeder, M.R.: Architectural acoustics. In: McGraw-Hill Yearbook Science and Technology, pp. 123–124 (1966)
93. Schroeder, M.R.: Vocoders: analysis and synthesis of speech. A review of 30 years of applied speech research. In: Proceedings of IEEE, 54(5), M. 5184 (1966)
94. Atal, B.S., Schroeder, M.R.: Predictive coding of speech signals. In: Proceedings of IEEE Conference on Communication Processing (1967)

95. Atal, B.S., Schroeder, M.R.: Study of sound decay using ray-tracing techniques on a digital computer. J. Acoust. Soc. Am. **41**(6), 1598 (A) E/AT4 (1967)
96. Schroeder, M.R.: Speech – Man's natural communication. IEEE Spectrum, 75–86 (1967)
97. Schroeder, M.R., Hanauer, S.L.: Interpolation of data with continuous speech signals. Bell Sys. Tech. J. **46**(8), 1931–1933 (1967)
98. Schroeder, M.R.: Electronic suppression of reverberation. J. Acoust. Soc. Am. **41**(6), 1597 (A) E/AT2 (1967)
99. Schroeder, M.R.: Determination of the geometry of the human vocal tract by acoustic measurements. J. Acoust. Soc. Am. **41**(4), 1002–1010 (1967)
100. Schroeder, M.R., Flanagan, J.L., Denes, P.B.: Speech research at bell yelephone laboratories. Research Inst. Logopedics & Phoniatrics Universität Tokio: Bell Telephone Laboratories, pp. 11–12 (1967)
101. Schroeder, M.R., Flanagan, J.L., Lundry, E.A.: Comments on "Bandwidth compression of speech by analytic-signal rooting". Proc. IEEE **55**(10), 1770 (1967)
102. Schroeder, M.R., Flanagan, J.L., Lundry, E.A.: Bandwidth compression of speech by analytic-signal rooting. Proc. IEEE **55**(3), 396–401 (1967)
103. Atal, B.S., Schroeder, M.R.: Predictive coding of speech signals. In: Wescon.Tech.Pap. 12, Paper 8/2, 7 (1968)
104. Atal, B.S., Schroeder, M.R.: Predictive coding of speech signals. In: 6th International Congress on Acoustics (ICA 6), Tokio, 13–16, Vortrag C–5–4 (1968)
105. Schroeder, M.R.: Directions for speech research. IEEE Spectrum, 83–88 (1968)
106. Schroeder, M.R.: Period histogram and product spectrum: new methods for fundamental-frequency measurement. IEEE Trans. Audio **AU-16**(3), 477 (A) 200 (1968)
107. Schroeder, M.R.: Reference signal for signal quality studies. J. Acoust. Soc. Am. **44**(6), 1735–1736 (L) (1968)
108. Schroeder, M.R.: Period histogram and product spectrum: new methods for fundamental-frequency measurement. J. Acoust. Soc. Am. **43**(4), 829–834 (1968)
109. Schroeder, M.R.: Similarity measure for automatic speech and speaker recognition. J. Acoust. Soc. Am. **43**(2), 375–377 (L) (1968)
110. Schroeder, M.R.: Computers in acoustics: symbiosis of an old science and a new tool. In: 6th International Congress on Acoustics (ICA 6), Tokio, Vortrag GP–6–1 (1968)
111. Atal, B.S., Schroeder, M.R.: Predictive coding of speech signals. IEEE Trans. Audio **AU-17** (1), 58 (A) 307 (1969)
112. Lummis, R.C., Schroeder, M.R.: Technique for enhancing signals for auditory detection. J. Acoust. Soc. Am. **45**(5), 1290 (L) (1969)
113. Schroeder, M.R.: Images from computers. IEEE Spectrum, 66–78 (1969)
114. Schroeder, M.R.: Images from computers and microfilm plotters. Communications ACM **12** (2), 95–101 (1969)
115. Schroeder, M.R.: Computers in acoustics: symbiosis of an old science and a new tool. IEEE Trans. Audio **AU-17**(4), 309 (A) 527 (1969)
116. Schroeder, M.R.: Reference signal for signal quality studies. IEEE Trans. Audio **AU-17**(2), 178 (A) 379 (1969)
117. Schroeder, M.R.: Relation between critical bands in hearing and the phase characteristics of cubic difference tones. J. Acoust. Soc. Am. **46**(6), 1488–1492 (1969)
118. Schroeder, M.R.: Spatial averaging in a diffuse sound field and the equivalent number of independent measurements. J. Acoust. Soc. Am. **46**(3), 534 (L) (1969)
119. Schroeder, M.R.: Effect of frequency and space averaging on the transmission responses of multimode media. J. Acoust. Soc. Am. **46**(2), 277–283 (1969)
120. Schroeder, M.R.: Computers in acoustics: symbiosis of an old science and a new tool. J. Acoust. Soc. Am. **45**(5), 1077–1088 (1969)
121. Sessler, G., West, J.E., Schroeder, M.R.: Toroidal microphones. IEEE Trans. Audio **AU-18** (1), 69–70 (A) 583 (1969)

122. Sessler, G., West, J.E., Schroeder, M.R.: Toroidal Microphones. J. Acoust. Soc. Am. **46**(1), 28–36 (1969)
123. Atal, B.S., Schroeder, M.R.: Adaptive predictive coding of speech signals. Bell Sys. Tech. J. **49**(8), 1973–1986 (1970)
124. Schroeder, M.R.: Parameter estimation in speech: a lesson in unorthodoxy. IEEE Trans. Audio **AU-18**(4), 505–506 (A) 806 (1970)
125. Schroeder, M.R.: Synthesis of low-peak-factor signals and binary sequences with low autocorrelation. IEEE Transact. Inf. Theory **IT-16**(1), 85–89 (1970)
126. Schroeder, M.R.: Digital simulation of sound transmission in reverberant spaces. J. Acoust. Soc. Am. **47**(2), 424–431 (1970)
127. Schroeder, M.R.: Parameter estimation in speech: a lesson in unorthodoxy. Proc. IEEE **58**(5), 707–712 (1970)
128. Hall, J.L., Schroeder, M.R.: Monaural phase effects for two-tone Signals. J. Acoust. Soc. Am. **51**(6), 1882–1884 (1972)
129. Henze, H., Schroeder, M.R.: Aufsteilung der Wellenfront durch Nichtlinearitäten (Ein elektrischer Modellversuch zur Entstehung von Stoßwellen). Acustica **26**(5), 302–303 (1972)
130. Schroeder, M.R.: Neural models in autditory perception. In: Proceedings of 20th International Congress on Psychology, Tokio, 210 (A) (1972)
131. Schroeder, M.R.: An integrable model for the basilar membrane. In: Symposium on Hearing Theory IPO, Eindhoven, pp. 123–133 (1972)
132. Hall, J.L., Schroeder, M.R.: Computer simulation of a new hair cell model. J. Acoust. Soc. Am. **54**(1), 283 (A) K8 (1973)
133. Schroeder, M.R.: Computer models for concert hall acoustics. Am. J. Phys. **41**(4), 461–471 (1973)
134. Schroeder, M.R., Hall, J.L.: A model for mechanical-to-neutral transduction in the auditory receptor ('Hair Cell'). J. Acoust. Soc. Am. **54**(1), 283 (A) K7 (1973)
135. Schroeder, M.R.: An integrable model for the Basilar membrane. J. Acoust. Soc. Am. **53**, 429–434 (1973)
136. Schroeder, M.R.: Models in hearing. In: Speech and Hearing, Chelsea College, Spring Meeting, London, Session A, Paper 73SHA2 (1973)
137. Atal, B.S., Schroeder, M.R.: Recent advances in predictive coding – Applications to speech synthesis. In: Speech Communication Seminar, SCS-74, Stockholm, vol. 1, pp. 27–31 (1974)
138. Schroeder, M.R.: Speech wave processing – Some personal reminiscences and expectations. In: Speech Communication Seminar, SCS-74, Stockholm (1974)
139. Schroeder, M.R., Hall, J.L.: A model for mechanical to neural transduction in the auditory receptor. Acustica **31**(6), 362 (1974)
140. Schroeder, M.R., Gerlach, R.E.: Diffusion, room shape and absorber location – Influence on reverberation time. J. Acoust. Soc. Am. **56**(4), 1300 (T) (1974)
141. Schroeder, M.R., Gottlob, D., Siebrasse, K.-F.: Comparative study of European concert halls: correlation of subjective preference with geometric and acoustic parameters. J. Acoust. Soc. Am. **56**(4), 1195–1201 (1974)
142. Schroeder, M.R., Hall, J.L.: A model for mechanical to neural transduction in the auditory receptor. J. Acoust. Soc. Am. **55**(5), 1055–1060 (1974)
143. Schroeder, M.R., Gerlach, R.E.: Diffusion, room shape and absorber location – Influence on reverberation time. J. Acoust. Soc. Am. **55**(2), 420 (A) Q7 (1974)
144. Schroeder, M.R.: Recent advances in hearing research nonlinear mechanics and neural transduction. In: 8th International Congress on Acoustics (ICA 8), London, pp. 47–61 (1974)
145. Atal, B.S., Schroeder, M.R.: Linear prediction analysis of speech based on a pole-zero representation. J. Acoust. Soc. Am. **58**(1), S96 (A) VV4 (1975)
146. Atal, B.S., Schroeder, M.R., Stover, V.: Voice-excited predictive coding system for lowbit rate transmission of speech. In: Proceedings of International Congress on Communications, San Francisco, pp. 30–37–30–40 (1975)

147. Calavrytinos, P.G., Schroeder, M.R.: Erkennung ganzer Zahlwörter über die Energieverteilung in der Zeit-Frequenz-Ebene. In: Fortschritte der Akustik – DAGA '75, Braunschweig; Weinheim: Physik, pp. 385–388 (1975)
148. Eysholdt, U., Gottlob, Siebrasse, K.-F., Schroeder, M.R.: Räumlichkeit und Halligkeit – Untersuchung zur Auffindung korrespondierender objektiver Parameter. In: Fortschritte der Akustik – DAGA '75, Braunschweig; Weinheim : Physik, pp. 471–474 (1975)
149. Gottlob, D., Siebrasse, K.-F., Schroeder, M.R.: Neuere Ergebnisse zur Akustik von Konzertsälen. In: Fortschritte der Akustik – DAGA '75, Braunschweig; Weinheim: Physik, pp. 467–470 (1975)
150. Schroeder. M.R.: The auditory pathway. In: Proceedings of International Congress. "From Theoretical Physics to Biology", Versailles (1975)
151. Schroeder, M.R.: Amplitude behavior of the cubic difference tone. J. Acoust. Soc. Am. **58**(3), 728–732 (1975)
152. Schroeder, M.R.: Diffuse sound reflection by maximum-length sequences. J. Acoust. Soc. Am. **57**(1), 149–150 (1975)
153. Schroeder, M.R.: Computers, speech and images. Matematik och datorer i sprak-, tal- och musikforskning TR Nr. 1, Centrum för talkommunikationsforskning och musikakustik, KTH Stockholm, pp. 45–52 (1975)
154. Schroeder, M.R.: New results and ideas for architectural acoustics. In: Auditorium Acoustics. Proceedings of International Symposium on Architecture and Acoustics, Edinburgh, 1974, hrsg. v. R.Mackenzie; London: Appl. Sci. Publ., pp. 197–212 (1975)
155. Schroeder, M.R.: Models of hearing. Proc. IEEE **63**(9), 1332–1350 (1975)
156. Calavrytinos, P.G., Schroeder, M.R.: Sprecherunabh ängige Worterkennung ohne Segmentierung. Acustica **36**(3), 184–191 (1976)
157. Calavrytinos, P.G., Schroeder, M.R.: Sprecherunabhängige Ganzworterkennung ohne Segmentierung. In: Fortschritte der Akustik – DAGA '76, Heidelberg; Düsseldorf: VDI, pp. 649–652 (1976)
158. Gerlach, R.E., Schroeder, M.R.: Die Anwendung von Maximalfolgendiffusoren in einem Modellhallraum. In: Fortschritte der Akustik – DAGA '76, Heidelberg; Düsseldorf: VDI, pp. 255–258 (1976)
159. Hall, J.L., Schroeder, M.R.: Model study of nonlinear response to paired clicks. J. Acoust. Soc. Am. **60**(1), S40 (A) R6 (1976)
160. Möller, W.-J., Atal, B.S., Schroeder, M.R.: Determination of articulatory parameters of the human vocal tract from acoustic measurements. J. Acoust. Soc. Am. 60(1), S77 (A) HH10 (1976)
161. Möller, W.-J., Schroeder, M.R.: Kodierung von Sprache durch einhüllende und gewichtete Augenblicksfrequenz. In: 2. DFG-Kolloquium "Sprachverarbeitung", Michelstadt-V, pp. 7–8 (1976)
162. Schroeder, M.R.: A cocktail-party processor. In: Arden House Conference (1976)
163. Schroeder, M.R., Gerlach, R.E.: Response to "Comments on 'Diffuse sound reflection by maximum-length sequences'". J. Acoust. Soc. Am. **60**(4), 954 (L) (1976)
164. Schroeder, M.R., Möller, W.-J.: Concert hall measurements during musical performances. J. Acoust. Soc. Am. **60**(1), S9 (A) D2 (1976)
165. Schroeder, M.R.: Concert hall measurements during musical performances. Vortragsmanuskript (1976)
166. Schroeder, M.R., Gawron, H.-J., Gottlob, D., Möller, W.-J.: Messung akustischer Parameter mit Musiksignalen in besetzten Konzertsälen. In: Fortschritte der Akustik – DAGA '76, Heidelberg; Düsseldorf: VDI, pp. 237–240 (1976)
167. Steingrube, A., Schroeder, M.R.: Abänderung der Grundperiode stimmhafter Sprachsignale. In: Fortschritte der Akustik – DAGA '76, Heidelberg; Düsseldorf: VDI, pp. 645–648 (1976)
168. Strube, H.W., Schroeder, M.R.: Rauschbefreiung von Sprache bei Übertragung über mehrere unabhängig verrauschte Kanäle. In: Fortschritte der Akustik – DAGA '76, Heidelberg; Düsseldorf: VDI, pp. 665–668 (1976)

169. Wagner, J.R., Schroeder, M.R.: Untersuchung zur automatischen Sprechererkennung mit Ganzworten. In: Fortschritte der Akustik – DAGA '76, Heidelberg; Düsseldorf: VDI, pp. 653–656 (1976)
170. Calavrytinos, P.G., Wagner, J.R., Schroeder, M.R.: Automatic word and speaker recognition based on time-frequency domain processing. In: 9th International Congress on Acoustics (ICA 9), Madrid, 485, Vortrag I79 (1977)
171. Calavrytinos, P.G., Wagner, J.R., Schroeder, M.R.: Automatische Wort- und Sprecherkennung mit gleichen Parametern und Erkennungsmethoden. In: 3. DFG-Kolloquium "Sprachverarbeitung", Muggendorf, pp. 51–52 (1977)
172. Gottlob, D., Gawron, H.-J., Schroeder, M.R.: Measuring acoustical parameters of concert halls with music. In: 9th International Congress on Acoustics (ICA 9), Madrid, 3, Vortrag A1 (1977)
173. Jacobs, U., Schneider, W., Schroeder, M.R., Strube, H.W.: Vorarbeiten zur Sprachsynthese aus interpolierten Lautübergängen. In: 3. DFG-Kolloquium "Sprachverarbeitung", Muggendorf, pp. 63–64 (1977)
174. Schroeder, M.R., Gottlob, D.: New measurements for architectural acoustics. In: Proceedings of Seminar Measuremant Methods in Architectural Acoustics, Lund, Schweden (1977)
175. Schroeder, M.R.: New Viewpoints in Binaural Interaction. Academic Press, New York, pp. 455–467 (1977)
176. Schroeder, M.R.: Bookreview: Jassem, W. (Ed.), Speech analysis and synthesis. IEEE Trans. Acoust. Speech Signal Process ASSP **25**(3), 268–269 (B) (1977)
177. Schroeder, M.R.: Bookreview: Cremer, L. (Ed.), Die wissenschaftlichen Grundlagen der Raumakustik. Band II: Wellentheoretische Raumakustik. J. Acoust. Soc. Am. **62**(6), 1524–1525 (B) (1977)
178. Schroeder, M.R.: One binaural unmasking mystery unmasked. In: 9th International Congress on Acoustics (ICA 9), Madrid, 365, Vortrag H41 (1977)
179. Schroeder, M.R., Gerlach, R.E.: Diffuse sound reflection surfaces. In: 9th International Congress on Acoustics (ICA 9), Madrid, 138, Vortrag D8 (1977)
180. Schroeder, M.R.: In: Recognition of Complex Acoustic Signals, Dahlem Konferenzen, hrsg. v. T.H. Bullock: Life Sci.Res.Rep vol. 5, pp. 323–328 (1977)
181. Schroeder, M.R.: Machine processing of acoustic signals: what machines can do better then organisms (and Vice Versa). In: Recognition of Complex Acoustic Signals. Dahlem Konferenzen, hrsg. v. T.H. Bullock: Life Sci.Res. Rep, vol. 5, pp. 183–207 (1977)
182. Sondhi, M.M., Schroeder, M.R.: Acoustic camera using inverse diffraction. J. Acoust. Soc. Am. **62**(1), S22 (A) I7 (1977)
183. Steingrube, A., Schroeder, M.R.: Modification of pitch and duration of speech signals. In: 9th International Congress on Acoustics (ICA 9), Madrid, 521, Vortrag I115 (1977)
184. Strube, H.W., Schroeder, M.R.: Extraction of a speech signal from a noisy environment (Cocktail-Party Effect). In: 9th International Congress on Acoustics (ICA 9), Madrid, 435, Vortrag I29 (1977)
185. Atal, B.S., Schroeder, M.R.: Linear prediction analysis of speech based on a pole-zero representation. J. Acoust. Soc. Am. **64**(5), 1310–1318 (1978)
186. Atal, B.S., Schroeder, M.R.: Predictive coding of speech signals and subjective error criteria. In: IEEE ICASSP 78, Tulsa, vol. 18.2, pp. 573–576 (1978)
187. Calavrytinos, P.G., Wagner, J.R., Schroeder, M.R.: Kombinierte Wort- und Sprechererkennung. In: Fortschritte der Akustik – DAGA '78, Bochum, 13.-16. März; Berlin: VDE, pp. 493–496 (1978)
188. Gottlob, D., Kohlrausch, A., Schroeder, M.R.: Raumakustische Untersuchungen mit Hilfe der digitalen Schallfeldsynthese. In: Fortschritte der Akustik – DAGA '78, Bochum, 13.-16. März; Berlin: VDE, pp. 575–578 (1978)
189. Gottlob, D., Kohlrausch, A., Schroeder, M.R.: Einfluß raumakustischer Parameter auf die subjektive Beurteilung von Schallfeldern. In: Fortschritte der Akustik – DAGA '78, Bochum. Berlin: VDE-Verlag, 575–578 (1978)

190. Guicking, D., Lutze, R., Schroeder, M.R.: Messungen zur Diffusität im Halltank. In: Fortschritte der Akustik – DAGA '78, Bochum, 13.-16. März; Berlin: VDE, pp. 309–312 (1978)
191. Lewien, T., Calavrytinos, P.G., Schroeder, M.R., Strube, H.W.: Worterkennung mit Hilfe von Singulärwertzerlegung des Wortspektrogramms. In: Fortschritte der Akustik – DAGA '78, Bochum, 13.-16. März; Berlin: VDE, pp. 489–492 (1978)
192. Mehrgardt, S., Schroeder, M.R.: Die Wahrnehmbarkeitsschwelle von Unterschieden beim Paarvergleich komplexer Signale. In: Fortschritte der Akustik – DAGA '78, Bochum, 13.-16. März; Berlin: VDE, pp. 515–518 (1978)
193. Schroeder, M.R.: Modulation transfer functions: definition and measurement. IEEE Trans. Acoust. Speech Signal Process ASSP **26**(2), 180 (A) (1978)
194. Schroeder, M.R., Atal, B.S., Hall, J.L.: Optimizing digital speech coders by exploiting masking properties of the human ear. J. Acoust. Soc. Am. **64**(1), S139 (A) YY5 (1978)
195. Schroeder, M.R., Gottlob, D., Eysholdt, U.: Binaural dissimilarity: its importance and realization in room-acoustical design. J. Acoust. Soc. Am. **64**(1), S120 (A) TT1 (1978)
196. Schroeder, M.R., Gerlach, R.E., Gottlob, D.: Wall and ceiling structures for efficient sound diffusion over wide frequency bands. J. Acoust. Soc. Am. **64**(1), S120 (A) TT2 (1978)
197. Schroeder, M.R.: New results in auditorium acoustics at Göttingen. J. Acoust. Soc. Am. **64** (1), S35 (A) M1 (1978)
198. Strube, H.W., Schroeder, M.R.: A Cocktail-Party processor: noise reduction for multichannel speech signals. IEEE Trans. Acoust. Speech Signal Processing ASSP **26**(2), 180 (A) (1978)
199. Strube, H.W., Jacobs, U., Schneider, W., Schroeder, M.R.: Zur Sprachsynthese aus interpolierten Lautübergängen. In: Fortschritte der Akustik – DAGA '78, Bochum, 13.-16. März; Berlin: VDE, pp. 477–480 (1978)
200. Atal, B.S., Schroeder, M.R.: Predictive coding of speech signals and subjective error criteria. IEEE Trans. Acoust. Speech Signal Processing ASSP **27**(3), 247–254 (1979)
201. Atal, B.S., Schroeder, M.R.: Optimizing predictive coders for minimum audible noise. In: IEEE ICASSP 79, Washington, 13.2, pp. 453–455 (1979)
202. Calavrytinos, P.G., Schroeder, M.R.: Untersuchungen zur subjektiven Wahrnehmung verrauschter Signale. In: 4. DFG-Kolloquium "Sprachverarbeitung", Darmstadt, pp. 18–21 (1979)
203. Flanagan, J.L., Schroeder, M.R., Atal, B.S., Crochiere, R.E., Jayant, N.S., Tribolet, J.M.: Speech coding. IEEE Trans. Commun. **27**(4), 710–737 (1979)
204. Lewien, T., Schroeder, M.R., Strube, H.W., Calavrytinos, P.G.: Ganzworterkennung mit Hilfe einer Singulärwertzerlegung von Frequenz-Zeit-Spektrogrammen. In: 4. DFG-Kolloquium "Sprachverarbeitung", Darmstadt, pp. 121–124 (1979)
205. Schroeder, M.R., Atal, B.S., Hall, J.L.: Objective measure of certain speech signal degradations based on masking properties of human auditory perception. For Fant, G. on his 60th Birthday (1979)
206. Schroeder, M.R., Atal, B.S., Hall, J.L.: Objective measure of certain speech signal degradations based on masking properties of human auditory perception. In: Frontiers of Speech Communication Research, hrsg. v. B. Lindblom, S. Öhman: Academic Press, New York, pp. 217–229 (1979)
207. Schroeder, M.R., Atal, B.S., Hall, J.L.: Optimizing digital speech coders by exploiting masking properties of the human ear. J. Acoust. Soc. Am. **66**(6), 1647–1652 (1979)
208. Schroeder, M.R.: Integrated-impulse method measuring sound decay without using impulses. J. Acoust. Soc. Am. **66**(2), 497–500 (1979)
209. Schroeder, M.R., Gerlach, R.E., Steingrube, A., Strube, H.W.: Response to "On the theory of optimal plane diffusors" by Berkhout, A.J. et al. J. Acoust. Soc. Am. **65**(5), 1336–1337 (L) (1979)
210. Schroeder, M.R.: Binaural dissimilarity and optimum ceilings for concert halls: more lateral sound diffusion. J. Acoust. Soc. Am. **65**(4), 958–963 (1979)

211. Schroeder, M.R.: Bookreview: Cremer, L., Müller, H.A. (Eds.), Die wissenschaftlichen Grundlagen der Raumakustik. Band I. J. Acoust. Soc. Am. **65**(1), 280–281 (B) (1979)
212. Schroeder, M.R.: Progress in architectural acoustics. J. Acoust. Soc. Japan **1**(1), 71–77 (1979)
213. Schroeder, M.R., Strube, H.W.: Acoustic measurements of articulator motions. Phonetica **36**, 302–313 (1979)
214. Schroeder, M.R.: What acoustic stimulation can teach us about physiological mechanisms underlying speech perception, Repr. Experimental Brain Research, Suppl. II: Hearing Mechanisms and Speech; Springer, Berlin, pp. 247–253 (1979)
215. Schroeder, M.R.: Music perception in concert halls. In: Seminar Okt. 1978, Publ. Royal Swedish Academy of Music, Stockholm, vol. 26 (1979)
216. Atal, B.S., Schroeder, M.R.: Improved quantizer for adaptive predictive coding of speech signals at low bit rates. In: IEEE ICASSP 80, Denver, vol. 2, pp. 535–538, S7.8 (1980)
217. Atal, B.S., Schroeder, M.R.: Adaptive predictive coding of speech signals at low bit rates. In: 10th International Congress on Acoustics (ICA 10), Sydney, Vortrag A1–10.4 (1980)
218. Calavrytinos, P.G., Schroeder, M.R.: Über die Mithörschwelle von Schmalbandrauschen, verdeckt durch Sinustöne. In: Fortschritte der Akustik – DAGA '80, München, 11.-13. März; Berlin: VDE, pp. 643–646 (1980)
219. Calavrytinos, P.G., Schroeder, M.R.: Reduction of quantizing noise loudness. In: Signal Processing 1: Theories and Applications, EURASIP 80, pp. 641–646 (1980)
220. Calavrytinos, P.G., Lewien, T., Helling, D., Strube, H.W., Schroeder, M.R.: Two systems for automatic word and speaker recognition. In: 10th International Congress on Acoustics (ICA 10), Sydney, Vortrag A2–1,4 (1980)
221. Calavrytinos, P.G., Schroeder, M.R.: Subjective perception of noisy signals. In: 10th International Congress on Acoustics (ICA 10), Sydney, Vortrag A1–15,2 (1980)
222. Lewien, T., Strube, H.W., Calavrytinos, P.G., Schroeder, M.R.: Ganzworterkennung mit Hilfe einer Singulärwertzerlegung von Frequenz-Zeit-Spektrogrammen. In: Fortschritte der Akustik – DAGA '80, München, 11.-13. März; Berlin: VDE, pp. 735–738 (1980)
223. Schroeder, M.R., Hall, J.L.: Loudness of noise in the presence of tones: measurements and nonlinear model results. In: 5th International Symposium on Hearing, Noordwijkerhout (1980)
224. Schroeder, M.R., Hackman, D.: Iterative calculation of reverberation time. Acustica **45**(4), 269–273 (1980)
225. Schroeder, M.R.: Constant-amplitude antenna arrays with beam patterns whose lobes have equal magnitudes, (For Severin, H. on his 60th Birthday). AEÜ, Electronics and Communication, vol. 34, pp. 165–168 (1980)
226. Schroeder, M.R.: Acoustics in human communications: room acoustics, music, and speech. J. Acoust. Soc. Am. **68**(1), 22–28 (1980)
227. Schroeder, M.R.: Hearing mechanisms and speech. Experimental brain research supplementum. J. Acoust. Soc. Am. **46**(2), 233 (1980)
228. Schroeder, M.R.: Bookreview: Almeida, A. (Ed.), The physics of speech. Phonetica **37**, 204 (1980)
229. Schroeder, M.R.: Bookreview: Fry, D.B. (Ed.), Nasalitätsdetektion und Vokalerkennung. Phonetica **37**, 196 (1980)
230. Schroeder, M.R.: Toward better acoustics for concert halls. Phys. Today **33**(10), 24–31 (1980)
231. Schroeder, M.R.: Modulation transfer functions: definition and measurement. Acustica **49**(3), 179–182 (1981)
232. Schroeder, M.R.: Architektur und Akustik: Erst der Raum macht die Musik. Bild der Wissenschaft **9**, 106–114 (1981)
233. Schroeder, M.R.: Direct (Nonrecursive) relations between cepstrum and predictor coefficients. IEEE Trans. Acoust. Speech Signal Processing ASSP **29**(2), 297–301 (1981)
234. Schroeder, M.R.: Number theory in physics, engineering and art. Interdisciplinary Sci. Rev. **6** (3), 239–248 (1981)

235. Schroeder, M.R.: Gute Akustik – aber wie? (Good Acoustics – but how?). J. Acoust. Soc. Am. **69**(5), 1531 (1981)
236. Schroeder, M.R., Dudley, H.W.: A tribute. J. Acoust. Soc. Am. **69**(4), 1222 (N) (1981)
237. Schroeder, M.R., Atal, B.S.: Rate distortion theory and predictive coding. In: IEEE ICASSP 81, Atlanta, vol. 1, pp. 201–204 (1981)
238. Lewien, T., Strube, H.W., Schroeder, M.R.: Improvement of speech intelligibility for persons with sensorineural hearing impairment. In: Fortschritte der Akustik – FASE/DAGA '82, Göttingen, 13.-17. September; Bad Honnef: DPG-GmbH, pp. 1055–1058 (1982)
239. Rollwage, M., Ebeling, K.-J., Guicking, D., Schroeder, M.R.: Statistische Parameter zweidimensionaler monofrequenter Schallfelder. In: Fortschritte der Akustik – FASE/DAGA '82, Göttingen, 13.-17. September; Bad Honnef: DPG-GmbH, pp. 211–214 (1982)
240. Schroeder, M.R.: Bookreview: gauss: a biographical study. Interdisciplinary Sci. Rev. **7**(4), 329–330 (1982)
241. Schroeder, M.R., Mehrgardt, S., Strube, H.W.: Monaural phase effects: relase from masking. J. Acoust. Soc. Am. **72**, S90 (A) (1982)
242. Schroeder, M.R.: Concert hall acoustics and number theory? Acoustic precision measurement with music? J. Acoust. Soc. Am. **72**, S18 (A) (1982)
243. Schroeder, M.R.: Linear prediction, extremal entropy and prior information in speech signal analysis and synthesis. Speech Commun. **1**, 9–20 (1982)
244. Schroeder, M.R., Mehrgardt, S.: Auditory masking phenomena in the perception of speech. Stockholm, Mai (1982)
245. Schroeder, M.R.: A simple function and its fourier transform. The Mathematical Intelligencer **4**(3), 158–161 (1982)
246. Schroeder, M.R., Mehrgardt, S.: Auditory masking phenomena in the perception of speech. In: The Representation of Speech in the Perpheral Auditory System, hrsg. v. R. Carlson, B. Grandström; Elsevier Biomedical Press, Amsterdam, pp. 79–87 (1982)
247. Schroeder, M.R., Atal, B.S.: Speech coding using efficient block codes. In: IEEE ICASSP 82, Paris, vol. 3, pp. 1668–1671 (1982)
248. Alrutz, H., Schroeder, M.R.: A fast hadamard transform method for the evaluation of measurements using pseudorandom test signals. In: 11th International Congress on Acoustics (ICA 11), Paris, vol. 6, pp. 235–238 (1983)
249. Atal, B.S., Schroeder, M.R.: Predictive speech signal coding with reduced noise effects. J. Acoust. Soc. Am. **73**(2), 717–718 (P) (1983)
250. Mehrgardt, S., Schroeder, M.R.: Monaural phase effects in masking with multicomponent signals. In: Hearing – Physiological Bases and Psychophysics. Symp. Bad Nauheim, Apr. 1983, hrsg. v. R. Klinke, R. Hartmann; Springer, Berlin, pp. 289–295 (1983)
251. Polack, J.D., Alrutz, H., Schroeder, M.R.: Acoustique. – Une nouvelle methode pour decrire la reverberation. C.R. Acad. Sc. Paris 297, Serie II, 21–24 (1983)
252. Polack, J.D., Alrutz, H., Schroeder, M.R.: Measuring the reverberation time with music. In: 11th International Congress on Acoustics (ICA 11), Paris, vol. 6, pp. 389–392 (1983)
253. Schroeder, M.R.: Fritz E. Borgnis 1906–1982. J. Acoust. Soc. Am. **74**(1), 371–372 (N) (1983)
254. Schroeder, M.R.: Fritz E. Borgnis. Obituary: Physics Today **36**(5), 82 (1983)
255. Schroeder, M.R.: Where is the next mersenne prime hiding? The Mathematical Intelligencer **5**(3), 31–33 (1983)
256. Schroeder, M.R.: The Eikonal equation. The Mathematical Intelligencer **5**(1), 36–37 (1983)
257. Schroeder, M.R.: Speech and hearing: some important interactions. In: Proceedings of 10th International Congress on Phonetic Science, Utrecht, pp. 41–52 (1983)
258. Atal, B.S., Schroeder, M.R.: Stochastic coding of speech signals at very low bit rates. In: Links for the Future, Proceedings of International Conference on Communications ICC '84, Amsterdam, Sci. Syst. & Serv. Communications. IEEE: Elsevier Science Publishers B.V, pp. 1610–1613 (1984)
259. Polack, J.D., Alrutz, H., Schroeder, M.R.: The modulation transfer function of music signals and its applications to reverberation measurement. Acustica **54**(5), 257–265 (1984)

260. Schroeder, M.R.: Number theory in science and communication (Einige Anwendungen der Zahlentheorie). In: German-Japanese Seminar on Nonlinear Problems in Dynamical Systems – Theory and Applications. Universität Stuttgart, Inst. für Mechanik, hrsg. v. M. Hiller, H. Sorg (1984)
261. Schroeder, M.R.: Number Theory in Science and Communication – With Applications in Cryptography, Physics, Biology, Digital Information, and Computing. Springer, Berlin (1984)
262. Schroeder, M.R.: Linear prediction, entropy and signal Analysis. IEEE ASSP Mag **1**(3), 3–11 (1984)
263. Schroeder, M.R.: Number theory in music, speech, and acoustics. J. Acoust. Soc. Am. **76**(1), S58 (A) AA6 (1984)
264. Schroeder, M.R.: Progress in architectural acoustics and artificial reverberation: concert hall acoustics and number theory. J. Audio Engrg. Soc. **32**(4), 194–203 (1984)
265. Schroeder, M.R., Guicking, D., Rollwage, M.: Electronic reverberation influence by controllable wall impedance. Final Report. Pp. 88 TIB/A86-80904/GAR(No. 11,1987) Dissertation Universität Göttingen (k. Ex.) (1984)
266. Schroeder, M.R.: Speech technology for all seasons. Speech Technology (1984)
267. Smith, B.K., Sieben, U.K., Mehrgardt, S., Schroeder, M.R.: Maskierung durch strukturiertes Rauschen. In: Fortschritte der Akustik – DAGA '84, Darmstadt, 26.-29. März; Bad Honnef: DPG-GmbH, pp. 761–764 (1984)
268. Schroeder, M.R.: Linear predictive coding of speech: review and current directions. IEEE Commun Mag **23**(8), 54–61 (1985)
269. Schroeder, M.R.: The unreasonable effectiveness of number theory in acoustics. J. Acoust. Soc. Am. **78**(1), S59 (A) DD1 (1985)
270. Schroeder, M.R.: Some work on hearing at Göttingen. J. Acoust. Soc. Am. **77**(1), S62 (A) CC1 (1985)
271. Schroeder, M.R., Atal, B.S.: Stochastic coding of speech signals at very low bit rates: the importance of speech perception. Speech Commun. **4**(1–3), 155–162 (1985)
272. Schroeder, M.R.: Number theory and the real world. The Mathematical Intelligencer **7**(4), 18–26 (1985)
273. Schroeder, M.R.: Speech and speaker recognition. Basel: Karger, Bibliotheca Phonetica, vol. 12 (1985)
274. Schroeder, M.R., Atal, B.S.: Code-excited linear prediction (CELP): high-quality speech at very low bit rates. In: IEEE ICASSP 85, Tampa, vol. 3, pp. 937–940, 25.1.1.–4 (1985)
275. Strube, H.W., Helling, D., Krause, A., Schroeder, M.R.: Word and speaker recognition based on entire words without framewise analysis. Speech and speaker recognition, hrsg. v. M.R. Schroeder; Basel: Karger, Bibliotheca Phonetica, vol. 12 (1985)
276. Schroeder, M.R.: Number Theory in Science and Communication – With Applications in Cryptography, Physics, Digital Information, Computing and Self-Similarity. Springer, Berlin (1986)
277. Schroeder, M.R.: Computers in acoustics. Acoustics Australia **14**(1), 9–12 (1986)
278. Schroeder, M.R.: Some new results of hearing research. Acta Acustica **11**(1), 30–43 (1986)
279. Schroeder, M.R., Strube, H.W.: Flat-spectrum speech. J. Acoust. Soc. Am. **79**(5), 1580–1583 (Ex. S0144) (1986)
280. Schroeder, M.R.: Auditory paradox based on fractal waveform. J. Acoust. Soc. Am. **79**(1), 186–189 (L) (1986)
281. Schroeder, M.R.: Speech and speaker recognition (Buchvorlage in der Akademie- Sitzung 6. Dez. 1985). Jahrbuch der Akademie der Wissenschaften in Göttingen – 1986 (1986)
282. Schroeder, M.R., Lauterborn, W., Ronneberger, D., Pottel, R.: DPI Annual Report. Report 1 (1986)
283. Schroeder, M.R.: Predictive coding of speech: historical review and directions for future research. In: IEEE ICASSP 86, Tokio, 4, pp. 3157–3164, PL.2.1–8 (1986)

284. Smith, B.K., Sieben, U.K., Kohlrausch, A., Schroeder, M.R.: Phase effects in masking related to dispersion in the inner ear. J. Acoust. Soc. Am. **80**(6), 1631–1637 (1986)
285. Schroeder, M.R.: Cooperative and interactive models of hearing an introduction. In: 11th International Congress on Phonetic Science, Tallinn (1987)
286. Schroeder, M.R.: Das menschliche Gehör: Seine exquisiten sensorischen Leistungen und Grenzen. Akademie Vorlage (1987)
287. Schroeder, M.R., Sloane, N.J.A.: New permutation codes using Hadamard unscrambling. IEEE Trans. Inf. Theory **IT-33**(1), 144–146 (1987)
288. Schroeder, M.R.: Computers in architectural acoustics. J. Acoust. Soc. Am. **81**(1), S30 (A) M1 (1987)
289. Schroeder, M.R.: Normal frequency and excitation statistics in rooms: model experiments with electric waves. J. Audio Engrg. Soc. **35**(5), 307–316 (1987)
290. Schroeder, M.R.: Statistical parameters of the frequency response curves of large rooms. J. Audio Engrg. Soc. **35**(5), 299–306 (1987)
291. Schroeder, M.R.: Speech and speaker recognition. Karger Gazette; Basel: Karger **49**, 1–3 (1987)
292. Schroeder, M.R.: Is there such a thing as fractal music? Nature **325**(6107), 765–766 (1987)
293. Schroeder, M.R.: The unreasonable effectiveness of number theory in science and communication. Rayleigh Lecture; London (1987)
294. Simon, S., Püschel, D., Schroeder, M.R.: Bestimmung raumakustischer Eigenschaften mit Musik. Fortschritte der Akustik – DAGA '87, Aachen; Bad Honnef: DPGGmbH, pp. 721–724 (1987)
295. Schroeder, M.R.: Solved and unsolved problems in concert hall acoustics. J. Acoust. Soc. Am. **83**(1), S49–S50 (A) V3 (1988)
296. Schroeder, M.R.: Number-theoretic phase arrays and diffraction gratings with broad radiation (scattering) characteristics. Phys. Acoust. **18**, 1–20 (1988)
297. Schroeder, M.R.: Selbstähnlichkeit und Fraktale in der Physik. In: Fortschritte der Akustik – DAGA '88, Braunschweig; Bad Honnef: DPG-GmbH, pp. 91–110 (1988)
298. Schroeder, M.R.: Solved and unsolved problems in concert hall acoustics. In: ASA Meeting 88, MTG Paper (1988)
299. Schroeder, M.R.: Self-Similarity. W.H. Freeman, New York (1989)
300. Schroeder, M.R.: Number theory – The paradigm of abstract mathematics has many applications in science, engineering, even artistic design. IEEE Potentials, 14–17 (1989)
301. Schroeder, M.R.: Self-similarity and fractals in science and art. J. Audio Engrg. Soc. **37**(10), 785–808 (1989)
302. Schroeder, M.R.: Selbstähnlichkeit: Fraktale, Chaos und Potenzgesetze. Physik in unserer Zeit **20**, 144–152 (1989)
303. Schroeder, M.R.: Moderne Anwendungen Gauß'scher Konzepte aus der Zahlentheorie. Gauß-Gesellschaft Göttingen, Mitteilungen **27**, 69–73 (1990)
304. Schroeder, M.R.: Models of speech and hearing and their applications: overview and outlook. J. Acoust. Soc. Am. **87**(1), S13 (A) E1 (1990)
305. Schroeder, M.R., Kuttruff, H.: Zum Geburtstag 60. J. Acoust. Soc. Am. **71**(4), 241–242 (1990)
306. Schroeder, M.R.: Chaos, Fraktale und Selbstähnlichkeit: Eine neue Agenda der Physik. Jahrbuch der Akademie der Wissenschaften in Göttingen, 22–55 (1990)
307. Schroeder, M.R.: Fractals, Chaos, Power Laws: Minutes From an Infinite Paradise. W.H. Freeman, New York (1991)
308. Schroeder, M.R.: Circularly polarized acoustic fields. The number theory connection. Acustica **75**, 94–98 (1991)
309. Schroeder, M.R.: Selbstähnlichkeit und Chaos in der Natur. G. Sessler aus Anlaß seines 60. Geburtstages gewidmet. Acustica **73**(2), 53–57 (1991)
310. Schroeder, M.R.: Gold Medal of the Acoustical Society of America. J. Acoust. Soc. Am. **89** (4), 4 (1991)

311. Schroeder, M.R.: Chaos, Fraktale und Selbstähnlichkeit. Universitas Z. Interdiszipl. Wiss. **8**, 722–735 (1991)
312. Schroeder, M.R.: Lösung des Gaußschen Kryptogramms von 1812. In: Gauss- Gesellschaft e.V. Göttingen, Mitteilungen; Göttingen, vol. 28, pp. 87–89 (1991)
313. Schroeder, M.R.: Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise. W.H. Freeman, New York (1992)
314. Schroeder, M.R.: An introduction to chaos and its control. J. Acoust. Soc. Am. **92**(4), 2402 (A) 4aMU1 (1992)
315. Schroeder, M.R.: The unreasonable effectiveness of number theory in physics, communication and music. Proc. Symp. Appl. Math. **46**, 1–19 (1992)
316. Schroeder, M.R.: From microwaves to architectural acoustics. In: 14th International Congress on Acoustics (ICA 14), Beijing, F 1–2 (1992)
317. Schroeder, M.R.: Synthetic speech: then and now. In: 14th International Congress on Acoustics (ICA 14), Beijing, P1 (1992)
318. Schroeder, M.R.: Listening with two ears. Music Perception **10**(3), 255–280 (1993)
319. Schroeder, M.R.: A brief history of synthetic speech. Speech Commun. **13**, 231–237 (1993)
320. Schroeder, M.R.: Zum Weltbild der Physik: Quantentheorie und Unbestimmtheit. Georgia Augusta; Göttingen, 35–38 (1994)
321. Schroeder, M.R.: Fraktale, Chaos und Selbstähnlichkeit – Notizen aus dem Paradies der Unendlichkeit. Heidelberg (1994)
322. Schroeder, M.R.: Zahlentheorie in der Physik. Physikalische Blätter **50**(12), 1123–1128 (1994)
323. Schroeder, M.R.: How probable is Fermat's last theorem? The Mathematical Intelligencer **16** (4), 19–20 (1994)
324. Schroeder, M.R.: Bookreview: Casti, John L. (Ed.), Searching for certainty: what scientists can know about the future. The Mathematical Intelligencer **16**(3), 72 (1994)
325. Schroeder, M.R.: How to generate thermal photons – On the computer. In: On Klauder's Path: a Field Trip, hrsg. v. G.G. Emch, G.C.Hegerfeldt, L. Streit; River Edge (NJ), pp. 197–202 (1994)
326. Schroeder, M.R.: Reverberation: theory and measurement. In: Proceedings of Wallace Clement Sabine Centennial Symposium, Cambridge, MA, pp. 75–80 (1994)
327. Schroeder, M.R.: Phase gratings with suppressed specular reflection. Acustica **81**, 364–369 (1995)
328. Schroeder, M.R.: Fractals in music. In: Fractals in the Future, hrsg. v. C.A.Pickover: St. Martin's Press, pp. 207–223 (1995)
329. Schroeder, M.R.: Objective parameters and subjective measurements related to the acoustic quality of concert halls. In: Proceedings of International Conference on Acoustic Quality of Concert Halls, hrsg. v. A.L. Sáenz, A.M. Arranz; Madrid, pp. 11–25 (1995)
330. Schroeder, M.R. (Hrsg.): Hundert Jahre Friedrich Hund. Göttingen, Nachr. d. Akad. d. Wiss. in Göttingen, II. Math.-phys. Klasse 1/1996 (1996)
331. Schroeder, M.R.: Chaos and fractals in music. Encyclopedia of Physics, ed. v. S.B. Parker; New York, 2nd edn (1996)
332. Schroeder, M.R.: The "Schroeder frequency" revisited. J. Acoust. Soc. Am. **99**, 3240–3241 (1996)
333. Schroeder, M.R.: Lucas numbers and continued fractions. Mathematics Magazine (1996)
334. Schroeder, M.R.: More on Magnus. The Mathematical Intelligencer **18**(1), 7 (1996)
335. Schroeder, M.R.: Probability and number theory in concert hall acoustics. In: Music and Concert Hall Acoustics, Conference Proceedings from MCHA 1995, hrsg. v. Y. Ando, D. Noson; London, Kap. 16, pp. 157–169 (1996)
336. Schroeder, M.R.: Number theory in science and communication – With applications in cryptography, physics, biology, digital information, and computing. Berlin: Springer, 3. erw. Auflage (1997)
337. Schroeder, M.R.: Number theory in acoustics. Curr. Topics Acoust. Res. **2**, 139–140 (1998)

338. Schroeder, M.R.: Acoustics in the audio engineering society. J. Audio Eng. Soc. **46**(1/2), 71–73 (1998)
339. Schroeder, M.R.: Concert halls: from magic to number theory. In: The Psychology of Music, hrsg. v. D. Deutsch; Academic Press, New York/London, pp. 25–46 (Kap. 2) (1998)
340. Schroeder, M.R.: Computer Speech. Springer, Berlin (1999)
341. Schroeder, M.R.: Reverberation and diffusion. In: Computational Acoustics in Architecture, hrsg. v. J.J. Sendra; Southampton: WIT Press, Chapter 1 (1999)
342. Schroeder, M.R.: Die Akustik von Konzertsälen, Physik und Psychophysik. Physikalische Blätter **55**(11), 47–50 (1999)
343. Schroeder, M.R.: Architectural acoustics. In: Wiley Encyclopedia of Electrical and Electronics Engineering, hrsg. v. J.G. Webster; Wiley, New York, vol. 1, pp. 631–635 (1999)
344. Schroeder, M.R.: Physik um die Jahrhundertwende – 1. Quanten und Unbestimmtheit. In: Europäische Jahrhundertwende, Wissenschaften, Literatur und Kunst um 1900, hrsg. v. U. Mölk; Göttingen: Wallstein, pp. 130–136 (1999)
345. Schroeder, M.R. (Hrsg.): Nachruf [auf] Friedrich Hund. In: Göttingen: Vandenhoeck & Ruprecht, Jahrbuch d. Akad. d. Wiss. zu Göttingen, pp. 193–200 (1999)
346. Strube, H.W.: Acoustic theory and modeling of the vocal tract. In: Computer speech, hrsg. v. M.R. Schroeder; Berlin/Heidelberg/New York: Springer, Appendix A, pp. 203–223 (1999)
347. Schroeder, M.R.: Erwin Meyer – Professor für Physik an der Universität Göttingen. In: Göttinger Jahrbuch 2000, hrsg. v. Geschichtsverein für Göttingen und Umgebung e.V.; Göttingen: Goltze, vol. 48, pp. 181–183 (2000)
348. Schroeder, M.R. (Hrsg.): "Computer Speech: Recognition, Compression, Synthesis" (Springer: Berlin, New York, 1999). In: Göttingen, Vandenhoeck & Ruprecht, Jahrbuch d. Akad. d. Wiss. zu Göttingen, pp. 227–241 (2001)
349. Schroeder, M.R.: Entwicklung und Zukunft der Physik. In: Wissenschaften 2001 – Diagnosen und Prognosen, hrsg. v. Akad. d. Wiss. zu Göttingen; Göttingen: Wallstein, pp. 157–178 (2001)
350. Quast, H., Schreiner, O., Schroeder, M.R.: Robust pitch tracking in the car envirement. In: Proceedings of ICASSP 2002, Orlando, FL, May 13–17, 2002, vol. 1, pp. 353–356 (on CD: 3087.pdf) (2002)
351. Schroeder, M.R.: Power laws: from Alvarez to Zipf. Glottometrics **4**, 39–44 (2002)
352. Schroeder, M.R.: Computer speech. In: Abstracts Eleventh International Colloquium on Numerical Analysis and Computer Science with Applications, Plovdiv (Bulgarien); Plovdiv, 64 [Abstract] (2002)
353. Schroeder, M.R.: Tammo's and Herman's Index. In: Past, presence and future of the Speech Transmission Index, hrsg. v. S.J. van Wijngaarden; Soesterberg: TNO Human Factors, 1–1 [Introduction] (2002)
354. Schroeder, M.R.: Sequences from number theory for physics, signal processing, and art. Acoust. Phys. **49**(1), 97–108 (2003)
355. Schroeder, M.R.: Automatyczne rozpoznawanie i synteza mowy [Automatic recognition and synthesis of speech]. Postepy Fizyki **54**, 107–112 (2003)
356. Xiang, N., Schroeder, M.R.: Reciprocal maximum-length sequence pairs for acoustical dual source measurements. J. Acoust. Soc. Am. **113**, 2754–2761 (2003)
357. Quast, H.: Speech dialogue systems and natural language processing. In: Computer Speech: Recognition, Compression, Synthesis, hrsg. v. M. R. Schroeder; Berlin/Heidelberg/New York: Springer, Chapter 4, pp. 67–106 (2004)
358. Schroeder, M.R.: Computer Speech: Recognition, Compression, Synthesis. Springer, Berlin (2004)
359. Schroeder, M.R.: Lise Meitner. Zur 125. Wiederkehr ihres Geburtstages. Jahrbuch der Akademie der Wissenschaften in Göttingen **2003**, 268–271 (2004)
360. Schroeder, M.R.: Werner Heisenberg. Erinnerungen eines ehemaligen Studenten. Jahrbuch der Akademie der Wissenschaften in Göttingen **2003**, 256–261 (2004)

361. Strube, H.W.: Acoustic theory and modeling of the vocal tract. In: Computer Speech: Recognition, Compression, Synthesis, hrsg. v. M.R. Schroeder; Springer, Berlin, Appendix A, pp. 247–268 (2004)
362. Schroeder, M.R.: Book reviews: acoustic absorbers and diffusers, theory, design and application. J. Acoust. Soc. Am. **117**, 3343 (2005)
363. Schroeder, M.R.: Music and mathematics. In: Nova Acta Leopoldina N.F. (Science and Music – The Impact of Music), hrsg. v. W. Auhagen, W. Ruf, U. Smilansky, H. Weidenmüller; Halle, 92 (341), pp. 9–15 (2005)
364. Schroeder, M.R.: Number theory in science and communication – With applications in cryptography, physics, digital information, computing and self similarity. Berlin: Springer, Springer Series in Information Sciences 7, 386 Seiten, 4. Auflage (2006)
365. Schroeder, M.R., Guicking, D., Kaatze, U.: Applied physics at the "Dritte" – Fruitful interplay of a wide range of interests. In: Oscillations, Waves, and Interactions: Sixty Years Drittes Physikalisches Institut, hrsg. v. T. Kurz, U. Parlitz, U. Kaatze; Göttingen: Universitäts, pp. 1–24 (2007)
366. Schroeder, M.R.: Number theory in science and communication – With applications in cryptography, physics, digital information, computing and self similarity. Berlin: Springer, Springer Series in Information Sciences 7, 432 Seiten, 5. Auflage (2008)
367. Schroeder, M.R.: Alltag mit der "Königin der Mathematik. Jahrbuch der Akademie der Wissenschaften in Göttingen **2008**, 157–170 (2009)
368. Schroeder, M.R.: Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise. Dover, New York, 448 Seiten, 4. Auflage (2009)

Index

A
Absorbers of underwater sound, 242–245, 272
ACF. *See* Autocorrelation functions (ACF)
Acoustic echo, 3, 7, 11, 271
Acoustics, 3–11, 15–28, 33–54, 58, 78, 93–109, 114, 145–173, 178, 199, 214, 229, 241–273, 349, 365, 404, 409, 433, 440
Adaptive design, 146
Adaptive predictive coding (APC), 206, 385, 388, 420
Aesthetics, 147, 148, 153, 306, 389
Alberich, 242, 243
Alpha rhythms, 146, 167, 169
Ambisonics, 10
Analog computer, 27
APC. *See* Adaptive predictive coding (APC)
Aperiodic modulation, 193
Apparent source width (ASW), 150, 153, 160, 164
Architectural acoustics, 13, 17, 33, 35, 42, 146–149, 152, 157, 170, 171, 417
Arrivals, 59–66, 69, 71–73, 151, 200, 317, 338, 339, 376, 380, 397, 403, 404, 435
Artificial reverberation, 9, 99–102, 366, 389, 394, 403
ASW. *See* Apparent source width (ASW)
Auditory evoked potentials, 168
Auditory filter, 118, 121, 122, 124, 125, 129, 135
Auditory neurocomputation, 164
Auditory system, 128, 133, 147, 160, 164, 414
Auralization, 9, 18
Auto-correlation, 103, 106–108, 153, 160, 162–164, 166, 169, 182, 183, 189, 386
Autocorrelation functions (ACF), 128, 130, 146, 149–151, 160, 161, 163, 166, 168–170, 271, 386

B
Background information, 42, 43, 48
Background noise, 35, 37, 38, 41, 83–85, 90
Basilar membrane, 117, 118, 121–127, 386, 421
Bayes' theorem, 42–44
Bayesian evidence, 45, 50, 53
Bayesian inference, 37, 40, 42–45, 53
Bayesian information criterion, 41, 45–46
Bayesian probabilistic inference, 34, 42
Beam tracing, 26
Bell laboratories, 114, 117, 200, 204, 213–226, 340, 363–399, 402, 410, 431, 440
Binaural, 9, 18, 99–101, 115, 117, 149, 150, 153, 154, 156, 160, 163, 165, 166
Binaural hearing, 117

C
Carl Friedrich Gauss, 181
CELP. *See* Code-excited linear prediction (CELP)
Channel impulse response, 98–99
Characteristic maximum-length sequence (MLS), 95, 102
Chinese remainder theorem, 106, 107, 178, 183
Classical reverberation time formulas, 200, 202
Close-talking microphone, 218
Coded sequence, 99, 100
Coded signal, 97
Code-excited linear prediction (CELP), 116, 199, 207–209, 388, 420
Coil spring absorber, 244, 245, 256, 258
Computer graphics, 27, 371, 389–392, 409, 431, 433, 440
Computer simulation of acoustics of concert halls, 200

N. Xiang and G.M. Sessler (eds.), *Acoustics, Information, and Communication*, Modern Acoustics and Signal Processing, DOI 10.1007/978-3-319-05660-9,

© Springer International Publishing Switzerland 2015

Concert hall, 6, 8, 21–24, 63, 69, 73, 81, 101, 115, 116, 146–149, 152, 154, 156, 158, 159, 166, 177, 199, 200, 286, 288, 349, 350, 352, 365, 368, 371, 372, 387, 389, 394, 396, 397, 409, 416, 419, 431, 433
  design, 8, 148–153, 156
Condenser microphone, 214, 224
Cone tracing, 26
Co-prime, 106
Critical bandwidth, 129, 134, 138
Critical frequency, 266–267, 349, 365
Cross-correlation, 52, 66, 82, 96, 98–100, 102, 103, 115, 150, 153, 160, 164, 166, 254, 386
  bound, 96, 97
  function, 96, 97, 99, 102, 254
Cross-over time, 57–73
Crosstalk canceler, 9, 10
Crystallography, 180

D
Decay compensation, 68, 84, 87
Decimation, 95, 96
Decimation factor, 95, 96
Decorrelation, 8, 94, 100–105
Deep water tank, 245–246, 269, 272
Degree of maximum-length sequence (MLS), 94
Delany–Bazley model, 234
Diffractal, 105, 190, 192, 193
Diffraction pattern, 108, 178, 182, 184, 185
Diffuseness, 18–20, 148, 150, 160, 164, 259–263
Diffuse reflection, 24–26
Diffusers, 19, 20, 105–109, 159, 177, 194, 256, 270–271
Diffusion, 37, 66, 72, 148, 180, 201
Diffusion coefficient, 186, 189–193
Diffusors
  aperiodic, 192
  binary amplitude, 195
  curvilinear, 194
  fractal, 105, 190
  primitive root, 105
  quadratic residue, 105, 106, 108, 185
Digital signal processing, 4, 114, 231, 371
Directional microphone, 11, 69, 216, 223, 260
Double-slope decay, 41
Dynamical properties, 58, 73

E
Early reflections, 6, 58, 70, 71, 100, 116, 148
Effective duration, 151, 152, 159, 162, 167, 169
Electret, 214–216, 218–226, 395
Electret microphone, 8, 214–223, 395
Electroacoustic transducer, 213, 214, 220–223
Electrorheological fluids, 272
Empirical distribution, 79, 85, 87, 89
Energy room impulse response, 33–37
Envelope, 37, 100, 115, 118, 119, 121, 123, 125, 128–138, 151
Envelope fluctuation, 128, 129, 132–138
Envelopment, 99–102, 104, 105, 153, 164
Ergodic, 59, 70, 71
Evidence, 44–46, 48, 50, 52, 53, 136, 321, 331, 345
Extensible Fourier transform (XFT), 58, 61, 64–70, 72, 73

F
Fast maximum-length-sequence (MLS) transform, 99
Finite-impulse-response (FIR) filter, 99, 100, 102
First-order Markov process, 203–204
Flat plate frequencies, 193
Flicker, 146, 169, 170
Flow absorber, 252, 270
Fourier transform, 78, 82, 87, 108, 189, 192, 265, 371, 391
Frequency shifting, 7, 8, 11, 253, 254, 393

G
Galois field, 95
GAs. *See* Genetic algorithms (GAs)
Gaussian, 58, 60, 72, 77–90, 128–138, 254, 272, 349, 387
Gaussian distribution, 45, 46, 60, 61, 79, 81, 83–85, 87, 128, 261
Genetic algorithms (GAs), 146, 156–160, 272
Gold sequence, 96
Goniometer, 186, 188, 189
Gradient microphone, 216–218, 223
Grating lobes, 108, 182, 189, 192, 193
Grazing incidence tube, 233–237

H
Harmonic complex tone, 115, 118–128
Headphone, 102, 117, 206, 219
Hearing research, 113–139, 386, 411–415
Hemispheric lateralization, 168, 169
Hilbert envelope, 130–132
Histogram, 85–89
Homogeneous, 27, 58, 71
Hybrid technique, 26

I
IACC. *See* Interaural cross-correlation coefficients (IACC)
IACF. *See* Interaural cross-correlation functions (IACF)
Impedance measurement, 231, 246–250, 270
Impedance tube, 230, 232, 233, 237, 244, 246–250, 272
Impulse response, 4, 6, 7, 16, 18, 26, 33–37, 41, 49, 57–73, 77–90, 93, 94, 97–101, 124, 186, 189, 190, 219, 350, 366, 387
Impulse response matrix, 97, 98
Individual preferences, 154
Integral equation, 25, 26, 267, 371
Interaural correlation, 150, 154, 155, 166, 167
Interaural cross-correlation, 103, 115, 164
Interaural cross-correlation coefficients (IACC), 100, 103, 148, 152–161, 163, 164, 167, 168
Interaural cross-correlation functions (IACF), 146, 149, 150, 152, 160, 161, 163, 165, 166
Interaural decorrelation, 100, 101
Interaural decorrelation coefficient, 100, 101
Invariant decimation, 95
Isotropy, 18, 58, 71, 256

K
Kasami sequence, 96–100, 102
Kolmogorov–Smirnov test, 79, 80, 85, 87, 89

L
Likelihood, 43–46, 48–52
Likelihood function, 44, 45, 49, 50
Lincoln Center, 3–5, 394
Linear convolution, 99
Linear feedback register, 94
Linear regression, 34, 37–42, 69
Listener envelopment, 102, 104, 105
Listening level, 149, 150, 154,

M
Masking, 115, 116, 121, 122, 125, 129, 130, 133–136, 138, 206–208, 388, 411
Matching pursuit (MP), 58, 61–64, 69–70, 72, 73
Mathematics, 113–142, 181, 301, 319, 331, 344, 345, 348, 351, 354, 362, 412, 414, 431, 433
Maximum-length sequence (MLS), 94–97, 101–105, 350
Mean free path, 16–18, 24, 69
MEMS microphone, 223, 224
Microphone (toroidal), 8, 216, 217
Microphone array, 6, 8, 9, 11, 31, 217, 220, 223, 395
Missing fundamental (flicker), 169
Mixing, 59, 70–73, 102, 313, 323
Mixing time, 59, 70–73
Model selection, 34, 42–46, 53
Modulation filter bank, 136
Multiple-slope decay, 35, 42, 53

N
Noise, 8, 34, 58, 77, 94, 116, 161, 205, 218, 233, 254, 303, 335, 344, 368, 418
Nonlinear regression, 34, 37–42
Number theory, 6, 94, 95, 109, 180, 182, 192, 271, 354, 362, 367, 375, 391, 408, 409, 413, 415, 419, 431, 433, 440

O
Optimization, 109, 130, 157, 193
Overlap, 57, 58, 63, 71, 78, 87, 246, 249, 252, 253, 255, 269, 272

P
Paired comparison tests, 150, 153, 154, 162, 164, 396
Parameter estimation, 34, 41, 42, 44–46, 52, 53
Peak factor, 113–142
Perception, 9–10, 103, 116, 136, 138, 139, 149, 161, 162, 164, 169, 231, 341, 349, 363, 372, 373, 389, 390
Perceptual error, 206
Phase grating diffuser, 106, 107
Physics Today, 177, 179, 320
Piezoelectret, 225, 226
Piezoelectric material, 225
Pitch, 7, 115, 119, 146, 150, 160, 162–163, 169, 206, 307, 309, 371, 384

Pitch perception, 119
Plate vibration, 263, 265
Posterior probability, 43–45, 48, 49
Predictive coding, 204–206, 385
Preferences, 9, 145–159, 161, 166–171
Preferred maximum-length sequence (MLS) pair, 96, 104
Preferred pair, 96
Pressure chamber, 247–248
Prime number, 105, 106, 180, 185, 353
Primitive polynomial, 95
Primitive root sequence, 105–107, 178
Prior probability, 43–45, 49, 50
Probability, 43–45, 50, 62, 69–71, 73, 78–80, 202, 203, 256
Probability distribution, 45, 48, 49, 78, 129, 202, 203
Pseudorandom noise, 100
Pseudorandom sequence, 94, 102
Psychoacoustics, 10, 71, 99, 114, 115, 118, 127, 139

Q
Quadratic residue sequence, 94, 105, 106, 108, 148, 182, 271

R
Random, 17, 25, 26, 34, 50, 51, 66, 77–81, 85, 102, 128, 130, 163, 208, 323, 329, 330, 350, 352, 372, 389, 391
Random noise, 34, 78, 79, 81, 94, 99, 100, 102, 122, 254, 255, 258
Ray approximation, 201
Ray tracing, 15–28, 42, 99, 199, 392
  on computers, 200–204
Reciprocal maximum-length sequence (MLS) pair, 95–97, 99–105
Reflection coefficient, 107–109, 230–232, 244, 272
Reflection phase grating, 177–210, 350, 396
Reflection time, 150–152, 155, 154, 159, 167, 168
Reflectors, 18–20, 22, 189, 191, 246, 248–251, 271
Reproducible, 129
Research contract, 241, 242, 245, 263, 271
Resiliently lined water duct, 248–249, 255, 263, 270
Resonance scattering, 271–272
Reverberance, 42, 99–102
Reverberation
  tail, 99–101
  tank, 242, 251–254, 256, 259–263
  time, 5, 6, 16–18, 20, 24, 33–35, 37, 38, 53, 58, 63, 70, 72, 78, 84, 89, 99–101, 116, 151, 152, 154–157, 200, 202–204, 232, 251, 252, 254, 257–259, 349, 350, 371, 396, 397
Room acoustics, 3–11, 15, 16, 18, 25–28, 33–54, 58–60, 62, 63, 70, 71, 78, 94, 99, 114, 229, 231, 232, 235, 237, 270, 271, 350, 433
Room impulse response, 6, 7, 33–37, 41, 49, 57–73, 81, 94, 99–101

S
Sabine and Eyring formulas, 203
Schroeder, M.R., 3–6, 9, 11, 15, 39, 42, 77, 94, 114, 122, 128, 131, 139, 146–148, 153, 160, 164, 171, 173, 177–179, 181, 190, 192–196, 199, 200, 205, 213–226, 229–237, 241, 242, 245, 252, 260, 270, 273
Schroeder decay curve, 36–38, 40, 41, 47
Schroeder decay function, 33–35, 37, 38, 41–43, 53
Schroeder decay model, 35, 36, 41, 46, 53
Schroeder diffusers, 270–271
Schroeder frequency, 57, 59, 60, 349, 365, 440
Schroeder integration, 33–35, 37, 53, 396, 440
Schroeder phase, 118–127, 385, 440
SCOTTs theory, 234
Seat effect, 5, 232, 237
Secure voice communication, 205
Self-imaging, 270
Self-similar maximum-length sequence (MLS), 95
Shallow water basin, 242, 250–251, 256, 265, 271
Shannon, C.E., 205, 368, 372
Shape optimizer, 194
Shell vibration, 263, 267–269
Signal phase, 125, 127
Silicon microphone, 224
Single-slope decay, 35, 42
Sound absorption measurement, 19, 256–258
Sound decay in two-dimensional rooms, 201–202
Sound field
  channel with absorber, 234
  empty channel, 234, 235
Sound power radiation, 251
Sound rays, 58, 59, 69, 71–73
Sound-soft water duct, 248–249
Sound transmission in rooms, 200

Source size effect, 235
Sparse signals, 61
Spatial audio, 3–11, 94, 102
Spatial envelopment, 99, 100
Spatial hearing, 10, 114, 117, 148
Spectral compensation, 87
Spectral estimator, 87
Specular reflection, 17, 19, 24, 26, 27
Speech coding, 115, 116, 204, 205, 207, 368, 371, 388, 389, 433
Speech perception, 138
Speech processing, 114, 231, 393
Squeeze film absorber, 244, 272
Statistical sound field theory, 252–255
Statistical test, 60, 79, 85, 87, 90
Statistics, 18, 20, 66, 68, 78, 128–138, 301, 320, 352, 371, 419
Steady-state energy decay, 34, 35
Subjective loudness, 206
Switch-microphones calibration, 233

T
TEF analyzer, 186–188
Teleconferencing, 3, 7, 10, 11, 223
Telephony, 7–9, 214, 216, 374, 385, 388
Temporal design, 146, 171
Texture, 146, 163, 169
Timbre, 102, 103, 115, 146, 152, 160, 163, 169
Toroidal microphone, 8, 216, 217
Touch dial, 219
Transducer, 8, 213–226, 246, 247, 249, 269, 270, 272, 395
Transfer function, 10, 78, 80–82, 85–89, 93, 161, 168, 230, 233, 237, 256, 260, 396
Tree search, 207, 208
Two-circuit resonance absorber, 243, 244
Two-microphone method, 217, 219, 220

U
Ultrasonic transducer, 214, 219
Underground sound, 180
Unidirectional microphone, 217, 218
Upper limit of the integration, 35, 37, 38, 40, 41

V
Variant decimation, 95
Vector base amplitude panning, 10
Viscometer, 270
Vocal tract
  geometry, 231–232, 234
  impedance, 229, 231, 234, 386
Vocoders, 115, 199, 204, 205, 367, 384–388, 411

W
Wave field synthesis, 10
Weyl, H., 57, 59, 354, 433
Wide impedance tube, 249–250
Wiener, N., 205

X
XFT. *See* Extensible Fourier transform (XFT)

Z
Zero-order Markov process, 203

PHYSICAL ACOUSTICS

Principles and Methods

Volume XV

CONTRIBUTORS TO VOLUME XV

LAWRENCE FLAX
GUILLERMO C. GAUNARD
KARL F. GRAFF
ARTHUR E. LORD, JR.
W. RICHARD SMITH
HERBERT ÜBERALL

PHYSICAL ACOUSTICS

Principles and Methods

Edited by WARREN P. MASON

SCHOOL OF ENGINEERING AND APPLIED SCIENCE
COLUMBIA UNIVERSITY
NEW YORK, NEW YORK

R. N. THURSTON

BELL LABORATORIES
HOLMDEL, NEW JERSEY

VOLUME XV



1981

ACADEMIC PRESS
A Subsidiary of Harcourt Brace Jovanovich, Publishers
New York London Toronto Sydney San Francisco

COPYRIGHT © 1981, BY ACADEMIC PRESS, INC.
ALL RIGHTS RESERVED.
NO PART OF THIS PUBLICATION MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY, RECORDING, OR ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM, WITHOUT PERMISSION IN WRITING FROM THE PUBLISHER.

ACADEMIC PRESS, INC.
111 Fifth Avenue, New York, New York 10003

United Kingdom Edition published by
ACADEMIC PRESS, INC. (LONDON) LTD.
24/28 Oval Road, London NW1 7DX

Library of Congress Cataloging in Publication Data

Mason, Warren Perry, Date ed.
Physical acoustics.

Includes bibliographies.
1. Sound. 2. Ultrasonics. I. Thurston, Robert N., joint ed. II. Title.
QC225.M42 534 63-22327
ISBN 0 12 477915 8 (v. 15)

PRINTED IN THE UNITED STATES OF AMERICA

81 82 83 84 9 8 7 6 5 4 3 2 1

CONTENTS

List of Contributors vii
Preface ix

1

A History of Ultrasonics

Karl F. Graff

I. Introduction 2
II. High-Frequency Acoustics in the 19th Century 3
III. Early Work in the 20th Century (1900–1915) 20
IV. Birth and Growth of Ultrasonics (1916–1926) 28
V. Intense Ultrasound and Other Developments (1927–1939) 41
VI. Ultrasonics Comes of Age (1940–1955) 60
VII. To the Present 89
References 90

2

Circuit-Model Analysis and Design of Interdigital Transducers for Surface Acoustic Wave Devices

W. Richard Smith

I. Introduction 100
II. General Description of Interdigital Transducers as Filter Elements 102
III. Relation of Transducer Response to Geometry 110
IV. "Second-Order" Effects and Their Compensation 122
V. Three-Port Circuit Model Description of Interdigital Transducers 132

VI. Matching-Circuit-Controlled Trade-offs among Insertion Loss, Bandwidth, Triple Transit Suppression, and Voltage Standing-Wave Ratio 160
VII. Specific Transducer Geometries (Array Factors) for Typical Filter Applications 177
References 187

3

Theory of Resonance Scattering

Lawrence Flax, Guillermo C. Gaunaurd, and Herbert Überall

I. Introduction 191
II. Acoustic-Wave Scattering from Elastic Targets 193
III. Elastic-Wave Scattering from Cavities and Inclusions 240
IV. The Inverse Scattering Problem 281
V. Resonances in Electromagnetic Scattering 285
References 292

4

Acoustic Emission—An Update

Arthur E. Lord, Jr.

Introduction 295
I. Determination of Acoustic Emission Source Characteristics 297
II. Optical Detection of Acoustic Emissions 334
III. Signal Analysis 334
IV. Acoustic Emission during Various Transformations 335
V. Dislocation Effects 340
VI. Magnetomechanical Acoustic Emission 342
VII. Monitoring Fatigue Damage with Acoustic Emission 348
VIII. Practical Applications 350
IX. Discussion 353
References 355

Author Index 361
Subject Index 369
Contents of Previous Volumes 379

CONTRIBUTORS

Numbers in parentheses indicate the pages on which authors' contributions begin.

LAWRENCE FLAX (191)
Naval Research Laboratory
Washington, D.C. 20375

GUILLERMO C. GAUNAURD (191)
Naval Surface Weapons Center
White Oak, Silver Spring, Maryland 20910

KARL F. GRAFF* (1)
Department of Engineering Mechanics
The Ohio State University
Columbus, Ohio 43210

ARTHUR E. LORD, JR. (295)
Department of Physics and Atmospheric Sciences
Drexel University
Philadelphia, Pennsylvania 19104

W. RICHARD SMITH (99)
Hughes Aircraft Company
Ground Systems Group,
Fullerton, California 92634

HERBERT ÜBERALL (191)
Physics Department
Catholic University
Washington, D.C. 20064

*Present address: Department of Welding Engineering, The Ohio State University, Columbus, Ohio 43210.

As in other recent volumes, the several themes treated here are not directly related to each other, except for their common bond to physical acoustics.

Physical acoustics is directly related to the science of ultrasonics since ultrasonic waves determine the physical properties of materials. The first chapter, by Karl F. Graff, gives a history of ultrasonics, which originated in the 19th century with the use of whistles, toothed wheels, etc. The modern science of ultrasonics was born through the development by Paul Langevin of a submarine detection system during World War I. The history of this development is covered in the chapter. Crystal transducers, oscillators, and selective wave filters followed the work in underwater sound. Dentistry and medicine provided fields for further applications. All these applications and many others are covered historically in this chapter.

Surface acoustic wave (SAW) devices are the most important recent application of acoustics to communication science and nondestructive evaluation. The interdigital transducer (IDT) is the cornerstone of SAW technology. Chapter 2, by W. Richard Smith, gives a comprehensive account of the use of circuit model analysis to design IDTs for SAW devices. The total filter design problem for the important case of SAW filters composed solely of IDTs and matching circuits is included.

The distance of an object is only the simplest information obtainable from the echoes of radar or sound waves. Because the incident wave excites resonances of the object itself, the scattered signal contains a wealth of information about the object. In Chapter 3, Lawrence Flax, Guillermo C. Gaunaurd, and Herbert Überall give a detailed discussion of resonance scattering theory, its application to acoustic-and elastic-wave scattering, and a discussion of relevant experiments.

Acoustic emissions, i.e., the acoustic noise generated in materials when they are strained, has become a widely used nondestructive method for determining the approach to failure in pressure vessels, bridges, cranes, and other mechanisms. A chapter in Vol. XI covers the applications of acoustic emissions (AE) up to 1972. Since then a large amount of work has been done on this subject and an update has become necessary. This is the subject of Chapter 4, by Arthur E. Lord, Jr. New material includes optical

detection of acoustic emissions, acoustic emissions during various transformations, and dislocation effects.

The editors owe a debt of gratitude to the contributors who have made this volume possible and to the publishers for their unfailing help and advice.

WARREN P. MASON
ROBERT N. THURSTON

—1—

A History of Ultrasonics

KARL F. GRAFF*

Department of Engineering Mechanics,
The Ohio State University, Columbus, Ohio

I. Introduction 2
II. High-Frequency Acoustics in the 19th Century 3
A. Pre-19th-Century Developments 3
B. Audibility Limits and High-Frequency Generators 4
C. Sensitive Flames and Other Detection Methods 9
D. How Fast and How Far? 13
E. Transduction Mechanisms and the Era of Electroacoustics 16
F. Contributions of Rayleigh to High-Frequency Acoustics 18
III. Early Work in the 20th Century (1900–1915) 20
A. Progress in High-Frequency Acoustic Generators and Detectors 20
B. Developments in Generating Electrical Waves and in Electroacoustics 22
C. Absorption of Acoustic Waves 24
D. Underwater Sound 26
IV. Birth and Growth of Ultrasonics (1916–1926) 28
A. The Birth of Ultrasonics 29
B. The Growth of Ultrasonics 33
C. Developments in the "Old" Ultrasonics and Related Areas 39
V. Intense Ultrasound and Other Developments (1927–1939) 41
A. Ultrasonics and the Loomis Laboratory 41
B. Appearance of Magnetostrictors 46
C. Optical Aspects of Ultrasonics 47
D. Progress in Physical Acoustics, Underwater Sound, and Bats 49
E. First Uses in Testing and Medicine 53
F. Development of Power Ultrasonics 55
G. Technical Literature 59
VI. Ultrasonics Comes of Age (1940–1955) 60
A. Transducer Materials and Ultrasonic Generators 61
B. High-Power Ultrasonics: Research and Application 64
C. Ultrasonic Nondestructive Testing, Delay Lines, Measurement, and Control ... 74

*Present Address: Department of Welding Engineering, The Ohio State University, Columbus, Ohio 43210.

PHYSICAL ACOUSTICS, VOL XV

Copyright © 1981 by Academic Press, Inc
All rights of reproduction in any form reserved
ISBN 0-12-477915-8

D. Medical Ultrasonics 79
E. Notes on Physical Acoustics, Underwater Sound, and Bats 83
F. Books 87
VII. To the Present 89
References 90

I. Introduction

The field of ultrasonics is a branch of acoustics dealing with frequencies generally beyond the audible limit. As such, the application of the principles of classical acoustics serves equally well in the audible and ultrasonic regimes. However, as noted by Hubbard (1940), "While it is a logical extension of well-known principles, its growth has been signalized by the discovery of phenomena peculiar to the new ranges of frequency." It is the research, development and use of the "peculiar phenomena" that has created the very broad field of ultrasonics. The objective of this work will be to review the history of that field.

Since ultrasonics is a part of acoustics, its development, particularly in the early years, is to some extent embedded in the broad developments in acoustics. However, the history of classical acoustics can be traced back to Pythagoras in the sixth century B.C., while investigations of high-frequency waves did not originate until the 19th century. While it will be generally possible to extricate developments in ultrasonics from the general development of acoustics, note will be made of activities in the latter having particular impact on ultrasonics.

Although studies of inaudible acoustic waves started in the 19th century, modern ultrasonics did not appear until about 1917, with investigations on submarine detection methods. Since that time, the field has grown enormously, with applications found in science, industry, medicine, and the military. The uses of ultrasound in nature, most ancient of all, are of recent and growing appreciation. The field of ultrasonics is sometimes organized into two broad areas, delineated as the high- and low-intensity applications. In high-intensity uses the intent is to alter permanently the characteristics of a system, whereas in low-intensity uses the intent is to convey information about or through a system. It is also possible to organize the field along broad use areas, falling roughly into the categories of industrial uses, communications, scientific uses, medical applications, underwater applications, and uses in nature. Both high- and low-intensity applications can be found in several of these categories. It will be this latter organization that will generally be followed within each historical period.

The review will be broken into five historical periods: (1) 19th century; (2) 1900–1915; (3) 1916–1926; (4) 1927–1939; (5) 1940–1955. While readily admitting some arbitrariness in these divisions, it seems that

reasonable justification can be attached. As best can be discovered, considerations of inaudible waves started in the 19th century. By the close of the century, a wide spectrum of developments, ranging from the discoveries of Joule and the Curies, through Bell's telephone, Galton's whistle, and Rayleigh's acoustics, had set the stage for the science of ultrasonics to appear as a unified discipline. The period 1900–1915 was one in which certain additional developments occurred. Advances were necessary in generating high-frequency electrical waves if uses for curious phenomena such as magnetostriction and piezoelectricity were to be found, and these advances were made in the second period. The third period was marked by the development of modern ultrasonics, marked also by its rapid spread into communication uses. The fourth period, 1927–1939, was highlighted by the discovery of the high-power uses of ultrasound. The fifth period was marked by a new surge in both developments and needs of electronics that carried ultrasonics into new uses.

The reader is reminded that this is an effort at an historical account, and not a review of the technical literature. Thus, the emphasis is on firsts and major contributions and not on a detailed technical evaluation of each area over the years.

II. High-Frequency Acoustics in the 19th Century

The earliest developments in acoustics have been traced back to the ancient times of Pythagoras (sixth century B.C.), and Aristotle (fourth century B.C.). In the ensuing centuries, Boethius (sixth century A.D.), Galileo (1564–1642), Sauveur (1653–1716), and many others advanced the science. However, at the opening of the 19th century, only the slightest notions of inaudible acoustic phenomena were evident. During the next 100 years, the first studies of high-frequency sound waves occurred, mainly in conjunction with pitch audibility limits of the ear. While the study of very-high-frequency waves did not progress greatly in a direct way, it could be said that nearly all of the important foundations for the modern "discovery" and advance of ultrasonics in the 20th century were laid during this period. Ultrasonic generators in the form of whistles and sirens appeared, basic transduction mechanisms were discovered, detection methods were developed, the science of electroacoustics was born, and the first uses of underwater sound occurred, along with many other developments to be described in the following sections.

A. Pre-19th-Century Developments

It has been questioned whether we can even name the age in which man first became aware of sounds that could not be heard (Wood, 1939).

Thus far, no reference has been found to the study of inaudible acoustic phenomena by pre-19th-century investigators. For that matter, no references to the possibility of such phenomena have yet been noted in the early writings. However, the fact that musical pitch and frequency of vibration were related was established and studied by several, including Benedetti, Beeckman, and Mersenne in the 16th and 17th centuries. The earliest actual determination of frequency was made by Mersenne in 1636 in studies on the vibrations of strings. Sauveur, who introduced the word *acoustics* for the science of sound, made the most thorough pioneering studies of the pitch-frequency relation in the late 17th century. Robert Hooke, in 1681, showed a way of making musical sounds using a toothed brass wheel. This device directly related pitch and frequency and provided the basis of the first ultrasonic generator of the 19th century (Whewell, 1857).

The acoustic horn, which plays an important role in several aspects of acoustics, including power ultrasonics, dates back many centuries. The first rational discussions of this device occurred in the 17th century (Miller, 1935). Thus, Kircher, in 1650, used a parabolic horn and in 1684, discussed the forerunner of the exponential horn. In 1671, Moreland described the design of a horn, and in 1672, Cassegrain proposed a hyperbolic horn. A logarithmic form of horn was suggested by Young in 1789.

A very interesting brush with ultrasonics occurred at the close of the 18th century. Lazzaro Spallanzani, a famous Italian natural philosopher still known to modern biology students as disproving the theory of spontaneous generation of life, made the first attempts to explain the ability of bats to avoid obstacles while flying in the dark. An account of this work appeared in 1794 and involved observing the obstacle-avoidance ability of bats flying under various handicaps, including cruel mutilation of the eyes. These tests led Spallanzani to advance a tentative hypothesis that bats possessed a new "sixth sense," a concept strongly denounced by naturalists of the time. However, similar tests by Jurine in 1798 then led Spallanzani to the conclusion that the ear was the means by which the bat directed its flight. This revised and ultimately correct viewpoint did not prevail against his earlier explanation, which had been discredited in favor of a "feel" hypothesis involving the wings of the bat. The true means of bat navigation, involving ultrasonic echoes, was to lie undiscovered throughout the 19th and well into the 20th century (Galambos, 1942; Griffin, 1958).

B. Audibility Limits and High-Frequency Generators

Once it was known that pitch was related to frequency, one of the logical directions of investigation was to determine the pitch limits of audibility, both in terms of low frequency and high frequency. It is not

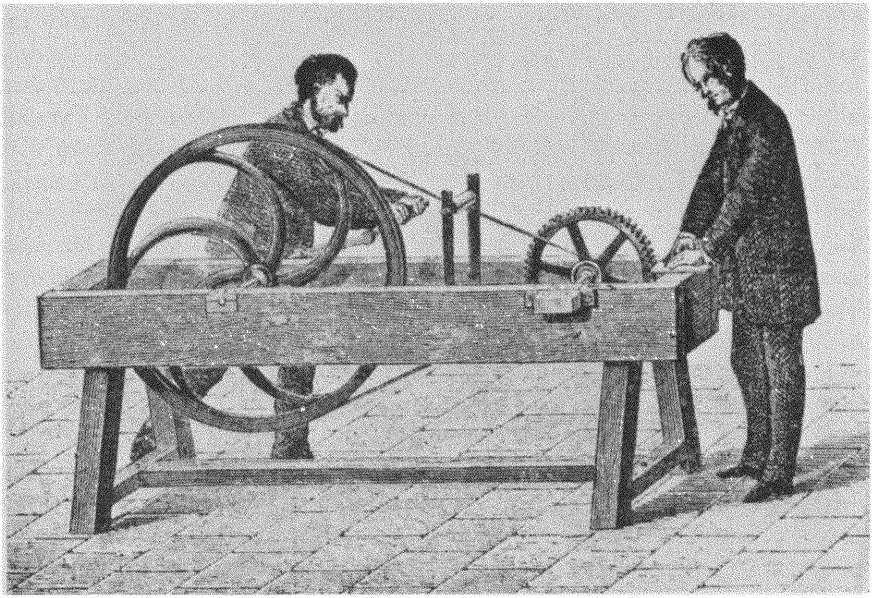


FIG. 1. The Savart wheel for determination of pitch (Miller, 1935).

known who deliberately produced the first ultrasonic waves for purposes of establishing the high-frequency limits of the ear. However, several later writers in acoustics, such as Savart (1830), Tyndall (1869), and Galton (1883b), cite the work of Dr. William H. Wollaston (1820) in this regard. A scientist with wide-ranging interests, but probably best known in his time for work in crystallography and optics, Wollaston made numerous observations on the variations in pitch audibility between individuals, and speculated that some insects might communicate by high notes inaudible to humans. He used small pipes as sound sources and may actually have achieved frequencies only up to 10,000 Hz.

The French physicist Felix Savart (1830) then conducted the most thorough early studies on pitch audibility limits. He devised a large toothed wheel, shown in Fig. 1, to achieve frequencies of up to 24,000 Hz. This device, generally known as Savart's wheel, was an elaboration of Hooke's toothed wheel, and may be tentatively identified as the first ultrasonic generator.

There were many other contributors to this area through the 1800s. Despretz (1845) set the upper limits of audibility at 30,000 Hz. However, this was based on estimates by trained musicians, and not considered reliable (Wood, 1939). Helmholtz is said to have used several types of apparatus to place the upper limit of audibility at 38,000 Hz (and the lower limit at 32 Hz) (Miller, 1935). However, in "Sensations of Tone,"

Helmholtz (1885) appears to have based the high-frequency limits on earlier works by A. Appunn and by Preyer (1876), where small tuning forks were used supposedly to reach frequencies of up to 40,960 Hz. It was noted that "these high notes were very painfully unpleasant." Tyndall (1869) referred to work by M. Depretz with very small tuning forks that showed 38,000 Hz to be audible. Rayleigh (1896) also cited Preyer's work, but as giving an upper limit of 20,000 Hz for normal ears. In his own work with bird calls, Rayleigh stated that nothing beyond 10,000 Hz was audible, but noted that such results were no doubt dependent on intensity. The fact that such variations in frequency were obtained for audibility were no doubt the result of subharmonics being generated in the instruments. This, plus the fact that widely different acoustic intensities were involved, as well as individual differences in audibility, resulted in very unreliable data on the limits of hearing.

In addition to toothed wheels and tuning forks, whistles made their appearance in the 1800s as sources of very high frequency. Rayleigh (1896) described the construction of a bird call and indicated that frequencies of up to 55,000 Hz could be obtained. For Rayleigh and many other acoustic investigators of the late 1800's the bird call was mainly a tool for generating short wavelength and for demonstrating various reflection, refraction, and diffraction phenomena. The actual means by which such a construction generated a tone was, as noted by Rayleigh, at least partially explained by Sondhauss. In fact, both A. Masson in France and C. Sondhauss in Germany noted in 1853 that a musical tone was produced by a jet of air impinging on a sharp edge (Wood, 1960). Later investigations of this phenomenon in the early 1900s were to lead to the edge tone theory of sound generation in lipped pipes, flute-type musical instruments, and the various ultrasonic whistles (Miller, 1935).

Without question, the most well-known ultrasonic generator to emerge from the 19th century was the whistle invented by Dr. Francis Galton in 1876 (Galton, 1883a). Nearly every textbook on ultrasonics uses this as the starting point of developments, for example. The whistle used edge tones and a resonant cavity to generate frequencies of 3,500–30,000 Hz in air. Switching to hydrogen gas, Galton (Fig. 2) was able to generate frequencies over 80,000 Hz (LeConte, 1883). Considered as perhaps the last of a now-extinct breed—the gentleman scientist—acoustics represented but a minute part of Galton's activities in statistics, anthropometry, experimental psychology, and heredity. He strongly believed that virtually everything in life was quantifiable. Galton was motivated to develop the whistle in order to study the audibility limits of both men and animals, with the results appearing as a small part of a book (Galton, 1883b) purporting to investigate every facet of human physical, intellectual, moral, and psychic facul-

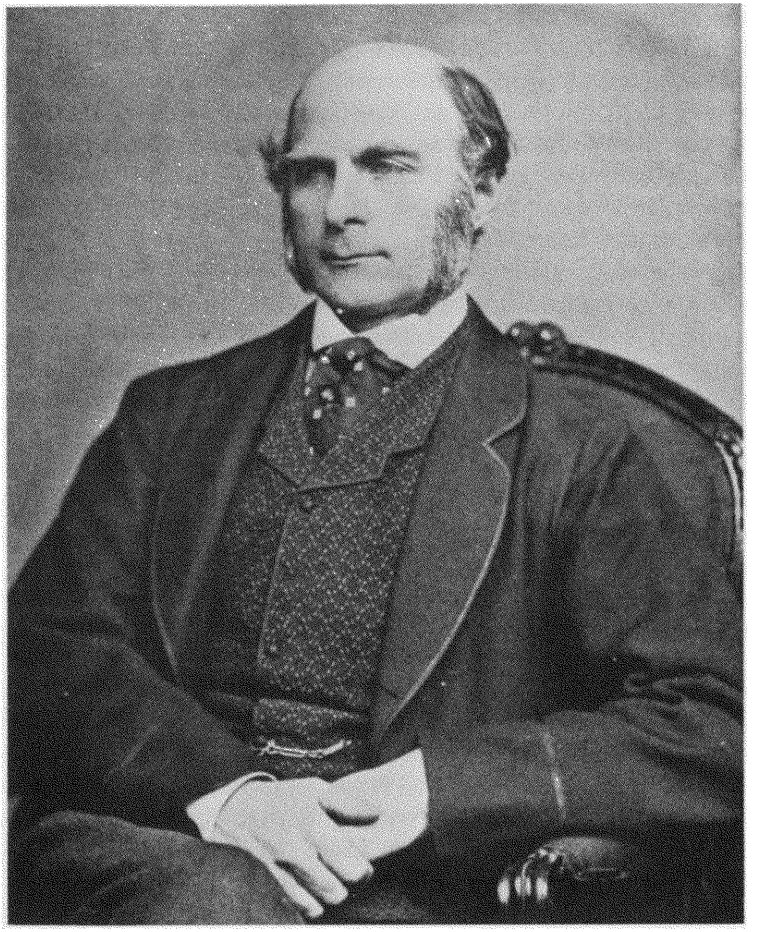


FIG. 2. Francis Galton, 1822–1911 (Pearson, 1914).

ties in order to determine whether improvements in the "human stock" could be made. Dr. Galton's description of his tests on animal audition makes amusing reading in the age of million-dollar laboratories and statistical controls. He notes, "I have gone through the whole of the Zoological Gardens, using an apparatus arranged for the purpose. It consists of one of my little whistles at the end of a walking stick. ... I hold it as near as is safe to the ears of the animals... and make a sound; then if they prick their ears it shows that they heard the whistle. ... I have walked through the streets of a town... and made nearly all the little dogs turn around. ..." Although many of Galton's studies are now viewed as histori-

cally curious, contributions such as establishment of fingerprint identification have remained. The Galton pipe was to remain for many years the standard ultrasonic apparatus for testing upper limits of hearing (Gillispie, 1971).

The most thorough studies of the limits of audibility were made by Koenig (1899) over a period of many years in the later part of the 19th century. He was known as a master instrument maker, as well as a major researcher who totally devoted his life to acoustics. His most famous set of apparatus was a tuning-fork tonemeter, a set of precision tuning forks covering the frequency range from 16 to 21,845.3 Hz. Other important equipment included the manometric-flame and improvements in the phonautograph, the precursor to Edison's phonograph. Koenig (1899) published results on the highest audible frequencies, where he used small tuning forks and bars to achieve frequencies of up to 90,000 Hz, and used dust figures to measure the resulting wavelengths. He noted his own change in frequency limits of audibility with advancing age, diminishing from 23,000 Hz at age 41 to 20,480 Hz at age 57 and to 18,432 Hz at age 67 (Miller, 1935; S.P.T., 1901; Gillispie, 1971).

Another important generator of acoustic waves, the siren, appeared in the 19th century. Professor John Robison (1801) is credited with first showing how to produce a musical sound by puffs of air, with the first discussion of the method being in 1799. It was Cagniard De La Tour who greatly improved the design in 1819 and is usually credited with the invention. He gave the name siren to the device because of its capacity to "sing under water," the same as the mythological figures of antiquity who lured mariners to their deaths by singing. Another 19th-century generalist whose career spanned many fields, Cagniard's studies on alcoholic fermentation have remained the most valuable of his works, although his reputation in his lifetime was made by the acoustical siren (Gillispie, 1971). The essential features of a modern siren were incorporated in the improvements of Seeback in 1841. Further improvements in the design were made by Dove in 1851. In a yet further improvement in 1862, Helmholtz doubled the parts of the Dove siren, leading to a version with greatly added power. Finally, Koenig in 1867 invented the wave siren for the study of tone quality. All the sirens described were used for various studies in acoustics, such as pitch discrimination and beats, and not as generators of powerful acoustic waves (Miller, 1935; Wood, 1960; Tyndall, 1903). However, powerful sirens were also devised for warning purposes in the 19th century. While sirens generally are not ultrasonic devices, their appearance in the 1800's was the forerunner of developments in the 20th century as generators of very intense acoustic waves for industrial processing and sonic testing.

C. Sensitive Flames and Other Detection Methods

The task of detecting, visualizing, and measuring sound was a far more formidable task in the 19th century than that of producing sounds. Several methods evolved, however, including the stethoscope, another somewhat similar instrument then identified as a microphone, the previously noted phonautograph, and Helmholtz's family of resonators. The writings of Tyndall, Helmholtz, Koenig, and others from that period show the many ingenious mechanical schemes used before the advent of electroacoustic methods for sound detection. However, in the case of devices specifically suited for detection of high-frequency acoustic waves, the selection of equipment was rather restricted.

One of the most interesting detection methods of the 19th century was the sensitive flame, first noted by Dr. John LeConte in 1857 while listening to chamber music with a small party of friends. Speaking with reference to one of the gas jets in the room, LeConte (1858) recounted,

> Soon after the music commenced, I observed that the flame of the last-mentioned burner exhibited pulsations in height which were *exactly synchronous* with the audible beats. This phenomenon was very striking to every one in the room, and especially so when the strong notes of the violoncello came in.

Tyndall's thorough investigation of the sensitive flame, which is not to be confused with the singing flame, established this as standard laboratory tool for the balance of the 19th century and into the early 1900s for study of audible as well as inaudible acoustic waves. John Tyndall (Fig. 3) was yet another 19th-century philosopher whose ranges of inquiry were enormous: These included heat, optics, glaciers, and the explanation of the blue color of the sky (Tyndall effect). He wrote a number of books on various aspects of science, and his one simply entitled "Sound" is still in print. He had an outstanding reputation as a lecturer, with his book on sound being based on his lectures. In the area of high frequencies, his sensitive flames provided the main basis for detecting inaudible acoustic waves until the advent of electroacoustic transducers (Miller, 1935; Gillespie, 1971; Eve and Creasey, 1945). One of the number of illustrations of his well-known account is shown in Fig. 4, with a gas burner adjusted to the point of roaring shown on the left, and the flaring configuration of the flame when subjected to sound waves from a whistle shown on the right. Tyndall described the basis of sensitivity in dramatic terms, noting that with the flame adjusted to a state of incipient flaring, "it stands on the brink of a precipice. The proper sound pushes it over." Rayleigh (1896) made further investigations of this detection method. The sensitive flame and the previously described bird call were frequently used by Rayleigh and other

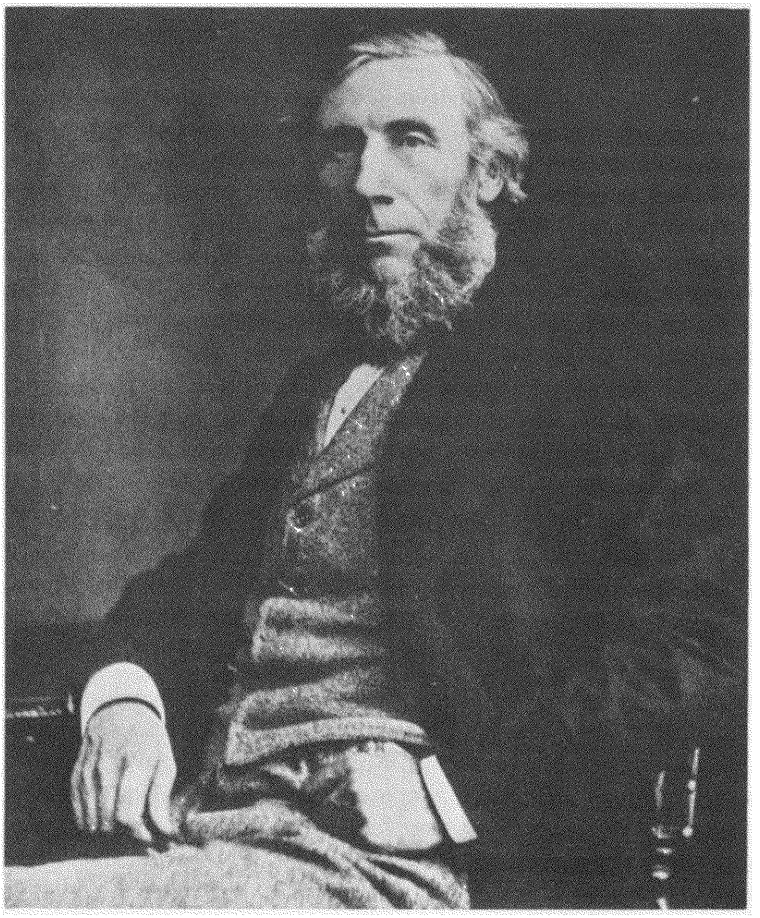


FIG. 3. John Tyndall, 1820–1893 (Eve and Creasey, 1945).

investigators to illustrate acoustical phenomena such as reflection, refraction, and diffraction.

The measurement of sound velocities relied, for the most part, on measuring the propagation time for a sound to travel long distances. Then Kundt (1866) of Berlin discovered a simple but effective method of using dust figures in a tube for obtaining sound velocities. While Kundt worked in optics and gas theory as well as acoustics, he became famous mainly for his method of determining sound velocities in gases (Gillespie, 1971). Essentially, the method involved producing standing waves in a closed glass tube by exciting longitudinal vibrations in another, smaller tube that

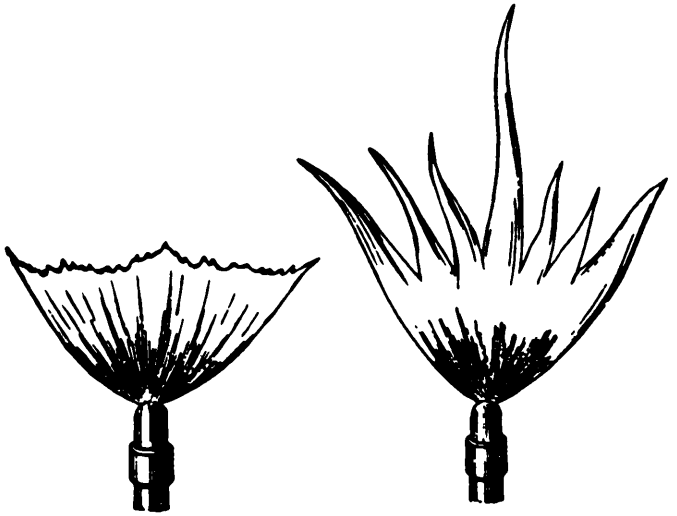


FIG. 4. Tyndall's sensitive flame (Tyndall, 1903).

acted as a piston, driving one end of the closed tube. The nodes of the waves were shown by the gathering of a light dust, such as lycopodium powder. Tyndall promptly incorporated this new method into his lecture demonstrations on sound.

Kundt's tubes are of interest in the history of ultrasonics beyond their direct use for measuring the wavelength of both audible and inaudible sound waves. It became apparent that the phenomena occurring in Kundt's tubes were more complicated than first supposed by early observers. Apparently Dvorak (1876) made some of the first observations of the unexpected air circulations that occurred in Kundt's tubes. Rayleigh (1883b), noting that "Experimenters in Acoustics have discovered more than one set of phenomena, apparently depending for their explanation upon the existence of regular currents of air resulting from vibratory motion, of which theory has as yet rendered no account," went on to present the first analysis of such currents and emphasized the importance of viscosity in their formations. These circulations were not the first encounter with acoustic streaming, as this phenomenon became known. Faraday (1831) had given the first explanation of the phenomenon with reference to vibrating plates, by noting that the tendency of very light powder to settle at the loops (instead of the nodes) was due to local air circulations above the plate. In 1827, Savart had made yet earlier observations of this effect, and, as noted by Faraday, Chladni had noted this

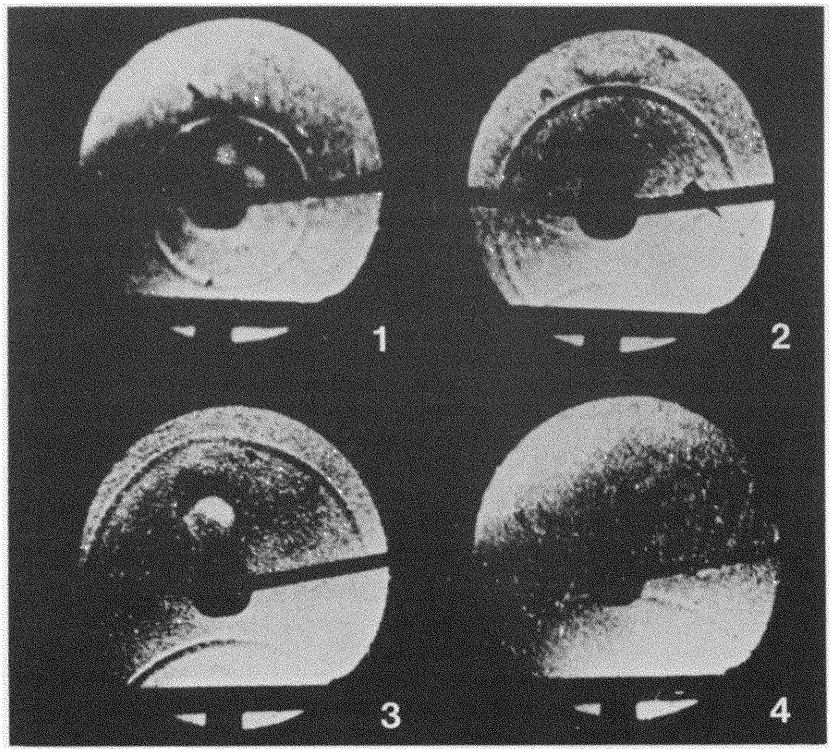


FIG. 5. Schlieren photograph of spherical sound waves reflecting from a flat plate (Wood, 1899).

anomalous behavior in the course of his famous studies of plate vibration patterns.

Additional to the streaming aspect, it should be noted that the tendency of dust to gather at the nodes in a Kundt tube represents a first illustration of agglomeration of fine particles, an ultrasonic process that was to be extensively studied in the 20th century. This may have been one of the first observations of actual physical changes in a system brought about by acoustic waves. Finally, the favored technique of producing vibrations in the Kundt tube was to induce longitudinal vibrations in the piston rod by stroking it with a wetted or resined cloth. This method of exciting vibrations, which was widely used in the 19th century, was later to result in a "new" type of ultrasonic generator in the early part of the 20th century.

In yet further efforts to visualize acoustic waves, both audible and inaudible, certain optical methods were developed. In 1828, J. B. Biot of

Paris, who had earlier made some of the first sound-velocity measurements in iron pipes, described a method for visualizing longitudinal vibrations in a glass rod by use of polarized light. Tyndall (1869) made this into a very effective lecture demonstration by using the forerunner of the polariscope to show how the strains set up during longitudinal vibration caused the sudden transmission of light. Tyndall used the rubbing method of exciting the vibrations discussed in the preceding paragraph.

A very powerful optical method for visualizing sound waves was introduced by Toepler. Using the stroboscopic method of illumination for one of the first times in exact measurements and making use of Foucault's method for making minute optical differences visible (and first using it himself to examine lens for defects), Toepler (1867a) observed the spherical waves from an electrical spark discharge. This method, well known as the Schlieren method and somewhat less well known as the striation method, was advanced by Ernst Mach in his famous studies on shock waves from bullets. Wood (1899), then at the University of Wisconsin, further improved the experimental method, and used it to photograph sound waves reflected and refracted by various obstacles. Figure 5 shows wave reflection from a flat surface. Wood, who was later to play a major role in the advancement of ultrasonics, was motivated to photograph sound waves in order to illustrate, by analogy, reflection and refraction phenomena of light waves.

D. How Fast and How Far?

Once the propagating nature of sound waves was recognized, the natural questions to ask were "how fast and how far" disturbances would travel. These were, of course, natural questions for audible acoustics and were not specifically related to high-frequency waves. However, efforts to resolve these matters of speed and attenuation in the 19th century opened lines of inquiry that were to lead to the whole field of physical acoustics, involving use of ultrasonic waves for fundamental studies of properties of materials.

Lindsay (1966) has reviewed the many early efforts to measure sound velocity. Suffice it to note that the first reasonably precise results in air were obtained in 1738 at the Paris Academy, with speeds within 1% of modern values. In the case of solids, Biot measured the speed of sound in an iron water pipe in Paris in 1808. The first scientific studies of underwater sound occurred in 1826, when Colladon and Sturm made their celebrated measurements of sound speed in Lake Geneva. Efforts to obtain a theoretical expression for the velocity of sound in air drew the attention of Newton in 1687, L. Euler in 1727 and 1749, d'Alembert in 1747, and



FIG. 6. George Gabriel Stokes, 1819–1903 (Miller, 1935).

Lagrange in 1759. In 1816, Laplace revised Newton's and Lagrange's formulations by using the adiabatic instead of the isothermal elasticity of the gas and thus obtained the accepted formula for sound velocity.

The influence of internal friction on fluid flow had been considered by such investigators as Navier, Poisson, and Saint Venant. Independently, Stokes (1845) (Fig. 6) included the effects of viscosity on the equations of fluid motion. Famous for his work in hydrodynamics, mathematics, optics, and geodesy, he was led by his work on fluid motion to the discovery of the effect of fluid viscosity on sound propagation. However, experimental means sufficiently accurate to check Stokes's theory did not exist at that time.

Many years later, in 1867, Kundt experimentally investigated the velocity of sound in narrow tubes. Helmholtz, in 1868, theoretically in-

vestigated the same problem, including friction effects, and obtained results only partially in agreement with Kundt's observations. Kundt concluded that frictional effects alone were not sufficient to account for the discrepancy and suggested that heat exchange was an important factor. This led Kirchhoff (1868) to investigate the second of what are now called the classical attenuation mechanisms, heat conduction. Known for his work on radiation, spectral analysis of light (including the concept of a blackbody), and electromagnetics, Kirchhoff analyzed the problem of acoustic waves in very general terms, including both viscosity and heat conduction. His results showed both effects to be of comparable importance in sound attenuation and also showed them to be of second order insofar as their influence on velocity was concerned. Rayleigh (1896) also noticed that the velocity of sound should be little affected by viscosity and that significant attenuation effects could be noted only at the higher range of audible frequencies. Using a simple example to illustrate his point, Rayleigh commented, "It must often have been noticed that the sound *s* is scarcely, if at all, returned by echos, and I have found that at a distance of 200 metres a powerful hiss loses its character, even when there is no reflection. Probably this effect also is due to viscosity."

Shortly afterward, Duff (1898) of Purdue University carried out experiments with high-frequency whistles with the intent of directly measuring attenuation. Using up to eight 7000-Hz whistles and the quiet of the Saint John River in New Brunswick as his outdoor laboratory, Duff measured attenuations far larger than expected from the theories of Stokes and Kirchhoff. He attributed the difference mainly to heat radiation effects. Rayleigh (1899) then reexamined the effects of conduction and viscosity as well as heat radiation effects (the latter had first been examined by Stokes in 1851) and showed the insignificance of radiation losses. In grappling with the causes of loss, Rayleigh made a suggestion that came very close to the issue when he noted energy interchanges within the gas molecules, that is, relaxation processes, could give rise to dissipation (Lindsay, 1974; Herzfeld, 1966).

Thus, at the end of the 19th century, there was serious question on the theory of sound attenuation. There was some recognition of the fact that the existing theories did not properly account for attenuation. However, other than Rayleigh's shrewd guess on intermolecular relaxation effects, the kinetic theory of gases had not been brought to bear on the matter. Dispersion had not been noted experimentally, and was relegated to a second-order effect of viscosity and heat conduction.

Another branch of physical-acoustics research that was opened in the 19th century was nonlinear wave propagation. The first efforts may be traced even further back, to Euler in 1759. In 1808, Poisson developed the

form for the particle velocity that described the effect of a finite displacement of the propagation speed. Then Stokes, in 1848, pointed out the progressive distortion and ultimate discontinuity in the wave that would occur due to nonlinear effects. Both the German mathematician Georg F. B. Riemann and the British mathematician S. Earnshaw investigated certain special cases in 1858–1859. The famous studies by Ernst Mach on shock waves appeared in 1879; the fundamental equations for such waves were derived in 1870 by Rankine and by Hugoniot in 1879. The practical importance of these studies was to be many years in the future (Beyer, 1974; Lindsay, 1973).

E. Transduction Mechanisms and the Era of Electroacoustics

The two electrical-to-mechanical transduction mechanisms that were to have the dominant impact on 20th-century ultrasonics, magnetostriction and piezoelectricity, were both discovered in the 19th century. Of these, magnetostriction made the first appearance, possibly in 1837 with the observations by Charles G. Page of musical sounds produced by interrupting current in a coil. Observations by a number of other investigators followed over the years, including those of Delezenne in 1838, Marrian in 1844 (credited with the first unambiguous observation of the magnetostrictive effect), Beatson and La Rive, both in 1845, and possibly others. All of the later investigators evidently had an intuitive understanding of the effect they were attempting to investigate (Hunt, 1954).

Nevertheless, the discovery of magnetostriction is usually credited to James Prescott Joule (1847). Noting that "About the close of the year 1841, Mr. F. D. Arstall, an ingenious machinist of this town, suggested to me a new form of electromagnetic engine," Joule, at the age of 23, undertook a very systematic investigation of the phenomena first noted by Page. He set forth preliminary observations in 1842, but continued his studies over the next several years, concurrent with his famous work on the mechanical equivalence of heat. His publication of extensive results on magnetostriction in 1847 then stimulated studies, including those of Barrett in 1882 on the volumetric effect and those of Villari in 1864 on the converse effect (Hunt, 1954; Gillespie, 1971).

Piezoelectricity was the last of the five electroacoustic transduction mechanisms to be discovered. As with magnetostriction, there were many precursors to the actual discovery, credited to Jacques-Paul and Pierre Curie (1880). Many of the background observations were tied up with the phenomenon of pyroelectricity. For example, the observation that tourmaline could attract and repel hot ashes dated far back in antiquity. Brewster introduced the name pyroelectricity in 1824. Following a conjecture by

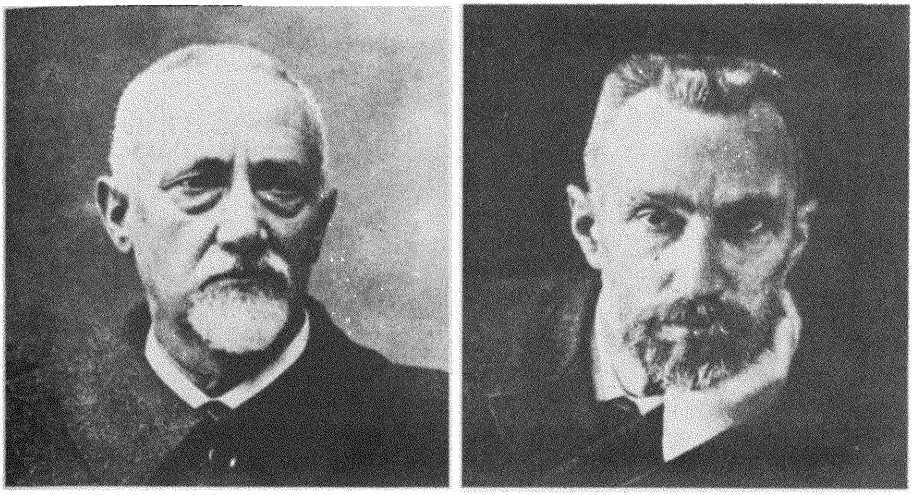


FIG. 7. Jacques Curie (left), 1855–1941, and Pierre Curie (right), 1859–1906 (Cady, 1946).

Coulomb that electricity might be produced by pressure, Haüy in 1817 and Becquerel in 1820 performed experiments in which certain crystals showed electrical effects when compressed. These observations led Lord Kelvin, in 1878, to postulate a state of permanent polarization in such crystals.

Based on their systematic studies of pyroelectricity and crystal properties, the Curie brothers (Fig. 7) were led to predict the classes of crystals and conditions under which the piezoelectric effect would be observed (Curie and Curie, 1880). Thus, they noted that "We have found a new method for the development of polar electricity in these same crystals, consisting in subjecting them to variations in pressure along their hemihedral axes." They showed that such charges were produced, that they were proportional to the applied pressure, and that they disappeared when the pressure was removed. This important discovery for electroacoustics occurred early in the brothers' scientific career, well before the marriage of Pierre Curie to Marie Sklodowska in 1895 and their subsequent work on radioactivity. Less than a year after their discovery, the converse piezoelectric effect was theoretically predicted by Lippmann, based on thermodynamic considerations. The verification of this, although more difficult to realize experimentally, was done by the Curie brothers only a few months later. Several instruments were devised by them using the piezoelectric effect for various static measurements. One, the quartz electrometer, was to find important use in later experiments in radioactivity (Curie, 1923; Hunt, 1954; Cady, 1946; Gillespie, 1971).

It was recognized in the 19th century that the piezoelectric and magnetostrictive effects were potential acoustic sources and receivers. However, the difficulty was in producing continuous electrical oscillations and coupling to solid vibrators. Hence, these new transduction mechanisms could not be truly exploited until the emergence of the continuous arcs or the vacuum-tube oscillator and amplifier early in the 20th century, although scientific interest continued in the phenomena.

The Curies' discovery of piezoelectricity in 1880 closed what has been termed the "golden age of electroacoustics" (Hunt, 1954). The period was opened by publication of Maxwell's classic "Treatise on Electricity and Magnetism" in 1873, followed by the also classic "Theory of Sound" by Rayleigh in 1877. However, the single paramount event was the invention of the telephone by Bell in 1876. This not only greatly stimulated interest in the field of electroacoustics, but had an incalculable influence on the future development of the whole field of acoustics. The remark by Lindsay (1966) with respect to work at a major industrial laboratory on the telephone, "we now know much more than might ever have been reasonably expected from mere human curiosity about the way that the human being hears and speaks," could, in fact, apply to all facets of generation, propagation, and detection of acoustic waves.

F. Contributions of Rayleigh to High-Frequency Acoustics

The 19th century was the time of giants in the field of acoustics and other branches of natural philosophy. Although high-frequency acoustics was at most a newborn area, leading scientific figures such as Helmholtz, Tyndall, and Koenig made specific contributions to the field, as noted in the preceding. From the standpoint of acoustics in general, and the emerging field of ultrasonics (not yet so named in the 19th century), the contributions of Lord Rayleigh (Fig. 8) loomed over all others.

Rayleigh was one of the last great 19th-century generalists whose work spanned all aspects of science. A partial breakdown of his 430 scientific papers shows contributions in mathematics, mechanics, hydrodynamics, properties of gases, electricity and magnetism, and optics. He and Ramsay won the Nobel Prize in physics in 1904 for their discovery of argon. His "Theory of Sound" has been called the "Principia of Acoustics," and so brilliantly summarized the field as to seemingly close it as a field of scientific endeavor, leaving it as one of engineering development. Some of his contributions relating to ultrasonics have already been noted. Others that were to directly bear on the field include

(a) His study of a liquid on a vibrating surface, which was the pioneering paper for later work on ultrasonic atomization (Rayleigh, 1883a);

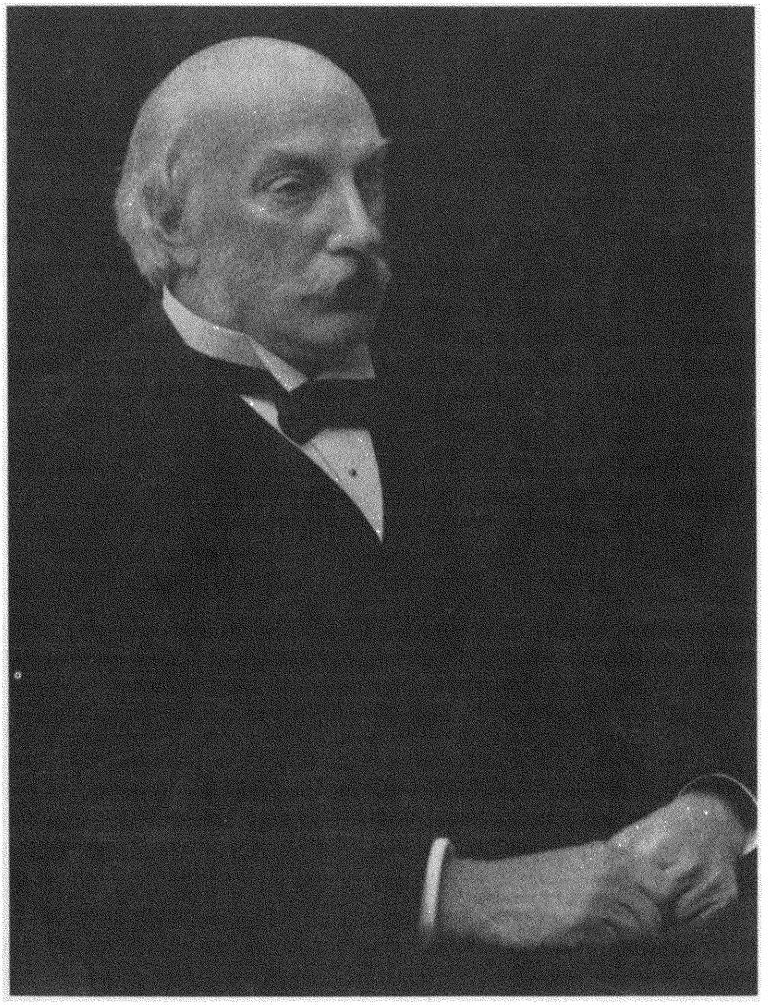


FIG. 8. Lord Rayleigh, 1842–1919 (Schuster, 1920–1921).

(b) His paper on elastic surface waves (Rayleigh waves), which was to provide a basis for mid-20th-century ultrasonic testing methods and delay lines (Rayleigh, 1885);

(c) His recognition of Waterson's contribution to the molecular theory of gases (Rayleigh, 1892) and his already noted speculation on molecular relaxation, which were precursers to the field of molecular acoustics;

(d) His study on the pressure of acoustic waves (Rayleigh, 1902), which provided the basis for measuring ultrasonic wave intensities;

(e) His work on finite amplitude waves (Rayleigh, 1910), which extended the earlier work of Rankine and Hugoniot in the field of nonlinear acoustics;

(f) His study on the pressure of collapse of a bubble in a liquid (Rayleigh, 1917), which was the pioneer model for ultrasonic cavitation.

While these various works were to have major application in modern ultrasonics, Rayleigh did not evidence direct interest in high-frequency acoustics as a field of study *per se*. However, he made frequent use of such devices as whistles and the sensitive flame to illustrate acoustical phenomena. Further, his many studies of acoustic diffraction and radiation were implicitly directed at high-frequency phenomena; that is, the results were of primary interest at high frequencies. His work bridged the 19th and 20th centuries, with his contribution on bubble collapse, two years before his death in 1919, occurring at the very birth of modern ultrasonics (Gillespie, 1971; Miller, 1935; Lindsay, 1966; Strutt, 1924).

III. Early Work in the 20th Century (1900–1915)

Developments in high-frequency acoustics in the early years of the 20th century were, to some extent, extensions of activities of the 19th century, including work on improving the Galton whistle and further development of detection methods. In addition, progress was occurring in such apparently unrelated areas as generation of high-frequency electromagnetic waves, absorption of acoustic waves, underwater sound, and electroacoustics that were to soon coalesce in the science of ultrasonics. By far the major events of the period influencing the future development of ultrasonics were the Titanic disaster of 1912 and the need, with the outbreak of World War I, to combat submarine warfare.

A. Progress in High-Frequency Acoustic Generators and Detectors

Work continued in the use of whistles for generation of high-frequency acoustic waves. Edelmann (1900) constructed an improved version of the Galton whistle that gave greater sound energy (Fig. 9). Using Kundt dust figures, frequencies as high as 110,000 Hz were recorded. The matter of the upper limit of hearing was apparently still in question at this time, since Edelmann claimed certain individuals could hear a tone of 50,000 Hz. Later, Schultze (1907) made a very careful study of the upper limits of the human ear. He measured the acoustic wavelengths using sand patterns on mica plates and a Kundt tube, and gave 20,000 Hz as the upper limit of

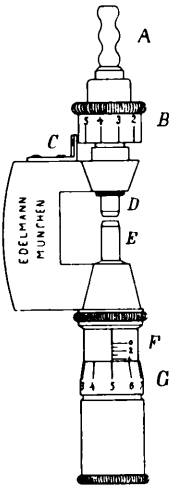


FIG. 9. The Edelmann whistle (Edelmann, 1900).

hearing. Schultze also made a number of observations on the pressure dependence of the pitch characteristics of a Galton whistle. Other developments in generating high-frequency acoustic waves by electrical arcs will be noted in Section III,B.

It was in 1902 that Rayleigh's previously noted contribution on acoustic radiation pressure was made. The existence of an electromagnetic radiation pressure had been deduced by Maxwell from his general field equations, and experimentally confirmed by the Russian physicist Petr N. Lebedev in 1899. This led Rayleigh (1902) to inquire whether other types of vibrations could exert pressure, and he obtained results for the pressure due to sound waves. He obtained more general results a few years later (Rayleigh, 1905). Shortly after Rayleigh's paper appeared, Altberg (1903) first used the principle as a means of measuring the pressure of sound waves. A disk of a torsion pendulum was arranged to measure the intensity of sound radiating from the end of a Kundt tube. Other developments in the generation and detection of sounds waves by Lebedev and Altberg will be noted in Section III,C. Note that Rayleigh's work on radiation pressure and the devices based on the phenomenon should not be confused with earlier work (Rayleigh, 1882) on a device for measuring particle velocities in acoustic waves (known as the Rayleigh disk).

Finally, in reference to optical methods, some of the techniques first developed by A. Toepler (1867a, b) in the 19th century for photographing sound waves were improved by M. Toepler (1908) by using stroboscopic illumination techniques and spark discharges. Yet further improvements to

the experimental arrangement were made by Foley and Souder (1912), yielding excellent photographs of the reflection and diffraction of sound waves.

B. Developments in Generating Electrical Waves and in Electroacoustics

A rapid pace of development was occurring in the electrical sciences by the close of the 19th century, a pace that seems to have continued to the present day. It was work involving the production of powerful, high-frequency electrical oscillations for wireless communications that was to have the greatest effect on the future development of ultrasonics.

The early experiments by Heinrich Hertz in the period 1884–1888 on generating high-frequency radio waves by discharging a capacitor through a spark gap, with refinements in 1894 by Lodge in England and in 1895–1896 by Popov in Russia, led to Marconi's invention and improvement of the wireless telegraph (from 1896 onward). The spark-discharge method of wave generation resulted in damped oscillations having a wide frequency spectrum, an undesirable feature for radio waves. Initial efforts to produce continuous electrical oscillations involved electroacoustic developments that first appeared in the 1890s.

Thus, in 1892, Elihu Thomson had showed that a direct current could be used to produce oscillatory currents by connecting capacitance and inductance across a spark gap. The arc oscillations created strong acoustical signals of up to 100,000 Hz. In addition to discovery of the "singing arc," as it was called, Thomson made many other contributions to engineering, including starting the company that was the predecessor of General Electric, inventing of electrical resistance welding, and serving as acting president of MIT in the early 1920s (Gillespie, 1971). Following Thomson's development, H. Simon in 1898 showed how the continuous arc could be used as a telephone receiver (Barton, 1908), or "speaking arc," as it was called. Then, at the start of the 20th century, Duddell (1900) improved Simon's system in 1900 to permit louder and clearer reproduction of speech. Duddell also independently rediscovered Thomson's arrangement for the musical arc in 1900. By altering the inductance and capacitance of the circuit, he was able to alter the frequency between 500 and 10,000 Hz. He also described an arrangement by which the arc could be used as a telephone transmitter.

Extending the work of Thomson and Duddell, the Danish physicist Valdemar Poulsen showed in 1902 that by immersing the musical arc in hydrogen gas, higher frequencies and more power could be obtained. The arc transmitter, which thus replaced the damped Hertzian oscillations, could deliver continuous waves at hundreds to thousands of watts, and

made modulation and broadcasting of speech possible (Elwell, 1923; Taton, 1964). Nikola Tesla, in the U.S., developed the high-frequency alternator for the same purpose. Reginald A. Fessenden, soon destined to advance the field of underwater sound, developed high-frequency alternators in the early 1900s for wireless communication. For a brief few years, the devices of Poulsen, Tesla, Fessenden, and Alexanderson were to provide the means of generating high-frequency electrical waves. The Poulsen arc was to be used to drive the first underwater ultrasonic transmitters.

Several investigators in the early 1900s used spark discharges and oscillatory arcs to generate high-frequency acoustic waves. Most well known were a series of investigations under the guidance of Lebedev at the University of Moscow. In these studies, to be discussed in more detail in Section III.C, a spark discharge was used to produce the acoustic waves. A Poulsen arc in illuminating gas was used by Dieckman (1908) to generate acoustic waves up to 780 kHz.

Returning to purely electrical developments, it was in 1907 that Lee deForest's thermionic triod, a "Device for Amplifying Feeble Electric Currents," was patented. For several years, this far-reaching invention was used as a sensitive detector of radio waves, its enormous potential unrecognized, apparently even by the inventor. The first use as a audio amplifier was in 1912. With improvements in vacuum-pump technology in the period 1913–1915 and the development of the vacuum-tube oscillator in 1915, the way was opened for development of high-power amplifiers (Chaffee, 1933; Fagen, 1975; Chipman, 1965; Gillespie, 1971). As in other applications, the vacuum-tube amplifier was quickly to replace the Poulsen arc in early ultrasonic developments.

While the practical means of generating high-frequency electrical signals were the most important developments in the early 1900s, certain theoretical concepts emerged that were to find application in ultrasonics. In 1912, A. E. Kennelly and G. W. Pierce, both of Harvard University, collaborated on some measurements of the electric impedance characteristics of telephone receivers. By chance, the two investigators followed slightly different procedures in making their measurements, and so obtained quite different impedance results. Careful observation revealed a different acoustic loading on the telephone diaphragm by the two procedures. Appreciating the significance of the effect, Kennelly and Pierce quickly worked out the theoretical explanation and thus introduced the concept of motional impedance into electroacoustics. By 1912, George Washington Pierce was several years into a long career of contributions to electrical engineering and electroacoustics that already enumerated important studies of circuit resonances and crystal rectifiers, with his most important contributions still to come (Saunders and Hunt, 1959; Gillespie,

1971; Hunt, 1954). In another development, it was shown theoretically by Butterworth (1915) that any mechanical vibrating device driven by a voltage across a condenser could be represented by a series *RLC* circuit shunted by the condenser. It was to be several years before this first equivalent circuit was rediscovered and applied to ultrasonic devices (Cady, 1946).

Finally, a few additional contributions were made by various investigators on piezoelectric crystals. Then, in 1910, Voigt published his monumental treatise on crystallography, "Lehrbuch der Kristallphysik." The piezoelectric formulations contained in that book remained the standard reference on the subject for many years. Cady (1946) has noted that Voigt came very close to being the originator of the piezoelectric resonator, since he gave the differential equations for the vibrations of crystals and mentioned the use of high frequencies in measuring the dielectric constants.

C. Absorption of Acoustic Waves

It was previously noted that discrepancies existed between predicted and measured attenuation of sound in gases at the close of the 19th century. Study of this topic continued to occupy the attention of a number of scientists in the early years of the 20th century. It will be recalled that Rayleigh speculated in 1899 on the role of energy exchange within the molecule in affecting attenuation. Although Rayleigh did not follow up his own suggestion, Jeans (1904) did so in his book "The Dynamical Theory of Gases." However, Jeans did not properly account for the exchange and concluded such effects would be much smaller than due to viscosity. Several papers were then to appear, under the general guidance of Professor Lebedev at Moscow University, investigating several aspects of absorption.

Lebedev, it will be recalled, had succeeded in measuring the pressure of light waves in 1899, thus triggering Rayleigh's 1902 analysis of the analogous acoustics problem. While most famous for his work with electromagnetic waves, Lebedev (Fig. 10) extended his research to hydrodynamic and acoustic waves, since many of their actions seemed subject to the same laws (Gillespie, 1971). He is credited with constructing a type of acoustic oscilloscope as far back as 1894 (Wołoshyn, 1963). Lebedev, said to be the father of Russian ultrasonics, was interested in generating sound waves as short as possible, and under his guidance, Altberg (1907) investigated the waves emitted by the spark discharge of a condenser. Altberg used a concave mirror to form a parallel beam of sound and an acoustic diffraction grating to determine the wavelength. A sound radiometer, employing Rayleigh's concepts of acoustic radiation pressure, was used to measure the deviated spectrum. In this way, frequencies of 332,000 Hz, higher than any previously measured, were recorded (Wood, 1939; Herzfeld, 1966).

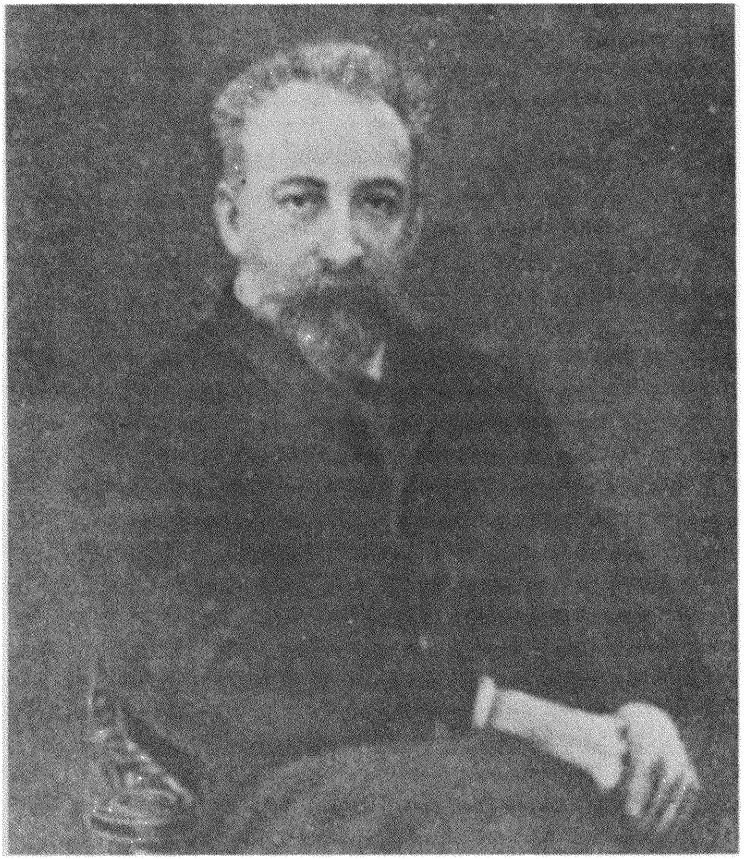


FIG. 10. Petr Nikoloevich Lebedev, 1866–1912 (Krasil'nikov, 1963).

The fact that waves of less than 1 mm were not detected was attributed by Lebedev (1911) to absorption. He then used Jeans's kinetic gas theory to calculate the attenuation of sound. In the same year, Neklepajew (1911), another co-worker of Lebedev's, used the same system as Altberg and, with acoustic waves in the range 132–415 kHz, obtained absorption coefficients still twice the predicted values.

Although no further work was to occur in this area for 14 years, these experiments were remarkable in two ways. First, they were the direct precursors of the very important branch of ultrasonics known as physical acoustics. Second, several years before the usually identified birth of ultrasonics associated with submarine detection, a total system for scientific work had been developed consisting of a source of high-frequency directed waves, a sound spectrometer, and an instrument for intensity measurement.

D. Underwater Sound

It has already been noted that the first systematic studies of underwater sound propagation were by Colladen and Sturm in 1826. However, as with other branches of acoustics, underwater sound as a means of signaling and detection traces far back in antiquity, whether to stories of the Ceylonese who signaled each other at sea by striking a submerged earthern "chatty," or to Leonardo da Vinci, who listened for ships by means of a submerged tube (Fay, 1963; Howeth, 1963). The modern science of ultrasonics is said to have been born in the sea, with the submarine-detection developments of World War I. However, the direct antecedents to the birth were in the activities in underwater sound and submarine signaling that occurred in the last years of 1800s and early years of the 1900s. In fact, as pointed out by Hunt (1972) there was already in existence in 1902, in both England and the U.S., a substantial industry devoted to exploitation of underwater sound as a means of navigation and obstacle detection.

The unreliability of light and airborne sound signals as warning means led the American Lighthouse Board to consider underwater sound signaling. In 1899, their report first mentioned use of a bell and microphone system devised by Professor Lucian Blake of Kansas. A complete report of his investigations was compiled by 1899, after which the matter was dropped. A short time later, Arthur J. Mundy, apparently unaware of Blake's work, conducted several experiments at his home in Massachusetts. Professor Elisha Gray, well known for his work with Thomas Edison on improving the telephone, joined him in this work. Gray's experiments on waterproofing telephone transmitters represented the first application of electroacoustics to underwater sound. Joshua Millet joined Mundy and Gray, and in 1901 they formed the Submarine Signal Company. Now the Submarine Signal Division of Raytheon, the company has had a continuous association with underwater sound throughout its history. By 1912, hundreds of ships were equipped with listening devices, and submarine bells were located in many areas of dangerous navigation (Lasky, 1977; Fay, 1963).

The seminal event initiating the chain of developments toward modern ultrasonics was, no doubt, the sinking of the Titanic after collision with an iceberg in 1912. Shortly after this event, Lewis Fry Richardson in England patented two schemes for obstacle avoidance, one for echo ranging with sound in air, the other for ultrasonic echo ranging with underwater sound. Both used the same principle and involved detecting the reflection of a narrow beam of sound from an obstacle, the beam being formed "by a projector having an aperture of at least three wavelengths" (Hunt, 1954). It was in this way that the theoretical acoustics set forth by

Rayleigh in the 19th century was wed to technological needs of the 20th century. However, while Richardson's specifications clearly outlined the essential features of underwater detection, including ingenious means for discriminating between transmitted and reflected signals, and means of secret, directional underwater signalling, no practical implementation occurred. [Apparently an effort was made by Sir Charles Parsons, the inventor of the vapor turbine, to build a device according to Richardson's ideas, but the apparatus was found unsuitable (Klein, 1948).]

It should be noted that the Titanic disaster brought forth many schemes of iceberg detection, not all so well conceived as Richardson's. Thus, Sir Hiram Maxim, remembered for his invention of the machine gun, revived the old sixth-sense concepts on bats, wherein bats navigated by perceiving echos from the air set in motion by their wings. He proposed ships be equipped with very-low-frequency (14–15 Hz) generators and sensitive mechanical ears connected to warning bells to detect objects (Galambos, 1942).

While Richardson's well-founded concepts did not materialize, and ill-founded concepts were being set forth, a major advancement in underwater communication was being made. In 1912, R. A. Fessenden, joined the Submarine Signal Company. After 20 eventful and sometimes tumultuous years in wireless telegraphy (Fessenden, 1940; Gillespie, 1971), Fessenden was charged with developing better methods of underwater sound signals and within a year had developed a novel form of oscillator. The Fessenden oscillator (Fig. 11) was a moving-coil transducer, weighing nearly 1200 pounds, having a large diaphragm and an operating frequency of 540 or 1100 Hz. Except for its low operating frequency, the oscillator exhibited the features of modern sonar transducers. Tests in 1914 proved the capability of detecting icebergs at distances up to two and one-half miles, and as a means of determining depth (Blake, 1914–1915). In the early years of World War I, the British installed Fessenden underwater sound telegraphy equipment on many warships, particularly on submarines. However, by 1917 this activity had ceased except for certain experimental uses. Both the English and the Americans tried out arrays of four oscillators mounted in destroyers as listening devices during the war (Hezlet, 1975; Fessenden, 1940). Ultimately, high-frequency ultrasonic methods were to prevail as means of underwater detection and signaling. However, for a number of years, the low-frequency oscillators were to find major use on ships for depth finding (Fay, 1963; Whitcroft, 1943). Many of the developments in underwater technology involving low-frequency oscillators, bells, and listening methods that occurred during this period, as well as some wartime developments, were later summarized by Aigner (1922).

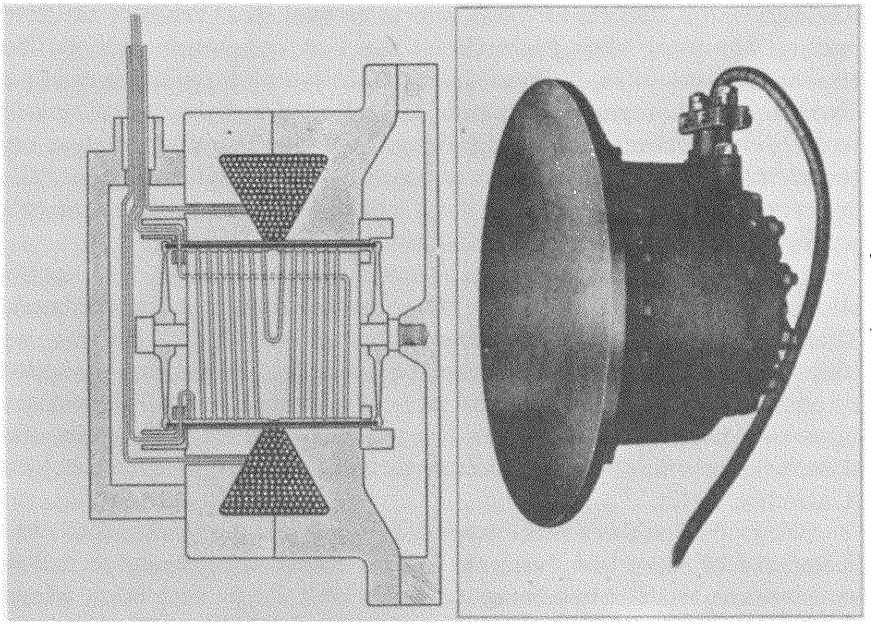


FIG. 11. The Fessenden Oscillator (Fay, 1963). Courtesy of the Submarine Signal Division, Raytheon Company.

The outbreak of the World War in 1914 turned attention to the need to detect a new kind of underwater obstacle, the submarine, and to bring together the discoveries and developments in piezoelectricity, electroacoustics, underwater sound, electrical engineering. This story will be told in Section IV.

IV. Birth and Growth of Ultrasonics (1916–1926)

It is quite evident that the field of high-frequency acoustics had an active ancestry dating well back into the 19th century. Nevertheless, the modern science of ultrasonics was born through the developments by Paul Langevin of a submarine-detection method during World War I. The story of Langevin's work has been told at length in several places, but possibly in the most accessible form by Hunt (1954). If each technology can be permitted to have its golden age, possibly the very opening decade of ultrasonics, 1917–1926, could be so designated. Langevin's inventions coalesced progress in piezoelectricity, theoretical acoustics, and electrical engineering at a time of pressing needs in such diverse areas as underwater

sound, communications, and physical acoustics. A rapid dispersion of the new developments occurred, both in scientific and geographic terms. The decade, marked primarily by progress with the new implements of ultrasonics, still saw progress in the "older" facets of the field and had a number of important developments in related fields. The golden age was to close on the eve of the next landmark event that would open the field of power ultrasonics.

A. The Birth of Ultrasonics

The outbreak of World War I in August 1914 and the menace of the submarine focused attention anew on means of underwater detection. Both France and Great Britain launched intense development programs in this area and put several types of listening devices into emergency service. The U.S. joined this effort after entry into the war, so that by 1918, an impressive variety of passive submarine detectors had been developed, including such items as C-tubes, "3-spot," the "Eel," the "Rat," and "Drifters Sets" to name but a few (Hayes, 1920, 1962; Drysdale, 1920; Klein, 1967; Lasky, 1974a). Close communications were established between the scientists working on the projects, so that a new development by one group became know to all. Even in the earliest stage of work, the utility of the piezoelectric effect was considered. It was noted that Sir Ernest Rutherford "was hopefully, if not very optimistically, scratching small pieces of quartz crystal (with a telephone headpiece connected) to discover if the piezoelectric effect of quartz was likely to prove useful" (Wood, 1962).

The key developments leading to modern ultrasonics occurred in France and started in 1915. A young Russian electrical engineer, M. Constantin Chilowsky, proposed a plan for submarine detection based on echo ranging that revived the original ideas of Richardson. After a somewhat roundabout procedure, it was forwarded to the proper ministry of the French government, who in turn asked Paul Langevin for an evaluation (February 1915). Langevin (Fig. 12) was a brilliant physicist, at the time a professor at the School of Physics and Chemistry in Paris, and already well known for major work on magnetic phenomena and for work on ionization. He was also known for pioneering thinking on the equivalence of mass and energy at the time of Einstein's development of relativity theory. It is interesting to note in passing that as a student in 1891, Langevin worked briefly with Pierre Curie, the co-discoverer of piezoelectricity (Biquard, 1972; Gillespie, 1971; CNRS, 1950).

After examining Chilowsky's original idea, which was based on using a magnetic transducer to drive a diaphragm, and considering and rejecting

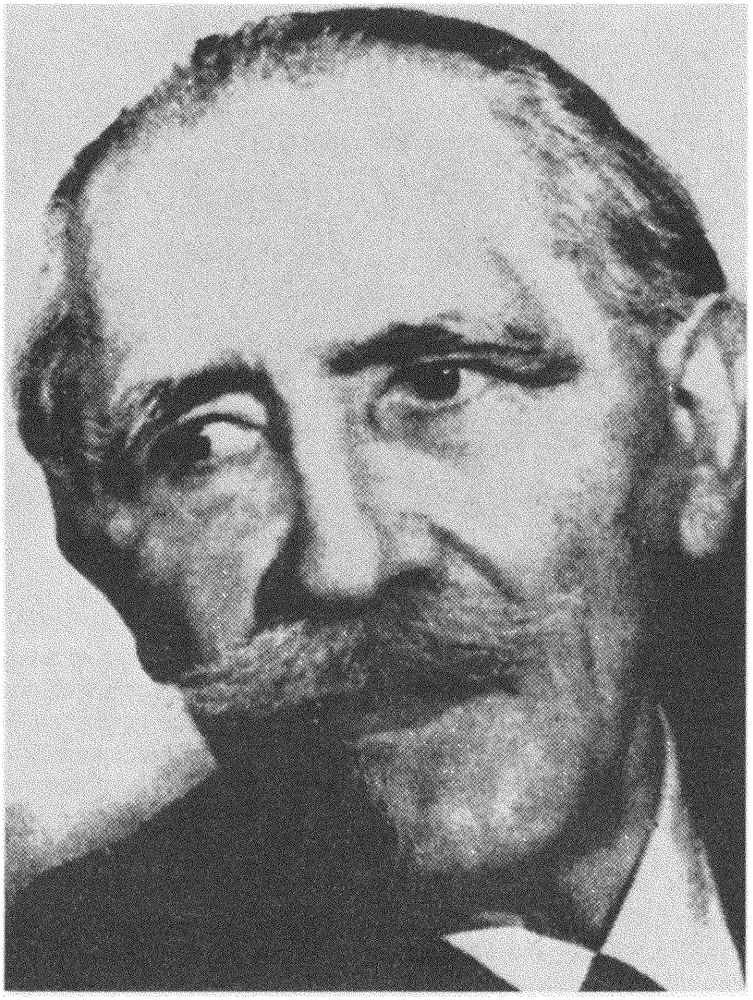


FIG. 12. Paul Langevin, 1872–1946 (Cady, 1963).

use of magnetostriction and piezoelectricity, Langevin decided to use the singing condenser approach. An experimental program was launched at Langevin's laboratories at the School of Physics and Chemistry in March 1915. The first tests were carried out in a tank $20 \times 10 \times 5$ ft. A Poulsen arc was used to drive the condenser–transducer at about 100 kHz, and a torsion pendulum was used to measure radiated power. By July 1915, after some redesigning of the transducer, power levels of a few tenths watt per square centimeter were obtained. By February 1916, tests were moved to

the Seine River and, using a carbon button microphone as a receiver, transmission distances of 2000 yds were achieved. On the basis of results with the singing condenser transducer, Chilowsky and Langevin applied for a patent on a complete submarine detection system (CNRS, 1950). The working relationship between Langevin and Chilowsky was apparently less than smooth, with the result that Chilowsky left the project at about that time. The test program was then transferred to a base at Toulon, on the Mediterranean. Further improvements in the transmitter and microphone were made, and the first echos from a target plate were obtained at a distance of 200 m by late 1916.

Langevin's work had been communicated to the British and, at the direction of Rutherford, a research group headed by Dr. Robert W. Boyle was organized. Using the electric singing arc of William Duddell and a laboratory in South Acton, the British group was, by mid-1916, paralleling the work of Langevin. By early 1917, work on both sides of the Channel was under the new impetus of unrestricted submarine warfare.

At this time the use of the piezoelectric effect, considered and discarded several times in the past as being too weak, was resurrected under more favorable circumstances. First, French-designed high-frequency vacuum-tube amplifiers became available to Langevin. Second, slices of quartz crystal of substantial size were obtained by Langevin from a Paris optician. These developments were to transform the problem completely. Using a quartz slice imbedded in wax as a receiver, excellent results were achieved, so that by November 1917 signals from the electrostatic projector were detected by a quartz receiver at a distance of 6 km. During the same period, the pure quartz crystals were also used as sound propagators, with spectacular results (April 1917). At the fundamental frequency of 150 kHz, the estimated power output of the single crystal was about 1 kW. Langevin reported that "fish placed in the beam in the neighborhood of the source operating in a small tank were killed immediately, and certain observers experienced a painful sensation on plunging the hand in this region."

There was then to follow a keystone invention in ultrasonics. While the quartz-transmitter results were naturally most encouraging, the voltages required were too high for practical use. Further, there would be great difficulty in finding quartz specimens in any number of the size required. These considerations led Langevin to conceive of the steel–quartz–steel sandwich construction, where the resonance of the transducer was determined by the overall thickness of the composite assembly. In the first tests, only single pieces of quartz were cemented between the steel plates. In order to radiate more energy, and at the same time to concentrate the beam, Langevin increased the lateral dimensions of the steel slabs to 20 cm

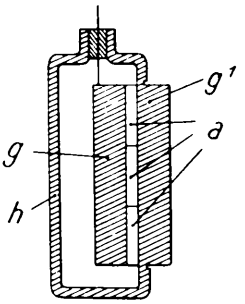


FIG. 13. The Langevin sandwich transducer (CNRS, 1950).

or more and cemented a mosaic of small, irregular slices of quartz ground to the same thickness. Tests following the completion of such a transducer in February 1918 increased the range of transmission to 8 km and for the first time clear echos were returned from a submarine. Further improvements led to echos being returned from up to 1500 m. This work was also the subject of a patent application by Langevin (CNRS, 1950), where the famous sandwich construction was disclosed for the first time (Fig. 13). The close exchange of information between the British and the French had permitted the group under R. W. Boyle nearly to parallel Langevin's developments, including the sandwich-transducer construction. Success in receiving an echo from a submarine was achieved at about the same time. Due to the continuous nature of the generated waves, however, it was not possible actually to range the targets.

With the entry of the U.S. into the war in April 1917, the exchange of scientific information was broadened. Even before this date, however, President Wilson had organized the National Research Council, headed by Robert A. Millikan. There were numerous committees and subcommittees for various special purposes, one being the Committee on Crystal Detectors headed by Walter G. Cady. Following declaration of unrestricted submarine warfare, a special antisubmarine committee, headed by Millikan, was formed. Special boards and committees were also formed by the U.S. Navy on submarine detection. As early as April 1917 work started in "supersonics," as it was identified in the U.S. for many years, at Columbia University under Professor Pupin and Professor Wills, using the tuned-condenser-type transducer. (Cady, 1963; Klein, 1967; Howeth, 1963).

It was in June 1917 that the French and British developments in ultrasonics were brought to the shores of the U.S. A French–British commission, including Rutherford, and Professor Abraham, and Professor Fabry of the University of Paris, met 1 June 1917 with a limited number of U.S. Navy officers and other representatives, including Millikan,

Mendenhall, and Wood. There followed a much larger meeting on 14–16 June 1917, attended by about 50 U.S. representatives, including groups from Western Electric, General Electric, and the National Bureau of Standards, plus representatives from about 20 universities. It was at this meeting that the French representatives communicated the successful results with piezoelectric crystals just being achieved by Langevin.

As a result of this meeting, several programs were initiated to devise improved submarine-detection systems, some of which led to the passive listening devices previously noted. In addition, specific programs to follow up Langevin's work were started at several places. These included General Electric; Western Electric, where A. M. Nicolson worked on applications to the telephone; the New London Naval Experiment Station, including Professor Pupin, Professor G. W. Pierce, Professor Cady and Professor S. L. Quimby; Columbia University, involving Professor Pupin, Professor Morecroft, and Professor Wills; Stanford University and San Pedro on the west coast; and the National Bureau of Standards. Professor Cady, of Wesleyan University, was involved in several of the programs. The most significant progress was made by the Columbia University group. Using a Pupin-designed amplifier, a Morecroft oscillator, a quartz sandwich transducer constructed by Wills, and a Rochelle-salt hydrophone designed by Cady, tests were first carried out in a large tank and then, by early 1918, in the Gulf of Mexico off Key West. Signals were transmitted up to 3 miles, and a pronounced Doppler effect was noted due to target motion. Work was then transferred to the Naval Experiment Station at New London, Connecticut, where some successful echo tests were made by the time of the armistice on 11 November 1918 (Klein, 1967, 1968; Cady, 1963).

Despite the enormous promise of ultrasonic detection, its development came too late in the war to be of use in combatting the submarine. In the U.S., developments had not yet caught up to the French–British level. Nevertheless, under wartime necessities, the new discoveries in ultrasonics had been widely disseminated, so that the stage was set for rapid developments in the postwar years.

B. The Growth of Ultrasonics

In tracing the growth of ultrasonics in the postwar years through 1926, it is logical first to continue with underwater applications. The Allied governments were working to put ultrasonic detection systems into operation when the war ended, with the British taking the lead in this regard. The momentum of development and information exchange was to continue for a short period. Before collaboration ended, the British demonstrated their perfected ASDIC System, as they called it, to the U.S. Navy.

[ASDIC, the British acronymn for ultrasonic echo ranging has been variously described as originating from Anti-Submarine Detection Inventions Committee (Howeth, 1963) or Anti-Submarine Division-ics (Lasky, 1974b).]

In the U.S., the Navy continued a small group, headed by Dr. H. C. Hayes, on problems in underwater sound at Annapolis. The Naval Research Laboratory was established in 1923, and the group under Hayes formed the Sound Division and undertook investigation of many aspects of underwater sound. A major task of the division was improvement of the quartz sandwich transducers developed in the war. This work continued for several years and resulted in improvements over transducers obtained from the French. Toward the later part of the 1920s, attention turned to the use of Rochelle salt instead of quartz in transducers, and in the ensuing years, considerable effort was made in this direction (Klein, 1963). By 1921, Professor Pierce, who was involved with the wartime developments in underwater detection, was providing the first formal instruction anywhere in underwater sound signaling, offered at Harvard University (Hunt, 1972).

In France, Langevin continued his experiments and, in the course of his work, observed the diurnal variation of echo ranges that came to be identified as the "afternoon effect." Although he correctly surmised it as being due to refraction, it was to be 18 years before his conjecture on this important feature of echo ranging was verified (Hunt, 1972). Langevin and some French entrepreneurs also exploited the commercial possibilities of ultrasound through development of a depth sounding system. The Langevin–Florisson system was installed in many ships during the 1920s and represented one of the main practical applications of ultrasonics for many years (J.B., 1925). Langevin was also interested in the use of ultrasonics for underwater telephony and carried out successful experiments in this area.

Although Langevin apparently did not publish any extensive scientific writings on his work in ultrasonics, his 1923–1924 university course was devoted to problems in this area, and some of the notes from this were published by others (Biquard, 1972). While he made further contributions to ultrasonics in the matter of radiation pressure and the equations of fluids, the field of acoustics was to remain a small part of Langevin's scientific world. Strongly associated with Einstein's concepts of relativity, Langevin's international influence in physics became paramount in 1928 when he became president of the Solvay International Physics Institute. Throughout his career, he was also known for his activity in many social–political causes. As a final footnote to his work in ultrasonics, he was invited by the French Navy to direct research on ultrasonic depth

finders. This was in 1940, during the final days of the Third Republic. Langevin died in 1946 (Gillespie, 1971).

It should be noted that ultrasonic sounding did not immediately displace low-frequency methods. The Submarine Signal Company continued development of sounding equipment based on the Fessenden oscillator, operating at 540 or 1050 Hz. The company also developed the "Fathometer," based on sending out short pulses from an impact oscillator and measuring the time interval for the returning echo. This apparatus was installed on large numbers of ships through the 1920s (Fay, 1963; Whitcroft, 1943; Lamson 1930).

By 1926, numerous schemes for underwater uses of ultrasound had been put forward. These included telegraphy and telephony, depth soundings, location of submerged objects, iceberg detection, and as underwater beacons (Boyle, 1928). Whether its present-day use as a fish finder was also proposed is not known. Nevertheless, in the years ahead, the main thrust for future development was to come from its military uses for submarine location.

A totally different application of ultrasonics, or piezoelectric engineering, that soon arose from war-stimulated research, and that was to have far-reaching consequences, was crystal control of frequency. This had its inception with the work of Cady. Professor Cady, it will be recalled, was associated from the start with the "supersonics" effort. He was later to note that "In the course of my tests of quartz and Rochelle salt in 1917 and 1918, I had noticed that specimens cut from these crystals reacted in a peculiar way on the driving circuit when the frequency was close to that for a natural mode of vibration. In quartz, the effect was extraordinarily sharp." (Cady, 1963). Continuing his research after the war, Cady discovered that when a quartz crystal was connected in the circuit of a vacuum-tube oscillator a remarkable stabilizing action was exerted on the frequency of oscillation. An early resonator configuration is shown in Fig. 14. These results were communicated via the technical literature in 1921 and 1922 and by the U.S. Patent applications filed in 1920 and 1921 (Hunt, 1954).

These disclosures, as it was to turn out, were to be but the opening round of a saga of intertwined technical development and legal battles that was to continue for over 30 years. It will be recalled that the Western Electric Company was represented at the important June 1917 conference where results of French–British efforts in supersonics were disclosed to U.S. representatives. Following the conference, Western Electric elected to pursue possible telephone applications of piezoelectricity, and A. M. Nicolson headed this effort. Nicolson's work was on Rochelle salt, and he demonstrated a number of electroacoustical devices such as loudspeakers,

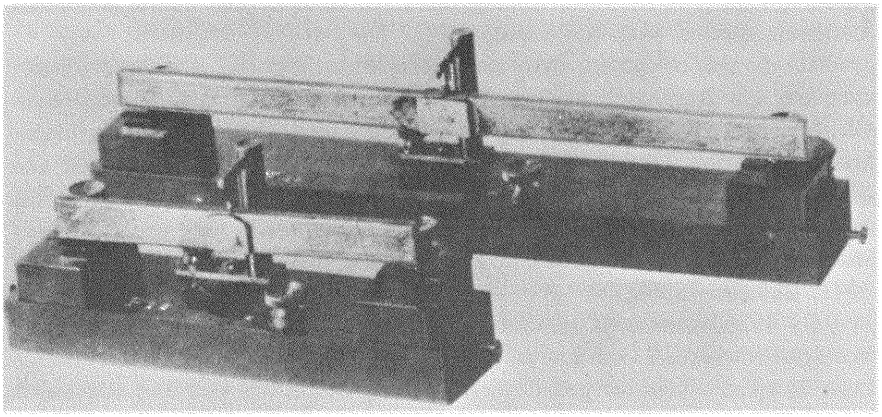


FIG. 14. Early types of piezoresonators (Cady, 1946).

microphones, and phonograph pickups (Fagen, 1975). These results were communicated by a patent application of 1918 and publications in 1919. Based on a division of the 1918 application, Nicolson's attorneys requested an interference against Cady's two patents and the legal battle was joined.

In the years that followed, Cady continued significantly to influence the direction of crystal frequency control. In the course of widely communicating his findings, he stimulated Professor G. W. Pierce of Harvard to take up the study of the crystal oscillator. Within a few months, Pierce invented several improved forms of the oscillator. Pierce duly published and patented his findings in 1923–1924 and duly encountered interferences from Nicolson, based on Nicolson's original 1918 patent and subsequent divisions. Ultimately, these matters were to decide in favor of Pierce (in 1938), and the famous Pierce oscillator circuits were to pass into almost universal use in the fields of frequency standardization and frequency control for radio transmitters and receivers (Saunders and Hunt, 1959; Hunt, 1954).

Cady's colleague at Wesleyan, Professor K. S. Van Dyke, had also undertaken the study of piezoelectric resonators. By 1922, he had made some improvements in Cady's original circuits. Cady (1963) was to note that the lack of a proper equivalent circuit hampered early work on piezoresonators. However, Van Dyke (1925) went on to develop the accepted equivalent circuit of the resonator consisting of a series *RLC* circuit in parallel with the electrostatic capacitance of the crystal. Neither Cady nor Van Dyke was aware of the earlier cited work of Butterworth on

equivalent circuits. However, independent of Van Dyke, Dye (1926) of England started with the work of Butterworth and derived the same equivalent circuit for the resonator.

Other aspects of resonators were studied in the mid- to later 1920s. One facet of investigation was initiated by Giebe and Scheibe (1925) with their work on luminous piezoresonators. A strong glow was noted in the vicinity of the nodes of resonators, especially when tested in a partial vacuum. A number of striking photographs of luminous fields were obtained for various resonators. The first observations of the strong air currents from the faces of resonators, known as "quartz wind," were made by Meissner (1926), and later used as a basis of a quartz motor. Work through this period was later summarized by Vigoureux (1931).

Frequency stabilization and frequency control were to have a profound effect on the whole development of the science of electronic communications. No doubt the recognition of this potential accounted for the closely contested legal battles over the various controlling patents. The original Nicolson versus Cady interference was finally decided in favor of Nicolson in 1940. In "Notes on the History of the Piezo Resonator," Cady (1946) gives a dispassionate accounting of these developments that ultimately led to the loss of the legal war in the area of crystal frequency control. However, in the view of Hunt (1954), in a legal decision involving Pierce and the American Communication Company in 1953, the credit was assigned to Cady as the first inventor of the crystal-controlled oscillator.

The scientific uses of ultrasound could be classified as the third major application that emerged from the developments of World War I. Professor R. W. Boyle, who had led the British development of ASDIC, returned to Canada to continue basic studies in ultrasonics and to work on certain practical systems. For the next several years, he and his co-workers were responsible for much of the research in ultrasonics, with at least 10 scientific publications appearing in the period 1922–1926, and over a dozen after 1927. A number of the studies were concerned with simple methods of detecting and visualizing sound fields, such as observing the behavior of pieces of cork or streams of bubbles and using a stethoscope. In another study Boyle and Taylor, in 1925, used dust patterns to measure standing waves in water to see if there was a frequency dependence of wave velocity. In 1926, Boyle and Taylor obtained reflection coefficients of ultrasonic waves from steel, granite, and ice, as part of their studies of iceberg detection. This followed up work done by Boyle in World War I, when he had made the earliest measurements on the transmission of ultrasonic waves through plates (Richardson, 1952). Boyle also made contributions in developing the torsion pendulum, the study of beam patterns due to diffraction, reflection, and interference effects and studies

of cavitation. These and many other results were later summarized by Boyle (1928) in the first major review paper on ultrasonics.

During this period, Rich and Pielemeier (1925) used a piezoquartz generator and a torsion vane receiver to study sound absorption in gases, and obtained a value between the theoretical expression derived by Lebedev and the experimental value obtained by Neklepajew. In the same year, Quimby (1925) used a composite resonator for the first time to make more precise measurements of the elastic constants of solid bodies. He determined that the correction for the influence of internal friction on the sound velocity was small. His work was done at Columbia University and thus represented an outgrowth of the wartime-initiated work of Wills and Morecroft.

Unquestionably, the single most important advance in the scientific uses of ultrasound was Pierce's (1925) development of the acoustic interferometer (Fig. 15). Using his previously developed oscillator circuits, stationary waves were formed between a vibrating quartz plate and a stationary reflecting surface, so that the precise measurement of acoustic wavelength in gases at very high frequency was possible. Working at about 1 MHz, Pierce noted, for the first time, dispersion in CO_2 and noted the very high absorption. Although he made no effort to explain these anomalous effects, Pierce's work brought to a close the 14-year gap since Neklepajew's 1911 work on absorption. The interferometer method was to prove extremely popular for measuring velocity and absorption in gases and was to spawn an enormous number of papers in the years ahead, as well as being extended to use in liquids and solids

Finally, it should be noted that the subject of ultrasonic cavitation received its first serious treatment during this period. This phenomenon had been observed in Langevin's early experiments and, apparently, some

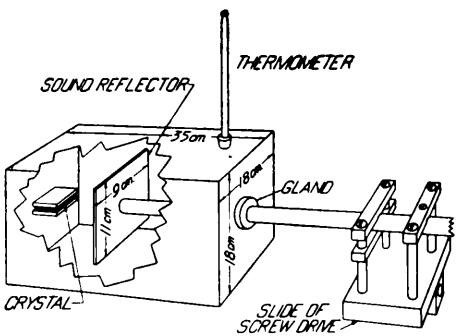


FIG. 15. The Pierce acoustic interferometer (Pierce, 1925).

American experimenters made unpublished observations in this area as early as 1918. It appears, however, that Boyle and his co-workers made the first serious efforts to study the phenomenon and, in the period 1922–1926, noted the effects of static pressure, liquid volatility, and power density on cavitation (Boyle, 1928). It was Lord Rayleigh who, in 1919, provided the first model for predicting pressures during collapse of cavitation bubbles. Since Rayleigh's work was applied to hydrodynamic cavitation, it does not appear that early investigators of ultrasonic cavitation were aware of the existence of this analysis.

C. Developments in the "Old" Ultrasonics and Related Areas

While developments in piezoelectric engineering were to dominate progress in ultrasonics from 1917 onwards, advances still occurred in the older facets of the technology as well as in areas closely related to ultrasonics. In the case of generation of ultrasonic waves, for example, one finds Palaiologos (1923) using a combination of old and new technologies to generate acoustic waves up to 200 kHz. Using the old concept of the electric arc but the new vacuum tubes to provide the AC current, he attempted to measure the possible variation of velocity in air with frequency. At least one contribution involving the Galton whistle may be noted (Schaefer, 1924). Both Lang and Boyle, in 1922, used the simple technique of setting a short metal bar into longitudinal oscillations by impact as a means of generating high-frequency waves (Boyle, 1928). Wente (1917) used an electrostatic transmitter whereas Hewlitt (1922) used an electrodynamic oscillator as a wave generator. A new twist was given to an old idea by Holtzmann (1925) to generate ultrasonic waves. By mechanizing the technique of longitudinally rubbing a glass or metal rod to induce vibrations, used by Tyndall and other pioneer acousticians, Holtzmann was able to generate waves of up to 33 kHz with considerably more power than the Galton whistle.

The first new development in whistles since the Galton whistle of the 1800s appeared when Hartmann (1922) introduced a gas current generator, known since then simply as the Hartmann generator. The essential feature of the device was that the acoustic waves were produced by the impingement of a supersonic jet on a cavity. By using hydrogen instead of air, frequencies of over 300 kHz were produced. Hartmann made considerable use of the knowledge of supersonic jets developed by Mach, Prandtl, and others in the late 1890s and early 1900s (Fig. 16). Hartmann's work in this area had started as early as 1916, and was to continue for many years, resulting in a monograph on the air-jet generator (Hartmann, 1939) and improvements to the device in the 1940s.

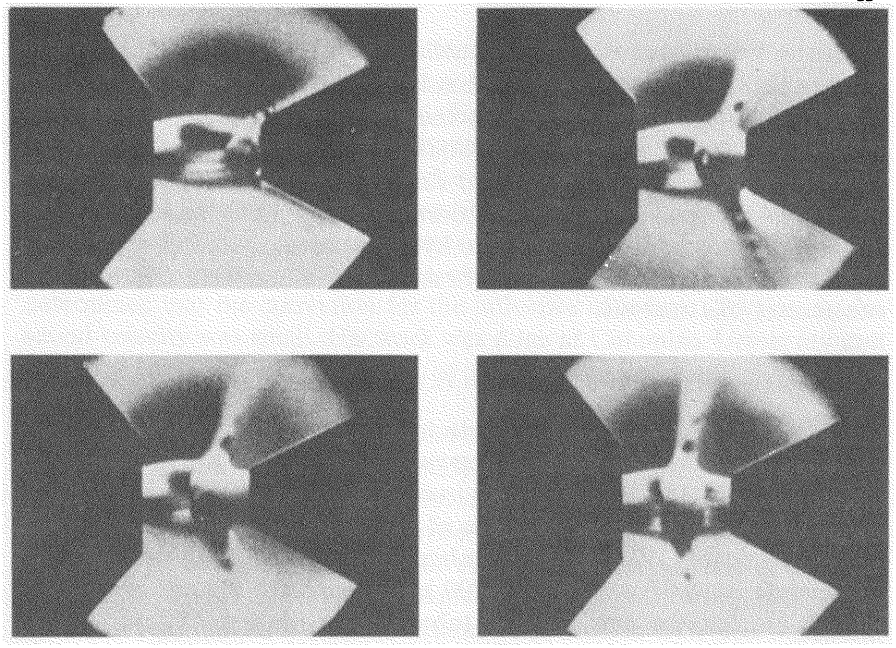


FIG. 16. Shock waves in a Hartmann generator (Bergmann, 1954).

An interesting update occurred in the matter of bats and ultrasonics in about 1920. H. Hartridge, a physiologist and specialist in acoustics, recalled the use of sound-detecting devices in World War I and proposed that bats heard reflections of their high-pitched cries and were thus able to guide themselves. Although he did not offer any experimental evidence to support the view, he clearly had in mind a hearing-based echo-location principle as the means of navigation. Actually, the matter had not been completely dormant since Spallanzani's time. A number of years earlier, in 1900, Rollinat and Trouessant had essentially repeated the 18th-century philosopher's work and concluded that the ear was more important than anything else in directing the bat. In 1908, Hahn again repeated Spallanzani's work with great care, and concluded that the sensitivity of the ear to slight condensations in the air was responsible for obstacle avoidance (Galambos, 1942; Griffin, 1958; Sales and Pye, 1974). Still, following Hartridge's 1920 reexamination of the matter, it was to be another 18 years before the next significant step occurred in understanding the method of bat navigation.

Note should also be taken of contributions in this period in areas closely related to ultrasonics. Webster (1919) extended the concept of impedance, first used by Oliver Heaviside in electric circuits, to the acoustic impedance of mechanical oscillations and waves. In the same 1919 paper, he introduced the differential equation for variable cross-

section horns. The "Webster horn equation," as it has become known, has found extensive application in ultrasonics in the design of concentrators for high-power applications. Webster was a professor of physics at Clark University and was a powerful mathematical physicist, well known for his treatises on electricity, magnetism, and dynamics, as well as for his contributions in acoustics. During the war he had been involved in the problems of underwater listening devices.

Two famous theoretical physicists, ordinarily not associated with the subject of acoustics, nevertheless made very important contributions during this period. Einstein (1920) suggested the use of sound dispersion to measure the rate of a fast gas reaction and, for the first time, developed the equations for sound propagation in a reacting mixture. The analysis incorporated the effects of molecular relaxation and was the most significant analysis in this area since Jean's 1904 formulations (Herzfeld, 1966). Brillouin (1914, 1922) published two papers on the diffusion of light by transparent bodies. According to Brillouin, the periodic variations in the density of a liquid being traversed by acoustic waves would give it a structure that could be revealed by light diffraction patterns. This concept was to be resurrected in a most significant way by Debye and Sears in the early 1930s.

V. Intense Ultrasound and Other Developments (1927–1939)

Following a decade of steady growth in underwater uses, communications, and scientific applications, the year 1927 represented a new landmark in ultrasonics with the appearance of a paper by Wood and Loomis on the effects of intense ultrasound. This effectively brought in a whole new field, power ultrasonics. Although this occurred in the U.S. 10 years after the birth of ultrasonics, the development was directly traceable to Langevin's first work.

Other new developments in the scientific area were in the optical effects of ultrasound and in understanding the physics of dispersion and absorption. A new type of transducer, based on magnetostriction, appeared. After the opening decade, an outpouring of technical literature was to occur during the period, including the first books on the subject. By 1939, ultrasonics had come of age.

A. Ultrasonics and the Loomis Laboratory

The physical effects of intense ultrasound had not gone unnoticed in the first decade of modern ultrasonics. Langevin's first tests with quartz plate and sandwich transducers had resulted in killing fish in the beam of

sound, as noted earlier in this paper. Cady (1963) notes that in 1924, Professor Van Dyke had observed the searing of skin when a strongly resonant quartz bar was touched, as well as almost explosive atomization of water drops from the end of the rod. Van Dyke also noted friction alleviation between a metal surface and the vibrating quartz. Boyle (1928) noted efforts to reproduce some of Langevin's physical effects, but with negative results. Although these and, quite possibly, other observations were made in the early years, no steps had been taken to investigate them more fully.

However, among those who had observed Langevin's work was Professor R. W. Wood (Fig. 17), a famous American physicist from Johns Hopkins University. Wood, it may be recalled, has been mentioned previously in this history in regard to his 1899 work on photographing sound waves. His major interests in a long academic career were optics and spectroscopy, with his book "Physical Optics" (Wood, 1905) a classic in

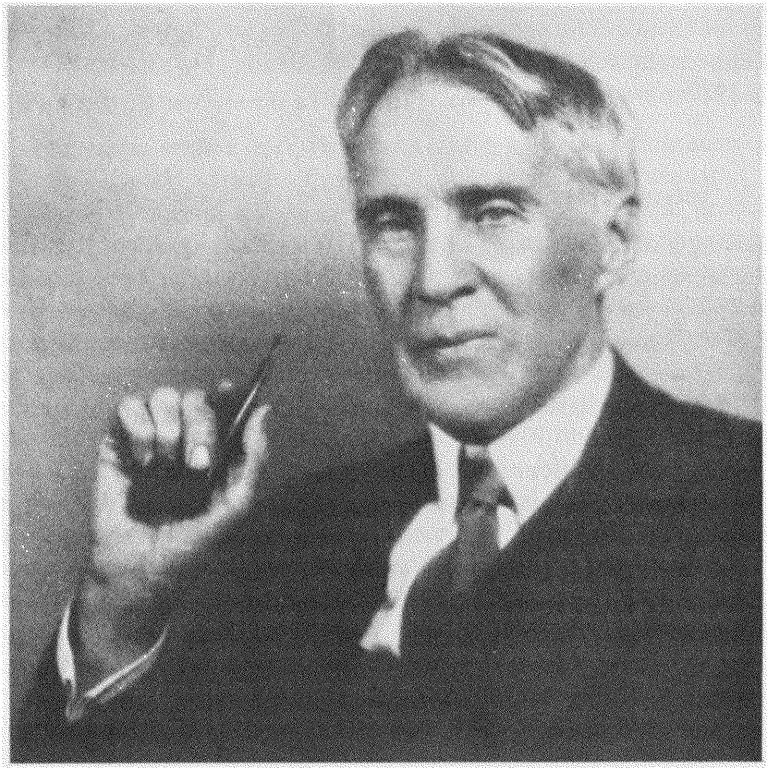


FIG. 17. Robert Williams Wood, 1868–1955 (Seabrook, 1941).

the field. In addition to his voluminous scientific writings, he was known as a brilliant lecturer and an occasional writer of popular fiction and verse (Gillespie, 1971; Seabrook, 1941).

Professor Wood's presence at Toulon, France, in 1917 deserves some explanation. As with many scientists, Wood had been asked to assist in the war effort. Shortly after the entry of the U.S. into World War I, he had participated in a meeting of the Naval Consulting Board held to interchange ideas on antisubmarine activities. In addition to suggesting the use of antitorpedo "blisters" on ships, a concept that was to come into later use, Wood suggested the use of trained seals for detection of submarines (Seabrook, 1941). This idea eventually filtered to Britain and may have been the origin of a serious study that was later described by A. B. Wood (1962). Professor Wood was also present at the June 1917 meetings in Washington with the French–British delegation where the work of Langevin was reported. Commissioned a major in the army, he was assigned to the Bureau of Inventions in Paris. Although involved with a wide range of research efforts, he asked permission to devote particular attention to the work of Langevin.

While the association with Langevin was brief, it was to make a powerful impression on Wood. Thus, he later noted (Wood, 1939)

> It was my good fortune during the war to be associated for a brief time with Prof. Langevin during his remarkable developments. At the arsenal at Toulon I witnessed many of the experiments with the high power generators. One was mounted in a large wooden tank filled with sea water, and when the Poulsen arc was started and the frequency adjusted for resonance the narrow beam of supersonic waves shot across the tank causing the formation of millions of minute air bubbles and killing small fish which occasionally swam into the beam. If the hand was held in the water near the plate an almost insupportable pain was felt, which gave one the impression that the bones were being heated.

Wood returned to the U.S. in January 1918. He briefed Professor Pupin of Columbia University on Langevin's work, including imparting details of the quartz vibrators. Wood then pursued other areas of war research, including development of signal lights, and returned to his work in optics and spectroscopy following the war. His interests in ultrasonics were to remain dormant for several years.

Another meeting that was to prove essential to the "invention" of power ultrasonics occurred during the war. During this period, Professor Wood met Alfred E. Loomis at the Aberdeen Proving Grounds. At the time, Loomis, a successful lawyer, was directing research at Aberdeen as

an Army Major. During this time, for example, he invented the "Loomis chronograph" for measuring the velocities of shells. After his World War I service, he entered investment banking, where he amassed a personal fortune during the 1920s. However, his interests returned increasingly to scientific research. About 1924, Loomis renewed the wartime acquaintance with Wood and offered to collaborate and underwrite any interesting joint research ventures. In the summer of 1926, Wood told Loomis of Langevin's experiments and suggested that the subject seemed to offer a wide field for research in physics, chemistry, and biology (N.Y.T., 1975; Seabrook, 1941).

Using a very-high-power vacuum-tube oscillator built for them by General Electric and a quartz plate transducer immersed in an oil-filled dish, experiments were started in Loomis's garage in Tuxedo Park, New York. As the scope of the work expanded, Loomis purchased a nearby mansion, and converted it into his well-known Tuxedo Park Laboratory. With the huge vacuum tubes, a large bank of oil condensers, an oversize variable condenser, and a large stepup induction coil, the resulting apparatus was a most imposing affair. Figure 18 shows a portion of the early apparatus with, presumably, a pioneering biological effects experiment in progress. Vibrations of 200–500 kHz were transmitted to the oil bath and from there into glass vessels or rods immersed in the bath, achieving a range of spectacular effects. Among these were (Wood and Loomis, 1927)

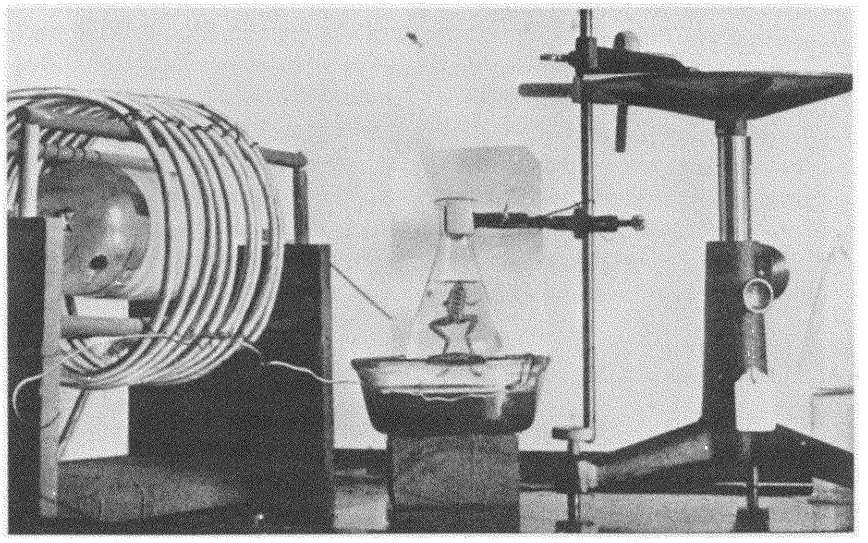


FIG. 18. Apparatus for early experiments with intense ultrasound (Wood and Loomis, 1927).

(a) radiation pressures of sufficient magnitude to support 150 g and to raise a pronounced mound of oil above the transducer;

(b) the intense searing of the skin when the vibrating glass rod was firmly grasped;

(c) the smoking and burning of wood chips pressed against the tips of glass rods;

(d) the etching and drilling through glass pressed against vibrating glass rods;

(e) the internal heating of liquids and solids;

(f) the formations of emulsions and fogs;

(g) the flocculation of particles suspended in a liquid;

(h) numerous observations of biological effects, including rupturing of red blood cells, killing of unicellular organisms, and harmful to lethal effects on small fish, frogs, and mice;

(i) preliminary observations on chemical reactions and crystallization.

This summary of pioneer results could almost be taken as a present-day litany of ultrasonic accomplishments. Additionally, Wood and Loomis made a number of observations of the modal patterns of rods, tubes, and plates and, in the process, gave some of the first experimental data on phase velocity in rods and disks. Another first was recorded when they made an ultrasonic horn by drawing down a glass tube to a tapered point for purposes of concentrating the energy at the point of application.

The publication of these results was to start avenues of work being exploited to the present day. Interestingly, Professor Wood did not continue with this line of investigation, but returned to optics and spectroscopy. He was to publish a short monograph in 1939 recounting the 1927 work and certain other aspects of ultrasonics. Nevertheless, in a scientific career spanning a half century and over 250 publications, this was to represent his involvement with "supersonics." Loomis, however, was to be involved with several other phases of ultrasonics.

The 1927 publication had made brief note of several areas that were under in-depth study by other collaborators of Loomis. Dr. W. T. Richard of Princeton University was investigating the chemical effects of intense ultrasound, and these results were pursued further by E. N. Harvey, with two publications by Harvey and Loomis appearing in 1928. In another facet touched on in the original study, an important new tool for physical acoustics was to emerge. This was essentially to be the liquid analog of the Pierce interferometer.

Except for Wood, each of these other collaborators of Loomis's was to

push on into further productive, independent research in ultrasonics. Loomis himself had a continuing interest in precision time measurement. For example, he collaborated with developers of a quartz-oscillator-stabilized electric clock to obtain precise measurements of the best pendulum clocks. Backed by a private fortune, he was able to be a 20th-century patron of the sciences. In the years ahead, Loomis was to play an important role in founding other major research laboratories and in directing World War II research (N.Y.T., 1975).

B. Appearance of Magnetostrictors

The magnetostrictive effect, although discovered by Joule nearly 40 years before the Curies's discovery of piezoelectricity, has not yet played a role in the field of ultrasonics. It was Pierce, whose uses of piezoelectric crystals in frequency control and the acoustic interferometer have already been noted, who was the first to develop a magnetostrictive oscillator. Pierce apparently took it as a personal challenge in 1926 when he noted a published statement that magnetostriction had no useful applications. Using his background on crystal oscillators as a starting point, he soon devised a self-excited magnetostrictive oscillator (Saunders and Hunt, 1959).

He published an extensive account of such oscillators (Pierce, 1928), noting they would be particularly useful in the lower frequency ranges (less than 25 kHz), where crystal oscillators were impractical. Pierce's work explored the properties of a number of magnetostrictive materials, considered their use in control of oscillators, as time standards, and as a sound source for acoustic interferometry. Vincent (1929), working independently, also devised a magnetostriction oscillator (Neppiras, 1968). In the years ahead, many others extended the work on the scientific and communications uses of this second source of continuous high-frequency waves.

The first use of magnetostrictors as a source of powerful ultrasound was by Newton Gaines at the University of Texas. Stimulated by the earlier work of Wood and Loomis and of Harvey and Loomis, Williams and Gaines (1930) described the effects of high-frequency sound waves on bacteria. A nickel-tube magnetostrictor oscillator, based on Pierce's designs, and having an 8800-Hz natural frequency was used. A much more extensive study was later published by Gaines (1932), using improved apparatus developed at Texas Christian University. Again, the nickel-tube oscillator, vibrating at the painfully (as noted by Gaines) audible 8900 Hz was used. Many of the effects noted by Wood and Loomis were again spectacularly reproduced, such as mounding of liquid above the transducer, the fountain effect, the pumping action, the production of dispersions, emulsions, flocculations and fogs, fracture of glass, heating effects

and biological actions. In addition, very pronounced cavitation erosion patterns were produced on metal and glass surfaces. Further, several times Gaines fatigue-fractured the nickel tubes when they were freely vibrating in air, and suggested this as a possible method for fatigue-testing certain metals.

Pierce's development of magnetostrictive oscillators was also quickly noted by his friend Dr. H. C. Hayes of the Naval Research Laboratory. One of Pierce's students, E. B. Dallin, was brought to the Naval Research Laboratory in 1927 to initiate studies of such oscillators. Out of this came several developments in depth finders and detection apparatus, as well as audible signaling horns (Klein, 1967).

C. Optical Aspects of Ultrasonics

A major facet of ultrasonics that was to emerge in the second decade, beyond that of high-intensity uses, was interest in various acousto-optic phenomena and methods of visualizing sound waves. Both scientific and industrial uses were to emerge from this area. The landmark development in this period was the discovery of the diffraction of light by ultrasonic waves. The fact that Brillouin had suggested this effect in 1922 has been noted earlier. However, his work was associated with the scattering of light by ultra-high acoustic frequencies (now called the pretersonic range). Professor Peter Debye felt that light scattering by sound waves of much lower frequencies would occur and, while on a lecture tour in the U.S., suggested to Francis W. Sears the experiments to demonstrate this. Although the initial experiments suggested by Debye, involving ultrasonic waves normally incident on a liquid surface, were unsuccessful, it was soon found that waves moving parallel to the liquid surface produced a strong diffraction pattern (Lindsay, 1974). Debye and Sears (1932) published their result, and in the same year, Lucas and Biquard (1932) independently published results showing the diffraction of light. Figure 19 illustrates the

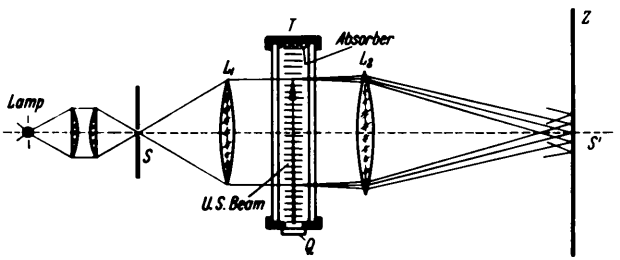


FIG. 19. Optical arrangement for observing light diffraction by ultrasound (Barone, 1962).

arrangement for observing diffraction. An interesting additional aspect of the latter work was the determination of the curvature of the light rays due to the variable refractive index of the liquid, and the fact that this precise behavior had been shown years earlier by R. W. Wood in his classical book on physical optics (Wood, 1939).

The appearance of the Debye–Sears and Lucas–Biquard papers was to spawn a host of investigations on light diffraction by ultrasound. Investigators such as Bergmann, Hiedemann, Bär, and Meyer, who were to contribute to ultrasonics for many years, made their early contributions in diffraction related problems. Mention should be made of modifications made by Bär and Meyer (1933) that permitted whole field diffraction patterns to be obtained and thus allowed sound wave interaction with obstacles to be visualized. Complex diffraction patterns from interacting sound waves in liquids were observed by several of the preceding workers almost simultaneously by 1933, while Schaefer and Bergmann (1934) extended their concepts to observing patterns from wave interactions in transparent solids (Bergmann, 1939).

Although a great many experimental studies followed the Debye–Sears and Lucas–Biquard works, the next most significant contribution was analytical in nature. While an analytical base had been provided for the initial diffraction observations, it did not account for all the observed effects, which included higher-order diffraction images and diffraction spectra of unexpected intensity distribution. However, using Rayleigh's theory of the diffraction of a wave by a corrugated surface, C. V. Raman and N. S. Nagendra Nath (1935, 1936) succeeded in providing a theory of diffraction that cleared up most of these problems (Barone, 1962).

Other aspects of acousto-optic developments in this period build on 19th-century discoveries on visualizing sound waves. The striation (or Schlieren) method used by Toepler, Mach, and Wood in the 1800s was revived by E. P. Tawil (1930) to observe the waves from a quartz oscillator. Even earlier than this, Tawil (1926) had revived the polarized-light method of Biot and Tyndall to study the strains in a quartz bar and foreseen uses of optical effects for light relays and television. The striation study prompted a number of follow-on investigations, with several improvements in procedure occurring. Pohlman, who also was to have a long and prolific career in ultrasonics, made his first contribution in 1935 with a modified striation system. Bergmann also contributed here (Bergmann, 1939).

The use of sound waves to control the intensity of light became of interest as soon as the various acousto-optic effects were observed. This led to a number of nearly simultaneous developments in light modulators, stroboscopes, and light relays. In 1932, Biquard and Karolus described a light relay using the diffraction effects that had just been discovered. In the period 1935–1936, a number of developments occurred in this area, all

based on using the diffraction effect. Acoustic waves in both solids and liquids were used as the diffracting medium in the work of Becker, Bär, Hiedemann, Jeffree, and others (Bergmann, 1939; Cady, 1946).

The use of sound waves to present visual images is a long-standing desire in the world of acoustics. In the 1930s, it was but a step to progress from the concepts of light modulation to ultrasonic television, and this was done by Jeffree in 1934 and 1936. The Scophony Supersonic Television System that emerged used the ultrasonic diffraction cell and other apparatus to project an image on a screen. Thus, it only acted as a television receiver. It was found possible to project very large images on motion-picture screens with this pioneering system (Bergmann, 1954; Cady, 1946).

Obtaining an image of an object "illuminated" by sound waves is the other phase of using acoustics to present visual images. In contrast to ultrasonic television, which did not survive, work on ultrasonic image tubes that started in the 1930s is still being pursued. In 1937, S. Ya. Sokolov noted that properly mounted and damped piezoelectric plates could be used to detect the point-by-point intensity of an incident sound field. The developments stemming from this were to lead to successful image tubes (Woloshyn, 1963). Pohlman (1939a) developed an image cell that used a suspension of aluminum dust in xylene to detect acoustic waves. The small platelets of aluminum acted as minute Rayleigh disks, orienting themselves in accord with the intensity of acoustic pressure over the face of the cell and thus presenting an image. This device was to find practical use in German industry in World War II in nondestructive testing applications.

In general, the field of acousto-optics flourished during the 1930s. However, many of the effects could not be exploited with the optical sources of that period, so that by the end of the decade, interest in the field had waned. A strong revival was to occur in the 1960s and 1970s with the invention of the laser (Stegeman, 1976).

D. Progress in Physical Acoustics, Underwater Sound, and Bats

The scientific uses of ultrasound for investigating the properties of materials had started several years before the high-intensity uses were discovered. Consequently, by the 1930s a voluminous literature was appearing in this field alone. Hence, only a few notes will be made of the major developments in the field of physical acoustics. In addition to the developments described below, a number of the previously described studies on the optical aspects of ultrasound were used for determining velocity and absorption characteristics.

Pierce's acoustic interferometer, developed in 1925, proved to be a very popular instrument for measuring velocity and absorption of sound in gases. His observations on absorption and velocity dispersion in gases

prompted many follow-on studies, such as those of Abello (1927), Pielemeier (1929), and Grossman (1932). Richards and Reid (1934) presented data on velocity and absorption over a full range of frequencies. During the 1930s, the interferometer was developed for testing over a wide range of temperatures and pressures, and the theory of the instrument was explored (Herzfeld, 1966; Hubbard, 1940; Richards, 1939).

The development of acoustic theories for explaining the anomalous absorption and velocity observations made with precision ultrasonic apparatus represented the most significant contributions of the 1927–1939 period. Possibly the landmark contribution was made by Herzfeld and Rice (1928) while at Johns Hopkins University. Interested in the general problem of energy supply to molecules during chemical reactions, they developed the first theoretical expressions for dispersion and absorption of sound in gases. As noted by Herzfeld (1966), this work was done without knowledge of Rayleigh's earlier suggestions in this area, or of Jean's and Einstein's work. The work of Herzfeld and Rice, which was more or less a continuum gas theory, was immediately followed by a comprehensive statistical theory put forth by D. G. Bourgin (Richards, 1939). Then Kneser (1931) used the Einstein theory and developed the dispersion-theory equations for the entire frequency range (Lindsay, 1974).

Studies of ultrasonic dispersion and absorption in liquids started later than in gases and were triggered by the already-noted development by Hubbard and Loomis (1927) of a liquid acoustic interferometer. In their paper, they reported results on the speed of sound in water, mercury, solutions of various salts, and some organic materials. The first ultrasonic absorption studies in liquids were done by Biquard (1931). Knesser (1938) published the first theory on excess absorption in liquids, suggesting the same operative mechanisms as in gases (Herzfeld, 1966).

The use of ultrasonics to establish physical theories of solids did not progress to the extent of gases and liquids in the 1930s. The acousto-optic effects already noted were used to establish elastic constants of transparent solids (Bergmann, 1939). A number of studies that were mainly extensions of the work initiated by Quimby in 1925 were carried out. These involved using composite piezoelectric resonators to determine the elastic moduli of materials. Thus, Balamuth (1934), then at Columbia University, embarked on a long career in ultrasonics by modifying Quimby's resonator configuration so that the cement joint between the driving quartz rod and the specimen was at a stress node of the resonator.

An interesting note to physical acoustics is provided by a contribution of Langevin to this subject. Truesdell (1953) reminds us that the original Kirchhoff equations for a perfect gas including heat conduction and viscosity were later shown by Langevin to be valid for pure fluids. These

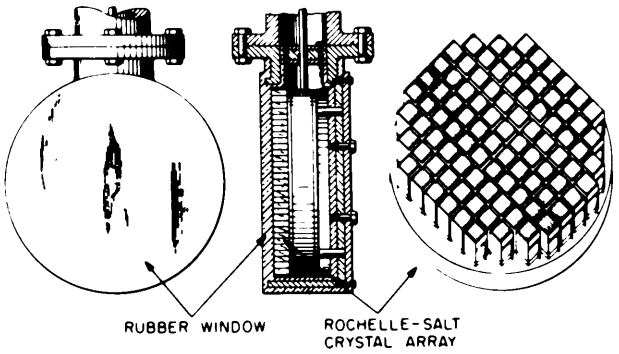


FIG. 20. Rochelle-salt crystal array and sound transparent window (Klein, 1967).

results were presented in Langevin's lectures and were published by Biquard in 1936.

In another important part of ultrasonics, that of underwater sound, progress was mainly in military uses for submarine detection. In the U.S., most of the work was done at, or under contract from, the Naval Research Laboratory (NRL). It was previously noted that a significant effort was made to improve the Langevin-type quartz sandwich transducer in the late 1920s. The Washington Navy Yard manufactured the quartz–steel echo-ranging equipment, producing about 20 sets by 1933 (Howeth, 1963). From about 1927 forward, a major effort was made to develop Rochelle-salt transducers. Despite certain disadvantages of the material, such as fragility and water solubility, the fact that it possessed a high piezoelectric activity and could be readily produced justified serious development. An important associated development in 1930 was the creation of "Rho–C" rubber by the Goodrich Company. This permitted construction of sound-transparent windows for underwater transducers (Fig. 20). By 1933, commercial production of echo-ranging equipment for the U.S. fleet was started at the Submarine Signal Company. This equipment, known as QB gear, was based on Rochelle-salt transducers (Fay, 1963; Klein, 1967).

It was noted that NRL undertook development of magnetostriction transducers in 1927, almost concurrently with Pierce's introduction of this type of ultrasound source. Work was at a low priority for several years until about 1932, when the pace of work was picked up and a variety of magnetostrictive materials and transducer configurations were produced. Starting in 1934, magnetostrictive-based QC gear was being installed on destroyers and submarines. Some research was done on magnetostrictive generators for foghorns and for altitude-drift meters for dirigibles. The foghorns proved impractical, and the demise of the airship ended the altimeter, however (Klein, 1967).

In 1940, after the outbreak of World War II, information exchange on antisubmarine warfare developments was renewed between the Americans and British. Although a gap of nearly 20 years had occurred since collaboration had stopped, it was found that the two nations had paralleled developments very closely and possessed similar fleet equipment. The main technical differences were that the British had continued use of the quartz–steel sandwich transducer, whereas the American units were nearly all magnetostrictive by that time. The British were far ahead in equipping their ships with ASDICs (termed by Winston Churchill as "the system for groping for submarines") having 220 ships fitted versus 60 American vessels (Hezlet, 1975; Lasky, 1974b).

The British had discontinued use of passive underwater sound detectors. The Germans, it was found, had elected not to pursue extensive development of echo-ranging techniques, but instead had concentrated on development of sophisticated passive listening devices. The German GHG (Gruppen Horch Gerät) system consisted of an array of Rochelle-salt hydrophones (up to 24 in number) mounted at the bow of surface ships and submarines. Thus, in the submarine warfare of World War II, the active "groping" of the American–British systems was pitted against the sophisticated listening system of the Germans (Hezlet, 1975; Lasky, 1974b).

While most of the effort in underwater sound between the wars was devoted to equipment development, some progress was made on sound measurements and propagation characteristics. Thus, G. W. Pierce discovered in some of his researches the "coincidence effect" of an immersed plate, whereby a plate could be effectively transparent to an incident sound wave at critical frequencies and angles (Saunders and Hunt, 1959). At NRL, a special apparatus, using tourmaline crystals, for hydrophone calibration was developed. Most noteworthy were underwater sound investigations in Guantanamo Bay in 1937–1938, where it was established that the "afternoon effect," first noted by Langevin many years before, was due to sound velocity gradients caused by sharp thermal gradients near the water surface. This advance notwithstanding, the overall understanding of the sea as an acoustic transmission medium was rather meager by the outbreak of World War II (Klein, 1967).

Finally, the matter of bats and ultrasonics in nature was reopened in the 1930s along lines leading ultimately to the solution of "Spallanzani's bat problem." Hartridge, it will be recalled, had suggested in 1920 that bats used high-frequency sounds to navigate. In 1932, the Dutch zoologist Sven Dijkgraaf recognized the auditory basis of obstacle avoidance by bats. He had made a study of the faint, audible clicks ("Ticklaut") from a flying bat and noted how closely they were correlated with the echolocation of obstacles (Griffin, 1958).

However, it was to be the encounter of a Harvard undergraduate and an eminent pioneer in ultrasonics that was to bring the problem nearly to solution. While an undergraduate, Donald R. Griffin was engaged in banding bats to study their migratory habits. Encouraged to experiment with bat obstacle avoidance, one of Griffin's friends called his attention to the presence in the physics department of an expert in "supersonics," Professor G. W. Pierce. Pierce's involvement in yet another phase of ultrasonics is worth special note, since it was to culminate contributions that had started in 1912 with observations on motional impedance. About 1934, he became interested in the matter of insect sounds, particularly in the supersonic range. Using stroboscopic motion pictures and special ultrasonic receivers and circuits for translating high-frequency sounds into the audible part of the spectrum, Pierce's studies were to extend over many years, leading to his last book on "The Songs of Insects," (Pierce, 1948). Proceeding from simply holding a cage full of bats in front of a microphone to more controlled tests, Pierce and Griffin conclusively showed that bats emitted ultrasonic sounds of up to 48,000 Hz. However, because of the highly directional nature of the microphone they used, as well as the bats sounds, they did not record a continuous emission of cries during flight. Consequently, their 1938 paper still did not connect the ultrasonic pulses with obstacle avoidance, but rather with a warning or call note (Griffin, 1958; Pierce and Griffin, 1938).

E. First Uses in Testing and Medicine

At a time when the scientific and high intensity uses of ultrasound were being vigorously exploited, the first uses appeared in two new areas, testing and medicine. In the former area, S. Ya. Sokolov of the Soviet Union and O. Mühlhäuser of Germany share credit for applying ultrasound to the testing of materials. Sokolov (1929) first proposed the use of ultrasound to find hidden flaws in metals using a through transmission technique (Bergmann, 1954). Sokolov had started working with the piezoelectric properties of quartz soon after starting his teaching career at the Leningrad Electro-technical Institute. By 1929, he headed an acoustics laboratory at the Institute, was investigating a wide range of uses of ultrasonics, and was a major contributor to the field for many years. Many of Sokolov's concepts in ultrasonics, which were to include microscopy and imaging systems, were far in advance of electronic and optical technology of the 1930s and were not to achieve fruition for many years (White, 1976).

Mühlhäuser (1931) had the first actual patent in the area of ultrasonic nondestructive testing. Both his and Sokolov's methods were based on continuous ultrasonic waves, and the concept that flaws in a material

would screen some of the energy from a receiving transducer. Others who were to contribute in this area of ultrasonics in the 1930s included Shraiber (1939), Kruse (1938, 1939), and Giacomini and Giacomini (1939) (Carlin, 1949; McMaster and Wenk, 1951).

Sokolov (1935a) then went on to devise an innovative system for flaw detection that used acousto-optic effects. This system was based on transmitting waves through the test piece into an illuminated liquid medium. A diffraction pattern was formed, based on the Debye–Sears effect, with differences in the pattern being used to distinguish flawed and unflawed test pieces. In a further simplification of the system, Sokolov (1935b) dispensed with the diffraction approach and used a light beam reflected from the surface of a liquid insonified by waves transmitted through the test specimen (Carlin, 1949). This was the original concept of the liquid-surface-levitation converter, and may actually have been proposed by Sokolov as early as 1929 (Gericke and Grubinskas, 1969).

In yet another pioneering contribution, Sokolov in 1937 devised an ultrasonic image tube based on the piezoelectric effect. He found that piezoelectric crystals, when damped on one side, would not resonate as a whole when excited at the fundamental frequency, but would resonate point-by-point in accord with the incident sound energy (Woloshyn, 1963). His patent (Sokolov, 1939), Fig. 21, was the first granted in the U.S. for this type of testing (Carlin, 1949). At about the same time, Sokolov (1935b) conducted a very extensive investigation of the properties of the Langevin

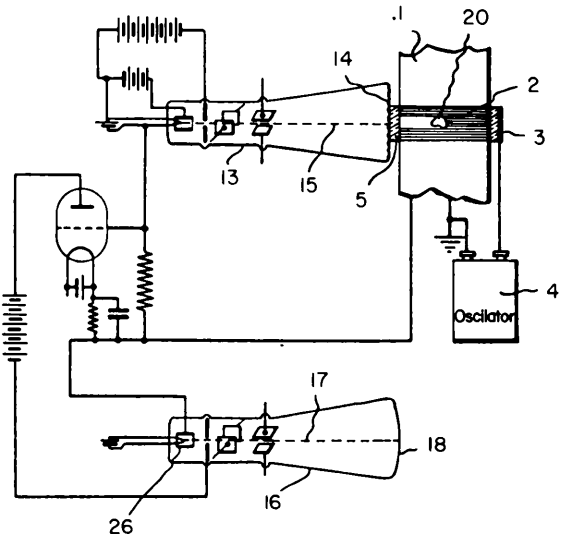


FIG. 21. The Sokolov patent for ultrasonic testing (Sokolov, 1939).

sandwich transducer. This included determining the natural frequencies of various configurations from 10 kHz to 130 Mhz and the use of a piezoelectric pickup for mapping the distribution of vibration amplitude on the surface of the transducer.

In addition to testing, the first developments in the new application area of medicine occurred during 1927–1939. The effects of intense ultrasound on living organisms were among the early observations of Langevin and were extended in the work of Wood and Loomis in 1927. As will be noted in Section V.F, a large number of studies were to be made in the 1930s on the biological effects of ultrasound. Actual medical uses were far less extensive, however.

According to Herrick and Krusen (1954), the first suggestions for the medical uses of ultrasound appeared in the German literature in 1928. These were in reference to means of transferring ultrasonic vibrations to the skin and for a particular application in otology. Somewhat later, Freundlich and co-workers made the first specific recommendation for the use of ultrasound in medicine for diathermy. In their work, they were able to show that bone marrow could be heated without the bone itself being affected (Herrick and Krusen, 1954). Several other studies in this area appeared in Germany and Japan in the period 1933–1935 on therapeutic uses (Bergmann, 1939; Richardson, 1957). The most extensive investigation on such uses was made by Pohlman (1939b). Nearly all of the suggested uses were based on the postulate that ultrasound would be effective in heating tissue. The "almost insupportable pain" that Wood and others had experienced when they placed their hands in the beam of Langevin's radiator was due to the heating of the tissue covering the bone.

F. Development of Power Ultrasonics

The pioneering paper of Wood and Loomis (1927) on the effects of intense ultrasound appeared in 1927 and was accompanied by a companion paper by Richards and Loomis (1927) specializing on the chemical aspects. It is of interest to follow the first few years of this infant technology and note the flow of contributions. Thus, in 1928, Harvey *et al.* (1928) and Harvey and Loomis (1928) published further on the biological effects, as did Schmidt *et al.* (1928). The contributions expanded in 1929 with Harvey (1929) investigating the effects on muscle tissue; Harvey and Loomis (1929), Johnson (1929), and Schmidt (1929) publishing on the biological effects; and Richards (1929) and Schmidt *et al.* (1929) investigating the chemical effects. Hopwood (1929) in England surveyed a range of effects. Biological effects were further studied in 1930 by Harvey (1930), Schmidt and Uhlemeyer (1930), and Williams and Gaines (1930). A new

area of treatment of molten metals was reported on by Mahoux (1930) in that year.

Most of the early contributions, quite naturally, were from the Richard, Harvey, and Loomis collaborations and oriented toward the chemical and biological effects. The Schmidt, Johnson, and Olson collaboration originated from the University of California and was apparently stimulated directly by Richard and Loomis's 1927 paper. Hopwood's 1929 paper, as well as a later paper (Hopwood, 1931) described work done in England and were general in nature, reviewing many of the phenomena shown by Boyle and by Wood and Loomis. Apparently Hopwood was independently pursuing studies of certain biological effects when the work of Wood and Loomis appeared. Williams and Gaines's 1930 publication has already been noted as the appearance of the first power use of magnetostrictive oscillators. It heralded Gaines's 1932 in-depth study, also previously noted, and a number of other papers by his collaborators. The 1930 publication by Mahoux in France was an exception to all the other pioneer works, being applied to the area of metallurgy. Specifically, it was applied to speeding up of nitrogen hardening of steels. Nearly all of the applications in this opening period and all through the 1930s were on liquid phase processes, since the means of coupling ultrasonic energy into liquids was well understood.

Following this brief opening period, the number of publications and the range of effects studied increased steadily. At least 150 publications related to power ultrasonics can be identified for the period 1927–1939. The main general areas covered were outgrowths of the Wood and Loomis work and were emulsification and dispersion, the coagulating action, fogging (or atomizing), and the chemical and biological effects (Bergmann, 1939; Richards, 1939). In the following, only the major advances can be noted.

Following the Wood and Loomis study, numerous studies on emulsification and dispersion effects were carried out throughout the 1930s. The early oil–water and mercury–water emulsion results were soon extended to dispersions of numerous metals and fusible alloys in alcohol, oil, and water. Some of these in-depth studies had appeared by 1933. The influence of intensity, frequency, and entrained gases were studied. The preparation of photographic emulsions and liquification of gels were investigated (Bergmann, 1939).

The emulsifying–dispersing action of ultrasonics led to several studies in metallurgy. Apparently, proposals for degassing metals had appeared in 1931 and 1934, but were not actually tested. Related to this was grain refinement, and here O. Mühlhäuser, who had the first ultrasonic testing patent, made the first contribution in 1931. Forced by necessity to use

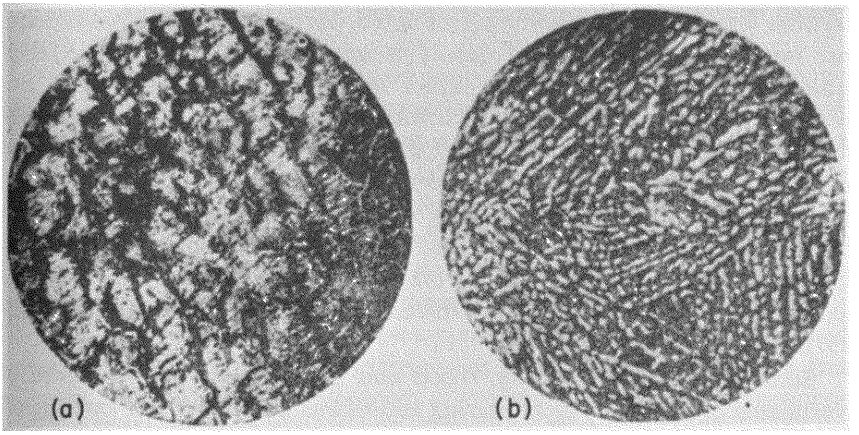


FIG. 22. Influence of ultrasonics on structure of cast zinc. Untreated (a) and ultrasonically treated (b) (Bergmann, 1939).

crude apparatus, his high-frequency generator was the toothed blade of a circular saw, the same principle as the 19th-century Savart apparatus (Hiedemann, 1954). Sokolov also made early contributions to this area in 1933 (Fig. 22). Another area was the production of alloys of certain metal combinations that were not ordinarily miscible in the liquid state. These studies were in progress by 1936. Yet other aspects were associated with dispersion of metals during electrolysis, in progress by 1935, and with electrodeposition of metals, in progress by 1936 (Bergmann, 1939; Richardson, 1957).

The coagulating action of ultrasonics had been noted by Wood and Loomis in 1927, although the early observations of the phenomenon dated back to the first Kundt-tube experiments. Consequently, some of the underlying mechanisms had been studied as early as 1867 by Bjerknes and 1891 by Koenig. Following Wood and Loomis, the process was subjected to a number of detailed studies starting in 1934. The almost instant agglomeration of aerosols was shown, and detailed studies of the motion of droplets in an acoustic field were made. Theories of coagulation were put forth, with it being shown that there were optimum particle sizes for given frequencies (Crawford, 1955; Richardson, 1957).

Following the Wood and Loomis and the Richard and Loomis 1927 studies, investigations of the chemical and biological effects were quickly initiated by a number of investigators. The almost immediate appearance of work by Schmidt, Johnson, and Olsen in 1928, 1929, and thereafter has already been noted. In fact, the number and diversity of studies makes it difficult to summarize. In the case of chemical effects, the acceleration of

reactions was the most comm effect noted, particularly where oxidations were involved. Hydrolysis a depolymerization were among the effects studied. The importance of avitation in the processes was noted by several investigators, and soe difficulty was encountered in separating the thermal influences from urely" ultrasonic effects on chemical reactions. In the case of biolgal effects, studies ranged from action on microorganisms to the lethaeffects on fish, frogs, and small animals. Some practical uses in milhomogenization were to emerge from this work. Related to the biolcal work were some of the first uses in medicine, which have alreadeen discussed (Richards, 1939; Richardson, 1957).

Some effects uncoveredy Wood and Loomis were not extensively investigated or applied. The mizing action was noted by other investigators in 1934, 1936, and probly afterward, but does not appear to have been pursued in any deptl The various spectacular actions noted in burning wood and drilling gss apparently remained as curiosities during this period. The interesting oding action of ultrasound noted by Gaines also was not apparently pursed further.

The study of cavitationceived a good deal of attention during this period, since the importancef this phenomenon in a number of ultrasonic effects was soon recognized t has earlier been noted that Boyle and his co-workers carried out somef the early studies of cavitation in the 1920s. Boyle gave the first serious view of the subject in 1928 and made other contributions in the late 19s. The suppression of cavitation in degassed liquids, or in liquids under hrostatic pressure, was soon noted. Frequency dependence of cavitation waalso noted as well as the cavitation threshold for specific conditions (Bemann, 1939). The faintly luminescent field accompanying cavitation s first noted in 1934 and called "sonoluminescence" (Flynn, 1964

While a number of expments involving cavitation and luminescence were carried out, a basic unrstanding of the phenomena was not reached during this period. In a rew of ultrasonics, Richards (1938) noted, "I have been told by every mathematical physicist I know that the analysis of cavitation is a task beyon the ability of present day mathematics." Fortunately, the task had ot been beyond the ability of earlier-day mathematicians, for, as h already been noted, Lord Rayleigh had analyzed the all-important oblem of bubble collapse in 1919. However, as has occurred before in austics, scientists were to discover cavitation before they were to discoveRayleigh. Actually, Bondy and Söllner (1935) were aware of Rayleigh's edictions of pressure, but the ramifications were not appreciated at t time. Likewise, while several theories of sono-luminescence were p forth, a satisfactory explanation was not

obtained. A number of years were to pass before the theoretical aspects were to be properly developed.

Advances were made in other areas of ultrasonics related to power uses. Sokolov's 1935 extensive investigations of the Langevin sandwich transducer have been noted. A disk-shaped quartz focusing radiator was constructed by Gruetzmacher in 1935. Several publications appeared by Hartmann in the 1930s advancing the theory of his gas current generator (Bergmann, 1939). Except for some preliminary work in 1937 on smoke precipitation, it does not appear that the Hartmann generator realized much use in ultrasonic processing during these years (Richardson, 1957).

In closing this section on developments, some effort should be made at assessing the status of power ultrasonics by 1939. That there was a great burst of enthusiasm for development of "supersonic" processes appears not to be in question. In his review of ultrasonics, Richards (1938) noted that several installations were regularly engaged in a wide variety of work and that, "In fact, about 1932 there was a feeling in the air that anyone who manufactured anything, with the possible exception of horn buttons, was either installing a supersonic outfit or wishing that he had the money for one. The chief beneficiaries of this movement were the electric power companies." Despite this enthusiasm, which appears to have a 15–20 year cycle, it can be said that the field was still mainly one of research and development by 1939. While apparently a number of dairies were using sonic treatment in preparation of milk, few other commercial uses yet existed. A review of the publications of the period does show a number of pieces of commercial equipment, but intended for research uses. The writings of the period are replete with traditional research and development phrases such as "whether such processes can have commercial value is not yet established" or "the ultrasonic method of ... *should* very soon acquire importance for the manufacture of ...." In another perceptive comment, Richards noted that the field was "in a somewhat insecure position because no compelling reason has been found for many of the phenomena which occur." Several more years were to pass before significant practical uses were to be established.

G. Technical Literature

Following a slow expansion of the technical literature on ultrasonics in its first decade, the period 1927–1939 witnessed an explosion of work in all areas of the field. The approximately 150 publications in the power area were, in fact, a small part of the overall output. Several reviews of the field occurred during this time, the most noteworthy being those of Boyle (1928), Hubbard (1932, 1940), and Richards (1938, 1939). Richards's paper

listed 348 references. The first books on the subject of ultrasonics also appeared. Dr. P. Vigoureux (1939) published a monograph on "Quartz Resonators and Oscillators" (this was revised in 1939); Dognon and Biancani published "Ultra-Sons et Biologie" in France in 1937 (Bergmann, 1939). A short monograph by Hiedemann (1935) was followed by an extensive treatment in book form (Hiedemann, 1939). Wood's (1939) monograph, "Supersonics—The Science of Inaudible Sounds" appeared at the same time. Wood's book was based on a series of lectures given at Brown University in 1937. Other short monographs included those of Marinesco (1937) and Rytov (1938). The most important book to appear was by Dr. Ludwig Bergmann, entitled "Der Ultraschall und seine Anwendung in Wisseneschaft und Technik," and appeared in 1937. An English translation with additional references was published in 1939 (Bergmann, 1939). The latter edition contained over 550 references to ultrasonics works, most of them appearing in the 1930s.

The period saw continuing contributions by pioneers of ultrasonics, such as Boyle, Cady, and Pierce, and the emergence of a second generation of researchers such as Bergmann, Hiedemann, Hubbard, Richards, Sokolov, and many others. The first contributions by people who were significantly to influence the following decades of ultrasonics, such as Mason, Pohlman, and Balamuth, occurred in the 1930s.

VI. Ultrasonics Comes of Age (1940–1955)

The years of World War II saw uneven progress in the development of ultrasonics. Sonar, born too late for World War I, was perfected and played a major role in antisubmarine warfare. Millions of piezoelectric crystals were used for crystal frequency control and filtering. Major new developments occurred in ultrasonic nondestructive testing, and the ultrasonic delay line was introduced for radar applications. The development of power ultrasonic processes, on the other hand, was somewhat retarded.

In the postwar years, the introduction of the piezoelectric ceramics, with their many superior properties, was to provide the foundation for ultrasonics as major industry. New applications in measurement and control were found, as were uses in medicine. There was a burst of activity in power ultrasonics, with the result that some processes, such as cleaning and machining advanced from laboratory phenomena to commercial development. Rapidly improving technology in the field of electronics was also of benefit to all fields of ultrasonics. By 1955, all the broad areas opened up by Wood and Loomis in 1927, as well as some new ones, were under development. Major advances were made in understanding underwater sound propagation, both during and after the war, while the field of

physical acoustics, exploiting many of the advances in ultrasonic technology, continued to expand. A greater appreciation of the role of ultrasound in nature was also achieved.

A. Transducer Materials and Ultrasonic Generators

The years 1940–1955 were a time of discovery and change in the means of generating ultrasonic waves. A new class of man-made piezoelectric materials was to arise to displace the long-established quartz and Rochelle salt from many applications, to be in turn displaced by yet newer man-made piezoelectrics. The piezoelectrics, in a further turn, were to be effectively challenged by magnetostrictors in several areas. The first ultrasonic horns were to appear. Even the field of fluid generators, long stagnate after Galton's ancient whistle and Hartmann's newer but relatively unused generator, was to see several new devices emerge.

Just after the war, Cady (1946) published his two-volume work "Piezoelectricity." This was not only a monument to a quarter century of personal involvement in expanding the field of ultrasonics, it effectively chronicled the growth of a field from infancy to maturity. In hindsight, it can also be seen as closing an era in the "old" piezoelectrics, with emphasis on the traditional materials of ultrasonics, quartz, and Rochelle salt. At about the same time, in 1945, the first standards on piezoelectric crystals were issued by the Institute of Radio Engineers.

Starting in the early 1940s, an intense search was being conducted for man-made piezoelectric crystals. This was in part motivated by the looming scarcity of natural, high-quality quartz brought on by the enormous war demand. This led to the development of several new materials, the most significant being ammonium and potassium dihydrogen phosphate (ADP and KDP), ethylene diamine tartrate (EDT), dipotassium tartrate (DKT), and lithium sulfate. Some of these materials, such as ADP, KDP, and certain arsenates, had actually been produced in the mid-1930s in Zurich by G. Busch and P. Scherrer (Mason, 1971). The crystal ADP, Cady (1946) was to note, had a meteoric career. For most purposes, it soon supplanted quartz and Rochelle salt, except for frequency control or where the strongest possible piezoelectric effect was needed. It was widely used in sonar systems during the war, for example.

However, this new material also was to be soon displaced. As early as 1942, certain anomalous polarization effects had been noted in barium titanate by a titanium-alloy manufacturer. Investigations at MIT from 1945 onward established that the titania ceramics were a new class of ferroelectric materials having most interesting piezoelectric properties. By 1947–1948, several articles had appeared in the U.S. on piezoelectricity in

poled ceramics. Apparently several Russian studies were also done about this time (Roberts, 1947; Mason, 1971). The many advantages of the new material, including a strong piezoelectric effect, high Curie temperature, and capability of being formed in a variety of shapes, soon resulted in its displacing ADP as the most widely used piezoelectric. The publication of a book on piezoelectric crystals by Mason (1950a) provided the first extensive discussion of these new ceramics. This book covered the properties of the traditional piezoelectrics, the recent man-made crystals (ADP, KDP, and others), and the even more recent ceramics. By the later years of the period, 1954–1955, a second generation of improved ceramics based on lead titanate zirconate, first proposed by B. Jaffe and H. Jaffee of the Clevite Company, was appearing (Mason, 1971; Berlincourt *et al.*, 1964). In closing these remarks on piezoelectrics, it should be noted that over the years, in the face of many new materials, the traditional quartz has remained the material of record in critical frequency-control applications.

From the materials standpoint, developments were less spectacular in the other basic transduction mechanism of ultrasound, magnetostriction. The nickel–nickel alloys used in echo sounders in the 1930s were also used in the sonar units of World War II, with the possible addition of cobalt–iron. The shortage of nickel and cobalt forced the Germans and Japanese to alternative materials, primarily aluminum–iron (Alfer) alloys. Magnetostrictive transducers were highly developed for sonar during the war and for the following five to six years. Since 1950, the development of ferromagnetic ceramics, or ferrites, has occurred, these being the magnetic analogues of the piezoelectric ceramics (Neppiras, 1968).

Two developments particularly important to high-power ultrasonics occurred in this period. The first was the appearance of the modern horn transducer. It may be recalled that Wood and Loomis had made a crude concentrator by drawing a heated glass tube to a point. In 1950, Mason (1950b) filed a patent application in which he described the use of an exponentially tapered half-wavelength rod to magnify the velocity of a resonant piezoelectric transducer. Mason, who was with Bell Telephone Laboratories, had first become interested in piezoelectric crystals as elements in transducers in the mid-1930s, at a time when he was also very active in the field of acoustic filtration. The work on ultrasonic horns was but another part of Mason's contributions to ultrasonics, which ultimately were to encompass nearly all aspects of physical acoustics (Lindsay, 1967). In the case of the horn transducers, Mason and his co-workers applied them in a number of ways for material testing, such as fatigue strength, cavitation resistance, and study of wear (Mason, 1958). The first such transducers were driven by ADP crystals (Fig. 23), but later Mason and Wick (1951a,b) described a barium titanate driven horn transducer. With

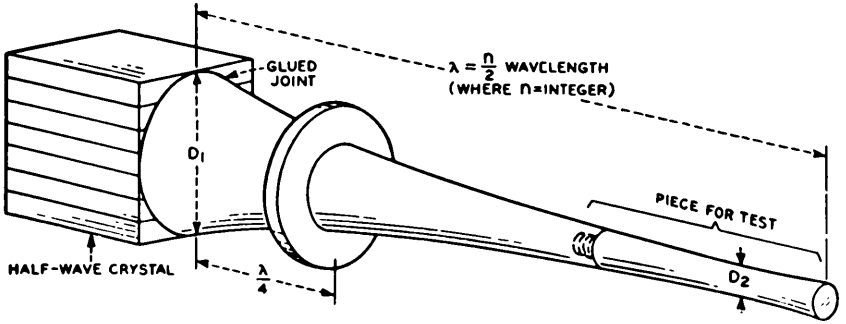


FIG. 23. Exponential horn driven by half-wavelength crystal stack (Mason, 1950a). Copyright (1950) Bell Telephone Laboratories. Reprinted by permission.

the use of tapered concentrators in an ultrasonic machining patent by Balamuth (1948) and the suggestion of various horns for ultrasonic processing by Lozinskii and Rozenberg (1950) in the USSR, the horn soon became a part of the technology. It effectively solved the problem of coupling energy into solid loads.

The second major advancement was the concept of a prestressed sandwich transducer. Ceramics and other piezoelectric crystals suffered the disadvantage of low mechanical strength in tension and difficulty in bonding to metals. The basic sandwich construction had been introduced by Langevin, but did not find extensive application. The ADP sonar units of World War II, for example, used quarter-wave crystals bonded to quarter-wave drivers. In 1954, Miller (1963) introduced the concept of mechanical bias by use of a center bolt to prestress a sandwich transducer in compression. In the opinion of Neppiras (1968), ceramics would have soon lost favor to magnetostrictors for high-power uses without this development. As in many inventions, some controversy attaches to priority in this area. Enz published designs of ferrite transducers under prestress in 1955, for example (Enz, 1955; van der Burgt 1973). Neppiras (1973) indicates the first mention of prestressing was with respect to accelerometers and that, in fact, the concept may have sprung up in several places about the same time.

Since Langevin's invention, the methods of generating ultrasonic waves had been dominated by piezoelectric and magnetostrictive transducers, with little development of other techniques beyond the Hartmann generator in the 1920s. This changed in 1940–1955 with several new generators appearing intended for use with gaseous and liquid loads. The first was an electromagnetic transducer devised by St. Clair (1941) for generating very intense waves in air, and intended for use in agglomeration

processes. The St. Clair transducer was a very-high-Q system consisting of a half-wave aluminum bar driven at resonance through inductive coupling and essentially operated on the principles of a loudspeaker.

Most other developments occurred with whistles and sirens. Hartmann made an important modification to his basic air-jet generator in the period 1948–1951 by adding a central stem through the nozzle to the resonant cavity. L. E. Savoy also devised a similar modification in the period 1944–1950 (Litsios, 1963). In contrast to the original Hartmann device, the resulting stem-jet whistle was able to generate ultrasonic waves at subsonic nozzle velocities. Then Janovsky and Pohlman (1948) presented details on a jet-edge whistle intended for use in liquid processes involving emulsification and dispersion. Pohlman's work in developing the jet-edge whistle represented but a small part of his many activities in ultrasonics. His earlier work on the image converter and on therapeutic uses of ultrasound have already been noted. He additionally worked in the areas of cavitation, material testing and air and underwater sonar. He was later to form the Laboratorium für Ultraschall at the Technische Hochschule in Aachen, one of the few laboratories in the world specializing in ultrasonics research (Anonymous, 1967). In the case of Janovsky–Pohlman whistle, the essential feature was that a jet of liquid impinged on a wedge, exciting it at its flexural resonance. Crawford (1955) has noted that a similar concept was used to test turbine blades. Also in the area of whistles was the design of a vortex whistle by Vonnegut (1954).

Developments in the area of sirens as intense sound generators had been essentially quiescent for a century since Seebeck's 1841 improvement of the La Tour siren. The idea of coupling a siren to a horn had also occurred in the 19th century (as shown by the giant steam siren evaluated by Tyndall). The giant siren was revived in 1941 for use in air-raid warning (Jones, 1946). Allen and Rudnick (1947) put essentially the final touches on the area of rotary sirens with a horn-coupled device capable of generating intense sound at frequencies from 2.5 to 35 kHz. Finally, G. Lavavasseur added a new concept in 1951 with his static siren, which had no moving parts (Crawford, 1955).

B. High-Power Ultrasonics: Research and Application

In considering the major developments in power ultrasonics for the period 1940–1955, the main categories used will be: fundamental studies of intense ultrasound in liquids; appearance of certain major processes (cutting, cleaning, soldering, metal welding, and plastic welding); agglomeration and emulsification; metallurgical processes; chemical–biological processes. In general, the postwar years marked the start of

significant developments in power ultrasonics. However, in exception to this was work done in Germany, where postwar investigations by the British revealed an extensive series of developments that had been underway for several years in power uses of ultrasound. The diversity of uses will be apparent by frequent reference in what follows.

Of all the phenomena associated with the passage of intense acoustic waves through liquids, cavitation was the most known. Nevertheless, although ultrasonic cavitation studies dated almost from Langevin's first tests and hydrodynamic cavitation observations dated back to 1894, little progress in fundamental understanding had occurred by 1940. However, very significant strides were to occur in the next 15 years. While some

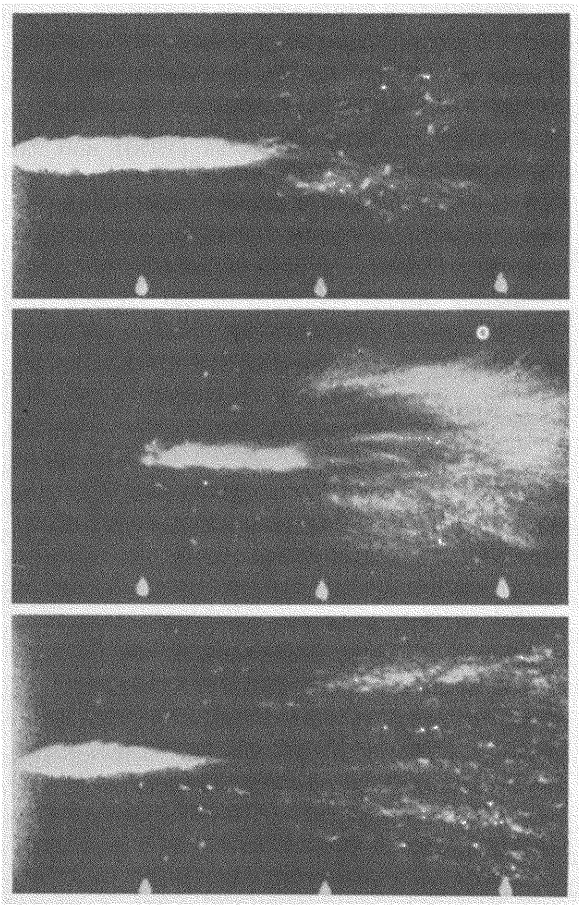


FIG. 24. Cavitation bursts in water (Willard, 1953).

contributions to the theory were made in the early 1940s, the most significant first study of the period was made by Harvey *et al.* (1944). Harvey, who had opened up chemical ultrasonics with Loomis in 1927, established the need for nuclei for onset of cavitation and furnished many of the fundamental concepts adopted in acoustic cavitation (Flynn, 1964).

What might be called the classical studies of cavitation were then done by Blake (1949) at Harvard and Willard (1953) at Bell Labs (Fig. 24). On the one hand, these studies revealed the overall complexity of the cavitation process and, on the other hand, greatly elucidated the major events. The concepts of gaseous and vaporous cavitation, the nature of the cavitation threshold and conditions for growth and decay of bubbles were explained. Other major experimental studies were those of Knapp and Hollender (1948) and those of Esche (1952). Knapp and Hollender made high-speed photos of cavitation bubbles and, among other things, confirmed the large collapse velocities originally predicted by Rayleigh. Esche made measurements of the noise spectrum from cavitation, showing the wide spectrum and the presence of subharmonic frequencies of large magnitude. Finally, Noltingk and Neppiras (1950) and Neppiras and Noltingk (1951) co-authored papers that significantly advanced the theory of cavitation with a more sophisticated modeling of the bubble dynamics.

The phenomenon of sono-luminescence continued to receive attention in this period. While several theories were put forth in the 1930s when it was first observed and others appeared in the period 1940–1955, an accepted explanation did not appear. However, Noltingk and Neppiras, in their 1950 paper on cavitation, noted that very high gas temperatures would be created during bubble collapse, thereby leading to incandescence and an explanation for sono-luminescence. Although unanimity does not prevail on the mechanism of luminescence even to the present day, the thermal explanation set forth by Noltingk and Neppiras in 1950 has become the prevalent one (Flynn, 1964; Noltingk, 1962; Webster, 1963).

Another important phenomenon of ultrasound in fluids, acoustic streaming, was considerably advanced in this period. The first observations of this phenomenon, it may be recalled (from earlier in this account), dated back to the observations of Savart in 1827 and Faraday in 1831, with Rayleigh providing the first analysis in 1884. Little else happened for many years, until Andrade (1931) studied the problem. The work by Schlichting (1932) on boundary layers was not directed at acoustic streaming, but was later to prove essential to the analysis. Cady (1963) has noted that late in World War II he made an experimental study of streaming in the course of measuring transducer power. Eckart (1948) further developed the role of viscosity and the nonlinearity of the equations in streaming. Later, Nyborg (1953) presented an advanced formulation of the streaming problem, and, in the same year, Westervelt (1953) presented a corrected solution to the

problem originally formulated by Rayleigh. Finally, Holtzmark *et al.* (1954) presented the classical result for the streaming pattern near a cylinder in an oscillating fluid, a result that seems to have appeared in every paper and talk remotely related to the subject since then. While other important contributions to the subject were certainly to occur in later years, the major features of the subject were well covered by 1955 (Nyborg, 1965).

Wood and Loomis had observed the drilling–cutting action of ultrasonics in 1927, but this was one area not further exploited in the following few years. Neppiras (1956) noted that the technique was applied on a limited scale during the war years, 1939–1945, for cutting and drilling precious stones, but other references to this effect have not been uncovered. Lewis Balamuth has stated (Rozenberg *et al.*, 1964) that he discovered the machining process accidently in 1942 while investigating the dispersion of solids in liquids by means of a magnetostrictively vibrating nickel tube. Balamuth's first contributions to ultrasonics, made in the 1930s with regard to damping in solids, have already been noted. The discovery of ultrasonic drilling was to be but one of a long series of inventions in the area of power ultrasonics. Balamuth was to be instrumental in forming several ultrasonic companies and became known as one of the leading entrepreneurs of the field. In any event, any ambiquity in the matter of ulrasonic drilling was cleared up by the issuance of British Patent 602801 to Balamuth (1948). By the early 1950s, several equipment makers were producing ultrasonic drills. Magnetostrictive transducers were used during this period. Neppiras's several articles on the details of the process that appeared at the time were most well known and served the additional purpose of introducing practical ultrasonics to a wide range of industrial users. Although the precise cutting mechanism was not completely understood until the 1960s, the process became well accepted by 1955 (Graff, 1975). The principles of ultrasonic machining were extrapolated on a spectacular scale to rock drilling in the early 1950s. The work was done at Battelle Memorial Institute and was aimed toward deep drilling of oil wells. Very large magnetostrictive transducers, operating in the range of 2000–3000 Hz were developed for this work (Dehlinger, 1950).

It is uncertain exactly when a specifically designated ultrasonic cleaner first apeared. The erosive action of ultrasound on surfaces had been noted early in the 1930s, and the ability of ultrasonics to clean and brighten the surfaces of submerged objects must surely have been observed on many occasions in the course of the researches going on with liquid processes at that time [possibly as early as 1932 (Frederick, 1965)]. Weissler (1953) notes a 1943 German patent on ultrasonic cleaning, as well as ultrasonic-laundering patents in 1949. Frederick (1965) indicates that by 1946, two companies were selling units suitable for ultrasonic cleaning. The General

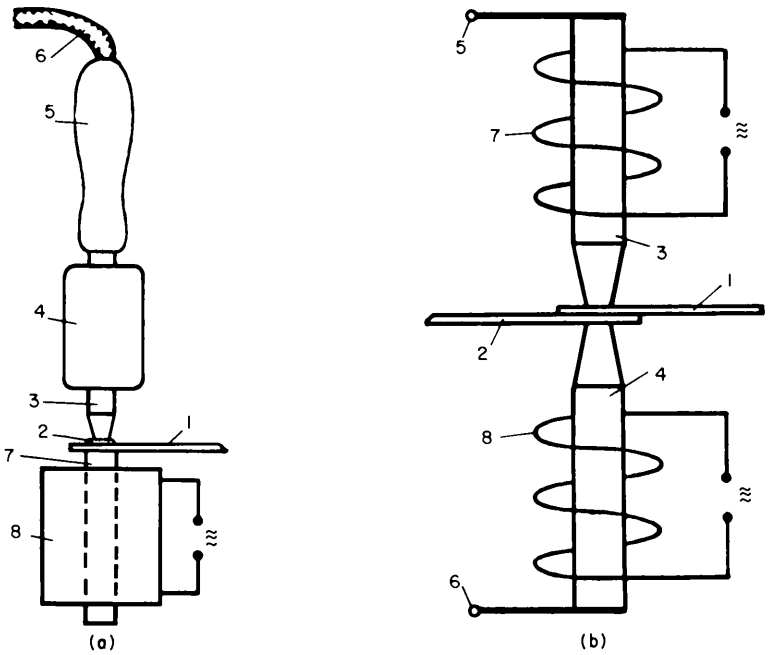


FIG. 25. Early concept of an (a) ultrasonic soldering iron and (b) an ultrasonic spot welding process (Alexander, 1946).

Electric Company was one of the first to market a true ultrasonic cleaner, around 1950. In the following few years, a number of companies manufactured small cleaning units; by 1954, the Detrex Corporation in the U.S., was marketing a large ultrasonic degreaser. The development of the process increased rapidly after that, soon dominating the commercial sales of ultrasonic equipmenmt (Ensminger, 1973).

The third power ultrasonic process, soldering, may in fact be the oldest of the five major processes. The original work was done in Germany as early as 1936, and patents were issued there in 1938 and 1939. The postwar investigations by the British revealed that extensive plans were in progress in Germany for wide use of ultrasonic soldering irons in electronics manufacture (Fig. 25a). Thus, the German firm of Siemens–Shuckert in Berlin developed and had planned to use several hundred ultrasonic soldering irons, but war work prevented more than a few from being built (Alexander, 1946). The technology was quickly picked up after the war, with details of equipment given by Thomas and Simon in 1948. The availability of commercial equipment dates from about the same year. In

1950, Noltingk and Neppiras presented more extensive information on the process. Siemens–Schuckert and Mullar Ltd. marketed soldering irons and small tinning baths from the early 1950s onward. Although enthusiasm for the process was high at that time, it was to fall into general disuse, to be resurrected in the 1970s (Graff, 1977).

The ultrasonic metal-welding process, to be widely used in later years, was only in the stage of research and development by 1955. After the war, it was found that a fairly extensive research program on applying vibrations to welding had been in progress in Germany. The BIOS report 1504 (Alexander, 1946) briefly noted the application of ultrasonics to the spot-welding process, applying vibrations to the electrodes during welding (Fig. 25b). Much more extensive results appeared in BIOS Report 1849, and these were reported in the welding literature by Willrich (1950). Both high- and low-frequency studies were done. Magnetostrictive transducers were attached to one of the spot-welding electrodes in various ways, and ultrasonics was applied during the spot weld with results that were not too encouraging. Quartz-focusing transducers were also tried on the automated arc-welding process. The most consequential result occurred in the low-frequency studies and was essentially accidental. The low-frequency (50 Hz) vibrator attached to the electrode generated a small, lateral rubbing movement and was found to produce a cold weld when no welding current was passed. This aspect was not immediately exploited, however.

In the early 1950s, the American Company of Aeroprojects carried out several government-sponsored projects on use of ultrasonics in welding. For the initial work in 1953, ultrasonics as an adjunct to the spot-welding process was investigated, and the ultrasound was applied in the same fashion used by the Germans, by longitudinal vibration of the electrodes. Some evidence of cold welding was obtained, and in further studies in 1953, this aspect was pursued in depth, using the longitudinal mode. However, in the next period of work, it was found that significant lateral components of vibration were present in the electrodes, so that by 1955, the modern form of the transverse vibrations of the "sonotrode" were being developed (Meyer, 1973). The actual commercial development of the process started about that time.

The fifth of the present-day major power ultrasonic processes, plastic welding, was also just appearing in the 1950s. There were several somewhat obscure patents relating to ultrasonic bonding in the later 1940s early 1950s. The most significant development was the Carwile (1953) patent assigned to the Raytheon company for an ultrasonic welder to seal plastic sheeting. Other general references to the topic suggest that by 1955 it was available as a production technique for joining polyester film. The main growth of the process was not to start until the mid-1960s.

The very broad process areas involving agglomeration (also called flocculation, also precipitation) and emulsification (also called dispersion) will now be considered, along with one or two other somewhat related applications. A large number of research studies had been done in these areas in the 1930s, and were to continue into the 1940s and 1950s. However, the attention here is focused on the practical uses found for these phenomena.

A very active area of industrial development in this period was flocculation of smokes and mists. Some work had apparently been done in Germany in 1938, but had not been continued due to high system costs (Richardson, 1957). In BIOS 1504 (Alexander, 1946), it was noted that laboratory scale tests on smoke coagulation had been done at Siemens, but the war prevented construction of any large-scale installations. The major pioneering effort was done by St. Clair (1940). The previously noted electromagnetic generator that St. Clair had developed was intended for smoke coagulation work, and was undertaken for the U.S. Bureau of Mines. Substantial sized experimental apparatus was shown in the 1948 report of results. Commercial development of the process took place under the leadership of the Ultrasonics Corporation in the U.S., but using high-power sirens instead of the St. Clair generator. The first commercial sonic agglomeration installation was at the S. D. Warren Paper Company in 1948, and numerous other installations followed, with applications being reported by 1951 to removing sulfuric acid mist and dust from lead smelter gases and molybdenum sulfide roasters (Crawford 1955; Neumann and Norton, 1951). Despite the encouraging start in the years 1940–1955, the process has not survived to the present day, probably due to high operating costs.

In the area of emulsification, it was previously reported that low-frequency oscillators found commercial use in milk homogenization during the 1930s. German wartime work on emulsification systems at Siemens was also covered in the report BIOS 1504 (Alexander, 1946; Crawford, 1955), and involved a two-stage emulsification process involving a liquid Galton-type whistle, and magnetostrictive and piezoelectric units for small-scale tests. However, industrial-scale uses were apparently not developed. Although not noted in the report, a 1944 Swiss patent covering emulsification had been issued to Siemens. It was the Janovsky–Pohlman whistle of 1948 and the collaboration of E. C. Cottell and J. E. Goodman, starting in 1951 and leading to new blade-type whistles, that opened up processing possibilities (Brown and Goodman, 1965). The British company of Ultrasonics, Ltd., and the West German companies of Siemens–Schuckert and Lehfeld provided commercial equipment for a number of industrial pilot plant or full-scale uses. About the same time, the Brush Development Company of

Cleveland marketed a high-power processing unit, which employed a large focusing transducer, for general use in process studies (Stokes and Vivian, 1951). Many citations to industrial use during this period, some very general, some quite specific, can be found. Among them were dispersion of pigment in paint and carbon black in lacquer, producing pharmaceutical products and rot-proofing emulsions for canvas, application in the food industry (soups, sauces, salad creams), production of fruit essences for soft drinks, production of cosmetic products, the dispersion of dyes in rayon, and manufacture of shoe polish (Crawford, 1955; Mattiat, 1954; Richardson 1957).

A variety of other practical uses were attempted during this period. The Zeiss company in Germany was degassing molten glass by 1953, and use for degassing transformer oil was also noted about that time. Also, by 1953, the Atlas Werke of Bremen, Germany, was constructing a pilot plant for descaling metal strips (Mattiat, 1954). Commercial apparatus for descaling boilers was also available about that time. These last two uses were basically ultrasonic cleaning applications. A variant of ultrasonic cleaning was proposed in 1949, this being the washing machine for clothes, but the concept did not prosper. The use of ultrasonics for ore flotation processes also was noted in BIOS 1504, where large-scale, relatively unsuccessful tests were reported. Further work on this was reported in 1950 and 1951, but it apparently did not progress to large-scale use. Related to this was the use of ultrasound to destroy foam in flotation processes, first investigated in 1950. The use of ultrasonics to produce sprays and fogs does not seem to have been exploited during this period, although some further studies of the phenomenon were made (Crawford, 1955; Richardson 1957).

Another very extensive area, metallurgical processes, will now be considered. These can be divided into the two categories, broad in themselves, of treatment of metals in the liquid state and treatment in the solid state. In general, it can be noted that by the mid-1950s, ultrasonics applications had been suggested or made to metals in all stages of manufacture, from the liquid melt to making finished products. In fact, the applications started with ore refining and extended to the testing of finished products. Some of the processes in the 1940–1955 period have already been discussed, such as soldering, metal welding, and machining. The balance of the field will now be covered.

The ultrasonic treatment of molten metals was a rich field of study in the 1930s, as has already been noted by the pioneering work of Mahoux and Sokolov. Several patents on processes also were issued in that time. The research then, and into the 1940s and 1950s, had shown several main effects, these being grain refinement, degassing of melts, and dispersion of one metal into another plus changes in some rate processes. In an excellent

review of the metallurgical aspects, Hiedemann (1954), a German pioneer in ultrasonics who had come to the U.S., noted the wide interest in these developments, as indicated by the many reviews and patents. However, he was critical of the poverty of original research in the area, the lack of understanding of process mechanisms, and the general unsatisfactory state of the art. The result was that by 1955, this potentially important technology was basically still in the laboratory stage, a situation still unchanged. Thus in a later book on ultrasonics (Brown and Goodman, 1965), one finds the statement that "The treatment of molten metals by ultrasonics is not a technique in common use in industry. At present, experiments are confined to the laboratory. . . ." Still later, a text (Rozenberg, 1973) refers to the "semi-industrial" equipment being used in metal-crystallization studies.

The application of ultrasonics to metals in the solid state, basically the areas of metal forming, metal removal, and metal joining, saw very little activity in the 1940–1955 period, beyond the already noted soldering, welding, and machining uses. The impression would be quite otherwise if one were to judge by patent activity alone. Over a dozen patents issued in wire and tube drawing, stretch forming, rolling, straightening, and powder metallurgy during that time. Ultrasonic metal-removal processes were proposed in Germany in the late 1930s, with claims of accelerated cutting, greater dimensional accuracy, and lower applied force in such operations as turning, sawing, milling, planing, and boring. Nearly a dozen patents were issued in the 1940–1955 period. In metal joining, excluding soldering and the ultrasonic cold welding process, at least 17 patents issued on application of ultrasound to the fusion welding process, with most of these being German (Hastings and Carter, 1949; Meyer, 1973). The fact remains, however, that this paper activity was not reflected in extensive research and development, let alone industrial use. Thus, by 1955, the use of ultrasonics in solid-state metal processes was in even a less advanced state than liquid processes.

However, at the very close of the period, 1955, a most significant publication appeared that was to break open the field of metal forming processes. Blaha and Langenecker (1955), in their paper Dehnung von Zink-Kristallen unter Ultraschalleinwirkung, essentially stated that an acoustic softening occurred in metals under ultrasonic actions, leading to a reduction in static stress necessary for plastic deformation. This was followed in rapid succession by other basic studies and was to lead to very extensive research (and controversy) on ultrasonic metal forming in the 1960s, as well as some industrial use.

The application of ultrasonics to chemical–biological processes is potentially of greater breadth than any area previously considered because

of the tremendous diversity of reactions that are encompassed. This difficulty was previously noted in reviewing activities during the 1930s. Just a rough breakdown of the field shows the chemical area to cover chemical reactions, polymerization–depolymerization, crystallization and electrochemical processes, with the last covering several areas in itself. Biological areas include biomacromolecules, microorganisms, viruses, and multicellular organisms. Some applications include extraction, bacterial destruction, and production of mutations. The field of agricultural applications, with its own organization, can be included in the biological area. It has been noted, however, that the intent here is to focus on practical applications and major new developments, and not just to chronicle the progress in scientific research. So, despite the tremendous range of the chemical–biological field, the task is drastically reduced, for the simple reason that very little emerged in the way of practical use during this period. Certain practical uses that were essentially chemical engineering oriented, such as agglomeration and emulsification, have already been noted. The biologically related field of medical applications, where practical use did occur, will be discussed separately. However, as will be noted, little else in practical application was to arise in the area.

The fact that interest in chemical biological effects of ultrasound that started in the 1930s continued into the years 1940–1955 is shown by the many research publications in the period. There were several specialty conferences on the subject in the U.S., such as a symposium on acoustics and chemistry in 1950 [see Yeager and Hovorka (1953)], and two symposia on ultrasonics held by the American Institute of Chemical Engineers in 1950 and 1951 [see Neumann and Norton (1951)]. The term "sonochemistry" was coined in 1953 to identify this particular branch of the chemical field (Weissler, 1953). By 1950, it was well established that cavitation was essential for nearly all of the observed chemical effects but which aspect of cavitation was the causitive factor was not known. The practical uses, while very few, were not nonexistent. One area that seems to have aroused considerable interest was the manufacture of alcoholic beverages. Ultrasonic extraction methods were reportedly in use at several German breweries by 1954 (Mattiat, 1954; Crawford, 1955). While the practical use of ultrasound for aging beverages apparently received conflicting reports, a device specifically for this application was produced in Germany. Biological laboratories used ultrasonic units to extract cell contents, a process still being used today. A good deal of attention was given to the agricultural uses of ultrasound, in particular speeding seed germination and increasing plant growth rate, but these did not transition to practical use.

A review of several statements in the 1950–1955 period shows a

general theme of reservation on ultrasonics in chemical–biological processes, to say the least. Many of them seem as appropriate today as at the time of presentation. Thus, Dudley Thompson, in introducing the proceedings of the two AIChE symposia [see Neumann and Norton (1951)] noted that

> Over a period of years, many investigators have reported certain phenomena, primarily qualitative, that have been attributed to the presence of an acoustic field.... Actual potentialities have been beclouded with spectacular and unwarranted claims in certain sections of the popular press.

Others noted lack of quantitative information and comparative studies between ultrasonics and other methods. Others noted the lack of ultrasonic equipment of high power levels and the consequent difficulty of scaling up to industrial size. Thus, by 1955, the chemical–biological process area, like many other ultrasonic areas, was still basically in the test tube.

C. Ultrasonic Nondestructive Testing, Delay Lines, Measurement, and Control

The coming of age of ultrasonics was nowhere better shown than in application to nondestructive testing (NDT). The development of pulse-echo techniques during World War II led to a major industrial application and the association of the word "ultrasonics" with that particular application by most engineers. The development of the ultrasonic delay line, initially used in radar systems, represented a new area in electronics where acoustic devices were to be important. At the same time a wide variety of other low-power applications of ultrasonics emerged, such as flowmeters and liquid level indicators.

In the area of ultrasonic testing, some of the techniques under development in the 1930s found use during the war years. In particular, the image cell developed by Pohlman in 1939 was incorporated into ultrasonic test systems and used by German industry during the war (Hastings and Carter, 1949). Inspection of shell casings was a particular application. The distinguishing characteristic of all this early work was that continuous ultrasonic waves were used, with flaw detection being based on a decrease in radiation intensity, in the same way as radiographic techniques. This approach had some major difficulties, such as generation of standing waves, lack of acoustic transparency of certain thicknesses of materials, and need for access to both material surfaces. There were also sensitivity problems, in that flaws might only intercept a small portion of the

transmitted energy and thus cause only a slight change in the energy received.

Since the pulse-echo concept was involved in the use of ultrasonics in submarine detection, there can be little doubt that the same notion occurred to pioneers in nondestructive testing. However, the difficulties of measuring the small time intervals involved in scaling down the sonar concept were beyond the electronics capabilities of the early 1930s. These conditions were changed with the advances stimulated by the work on radar that started in 1935 and achieved practical use by 1938. The electronics developments for pulse-echo radar were directly applicable to ultrasonic technology.

In 1940, Dr. Floyd A. Firestone, an associate professor of physics at the University of Michigan, filed application for a patent on a "flaw

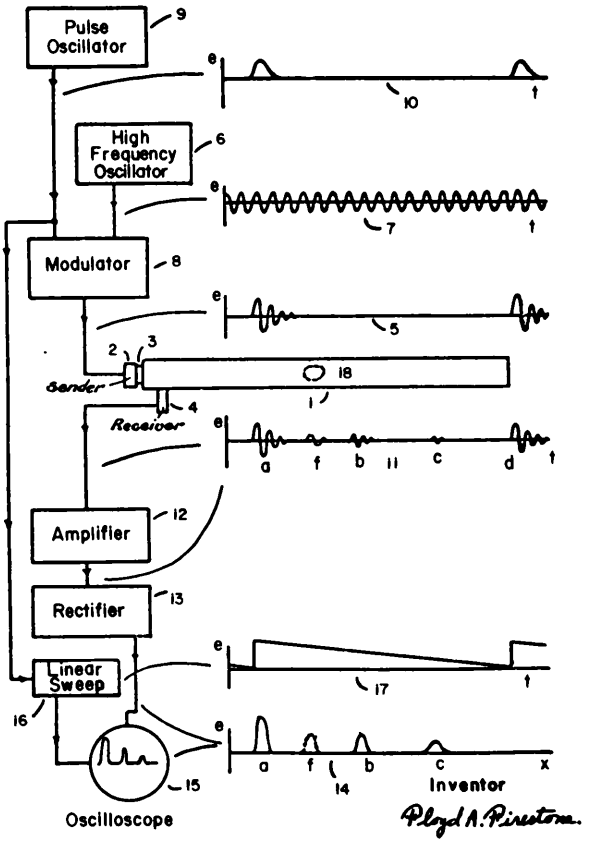


FIG. 26. The first patent on pulse-echo nondestructive testing (Firestone, 1942).

detecting device and measuring instrument" (Fig. 26), that used ultrasonic pulse reflection techniques (Firestone, 1942). Firestone had been working in acoustics for many years while at Michigan and had received his early training from the famous acoustic experts Professor Dayton Miller at Case and Dr. Paul Sabine at the Riverbanks Laboratory of Acoustics. During the 1930s, Firestone was pioneering the mobility method of analyzing vibrating systems versus impedance techniques. The new pulse-echo instrument for ultrasonic testing, dubbed the "Reflectoscope," eliminated most of the difficulties inherent in a continuous wave system. For example, the sensitivity problem was completely transformed since a flaw was now shown by the presence of any signal, versus a slight change in a continuous signal level.

Independently, work was being done in England at the same time, headed up by D. O. Sproule of H. Hughes & Sons, Ltd., of London. The company had been requested to examine supersonic methods of flaw detection by a subcommittee of the Alloy Steels Research Committee in April 1940. Continuous-wave techniques, using a Pohlman cell, had first been tried but discarded. By 1942, Sproule had devised the pulse-echo technique, using separate sending and receiving transducers (Desch *et al.*, 1947). This use of dual transducers was to distinguish the "British" approach from the "American" way of a single transducer for a number of years. Postwar investigations by the British were to reveal that Sproule and Firestone were not alone in their developments. Apparently F. Kruse in Germany, who at the time of his doctoral thesis in 1938 had been quite discouraged over the prospects of ultrasonic nondestructive testing, had also developed a flaw detector similar to the Firestone and Sproule instruments (Hastings and Carter, 1949). Sokolov, in 1941, also had devised an NDT method involving ultrasonic pulses and pulse frequency modulation concepts. This was based on yet earlier concepts of Hiedemann, and was to find use in thickness gauging (Krautkramer and Krautkramer, 1969).

Following Firestone's and Sproule's inventions, developments in pulse-echo ultrasonic testing were rapid. The first instruments appeared on the market in 1943, with Sperry Products producing the Reflectoscope and Kelvin & Hughes, Ltd., of London producing the Sproule instrument (Krautkramer and Krautkramer, 1969). Many further contributions were made by Firestone and his co-workers, including shear wave testing of welds and methods of damping transducers. A particularly notable one was the use of surface waves in NDT, introduced in 1946, and representing a further example of the impact of Rayleigh on the field of ultrasonics (Firestone, 1948).

Not all ultrasonic testing was in pulse-echo methods during that time. Based on a patent by Erwin and Rassweiler (1944), General Motors built the first resonance thickness tester in 1947, known as the Sonigage (Krautkramer and Krautkramer, 1969). Branson Instruments was founded by Norman Branson in 1946, and introduced the Audigage (under the Erwin patent) and the Vidigage, both thickness-measuring instruments (Posakony, 1976). The Magnaflux Corporation marketed a similar instrument known as the Sonizon. These instruments found wide use from the early 1950s onward in such areas as testing of rails, pipe, and plate.

Returning to the pulse-echo testing method, Donald Erdman introduced immersion testing techniques in 1948 (McMaster, 1959). Erdman (1952) also pioneered the use of the B-scan technique of ultrasonic testing. The company Electrocircuits, with Erdman as President, was formed in 1951, to market ultrasonic immersion testing units. In later years, 1959–1960, serious litigation was to arise between Sperry Products and Electrocircuits over numerous claims of the Firestone patents, some as they related to immersion testing (Federal Supplement, 1959; Federal Reporter, 1960). Defense of the Firestone claims was to involve comparisons with early Langevin developments in depth sounding and Sokolov ultrasonic testing methods of the 1930s. In general, developments in ultrasonic testing in this period were such that by 1949 a special session of the American Society for Testing Materials was devoted to the topic [see Hastings and Carter (1949)]. The techniques of ultrasonic NDT were soon to spin off to such diverse nonindustrial uses as medical imaging and measurement of back fat on hogs. By 1955, the pulse-echo method was the dominant ultrasonic testing technique.

While ultrasonics was a beneficiary of radar developments in nondestructive testing, it was to be the benefactor of radar in another area. Radar requirements for delaying electronic signals for microsecond to millisecond durations, such as needed for identifying moving targets, resulted in development of ultrasonic delay lines. Although the earliest recorded use of an acoustic delay line was the "speaking tube" used in some early telephone systems, the ultrasonic delay line was the first completely new use of ultrasound to emerge in many years. The pioneering work was done in England at the Telecommunications Research Establishment (TRE, later called the Royal Radar Establishment) on delay lines for Doppler radar. The first system put into practical use utilized quartz transducers and water as a delay medium. Work going on at Bell Labs at the same time resulted in the first delay line to be used in a wartime application in 1942. This was known as the Shockley delay unit and was used to provide a variable calibrated trigger for radar. It used quartz

transducers and a water–ethylene glycol mixture as a delay medium. Water was recognized as being far from an ideal delay medium, leading to investigation of mercury delay lines, starting at Penn State in 1943 and a bit later at the MIT Radiation Laboratory. Several different radar systems were developed with mercury delay lines, with these types of lines being widely used through the mid-1950s (Brockelsby *et al.*, 1963; Arenburg, 1948).

The advantages of using solids over liquids (the need for liquid tight cells being one factor) led to early investigations of various solids by TRE in 1941, and the identification of vitreous fired quartz as an excellent delay medium by 1943. Work on solids started at MIT in 1943, and was later taken up at Bell Labs. Although development difficulties involving such items as propagation modes and material attenuation led to a temporary suspension of work, a successful fused quartz folded delay line was successfully operated by October 1945. Problems of bonding transducers to solid lines existed, but were overcome by 1954. Apparently the Germans were actually the first to use solid delay lines in the form of glass rods for producing short delays in ultrasonic "echo boxes" used for radar calibration, and also used in antijamming devices. By 1948, guided wave delay lines in the form of thin wires driven by magnetostrictive transducers were in use (May, 1964). Although early uses of delay lines were in radar, uses in computers, television and other communication systems were to evolve (Arenberg, 1948; Brockelsby *et al.*, 1963).

Many other applications of low intensity ultrasonics were to arise in 1940–1955. A large number were in the scientific area, aimed at measurement of material properties and thus represented extensions of uses triggered by the development of resonance and interferometer methods started in the 1920s. The many other industrial and consumer uses that arose can only be briefly noted, giving approximate dates of appearance. Ultrasonic flow meters were developed by 1949–1950, although the earliest use of sound to measure flow dated back to a 1931 German patent. Apparently attempts to commercialize flow meters in the 1950s were not too successful, although acceptance was to come in the 1960s (Lynnworth, 1975). Ultrasonic burglar alarms (and flame detectors) were proposed by 1952 (Crawford, 1955), and liquid level sensors were in existence by 1953 (Bergmann, 1954). The use of airborne ultrasound as an aid to the blind was being developed by 1948, and as a driving aid in fog by 1949. History records the ultrasonic garage door opener appeared in 1947 (Carlin, 1949). The phenomenon of acoustic emission, in later years to be a widely developed NDT tool and generally regarded as a relative of ultrasonics, made its appearance with observations by Mason in 1948–1950 and by J. Kaiser in 1950–1953 of emissions from metal specimens (Lord, 1975).

Although the field of acoustic—or ultrasonic—holography was many years away as a formal discipline, what have been construed as the first acoustic holograms were made in 1950 (Kock, 1973).

D. Medical Ultrasonics

Following Pohlman's 1939 investigation on medical uses, the 1940–1955 period saw extensive application of ultrasound in medicine. The range of uses could be divided into therapeutic, diagnostic (and measurement), surgical, and safety areas, with uses in dentistry being a related field. By 1955, all of the major areas of ultrasound in medicine had been established, although the degree of emphasis of the various areas was quite different from the present day.

During the later 1940s and early 1950s there was a major surge in development and use of ultrasonic therapy. The therapeutic effects appeared to be similar to those obtained by any other heating techniques and were nonspecific. Unfortunately, ultrasound was used indiscriminately in treating a great variety of conditions (White, 1976). Beyond simple diathermy, "Ultrasonotherapy," as it was known, was claimed to cure, improve, or relieve an amazing list of ills, ranging from abscesses to warts. The treatment of cancer received particular attention in the late 1940s (Kelly, 1965), but, along with many other therapeutic uses, was called into question by later work. There was considerable variation in use from country to country. It was widely used in Germany, had some use in Switzerland, Austria, and France, and very little use in Great Britain. There was apparently some resistance to its use in the U.S., accompanied by criticism of medical practice elsewhere that had adopted an unproven technique (Crawford, 1955; Herrick and Krusen, 1954; Kelley, 1957; Richardson, 1957). A review of a 1950 International Ultrasonics Congress in Rome noted (Fox, 1951), with reference to experiments in biology and medicine, that even though an enormous quantity and variety of experimental observaions had been reported, the work was largely uncoordinated and not leading to a fundamental understanding of the processes involved in medical and other biological uses. This perhaps reflected the general trend of thinking, with the result being that by 1955, enthusiasm for ultrasonic therapy had considerably declined.

The history of the use of ultrasound for diagnostic purposes has been extensively recounted by White (1976). Apparently the first effort was made as early as 1937 by Dr. K. T. Dussik, a psychiatrist at a hospital in Bad Ischl, Austria, and his brother F. Dussik, the hospital physicist. In that year they began to study the possibility of making images of the brain by recording variations in the intensity of an ultrasonic beam transmitted

across the head. In 1942, they published their first "hyper-phonogram." In 1946, A. Denier of France also attempted to image ultrasound transmitted through various regions of the body. The work of the Dussiks stimulated efforts in the U.S., where development of a transmission imaging system was started by a team from the MIT Acoustics Laboratory and Massachusetts General Hospital. The first publications from this group appeared in 1950 (Herrick and Krusen, 1954).

The development of the ultrasonic pulse-echo method of nondestructive testing led to its early use in imaging sudies. In 1949, Ludwig, associated with the MIT group, had demonstrated the feasibility of detecting gallstones implanted in dogs from their ultrasonic reflections. By 1950, the MIT group was considering both the reflection and transmission techniques, but their interest in these methods declined. However, pulse-echo work was started by John J. Wild at about the same time at St. Barnabas Hospital in Minneapolis. In 1949, Wild became interested in using ultrasound to measure the thickness of the bowel. Using an ultrasonic tank from a radar simulator at a Naval Air Station, he found it possible to obtain echoes from several thicknesses of bowel and from different layers of tissue. In 1951, working with John Reid, a correct diagnosis was given for a malignant cerebral tumor. This work led to development of the "Echoscope" as a diagnostic instrument (White, 1976; Kelley, 1957).

A third group involved in ultrasonic imaging in the U.S. was started by Dr. Douglass Howery in the late 1940s. Starting work in his basement in 1947, Howery and his co-workers were displaying echoes from soft tissue interfaces by 1949. Dr. Joseph Holmes, at the University of Colorado with Howery, also became involved with this work at an early date. Concentrating their efforts from the start on developing an imaging system, the first two-dimensional tomograms were published in 1952, shortly after those made by the Minneapolis workers had appeared. The early apparatus required complete water-bath immersion, with a surplus gun turret from a B-29 bomber being used for the first scanning tank. The first tomogram of the brain was made in 1954 (Fig. 27). Howery was disappointed at the lack of echoes in the image and concentrated afterwards on soft tissue imaging (White, 1976; Anonymous, 1975).

Investigations on the use of ultrasound for diagnostic purposes sprang up several other places in the early 1950s. The first European work was started at the Royal Cancer Hospital in London in 1951 by Walton and Durden-Smith, and later, by Turner. The work was directed toward imaging the skull and continued until 1955. By 1952, Kikuchi, Tanaka, and Uchida were active in Japan, working on obtaining echoes from structure within the brain. In 1953, Carl Hellmuth Hertz, whose great uncle was

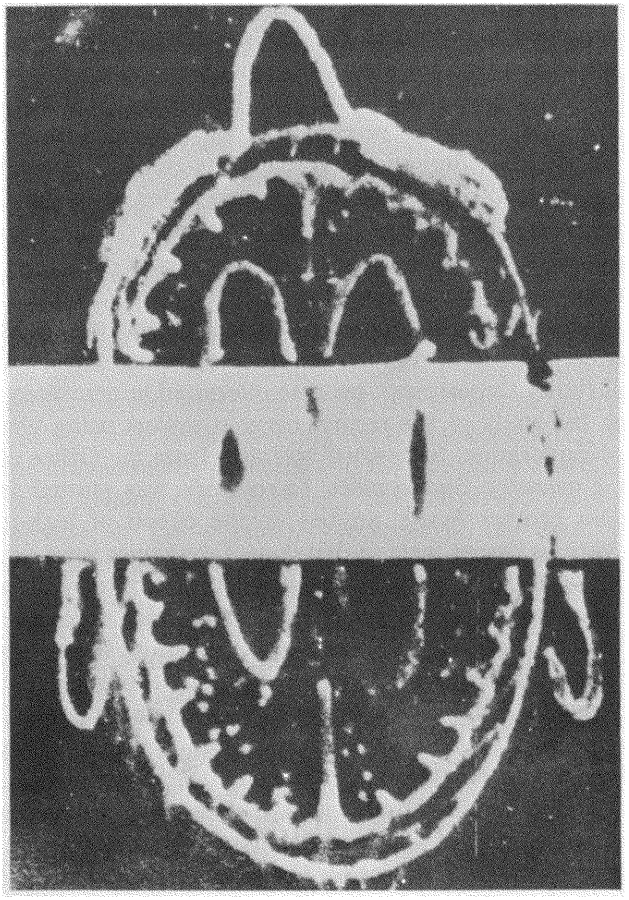


FIG. 27. The first ultrasonic tomogram known to have been made of the brain (White, 1976). Reproduced by permission of Ultramedison of Kingston, Ontario, from "Ultrasound in Medical Diagnosis."

Heinrich Hertz, and Inge Edler in Lund, Sweden, became interested in applying ultrasound to cardiac diagnosis. Using NDT equipment borrowed from a shipyard, they obtained the first echoes from the heart structure, and went on to open up the field of echocardiography. Lars Leskell, also of Lund, after a discouraging effort at using ultrasound to obtain echoes from the brain in 1950, made a new attempt in this area after the successful tests of Hertz and Edler and by 1953 had obtained the first important measurements of midline shift in the brain. The work by Leskell and by Douglas

Gordon in England in the 1954–1955 period effectively opened the field of echoencephalography (Edler and Hertz, 1977; White, 1976).

Another application of ultrasound to medical measurement was in blood flow. The development of ultrasonic flowmeters in the late 1940s drew the attention of medical people. The first blood flowmeter was developed at the National Bureau of Standards by Henry Kalmus in 1953. Dr. Herrick of the Mayo Clinic also became interested in this problem in 1951, and reproduced the Kalmus technique by 1954. In 1953, an ultrasonic viscosimeter, of use in measuring blood viscosity changes during clotting, was available (Herrick and Krusen, 1954; Kelley, 1957).

The use of ultrasonics for surgical procedures first appeared in the 1940–1955 period. At the University of Illinois, William J. Fry, of the Electrical Engineering Department, became interested in neurology and by 1947 undertook the study of the central nervous system using ultrasonics as the means of exploration. Supported by funds from the Office of Naval Research, the University Bioacoustics Laboratory was started in 1948. Working with his brother Frank, studies on the effects of high-intensity ultrasound on various tissues were undertaken. Techniques were developed for focusing ultrasound at select locations in the brains of test animals, and a program of mapping the nerve paths within the brain was undertaken. By 1953, the first reports were made on producing lesions in the brain of a cat by "neurosonic surgery." In 1955, a cooperative program with Dr. Russell Meyers and the State University of Iowa Hospital at Iowa City was started that was to lead, in 1958, to the first use of ultrasonic neurosurgery on humans (Herrick and Krusen, 1954; Kelly, 1957; Kingery *et al.*, 1967).

The treatment of Meniere's disease, a malady of the inner ear, was another surgical application of ultrasound that arose in the 1950s. F. Krejci is credited with first using ultrasonics for treating the disease in 1952, while Dr. Michele Arslan from Italy, often credited with first uses, developed the ultrasonic tool and technique starting in 1952 and published the first favorable results in 1953 (Kelly, 1965; Gordon, 1964).

Another aspect of ultrasound and medicine that arose in this period was concern for potentially harmful effects of ultrasonic radiation on humans. A part of this concern was in connection with the deliberate exposure of humans to ultrasound during therapy and diagnosis. In 1955, J. H. Aldes extensively summarized the experience in safe dosages for therapeutic uses, with many people addressing the matter prior to then (Kelly, 1957). The other part of the concern was the inadvertent exposure of humans to acoustic radiation while using ultrasonic tools or other systems capable of generating high-frequency noise, such as the newly developed jet engines. The term "ultrasonic sickness" was being used near the end of World War II to denote some of the ill effects noted from

exposure to jet noise and in 1947, an Ultrasonics Panel was formed by the Aeronautical Board in the U.S. to investigate such problems. It is recorded that by 1948 the first legal case based on claim of injury by ultrasonics occurred in England (Davis, 1948). C. H. Allen and co-workers were noting some of the ill effects of intense airborne ultrasound by 1948 (Allen *et al.*, 1948). In addition to lethal effects on roaches and mice, momentary dizziness and heating of exposed parts of the body were noted in humans. Others noted fatigue, nausea, headaches, and the like. Certain of these reports were picked up by the popular press and led to articles on a new "death ray" (Michael *et al.*, 1974). In a review by Gorskhkov *et al.* (1966), it is said that long exposure of 80 volunteers in Spain to intense jet noise resulted in the death of 28 and permanent debilitating effects in the remainder. Numerous other studies of the effects of intense ultrasound occurred in the early 1950s, covering the effects on hearing, the absorption of airborne ultrasound by animals, and the subjective as well as physiological effects on humans (Michael *et al.*, 1974). By 1955, there was considerable awareness of the potential problems in the area, but the most extensive research was to occur in later years.

Ultrasonics also found limited use in dentistry during this period. Lewis Balamuth, responsible for many new applications of ultrasonics over the years, stated that ultrasonics in dentistry began in November 1952, when an industrial ultrasonic impact grinder was used to drill extracted teeth, with the first use in human patients reported in 1954 (Balamuth, 1963; Brown and Gordon, 1967). However, more recent information indicated that work in the dental area was ongoing at Battelle Memorial Institute in the period 1949–1950 (Ensminger, 1978). The U.S. Navy also developed an experimental ultrasonic dental drill, reported in 1955 (Carlin, 1964). While a number of ultrasonic dental drills were manufactured (about 1500), the development of a new, faster, high-speed rotary drill occurred at about the same time and soon replaced the ultrasonic technique. While the drill did not succeed, new applications of ultrasonics to peridontal work were to arise in the later 1950s and to become accepted in later years.

E. Notes on Physical Acoustics, Underwater Sound, and Bats

In nearing the end of this historical account of ultrasonics, final attention will be given to several other areas that have been followed through their years of development. Physical acoustics, a well-defined, very extensive part of ultrasonics by 1940–1955, will be noted first. In the area of liquids, light-scattering studies of the type initiated by Debye and Sears were extended to the very-high-frequency ranges (10^{10} Hz) by C. S.

Venkateswaran in India in 1942. A great deal of important experimental progress occurred after the war when people experienced in the radar pulsing techniques returned to scientific work. However, P. Biquard, who had also pioneered in light diffraction work in the 1930s, was far ahead of others in applying pulse techniques to liquid measurements in 1943 (Herzfeld, 1966; Stephens, 1975). Similar work was being done at other places, such as MIT and Cambridge University by 1946–1947. The wartime work on underwater sound had focused attention on the inadequate understanding of sound attenuation in water, and seawater in particular. By 1946, F. E. Fox and G. D. Rock had significantly extended understanding of absorption in water. The first general theory for absorption in seawater was given by L. N. Liebermann in 1948, noting the effects on the absorption coefficient due to chemical reactions of the magnesium ion. In 1947, J. M. M. Pinkerton established a classification of liquids according to processes in the liquids, leading to Kneser liquids, rotational isomers, the class of associated liquids, viscous associated liquids, and the class of solutions and pure liquids (Herzfeld, 1966). W. P. Mason and other workers in the late 1940s were using material impedance measurements at ultrasonic frequencies to investigate liquid viscoelastic properties (Stephens, 1975). Truesdell (1953), then at Indiana University, set forth his theory for absorption and dispersion of acoustic waves in fluids. Finally, C. Zeners, in his 1949 boŏk "Elasticity and Anelasticity in Metals," gave the first extensive explanation of damping in metals (Herzfeld, 1966).

The field of underwater sound was one of enormous activity during the 1940–1955 period, spurred by World War II antisubmarine needs and the follow-on period of Cold War tensions. During the war, the American submarine-detection systems used the magnetostrictive and newly developed ADP transducers previously described. The official acronym of SONAR (for *so*und *n*avigation *a*nd *r*anging) was attached to the American ultrasonic system of detection by F. V. Hunt (Bouyoucos, 1975). Many improvements in mechanical and electronic instrumentation were added to sonar systems to increase their effectiveness. These included such items as streamlined domes, reverberation control of gain (RCG) to reduce "main bang" noise and equipment operational simplifications. Many other acoustic developments were made, such as development of sonobuoys, acoustic torpedos, and new listening hydrophones (Keller, 1947; Lasky, 1975a,b).

The results of much of the work in underwater sound done in the U.S. during the 1940–1945 period appeared in 19 volumes of the Summary Technical Reports of Division 6 of the NDRC. In a brief retrospection, Hunt (1972) encapsulated this period of activity in underwater acoustics as

follows:

> This brings my half-century to the brink of World War II, when the field of underwater acoustics was invaded by a host of civilian scientists operating under the sponsorship of NDRC and its second-growth parent, OSRD. During the period 1941–1945, frontiers were breached in almost every direction. Major strides were made in both the understanding and technology of transducer design. The directionality of sound transmission and reception was exploited through the bearing deviation indicator, the sector scan indicator, and the design of directional listening arrays. The basic advantages of scanning sonar were established and demonstrated. The acoustic torpedo, which had been invented with an average frequency of about once a month from 1920 to 1940, was developed successfully in the remarkable short period of 18 months from project assignment to operational use; and it was the first homing missile of any type to become operational. We compensated for Doppler with one hand and tried to exploit it with the other. The foundations of correlation and matched filtering were laid down in Norbert Wiener's classic Radiation Laboratory report that became known as the "yellow peril." Finally, and not least among these achievements, the classic 1943 report by Ewing, Worzell and Vine set forth the remarkable properties of the deep sound channel.

While the NDRC documents chronicled the past, a panel on underwater acoustics of the Committee on Undersea Warfare met in 1948 to establish the basic directions of future research in underwater acoustics. The so-called "green book" (P.U.A., 1950) outlined problems in generation and reception of sound, standards, noise, transmission and scattering, and other areas. By that time, the complexity of sound propagation in the sea was well recognized, although far from understood. The previously noted attenuation problem was but one of the factors that included thermal effects, reflection and scattering from the ocean bottom and surface, a wide class of noise sources, and existence of sound channels, to name a few. The need to increase propagation distances brought forth lower transducer frequencies (Berman and Guthrie, 1972). Thus, the field of sonar, which had given birth to modern ultrasonics, was soon to move back to the subultrasonic frequencies for some uses. Many new sonar concepts were soon to be developed. The original concept of the narrow beam "searchlight" sonar was widely used during World War II. The improved scanning sonar was developed at the end of the war, and by the 1950s the further improved rotational directional transmission sonar was

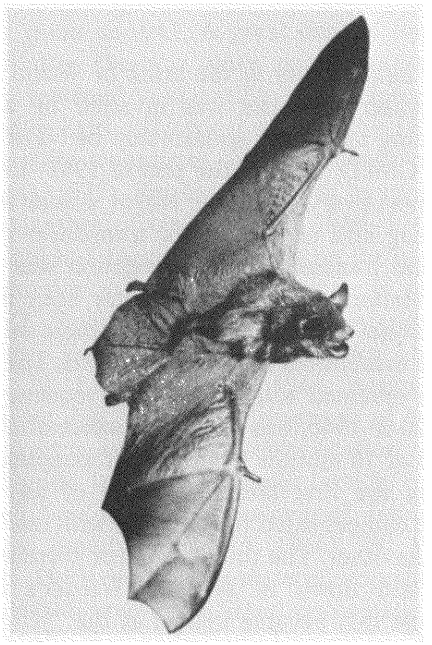


FIG. 28. The first ultrasonic echo-ranging system—the bat (Griffin, 1958).

arriving (Winder, 1975). Sophisticated new methods for beam forming were being developed by 1952. Not all applications were militarily based, however. Proposals for acoustically delineated channels for ship-to-ship communication were made in 1948 although the concept dated back to Boyle and other pioneers in the 1920s. The fish echoes that had been noted from depth sounders as early as 1934, were successfully applied to commercial ultrasonic fish finders by 1951 (Crawford, 1955; Richardson, 1957).

The story of the bat (Fig. 28), the creature whose natural ultrasonics opened this history in the 1790s, will now be completed. Recall that Griffin and Pierce had detected the bat's ultrasonic cries in 1938, but had suggested it was a type of warning note. In the following year, Griffin and Robert Galambos further pursued this matter. They quickly found that the directional characteristic of the parabolic horn sonic detector had led them to the erroneous conclusion that bats only occasionally emitted cries; there was, in fact, a continuous emission of pulses. Repeating the types of experiments first carried out by Spallanzani and Jurine nearly one-and-a-half centuries before, but now more humanely, they showed in studies published in 1941 and 1942 that obstacle avoidance was based on the ultrasonic cries. By 1945, the term "echolocation" was applied to the bat's

capability of perceiving objects, and it was clear that this was the means bats used to orient themselves. By 1951, Griffin was investigating the bat's use of ultrasound in pursuing flying insects (Griffin, 1958). Although it was shown by 1950 that many species of moths and other insects had excellent sensitivity to ultrasound, it was to be somewhat after the period, 1957, when it was first shown that the moths had, in fact, built in evasive countermeasures to the bat's ultrasonics (Sales and Pye, 1974). The starting point for most investigations on insect ultrasound was Pierce's (1948) book on the songs of insects.

With the matter of echolocation in bats well established by 1945, the question of such capability in other creatures was considered. During the war, the wide use of refined underwater listening devices had revealed the immense variety of sounds from marine life, some very definitely in the ultrasonic range. American submarines in the Pacific were even provided with charts of areas where intense shrimp noises could be used to blanket the sounds generated by the submarines (Howeth, 1963). Porpoises and dolphins in particular had been identified as a source of "weird" sounds, with one author of a sonar book (Horton, 1959) noting almost disparingly that "the description of the noises made by porpoises are so varied as to be useless for purposes of identification. They have been said to bark like a dog, gobble like a turkey, or to make a distinctive bubbling whistle, like nothing but a porpoise. They are often heard during echo-ranging operations over equipment responsive to a narrow band of ultrasonic frequencies only. They seem, at such times, to be commenting on the echo-ranging transmission." Possibly the first person to suspect that dolphins and porpoises used sound for navigation was McBride, a curator at the Marineland Aquarium in Florida. Unpublished notes made in 1947 suggested a "specialized mechanism of information by sound" (Sales and Pye, 1974). By 1950, several investigators were studying the sonar possibilities of dolphins and porpoises, most notably William Schevill and Barbara Lawrence, who were analyzing dolphin sounds in 1949, and W. N. Kellogg and co-workers at the Oceanographic Institute at Florida State. By 1953, Kellogg *et al.* were strongly suggesting the possibilities of echolocation by these mammals. The conclusive tests on this, showing the echolocation capability of dolphins, were started by Schevill and Lawrence at Woods Hole, Massachusetts, in 1955, with the results being published in 1956 (Griffin, 1958; Sales and Pye, 1974).

F. Books

As has been noted, a mere handful of books on ultrasonics had appeared prior to 1940, with Bergmann's "Ultraschall" that appeared in

1938 being the most important. From 1940 until 1955 a number of books, ranging from the quite general to the highly specialized, appeared on various aspects of ultrasonics. By organizing the body of knowledge that was by then a comparatively mature field, these books reflected the fact that ultrasonics had indeed come of age. Proceeding somewhat in chronological order, it should be noted that Mason's (1942) "Electromechanical Transducers and Wave Filters" covered a wide gamut of filtering elements, including a part devoted to piezoelectric and magnetostrictive systems. Cady's (1946) treatise on piezoelectricity has already been noted. Mason's (1950a) book on "Piezoelectric Crystals and Their Application to Ultrasonics" set forth the mathematical basis for transducers and resonators and served as the standard reference of rational design for many years.

Reference should be made to two important books on acoustics that appeared in this period. Wood's (1940) "Acoustics," serves as a valuable reference even today. Many references to high-frequency phenomena appear, and particular attention is given to citing the early works in each area. Late in the period Skudrzyk (1954) published "Die Grundlagen der Akustik," a 1000-page treatise that has both extensive mathematical development and a wide range of applications, including some ultrasonics.

Continuing in the area of ultrasonics, what might be termed the first popular book on ultrasonics entitled simply "Ultrasonics," was published by Carlin (1949). Carlin, who had worked at the University of Michigan with Firestone during the development of pulse-echo NDT, wrote a book nonmathematical in nature, which presented the essential technical features of a wide range of ultrasonics uses. A short book by Vigoureux (1950), "Ultrasonics," was oriented toward physical acoustics. Similarly, Richardson's (1952) book, "Ultrasonics Physics" was essentially devoted to physical acoustics. Particular mention should be given to the appearance of a collection of some of the scientific works of Paul Langevin (CNRS, 1950). The father of ultrasonics had died in 1946.

The years 1954–1955 saw the appearance of several books that have use down to the present day. Hunt's (1954) "Electroacoustics" has continued to serve as the standard reference on impedance concepts for transducers. A large portion of the book is devoted to the historical development of the field. In the same year, the monumental sixth edition of Bergmann's (1954) "Der Ultraschall" appeared. Over 1100 pages in length, and containing 5162 references, the book still serves as the reference source of all early work (a 1957 literature supplement appeared that extended the period of coverage). Shortly after, Crawford's (1955) book "Ultrasonic Engineering" appeared. Like Carlin's book, this was nonmathematical in nature. It gave special emphasis to high-power ultrasonics,

one of the very few to do so. Finally, Hueter and Bolt (1955) published "Sonics" in the same year. Fairly broad in coverage, and giving attention to the mathematical aspects, the book has served as the best single textbook on the subject of ultrasonics since its publication.

VII. To the Present

Where the dividing line between the historical and contemporary period of a subject lies may be well known to professional historians, but has been somewhat arbitrarily decided here. The appearance of several important textbooks on the subject in 1954–1955 was one factor suggesting that as a good closing period. Only a few remarks will be made on the "nonhistorical" period since 1955. As a start, it may be noted that acoustic emission has become a major field of research and development in nondestructive testing, although large scale industrial use has still not occurred. It also appears that major new efforts are being made in the way of fundamental studies of ultrasonic NDT (or, in the modern parlance, NDE, nondestructive evaluation). The field of ultrasonics and electronics has been further intermingled with the appearance of surface acoustic wave (SAW) devices in 1965. Judging from the outpouring of technical literature, one could believe this was the only field of ultrasonics. The advent of the laser resurrected interest in many of the acousto-optic phenomena of the 1930s, as well as interest in the interaction of laser beams with surface waves. The field of acoustical holography has emerged as a major field of study. In the area of physical acoustics, higher and higher frequencies have been generated, with 10^{11} Hz having been achieved. The field of ultrasonic imaging in medicine, just underway in 1950, had become a major area by the 1970s. In the areas of underwater sound, medical imaging, and NDT, sophisticated signal processing of ultrasonic signals have become common.

The field of power ultrasonics had grown in terms of commerical use by the 1970s, although it was still highly restricted in terms of range of processes. Ultrasonic cleaning had become the major use since 1955. Plastic welding, new on the scene at that date, was extensively developed from the early 1960s onward and was coequal with cleaning by 1975. The pioneer process of ultrasonic machining found only restricted uses in the 1970s. Another early use, ultrasonic soldering, had only minimal use for many years, but saw enthusiasm redeveloping in the the 1970s. Ultrasonic metal forming, just starting in 1955, was widely researched through the 1960s, but little industrial use has emerged. The metal-welding process was developed in the late 1950s into an accepted industrial tool and is now starting to achieve widespread use.

References

Abello, T. P. (1927). *Proc. Natl. Acad. Sci.* **13**, 699.

Aigner, F. (1922). "Unterwasserschalltechnik (Grundlagen, Ziele und Grenzen)." Krayn, Berlin.

Albers, V. M. (ed.) (1972). "Underwater Sound." Dowden, Hutchinson, and Ross, Stroudsburg, Pennsylvania.

Alexander, P. (1946). Industrial Applications of Ultrasonics: Researches Carried Out at Siemens–Schuckert–Werke, Berlin. BIOS Rep. No. 1504

Allen, C. H., and Rudnick, I. (1947). *J. Acoust. Soc. Am.* **19**, 857.

Allen, C. H., Frings, H., and Rudnick, I. (1948). *J. Acoust. Soc. Am.* **20**, 62.

Altberg, W. (1903). *Ann. Phys. (Leipzig)* **11**, 2, 405.

Altberg, W. (1907). *Ann. Phys. (Leipzig)* **23**, 267.

Andrade, E. N. (1931). *Proc. R. Soc. London, Ser. A* **134**, 445.

Anonymous (1967). *Ultrasonics* **5**, 213.

Anonymous (1975). *SONIX* **1**, 1, 28.

Arenberg, D. L. (1948). *J. Acoust. Soc. Am.* **20**, 1, 1.

Balamuth, L. (1934). *Phys. Rev.* **45**, 715.

Balamuth, L. (1948). Method of Abrading. British Patent No. 602801 (filed April, 1945).

Balamuth, L. (1963). *Sound* **2**, 2, 15.

Bär, R., and Meyer, E. (1933). *Helv. Phys. Acta* **6**, 242.

Barone, A. (1962). *In* "Handbuch der Physik" (J. Flugge, ed.), Vol. XI/2, p. 75. Springer-Verlag, Berlin and New York.

Barton, E. H. (1908). "A Text-Book on Sound." Macmillan, New York.

Bergmann, L. (1939). "Ultrasonics." Wiley, New York.

Bergmann, L. (1954). "Der Ultraschall." Hirzel, Stuttgart.

Berlincourt, D. A., Curran, D. R., and Jaffe, H. (1964). *In* "Physical Acoustics," Vol. 1A (W. P. Mason, ed.). Academic Press, New York.

Berman, A., and Guthrie, A. N. (1972). *J. Acoust. Soc. Am.* **51**, 3(2), 994.

Beyer, R. T. (1974). Nonlinear Acoustics. Naval Ship Systems Command, Department of the Navy.

Biquard, P. (1931). *C. R. Acad. Sci. Paris* **193**, 226.

Biquard, P. (1972). *Ultrasonics* September, 213.

Blaha, F., and Langenecker, B. (1955). *Z. Naturwiss.* **20**, 556.

Blake, F. G., Jr. (1949). Technical Memo No. 12. Acoustics Research Laboratory, Harvard Univ., Cambridge, Massachusetts.

Blake, R. F. (1914–1915). Submarine signaling—the protection of shipping by a wall of sound and other uses of the submarine telegraph oscillator. Annual Report of the Smithsonian Institution.

Bondy, C., and Söllner, K. (1935). *Trans. Faraday Soc.* **31**, 835.

Bouyoucos, J. V. (1975). *J. Acoust. Soc. Am.* **57**, 6(1), 1241.

Boyle, R. W. (1928). *Sci. Prog.* **23**, 75.

Brillouin, L. (1914). *C. R. Acad. Sci. Paris* **158**, 1331. Translated in Lindsay (1974).

Brillouin, L. (1922). *Ann. Phys. (Leipzig)* **17**, 88.

Brockelsby, C. F., Palfreeman, J. S., and Gibson, R. W. (1963). "Ultrasonic Delay Lines." Iliffe, London.

Brown, B., and Goodman, J. E. (1965). "High-intensity Ultrasonics." Iliffe, London.

Brown, B., and Gordon, D. (eds.) (1967). "Ultrasonic Techniques in Biology and Medicine." Thomas, Springfield, Illinois.

Butterworth, S. (1915). *Proc. Phys. Soc. London* **27**, 410.

Cady, W. G. (1946). "Piezoelectricity." McGraw-Hill, New York. Printed in revised form by Dover, New York (1964).
Cady, W. G. (1963). *Sound* **2**, 1, 46.
Carlin, B. (1949). "Ultrasonics." McGraw-Hill, New York.
Carlin, B. (1964). *In* "Physical Acoustics," Vol. 1B (W. P. Mason, ed.). Academic Press, New York.
Carwile, P. B. (1953). U.S. Patent No. 2,633,894. Assigned to Raytheon Mfg. Co.
Chaffee, E. L. (1933). "Theory of Thermionic Vacuum Tubes." McGraw Hill, New York.
Chipman, R. A. (1965). *Sci. Am.* **212**, 2, 93.
CNRS (Centre National de La Recherche Scientific) (1950). "Oeuvres Scientific de Paul Langevin." CNRS, Paris.
Crawford, A. E. (1955). "Ultrasonic Engineering." Butterworths, London.
Curie, J.-P., and Curie, P. (1880). *C. R. Acad. Sci. Paris* **91**, 294. Translated in Lindsay (1973).
Curie, M. (1923). "Pierre Curie." Macmillan, New York.
Davis, H. (1948). *J. Acoust. Soc. Am.* **20**, 5, 605.
Debye, P., and Sears, F. W. (1932). *Proc. Natl. Acad. Sci. U.S.* **18**, 409.
Dehlinger, P. *et al.* (1950). A Study of Rock Properties Affecting Vibrating Drilling of Oil Wells and of Electromechanical Transducers for Drilling Oil Wells. Series of Battelle Memorial Institute Technical Reports.
Desch, C. H., Sproule, D. O., and Dawson, W. J. (1947). *Weld. J.* **26**, Suppl. 1s.
Despretz, C. (1845). *C. R. Acad. Sci. Paris* **20**, 1214.
Dieckmann, M. (1908). *Ann. Phys. (Leipzig)* **27**, 1066.
Drysdale, C. V. (1920). *J. Inst. Elec. Eng.* **58**, 572.
Duddell, W. (1900). *Electrician* **46**, 269.
Duff, A. W. (1898). *Phys. Rev.* **6**, No. 3, 129.
Dvorak, V. (1876). *Ann. Phys. Chem.* **157**, 42.
Dye, D. W. (1926). *Proc. Philos. Soc.* **38**, 399.
Eckart, C. (1948). *Phys. Rev.* **73**, 68. Reprinted in Lindsay (1974).
Edelmann, M.Th. (1900). *Ann. Phys. (Leipzig)* **2**, 469.
Edler, I., and Hertz, C. H. (1977). *J. Clin. Ultrasound* **5**, 5, 352.
Einstein, A. (1920). *Sitzungsber. Preuss. Akad. Wiss. Phys. Math. Kl.* **24**, 380. Translated in Lindsay (1974).
Elwell, C. F. (1923). "The Poulsen Arc Generator." Van Nostrand-Reinhold, Princeton, New Jersey.
Ensminger, D. (1973). "Ultrasonics: The Low and High-Intensity Applications." Dekker, New York.
Ensminger, D. (1978). Private communication.
Enz, U. (1955). *Tech. Mitt. PTT* **33**, 6, 209.
Erdman, D. C. (1952). Ultrasonic Flaw Detector. U.S. Patent No. 2,593,865.
Erwin, W. S., and Rassweiler, G. M. (1944). Supersonic Measuring Means. U.S. Patent No. 2,431,233.
Esche, R. (1952). *Acustica* **2**, AB 208.
Eve, A. S., and Creasey, C. H. (1945). "Life and Work of John Tyndall." Macmillan, New York.
Fagen, M. D. (ed.) (1975). A History of Engineering and Science in the Bell System; The Early Years (1875–1925). Bell Telephone Laboratories.
Faraday, M. (1831). *Philos. Trans. R. Soc. London* **121**, 299. Reprinted in Lindsay (1974).
Fay, H. J. W. (1963). Submarine Signal Log. Raytheon Company, Submarine Signal Division, Portsmouth, Rhode Island.

Federal Reporter (1960). Aluminum Co. of America vs. Sperry Products, Inc., Cite 285 F.2d 911 (1960).
Federal Supplement (1959). Sperry Products, Inc. vs. Aluminum Co. of America, Cite 171 F. Supp. 901.
Fessenden, H. M. (1940). "Fessenden: Builder of Tommorrows." Coward-McCann, New York.
Firestone, F. A. (1942). Flaw Detecting Device and Measuring Instrument. U.S. Patent No. 2,280,226 (filed May, 1940).
Firestone, F. A. (1948). *Non-Destr. Test.* Fall Number, p. 5.
Flynn, H. G. (1964), *In* "Physical Acoustics," Vol. 1B. (W. P. Mason, ed.). Academic Press, New York.
Foley, A. L., and Souder, W. H. (1912). *Phys. Rev.* **35**, 373.
Fox, F. E. (1951). *J. Acoust. Soc. Am.* **23**, 2, 233.
Frederick, J. (1965). "Ultrasonic Engineering." Wiley, Inc.
Gaines, N. (1932). *Physics* **3**, 5, 209.
Galambos, R. (1942). *Isis* **34**, 132.
Galton, F. (1883a). *Nature (London)* **27**, 491.
Galton, F. (1883b). "Inquiries Into Human Faculty and Its Development." Dent, London. Republished (1907) by Richard Clay and Sons, London and Bungay.
Gericke, O. R., and Grubinskas, R. C. (1969). *J. Acoust. Soc. Am.* **45**, 4, 872.
Giacomini, A., and Giacomini, A. B. (1939). *Ric. Sci. Prog. Tec.* **10**, Part I, 921.
Giebe, E., and Scheibe, A. (1925). *Z. Phys* **33**, 335.
Gillispie, C. C. (ed.-in-chief) (1971). "Dictionary of Scientific Biography." Scribner's, New York.
Gordon, D. (ed.) (1964). "Ultrasound as a Diagnostic and Surgical Tool." Williams and Wilkins, Baltimore, Maryland.
Gorskhkov, S. I., Gorbunov, O. N., and Antropov, G. A. (1966). "Biological Effect of Ultrasound" (Translation). U.S. Department of Commerce, *JPRS* **36**, 924.
Graff, K. F. (1975). *Ultrasonics* May, 103.
Graff, K. F. (1977). *Ultrasonics* March, 75.
Griffin, D. R. (1958). "Listening in the Dark." Yale Univ. Press, New Haven, Connecticut.
Grossmann, E. (1932). *Phys. Z.* **33**, 202.
Hartmann, J. (1922). *Phys. Rev.* (2) **20**, 719.
Hartmann, J. (1939). "The Acoustic Air-Jet Generator," Bulletin No. 4. Academy of Technical Science, Copenhagen.
Harvey, E. N. (1929). *Am. J. Physiol.* **91**, 284.
Harvey, E. N. (1930). *Biol. Bull.* **59**, 306.
Harvey, E. N., and Loomis, A. L. (1928). *Nature (London)* **121**, 622.
Harvey, E. N., and Loomis, A. L. (1929). *J. Bacteriol.* **17**, 373.
Harvey, E. N., Harvey, E. B., and Loomis, A. L. (1928). *Biol. Bull. Mar. Biol. Lab.* **55**, 459.
Harvey, E. N., Barnes, D. K., McElroy, W. D., Whiteley, A. H., Pease, D. C., and Cooper, K. W. (1944). *J. Cell. Comp. Physiol.* **24**, 1.
Hastings, C. H., and Carter, S. W. (1949). "Inspection, Processing, and Manufacturing Control of Metals by Ultrasonic Methods," Special Tech. Publ. No. 101. American Society for Testing Materials, Philadelphia, Pennsylvania.
Hayes, H. C. (1920). *Proc. Am. Philos. Soc.* **59**, 1.
Hayes, H. C. (1962). *Sound* **1**, 5, 47.
Helmoltz, H. L. F. (1885). "On the Sensations of Tone," 2nd English ed. Reprinted (1954) by Dover Publ., New York.

Herrick, J. F., and Krusen, F. H. (1954). *IRE Trans. Ultrasonic. Eng.* **UE-7**, 4.
Herzfeld, K. F. (1966). *J. Acoust. Soc. Am.* **39**, 5, Part 1, 813.
Herzfeld, K. F., and Rice, F. O. (1928). *Phys. Rev.* **31**, 691.
Hewlitt, A. (1922). *Phys. Rev.* **19** (2), 52.
Hezlet, A. (1975). "The Electron and Sea Power." Davies, London.
Hiedemann, E. (1935). Ultraschall, *In* "Ergebnisse der Exakten Naturwissenschaften," Vol. 14. Springer-Verlag, Berlin and New York.
Hiedemann, E. (1939). "Grundlagen und Ergebnisse der Ultraschallforschung." de Gruyter, Berlin.
Hiedemann, E. (1954). *J. Acoust. Soc. Am.* **26**, No. 5, 831.
Holtzmann, M. (1925). *Phys. Z.* **26**, 147.
Holtzmark, J., Johnson, I., Sikkeland, T., and Skavlem, S. (1954). *J. Acoust. Soc. Am.* **26**, 1, 26.
Hopwood, F. L. (1929). *J. Sci. Instrum.* **6**, 34.
Hopwood, F. L. (1931). Ultrasonics: Some properties of inaudible sound. *Nature (London)* **128**, 748.
Horton, J. W. (1959). Fundamentals of Sonar. United States Naval Institute, Annapolis, Maryland.
Howeth, L. S. (1963). "History of Communications-Electronics in the United States Navy." US Govt. Printing Office, Washington, D.C.
Hubbard, J. C. (1932). *J. Acoust. Soc. Am.* **4**, 2, 99–107.
Hubbard, J. C. (1940). *Am. J. Phys.* **8**, 207.
Hubbard, J. C., and Loomis, A. L. (1927). *Nature (London)* **120**, 189.
Hueter, T. F., and Bolt, R. H. (1955). "Sonics." Wiley, New York.
Hunt, F. V. (1954). "Electroacoustics." Harvard Univ. Press, Cambridge, Massachusetts.
Hunt, F. V. (1972). *J. Acoust. Soc. Am.* **51**, 3 (Part 2), 992.
Janovsky, W., and Pohlman, R. (1948). *Z. Angew. Phys.* **1**, 222.
J. B. (1925). *Nature (London)* **115**, 2897, 689.
Jeans, J. H. (1904). "The Dynamical Theory of Gases." Cambridge Univ. Press, London and New York.
Johnson, C. H. (1929). *J. Physiol.* **67**, 356.
Jones, R. C. (1946). *J. Acoust. Soc. Am.* **18**, 371.
Joule, J. P. (1847). *Philos. Mag.* **30**, 76, 225. Also Lindsay (1973).
Keller, A. C. (1947). *AIEE Trans.* **66**, 1217.
Kelly, E. (ed.) (1957). "Ultrasound in Biology and Medicine." American Institute of Biological Sciences, Washington, D.C.
Kelly, E. (ed.) (1965). "Ultrasonic Energy, Biological Investigations and Medical Applications." Univ. of Illinois Press, Urbana.
Kingery, R. A., Berg, R. D., and Schillinger, E. H. (1967). "Men and Ideas in Engineering: Twelve Histories from Illinois." Univ. of Illinois Press, Urbana.
Kirchhoff, G. (1868). *Ann. Phys. Chem.* **134**, 177. Translated in Lindsay (1974).
Klein, E. (1948). *J. Acoust. Soc. Am.* **20**, 5, 601.
Klein, E. (1967). ONR Rep. ACR-135, Office of Naval Research, Washington, D.C. See also Klein (1968).
Klein, E. (1968). *J. Acoust. Soc. Am.* **43**, 5, 931.
Knapp, R. T., and Hollander, A. (1948). *Trans. ASME* **70**, 419.
Kneser, H. O. (1931). *Ann. Phys. (Leipzig)* **11**, 777.
Kneser, H. O. (1938). *Ann. Phys. (Leipzig)* **32**, 277.
Kock, W. E. (1973). *In* "Physical Acoustics," Vol. 10 (W. P. Mason and R. N. Thurston, eds.). Academic Press, New York.

Koenig, R. (1899). *Ann. Phys. Chem.* **69**, 626, 721.

Krasil'nikov, V. A. (1963). "Sound and Ultrasound Waves," 3rd ed. Translated from the Russian. Published for the National Science Foundation and the National Aeronautics and Space Administration by the Israel Program for Scientific Publ.

Krautkramer, J., and Krautkramer, H. (1969). "Ultrasonic Testing of Materials." Springer-Verlag, Berlin and New York.

Kruse, F. (1938). On the Testing of Materials by Supersonics. Ph.D. Thesis, Hannover Technische Hochschule.

Kruse, F. (1939). *J. Acoust. Soc. Am.* **11**, 367.

Kundt, A. (1866). *Ann. Phys. Chem.* **127**, 4, 497.

Lamson, H. W. (1930). *J. Acoust. Soc. Am.* **1**, 403.

Lasky, M. (1974a). *U.S. Navy J. Underwater Acoust.* **24**, 3, 363.

Lasky, M. (1974b). *U.S. Navy J. Underwater Acoust.* **24**, 4, 597.

Lasky, M. (1975a). *U.S. Naval J. Underwater Acoust.* **25**, 3, 567.

Lasky, M. (1975b). *U.S. Navy J. Underwater Acoust.* **25**, 4, 885.

Lasky, M. (1977). *J. Acoust. Soc. Am.* **61**, 2, 283.

Lebedev, P. (1911). *Ann. Phys. (Leipzig)* **35**, 171.

LeConte, J. (1858). *Philos. Mag.* **25**, 235. Also reprinted in Tyndall (1903).

LeConte, J. (1883). *Nature (London)* **27**, 54.

Lindsay, R. B. (1966). *J. Acoust. Soc. Am.* **39**, 630. Reprinted in Lindsay (1974).

Lindsay, R. B. (1967). *J. Acoust. Soc. Am.* **41**, 4(2), 905.

Lindsay, R. B. (ed.) (1973). "Acoustics: Historical and Philosophical Development." Dowden, Hutchinson, and Ross, Stroudsburg, Pennsylvania.

Lindsay, R. B. (ed.) (1974). "Physical Acoustics." Dowden, Hutchinson, and Ross, Stroudsburg, Pennsylvania.

Litsios, J. (1963). *IEEE Trans. Ultrason. Eng.* September, 91.

Lord, A. E., Jr. (1975). *In* "Physical Acoustics," Vol II, (W. P. Mason and R. N. Thurston, eds.). Academic Press, New York.

Lozinskii, M. G., and Rozenberg, L. D. (1950). A Method of Concentrating Ultrasonic Energy. USSR Patent 85,196.

Lucas, R., and Biquard, P. (1932). *C. R. Acad. Sci. Paris* **194**, 2132.

Lynnworth, L. C. (1975). *IEEE Trans. Sonics Ultrason.* **SU-22** 2, 71.

McMaster, R. C. (ed.) (1959). "Nondestructive Testing Handbook." Ronald Press, New York.

McMaster, R. C., and Wenk, S. (1951). "A Basic Guide for Managements Choice of Non-destructive Tests," Special Technical Publ. No. 112. Amer. Society for Testing Materials, Philadelphia, Pennsylvania.

Mahoux, G. (1930). *C.R. Acad. Sci. Paris* **191**, 1328.

Marinesco, N. (1937). "Actualités Scientifiques et Industrielles," No. 522, 523. Hermann, Paris.

Mason, W. P. (1942). "Electromechanical Transducers and Wave Filters." Van Nostrand-Reinhold, Princeton, New Jersey.

Mason, W. P. (1950a). "Piezoelectric Crystals and Their Applications to Ultrasonics." Van Nostrand-Reinhold, Princeton, New Jersey.

Mason, W. P. (1950b). Method of Obtaining High Velocity with Crystals. U.S. Patent 2,514,080 (filed July 1950).

Mason, W. P. (1958). "Physical Acoustics and the Properties of Solids." Van Nostrand-Reinhold, Princeton New Jersey.

Mason, W. P. (1971). *J. Acoust. Soc. Am.* **50**, 5(2), 1281.

Mason, W. P., and Wick, R. F. (1951a). Mechanical Impedance Transformers. U.S. Patent No. 2,573,168 (filed October 1951).

Mason, W. P., and Wick, R. F. (1951b). *J. Acoust. Soc. Am.* **23**, 2, 209.

Mattiat, O. (1954). *J. Acoust. Soc. Am.* **26**, 2, 241.
May, J. E. Jr. (1964). *In* "Physical Acoustics," Vol 1A (W. P. Mason, ed.). Academic Press, New York.
Meissner, A. (1926). *Z. Tech. Phys.* **7**, 585.
Meyer, F. R. (1973). Engineering Feasibility Study of Ultrasonic Applications for Aircraft Manufacture, Research Report 73-15. Aeroprojects, West Chester, Pennsylvania.
Michael, P. L., Kerlin, R. L., Bienvenue, G. R., and Prout, J. H. (1974). An Evaluation of Industrial Acoustic Radiation Above 10 kHz. Final Report on Contract No. HSM-99-72-125. U. S. Department of Health, Education and Welfare.
Miller, D. C. (1935). "Anecdotal History of the Science of Sound." Macmillan, New York.
Miller, H. B. (1963). *J. Acoust. Soc. Am.* **35**, 9, 1455. Reprinted in Albers (1972).
Mühlhäuser, O. (1931). Verfahren zur Zustandsbestimmung von Werkstoffen, besonders zur Ermittlung von Fehlern darin, DRP Nr. 569 598.
Neklepajew, N. (1911). *Ann. Phys. (Leipzig)* **35**, 175.
Neppiras, E. A. (1956). *Metalwork. Product.* **100**, 1283.
Neppiras, E. A. (1968). *J. Sound Vib.* **8**, 3, 408.
Neppiras, E. A. (1973). *Ultrason. Int. Conf. Proc.* IPC Science and Technology Press.
Neppiras, E. A., and Noltingk, B. E. (1951). *Proc. Phys. Soc. London* **B64**, 1032.
Neumann, E. P., and Norton, J. L. (1951). *In* "Ultrasonics—Two Symposia" (F. J. Van Antwerpen, ed.), Chemical Engineering Progress Symposium Series, No. 1, Vol. 47. Amer. Institute of Chemical Engineers, New York.
Noltingk, B. E. (1962). *In* "Handbuch der Physik," (S. Flügge, ed.), Vol. XI/2 Springer-Verlag, Berlin and New York.
Noltingk, B. E., and Neppiras, E. A. (1950). *Proc. Phys. Soc. London* **B63**, 674.
Nyborg, W. L. (1953). *J. Acoust. Soc. Am.* **25**, 1, 68.
Nyborg, W. L. (1965). *In* "Physical Acoustics," Vol. 2B (W. P. Mason, ed.). Academic Press, New York.
N.Y.T. (1975). Alfred L. Loomis obituary notice, New York Times, August 12, p. 32.
Palaiologos, K. (1923). *Z. Phys.* **12**, 375.
Pearson, K. (1914). "The Life, Letters and Labours of Francis Galton." Cambridge Univ. Press, London and New York.
Pielemeier, W. H. (1929). *Phys. Rev.* **34**, 8, 1184.
Pierce, G. W. (1925). *Proc. Am. Acad. Arts Sci.* **60**, 271. Also Lindsay (1974).
Pierce, G. W. (1928). *Proc. Am. Acad. Arts Sci.* **63**, 1.
Pierce, G. W. (1948). "The Songs of Insects." Harvard Univ. Press, Cambridge, Massachusetts.
Pierce, G. W., and Griffin, D. R. (1938). *J. Mammal.* **19**, 454.
Pohlman, R. (1939a). *Z. Phys.* **113**, 697.
Pohlman, R. (1939b) *Verhandl. Deut. Phys. Ges.* **20**, 52.
Posakony, G. J. (1976). *Mater. Eval.* December, 10A.
Preyer, W. (1876). *Physiol. Abhandl. Jena.*
P.U.A. (1950). Underwater Acoustics Research. Prepared by the Panel on Underwater Acoustics, Committee on Undersea Warfare, National Research Council, Washington, D.C.
Quimby, S. L. (1925). *Phys. Rev.* **25**, 6.
Raman, C. V., and Nath, R. S. N. (1935). *Proc. Indian Acad. Sci. A* **2**, 406, 413.
Raman, C. V., and Nath, R. S. N. (1936). *Proc. Indian Acad Sci. A* **3**, 75, 119, 495.
Rayleigh, Lord (1882). *Philos. Mag.* **14**, 186. Also Rayleigh (1964, Vol. II, p. 132).
Rayleigh, Lord (1883a). *Philos. Mag.* **16**, 50. Also Rayleigh (1964, Vol. II, p. 212).
Rayleigh, Lord (1883b). *Philos. Trans. R. Soc. London* **175**, 1. Also Rayleigh (1964, Vol. II, p. 239).

Rayleigh, Lord (1885). *London Math. Soc. Proc.* **17**, 4. Also Rayleigh (1964, Vol. II, p. 441).
Rayleigh, Lord (1892). *Philos. Trans.* **183A**, 1. Also Rayleigh (1964, Vol. III, p. 558).
Rayleigh, Lord (1896). "The Theory of Sound," 2nd ed. Macmillan, New York. Reprinted by Dover, New York.
Rayleigh, Lord (1899). *Philos. Mag.* **47**, 308. Also Rayleigh (1964, Vol. 4, p. 376).
Rayleigh, Lord (1902). *Philos. Mag.* **3**, 338. Also Rayleigh (1964, Vol. V, p. 41).
Rayleigh, Lord (1905). *Philos. Mag.* **10**, 364. Also Rayleigh (1964, Vol. V, p. 262).
Rayleigh, Lord (1910). *Proc. R. Soc. London Ser. A* **84**, 247. Also Rayleigh (1964, Vol. V, 573).
Rayleigh, Lord (1917). *Philos. Mag.* **34**, 94. Also Rayleigh (1964, Vol. VI, p. 504).
Rayleigh, Lord (1964). "Scientific Papers by Lord Rayleigh (John William Strutt)." Dover, New York.
Rich, D. L., and Pielemeyer, W. H. (1925). *Phys. Rev.* **25**, 117.
Richards, W. T. (1929). *J. Am. Chem. Soc.* **51**, 1724.
Richards, W. T. (1938). *J. Appl. Phys.* **9**, 298.
Richards, W. T. (1939). *Rev. Mod. Phys.* **11**, 36.
Richards, W. T., and Loomis, A. L. (1927). *J. Am. Chem. Soc.* **49**, 3086.
Richards, W. T., and Reid, J. A. (1934). *J. Chem. Phys.* **2**, 4, 193.
Richardson, E. G. (1952). "Ultrasonic Physics." Elsevier, Amsterdam.
Richardson, E. G. (ed.) (1957). "Technical Aspects of Sound—II." Elsevier, Amsterdam.
Roberts, S. (1947). *J. Acoust. Soc. Am.* **71**, 12, 890.
Robison, J. (1801). "Encyclopedia Brittanica," 3rd ed., Vol. 2 of Supplement.
Rozenberg, L. D. (ed.) (1973). "Physical Principles of Ultrasonic Technology," Vol. 2. Plenum, New York.
Rozenberg, L. D., Kazantsev, V. F., Makarov, L. O., and Yakhimovich, D. F. (1964). "Ultrasonic Cutting." Consultants Bureau, New York.
Rytov, S. (1938). "Actualites Scientifiques et Industrielles," No. 613. Hermann, Paris.
Sales, G., and Pye, D. (1974). "Ultrasonic Communication by Animals." Chapman and Hall, London.
Saunders, F. A., and Hunt, F. V. (1959). "Biographical Memoirs," National Academy of Sciences, Vol. 33. Columbia Univ. Press, New York.
Savart, F. (1830). *Ann. Phys. Chem.* **20**, 290. Translated in Lindsay (1974).
Schaefer, Cl., and Bergmann, L. (1934). *Sitz. Ber. Berliner Akad. Wiss. Phys. Math.* **10**, 155.
Schaefer, K. L. (1924). *Beitr. Anat.* **20**, 142.
Schlichting, H. (1932). *Phys. Z.* **33**, 327.
Schmidt, F. O., (1929). *Protoplasma* **7**, 332.
Schmidt, F. O., and Uhlemeyer, B. (1930). *Proc. Soc. Exp. Biol. Med.* **27**, 626.
Schmidt, F. O. Johnson, C. H., and Olson, A. R. (1928). *Proc. Soc. Exp. Biol. Med.* **25**, 728.
Schmidt, F. O., Johnson, C. H., and Olson, A. R. (1929). *J. Am. Chem. Soc.* **51**, 370.
Schultze, F. A. (1907). *Ann. Phys. (Leipzig)* **24**, 785.
Schuster, A. (1920–1921). *Proc. R. Soc. London Ser. A* **98**, i.
Seabrook, W. (1941). "Doctor Wood." Harcourt, New York.
Skraiber, D. S. (1939). *Zavod. Lab.* **8**, 816.
Skudrzyk, E. (1954). "Die Grundlagen der Akustik." Springer-Verlag, Berlin and New York.
Sokolov, S. (1929). *Elek. Nachr. Tech.* **6**, 454.
Sokolov, S. (1935a). *Phys. Z.* **36**, 142.
Sokolov, S. (1935b). *Tech. Phys. (USSR)* **2**, 6, 522.
Sokolov, S. (1939). Means for Indicating Flaws in Materials, U.S. Patent 2, 164, 125 (filed 1937).
S.P.T. (1901). *Nature (London)* **64**, 630.
St. Clair, H. W. (1940). U.S. Patent No. 2, 215, 484.

St. Clair, H. W. (1941). *Rev. Sci. Instrum.* **12**, 250.
Stegeman, G. I. (1976). *IEEE Trans. Sonics Ultrason. SU-23*, 1, 33.
Stephens, R. W. B. (1975). *Ultrason. Int. Conf. Proc.* IPC Science and Technology Press.
Stokes, C. A., and Vivian, J. E. (1951). *In* "Ultrasonics—Two Symposia," Chemical Engineering Progress Symposium Series, No. 1, Vol. 47. Amer. Institute of Chemical Engineers, New York.
Stokes, G. G (1845). *Trans. Cambridge Philos. Soc.* **8**, 287. Reprinted in part in Lindsay (1973).
Strutt, R. J. (Fourth Baron Rayleigh) (1924). "John William Strutt, Third Baron Rayleigh." Edward Arnold, London.
Taton, R. (ed.) (1964). "The Beginnings of Modern Science" (from 1458 to 1800). Basic Books, New York.
Tawil, E. P. (1926). *C. R. Acad. Sci. Paris* **183**, 1099.
Tawil, E. P. (1930). *C. R. Acad. Sci. Paris* **191**, 92, 168.
Toepler, A. (1867a). *Ann. Phys. (Leipzig)* **131**, 33.
Toepler, A. (1867b). *Philos. Mag.* **33**, 16.
Toepler, M. (1908). *Ann. Phys. (Leipzig)* **27**, 1043.
Truesdell, C. (1953). *J. Ration. Mech. Anal.* **2**, 4, 643.
Tyndall, J. (1869). "Sound," 2nd ed. Longmann, Green, London.
Tyndall, J. (1903). "Sound," 3rd ed. Appleton, New York. Reprinted by Greenwood Press, New York (1969).
Van der Burgt, C. M. (1973). Private communication.
Van Dyke, K. S. (1925). *Phys. Rev.* **25**, 895.
Vigoureux, P. (1931). "Quartz Resonators and Oscillators." HM Stationery Office, London.
Vigoureux, P. (1950). "Ultrasonics." Chapman and Hall, London.
Vincent, J. H. (1929). Proc. Phys. Soc. London **41**, 476.
Vonnegut, B. (1954). *J. Acoust. Soc. Am.* **26**, 18.
Webster, A. G. (1919). *Proc. Natl. Acad. Sci. U.S.* **5**, 275. Also Lindsay (1974).
Webster, E. (1963). *Ultrasonics* January–March, p. 39.
Weissler, A. (1953). *J. Acoust. Soc. Am.* **25** (4), 651.
Wente, E. C. (1917). *Phys. Rev.* **10**, 39.
Westervelt, P. J. (1953). *J. Acoust. Soc. Am.* **25**, 1, 60.
Whewell, W. (1857). "History of the Inductive Sciences: History of Acoustics." Reprinted as "The Historical and Philosophical Works of William Whewell" (G. Buchdahl and L. L. Laudan, eds.) Vol. III. Frank Cass, London, 1967.
Whitcroft, T. H. (1943). *U.S. Naval Inst. Proc.* **69**, 216.
White, D. N. (1976). "Ultrasound in Medical Diagnosis." Ultramedison, Kingston, Ontario.
Willard, G. W. (1953). *J. Acoust. Soc. Am.* **25**, 669.
Williams, O. B., and Gaines, N. (1930). *J. Inf. Disc.* **47**, 485.
Willrich, H. O. (1950). *Welding* **18**, 61.
Winder, A. A. (1975). *IEEE Trans. Sonics Ultrason.* **SU-22**, 5, 291.
Wollaston, W. H. (1820). *Philos. Trans. R. Soc. London Part* I **110**, 306.
Woloshyn, I. (1963). *Ultrasonics* January–March, p. 14.
Wood, A. B. (1940). "Acoustics." Blackie, London.
Wood, A. B. (1960). "Acoustics," 2nd ed., Blackie, London and Dover, New York (1966).
Wood, A. B. (1962). *Sound* **1**, 3, 8.
Wood, R. W. (1899). *Philos. Mag.* **48**, 218.
Wood, R. W. (1905). "Physical Optics." Macmillan, New York.
Wood, R. W. (1939). "Supersonics, The Science of Inaudible Sounds." Brown Univ., Providence, Rhode Island.
Wood, R. W., and Loomis, A. L. (1927). *Philos. Mag.* **4**, 22, 417.
Yeager, E., and Hovorka, F. (1953). *J. Acoust. Soc. Am.* **25**, 3, 443.

—2—

Circuit-Model Analysis and Design of Interdigital Transducers for Surface Acoustic Wave Devices

W. RICHARD SMITH

Hughes Aircraft Company,
Ground Systems Group, Fullerton, California

I. Introduction 100
II. General Description of Interdigital Transducers as Filter Elements 102
A. SAW Delay Line with Interdigital Transducers 102
B. Transversal Filter Background 104
C. Comparison of the Interdigital Transducer to the Versatile Transversal Filter 106
D. Relation of SAW Filter and Transducer Designs 109
III. Relation of Transducer Response to Geometry 110
A. Array and Element Factors 111
B. Delta-Function Model 113
C. Impulse Model 114
D. Design of the Transducer Geometry for a Required Array Factor Response 115
E. Source-Withdrawal Weighting 120
IV. "Second-Order" Effects and Their Compensation 122
A. Diffraction and "Beam Steering" 123
B. Wave-Front Distortion from Apodization; "Dummy" Electrodes 124
C. Electrode "Mass/Electrical Loading"; Double Electrodes 125
D. Regeneration Distortion 127
E. Triple-Transit Echoes 128
F. Bulk or Plate Waves 129
G. Electromagnetic Feedthrough and Ground Loops 130
H. Electrode Resistivity and Acoustic Propagation Loss 131
V. Three-Port Circuit Model Description of Interdigital Transducers 132
A. General Capabilities of Three-Port Circuit Models 132
B. Crossed-Field Model 137
C. Generalized Circuit Model 142
D. Computer Simulation of SAW Filters 146
E. Relation of the Circuit Model Transfer Function T_{13} to Simpler Array-Factor Analyses 154

PHYSICAL ACOUSTICS, VOL XV

Copyright © 1981 by Academic Press, Inc
All rights of reproduction in any form reserved
ISBN 0-12-477915-8

VI. Matching-Circuit-Controlled Trade-offs among Insertion Loss Bandwidth, Triple-Transit Suppression, and Voltage Standing-Wave Ratio 160
A. Fundamental Limitations Imposed on Filter Insertion Loss by the Required Bandwidth 161
B. Acoustic Reflection Coefficient T_{11} 163
C. Circuit Factors for Simple Matching Circuits 164
D. The Direct-Connection or "Untuned" Transducer (Matching Circuit "Absent") 165
E. Shunt- and Series-Tuned Transducers 166
F. Choice Between Series and Shunt Tuning; Other Matching Circuits 177
VII. Specific Transducer Geometries (Array Factors) for Typical Filter Applications ... 177
A. Nondispersive Bandpass Filters 177
B. Dispersive Filter Design 180
C. Tapped Delay Lines 183
D. Unapodized Dispersive Transducers in Nondispersive Filters, RAC's, Convolvers, and Acousto-optic Devices 184
References 187

I. Introduction

This article is concerned with the analysis and design of interdigital transducers (IDT's) and surface acoustic wave (SAW) filters using them. Some contemporary SAW signal processing devices contain integrated components in addition to the IDT, e.g., etched groove arrays in reflective-array compressors (RAC's) and metal plates in nonlinear convolvers. Nevertheless, virtually all SAW devices require IDT's for launching and/or detection of surface waves, and it is for this reason that the IDT is commonly regarded as the "cornerstone" of SAW technology. Thus, although the design of nontransducer components such as reflective arrays is not discussed in this article, the transducer analysis and design principles are still useful for devices such as RAC's and convolvers. A large fraction (probably a majority) of today's SAW filters are composed solely of IDT's and their associated matching circuits. For these devices, we *do* consider the total filter design problem, specifically in Section II, D and also in Sections IV and VII.

The rather voluminous literature on interdigital transducers contains a variety of analytical approaches ranging from simple summations of plane waves through three-port "equivalent" circuit models and (Rayleigh) normal mode analysis to a complete Green's function calculation of the total (SAW and bulk wave) acoustic wave fields radiated by a transducer. In this article we shall consider only those models that treat the acoustic fields of an interdigital transducer as a single (Rayleigh) mode whose amplitude is conveniently represented as a sum of weighted, scalar plane-wave contributions from the various electrodes. In fact, one of the most important objectives is to show that such a sum or *array factor* is common

to models as simple as the delta-function model and as complex as a generalized three-port circuit model capable of describing harmonic operation of transducers with arbitrary electrode polarity sequences and metalization ratios. The reason for using array-factor characterization as the criterion for limiting this article's scope is that the array factor provides the basis for first-order *design* of transducers of arbitrary geometry.

In this article we shall place the heaviest emphasis on three-port circuit models. This is because they combine the array-factor design principles with a concise and straightforward way of obtaining true (i.e., unnormalized) numerical values for all critical parameters. The circuit models give direct expressions for transducer insertion loss and phase plus analysis and design of the matching circuits, yielding important trade-offs of insertion loss and spurious signals. The circuit models also cast the transducer networks and transfer function expressions in a form that explicitly displays the method of designing *two* transducers for a required filter transfer function. The author feels that these rather extensive capabilities of circuit models justify omission of more rigorous physical analyses, no one of which provides as much information.

TABLE I
ABRIDGED LIST OF IMPORTANT METHODS OF IDT ANALYSIS NOT DISCUSSED IN THIS ARTICLE

Author(s) and Date	Description
1. Auld (1973)	"Acoustic Fields and Waves in Solids" (book on general formalism)
2. Coquin and Tiersten (1967)	Tensor Field Analysis—Periodic Transducers on Quartz—Impedance and Efficiency
3. Joshi and White (1968)	Tensor Field Analysis Giving One-Port Electrical Circuit and Insertion Loss
4. Auld and Kino (1971)	Normal Mode Theory—Transducer Impedance Calculations for Rayleigh Wave (partially justifies circuit models)
5. Szabo and Slobodnik (1973, 1974)	Effect of Diffraction on SAW Device Design
6. Milsom *et al.* (1977)	Green's Function Approach for Surface and Bulk Waves
7. Wagers (1976, 1977)	Plate Mode Coupling in SAW Devices
8. Wagers (1979)	Two-dimensional Electrostatic Solution Transverse Effects in Finite-Width Transducers

Since this scope and philosophy necessarily leaves untreated some alternative analytical approaches that play important roles in the literature, it is necessary to mention here some of the important literature that will not be discussed. Szabo *et al.* (1979) have recently published a summary of various transducer models, discussing the choice of a model that is best suited to the user's needs. Table I is an abridged list of additional articles that provide transducer analysis that is different from, and complementary to, the models discussed herein. Omission of many other papers from this list is not to be construed as a judgment of relative unimportance, but the list is intended to sample the most important general approaches.

It is only for the sake of brevity that we have also omitted discussion of *unidirectional* transducers (Smith *et al.*, 1969b; Hartmann *et al.*, 1972; Krimholtz *et al.*, 1974; Yamanouchi *et al.*, 1975), which are actually quite well described by circuit-model analysis but require the use of networks with at least four ports.

II. General Description of Interdigital Transducers as Filter Elements

A. SAW Delay Line with Interdigital Transducers

The simplest SAW device is the nondispersive delay line illustrated schematically in Fig. 1. A piezoelectric crystal with a highly polished surface provides the substrate on which a thin metal (usually aluminum) film is deposited. For all but the highest-frequency devices, conventional photolithography is used so that a reusable photomask controls the selective removal of metal resulting in the interdigital transducer patterns. One transducer is connected to an electric source and the other to a detector by means of bonded gold wires; sometimes an electric matching network with inductors and/or resistors is also included between the transducer and source or detector. The driven transducer sets up an electric field in the substrate, which produces a Rayleigh surface wave (Rayleigh, 1885) by the piezoelectric effect; by the converse piezoelectric effect, the other transducer converts the surface wave to an electric signal. At the *synchronous* frequency where the interdigital period L equals the acoustic wavelength, the transduction is most efficient since the waves from all electrodes add in phase. By symmetry, the driven transducer launches both a forward acoustic wave (toward the other transducer) and an unwanted backward wave, which is eliminated by an acoustic absorber so that it will not be reflected from the end of the crystal. There is a direct relationship between the transducer geometry and its impulse response or its transfer function, which is the Fourier transform of the impulse response; this centrally

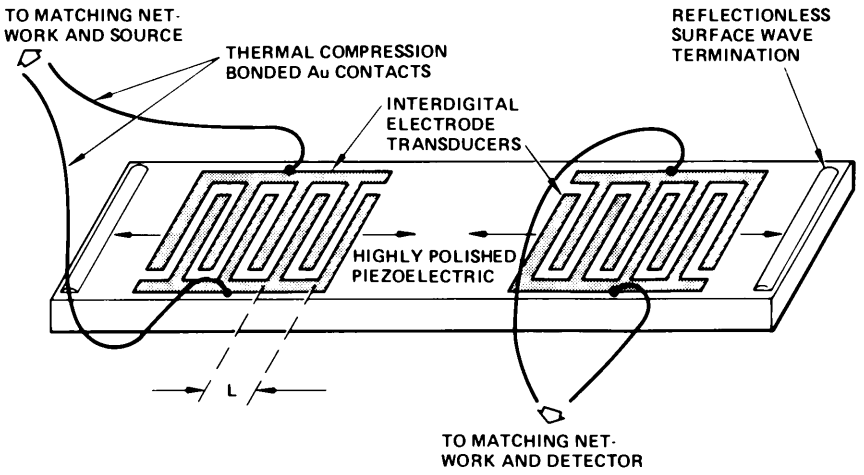


FIG. 1. Essential elements of a SAW delay line with periodic transducers. (After Smith *et al.*, 1969a.)

important feature pervades all the transducer models described in this article.

Because the filter of Fig. 1 has identical periodic transducers, its insertion loss function is restricted to the general shape shown schematically in Fig. 2. The transfer function is approximately

$$T(f) \cong (\sin x/x)^2, \tag{1}$$

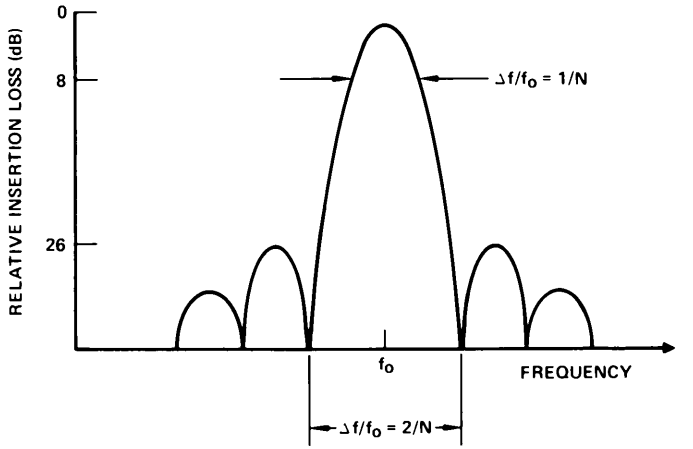


FIG. 2. Bandshape of approximate $(\sin x/x)^2$ form associated with the periodic-transducer delay line of Fig. 1.

where $x = N\pi(f - f_0)/f_0$ and N is the number of interdigital periods. The use of the more general transducer geometries illustrated schematically in Section II, C makes more general filters possible.

B. Transversal Filter Background

The transversal filter concept was introduced by Kallmann (1940) and provides a means of implementing a matched filter for a signal that has finite bandwidth but is otherwise arbitrary in that the amplitude and phase can be independently specified. It provides the required transfer function by repeatedly delaying and sampling a signal.

The filter consists of a delay line with N noninteracting taps of uniform spacing T, connected to a network that multiplies (weights) the nth tap output by a complex number $A_n \exp(j\phi_n)$ and sums the N weighted outputs. This delay line appears in the network of Fig. 3. Its transfer function is

$$H(f) = \sum_{n=1}^{N} A_n \exp(j\phi_n) \exp(-j2\pi fnT), \tag{2}$$

which is periodic with period $\Delta f = 1/T$. We want to avoid aliasing in its output signal, so we must use it to process signals whose power spectral density is nonzero only in the band $|f - f_0| < B/2$, where $T < 1/B$. To ensure this, we add the ideal bandpass filter shown in Fig. 3. The transfer function of the complete network (bandpass filter and tapped delay line

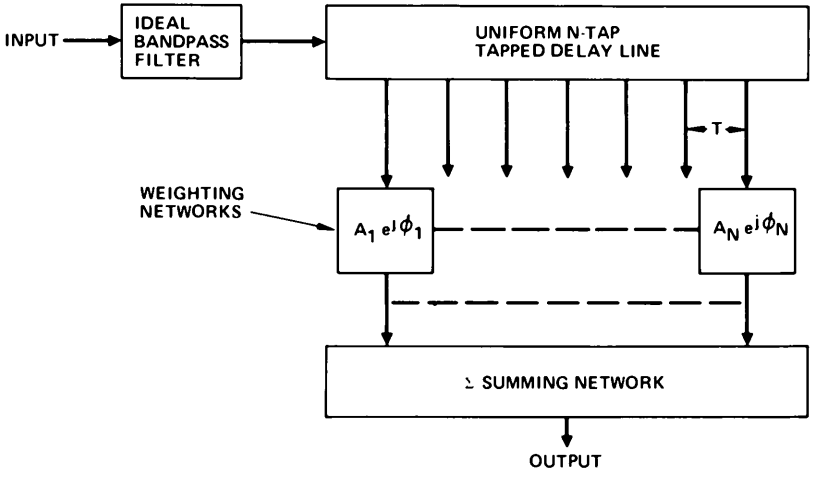


FIG. 3. Schematic diagram of an ideal transversal filter for producing or matching an arbitrary signal of finite bandwidth.

together) is

$$H(f)=\begin{cases}\dfrac{1}{B}\displaystyle\sum_{n=1}^{N} A_n \exp(j\phi_n)\exp(-j2\pi fnT), & |f-f_0|<B/2\\ 0, & |f-f_0|>B/2.\end{cases} \tag{3}$$

Its impulse response is given by the Fourier transform of Eq. (3), viz.,

$$h(t)=\sum_{n=1}^{N} A_n \frac{\sin[\pi B(t-nT)]}{\pi B(t-nT)} \exp[j2\pi f_0(t-nT)+j\phi_n]. \tag{4}$$

The usefulness of the transversal filter for producing and matching signals stems from its close relation to the sampling theorem (Shannon, 1949) applied to signals $s(t)=\tilde{s}(t)\exp(j2\pi f_0 t)$ whose power spectral density is nonzero only inside the band $|f-f_0|<B/2$. Provided that we choose a sampling interval T satisfying $T<1/B$, the signal $s(t)$ may be reconstructed for all times t from samples

$$\tilde{s}(nT)\equiv A_n\exp(j\phi_n) \qquad (-\infty<n<\infty) \tag{5}$$

by means of the formula

$$s(t)=\sum_{n=-\infty}^{\infty} A_n \frac{\sin[\pi B(t-nT)]}{\pi B(t-nT)} \exp[j2\pi f_0(t-nT)+j\phi_n]. \tag{6}$$

The spectrum of $s(t)$ is given by

$$S(f)=\begin{cases}\dfrac{1}{B}\displaystyle\sum_{n=-\infty}^{\infty} A_n \exp(j\phi_n)\exp(-j2\pi fnT), & |f-f_0|<B/2\\ 0, & |f-f_0|>B/2,\end{cases} \tag{7}$$

which has the form of an infinite Fourier-series expansion whose coefficients are just the sample values $\tilde{s}(nT)$.

The transversal filter response [Eqs. (3) and (4)] is an N-term approximation to the signal's Fourier series [Eqs. (6) and (7)], because the tapped delay line takes N samples over a *finite* interval $0<t<NT$, while Eq. (5) specifies *infinite* record length. However, one can reconstruct $s(t)$ accurately in a finite interval $0<t<NT$ from the N complex samples there (see DiFranco and Rubin, 1968). Interdigital transducers take $2N$ (or more) *real* samples there (cf. Section II, B).

Given a required transfer function $S(f)$, the prescription for assigning tap weight coefficients so that $H(f)\cong S(f)$ is simply

$$A_n\exp(j\phi_n)=\tilde{s}(nT), \qquad n=1,\ldots,N \tag{8}$$

where $s(t)$ and $S(f)$ are a Fourier transform pair. In addition we can design the *matched* filter to $S(f)=\text{F.T.}\{s(t)\}$, which is defined as the filter

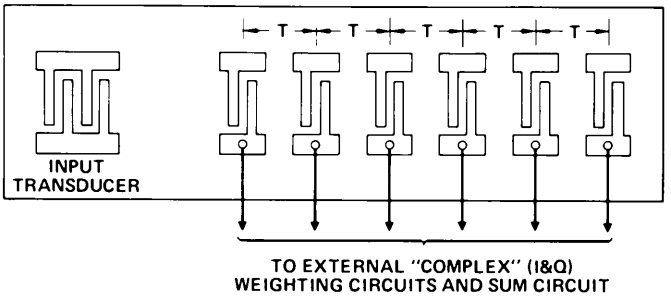


FIG. 4. A SAW tapped delay line used for a good approximate physical implementation of a transversal filter.

whose impulse response is $s(-t)$. For the matched filter we note that F.T.$\{s(-t)\} = S^*(f)$ and choose the transversal filter coefficients so that

$$A_n \exp(-j\phi_n) = \tilde{s}(nT), \qquad n = 1, \ldots, N. \tag{9}$$

Note that the transversal filter is an extremely powerful tool for matched filtering since it can be used to synthesize transfer functions with independently specified amplitude and phase. In order to use the transversal filter concept, it is necessary to have physical components for implementing the structure of Fig. 3. Most critical is the need to be able to construct a tapped delay line whose range of tap amplitudes A_n is wide and whose taps neither appreciably perturb the signal nor interact with (provide feedback to) each other.

SAW interdigital transducers are attractive components for meeting the above requirements. We first consider a rather specialized SAW device —the approximate transversal filter (shown in Fig. 4), which uses a SAW tapped delay line in conjunction with external weighting circuitry. The following nonideal properties of this delay line can be kept within acceptable limits: (1) small reflections of the acoustic wave by each tap; (2) small depletion of energy from the SAW by each tap; and (3) the nonzero spatial extent of each tap, which introduces a large but finite bandwidth into the action of the taps themselves. The SAW tapped delay line has been used in this way to build an adaptive transversal filter for processing coded communication signals with a high degree of interference rejection (Masenten, 1979).

C. Comparison of the Interdigital Transducer to the Versatile Transversal Filter

The foregoing specialized device illustrates the use of SAW technology in a direct application of the transversal filter design prescription [Eq. (8)]. However, the main reason for the emphasis on transversal filter theory is

that *a single interdigital transducer is in many respects similar to the transversal filter structure shown in Fig.* 3. Thus, all SAW filters are devices whose basic building blocks are in fact approximate transversal filters. This observation in fact provides the basis for understanding all the transducer analysis and design to be presented in this article.

Let us examine the specific similarities and differences between the interdigital transducer (IDT) and the ideal transversal filter, with the help of illustrations of typical transducer geometries. Most important, the IDT is a distributed device with N electrodes that take samples of a signal that manifests itself as a propagating SAW, and these samples are weighted in proportion to the apertures (w_n of Fig. 5) and summed as currents flowing into the two parallel bus bars on which the electrodes terminate. However, the electrodes are not necessarily uniformly spaced. Also, there is no provision to allow the tap weight coefficients to be complex: each electrode delivers current to one of the two bus bars so that the associated weight coefficient has $\phi_n = 0$ or $\phi_n = \pi$ according as the electrode is connected to the positive or negative polarity bus bar. In Fig. 5, both transducers are dispersive because they are nonperiodic, i.e., the finger spacing varies through the comb. The positions x_n of the N fingers provide N degrees of freedom for the transducer design. The most widely spaced fingers are most efficient for launching and detecting low-frequency waves with which they are synchronous while the closely spaced fingers are synchronous with, and most efficient for, higher-frequency waves. Hence, waves of different frequency undergo different delays in the filter of Fig. 5, which has a geometry typically used in radar pulse compression systems (Klauder *et al.*, 1960). The "apodized" transducer in Fig. 5 is so named (Tancrell and Holland, 1970) because the different fingers or electrodes have different lengths, so that the active apertures w_n, which are the regions of electroacoustic interaction, are different for every gap between fingers. The effect of the unequal apertures is to assign different amplitude weights to the electroacoustic interaction of the N fingers and thus provide another N degrees of freedom for the transducer design. By using arbitrary electrode positions (x_n in Fig. 5), one effectively varies the temporal position of each sample, and at a single frequency, this is equivalent to

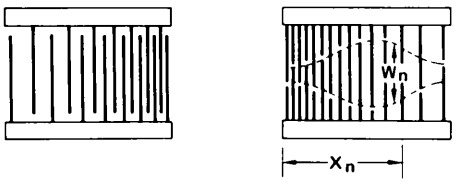


FIG. 5. A pair of nonperiodic, dispersive transducers of the type typically used for radar pulse compression filters. Only one transducer is apodized. (After Smith, 1976a.)

varying its phase. Thus the nonuniformly spaced, real weight coefficients realizable in a transducer give an approximation to the uniformly spaced complex weight coefficients of an ideal transversal filter.

Another property of the IDT that fails to conform to the ideal transversal filter definition is the ability of the electrodes to interact. In Section VI we shall see specifically how this interaction can be made arbitrarily small by invoking appropriate trade-offs. Finally, the IDT, which cannot contain an "ideal bandpass filter," is actually more properly matched to signals that are of finite time extent (corresponding to the transducer length). Such signals cannot be strictly band-limited. Nevertheless, signals of practical interest are negligibly small outside a finite portion of both the time and frequency domains, so the consequences of finite transducer length are tenable.

The geometry of Fig. 6 is typically used for SAW implementation of practical nondispersive bandpass filters, for which some specific design algorithms are listed in Section VII. Again there is one apodized transducer and one unapodized transducer, but in this case the electrodes of both transducers all have the same electrode spacing. Each transducer is nondispersive by virtue of its symmetry with respect to the plane normal to the acoustic propagation direction and running through the center of the transducer. In the unapodized transducer, a certain degree of passband shaping (filtering) is realized by choosing an appropriate electrode polarity sequence—the electrodes need not be connected alternately to the positive and negative polarity summing bars. Most of the filtering is provided by the apodized transducer. Since real-valued amplitude weights of the equally spaced electrodes are appropriate for nondispersive filters, nothing is lost by the inability to implement complex weights as required in the general transversal filter.

The filtering properties of IDT's are then inherently due to the same sampling properties as found in transversal filters. They are basically nonminimal phase networks (Weinberg, 1962) with zeros but no poles in

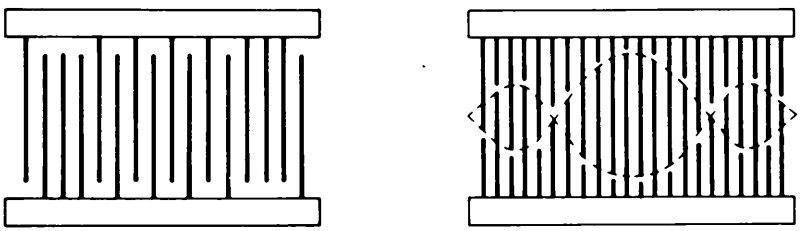


FIG. 6. A pair of nondispersive transducers of the type typically used for bandpass filters. Again only one transducer is apodized. (After Smith, 1976a.)

their transfer functions. This is in contrast to lumped element "LC" filters. The most important reason for the success of SAW–IDT technology may be identified as the ability of the IDT to provide one of the few practical, compact approximations to a transversal filter, i.e., a matched filter to arbitrary signals whose amplitude and phase may be specified independently.

D. Relation of SAW Filter and Transducer Designs

Inasmuch as a SAW filter requires one launching and one receiving transducer, it is important to recognize that the filter transfer function depends on the design of both transducers. In order to make the filter design problem manageable, it is necessary that we be able to express the filter transfer function $H(f)$ as the product of two transducer transfer functions, $H_{in}(f)$ and $H_{out}(f)$ and a factor expressing intertransducer delay T, viz.,

$$H(f) = H_{in}(f)H_{out}(f)\exp(-j2\pi fT). \tag{10}$$

In Sections III and V we deal with the analysis and design of a single transducer. They are applied under the presumption that the separability expressed by Eq. (10) is valid. They also presume that, given a required

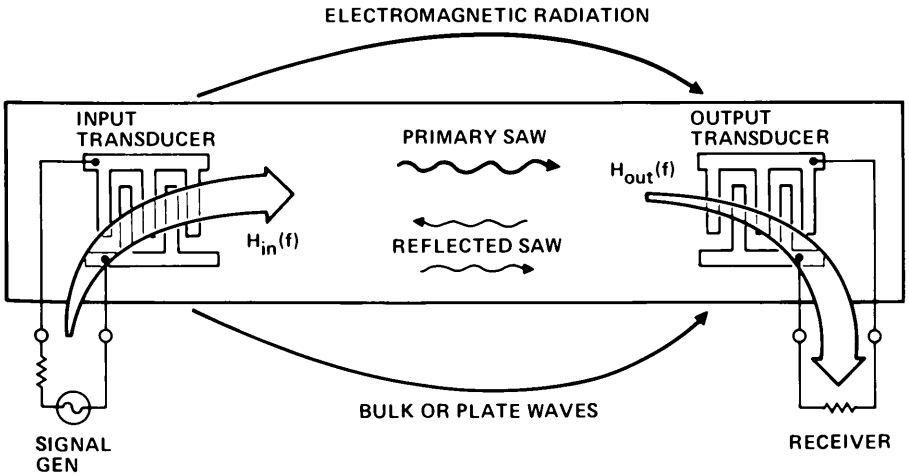


FIG. 7. Schematic diagram of a two-transducer filter, showing the spurious signals that must be minimized so that the filter transfer function is the product of the two transducer transfer functions.

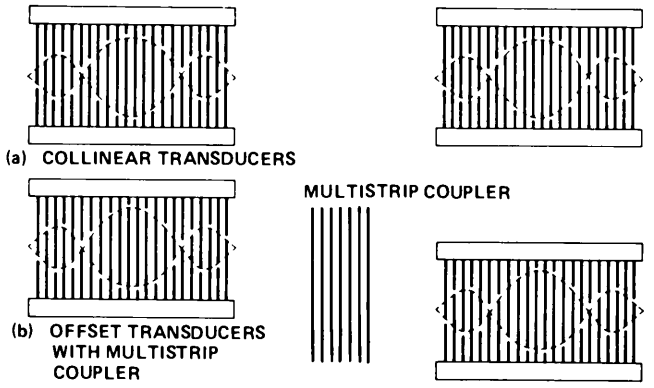


FIG. 8. Two possible configurations for a filter with two apodized transducers. Only with the intervening multistrip coupler can the filter transfer function be expressed as a product of transducer transfer functions. (After Smith, 1976a.)

transfer function $T(f)$ for the SAW filter, two factors $T_1(f)$ and $T_2(f)$ have been chosen such that $T_1(f)T_2(f) = T(f)$, yielding two transducer design problems

$$H_{in}(f) \propto T_1(f) \qquad \text{and} \qquad H_{out}(f) \propto T_2(f). \tag{11}$$

Figure 7 illustrates schematically some of the spurious signals that must be made negligibly small in order for Eq. (10) to be realized. These spurious signals are undelayed feedthrough from electromagnetic radiation and ground-loop currents, piezoelectrically coupled signals propagating as unwanted bulk or plate acoustic modes, and multiple-transit echoes arising from surface waves reflected by the transducers.

Another condition for the validity of Eq. (10) is that at least one of the two transducers be unapodized. It has been shown by Tancrell and Holland (1970) that two apodized transducers aligned collinearly as in Fig. 8a do *not* produce a filter transfer function of the form given by Eq. (10). However, this difficulty can be removed by arranging two apodized transducers with an intervening multistrip coupler as shown in Fig. 8b.

The choice of factors $T_1(f)$ and $T_2(f)$ is of course somewhat arbitrary [we require only that $T_1(f)T_2(f) = T(f)$], but specific examples in the ensuing sections will show that there is often a particularly simple and obvious choice corresponding to the simplest available pair of transducers.

III. Relation of Transducer Response to Geometry

In this section we describe the basic one-to-one relation between transducer geometry and impulse response, invoking at appropriate stages some geometry-based models from the literature. The most important or "first-

order" principles of transducer design follow directly. Circuit models that encompass more detailed aspects of transducer operation are reserved for Section V.

A. Array and Element Factors

Figure 9 is a schematic end-on view of a series of electrodes constituting a portion of an IDT. The transducer is excited by an rf voltage source. The relative polarity of each electrode in Fig. 9 is indicated by the plus and minus signs. We desire to determine the total SAW amplitude emergent from the end of the transducer (at the right-hand side of Fig. 9), which is directed at the other transducer in a SAW filter. For the purpose of this discussion, we consider each electrode to be a source of a Rayleigh wave propagating to the right and need not be concerned with the detailed nature of the electric and elastic fields in this wave. The total wave that emerges at the right-hand end of the transducer is the sum of the waves from all electrodes, each delayed by an amount corresponding to the distance from the electrode of origin (at $x = x_n$) to some reference plane $x = 0$ beyond the right-hand end of the transducer.

Viewed in the frequency domain, this total amplitude takes the form of an *array factor* $A(f)$ given by

$$A(f) = \sum_{n=1}^{N} e_n(f) \exp\left(\frac{-j2\pi f x_n}{v}\right). \tag{12}$$

Here f is the frequency of the rf source and v is the SAW velocity. SAW excitation is distributed in the region just below the crystal surface around the electrodes, and the radiation does not originate solely at the discrete planes $x = x_n$. It is convenient, however, to associate the SAW excitation at any point with that electrode whose center plane $x = x_n$ is nearest, there being a distributed "source field" or "excitation function" in the region

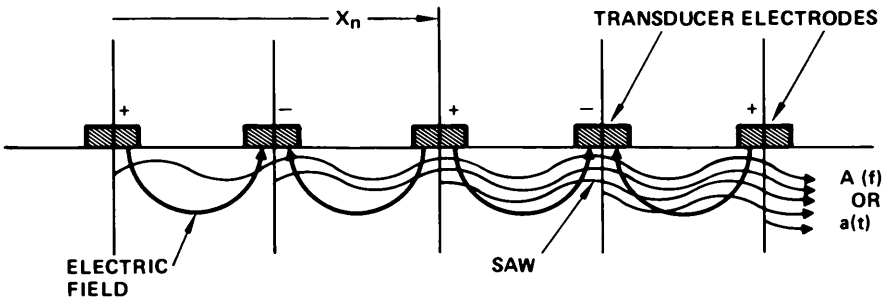


FIG. 9. Schematic end-on view of a sequence of electrodes in an arbitrary transducer, with the emergent SAW represented as a superposition of waves from the various electrodes.

around that electrode. It is because of this distributed excitation that we also write for each electrode an "*element factor*" $e_n(f)$, which is the Fourier transform of the excitation function. Thus

$$e_n(f) \propto a_n \text{F.T.} \left\{ \frac{d}{dx} \left[\mathcal{E}_n\left(\frac{x}{v}\right) \right] \right\}, \tag{13}$$

where a_n is an amplitude coefficient associated with the aperture w_n, F.T. denotes the Fourier transform, and the quantity inside the braces is the excitation function in the region near the nth electrode. It is written as a function of (x/v) to emphasize the Fourier-transform relation between the domains of frequency and electrode position expressed as a time $t = x/v$. This particular dependence of the excitation function on an applied electric field component $\mathcal{E}_n$ is used for the circuit models in Section V and will be discussed further there. For convenience, we let the element factors be normalized such that $a_n = e_n(f_n)$, where f_n is the synchronous frequency at the nth electrode, i.e.,

$$f_n \cong \tfrac{1}{2} v (x_{n+1} - x_n)^{-1}. \tag{14}$$

The coefficient a_n is an "effective amplitude" to be used for designing apertures as described in Section III, D.

The array impulse response $a(t)$ is the Fourier transform of the array factor, viz.,

$$a(t) \propto \sum_{n=1}^{N} a_n \delta\left(t - \frac{x_n}{v}\right) \frac{d}{dx} \left[\mathcal{E}_n\left(\frac{x}{v}\right) \right]. \tag{15}$$

Note that the bandwidth of the array factor is on the order of the reciprocal of the temporal length of the entire transducer, while the element factor bandwidth is approximately the reciprocal of the time delay between two successive electrodes. Thus, the shape of the fundamental

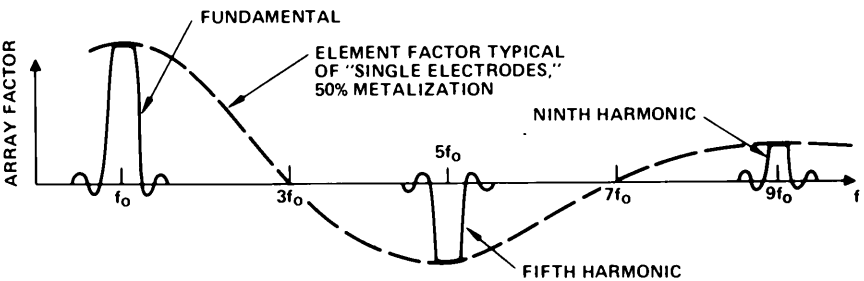


FIG. 10. An illustration of the roles of the array factor (solid curves) and element factor (dashed curve). The curves shown are typical of a bandpass filter transducer with "single electrodes" and 50% metalization.

frequency passband of the transducer is essentially that of the array factor. The effect of the element factor is to introduce a slowly varying envelope that controls the relative amplitudes of the overtone passbands. This is illustrated schematically in Fig. 10, which shows a typical bandpass-filter-transducer array factor with a typical element factor superimposed as a dashed curve.

It is also important to note that the fundamental passband of the array factor is insensitive to the exact distribution of the electric excitation function near one electrode or to the choice of model for that function. It is the relative amplitudes of the various overtone passbands that are controlled by these factors.

B. Delta-Function Model

The delta-function model was introduced by Tancrell and Holland (1970). It is best described by reference to Fig. 11, which shows an end-on view of a sequence of electrodes in a transducer driven by an external ac voltage of the form $e^{j\omega t}$ and radiating surface waves in the $\pm x$ directions. Delta-function sources at the edges of the electrodes are taken as discrete radiators of plane acoustic waves since the edges of the electrodes are the locations where the derivative with respect to x of the applied electric field is largest (Farnell *et al.*, 1970). We note the analogy that the same derivative is the driving term in the equations of motion for longitudinal bulk waves propagating along appropriate crystal axes (Berlincourt *et al.*, 1964, Eq. 42 or Eq. 84).

Each transmitter finger edge at $x_n \pm \frac{1}{2} S_n$ radiates a wave traveling in the positive x direction of the form

$$I_n \exp\left[j(2\pi f/v)\left(x_n \pm \tfrac{1}{2} S_n\right)\right] \exp\left[j2\pi f(t - x/v)\right], \tag{16}$$

where f is the frequency, v is the surface wave velocity, and I_n is an amplitude coefficient for the nth electrode. I_n is taken to have the sign of the associated electrode polarity and magnitude proportional to the aperture w_n. The magnitude is also dependent on the metalization ratio

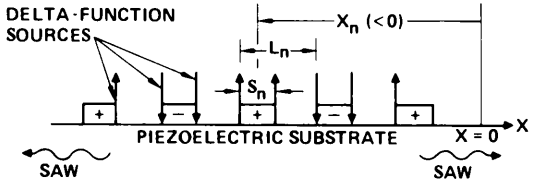


FIG. 11. Schematic end-on view of a sequence of transducer electrodes illustrating the electroacoustic source distribution of the delta-function model. (After Smith, 1976a.)

$S_n/(x_{n+1} - x_n)$. If we sum all of the waves given by Eq. (16) and observe them at the reference plane $x = 0$ (cf. Fig. 11), we find by comparison with Eqs. (12)–(15) that the element factor for the delta-function model is

$$e_n(f) = 2I_n \cos(\pi f S_n/v). \tag{17}$$

While the delta-function model does not specify a physically derived coupling factor between electric and acoustic fields, it gives a complete relationship between the array factor bandshape and the transducer geometry (x_n and w_n). As such, it is useful for transducer design, and the synthesis procedure given for the x_n and w_n in Section III, D may be applied with the delta-function model as the underlying description of transducer operation. Specifically, the delta-function model gives

$$a_n = e_n(f_n) = 2I_n \cos\left(\tfrac{1}{2}\pi S_n/L_n\right). \tag{18}$$

Here I_n is proportional to the aperture w_n and a_n is the effective amplitude coefficient to be used for aperture design in Section III, D.

C. Impulse Model

The impulse model (Hartmann *et al.*, 1973), like the delta-function model, arbitrarily assigns a convenient form for the element factor of each electrode in an IDT. As with the delta-function model, the resultant array factor does not predict the relative amplitudes of overtone passbands but provides accurate analysis and design for realizing a desired array factor at the fundamental (or any one higher-harmonic-frequency) passband. An additional feature of the impulse model is that by appealing to results found from the crossed-field circuit model (cf. Section V, B), a dimensioned scale factor can be assigned to each element factor so that the array factor gives the correct value of the electroacoustic coupling.

The impulse model gives the following prescription for constructing the transducer impulse response:

(1) Construct a waveform having one half-cycle of a sine wave between each two successive electrodes.

(2) Multiply the amplitude of the half-cycle at the nth electrode by $(w_n/w_{\max})f_n^{3/2}$, where w_n is the aperture at the nth electrode and f_n is the instantaneous frequency at the nth electrode given by Eq. (14).

(3) Multiply the entire waveform by $4k\sqrt{2C_s}$, where C_s is the electrostatic capacitance per electrode for an aperture equal to the largest aperture in the transducer and k is the SAW electro-mechanical coupling constant defined in Section V.

The resultant waveform represents the impulse response of the transducer, i.e., the Rayleigh wave signal that emerges from the transducer if a 1-V electrical impulse is applied. The amplitude of this Rayleigh wave is such that its square gives the SAW power. The element factor for one electrode is (as usual) the Fourier transform of the particular half-cycle sinusoid centered on that electrode. Thus,

$$e_n(f) = 4k\sqrt{2C_s}\left(\frac{w_n}{w_{\max}}\right) f_n^{3/2} \int_0^{(2f_n)^{-1}} \sin(2\pi f_n t)\exp(-j2\pi f t)\,dt$$

$$= \frac{2k}{\pi}(2C_s f_n)^{1/2}\left(\frac{w_n}{w_{\max}}\right)\left[\frac{1+\exp(-j\pi f/f_n)}{1-f^2/f_n^2}\right] \tag{19}$$

for the impulse model. Array-factor design (cf. Section III, D) is not concerned with the frequency variation of the element factor but uses the value $e_n(f_n)$ in determining apertures. For the impulse model we have (dropping a constant factor $-j$)

$$a_n = e_n(f_n) = k(2C_s f_n)^{1/2}(w_n/w_{\max}). \tag{20}$$

This is again the effective amplitude coefficient that is used in Section III, D for designing apertures w_n.

D. Design of the Transducer Geometry for a Required Array-Factor Response

In the foregoing sections we have characterized the transducer response as an array factor given by Eq. (12), with the form of the element factor dependent on the choice of model. The array factor will also be applicable to the circuit models given in Section V. Therefore the design problem for a general apodized transducer can be reduced to finding the electrode positions x_n and apertures w_n that cause the array factor to be the best available approximation to some specified transfer function. Equivalently, in the time domain, the transducer impulse response is to be the best available approximation to the required waveform.

Let the required waveform be

$$g(t) = e(t)\exp[\,j\phi(t)\,], \tag{21}$$

where $e(t)$ is a real-valued envelope although not necessarily everywhere positive. Alternatively, the spectrum

$$G(f) = \text{F.T.}\{g(t)\} \tag{22}$$

may be given. In this case the first step is to calculate $g(t)$ by Fourier transform and express it in the form given by Eq. (21). If its time extent is

infinite, it must be truncated at convenient points to give a transducer of finite length.

Next we look at the array-factor expression [Eq. (12)] and ignore the frequency dependence of the element factors, letting $e_n(f) = a_n$, so that the array factor is

$$A(f) \cong \sum_n a_n \exp(-j2\pi f t_n). \tag{23}$$

The corresponding impulse response is

$$a(t) = \sum_n a_n \delta\left(t - \frac{x_n}{v}\right), \tag{24}$$

which is a series of real-valued impulses. Note the similarity between $A(f)$ and the transversal filter transfer function given by Eq. (2). In order to make $A(f)$ the best approximation to $G(f)$, we should like to apply the transversal filter design prescription given by Eq. (8). However, the coefficients a_n are constrained to be real-valued, and the electrode temporal positions t_n can be nonuniformly spaced. These differences are accommodated by choosing the electrode positions x_n such that

$$\phi(x_n/v) = n\pi \tag{25}$$

and choosing the coefficients a_n as

$$a_n = g(t_n) \Big/ \left[\dot{\phi}(t_n)\right]. \tag{26}$$

In this way, the impulse response is the best available sampled version of $g(t)$; since the transducer samples the SAW at times t_n where $g(t)$ is real valued and the amplitudes of the samples are proportional to the local value of $g(t)$. Note the extra factor $[\dot{\phi}(t_n)]^{-1}$ in Eq. (26), which means that the array amplitude coefficient a_n should also be inversely proportional to the instantaneous frequency

$$f_n \equiv (1/2\pi)\left[\dot{\phi}(t)\right]_{t=t_n}. \tag{27}$$

Physically, this arises because the density of samples (per unit time) is proportional to the instantaneous frequency, and the sample amplitudes must be taken in inverse proportion to the instantaneous frequency to compensate for the varying sample density.

In the transducer design, we are interested in apertures w_n rather than amplitude coefficients a_n. It has already been found from the impulse-model expression for $a_n = e_n(f_n)$ [Eq. (20)] that

$$w_n \propto a_n f_n^{-1/2}. \tag{28}$$

The presence of the factor $f^{-1/2}$ has been derived from energy considera-

tions (Hartmann *et al.*, 1973) and is also found for the crossed-field model in Section V, B. Thus, combining Eqs. (26) and (28), the aperture design formula is

$$w_n \propto f_n^{-1/2} a_n \propto f_n^{-3/2} g(t_n). \tag{29}$$

It should be noted that the delta-function model ignores the factor $f^{-1/2}$ that appears in Eq. (28). However, the whole issue of the dependence of w_n and a_n on instantaneous frequency f_n is relevant only for dispersive transducers whose electrodes may have different synchronous frequencies f_n.

1. *Array Factor Design Examples*

a. *Nondispersive filter transducer.* Here we illustrate the application of Eqs. (25)–(29) to the design of a nondispersive filter transducer with general amplitude envelope. For this case

$$g(t) = e(t) \exp[\, j\phi(t)\,] \tag{30}$$

with

$$\phi(t) = 2\pi f_0 t. \tag{31}$$

From Eq. (25) the electrode positions are

$$t_n = n/2f_0, \tag{32}$$

while the apertures w_n are proportional to $e(t_n)$. The resultant transducer with uniformly spaced electrodes is illustrated schematically in Fig. 12. Specific examples of commonly used envelopes $e(t)$ will be discussed in Section VII, A.

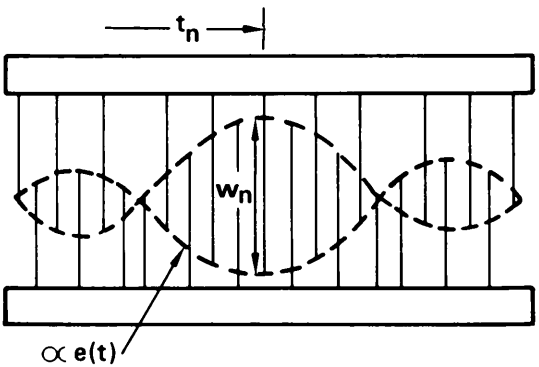


FIG. 12. A typical electrode geometry for a bandpass filter transducer, showing the relation between the apertures w_n and the required impulse response envelope $e(t)$.

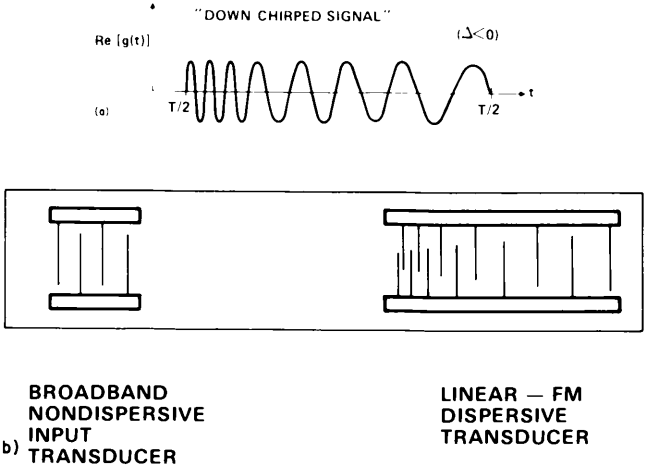


FIG. 13. A down-chirped linear-FM waveform and the electrode geometry of a SAW filter for producing it.

b. *Linear-FM dispersive transducer.* The envelope and phase of the linear FM waveform are

$$e(t) = \begin{cases} 1, & |t| \leqslant T/2 \\ 0, & |t| > T/2 \end{cases}, \tag{33}$$

$$\phi(t) = 2\pi\left(f_0 t + \frac{\Delta}{T}\frac{t^2}{2}\right). \tag{34}$$

This waveform is illustrated schematically for the "down-chirped" case ($\Delta < 0$) in Fig. 13a. The electrode positions are

$$t_n = -\frac{f_0 T}{\Delta}\left\{1 - \left[1 + \left(\frac{n}{f_0 T}\right)\left(\frac{\Delta}{f_0}\right)\right]^{1/2}\right\}, \tag{35}$$

where

$$T\left(\frac{\Delta}{4} - f_0\right) \leqslant n \leqslant T\left(\frac{\Delta}{4} + f_0\right). \tag{36}$$

The apertures are

$$w_n = w_0(f_n/f_0)^{-3/2}, \tag{37}$$

where w_0 is the aperture located where

$$n = 0, \qquad t = 0, \qquad f = f_0. \tag{38}$$

This transducer is illustrated in Fig. 13b, along with a broadband, nondis-

persive input transducer such as might be used to complete a linear FM filter.

c. *Biphase coded tapped delay line.* A biphase coded waveform takes the form

$$g(t) = \left[U\left(t + \frac{\tau}{2}\right) - U\left(t - \frac{\tau}{2}\right) \right] \exp(j2\pi f_0 t) * \sum_n \exp(j\phi_n)\, \delta[t - (n-1)\tau], \tag{39}$$

where U is the unit step function, the asterisk denotes convolution, and the code phases ϕ_n take the value 0 or π. Each of the two transducers in this delay line is matched to one of the two portions of the waveform, which are separated by the convolution sign. The portion

$$\left[U\left(t + \frac{\tau}{2}\right) - U\left(t - \frac{\tau}{2}\right) \right] \exp(j2\pi f_0 t) \tag{40}$$

is commonly called one "chip" of carrier and is shown in Fig. 14a. The portion

$$\sum_n \exp(j\phi_n)\, \delta[t - (n-1)\tau]$$

is the biphase code and is shown in Fig. 14b. The total waveform $g(t)$ is shown in Fig. 14c. Applying Eqs. (25)–(29), we find that the transducer for the carrier chip is characterized by

$$t_m = m/2f_0, \qquad -f_0\tau < m < f_0\tau \tag{41}$$

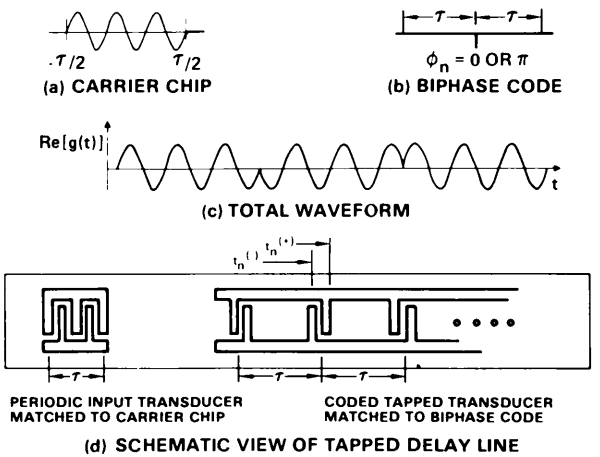


FIG. 14. A biphase-coded waveform and a SAW tapped delay line for producing it.

and

$$w_m = w_0 = \text{const}, \tag{42}$$

i.e., it is a periodic transducer. For the biphase code, Eqs. (25)–(29) would strictly call for taps of one sampling electrode each, but in order to obtain piezoelectric coupling with the polarities suitably defined we require taps with at least two electrodes each. Hence we take two-electrode taps with

$$t_n^{(\pm)} = n\tau \pm \exp(j\phi_n)/4f_0 \tag{43}$$

and

$$w_n = w_0 = \text{const}. \tag{44}$$

The tapped delay line composed of these two transducers is shown in Fig. 14d. In practice, the tapped transducer may have apertures w_n that gradually increase with t to compensate for propagation loss.

E. Source-Withdrawal Weighting

The design prescription given by Eqs. (25)–(29) calls for apodized transducers for matching general waveform envelopes. However, the use of two apodized transducers yields to tractable design procedures only in broadband filters on lithium niobate substrates when an intervening multistrip coupler is used. In filters on weakly piezoelectric substrates where multistrip couplers are not useful, it is desirable to be able to implement unapodized transducers which have more band-shaping flexibility than simple periodic transducers. Source-withdrawal weighting is a technique that is especially useful for improving the selectivity of narrowband, nondispersive unapodized transducers. Conceptually, it is analogous to the design of thinned, end-fire antenna arrays (Collin and Zucker, 1969). It uses the polarities (choice of positive or negative polarity bus bar connection) of the electrodes as degrees of freedom and recognizes that the polarity sequence is not constrained to alternate as in periodic transducers.

The first withdrawal weighting technique (Hartmann, 1973) consisted of selective omission of electrodes from a transducer whose electrode spacing was otherwise uniform. In the source-withdrawal weighting technique, we retain a continuous grating of electrodes that are identical except for their polarities. Figure 15 is a schematic end-view of a sequence of "double electrodes" (see Section IV, C) in an unapodized transducer. The electric fields are strong in the regions labeled "ordinary sources" because adjacent electrodes have opposite polarity. In the regions labeled "missing or 'withdrawn' sources," the electric fields are weaker because adjacent electrodes have like polarity and are shielded from oppositely polarized electrodes that are at least two electrode positions removed.

Note that an odd number of consecutive missing sources corresponds to an electrode phase reversal with respect to the alternating polarity

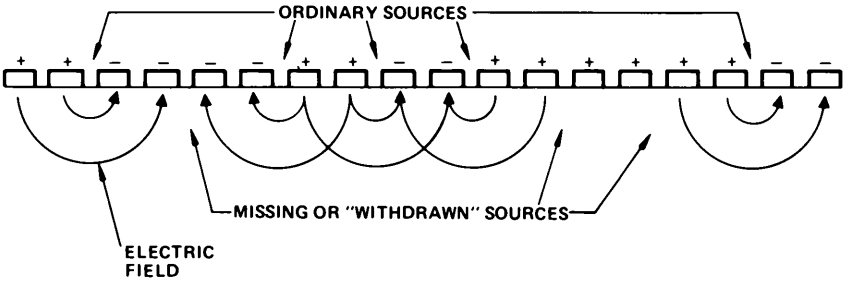


FIG. 15. Schematic end-on view of a sequence of "double electrodes" illustrating the concept of "withdrawn sources" used for amplitude weighting in unapodized transducers.

sequence of a periodic transducer. Such a sequence is analogous to the two adjacent electrodes of like polarity that occur in apodized transducers where $e(t)$ changes sign. By contrast, an even number of consecutive missing sources corresponds to a region in an unapodized transducer where the apertures are smaller than the maximum but where $e(t)$ does not change sign. Thus, in the following withdrawal weighting procedure, consecutive sources must be withdrawn in even numbers except at times t where $e(t)$ changes sign, in which case an odd number of sources must be withdrawn.

In its simplest form, the withdrawal weighting procedure assumes that "ordinary" sources have amplitude ± 1, while "missing" or "withdrawn" sources have amplitude zero. Assuming that the required waveform is nondispersive [i.e., $\phi(t) = 2\pi f_0 t$] and that $e(t)$ is normalized such that $\max\{|e(t)|\} = 1$, we calculate the envelope integral

$$I(t) = \frac{1}{T}\int_0^t e(\xi)\, d\xi \qquad (0 < t < T), \tag{45}$$

where the ends of the transducer are at $t = 0$ and $t = T$. We arbitrarily assign positive polarity to electrode number "zero," which is the first electrode at $t = 0$.

Define

$$\mathcal{N}(n) = \sum_{i=1}^{n} p_i\left(1 - \delta_{p_i, p_{i-1}}\right), \tag{46}$$

where $p_i = \pm 1$ is the polarity of the ith electrode and δ_{kl} is the Kronecker delta. The quantity $\mathcal{N}(n)$ is thus the (signed) cumulative number of sources with due regard that sources can have negative sign if an odd number of sources has been "withdrawn" in the interval $(0, t)$.

The source-withdrawl prescription is to proceed sequentially to electrode positions $n = 1, \ldots, N = (2f_0 T + 1)$ and choose the polarity $p_n = \pm 1$

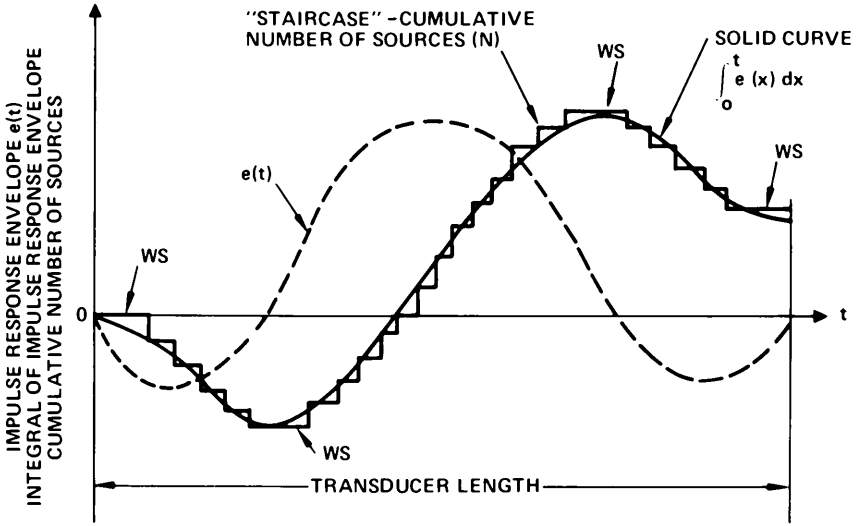


FIG. 16. Illustration of the procedure for designing source-withdrawal weighted transducers. The cumulative number of sources between the end of the transducer and any position within it is proportional to the corresponding area under the required impulse response envelope. WS = withdrawn source.

such that

$$\mathcal{N}(n) \cong (2f_0T + 1)\, I(n/2f_0). \tag{47}$$

Here the approximate equality sign means that $\mathcal{N}(n)$ is the integer nearest to the quantity on the right-hand side, subject to the constraint that an odd number of consecutive sources can be omitted only where $e(t)$ changes sign.

The prescription given by Eq. (47) is illustrated schematically in Fig. 16, where the dashed curve represents a typical envelope $e(t)$, the solid curve represents $I(t)$, and the "staircase" represents $\mathcal{N}(n)$. The cumulative number of sources is proportional to the indefinite integral of the impulse response envelope. The local density of sources is proportional to the magnitude of the envelope. Source-withdrawal weighting is most useful and accurate in narrow-band filters because the total number of electrodes is large and the staircase-shaped function $\mathcal{N}(n)$ becomes a better approximation to $I(t)$.

IV. "Second-Order" Effects and Their Compensation

Generally, by "second-order" effects in interdigital transducers we mean any phenomena that cause the transducer transfer function to deviate from the form given by an array factor in Section III. In this section we shall

describe several of the more important second-order effects and the appropriate measures for minimizing or compensating for them. Some of them require the circuit model concepts of Section V for a detailed understanding. However, that machinery is not needed for a conceptual introduction to the important effects and it is particularly desirable to introduce "double electrodes," the remedy for electrode surface loading distortion (cf. Section IV, C), before considering the circuit models. Later, using the circuit models, we shall see how to minimize or compensate "regeneration distortion" (cf. Section IV, D). Except in connection with these two particular "second-order" effects, whose remedies are discussed in some detail, our intent in Section IV is merely to make the reader aware of some potential problems and to direct him to the appropriate literature for more detailed discussion.

A. Diffraction and "Beam Steering"

The array-factor characterization of interdigital transducers given in Section III assumes that each electrode radiates a plane acoustic wave that propagates without diffraction in a direction perpendicular to the electrode. In practice, the waves diffract in a manner that is in general complicated by the elastic, dielectric, and piezoelectric anisotropy inherent in all piezoelectric substrates. The shape of a diffracted wave front can be well characterized (Szabo and Slobodnik, 1973, 1974) as a function of propagation distance from knowledge of the size of the launching aperture and of the SAW velocity as a function of propagation direction. Typically the latter can be expressed by a parabolic velocity-versus-angle approximation for small to moderate angles near the nominal direction of launch. Constants appropriate to this approximation are tabulated by Szabo and Slobodnik (1973) for the more important SAW materials in their common orientations.

In compensating for diffraction in the transducers of a SAW filter, it is necessary to treat the interaction between a finite-aperture electrode in the receiving transducer and the diffracted field from the launch transducer. The simple transducer design rules of Section III, which ignore diffraction, can be modified (Szabo and Slobodnik, 1974) to correct for both the amplitude and phase distortion caused by diffraction. The details of the correction will not be discussed in this article.

Another consequence of the anisotropy of piezoelectric substrates is that the phase and group velocities of the SAW are in general not collinear. Thus the SAW power may have a propagation direction different from that of the phase fronts. Most of the commonly used crystal orientations have both the surface normal and the SAW propagation along crystal

symmetry axes so that the phase and group velocities are collinear. For devices using these materials, then, the problem becomes one of imperfect alignment of the transducers or the polished crystal surface with respect to the desired crystal axes. When this occurs, the propagation of SAW power will not be properly directed at the receiving transducer and some of the available power will be lost. Coefficients that in effect give the size of the power "walkoff" angle as a function of transducer misalignment are also given by Szabo and Slobodnik (1973). This information is useful for establishing tolerances on the crystal and transducer alignment errors.

B. Wave-Front Distortion from Apodization; "Dummy" Electrodes

Nonplane acoustic wave fronts can occur in apodized transducers constructed as shown in Fig. 17a. They result from the slowing of surface waves in metalized regions, both because of the shorting of the tangential electric field at the surface and by mass loading. The former effect increases with the substrate k^2, while the latter increases with the electrode thickness. As described by Gerard *et al.* (1972), the nonplane wavefronts can be eliminated by constructing apodized transducers as shown in Fig. 17b. It is assumed that all apodized transducers are constructed in this manner so that distortions caused by nonplane wave fronts are eliminated, and it is not necessary to account for them in the analysis.

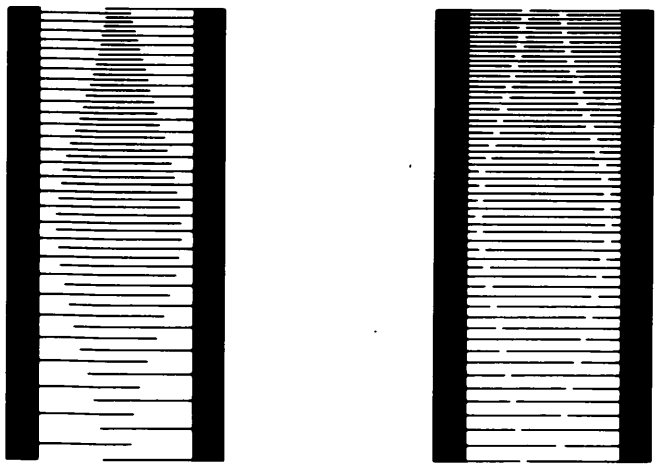


Fig. 17. Illustration of the use of "dummy electrodes" (b) to prevent the nonplane wave fronts associated with the geometry shown in (a). (After Smith *et al.*, 1972.)

C. Electrode "Mass/Electrical Loading"; Double Electrodes

Metal electrodes influence the SAW propagation in two ways. First, their edges introduce a discontinuity that is not only topographic but also a transition to a region where the overlay material (usually aluminum) has elastic properties different from those of the substrate. Second, the metal electrodes, being nearly perfect conductors, introduce a boundary condition that the tangential electric field is zero at the substrate–electrode interface—a condition that is not present at the free crystal surface in the region between electrodes. The first effect is commonly called "mass loading" and the second, "electrical loading." The most important consequence of these two types of discontinuity is that there is a small reflection of the SAW at each edge of each electrode. For mass loading it has been derived heuristically (Li and Melngailis, 1975; R. C. M. Li and R. C. Williamson, personal communication) and verified experimentally that the magnitude of the reflection coefficient of one edge is

$$r_m = \pm \tfrac{1}{3}(\rho_S/\rho_G)h/\lambda, \tag{48}$$

where ρ_S and ρ_G are the mass densities of the electrode metal and substrate, h is the electrode thickness, and λ is the acoustic wavelength. It has been similarly found for electrical loading (Smith *et al.*, 1972; Jones *et al.*, 1972) that the reflection coefficient of one edge is

$$r_e \simeq \pm \tfrac{1}{4}k^2, \tag{49}$$

where k^2 is the electromechanical coupling constant of the substrate. In Eqs. (48) and (49), the $\pm$ signs indicate that the signs of the reflections on entry and exit from the electrode region are opposite.

The reflections caused by electrode mass/electrical loading can be particularly troublesome in devices with large numbers of conventional quarter-wave electrodes spaced on half-wave centers ("single electrodes"). This is because the reflections from all electrode edges add coherently (see Fig. 18a). If the width of each electrode is exactly one-quarter wavelength, the problem is compounded by exact coherent addition of the reflections from the leading and trailing edges of one electrode as well as by coherent addition of reflections from corresponding edges of successive electrodes. In effect, the transducer with single electrodes has an acoustic reflective grating stopband right at the synchronous frequency where the electroacoustic excitation is most efficient.

The technique for alleviating—in fact, almost completely eliminating—the stopband problem is the use of "double electrodes" (Bristol *et al.*, 1972). A "double electrode" consists of a pair of electrodes of the same electric polarity, spaced by one-quarter wavelength and each approximately one-eighth wavelength wide. Figure 18b illustrates schematically

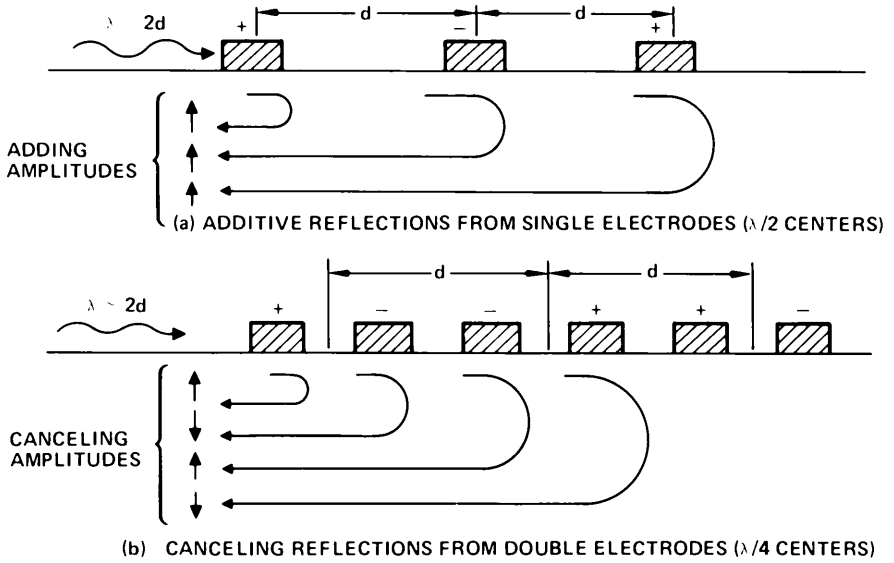


FIG. 18. Schematic end-on view of "single" and "double" electrodes, illustrating the addition and cancellation of "mass/electrical loading" induced reflections near acoustic synchronism. (After Smith, 1974.)

the manner in which the mass/electrical loading reflections from double electrodes largely cancel near the synchronous frequency where $\lambda = 2d$, the residual reflection from the transducer being approximately the size of the reflection from one electrode edge.

Note that the use of double electrodes does not make a dramatic change in the electroacoustic excitation—the electrode polarities shown in Fig. 18 indicate that the electrical period of the structure is the same for single and double electrodes. Since the exact distribution of the exciting electric fields is somewhat different, the *element factor* (cf. Section III) is different for double electrodes, which therefore exhibit a completely different pattern of overtone excitation strengths. Near the fundamental synchronous frequency, however, the array-factor responses of corresponding arrays of single and double electrodes are virtually identical.

The price of obtaining the superior reflection suppression of double electrodes is the necessity of fabricating lines with twice the resolution requirement of single electrodes operating at the same frequency. This can often be overcome by operating double-electrode transducers at their third-harmonic passband, which we shall find to be efficient in Section V, D. Double electrodes should generally be used unless prohibited by the resolution capability of the fabrication process.

D. Regeneration Distortion

The array-factor view of the interdigital transducer does not account for the fact that the driving source or receiver load has finite impedance, giving rise to the generation of secondary surface waves that are a form of distortion. Although this effect will be analyzed in some detail in Section V, E, its general nature as a "second-order" effect is easily described here.

Figure 19 is a schematic diagram of an interdigital transducer receiving an incoming SAW and connected to a receiver load of finite impedance. The direct effect of the incoming SAW, as a consequence of the piezoelectric effect, is to set up a voltage V_0 across the receiver load. This voltage in turn acts as a source driving the transducer, so that each electrode reradiates secondary surface waves in both directions. These waves are in turn detected by all electrodes in the same way that the incoming SAW was detected, producing a secondary voltage V_1 in the load. Of course there are "higher-order" regenerated waves as the voltage V_1 excites the transducer and so forth, in an infinite series. Since these secondary voltages represent a distortion superimposed on the desired signal that has the form of the array-factor response, they can be called "regeneration distortion" and categorized as a second-order effect.

In Sections V and VI it will be shown how the regeneration distortion can be characterized in closed form, its severity traded off against filter insertion loss, and (especially in the case of dispersive transducers) how the transducer design can actually be modified to compensate for the distortion.

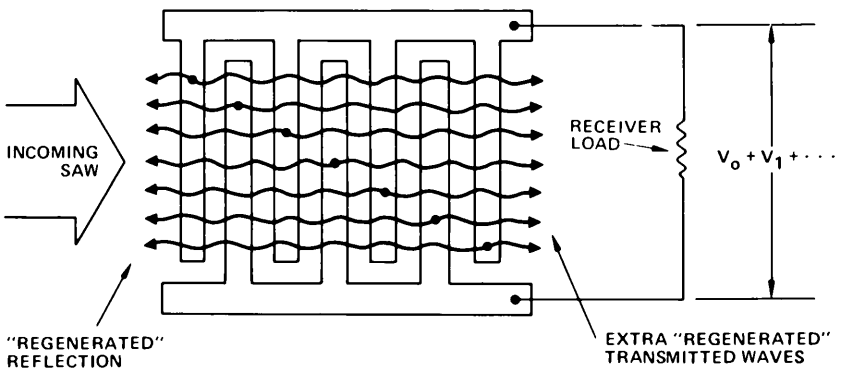


FIG. 19. A schematic drawing of the secondary "regenerated" waves caused by the voltage induced in a receiver load impedance by an incoming SAW. V_0 = "main signal," V_1 = "first-order regeneration," . . . .

E. Triple-Transit Echoes

When a surface wave is incident on an interdigital transducer connected to a finite-impedance source or load, a portion of the wave is reflected. This occurs partly because of electrode "mass/electrical loading" (cf. Section IV,C—even with double electrodes there is a small residual reflection). The main source of reflected waves, however, is usually the backward regenerated wave described in Section IV,D. In a SAW filter this "reflection" gives rise to multiple-transit echoes of which the most important is usually the triple-transit echo diagrammed in Fig. 20. The electroacoustic power conversion efficiency of each transducer is p_{13}. Thus the filter insertion loss for the desired (single-transit) output signal is given in decibels by $20 \log_{10} p_{13}$.

Also, each transducer has a power reflection coefficient p_{11}, where $0 < p_{11} < 1$. The triple-transit echo is that signal which is launched as a SAW by the input transducer, is reflected by the output transducer and again by the input transducer, and finally is converted back to an electric signal by the output transducer. The overall insertion loss for this signal is $20 \log_{10}(p_{11} p_{13})$, and the *triple-transit suppression* is defined as the difference in the insertion losses of the triple-transit echo and the desired signal. Thus the triple-transit suppression is given in decibels by

$$\text{TTS} = 20 \log_{10} p_{11}. \tag{50}$$

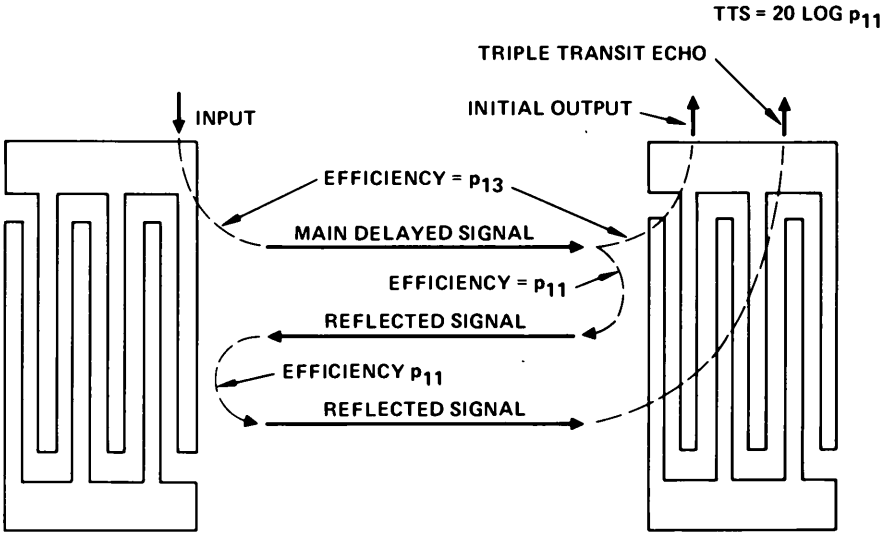


FIG. 20. An illustration of the paths taken by the main delayed signal and the triple-transit echo in a SAW filter.

If a pulsed signal is applied to the filter, the triple-transit output follows the main output by twice the filter delay. If a swept-cw insertion loss measurement is made, the triple-transit signal interferes with the main signal and causes an amplitude and phase ripple superimposed on the desired filter transfer function.

The circuit models of Section V enable calculation of the triple-transit suppression and lead to the trade-offs involving the TTS that are given in Section VI.

F. Bulk or Plate Waves

In addition to the Rayleigh wave, which is usually the desired surface wave mode, there are a number of other piezoelectric waves that may be excited by an interdigital transducer. It is not our purpose here to present a theoretical description of bulk wave excitation, but only to list the commonly observed types of spurious signals from undesired acoustic modes and to outline measures for suppressing them.

Using available suppression techniques as a criterion for classifying "bulk" or plate waves, we find that there are two general types of unwanted modes. The first are sometimes referred to as "shallow propagating bulk waves" and are really modes that propagate parallel to the crystal surface with velocities higher than that of the Rayleigh wave. They are shown schematically as "shallow bulk waves" in Fig. 21. They might be considered "degenerate" solutions of the wave equation for the given crystal medium in that they satisfy the traction-free surface boundary conditions without being composed of four surface-confined partial waves like the general Rayleigh wave. They are able to satisfy this boundary condition only as a consequence of crystal symmetry, which automatically renders the required stress components identically equal to zero. Modes of this type are usually classified as longitudinal, or vertically or horizontally

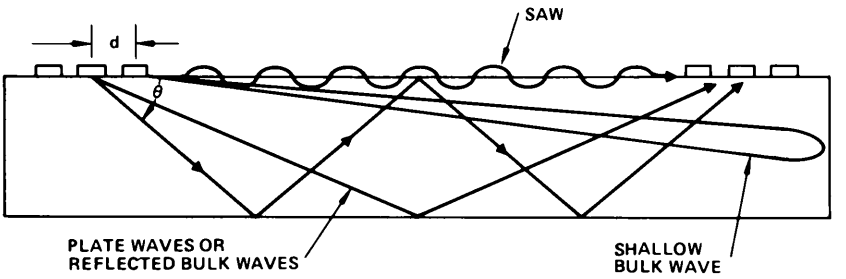


Fig. 21. The paths taken by various bulk and plate waves that can produce spurious output signals in a SAW filter.

polarized shear waves (Auld, 1973, Chapter 10). Since they travel at fixed velocities characteristic of the substrate, they give rise to spurious filter passbands at various frequencies, which are higher than the fundamental SAW passband. The only means of suppressing them is the use of electrical matching circuits that augment the SAW filter rejection at these frequencies, or choice of substrates that do not support the modes at all.

The second type of unwanted "bulk" waves is really the set of "plate modes," which are in fact normal modes of the substrate whose thickness is finite although large compared to the SAW wavelength. A theory that recognizes that these are in fact normal modes of the plate (Wagers, 1976, 1977) has produced a highly accurate prediction of the associated spectrum of unwanted signals in a SAW filter. For the purpose of identifying means of suppressing these unwanted signals, we may consider them to be bulk waves (as shown schematically in Fig. 21) that propagate down into the substrate at angle θ and reach the output transducer after one or more reflections from the back crystal surface. For each angle of propagation θ, there is a bulk wave velocity $v(\theta)$ and a corresponding frequency of synchronous excitation given by

$$f(\theta) = v(\theta)\cos\theta/2d, \tag{51}$$

where d is the electrode spacing. For transducers of finite length, there is a broad spectrum of waves that can be transmitted between the input and output transducers by means of one or more reflections from the back crystal surface.

The obvious means for suppressing spurious output signals of this type consists of eliminating the specular reflection at the back crystal surface. The available degree of suppression varies with the complexity of the techniques used to treat this surface. The simplest technique is roughening with coarse grit. More effective but expensive means include making the two crystal surfaces nonparallel, cutting oblique grooves in the back surface, and applying absorbing materials to the cut or roughened back surface.

G. Electromagnetic Feedthrough and Ground Loops

There are two mechanisms by which an undelayed, unfiltered signal can be transmitted directly from the input to output transducer circuits. The first of these is electromagnetic radiation between the input–output transducers and their associated matching circuits, which may contain tuning coils. The most reliable way to prevent crosstalk between coils and other matching circuit components is to isolate them in separate cavities in a metal package and connect them to the transducers via insulated feedthrough pins to a third cavity, which contains the SAW substrate. The

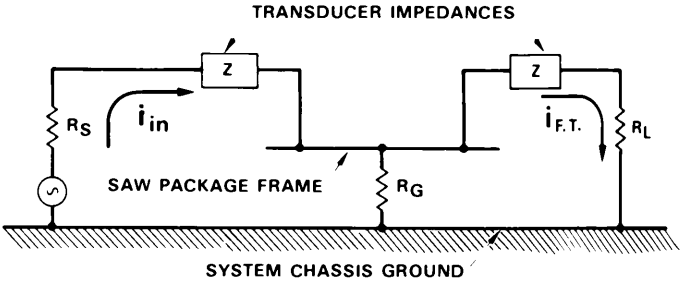


FIG. 22. Equivalent circuit showing the cause of spurious undelayed output signals in a SAW filter with inadequately grounded transducers.

desirability of doing this must be weighed against the possibility of encapsulating the entire filter in an inexpensive, commercially available microcircuit package. The radiative coupling between the transducers themselves cannot be completely blocked with isolation barriers since an opening at least as large as the substrate cross section is required. The problem of minimizing the intertransducer radiation is equivalent to avoiding excitation of the electromagnetic guided modes of the encapsulating package. A balanced-transformer feed is often helpful.

The other source of undelayed, unfiltered transmission might be called ground loops. Usually the rf source and the output load on a SAW filter share a common ground on the chassis of the system in which the filter is a component. It is essential that a good, low-resistance connection be established between the transducer ground terminals and the "system" ground. Thus, the transducers must be well grounded to the package that encapsulates the SAW filter, which must in turn be in good contact with the system chassis. The consequence of significant resistance in these ground connections can be seen by referring to Fig. 22. The transducers each have impedance Z and the detector resistance is R_L. If the ground connection resistance R_G is not zero, a portion of the input current given by

$$i_{F.T.} = \frac{R_G}{R_G + Z + R_L} i_{in} \tag{52}$$

flows directly into the output circuiting, given an undelayed, unfiltered output signal.

H. Electrode Resistivity and Acoustic Propagation Loss

The relatively thin films used for transducer electrodes give rise to sheet resistivities on the order of 1 Ω per square. The composite ohmic loss resistance of the entire transducer is proportional to the aperture and

inversely proportional to the number of electrodes. In a typical transducer the input resistance component from electrode ohmic losses is a few ohms. Since the ohmic loss in each electrode is distributed along its length, the acoustic excitation can be nonuniformly distributed across the aperture in electrodes with very wide apertures. If an apodized transducer has a large maximum aperture (say more than 100 wavelengths), the ohmic electrode resistance can cause the relative excitation strengths of the electrodes to deviate from exact proportionality to the apertures (see Lakin, 1974). In this event the apertures can be adjusted to compensate (see Wagers, 1979). Usually maximum apertures are kept less than 100 wavelengths to minimize ohmic losses.

Aside from an overall increase in filter loss, acoustic propagation attenuation can introduce distortion in two ways. First, the electrodes at the facing "inside" ends of the transducers interact via a SAW which propagates over the shortest possible distance. Thus, in long transducers, these electrodes have larger effective interaction amplitudes than electrodes at the more distant "outside" ends. Second, the propagation loss usually varies as the second or higher power of the frequency, which in dispersive transducers effectively reduces the amplitudes of closely spaced electrodes that are synchronous at high frequencies. In either case the distortion can be compensated by modifying the apertures in the apodized transducer.

V. Three-Port Circuit Model Description of Interdigital Transducers

A. General Capabilities of Three-Port Circuit Models

Because the interdigital transducer launches and detects surface waves from two opposed directions and is in addition connected to an external electric network, a three-port network is appropriate for describing all the transfer functions of a transducer. In addition to allowing a description of the electroacoustic interactions even in the presence of strong coupling, the three-port transducer network can also be combined with other linear networks for electric matching circuits and acoustic delay between transducers so that a two-port network can be formed for representing a filter or delay line. Details of the latter network constructions are given in Section V, D.

The basic three-port network for an interdigital transducer (Smith *et al.*, 1969a, 1972) is shown schematically in Fig. 23. At the electric port 3, the voltage E_3 and current I_3 represent the actual voltage and current

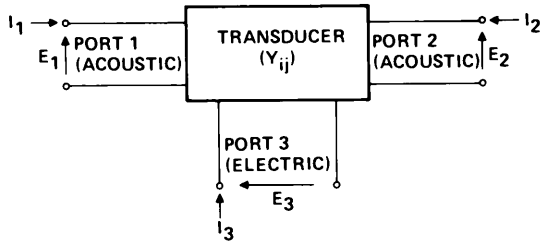


FIG. 23. Basic three-port block diagram showing the terminal variables that can be related by a 3×3 matrix such as the admittance matrix Y. At acoustic ports, voltages represent force and currents represent particle velocity. (After Smith *et al.*, 1969.)

applied to the bus bars of the transducer. At the acoustic ports 1 and 2, "voltage" E_1 and E_2 may be considered to be acoustic forces (sum of the incoming and outgoing waves) while "currents" I_1 and I_2 are particle velocities for the same waves. As such, the "voltage" and "current" forming either the incoming or outgoing wave at an acoustic port constitute a pair of dynamical variables whose product gives time average acoustic power flow and whose ratio is a characteristic wave impedance.

A rigorous characterization of acoustic forces and particle velocities of a Rayleigh wave in an anisotropic substrate requires use of the crystal tensors (Auld, 1973). However, a fundamental assumption of the three-port circuit approach is that for a single surface wave mode, in this case the fundamental Rayleigh wave mode, the "voltage," "current," and wave impedance of an acoustic wave are scalar quantities as in conventional transmission line theory. Furthermore, it turns out that the acoustic wave impedance may be taken as dimensionless without loss of generality; we choose it as unity except in modeling the "mass/electrical loading" described in Section IV,C, where it is unity in unmetalized regions and slightly different from unity in electroded regions.

The three-port characterization of a transducer is quite useful because appropriate ratios among the terminal variables give all the properties of interest for the transducer. By using networks to find the transfer functions T_{ij} defined below, we obtain a description of filter insertion loss, bandshape, triple-transit echoes, regeneration distortion, and extract many key design trade-off rules. In this article we consider two different approaches to finding the transfer functions from specific circuit models that are used in the three-port transducer networks. The first approach makes use of a special property of the circuit models introduced in Sections V,B and C. When electrode mass/electrical loading is negligible, either model yields an enlightening formula for the principal electroacoustic transfer function, and the derivation and use of this formula are the subject of Section V,E.

This formula contains the array factor described in Section III and also shows exactly how "regeneration distortion" (cf. Section IV,D) manifests itself.

The second approach to finding transfer functions must be used when mass/electrical loading reflections are appreciable. For developing a generalized computer program for filter performance simulation, it is more systematic to use one of the matrices from network theory that relate the terminal variables E_i, I_i. One possible choice that has found frequent usage in the literature is the admittance matrix Y_{ij} defined by

$$I_i = \sum_{j=1}^{3} Y_{ij} E_j, \qquad i = 1, 2, 3. \tag{53}$$

Using these coefficients and the appropriate equations specifying the interconnection of the transducer with other circuit elements, all the terminal variables E_i, I_i can be calculated so that filter performance is determined.

1. *Construction of the Admittance Matrix from Circuit Models*

The specific circuit models described in Sections V,B and C characterize a single electrode in a transducer as a three-port network. The network for the entire transducer is formed by interconnecting the N individual electrode networks according to the block diagram of Fig. 24. If it is desired to characterize these networks by admittance matrices for computer simulation, the straightforward relations between the individual electrode admittance matrices and those of the entire transducer (Smith *et al.*, 1972) may be employed.

2. *Transducer Transfer Functions*

It is convenient to define a set of complex transfer functions T_{ij} by

$$T_{ij} = 2\left(\sqrt{G_i / G_j}\right) E_i / E_j, \tag{54}$$

where E_i is the "voltage" of the transmitted or reflected wave delivered to a load G_i at port i when a Thévenin generator of voltage E_j and a series conductance G_j drives the transducer at port j. By reciprocity, $T_{ji} = T_{ij}$. These functions are normalized so that the power scattering coefficients are

$$p_{ij} = P_i / (P_{\text{avail}})_j = |T_{ij}|^2, \tag{55}$$

where P_i is the power transmitted or reflected from port i and $(P_{\text{avail}})_j$ is the

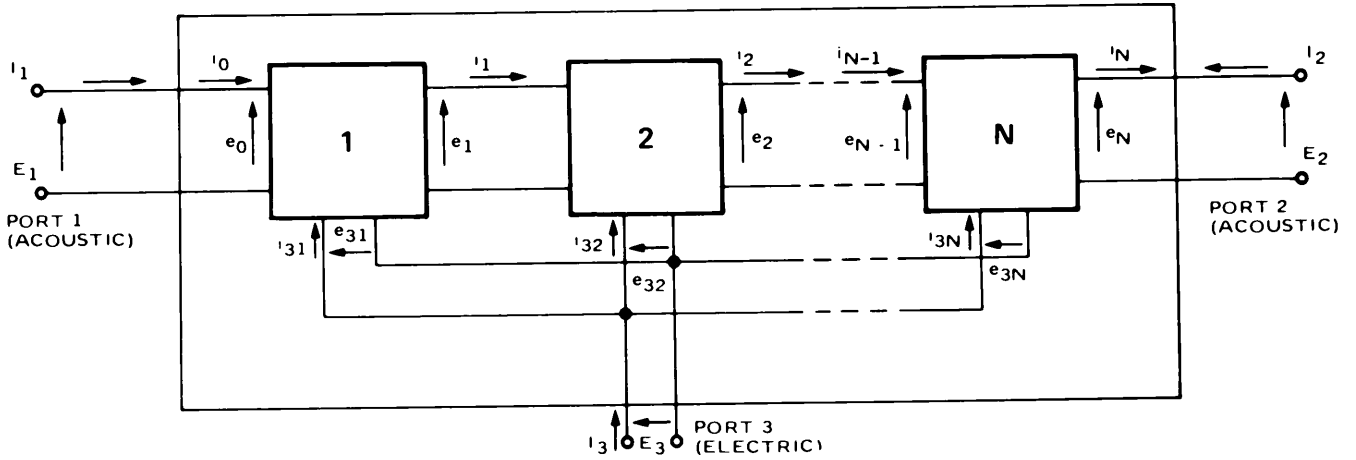


FIG. 24. Block diagram showing the manner in which the three-port network for an entire transducer is formed from the networks for the N individual electrodes. (After Smith *et al.*, 1969, 1972.)

power available from a matched generator at port j. Thus p_{ii} is the fraction of power reflected when power is incident at port i, and p_{ij} $(i \neq j)$ is the fraction of power transmitted to port i when power is incident at port j. For convenience, power ratios are usually given in decibels and thus we define scattering loss in decibels by

$$L_{ij} = -10 \log_{10}(p_{ij}) \quad \text{dB}. \tag{56}$$

The phase of the transfer functions is given by

$$\phi_{ij} = \angle T_{ij}. \tag{57}$$

The two scattering quantities of greatest interest in delay lines and filters are the electroacoustic transfer functions T_{13} and T_{23} and acoustic reflection coefficients T_{11} and T_{22}. These will be discussed in detail in Sections V,E and VI,A.

3. *Transducer Electric Admittance*

A quantity that is important in the calculation of the transfer functions T_{ij} is the admittance measured at electric port 3 when the transducer radiates acoustic waves onto the reflection-free substrate. This quantity, which we denote by Y_{in}, is defined as the ratio I_3/E_3 under the condition that acoustic ports 1 and 2 are terminated in a "matched" conductance G_0 that represents absence of an incoming acoustic wave. Note that this quantity is easily calculated from Eq. (53) and the additional conditions $I_i = -G_0 E_i$, $i = 1, 2$. The resultant admittance will always be conveniently represented by a circuit of the form shown in Fig. 25. Here the capacitor C_T represents the electrostatic capacitance of the transducer as an interdigital electrode structure on a dielectric substrate. There is a (real) radiation conductance G_a, which represents the time average "absorption" of electric energy into the radiated acoustic waves. In addition, since the electroacoustic transduction is a causal process, there is an accompanying radiation susceptance B_a, where the frequency dependence of G_a and B_a are

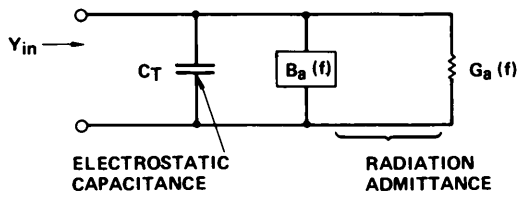


FIG. 25. A parallel-circuit representation for the transducer electric admittance Y_{in}. The radiation admittance branches represent transfer between electric and acoustic power. (After Smith *et al.*, 1969.)

related by the Hilbert transform (Bracewell, 1965)

$$B_a(f) = \text{H.T.}\{G_a(f)\}. \tag{58}$$

Knowledge of the admittance Y_{in} is important for the design of electric matching circuits as well as for the calculation of transfer functions.

B. Crossed-Field Model

The crossed-field Mason circuit (Mason, 1948; Berlincourt *et al.*, 1964) is a three-port circuit that describes the launching and detection of bulk acoustic waves having a single component of acoustic displacement and propagating perpendicular to the electric field in a side-electroded piezoelectric bar. A comparison of the bulk wave geometry of the crossed-field model and the actual surface wave geometry is given in Fig. 26. Because the field geometry in the crossed field bulk wave case is considerably different from the surface wave fields, we might expect the *element factor* $e(f)$ for one electrode to be rather inaccurate when the crossed field model is used to describe the IDT. However, if the crossed field equivalent circuit model for one electrode (Fig. 27) is used for each electrode and these circuits are combined according to the block diagram of Fig. 24, it is reasonable to expect that the transducer transfer function T_{13} will resemble

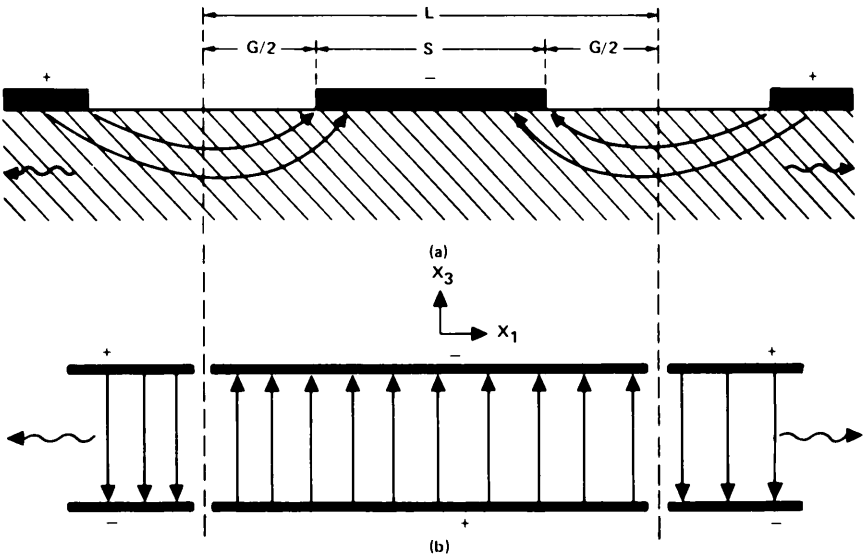


FIG. 26. The geometry represented by the crossed-field bulk-wave model (b) compared to the surface wave transducer geometry (a). (After Smith *et al.*, 1972.)

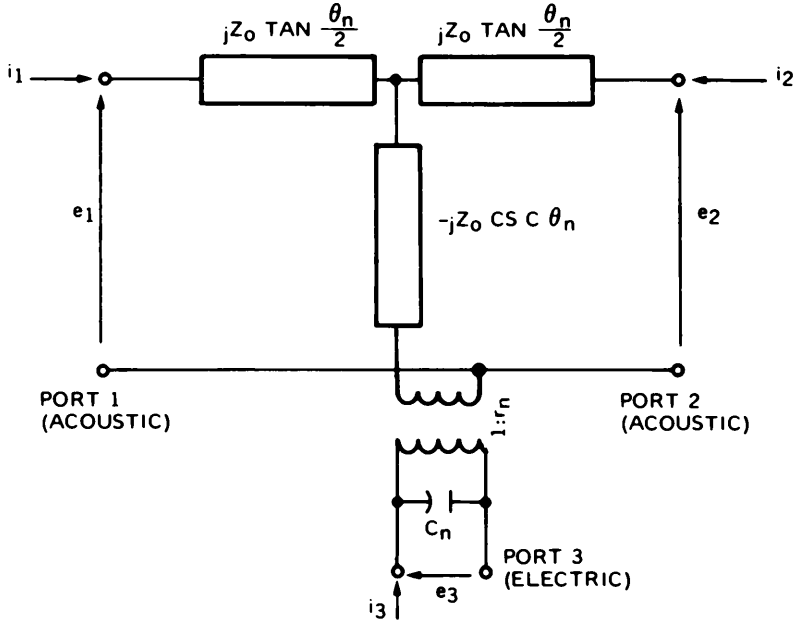


FIG. 27. The three-port equivalent circuit for the crossed-field model for one electrode. (After Smith *et al.*, 1969.)

an *array factor* $A(f)$ that will be rather accurate for operation in the fundamental pass band, over which the element factors $e_n(f)$ do not vary appreciably. The main shortcoming of the crossed-field model is its inability to describe overtone operation because the element factors $e_n(f)$ incur severe errors so far from fundamental passband. Near acoustic synchronism, however, the crossed-field model is capable of describing all the transfer functions defined in Section V,A above (Smith *et al.*, 1969a, 1972).

1. *Three-Port Circuit for Crossed-Field Model*

In Fig. 27, the element C_n is the electrostatic capacitance associated with the charge on the nth electrode. It is given by (Smith, 1976a)

$$C_n = \tfrac{1}{2} w_n \left[\varepsilon_{xx}\varepsilon_{zz} - \varepsilon_{xz}^2 \right]^{1/2} K(q_n)/K(q_n'), \tag{59}$$

where w_n is the acoustic aperture, ε_{xx}, ε_{zz}, and ε_{xz} are the dielectric tensor components of the substrate, and K is the Jacobian complete elliptic integral of the first kind with $q_n = \sin(\frac{1}{2}\pi S_n/L_n)$ and $q_n' = (1 - q_n^2)^{1/2}$.

Note that C_n is an increasing function of the metalization ratio S_n/L_n. The "tee" network between the acoustic ports is the equivalent circuit for a transmission line (acoustic) with wave impedance Z_0 and transit angle $\theta_n = 2\pi f L_n/v$. The electroacoustic transformer ratio r_n is given by

$$r_n = \pm\sqrt{2 f_n k^2 C_n Z_0}\sqrt{\frac{w_n}{w_{\max}}}\left[\frac{K(2^{-1/2})}{K(q_n)}\right], \tag{60}$$

where $f_n = v/2L_n$ and k^2 is the electromechanical coupling constant, which for surface waves is taken as

$$k^2 = 2\,\Delta v/v, \tag{61}$$

where $\Delta v/v$ is the fractional velocity change resulting from placing an infinitesimally thin, massless perfect conductor on the substrate surface (Campbell and Jones, 1968).

The sign in Eq. (60) is that of the electrode polarity. Henceforth we take $Z_0 = 1$ as may be done without loss of generality since the proper dimensions for the electroacoustic coupling are automatically absorbed in the transformer ratio r_n. The aperture of the nth electrode is w_n, while $w_{\max}$ represents the largest aperture in the transducer. Hence, the transformer ratio is proportional to w_n since one factor $w_n^{1/2}$ appears explicitly in Eq. (60) and another factor $w_n^{1/2}$ is implicit via the capacitance C_n.

The three-port circuit for an entire transducer is obtained by interconnecting the Mason circuits of the individual electrodes according to the block diagram of Fig. 24. Explicit expressions for the admittance coefficients $y_{ij}^{(n)}$ are given by Smith *et al.* (1972), as are the recursion relations for finding the transducer admittance coefficients Y_{ij} from the $y_{ij}^{(n)}$. Note that the crossed-field model in this form is applicable to periodic transducers where all L_n and S_n are equal, and also to dispersive transducers where the L_n and S_n are slowly varying functions of the index n. It is also applicable to unapodized transducers (all w_n equal) or to apodized transducers where w_n is a slowly varying function of the index n.

2. *Element Factor for the Crossed-Field Model*

By analogy to the general concept of a one-electrode element factor introduced in Section III,A, we can define an element factor for the crossed-field model as the ratio

$$e_n(f) = (e_2/e_3)\exp(j\theta/2) \tag{62}$$

when the circuit of Fig. 27 is used alone to represent the nth electrode taken from an arbitrary transducer. Elementary circuit analysis shows that

$$e_n(f) = jr_n \sin\left(\tfrac{1}{2}\pi f/f_n\right). \tag{63}$$

If we evaluate the electroacoustic transformer ratio [Eq. (60)] for $S_n/L_n = 0.5$, we find that the ratio of elliptic integrals is unity. Then, evaluating $a_n = e_n(f_n)$ with the choice $Z_0 \equiv 1$, we obtain the same result as for the impulse model,* as was given by Eq. (20). Therefore the crossed-field model specifies the same design relation between w_n and f_n in a dispersive transducer.

The associated impulse response of one electrode is the Fourier transform of $e_n(f)$, viz.,

$$h_n(t) = \frac{r_n}{2}\left[\delta\left(t - \frac{x_n - \frac{1}{2}L_n}{v}\right) - \delta\left(t - \frac{x_n + \frac{1}{2}L_n}{v}\right)\right], \tag{64}$$

which is nearly identical to that for the delta-function model. The differences are (1) the proper dimensions and magnitude are included via r_n, and (2) the impulses are located at the ends of the unit cell of Fig. 26 (i.e., in the centers of the gaps) rather than at the edges of the electrode. Note that this impulse response is proportional to $d(\mathcal{E})/dx$, which consists of a pair of impulses at the ends of the cell. This property of the crossed-field model is a direct consequence of the equations of motion for the bulk wave geometry that it represents. In the wave equation for the elastic displacement (Berlincourt *et al.*, 1964, Eq. 84), the inhomogeneous source term is proportional to $d(\mathcal{E}_\perp)/dx$ where $\mathcal{E}_\perp$ is the electric field which is perpendicular to the x direction. In applying the crossed-field model to Rayleigh waves we are implicitly letting the acoustic force variable E_2 be a generated field that is analogous to the one-dimensional displacement, particle velocity, or force of the crossed-field bulk wave geometry.

We shall retain the analogy that $d(\mathcal{E}_\perp)/dx$ is a source term for Rayleigh wave excitation in formulating the generalized circuit model of Section V,C.

3. *Crossed-Field Model Modified to Describe Electrode Mass/Electrical Loading*

In computer simulations that characterize the transducer via admittance matrices (Y_{ij}), mass/electrical loading reflections are described by modifying the crossed-field model as shown in Fig. 28. Separate crossed-field Mason circuits are used for the electrode and gap regions, and they differ only in that the transmission line wave impedance (Z_m) for the electrode region is slightly different from that of the gap region (Z_0). The discontinuities in wave impedance at the electrode edges give rise to

*Note that C_n as applied to the crossed field model corresponds to $C_s w_n/w_{max}$ as applied to the impulse model.

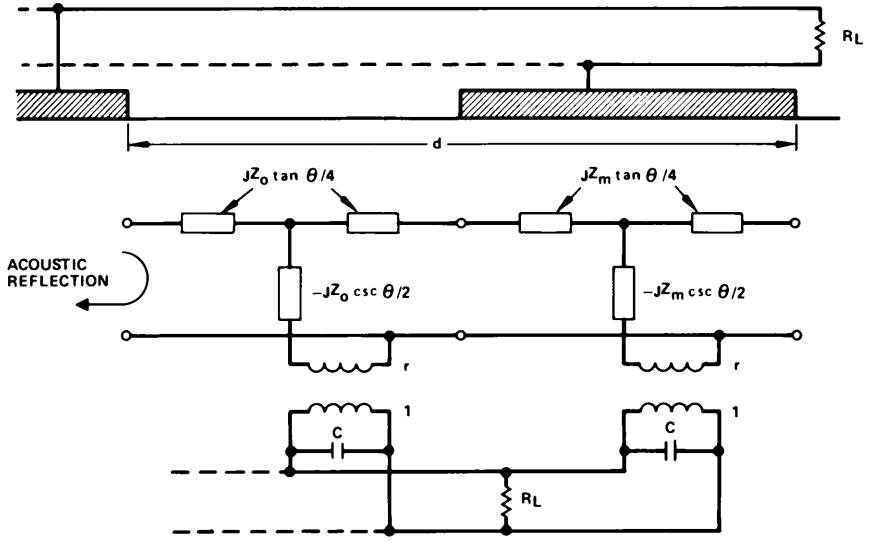


FIG. 28. The crossed-field model modified to represent "mass/electrical loading" reflections by use of different acoustic wave impedances in the electrode and gap regions. (After Smith, 1974.)

acoustic reflections there. The value of Z_m can be chosen to represent mass loading, electrical loading, or both. The value corresponds to the reflection coefficients r_e and r_m given in Eqs. (48) and (49). If the combined reflection coefficient from these effects is r_{tot}, then the appropriate wave impedance mismatch is

$$Z_m/Z_0 \cong 1 - 2|r_{tot}|. \tag{65}$$

The effect of including the wave impedance mismatch is most dramatic for single electrodes near acoustic synchronism where they have a stopband. Even for double electrodes, inclusion of the wave impedance mismatch describes the mass/electrical loading reflections which are smallest at acoustic synchronism and somewhat larger at other frequencies.

4. *Summary of Capabilities of the Crossed-Field Model*

The properties of the crossed-field model may be summarized as follows. The element factor has a form similar to that of the delta-function model, but in dispersive transducers, the relative amplitudes of element factors of different electrodes vary with synchronous frequency as in the impulse model. Therefore, array-factor design is like that of the impulse model. The crossed-field model, unlike the delta-function and impulse models, is a complete three-port circuit model that enables calculation of

all the transducer transfer functions. Thus it allows evaluation of insertion loss, electric impedance (Y_{in}), and triple-transit echoes, and can be used in computer simulations using the filter network described in Section V,D. The required admittance coefficients (Y_{ij}) are given by Smith *et al.* (1969a, 1972), along with expressions for the impedance (Y_{in}) for periodic transducers and some analysis and design results for dispersive transducers.

Since the crossed-field model's element factor does not accurately represent the true distribution of SAW excitation for the geometry of Fig. 26a, the crossed-field model does not describe overtone operation. This shortcoming is overcome by the generalized circuit model.

C. Generalized Circuit Model

1. *Scope of the Model*

The generalized circuit model described in detail by Smith and Pedler (1975) is a three-port circuit model having all the properties described in Section V,A. However, instead of using the bulk wave electric field distribution of the crossed-field model as a source term for acoustic waves, in the generalized circuit model we use electric field distribution actually found in an arbitrary finite interdigital transducer. The generalized circuit model is therefore capable of giving an accurate three-port characterization for both fundamental frequency and higher-harmonic operation. The transducer may contain any number of electrodes, single or double, or in arbitrary polarity sequence, and with arbitrary metalization ratio η, defined with reference to Figs. 26 and 29 as $\eta = S/L$.

2. *Electric Source Term for Acoustic Waves*

The electric fields in the region of any one electrode obviously depend on the polarities and geometry of nearby electrodes. For the generalized circuit model, we make the approximation that the fields in the region of any one electrode depend only on the dimensions and polarities of that electrode and its nearest and next-nearest neighbors (*local environment*). It is also assumed that the electrode dimensions S and L and the aperture w vary slowly enough from one electrode to the next that all electrodes within the local environment may be assumed to have identical dimensions. Therefore, a single metalization ratio and a sequence of polarities describe the local environment and determine the electric fields under a given electrode. Obviously there is a finite number of polarity combinations, with end electrodes and next-to-the-end electrodes being special cases. By solving a set of electrostatic boundary value problems containing all possible local environments, we can develop expressions for the electric

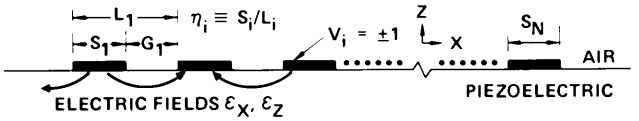


FIG. 29. Schematic diagram of the electrodes, coordinates, applied voltages, and electric fields in the electrostatic boundary value problem whose solution yields the generalized circuit model. (After Smith and Pedler, 1975.)

fields under any one electrode in any environment. In this way we may determine an element factor and a three-port circuit separately for each electrode in any transducer.

In order to establish formulas for the element factor and admittance coefficients for an electrode in the generalized circuit model, we set up Laplace's equation with potentials $\pm V$ specified on the electrodes as shown in Fig. 29. The coordinate along the acoustic propagation direction is x and that normal to the surface is z. The electrodes are assumed to be infinitesimally thin perfect conductors. We assume that the electric field component that is a source term for acoustic waves is $\mathscr{E}_z(x, z=0)$, which is zero in the gaps and given by $\{-\epsilon_{zz}[1+\epsilon_0(\epsilon_{xx}\epsilon_{zz}-\epsilon_{xz}^2)^{-1/2}]\}^{-1}$ times the surface charge density $\sigma(x)$ in the electrode regions. Therefore we need to solve for $\sigma(x)$ the boundary value problem

$$V(x,0) = c_0 \int_{\text{array}} \ln(x-x')^2 \sigma(x')\,dx' \tag{66}$$

where $c_0 = \{2\pi[(\epsilon_{xx}\epsilon_{zz}-\epsilon_{xz}^2)^{1/2}+\epsilon_0]\}^{-1}$, with zero total charge:

$$0 = Q = \int_{\text{array}} \sigma(x)\,dx \tag{67}$$

and with potentials $V_i = \pm 1$ specified on the electrodes. Smith and Pedler (1975) have established the solution to this boundary problem in the following manner. Equations (66) and (67) were solved numerically for $\sigma(x)$ in a finite array for various different values of the metalization ratio η and various combinations of electrode polarities. Then, by using a two-dimensional least-squares fit, a "universal" set of expansion coefficients $\alpha_{nm}^{(p)}$ were found so that the charge density σ can be expressed as

$$\sigma(x,\eta_i)\big|_{i\text{th electrode}} = \frac{1}{c_0}\sum_{m=0}^{5}\left(\eta_i-\tfrac{1}{2}\right)^m \sum_{n=0}^{5}\alpha_{nm}^{(p_i)}\frac{T_n\left(2x^{(i)}/\eta_i L_i\right)}{\left[(\eta_i L_i/2)^2-(x^{(i)})^2\right]^{1/2}}. \tag{68}$$

Here T_n is the Chebyshev polynomial of order n and the expansion over $n=0,\ldots,5$ gives the dependence of σ on x. The index m runs from 0 to 5

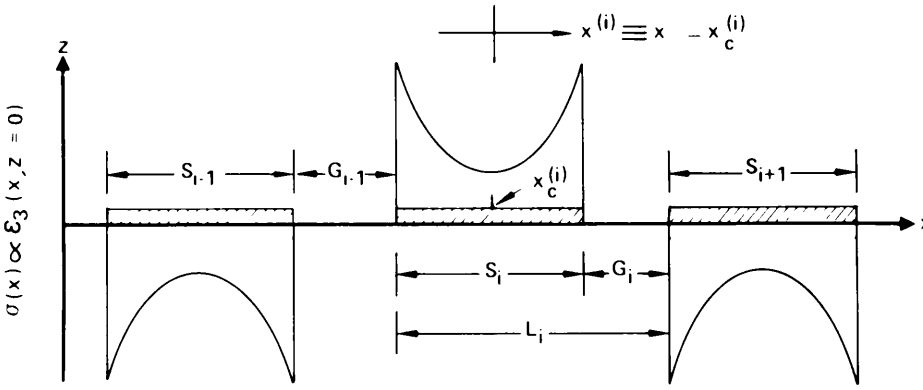


FIG. 30. A schematic graphical representation of the surface charge density on a few electrodes of a transducer. This charge is the distributed electroacoustic source in the generalized circuit model. The metalization ratio of the ith electrode is $\eta_i \equiv S_i/L_i$.

and the polynomial expansion in $(\eta_i - \frac{1}{2})^m$ gives the dependence of σ on the metalization ratio, which varies sufficiently slowly through the transducer that it may be assumed constant over the few electrodes that make up the local environment. The index p_i is a label for the particular sequence of electrode polarities in the local environment around the ith electrode. Since the local environment is defined to extend only to the nearest and next-nearest neighbor electrodes, it contains at most five electrodes (when electrode i is an interior electrode) and at least three electrodes (when electrode i is an end electrode). Inspection reveals that 22 independent polarity sequences are sufficient to describe any local environment, and accordingly, Smith and Pedler (1975) have tabulated the 22 independent* six-by-six matrices of $\alpha_{nm}^{(p)}$ coefficients needed to complete Eq. (68). In order to express the charge density on an electrode, one examines the polarities of all electrodes in the local environment, selects the proper set of coefficients $\alpha_{nm}^{(p_i)}$ from the aforementioned tables, and applies Eq. (68).

A graphical representation of the charge density distribution is shown in Fig. 30. Note that a coordinate $x^{(i)}$ is separately defined with its origin at the center of each electrode in order to facilitate independent circuit model and element factor representation for each electrode. The charge distribution on each electrode has "cusp" singularities at the electrode edges whose form is given by the denominator of Eq. (68).

*The charge density of any polarity sequence not included in the table of 22 matrices is easily found from one of these matrices by applying simple symmetry arguments.

The capacitance contribution from one electrode is found from integrating $\sigma(x)$ on that electrode, with the result

$$C_i = \frac{|Q_i|}{4V} = \frac{\pi^2}{2}\left[\varepsilon_{xx}\varepsilon_{zz} - \varepsilon_{xz}^2\right]^{1/2}\frac{w_i}{w_{\max}}\left|\sum_{m=0}^{5}\alpha_{0m}^{(p_i)}\left(\eta_i - \tfrac{1}{2}\right)^m\right|, \qquad (69)$$

where the local environment around electrode i is described by metalization ratio η_i and polarity sequence $\mathcal{S}_{p_i}$.

3. *Element Factor for Generalized Circuit Model*

We define the element factor for one electrode by analogy to the known property of the crossed-field model, where the element factor is the Fourier transform of the quantity $d(\mathcal{E}_\perp)/dx$. Thus, for the generalized circuit model, the element factor is proportional to the Fourier transform of $\sigma(x)$ and hence of $d[\mathcal{E}_z(x, z=0)]/dx$; specifically,

$$e_i(f) = \tfrac{1}{2}j\beta F_i(\beta, \eta_i), \qquad (70)$$

where $\beta = 2\pi f/v$ is the wave number and

$$F_i(\beta,\eta_i) = \left[\frac{4}{\beta L_i} f_i k_0^2 (\varepsilon_{xx}\varepsilon_{zz} - \varepsilon_{xz}^2)^{1/2}\frac{w_i}{2}\right]^{1/2}(\pi^2 L_i)\left(\frac{w_i}{w_{\max}}\right)^{1/2}$$
$$\cdot \sum_{m=0}^{5}\left(\eta_i - \tfrac{1}{2}\right)^m \sum_{n=0}^{5}\alpha_{nm}^{(p_i)}(-j)^n J_n\left(\frac{\beta\eta_i L_i}{2}\right), \qquad (71)$$

J_n being the cylindrical Bessel function of order n. Equations (69)–(71) with the coefficients $\alpha_{nm}^{(p)}$ tabulated by Smith and Pedler (1975) describe any electrode in the generalized circuit model. The factor in square brackets in Eq. (71) is closely related to the transformer ratio of the crossed-field model. This factor is inversely proportional to βL_i because the electric field penetration depth is proportional to L_i while the penetration depth of surface elastic waves varies inversely as β (Bahr and Lee, 1973). The substrate coupling constant k_0^2 is taken as $2\Delta v/v$ where $\Delta v/v$ is the fractional velocity change calculated by Campbell and Jones (1968). The dimension L_i is one-half acoustic wavelength at the frequency $f_i = v/2L_i$. The element factor is normalized so that the generalized circuit model agrees with the results of Bahr and Lee (1973) and of Auld and Kino (1971) when applied to a periodic transducer with many single electrodes.

4. *Three-Port Circuit for the Generalized Circuit Model*

We use the circuit introduced by Krimholtz (1972) and shown in Fig. 31 for one electrode of the generalized circuit model because, as explained

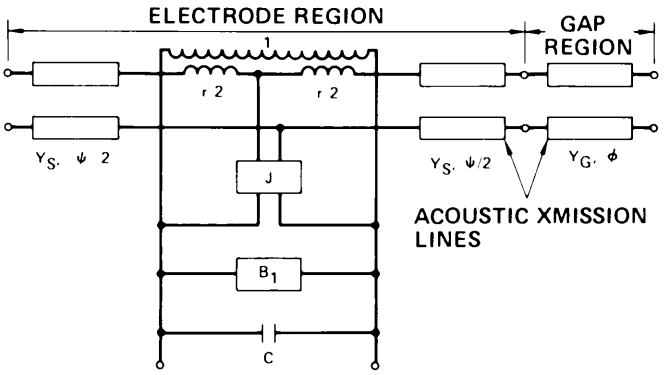


FIG. 31. An equivalent circuit for the generalized circuit model of one electrode. $\psi = S/V_S$, $\phi = G/V_G$. (After Smith and Pedler, 1975.)

by Bahr and Lee (1973), Krimholtz (1972), and Leedom *et al.* (1971), it is valid for a source term that is an arbitrary scalar function of x and in particular for $\mathfrak{S}_z(x, z = 0)$. The admittance coefficients (y_{ij}) for this circuit in the generalized circuit model are given by Smith and Pedler (1975) as functions of C_i and of $F_i(\beta, \eta_i)$. The properties and values of the elements r, J, and B_1 are also described by Smith and Pedler (1975), Bahr and Lee (1973), Krimholtz (1972), and Leedom *et al.* (1971).

Our circuit model computer program includes an algorithm for examining the polarities of the electrodes in the local environment of an electrode and selecting the proper set of α's. Using the element factor given by Eq. (70) and the usual definition of the array factor [Eq. (12)], all the transducer transfer functions may be evaluated from the formulas of Section V,E provided $Y_S = Y_G$. In computer simulations, however, we use the admittance coefficients y_{ij} given by Smith and Pedler (1975) and it is not required that $Y_S = Y_G$, so that mass/electrical loading can be described.

D. Computer Simulation of SAW Filters

1. *Two-Port Network for a Two-Transducer Filter*

From either the crossed-field or generalized three-port circuit model, it is straightforward to construct a two-port network that represents a filter or delay line consisting of two transducers, a delay between transducers, and electric matching circuits (Smith, 1976a). The general network for doing this is shown schematically in Fig. 32. This circuit is useful for computer programs for finding filter transfer functions. The acoustic

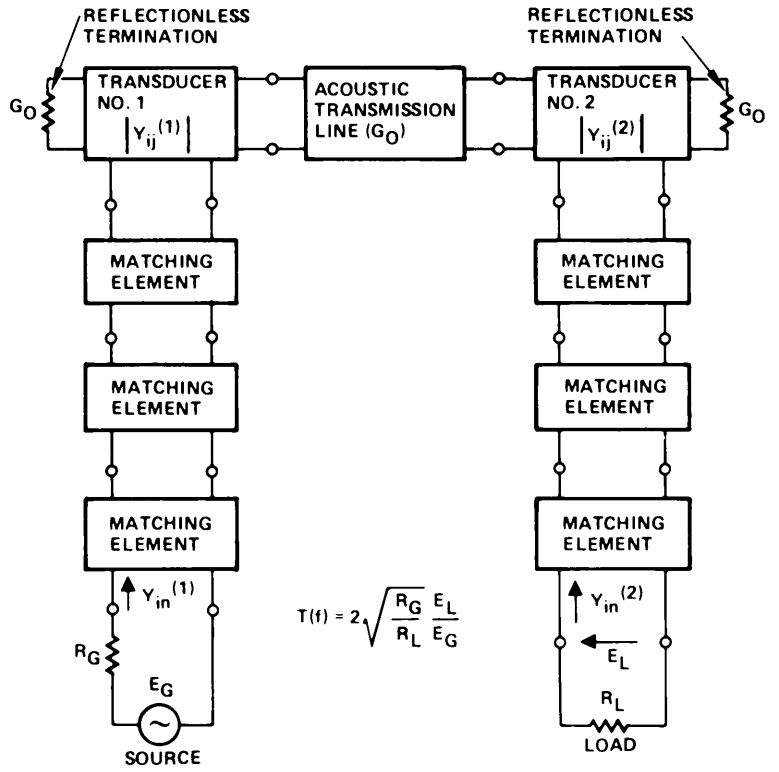


FIG. 32. A network representation for a SAW filter having two transducers with arbitrary matching circuits. (After Smith, 1976a.)

transmission line represents the delay between transducers and may contain acoustic propagation loss. Any number of two-port matching elements may be applied to either transducer. All that is required for each matching element is that its *ABCD* matrix be specified. Typical matching elements are

series or shunt inductor,
series or shunt capacitor,
series or shunt resistor,
ideal transformer.

The transfer function of the filter is the complex voltage ratio

$$T(f) = 2\left(\sqrt{R_G/R_L}\right)E_L/E_G, \tag{72}$$

while the insertion loss and phase are

$$\mathrm{IL}(f) = -10\log_{10}|T(f)|^2 \tag{73}$$

and

$$\phi(f) = \underline{/T(f)}. \tag{74}$$

2. *Illustrative Results of Computer Simulation for SAW Filters Using Three-Port Transducer Networks*

a. *Coupling for single electrodes with arbitrary metalization ratios up to the eleventh harmonic.* The first case investigated with the generalized circuit model is a periodic, single-electrode transducer with 20 electrodes, whose interior electrodes behave like those in an infinite transducer. A convenient measure of the coupling is the effective k^2 defined by

$$k_{\mathrm{eff}}^2 = \pi Q_{\mathrm{r}}^{-1}/2N, \tag{75}$$

where N is the number of electrodes in the transducer and Q_{r} is the radiation Q, given by

$$Q_{\mathrm{r}} = 2\pi f_M C_{\mathrm{T}}/G_{\mathrm{a}}(f_M) \tag{76}$$

i.e., the ratio of capacitive susceptance to the radiation conductance at the Mth harmonic frequency f_M. The effective k^2 is proportional to the intrinsic substrate coupling $k_0^2 = 2\Delta v/v$. Therefore, in Fig. 33 we plot k_{eff}^2/k_0^2 versus metalization ratio η for the harmonics $M = 1, 3, 5, 7, 9, 11$, for 20 (many) single electrodes. These curves are in agreement with the

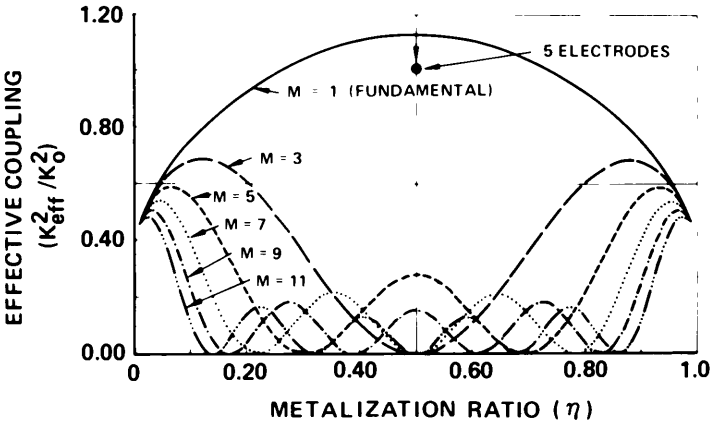


FIG. 33. Effective coupling versus metalization ratio for the first eleven harmonics of a periodic transducer with many single electrodes. $k_{\mathrm{eff}}^2 = \pi/2NQ_{\mathrm{r}}$. (After Smith and Pedler, 1975.)

work of Engan (1969) and with the formula given by Bahr and Lee (1973), where experimental points from various authors were also found to agree reasonably well with the theory for $M = 1, 3, 5$. We reemphasize that the effective k^2 at the harmonic frequencies is only one representative result from the circuit model that can calculate all the transfer functions T_{ij} at all frequencies. One point is provided in Fig. 33 to illustrate end effects in short transducers. The effective k^2 for a five-electrode transducer with $\eta = 0.5$ operating at the fundamental frequency is shown by the large dot labeled "5 electrodes." There is an 8% reduction in k^2_{eff} caused by end effects in the five-electrode transducer. The shape of the insertion loss function (not shown) also deviates somewhat from the usual $(\sin x/x)^2$ shape because of end effects.

b. *Coupling for double electrodes with arbitrary metalization ratios up to the eleventh harmonic.* Figure 34 shows for double electrodes the effective k^2 curves that are analogous to those given in Fig. 33 for single electrodes. The curves were again calculated for a 20-electrode transducer, so that the interior electrodes behave like those in an infinite transducer. Note that the effective k^2 is the same for $M = 1$ and 3, for $M = 5$ and 7, and for $M = 9$ and 11. A similar result (coalescence) was found for the quantity $|\beta F(\beta)|$ for an infinite array of double electrodes by Hartmann and Secrest (1972). Again, the reduction in k^2_{eff} caused by end effects on a five-electrode transducer with $\eta = 0.5$ is indicated by the heavy dot for $M = 1$ (fundamental) and for $M = 3$.

c. *Transducer capacitance for single and double electrodes.* Equation (69) of the generalized circuit model gives the capacitance contribution

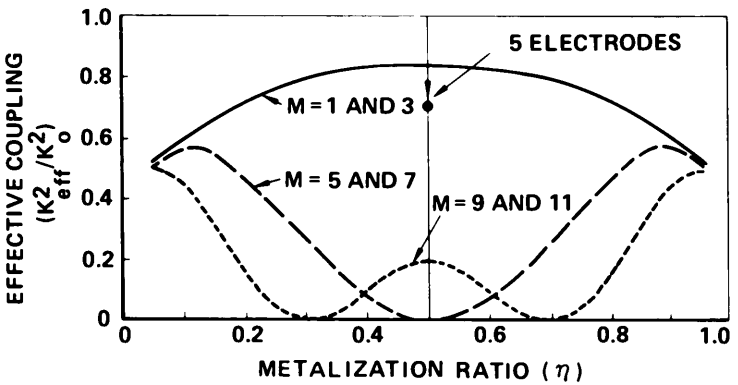


FIG. 34. Effective coupling versus metalization ratio for the first eleven harmonics of a periodic transducer with many double electrodes. $k^2_{eff} = \pi/2NQ_r$. (After Smith and Pedler, 1975.)

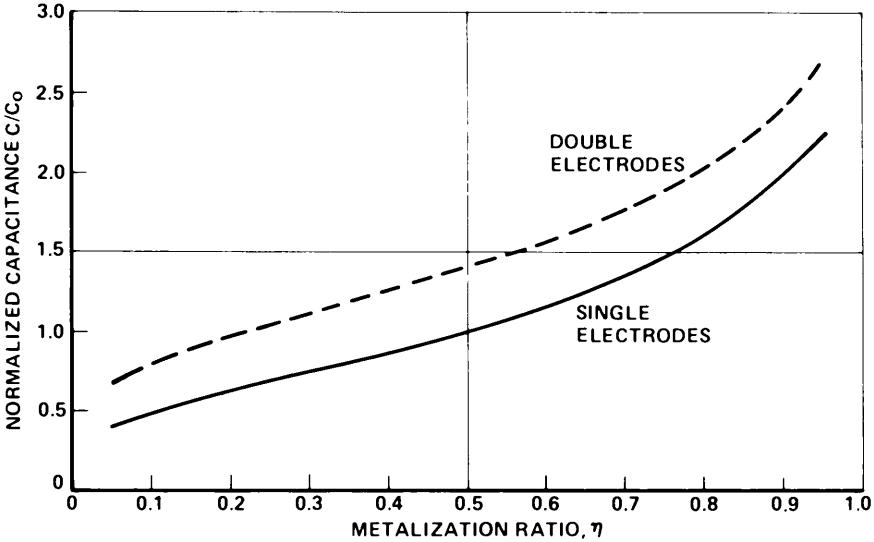


FIG. 35. Capacitance versus metalization ratio for single- and double-electrode transducers, normalized to the value for single electrodes with 50% metalization. (After Smith, 1976a.)

from one electrode in any environment. Figure 35 shows the results of this equation for finding the capacitance per period per unit aperture in infinite arrays of single and double electrodes, as a function of the metalization ratio η. All values are normalized to the value for single electrodes with $\eta = 0.5$, namely $C_0 = (\varepsilon_{xx}\varepsilon_{zz} - \varepsilon_{xz}^2)^{1/2}$. Note that for $\eta = \frac{1}{2}$, one period of double electrodes has 1.41 times the capacitance of one period of single electrodes.

d. *Theoretical and experimental results for an efficient ninth-harmonic transducer.* Two devices having transducers of 10 double electrodes were designed, fabricated, and tested with two goals: (1) to verify the curves in Fig. 34, and (2) to achieve efficient operation at a high harmonic (in this case the ninth harmonic). According to the data of Fig. 34, the effective k^2 for the ninth and eleventh harmonics has a relative maximum for $\eta = 0.5$ and absolute maxima near $\eta = 0.1$ and 0.9. Consequently, two devices were designed with planned metalization ratios $\eta = 0.5$ and $\eta = 0.9$. In the actual devices, measured values of η were 0.56 ± 0.05 and 0.88 ± 0.05. Theoretical curves for the insertion loss of each delay line were calculated, and it was found that $\eta = 0.58$ and $\eta = 0.90$ give best agreement with the measured results.

Figure 36 (top: "theory") shows the calculated insertion loss function for the $\eta = 0.58$ delay line. Note that the insertion loss itself is not

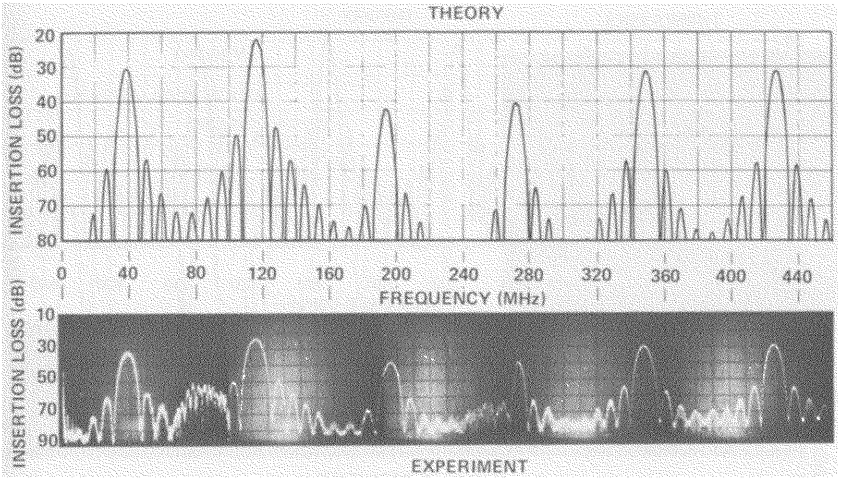


FIG. 36. Insertion loss (generalized circuit model theory and experiment) for a double-electrode-transducer delay line with 58% metalization. (After Smith and Pedler, 1975.)

$-20\log_{10}k_{\text{eff}}^2$, but is $2L_{13}$ where L_{13} is given by Eqs. (54)–(56). The transducer aperture was chosen so that $2\pi fC_T = G_L$ near the ninth harmonic, a design that minimizes the ninth-harmonic insertion loss at the expense of higher insertion loss at other harmonics.

The measured insertion loss for this device is shown in Fig. 36 (bottom: "experiment"). The agreement between theory and experiment is generally excellent with all harmonic insertion losses agreeing within 2 dB or better, and all sidelobes at approximately the correct level with the proper sense of tilt (up or down) between the nearest sidelobes at each harmonic. That the metalization ratio is somewhat larger than intended ($\eta = 0.58$ instead of 0.50) also makes the ninth and eleventh harmonics less efficient than planned.

Aside from the fact that the theory does not account for bulk waves (note bulk-wave response near 80 and 250 MHz), the agreement between theory and experiment is excellent. Most important is the correct prediction of relative harmonic levels, which is most significant for overtone operation. This capability is totally absent from the ordinary crossed-field bulk-wave model.

Agreement between theory and experiment is even better for the 90% metalization-ratio transducers, as evidenced by the data of Fig. 37. The effective k^2, from Fig. 34, is roughly equal for the first eleven harmonics near $\eta = 0.9$, and the ninth harmonic k^2 is more than three times higher at $\eta = 0.9$ than at $\eta = 0.58$. Consequently, this device has about 10 dB less

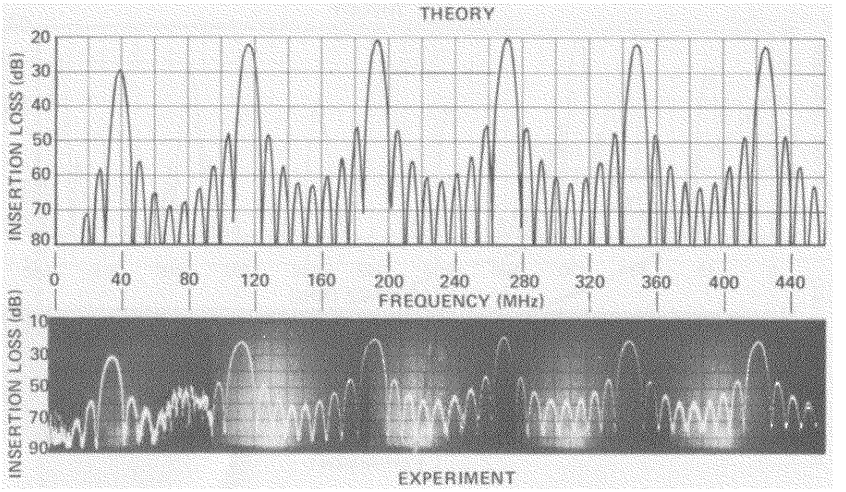


FIG. 37. Insertion loss (generalized circuit model theory and experiment) for a double-electrode-transducer delay line with 90% metalization. (After Smith and Pedler, 1975.)

insertion loss at the ninth harmonic than the $\eta = 0.58$ device. Although fabrication of transducers with $\eta = 0.9$ is difficult at higher frequencies, the data of Figs. 36 and 37 confirm the accuracy of the generalized circuit model. Furthermore, we have achieved efficient operation at the ninth and eleventh harmonics. By tuning and matching at the ninth harmonic, the insertion loss can theoretically be reduced to the 6-dB bidirectional loss. The most important property, however, is the insertion-loss/bandwidth trade-off (see Section VI,A) fixed by the effective k^2 given in Fig. 34. For double electrodes at $\eta = 0.9$, the effective k^2 is slightly less than half the intrinsic k^2, i.e., $k_0^2 = 2\,\Delta v/v$.

e. *Theoretical and experimental results for a phase-reversal transducer.* An example of a device with a nonalternating polarity sequence is a 13-electrode phase-reversal transducer (PRT) (Bristol, 1972) where the last two electrodes on each end are reversed in polarity with respect to the normal alternating sequence in the central nine electrodes. We have fabricated two delay lines using 13-electrode PRT's on YZ lithium niobate with a fundamental frequency of 30 MHz. The first such delay line has single electrodes with $\eta = 0.5$ and its theoretical and experimental insertion loss functions are compared in Fig. 38. The generalized circuit model correctly explains the following features not found in the crossed-field model predictions: (1) absence of a third harmonic response, (2) roll-up of about 2 dB in the main passband, and (3) relatively high level of the first sidelobes.

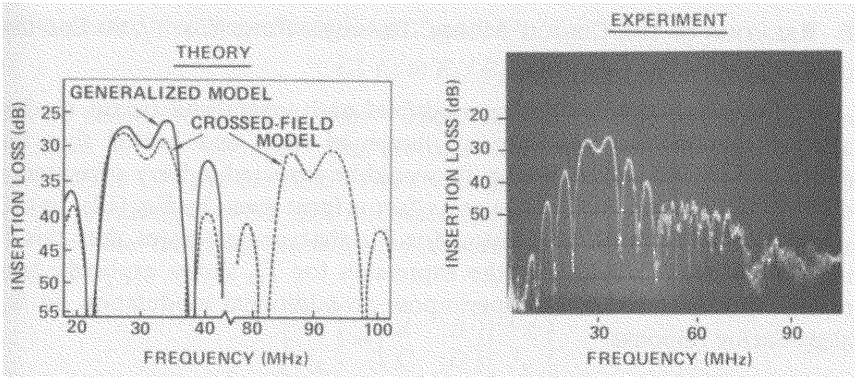


FIG. 38. Crossed-field and generalized circuit model theories and experimental results for the insertion loss of a delay line with a pair of 13-single-electrode phase-reversal transducers. (After Smith and Pedler, 1975.)

The same comparison is made for a PRT with 13 double electrodes in Fig. 39. The generalized circuit model is superior in much the same way as for single electrodes. First, it correctly predicts a strong third-harmonic response: the insertion loss is slightly lower at the third harmonic than at the fundamental, while the crossed-field model predicts the opposite. Second, it correctly predicts a roll-up in both the fundamental and third-harmonic passbands, although somewhat less accurately at the third harmonic than at the fundamental. Third, the generalized circuit model gives a good prediction of the nearest sidelobe levels while the crossed-field model does not.

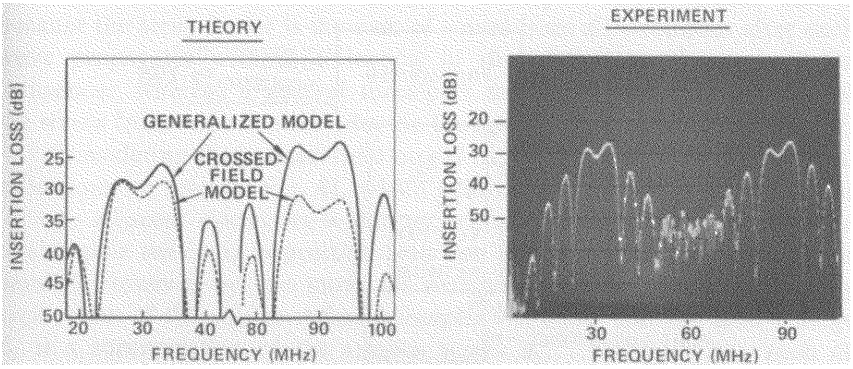


FIG. 39. Crossed-field and generalized circuit model theories and experimental results for the insertion loss of a delay line with a pair of 13-double-electrode phase-reversal transducers. (After Smith and Pedler, 1975.)

E. Relation of the Circuit Model Transfer Function T_{13} to Simpler Array-Factor Analyses

In the discussion of the crossed-field and generalized circuit models above, we presented the element factor of each and hinted that the electroacoustic transfer function T_{13} would be expected to have a frequency dependence similar to the basic array factor form given by Eq. (12). In this section we develop an expression for T_{13} that demonstrates that this is indeed the case. In addition, the expression for T_{13} shows explicitly how regeneration distortion can superimpose an additional modulation on the array-factor response.

1. *Transducer and Circuit Factors*

Figure 40 shows the network for an interdigital transducer that is driven by an electric generator and is launching surface waves onto a reflectionless substrate. We have already defined the electroacoustic transfer function for this circuit as

$$T_{13}(f) = \left(\sqrt{G_0/G_L}\right)E_1/E_3. \tag{77}$$

Here we find it convenient to rewrite this expression

$$T_{13}(f) = \left(\sqrt{G_0/G_L}\right)a(f)/C(f), \tag{78}$$

where, with reference to Fig. 40,

$$a(f) \equiv E_1/E_3. \tag{79}$$

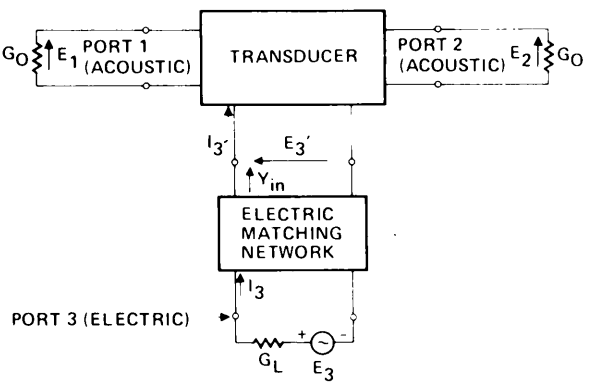


Fig. 40. Network representation for finding the electroacoustic transfer function of a transducer being driven by an electric generator and launching acoustic waves. (After Smith, 1976a.)

will be called the "transducer factor," and

$$C(f) \equiv E_3/E_{3'} \tag{80}$$

will be called the "circuit factor." These names are used because $a(f)$ is the ratio of two "voltages" that are measured directly at two transducer ports, while $C(f)$ is the ratio of two voltages at points that are separated by the matching network and internal impedance of the source.

Let the matching network be characterized by an $ABCD$ matrix defined by

$$\begin{bmatrix} E_3 \\ I_3 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E_{3'} \\ I_{3'} \end{bmatrix}. \tag{81}$$

Then the circuit factor is given by

$$C(f) = A + B \cdot Y_{\text{in}} + R_L(C + D \cdot Y_{\text{in}}), \tag{82}$$

where $R_L = G_L^{-1}$ and Y_{in} is the transducer admittance labeled in Fig. 40. Note that the "circuit" factor is appropriately named because Y_{in} is the only quantity appearing in $C(f)$ that depends on the transducer geometry. Often Y_{in} is dominated by the electrostatic capacitance C_T of Fig. 25, so that the dependence of $C(f)$ on transducer geometry is extremely weak.

2. *Applicability of the Simple Array-Factor Concept to the Crossed Field and Generalized Circuit Models*

The key step in developing the expression for T_{13} to its most useful form is the demonstration that the "transducer factor" $a(f)$ is (to within a multiplicative constant) just the array factor defined by Eq. (12). Note that this would not be true for an arbitrary three-port circuit model. This is because the array factor is the sum of waves from all electrodes after each wave propagates without reflection or disturbance to the end of the transducer, whereas a general three-port model allows for interaction of the waves from various electrodes via the interconnection shown in Fig. 24.

Considering the crossed-field model in the absence of mass/electrical reflections we refer to Fig. 41, which shows the three-port Mason circuits for two adjacent electrodes within an arbitrary transducer. Normally, the double-pole switch is in position 2 so that all electrodes are connected in parallel. However, we can move the switch to position 1 for the following application of the superposition principle. We know that the total voltage E_1 is a superposition of the various waves $a_1, \ldots, a_n, a_{n+1}, \ldots$, each of which is generated when the voltage $E_{3'}$ is applied to one electrode at a time with zero voltage applied to the other electrodes. Placing the switch in position 1 applies the voltage $E_{3'}$ to electrode n and a short circuit (zero voltage) to electrode $n + 1$. The effect of this condition is to short-circuit

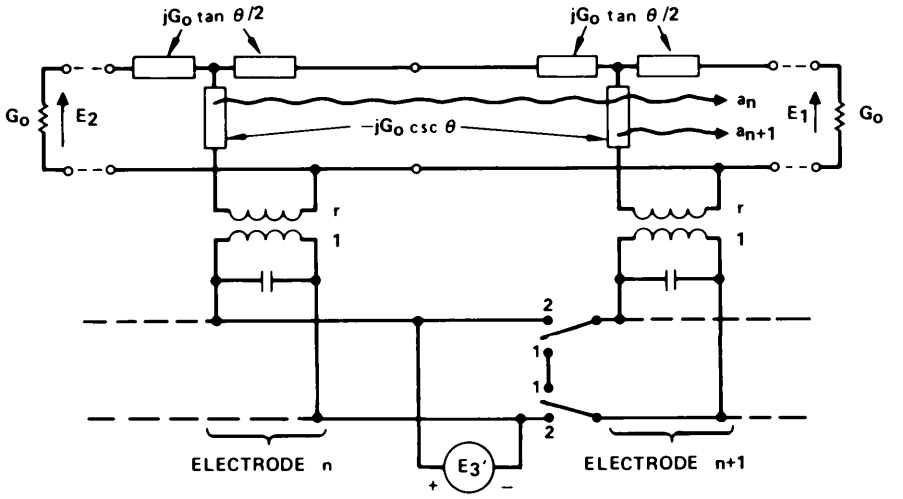


FIG. 41. Crossed-field circuit for two adjacent electrodes in a transducer. The movable switch facilitates an application of the superposition principle to show that the network "transducer factor" is in fact the same as the array factor introduced earlier.

both the primary and secondary windings in the electroacoustic transformer for electrode $n+1$. This completely decouples the acoustic transmission line from the electric circuit, so that the wave a_n, which has been launched by electrode n, passes through electrode $n+1$ without reflection or interaction. It follows from the superposition principle that the transducer factor defined in Eq. (79) is a sum of individual electrode waves a_i that propagate to the end of the transducer as if all other electrodes were absent. Hence, *for the crossed-field model without mass/electrical loading, the transducer factor defined in Eq.* (79) *reduces to the array factor form given by Eq.* (12), namely,

$$a(f) \equiv \frac{E_1}{E_{3'}} = \sum_{n=1}^{N} e_n(f) \exp(-j2\pi f t_n). \tag{83}$$

Therefore, the transducer factor will henceforth be renamed the array factor and written in the form given by Eq. (12). Furthermore, a similar demonstration is valid if the generalized circuit model without mass electrical loading (i.e., Fig. 31 with $Y_S = Y_G$) is used instead of the crossed field model.

It also follows from the above argument that a complete interdigital transducer will not reflect an acoustic wave incident at port 1 or 2 if a short circuit is applied to port 3′ provided that mass/electrical loading is ignored. It has been demonstrated experimentally (Smith, 1974) that this is

the case and one may argue that such a property is indeed requisite for a good three-port model of the interdigital transducer. To the latter end, merely note that a shorted transducer is equivalent to a grating where all electrodes are connected to both bus bars. This structure cannot sustain an electric potential that varies with periodicity comparable to a SAW wavelength so no reflection would be expected. These arguments rule out many possible candidate three port models, such as the "in-line" model (Smith *et al.*, 1969a) for the interdigital transducers.

3. *The Electroacoustic Transfer Function* T_{13}

a. *Normalized array factor and expression for* $G_a(f)$. In order to write the expression for T_{13} in its most useful form we first seek a suitable normalization for the array factor. To do this we first note that, by the same argument given in Section V,E,2, $|E_2| = |E_1|$ in Fig. 40, so that equal power is radiated from ports 1 and 2. Indeed we note that T_{13} and T_{23} differ only in the sign of their phase, i.e.,

$$T\left\{\begin{matrix}1\\2\end{matrix}\right\}_3 = \frac{1}{C(f)} \sum_{n=1}^{N} e_n(f) \exp(\mp j2\pi f t_n), \tag{84}$$

where the sum is of course just the array factor. Now, assuming the transducer to be internally lossless, we equate the time-average power radiated into acoustic waves to the time-average electric power delivered to terminals 3′. The total acoustic power (P_{ac}) radiated in either direction is $G_0|E_1|^2$, and using $G_0 \equiv 1$, and Eq. (83) we have

$$P_{ac} = |E_{3'}|^2 \left| \sum_{n=1}^{N} e_n(f) \exp(-j2\pi f t_n) \right|^2. \tag{85}$$

The time-average electric power delivered to the transducer at the electric terminals 3′ is

$$(P_{in})_{3'} = |E_{3'}|^2 G_a(f) \tag{86}$$

since only the real part of Y_{in}, namely, $G_a(f)$, absorbs power on a time averaged basis. Now, by equating $(P_{in})_{3'}$ to $2P_{ac}$ we obtain the following expression for $G_a(f)$:

$$G_a(f) = 2\left| \sum_{n=1}^{N} e_n(f) \exp(-j2\pi f t_n) \right|^2 \equiv 2|a(f)|^2. \tag{87}$$

The normalized array factor that we seek is then

$$A(f) = \sqrt{\frac{2}{G_a(f_0)}} \sum_{n=1}^{N} e_n(f) \exp(-j2\pi f t_n), \tag{88}$$

which satisfies, at a conveniently chosen reference frequency f_0,

$$|A(f_0)| = 1. \tag{89}$$

b. *Radiation and load Q parameters.* The next step toward writing the most convenient expression for T_{13} is the definition of two Q parameters that are easily estimated without computation or a detailed knowledge of the transducer geometry. The first of these will be called the *radiation Q* and is defined by

$$Q_r \equiv 2\pi f_0 C_T / G_a(f_0). \tag{90}$$

It is the ratio of the transducer capacitive susceptance to the acoustic radiation conductance at a convenient reference frequency f_0. It is a measure of electroacoustic coupling strength: in the limit of very weak acoustic radiation the transducer is nearly a pure capacitor and Q_r is very large, while for strong radiation, Q_r is small and G_a may even become the dominant component of Y_{in}. It is easily shown for specific transducer geometries and empirically determined for arbitrary geometries that Q_r may be estimated from

$$Q_r \cong (\pi/4k^2)\,\Delta f/f_0. \tag{91}$$

Here k^2 is the electromechanical coupling constant of the substrate and $\Delta f/f_0$ is the fractional half-power bandwidth of the array factor, so that Q_r may be estimated for any transducer before detailed design begins.

The second Q parameter will be called the *load Q*, defined by

$$Q_L = 2\pi f_0 C_T / G_L. \tag{92}$$

It thus measures the ratio of transducer and load admittances and is a mismatch parameter. As with the radiation Q, practically no a priori knowledge of the transducer geometry is needed to calculate the load Q because it depends only on the capacitance. If necessary, the final capacitance value can be forced equal to its preliminary estimate with the appropriate choice of a maximum aperture following the detailed array design.

c. *Interpretation of the transfer function T_{13}.* Combining the defining expression, Eq. (77) or Eq. (78) for T_{13} with Eq. (87) for $G_a(f)$, Eq. (88) for the normalized array factor $A(f)$, and Eqs. (90) and (92), which define Q_r and Q_L, we obtain the following convenient form for T_{13}:

$$T_{13}(f) = \left(\sqrt{2Q_L/Q_r}\right) A(f)/C(f). \tag{93}$$

The expression for T_{13} in this form has the following useful properties:

(1) $A(f)$ is the normalized, dimensionless array factor given by Eq. (88). Its magnitude is near unity and the array-factor design principles given in Section III are applicable.

(2) $C(f)$ is the dimensionless circuit factor defined by Eq. (82). Using ordinary circuit analysis to find the $ABCD$ matrix parameters we can calculate $C(f)$ for an arbitrary matching network.

(3) The circuit factor $C(f)$ causes the frequency dependence of T_{13} to deviate from the array factor. Usually $C(f)$ is slowly varying in comparison with the array factor. Figure 42a shows schematically the amplitude versus frequency for the array and circuit factors in a typical bandpass filter transducer with a tuning inductor in the matching network.

(4) The phase response for the array and circuit factors is shown schematically in Fig. 42b. Note that the phase of the array factor, for this nondispersive filter example, is strictly linear, a consequence of the non-minimal-phase property of array factors and transversal filters, whose transfer functions have zeros but no poles. The circuit factor contributes the only phase nonlinearity for nondispersive bandpass filters. The circuit factor in general represents lumped elements—both in the matching circuit and also the transducer capacitance via the admittance Y_{in}. As such, the circuit factor can have both poles and zeros.

(5) The circuit factor is the sole manifestation of regeneration distortion (cf. Section IV,D) in the transfer function. This important fact can be used to provide compensation for the elimination of regeneration distortion as in Section V,E,4. It is found in Section VI that regeneration distortion can be minimized at a sacrifice in increased insertion loss.

(6) Often the circuit factor has magnitude near unity (e.g., for $Q_L \ll 1$ with the matching network a pair of wires only). Then the insertion loss is given directly from the factor $(2Q_L/Q_r)^{1/2}$ in Eq. (93), so that by using

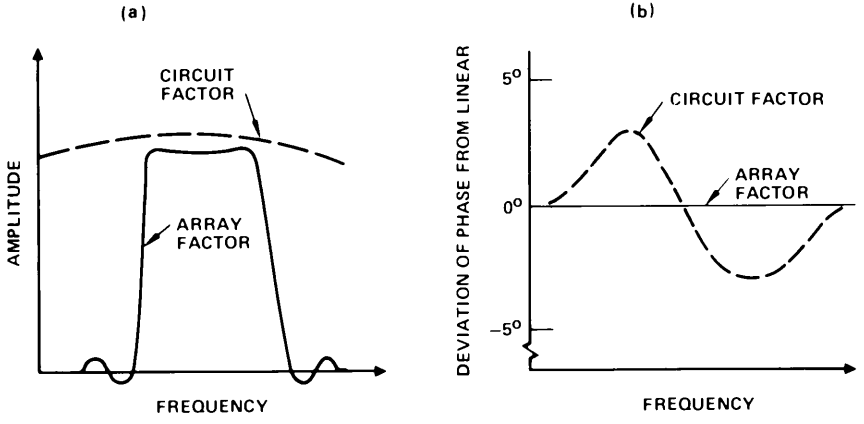


FIG. 42. An illustration of the roles of the array and circuit factors. Curves are typical of nondispersive bandpass filter transducers. (a) Amplitude; (b) phase.

Eqs. (91) and (92), insertion loss can be estimated without detailed knowledge of transducer geometry. Even for more complicated matching circuits, by using Eq. (82), the circuit factor can be separated into the product of a frequency-dependent part with magnitude near unity and a constant part containing Q_L and Q_r. Therefore Eq. (93) can always be manipulated into a form that has a frequency-dependent part whose magnitude is near unity, with a constant factor containing Q_L and Q_r giving the insertion loss.

4. *Circuit Factor Compensation*

Section III,D gave the prescription for designing a transducer so that the array factor conforms to a prescribed function. In practice, we are usually given a desired transfer function $H(f)$ and want to arrange $T_{13}(f) \propto H(f)$. The formula for T_{13} given by Eq. (93) enables us to do so by eliminating regeneration distortion and reducing the problem back to array factor design.

The procedure calls for estimating $C(f)$ a priori, which is usually done by assuming $Y_{in} = j2\pi f C_T$. We then calculate a new function

$$G(f) \equiv H(f)C(f) \equiv \text{F.T.}[g(t)] \tag{94}$$

and use the array factor design procedure of Section III,D to design the x_n and w_n such that

$$A(f) = G(f) \equiv \text{F.T.}\{g(t) \equiv e(t)\exp[j\phi(t)]\}. \tag{95}$$

This circuit factor compensation procedure is an application of "predistortion"—we calculate a new "distorted" transfer function $G(f)$, so that when the array factor is designed to produce $G(f)$, the circuit factor will cause $T_{13}(f)$ to equal the desired function $H(f)$. The procedure is easily applied only to apodized dispersive transducers whose electrode position x_n and apertures w_n are not otherwise constrained.

VI. Matching-Circuit-Controlled Trade-offs among Insertion Loss, Bandwidth, Triple-Transit Suppression, and Voltage Standing-Wave Ratio

In this section it is assumed that the transducer geometry has already been designed so that the array factor $A(f)$ has the required frequency dependence. The next step in the filter design is the choice of the piezoelectric substrate material that is best suited to the temperature-stability and bulk wave suppression requirements and is also found to be capable of a satisfactory insertion loss for the required bandwidth according to Section VI, A. In view of these assumptions it follows from Eq. (91) that the

parameter Q_r is determined and fixed. Thus, the available trade-offs among insertion loss, bandwidth, triple-transit suppression, and voltage standing-wave ratio (VSWR) are controlled by choosing the form and element values of the matching circuit and by using the transducer aperture to adjust the parameter Q_L.

The insertion-loss/bandwidth trade-offs of Section VI, A and the reflection coefficient of Section VI, B are valid regardless of the choice of matching circuit. Sections VI, D and E give some specific, quantitative trade-offs for simple, commonly used matching circuits.

A. Fundamental Limitations Imposed on Filter Insertion Loss by the Required Bandwidth

We have seen that SAW transducers are basically transversal filters and the transducer bandwidth is primarily that of the array factor, which is strictly a function of transducer geometry. However, the transducer is coupled to an electrical source (or load), and we found in Section V, A that the transducer input admittance (Y_{in}) contains an electrostatic transducer capacitance (C_T in Fig. 25) that is often the dominant term in Y_{in}.

The problem of coupling between the electric source and the transducer is thus one of matching to a circuit containing susceptance. It has been established from the work of Bode (1945) and Fano (1950) that the gain-bandwidth product of such a network is limited by the quality factor (Q) of the RC load, which in our case is the transducer. Their results are expressed in terms of the reflection coefficient (Γ) from the load (transducer) by the integral equation

$$\int_0^\infty \ln(|\Gamma(f)|^{-1})\, df = \frac{\pi f_0}{Q}, \tag{96}$$

where the load has been made resonant at frequency f_0. An "ideal" matching network would be one that couples a constant fraction of the available power in the desired passband while reflecting all power [$\Gamma(f) = 1$] at all other frequencies. If the width of the passband is Δf, the one-transducer mismatch loss using this network would be

$$(L_{13})_{\text{ideal network}} = -10 \log_{10}\left\{\frac{1}{2}\left[1 - \exp\left(-\frac{2\pi}{Q\Delta f/f_0}\right)\right]\right\} \tag{97}$$

where the factor of $\frac{1}{2}$ accounts for transducer bidirectionality.

This result shows that the minimum achievable loss increases as a function of the required bandwidth Δf and the resonant Q of the transducer. Practical tuning networks can enable L_{13} to be remarkably close to

the ideal value given by Eq. (97). Examples of such networks are the shunt- or series-tuned network with padding (cf. Section VI, E) and the quarter-wave impedance inverter network (Reeder *et al.*, 1972; Reeder and Sperry, 1972). If no external resistive pad is used, the resonant Q in Eq. (96) may be approximated by the radiation Q (Q_r) defined in Eq. (90). In making this approximation we note that Eq. (96) strictly applies to an RC load whose resistance R is constant whereas the transducer radiation conductance G_a is frequency dependent.

Hartmann *et al.* (1973) have used the foregoing to determine the minimum (two-transducer) filter loss as a function of required bandwidth (Δf) for commonly used substrate materials. Their results make use of the fact that Q_r is dependent on the bandwidth $\Delta f/f_0$ and the substrate electromechanical coupling constant k^2 via Eq. (91). For any real matching networks in a two-transducer filter, Eq. (97) then implies that the insertion loss satisfies

$$2L_{13} \geq -20 \log_{10}\left\{\frac{1}{2}\left[1 - \exp\left(\frac{-8k^2}{\{\Delta f/f_0\}^2}\right)\right]\right\}. \tag{98}$$

Therefore, given a material and its k^2, we see that a filter can be operated near the 6-dB inherent bidirectional loss as long as the required bandwidth satisfies

$$\Delta f/f_0 \lesssim \sqrt{k^2}\ . \tag{99}$$

At the opposite extreme, if

$$\Delta f/f_0 \gg \sqrt{k^2}\ , \tag{100}$$

the achievable loss is seen from Eq. (98) to increase by 12 dB for each

TABLE II
MAXIMUM BANDWIDTH FOR 6-DECIBEL FILTER INSERTION LOSS ON VARIOUS SUBSTRATE MATERIALS

Material and Cut	k^2	$(\Delta f/f_0)_{\max,6\,\mathrm{dB}} \cong \sqrt{k^2}$
YZ $LiNbO_3$	0.044	0.21
YX Quartz	0.0022	0.05
ST-X Quartz	0.0015	0.04
YZ $LiTaO_3$	0.0067	0.08
"M.D.C." $LiTaO_3$ [a]	0.0108	0.10

[a] Slobodnik *et al.* (1975).

octave increase in $\Delta f/f_0$. In practice, this increase would be implemented by addition of padding resistance (as in Section VI, E) to lower the resonant electric Q. Without this padding resistance, the resonant electrical bandwidth would be smaller than the required filter bandwidth. With the padding resistance added, some of the available power is dissipated in the pad resistor, but the resonant electrical bandwidth is somewhat broader than the required filter bandwidth so that the filter bandshape is close to that of the transducer array factors.

It is convenient to use Eq. (98) to define the maximum bandwidth $(f/f_0)_{\text{max, 6 dB}}$ for which a 6-dB filter loss is attainable. This is given in Table II for several commonly used substrate materials. For larger bandwidths with a given material, the loss is

$$\text{IL} \cong 6\ \text{dB} + 10 \log_{10}\left[\frac{(\Delta f/f_0)}{(\Delta f/f_0)_{\text{max, 6 dB}}}\right].$$

This result is given graphically by Hartmann *et al.* (1973), Fig. 13.

B. Acoustic Reflection Coefficient T_{11}

Figure 43 shows the network representation appropriate for a transducer with a SAW incident at port 1, with the receiver load resistance being R_L and, as before, with an arbitrary network characterized by its *ABCD* matrix via Eq. (81). We know by reciprocity that the transfer function T_{31} is equal to the transfer function T_{13} found in Section V. The other transfer function of interest here is the acoustic reflection coefficient T_{11}. Before giving the result, we briefly outline how it is derived. We first

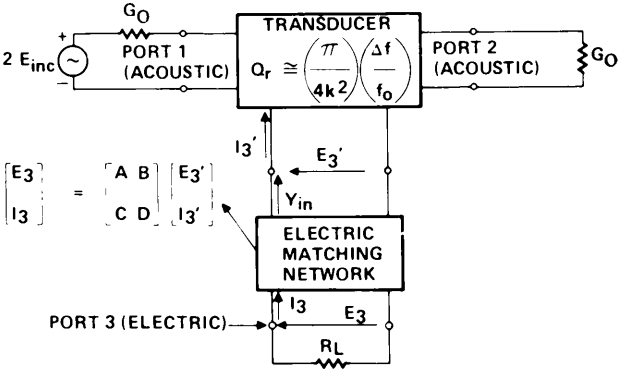


FIG. 43. Network representation for finding the acoustic reflection coefficient of a transducer receiving acoustic waves. (After Smith, 1976a.)

find T_{31} by applying a generator (acoustic) at port 1 and then treat the resultant voltage E_3 as a source that relaunches acoustic waves from ports 1 and 2. The relaunching process is treated in essentially the same way as the calculation of T_{13} (cf. Section V, E, 3). In finding the "regenerated" wave that emanates from port 1, it is *not* necessary to superimpose a contribution corresponding to the generator present at port 1 and a short circuit ($E_{3'} = 0$) at terminals 3'. This is because it is known from Section V that the crossed field and generalized circuit models predict *zero* acoustic reflection when port 3' is short-circuited and mass/electrical loading is negligible. When the above procedure is carried out it is found that

$$T_{11}(f) = T_{22}(f) = (Q_L/Q_r)[A(f)]^2/H(f), \tag{101}$$

where Q_L, Q_r, and $A(f)$ have the same meanings as in Section V, and $H(f)$ is another circuit factor, given by

$$H(f) = R_L[Y_{in} + (A + CR_L)/(B + DR_L)]. \tag{102}$$

Although Eq. (101) gives the reflection coefficient for all frequencies, the usual concern is to minimize the reflection coefficient over the passband of a SAW filter. The reflection coefficient is usually largest at frequencies where the magnitude of the electroacoustic transfer function is large, since both T_{11} and T_{13} contain $A(f)$ in the numerator. Usually, then, we are interested in the quantity $|T_{11}(f_0)|^2$ in order to establish trade-offs between insertion loss and triple-transit suppression in the passband of a SAW filter.

The insertion loss in a filter with two identical transducers is

$$\mathrm{IL}(f_0) = 2L_{13}(f_0) = -20\log_{10}|T_{13}(f_0)|^2, \tag{103}$$

while the triple transit suppression is

$$\mathrm{TTS}(f_0) = 2L_{11}(f_0) = -20\log_{10}|T_{11}(f_0)|^2. \tag{104}$$

For simplicity, in Section VI we shall assume a filter with two transducers that are identical at least with respect to their values of Q_L and Q_r. When this is not the case it is of course necessary to add the values of $L_{13}(f_0)$ and $L_{11}(f_0)$ for the two separate transducers.

C. Circuit Factors for Simple Matching Circuits

Having found the circuit factors for the electroacoustic transfer function T_{13} and the reflection coefficient T_{11}, we examine their form for some simple and commonly used matching circuits (Smith, 1976b). Table III gives the circuit factors for the following matching circuits: (1) direct connection, (2) shunt elements (typically a tuning coil and/or pad resistor), and (3) series element (also typically a tuning coil and/or pad resistor).

TABLE III
CIRCUIT FACTORS FOR TYPICAL MATCHING CIRCUITS

Matching Circuit	*ABCD* Matrix	Circuit Factors $C(f)$, $H(f)$
Direct connection (wires only)	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$C(f) = H(f) = 1 + R_L Y_{in}$
Shunt Element Y_{sh} e.g., coil and/or pad resistor	$\begin{pmatrix} 1 & 0 \\ Y_{sh} & 1 \end{pmatrix}$	$C(f) = H(f) = 1 + R_L(Y_{in} + Y_{sh})$
Series Element Z_{ser} e.g., coil and/or pad resistor	$\begin{pmatrix} 1 & Z_{ser} \\ 0 & 1 \end{pmatrix}$	$C(f) = 1 + Y_{in}(R_L + Z_{ser})$ $H(f) = R_L\left(Y_{in} + \frac{1}{Z_{ser} + R_L}\right)$
Any network other than pure shunt elements	$\begin{pmatrix} A & B \neq 0 \\ C & D \end{pmatrix}$	$\lvert H(f)\rvert$ and $\lvert C(f)\rvert$ may have widely dissimilar form and dependence on R_L

Note that when the matching circuit is a direct connection or consists of shunt elements only, the circuit factor $C(f)$ and $H(f)$ are identical. For other types of circuits, they may be widely dissimilar, so that insertion loss/triple-transit trade-offs can be quite different functions of the load resistance for different types of matching circuits.

D. THE DIRECT-CONNECTION OR "UNTUNED" TRANSDUCER (MATCHING CIRCUIT "ABSENT")

When the matching circuit is a direct connection with the *ABCD* matrix the unit matrix, the only available trade-off is effected by adjusting Q_L by choice of the transducer aperture or use of a transformer to vary the load resistance. The effect is to trade off insertion loss and triple-transit suppression. The insertion loss is

$$\mathrm{IL}(f_0) = -20 \log_{10}\left[\frac{2Q_L Q_r}{(Q_L + Q_r)^2 + Q_L^2 Q_r^2}\right] \tag{105}$$

and the triple-transit suppression is

$$\mathrm{TTS}(f_0) = -20 \log_{10}\left[\frac{Q_L^2}{(Q_L + Q_r)^2 + Q_L^2 Q_r^2}\right]. \tag{106}$$

The triple transit suppression is greatest for the smallest possible value of Q_L. The insertion loss is minimized by choosing

$$Q_L^2 = Q_r^2/(1 + Q_r^2) \quad \cong 1 \quad \text{for} \quad Q_r^2 \gg 1. \tag{107}$$

When Q_L is chosen smaller than this value, the triple-transit suppression is improved at a penalty in insertion loss. When Q_L is chosen larger than this value, the insertion loss increases while the triple transit suppression is also degraded. We conclude that Q_L should never be chosen larger than the value given by Eq. (107), since it would always be possible to find a different value of Q_L that is less than that given by Eq. (107) and yields a much better triple-transit suppression for the same insertion loss.

Often high triple-transit suppression is of primary importance to the extent that a device specification calls for a minimum acceptable triple-transit suppression, with the insertion loss to be minimized subject to this constraint. Typically, this situation requires $Q_L \ll 1$ ("weak coupling") to the extent that Q_L may be considered negligibly different from zero in the denominator of Eqs. (105) and (106). In this limit we obtain the following simple trade-off:

$$\text{TTS}(f_0) = 2\ \text{IL}(f_0) + 12\ \text{dB}. \tag{108}$$

Over most of the useful range for Q_L, the triple-transit suppression increases by approximately 2 dB for a 1-dB increase in insertion loss.

E. Shunt- and Series-Tuned Transducers

For the simple, commonly used tuning circuits that use a single inductor (in shunt or series) to resonate the transducer capacitance, it is useful to pursue in some detail the trade-offs among bandwidth, insertion loss, triple-transit suppression, and VSWR (Smith, 1976b). The behavior of the shunt- and series-tuned circuits is dual except with respect to triple-transit suppression, as will be shown explicitly in the following.

1. *Shunt-Tuned Transducers*

Figure 44a is a circuit diagram for a shunt-tuned transducer (the shunt coil L resonates the transducer capacitance C_T at the synchronous frequency f_0). Optionally, we add a pad resistor of conductance G_p. Conditions for using a pad and criteria for choosing its value are given in Section VI, E, 3, C. Also, an optional transformer is allowed in order to supplant the transducer aperture as a variable for adjusting Q_L, since the resistance (G_s^{-1}) of the source or load may be constrained to a standard value such as 50 Ω.

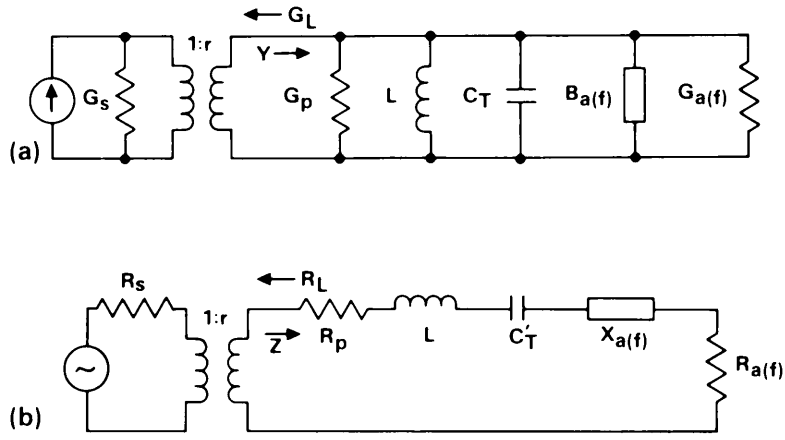


FIG. 44. Shunt-tuned (a) and series-tuned (b) transducer matching circuits with optional impedance transformers and pad resistors.

The introduction of the pad resistor requires another independent Q parameter, which we define as

$$q_p = G_p/G_L \qquad \text{(shunt-tuned transducer)}. \tag{109}$$

For finding insertion loss and triple transit suppression we need in addition to q_p only the ratio

$$q_a = G_a(f_0)/G_L = Q_L/Q_r \qquad \text{(shunt-tuned transducer)}, \tag{110}$$

which is proportional to the aperture. In terms of these parameters, the insertion loss is

$$\mathrm{IL}(f_0) = -20 \log_{10}\left[\frac{2q_a}{(1+q_a+q_p)^2}\right] \tag{111}$$

and the triple-transit suppression is

$$\mathrm{TTS}(f_0) = -20 \log_{10}\left[\frac{q_a^2}{(1+q_a+q_p)^2}\right]. \tag{112}$$

2. *Series-Tuned Transducers*

The circuit diagram for a series-tuned transducer is shown in Fig. 44b. The shunt elements C_T, B_a, and G_a of the transducer admittance have been converted to the equivalent series elements C'_T, X_a, and R_a. Optionally, a series pad resistor R_p is used. This circuit is dual to that shown in Fig. 44a.

Equation (111) for the insertion loss remains valid provided that we redefine q_p by

$$q_p = R_p/R_L \qquad \text{(series-tuned transducer)} \tag{113}$$

and q_a by

$$q_a = R_a(f_0)/R_L = (Q_r/Q_L)(1+Q_r^2)^{-1} \qquad \text{(series-tuned transducer)}. \tag{114}$$

The parameter q_a is now inversely proportional to the aperture instead of being directly proportional to aperture as was the case for the shunt-tuned transducer.

The behavior of the series-tuned transducer is *not completely dual* to that of the shunt-tuned transducer for the sole reason that the triple-transit suppression is *not* given by Eq. (112), even with the new definitions of q_a and q_p. Instead, the triple-transit suppression is given by the more complicated expression

$$\mathrm{TTS}(f_0) = -20\log_{10}\left\{\frac{1}{(1+Q_r^2)^2}\left[Q_r^2 + \left(1 - \frac{1+Q_r^2}{1+\rho}\right)^2\right]\right\}, \tag{115}$$

where

$$\rho = (R_p + R_L)/R_a(f_0) \equiv (R_p + R_L)G_a(f_0)(1+Q_r^2). \tag{116}$$

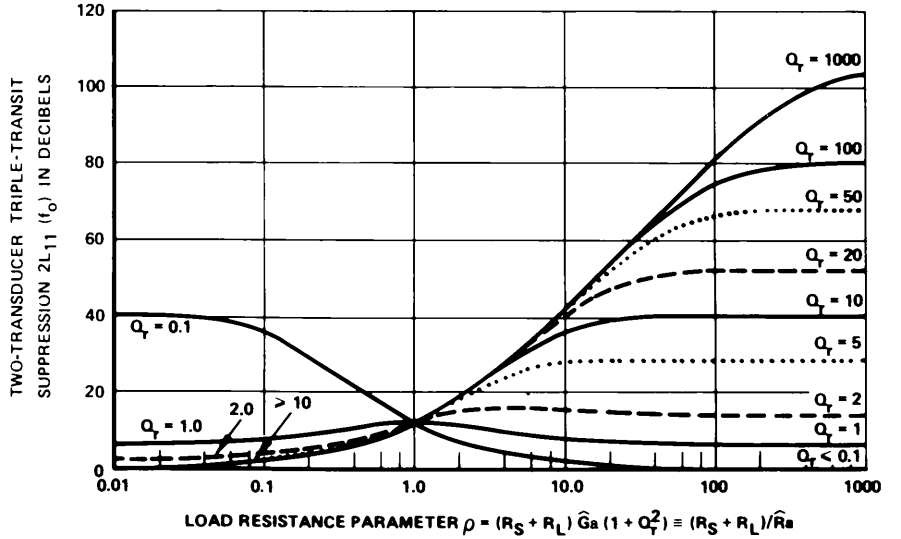


FIG. 45. The triple transit suppression of a pair of acoustically synchronous series tuned transducers as a function of the load resistance with the radiation Q as a parameter. (After Smith, 1976b.)

Figure 45 is a plot of the triple-transit suppression of two series-tuned transducers, with ρ as the independent variable and Q_r as a parameter. In the limit $Q_r \to \infty$ (i.e., a low substrate k^2 and broad fractional bandwidth), Eq. (115) does in fact reduce to Eq. (112) provided we use the series-circuit definitions (113) and (114) for q_p and q_a. In that limit, the behavior of the series- and shunt-tuned circuits is completely dual, including the triple-transit suppression.

For an arbitrary value of Q_r, the maximum attainable value of the triple-transit suppression is

$$\max\{\mathrm{TTS}(f_0)\} = -20\log_{10}\left[Q_r^2/(1+Q_r^2)\right]^2, \tag{117}$$

and this value is obtained with the load resistance parameter ρ set at $\rho = Q_r^2$. Therefore, for $Q_r > 1$, the triple-transit suppression tends to "saturate" and actually decrease slightly if ρ is increased beyond the value $\rho = Q_r^2$. For $Q_r < 1$ (an unusual situation with many electrodes on a high-k^2 substrate) the triple-transit suppression decreases, rather than increasing, as ρ increases. In fact, the triple-transit suppression for any pair of values (ρ, Q_r) is the same as the suppression for the "inverse" pair of values (ρ^{-1}, Q_r^{-1}).

3. *Simultaneous Trade-offs of Bandwidth, Insertion Loss, Triple-Transit Suppression, and VSWR*

a. *Bandwidth parameter for the circuit factor C(f).* In order to design the transducer for operation over a required bandwidth, it is necessary to recognize that the overall transducer response is the composite response of the array and circuit factors as specified by Eq. (93). Figure 46 is a

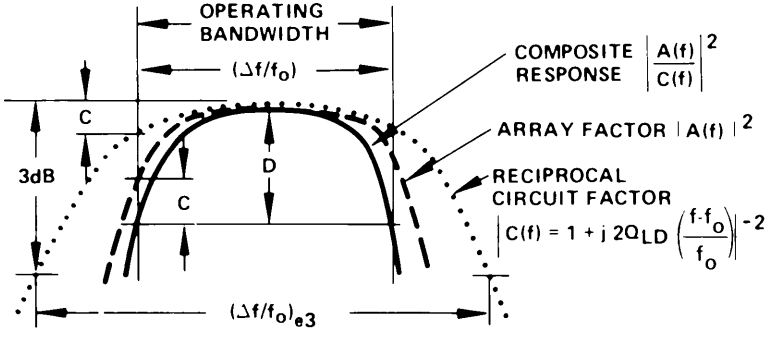


FIG. 46. An illustration that the bandwidth of a tuned transducer depends on both the array and circuit factors. With the transducer geometry and array factor fixed and a required operating bandwidth specified, one can determine the maximum allowable resonant electrical Q. (After Smith, 1976b.)

schematic diagram showing the composite transducer response (solid line), which is proportional to $|A(f)|^2$ (array-factor contribution—dashed line) and also proportional to $|C(f)|^{-2}$ (circuit factor contribution—dotted line). We assume that the circuit factor takes the form

$$C(f) \cong 1 + j2Q_{LD}[(f - f_0)/f_0], \tag{118}$$

which is a good approximation for a simple shunt- or series-tuned circuit when the loaded Q (Q_{LD}) exceeds 3 or 4.

Suppose that it is required to operate the transducer (composite response) over a fractional bandwidth $\Delta f/f_0$ with a roll-off of D decibels at the band edge (Fig. 46). First choose a geometrical transducer design so that the array factor has a somewhat broader bandwidth than the required final operating bandwidth. Calculate the array factor (dashed curve) and note that it rolls off by less than D decibels at the band edge. The difference of c decibels is therefore the allowed roll-off in the circuit factor at the bandwidth $\Delta f/f_0$. Denote the 3-dB bandwidth of the circuit factor by $(\Delta f/f_0)_{e3}$ and denote by β the ratio

$$\beta = \left(\frac{\Delta f}{f_0}\right)_{e3} \Big/ \left(\frac{\Delta f}{f_0}\right). \tag{119}$$

With the circuit factor restricted to the form (118), we see that the value of β depends only on the value of c. Figure 47 is a plot of β as a

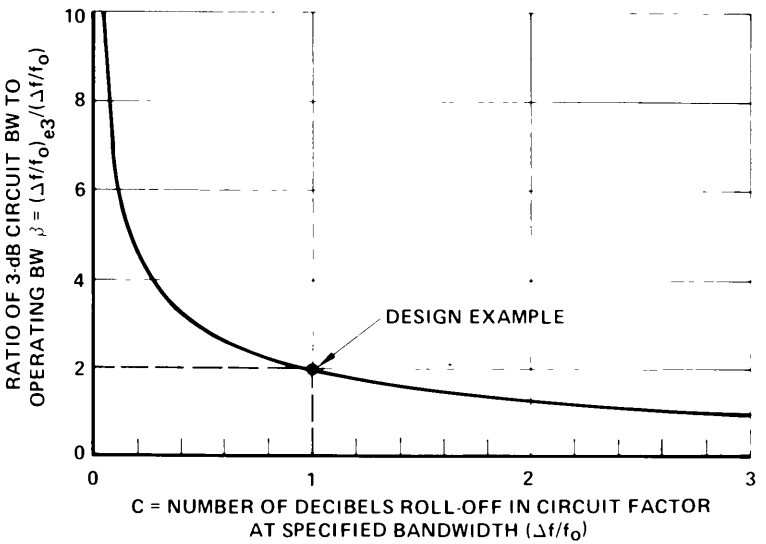


FIG. 47. Required 3-dB bandwidth of the resonant tuning circuit as a function of its allowed roll-off at the edge of the desired transducer operating band. (After Smith, 1976b.)

function of c, obtained from Eqs. (118) and (119). The parameter β is a measure of the required circuit factor bandwidth at the 3-dB points and hence of the required loaded Q. The smaller the allowed circuit factor roll off c, the larger is the parameter β and the required 3-dB bandwidth $(\Delta f/f_0)_{e3}$ and the lower is the required loaded Q. We shall see below how the parameter β affects the achievable insertion loss for the required bandwidth.

b. *Expressions for the loaded Q and VSWR.* The quantity controlling the bandwidth of the circuit factor $C(f)$ is the "loaded Q" (Q_{LD}). For the shunt tuned circuit of Fig. 44a, it is the ratio of the capacitive susceptance, $2\pi f_0 C_T$, to the total conductance, $G_L + G_P + G_a$. For the series circuit of Fig. 44b, it is the ratio of the capacitive reactance, $(2\pi f_0 C_T')^{-1}$ to the total resistance, $R_L + R_p + R_a$. In either case, if we select the appropriate definitions for q_a and q_p, we may write

$$Q_{LD} = q_a Q_r / (1 + q_a + q_p). \tag{120}$$

At the synchronous frequency f_0, where the tuning inductor and transducer capacitance are made resonant, the input admittance (Y in Fig. 44a) or impedance (Z in Fig. 44b) is real. At this frequency, the VSWR is

$$r_0 = (q_a + q_p)^{\mp 1}, \tag{121}$$

where the sign is taken to give $r_0 \geq 1$. The VSWR over the full operating band involves the susceptance or reactance of the coil, capacitance, and acoustic radiation. The value at the band edge depends on the bandwidth and the loaded Q, as will be seen in an example below.

c. *Conditions for requiring a resistive pad.* There are three reasons that a pad resistor (G_p or R_p) might be required in the circuits of Fig. 44. The first is to ensure that the loaded Q is low enough for the required tuned electrical bandwidth. This is expressed by

$$1/Q_{LD} \geq (\Delta f/f_0)_{e3} = \beta(\Delta f/f_0), \tag{122}$$

where β is defined by Eq. (119) and plotted in Fig. 47. Using Eq. (120), this condition becomes

$$q_p \geq q_a(x - 1) - 1, \tag{123}$$

where

$$x = \beta Q_r(\Delta f/f_0) \tag{124}$$

will be called the *bandwidth parameter*. Note that x increases as the required bandwidth $(\Delta f/f_0)$, as the radiation Q, $Q_r \propto 1/k^2$, and as the circuit factor parameter β. An increase in any of these three quantities necessitates more padding and insertion loss to maintain the bandwidth.

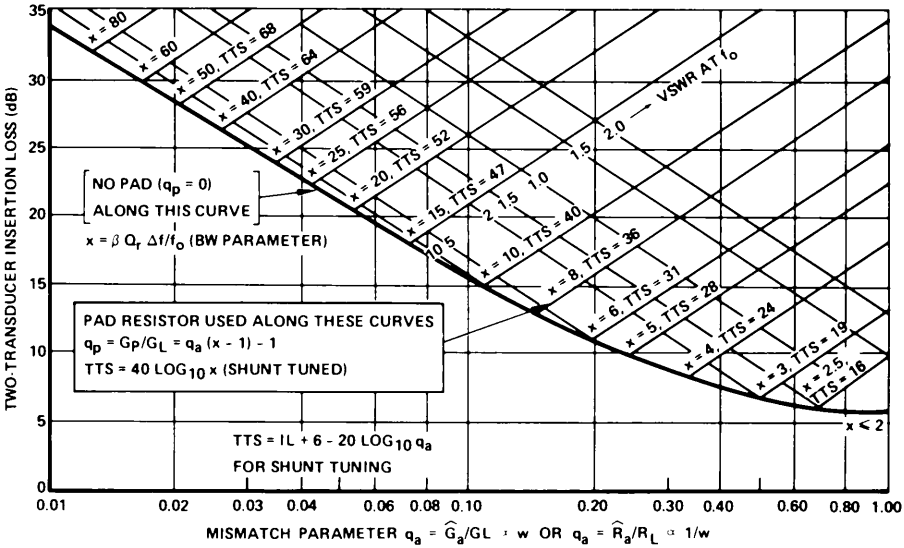


FIG. 48. A graphical representation for insertion loss/bandwidth/triple-transit suppression/VSWR trade-offs of shunt- and series-tuned transducers. TTS data are not applicable to series-tuned transducers unless $Q_r \gg 1$ and $Q_r^2 \gg \rho$. (After Smith, 1976b.)

The second reason that a pad might be required is to increase the triple-transit suppression given by Eq. (112) or Eq. (115), in the event that the more efficient (lower IL) procedure of lowering q_a implies an unreasonable aperture. The third reason for requiring a pad is to lower the VSWR. If only the VSWR at the center frequency is of concern, Eq. (121) determines the required pad. If a specified VSWR must be maintained over the full operating bandwidth ($\Delta f/f_0$), more padding is required and is determined by a graphical construction below.

d. *Graphical representation for insertion loss/bandwidth/triple-transit/VSWR tradeoffs.* The use of Eqs. (109)–(124) in transducer design is facilitated by the graphical representation of Fig. 48. According to Eq. (123), if $x \leqslant 2$, no pad resistor is required to maintain the required bandwidth. Correspondingly, the lowest, heavy curve in Fig. 48 is a plot of Eq. (111) (in decibels for *two* transducers) for the case $q_p = 0$ (no pad). The lowest possible insertion loss is 6 dB where $q_a = 1$. For all values of q_a on this curve, the VSWR at f_0 is given by q_a^{-1}. For shunt-tuned transducers the (two-transducer) triple-transit suppression is related to q_a, and the insertion loss by

$$\text{TTS} = \text{IL} - 20 \log_{10} q_a + 6 \text{ dB}. \tag{125}$$

This expression is also a good approximation for series-tuned transducers provided $Q_r \gg 1$ and $Q_r^2 \gg \rho$, with ρ given by Eq. (116).

For larger values of the bandwidth parameter ($x > 2$), a pad is required to maintain bandwidth when q_a is large enough that the right-hand side of Eq. (123) is greater than zero. Accordingly, the minimum insertion loss consistent with the required bandwidth is given as a function of q_a by the heavy curve down to its intersection with the line labeled with the appropriate value of x, and then up that line as a pad is added. Along that line, the value of the pad is given by Eq. (123) with the equality sign. For shunt-tuned transducers the triple-transit suppression, while still given by Eq. (112), also takes the simpler form TTS = const = 40 $\log_{10} x$. Minimum insertion loss is obtained with $q_p = 0$ and $q_a = 1/(x-1)$, which should be chosen unless $q_p > 0$ is required to lower the VSWR, or unless $q_p > 0$ and/or $q_a < 1/(x-1)$ is required to achieve the required TTS.

Lines of constant VSWR at f_0, from Eq. (121), are also plotted in Fig. 48. When an upper limit is specified for the VSWR at f_0 so as to require a pad, q_a and q_p should be chosen by operating at the intersection of the lower of the two lines having the specified VSWR value with the line having x given by Eq. (124) or by TTS = 40 $\log_{10} x$ (whichever is larger). The method for finding the additional padding required to maintain the VSWR over the *full* operating bandwidth ($\Delta f/f_0$) will be illustrated in an example below.

e. *Design examples.* For all the examples to be considered here, we shall assume that it is required to design a delay line with a center frequency of 100 MHz and a 3-dB operating bandwidth of 20 MHz so that $\Delta f/f_0 = 0.2$ and D (Fig. 46) is 1.5 dB for each of the two transducers. We shall seek the minimum insertion loss consistent with various combinations of triple-transit and VSWR restrictions.

We assume the same geometrical transducer design throughout: a pair of identical, unapodized, 15-double electrode phase reversal transducers (PRT's) (Bristol, 1972) on YZ lithium niobate having the polarity sequence (+ − − + − + − + − + − + − − +). It is found from Eq. (12) that the 3-dB bandwidth of the array factor is 30 MHz, so that Eq. (91) yields $Q_r = 6.53$ assuming an effective k^2 for double electrodes of $0.82 \times 0.044 = 0.036$. The array factor rolls off by 0.5 dB at the operating band edges of 90 and 110 MHz, so, for each transducer, the quantity c in Fig. 46 is 1 dB. From Fig. 47, the ratio β is 2.0, so that the bandwidth parameter x is 2.6 from Eq. (124).

Example A. What is a design for achieving minimum insertion loss with the specified bandwidth without restriction on the VSWR or triple-transit suppression? Solution: From Eq. (123) the bandwidth can be maintained without a pad resistor for $q_a < 1/(x-1) = 0.62$, or with a pad for larger q_a, with q_p given by Eq. (123). In Fig. 49 the designs that have adequate bandwidth lie on or above the V-shaped curve formed by the curve NO PAD down to its intersection with the line $x = 2.6$ and then up

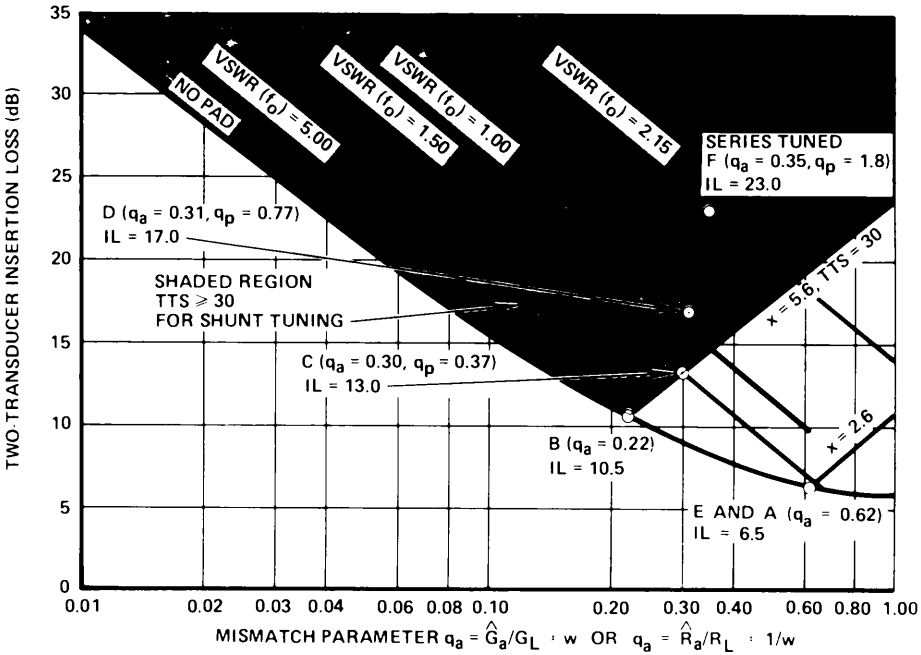


FIG. 49. Tradeoff examples illustrating the use of the graphical representation of Fig. 48. (After Smith, 1976b.)

that line. The minimum loss of 6.5 dB is obtained for the parameters of point *A* ($q_a = 0.62, q_p = 0$). In a 50-Ω system, $q_a = 0.62$ implies $G_a = 12.4$ $m\Omega^{-1}$. Since apertures larger than about 100 wavelengths are undesirable from the standpoint of substrate area and electrode resistivity, we choose a 100-wavelength aperture so that $G_a = 1.24$ $m\Omega^{-1}$. An impedance transformer (Fig. 44) that transforms the 50-Ω source to $G_L = 2$ $m\Omega^{-1}$ is needed to achieve $q_a = 0.62$. The VSWR is 1.6 at 100 MHz, and the triple-transit suppression is about 16 dB.

Example B. What insertion loss must be tolerated if we add the restriction of at least 30-dB triple-transit suppression? Solution: Since Eq. (112) is always satisfied for a shunt-tuned transducer, the triple-transit suppression is greater than 30 dB in the shaded region of Fig. 49 and is exactly 30 dB along the line marked $x = 5.6$, TTS = 30. Since there is no specification given on VSWR, minimum insertion loss is obtained at point *B* in Fig. 49, where $q_a = 0.22$ and there is no pad resistor ($q_p = 0$). The insertion loss is 10.5 dB and the VSWR at 100 MHz is about 5. If the transducer aperture is held to 100 wavelengths, a 3 : 1 impedance transformer is still required to transform the 50-Ω source to $G_L \cong 5.6$ $m\Omega^{-1}$ so

that $q_a = 0.22$. The required bandwidth is maintained since the operating point B lies well *above* the line $x = 2.6$.

Example C. Retain the same bandwidth and 30-dB triple-transit suppression requirement but add the restriction that the VSWR at 100 MHz must be 1.5. What is the minimum insertion loss? Solution: add a shunt pad resistor (Fig. 44a), which in Fig. 49 yields point C, at the intersection of the lines $x = 5.6$, $\mathrm{TTS} = 30$ and $\mathrm{VSWR}(f_0) = 1.5$. The resultant parameters $q_a = 0.30$, $q_p = 0.37$ yield about 13.2-dB insertion loss.

Example D. Retain the same bandwidth and triple-transit requirements but require that a VSWR of 1.5 or less be maintained over the full 20-MHz operating bandwidth. Again determine the optimum q_a and q_p for minimum insertion loss. Solution: The VSWR tends to increase at the band edge because there is acoustic susceptance (B_a) and L–C susceptance. The latter dominates, and we ignore the acoustic susceptance. In the Smith chart plot of Fig. 50 the largest normalized susceptance consistent with a VSWR of 1.5 is $b_{max} \equiv B_{max}/G_L = 0.42$, since the curve $b = 0.42$ is just tangent to the circle $r = 1.5$. Following the arrow in to the real axis, we see that a total normalized conductance $q_a + q_p = 1.08$ is required. Subject to this condition we want q_a as large as possible and q_p as small as possible to minimize insertion loss. However, the normalized admittance

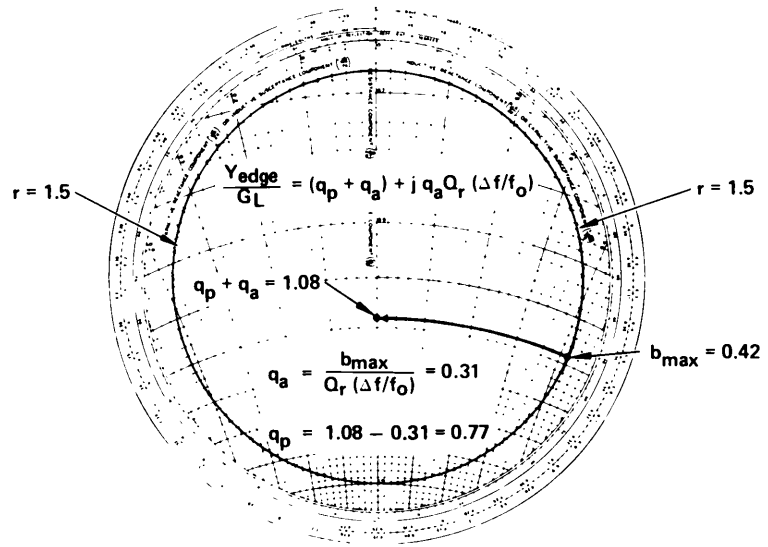


FIG. 50. Smith-chart construction used in conjunction with Fig. 48 when it is specified that the VSWR not exceed an assigned value over the full operating bandwidth. (After Smith, 1976b.)

Y/G_L (Fig. 44a) is

$$\frac{Y}{G_L} = \frac{G_p + G_a + j(\omega C - 1/\omega L)}{G_L}$$

$$\cong q_a + q_p + j2q_aQ_r\left(\frac{f - f_0}{f_0}\right). \tag{126}$$

Thus the condition $b_{max} = 0.42$ at $(f - f_0)/f_0 = \frac{1}{2}(\Delta f/f_0)$ requires $q_a \leq b_{max}/[Q_r(\Delta f/f_0)] = 0.31$, so we take $q_a = 0.31$ and $q_p = 1.08 - 0.31 = 0.77$. The result is point D in Fig. 49, giving an insertion loss of 17 dB and a triple-transit suppression of 33 dB. Note that we have assumed here that G_a is just as large at the band edges (90 and 110 MHz) as it is at $f_0 = 100$ MHz. In reality, G_a is slightly smaller at the band edge since the array factor rolls off by 0.5 dB there. On the other hand, the acoustic susceptance B_a is opposite in sign to the L–C susceptance and thus slightly lowers the total susceptance at the band edge. Thus, the VSWR at the band edge will be about 1.5 while at f_0 the circuit is actually slightly overcoupled and the VSWR is $q_a + q_p = 1.08$.

Example E. Revert to the requirements of Example A, with no VSWR or TTS requirements, but assume a series-tuned transducer as in Fig. 44b. The point A in Fig. 49 again satisfies the requirements, yielding $q_a = 0.62$, $q_p = 0$, IL = 6.5 dB, and VSWR$(f_0) = 1.6$. Note that now q_a is defined by $q_a = R_a/R_L$ (see Fig. 44b). The value $q_a = 0.62$ implies $R_L = 50$ Ω and $R_a = 31$ Ω, the latter being easily achieved with an aperture of 56 wavelengths and without an impedance transformer.

Example F. Retain the series-tuned circuit and add the requirement TTS ⩾ 30 dB. What are optimum values of q_a and q_p and the minimum insertion loss? Solution: The triple-transit results shown in Figs. 48 and 49 do not apply, and we must refer to Fig. 45. With $Q_r = 6.5$, the largest triple-transit suppression obtainable for any value of the load parameter ρ is 33 dB. Achieving TTS = 30 dB requires $\rho = (R_p + R_L)/\hat{R}_a = 8$. With $R_L = 50$ Ω, $\rho = 8$ could be achieved with $R_p = 0$ and $\hat{R}_a = 6.25$ Ω, but this would require an unreasonably large aperture of 300 wavelengths or a transformer to achieve $\hat{R}_a = 6.25$ Ω with a 100-wavelength aperture. With such a transformer, the insertion loss would be 14 dB, and the VSWR at 100 MHz is 8. If, on the other hand, the aperture is restricted to 100 wavelengths and no transformer is used, then $\hat{R}_a = 17.5$ Ω ($q_a = 0.35$) and we need to add a pad resistor of 90 Ω ($q_p = 1.8$) so that $\rho = (R_p + R_L)/\hat{R}_a = 8$. The resultant insertion loss is 23 dB and the VSWR at 100 MHz is (107.5 Ω/50 Ω) = 2.15, as given by point F in Fig. 49. The triple-transit suppression given by Eqs. (115) and (116) and plotted in Fig. 45 has severely limited the available trade-offs in insertion loss and VSWR, a typical problem for series tuned transducers unless Q_r is very large.

F. Choice between Series and Shunt Tuning; Other Matching Circuits

Shunt tuning has two inherently attractive features:

(1) it has a simple form of the circuit factors $C(f)$ and $H(f)$;
(2) it typically allows a higher triple-transit suppression for a given insertion loss.

Shunt tuning is therefore recommended unless prohibited by the following circumstances.

With the shunt-tuned circuit, it is often difficult to attain values of the mismatch parameter q_a that approach unity. This is because the radiation conductance is given (e.g., for periodic transducers) by

$$G_a(f_0) = 8k^2 f_0 C_T N, \tag{127}$$

which tends to be small for design combinations involving the following: few electrodes N, low center frequency f_0, low k^2, and low dielectric constant. The latter two conditions are found with quartz substrates, and are usually not very effectively offset by the typically higher f_0 and N used in designs on quartz. Even with lithium niobate substrates, broadband transducers with few electrodes may still require an impedance transformer to allow q_a to approach unity for low insertion loss. It is for this reason that the series-tuned circuit shown in Fig. 44b is often used.

In addition to the single series- or shunt-tuning inductor, a variety of other matching circuits are sometimes used. For example, exact impedance matching at a single frequency can be achieved by use of an appropriate combination of an inductor and a capacitor, one of which is a shunt element and the other a series element, i.e., and "el" network. Also, distributed or lumped-element transmission line impedance inverters (Reeder *et al.*, 1972) are often useful. While presentation of trade-offs for these specific circuits is beyond the scope of this article, it should be emphasized that Eqs. (93) and (101) for T_{13} and T_{11} enable the derivation of insertion loss/triple-transit trade-offs for an arbitrary matching circuit by use of the $ABCD$ matrix.

VII. Specific Transducer Geometries (Array Factors) for Typical Filter Applications

A. Nondispersive Bandpass Filters

Any transducer having symmetry about a mirror plane normal to the propagation direction is nondispersive since the array factor is easily seen to be real valued. Most bandpass filters employ two transducers, each

having this symmetry and also having uniformly spaced electrodes as described in Section III, D, 1, a. Here we summarize some of the typical choices for the impulse response envelope $e(t)$ and hence the associated band shape $|A(f)|$. Given the envelope $e(t)$, one can design an apodized transducer as described in Section III, D or, if $e(t)$ does not have extensive regions with very small amplitude, one can instead design an unapodized source-withdrawal transducer as described in Section III, E. Thus, both transducers can be used in designing the filter transfer function.

1. *"Brick-Wall" Filters*

The majority of the synthesis algorithms are for the "brick-wall" type of band shape characterized by a passband with a flat central region, a stopband with low-level sidelobes, and a narrow transition region between the passband and the stopband. The basic problem is that the spatial extent (and hence the impulse response duration) of a transducer is necessarily finite, requiring trade-offs among the aforementioned characteristics. Tancrell (1974) gives a good tutorial review of the appropriate design techniques, which fall into two classes: the basic Fourier-transform principles of Section III, where the impulse response contains a window function to reduce ripples, and numerical optimization schemes that trade off the required characteristics. In general, the required number of electrodes increases as the allowable passband and stopband ripple amplitudes and transition bandwidth decrease. Therefore, practical limits are imposed by the available substrate length and the accuracy with which very small electrode "weights" can be implemented with narrow aperture.

In the first approach, the "desired" impulse response is calculated from the desired transfer function by Fourier transform. The impulse response is then weighted by multiplication by a window function whose length is finite; the envelope of the weighted impulse response is then used as $e(t)$ in the procedure outlined in Section III, D above. Particular window functions are described by Tancrell (1974) and Harris (1978). Also noteworthy is the special Fourier-transform technique given by Vasile (1974), where the transfer function is constructed by superposition of approximately triangular "frequency eigenfunctions" each of whose Fourier transform is a "combination cosine-squared" weighting function. This technique lacks full flexibility in trading off passband and stopband ripples against transition bandwidth, but it is particularly simple to use and has an inherently low stopband sidelobe level.

Optimization schemes usually begin with a set of criteria such as maximum allowable passband and stopband ripple amplitude and maximum transition bandwidth. Given a trial number of electrodes N, the computer algorithm performs iterations on the apertures w_n until the

criteria are met or approached as closely as possible. Of particular interest are schemes that yield equiripple in the passband and stopband. Among these are the interpolation schemes of Hofstetter *et al.* (1970, 1971) and of Parks and McClellan (1972). Recently, Kodama (1979) has described a scheme using nonlinear programming that is particularly effective when designing complex passband shapes whose specifications in some parts are more strict than in others.

2. *Other Band Shapes*

The "brick-wall" type of filter is not optimum for all applications. For example, a channelized filter bank might be required for frequency identification of signals ranging from cw down to short tone bursts. While the "brick-wall" filters provide maximum selectivity for cw signals, their response to short pulses is such that large undesired transients appear in channels whose frequency is far from that of the unknown carrier. These transients are reduced by use of filters with a smooth, rounded band shape. Typical examples are the Gaussian function and the "Hamming-squared" function given by

$$H^2(f) = \left[c_1 \frac{\sin x}{x} + c_2 \left(\frac{\sin(x+\pi)}{x+\pi} + \frac{\sin(x-\pi)}{x-\pi} \right) \right]^2, \qquad (128)$$

with $x = \pi(f - f_0)T$ and T the length of each of the two transducers. The latter case has also been shown (Smith, 1977) to provide a good approximation to the matched filter for a class of spread-spectrum communication signals known as minimum-shift-keyed (MSK) or continuous phase-shift modulated (CPSM) waveforms.

Both the Gaussian and Hamming-squared filters are easy to design because the desired impulse response is the same for both transducers, and its smooth shape without phase reversals is especially suitable for accurate approximation by unapodized source-withdrawal transducers.

3. *Filter Example*

A simple design for a filter of the "brick-wall" type uses an unapodized source-withdrawal transducer having a Hamming transfer function $H(f)$ from Eq. (128) and an apodized transducer designed by the method of Vasile (1974). The apodized transducer's transfer function is synthesized from three quasi-triangular, frequency-overlapping "eigenfunctions," and by assigning a smaller amplitude to the central one, the rounded band shape of the (other) Hamming transducer is compensated and the overall filter response is flat.

Using this design technique, a filter having a fractional bandwidth of about 0.04 was fabricated with double electrodes on an ST-X quartz

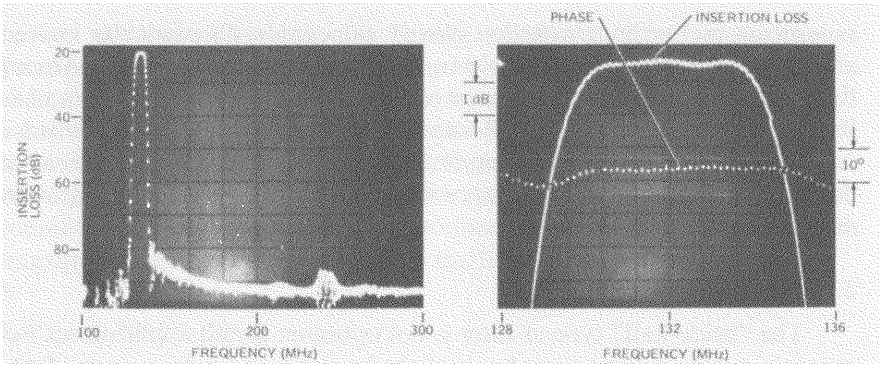


FIG. 51. Measured insertion loss and phase for a nondispersive bandpass filter with one apodized transducer and one unapodized "source-withdrawal" transducer.

substrate. Each transducer was tuned with a series inductor. The left-hand panel of Fig. 51 is a measured insertion loss plot showing the good stopband rejection, and the right-hand panel shows the small amplitude ripples and the small deviations from linear phase caused by the circuit factor and by triple-transit echoes. The ratio of the 50- to 3-dB bandwidths (shape factor) is 1.86. Lower shape factors require more sophisticated design techniques (usually an optimization scheme), but this example illustrates that other desirable properties (high stopband rejection and low ripple) can be obtained in a filter with a simple straightforward design.

B. Dispersive Filter Design

1. *Dispersive Filter Devices and Applications*

Radar pulse compression (Klauder *et al.*, 1960) was one of the earliest applications envisioned for SAW filters. One filter is impulse-excited to produce a transmitted waveform (typically linear-FM) whose bandwidth is larger than the reciprocal of the pulse duration. After reflection from a target, the return waveform is introduced into another SAW filter that is essentially the matched filter for the transmitted waveform except that additional weighting is applied to suppress the time sidelobes surrounding the principal narrow output pulse. Unless the filter phase transfer function (which may vary by thousands or even millions of degrees across the passband) is accurate to a few degrees, the undesired time sidelobes will rise rapidly and appear as spurious targets in the radar system. The same phase accuracy is required in dispersive filters for the microscan receiver (Gerard *et al.*, 1973) and the chirp-Z transform processor (Gerard *et al.*, 1977), which is a generalized analog cross-correlation processor.

The earliest SAW dispersive filters used the basic configuration described throughout this article—a pair of interdigital transducers oriented end-to-end on a single substrate. The simplest filter consists of a single linear-FM transducer (cf. Section III, D, 1,b) and a nondispersive periodic transducer with a few electrodes for launching the surface wave. On substrates with weak piezoelectric coupling such as quartz, the use of a broadband periodic transducer with a few electrodes (and a correspondingly high impedance) would result in high insertion loss. Therefore, the geometry shown in Fig. 5 is preferred for dispersive filters on quartz, and may also be used on strongly piezoelectric substrates. The design is slightly more complicated when both transducers are dispersive because the required filter transfer function must be the product of the two transducer transfer functions.

In the last few years, the reflective array compressor or RAC (Williamson and Smith, 1973; Williamson, 1976) has replaced dispersive interdigital transducers as the most attractive means of implementing dispersive filters whose time–bandwidth product is a few hundred or greater. There are three basic reasons for the superiority of the RAC for large time–bandwidth product devices. First, the filter phase transfer function, which must be accurate to within a few degrees, is less sensitive to fabrication imperfections in reflective groove arrays than in transducers with many electrodes. Second, the phase errors in reflective groove arrays can be further reduced with metallic phase correction films custom-tailored to the measured performance of each individual RAC filter (Williamson, 1976). Third, it is difficult to design and impedance-match interdigital transducers having extremely large numbers of electrodes without encountering severe regeneration distortion. Another device that operates on the same principle as the RAC is the reflective dot array (RDA) described by Solie (1977). This device is potentially less expensive to fabricate than the RAC because its metal reflecting dots, unlike etched grooves, can be fabricated along with the transducers in a single step.

In spite of the superiority of the RAC and the RDA for large time–bandwidth product dispersive filters, there remain applications for filters with more modest time–bandwidth products (roughly 5–200) and the basic SAW interdigital transducer filter remains an ideal vehicle for implementing these devices.

2. *Design Procedure for Dispersive Filter with Two Dispersive Transducers*

In designing a dispersive filter with two interdigital transducers, several points must be noted in seeking two transducer transfer functions whose product is the required filter transfer function, per Eq. (10). First, although the product of two linear-FM spectra is *not* a linear-FM spectrum, it is

sometimes acceptable to design an *approximately* linear-FM filter with two linear-FM transducers. The simplicity of this approach can be weighed against the sidelobe degradation as predicted, for example, by the author's computer program that uses the generalized circuit model to simulate the performance of transducers, filters, and pairs of filters in cascade. If, instead, the designer decides to use Eq. (10) directly, then the following procedure is suggested:

1. Choose an "unapodized" transducer having a bandwidth slightly greater than the required filter bandwidth and approximately half the required dispersive delay.
2. Use a simple dispersion law (such as linear-FM) in this transducer.
3. Introduce a perturbation to the unapodized design by making the electrodes near each end form a smooth taper down to zero aperture at the end.
4. Calculate the transfer function of this transducer (including the circuit factor) by circuit-model simulation.
5. Using numerical complex division, divide the required filter transfer function by the transfer function found in step 4.
6. Make an estimate of the anticipated capacitance and matching circuit design for the second transducer. Calculate the anticipated circuit factor given by Eq. (82) and multiply it by the quotient found in step 5. The result is the required array factor for the second transducer.
7. Use a numerical Fourier transform on the array factor found in step 6 to determine the waveform $g(t)$ for the second transducer. Make the duration of the resultant waveform $g(t)$ finite by truncating it at both ends. Choose the points of truncation where $|g(t)|$ is as small as is consistent with the transducer length that can be accommodated by the available substrate area.
8. Design the second transducer from $g(t)$ using Eqs. (25) and (29).

3. *Dispersive Filter Example*

The SAW dispersive filter with two dispersive transducers is an ideal vehicle for implementing the "reciprocal ripple" filters developed by Judd (1973). The reciprocal ripple design is used in a pair of radar pulse compression filters (one filter for the transmitter and one for the receiver) when the required time–bandwidth product is low enough that the Fresnel ripple of the linear-FM spectrum (see Klauder *et al.*, 1960) causes unacceptably high time sidelobes in the radar system. The transmitter filter is an ordinary dispersive filter (typically linear-FM) designed according to the steps in Section VII, B, 2 above. The receiver filter has built-in spectral

weighting (typically Hamming), but also has amplitude and phase ripples that are inverse to the Fresnel ripples of the transmitter filter. It is also designed as described in Section VII, B, 2 above. As a result, the system consisting of the transmitter and receiver filters in cascade is nondispersive and the magnitude of its transfer function assumes the smooth shape of the desired (Hamming) weighting function, with very little superimposed ripple. If the compression filter could have an impulse response of infinite duration, the ripple could be eliminated completely and the time sidelobes of the radar system would be suppressed by the amount inherent in the weighting function (e.g., 43 dB for Hamming weighting). In practice, for time–bandwidth products ranging from 5 to 30, the reciprocal ripple technique can reduce time–sidelobe levels by an amount in excess of 10 dB as compared with the sidelobes found when the filters do not use the reciprocal ripple technique. It is useful for time–bandwidth products as large as 100.

C. Tapped Delay Lines

Both of the two principal types of tapped delay line geometries have been illustrated in earlier sections. The line with independent taps (cf. Fig. 4) is used in conjunction with external weighting circuits to implement programmable or adaptive transversal filters. The line with all taps connected to a common sum bar (cf. Fig. 14d) is convenient for implementing a fixed biphase coded tapped delay line. Section III, D, 1, c already described the manner in which the basic geometries are chosen—i.e., the input transducer forms one chip and the tapped transducer determines the code via its tap polarities.

In practice, there are several important design considerations (beyond the fundamental principles of Section III) that must be applied to long tapped delay lines, whether the taps are independent transducers or connected to a common sum bar. First, it is desired that all taps have equal output in the presence of acoustic propagation loss and depletion of SAW power by the "early" taps. Judd *et al.* (1972a) give the procedure for accomplishing this by use of a set of monotonically increasing tap apertures (smallest aperture nearest the input transducer and largest aperture most distant from the input transducer). Another important consideration is that a long sequence of output taps can create a significant regeneration distortion, which is described by the circuit factor as we have seen in Section V. For the case of tapped delay lines, it is useful to view these spurious signals as singly or multiply reflected echoes emergent at each tap after reflection by other taps. This description has been furnished by Judd *et al.* (1973); the design trade-offs between insertion loss and these spurious signals were discussed by Judd *et al.* (1972a).

The basic design of tapped delay lines assumes that each output tap is a spatially discrete sampler with infinite bandwidth. In practice, each tap must contain at least two electrodes, and it may be desirable to use more electrodes in order to lower insertion loss. In doing this, it is necessary to recognize that the taps have finite bandwidth; it may be desirable to increase the input transducer bandwidth to compensate. Also, varying the number of electrodes per tap strongly affects the aforementioned trade-offs between insertion loss and spurious (echo) signals, and it may be necessary to adjust (scale) all the apertures to compensate.

Tapped delay lines often require special compensation for electrode "mass/electrical loading." Even with the use of double electrodes in each tap, the small reflections from the various taps can add in phase and create undesired acoustic transmission stopbands at harmonics of one-half the chip rate. Usually with quartz substrates, this problem is predominantly caused by mass loading and can often be reduced to acceptable levels by using appropriately thin electrode metalization. On strongly piezoelectric substrates such as lithium niobate, the electrical loading dominates and the appropriate remedy is to fill the regions between taps with sets of "dummy" double electrodes as described by Judd *et al.* (1972b). This procedure eliminates all acoustic transmission stopbands within the operating bandwidth of the line, since the lowest-frequency stopband is at twice the operating center frequency.

D. Unapodized Dispersive Transducers in Nondispersive Filters, RAC's, Convolvers, and Acousto-optic Devices

There are numerous applications calling for broadband, unapodized transducers whose transfer function magnitude ($|T_{13}|$) can be arbitrarily specified, but whose phase (ϕ_{13}) is unconstrained. For example, a delay line composed of two identical, unapodized dispersive transducers separated by a translation (cf. Fig. 52) is a nondispersive delay line. This is seen by noting that the filter transfer function is related to the transducer transfer functions (T_{13}, T_{32}) and the delay (τ) by

$$T(f) = T_{13}(f)T_{32}(f)\exp(-j2\pi f\tau) \tag{129}$$

and also noting from Eq. (84) for any choice of electrode positions x_n that $T_{32} = T_{13}^*$. By varying the electrode positions x_n, we can control the shape of $|T(f)|$.

A reflective array compressor (RAC) or acoustic convolver can be made broadband and/or have amplitude weighting introduced by using this type of transducer. In effect, the aforementioned nondispersive delay line is created with the reflective arrays or convolver electrodes inserted

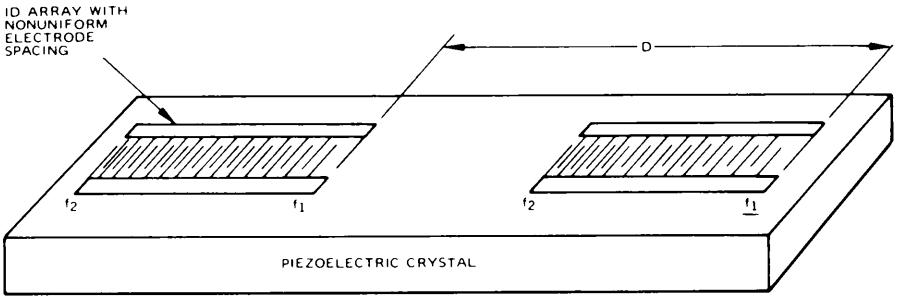


FIG. 52. A nondispersive delay line geometry formed from two identical dispersive transducers separated by a translation. (After Smith *et al.*, 1972.)

between the transducers. Acousto-optic devices such as the integrated optic spectrum analyzer (Barnoski *et al.*, 1978) employ a single transducer for launching a surface wave, but the transducer dispersion is of no concern, so this type of transducer provides a convenient means of achieving broadband operation. In all these devices, an acoustic beam whose amplitude is transversely uniform is required. The *unapodized* dispersive transducer satisfies this requirement as well as a simple periodic transducer, with the only deviation from a uniform beam being due to diffraction.

Although we desire $|T_{13}(f)|^2 \propto e(f)$, we design for $|A(f)|^2 = e(f)$ and predistort $e(f)$ to compensate for the circuit factor $C(f)$ if desired. The method for designing the electrode positions x_n for making $|A(f)|^2$ proportional to a prescribed function $e(f)$ has been described by Smith (1973) and by Nudd *et al.* (1974). The electrode positions satisfy Eq. (25); if we define an "electrode position function" $g(t)$ by $g(t) = \phi(t)/2\pi$, then Eq. (25) becomes

$$g(t_n) = \tfrac{1}{2}n, \qquad 1 \leqslant n \leqslant N. \tag{130}$$

This electrode position function is found from the required bandshape $e(f)$, the desired passband frequency limits f_1 and f_2, and the desired number of electrodes N, according to the differential equation

$$g''(t) = k[g'(t)]^3/e[g'(t)] \tag{131}$$

with initial conditions $g(0) = 0$, $g'(0) = f_1$. The constant k is given by

$$k = 2(f_2 - f_1)e(f_0)/Nf_0^2, \tag{132}$$

where f_0 is a convenient reference frequency in the passband; we choose $f_0 = \frac{1}{2}(f_1 + f_2)$. Note that $e(f)$ is the required *power* transfer function ($|A(f)|^2$) for one transducer and hence the required *amplitude* transfer function ($|T(f)|$) for a two-transducer filter.

Typical choices for $e(f)$ are $e(f) \equiv 1$ (a flat passband),

$$e(f) = a + (1-a)\cos^2\left(\pi \frac{f - f_0}{f_2 - f_1}\right) \tag{133}$$

(Hamming weighting) or $e(f) = |\text{P.L.}\ (f)|^{-1}$ (propagation loss compensation). Furthermore, any of these functions $e(f)$ may be predistorted via multiplication by $|C(f)|^2$ to compensate for a preestimated circuit factor if desired.

The transducer designs that are solutions to Eqs. (130)–(132) turn out to have a total number of electrodes that is approximately equal to N, the "input" or "desired" number. This approximation is most accurate for the flat passband choice, $e(f) \equiv 1$; for more complicated band shapes it is sometimes desirable to use multiple iterations of the design procedure until the desired number of electrodes is achieved.

Throughout Section VII, D it has been stated that both dispersive transducers are unapodized. This is of course a necessary condition (cf. Section II, D) for the filter transfer function to be the square of the transfer function of one transducer. In practice, if the transducers are completely unapodized, such an abrupt truncation of the impulse response would introduce severe ripple on the transfer function, resembling the Fresnel ripple on the spectrum of an abruptly gated linear-FM waveform. Therefore, the transducers are usually designed so that the frequencies f_1 and f_2, which are the lowest and highest electrode synchronous frequencies, span a bandwidth somewhat larger than that of the required operating band. Then, a few electrodes at each end of the transducer are apodized so that the apertures taper smoothly down to zero at the ends of the transducer. The synchronous frequencies of these electrodes lie outside the desired operating band, and the effect of these apodized "tails" is to reduce ripple on the transfer function without significant effect on the transfer function in the operating band.

While the solution to Eqs. (130)–(132) can be written in closed form for many simple functions $e(f)$, we have developed a computer program that finds the solution numerically for an arbitrary choice for $e(f)$. In Fig. 53 we show the results of designing and testing a filter for Hamming weighting according to the transfer function

$$H(f) = 0.08 + 0.92\cos^2\left[\pi \frac{f - 40\ \text{MHz}}{25\ \text{MHz}}\right]. \tag{134}$$

In Fig. 53 (left-hand panel: "theory") the dashed curve is a plot of this desired transfer function and the solid curve is the transfer function obtained from crossed-field model simulation using the 152-electrode designs obtained from numerical solution of Eqs. (130)–(132). The filter was

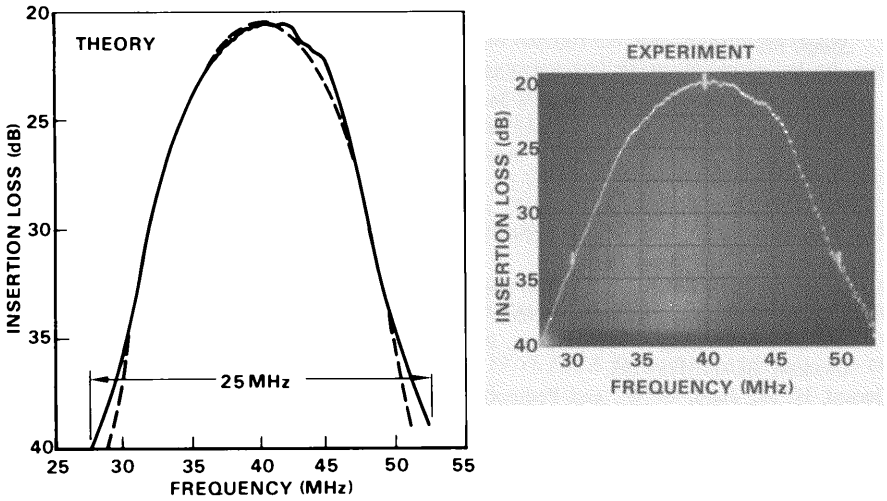


FIG. 53. The ideal, simulated, and measured insertion loss functions of a filter designed for a Hamming bandshape and using the generic geometry of Fig. 52.

fabricated on YZ lithium niobate and its measured insertion loss, shown in Fig. 53 (right-hand panel: "experiment") is in good agreement with the desired and simulated insertion loss curves. This example was chosen to illustrate the use of the design technique for a bandshape having about 20-dB change in insertion loss over the bandwidth. In practice, this type of transducer could be used to implement integral Hamming weighting in a linear-FM RAC filter, or to make a separate SAW Hamming weighting filter for use with any linear-FM filter.

ACKNOWLEDGMENTS

The author wishes to thank T. M. Reeder for a critical reading of the manuscript and L. D. Schenet for typing it.

REFERENCES

Auld, B. A. (1973). "Acoustic Fields and Waves in Solids." Wiley (Interscience), New York.
Auld, B. A., and Kino, G. S. (1971). *IEEE Trans. Electron Devices* **ED-18**, 898–908.
Bahr, A. J., and Lee, R. E. (1973). *Electron. Lett.* **9**, 281–282.
Barnoski, M. K. *et al.* (1978). *Proc. IEEE Ultrason. Symp.* 74–78.
Berlincourt, D. A., Curran, D. R., and Jaffe, H. (1964). *In* "Physical Acoustics" (W. P. Mason, ed.) Vol. IA, pp. 169–270. Academic Press, New York.
Bracewell, R. M. (1965). "The Fourier Transform and Its Applications," McGraw-Hill, New York.
Bristol, T. W. (1972). *Proc. IEEE Ultrason. Symp.* 377–380.

Bristol, T. W., Jones, W. R., Snow, P. B., and Smith, W. R. (1972). *Proc. IEEE Ultrason. Symp.* 343–345.

Bode, H. W. (1945). "Network Analysis and Feedback Amplifier Design," pp. 360–371. Van Nostrand-Reinhold, Princeton, New Jersey.

Campbell, J. J., and Jones, W. R. (1968). *IEEE Trans. Sonics Ultrason.* **SU-15**, 209–217.

Collin R. E., and Zucker, F. J. (1969). "Antenna Theory," Part I, p. 145. McGraw-Hill, New York.

Coquin, G. A., and Tiersten, H. F. (1967). *J. Acoust. Soc. Am.* **41**, 921–939.

DiFranco, J. V., and Rubin, W. L. (1968). "Radar Detection," pp. 51–66. Prentice Hall, Englewood Cliffs, New Jersey.

Engan, H. (1969). *IEEE Trans. Electron Devices* **ED-16**, 1014–1017.

Fano, R. M. (1950). *J. Franklin Inst.* **249**, 57–84, 139–154.

Farnell, G. W., Cermak, I. A., Silvester, P., and Wong, S. K. (1970). *IEEE Trans. Sonics Ultrason.* **SU-17**, 188–195.

Gerard, H. M., Judd, G. W., and Pedinoff, M. E. (1972). *IEEE Trans. Microwave Theory Tech.* **MTT-20**, 188–192.

Gerard, H. M., Smith, W. R., Jones, W. R., and Harrington, J. B. (1973). *IEEE Trans. Microwave Theory Tech.* **MTT-21**, 176–186.

Gerard, H. M., Yao, P. S., and Otto, O. W. (1977). *Proc. IEEE Ultrason. Symp.* 947–951.

Harris, J. F. (1978). *Proc. IEEE* **66**, 51–83.

Hartmann, C. S. (1973). *Proc. IEEE Ultrason. Symp.* 423–426.

Hartmann, C. S., Jones, W. S., and Vollers, H. (1972). *IEEE Trans. Sonics Ultrason.* **SU-19**, 378–381.

Hartmann, C. S., and Secrest, B. G. (1972). *Proc. IEEE Ultrason. Symp.* 413–416.

Hartmann, C. S., Bell, Jr., D. T., and Rosenfeld, R. C. (1973). *IEEE Trans. Microwave Theory Tech.* **MTT-21**, 162–175.

Hofstetter, E. M. (1970). M.I.T. Lincoln Laboratory, Lexington, Massachusetts, Technical Note 1970-42.

Hofstetter, E. M., Openheim, A. V., and Siegel, J. (1971). A new technique for the design of non-recursive digital filters, *Ann. Princeton Conf. Informat. Sci. Syst., 5th, Princeton Univ.*

Jones, W. S., Hartmann, C. S., and Sturdivant, T. D. (1972). *IEEE Trans. Sonics Ultrason.* **SU-19**, 368–377.

Joshi, S. G., and White, R. M. (1968). *J. Appl. Phys.* **39**, 5819–5827.

Judd, G. W. (1973). *Proc. IEEE Ultrason. Symp.* 478–481.

Judd, G. W., Morse, F., and Smith, W. R. (1972a). Acoustic Signal Processing Devices. USAECOM Tech. Rep. 0023-1; Hughes Aircraft Co. Rep. No. FR 72-14-686.

Judd, G. W., Jones, W. R., and Bristol, T. W. (1972b). *Proc. IEEE Ultrason. Symp.* 373–376.

Judd, G. W., Morse, F., Smith, W. R., and Templin, L. (1973). Acoustic Signal Processing Devices. USAECOM Tech. Rep. 0023-F; Hughes Aircraft Co. Rep. No. FR 73-14-1035.

Kallmann, H. E. (1940). *Proc. IRE* **28**, 302–310.

Klauder, J. R., Price, A. C., Darlington, S., and Albersheim, W. J. (1960). *Bell Syst. Tech. J.* **39**, 745–808.

Kodama, T. (1979). *Proc. IEEE Ultrason. Symp.* 522–526.

Krimholtz, R. (1972). *IEEE Trans. Sonics Ultrason.* **SU-19**, 427–436.

Krimholtz, R., Matthaei, G. L., and Hoiem, B. E. (1974). *IEEE Trans. Sonics Ultrason.* **SU-21**, 23–32.

Lakin, K. M. (1974). *IEEE Trans. Microwave Theory Tech.* **MTT-22**, 418–424.

Leedom, D. A., Krimholtz, R., and Matthaei, G. L. (1971). *IEEE Trans. Sonics Ultrason.* **SU-18**, 128–141.

Li, R. C. M., and Melngailis, J. (1975). *IEEE Trans. Sonics Ultrason.* **SU-22**, 189–198.

Masenten, W. K. (1979). Adaptive signal processing, *Int. Specialist Sem. Case Stud. Adv. Signal Processing, Edinburgh, Scotland*. Hughes Aircraft Co. Rep. No. TP 79-14-13.

Mason, W. P. (1948). "Electromechanical Transducers and Wave Filters," 2nd ed., pp. 201–209, 399–409, Van Nostrand-Reinhold, Princeton, New Jersey.

Milsom, R. F., Reilly, N. H. C., and Redwood, M. (1977). *IEEE Trans. Sonics Ultrason.* **SU-24**, 147–166.

Nudd, G. R., Waldner, M., and Zimmerman, R. L. (1974). *IEEE Trans. Microwave Theory Tech.* **MTT-22**, 25–32.

Parks, T. W., and McClellan, J. H. (1972). *IEEE Trans. Audio and Electroacoust.* **AU-20**, 195–199.

Rayleigh, Lord (1885). *Proc. London Math. Soc.* **17**, 4–11.

Reeder, T. M., and Sperry, W. R. (1972). *IEEE Trans. Microwave Theory Tech.* **MTT-20**, 453–458.

Reeder, T. M., Shreve, W. R., and Adams, P. L. (1972). *IEEE Trans. Sonics Ultrason.* **SU-19**, 466–470.

Shannon, C. E. (1949). *Proc. IRE* **37**, 10–21.

Slobodnik, Jr., A. J., Fenstermacher, T. E., Kearns, W. J., Roberts, G. A., and Silva, H. J. (1975). *Proc. IEEE Ultrason. Symp.* 405–407.

Smith, W. R. (1973). *Proc. IEEE G-MTT Microwave Symp.* 117–119.

Smith, W. R. (1974). *IEEE Trans. Microwave Theory Tech.* **MTT-22**, 960–964.

Smith, W. R. (1976a). *Wave Electron.* **2**, 25–63.

Smith, W. R. (1976b). *Proc. IEEE Ultrason. Symp.* 547–552.

Smith, W. R. (1977). *Proc. IEEE Ultrason. Symp.* 524–528.

Smith, W. R., and Pedler, W. F. (1975). *IEEE Trans. Microwave Theory Tech.* **MTT-23**, 853–864.

Smith, W. R., Gerard, H. M., Collins, J. H., Reeder, T. M., and Shaw, H. J. (1969a). *IEEE Trans. Microwave Theory Tech.* **MTT-17**, 856–864.

Smith, W. R., Gerard, H. M., Collins, J. H., Reeder, T. M., and Shaw, H. J. (1969b). *IEEE Trans. Microwave Theory Tech.* **MTT-17**, 865–873.

Smith, W. R., Gerard, H. M., and Jones, W. R. (1972). *IEEE Trans. Microwave Theory Tech.* **MTT-20**, 458–471.

Solie, L. P. (1977). *Proc. IEEE Ultrason. Symp.* 579–584.

Szabo, T. L., and Slobodnik, Jr., A. J. (1973). *IEEE Trans. Sonics Ultrason.* **SU-20**, 240–251.

Szabo, T. L., and Slobodnik, Jr., A. J. (1974). *IEEE Trans. Sonics Ultrason.* **SU-21**, 114–119.

Szabo, T. L., Laker, K. R., and Cohen, E. (1979). *IEEE Trans. Sonics Ultrason.* **SU-26**, 321–333.

Tancrell, R. H. (1974). *IEEE Trans. Sonics Ultrason.* **SU-21**, 12–22.

Tancrell, R. H., and Holland, M. G. (1970). *Proc. IEEE* **59**, 393–409.

Vasile, C. F. (1974). *IEEE Trans. Sonics Ultrason.* **SU-21**, 7–11.

Wagers, R. S. (1976). *IEEE Trans. Sonics Ultrason.* **SU-23**, 113–127.

Wagers, R. S. (1977). *In* "Physical Acoustics" (W. P. Mason and R. N. Thurston, eds.) Vol. XIII, pp. 49–78. Academic Press, New York.

Wagers, R. S. (1979). *IEEE Trans. Sonics Ultrason.* **SU-26**, 105–111.

Weinberg, L. (1962). "Network Analysis and Synthesis," p. 282. McGraw-Hill, New York.

Williamson, R. C., and Smith, H. I. (1973). *IEEE Trans. Microwave Theory Tech.* **MTT-21**, 195–205.

Williamson, R. C. (1976). *Proc. IEEE* **64**, 702–710.

Yamanouchi, K., Nyffeler, F., and Shibayama, K. (1975). *Proc. IEEE Ultrason. Symp.* 317–321.

—3—

Theory of Resonance Scattering

LAWRENCE FLAX

Naval Research Laboratory, Washington, D.C.

GUILLERMO C. GAUNAURD

Naval Surface Weapons Center, White Oak, Silver Spring, Maryland

and

HERBERT ÜBERALL

Physics Department, Catholic University, Washington, D.C.

I. Introduction 191
II. Acoustic-Wave Scattering from Elastic Targets 193
A. Solid Cylinders and Spheres 194
B. Shells and Coated Shells 208
C. Experiments 223
D. Gas Bubbles in Fluids 228
E. Submerged Plates 232
III. Elastic-Wave Scattering from Cavities and Inclusions 240
A. Spherical Fluid-Filled Cavities 240
B. Cylindrical Fluid-Filled Cavities 268
C. Solid Inclusions 273
IV. The Inverse Scattering Problem 281
V. Resonances in Electromagnetic Scattering 285
A. Theory 285
B. Numerical Results 289
References 292

I. Introduction

Most material objects encountered in nature, and most man-made objects as well, are capable of sustaining vibrational or oscillatory processes of one type or another. These may be elastic vibrations of solid bodies, electromagnetic vibrations of dielectrics, acoustic vibrations of fluids or gaseous

PHYSICAL ACOUSTICS, VOL. XV

Copyright © 1981 by Academic Press, Inc.
All rights of reproduction in any form reserved.

media, vibrating ripples on fluid surfaces, and many other types of oscillations. In extended media, such vibrations will occur in the fashion of propagating waves. For bodies of limited size, however, repeated back-and-forth reflections of these waves from the boundaries of the objects will in general, by destructive interference, prevent the appearance of such oscillations *except* for those discrete frequencies (or wavelengths) where the waves may interfere constructively, and form self-sustaining standing waves, known as the eigenvibrations of the object. The corresponding frequencies that belong to the eigenvibrations, which very often form a discrete but infinite set, are called the eigenfrequencies of the object. Of course losses will attenuate the eigenvibrations and broaden the eigenfrequencies into a more or less narrow frequency band.

Suppose such a finite material object is imbedded in another (external) medium with properties different from those of the object. Waves may propagate in the external medium and subsequently interact with the object; reflections and diffraction processes will occur, designated by the common term "scattering." This term also describes the process of particle beams interacting with atoms or atomic nuclei, which is another example appropriate for consideration here since according to quantum mechanics, the incident particle beams may alternatively be represented by a wave picture also.

It is intuitively clear that if the frequency of the incident propagating wave coincides with one of the eigenfrequencies of the target object, the corresponding eigenvibration will appear excited by the incident wave. (There are, of course, many other ways of exciting eigenvibrations, e.g., by the action of point forces). This process of generating eigenvibrations will at the same time strongly affect the scattering process of the incident wave, not the least by the fact that energy is removed from the wave and goes into setting up the vibration. In fact, many previous calculations, both of elastic wave scattering (Johnson and Truell, 1965; McBride and Kraft, 1972) or radar scattering (see, e.g., Ruck *et al.*, 1970), had obtained results that exhibited distinct resonance phenomena in the scattering amplitudes at the eigenfrequencies of the target. In the classical theoretical literature of acoustic, electromagnetic, or elastic-wave scattering, the detailed connection between scattering amplitudes and eigenvibrations of the target has never been explored in any detail (although occasional hints in this direction appear; see, e.g., Junger and Feit, 1972) while in quantum-mechanical scattering (see, e.g., Roman, 1965), the corresponding theory has been well established.

In the fundamental paper of Flax *et al.* (1968), the quantum-mechanical theory of resonance scattering was for the first time applied to a classical phenomenon, namely, acoustic scattering from solid elastic

targets. This study opened up a host of further possibilities in the domain of classical physics, and subsequently, a large number of investigations were performed, applying the resonance method to the scattering of elastic waves from fluid-filled cavities (Haug *et al.*, 1978; Gaunaurd and Überall, 1978, 1979) or from solid inclusions (Flax and Überall, 1980b); of acoustic waves from shells (Murphy *et al.*, 1978), from gas bubbles (Sage *et al.*, 1980; Überall *et al.*, 1979), and from infinite plates (Fiorito and Überall, 1979; Fiorito *et al.*, 1979); to the scattering of electromagnetic waves from dielectrics (Murphy *et al.*, 1980; Gaunaurd *et al.*, 1980b); and to nuclear scattering processes (Farhan *et al.*, 1978; Dragún and Überall, 1980). The acoustical aspects of resonance scattering have been reviewed by Derem (1979).

As a further step in this series of investigations, the target resonances were given physical interpretations in terms of surface waves [mathematically described by "Regge poles" (see, e.g., Newton, 1966)], which, at an eigenfrequency, match phases during repeated circumnavigations of the (finite) target and hence, by constructive interference, build up the resonance in the target. This physical picture provides a basic explanation for the very existence of eigenvibrations in any material object.

In the following, we shall discuss in some detail the development of resonance scattering theory in classical physics, and its application to various examples of acoustic wave scattering together with a discussion of relevant experiments (Section II), of elastic-wave scattering (Section III), and of the corresponding inverse-scattering problem (Section IV). The examples in these sections all refer to scattering objects of regular shapes (separable geometries), but the resonance scattering theory can be applied in a straightforward manner to objects of general shapes using methods of T-matrix theory (Waterman, 1965, 1969; Su 1980; Flax 1981).

Finally, some examples of resonance theory in radar scattering are considered in Section V, in order to demonstrate the basic similarity of the theory in this case to that in the cases of acoustic or elastic scattering. It is believed that the mathematical clarification of resonance excitation in the course of the scattering of incident waves, and the interpretation of the resonances in terms of phase-matched circumferential waves, represents major progress in our understanding of the mechanism of classical scattering processes.

II. Acoustic-Wave Scattering from Elastic Targets

In this section we shall be concerned with the theory of resonance scattering as applied to the case of acoustic waves scattered by elastic objects. In Section II, A, on acoustic scattering from solid cylinders and

spheres, we shall outline the resonance theory in some detail following Flax *et al.* (1978), whereas in the following subsections, on scattering by shells, gas bubbles, and plates, the corresponding theory will be treated more briefly.

A. Solid Cylinders and Spheres

1. *Classical Scattering Theory*

Let us consider an infinite plane acoustic wave $p_0 \exp i(kx - \omega t)$ with propagation constant $k = \omega/c$, incident along the x axis on a solid elastic cylinder of radius a and density ρ_c whose axis coincides with the z axis. At a point $P(r, \phi)$ located in the fluid of density ρ_w surrounding the cylinder, it produces the following scattered field p_{sc}:

$$P_{sc}(r, \phi) = p_0 \sum_{n=0}^{\infty} \varepsilon_n i^n R_n(x) H_n^{(1)}(kr) \cos n\phi, \tag{1a}$$

where

$$R_n(x) = -\frac{J_n(x)F_n - xJ_n'(x)}{H_n^{(1)}(x)F_n - xH_n^{(1)\prime}(x)}. \tag{1b}$$

The geometry is described in Fig. 1. The time dependence $\exp(-i\omega t)$ is

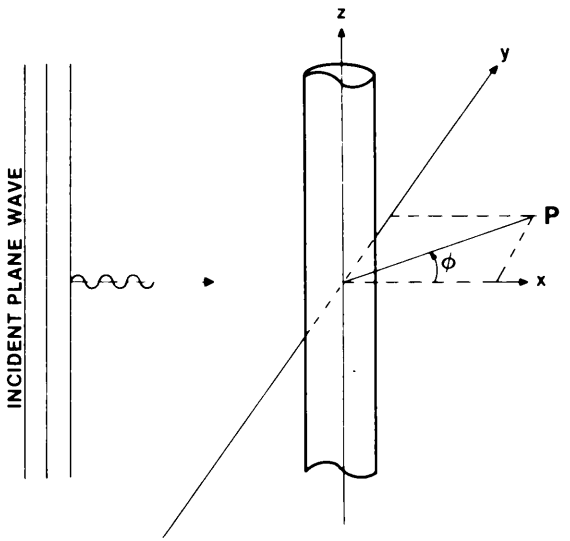


Fig. 1. Geometry used for formulating the problem of sound scattering from an infinite circular cylinder.

suppressed. In Eq. (1a), p_0 is the incident pressure amplitude, ε_n is the Neumann factor ($\varepsilon_n = 2$ for $n = 0$, and $\varepsilon_n = 1$ for $n > 0$), J_n and $H_n^{(1)}$ are the Bessel function and the Hankel function of the first kind, respectively, and the quantities F_n, related to the modal mechanical impedance of the cylinder (Junger and Feit, 1972), are essentially the quotients of two 2×2 determinants

$$F_n(x) = \frac{\rho_w}{\rho_c} x_T^2 \frac{D_n^{(1)}(x)}{D_n^{(2)}(x)}, \tag{1c}$$

where

$$D_n^{(1)} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}, \qquad D_n^{(2)} = \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} \tag{1d}$$

with elements

$$\begin{aligned} a_{12} &= (x_T^2 - 2n^2) J_n(x_L) + 2x_L J_n'(x_L), \\ a_{22} &= x_L J_n'(x_L), \\ a_{32} &= 2n[J_n(x_L) - x_L J_n'(x_L)], \\ a_{13} &= 2n[J_n(x_T) - x_T J_n'(x_T)], \\ a_{23} &= n J_n(x_T), \\ a_{33} &= (x_T^2 - 2n^2) J_n(x_T) + 2x_T J_n'(x_T). \end{aligned} \tag{1e}$$

The argument x of the Bessel and Hankel functions in Eq. (1a) is $x = ka = \omega a / c_w$, where c_w is the speed of sound in the ambient fluid. The prime denotes differentiation with respect to the argument. The elements of Eqs. (1e) contain Bessel functions with arguments $x_L = k_L a = \omega a / c_L$ and $x_T = k_T a = \omega a / c_T$ where c_L and c_T are, respectively, the speeds of longitudinal and transverse waves in the cylinder material.

In the far field where $r \gg a$, $H_n^{(1)}(kr)$ may be written in its asymptotic form

$$H_n^{(1)}(kr) \sim (2/\pi kr)^{1/2} \exp(ikr - in\pi/2 - i\pi/4). \tag{2}$$

Then the far-field scattered pressure becomes

$$p_{sc}(\phi) = p_0 e^{ikr} \left(\frac{2}{i\pi kr} \right)^{1/2} \sum_{n=0}^{\infty} \varepsilon_n R_n(x) \cos n\phi. \tag{3a}$$

For monostatic reflection $\phi = \pi$, this becomes

$$p_{sc}(\pi) = p_0 e^{ikr} \left(\frac{2}{i\pi kr} \right)^{1/2} \sum_{n=0}^{\infty} \varepsilon_n (-1)^n R_n(x). \tag{3b}$$

A quantity called the far-field form function f_∞ is defined to give a

nondimensional representation of the scattered pressure

$$f_\infty(\phi) = (2r/a)^{1/2}(p_{sc}/p_0)e^{-ikr}. \tag{4}$$

From Eqs. (3), we find

$$f_\infty(\phi) = \frac{2}{(i\pi x)^{1/2}} \sum_{n=0}^{\infty} \varepsilon_n R_n(x)\cos n\phi, \tag{5a}$$

$$f_\infty(\pi) = \frac{2}{(i\pi x)^{1/2}} \sum_{n=0}^{\infty} \varepsilon_n(-1)^n R_n(x). \tag{5b}$$

The individual normal modes, or "partial waves," which make up the form functions, are defined as

$$f_n(\phi) = \left[2/(i\pi x)^{1/2}\right]\varepsilon_n R_n(x)\cos n\phi, \tag{5c}$$

so that

$$f_\infty(\phi) = \sum_{n=0}^{\infty} f_n(\phi). \tag{5d}$$

Two limiting cases of these results are the following. If $\rho_c \to \infty$, the

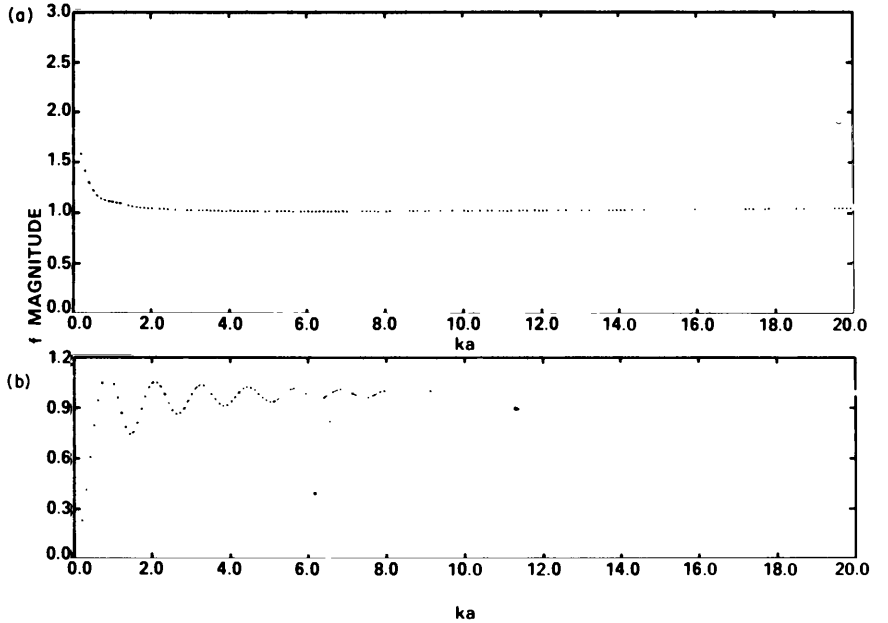


FIG. 2. Modulus of form function for acoustic backscattering from a soft (a) and a rigid (b) cylinder.

solution applies to scattering by a rigid cylinder:

$$[R(x)]_{\text{rigid}} = -J_n'(x)/H_n^{(1)'}(x). \tag{6a}$$

If $\rho_c \to 0$, the solution for a soft cylinder is obtained:

$$[R_n(x)]_{\text{soft}} = -J_n(x)/H_n^{(1)}(x). \tag{6b}$$

In general, the above-defined form function, Eqs. (5), describes the steady-state pressure amplitude scattered from an elastic cylinder by means of a partial-wave (normal-mode) expansion. This formal mathematical series expansion contains all the physical features of the solution, i.e., reflected waves, creeping (or circumferential) waves, and transmitted (or through) waves, see, e.g., Überall (1973) and Brill and Überall (1971). Agreement between partial wave theory and experiment has been shown to be excellent (Dragonette *et al.*, 1974; Dragonette, 1978), and recent investigations demonstrate (Flax, 1977) that even at ka of the order of 1000 no limiting value of the form function is approached, but that for an elastic cylinder, the form function remains oscillating.

Figure 2 shows computed values of $|f_\infty(ka)|$ for a soft and a rigid cylinder, respectively, plotted as functions of ka.

2. *Resonance Formulation*

In Fig. 3, we present the form function $|f_\infty(ka)|$ as calculated from Eq. (5) for an aluminum cylinder immersed in water, plotted versus ka. The curve is calculated in ka steps of $\Delta ka = 0.05$. The form function shows that over the range of $1.5 \leqslant ka \leqslant 4.0$, the reflection from the aluminum cylinder is similar to that from the rigid cylinder (Fig. 2). This region of similitude is followed, however, by abrupt peaks and valleys, and these irregular oscillations continue as $ka \to \infty$ (Flax, 1977). A comparison between this theoretical computation and an experiment using short-pulse experimental techniques (Dragonette *et al.*, 1974; Dragonette, 1978; Dragonette and Flax 1979, 1980) is given in Fig. 4. Agreement between theory and experiment is within 2%.

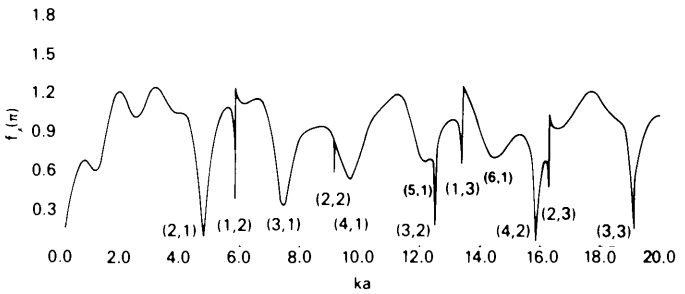


FIG. 3. The form function modulus for an aluminum cylinder in water (Flax *et al.*, 1978).

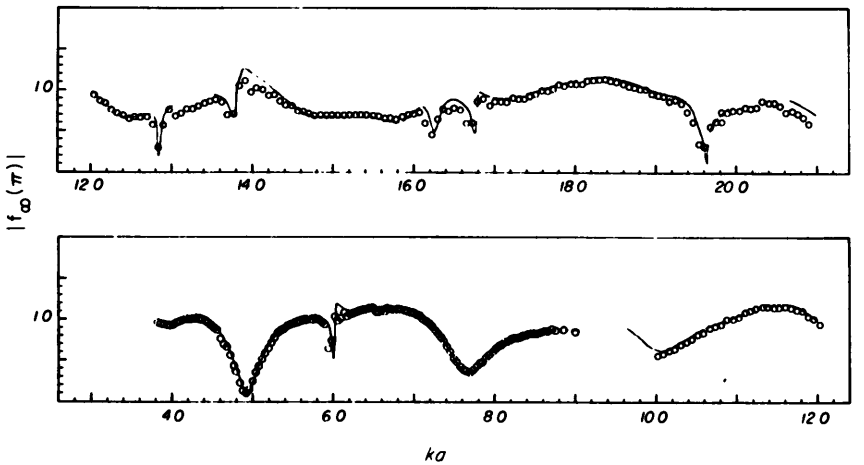


FIG. 4. Comparison of theory (_____) and experiment (OOOOO) for the form function of an aluminum cylinder in water (Dragonette, 1978).

Figure 3 shows, however, that the behavior of a solid aluminum cylinder in water is close to that of a rigid one except for resonances superimposed on the "rigid background." This suggests that the form function should be separable into two terms, a rigid-background contribution and a resonance term. In the following, such a separation will be carried out mathematically.

The scattered pressure of Eq. (1a) may be written as the normal-mode series (setting $p_0 = 1$)

$$p_{sc} = \frac{1}{2} \sum_{n=0}^{\infty} \epsilon_n i^n (S_n - 1) H_n^{(1)}(kr) \cos n\phi, \tag{7}$$

where we introduced the scattering function

$$S_n = e^{2i\delta_n} \tag{8a}$$

of the nth mode, containing the scattering phase shifts δ_n familiar from quantum-mechanical scattering theory; see, e.g., Roman (1965). From Eq. (1a), we find

$$S_n - 1 = 2R_n(x). \tag{8b}$$

For rigid or soft cylinders, we have, respectively,

$$S_n^r = -H_n^{(2)\prime}(x)/H_n^{(1)\prime}(x) = e^{2i\epsilon_n^r} \tag{9a}$$

and

$$S_n^s = -H_n^{(2)}(x)/H_n^{(1)}(x) = e^{2i\epsilon_n^s}. \tag{9b}$$

The corresponding phase shifts can be shown to be given by the real quantities

$$\tan \varepsilon_n^r = J_n'(x)/Y_n'(x), \qquad \tan \varepsilon_n^s = J_n(x)/Y_n(x). \tag{10}$$

A rigid or soft S function may be factored out from the elastic S function, giving

$$S_n = S_n^r\left(F_n^{-1} - z_n^{(2)-1}\right)/\left(F_n^{-1} - z_n^{(1)-1}\right) \tag{11a}$$

or

$$S_n = S_n^s\left(F_n - z_n^{(2)}\right)/\left(F_n - z_n^{(1)}\right), \tag{11b}$$

where the quantities

$$z_n^{(i)} = xH_n^{(i)'}(x)/H_n^{(i)}(x), \qquad i = 1,2 \tag{12}$$

are related to the modal specific acoustic impedances (Junger and Feit, 1972). Equations (11) show that the limits of a soft ($\rho_c \ll \rho_w$) or rigid ($\rho_c \gg \rho_w$) cylinder correctly reproduce the corresponding soft- or rigid-cylinder scattering functions S_n^s and S_n^r, respectively.

The quantities $z_n^{(i)}$ of Eq. (12) can be separated into real and imaginary parts:

$$z_n^{(1,2)-1} = \Delta_n^r \pm is_n^r \tag{13a}$$

or

$$z_n^{(1,2)} = \Delta_n^s \pm is_n^s \tag{13b}$$

where, e.g.,

$$s_n^r = -\frac{2}{\pi x^2}\frac{1}{\left[J_n'(x)\right]^2 + \left[Y_n'(x)\right]^2} \qquad (<0). \tag{14}$$

The linear-approximation method of nuclear resonance theory is now used, in which the resonance frequencies x_n^r or x_n^s are defined by the conditions

$$F_n^{-1}(x_n^r) = \Delta_n^r, \tag{15a}$$

$$F_n(x_n^s) = \Delta_n^s. \tag{15b}$$

The quantities $F_n^{-1} - \Delta_n^r$ or $F_n - \Delta_n^s$ will be expanded in a Taylor series in x in the vicinity of one of the resonance frequencies:

$$F_n^{-1}(x) \cong \Delta_n^r + \beta_n^r(x - x_n^r), \tag{16a}$$

$$F_n(x) \cong \Delta_n^s + \beta_n^s(x - x_n^s). \tag{16b}$$

It will be kept in mind that the multiplicity of solutions $x_n^{r,s}$ of Eqs. (15) should be distinguished by a further index l in Eqs. (16). Resonance widths

are defined by

$$\Gamma_n^{\mathrm{r}} = -2s_n^{\mathrm{r}}/\beta_n^{\mathrm{r}} \qquad (>0), \tag{17a}$$

$$\Gamma_n^{\mathrm{s}} = -2s_n^{\mathrm{s}}/\beta_n^{\mathrm{s}} \qquad (>0), \tag{17b}$$

and the scattering functions may be rewritten in either of two resonance forms

$$S_n = e^{2i\delta_n} = S_n^{\mathrm{r}} \sum_{l=1}^{\infty} \frac{x - x_{nl}^{\mathrm{r}} - \frac{1}{2}i\Gamma_{nl}^{\mathrm{r}}}{x - x_{nl}^{\mathrm{r}} + \frac{1}{2}i\Gamma_{nl}^{\mathrm{r}}}, \tag{18a}$$

$$S_n = S_n^{\mathrm{s}} \sum_{l=1}^{\infty} \frac{x - x_{nl}^{\mathrm{s}} - \frac{1}{2}i\Gamma_{nl}^{\mathrm{s}}}{x - x_{nl}^{\mathrm{s}} + \frac{1}{2}i\Gamma_{nl}^{\mathrm{s}}}, \tag{18b}$$

labeling the lth resonance frequency of the nth mode x_{nl} by the integer l. They contain a resonance pole at the complex frequencies $x = x_{\mathrm{p}}$ given by

$$x_{\mathrm{p}} = x_n - \tfrac{1}{2}i\Gamma_n, \tag{19}$$

and zero $x = x_z$ at

$$x_z = x_n + \tfrac{1}{2}i\Gamma_n. \tag{20}$$

The pole is located in the lower half of the complex-x plane, and the zero in the upper half-plane, at the same position along the real axis, and at the same distance above the axis that the pole lies below.

Equations (9) and (18) demonstrate the unitarity of S_n: the ensuing relations

$$|S_n| = |S_n^{\mathrm{r}}| = |S_n^{\mathrm{s}}| = 1 \tag{21}$$

express the fact that no energy is lost in the scattering process (for elastic scatterers, this is due to our implied assumption of no absorption, i.e., real values of F_n).

Recalling Eq. (8a), the quantity $S_n - 1$ that appears in Eq. (7) can be written

$$S_n - 1 = 2ie^{i\delta_n} \sin \delta_n. \tag{22}$$

Using Eqs. (11) and (12), this can be expressed (for the rigid-background case) as

$$S_n - 1 = 2ie^{2i\varepsilon_n^{\mathrm{r}}} \left[\frac{s_n^{\mathrm{r}}}{F_n^{-1} + \Delta_n^{\mathrm{r}} - is_n^{\mathrm{r}}} + e^{-i\varepsilon_n^{\mathrm{r}}} \sin \varepsilon_n^{\mathrm{r}} \right] \tag{23a}$$

or using Eqs. (18), in the resonance form

$$S_n - 1 = 2ie^{2i\varepsilon_n^{\mathrm{r}}} \left[\sum_{l=1}^{\infty} \frac{\frac{1}{2}\Gamma_{nl}^{\mathrm{r}}}{x_{nl}^{\mathrm{r}} - x - \frac{1}{2}i\Gamma_{nl}^{\mathrm{r}}} + e^{-i\varepsilon_n^{\mathrm{r}}} \sin \varepsilon_n^{\mathrm{r}} \right], \tag{23b}$$

with analogous expressions involving the corresponding "soft" parameters. With Eq. (5a), one then has the partial wave form function

$$f_n(\phi) = 2i\varepsilon_n(\pi ika)^{-1/2} e^{i\delta_n} \sin\delta_n \cos n\phi \tag{24a}$$

$$= 2i\varepsilon_n(\pi ika)^{-1/2} e^{2i\varepsilon_n^r} \times \left[\sum_{l=1}^{\infty} \frac{\frac{1}{2}\Gamma_{nl}^r}{x_{nl}^r - x - \frac{1}{2}i\Gamma_{nl}^r} + e^{-i\varepsilon_n^r} \sin\varepsilon_n^r \right] \cos n\phi. \tag{24b}$$

The first term of Eq. (24b) represents the resonance contribution and the second term represents the smooth "background" corresponding to a rigid scatterer. It will be seen below for the example of scattering from an aluminum cylinder in water that the background corresponds to the scattering amplitude from a rigid cylinder, upon which a large number of interfering resonances are superimposed. In the case of sound scattering from air-filled, very thin elastic shells in a fluid, the background corresponds to a soft scatterer (see Section II, B). The representation of Eq. (24b) shows that the complex eigenfrequencies of the scatterer are determined by the location of the resonance poles in the complex frequency (x) plane, whose real parts determine the resonance frequencies in the scattering amplitude. The imaginary parts describe the widths of the resonances, i.e., the decay of the resonating fields.

Other authors (Faran, e.g., 1951) have observed that the locations in x of the resonances, obtained both theoretically and experimentally in the scattering amplitude, coincide with the eigenfrequencies of elastic vibrations of the scattering object. In fact, resonances occur approximately at ka values of which

$$D_n^{(2)}(ka) = 0 \tag{25}$$

with $D_n^{(2)}$ the determinant introduced in Eq. (1). (This point will be refined in Section II, D). This leads to real eigenfrequencies for a free body ($\rho_w \ll \rho_c$), but to complex eigenfrequencies for the fluid-loaded body. For the example of an aluminum cylinder in water, the solutions of Eq. (25) agree closely with the real eigenfrequencies. From our formalism, it is clear that the complex eigenfrequencies of the scatterer are precisely the locations of the resonance poles, Eq. (19), in the complex frequency plane, whose real parts determine the resonance frequencies in the scattering amplitude.

In addition to the scattering field, the interior fields of the scatterer can also be determined. The vector displacement **u** within the cylinder may be derived from a scalar (Ψ) and a vector potential (**A**). Then

$$\mathbf{u} = -\nabla\Psi + \nabla \times \mathbf{A}. \tag{26}$$

If one assumes that Ψ and **A** satisfy the scalar and vector wave equation,

then

$$\Psi = \sum_{n=0}^{\infty} \varepsilon_n i^n B_n J_n(k_L r) \cos n\phi, \tag{27a}$$

$$A_z = \sum_{n=0}^{\infty} \varepsilon_n i^n E_n J_n(k_T r) \sin n\phi, \tag{27b}$$

where A_z is the only nonvanishing component of **A**. The coefficients are given in terms of our preceding notation as

$$B_n = \frac{2i}{\pi \rho_w \omega^2} \frac{a_{33}}{x H_n^{(1)\prime}(x) D_n^{(1)}\left(z_n^{(1)-1} - F_n^{-1}\right)}, \tag{28a}$$

$$E_n = \frac{2i}{\pi \rho_w \omega^2} \frac{a_{32}}{x H_n^{(1)\prime}(x) D_n^{(1)}\left(z_n^{(1)-1} - F_n^{-1}\right)}, \tag{28b}$$

where only the version leading to the rigid background has been written. Using the expansion of F_n^{-1} in Eq. (16) leads to the resonance expressions for Ψ and A_z:

$$\Psi = \frac{2}{\pi i \rho_w \omega^2} \sum_{\substack{n=0 \\ l=1}}^{\infty} \frac{\varepsilon_n i^n}{\beta_n^r} \frac{a_{33}}{x H_n^{(1)\prime} D_n^{(1)}} \frac{J_n(k_L r) \cos n\phi}{x - x_{nl}^r + \frac{1}{2} i \Gamma_{nl}^r}, \tag{29a}$$

$$A_z = \frac{2}{\pi i \rho_w \omega^2} \sum_{\substack{n=0 \\ l=1}}^{\infty} \frac{\varepsilon_n i^n}{\beta_n^r} \frac{a_{32}}{x H_n^{(1)\prime} D_n^{(1)}} \frac{J_n(k_T r) \sin n\phi}{x - x_{nl}^r + \frac{1}{2} i \Gamma_{nl}^r}. \tag{29b}$$

These results are of a pure resonance form, without (rigid or soft) background terms. This is expected since by definition, rigid or soft objects are impenetrable and admit no fields in their interior. Numerical evaluations of Eqs. (29) were done by Flax *et al.* (1980).

Coming back to the analysis of Eq. (24), we note that for scattering at a frequency between two eigenfrequencies of vibration of the elastic body, the scatterer appears as an impenetrable object and scatters sound accordingly. This implies that far from a narrow resonance, the scatterer behaves as a rigid or soft cylinder that the incident wave hardly penetrates. In contrast, at frequencies in the immediate neighborhood of an eigenfrequency, a scattering resonance is excited by the incident wave and the field penetrates into the body. In the scattering amplitude, the resonance scattering interferes with the background, and this interference causes the strongly oscillatory structure that is observed in the total scattering amplitude as a function of frequency.

3. *Scattering from Elastic Spheres*

The case of plane-wave acoustic scattering from an elastic sphere of radius a can be treated quite analogously to that of a cylinder (Flax, 1974,

1975). The scattered amplitude (for an incident wave of unit amplitude) is

$$p_{sc} = \frac{1}{2} \sum_{n=0}^{\infty} (2n+1) i^n (S_n - 1) h_n^{(1)}(kr) P_n(\cos\theta) \tag{30}$$

with $S_n = e^{2i\delta_n}$. For this latter quantity, Eq. (11) is obtained again, with

$$S_n^r = -h_n^{(2)'}(x)/h_n^{(1)'}(x) \equiv e^{2i\xi_n^r}, \tag{31a}$$

$$S_n^s = -h_n^{(2)}(x)/h_n^{(1)}(x) \equiv e^{2i\xi_n^s}, \tag{31b}$$

and

$$z_n^{(i)} = x h_n^{(i)'}(x)/h_n^{(i)}(x), \qquad i = 1, 2, \tag{31c}$$

where $x = ka$. The real quantity F_n is now given by Hickling (1962); it contains spherical cylinder functions with arguments x_L and x_T, and it is proportional to ρ_w/ρ_s, where ρ_s is the density of the sphere, so that as for the cylinder, the correct soft ($\rho_s \ll \rho_w$) and rigid ($\rho_s \gg \rho_w$) limits for the S function are obtained. The remaining analysis is similar to that of the cylinder, with slightly different resonance parameters, e.g.,

$$s_n^s = x^{-2}\left[j_n^2(x) + y_n^2(x) \right]^{-1} \tag{32}$$

etc., and one obtains for the form function

$$f_n(\theta) = (2/ka)(2n+1) e^{i\delta_n} \sin\delta_n P_n(\cos\theta) \tag{33a}$$

again an expression (e.g. for the rigid-background case)

$$f_n(\theta) = \frac{2}{ka}(2n+1) e^{2i\xi_n^r} \times \left[\sum_{l=1}^{\infty} \frac{\frac{1}{2}\Gamma_{nl}^r}{x_{nl}^r - x - \frac{1}{2} i \Gamma_{nl}^r} + e^{-i\xi_n^r} \sin\xi_n^r \right] P_n(\cos\theta), \tag{33b}$$

which consists of resonances superimposed on an impenetrable background.

4. *Numerical Results and Discussion*

Results of the preceding resonance theory will now be illustrated by numerical calculations for an aluminum cylinder immersed in water. We have demonstrated that the irregular characteristics of the form function for an elastic scatterer are related to the normal modes of free vibration of the body. It will be shown in the following figures that these resonances are superimposed on a background of reflection associated with rigid scatterers. Hence the elastic cylinder can be regarded as rigid except in the frequency intervals over which the resonances occur.

In Fig. 5, we plot $|f_n(\pi)|$ from Eq. (5c) versus $x = ka$ for a number of partial waves, $n = 0$ up to $n = 5$. In each amplitude one recognizes a series

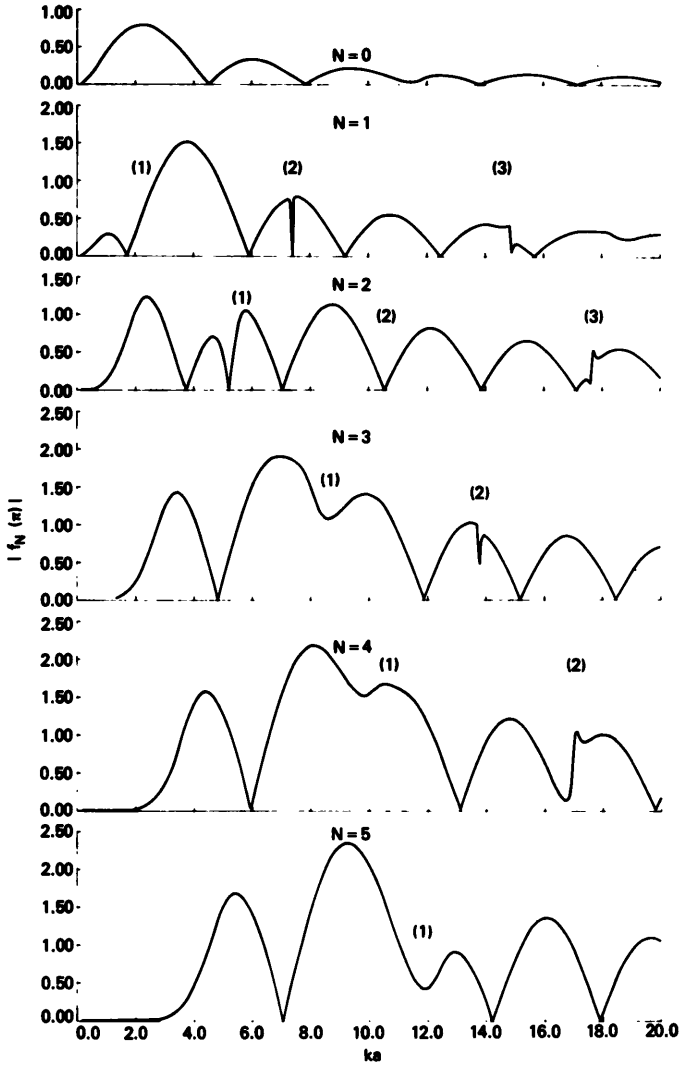


FIG. 5. Partial wave amplitude moduli $|f_n(\pi)|$ for the first six partial waves ($n = 0 - 5$)), plotted versus ka for an aluminum cylinder (Flax *et al.*, 1978).

of narrow resonances, which are labeled by the integer l, superimposed on (and interfering with) a smooth background; the latter corresponds to the scattering amplitude of the rigid cylinder, as seen by a comparison with Fig. 6. The individual partial waves in Fig. 6 have no resonance irregularities.

A more dramatic example of the elastic solution for a solid cylinder is shown in Fig. 7. Here, we subtract the rigid cylinder form function

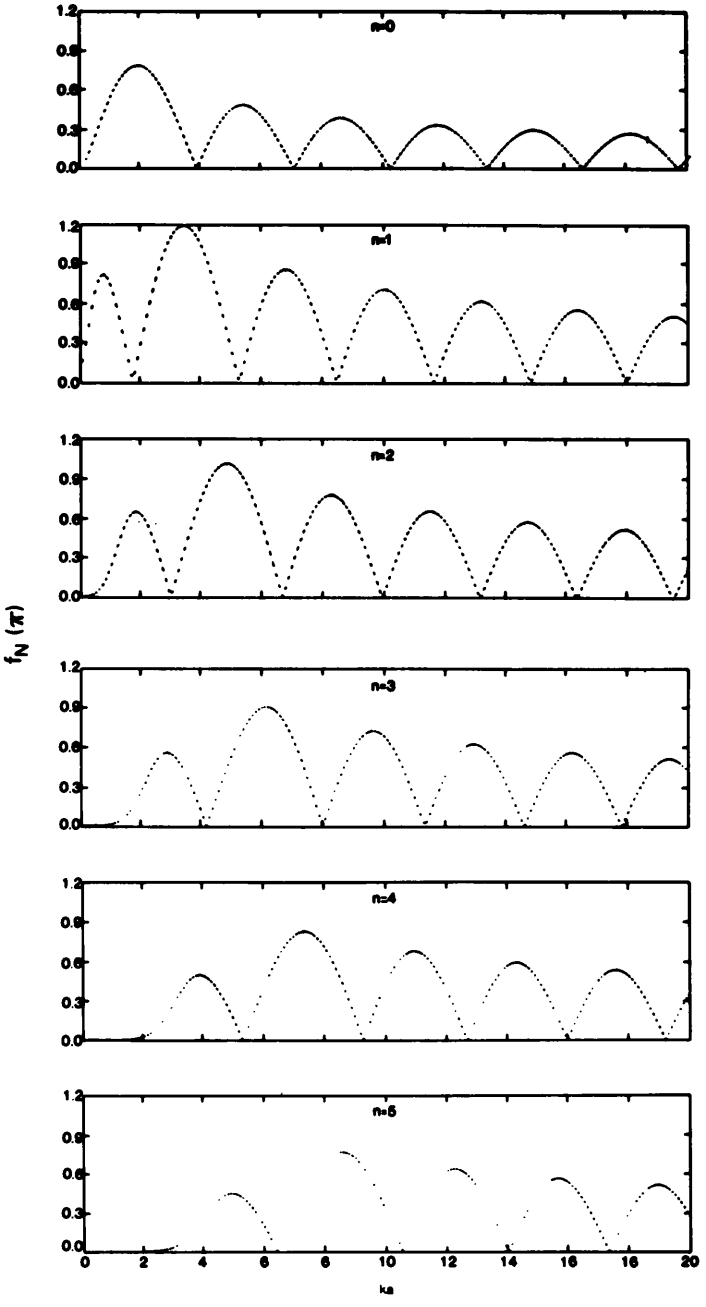


FIG. 6. Moduli of the individual partial wave amplitudes $|f_n(\pi)|$ for $n = 0 - 5$ for a rigid cylinder.

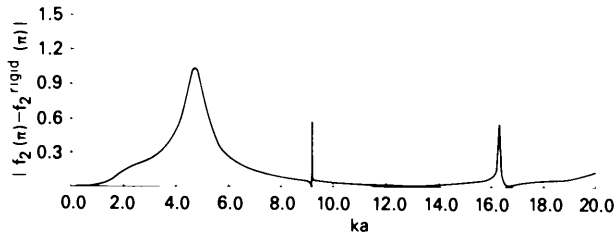


FIG. 7. Modulus of $n = 2$ partial-wave scattering amplitude with rigid background suubtracted $|f_2(\pi) - f_2^{\text{rigid}}(\pi)|$, for aluminum cylinder in water (Flax *et al.*, 1978).

$f_n^{\text{rigid}}(\pi)$, obtained in the limit $\rho_c \to \infty$, from $f_n(\pi)$ and plot the resulting quantity $|f_n(\pi) - f_n^{\text{rigid}}(\pi)|$ versus ka. This procedure is seen to leave us with pure resonances, whose widths in x can be clearly determined from the figure.

In Fig. 5 one may discern families of resonances labeled by $l = 1, 2, 3, \ldots$ whose members appear in all the partial waves, shifting to higher frequencies from one partial wave to the next. Each family may be identified with one corresponding elastic "creeping wave" pole (Regge pole) in the scattering amplitude as will be discussed in Section II, B. The individual resonances that show up in the total form function (summed over all modes n) of Fig. 3 are labeled by the integers (n, l), where n is the mode number and l the eigenfrequency label for a given mode. Physically (see below, and cf. Fig. 7), $l = 1$ represents the Rayleigh wave, and $l = 2, 3, \ldots$ the whispering-gallery waves.

The physical reason that causes the nth mode to resonate at its eigenfrequency can be explained by studying the vibrational features of the individual normal modes. The $n = 0$ partial wave corresponds to the "breathing mode" where the pressure contours around the cylinder move radially in and out. The $n = 1$ term is the dipole mode whose contour motion is rigidly back and forth. The $n = 2$ term is the quadrupole mode whose contours oscillate between pro- and oblate ellipses, etc. Each of these motions $n \geq 1$ can be represented by a standing wave, equivalent to a pair of traveling waves $\exp[i(\pm n\phi - \omega t)]$ propagating in opposite directions with phase velocities

$$c_n(ka) = kac/n, \tag{34a}$$

so that exactly n wavelengths fit over the circumference of the cylinder. For each given mode n, there appears a multiplicity $l = 1, 2, \ldots$ of eigenfrequencies obtained from Eq. (15) or Eq. (25). The multiplicity represents the possibility of various radial deformations of the pressure contours in the fluid belonging to one and the same azimuthal deformation (the latter

characterizing the nth mode). The breathing modes $(0, l)$ are not strongly excited, as evidenced from Fig. 5. The radial deformation of the pressure contours for the breathing mode may occur uniformly radially outward, $(0, 1)$, or outward near the cylinder but inward further away, $(0, 2)$, etc., and similarly for the higher modes $n \geqslant 1$.

In the following, we shall show that the lth resonance family, considered as a function of a continuous (and even complex) mode number $\hat{n}_l$, represents a circumferential (or surface) wave $\exp[i(\hat{n}_l\phi - \omega t)]$ on the cylinder, with phase velocity

$$c_l(ka) = kac/\mathrm{Re}\,\hat{n}_l. \tag{34b}$$

It manifests itself physically at those frequencies at which $\hat{n}_l$ lies close to the integer n, so that $c_l(ka) \to c_n(ka)$. Its phase velocity then coincides with the modal phase velocity c_n, and according to Cremer's coincidence condition (Cremer and Heckl, 1960), this coincidence excites the physical resonance. Viewed differently, the wavelength of the surface wave will then span the circumference an integer n number of times, and the phase match of the surface wave after its repeated circumnavigations leads to constructive interference, and hence to a resonant buildup. This interpretation provides a physical explanation for the very existence of the eigenfrequencies of an elastic object.

Figures 5 and 7 reveal mostly narrow resonances with slightly different shapes. The $(n, 1)$ resonances are related to the surface wave, which we denote as the Rayleigh wave. Its phase velocity approaches that of the familiar Rayleigh wave on a flat infinite half-space as $ka \to \infty$ (Frisk *et al.* 1975). The $(1, 1)$ mode is generated in a region in Fig. 3 where the Franz wave or rigid reflection predominates, and is not observed in this figure. The $(2, 1)$, $(3, 1)$, and $(4, 1)$ Rayleigh resonances are observed to have a striking effect on the form function in the range $4 \leqslant ka \leqslant 10$. They have considerably larger width than the whispering-gallery resonances (Fig. 7). Nulls in the form function at these ka positions stem from the interference of the Rayleigh wave with the specular-reflection amplitude. The excitation of the $(2, 1)$ resonance at the value $ka \cong 4.4$, where two Rayleigh wavelengths span the circumference of the cylinder, marks the transition between rigid and elastic behavior: below, there exists a region where the behavior of the cylinder is purely rigid, i.e., dominated by the interference of the specular reflection and the Franz wave [a slow circumferential wave propagating in the exterior fluid (Überall, 1973)]. This is true for most elastic materials.

The whispering-gallery modes (Dickey *et al*, 1976), corresponding to quite narrow resonances (Fig. 7) labeled by $l = 2, 3, \ldots$, become lateral waves in the limit as $ka \to \infty$ (Frisk and Überall, 1976).

B. Shells and Coated Shells

1. *Cylindrical Shells; Intermediate Background*

The resonance theory for cylindrical shells is identical to that of Sections II, A, 1 and 2 except for a replacement of the two 2×2 determinants $D_n^{(i)}$ in Eqs. (1c) and (1d) by two 5×5 determinants (Murphy *et al.*, 1978), which may be obtained from elasticity theory as given in the literature (Doolittle and Überall, 1966; Uginčius and Überall, 1968), and except for the use of an "intermediate background" S function S_n^i in place of either the rigid-cylinder (S_n^r) or the soft-cylinder (S_n^s) background function in Eqs. (11) if necessary.

We consider a fluid-filled elastic cylindrical shell of outer radius a and inner radius b, which is imbedded in another fluid and whose axis coincides with the z axis. It is insonified by a plane wave incident along the x axis. The mechanical impedance F_n of Eq. (1c) contains cylinder functions with arguments $x_i = k_i a = \omega a/c_i$, $y_i = k_i b = \omega b/c_i$ $(i = w, \mathrm{L}, \mathrm{T}, 3)$ where $i = w$ (often suppressed) indicates the external and $i = 3$ the internal fluid; c_L and c_T are longitudinal and transverse bulk wave speeds in the shell material.

We have obtained numerical results of the appropriate modal form functions $f_n(\phi)$, Eq. (5c), and the resonances contained therein assuming air-filled (i.e., for practical purposes, evacuated) aluminum shells imbedded in water, of various thicknesses measured by the ratio b/a. It is intuitively clear that for a thick shell (in fact, valid for $b/a \lesssim 0.9$), the air-filled cylinder will behave as a rigid cylinder (as in the case of the solid cylinder of Sections II, A, 1 and 2) except for the narrow shell resonances, and for a very thin shell ($b/a \geqslant 0.995$), the behavior will be that of a soft (pressure-release) cylinder since the shell material is here insignificant. However, between these b/a values the "background" (the term to be subtracted from the total form function in order to leave over the pure resonance terms) will have neither rigid nor soft, but "intermediate" properties as will be demonstrated.

In Fig. 5, we showed $|f_n(\pi)|$, plotted versus $x = ka$ for $n = 0$–5, for a solid aluminum cylinder in water. We find a practically identical plot for an air-filled aluminum shell with $b/a = 0.2$, while as b/a increases, the resonances become much more widely spaced. In Fig. 8 (Breitenbach, 1979) we show plots of the modal form function moduli $|f_n(\pi) - f_n^{\text{rigid}}(\pi)|$, with a rigid background subtracted so that pure resonances are left over, for $b/a = 0.85$ and for the partial waves $n = 0$–10. Here, only three resonances are visible in the range $0 < ka < 40$.

If the rigid background were subtracted for a very thin air-filled shell also, no pure resonances would be left over; this would, however, be the

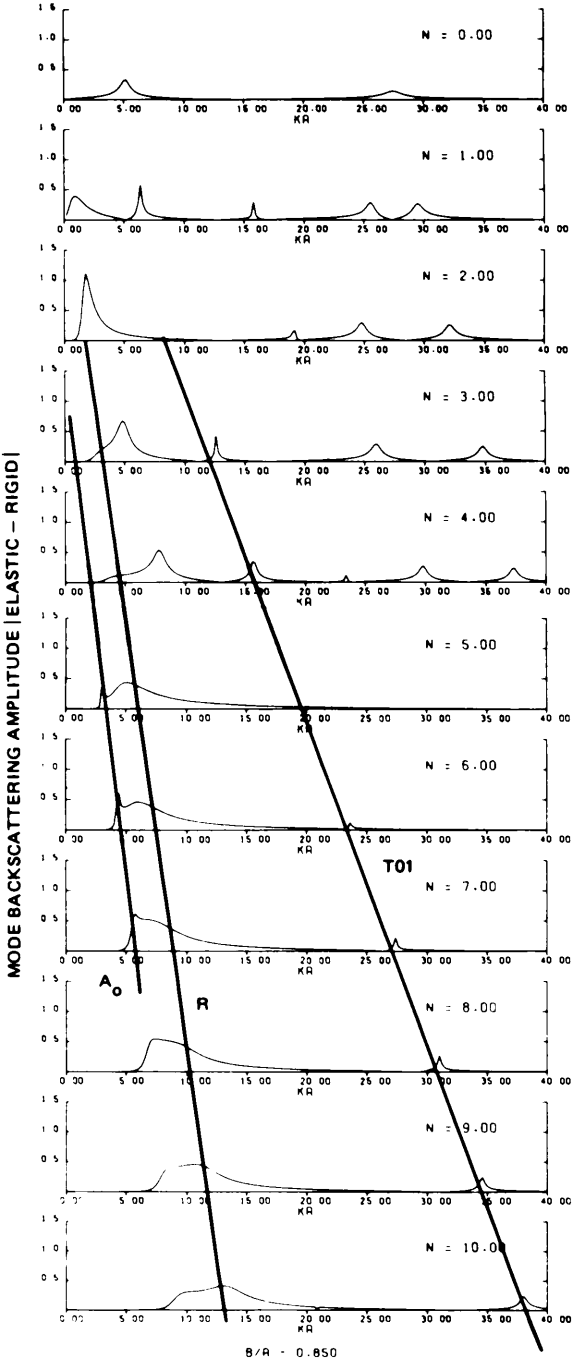


FIG. 8. Modulus $|f_n(\pi) - f_n^{\text{rigid}}(\pi)|$ for an air-filled cylindrical aluminum shell in water with $b/a = 0.85$, $n = 0$–10 plotted versus $ka < 40$. (Breitenbach, 1979.)

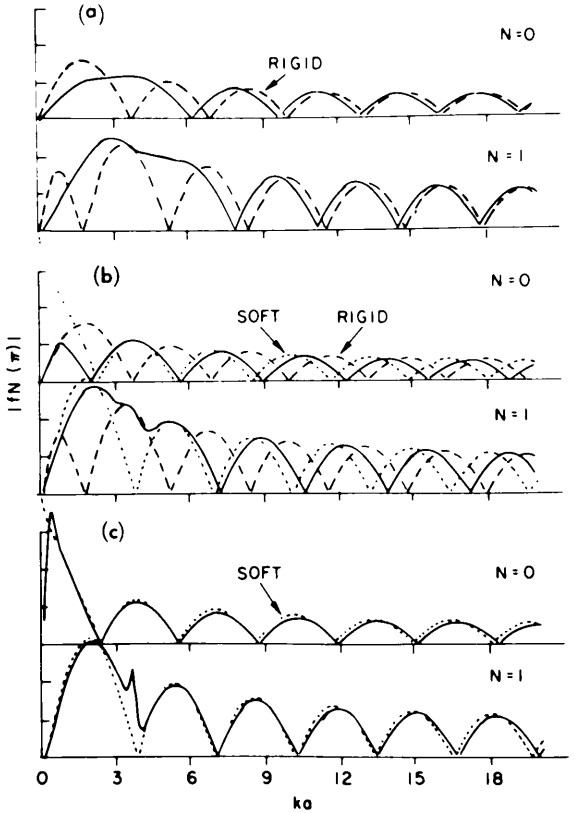


FIG. 9. Moduli $|f_n(\pi)|$ $(n = 0, 1)$ for $b/a = 0.9$ with rigid background, $b/a = 0.975$ with rigid and soft background, and $b/a = 0.995$ with soft background, for an air-filled cylindrical aluminum shell in water (Murphy *et al.*, 1978).

case if the soft background were subtracted. Figure 9 illustrates the situation. Here, $|f_n(\pi)|$ is plotted versus ka, for the partial waves $n = 0$ and 1, and for shell thicknesses $b/a = 0.900$, 0.975, and 0.995 as indicated. Smooth results are obtained with only one interfering resonance feature at $x \sim 3$. For comparison, $|f_n^{\text{rigid}}(\pi)|$ and $|f_n^{\text{soft}}(\pi)|$ are also entered, and it is seen that for $b/a = 0.900$, the background is still almost rigid, and for $b/a = 0.995$, it is very close to soft, while at $b/a = 0.975$ it is halfway in between the rigid and the soft background.

The theory of the intermediate background (at least for the case of empty elastic shells) has been worked out recently (Murphy *et al.*, 1978, 1979, 1979a). It is based on the observation that $F_n(x)$ of Eq. (1c) is relatively slowly varying, except in the vicinity of certain frequencies determined by the vanishing of $D_n^{(2)}$, Eq. (25), where $F_n(x)$ has a narrow

singularity. The magnitude of the slowly varying "plateau" of $F_n(x)$ is large for a very thin (near-soft), and close to zero for a thick (near-rigid) shell. If we subtract from $F_n(x)$ a quantity $F_n(x)$ where x_0 is any point on the plateau of $F_n(x)$, the resulting graph of the difference function will look like a graph of $F_n(x)$ for the near-rigid case. Accordingly, we define the difference functions

$$\bar{F}_n(x) = F_n(x) - F_n(x_0) \tag{35a}$$

and

$$\bar{z}_n^{(i)}(x) = z_n^{(i)}(x) - F_n(x_0), \qquad i = 1, 2. \tag{35b}$$

Using these definitions in Eqs. (11a) or (11b), we obtain the S function in the form

$$S_n = S_n^i\left(\bar{F}_n^{-1} - \bar{z}_n^{(2)-1}\right)/\left(\bar{F}_n^{-1} - \bar{z}_n^{(1)-1}\right), \tag{36a}$$

with an intermediate-background S function given by

$$S_n^i = S_n^s\left(\bar{z}_n^{(2)}/\bar{z}_n^{(1)}\right). \tag{36b}$$

Equation (36a) may then be subjected to the resonance treatment as before.

The use of the intermediate background formalism is illustrated in Fig. 10. Figure 10a shows a plot of F_1 versus x for an aluminum cylinder ($b/a = 0.9$) in water, with one resonance feature near $x = 6$. The above-mentioned plateau and singularity behavior of F_1 are visible, and the resonance is near the intersection of $\mathrm{Re}\, z_1^{(i)}$ (dashed curve) with F_1. Figure 10b shows $|f_1(\pi)|$. Figure 10c gives $|f_1(\pi) - f_1^i(\pi)|$, and one sees that all the background has been removed and the resonance amplitude alone remains. Figures 10d and 10e demonstrate that the resonance cannot be obtained by subtracting a rigid or a soft background.

It was mentioned in connection with Eq. (34b) that the resonances may be thought to be caused by the phase matching of surface waves during their repeated circumnavigations of the cylinder, an integer number of their wavelengths spanning the circumference at resonance. A detailed proof will be given for the sphere below; at this point, we sketch the proof for the cylinder.

The S function of Eqs. (18) has resonance poles x_p in the complex frequency plane which show up as resonances, e.g., in Figs. 7 and 8. If the mode number n is considered a variable, S_n will also have poles in the complex-n plane, obtained by expanding (suppressing the background index r, s, or i):

$$x_{nl} \cong x_{n_l l} + (n - n_l)x'_{n_l l} + \cdots \tag{37}$$

with $x'_{n_l l} > 0$ since, e.g., in Fig. 8, the resonance frequencies are seen to

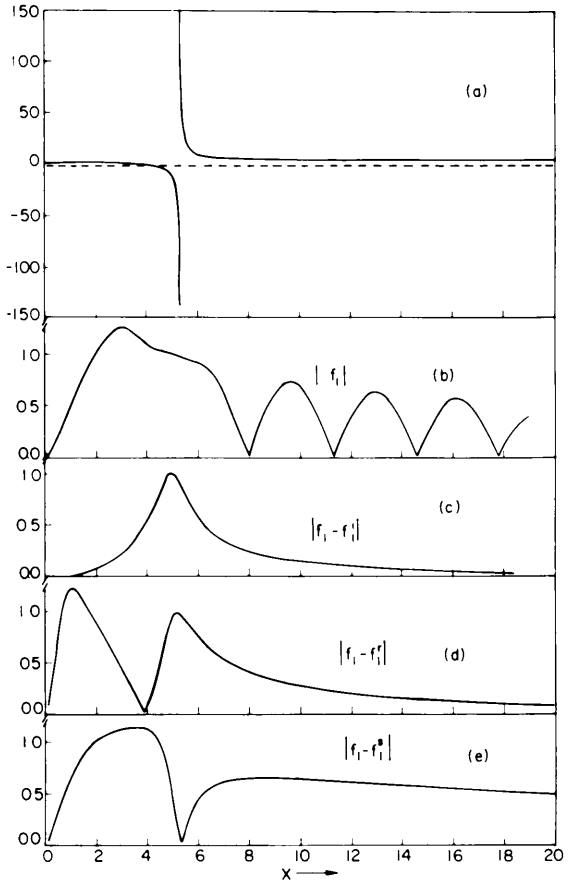


FIG. 10. Plots of (a) F_1(_____) and Re $z_n^{(1)}$(- - - - - -), (b) $|f_1(\pi)|$, (c) $|f_1(\pi) - f_1^{\text{int}}(\pi)|$, (d) $|f_1(\pi) - f_1^{\text{rig}}(\pi)|$, (e) $|f_1(\pi) - f_1^{\text{soft}}(\pi)|$ versus $x = ka$ for an air-filled cylindrical aluminum shell ($b/a = 0.9$) in water (Murphy *et al.*, 1979).

increase monotonically with mode number. The n value n_l about which we expand may be chosen such that $x_{n_l l} = x$. Equations (18) then become

$$S_n = e^{2i\epsilon_n} \sum_{l=0}^{\infty} \frac{n - \hat{n}_l^*}{n - \hat{n}_l}. \tag{38a}$$

The poles of these expressions, known as "Regge poles" (see, e.g., Newton 1966), are located in the complex-n plane at

$$n_p \equiv \hat{n}_l(x) = n_l + i\hat{\Gamma}_{nl}/2, \tag{38b}$$

where the resonances widths in n are

$$\hat{\Gamma}_{n,l} = \Gamma_{n,l}/x'_{n,l} > 0. \tag{38c}$$

If n is also considered continuous in the scattering amplitude of Eq. (1a), which at this point represents an unphysical case but will be shown to be meaningful in Section II,B,2, the principal contribution will come from these poles so that the angular function in Eq. (1a) together with the suppressed time exponential becomes

$$\exp\left[\pm i(\hat{n}_l\phi - \omega t)\right]. \tag{39}$$

This represents a circumferential (surface) wave with linear propagation constant $\mathrm{Re}\,\hat{n}_l/a$, i.e., a phase velocity $c_l(x)$ given by Eq. (34b), a wavelength

$$\lambda_l(x) = 2\pi a/n_l, \tag{40a}$$

and an attenuation (due to radiation) given by $\exp(-\phi/\phi_l)$, where

$$\phi_l = 2/\hat{\Gamma}_{n,l}. \tag{40b}$$

Indeed, at resonance, where from Eq. (38a), $n_l \to n$, Eq. (40a) shows that n wavelengths span the circumference of the cylinder.

Equation (34b) for $c_l(x)$ as a function of frequency represents the dispersion curve for the lth surface wave. The corresponding numerical values can be taken from graphs such as Fig. 7 or 8. Corresponding dispersion curves are shown in Figs. 11 and 12 (Breitenbach, 1979) for a thick aluminum shell ($b/a = 0.1$); these agree with earlier results for a solid aluminum cylinder obtained by the Watson transformation (Frisk *et al.*, 1975; Dickey *et al.*, 1976). Figure 11 presents the dispersion curves for the "Rayleigh wave" R, so designated since for an infinite cylinder, it goes over into the well-known Rayleigh wave on a flat half-space (see, e.g., Überall, 1973), and for the "transverse" (T_i) types of whispering-gallery modes; Fig. 12 corresponds to the "longitudinal" (L_i) types of whispering-gallery modes (Frisk *et al.*, 1975; Dickey *et al.*, 1976). Besides by the shape of their dispersion curves, these surface wave types can also be distinguished by their widths in Fig. 7 or 8 (a wide peak for the Rayleigh wave, narrow peaks for the whispering-gallery modes). For $x \to \infty$, the curves approach the Rayleigh speed and the transverse and longitudinal bulk speeds of aluminum, respectively.

In Fig. 8, we also notice a narrow peak corresponding to a wave we shall denote by A, which at $n = 3$ appears at $x \cong 1$ and for $n = 7$ has moved up to $x \cong 6$, but does not appear below $n = 3$ and only vaguely above $n = 7$. The dispersion curves corresponding to Fig. 8 are shown in Fig. 13, displaying the transverse modes T_1 and T_2, and demonstrating that at low

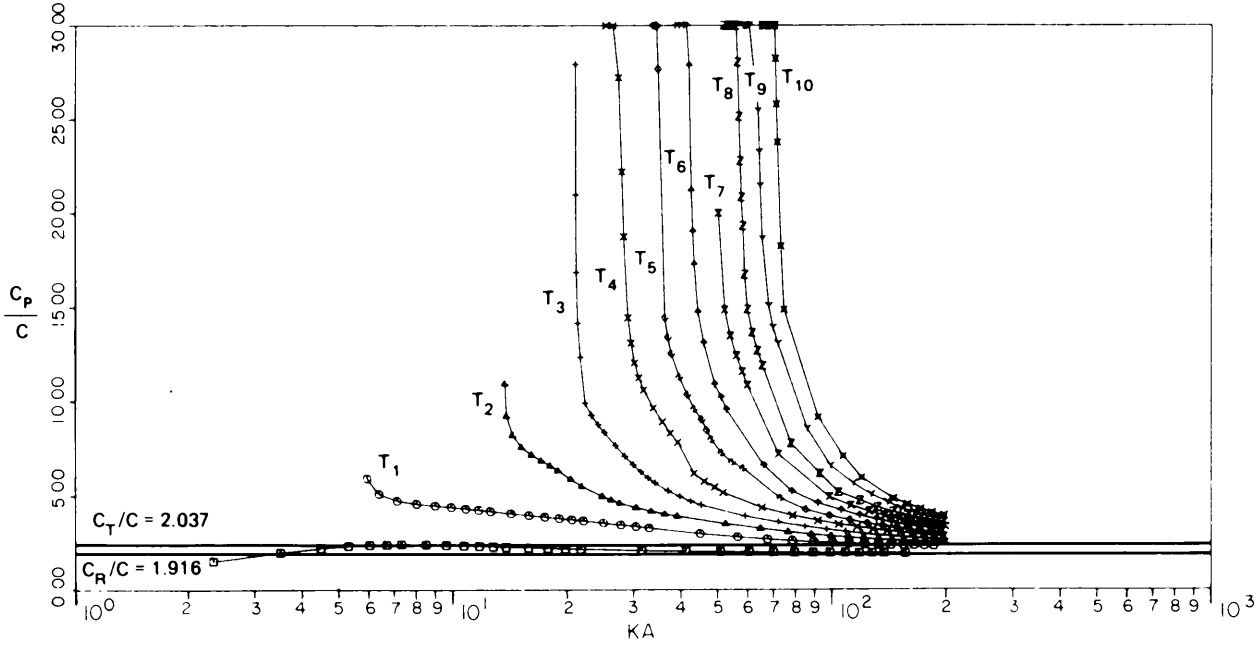


FIG. 11. Normalized phase velocity versus *ka* for Rayleigh and transverse surface waves for a cylindrical aluminum shell ($b/a = 0.1$) in water (Breitenbach, 1979).

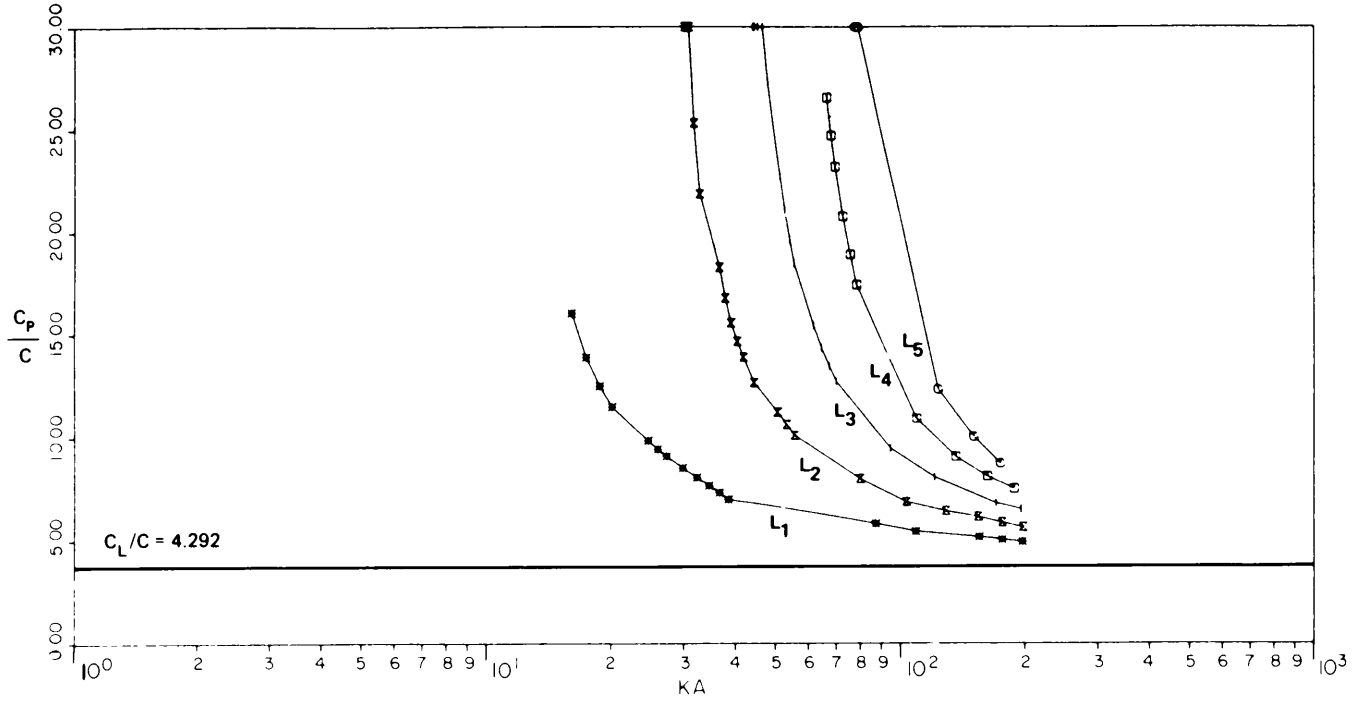


FIG. 12. Normalized phase velocity versus ka for longitudinal surface waves for a cylindrical aluminum shell ($b/a = 0.1$) in water (Breitenbach, 1979).

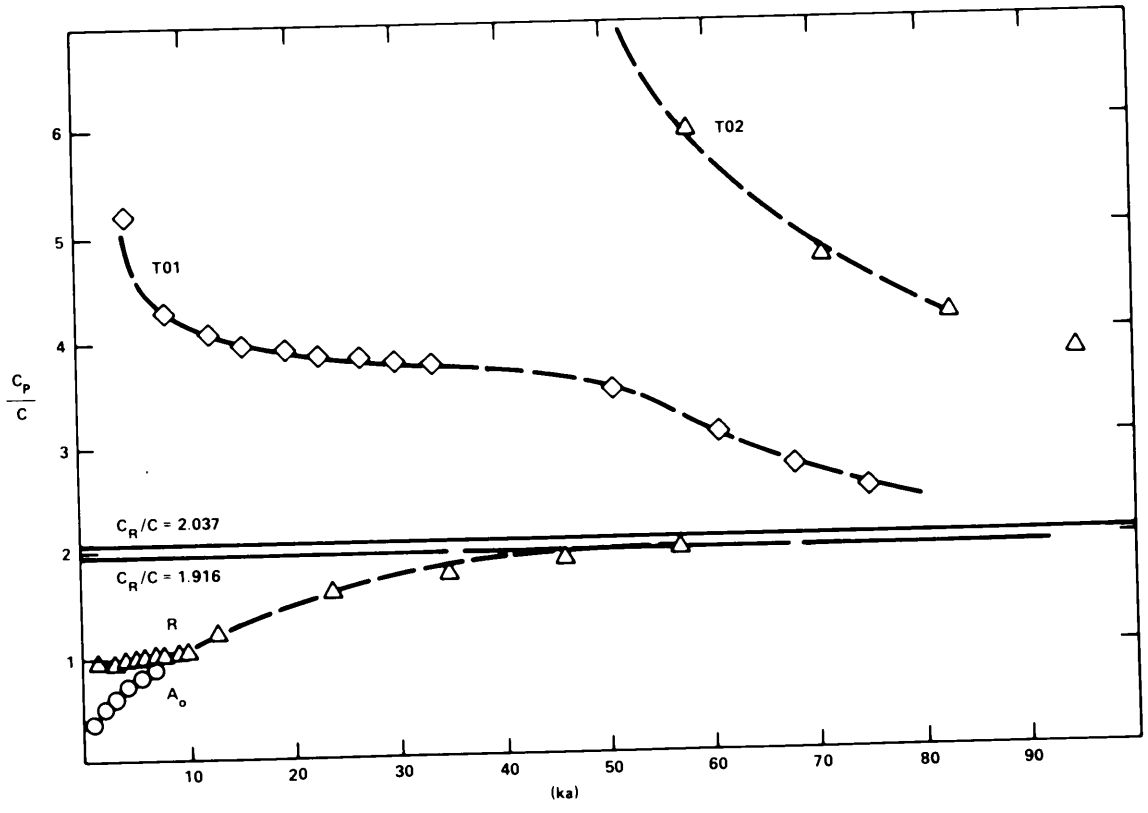


FIG. 13. Dispersion curves of the lowest surface-wave modes on an air-filled cylindrical aluminum shell ($b/a = 0.85$) in water (Breitenbach, 1979).

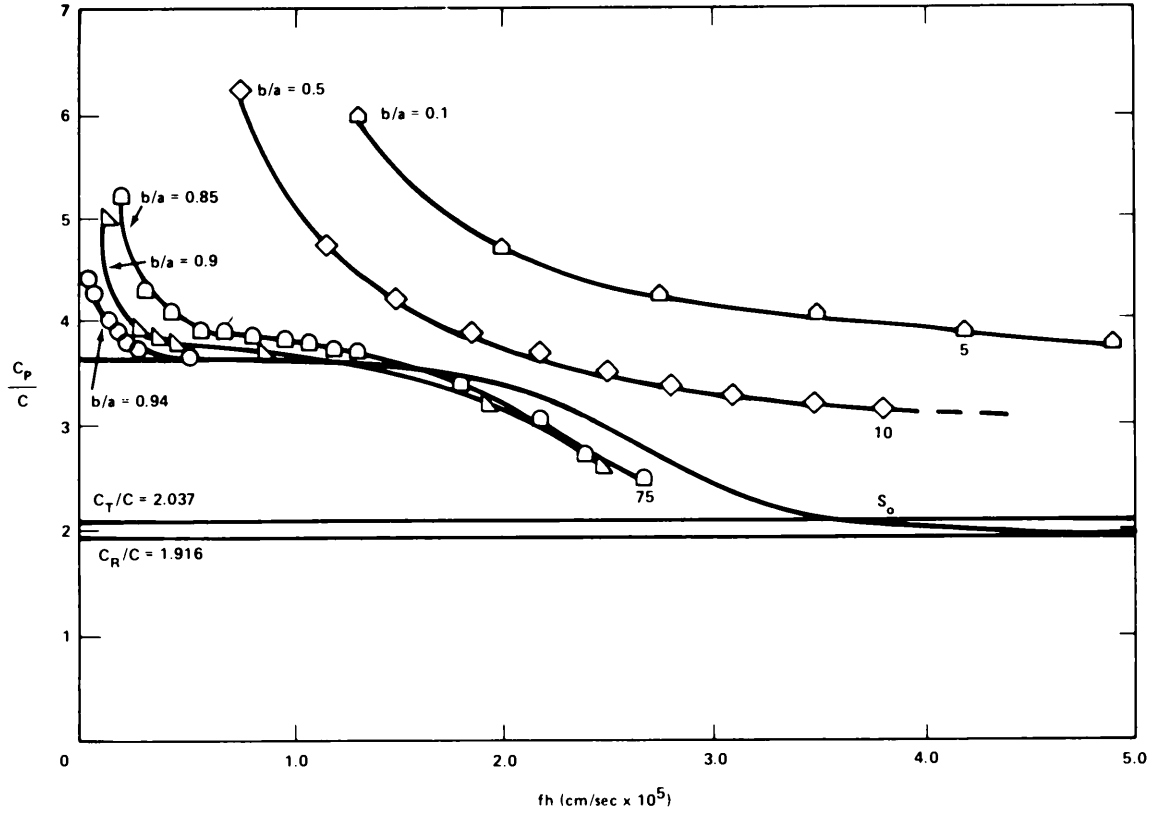


FIG. 14. Dispersion curves for the s_0 Lamb plate mode and the first transverse mode of an aluminum shell as a function of b/a. (Breitenbach, 1979).

ka values, the Rayleigh wave speed branches off from its previous curve, the latter being continued by that of the A wave in Fig. 8. This situation has been experimentally confirmed in sound scattering measurements from shells by Marshall and Olson (1970). It is analogous to the Lamb waves on a flat elastic plate (see, e.g., Überall, 1973). The dispersion curves of these waves, usually plotted versus the variable fh where f is the frequency and h is the plate thickness, related to our present variable $x \equiv ka$ by

$$fh = (c/2\pi)(1 - b/a)x \tag{41}$$

(note that the factor $1 - b/a$ for thin shells compresses the x scale and brings the widely spaced resonances, e.g., of Fig. 8 back to the left) consist of two lowest branches, a_0 and s_0, and an infinity of higher ones starting at infinity from cutoff frequencies just as in Figs. 11 and 12. Of the lowest branches, the antisymmetric (flexural) mode a_0 has a dispersion curve like that of R–A in Fig. 13, and may be identified with this (combined) mode. The symmetric (compressional) Lamb mode s_0 on a plate is shown in Fig. 14, from which it may also be seen that the lowest transverse mode T_1 on the shell approaches a_0 in the limit $b/a \to 1$.

Regarding the A mode, we observed above that it is not excited by sound scattering from the cylindrical shell in all partial waves (or, equivalently, at all frequencies) while the Rayleigh and whispering-gallery modes are. This fact has also been observed experimentally (Likhodaeva and Shenderov, 1971), and noted theoretically (Nigul, 1974).

2. *Spherical Shells; Interpretation in Terms of Surface Waves*

The resonance theory for spherical shells (Murphy *et al.*, 1979) is similar to the general resonance theory discussed in the preceding. As in the case of the cylinder, it is obtained from the theory of the solid sphere, Section II, A, 3, by replacing the mechanical impedance F_n mentioned after Eqs. (31) by one appropriate for a spherical shell, given as a ratio of two 5×5 determinants. As an example of this calculation, we show in Fig. 15 for an air-filled spherical aluminum shell ($b/a = 0.7$) in water the quantities F_1^{-1} and $\mathrm{Re}\, z_1^{(1)-1}$ plotted versus $ka \leqslant 100$. At the intersections of these curves, the amplitude $|f_1 - f_1^i|$ (intermediate background subtracted) exhibits the resonances.

It was mentioned in previous sections that the resonances can be interpreted in terms of surface waves on the scattering object. A detailed proof of this will now be given for the case of a spherical object (Murphy *et al.*, 1979; Gaunaurd and Überall, 1979; Überall *et al.*, 1979). The scattered pressure of Eq. (30) may, by the use of the Watson transformation (see, e.g., Überall, 1973)

$$\sum_{n=0}^{\infty} f_n P_n(\cos\theta) = \frac{1}{2i} \oint_C \frac{P_{\lambda-1/2}(-\cos\theta) f_{\lambda-1/2}\, d\lambda}{\cos \pi\lambda}, \tag{42}$$

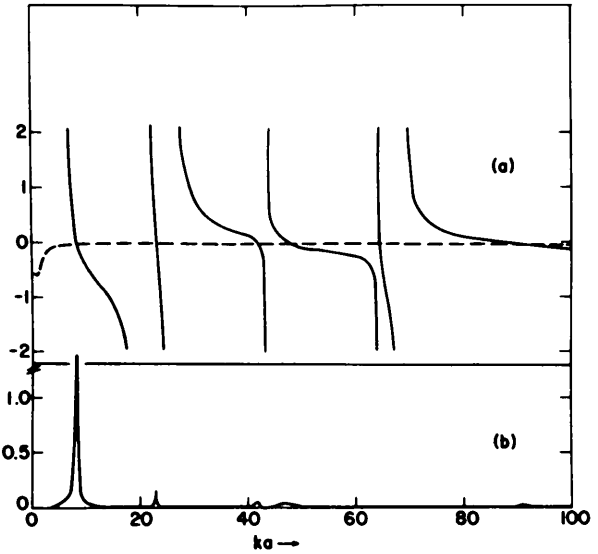


FIG. 15. Plots of (a) F_1^{-1}(_____) and $\mathrm{Re}(z_1^{(1)-1})$(- - - - - - -), and (b) $|f_1(\pi) - f_1^{\mathrm{int}}(\pi)|$ for an air-filled spherical aluminum shell in water ($b/a = 0.7$) (Murphy *et al.*, 1979a).

where the contour C encloses clockwise the positive real axis of the λ plane, be transformed into a contour integral which, in turn, may be reevaluated at the Regge poles $\lambda = \hat{n}_l + \frac{1}{2}$, Eq. (38b), of the appropriate function $f_{\lambda-1/2}$, which according to Eq. (38a) we may write in the form

$$f_{\lambda-1/2} = \sum_{l=0}^{\infty} \frac{f_{\lambda-1/2}^{(l)}}{\lambda - \frac{1}{2} - \hat{n}_l} \tag{43a}$$

in order to exhibit the resonance denominator. This leads to the exact expression

$$p_{\mathrm{sc}} = \sum_{l=0}^{\infty} f_{\hat{n}_l}^{(l)} \frac{\pi P_{\hat{n}_l}(-\cos\theta)}{\cos\left[\pi\left(\hat{n}_l + \frac{1}{2}\right)\right]} \tag{43b}$$

with

$$f_{\hat{n}_l}^{(l)} = i\left(\hat{n}_l + \tfrac{1}{2}\right) e^{i\pi\hat{n}_l} h_{\hat{n}_l}^{(1)}(kr)\hat{\Gamma}_{n_l l} S_{\hat{n}_l}^{(0)}, \tag{43c}$$

$S_n^{(0)}$ standing for S_n^{r} or S_n^{s} in Eqs. (31) as applicable. Equation (43b) has the same poles and residues in the variable $\hat{n}_l$ at all the integers n as does Eq. (30) using Eq. (38a), and we see that our use of the Watson transformation has furnished generalized resonance denominators which interpolate between the resonance positions of Eq. (38a), and which will provide us with a more rigorous circumferential-wave picture.

This is obtained by using the asymptotic form (see, e.g., Newton, 1966) valid for $\nu \sin\theta \gg 1$:

$$P_\nu(-\cos\theta) \cong \left(\frac{2}{\pi\left(\nu+\frac{1}{2}\right)\sin\theta}\right)^{1/2} \cos\left[\left(\nu+\tfrac{1}{2}\right)(\pi-\theta)-\tfrac{1}{4}\pi\right]; \quad (44a)$$

even for small values of ν, this provides an approximate description of the behavior of the amplitude. One may then split the cosine in Eq. (44a) into two exponentials, and also expand the denominator of Eq. (43b) as

$$\frac{1}{\cos\left[\pi\left(\hat{n}_l+\frac{1}{2}\right)\right]} = 2i\sum_{m=0}^{\infty} \exp\left[i\pi(2m+1)\hat{n}_l\right], \quad (44b)$$

so that

$$p_{sc} \cong i\pi \sum_{l=0}^{\infty} \left(\frac{2}{\pi\left(\hat{n}_l+\frac{1}{2}\right)\sin\theta}\right)^{1/2} f_{\hat{n}_l}^{(l)} \sum_{\varepsilon=\pm 1} \sum_{m=0}^{\infty} \exp\left[-i\varepsilon\pi/4 - i(m-\hat{n}_l)\pi\right]$$

$$\times \exp\left\{i\left(\hat{n}_l+\tfrac{1}{2}\right)\left[\varepsilon(\pi-\theta)+2m\pi\right]\right\}. \quad (45)$$

This expression shows that for $m=0$ and a given value of l, two surface waves with angular propagation constant $\hat{n}_l+\frac{1}{2}$ propagate opposite to each other around a meridian of the spherical shell, with an amplitude factor $\sin^{-1/2}\theta$ that describes their convergence at the north and south pole of the sphere. These "tidal waves" are joined by other such waves with $m>0$ that already have encircled the shell m times previously (of course with larger and larger attenuation, which is furnished by $\operatorname{Im}\hat{n}_l \neq 0$), corresponding to steady-state scattering. Wavelength, attenuation, and phase velocity of these waves are now given by

$$\lambda_l(x) = 2\pi a/\left(n_l+\tfrac{1}{2}\right), \quad (46a)$$

$$\theta_l(x) = 2/\hat{\Gamma}_{n_l l}, \quad (46b)$$

$$c_l(x) = xc/\left(n_l+\tfrac{1}{2}\right), \quad (46c)$$

which differ from Eqs. (34b) and (40a) for the cylinder by having n_l replaced by $n_l+\frac{1}{2}$. This does *not* mean, however, that at resonance ($n_l \to n$) an integer-plus-half wavelengths span the sphere, since inspection of Eq. (45) shows that here, the surface waves suffer a phase loss of $\pi/2$ (a quarter-wavelength) at each of the convergence points. Phase matching is therefore again seen to take place at resonance in the course of repeated circumnavigations of the surface waves.

3. *Shells with Absorbing Coating*

Echoes reflected monostatically and bistatically at normal incidence from acoustically coated infinite elastic air-filled cylindrical shells have been studied by Gaunaurd (1977), Flax and Neubauer (1977), Gaunaurd and Überall (1979a), and Gaunaurd (1977a). The Navier equations of elasticity for the displacement fields were assumed to hold in the shell, and their viscoelastic counterparts in the Kelvin–Voigt model of viscoelasticity (Gaunaurd, 1976) were assumed to hold in the sound-absorbing coating. It was shown (Gaunaurd, 1976) that the equations of viscoelasticity are satisfied if three independent scalar Helmholtz equations with complex propagation constants hold for three independent scalar potentials. All the stress and displacement components can be obtained from these three potentials (Gaunaurd, 1976) once ten boundary conditions are satisfied at the three interfaces of the coated shell, as described elsewhere (Gaunaurd, 1977). This leads to form function expressions similar to Eqs. (5) and (8b), and to S functions as in Eqs. (10), except that F_n is now a ratio of two 9×9 determinants containing cylinder functions. The monostatic sonar cross section, defined as

$$\sigma = \pi a |f_\infty(\pi)|^2, \tag{47}$$

is plotted versus x in Fig. 16. These cross-sectional plots are computed for a 21-in. overall outer diameter coated shell having a $\frac{1}{8}$-in. coating around it. Successive plots in the graph are for successively thinner shells. The three curves shown for each shell thickness correspond to three increasingly higher levels of viscosity in the coating. The rapid oscillations observable in the cross-sectional plots are due to the shell resonances, and it is seen that these are damped out as the coating viscosity increases. A high-frequency analysis of this coated shell based on the Watson–Sommerfeld transform has also been carried out (Gaunaurd, 1977a), showing that a one-term Watson–Sommerfeld solution was sufficiently accurate as compared with the normal-mode solution, retaining many terms in the normal-mode series.

The resonance theory was used recently (Gaunaurd and Überall, 1979a) to analyze the problem of a rubber-coated steel shell, again by subtracting a background to isolate the pure resonances contained in the successive normal mode contributions to the scattering amplitude. It was established that an increase in coating viscosity resulted in a decrease of the height of the modal resonances. A quantitative study was performed to indicate how much coating viscosity was needed in order to reduce the resonances to certain prescribed levels. We shall now present a calculation of the resonances in the modes of a rubber-coated, air-filled aluminum shell submerged in water. Figure 17 shows the partial-wave amplitudes

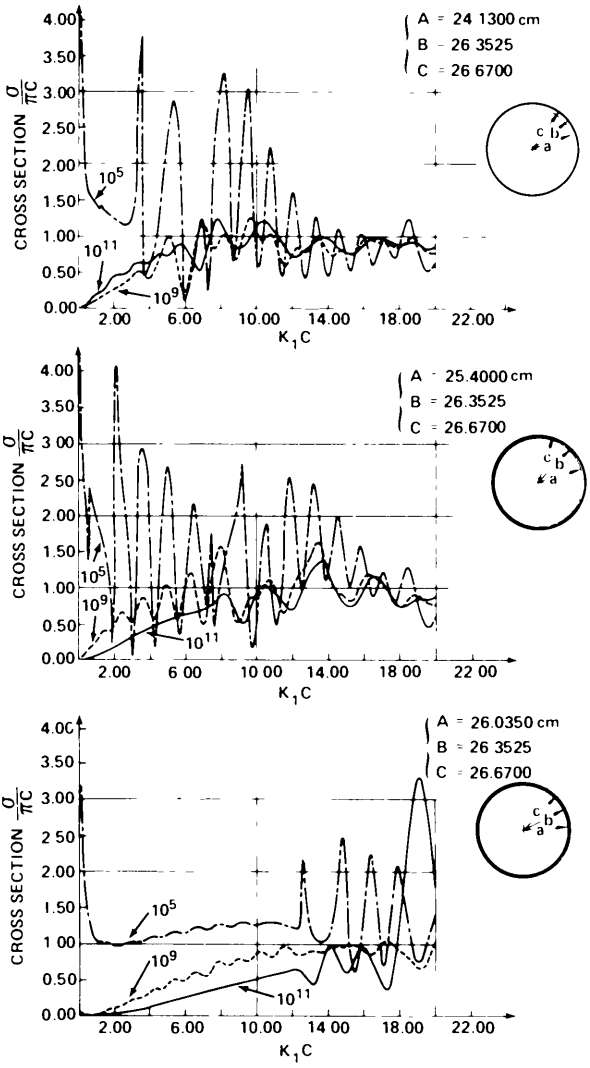


FIG. 16. Sonar cross section $\sigma/\pi c$ versus wave size $k_1 c$ for a coated steel shell in water. Shell thickness decreases from top to bottom graph. Coating thickness and overall outer coating diameter remain fixed. Increasing viscosity levels (i.e., $10^5, 10^9, 10^{11}$ dyn/cm^2) are seen to damp out resonances present in the summed cross sections (Gaunaurd, 1977).

($n = 0$, 1, 2, 3 from top) for a shell of outer radius $c = 26.59$ cm and thickness $b - a = 7.98$ cm, coated by a thin rubber layer of thickness $c - b = 0.08$ cm. The left-hand column shows the moduli of the partial-wave amplitudes, the right-hand column the resonances after subtraction of a suitable nonresonant background. These resonances are not much different from those of an uncoated aluminum shell studied earlier (Murphy *et al.*, 1978), but extra spikes at $x \equiv kc \cong 33$ for $n = 1$, 2, and 3 are now present, which can be attributed to the coating.

If the coating is inhomogeneous and contains perforations, there appear extra resonances caused by them. These have been analyzed earlier (Gaunaurd, 1977b) for flat plates coated with perforated rubber layers, the perforations having the shape of short cylindrical cavities. The treatment of coating layers containing inclusions or fluid-filled cavities is still a problem subject to investigation.

C. Experiments

Experiments for verifying the surface-wave interpretation of scattering resonances have been carried out by Dragonette (1978, 1979). These consisted in the experimental observation of the backscattered reradiation of a circumferentially traveling Rayleigh wave on a submerged aluminum cylinder, constituting the first observation of this phenomenon. A second "computer experiment" consisted in the calculation of a low-*ka* echo response of the aluminum cylinder to an incident short acoustic pulse.

1. *Tank Experiment*

A sound-scattering experiment was carried out on an aluminum cylinder submerged in a water tank. It was necessary that the apparatus satisfy several conditions in order that the measurements correspond to the assumptions of the theory. For example, the incident wave had to be a plane sound wave. Furthermore, the measurements had to be performed in the far field so that the asymptotic expressions for the Hankel functions were valid.

Figure 3 shows that the Rayleigh wave manifests itself almost exclusively via the three nulls labeled (2, 1), (3, 1), and (4, 1), which are all in the low-*ka* region. One of the major difficulties in obtaining an experimental result for the Rayleigh wave in that region was in finding the correct combination of cylinder radius, frequency, and pulse length such that the specular reflection from the cylinder's apex would be separated from the Rayleigh wave. The best results were obtained with an aluminum cylinder of 0.625-cm radius, using a short pulse centered at a frequency of $f_0 = 386$ kHz (corresponding to $k_0 a = 10.4$), and the pulse length was 5 cycles.

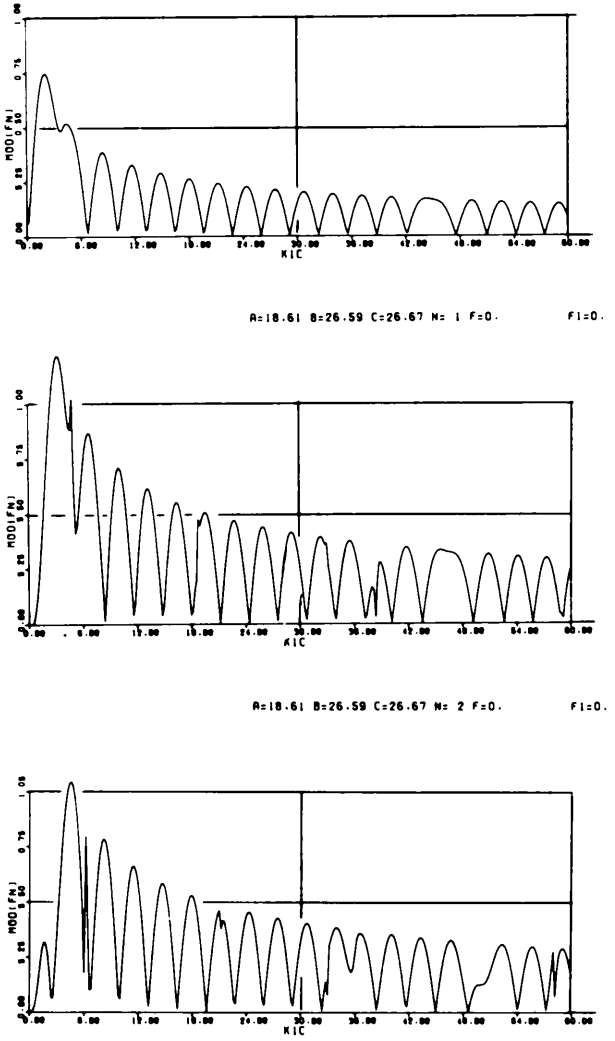


FIG. 17. Plot of scattering amplitudes $n = 0-2$ (from top) for an air-filled cylindrical aluminum shell, coated with rubber, immersed in water. Left-hand column: modulus of total amplitudes; right-hand column: resonances of shell and coating, after subtraction of background (Gaunaurd and Überall, 1979a).

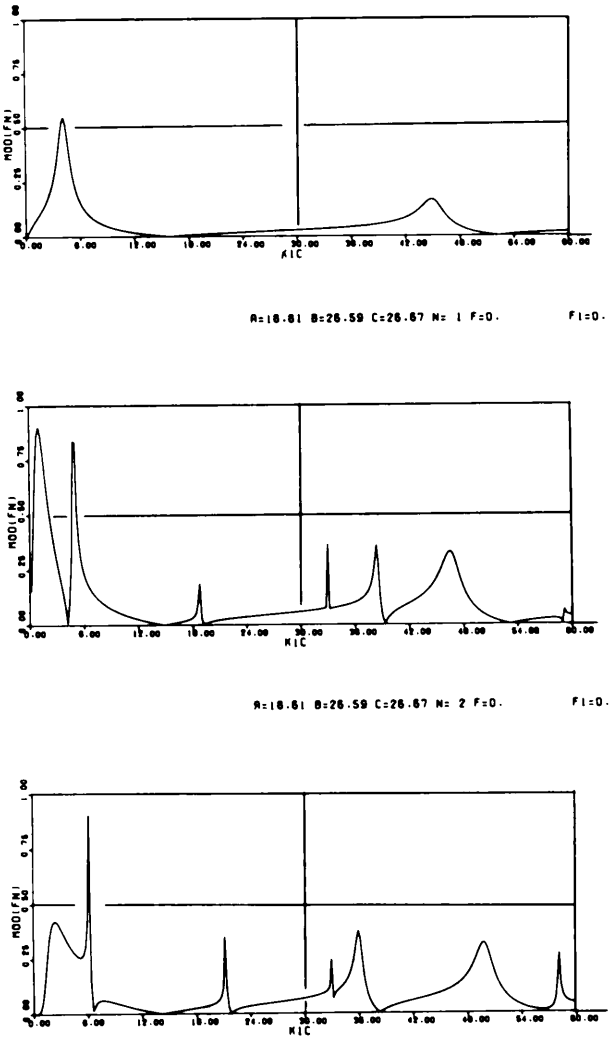


FIG. 17 *Continued.*

The result of the experimental measurement is presented in Fig. 18, showing that the backscattering echo consists of a specular return followed by a Rayleigh circumferential wave return that is 180° out of phase with the specular return. This is followed by a return from the second circumnavigation of the Rayleigh wave.

The slight change in pulse shape is due to the fact that the phase velocity of the Rayleigh wave is a function of frequency, i.e., it is slightly

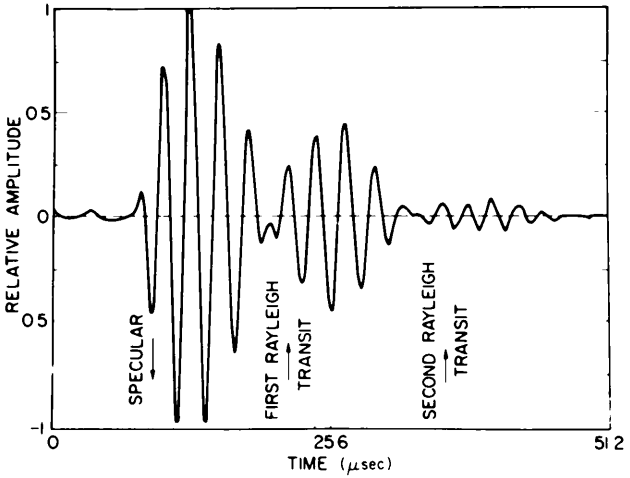


FIG. 18. Experimental observation of the Rayleigh circumferential wave on an aluminum cylinder at $k_0a = 10.4$ (Dragonette, 1979).

dispersive. The pulse shape is sufficiently preserved, however, to be able to establish the 180° phase change. The latter was predicted, e.g., for the Rayleigh wave on flat surfaces (Bertoni and Tamir, 1973).

2. *Computer Experiment*

A pressure pulse $p(\tau)$ contains the frequencies expressed by a spectrum $g(ka)$ in ka space. The Fourier theorem expresses these as

$$p(\tau) = \int_{-\infty}^{\infty} g(ka) e^{ika\tau} d(ka), \tag{48a}$$

with an inverse transform

$$g(ka) = \frac{1}{2\pi} \int_{-\infty}^{\infty} p(\tau) e^{-ika\tau} d\tau. \tag{48b}$$

Here $p(\tau)$ is the instantaneous pressure and τ the dimensionless time parameter $\tau = (ct - r)/a$, which is normalized to zero when the incident pulse coincides with the center of the cylinder. The backscattering form function modulus, Eq. (4), with argument $\phi = \pi$ suppressed, can now be obtained from

$$|f_\infty(ka)| = \left(\frac{2r}{a}\right)^{1/2} \left| \int_{-\infty}^{\infty} p_{sc}(\tau) e^{-ika\tau} d\tau \Big/ \int_{-\infty}^{\infty} p_{inc}(\tau) e^{-ka\tau} d\tau \right|. \tag{49}$$

This equation is the basis upon which a steady-state quantity $|f_\infty(ka)|$ can be obtained over a broad frequency range by a single short-pulse experi-

ment. The incident and reflected pulses must here be digitized, and their Fourier transforms computed and inserted into this equation (Dragonette *et al.*, 1974; Dragonette, 1978).

A computation of the response of an elastic cylinder to an incident pulse will also allow an examination of the properties of circumferential waves. A theoretical one-cycle pulse, which is not readily attainable in the laboratory with normally available transducers, has been programmed as the incident pulse. The computation of the reflection of such a pulse, centered at $k_0a = 8.8$, is obtained from

$$p_{sc}(\tau) = \left(\frac{a}{2r}\right)^{1/2} p_0 \int_{-\infty}^{\infty} e^{ika(r/a)} f_\infty(ka) g(ka) e^{ika\tau} d(ka) \qquad (50)$$

[see Eq. (4)] is given in Fig. 19 (Überall *et al*, 1977). Once again, the specular and the Rayleigh wave echoes shown in this figure are 180° out of phase.

The phase velocities of circumferential waves on a cylinder are, from Eq. (34a),

$$c_n(ka) = c(ka)_{res}/n, \qquad (51a)$$

and the group velocities

$$c_n^g(ka) = cd(ka)_{res}/dn. \qquad (51b)$$

The group velocity obtained from Fig. 19 is $c^g/c = 2.4$, which compares well with $c^g/c = 2.3$ as estimated by Doolittle *et al.* (1968).

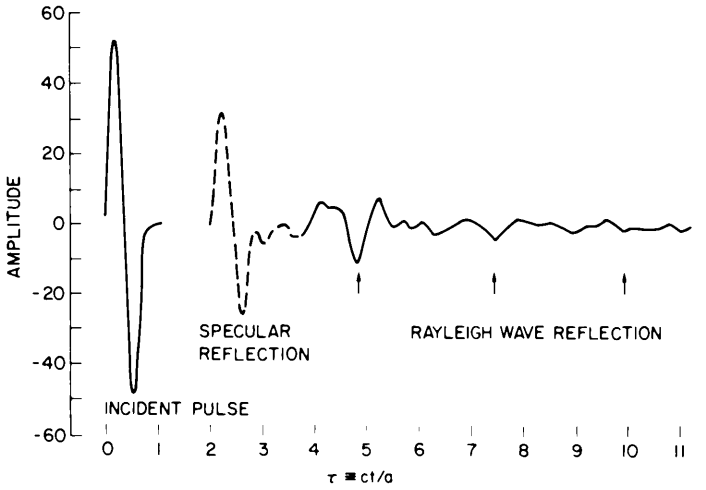


FIG. 19. Computation of the echoes reflected from an aluminum cylinder at $k_0a = 8.8$ (Überall *et al.*, 1977.)

D. Gas Bubbles in Fluids

While numerous investigations of the radiation from pulsating gas-filled bubbles in fluids (especially air bubbles in water) have been reported (Minnaert, 1933; Strasberg, 1953, 1956; Devin, 1959), only one comprehensive study of sound scattering from bubbles appears in the literature (Nishi, 1975). This example represents the simplest kind of scattering problem, but in principle it exhibits all the basic features of the surface-wave resonances. A bubble differs from an elastic scattering target principally in that the acoustic impedance ρc is much less than that of the surrounding medium [e.g., for an air bubble in water, $(\rho c)_{\text{air}}/(\rho c)_{\text{water}} = 5 \times 10^3$]. As a consequence, the bubble exhibits a giant fundamental monopole resonance at a very low frequency (at resonance, the scattering cross section of a typical bubble in sea water is of the order 10^3 times its geometrical cross section). This resonance is well known (Medwin, 1975, 1977; Clay and Medwin, 1977), and for small bubbles (radius less than about 100 μm), it is subject to many corrections, such as the effects of surface tension, viscosity, and thermal conductivity.

In recent papers (Sage *et al.*, 1979; Überall *et al.*, 1979; Gaunaurd, 1979; Gaunaurd *et al.*, 1979; Gaunaurd and Überall, 1980) the resonance theory has been applied to sound scattering from bubbles. The mentioned corrections were here not taken into account so that the theory applies to large bubbles only. However, a complete study was performed not only of the fundamental monopole resonance, but also of the fundamental higher-multipole resonances of which only scant knowledge existed (Strasberg, 1956), and of the overtones of all these resonances, of which no previous knowledge existed.

The scalar potential representing an incident plane wave in the liquid is written

$$\phi_{\text{inc}} = \phi_0 e^{-i\omega t} \sum_{n=0}^{\infty} (2n+1) i^n j_n(kr) P_n(\cos\theta), \tag{52a}$$

where k is the wave number in the liquid and a the bubble radius. The scattered wave is

$$\phi_{\text{sc}} = \phi_0 e^{-i\omega t} \sum_{n=0}^{\infty} (2n+1) i^n A_n h_n^{(1)}(kr) P_n(\cos\theta). \tag{52b}$$

As in Section II, A, 3, we introduce the S function

$$S_n = 1 + 2A_n, \tag{53}$$

for which the boundary conditions at $r = a$ (continuity of pressure, and of the normal component of the particle velocity) give

$$S_n = S_n^s\left[\left(z_n^{(2)} - F_n\right)/\left(z_n^{(1)} - F_n\right)\right], \tag{54a}$$

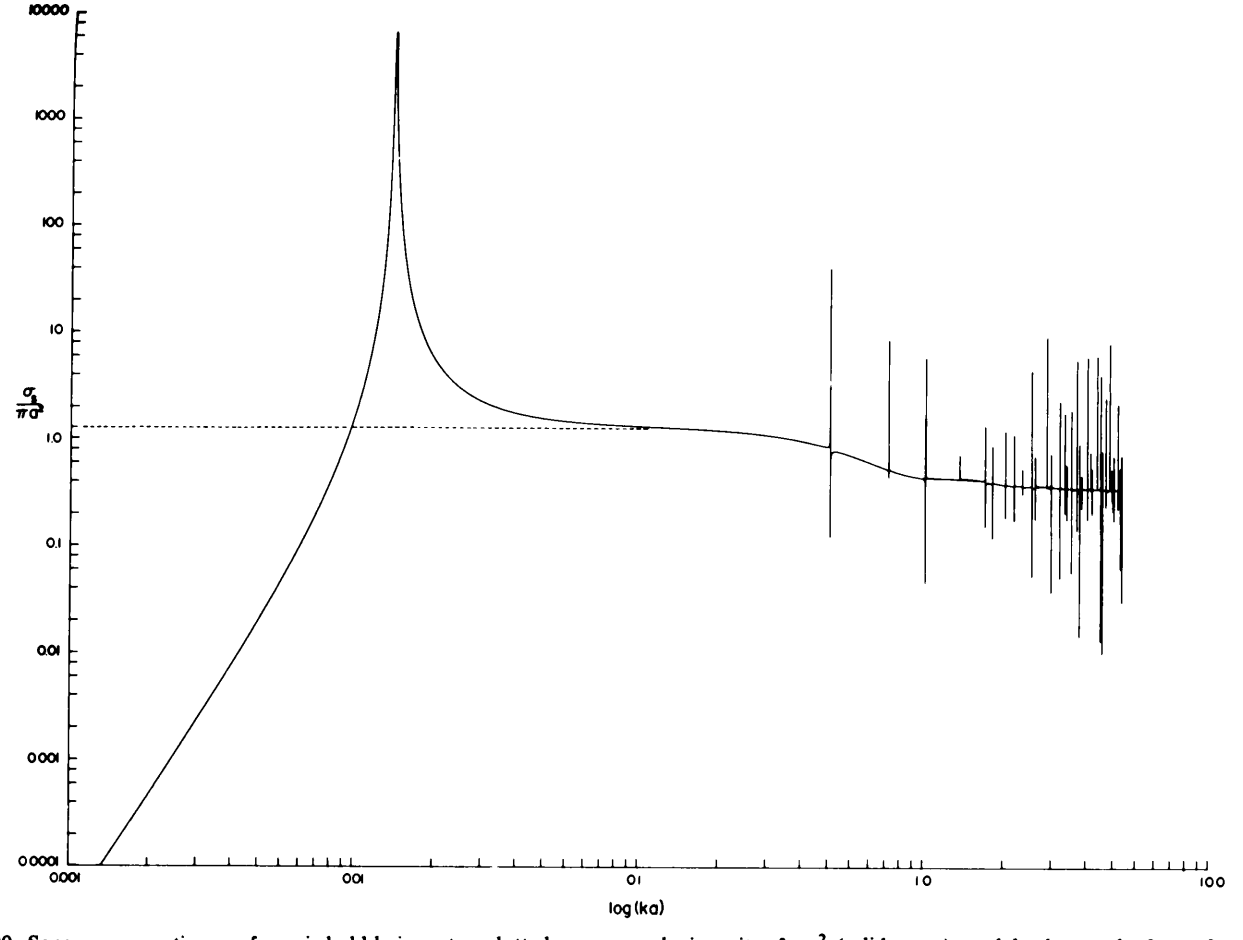


FIG. 20. Sonar cross section σ_s of an air bubble in water, plotted versus $x = ka$ in units of πa^2 (solid curve), and background of a soft sphere (dashed curve) (Sage *et al*, 1979).

where

$$z_n^{(i)} = xh_n^{(i)}(x)/h_n^{(i)}(x), \qquad i = 1, 2, \tag{54b}$$

$$F_n = (\rho_w/\rho_b)x_b j_n'(x_b)/j_n(x_b) \tag{54c}$$

with ρ_b, ρ_w the density of the bubble gas and the surrounding fluid, respectively, $x = ka$, $x_b = k_b a$ where k_b is the wave number in the bubble gas, and where

$$S_n^s = -h_n^{(2)}(x)/h_n^{(1)}(x) \tag{54d}$$

is the S function for scattering from a soft (pressure release) sphere (which is here for appropriate background). The resonance theory then proceeds as before. As to numerical results, Fig. 20 presents the resonance spectrum of an air bubble in water as it appears in the sonar (backscattering) cross section

$$\sigma_s = 4\pi \left| \sum_{n=0}^{\infty} f_n^{\text{sphere}}(\pi) \right|^2, \tag{55}$$

plotted versus log ka so that the low-frequency monopole resonance is clearly exhibited. The normalization πa^2 is the geometrical bubble cross section, and the dotted line is the soft-sphere cross section (Sage *et al.*, 1979).

In a further study (Überall *et al.*, 1979), the surface motions of the bubble were resolved into circumferential waves as in Eq. (45), and the

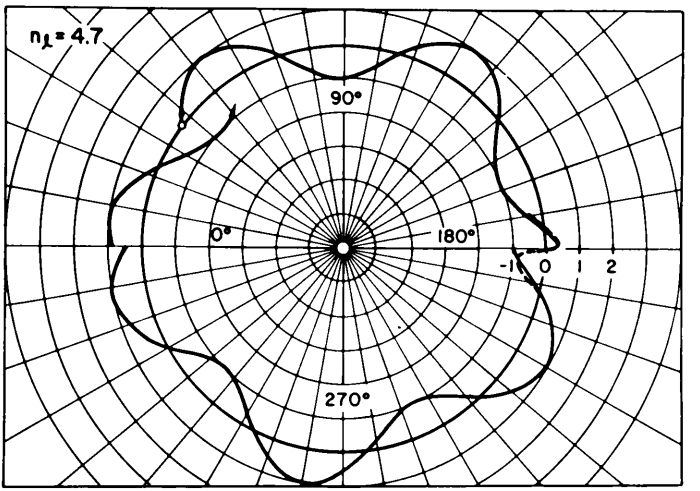


FIG. 21. Circumferential wave on a gas bubble in water off resonance, showing absence of phase matching, and $\pi/2$ phase lag at the convergence points (Überall *et al.*, 1979).

corresponding wave with $n = 4.7$ (i.e., somewhat off the $n_l = 5$ resonance) is shown in Fig. 21. The $\pi/2$ phase jump at the convergence point $\theta = 180°$ is seen to be abrupt if the asymptotic form Eq. (44a) is used (solid line), and smoothed out if the exact expression for P_ν is used (dashed line). In the same study, the motion of the gas on and off resonance was studied. It was found that the surface motions of the bubble do not exhibit very large amplitudes at resonance, but rather that the resonance is characterized by resonant amplitudes of the order $\rho_w/\rho_b \sim 10^3$ in the radial motion of air particles inside the bubble (but not near the surface, except for the fundamental monopole), and in the angular motion throughout the bubble interior right up to its surface (where the angular air motion slips against the nonresonant motion of the water particles).

As the simplest scattering system, the gas bubble has been made the object of a refinement of resonances theory (Gaunaurd and Überall, 1981), which in nuclear physics is known as the R-matrix theory (see, e.g., Lane and Thomas 1958), developed by Wigner (1946) and by Wigner and Eisenbud (1947). In this approach, it is noted that the mechanical impedance $F_n(x)$ and its inverse are meromorphic functions of x (as exemplified, e.g., in Figs. 10 and 15), with poles on the real-x axis if the scatterer is nonabsorbing, or in the complex-x plane if it is. For the simple case of Eq. (54c), the meromorphic representation of F_n in terms of a Mittag–Leffler series can easily be determined as

$$\frac{x}{F_n(x)} \equiv \mathcal{R}_n(x) = \sum_{l=-\infty}^{\infty}{}'' \frac{\gamma_{nl}^2}{x'_{nl} - x} \tag{56a}$$

where $\mathcal{R}_n(x)$ is a Wigner-type R function, and where

$$x'_{nl} = z'_{nl} c_b / c, \tag{56b}$$

$$\gamma_{nl}^2 = -\left(\rho_b c_b^2/\rho_w c^2\right) j_n(z'_{nl})/j_n''(z'_{nl}) \tag{56c}$$

(γ_{nl} being called the "reduced widths" in nuclear physics). Here, z'_{nl} are the (real) zeros of $j'_n(z)$, labeled by the integers $l = 0, \pm 1, \pm 2, \ldots$; the double prime on the sum indicates that the first monopole ($n = 0$) zero that is located at the origin, i.e., $z'_{00} \equiv 0$, is to be included in the sum but the zeros at the origin for $n \geqslant 2$, i.e., $z'_{n0} \equiv 0$, are *not*. Substituting in Eq. (54a) gives an *exact* resonance expression for S_n,

$$S_n = S_n^s \frac{1 - (\sigma_n - i\pi_n) \sum'' \gamma_{nl}^2/(x'_{nl} - x)}{1 - (\sigma_n + i\pi_n) \sum'' \gamma_{nl}^2/(x'_{nl} - x)}, \tag{56d}$$

where we denoted

$$z_n^{(i)}(x)/x = \sigma_n(x) \pm i\pi_n(x), \qquad i = 1, 2, \tag{56e}$$

which is the form of the S function given by R-matrix theory. This expression of course does not directly equal our previous (approximate) resonance formula

$$S_n \cong \sum_l S_n^s \frac{x - x_{nl} - \frac{1}{2}i\Gamma_{nl}}{x - x_{nl} + \frac{1}{2}i\Gamma_{nl}}, \tag{57}$$

i.e., the sphere counterpart of Eq. (18b). One may, however, reduce Eq. (56d) to Eq. (57) by the so-called "one-level approximation," which holds if the resonances are narrow and well separated, i.e., nonoverlapping. [This assumption, incidentally, already went into Eq. (57), in which the sum over l is a symbolic sum, the expression being intended to hold only for one term at a time in the vicinity of each corresponding resonance frequency]. Dropping the sum makes Eq. (56d) indeed reduce to Eq. (57) without the sum, where now

$$x_{nl} = x'_{nl} + \Delta_{nl}. \tag{58a}$$

Here

$$\Delta_{nl} = -\sigma_n(x'_{nl})\gamma_{nl}^2 \tag{58b}$$

is the "level shift" between the (real) pole positions x_n of the $\mathfrak{R}_n$ and the resonance frequencies x_n; furthermore, the resonance widths Γ_{nl} in Eq. (57) are given by

$$\Gamma_{nl} = 2\pi_n(x'_{nl})\gamma_{nk}^2. \tag{58c}$$

Accordingly, σ_n in nuclear physics is called the shift function, while π_n is known as the penetration factor.

As mentioned, for the absorbing-target case, complex Mittag–Leffler poles will lead to resonance widths already in Eq. (56a), and the one-level approximation then just increases their widths via the term of Eq. (58c). Altogether, it is seen that resonance formulas such as Eqs. (18) or (57) are correct if the resonances are nonoverlapping, but the R-matrix expression of Eq. (56d) always gives the correct S function and is applicable even for overlapping resonances.

E. Submerged Plates

The preceding sections have concerned resonances in the scattering from smooth curved objects, caused by the phase matching of repeatedly circumnavigating surface waves. Resonances, however, also appear in the transmission and reflection of sound by submerged plates. These are well known to acoustic engineers, and are caused (for fluid plates) by the phase matching of back-and-forth reflected internal waves. While sound interac-

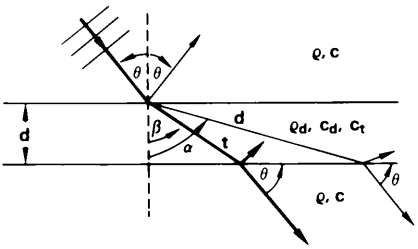


FIG. 22. Sound reflection and transmission for an elastic plate submerged in a fluid.

tion with plates is a well-studied subject (Osborne and Hart, 1945; Schoch, 1952), the resonances that appear in the transmission and reflection coefficients have only recently been the subject of detailed investigation, both for a fluid layer in a fluid (Fiorito and Überall, 1979), and for a submerged elastic plate (Fiorito *et al.*, 1979). Similar procedures as in the preceding have been used here in order to bring the reflected and transmitted waves into an explicit resonance form.

We consider an infinite plane parallel elastic plate of thickness d, density ρ_p, dilatational wave speed c_d, and shear wave speed c_t, imbedded in a fluid of density ρ and sound speed c. A plane sound wave of frequency f is incident at an angle θ, creating dilatational and shear waves which are refracted into the plate at angles α and β, respectively (Fig. 22). The angles are measured with respect to the plate normal. A transmitted wave emerges on the opposite side of the plate at an angle θ. Expressions for the transmission (T) and reflection (R) coefficients of the sound wave can be obtained from the work of Schoch (1952). Here, we write his equations in a novel form (Fiorito *et al.*, 1979), which easily allows a decomposition of T and R into resonance terms, as follows:

$$T = i\tau\left\{\frac{1}{C_s - i\tau} + \frac{1}{C_a + i\tau}\right\}, \tag{59a}$$

$$R = \frac{C_s C_a - \tau^2}{C_s + C_a}\left\{\frac{1}{C_s - i\tau} + \frac{1}{C_a + i\tau}\right\}, \tag{59b}$$

where

$$\tau = \frac{\rho c/\cos\theta}{\rho_p c_d/\cos\alpha} = \frac{\rho}{\rho_p}\frac{(n_d^2 - \sin^2\theta)^{1/2}}{\cos\theta}, \tag{60}$$

$$\left\{\begin{matrix} C_s \\ C_d \end{matrix}\right\} = \cos^2 2\beta\left\{\begin{matrix} \cot\delta \\ \tan\delta \end{matrix}\right\} + r^2 \sin 2\alpha \sin 2\beta\left\{\begin{matrix} \cot\varepsilon \\ \tan\varepsilon \end{matrix}\right\}, \tag{61}$$

$$\delta = \frac{\omega d}{2c_d}\cos\alpha = \frac{kd}{2}\left(n_d^2 - \sin^2\theta\right)^{1/2} = \frac{1}{2}x\left(n_d^2 - y^2\right)^{1/2}, \tag{62a}$$

$$\varepsilon = \frac{\omega d}{2c_t}\cos\beta = \frac{kd}{2}\left(n_t^2 - \sin^2\theta\right)^{1/2} = \frac{1}{2}x\left(n_t^2 - y^2\right)^{1/2}, \tag{62b}$$

$$\gamma = c_t/c_d, \qquad n_d = c/c_d, \qquad n_t = c/c_t, \tag{63a}$$

$$x = kd, \qquad y = \sin\theta, \qquad k = \omega/c. \tag{63b}$$

For a lossless medium, one has $|T|^2 + |R|^2 = 1$. The case of a plane fluid layer imbedded in another fluid (Fiorito and Überall, 1979) is obtained from the preceding in the limit $c_t = \beta = 0$ so that $C_s \to \cot\delta$, $C_a \to \tan\delta$.

In the following, we shall consider the case $c_d > c_t > c$ so that $n_d < n_t < 1$, and critical angles of total reflection

$$\theta_d = \sin^{-1} n_d, \qquad \theta_t = \sin^{-1} n_t \tag{64}$$

appear.

Total transmission ($|T| = 1$) occurs when

$$C_s C_a = \tau^2. \tag{65a}$$

The eigenvibrations of the plate are determined by requiring a transmitted wave to be present without the presence of an incident wave. This corresponds to $|T| \to \infty$, or the vanishing of the denominators in Eq. (59), i.e., for the fluid-loaded plate:

$$C_s = i\tau \qquad \text{or} \qquad C_a = -i\tau; \tag{65b}$$

for the free plate ($\rho = 0$, i.e., $\tau = 0$) this becomes

$$C_s = 0 \qquad \text{or} \qquad C_a = 0. \tag{65c}$$

Equations (65b) and (65c) lead to the dispersion relations for the fluid-loaded and the free plate, respectively. All three equations [Eqs. (65)] become approximately equal if $\tau \ll 1$, this being the basis of our following resonance approximation.

The dispersion relations are given in terms of the trace velocity v of the incident wave on the plate, with

$$v = c/\sin\theta = c_d/\sin\alpha = c_t/\sin\beta \tag{66}$$

(using Snell's law), plotted versus the frequency–thickness product fd. For the fluid-loaded plate, they must be given in terms of a complex angle of incidence θ (Pitts *et al.*, 1976) due to the complexity of Eq. (65b); but for the free plate, θ remains real. This will also be the case in our resonance formalism below.

Figure 23 shows the free-plate dispersion curves for a steel plate. These fall into two families, corresponding to antisymmetric (*a*, solid

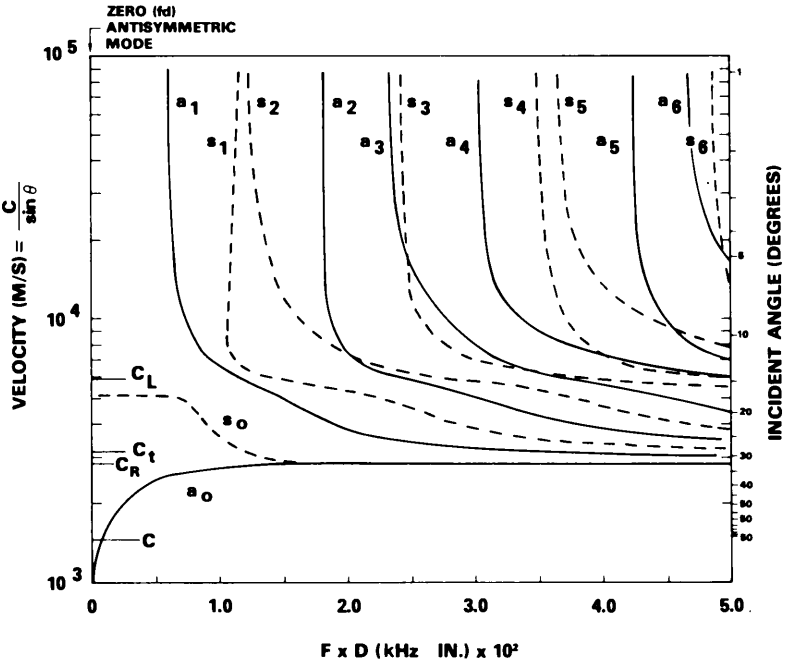


FIG. 23. Rayleigh and Lamb-mode dispersion curves for a steel plate (Fiorito *et al.*, 1979).

curves) and symmetric (s, dashed curves) plate deformations. Each curve corresponds to a vibrational mode, labeled by integer indices m_a or m_s. The "Lamb modes" a_{m_a} and $s_{m_s}(m_a > 0, m_s > 0)$ for $fd \to \infty$ approach the bulk sheer speed c_t from above; the Rayleigh modes a_0 and s_0 both approach c_R (the speed of the Rayleigh wave on the surface of an elastic half-space in vacuo) as $fd \to \infty$.

At the cutoff points of the Lamb wave dispersion curves (corresponding to $\theta = 0$, i.e., $v \to \infty$), one has an integral number m of half-wavelengths of the free plate waves equaling the plate thickness (see, e.g., Brekhovskikh, 1960), i.e., $k_{d,t} = m\pi/d$, so that a constructive interference of back-and-forth reflected waves occurs. For the fluid layer, this condition holds along the entire dispersion curve (using the normal component of k_d), i.e., for oblique incidence, while for the elastic plate, this is then no longer the case and only the condition of vanishing stresses is satisfied on the plate surfaces.

Equations (59) show that each of the two terms in T as well as in R will exhibit a resonance behavior when the real parts of the denominators vanish; this corresponds exactly to Eqs. (65c) describing the free-plate

dispersion relations. These resonances will be more pronounced the smaller the imaginary part of the denominator τ, i.e., the lighter the fluid loading.

The two independent variables of the problem are taken as a frequency–thickness variable $x \equiv kd$, and an angular variable $y \equiv \sin\theta$. Resonance forms of T and R may be obtained in either of these variables, holding the other one fixed.

Taking y fixed, we expand the real parts of the denominators of Eqs. (59) around their resonance values, i.e. around the free-plate dispersion curves:

$$C_{s,a}(x) \cong C_{s,a}(x_{m_{s,a}}) + (x - x_{m_{s,a}})C_{s,ax}(x_{m_{s,a}}) + \cdots, \tag{67a}$$

where $C_{s,ax} = \partial C_{s,a}/\partial x$, with the absolute terms vanishing, since Eqs. (65c),

$$C_s(x_{m_s}) = 0, \qquad C_a(x_{m_a}) = 0 \tag{67b}$$

determine the resonance frequencies x_{m_s}, x_{m_a}, respectively. Although these equations must be solved numerically, they are purely real (for a nonabsorbing plate) and hence easier to solve than, e.g., Eqs. (65b) for the fluid-loaded plate. This points out one advantage of our resonance approach.

With Eqs. (67), T and R assume the standard resonance forms

$$T = \sum_{m_s} \frac{-\frac{1}{2}i\Gamma_{m_s}}{x - x_{m_s} + \frac{1}{2}i\Gamma_{m_s}} + \sum_{m_a} \frac{\frac{1}{2}i\Gamma_{m_a}}{x - x_{m_a} + \frac{1}{2}i\Gamma_{m_a}}, \tag{68a}$$

$$R = \sum_{m_s} \frac{x - x_{m_s}}{x - x_{m_s} + \frac{1}{2}i\Gamma_{m_s}} + \sum_{m_a} \frac{x - x_{m_a}}{x - x_{m_a} + \frac{1}{2}i\Gamma_{m_a}}, \tag{68b}$$

each resonance expression being valid in the vicinity of its own resonance position $x = x_{m_{s,a}}$. The resonance widths are

$$\Gamma_{m_{s,a}} = \mp 2\tau / C_{s,ax}(x_{m_{s,a}}). \tag{68c}$$

The resonances in T and R may be interpreted as poles in the lower half of the complex-x plane, located at

$$x_{\text{pole}} = x_{m_{s,a}} - \tfrac{1}{2}i\Gamma_{m_{s,a}}. \tag{69a}$$

Furthermore, R has nearby zeros located at the real positions

$$x_{\text{zero}} = x_{m_{s,a}} \tag{69b}$$

We note that the resonance values $(fd)_{m_{s,a}}$ may be read off the free-plate dispersion curves of Fig. 23, by intersecting all the branches with a horizontal line $c/\sin\theta$ corresponding to the given angle of incidence θ. The widths $\Gamma_{m_{s,a}}$, however, depend on the fluid loading of the plate through the parameter τ.

The resonance expressions Eqs. (68) represent T and R only in the vicinity of the resonances, while Eqs. (59) represent these quantities everywhere. The value of the resonance formalism lies in the fact that it furnishes explicit expressions for the position, width, and strength of each resonance contribution to T and R as a function of all the variables and parameters. To this, we may add the potentially very important observation that, with the resonance characteristics being determined as explicit functions of the material constants, it appears possible to carry out an inverse determination of the material constants of a plate by examining the resonances experimentally. This, in other words, represents a novel method to determine the plate's composition from its acoustic response at resonance. We may illustrate this for the case of a fluid layer where (Fiorito and Überall, 1979)

$$(kd)_{\text{res}} = m\pi/\left(n_d^2 - \sin^2\theta\right)^{1/2} \tag{70a}$$

and

$$\Gamma = (\rho/\rho_{\text{p}})(4/\cos\theta), \tag{70b}$$

so that the position of the resonance measures, via $n_{\text{d}} = c/c_{\text{d}}$, the sound speed in the layer, and the width of the resonances measures the density ratio ρ/ρ_{p}. This is the simplest example for the solution of the inverse scattering problem (determination of the properties of the scattering object from a measurement of the echo) via a study of the resonances. An analogous solution for the inverse scattering of elastic waves from a fluid-filled cavity using the resonance method will be given in Section IV.

Resonances in the angular variable y, with x fixed, may be obtained similarly as in the preceding, with the result

$$T = \sum_{m_{\text{s}}} \frac{\frac{1}{2}i\gamma_{m_{\text{s}}}}{y - y_{m_{\text{s}}} - \frac{1}{2}i\gamma_{m_{\text{s}}}} + \sum_{m_{\text{a}}} \frac{-\frac{1}{2}i\gamma_{m_{\text{a}}}}{y - y_{m_{\text{a}}} - \frac{1}{2}i\gamma_{m_{\text{a}}}}, \tag{71a}$$

$$R = \sum_{m_{\text{s}}} \frac{y - y_{m_{\text{s}}}}{y - y_{m_{\text{s}}} - \frac{1}{2}i\gamma_{m_{\text{s}}}} + \sum_{m_{\text{a}}} \frac{y - y_{m_{\text{a}}}}{y - y_{m_{\text{a}}} - \frac{1}{2}i\gamma_{m_{\text{a}}}}, \tag{71b}$$

where the resonance positions $y_{m_{\text{s,a}}}$ are the solutions of

$$C_{\text{s}}(y_{m_{\text{s}}}) = 0, \qquad C_{\text{a}}(y_{m_{\text{a}}}) = 0, \tag{71c}$$

and with resonance widths

$$\gamma_{m_{\text{s,a}}} = \pm 2\tau_{\text{s,a}}/C_{\text{s,a}y}(y_{m_{\text{s,a}}}); \tag{71d}$$

we here defined

$$\tau_{\text{s,a}} = \frac{\rho}{\rho_p}\left[\frac{n_{\text{d}}^2 - y_{m_{\text{s,a}}}^2}{1 - y_{m_{\text{s,a}}}^2}\right]^{1/2}. \tag{71e}$$

T and R are meromorphic functions with poles

$$y_{\text{pole}} = y_{m_{s,a}} + \tfrac{1}{2} i \gamma_{m_{s,a}} \tag{72a}$$

in the upper half of the complex-y plane; R also has real zeros at

$$y_{\text{zero}} = y_{m_{s,a}}. \tag{72b}$$

Graphically, the y-resonance positions may be read off the dispersion curves of Fig. 23 by intersecting all the branches with a vertical line at a given frequency, the intersection having the ordinate value $c/y_{m_{s,a}}$.

Numerical agreement between the exact and the resonance-theoretical values of T and R for a steel plate in water were found to be excellent (Fiorito *et al.*, 1979). Figure 24a shows the individual y resonances in $|T|^2$ for a value $fd = 116.29$ kHz in. Fig. 24b shows the exact expression for $|T|^2$. The resonances are seen to interfere in this latter expression at $\theta = 13°$ and 30°. More important interferences are illustrated in Fig. 25 where for $fd = 73.1$ kHz in. we plot the transmission loss TL $\equiv \log |T|^2$ for three individual resonances in the resonance approximation (a), and using the exact expression for T (b). The latter exhibits severe interferences between the three overlapping resonances, which would be difficult to interpret as such, but it is seen that the individual resonances furnished by our resonance approximation allow us to disentangle and to interpret the interference features. Further examples of this kind are provided by Madigosky and Fiorito (1979).

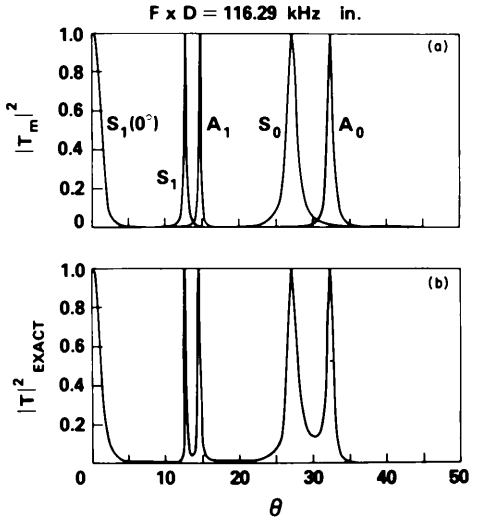


FIG. 24. Individual transmission resonances (a), and exact transmission through a steel plate in water (b) at $fd = 116.29$ kHz in., plotted versus angle of incidence θ (Fiorito *et al.*, 1979).

The resonance at $\theta = 0$ in Fig. 24 has been found to differ from the form of Eqs. (71) since for $y \to L$, one has $C_{ys,a} \to 0$. In the expansion of $C_{s,a}$ around the resonance value of y, the quadratic term has to be retained in this case as the lowest nonvanishing term, with a corresponding change in the resonance expression of Eq. (71).

We may add that our explicit expressions for y pole and zero positions of R as functions of x, i.e., Eqs. (72), permitted analytic proofs for the empirical findings, via numerical computations, of Pitts *et al.* (1976) regarding the limiting forms at $x \to \infty$ of these quantities. In particular, it is demonstrated (Fiorito *et al.*, 1979) that of all the plate modes shown in Fig. 23, only the contribution to R of the symmetric Rayleigh mode s_0 will

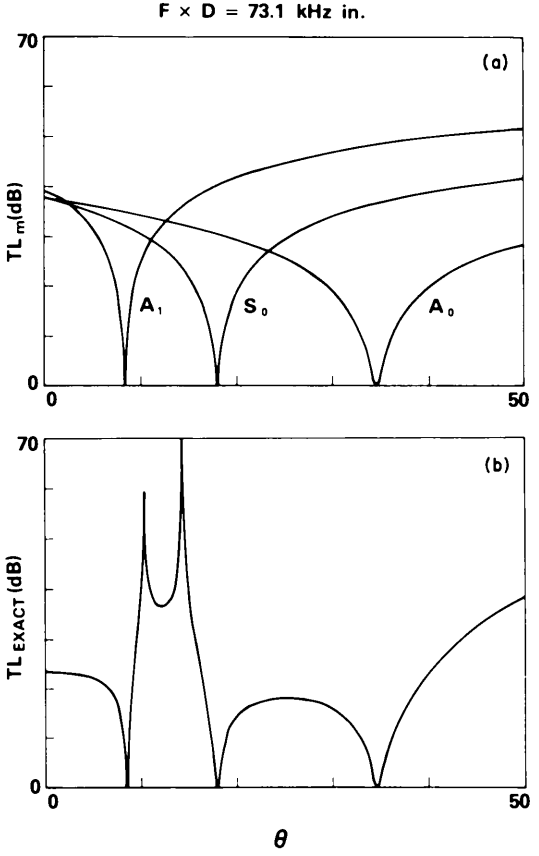


FIG. 25. Individual transmission resonances (a), and their interferences in the exact transmission through a steel plate in water (b) at $fd = 73.1$ kHz in., plotted versus angle of incidence θ (Fiorito *et al.*, 1979).

persist in the half-space limit $d \to \infty$ (i.e., $fd \to \infty$), and represent here the Rayleigh wave for this case, while the contributions of all other modes (a_0, and the Lamb modes) will drop out since for them, poles and zeros in the limit $x \to \infty$ approach each other identically. Only for the surviving s_0-Rayleigh mode, one has

$$R \to \left(y - y_0 + \tfrac{1}{2} i\gamma_o\right) / \left(y - y_0 - \tfrac{1}{2} i\gamma_0\right), \tag{73}$$

where pole and zero form a complex-conjugate pair.

III. Elastic-Wave Scattering from Cavities and Inclusions

In this section, the theory of resonance scattering will be developed for the case of the scattering of elastic waves from fluid-filled cavities or from solid inclusions. The principal difference between these problems and the acoustic one discussed above lies in the presence of *two* types of waves, namely, compressional (dilatational, longitudinal) waves as well as shear (transverse) waves, and in the possibility of "mode conversion", i.e., the conversion of one type of such waves into another during the scattering process. In the following, we shall discuss scattering from fluid-filled spherical cavities, from cylindrical cavities, and finally from spherical solid inclusions. The applicability of the results obtained here extends from nondestructive testing to seismology, oil and mineral prospecting, and finally to medical ultrasonics.

A. Spherical Fluid-Filled Cavities

Several recent publications (Gaunaurd and Überall, 1978, 1979, 1979b, 1979c; Gaunaurd, 1980; Gaunaurd *et al.*, 1978, 1979; Brill *et al.*, 1980, 1980a) have discussed the scattering of elastic (viz, compressional and shear) waves from spherical fluid-filled cavities in solid media in the light of the resonance theory of scattering. We first consider compressional waves and later shear waves incident on the same scatterer. The analysis presented here assumes the solid medium containing the cavity to be either merely elastic, or else sound-absorbing and thus viscoelastic. The basic Kelvin–Voigt viscoelastic model (Gaunaurd, 1976, 1977) was used (Gaunaurd and Überall, 1978) just as in the coated-shell analysis discussed in Section II,B to describe the material behavior and its absorptive capacity, and to construct the general formulation in a way directly connected to material properties. The net effect of introducing viscous absorptive losses into the medium is the appearance of complex dilatational and shear wave numbers into the mathematical description, whose real parts coincide with

the real wave numbers of the purely elastic medium when it is assumed lossless. The need for fundamental viscoelastic models to describe the losses is based on the need to express the complex wave numbers in terms of actual material parameters. It is noteworthy that if the solid medium exterior to the cavity becomes fluid in composition we again recover the compressional results for bubbles in fluids, which we discussed in Section II,D.

A basic characteristic of wave propagation in solid media is that mode conversion is always present since for every type of incident wave (either dilatational or shear) there are always two associated scattered waves, one of each type, which travel with different speeds. This is due to the fact that solid media can support two basically distinct types of waves, and these are generated in the scattering process even though an incident wave of only one type may have initially been present. In the terminology of nuclear scattering, elastic wave scattering is a multichannel process (Newton, 1966) with *two* channels. The possible additional presence of transverse shear waves (of arbitrary polarization) traveling at their own speed is the distinctive characteristic of vector waves in elastic media. This situation is more general than either acoustics (i.e., scalar problem with one speed) or electromagnetism (i.e., vector problem with one speed). There exists no electromagnetic wave of the compressional (or longitudinal) type.

1. *Incident Compressional Waves*

When a compressional elastic wave is incident on a fluid-filled cavity, the fluid filler is resonantly excited into its eigenvibrations, which are then communicated to the cavity wall. The cavity filler appears impenetrable and does not vibrate at most frequencies; however, in the vicinity of the cavity resonance frequencies, there is appreciable wave penetration into the fluid filler, with noticeable vibration of the fluid mass. Since only a limited amount of physical information can usually be extracted from calculations of (summed) cross sections, we have started the trend to analyze each partial wave one at a time. It turns out that each modal contribution can be decomposed into two interfering portions, which have been termed the "background" and the "resonances." The first corresponds to geometrical scattering from the cavity boundary (assumed evacuated), while the second corresponds to the resonance vibrations of the filler.

Let a be the radius of a spherical fluid-filled cavity contained in an infinite viscoelastic medium. A plane compressional elastic wave traveling through the medium (see Fig. 26) is completely specified by the scalar

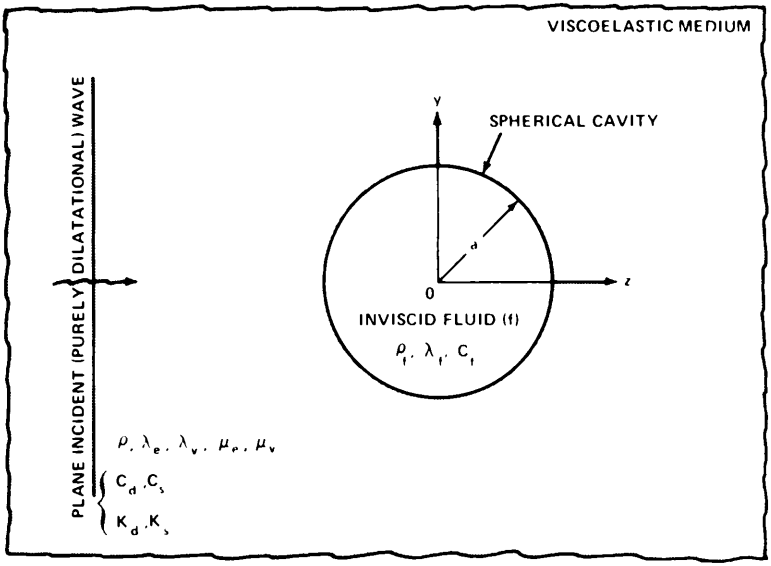


FIG. 26. The geometry for a fluid-filled spherical cavity in a viscoelastic medium of infinite extent. $c_f^2 = \lambda_f/\rho_f$, $k_f = \omega/c_f$.

potential (Gaùnaurd and Überall, 1978)

$$\phi_{\text{inc}} = \phi_0 e^{-i\omega t} \sum_{n=0}^{\infty} i^n (2n+1) j_n(\kappa_d r) P_n(\cos\theta). \tag{74}$$

Scattered dilatational and shear waves are present in the viscoelastic medium which can be represented, respectively, by the potential

$$\phi_{\text{sc}} = \phi_0 e^{-i\omega t} \sum_{n=0}^{\infty} i^n A_n (2n+1) h_n^{(1)}(\kappa_d r) P_n(\cos\theta), \tag{75a}$$

$$\psi_{\text{sc}} = \phi_0 e^{-i\omega t} \sum_{n=0}^{\infty} i^n B_n (2n+1) h_n^{(1)}(\kappa_s r) P_n(\cos\theta), \tag{75b}$$

where ψ is the scalar potential associated with the shear wave.

The potential for the fluid region interior to the cavity is given by

$$\phi_f = \phi_0 e^{-i\omega t} \sum_{n=0}^{\infty} i^n C_n (2n+1) j_n(k_f r) P_n(\cos\theta), \tag{76}$$

where $k_f = \omega/c_f$ and $\psi_f \equiv 0$ since the inner fluid cannot support shear stresses. The complex wave numbers in the viscoelastic medium are given by

$$\kappa_d^2 = \omega^2/c_d^2(1 - i\omega M), \qquad \kappa_s^2 = \omega^2/c_s^2(1 - i\omega N) \tag{77}$$

where

$$M = (\lambda_v + 2\mu_v)/(\lambda_e + 2\mu_e), \qquad N = \mu_v/\mu_e,$$
$$c_d^2 = (\lambda_e + 2\mu_e)/\rho, \qquad c_s^2 = \mu_e/\rho, \tag{78}$$

and λ_e, μ_e are the elastic Lamé parameters while λ_v, μ_v are their viscous counterparts. The sets of coefficients A_n, B_n, and C_n are determined from the boundary conditions at the cavity wall $r = a$. These conditions are statements of continuity of normal stresses, shear stresses, and radial displacements at the wall. These sets of coefficients satisfy a linear system of three equations in three unknowns:

$$\begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} A_n \\ B_n \\ C_n \end{bmatrix} = \begin{bmatrix} A_1^* \\ A_2^* \\ A_3^* \end{bmatrix}. \tag{79}$$

Cramer's rule can be used to solve this system for A_n, B_n, and C_n. The known coefficients d_{ij} and A_i^* in Eq. (79) are all found from the boundary conditions and they can be shown to be

$$\begin{aligned}
d_{11} &= \kappa_d a h_n^{(1)\prime}(\kappa_d a), \\
d_{12} &= n(n+1) h_n^{(1)}(\kappa_s a), \\
d_{13} &= -k_f a j_n'(k_f a), \\
d_{21} &= \left[2n(n+1) - \kappa_s^2 a^2\right] h_n^{(1)}(\kappa_d a) - 4\kappa_d a h_n^{(1)\prime}(\kappa_d a), \\
d_{22} &= 2n(n+1)\left[\kappa_s a h_n^{(1)\prime}(\kappa_s a) - h_n^{(1)}(\kappa_s a)\right], \\
d_{23} &= (\rho_f/\rho)\kappa_s^2 a^2 j_n(k_f a), \\
d_{31} &= \kappa_d a h_n^{(1)\prime}(\kappa_d a) - h_n^{(1)}(\kappa_d a), \\
d_{32} &= \left[n(n+1) - 1 - \tfrac{1}{2}\kappa_s^2 a^2\right] h_n^{(1)}(\kappa_s a) - \kappa_s a h_n^{(1)\prime}(\kappa_s a), \\
d_{33} &= 0, \\
A_1^* &= -\kappa_d a j_n'(\kappa_d a), \\
A_2^* &= \left[\kappa_s^2 a^2 - 2n(n+1)\right] j_n(\kappa_d a) + 4\kappa_d a j_n'(\kappa_d a), \\
A_3^* &= j_n(\kappa_d a) - \kappa_d a j_n'(\kappa_d a).
\end{aligned} \tag{80}$$

It is immediate that, for example,

$$A_n = \frac{1}{D_n} \begin{vmatrix} A_1^* & d_{12} & d_{13} \\ A_2^* & d_{22} & d_{23} \\ A_3^* & d_{32} & 0 \end{vmatrix}, \tag{81a}$$

$$D_n = \begin{vmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & 0 \end{vmatrix}, \tag{81b}$$

and analogously for the other coefficients. These are related to two of the four elements of the scattering matrix S_n by the formulas

$$S_n^{pp} = 1 + 2A_n, \qquad S_n^{ps} = 2B_n. \tag{82}$$

In the far field, and ignoring the time dependence, the factors multiplying the quantities $\exp(i\kappa_d r)/r$ and $\exp(i\kappa_s r)/r$ are the scattering amplitudes $f^{pp}(\theta)$ and $f^{ps}(\theta)$ of the compressional and shear waves scattered by the cavity. In the backscattering direction $\theta = \pi$, these partial-wave scattering amplitudes, normalized to the cavity radius, are

$$\frac{1}{a} f^{pp}(\pi) = \sum_{n=0}^{\infty} \frac{1}{a} f_n^{pp}(\pi) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2i\kappa_d a}(2n+1)(S_n^{pp} - 1), \tag{83a}$$

$$\frac{1}{a} f^{ps}(\pi) = \sum_{n=0}^{\infty} \frac{1}{a} f_n^{ps}(\pi) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2i\kappa_s a}(2n+1) S_n^{ps}. \tag{83b}$$

The expressions containing S-matrix elements appearing in Eqs. (83) can be put in the following form (Gaunaurd and Überall, 1978, 1979):

$$S_n^{pp} - 1 = \exp(2i\xi_n^s)\left\{ \frac{z_2 - z_1}{z_1 - F_n} + 2i \exp(-i\xi_n^s) \sin \xi_n^s \right\}, \tag{84a}$$

$$S_n^{pp} = \exp(2i\xi_n^s) \frac{z_2 - F_n}{z_1 - F_n}, \tag{84b}$$

$$S_n^{ps} = \exp(2i\xi_n^s)\left\{ \frac{\tilde{z}_2 - z_1}{z_1 - F_n} + 1 \right\}$$

$$= \exp(2i\xi_n^s) \frac{\tilde{z}_2 - F_n}{\tilde{z}_1 - F_n}, \tag{84c}$$

provided the following definitions are introduced:

$$z_1 \equiv -\kappa_s^2 a^2 \begin{vmatrix} d_{11} & d_{12} \\ d_{31} & d_{32} \end{vmatrix} \Big/ \begin{vmatrix} d_{21} & d_{22} \\ d_{31} & d_{32} \end{vmatrix} \equiv \Delta_n^s + i s_n^s \equiv \tilde{z}_1, \tag{85a}$$

$$z_2 \equiv -\kappa_s^2 a^2 \begin{vmatrix} f_1 & d_{12} \\ f_3 & d_{32} \end{vmatrix} \Big/ \begin{vmatrix} f_2 & d_{22} \\ f_3 & d_{32} \end{vmatrix} \equiv \Lambda_n^s + i\sigma_n^s, \tag{85b}$$

$$\tilde{z}_2 \equiv -\kappa_s^2 a^2 \begin{vmatrix} d_{11} & A_1^* \\ d_{31} & A_3^* \end{vmatrix} \Big/ \begin{vmatrix} d_{21} & A_2^* \\ d_{31} & A_3^* \end{vmatrix} \equiv \tilde{\Lambda}_n^s + i\tilde{\sigma}_n^s, \tag{85c}$$

$$F_n = \frac{\rho}{\rho_f} \frac{k_f a j_n'(k_f a)}{j_n(k_f a)}. \tag{86}$$

The three quantities f_i appearing above are

$$f_1 = -\kappa_d a h_n^{(2)}(\kappa_d a), \tag{87a}$$

$$f_2 = \left[\kappa_s^2 a^2 - 2n(n+1)\right] h_n^{(2)}(\kappa_d a) + 4\kappa_d a h_n^{(2)}(\kappa_d a), \tag{87b}$$

$$f_3 = h_n^{(2)}(\kappa_d a) - \kappa_d a h_n^{(2)}(\kappa_d a). \tag{87c}$$

We also define

$$S_n^{(0)pp} \equiv \exp(2i\xi_n^s) = \begin{vmatrix} f_2 & d_{22} \\ f_3 & d_{32} \end{vmatrix} \Big/ \begin{vmatrix} d_{21} & d_{22} \\ d_{31} & d_{32} \end{vmatrix}, \tag{88a}$$

$$S_n^{(0)ps} \equiv \exp(2i\tilde{\xi}_n^s) = 2\begin{vmatrix} d_{21} & A_2^* \\ d_{32} & A_3^* \end{vmatrix} \Big/ \begin{vmatrix} d_{21} & d_{22} \\ d_{31} & d_{32} \end{vmatrix}, \tag{88b}$$

and note that these are the respective limits of S_n^{pp}, S_n^{ps} for the case of an empty cavity for which $\rho_f \to 0$ (and $F_n \to \infty$). Substituting these definitions into Eqs. (84), and the results into Eqs. (83), we find the following (still exact) expressions for the scattering amplitudes of the partial waves:

$$\frac{1}{a} f_n^{pp}(\pi) = \frac{(-1)^n}{2i\kappa_d a}(2n+1)\left[\exp(2i\xi_n^s)\left(\frac{z_2 - z_1}{z_1 - F_n} + 2i\exp(-i\xi_n^s)\sin\xi_n^s\right)\right], \tag{89a}$$

$$\frac{1}{a} f_n^{ps}(\pi) = \frac{(-1)^n}{2i\kappa_s a}(2n+1)\left[\exp(2i\tilde{\xi}_n^s)\left(\frac{\tilde{z}_2 - z_1}{z_1 - F_n} + 1\right)\right]. \tag{89b}$$

To continue the analysis in the frequency domain, there are two alternatives at this point. One is to continue with an exact analysis by means of the Mittag–Leffler expansion of Eq. (86) (Gaunaurd and Überall, 1981), and the other is to introduce a linear approximation based on a Taylor expansion (Gaunaurd and Überall, 1978). Following the latter approach, we write

$$F_n(x) - \Delta_n^s(x) = \beta_{nl}^s(x - x_{nl}^s). \tag{90}$$

This is a one-term Taylor expansion about each of the roots of the characteristic equation

$$F_n(x_{nl}^s) = \Delta_n^s(x_{nl}^s), \qquad l = 1, 2, \ldots \tag{91}$$

where $x = \kappa_d a$. In this way, Eqs. (89) can be linearized to read

$$\frac{1}{a} f_n^{pp}(\pi) = \frac{(-1)^n}{2i\kappa_d a}(2n+1)e^{2i\xi_n^s}\left[\sum_{l=1}^{\infty} \frac{M_{nl}^s}{x - x_{nl}^s + \frac{1}{2}i\Gamma_{nl}^s} + 2i\exp(-i\xi_n^s)\sin\xi_n^s\right], \tag{92a}$$

$$\frac{1}{a} f_n^{ps}(\pi) = \frac{(-1)^n}{2i\kappa_s a}(2n+1)e^{2i\xi_n^s}\left[\sum_{l=1}^{\infty} \frac{\tilde{M}_{nl}^s}{x - x_{nl}^s + \frac{1}{2}i\Gamma_{nl}^s} + 1\right], \tag{92b}$$

where the resonance widths are defined by

$$\Gamma_{nl}^s = -2s_n^s/\beta_{nl}^s, \qquad \beta_{nl}^s = \frac{d}{dx}(F_n - \Delta_n)_{x=x_{nl}^s}, \tag{93}$$

and the numerators M_{nl}^s and $\tilde{M}_{nl}^s$ are

$$M_{nl}^s = (z_1 - z_2)/\beta_{nl}^s, \qquad \tilde{M}_{nl}^s = (z_1 - \tilde{z}_2)/\beta_{nl}^s. \tag{94}$$

The formulas of Eqs. (92) clearly show how each partial wave is composed of two portions that add together and interfere with each other. The first portion is the resonance contribution represented by the sums, and the second is the smooth background contribution of an empty cavity, represented by the last terms inside the square brackets. The rapidly fluctuating nature of the cavity cross section found in the subsequent numerical evaluation, or of the scattering amplitudes contained in it, is due to this interference between resonances and background within each partial wave.

The above analysis of the resonances in the partial-wave amplitudes was done in the frequency (i.e., $\kappa_d a$) domain for fixed mode numbers. The resonances can also be studied as functions of continuous mode order n at fixed frequencies, just as bubbles and shells were analyzed previously in Section II. This can be done by noting that the resonance terms contained in Eqs. (92) show the presence of poles in the lower half of the complex-x plane, located at

$$x_{\text{pole}} = x_{nl} - \tfrac{1}{2}i\Gamma_{nl}. \tag{95}$$

Expansions such as in Eq. (37) also apply in the present case, which lead to Regge poles in the scattering function appearing at the same positions in the n plane given by Eq. (38b). To interpret the resonances in terms of ("creeping") surface waves along the cavity meridians, the Watson-Sommerfeld method can be used, with a result analogous to that in Eqs. (43b) and (43c). It is found (Gaunaurd and Überall, 1979) that for

fluid-filled cavities, one has

$$f_{\hat{n}_l}^{(l)} = \phi_0 \exp(-i\omega t)\left(\hat{n}_l + \tfrac{1}{2}\right) \exp\left[\tfrac{1}{2} i\pi\left(n_l + \tfrac{1}{2}\right) - \tfrac{1}{4} i\pi\right]$$

$$\times h_{\hat{n}_l}^{(1)}(\kappa_d r) S_{n_l}^{(0)pp}\left\{ i\hat{\Gamma}_{n_l l} + \frac{z_2 - z_1^*}{\beta_{nl} x'_{n_l l}} \right\}, \tag{96}$$

using Eqs. (38b) and (38c) and other notation of Section II,B,1, which again provides us with a rigorous picture of circumferential waves propagating around the cavity.

Equation (45) carries over for the spherical cavity with an analogous interpretation. Wavelength, attenuation, and phase velocities of the surface waves are also given by Eqs. (46), and numerical results based on these equations will be plotted and discussed below. We conclude (Gaunaurd and Überall, 1979) that resonances become physically apparent due to the resonant reinforcement or buildup caused by phase matching of the circumferential waves during repeated circumnavigations. It also becomes clear that a resonance appears at frequencies x_{nl} at which the speed of the lth surface wave launched around the cavity coincides with the phase velocity of its nth multipole vibrational mode.

2. *Incident Shear Waves*

If $\hat{\varepsilon}_z$ is the propagation direction of an incident shear wave polarized in the $\hat{\varepsilon}_x$ direction, then the incident shear wave $\psi_0 \exp(i\kappa_s z)\hat{\varepsilon}_y$ can be expressed in terms of the two nonvanishing Debye potentials as follows (Brill *et al.*, 1980):

$$\psi_{inc} = \psi_0 (4\pi)^{1/2} \sum_{n=1}^{\infty} i^n \left(\frac{2n+1}{n(n+1)} \right)^{1/2} j_n(\kappa_s r) Y_{e,n,1}(\theta, \phi), \tag{97a}$$

$$\chi_{inc} = \psi_0 (4\pi)^{1/2} \sum_{n=1}^{\infty} i^n \left(\frac{2n+1}{n(n+1)} \right)^{1/2} \frac{i}{\kappa_s} j_n(\kappa_s r) Y_{o,n,1}(\theta, \phi), \tag{97b}$$

where even (e) and odd (o) harmonics are given by the expressions

$$\begin{pmatrix} 1 \\ i \end{pmatrix} Y_{\substack{e \\ o}, n, m} = (-1)^m \left[\frac{2n+1}{4\pi} \frac{(n-m)!}{(n+m)!} \right]^{1/2} P_n^m(\cos\theta) \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix}. \tag{98}$$

Since mode conversion takes place, there will appear both a scattered compressional wave ϕ_{sc} and a scattered shear wave described by two potentials ψ_{sc} and χ_{sc}. There is only a compressional wave transmitted into the fluid filler, described by the potential ϕ^f. These four potentials depend, respectively, on sets of constants A_n'', B_n', C_n', and E_n', which are determined

from the boundary conditions. Using Eqs. (97) and (98), the total (i.e., incident plus scattered) fields can be written as follows:

(a) In the viscoelastic solid:

$$\phi = -\psi_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} A_n'' h_n^{(1)}(\kappa_d r) P_n^1(\cos\theta)\cos\phi, \tag{99a}$$

$$\psi = -\psi_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} \left[j_n(\kappa_s r) + B_n' h_n^{(1)}(\kappa_s r) \right] P_n^1(\cos\theta)\cos\phi, \tag{99b}$$

$$\chi = -\psi_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} \frac{i}{\kappa_s} \left[j_n(\kappa_s r) + C_n' h_n^{(1)}(\kappa_s r) \right] P_n^1(\cos\theta)\sin\phi. \tag{99c}$$

(b) In the inner fluid:

$$\phi^f = -\psi_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} E_n' j_n(k_f r) P_n^1(\cos\theta)\cos\phi. \tag{100}$$

The boundary conditions at the cavity wall determine the constants A_n'', B_n', C_n', and E_n'. Using the new quantity

$$A_n' = A_n'' / \{ n(n+1)(\kappa_d/\kappa_s) \}^{1/2}, \tag{101}$$

and introducing scattering amplitudes f^{ij} by the relations

$$\phi_{sc} = (\psi_0/r)\exp(i\kappa_d r) f^{sp}(\theta,\phi), \tag{102a}$$

$$\psi_{sc} = (\psi_0/r)\exp(i\kappa_s r) f^{ss}(\theta,\phi), \tag{102b}$$

$$\chi_{sc} = (\psi_0/r)\exp(i\kappa_s r) f^{tt}(\theta,\phi), \tag{102c}$$

we can construct the additional elements of the S matrix, now pertaining to shear-wave incidence, as follows:

$$S_n^{sp} = 2A_n', \qquad S_n^{ss} = 1 + 2B_n', \qquad S_n^{tt} = 1 + 2C_n'. \tag{103}$$

Substitution yields these elements in the form

$$S_n^{sp} = -\frac{2}{n(n+1)(\kappa_d/\kappa_s)} \frac{1.}{D_n} \begin{vmatrix} \mathrm{Re}\, d_{12} & d_{12} & d_{13} \\ \mathrm{Re}\, d_{22} & d_{22} & d_{23} \\ \mathrm{Re}\, d_{32} & d_{32} & 0 \end{vmatrix}, \tag{104}$$

$$S_n^{ss} = \frac{-1}{D_n} \begin{vmatrix} d_{11} & d_{12}^* & d_{13} \\ d_{21} & d_{22}^* & d_{23} \\ d_{31} & d_{32}^* & 0 \end{vmatrix}, \qquad S_n^{tt} = -\frac{d_{22}^*}{d_{22}}, \tag{105}$$

where an asterisk denotes complex conjugation and Re denotes the real

part. This form of the result will be valid only for real wave numbers κ_s, κ_d. The matrix elements for p-wave incidence found in the first part (1), Eq. (82), are explicitly shown here for completeness:

$$S_n^{pp} = \frac{-1}{D_n}\begin{vmatrix} d_{11}^* & d_{12} & d_{13} \\ d_{21}^* & d_{22} & d_{23} \\ d_{31}^* & d_{32} & 0 \end{vmatrix}, \tag{106a}$$

$$S_n^{ps} = \frac{-2}{D_n}\begin{vmatrix} d_{11} & \operatorname{Re} d_{11} & d_{13} \\ d_{21} & \operatorname{Re} d_{21} & d_{23} \\ d_{31} & \operatorname{Re} d_{31} & 0 \end{vmatrix}. \tag{106b}$$

The complete S matrix for elastic-wave scattering in this case is thus

$$S_n = \begin{bmatrix} S_n^{tt} & 0 & 0 \\ 0 & S_n^{ss} & S_n^{ps} \\ 0 & S_n^{sp} & S_n^{pp} \end{bmatrix}. \tag{107}$$

The scattering amplitudes in Eq. (102) now reduce to

$$\frac{1}{a} f^{sp}(\theta,\phi) = \sum_{n=1}^{\infty} \frac{1}{a} f_n^{sp}(\theta,\phi)$$

$$= -\frac{1}{2i\kappa_s a} \sum_{n=1}^{\infty} (2n+1) S_n^{sp} P_n^1(\cos\theta)\cos\phi, \tag{108a}$$

$$\frac{1}{a} f^{ss}(\theta,\phi) = \sum_{n=1}^{\infty} \frac{1}{a} f_n^{ss}(\theta,\phi)$$

$$= -\frac{1}{2i\kappa_s a} \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} (S_n^{ss} - 1) P_n^1(\cos\theta)\cos\phi, \tag{108b}$$

$$\frac{1}{a^2} f^{tt}(\theta,\phi) = \sum_{n=1}^{\infty} \frac{1}{a^2} f_n^{tt}(\theta,\phi)$$

$$= -\frac{1}{2i\kappa_s a} \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} \frac{i}{\kappa_s a} (S_n^{tt} - 1) P_n^1(\cos\theta)\sin\phi. \tag{108c}$$

The S matrix in Eq. (107) can be shown to be symmetric and unitary.

To develop now the resonance theory of elastic-wave scattering for shear-wave incidence (Brill *et al.*, 1980, 1980a), we note that the S-matrix elements in Eqs. (104) and (105) can be written in the form

$$S_n^{ss} = S_n^{(s)ss} \frac{F_n - z_3}{F_n - z_1}, \qquad S_n^{sp} = S_n^{(s)sp} \frac{F_n - \tilde{z}_3}{F_n - z_1}, \tag{109}$$

where z_1 and F_n are given in Eqs. (85) and (86), respectively, and z_3, $\tilde{z}_3$, $S_n^{(s)ss}$ and $S_n^{(s)sp}$ are

$$z_3 = -\kappa_s^2 a^2 \begin{vmatrix} d_{11} & d_{12}^* \\ d_{31} & d_{32}^* \end{vmatrix} \Big/ \begin{vmatrix} d_{21} & d_{22}^* \\ d_{31} & d_{32}^* \end{vmatrix}, \tag{110a}$$

$$\tilde{z}_3 = -\kappa_s^2 a^2 \begin{vmatrix} d_{12} & d_{12}^* \\ d_{32} & d_{32}^* \end{vmatrix} \Big/ \begin{vmatrix} d_{22}^* & d_{22} \\ d_{32}^* & d_{32} \end{vmatrix}, \tag{110b}$$

$$S_n^{(s)ss} = -\begin{vmatrix} d_{21} & d_{22}^* \\ d_{32} & d_{32}^* \end{vmatrix} \Big/ \begin{vmatrix} d_{21} & d_{22} \\ d_{31} & d_{32} \end{vmatrix} = \exp(2i\eta_n^s), \tag{111a}$$

$$S_n^{(s)sp} = -\begin{vmatrix} \mathrm{Re}\, d_{22} & d_{22} \\ \mathrm{Re}\, d_{32} & d_{32} \end{vmatrix} \Big/ \begin{vmatrix} d_{21} & d_{22} \\ d_{31} & d_{32} \end{vmatrix} = \exp(2i\tilde{\eta}_n^s). \tag{111b}$$

Following our previous linearization procedure, it can be quickly shown that the scattering amplitudes in Eqs. (102) can be expressed in the resonance form

$$\frac{1}{a} f_n^{sp}(\theta,\phi) = -\frac{2n+1}{n(n+1)} \frac{\exp(2i\tilde{\eta}_n^s)}{2i\kappa_d a} \left\{ \sum_{l=1}^{\infty} \frac{\tilde{N}_n}{x - x_{nl} + \frac{1}{2} i\Gamma_{nl}} + 1 \right\}$$
$$\times P_n^1(\cos\theta)\cos\phi, \tag{112a}$$

$$\frac{1}{a} f_n^{ss}(\theta,\phi) = -\frac{2n+1}{n(n+1)} \frac{\exp(2i\tilde{\eta}_n^s)}{2i\kappa_s a} \left\{ \sum_{l=1}^{\infty} \frac{N_n}{x - x_{nl} + \frac{1}{2} i\Gamma_{nl}} + 2ie^{-i\eta_n^s} \sin\eta_n^s \right\}$$
$$\times P_n^1(\cos\theta)\cos\phi, \tag{112b}$$

$$\frac{1}{a^2} f_n^{tt}(\theta,\phi) = -\frac{2n+1}{n(n+1)} \frac{i}{\kappa_s^2 a^2} [\exp(i\xi_n)] \sin\xi_n P_n^1(\cos\theta)\sin\phi, \tag{112c}$$

which are the counterparts of Eqs. (92) for shear-wave incidence. The modulus of these expressions at $\theta = \pi$ and for suitable values of ϕ [i.e., $\phi = 0$ for Eqs. (112a, b) and $\phi = \pi/2$ for Eq. (112c)], are usually plotted versus $k_d a$ and/or n. In the above equations, we have defined

$$\Gamma_{nl} = -2\,\mathrm{Im}\, z_1/\beta_n, \qquad N_n = (z_1 - z_2)/\beta_n, \qquad \tilde{N}_n = (z_1 - \tilde{z}_3)/\beta_n. \tag{113}$$

Equations (112a) and (112b) exhibit the interference between resonances and background terms. Equation (112c) has no resonance contributions, and thus, it consists only of a smooth background. This absence of resonances in the tt-mode of scattering follows because S_n^{tt} depends only on x_s, and it is independent of the fluid contained in the cavity, (i.e., of k_f). The mode-order analysis follows the same pattern as in Section I (Gaunaurd and Überall, 1979).

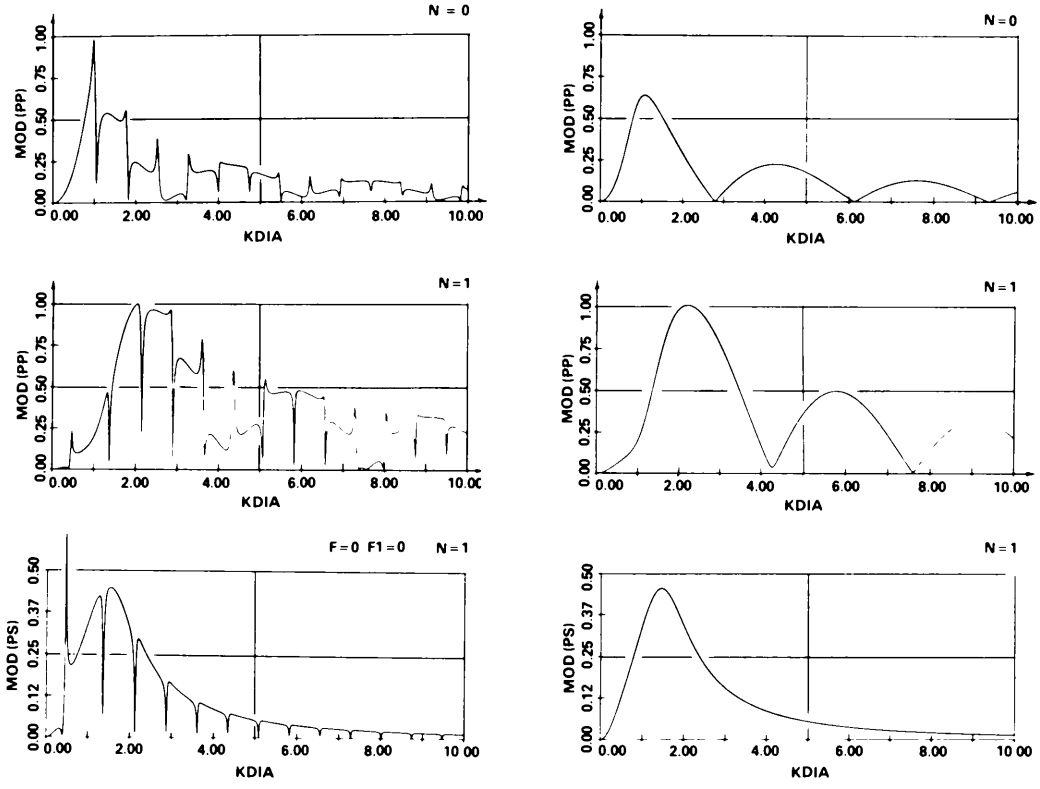


FIG. 27. Plots of $|f_n^{PP}/a|$ (top and center) and $|f_n^{PS}/a|$ (bottom) versus $k_d a$ for the $n = 0$ or 1 mode (as marked), of a water-filled cavity in an aluminum matrix. Left-hand column: Total contributions. Right-hand column: smooth backgrounds of an empty cavity (Gaunaurd and Überall, 1979).

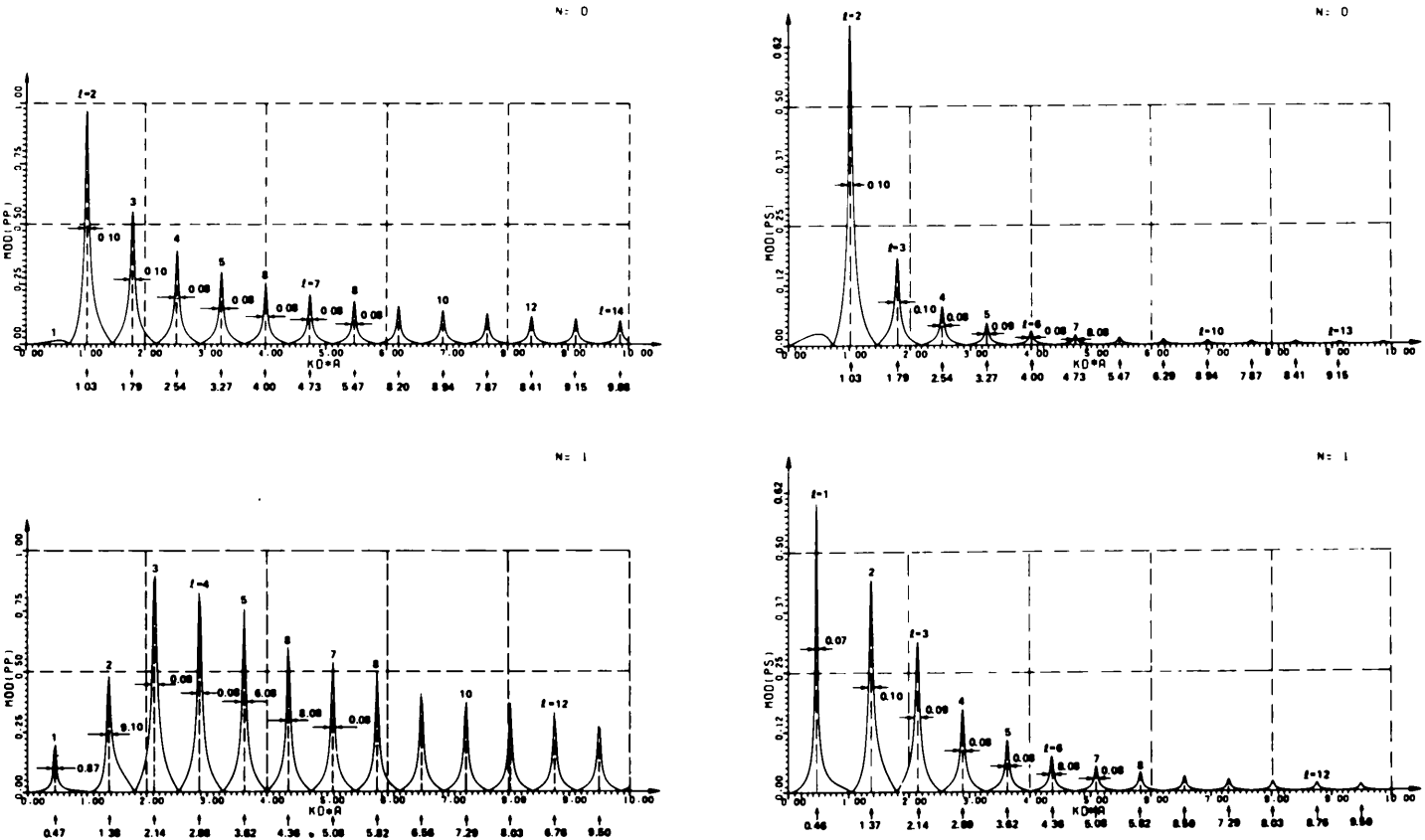


FIG. 28. Resonances (labeled by the index l) of the moduli $|f_n^{PP}/a|$ (left-hand column) and $|f_n^{PS}/a|$ (right-hand column), obtained by subtracting the two complex quantities whose moduli are displayed in both columns of Fig. 27. The first two modes of a water-filled cavity in aluminum. Locations and widths of the resonances are shown (Gaunaurd and Uberall, 1979).

3. *Numerical Results*

Figure 26 shows the geometry and material parameters of the spherical cavity in an infinite viscoelastic medium. Figure 27 presents plots of the scattering amplitudes $|f_n^{PP}/a|$ and $|f_n^{Ps}/a|$ as obtained from Eqs. (83) and (92) for a water-filled (left-hand column) and empty (right-hand column) cavity in an aluminum matrix, plotted versus $k_d a$. The two upper graphs in both columns are for the $n = 0$ and 1 modes of f_n^{PP}, and the bottom graphs are for the $n = 1$ mode of the mode-converted amplitude f_n^{Ps}. Subtracting the quantities whose moduli are plotted in the right- and left-hand columns of Fig. 27 yield the resonances. These are exhibited (in both cases) in Fig. 28 for the same substances and for the $n = 0$ and 1 modes of vibration, respectively. It is clear from Fig. 28 that for this material combination, the resonances are relatively wide. For air-filled cavities in rubber, the resonances are quite narrow, and Fig. 29 displays the first two partial waves of

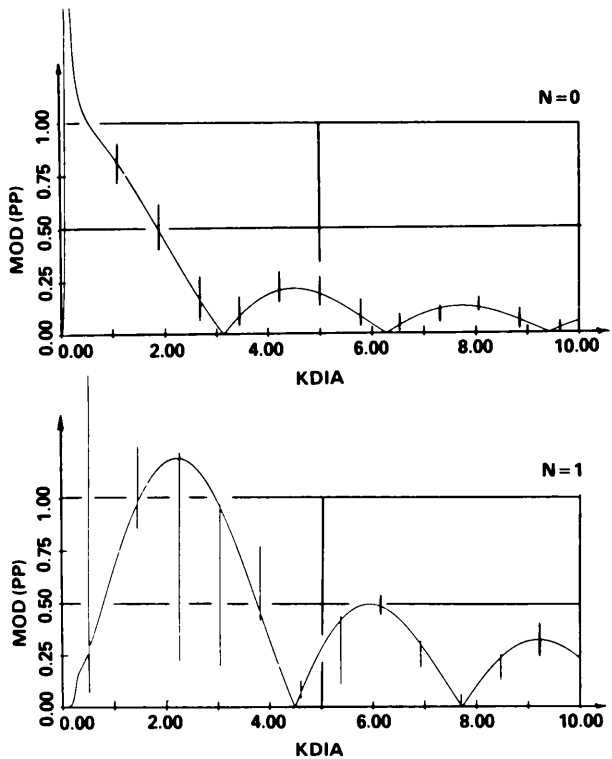


FIG. 29. Plots of $|f_n^{PP}/a|$ versus $k_d a$ for $n = 0$ and 1 for an air-filled cavity in a rubber matrix. Here the resonances are thin lines superimposed on the smooth backgrounds of an evacuated cavity (Gaunaurd and Überall, 1979).

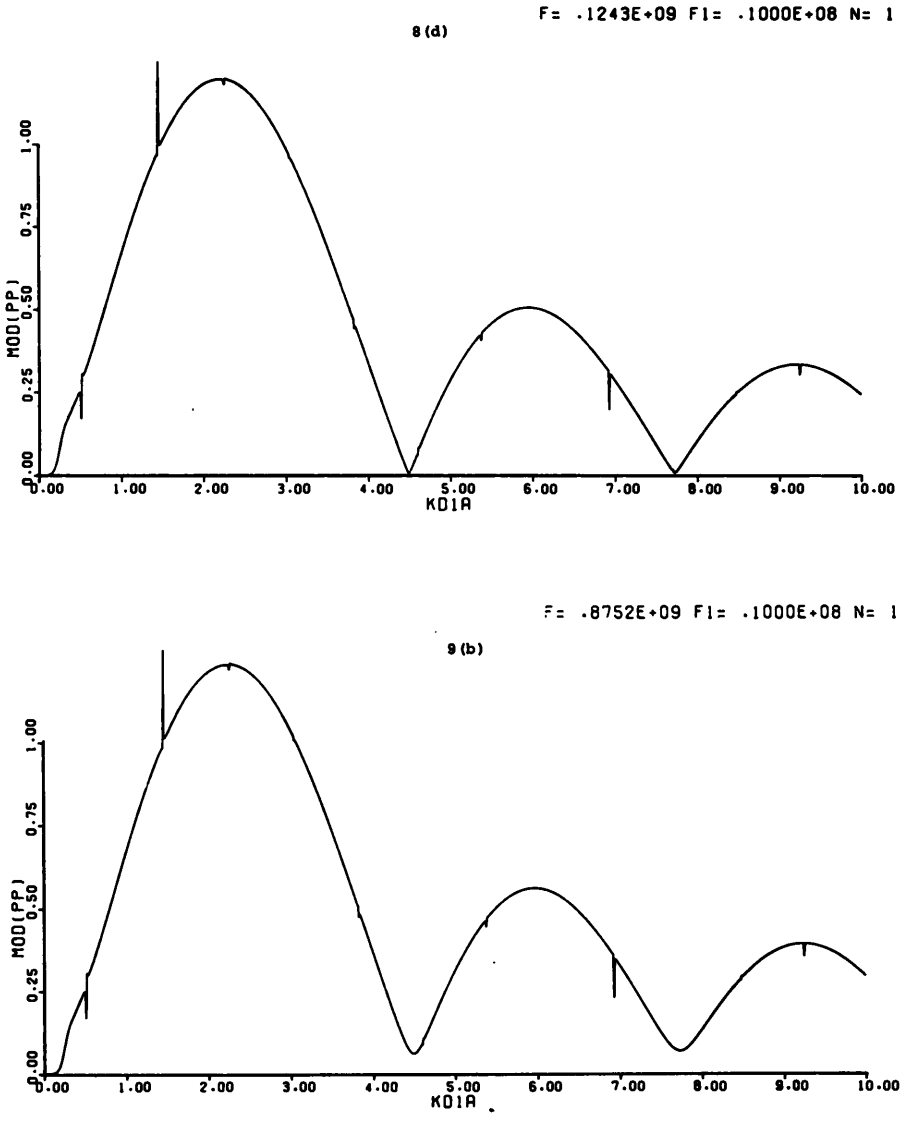


FIG. 30. The first mode of $|f_n^{PP}/a|$ $(n = 1)$ plotted versus $k_d a$ for two values of the parameters F, F_1. Higher absorption levels fills in the regions near the zeros of $|f_1^{PP}|$. This shows a further energy leak in addition to mode conversion (Gaunaurd and Überall, 1978).

$|f_n^{pp}/a|$ in this instance. We can concentrate on a specific partial wave, say the first, and study how it varies with changing viscosity. Particular results in this case are displayed in Fig. 30. Viscosity in the model used is controlled by two parameters β_d, β_s, which are

$$\beta_d = (F + 2F_1)/2\rho c_d^2, \qquad \beta_s = F_1/2\rho c_s^2. \tag{114}$$

Values of these parameters are shown in Table II of Gaunard and Überall (1978). Figure 30 shows changes in $|f^{pp}|$ that occur when F takes on different values along the first row of the mentioned Table II. The overall effect of viscosity is as follows. Absorption does not alter the "resonance features" of the scattering amplitudes. It only alters the "backgrounds," since absorption takes place only at the viscoelastic cavity *wall*. It is clear from Fig. 30 that for the same shear absorption values, controlled by F_1, an increase in dilatational absorption, controlled by F, fills in the zeros of the pp-scattering amplitude. It was also shown (Gaunard and Überall, 1978) that for the same amount of dilatational absorption, an increase in shear absorption increases the high-frequency values of $|f^{pp}|$ with hardly any change in the low-frequency region.

Equation (46c) gives the phase velocity of the lth surface wave. These velocities are displayed nondimensionally for a water-filled cavity in an aluminum matrix in Fig. 31. The scattering amplitudes $|f^{pp}|$ and $|f^{ps}|$ can be plotted versus mode order n for fixed $k_d a$ values. Figure 32 (left-hand column) shows the plots of $|f^{pp}/a|$ versus n (assumed continuous) for $k_d a = 5$ and 10. After the subtraction of a suitable nonresonant background corresponding to an empty cavity, the resonances are isolated in the *mode-order* domain n, as studied in Section I). The right-hand column in Fig. 32 shows these resonances for the same $k_d a$ values and materials. The width $\hat{\Gamma}_{nl}$ of the resonances in the n domain shown in Fig. 32 can be plotted versus n for each resonance peak ($l = 1, 2, \ldots$) as $k_d a$ is increased. This generates the family of "Regge trajectories" shown in Fig. 33. The ordinates $\operatorname{Im} \nu|_{\nu = n_l}$ plotted in Fig. 33 can be used in Eq. (46b) to construct the angular amplitude decay parameter $\theta_l(x)$ (or "life angle") of the creeping waves, Fig. 34 showing the plot of θ_l versus x for the first ten surface waves launched around the cavity. Isometric views of the resonances, whose "side views" were given in Figs. 28 and 32, are presented in Fig. 35. These three-dimensional plots of the modular surfaces of $|f_n^{pp}|$ and $|f_n^{ps}|$ contain all the information about these scattering amplitudes which we discussed in the text, in the frequency and mode-order domain.

Further three-dimensional plots, now for the case of shear wave incidence discussed in Section III,A,2), are shown in Fig. 36. The upper (bottom) graph corresponds to the mode-converted (non-mode-converted) scattering amplitude $|f_n^{ss}/a|_{res}(|f_n^{sp}/a|_{res})$. Slices of these plots along planes

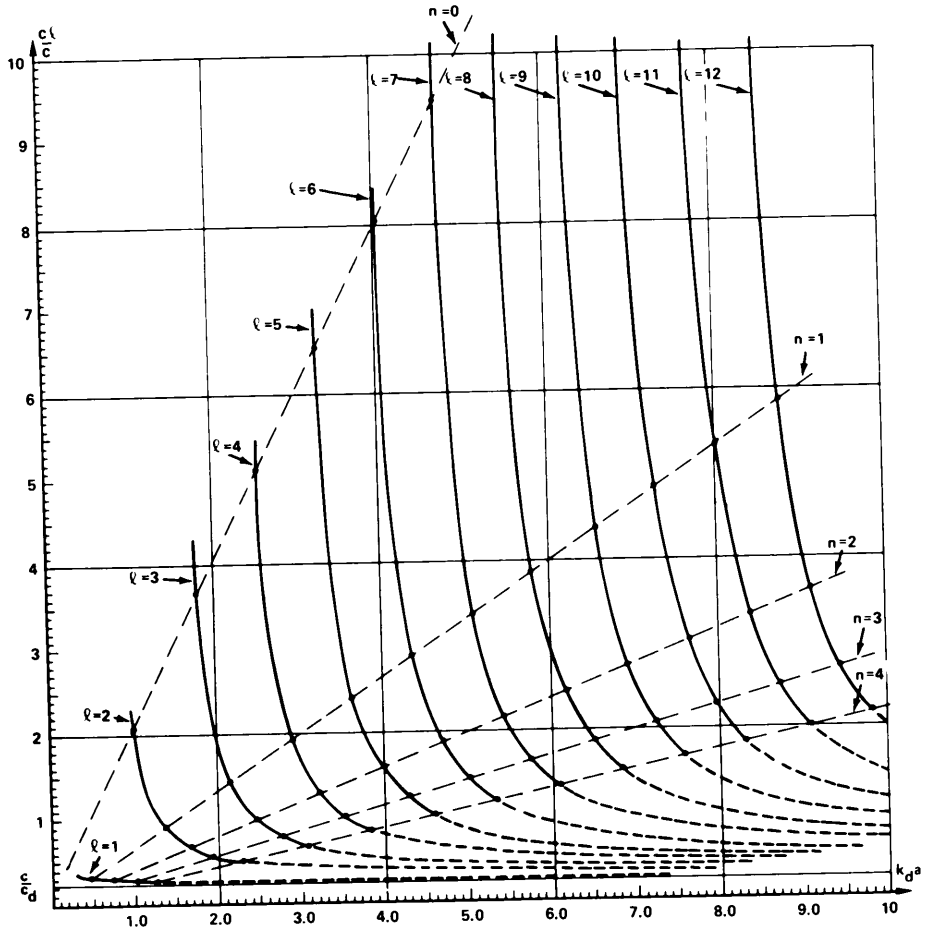


FIG. 31. Dispersion curves of the surface waves (labeled by l) on a water-filled cavity in an aluminum matrix (Gaunaurd and Überall, 1979).

parallel to the $k_d a$ or the n axes generate the families of plots of resonances versus n or x, analogous to those shown in Figs. 32 and 28 for compressional wave incidence.

The (summed) scattering amplitudes for shear wave incidence in Eqs. (108) or (112a) and (112b) are plotted in Fig. 37 (top and center) for $\theta = \pi/2$ and $\phi = 0$, adding 20 terms in the sums that define these quantities. The plots are produced at $\theta = \pi/2$ rather than at $\theta = \pi$ because the plotted quantities vanish in this latter direction. The bottom portion of Fig. 37 shows the summed amplitude $|f^{tt}/a^2|$ as given in Eq. (112c) with

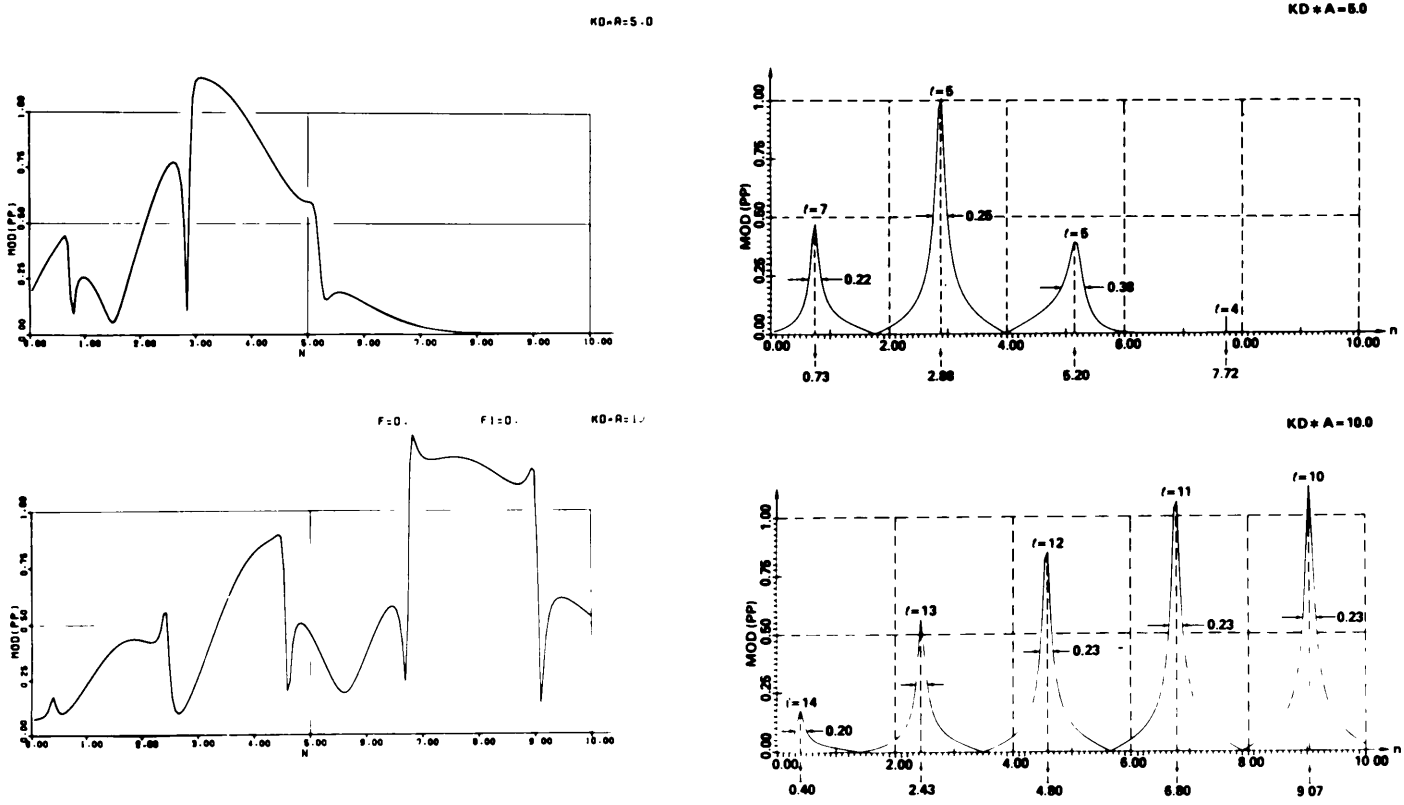


FIG. 32. Plot of $|f_n^{PP}/a|$ versus n for fixed values of $k_d a$ (i.e., $k_d a = 5, 10$) before (left-hand column) and after (right-hand column) the subtraction of smooth backgrounds to isolate the resonances displayed in the right-hand column. For a water-filled cavity in an aluminum matrix (Gaunaurd and Überall, 1979).

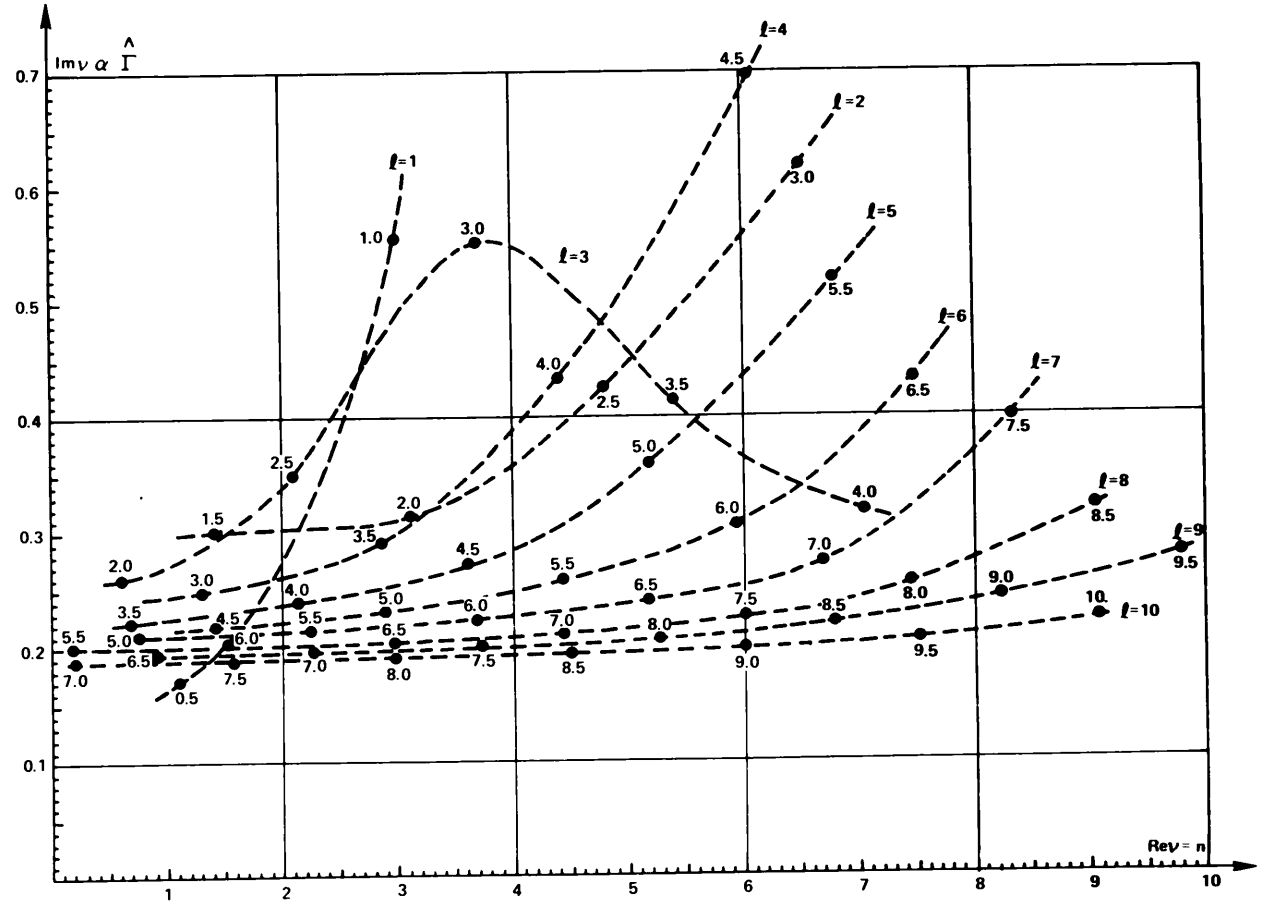


FIG. 33. Regge trajectories of the poles in the scattering amplitude associated with the surface waves labeled $l = 1-10$. Numbers along each trajectory indicate the increasing values of $x = k_0 a$ (Gaunaurd and Überall, 1979).

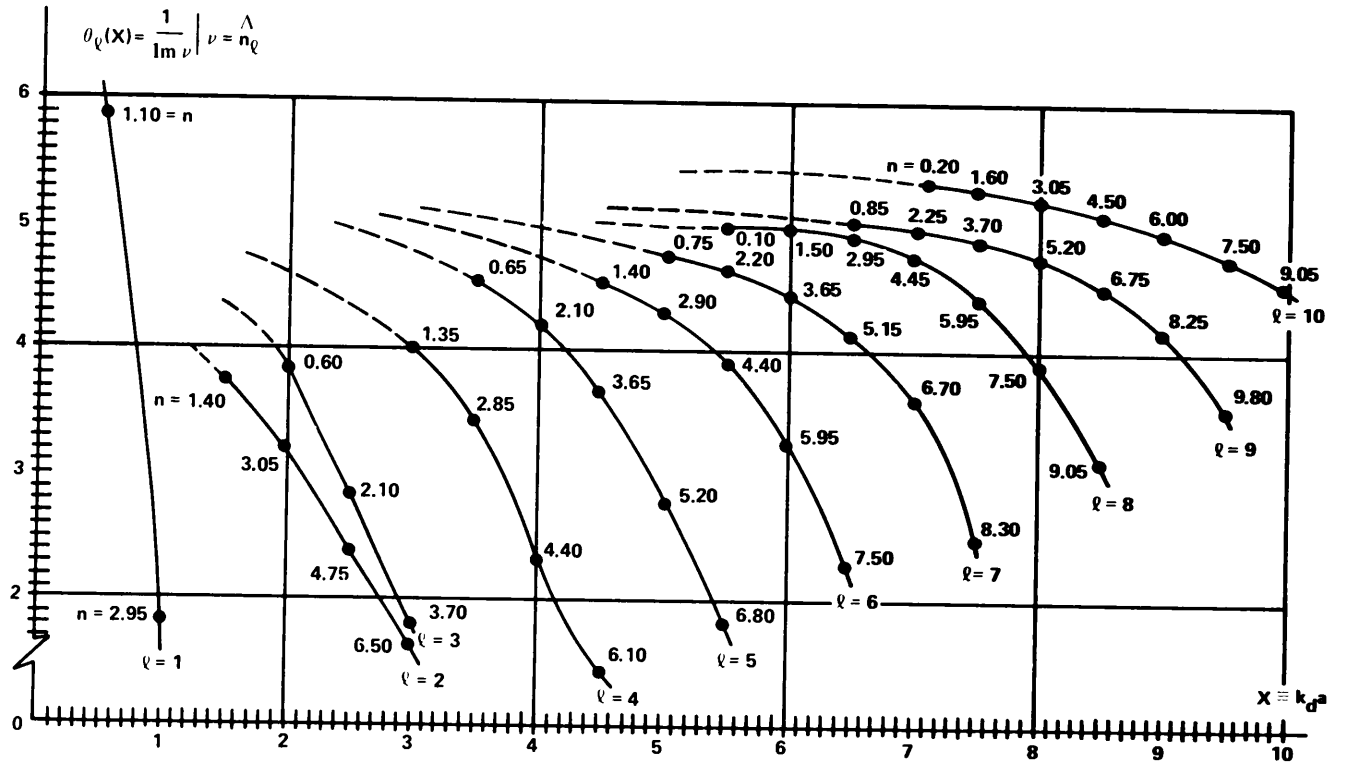


FIG. 34. Decay angle θ_l plotted versus $x \equiv k_d a$ for the surface waves (labeled by l). Numbers along each curve are values of mode order n (Gaunaurd and Überall, 1979).

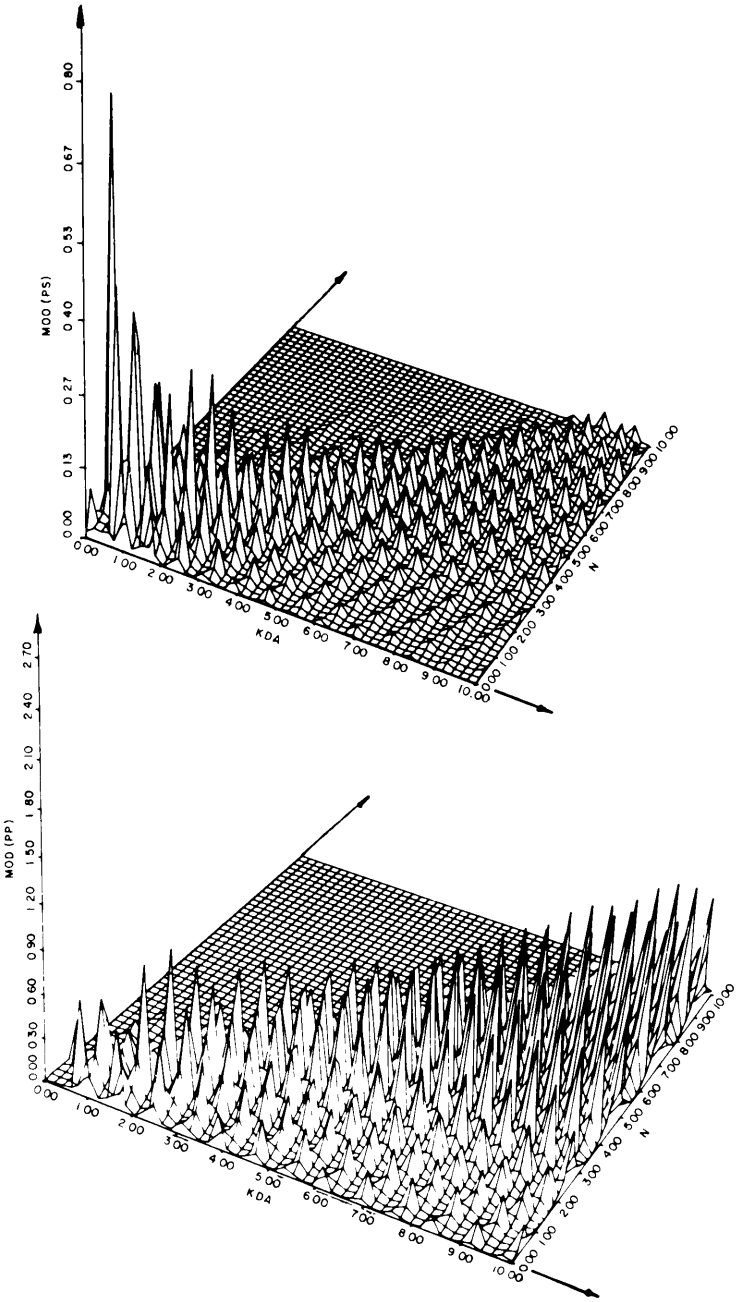


FIG. 35. Modular surfaces of the quantities $|f_n^{ps}/a|$ and $|f_n^{pp}/a|$ once the nonresonant backgrounds have been subtracted, plotted three-dimensionally versus $k_d a$ *and* n, for water-filled spherical cavity in aluminum, under compressional wave incidence (Gaunaurd and Überall, 1979).

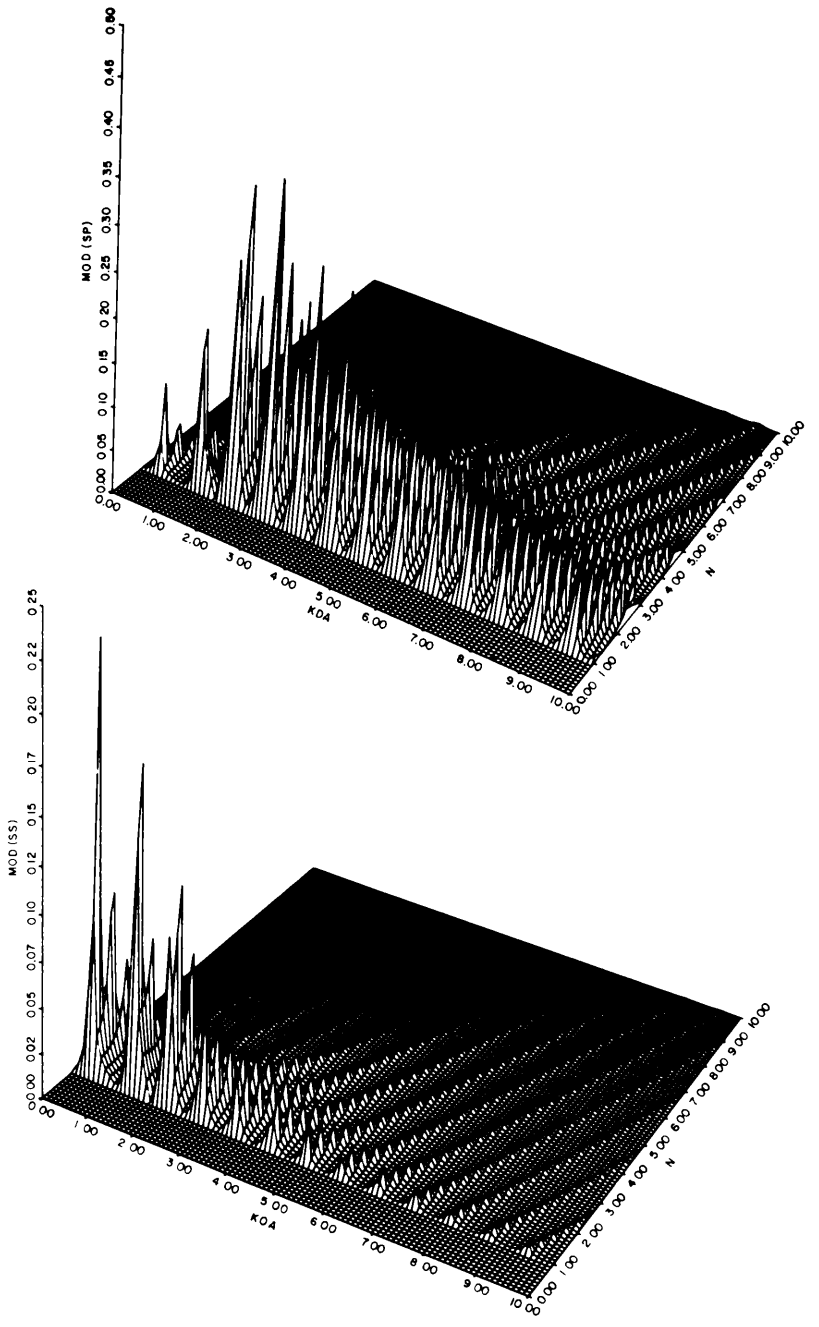


FIG. 36. The counterpart of Fig. 35 now for the quantities $|f_n^{sp}/a|$ and $|f_n^{ss}/a|$ in the case of shear-wave incidence. The resonance spikes group along clearly visible families of ripples (i.e., "Regge poles") in this type of plot (Brill *et al.*, 1980).

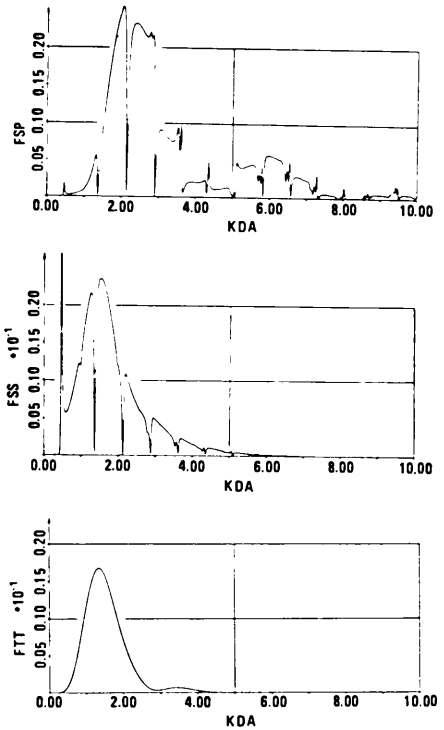


FIG. 37. Plots of the (summed squared) amplitudes $|a^{-1}f^{sp}(\theta=\pi/2,\ \phi=0)|^2$, $|a^{-1}f^{ss}(\theta=\pi/2,\phi=0)|^2$, $|a^{-2}f^{tt}(\theta=\pi/2,\ \phi=\pi/2)|^2$ versus $k_d a$ for a water-filled cavity in an aluminum matrix, adding 20 terms in the defining sum. The bottom graph shows no deep minima since t waves can excite no resonances in the filler substance (Brill *et al.*, 1980).

$\theta=\phi=\pi/2$, adding 20 terms in the series. This amplitude contains no resonances and it consists entirely of background contributions. Much additional material is found in the original references.

4. *The Giant Monopole Resonance (Cavities and Bubbles)*

The study of this resonance in the monopole mode of vibration of fluid-filled cavities in solids under *p*-wave incidence (Gaunaurd *et al.*, 1979) has shown the normalized backscattered amplitude of this $n=0$ mode to be

$$G\equiv(1/a)f_0^{pp}(\pi)=A_0/ix, \tag{115}$$

where

$$A_0=-\frac{\rho_f x j_0'(x)j_0(\beta x)-\rho\beta x j_0'(\beta x)\left[j_0(x)+4xj_0'(x)/x_s^2\right]}{\rho_f x h_0^{(1)\prime}(x)j_0(\beta x)-\rho\beta x j_0'(\beta x)\left[h_0^{(1)}(x)+4xh_0^{(1)\prime}(x)/x_s^2\right]}, \tag{116}$$

with $x = k_d a$ and $\beta = k_f/\kappa_d$. For bubbles in liquids, the above expression admits some simplification since $x_s \to \infty$. For cavities in solids, it is possible to write Eq. (115) in the resonance form

$$G = \frac{1}{x}\left\{ \frac{\exp(i\xi_0^s)\sin\eta_0^s}{\exp(-i\xi_0^s)\left[\cos\eta_0^s + \left(F_0/\alpha^2 x^2 p_0\right) - i\sin\eta_0^s\right.} + \exp(i\xi_0^s)\sin\xi_0^s \right\} \tag{117}$$

where the following quantities have been introduced:

$$\begin{aligned}
\xi_0^s &= -x - \tan^{-1}(x/y), \\
\eta_0^s &= \tan^{-1} x + \tan^{-1}(x/y), \\
F_0 &= \frac{\rho}{\rho_f}\frac{\beta x j_0'(\beta x)}{j_0(\beta x)} = \frac{\rho}{\rho_f}(\beta x \cot\beta x - 1), \\
y &= \alpha^2 x^2 - 1, \\
\alpha &= \kappa_s/2\kappa_d, \\
p_0 &= \left(\frac{1+x^2}{y^2+x^2}\right)^{1/2}.
\end{aligned} \tag{118}$$

Equation (117) immediately shows the resonances in the first term and the background in the second, interacting with each other in the usual form of the resonance theory, but all in an *exact* fashion since no approximations have been introduced thus far. The resonances, obtained as the roots of the characteristic equation

$$\beta x \cot\beta x = 1 - \frac{\rho_f}{\rho}\frac{\alpha^2 x^2}{\alpha^2 x^2 - 1}, \tag{119}$$

are infinite in number even though this analysis and characteristic equation pertain only to the *monopole* mode of oscillation. Since the giant monopole resonance always occurs for $x \ll 1$, it is of interest to find approximations to Eq. (115) in the Rayleigh region. It is found that for $x \ll 1$,

$$G = \frac{1}{e^{ix}}\frac{x^2/x_0^2 - x^2/3}{1 - x^2/x_0^2 - ix} \tag{120}$$

where

$$x_{0s} = 1/\alpha, \qquad x_{0f} = \left(3\rho_f/\rho\beta^2\right)^{1/2}, \qquad x_0^2 = x_{0s}^2 + x_{0f}^2, \tag{121}$$

the last relation corresponding to circular resonance frequencies of

$$\omega_0^2 = \omega_{0s}^2 + \omega_{0f}^2 = \left(\frac{2c_s}{a}\right)^2 + \frac{3\rho_f}{\rho}\frac{c_f^2}{a^2}. \tag{122}$$

The first part, ω_{0s}, is the Meyer–Brendel–Tamm (MBT) resonance (Meyer *et al.*, 1958) for empty cavities in solids and the second part, ω_{0f}, is the Minnaert (1933) resonance correction due to the presence of the fluid filler.

For evacuated ($\rho_f = 0$) cavities in solids, it follows that

$$|G| = \alpha^2 x^2 / \left[(1 - \alpha^2 x^2)^2 + x^2 \right]^{1/2}, \tag{123}$$

which is the MBT result. For gas bubbles in water (i.e., $\omega_{0s} = 0$) it follows from Eq. (120) that

$$|G| = |1/\left[(1 - \omega_{0f}^2/\omega^2) + ix \right]|, \tag{124}$$

which is the standard result for bubbles found by Eckart (1946).

The second term in Eq. (117) is a pure background contribution proper to an evacuated cavity. With the relations in Eqs. (118), it can be written in the form

$$G = \frac{\cos x + (\alpha^2 x^2 - 1)(\sin x/x)}{1 - \alpha^2 x^2 - ix}. \tag{125}$$

This "background" contribution, in the Rayleigh region ($x \ll 1$), reduces to

$$G \cong \alpha^2 x^2 / (1 - \alpha^2 x^2 - ix), \tag{126}$$

which agrees with Eq. (123). This result can be linearized and put in the standard Breit–Wigner (see, e.g., Roman 1965) form of the resonance theory with the result

$$G \cong (-x_{0s}/2) / \left(x - x_{0s} + \tfrac{1}{2} i\Gamma \right), \tag{127}$$

where $x_{0s} = 1/\alpha$ and $\Gamma = 1/\alpha^2$. It is possible to generalize this result for fluid-filled cavities in absorptive rubber. The final linearized result is

$$G \cong \frac{(-x_{0s}/2)(1 + 2i\delta_s)}{x_R - x_0' + \tfrac{1}{2} i\Gamma_0}, \tag{128}$$

where

$$\begin{aligned} x_0' &= x_0 + x_0^2 \delta_d/2 + \delta_s x_{0f}^2, \\ \Gamma_0 &= x_0^2 + 2\delta_s x_{0s}^2/x_0; \end{aligned} \tag{129}$$

x_0, x_{0s}, and x_{0f} are given in Eq. (121) and δ_d, δ_s are the loss tangents contained in the expressions

$$\begin{aligned} x &= x_R(1 + i\delta_d), \\ \alpha x &= (x_R/x_{0s})(1 + i\delta_s). \end{aligned} \tag{130}$$

Converting to circular frequencies, we can write the relative half-width in

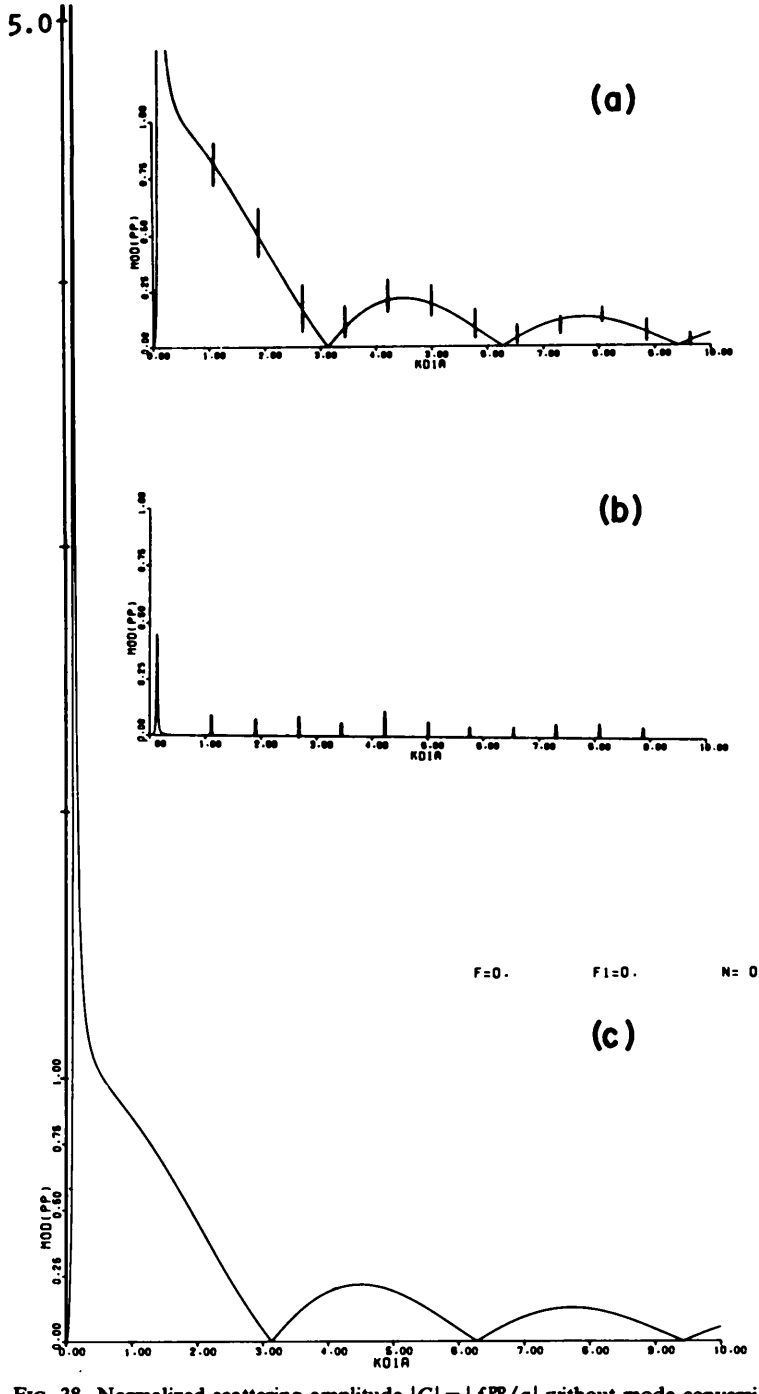


FIG. 38. Normalized scattering amplitude $|G| = |f_0^{PP}/a|$ without mode conversion, for the monopole ($n = 0$) contribution in the case of an air-filled cavity in lossless rubber, versus $k_d a$; (a) combined background plus narrow resonances, (c) isolated background ($\rho_f = 0$) showing the giant monopole spike, and (b) isolated resonances from the differences [(a)–(c)] (Gaunaurd *et al.*, 1979).

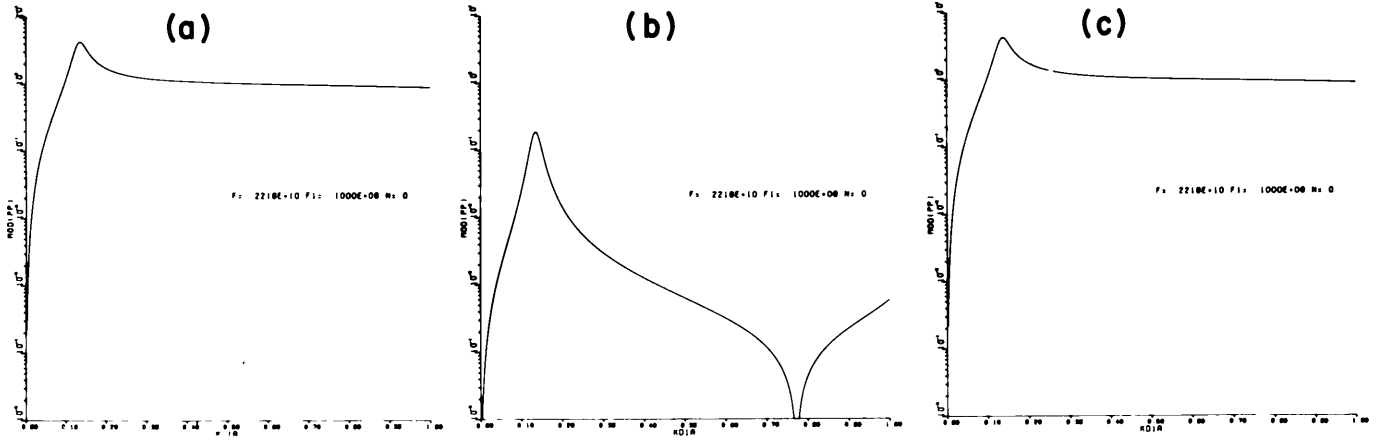


FIG. 39. Plot of $|f_n^{pp}/a|$ versus $k_d a$ ($n = 0$, monopole mode) for an air-filled cavity in rubber of viscous level controlled by the parameters $F = 2.218 \times 10^9$ and $F_1 = 10^7$ dyn/cm^2. The composite contribution (a) is split into its background (c) and pure resonance (b) portions, in the Rayleigh region $k_d a < 1$ (Gaunaurd *et al.*, 1979).

Eq. (129) in the form

$$\Gamma_0/2\omega_0 = mf_0 + I, \tag{131}$$

where

$$I = (c_s/c_d)\eta^{1/2}, \qquad m = \pi N/\eta, \tag{132}$$

$$\eta = 1 + 3\rho_f c_f^2/4\rho c_s^2, \tag{133}$$

and N is defined in Eq. (78). If Eq. (131) is plotted versus resonance frequency f_0, the result is a straight line of slope m and intercept I. This procedure can be used to determine the shear viscosity parameter N of a rubber substance by means of the resonance theory (Gaunaurd *et al.*, 1978).

We start the numerical discussion of this section with a picture of the giant monopole resonance. A plot of $|G|$ of Eq. (115), i.e., the monopole contribution to the scattering amplitude for air-filled cavities in rubber, is displayed in Fig. 38a versus $k_d a$. Subtraction of the pure background in Fig. 38c, yields the resonances, Fig. 38b. The large spike present in the background plot (c) is the "giant monopole resonance" of air-filled cavities in rubber (Gaunaurd *et al.*, 1978; Gaunaurd and Überall, 1979b). Note that this misnamed *resonance* is present in the *background* graph. Equation (115) is also plotted logarithmically versus $k_d a$ in Fig. 39 for an air-filled cavity in absorbing rubber. The absorbing level is controlled by the parameter values $F = 2.218 \times 10^9$ dyn/cm^2, and $F_1 = 10^7$ dyn/cm^2. Other levels and substances were also examined. Figure 39a corresponds to the composite monopole contribution, which is then separated into its resonance part (b) and its background part (c). Other multipole contributions can be also analyzed, in the presence or absence of mode conversion. The main conclusion that emerges from the multipole analysis is the dominance of the monopole contribution, which is several orders of magnitude above that of all other multipoles. When summed amplitudes are displayed adding many multipole contributions, the overall curves have the same shape as the monopole contribution. The higher-order multipole contributions appear as small wiggles superimposed on top of the monopole result.

Equation (131) is plotted in Fig. 40, and is seen to compare well with data available in the literature (Meyer *et al.*, 1958). The resonance widths are linearly proportional to the resonance frequency of cavities in rubber, and this fact was used to develop a process to determine the shear absorption parameter of rubbers entirely based on the Resonance Theory (Gaunaurd *et al.*, 1978; Gaunaurd, Scharnhorst, and Überall, U.S. Navy Pat. Pend.)

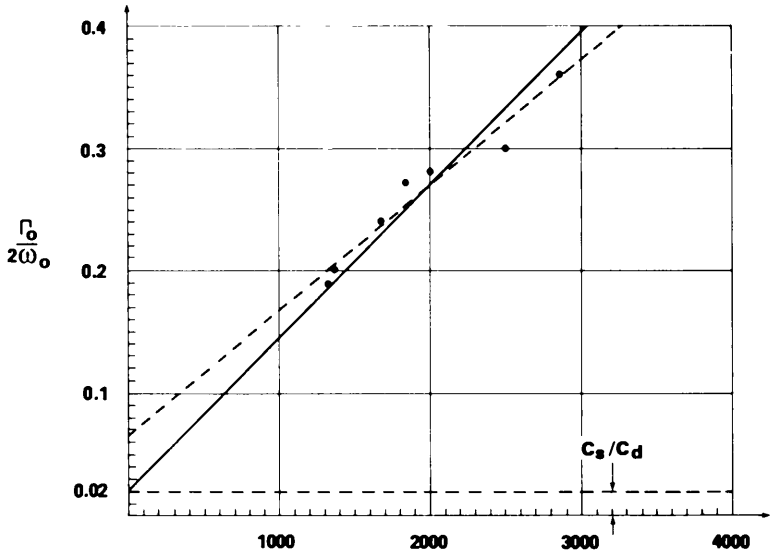


FIG. 40. Relative resonance half-widths versus resonance frequencies for gas-filled spherical cavities of various sizes in FJ-95 rubber, as found from our theory and from a least squares (dashed) and Chebychev (solid) fit of the experimental points of Meyer *et al.* (1958). The horizontal dotted line is the prediction of Meyer *et al.* (1958) for evacuated cavities in lossless rubbers (Gaunaurd *et al.*, 1978).

B. Cylindrical Fluid-Filled Cavities

A resonance analysis similar to that described in the foregoing has also been carried out for the scattering of elastic waves from cylindrical fluid-filled cavities (Haug *et al.*, 1978; Solomon, 1980; Solomon and Überall, 1981). Previous solutions of this problem are due to White (1958), with corrections by Lewis and Kraft (1974) and Flax and Neubauer (1978). Numerical calculations have indicated the presence of resonancelike phenomena in the scattered wave amplitudes when plotted versus the incident wave number. These resonances have been shown to coincide with the eigenfrequencies of the (damped) vibrations of the fluid contained in the elastic-walled cavity (Mow and Workman, 1966; Peck and Miklowitz, 1969; Pao and Mow, 1976).

A plane p wave incident along the x axis on the cylindrical cavity (of radius a, and with axis parallel to z) is given by the scalar potential expanded in cylindrical coordinates:

$$\phi_{\text{inc}} = \phi_0 \sum_{n=0}^{\infty} \varepsilon i^n J_n(k_p r) \cos n\theta, \tag{134a}$$

where $k_p = \omega/c_p$ is the longitudinal wave number. The scattered wave may

be longitudinal,

$$\phi_{sc} = \sum_{n=0}^{\infty} \varepsilon_n i^n A_n H_n^{(1)}(k_p r) \cos n\theta, \tag{134b}$$

or transverse and described by the z component of the vector potential

$$\psi_{sc} = \sum_{n=0}^{\infty} \varepsilon_n i^n B_n H_n^{(1)}(k_s r) \sin n\theta, \tag{134c}$$

with a shear wave number $k_s = \omega/c_s$. A standing compressional wave is excited in the cavity fluid:

$$\phi_F = \sum_{n=0}^{\infty} \varepsilon_n i^n C_n J_n(k_F r) \cos n\theta, \tag{134d}$$

with a fluid wave number k_F.

A plane incident s wave

$$\psi_{inc} = \psi_0 \sum_{n=0}^{\infty} \varepsilon_n i^n J_n(k_s r) \cos n\theta \tag{135a}$$

gives rise to a scattered p wave

$$\phi_{sc} = \sum_{n=0}^{\infty} \varepsilon_n i^n F_n H_n^{(1)}(k_p r) \sin n\theta, \tag{135b}$$

or to a scattered s wave

$$\psi_s = \sum_{n=0}^{\infty} \varepsilon_n i^n G_n H_n^{(1)}(k_s r) \cos n\theta, \tag{135c}$$

and to a standing wave in the cavity fluid,

$$\phi_F = \sum_{n=0}^{\infty} \varepsilon_n i^n R_n J_n(k_F r) \sin n\theta. \tag{135d}$$

The boundary conditions on the cavity will determine the coefficients A_n, B_n, C_n, F_n, G_n, and R_n, which are given in the literature (Solomon, 1980; Solomon and Überall, 1981). Defining the far-field scattering amplitudes f^{ij} as the coefficients of $r^{-1/2} \exp(ik_{p,s} r)$ in Eqs. (134) and (135), we may write

$$f^{pp}(\theta) = \sum_{n=0}^{\infty} f_n^{pp}(\theta) = (2\pi i k_p)^{-1/2} \sum_{n=0}^{\infty} \varepsilon_n (S_n^{pp} - 1) \cos n\theta, \tag{136a}$$

$$f^{ps}(\theta) = \sum_{n=0}^{\infty} f_n^{ps}(\theta) = (2\pi i k_s)^{-1/2} \sum_{n=0}^{\infty} \varepsilon_n S_n^{ps} \sin n\theta, \tag{136b}$$

$$f^{sp}(\theta) = \sum_{n=0}^{\infty} f_n^{sp}(\theta) = (2\pi i k_p)^{-1/2} \sum_{n=0}^{\infty} \varepsilon_n S_n^{sp} \sin n\theta, \tag{136c}$$

$$f^{ss}(\theta) = \sum_{n=0}^{\infty} f_n^{ss}(\theta) = (2\pi i k_s)^{-1/2} \sum_{n=0}^{\infty} \varepsilon_n (S_n^{ss} - 1) \cos n\theta, \quad (136d)$$

where

$$S_n^{PP} = 1 + 2A_n, \quad (137a)$$

$$S_n^{ps} = 2B_n, \quad (137b)$$

$$S_n^{sp} = -2F_n, \quad (137c)$$

$$S_n^{ss} = 1 + 2G_n. \quad (137d)$$

This defines the four elements of the S matrix, which may be shown to be symmetric and unitary.

The resonance theory for scattering from a cylindrical cavity proceeds

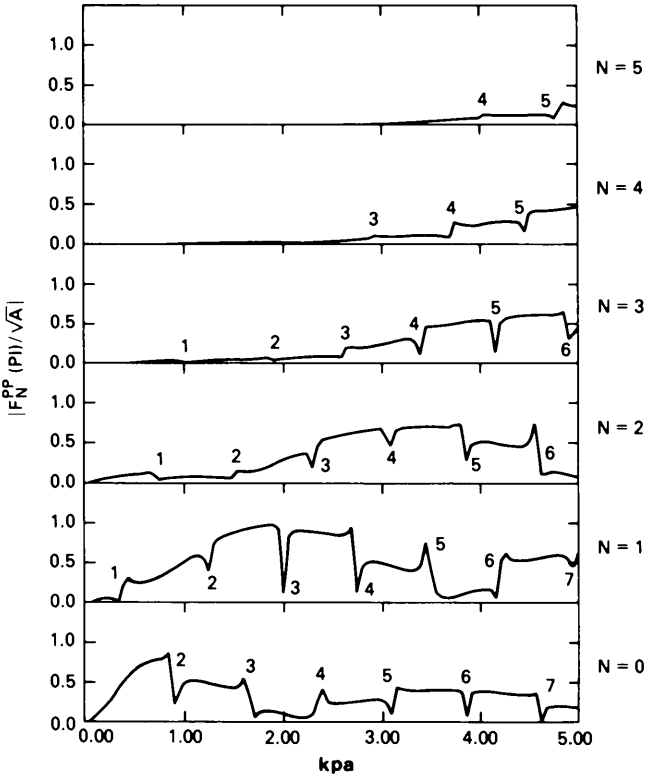


FIG. 41. Modulus of non-mode-converted p-wave partial scattering amplitudes ($n = 0$–5) for a water-filled cylindrical cavity in aluminum plotted versus $k_d a$ (Solomon, 1980).

as in the cases discussed earlier, with the results (e.g., for p wave incidence):

$$S_n^{PP} - 1 = e^{2i\xi_n}\left(\sum_{l=1}^{\infty} \frac{M_{nl}}{x_p - x_p^{nl} + \frac{1}{2}i\Gamma_n^l} + 2ie^{-i\xi_n}\sin\xi_n\right), \tag{138a}$$

$$S_n^{PS} = e^{2i\xi_n}\left(\sum_{l=1}^{\infty} \frac{\tilde{M}_{nl}}{x_p - x_p^{nl} + \frac{1}{2}i\Gamma_n^l} + 1\right), \tag{138b}$$

where $x_p = k_p a$, $\exp(2i\xi_n)$ is the empty-cavity S_n^{PP} function, x_p^{nl} the eigenfrequency of the fluid-filled cavity obtained from a characteristic equation

$$F_n(x_p^{nl}) = \Delta_n \tag{139a}$$

where

$$F_n(x_p) = (\rho/\rho_F)x_F J_N'(x_F)/J_n(x_F). \tag{139b}$$

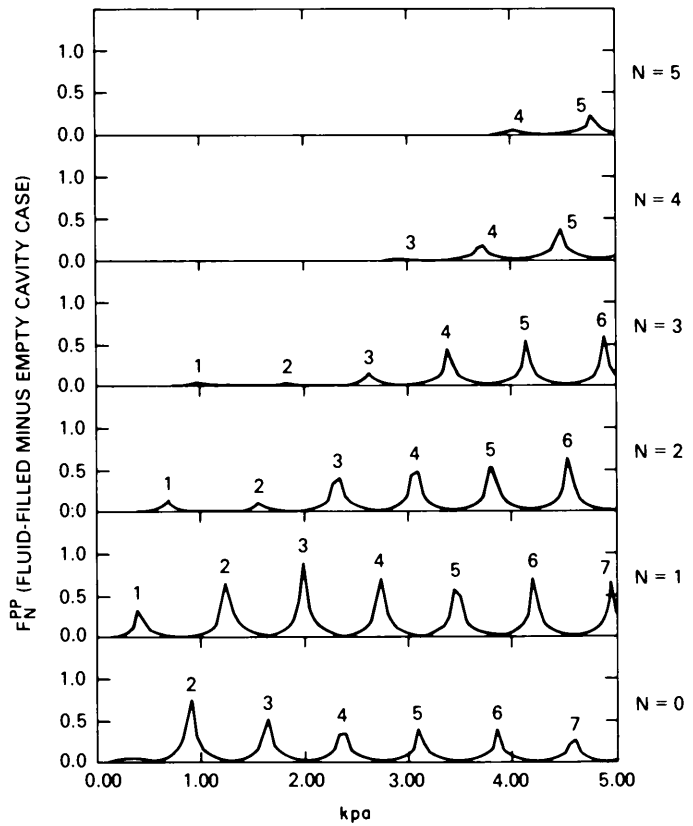


FIG. 42. Same as Fig. 41, after subtraction of soft background (Solomon, 1980).

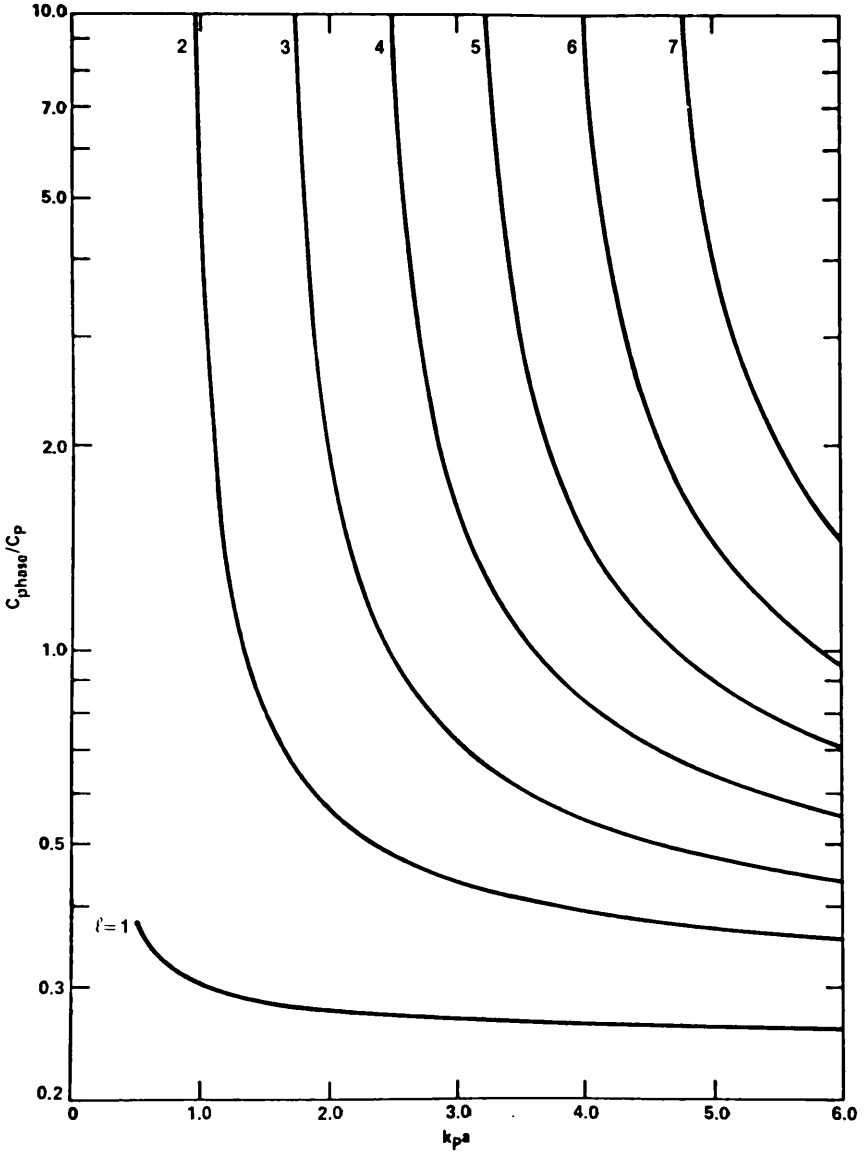


FIG. 43. Dispersion curves for the surface waves on a water-filled cylindrical cavity in aluminum (Solomon, 1980).

Here, $x_f = k_F a$, ρ_F and ρ are the fluid and ambient medium densities, respectively, and Δ_n, M_{nl}, $\tilde{M}_{nl}$ are given in Haug *et al.* (1978). The linearization

$$F_n(x_p) \cong \Delta_n + \beta_{nl}(x_p - x_p^{nl}) \tag{139c}$$

is made, and the resonance widths are found as

$$\Gamma_n^l = -2s_n/\beta_{nl}, \tag{139d}$$

using $z_n^{(1)} = \Delta_n + is_n$, where $z_n^{(i)}$ is given in Haug *et al.* (1978). Equations (138) show that the scattering amplitudes consist of resonance and background contributions.

Figure 41 presents, for a water-filled cylindrical cavity in aluminum, the modulus of the pp scattering amplitude for the partial waves $n = 0$–5, plotted versus $k_p a$, and Fig. 42 shows the amplitude after subtraction of the empty-cavity background, exhibiting the resonances of the fluid filler. As before, the resonance families which shift to higher frequencies from one partial wave to the next can be interpreted in terms of surface waves whose phase matching after repeated circumnavigations gives rise to the resonances. In Fig. 43, the dispersion curves for these circumferential waves are shown. They are labeled by the integer l, which characterizes the lth surface wave, and correspond again to a water-filled cylindrical cavity in aluminum. Analogous dispersion curves for the creeping waves around a *finite*-length fluid cylinder in the acoustic case have recently been obtained by Gaunaurd *et al.* (1980a).

C. Solid Inclusions

The resonance theory of elastic-wave scattering from solid inclusions will now be illustrated by the example of scattering of longitudinal waves from solid spherical inclusions (Flax and Überall, 1980). Previous investigations of this problem (Ying and Truell, 1956; Einspruch *et al.*, 1960; Johnson and Truell, 1965; McBride and Kraft, 1972) have furnished expressions for the scattering cross sections which, upon numerical evaluation, exhibited resonance features as a function of frequency. No detailed analysis of the resonance phenomena had been attempted at that time, however.

1. *Theory*

The center of an elastic sphere of radius a is taken as the origin of a spherical coordinate system (r, θ, ϕ). The longitudinal and shear speeds and the density are designated by c_p, c_s, and ρ, respectively, for the elastic matrix in which the obstacle is imbedded, and by v_p, v_s, and ρ_q for the spherical obstacle itself.

We shall assume that the incoming wave is a plane longitudinal wave which is simple harmonic propagating in the direction of the z axis; it can be expanded in spherical coordinates. As a result of the scattering and mode conversion processes, both longitudinal (Φ) and shear-type (Ψ) scattered waves are present, i.e. (for time dependence exp $i\omega t$),

$$\Phi_{sc} = \sum_{n=0}^{\infty} (2n+1)(-i)^n A_n h_n^{(2)}(k_p r) P_n(\cos\theta) \tag{140a}$$

(longitudinal), and

$$\Psi_{sc} = \sum_{n=0}^{\infty} (2n+1)(-i)^n B_n h_n^{(2)}(k_s r) P_n^1(\cos\theta) \tag{140b}$$

(shear). The wave numbers are here $k_p = \omega/c_p$ and $k_s = \omega/c_s$, and P_n, P_n^1 are the ordinary and associated Legendre polynomials. The coefficients A_n and B_n have to be determined from the boundary conditions at $r = a$.

Inside the scatterer, the refracted waves (labeled by a subscript q) are represented by

$$\Phi_q = \sum_{n=0}^{\infty} (2n+1)(-i)^n C_n j_n(q_p r) P_n(\cos\theta) \tag{140c}$$

and

$$\Psi_q = \sum_{n=0}^{\infty} (2n+1)(-i)^n D_n j_n(q_s r) P_n^1(\cos\theta), \tag{140d}$$

where the wave numbers are $q_p = \omega/v_p$ and $q_s = \omega/v_s$.

There are four separate boundary conditions at $r = a$ that must be satisfied by the solution to this problem. They are sufficient to specify the four sets of unknown coefficients A_n, B_n, C_n, and D_n. These coefficients are given as the quotients of two fourth-order determinants. Details of setting up these boundary conditions as well as obtaining the matrix elements for these coefficients are given in the literature (Flax and Überall, 1980), and we shall here simply state the results.

The quantity of physical interest is the scattered flux dF as obtained, e.g., by Ying and Truell (1956), and is given by

$$dF = \frac{1}{4} i\rho\omega^2 \, d\Omega \left\{ \left| \sum_{n=0}^{\infty} f_n^{pp}(\theta) \right|^2 + \left| \sum_{n=0}^{\infty} f_n^{ps}(\theta) \right|^2 \right\}, \tag{141}$$

where

$$a^{-1/2} f_n^{pp}(\theta) = (2n+1) Z_p^{-1/2} A_n P_n(\cos\theta) \tag{142a}$$

and

$$a^{-1/2} f_n^{ps}(\theta) = (2n+1) Z_s^{-1/2} B_n P_n^1(\cos\theta). \tag{142b}$$

Here, $Z_p = k_p a$, $Z_s = k_s a$, and $d\Omega$ is the element of solid angle.

The resonance theory of elastic wave scattering has been described in detail above, so that we shall only show the results here.

Two S-matrix elements are introduced by

$$S_n^{pp} = 1 + 2A_n, \tag{143a}$$

$$S_n^{ps} = 2B_n. \tag{143b}$$

The scattering amplitude of Eq. (140a) becomes, in terms of the resonance formulation,

$$\frac{1}{2}(S_n^{pp} - 1) = \tfrac{1}{2} e^{2i\xi_n^s} \left\{ \sum_{l=1}^{\infty} \frac{M_{nl}}{Z_p - Z_p^{nl} + \frac{1}{2} i\Gamma_{nl}} + 2ie^{-i\xi_n^s} \sin \xi_n^s \right\}, \tag{144}$$

where $\exp(2i\xi_n^s)$ is the S function for pp scattering from a soft sphere,

$$M_{nl} = (\hat{G}_n - G_n)/\mathrm{Re}\, G_n'(Z_p^{nl}), \tag{145a}$$

and the resonance widths are

$$\Gamma_{nl} = 2\,\mathrm{Im}\, G_n(Z_p^{nl})/\mathrm{Re}\, G_n'(Z_p^{nl}). \tag{145b}$$

The quantities $\hat{G}_n$ and G_n comprise matrix elements given by Flax and Überall (1980). This shows that the scattering amplitude consists of a nonresonant background term [the second term in Eq. (144)], which is due to the scattering of p waves from a soft sphere, with a series of superimposed resonances (first term in the bracket), which will interfere with the background amplitude. A corresponding expression may be derived for B_n of Eq. (140b) and (143b), i.e., the mode-converted scattering amplitude

$$\frac{1}{2} S_n^{ps} = \tfrac{1}{2} e^{2i\xi_n^s} \left\{ \sum_{l=1}^{\infty} \frac{\tilde{M}_{nl}}{Z_p - Z_p^{nl} + \frac{1}{2} i\Gamma_{nl}} + 1 \right\}. \tag{146}$$

2. *Results*

Scattering amplitudes have been evaluated numerically for the case of (a) a lucite sphere and (b) an iron sphere, both imbedded in an aluminum matrix (Flax and Überall, 1980). Due to the much smaller impedances of the lucite sphere ($\rho_q v_p, \rho_q v_s$) as compared to those of the aluminum matrix ($\rho c_p, \rho c_s$), it is expected that the background resembles closely that of scattering from an empty cavity in aluminum. We therefore plot in Figs. 44

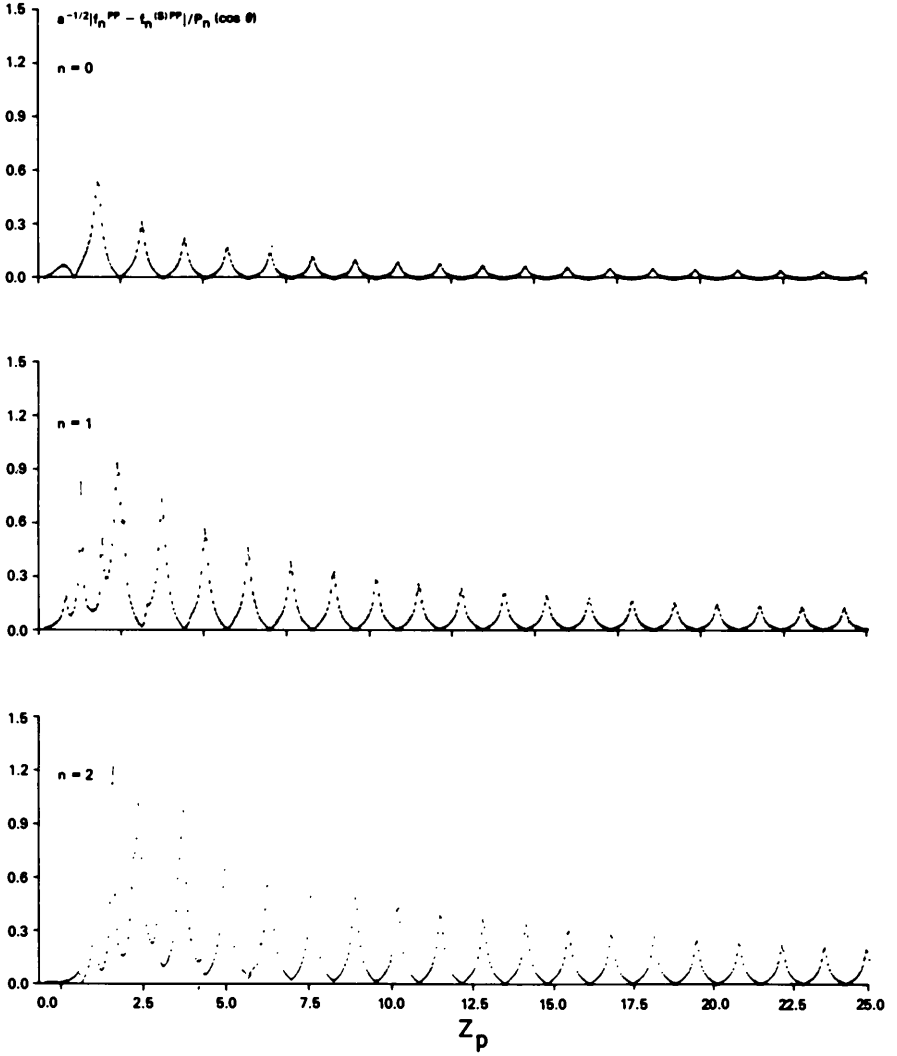


FIG. 44. Plot of $a^{-1/2}\,|f_n^{PP} - f_n^{(s)PP}|/P_n(\cos\theta)$ versus Z_p (where $f_n^{(s)}$ is the empty-cavity scattering amplitude) for the partial waves $n = 0$, 1, and 2, case of a lucite sphere in aluminum (Flax and Überall, 1980).

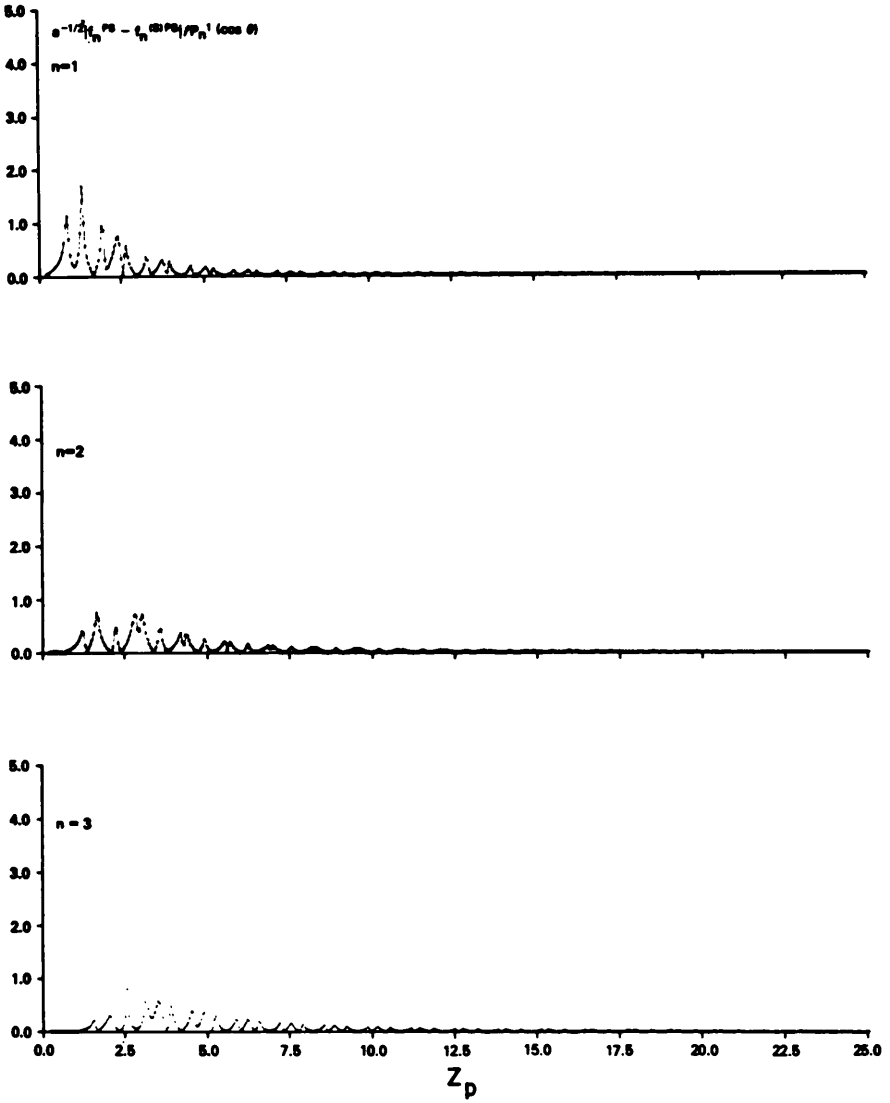


FIG. 45. Plot of $a^{-1/2}|f_n^{ps} - f_n^{(s)\,ps}|/P_n^1(\cos\theta)$ versus Z_p (where $f_n^{(s)}$ is the empty-cavity scattering amplitude) for the partial waves $n = 0$, 1, and 2, case of lucite sphere in aluminum (Flax and Uberall, 1980).

and 45 the quantities

$$a^{-1/2}\left|f_n^{pp} - f_n^{(s)pp}\right|/P_n, \qquad a^{-1/2}\left|f_n^{ps} - f_n^{(s)ps}\right|/P_n^1 \tag{147}$$

versus $k_p a = Z_p$ of aluminum. The "soft" background amplitudes $f_n^{(s)pp}$ or $f_n^{(s)ps}$ obtained when $\rho_q \to 0$ were subtracted out of the total partial wave amplitudes. The resonance frequencies are the same in both cases, but it is seen that two families or resonances are present, and only one of these is being excited in p→p scattering while both families are excited in p→s scattering.

Similar numerical results were obtained (not shown here) for case (b), iron in aluminum. Due to the density of the inclusion being larger than that of the matrix, the background is now expected to correspond to scattering from a rigid sphere ($\rho/\rho_q \to 0$). In fact, if this background is subtracted from f_n^{pp} or f_n^{ps}, we find that the pure resonance amplitudes are indeed obtained in this way.

From Figs. 44 and 45, and corresponding ones for case (b), the resonance frequencies may be read off and tabulated. They may serve to obtain the dispersion curves of circumferential waves propagating over the surface of the inclusion, which have been shown to be generated during the scattering process, and to provide an interpretation for the existence of the resonances.

The resonance frequencies Z_p^{nl} in the resonance expression for the S matrix, Eq. (144), are functions of the mode number n, which we may consider a continuous variable. The phase velocity of the lth surface wave at resonance is given by (Flax and Überall, 1980)

$$c_l^{(n)}(Z_p) = Z_p^{nl} c_p / \left(n + \tfrac{1}{2}\right); \tag{148}$$

the denominator $n + \frac{1}{2}$ is a direct consequence of the spherical geometry. This implies a wavelength of the surface wave

$$\lambda_l(Z_p) = 2\pi a / \left(n + \tfrac{1}{2}\right), \tag{149}$$

so that at resonance, $n + \frac{1}{2}$ wavelengths span the circumference of the spherical inclusion.

Using the resonance frequencies that appear in Figs. 44 and 45 (as well as in corresponding graphs for higher partial waves), we may obtain dispersion curves of the surface waves from Eq. (148). These are shown in Fig. 46 (for lucite in aluminum) and Fig. 47 (for iron in aluminum), where the quantity c_l/c_p is plotted versus Z_p for all the surface waves labeled by l. From these dispersion curves, we see that the surface waves on an elastic spherical inclusion imbedded in an elastic medium fall into two classes (represented in Figs. 46 and 47 by solid and dashed curves, respectively), which presumably correspond to longitudinal and shear-type surface waves.

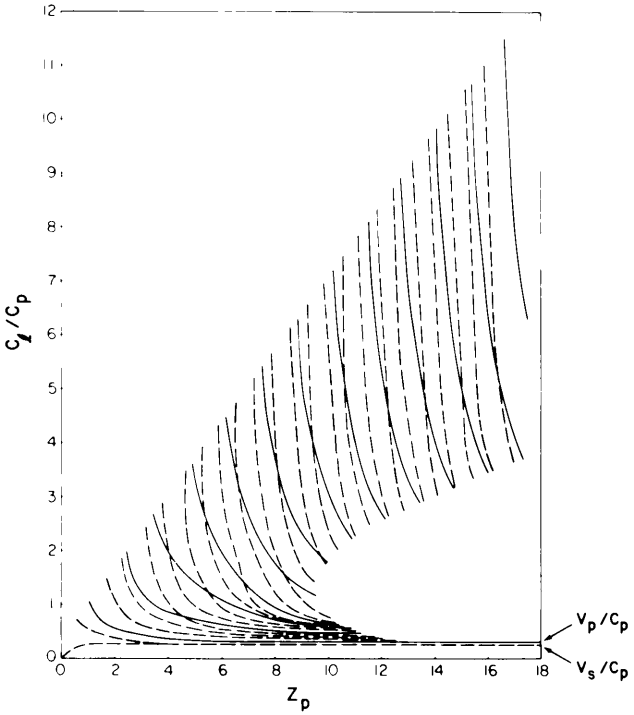


FIG. 46. Dispersion curves for the two families of surface waves on a lucite sphere in aluminum. Arrows on the right-hand side represent the values $v_p/c_p = 0.408$ and $v_s/c_p = 0.201$ (Flax and Überall, 1980).

The distinction between these two classes of curves, when producing the plots, has been greatly facilitated by the fact that in p→p scattering (Fig. 44) only one class of surface waves seems to have been excited, while in p→s scattering (Fig. 45), both are excited. For small values of Z_p, however, the distinction in the excitation mechanism, and hence in the dispersion curves, is no longer so clear-cut.

In the high-frequency limit $Z_p \to \infty$, the dispersion curves approach horizontal asymptotic values, while physically, the boundary of the inclusion tends toward the flat boundary between two infinite elastic half-spaces. In the latter case, the speed of surface waves on the boundary satisfies the Stoneley equation (Ewing *et al.*, 1957):

$$c^4\left[(\rho-\rho_q)^2(\rho M_q+\rho_q M)(\rho N_q+\rho_q N)\right] + 2Rc^2(\rho M_q N_q-\rho_q MN-\rho+\rho_q) + R^2(MN-1)(M_q N_q-1)=0, \tag{150}$$

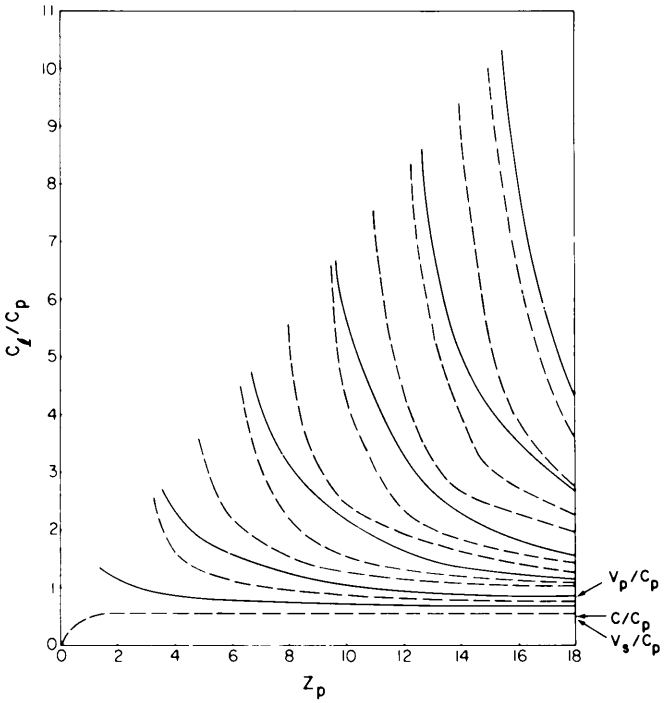


FIG. 47. Dispersion curves for the two families of surface waves on an iron sphere in aluminum. Arrows on the right-hand side represent the values $v_p/c_p = 0.8815$, $c/c_p = 0.504$ (the Stoneley wave speed), and $v_s/c_p = 0.472$ (Flax and Überall, 1980).

where

$$
\begin{aligned}
M &= \left(1 - c^2/c_p^2\right)^{1/2}, \\
M_q &= \left(1 - c^2/v_p^2\right)^{1/2}, \\
N &= \left(1 - c^2/c_s^2\right)^{1/2}, \\
N_q &= \left(1 - c^2/v_s^2\right)^{1/2}, \\
R &= 2\left(\rho c_s^2 - \rho_q v_s^2\right).
\end{aligned}
\tag{151}
$$

We have solved this equation numerically, and have found a Stoneley root $c/c_p = 0.504$ for the case of iron in aluminum; no root was found for the case of lucite in aluminum. [It is well-known (Ewing *et al.*, 1957) that Stoneley wave solutions do not always exist on the flat boundary between two elastic half spaces.]

The value $c/c_p = 0.504$ is indicated by an arrow on Fig. 47, as are the values of v_p/c_p and v_s/c_p in Fig. 46 and 47. It was found that v_s/c_p represents an asymptote for the lowest dispersion curve while c/c_p does not; the latter quantity may, however, constitute an asymptote for higher-order dispersion curves.

IV. The Inverse Scattering Problem

Inverse scattering means many things to many people. In quantum mechanics, inverse scattering methods try to determine the potential if the scattering cross section is known. Geophysicists try to determine the distribution of density and of seismic velocity with depth, from observed reflection responses (Gjevik *et al.*, 1976). In general, inverse scattering methods try to obtain information about a source distribution, a scattering obstacle or inhomogeneity of a medium from remotely sensed field data. The applications of work of this type make up quite a lengthy list (Bleistein and Bojarski, 1975), and most formulations lead to Fredholm integral equations of the first kind. The large number of papers on this subject presented at conferences (Sáenz, 1980) demonstrate that this is quite an active area of research. In this section, we shall discuss a single application of the resonance theory of (visco)elastic wave-scattering that solves an inverse scattering problem of elastodynamics. The procedure sets the pace for extensions and generalizations to more general situations, as well as for direct transfer to other neighboring scientific areas.

The amplitudes of backscattered waves returned by obstacles in viscoelastic solids, when plotted as a function of frequency, exhibit so many rapid oscillations and complicated features that until very recently it was not possible to extract the physical information contained in them about the nature of the obstacle. We can analyze these backscattered "echoes" using the resonance theory of viscoelastic wave- scattering from cavities in solids, and obtain from them, for a given shape of the cavity, the material composition of the filler substance (assumed fluid in this example, for simplicity).

When a (spherical) cavity filler is sent into oscillation by elastic (say, compressional) incident waves, a set of modal resonances (fundamental and overtones) are excited in it, that characterize the filler as if it were its signature. The way these resonances can be used for material discrimination purposes resembles the way chemical elements are identified from their optical spectra (Gaunaurd and Überall, 1979c, 1980, 1980a).

We obtained earlier [i.e., Eqs. (83)] expressions for the scattering amplitudes of p waves returned by a spherical fluid-filled cavity in a solid under p-wave incidence. Figure 48 shows the plot of the modulus of the

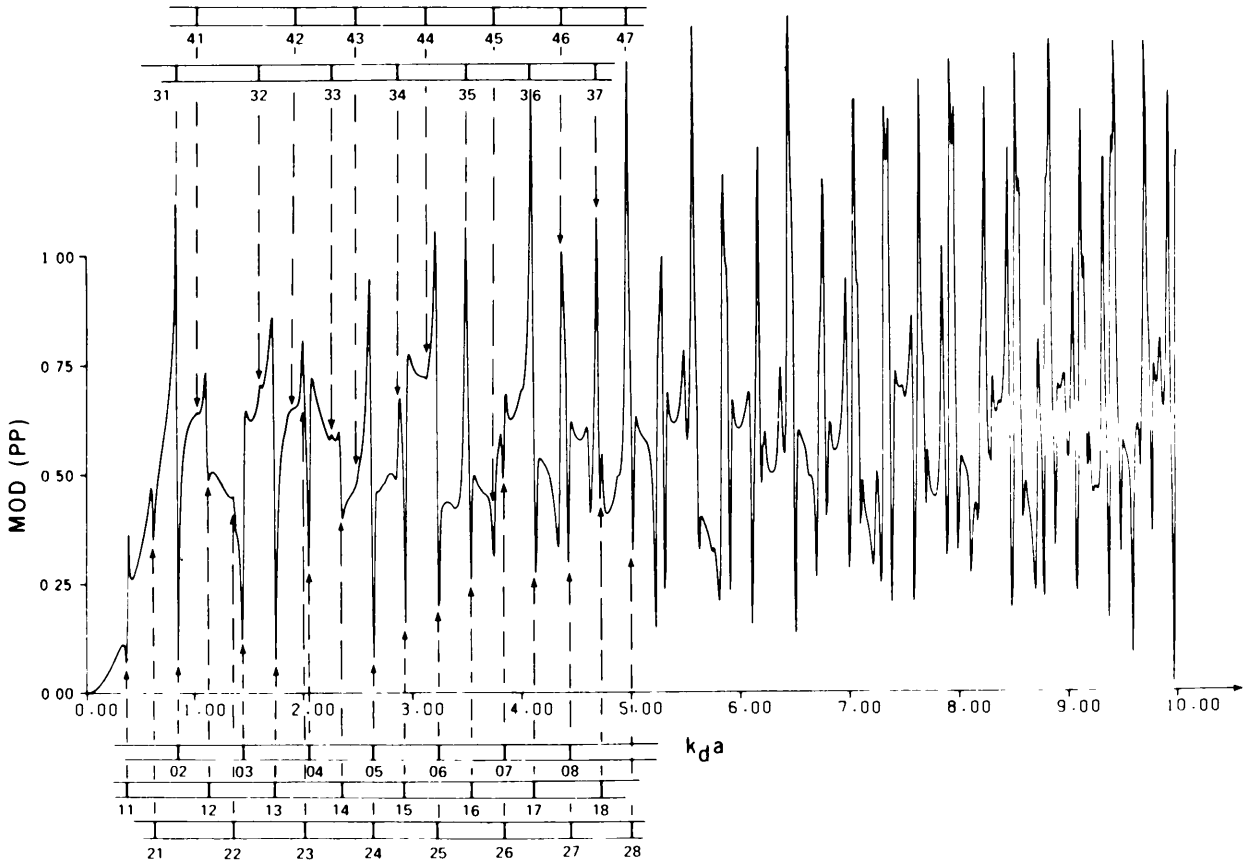


FIG. 48. Modulus of summed p→p backscattered amplitude plotted versus $k_d a$ (adding 15 terms in the mode series) for an alcohol-filled cavity in aluminum. Some of the rapid oscillations in this plot are accounted for (as indicated) by the resonances shown in the following figures. From Gaunaurd and Überall (1979c). Copyright 1979 by the American Association for the Advancement of Science.

summed amplitude in the backscattering direction $\theta = \pi$ (i.e., $|f^{PP}(\pi)/a|$) for a cavity filled with ethyl–alcohol in an aluminum matrix. This is the "echo" containing the rapid oscillations and complex features mentioned above. Fifteen partial waves were added to produce that plot, which for the range $0 \leqslant k_d a \leqslant 10$ of Fig. 48, yields enough accuracy. This is the amplitude as a sensor would record it. An analysis of the first partial-wave contributions making up the sum is shown in Fig. 49. Each mode can be split into a background (b) and resonances (c). An index l labels each resonance of each mode n. Overlaying the resonances which we just isolated, on top of the summed amplitude of Fig. 48, identifies each extremum in that plot with the sets of modal resonances contained in the partial waves. The resonance responsible for each wiggle of Fig. 48 is labeled by indices (n, l), and the graph shows over 30 wiggles uniquely identified with modal resonances in this fashion. The location of *all* the resonance spikes in fact identifies the ratio c_f/c_d of the filler-to-matrix wavespeeds. The filler's sound-speed c_f can be found from the spacing Δ between any two high-order consecutive overtones shown in Fig. 49. For spherical cavities in solids, we have shown (Gaunaurd and Überall, 1979, 1979c, 1980) that for $l \gg 1$, $\Delta \cong \pi(c_f/c_d)$. For alcohol in aluminum, that relation gives $\Delta \cong 0.59$, just as observed in Fig. 49. Thus, knowing c_d for the matrix, and the spacing Δ between consecutive high-order modal resonances, determines the sound speed c_f of the filler, and vice versa.

The density ratio of filler-to-matrix materials (i.e., ρ/ρ_f) is found from the resonance *widths*. Equation (93) for the widths leads to the relation (Gaunaurd and Überall, 1978, 1980):

$$F_n\left(\frac{c_d}{c_f}x\right) = \operatorname{Re} z_1(x) - \frac{\operatorname{Im} z_1(x)}{\Gamma_{nl}/2}(x - x_{nl}), \tag{152}$$

where z_1 and F_n are given in Eqs. (85a) and (86), and x_{nl} are the roots of the characteristic Eq. (91).

Equation (152) can be solved for the density ratio ρ_f/ρ contained in the expression for F_n. The result can be evaluated at a point one half-width below any resonance peak (i.e., $x = x_{nl} - \Gamma_{nl}/2 \equiv X_{nl}$) and then expanded for $x(\equiv k_d a) \gg 1$. The result is

$$\frac{\rho_f}{\rho} \underset{x \gg 1}{\rightarrow} \left\{ -\frac{\pi^2}{\Delta^2}x + \left[\frac{\pi^2}{\Delta^2}\left(l + \frac{n}{2} - \frac{1}{2}\right)\Delta - \frac{1}{\left(l + \frac{1}{2}n - \frac{1}{2}\right)\Delta} \right] \right\}_{x = X_{nl}}, \tag{153}$$

which holds for integer n and $\Gamma_{nl} < 2\Delta$, so that consecutive resonances do not overlap. The evaluation indicated above yields

$$\frac{\rho_f}{\rho} \underset{x \gg 1}{\rightarrow} -\frac{1}{\left(l + \frac{1}{2}n - \frac{1}{2}\right)\Delta} + \frac{\pi^2}{\Delta^2}\frac{\Gamma_{nl}}{2} + \cdots, \qquad n, l \text{ integers} \tag{154}$$

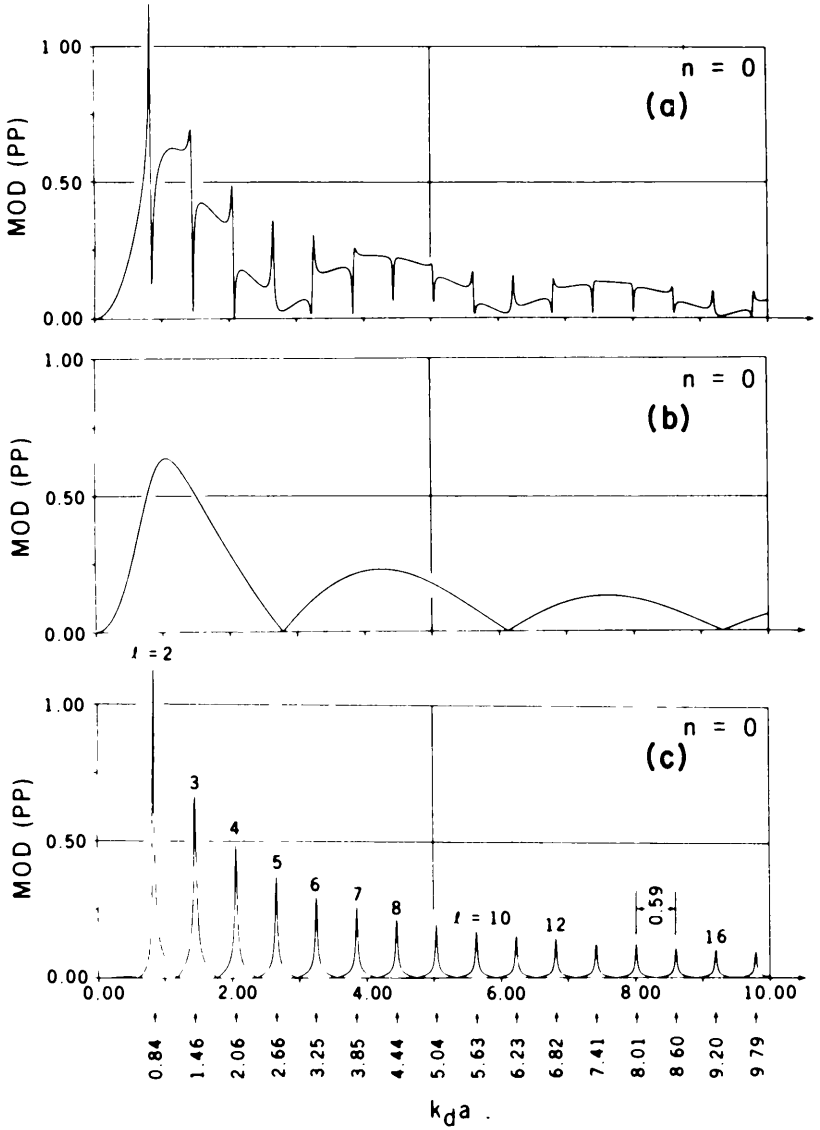


FIG. 49. The zeroth (i.e., $n = 0$) modal contribution (a) of the $|f^{PP}/a|$ amplitude of an alcohol-filled spherical cavity in an aluminum matrix, and its separation into smooth background (b) and isolated resonances (c). The resonance locations in this mode ($n = 0$) are labeled and marked. The asymptotic spacing is seen to $\Delta \simeq 0.59$. From Gaunaurd and Überall (1979c). Copyright 1979 by the American Association for the Advancement of Science.

In the high-frequency limit, this gives the filler-to-matrix density ratio in terms of the uniform asymptotic spacing Δ, and the width Γ_{nl} of any (high-index) overtone l, of any mode n, which are either all known or previously determined quantities.

For ethyl alcohol ($\rho_f = 0.79$ g/cm^3, $c_f = 1.213 \times 10^5$ cm/sec) in aluminum ($\rho = 2.7$ g/cm^3, $c_d = 6.420 \times 10^5$ cm/sec) we find $\Delta \simeq 0.593$, and for the first (i.e., $n = 1$) mode, the width of its tenth overtone ($l = 10$), which occurs at $x_{nl} \simeq l\Delta = 5.93$, is $\Gamma_n \simeq 0.06$. Since this width is $\sqrt{3}$ too large (Gaunaurd and Überall, 1979), we divide it by $\sqrt{3}$, and substitution into Eq. (154) yields $\rho_f/\rho = 0.296$, which contains a negligible error when compared to the true ratio 0.79/2.70. The (fluid) filler is completely identified once its sound speed and density are determined by this asymptotic procedure. We conclude that the sound-speed ratio between filler and matrix is extracted from the spacing between any two consecutive overtones of any mode, and the matrix-to-filler density ratio is extracted from the width of any high-order resonance by means of Eq. (154). Thus, the high-frequency region $x \gg 1$, $l \gg 1$ contains all the information about the material composition of the filler. The filler has been identified from its "echo," and an inverse scattering problem has been solved in a novel fashion. It should also be noted that a determination of the size of water droplets from their acoustic resonances was discussed by George (1980).

V. Resonances in Electromagnetic Scattering

Resonances may appear in scattering processes belonging to many different fields of physics. Besides the acoustic and elastic-wave scattering problems that constitute the main topic of this article, there are also many nuclear and heavy-ion scattering processes that display resonance effects as we pointed out in the Introduction, as do optical and electromagnetic-wave (or radar) scattering phenomena. In order to provide a glimpse of one of these latter problems, and thus to demonstrate the universal nature of scattering resonances throughout various domains of physics, we shall here briefly discuss the case of electromagnetic-wave scattering from a dielectric sphere.

A. Theory

The electromagnetic field is a vector field similar to the elastic-wave field, but it differs from the latter by not possessing any longitudinal component. Its two transverse components have one phase velocity (the speed of light), so that the electromagnetic field resembles an elastic field that contains shear waves only. Nevertheless, two types of resonances

appear, namely, electric or magnetic resonances, which give rise to corresponding types of surface waves with different families of dispersion curves each (although not as drastically different as those of Figs. 46 or 47). Resonances in radar cross sections have been found to be present in many measured and numerically calculated examples (see, e.g., Ruck *et al.*, 1970), and have been made the subject of a recent investigation, including their interpretation in terms of surface waves (Murphy *et al.*, 1980).

For an incident plane electromagnetic wave of propagation constant $k = \omega/c$, i.e.,

$$\begin{aligned} \mathbf{E}_{\text{inc}} &= E_0 \hat{x} e^{-ikz}, \\ \mathbf{H}_{\text{inc}} &= -E_0 \hat{y} e^{-ikz} \end{aligned} \tag{155}$$

[with time dependence $\exp(-i\omega t)$ suppressed], the far field scattered from a sphere of radius a is given by (Ruck *et al.*, 1970)

$$\mathbf{E}_{\text{sc}} = E_0 \frac{e^{ikr}}{kr} \left[\hat{\theta} S_1(\theta) \cos\phi - \hat{\phi} S_2(\theta) \sin\phi \right], \tag{156}$$

with (r, θ, ϕ) the coordinates of the observer, $\hat{\theta}$ and $\hat{\phi}$ the unit vectors in the θ and ϕ directions, and the far-field scattering amplitudes given by

$$S_1(\theta) = \sum_{n=1}^{\infty} (-i)^{n+1} \left[A_n \frac{P_n^1(\cos\theta)}{\sin\theta} + iB_n \frac{dP_n^1(\cos\theta)}{d\theta} \right], \tag{157a}$$

$$S_2(\theta) = \sum_{n=1}^{\infty} (-i)^{n+1} \left[A_n \frac{dP_n^1(\cos\theta)}{d\theta} + iB_n \frac{P_n^1(\cos\theta)}{\sin\theta} \right]. \tag{157b}$$

The coefficients A_n and B_n (Mie coefficients) are

$$A_n = (-i)^n \frac{2n+1}{n(n+1)} \frac{xj_n(x) - iZ_n[xj_n(x)]'}{xh_n^{(1)}(x) - iZ_n[xh_n^{(1)}(x)]'}, \tag{158a}$$

$$B_n = (-i)^{n+1} \frac{2n+1}{n(n+1)} \frac{xj_n(x) - iY_n[xj_n(x)]'}{xh_n^{(1)}(x) - iY_n[xh_n^{(1)}(x)]'}, \tag{158b}$$

where $x = ka$; for the case of a homogeneous dielectric sphere one has the impedances and admittances

$$Z_n = -i\mu \frac{k}{k_1} \frac{k_1 a j_n(k_1 a)}{[k_1 a j_n(k_1 a)]'}, \tag{159a}$$

$$Y_n = -i \frac{1}{\mu} \frac{k_1}{k} \frac{k_1 a j_n(k_1 a)}{[k_1 a j_n(k_1 a)]'}, \tag{159b}$$

where μ is the permeability of the sphere (which henceforth will be set

equal to unity), and $k_1 = \omega/c^{(1)} = n^{(1)}k$ is the wave number in the dielectric (we designate the wave speed in the dielectric by $c^{(1)}$ so that the index of refraction is $n^{(1)} = c/c^{(1)} = \sqrt{\mu\epsilon}$). This is also valid for a lossy dielectric with ϵ (and μ) complex. The limit of a perfectly conducting sphere is obtained by letting $Z_n \to 0$ and $Y_n \to \infty$.

The eigenvibrations of the dielectric sphere are obtained by the usual requirement that an outgoing (scattered) wave be present even in the absence of an incident wave ($E_0 \to 0$). This is the case if the denominators in either A_n or B_n vanish, resulting in the characteristic (complex) equations for the eigenfrequencies:

$$1/(iZ_n) = \left[xh_n^{(1)}(x)\right]'/\left[xh_n^{(1)}(x)\right], \tag{160a}$$

with the corresponding solutions being known as "transverse electric" (or TE) modes, and

$$iY_n = xh_n^{(1)}(x)/\left[xh_n^{(1)}(x)\right]' \tag{160b}$$

("transverse magnetic" or TM modes). For the conducting sphere, these eigenvalue equations reduce to

$$h_n^{(1)}(x) = 0 \qquad \text{(TE modes)}, \tag{160a'}$$

$$\left[xh_n^{(1)}(x)\right]' = 0 \qquad \text{(TM modes)}. \tag{160b'}$$

Equations (160) have been solved numerically by Gastine *et al.* (1967), resulting in complex eigenfrequencies. In many cases (Ruck *et al.*, 1970), however, the resonances appearing in the scattering amplitudes are quite high and narrow; this shows that the imaginary parts of the denominators in A_n and B_n, and hence in Eqs. (160), are quite small, and the resonance frequencies are obtained to a good approximation (the "resonance approximation") by equating the *real parts* of the denominators of A_n and B_n to zero. This leads to the *real* eigenvalue equations

$$1/(iZ_n) = \mathrm{Re}\left\{\left[xh_n^{(1)}(x)\right]'/\left[xh_n^{(1)}(x)\right]\right\} \tag{161a}$$

(TE modes), and

$$iY_n = \mathrm{Re}\left\{xh_n^{(1)}(x)/\left[xh_n^{(1)}(x)\right]'\right\} \tag{161b}$$

(TM modes) where for simplicity we have assumed a lossless dielectric. Equations (161) are much simpler to solve than the complex Eqs. (160) solved by Gastine *et al.* (below, we discuss our solutions) and wherever the resonance approximation holds, they should lead to *real* solutions in good agreement with the real parts of the complex solutions of Eqs. (160). This will be seen below to be actually the case.

Equations (160) were written in this particular form so that in both equations, as seen from Eqs. (159), the left-hand side is of order $\epsilon^{1/2}$ (while the right-hand side is of order unity). The neglect of the imaginary part on the right-hand side, leading to Eqs. (161), is then justified for $\epsilon^{1/2} \gg 1$ ($\epsilon \gtrsim 3$, say), and this condition has to be satisfied for our resonance approximation to be valid.

The resonance denominators of Eqs. (158) may be brought into an explicit resonance form, as follows. Let us denote

$$\left[xh_n^{(1)}(x) \right]' / \left[xh_n^{(1)}(x) \right] \equiv g_n(x) = r_n + i\Delta_n, \tag{162a}$$

$$1/g_n = \bar{r}_n + i\bar{\Delta}_n, \tag{162b}$$

$$1/(iZ_n) = M_n^{\mathrm{TE}}(x), \tag{162c}$$

$$1/(iY_n) = M_n^{\mathrm{TM}}(x), \tag{162d}$$

where M_n^{TE} and M_n^{TM} are real for a lossless dielectric. The denominators of A_n and B_n are, respectively, proportional to

$$D_n^{\mathrm{TE}} = g_n - M_n^{\mathrm{TE}} = r_n - M_n^{\mathrm{TE}} + i\Delta_n \tag{163a}$$

and

$$D_n^{\mathrm{TM}} = (1/g_n) - \left(1/M_n^{\mathrm{TM}}\right) = \bar{r}_n - \left(1/M_n^{\mathrm{TM}}\right) + i\bar{\Delta}_n. \tag{163b}$$

As in Eqs. (161), the eigenfrequencies arise from

$$r_n - M_n^{\mathrm{TE}} = 0, \qquad x = x_{nl}^{\mathrm{TE}} \tag{164a}$$

or

$$\bar{r}_n - \left(1/M_n^{\mathrm{TM}}\right) = 0, \qquad x = x_{nl}^{\mathrm{TM}}, \tag{164b}$$

where the various solutions of these equations, for a given mode number n, are labeled by $l = 1, 2, 3, \ldots$. Expanding around x_{nl}, one has (exemplifying the development by the TE mode whose index we suppress)

$$M_n - r_n \cong (x - x_{nl})(M_n - r_n)'_{x_{nl}} + \cdots, \tag{165}$$

so that the denominator becomes

$$D_n \propto x - x_{nl} + \tfrac{1}{2} i\Gamma_{nl}. \tag{166}$$

This represents a resonance denominator with width

$$\Gamma_{nl} = 2\left[\Delta_n / (r_n - M_n)' \right]_{x_{nl}}, \tag{167}$$

indicating the resonant character of the scattering amplitudes. While Eq. (166) shows the presence of a pole in the complex-x plane at the position

$$\hat{x} = x_{nl} - \tfrac{1}{2} i\Gamma_{nl}, \tag{168}$$

the scattering amplitudes A_n and B_n may actually be considered functions of two variables, namely, x and the mode number n (now taken as a continuous variable), and Eq. (166) may be rewritten as a resonance denominator in the n variable, by expanding x_{nl} about an n value n_l, so chosen that $x_{n_l l} = x$. This development is similar to that given in Sections II, B, 1 and 2, and leads as in that case to a demonstration that during the scattering process, surface waves are generated which circumnavigate the scattering object repeatedly (and, in the case of a dielectric sphere, suffer a phase lag of $\pi/2$ at each of the two convergence points on the sphere). Wavelength λ_l, decay angle θ_l, and phase speed c_l of these waves are given by Eqs. (46), and in general they will extinguish themselves by destructive interference except at the eigenfrequencies, where they build up to a resonance by phase matching during their successive circumnavigations.

B. Numerical Results

We shall now evaluate the dispersion curves, using Eq. (46c), for the velocities of surface waves on a dielectric sphere. These are obtained from a calculation of the eigenfrequencies of the sphere, by an extension of the work of Gastine *et al.* (1967). In the latter reference, the complex solutions of Eqs. (160) were found, of which we here need the real parts only; see Eq. (46c). Instead, we solve the real equations Eqs. (161), which is computationally simpler, and which was found to lead to real solutions which agree with the real parts of Gastine's solutions in all cases where the resonance theory is applicable.

As an example of our results, we show in Fig. 50 the real parts of the eigenfrequencies $(k_1 a)_{nl}^{TE}$ for the lowest TE modes of a dielectric sphere, labeled *nEl*, as obtained in the resonance approximation, and plotted versus the dielectric constant ϵ. For $\epsilon \gtrsim 3$, they agree with Gastine's results wherever the latter are available.

From Fig. 50 and from similar results for the TM modes, and using Eq. (46c), we obtain the dispersion curves of the surface waves that give rise to the TE or TM mode resonances as displayed in Figs. 51 and 52, respectively. These show $c_l/c^{(1)}$ plotted versus $k_1 a$, at $\epsilon = 10$.

In the limit $k_1 a \to \infty$, all the phase velocities approach $c_l \to c^{(1)}$, i.e., the bulk wave speed in the dielectric, which proves that the surface waves propagate in the dielectric material of the sphere, i.e., they are "Rayleigh-type" waves rather than those of "Franz type," which propagate in the space surrounding the sphere (Überall, 1973). The arrows on top of Figs. 51 and 52 indicate the cutoff frequencies of the lth surface waves.

As in the problem of acoustic or elastic-wave scattering, we have thus analyzed the resonances in a similar way for the case of radar scattering,

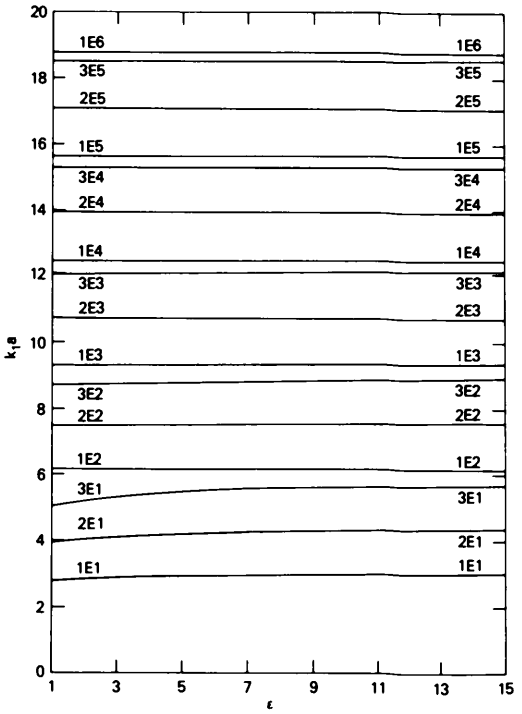


FIG. 50. Real parts of the eigenfrequencies $(k_1a)_{nl}^{TE}$ for the TE modes of a dielectric sphere, labeled *nEl*, obtained in the resonance approximation, and plotted versus the dielectric constant ϵ. [From Murphy *et al.* (1980). © 1980 IEEE.]

and have interpreted them in terms of circumferential waves. The dispersion curves of the latter have been obtained from the resonance frequencies in a very direct fashion for all the cases discussed.

We state in closing that the complete formulation of the resonance scattering theory for the electromagnetic case has been recently given by Gaunaurd *et al.* (1980b). The formulation given in this work is illustrated with the example of a dielectrically coated conducting sphere, and it also shows how to solve the inverse scattering problem for the thickness and dielectric constant of the coating, from the features contained in the returned echo, in a way analogous to that shown in Section IV for elastic waves. The *R*-matrix theory for this problem was formulated by Moser *et al.* (1981).

Additional pertinent work on the electromagnetic resonances and the surface waves associated with them, which travel on helicoidal paths around a *finite*-length cylindrical cavity in a conducting medium, has recently been completed by Subrahmanyam *et al.* (1981). This work

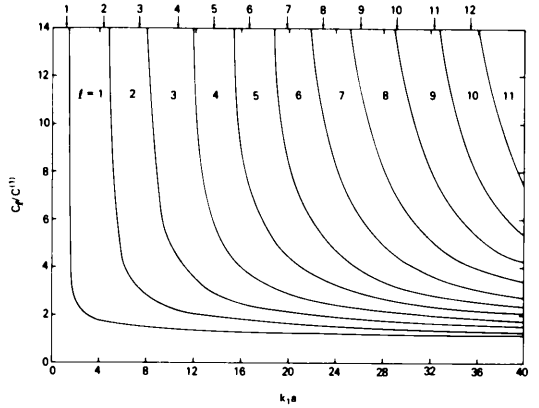


FIG. 51. Dispersion curves for the surface waves on a dielectric sphere, plotted versus k_1a, for the TE modes at $\epsilon = 10$. [From Murphy *et al.* (1980). © 1980 IEEE.]

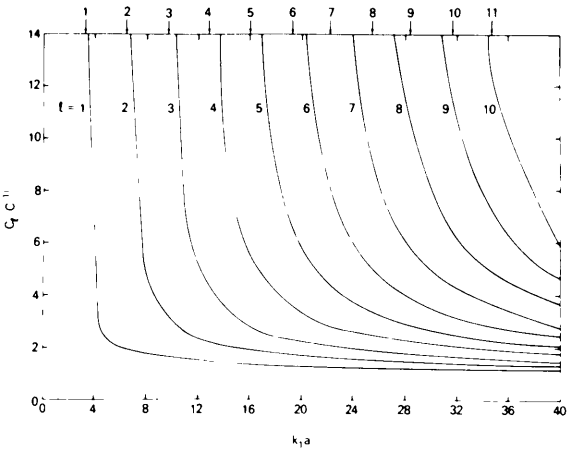


FIG. 52. Same as Fig. 51, for the TM modes. [From Murphy *et al.* (1980). © 1980 IEEE.]

includes the calculation and display of the phase velocities of the surface waves for various length-to-diameter ratios of the cylindrical cavity.

Finally, a discussion of greater length than can be given here, on the general inverse scattering problem and its connection to the acoustic, elastic, and electromagnetic resonances of a target-obstacle whose material composition and/or shape features are to be determined from the echoes they return, has been completed recently by Gaunaurd and Überall (1981a); see also Flax (1981).

ACKNOWLEDGMENTS

The first author (L. F.) would like to thank Dr. L. R. Dragonette for experimental discussions, as well as Dr. John Munson, Dr. Joseph Bucaro, and Dr. Werner G. Neubauer,

all of Naval Research Laboratory, for encouragement and support. The other authors (G. C. G. and H. Ü.) acknowledge the support of the Independent Research Board of the Naval Surface Weapons Center, and the assistance of John Barlow in the preparation of the computed figures. The last author (H. Ü.) further acknowledges the support of Code 421 of the Office of Naval Research concerning the acoustic work presented here, and the support of AIR-310-B, Naval Air Systems Command, regarding the electromagnetic section. He also wishes to thank Miss Reyna Tosta for her most competent typing of the manuscript.

References

Bertoni, H. L., and Tamir, T. (1973). *Appl. Phys.* **2**, 157.

Bleistein, N., and Bojarski, N. (1975). Recently Developed Formulations of the Inverse Scattering Problem in Acoustics and Electromagnetics. Univ. of Denver, Mathematics Dept. Rep. MS-R-7501, Denver, Colorado (1975).

Breitenbach, E. D. (1979). Thesis, The Catholic Univ. of America, Washington, D.C. 20064.

Brekhovskikh, L. M. (1960). "Waves in Layered Media." Academic Press, New York.

Brill, D., and Überall, H. (1971). *J. Acoust. Soc. Am.* **50**, 921.

Brill, D., Gaunaurd, G. C., and Überall, H. (1980). *J. Acoust. Soc. Am.* **67**, 414.

Brill, D., Gaunaurd, G. C., and Überall, H. (1980a). *J. Appl. Phys.* (to be published).

Clay, C. S., and Medwin, H. (1977). "Acoustical Oceanography." Wiley, New York.

Cremer, L., and Heckl, M. (1973). "Structure-Borne Sound." Springer-Verlag, Berlin and New York.

Derem, A. (1979). *Rev. CETHEDEC* **58**, 43–79.

de Santo, J., Sáenz, A. W., and Zachary, W. (Eds.) (1980). *Proc. Symp. Math. Methods Appl. Scattering Theory, Catholic Univ., Washington, D.C.* Springer-Verlag, Berlin and New York.

Devin, Ch. (1959). *J. Acoust. Soc. Am.* **31**, 1654.

Dickey, J. W., Frisk, G. V., and Überall, H. (1976). *J. Acoust. Soc. Am.* **59**, 1339.

Doolittle, D., and Überall, H. (1966). *J. Acoust. Soc. Am.* **39**, 272.

Doolittle, R. D., Überall, H., and Uginčius, P. (1968). *J. Acoust. Soc. Am.* **43**, 1.

Dragonette, L. R. (1978). Thesis, The Catholic Univ. of America, Washington, D. C. 20064; NRL Rep. 8216, Naval Research Laboratory, Washington, D. C. 20375 (September 1978).

Dragonette, L. R. (1979). *J. Acoust. Soc. Am.* **65**, 1570.

Dragonette, L., and Flax, L. (1979a). *Proc. Symp. Math. Meth. Appl. Scattering Theory, Catholic Univ., Washington D.C.* Springer-Verlag, Berlin and New York.

Dragonette, L., and Flax, L. (1979b). *Proc. Symp. Recent Dev. Classical-Wave Scattering: Focus T-Matrix Approach* (V. K. Varadan, ed.). Pergamon, Oxford.

Dragonette, L. R., Vogt, R. H., Flax, L., and Neubauer, W. G. (1974). *J. Acoust. Soc. Am.* **55**, 1130.

Dragún, O., and Überall, H. (1980). *Phys. Lett.* **94B**, 24.

Eckart, J. (Ed.) (1946). "Principles and Applications of Underwater Sound." Summ. Tech. Rep. of Division G, NDRC, Vol. 7, Chapter 28, Eq. 12. Washington, D. C.

Einspruch, N. G., Witterholt, E. J., and Truell, R. (1960). *J. Appl. Phys.* **31**, 806.

Ewing, W. M., Jardetzky, W. S., and Press, F. (1957). "Elastic Waves in Layered Media." McGraw-Hill, New York.

Faran, J. J. (1951). *J. Acoust. Soc. Am.* **23**, 405.

Farhan, A. R., George, J., and Überall, H. (1978). *Nucl. Phys.* **A305**, 189.

Fiorito, R., and Überall, H. (1979). *J. Acoust. Soc. Am.* **65**, 9.

Fiorito, R., Madigosky, W., and Überall, H. (1979). *J. Acoust. Soc. Am.* **66**, 1857.

Flax, L. (1974). *J. Acoust. Soc. Am.* **55**, 1130.

Flax, L. (1975). *J. Acoust. Soc. Am.* **57**, 558.
Flax, L. (1977). *J. Acoust. Soc. Am.* **62**, 1501.
Flax, L. (1981). *J. Acoust. Soc. Am.* (to be published).
Flax, L., and Neubauer, W. (1977). *J. Acoust. Soc. Am.* **61**, 307.
Flax, L., and Neubauer, W. (1978). *J. Acoust. Soc. Am.* **63**, 675.
Flax, L., and Überall, H. (1980). *J. Acoust. Soc. Am.* **67**, 1432.
Flax, L., and Dragonette, L. R., and Überall, H. (1978). *J. Acoust. Soc. Am.* **63**, 723.
Flax, L., Raju, P. K., George, J., and Überall, H. (1980). *J. Acoust. Soc. Am.* **68**, 973.
Frisk, G. V., and Überall, H. (1976). *J. Acoust. Soc. Am.* **59**, 46.
Frisk, G. V., Dickey, J. W., and Überall, H. (1975). *J. Acoust. Soc. Am.* **58**, 996.
Gastine, M., Courtois, L., and Dormann, J. L. (1967). *IEEE Trans.* **MTT-15**, 694.
Gaunaurd, G. C. (1976). Naval Surface Weapons Center, White Oak, Silver Spring, MD 20910, Tech. Rep. 76-20 (# AD-A025, 302).
Gaunaurd, G. C. (1977). *J. Acoust. Soc. Am.* **61**, 360.
Gaunaurd, G. C. (1977a) *J. Acoust. Soc. Am.* **62**, 503.
Gaunaurd, G. C. (1977b) *J. Acoust. Soc. Am.* **62**, 298.
Gaunaurd, G. C. (1980). *Proc. Symp. Math. Methods Appl. Scattering Theory, Catholic Univ., Washington, D. C.* (J. de Santo, A. W. Sáenz, and W. Zachary, eds.). Springer-Verlag, Berlin and New York.
Gaunaurd, G. C., and Überall, H. (1978). *J. Acoust. Soc. Am.* **63**, 1699.
Gaunaurd, G. C., and Überall, H. (1979). *J. Appl. Phys.* **50**, 4692.
Gaunaurd, G. C., and Überall, H. (1979a). *Proc. Symp. Ship-Related Acoust. R&D, 4th, Monterey, California* p. 189.
Gaunaurd, G. C., and Uberall, H. (1979b). *Proc. Symp. Recent Dev. Classical-Wave Scattering: Focus T- Matrix Approach* (V. K. Varadan, ed.). Pergamon, Oxford.
Gaunaurd, G. C., and Überall, H. M. (1979c). *Science* **206**, 61.
Gaunaurd, G. C., and Überall, H. (1980). *Ultrasonics* **18**, 261.
Gaunaurd, G. C., and Überall, H. (1980a). *Nature* **287**, 708.
Gaunaurd, G. C. and Überall, H. (1981). *J. Acoust. Soc. Am.* (to be published).
Gaunaurd, G. C. and Uberall, H. (1981a). *IEEE Trans. Antennas Propag.* **AP-29**, No. 2.
Gaunaurd, G. C., Scharnhorst, K. P., and Überall, H. (1978). *J. Acoust. Soc. Am.* **64**, 1211.
Gaunaurd, G. C., Scharnhorst, K. P., and Überall, H. (1979). *J. Acoust. Soc. Am.* **65**, 573.
Gaunaurd, G. C., Tanglis, E., and Überall, H. (1980a). *J. Acoust. Soc. Am.* **67**, 764.
Gaunaurd, G. C., Überall, H., and Moser, P. (1980b). *J. Appl. Phys.* (to be published).
George, J. (1980). *J. Acoust. Soc. Am.* **67**, 336.
Gjevik, B., Nilsen, A., and Höyen, J. (1976). *Geophys. Prospect.* **24**, 492.
Haug, A. J., Solomon, S. G., and Überall, H. (1978). *J. Sound Vib.* **57**, 51.
Hickling, R. (1962). *J. Acoust. Soc. Am.* **34**, 1582.
Johnson, G., and Truell, R. (1965). *J. Appl. Phys.* **36**, 3466.
Junger, M. C., and Feit, D. (1972). "Sound, Structures, and Their Interaction." MIT Press, Cambridge, Massachusetts.
Lane, A. M., and Thomas, R. G. (1958). *Rev. Mod. Phys.* **30**, 257.
Lewis, S. T., and Kraft, D. W. (1974). *J. Acoust. Soc. Am.* **56**, 1899.
Likhodaeva, E. A., and Shenderov, E. L. (1971). *Sov. Phys. Acoust.* **17**, 63.
McBride, R. J., and Kraft, D. W. (1972). *J. Appl. Phys.* **43**, 4853.
Madigosky, W., and Fiorito, R. (1979). *J. Acoust. Soc. Am.* **65**, 1105.
Marshall, S. W., and Olson, J. G. (1970). *J. Acoust. Soc. Am.* **47**, 949.
Medwin, H. (1975). *Proc. Conf. Oceanic Acoust. Modelling* No. 17, Part 2, Saclantcen, La Spezia, Italy.
Medwin, H. (1977). *J. Acoust. Soc. Am.* **62**, 1041.
Meyer, E., Brendel, K., and Tamm, K. (1958). *J. Acoust. Soc. Am.* **30**, 1116.

Minnaert, M. (1933). *Philos. Mag.* **26**, 235.
Moser, P. J., Murphy, J. D., Nagl, A., and Überall, H. (1981). *Radio Science* **16**.
Mow, C. C., and Workman, J. W. (1966). *J. Appl. Mech.* **33**, 793.
Murphy, J. D., Breitenbach, E. D., and Überall, H. (1978). *J. Acoust. Soc. Am.* **64**, 677.
Murphy, J. D., George, J., and Überall, H. (1979). *Wave Motion* **1**, 141.
Murphy, J. D., George, J., Nagl, A., and Überall, H. (1979a). *J. Acoust. Soc. Am.* **65**, 368.
Murphy, J. D., Moser, P. J., Nagl, A., and Überall, H. (1980). *IEEE Trans. Antennas Propag.* **AP-29**, No. 2.
Newton, R. G. (1966). "Scattering Theory of Waves and Particles." McGraw-Hill, New York.
Nigul, U. K. (1974). "Echo Signals from Elastic Objects," Vols. I and II. Academy of Sciences of the Estonian SSR, Tallinn. [*English translations*: Vol. II (1975), Joint Publications Research Service, Arlington, Virginia, Rep. No. JPRS-63937; Vol. I (1980), Navy Dept. (NISC), Washington, D. C., NISC Transl. No. 6271.]
Nishi, R. Y. (1975). *Acustica* **33**, 65.
Osborne, M. F. M., and Hart, S. D. (1945). *J. Acoust. Soc. Am.* **17**, 1.
Pao, Y. H., and Mow, C. C. (1976). *J. Acoust. Soc. Am.* **59**, 1046.
Peck, J. C., and Miklowitz, J. (1969). *Int. J. Solids Struct.* **5**, 437.
Pitts, L. E., Plona, T. J., and Mayer, W. G. (1976). *J. Acoust. Soc. Am.* **60**, 374.
Roman, P. (1965). "Advanced Quantum Theory." Addison-Wesley, Reading, Massachusetts.
Ruck, G. T., Barrick, D. E., Stuart, W. D., and Krichbaum, C. K. (1970). "Radar Cross Section Handbook." Plenum Press, New York.
Sage, K. A., George, J., and Überall, H. (1979). *J. Acoust. Soc. Am.* **65**, 1413.
Schoch, A. (1952). *Acustica* **2**, 1. [*English transl.*: Associated Technical Services, Inc., P.O. Box 271, East Orange, New Jersey.]
Solomon, S. G. (1980). Thesis, The Catholic Univ. of America, Washington, D. C.
Solomon, S. G., and Überall, H. (1981). *J. Acoust. Soc. Amer.* (to be published).
Strasberg, M. (1953). *J. Acoust. Soc. Am.* **25**, 536.
Strasberg, M. (1956). *J. Acoust. Soc. Am.* **28**, 20.
Su, J., Varadan, V., Varadan, V., and Flax, L. (1980). *J. Acoust. Soc. Am.* **68**, 686.
Subrahmanyam, J. V., Cowart, G., Keskin, M., Überall, H., Gaunaurd, G. C., and Tanglis, E. (1981). *IEEE Trans. Microwave Theory Tech.* (to be published).
Überall, H. (1973). Surface waves in acoustics, *In* "Physical Acoustics" (W. P. Mason and R. N. Thurston, eds.), Vol. 10, p. 1. Academic Press, New York.
Überall, H., Dragonette, L. R., and Flax, L. (1977). *J. Acoust. Soc. Am.* **61**, 711.
Überall, H., George, J., Farhan, A., Mezzorani, G., Nagl, A., and Sage, K. (1979). *J. Acoust. Soc. Am.* **66**, 1161.
Uginčius, P., and Überall, H. (1968). *J. Acoust. Soc. Am.* **43**, 1025.
Waterman, P. C. (1965). *Proc. IEEE* **53**, 805.
Waterman, P. C. (1969). *J. Acoust. Soc. Am.* **45**, 1417.
White, R. M. (1958). *J. Acoust. Soc. Am.* **30**, 771.
Wigner, E. P. (1946). *Phys. Rev.* **70**, 15, 606.
Wigner, E. P., and Eisenbud, L. (1947), *Phys. Rev.* **72**, 29.
Ying, C. F., and Truell, R. (1956). *J. Appl. Phys.* **27**, 1086.

—4—

Acoustic Emission – An Update

ARTHUR E. LORD, JR.

Department of Physics and Atmospheric Science,
Drexel University, Philadelphia, Pennsylvania

Introduction 295
I. Determination of Acoustic Emission Source Characteristics 297
II. Optical Detection of Acoustic Emissions 334
III. Signal Analysis 334
IV. Acoustic Emission during Various Transformations 335
V. Dislocation Effects 340
VI. Magnetomechanical Acoustic Emission 342
VII. Monitoring Fatigue Damage with Acoustic Emission 348
VIII. Practical Applications 350
IX. Discussion 353
References 355

Introduction

The first version of the review on acoustic emission (AE) appeared in Volume 11 of this work (Lord, 1975). This was published in 1975, but the review included references only to 1972. A great deal of work in AE has taken place since that time, and it was felt that an update was definitely needed.

At the end of that review it was summarized that AE had great applied use and potential, but that in the fundamental area not much had been forthcoming. In the past few years the practical area has grown and reached out to many new applications. Happily, in the fundamental area, there are recent strong signs that AE is coming of age as a viable tool in the materials and mechanics area.

A number of reviews have appeared since the last writing. Drouillard (1979) has published an extensive bibliography with references up to 1978. This is an excellent source book for a complete survey of the entire field. A

PHYSICAL ACOUSTICS, VOL. XV

Copyright © 1981 by Academic Press, Inc.
All rights of reproduction in any form reserved.
ISBN 0-12-477915-8

review published by the Materials Department of the University of California at Los Angeles under the editorship of Kanji Ono (1980) comprises the papers presented at a session on AE at the joint meeting of the Acoustical Societies of Japan and America in Hawaii in December 1978. A review has appeared by Ying (1973). Wadley and Scruby (1979) have also written a review. Current trends in the applied area have been surveyed by Spanner (1978, 1979). Recently, the use of AE in structural integrity monitoring was the subject of an ASTM monograph (Spanner and McElroy, 1974). A book dealing with AE in welding (edited by Nichols) appeared in 1975. There have been four AE symposia in Japan (the fourth in September 1978); unfortunately the proceedings seem to be difficult to obtain. The geotechnical area will not be covered here; it was reviewed by Lord and Koerner (1980).

I intend this to be a short review, devoted almost entirely to fundamental source characterization work. Most practical areas such as (the very important) pressure vessel monitoring, etc., will not be covered. Some excellent work in the area of composites and ceramics will not be covered; I apologize to workers in these areas. Another article to consult is that of Fowler *et al.* (1979), who discuss AE testing of fiber-reinforced plastic vessels. In the area of dislocation dynamics and deformation of metals, reliance will be heavy (almost total) on reviews by others. Controlled crack growth measurements will not be reviewed.

The areas to be covered are:

(1) Determination of AE source characteristics. This is mainly the work of the National Bureau of Standards (NBS), Cornell, Harwell, and Rockwell groups. The work involves careful system calibration (the entire system should be very flat) and the theory of Green's function propagators in simple geometries like half-spaces and infinite plates. Transducer calibration is most important here.

(2) Optical detection of AE and new signal analysis techniques. These will be treated briefly.

(3) Transformation work. The martensitic transformation work, which has been discussed previously, will be briefly looked at. The work in metallic glasses, silicate glasses, tin, silica firebrick, hydride forming materials, and electrical transformations will be discussed.

(4) Magnetomechanical effects. The acoustic emissions generated during domain wall movement will be treated. This area promises to be of value in nondestructive testing, e.g., residual stress determination.

(5) Dislocation effects. Heavy reliance will be made here on the review by Carpenter and Heiple (1980).

(6) Fatigue and AE. The work of Green and co-workers at Johns Hopkins, where AE and ultrasonic attenuation are measured simultaneously, will be discussed.

(7) New applications. The use of AE in welding and soldering on a production line will be highlighted. A very novel application is the testing of large aerial lift equipment. In-flight monitoring and measurement of coolant flow are also interesting applications of AE.

I. Determination of Acoustic Emission Source Characteristics

As indicated in the Introduction, the emphasis in this article will be on the recent work on identification of AE sources. Traditional methods of acoustic emission monitoring have been unsuccessful in actually giving solid (quantitative) detail as to the source generation mechanism. A seminal paper in this area is that of Breckenridge *et al.* (1975). [An earlier paper by Kolsky (1973) should also be consulted. This paper seems to be somewhat overlooked in the AE citations.] Breckenridge *et al.* used a capacitive (electrostatic) transducer and used only the initial arrivals in a large propagating media. In this way they eliminated the severe modification of the signal by the usual resonances in the transducer and the normal modes of the specimen. This effect is often called "ringing" of the transducer and specimen. [The first use of a capacitive transducer in AE measurements seems to have been Graham and Alers (1972). Pardee and Graham (1975) reported its use again later.] Figure 1 is a schematic

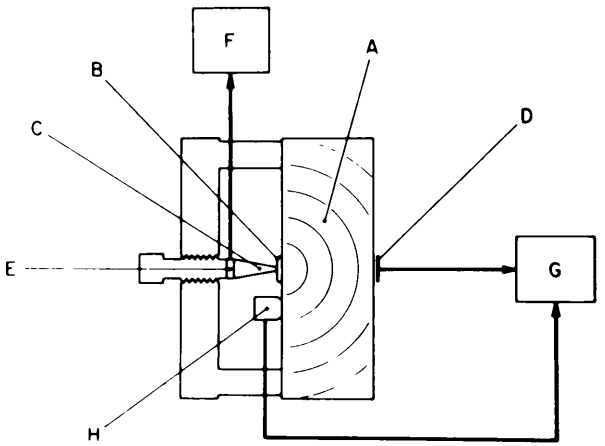


FIG. 1. Schematic of setup for recording AE signatures: A, transfer block; B, test sample; C, indenter; D, transducer; E. PZT disk; F, charge amplifier; G, high-speed storage scope; H, trigger signal pickup. [From Breckenridge *et al.* (1975).]

representation of the experimental setup of Breckenridge *et al.* As a standard source event they used the fracture of a thin-walled glass capillary. The thin wall allows the built-up force (before fracture) to be released very rapidly in the order of 0.1 μsec, and to approximate a step function release of force. The transducer responds to displacement of the sample (more on this later), and its amplitude response is quite flat, with frequency over the entire range of AE frequencies. For example, the first mechanical resonance of the air gap at a spacing of 4 μm is about 40 MHz.

The problem of the point step function of force applied normal to the plane surface of a semi-infinite isotropic solid was first studied by Lamb (1904). Pekeris (1955) has given explicit results, and they are shown in Fig. 2. The ordinate is the normalized vertical displacement of the surface U, and the abscissa is the time normalized to the shear wave arrival time. Z is the magnitude of the force, μ the shear modulus, r the distance from the force application point to the position where U is measured, and c is the shear wave velocity. P, S, and R depict the arrivals of longitudinal, shear, and Rayleigh waves, respectively. With varying Poisson ratio ν, the relative times of arrival change, but not the shape of the curve. Figure 3 shows the results obtained for the breaking glass capillary on a glass plate ($\nu = 0.226$) using the wide band detection scheme shown in Fig. 1 (except that the detector is on same surface as source). The exact details of the experiment are given in the figure caption. The jog near the end of the trace in Fig. 3 indicates the first reflection from the back of the block, which shows that the sample itself interferes with signal analysis after about 8 μsec for a

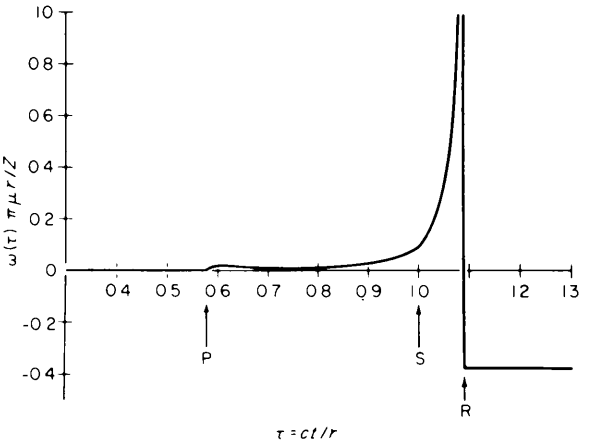


FIG. 2. Vertical displacement for seismic surface pulse calculated from expression by Pekeris for $\nu = \frac{1}{4}$. P, S, and R show arrival of longitudinal, shear, and Rayleigh waves, respectively. [From Pekeris (1975).]

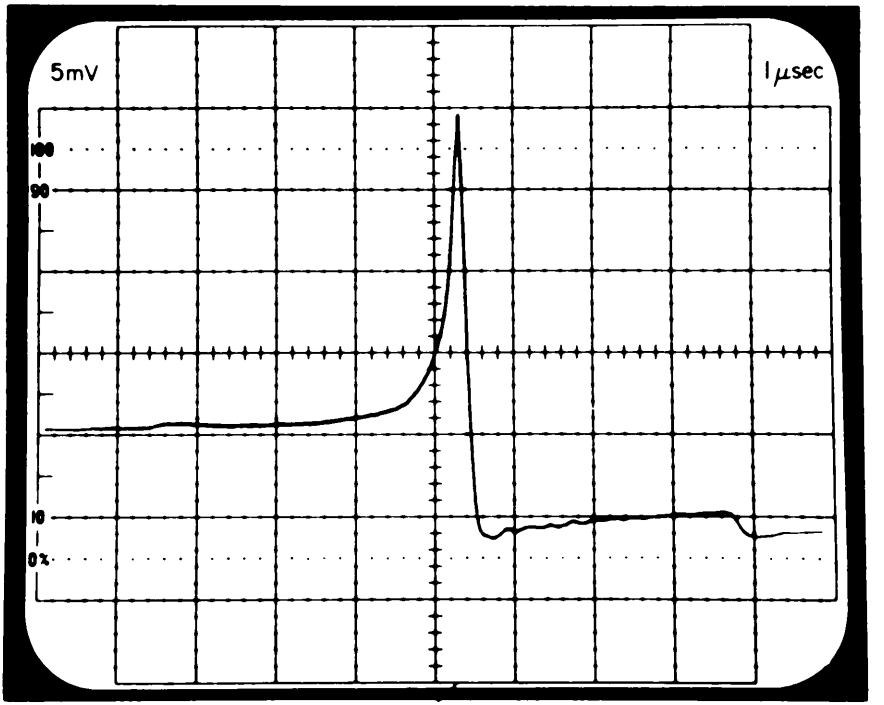


FIG. 3. Scope result for vertical displacement of breaking glass capillary (0.15 mm diam) on glass block. Glass block 3.3 cm thick, lateral dimensions greater than 9 cm. Brass electrode receiver: 6.33 mm diam; 2 μm air gap, 20 V polarization. [From Breckenridge (1975).]

3.3-cm-thick glass plate. Comparing Figs. 2 and 3, it is seen that the agreement with theory is excellent. The same agreement was also obtained for an aluminum alloy plate ($\nu = 0.343$). Theory predicts an infinity in displacement and zero fall time, but considering the impossibility of experimentally achieving a zero rise time pulse and perfectly flat transducer and electronics, the agreement is truly outstanding. [The tangential force with step function time dependence was first solved analytically by Chou (1960). No experimental work has been reported on the tangential force as yet.]

The problem of the buried pulse was first considered by Lamb (1904). Pekeris and Lifson (1957) have given explicit expressions for the problem. A step function of force is applied normal to the surface of an elastic half-space. The vertical displacement directly above the force (the so-called epicentral position) is shown in Fig. 4. In this situation h is the depth below the surface of the point of force application. The experimental result, obtained with the setup of Fig. 1, is shown in Fig. 5. The agreement

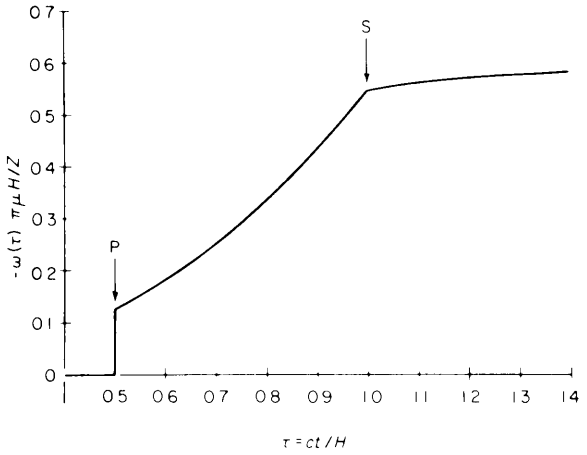


FIG. 4. Vertical displacement for seismic buried pulse calculated from Pekeris and Lifson for $\nu = \frac{1}{3}$. [From Pekeris and Lifson (1957).]

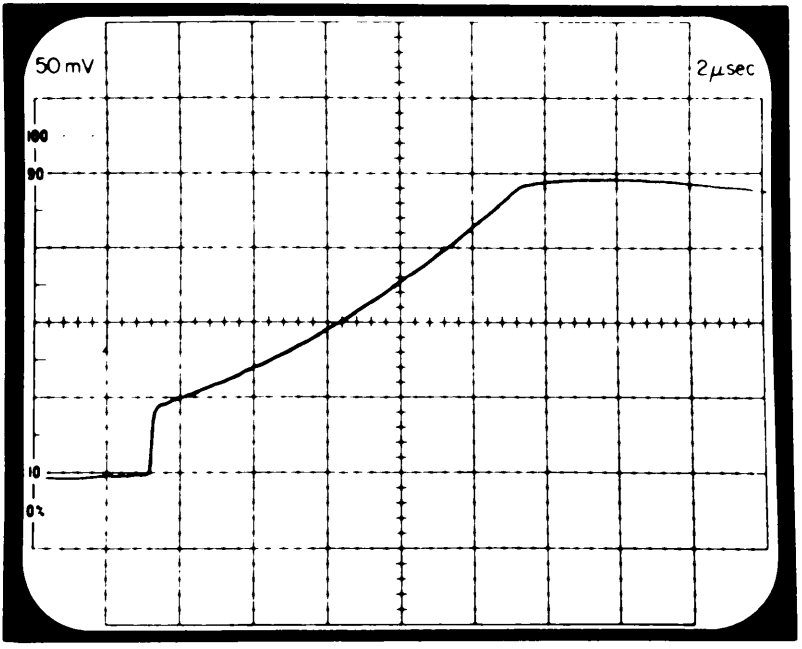


FIG. 5. Scope result for vertical displacement of breaking glass capillary on far surface of aluminum block, tungsten carbide electrode receiver: 9.6 mm diam; 4.1 μm air gap, 200 V polarization. [From Breckenridge (1975).]

between Figs. 4 and 5 is again extraordinary. The theoretical ratio of displacement at $\tau = 1.0$ to that at $\tau = 0.5$ is 4.38; the experimental value was 4.4. This paper by Breckenridge *et al.* (1975) gave the first real indication that it might just be possible to determine characteristics of actual AE sources. As such, it must be considered a classic in the AE literature. Coincidentally, it provides the basis for the best method (so far) of calibrating transducers. The details of its use in transducer calibration is given in Sachse and Ceranoglu (1970a) and Feng and Whittier (1979).

The next advance seems to be the work of Hsu *et al.* (1977). In this paper the convolution integral needed for source identification was formulated and numerically evaluated, using data obtained in a setup similar to Fig. 1. [Simmons and Clough (1976) also formulated the problem.] They modeled an AE event as a local force (or stress) drop $\mathbf{S}(x_0, t)$ at the point $\mathbf{x}_0$. In general $\mathbf{S}$ is a tensor. Of course AE events have a finite extent, but this is one of the simplest models that contains some essential physics, and may be a good approximation if the wavelengths of the AE's generated are very large with respect to the source dimension. If all aspects of the problem are linear, the displacement $\mathbf{U}(\mathbf{x}, t)$ can be written as the convolution integral

$$\mathbf{U}(\mathbf{x}, t) = \int_{-\infty}^{t} \mathbf{G}(\mathbf{x}, \mathbf{x}_0, t - \tau)\mathbf{S}(\mathbf{x}_0, \tau)\, d\tau, \tag{1}$$

where $\mathbf{G}(\mathbf{x}, \mathbf{x}_0, t - \tau)$ is the medium transfer function (tensor) usually called a Green's function, and represents the displacement in the structure at the point $\mathbf{x}$ and time t due to an impulsive force applied at the point $\mathbf{x}_0$ at time τ. If the force is a step function (Heaviside function) instead of an impulse, then the Green's function is denoted $\mathbf{G}^{\mathrm{H}}$·$\mathbf{G}^{\mathrm{H}}$ is related to $\mathbf{G}$ by the usual relations

$$\mathbf{G}^{\mathrm{H}}(\mathbf{x}, \mathbf{x}_0, t) = \int_0^t \mathbf{G}(\mathbf{x}, \mathbf{x}_0, \tau)\, d\tau \tag{2}$$

and

$$\frac{\partial G^{\mathrm{H}}}{\partial t}(\mathbf{x}, \mathbf{x}_0, \tau) = G(\mathbf{x}, \mathbf{x}_0, \tau). \tag{3}$$

Very little work has been done in the Green's function area. Stokes (1851) gave the solution for a point force in an infinite medium. The work of Lamb (1904), Pekeris (1955), and Pekeris and Lifson (1957) for the half-space have already been mentioned. Eringen (1957) treated the spherical cavity problem in the infinite space. Johnson (1974) has recently treated the half-space problem. Knopoff (1958) seems to be the first to treat the transients in a plate, and only relatively recently has there been renewed interest in the plate problem (Pardee, 1977). The point source problem has been treated by the geometric ray method (Mueller, 1968; Pao and Gajewski, 1977; Pao *et al.*, 1979). The Green's function work is quite

detailed and sophisticated mathematics and not casual fare for the experimentalist. The author of this review suggests that a good place to start might be Pao (1978). Here the Green's function for various "nuclei of strain" in an infinite space are given both in the time and frequency domain. The nuclei of strain are

(1) a concentrated force;
(2) a double force;
(3) a center of dilation (only P waves generated);
(4) a single couple;
(5) a center of rotation (only S waves generated);
(6) a double couple without moment.

Detailed numerical calculations for the transient waves generated by these "nuclei of strain" in an elastic plate are given by Ceranoglu and Pao (1980a, b, c) in work of significant importance. It is hoped that some of these sources, or a combination of them, can be used to model actual dynamic processes of material defects—as in the earthquake field. The method is based on the generalized ray theory and Cagniard's method.

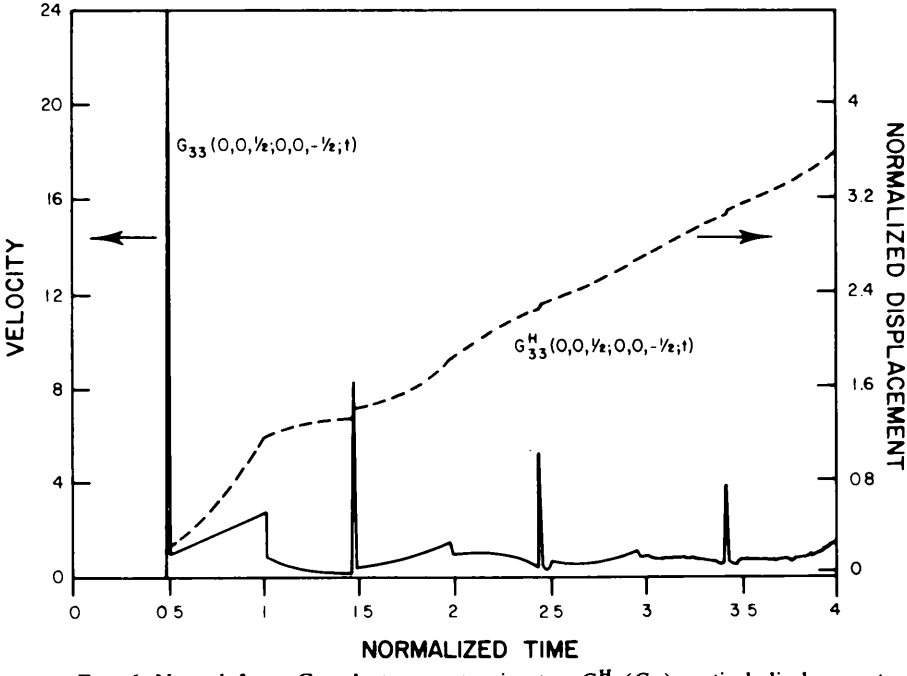


FIG. 6. Normal force Green's tensor at epicenter. G^{H}_{33} (G_{33}) vertical displacement (velocity) due to vertical force. [From Hsu *et al.* (1977).]

(These articles by Ceranoglu and Pao are an excellent source of references for the general transient elastic wave problem). Both buried and surface sources are considered and normal and transverse displacements are calculated. Surface epicentral responses are obtained for up to ten transit times of P waves across the plate. Time dependencies of the sources considered are step function, linear ramp, and parabolic ramp. Off-epicenter responses at locations up to six plate thicknesses are derived. The results are very detailed and depend on all the variables: type of source, orientation of source, time dependence of source, etc. This makes the deciphering of an actual source from detected displacement extremely difficult. However, if the AE source characterization problem is ever to be solved, such work as that of Ceranoglu and Pao is mandatory. Actual details of the calculation are beyond the intended scope of this review; the interested reader can consult the references for further details. Hsu *et al.* (1977) used solutions for the plate Green's function obtained using the Willis Fourier inversion technique (Willis, (1973). The results are shown in Figs. 6 and 7. G_{33}^{H} (G_{33}) is the Green's function for the vertical displacement (velocity) caused by a vertical force and G_{22}^{H} (G_{22}) for the horizontal

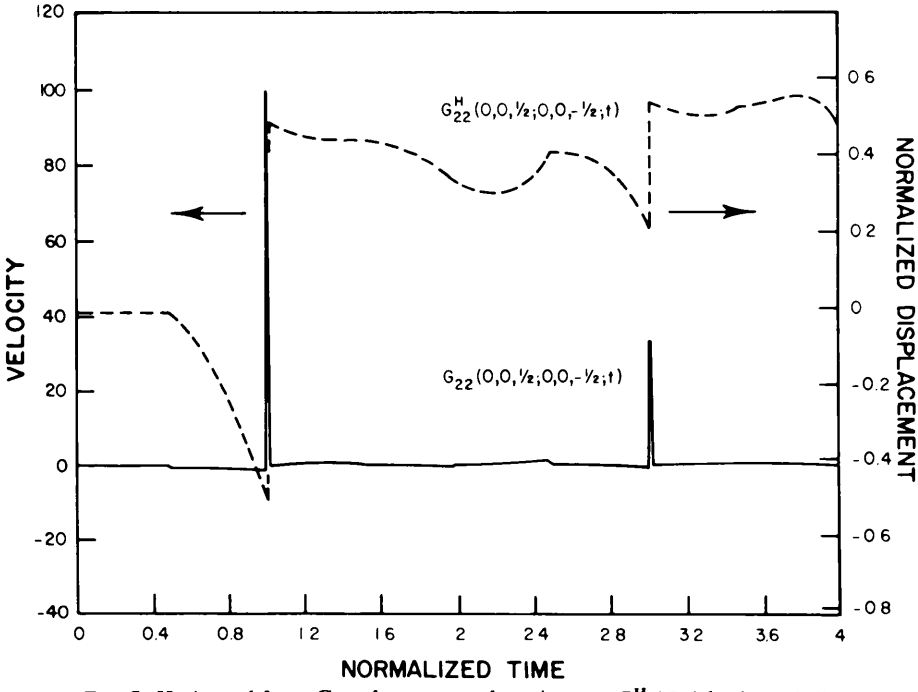


FIG. 7. Horizontal force Green's tensor at the epicenter. G_{22}^{H} (G_{22}) horizontal displacement (velocity) due to horizontal force. $G_{32}^{H} = G_{12}^{H} = 0$ ($G_{32} = G_{12} = 0$). [From Hsu *et al.* (1977).]

displacement (velocity) caused by a horizontal force. (Both are for the epicenter problem.) Note that G is obtained from G^H via Eq. (3). Knopoff (1958) had earlier derived G_{33} for the first few reflections, and the two treatments agree.

It might be appropriate here to discuss the ray method for the Green's function calculation. The description of Hsu *et al.* (1977) is followed. The solution is in the form of an infinite series expansion called a generalized ray expansion. The solution is exact for an interval of time dependent on the numbers of terms taken. The initial source function is resolved into its characteristic components—longitudinal (P), horizontal shear (SH), and vertical shear (SV)—and each of these components is followed through a series of reflections at the top and bottom faces of the plate until the receiver is reached, at which time they are recombined. One *ray* is the contribution of one such component followed through a particular series of reflections—taking into account mode conversions where longitudinal or vertical shear components split into each other at each reflection. All rays arriving at the same time at the receiver are grouped together and classed by arrival time. The solution can be generalized for force monopoles, dipoles, etc., located on the surface or within the plate. The theory is exact until such a time as arrivals occur from the lateral surfaces of the finite-sized plate.

Figure 8 shows a schematic of the experimental setup used by Hsu *et al.* (1977). The signal detected by the capacitive transducer and amplified is fed to a digital oscilloscope. The oscilloscope performs second-stage amplification, converts the analog signal to digital 8-bit binary signals at the sampling rate of 10 MHz, stores the digital information, and displays the recorded waveform. The oscilloscope was interfaced with a magnetic tape recorder so that the transient displacement signal could be recorded on magnetic tapes, written in a standard format readable by any large computer. The convolution integral [Eq. (1)] could then be numerically

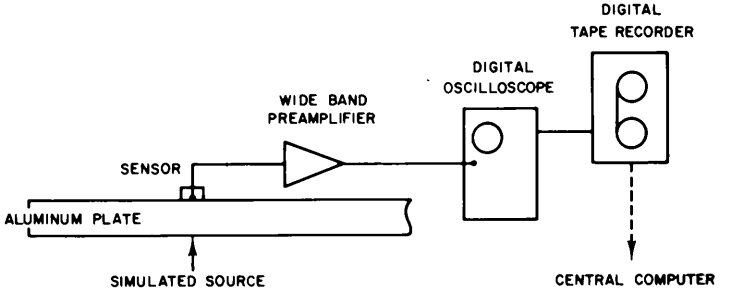


FIG. 8. Schematic diagram of the experimental system used for AE signal analysis. [From Hsu *et al.* (1977).]

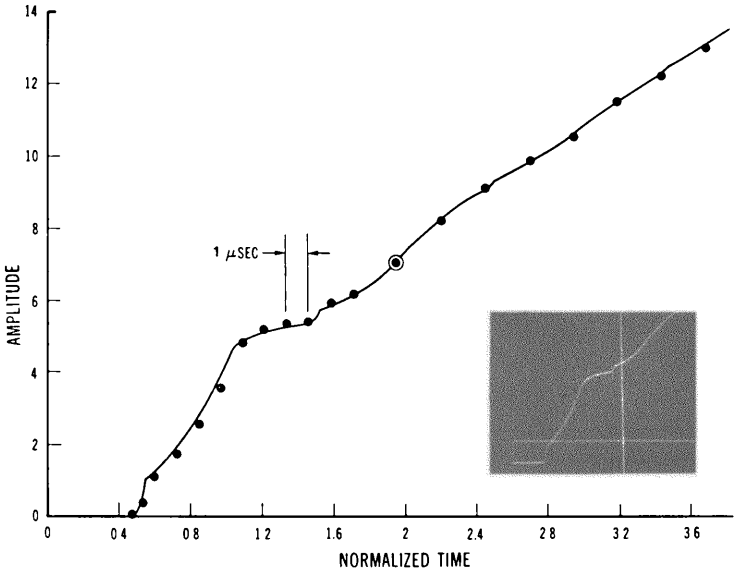


FIG. 9. Comparison between theoretical calculation of vertical displacement and experimentally measured voltage output of capacitive transducer at epicenter of large aluminum plate. ●, experimental points; ⊙ scaling reference point;—, theoretical calculation. [From Hsu *et al.* (1977).]

evaluated. Very good agreement was obtained between theory and experiment for the epicenter displacement produced by a step function normal force as shown in Fig. 9. The calculated curve was for a steplike force with a rise time of 0.54 μsec. The experimental result was produced by breaking a glass capillary of 0.5 mm diameter. Figure 9 applies for the first 40 μsec of transient wave. This quantitative agreement between theory and experiment gives strong evidence that:

(1) the simulated signal actually resembles a steplike force function;
(2) the capacitive transducer actually measures the true vertical displacement; and
(3) the ray theory method of calculating the Green's function is valid.

Having established the validity of the overall system, it should prove quite profitable to characterize AE sources. Before embarking on this aspect, however, it is instructive to show the response of the capacitive transducer and a commercially available piezoelectric transducer to the same step function of force. The result is shown in Fig. 10. The responses are completely different. In a crude, qualitative sense the piezoelectric transducer response resembles the velocity—for an exact comparison look at

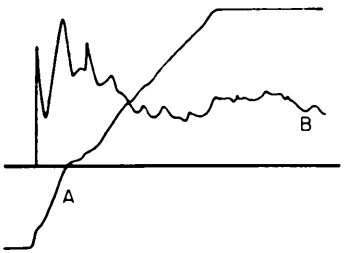


FIG. 10. Comparison of detected signals for step function force with two different sensors: (a) capacitive displacement transducer, and (b) piezo-electric transducer. [From Hsu *et al.* (1977).]

G_{33} of Fig. 6. This comparison gives vivid and ample proof as to why piezoelectric transducers cannot be used for AE source characterization.

As an example of "AE source" characterization, a wideband piezoelectric (ultrasonic) transducer excited by a short-duration electrical pulse was used in place of the glass capillary. The transducer was coupled to the plate with a thin layer of silicone vacuum grease. Figure 11a show the oscilloscope trace of the detected displacement at the epicenter as a function of time. (Because the signal here could be made repetitive, signal averaging was performed to reduce noise). Figure 11b shows the derived input force function obtained through direct deconvolution of the detected signal with the calculated Green's function G_{33}. Figure 11c is the reconstructed input displacement obtained by convolution of Figure 11b with G_{33}. The agreement is quite good considering the limited number of points used in the mathematical operations. This work gives strong preliminary evidence that it is possible to identify nonzero rise time finite AE sources.

More work by the NBS group is discussed in Hsu and Hardy (1978) and Hsu and Eitzen (1979). More signal processing is described here (Fig. 12). Detected signals are amplified and digitized at a 10 MHz sampling rate, 8-bit work length, and maximum record length of 4096 points. The digitized signals are transferred to a minicomputer. The minicomputer has a fast double-precision floating arithmetic processing unit and two 50 Mbyte disks; it is used to control experiments, to acquire and process data, and to store both raw and processed data.

Hsu and Hardy (1978) and Hsu *et al.* (1977) describe an alternative method for producing step function AE's. Instead of the capillary, they break fine pencil leads (in a mechanical pencil) on the surface of the plate. The method is easier and more reproducible than the capillary method. Figure 13 gives results of the displacement picked up by the capacitive transducer as a function of force applied before fracture of the lead. Figures 14a and b compare the force–time functions for the breaking

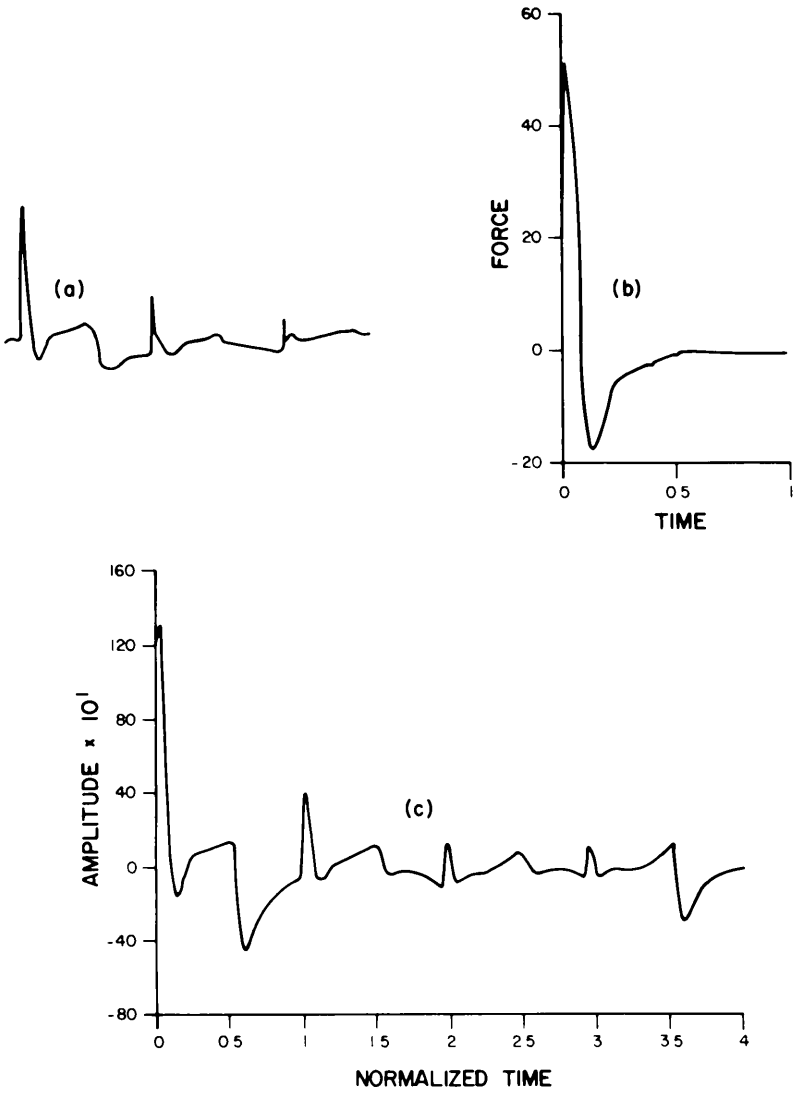


FIG. 11. (a) Scope trace of detected epicenter displacement due to unknown stress pulse produced by wide band ultrasonic transducer; (b) numerically computed input force function through deconvolution of (a) with theoretical Green's function; (c) is reconstructed wave form of (b) convoluted with Green's function. [From Hsu *et al.* (1977).]

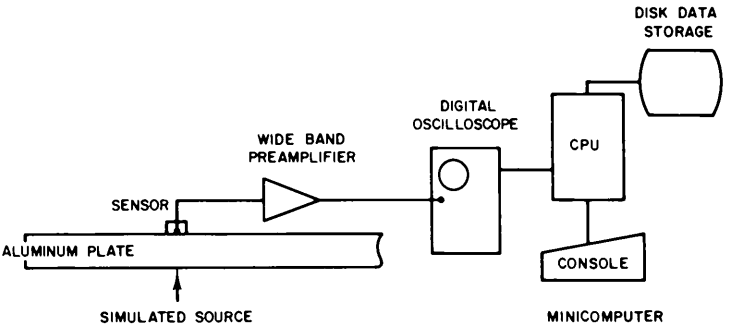


FIG. 12. Schematic diagram of experimental configuration for characterizing the components of an AE system. [From Hsu and Hardy (1978).]

capillary and breaking lead. The breaking lead is also seen to be a sharp rise time "step function."

Hsu and Hardy (1978) and Hsu and Eitzen (1979) have presented the mathematics of the convolution and deconvolution needed for source identification. Equation (1) is written in the more simplified form

$$U(t) = \int_0^t G(t-\tau)S(\tau)\,d\tau, \tag{1a}$$

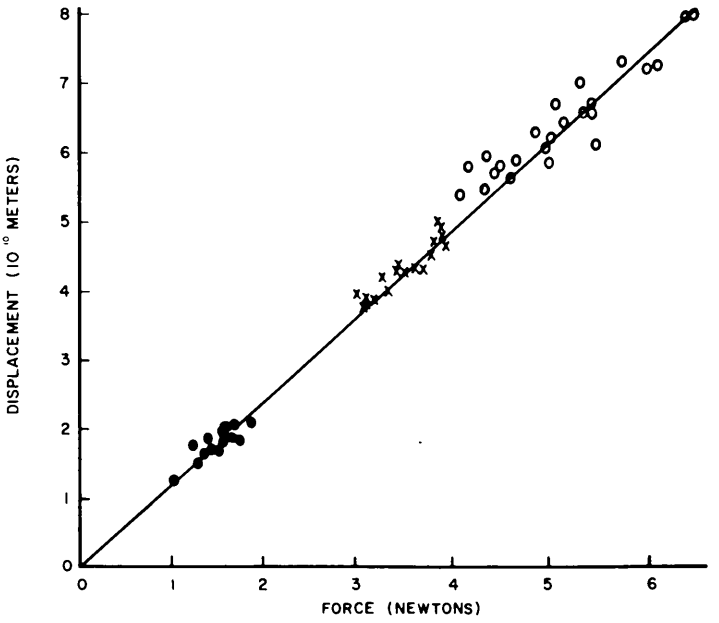


FIG. 13. Displacement versus breaking force for pencil lead. ●, 0.2-mm lead (~1 mm long); ×, 0.3-mm lead (~1.5 mm long); 0, 0.5 mm lead (~2 mm long). [From Hsu and Hardy (1978).]

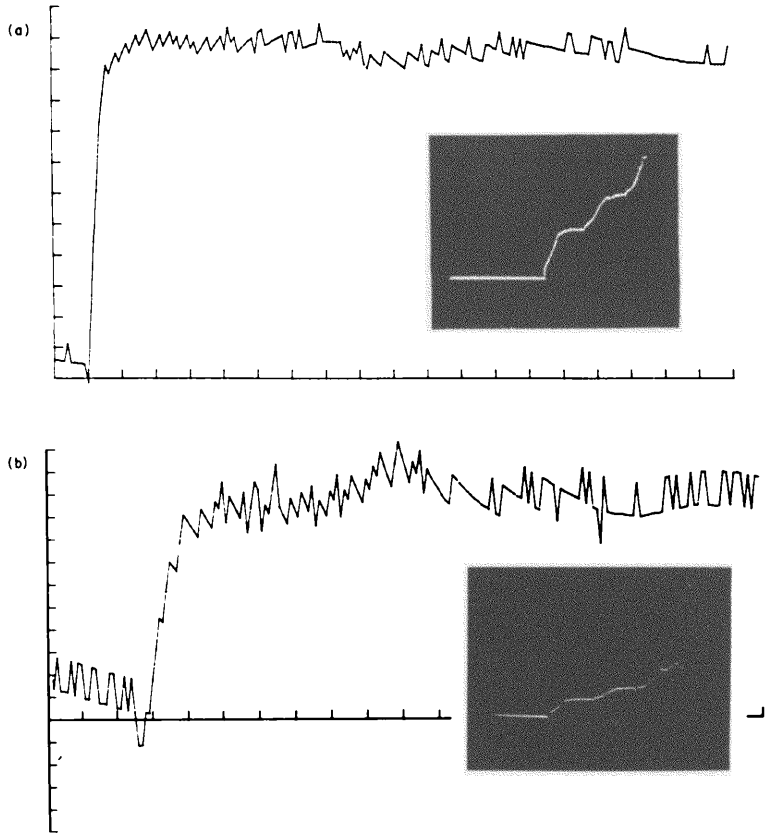


FIG. 14. (a) Source force–time function of breaking glass capillary obtained by time-domain deconvolution of recorded epicenter displacement. Insert trace is recorded epicenter displacement. (b) Breaking 0.5-mm lead. Source force–time function of breaking 0.5-mm pencil lead. Insert trace is the recorded epicenter displacement. Full scale is 20 μsec. [From Hsu and Hardy (1978).]

assuming the source does not exist for $t < 0$. Solving explicitly for the force,

$$S(t) = \int_{-\infty}^{\infty} U(\tau) G^{-1}(t - \tau)\, d\tau. \tag{4}$$

The inverse Green's function is defined as

$$\int_{-\infty}^{\infty} G(\tau) G^{-1}(\tau - t)\, d\tau = \delta(0). \tag{5}$$

The following relations are also given:

$$U(t) = \int_{0}^{t} \dot{S}(\tau) G^{H}(t - \tau)\, d\tau, \tag{6}$$

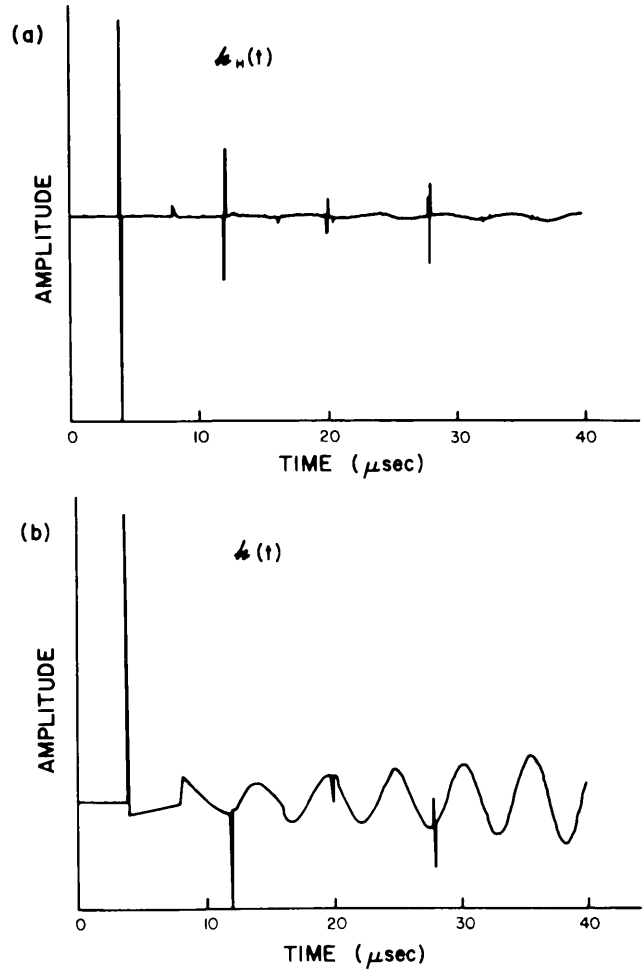


FIG. 15. (a) $[G^{H}(t)]^{-1}$, inverse function of $G^{H}(t)$; (b) $[G(t)]^{-1}$, inverse function of $G(t)$. [From Hsu and Hardy (1978).]

where $\dot{S}(t)$ is the time derivative of the source function and G^{H} (as given before) is the step function Green's function. The deconvoluted form of Eq. (6) is

$$\dot{S}(t) = \int_{-\infty}^{\infty} U(\tau)(G^{H})^{-1}(t-\tau)\, d\tau, \tag{7}$$

where $(G^{H})^{-1}$ is the inverse of G^{H}. Figures 15a and b show the inverse functions $(G^{H})^{-1}$ and G^{-1} for the vertical force–epicenter displacement problem in an aluminum plate. Both functions were obtained numerically

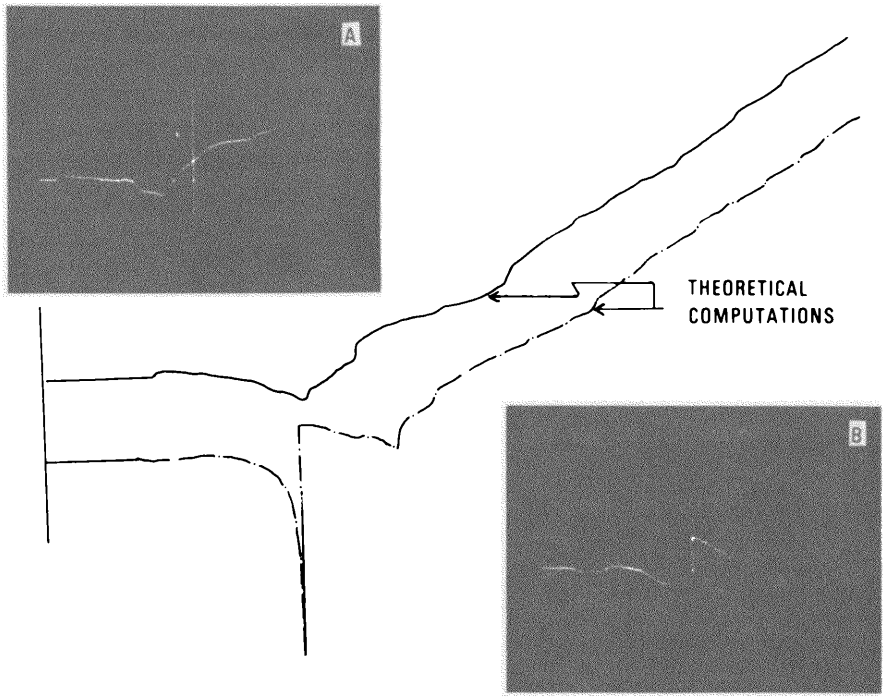


FIG. 16. Experimentally determined off-epicenter vertical displacements due to vertical step impulse. Both two plate thicknesses off center. (a) Sensor and source on opposite sides. (b) Sensor and source on same side. [From Hsu and Hardy (1978).]

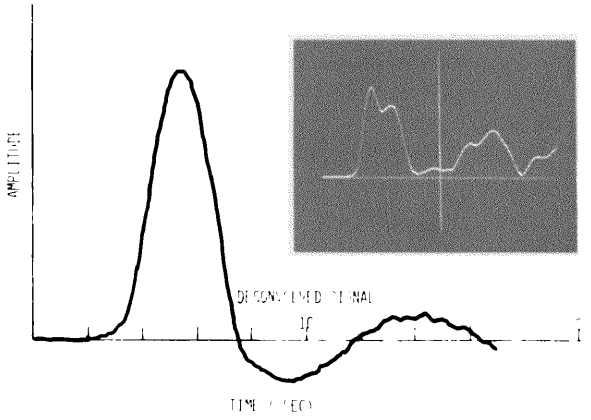


FIG. 17. Source–time function of dropping 1.5-mm steel ball from 5-cm height. Insert trace is the recorded epicenter displacement. [From Hsu and Hardy (1978).]

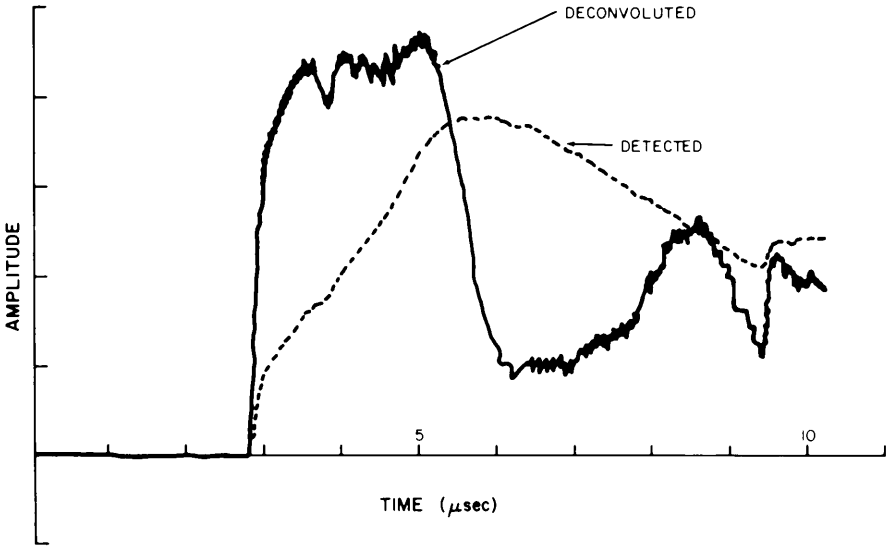


FIG. 18. Brittle fracture produced on a glass plate by impact of a diamond indentor. Dotted line is recorded epicenter displacement. Solid line is the deconvoluted source force–time function. [From Hsu and Hardy (1978).]

from the appropriate Green's function. Physically the curve of G^{-1} (Fig. 15b) is the force needed to produce a delta function (in time) displacement at the epicenter. Figure 16 shows the results for the displacement picked up off-epicenter. The captions give the details of the experiment. Figure 17 gives the force–time function obtained in dropping a 1.5-mm steel ball from a height of 5 cm onto the propagation plate. The inset in Fig. 17 gives the detected displacement–time relation. The results agree well with elasticity theory (Goldsmith, 1960; Landau and Lifshitz, 1970). Figure 18 shows the detected displacement and calculated force (assuming a given Green's function) for a brittle fracture event from impact of a diamond indenter on a glass plate. It is very difficult (if not impossible) to know the exact Green's function to use in fracture problems.

The algorithms used in the convolution and deconvolution computations are given explicitly in Hsu and Hardy (1978) and Hsu and Eitzen (1979).

Experimental work in this regard at Cornell University is discussed in Sachse and Ceranoglu (1979a). They use a well-characterized AE system to obtain the force–time functions of various surface sources. These include the fracture of glass capillaries, the impact of spheres, electrical arcs, and piezoelectric transducers. The transfer function of any transducer can be

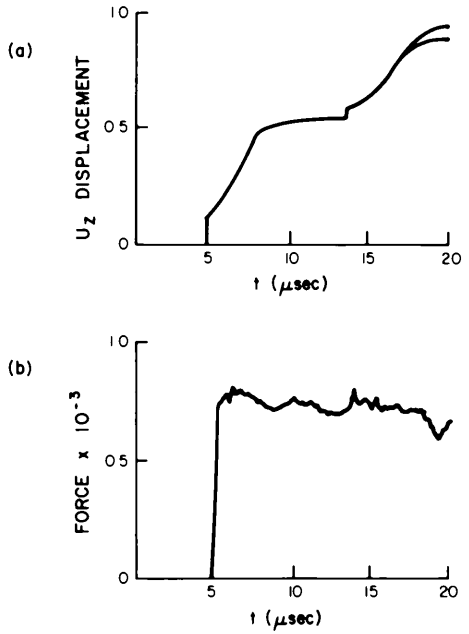


FIG. 19. (a) Epicentral displacement, theory and experiment. (b) Result of deconvolution. [From Sachse and Ceranoglu (1979a).]

obtained and then converted into absolute displacement or velocity. The dynamic Green's function was found experimentally for a thick plate for various source to receiver locations. The measurements were made with a broadband (100 Hz to 10 MHz) low noise AE system (capacitive transducer). The signals were digitized to 8-bit accuracy with a transient recorder operating at sampling rates up to 100 MHz. The recorder was interfaced directly with a minicomputer, which is used for subsequent deconvolution (and other signal processing operations on the signals). Figure 19 shows the results for the theory and experiment for the breaking glass capillary (step function force release). The transfer function (i.e., response ratio of voltage to displacement) of a broadband piezoelectric transducer is shown in Fig. 20c; Fig. 20a and b give the source displacement and piezoelectric transducer signal, respectively. Figure 21 shows the result of comparing the response from the same source (ball impact) of (1) the piezoelectric and (2) the capacitive transducer "corrected" to the piezoelectric response using the previously determined transfer function (Fig. 20c).

Sachse and Ceranoglu (1979b) performed a very interesting experiment in order to determine the dynamic Green's function for a plate with

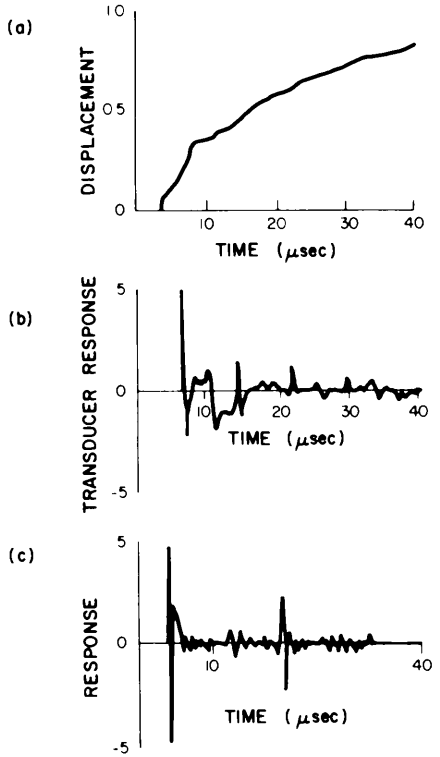


FIG. 20. Determination of piezoelectric transducer transfer function (a) epicenter displacement, step unloading; (b) piezoelectric transducer signal, step unloading source; (c) result of deconvolution. [From Sachse and Ceranoglu (1979a).]

various positions of the step function source and the capacitive transducer. Two experiments are required. In the first, the voltage $V_1(t)$ is measured with a transducer coupled to a structure whose impulse response $G_1(t)$ is known when the source $S(t)$ is operative:

$$[S(t)*T(t)]^{-1} = [V_1(t)]^{-1}*G_1(t), \tag{8}$$

where $T(t)$ is the transfer function. The experiment is repeated with an identical source and sensor but on a structure whose impulse response is to be determined:

$$G_2(t) = [V_2(t)]^{-1}*G_1(t)*V_1(t). \tag{9}$$

Figure 22 shows the calculated Green's function for the response when the receiver is located two and four plate thicknesses from the epicentral location (a and b); c and d show the Green's function calculated from Eq.

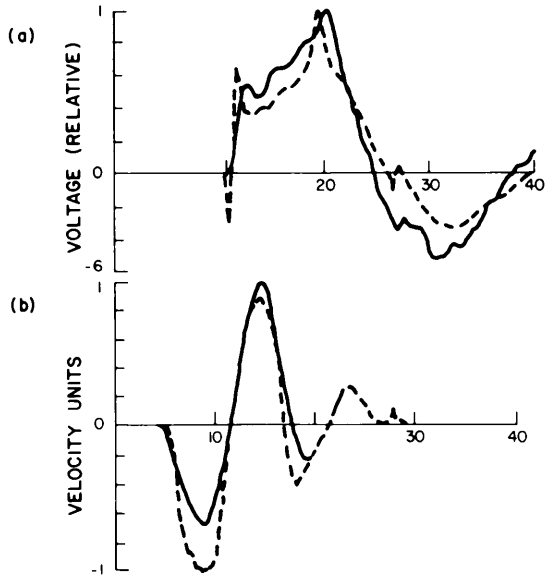


FIG. 21. (a) Piezoelectric transducer voltage signal measured by a capacitive transducer and actual voltage signal; (b) velocity signal measured by piezoelectric transducer and actual voltage signal. ---, measured signal; —, signal obtained by deconvolution. [From Sachse and Ceranoglu (1979a).]

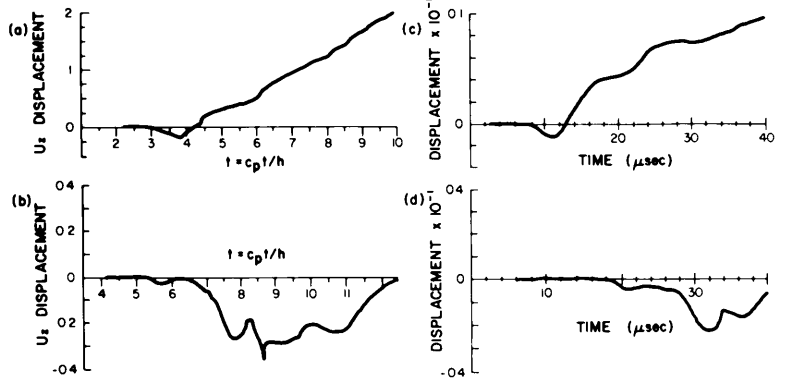


FIG. 22. (a) $r = 2h$ and (b) $r = 4h$ calculated Green's function. (c) $r = 2h$ and (d) $r = 4h$ experimentally measured Green's function using a capacitive transducer. [From Sachse and Ceranoglu (1979b).]

(9) and the experimental data. The general agreement is good, but some high-frequency detail is missing in the measured value.

The group at Harwell in England has contributed significantly to the understanding of AE sources. Their approach was first described in Scruby *et al.* (1978). Although they used the source models of Pekeris (1955) and Pekeris and Lifson (1957), mention is made of the work of Burridge and Willis (1969) and Willis (1973) who published solutions for the motion of the surface due to expanding cracks. More will be said about the realism of the source models as we progress. Figure 23a shows again the model of Pekeris and Lifson (1957); Fig. 23a (ii) is for a finite rise time source. Figure 23b gives the displacement versus time for the force dipole; this is obtained from the force monopole by differentiation. The convolution integral similar to Eq. (1) is set up:

$$\mathbf{U}(t, \mathbf{r}) = \mathbf{G}(t, \mathbf{r}) * \mathbf{S}(t), \tag{1b}$$

where * means the convolution process defined in Eq. (1). The Fourier transform of Eq. (1b) is

$$\mathbf{u}(f, \mathbf{r}) = g(f, r)s(f), \tag{10}$$

where u, g, and s are the Fourier transforms of U, G, and S, respectively. The source function at a particular frequency f can therefore be given by

$$s(f) = u(f,r)/g(f,r). \tag{10a}$$

Thus the characteristics of the source can be obtained in the frequency domain as well as the time domain. The time domain approach seems

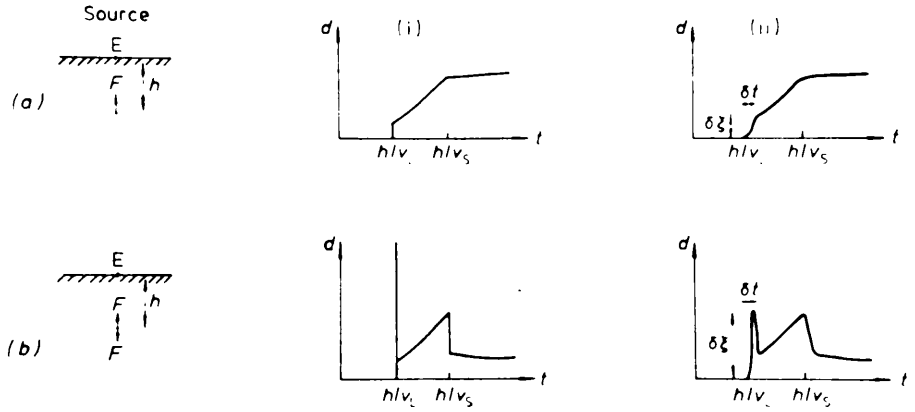


FIG. 23. The vertical surface displacement of epicenter due to point source at a depth h. The source is an unbalanced force in (a) and a force dipole in (b). In each case the surface displacement are given for sources with (i) zero and (ii) nonzero (δt) rise time. The longitudinal leading edge amplitude is $\delta\xi$. [From Scruby *et al.* (1978).]

favored due to problems with normal modes, phase effects, etc., on the transfer functions in the frequency domain approach. Frequency domain work is described in Fleischmann *et al.* (1975), McBride and Hutchison (1976), and Houghton *et al.* (1978).

The theory of Pekeris and Lifson (1957) is used to obtain a numerical result for the magnitude of the first arrival in displacement ΔU (in steel)

$$\Delta U = 0.047(\Delta F/\mu h)H(t - h/v_L), \tag{11}$$

where ΔF is the magnitude of the force step function and H is the Heaviside step function. So a measurement of the initial displacement allows a determination of the force (stress) release causing the AE. Scruby

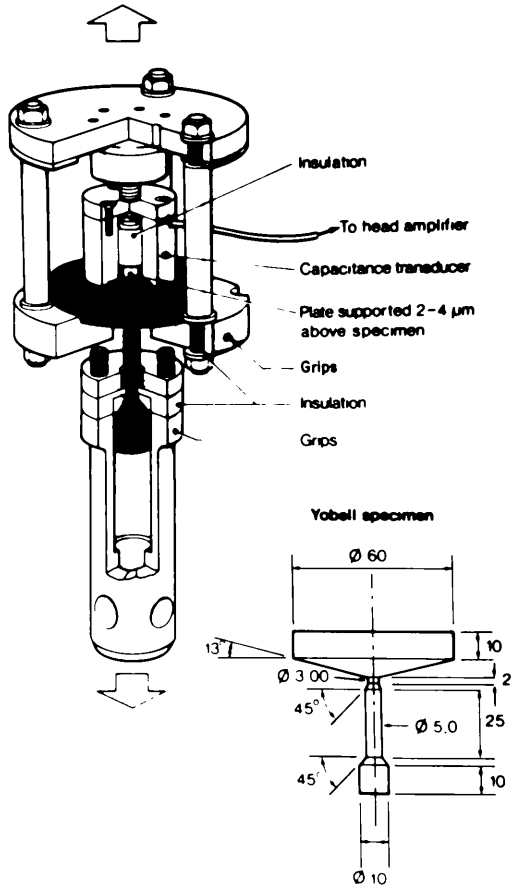


FIG. 24. Cutaway drawing of capacitance transducer and grips used for testing "Yobell" specimens under tensile load. Dimensions in millimeters. [From Scruby *et al.* (1978).]

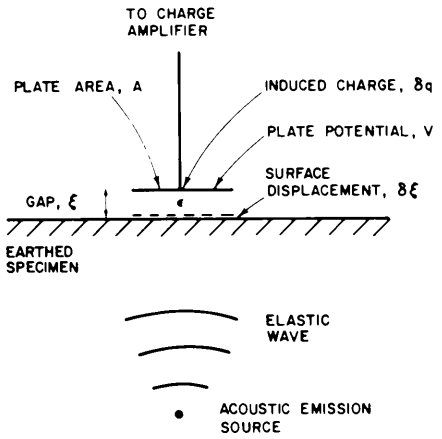


FIG. 25. Schematic diagram of a capacitance transducer. [From Scruby and Wadley (1978).]

et al. (1978) have developed a sample geometry that is a compromise between a large body for propagation purposes and a long slender body for tensile testing purposes and they call it a "Yobell" geometry. (Refer to Fig. 24.) The sample can be tested in a conventional tensile testing machine and the upper large portion should allow 6 μsec of unimpeded (by boundaries) wave propagation in steel. The AE's are forced to occur in the small (2-mm-long) gauge section immediately below the large portion. Scruby and Wadley (1978) have discussed their capacitance transducer in detail. The capacitance transducer has a much larger bandwidth than the conventional piezoelectric AE transducers, but larger bandwidth always implies lower sensitivity. Thus capacitance transducers can detect only the more energetic (higher amplitude) AE sources. Consider the schematic of the transducer shown in Fig. 25. The device used here consists of two parallel plates of area A separated by a small air gap ξ. One plate is the polished surface of the sample (grounded). The other plate in the transducer is maintained at a constant voltage V. Therefore oscillation of the surface due to arrival of an elastic wave causes the capacitance C of the transducer to change. Keeping V constant, a charge δq is induced on the plates:

$$\delta q = V\delta C. \tag{12}$$

Now for a parallel plate capacitor

$$C = \epsilon A/\xi, \tag{13}$$

where ϵ is the permittivity of the gap,

$$\delta q = -\epsilon A V \delta\xi/\xi^2. \tag{14}$$

Here $\delta\xi = U$, the elastic displacement. The sensitivity of the transducer to vertical surface displacement is

$$\frac{dq}{d\xi} = -\frac{\epsilon VA}{\xi^2}. \tag{15}$$

Equation (15) shows the sensitivity is increased by increasing plate area, decreasing plate separation, and increasing V. The plate area is, however, limited by bandwidth considerations. Consider a source buried at a depth h below the transducer plate (of radius a) on the axis of the transducer. If $a \ll h$, the path difference between waves reaching the center and edge of the transducer is

$$\delta h = a^2/2h. \tag{16}$$

With a wave velocity v, the difference in transit times is

$$\delta t = a^2/2vh. \tag{17}$$

If $h = 20$ mm, $a = 3$ mm, and $v = 6000$ msec^{-1}, δt is 37 nsec. This essentially determines the rise time of the transducer, and the reciprocal, roughly the upper frequency limit, i.e., about 30 MHz. The position of the source with respect to the axis of the transducer as well as the orientation of source places limits on the bandwidth. Typical operating conditions are $\xi = 4$ μm, $a = 3$ mm, and $V = 50$ V. This gives a sensitivity of 0.8 pC/nm. For a 4-μm air gap, the lowest resonant frequency of the transducer spacing itself is about 40 MHz. At these very small spacings the surfaces must be polished flat and parallel and kept quite clean. The transducer is calibrated by measuring its capacitance for independently measured plate separations; the details are given in Scruby and Wadley (1978). It is possible to detect displacements as small as 10^{-12} m (0.01 A).

The transducer is coupled to high impedance, low noise, wide band charge amplifier (the impedance must be high for the transducer impedance is very high–capacitance typically 120 pF; resistance > 10 GΩ.) A schematic of the setup is shown in Fig. 26. The output from the charge amplifier is passed through a bandpass filter (35 kHz to 45 MHz) before further amplification. The filter limits being set to remove low-frequency machine noise and to prevent high-frequency aliasing. Radio-frequency interference is reduced by putting the specimen, transducer, and charge amplifier in a Faraday cage. The signal then feeds into a Biomation 8100 transient recorder where it is digitized prior to memory storage. The recorder was linked to a PDP 8/E computer and data stored on a magnetic disk. The accuracy with which transient signals could be recorded is limited by the precision (8-bit) and sampling frequency (100 MHz) of the analog-to-digital converter. The overall sensitivity of the

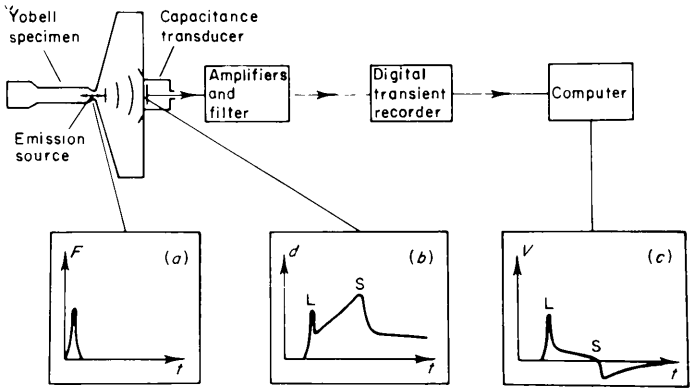


FIG. 26. Schematic diagram of the transient recording system, showing the "Yobell" specimen, which is tested in uniaxial tension along its axis. (a) Source function, (b) vertical surface displacement, (c) stored digitized voltage signal. [From Scruby and Wadley (1978).]

system (for a 4-μm transducer air gap) is 32 V nm^{-1}, while the dynamic range was limited by the 8-bit precision of the recorder to surface displacements of 1–18 pm. The rise time of the recording system (but excluding the transducer) was found to be 40 nsec. This part of the system had a flat frequency response over an equivalent bandwidth.

The system has been used to record AE's from a range of fracture processes in steel. The results given in Scruby *et al.* (1978) are for a temper-embrittled EN30A (low alloy) steel. The fracture mode was a ductile cup and cone, and 36 AE's were detected by the broadband system in the region of the stress–strain curve immediately before failure. Only 24 of these were within the dynamic range of the system, and 3 of these are shown in Fig. 27. The delta function of the theoretical longitudinal arrival is replaced experimentally by a peak of finite height and width. The results are analyzed in terms of the pulse amplitude [from Eq. (11)] and rise time (time to rise from 10% to 90% of its maximum). Provided that the rise time is longer than the system response time (which was about 40 nsec), it is a measure of the source life time, while the amplitude is a measure of the force drop during AE generation. If the source is a propagating crack, and if the local stress level is known, a static point source model can be used to estimate the area of the new crack surface. The speed of the crack can be obtained from the slope of the leading edge of the arrival. Histograms of the rise times and amplitudes are given in Fig. 28. Transient rise times range from 40 to 120 nsec and amplitudes from 1 to 18 pm [0.03–0.6 N from Eq. (11)]. The typical source has a life time of about 60 nsec and a magnitude of 0.2 N for a vertical force relaxation. The macroscopic stress level during the AE generation was 10^9 Pa and the local stress in the

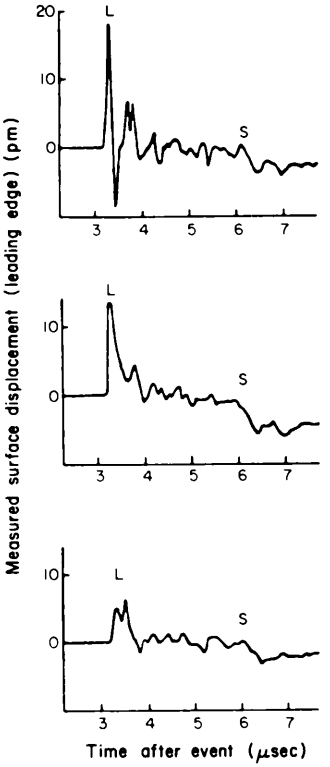


FIG. 27. Three surface displacement transients measured during the ductile fracture of a low alloy steel. [From Scruby *et al.* (1978).]

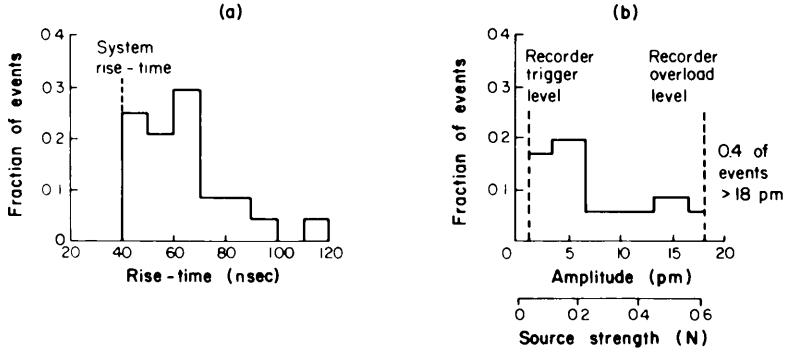


FIG. 28. Normalized rise time and amplitude distributions for the displacement transients recorded during the ductile fracture of EN30A steel. There were 24 emissions within the dynamic range of the system. [From Scruby *et al.* (1978).]

vicinity of the source was undoubtedly higher. This gives an estimate of the crack area A of

$$A = 0.2\ \mathrm{N}/10^9\ (\mathrm{N/m^2}) = 2 \times 10^{-10}\ \mathrm{m^2} = 200\ \mu\mathrm{m^2},$$

which is at least roughly consistent with SEM fractography observations. A typical measured surface velocity of 100 pm $\mu\mathrm{sec}^{-1}$ corresponds to a crack growth rate of

$$200\ \mu\mathrm{m^2}/60\ \mathrm{nsec} = 3300\ \mu\mathrm{m^2}\ \mu\mathrm{sec}^{-1},$$

using the model of a typical source event which is the propagation of a crack over an area of 200 $\mu\mathrm{m}^2$ in 60 nsec. A square crack of area 200 $\mu\mathrm{m}^2$ has a linear dimension of about 14 μm. If this crack opening takes place in 60 nsec, the average velocity of the crack is

$$14 \times 10^{-6}\ \mathrm{m}/60 \times 10^{-9}\ \mathrm{sec} = 2.3 \times 10^2\ \mathrm{msec}^{-1}.$$

This is the order of 0.1 the shear velocity in steel.

Wadley and Scruby (1979a) have used their wide band detection system to investigate the characteristics of the fracture events in a low alloy steel (4 Ni, 1 Cr), heat treated to different mechanical conditions. The conditions were quenched, tempered, and temper-embrittled. Some samples were hydrogen treated (embrittled) before testing. The hydrogen was introduced by cathodic charging. The Yobell samples were tested to fracture and the AE transients analyzed as described earlier (Scruby *et al.*, 1978). The essential results are shown in Figs. 29 and 30. The data are not complete, but some interesting features begin to emerge. The rise times of hydrogen-free material tended to be shorter than those of hydrogen-charged samples (compare Figs. 29b and c). Temper-embrittling leads to shorter rise times for hydrogen-free material (compare Figs. 29a and b). The distributions of rise times for the hydrogen-embrittled material in the quenched condition (Fig. 29d) appeared to be the same as that of unembrittled material (Fig. 29a). The material tested in the temper-embrittled condition which underwent intergranular fracture in the absence of hydrogen (Fig. 29c) gave a distribution that was different from material in the same condition which underwent either a ductile fracture (Fig. 29b) or a hydrogen-assisted intergranular fracture (Fig. 29e). In general, events that generated the emissions during ductile fracture are faster than those generated during hydrogen-assisted intergranular brittle fracture.

Amplitude distributions are given in Fig. 30. Initial surface displacements were from 1 to 20 pm, which corresponds to force magnitudes of 0.03–0.06 N. This gives crack increments of 40–750 $\mu\mathrm{m}^2$ assuming a local stress of 800 MN m^{-2}. SEM observations of fractures indicated the chief difference between tempered materials which generated little emission and temper-embrittled material which was an active generator was the presence

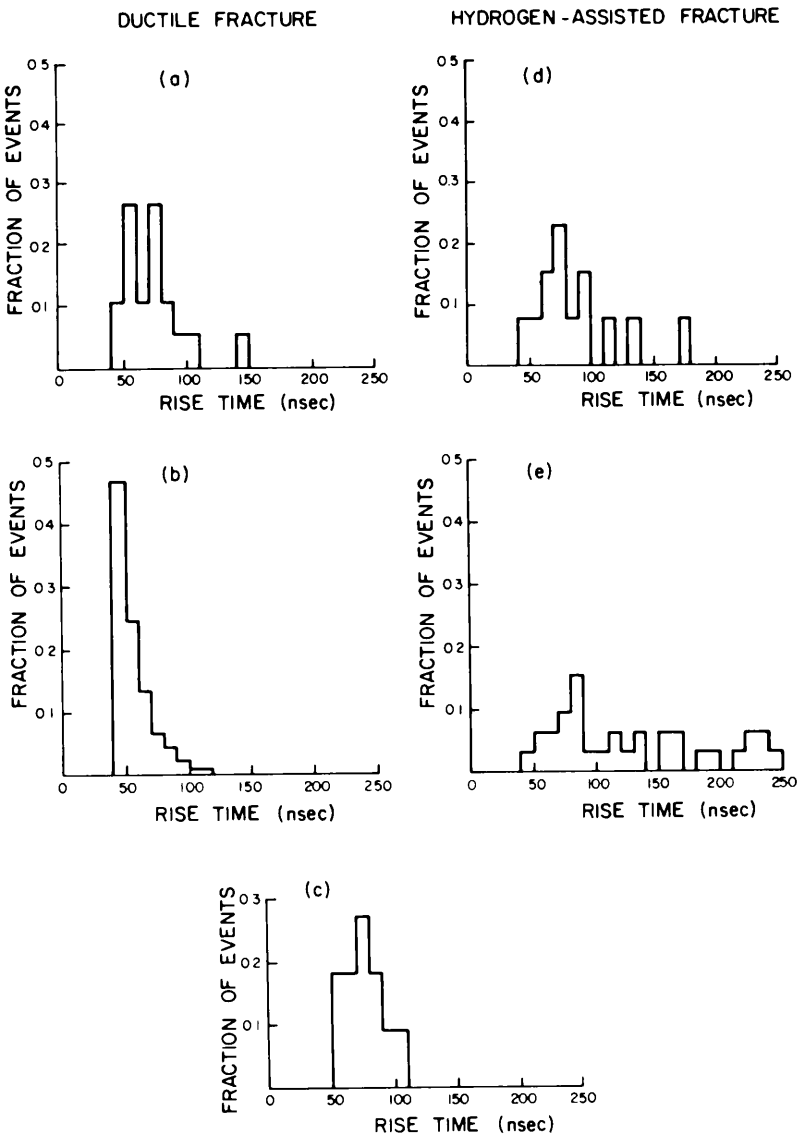


FIG. 29. Risetime distributions of AE transients recorded during the fracture of EN30A steel in different conditions. (a) Quenched; (b) temper embrittled (duct. 6), (c) temper embrittled (intergranular); (d) quenched; and (e) temper embrittled. [From Wadley and Scruby (1979a).]

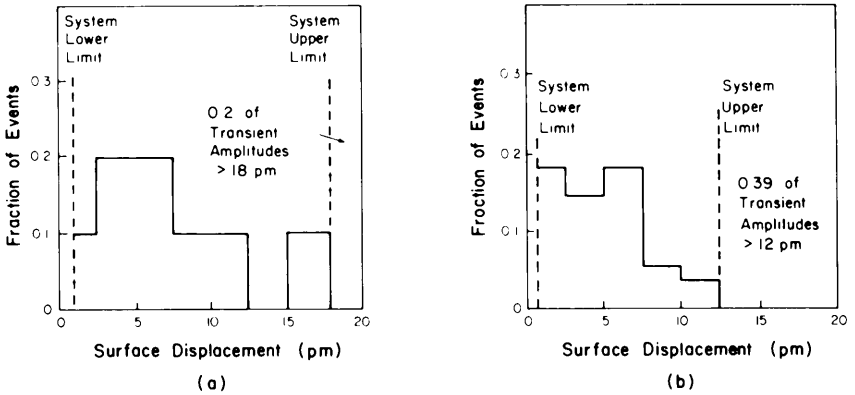


FIG. 30. The distribution of leading edge amplitudes for EN30A steel tested in the (a) quenched and (b) temper-embrittled in the absence of hydrogen. [From Wadley and Scruby (1979a).]

of areas of secondary cracking which covered areas of about the same order as predicted from amplitude measurements. Those possibly could be the main sources of emission in the temper-embrittled condition.

Scruby *et al.* (1979) have analyzed AE transients but have used a source model different from the one discussed previously (Wadley and Scruby, 1979a). The experimental scheme was similar to that described previously. Even though fracture process AE sources are not possible to model in a completely accurate manner, all planar microcrack processes will appear from a distance to be equivalent to the appearance of a small dislocation loop. A crack loaded normally will be equivalent to an edge loop whose strength $b\delta A$ (Burgers vector x loop area) is equal to the crack volume (see Fig. 31). The dislocation loop may be equivalently represented by three orthogonal dipoles (Burridge and Knopoff, 1964). The Green's function for the dipoles in an elastic half-space are known (Pao, 1978), and the loop dislocation source has been treated by Sinclair (1979). For a crack in a half-space, normal displacement measurements at the epicenter can detect only mode I opening, and only one orientation, the inclination of the crack to the horizontal. The strength of the delta function part of the longitudinal arrival of the crack modeled in Fig. 31 is

$$S = (v_s^2/2\delta h v_L^3) b\delta A, \tag{18}$$

where S is the area under the displacement versus time initial pulse, v_s and v_L are the shear and longitudinal velocities, and h is the depth of the source.

The effect of the gauge section was investigated. Three specimens were tested with the notch at the top, middle, and bottom of the 2-mm-

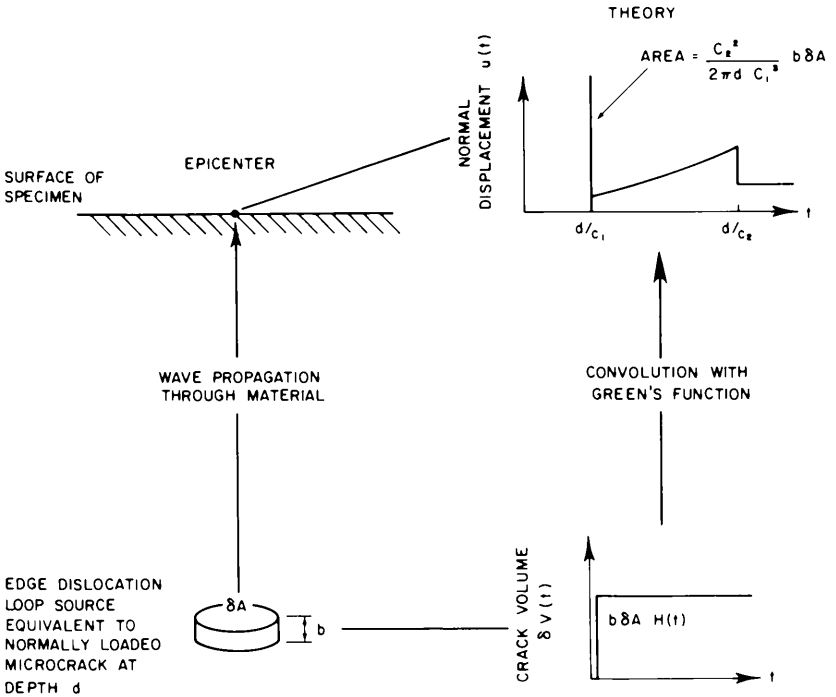


FIG. 31. The dislocation loop model for the source event, and the vertical surface displacement at the epicenter. [From Scruby *et al.* (1979).]

long gauge section. The fracture mode chosen was ferrite cleavage fracture in a mild steel for it generates AE and is metallurgically one of the best defined fracture processes.

Figure 32a shows the waveforms emanating from the notches at various positions in the gauge length. They resemble the theoretical waveform for a dislocation loop source when account is taken of removal of low-frequency components due to filtering. The delta function of the calculated L arrival is replaced by a pulse of finite duration and height, because of finite spatial extent and lifetime of the sources. An echo pulse was consistently present during tests on specimens B and C and never with A is probably due to a reflection from the edge of the gauge section. Deconvolution of the detected displacement with the Green's function for a half-space resulted in a source volume history for each waveform, and the results are seen in Fig. 32b. The lowering of calculated volume to zero and negative values is a result of low-frequency signal removal by filtering. The maximum crack volume obtained (V_1) is shown in Fig. 33. In order to reduce echo effects when comparing the calculated crack volumes from regions A, B, and C, a different volume parameter (V_2) was considered.

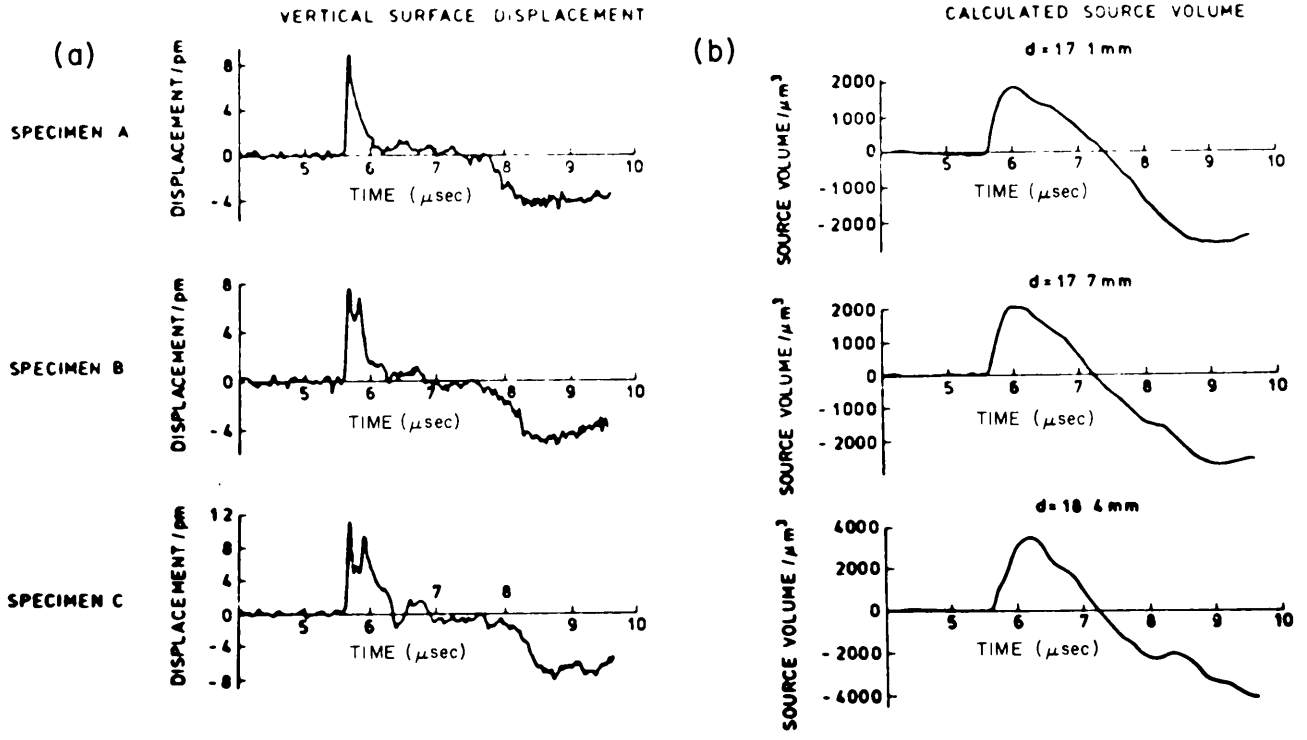


FIG. 32. (a) Measured surface displacement for a typical emission from each specimen. Note echo pulse in B and C, which is absent in A. (b) Surface volume of crack calculated assuming a horizontal dislocation loop model at depth d. The volume drops from a maximum and becomes negative because of low-frequency filtering. [From Scruby *et al.* (1979).]

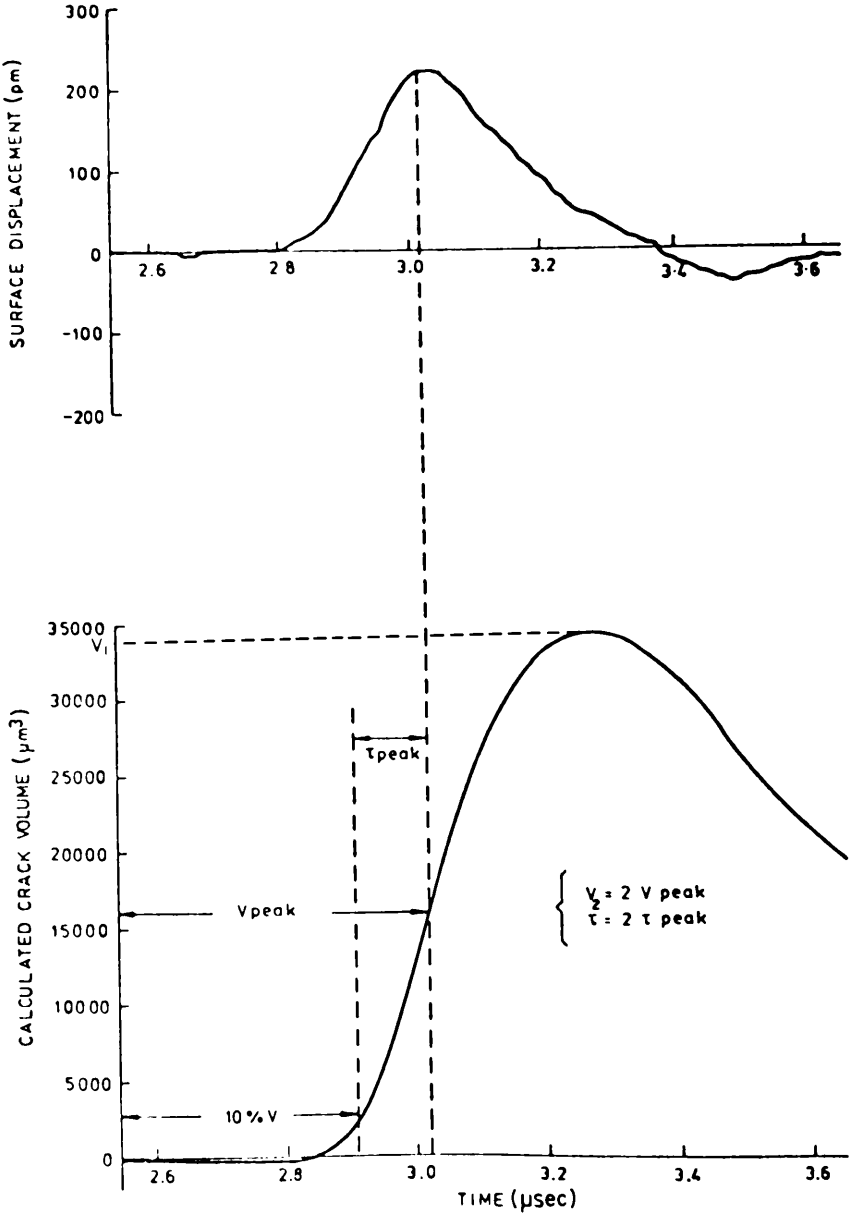


FIG. 33. Determination of the crack volume V_2 and lifetime τ of the growing microcrack. [From Scruby *et al.* (1979).]

The volume was measured at the time of the first peak in the transient and then, assuming that the peak was symmetrical, this value was doubled to give V_2 (refer to Fig. 33). The source lifetime is given as the time τ for the volume to rise from zero to V_2. Certainly there are problems with the ultimate physical interpretation of V_2 and τ, but if histograms of V_2 and τ are presented for A, B, and C, the results are essentially identical (as they should be). This indicates that it is possible to eliminate the effects of the gauge section in source character determination. The typical source event was found to be a microcrack whose volume increased by 600 μm^3 ($\bar{V}_2$) in 88 nsec ($\bar{\tau}$).

If a horizontal, circular elastic crack of radius a is assumed, then with normal (mode I) loading under uniform applied stress σ, the crack will open a distance $2\tilde{b}$ given by (Knott, 1973)

$$\tilde{b} = 2(1 - \nu^2)\sigma a / E, \tag{19}$$

where ν is Poisson's ratio and E is Young's modulus. The volume of the assumed oblate ellipsoidal crack is

$$V = \tfrac{4}{3}\pi a^2 \tilde{b}. \tag{20}$$

Combining Eqs. (19) and (20) and using the parameters $\sigma = 10^9\ N\ m^{-2}$, $\nu = 1/3$, and $E = 2.1 \times 10^{11}\ N\ m^{-2}$, the following is obtained:

$$V = 3.5 \times 10^{-2} a. \tag{21}$$

Thus $a = 26\ \mu m$ for the typical source volume of 600 μm^3 and

$$b = 8.5 \times 10^{-3} a = 0.22\ \mu m.$$

The typical crack area was estimated from fractography as 1000 μm^2. If this were a circular crack, it would have a radius of 18 μm, in very good agreement with that value calculated from the AE analysis.

An estimate can be made of the speed of the crack opening. It can be taken as $2a/\tau = 590\ msec^{-1}$. This value, which ignores acceleration effects, is 18% of the shear wave velocity.

Further source characterization work is described in Wadley and Scruby (1979b). The experimental approach, source model, and analysis techniques were essentially the same as that of Scruby *et al.* (1979). Two types of low-temperature (77°K) brittle fracture processes were studied. The first was cleavage fracture of quenched mild steel, where the microcrack length was controlled by lath packet size. The second was intergranular fracture of a high oxygen electrolytic iron, where the microcrack length was greater than the grain size. Detected AE's and derived crack volume versus time had the same form as those shown in Fig. 32. Table I presents the derived mean and standard deviation for the source volume

TABLE I
ACOUSTIC EMISSION SOURCE MEASUREMENTS FOR MILD STEEL

Specimen	Crack depth (mm)	Number of emissions	Source mean	Volume standard deviation (μm^3)	Source mean	Lifetime standard deviation (nsec)
238	19.7	10	4954	8300	122	37
304	17.5	22	1678	1359	110	30
239	19.7	33	1134	1666	107	46
240	11.6	12	5175	7984	113	30
242	17.8	22	3466	5452	106	46
243	17.5	8	5575	9347	120	63
244	19.8	3	57660	72526	205	115
245	17.3	3	9827	7647	124	13
247	19.7	3	1073	664	104	18
306	19.8	1	3200	—	130	—

and lifetime for the mild steel fractures. Table II gives the same quantities for electrolytic iron. Using analysis similar to that described by Scruby *et al.* (1979), the crack growing time (τ), crack length ($2a$), crack area (πa^2), and crack growth rate were obtained for all detected emissions. Again the data are presented in tabular form. Table III indicates the results for mild steel. There is reasonably good correlation between deduced mean crack diameter and lath size. (With the transducer sensitivity used in the experiments it was estimated that cracks lengths smaller than 30 μm could not be detected.) There is no independent method of determining microcrack

TABLE II
ACOUSTIC EMISSION SOURCE MEASUREMENTS FOR ELECTROLYTIC IRON

Specimen	Crack depth (mm)	Number of emissions	Source mean	Volume standard deviation (μm^3)	Source mean	Lifetime standard deviation (nsec)
99	16.9	2	4320	3480	132	19
100	16.9	26	3047	5871	142	58
101	18.6	22	5346	10242	155	52
102	16.9	7	9226	10639	197	44
103	18.7	14	31213	46740	202	103
104	18.7	14	3703	3799	162	95
106	16.6	55	14977	75519	138	63

TABLE III

MEAN AND STANDARD DEVIATION OF VARIOUS PARAMETERS DEFINING AN AVERAGE MICROCRACK FOR EACH MILD STEEL SPECIMEN

Specimen	Number of emissions	Lifetime (nsec)		Crack length (μm)		Crack area (μm^2)		Crack growth rate ($msec^{-1}$)		Prior austenite grain size (μm)	Lath packet size (μm)
		Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation		
238	10	122	37	75	37	5500	5900	635	262	30	~30
304	11	110	30	56	16	2700	1500	525	121	30	~30
239	33	106	46	56	17	2700	2000	586	190	45	35
240	12	113	30	69	38	4900	5300	597	220	90	80
242	22	106	46	84	41	7000	8400	825	230	105	80
243	8	120	63	71	42	5400	6500	602	235	123	90
244	3	205	115	151	101	25900	28000	675	159	127	100
245	3	124	13	95	44	8500	5600	773	360	131	90
247	3	104	18	50	10	2000	800	494	130	176	95
306	1	130	0	94	0	7000	0	726	0	176	95

TABLE IV

MEAN AND STANDARD DEVIATION OF VARIOUS PARAMETERS DEFINING AN AVERAGE MICROCRACK FOR EACH ELECTROLYTIC IRON SPECIMEN

		Lifetime (nsec)		Crack length (μm)		Crack area (μm^2)		Crack growth rate ($msec^{-1}$)			
Specimen	Number of emissions	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Ferrite grain size (μm)	Crack length/ grain size
99	2	132	19	182	69	29,800	19,600	1333	328	36.1	5.1
100	26	136	53	185	83	32,300	34,000	1410	544	45	4.1
101	22	155	51	151	72	22,000	23,600	983	279	43	3.5
102	7	197	44	220	78	42,700	32,200	1134	342	60	3.7
103	14	202	103	209	139	49,600	58,000	1004	353	45	4.6
104	14	112	55	174	57	26,500	16,500	1206	375	46	3.8
106	55	139	64	188	159	47,700	130,100	1311	469	47	4.0

lifetimes; however, the estimated microcrack velocities are at least in general agreement with other experimental values and theoretical estimates. The estimated parameters for electrolytic iron are given in Table IV. In source modeling studies significant difference was found between the waveform from a deformation twin and a crack, enough so it was concluded that few of the detected emissions were associated with twin formation, However, the distinction between a cleavage crack and an intergranular crack of similar length is not possible. The typical microcrack had a lifetime of 150 nsec, propagated a distance of 185 μm at a velocity of 1230 msec^{-1}. The crack length is about four times the grain diameter and this ratio (last column of Table IV) is relatively constant for all samples even though there is large variability in all the other crack parameters. The crack velocity in electrolytic iron was about twice the value obtained from cleavage fracture of mild steel and may be attributed to the lower resistance offered to a propagating crack by an embrittled grain boundary. [Recall that events seemed to be faster in ductile fracture than in those generated during hydrogen assisted intergranular brittle fracture—see Wadley and Scruby (1979a).] It is now concluded that the histogram approach to source characterization is the most reliable due to the variability of individual source parameters.

The following sections will include much less detail than Section I. Details will be found in the individual references.

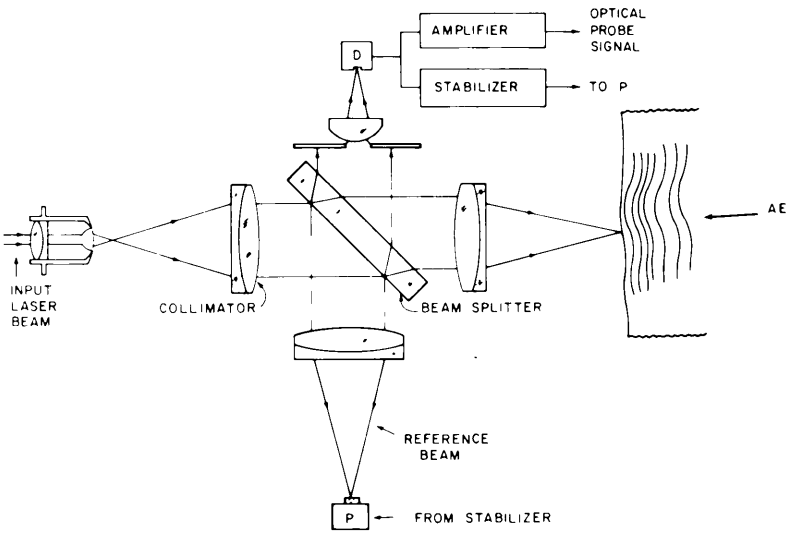
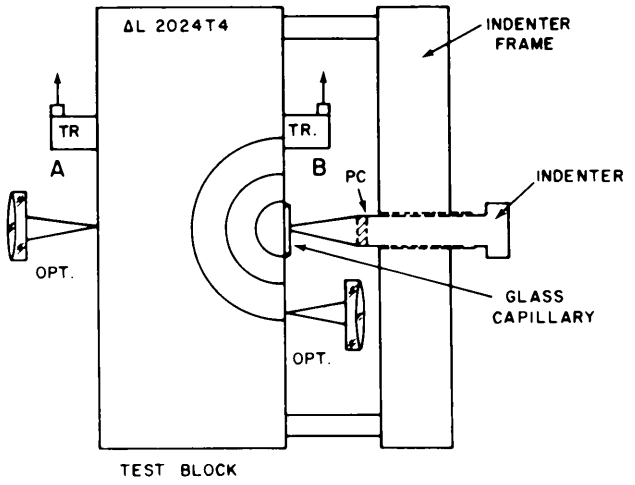
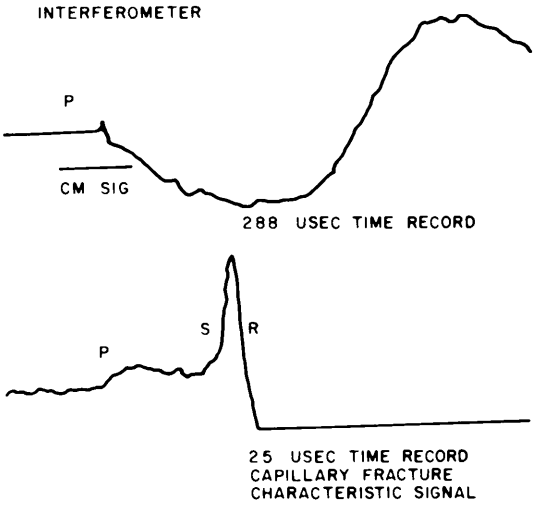


FIG. 34. Interferometric detection of AE with modified Michelson interferometry. (Figure slightly altered from original: AE replaces ultrasonic wave.) [From Palmer and Green (1977).]



(a)



(b)

FIG. 35. Optical (interferometric) detection of AE from breaking glass capillary: (a) experimental setup; (b) results for displacement for 200 m/sec time record, for 25 m/sec time record. [From Djordjevic and Green (1979).]

II. Optical Detection of Acoustic Emissions

Optical sensing of AE transients has been used by the Johns Hopkins group (Green and Duke, 1979; Djordjevic and Green, 1979; Kline *et al.*, 1978; Palmer and Green, 1977; Green, 1977). A schematic diagram of the experimental setup is shown in Fig. 34. The instrument is a modified Michelson interferometer, one of whose paths contains reflections from the surface of the sample. Optical interferometry is used to sense the normal surface displacements produced by the propagating elastic waves. Optical detection has the advantage (as do capacitance transducers) that direct contact to the sample is unnecessary, and hence very wide bandwidth is achieved. Absolute calibration is possible. The active area (focused laser beam) can be as small as a few hundredths of a millimeter. Optical detection can probe internally in transparent media. Figure 35 shows the response of the optical detection system to breaking glass capillary. The system has been used to investigate AE generated in stress corrosion cracking, twinning, thermal shock cracking, and intentional piezoelectric transduction.

III. Signal Analysis

A great deal of work has gone into characterization of AE signals. A complete description of the AE source has not been given by count rate, "ring down" count rate, energy per pulse, energy and amplitude distribution, pulse amplitude, pulse duration, pulse rise time, or others. Ample references in this regard can be found in the general references in the bibliography. Included here are a few recent articles in characterization that bring to bear some of the new techniques in signal processing and handling. Graham (1979), Randall and Graham (1978), and Elsley and Graham (1976) have developed an AE multiparameter analyzer system. Information selected for measurement included shape, magnitude and frequency content of each AE burst, and the time, load level, and linear position on the specimen at which it occurred. At the end of a specimen test the characteristics of each AE event are represented by 23 numbers that can be used in a computer *pattern recognition* routine to define the statistical mix of different AE types as a function of time or test condition. Characterization of signals originally came from manual scanning (and identification of various sources) of tapes of deformation tests in ceramics, metals, and composites. Results are given using the system in graphite–epoxy composite test specimens.

Beattie (1980) has looked into digitization of AE data in a critical manner, investigating methods of retrieving AE information by means of signal digitization and subsequent computer analysis. Covered are signal

digitization, spectral computations with a fast Fourier transform (FFT), the effect of "windowing" on the calculations, and attempts to calculate and use transducer transfer functions. Four experiments were reported using digital processing of AE signals. They were stress corrosion in a U-4$\frac{1}{4}$ wt %-Nb alloy, a three-point bend test of a carbon–epoxy composite, tensile tests of single edge notch specimens of four alloys, and a compression test of a nodular cast iron.

IV. Acoustic Emission during Various Transformations

Steinberg *et al.* (1980a, b) have investigated the thermally induced AE in silicate glasses and metallic glasses. A commercially available AE setup was used. The total gain was about 100 dB and a resonant 175 kHz transducer was used. Three commercial silicate glasses were used; they are

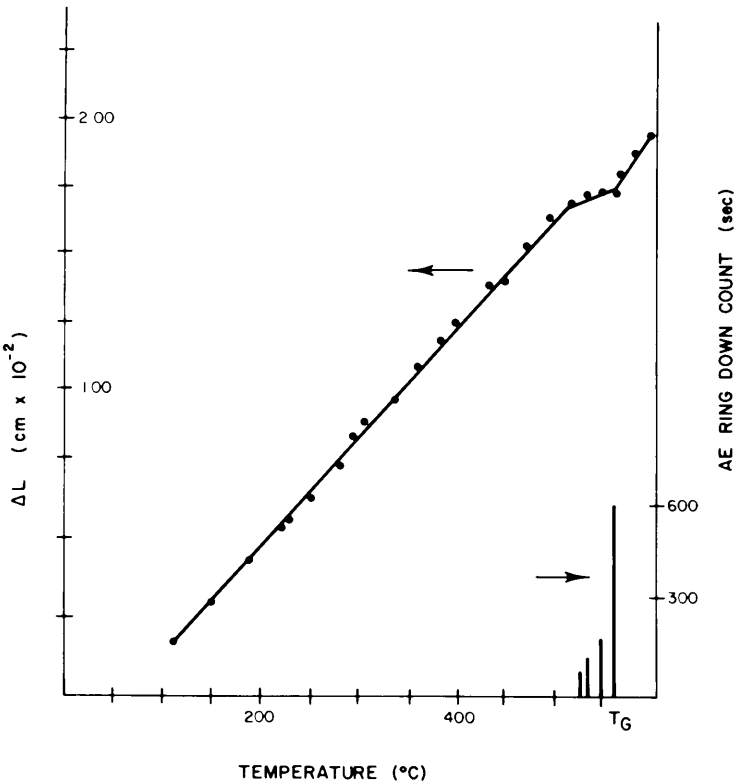


FIG. 36. AE rate and length change as a function of temperature for Pyrex glass. Heating rate about 10°C/min. [From Steinberg and Lord (1980).]

listed below with the "supplier-given" glass temperatures (T_g):

"soft-glass"–alkali lead	$T_g = 435°C$
Pyrex–borosilicate	$T_g = 565°C$
aluminosilicate	$T_g = 715°C$

In each case bursts of AE activity were first detected (upon heating) at T_g. Figure 36 shows the results for Pyrex. The length measurements also show a densification (shortening) at T_g. One possible cause of the AE is the generation of elastic waves during the escape of free volume during structural relaxation as predicted by Hirai and Eyring (1958).

Metallic glasses are rapidly cooled alloys that have no long-range order (Gilman, 1975). The lack of crystallinity leads to many interesting and unique properties, e.g., high hardness, high yield point, and very soft magnetic properties. Figure 37 shows the results of some thermally induced AE measurements in the metallic glass $Fe_{40}Ni_{40}P_{14}B_6$ tapes. The self twist θ is also shown. There appears to be strong correlation between any change in self twist and AE activity. Both must reflect the same internal process. The AE from about 150 to 275°C (Fig. 38a) may result from the recovery of secondary cooling stresses (Egami, 1978). The activity from 275 to about 350°C (Fig. 38b) may result from the relief of primary quenching stresses

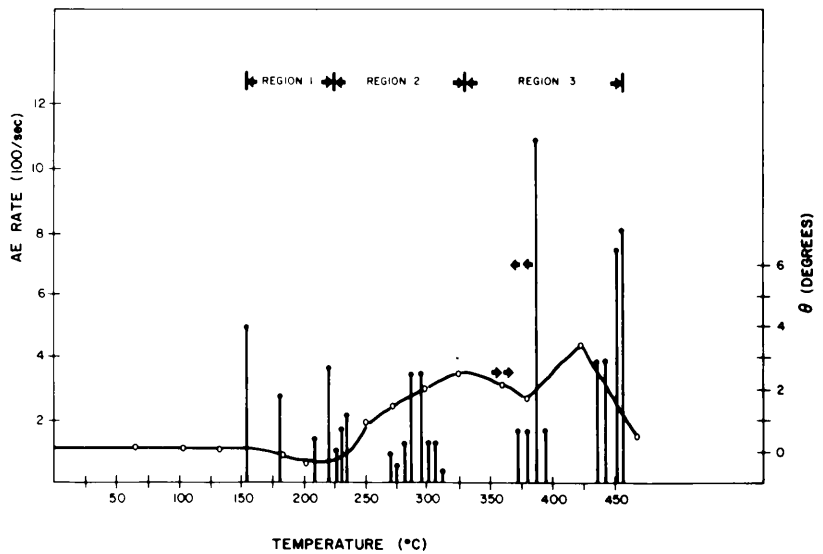


FIG. 37. AE rate and self-twist (ST) in $Fe_{40}Ni_{40}P_{14}B_6$ (Metglas 2826) as a function of temperature. Heating rate about 10°C/min. [From Steinberg *et al.* (1980).]

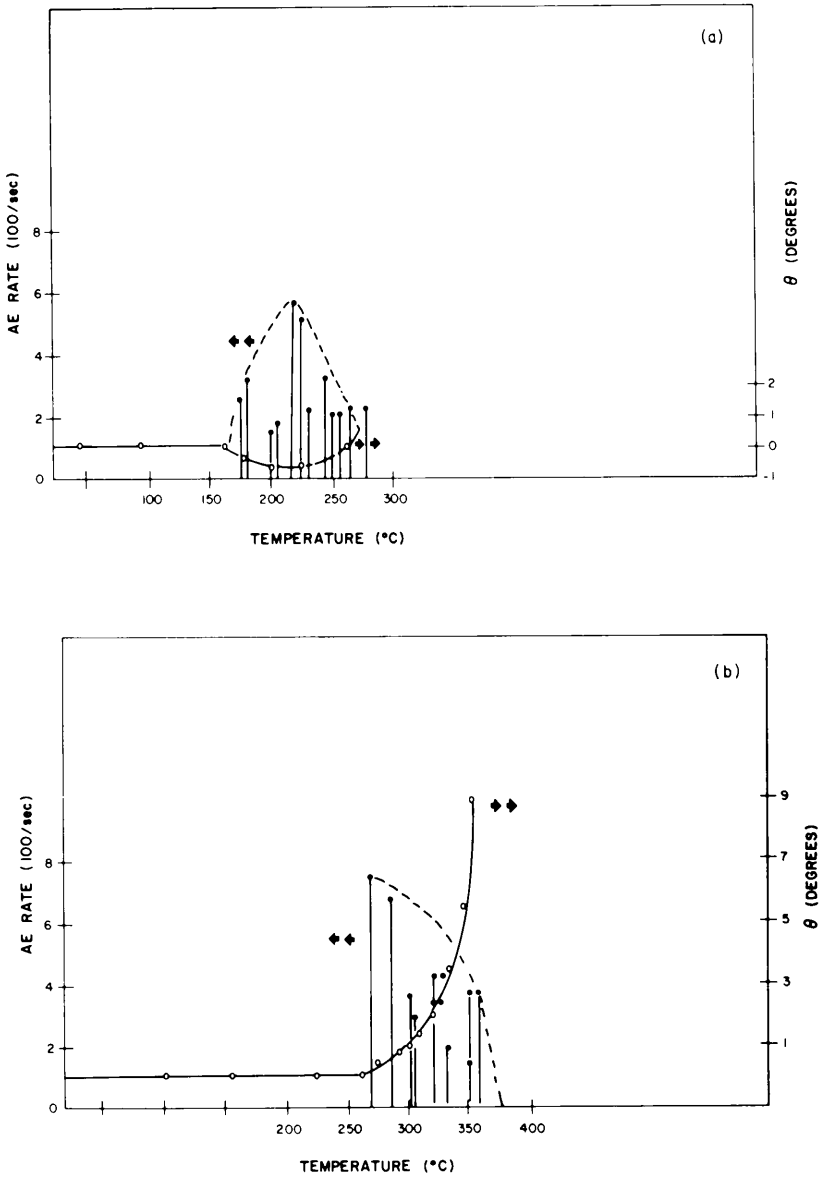


FIG. 38. AE rate and ST in Metglas 2826 as a function of temperature. (a) Run from room temperature to 275°C. (b) Run from room temperature to 350°C. (c) Run from room temperature to 475°C.●—, AE:○---, self-twist. [From Steinberg *et al.* (1980).]

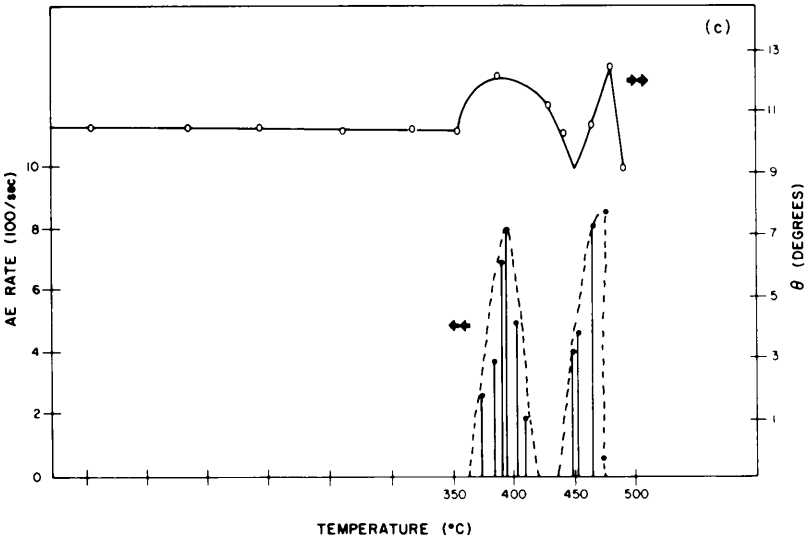


FIG. 38. *Continued.*

(Egami, 1978; Luborsky and Walter, 1978). The activity from 375 to 410°C (Fig. 38c) is due to crystallization of the alloy. The activity starting around 450°C (Fig. 38c) seems due to crystallization of a second phase. Note in all cases (as depicted in Fig. 38a–c) AE activity does not commence until the upper limiting temperature of the previous run is reached—something like "thermal Kaiser effect." From this preliminary work the following conclusions were drawn:

(1) That we are able to detect AE at 175 kHz means that at least a portion of the relaxation processes involved occur impulsively on a time scale on the order of 10^{-5} sec.

(2) The self-twist or torsional creep and acoustic emission seem to be strongly correlated and are probably caused by same physical process in amorphous $Fe_{40}Ni_{40}P_{14}B_6$.

(3) It may be possible to use an AE measurement as a quality control technique for rapidly quenched metals to determine the effects of secondary cooling and residual stresses.

(4) It is possible to observe the glass transition temperature of silicate glasses by means of AE, which could be free volume diffusing to the surface.

(5) Metallic glasses are much more emittive on a per volume basis than silicate glasses.

Hunderi and Lönvik (1978) have performed thermally induced AE measurements in metallic glass tapes. They call these measurements "thermosonimetry." Their data for crystallization temperature and kinetics correlated very well with differential scanning calorimetry (DSC) results. AE activity was observed below the crystallization temperature as in Steinberg *et al.* (1980a, b).

Papa *et al.* (1975) and Stagni and Castellano (1977) have investigated the AE generated during rapid heating of tin single crystals. It appeared that they were able to detect the $\alpha \leftrightarrow \beta$ crystal structure change with AE measurements, but they could not verify the actual transformation with independent (i.e., non-AE) measurements. The heating rates used were much in excess of those usually needed to observe the structural change in tin.

Shiraiwa *et al.* (1978) monitored the AE generation during thermal cycling of silica firebrick. There was significant AE generation upon passing through the various crystal transformations in silica. Much higher amplitude emissions were produced when there was cracking in the bricks.

Cannelli and Cantelli (1979) have investigated temperature-stimulated AE in the niobium–hydrogen system. The hydride precipitation problem was studied. After severe precipitation, and a certain degree of deterioration of the material, AE bursts are stimulated by hydride formation and persists after high-temperature annealing. The AE may be attributed to the nucleation and growth of nonrecoverable defects. Northrup *et al.* (1977) and Kass *et al.* (1977) have looked into the effects of hydride formation and AE in Nb, FeTi, $LaNi_5$, and UA_2. Some studies went to as high as 20 MPA (3000 psi), and temperatures from room temperature to 150°C were used. Bursts of AE accompanied the formation of a hydride, and the AE was used to determine phase boundaries. Cracking of the hydride as the reaction proceeds may be a cause of the emissions.

The martensitic transformation has been studied using AE techniques. A very interesting article is that by Beattie (1972) where the fcc $\rightarrow$ fct phase transition in In–23 at. %Tl single crystals was investigated. In this work the total energy was determined as a function of both polarization and wave propagation direction with respect to the plane of transformation of the sample. The frequency content of the AE was also determined. Even though the work did not give the exact source mechanism of AE generation it placed definite limitations on the types of mechanisms which could generate the signals. A very recent article on thermoelastic phase transformations investigated with AE is Baram and Rosen (1979).

McBride *et al.* (1974) have monitored AE while a thin film sample (5000 Å $\times$ 1.5 cm^2 area) of vanadium dioxide was cycled through the semiconductor–metal transition ($\sim$68°C). AE was observed to occur

during and after the electrical transition (as determined by electrical resistivity measurements) on heating and cooling. Thus the unit cell change that coincided with the resistivity change is not the chief cause of AE. It is possible that twinning could be the cause of the AE.

Beattie *et al.* (1974) have studied the electrical switching in bulk semiconducting chalcogenide glasses by means of AE techniques. During switches where the current rise was faster than 30 μsec, a burst of AE is produced some 2–25 μsec prior to the rise in current. This AE is associated with the initiation of the switching process. Another type of AE, which occurs after the rise in current, is shown to have its origin in propagating cracks in the sample. It appears that AE can give information about both the physics of the switching process and the integrity of the material after the switch.

Dielectric breakdown has been monitored with AE techniques. Although not strictly a transformation per se, it does represent a change of state as far as conductivity of a material is concerned. Howells (1974) determined the AE from electrically stressed dielectric liquids (e.g., PCB and mineral oil). Howells and Norton (1977) made a study of AE from transformers and were able to separate partial discharges from other noise sources like magnetostrictive, Barkhausen, and background 60Hz. It is possible that the occurrence and *location* of the discharges can be detected. Garnier *et al.* (1976) have also detected dielectric breakdown in insulating materials with AE techniques.

V. Dislocation Effects

The area of AE generation due to dislocations is a very active area of research. This interesting area was reviewed thoroughly (up to the end of 1978) by Carpenter and Heiple (1980). No attempt will be made to go into detail on dislocation effects in this review. The reader is referred to the excellent review by Carpenter and Heiple. Their work analyzed the characteristics of the emissions and attempted to identify source mechanisms. The following effects were analyzed: crystal structure, purity, grain size, heat treatment, prior mechanical work, strain rate, test temperature, stacking fault energy, sample size, difference in AE obtained during tensile, compressive and biaxial deformation, and the method of taking data (ringdown counts, burst rate, or rms voltage).

Hatano (1975) determined the autocorrelation function of the "continuous AE" from aluminum and showed that it can indeed be accounted for by random pulses.

It has been shown that it is extremely helpful to measure another structurally sensitive parameter in addition to AE (Carpenter and Higgins,

1976; Higgins and Carpenter, 1978). In these studies dislocation damping (internal friction) was measured at the same time as AE. The dependence of dislocation damping in the kHz frequency range on dislocation density and average dislocation loop length is well known and can be used to identify particular sources of AE. Figure 39 gives a schematic representation of the types of data that can be collected during a tensile test (including dislocation damping). The use of AE measurements in both tension and compression and simultaneous dislocation damping measurements has allowed the identification of precipitate (inclusion) fracture as a source mechanism in 7075-TS aluminum (Carpenter and Higgins, 1977).

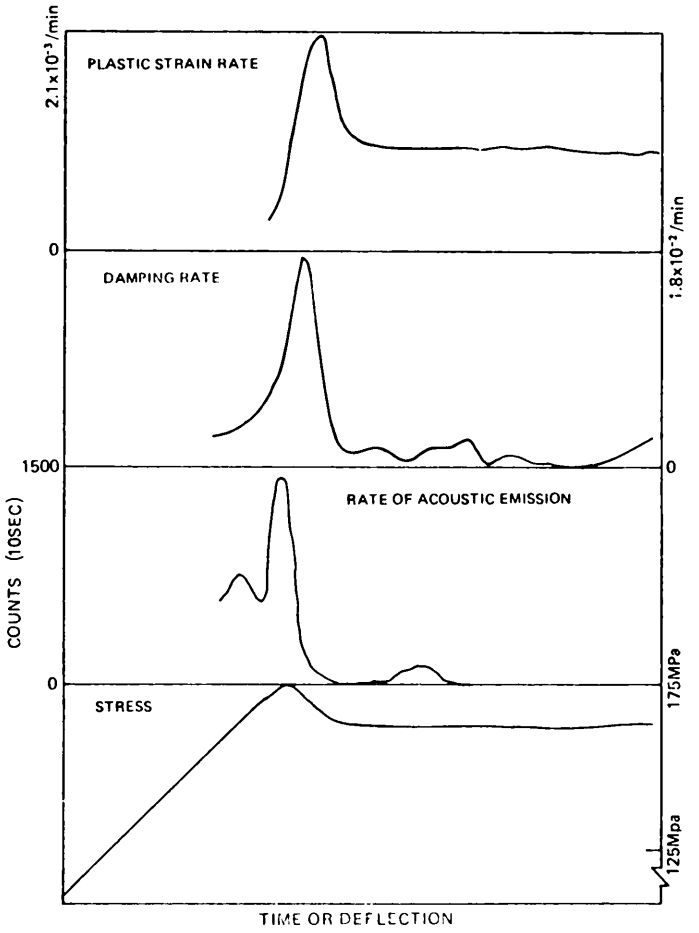


FIG. 39. Plastic strain rate, damping rate, AE rate and stress versus time for Armco iron. [From Carpenter and Higgins (1976).]

The source had previously been widely attributed to dislocations. There has been other work on the AE due to inclusion fracture (Ono *et al.*, 1977; Bianchetti *et al.*, 1976).

The theoretical models for AE generation due to dislocation movements are discussed in detail in Carpenter and Heiple (1980).

VI. Magnetomechanical Acoustic Emission

Lord (1975) was the first to show that ferromagnetic material (nickel in particular) emit significant AE while being magnetized. This is postulated to be due to abrupt and discontinuous displacement of domain walls. It was termed the "acoustic Barkhausen effect" (or later the "magnetomechanical AE") because of the many similarities to the magnetic effect of that name. Lord's work used dc magnetic fields, but the AE was very dependent on the rate of field application. Kusanagi *et al.* (1979) showed that significant AE was generated during ac (50-Hz) magnetic field application in nickel and mild steel. They were the first to report that the magnetomechanical AE is very dependent upon applied stress. Figure 40 shows their results for AE (as rms volts) versus magnetic field at two different levels of stress. (They worked with a 250-kHz resonant frequency transducer.) Figures 41 and 42 show the AE as a function of stress at two different drive fields for mild steel and nickel, respectively. (The materials were annealed before the runs.) It is seen that the stress dependence is

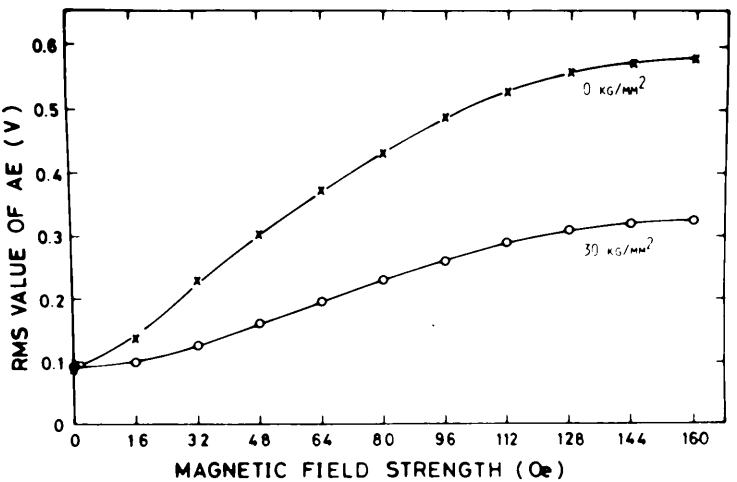


FIG. 40. AE (rms volts) as a function of ac (50-Hz) magnetic field drive in mild steel for two different values of applied stress. [From Kusanagi *et al.* (1979).]

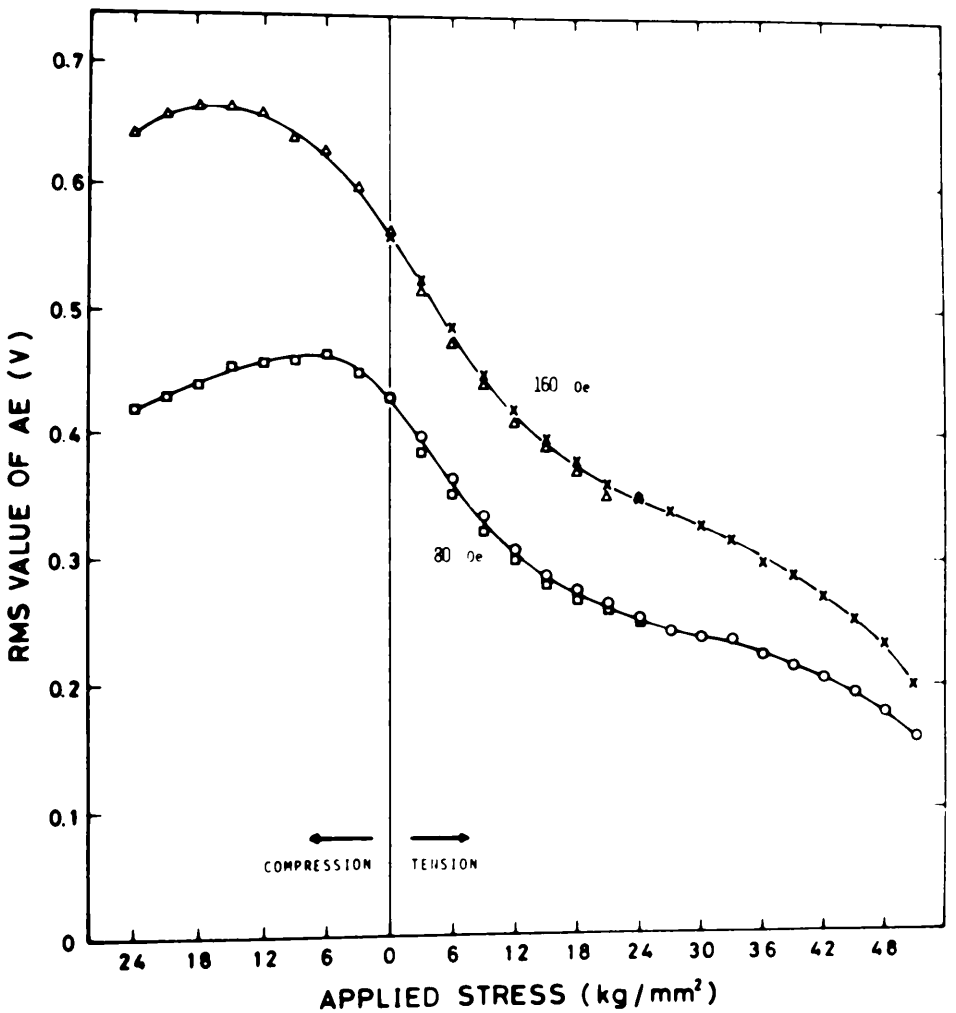


FIG. 41. AE (rms volts) as a function of applied stress in mild steel for two different ac magnetic field drives. [From Kusanagi *et al.* (1979).]

quite complicated, which indicates that a great deal more work is needed before the acoustic Barkhausen effect can be used for residual stress determination.

A more detailed study of the magnetomechanical effect has been made by Ono and Shibata (1980). Systematic studies of the magnetic field and applied stress dependence of the effect in several carbon steels, A 533 B low alloy steel and a commercially available pure iron in the annealed

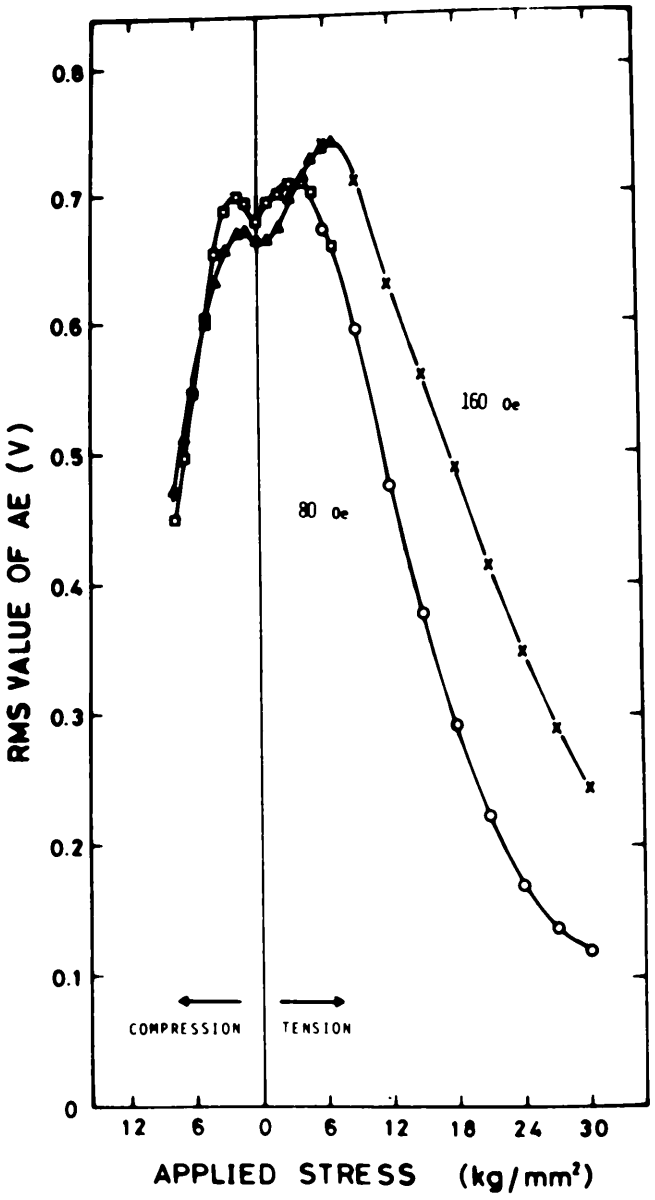


FIG. 42. AE (rms volts) as a function of applied stress in nickel for two different ac magnetic field drives. [From Kusanagi *et al.* (1979).]

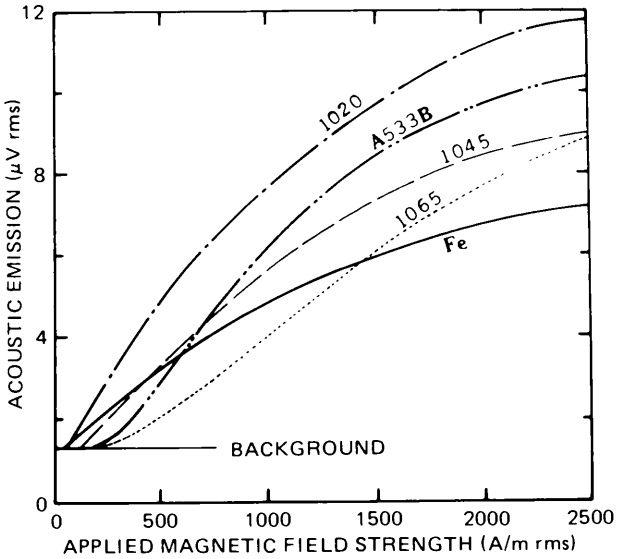


FIG. 43. AE (rms volts) versus applied ac (60-Hz) magnetic field drive for five materials. AC 175L. $\sigma^A = 0$. [From Ono and Shibata (1980).]

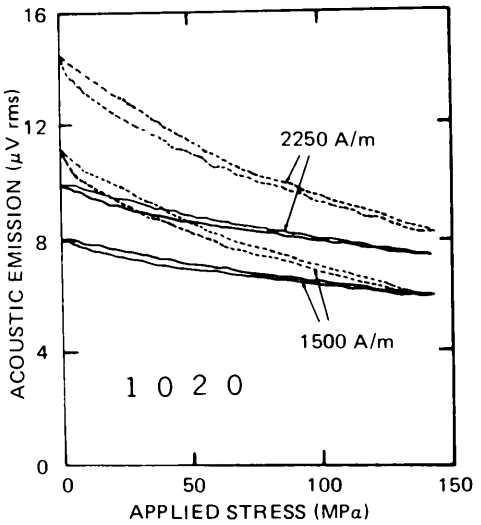


FIG. 44. AE (rms volts) versus applied stress for 1020 steel at two magnetization levels. Note hysteresis on loading and unloading.—, MAC500; —, AC175L. [From Ono and Shibata (1980).]

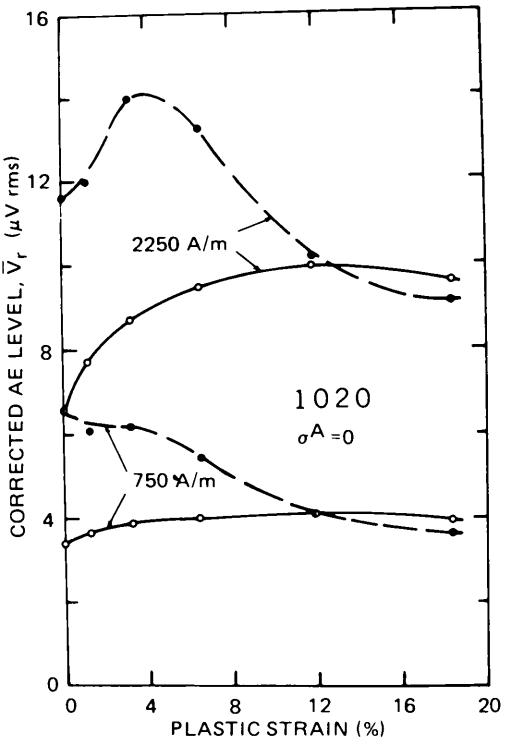


FIG. 45. AE (rms) versus previous plastic strain for 1020 steel. Applied stress was absent. —, MAC500; ---, AC175L. [From Ono and Shibata (1980).]

and normalized condition were performed. The drive field was 60 Hz, and two AE transducers were used (175 and 500 kHz). The effect of carbon concentration, plastic deformation, and microstructure were also investigated. The 1020 steel had the highest magnetomechanical AE response of all the materials tested (refer to Fig. 43). Figure 44 shows the stress dependence in 1020 steel. Figure 45 shows the effect of prior plastic strain on the magnetomechanical AE output. In a somewhat practical sense, welds in mild steel plate and bolts of 1045 steel showed an AE-stress sensitivity of about 0.25%/MPA—probably sufficient for residual stress determination. The use of data from two transducers (at different frequencies) was of help in determining structural characteristics (e.g., residual stress). Again the stress dependence was most complicated, indicating that significant work needs to be done before it can become a viable technique for material characterization.

The author has done some work in this regard in metallic glasses. The amorphous alloy $Fe_{40}Ni_{40}P_{14}B_6$ was investigated. A 175-kHz transducer was used, and the overall system gain was about 95 dB. Ring down count

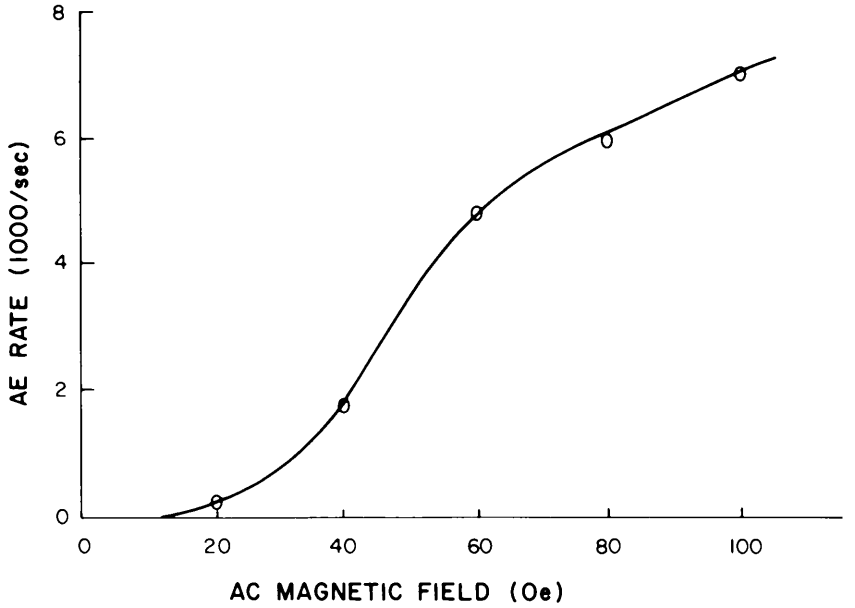


FIG. 46. AE rate versus ac (60-Hz) magnetic field for Metglass 2826 ($Fe_{40}Ni_{40}P_{14}B_6$). Stress = 2×10^7 N/m^2. [From Steinberg *et al.* (1980).]

rate will be displayed, but essentially the same results were obtained with AE rms voltage. Figure 46 shows AE rate as a function of ac magnetic field drive. The stress dependence of the AE at room temperature and 210°C is found in Fig. 47. Figure 48 gives the AE as a function of time of hold at 210°C. Comparing Figs. 47 and 48 at face value, it can be argued the quenched-in stress has been relieved from about 10×10^7 to 9×10^7 N/m^2 (Point A $\rightarrow$ Point B on Fig. 47). This value of stress relief is in order-of-magnitude agreement with that found by other methods (Fisk and Lord, 1980; O'Handley, 1976). This result must be considered as a very preliminary (and promising) application of AE to residual stress determination. There are, however, other explanations of this time-dependent result. [Evans (1975) has used AE to determine residual stress in ceramics.]

Matzkanin *et al.* (1979) have discussed the magnetic Barkhausen effect in great detail. They point out that from the standpoint of practical application of nondestructive testing, the acoustic Barkhausen effect has a definite advantage over the magnetic effect. When inductively sensing the Barkhausen effect from the surface of a specimen, the effective measurement depth is limited by eddy current shielding to a fraction of a millimeter. Thus it can only be used to investigate surface condition of ferromagnetic materials. On the other hand, the elastic waves (AE) generated by the

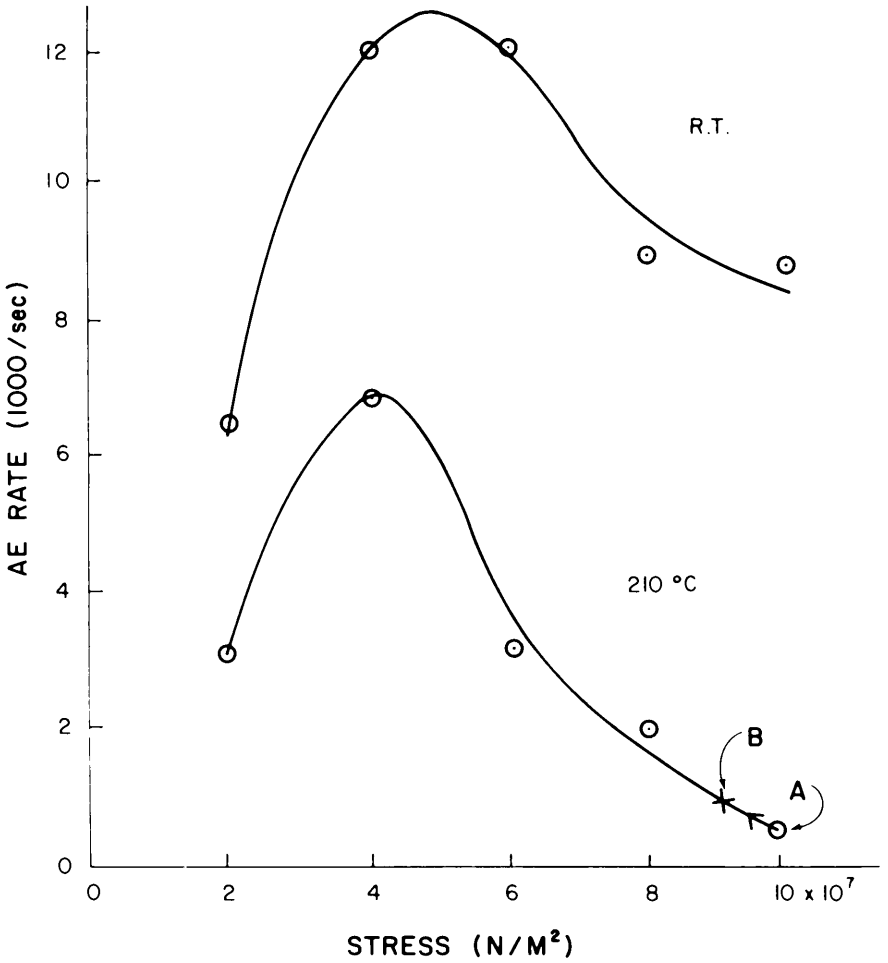


FIG. 47. AE rate versus applied stress for Metglass 2826 (at 60 Hz) at room temperature and 210°C. Drive field = 3A (AC) = 60 Oe. [From Steinberg *et al.* (1980).]

Barkhausen jumps should travel essentially unimpeded, so the AE from all the sample volume are detected. Of course, if the residual stress state has a very complicated spatial dependence, and the AE versus stress is complicated, this great depth of penetration may not be an advantage.

VII. Monitoring Fatigue Damage with Acoustic Emission

The use of AE in monitoring fatigue damage is discussed in Green (1979), Duke and Green (1978, 1979), and Green and Duke (1979). A very good review of the earlier literature is given here. These authors used a specially

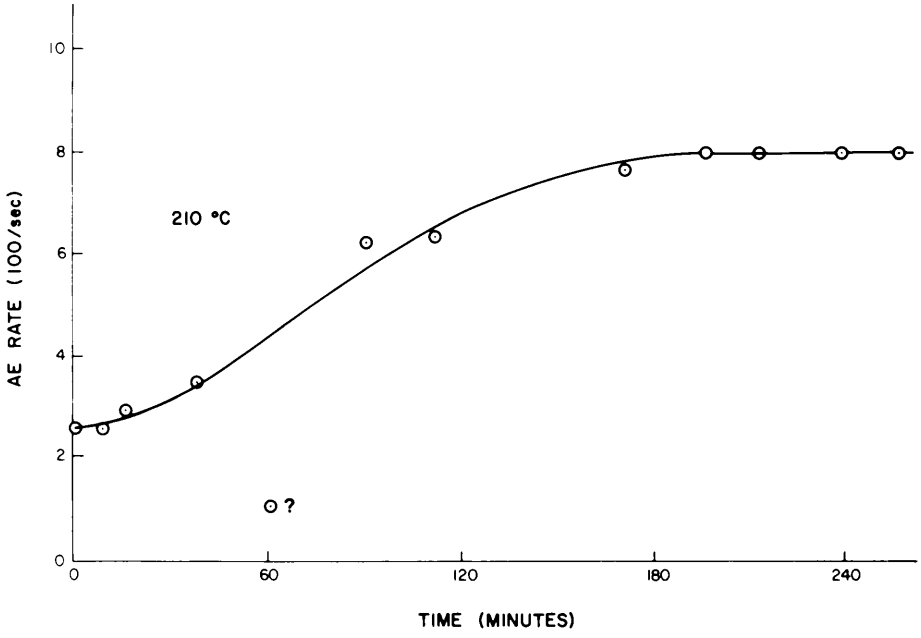


FIG. 48. AE rate versus time at constant *applied* stress and constant magnetic field in Metglass 2826 versus time at 210°C. Stress = 10×10^7 N/m^2; drive field = 3A (AC) = 60 Oe. [From Steinberg *et al.* (1980).]

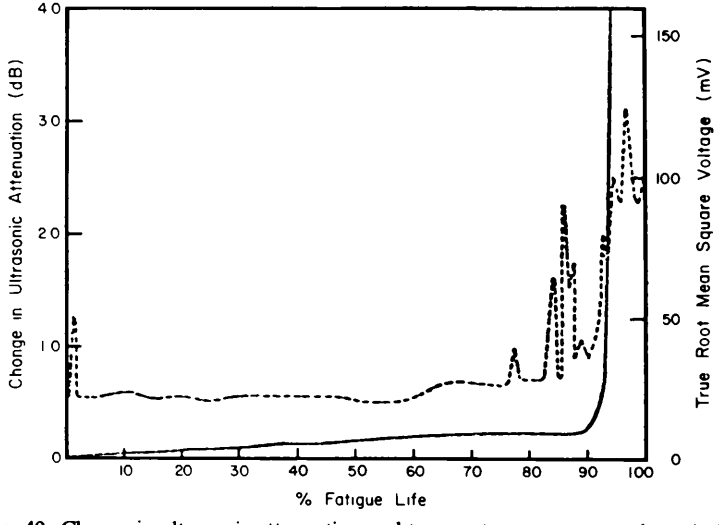


FIG. 49. Change in ultrasonic attenuation and true root-mean-square voltage (---) of AE versus fatigue life for solution-treated 7075 aluminum. [From Green (1979).]

designed system that allowed AE measurements to be made in spite of the high-level background due to the fatigue testing machine and associated grips (response of the system was centered about 300 kHz). Figure 49 shows the results of an AE-fatigue run on 7075 aluminum. AE gives warning of the fatigue failure at about 80% of the fatigue life. Green and Duke also measured ultrasonic attenuation (in the 5–10 MHz range) as well as AE. Figure 49 shows the attenuation results also. At 90% of the fatigue life there is a dramatic change in the attenuation. These changes occur well before any new echoes occur from fatigue cracks in the sample.

VIII. Practical Applications

As in many other fields, the fundamental AE work lags far behind the practical area. One of the most documented practical applications is the testing of large pressure vessels in the nuclear and chemical industries (refer to the general references). Pipeline integrity has also been investigated with AE. At the recent International AE Conference in Anaheim, California, in September 1979, there were indications of other interesting practical applications emerging (some of these have been going on for a while).

In-flight AE monitoring of critical parts of the airframe seems to be a promising area. This is difficult and demanding work, but extremely important in preventing aircraft crashes from fatigue crack propagation [refer to Bailey (1979), Hutton and Skorpik (1979), and McBride (1978)].

An interesting and novel application of AE is to the measurement of coolant flow in reactor channels (Leemans *et al.* 1979). The method can be used while the reactor is in operation, and is nonintrusive. Frequency spectra and pattern recognition techniques were used. Results were promising and indicated that both quality and flow rate could be estimated satisfactorily and independently.

Swain (1979) reported on the AE testing of cranes and lift equipment by his firm over a two-year period. AE results were verified by x-ray or other techniques. Amplitude distribution studies were made, threshold settings were established, and, ultimately, procedures for testing each type of aerial lift equipment were established. Many hundreds of pieces of equipment have been tested; this is indeed a novel and needed application.

AE techniques are finding use in production lines of the electronics industry. Drew and Seitz (1979) reported on the use of AE in the quality control of ceramic components at Seimens in West Germany. Both electrical and mechanical (production) phase testing are monitored.

An extensive amount of work has been done by the Western Electric Company in monitoring ceramic stability and integrity of solder joints and

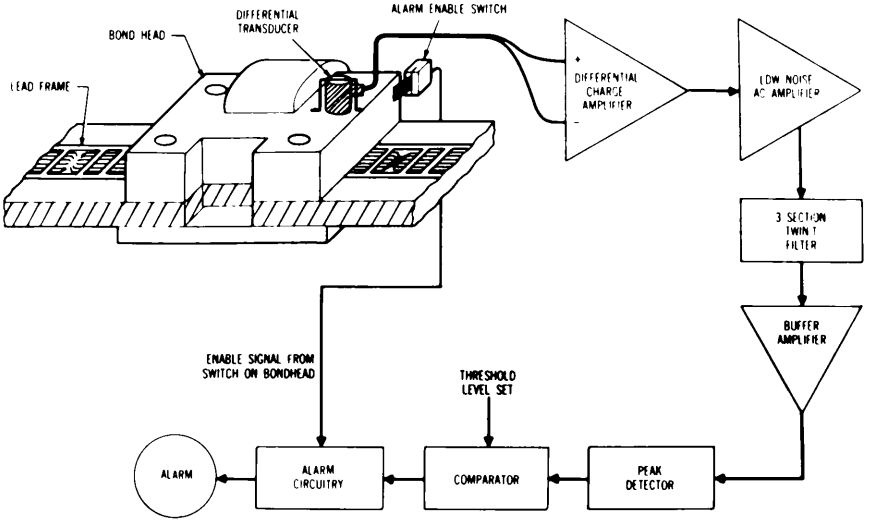


FIG. 50. AE crack detection system for monitoring thermocompression bonding of integrated circuit lead frames to brittle ceramic substrates. [From Jon *et al.* (1979a).]

welds in production line situations. A major portion of the *Western Electric Engineer*'s October 1979 issue is devoted to this acoustic emission work. This is definitely recommended reading for an indication of what can be done with AE in small components quality control. Kahn and Miller (1979) and Jon *et al.* (1979a) have developed an AE detection system for ceramic substrate cracking during thermocompression bonding of integrated circuit lead frames to brittle ceramic substrates. Figure 50 shows a schematic diagram of the crack detection system. A transducer of 650 kHz or 1 MHz is used with appropriate filtering. Peak detection is used. A dc voltage proportional to the maximum peak-to-peak amplitude of the AE burst is compared to a preset dc level. If the burst level is too high, an alarm goes off and the part is automatically marked as being defective. Gahr and Payne (1979) and Jon *et al.* (1978a, b, 1980) have used AE techniques to monitor the quality of resistance and spot welds. A transducer of 1 MHz or 700 kHz was used and a passband (300–700 kHz) or high pass (600 kHz) filter was also employed. It was found that the AE generation during the welding period did *not* correspond to weld quality. However, the AE generation during the cooling period, determined in a "window-of-time" starting some few milliseconds after the initiation of the current for the weld, corresponded well to the weld quality. The stronger the weld, the more the AE generation. This is shown in Fig. 51. It is important to note that the general result of Fig. 51 applies to various

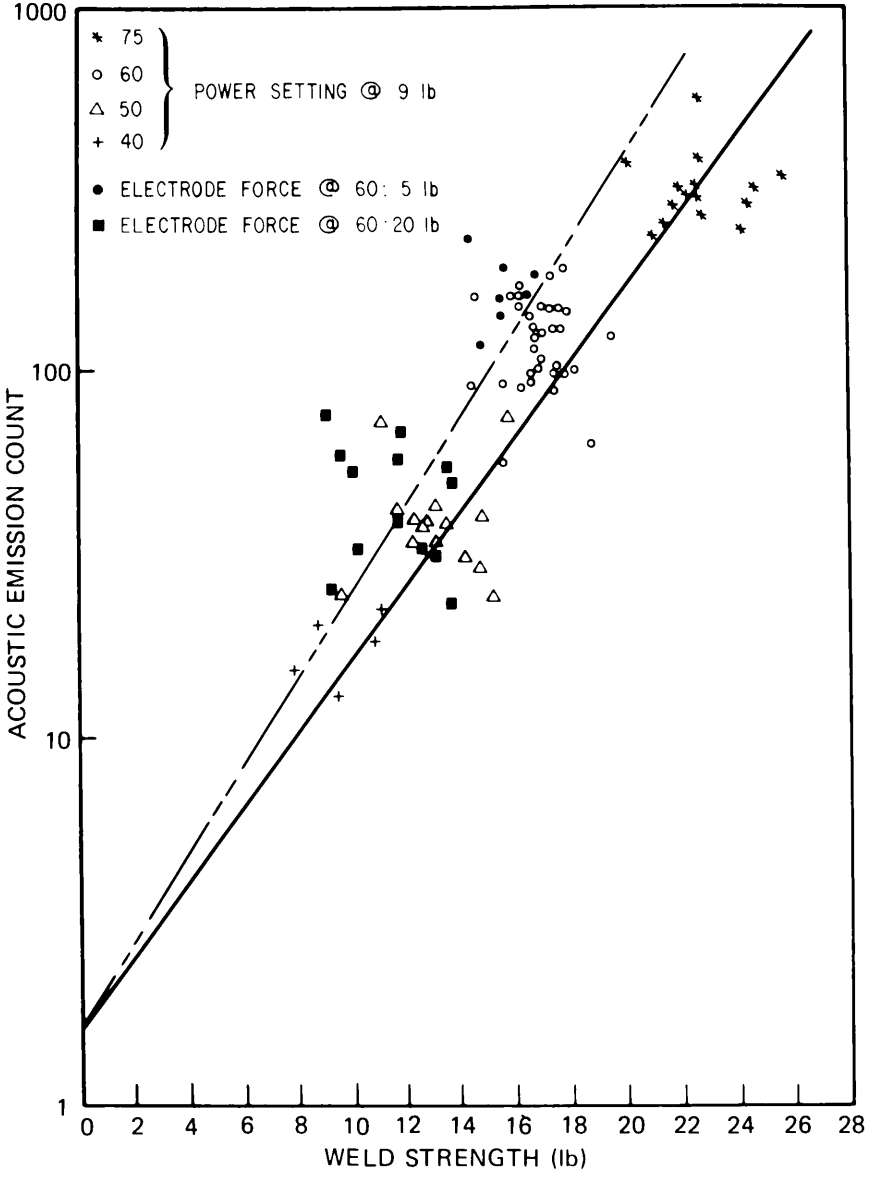


FIG. 51. Relation between AE count and weld strength at various power settings and electrode forces. [From Jon *et al.* (1980).]

mechanical loads and power settings on the welder. Jon *et al.* (1979b) and Carlos and Jon (1978) have applied AE techniques to the quality control of the manufacture of the very critical undersea repeater used in the undersea cables. The repeater is used in boosting the transmission signals in the undersea cable where maintenance is impossible and replacement extremely expensive. AE monitoring is now used on a production-line basis to monitor the structural integrity for oil-filled, paper capacitors in the repeater and to ensure a more certain sealing of the repeater's waterproof housing. (The repeaters can be under as much as 11,000 psi water pressure on the ocean floor.) Due to the continuous current during the soldering operation, "windowing" was impossible—the cracking had to be distinguished from the noise by special signature-recognition techniques. Using the AE system, it was possible to detect internal cracks that were not found with existing NDT inspection techniques. A system was also developed to monitor the integrity of the pinchweld that seals the pressurized nitrogen gas in the repeater housing. The details of the signal analysis are given in Jon *et al.* (1979b). The field trial showed that the method was equally as good as the previous "radioactivity leak" method, and much easier to use.

Laser welds have also been monitored with AE techniques (Jon 1979). There are references in this article for work involving the generation of elastic waves in materials due to laser impingement. [Egle and Brown (1976) have reported on using laser beams to produce transient elastic waves.]

IX. Discussion

It is certain that AE will continue to be used in the currently found applications (scientific and technological) and be applied to many more processes in the lab and used to monitor the stability of many more types of structures (large and small). Undoubtedly more will be learned about materials and structures from AE.

If AE is ever to be universally accepted as a nondestructive testing technique (e.g., in the ASME code), it is essential to understand why growing cracks in some materials emit many AE's, while in other materials growing cracks emit hardly any AE's. This would seem to be a research area of absolutely top priority. [Scruby *et al.* (1979) and Wadley *et al.* (1979) address themselves to this point.] Both wide band system and resonant transducer system work is needed. It must be found if certain crack growth and fracture modes do not emit. The effect of micro-structure on AE should be researched a great deal. (The Harwell group is active in this area.)

The experimental determination of Green's functions for difficult geometries might be a very large help in guiding and checking the results of theoretical calculations. The work of Sachse and Ceranoglu (1979b) for off-epicenter displacement of the plate is an indication of what can be done. A next step would seem to be the cylinder, where the Green's functions are very difficult to obtain analytically. This is an extremely difficult area in all respects, but one of extreme fundamental importance. If it could be "pulled off," it would be a truly classic use of AE in the fundamental theory of elasticity.

The use of multiple capacitive transducers at suitable positions on an appropriate large structure might possibly give added information to describe the source more fully. The ability to measure transverse (shear) displacements at a surface would be a definite help in interpreting AE source characteristics.

The work of Ceranoglu and Pao (1980a, b, c) on numerical evaluation of transient waves generated by dynamic "nuclei of strain" is most important. This type work must be performed if meaningful AE source characterization is ever to be accomplished. It appears, however, that a unique characterization of the general AE source is not yet possible from the detected displacement, due to the variability of the displacement with source type, orientation, time dependence, etc. Possibly, the use of multiple sensors (both longitudinal and shear) and a large catalog of theoretical results (of the Ceranoglu and Pao type) might make source characterization a reality. This is work of the utmost importance and unfortunately of the utmost difficulty. It might be possible to learn about the time dependences of source processes without actually knowing the exact nature of the forces involved. This could be a unique application of AE and could very well produce information obtainable by no other experimental technique. Of course, this type of work will require great attention to experimental and theoretical detail.

The magnetomechanical effect has real possibilities in being of value in determining residual stress in ferromagnetic materials. New methods of determining residual stress are definitely needed.

The fatigue crack problem has such disastrous consequences in structural failures that work should definitely continue here on critical structures (like in-flight monitoring).

All the modern techniques of signal analysis and pattern recognition should be applied to AE analysis.

ACKNOWLEDGMENTS

I should like to thank Warren Mason for allowing me to write this update. Bob Koerner of Drexel has been a steadfast colleague in the AE area. John Brugger of EPA has been most generous in his encouragement and support. The atmosphere in my department and college at

Drexel has been conducive to getting work done. During the latter stages of writing, discussions with Y. H. Pao and Wolfgang Sachse proved most helpful while the author was on sabbatical stay at Cornell University. Special thanks are due my wife, Rose Marie, and children, Susie and Kathy, for their love and patience.

References

Bailey, C. D. (1979). Aircraft in-flight acoustic emission systems: some unique features, Presented at *Int. Conf. Acoust. Emission, Anaheim, California, September.*

Baram, J., and Rosen, M. (1979). Some observations on acoustic emission generated during thermoelastic phase transformation in Al–Ni and Ti–Ni alloys, *Scr. Metall.* **13**, 565–568.

Beattie, A. G. (1972). Characteristics of acoustic emission signals generated by a phase transition, *Ultrason. Symp. Proc.* IEEE Cat. No. 72 CHO 708-850, pp. 13–17.

Beattie, A. G. (1980) Studies in digital analysis of acoustic emission signals, *in* "Fundamentals of Acoustic Emission," (K. Ono, ed.). Materials Dept., UCLA.

Beattie, A. G., Johnson, R. T., Jr., and Quinn, R. K. (1974) Spontaneous generation of acoustic signals during switching of amorphous semiconductors, *Proc. Int. Conf. Amorphous Liquid Semiconduct., Garmisch-Partenkirchen, Germany, September*, (J. Stuke and W. Brenig, eds.). Taylor and Francis, London.

Bianchetti, R., Hamstad, M. A., and Mukherjee, A. K. (1976). Correlation between acoustic emission and microstructure in a dispersion hardened aluminum alloy, *Proc. Int. Conf. Mech. Behavior Mater., 2nd Boston, Massachusetts, August* Amer. Soc. Metall.

Breckenridge, F. R., Tschiegg, C. E., and Greenspan, M. (1975). Acoustic emission: some applications of Lamb's problem, *J. Acoust. Soc. Am.* **57**, 626–631.

Burridge, R., and Knopoff, L. (1964). Body force equivalents for seismic dislocations, *Bull. Seismol. Soc. Am.* **54**, 1875–1888.

Burridge, R., and Willis, J. R. (1969). The self-similar problem of the expanding elliptical crack in an anisotropic solid, *Proc. Cambridge Philos. Soc.* **66**, 443–468.

Cannelli, G., and Cantelli, R. (1979). Temperature-stimulated acoustic emission in the niobium–hydrogen system, *J. Appl. Phys.* **50**, 5666–5675.

Carlos, M. F., and Jon, M. C. (1978). Detection of cracking during rotational soldering of a high reliability and voltage ceramic capacitor, *Proc. IEEE Electron. Components Conf., 28th, Anaheim, California, April* pp. 336–339.

Carpenter, S. H., and Heiple, C. R. (1980). Acoustic emission generated by dislocation mechanisms during deformation of metals, *in* "Fundamentals of Acoustic Emission" (K. Ono, ed.). Materials Dept., UCLA.

Carpenter, S. H., and Higgins, F. P. (1976). The identification of sources of acoustic emission operative during deformation, *Proc. Acoust. Emission Symp., 3rd* pp. 288–304. Japan Industrial Planning Association, Tokyo.

Carpenter, S. H., and Higgins, F. P. (1977). Sources of acoustic emission generated during plastic deformation of 7075 aluminum alloy, *Metall. Trans.* **A8**, 1629–1632.

Ceranoglu, A. N., and Pao, Y. H. (1980a). Propagation of elastic pulses and acoustic emissions in a plate. I. Theory, *J. Appl. Mech.* (to be published).

Ceranoglu, A. N., and Pao, Y. H. (1980b). Propagation of elastic pulses and acoustic emission in a plate. II. Epicentral responses, *J. Appl. Mech.* (to be published).

Ceranoglu, A. N., and Pao, Y. H. (1980c). Propagation of elastic pulses and acoustic emission in a plate. III. General responses, *J. Appl. Mech.* (to be published).

Chou, C. C. (1960). Dynamical response of an elastic half-space to tangential surface loading, *J. Appl. Mech.* **27**, 559–567.

Djordjevic, B. B., and Green, R. E., Jr. (1979). High speed digital capture of acoustic emission

and ultrasonic transients as detected with optical laser beams probes, *Proc. Ultrason. Int. Conf., Graz, Austria, May.*

Drew, J. M., and Seitz, M. G. (1979). Applications of acoustic emission analysis with industrial ceramics, presented at *Int. Conf. Acoust. Emission, Anaheim, California*, September.

Drouillard, T. F. (1979). "Acoustic Emission—A Bibliography with Abstracts." Plenum, New York.

Duke, J. C., and Green, R. F. (1978). Capability of determining fatigue mechanisms in 7075 aluminum by combining monitoring, Proc. ARPA/AFML Rev. Prog. Quant. Nondestr. Eval. Meeting, La Jolla, California, July.

Duke, J. C., and Green, R. E. (1979). Simultaneous monitoring of acoustic emission and ultrasonic attenuation during fatigue of 7075 aluminum, *Int. J. Fatigue* **1**, 125–132.

Egami, T. (1978). Structural relaxation in amorphous $Fe_{40}Ni_{40}P_{14}B_6$ studied by energy dispersive x-ray diffraction, *J. Mater. Sci.* **13**, 2587–2599.

Egle, D. M., and Brown, A. E. (1976). A note on pseudo-acoustic emission sources, *J. Test. Eval.* **4**, 196–199.

Elsley, R. K., and Graham, L. J. (1976). Pattern recognition techniques applied to sorting acoustic emission signals, *Ultrason. Symp. Proc.* IEEE Cat. No. 76 CH1120-5SU.

Eringen, A. C. (1957). Elasto-dynamic problem concerning the spherical cavity, *Q. J. Mech. Appl. Math.* **10**, 257–270.

Evans, A. G. (1975). Residual stress measurement using acoustic emission, *J. Am. Ceram. Soc.* **58**, 239–243.

Feng, C., and Whittier, R. M. (1979). Acoustic emission transducer calibration using transient surface waves and signal analysis, Dunegan/Endevco Tech. Rep. DE 79-1, September.

Fisk, C. and Lord, A. E., Jr. (1980). The residual twist in as-prepared metallic glass alloys, *Lett. Appl. Eng. Sci.* **17**, 993–995.

Fleischmann, P., Rouby, D., Lakestani, F., and Baboux, J. C. (1975). A spectrum analysis of acoustic analysis, *Non-destr. Test. (Guildford, England)* **8**, 241–244.

Fowler, T. J., Jeter, J. R. and Scarpellini, R. S. (1979). Acoustic emission testing of fiber-reinforced plastic equipment, *Proc. Conf. Managing Corrosion with Plastics*, 1979, Vol. 4, pp. 82–90.

Gahr, S. A., and Payne, C. H. (1979). Acoustic emission detection—Part III—Real-time monitoring of the quality of resistance welds, *West. Electr. Eng.* **23**, 21–29.

Garnier, J. E., Repal, A. S., and Bates, J. L. (1976). Application of acoustic emission to evaluation of dielectric breakdown in insulator materials, *Bull. Am. Ceram. Soc.* **55**, 533–534.

Gilman, J. J. (1975). Metallic glasses, *Phys. Today* May, pp. 46–53.

Goldsmith, W. (1960). "Impact." Richard Clay, Bungay, Suffold, England.

Graham, L. J. (1979). Multi-parameter analysis for acoustic emission source identification, *Proc. Int. Instrum. Symp., 25th, ISA, Anaheim, California.*

Graham, L. J., and Alers, G. A. (1972). Frequency spectra of acoustic emissions generated by deforming metals and ceramics, *IEEE Ultrason. Symp. Proc.* 18–21.

Green, R. E., Jr. (1977). Acoustic emission: a critical comparison between theory and experiment, *Proc. Ultrason., Int. Conf., Brighton, England, June* pp. 235–244.

Green, R. E., Jr. (1979). Nondestructive methods for the early detection of fatigue damage in aircraft components, AGARD Lecture Series No. 103, NATO, pp. 6-1–6-31.

Green, R. E., and Duke, J. C., Jr. (1979). Ultrasonic and acoustic emission detection of fatigue damage, in "International Advances in Nondestructive Testing" (W. J. McGonnagle, ed.), Vol. 6. Gordon and Breach, New York.

Green, R. E., Jr., Djordjevic, B. B., Palmer, C. H., and Fick, S. E. (1979). Laser beam

detection of ultrasonics and acoustic emission signals and nondestructive testing of materials, *Proc. Conf. Appl. Lasers Mater. Proc., Washington, D. C., April.*

Hamstad, M. A. (1980). Deformation and failure information from composite materials via acoustic emission, *in* "Fundamentals of Acoustic Emission" (K. Ono, ed.). Materials Dept., UCLA.

Hatano, H. (1975). Quantitative measurements of acoustic emission related to microscopic mechanisms, *J. Acoust. Soc. Am.* **57**, 639–645.

Higgins, F. P., and Carpenter, S. H. (1978). Sources of acoustic emission generated during tensile deformation of pure iron, *Acta Metall.* **26**, 133–139.

Hirai, N., and Eyring, H. (1958). Bulk viscosity of liquids, *J. Appl. Phys.* **29**, 810–816.

Houghton, J. R., Townsend, M. A., and Packman, P. F. (1978). The application of a time-domain deconvolution technique for identification of experimental acoustic emission signals, *Exp. Mech.* **18**, 234–239.

Howells, E. (1974). Acoustic emissions from stressed dielectric liquids, *Proc. ISA Int. Instrum. Symp., 20th, May, Albuquerque, New Mexico.*

Howells, E., and Norton, E. T. (1977). Detection of partial discharges in transformers using acoustic emission techniques, presented at *IEEE Power Eng. Soc. Meeting, Mexico City, July.*

Hsu, N. N., Eitzen, D. G. (1979). Acoustic emission characterization through direct-time deconvolution, *Proc. ARPA/AFML Rev. Quant. NDE.*

Hsu, N. N., and Hardy, S. C. (1978). Experiments in acoustic emission waveform analysis for characterization of AE sources, sensors and structures, *in* "Elastic Waves and Non-Destructive Testing of Materials" (Y. H. Pao, ed.), AMD-Vol. 29, pp. 85–106. ASME, New York.

Hsu, N. N., Simmons, J. A., and Hardy, S. C. (1977). An approach to acoustic emission signal analysis-theory and experiment, *Mater Eval.* **35**, 100–106.

Hunderi, O., and Lönvik, K. Thermosonimetric investigation of the crystallization of metallic glasses, *in* "Rapidly Quenched Metals III," Vol. 1, pp. 357–382. Metals Soc., London.

Hutton, P. H., and Skorpik, J. R. (1979). Application of digital memory acoustic emission system to aircraft flaw monitoring, presented at *Int. Conf. Acoust. Emission, Anaheim, California, September.*

Johnson, L. J. (1974). Green's function for Lamb's Problem, *Geophys. J. R. Astron. Soc.* **37**, 99–131.

Jon, M. C. (1979). Monitoring laser welds using stress wave emission (SWE) techniques, *in* "International Advances in Nondestructive Testing" (W. J. McGonnagle, ed.), Vol. 6. Gordon and Breach, New York.

Jon, M. C., Duncan, H. A., and Vahaviolos, S. J. (1978a). Analysis of stress wave emission (SWE) in resistance welding of tantalum capacitor, *Mater. Eval.* **36**, 40–44.

Jon, M. C., Keskimaki, C. A., and Vahaviolos, S. J. (1978b). Testing resistance spotwelds using stress wave emission (SWE) techniques, *Mater. Eval.* **36**, 41–45.

Jon, M. C., Palazzo, V., and Fekula, R. G. (1979a). Analysis of thermocompression (TC) bond failure modes using a microcomputer controlled pull tester and stress wave emission (SWE) techniques, *Proc. IEEE Electron. Components Conf., 24th, Cherry Hill, New Jersey, May*, pp. 200–205.

Jon, M. C., Ware, G., and Stapleton, J. (1979b). Acoustic emission detection—Part IV—applications in undersea repeater manufacture, *West. Electr. Eng.* **23**, 31–38.

Jon, M. C., Keskimaki, C. A., Grobos, S. T., and Califano, V. E. (1980). Testing low profile (LP) relay break spring welds using stress wave emission (SWE) techniques, *Weld. J.* **59**, 23–27.

Kahn, S., and Miller, D. (1979). Acoustic emission detection—Part II—Detecting ceramic

substrate cracking during thermocompression bonding, *West. Electr. Eng.* **23**, 15–19.

Kass, W. J., Northrup. C. J. M., and Beattie, A. G. (1977). The use of acoustical emission in determining the phase diagrams of hydrides, *Proc. Conf. Environ. Degrad. Eng. Mater., Blacksburg, Virginia, October*, pp. 557–565.

Kline, R. A., Green, R. E., Jr., and Palmer, C. H. (1978). A comparison of optically and piezoelectrically sensed acoustic emission signals, *J. Acoust. Soc. Am.* **64**, 1633–1639.

Knopoff, L. (1958). Surface motions of a thick plate, *J. Appl. Phys.* **29**, 661–670.

Knott, J. F. (1973). "Fundamentals of Fracture Mechanics," p. 58. Butterworths, London.

Kolsky, H. (1973). The stress pulses propagated as a result of the rapid growth of brittle fracture, *Eng. Fract. Mech.* **5**, 513–522.

Kusanagi, H., Kimura, H., and Sasaki, H. (1979). Stress effect on the magnitude of acoustic emission during magnetization of ferromagnetic materials, *J. Appl. Phys.* **50**, 2985–2987.

Leemans, D. V., Baron, J. A., and Kup, O. A. (1979). Measurement of two-phase flow by acoustic emission techniques, presented at *Int. Conf. Acoust. Emission, Anaheim, California*, September.

Lamb, H. (1904). On the propagation of tremors over the surface of an elastic solid, *Philos. Trans. R. Soc.* **A203**, 1–42.

Landau, L. D., and Lifshitz, E. M. (1970). "Theory of Elasticity, Course of Theoretical Physics," Vol. 17, 2nd ed., pp. 30–37. Pergamon, Oxford.

Lord, A. E., Jr. (1975). Acoustic emission, *in* "Physical Acoustics," Vol. 11, pp. 289–353 (W. P. Mason and R. N. Thurston, eds.). Academic Press, New York.

Lord, A. E., Jr., and Koener, R. M. (1980). Acoustic emission in geological materials, *in* "Fundamentals of Acoustic Emission." (K. Ono, ed.). Materials Dept., UCLA.

Luborsky, F., and Walter, J. (1978). Stress relaxation in amorphous alloys, *Mater. Sci. Eng.* **35**, 255–261.

McBride, S. L. (1978). Canadian forces in-flight acoustic emission monitoring program, *Proc. ARPA/AFML Rev. Progr. Quant. NDE* pp. 101–102.

McBride, S. L., and Hutchison, T. S. (1976). Helium gas jet spectral calibration of acoustic emission transducers and systems, *Can. J. Phys.* **54**, 1824–1830.

McBride, S. L., Hutchison, T. S., and Murphy, R. (1974). Acoustic emission and the semiconductor-metal transition in vanadium dioxide thin films, *Scr. Metall.* **8**, 431–434.

Matzkanin, G. A., Beissner, R. E., and Teller, C. M. (1979). The Barkhausen Effect and its Applications to Nondestructive Evaluation, NTIAC-79-2, Southwest Research Institute, San Antonio, Texas.

Mueller, G. (1968). Theoretical seismograms for some types of point-sources in layered media — Part II—Numerical calculation, *Z. Geophys.* **34**, 147–162.

Northrup, C. J. M., Kass, W. J., and Beattie, A. G. (1977). Acoustic emissions during hydride formation, *Proc. Int. Symp. Hydrides Energy Storage*, Geilo, Norway, August.

O'Handley, R. C. (1975). Magnetic core loss and internal stress in metallic glass, *AIP Conf. Proc.* **29**, 206–208.

Ono, K. (ed.) (1980). "Fundamentals of Acoustic Emission" (*Proc. Joint Meeting of Acoust. Soc. Jpn., and Am., Honolulu, Hawaii, December*, 1978). Materials Dept., UCLA.

Ono, K., and Shibata, M. (1980). Magnetomechanical acoustic emission of iron and steels, *Mater. Eval.* **38**, 55.

Ono, K., Huang, G., and Kawamoto, A. (1977). Anisotropic acoustic emission behavior of HSLA steels, *Proc. Conf. Internal Friction Ultrason. Attenuat. Solids*, pp. 829–834. Univ. Tokyo Press, Tokyo, Japan.

Palmer, C. H., and Green, R. E., Jr. (1977). Materials evaluation by optical detection of acoustic emission signals, *Mater. Eval.* **35**, 107–112.

Papa, T., Sette, D., Stagni, L., and Castellano, A. C. (1975). Acoustic emission produced

during heating of tin and zinc single crystals, *J. Test. Eval.* **5**, 48–50.

Pao, Y. H. (1978). Theory of acoustic emission, *in* "Elastic Waves and Non-destructive Testing of Materials" (Y. H. Pao, ed.), AMD-Vol. 29, pp. 107–128. ASME, New York.

Pao, Y. H., and Gajewski, R. (1977). The generalized ray-theory and transient elastic wave in layered media, *in* "Physical Acoustics," Vol. 13, pp. 184–226 (W. P. Mason and R. N. Thurston, eds.). Academic Press, New York.

Pao, Y. H., Gajewski, R., and Ceranoglu, A. N. (1979). Acoustic emission and transient waves in an elastic plate, *J. Acoust. Soc. Am.* **65**, 96–105.

Pekeris, C. L. (1955). The seismic surface pulse, *Proc. Natl. Acad. Sci. U.S.A.* **41**, 469–480.

Pekeris, C. L., and Lifson, H. (1957). Motion of the surface of a uniform elastic half space produced by a buried pulse, *J. Acoust. Soc. Am.* **29**, 1233–1238.

Pardee, W. J. (1977). Acoustic emission and the plate Green's function, *J. Math. Phys.* **18**, 676–686.

Pardee, W. J., and Graham, L. J. (1975). Analysis of acoustic emission frequency spectra, *IEEE Ultrason. Symp. Proc.*, Cat. No. 75 CHO 994-450, 597–600.

Randall, R., and Graham, L. J. (1978). Acoustic emission characterization system, *Proc. ISA Int. Instrum. Symp., 24th, Albuquerque, New Mexico* pp. 439–444.

Sachse, W., and Ceranoglu, A. N. (1979a). Experiments with a well-characterized acoustic emission system, *Proc. Ultrason. Int. Conf., Graz, Austria* pp. 138–145.

Sachse, W., and Ceranoglu, A. N. (1979b). Absolute measurements with piezo-electric transducers, *Ultrason. Symp. Proc.* IEEE Cat. No. 79CH1482 9SU.

Scruby, C. B., and Wadley H. N. G. (1978). A calibrated capacitance transducer for the detection of acoustic emission, *J. Phys. D.* **11**, 1487–1494.

Scruby, C. B., Collingwood, J. C., and Wadley, H. N. G. (1978). A new technique for the measurement of acoustic emission transients and their relationship to crack propagation, *J. Phys. D.* **11**, 2359–2369.

Scruby, C. B., Wadley, H. N. G., and Sinclair, J. E. (1979). The relationship between acoustic emission and fracture mechanisms illustrated by a study of cleavage in mild steel, *Int. Mech. Eng. Conf., London*, pp. 127–136.

Shiraiwa, T., Sakamoto, Y., Yamaguchi, H., Suzuki, T., Fujisawa, K., and Arahori, T. (1978). Acoustic emission characteristics of firebricks, *Proc. Acoust. Emission Symp., 4th, Tokyo, September.*

Simmons, J. A., and Clough, R. B. (1976). Theoretical aspects of acoustic emission spectral measurements, *Proc. World Conf. Nondestr. Testing, 8th, Cannes, France*, 3K, pp. 1–8.

Sinclair, J. E. (1979). Epicentre solutions for point multipole sources in an elastic halfspace, *J. Phys. D* **12**, 1309–1315.

Spanner, J. C. (1978). Acoustic emission—Applications and trends, *Proc. Symp. Elastic Waves Nondestr. Testing Mater.* ASME AMD, San Francisco, California, December.

Spanner, J. C. (1979). Acoustic emission—Some examples of increasing industrial maturity, paper presented at *ASTM Symp. Acoust. Emission Monitoring of Pressurized Syst., Fort Lauderdale, Florida, January.* Hanford Engineering Development Lab. Rep. No. HEDL-SA-1541.

Spanner, J. C., and McElroy, J. W. (eds.) (1974). "*Monitoring Structural Integrity by Acoustic Emission*," ASTM STP 571. American Society for Testing and Materials, Philadelphia, Pennsylvania.

Stagnic, L., and Castellano, A. C. (1977). Experiments on acoustic emission generated during rapid heating of tin single crystals, *J. Mater. Sci* **12**, 1431–1437.

Steinberg, J., and Lord, A. E. Jr. (1980). Acoustic emission generated at the glass transition in silicate glasses, *J. Am. Ceram. Soc.* **63**, 234–235.

Steinberg, J., Tyagi, S., and Lord, A. E., Jr. (1980). Thermally-induced acoustic emission in

glasses, *NDT Intern.* (October), pp. 225–229.

Stokes, G. G. (1851). On the dynamic theory of diffraction, *Trans. Cambridge Philos. Soc.* **9**, 1–62.

Swain, W. E. (1979). Nondestructive testing of aerial man-lift and derrick equipment using acoustic emission instrumentation, Presented at *Int. Conf. Acoust. Emission, Anaheim, California, September.*

Wadley, H. N. G., and Scruby, C. B. (1979a). A study of deformation and fracture processes in a low-alloy steel by acoustic emission transient analysis, *Acta Metall.* **27**, 613–625.

Wadley, H. N. G., and Scruby, C. B. (1979b). Acoustic emission source characterization, paper presented at *Int. Conf. Acoust. Emission, Anaheim, California, September* 10–13.

Wadley, H. N. G., Scruby, C. B., and Speake, J. H. (1979). Acoustic Emission for the Physical Examination of Metals, Rep. No. AERE R-9335, Harwell, January.

Willis, J. R. (1973). Self-similar problems in elastodynamics, *Philos. Trans. R. Soc. London* **274**, 435–491.

Ying, S. P. (1973). Characteristics and mechanisms of acoustic emission from solids under applied stress, *Crit. Rev. Solid State Sci.* **4**, 85–123.

Author Index

A

Abello, T. P., 50, *90*
Adams, P. L., 162, 177, *189*
Aigner, F., 27, *90*
Albers, V. M., *90*, 95
Albersheim, W. J., 107, 180, 182, *188*
Alers, G. A., 297, *356*
Alexander, P., 68, 69, 70, *90*
Allen, C. H., 64, 83, *90*
Altberg, W., 24, *90*
Andrade, E. N., 66, *90*
Antropov, G. A., 83, *92*
Arahori, T., 339, *359*
Arenberg, D. L., 78, *90*
Auld, B. A., 101, 130, 133, 145, *187*

B

Baboux, J. C., 317, *356*
Bär, R., 48, *90*
Bahr, A. J., 145, 146, 149, *187*
Bailey, C. D., 350, *355*
Balamuth, L., 50, 63, 67, 83, *90*
Baram, J., 339, *355*
Barnes, D. K., 66, *92*
Barnoski, M. K., 185, *187*
Baron, J. A., 350, *358*
Barone, A., 47, 48, *90*
Barrick, D. E., 192, 286, 287, *294*
Barton, E. H., 22, *90*
Bates, J. L., 340, *356*
Beattie, A. G., 334, 339, 340, *355*, *357*, *358*
Beissner, R. E., 347, *358*
Bell, D. T., Jr., 114, 117, 162, *188*
Berg, R. D., 82, *93*
Bergmann, L., 40, 48, 49, 50, 53, 55, 56, 57, 58, 59, 60, 78, 88, *90*, *96*
Berlincourt, D. A., 62, *90*, 113, 137, 140, *187*
Berman, A., 85, *90*
Bertoni, H. L., 226, *292*
Beyer, R. T., 16, *90*
Bianchetti, R., 342, *355*
Bienvenue, G. R., 83, *95*
Biquard, P., 29, 34, 47, 50, *90*, *94*
Blaha, F., 72, *90*
Blake, F. G., Jr., 66, *90*
Blake, R. F., 27, *90*
Bleistein, N., 281, *292*
Bode, H. W., 161, *188*
Bojarski, N., 281, *292*
Bolt, R. H., 89, *93*
Bondy, C., 58, *90*
Bouyoucos, J. V., 84, *90*
Boyle, R. W., 35, 38, 39, 42, 59, *90*
Bracewell, R. M., 137, *187*
Breckenridge, F. R., 297, 299, 300, 301, *355*
Breitenbach, E. D., 193, 208, 209, 210, 213, 214, 215, 216, 217, 223, *292*, *294*
Brekhovskikh, L. M., 235, *292*
Brendel, K., 267, 268, *293*
Brill, D., 197, 240, 249, 261, 262, *292*
Brillouin, L., 41, *90*
Bristol, T. W., 125, 152, 184, *187*, *188*
Brockelsby, C. F., 78, *90*
Brown, A. E., 353, *356*
Brown, B., 70, 72, 83, *90*
Burridge, R., 316, 324, *355*
Butterworth, S., 24, *90*

C

Cady, W. G., 17, 24, 30, 32, 33, 35, 36, 37, 42, 49, 61, 66, 88, *91*
Califano, V. E., 351, 352, *357*
Campbell, J. J., 139, 145, *188*
Cannelli, G., 339, *355*
Cantelli, R., 339, *355*

Carlin, B., 54, 78, 83, 88, *91*
Carlos, M. F., 353, *355*
Carpenter, S. H., 296, 340, 341, 342, *355*, *357*
Carter, S. W., 72, 74, 76, 77, *92*
Carwile, P. B., 69, *91*
Castellano, A. C., 339, *358*, *359*
Ceranoglu, A. N., 301, 302, 312, 313, 314, 315, 354, *355*, *359*
Cermak, I. A., 113, *188*
Chaffee, E. L., 23, *91*
Chipman, R. A., 23, *91*
Chou, C. C., 299, *355*
Clay, C. S., 228, *292*
Clough, R. B., 301, *359*
Cohen, E., 102, *189*
Collin, R. E., 120, *188*
Collingwood, J. C., 316, 317, 320, 321, 322, *359*
Collins, J. H., 102, 103, 132, 133, 135, 136, 138, 142, 157, *189*
Cooper, K. W., 66, *92*
Coquin, G. A., 101, *188*
Courtois, L., 287, 289, *293*
Cowart, G., 290, *294*
Crawford, A. E., 57, 64, 70, 71, 73, 78, 79, 86, 88, *91*
Creasey, C. H., 9, 10, *91*
Cremer, L., 207, *292*
Curie, J.-P., 16, 17, *91*
Curie, M., 17, *91*
Curie, P., 16, 17, *91*
Curran, D. R., 62, *90*, 113, 137, 140, *187*

D

Darlington, S., 107, 180, 182, *188*
Davis, H., 83, *91*
Dawson, W. J., 76, *91*
Debye, P., 47, *91*
Dehlinger, P., 67, *91*
Derem, A., 193, *292*
de Santo, J., 281, *292*
Desch, C. H., 76, *91*
Despretz, C., 5, *91*
Devin, Ch., 228, *292*
Dickey, J. W., 207, 213, *292*, *293*
Dieckmann, M., 23, *91*
Di Franco, J. V., 105, *188*
Djordjevic, B. B., 333, 334, *355*, *356*
Doolittle, D., 208, 227, *292*
Dormann, J. L., 287, 289, *293*
Dragonette, L. R., 197, 198, 204, 206, 223, 226, 227, *292*, *293*, *294*
Dragún, L. R., 193, *292*
Drew, J. M., 350, *356*
Drouillard, T. F., 295, *356*
Drysdale, C. V., 29, *91*
Duddell, W., 22, *91*
Duff, A. W., 15, *91*
Duke, J. C., 348, *356*
Duncan, H. A., 351, *357*
Dvorak, V., 11, *91*
Dye, D. W., 37, *91*

E

Eckart, C., 66, *91*
Eckart, J., 264, *292*
Edelmann, M. Th., 20, 21, *91*
Edler, I., 82, *91*
Egami, T., 336, *356*
Egle, D. M., 353, *356*
Einspruch, N. G., 273, *292*
Einstein, A., 41, *91*
Eisenbud, L., 231, *294*
Eitzen, D. G., 306, 308, 312, *357*
Elsley, R. K., 334, *356*
Elwell, C. F., 23, *91*
Engan, H., 149, *188*
Ensminger, D., 68, 83, *91*
Enz, U., 63, *91*
Erdman, D. C., 77, *91*
Eringen, A. C., 301, *356*
Erwin, W. S., 77, *91*
Esche, R., 66, *91*
Evans, A. G., 347, *356*
Eve, A. S., 9, 10, *91*
Ewing, W. M., 279, 280, *292*
Eyring, H., 336, *357*

F

Fagen, M. D., 23, 36, *91*
Fano, R. M., 161, *188*
Faraday, M., 11, *91*
Faran, J. J., 201, *292*
Farhan, A. R., 193, 218, 228, 230, *292*, *294*
Farnell, G. W., 113, *188*
Fay, H. J. W., 26, 27, 28, 35, 51, *91*

Feit, D., 192, 195, 199, *293*
Fekula, R. G., 351, *357*
Feng, C., 301, *356*
Fenstermacher, T. E., 162, *189*
Fessenden, H. M., 27, *92*
Fiorito, R., 193, 233, 234, 235, 237, 238, 239, *292*, *293*
Firestone, F. A., 75, 76, *92*
Fisk, C., 347, *356*
Flax, L., 193, 194, 202, 204, 206, 221, 227, 268, 273, 274, 275, 276, 277, 278, 279, 280, 291, *292*, *293*, *294*
Fleischmann, P., 317, *356*
Flynn, H. G., 58, 66, *92*
Foley, A. L., 22, *92*
Fowler, T. J., 296, *356*
Fox, F. E., 79, *92*
Frederick, J., 67, *92*
Frings, H., 83, *90*
Frisk, G. V., 207, 213, *292*, *293*
Fujisawa, K., 339, *359*

G

Gahr, S. A., 351, *356*
Gaines, N., 46, 55, *92*, *97*
Gajewski, R., 301, *359*
Galambos, R., 4, 27, 40, *92*
Galton, F., 5, 6, *92*
Garnier, J. E., 340, *356*
Gastine, M., 287, 289, *293*
Gaunaurd, G. C., 193, 218, 221, 222, 223, 224, 228, 231, 240, 242, 245, 246, 247, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 265, 266, 267, 268, 281, 282, 283, 284, 285, 290, 291, *292*, *293*, *294*
Gerard, H. M., 102, 103, 124, 125, 132, 133, 134, 135, 136, 137, 138, 139, 142, 157, 180, 185, *188*, *189*
George, J., 193, 202, 210, 212, 218, 219, 228, 229, 230, 285, 286, *292*, *293*, *294*
Gericke, O. R., 54, *92*
Giacomini, A., 54, *92*
Giacomini, A. B., 54, *92*
Gibson, R. W., 78, *90*
Giebe, E., 37, *92*
Gillispie, C. C., 8, 9, 10, 16, 17, 20, 22, 23, 24, 27, 29, 35, 43, *92*
Gilman, J. J., 336, *356*
Gjevik, B., 281, *293*
Goldsmith, W., 312, *356*
Goodman, J. E., 70, 72, *90*
Gorbunov, O. N., 83, *92*
Gordon, D., 82, 83, *90*, *92*
Gorskhkov, S. I., 83, *92*
Graff, K. F., 67, 69, *92*
Graham, L. J., 297, 334, *356*, *359*
Green, R. E., Jr., 332, 333, 334, 348, 349, *355*, *356*, *358*
Greenspan, M., 297, 299, 301, *355*
Griffin, D. R., 4, 40, 52, 53, 86, 87 *92*, *95*
Grobos, S. T., 351, 352, *357*
Grossmann, E., 50, *92*
Grubinskas, R. C., 54, *92*
Guthrie, A. N., 85, *90*

H

Hamstad, M. A., 342, *355*, *357*
Hardy, S. C., 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, *357*
Harrington, J. B., 180, *188*
Harris, J. F., 178, *188*
Hart, S. D., 233, *294*
Hartmann, C. S., 102, 114, 117, 120, 125, 149, 162, *188*
Hartmann, J., 39, *92*
Harvey, E. B., 55, *92*
Harvey, E. N., 55, 66, *92*
Hastings, C. H., 72, 74, 76, 77, *92*
Hatano, H., 340, *357*
Haug, A. J., 193, 268, 273, *293*
Hayes, H. C., 29, *92*
Heckl, M., 207, *292*
Heiple, C. R., 296, 340, 342, *355*
Helmoltz, H. L. F., 6, *92*
Herrick, J. F., 55, 79, 80 82, *93*
Hertz, C. H., 82, *91*
Herzfeld, K. F., 15, 24, 41, 50, 84, *93*
Hewlitt, A., 39, *93*
Hezlet, A., 27, 52, *93*
Hickling, R., 203, *293*
Hiedemann, E., 57, 60, 72, *93*
Higgins, F. P., 341, *355*, *357*
Hirai, N., 336, *357*
Höyen, J., 281, *293*
Hofstetter, E. M., 179, *188*
Hoiem, B. E., 102, *188*
Holland, M. G., 107, 110, 113, *189*

Hollander, A., 66, *93*
Holtzmann, M., 39, *93*
Holtzmark, J., 67, *93*
Hopwood, F. L., 55, 56, *93*
Horton, J. W., 87, *93*
Houghton, J. R., 317, *357*
Hovorka, F., 73, *97*
Howells, E., 340, *357*
Howeth, L. S., 26, 32, 34, 51, 87, *93*
Hsu, N. N., 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, *357*
Huang, G., 342, *358*
Hubbard, J. C., 2, 50, 59, *93*
Hueter, T. F., 89, *93*
Hunderi, O., 339, *357*
Hunt, F. V., 16, 17, 18, 23, 24, 26, 28, 34, 35, 36, 37, 46, 52, 84, 88, *93*, *96*
Hutchison, T. S., 317, 339, *358*
Hutton, P. H., 350, *357*

J

Jaffe, H., 62, 113, 137, 140, *90*, *187*
Janovsky, W., 64, *93*
Jardetzky, W. S., 279, 280, *292*
J. B., 34, *93*
Jeans, J. H., 24, *93*
Jeter, J. R., 296, *356*
Johnson, C. H., 55, *93*, *96*
Johnson, G., 192, 273, *293*
Johnson, I., 67, *93*
Johnson, L. J., 301, *357*
Johnson, R. T., Jr., 340, *355*
Jon, M. C., 351, 352, 353, *355*, *357*
Jones, R. C., 64, *93*
Jones, W. R., 124, 125, 132, 134, 135, 137, 139, 142, 145, 180, 184, 185, *188*, *189*
Jones, W. S., 102, 125, *188*
Joshi, S. G., 101, *188*
Joule, J. P., 16, *93*
Judd, G. W., 124, 182, 183, 184, *188*
Junger, M. C., 192, 195, 199, *293*

K

Kahn, S., 351, *357*
Kallmann, H. E., 104, *188*
Kass, W. J., 339, *357*, *358*
Kawamoto, A., 342, *358*
Kazantsev, V. F., 67, *96*
Kearns, W. J., 162, *189*
Keller, A. C., 84, *93*
Kelly, E., 79, 80, 82, 84, *93*
Kerlin, R. L., 83, *95*
Keskimaki, C. A., 351, 352, *357*
Keskin, M., 290, *294*
Kimura, H., 342, 343, 344, *358*
Kingery, R. A., 82, *93*
Kino, G. S., 101, 145, *187*
Kirchhoff, G., 15, *93*
Klauder, J. R., 107, 180, 182, *188*
Klein, E., 27, 29, 32, 33, 34, 47, 51, 52, *93*
Kline, R. A., 334, *358*
Knapp, R. T., 66, *93*
Kneser, H. O., 50, *93*
Knopoff, L., 301, 304, 324, *355*, *358*
Knott, J. F., 328, *358*
Kock, W. E., 79, *93*
Kodama, T., 179, *188*
Koener, R. M., 296, *358*
Koenig, R., 8, *94*
Kolsky, H., 297, *358*
Kraft, D. W., 192, 268, 273, *293*
Krasil'nikov, V. A., 25, *94*
Krautkramer, H., 76, 77, *94*
Krautkramer, J., 76, 77, *94*
Krichbaum, C. K., 192, 286, 287, *294*
Krimholtz, R., 102, 145, 146, *188*
Kruse, F., 54, *94*
Krusen, F. H., 55, 79, 80, 82, *93*
Kundt, A, 10, *94*
Kup, O. A., 350, *358*
Kusanagi, H., 342, 343, 344, *358*

L

Laker, K. R., 102, *189*
Lakestani, F., 317, *356*
Lakin, K. M., 132, *188*
Lamb, H., 298, 299, 301, *358*
Landau, L. D., 312, *358*
Lane, A. M., 231, *293*
Lamson, H. W., 35, *94*
Langenecker, B., 72, *90*
Lasky, M., 26, 29, 34, 52, 84, *94*
Lebedev, P., 25, *94*
LeConte, J., 6, 9, *94*
Lee, R. E., 145, 146, 149 *187*
Leedom, D. A., 146, *188*

Leemans, D. V., 350, *358*
Lewis, S. T., 268, *293*
Li, R. C. M., 125, *188*
Lifshitz, E. M., 312, *258*
Lifson, H., 299, 300, 301, 316, 317, *359*
Likhodaeva, E. A., 218, *293*
Lindsay, R. B., 13, 15, 16, 18, 20, 47, 50, *94*
Litsios, J., 64, *94*
Lönvik, K., 339, *357*
Loomis, A. L., 44, 50, 55, *92*, *93*, *96*, *97*
Lord, A. E., Jr., 78, *94*, 295, 296, 335, 336, 337, 339, 342, 347, 348, 349, *356*, *358*, *359*
Lozinskii, M. G., 63, *94*
Luborsky, F., 336, *358*
Lucas, R., 47, *94*
Lynnworth, L. C., 78, *94*

M

McBride, R. J., 192, 273, *293*
McBride, S. L., 317, 339, 350, *358*
McClellan, J. H., 179, *189*
McElroy, J. W., 296, *359*
McElroy, W. D., 66, *92*
McMaster, R. C., 77, *94*
Madigosky, W., 233, 235, 238, 239, *292*, *293*
Mahoux, G., 56, *94*
Makarov, L. O., 67, *96*
Marinesco, N., 60, *94*
Marshall, S. W., 218, *293*
Masenten, W. K., 106, *188*
Mason, W. P., 61, 62, 63, 88, *94*, 137, *189*
Matthaei, G. L., 102, 146, *188*
Mattiat, O., 71, 73, *95*
Matzkanian, G. A., 347, *358*
May, J. E., Jr., 78, *95*
Mayer, W. G., 234, 239, *294*
Medwin, H., 228, *292*, *293*
Melngailis, J., 125, *188*
Meissner, A., 37, *95*
Meyer, E., 48, *90*, 267, 268, *293*
Meyer, F. R., 69, 72, *95*
Mezzorani, G., 193, 218, 228, 230, *294*
Michael, P. L., 83, *95*
Miklowitz, J., 268, *294*
Miller, D., 351, *357*
Miller, D. C., 4, 5, 6, 8, 9, 14, 20, *95*
Miller, H. B., 63, *95*
Milsom, R. F., 101, *189*
Minnaert, M., 228, *294*
Morse, F., 183, *188*
Moser, P. J., 193, 290, 291, *293*, *294*
Mow, C. C., 268, *294*
Mühlhäuser, O., 53, *95*
Mueller, G., 301, *358*
Mukherjee, A. K., 342, *355*
Murphy, J. D., 193, 208, 210, 212, 218, 219, 223, 286, 290, 291, *294*
Murphy, R., 339, *358*

N

Nagl, A., 193, 210, 218, 219, 228, 230, 286, 290, 291, *294*
Nath, R. S. N., 48, *95*
Neklepajew, N., 25, *95*
Neppiras, E. A., 46, 62, 63, 66, 67, *95*
Neubauer, W. G., 194, 197, 221, 227, 268, *292*, *293*
Neumann, E. P., 70, 73, 74, *95*
Newton, R. G., 193, 212, 220, 241, *294*
Nigul, U. K., 218, *294*
Nilsen, A., 281, *293*
Nishi, R. Y., 228, *294*
Noltingk, B. E., 66, *95*
Northrup, C. J. M., 339, *357*, *358*
Norton, E. T., 340, *357*
Norton, J. L., 70, 73, 74, *95*
Nudd, G. R., 185, *189*
Nyborg, W. L., 66, 67, *95*
Nyffeler, F., 102, *189*

O

O'Handley, R. C., 347, *358*
Olson, A. R., 55, *96*
Olson, J. G., 218, *293*
Ono, K., 296, 342, 343, 345, 346, *358*
Openheim, A. V., 179, *188*
Osborne, M. F. M., 233, *294*
Otto, O. W., 180, *188*

P

Packman, P. F., 317, *357*
Palaiologos, K., 39, *95*
Palazzo, V., 351, *357*

Palfreeman, J. S., 78, *90*
Palmer, C. H., 332, 334, *356*, *358*
Pao, Y. H., 268, *294*, 301, 302, 324, 354, *355*, *359*
Papa, T., 339, *358*
Pardee, W. J., 301, *359*
Parks, T. W., 179, *189*
Payne, C. H., 351, *356*
Pearson, K., 7, *95*
Pease, D. C., 66, *92*
Peck, J. C., 268, *294*
Pedinoff, M. E., 124. *188*
Pedler, W. R., 142, 143, 144, 145, 146, 148, 149, 151, 152, 153, *189*
Pekeris, C. L., 298, 299, 300, 301, 316, 317, *359*
Pielemeier, W. H., 38, 50, *95*, *96*
Pierce, G. W., 38, 46, 53, 87, *95*
Pitts, L. E., 234, 239 *294*
Plona, T. J., 234, 239, *294*
Pohlman, R., 49, 55, 64, *93*, *95*
Posakony, G. J., 77, *95*
Press, F., 279, 280, *292*
Preyer, W., 6, *95*
Price, A. C., 107, 180, 182, *188*
Prout, J. H., 83, *95*
P. U. A., 85, *95*
Pye, D., 40, 87, *96*

Q

Quimby, S. L., 38, *95*
Quinn, R. K., 340, *355*

R

Raju, P. K., 202, *293*
Raman, C. V., *95*
Randall, R., 334, *359*
Rassweiler, G. M., 77, *91*
Rayleigh, Lord, 6, 9, 11, 15, 18, 19, 20, 21, *95*, *96*, 102, *189*
Redwood, M., 101, *189*
Reeder, T. M., 102, 103, 132, 133, 135, 136, 138, 142, 157, 162, 177, *189*
Reid, J. A., 50, *96*
Reilly, N. H. C., 101, *189*
Repal, A. S., 340, *356*
Rice, F. O., 50, *93*
Rich, D. L., 38, *96*
Richards, W. T., 50, 55, 56, 58, 59, *96*
Richardson, E. G., 37, 55, 57, 58, 59, 70, 71, 79, 86, 88, *96*
Roberts, G. A., 162, *189*
Roberts, S., 62, *96*
Robinson, J., 8, *96*
Roman, P., 192, 198, *294*
Rosen, M., 339, *355*
Rosenfeld, R. C., 114, 117, 162, *188*
Rouby, D., 317, *356*
Rozenberg, L. D., 63, 67, 72, *94*, *96*
Rubin, W. L., 105, 188
Ruck, G. T., 192, 286, 287, *294*
Rudnick, I., 64, 83, *90*
Rytov, S., 60, *96*

S

Sachse, W., 301, 312, 313, 314, 315, 354, *359*
Sáenz, A. W., 281, *292*
Sage, K. A., 193, 218, 229, 230, *294*
St. Clair, H. W., 63, 70, *96*, *97*
Sakamoto, Y., 339, *359*
Sales, G., 40, 87, *96*
Sasaki, H., 342, 343, 344, *358*
Saunders, F. A., 23, 36, 46, 52, *96*
Savart, F., 5, *96*
Scarpellini, R. S., 296, *356*
Schaefer, Cl., 39, *96*
Schaefer, K. L., *96*
Scharnhorst, K. P., 265, 266, 268, *293*
Scheibe, A., 37, *92*
Schillinger, E. H., 82, *93*
Schlichting, H., 66, *96*
Schmidt, F. O., 55, *96*
Schoch, A., 233, *294*
Schultze, F. A., 20, *96*
Schuster, A., 19, *96*
Scruby, C. B., 296, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 332, 353, *359*, *360*
Seabrook, W., 42, 43, 44, *96*
Sears, F. W., 47, *91*
Secrest, B. G., 102, 149, *188*
Seitz, M. G., 350, *356*
Sette, D., 339, *358*
Shannon, C. E., 105, *189*
Shaw, H. J., 102, 103, 132, 133, 135, 136, 138, 142, 157, *189*

Shenderov, E. L., 218, *293*
Shibata, M., 343, 345, 346, *358*
Shibayama, K., 102, *189*
Shiraiwa, T., 339, *359*
Shreve, W. R., 162, 177, *189*
Siegel, J., 179, *188*
Sikkeland, T., 67, *93*
Silva, H. J., 162, *189*
Silvester, P., 113, *188*
Simmons, J. A., 301, 302, 303, 304, 305, 306, 307, *357*, *359*
Sinclair, J. E., 322, 324, 325, 326, 327, 328, 329, 353, *359*
Skavlem, S., 67, *93*
Skorpik, J. R., 350, *357*
Skraiber, D. S., 54, *96*
Skudrzyk, E., 88, *96*
Slobodnik, Jr., A. J., 101, 123, 124, 162, *189*
Smith, H. I., 181, *189*
Smith, W. R., 102, 103, 107, 108, 110, 113, 124, 125, 126, 132, 133, 134, 135, 136, 137, 138, 139, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 156, 157, 163, 164, 166, 168, 169, 170, 172, 174, 175, 179, 180 183, 185, *188*, *189*
Snow, P. B., 125, *188*
Söllner, K., 58, *90*
Sokolov, S., 53, 54, *96*
Solie, L. P., 181, *189*
Solomon, S. G., 193, 268, 269, 270, 271, 272, 273, *293*, *294*
Souder, W. H., 22, *92*
Spanner, J. C., 296, *359*
Speake, J. H., 353, *360*
Sperry, W. R., 162, 177, *189*
Sproule, D. O., 76, *91*
S. P. T., 8, *96*
Stagni, L., 339, *358*
Stagnic, L., *359*
Stapleton, J., 353, *357*
Stegeman, G. I., 49, *97*
Steinberg, J., 335, 336, 337, 339, 347, 348, 349, *359*
Stephens, R. W. B., 84, *97*
Stokes, C. A., 71, *97*
Stokes, G. G., 14, *97*, 301, *360*
Strasberg, M., 228, *294*
Strutt, R. J., 20, *97*
Stuart, W. D., 192, 286, 287, *294*
Sturdivant, T. D., 125, *188*
Su, J., 193, 294
Subrahmanyan, J. V., 290, *294*
Suzuki, T., 339, *359*
Swain, W. E., 350, *360*
Szabo, T. L., 101, 102, 123, 124, *189*

T

Tamm, K., 267 268, *293*
Tamir, T., 226, *292*
Tancrell, R. H., 107, 110, 113, 178, *189*
Tanglis, E., 290, *293*, *294*
Taton, R., 23, *97*
Tawil, E. P., 48, *97*
Teller, C. M., 347, *358*
Templin, L., 183, *188*
Thomas, R. G., 231, *293*
Tiersten, H. F., 101, *188*
Toepler, A., 13, 21, *97*
Toepler, M., 21, *97*
Townsend, M. A., 317, *357*
Truell, R., 192, 273, 274, *292*, *294*
Truesdell, C., 50, 84, *97*
Tschiegg, C. E., 297, 299, 301, *355*
Tyagi, S., 335, 336, 337, 339, 347, 348, 349, *359*
Tyndall, J., 5, 6, 8, 11, 13, *97*

U

Überall, H., 193, 197, 202, 204, 206, 207, 208, 210, 212, 213, 218, 219, 221, 222, 223, 227, 228, 229, 230, 231, 233, 234, 235, 237, 238, 239, 240, 242, 245, 246, 247, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 265, 266, 267, 268, 269, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 289, 290, 291, *292*, *293*, *294*
Uginčius, P., 208, 227, *292*, *294*
Uhlemeyer, B., 55, *96*

V

Vahaviolos, S. J., 351, *357*
Van der Burgt, C. M., 63, *97*
Van Dyke, K. S., 36, *97*
Varadan, V., 193, *294*

Varadan, V., 193, *294*
Vasile, C. F., 178, 179, *189*
Vigoureux, P., 37, 60, 88, *97*
Vincent, J. H., 46, *97*
Vivian, J. E., 71, *97*
Vogt, R. H., 197, 227, *292*
Vollers, H., 102, *188*
Vonnegut, B., 64, *97*

W

Wadley, H. N. G., 296, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 332, 353, *359*, *360*
Wagers, R. S., 101, 130, 132, *189*
Waldner, M., 185, *189*
Walter, J., 336, *358*
Ware, G., 353, *357*
Waterman, P. C., 193, *294*
Webster, A. G., 40, *97*
Webster, E., 66, *97*
Weinberg, L., 108, *189*
Weissler, A., 67, 73, *97*
Wenk, S., *94*
Wente, E. C., 39, *97*
Westervelt, P. J., 66, *97*
Whewell, W., 4, *97*
Whitcroft, T. H., 27, 35, *97*
White, D. N., 53, 79, 80, 81, 82, *97*
White, R. M., 101, *188*, 268, *294*
Whiteley, A. H., 66, *92*
Whittier, R. M., 301, *356*
Wick, R. F., 62, *94*
Wigner, E. P., 231, *294*
Willard, G. W., 65, 66, *97*
Williams, O. B., 55, *97*
Williamson, R. C., 181, *189*
Willis, J. R., 303, 316, *355*, *360*
Willrich, H. O., 69, *97*
Winder, A. A., 86, *97*
Witterholt, E. J., 273, *292*
Wallaston, W. H., 5, *97*
Woloshyn, I., 24, 49, 54, *97*
Wong, S. K., 113, *188*
Wood, A. B., 6, 8, 29, 43, 88, *97*
Wood, R. W., 3, 5, 12, 24, 32, 42, 43, 44, 48, 55, 60, *97*
Workman, J. W., 268, *294*

Y

Yakhimovich, D. F., 67, *96*
Yamaguchi, H., 339, *356*
Yamanouchi, K., 102, *189*
Yao, P. S., 180, *188*
Yeager, E., 73, *97*
Ying, C. F., 273, 274, *294*
Ying, S. P. 296, *360*

Z

Zachary, W., 281, *292*
Zimmerman, R. L., 185, *189*
Zucker, F. J., 120, *188*

Subject Index

A

Aberdeen Proving Grounds, 43–44
Acoustic Barkhausen effect, 342
Acoustic emission
  data digitation for, 334–335
  dislocation effects and, 340–342
  in-flight monitoring with, 350
  magnetomechanical, 342–348
  in martensitic transformation, 339
  for metallic and silicate glasses and tapes, 338–339
  monitoring fatigue damage with, 348–350
  as nondestructive testing technique, 353
  optical detection of, 333–334
  plastic strain rate and, 341
  practical applications of, 350–353
  signal pattern recognition in, 334
  system analysis for, 304, 334–335
  update on, 295–354
  during various transformations, 335–340
  in weld strength monitoring, 352–353
Acoustic emission crack detection system, 351
Acoustic emission source
  determination of characteristics of, 297–333
  for electrolytic iron, 329
  for mild steel, 329
Acoustic horn, 4
Acoustic impedance, 40–41
Acoustic interferometer, 38, 45, 49–50
Acousto-optic effects, 48–50
Acoustic oscilloscope, 24
Acoustic interferometer, 49–50
Acoustic propagation loss, 131–132
Acoustic radiation pressure, 24
Acoustic reflection coefficient, 163–164
Acoustics
  high-frequency, *see* High-frequency acoustics
  history of, 2–20
  ultrasonics and, 2
Acoustic streaming, 66–67
Acoustic theories, development of, 50
Acoustic waves
  absorption of, 24–25
  electric source term for, 142–145
  inaudible, 2
  surface, *see* Surface acoustic wave filters; Surface acoustic waves
  visualization of, 12
Acoustic-wave scattering
  from elastic targets, 193–240
  solid cylinders and spheres in, 194–207
Admittance matrix, in three-port network, 134
AE, *see* Acoustic emission
"Afternoon effect," 34
Air bubble, sonar cross section and, 229
Air-jet generator, 39
  improvements in, 64
Alloys Steel Research Committee, 76
Aluminum dust, in sound wave detection, 49, 63
American Communication Company, 37
American Company of Aeroprojects, 69
American Lighthouse Board, 26
Ammonium dihydrogen phosphate, 61–62
Animal audition, ultrasonics and, 7–8
Antisubmarine warfare, 52
Apodization, wave-front distortion and, 124
Apodized transducers, dummy electrodes and, 124–125
Arc oscillations, in telephone transmission, 22

Array factor
for crossed-field model, 138
delta-function model and, 113–114
diffraction or "beam steering" and, 123–124
in interdigital transducers, 111
in transducer models, 100–101, 111
for typical applications, 177–187
Array-factor analysis, crossed-field and generalized circuit models in relation to, 154–160
Array-factor design, examples of, 117–120
Array-factor response, transducer geometry design and, 115–120
Array-impulse response, 112
ASDIC ("Antisubmarine Detection Inventions Committee"), 33–34, 37
Attenuation mechanisms, 15
Audibility limits, 5–6, 20–21
high-frequency generators and, 4–8

B

Backscattered waves, amplitudes of, 281
Bandwidth parameter, 171
Barkhausen effect
acoustical, 342
magnetic, 347–348
Bats
audible clicks from, 52–53
echolocation in, 86–87
as "first ultrasonic echo-ranging system," 86–87
"sixth sense" of, 4, 27
ultrasound and, 52
Bell Telephone Laboratories, 62, 77
Biological processes, ultrasound applications in, 72–74
BIOS Reports, 69–70
Bird calls, ultrasonic, 6
Brain, echo technique in study of, 81–82
"Brick wall" filters, 178–179
Brush Development Company, 70–71
Bubbles, *see* Air bubble; Cavitation bubbles; Gas bubbles
Buried pulse, 299–300

C

Capacitance transducers, *see also* Transducer
Green's function and, 315
schematic drawing of, 318
"Yobell" specimen testing and, 317
Cavitation
classical studies in, 66
ultrasonic, 20, 38–39, 58
Cavitation bubbles, collapse of, 39, 66
Cavities, elastic-wave scattering from, 240–281, *see also* Cylindrical fluid-filled cavities; Spherical fluid-filled cavities
Ceramic stability, acoustic-emission monitoring and, 350–351
Charge density distribution, for transducer electrodes, 144
Chemical actions, ultrasonic effects on, 58
Chemical–biological processes, ultrasonics in, 72–74
Circuit factor
for simple matching circuits, 164–166
transducer and, 155
for typical matching circuits, 165
Circuit-factor compensation, 160
Coagulating action, of ultrasonics, 57
Coated shells, resonance scattering and, 221–223
Columbia University, 38
Compressional waves, 241–247
Condenser, spark discharge of, 24
Continuous-wave generation, 22–23
Convolvers, unapodized dispersive transducers and, 184–187
Cornell University, 312
Crack detection, AE system for, 351, *see also* Microcracks
Cracking process
dislocation loop in, 324–325
surface displacement for typical emission in, 326
Crack volume, growing microcrack lifetime and, 327
Cramer's rule, 243
Cranes, acoustic emission testing of, 350
Creeping waves, 206, 246
Crossed-field Mason circuit, 137
Crossed-field model
array-factor analyses and, 154–160

capabilities of, 141–142
electrode mass/electrical loading and, 140–141
element factor for, 139–140
simple array-factor concept and, 155–157
three-port circuit for, 138–139
Crystal detectors, 32, *see also* Piezoelectric effect, 32
Crystallography, Voigt investigations in, 24
Crystal oscillator, 36
Cylinders and spheres
resonance formulation for, 197–202
scattering theory and, 194–207
Cylindrical fluid-filled cavities, 268–273
dispersion curves for, 272
resonance theory for scattering from, 270–271
Cylindrical shells, resonance theory for, 208

D

"Death ray," 83
Debye–Sears effect, 47, 54
DeForest triode, 23
Delay lines, 77–78
biphase coded tapped, 119–120
with interdigital transducers, 102–104
tapped, 183–184
transversal filter and, 104
Delta-function model, interdigital transducer and, 113–114
Detrex Corporation, 68
Dielectric sphere, dispersion curves for surface waves of, 291
Diffraction, ultrasound "prediction" of, 47–48
Diffraction effect, acousto-optic effects and, 48–49
Diffraction theory, 49
Dipotassium tartrate, 61–62
Direct-connection transducer, 165–166
Dislocation effects, acoustic emission generation due to, 340–342
Dislocation loop model
in microcrack processes, 324
for source event, 325
Dispersion curves, for Rayleigh wave, 213–217
Dispersive filters, 180–183, *see also* Surface acoustic wave filters
design procedure for, 181–182
example of, 182–183
Dispersive transducers, 181–182
unapodized, 184–187
Doppler radar, delay lines for, 77
Double electrode, stopband problem and, 125–126
Dove siren, 8
Ductile fracture, surface displacement transient measurement during, 321–323
Dummy electrodes, apodized transducers and, 124–125
Dust figures. in sound velocity measurement, 10–11, 37
Dynamical Theory of Gases, The (Jeans), 24

E

Echo boxes, in radar calibration, 78
Echocardiography, 81
Echolocation, in bats, 86–87
Echo ranges, diurnal variation in, 34
Echoscope, as diagnostic instrument, 80
Edelmann whistle, 20–21
Elasticity and Anelasticity in Metals (Zeners), 84
Elastic spheres, scattering from, 202–203
Elastic targets, acoustic-wave scattering from, 193–240
Elastic-wave scattering, from cavities and inclusions, 240–281
Electrical waves, generation of in early 20th century, 22–24
Electroacoustical devices, early, 35–36
Electroacoustics, early 20th-century developments in, 16–18, 22–26
Electroacoustics (Hunt), 88
Electroacoustic transfer function T_{13}, 157–158
Electroacoustic transformer ratio, 139
Electrode "mass/electrical loading," 125–126
Electrodes, single and double, 125–126
Electrolytic iron, acoustic emission source for, 329
Electromagnetic feedthrough, 130–131

Electromagnetic scattering
resonances in, 285–292
theory of, 285–289
transverse electric and magnetic modes in, 287
Electromagnetic transducer, 63–64
Electronics industry, acoustic emission testing in, 350
Element factor
in crossed-field model, 137–140
for double electrodes, 126
for generalized circuit model transducer, 145
Empty-cavity scattering amplitude, 275–277
Emulsifying-dispersing action, of ultrasound, 56–57
Ethylene diamine tartrate, 61

F

Fast gas reactor, sound dispersion in measurement of, 41
Fessenden oscillator, 27, 35
Filter insertion loss, in SAW filters, 161
Filters
brick-wall, 178
nondispersive bandpass, 177–180
SAW, *see* Surface acoustic wave filters
Finite-amplitude waves, 20
Flame, sensitive, 9–13
Fluid-filled cavities, cylindrical, 268–273
Fluids, gas bubbles in, 228
Free-plate dispersion curves, 234–235

G

Galton whistle, 6–7, 20, 39, 61
Gas bubbles
circumferential wave in, 230
in fluids, 228–232
Gas current generator, 39
Gases, molecular theory of, 19
General Electric Company, 22, 33, 44, 67–68
Generalized circuit model
element factor for, 145
three-port circuit for, 145–146
German GHG system, 52
Giant monopole resonance, 262–267
Glasses, acoustic emission from, 338–339, 346–347
Goodrich Company, 51
Green's function
in acoustic emission, 301–307, 312, 314
calculated vs. experimentally measured, 315
dynamic, 313
experimental determination of, 315, 354
inverse, 309
step-function, 310
Ground loops, in SAW filter, 131

H

Hamming weighting filter, 183, 187
Hartmann generator, 39–40, 61, 63
improvements in, 64
Harvard University, 23
Hearing, upper limit of, 20–21
Heat conduction, sound and, 15
Helmholtz resonators, 9
High-frequency acoustics
early 20th-century work in, 20–28
in 19th century, 3–20
pre-19th-century developments in, 3–4
Rayleigh's contributions to, 18–20
High-frequency acoustic waves, generation of, 23
High-frequency alternator, 23
High-frequency generators
audibility limits and, 4–8
in early 20th century, 20–22
High-power ultrasonics, 64–74
Hydrodynamic cavitation, 38–39, 58, 66

I

IDTs, *see* Interdigital transducers
Impedance, acoustic, 40–41
Impulse model, interdigital transducer and, 114–115
Incident shear waves, 247–252
Inclusions, elastic-wave scattering from, 240–281
Indiana University, 84
Insertion loss/bandwidth triple-transit tradeoffs, 169–175

Institute of Radio Engineers, 61
Integrated optic spectrum analyzer, 185
Intense ultrasound, 41–60, *see also* Ultrasonics; Ultrasound
  early apparatus for, 44
  Loomis' work in, 43–45
  physical effects of, 41–42
  Wood's work in, 42–44
Interdigital transducers
  acoustic propagation loss in, 131–132
  array factors in, 100–101, 111–113
  bulk or plate waves and, 129–130
  circuit factors in, 154–155
  circuit-model analysis and design of, 99–187
  crossed-field model in, 137–142
  delta-function model in, 113–114
  electric admittance for, 136–137
  as filter elements, 102–110
  generalized circuit model of, 142–146
  impulse model of, 114–115
  phase-reversal, 152
  power reflection coefficient for, 128
  Rayleigh wave mode and, 133
  regeneration distortion and, 127
  second-order effects in, 1–2–132
  source-withdrawal weighting and, 120–122
  three-port circuit models of, 101, 132–160
  tradeoffs of bandwidth and other factors in, 169–176
  transfer functions in, 134–136
  vs. transversal filters, 106–109
Interferometer, acoustic, *see* Acoustic interferometer

J

Janovsky–Pohlman whistle, 64, 70
Johns Hopkins University, 42

K

Kelvin & Hughes, Ltd., 76
Kelvin–Voigt model of viscoelasticity, 221
Kirchhoff equations, for perfect gas, 50
Kneser liquids, 84
Kundt tubes, 10–12, 20–21, 57

L

Lamb wave dispersion curves, 235
Langevin sandwich transducer, 31–32, 51, 54–55, 78
Laser welds, AE monitoring of, 353
La Tour siren, 64
Leningrad Electro-technical Institute
Linear-FM dispersive transducer, 118–119
Lithium sulfate, 61
Loomis Laboratory, 41–46
Low-frequency oscillations, 27

M

Magnetic Barkhausen effect, 347–348
Magnetic field drives, 344–345
Magnetomechanical acoustic emission, 342–348
Magnetomechanical effect, in stress determination for ferromagentic materials, 354
Magnetostriction
  discovery of, 16
  later developments in, 62
  in submarine detection, 30
  in ultrasonics, 46–47
Magnetostrictive-based QG gear, on destroyers and submarines, 51
Magnetostrictive oscillators, 47
Magnetostrictive transducers, 67
Magnetostrictors, 46–47
Massachusetts Institute of Technology, 22, 78, 84
Matching circuits, circuit factors for, 165
Medical ultrasonics, 79–83
Meniere's disease, 82
Metallic glasses, acoustic emission from, 338–339, 346–347
Metallurgic processes
  acoustic emission in, 339–340
  ultrasonic applications in, 71–72
Metglass, acoustic emission from 347–349
Microcracks
  crack volume and, 327
  lifetime and propagation rate for, 332
  parameters of for various mild steel specimens, 330–331
Mittag–Leffler expansion, 245

Molecular theory of gases, 19
Molecular transformations, acoustic emission during, 335–340
Molten metals, ultrasonic treatment of, 72
Moving-coil transducer, 27
Musical tone, production of, 6

N

National Bureau of Standards, 33
National Research Council, 32
Naval Consulting Board, 43
Naval Research Laboratory, 51
NDT, *see* Nondestructive testing
New London Naval Experiment Station, 33
Nobel Prize, 18
Nondestructive testing
  acoustic emission in, 353
  advances in, 89
  pulse–echo, 75–77
  ultrasonics in, 74–79
Nondispersive bandpass filters, transducer design and, 177–180
Nondispersive filter transducer, 117–118
Nonlinear wave propagation, 15–16
Nonplane acoustic wave fronts, in apodized transducers, 124
Nuclei of strain, transient wave generation by, 354

O

Optical sensing, of acoustic emission transients, 334
Oscillators
  crystal, 36, *see also* Peizoelectricity
  low-frequency, 27

P

Partial-wave scattering amplitude, 206
Phonautograph, 8–9
Physical acoustics, *see also* Acoustics; Sound
  development of, 25
  progress in, 49–51
  ultrasonics and, 83–87
Physical effects, early awareness of, 41–42
Physical Optics (Wood), 42
Pierce acoustic interferometer, 38, 45, 49–50
Piezoelectric acoustic emission transducer, 318
Piezoelectric effect, 17
  in submarine detection, 29–31
Piezoelectric engineering, 35
Piezoelectric materials
  anisotropy of, 123–124
  man-made, 61
Piezoelectric resonators (piezoresonators), 36–37
Piezoelectric waves, 129
Piezoelectricity
  discovery of, 16
  telephone applications of, 35
Piezoelectricity (Cady), 61
Plastic strain rate, acoustic emission and, 341
Plastic welding, 69
Power ultrasonics, development of, 55–59
Prestressed sandwich transducer, 63
Princeton University, 45
Pulse–echo nondestructive testing, 75–77
  medical applications of, 80
Pyroelectricity, 17

Q

Quartz crystals
  displacement of by man-made crystals, 61
  as sound propagator, 31
  in vacuum-tube oscillators, 35
Quartz sandwich transducer, 31–32, 51, 54–55, 78
"Quartz wind," 37

R

RAC, *see* Reflective array compressor
Rayleigh circumferential wave, 225–226
Rayleigh disk, 21
Rayleigh-mode dispersion curves, for steel plate, 235
Rayleigh resonances, 207
Rayleigh waves, 19, 129
  crossed-field model and, 140
  crystal tensors and, 133
  dispersion curves for, 213–214
  scattering and, 206
Raytheon Company, 69

Reflective-array compressors, 100
unapodized dispersive transducers in, 184–187
Reflectoscope pulse–echo device, 76
Regeneration distortion, 127
Regge poles, 193, 212, 219, 246, 258
Resonance formulation, for cylinders and spheres, 197–202
Resonances, of submerged plates, 232–240
Resonance scattering, 191–292
quantum mechanical theory of, 192–193
rubber-coated steel shell and, 221
Reverberation control of gain, 84
Reverse scattering problem, 281–285
Rochelle-salt crystal array, 35, 51, 61
Royal Cancer Hospital, 80

S

St. Clair generator, 70
St. Clair transducer, 63–64
Savart wheel, in pitch determination, 5
SAW, *see* Surface acoustic wave
Scattering
acoustic-wave, *see* Acoustic-wave scattering
electromagnetic, *see* Electromagnetic scattering
inverse, 281–285
as reflection or diffraction process, 192
from solid spherical inclusions, 273–281
sound, 223–227
Scattering resonances, surface-wave interpretation of, 223–227
Scattering theory, classical, 194–197
Schlieren photography, invention of, 12–13
School of Physics and Chemistry (Paris), 29–30
Scophony Supersonic Television System, 49
"Searchlight" sonar, 85
"Second-order" effects, interdigital transducers and, 122–132
Sensitive flames, 9–13
Series-tuned transducers, 167–169
Series tuning, vs. shunt tuning, 177
Shear waves, incident, 247–251
Shells
coated, 208–218
resonance theory for, 208
spherical, 218–223
Shockley delay unit, 77–78
Shock waves, in Hartmann generator, 40
Shunt-tuned transducers, 166–167
Siemens Schucker equipment, 70
Silica firebrick, acoustic emissions from, 339
Simple matching circuits, circuit factors for, 164–165
"Singing arc," 22
Siren, acoustical, 8, 64
S matrix, 248–249
Smoke coagulation, ultrasound in, 70
Snell's law, 234
Sokolov patent, for ultrasonic testing, 54
Solid cylinders and spheres, resonance formulation for, 197–202
Solid inclusions, scattering of longitudinal waves from, 273–281
Solvay International Physics Institute, 34
Sonar, improvements in, 85–86
Song of Insects, The (Pierce), 53
Sonics (Hueter and Bolt), 89
Sono-luminescence, 66
Sound, *see also* Acoustic emission; Acoustics; Ultrasound
attenuation of, 13–15, 25
fluid flow analogy and, 14
fluid viscosity in propagation of, 14
heat conduction and, 15
in narrow tubes, 14–15
velocity of, 13–15
Sound propagation
finite displacement and, 16
fluid viscosity and, 14
Sound radiometer, 24
Sound scattering, tank experiment in, 223–227, *see also* Scattering
Sound velocity, finite displacement and, 13–16
Sound waves
in light intensity control, 48–49
speed and propagation of, 13–16
visualizing of, 48
Source-withdrawal weighting, 120–122
Spark discharges, 21–22
from condensers, 24
"Speaking arc," 22
Spheres, resonance formulation for, 197–202, *see also* Solid cylinders and spheres
Spherical fluid-filled cavities, 240–267
compressional waves and, 241–247
dispersion curves of surface waves for, 256

Spherical fluid-filled cavities, (*Continued*)
giant monopole resonance and, 262–267
incident shear waves and 247–251
modular surfaces for, 260–261
plots for, 251–254
Spherical shells, interpretation of in terms of surface waves, 218–223
Steam siren, 64
Steel
acoustic emission source for, 329
ductile fracture of, 321–323
magnetomechanical AE effect in, 343
Steel plate, Rayleigh and Lamb-mode dispersion curves for, 235
Steel–quartz–steel sandwich transducer, 31, 51
Stopband problem, double electrodes and, 125–126
Stroboscopic illumination techniques, 21
Submarine detection, 26–33
Chilowsky's work in, 29–30
piezoelectric effect in, 29–31
transducers in, 31–33
Submarine detectors, passive, 29
Submarine Signal Company, 27, 35, 51
Supersonic jet, impingement of on cavity, 39
Surface acoustic wave delay line, 102–104, *see also* Delay lines
tapped, 106
Surface acoustic wave devices, transducer design for, 99–187
Surface acoustic wave filters
"brick-wall," 178–179
bulk or plate waves and, 129–130
computer simulation of, 146–153
diffraction or beam steering and, 123–124
dispersive, 180–183
electrode resistivity and acoustic propagation loss in, 131–132
electromagnetic feedthrough and ground loops in, 130–131
filter insertion loss and, 161–163
specific transducer geometrics for, 177–187
three-port transducer networks for, 148–149
transducer design and, 109–110
types of, 179–180
Surface acoustic wave propagation, metal electrodes in, 125
Surface acoustical waves, regeneration distortion in, 127
Surface acoustic wave signal processing devices, 100
Surface acoustic wave transducers, as transversal filters, 161
Surface waves, spherical shells and, 218–220

T

Tapped delay lines, 183–184, *see also* Delay lines
Telecommunications Research Establishment, 77
Telephone receivers, impedance characteristics of, 23
TE mode, *see* Transverse electric mode
Texas, University of, 46
Texas Christian University, 46
Theory of Sound (Rayleigh), 18
Thermal Kaiser effect, 338
Thermionic triode, 23
Three-port circuit
for crossed-field model, 138–139
general capabilities of, 132–137
for generalized circuit model transducer, 145–146
Titania ceramics, piezoelectric properties of, 61
Titanic, S.S., 26–27
TM mode, *see* Transverse magnetic mode
Tradeoffs, in interdigital transducers, 170–176
Transducer(s)
capacitance, 317–318
direct-connection or "untuned," 165–166
electromagnetic, 63–64
interdigital, *see* Interdigital transducers
linear-FM dispersive, 118–119
magnetostrictive, 67
moving-coil, 27
nondispersive filter, 117–118
piezoelectric acoustic emission type, 318
quartz-focusing, 69
quartz–steel–quartz type, 31, 51
shunt- and series-tuned, 166–177
ultrasonic, 31–32
"untuned," 165–166
Transducer geometry
array-factor response and, 115–120
transducer response and, 110–122
for typical filter applications, 177–187

Transducer materials, ultrasonic generators and, 61–64
Transducer response, transducer geometry and, 110–122
Transducer transfer functions, in interdigital transducers, 134–136
Transduction mechanisms, 16–18
Transfer functions, for interdigital transducers, 135–136
Transient secondary system, schematic of, 320
Transient waves, "nuclei of strain" generation of, 354
Transversal filter, 104–106, *see also* Surface acoustic wave filters
  versatility of, 106–109
Transverse electric modes, 287
Transverse magnetic modes, 287
Treatise on Electricity and Magnetism (Maxwell), 18
Triple-transit signal or echo, 128–129
Triple-transit suppression, defined, 128
Tuning-fork tonemeter, 8
Tuxedo Park Laboratory, 44
Two-port network, for two-transducer filter, 146–148
Tyndall effect, 9

U

Ultrasonic agglomeration, 70
Ultrasonic burglar alarms, 78
Ultrasonic cavitation, 20, 38–39, 58
Ultrasonic cleaning, 67–68
Ultrasonic degassing, 71
Ultrasonic degreaser, 68
Ultrasonic dental drills, 83
Ultrasonic echo-ranging system, 34, 86
Ultrasonic emulsification, 70
Ultrasonic garage-door openers, 78
Ultrasonic generators
  transducer materials and, 61–64
  whistles as, 6
Ultrasonic machining, 67
Ultrasonic Physics (Richardson), 88
Ultrasonic plastic welding, 69
Ultrasonic radiation, harmful effects of, 82
Ultrasonics, *see also* Ultrasound
  acoustic emission and, 78–79
  acoustic streaming and, 66–67
  as aid to blind, 78
  birth of, 28–33
  cavitation and, 20, 38–39, 58, 65–66
  in chemical–biological processes, 72–74
  coagulation action in, 57
  "coming-of-age" of, 60–89
  commercial uses of, 89
  defined, 2
  in dentistry, 83
  drilling and cutting action in, 67
  emulsifying–dispersing action of, 56–57
  in fluids, 65–66
  growth of science of, 33–39
  high-power, 64–74
  Langevin's work in, 28–30
  Loomis Laboratory and, 41–46
  metallurgical applications of, 71–72
  in nondestructive testing, 74–79
  "old" techniques in, 39–40
  optical aspects of, 47–49
  in ore flotation, 71
  physical acoustics and, 25, 83–87
  postwar books on, 87–89
  postwar developments in, 60–89
  power, 55–59
  "sea birth" of, 26
  steel–quartz–steel transducer in, 31, 51
  submarine detection and, 26–33
  in surgery, 82
  technical literature on, 59–60
  underwater sound and, 83–87
  for underwater telephony, 34
  World War II developments in, 60–62
"Ultrasonic sickness," 82–83
Ultrasonic soldering, 68, 89
Ultrasonic therapy, 79
Ultrasonic tomography, 81
Ultrasonic underwater transmitters, 23
Ultrasonic welding, 69, 89
Ultrasound, *see also* Ultrasonics
  Edelmann whistle and, 20
  first uses of in testing and medicine, 53–55, 79
  history of, 2–20
  intense, *see* Intense ultrasound
  light diffraction by, 47
  in medical diagnosis, 79–80
  scientific uses of, 37, 53–55
  sea birth of, 28–29
  submarine detection with, 26–33
Underwater sound, 26–33, 85

Underwater ultrasound transmitters, 23
"Untuned" transducer, 165–166

V

Very-low-frequency generators, 27
Viscoelasticity, Kelvin–Voigt model of, 221
Vortex whistle, 64

W

Warren, S. D., Paper Company, 70
Washington Navy Yard, 51
Watson–Sommerfeld solution, in coated-shell analysis, 221
Watson transformation, 218–219
Wave propagation, mode conversion and, 241
Webster horn equation, 41
Weld strength, acoustic emission monitoring of, 352
Western Electric Company, 33, 35, 350
Whispering-gallery waves, 206–207, 213
Whistles
  heat radiation effects and, 15
  high-frequency, 15
  as ultrasonic generators, 6, 39, 64, 70
Withdrawal weighting technique, 120–122
World War I
  submarine detection in, 26–30
  ultrasound developments in, 30–32, 37
World War II, submarine detection in, 52

Y

"Yobell" geometry or specimen, 316–318, 320
  testing of, 317

Z

Zeiss Company, 71

Contents of Previous Volumes

Volume I, Part A—Methods and Devices

Wave Propagation in Fluids and Normal Solids—*R. N. Thurston*
Guided Wave Propagation in Elongated Cylinders and Plates—*T. R. Meeker and A. H. Meitzler*
Piezoelectric and Piezomagnetic Materials and Their Function in Transducers—*Don A. Berlincourt, Daniel R. Curran, and Hans Jaffe*
Ultrasonic Methods for Measuring the Mechanical Properties of Liquids and Solids—*H. J. McSkimin*
Use of Piezoelectric Crystals and Mechanical Resonators in Filters and Oscillators—*Warren P. Mason*
Guided Wave Ultrasonic Delay Lines—*John E. May, Jr.*
Multiple Reflection Ultrasonic Delay Lines—*Warren P. Mason*

Volume I, Part B—Methods and Devices

The use of High- and Low-Amplitude Ultrasonic Waves for Inspection and Processing—*Benson Carlin*
Physics of Acoustic Cavitation in Liquids—*H. G. Flynn*
Semiconductor Transducers—General Considerations—*Warren P. Mason*
Use of Semiconductor Transducers in Measuring Strains, Accelerations, and Displacements—*R. N. Thurston*
Use of *p–n* Junction Semiconductor Transducers in Pressure and Strain Measurements—*M. E. Sikorski*
The Depletion Layer and Other High-Frequency Transducers Using Fundamental Modes—*D. L. White*
The Design of Resonant Vibrators—*Edward Eisner*

Volume II, Part A—Properties of Gases, Liquids, and Solutions

Transmission of Sound Waves in Gases at Very Low Pressure—*Martin Greenspan*
Phenomenological Theory of the Relaxation Phenomena in Gases—*H. J. Bauer*
Relaxation Processes in Gases—*H. O. Kneser*
Thermal Relaxation in Liquids—*John Lamb*
Structural and Shear Relaxation in Liquids—*T. A. Litovitz and C. M. Davis*
The Propagation of Ultrasonic Waves in Electrolytic Solutions—*John Stuehr and Ernest Yeager*

Volume II, Part B—Properties of Polymers and Nonlinear Acoustics

Relaxations in Polymer Solutions, Liquids and Gels—*W. Philoppoff*
Relaxation Spectra and Relaxation Processes in Solid Polymers and Glasses—*I. L. Hopkins and C. R. Kurkjian*
Volume Relaxations in Amorphous Polymers—*Robert S. Marvin and John E. McKinney*
Nonlinear Acoustics—*Robert T. Beyer*
Acoustic Streaming—*Wesley Le Mars Nyborg*
Use of Light Diffraction in Measuring the Parameter of Nonlinearity of Liquids and the Photoelastic Constants of Solids—*L. E. Hargrove and K. Achyuthan*

Volume III, Part A—Effect of Imperfections

Anelasticity and Internal Friction Due to Point Defects in Crystals—*B. S. Berry and A. S. Nowick*

Determination of the Diffusion Coefficient of Impurities by Anelastic Methods—*Charles Wert*

Bordoni Peak in Face-Centered Cubic Metals—*D. H. Niblett*

Dislocation Relaxations in Face-Centered Cubic Transition Metals—*R. H. Chambers*

Ultrasonic Methods in the Study of Plastic Deformation—*Rohn Truell, Charles Elbaum, and Akira Hikata*

Internal Friction and Basic Fatigue Mechanisms in Body-Centered Cubic Metals, Mainly Iron and Carbon Steels—*W. J. Bratina*

Use of Anelasticity in Investigating Radiation Damage and the Diffusion of Point Defects—*Donald O. Thompson and Victor K. Paré*

Kinks in Dislocation Lines and Their Effects on the Internal Friction in Crystals—*Alfred Seeger and Peter Schiller*

Volume III, Part B—Lattice Dynamics

Use of Sound Velocity Measurements in Determining the Debye Temperature of Solids—*George A. Alers*

Determination and Some Uses of Isotropic Elastic Constants of Polycrystalline Aggregates Using Single-Crystal Data—*O. L. Anderson*

The Effect of Light on Alkali Halide Crystals—*Robert B. Gordon*

Magnetoelastic Interactions in Ferromagnetic Insulators—*R. C. LeCraw and R. L. Comstock*

Effect of Thermal and Phonon Processes on Ultrasonic Attenuation—*P. G. Klemens*

Effects of Impurities and Phonon Processes on the Ultrasonic Attenuation of Germanium, Crystal Quartz, and Silicon—*Warren P. Mason*

Attenuation of Elastic Waves in the Earth—*L. Knopoff*

Volume IV, Part A—Applications to Quantum and Solid State Physics

Transmission and Amplification of Acoustic Waves in Piezoelectric Semiconductors—*J. H. McFee*

Paramagnetic Spin-Phonon Interaction in Crystals—*Edmund B. Tucker*

Interaction of Acoustic Waves with Nuclear Spins in Solids—*D. I. Bolef*

Resonance Absorption—*Leonard N. Liebermann*

Fabrication of Vapor-Deposited Thin Film Piezoelectric Transducers for the Study of Phonon Behavior in Dielectric Materials at Microwave Frequencies—*J. de Klerk*

The Vibrating String Model of Dislocation Damping—*A. V. Granato and K. Lücke*

The Measurement of Very Small Sound Velocity Changes and Their Uses in the Study of Solids—*G. A. Alers*

Acoustic Wave and Dislocation Damping in Normal and Superconducting Metals and in Doped Semiconductors—*Warren P. Mason*

Ultrasonics and the Fermi Surfaces of the Monovalent Metals—*J. Roger Peverley*

Volume IV, Part B—Applications to Quantum and Solid State Physics

Oscillatory Magnetoacoustic Phenomena in Metals—*B. W. Roberts*

Transmission of Sound in Molten Metals—*G. M. B. Webber and R. W. B. Stephens*

Acoustic and Plasma Waves in Ionized Gases—*G. M. Sessler*

Relaxation and Resonance of Markovian Systems—*Roger Cerf*

Magnetoelastic Properties of Yttrium-Iron Garnet—*Walter Strauss*

Ultrasonic Attenuation Caused by Scattering in Polycrystalline Media—*Emmanuel P. Papadakis*

Sound Velocities in Rocks and Minerals: Experimental Methods, Extrapolations to Very High Pressures, and Results—*Orson L. Anderson and Robert C. Liebermann*

Volume V

Acoustic Wave Propagation in High Magnetic Fields—*Y. Shapira*
Impurities and Anelasticity in Crystalline Quartz—*David B. Fraser*
Observation of Resonant Vibrations and Defect Structure in Single Crystals by X-ray Diffraction Topography—*W. J. Spencer*
Wave Packet Propagation and Frequency-Dependent Internal Friction—*M. Elices and F. Garcia-Moliner*
Coherent Elastic Wave Propagation in Quartz at Ultramicrowave Frequencies—*John Ilukor and E. H. Jacobsen*
Heat Pulse Transmission—*R. J. von Gutfeld*

Volume VI

Light Scattering as a Probe of Phonons and Other Excitations—*Paul A. Fleury*
Acoustic Properties of Materials of the Perovskite Structure—*Harrison H. Barrett*
Properties of Elastic Surface Waves—*G. W. Farnell*
Dynamic Shear Properties of Solvents and Polystyrene Solutions from 20 to 300 MHz—*R. S. Moore and J. H. McSkimin*
The Propagation of Sound in Condensed Helium—*S. G. Eckstein, Y. Eckstein, J. B. Ketterson, and J. H. Vignos*

Volume VII

Ultrasonic Attenuation in Superconductors: Magnetic Field Effects—*M. Gottlieb, M. Garbuny, and C. K. Jones*
Ultrasonic Investigation of Phase Transitions and Critical Points—*Carl W. Garland*
Ultrasonic Attenuation in Normal Metals and Superconductors: Fermi-Surface Effects—*J. A. Rayne and C. K. Jones*
Excitation, Detection, and Attenuation of High-Frequency Elastic Surface Waves—*K. Dransfeld and E. Salzmann*
Interaction of Light with Ultrasound: Phenomena and Applications Spin-Phonon Spectrometer—*R. W. Damon, W. T. Maloney, and D. H. McMahon*

Volume VIII

Spin-Phonon Spectrometer—*Charles H. Anderson and Edward S. Sabisky*
Landau Quantum Oscillations of the Velocity of Sound and the Strain Dependence of the Fermi Surface—*L. R. Testardi and J. H. Condon*
High-Frequency Continuous Wave Ultrasonics—*D. I. Bolef and J. G. Miller*
Ultrasonic Measurements at Very High Pressures—*P. Heydemann*
Third-Order Elastic Constants and Thermal Equilibrium Properties of Solids—*J. Holder and A. V. Granato*
Interactions of Sound Waves with Thermal Phonons in Dielectric Crystals—*Humphrey J. Maris*
Internal Friction at Low Frequencies Due to Dislocations: Applications to Metals and Rock Mechanics—*Warren P. Mason*

Volume IX

Difference in Electron Drag Stresses on Dislocation Motion in the Normal and the Superconducting States for Type I and Type II Superconductors—*M. Suenaga and J. M. Galligan*
Elastic Wave Propagation in Thin Layers—*G. W. Farnell and E. L. Adler*
Solid State Control Elements Operating on Piezoelectric Principles—*F. L. N-Nagy and G. C. Joyce*
Monolithic Crystal Filters—*W. J. Spencer*

Architectural Acoustics



MARSHALL LONG

AP

A R C H I T E C T U R A L A C O U S T I C S

This page intentionally left blank

ARCHITECTURAL ACOUSTICS

by Marshall Long

from the
Applications of Modern Acoustics Series
Edited by Moises Levy and Richard Stern



ELSEVIER
ACADEMIC
PRESS

Amsterdam • Boston • Heidelberg • London • New York • Oxford
Paris • San Diego San Francisco • Singapore • Sydney • Tokyo

Elsevier Academic Press
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
525 B Street, Suite 1900, San Diego, California 92101-4495, USA
84 Theobald's Road, London WC1X 8RR, UK

This book is printed on acid-free paper.

Copyright © 2006, Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone: (+44) 1865 843830, fax: (+44) 1865 853333, E-mail: permissions@elsevier.co.uk. You may also complete your request on-line via the Elsevier homepage (http://elsevier.com), by selecting "Customer Support" and then "Obtaining Permissions."

Cover image: The cover shows Grosser Musikvereinssaal in Vienna, Austria. The photograph was provided by AKG Acoustics, U.S., and is reproduced with permission.

Library of Congress Cataloging-in-Publication Data
Application submitted

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 13: 978-0-12-455551-8
ISBN 10: 0-12-455551-9

For all information on all Elsevier Academic Press publications visit our Web site at www.books.elsevier.com

Printed in the United States of America
05 06 07 08 09 10 9 8 7 6 5 4 3 2 1

Working together to grow libraries in developing countries
www.elsevier.com | www.bookaid.org | www.sabre.org
ELSEVIER BOOK AID International Sabre Foundation

The preparation of this book, which spanned more than ten years, took place in snatches of time – a few hours every evening and several more each weekend. It was time that was taken from commitments to family, home maintenance projects, teaching, and other activities forgone, of a pleasurable and useful nature. During that time our two older sons grew through their teens and went off to college. Our youngest son cannot remember a time when his father did not go upstairs to work every evening. So it is to my wife Marilyn and our sons Jamie, Scott, and Kevin that I dedicate this work. I am grateful for the time. I hope it was worth it. And to my environmentally conscious children, I hope it is worth the trees.

This page intentionally left blank

CONTENTS

PREFACE **xxv**
ACKNOWLEDGMENTS **xxvii**

1 HISTORICAL INTRODUCTION **1**
1.1 GREEK AND ROMAN PERIOD (650 BC–AD 400) 1
Early Cultures 1
Greeks 2
Romans 4
Vitruvius Pollio 5
1.2 EARLY CHRISTIAN PERIOD (AD 400–800) 7
Rome and the West 7
Eastern Roman Empire 8
1.3 ROMANESQUE PERIOD (800–1100) 10
1.4 GOTHIC PERIOD (1100–1400) 11
Gothic Cathedrals 11
1.5 RENAISSANCE PERIOD (1400–1600) 14
Renaissance Churches 14
Renaissance Theaters 15
1.6 BAROQUE PERIOD (1600–1750) 16
Baroque Churches 16
Baroque Theaters 17
Italian Opera Houses 17
Baroque Music 18
Protestant Music 19
1.7 ORIGINS OF SOUND THEORY 20
1.8 CLASSICAL PERIOD (1750–1825) 21
1.9 ROMANTIC PERIOD (1825–1900) 23
Shoebox Halls 26
1.10 BEGINNINGS OF MODERN ACOUSTICS 30
1.11 TWENTIETH CENTURY 33

2 FUNDAMENTALS OF ACOUSTICS **37**
2.1 FREQUENCY AND WAVELENGTH 37
Frequency 37
Wavelength 37

	Frequency Spectrum	40
	Filters	40
2.2	SIMPLE HARMONIC MOTION	40
	Vector Representation	43
	The Complex Plane	43
	The Complex Exponential	44
	Radial Frequency	45
	Changes in Phase	46
2.3	SUPERPOSITION OF WAVES	46
	Linear Superposition	46
	Beats	48
2.4	SOUND WAVES	50
	Pressure Fluctuations	50
	Sound Generation	50
	Wavelength of Sound	51
	Velocity of Sound	51
	Waves in Other Materials	55
2.5	ACOUSTICAL PROPERTIES	55
	Impedance	55
	Intensity	57
	Energy Density	59
2.6	LEVELS	59
	Sound Levels — Decibels	59
	Sound Pressure Level	61
	Sound Power Level	62
2.7	SOURCE CHARACTERIZATION	65
	Point Sources and Spherical Spreading	65
	Sensitivity	67
	Directionality, Directivity, and Directivity Index	68
	Line Sources	70
	Planar Sources	71
3	**HUMAN PERCEPTION AND REACTION TO SOUND**	**73**
3.1	HUMAN HEARING MECHANISMS	73
	Physiology of the Ear	73
3.2	PITCH	77
	Critical Bands	77
	Consonance and Dissonance	78
	Tone Scales	79
	Pitch	81
3.3	LOUDNESS	81
	Comparative Loudness	81
	Loudness Levels	82
	Relative Loudness	83

Electrical Weighting Networks 84
Noise Criteria Curves (NC and RC) 85
Just Noticeable Difference 88
Environmental Impact 90
3.4 INTELLIGIBILITY 91
Masking 91
Speech Intelligibility 93
Speech Interference Level 94
Articulation Index 95
AL_{CONS} 96
Privacy 97
3.5 ANNOYANCE 98
Noisiness 98
Time Averaging – L_{eq} 100
Twenty-Four Hour Metrics – L_{dn} and CNEL 101
Annoyance 101
3.6 HEALTH AND SAFETY 105
Hearing Loss 105
3.7 OTHER EFFECTS 107
Precedence Effect and the Perception of Echoes 107
Perception of Direction 110
Binaural Sound 113

4 ACOUSTIC MEASUREMENTS AND NOISE METRICS 115
4.1 MICROPHONES 115
Frequency Response 118
Directional Microphones 118
Sound Field Considerations 120
4.2 SOUND LEVEL METERS 121
Meter Calibration 122
Meter Ballistics 124
Meter Range 125
Detectors 125
Filters 125
4.3 FIELD MEASUREMENTS 125
Background Noise 127
Time-Varying Sources 129
Diurnal (24-Hour) Traffic Measurements 130
4.4 BROADBAND NOISE METRICS 132
Bandwidth Corrections 133
Duration Corrections 135
Variability Corrections 135
Sound Exposure Levels 136
Single Event Noise Exposure Level 137

	4.5	BAND LIMITED NOISE METRICS	137
		Preferred Noise Criterion (PNC) Curves	138
		Balanced Noise Criterion (NCB) Curves (Beranek, 1989)	139
		Other Octave-Band Metrics	140
		Octave-Band Calculations	141
		Third-Octave Bandwidth Metrics	142
		Aircraft Noise Rating Systems	142
		Narrow-Band Analysis	143
	4.6	SPECIALIZED MEASUREMENT TECHNIQUES	145
		Time-Delay Spectrometry	145
		Energy-Time Curves	146
		Sound Intensity Measurements	148
		Modulation Transfer Function and RASTI	149
		Speech Transmission Index	151
		RASTI	154
5	**ENVIRONMENTAL NOISE**		**157**
	5.1	NOISE CHARACTERIZATION	157
		Fixed Sources	157
		Moving Sources	158
		Partial Line Sources	159
	5.2	BARRIERS	161
		Point Source Barriers	161
		Practical Barrier Constraints	162
		Line Source Barriers	163
		Barrier Materials	165
		Roadway Barriers	166
	5.3	ENVIRONMENTAL EFFECTS	167
		Air Attenuation	168
		Attenuation Due to Ground Cover	175
		Grazing Attenuation	175
		Focusing and Refraction Effects	177
		Combined Effects	181
		Doppler Effect	182
	5.4	TRAFFIC NOISE MODELING	183
		Soft Ground Approximation	184
		Geometrical Mean Distance	186
		Barrier Calculations	186
		Roadway Computer Modeling	188
		Traffic Noise Spectra	189
	5.5	RAILROAD NOISE	189
	5.6	AIRCRAFT NOISE	194
6	**WAVE ACOUSTICS**		**199**
	6.1	RESONANCE	199
		Simple Oscillators	199

Air Spring Oscillators 201
Helmholtz Resonators 203
Neckless Helmholtz Resonators 203
6.2 WAVE EQUATION 205
One-Dimensional Wave Equation 205
Three-Dimensional Wave Equation 207
6.3 SIMPLE SOURCES 208
Monopole Sources 208
Doublet Sources 208
Dipole Sources and Noise Cancellation 210
Arrays of Simple Sources 211
Continuous Line Arrays 213
Curved Arrays 214
Phased Arrays 217
Source Alignment and Comb Filtering 217
Comb Filtering and Critical Bands 218
6.4 COHERENT PLANAR SOURCES 219
Piston in a Baffle 219
Coverage Angle and Directivity 222
Loudspeaker Arrays and the Product Theorem 222
Rectangular Pistons 224
Force on a Piston in a Baffle 225
6.5 LOUDSPEAKERS 226
Cone Loudspeakers 226
Horn Loudspeakers 228
Constant-Directivity Horns 230
Cabinet Arrays 233
Baffled Low-Frequency Systems 233

7 SOUND AND SOLID SURFACES 235
7.1 PERFECTLY REFLECTING INFINITE SURFACES 235
Incoherent Reflections 235
Coherent Reflections—Normal Incidence 236
Coherent Reflections—Oblique Incidence 238
Coherent Reflections—Random Incidence 239
Coherent Reflections—Random Incidence, Finite Bandwidth 239
7.2 REFLECTIONS FROM FINITE OBJECTS 240
Scattering from Finite Planes 240
Panel Arrays 244
Bragg Imaging 245
Scattering from Curved Surfaces 247
Combined Effects 248
Whispering Galleries 249
7.3 ABSORPTION 249
Reflection and Transmission Coefficients 249

	Impedance Tube Measurements	250
	Oblique Incidence Reflections—Finite Impedance	251
	Calculated Diffuse Field Absorption Coefficients	254
	Measurement of Diffuse Field Absorption Coefficients	255
	Noise Reduction Coefficient (NRC)	255
	Absorption Data	255
	Layering Absorptive Materials	256
7.4	ABSORPTION MECHANISMS	261
	Porous Absorbers	261
	Spaced Porous Absorbers—Normal Incidence, Finite Impedance	263
	Porous Absorbers with Internal Losses—Normal Incidence	265
	Empirical Formulas for the Impedance of Porous Materials	266
	Thick Porous Materials with an Air Cavity Backing	267
	Practical Considerations in Porous Absorbers	268
	Screened Porous Absorbers	269
7.5	ABSORPTION BY NONPOROUS ABSORBERS	271
	Unbacked Panel Absorbers	271
	Air Backed Panel Absorbers	272
	Perforated Panel Absorbers	274
	Perforated Metal Grilles	276
	Air Backed Perforated Panels	276
7.6	ABSORPTION BY RESONANT ABSORBERS	277
	Helmholtz Resonator Absorbers	277
	Mass-Air-Mass Resonators	278
	Quarter-Wave Resonators	279
	Absorption by Seats	282
	Quadratic-Residue Diffusers	282
8	**SOUND IN ENCLOSED SPACES**	**285**
8.1	STANDING WAVES IN PIPES AND TUBES	285
	Resonances in Closed Tubes	285
	Standing Waves in Closed Tubes	286
	Standing Waves in Open Tubes	287
	Combined Open and Closed Tubes	288
8.2	SOUND PROPAGATION IN DUCTS	289
	Rectangular Ducts	289
	Changes in Duct Area	291
	Expansion Chambers and Mufflers	292
8.3	SOUND IN ROOMS	293
	Normal Modes in Rectangular Rooms	294
	Preferred Room Dimensions	297
8.4	DIFFUSE-FIELD MODEL OF ROOMS	298
	Schroeder Frequency	298
	Mean Free Path	299

		Decay Rate of Sound in a Room	299
		Sabine Reverberation Time	300
		Norris Eyring Reverberation Time	301
		Derivation of the Sabine Equation	301
		Millington Sette Equation	302
		Highly Absorptive Rooms	302
		Air Attenuation in Rooms	302
		Laboratory Measurement of the Absorption Coefficient	303
	8.5	REVERBERANT FIELD EFFECTS	304
		Energy Density and Intensity	304
		Semireverberant Fields	305
		Room Effect	305
		Radiation from Large Sources	307
		Departure from Diffuse Field Behavior	307
		Reverberant Falloff in Long Narrow Rooms	309
		Reverberant Energy Balance in Long Narrow Rooms	310
		Fine Structure of the Sound Decay	312
9	**SOUND TRANSMISSION LOSS**		**315**
	9.1	TRANSMISSION LOSS	315
		Sound Transmission Between Reverberant Spaces	315
		Measurement of the Transmission Loss	316
		Sound Transmission Class (STC)	317
		Field Sound Transmission Class (FSTC)	318
		Noise Reduction and Noise Isolation Class (NIC)	318
	9.2	SINGLE PANEL TRANSMISSION LOSS THEORY	319
		Free Single Panels	319
		Mass Law	320
		Large Panels—Bending and Shear	322
		Thin Panels—Bending Waves and the Coincidence Effect	323
		Thick Panels	326
		Finite Panels—Resonance and Stiffness Considerations	330
		Design of Single Panels	330
		Spot Laminating	332
	9.3	DOUBLE PANEL TRANSMISSION LOSS THEORY	333
		Free Double Panels	333
		Cavity Insulation	335
		Double-Panel Design Techniques	338
	9.4	TRIPLE-PANEL TRANSMISSION LOSS THEORY	342
		Free Triple Panels	342
		Comparison of Double and Triple-Panel Partitions	343
	9.5	STRUCTURAL CONNECTIONS	345
		Point and Line Connections	345
		Transmission Loss of Apertures	347

10 SOUND TRANSMISSION IN BUILDINGS 351
10.1 DIFFUSE FIELD SOUND TRANSMISSION 351
Reverberant Source Room 351
Sound Propagation through Multiple Partitions 353
Composite Transmission Loss with Leaks 353
Transmission into Absorptive Spaces 353
Transmission through Large Openings 355
Noise Transmission Calculations 356
10.2 STC RATINGS OF VARIOUS WALL TYPES 357
Laboratory vs Field Measurements 357
Single Wood Stud Partitions 357
Single Metal Stud Partitions 357
Resilient Channel 358
Staggered-Stud Construction 361
Double-Stud Construction 362
High-Mass Constructions 363
High Transmission Loss Constructions 364
10.3 DIRECT FIELD SOUND TRANSMISSION 365
Direct Field Sources 365
Direct Field Transmission Loss 366
Free Field—Normal Incidence 367
Free Field—Non-normal Incidence 367
Line Source—Exposed Surface Parallel to It 367
Self Shielding and G Factor Corrections 368
10.4 EXTERIOR TO INTERIOR NOISE TRANSMISSION 369
Exterior Walls 369
Windows 369
Doors 372
Electrical Boxes 375
Aircraft Noise Isolation 378
Traffic Noise Isolation 378

11 VIBRATION AND VIBRATION ISOLATION 381
11.1 SIMPLE HARMONIC MOTION 381
Units of Vibration 381
11.2 SINGLE DEGREE OF FREEDOM SYSTEMS 382
Free Oscillators 382
Damped Oscillators 383
Damping Properties of Materials 385
Driven Oscillators and Resonance 386
Vibration Isolation 387
Isolation of Sensitive Equipment 391
Summary of the Principles of Isolation 392

11.3 VIBRATION ISOLATORS 392
Isolation Pads (Type W, WSW) 393
Neoprene Mounts (Type N, ND) 393
Steel Springs (Type V, O, OR) 393
Hanger Isolators (Type HN, HS, HSN) 394
Air Mounts (AS) 394
Support Frames (Type IS, CI, R) 394
Isolator Selection 395
11.4 SUPPORT OF VIBRATING EQUIPMENT 395
Structural Support 395
Inertial Bases 399
Earthquake Restraints 400
Pipe Isolation 401
Electrical Connections 402
Duct Isolation 402
11.5 TWO DEGREE OF FREEDOM SYSTEMS 404
Two Undamped Oscillators 404
Two Damped Oscillators 405
11.6 FLOOR VIBRATIONS 407
Sensitivity to Steady Floor Vibrations 407
Sensitivity to Transient Floor Vibrations 407
Vibrational Response to an Impulsive Force 409
Response to an Arbitrary Force 410
Response to a Step Function 411
Vibrational Response of a Floor to Footfall 412
Control of Floor Vibrations 413

12 NOISE TRANSMISSION IN FLOOR SYSTEMS 417
12.1 TYPES OF NOISE TRANSMISSION 417
Airborne Noise Isolation 417
Footfall 417
Structural Deflection 418
Squeak 418
12.2 AIRBORNE NOISE TRANSMISSION 418
Concrete Floor Slabs 418
Concrete on Metal Pans 419
Wood Floor Construction 420
Resiliently Supported Ceilings 420
Floating Floors 423
12.3 FOOTFALL NOISE 424
Impact Insulation Class—IIC 424
Impact Insulation Class Ratings 427
Vibrationally Induced Noise 427
Mechanical Impedance of a Spring Mass System 430

Driving Point Impedance 432
Power Transmitted through a Plate 433
Impact Generated Noise 434
Improvement Due to Soft Surfaces 437
Improvement Due to Locally Reacting Floating Floors 440
Improvement Due to Resonantly Reacting Floating Floors 440
12.4 STRUCTURAL DEFLECTION 441
Floor Deflection 441
Low-Frequency Tests 444
Structural Isolation of Floors 446
12.5 FLOOR SQUEAK 447
Shiners 447
Uneven Joists 449
Hangers 449
Nailing 449

13 NOISE IN MECHANICAL SYSTEMS 451
13.1 MECHANICAL SYSTEMS 451
Manufacturer Supplied Data 453
Airborne Calculations 453
13.2 NOISE GENERATED BY HVAC EQUIPMENT 453
Refrigeration Equipment 454
Cooling Towers and Evaporative Condensers 454
Air Cooled Condensers 456
Pumps 456
13.3 NOISE GENERATION IN FANS 457
Fans 458
Fan Coil Units and Heat Pumps 462
VAV Units and Mixing Boxes 464
13.4 NOISE GENERATION IN DUCTS 466
Flow Noise in Straight Ducts 466
Noise Generated by Transitions 469
Air Generated Noise in Junctions and Turns 469
Air Generated Noise in Dampers 472
Air Noise Generated by Elbows with Turning Vanes 473
Grilles, Diffusers, and Integral Dampers 474
13.5 NOISE FROM OTHER MECHANICAL EQUIPMENT 476
Air Compressors 476
Transformers 477
Reciprocating Engines and Emergency Generators 478

14 SOUND ATTENUATION IN DUCTS 481
14.1 SOUND PROPAGATION THROUGH DUCTS 481
Theory of Propagation in Ducts with Losses 481

Attenuation in Unlined Rectangular Ducts 486
Attenuation in Unlined Circular Ducts 486
Attenuation in Lined Rectangular Ducts 487
Attenuation of Lined Circular Ducts 487
Flexible and Fiberglass Ductwork 488
End Effect in Ducts 488
Split Losses 490
Elbows 490
14.2 SOUND PROPAGATION THROUGH PLENUMS 492
Plenum Attenuation—Low-Frequency Case 492
Plenum Attenuation—High Frequency Case 493
14.3 SILENCERS 495
Dynamic Insertion Loss 496
Self Noise 496
Back Pressure 497
14.4 BREAKOUT 497
Transmission Theory 498
Transmission Loss of Rectangular Ducts 500
Transmission Loss of Round Ducts 501
Transmission Loss of Flat Oval Ducts 502
14.5 BREAK-IN 503
Theoretical Approach 503
14.6 CONTROL OF DUCT BORNE NOISE 504
Duct Borne Calculations 504

15 DESIGN AND CONSTRUCTION OF MULTIFAMILY DWELLINGS 509
15.1 CODES AND STANDARDS 510
Sound Transmission Class—STC 510
Reasonable Expectation of the Buyer 512
Impact Insulation Class—IIC 513
Property Line Ordinances 514
Exterior to Interior Noise Standards 515
15.2 PARTY WALL CONSTRUCTION 515
General Principles 515
Party Walls 516
Structural Floor Connections 518
Flanking Paths 519
Electrical Boxes 519
Wall Penetrations 520
Holes 521
15.3 PARTY FLOOR-CEILING SEPARATIONS 522
Airborne Noise Isolation 522
Structural Stiffness 523
Structural Decoupling 524

Floor Squeak 527
Floor Coverings 528
15.4 PLUMBING AND PIPING NOISE 529
Supply Pipe 529
Water Hammer 534
Waste Stacks 535
Tubs, Toilets, and Showers 535
Pump and Piping Vibrations 536
Fluid Pulsations 537
15.5 MECHANICAL EQUIPMENT 537
Split Systems 537
Packaged Units 538
15.6 APPLIANCES AND OTHER SOURCES OF NOISE 540
Stairways 540
Appliances 540
Jacuzzis 540
Trash Chutes 541
Elevator Shafts and Equipment Rooms 541
Garage Doors 541

16 DESIGN AND CONSTRUCTION OF OFFICE BUILDINGS 543
16.1 SPEECH PRIVACY IN OPEN OFFICES 543
Privacy 543
Privacy Calculations 544
Articulation Weighted Ratings 548
Speech Reduction Rating and Privacy 550
Source Control 551
Partial Height Panels 553
Absorptive and Reflective Surfaces 556
Open-Plan Ceilings 558
Masking Sound 560
Degrees of Privacy 562
16.2 SPEECH PRIVACY IN CLOSED OFFICES 563
Private Offices 563
Full-Height Walls 563
Plenum Flanking 564
Duct Flanking 567
Exterior Curtain Walls 567
Divisible Rooms 569
Masking in Closed Offices 569
16.3 MECHANICAL EQUIPMENT 571
System Layout 571
Mechanical Equipment Rooms 572
Roof-Mounted Air Handlers 574

Fan Coil and Heat Pump Units 576
Emergency Generators 577

17 DESIGN OF ROOMS FOR SPEECH 579
17.1 GENERAL ACOUSTICAL REQUIREMENTS 579
General Considerations 579
Adequate Loudness 579
Floor Slope 580
Sound Distribution 583
Reverberation 585
Signal-to-Noise Ratio 587
Acoustical Defects 587
17.2 SPEECH INTELLIGIBILITY 588
Speech-Intelligibility Tests 588
Energy Buildup in a Room 589
Room Impulse Response 590
Speech-Intelligibility Metrics—Articulation Index (AI) 592
Articulation Loss of Consonants (AL_{cons}) 593
Speech Transmission Index (STI) 596
Signal-to-Noise Ratios (C_t and U_t) 599
Weighted Signal-to-Noise Ratios (C_t^{α} and U_t^{α}) 600
A-Weighted Signal-to-Noise Ratio 602
Comparison of Speech-Intelligibility Metrics 602
17.3 DESIGN OF ROOMS FOR SPEECH INTELLIGIBILITY 603
The Cocktail Party Effect 603
Restaurant Design 604
Conference Rooms 606
Classrooms 606
Small Lecture Halls 607
Large Lecture Halls 607
17.4 MOTION PICTURE THEATERS 609
Reverberation Times 610

18 SOUND REINFORCEMENT SYSTEMS 611
18.1 LOUDSPEAKER SYSTEMS 611
Loudspeaker Types 611
Loudness 612
Bandwidth 614
Low-Frequency Loudspeakers 615
Loudspeaker Systems 615
Distributed Loudspeaker Systems 615
Single Clusters 617
Multiple Clusters 617
Other Configurations 618

18.2 SOUND SYSTEM DESIGN 618
Coverage 619
Intelligibility 619
Amplifier Power Handling 621
Electrical Power Requirements 622
Heat Load 623
Time Coincidence 624
Imaging 625
Feedback 627
Multiple Open Microphones 631
Equalization 631
Architectural Sensitivity 634
18.3 CHARACTERIZATION OF TRANSDUCERS 635
Microphone Characterization 635
Loudspeaker Characterization 638
The Calculation of the On-axis Directivity 642
18.4 COMPUTER MODELING OF SOUND SYSTEMS 644
Coordinate Systems and Transformation Matrices 644
Determination of the Loudspeaker Coordinate System 646
Directivity Angles in Loudspeaker Coordinates 649
Multiple Loudspeaker Contributions 650

19 DESIGN OF ROOMS FOR MUSIC 653
19.1 GENERAL CONSIDERATIONS 653
The Language of Music 653
The Influence of Recording 653
Concert Halls 655
Opera Houses 656
19.2 GENERAL DESIGN PARAMETERS 657
The Listening Environment 657
Hall Size 658
Hall Shape 658
Hall Volume 658
Surface Materials 659
Balconies and Overhangs 660
Seating 661
Platforms 663
Orchestra Shells 664
Pits 665
19.3 QUANTIFIABLE ACOUSTICAL ATTRIBUTES 666
Studies of Subjective Preference 667
Modeling Subjective Preferences 670
Early Reflections, Intimacy, and Clarity 671
Liveness, Reverberation Time, and Early Decay Time 674

Envelopment, Lateral Reflections, and Interaural Cross-correlation 675
Loudness, G_{mid}, Volume, and Volume per Seat 678
Warmth and Bass Response 678
Diffusion, SDI 681
Ensemble, Blend, and Platform Acoustics 681
19.4 CONCERT HALLS 682
Grosser Musikvereinssaal, Vienna, Austria 682
Boston Symphony Hall, Boston, MA, USA 684
Concertgebouw, Amsterdam, Netherlands 685
Philharmonie Hall, Berlin, Germany 687
Eugene McDermott Concert Hall in the Morton H. Meyerson Symphony Center, Dallas, TX, USA 688
19.5 OPERA HALLS 691
Theatro Colon, Buenos Aires, Argentina 691
Theatro Alla Scala, Milan, Italy 692

20 DESIGN OF MULTIPURPOSE AUDITORIA AND SANCTUARIES 697
20.1 GENERAL DESIGN CONSIDERATIONS 697
Program 698
Room Shape 698
Seating 700
Room Volume 701
Reverberation Time 701
Absorption 701
Balconies 702
Ceiling Design 703
Audio Visual Considerations 705
20.2 DESIGN OF SPECIFIC ROOM TYPES 706
Small Auditoria 706
Mid-Sized Theaters 710
Large Auditoria 711
Traditional Churches 712
Large Churches 715
Synagogues 719
20.3 SPECIALIZED DESIGN PROBLEMS 720
Wall and Ceiling Design 720
Shell Design 720
Platform Risers 724
Pit Design 725
Diffusion 725
Variable Absorption 728
Variable Volume 729
Coupled Chambers 730
Sound System Integration 734
Electronic Augmentation 736

21 DESIGN OF STUDIOS AND LISTENING ROOMS 741
21.1 SOUND RECORDING 741
Early Sound Recording 741
Recording Process 742
Recording Formats 744
21.2 PRINCIPLES OF ROOM DESIGN 745
Standing Waves 745
Bass Control 746
Audible Reflections 749
Flutter Echo 753
Reverberation 753
Diffusion 754
Imaging 756
Noise Control 758
Noise Isolation 759
Flanking 761
HVAC Noise 762
21.3 ROOMS FOR LISTENING 764
Music Practice Rooms 764
Listening Rooms 765
Screening Rooms 766
Video Post Production 767
21.4 ROOMS FOR RECORDING 768
Home Studios 768
Sound Stages 769
Scoring Stages 770
Recording Studios 771
Foley and ADR 772
21.5 ROOMS FOR MIXING 774
Dubbing Stages 774
Control Rooms 774
21.6 DESIGN DETAILS IN STUDIOS 776
Noise Isolation 776
Symmetry 777
Loudspeaker Placement 777
Bass Control 777
Studio Window Design 778
Diffusion 779

22 ACOUSTIC MODELING, RAY TRACING, AND AURALIZATION 781
22.1 ACOUSTIC MODELING 781
Testing Scale Models 782
Spark Testing 782
Ray Casting 784

Image Source Method 786
Hybrid Models 786
22.2 RAY TRACING 787
Rays 787
Surfaces and Intersections 788
Planar Surfaces 789
Ray-Plane Intersection 789
Ray-Polygon Intersections 790
Ray-Sphere Intersection 791
Ray-Cylinder Intersection 792
Ray-Quadric Intersections 793
Ray-Cone Intersection 794
Ray-Paraboloid Intersection 794
22.3 SPECULAR REFLECTION OF RAYS FROM SURFACES 795
Specular Reflections 795
Specular Reflections with Absorption 796
Specular Absorption by Seats 797
22.4 DIFFUSE REFLECTION OF RAYS FROM SURFACES 798
Measurement of the Scattering Coefficient 799
Diffuse Reflections 799
Multiple Reflections 801
Edge Effects 802
Hybrid Models and the Reverberant Tail 803
22.5 AURALIZATION 804
Convolution 804
Directional Sound Perception 807
Directional Reproduction 808

REFERENCES **813**

INDEX **829**

This page intentionally left blank

PREFACE

Architectural acoustics has been described as something of a black art or perhaps more charitably, an arcane science. While not purely an art, at its best it results in structures that are beautiful as well as functional. To produce art, however, the practitioner must first master the science of the craft before useful creativity is possible, just as a potter must learn clay or a painter his oils.

Prior to Sabine's work at the beginning of the 20th century there was little to go on. Jean Louis Charles Garnier (1825-1898), designer of the Paris Opera House, expressed his frustration at the time,

> "*I gave myself pains to master this bizarre science [of acoustics] but . . . nowhere did I find a positive rule to guide me; on the contrary, nothing but contradictory statements . . . I must explain that I have adopted no principle, that my plan has been based on no theory, and that I leave success or failure to chance alone . . . like an acrobat who closes his eyes and clings to the ropes of an ascending balloon.*" (Garnier, 1880).

Since Sabine's contributions in the early 1900's, there has been a century of technical advances. Studies funded by the EPA and HUD in the 1970's were particularly productive. Work in Canada, Europe, and Japan has also contributed greatly to the advancement of the field.

When Dick Stern first suggested this work, like Garnier one-hundred years earlier, I found, at first, few guides. There were many fine books for architects that graphically illustrate acoustic principles. There were also excellent books on noise and vibration control, theoretical acoustics, and others that are more narrowly focused on concert halls, room acoustics, and sound transmission. Many of these go deeper into aspects of the field than there is room for here, and many have been useful in the preparation of this material. Several good books are, unfortunately, out of print so where possible I have tried to include examples from them.

The goal is to present a technical overview of architectural acoustics at a level suitable for an upper division undergraduate or an introductory graduate course. The book is organized as a step-by-step progression through acoustic interactions. I have tried to include practical applications where it seemed appropriate. The algorithms are useful not only for problem solving, but also for understanding the fundamentals. I have included treatments of certain areas of audio engineering that are encountered in real-life design problems, which are not

normally found in texts on acoustics. There is also some material on computer modeling of loudspeakers and ray tracing. Too often designers accept the conclusions obtained from software models without knowing the underlying basis of the computations. Above all I hope the book will provide a intellectual framework for thinking about the subject in a logical way and be helpful to those working in the field.

ACKNOWLEDGMENTS

Many people have contributed directly and indirectly to the preparation of this book. Many authors have been generous in granting permission to quote figures from their publications and in supplying helpful comments and suggestions. Among these were Mark Alcalde, Don Allen, Michael Barron, Leo Beranek, John Bradley, Jerry Brigham, Bob Bronsdon, Howard Castrup, Bob Chanaud, John Eargle, Angelo Farina, Jean Francois Hamet, George Hessler, Russ Johnson, David Klepper, Zyun-iti Maekawa, Nelson Meacham, Shawn Murphy, Chris Peck, Jens Rindel, Thomas Rossing, Ben Sharp, Chip Smith, Dick Stern, Will and Regina Thackara, and Floyd Toole. Jean Claude Lesaca and Richard Lent prepared several of the original drawings. My secretary Pat Behne scanned in the quoted drawings and traced over them in AutoCad before I did the final versions. She also reviewed and helped correct the various drafts.

The staff of Academic Press including Zvi Ruder, Joel Stein, Shoshanna Grossman, Angela Dooley, and Simon Crump were helpful in shepherding me through the process. Dick Stern was present at the beginning and his steady hand and wise counsel were most appreciated. My wife Marilyn McAmis and our family showed great patience with the long hours required, for which I am very grateful.

Although I have tried to purge the document of errors, there are undoubtedly some that I have missed. I hope that these are few and do not cause undue confusion.

This page intentionally left blank

1 HISTORICAL INTRODUCTION

The arts of music, drama, and public discourse have both influenced and been influenced by the acoustics and architecture of their presentation environments. It is theorized that African music and dance evolved a highly complex rhythmic character rather than the melodic line of early European music due, in part, to its being performed outdoors. Wallace Clement Sabine (1868–1919), an early pioneer in architectural acoustics, felt that the development of a tonal scale in Europe rather than in Africa could be ascribed to the differences in living environment. In Europe, prehistoric tribes sought shelter in caves and later constructed increasingly large and reverberant temples and churches. Gregorian chant grew out of the acoustical characteristics of the Gothic cathedrals, and subsequently baroque music was written to accommodate the churches of the time. In the latter half of the twentieth century both theater design and performing arts became technology-driven, particularly with the invention of the electronic systems that made the film and television industries possible. With the development of computer programs capable of creating the look and sound of any environment, a work of art can now not only influence, but also define the space it occupies.

1.1 GREEK AND ROMAN PERIOD (650 BC - AD 400)

Early Cultures

The origin of music, beginning with some primeval song around an ancient campfire, is impossible to date. There is evidence (Sandars, 1968) to suggest that instruments existed as early as 13,000 BC. The understanding of music and consonance dates back at least to 3000 BC, when the Chinese philosopher Fohi wrote two monographs on the subject (Skudrzyk, 1954).

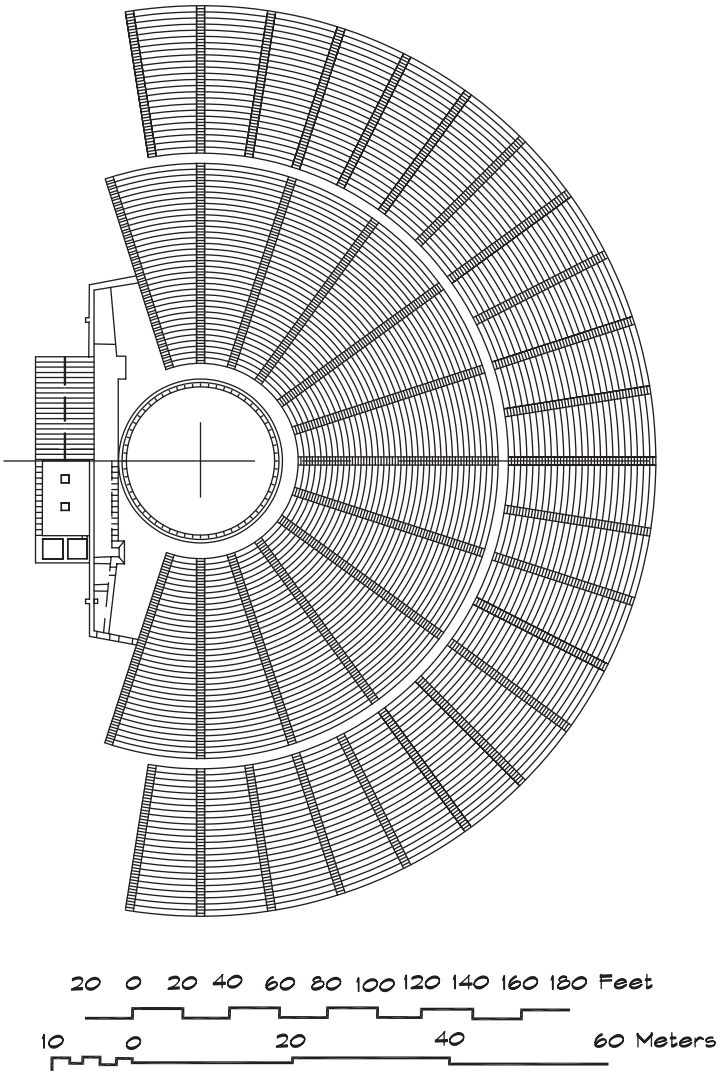
The earliest meeting places were probably no more than conveniently situated open areas. Their form was whatever existed in nature and their suitability to purpose was haphazard. As the need arose to address large groups for entertainment, military, or political purposes, it became apparent that concentric circles brought the greatest number of people close to the central area. Since the human voice is directional and intelligibility decreases as the listener moves off axis, seating arrangements were defined by the vocal polar pattern and developed naturally, as people sought locations yielding the best audibility. This led to the construction of earthen or stone steps, arranging the audience into a semicircle in

front of the speaker. The need to improve circulation and permanence evolved in time to the construction of dedicated amphitheaters on hillsides based on the same vocal patterns.

Greeks

The Greeks, perhaps due to their democratic form of government, built some of the earliest outdoor amphitheaters. The seating plan was in the shape of a segment of a circle, slightly more than 180°, often on the side of a hill facing the sea. One of the best-preserved examples of the Greco-Hellenistic theater is that built at Epidaurus in the northeastern Peloponnese in 330 BC, about the time of Aristotle. A sketch of the plan is shown in Fig. 1.1. The seating was steeply sloped in these structures, typically 2:1, which afforded good sight lines and reduced grazing attenuation. Even with these techniques, it is remarkable that this theater, which seated as many as 17,000 people, actually functioned.

FIGURE 1.1 Ancient Theater at Epidaurus, Greece (Izenour, 1977)



The ancient Greeks were aware of other acoustical principles, at least empirically. Chariot wheels in Asia Minor were heavy, whereas those of the Greeks were light since they had to operate on rocky ground. To achieve high speed, the older Asian design was modified, so that the four-spoke wheels were smaller and the wooden rims were highly stressed and made to be very flexible. They were so light that if left overnight under the weight of the chariot they would undergo deformation due to creep. Telemachus, in Homer's story of the Odyssey, tipped his vehicle vertically against a wall, while others removed their wheels in the evening (Gordon, 1978) to prevent warping. The wheels were mounted on light cantilevered shafts and the vehicle itself was very flexible, which helped isolate the rider from ground-induced vibrations.

Greek music and dance were also highly developed arts. In 250 BC at a festival to Apollo, a band of several hundred musicians played a five-movement piece celebrating Apollo's victory over Python (Rolland et al., 1948). There is strong evidence that the actors wore masks that were fitted out with small megaphones to assist in increasing the directivity of the voices. It is not surprising that the Greek orator Demosthenes (c 384–322 BC) was reputed to have practiced his diction and volume along the seashore by placing pebbles in his mouth. Intelligibility was enhanced, not only by the steeply raked seating, but also by the naturally low background noise of a preindustrial society.

The chorus in Greek plays served both as a musical ensemble, as we use the term today, and as a group to chant the spoken word. They told the story and explained the action, particularly in the earlier plays by Aeschylus (Izenour, 1977). They may have had a practical as well as a dramatic purpose, which was to increase the loudness of the spoken word through the use of multiple voices.

Our knowledge of the science of acoustics also dates from the Greeks. Although there was a general use of geometry and other branches of mathematics during the second and third millennia BC, there was no attempt to deduce these rules from first principles in a rigorous way (Dimarogonas, 1990). The origination of the scientific method of inquiry seems to have begun with the Ionian School of natural philosophy, whose leader was Thales of Miletos (640–546 BC), the first of the seven wise men of antiquity. While he is better known for his discovery of the electrical properties of amber (*electron* in Greek), he also introduced the logical proof for abstract propositions (Hunt, 1978), which led in time to the formal mathematics of geometry, based on the theorem-proof methods of Euclid (330–275 BC).

Pythagoras of Samos (c 570–497 BC), a contemporary of Buddha, Confucius, and Lao-Tse, can be considered a student of the Ionian School. He traveled to Babylon, Egypt, and probably India before establishing his own school at Crotone in southern Italy. Pythagoras is best known for the theorem that bears his name, but it was discovered much earlier in Mesopotamia. He and his followers made important contributions to number theory and to the theory of music and harmony. The word *theorii* appeared in the time of Pythagoras meaning "the beauty of knowledge" (Herodotos, c 484–425 BC). Boethius (AD 480–524), a Roman scholar writing a thousand years later, reports that Pythagoras discovered the relationship between the weights of hammers and the consonance of their natural frequencies of vibration. He is also reported to have experimented with the relationship between consonance and the natural frequencies of vibration of stretched strings, pipes, shells, and filled vessels. The Pythagorean School began the scientific exploration of harmony and acoustics through these studies. They understood the mechanisms of generation, propagation, and perception of sound (Dimarogonas, 1990). Boethius describes their knowledge of sound in terms of waves generated by a stone falling into a pool of water. They probably realized that sound was a

wave propagating through the air and may have had a notion of the compressibility of air during sound propagation.

Aristotle (384–322 BC) recognized the need for a conducting medium and stated that the means of propagation depended on the properties of the material. There was some confusion concerning the relationship between sound velocity and frequency, which was clarified by Theophrastos of Eresos (370–285 BC): "The high note does not differ in speed, for if it did it would reach the hearing sooner, and there would be no concord. If there is concord, both notes must have the same speed." The first monograph on the subject, *On Acoustics*, is attributed to Aristotle, although it may have been written by his followers. Whoever wrote it, the author had a clear understanding of the relationship between vibration and sound: "bodies that are capable of vibrating produce sounds . . . strings are examples of such bodies."

Romans

The Roman and the late Hellenistic amphitheaters followed the earlier Greek seating pattern, but limited the seating arc to 180°. They also added a stagehouse (*skene*) behind the actors, a raised acting area (*proskenion*), and hung awnings (*valeria*) overhead to shade the patrons. The chorus spoke from a hard-surfaced circle (*orchestra*) at the center of the audience. A rendering of the Roman theater at Aspendius, Turkey is shown in Fig. 1.2. The Romans were better engineers than the early Greeks and, due to their development of the arch and the vault, were not limited to building these structures on the natural hillsides.

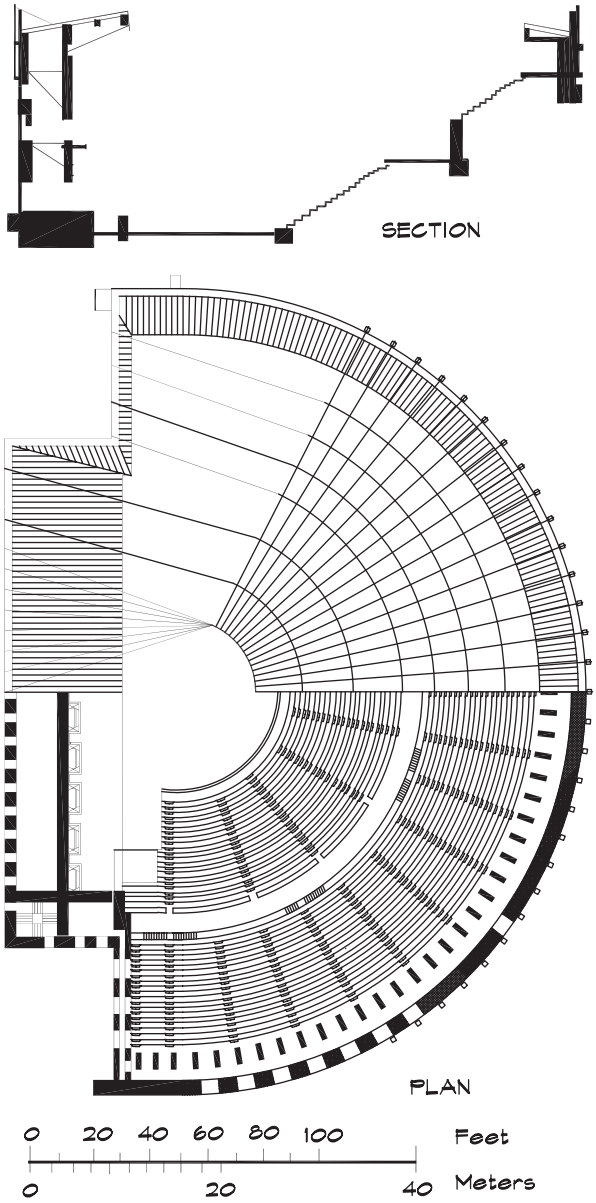
The most impressive of the Roman amphitheaters, the Flavian amphitheater was built between AD 70 and 81, and was later called the Colosseum, due to its proximity to a colossal statue of Nero. With a total seating capacity of about 40,000 it is, except for the Circus Maximus and the Hippodrome (both racecourses), the largest structure for audience seating of the ancient world (Izenour, 1977). Its architect is unknown, but his work was superb. The sightlines are excellent from any seat and the circulation design is still used in modern stadia. The floor of the arena was covered with sand and the featured events were generally combats between humans, or between humans and animals. This type of spectacle was one of the few that did not require a high degree of speech intelligibility for its appreciation by the audience. The floor was sometimes caulked and filled with water to a depth of about a meter for mock sea battles.

Smaller indoor theaters also became a part of the Greek and Roman culture. These more intimate theaters, called *odea*, date from the age of Pericles (450 BC) in Greece. Few remain, perhaps due to their wood roof construction. The later Greek playwrights, particularly Sophocles and Euripides, depended less on the chorus and more on the dialogue between actors to carry the meaning of the play, particularly in the late comedies. These dramatic forms developed either because of the smaller venues or to accommodate the changing styles.

In the Roman theater the chorus only came out at intermission so the *orchestra* shrunk to a semicircle with seats around it for the magistrates and senators. The front wall or *scaena* extended out to the edges of the semicircle of seats and was the same height as the back of the seating area. It formed a permanent backdrop for the actors with a palace decor. The *proskenium* had a curtain, which was lowered at the beginning of the performance and raised at the end. (Breton, 1989)

The Odeon of Agrippa, a structure built in Athens in Roman times (12 BC), was a remarkable building. Shown in Fig. 1.3, it had a wood-trussed clear span of over 25 meters (83 feet). It finally collapsed in the middle of the second century. Izenour (1977) points out

FIGURE 1.2 Roman Theater at Aspendus, Turkey (Izenour, 1977)

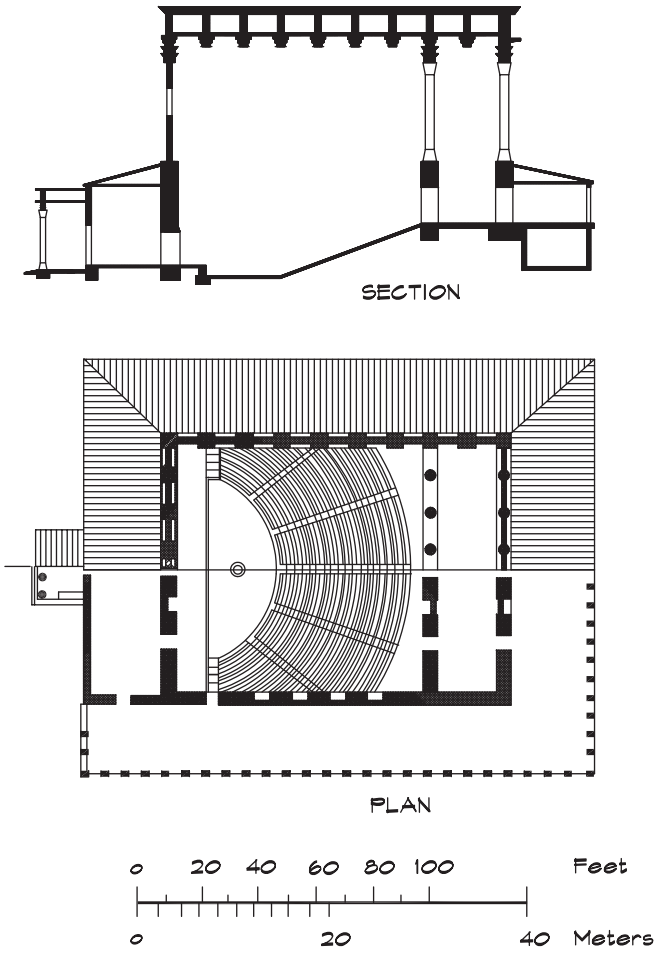


that these structures, which ranged in size from 200 to 1500 seats, are found in many of the ancient Greek cities. He speculates that, "during the decline of the Empire these roofed theaters, like the small noncommercial theaters of our time, became the final bastion of the performing arts, where the more subtle and refined stage pieces—classical tragedy and comedy, ode and epoch—were performed, the latter to the accompaniment of music (lyre, harp, double flute and oboe) hence the name odeum, 'place of the ode'."

Vitruvius Pollio

Much of our knowledge of Roman architecture comes from the writings of Vitruvius Pollio, a working architect of the time, who authored *De Architectura*. Dating from around 27 BC,

FIGURE 1.3 Odeon of Agrippa at Athens, Greece (Izenour, 1977)

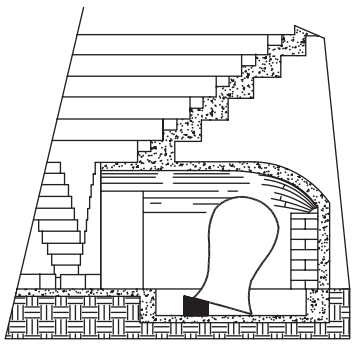


this book describes his views on many aspects of architecture, including theater design and acoustics. Some of his ideas were quite practical—such as his admonition to locate theaters on a "healthy" site with adequate ventilation (away from swamps and marshes). Seating should not face south, causing the audience to look into the sun. Unrestricted sightlines were considered particularly important, and he recommended that the edge of each row should fall on a straight line from the first to the last seat. His purpose was to assure good speech intelligibility as well as good sightlines.

Vitruvius also added one of the great historical mysteries to the acoustical literature. He wrote that theaters should have large overturned amphora or sounding vases placed at regular intervals around the space to improve the acoustics. These were to be centered in cavities on small, 150 mm (6") high wedges so that the open mouth of the vase was exposed to the stage, as shown in a conjectural restoration by Izenour in Fig. 1.4, based on an excavation of a Roman theater at Beth Shean in Israel. The purpose, and indeed the existence of these vases, remains unclear. Even Vitruvius could not cite an example of their use, though he assures us that they existed in the provinces.

FIGURE 1.4 Hypothetical Sounding Vases (Izenour, 1977)

A conjectural restoration in section of sounding vases in a cavity found at a Roman theater at Beth Shean, Israel.



1.2 EARLY CHRISTIAN PERIOD (AD 400–800)

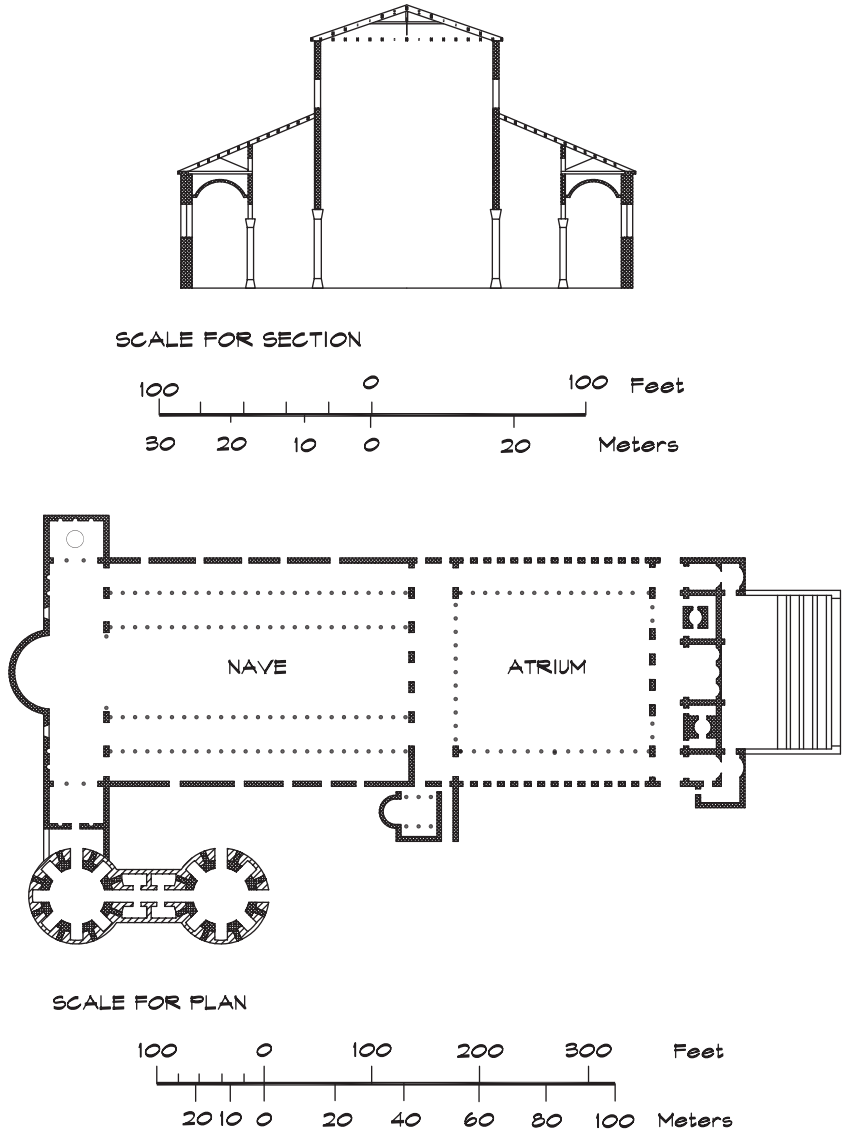
Rome and the West

The early Christian period is dated from the Roman emperor Constantine to the coronation of Charlemagne in 800. Following the official sanction of Christianity by Constantine in 326 and his relocation from Rome to Byzantium in 330, later renamed Constantinople, the age was increasingly dominated by the church, which provided the structural framework of everyday life as the Roman and then the Byzantine empires slowly decayed. Incursions by the Huns in 376 were followed by other serious invasions. On the last day of December in the winter of 406, the Rhine river froze solid, forming a bridge between Roman-controlled Gaul and the land of the Germanic tribes to the east (Cahill, 1995). Across the ice came hundreds of thousands of hungry Germans, who poured out of the eastern forests onto the fertile plains of Gaul. Within a few years, after various barbarian armies had taken North Africa and large parts of Spain and Gaul, Rome itself was sacked by Alaric in 410.

In these difficult times, monasteries became places of refuge, which housed small self-sustaining communities—repositories of knowledge, where farming, husbandry, and scholarship were developed and preserved. These were generally left unmolested by their rough neighbors, who seemed to hold them in religious awe (Palmer, 1961). In time, the ablest inhabitants of the Empire became servants of the Church rather than the state and "gave their loyalty to their faith rather than their government" (Strayer, 1955). "Religious conviction did not reinforce patriotism and men who would have died rather than renounce Christianity accepted the rule of conquering barbarian kings without protest." Under the new rulers a Romano-Teutonic civilization arose in the west, which eventually led to a division of the land into the states and nationalities that exist today.

After the acceptance of Christianity, church construction began almost immediately in Rome, with the basilican church of St. Peter in 330 initiated by Constantine himself. The style, shown in Fig. 1.5, was an amalgam of the Roman basilica (hall of justice) and the Romanesque style that was to follow. The basic design became quite popular—there were 31 basilican churches in Rome alone. It consisted of a high central nave with two parallel aisles on either side separated by colonnades supporting the upper walls and low-pitched roof, culminating in an apse and preceded by an atrium or forecourt (Fletcher, 1963). The builders generally scavenged columns from older Roman buildings that they could not

FIGURE 1.5 Basilican Church of St. Peter, Rome, Italy (Fletcher, 1963)

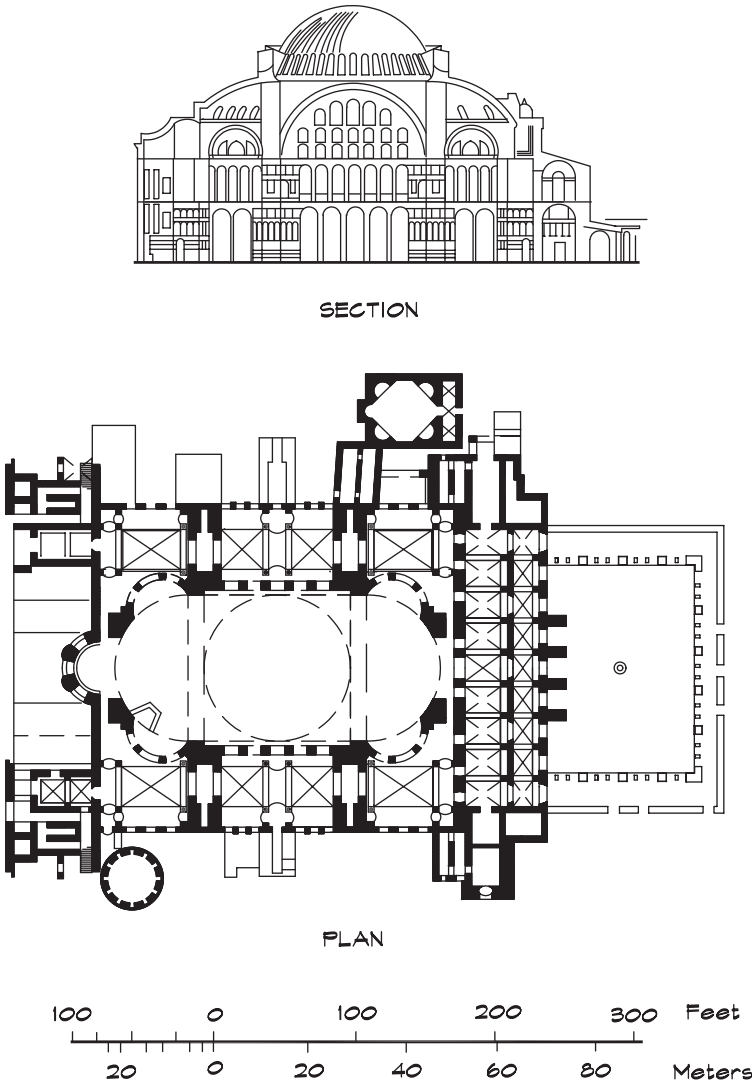


match or maintain, and which had therefore fallen into decay. The basilica style became a model for later church construction throughout Western Europe, eventually leading to the Gothic cathedrals.

Eastern Roman Empire

In the Eastern Roman Empire the defining architectural feature was the domed roof, used to cover square or polygonal floor plans. This form was combined with classical Greek columns supporting the upper walls with a series of round arches. The primary construction material was a flat brick, although marble was used as a decorative facade. The best known building of the time was St. Sophia (532–537) (*Hagia Sophia*, or divine wisdom) in Constantinople. This massive church, still one of the largest religious structures in the world, was built for

FIGURE 1.6 St. Sophia, Constantinople, Turkey (Fletcher, 1963)



Emperor Justinian by the architects Anthemius of Tralles and Isodorus of Miletus between 532 and 537. Its enormous dome, spanning 33 meters (107 feet) in diameter, is set in the center of a 76 meter (250 foot) long central nave. St. Sophia, shown in Fig. 1.6, was the masterpiece of Byzantine architecture and later, following the Turkish capture of the city in 1453, became the model for many of the great mosques.

In the sixth century the territory of the former Roman Empire continued to divide. The Mediterranean world during this period was separated into three general regions: 1) the Byzantine empire centered in Asia minor, which controlled the Balkans, Greece, and eventually expanded into Russia; 2) the Arab world of Syria, Egypt, and North Africa, which under the leadership of Mohammed (570–632) swept across Africa and into southern Italy, Sicily, and Spain; and 3) the poorest of the three, Western Europe, an agricultural backwater with basically a subsistence economy. Holding the old empire together proved to be more than the Byzantine emperors could afford. Even the reign of the cautious Justinian (527–565),

whose generals temporarily recaptured Italy from the Ostrogoths, North Africa from the Vandals, and southeastern Spain from the Visigoths, did so on the backs of heavy taxation and loss of eastern provinces. The Lombards soon recaptured much of Italy, but the Byzantine representatives managed to hang onto Rome and the neighboring areas. The troubled sixth century closed with the successful pontificate of Pope Gregory I, who strove to standardize the liturgy and is traditionally regarded as the formulator of the liturgical chant, which bears his name.

Gregorian chant or plainsong, which became part of the liturgy in the Western Church, had antecedents in the rich tradition of cantillation in the Jewish synagogues, as well as the practices in the Eastern Church. Plain chant combined the simple melody and rhythm that dominated church music for several centuries.

Until a common system of musical notation was developed in the ninth century, there was little uniformity or record of the music. The early basilican churches were highly reverberant, even with open windows, and the pace and form of church music had to adjust to the architecture to be understood. Even with a simple monodic line, the blending of sounds from chants in these reverberant spaces is hauntingly beautiful.

The eastern and western branches of the Christian church became divided by ideological differences that had been suppressed when the church was clandestine. An iconoclastic movement resulted from a decree from the eastern emperor, Leo III (717–741), forbidding any representation of human or animal form in the church. Subsequently many Greek artisans left Constantinople for Italy, where they could continue their professions under Pope Gregory II. This artistic diaspora caused Leo to relent somewhat and he allowed painted figures on the walls of eastern churches but continued the prohibition of sculpture. His decrees led, in part, to the Byzantine style—devoid of statuary, and unchanging in doctrine and ritual. In contrast, the western church embraced statuary and sculpture, which in time begot the highly ornamented forms of the Baroque period and the music that followed. The split between the eastern and western branches, which had begun in the ninth century with a theological argument over the nature of the divine spirit, finally ended with a formal schism in 1054 when the two churches solemnly excommunicated each other.

1.3 ROMANESQUE PERIOD (800–1100)

The Romanesque period roughly falls between the reign of Charlemagne and the era of the Gothic cathedrals of the twelfth century. In the year 800 it was rare to find an educated layman outside of Italy (Strayer, 1955). The use of Latin decreased and languages fragmented according to region as the influence of a central authority waned. The feudal system developed in its place, not as a formal structure based on an abstract theory of government, but as an improvisation to meet the incessant demands of the common defense against raiders.

The influence of both Roman and Byzantine traditions is evident in the architecture of the Romanesque period. From the Roman style, structures retained much of the form of the basilica; however, the floor plans began to take on the cruciform shape. The eastern influence entered the west primarily through the great trading cities of Venice, Ravenna, and Marseilles and appeared in these cities first. Romanesque style is characterized by rounded arches and domed ceilings that developed from the spherical shape of the east into vaulted structures in the west. The narrow upper windows, used in Italy to limit sunlight, lead to larger openings in the north to allow in the light, and the flat roofs of the south were sharpened in the north to throw off rain and snow. Romanesque structures remained massive until the

introduction of buttresses, which allowed the walls to be lightened. Construction materials were brick and stone and pottery, as well as materials scavenged from the Roman ruins. The exquisite marble craftsmanship characteristic of the finest Greek and Roman buildings had been lost and these medieval brick structures seemed rough and plain compared with the highly ornamented earlier work.

One notable exception was St. Mark's Cathedral in Venice. It was built on the site of the basilica church, originally constructed to house the remains of St. Mark in 864. The first church burned in 976 and was rebuilt between 1042 and 1085. It was modeled after the Church of the Apostles in Constantinople as a classic Romanesque structure in a nearly square cruciform shape, with rounded domes reminiscent of later Russian orthodox churches. St. Mark's, illustrated in Fig. 1.7, was later home to a series of brilliant composers including Willaert (1480–1562), Gabrielli (1557–1612), and Monteverdi (1567–1643).

The music, which we now associate with Gregorian chant, developed as part of the worship in the eighth and ninth centuries. The *organum*, a chant of two parts, grew slowly from the earlier monodic music. At first this form consisted of a melody that was sung (held) by a tenor (*tenere*, to hold) while another singer had the same melodic line at an interval a forth above. True polyphony did not develop until the eleventh century.

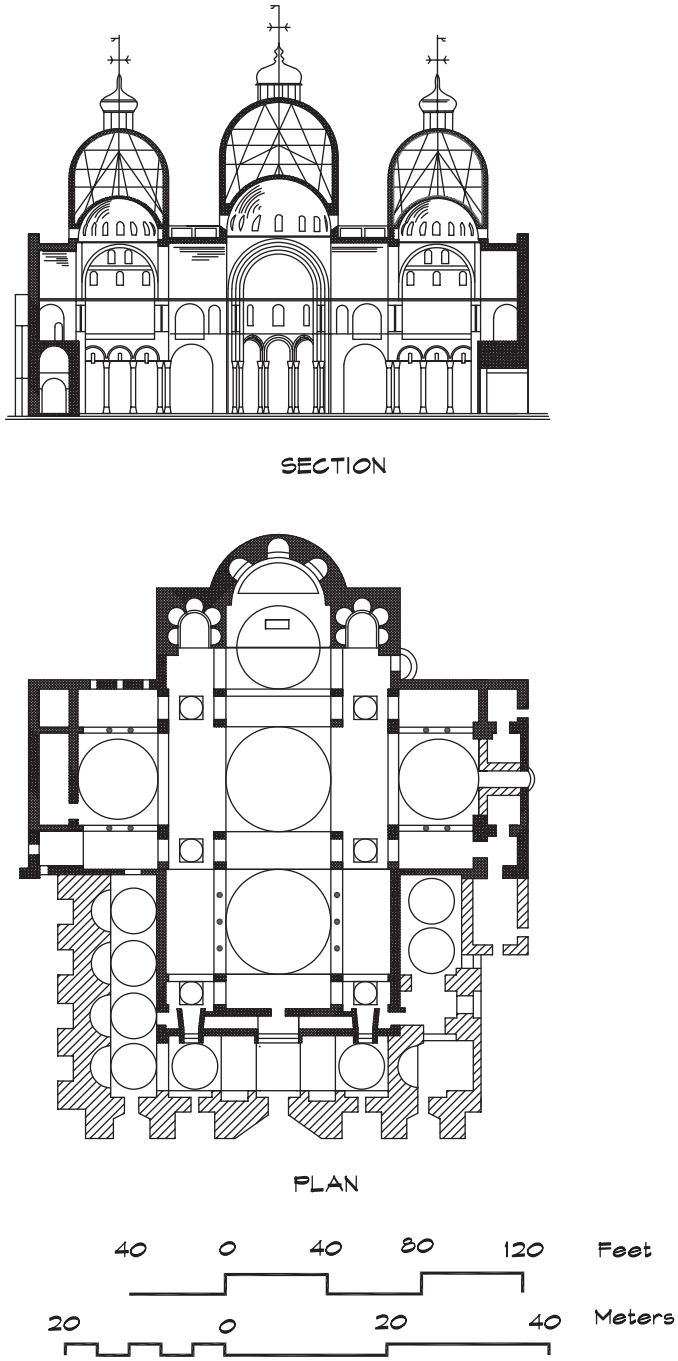
1.4 GOTHIC PERIOD (1100–1400)

Gothic Cathedrals

Beginning in the late middle ages, around 1100, there was a burst in the construction of very large churches, the Gothic cathedrals, first in northern France and later spreading throughout Europe. These massive structures served as focal points for worship and repositories for the religious relics that, following the return of the crusaders from the holy lands, became important centers of the valuable pilgrim trade. The cathedrals were by and large a product of the laity, who had developed from a populace that once had only observed the religious forms, to one that held beliefs as a matter of personal conviction. Successful cities had grown prosperous with trade and during the relatively peaceful period of the late middle ages the citizens enthusiastically supported their construction. The first was built by Abbot Suger at St. Denis near Paris between 1137 and 1144 and was made possible by the hundreds of experiments in the building of fortified towns and churches, which had produced a skilled and knowledgeable work force. Suger was a gifted administrator and diplomat who also had the good fortune to attend school and become best friends with the young prince who became King Louis VI. When the king left on the Second Crusade he appointed Suger regent and left him in charge of the government. Following the success of St. Denis, other cathedrals were soon begun at Notre Dame (1163–c1250), Bourges (1192–1275), Chartes (1194–1260), and Rheims (1211–1290). These spectacular structures (see Fig. 1.8) carried the art and engineering of working in stone to its highest level. The vaulted naves, over 30 meters (100 feet) high, were lightened with windows and open colonnades and supported from the exterior with spidery flying buttresses, which gave the inside an ethereal beauty. Plain chant was the music of the religious orders and was suited perfectly to the cathedral. Singing was something that angels did, a way of growing closer to God. It was part of the every day religious life, done for the participants rather than for an outside listener.

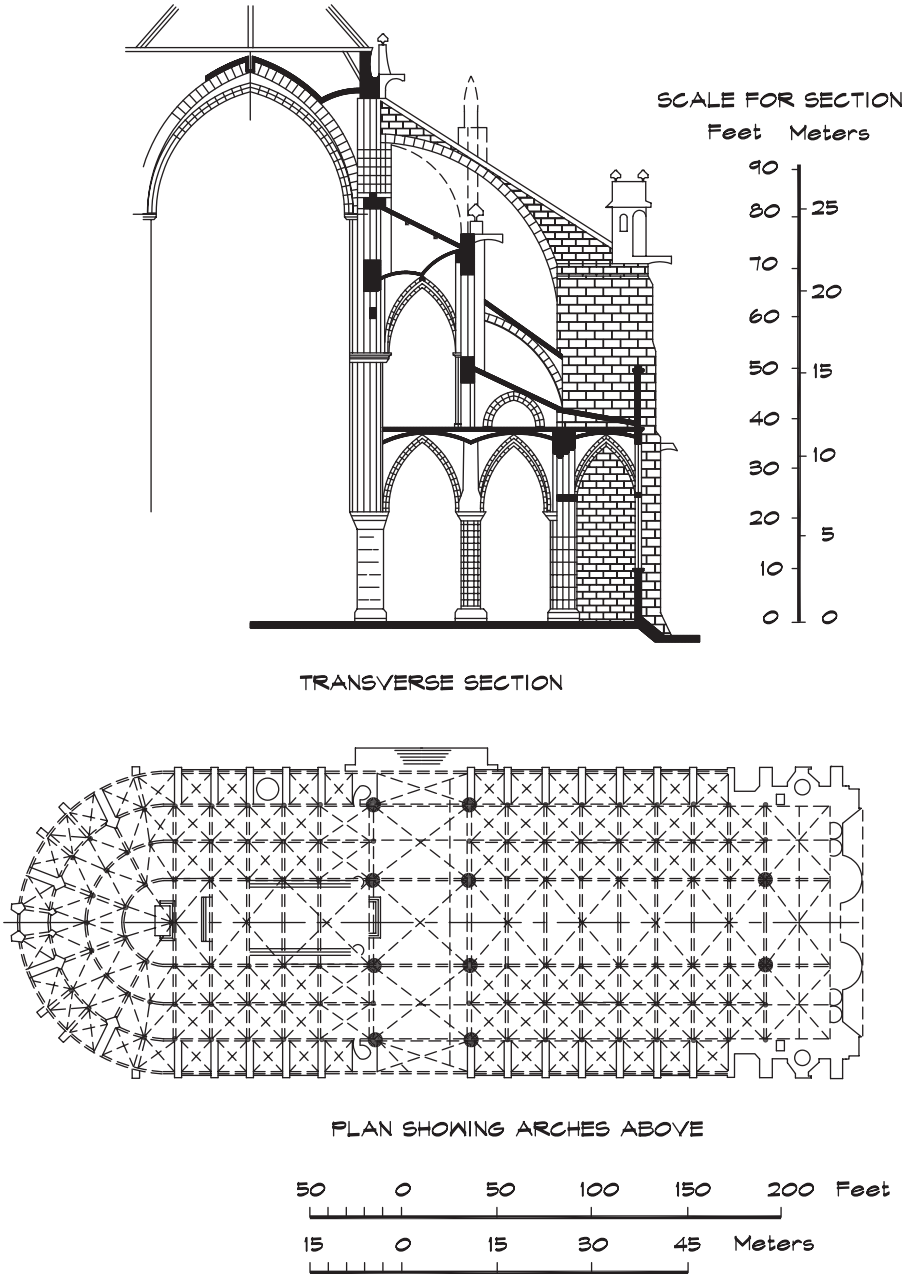
In the second half of the twelfth century the beginnings of polyphony developed in the School of Notre Dame in Paris from its antecedents in the great abbey of St. Martial in Limoges. The transition began with the two-part *organum* of Leonin, and continued with the

FIGURE 1.7 St. Mark's Cathedral, Venice, Italy (Fletcher, 1963)



three and four-part *organum* of his successor Perotin. The compositions were appropriate for the large reverberant cathedrals under construction. A slowly changing plainsong pedal note was elaborated by upper voices, which did not follow the main melody note for note as before. This eventually led, in the thirteenth and fourteenth centuries, to the polyphonic motets in which different parts might also have differing rhythms. Progress in the development of

FIGURE 1.8 Notre Dame Cathedral, Paris, France (Fletcher, 1963)



serious music was laborious and slow. Outside the structured confines of church music, the secular troubadours of Provence, the trouveres of northern France and southern England, the story-telling jongleurs among the peasantry, and the minnesingers in Germany also made valuable contributions to the art.

The influence of the Church stood at its zenith in the thirteenth century. The crusades, of marginal significance militarily, had served to unite Western Europe into a single religious community. An army had pushed the Muslims nearly out of the Iberian peninsula.

Beginning in the fourteenth century, however, much of the civilized world was beset by the ravages of the bubonic plague. Between the years 1347 and 1350 it wiped out at least one third of the population. The Church was hit harder than the general populace, losing more than half its members. Many men, largely illiterate, had lost their wives to the plague and sought to join the religious orders. Lured by offers of money from villages that had no priest, others came to the church for financial security. Money flowed into Rome and supported a growing bureaucracy and opulence, which ultimately led to the Reformation. This worsened a problem already confronting the religious leadership, "the danger of believing that the institution exists for the benefit of those who conduct its affairs." (Palmer, 1961)

With the rise of towns and commerce, public entertainment became more secular and less religious in its focus. Theater in the late middle ages was tolerated by the Church largely because it had been co-opted as a religious teaching aid. Early plays, dating from the tenth century, were little more than skits based on scripture, which were performed in the streets by troupes. These evolved, in the thirteenth and fourteenth centuries, into the miracle and mystery plays that combined singing and spoken dialogue. The language of the early medieval theater was Latin, which few understood. This changed in time to the local vernacular or to a combination of Latin and vernacular. The plays evolved from a strictly pedagogical tool to one that contained more entertainment. As the miracle plays developed, they were performed in rooms that would support the dialogue and make it understandable. By 1400, the pretext of the play remained religious, but the theater was already profane (Hindley, 1965).

1.5 RENAISSANCE PERIOD (1400–1600)

Renaissance Churches

The great outpouring of art, commerce, and discovery that was later described as the Renaissance or rebirth, first started in northern Italy and gradually spread to the rest of Europe. The development of new music during these years was rich and profuse. Thousands of pieces were composed and, while sacred music still dominated, secular music also thrived (Hemming, 1988).

Church construction still continued to flourish in the early years of the Renaissance. St. Peter's Cathedral in Rome, the most important building of the period, was begun in 1506 and was created by many of the finest architects and artists of the day. A competition produced a number of designs, still preserved in the Uffizi Gallery in Florence, from which Bramante (1444–1514) was selected as architect (Fletcher, 1963). After the death of Pope Julius II a number of other architects, including Raphael (1483–1520), worked on the project—the best known being Michelangelo (1475–1564). He began the construction of the dome, which was completed after his death, from his models. Some time later Bernini erected (1655–1667) the immense piazza and the baroque throne of St. Peter.

The construction of this great cathedral in Rome also reached out to touch an obscure professor of religion at the university in Whitenberg. In 1517, a friar named Tetzel was traveling through Germany selling indulgences to help finance it. Martin Luther felt that the people were being deluded by this practice and, in the manner of the day, posted a list of 95 theses on the door of the castle church in protest (Palmer, 1961). By 1560, most of northern Europe including Germany, England, Netherlands, and the Scandinavian countries had officially adopted some form of Protestantism.

Renaissance Theaters

Theater construction began again in Italy in the early Renaissance, more or less where the Romans had left it a thousand years earlier. In 1580, the Olympic Academy in Vicenza engaged Palladio (1518–1580) to build a permanent theater (Fig. 1.9), the first since the Roman Odeons. The seating plan was semi-elliptical, following the classical pattern, and the stage had much the same *orchestra* and *proskenium* configuration that the old Roman theaters had. Around the back of the audience was a portico of columns with statues above. The newly discovered art of perspective captured the imagination of designers and they crafted stages, which incorporated a rising stage floor and single point perspective. The terms upstage and downstage evolved from this early design practice. After the death of Palladio, his pupil Scamozzi added five painted streets in forced perspective angling back from the *scaena*.

In 1588, Scamozzi further modified the Roman plan in a new theater, the Sabbioneta. The semi-elliptical seating plan was pushed back into a U shape, the stage wall was removed, and a single-point perspective backdrop replaced the earlier multiple-point perspectives. This theater is illustrated in Fig. 1.10. Its seating capacity was small and there was little acoustical support from reflections off the beamed ceiling.

In mid-sixteenth century England, traveling companies of players would lay out boards to cover the muddy courtyards of inns, while the audience would stand around them or line the galleries that flanked the main yard (Breton, 1989). Following the first permanent theater built in 1576 by James Burbage, this style became the model for many public theaters, including Shakespeare's Globe. The galleries surrounding the central court were three tiers high with a roofed stage, which looked like a thatched apron at one end. Performances were held during the day without a curtain or painted backdrop. The acoustics of these early theaters was probably adequate. The side walls provided beneficial early reflections and the galleries yielded excellent sightlines. The open-air courtyard reduced reverberation problems and outside noise was shielded by the high walls. It is remarkable that such simple structures sufficed for the work of a genius like Shakespeare. Without good speech intelligibility provided by this type of construction, the complex dialogue in his plays would not only have been lost on the audience, it would probably not have been attempted at all.

FIGURE 1.9 Teatro Olimpico, Vicenza, Italy (Breton, 1989)

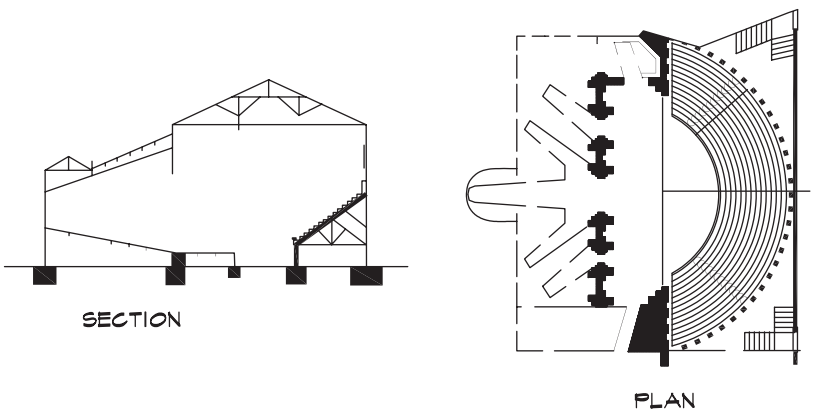
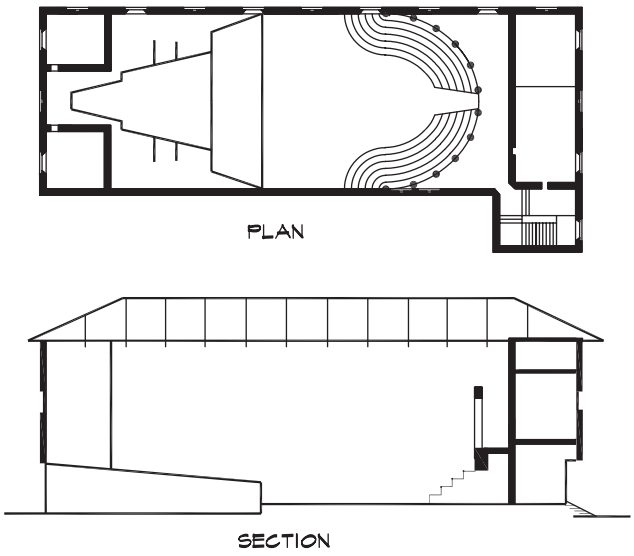


FIGURE 1.10 Sabbioneta Theater, Italy (Breton, 1989)



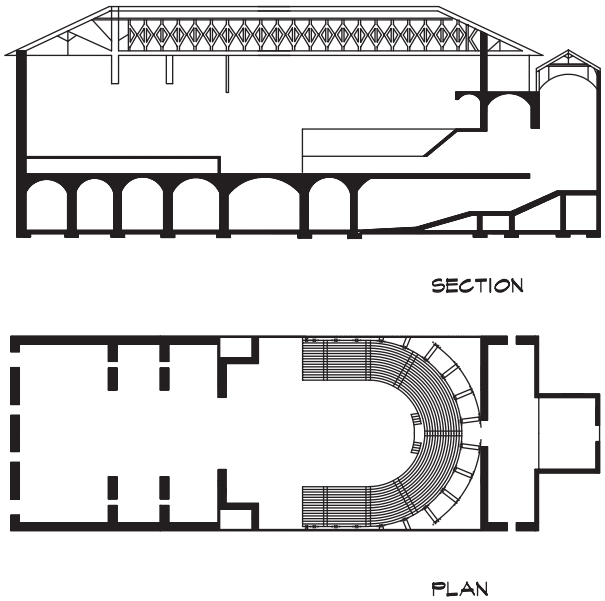
1.6 BAROQUE PERIOD (1600–1750)

Baroque Churches

The first half of the seventeenth century was dominated by the Thirty Years War (1618–1648), which ravaged the lands of Germany and central Europe. This confusing struggle was one of shifting alliances that were formed across religious and political boundaries (Hindley, 1965). The end result was a weakening of the Hapsburg empire and the rise of France as the dominant power in Europe. Italy became a center for art and music during that period, in large part because it was relatively unscathed by these central European wars. In northern Italy a style, which became known as the Baroque (after the Portuguese *barocco*, a term meaning a distorted pearl of irregular shape), grew out of the work of a group of Florentine scholars and musicians known as the *Camerata* (from the Italian *camera*, or chamber). This group abandoned the vocal polyphony of Renaissance sacred music and developed a new style featuring a solo singer with single instrumental accompaniment (the *continuo*) to provide unobtrusive background support for the melodic line. The new music was secular rather than sacred and dramatic, and passionate rather than ceremonial (Hemming, 1988), and allowed for considerably more freedom by the performer.

Both the music and the architecture of the Baroque period was more highly ornamented than that of the Renaissance. Composers began writing in more complicated musical forms such as the fugue, chaconne, passacaglia, toccata, concerto, sonata, and oratorio. Some of the vocal forms, such as the cantata, oratorio, and opera, grew out of the work of the Camerata. Others developed from the architecture and influence of a particular space. St. Mark's Cathedral in Venice was shaped like a nearly square cross with individual domes over each arm and above the center (see Fig. 1.7). These created localized reverberant fields, which supported the widely separated placement of two or three ensembles of voices and instruments that could perform as separate musical bodies. Gabrielli (1557–1612), who was organist there for 27 years, exploited these effects in his compositions, including separate

FIGURE 1.11 Theatro Farnese, Parma, Italy (Breton, 1989)



instrument placement, call and response sequences, and echo effects. In less than 100 years this style had been transformed into the concerto grosso (Burkat, 1998).

Baroque Theaters

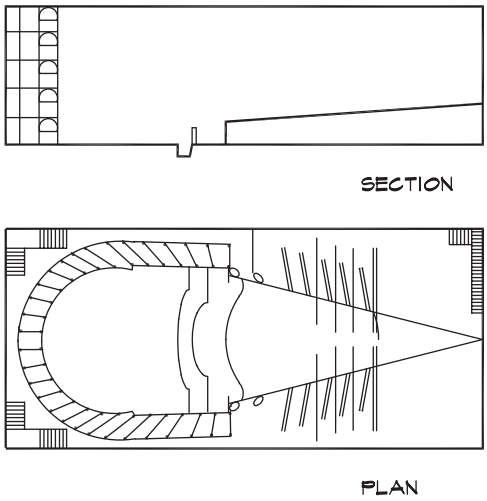
The progress in theater construction in Northern Italy was also quite rapid. The illusion stages gave way to auditoria with horizontally sliding flats, and subsequently to moveable stage machinery. The *Theatro Farnese* in Parma, constructed between 1618 and 1628 by Giovanni Battista Aleotti, had many features of a modern theater. Shown in Fig. 1.11, it featured horizontal set pieces, which required protruding side walls on either side of the stage opening to conceal them. This allowed set changes to be made and provided entrance spaces on the side wings for the actors to use without appearing out of scale. The U-shaped seating arrangement afforded the patrons a view, not only of the stage, but also of the prince, whose box was located on the centerline.

In Florence at the Medici court, operas were beginning to be written. The first one was *Dafne*, which is now lost, written between 1594 and 1598 by Peri (Forsyth, 1985). The first known opera performance was Peri's *Euridice*, staged at a large theater in the Pitti Palace to celebrate the wedding of Maria de'Medici and King Henri IV of France in 1600. This was followed by Monteverdi's *Orfeo*, first performed in 1607 in Mantua, which transformed opera from a somewhat dry and academic style to a vigorous lyric drama.

Italian Opera Houses

By 1637, when the first public opera house was built in Venice (Fig. 1.12), the operatic theater had become the multistory U-shaped seating arrangement of the Theatro Farnese, with boxes in place of tiers. Later the seating layout further evolved from a U shape into a truncated elliptical shape. The orchestra, which had first been located at the rear of the stage and then in the side balconies, was finally housed beneath the stage as is the practice today

FIGURE 1.12 Theater of SS. Giovanni e Paolo, Venice, Italy (Forsyth, 1985)



(Breton, 1989). The stage had widened further and now had a flyloft with winches and levers to manipulate the scenery. This became the typical Baroque Italian opera house, which was the standard model replicated throughout Europe with little variation for 200 years.

Italy immediately became the center of opera in Europe. In the years between 1637 and the end of the century, 388 operas were produced in Venice alone. Nine new opera houses were opened during this period, and after 1650, never fewer than four were in simultaneous operation (Grout, 1996). These early opera houses served as public gathering places. For the equivalent of about 50 cents, the public could gain entry to the main floor, occupied by standing patrons who talked and moved about during the performances. The high background noise is documented in many complaints in writings of the time. It led to the practice of loudly sounding a cadential chord to alert the audience of an impending aria. In a forerunner of contemporary films, special effects became particularly popular. As the backstage equipment grew more complicated and the effects more extravagant, the noise of the machines threatened to drown out the singing. Composers would compensate by writing instrumental music to mask the background sounds. The popularity of these operas was so great that the better singers were in considerable demand. Pieces were written to emphasize the lead singer's particular ability with the supporting roles de-emphasized.

Baroque Music

The seventeenth century also saw the rise of the aristocracy and with it, conspicuous consumption. Churches and other public buildings became more ornate with applied decorative elements, which came to symbolize the Baroque style. Music began to be incorporated into church services in the form of the oratorio, a sort of religious opera staged without scenery or costumes. In Rome the Italian courts were opulent enough to embrace opera as a true spectacle. Pope Urban VII commissioned the famous Barberini theater based on a design of Bernini, which held 3000 people and opened in 1632 with a religious opera by Landi.

In the Baroque era instrumental music achieved a status equal to vocal music. Musical instruments became highly sophisticated in the seventeenth and eighteenth centuries and in some cases achieved a degree of perfection in their manufacture that is unmatched today.

The harpsichord and the instruments of the violin family became the basic group for ensemble music. Violins fashioned by craftsmen such as Nicolo Amati (1569–1684), Giuseppi Guarneri (1681–1742) and Antonio Stradivari (1644–1737) are still the best instruments ever made. The lute, which was quite popular at the beginning of the period, was rarely used at the end. Early wind instruments had been mainly shawms (later oboes), curtals (later bassoons), crumhorns, bagpipes, fifes and drums, cornets, and trumpets. New instruments were developed, specifically the recorder, the transverse flute, oboe, and bassoon. The hunting horn having a five-and-one-half-foot tube wound into four or five loops before flaring into a bell, was improved in France by reducing the number of loops and enlarging the bell. When it became known in England, it was given the name French horn. By the early 1600s, the pipe organ had developed into an instrument of considerable technical development.

Antonio Vivaldi (1678–1741), now recognized as one of the foremost Baroque composers, first learned violin from his father, who was a violinist at St. Mark's in Venice. He was a priest and later (1709) music director at a school for foundling girls, the Seminario dell'Ospitale della Pieta. His intricate compositions for the violin and other instruments of the time feature highly detailed passages characteristic of what is now known as chamber music, written for small rooms or salons.

Protestant Music

In Protestant northern Europe the spoken word was more important to the religious service than in the Catholic south. The volume of the northern church buildings was reduced to provide greater clarity of speech. The position of the pulpit was centrally placed and galleries were added to the naves and aisles. Many existing churches, including Thomaskirche in Leipzig, were modified by adding hanging drapes and additional seating closer to the pulpit (Forsyth, 1985). Johann S. Bach (1685–1750) was named cantor there in 1722, to the disappointment of the church governors. He was their second choice behind Georg Philip Telleman (1681–1767). Bach was influenced by the low reverberation time of the church, which has been estimated to have been about 1.6 seconds (Bagenal, 1930). His B-Minor Mass and the St. Matthew Passion were both composed for this space.

Bach wrote music for reverberant spaces as well as for intimate rooms. During his early years in Weimar (1703–1717) he composed mostly religious music including some of his most renowned works for organ, the Passacaglia and double Fugue in C minor and the Toccata and Fugue in D minor. His Brandenberg Concertos, composed for the orchestra at the little court of Anhalt-Cothen, were clearly meant to be played in a chamber setting, as were the famous keyboard exercises known as the *Well Tempered Clavier*, which were written for each of the 24 keys in the system of equal-tempered tuning, completed about the same time.

Baroque music was performed in salons, drawing rooms, and ballrooms, as well as in churches. In general the former were not specifically constructed for music and tended to be small. The orchestras were also on the smallish side, around twenty-five musicians, much like chamber orchestras today. As rooms and audiences grew larger, louder instruments became more popular. The harpsichord gave way to the piano, the viola da gamba to the cello, and the viol to the violin. The problem of distributing the sound evenly to the listener was soon recognized, but there were few useful guidelines. In England Thomas Mace published (1676) suggestions for the designer in his *Musick's Monument or a Rememberancer of the best practical Musick*. He recommended a square room with galleries on all sides surrounding the musicians, much like a theater in the round. Mace advocated piping the sound from the

musicians to the rear seats through tubes beneath the floor, a device that was used extensively in the Italian opera houses of the day, and contemporaneously in loud-speaking trumpets, which were employed as both listening and speaking devices (Forsyth, 1985).

1.7 ORIGINS OF SOUND THEORY

The understanding of the theory of fluids including sound propagation through them made little progress from the Greeks to the Renaissance. Roman engineers did not have a strong theoretical basis for their work in hydraulics (Guillen, 1995). They knew that water flowed downhill and would rise to seek its own level. This knowledge, along with their extraordinary skills in structural engineering, was sufficient for them to construct the massive aqueduct systems including rudimentary siphons. However, due to the difficulty they had in building air-tight pipes it was more effective for them to bridge across valleys than to try to siphon water up from the valley floors. Not until Leonardo da Vinci (1452–1519) studied the motion and behavior of rivers did he notice that, "A river of uniform depth will have more rapid flow at the narrower section than at the wider." This is what we now call the equation of continuity, one of the relationships necessary for the derivation of the wave equation.

Galileo Galilei (1564–1642) along with others noted the isochronism of the pendulum and was aware, as was the French Franciscan friar Marin Mersenne (1588–1648), of the relationship between the frequency of a stretched string and its length, tension, and density. Earlier Giovanni Battista Benedetti (1530–1590) had related the ratio of pitches to the ratio of the frequencies of vibrating objects. In England Robert Hooke (1635–1703), who had bullied a young Isaac Newton (1642–1727) on his theory of light (Guillen, 1995), published in 1675 the law of elasticity that now bears his name, in the form of a Latin anagram CEII-INOSSSTTUV, which decoded is "ut tensio sic vis" (Lindsay, 1966). It established the direct relationship between stress and strain that is the basis for the formulas of linear acoustics. The first serious attempt to formalize a mathematical theory of sound propagation was set forth by Newton in his second book (1687), *Philosophiae Naturalis Principia Mathematica*. In this work he hypothesized that the velocity of sound is proportional to the square root of the absolute pressure divided by the density. Newton had discovered the isothermal velocity of sound in air. This is a less generally applicable formula than the adiabatic relationship, which was later suggested by Pierre Simon Laplace (1749–1827) in 1816. A fuller understanding of the propagation of sound waves had to wait until more elaborate mathematical techniques were developed.

Daniel Bernoulli (1700–1782), best known for his work in fluids, set forth the principle of the coexistence of small amplitude oscillations in a string, a theory later known as superposition. Soon after, Leonhard Euler (1707–1783) published a partial differential equation for the vibrational modes in a stretched string. The stretched-string problem is one that every physics major studies, due both to its relative simplicity and its importance in the history of science. The eighteenth century was a time when mathematics was just beginning to be applied to the study of mechanics. Prizes were offered by governments for the solution of important scientific problems of the day and there was vigorous and frequently acrimonious debate among natural philosophers in both private and public correspondence on the most appropriate solutions.

The behavior of sound in pipes and tubes was also of interest to mathematicians of the time. Both Euler (1727) and later J. L. Lagrange (1736–1830) made studies of the subject. Around 1759 there was much activity and correspondence between the two of them

(Lindsay, 1966). In 1766, Euler published a detailed treatise on fluid mechanics, which included a section entirely devoted to sound waves in tubes.

The tradition of offering prizes for scientific discoveries continued into the nineteenth century. The Emperor Napoleon offered, through the Institute of France, a prize of 3000 francs for a satisfactory theory of the vibration of plates (Lindsay, 1966). The prize was awarded in 1815 to Sophie Germain, a celebrated woman mathematician, who derived the correct fourth-order differential equation. The works of these early pioneers, along with his own insights, ultimately were collected into the monumental two-volume work, *Theory of Sound*, by John W. Strutt, Lord Rayleigh (1842–1919) in 1877. This classic work contains much that is original and insightful even today.

1.8 CLASSICAL PERIOD (1750–1825)

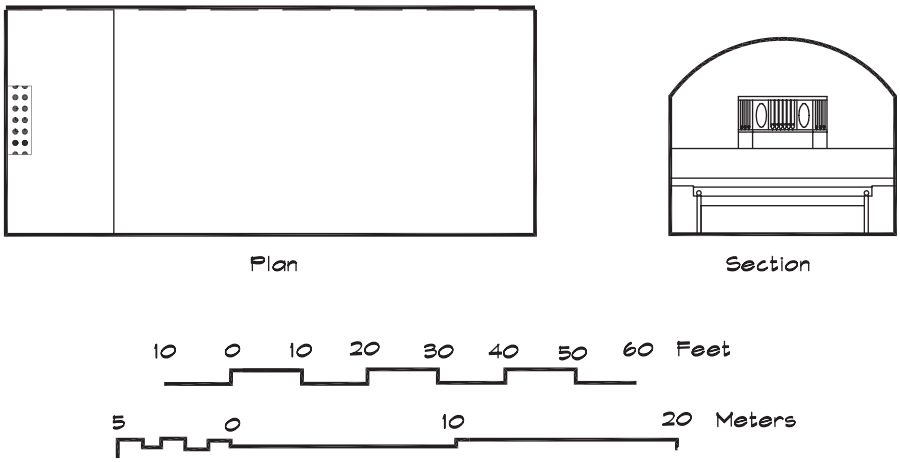
The eighteenth century in Europe was a cosmopolitan time when enlightened despots (often foreign born) were on the throne in many countries, and an intellectual movement known as the Enlightenment held that knowledge should evolve from careful observation and reason. The French *philsophes*, Rousseau, Montesquieu, and Voltaire reacted to the social conditions they saw and sought to establish universal rights of man. In both the visual and performing arts, there was a classic revival, a return to the spirit of ancient Greece and Rome. The paintings of Jacques Louis David, such as the *Oath of the Horatii* (1770), harkened back to Republican Rome and the virtues of nobility, simplicity, and perfection of form. The excavations of Pompeii and Herculeum had created public interest in the history of this earlier era and, with the American Revolution in 1776 and the French revolution in 1789, the interest took on political overtones.

The period referred to as Classical in music occurred during these years, though some historians, such as Grout and Palisca (1996) date it from 1720 to 1800. Classical refers to a time when music was written with careful attention to specific forms. One of these had a particular three-part or ternary pattern attributed to J. S. Bach's son, Carl Philipp Emanuel Bach (1714–1788), which is now called sonata form. Others included the symphony, concerto, and rondo. Compositions were written within the formal structure of each of the types. The best known composers of that time were Franz Joseph Haydn (1732–1809), Wolfgang A. Mozart (1756–1791), and later Ludwig Beethoven (1770–1827). During the Classical period musical pieces were composed for the first time with a formal concert hall performance in mind. Previously rooms that were used for musical concerts were rarely built specifically for that sole purpose.

In England in the middle of the eighteenth century, buildings first were built for the performance of nontheatrical musical works. Two immigrant musicians, Carl Fredrick Abel (1723–1787) and Johann (known as John) Christian Bach (1735–1782), the eighteenth child of J. S. Bach, joined forces with Giovanni Andrea Gallini, who provided the financing, to build between 1773 and 1775 what was to become the best-known concert hall in London for a century, the Hanover Square Rooms. The *Illustrated London News* of 1843 showed an engraving of the main concert hall (Forsyth, 1985) from which Fig. 1.13 was drawn.

When Haydn came to England in 1791–1792 and 1793–1794, he conducted his London Symphonies (numbers 93 to 101), which he had written specifically for this room. The main performance space was rectangular and, according to the London *General Evening Post* of February 25, 1794 (Forsyth, 1985), it measured 79 ft (24.1 m) by 32 ft (9.7 m). The height has been estimated at 22 to 28 ft. (6.7 to 8.5m). In Victorian times, it was lengthened to between 90 and 95 ft (Landon, 1995). It was somewhat small for its intended capacity (800)

FIGURE 1.13 Hanover Square Room, London, England (Forsyth, 1985)

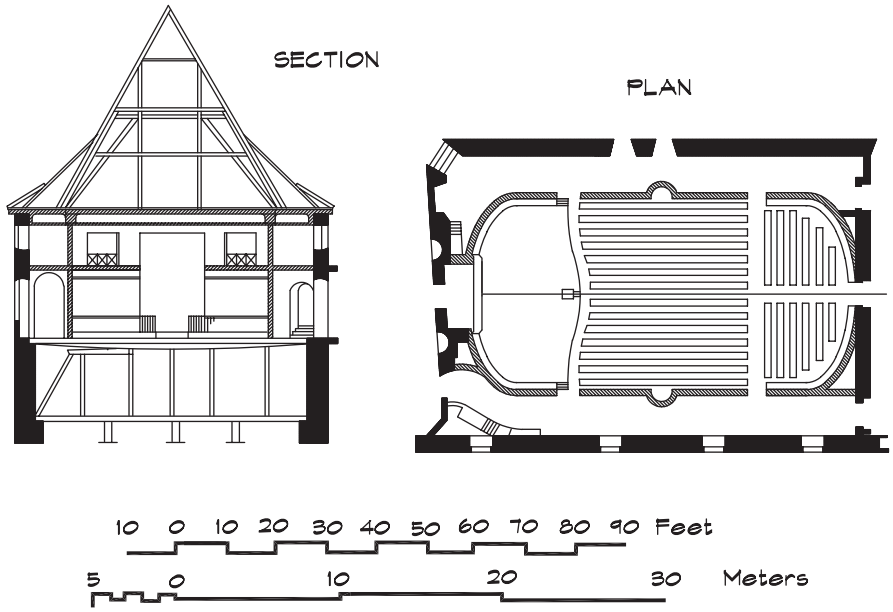


and probably had a reverberation time of less than one second when fully occupied (J. Meyer, 1978). The low volume and narrow width would have provided strong lateral reflections and excellent clarity, albeit a somewhat loud overall level. The room was well received at the time. The *Berlinische Musikalische Zeitung* published a letter on June 29, 1793 describing a concert there by a well-known violinist, Johann Peter Salomon (1745–1815): "The room in which [the concert] is held is perhaps no longer than that in Stadt Paris in Berlin, but broader, better decorated, and with a vaulted ceiling. The music sounds, in the hall, beautiful beyond any description." (Forsyth, 1985)

In the eighteenth century the center of gravity of the music in Europe shifted northward from Italy. Orchestras in London, Paris, Mannheim, Berlin, and Vienna were available to composers of all nationalities. Halls were built in Dublin, Oxford, and Edinburgh, many years before they appeared in cities on the continent. The Holywell Music Room at Oxford, which opened in 1748, still stands today. These halls were relatively small by today's standards with seating capacities ranging from 400 to 600, and reverberation times were generally less than 1.5 seconds (Bagenal and Wood, 1931). Music was also played at public concerts held outdoors in pleasure gardens. In 1749 some 12,000 people paid two shillings sixpence each to hear Handel's 100-piece band rehearse his *Royal Fireworks Music* at Vauxhall Gardens (Forsyth, 1985).

In continental Europe in the mid eighteenth century there was not yet a tradition of public concerts open to all. Concert-goers were, by and large, people of fashion and concerts were usually held in rooms of the nobility, such as Eisenstadt Castle south of Vienna or Eszterhaza Castle in Budapest, which was the home of Haydn during his most productive years. It was not until 1761 that a public hall was built in Germany, the Konzert-Saal auf dem Kamp in Hamberg. In Leipzig, perhaps because it did not have a royal court, the architect Johann Carl Friedrich Dauthe converted a Drapers' Hall or Gewandhaus into a concert hall in 1781. Later known as the Altes Gewandhaus, it seated about 400 with the orchestra located on a raised platform at one end occupying about one quarter of the floor space. It is pictured in Fig. 1.14. The room had a reverberation time of about 1.3 seconds (Bagenal and Wood, 1931) and was lined with wood paneling, which reduced the bass build up. Recognized for its fine acoustics, particularly during Felix Mendelssohn's directorship in the

FIGURE 1.14 Altes Gewandhaus, Leipzig, Germany (Bagenal and Wood, 1931)



mid-nineteenth century (1835–1847), it was later replaced by the larger Neus Gewandhaus late in the century.

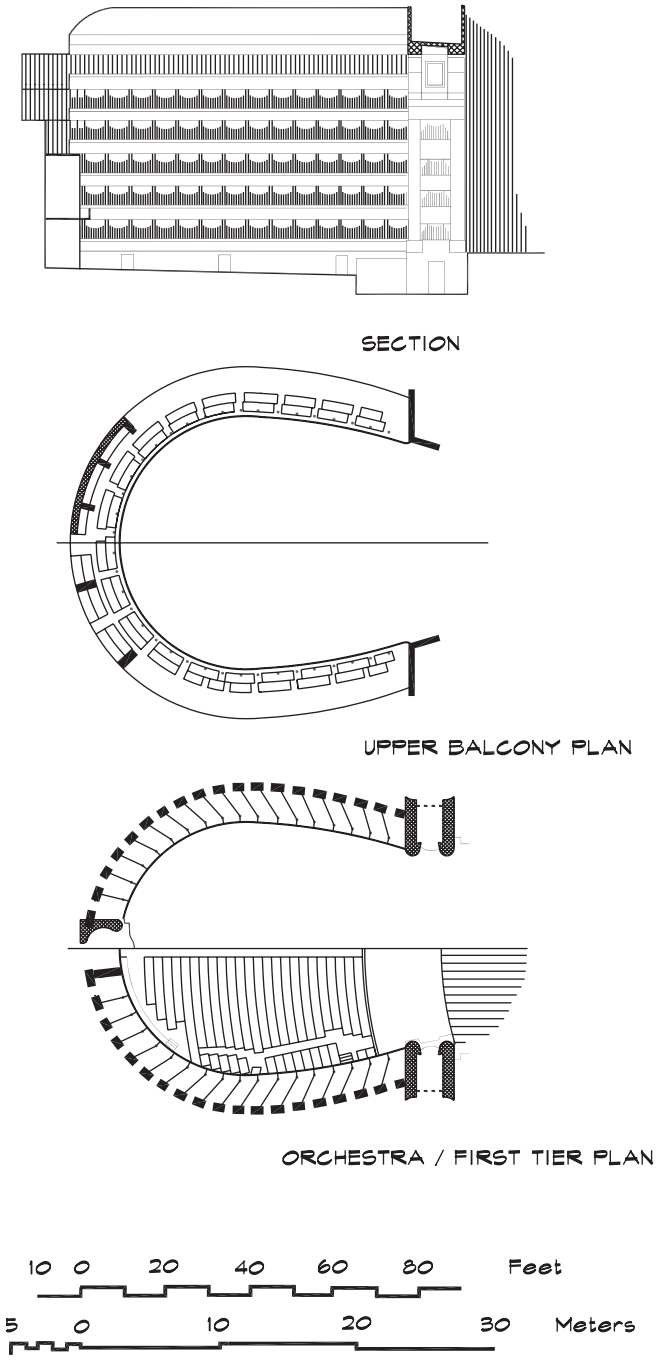
Vienna became an international cultural center where artists and composers from all over Europe came to work and study, including Antonio Salieri (1750–1825), Mozart, and Beethoven. Two principal concert halls in Vienna at the time were the Redoutensaal at Hofburg and the palace of the Hapsburg family. Built in 1740, these two rooms, seating 1500 and 400, respectively, remained in use until 1870. The larger room was rectangular, had a ceiling height of about 30 ft, and side galleries running its full length. The reverberation time was probably slightly less than 1.6 seconds when fully occupied. The rooms had flat floors and were used for balls as well as for concerts. Haydn, Mozart, and Beethoven composed dances for these rooms, and Beethoven's Seventh Symphony was first performed here in 1814 (Forsyth, 1985).

Meanwhile in Italy little had changed. Opera was the center of the cultural world and opera-house design had developed slowly over two centuries. In 1778 La Scalla opened in Milan and has endured, virtually unchanged, for another two centuries. Shown in Fig. 1.15, it has the form of a horseshoe-shaped layer cake with small boxes lining the walls. The sides of the boxes are only about 40% absorptive (Beranek, 1979) so they provide a substantial return of reflected sound back to the room and to the performers. The orchestra seating area is nearly flat, reminiscent of the time when there were no permanent chairs there. The seating arrangement is quite efficient (tight by modern standards), and the relatively low (1.2 sec) reverberation time makes for good intelligibility.

1.9 ROMANTIC PERIOD (1825–1900)

The terms Classic and Romantic are not precisely defined nor do they apply strictly to a given time period. Music written between about 1770 and 1900 lies on a continuum, and every composer of the age employed much the same basic harmonic vocabulary (Grout and

FIGURE 1.15 Theatro Alla Scalla, Milan, Italy (Beranel, 1979)



Palisca, 1996). Romantic music is more personal, emotional, and poetic than the Classical and less constrained by a formal style. The Romantic composers wanted to describe thoughts, feelings, and impressions with music, sometimes even writing music as a symphonic poem or other program to tell a story. Although Beethoven lived during the Classical time period, much of his music can be considered Romantic, particularly his sixth and ninth symphonies.

Clearly he bridged the two eras. The best known Romantic composers were all influenced by Beethoven including Franz Schubert (1797–1828), Hector Berlioz (1803–1869), Felix Mendelssohn (1809–1847), Johannes Brahms (1833–1897), and Richard Wagner (1813–1883).

A common characteristic of Classical composers was their familiarity with the piano, which had become the most frequently used instrument. Some Romantic composers were also virtuoso pianists including Franz Liszt (1811–1886), Edvard Grieg (1843–1907), Frederic Chopin (1810–1849), and of course Beethoven. The wide dynamic range of this instrument originally led to its name, the forte (loud) piano (soft), and socially prominent households were expected to have one in the parlor.

As musical instruments increased in loudness they could be heard by larger audiences, which in turn encouraged larger concert halls and the use of full orchestras. As performance spaces grew larger there arose an incentive to begin thinking more about their acoustical behavior. Heretofore room shapes had evolved organically, the Italian opera from the Greek and Roman theaters, and the Northern European concert halls from basilican churches and rectangular ballrooms. Many of these rooms were enormously successful and are still today marvels of empirical acoustical design, although there were also those that were less than wonderful. The larger rooms begat more serious difficulties imposed by excessive reverberation and long delayed reflections.

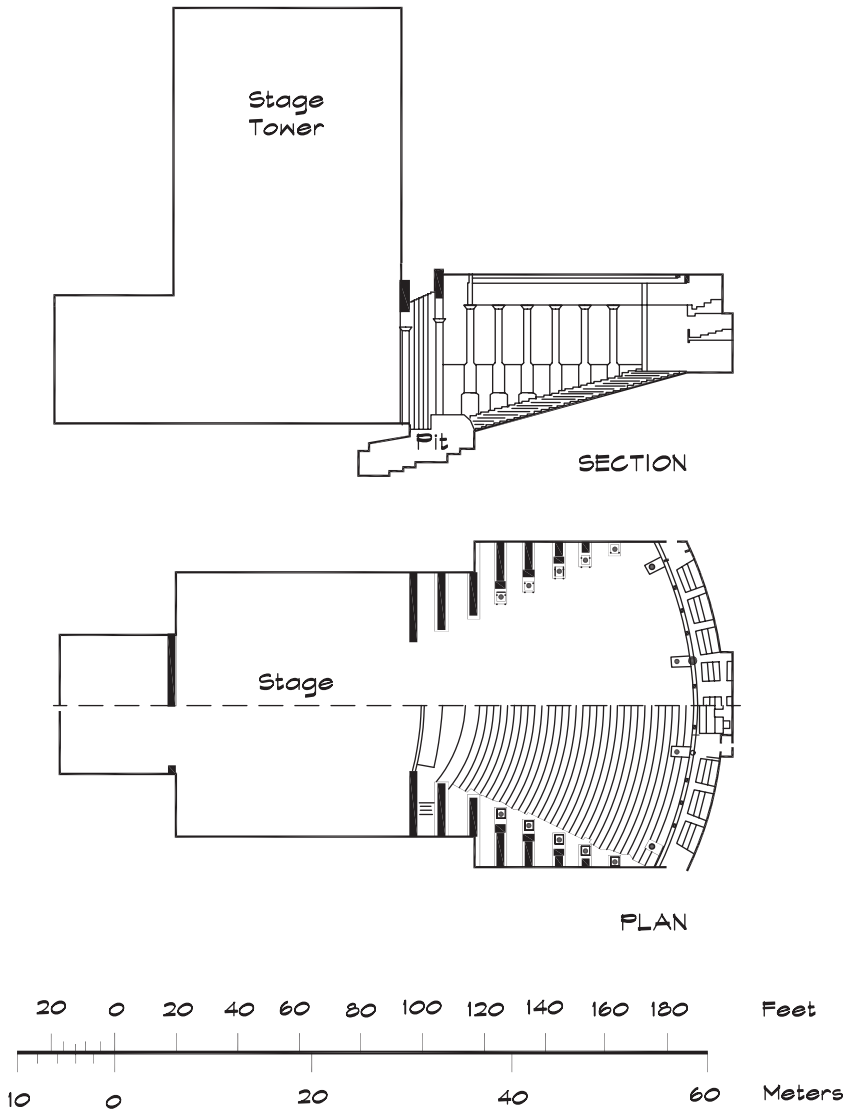
Concerts were performed in the famous Crystal Palace designed by Joseph Paxton, which had housed the Great Exhibition of 1851 and was later moved from Hyde Park to Sydenham Hill in 1854. This huge structure was built of glass, supported by a cast iron framework, and became a popular place for weekly band concerts. Occasionally mammoth festival concerts were held there, which, for example, in 1882 played to an audience of nearly 88,000 people using 500 instrumentalists and 4000 choir members (Forsyth, 1985).

Knowledge of the acoustical behavior of rooms had not yet been set out in quantitative form. Successful halls were designed using incremental changes from previously constructed rooms. The frustration of many nineteenth-century architects with acoustics is summarized in the words of Jean Louis Charles Garnier (1825–1898), designer of the Paris Opera House, "I gave myself pains to master this bizarre science [of acoustics] but . . . nowhere did I find a positive rule to guide me; on the contrary, nothing but contradictory statements . . . I must explain that I have adopted no principle, that my plan has been based on no theory, and that I leave success or failure to chance alone . . . like an acrobat who closes his eyes and clings to the ropes of an ascending balloon." (Garnier, 1880)

One of the more interesting theatrical structures to be built in the century, Wagner's opera house, the Festspielhaus in Bayreuth, Germany built in 1876, was a close collaboration between the composer and the architect, Otto Brueckwald, and was designed with a clear intent to accomplish certain acoustical and social goals. The auditorium is rectangular but it contains a fan-shaped seating area with the difference being taken up by a series of double columns supported on wing walls. The plan and section are shown in Fig. 1.16. The seating arrangement in itself was an innovation, since it was the first opera house where there was not a differentiation by class between the boxes and the orchestra seating. The horseshoe shape with layered boxes, which had been the traditional form of Italian opera houses for three centuries, was abandoned for a more egalitarian configuration.

Most unusual, however, was the configuration of the pit, which was deepened and partially covered with a radiused shield that directed some of the orchestral sound back toward the actors. This device muted the orchestral sound heard by the audience, while allowing the musicians to play at full volume out of sight of the audience. It also changed the

FIGURE 1.16 Festspielhaus, Bayreuth, Germany (Beranek, 1979)

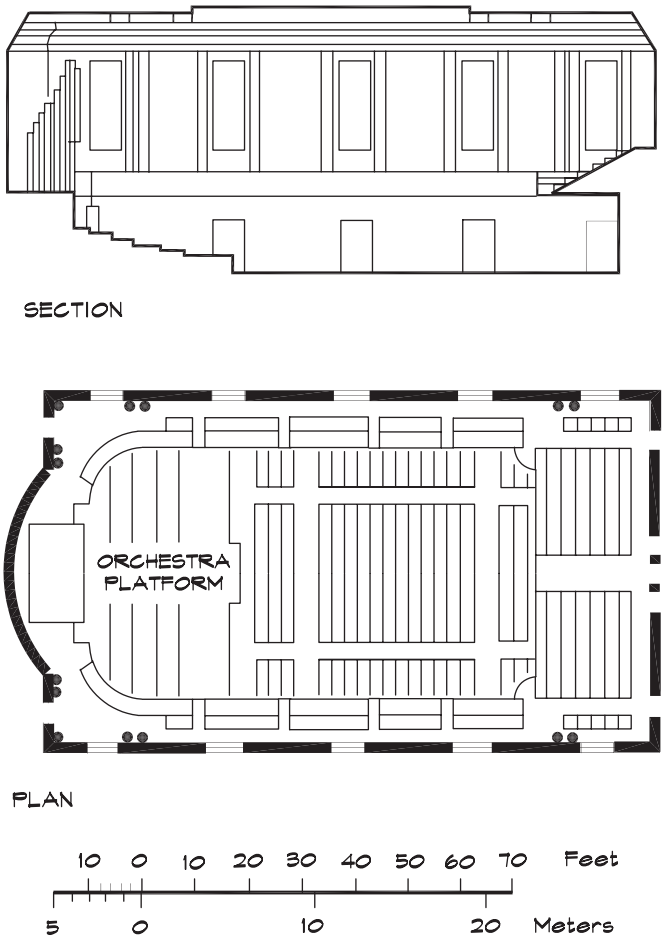


loudness of the strings with respect to the horns, improving the balance between the singers and the orchestra. The reverberation time, at 1.55 seconds (Beranek, 1996), was particularly well suited to Wagner's music, perhaps because he composed pieces to be played here, but the style has not been replicated elsewhere.

Shoebox Halls

Several of the orchestral halls constructed in the late eighteenth and early nineteenth centuries are among the finest ever built. Four of them are particularly noteworthy, both for their fine acoustics and for their influence on later buildings. They are all of the shoebox type with high ceilings, multiple diffusing surfaces, and a relatively low seating capacity. The oldest is the Stadt Casino in Basel, Switzerland, which was completed in 1776. Shown in Fig. 1.17,

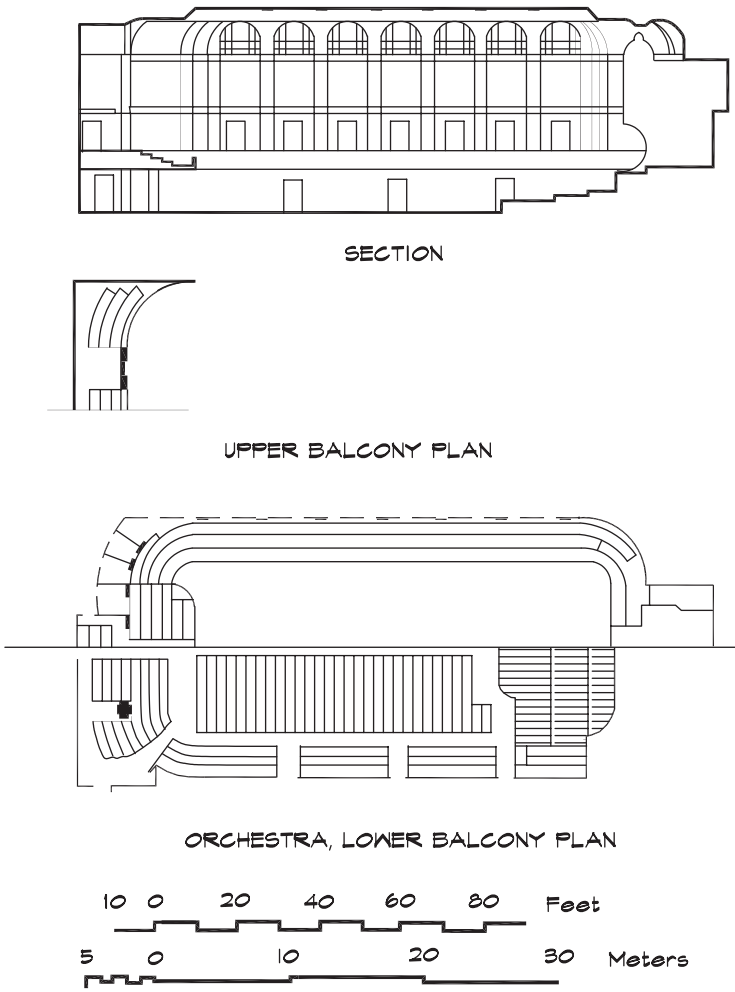
FIGURE 1.17 Concert Hall, Stadt Casino, Basel, Switzerland (Beranek, 1979)



it is very typical of the age with a flat floor reminiscent of the earlier ballrooms, small side and end balconies, and a coffered ceiling. The orchestra was seated on a raised platform with risers extending across its width. Above and to the rear of the orchestra was a large organ. The hall seated 1448 people and had a mid-frequency reverberation time of about 1.8 seconds (Beranek, 1996) making it ideal for Classical and Romantic music.

Ten years later the Neues Gewandhaus was built to provide a larger space for concerts in Leipzig. After it was completed, the old Altes Gewandhaus was torn down. The building was based on a design by the architects Martin K. P. Gropius (1824–1880) and Heinrich Schmieden (1835–1913) and was finally completed in 1882 after Gropius' death, remaining extant until it was destroyed in World War II. A sketch of the hall is shown in Fig. 1.18. Its floor plan is approximately two squares, side by side, measuring 37.8 m (124 ft) by 18.9 m (62 ft) with a 14.9 m (49 ft) high ceiling. The new room housed 1560 in upholstered seats and its reverberation time at 1.55 seconds was less than that of the other three, making it ideal for the works of Bach, Mozart, Haydn, and other Classical chamber music. The upper walls were pierced with arched clerestory windows, looking like the brim of a baseball cap, which let in light and helped to control the bass reverberation. The structural interplay of the

FIGURE 1.18 Neues Gewandhaus, Leipzig, Germany (Beranek, 1979)

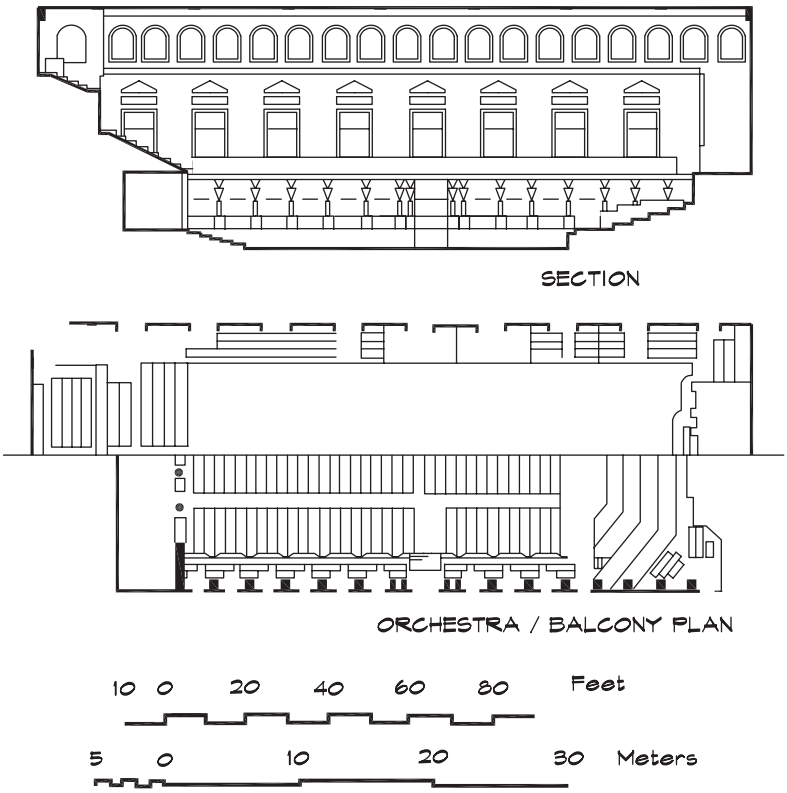


curved transition to the ceiling yielded a highly dramatic form, which, along with three large chandeliers, added diffusion to the space. Like the other halls of this type it had a narrow balcony around its perimeter of about three rows of seating, with a large organ towering over the orchestra.

Grosser Musikvereinssaal (Fig. 1.19) in Vienna, Austria, which is still in use today, is considered one of the top three or four concert halls in the world. It was opened in 1870 and has a long (50.3 m or 185 ft) and narrow (19.8 m or 65 ft) rectangular floor plan with a high (15 m or 50 ft), heavily beamed ceiling. The seating capacity, at 1680 in wooden seats, is relatively small for so long a room.

The single narrow balcony is supported by a row of golden caryatids, much like giant Oscars, around the side of the orchestra seating. Reflections from the underside of the balcony and the statuary are particularly important in offsetting the grazing attenuation due to the audience seated on a flat floor. The high windows above the balcony provided light for afternoon concerts and reduced the bass buildup.

FIGURE 1.19 Grosser Musikvereinssaal, Vienna, Austria (Beranek, 1979)

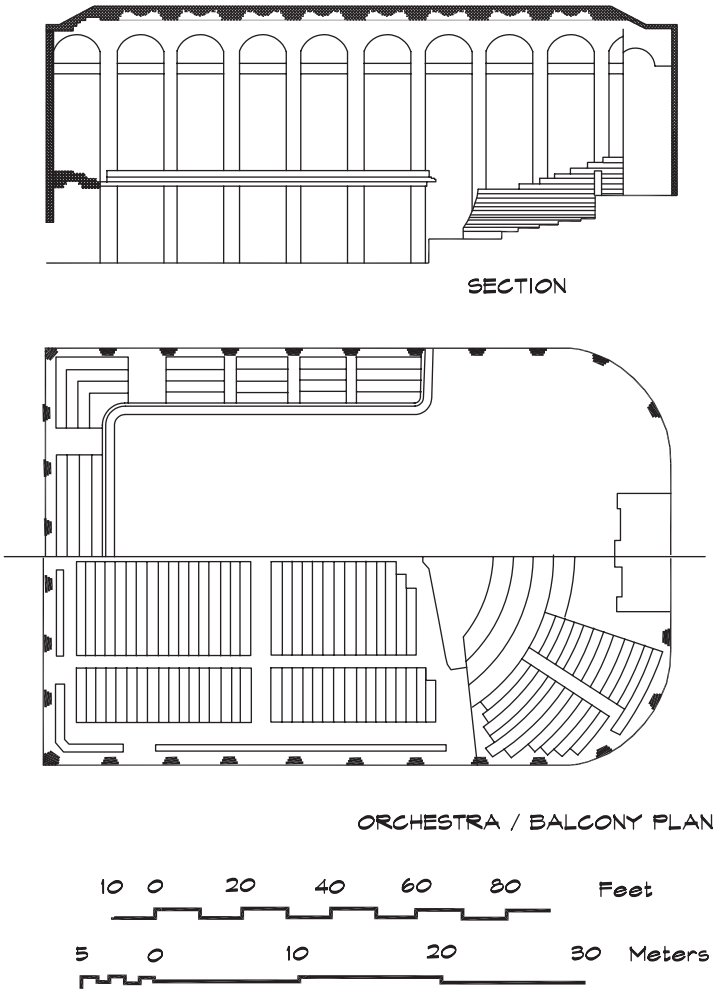


Grosser Musikvereinssaal also was known as the Goldener Saal, since its interior surfaces are covered by meticulously applied paper-thin sheets of gold leaf. The sound in this hall is widely considered ideal for Classical and Romantic music. Its reverberation time is long, just over 2 seconds when fully occupied, and the narrowness of the space provides for strong lateral reflections that surround or envelop the listener in sound. The walls are constructed of thick plaster that supports the bass, and the nearness of the reflecting surfaces and multiple diffusing shapes gives an immediacy and clarity to the high strings. It is this combination of clarity, strong bass, and long reverberation time that is highly prized in concert halls, but rarely achieved.

Concertgebouw in Amsterdam, Netherlands (Fig. 1.20) is the last of the four shoebox halls. Designed by A. L. Van Gendt, it opened in 1888. Like the others it is rectangular; however, at 29 m (95 ft) it is wider than the other three and seats 2200 people on a flat floor. Consequently it is more reverberant at 2.2 seconds and has somewhat less clarity than Grosser Musikvereinssaal. It is best suited to large-scale Romantic music, providing a live, full, blended tone.

The four halls cited here have similar features that contribute to their excellent acoustics. They are all rectangular and relatively narrow (except in the case of Concertgebouw). The construction is of thick plaster and heavy wood with a deeply coffered ceiling about 15 meters high. The floors are generally flat and the orchestra is seated above the heads of the patrons on a high, raked, wooden platform. The orchestra is located in the same room as the audience rather than being set back into a stage platform. All these rooms are

FIGURE 1.20 Concertgebouw, Amsterdam, Netherlands (Beranek, 1979)



highly ornamented with deep fissures, statuary, recessed windows, organs, and overhanging balconies to help diffuse the sound. They all had highly ornate chandeliers that also scatter the sound.

The capacity of these rooms is not great by modern standards and the seating is tight. No seat is far from a side wall or from the orchestra. The orchestra is backed by a hard reflecting surface to help project the sound, particularly the bass, out to the audience. There is a notable absence of thin wood paneling in these structures. Paneling at one time was considered acoustically desirable in accordance with the hall as a musical instrument theory. These rooms provided excellent acoustics and became the examples to be emulated in the scientific approach to concert hall performance, begun early in the following century.

1.10 BEGINNINGS OF MODERN ACOUSTICS

The nineteenth century produced the beginnings of the study of acoustics as a science and its dissemination in the published literature via technical books and journals. Heretofore scientific ideas had a relatively limited audience and were often distributed through personal

correspondence between leading scholars of the day. Frequently written in Latin they were not generally accessible to the public. In the nineteenth century, books written in English or German, such as Hermann von Helmholtz (1821–1894) *Sensations of Tone* in 1860, established the field as a science where measurement, observation, and a mathematical approach could lead to significant progress. Later in the century (1877) John W. Strutt, Lord Rayleigh published the first of his two-volume set, *Theory of Sound*, followed by the second between 1894 and 1896, which was one of the most important books ever written in the field. In it he pulled together the disparate technical articles of the day and added many valuable contributions of his own. It is remarkable that such a clear presentation of acoustical phenomena was written before careful experimental work was possible. In Rayleigh's time the only practical sound source was a bird whistle (Lindsay, 1966) and the most sensitive detection device (besides the ear) was a gas flame.

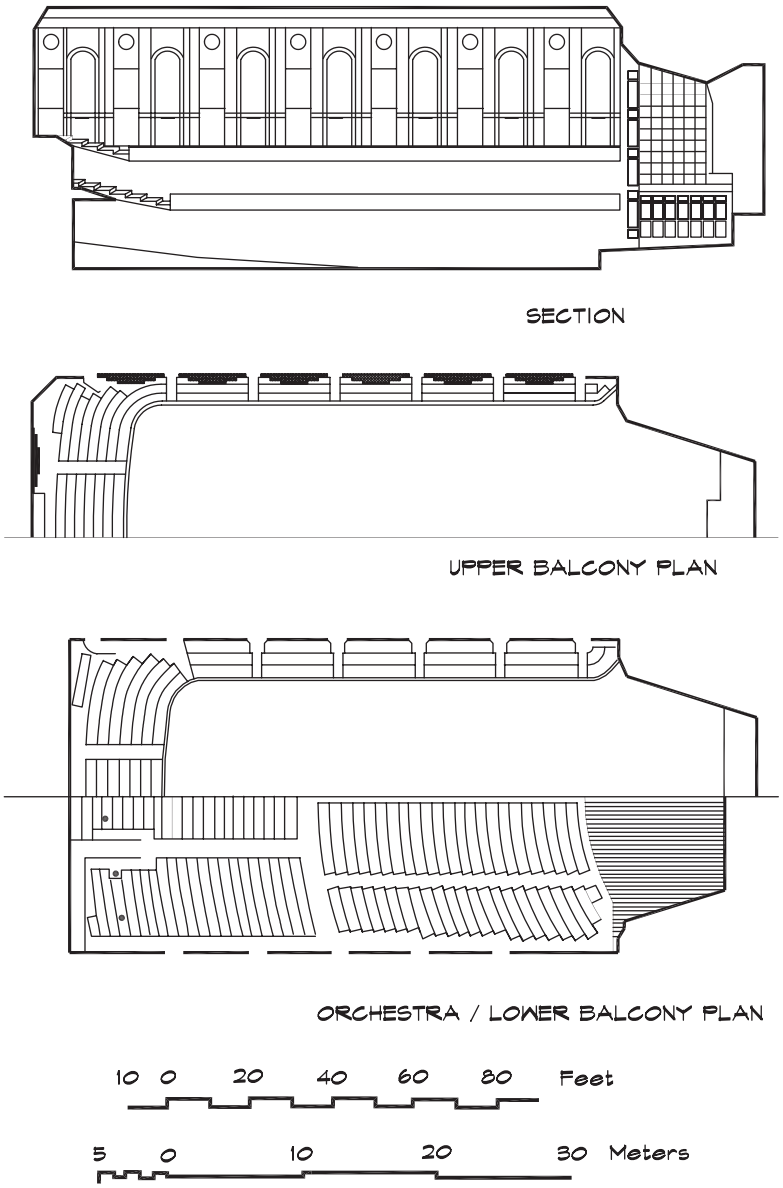
About the same time, in the remarkable decade of the 1870s, there was a surge in the development of practical electroacoustic devices. In Germany, Ernst W. Siemens patented in 1874 the moving coil transducer, which eventually led to today's loudspeaker. In 1887 the U.S. Supreme Court held in favor of the patent, originally filed in 1876, and probably the single most valuable patent ever issued, of Alexander Graham Bell (1847–1922) for the telephone. It incorporated the granular carbon microphone, the first practical microphone, and one of the few instruments that is improved by banging it on a table. Within a year (1877), Thomas A. Edison had patented the phonograph and somewhat later, in 1891, motion pictures. Thus within a decade the technical foundation for the telephone, sound recording, music reproduction, and motion-picture industries had been developed.

In the late nineteenth and early twentieth centuries, the theoretical beginnings of architectural acoustics were started by a young physics professor at Harvard College, W. C. Sabine. Sabine's work began inauspiciously enough following a request by president Elliot to "do something" about the acoustical difficulties in the then new Fogg Art Museum auditorium, which had been completed in 1895 (Sabine, 1922). Sabine took a rather broad view of the scope of this mandate and commenced a series of experiments in three Harvard auditoria with the goal of discovering the reasons behind the difficulties in understanding speech. By the time he had completed his work, he had developed the first theory of sound absorption of materials, its relationship to sound decay in rooms, and a formula for the decay (reverberation) time in rooms. His key discovery was that the product of the total absorption and the reverberation time was a constant.

Soon after this discovery in 1898 he helped with the planning of the Boston Music Hall, now called Symphony Hall. He followed the earlier European examples, using a shoebox shape and heavy plaster construction with a modest ceiling height to maintain a reverberation time of 1.8 seconds. Narrow side and rear balconies were used to avoid shadow zones and a shallow stage enclosure, with angled walls and ceiling, directed the orchestra sound out to the audience. The deeply coffered ceiling and wall niches containing classical statuary helped provide excellent diffusion (Hunt, 1964). The auditorium, pictured in Fig. 1.21, opened in 1900 and is still one of the three or four best concert halls in the world.

While the designers of Boston Symphony Hall followed one European design tradition, the designers of New York's Metropolitan Opera House (Fig. 1.22) followed another, that of the Italian opera houses. Opening in 1883 the Met, seating over 3600, is one of the largest opera houses in the world. Despite its size it has reasonably good acoustics in the middle balconies; however, the orchestra seats and the upper balcony seats are less satisfactory (Beranek, 1979). With a volume nearly twice that of La Scalla, it is difficult for singers to sound as loud as in Milan. The hall, with some ceiling and balcony front additions by

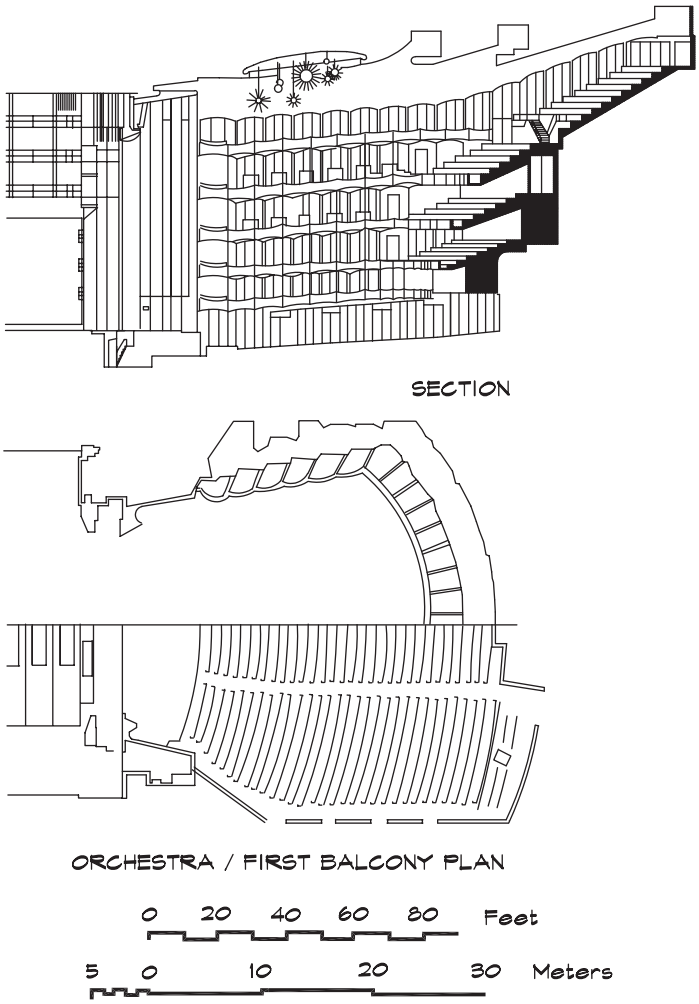
FIGURE 1.21 Symphony Hall, Boston, MA, USA (Beranek, 1979)



architect Wallace K. Harrison and acousticians Cyril Harris and Vilhelm Jordan to increase diffusion and the sound in the balconies, is in active use today.

Another American hall, constructed around the turn of the century, was Carnegie Hall (Fig. 1.23) in New York. Andrew Carnegie, an entrepreneur and steel baron, was fishing at his vacation home in Scotland with a young American musician, Walter Damrosch, whose father Leopold was director of the New York Symphony Society. The idea to provide a permanent building to house its activities arose while the two were casting in midstream (Forsyth, 1985). The plans were prepared by architect William B. Turnhill and the hall opened in 1891. Carnegie Hall was designed as a shoebox hall but like a theater. The orchestra was

FIGURE 1.22 Metropolitan Opera House, New York, NY, USA (Beranek, 1979)

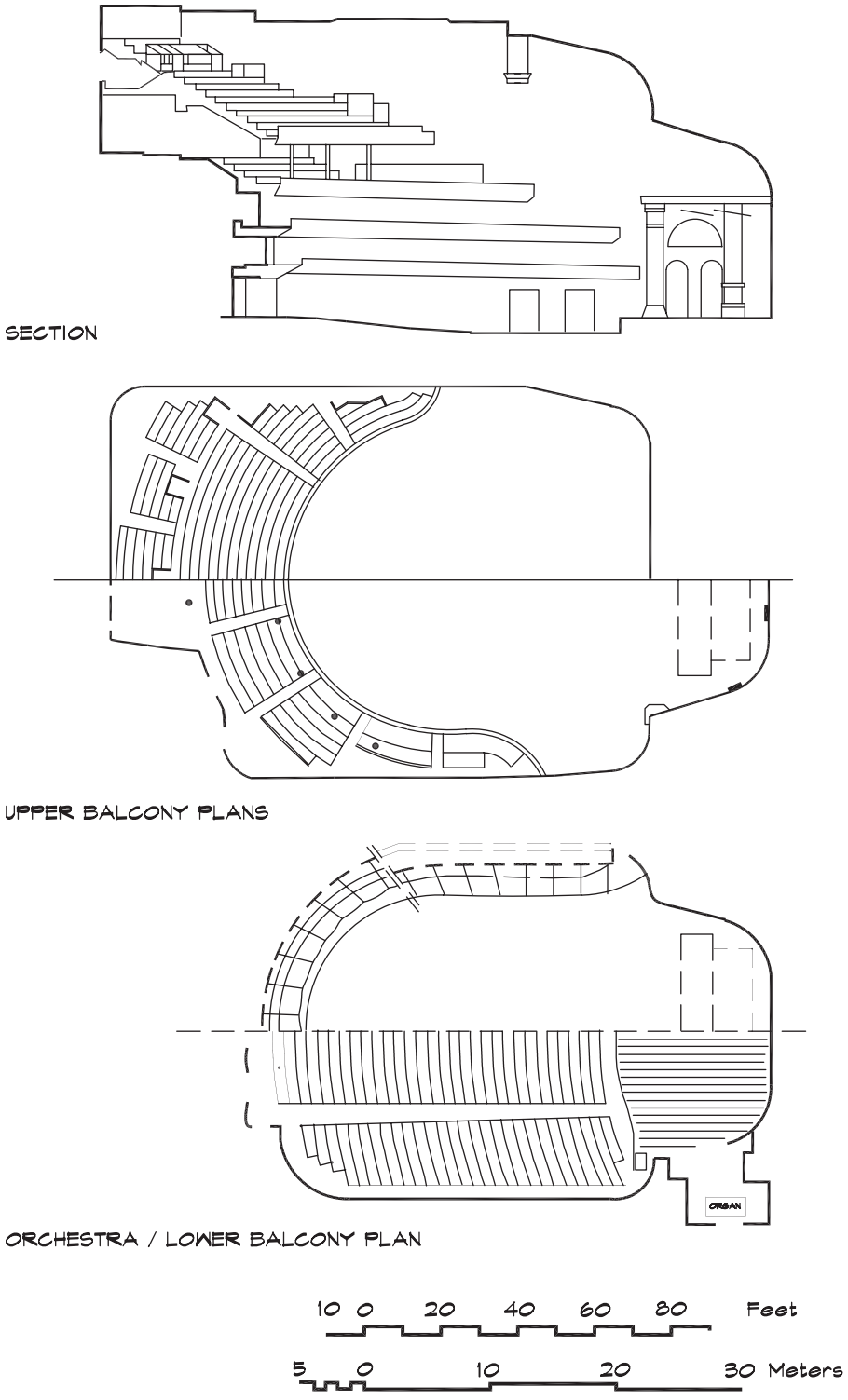


located on stage behind a proscenium arch under a curved orchestra shell. The audience is seated on a nearly flat floor and in four balconies, whose rounded front faces are stacked on an imaginary cylinder. Each balcony flares out into side balconies, which almost reach the stage at the lowest level. Carnegie Hall is known for the clarity of its high frequency sound. At 1.7 seconds it has a slightly dry reverberation with less bass support than in Boston. It was recently refurbished with the stated objective of leaving the acoustical properties unchanged.

1.11 TWENTIETH CENTURY

In the twentieth century, architectural acoustics came to be recognized as a science as well as an art. Although the number and quality of the published works increased, our understanding of many of the principles of acoustical design did not in all cases lead to improvements in concert halls. The more routine aspects of room acoustics, including noise and vibration control and development of effective acoustical materials, experienced marked improvements.

FIGURE 1.23 Carnegie Hall, New York, NY, USA (Beranek, 1979)



The development of electroacoustic devices including microphones, amplifiers, loudspeakers, and other electronic processing instruments flourished. The precision, which is now available in the ability to record and reproduce sound, has in a sense created an expectation of excellence that is difficult to match in a live performance. The high-frequency response in a hall is never as crisp as in a close-miked recording. The performance space is seldom as quiet as a recording studio. The seats are never as comfortable as in a living room. Ironically, just as we have begun to understand the behavior of concert halls and are able to accurately model their behavior, electroacoustic technology has developed to the point where it may soon provide an equivalent or even superior experience in our homes.

This page intentionally left blank

2 FUNDAMENTALS OF ACOUSTICS

2.1 FREQUENCY AND WAVELENGTH

Frequency

A steady sound is produced by the repeated back and forth movement of an object at regular intervals. The time interval over which the motion recurs is called the *period*. For example if our hearts beat 72 times per minute, the period is the total time (60 seconds) divided by the number of beats (72), which is 0.83 seconds per beat. We can invert the period to obtain the number of complete cycles of motion in one time interval, which is called the *frequency*.

$$f = \frac{1}{T} \tag{2.1}$$

where f = frequency (cycles per second or Hz)
T = time period per cycle (s)

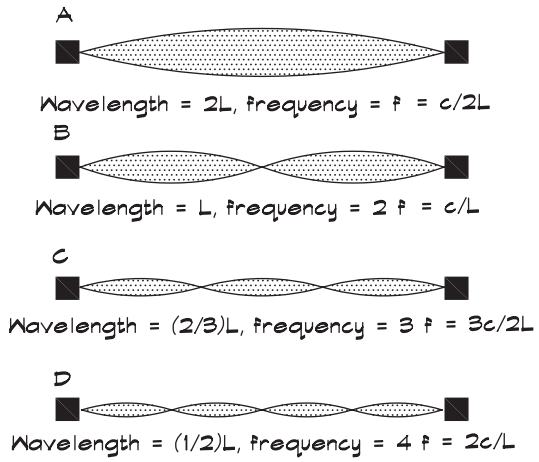
The frequency is expressed in units of cycles per second, or Hertz (Hz), in honor of the physicist Heinrich Hertz (1857–1894).

Wavelength

Among the earliest sources of musical sounds were instruments made using stretched strings. When a string is plucked it vibrates back and forth and the initial displacement travels in each direction along the string at a given velocity. The time required for the displacement to travel twice the length of the string is

$$T = \frac{2\,L}{c} \tag{2.2}$$

FIGURE 2.1 Harmonics of a Stretched String (Pierce, 1983)



where T = time period (s)
L = length of the string (m)
c = velocity of the wave (m/s)

Since the string is fixed at its end points, the only motion patterns allowed are those that have zero amplitude at the ends. This constraint (called a *boundary condition*) sets the frequencies of vibration that the string will sustain to a fundamental and integer multiples of this frequency, $2f, 3f, 4f, \ldots$, called *harmonics*. Figure 2.1 shows these vibration patterns.

$$f = \frac{c}{2L} \tag{2.3}$$

As the string displacement reflects from the terminations, it repeats its motion every two lengths. The distance over which the motion repeats is called the *wavelength*, and is given the Greek symbol lambda, λ, which for the fundamental frequency in a string is 2 L. This leads us to the general relation between the wavelength and the frequency

$$\lambda = \frac{c}{f} \tag{2.4}$$

where λ = wavelength (m)
c = velocity of wave propagation (m/s)
f = frequency (Hz)

When notes are played on a piano the strings vibrate at specific frequencies, which depend on their length, mass, and tension. Figure 2.2 shows the fundamental frequencies associated with each note. The lowest note has a fundamental frequency of about 27 Hz, while the highest fundamental is 4186 Hz. The frequency ranges spanned by other musical

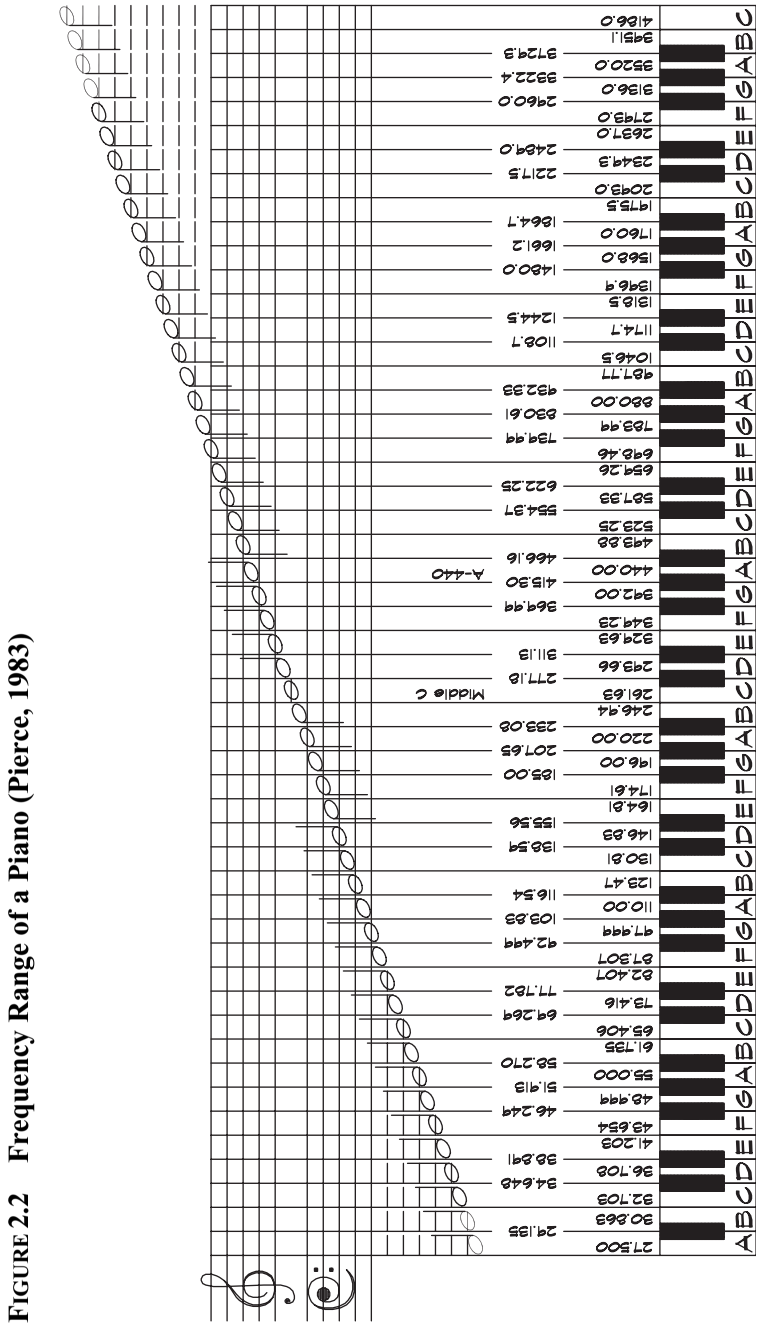


FIGURE 2.2 Frequency Range of a Piano (Pierce, 1983)

instruments, including the human voice, are given in Fig. 2.3. If a piano string is vibrating at its fundamental mode, the maximum excursion occurs at the middle of the string. When a piano key is played, the hammer does not strike precisely in the center of the string and thus it excites a large number of additional modes. These harmonics contribute to the beauty and complexity of the sound.

Frequency Spectrum

If we were to measure the strength of the sound produced by a particular note and make a plot of sound level versus frequency we would have a graph called a *spectrum*. When the sound has only one frequency, it is called a *pure tone* and its spectrum consists of a single straight line whose height depends on its strength. The spectrum of a piano note, shown in Fig. 2.4, is a line at the fundamental frequency and additional lines at each harmonic frequency. For most notes the fundamental has the highest amplitude, followed by the harmonics in descending order. For piano notes in the lowest octave the second harmonic may have a higher amplitude than the fundamental if the strings are not long enough to sustain the lowest frequency.

Sources such as waterfalls produce sounds at many frequencies, rather than only a few, and yield a flat spectrum. Interestingly an impulsive sound such as a hand clap also yields a flat spectrum. This is so because in order to construct an impulsive sound, we add up a very large number of waves of higher and higher frequencies in such a way that their peaks all occur at one time. At other times they cancel each other out so we are left with just the impulse spike. Since the two forms are equivalent, a sharp impulse generates a large number of waves at different frequencies, which is a flat spectrum. A clap often is used to listen for acoustical defects in rooms.

Electronic signal generators, which produce all frequencies within a given bandwidth, are used as test sources. The most commonly encountered are the pink-noise (equal energy per octave or third octave) or white-noise (equal energy per cycle) generators.

Filters

In analyzing the spectral content of a sound we might use a meter that includes electronic filters to eliminate all signals except those of interest to us. Filters have a center frequency and a bandwidth, which determines the limits of the filter. By international agreement certain standard center frequencies and bandwidths are specified, which are set forth in Table 2.1. The most commonly used filters in architectural acoustics have octave or third-octave bandwidths. Three one-third octaves are contained in each octave, but these do not correspond to any given set of notes. Narrow bandwidth filters, 1/10 octave or even 1 Hz wide, are sometimes used in the study of vibration or the details of reverberant falloff in rooms.

2.2 SIMPLE HARMONIC MOTION

Periodic motions need not be smooth. The beat of a human heart, for example, is periodic but very complicated. It is easiest, however to begin with a simple motion and then to move on to more complicated wave shapes. If we examine the vibration of a stretched string it is quite regular. Such behavior is called simple harmonic motion and can be written in terms of a sinusoidal function.

FIGURE 2.3 Frequency Ranges of Various Musical Instruments (Pierce, 1983)

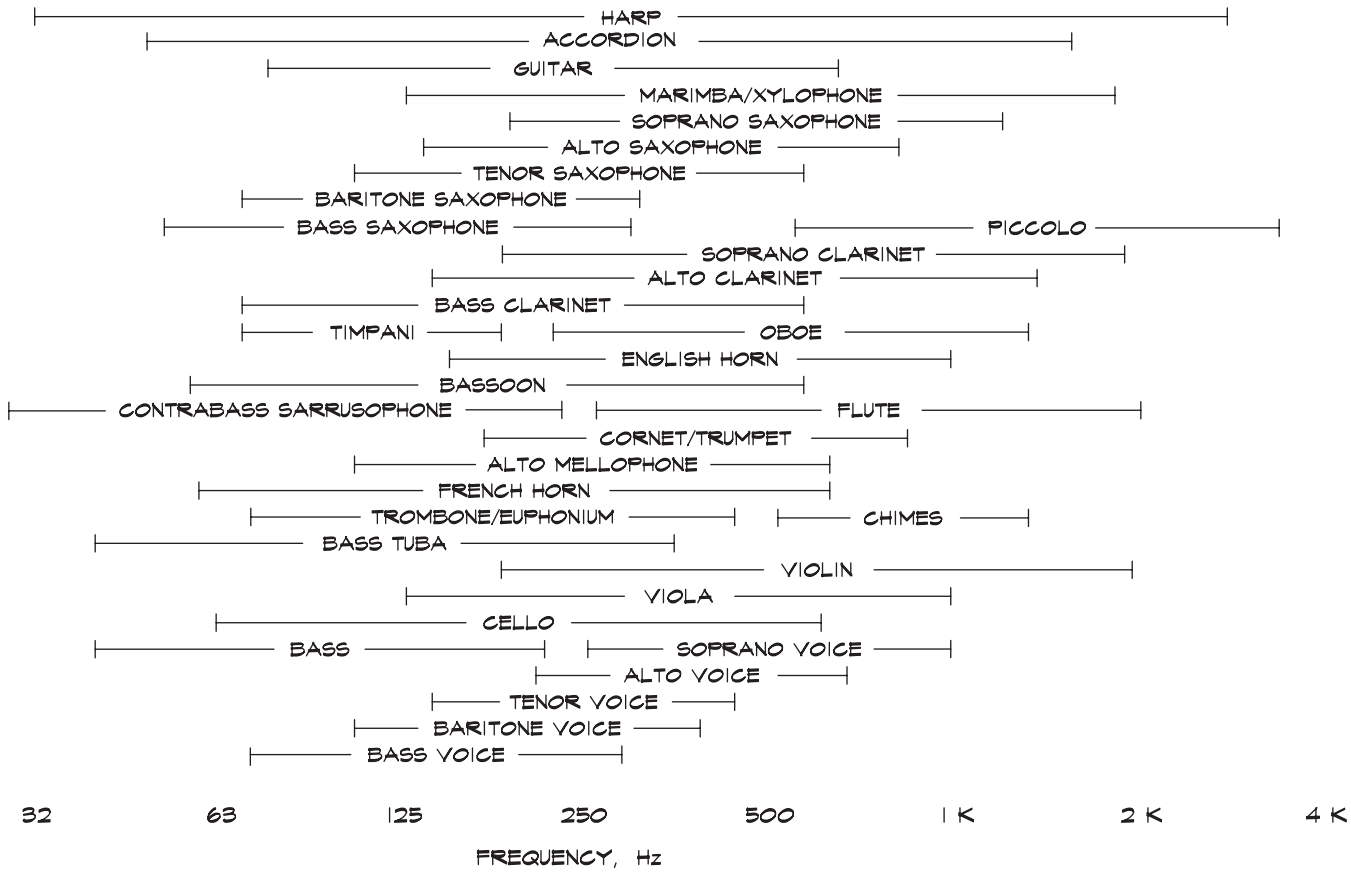


FIGURE 2.4 Frequency Spectrum of a Piano Note

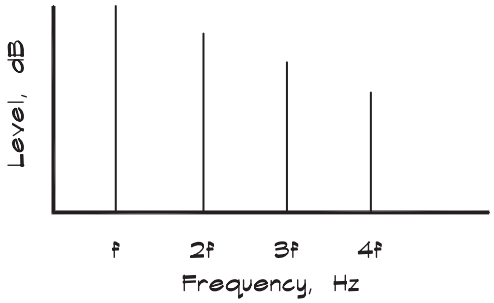
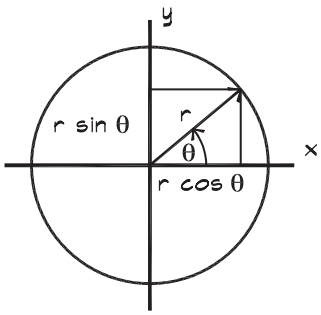


TABLE 2.1 Octave and Third-Octave Band Frequency Limits

	Frequency, Hz					
	Octave			One-third Octave		
Band	Lower Limit	Center	Upper Limit	Lower Limit	Center	Upper Limit
12	11	16	22	14.1	16	17.8
13				17.8	20	22.4
14				22.4	25	28.2
15	22	31.5	44	28.2	31.5	35.5
16				35.5	40	44.7
17				44.7	50	56.2
18	44	63	88	56.2	63	70.8
19				70.8	80	89.1
20				89.1	100	112
21	88	125	177	112	125	141
22				141	160	178
23				178	200	224
24	177	250	355	224	250	282
25				282	315	355
26				355	400	447
27	355	500	710	447	500	562
28				562	630	708
29				708	800	891
30	710	1,000	1,420	891	1,000	1,122
31				1,122	1,250	1,413
32				1,413	1,600	1,778
33	1,420	2,000	2,840	1,778	2,000	2,239
34				2,239	2,500	2,818
35				2,818	3,150	3,548
36	2,840	4,000	5,680	3,548	4,000	4,467
37				4,467	5,000	5,623
38				5,623	6,300	7,079
39	5,680	8,000	11,360	7,079	8,000	8,913
40				8,913	10,000	11,220
41				11,220	12,500	14,130
42	11,360	16,000	22,720	14,130	16,000	17,780
43				17,780	20,000	22,390

FIGURE 2.5 Vector Representation of Circular Functions



Vector Representation

Sinusoidal waveforms are components of circular motion. In Fig. 2.5 we start with a circle whose center lies at the origin, and draw a radius at some angle θ to the x (horizontal) axis. The angle *theta* can be measured using any convenient fractional part of a circle. One such fraction is 1/360 of the total angle, which defines the unit called a *degree*. Another unit is $1/2\pi$ of the total angle. This quantity is the ratio of the radius to the circumference of a circle and defines the *radian* (about 57.3°). It was one of the Holy Grails of ancient mathematics since it contains the value of π.

In a circle the triangle formed by the radius and its x and y components defines the trigonometric relations for the sine

$$\mathrm{y} = \mathrm{r} \sin\ \theta \tag{2.5}$$

and cosine functions

$$\mathrm{x} = \mathrm{r} \cos\ \theta \tag{2.6}$$

The cosine is the x-axis projection and the sine the y-axis projection of the radius vector. If we were to rotate the coordinate axes counterclockwise a quarter turn, the x axis would become the y axis. This illustrates the simple relationship between the sine and cosine functions

$$\cos\ \theta = \sin\left(\theta + \frac{\pi}{2}\right) \tag{2.7}$$

The Complex Plane

We can also express the radius of the circle as a vector that has x and y components by writing

$$\mathbf{r} = \mathbf{i}\,\mathrm{x} + \mathbf{j}\,\mathrm{y} \tag{2.8}$$

where **i** and **j** are the unit vectors along the x and y axes. If instead we define x as the displacement along the x axis and j y as the displacement along the y axis, then the vector can be written

$$\mathbf{r} = \mathrm{x} + \mathrm{j}\,\mathrm{y} \tag{2.9}$$

We can drop the formal vector notation and just write the components, with the understanding that they represent displacements along different axes that are differentiated by the

presence or absence of the j term.

$$r = x + j\,y \tag{2.10}$$

The factor j has very interesting properties. To construct the element j y, we measure a distance y along the x axis and rotate it 90° counterclockwise so that it ends up aligned with the y axis. Thus the act of multiplying by j, in this space, is equivalent to a 90° rotation. Since two 90° rotations leave the negative of the original vector

$$j^2 = -1 \tag{2.11}$$

and

$$j = \pm\sqrt{-1} \tag{2.12}$$

which defines j as the fundamental complex number. Traditionally, we use the positive value of j.

The Complex Exponential

The system of complex numbers, although nonintuitive at first, yields enormous benefits by simplifying the mathematics of oscillating functions. The exponential function, where the exponent is imaginary, is the critical component of this process. We can link the sinusoidal and exponential functions through their Taylor series expansions

$$\sin\,\theta = \theta - \frac{\theta^3}{3\,!} + \frac{\theta^5}{5\,!} + \cdots \tag{2.13}$$

and

$$\cos\,\theta = 1 - \frac{\theta^2}{2\,!} + \frac{\theta^4}{4\,!} + \cdots \tag{2.14}$$

and examine the series expansion for the combination $\cos\theta + j\sin\theta$

$$\cos\,\theta + j\,\sin\,\theta = 1 + j\,\theta - \frac{\theta^2}{2\,!} - j\,\frac{\theta^3}{3\,!} + \frac{\theta^4}{4\,!} + \cdots \tag{2.15}$$

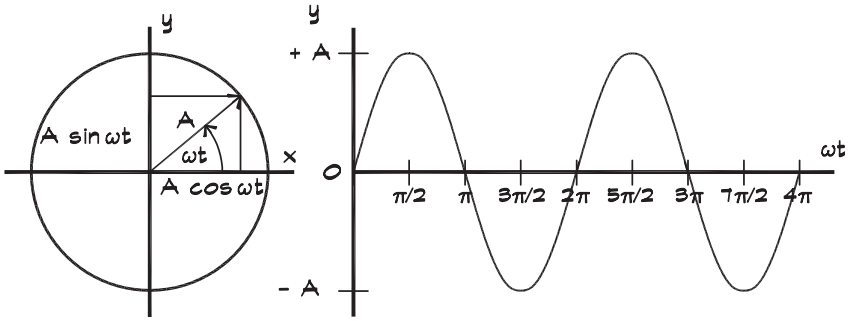
which can be rewritten as

$$\cos\,\theta + j\sin\,\theta = 1 + j\,\theta + \frac{(j\,\theta)^2}{2\,!} + \frac{(j\,\theta)^3}{3\,!} + \frac{(j\,\theta)^4}{4\,!} + \cdots \tag{2.16}$$

This sequence is also the series expansion for the exponential function $e^{j\,\theta}$, and thus we obtain the remarkable relationship originally discovered by Leonhard Euler in 1748

$$e^{j\,\theta} = \cos\,\theta + j\sin\,\theta \tag{2.17}$$

FIGURE 2.6 Rotating Vector Representation of Harmonic Motion



Using the geometry in Fig. 2.6 we see that the exponential function is another way of representing the radius vector in the complex plane. Multiplication by the exponential function generates a rotation of a vector, represented by a complex number, through the angle θ.

Radial Frequency

If the angle θ increases with time at a steady rate, as in Fig. 2.6, according to the relationship

$$\theta = \omega\, \mathrm{t} + \phi \tag{2.18}$$

the radius vector spins around counterclockwise from some beginning angular position ϕ (called the *initial phase*). The rate at which it spins is the radial frequency ω, which is the angle θ divided by the time t, starting at $\phi = 0$. Omega (ω) has units of radians per second. As the vector rotates around the circle, it passes through vertical ($\theta = \pi/2$) and then back to the horizontal ($\theta = \pi$) . When it is pointed straight down, θ is $3\,\pi/2$, and when it has made a full circle, then θ is $2\,\pi$ or zero again.

The real part of the vector is a cosine function

$$\mathrm{x} = \mathrm{A} \cos\left(\omega\, \mathrm{t} + \phi\right) \tag{2.19}$$

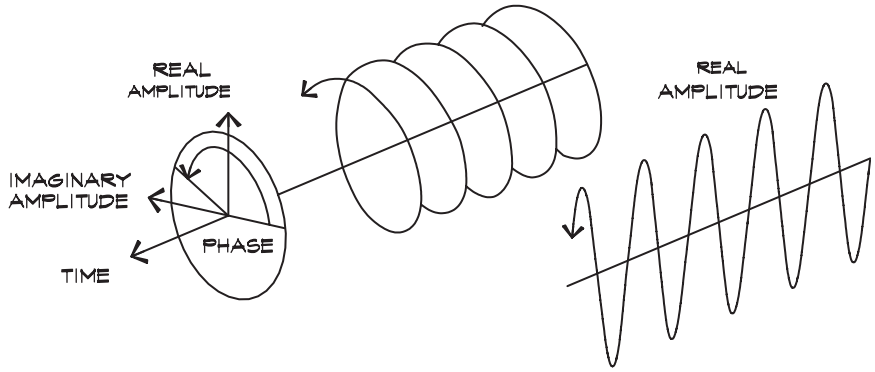
where x, which is the value of the function at any time t, is dependent on the amplitude A, the radial frequency ω, the time t, and the initial phase angle ϕ. Its values vary from $-$A to $+$A and repeat every $2\,\pi$ radians.

Since there are $2\,\pi$ radians per complete rotation, the frequency of oscillation is

$$\mathrm{f} = \frac{\omega}{2\,\pi} \tag{2.20}$$

where f = frequency (Hz)
ω = radial frequency (rad / s)

FIGURE 2.7 Sine Wave in Time and Phase Space



It is good practice to check an equation's units for consistency.

$$\text{frequency} = \text{cycles/sec} = \frac{(\text{radians/sec})}{(\text{radians/cycle})} \tag{2.21}$$

Figure 2.7 shows another way of looking at the time behavior of a rotating vector. It can be thought of as an auger boring its way through phase space. If we look at the auger from the side, we see the sinusoidal trace of the passage of its real amplitude. If we look at it end on, we see the rotation of its radius vector and the circular progression of its phase angle.

Changes in Phase

If a second waveform is drawn on our graph in Fig. 2.8 immediately below the first, we can compare the two by examining their values at any particular time. If they have the same frequency, their peaks and valleys will occur at the same intervals. If, in addition, their peaks occur at the same time, they are said to be in phase, and if not, they are out of phase. A difference in phase is illustrated by a movement of one waveform relative to the other in space or time. For example, a $\pi/2$ radian (90°) phase shift slides the second wave to the right in time, so that its zero crossing is aligned with the peak of the first wave. The second wave is then a sine function, as we found in Eq. 2.6.

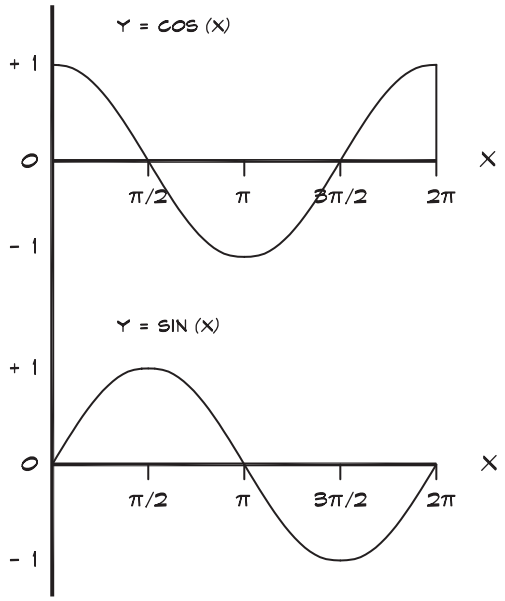
2.3 SUPERPOSITION OF WAVES

Linear Superposition

Sometimes a sound is a pure sinusoidal tone, but more often it is a combination of many tones. Even the simple dial tone on a telephone is the sum of two single frequency tones, 350 and 440 Hz. Our daily acoustical environment is quite complicated, with a myriad of sounds striking our ear drums at any one time. One reason we can interpret these sounds is that they add together in a linear way without creating appreciable distortion.

In architectural acoustics, the wave motions we encounter are generally linear; the displacements are small and forces and displacements can be related by a constant. Algebraically it is an equation called Hooke's law, which when plotted yields a straight line—hence the term linear. When several waves occur simultaneously, the total pressure or displacement amplitude is the sum of their values at any one time. This behavior is referred

FIGURE 2.8 Two Sinusoids 90° Out of Phase



to as a linear superposition of waves and is most useful, since it means that we can construct quite complicated periodic wave shapes by adding up contributions from many different sine and cosine functions.

Figure 2.9 shows an example of the addition of two waves having the same frequency but a different phase. The result is still a simple sinusoidal function, but the amplitude depends on the phase relationship between the two signals. If the two waves are

$$x_1 = A_1 \cos(\omega t + \phi_1) \tag{2.22}$$

and

$$x_2 = A_2 \cos(\omega t + \phi_2) \tag{2.23}$$

FIGURE 2.9 The Resultant of Two Complex Vectors of Equal Frequency

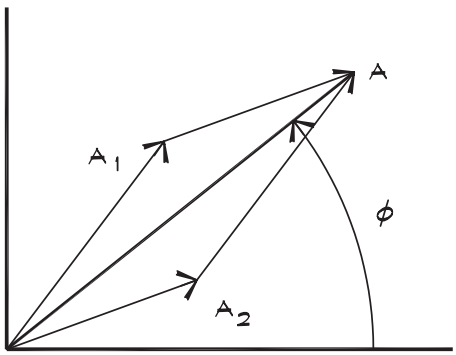
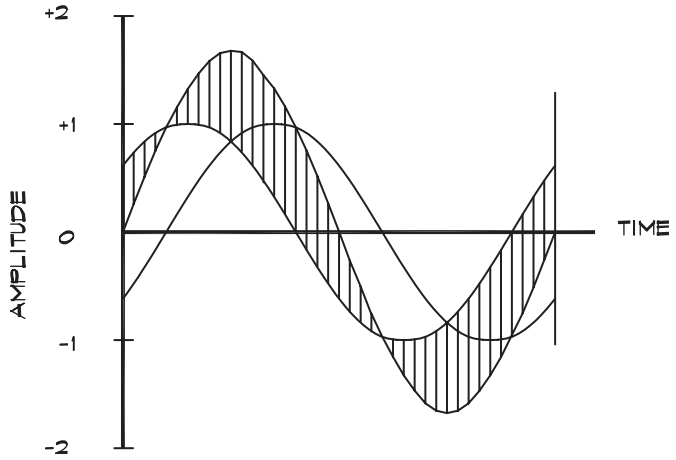


FIGURE 2.10 Sum of Two Sine Waves Having the Same Frequency but Different Phase



Adding the two together yields

$$x_1 + x_2 = A_1 \cos(\omega t + \phi_1) + A_2 \cos(\omega t + \phi_2) \tag{2.24}$$

The combination of these two waves can be written as a single wave.

$$x = A \cos(\omega t + \phi) \tag{2.25}$$

Figure 2.9 shows how the overall amplitude is determined. The first radius vector drawn from the origin and then a second wave is introduced. Its rotation vector is attached to the end of the first vector. If the two are in phase, the composite vector is a single straight line, and the amplitude is the arithmetic sum of $A_1 + A_2$. When there is a phase difference, and the second vector makes an angle ϕ_2 to the horizontal, the resulting amplitude can be calculated using a bit of geometry

$$A = \sqrt{(A_1 \cos\phi_1 + A_2 \cos\phi_2)^2 + (A_1 \sin\phi_1 + A_2 \sin\phi_2)^2} \tag{2.26}$$

and the overall phase angle for the amplitude vector A is

$$\tan\phi = \frac{A_1 \sin\phi_1 + A_2 \sin\phi_2}{A_1 \cos\phi_1 + A_2 \cos\phi_2} \tag{2.27}$$

Thus superimposed waves combine in a purely additive way. We could have added the wave forms on a point-by-point basis (Fig. 2.10) to obtain the same results, but the mathematical result is much more general and useful.

Beats

When two waves having different frequencies are superimposed, there is no one constant phase difference between them. If they start with some initial phase difference, it quickly becomes meaningless as the radius vectors precess at different rates (Fig. 2.11).

FIGURE 2.11 Two Complex Vectors (Feynman et al., 1989)

The resultant of two complex vectors of unequal frequencies, as seen in the frame of reference of one vector. The composite is shown for positions 1, 5, and 9.

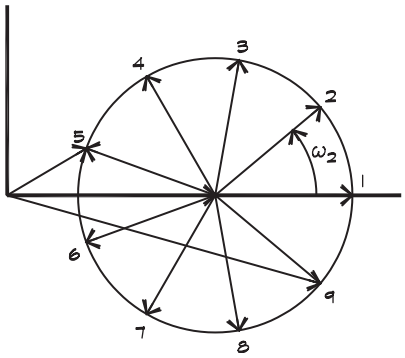
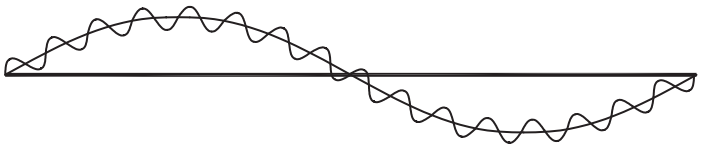


FIGURE 2.12 The Sum of Two Sine Waves with Widely Differing Frequencies



If they both start at zero, then

$$x_1 = A_1 \cos(\omega_1 t) \tag{2.28}$$

and

$$x_2 = A_2 \cos(\omega_2 t) \tag{2.29}$$

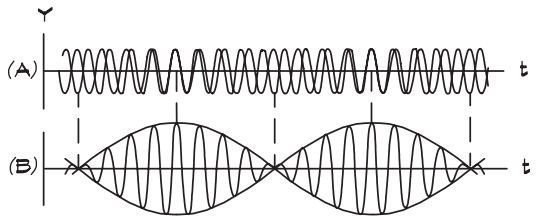
The combination of these two signals is shown in Fig. 2.12. Here the two frequencies are relatively far apart and the higher frequency signal seems to ride on top of the lower frequency. When the amplitudes are the same, the sum of the two waves is[1]

$$x = 2A \cos\left[\frac{(\omega_1 - \omega_2)}{2}\right] \cos\left[\frac{(\omega_1 + \omega_2)}{2}\right] \tag{2.30}$$

If the two frequencies are close together, a phenomenon known as beats occurs. Since one-half the difference frequency is small, it modulates the amplitude of one-half the sum frequency. Figure 2.13 shows this effect. We hear the increase and decrease in signal strength of sound, which is sometimes more annoying than a continuous sound. In practice, beats

[1] The following trigonometric functions were used:
$\cos(\theta + \varphi) = \cos\theta \cos\varphi - \sin\theta \sin\varphi$
$\cos(\theta - \varphi) = \cos\theta \cos\varphi + \sin\theta \sin\varphi$

FIGURE 2.13 The Phenomenon of Beats



are encountered when two fans or pumps, nominally driven at the same rpm, are located physically close together, sometimes feeding the same duct or pipe in a building. The sound waxes and wanes in a regular pattern. If the two sources have frequencies that vary only slightly, the phenomenon can extend over periods of several minutes or more.

2.4 SOUND WAVES

Pressure Fluctuations

A sound wave is a longitudinal pressure fluctuation that moves through an elastic medium. It is called longitudinal because the particle motion is in the same direction as the wave propagation. If the displacement is at right angles to the direction of propagation, as is the case with a stretched string, the wave is called transverse. The medium can be a gas, liquid, or solid, though in our everyday experience we most frequently hear sounds transmitted through the air. Our ears drums are set into motion by these minute changes in pressure and they in turn help create the electrical impulses in the brain that are interpreted as sound. The ancient conundrum of whether a tree falling in a forest produces a sound, when no one hears it, is really only an etymological problem. A sound is produced because there is a pressure wave, but a noise, which requires a subjective judgment and thus a listener, is not.

Sound Generation

All sound is produced by the motion of a source. When a piston, such as a loudspeaker, moves into a volume of air, it produces a local area of density and pressure that is slightly higher than the average density and pressure. This new condition propagates throughout the surrounding space and can be detected by the ear or by a microphone.

When the piston displacement is very small (less than the mean free path between molecular collisions), the molecules absorb the motion without hitting other molecules or transferring energy to them and there is no sound. Likewise if the source moves very slowly, air flows gently around it, continuously equalizing the pressure, and again no sound is created (Ingard, 1994). However, if the motion of the piston is large and sufficiently rapid that there is not enough time for flow to occur, the movement forces nearby molecules together, locally compressing the air and producing a region of higher pressure. What creates sound is the motion of an object that is large enough and fast enough that it induces a localized compression of the gas.

Air molecules that are compressed by the piston rush away from the high-pressure area and carry this additional momentum to the adjacent molecules. If the piston moves back and forth a wave is propagated by small out-and-back movements of each successive volume

element in the direction of propagation, which transfer energy through alternations of high pressure and low velocity with low pressure and high velocity. It is the material properties of mass and elasticity that ensure the propagation of the wave.

As a wave propagates through a medium such as air, the particles oscillate back and forth when the wave passes. We can write an equation for the functional behavior of the displacement y of a small volume of air away from its equilibrium position, caused by a wave moving along the positive x axis (to the right) at some velocity c.

$$y = f(x - c\,t) \tag{2.31}$$

Implicit in this equation is the notion that the displacement, or any other property of the wave, will be the same for a given value of $(x - c\,t)$. If the wave is sinusoidal then

$$y = A \sin [k\,(x - c\,t)] \tag{2.32}$$

where k is called the wave number and has units of radians per length. By comparison to Eq. 2.19 the term (k c) is equal to the *radial frequency* omega.

$$k = \frac{2\,\pi}{\lambda} = \frac{\omega}{c} \tag{2.33}$$

Wavelength of Sound

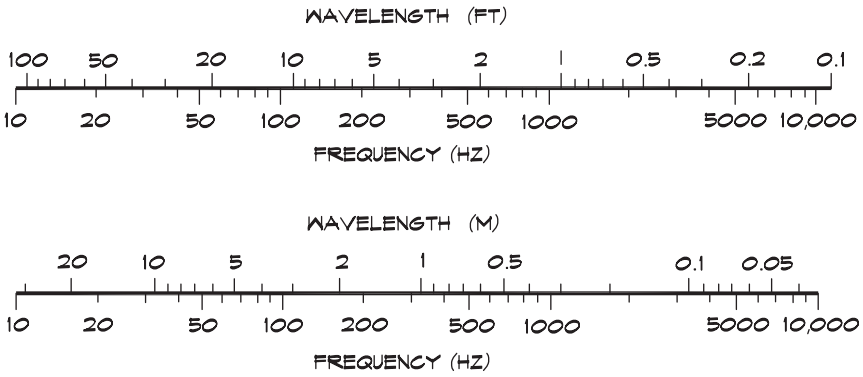
The wavelength of a sound wave is a particularly important measure. Much of the behavior of a sound wave relates to the wavelength, so that it becomes the scale by which we judge the physical size of objects. For example, sound will scatter (bounce) off a flat object that is several wavelengths long in a specular (mirror-like) manner. If the object is much smaller than a wavelength, the sound will simply flow around it as if it were not there. If we observe the behavior of water waves we can clearly see this behavior. Ocean waves will pass by small rocks in their path with little change, but will reflect off a long breakwater or similar barrier.

Figure 2.14 shows typical values of the wavelength of sound in air at various frequencies. At 1000 Hz, which is in the middle of the speech frequency range, the wavelength is about 0.3 m (1 ft) while for the lowest note on the piano the wavelength is about 13 m (42 ft). The lowest note on a large pipe organ might be produced by a 10 m (32 ft) pipe that is half the wavelength of the note. The highest frequency audible to humans is about 20,000 Hz and has a wavelength of around half an inch. Bats, which use echolocation to find their prey, must transmit frequencies as high as 100,000 Hz to scatter off a 2 mm (0.1 in) mosquito.

Velocity of Sound

The mathematical description of the changes in pressure and density induced by a sound wave, which is called the wave equation, requires that certain assumptions be made about the medium. In general we examine an element of volume (say a cube) small enough to smoothly represent the local changes in pressure and density, but large enough to contain very many molecules. When we mathematically describe physical phenomena created by a sound wave, we are talking about the average properties associated with such a small volume element.

FIGURE 2.14 Wavelength vs Frequency in Air at 20° C (68° F) (Harris, 1991)



Let us construct (following Halliday and Resnick, 1966), a one-dimensional tube and set a piston into motion with a short stroke that moves to the right and then stops. The compressed area will move away from the piston with a velocity c. In order to study the pulse's behavior it is convenient to ride along with it. Then the fluid appears to be moving to the left at the sound velocity c. As the fluid stream approaches our pulse, it encounters a region of higher pressure and is decelerated to some velocity $c - \Delta c$. At the back (left) end of the pulse, the fluid is accelerated by the pressure differential to its original velocity, c.

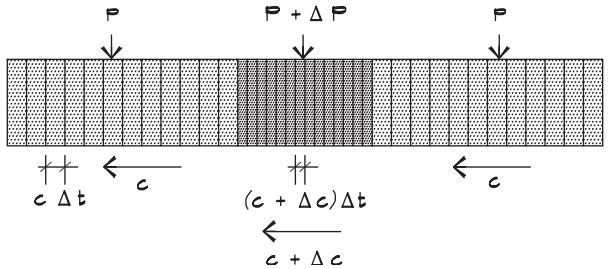
If we examine the behavior of a small element (slice) of fluid such as that shown in Fig. 2.15, as it enters the compressed area, it experiences a force

$$F = (P + \Delta P)S - PS \tag{2.34}$$

where S is the area of the tube. The length of the element just before it encountered our pulse was $c\,\Delta t$, where Δt is the time that it takes for the element to pass a point. The volume of the element is $c\,S\,\Delta t$ and it has mass $\rho\,c\,S\,\Delta t$, where ρ is the density of the fluid outside the pulse zone. When the fluid passes into our compressed area, it experiences a deceleration

FIGURE 2.15 Motion of a Pressure Pulse (Halliday and Resnick, 1966)

A high pressure region moving along a tube from left to right. The frame of reference is riding along with the pulse and thus the fluid appears to be moving from right to left. Note that Δc is negative.



equal to $-\Delta c/\Delta t$. Using Newton's law to relate the force and the acceleration

$$F = m\,a \tag{2.35}$$

which can be written as

$$\Delta P\,S = (\rho\,S\,c\,\Delta t)\,(-\Delta c/\Delta t) \tag{2.36}$$

and rearranged to be

$$\rho\,c^2 = \frac{\Delta P}{(\Delta c/c)} \tag{2.37}$$

Now the fluid that entered the compressed area had a volume $V = S\,c\,\Delta\,t$ and was compressed by an amount $S\,\Delta\,c\,\Delta t = \Delta V$. The change in volume divided by the volume is

$$\frac{\Delta V}{V} = \frac{S\,\Delta c\,\Delta\,t}{S\,c\,\Delta\,t} = \frac{\Delta c}{c} \tag{2.38}$$

so

$$\rho\,c^2 = -\frac{\Delta P}{(\Delta V/V)} \tag{2.39}$$

Thus, we have related the velocity of sound to the physical properties of a fluid. The right-hand side of Eq. 2.39 is a measurable quantity called the *bulk modulus*, B. Using this symbol the velocity of sound is

$$c = \sqrt{\frac{B}{\rho}} \tag{2.40}$$

where c = velocity of sound (m/s)
B = bulk modulus of the medium (Pa)
ρ = density of the medium (kg/m^3)
which for air = 1.21 kg/m^3

The bulk modulus can be measured or can be calculated from an equation of state, which relates the behavior of the pressure, density, and temperature in a gas. In a sound wave, changes in pressure and density happen so quickly that there is little time for heat transfer to take place. Processes thus constrained are called *adiabatic*, meaning no heat flow. The appropriate form of the equation of state for air under these conditions is

$$P\,V^{\gamma} = \text{constant} \tag{2.41}$$

where P = equilibrium (atmospheric) pressure (Pa)
V = equilibrium volume (m^3)
γ = ratio of specific heats (1.4 in air)

Under adiabatic conditions the bulk modulus is γ P, so the speed of sound is

$$c = \sqrt{\gamma \, P/\rho_0} \tag{2.42}$$

Using the relationship known as Boyle's Law ($P V = \mu R T$ where μ is the number of moles of the gas and R = 8.314 joules/mole °K is the gas constant), the velocity of sound in air (which in this text is given the symbol c_0) can be shown to be

$$c_0 = 20.05\sqrt{T_C + 273.2} \tag{2.43}$$

where T_C is the temperature in degrees centigrade. In FP (foot-pound) units the result is

$$c_0 = 49.03\sqrt{T_F + 459.7} \tag{2.44}$$

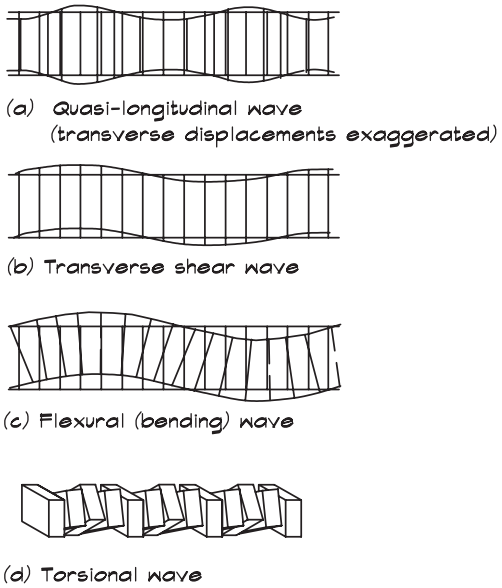
where T_F is the temperature in degrees Fahrenheit.

Table 2.2 shows the velocity of longitudinal waves for various materials. It turns out that the velocities in gasses are relatively close to the velocity of molecular motion due to thermal excitation. This is a reasonable result since the sound pressure changes are transmitted by the movement of molecules.

TABLE 2.2 Speed of Sound in Various Materials (Beranek and Ver, 1992; Kinsler and Frey, 1962)

Material	Density	Speed of Sound (Longitudinal)	
	(kg/m^3)	(m/s)	(ft/s)
Air @ 0° C	**1.293**	**331**	**1086**
Air @ 20° C	**1.21**	**344**	**1128**
Hydrogen @ 0° C	**0.09**	**1286**	**4220**
Oxygen @ 0° C	**1.43**	**317**	**1040**
Steam @ 100° C	**0.6**	**405**	**1328**
Water @ 15° C	**998**	**1450**	**4756**
Lead	**11300**	**1230**	**4034**
Aluminum	**2700**	**5100**	**16700**
Copper	**8900**	**3560**	**11700**
Iron (Bar)	**7700**	**5130**	**16800**
Steel (Bar)	**7700**	**5050**	**16600**
Glass (Rod)	**2500**	**5200**	**17000**
Oak (Bulk)	**720**	**4000**	**13100**
Pine (Bulk)	**450**	**3500**	**11500**
Fir Timber	**550**	**3800**	**12500**
Concrete (Dense)	**2300**	**3400**	**11200**
Gypsum board (1/2" to 2")	**650**	**6800**	**22300**
Cork	**240**	**500**	**1640**
Granite	**—**	**6000**	**19700**
Vulcanized rubber	**1100**	**54**	**177**

FIGURE 2.16 Shapes of Various Wave Types



Waves in Other Materials

Sound waves in gasses are only longitudinal, since a gas does not support shear or bending. Solid materials, which are bound tightly together, can support more types of wave motion than can a gas or liquid, including shear, torsion, bending, and Rayleigh waves. Figure 2.16 illustrates these various types of wave motion and Table 2.3 lists the formulas for their velocities of propagation. In a later chapter we will discuss some of the effects of flexural (bending) and shear-wave motions in solid plates. Rayleigh waves are a combination of compression and shear waves, which are formed on the surface of solids. They are most commonly encountered in earthquakes when a compression wave, produced at the center of a fault, propagates to the earth's surface and then travels along the surface of the ground as a Rayleigh wave.

2.5 ACOUSTICAL PROPERTIES

Impedance

The acoustical *impedance*, which is a measure of the resistance to motion at a given point, is one of the most important properties of a material. A substance such as air has a low characteristic impedance, a concrete slab has a high impedance. Although there are several slightly different definitions of impedance, the *specific acoustic impedance*, which is the most frequently encountered in architectural acoustics, is defined as the ratio of the

TABLE 2.3 Types of Vibrational Waves and Their Velocities

Compressional

Gas	Liquid	Infinite Solid	Solid Bar
$\sqrt{\frac{\gamma\, P}{\rho}}$	$\sqrt{\frac{B}{\rho}}$	$\sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$	$\sqrt{\frac{E}{\rho}}$

Shear

String (Area S)	Solid
$\sqrt{\frac{T}{S\,\rho}}$	$\sqrt{\frac{E}{2\,\rho(1+\nu)}}$

Torsional

Bar
$\sqrt{\frac{E\,K_B}{2\,\rho I\,(1+\nu)}}$

Bending

Rectangular Bar	Plate (Thickness – h)
$\left[\frac{E\,h^2\,\omega^2}{12\,\rho}\right]^{1/4}$	$\left[\frac{E\,h^2\,\omega^2}{12\,\rho(1-\upsilon^2)}\right]^{1/4}$

Rayleigh

Surface of a Solid
$0.385\sqrt{\frac{E\,(2.6+\upsilon)}{\rho\,(1+\upsilon)}}$

where **P = equilibrium pressure (Pa)**
atmospheric pressure = 1.01 × 10^5 Pa
γ **= ratio of specific heats (about 1.4 for gases)**
B = isentropic bulk modulus (Pa)
K_B **= torsional stiffness (m^4)**
I = moment of inertia (m^4)
ρ **= mass density (kg / m^3)**
E = Young's modulus of elasticity (N / m^2)
ν **= Poisson's ratio ≅ 0.3 for structural materials and ≅ 0.5 for rubber-like materials**
T = tension (N)
ω **= angular frequency (rad / s)**

sound pressure to the associated *particle velocity* at a point

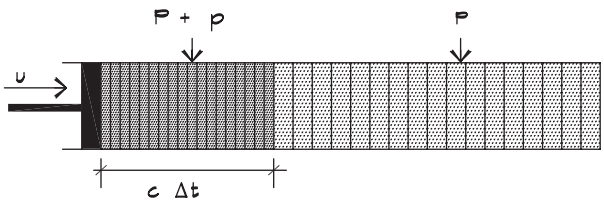
$$z = \frac{p}{u} \tag{2.45}$$

where z = specific acoustic impedance (N s / m^3)
p = sound pressure (Pa)
u = acoustic particle velocity (m / s)

The specific impedance of a gas can be determined by examining a simple example (Ingard, 1994). We construct a hypothetical one-dimensional tube with a piston in one end, as shown in Fig. 2.17. We push the piston into the tube at some steady velocity, u. After a time Δ t, there will be a region of the fluid in front of the piston that is moving at the piston velocity. The information that the piston is moving is conveyed to the gas in the tube at the speed of sound. The length of the region that is aware of this movement is the velocity of sound, c, times the time Δ t, and beyond this point the fluid is quiescent. The fluid in the tube has acquired a momentum (mass times velocity) of $(S\rho\, c\, \Delta\, t)(u)$, where ρ is the mass of density of the fluid, in a time Δ t. Newton's Law tells us that the force is the rate change

FIGURE 2.17 Progression of a Pressure Pulse

A region of high pressure generated by a piston moving to the right at velocity u. The information is transmitted to the fluid at the sound velocity c.



of momentum so

$$p\,S = (S\,\rho\,c)\,u \tag{2.46}$$

The specific acoustic impedance of the fluid is

$$z = \frac{p}{u} = \rho\,c \tag{2.47}$$

where z = specific acoustic impedance (N s / m^3 or mks rayls)
ρ = bulk density of the medium (kg / m^3)
c = speed of sound (m / s)

The dimensions of impedance are known as rayls (in mks or cgs units) to honor John William Strutt, Baron Rayleigh. The value of the impedance frequently is used to characterize the conducting medium and is called the characteristic impedance. For air at room temperature it is about 412 mks or 41 cgs rayls.

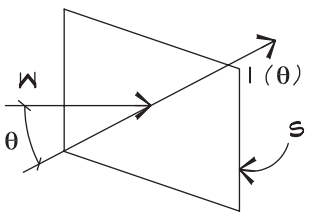
Intensity

Another important acoustical parameter is the measure of the energy propagating through a given area during a given time. This quantity is the *intensity*, shown in Fig. 2.18. For a plane wave it is defined as the acoustic power passing through an area in the direction of the surface normal

$$I(\theta) = \frac{E\,\cos(\theta)}{T\,S} = \frac{W\,\cos(\theta)}{S} \tag{2.48}$$

FIGURE 2.18 Intensity of a Plane Wave

The intensity, I (θ), of a plane wave is the power, W, passing through an area, S, in a direction normal to the plane of the area.



where E = energy contained in the sound wave (N m / s)
W = sound power (W)
I (θ) = intensity (W / m^2) passing through an area in the direction of its normal
S = measurement area (m^2)
T = period of the wave (s)
θ = angle between the direction of propagation and the area normal

The maximum intensity, I, is obtained when the direction of propagation coincides with the normal to the planar surface, when the angle $\theta = 0$.

$$\mathrm{I} = \frac{\mathrm{W}}{\mathrm{S}} \tag{2.49}$$

Plane waves are the most commonly analyzed waveform because the mathematics are simple and the form ubiquitous. A wave is considered planar when its properties do not change in the plane whose normal is the direction of propagation. Intensity is a vector quantity. Its direction is defined by the direction of the normal of the measurement area. When the normal is oriented along the direction of propagation of the sound wave, the intensity has its maximum value, which is not a vector quantity.

Sound power is the sound energy being emitted by a source each cycle. The energy, which is the mechanical work done by a wave, is the force moving through a distance

$$\mathrm{E} = \mathrm{p\ S\ dx} \tag{2.50}$$

where p is the root-mean-square acoustic pressure, and S is the area. The power, W, is the rate of energy flow so

$$\mathrm{W} = \frac{\mathrm{p\ S\ d\ x}}{\mathrm{d\ t}} = \mathrm{p\ S\ u} \tag{2.51}$$

where u is the velocity of a small region of the fluid, and is called the particle velocity. It is not the thermal velocity of individual molecules but rather the velocity of a small volume of fluid caused by the passage of the sound wave. For a plane wave

$$\mathrm{I} = \mathrm{p\ u} \tag{2.52}$$

where I = maximum acoustic intensity (W / m^2)
p = root-mean-square (rms) acoustic pressure (Pa)
u = acoustic rms particle velocity (m / s)

Using the definition of the specific acoustic impedance from Eq. 2.37

$$\mathrm{z} = \frac{\mathrm{p}}{\mathrm{u}} = \rho\ \mathrm{c} \tag{2.53}$$

we can obtain for a plane wave

$$\mathrm{I} = \frac{\mathrm{p}^2}{\rho\ \mathrm{c}} \tag{2.54}$$

where I = maximum acoustic intensity (W/m^2)
p = rms acoustic pressure (Pa)
ρ = bulk density (kg/m^3)
c = velocity of sound (m/s)

The acoustic pressure shown in Eq. 2.44 is the root-mean-square (rms) sound pressure averaged over a cycle

$$p = p_{rms} = \left[\frac{1}{T} \int_0^T P^2 \sin^2 \omega t \, dt \right]^{\frac{1}{2}} = \frac{P}{\sqrt{2}} \tag{2.55}$$

which, for a sine wave, is 0.707 times the maximum value. The average acoustic pressure is zero because its value swings an equal amount above and below normal atmospheric pressure. The energy is not zero but must be obtained by averaging the square of the pressure. Interestingly, the rms pressure of the combination of random waveforms is independent of the phase relationship between the waves.

The intensity (generally taken to be the maximum intensity) is a particularly important property. It is directly measurable using a sound level meter and is audible. It is proportional to power so that when waves are combined, their intensities may be added arithmetically. The combined intensity of several sounds is the simple sum of their individual intensities. The lowest intensity that we are likely to experience is the threshold of human hearing, which is about $10^{-12} W/m^2$. A normal conversation between two people might take place at about $10^{-6} W/m^2$ and a jet aircraft could produce $1 W/m^2$. Thus the acoustic intensities encountered in daily life span a very large range, nearly 12 orders of magnitude. Dealing with numbers of this size is cumbersome, and has lead to the adoption of the decibel notation as a convenience.

Energy Density

In certain instances, the *energy density* contained within a region of space is of interest. For a plane wave if a certain power passes through an area in a given time, the volume enclosing the energy is the area times the distance the sound has traveled, or c t. The energy density is the total energy contained within the volume divided by the volume

$$D = \frac{E}{S\,c\,t} = \frac{W}{S\,c} = \frac{p^2}{\rho\,c^2} \tag{2.56}$$

2.6 LEVELS

Sound Levels — Decibels

Since the range of intensities is so large, the common practice is to express values in terms of *levels*. A level is basically a fraction, expressed as 10 times the logarithm of the ratio of two numbers.

$$\text{Level} = 10 \log \left[\frac{\text{Number of interest}}{\text{Reference number}} \right] \tag{2.57}$$

Since a logarithm can be taken of any dimensionless number, and all levels are the logarithm of some fraction, it is useful to think of them as simple fractions. Even when the denominator

TABLE 2.4 Reference Quantities for Sound Levels (Beranek and Ver, 1992)

Level (dB)	Formula	Reference (SI)
Sound Intensity	$L_I = 10 \log (I/I_o)$	$I_o = 10^{-12}$ W/ m^2
Sound Pressure	$L_p = 20 \log (p/p_o)$	$p_o = 20\ \mu$ Pa $= 2\times10^{-5}$ N/ m^2
Sound Power	$L_W = 10 \log (W/ W_o)$	$W_o = 10^{-12}$ W
Sound Exposure	$L_E = 10 \log (E/E_o)$	$E_o = (20\ \mu\text{ Pa})^2$ s $= (2\times10^{-5}\text{ Pa})^2$ s

Note: Decimal multiples are: 10^{-1} = deci (d), 10^{-2} = centi (c), 10^{-3} = milli (m), 10^{-6} = micro (μ), 10^{-9} = nano (*n*), and 10^{-12} = pico (*p*).

has a numeric value of 1, such as 1 second or 1 square meter, there must always be a reference quantity to keep the ratio dimensionless.

The logarithm of a number divided by a reference quantity is given the unit of bels, in honor of Alexander Graham Bell, the inventor of the telephone. The multiplication by 10 has become common practice, in order to achieve numbers that have a convenient size. The quantities thus obtained have units of decibels, which is one tenth of a bel. Typical levels and their reference quantities are shown in Table 2.4. Levels are denoted by a capital L with a subscript that indicates the type of level. For example, the sound power level is shown as L_w, while the sound intensity level would be L_I, and the sound pressure level, L_p.

Recalling that quantities proportional to power or energy can be combined arithmetically we can combine two or more levels by adding their intensities.

$$I_{Total} = I_1 + I_2 + \cdots + I_n \tag{2.58}$$

If we are given the intensity level of a sound, expressed in decibels, then we can find its intensity by using the definition

$$L_I = 10\ \log \frac{I}{I_{ref}} \tag{2.59}$$

and the definition of the antilogarithm

$$\frac{I}{I_{ref}} = 10^{0.1\ L_I} \tag{2.60}$$

When the intensities from several signals are combined the total overall intensity ratio is

$$\frac{I_{Total}}{I_{ref}} = \sum_{i=1}^{n} 10^{0.1\ L_i} \tag{2.61}$$

and the resultant overall level is

$$L_{Total} = 10 \log \frac{I_{Total}}{I_{ref}} = 10\ \log \sum_{i=1}^{n} 10^{0.1\ L_i} \tag{2.62}$$

As an example, we can take two sounds, each producing an intensity level of 70 dB, and ask what the level would be if we combined the two sounds. The problem can be formulated as

$$L_1 = L_2 = 70 \text{ dB} \tag{2.63}$$

which, if combined, would yield

$$L_{1+2} = 10 \log \left[\left(10^7 + 10^7 \right) \right] = 73 \text{ dB} \tag{2.64}$$

Thus when two levels of equal value are combined the resultant level is 3 dB greater than the original level. By doing similar calculations we learn that when two widely varying levels are combined the result is nearly equal to the larger level. For example, if two levels differ by 6 dB, the combination is about 1 dB higher than the larger level. If the two differ by 10 or more the result is essentially the same as the larger level.

When there are a number of equal sources, the combination process can be simplified

$$L_{\text{Total}} = L_i + 10 \log n \tag{2.65}$$

where L_i is the level produced by one source and n is the total number of like sources.

Sound Pressure Level

The *sound pressure level* is the most commonly used indicator of the acoustic wave strength. It correlates well with human perception of loudness and is measured easily with relatively inexpensive instrumentation. A compilation of the sound pressure levels generated by representative sources is given in Table 2.5 at the location or distance indicated.

The reference sound pressure, like that of the intensity, is set to the threshold of human hearing at about 1000 Hz for a young person. When the sound pressure is equal to the reference pressure the resultant level is 0 dB. The sound pressure level is defined as

$$L_p = 10 \log \frac{p^2}{p_{\text{ref}}^2} \tag{2.66}$$

where p = root-mean-square sound pressure (Pa)
p_{ref} = reference pressure, 2×10^{-5} Pa

Since the intensity is proportional to the square of the sound pressure as shown in Eq. 2.44 the intensity level and the sound pressure level are almost equal, differing only by a small number due to the actual value versus the reference value of the air's characteristic impedance. This fact is most useful since we both measure and hear the sound pressure, but we use the intensity to do most of our calculations.

It is relatively straightforward (Beranek and Ver, 1992) to work out the relationship between the sound pressure level and the sound intensity level to calculate the actual difference

$$L_p = L_I + 10 \log \left(\rho_0 c_0 / 400 \right) \tag{2.67}$$

For a typical value of $\rho_0 c_0$ of 412 mks rayls the correction is 0.13 dB, which is ignored in most calculations.

TABLE 2.5 Representative A-Weighted Sound Levels (Peterson and Gross, 1974)

AT A GIVEN DISTANCE	dBA re 20 μN/m^2	ENVIRONMENTAL
	140	
50 HP Siren (100')		
	130	
Jet Takeoff (200')		
	120	
Riveting Machine	110	Casting Shake-Out Area
Cut Off Saw		Disco
	100	Electric Furnace Area
Textile Weaving Plant		
Subway Train (20')	90	Boiler Room
Jack Hammer (50')		Printing Press Plant
Heavey Truck (50')	80	
Freight Train (100')		
Vacuum Cleaner (10')	70	Near Freeway (Auto Traffic)
Speech (3')		Inside Car (50 MPH)
	60	Large Store
Large Transformer (200')		
	50	Light Traffic (100')
		Quiet Residence Exterior
	40	Min Levels Residential Areas in Chicago at Night
		Private Business Office
Soft Whisper (5')	30	
		Studio for Sound Pictures
	20	Studio (Voice Over)
	10	
Threshold of Hearing	0	

Sound Power Level

The strength of an acoustic source is characterized by its *sound power*, expressed in Watts. The sound power is much like the power of a light bulb in that it is a direct characterization of the source strength. Like other acoustic quantities, the sound powers vary greatly, and a *sound power level* is used to compress the range of numbers. The reference power for this level is 10^{-12} Watts. Sound power levels for several sources are shown in Table 2.6.

Sound power levels can be measured by using Eq. 2.49.

$$I = \frac{W}{S} \tag{2.68}$$

TABLE 2.6 Sound Power Levels of Various Sources (Peterson and Gross, 1974)

POWER (Watts)	POWER LEVEL (dB re 10^{-12} Watts)	SOURCE
25 to 40 Million	195	Saturn Rocket
100,000	170	Ram Jet
		Turbo Jet Engine with Afterburner
10,000	160	Turbo Jet Engine, 7000-lb Thrust
1,000	150	4-Propellor Aircraft
100	140	
10	130	75-Piece Orchestra } Peak RMS Levels in 1/8 Second Intervals
		Pipe Organ
		Small Aircraft Engine
1	120	Large Chipping Hammer
0	110	Blaring Radio
		Centrifugal Ventillating Fan (13,000 CFM)
0.1	100	Auto on Highway
0.01	90	Vane Axial Ventillating Fan (1500 CFM)
0.001	80	Voice - Shouting (Average Long-Time RMS)
0.000,01	70	Voice - Conversational Level (Average Long-Time RMS)
0.000,001	60	
0.000,000,1	50	
0.000,000,01	40	
0.000,000,001	30	Voice - Very Soft Whisper

If we divide this equation by the appropriate reference quantities

$$\frac{I}{I_0} = \frac{\left(W/W_0\right)}{\left(S/S_0\right)} \tag{2.69}$$

and take 10 log of each side we get

$$L_I = L_w - 10 \log S \tag{2.70}$$

where S_0 is equal to 1 square meter. Recalling that the sound intensity level and the sound pressure level are approximately equal,

$$L_p = L_w - 10 \log S + K \tag{2.71}$$

where L_w = sound power level (dB re 10^{-12} W)
L_p = sound pressure level (dB re 2×10^{-5} Pa)
S = measurement area (m^2 or ft^2)
$K = 10 \log (\rho_0 c_0/400) + 20 \log(r_0)$
= 0.1 for r in m, or 10.5 for r in ft
r_0 = 1 m for r in m or 3.28 ft for r in ft

The small correction for the difference between the sound intensity level and the sound pressure level, when the area is in square meters, is ignored. When the area S in Eq. 2.68 is in square feet, a conversion factor is applied, which is equal to 10 log of the number of square feet in a square meter or 10.3. We then add in the small factor, which accounts for the difference between sound intensity and sound pressure level.

These formulas give us a convenient way to measure the sound power level of a source by measuring the average sound pressure level over a surface of known area that bounds the source. Perhaps the simplest example is low-frequency sound traveling down a duct or tube having a cross-sectional area, S. The sound pressure levels are measured by moving a microphone across the open area of the duct and by taking the average intensity calculated from these measurements. The overall average sound intensity level is obtained by taking 10 log of the average intensity divided by the reference intensity. By adding a correction for the area the sound power level can be calculated. This method can be used to measure the sound power level of a fan when it radiates into a duct of constant cross section. Product manufacturers provide sound power level data in octave bands, whose center frequencies range from 63 Hz (called the first band) through 8 kHz (called the eighth band). They are the starting point for most HVAC noise calculations.

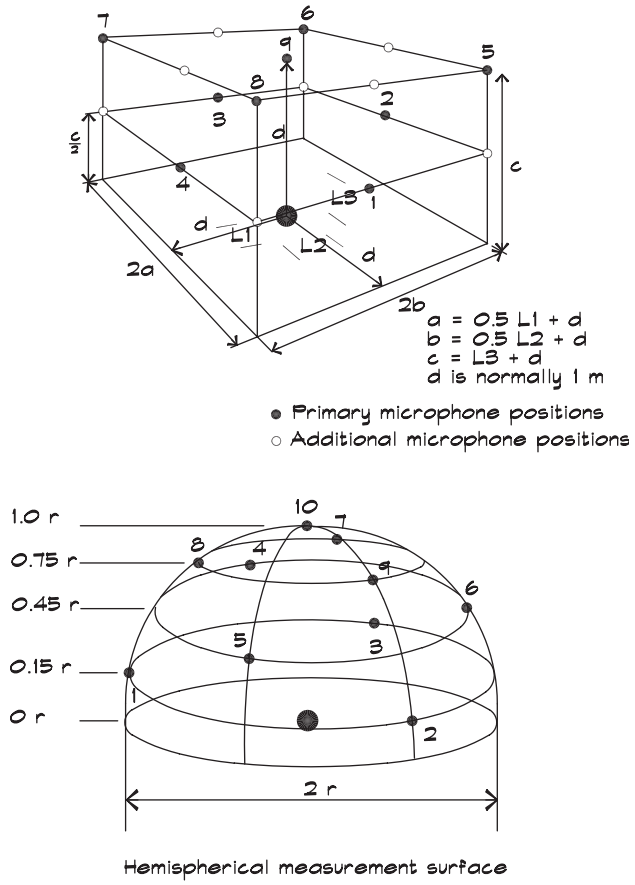
If the sound source is not bounded by a solid surface such as a duct, the area shown in Eq. 2.68 varies according to the position of the measurement. Sound power levels are determined by taking sound pressure level data at points on an imaginary surface, called the *measurement surface*, surrounding the source. The most commonly used configurations are a rectangular box shape or a hemispherical-shaped surface above a hard reflecting plane. The distance between the source and the measurement surface is called the *measurement distance*. For small sources the most common measurement distance is 1 meter. The box or hemisphere is divided into areas and the intensity is measured for each segment.

$$W = \sum_{i=1}^{n} I_i \, S_i \tag{2.72}$$

where W = total sound power (W)
I_i = average intensity over the i th surface (W/m^2)
S_i = area of the i th surface (m^2)
n = total number of surfaces

Measurement locations are set by international standards (ISO 7779), which are shown in Fig. 2.19. The minimum number of microphone positions is nine with additional positions

FIGURE 2.19 Sound Power Measurement Positions on a Parallelepiped or Hemisphere (ISO 7779)



required if the source is long or the noise highly directional. The difference between the highest and lowest measured level must be less than the number of microphone positions. If the source is long enough that the parallelepiped has a side that is more than twice the measurement distance, the additional locations must be used.

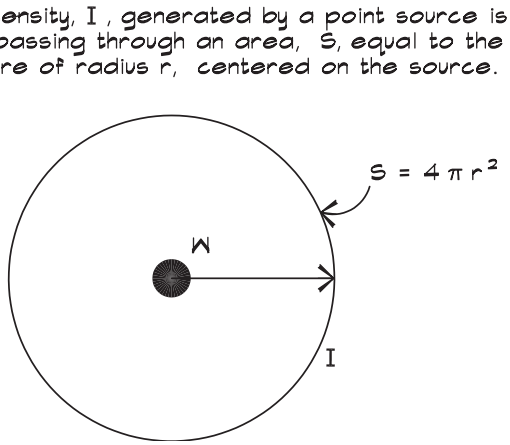
2.7 SOURCE CHARACTERIZATION

Point Sources and Spherical Spreading

For most sources the relationship between the sound power level and the sound pressure level is determined by the increase in the area of the measurement surface as a function of distance. Sources that are small compared with the measurement distance are called *point sources*, not because they are so physically small but because at the measurement distance their size does not influence the behavior of the falloff of the sound field. At these distances the measurement surface is a sphere with its center at the center of the source as shown in Fig. 2.20, with a surface area given by

$$S = 4\,\pi\,r^2 \tag{2.73}$$

FIGURE 2.20 Spherical Spreading of a Point Source



where S = area of the measurement surface (m^2 or ft^2)
r = measurement distance (m or ft)

When Eq. 2.73 holds, the falloff is referred to as *free field* behavior and the power-pressure relationship for a nondirectional source is

$$L_p = L_w + 10 \log \left[\frac{1}{4 \pi r^2} \right] + K \tag{2.74}$$

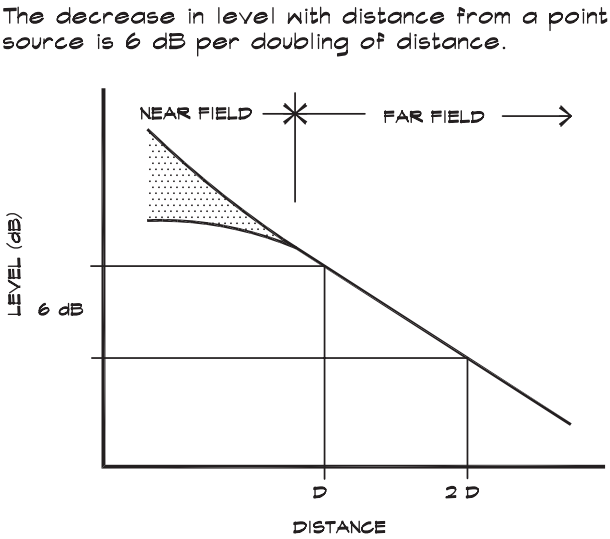
where L_w = sound power level (dB re 10^{-12} W)
L_p = sound pressure level (dB re 2×10^{-5} Pa)
r = measurement distance (m or ft)
$K = 10 \log (\rho_0 c_0 / 400) + 20 \log (r_{ref})$
= 0.1 for r in m, or 10.5 for r in ft (for standard conditions)
r_{ref} = 1 m for r in m or 3.28 ft for r in ft

The designation free field means that sound field is free from any reflections or other influences on its behavior, other than the geometry of spherical spreading of the sound energy. For a given sound power level the sound pressure level decreases 6 dB for every doubling of the measurement distance. Free-field falloff is sometimes described as 6 dB per distance doubling falloff.

Figure 2.21 shows the level versus distance behavior for a point source. If the measurement distance is small compared with the size of the source, where this falloff rate does not hold, the measurement position is in the region of space described as the *near field*. In the near field the source size influences the power-pressure relationship. Occasionally there are nonpropagating sound fields that contribute to the sound pressure levels only in the near field.

For a given source, we can calculate the sound pressure level in the free field at any distance, if we know the level at some other distance. One way to carry out this calculation is to compute the sound power level from one sound pressure level measurement and then to use it to calculate the second level at a new distance. By subtracting the two equations used

FIGURE 2.21 Falloff from a Point Source



to do this calculation we obtain

$$\Delta L_p = 10 \log \frac{r_2^2}{r_1^2} = 20 \log \frac{r_2}{r_1} \tag{2.75}$$

where ΔL_p = change in sound pressure level $(L_1 - L_2)$
r_1 = measurement distance 1 (m or ft)
r_2 = measurement distance 2 (m or ft)

Note that the change in level is positive when $L_1 > L_2$, which occurs when $r_2 > r_1$. As expected, the sound pressure level decreases as the distance from the source increases.

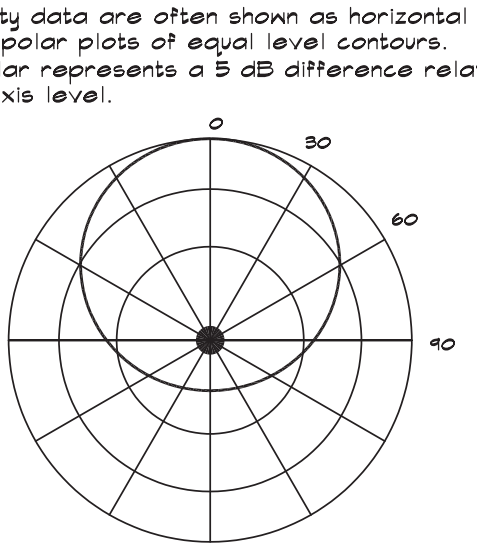
Sensitivity

Although the strength of many sources, particularly mechanical equipment, is characterized by the sound power level, in the audio industry loudspeakers are described by their *sensitivity*. The sensitivity is the sound pressure level measured at a given distance (usually 1 meter) on axis in front of the loudspeaker for an electrical input power of 1 Watt. Sensitivities are measured in octave bands and are published along with the maximum power handling capacity and directivity of the device. The on-axis sound level, expected from a speaker at a given distance, can be calculated from

$$L_p = L_S + 10 \log J - 20 \log \left(\frac{r}{r_{ref}} \right) \tag{2.76}$$

where L_p = measured on axis sound pressure level (dB)
L_S = loudspeaker sensitivity (dB at 1 m for 1 W electrical input)
J = electrical power applied to the loudspeaker (W)
r = measurement distance (m or ft)
r_{ref} = reference distance (m or ft)

FIGURE 2.22 Source Directivity Shown as a Polar Plot



Directionality, Directivity, and Directivity Index

For many sources the sound pressure level at a given distance from its center is not the same in all directions. This property is called *directionality*, and the changes in level with direction of a source are called its *directivity*. The directivity pattern is sometimes illustrated by drawing two- or three-dimensional equal-level contours around it, such as that shown in Fig. 2.22. When these contours are plotted in two planes, a common practice in the description of loudspeakers, they are called horizontal and vertical polar patterns.

The sound power level of a source gives no specific information about the directionality of the source. In determining the sound power level, the sound pressure level is measured at each measurement position, the intensity is calculated, multiplied by the appropriate area weighting, and added to the other data. A highly directional source could have the same sound power level as an omnidirectional source but would produce a very different sound field. The way we account for the difference is by defining a *directivity index*, which is the difference in decibels between the sound pressure level from an omnidirectional source and the measured sound pressure level in a given direction from the real source.

$$\mathrm{D}\,(\theta, \phi) = \mathrm{L_p}(\theta, \phi) - \overline{\mathrm{L}}_\mathrm{p} \tag{2.77}$$

where $\mathrm{D}\,(\theta, \phi)$ = directivity index (gain) for a given direction (dB)
$\mathrm{L_p}(\theta, \phi)$ = sound pressure level for a given direction (dB)
$\overline{\mathrm{L}}_\mathrm{p}$ = sound pressure level averaged over all angles (dB)
θ, ϕ = some specified direction

The directivity index can also be specified in terms of a directivity, which is given the symbol Q for a specific direction

$$\mathrm{D}\,(\theta, \phi) = 10 \log \mathrm{Q}\,(\theta, \phi) \tag{2.78}$$

where $\mathrm{Q}\,(\theta, \phi)$ = directivity for a given direction (θ, ϕ)

The directivity can be expressed in terms of the intensity in a given direction compared with the average intensity

$$Q\ (\theta, \phi) = \frac{I\ (\theta, \phi)}{I_{Ave}} \tag{2.79}$$

The average intensity is given by

$$I_{Ave} = \frac{W}{4\ \pi\ r^2} \tag{2.80}$$

and the intensity in a particular direction by

$$I\ (\theta, \phi) = \frac{Q\ (\theta, \phi)W}{4\ \pi\ r^2} \tag{2.81}$$

When the directivity is included in the relationship between the sound power level and the sound pressure level in a given direction, the result for a point source is

$$L_p(\theta, \phi) = L_w + 10 \log \frac{Q\ (\theta, \phi)}{4\ \pi\ r^2} + K \tag{2.82}$$

In the audio industry the Q of a loudspeaker is understood to mean the on-axis directivity, $Q\ (0, 0) = Q_0$.

The sound power level of a loudspeaker can be calculated from its sensitivity and its Q_0 for any input power J

$$L_w = L_S - 10 \log \frac{Q_0}{4\ \pi\ r^2} + 10 \log J - K \tag{2.83}$$

where L_S = loudspeaker sensitivity (dB at 1 m for 1 W input)
r = standard measurement distance (usually = 1 m)
J = input electrical power (W)

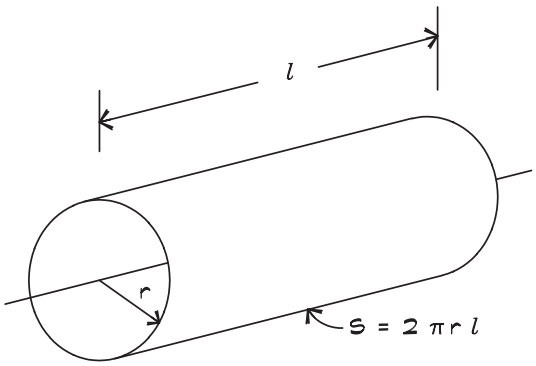
The sound pressure level emitted by the loudspeaker at a given angle can then be calculated from the sound power level

$$L_p = L_w + 10 \log \frac{Q\ (\theta, \phi)}{4\ \pi\ r^2} + K \tag{2.84}$$

where $Q\ (\theta, \phi)$ = loudspeaker directivity for a given direction
$Q\ (\theta, \phi) = Q_0\ Q_{rel}\ (\theta, \phi)$
Q_0 = on - axis directivity
$Q_{rel}\ (\theta, \phi)$ = directivity relative to on - axis
θ, ϕ = latitude and longitude angles with respect to the aim point direction and the horizontal axis of the loudspeaker

Normally $Q_0 \gg 1$ and $Q_{rel}(\theta, \phi) < 1$. These relationships will be discussed in greater detail in Chap. 18.

FIGURE 2.23 Falloff of a Line Source



Line Sources

Line sources are one-dimensional sound sources such as roadways, which extend over a distance that is large compared with the measurement distance. With this geometry the measurement surface is not a sphere but rather a cylinder, as illustrated in Fig. 2.23, with its axis coincident with the line source. Since the geometry is that of a cylinder the surface area (ignoring the ends) is given by the equation

$$S = 2\,\pi\,r\,l \tag{2.85}$$

where S = surface area of the cylinder (m^2 or ft^2)
r = radius of the cylinder (m or ft)
l = length of the cylinder (m or ft)

With a line source, the concept of an overall sound power level is not very useful, since all that matters is the portion of the source closest to the observer. Line sources are characterized by a sound pressure level at a given distance. From this information the sound level can be determined at any other distance.

Assume for a moment that a nondirectional line source of length l emits a given sound power. Then the maximum intensity at a distance r is

$$I = \frac{W}{S} = \frac{W}{2\,\pi\,r\,l} \tag{2.86}$$

and the difference in intensity levels at two different distances can be calculated from the ratio of the two intensities

$$\Delta L = L_1 - L_2 = 10 \log \frac{I_1}{I_{ref}} - 10 \log \frac{I_2}{I_{ref}} \tag{2.87}$$

So for an infinite (very long) line source the change in level with distance is given by

$$\Delta L = 10\ \log \frac{I_1}{I_2} = 10\ \log \frac{r_2}{r_1} \tag{2.88}$$

where ΔL = change in level (dB)
L_1 = sound intensity level at distance r_1 (dB re 10^{-12} W/m^2)
L_2 = sound intensity level at distance r_2 (dB re 10^{-12} W/m^2)
r_1 = distance 1 (m or ft)
r_2 = distance 2 (m or ft)

If we measure the sound pressure level at a distance, r_1, from an unshielded line source, we can use Eq. 2.88 to calculate the difference in level at some new distance r_2. If $r_2 > r_1$ then the change in level is positive—that is, sound level decreases with increasing distance from the source. The falloff rate is gentler with a line source than it is for a point source—3 dB per distance doubling.

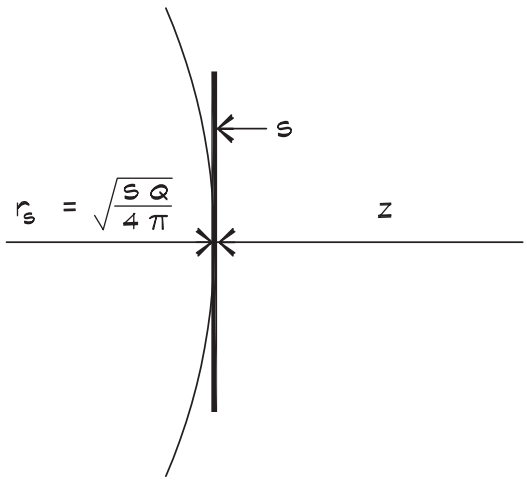
Planar Sources

A *planar source* is a two-dimensional surface that is large compared to the measurement distance and usually, though not always, relatively flat. For purposes of this analysis a planar source is assumed to be incoherent, which is to say that there is no fixed phase relationship among the various points on its surface. From our previous analysis we know that if a surface radiates a certain acoustic power, W, and if that power is uniformly distributed over the surface, then close to the surface the intensity is given by

$$I = \frac{W}{S} \tag{2.89}$$

where S is the area of the surface. We also know that if we are far enough away from the surface, it is small compared to the measurement distance, and it must behave like a point source; the intensity is given by Eq. 2.82. To model (Long, 1987) the behavior in both regions, it is convenient to imagine the planar source shown in Fig. 2.24 as a portion of a

FIGURE 2.24 Falloff from a Planar Source



large sphere that has a radius equal to $\sqrt{\frac{S\,Q}{4\,\pi}}$. Since the measurement distance is taken from the surface of the plane, the distance to the center of the sphere from the measurement point is $z + \sqrt{\frac{S\,Q}{4\,\pi}}$. The intensity is then given by

$$I = \frac{W\,Q}{4\,\pi\left[z + \sqrt{\frac{S\,Q}{4\,\pi}}\right]^2} \tag{2.90}$$

When this equation is written as a level by taking 10 log of both sides

$$L_p = L_w + 10\log\left\{\frac{Q}{4\,\pi\left[z + \sqrt{\frac{S\,Q}{4\,\pi}}\right]^2}\right\} + K \tag{2.91}$$

where L_p = sound pressure level (dB re 2×10^{-5} Pa)
L_w = sound power level (dB re 10^{-12} W)
Q = directivity (dimensionless)
S = area of the radiating surface (m^2 or ft^2)
z = measurement distance from the surface (m or ft)
K = 0.1 (z in m) or 10.5 (z in ft) for standard conditions

Equation 2.91 gives the sound pressure vs sound power relationship for a planar surface at all distances. When a measurement is made close to the surface, the distance z goes to zero, and we obtain Eq. 2.71. When z is large compared to $\sqrt{S\,Q\,/\,4\,\pi}$, the behavior approaches Eq. 2.82. Note that the directivity is meaningless when the receiver is very close to the surface since the concept of the direction to the surface is not well defined.

3 HUMAN PERCEPTION AND REACTION TO SOUND

3.1 HUMAN HEARING MECHANISMS

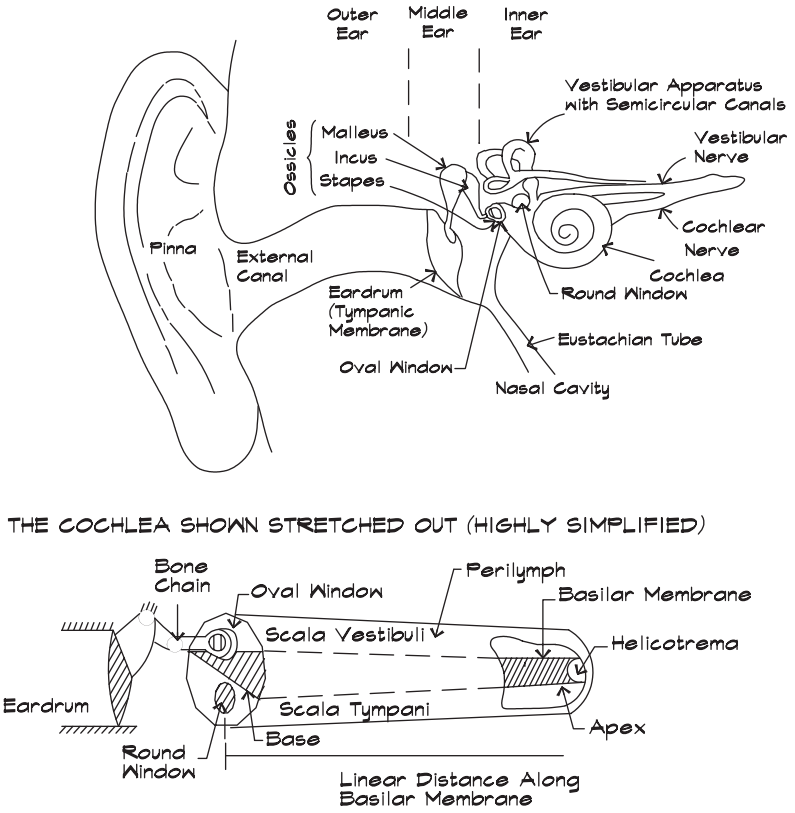
Physiology of the Ear

The human ear is an organ of marvelous sensitivity, complexity, and robustness. For a person with acute hearing, the range of audible sound spans ten octaves, from 20 Hz to 20,000 Hz. The wavelengths corresponding to these frequencies vary from 1.7 centimeters (5/8 inch) to 17 meters (57 feet), a ratio of one thousand. The quietest sound audible to the average human ear, about zero dB at 1000 Hz, corresponds to an acoustic pressure of 20×10^{-6} N/m^2 or Pa. Since atmospheric pressure is about 101,000 Pa (14.7 lb/sq in), it is clear that the ear is responding to extraordinarily small changes in pressure. Even at the threshold of pain, 120 dB, the acoustic pressures are still only about 20 Pa.

The excursion of the ear drum at the threshold of hearing is around 10^{-9} m (4×10^{-7} in) (Kinsler et al., 1982). Most atoms have dimensions of 1 to 2 angstroms (10^{-10} meters) so the ear drum travels a distance of less than 10 atomic diameters at the threshold of hearing. Were our ears only slightly more sensitive, we would hear the constant background noise due to Brownian movement, molecules set into motion by thermal excitation. Indeed, it is thermal motion of the hair cells in the *cochlea* that limits hearing acuity. In very quiet environments the flow of blood in the vessels near the eardrum is plainly audible as a disquieting shushing sound.

The anatomy of the ear, shown in Fig. 3.1, is organized into three parts, termed outer, middle, and inner. The outer and middle ear are air filled, whereas the inner ear is fluid filled. The outer part includes the *pinna*, the fleshy flap of skin that we normally think of as the ear, and a tube known as the *meatus* or auditory canal that conducts sound waves to the *tympanic membrane* or ear drum, separating the outer and middle ear sections. The pinna gathers the sound signals and assists in the localization of the height of a sound source. The 2.7 centimeter (one-inch) long auditory canal acts like a broadband quarter-wavelength tube resonator, whose lowest natural frequency is about 2700 Hz. This helps determine the range of frequencies where the ear is most sensitive—a more or less 3 kHz wide peak centered at about 3400 Hz. The auditory canal resonance increases the sound level at the ear drum around this frequency by about 10 dB above the level at the canal entrance. With the

FIGURE 3.1 A Schematic Representation of the Ear (Flanagan, 1972)



diffraction provided by the pinna and the head, there can be as much as a 15 to 20 dB gain at certain frequencies at the ear drum relative to the free-field level.

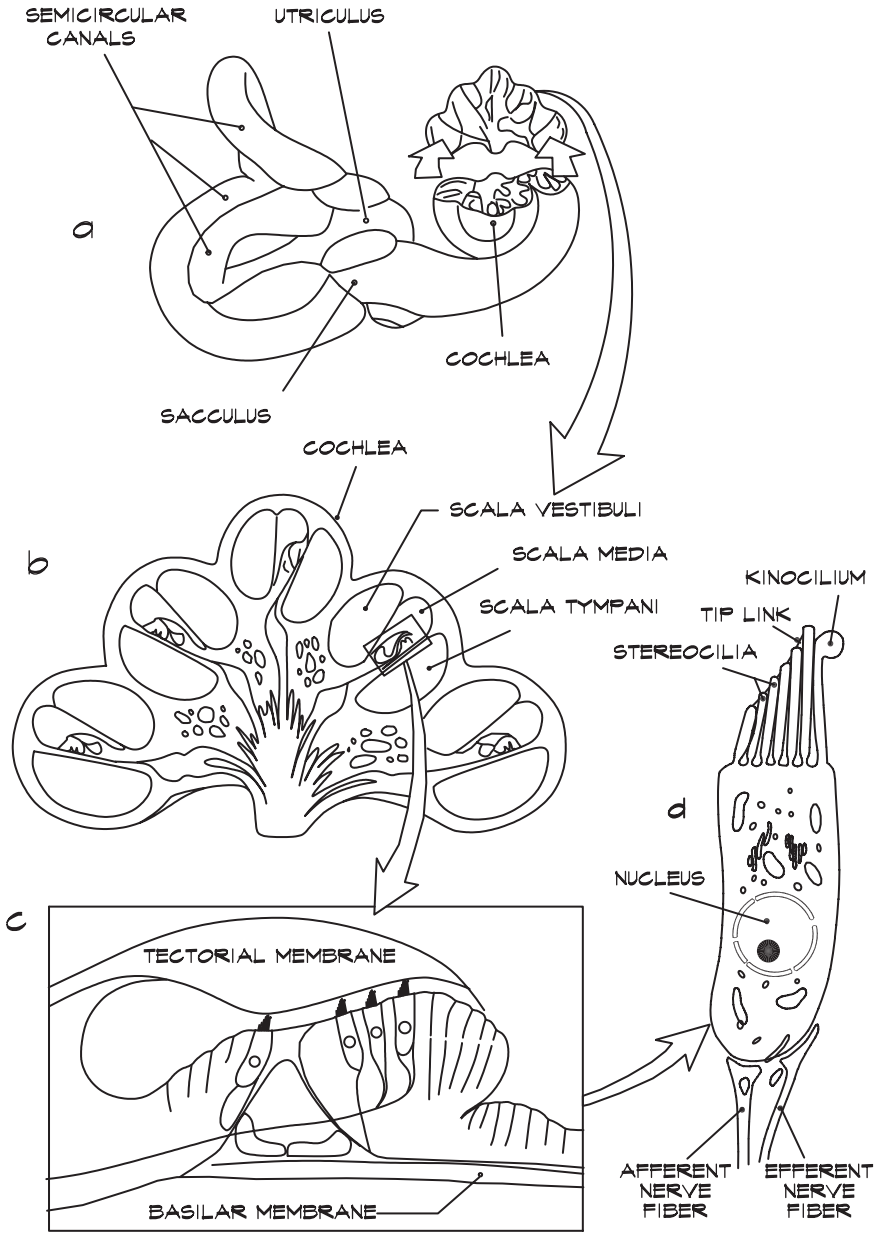
The middle ear is an air-filled cavity about 2 cu cm in volume (about the same as a sugar cube), which contains the mechanisms for the transfer of the motion of the eardrum to the cochlea in the inner ear. The ear drum is a thin conical membrane stretched across the end of the auditory canal. It is not a flat drum head, as might be inferred from its name, but rather a tent-like sheath with its peak pointing inward. Near its center, the eardrum is attached to the malleus bone, which is connected in turn to two other small bones. These three, the *malleus* (hammer), *incus* (anvil), and *stapes* (stirrup) act as a mechanical linkage, which couples the eardrum to the fluid-filled cochlea. The stapes resembles a stirrup with its base pressed up against the *oval window*, a membrane that covers the entrance to the cochlea. Because of the area ratio of the eardrum to that of the oval window (about 20 to 1) and the lever action of the ossicles, which produces another factor of 1.5:1, the middle ear acts as an impedance matching transformer, converting the low-pressure, high-displacement motion of the ear drum into a high-pressure, low-displacement motion of the fluid of the cochlea. Atmospheric pressure in the middle ear is equalized behind the eardrum by venting this area to the throat through the *eustation tube*, which opens when we yawn or swallow.

The motion transfer in the middle ear is not linear but depends on amplitude. An *aural reflex* protects the inner ear from loud noises by tightening the muscles holding the stapes to reduce its excursion at high amplitudes, just as the eye protects itself from bright light by

contracting the pupil. The contraction is involuntary in both cases and seldom is noticed by the individual. Pain is produced at high noise levels when the muscles strain to protect nerve cells. Unfortunately the aural reflex is not completely effective. There is a reaction time of about 0.5 msec so it cannot block sounds having a rapid onset, such as gunshots and impact-generated noise. A second reason is that the muscles cannot contract indefinitely. Under a sustained bombardment of loud noise they grow tired and allow more energy to pass.

The inner ear, shown in Fig. 3.2, contains mechanisms that sense balance and acceleration as well as hearing. Housed in the hard bone of the skull, the inner ear contains five

FIGURE 3.2 Structure of the Inner Ear (Hudspeth and Markin, 1994)



separate receptor organs, each sensitive to a specific type of acceleration, as well as the cochlea, which detects the loudness and frequency content of airborne sound waves. The sacculus and utriculus include about 15,000 and 30,000 hair cells in planar sheets that react to vertical and horizontal linear accelerations respectively. These organs have the capability of encoding a unique signal for an acceleration in any given direction within a plane. Three *semicircular canals* are arranged to sense the orthogonal directions of angular acceleration. Each consists of a fluid-filled tube interrupted by a diaphragm containing about 7000 hair cells. They provide information on the orientation and acceleration of the human head. The bilateral symmetry of the ears gives us not only backup capability but extra information for the decomposition of motions in any direction.

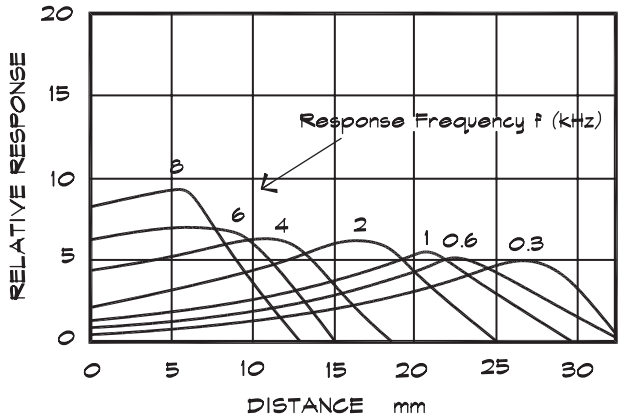
The cochlea is a fluid-filled tube containing the hair cell transducers that sense sound. It is rolled up two and one-half turns like a snail and if we unroll the tube and straighten it out, we would find a narrow cavity 3.5 cm long, about the size and shape of a golf tee scaled down by two-thirds. At its beginning, called the basal end, it is about 0.9 cm in diameter and at the apical end it is about 0.3 cm in diameter. It has two thin membranes running down it near its middle. The thicker membrane is called the *basilar membrane* and divides the cochlea more or less in half, separating the upper gallery (*scala vestibuli*) from the lower gallery (*scala tympani*). Along the membrane lies the *auditory nerve*, which conducts the electrochemical impulses and snakes through a thin sliver of bone called the bony ridge to the brain.

The entrance to the cochlea, in the upper gallery, is the *oval window* at the foot of the stapes. At the upper end of the cochlea near its apex there is a small passageway connecting the upper and lower galleries called the *helicotrema*. At the distal end of the lower gallery near the oval window is another membrane, the *round window*. It acts like the back door to the cochlea, a pressure release surface for fluid impulses traveling along its length and back into the middle ear. The two membranes, the oval window and the round window, seal in the fluid of the cochlea. Otherwise the rest of the cochlea is completely surrounded and protected by bone.

Figure 3.2b shows a cross section of one of the spirals of the cochlea. The upper gallery is separated from a pie-shaped section called the middle gallery (*scala media*) by *Reissner's membrane*. Within this segment and attached to the basilar membrane is the *organ of Corti*, which includes some 16,000 small groups of hair cells (*stereocilia*), arranged in four rows, acting as motion transducers to convert fluid and basilar membrane movement into electrical impulses (Hudspeth and Markin, 1994). The stereocilia are cylindrical rods arranged in a row in order of increasing height and move back and forth as a group in response to pressure waves in the endolymphatic fluid. The hair cells are relatively stiff and only move about a diameter. Through this movement they encode the magnitude and the time passage of the wave as an electrochemical potential, which is sent along to the brain.

Each stereocilia forms a bond between its end and an area on the adjacent higher neighbor much like a spring pulling on a swinging gate (see Fig. 3.2d). When a gate is opened a nerve impulse is triggered and sent to the brain. If the bundle of stereocilia is displaced in the positive direction, toward the high side of the bundle, a greater relative displacement occurs between each stalk and more gates are opened. A negative displacement, towards the short side of the bundle reduces the tension on the biomechanical spring and closes gates. Orthogonal motion results in no change in tension and no change in the signal. The amplitude of the response to sound waves is detected by the number of gate openings and closings and thus the number of impulses sent up the auditory nerve.

FIGURE 3.3 Longitudinal Distance along the Cochlea Showing the Positions of Response Maxima (Hassall and Zeveri, 1979)



As the stereocilia move back and forth they are sometimes stimulated to a degree that pushes them farther than their normal excursion. In these cases a phenomena known as *adaptation* occurs wherein the hair cells acquire a new resting point that is displaced from their original point. The cells find a new operating position and do a recalibration or reattachment of the spring to a gate at a slightly different point on the neighboring cell. Adaptation also suggests a mechanism for hearing loss when hair cells are displaced beyond the point where they can recover due to exposure to loud sounds over a long time period.

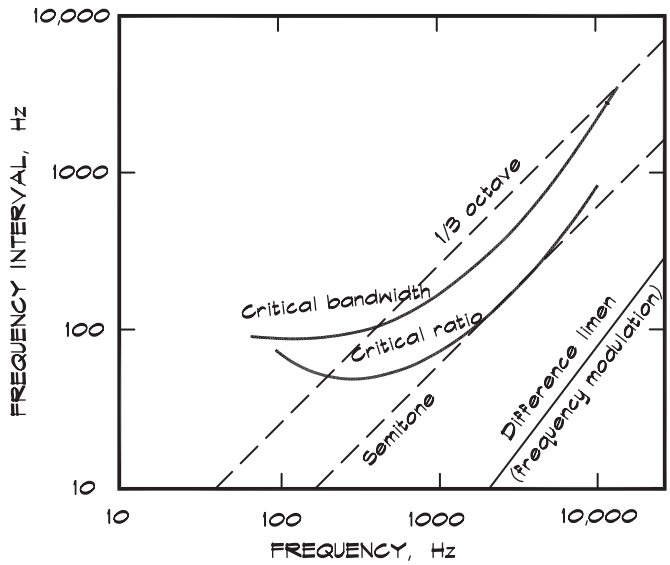
The frequency of the sound is detected by the position of greatest response along the basilar membrane. As a pressure wave moves through the cochlea it induces a ripple motion in the basilar membrane and for each frequency there is a maximum displacement in a certain region. The high frequencies stimulate the area closest to the oval window, while the low frequencies excite the area near the helicotrema. Figure 3.3 illustrates this phenomenon, known as the place theory of pitch detection. It was originated by von Bekesy (1960), and he received a Nobel Prize for his work. The brain can interpret nerve impulses coming from a specific area of the cochlea as a certain sound frequency. There are about 5000 separately detectable pitches over the 10 octaves of audibility.

3.2 PITCH

Critical Bands

Pitch is sensed by the position of maximum excursion along the basilar membrane. This is the ear's spectrum analyzer. There are some 24 discrete areas, each about 1.3 mm long and each contains about 1300 neurons (Scharf, 1970). These act as a set of parallel band-pass filters to separate the incoming sounds into their spectral components. Like electronic filters, the cochlear filters have bandwidths and filter skirts that overlap other bands. When two tones are close enough together that there is significant overlap in their skirts they are said to be within the same *critical band*. The region of influence, which constitutes a critical band, is illustrated in Fig. 3.4. For many phenomena it is about one-third-octave wide. The lower frequencies are sensed by the cochlea at a greater distance from the stapes. The shape of the resonance is not symmetric, having a tail that extends back along the basilar membrane (upward in frequency) from the center frequency of the band. Thus a lower pitched sound

FIGURE 3.4 Critical Bandwidths of the Ear (Kinsler et al., 1982)



can have a region of influence on a higher pitched sound, but not vice versa unless the sounds are quite close in frequency.

The phenomenon of critical bands is of great significance for many aspects of human hearing. They play a role in music by defining regions of consonance and dissonance. They influence the calculation of loudness by determining the method of combination used for multiple tones. They are critical to the phenomenon of masking, explaining many of the varied masking experiments.

Consonance and Dissonance

When two tones are played together, there is a frequency range over which they sound rough or dissonant (Fig. 3.5). Hermann von Helmholtz in his famous book, *On the Sensations of Tone*, hypothesized that the phenomenon of consonance was closely related to the frequency separation of tones and their harmonics. He thought that when two tones or their partials had a difference frequency of 30 to 40 Hz, this caused unpleasant beats. Subsequent experiments by Plomp and Levelt (1965) added some additional factors to his hypothesis.

Plomp's experiments revealed that the maximum dissonance occurs at about 25% of the critical bandwidth. Figure 3.6 gives a graph of consonance and dissonance as a function

FIGURE 3.5 Auditory Perception within a Critical Band (Pierce,1983)

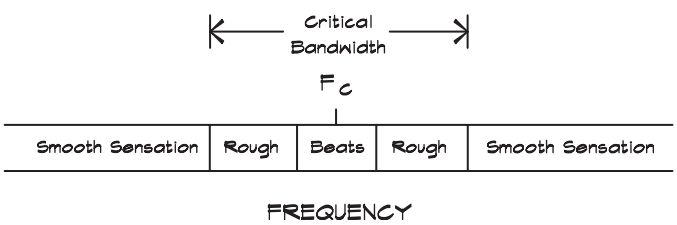
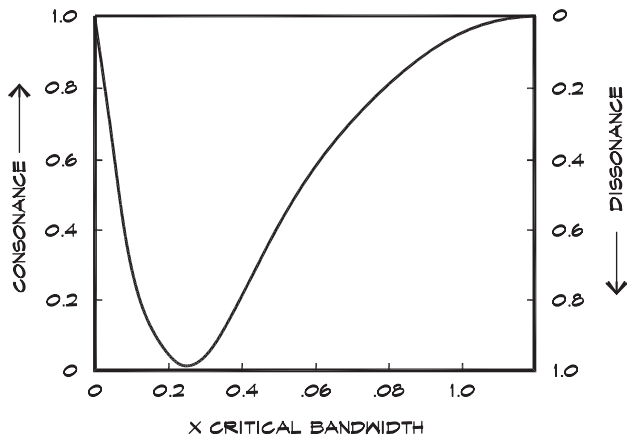


FIGURE 3.6 Consonance and Dissonance as a Function of the Critical Bandwidth (Pierce, 1983)



of the difference frequency, shown in terms of the critical bandwidth. When two tones are very close together the difference frequency is too small to be detected as dissonance but is rather a *tremolo*, a rising and falling of level. As they move apart the two tones interfere in such a way as to produce a roughness. When the frequency difference increases still further, the tones begin to be sensed separately and smoothly. For all frequency differences greater than a critical band, separate tones sound consonant.

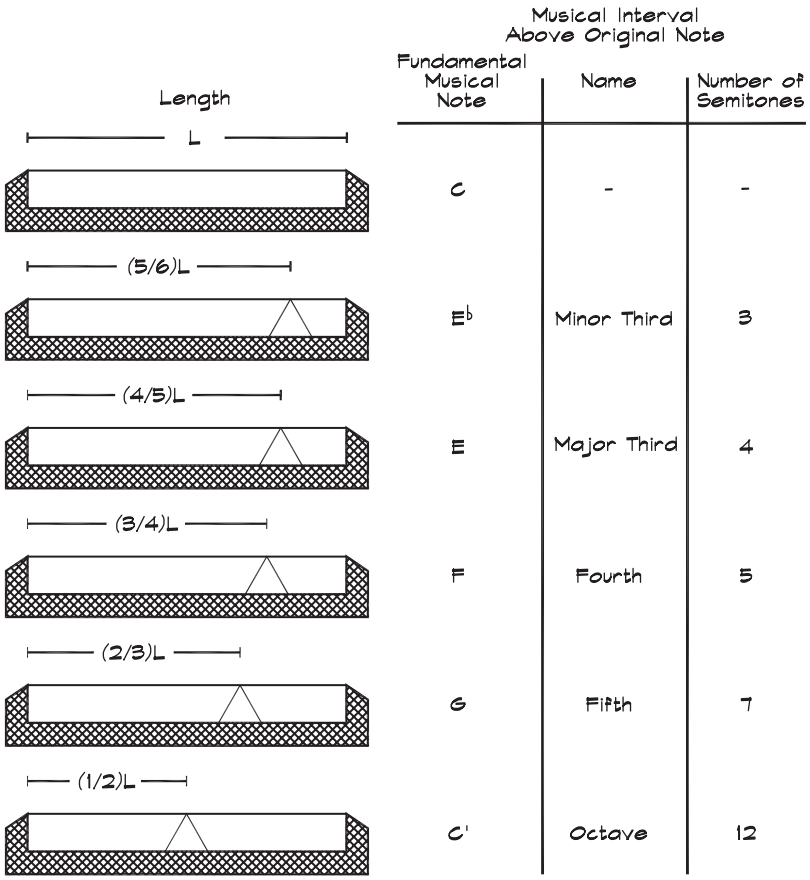
Tone Scales

One of the traditional problems of music is the establishment of a scale of notes, based on frequency intervals, that sound pleasant when played together. The most fundamental scale division is the octave or doubling of frequency, which is the basis for virtually all music. The octave has been variously divided up into 5 notes (pentatonic scale), 7 notes (diatonic scale), or 12 notes (chromatic scale) by different cultures. Western music uses 12 intervals called *semitones*, but a particular piece usually employs a group of seven selected notes, which are designated as a particular key, bearing the name of the starting note. These scales are called major or minor depending on the order of the single or double steps between the notes selected.

Since many of the early instruments were stringed, not only would the player hear the note's fundamental pitch but also its harmonics, which are integer multiples of the fundamental. Note that the second harmonic frequency is twice the fundamental, the third three times and so forth. The first harmonic is the fundamental. Overtones are sometimes taken to mean the same thing as harmonics. In this work overtones are any significant tonal component of the spectrum of a note whether or not this tone has a harmonic relationship to the note's fundamental. It is not uncommon, particularly in percussion instruments such as chimes, to find nonharmonic overtones, some of which change in frequency as they decay. This group of sounds constitutes a musical instrument's spectrum or timbre, the particular character it has that distinguishes its sound from other similar sounds.

Pythagoras of Samos (sixth century BC) is credited with the discovery that when a string is bridged, with one segment having a certain fractional ratio to the overall length, namely 1/2, 2/3, 3/4, 4/5, and 5/6, the two notes had a pleasing sound. These ratios are called the

FIGURE 3.7 Pythagorean Pitch Intervals (Pierce,1983)



A stretched string held at a constant tension produces different frequencies depending on the length of the string. The whole number ratios and their relation to the pitch were studied by Pythagoras.

perfect intervals in music and traditionally are given special names based on the number of diatonic intervals between them. Figure 3.7 shows the ratios and how they may be obtained from a stretched string and how they relate to the notes we use today. The 2/3 ratio is called a fifth since it has 5 diatonic intervals, the 3/4 ratio is a fourth, 4/5 a major third, and 5/6 a minor third.

Having established the first two consonant intervals, several systems were used to fill in the remaining notes to create a musical scale. One was the just scale, which set all note intervals to small integer ratios. Another was the Pythagorean scale, which sought to produce the most equal whole number ratio intervals. Finally was the *equal-tempered scale*, which set the steps between notes to the same ratio. Each system has some problem. The first two do not transpose well to another key, that is, they do not sound the same since the notes have different relationships. The equal-tempered scale introduced about 300 years ago abandoned adherence to perfect integer ratios but chose intervals that did not differ significantly from those ratios. In this scheme, advocated by J. S. Bach, each note is separated from the following one by a factor of $\sqrt[12]{2} = 1.059463$, called a semitone. Every note interval, is in turn, divided

into 100 cents so that there are 1200 cents in an octave. The frequency ratio between each cent is the 1200th root of 2, or 1.00057779. In this system a scale may begin on any white or black key on the piano and sound alike. Bach wrote his series, *The Well Tempered Clavier*, which contains pieces in all major and minor keys, in part to illustrate this method of tuning.

Once the system of note intervals had been established, there was the problem of choosing where to begin. For many years there were no pitch standards and, according to Helmholtz (1877), pipe organs were built with A's ranging from 374 Hz to 567 Hz. Handel's tuning fork was measured at 422.5 Hz and that became the standard for the classical composers Haydn, Bach, Mozart, and Beethoven. In 1859 the standard A rose to 435 Hz, set by a French government commission, which included Berlioz, Meyerbeer, and Rossini. The so-called "scientific" pitch was introduced in the early twentieth century with the C note frequencies being integer powers of 2, much like the designation of today's computer memory chips. This resulted in an A of 431 Hz. Later, in 1939, an international conference in London adopted the current standard, A equal to 440 Hz at 20° C.

Tunings still vary with individual instruments and musical taste. The natural frequency in woodwinds rises about 3 cents per degree C due to the increase in the velocity of sound. In stringed instruments the fundamental frequency falls slightly with temperature due to the thermal expansion of the strings. Some musicians raise the pitch of their tunings to get additional edge or brightness. This is an unfortunate trend since it stresses older instruments and adds a shrillness to the music, particularly in the strings.

Pitch

Pitch is the human perception of how high or low a tone sounds, based on its relative position on a scale. Musical pitch is defined in terms of notes however, there are psychoacoustical experiments to measure human perception of relative pitch as well. Absolute pitch discrimination is rather rare occurring in only 0.01 percent of the population (Rossing, 1990). Relative pitch discrimination can be measured by asking subjects to respond when one tone sounds twice as high as another. Like loudness experiments, the results are complex, for while they depend primarily on frequency, they also can vary with intensity and waveform. For example if a 100 Hz tone is sounded at 60 dB and then at 80 dB, the louder sound will be perceived as having the lower pitch. This phenomenon is primarily one that occurs at frequencies below 300 Hz. At the mid frequencies (500 Hz to 3000 Hz), pitch is relatively independent of intensity, whereas at frequencies above 4000 Hz, pitch increases with level.

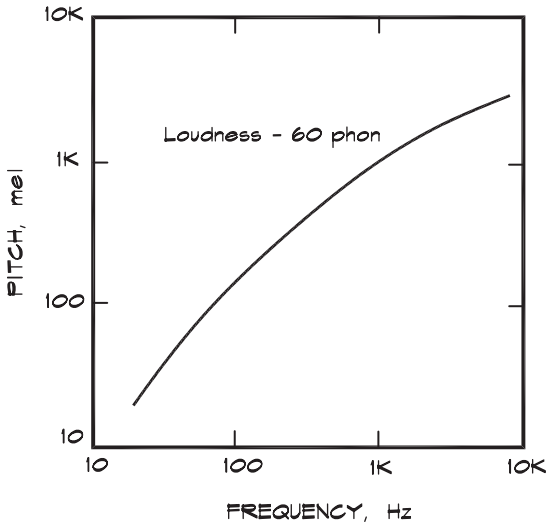
Pitch, as measured in these types of experiments, is different from harmonic relationships found in music. The former is expressed in units of mels, which are similar to sones in that a tone having 2000 mels is judged to be twice as high as one with 1000 mels. The reference frequency for a tone at a given loudness is 1000 Hz, which is defined as 1000 mels. For a given loudness it is possible to define a curve of constant pitch versus frequency as in Fig. 3.8.

3.3 LOUDNESS

Comparative Loudness

Loudness is the human perception of the magnitude of a sound. Early efforts to quantify loudness were undertaken in the field of music. The terms "very loud," "loud," "moderately loud," "soft," and "very soft" were given symbols in musical notation—ff, f, mf, p, and pp, after these words in Italian. But the terms are not sufficiently precise for scientific use,

FIGURE 3.8 Relative Pitch Discrimination vs Frequency (Kinsler et al., 1982)



and depend on the hearing acuity and custom of the person using them. It would seem straightforward to use the measured intensity of a sound to determine its loudness, but unfortunately no such simple relationship exists. Loudness is ultimately dependent on the number of nerve impulses reaching the brain in a given time, but since those impulses come from different regions of the cochlea there is also a variation with the frequency content of the sound.

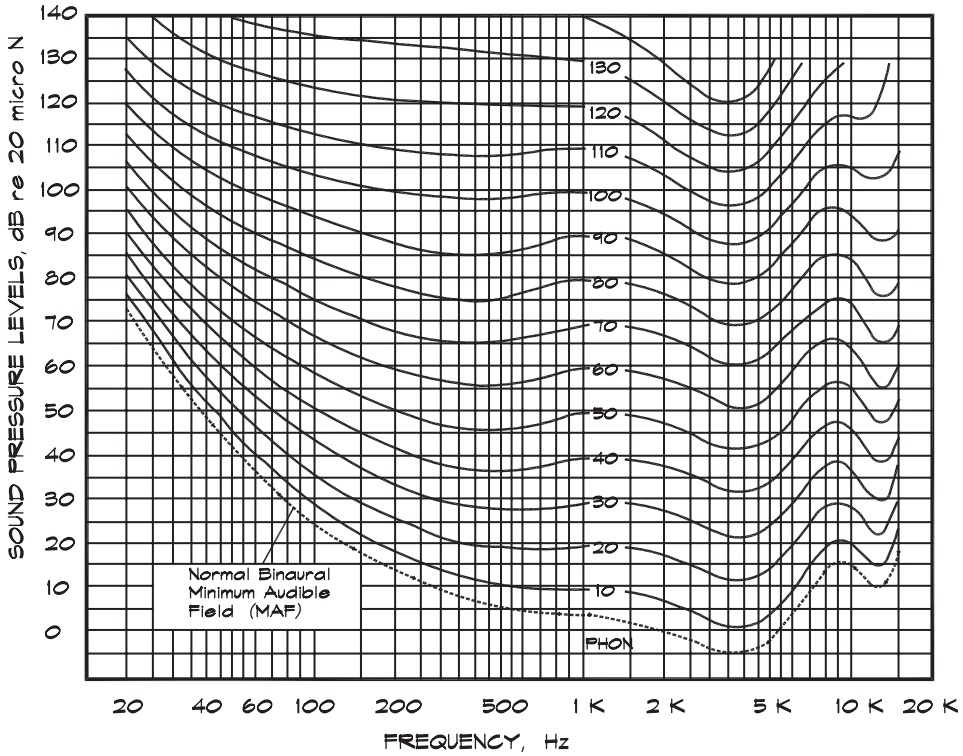
Even when the same sound signal is heard at differing intensities, there will be some variability from listener to listener and indeed even variation with the same listener, depending on his psychological and physiological state. Of general interest is the expected reaction of a listener in a typical environment, which can be determined by testing a number of subjects under controlled conditions. The average of an ensemble of listeners is taken as the result expected from a typical listener, a premise known as the *ergodic* hypothesis in statistics.

Loudness Levels

Comparative loudness measurements were made in the 1920's and 30's by scientists at Bell Laboratories. These tests were done on a group of subjects with acute hearing by presenting them with a controlled set of sounds. Various signals were used, however, in the classic study by Fletcher and Munson, published in 1933, pure tones (sine waves) of short duration were employed. The procedure was to compare the loudness of a tone, presented to the listener at a particular frequency and amplitude, to a fixed reference tone at 1000 Hz having an amplitude that was set in 10 dB intervals to between 0 and 120 dB. Tones were presented to the listeners, by means of headphones, for a one-second duration with a 1.5 second pause in between. Subjects were asked to choose whether a given tone was above, below, or equal to the reference tone, which resulted in a group of loudness-level contours known as the *Fletcher-Munson curves*.

In 1956 Robinson and Dadson repeated the Fletcher-Munson measurements, this time using loudspeakers in an anechoic chamber. The resulting *Robinson-Dadson* curves are shown in Fig. 3.9. The lowest of these curves is the threshold of hearing, which passes through 0 dB at about 2000 Hz and drops below this level at 4000 Hz, where the ear is most sensitive.

FIGURE 3.9 Normal Loudness Contours for Pure Tones (Robinson and Dadson, 1956)



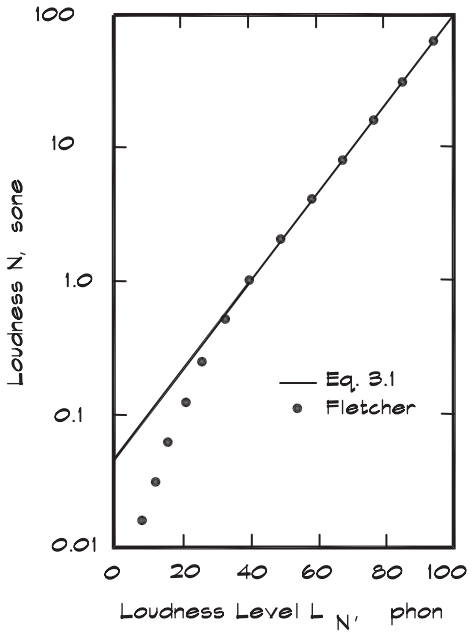
The graph shows that human hearing is significantly less sensitive to low-frequency sounds. At lower frequencies the level rises rapidly as the frequency decreases until, at 30 Hz, the intensity level must be about 65 dB before it is audible. As the intensity of a sound increases, the ear's frequency response becomes flatter. Near 100 dB the ear's response is almost flat except for a small increase in sensitivity around 4000 Hz. These experiments have been repeated many times since the original work was done with various types of signals. They have also been performed on the general population to determine the behavior of hearing acuity with age and other factors.

The curves in Fig. 3.9 are called equal-loudness contours. For any two points along one of these curves the perceived loudness of tones is the same. A loudness level (having units of phons) is assigned to each curve, which is numerically equal to the intensity level of the tone at 1000 Hz. Thus if we follow the 40 phon line we see that a tone having an intensity level of 50 dB at 100 Hz falls on the line and thus has a loudness level of 40 phons. At 1000 Hz the loudness level is equal to the intensity level. At 10,000 Hz the intensity level on the 40 phon line is about 46 dB.

Relative Loudness

The loudness level contours are based on human judgments of absolute loudness. The question asked each subject is whether the tone is louder or quieter than the reference. Another question can be asked, based on a relative comparison, namely, "When is it twice as loud?"

FIGURE 3.10 Loudness vs Loudness Level (Fletcher, 1933)



This gives rise to a measure of relative loudness having units of sones. In this scheme the loudness metric is linear; a sound having a relative loudness of 2 sones is twice as loud as a sound of 1 sone and so forth. The baseline is the 40 phon curve that is given the value of 1 sone. Figure 3.10 shows the relationship between (relative) loudness in sones and loudness level in phons. The result of these measurements is that loudness doubles every 9 phons or about 9 dB at the mid frequencies. A general equation can be written for the relationship between loudness and loudness level of pure tones in the linear region of the curve as follows (Kinsler and Frey, 1962)

$$L_N \cong 30 \log N + 40 \tag{3.1}$$

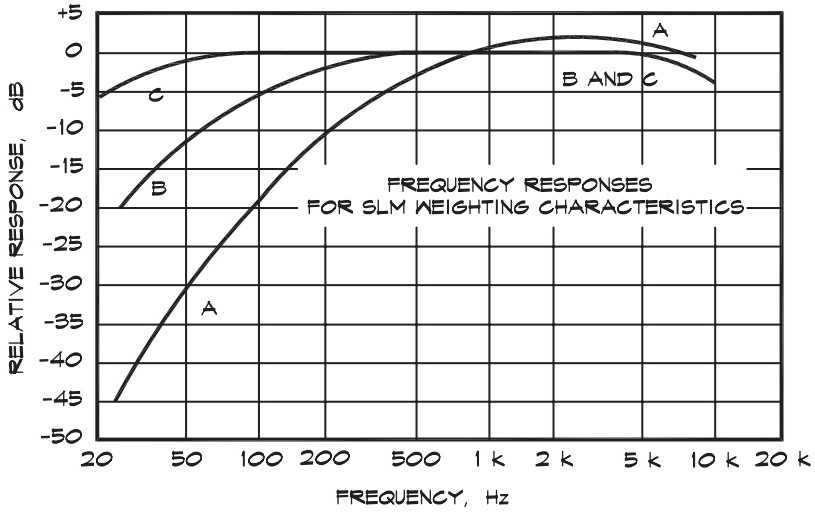
where N = loudness (sones)
L_N = loudness level (phons)

A determination can also be made of the loudness of sounds, having a spectral content more complicated than pure tones. Starting with measured levels in one-third-octave or full-octave bandwidths, various schemes have been proposed by Kryter, Stevens, and Zwicker for the calculation of both loudness and loudness levels. While these are generally more successful than simple electronic filtering in correlating with the human perception of loudness, their complexity limits their usefulness.

Electrical Weighting Networks

Although the Fletcher Munson curves provided an accurate measure of the relative loudness of tones, their shape was too complicated for use with an analog sound level meter. To overcome this problem, electrical weighting networks or filters were developed, which approximate the Fletcher Munson curves. These frequency weighting schemes were designated by letters of the alphabet, A, B, C, and so forth. The A, B, and C-weighting networks were designed

FIGURE 3.11 Frequency Response Characteristics in the American National Standards Specification for Sound Level Meters (ANSI-S1.4 – 1971)



to roughly mirror the 40, 60, and 80 phon lines (turned upside down) shown in Fig. 3.11. The relative weightings in each third-octave band for the A and C filters are set forth in Table 3.1. Since the time of their original development other weighting curves have been suggested. Several D-weighting curves are detailed by Kryter (1970) and an E curve has been suggested, but these systems, along with B-weighting, have not found widespread acceptance.

Only the A and C curves are still in general use. The A-weighted level (dBA) is the most common single number measure of loudness. The C-weighting network is used mainly as a measure of the broadband sound pressure level. Occasionally an $(L_C - L_A)$ level is used to describe the relative contribution of low-frequency noise to a spectrum. The weightings can be applied to sound power or sound pressure levels.

Once a weighting is applied it should not be reapplied. For example, if a recording is made of environmental noise using an A-weighting filter (not a recommended practice) it should not be analyzed by playing it back through a meter using the A-weighting network. It is, however, not uncommon to use a C-weighting filter when recording environmental noise, since it limits the low-frequency sounds that can overload the tape recorder. Replaying a tape thus recorded back through an A-weighting filter would be a reasonable practice so long as the main frequencies of interest were those unaffected by the C-weighting.

It is not uncommon to encounter A-weighted octave or third-octave band levels. These levels, if appropriately designated, are understood to have had the weighting already applied. A-weighted octave-band levels can be calculated from third-octave levels using Eq. 2.62 to combine the three levels within a particular octave band. Likewise overall A-weighted levels can be calculated from A-weighted octave-band levels by using the same formula.

Noise Criteria Curves (NC and RC)

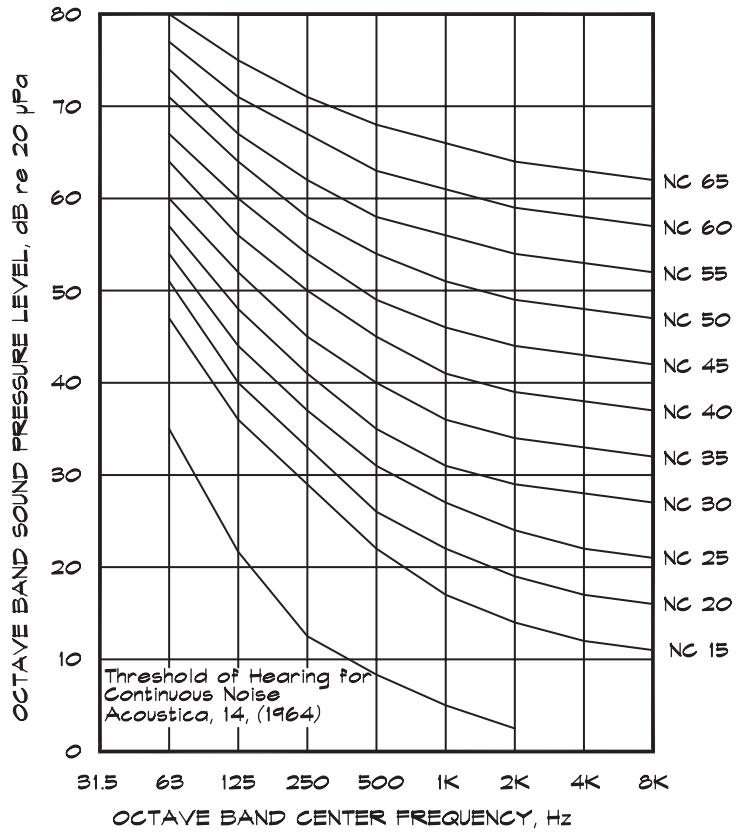
Loudness curves based on octave-band sound pressure level measurements are commonly used in buildings to establish standards for various types of activities. The *noise criterion* (NC) curves shown in Fig. 3.12 were developed by Beranek in 1957 to establish satisfactory

TABLE 3.1 Electrical Weighting Networks

Frequency Hz	A-Weighting Relative Response, dB	C-Weighting Relative Response, dB
12.5	−63.4	−11.2
16	**−56.7**	**−8.5**
20	−50.5	−6.2
25	−44.7	−4.4
31.5	**−39.4**	**−3.0**
40	−34.6	−2.0
50	−30.2	−1.3
63	**−26.2**	**−0.8**
80	−22.5	−0.5
100	−19.1	−0.3
125	**−16.1**	**−0.2**
160	−13.4	−0.1
200	−10.9	0
250	**−8.6**	**0**
315	−6.6	0
400	−4.8	0
500	**−3.2**	**0**
630	−1.9	0
800	−0.8	0
1,000	**0**	**0**
1,250	+0.6	0
1,600	+1.0	−0.1
2,000	**+1.2**	**−0.2**
2,500	+1.3	−0.3
3,150	+1.2	−0.5
4,000	**+1.0**	**−0.8**
5,000	+0.5	−1.3
6,300	−0.1	−2.0
8,000	**−1.1**	**−3.0**
10,000	−2.5	−4.4
12,500	−4.3	−6.2
16,000	**−6.6**	**−8.5**
20,000	−9.3	−11.2

conditions for speech intelligibility and general living environments. They are expressed as a series of curves, which are designated NC-30, NC-35, and so on, according to where the curve crossed the 1750 Hz frequency line in an old (now obsolete) octave-band designation. The NC level is determined from the lowest NC curve, which may be drawn such that no point on a measured octave-band spectrum lies above it. Since the NC curves are defined in 5 dB intervals, in between these values the NC level is interpolated.

FIGURE 3.12 Noise Criterion (NC) Curves (Beranek, 1957)



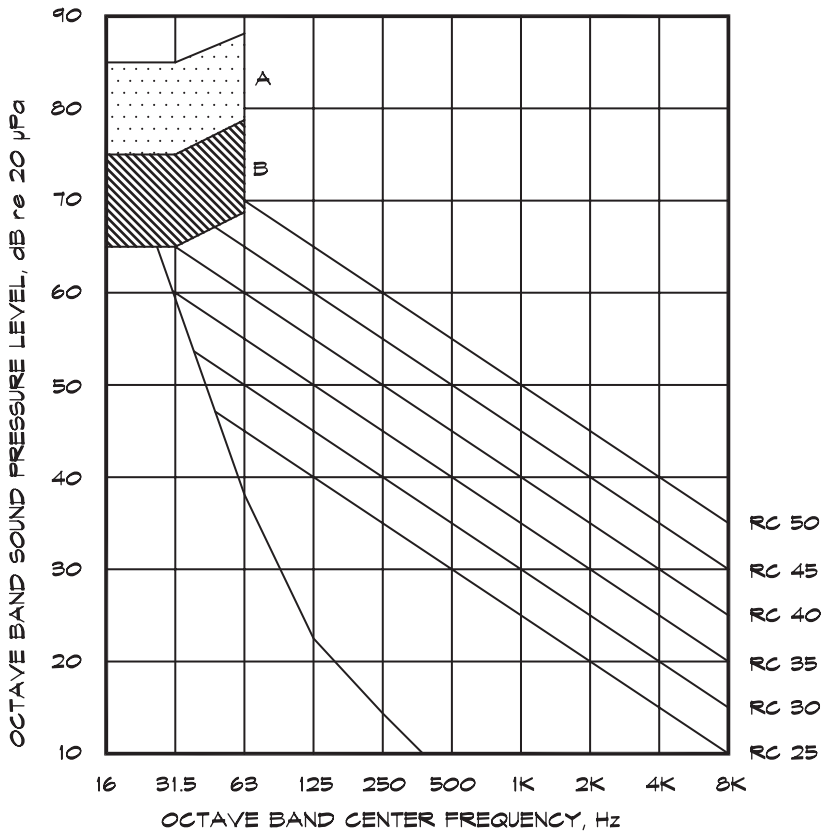
The NC level depends on the actual measured (or calculated) spectrum of the sound but they can be generally related to an overall A-weighted level (Kinsler et al., 1982)

$$NC \cong 1.25\,(L_A - 13) \tag{3.2}$$

where NC = NC level, dB
L_A = sound pressure level, dBA

In 1981 Blazier developed the set of curves, shown in Fig. 3.13, called *room criterion* (RC) curves, based on an American Society of Heating, Refrigeration, and Air Conditioning Engineers (ASHRAE) study of heating, ventilating, and air conditioning (HVAC) noise in office environments. Since these curves are straight lines, spaced 5 dB apart, there is no confusion about the level, which can occur sometimes with NC levels. RC curves are more stringent at low frequencies but include 5 dB of leeway in the computation methodology. The RC level is the arithmetic average of the 500, 1000, and 2000 Hz octave-band values taken from the measured spectrum. At frequencies above and below these center bands a second parallel line is drawn. Below 500 Hz the line is 5 dB above the corresponding RC line and above 2000 Hz it is 3 dB above the line. If the measured spectrum exceeds the low-frequency line the RC level is given the designation R for rumble. If it exceeds the high-frequency line the designation is H for hissy. Otherwise the designation is N for normal.

FIGURE 3.13 Room Criterion (RC) Curves (Blazier, 1981)



Blazier added two other regions to these curves at the low frequencies where mechanical vibrations can be a nuisance in lightweight structures. In region A there is a high probability of noticeable vibrations accompanying the noise while in region B there is a low probability of that occurring, particularly in the lower portion of the curve.

From time to time ASHRAE publishes a guide to assist in the calculation and treatment of HVAC generated noise levels. As part of this guide they include suggested levels of noise for various classifications of interior space. In general these guidelines are applied to noise generated by equipment associated with a building, such as HVAC systems within a dwelling or office, or noise generated in an adjacent room by HVAC, pumps, fans, or plumbing. The standards are not generally applied to noise generated by appliances, which plug into wall sockets within the same dwelling unit. A portion of the 1987 ASHRAE guidelines are shown in Table 3.2.

Just Noticeable Difference

One of the classic psychoacoustic experiments is the measurement of a *just noticeable difference* (jnd), which is also called a difference limen. In these tests a subject is asked to compare two sounds and to indicate which is higher in level, or in frequency. What is found is that the jnd in level depends on both the intensity and frequency. Table 3.3 shows jnd level values at various sound pressure levels and frequencies. For sound levels exceeding 40 dB and at frequencies above 100 Hz, the jnd is less than 1 dB. At the most sensitive levels

TABLE 3.2 Interior Noise Design Goals (ASHRAE, 1987)

	Type of Area	Recommended NC or RC Criteria Range
1	Private Residences	25 to 30
2	Apartments	25 to 30
3	Hotels/motels	
	a Individual rooms or suites	30 to 35
	b Meeting/banquet rooms	25 to 30
	c Halls, corridors, lobbies	35 to 40
	d Service/support areas	40 to 45
4	Offices	
	a Executive	25 to 30
	b Conference room	25 to 30
	c Private	30 to 35
	d Open plan areas	35 to 40
	e Computer equipment rooms	40 to 45
	f Public circulation	40 to 45
5	Hospitals and clinics	
	a Private rooms	25 to 30
	b Wards	30 to 35
	c Operating rooms	35 to 40
	d Corridors	35 to 40
	e Public areas	35 to 40
6	Churches	25 to 30
7	Schools	
	a Lecture and classrooms	25 to 30
	b Open plan classrooms	30 to 35
8	Libraries	35 to 40
9	Concert halls	5 to 15*
10	Legitimate theaters	20 to 30
11	Recording studios	10 to 20*
12	Movie theaters	30 to 35

***Note: Where ASHRAE has recommended that an acoustical engineer be consulted, the author has supplied the NC or RC levels.**

TABLE 3.3 Minimum Detectable Changes (JND) in Level for Sine Waves, dB (Pierce, 1983)

Freq.	Signal Level, dB											
Hz	5	10	20	30	40	50	60	70	80	90	100	110
35	9.3	7.8	4.3	1.8	1.8							
70	5.7	4.2	2.4	1.5	1.0	.75	.61	.57	1.0	1.0		
200	4.7	3.4	1.2	1.2	.86	.68	.53	.45	.41	.41		
1000	3.0	2.3	1.5	1.0	.72	.53	.41	.33	.29	.29	.25	.25
4000	2.5	1.7	.97	.68	.49	.41	.29	.25	.25	.21	.21	
8000	4.0	2.8	1.5	.90	.68	.61	.53	.49	.45	.41		
10,000	4.7	3.3	1.7	1.1	.86	.75	.68	.61	.57			

TABLE 3.4 Minimum Detectable Changes (JND) in Frequency for Sine Waves, Cents (Pierce, 1983)

Frequency	Signal Level, dB										
Hz	5	10	15	20	30	40	50	60	70	80	90
31	220	150	120	97	76	70					
62	120	120	94	85	80	74	61	60			
125	100	73	57	52	46	43	48	47			
250	61	37	27	22	19	18	17	17	17	17	
500	28	19	14	12	10	9	7	6	7		
1000	16	11	8	7	6	6	6	6	5	5	4
2000	14	6	5	4	3	3	3	3	3	3	
4000	10	8	7	5	5	4	4	4	4		
8000	11	9	8	7	6	5	4	4			
11,700	12	10	7	6	6	6	5				

(greater than 60 dB) and frequencies (1000–4000 Hz), the jnd is about a quarter of a dB. When the jnd is 0.25 dB it means that we can notice a sound, with the same spectrum, which is 13 dB below the level of the background. This has important implications for both privacy and intelligibility in the design of speech reinforcement systems.

The jnd values in frequency for sine waves are shown in Table 3.4. Like the jnd in level, it is also dependent on both intensity and frequency. At 2000 Hz, where we are most sensitive, the jnd is about 3 cents (0.3% of an octave) or about 0.5% of the pure tone frequency for levels above 30 dB. This is about 10 Hz. Some trained musicians can tell the difference between a perfect fifth (702 cents) and an equal tempered fifth (700 cents), so that greater sensitivity is not uncommon. Note that these comparisons are done by sounding successive tones or by varying the tone, not by comparing simultaneous tones where greater precision is obtainable by listening for beats. Piano tuners who tune by ear use this latter method to achieve precise tuning.

Environmental Impact

Environmental Impact Reports (EIR) in California or Environmental Impact Statements (EIS) for Federal projects are prepared when a proposed project has the potential of creating a significant adverse impact on the environment (California Environmental Quality Act, 1972). Noise is often one of the environmental effects generated by a development, through increases in traffic or fixed noise sources. Impact may be judged either on an absolute scale through comparison with a standard such as a property line ordinance or a Noise Element of a General Plan, or on a relative scale through changes in level. In the exterior environment the sensitivity to changes in noise level is not as great as in the laboratory under controlled conditions. A 1 dB change is the threshold for most people. Since the change in level due to multiple sound sources is equal to 10 log N, where N is the ratio of the new to the old number, it takes a 1.26 ratio or a 26% increase in traffic passing by on a street to produce a 1 dB change. Table 3.5 shows a general characterization of human reaction to changes in level.

TABLE 3.5 Human Reaction to Changes in Level

Change in Level (dB)	Reaction
1	Noticeable
3	Very Noticeable
6	Substantial
10	Doubling (or Halving)

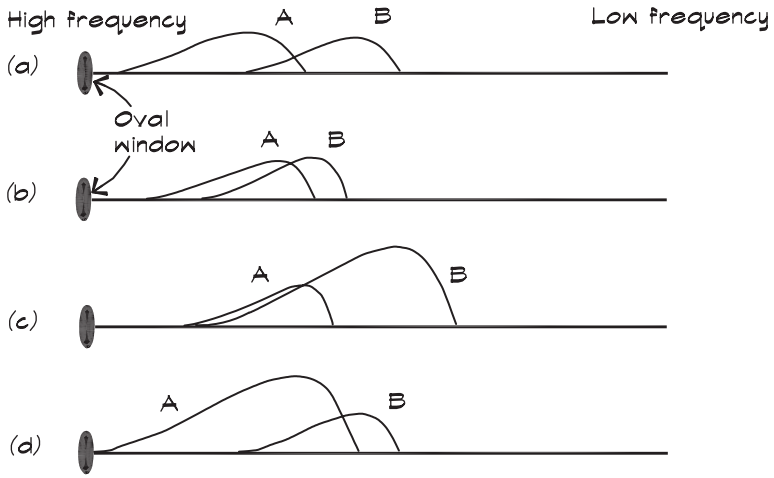
The characterizations listed in Table 3.5 are useful in gauging the reaction to changes in environmental noise. If a project increases the overall noise level at a given location by 1 dB or more, it is likely to be noticeable and could potentially constitute an adverse environmental impact. A 3 dB increase is very noticeable and, in the case of traffic flow, represents a doubling in traffic volume.

3.4 INTELLIGIBILITY

Masking

When we listen to two or more tones simultaneously, if their levels are sufficiently different, it becomes difficult to perceive the quieter tone. We say that the quieter sound is *masked* by the louder. Masking can be understood in terms of a threshold shift produced by the louder tone due to its overlap within the critical band on the cochlea. Figure 3.14 illustrates this principle. The figure is helpful in understanding the findings of experiments associated with

FIGURE 3.14 Overlap of Regions of the Cochlea (Rossing, 1990)

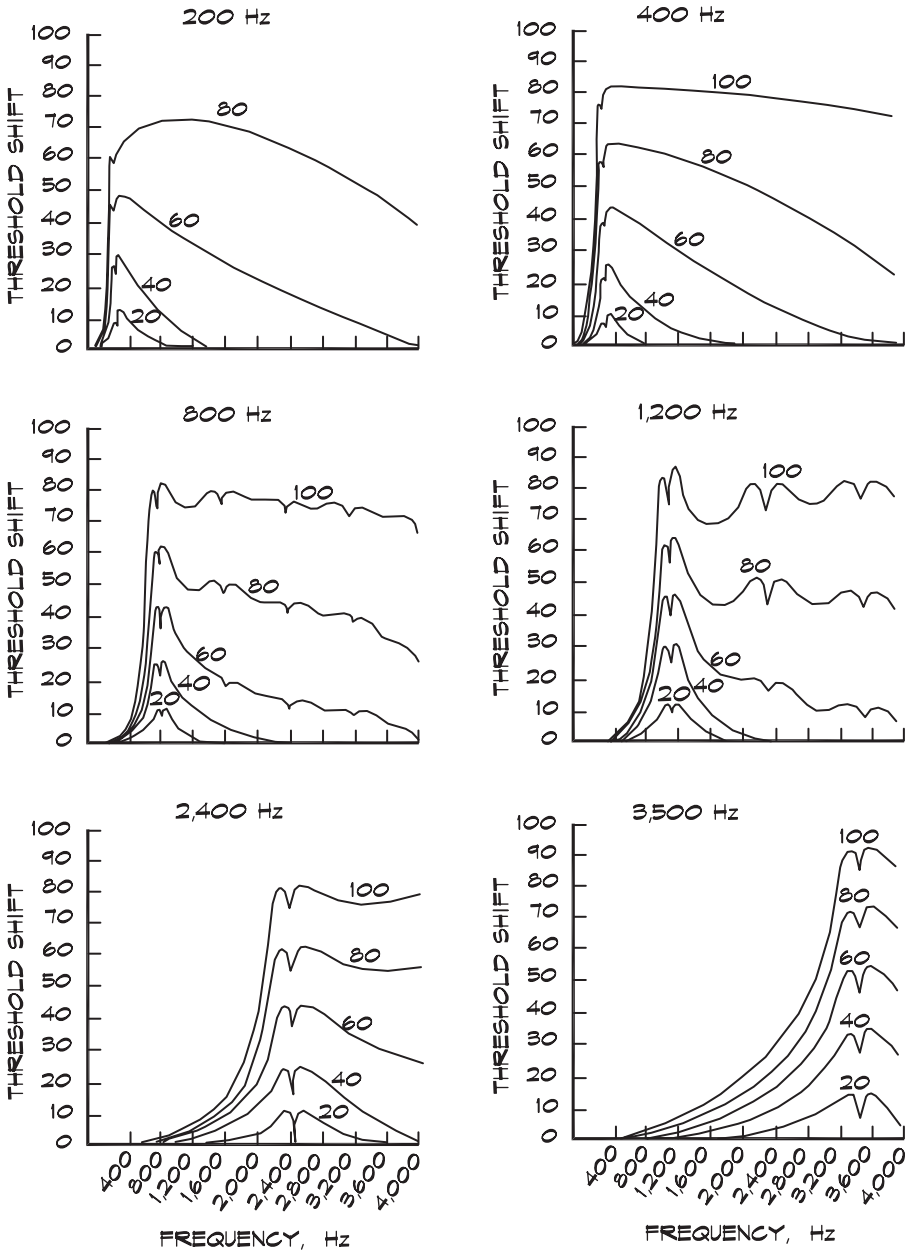


Simplified response of the basilar membrane for two pure tones A and B. (a) The excitations barely overlap; little masking occurs. (b) There is an appreciable overlap; tone B masks tone A and somewhat more than the reverse. (c) The more intense tone B almost completely masks the higher-frequency tone A. (d) The more intense tone A does not completely mask the lower-frequency tone B.

masking. Tones that are close in frequency mask each other more than those that are widely separated.

Tones mask upward in frequency rather than downward. The louder the masking tone the wider the range of frequencies it can mask. Masking by narrow bands of noise mimics that of pure tones and broad bands of noise mask at all frequencies. Early experiments on masking the audibility of tones in the presence of noise were performed by Wegel and Lane (1924) and subsequently published by Fletcher (1953). Two tones are presented to each subject. The first is a constant masking tone at a given level and frequency. A second tone

FIGURE 3.15 Pure Tone Masking Curves (Fletcher, 1953)



is introduced at a selectable frequency and its level is reduced until it is no longer audible. Based on these types of tests a series of masking curves can be drawn, which are shown in Fig. 3.15. Each curve shows the threshold shift of the masked tone or the difference between its normal threshold of audibility and the new threshold in the presence of the masking tone. For example in the second curve the masking tone is at 400 Hz. At 80 dB it induces a 60 dB threshold shift in an 800 Hz tone. From Fig. 3.9 we can see that the threshold of tonal hearing at 800 Hz is 0 dB, so the level above which the 800 Hz tone is audible is 60 dB. The fine structure on the masking curves is interesting. Around the frequency of the masking tone there are little dips in threshold shift—which is to say the ear becomes more sensitive. These dips repeat at the harmonics of the masking frequency. The reason is that when the two frequencies coincide, beats are generated that alert us to the presence of the masked tone.

As the level of the masking tone is increased, the breadth of its influence increases. A 400 Hz masking tone at 100 dB is effective in swamping the ear's response to 4000 Hz tones, while a 40 or 60 dB masking tone does little at these frequencies. At low levels the bandwidth of masking effectiveness is close to the critical bandwidth. High-frequency masking tones have little or no effect on lower frequency tones. A 2400 Hz masking tone will not mask a 400 Hz tone no matter how loud it is. This graphically illustrates the effect of the shape of the cochlear filter skirts.

Masking experiments also can be used to define critical bands. In the standard masking test a tone is not audible in the presence of masking noise until its level exceeds a certain value. The masking noise can be configured to have an adjustable bandwidth and the tests can be repeated for various noise bandwidths. It has been found (Fletcher and Munson, 1937) that masking is independent of noise bandwidth until the bandwidth is reduced below a certain value. This led to a separate way of measuring critical bandwidths, which gave results similar to those achieved using consonance and dissonance.

Masking is an important consideration in architectural acoustics. It is of particular interest to an acoustician whether speech will be intelligible in the presence of noise. In large indoor facilities, such as air terminals or sports arenas, low-frequency reverberant noise can mask the intelligibility of speech. This can be partially treated by limiting the bandwidth of the sound system or by adding low-frequency absorption to the room. The former is less expensive but limits the range of uses. Multipurpose arenas, which are hockey rinks one day and rock venues the next, should have an acoustical environment that does not limit the uses of the space.

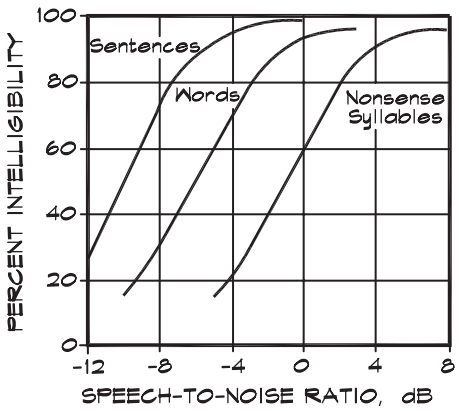
Speech Intelligibility

Speech intelligibility is a direct measure of the fraction of words or sentences understood by a listener. The most direct method of measuring intelligibility is to use sentences containing individual words or nonsense syllables, which are read to listeners who are asked to identify them. These can be presented at various levels in the presence of background noise or reverberation. Both live and recorded voices are used, however recorded voices are more consistent and controllable.

Three types of material are typically used: sentences, one syllable words, and nonsense syllables, with each type being increasingly more difficult to understand in the presence of noise. In sentence tests a passage is read from a text. In a word test individual words are read from a predetermined list, called a closed response word set, and subjects are asked to pick the correct one. A modified rhyme test uses 50 six-word groups of monosyllabic rhyming or similar-sounding English words. Subjects are asked to correctly identify the spoken word

FIGURE 3.16 Results of a Typical Intelligibility Test (Miller et al., 1951)

Intelligibility of different types of test materials in the presence of noise. Speech intelligibility is shown as a function of speech-to-noise ratio for sentences, monosyllabic words, and nonsense syllables.



from the list of six possible choices. A group of words might be: sag, sat, sass, sack, sad, and sap. Tests of this type lead to a measure of the fraction, ranging from 0 to 1, of words that are correctly identified. Figure 3.16 shows some typical results.

The degree to which noise inhibits intelligibility is dependent on the *signal-to-noise ratio*, which is simply the signal level minus the noise level in dB. When the noise is higher than the speech level, the signal-to-noise ratio is negative. A signal-to-noise ratio is a commonly used concept in acoustics, audio, and electrical engineering. It is called a ratio since it represents the energy of the signal divided by the energy of the noise, expressed in decibels. For a typical test the noise is broad band steady noise such as that produced by a waterfall.

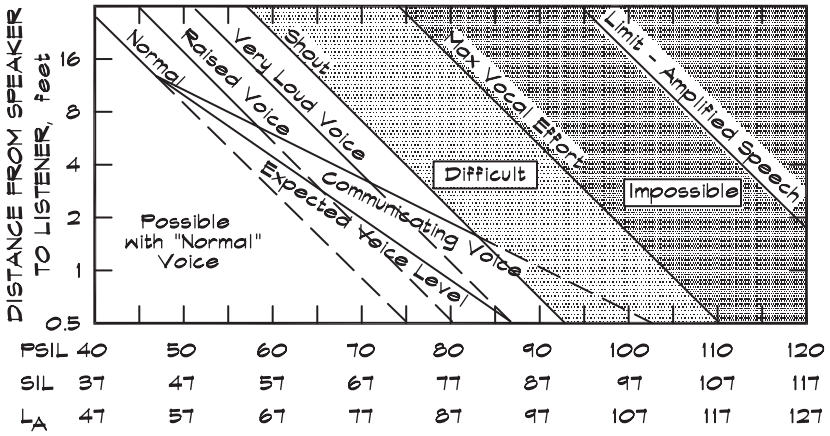
What is apparent from Fig. 3.16 is that even when the signal-to-noise is negative, speech is still intelligible. This is not surprising since the brain is an impressive computer, which can select useful information and fill in the gaps between the words we understand. For most applications if we can grasp more than 85–90% of the words being spoken we achieve very good comprehension—virtually 100% of the sentences. With an understanding of more than 60% of the words we can still get 90% of the sentences and that is quite good. If we understand fewer than 60% of the words the intelligibility drops off rapidly.

Speech Interference Level

Signal-to-noise ratio is the key to speech intelligibility, and we obtain more precise estimates of the potential interference by studying the background noise in the speech frequency bands. The *speech interference level* (SIL) is a measure of a background noise's potential to mask speech. It is calculated by arithmetically averaging separate background noise levels in the four speech octave bands, namely 500, 1000, 2000, and 4000 Hz. The SIL can then be compared to the expected speech sound pressure level to obtain a relevant speech to noise ratio.

Figure 3.17 shows the expected distance over which just-reliable communications can be maintained for various speech interference level values. The graph accounts for the

FIGURE 3.17 Rating Chart for Determining Speech Communication (Webster, 1969)



expected rise in voice level, which occurs in the presence of high background noise. Note that these types of analyses assume a flat spectrum of background noise, constant speech levels, and non-reverberant spaces.

The *preferred speech interference level* (PSIL) is another similar metric, calculated from the arithmetic average of background noise levels in the 500, 1000, and 2000 Hz octave bands. The PSIL can also be used to obtain estimates of the intelligibility of speech in the presence of noise.

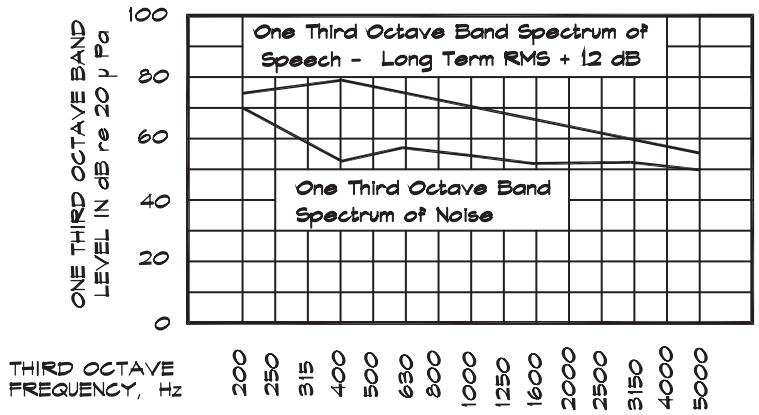
Articulation Index

The *articulation index* (AI) is a detailed method of measuring and calculating speech intelligibility (French and Steinberg, 1947). To measure the AI, a group of listeners is presented a series of phonemes for identification. Each of the test sounds consists of a logatom, or structured nonsense syllable in the form of *a consonant-vowel-consonant* (CVC) group embedded in a neutral carrier sentence, which cannot be recognized from its context in the sentence. An example might be, "Now try pom." The fraction of syllables understood is the AI.

Developed by researchers at Bell Laboratories in the late 1920s and early 1930s, including Fletcher, French, Steinberg, and others, it also included a method of calculating the expected speech intelligibility by using signal-to-noise ratios in third-octave bands, which are then weighted according to their importance. In this method speech intelligibility is proportional to the long-term rms speech signal plus 12 dB minus the noise in each band. The proportionality holds provided the sum of the terms falls between 0 and 30 dB. Figure 3.18 shows the calculation method along with the weighting factors used in each band. In each of 15 third-octave bands the signal-to-noise ratio is multiplied by a factor and the results are added together.

AI calculations can be made even when the spectrum of the background noise is not flat and is different from that of speech. It also accounts in part for the masking of speech by low-frequency noise. AI uses the peak levels generated by speech as the signal level and the energy average background levels as the noise. Consequently the signal-to-noise ratios are somewhat higher than other methods, such as SIL, which are based on average speech levels for the same conditions.

FIGURE 3.18 An Articulation Index Calculation (Kryter, 1970)



BAND	SPEECH PEAKS MINUS NOISE (dB)	WEIGHT	COLUMN 2 x 3
200	5	0.0004	0.0020
250	12	0.0010	0.0120
315	18	0.0010	0.0180
400	26	0.0014	0.0364
500	23	0.0014	0.0322
630	18	0.0020	0.0360
800	17	0.0020	0.0340
1000	16	0.0024	0.0384
1250	15	0.0030	0.0450
1600	14	0.0037	0.0518
2000	12	0.0037	0.0444
2500	10	0.0034	0.0340
3150	8	0.0034	0.0272
4000	6	0.0024	0.0144
5000	5	0.0020	0.0100
			AI = 0.4358

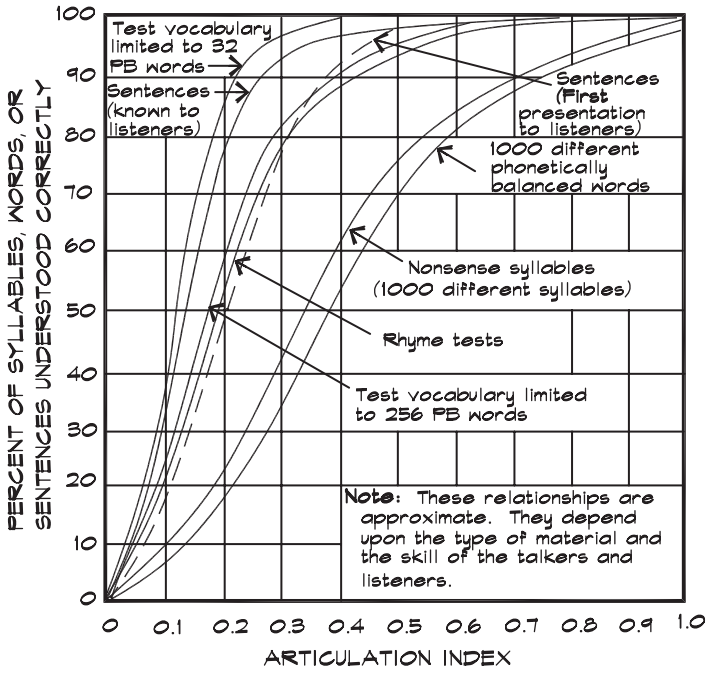
A typical AI calculation using the data shown in the graph. The one-third octave band spectrum of speech is used plus 12 dB and the noise spectrum is subtracted in each band. The result is multiplied by a weighting factor according to the band and the result is summed to obtain an overall Articulation Index figure.

The result of an AI calculation is a numerical factor, which ranges from 0 to 1, with 1 being 100% word or sentence comprehension. Beranek (1947) suggested that a listening environment with an AI of less than 0.3 will be found to be unsatisfactory or marginally satisfactory, while AI values between 0.3 to 0.5 will generally be acceptable. For AI values of 0.5 to 0.7 intelligibility will be good, and above 0.7 intelligibility will be very good to excellent. Figure 3.19 shows the relation between the AI and other measures of speech intelligibility.

AL_{CONS}

The *articulation loss of consonants* (AL_{CONS}), expressed as a percentage, is another way of characterizing the intelligibility of speech. Similar to the articulation index, it measures the proportion of consonants wrongly understood. V. Peutz also found that the correlation

FIGURE 3.19 Relation between AI and Other Speech Intelligibility Tests (ANSI S3.5, 1969)



between the loss of consonants (in Dutch) is much more reliable than a similar test with vowels. He published (1971) a relationship to predict intelligibility for unamplified speech in rooms, which had been studied much earlier at Bell Labs.

$$AL_{CONS} = \frac{200\, r^2\, T_{60}^2}{V} \tag{3.3}$$

Beyond the limiting distance $r_\ell = 0.21\sqrt{V/T_{60}}$

$$AL_{CONS} = 9\, T_{60} \tag{3.4}$$

where T_{60} = reverberation time, (s)
V = room volume, (m^3)
r = talker to listener distance, (m)

Privacy

The inverse of intelligibility is *privacy*, and articulation index is equally useful in the calculation of privacy as it was for intelligibility. Both are ultimately dependent on signal-to-noise ratio. Chanaud (1983) defined five levels of privacy, which are shown in Table 3.6, and has related them to AI in Fig. 3.20.

TABLE 3.6 Degrees of Acoustical Privacy (Chanaud, 1983)

Degree of Privacy	Acoustical Condition	Possible Subjective Response
Confidential Privacy	**Cannot converse with others. Cannot understand speech of others. May not be aware of presence of others. May not hear activity sounds of others. Confidential conversations possible. No distractions.**	**Complete privacy. Sense of isolation. No privacy complaints expected.**
Normal Privacy	**Difficult to converse with others. Occasionally hear the activity sounds of others. Aware of the presence of others. Speech and machines audible but not distracting. Confidential conversations possible only under special conditions.**	**Sense of privacy. Some isolation. No privacy complaints expected.**
Marginal Privacy	**Possible to converse with others by raising voice. Often hear activity sounds and speech of others. Aware of each others presence. Conversations of others occasionally understood.**	**Sense of community. Sense of privacy weakened. Some privacy complaints expected.**
Poor Privacy	**Possible to converse with others at normal voice levels. Activity sounds, speech, and machines will be continually heard. Continually aware of each others presence. Frequent distractions.**	**Sense of community. Loss of privacy. Some loss of territory. Privacy complaints expected.**
No Privacy	**Easy to converse with others. Machine and activity sounds clearly audible. Total distraction from other tasks.**	**Sense of community. Sense of intrusion on territory. No sense of privacy. Many privacy complaints expected.**

3.5 ANNOYANCE

Noisiness

Comparative systems, similar to those used in the judgment of loudness, have been developed to measure *noisiness*. Subjects were asked to compare third-octave bands of noise at differing levels based on a judgment of relative or absolute noisiness. Somewhat surprisingly, the results shown in Fig. 3.21 (Kryter, 1970) differ from a loudness comparison. Relative noisiness is described by a unit called noys, a scale that is linear in noisiness much like the sone is for loudness. It is converted into a decibel-like scale called *perceived noise level* (PNL) with units of PNdB, by requiring the noisiness to double every 10 dB. The perceived noise level scale is used extensively in the evaluation of aircraft noise. The work of other investigators: Ollerhead (1968), Wells (1967), and Stevens (1961) is also shown in the figure.

Noisiness is affected by a number of factors that do not influence loudness (Kryter, 1970). Two that do affect loudness are the spectrum and the level. Others that do not

FIGURE 3.20 Level of Privacy vs Articulation Index (Chanaud, 1983)

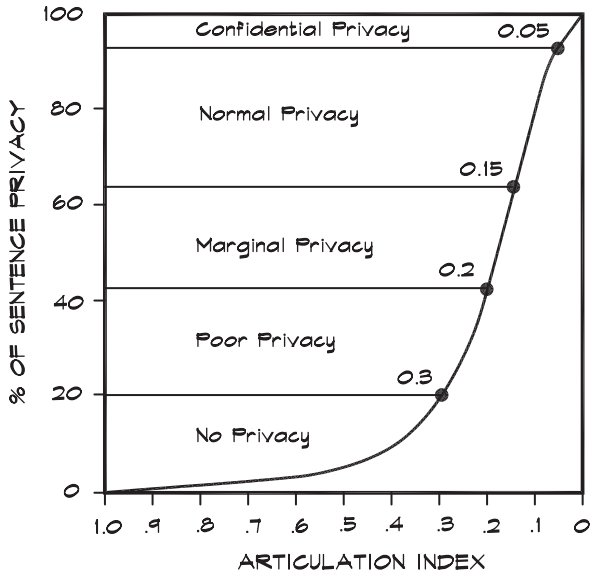
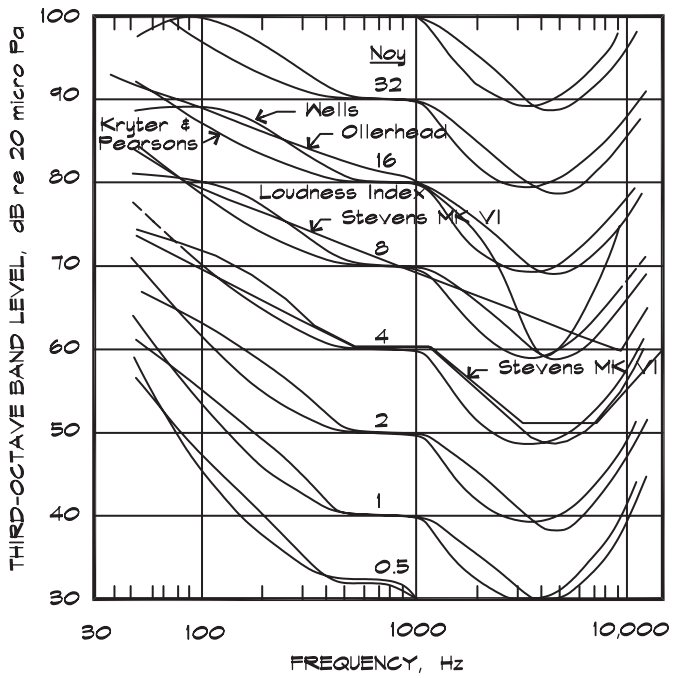


FIGURE 3.21 Equal Noisiness Contours of Various Authors (Kryter, 1970)



include: 1) spectrum complexity, namely the concentration of energy in pure tone or narrow frequency bands; 2) total duration; 3) in nonimpulsive sounds, the duration of the increase in level prior to the maximum level, called onset time; and 4) in impulsive sounds, the increase in level in a 0.5 second interval. Although these factors normally are not encountered in architectural acoustics, they contribute to various metrics in use in the United States and

in Europe, particularly in the area of aircraft noise evaluation. For further details refer to Kryter (1970).

Time Averaging - L_{eq}

Since the duration of a sound can influence its perceived noisiness, schemes have been developed to account for the tradeoff between level and time. Some of these systems are stated implicitly as part of a particular metric, while others appear in noise standards such as those promulgated by the Occupational Safety and Health Administration (OSHA) or in various noise ordinances.

To a casual observer the simplest averaging scheme would appear to be the arithmetic average of measured levels over a given time period. The advantage of this type of metric is that it is simple to measure and is readily understandable to the layman. Two disadvantages of the arithmetic average are: 1) when there are large variations in level it does not accurately account for human reaction to noise, and 2) for doing prediction calculations on moving sources it is enormously cumbersome.

When a sound level varies in time it is convenient to have a single number descriptor, which accurately represents the effect of the temporal variation. In 1953, Rosenblith and K.N. Stephens suggested that a metric be developed, which included frequency weighting and a summation of noise energy over a 24-hour period. A number of metrics have since evolved that include some form of energy summation or energy averaging. The system most commonly encountered is the equivalent level or L_{eq} (pronounced ell-ee-q), defined as the steady A-weighted level that contains the same amount of energy as the actual time-varying A-weighted level during a given period. The L_{eq} can be thought of as an average sound pressure level, where the averaging is based on energy. To calculate the L_{eq} level from individual sound pressure level readings, an average is taken of the normalized intensity values, and that average is converted back into a level. Mathematically it is written as an integral over a time interval

$$L_{eq} = 10 \log \frac{1}{T} \int_{t=0}^{T} 10^{0.1\, L(t)} \tag{3.5}$$

or as a sum over equal-length periods

$$L_{eq} = 10 \log \frac{1}{N\,\Delta t} \sum_{i=1}^{N} 10^{0.1\, L_i}\, \Delta t = 10 \log \frac{1}{N} \sum_{i=1}^{N} 10^{0.1\, L_i} \tag{3.6}$$

where L_{eq} = equivalent sound level during the time period of interest (dB)
$L(t)$ = the continuous sound level as a function of time
L_i = an individual sample of the sound level, which is representative of the i th time period Δt

It has been found that human reaction to time-varying noise is quite accurately represented by the equivalent level. L_{eq} emphasizes the highest levels that occur during a given time period, even if they are very brief. For example it is clear that for a steady noise level

that does not vary over a time period, the L_{eq} is the same as the average level L_{ave}. However if there is a loud noise, say 90 dBA for one second, and 30 dBA for 59 seconds, then the L_{eq} for the minute time period would be 72.2 dBA. The L_{ave} for the same scenario is 31 dBA. The L_{eq} is much more descriptive of the noise experienced during the period than the L_{ave}. When equivalent levels are used in environmental calculations, they are often based on a one-hour time period. When they begin and end on the hour they are called hourly noise levels, abbreviated HNL.

Twenty - Four Hour Metrics – L_{dn} and CNEL

One metric enjoying widespread acceptance is the L_{dn} or *day-night level*, which was recommended by the U.S. Environmental Protection Agency (EPA) for use in the characterization of environmental noise (von Gierke, 1973). The L_{dn}, or as it is sometimes abbreviated, the DNL, is a 24-hour L_{eq} with the noise occurring between the hours of 10 PM and 7 AM the next day increased by 10 dB before averaging. The L_{dn} is always A-weighted but may be measured using either fast or slow response.

$$L_{dn} = 10 \log \left\{ \frac{1}{24} \left[\sum_{i=8}^{22} 10^{0.1\, HNL_i} + (10) \sum_{i=23}^{7} 10^{0.1\, HNL_i} \right] \right\} \tag{3.7}$$

Each of the individual sample levels in Eq. 3.7 is for an hour-long period ending at the military time indicated by the subscript. The multiplier of 10, which is the same as adding 10 dB, accounts for our increased sensitivity to nighttime sounds.

Another system, the Community Noise Equivalent Level (CNEL), is in use in California and predates the day-night level. It is similar to the L_{dn} in that it is an energy average over 24 hours with a 10 dB nighttime penalty; however, it also includes an additional evening period from 7 P.M. to 10 P.M., with a multiplier of 3 (equal to adding 4.8 dB).

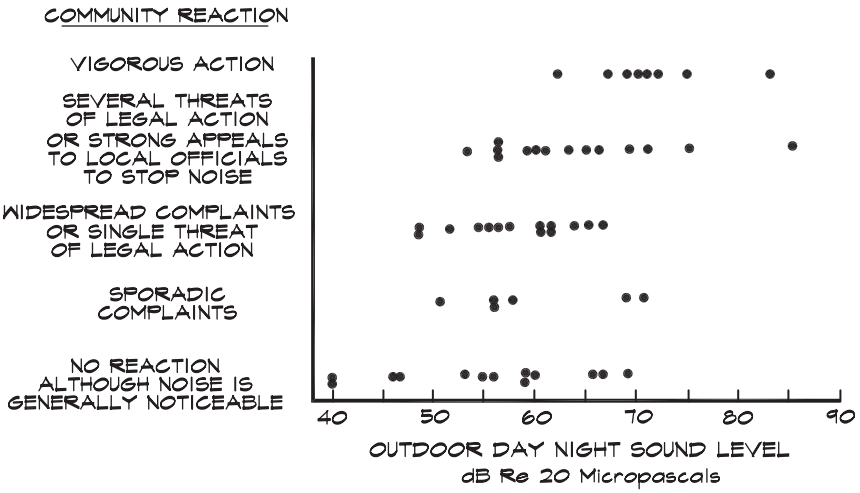
$$\text{CNEL} = 10 \log \left\{ \frac{1}{24} \left[\sum_{i=8}^{19} 10^{0.1\, HNL_i} + 3 \sum_{i=20}^{22} 10^{0.1\, HNL_i} + 10 \sum_{i=23}^{7} 10^{0.1\, HNL_i} \right] \right\} \tag{3.8}$$

Since the CNEL includes the extra evening factor it is always slightly higher than the L_{dn} level over the same time period. For most cases the two are essentially equal. Like the L_{dn}, the CNEL is A-weighted but is defined using the slow response.

Annoyance

The annoyance due to a sound can be highly personal. Any sound that is audible is potentially annoying to a given individual. Studies of annoyance have generally been based on the aggregate response of people exposed to various levels of noise. Much of the work in this field was done in the area of aircraft noise, and much of it is based on exterior noise levels. The U.S. EPA, following the mandate of the Noise Control Act of 1972, undertook a study of both the most appropriate metric to use for environmental noise and also the most appropriate levels. Since aircraft noise varies both from aircraft to aircraft and from day to day, the EPA study (von Gierke, 1973) recommended a 24-hour metric, namely the day-night level. They then developed recommendations on levels appropriate for public health and welfare (EPA Levels Document, 1974). Figure 3.22 shows part of the results of that study, specifically, the community reaction to exterior community noise of various types.

FIGURE 3.22 Community Reaction to Intrusive Noise (EPA Levels Document, 1974)



The data were very scattered. As a result of the wide variations in response an attempt was made to include factors other than level, duration, and time of day to normalize the results. A number of corrections were introduced, which were added to the raw day-night levels. These are listed in Table 3.7. Once the corrections had been applied, the data were replotted. The result is shown in Fig. 3.23, and the scatter is considerably reduced.

The final recommendations in the EPA Levels Document, for the levels of noise requisite to protect public health and welfare, are summarized in Table 3.8. It is interesting to note that the recommended exterior noise level of L_{dn} 55 does not guarantee satisfaction, and indeed according to the Levels Document it still leaves 17% of the population highly annoyed. Another interesting finding from this document, shown in Fig. 3.24, is that in one aircraft noise study, which related annoyance and complaints, the number of complaints lag well behind the number of people who are highly annoyed.

Satisfactory levels of interior noise are less well defined. Statutory limits in multifamily dwellings in California (CAC Title 24) are set at a CNEL (L_{dn}) 45 for noise emanating from outside the dwelling unit. It should be emphasized that statutory limits do not imply happiness. Rather they are the limits at which civil penalties are imposed. Many people are not happy with interior noise levels of L_{dn} 45 when the source of that noise is outside of their homes.

The 1987 ASHRAE guide suggests an NC 25 to 30 (30 to 35 dBA) as appropriate for residential and apartment dwellings. The EPA aircraft noise study (von Gierke, 1973) indicates that a nighttime level of 30 dBA in a bedroom would produce no arousal effects. Their recommendation of a maximum exterior L_{dn} of 60 dBA was based, in part, on a maximum interior level of 35 dBA at night with closed windows. The same reasoning, when applied to the Levels Document recommendations of L_{dn} 55 dBA, would yield a maximum nighttime level of 30 dBA with windows closed. Note that most residential structures provide about 20–25 dB of exterior to interior noise reduction with windows closed and about 10–15 dB with windows open. For purposes of this brief analysis maximum levels are taken to be 10 dB greater than the L_{eq} level. Van Houten (Harris, 1994) states that levels of plumbing related noise between 30 and 35 dBA in an adjacent unit can be "a source of

TABLE 3.7 Corrections to Be Added to the Measured Day-Night Sound Level (DNL) to Obtain the Normalized DNL (EPA Levels Document, 1974)

Type of Correction	Description	Correction (dBA)
Seasonal Correction	Summer (or year-round operation).	0
	Winter only (or windows always closed).	−5
Correction for Outdoor Noise Level Measured in Absence of Intruding Noise	Quiet suburban or rural community (remote from large cities and industrial activities).	+10
	Normal suburban community (not located near industrial activities).	+5
	Urban residential community (not located adjacent to heavily traveled roads or industrial activities).	0
	Noisy urban residential community (near relatively busy roads or industrial areas).	−5
	Very noisy urban residential community.	+10
Correction for Previous Exposure and Community Attitudes	No prior experience with intruding noise.	+5
	Community has had some previous exposure to intruding noise but little effort is being made to control the noise. This correction may also be applied in a situation where the community has not been exposed to the noise previously, but the people are aware that bona fide efforts are being made to control it.	0
	Community has had considerable previous exposure to intruding noise and the noise maker's relations with the community are good.	−5
	Community is aware that operation causing noise is very necessary and it will not continue indefinitely. This correction can be applied for an operation of limited duration and under emergency circumstances.	−10
Pure Tone or Impulse	No pure tone or impulsive character.	0
	Pure tone or impulsive character present.	+5

FIGURE 3.23 Community Reaction to Intrusive Noise vs Normalized DNL Levels (EPA Levels Document, 1974)

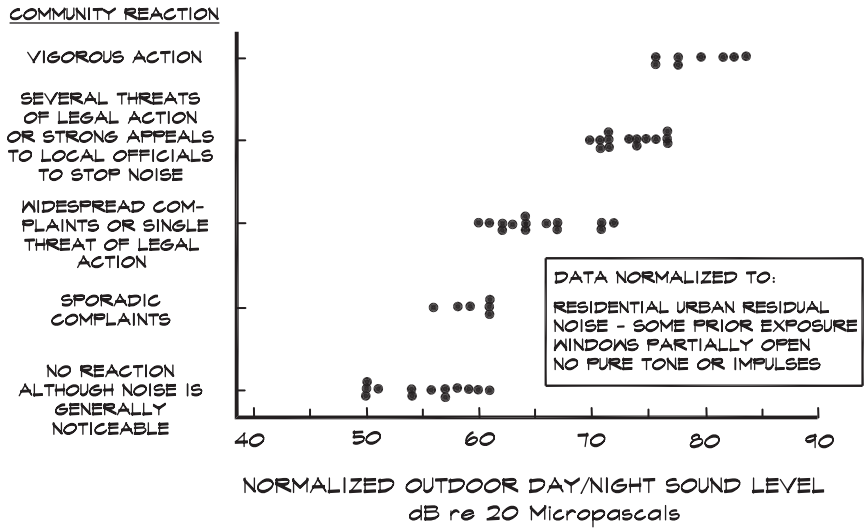
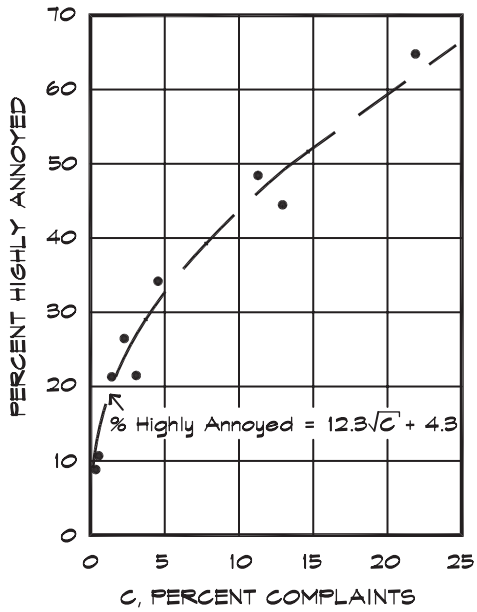


TABLE 3.8 Summary of Noise Levels Identified as Requisite to Protect Public Health and Welfare with an Adequate Margin of Safety (EPA Levels Document 1974)

EFFECT	LEVEL	AREA
Hearing loss	$L_{eq(24)} < 70$ dBA	**All areas**
Outdoor activity interference and annoyance	$L_{dn} < 55$ dBA	**Outdoors in residential areas and farms and other outdoor areas where people spend widely varying amounts of time and other places in which quiet is a basis for use.**
	$L_{eq(24)} < 55$ dBA	**Outdoor areas where people spend limited amounts of time, such as school yards, playgrounds, etc.**
Indoor activity interference and annoyance	$L_{eq} < 45$ dBA	**Indoor residential areas.**
	$L_{eq(24)} < 45$ dBA	**Other indoor areas with human activities such as schools, etc.**

concern and embarrassment." In the author's practice in multifamily residential developments, intrusive levels generated by activities in another unit are rarely a problem below 25 dBA. At 30 dBA they are clearly noticeable and can be a source of annoyance, and above 35 dBA they frequently generate complaints.

FIGURE 3.24 Percentage Highly Annoyed as a Function of Percentage of Complaints (EPA Levels Document, 1974)



3.6 HEALTH AND SAFETY

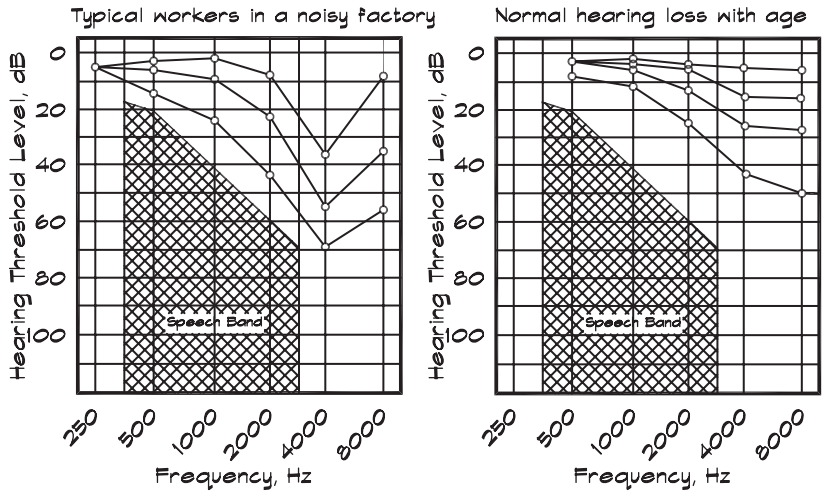
Hearing Loss

Noise levels above 120 dB produce physical pain in the human ear. The pain is caused by the ear's unsuccessful attempt to protect itself against sound levels about 80 dB above the auditory threshold by reducing its own sensitivity through use of the aural reflex. Exposure to loud noise damages the cochlear hair cells and a loss of hearing acuity results. If the exposure to noise is brief and is followed by a longer period of quiet, the hearing loss can be temporary. This phenomenon, called *temporary threshold shift* (TTS), is a common experience. The normal sounds we hear seem quieter after exposure to loud noises. If the sound persists for a long time at a high level or if there is repeated exposure over time, the ear does not return to its original threshold level and a condition called *permanent threshold shift* (PTS) occurs. The damage is done to the hair cells in the cochlea and is irreversible. The process is usually a gradual one that occurs at many frequencies, but predominantly at the upper end of the speech band. The loss progressively inhibits the ability to communicate as we age.

Human hearing varies considerably with age, particularly in its frequency response. In young people it is not uncommon to find an upper limit of 20 to 25 kHz, while in a 40 to 50 year old an upper limit of between 10 kHz and 15 kHz is more normal. Most hearing losses with age occur at frequencies above 1000 Hz, with the most typical form being a deepening notch centered around 3500 Hz. Noise induced hearing loss contributes to presbycusis, which is hearing loss with age. Figure 3.25 shows some typical hearing loss curves, one set for workplace noise induced loss, and the other for age-related loss. Scientists do not agree to what extent presbycusis should be considered "natural" and to what extent it is brought on by environmental noise.

A task group was appointed by the EPA to review the research on the levels of noise that cause hearing loss (von Gierke, 1973). It published, as part of its final report, the relationship

FIGURE 3.25 Progressive Hearing Loss Curves (Schneider et al., 1970)



between daily noise exposure and noise induced hearing loss for the most sensitive 10% of the population. Based on this and other data the EPA task force recommended no more than a day-night level of 80 dBA to protect the population from adverse health effects on hearing. The EPA Levels Document (EPA, 1974) went further and recommended an 8-hour L_{eq} level of no more than 75 dBA to protect public health for purposes of hearing conservation alone. These studies found no physiological effects for levels below 70 dBA.

The U.S. Occupational Safety and Health Administration (OSHA) has set legal standards concerning noise exposure of workers in the workplace. The legal limit is 90 dBA (as measured using the slow meter response) for an 8-hour workday with a 5 dBA per time halving tradeoff. This means that a worker may be exposed to no more than 85 dBA for 16 hours, 90 dBA for 8 hours, 95 dBA for 4 hours, 100 dBA for 2 hours, 105 dBA for 2 hours, 110 dBA for 1 hour, or 115 dBA for any time. The OSHA standard exposure is based on a finding by the American Academy of Otolaryngology—Head and Neck Surgery (AAO-HNS) that a hearing loss is significant only when the average hearing threshold at 500, 1000, 2000, and 4000 Hz has increased 25 dB. Clearly OSHA standards are much less restrictive than EPA recommendations, which were established without consideration of cost.

Workplace noise limits are characterized in terms of a noise dose, which is a relative intensity multiplied by a time, expressed as a percentage of an allowable limit. Over a given time period the dose can be calculated from levels, L_i measured in a series of intervals T_i

$$D = (100/T_n) \sum_{i=1}^{N} (T_i)\ 10^{\left[(L_i - L_c)/(10)\,q\right]} \tag{3.9}$$

where D = noise dose expressed as a percentage of the allowable daily dose
T_n = normalization time period, usually 8 hours
T_i = duration of the ith time period (hours)

L_i = A-weighted, slow meter response, sound pressure level for the ith time interval lying within the range of 80 to 115 dBA
L_c = criterion noise level, usually 90 dBA
q = nondimensional parameter, which determines the exchange rate over time; e.g., q = 5/(log 2) for a 5 dB per time halving exchange rate, q = 3/(log 2) for a 3 dB exchange rate
N = total number of intervals

The OSHA standards are expressed in terms of a total allowable level, which is based on 90 dBA over an 8-hour day and the 5 dB exchange rate. In these terms

$$L_{TWA} = 90 + (q)\log(D/100) \tag{3.10}$$

where L_{TWA} = time weighted average level (dBA) normalized to 8 hours
D = noise dose as defined above
q = exchange rate factor = 16.61 for the 5 dB rate

A 3 dB exchange rate assumes that hearing damage is proportional to the total energy of sound impacting the ear over a working lifetime. The 5 dB exchange rate allows more energy to impact the ear based on the understanding that the noise is intermittent and gives the ear some time to recover.

OSHA standards limit the time-weighted-average level to 90 dBA and requires that hearing protection be worn and that administrative and other controls on equipment be initiated. At 85 dBA OSHA requires that periodic hearing tests be performed on workers and that records on these tests be maintained.

3.7 OTHER EFFECTS

Precedence Effect and the Perception of Echoes

When a sound is reflected off a wall or other solid surface, the returning sound wave can be perceived as an echo under certain conditions. If the delay time between the initial sound and a second sound is decreased, the echo eventually disappears. A simple experiment can be carried out by clapping hands 15 meters (50 feet) or more away from a large flat wall and listening for the echo. At about 6 meters from the wall, the echo goes away and we hear a single sound. This is known as the precedence, or Haas effect (Haas, 1951), although its recognition predates Haas. The American scientist, Joseph Henry, demonstrated a similar effect at the Smithsonian in the 1840s (Davis and Davis, 1987). What happened in the hand clapping experiment is that the reflected sound finally fell below the level/delay threshold of perceptibility.

The perceptibility of a single echo is shown in Fig. 3.26 as measured in an anechoic space using speech at 70 dB and a loudspeaker placed in front. Figure 3.27 includes the results of the experiments of Haas, which were performed using a pair of loudspeakers, separated by 90°, placed in front of a listener. The source material was speech at different speaking rates presented at equal levels from each loudspeaker. One loudspeaker was delayed with respect to the other and the subjects were asked whether they felt disturbed by the reflection or echo. When the delay exceeded about 65 msec an annoying echo was perceived. At delay times less than 50 msec, echoes were perceived, which were not annoying even in cases where the reflection was 5 to 10 dB stronger than the primary sound (Blauert, 1983).

FIGURE 3.26 Absolute Perceptibility of a Delayed Signal (Kuttruff, 1973)

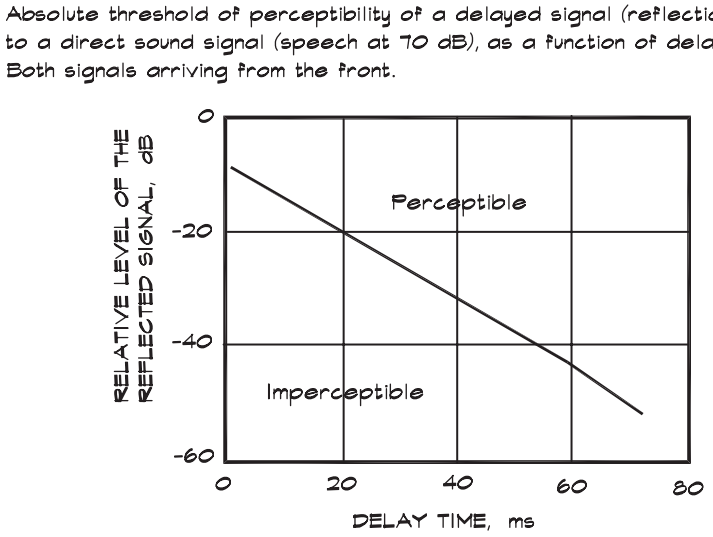
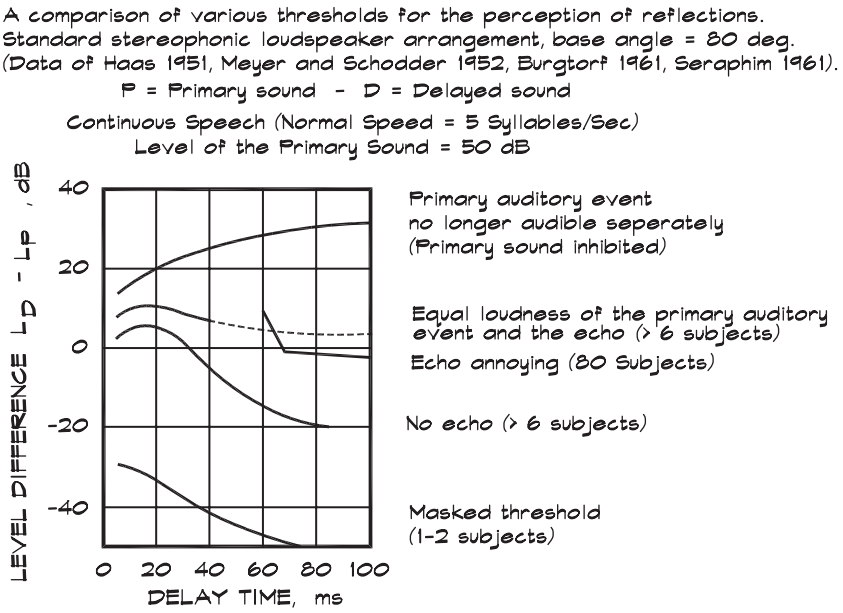


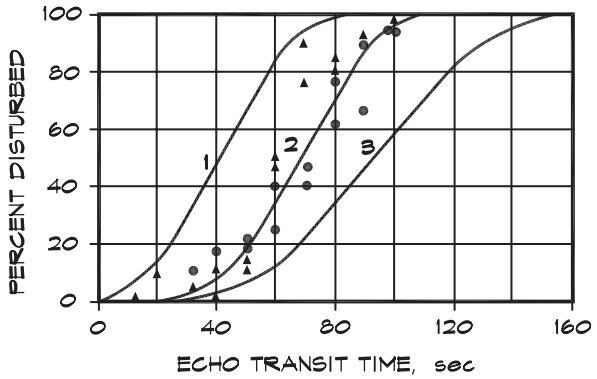
FIGURE 3.27 Thresholds for Perception of Reflections (Blauert, 1983)



Where the delayed sound was less than 30 msec after the initial event, the two sounds merged and no echo was perceived, even for sounds 5 or more dB above the initial sound.

The precedence effect is of considerable importance in architectural acoustics both for the natural reinforcement of live sounds coming from reflecting surfaces as well as for electronic reinforcement of speech or music. If a sound, originating from a performer on stage, is amplified and projected to an audience from a front loudspeaker, the image will

FIGURE 3.28 Disturbance Due to an Echo (Kuttruff, 1973)



Percentage of listeners disturbed by a delayed signal at the same level as the undelayed sound signal (speech). The abscissa is the delay time.

1. 7.4 syllables per second
2. 5.3 syllables per second
3. 3.5 syllables per second

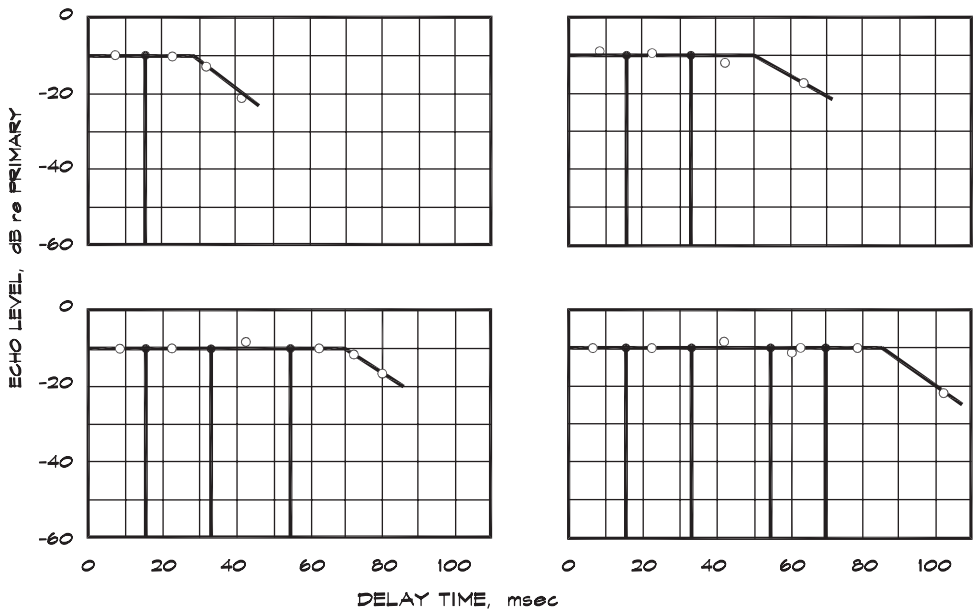
appear to come from the performer so long as the delay time is sufficiently short. The tradeoff between delay time and level, necessary to preserve the illusion of a single source, depends on the type of sound, i.e. speech or music, and the direction of the loudspeaker or reflector. If a reinforcing sound is within 25 msec of the initial sound, then speech is clearly understood. For music, a delay of 35 msec is normally not a problem even in rapidly played passages. For romantic music, delays as high as 50 msec can be tolerated. The three curves in Fig. 3.28 give the results of different rates of speaking on listener disturbance due to delayed echoes. The experiment is done with two loudspeakers, having the same level, placed in a relatively dead listening room (0.8 s reverberation time). The figure shows the importance of eliminating long-delayed reflections, and indeed even a single reflection, for the intelligibility of speech.

Data taken by Seraphim (1961), which were reproduced by Kuttruff (1973), led to the belief that when a series of delayed reflections arrives at a listener, the so-called Haas zone, where only a single sound is perceived, could be extended in time. Figure 3.29 gives the results of Seraphim's experiments using sounds coming only from the front in an anechoic environment. Here the threshold of perceptibility of the delayed sound is constant with delay time even when delays extend to 70 msec. Olive and Toole (1989) pointed out that this experiment, which was carried out in rather unrealistic conditions with all echoes having the same level, has been misapplied to reflections coming from many directions.

This is not to say that temporal forward masking does not occur for sounds having the same spectral content. Data published by Olive and Toole (1989) in Fig. 3.30 relate the absolute thresholds of perception for single lateral reflections to different types of source signal. In general the perception thresholds for speech and percussive reflections decrease with delay time, while those for music remain flat. The normal perception of a separate reflection, which would occur at a given delay time at much lower levels of reflected sound (Fig. 3.30), is masked. Experiments such as these indicate the importance of the smoothness

FIGURE 3.29 Absolute Perceptibility of a Delayed Signal (Kuttruff, 1973)

Absolute threshold of perceptibility of a delayed (reflection) being added to a sound field consisting of the direct sound plus one, two, three, and four reflections at fixed delay times and relative levels, which are denoted by the vertical lines. The original sound signal is speech. All sound components arriving from the front.



of the series of early reflections, following the arrival of the initial sound, in a critical listening space. These effects will be discussed in more detail in Chapt. 21.

Perception of Direction

The perception of direction is controlled by two factors: 1) the interaural delay time between the ears, and 2) the level difference created by the interaction between the head and the ears. When a sound originates to the left of the head in Fig. 3.31, it arrives at the left ear about a millisecond before the right ear. The high-frequency components are louder for the left ear than the right ear due to the shielding provided by the head itself. The brain uses the combination of level and time differences to decode the sound source direction.

When two sounds arrive at the listener the perceived direction is determined by first sound to arrive, even when the second sound is as much as 10 dB stronger. For equal level sources, delay gaps as low as one millisecond can bias the perceived direction to one side. This is called the precedence effect. Figure 3.32 shows the results of experiments (Madsen, 1970) on the tradeoff between time delay and intensity difference using two loudspeakers. At delay times below 1 msec, the higher level loudspeaker tends to dominate. Between 1 and 30 msec the precedence effect controls the direction, and above this point the two loudspeakers are increasingly perceived as separate sources.

When two sounds arrive at a listener simultaneously, the louder sound determines the direction. The apparent direction of a sound coming from two equidistant loudspeakers can be controlled by adjusting the level (panning) between them. Figure 3.33 contains the results of listening experiments performed by de Boer (1940) and Wendt (1963) by varying the level

FIGURE 3.30 Absolute Thresholds for a Single Lateral Anechoic Reflection (Olive and Toole, 1989)

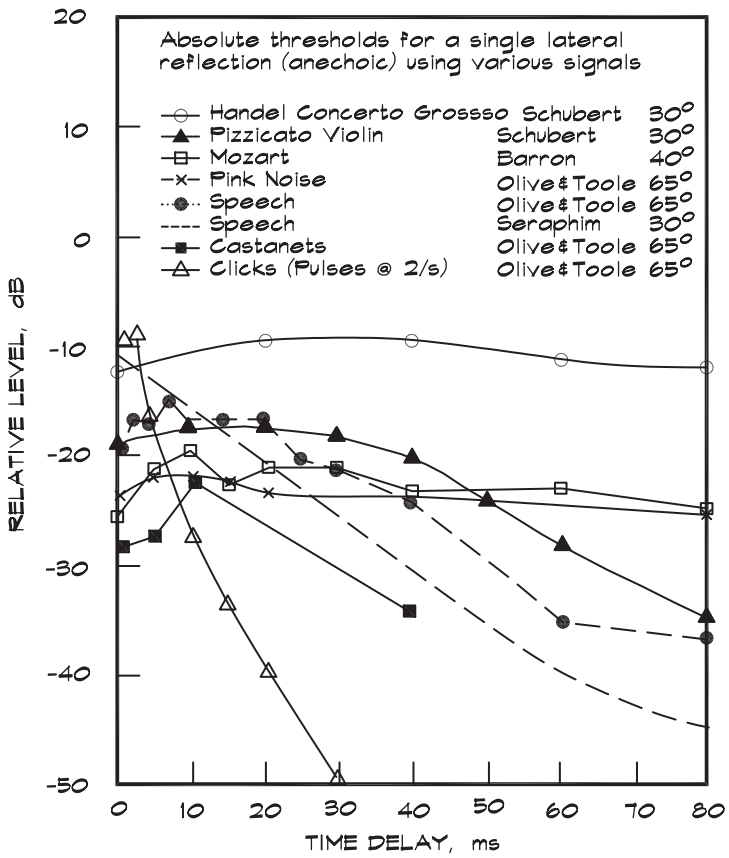
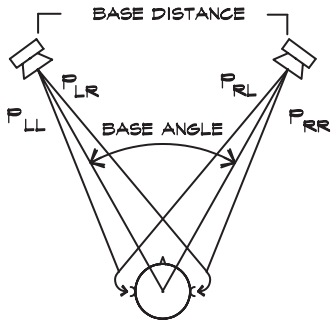


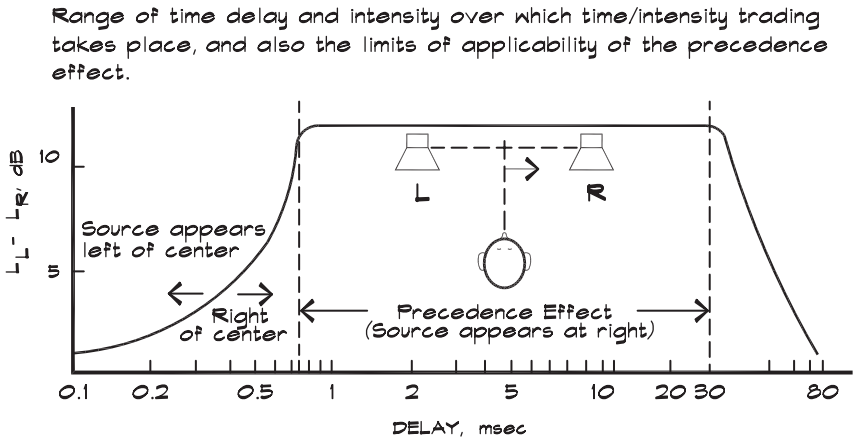
FIGURE 3.31 Head Geometry Relative to a Stereo Source (Blauert, 1983)



between two loudspeakers positioned 60° apart in the horizontal plane. When loudspeakers were placed 90° apart the error in the perceived direction increased significantly (Long, 1993). In most recording studios and mixdown rooms, loudspeakers spacing has standardized to a 60° spacing. Here the stereo image can be maintained and comfortably manipulated with panning.

In the vertical plane the ability of the brain to interpret time delays is much weaker, since our ears are on the sides of our heads. Results of localization tests in the vertical plane

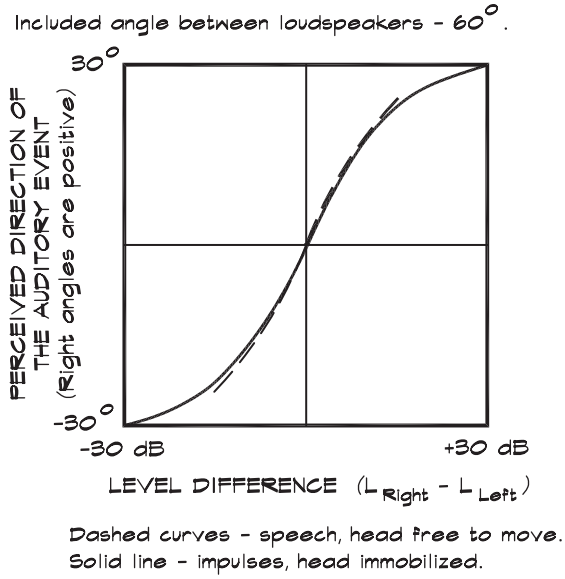
FIGURE 3.32 Perception of Source Direction with Delay (Madsen, 1970)



show a greater error and greater tolerance of wide loudspeaker placement. Our inability to precisely locate a vertical source makes realistic sound reinforcement systems possible. A properly designed loudspeaker cluster located above a stage can be used to augment the natural sound of the performers while maintaining the illusion that all the sound is coming from the stage.

The level-delay tradeoff has been carefully studied (Meyer and Schodder, 1952) by asking subjects to indicate the level difference at which the sound seemed to come from midway between a pair of stereo loudspeakers for various delays (Fig. 3.34). Study of this experiment is most helpful in the design of sound systems for it shows how far one can go in raising the loudspeaker level to augment the natural sound. It also shows how the apparent

FIGURE 3.33 Perception of Source Direction with Level (de Boer, 1940 and Wendt, 1963)



direction of sound can be moved about by using two loudspeakers and adjusting the time delay and level between them.

Clearly a stereo image, where a sound is perceived as originating between two loudspeakers, is difficult to maintain. The center image shifts to one side when one sound arrives only a few milliseconds earlier. Thus true stereo imaging is limited to a relatively small listener region close to the centerline between two carefully balanced loudspeakers. In a large room, such as a church or theater, a true stereo image can seldom be achieved.

Directional cues are best introduced by placing loudspeakers near the location of origin of the sound. For example, in motion picture sound systems, three loudspeaker clusters are arranged behind the screen in a left-center-right configuration, and the sound is panned to the proper level during the mix. In theme park attractions localization loudspeakers are placed in or near animatronic figures to provide a directional cue, even when most of the sound energy may be coming from a separate loudspeaker cluster.

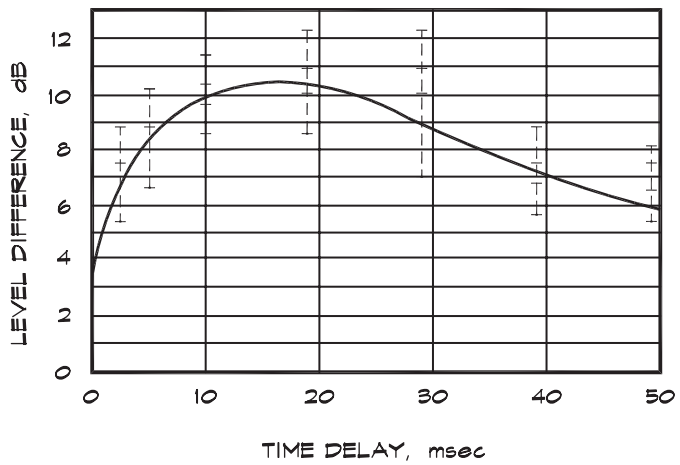
Binaural Sound

It is possible to reproduce many of the three-dimensional spatial attributes we hear in real life by recording sound using a dummy head with microphones in the ears and listening to the sound through stereo headphones, one for each microphone. This recording technique is referred to as dummy head stereophony or binaural reproduction, and is used in the study of concert hall design as well as in highly specialized entertainment venues. The results are startlingly realistic, particularly when the sound sources are located behind and close to the head.

When sounds are recorded binaurally, events that occur on the side or to the rear of our head are clearly localized. Sound sources located in front sound like they originate inside our head, overhead, or even behind. Several explanations for this phenomenon have been offered: 1) the effects of the pinnae are not duplicated when the playback system is a pair of headphones, 2) headphones affect the impedance of the aural canal by closing off the tube, and 3) the cues available from head motion are not present.

FIGURE 3.34 Equal Loudness Curve for Delayed Signals (Kuttruff, 1973)

Critical level difference between a delayed and undelayed signal which results in a perceived equal loudness of both signals (speech). (Meyer and Schodder, 1952)



This page intentionally left blank

4 ACOUSTIC MEASUREMENTS AND NOISE METRICS

4.1 MICROPHONES

Both microphones and loudspeakers are transducers—electromechanical devices for converting sound waves into electrical signals and vice versa. Microphones sense small changes in sound pressure through motion of a thin diaphragm. Cone loudspeakers create changes in pressure through the motion of a diaphragm driven by a coil of wire, immersed in a magnetic field. Since both microphones and loudspeakers operate in a similar manner, microphones can be used as loudspeakers and loudspeakers as microphones. Even the human eardrum can act as a loudspeaker.

The most common types of microphones in use are: 1) dynamic, 2) condenser, 3) electret, 4) ceramic, and 5) ribbon. All microphones consist of a diaphragm, which moves back and forth in response to changes in pressure or velocity brought about by a sound wave, and electronic components that convert the movement into an electric signal. Microphones are characterized by a sensitivity, which is the open circuit output voltage produced by a given pressure, expressed in decibels re 1 V/Pa. A one-inch diameter instrumentation microphone might produce 54 mV for an rms pressure of 1 Pa, yielding a sensitivity of $20 \log [(54\,\text{mV}) / (1\,\text{Pa})] [(1\,\text{Pa}) / (1\,\text{V})] = -25\,\text{dB}$. Note that 1 Pa is the sound pressure that corresponds to the 94 dB sound pressure level generated by standard pistonphone calibrators.

A dynamic microphone, illustrated in Fig. 4.1, operates on the same principal as a loudspeaker. A diaphragm moves in response to the changes in sound pressure and is mechanically connected to a coil of wire that is positioned in a magnetic field. The induced current, produced by the motion of the coil, is the microphone's output signal. Both the diaphragm and the coil must be very light to produce adequate high-frequency response. Most dynamic microphones produce a very low output voltage; however, since the electrical output impedance is low, the microphone can be located relatively far away from the preamplifier. Dynamic microphones are rugged and are primarily used in sound reinforcement applications, where low fidelity is good enough. One manufacturer of dynamic microphones used to demonstrate its product's toughness by using the side of it to pound a nail into a block of wood.

A condenser microphone, in Fig. 4.2, consists of a thin stretched stainless-steel diaphragm that is separated from a back plate by a narrow air gap. The two parallel plates

FIGURE 4.1 Dynamic (Moving Coil) Microphone (Rossing, 1990)

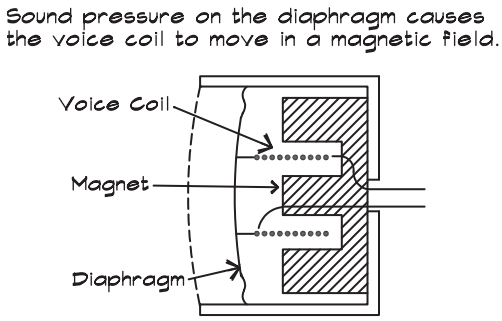
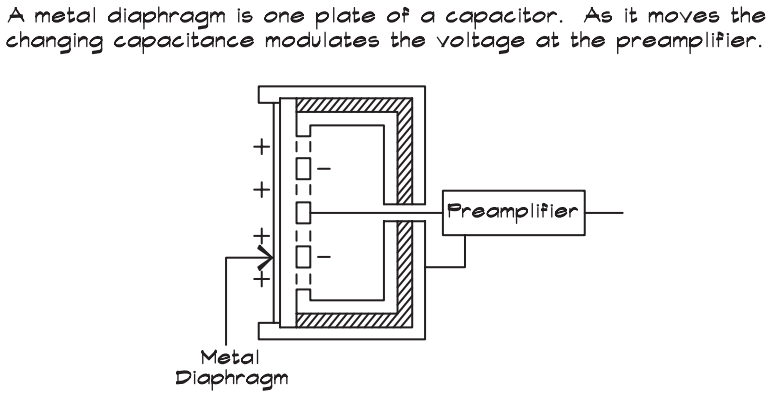


FIGURE 4.2 Condenser Microphone (Rossing, 1990)



become a capacitor when a DC polarizing voltage, typically 150 to 200 V, is applied. Motion of the diaphragm generates an electrical signal by varying the capacitance and thus the voltage between the plates. These microphones are very sensitive and accurate and have excellent frequency response characteristics. They are less rugged than dynamics and require a source of the polarizing voltage.

An electret microphone, in Fig. 4.3, is another form of condenser, which is sometimes called an electret condenser. It includes a thin polymeric diaphragm, where the polarizing voltage is not externally applied but is built into the polymer so that it is permanent. Otherwise the microphone operates in much the same way as the condenser does.

The ceramic microphone, in Fig. 4.4, has a diaphragm that is mechanically coupled to a piezoelectric material. A piezoelectric generates a voltage when strained. Many such materials exist such as lead zirconate titanate, called PZT, barium titanate, and rochelle salt. These microphones are more rugged than the capacitive types, are less sensitive, and do not require an external polarization voltage.

A ribbon microphone, sometimes referred to as a velocity microphone, works by suspending a thin metallic foil in a magnetic field. Figure 4.5 shows an example. The conducting ribbon is light enough that it responds to the particle velocity rather than the pressure. Since the ribbon is open to the back and shielded on the sides by the magnet, these microphones have a bidirectional polarity pattern. Ribbon microphones are very sensitive to moving air

FIGURE 4.3 Electret Condenser Microphone (Rossing, 1990)

A thin metallized plastic diaphragm is tightly stretched across a perforated backing plate. The holes in the back plate couple to an air cavity.

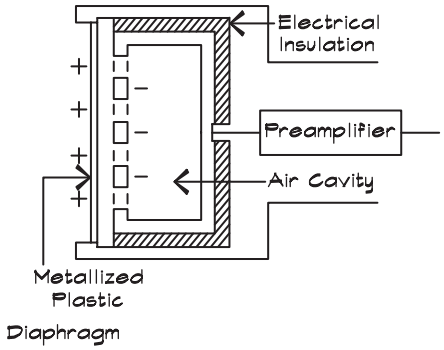


FIGURE 4.4 Ceramic Microphone (Rossing, 1990)

Sound pressure on the diaphragm causes deformation of the crystal, generating an electrical signal.

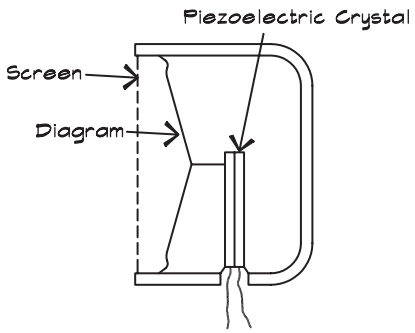


FIGURE 4.5 Ribbon Microphone (Rossing, 1990)

A lightweight ribbon diaphragm moves in a magnetic field, thus generating an electrical signal.

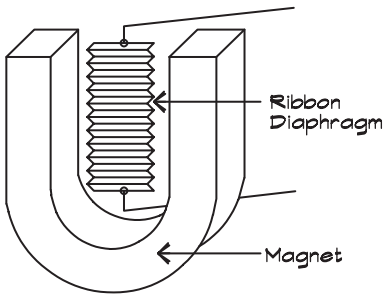
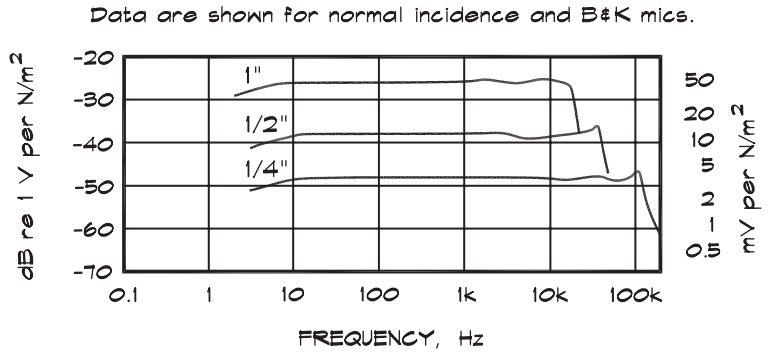


FIGURE 4.6 Sensitivity of Condenser Microphones (Hassall and Zaveri, 1979)



currents as well as high sound pressure levels. An unsuspecting acoustician, seeking to determine the characteristics of a reverberation chamber, once fired a blank pistol in a room full of ribbon microphones, quickly converting them into expensive paperweights. Due to the fragility of this type of microphone, its use is limited to the studio.

Frequency Response

Instrumentation microphones, so called because they can be calibrated using a pistonphone calibrator, are cylindrical and come in nominal sizes: one-inch (actually 0.936 in or 23.8 mm), half-inch (12.7 mm), and quarter-inch (6.5 mm) diameters. The size of a microphone affects its performance. Small microphones can measure sounds at higher frequencies and generally are less directional and less sensitive since they have a lower surface area. A one-inch instrumentation microphone, for example, might be able to measure levels as low as 0 dBA, while having an upper frequency limit of 10 kHz. A half-inch microphone might be good to 10 dBA and 30 kHz, and a quarter-inch microphone typically can measure down to 20 dBA and as high as 70 kHz. Examples of their response curves are given in Fig. 4.6.

Directional Microphones

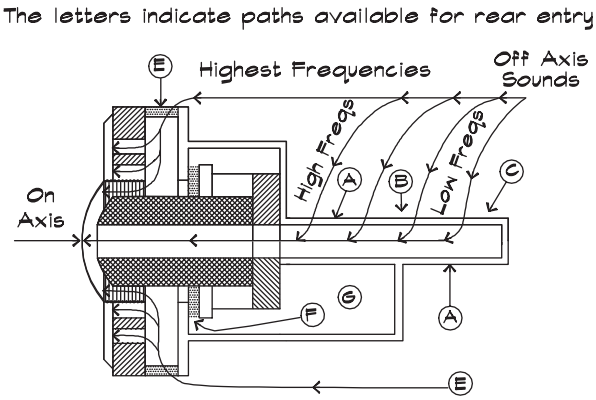
Microphones, like sound sources, can have a response that varies with angle, which is represented by a polar diagram with angles measured relative to the normal to the diaphragm. Ideally, instrumentation microphones are nondirectional; however, at high frequencies there is some self shielding and loss of sensitivity, which is often greatest at a 120° to 150° angle of incidence. The polar diagrams for several types of microphones are shown in Table 4.1. Directional microphones are not used for precision measurements, but are quite useful for recording and sound-reinforcement systems. When the microphone capsule is smaller than a quarter wavelength, it is not directional; however, directivity can be built in by manipulating the construction of the housing. Figure 4.7 illustrates the design of a cardioid housing. By leaving an opening at the rear, sound coming from the rear arrives at the front and back of the diaphragm at the same time, thus canceling. Sound arriving from the front takes some additional time to reach the rear of the microphone diaphragm. By carefully attenuating selected frequencies traveling along certain paths the sound entering the rear cavities can be delayed so that it arrives close to 180° out of phase and does not cancel out the frontal sound.

TABLE 4.1 Directional Characteristics of Microphones (Shure Inc., 2002)

	Omni-Directional	Bidirectional	Cardioid	Hypercardioid	Super-Cardioid
Polar Response Pattern					
Polar Equation	1	Cos θ	1/2(1 + Cos θ)	1/4(1 + Cos θ)	0.37 + 0.63 Cos θ
Pickup Arc 3 dB Down	360°	90°	131°	105°	115°
Pickup Arc 6 dB Down	360°	120°	180°	141°	156°
Relative Output At 90° (dB)	0	−∞	−6	−12	−8.6
Relative Output At 180° (dB)	0	0	−∞	−6	−11.7
Angle at Which Output = 0	−−	90°	180°	110°	126°
Random Energy Efficiency	0 dB	0.333 −4.8 dB	0.333 −4.8 dB	0.250* −6.0 dB	0.268** −5.7 dB
Distance Factor	1	1.7	1.7	2	1.9

* Minimum random energy efficiency for a first-order cardioid.
** Maximum front to total random energy efficiency for a first-order cardioid.

FIGURE 4.7 Cross-Section of an Electrovoice Variable-D Cardioid (Burroughs, 1974)



Highly directional microphones can be made using a series of openings in a tube, or a group of different length tubes, leading to the diaphragm. These so-called shotgun microphones work because sounds arriving on axis and entering through the holes combine in the tube in the proper phase relationship. Sounds arriving from the side and traveling down the tube combine with a random phase relationship that attenuates the signal at the diaphragm.

Directional microphones are very important in sound reinforcement systems. They selectively amplify sound coming from one direction, ideally from the user, and attenuate sound from other directions. This reduces feedback and allows a greater system gain. Properly designed directional microphones should have a consistent directivity pattern over a range of frequencies, otherwise they would color the off-axis sound. The more directional a microphone, the greater the coloration and the greater the directional lobing. Sometimes highly directional microphones can generate more system feedback than cardioid microphones, due to the influence of off-axis lobing patterns. In general, the less directional the microphone the more natural sounding it is.

Sound Field Considerations

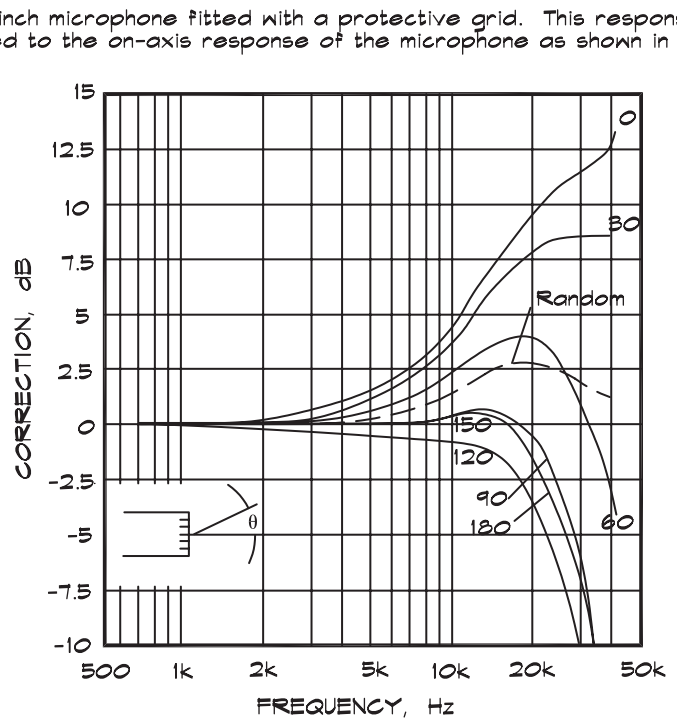
Microphone directivity sometimes influences the method of making measurements, even with instrumentation microphones. Typical microphones have their greatest sensitivity for sound incident on the diaphragm at 0°, called normal incidence. When the sound is traveling in a direction that is parallel to the plane of the diaphragm, at 90° to the normal, it is called grazing incidence. Most microphones have an angle for which their response is the flattest, usually 0° or 90°, but sometimes it can be another angle.

Microphones are described by their preferred type of sound field; for example, free field, random incidence, or pressure field. All microphones respond to pressure, but their sensitivity can be adjusted to produce the flattest response for a given angle of incidence or type of sound field. A free field is characterized by direct, unimpeded propagation of the wave from the source to the receiver. A diffuse or random field is one where the sound arrives from every direction with equal probability, and in a pressure field the sound pressure has the same magnitude throughout the space.

For a half-inch instrumentation microphone, below 5000 Hz all orientations produce a frequency response that is flat to within 2 dB. If a measurement is being made in a free field above 5000 Hz, the microphone should be oriented so that its flattest response direction is used, but this may vary with frequency, as can be seen in Fig. 4.8. Different standards organizations make different recommendations for proper free-field measurements (Fig. 4.9). IEC standards specify that the meter be switched to frontal mode and be oriented for normal incidence. ANSI standards require the selection of the random mode and an orientation of 70° to 80° to the source. For moving sources the microphone should be oriented for grazing incidence so that the directivity does not change with the motion of the source. This is achieved by angling the microphone upward.

When measurements are being done indoors, the random correction should be selected. Measuring with a free-field microphone in a diffuse field or with a random-incidence microphone in a free field yields only small inaccuracies, usually at high frequencies. The most accurate results will be obtained by using the setting appropriate to the type of sound field, but the differences are generally small.

FIGURE 4.8 Free Field Correction Curves for a Microphone (Bruel and Kjaer, 1986)



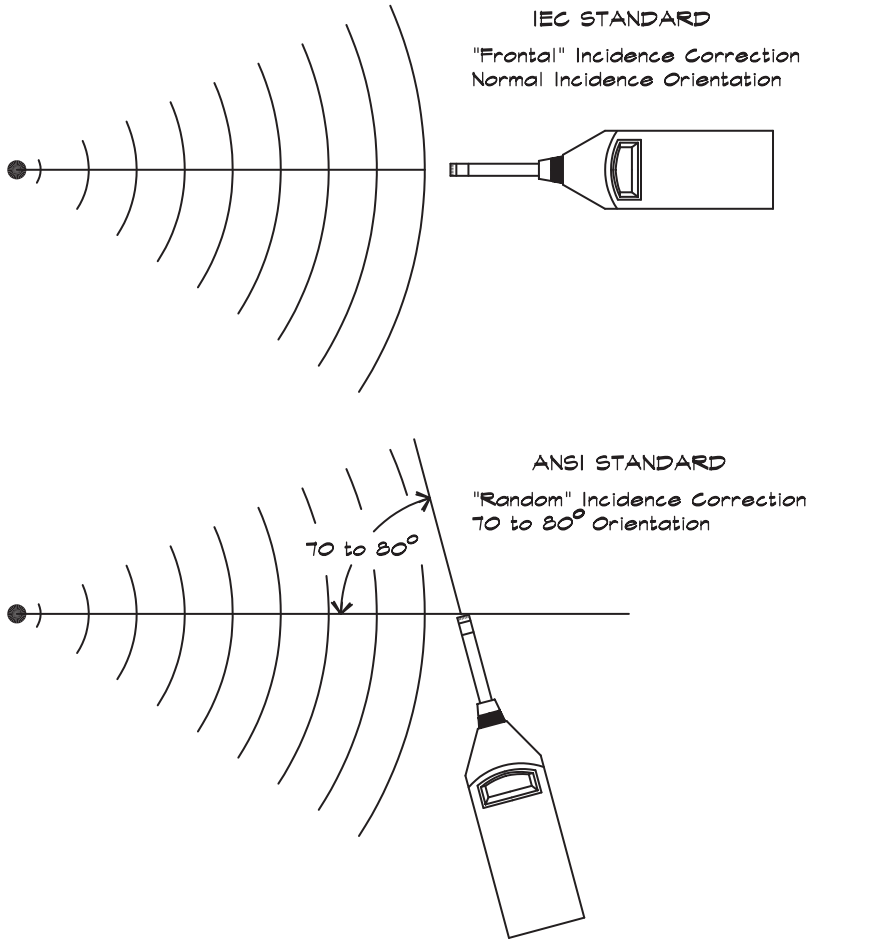
4.2 SOUND LEVEL METERS

The sound level meter, such as that shown in Fig. 4.10, is the fundamental acoustical instrument. Meters are battery powered and have become increasingly sophisticated, frequently containing internal processing, which automates many of the measurement functions. The individual controls vary from meter to meter; however, in general, there is a commonality of features. The basic controls allow for a selection of time weightings—fast, slow, and impulse—each of which represents a different ballistic time constant. Several frequency weightings are available: linear (unweighted), A-weighted, C-weighted, and a band limited linear scale. Frequency bandwidths may be selected from all pass, octave, and third-octave bands. There is a range selection that determines the highest and lowest levels measurable by the meter. Depending on the meter, there may be various types of automatic processing.

The internal parts of a meter include a microphone, preamplifier, various filters, a range control, time averager, and level indicator. The filters sometimes are contained in a separate module that may be attached to the meter, or are an integral part of the meter itself. On most hand-held sound level meters the filter selection is made manually. Where a group of filters operate simultaneously and display a number of levels on a bar graph in real time, the meter is called a spectrum analyzer or real-time analyzer.

Sound level meters are classified into three different groups by accuracy. Each class has a slightly different tolerance allowed in its precision. These standards are defined by the

FIGURE 4.9 Free Field Sound Measurements (Bruel and Kjaer, 1986)



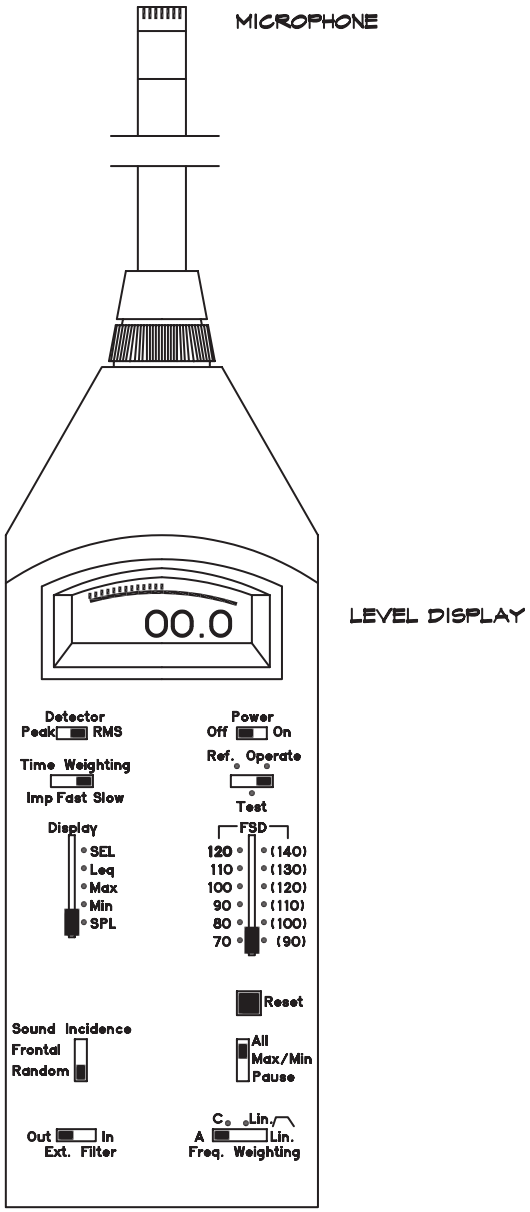
American National Standard Specification for Sound Level Meters, ANSI S1.4-1983.

Class 0	Laboratory	±0.2 dB 22.4 – 11200 Hz
Class 1	Precision	±0.5 dB 22.4 – 11200 Hz
Class 2	General Purpose	±0.5 dB 63.0 – 2000 Hz
		±1.0 dB 22.4 – 11200 Hz

Meter Calibration

Sound level meters should be calibrated before use, using a pistonphone calibrator placed over the microphone. These calibrators generate a steady tone, usually at 1000 Hz, by means

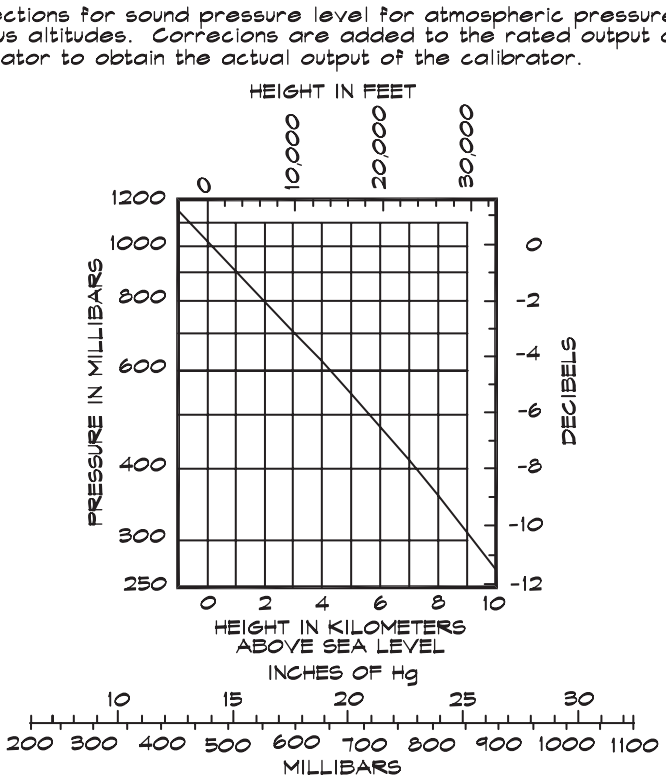
FIGURE 4.10 Sound Level Meter



of an oscillating piston in one end of a small cavity. The calibrator produces a nominal 94 dB, or with some calibrators a 114 dB, pure tone signal. The meter is adjusted to the proper level using a screw adjustment.

Pistonphone calibrators produce changes in volume in the cavity, which can be translated into changes in pressure using an equation of state. Most calibrators are set to produce the reference level at normal atmospheric pressure of 1013 millibars (1.01×10^5 Pa). Since atmospheric pressure varies, there is a correction given in Fig. 4.11 that must be applied according to altitude. This is the same correction as the term $10 \log(\rho_0 c_0/400)$ in Eq. 2.67, including a density that changes with altitude.

FIGURE 4.11 Sound Level Meter Calibration Corrections (Peterson and Gross, 1974)



Calibrators themselves should be calibrated periodically against a microphone of known sensitivity. Since microphones are used to calibrate calibrators and vice versa, we encounter a classic chicken and egg conundrum; that is, how do we calibrated the original reference? The original microphone must be calibrated using another microphone in what is called a reciprocity calibration. The microphones used are identical and both transducers are used as loudspeakers and microphones in this technique. Refer to Kinsler et al. (1982) for further details.

Meter Ballistics

Early sound level meters were equipped with a d'Arsinval galvanometer, which responds to a voltage and indicates the sound level with a needle pointer. These early meters were very sensitive and tended to chatter or move back and forth rapidly. Electrical damping was added, which slowed the needle's response and made it more readable. The choice of the damping resistor in the indicator circuit, along with the capacitance of the microphone, set the exponential time constant of the circuit. Three response speeds are now used—*slow, fast*, and *impulse*. The slow setting has a time constant of 1000 ms (1 second), while for fast response it is 125 ms. A time constant has a precise mathematical meaning in engineering. In one time constant the value rises to $(1 - 1/e)$ or falls to $1/e = 1/2.718$ of its steady value. If a sound is instantaneously raised to a certain level the meter will rise to within 2 dB of the actual level in one time constant. Standard practice is to use 200 ms tone bursts at 1000 Hz to test a meter's response, since real sine waves have a finite rise time. The fast meter response

must read within 2 dB of the steady level, and the slow meter response must be between 3 to 5 dB of the steady level (ANSI S4.1).

The rise time for fast and slow response is about the same as the fall time, so for integrating sound level meters, which measure L_{eq} levels, either fast or slow response gives about the same result. Some metrics, such as the CNEL level in California, require a particular response time, in this case, the slow response. For general use, the fast response is preferred. Impulse response is only employed to measure impact noise and other rapidly rising waveforms. The impulse time constant for a rising signal is 35 ms and for a falling signal is 1500 ms. Thus the meter holds the reading near its maximum level.

Meter Range

Sound level meters have an adjustable scale that allows the range of measurable levels to be set. If the range is set too low, then when a high level event occurs the meter will overload and not yield an accurate reading. If the range is too high, the indicated level will not fall below a certain value, and quiet events will not be measured accurately. Most meters have an overload indicator that signals the user to change the range. The range should be set as low as possible without tripping the overload indicator.

Detectors

There are two types of detector circuits found on most meters, peak, and rms (root mean square). Peak circuits sense the maximum amplitude present in the waveform. Mean-square detectors measure the time average of the square of the wave. Since the energy in the wave is proportional to the mean-square value, the rms detector is the most commonly used setting. Peak amplitudes are often of interest in vibration measurements. Peak-hold circuits, which capture the highest level during the measurement period, are utilized in the measurement of special sources such as sonic booms, where the wave shapes are not sinusoidal.

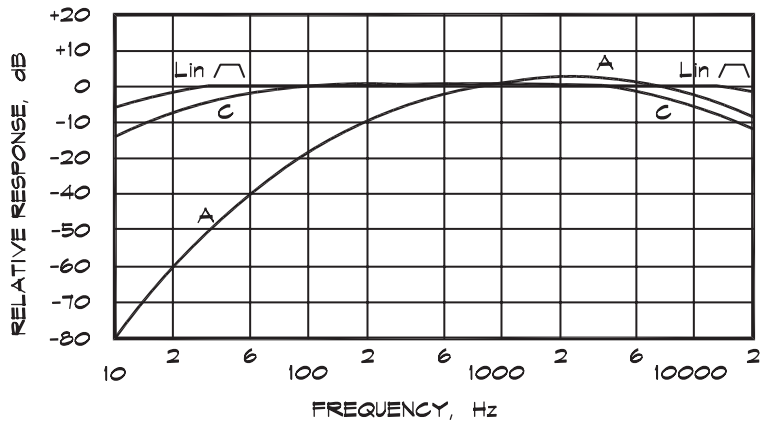
Filters

Sound meters come equipped with various selectable filters. The simplest is the linear filter, which passes sounds within the overall band limits of the instrument, for example 5 Hz to 100 kHz. This is not of particular interest in architectural acoustics, since it includes sounds that are well beyond our hearing capability. A second selection, the band-limited linear setting, includes a bandpass filter between 20 Hz and 20 kHz, and is quite useful for recording, since it blocks out low-frequency sounds that would otherwise overload a tape recorder. The characteristics of this filter along with the A and C weighting networks are shown in Fig. 4.12. Octave and third-octave bandwidth filters are also available. The standard frequency ranges have been given in Table 2.1. Filters may be cascaded, for example both octave band and A-weighting may be applied, yielding an A-weighted octave-band level. It is preferable to use the linear or band-limited linear settings when narrow-band filtering is done. This yields a consistent measurement methodology that does not require undue bookkeeping.

4.3 FIELD MEASUREMENTS

Field measurements are a critical part of architectural and environmental acoustics. Even with the simplest sources, care must be taken to follow proper procedure. A meter appropriate to the task must be selected. For environmental survey work a meter, tripod, calibrator, windscreen

FIGURE 4.12 A, C, and Lin Weighting Characteristics (Bruel and Kjaer, 1979)



(to reduce wind generated noise), logbook, distance measuring device (tape or rolling ruler), and watch are the standard kit. A small screwdriver is used to set the calibration. Spare batteries are a good idea. If they are left in the original packaging they can be distinguished from used ones. A camera is handy to record any unusual features of the site.

Headphones sometimes are included for listening to the sound being measured through the meter. They are essential for tape recording. Sometimes extraneous noise occurs that is not audible except through headphones. An example is arcing of the microphone, which can be caused by high humidity. Arcing produces a spurious popping sound that affects the data. Thus headphones are recommended when the relative humidity exceeds 90%.

For all measurements a record should be kept, noting the following information where it is relevant:

1) Location
2) Source description
3) Pertinent source details (e.g., manufacturer, model, operating point conditions)
4) Date and time
5) Engineer
6) Source dimensions and the radiating surfaces
7) Distance and direction to the source or a description of the measurement location
8) Meter settings
9) Background noise levels
10) Any unusual conditions
11) Time history
12) Measured data

Sources, which are outdoors and well away from reflecting surfaces, are the most straightforward. If the source is a piece of mechanical equipment the measurement position is selected based on the number of locations necessary to characterize the directivity of the source. For estimation of far-field levels from near-field measurements, data should be taken no closer than the largest dimension of the source, unless the area of the source is taken into account, by using Eq. 2.91.

The measurement distance for source characterization in a free field should be greater than a wavelength. For frequencies of 100 Hz the minimum distance is about 11 ft (3.4 m), while for 50 Hz the distance is about 22 ft (1.7 m). The danger of taking measurements too close is the possibility of including energy from only a portion of the source. If the source includes several separate pieces of equipment, the overall level will not be accurately represented if measurements are made too close to one individual component. Sometimes sound waves close to a source are not planar or are nonpropagating. Low-frequency emissions from large transformers are a good example of this type. Often low-frequency measurements require multiple samples and the microphone locations should be at least $\frac{\lambda}{4}$ apart.

Some sources are simply too large to conveniently get away from them. A good example is a refinery or a power plant. In such cases noise levels should be taken at regular distance intervals around the source and the results logged, according to where they were taken. Measurement locations should be spaced so that there is no more than a few decibels difference from one location to the next.

Measurements that are made to characterize a source rather than a location should be taken well away from reflecting surfaces. A minimum distance of $\frac{\lambda}{4}$ is recommended. If octave-band measurements are being taken and the 63 Hz band is of interest, then a distance of 4 to 5 feet is appropriate. Measurements will include reflections from the ground or other reflecting surfaces. Reflections from the observer can cause high-frequency comb filtering (Fig. 4.13), so the common practice is to hold the meter so that the microphone is extended away from the body or to support the instrument on a tripod.

An accurate measurement for source characterization is also difficult if the source receiver distance is too great. Even if the line-of-sight path is unimpeded, wind, atmospheric turbulence, ground cover, and air attenuation all play an important role in determining the measured noise level given off by a fixed source. At distances greater than 60 m (200 ft), noise level measurements can be dependent on wind velocity and direction. At distances greater than 150 m (500 ft), sound levels can be greatly influenced, even on a calm day, by ground cover, atmospheric turbulence, and air attenuation. At greater distances, thermal inversion layers can also be a major contributor. For all these reasons it is difficult to perform characterization measurements at large distances (say $>$ 60 m or 200 ft) from the source. Such measurements may be representative of a noise environment at a particular location under the measurement conditions, but may not be sufficiently accurate to characterize the source.

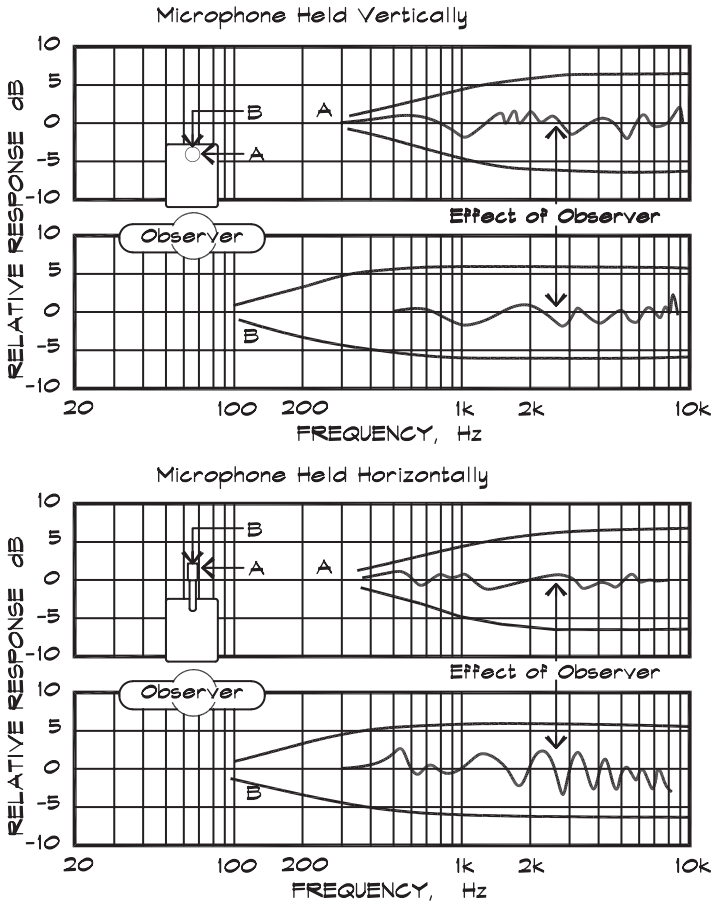
Background Noise

If there is a significant background (ambient) level present it should be measured. For a steady background it is best to turn off the source to be measured and note the ambient separately in all frequency bands of interest. The actual source-generated level then can be calculated from

$$L_{Source} = 10 \log \left[10^{0.1L_{Tot}} - 10^{0.1L_{Amb}} \right] \tag{4.1}$$

where L_{Source} = source sound pressure level (dB)
L_{Tot} = total combined source + ambient sound pressure level (dB)
L_{Amb} = ambient sound pressure level (dB)

FIGURE 4.13 Effect on Frequency Response as a Result of the Microphone Position (Petersen and Gross, 1974)



If the sound source cannot be turned off, it may be possible to measure the ambient noise level at a location that is similar to the location of interest but is away from the influence of the source. Locations may be available in shielded areas or, if the ambient noise is due to a roadway, at another site that is the same distance from the roadway.

When the background noise is variable and the source is steady, it is often easiest to measure the minimum combined level at a time when the ambient is quiescent. This gives an accurate source level if the ambient is sufficiently low. When the ambient is quiet, usually 10 dB below the source, its contribution can be ignored. With a varying ambient, if the source can be turned off, the minimum ambient can be recorded and then the minimum combined level measured. This gives a good value for the source level after adjustment using Eq. 4.1 as long as the minimum ambient levels are repeatable.

If the ambient is relatively steady and close to the source level, it can be measured separately using an averaging meter on the L_{eq} setting. The combined level then is measured in the same way and the source level calculated as before. This technique is also useful if the source, or background level, varies periodically, as it might with a pump motor or multiple sources such as fans or pumps, which produce beats. In taking data of this type, it is important to average over several beat cycles so that variations are properly taken into account.

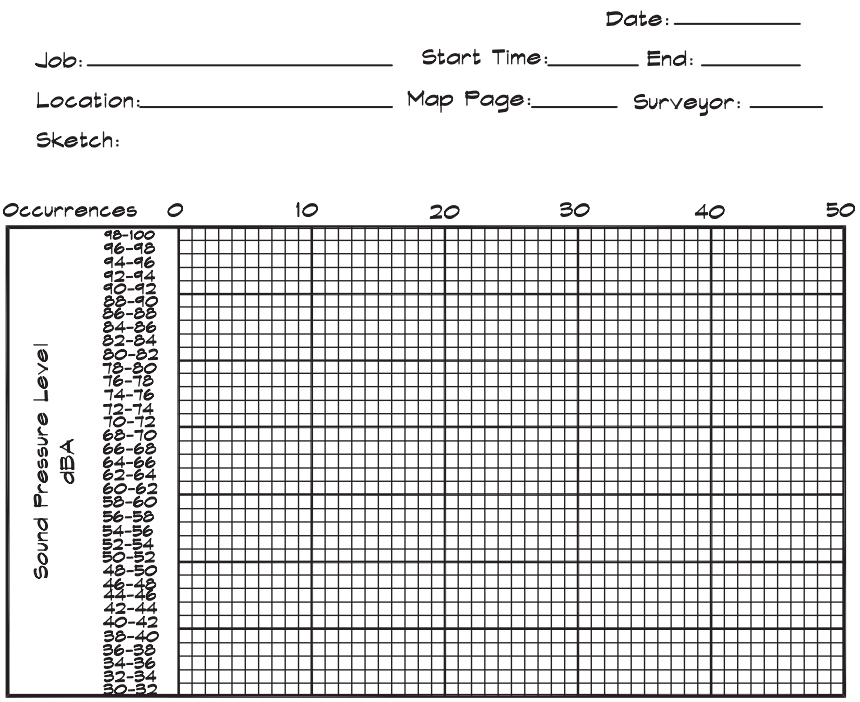
When the source level is less than the ambient, accurate measurements are difficult unless both the source and ambient levels are very steady. Even in these cases long averaging times are required to get good results. If the source is steady and the ambient varies, the minimum level gives the most accurate source level.

Time-Varying Sources

When traffic or other time-varying sources are to be measured, certain additional steps are useful. Although integrating meters are highly accurate, the nature of their output (i.e., one number) is sometimes not ideal, particularly when the data must be presented to a nontechnical audience. In these cases a log sheet such as that shown in Fig. 4.14 is helpful. In taking the data the meter is read at regular intervals, usually 5 or 10 seconds apart, and a notation is made on the log of the level that the meter shows at the interval mark. A representative number of samples are taken as determined either by the metric or the time period. One advantage to this methodology lies in the ability of the user to analyze the sampled data and extract more than one metric from the record. It also allows the engineer to ignore spurious signals such as barking dogs or aircraft flyovers that may not be relevant to the data being collected. Recording data, either on tape or in a recording sound level meter for later analysis, is another way of accomplishing the same goal. Data can be regularly sampled, and average levels calculated over a fixed time period and saved internally on a storage device for later analysis.

When a single moving source is to be measured, data are taken at a standard distance, say 15 m (50 ft), under prescribed conditions of velocity or acceleration. Data may be analyzed internally within the meter, or captured on a digital or analog recording device, or displayed

FIGURE 4.14 Noise Survey Log



graphically on a strip chart. When a recording is made, the calibration should flow through to all devices downstream of the meter. A tone is introduced using a pistonphone calibrator and is recorded along with the data. The meter range may then be adjusted by a known amount to accommodate the actual range of the data. A record should be made in a log or on the strip chart or verbally on tape noting the change in scale.

Both analog and digital recording devices can overload when signal levels exceed their dynamic range. When digital devices run out of headroom the resultant sound is most unpleasant. Analog tape recorders overload by producing a nonlinear or compressed version of the actual signal. If a two-channel device is available, the data may be recorded simultaneously on both channels at different level settings. This technique allows the data having the greater signal-to-noise ratio to be used, while retaining a margin of safety on the attenuated channel in case of overload.

Diurnal (24-Hour) Traffic Measurements

If a diurnal noise metric such as an L_{dn} or CNEL is to be measured, the ideal methodology is to position monitoring equipment at the location of interest for the entire 24-hour period. Often this is not practical due to the security, financial, or technical difficulties involved. In such cases a good estimate of the actual metric can be obtained by short-term monitoring if the hour by hour distribution of traffic is known or can be approximated. Measured distributions (Wyle, 1971) are given for urban traffic in Fig. 4.15 and for highway traffic in Fig. 4.16. The interesting feature about these data is that although they were taken 10 years apart they are almost identical. This implies that average diurnal traffic patterns are relatively stable.

If the reference L_{eq} level is known for the passage of one vehicle then the L_{eq} for N_h identical vehicles over the same time period is

$$L_{eq} = L_{ref} + 10 \log N_h \tag{4.2}$$

FIGURE 4.15 Typical Hourly Distribution of Total Daily Urban Vehicle Traffic (Wyle Laboratories, 1971)

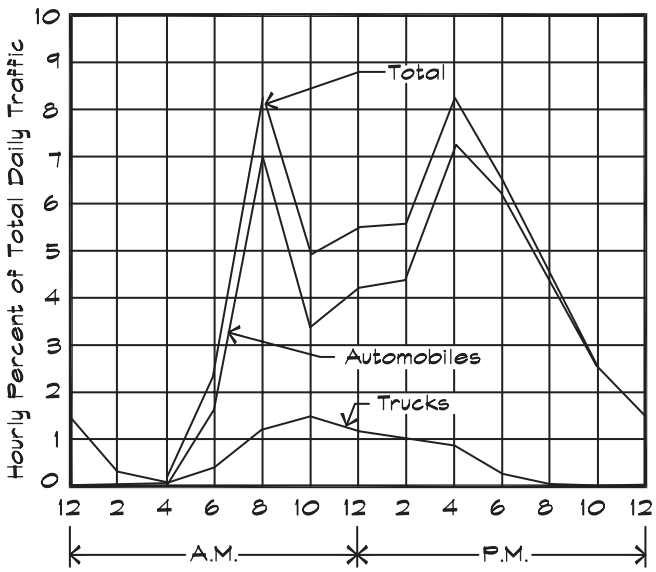
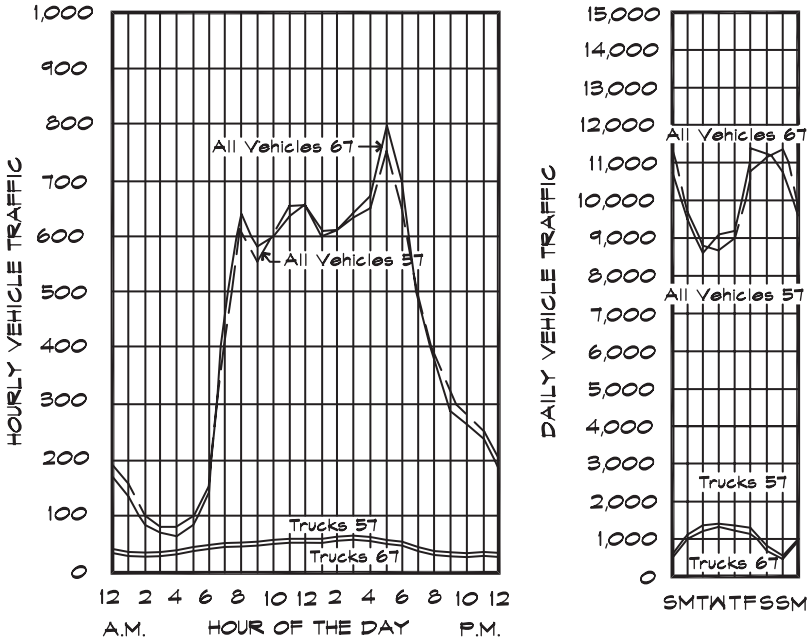


FIGURE 4.16 Hourly and Daily Variations in Intercity Highway Traffic in California (Wyle Laboratories, 1971)



Expressed in terms of vehicles per 10,000 annual daily traffic.
Data shown for 1957 and 1967.

where L_{eq} = equivalent sound level during the time period of interest (dBA)

L_{ref} = equivalent sound level for one vehicle passage during the time of interest (dBA)

N_h = number of like vehicles passing the measurement point during the time period of interest (usually one hour)

Assume that we can obtain the L_{eq} level for a given hour by direct measurement at a site. This can be accomplished by measuring over an hour period or by sampling the noise over a shorter time period and by assuming that the sample is representative of the hour period. Once the data have been obtained for the known hour, they can be adjusted for the time of day in which they were measured using standard distributions such as those in Figs. 4.15 and 4.16 or the actual site-specific traffic distribution, if it is known.

A traffic calculation uses a weighted hourly number of vehicles passing a point that yields the L_{dn} or CNEL level if inserted into Eq. 4.2. Thus

$$L_{dn} = L_{ref} + 10 \log N_{dn\,ave} \tag{4.3}$$

where L_{dn} = day night noise level (dBA)

L_{ref} = equivalent sound level for one vehicle passing by during an hour period (dBA)

$N_{dn\,ave}$ = weighted average number of like vehicles passing the measurement point during an equivalent hour

The day-night average number can be calculated from the distributions for urban and highway conditions using Eq. 3.7 for L_{dn} or Eq. 3.8 for CNEL.

$$N_{dn\,ave} = \left\{ \frac{1}{24} \left[\sum_{i=8}^{22} N_i + (10) \sum_{i=23}^{7} N_i \right] \right\} \tag{4.4}$$

where N_i = number of vehicles passing the measurement point during the i th hour

Finally by subtracting Eq. 4.2 and 4.3 we can obtain the difference in decibels between an L_{eq} level in any particular hour and the day-night level over a 24-hour period for a known traffic distribution.

$$L_{dn} \cong L_{eq}(h) + C(h) \tag{4.5}$$

where L_{dn} = day - night noise level (dBA)

$L_{eq}(h)$ = equivalent sound level for a given hour, h (dBA)

$$C(h) = 10 \log \frac{N_h}{N_{dn\,ave}}$$

= correction (dB) for the hour, h, based on the appropriate traffic distribution

The result is given in Table 4.2 for the Wyle urban and highway distributions for L_{dn}. The CNEL for these distributions is about 0.5 dB higher. If the traffic pattern at a particular site differs from those given here and is known, a similar calculation can be done for the specific distribution.

Included in these approximations is the assumption that the traffic speed and other factors that affect traffic noise, such as truck percentage, remain nearly the same over a 24-hour period. On crowded city streets this may not be the case. If traffic is free-flowing during the measurement period this method gives a conservative (high) estimate of the L_{dn} level. If traffic is slowed due to congestion, the noise levels will not be representative of a free-flowing condition.

If readings are taken during congested periods, the method will underestimate the actual 24-hour levels. If traffic slows significantly during rush hour, measurements made during off-peak periods, when traffic is flowing freely, will yield a result that is somewhat higher than the actual L_{dn} value.

The distribution of truck traffic over the day does not exactly track the automobile distribution. A similar calculation can be undertaken that includes truck percentages, with a knowledge of the difference between the reference level for trucks and cars. Naturally this introduces additional complexity. Based on 24-hour measurements, the method has been found to yield levels within one or two dB of the actual values, even without inclusion of a separate truck percentage distribution.

4.4 BROADBAND NOISE METRICS

At first glance the number and variety of acoustic metrics is overwhelming. In no other science are there as many different fundamental ways of measuring and characterizing the

TABLE 4.2 Approximate Conversion from Leq to Ldn or CNEL (Based on the traffic distributions shown in Figs. 4.15 and 4.16)

Hour	Highway Distribution		Urban Vehicle Distribution	
	$CNEL - L_{eq}$ (dB)	$L_{dn} - L_{eq}$ (dB)	$CNEL - L_{eq}$ (dB)	$L_{dn} - L_{eq}$ (dB)
1	8.2	7.7	10.9	10.4
2	10.4	9.9	15.1	14.6
3	11.2	10.7	16.9	16.4
4	11.6	11.1	19.9	19.4
5	10.6	10.1	9.5	9.0
6	8.2	7.7	6.5	6.0
7	3.6	3.1	2.6	2.1
8	1.6	1.1	0.8	0.3
9	2.0	1.5	1.7	1.2
10	1.9	1.4	2.9	2.4
11	1.6	1.1	2.7	2.3
12	1.5	1.0	2.6	2.2
13	1.9	1.4	2.5	2.1
14	1.8	1.3	2.4	1.9
15	1.6	1.1	1.5	1.0
16	1.4	0.9	0.8	0.3
17	0.7	0.2	1.3	0.8
18	1.3	0.8	1.8	1.3
19	2.7	2.2	2.7	2.2
20	4.0	3.5	3.6	3.1
21	5.0	4.5	4.3	3.7
22	5.3	4.8	5.6	5.1
23	5.9	5.4	6.9	6.4
24	7.1	6.6	7.8	7.4

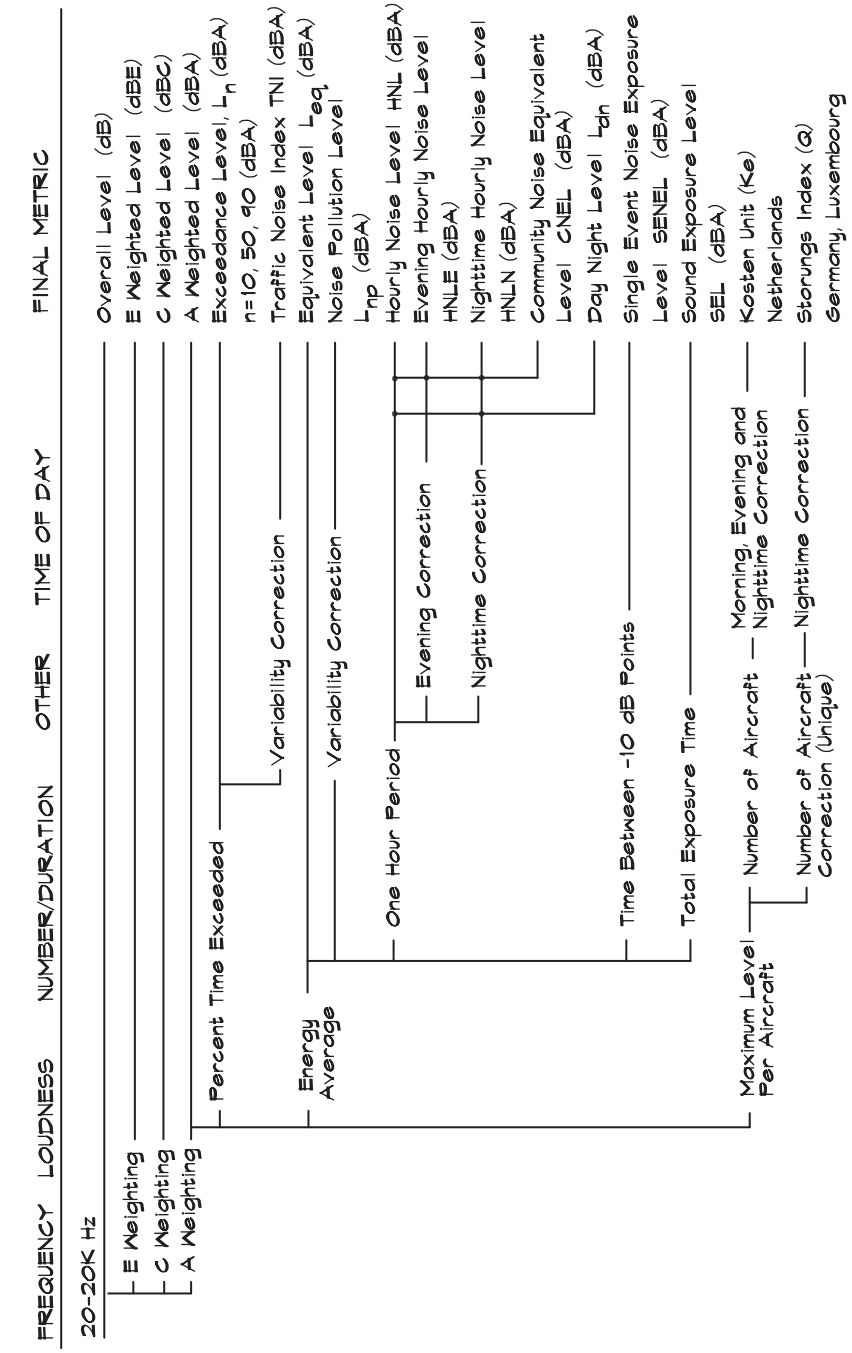
basic parameters. In physics the kilogram, meter, and second do not change. In electronics the volt, ohm, and ampere are stable and well defined. In environmental acoustics, however, different countries, states, cities, and counties often use different measurement schemes, which may not be directly convertible from one to another.

Even though the absolute number of metrics is large, the number of types of corrections applied to the measured data is rather modest. For example, a frequency correction for the loudness of a sound is included in most sound metrics but there are a number of ways to account for it, including A-weighting, NC curves, noys, and so on. The fundamental types of corrections include bandwidth, loudness, source number or duration, time of day, variability, onset, and pure tone content. The way each is included in a particular metric varies, but several are usually included in some fashion.

Bandwidth Corrections

The first correction category is the bandwidth of the measurement. Generally this is either wide band (i.e., 20 to 20 kHz) or band limited to octave or third-octave bandwidths. Narrow-band or chirped (swept) filters are also employed but the other corrections are seldom

FIGURE 4.17 Broadband Noise Metrics



applied to these measurements. Several metrics based on wide-band measurements are shown in Fig. 4.17. The loudness corrections in these measurements are applied by means of electronic filters, such as the A-weighting network, which are included in the meter itself. Subsequent corrections can be applied internally by the meter or can be added by a separate calculation.

Duration Corrections

One of the earliest metrics for describing traffic generated noise was the L_{10} (pronounced *ell-ten*) level. An L_n level is defined as the A-weighted sound level exceeded n % of the time during the measurement period. The L_{10} level is close to the maximum level occurring during a time period and its use reflects the fact that the highest levels are the most annoying. L_{10} levels are measured by using a histogram sampling technique, either manually or internally within the meter. If a histogram of measurements is made and there are 100 total samples, the L_{10} level is determined by counting 10 (10% of the total) measurements down from the highest level.

In a similar fashion the other excedance levels of interest can be determined. The L_{50} level or median is sometimes used. The L_{90} level is frequently used to characterize the residual background. L_n levels are expressed as whole numbers. From the statistical distribution of noise levels that can be characterized as normally distributed, certain relationships can be developed relating excedance levels to L_{eq} levels. For example, the energy average level, expressed in terms of the mean value (Barry and Reagan, 1978), is

$$L_{eq} = L_{50} + 0.115\,\sigma^2 \tag{4.6}$$

where L_{eq} = equivalent sound level (dB)
L_{50} = mean value sound level (dB)
σ = standard deviation of the sound levels (dB)

For a normal distribution, the L_{50} level and the L_{10} level are related

$$L_{10} = L_{50} + 1.28\,\sigma \tag{4.7}$$

The relationship between L_{10} and L_{eq} can be obtained

$$L_{eq} = L_{10} - 1.28\,\sigma + 0.115\,\sigma^2 \tag{4.8}$$

where L_{eq} = equivalent sound level (dB)
L_{10} = sound level exceed 10% of the time (dB)
σ = standard deviation of the sound levels (dB)

The standard deviation of highway traffic noise is usually 2 to 5 dB, so the L_{10} level is higher than the L_{eq} level. For traffic noise, the L_{eq} level is about equal to the L_{20} level. Not all outdoor noise distributions are normal, so these equations should be used carefully as general estimates of the actual values.

Variability Corrections

Metrics have been developed that include a term for the variability of the sound, the theory being that the more variable the sound distribution, the more annoying it is. The noise pollution level is one of these and is used to characterize community noise impacts. It is defined as

$$L_{NP} = L_{eq} + 2.56\,\sigma \tag{4.9}$$

where L_{NP} = noise pollution level (dBA)
L_{eq} = equivalent sound level (dBA)
σ = standard deviation of the sound levels (dBA)

Note that the noise pollution level uses A-weighting.

The traffic noise index (TNI) is another metric that includes a term for the variability of the noise environment. In this metric the variability is characterized in terms of the difference between the L_{10} and the L_{90} levels. The traffic noise index is given by

$$TNI = 4(L_{10} - L_{90}) + L_{90} - 30(\text{dBA}) \tag{4.10}$$

where TNI = traffic noise index (dBA)
L_{10} = level exceeded 10% of the time (dBA)
L_{90} = level exceeded 90% of the time (dBA)

Both the noise pollution level and the traffic noise index were developed for use in characterizing traffic noise and are not as accurate in predicting human reaction to other environmental noise sources.

Sound Exposure Levels

Metrics that utilize the format of energy times time are called exposure levels and are expressed in decibels with a reference period time of one second. There is considerable usefulness in such metrics in that they contain all the energy that occurs during a given event packed into a period one second long. The sound exposure level (SEL) is one such metric and is defined as

$$SEL = 10 \log \left[\sum_{i=1}^{N} 10^{0.1L_i} \right] \tag{4.11}$$

where SEL = sound exposure level (dBA)
L_i = sound level for a given one - second time period (dBA)
N = number of seconds during the measurement period

The SEL can be measured directly by many sound level meters. The meter can be set to display the SEL, which is internally computed, following the initiation of the measurement, by pushing the meter reset button. The L_{eq} can be calculated from the SEL for a given time period T

$$L_{eq} = SEL - 10 \log(T) \tag{4.12}$$

where SEL = sound exposure level (dBA)
L_{eq} = equivalent sound level for a given time period (dBA)
T = time (s)

When there are several events, the L_{eq} level can be calculated from the SEL levels for each event. The SEL levels are combined using Eq. 2.62 and the L_{eq} level is calculated using Eq. 4.12. If both the L_{eq} and the SEL are measured simultaneously, the measurement time period can also be calculated using Eq. 4.12.

Single Event Noise Exposure Level

The single event noise exposure level (SENEL) is similar to the SEL in that it sums the energy times the time associated with an event. Originally, it was developed to measure the noise energy of the flyby of a single aircraft. In such measurements it is sometimes difficult to tell when to begin and when to stop the readings. If the data are recorded on a strip chart or tape recorder it is unclear at what point on either side of the peak to stop adding up the energy. To short cut the process the SENEL was developed. This metric is the exposure level contained in the top 10 dB of a single event sound level record. The duration of the event in a SENEL is the time between the two points at which the level falls 10 dB below the maximum. Figure 4.18 shows the L_{eq} for a triangular sound pattern. The SEL or SENEL may be calculated from these L_{eq} levels by using an equation similar to 4.12, where the time period is equal to the pulse duration τ. Once the SENEL is known, the L_{eq} can be calculated for any period of time containing the event.

$$L_{eq} = \text{SENEL} - 10\log(T) \tag{4.13}$$

where SENEL = single event noise exposure level (dBA)
L_{eq} = equivalent sound level for a given time period (dBA)
T = time period for which the L_{eq} is to be calculated (s)

Note that it is necessary that the time period T in both Eqs. 4.12 and 4.13 be equal to or greater than the time period over which the SEL or SENEL was measured; otherwise, the event is not accurately represented. As with SEL, if several events occur within a given time period, then the individual SENEL levels may be combined using Eq. 2.62. An equivalent level can be calculated using Eq. 4.13 from the combined SENEL level.

4.5 BAND LIMITED NOISE METRICS

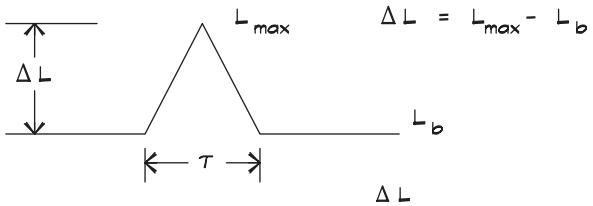
Techniques used for measurements employing octave band or other bandwidth filters vary little from those described for measuring broadband levels. Care must be taken in measuring low-frequency sounds so that the appropriate spacing between the source, reflecting surface, and the measurement location is observed. Sufficient sampling time is also a factor with low-frequency measurements because some low-frequency sources produce beat frequencies, which may be on the order of 1 Hz or less and may vary slowly over time.

Figure 4.19 shows a summary of the types of metrics obtained from octave-band measurements. As with the broadband systems there are a number of different metrics; however, the number of correction categories is relatively small. A loudness can be measured using electronic filters such as the A-weighting network. The A-weighted octave-band spectrum is useful as an aid in the determination of the frequency band making the most significant contribution to the overall A-weighted noise level. If most of the A-weighted energy is contained in one frequency band, then noise control efforts should be concentrated there.

A simple unweighted octave-band level is the basis for a number of metrics that determine the loudness by a direct comparison of the measured data to a standard curve of values. Several standards have been developed over the years, having to do principally with heating, ventilating, and air conditioning (HVAC) noise. The NC and RC curves are described in Chapt. 3.

FIGURE 4.18 L_{eq} Levels for Various Time Patterns (US EPA, 1973)

For triangular shaped patterns - time period = T



$$L_{eq} = L_b + 10 \log \frac{10}{2.3\,\Delta L} \left(10^{\frac{\Delta L}{10}} - 1\right)$$

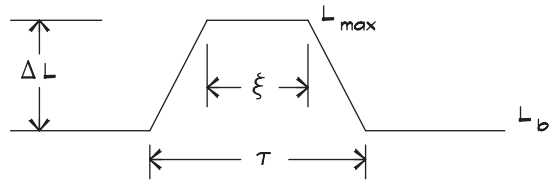
When ΔL is greater than 10 dB the following approximation is accurate.

$$L_{eq} = L_{max} - 10 \log \frac{2.3\,\Delta L}{10}$$

When there are a series of n identical triangular time patterns as shown above occurring within an interval T then

$$L_{eq} = L_b + 10 \log \left[1 + \frac{n\,T}{T} \left\{ \frac{\left(10^{\frac{\Delta L}{10}} - 1\right)}{2.3} - \frac{\Delta L}{10} \right\}\right]$$

For trapezoidial shaped patterns - time period = T



$$L_{eq} = 10 \log \frac{1}{\frac{(T - \xi)\,\Delta L}{10} + \frac{\xi}{2}} \left[10^{\frac{L_b}{10}} \frac{(T - \xi)}{2.3} \left(10^{\frac{\Delta L}{10}} - 1\right) + 10^{\frac{L_{max}}{10}} \frac{\xi}{2}\right]$$

When ΔL is greater than 10 dB and ξ is small compared to T.

$$L_{eq} = L_{max} - 10 \log \frac{2.3\,\Delta L}{10} + 10 \log \xi$$

When there is a series of n identical trapezodial time patterns as shown above occurring within an interval T then

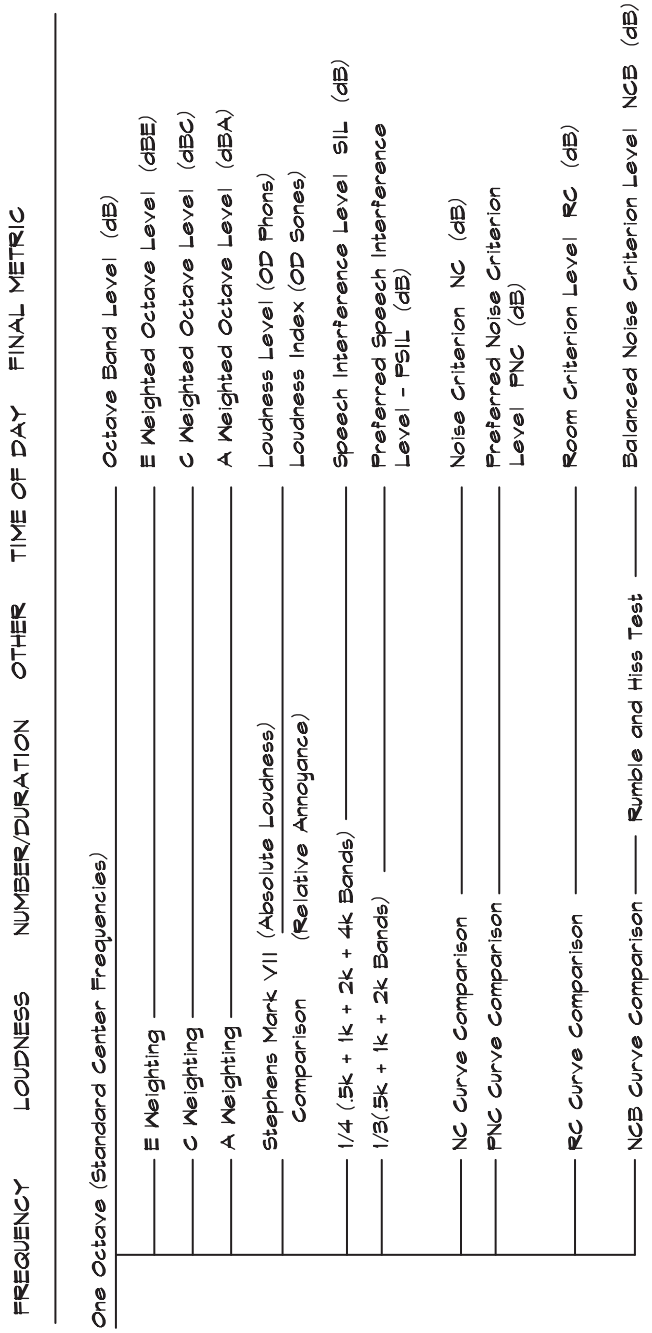
$$L_{eq} = L_{max} + 10 \log \frac{n\,T}{2.3\,T} + 10 \log n\,\xi$$

For a series of triangular pulses the last term above can be omitted.

Preferred Noise Criterion (PNC) Curves

PNC curves were introduced by Beranek in 1971 and are a revision of his earlier (1957) NC curves. PNC curves altered the high- and low-frequency octave values somewhat. The difference between the two has not been sufficient to result in the wide acceptance of the PNC version. The PNC curves are shown in Fig. 4.20. The use of the PNC curve is similar to that of the NC curve in that the PNC level is determined using the method of tangency.

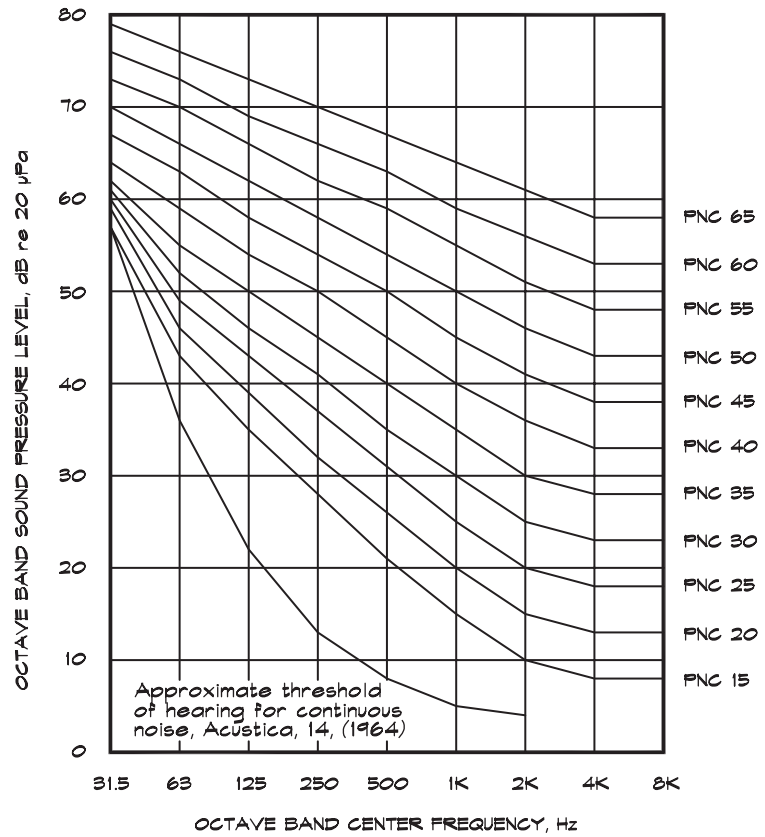
FIGURE 4.19 Octave Band Noise Metrics



Balanced Noise Criterion (NCB) Curves (Beranek, 1989)

In 1989, Beranek introduced another version of his 1957 NC curves, which he suggested for application to unoccupied rooms. These NCB levels, given in Fig. 4.21, are similar to the NC curves; however, the frequency range extends to 16 Hz and the metric calls for the calculation of the speech interference level (SIL) from the noise spectrum. This is rounded

FIGURE 4.20 Preferred Noise Criterion (Beranek, 1971)



to the nearest dB and compared with the NCB curve designation, which is also characterized by its SIL. If the measured SIL is equal to or below the curve designation, then the noise level meets the NCB criterion for speech interference.

Next comes a test for rumble or low-frequency energy. To check for this condition, 3 dB is added to the measured SIL and the NCB curve corresponding to this level is overlaid on the measured data. Where there are excedances of the new curve in the octave bands below 1 k Hz, they must be reduced to the elevated curve levels to comply with the standard.

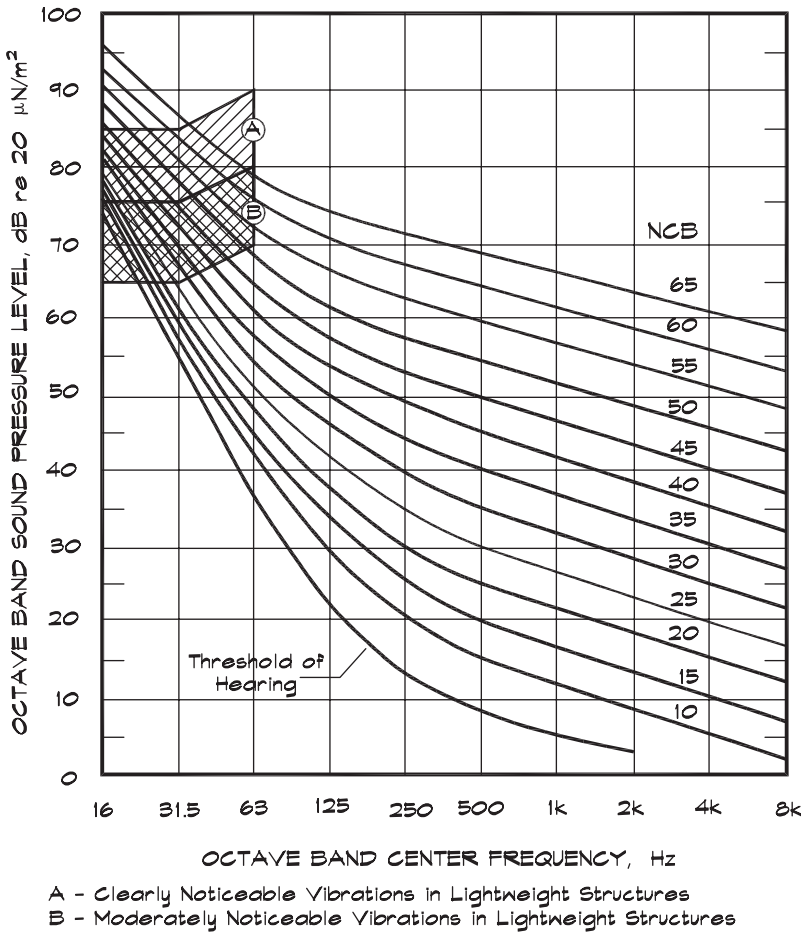
Finally there is the NCB test for hiss or high-frequency annoyance. An NCB curve is selected that provides the best fit to the measured data in the 125–500 Hz bands. Then this curve is plotted against the measured data. If the measured data exceed it in any of the three bands above 1 k Hz, then they must be reduced to meet the hiss criterion.

In occupied spaces Beranek calls for the measurement or estimation of noise levels due to normal work activities, which are to be combined with the unoccupied (HVAC) levels before a comparison to the NCB curve is made.

Other Octave-Band Metrics

Other systems exist for the determination of loudness based on measured octave-band data. They are based on empirical tests of relative or absolute comparisons presented to listeners in much the same way that the Fletcher-Munson experiments were done.

FIGURE 4.21 Balanced Noise Criterion Curves (NCB) (Beranek, 1989)



Robinson and Whittle (1962) constructed relative loudness curves in a very similar way. Stevens (1972) developed a series of systems for the calculation of loudness from octave band and other narrower bandwidth data. These systems rarely are encountered in architectural acoustics.

Octave-Band Calculations

It is frequently necessary to obtain an overall A-weighted level from unweighted octave-band data. The calculation is done by first adding the corrections for A-weighting, given in Table 3.1, to the level in each octave, and then by combining the A-weighted octave-band levels together using Eq. 2.62.

Occasionally it is necessary to generate an octave-band spectrum to match a given A-weighted level. This is straightforward if the spectrum shape of the sound source can be obtained. For example, let us assume that it is known that recorded music has a given octave-band spectrum and that this spectrum generates an overall A-weighted sound pressure level of 70 dBA. If we wish to obtain the octave-band spectrum of music that will yield an overall A-weighted level of some other level, for example 80 dBA, it is only necessary to add the difference between 80 and 70 to each octave-band level. It is assumed that the spectrum

shape does not change with level for this source. It is useful to prepare normalized spectra for standard sources, which, when added to the overall A-weighted sound level, will yield an unweighted octave-band level having the same overall value.

If there are two sources present at the same time and we know the octave-band spectrum levels of each source independently, the spectrum for the two sources combined is obtained by applying Eq. 2.62 to the pairs of levels in each octave.

Third-Octave Bandwidth Metrics

Third-octave band metrics are similar to octave-band levels—they are simply a thinner slice of the same pie. They can be combined into groups of three centered around the octave-band center frequencies using Eq. 2.62 to obtain octave-band levels.

A summary of various third-octave and narrow band metrics is shown in Fig. 4.22. As with the octave-band metrics there are different versions of loudness and annoyance comparisons. One of these, the perceived noise level (PNdB) developed by Kryter (1970), has been used as the basis for several of the standard metrics for characterizing aircraft noise.

Aircraft Noise Rating Systems

Aircraft noise ratings vary principally in the methodologies they use for adjusting for the number of aircraft, the addition of pure tone corrections, and the inclusion of nighttime penalties. An excellent review of aircraft metrics was prepared by (Schuller et al., 1995). He summarizes the descriptors using the equation

$$\text{Level} = A \log \left(\sum_{i=1}^{N} n_i \, w_i \, 10^{L_i/B} \right) - C \tag{4.14}$$

where A, B, C = constants
i = aircraft type category index
N = total number of aircraft type categories
n_i = number of noise events for aircraft category i per 24 - hour day
w_i = penalty (or weighting) factor for aircraft operation i
L_i = single event noise level for aircraft category i

The parameters used in Eq. 4.14 for various environmental metrics are given in Table 4.3.

Most of the metrics used for aircraft correlate well with the simpler L_{dn} level, which is the most commonly used system in the United States. For estimation purposes the following formulas may be used:

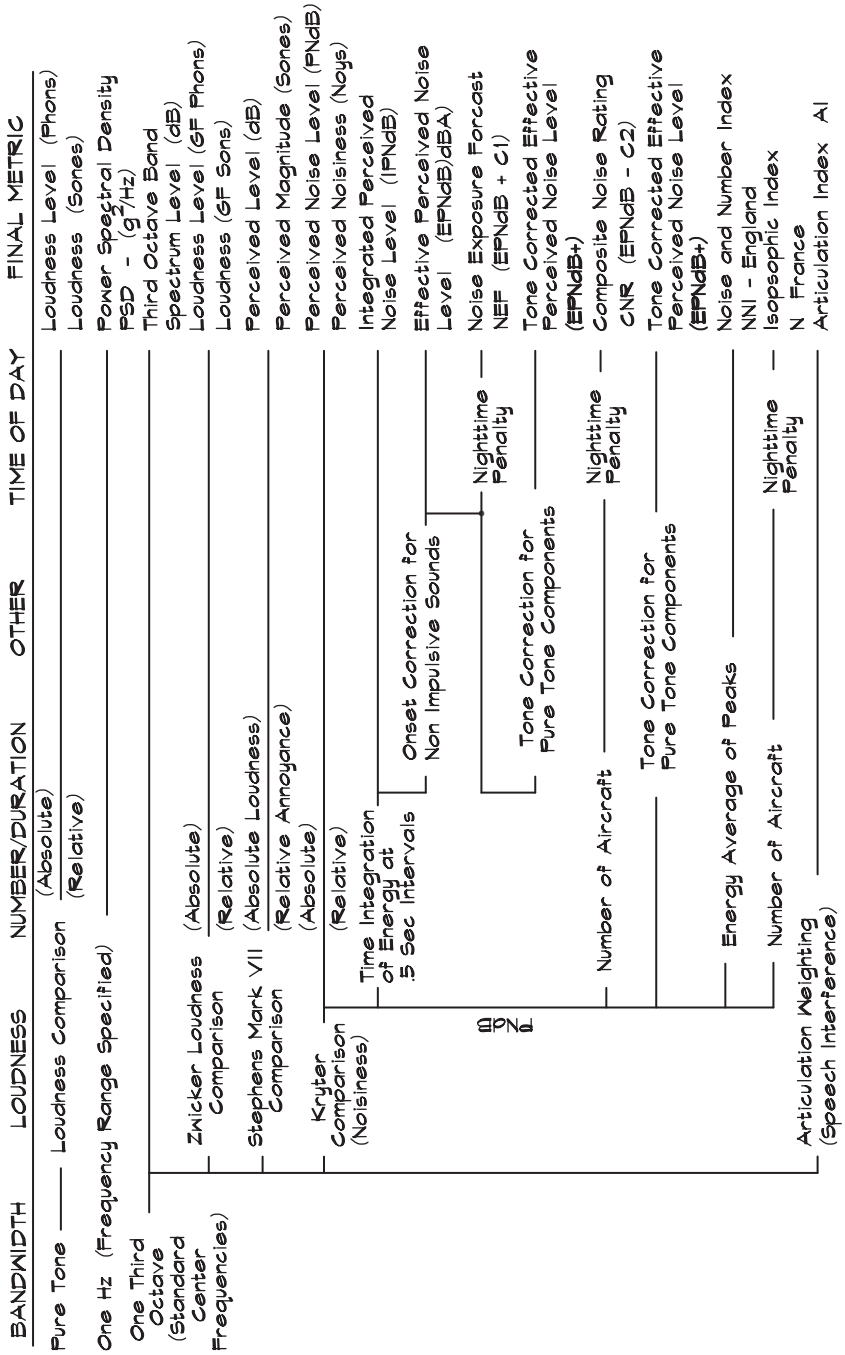
$$L_{dn} \cong \text{CNEL} \tag{4.15}$$

$$L_{dn} \cong \text{NEF} + 35(\pm 3) \tag{4.16}$$

$$L_{dn} \cong \text{CNR} - 35(\pm 3) \tag{4.17}$$

Similar relationships can be derived for the other metrics currently in use.

FIGURE 4.22 Narrow Band Noise Metrics



Narrow-Band Analysis

The analysis of sound in frequency bands of one-third octave and less is often useful for the detailed analysis of room acoustics and vibration. Instruments used for this type of measurement in real time are called spectrum analyzers or real-time analyzers (RTA). Two types of meters are most frequently encountered, those having a constant percentage bandwidth

TABLE 4.3 Parameters Used in Equation 4.14 (Schuller et al., 1995)

Metric	Constants A	B	C	Day Interval (Hours)	W_i	Morning, Evening Interval (Hours)	W_i	Night Interval (Hours)	W_i	L_i (dB)
Ke	20	15	105.8	08–18	1	06–08, 18–23	2–8	23–06	10	L_{ASmx}
L_n	10	10	44	06–23	0			23–06	1	L_{AE}
L_d	10	10	47.6[1]	07–23	1			23–07	0	L_{AE}
Q	13.3	13.3	65.7[2]	06–22	1			22–06	5[3]	L_{ASmx}
IP	10	10	49.4	07–22	1			22–07	10	L_{AE}
L_{dn}	10	10	44	06–23	0			23–06	1	L_{AE}
CNEL	10	10	44	07–19	1	19–22	3	22–07	10	L_{AE}
L_{24h}	10	10	44	00–24	1					L_{AE}
NNI	10	10	$80 - 5 \log \sum_{i=1}^{N} n_i w_i$	06–18	1			18–06	0	L_{pnmx}
NEF	10	10	88	07–22	1			22–07	16.7	L_{epn}

Single Event Noise Level Descriptors

L_{AE} = A-weighted sound exposure level

L_{ASmx} = Maximum S (slow) A-weighted sound level

L_{pnmx} = Maximum perceived noise level

L_{epn} = Effective perceived noise level

1) 7 hour night from 00.00 to 06.00 and 23.00 to 24.00 hours on a given day

2) 16 hour daytime period from 07.00 to 23.00 hours on a given day

3) Seperate calculations are specified for day and night. Values shown here are for calculations with emphasis on the contributions from nightime flight operations, Q_n. The weighting penalty includes a multiplication by the duration, in seconds, beween the first and last times that the instantaneous A-weighted sound level is within 10 dB of the maximum A-weighted sound level.

filter such as octave, third-octave, twelfth-octave, and so on, and those having a constant bandwidth such as 1 Hz. The latter type is used primarily in a laboratory while the former are the more common field instruments.

Instruments have filters of one of two types: analog and digital. An ideal bandpass filter is a device that passes all the electrical signals within its bandwidth and totally rejects all other signals. Analog filters are made of passive (resistors, inductors, and capacitors) or active (operational amplifiers) that approximate this ideal behavior. A meter having a group of such filters, operating in parallel, with each center frequency separated from the next by one-third octave, constitutes a real-time analyzer. These devices are robust and responsive. Some offer internal processing, such as energy averaging, and others feature only a freeze-and-save capability. Internal averaging is preferred since it is difficult to catch a varying signal at a point where all frequencies of interest are simultaneously at an average value.

A second type of system utilizes a mathematical filter, sometimes referred to as a digital filter. Digital filters can be constructed with the same characteristics as their analog counterparts. In these instruments the electrical signal is sampled periodically and the resulting string of numbers analyzed mathematically. One such procedure is the Fourier analysis (Joseph Fourier, 1768–1830), whereby a periodic signal is decomposed into its various harmonic components. Fourier's mathematical theorem states that any periodic waveform can be constructed from the sum of a specific sinusoidal wave called the fundamental, and a series of harmonics of the fundamental, multiplied by suitably selected constants. A graph of the amplitude versus frequency of these components is the spectrum of the original signal. A signal that has been digitally sampled can be sorted into its component frequencies using a mathematical process called the fast Fourier transform (FFT). Using similar techniques, filters can be constructed mathematically and applied to the digital number stream. The advantages of the digital filter are its flexibility, its low-frequency resolution, and its low cost. Disadvantages are its high-frequency limitations (eventually we cannot sample and calculate fast enough) and the features available on a given instrument.

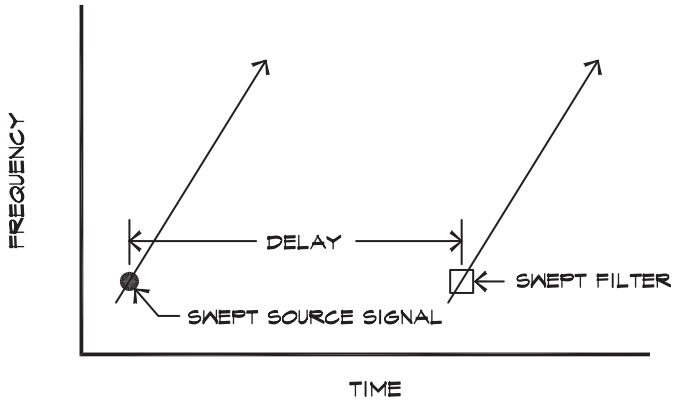
4.6 SPECIALIZED MEASUREMENT TECHNIQUES

Time-Delay Spectrometry

It is desirable to exclude noise intrusions from the measurements of a given signal. Anechoic chambers have been used for this purpose, since they are constructed not only to reduce sound from outside sources, but also to minimize the sound that is reflected off the walls and other surfaces of a room. A measurement technique originally developed by Richard Heyser (1931–1987) called time-delay spectrometry (TDS) can be used to make isolated measurements, even in a reverberant environment. The technique is based on the idea that when a sound source emits a signal it arrives at the receiving microphone after a given time. All the reflected sounds associated with the original signal arrive at some later time since they traveled along longer paths. If the measurement is made during a narrow time interval centered about the arrival time of the direct sound, later sounds are excluded and a nearly anechoic result can be achieved.

This is accomplished by converting the time delay into a frequency change. Figure 4.23 illustrates the principle. A loudspeaker is fed a sinusoidal signal that is chirped, or swept upward in frequency, at a fixed rate. At the receiver a narrow-band filter also is swept upward at the same rate. If the timing is correct the signal at a given frequency will arrive precisely when its filter window arrives. This technique is the same as that used by a quarterback

FIGURE 4.23 Time Delay Spectrometry



to throw a pass to a moving receiver. The ball (signal) and the receiver (filter window) must arrive at the same point at the same time for a reception. Passes that are delayed by reflections (off the ground or defensive linemen) do not arrive at the proper time and thus are not received.

Time-delay spectrometry can be used to measure the spectral response curve of a loudspeaker. The narrow-band analysis in Fig. 4.24 illustrates the detailed variations found in a typical loudspeaker. To obtain third-octave or octave-band data, the narrow-band energy data must be summed together over the appropriate frequency range. This process tends to smooth out the ripples in the curve and yields a more charitable portrait of the frequency response.

Energy-Time Curves

If a loudspeaker system is excited electronically with an impulsive signal, the signal received by a microphone can be plotted with time. This type of plot is called an energy-time curve (ETC) and contains useful information about the loudspeaker system as well as the room it is in. Turning first to loudspeakers, ETC plots are used to align transducers so that the signals from different components arrive at the listener at the same time. An example is shown in Fig. 4.25. Alignment is critical since a time delay is equivalent to a phase shift, which can produce a cancellation at the crossover frequency between transducers. Note that crossover points can be either electronic or spatial.

When two loudspeakers are misaligned, the ETC plot shows two distinct spikes. If this misalignment is sufficiently large, the result is a lack of clarity. When the two are aligned the overall level increases by 6 dB (due to an in-phase pressure doubling) and the peaks coincide in time. Loudspeaker alignment can be accomplished either by physical arrangement or by electronically delaying the signal transmitted to the forward transducer or both.

ETC plots can also reveal important information about reflections in rooms. A long-delayed reflection from the rear wall of a room, if sufficiently loud, can be disturbing to the perception of speech. Sometimes it is difficult in practice to identify the exact path that is causing the problem, particularly when multiple reflections are involved. An ETC plot can reveal the delay time of a given reflection and aid in the identification of the problem path. Patches of absorption can then be placed on the appropriate surfaces and the ETC measurements repeated for confirmation.

FIGURE 4.24 TDS Loudspeaker Measurements (Community, 1991)

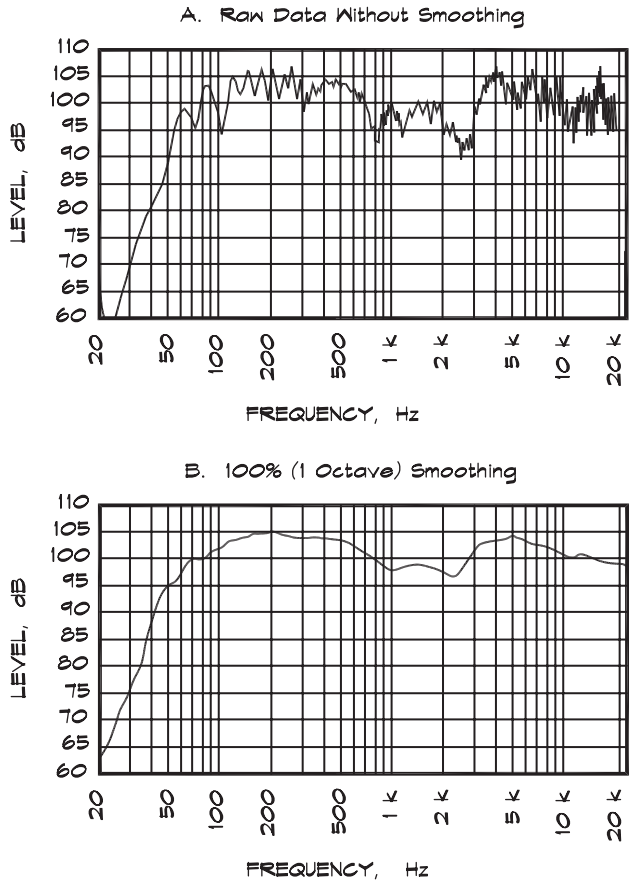
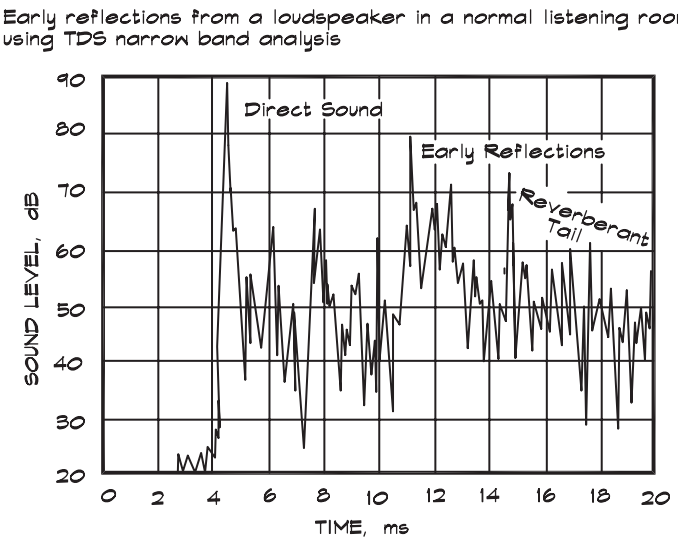


FIGURE 4.25 Energy Time Curves–ETC (Biering and Pedersen, 1983)



Sound Intensity Measurements

Direct measurement of the sound intensity has become possible through recent developments in commercial instrumentation. The intensity in a plane wave is defined as

$$I = p\,u \tag{4.18}$$

where I = maximum acoustic intensity (W/m^2)
p = acoustic pressure (Pa)
u = acoustic particle velocity (m/s)

The pressure is easily measured; however, direct measurement of the particle velocity is difficult. Instead the pressure can be measured using two closely spaced microphones, shown in Fig. 4.26, from which the change in pressure or pressure gradient can be obtained.

The reasoning is based on Newton's second law (F = m a) in one direction

$$\frac{d\,p}{d\,x} = -\rho_0 \frac{d\,u}{d\,t} \tag{4.19}$$

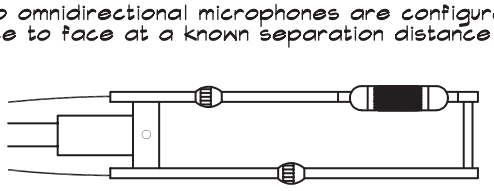
where ρ_0 = density of the bulk fluid (kg/m^3)
$d\,p$ = acoustic pressure change over a small distance $d\,x$ (Pa)
$d\,u$ = acoustic particle velocity change in time $d\,t$ (m/s)

The minus sign is there to indicate in which direction the slice accelerates. This equation is a well-known fluid dynamic relationship called Euler's equation. It relates the difference in pressure across a slice of fluid to an acceleration in the fluid slice that is proportional to its mass. The intensity is calculated by solving Eq. 4.19 for the particle velocity by integration

$$u = -\frac{1}{\rho_0 \Delta x} \int (p_a - p_b) \tag{4.20}$$

where ρ_0 = density of the bulk fluid (kg/m^3)
p = acoustic pressure measured at two points a and b which are Δx apart (Pa)
u = acoustic particle velocity (m/s)

FIGURE 4.26 Microphones Used in Intensity Measurements (Gade, 1982)



The intensity is then obtained by multiplying the pressure and the particle velocity.

$$I(\theta) = p\,u(\theta) = -\frac{p_a + p_b}{2\,\rho_0\,\Delta x}\int (p_a - p_b) \tag{4.21}$$

where $I(\theta)$ = acoustic intensity in a given direction (W/m^2)
p = acoustic pressure, which when measured at two points a and b that are Δx apart, is designated with a subscript (Pa)
$u(\theta)$ = acoustic particle velocity in a given direction (m/s)
ρ_0 = density of the bulk fluid (kg/m^3)

Since the intensity is a vector, its magnitude depends on the direction in which the microphones are oriented. Using this feature the intensity probe can be used for source location and strength.

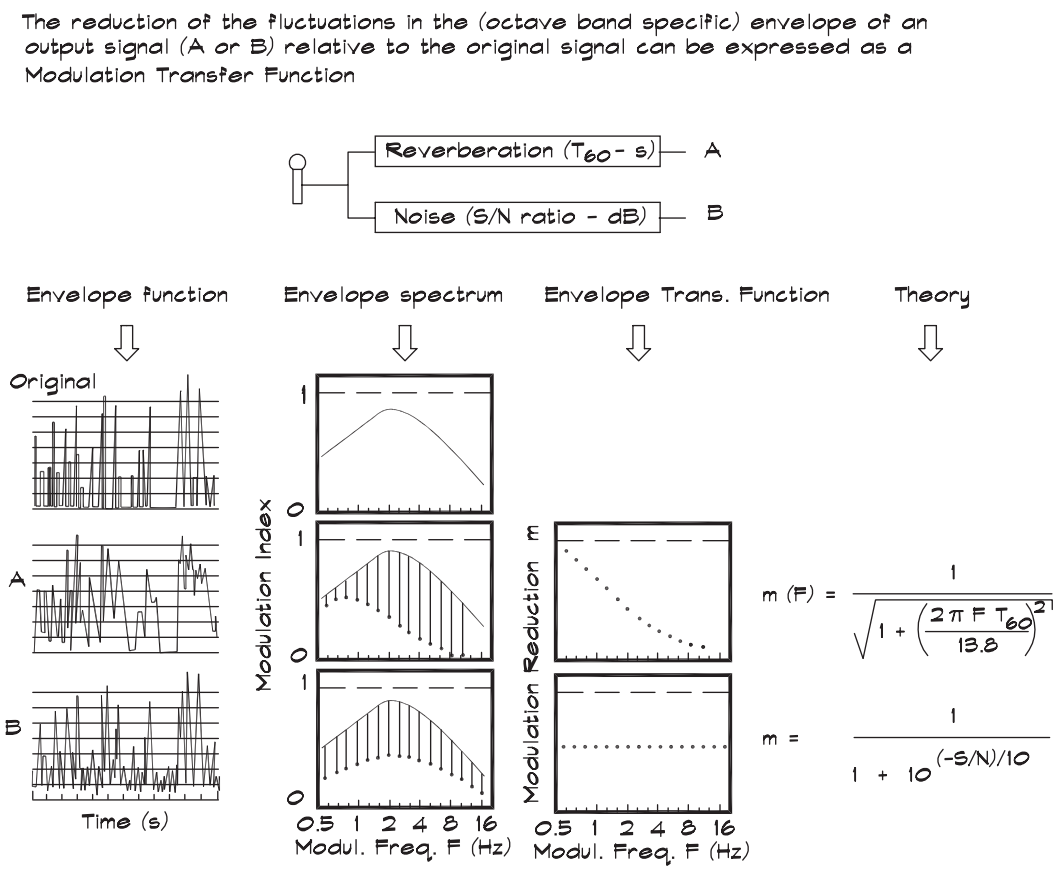
Modulation Transfer Function and RASTI

The intelligibility of speech has traditionally been measured by conducting tests, using various word lists, in rooms with human listeners. Although this methodology is the basis of most of our systems for predicting the intelligibility, it is highly desirable to have an electronic method of directly measuring these quantities. Human speech patterns are complex, and simple sinusoidal signals do not accurately mimic their behavior. Two Dutch scientists, Houtgast and Steeneken (1973), developed a measurement system, called the modulation transfer function (MTF), which replicates many of the properties of human speech. The concept is illustrated in Fig. 4.27. The idea behind MTF is that speech consists of modulated bands of noise. Our vocal cords vibrate to produce a band of noise, whereas our mouths modulate it at various frequencies to form words. To recreate this pattern, we start with an octave-wide band of noise and modulate it with a low-frequency tone. Mathematically this means that the carrier is multiplied by a sinusoidal function having a peak-to-peak amplitude of one. The result is a source signal that looks like the one on the left side of Fig. 4.28. For an accurate measurement the test signal level must be set to that of an average speaker and positioned where his mouth would be.

When this signal is transmitted to a listener, it is altered by the environment to some degree and can result in reduced speech intelligibility. The distortion mechanisms include background and reverberant noise, which raise the bottom of the signal above zero, and reflections, which add back a delayed and perhaps distorted copy of the signal. A typical receiver signal, shown on the right side of Fig. 4.28, is less modulated than the original, where the degree of modulation is defined by the depth of the modulation envelope. The reduction in modulation is characterized by a modulation reduction factor, $m(f_m)$, which is a function of the modulation frequency f_m. The modulation reduction factor varies from 0 for no reduction to 1 for 100% modulation reduction. Curves can be measured of the behavior of m versus f_m as shown on the bottom of Fig. 4.28.

When background noise is the principal source of the distortion, the effect on modulation reduction appears in terms of a signal-to-noise ratio, which is independent of modulation frequency. The noise raises all levels at the receiver within the carrier band and thus reduces modulation equally. When the distortion is produced by reverberation, the modulation reduction has the form of a low-pass filter with the faster fluctuations more sensitive to the effects of reverberation. This effect is characterized by the product of the modulation frequency and the

FIGURE 4.27 Basis of the Modulation Transfer Function (Houtgast and Steeneken, 1985)



room reverberation time. The overall modulation reduction factor is given mathematically as the product of these two effects for an unamplified signal

$$m(f_m) = \frac{1}{\sqrt{1+\left[2\pi\, f_m \frac{T_{60}}{13.8}\right]^2}} \frac{1}{1+10^{\left(-0.1L_{SN}\right)}} \tag{4.22}$$

where $m(f_m)$ = modulation reduction factor
L_{SN} = signal to noise level (dB)
f_m = modulation frequency (Hz)
T_{60} = room reverberation time (s)

The modulation frequency f_m ranges in value from 0.63 Hz to 12.5 Hz in third-octave intervals. The input-output analysis for a given system is done at 7 octave bands and 14 modulation frequencies, for a total of 98 separate values of m.

FIGURE 4.28 Modulation Transfer Function (Houtgast and Steeneken, 1985)

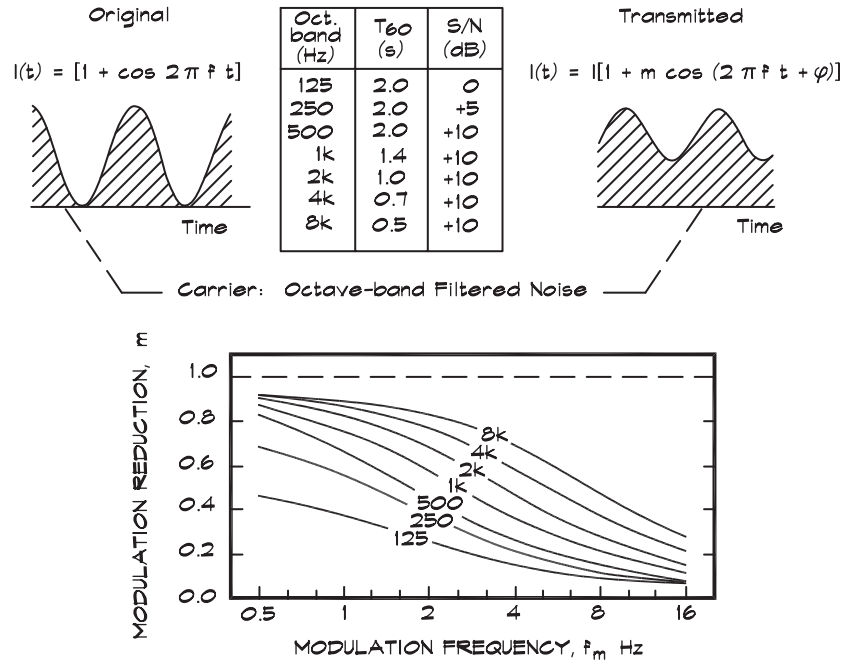


Illustration of how an MTF analysis is performed by using an octave-band filtered noise carrier, 100% intensity modulated, for each modulation frequency sucessively. This leads to a family of MTF curves. Each curve is calculated for the data given in the table.

Speech Transmission Index

With the MTF we have a quantity that mimics the behavior of speech, and can be physically measured with a properly constructed instrument. The missing link is the relationship between MTF and speech intelligibility. This is given in Fig. 4.29 by a speech transmission index (STI), which is similar to an articulation index or a percentage loss of consonants, in that it is a direct measure of speech intelligibility. All three are numerical schemes used to quantify the intelligibility of speech. Fig. 4.30 shows the relation between STI and Alcons, and Fig. 4.31 shows the similarity of STI to AI.

Steeneken and Houtgast (1980, 1985) developed an algorithm for transforming a set of m values into a speech transmission index (STI) by means of an apparent signal-to-noise ratio expressed as a level. This level is the signal-to-noise ratio that would have produced the modulation reduction factor, had all the distortion been caused by noise intrusion, irrespective of the actual cause of the distortion.

$$L_{SNapp} = 10 \log \frac{m}{1 - m} \tag{4.23}$$

where L_{SNapp} = apparent signal to noise ratio (dB)
m = modulation reduction factor

A weighted average of the 98 apparent signal-to-noise ratios yields the STI after applying a normalization such that

FIGURE 4.29 Typical Relations between the STI and Intelligibility Scores for the Various Types of Tests (Houtgast et al., 1985)

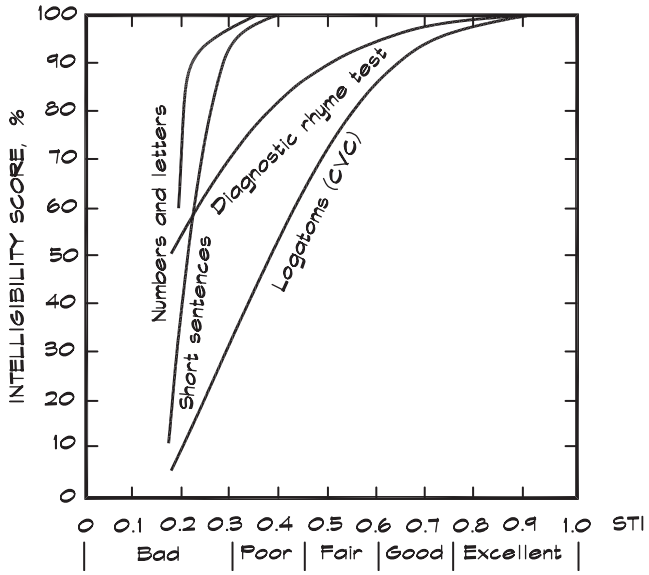


FIGURE 4.30 A Comparison of Articulation Index and Speech Transmission Index (Houtgast et al., 1980)

In this comparison the interference is only due to speech-shaped noise. The intelligibility rating is based on CVC (consonant, vowel, consonant) nonsense syllables embedded in a neutral carrier sentence.

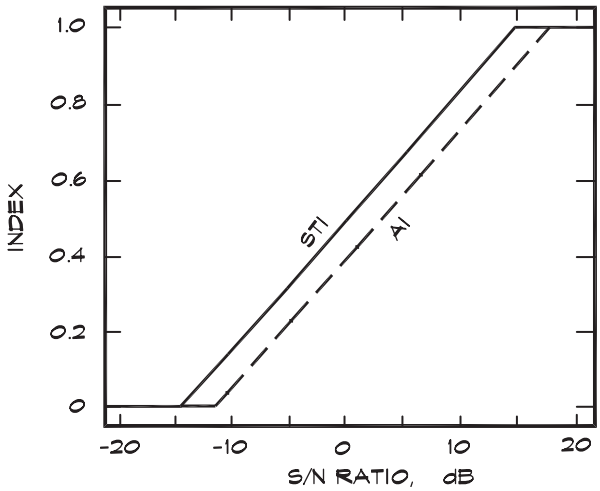
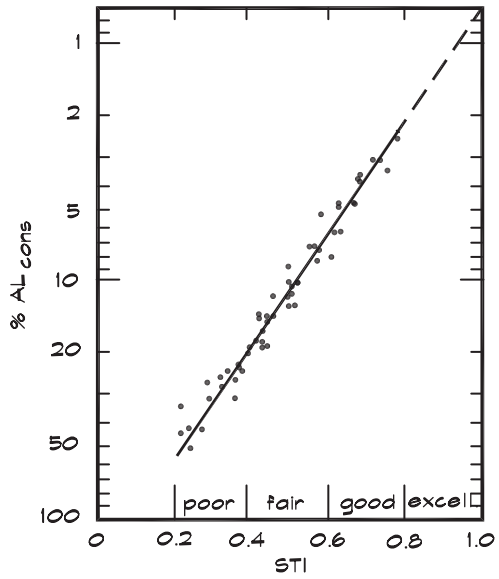


FIGURE 4.31 Relation between STI and Intelligibility Score (Houtgast et al., 1980)

The relation between intelligibility score and STI for a wide variety of conditions comprising many S/N ratios, reverberation times, and echo-delay times.



$\text{STI} = 1.0$ when $L_{\text{SNapp}} \geq 15$ dB for all 98 data points

$\text{STI} = 0.0$ when $L_{\text{SNapp}} \leq -15$ dB for all 98 data points

and

$$\overline{L_{\text{SNapp}}} = \sum_{i=1}^{7} w_i \left(L_{\text{SNapp}}\right)_i \tag{4.24}$$

where $\overline{L_{\text{SNapp}}}$ = average apparent signal-to-noise ratio (dB)

w_i = weighting for octave bands from 125 Hz to 8 k Hz

= 0.13, 0.14, 0.11, 0.12, 0.19, 0.17, and 0.14

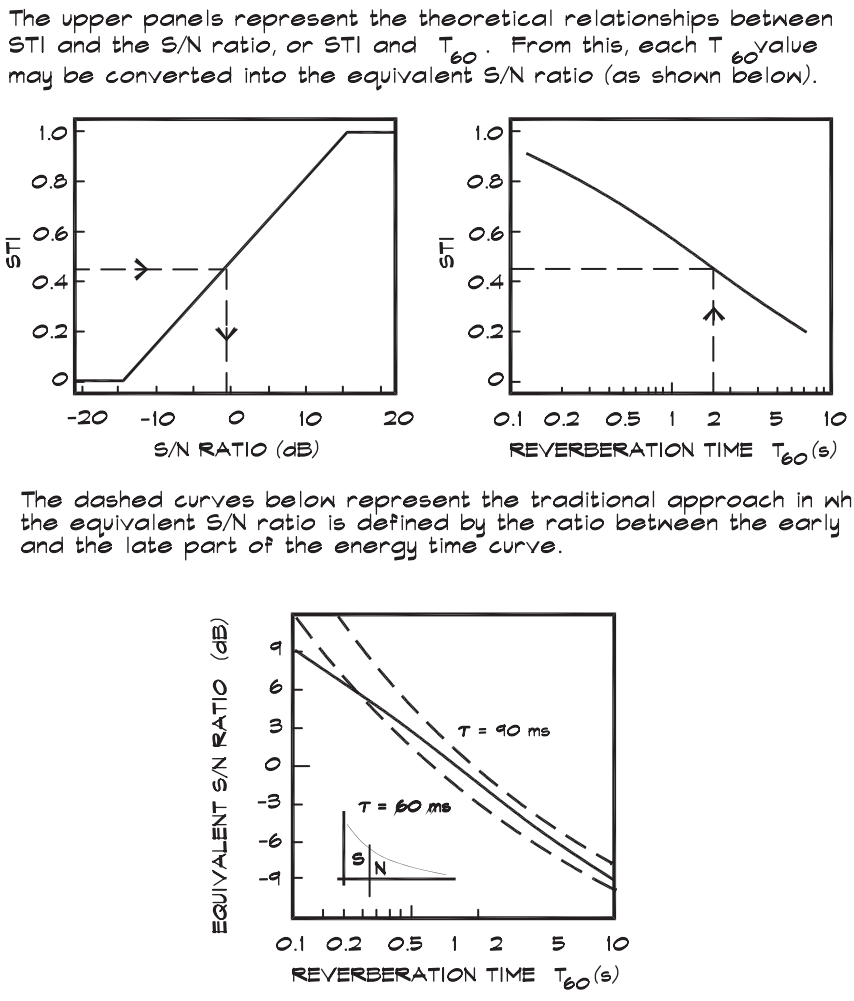
then

$$\text{STI} = \left[\overline{L_{\text{SNapp}}} + 15\right] / 30 \tag{4.25}$$

Figure 4.32 shows the relationship between STI and the signal-to-noise ratio as well as the reverberation time.

The bottom part of Fig. 4.32 represents an alternative way of interpreting the effect of reverberation. The early part of the reverberant tail is considered helpful to the understanding of speech, whereas the end is considered detrimental. The boundary between the two regions occurs in the neighborhood of 70 to 80 ms. Though it may seem that this is rather long, in that a single 65 ms delay can be detected as an echo, it should be remembered that in normal rooms the listener hears a series of reflections and thus the Haas region is extended somewhat (Fig. 3.29).

FIGURE 4.32 Relationship between Modulation Transfer Function and the Speech Transmission Index (Houtgast and Steeneken, 1985)

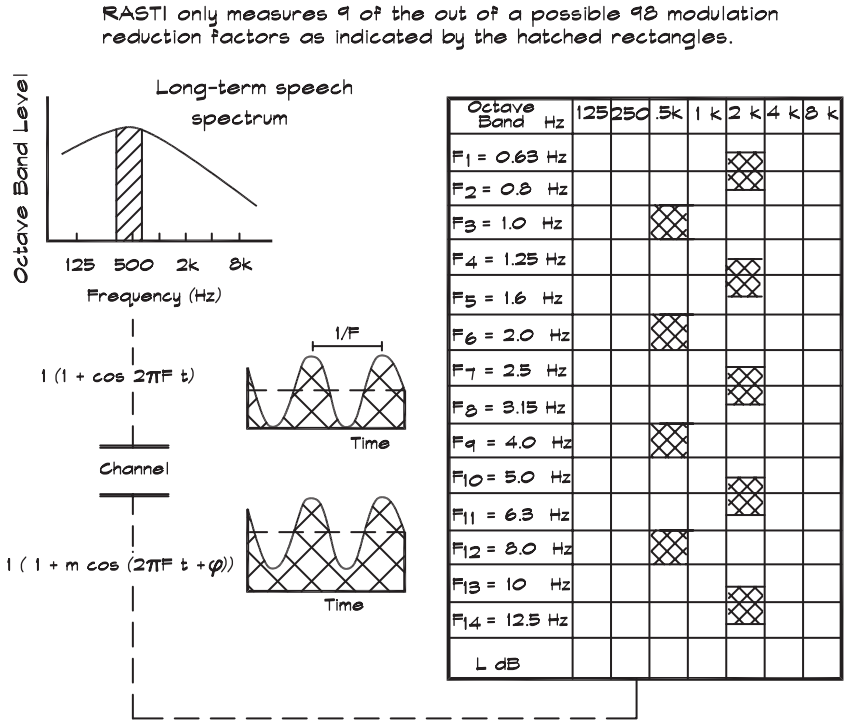


The research done by Houtgast and Steeneken established a way of measuring speech and intelligibility using an electronically generated test signal rather than a group of human subjects. Their calculation method is useful in evaluating rooms for an omnidirectional source, but does not include consideration of loudspeaker directivity, so necessary to the design of reinforcement systems. Once the method has been established as equivalent to other measures of intelligibility without amplification, the measurement system can be used to evaluate installed sound systems.

RASTI

RASTI or RApid STI is an approximation of the full STI taken by doing a measurement of nine of the 98 m values marked on the graph shown in Fig. 4.33. Two octave bands 500 and 2000 Hz are sampled. At 500 Hz, values of m are measured at four modulation frequencies, 1, 2, 4, and 8 Hz. At 2000 Hz, five modulation frequencies are measured, 0.7, 1.4, 2.8, 5.6, and 11.2 Hz. An apparent signal-to-noise ratio is calculated from the measured m values in each band and truncated so as to fall within the range of ± 15 dB. The L_{SNapp} values are

FIGURE 4.33 The RASTI Analysis System (Houtgast and Steeneken, 1985)



averaged and a RASTI value is calculated

$$\text{RASTI} = \left[\overline{L_{\text{SNapp}}} + 15\right] / 30 \tag{4.26}$$

where RASTI = rapid STI measurement index
$\overline{L_{\text{SNapp}}}$ = average apparent signal to noise ratio (dB)

In practice RASTI measurements can be made to evaluate the intelligibility of speech both for an unamplified talker as well as for an amplified sound system. The RASTI source is positioned at the talker location. If there is a microphone, the source is set in front of it so that the public address system can be tested. The receiver microphone is located at various points throughout an auditorium to determine the RASTI rating.

This page intentionally left blank

5 ENVIRONMENTAL NOISE

5.1 NOISE CHARACTERIZATION

Outdoor noise transmission from point to point is discussed in terms of a *source-path-receiver model*, where the source is described by its sound power level and perhaps a directivity, the path is characterized by various attenuation mechanisms such as distance or barriers, and the receiver is a location where a level is to be calculated or a criterion is to be met. When the measurement point is close to the source, attenuation mechanisms, other than distance and directivity, have little effect on the received level. As the source-receiver distance increases, more and more mechanisms come into play until, at large distances, environmental considerations such as air losses, ground attenuation, wind direction, and velocity can be of primary importance.

Fixed Sources

Stationary sources such as pumps, compressors, fans, and emergency generators, which are a fundamental part of buildings, can radiate noise into adjoining properties. In previous chapters various metrics for characterizing the noise from these sources have been discussed. The equivalent sound level L_{eq} will be used here as the primary descriptive metric. It has the advantage of being mathematically efficient for both fixed and moving sources, and correlates well with human reaction. The L_{eq} for a fixed source can be calculated from the steady level emitted over a given time period.

$$L_{eq} = L_s + 10 \log (t / T) \tag{5.1}$$

where L_{eq} = equivalent sound level (dB or dBA)
L_s = steady sound level (dB or dBA)
t = time the source is on (sec)
T = total time $T \geq t$ (sec)

When the on-time is equal to the total time, the L_{eq} is equal to the steady level. Multiple sources may be combined using their individual L_{eq} levels and Eq. 2.62, even if the on-times are not coincident. Twenty-four hour metrics such as L_{dn} can be calculated for fixed sources

in a similar way from a knowledge of the time history. For example, for a constant 24-hour source, the L_{dn} is 6.4 dB higher than the steady level.

Moving Sources

Moving sources such as automobiles, trucks, railroads, and aircraft often dominate fixed sources in the urban environment, but are more transient and difficult to control. When a sound source moves, the measurement distance changes in time as shown in Fig. 5.1.

As the distance increases the sound pressure level decreases by 10 log of the square of the overall source-to-receiver distance. Assuming a source moves at a constant speed, v, along a straight line path a distance, d, away from an observer, the intensity can be written as a function of time

$$I(t) = \frac{I_r \, d_r^2}{d^2 + v^2 \, t^2} \tag{5.2}$$

where $I(t)$ = sound intensity as a function of time (W/m^2)
I_r = measured sound intensity at distance d_r (W/m^2)
d = distance of closest approach (m)
v = source speed (m/s)
t = time (s)

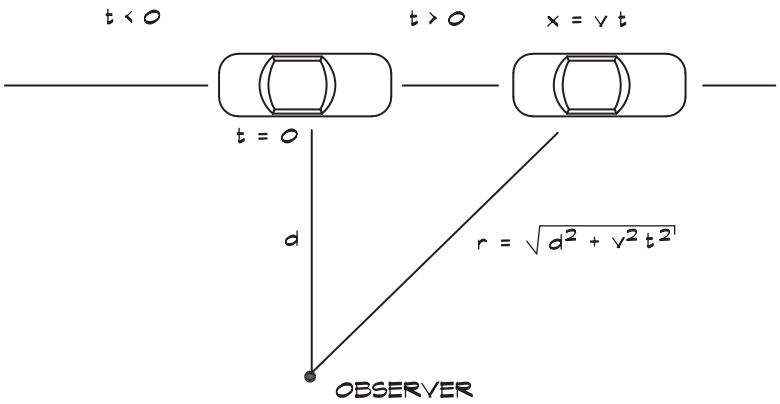
Equation 5.2 can be converted to levels by dividing by the reference intensity, I_o, and taking 10 log of each side.

$$L(t) = L_r + 10 \log \left[\frac{d_r^2}{d^2 + v^2 \, t^2} \right] \tag{5.3}$$

where $L(t)$ = sound level as a function of time (dB)
L_r = sound level at distance d_r (dB)

At $t = 0$, which is the point of closest approach, the sound level is dependent only on the ratio of the square of the reference distance to the minimum distance.

FIGURE 5.1 Geometry of a Moving Source



The L_{eq} for a single vehicle moving along a roadway can be calculated by adding (integrating) the contributions from each point along the source path for the time period under consideration.

$$L_{eq} = L_r + 10 \log \frac{1}{T} \int_{t_1}^{t_2} \frac{d_r^2}{d^2 + v^2 t^2} \, dt \tag{5.4}$$

The integral is done by letting the limits of integration go to plus and minus infinity, because most of the energy is contributed when the vehicle is close to the receiver.

$$L_{eq} = L_r + 10 \log \frac{\pi d_r}{v T} + 10 \log \frac{d_r}{d} \tag{5.5}$$

This is the equivalent sound level for a single vehicle traveling along a long roadway past a stationary observer in a time T. It is interesting to note that even for a single vehicle the falloff behavior with distance is that of a line source. If N similar vehicles pass a point during time T, a factor of 10 log N is added to Eq. 5.5 to account for them. The vehicle spacing does not matter since the levels combine on an energy basis.

When the noise level emitted by a group of sources varies in a normally distributed (Gaussian) way about a mean value, the L_{eq} level is calculated for the group (Reagan and Barry, 1978) by adding an adjustment that accounts for the variation, based on the standard deviation. The equivalent sound level for a long line of N vehicles passing a point in time T is

$$L_{eq} = \bar{L}_r + 0.115 \sigma^2 + 10 \log \frac{N \pi d_r}{v T} + 10 \log \frac{d_r}{d} \tag{5.6}$$

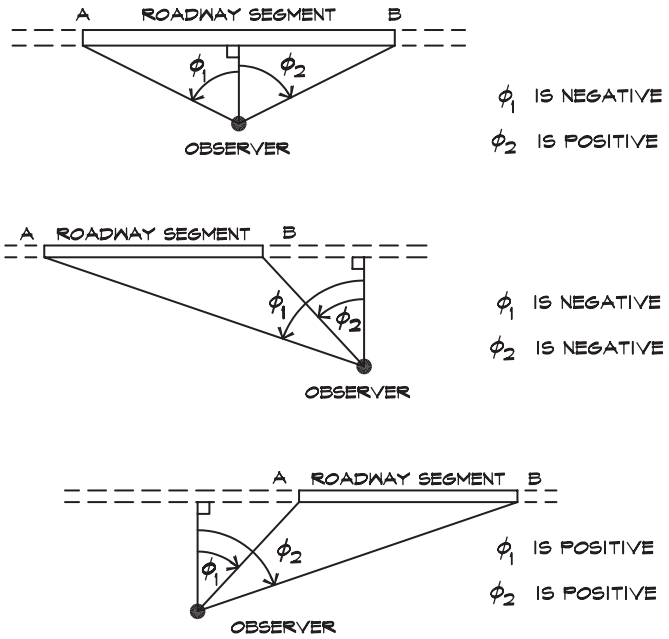
where L_{eq} = equivalent sound level (dB or dBA)
$\bar{L}_r$ = average reference sound level at distance d_r (dB or dBA)
d = distance of closest approach (m or ft)
v = source speed (m/s or ft/s)
T = time (s)
N = number of vehicles passing the measurement point in time T
σ = standard deviation of the reference sound level (dB or dBA)

Equation 5.6 is the fundamental relationship for modeling vehicle noise from a long unshielded roadway. When trucks or other classes of vehicles are present, their contributions are calculated separately and the levels combined. Similar calculations can be done to predict noise from other moving sources such as railroads, crawler tractors, earth movers, and construction vehicles simply by using the appropriate reference sound levels.

Partial Line Sources

When a sound source traverses a line segment, or when only part of a straight roadway is to be modeled, the configuration is called a *partial line source*. An example might be a crawler tractor moving back and forth along a fixed path or a segment of roadway such as that shown in Fig. 5.2. The geometry is described in terms of an angular segment, $\Delta\phi = \phi_2 - \phi_1$

FIGURE 5.2 Angular Designation of Partial Line Sources (Barry and Reagan, 1978)



(in radians), which modifies the integration in Eq. 5.4 by changing the time limits to angular limits, and yields the equation for the level generated by a partial line source

$$L_{eq} = \bar{L}_r + 0.115\,\sigma^2 + 10\,\log\frac{N\,\pi\,d_r}{v\,T} + 10\,\log\frac{d_r}{d} + 10\,\log\frac{\Delta\phi}{\pi} \tag{5.7}$$

If the source is a single vehicle emitting a steady noise level, then N = 1 and the standard deviation is set to zero.

Equation 5.7 models the behavior of the noise level from a given segment of roadway. The level varies as the included angle $\Delta\phi$ of the line segment, regardless of whether the segment is near or far away. Note that the distance d is measured perpendicular to the line of travel, from the point on the line closest to the observer, which does not change with the angle under consideration. The formula states that, for a given receiver position, equal noise levels are generated by equal angular segments. This holds as long as the angular segment is not so far away that atmospheric and ground effects come into play and decrease the levels from the more distant sources.

If the measurement point is moved farther away from the line element the change in level can be obtained from the new distance and angle

$$\Delta L_{eq} = 10\,\log\frac{d_1}{d_2} + 10\,\log\frac{\Delta\phi_2}{\Delta\phi_1} \tag{5.8}$$

With increasing distance the line element behaves more and more like a point source. Let us take the example of a partial line source, where $\phi_1 = 0$, and use the trigonometric relationship

$\tan\phi = \frac{d_s}{d}$, where d_s is the length of the line segment. We then use the approximation for small angles, $\tan\phi \cong \phi$, where ϕ is in radians. The segment length stays the same so that when the measurement distance is large ($d >> d_s$), the change in level with distance is given by

$$\Delta L_{eq} \cong 10 \log \frac{d_1}{d_2} + 10 \log \frac{d_1}{d_2} = 10 \log \left[\frac{d_1}{d_2}\right]^2 \tag{5.9}$$

Thus we regain the expected point source falloff for the changes in level from distant line source elements.

5.2 BARRIERS

Point Source Barriers

Barriers are the most commonly used way of controlling exterior noise. Figure 5.3 shows a simple barrier geometry. When a plane wave encounters a barrier, the lower portion of it is cut off leaving the rest to propagate over the wall. The high and low-pressure regions of the wave impinge on the quiescent fluid in the shadow zone and propagate into it. In this manner the wave diffracts or is bent into the space behind the barrier. The greater the diffraction angle the greater the attenuation.

Barrier attenuation for a point source is calculated (Maekawa, 1965) using the maximum Fresnel number, which is determined from the difference between the shortest propagation path that touches the edge of the barrier and the direct path through the barrier. The geometry is given in Fig. 5.4. The maximum Fresnel number N is

$$N = \pm \frac{2}{\lambda} (A + B - r) \tag{5.10}$$

FIGURE 5.3 Geometry of a Simple Barrier

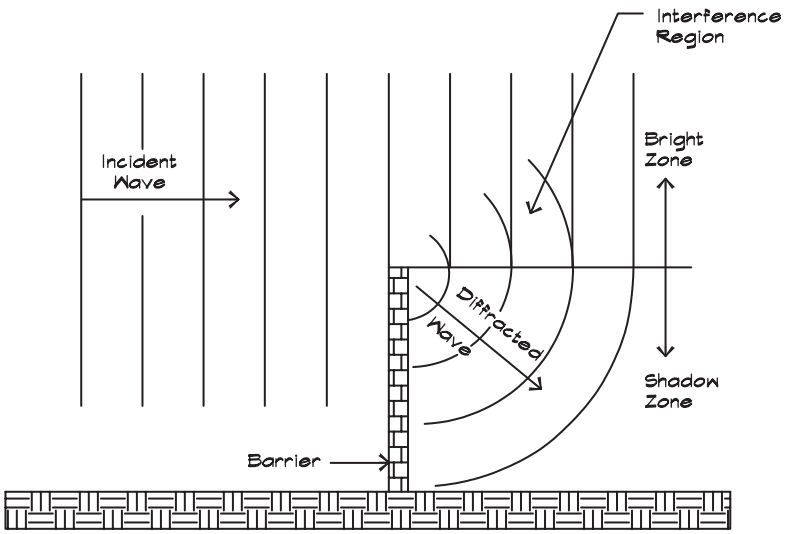
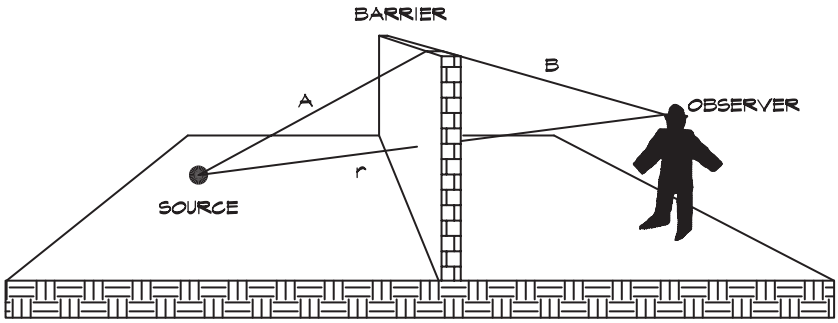


FIGURE 5.4 Path Length Difference for a Simple Barrier



where (A + B − r) is the minimum path length difference. The sign is positive in the shadow zone and negative in the bright zone. For a simple point source the barrier attenuation is

$$\Delta L_b = 20 \log \frac{\sqrt{2\pi N}}{\tanh \sqrt{2\pi N}} + K_b \tag{5.11}$$

where ΔL_b = barrier attenuation for a point source (dB)
A, B, r = minimum source to receiver distances over and through the barrier (m or ft)
N = maximum Fresnel number defined by Equation 5.10 ($-0.19 \leq N \leq 5$)
λ = wavelength of the frequency of interest (m or ft), (usually taken to be 0.63 m or 2 ft as an average value for roadways - equivalent to 550 Hz)
K_b = barrier constant which is 5 dB for a wall and 8 dB for a berm

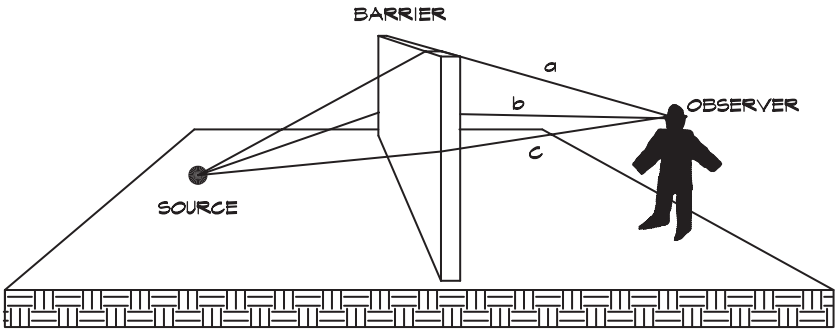
When N is zero, that is when the line of sight between the source and the receiver is just broken by the top of the barrier, the theoretical attenuation afforded by a wall is 5 dB. For every 0.3 m (1 ft) of barrier above this line the barrier provides about one additional dB of attenuation at 500 Hz. This is a rough rule of thumb, which is useful for estimation purposes. Detailed attenuation calculations should be done for the actual source spectrum and barrier geometry. If the barrier has an unusual shape, such as a truncated triangle in section, the total path length across the top of the barrier must be calculated. For large values of N, the attenuation has a practical limit of 20 dB for walls and 23 dB for berms.

If a receiver, located in the bright zone where the attenuation is zero, is lowered toward the shadow zone the attenuation does not jump instantaneously from zero to 5 dB. Instead, theory predicts a transition zone where sound waves are scattered from the top of the barrier and combine out of phase with the direct path waves, resulting in some attenuation. In the transition zone N is negative (N ranges between −0.19 and 0 for walls and between −0.25 and 0 for berms) and the radical in Eq. 5.11 yields an imaginary number. In this region the hyperbolic tangent becomes a simple tangent function since $\tan(\theta) = \tanh(j\theta)$. In practice, the transition zone is narrow and little attenuation should be expected when the line of sight falls above the top of the barrier.

Practical Barrier Constraints

There are practical limitations to barrier theory. If barriers are not long, the sound can travel around them. Barrier attenuations can be calculated for each of these paths using Eq. 5.11,

FIGURE 5.5 Possible Sound Paths around a Finite Barrier



and the resulting levels combined at the receiver location. Reflections from nearby buildings can produce flanking paths where the sound travels around a barrier. Examples can be found in Fig. 5.5.

A high building located behind a barrier can scatter sound back into the shadow zone, particularly where there is an overhanging roof, thereby reducing the barrier's effectiveness. Measured data for these types of conditions are shown in Fig. 5.6. A local reverberant field set up between a barrier and a building can decrease barrier effectiveness, especially at low frequencies. Low-frequency reverberation can also generate an increase in noise level compared with the no-barrier condition.

When a wind is blowing from the source to the receiver and the distance is sufficiently long, the barrier effectiveness is much reduced due to the downward bending of the sound waves. This effect occurs when the barrier is relatively far (say A and B $>$ 100 m) from the source and receiver.

The construction of sound walls on top of berms presents a curious dilemma. Due to the interaction of sound with the top of the berm, an additional attenuation of about 3 dB is achieved over that which would be obtained from a wall. If a 0.3 m (1 ft) wall is built on top of a berm the attenuation would increase about one dB for the extra height and decrease about 3 dB due to its being a wall, yielding a net increase in sound level. Thus walls on top of berms must be sufficiently high to offset the loss of the berm effect. In practice, although berms are more efficient attenuators, they are difficult to build very high. Most berms must be constructed with a 2:1 slope, so they end up being four times wider than they are high, and space constraints limit their use.

Ground attenuation, which occurs when sound waves graze or skim across the ground, is reduced when sound diffracts over a high wall since the barrier changes the angle of approach. Barrier attenuation may be partially offset by the loss of ground attenuation. By doing a few sample barrier calculations one can quickly discover that barriers of a given height are most effective when they are located close to the source or receiver and least effective when they are positioned half-way in between.

Line Source Barriers

When a barrier is constructed along a line source, such as a highway, the geometry and thus the Fresnel number changes for each angle of roadway covered by the barrier. Figure 5.7 illustrates this condition and Fig. 5.2 gives the sign convention for the angle segments.

FIGURE 5.6 Sound Attenuation by a Barrier in Front of a Reflecting Surface (Sharp, 1973)

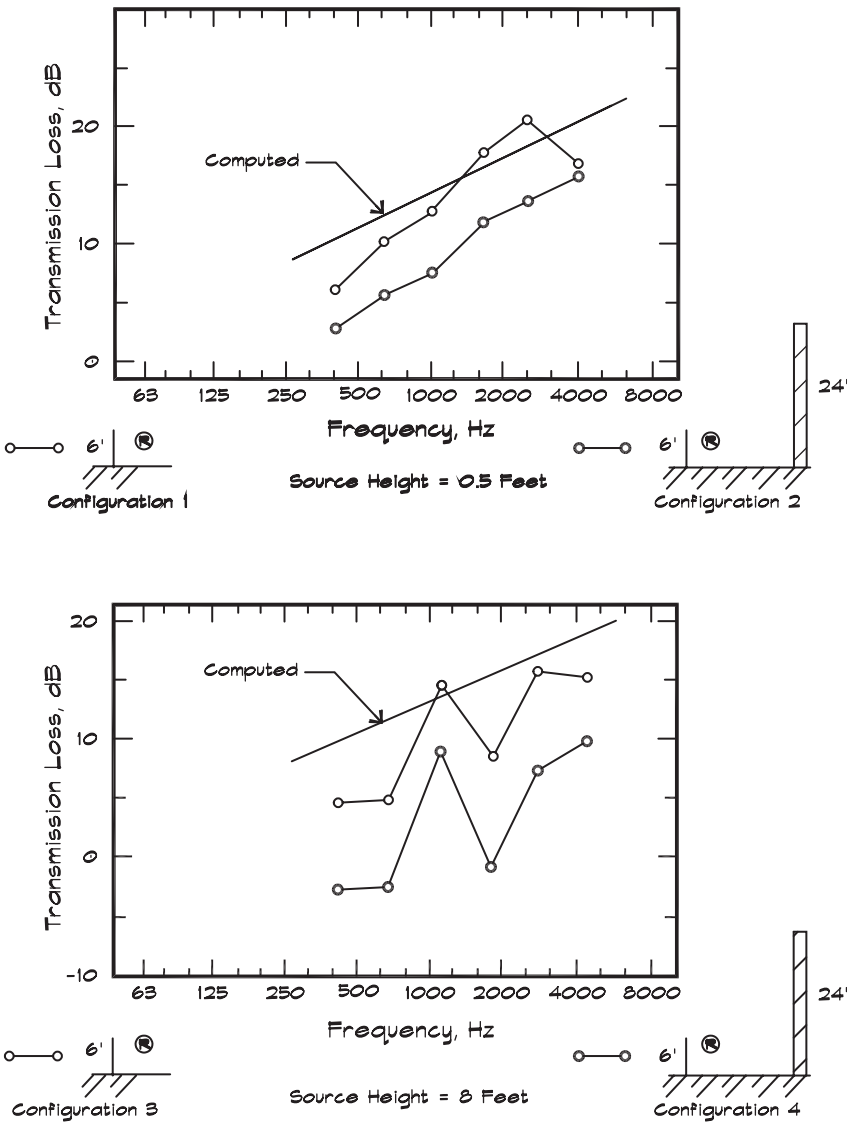
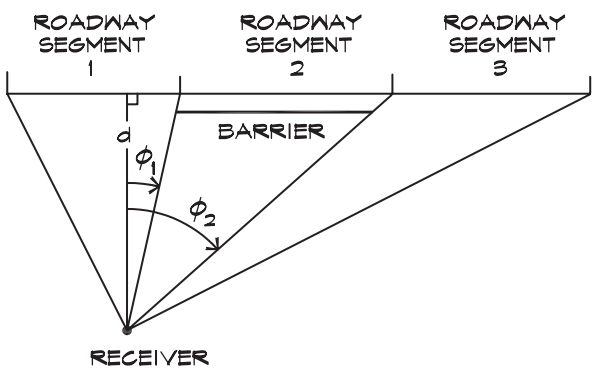


FIGURE 5.7 Finite Roadway/Finite Barrier Geometry



To calculate the barrier shielding for a segment of roadway the attenuation can be integrated over all angles of ϕ that are covered. The formula for the barrier shielding of a partial line source is (Barry and Reagan, 1978)

$$\Delta L_{\phi} = -10 \log \frac{1}{\Delta\phi} \int_{\phi_1}^{\phi_2} 10^{-0.1\ \Delta L_b(\phi)}\, d\phi \tag{5.12}$$

where ΔL_{ϕ} = barrier attenuation for a line source element (dB)
$\Delta L_b(\phi)$ = barrier attenuation for a point source located at ϕ (dB)
$\Delta\phi = |\phi_2 - \phi_1|$ = angle of the barrier element (radians)
ϕ_1 = angle from the perpendicular to the left edge of the line element (radians)
ϕ_2 = angle from the perpendicular to the right edge of the line element (radians)

The integral is cumbersome and is done numerically using an approximation for the angular dependence of the Fresnel number, namely

$$N(\phi) \cong N_0 \cos \phi \tag{5.13}$$

where $N(\phi)$ = Fresnel number for a small line segment located at ϕ
N_0 = Fresnel number determined along the perpendicular path between the receiver and the line source
ϕ = angle from the perpendicular to the segment (radians)

Barry and Reagan have provided extensive tables showing the results of the integration for barrier segments at various angles. Computer programs for doing the barrier calculations are commercially available and are straightforward to write. When a roadway is divided into angular segments of less than about 25°, the point source barrier attenuation can be applied using the center of each segment and the results combined. For an infinite roadway and an infinite barrier, the integration has been done by Kurze and Anderson (1971) and is shown in Fig. 5.8. Experimental results also are shown in the figure. Note that the barrier attenuation for an infinite line source is about 5 dB less than for a point source at the same perpendicular distance.

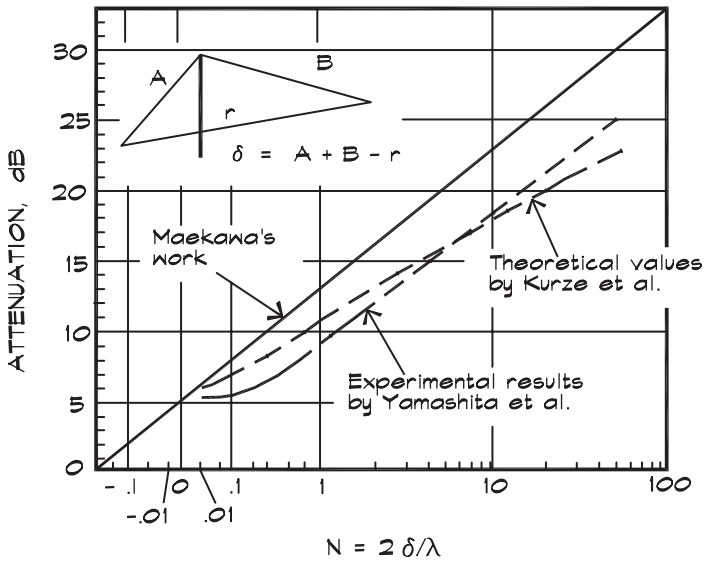
Barrier Materials

Many materials are available to the designer; however, there are a few important considerations. First, barriers must be nonporous—that is, they must block the passage of air through them. Second, they must have sufficient mass so that the sound traveling through the barrier is significantly less than the sound diffracting over or around the barrier. This consideration leads to the requirement that barriers be built of a material having a total surface mass density of at least 20 kg/sq m (4 lbs/sq ft). Third, they must be weather resistant and properly designed to withstand wind and other structural loads appropriate for the location.

The mass requirement can be fulfilled using a support structure with one layer of 16 mm (5/8") and one layer of 19 mm (3/4") plywood sandwiched. When the panels are applied to both sides of a stud, 90 mm (3 1/2") wide, they may be less massive, typically 13 mm (1/2") to 16 mm (5/8") plywood. Virtually any thickness of concrete or concrete masonry unit that is self-supporting will meet the mass requirement. A stucco wall is very effective. Stucco is 22 mm (7/8") thick and weighs about 42 kg/sq m (9 lbs/sq ft) at that

FIGURE 5.8 Line Source Barrier Attenuation (Maekawa, 1977)

Sound attenuation by a thin screen in free space versus the Fresnel zone number N for a point source (solid line) and for an incoherent line source (dotted and dashed curves). Attenuation is relative to propagation in free space.



thickness. Precast concrete panels, treated to look like wood or brick and supported by I beam columns, are commercially available. The panels are held in place at their ends by the flange of the I beam.

Corrugated sheet metal panels sometimes are used to construct noise barriers. Commercial barriers with both solid and sound-absorbing perforated skins are available, usually in 18 Ga. steel supported by steel columns. Absorbing materials such as fiberglass can be incorporated behind the perforated panels to reduce barrier reflections. The fiberglass is encased in a plastic bag to protect it from the weather. For very thin layers of plastic there is little reduction in the absorptive properties of the fill.

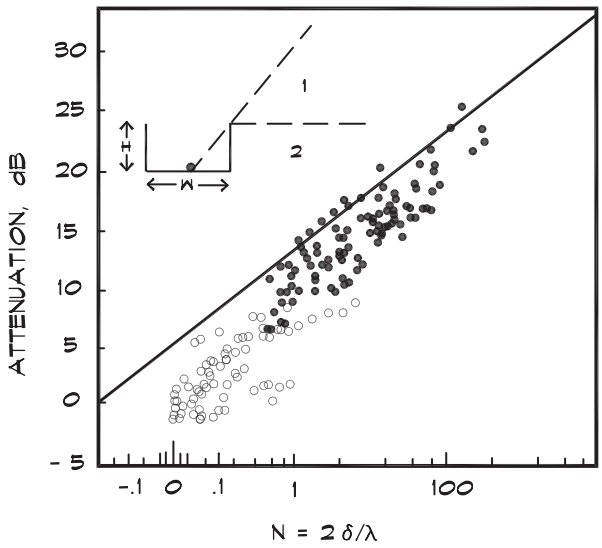
Noise barriers should be constructed so that there are no openings between the barrier and the ground. Openings allow the sound to pass under the barrier and can reduce its effectiveness. Trees, shrubs, and other foliage are not effective. They are porous and do not meet the mass requirement. Rows of trees, heavy grass, and dense foliage can provide some excess ground attenuation, on the order of 0.1 dB/ m thick (3 dB/100 ft). They are also useful in giving a psychological sense of privacy, or in landscaping a sound barrier to make it more aesthetically acceptable.

Roadway Barriers

Sound barriers constructed along a roadway to protect residential areas have become a common sight in urban areas. Barriers along freeways incorporate a safety shape at their base to protect vehicles from direct impacts. When barriers are located on both sides of a roadway, the multiple reflection of sounds back and forth between them reduces the attenuation that would be expected from calculations using standard formulas. Maekawa (1977), testing scale models, measured the attenuation of parallel walls. The results are shown in Fig. 5.9.

FIGURE 5.9 Measured Sound Attenuation from Reflective Screens (Maekawa, 1977)

Sound attenuation measured by the pulse method with a 1:40 scale model. The road width W = 20.2 m and the screen height H = 15 m full scale. The plotted data ○ and • are for the values obtained at receiving points in zones 1 and 2 respectively. The maximum Fresnel number N is calculated for the near side screen neglecting the one on the far side, for the center frequency of each one-third octave band. Attenuation is relative to propagation in free space.



When absorptive materials are applied to the roadway side of these walls, the attenuation values return to their expected levels (Fig. 5.10).

A row of buildings, one story tall, can provide some shielding over and above distance attenuation. The amount depends on the percent of coverage of the line source. For 40% to 65% coverage, about 3 dB is achieved; for 65% to 90% shielding we get about 5 dB. Additional rows of buildings can add about 1.5 dB of extra shielding per row, up to a maximum of 10 dB.

Barrier shielding due to buildings sometimes produces a curious phenomenon. When a receiver is standing behind a row of buildings, not infrequently the sound will be perceived as coming from the side away from the source. This is because the sound is reflected from another building behind the receiver. The sound coming by way of this path is louder than the direct path over the barrier due to the reduced barrier effectiveness at the elevation of the reflection point as in Fig. 5.11.

Figure 5.12 summarizes barrier and ground effects. Where sound propagates over a soft ground cover such as grass, it is common practice to calculate the attenuation by combining the distance and ground effects into a 4.5 dB per distance doubling falloff rate. This is not the most accurate method, but without knowledge of the impedance of the surface it provides a useful estimate.

5.3 ENVIRONMENTAL EFFECTS

Several attenuating mechanisms, over and above those associated with geometrical spreading and barrier losses, influence the propagation of a sound wave. These are grouped in categories

FIGURE 5.10 Measured Sound Attenuation from Absorptive Screen (Maekawa, 1977)

The figure is the same as Fig. 5.9 except that the inner surfaces of the screens have been lined with sound-absorbing materials. The normal incidence absorption coefficient α = 0.48 - 0.93 was used for the measurement frequency range.

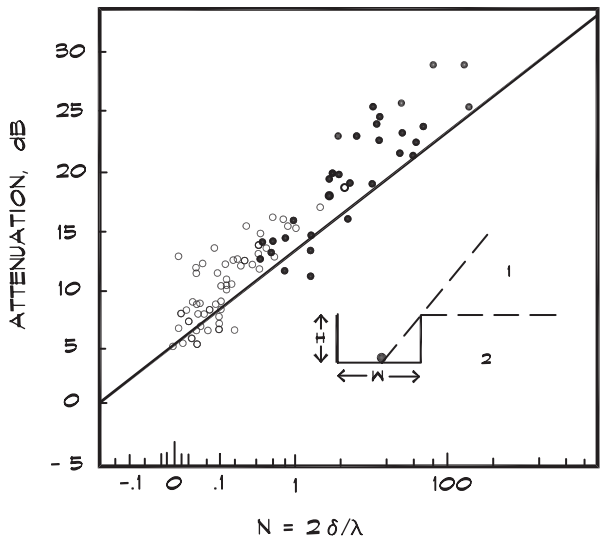
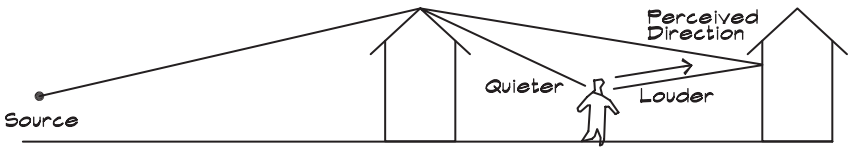


FIGURE 5.11 Shift in the Perceived Direction Due to Shielding

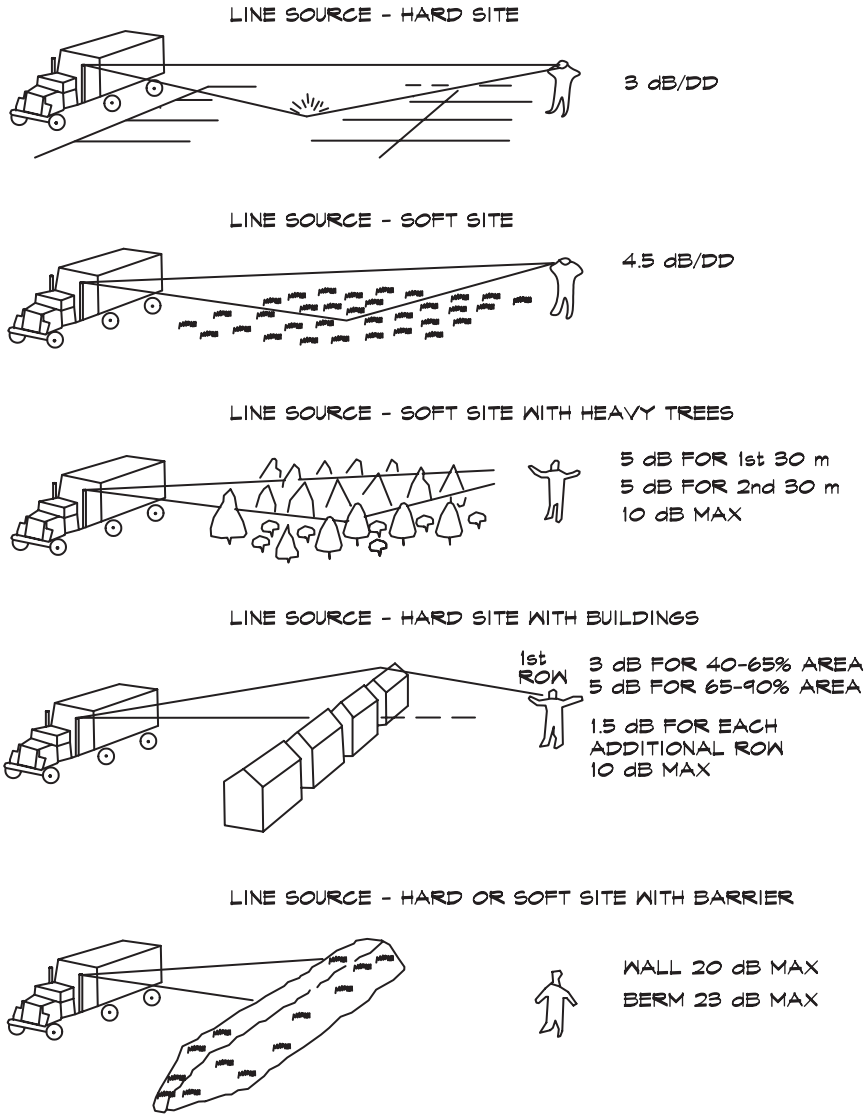


as follows: 1) air attenuation, 2) ground effects, 3) losses due to focusing from wind and thermal gradients, and 4) channeling effects. Air attenuation is always present, even indoors, and contributes to the acoustics of concert halls as well as sound propagation outdoors. Ground effects occur when the sound grazes at a shallow angle over acoustically soft materials such as thick grass, plowed ground, fresh snow, or in theaters, padded opera chairs. Ground effects are not significant unless there is grazing. Wind and thermal focusing are other commonly occurring outdoor phenomena. Channeling effects are more rare, occurring over water, in gullies at night, or in the atmosphere when inversion conditions are present.

Air Attenuation

The theory of sound attenuation, which was discussed in Chapt. 2, was based on the geometrical spreading of acoustical energy due to distance. No internal losses were assumed in that analysis. In a real fluid there are several additional mechanisms that attenuate a sound, including viscous and thermal losses and various relaxation effects. The combination of these

FIGURE 5.12 Attenuation of Highway Noise (Barry and Reagan, 1978)



effects is termed atmospheric attenuation and is made up of four components

$$\Delta L_a = \Delta L_{cl} + \Delta L_{rot} + \Delta L_{vib}(O_2) + \Delta L_{vib}(N_2) \tag{5.14}$$

where ΔL_a = total atmospheric attenuation (dB/km)

ΔL_{cl} = classical losses due to viscosity and thermal effects (dB/km)

ΔL_{rot} = molecular absorption for rotational relaxation of oxygen and nitrogen molecules (dB/km)

$\Delta L_{vib}(O_2)$ = molecular absorption losses for vibrational relaxation of O_2 molecules (dB/km)

$\Delta L_{vib}(N_2)$ = molecular absorption losses for vibrational relaxation of N_2 molecules (dB/km)

All these terms represent different ways in which sound energy is converted into heat or internal energy of the air, thus reducing the strength of the sound wave.

Classical attenuation comes about through the effects of viscosity and thermal conductivity, illustrated in Fig. 5.13. As a wave passes a point, the pressure and thus the temperature increases. Temperature is a measure of the amount of random molecular motion in the fluid. In a region of high temperature, there are more high-speed molecules, which will diffuse into the surrounding cooler regions to equalize the temperature. Once diffused this energy is not available to the sound wave and is lost. Viscosity is also a diffusion effect, the diffusion of particle momentum. A portion of the fluid with a high momentum slides past a region of lower momentum and some of the molecules lose energy due to collisions with the adjacent fluid. Viscosity and heat conductivity make approximately equal contributions to sound attenuation.

When a property of a system, say the temperature of a fluid, is forced away from its equilibrium state and then allowed to relax back to equilibrium, the time it takes to return to the original state is called a *relaxation time*. Relaxation times of natural phenomena can range from microseconds to centuries, depending on the physical process in question (Reif, 1965). Several relaxation effects increase the attenuation of sound propagating through the air. All have to do with the transfer of energy from a translational mode to other molecular energy modes, either rotational or vibrational. Figure 5.14 shows several examples. If a molecule of air (either N_2 or O_2) undergoes an increase in velocity due to the presence of a sound wave, it will, in turn, transfer that energy to other molecules through collisions.

FIGURE 5.13 Classical Air Attenuation Mechanisms

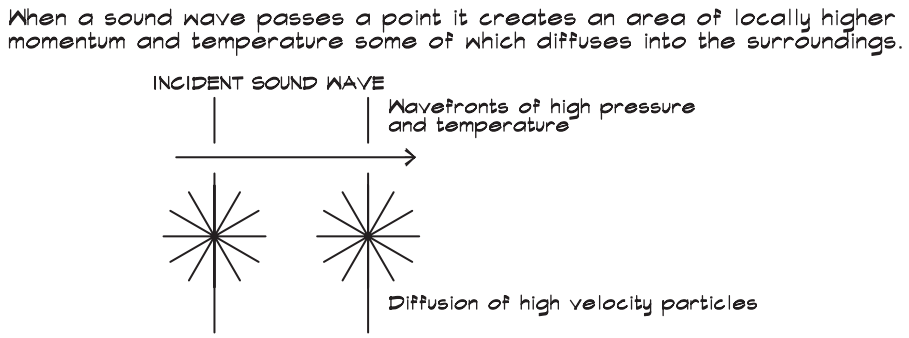
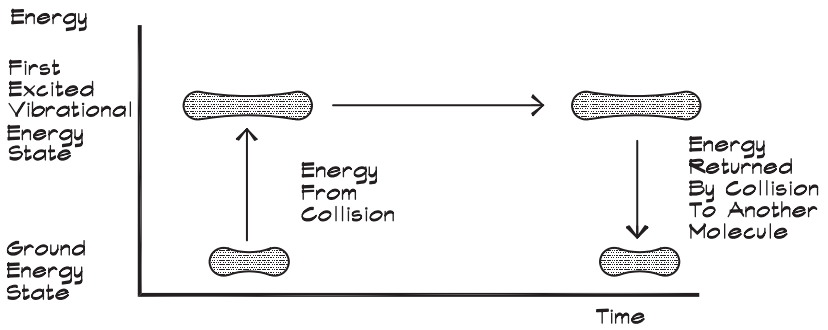


FIGURE 5.14 Energy Transfer to Vibrational Energy States



Some energy may be transferred into exciting rotational or vibrational modes of the molecule it impacts and some into creating pure translational motion. When a molecule in an excited state impacts another, some or all of the rotational or vibrational energy may be converted back into particle velocity. The temperatures of translation, rotation, and vibrational motion locally tend toward equilibrium, an effect called the *equipartition of energy*. Each of these equilibrium processes has a different characteristic relaxation time.

If the relaxation time is very long—that is, if it takes a long time to transfer energy back and forth between translation and vibrational motion—then a sound wave, which generates rapid increases or decreases in sound pressure and therefore sound temperature, is unaffected by energy transfer to other modes, which takes place too slowly to influence its passage. The fluid is said to be in a "frozen" condition in so far as this energy transfer mechanism is concerned. Similarly if the relaxation time is very short—that is, much shorter than the time for changes in pressure due to the sound wave to occur—then the energy transfer back and forth between translation and vibration happens so quickly that the fluid is in a state of thermal "equilibrium" between the various energy modes, and again, there is little effect on the passage of the sound wave.

If the relaxation time is just the right value, then when a pressure wave passes by, the increased translational energy is converted into vibration and then back into translation coincident with the arrival of the low pressure region. The wave amplitude is attenuated since the acoustic energy is converted either to random molecular motion (heat) or to pressure that is out of phase. When the acoustic frequency is of the same order of magnitude as the relaxation frequency $(1/2\pi\tau)$ of a particular vibrational mode, air can induce significant sound attenuation. The relaxation frequencies and maximum attenuation amplitudes vary with the type of molecule, the mode of vibration, and the presence of other types of molecules such as water vapor.

Rotational relaxation times are very short and equilibrium conditions can be achieved within a few molecular collisions. Energy losses at normal acoustic frequencies are small and can be combined with classical effects. The expected loss due to both classical and rotational effects is (ISO 9613-1, 1990).

$$\Delta L_{cl} + \Delta L_{rot} = (1.60 \times 10^{-7})\sqrt{\frac{T_k}{T_0}} f^2 / p^* \tag{5.15}$$

where T_k = absolute temperature in degrees K = temperature in degrees Celsius plus 273.15
T_0 = reference temperature = 293.15 (degrees K)
f = frequency (Hz)
p^* = normalized atmospheric pressure in standard atmospheres (pressure in k Pa divided by 101.325)

For a mild day

$$T_k = 288°K \text{ (15°C or 59°F)}$$
$$p^* = 1$$

and

$$\Delta L_{cl} + \Delta L_{rot} = 1.6 \times 10^{-7} f^2 \tag{5.16}$$

or about 0.6 dB per kilometer at 2000 Hz.

The remaining two absorption terms in Eq. 5.14, which account for most of the air absorption, are due to vibration relaxation mechanisms in nitrogen and oxygen molecules. In this process diatomic molecules are excited by a collision with another molecule into a higher vibrational energy state, and return the energy at a later time through another collision. The process takes many molecular collisions to complete and the attenuation follows the form (ISO 9613-1, 1990)

$$\Delta L_{vib} = 8686\, \mu_{max}\, \frac{f}{c} \left[\frac{2\left(f / f_{r\,i}\right)}{1 + \left(f / f_{r\,i}\right)^2} \right] \tag{5.17}$$

where ΔL_{vib} = vibrational air attenuation (dB/km)
μ_{max} = maximum loss in nepers for one wavelength (Np)
$f_{r\,i}$ = frequency of maximum loss per wavelength (Hz) where the subscript i refers to oxygen or nitrogen
f = frequency of sound wave (Hz)
c = speed of sound (m/s)

Two variables that appear in Eq. 5.17 require some explanation: the maximum loss per wavelength, μ_{max}, and the relaxation frequency, $f_{r\,i}$, at which this maximum loss occurs. The thermodynamic basis for the value of μ_{max} has been given in several texts (Kinsler et al., 1982, or Pierce, 1981). For oxygen and nitrogen, it is dependent on the absolute temperature

$$\mu_{max} = \frac{2\,\pi K}{35} \left(\frac{\theta_i}{T_k} \right)^2 e^{-\theta_i / T_k} \tag{5.18}$$

where θ_i = characteristic temperature corresponding to a particular vibrational mode, (°K) for oxygen, $\theta_i = 2239.1$°K and for nitrogen, $\theta_i = 3352.0$°K
K = volume concentration of the gas in the air which for oxygen = 0.209 and for nitrogen = 0.781
T_k = absolute temperature (°K)

At 20°C the predicted values of μ_{max} are 0.00105 for oxygen and 0.00020 for nitrogen, which are quite close to the actual measured values. Since μ_{max} is expressed in terms of a loss per wavelength, the relaxation loss curve has the peak shown in Fig. 5.15. To convert this into a loss per distance we must multiply it by the number of wavelengths per distance, which increases with frequency, to obtain the curves shown in Fig. 5.16. The conversion from nepers to decibels (1 Np = 20 log e = 8.686 dB) is used to obtain Eq. 5.17.

The molecular vibration losses for O_2 and N_2 molecules are strongly influenced by the presence of water vapor in the air, since a vibrational energy transfer is more likely for a collision between H_2O and O_2 or N_2. Thus water vapor catalyzes the transfer of energy between the modes and reduces the vibrational relaxation time. This phenomenon is particularly important in the design of concert halls, where dry air can result in considerable loss of high-frequency energy. The HVAC systems in large halls must be designed to control not only temperature but also humidity.

Having defined μ_{max} there remains the definition of the two relaxation frequencies, $f_{r\,o}$ and $f_{r\,n}$, for the vibrational modes of diatomic oxygen and nitrogen. Both these frequencies are a function of humidity, atmospheric pressure, and temperature. The following equation

FIGURE 5.15 Air Absorption Due to Relaxation Processes (Pierce, 1981)

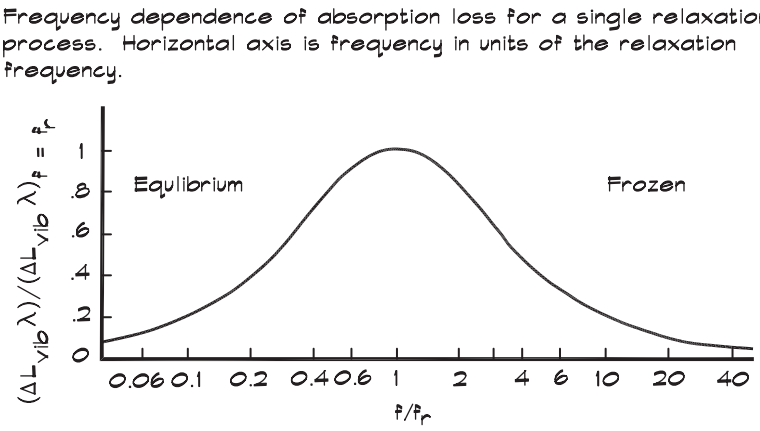
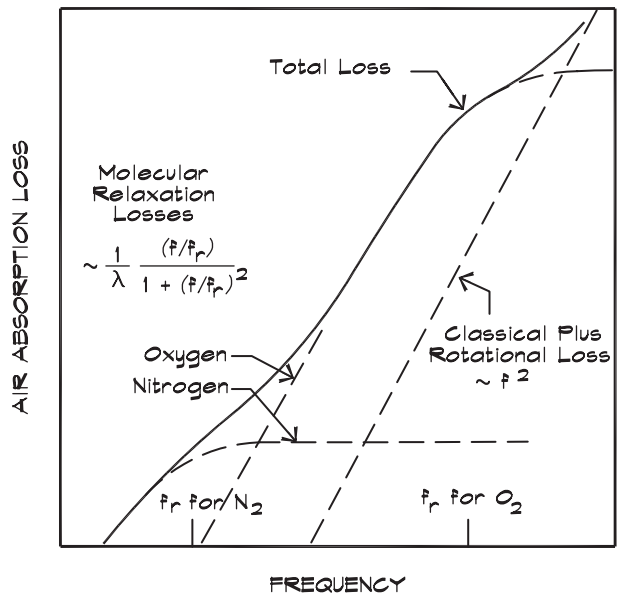


FIGURE 5.16 Components of Atmospheric Absorption Loss



for the relaxation frequency for oxygen was developed by Piercy (1971) and slightly modified in the ISO standard

$$f_{r\,o} = p^* \left\{ 24 + 40\,400\, h \left[\frac{0.02 + h}{0.391 + h} \right] \right\} \tag{5.19}$$

where p^* = absolute pressure in standard atmospheres

h = absolute humidity (% mole ratio)

$$= h_r \frac{10^{\left[-6.8346(T_0/T_k)^{1.261} + 4.6151\right]}}{p^*}$$

h_r = relative humidity (%)

TABLE 5.1 Air Attenuation at 20° C, dB/km (ISO 9613-1, 1990)

Freq. (Hz) Rel. Hum (%)	10	15	20	30	40	50	60	70	80	90	100
50	2.70 −1	2.14 −1	1.74 −1	1.25 −1	9.65 −2	7.84 −2	6.60 −2	5.70 −2	5.01 −2	4.47 −2	4.03 −2
63	3.70 −1	3.10 −1	2.60 −1	1.92 −1	1.50 −1	1.23 −1	1.04 −1	8.97 −2	7.90 −2	7.05 −2	6.37 −2
80	4.87 −1	4.32 −1	3.77 −1	2.90 −1	2.31 −1	1.91 −1	1.62 −1	1.41 −1	1.24 −1	1.11 −1	1.00 −1
100	6.22 −1	5.79 −1	5.29 −1	4.29 −1	3.51 −1	2.94 −1	2.52 −1	2.20 −1	1.94 −1	1.74 −1	1.58 −1
125	7.76 −1	7.46 −1	7.12 −1	6.15 −1	5.21 −1	4.45 −1	3.86 −1	3.39 −1	3.02 −1	2.72 −1	2.47 −1
160	9.65 −1	9.31 −1	9.19 −1	8.49 −1	7.52 −1	6.60 −1	5.82 −1	5.18 −1	4.65 −1	4.21 −1	3.84 −1
200	1.22+0	1.14+0	1.14 +0	1.12 +0	1.05 +0	9.50 −1	8.58 −1	7.76 −1	7.05 −1	6.44 −1	5.91 −1
250	1.58 +0	1.39 +0	1.39 +0	1.42 +0	1.39 +0	1.32 +0	1.23 +0	1.13 +0	1.04 +0	9.66 −1	8.95 −1
315	2.12 +0	1.74 +0	1.69 +0	1.75 +0	1.78 +0	1.75 +0	1.68 +0	1.60 +0	1.50 +0	1.41 +0	1.33 +0
400	2.95 +0	2.23 +0	2.06 +0	2.10 +0	2.19 +0	2.23 +0	2.21 +0	2.16 +0	2.08 +0	2.00 +0	2.90 +0
500	4.25 +0	2.97 +0	2.60 +0	2.52 +0	2.63 +0	2.73 +0	2.79 +0	2.80 +0	2.77 +0	2.71 +0	2.63 +0
630	6.26 +0	4.12 +0	3.39 +0	3.06 +0	3.13 +0	3.27 +0	3.40 +0	3.48 +0	3.52 +0	3.52 +0	3.49 +0
800	9.36 +0	5.92 +0	4.62 +0	3.84 +0	3.77 +0	3.89 +0	4.05 +0	4.19 +0	4.31 +0	4.39 +0	4.43 +0
1 000	1.41 +0	8.72 +0	6.53 +0	5.01 +0	4.65 +0	4.66 +0	4.80 +0	4.98 +0	5.15 +0	5.30 +0	5.42 +0
1 250	2.11 +1	1.31 +1	9.53 +0	6.81 +0	5.97 +0	5.75 +0	5.78 +0	5.92 +0	6.10 +0	6.29 +0	6.48 +0
1 600	3.13 +1	1.98 +1	1.42 +1	9.63 +0	8.00 +0	7.37 +0	7.17 +0	7.18 +0	7.31 +0	7.48 +0	7.68 +0
2 000	4.53 +1	2.99 +1	2.15 +1	1.41 +1	1.12 +1	9.86 +0	9.25 +0	9.02 +0	8.98 +0	9.06 +0	9.21 +0
2 500	6.35 +1	4.48 +1	3.26 +1	2.10 +1	1.61 +1	1.37 +1	1.25 +1	1.18 +1	1.15 +1	1.13 +1	1.13 +1
3 150	8.54 +1	6.62 +1	4.94 +1	3.18 +1	2.39 +1	1.98 +1	1.75 +1	1.61 +1	1.53 +1	1.48 +1	1.45 +1
4 000	1.09 +2	9.51 +1	7.41 +1	4.85 +1	3.61 +1	2.94 +1	2.54 +1	2.29 +1	2.13 +1	2.02 +1	1.94 +1
5 000	1.33 +2	1.32 +2	1.09 +2	7.39 +1	5.51 +1	4.44 +1	3.79 +1	3.36 +1	3.06 +1	2.86 +1	2.71 +1
6 300	1.56 +2	1.75 +2	1.56 +2	1.12 +2	8.42 +1	6.78 +1	5.74 +1	5.04 +1	4.54 +1	4.18 +1	3.91 +1
8 000	1.75 +2	2.21 +2	2.15 +2	1.66 +2	1.28 +2	1.04 +2	8.78 +1	7.66 +1	6.86 +1	6.26 +1	5.81 +1
10 000	1.93 +2	2.67 +2	2.84 +2	2.42 +2	1.94 +2	1.59 +2	1.35 +2	1.18 +2	1.05 +2	9.53 +1	8.79 +1

The relaxation frequency for nitrogen in air is based in part on a compilation of data by Evans (1972)

$$f_{rn} = \frac{p^*}{\sqrt{T_k/T_0}} \left\{ 9 + 280\, h\, e^{-4.17\left[(T_0/T_k)^{1/3} - 1\right]} \right\} \tag{5.20}$$

Even though the equations for air attenuation are somewhat convoluted, the calculations are straightforward. Table 5.1 contains a partial summary of the results. At 4000 Hz, about the highest note on the piano, the air loss at room temperature is 109 dB/km at 10% relative humidity, but only 23 dB/km at 70% relative humidity. In a large concert hall where a 2-second reverberation time is desirable the air loss alone is 36 dB/sec at 10% relative humidity. This forces the reverberation time down to 1.67 s solely due to air losses. Clearly, the humidity must be controlled to limit high-frequency attenuation.

Attenuation Due to Ground Cover

Ground attenuation is caused by several effects including the losses in propagating through dense woods or heavy foliage, and the effect of grazing propagation at shallow angles over an acoustically soft surface such as heavy grass, plowed ground, or new fallen snow. *Grazing attenuation* is also present in concert halls and theaters, where sounds, emanating from a performer on stage, pass over seated patrons or padded opera chairs and induce losses over and above those expected from geometrical spreading and air absorption alone.

The absorption mechanism associated with sound propagation through dense forests and foliage is mostly scattering from the trunks and limbs. Sound absorption by leaves is not a significant contributor (Beranek and Ver, 1992). Hoover (1961) has developed an approximate formula for excess attenuation in forests

$$\Delta L_f \cong 10 \left(\frac{f}{1000} \right)^{1/3} \left(\frac{r}{100} \right) \text{ (dB)} \tag{5.21}$$

which works out to be about 10 dB per 100 m at 1 kHz. Other authors have measured attenuations ranging from 3 dB per 100 m for bare trees, to 18–27 dB per 100 m for heavy Canadian forests. Equation 5.21 represents the average of data compiled for all types of American forests.

Grazing Attenuation

Grazing attenuation is a phenomenon that occurs when an acoustic wave interacts with an absorbing surface at a shallow angle of incidence. When a sound wave reflects off a surface, the reflected wave combines with the original wave and may produce an increase or decrease in overall signal amplitude. A diagram of the geometry is shown in Fig. 5.17. When the reflecting surface is very hard, the reflected wave combines with the incident wave to produce an increase in sound pressure. At low frequencies and short distances this can be as high as 6 dB near the boundary. At very high frequencies the two waves combine incoherently with no particular phase relationship, which can produce a doubling in intensity and a 3 dB increase in level. When the reflecting material is soft the reflected wave is out of phase with the incident wave. Since the two waves are traveling along closely matching paths the combined wave is highly attenuated. This phenomenon and the general theory of reflection from surfaces is discussed in more detail in Chapt. 7.

FIGURE 5.17 Geometry of Grazing Attenuation

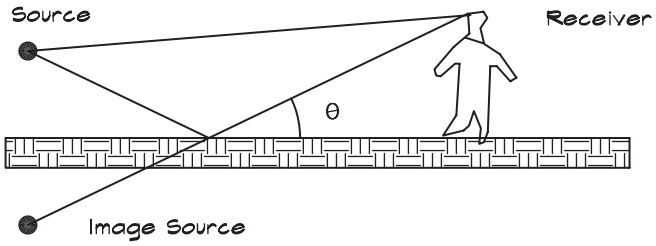


FIGURE 5.18 Grazing Attenuation Regions



Empirical formulas have been developed (ISO 9613-2.2, 1994) for the prediction of excess grazing attenuation. This method subdivides the ground into three regions, shown in Fig. 5.18: source and receiver regions, no more than 30 times the respective receiver heights from the source or receiver; and a middle region that includes the remainder of the intervening ground. If the overall separation distance is less than $30(h_s + h_r)$, where h_s and h_r are the source and receiver heights, then there is no middle ground and the near and far regions overlap. Each region is rated by a coefficient, ξ ranging from 0 to 1, according to the fraction that is soft (porous). The equations are grouped into the standard octave-band frequencies. Note that negative values indicate an increase in signal level.

$$\Delta L_g = \begin{cases} -3 - 3M & \text{at 63 Hz} \\ -3 + \xi_s A_s + \xi_r A_r - 3M(1 - \xi_m) & \text{at 125 Hz} \\ -3 + \xi_s B_s + \xi_r B_r - 3M(1 - \xi_m) & \text{at 250 Hz} \\ -3 + \xi_s C_s + \xi_r C_r - 3M(1 - \xi_m) & \text{at 500 Hz} \\ -3 + \xi_s D_s + \xi_r D_r - 3M(1 - \xi_m) & \text{at 1 kHz} \\ -1.5(1 - \xi_s) - 1.5(1 - \xi_r) - 3M(1 - \xi_m) & 2k - 8\ \text{kHz} \end{cases} \tag{5.22}$$

The constants A, B, C, D, and M are defined in terms of either h_s or h_r, and the source-receiver separation distance, r (in meters)

$$M = 1 - \frac{30\left(h_s + h_r\right)}{r} \quad \text{when} \quad r > 30\left(h_s + h_r\right)$$

$$M = 0 \quad \text{when} \quad r \leq 30\left(h_s + h_r\right)$$

$$A = 1.5 + 3.0\left[1 - e^{-r/50}\right] e^{-0.12(h-5)^2} + 5.7\left[1 - e^{-2.8 \times 10^{-6} r^2}\right] e^{-0.09\, h^2}$$

$$B = 1.5 + 8.6\left[1 - e^{-r/50}\right] e^{-0.09\,h^2}$$

$$C = 1.5 + 14.0\left[1 - e^{-r/50}\right] e^{-0.46\,h^2}$$

$$D = 1.5 + 5.0\left[1 - e^{-r/50}\right] e^{-0.9\,h^2}$$

Grazing attenuation is only present at very shallow angles, less than 5°. As the angle increases the ground becomes more reflective. Since grazing waves are highly attenuated the effect has been described as a shadow zone for source and receiver locations close to the ground.

Ground surfaces are grouped according to their acoustical properties as follows:

1. *Hard ground:* Concrete, asphalt, water, ice, or other surfaces having a low porosity such as compacted earth or rock.
2. *Soft ground:* Grass, plowed earth, dense vegetation, soft snow.
3. *Mixed ground:* A mixture of hard and soft areas.

For hard ground the constant ξ is 0; for soft ground it is 1. For a mixture of hard and soft ground the constant is the fraction of soft ground.

Focusing and Refraction Effects

The path taken by a sound wave propagating through a conducting medium such as the air is the one that minimizes the time it takes to get from the source to the receiver. This physical law is called the *principle of least time*, and applies to the motion of objects. Since the velocity of sound varies with the temperature as well as the velocity due to the motion of the medium, the path of least time does not always correspond to the path of minimum distance. If the air itself is moving, its velocity is added vectorially to the sound propagation velocity to obtain an overall velocity. These changes in velocity can affect the path taken by the sound rays and the sound levels at the receiver.

When a wind blows along the ground its velocity increases with height. Thus, if a sound wave propagates in the downwind direction, the upper part, which is at a higher elevation, will travel faster than its lower part. A surface of constant phase connecting the top and bottom parts will bend downward as illustrated in Fig. 5.19. Sound that propagates downwind can travel along a path that is unaffected by grazing and even barrier attenuation. Therefore for long propagation distances the average levels encountered in quiescent conditions, where the effects of shielding and ground effects come into play, can be much lower than levels experienced when the wind blows toward the receiver.

FIGURE 5.19 Downwind Wave Propagation in a Parabolic Velocity Profile (Scale Exaggerated)

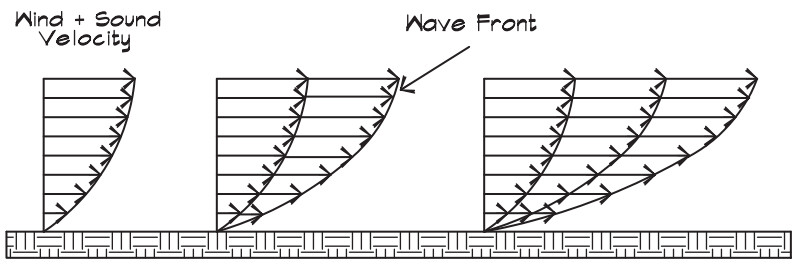


FIGURE 5.20 Wave Propagation in a Wind Gradient

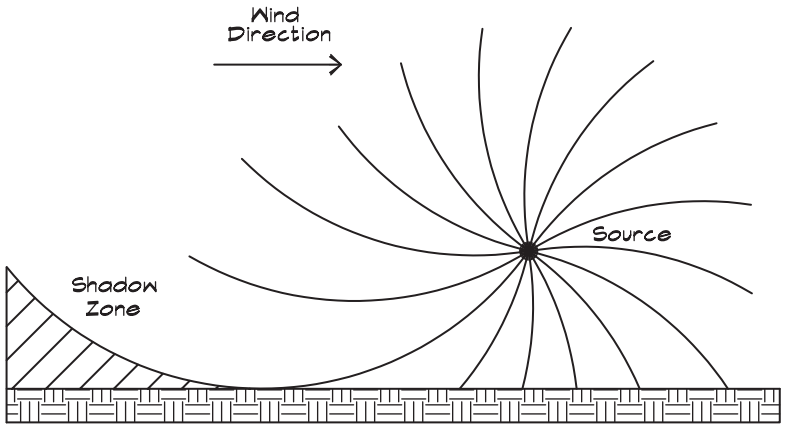
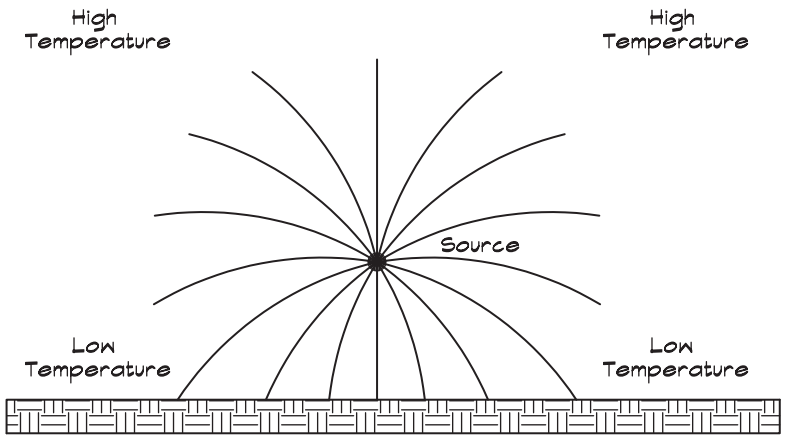


FIGURE 5.21 Wave Propagation in a Thermal Inversion



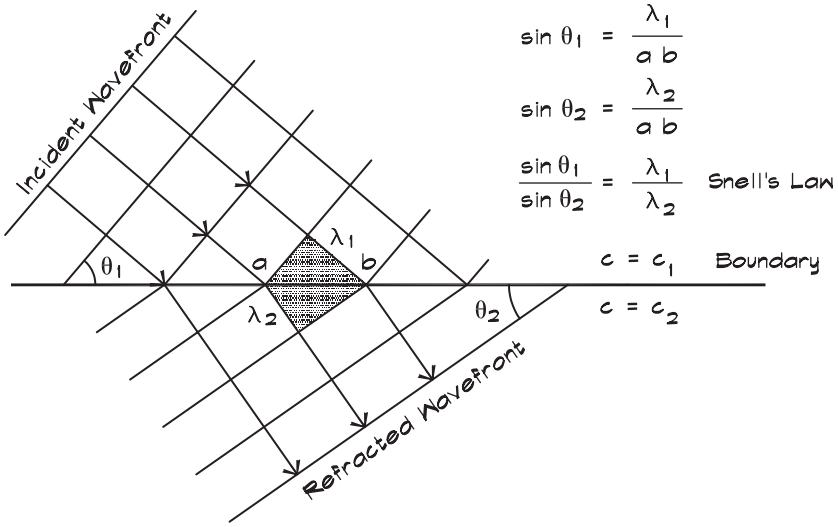
In Fig. 5.20, when a wave travels upwind, and the upper part of the wave is slowed by the higher wind velocity, the propagation path bends upward. In the upwind direction, if a sound emanates from a source at some height above the ground, there will be a curved sound path that just impacts the ground, leaving a shadow zone beyond the impact point, where in theory no sound penetrates. In practice, upwind losses in the shadow zone can be 20 dB or more.

Sound velocities can vary with temperature as well as with wind velocity. Since the velocity is dependent on the square root of the absolute temperature, when the temperature decreases with height the sound rays bend upward in all directions. If an inversion layer is encountered, where temperatures increase with height, sound rays bend down in all directions, much like a fireworks starburst shown in Fig. 5.21.

The change in the sound velocity causes a change in direction of the sound path, a situation illustrated in Fig. 5.22. If two adjacent regions have sound velocities c_1 and c_2, their wavelengths are given by

$$\lambda = \frac{c}{f} \Rightarrow \lambda_1 = \frac{c_1}{f} \Rightarrow \lambda_2 = \frac{c_2}{f} \tag{5.23}$$

FIGURE 5.22 Wave Propagation at a Change in Velocity



In order for the wavelengths to match at the interface between the two zones there must be a change in direction such that

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{\lambda_1}{\lambda_2} \tag{5.24}$$

where θ is the angle between the wavefront and the horizontal. This relationship is known as Snell's law, and the ratio of the sine terms is called the index of refraction, when the waves are light rather than sound. Snell's law can also be expressed in terms of the velocities of sound in the two media as

$$\frac{\sin\theta_1}{c_1} = \frac{\sin\theta_2}{c_2} \tag{5.25}$$

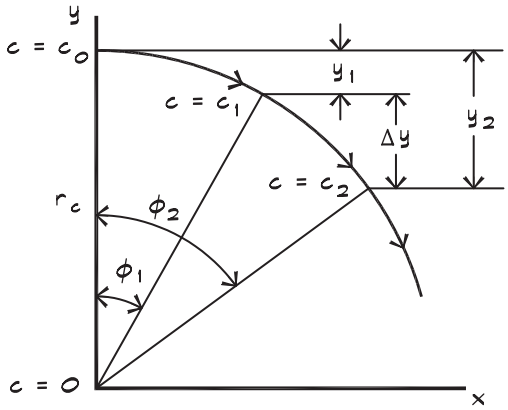
When the sound velocity is a linear function of the height above the ground we can calculate the shape of the sound path. If we assume sound velocity is a function of height, in Fig. 5.23, which follows the linear relationship, $c(y) = A\,y + B$, where A and B are constants. A is the slope of the line and B is the y axis intercept, which is taken to be zero at $c(y) = 0$. Substituting into Eq. 5.25 we obtain

$$\cos\,\phi\,(y) = \frac{y}{r_c} \quad \text{and} \quad c(y) = A\,y \tag{5.26}$$

This is the formula for a circle having radius r_c, where at the top of the circle $\cos\phi\,(y) = 1$ and $c(y) = c_0$. At this point we can solve for the radius r_c of curvature of the sound ray

$$r_c = \frac{c_0}{A} \tag{5.27}$$

FIGURE 5.23 Relation Between a Velocity Gradient and the Radius of Curvature of a Ray (Kinsler et al., 1982)



where the origin (y = 0) is measured from the point where the sound velocity extrapolates to zero, which is usually below ground level for a positive gradient. Note that the term A in Eq. 5.27 could be due to changes in both wind and thermal gradients. The radius of curvature is (Gutenberg, 1942)

$$r_c = \frac{c_0}{(d\,c / d\,y)\cos\phi + d\,v_x / dy} \tag{5.28}$$

where r_c = radius of curvature (m)

c_0 = velocity of sound where the ray is horizontal (m/s)

$\frac{1}{c_0} = \frac{\cos\phi}{c}$ is the velocity at angle ϕ (Snell's Law)

$\frac{d\,c}{d\,y}$ = rate of change of sound velocity with height (1/s)

$\frac{d\,v_x}{d\,y}$ = rate of change of wind velocity with height (1/s)

ϕ = angle that the wavefront makes with the y axis (rad)

When r_c is positive (sound velocity increasing with height) the sound ray bends down and when it is negative (sound velocity decreasing with height) it bends up.

If the wind velocity changes from 0 km/hr to 20 km/hr (5.6 m/s) in 5 meters, A would be 4 km/m hr (1.1 s^{-1}). If the velocity at the top of the circle is 40 km/hr (11.2 m/s), then the radius of curvature is the quiescent speed of sound 344 + 11.2 = 355.2 m/s divided by A, yielding a radius of about 322 m. If we assume that a high school band (L_w = 100 dB) is playing and we are located 150 m away, the arc of the sound ray at its highest point is about 10 m above the ground. This path could comfortably clear intervening barriers resulting in a clearly audible sound (45 dB). Without the wind the same source could be inaudible at this distance due to ground effects and shielding. This type of calculation is based on a worst-case scenario without focusing. The most likely scenario would probably include barrier shielding and ground effects if appropriate.

Beranek and Ver (1992) have cited unpublished data by G. S. Anderson giving the approximate attenuations due to refraction over soft ground with and without barrier shielding. Without a barrier the predicted refraction attenuation is

$$\Delta L_r = \begin{cases} -3.0 \log \dfrac{r}{15} & \text{for case 1} \\ 0 & \text{for case 2} \\ \left(10 - 6.2\,h + 0.03\,h^2\right) \log \dfrac{r}{15} \geq 0 & \text{for case 3} \\ \left(14 - 7.9\,h + 0.3\,h^2\right) \log \dfrac{r}{15} \geq 0 & \text{for case 4} \end{cases} \tag{5.29}$$

where h is the average unrefracted height over the ground and r is the propagation distance.

Where barriers are present the refraction attenuation can be estimated by

$$\Delta L_r = \begin{cases} -3 & \text{for case 1} \\ 0 & \text{for case 2} \\ +3 & \text{for case 3} \end{cases} \tag{5.30}$$

The case numbers refer to (1) moderate downward refraction, (2) no refraction, (3) moderate upward refraction, and (4) strong upward refraction. These equations were developed for a wind velocity of about 5 m/s at 3 m above ground. In case 4 it is assumed that both wind and thermal gradients are present.

When wind or thermal gradients are present above a highly reflecting surface, such as the surface of a lake or other large body of water, the sound rays are refracted downward and then reflected off the surface of the water. This process is repeated and the outcome is a channeling of the sound. Since the usual geometric spreading does not occur downwind, levels are considerably higher than normally would be expected. Some channeling can also occur along gullies. In the evening the bottoms of gullies are cooler than the atmosphere above them and are often rocky, providing a highly reflecting surface. Instances of atmospheric focusing have been encountered in which sound is transmitted over distances of a mile or more with localized attenuations much less than geometric falloff would predict (Bronsdon, 1998).

Combined Effects

When calculating the sound pressure levels from a generalized point source out of doors the various loss factors must be combined. The overall level is obtained from the sound power level of the source

$$L_p = L_w + 10 \log \frac{Q}{4\pi r^2} - \Delta L_b - \Delta L_g - \Delta L_f - \Delta L_r - \Delta L_a + 0.5 \tag{5.31}$$

Appropriate loss terms should be included in accordance with the dictates of the physical situation. For example, air attenuation is always present but it would be inappropriate to include both barrier and refraction losses without modification by a relationship such as

Eq. 5.30. If downward refraction is present then grazing loss would not be included in a calculation but forest attenuation might be, depending on the situation.

Doppler Effect

When a sound source moves, its motion shifts the frequency heard by a stationary observer. A familiar example is a railroad train and the lowering of the frequency of the sound it emits as it passes by. Let us examine in Fig. 5.24 a source located at point a, emitting a square wave that propagates a distance ($c\ \Delta t$), in time Δt, to reach a point b. If the same source were to move from a to a′ with a velocity u, while emitting the same square wave, a compressed signal would result. The front of the first pulse would arrive at b at the same time Δt, but the end of the last pulse would be emitted at the point a′. Thus the pulse train would be compressed into a distance that is ($u\ \Delta t$) less than $\overline{a\ b}$.

$$\overline{a\ b} - \overline{a'\ b} = u\ \Delta t \tag{5.32}$$

The velocity of sound is not affected by the source movement but the apparent wavelength λ' is. The velocity $c = \lambda\ f = \lambda'\ f'$ can be substituted into Eq. 5.32

$$\lambda\ f\ \Delta t - \lambda'\ f\ \Delta t = u\ \Delta t \tag{5.33}$$

and

$$\lambda' = (f\ \lambda - u)\ /\ f \tag{5.34}$$

Hence there is an apparent frequency at the receiver

$$f' = f\ \lambda\ /\ \lambda' = \frac{c}{c - u}\ f \tag{5.35}$$

If we examine the case where the receiver is moving along the same straight line at a velocity v, the apparent frequency is

$$f' = \frac{c - v}{c}\ f \tag{5.36}$$

FIGURE 5.24 Doppler Effect—Frequency Shift Due to Relative Source and Receiver Velocity (Seto, 1971)

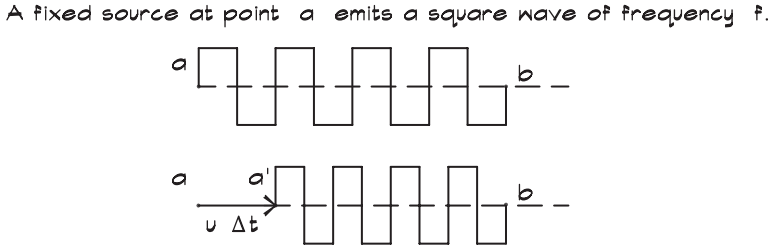
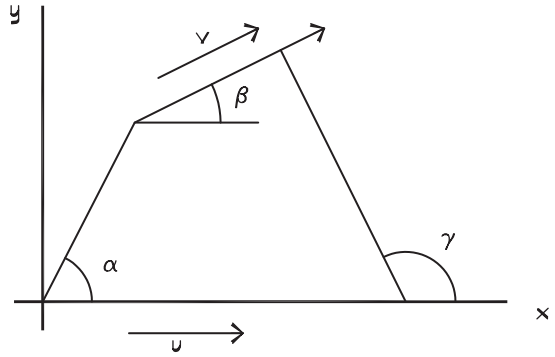


FIGURE 5.25 Geometry of the Doppler Effect for Planar Motion (Seto, 1971)



If both the source and receiver are moving along the line between them then the apparent frequency is

$$f' = \frac{c - v}{c - u} f \tag{5.37}$$

In the case of generalized planar motion of both source and receiver, as shown in Fig. 5.25, the observed frequency is

$$f' = \frac{c - v \cos(\gamma - \beta)}{c - u \cos \alpha} f \tag{5.38}$$

When there is a wind present with a velocity w, the frequency becomes

$$f' = \frac{c - v_x + w_x}{c - u_x + w_x} f \tag{5.39}$$

where the velocities are shown in terms of their x components for simplicity. Note that a wind has no effect on the apparent frequency unless there is also motion of the source or receiver.

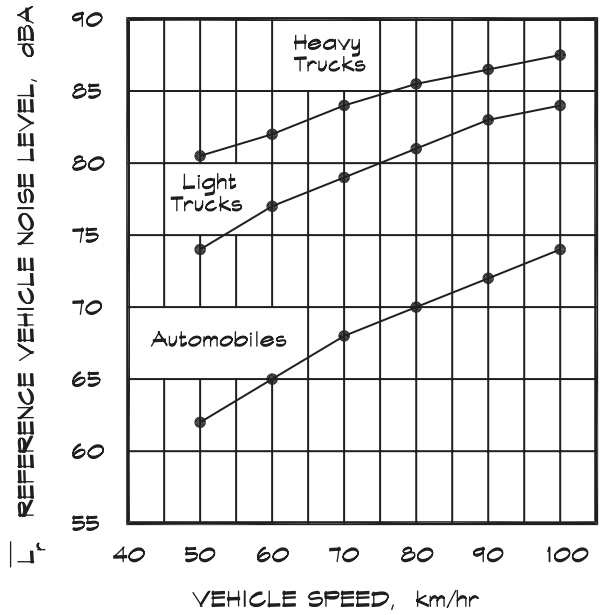
5.4 TRAFFIC NOISE MODELING

Traffic noise modeling starts with a determination of the radiated sound levels of individual vehicles. For this purpose vehicles are grouped into three categories: automobiles, light trucks, and heavy trucks. Both the measured sound pressure level at a known distance, typically 15 m, and the effective source height are necessary for each category. Vehicle noise is reported as a sound level versus speed.

The vehicle categories for sound prediction purposes are defined as follows (Barry and Reagan, 1978):

1. *Automobiles (A)* - All vehicles having two axles and four wheels designed primarily for the transportation of nine or fewer passengers (automobiles) or the transportation of cargo (light trucks). The gross vehicle weight is less than 4500 kilograms.

FIGURE 5.26 Reference Energy Mean Vehicle Emission Levels (Barry and Reagan, 1978)



2. *Medium Trucks (MT)* - All vehicles having two axles and six wheels designed for the transportation of cargo. The gross vehicle weight is greater than 4500 kilograms but less than 12,000 kilograms.
3. *Heavy Trucks (HT)* - All vehicles having three or more axles and designed for the transportation of cargo. The gross vehicle weight is greater than 12,000 kilograms.

It is important to determine the vehicle weight definition used by the source of traffic data. Frequently state highway departments include medium trucks in their heavy truck category. When this type of count is used, the calculated noise levels are too high, so the truck count must be adjusted.

The expected noise levels at 15 m are given in Fig. 5.26 and

$$\begin{aligned} \overline{L_r}(A) &= 38.1 \log (v_k) - 2.4 \\ \overline{L_r}(MT) &= 33.9 \log (v_k) + 16.4 \\ \overline{L_r}(HT) &= 24.6 \log (v_k) + 38.5 \end{aligned} \tag{5.40}$$

for each class of vehicle, where v_k is the vehicle speed in km/hr.

Soft Ground Approximation

When roadway sound propagates over soft ground, such as grass or soft earth, the rate of falloff with distance can be adjusted to approximate the loss due to ground effect. The way this normally is done is to assume a 4.5 dB/distance doubling falloff rate over the soft site

region, rather than the normal 3 dB/dd falloff rate used for hard sites. The 3 dB rate is used 1) in all cases where the source and receiver are located 3 m or more above the ground, 2) whenever there is an intervening barrier 3 m or greater in height, and 3) when the line of sight is less than 3 m above the ground and the ground is hard and there are no intervening structures. The 4.5 dB falloff rate is used when the view of the roadway is interrupted by isolated buildings, clumps of bushes, trees, or the intervening ground is soft or covered with vegetation.

When the soft ground falloff rate is used the adjustment factor for the point source attenuation for a single vehicle moving along a roadway can be written as

$$\Delta L_d = 10 \log \left(\frac{d_r}{r} \right)^{\frac{\xi + 4}{2}} \tag{5.41}$$

where ΔL_d = attenuation due to distance (dB)
r = distance between the source and receiver (m or ft)
d_r = distance at which the reference level was measured (usually 15 m or 50 ft)
$\xi = 0$ for a hard site
$\xi = 1$ for a soft site

The differences in the maximum levels measured at two distances, d and 2 d from the roadway, would be 6 dB for hard sites and 7.5 dB for the soft sites, 6 dB for divergence and 1.5 dB for ground attenuation. The corresponding line source falloff rate is 3 dB/dd for a hard site and 4.5 dB/dd for a soft site.

When calculating levels from roadway segments using the soft-site approximation, the rule of equal noise from equal angle segments no longer holds. The road segment correction, which is given by the last term in Eq. 5.4 for a hard site, must now be integrated over the included angles (Barry and Reagan, 1978)

$$L_{eq} = \overline{L}_r + 0.115\,\sigma^2 + 10 \log \frac{N}{T} + 10 \log \int_{t_1}^{t_2} \frac{d_r^{\frac{\xi + 4}{2}}}{\left(\sqrt{d^2 + v^2\,t^2} \right)^{\frac{\xi + 4}{2}}}\, dt \tag{5.42}$$

where L_{eq} = equivalent sound level (dB or dBA)
$\overline{L}_r$ = average reference sound level at distance d_r (dB or dBA)
d = distance of closest approach (m or ft)
v = source speed (m/s or ft/s)
t = time (s)
N = number of vehicles passing the measurement point in time T
σ = standard deviation of the reference sound level (dB or dBA) - usually taken to be 2.5 dBA for automobiles, 3 dBA for medium trucks, and 3.5 dBA for heavy trucks

The integral is done by making the substitution $vt = d\tan\phi$ and $vdt = d\sec^2\phi\, d\phi$. The result is

$$L_{eq} = \bar{L}_r + 0.115\,\sigma^2 + 10\,\log\frac{N\,\pi\,d_r}{T\,v} + 10\,\log\frac{d_r}{d} + 10\,\log\left(\frac{d_r}{d}\right)^{\frac{\xi}{2}} + \psi_\xi\,(\phi_1, \phi_2) \tag{5.43}$$

where $\psi_\xi\,(\phi_1, \phi_2) = 10\log\frac{1}{\pi}\int_{\phi_1}^{\phi_2}(\cos\phi)^{\frac{\xi}{2}}\,d\phi$ is the adjustment factor for included angle.

The first line of Eq. 5.43 is the equivalent sound level due to a line of vehicles moving along an infinite roadway. The second line is the adjustment due to distance and the finite roadway segment, which is applied to sites having excess attenuation. When $\xi = 0$, Eq. 5.43 reduces to Eq. 5.7. When $\xi = 1$, the integral can be done numerically and the results are shown in Fig. 5.27. The soft-site approximation is convenient mathematically, but for more accurate calculations it is recommended that the line source be subdivided into segments less than 25° and the ground effects be calculated for each segment using Eq. 5.22. For values of ξ falling between 0 and 1 the segment approach must be used.

Geometrical Mean Distance

Accurate traffic calculations can be done by apportioning the average daily traffic (ADT) on a roadway among the various lanes and calculating the levels for each lane. This technique preserves the geometrical relationship between the lanes and any roadside barriers. Heavy trucks are assumed to be located in the outside lanes. Standard lane spacing in the United States is 12 ft (3.66 m) for traffic lanes and 10 ft (3 m) for parking lanes. In a computer model of traffic noise, a near-lane and a far-lane distance can be used, from which it is straightforward to increment the distances by 12 feet for each lane. This system accounts for any median width.

For rough calculations, when the distance to the roadway is large compared with the road width, the distance to the centerline can be used. For approximate calculations at a closer distance the geometrical mean distance is sometimes used. It is defined as

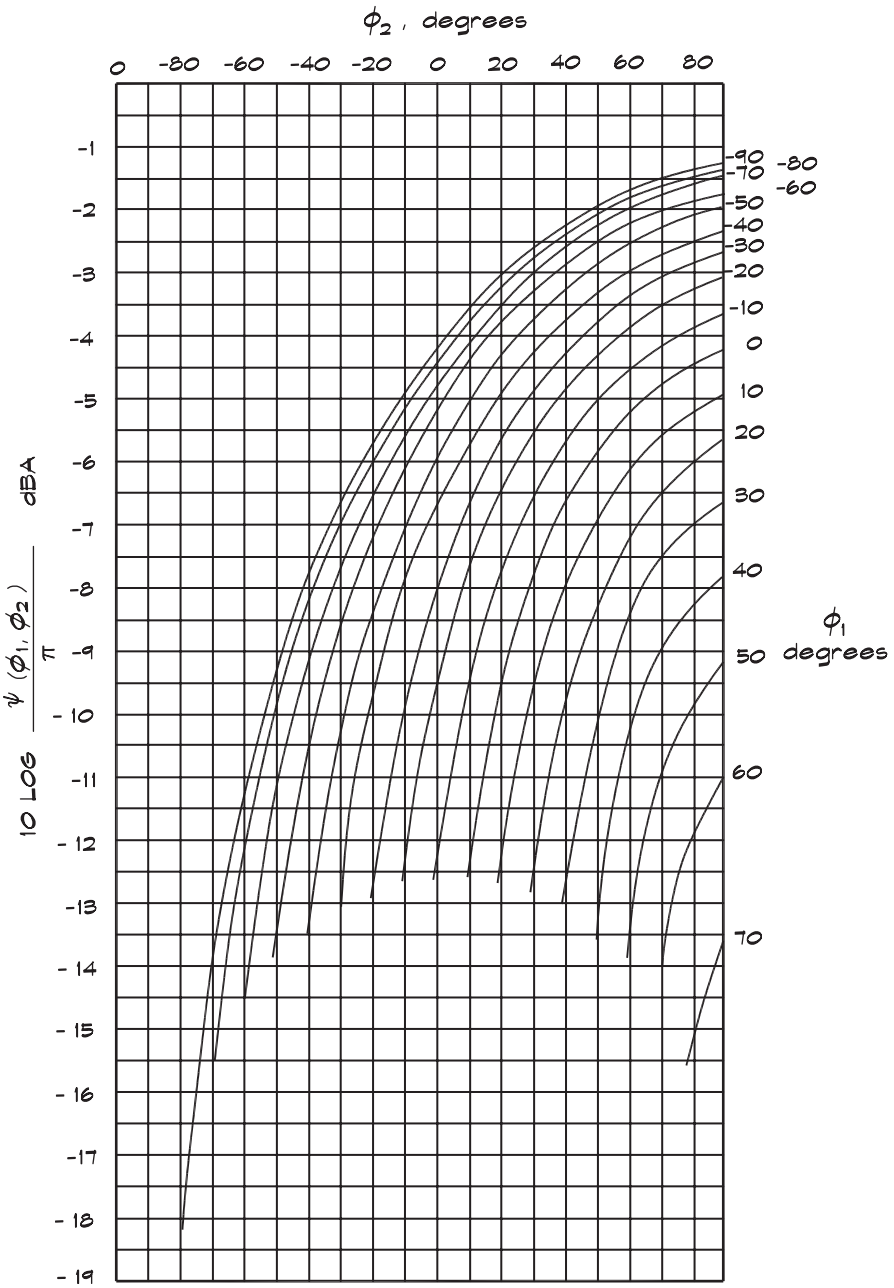
$$d_{g\,m} = \sqrt[n]{d_1\,d_2 \ldots d_n} \tag{5.44}$$

where $d_{g\,m}$ = geometrical mean distance to the roadway (m or ft)
n = number of lanes
d_1 = perpendicular distance to the center of lane 1 (m or ft)
d_2 = perpendicular distance to the center of lane 2 (m or ft)
d_n = perpendicular distance to the center of lane n (m or ft)

Barrier Calculations

For barrier calculations a source height is required that depends on the part of the vehicle that creates the noise. For automobiles above about 55 km/hr (35 mph), the road-tire interaction is the predominant noise source so the source height is zero. The actual interaction between vehicle tires and the roadway is quite interesting and is the subject of considerable study.

FIGURE 5.27 Adjustment Factor for Finite Length Roadways for Absorbing Sites (Barry and Reagan, 1978)



Much progress has been made, principally in Europe, on the development of porous road surfaces, which have been quite effective in reducing traffic-generated noise (Hamet, 1996). For trucks, the noise source is a combination of the truck exhaust, the motor and cooling fan, and the road-tire interaction, with the exhaust stack predominating at highway speeds. The usual height is 2.44 m (8 feet) above the roadway. For medium trucks the source height is 0.7 m (2.3 feet).

When barrier attenuations are calculated, the Fresnel number in Eq. 5.11 requires that the wavelength and thus the frequency of interest be known. Calculations also can be carried out in each octave band and the resultant levels combined in the normal way. In order to reduce the number of calculations, studies were undertaken to determine the frequency that best matched the overall result from the full calculation, based on a traffic spectrum. The result was 550 Hz, which corresponds to a wavelength of about 0.63 m (2 feet). Although the spectra of the various classes vary somewhat, the 550 Hz value is used for all vehicle types.

Roadway Computer Modeling

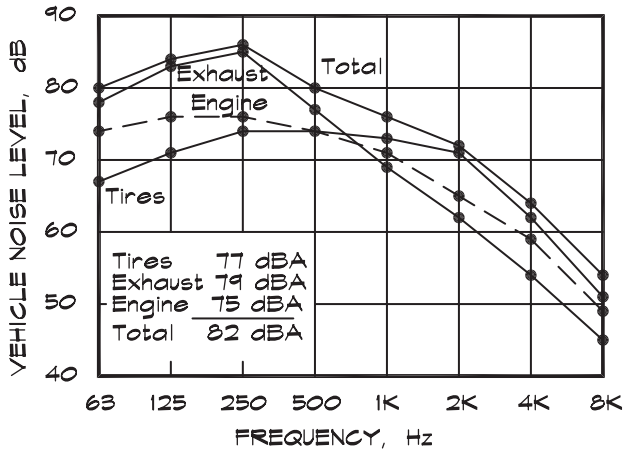
Where a series of partial barriers shield a roadway, the sound level from the unshielded portion can be calculated and combined with that received from the shielded portion. The calculation of the barrier shielding can be done in increments by assuming that a short line segment is a point source or by integrating the barrier attenuation over the included angles. In practice incremental technique is simpler to use for computer calculations. In both cases simplifying assumptions are made for ease of calculation. In the FHWA hard site model the Fresnel number is approximated using Eq. 5.13 to allow computation of the integral. Where this technique is employed the equation for the equivalent level from an increment of line source is

$$L_{eq}(\phi) = \overline{L}_r + 0.115\,\sigma^2 + 10\log\frac{N\,\pi\,d_r}{v\,T} + 10\,\log\frac{d_r}{d} + 10\,\log\frac{\Delta\phi}{\pi} - \Delta L_{bg}(\phi) \tag{5.45}$$

where $L_{eq}(\phi)$ = equivalent sound level (dB or dBA)
$\overline{L}_r$ = average reference sound level at distance d_r (dB or dBA)
d = distance of closest approach (m or ft)
v = source speed (m/s or ft/s)
t = time (s)
N = number of vehicles passing the measurement point in time T
σ = standard deviation of the reference sound level (dB or dBA)
$\Delta L_{bg}(\phi)$ = barrier or ground attenuation as defined in Eq. 5.11 or Eq. 5.22 for a point source located at the centerline of travel at angle ϕ (dB)
$\Delta\phi$ = included angle of the barrier element (radians)

Since the geometry of the source-barrier configuration changes with angle, vehicle type, and lane, separate calculations should be carried out for each of these groups, and the results combined. If the source-receiver distance is sufficiently great, air attenuation may also have to be included for each segment. For overall A-weighted levels it is sufficiently accurate to

FIGURE 5.28 Spectral Components of Heavy Truck Noise at 50 Feet at Highway Cruising Speeds (Anderson et al., 1973)



calculate air attenuation for traffic sources at 250 Hz. Software is available from the Federal Highway Administration for doing these computations.

Traffic Noise Spectra

Detailed acoustical analyses such as those required for the transmission of sound from the exterior to the interior of a building are done using the octave-band noise spectrum of a source. For traffic noise a composite spectrum is used that depends on the percentage of heavy trucks present in the vehicle mix. For an individual heavy truck, noise is emitted from the exhaust, engine, cooling system, and tires. The contribution from each source is shown in Fig. 5.28 (Anderson et al., 1973). The overall sound pressure level for these data is 82.4 dBA. By subtracting this number from each octave-band level, a generalized unweighted truck noise spectrum is obtained, normalized to zero dBA. These differences can be added to the desired overall A-weighted truck noise level to obtain the individual octave-band levels.

Figure 5.29 gives normalized heavy truck spectra for various speeds, which when subtracted from an overall A-weighted level results in an A-weighted third-octave band level. For A-weighted octave band levels three adjacent bands must be combined or 4.8 dB added to the center band to approximate the octave-band level. In a similar way A-weighted spectra can be obtained for automobiles from Fig. 5.30, and for medium trucks from Fig. 5.31.

It is frequently necessary to generate a noise spectrum given a certain vehicle mix. For example, we might predict an overall level of 70 dBA from a given roadway for a vehicle mix of 3% heavy trucks and 97% automobiles. To determine the overall spectrum we need to calculate the level generated by each class of vehicle, obtain the spectrum of each vehicle type using the normalized adjustment, and combine the results.

5.5 RAILROAD NOISE

The prediction of railway noise is similar to the calculation of traffic noise. There are two major sources, locomotive engines and rail cars. For purposes of this analysis only line

FIGURE 5.29 Emission Spectra, Heavy Trucks, Cruise Throttle, Average Pavement (TNM, 1998)

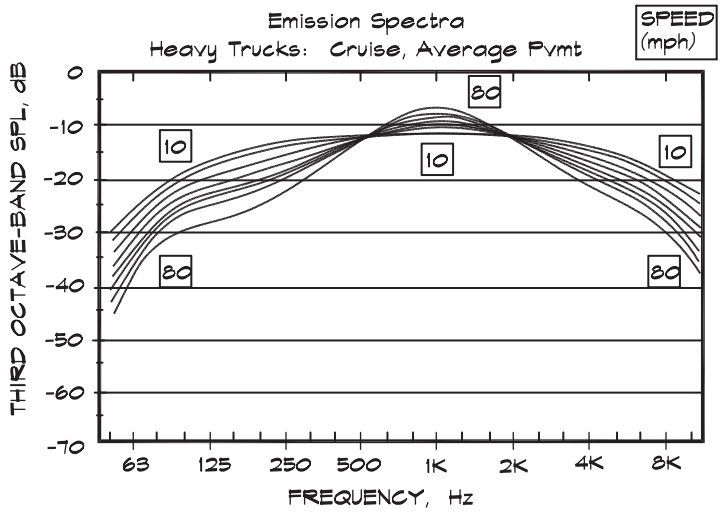
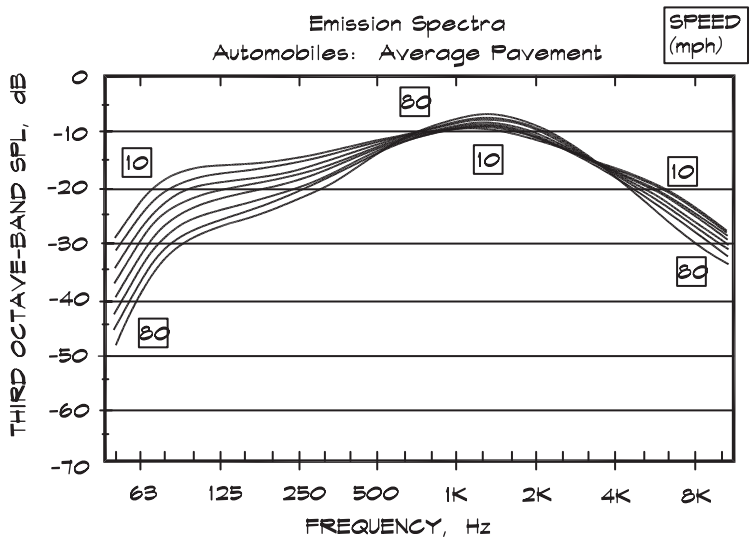


FIGURE 5.30 Emission Spectra, Automobiles, Average Pavement (TNM, 1998)



operations will be addressed. Line operations refer to the movements of railroad locomotives and freight or passenger cars over a main line or branch line of tracks. General railroad yard noise and noise due to horns and crossing signals will not be included. For a more detailed analysis of yard noise and other sources associated with general railroad operations, refer to Swing and Pies (1973).

A train noise time history is shown in Fig. 5.32. Locomotive noise is generated by the large diesel-electric engines, typically 2500 to 3000 horsepower (2 mw), which singly or in groups pull the long strings of cars. Locomotive noise depends on the engine throttle setting

FIGURE 5.31 Emission Spectra, Medium Trucks, Cruise Throttle, Average Pavement (TNM, 1998)

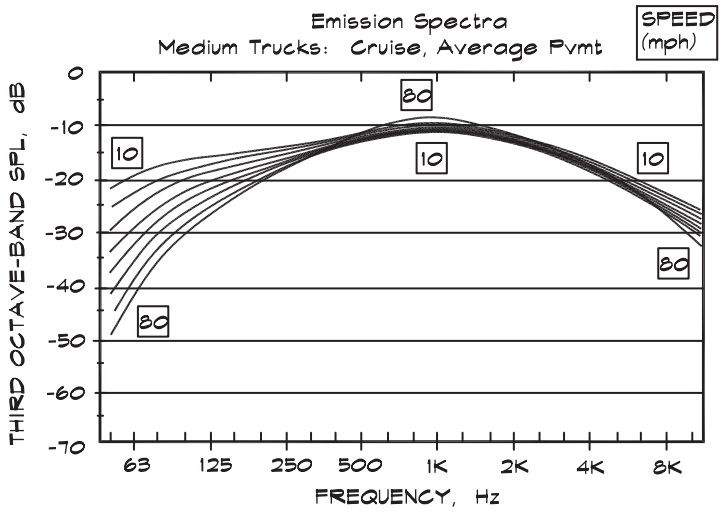
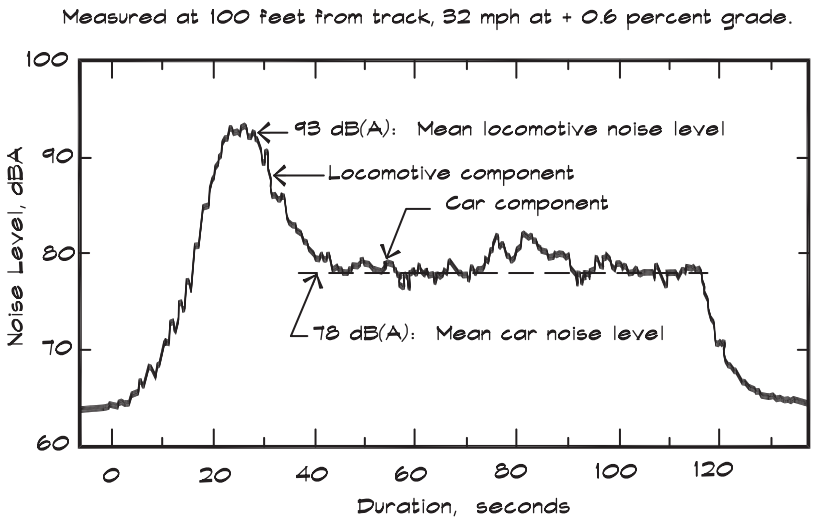


FIGURE 5.32 Typical Time History of a Train Passby (Swing and Pies, 1973)



and somewhat on the grade but is relatively independent of velocity. Reference sound levels from a sample of engine passbys are shown in Fig. 5.33. Levels average 91 dBA at 100 ft. Grade adjustments are given in Fig. 5.34.

Rail car noise is produced by wheel-rail interaction and is a strong function of velocity. Rail car noise levels can be estimated using an empirical formula (Swing and Pies, 1973)

$$L_r(\text{cars}) = 50 + 20 \log v_m \tag{5.47}$$

FIGURE 5.33 Noise Levels of a Locomotive at Level Grade (Swing and Pies, 1973)

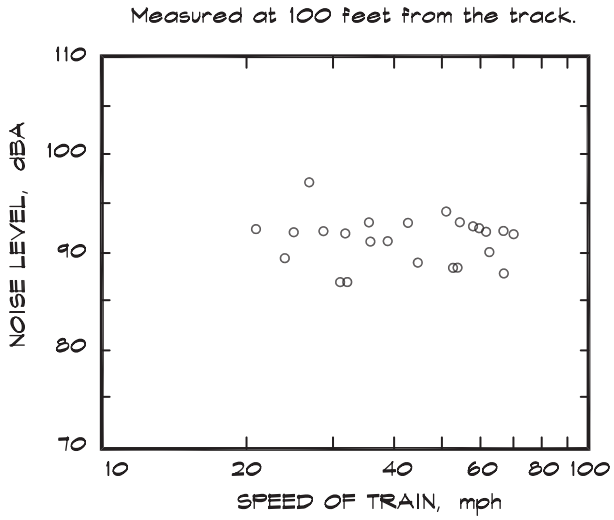
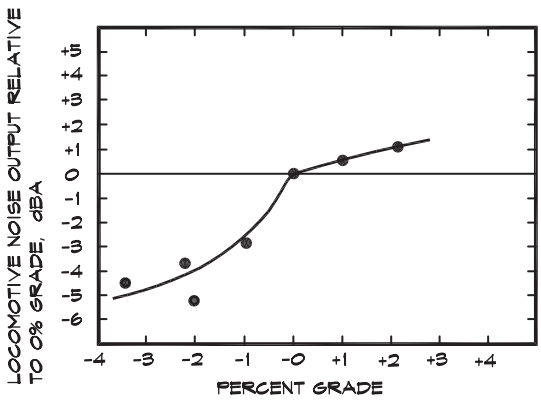


FIGURE 5.34 Grade Dependence of Locomotive Noise (Swing and Pies, 1973)



where $L_r(\text{cars})$ = reference rail car level at 30 m (100 ft) (dBA)
v_m = speed of train (mph = 1.61 kph)

Some variations in level are encountered due to different track conditions, which are summarized in Table 5.2. The base level in Eq. 5.47 assumes jointed rails, no crossings, smooth wheels, and no bridgework or short-radius curves.

The normal modeling procedure is to establish a sound exposure level (SEL) for a given train passage and then to adjust the level by the number of like events to determine a day-night level or other metric of interest. Recall that the sound exposure level was defined in Eq. 4.11 and is the energy-time product expressed as a level. Combining this equation with Eq. 5.5 we obtain the SEL value for a locomotive passby

$$\text{SEL}_{\text{loc}} = L_r(\text{loc}) + 10\ \log \frac{\pi d_r}{v} + 10\ \log \frac{d_r}{d} \tag{5.48}$$

TABLE 5.2 Variables Affecting Freight Car Wheel/Rail Noise Emission (Swing and Pies, 1973)

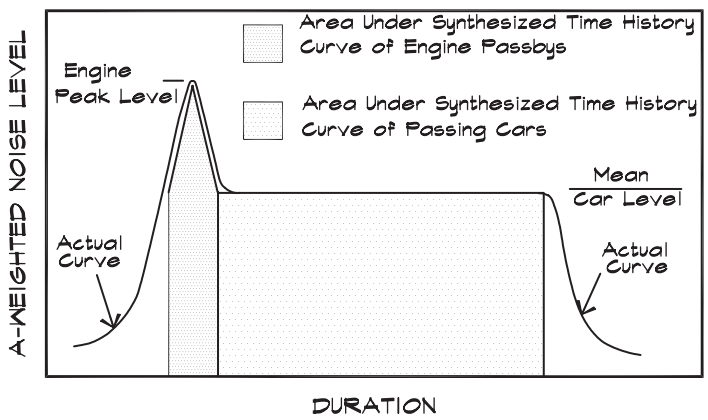
Variable	Increase in Noise Level*	Comments
1. Jointed Rails (vs Welded)	4 to 8 dB(A)	Generally no correction for main line tracks; assign higher value to low speed classified track
2. Presence of Grade Crossings and Frogs	6 to 8 dB(A)	
3. Wheel Irregularities – Flat Spots or Built-up Tread	to 15 dB(A)	
4. Passage Over Bridgework a. Light Steel Structure b. Heavy Steel Structure c. Concrete Structure	 to 30 dB(A) to 15 dB(A) 0 to 12 dB(A)	Use lower range of corrections for heavier structures
5. Short Radius Curves a. Less than 600 ft. Radius b. 600 to 900 ft. Radius	 15 to 25 dB(A) 5 to 15 dB(A)	Random occurrence of wheel squeal

* These factors are assumed to act individually. When in combinations of two or more the net increase will not be equal to the sum of each component, but most likely the largest individual factor.

where $L_r(\text{loc})$ = reference locomotive sound level at d_r (dBA)
v = speed of train (m/s)
d = perpendicular distance to the track (m)

When there are multiple locomotives a factor of 10 log N is added in. Figure 5.35 sets forth the modeling scheme. The locomotives are calculated as single passbys. The cars are calculated

FIGURE 5.35 Railroad Noise Modeling Scheme (Swing and Pies, 1973)



as a steady level having a given duration. For a line of rail cars

$$SEL_{cars} = L_r(cars) + 10\ \log\ t + 10\ \log\ \frac{d_r}{d} \tag{5.49}$$

where L_r (cars) = reference rail car sound level at d_r (dBA)
t = time of car passage (s) = .68 l (ft) / v (mph) where l is the length of the train in feet or = 3.6 l (m) / v (kph) where l is in meters

Train lengths and velocities are available from the local railroad dispatch office. In the United States standard train cars are 60 ft (18.3 m) long and stretched cars used for automobile transport are 85 ft (25.9 m) long.

SEL levels from shielded train passbys can be obtained by applying partial line source, barrier, and ground effect adjustments to Eqs. 5.48 and 5.49. The source height of a locomotive engine stack is about 15 ft (4.6 m) and the car source height is zero. The barrier calculation frequency for railroads has not been published. A value of 550 Hz can be used if we assume that locomotive and truck spectra are similar.

Twenty-four hour L_{dn} noise levels can be calculated from SEL values by using the weighted number of trains during an average day

$$N_{dn} = N_d + N_e + 10\ N_n \tag{5.50}$$

where N_{dn} = weighted number of trains per day
N_d = number of daytime (7 am to 7 pm) trains
N_e = number of evening (7 pm to 10 pm) trains
N_n = number of nightime (10 pm to 7 am) trains

The day-night noise level due to all train events is

$$L_{dn} = 10\ \log\left[10^{0.1\ SEL_{loc}} + 10^{0.1\ SEL_{cars}}\right] + 10\ \log N_{dn} - 49.4 \tag{5.51}$$

The constant 49.4 is 10 log (3600 × 24), which normalizes the L_{dn} to the number of seconds in a day.

The spectrum for locomotives is shown in Fig. 5.36, and for cars, in Fig. 5.37. A composite spectrum can be generated by combining a locomotive spectrum adjusted to the SEL for locomotives and a car spectrum adjusted to the SEL for cars. The combined spectrum is then normalized to the L_{dn} level.

5.6 AIRCRAFT NOISE

Jet aircraft noise is generated primarily by the interaction of the high-velocity exhaust gasses with the relatively still atmosphere through which the aircraft passes. As the gasses mix with the surrounding air the resulting turbulence creates large pressure fluctuations, which radiate as sound. The region where most of the sound is formed is 5 to 8 diameters behind the exhaust nozzle. Consequently jet noise is difficult to enclose or otherwise muffle. Jet noise has several distinctive features including a very strong (eighth power) dependence on the velocity (v) of the flow. The frequency spectrum of jet noise is a flat haystack-shaped curve with the maximum scaled to the diameter (d) of the exhaust opening. Figure 5.38 illustrates this behavior. The peak occurs at a frequency $f_o = 0.13\ \frac{v}{d}$.

FIGURE 5.36 Spectral Components of Locomotive Noise (Swing and Pies, 1973) Measurements at 50 feet, 0% grade, and 58 mph

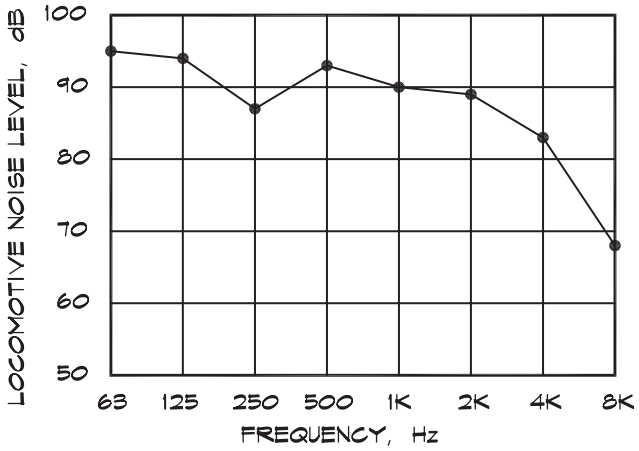
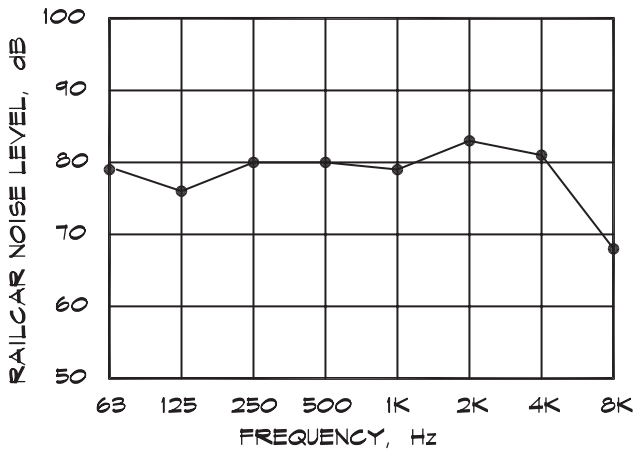


FIGURE 5.37 Spectral Components of Railroad Car Noise (Swing and Pies, 1973) Measurements at 100 feet, 0% grade, and 58 mph



Jet noise is highly directional. The exhaust produces lobes that have their maxima at between 30° and 45° degrees from the axis of the jet. At the front of the engine, high-frequency tonal components of the compressor fans are radiated from the intake. Thus on the approach side of an airport there is a greater high-frequency noise component than on the takeoff side.

Older commercial jet engines and most military aircraft engines are primarily turbojets. In a turbojet, the intake air is compressed by means of a set of rotating and a set of fixed blades. Jet fuel and compressed air are mixed together and burned in the combustion chamber and the hot gas expands and accelerates out the rear of the engine. A rear turbine is driven by the exhaust gasses and provides the power to turn the compressor. The turbojet produces a very high velocity exhaust flow, which dominates other noise sources, particularly at low frequencies. Some high-frequency noise is radiated from the intake compressor toward the front of the aircraft. The spectrum is in Fig. 5.39.

FIGURE 5.38 Generalized Power Spectrum of Jets (Harris, 1957)

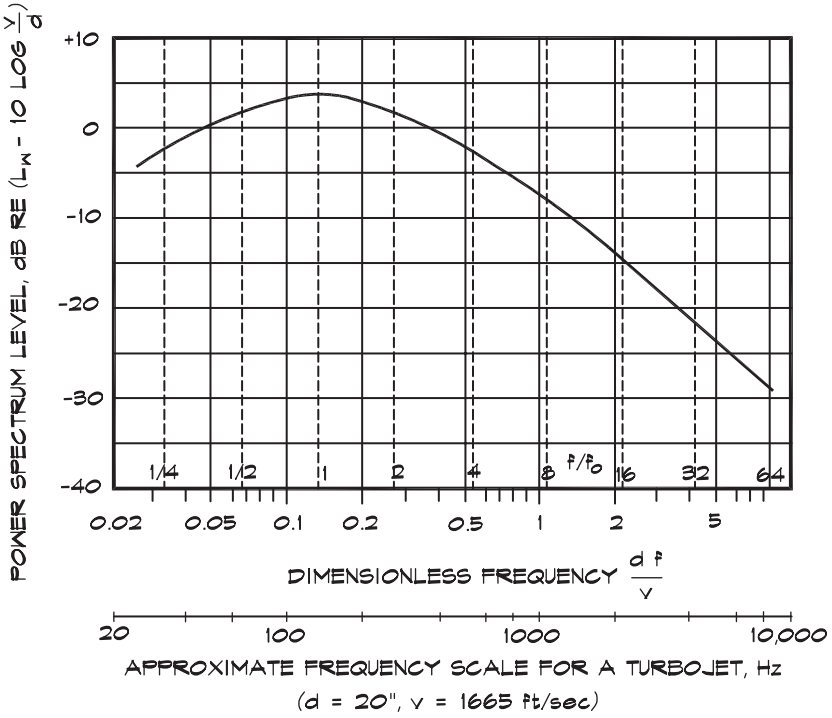


FIGURE 5.39 Noise Levels and Spectra of 2-3 Engine Low Bypass Ratio Turbofan Aircraft (Wyle Laboratories, 1971)

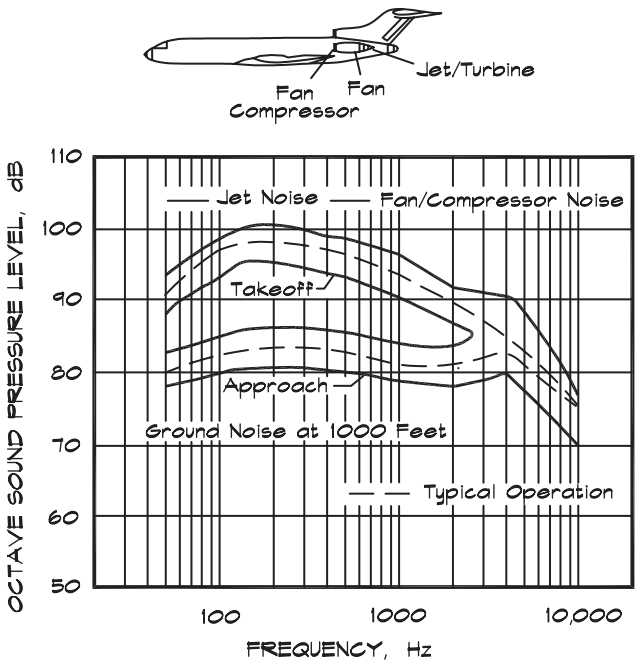
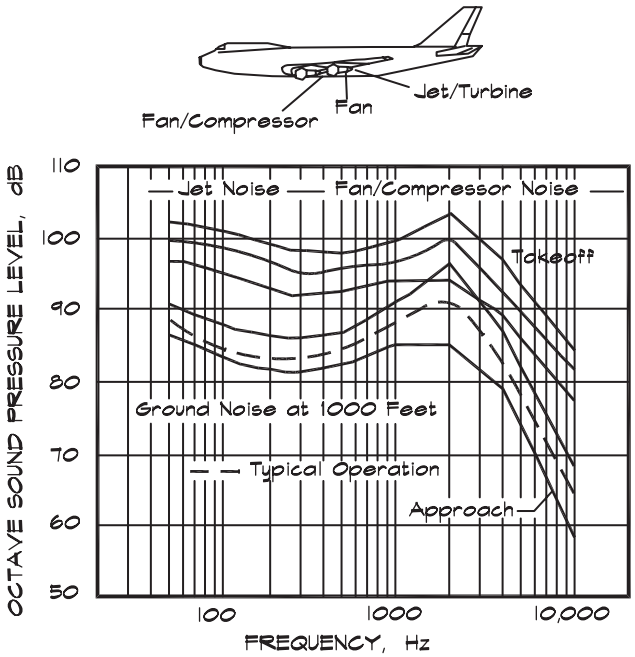


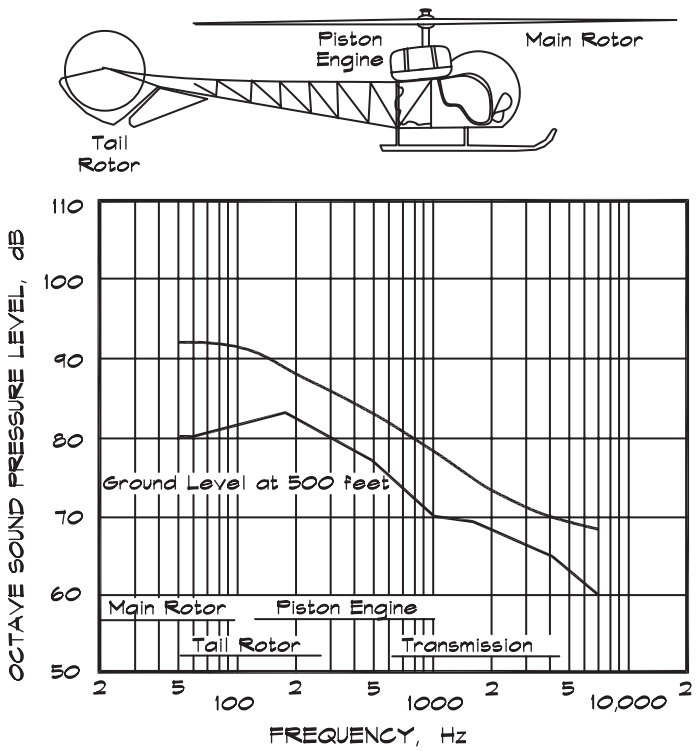
FIGURE 5.40 Noise Level and Spectra of a 4-Engine High Bypass Ratio Turbofan Aircraft (Wyle Laboratories, 1971)



The turbofan engine was developed, in part, as an attempt to reduce turbojet noise. The idea behind a turbofan design is to increase the diameter of the engine and to use the intake compressor as a propeller. Some of the inlet air is bypassed or routed around the combustion chamber and mixed back in with the combustion gas at the rear of the engine. This accomplishes several useful things. First, since the diameter of the intake is large, more air (mass) is forced through the opening. The engine thrust (fluid mass times acceleration) can increase even while the exhaust velocity and the noise level is reduced. Second, the bypassed air eases the transition between the high-velocity jet core and the quiescent atmosphere, so that the turbulent fluctuations due to mixing are less. Again this helps lower the noise levels. The amount of air bypassed divided by the total air passing through the intake is called the bypass ratio. Early turbofan engines such as the Pratt and Whitney JT8D used on the Boeing 707 are characterized as low bypass ratio engines. Engines such as the JT9D used on the Boeing 747 and the General Electric CF6 used on the McDonnell Douglas DC-10 aircraft are termed high bypass ratio engines and have a much lower noise output. A sketch of a turbofan powered aircraft and its spectrum is in Fig. 5.40.

Airport noise is measured at fixed monitoring stations on poles or buildings located around the airfield. In addition, computer models such as the Integrated Noise Model (INM), available from the Federal Aviation Administration (FAA), are used to calculate airport noise contours based on data measured on individual aircraft types. These contours are prepared for most major airports and regularly updated. For military airports Air Installation Compatible Land Use Zone (AICUZ) studies are prepared and are available to the public. Noise levels are shown as contours of equal level in terms of L_{dn}.

FIGURE 5.41 Typical Noise Spectra of Light Piston-Engine Helicopters (Wyle Laboratories, 1971)



For architectural acoustics calculations of aircraft noise, it is sufficient to obtain the day-night levels from a contour map. If maximum passby levels are of interest then on-site measurements are sometimes necessary. For a major airport such as Los Angeles International, a rough estimate can be obtained by adding 20 dBA to the day-night level. Detailed calculations of interior noise from passing aircraft require a knowledge of the specific spectrum. Where the source is a helicopter a typical noise spectrum is shown in Fig. 5.41.

6 WAVE ACOUSTICS

Much of architectural acoustics can be addressed without consideration of the wave nature of sound. For example, environmental acoustics and the transmission of outdoor sound, for the most part, can be visualized and modeled as a flow of energy from point to point, although many effects, such as ground and barrier attenuation, are frequency dependent. Nevertheless, for many critical aspects of acoustics, knowledge of wave phenomena is essential. Wave acoustics takes into account fundamental properties that are wavelength and phase dependent, including the scaling of interactions to wavelength, the phenomenon of resonance, and the combination of amplitudes based not only on energy but also on phase.

6.1 RESONANCE

Simple Oscillators

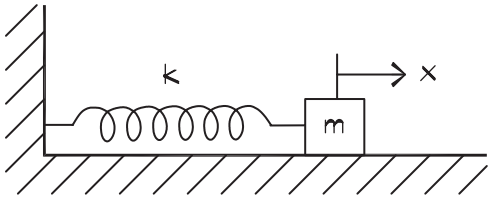
Many mechanical systems have forces that restore a body to its equilibrium position after it has been displaced. Examples include a spring mass, a child's swing, a plucked string, and the floor of a building. When such a system is pulled away from its rest position, it will move back toward equilibrium, transition through it, and go beyond only to return again and repeat the process. All linear oscillators are constrained such that, once displaced, they return to the initial position. The movement repeats at regular intervals, which have a characteristic duration and thus a characteristic frequency, called the natural frequency or resonant frequency of the system.

Although many such systems exist, the simplest mechanical model is the spring and mass shown in Fig. 6.1. A frictionless mass, m, is attached to a linear spring, whose restoring force is proportional to the displacement away from equilibrium, x. This relationship is known as Hooke's law and is usually written as

$$\mathrm{F} = -k\,\mathrm{x} \tag{6.1}$$

where k is the constant of proportionality or the spring constant.

FIGURE 6.1 A Simple Spring-Mass System



When the body is in motion, inertial forces, due to the mass, counterbalance the spring force, according to Newton's second law

$$F = m\,a = m\,\frac{d^2x}{d\,t^2} \tag{6.2}$$

where F = force applied to the mass, (N)
m = mass, (kg)
a = acceleration or the second time derivative of the displacement, (m/s^2)

The forces in a simple spring mass system are: 1) the spring force, which depends on the displacement away from the equilibrium position, and 2) the inertial force of the accelerating mass. The equation of motion is simply a summation of the forces on the body

$$m\,\frac{d^2\,x}{d\,t^2} + k\,x = 0 \tag{6.3}$$

If we introduce the quantity

$$\omega_n^2 = \frac{k}{m} \tag{6.4}$$

the equation can be written as

$$\frac{d^2\,x}{d\,t^2} + \omega_n^2\,x = 0 \tag{6.5}$$

The solution requires that the second derivative be the negative of itself times a constant. This is a property of the sine and cosine functions, so we can write a general solution as

$$x = A\ \sin\ \omega_n\,t + B\ \cos\ \omega_n\,t \tag{6.6}$$

where A and B are arbitrary constants defined by the initial conditions. If the system is started at t = 0 from a position x_0 and a velocity v_0, then the constants are defined and the equation becomes

$$x = \frac{v_0}{\omega_n}\ \sin\ \omega_n\,t + x_0\ \cos\ \omega_n\,t \tag{6.7}$$

Since we can write the coefficients in Eq. 6.6 in terms of trigonometric functions

$$a\ \sin\ \omega\,t + b\ \cos\ \omega\,t = \cos\ \phi\ \cos\ \omega\,t - \sin\ \phi\ \sin\ \omega\,t = \cos(\omega\,t + \phi) \tag{6.8}$$

and we can form a second general solution

$$x = X \cos(\omega_n t + \phi) \tag{6.9}$$

where

$$X = \sqrt{x_0^2 + (v_0 / \omega_n)^2} \quad \text{and} \quad \phi = \tan^{-1}\left[\frac{-v_0}{\omega_n x_0}\right] \tag{6.10}$$

and X = amplitude of the maximum displacement of the system, (m)
ϕ = initial phase of the system, (rad)

According to Eq. 6.9 the behavior of the spring mass system is harmonic, with the motion repeating every time period $t = T$, where $\omega_n T = 2\pi$. The period is

$$T = 2\pi\sqrt{m / k} \tag{6.11}$$

and the natural frequency of vibration is the reciprocal of the period

$$f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}} \tag{6.12}$$

The natural frequency of an undamped spring mass system can be specified in terms of the deflection δ of the spring, under the load of the mass since $k\delta$ = force = m g, where g is the acceleration due to gravity. Using $g = 386$ in/s^2 = 9.8 m/s^2, the natural frequency can be written as

$$f_n = \frac{3.13}{\sqrt{\delta_i}} = \frac{5}{\sqrt{\delta_{cm}}} \tag{6.13}$$

where f_n = natural frequency of the system, (Hz)
δ = deflection of the spring under the weight of the mass $-$ δ_i in inches and δ_{cm} in centimeters

This provides a convenient way of remembering the natural frequency of a spring mass system. A one-inch deflection spring is a 3-Hz oscillator, and a one-centimeter deflection spring is a 5-Hz oscillator. These simple oscillators appear over and over in various forms throughout architectural acoustics.

Air Spring Oscillators

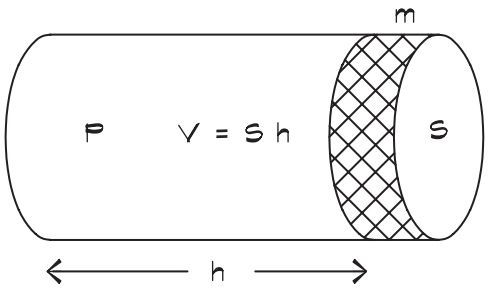
When air is contained in an enclosed space, it can act as the spring in a spring mass system. In the example shown in Fig. 6.2, a frictionless mass is backed by a volume of air. When the mass moves into the volume, the pressure increases, creating a force that opposes the motion. The spring constant of the enclosed air is derived from the equation of state

$$P V^{\gamma} = \text{constant} \tag{6.14}$$

differentiating

$$\gamma P V^{\gamma - 1}\, dV + V^{\gamma}\, dP = 0 \tag{6.15}$$

FIGURE 6.2 A Frictionless Mass and an Air Spring



The rate of change of pressure with volume is

$$dP = -\frac{\gamma P}{V} dV \tag{6.16}$$

from which we can obtain the spring constant for a trapped air volume of depth h

$$k = \frac{\gamma\ P\ S}{V}\ \frac{dV}{dh} = \frac{\gamma\ P\ S^2}{V} = \frac{\gamma\ P\ S}{h} \tag{6.17}$$

and the natural frequency is

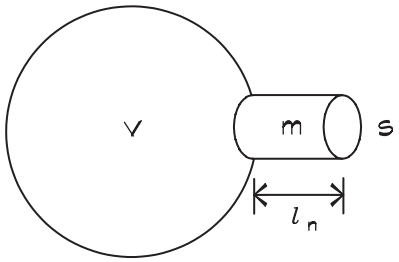
$$f_n = \frac{1}{2\,\pi}\ \sqrt{\frac{\gamma\ P_0\ S}{m\ h}} \tag{6.18}$$

where f_n = resonant frequency of the system, (Hz)
P_0 = atmospheric pressure, (1.013×10^5 Pa)
m = mass of the piston, (kg)
γ = ratio of specific heats = 1.4 for air
S = area of the piston (m^2)
h = depth of the air cavity (m)

For normal atmospheric pressure, the resonant frequency of an unsupported panel of drywall, 16 mm thick (5/8") weighing 12.2 kg/sq m (2.5 lbs/sq ft) with a 9 cm (3.5") air cavity behind it, is about 18 Hz. If the air gap is reduced to 1.2 cm (about 1/2"), the natural frequency rises to about 50 Hz. If the drywall panel were supported on resilient channel having a deflection due to the weight of the drywall of 1 mm, its spring mass resonant frequency would be about 16 Hz. The presence of the air spring caused by the 1/2" air gap stiffens the connection between the drywall and the support system, which results in poor vibration isolation when resilient channel is applied directly over another panel. A similar condition occurs in the construction of resiliently mounted (floating) floors if the trapped air layer is very shallow.

The air spring was the basis behind the air-suspension loudspeaker system, which was developed in the 1960s. The idea was that a cone loudspeaker could be made lighter and more compliant (less stiff) if the spring constant of the air in the box containing it made up for the lack of suspension stiffness. The lighter cone made the loudspeaker easier to accelerate and thus improved its high-frequency response.

FIGURE 6.3 Geometry of a Helmholtz Resonator



Helmholtz Resonators

A special type of air-spring oscillator is an enclosed volume having a small neck and an opening at one end. It is called a *Helmholtz resonator*, named in honor of the man who first calculated its resonant frequency. Referring to the dimensions in Fig. 6.3, the system mass is the mass of the air in the neck, $m = \rho_0 S l_n$ and the spring constant k is $\gamma P_0 S^2 / V$. Using the relationship shown in Eq. 6.12 the natural frequency is given by

$$f_n = \frac{c_0}{2\pi} \sqrt{\frac{S}{V l_n}} \tag{6.19}$$

Note that the frequency increases with neck area and decreases with enclosed volume and neck length.

Helmholtz resonators are used in bass-reflex or ported loudspeaker cabinets to extend the bass response of loudspeakers by tuning the box so that the port emits energy at low frequencies. The box-cone combination is not a simple Helmholtz resonator, but acts like a high-pass filter. Thiele (1971) and Small (1973) did extensive work on developing equivalent circuit models of the combined system, which is designed so that its resonant frequency is just below the point at which the loudspeaker starts to loose efficiency. An example of a low-frequency loudspeaker response curve is shown in Fig. 6.4. Ported boxes radiate low-frequency sound out the opening, where it combines in phase with the direct cone radiation. Potentially detrimental high-frequency modes, due to resonances within the cavity, are dampened by filling the box with fiberglass.

A similar technique is used to create tuned absorbers in studios and control rooms, where they are called bass traps. Relatively large volumes, filled with absorption, are tucked into ceilings, under platforms, and behind walls to absorb low-frequency sound. To act as a true Helmholtz resonator they must have a volume, a neck, and an opening, whose dimensions are small compared with the wavelength of sound to be absorbed.

Neckless Helmholtz Resonators

When an enclosed volume has an opening in it, it can still oscillate as a Helmholtz resonator even though it does not have an obvious neck. Since the air constricts as it passes through the opening, it forms a virtual tube, illustrated in Fig. 6.5, sometimes called a *vena contrata* in fluid dynamics. This acts like a neck, with an effective length equal to the thickness of the plate l_0 plus an additional "end effect" factor of (0.85 a), where a is the radius of the opening.

FIGURE 6.4 Frequency Response of a Sealed Cabinet vs a Bass-Reflex Cabinet (Roozen et al., 1998)

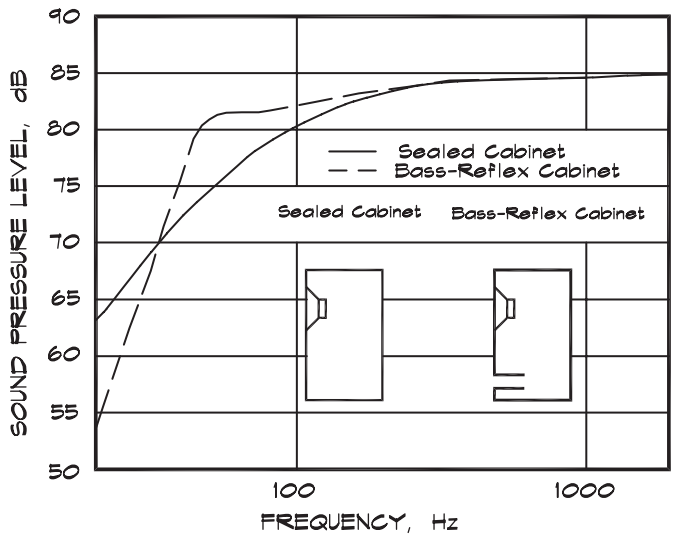
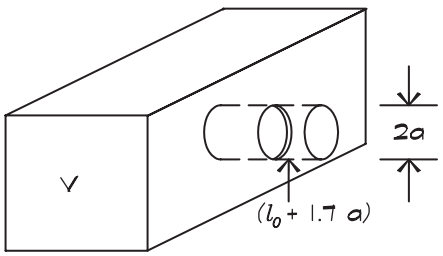


FIGURE 6.5 Geometry of a Neckless Helmholtz Resonator



Since there are two open ends of the tube, the natural frequency becomes

$$f_n = \frac{c_0}{2\pi}\sqrt{\frac{\pi a^2}{V\,(l_0 + 1.7\,a)}} \tag{6.20}$$

When there are multiple openings in the side of an enclosure, such as with a perforated plate spaced out from a wall, a Helmholtz resonance effect can still occur. The length of the neck is still the same as that used in Eq. 6.20, but the open area is the area of each hole times the number of holes in the plate.

For a perforated panel located a distance d from a solid wall, the resonant frequency is

$$f_n = \frac{c_0}{2\pi}\sqrt{\frac{\sigma}{d\,(l_0 + 1.7\,a)}} \tag{6.21}$$

where σ = fraction of open area in the panel which for round holes, staggered is 0.9 $(2a/b)^2$ and for round holes, straight is 0.785 $(2a/b)^2$, where b is the hole spacing
d = depth of the airspace behind the panel in units consistent with those of l_0, a, and c_0

6.2 WAVE EQUATION

One-Dimensional Wave Equation

The wave equation is a differential equation that formally defines the behavior in space and time of the pressure, density, and other variables in a sound wave. It is rarely used directly in architectural acoustics, but its solutions are the basis of wave acoustics, which is important to the understanding of many phenomena. Its derivation includes several assumptions about the nature of the medium through which sound passes. It assumes that the conducting medium follows the equation of continuity (conservation of mass), Newton's second law of motion, and an equation of state, relating the pressure and density. Refer to Kinsler et al., (1982) for a more detailed treatment.

To derive the wave equation we examine a small slice of a fluid (such as air), having thickness dx, shown from one side in Fig. 6.6. As a sound wave passes by, the original dimensions of the box ABCD move, in one dimension, to some new position A'B'C'D'. If S is the area of the slice, having its normal along the x axis, then there is a new box volume

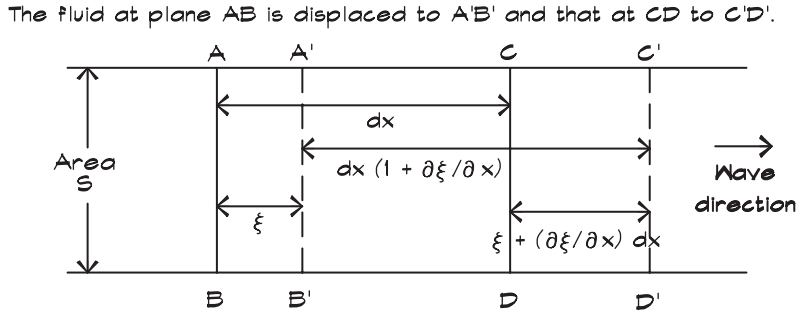
$$V + dV = S\,dx\left(1 + \frac{\partial \xi}{\partial x}\right) \tag{6.22}$$

where ξ is the displacement of the slice. In Chapt. 2, the bulk modulus was defined in Eq. 2.39 as the ratio of the change in pressure to the fractional change in volume

$$dP = -B\,\frac{dV}{V} \tag{6.23}$$

In terms of the volume in Fig. 6.6, the change in the total pressure P is the acoustic pressure p and the change in volume is S dx. So comparing Eqs. 6.22 and 6.23, we obtain the relation

FIGURE 6.6 The Fluid Displacement during the Passage of a Plane Wave (Rossing and Fletcher, 1995)



between the acoustic pressure and the change in length. The minus sign means that if the length decreases, the pressure increases.

$$p = - B \frac{\partial \xi}{\partial x} \tag{6.24}$$

The motion of the slice is described by Newton's second law, so we set the pressure gradient force equal to the slice mass times its acceleration,

$$- S \left(\frac{\partial p}{\partial x} d x \right) = \rho_0 S dx \frac{\partial^2 \xi}{\partial t^2} \tag{6.25}$$

and simplifying we obtain an expression known as Euler's equation

$$- \frac{\partial p}{\partial x} = \rho_0 \frac{\partial^2 \xi}{\partial t^2} \tag{6.26}$$

Then using Eqs. 6.24 and 6.26

$$\frac{\partial^2 \xi}{\partial t^2} = \frac{B}{\rho_0} \frac{\partial^2 \xi}{\partial x^2} \tag{6.27}$$

By differentiating Eq. 6.26 once with respect to x, and Eq. 6.24 twice with respect to t, and adding them

$$\frac{\partial^2 p}{\partial t^2} = c^2 \frac{\partial^2 p}{\partial x^2} \tag{6.28}$$

where we have used Eq. 2.40 for the speed of sound. This is the one-dimensional wave equation, expressed in terms of pressure. It relates the spatial and time dependence of the sound pressure within the wave.

Solutions to Eq. 6.28 are not difficult to find; in fact, any pressure wave that is a function of the quantity $(x \pm c\, t)$ will do. This reveals the infinite number of possible waveforms that a sound wave can take. The plus and minus signs indicate the direction of propagation of the wave. Although many functions are solutions to the wave equation, not all are periodic. In architectural acoustics, the periodic solutions are of primary interest to us, although nonperiodic phenomena such as sonic booms are occasionally encountered. A periodic solution, which describes a plane wave traveling in the $+x$ direction is

$$\mathbf{p} = \mathbf{A}\, e^{-j k x}\, e^{j \omega t} = \mathbf{A}\, \cos(-k x + \omega t) \tag{6.29}$$

where $\mathbf{p}$ = acoustic pressure, (Pa)
$\mathbf{A}$ = maximum pressure amplitude, (Pa)
$j = \sqrt{-1}$
ω = radial frequency, (rad / s)
k = wave number = ω / c, (rad / m)

The right side of Eq. 6.29 is a familiar form, which we derived using heuristic arguments in Eq. 2.32. The terms on the left are another way of expressing periodic motion using exponentials, which are more convenient for many calculations.

Using Eq. 6.26, the particle velocity can be obtained

$$-\frac{\partial \mathrm{p}}{\partial \mathrm{x}} = \rho_0 \frac{\partial \mathrm{u}}{\partial \mathrm{t}} \tag{6.30}$$

Assuming that u has the same form as Eq. 6.29, the time derivative can be replaced by $\mathrm{j}\,\omega$ or $\mathrm{j\,c\,k}$

$$\mathbf{u} = \frac{\mathrm{j}}{\mathrm{k}\,\rho_0\,\mathrm{c}} \frac{\partial \mathbf{p}}{\partial \mathrm{x}} \tag{6.31}$$

Three-Dimensional Wave Equation

When dealing with sound waves in three dimensions we have a choice of several coordinate systems. Depending on the nature of the problem, one system may be more appropriate than another. In a rectilinear (x, y, z) system, we can write separate equations in each direction similar to Eq. 6.28 and combine them to obtain

$$\frac{\partial^2 \mathrm{p}}{\partial\, \mathrm{t}^2} = \mathrm{c}^2 \left[\frac{\partial^2 \mathrm{p}}{\partial\, \mathrm{x}^2} + \frac{\partial^2 \mathrm{p}}{\partial\, \mathrm{y}^2} + \frac{\partial^2 \mathrm{p}}{\partial\, \mathrm{z}^2} \right] = \mathrm{c}^2\, \nabla^2 \mathrm{p} \tag{6.32}$$

The term in the bracket in Eq. 6.32 is called the Laplace operator and is given the symbol ∇^2.

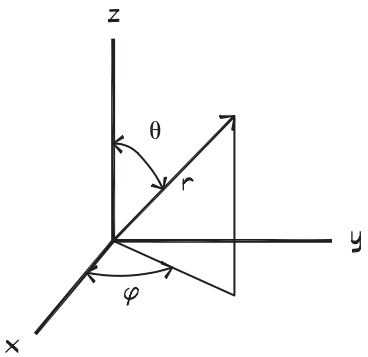
In the spherical coordinate system in Fig. 6.7, there are two angular coordinates, usually designated θ and ϕ, and one radial coordinate designated r. The Laplace operator in spherical coordinates is

$$\nabla^2 = \frac{1}{\mathrm{r}^2} \frac{\partial}{\partial\, \mathrm{r}} \left(\mathrm{r}^2 \frac{\partial}{\partial\, \mathrm{r}} \right) + \frac{1}{\mathrm{r}^2\, \sin\, \theta} \frac{\partial}{\partial\, \theta} \left(\sin\, \theta \frac{\partial}{\partial\, \theta} \right) + \frac{1}{\mathrm{r}^2\, \sin^2\, \phi} \frac{\partial^2}{\partial\, \phi^2} \tag{6.33}$$

For a nondirectional source we can dispense with consideration of the angular coordinates and examine only the dependence on r. This yields the one-dimensional wave equation for a spherical wave

$$\frac{\partial^2 \mathrm{p}}{\partial\, \mathrm{t}^2} = \mathrm{c}^2 \left[\frac{1}{\mathrm{r}^2} \frac{\partial}{\partial \mathrm{r}} \left(\mathrm{r}^2 \frac{\partial \mathrm{p}}{\partial\, \mathrm{r}} \right) \right] \tag{6.34}$$

FIGURE 6.7 The Spherical Coordinate System



6.3 SIMPLE SOURCES

Monopole Sources

The general solution to Eq. 6.34 for a wave moving in the positive r direction is

$$\mathbf{p} = \frac{\mathbf{A}}{\mathrm{r}} \, \mathrm{e}^{-\mathrm{j}\,\mathrm{k}\,\mathrm{r}} \, \mathrm{e}^{\mathrm{j}\omega\mathrm{t}} \tag{6.35}$$

Using Eq. 6.31, we can solve for the particle velocity

$$\mathbf{u} = \frac{\mathbf{A}}{\mathrm{r}\,\rho_0\,\mathrm{c}_0} \left(1 + \frac{1}{\mathrm{j}\,\mathrm{k}\,\mathrm{r}}\right) \mathrm{e}^{-\mathrm{j}\,\mathrm{k}\,\mathrm{r}} \, \mathrm{e}^{\mathrm{j}\omega\mathrm{t}} \tag{6.36}$$

When we are far away from the source and k r >> 1, the particle velocity reverts to its plane wave value, $\mathbf{p} / \rho_0 \, \mathrm{c}_0$. Equations 6.35 and 6.36 describe the behavior of a simple point source, sometimes called a monopole.

Doublet Sources

When two point sources are placed at x = 0 and x = d, they form an acoustic doublet. The sources may be in phase or out of phase. For this analysis, they are assumed to be radiating at the same source strength and frequency. Figure 6.8 shows the geometry for this configuration. When the receiver is close to the source the geometrical relationships are a bit complicated. If the receiver is far away, we can make the approximation that the lines between the sources and the receiver are almost parallel. Under these constraints, the pressure for the two sources combined is

$$\mathbf{p} = \frac{\mathbf{A}}{\mathrm{r}} \, \mathrm{e}^{\mathrm{j}\omega\mathrm{t}} \, \mathrm{e}^{-\mathrm{j}\,\mathrm{k}\,\mathrm{r}} \left(1 \pm \mathrm{e}^{\mathrm{j}\,\mathrm{k}\,\mathrm{d}\,\sin\,\theta}\right) \tag{6.37}$$

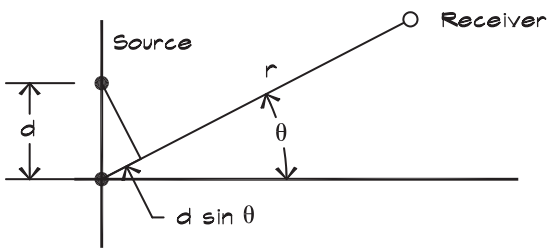
where the plus sign is for in-phase sources and the minus sign is for out-of-phase sources. The value of the pressure for in-phase sources is

$$\mathrm{p} = \frac{2\,\mathrm{A}}{\mathrm{r}} \cos\left(\frac{1}{2}\,\mathrm{k}\,\mathrm{d}\,\sin\,\theta\right) \tag{6.38}$$

and for out-of-phase sources is

$$\mathrm{p} = \frac{2\,\mathrm{A}}{\mathrm{r}} \sin\left(\frac{1}{2}\,\mathrm{k}\,\mathrm{d}\,\sin\,\theta\right) \tag{6.39}$$

FIGURE 6.8 The Geometry of a Doublet Source



The total power radiated by the doublet can be calculated by integrating the square of the pressure over all angles

$$W = \frac{1}{2} \iint \left(\frac{p^2}{\rho_0 \, c_0} \right) r^2 \, \sin \theta \, d\theta \, d\phi \tag{6.40}$$

to obtain

$$W = \frac{A^2}{\rho_0 \, c_0} \left[1 \pm \frac{\sin k \, d}{k \, d} \right] \tag{6.41}$$

where the plus sign refers to an in-phase doublet and the minus sign to an out-of-phase doublet.

Now there is a great deal of important information in these few equations. Plotting Eq. 6.41 in Fig. 6.9, we can examine the relative power of a doublet compared to a simple source. The top curve shows the power of an in-phase pair as a function of k d. When the two sources are close (compared with a wavelength) together, k d is less than 1.

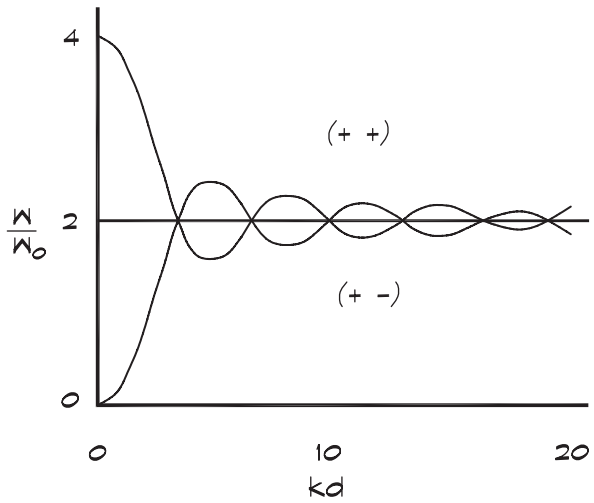
Recall that the wave number

$$k = \omega / c = 2 \pi / \lambda \tag{6.42}$$

Therefore, when k d is less than 1, the separation distance d between sources is less than a sixth of a wavelength. For this configuration, the acoustic pressure effectively doubles, the combined source power increases by a factor of four, and the sound power level increases by 6 dB.

FIGURE 6.9 Total Power Radiated from a Doublet Source (Rossing and Fletcher, 1995)

The total power W of two omnidirectional sources having the same (+ +) or the opposite (+ −) polarity and separation distance d, as a function of the frequency parameter kd. The power radiated by a single source is W_0.



One way this can occur in buildings is when a source is placed close to a hard surface such as a concrete floor or wall. The surface acts like an acoustic mirror and the original source energy is reflected, as if the source were displaced by a distance d/2 behind the surface of the mirror. If the distances are small enough and the frequency low enough, the pressure radiates from the original and reflected source in phase in all directions. When sound level measurements are made very close to a reflecting surface, a 6 dB increase in level can be expected due to pressure doubling.

As the distance d between each source increases, the angular patterns become more complicated. The radiated power for a doublet at higher values of k d, also shown in Fig. 6.9, is about twice the power of a single source whether or not the sources are in phase. This is the same result that we found for incoherent (random phase) sources—a 3 dB (10 log N) increase for two sources combined.

Although the overall power of a doublet has a relatively simple behavior, the directivity pattern is more complicated. Typically, these directivity patterns are displayed in the form of polar plots. The front of the source, which is usually the loudest direction, is shown pointing toward the top of the diagram and the decrease in level with angle is plotted in increments around the source. A uniform directivity pattern is a perfect circle. The *directional characteristic* R_θ is one commonly encountered descriptor. It is the terms in the parentheses in Eq. 6.37 and represents the directional pattern of the sound pressure for a doublet. The directivity is its square and the directivity index is 10 log of that. The directional characteristic pattern produced by an in-phase doublet at various frequencies is shown in Fig. 6.10. Note that the direction of greatest level, when θ is 0, is at right angles to the line between the sources. The directivity patterns vary with frequency. The half-beamwidth angle, which is defined as the angle between on axis and the first zero, occurs when $(\mathrm{k\ d}/2)\sin\theta = \pi/2$. The half beamwidth for an in-phase doublet is

$$\psi = \sin^{-1}(\lambda/2\,\mathrm{d}) \tag{6.43}$$

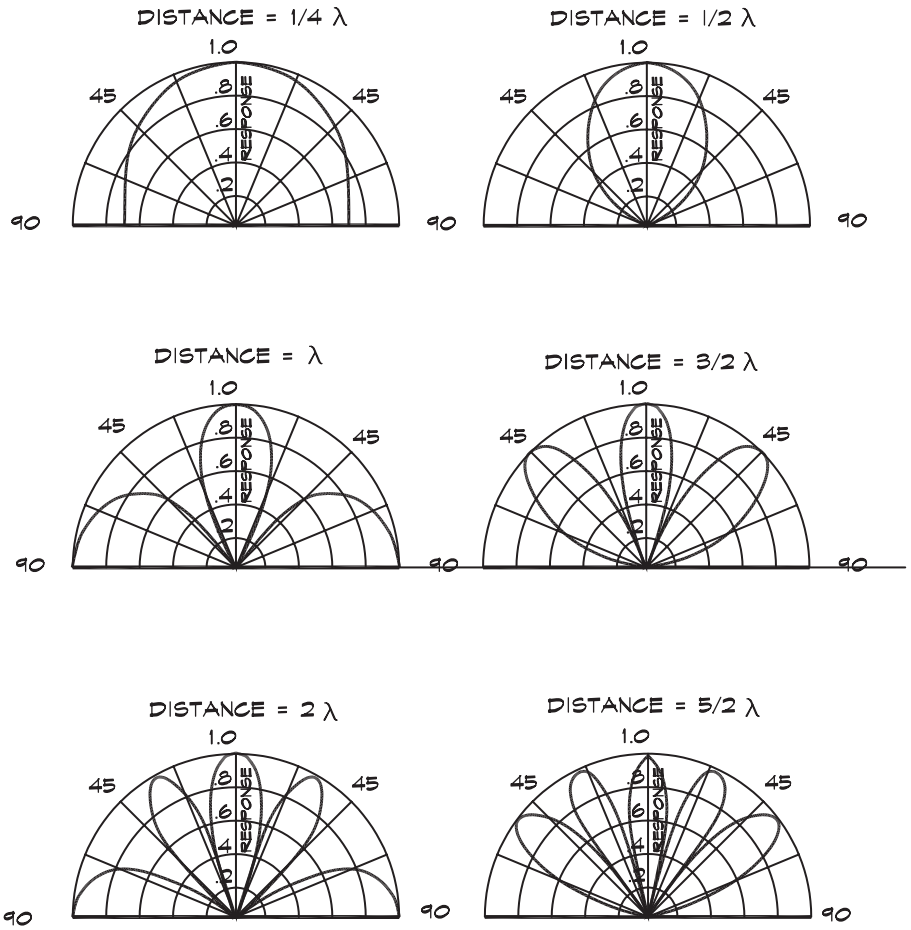
Dipole Sources and Noise Cancellation

If we have a doublet, where the two sources have opposite polarities, the configuration is called a dipole. A practical example of this type of source is an unbaffled loudspeaker. Since sound radiates from the rear of the loudspeaker cone as well as from the front, and the two signals are out of phase, the signal from the rear can combine with the front signal and produce a null pattern at right angles to the axis of the cone.

If the dipole sources are close together, when $\mathrm{k\ d} < 1$, from Fig. 6.9 we see that the total power radiated approaches zero. This is the basis for the field of active noise cancellation in a three-dimensional acoustic space. Two sources of opposite polarity, when positioned close enough together, radiate a combined null signal. In practice, this can be accomplished by generating a cancellation signal quite close (d less than $\lambda/6$) to the source or to the receiver. A microphone can be used to sense the primary noise signal and by appropriate processing a similar signal having the opposite phase can be produced. Active noise cancellation systems are available in the form of headphones, which suppress sounds having a frequency of less than about 300–400 Hz. Source-cancellation systems for environmental noise control are less common, but have been applied successfully to large transformers and exhaust stacks. Noise cancellation systems have not been applied to the general run of noise problems, due

FIGURE 6.10 Directional Characteristics in Terms of R_θ of an In-Phase Doublet Source as a Function of the Distance between the Sources and the Wavelength (Olson, 1957)

The direction corresponding to the angle θ is measured relative to the perpendicular to the line connecting the two sources. The three dimensional polar plots are surfaces of revolution about the line joining the two sources.

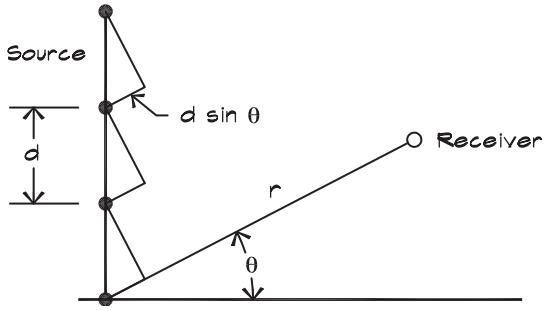


to the distance requirements outlined earlier, although one-dimensional problems such as duct-borne noise have been treated successfully. Due to the one-dimensional nature of ducts at low frequencies, the distance requirements for active noise control are not the same as they are for three-dimensional spaces.

Arrays of Simple Sources

When several simple sources are arrayed in a line, the phase relationship between them increases the directivity of the group along the $\theta = 0$ axis, shown in Fig. 6.11. Loudspeaker systems so configured are called line arrays and are a common arrangement

FIGURE 6.11 The Geometry of a Line Array



in loudspeaker cluster design. If n in-phase sources are equally spaced along a line, we can calculate the pressure in the far field following the same reasoning we used to derive Eq. 6.37. The pressure is

$$\mathbf{p} = \frac{\mathrm{n}\,\mathbf{A}\,\mathrm{e}^{\mathrm{j}\omega\mathrm{t}}\,\mathrm{e}^{-\mathrm{j}\mathrm{k}\mathrm{r}}}{\mathrm{r}}\left[\frac{1}{\mathrm{n}}\sum_{\mathrm{m}=0}^{\mathrm{n}-1}\mathrm{e}^{\mathrm{j}\,\mathrm{k}\,\mathrm{m}\mathrm{d}\,\sin\,\theta}\right] \tag{6.44}$$

and the summation can be expressed in terms of trigonometric functions

$$\mathbf{p} = \frac{\mathrm{n}\,\mathbf{A}\,\mathrm{e}^{\mathrm{j}\omega\mathrm{t}}\,\mathrm{e}^{-\mathrm{j}\mathrm{k}\mathrm{r}}}{\mathrm{r}}\left[\frac{\sin\left(\dfrac{\mathrm{n}\,\pi\,\mathrm{d}}{\lambda}\,\sin\,\theta\right)}{\mathrm{n}\,\sin\left[\dfrac{\pi\,\mathrm{d}}{\lambda}\,\sin\,\theta\right]}\right] \tag{6.45}$$

The term on the left of the brackets in Eq. 6.45 is the source strength for all n simple sources. At $\theta = 0$ the bracketed term goes to one so that the overall source strength is the same, as we would expect from a coherent group.

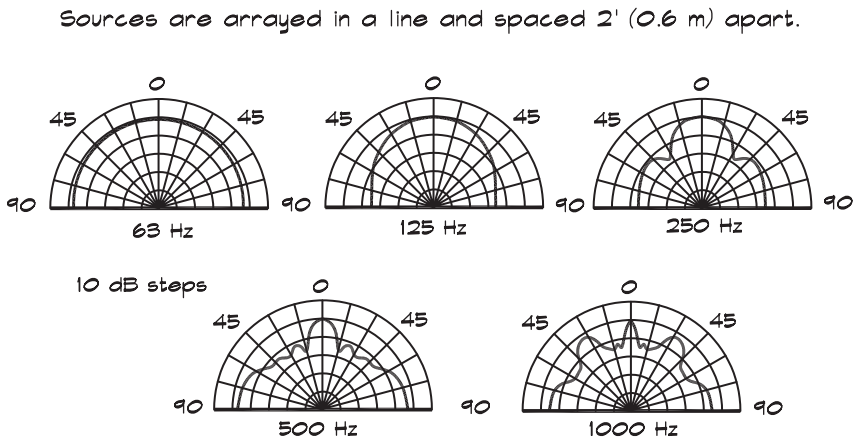
The polar patterns in terms of the directivity index for an array of four omnidirectional sources are shown in Fig. 6.12. The spacing is 2 ft (0.6 m) between sources. The angle to the point where the sound level is down 6 dB from the on-axis value occurs when the bracketed term in Eq. 6.45 equals 1/2, which must be solved numerically.

The half-beamwidth angle is more easily determined

$$\psi = \sin^{-1}\left[\frac{2\,\pi}{\mathrm{n}\,\mathrm{k}\,\mathrm{d}}\right] = \sin^{-1}\left[\frac{\lambda}{\mathrm{n}\,\mathrm{d}}\right] \tag{6.46}$$

so that, when the total length of the line array is about 1.4 wavelengths, the first zero is at about $\pi/4$ radians (45°). Hence, to begin to achieve appreciable control over the beamwidth, line arrays must be at least 1.4 wavelengths long. For narrower coverage angles, the length must be longer. The region of pattern control, without undue lobing, ranges from an overall array length of about 1.4 wavelengths to the point where the spacing between elements approaches a wavelength.

FIGURE 6.12 Directivity Index of Four Point Sources



This has profound architectural consequences because it says that loudspeaker systems must be large in the vertical direction to control directivity in a vertical plane. For example, to limit the beamwidth angle to $\pi/3$ radians (60°) at 500 Hz, the array should be about 750 mm (30 inches) high. In cabinet systems the horn, which emits the 500 Hz signal, must be about 30 inches high to achieve a 40° vertical coverage angle. (Note that the coverage angle, which is the angle between −6 dB points, is less than the line array beamwidth.) Architecturally this means that in large rooms such as churches and auditoria, which often require highly directional loudspeaker systems to achieve adequate intelligibility, a space at least 4 feet (1.2 m) high must be provided for a speech reinforcement system. If live music is going to be miked, the directivity should extend down an octave lower to reduce feedback. This requires a line array about 8 feet long, similar to that shown in Fig. 6.12 in a concert venue, or significant loudspeaker displacement or barrier shielding in a permanent installation.

Line source configurations can also be used to control low-frequency directivity in concert systems. When concert loudspeaker systems are arranged by unloading truckloads of multiway cabinets and stacking them up on either side of the stage, there is little control of the low-frequency energy and extremely loud sound levels are generated near the front of the stage and at the performers. If instead, low-frequency cabinets (usually dual 18-inch woofers) are stacked vertically to a height of 20 to 30 feet (6 to 9 m), a line source is constructed that controls bass levels at the stage apron. The front row seats are at an angle of nearly 90° to the midpoint axis of the line source so even coverage is maintained from the front to the back of the seating area.

Continuous Line Arrays

The continuous line array is a convenient mathematical construct for modeling rows of sources that are all radiating in phase. Line arrays have a relatively narrow frequency range over which they maintain a simple directivity pattern. If their length is less than a half-wavelength, they will not provide appreciable directional control. At high frequencies line sources have a very narrow beamwidth, so that off-axis there can be a coloration of the sound.

The directional characteristic of a coherent line source can be obtained (Olson, 1957) by substituting $l \cong n\ d$ into Eq. 6.45. This approximation is true for large n.

$$R_\theta = \frac{\sin\left(\frac{\pi l}{\lambda}\sin\theta\right)}{\frac{\pi l}{\lambda}\sin\theta} \tag{6.47}$$

where R_θ is the directional characteristic of the sound pressure relative to the on-axis sound pressure, and l is the length of the line source.

The directivity plots are shown in Fig. 6.13. Practical considerations limit the size of a loudspeaker array to an overall length of about λ to 4 λ or so. This two-octave span is adequate for many sound source applications, where horns are used on the high end and line arrays are used for the midrange. Directional control is seldom required below the 250 Hz octave band except in concert venues.

As sources are added to a line array, the beamwidth decreases and the number of lobes increases. To compensate for this effect, a line source can be tapered by decreasing the level of the signal fed to loudspeakers farther from the center. The directional characteristic (Olson, 1957) of a tapered line source, whose signal strength varies linearly from the center to zero at the ends is

$$R_\theta = \frac{\sin^2\left(\frac{\pi l}{\lambda}\sin\theta\right)}{\left(\frac{\pi l}{\lambda}\sin\theta\right)^2} \tag{6.48}$$

As Fig. 6.14 shows, tapering broadens the center lobe of the directivity pattern and decreases the off-axis lobing and the expense of overall sound power.

Curved Arrays

Loudspeakers can be configured in other ways, including convex or concave arcs, twisted line arrays, or helical line sources, which look like a stack of popsicle sticks. For a series of sources arranged in a curve the directivity pattern in the plane of the arc is (Olson, 1957)

$$R_\theta = \frac{1}{2n+1}\left\{\sum_{m=-n}^{m=n}\cos\left[\frac{2\pi r_a}{\lambda}\cos(\theta + m\phi)\right] + j\sum_{m=-n}^{m=n}\sin\left[\frac{2\pi r_a}{\lambda}\cos(\theta + m\phi)\right]\right\} \tag{6.49}$$

where R_θ = directional characteristic of the array sound pressure relative to the on axis sound pressure
θ = angle between the radius to the center source and the line to the receiver, (rad)

$2n + 1$ = number of sources in the array
$j = \sqrt{-1}$
λ = wavelength, (m)
m = integer variable
ϕ = angle subtended by adjacent sources on the arc, (rad)

FIGURE 6.13 Directional Characteristics of a Coherent Line Source (Olson, 1957)

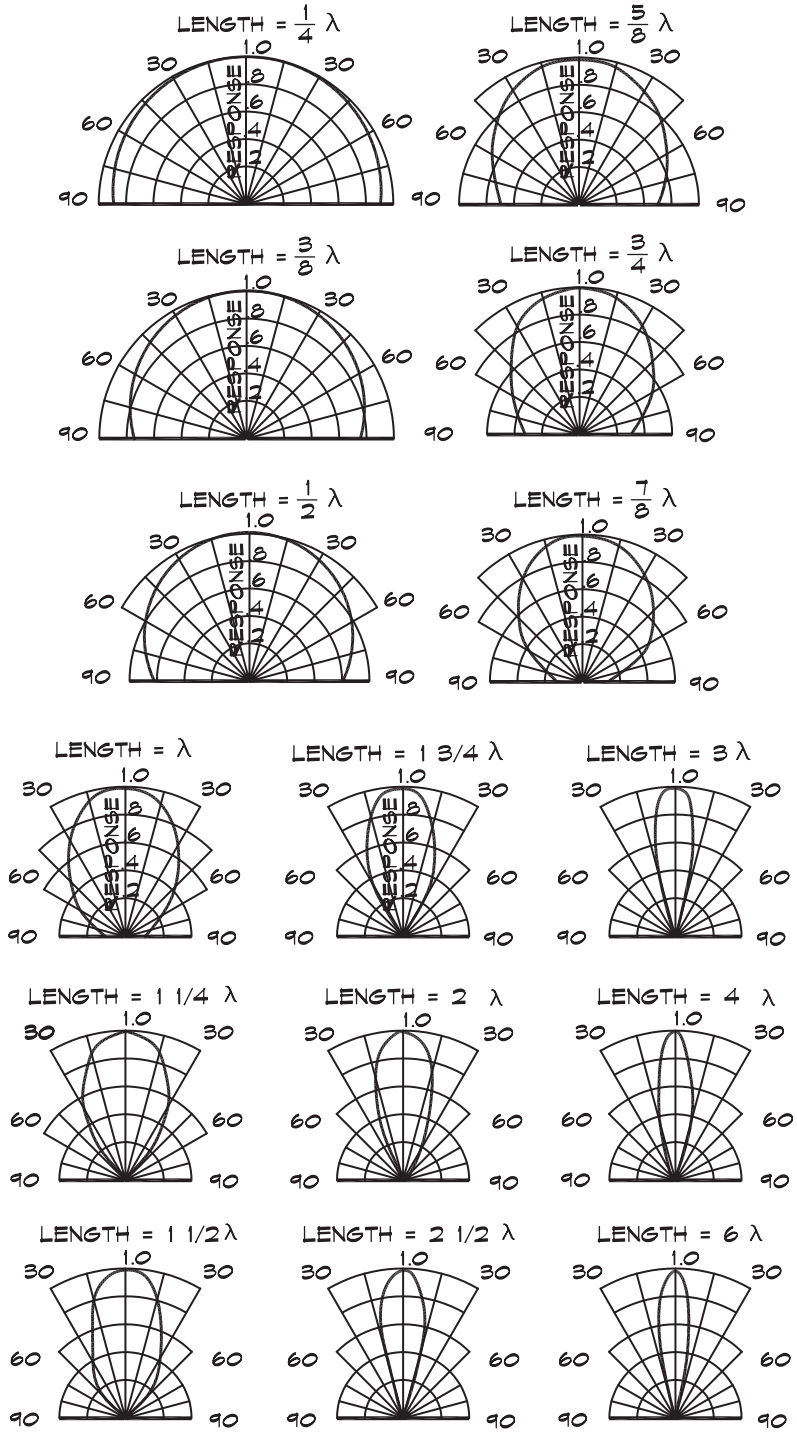


FIGURE 6.14 Directional Characteristics R_θ of a Tapered Line Source—Linear Taper (Olson, 1957)

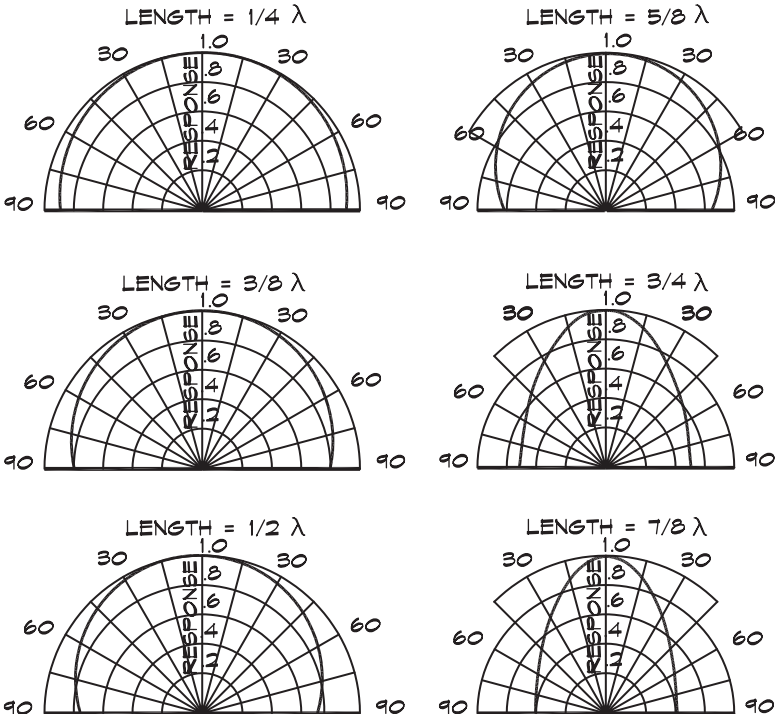
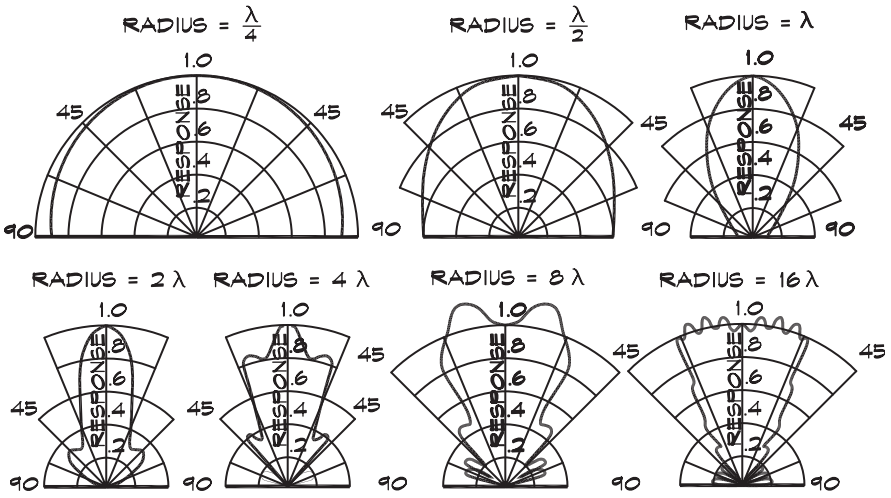


FIGURE 6.15 Directional Characteristics R_θ of a 60° Segment of Arc (Olson, 1957)



An example of the directivity pattern for a 60° arc is shown in Fig. 6.15. At very high frequencies, the directivity pattern starts to look like a wedge. This behavior was the basis for the design of multicellular horns, which were developed to reduce the high-frequency beaming associated with large horn mouths.

Phased Arrays

A phased array consists of a line or planar group of sound sources, which are fed an electronic signal such that the direction of the emitted wavefront can be steered by controlling the phase or time delay (n τ) to each source. The directional factor becomes (Kinsler et al., 1982)

$$R_\theta = \frac{1}{n}\left[\frac{\sin\left[\frac{n\pi d}{\lambda}\left(\sin\theta - \frac{c\tau}{d}\right)\right]}{\sin\left[\left(\frac{\pi d}{\lambda}\sin\theta - \frac{c\tau}{d}\right)\right]}\right] \tag{6.50}$$

This technique may be used to electronically direct signals radiated from a line or panel of sources. It also can be used to detect the direction of an incoming signal incident on a line of microphones by sensing the time delay between transducers. The major lobe is pointed in a direction given by

$$\sin\theta_0 = \frac{c\tau}{d} \tag{6.51}$$

which is independent of frequency.

Source Alignment and Comb Filtering

When two sources are separated by a distance d, and a receiver lies at an angle θ to their common axis, as shown in Fig. 6.8, there is a difference in distance from each source to the receiver. This difference is d $\sin\theta$ so depending on the frequency of the sound radiating from the doublet the signals may be in or out of phase. If the path length difference is an even multiple of a wavelength then the signals will add and the composite signal will be 6 dB higher. If the path length difference is an odd multiple of a half wavelength then the signals will cancel and the composite signal will be a null. The consequence of this is that as we sweep across a range of frequencies the doublet source will generate a series of filters whose maximum frequencies are given by

$$f_n = \frac{n\,c}{d\,\sin\theta} \tag{6.52}$$

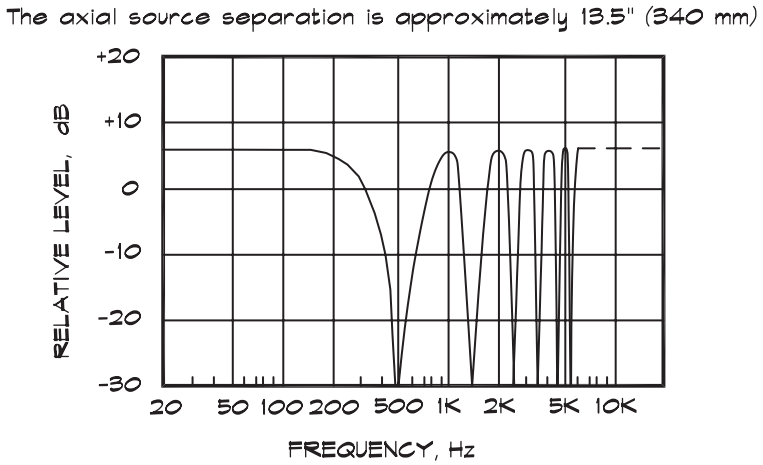
where the number n is an integer that ranges from one to infinity, or at least to the upper frequency limit of audibility. The null frequencies are given by

$$f_n = \frac{(2n-1)\,c}{2\,d\,\sin\theta} \tag{6.53}$$

At $\theta = \pi/4$ (45°) and a doublet separation distance of 0.5 m (1.6 ft), the null frequencies are 486, 1458, 2430, . . . Hz.

If a broadband signal such as speech is transmitted by means of the doublet source, the resultant signal shown in Fig. 6.16 displays a series of dips, which are shaped much like the teeth of a comb (hence the moniker, comb filter). Deep notches at each of these frequencies can have a negative influence on the clarity of the received signal. The extent of the influence depends on the separation between frequencies and the depth of the nulls. The effect can be corrected somewhat by introducing an electronic delay; however, the delay is exact for one direction only. A loudspeaker design strategy, which reduces the time difference between

FIGURE 6.16 The Comb Filter Produced by Two Sound Sources (Everest, 1994)

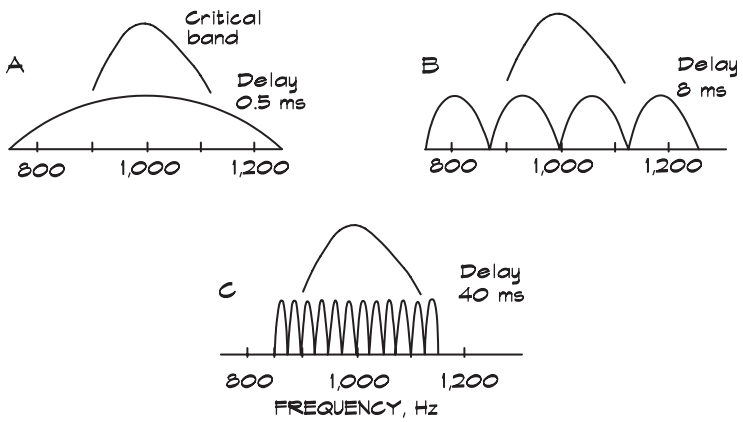


components, reduces the comb filtering effect. It is best to match the time delays at points where the two signals have nearly equal amplitudes. If one source is substantially (6 dB or so) louder than a second, the comb filtering effects are much less.

Comb Filtering and Critical Bands

Everest (1994) provides an interesting analysis of the audibility of comb filtering, which is illustrated in Fig. 6.17. The perceptual importance of comb filtering can be understood in terms of the effective bandwidth of the individual filter, compared to the width of a critical band. When the time delay between sources is small, say 0.5 ms, the range of frequencies between nulls is quite broad—much greater than the width of a critical band at 1000 Hz, which is about 128 Hz. Therefore, the effect of the delay is perceptible. When the time delay is large, say 40 ms, there are many nulls within one critical band, and the effects are integrated by the ear and are not perceptible. This helps explain why comb-filtering effects are not a problem in large auditoria, where reflections off a wall frequently create a delayed

FIGURE 6.17 Audibility of Comb Filtering (Everest, 1994)



signal of 40 ms or more, without appreciably coloring the sound. In loudspeaker clusters and small studios, small loudspeaker misalignments and reflections from nearby surfaces can be quite noticeable.

6.4 COHERENT PLANAR SOURCES

Piston in a Baffle

The physical effects of sound sources of finite extent are described using a few simple models, which can be applied to a wide range of more complicated objects. The most frequently utilized example is the piston source mounted in a baffle, which was first analyzed by Lord Rayleigh in the nineteenth century. The baffle in this example is an infinite solid wall. The piston may be a loudspeaker or simply a slice of air with all portions of the slice moving in phase at a velocity $\mathbf{u} = u_0\, e^{j\omega t}$. The piston is located at the origin, on the surface of the wall, and pointed along the r axis as shown in Fig. 6.18. The sound pressure at a distance r due to a small element of surface on the piston is

$$\mathbf{p}(r,\ \theta,\ t) = j\,\frac{\rho_0\, c_0\, u_0\, k}{2\,\pi}\int_s \frac{e^{j\,(\omega t - k\,r')}}{r'}\,ds \tag{6.54}$$

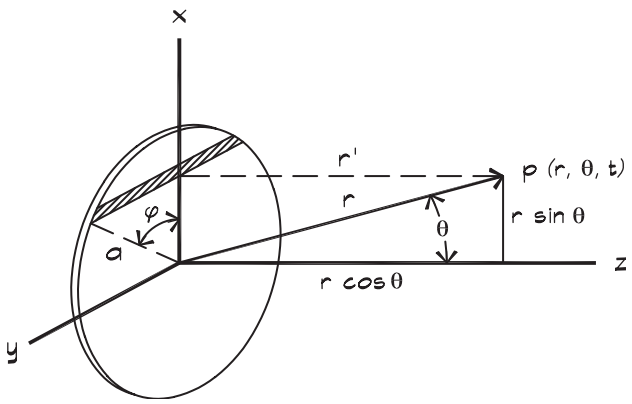
If we divide the surface into horizontal slices the incremental pressure due to a slice dx high by 2 a sin ϕ wide is

$$d\mathbf{p} = j\,\rho_0\, c_0 \frac{u_0}{\pi r'}\, k\, a\, \sin\,\phi\, e^{j\,(\omega t - k\,r')}\,dx \tag{6.55}$$

In the far field where $r >> a$ we use the approximate value for r'

$$r' \cong r\left(1 - \frac{a}{r}\,\sin\,\theta\,\cos\,\phi\right) \tag{6.56}$$

FIGURE 6.18 Geometry of a Piston in a Baffle (Kinsler et al., 1982)



The pressure is then calculated by integrating over the surface of the piston.

$$\mathbf{p} = \mathrm{j}\,\rho_0\,\mathrm{c}_0\,\frac{\mathrm{u}_0}{\pi\,\mathrm{r}}\,\mathrm{k\,a}\,\mathrm{e}^{\mathrm{j}\,(\omega\,\mathrm{t}\,-\,\mathrm{k\,r})}\int_{-\mathrm{a}}^{\mathrm{a}}\mathrm{e}^{\mathrm{j\,k\,a}\,\sin\theta\,\cos\phi}\,\sin\,\phi\,\mathrm{dx} \tag{6.57}$$

This can be done by making the substitution $\mathrm{x} = \mathrm{a}\,\cos\,\phi$ and integrating over the angle (see Kinsler et al., 1982).

$$\mathbf{p} = \frac{\mathrm{j}\,\rho_0\,\mathrm{c}\,\mathrm{u}_0\,\mathrm{k\,a}^2}{2\,\mathrm{r}}\,\mathrm{e}^{\mathrm{j}\,(\omega\,\mathrm{t}\,-\,\mathrm{k\,r})}\left[\frac{2\,\mathrm{J}_1\,(\mathrm{k\,a}\,\sin\,\theta)}{(\mathrm{k\,a}\,\sin\,\theta)}\right] \tag{6.58}$$

where θ = angle between the center axis and the line to the receiver, (rad)
J_1 = Bessel function of the first kind
$\mathrm{j} = \sqrt{-1}$
λ = wavelength, (m)
ω = radial frequency, (rad/s)
t = time, (s)

The Bessel function produces a number, which depends on an argument, similar to a trigonometric function. It is the solution to a particular type of differential equation and can be calculated from an infinite series of terms, which follow the pattern

$$\mathrm{J}_1\,(\mathrm{x}) = \frac{\mathrm{x}}{2} - \frac{2\,\mathrm{x}^3}{2\cdot 4^2} + \frac{3\,\mathrm{x}^5}{2\cdot 4^2\cdot 6^2} - \cdots \tag{6.59}$$

The resulting directional characteristic for a piston in a baffle is given in Fig. 6.19. The pattern in the far field depends on the ratio of the circumference of the piston to the wavelength of sound being radiated, which is the term (k a).

The directivity relative to the on-axis intensity is

$$\mathrm{Q}_{\mathrm{rel}} = \left[\frac{2\,\mathrm{J}_1\,(\mathrm{k\,a}\,\sin\,\theta)}{(\mathrm{k\,a}\,\sin\,\theta)}\right]^2 \tag{6.60}$$

and the beamwidth as defined by the angle between the -6 dB down points on each side occurs at

$$\mathrm{Q}_{\mathrm{rel}} = .25 \tag{6.61}$$

This defines (Long, 1983) the relationship between the coverage angle (between the -6 dB points) and the piston diameter as

$$\mathrm{k\,a}\,\sin\,\theta = 2.2 \tag{6.62}$$

When the piston diameter, 2 a, is equal to a wavelength the coverage angle $2\,\theta \cong 90°$. This is a useful rule of thumb in loudspeaker and horn design.

FIGURE 6.19 Directional Characteristics of a Circular Piston in a Baffle (Olson, 1957)

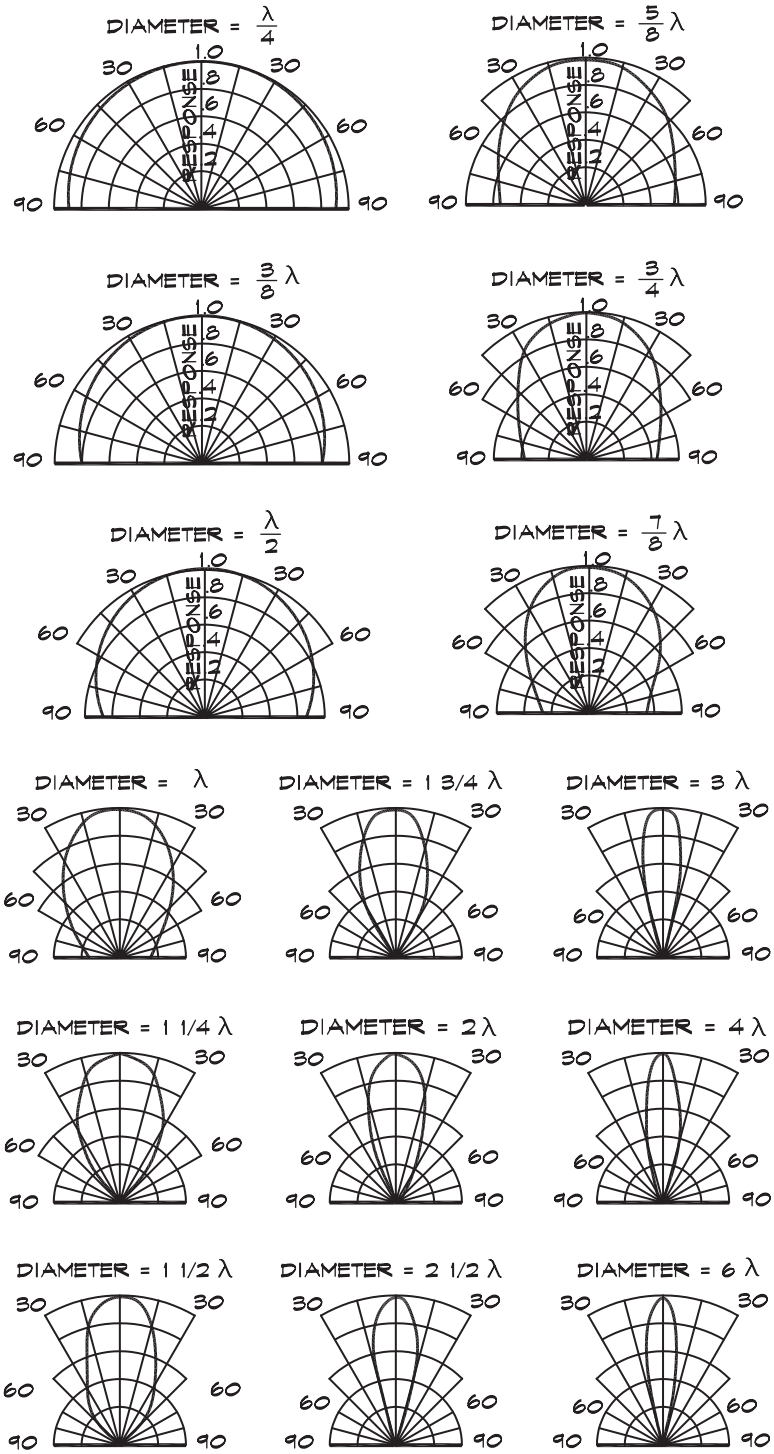
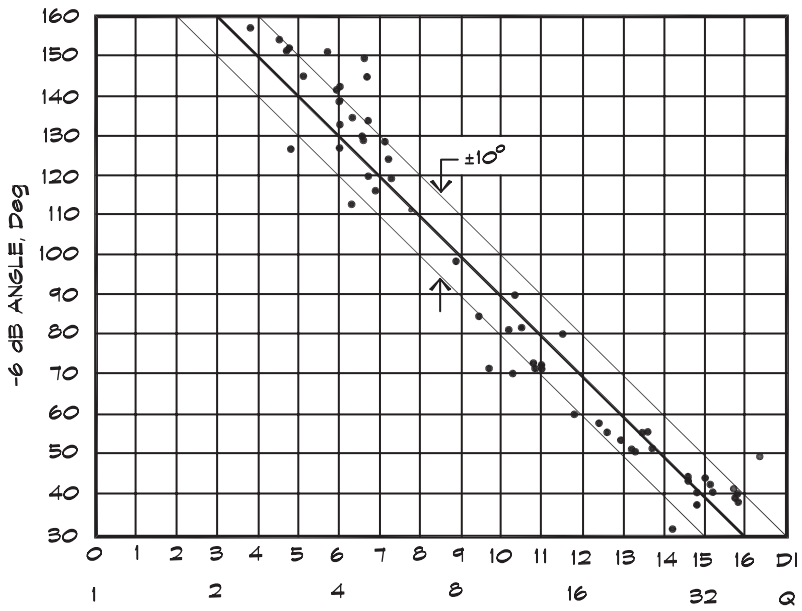


FIGURE 6.20 Relationship of Directivity Index to −6 dB Angle for Many Different Cone Type Loudspeakers and Baffles (Henricksen, 1980)



Coverage Angle and Directivity

Henricksen (1980) has compiled a chart of the coverage angle, which is defined as the included angle between the −6 dB down points on a polar plot, versus the source directivity for various sizes of cone loudspeakers. It is reproduced as Fig. 6.20. Referring to this figure, a coverage angle of 90° is equivalent to a Q of about 10. Therefore, if a cone loudspeaker or a horn is to achieve significant directional control, its mouth must be at least a wavelength long in the plane of the coverage angle. Typical directivity patterns for various sizes of cone loudspeakers and baffles are shown in Fig. 6.21 (Henricksen, 1980). The low-frequency directivity is determined by the size of the baffle, which sets the point at which the loudspeaker-baffle combination reverts to a point source with 360° coverage. Henricksen generalized the expected behavior from various loudspeaker and baffle sizes in Fig. 6.22. Here the high-frequency directivity is controlled by cone size, with the smaller cones being less directional. It is interesting to note that in home and near-field monitor loudspeakers, where wide dispersion angles are desired, the crossover to a smaller loudspeaker is made when the wavelength is equal to the cone diameter (k a $= \pi$). In sound reinforcement systems, where pattern control is of paramount importance, the crossover is made to a larger format driver when the wavelength is equal to a diameter. This is one reason why monitor loudspeakers are usually not appropriate for commercial sound reinforcement applications and why these systems can be made much smaller than commercial systems.

Loudspeaker Arrays and the Product Theorem

When an array of identical loudspeakers is constructed, the composite directivity pattern is determined by the directional characteristics of the array as well as the inherent directivity of the loudspeakers. The relationship between these directivities is given by the product theorem, which states that the overall directivity of an array of identical sources is the product

FIGURE 6.21 A Chart for Predicting the −6 dB Angle Response for Piston and Box Systems (Henricksen, 1980)

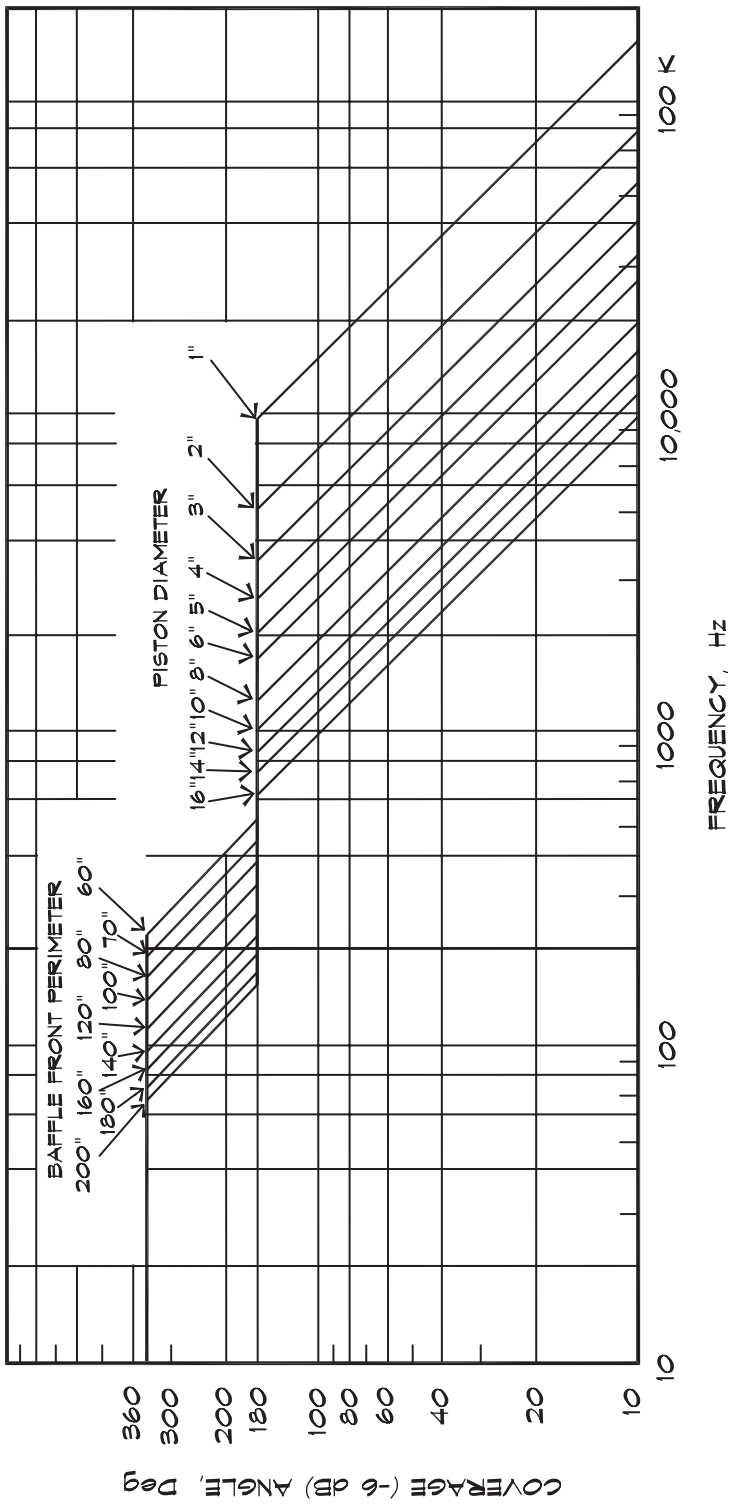
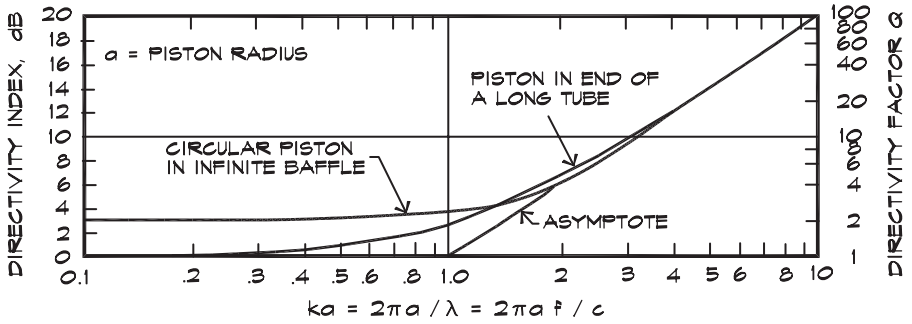


FIGURE 6.22 Directivity Index Response of Direct-Radiator Pistons in Various Configurations (Beranek, 1954)



of the directivities of the individual sources and the directivity due to the array.

$$Q_\theta\,(\theta,\,\phi) = Q_0\;Q_{rel}\,(\theta,\,\phi)\;R^2_\theta\,(\theta,\,\phi) \tag{6.63}$$

where $Q_\theta\,(\theta,\,\phi)$ = overall array directivity

Q_0 = on - axis directivity for an individual speakers

$Q_{rel}\,(\theta,\,\phi)$ = off - axis directivity of a given loudspeaker

$R^2_\theta\,(\theta,\,\phi)$ = array directivity relative to the on - axis sound intensity

If the array is composed of different types of loudspeakers or loudspeakers at various levels, for which there is no ready directional characteristic, the directivity must be calculated for each element of the array from its off-axis level and the relative phase due to its position in the array. The off-axis phase behavior for individual loudspeakers may also be important but is seldom available in the data published by loudspeaker manufacturers. Computer programs that perform these calculations are commercially available.

There are theoretical directivity values published for arrays that have a tapered volume level or other more exotic shadings. Olson (1957) has given a general equation for the directivity of a linear array having an arbitrary taper. Davis and Davis (1987) have published a scheme for improving the directivity of an array of loudspeakers in a baffle using a Bessel function weighting. Line arrays have been designed using axially rotated elements to disperse the high frequencies or vertically angled elements to smooth the frequency response. In these arrayed systems, it usually is assumed that each element radiates individually without influence from the others. For this assumption to hold true each transducer should be housed in a separate enclosure so that the loading produced by back radiation into a common cabinet does not influence the other loudspeakers.

Rectangular Pistons

The directional characteristic of a coherent rectangular piston source can be calculated in a similar fashion to that used for the circular piston in a baffle. McLachlan published these calculations in 1934. He showed that a rectangular rigid plate of length 2 a and height 2 b, which vibrates in an infinite rigid baffle, generates a far field directional characteristic

$$R_{\theta,\phi} = \left\{\left[\frac{\sin\,(k\;a\;\cos\,\phi)}{(k\;a\;\cos\,\phi)}\right]\left[\frac{\sin\,(k\;b\;\cos\,\theta)}{(k\;b\;\cos\,\theta)}\right]\right\} \tag{6.64}$$

The term in the second set of brackets is the same as that found for a line source in Eq. 6.47. In fact, the rectangular piston has the same directional characteristic as the product of two line sources of length 2a and 2b arranged at right angles to one another, whose strength and phase are the same. The rectangular mouth of a horn loudspeaker can be modeled using this equation.

Force on a Piston in a Baffle

The impedance seen by a piston in a baffle can be obtained from Eq. 6.54; in this case the integral is evaluated close to the piston rather than in the far field. The pressure varies across the face of the piston but the quantity of interest is the force on the whole piston. This, along with the piston velocity, yields the mechanical radiation impedance seen by the piston (see Morse, 1948 or Kinsler et al., 1982).

$$\mathbf{z}_\mathrm{r} = \int \frac{\mathrm{d}\mathbf{F}_\mathrm{S}}{\mathbf{u}} = \rho_0 \, \mathrm{c}_0 \, \mathrm{S} \left[w_\mathrm{r}(2\mathrm{ka}) + \mathrm{j} \, x_\mathrm{r} \, (2\mathrm{ka}) \right] \tag{6.65}$$

The integration of the force over the face of the piston is complicated and will not be reproduced in detail. The impedance terms include a real part

$$w_r = 1 - \frac{\mathrm{J}_1(2\mathrm{k\,a})}{\mathrm{k\,a}} = \frac{(\mathrm{k\,a})^2}{2} - \frac{(\mathrm{k\,a})^4}{2^2 \cdot 3} + \frac{(\mathrm{k\,a})^6}{2^2 \cdot 3^2 \cdot 4} - \cdots \tag{6.66}$$

which has limiting values of

$$\begin{aligned} w_r &\rightarrow \frac{(\mathrm{k\,a})^2}{2} \quad \text{for } \mathrm{k\,a} << 1 \\ &\rightarrow \quad 1 \quad \text{for } \mathrm{k\,a} >> 1 \end{aligned} \tag{6.67}$$

and an imaginary part

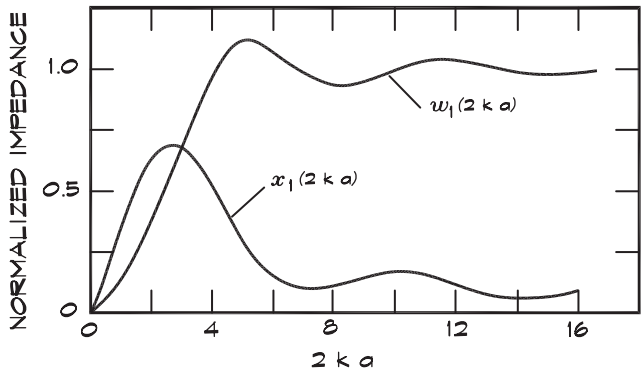
$$x_r = \frac{1}{\pi(\mathrm{k\,a})^2} \left[\frac{(2\mathrm{k\,a})^3}{3} - \frac{(2\mathrm{k\,a})^5}{3^2 \cdot 5} + \frac{(2\mathrm{k\,a})^7}{3^2 \cdot 5^2 \cdot 7} - \cdots \right] \tag{6.68}$$

with limits

$$\begin{aligned} x_r &\rightarrow \frac{8\,\mathrm{k\,a}}{3\,\pi} \quad \text{for } \mathrm{k\,a} << 1 \\ &\rightarrow \frac{2}{\pi \mathrm{k\,a}} \quad \text{for } \mathrm{k\,a} >> 1 \end{aligned} \tag{6.69}$$

Figure 6.23 shows the resistive (real) and reactive (imaginary) components of the radiation impedance. At low frequencies, the reactive component dominates, whereas at high frequencies, the resistance becomes more important. The impedance at high frequencies approaches the area times $\rho_0 \, \mathrm{c}_0$. We will need these results in our later analysis of the absorption due to quarter-wave resonator tubes and quadratic-residue diffusers.

FIGURE 6.23 Impedance Functions of a Baffled Piston (Kinsler et al., 1982)



6.5 LOUDSPEAKERS

Cone Loudspeakers

A moving coil or cone loudspeaker, illustrated in Fig. 6.24, is the most commonly used type. It consists of a circular cone of treated paper or other lightweight material, which is attached to a coil of wire suspended in a permanent magnetic field. When a current passes through the wire, the coil is forced out of the magnetic field in one direction or another depending on the direction of the current. A sinusoidal voltage applied to the wire results in sinusoidal motion of the cone. Many broadband cone loudspeakers have a small dome or dust cap at their center, which helps disperse the high frequencies.

FIGURE 6.24 A Simple Moving Coil Loudspeaker (Kinsler et al., 1982)

A) Magnet, B) Voice Coil, C) Diaphragm, D) Corrugated Rim, E) Spider (for stiffness), F) Dome

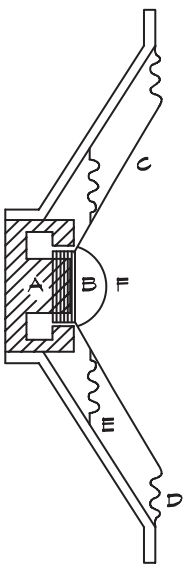
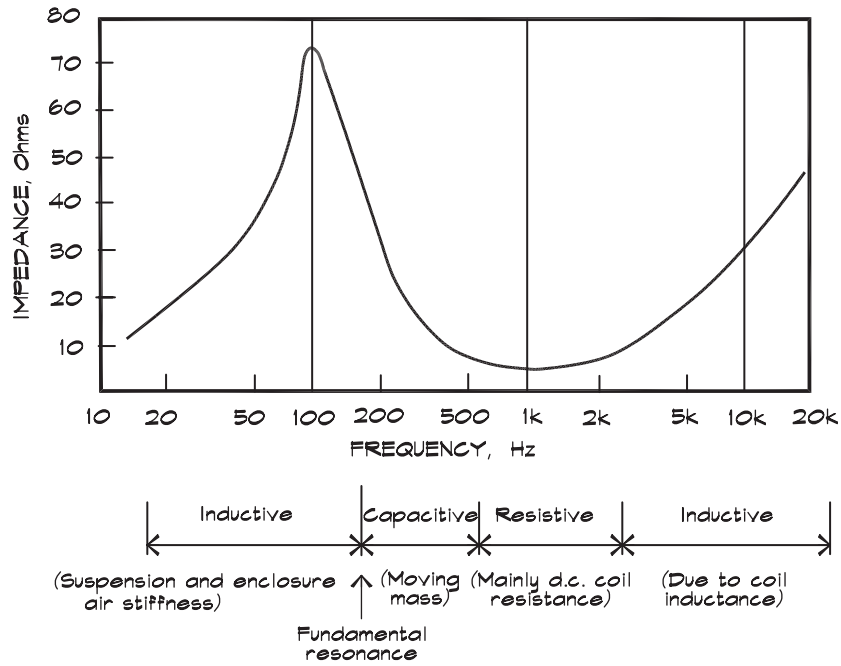


FIGURE 6.25 Input Impedance of a Moving Coil Driver (Colloms, 1980)



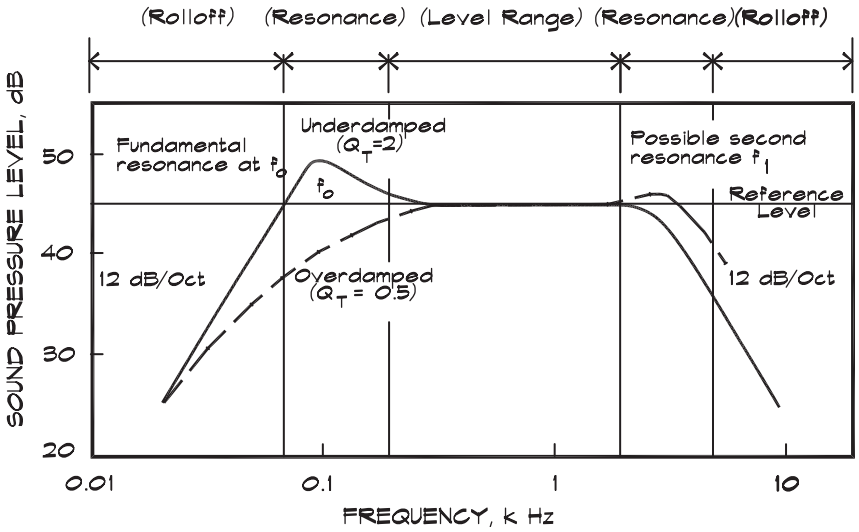
Cone loudspeakers are not particularly efficient sound radiators. Typically, they convert between 0.5 and 2% of the electrical energy to sound. A loudspeaker, driven with 1 electrical Watt, will radiate about 0.01 acoustical Watts of energy, which is equal to a sound power level of 100 dB. At a distance of 1 meter, such a loudspeaker would result in a sound pressure level of about 92 dB assuming a Q of two. This number, that is the on-axis level generated at 1 m for 1 W of power, is the sensitivity of the loudspeaker and was defined in Eq. 2.76.

Loudspeakers are characterized not only by their sensitivity but also by their impedance, frequency response, directivity, and polar pattern or coverage angle. Each of these parameters is useful to the designer. The electrical impedance is like the acoustical impedance in that it represents the electrical resistance of the loudspeaker, which is a complex number. Not all of the resistance is electrical. The mechanical impedance is reflected back as electrical impedance, which is not constant with frequency. Figure 6.25 gives a typical impedance curve for a loudspeaker in a baffle.

The low-frequency peak is at the fundamental resonance of the loudspeaker cone's spring mass system, including the spring effect of the air suspension system. Above the resonant frequency, the impedance drops to a region where it mainly consists of the dc resistance of the coil and reaches a minimum value, which can be measured with an ohmmeter. At the high- and low-frequency extremes, the impedance is mainly inductive, due to the stiffness of the suspension system at low frequencies, and to the coil inductance at high frequencies. Usually the minimum is listed by loudspeaker manufacturers since it is this value that controls the maximum current flow at a given applied voltage.

The frequency response of a loudspeaker is available from the manufacturer, and there is an example in Fig. 6.26. It is measured by sweeping a signal of constant voltage across the frequency range of interest and measuring the sound pressure level of the loudspeaker

FIGURE 6.26 Response of a Moving Coil Driver (Colloms, 1980)



on-axis at a given distance. At both high and low frequencies the response curve rolls off at about 12 dB per octave. In between, it is relatively flat.

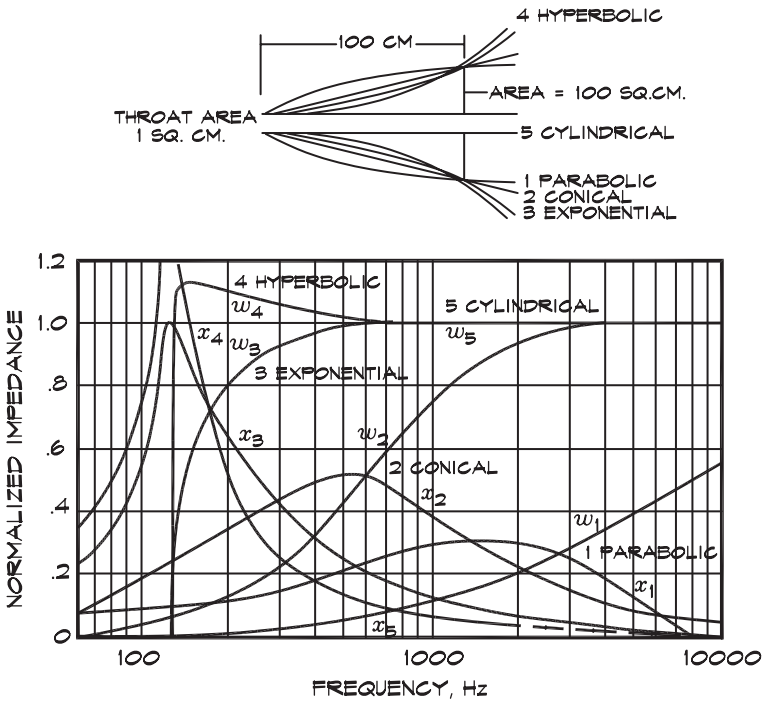
The low-frequency portion of the curve is influenced by the configuration of the enclosure. If the loudspeaker is not enclosed the cone radiates as a dipole, with the sound coming from the back canceling out the sound coming from the front. An infinite baffle reduces the dipole effect but is not always convenient to build. An enclosed box helps improve the low-frequency response by eliminating the dipole effect, but the air spring increases the resonant frequency of the cone. A ported enclosure acts as a second loudspeaker at low frequencies, which radiates in phase with the front of the cone. Due to the port resonance, the box emits more energy at low frequencies than the cone does.

Horn Loudspeakers

The use of horns undoubtedly originated with the cupping of the hands around the mouth to increase projected level. The Greeks applied the same idea when they attached conical horns to their theatrical masks to amplify the actors' voices. The development of brass instruments, at least as far back as the Romans, used a flared bell mouth. In the early twentieth century the exponential horn was studied and incorporated by Edison into his phonograph. The exponential horn shape was utilized in one form or another until the early 1980s, when the constant-directivity horn was introduced. Most loudspeaker manufacturers now offer a version of this type of horn.

A horn serves three functions. First, it provides an increased resistive loading for the driver so that it can work against a higher impedance than the air alone would present. Second, it improves the efficiency of the driver by constraining the air that is moved, gradually transitioning the air column into the surrounding space. Third, it controls the coverage pattern of the sound wave by providing side walls, which direct the beam of energy as it radiates away from the driver. Each of these functions may make conflicting demands on the horn designer. The shape that is the most efficient for impedance matching does not always provide the required directivity. The designer has to determine the functions the horn must perform, and trade off the advantages and disadvantages of each to get the best compromise solution.

FIGURE 6.27 Resistive and Reactive Components of the Normalized Acoustic Impedance of Infinite Horns (Olson, 1957)



The presence of a horn increases the impedance that is presented to the driver diaphragm. This causes the diaphragm to push against a higher pressure, which, although it makes the driver work harder, also allows more energy to be transmitted to the air. The resistive load placed on the diaphragm is almost totally dependent on the shape of the horn. Figure 6.27 plots the acoustical resistance of an infinitely long horn for five different shapes. As the diagram shows, a conical (straight-sided) horn does not add significant loading to the diaphragm. The cylindrical tube presents an even loading but has no increase in mouth area. The exponential horn, so called because the shape of its sides follows the exponential equation

$$S = S_0 \, e^{m x} \tag{6.70}$$

where S = horn area at distance x (m^2)
S_0 = throat area at $x = 0$ (m^2)
m = flare constant (m^{-1})
x = distance from the throat (m)

provides smooth loading down to a cutoff frequency

$$f_c = \frac{m\,c}{4\,\pi} \tag{6.71}$$

below which sound does not propagate without loss. Although the loading is constant with frequency for an exponential horn, the shape is not necessarily the best choice. For low-frequency directional control, the horn mouth size has to be relatively large, which leads to high-frequency control problems.

As we have seen from the piston in a baffle analysis, loudspeakers undergo a narrowing in their coverage pattern, called beaming, at high frequencies. To achieve wide coverage, high-frequency drivers must be physically small. Small drivers, however, are inefficient since they neither travel very far, nor push much air. Coupling a small driver to a horn helps solve both the beaming and the efficiency problems. Horn efficiencies as high as 50% can be achieved over a narrow range of frequencies; however, for broadband signals an efficiency of 10% is more likely. The sensitivity of a typical large format horn/driver is about 113 dB, which, with an on-axis Q of 20, represents an efficiency of a little more than 13% or an acoustic power of about 0.13 Watts.

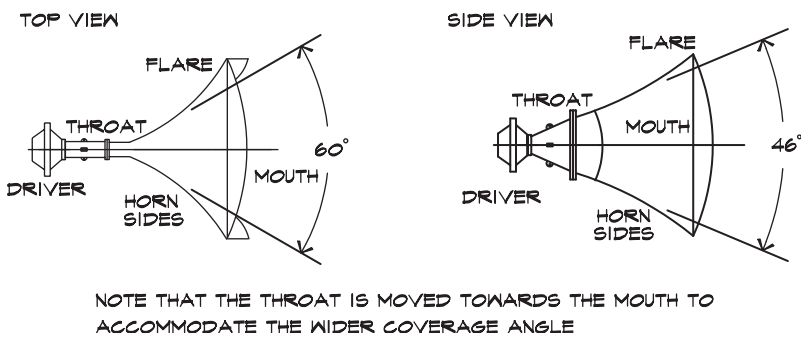
At mid frequencies, where the size of the driver mouth is small with respect to the wavelength, the sound illuminates the side walls and allows them to control the directional pattern emanating from the horn. With constant directivity horns, the side walls are either straight or slightly curved, having different centers of expansion in the horizontal and vertical dimensions. This innovation was introduced by Paul Klipsch in 1951 and has been used in most subsequent horn designs. It allows for a different coverage angle in the horizontal and vertical planes.

Constant-Directivity Horns

Constant-directivity horns are specifically designed to provide an even frequency distribution with direction. A typical example is shown in Fig. 6.28. In the ideal case, the sound spectrum measured at any particular location should be no different from that measured at any other location within the field of the horn's coverage. One result of this type of behavior is that the spectrum at any point is quite close to the actual power spectrum of the driver, since the power is evenly distributed. This feature is critical for successful sound system design and greatly simplifies the design process.

Modern constant directivity horn design started with the work of D. B. (Don) Keele, Jr., John Gilliom, and Ray Newman at Electrovoice in the middle of the 1970s. Their work introduced the features that are the central methods for controlling directivity in all current horn design. Their first design idea was a contraction in the throat immediately following

FIGURE 6.28 Constant Directivity Horn—Nominal 60 × 40 (JBL 2365)



the driver. This detail is emphasized in the White horn series they designed; however, the idea predates this horn (Keele, 1983). The narrow throat allowed the use of a larger driver and improved the directional control by letting the sound illuminate the sides of the horn, without the high-frequency beaming usually associated with larger throat sizes.

The second feature was a conical-exponential (CE) throat shape, which consisted of an exponential throat over a certain distance, after which there was a smooth transition into a conical (straight-sided) shape. The combination of these two curves allowed the control of low-frequency impedance by the use of the quasi-exponential throat expansion, and still maintained excellent directional coverage afforded by the conical shape of the sides.

The third step the group took was to address the problem of mid-range narrowing, present in most horns before this design. Their approach was to flare the mouth of the horn at a point, which was about two-thirds the distance from the beginning of the conical section to the mouth. The flare was added at an angle, which was about twice the angle of the original conical section. The added flare resulted in the high frequencies seeing one mouth size and the lower frequencies seeing another. The flare also allowed the transition between the horn mouth and the surrounding air to be less abrupt. The sound pressure distribution across the mouth of the horn was no longer constant but was higher in the center of the horn. The horn mouth no longer looked like a pure piston in a baffle and as a result the mid-frequency narrowing problems (predicted by the piston model) that were associated with previous designs were no longer present.

The White horn series was highly successful. The design gave good horizontal-directivity control without the mid-frequency beaming that had been associated with most previous horns. The vertical frequency control was not emphasized in the design in favor of a smaller vertical dimension. This shows the design tradeoff that is made between horn mouth height and the capability of controlling the vertical directivity over a wide range of frequencies. Because the vertical dimension of the mouth is relatively small, the frequency at which the vertical control begins is rather high—1.2 kHz. The White horn series was the first commercial product to be a true constant-directivity type.

The next chronological development in horn design was the introduction of the Mantaray horn series by Mark Ureda and Cliff Henricksen at Altec. They wanted to produce a horn with strong directivity control both horizontally and vertically. Ureda and Henricksen decided that if bidirectional control was to be achieved, then the mouth had to be square and relatively large. Once the low-frequency limit was known, the mouth size could be determined from the piston in a baffle formula. With the mouth size fixed, the coverage angles allowed the sides to be drawn back to the driver's mouth. The large vertical mouth dimension resulted in narrow angled top and bottom walls extending back to the throat opening, which controlled length of the horn as well as the loading on the driver. In the horizontal plane the wider coverage angle resulted in the sides converging to a point that was displaced further down the horn from the throat. The throat was connected to this point by an opening having a relatively narrow cross section. The Mantaray design used the flare idea developed by Keele et al., but its flare started further down the horn than the two-third's point.

This horn had some advantages over previous designs. Because the vertical mouth dimension was large, the narrowest side wall angle expanded over a longer distance and was connected directly to the driver. This displaced the driver from the throat of the wide-angle portion of the horn and made it easier for the driver to couple to the horn. This feature is an advantage particularly when using 2-inch drivers, which have a difficult time driving directly into a 90° angle opening without beaming.

The Mantaray horns emphasize the control of directivity in both planes, but do so at the expense of low-frequency loading, which is brought about by an exponential area expansion. The sides of the Mantaray are virtually straight in both the horizontal and vertical planes. The Altec patent claims that the straight-sided walls improve the waistbanding effect (a sideways lobing of the midfrequencies), which is said to have been found in other designs. The Mantaray design yields a directivity that is highly controlled down to 800 Hz and out to 20 kHz. Because of their size, they require more space than other horns. The tradeoffs are good vertical directivity control and excellent high-frequency response against a large physical size and minimal low-frequency loading.

Another horn manufacturer, JBL, subsequently developed its own constant directivity horns, which were also designed by Keele. In these designs, called Bi-Radial horns, Keele used a very general polynomial formula to develop the side shape from the throat to the mouth.

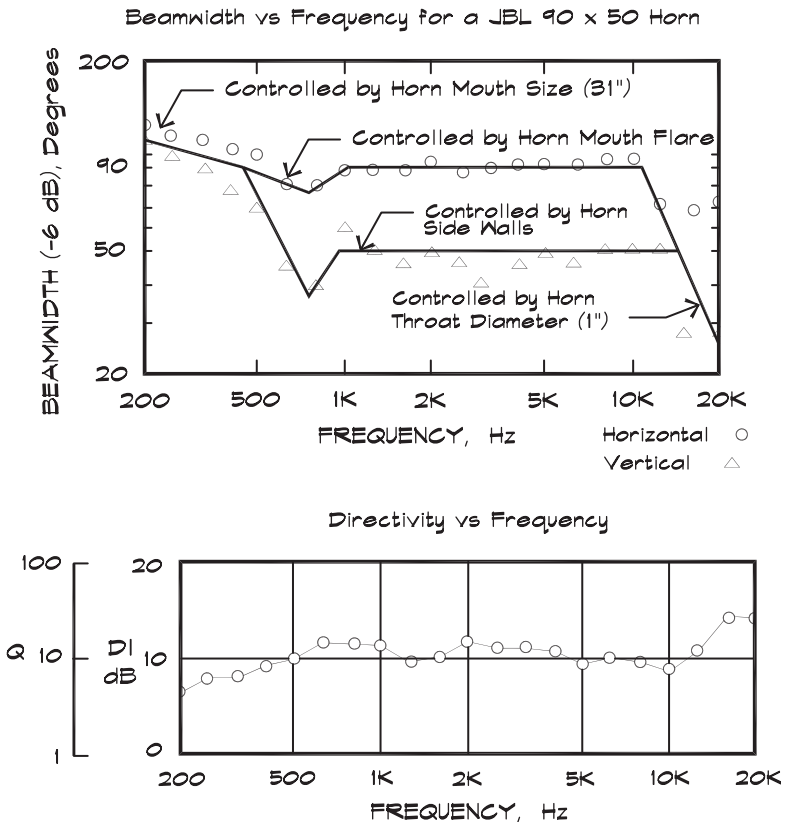
$$y = a + bx + cx^n \tag{6.72}$$

where x = distance along the centerline from the mouth
y = distance perpendicular to the x axis
a = half the throat height
b = [tan (0.9beamwidth)] /2
$c = [w/2 - bL - a] / L^n$
n = constant between 2 and 8
L = horn length
w = mouth width

The Bi-Radial design, as with all horn designs, is a compromise. The mouth size is selected from the piston in a baffle equation. The throat expansion is quasi-exponential as it is a combination of constant linear taper and an exponential area expansion. The loading is better than the straight-sided Mantaray conical loading, but not as good as the more purely exponential loading used in the Electrovoice designs. The mouth size is large in both dimensions, so that the directivity is controlled to relatively low frequencies. Probably the most interesting design feature is the use of the equation for the flare rate. According to Keele (1983), this curve gives a smooth response both along the horizontal and vertical directions as well as off-axis between the two planes. The line determined by the equation is rotated about a point on two sides and hence the name Bi-Radial. The horn does not have a contraction near the throat so that the horn tends to beam above 10 kHz. The Bi-Radial design is generally a good compromise between loading and directivity. It provides control in both directions and smooth off-axis response.

In Fig 6.29, we can see the various regions of the horn and what controls the directivity in each region. At low frequencies, the size of the mouth is the determinant. It sets the frequency at which the horn begins to control. Above the control point, the angle of the horn sides sets the coverage pattern. At very high frequencies, the diameter of the driver opening controls the beamwidth since the sound no longer interacts with the sides of the horn. Between the side-angle region of control and the low-frequency cutoff point, there is waistbanding or narrowing of the coverage pattern. This is controlled by the horn flare, which prevents the mouth of the horn from acting like a pure piston in a baffle.

FIGURE 6.29 Beamwidth and Directivity of Constant Directivity Horns (Long, 1983)



Cabinet Arrays

In recent years, it has become popular for manufacturers to provide two- or three-way cabinets that are trapezoidal-shaped. The idea behind the shape is that the cabinets may be arrayed side by side to provide the appropriate coverage. With the advent of packaged computer design programs, the directivity patterns of these products often are buried in the computer code and not available to the designer. Cabinets are subject to the same physical limitations on directional control imposed by the size of the radiating components as any other device. Arraying cabinets in a line can does not narrow the coverage angle in the plane normal to the line, where directional control may also be needed. At frequencies below the point where the spacing is equal to a wavelength, horizontal stacking will narrow the coverage angle in the horizontal plane, which may or may not be useful. Stacking cabinets in a line can control directivity over a certain frequency range, where the components act as a line array, but is probably not particularly beneficial above that frequency.

Baffled Low-Frequency Systems

The installation of low-frequency cabinets in a baffle wall is a technique that can be used to increase their directivity somewhat. The theoretical result is illustrated in Fig. 6.21. The materials used in constructing these baffles are usually not 100% reflective at the frequencies of interest. For example, a single sheet of drywall is about 30% absorptive at 125 Hz and as much as 40% absorptive at slightly lower frequencies. This significantly reduces the

theoretical effectiveness of a lightweight baffle wall. Double drywall, being about 20% absorptive at 125 Hz and 35% absorptive at 80 Hz, is not significantly better.

Since subwoofer cabinets usually are used below 125 Hz, where the wavelength is about 9 ft (2.8 m) long, it can be helpful to place them on the floor or up against a concrete wall where they would be less than a third of a wavelength away from their image source. From Fig. 6.9 we can see that a power doubling, if not a pressure doubling, would probably be achieved for $k\,d = 2$. When there are two reflecting surfaces, the floor and the wall, a 6 dB increase could be achieved.

Baffle walls, if improperly constructed, can sometimes do more harm than good. When an unsealed gap is left around a baffled loudspeaker, it can become the throat of a Helmholtz resonator with the enclosed volume behind the baffle wall acting as the resonator volume. The resulting resonance can significantly color the sound and offset the advantage of a small increase in level at low frequencies provided by the wall.

7 SOUND AND SOLID SURFACES

The interaction of sound with solid surfaces could well be taken as the beginning of architectural acoustics. Sound undergoes three types of fundamental interactions upon encountering an object: reflection, absorption, and transmission. Each of these occurs to some degree when an impact takes place, although usually we are concerned with only one at a time.

7.1 PERFECTLY REFLECTING INFINITE SURFACES

Incoherent Reflections

Up to this point we have considered sound waves to be free to propagate in any direction, unaffected by walls or other surfaces. Now we will examine the effect of reflections, beginning with a perfectly reflecting infinite surface. The simplest model of this interaction occurs with sound sources that can be considered incoherent; that is, where phase is not a consideration. If an omnidirectional source is placed near a perfectly reflecting surface of infinite extent, the surface acts like a mirror for the sound energy emanating from the source. The intensity of the sound in the far field, where the distance is large compared to the separation distance between the source and its mirror image, is twice the intensity of one source. Figure 7.1 shows this geometry. In terms of the relationship between the sound power and sound pressure levels for a point source given in Eq. 2.61,

$$L_p = L_w + 10 \log \frac{Q}{4 \pi r^2} + K \tag{7.1}$$

where L_p = sound pressure level (dB re 20 μN/ m^2)
L_w = sound power level (dB re 10^{-12} W)
Q = directivity
r = measurement distance (m or ft)
K = constant (0.13 for meters or 10.45 for ft)

When the source is near a perfectly reflecting plane, the sound power radiates into half a sphere. This effectively doubles the Q since the area of half a sphere is $2 \pi r^2$. If the source is near two perfectly reflecting planes that are at right angles to one another, such as a floor and a wall, there is just one quarter of a sphere to radiate into, and the effective Q is 4.

FIGURE 7.1 Construction of an Image Source

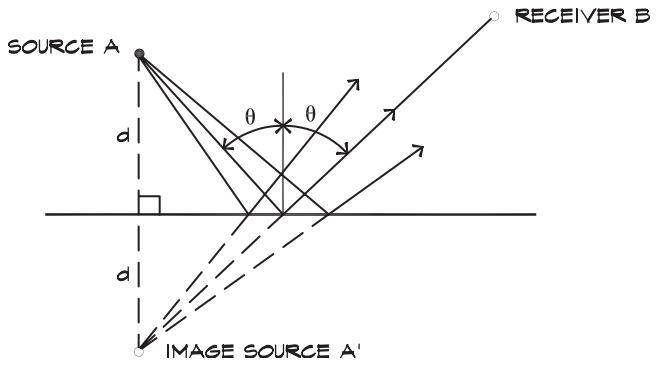
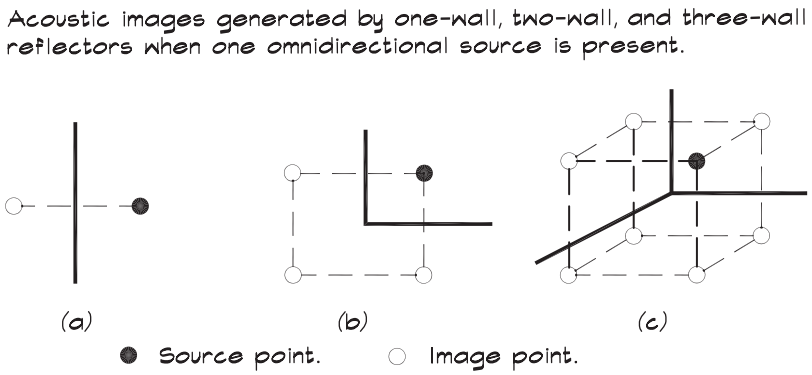


FIGURE 7.2 Multiple Image Sources



For a source located in a corner bounded by three perpendicular surfaces, the effective Q is 8. Figure 7.2 illustrates these conditions. For a nondirectional source such as a subwoofer, clearly the corner of a room is the most efficient location.

Note that the concept of Q is slightly different here than it is for the inherent directivity associated with a source. The directivity associated with the position of a source must be employed with some discretion. If a directional source such as a horn loudspeaker is placed in the corner of a room pointed outward, then the overall directivity does not increase by a factor of 8, since most of the energy already is focused away from the reflecting surfaces. The mirror image of the horn, pointed away from the corner, also contributes, but only a small amount at high frequencies. Thus changes in Q due to reflecting surfaces must also account for the inherent directivity of the source.

Coherent Reflections—Normal Incidence

When the sound is characterized as a plane wave, moving in the positive x direction, we can write an expression for the behavior of the pressure in space and time

$$\mathbf{p}\,(\mathrm{x}) = \mathbf{A}\,\mathrm{e}^{\,\mathrm{j}\,(\omega\,\mathrm{t}\,-\,\mathrm{k}\,\mathrm{x})} \tag{7.2}$$

If we place an infinite surface at $\mathrm{x} = 0$, with its normal along the x axis, the equation for the combined incident and reflected waves in front of the surface is

$$\mathbf{p}\,(\mathrm{x}) = \mathbf{A}\,\mathrm{e}^{\,\mathrm{j}\,(\omega\,\mathrm{t}\,-\,\mathrm{k}\,\mathrm{x})} + \mathbf{B}\,\mathrm{e}^{\mathrm{j}\,(\omega\,\mathrm{t}\,+\,\mathrm{k}\,\mathrm{x})} \tag{7.3}$$

The particle velocity, **u**, defined in Eq. 6.31 as

$$\mathbf{u}(\mathrm{x}) = \frac{\mathrm{j}}{\mathrm{k}\,\rho_0 \mathrm{c}_0}\left(\frac{\partial \mathbf{p}}{\partial \mathrm{x}}\right) \tag{7.4}$$

becomes

$$\mathbf{u}(\mathrm{x}) = \frac{\mathrm{j}}{\mathrm{k}\,\rho_0 \mathrm{c}_0}\left[-\mathrm{j}\,\mathrm{k}\,\mathbf{A} + \mathrm{j}\,\mathrm{k}\,\mathbf{B}\right] \mathrm{e}^{\mathrm{j}\,\omega\,\mathrm{t}} \tag{7.5}$$

or

$$\mathbf{u}(\mathrm{x}) = \frac{1}{\rho_0 \mathrm{c}_0}\left[\mathbf{A} - \mathbf{B}\right] \mathrm{e}^{\mathrm{j}\,\omega\,\mathrm{t}} \tag{7.6}$$

When the surface is perfectly reflecting, the amplitude $\mathbf{A} = \mathbf{B}$ and the particle velocity is zero at the boundary. Mathematically the reflected particle velocity cancels out the incident particle velocity at $\mathrm{x} = 0$.

The ratio of the incident and reflected-pressure amplitudes can be written as a complex amplitude ratio

$$\boldsymbol{r} = \frac{\mathbf{B}}{\mathbf{A}} \tag{7.7}$$

When $\boldsymbol{r} = 1$, Eq. 7.2 can be written as

$$\mathbf{p}(\mathrm{x}) = \mathbf{A}\ \mathrm{e}^{\mathrm{j}\,\omega\,\mathrm{t}}\left[\mathrm{e}^{\mathrm{j}\,\mathrm{k}\,\mathrm{x}} + \mathrm{e}^{-\mathrm{j}\,\mathrm{k}\,\mathrm{x}}\right] = 2\ \mathbf{A}\ \cos\left(\mathrm{k}\,\mathrm{x}\right)\ \mathrm{e}^{\mathrm{j}\,\omega\,\mathrm{t}} \tag{7.8}$$

which has a real part

$$\mathrm{p}(\mathrm{x}) = 2\ \mathrm{A}\ \cos\left(\mathrm{k}\,\mathrm{x}\right)\ \cos\left(\omega\,\mathrm{t} + \varphi\right) \tag{7.9}$$

The corresponding real part of the particle velocity is

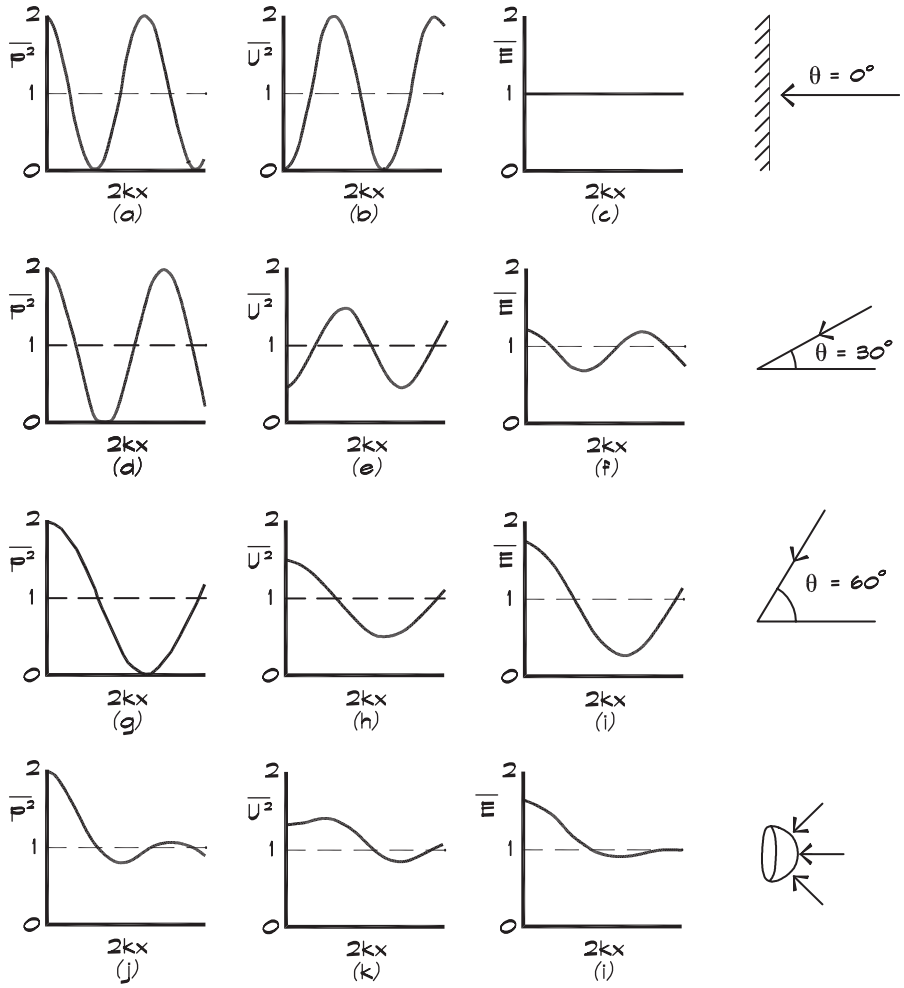
$$\mathrm{u}\,(\mathrm{x}) = \frac{2\ \mathrm{A}}{\rho_0 \mathrm{c}_0} \sin\left(\mathrm{k}\,\mathrm{x}\right) \cos\left(\omega\,\mathrm{t} + \varphi - \frac{\pi}{2}\right) \tag{7.10}$$

so the velocity lags the pressure by a 90° phase angle.

Equation 7.8 shows that the pressure amplitude, 2 A at the boundary, is twice that of the incident wave alone. Thus the sound pressure level measured there is 6 dB greater than that of the incident wave measured in free space. Figure 7.3 (Waterhouse, 1955) gives a plot of the behavior of a unit-amplitude plane wave incident on a perfectly reflecting surface at various angles of incidence. Note that since both the incident and reflected waves are included, the sound pressure level of the combined waves at the wall is only 3 dB higher than farther away.

The equations illustrated in Fig. 7.3a describe a standing (frozen) wave, whose pressure peaks and valleys are located at regular intervals away from the wall at a spacing that is related to frequency. The velocity in Fig. 7.3b exhibits a similar behavior. As we have seen, the particle velocity goes to zero at a perfectly reflecting wall. There is a maximum in the particle velocity at a distance $(2\mathrm{n} + 1)\,\lambda/4$ away from the wall, where $\mathrm{n} = 0, 1, 2$, and so on.

FIGURE 7.3 Interference Patterns When Sound Is Incident on a Plane Reflector from Various Angles (Waterhouse, 1955)



Interference patterns produced when sound is incident on a plane reflector at angles of 0°, 30°, 60°, and from all directions over a hemisphere. $\overline{p^2}$ is the normalized mean squared pressure, k the wave number, and x the distance from the reflector. $\overline{u^2}$ is the normalized mean square velocity and $\overline{E}$ is the normalized mean energy density.

Coherent Reflections—Oblique Incidence

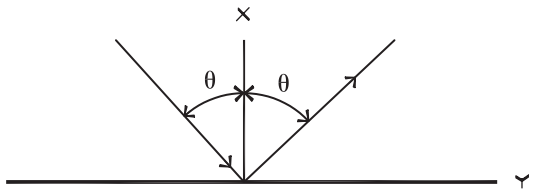
When a plane wave moving in the $-$x direction is incident at an oblique angle as in Fig. 7.4, the incident pressure along the x axis is given by

$$\mathbf{p} = \mathbf{A}\, e^{j\,k\,(x\,\cos\,\theta - y\,\sin\,\theta) + j\,\omega t} \tag{7.11}$$

For a perfectly reflecting surface the combined incident and reflected waves are

$$\mathbf{p} = \mathbf{A}\left[e^{j\,k\,x\cos\,\theta - j\,k\,y\,\sin\,\theta} + e^{-j\,k\,x\,\cos\,\theta - j\,k\,y\,\sin\,\theta}\right] e^{j\,\omega t} \tag{7.12}$$

FIGURE 7.4 Oblique Incidence Reflection



which is

$$\mathbf{p} = 2\,\mathbf{A}\, e^{-j\,k\,y\,\sin\,\theta + j\,\omega t} \cos\,(k\,x\,\cos\,\theta) \tag{7.13}$$

and the interference is still sinusoidal but has a longer wavelength. Looking along the x axis, the combined incident and reflected waves produce a pattern, which can be written in terms of the mean-square unit-amplitude pressure wave for perfectly reflecting surface given by

$$\langle \mathbf{p}^2 \rangle = [1 + \cos\,(2\,k\,x\,\cos\,\theta)] \tag{7.14}$$

As the angle of incidence θ increases, the wavelength of the pattern also increases. Figures 7.3a, d, and g show the pressure patterns for angles of incidence of 0°, 30°, and 60°.

Coherent Reflections—Random Incidence

When there is a reverberant field, the sound is incident on a boundary from any direction with equal probability, and the expression in Eq. 7.14 is averaged (integrated) over a hemisphere. This yields

$$\langle \mathbf{p}_r^2 \rangle = [1 + \sin\,(2\,k\,x)/\,2\,k\,x] \tag{7.15}$$

which is plotted in Fig. 7.3j.

The velocity plots in this figure are particularly interesting. Porous sound absorbing materials are most effective when they are placed in an area of high particle velocity. For normal incidence this is at a quarter wavelength from the surface. For off-axis and random incidence the maximum velocity is still at a quarter wavelength; however, there is some positive particle velocity even at the boundary surface that has a component perpendicular to the normal. Thus materials can absorb sound energy even when they are placed close to a reflecting boundary; however, they are more effective, particularly at low frequencies, when located away from the boundary.

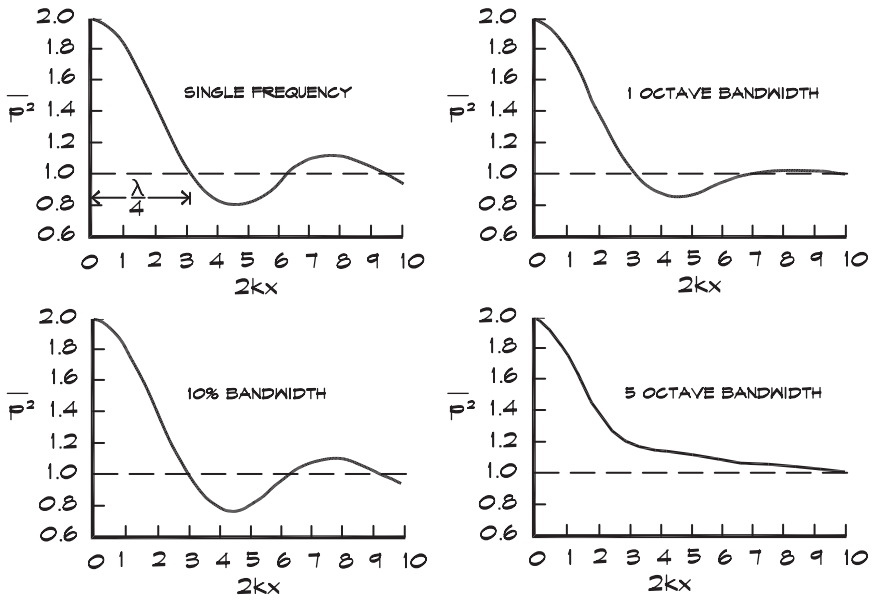
Coherent Reflections—Random Incidence, Finite Bandwidth

When the sound is not a simple pure tone, there is a smearing of the peaks and valleys in the pressure and velocity standing waves. Both functions must be integrated over the bandwidth of the frequency range of interest.

$$\langle \mathbf{p}_r^2 \rangle = \left[1 + \frac{1}{k_2 - k_1} \int_{k_1}^{k_2} \frac{\sin\,(2\,k\,x)}{2\,k\,x}\,dk \right] \tag{7.16}$$

FIGURE 7.5 Intensity vs Distance from a Reflecting Wall (Waterhouse, 1955)

Normalized intensity or mean square pressure $\overline{p^2}$ versus distance x of the sound reflected from a solid wall in a reverberant sound field for various bandwidths. In the abscissa $k = (k_1 + k_2) / 2$ where k_1 and k_2 are the wave numbers at the extremes of the band.



The second term is a well-known tabulated integral. Figure 7.5 shows the result of the integration. Near the wall the mean-square pressure still exhibits a doubling (6 dB increase) and the particle velocity is zero.

7.2 REFLECTIONS FROM FINITE OBJECTS

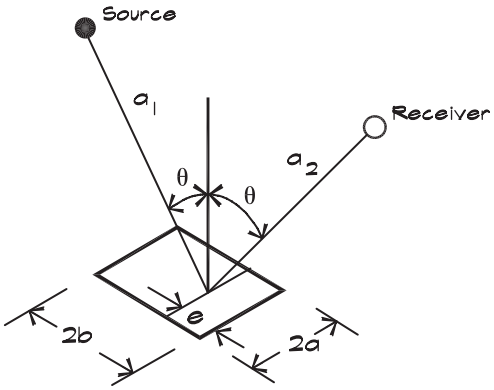
Scattering from Finite Planes

Reflection from finite planar surfaces is of particular interest in concert hall design, where panels are frequently suspended as "clouds" above the orchestra. Usually these clouds are either flat or slightly convex toward the audience. A convex surface is more forgiving of imperfect alignment since the sound tends to spread out somewhat after reflecting.

If a sound wave is incident on a finite panel, there are several factors that influence the scattered wave. For high frequencies impacting near the center of the panel, the reflection is the same as that which an infinite panel would produce. Near the edge of the panel, diffraction (bending) can occur. Here the reflected amplitude is reduced and the angle of incidence may not be equal to the angle of reflection. At low frequencies, where the wavelength is much larger than the panel, the sound energy simply flows around it like an ocean wave does around a boulder.

Figure 7.6 shows the geometry of a finite reflector having length 2b. When sound impacts the panel at a distance e from the edge, the diffraction attenuation depends on the closeness of the impact point to the edge, compared with the wavelength of sound. The reflected sound field at the receiver is calculated by adding up contributions from all parts of the reflecting surface. The solution of this integral is treated in detail using the

FIGURE 7.6 Geometry of the Reflection from a Finite Panel



Kirchoff-Fresnel approximation by Leizer (1966) or Ando (1985). The reflected intensity can be expressed as a diffraction coefficient K multiplied times the intensity that would be reflected from a corresponding infinite surface. For a rectangular reflector the attenuation due to diffraction is

$$\Delta L_{dif} = 10 \log K = 10 \log (K_1 K_2) \tag{7.17}$$

where K = diffraction coefficient for a finite panel
K_1 = diffraction coefficient for the x panel dimension
K_2 = diffraction coefficient for the y panel dimension

The orthogonal-panel dimensions can be treated independently. Rindel (1986) gives the coefficient for one dimension

$$K_1 = \frac{1}{2}\left\{\left[C(v_1) + C(v_2)\right]^2 + \left[S(v_1) + S(v_2)\right]^2\right\} \tag{7.18}$$

where

$$v_1 = \sqrt{\frac{\lambda}{2}\left[\frac{1}{a_1} + \frac{1}{a_2}\right]}\, e \cos\theta \tag{7.19}$$

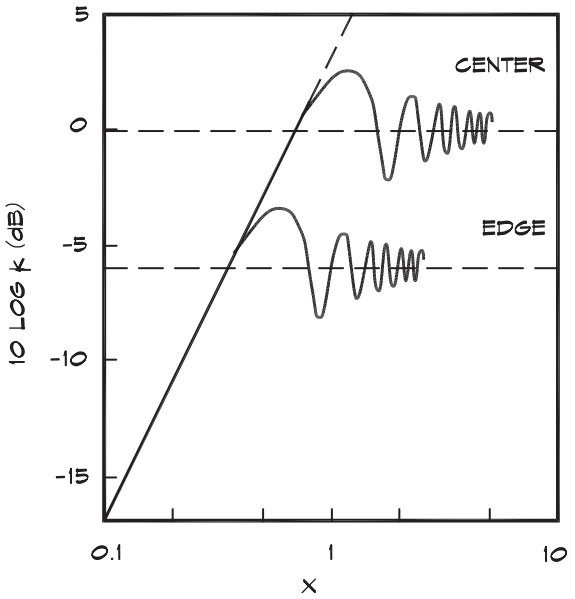
and

$$v_2 = \sqrt{\frac{\lambda}{2}\left[\frac{1}{a_1} + \frac{1}{a_2}\right]}\, (2b - e) \cos\theta \tag{7.20}$$

The terms C and S in Eq. 7.18 are the Fresnel integrals

$$C(v) = \int_0^v \cos\left(\frac{\pi}{2} z^2\right) dz, \quad S(v) = \int_0^v \sin\left(\frac{\pi}{2} z^2\right) dz \tag{7.21}$$

FIGURE 7.7 Attenuation of a Reflection Due to Diffraction (Rindel, 1986)



The integration limit v takes on the values of v_1 or v_2 according to the term of interest in Eq. 7.18. For everyday use these calculations are cumbersome. Accordingly we examine approximate solutions appropriate to regions of the reflector.

Rindel (1986) considers the special case of the center of the panel where e = b and $v_1 = v_2 = x$. Then Eq. 7.18 becomes

$$K_{1,\ \text{center}} = 2\left\{[C\ (x)]^2 + [S\ (x)]^2\right\} \tag{7.22}$$

where

$$x = 2b\cos\theta/\sqrt{\lambda\, a^*} \tag{7.23}$$

and the characteristic distance a^* is

$$a^* = 2\, a_1 a_2 / (a_1 + a_2) \tag{7.24}$$

Figure 7.7 gives the value of the diffraction attenuation. At high frequencies where $x > 1$, although there are fluctuations due to the Fresnel zones, a panel approaches zero diffraction attenuation as x increases. At low frequencies ($x < 0.7$) the approximation

$$K_{1,\ \text{center}} \cong 2\, x^2 \quad \text{for} \quad x < 0.7 \tag{7.25}$$

yields a good result.

At the edge of the panel where e = 0, $v_1 = 0$, and $v_2 = 2x$ we can solve for the value of the diffraction coefficient (Rindel, 1986)

$$K_{1,\ \text{edge}} = \frac{1}{2}\left\{[C\ (2x)]^2 + [S\ (2x)]^2\right\} \tag{7.26}$$

which is also shown in Fig. 7.7. The approximations in this case are

$$K_{1,\ \text{edge}} \cong 2\,x^2 \quad \text{for} \quad x \leq 0.35 \tag{7.27}$$

and

$$K_{1,\ \text{edge}} \cong 1/4 \quad \text{for} \quad x > 1 \tag{7.28}$$

Based on these special cases Rindel (1986) divides the panel into three zones according to the nearness to the edge of the impact point

a) $x \leq 0.35$: $K_1 \cong 2\,x^2$, independent of the value of e.

b) $0.35 < x \leq 0.7$: $K_1 \cong \frac{1}{4} + (e/b)\left(2\,x^2 - \frac{1}{4}\right)$, is a linear interpolation between the edge and center values.

c) $x > 0.7$: Here the concept of an edge zone is introduced whose width, e_o, is given by

$$e_o = \frac{b}{x\sqrt{2}} = \frac{1}{\cos\theta}\sqrt{\frac{1}{8}\lambda\, a^*} \tag{7.29}$$

If $e \geq e_o$, then we are in the region of specular reflection. When $e < e_o$, then diffraction attenuation must be considered. Rindel (1986) gives

$$K_1 \cong \begin{cases} 1 & \text{for } e \geq e_o \\ \frac{1}{4} + \frac{3\,e}{4\,e_o} & \text{for } e < e_o \end{cases} \tag{7.30}$$

Figure 7.8 compares these approximate values to those obtained from a more detailed analysis.

Rindel (1986) also cites results of measurements carried out in an anechoic chamber using gated impulses, which are reproduced in Fig. 7.9. He concludes that for values of

FIGURE 7.8 Calculated Values of K_1 (Rindel, 1986)

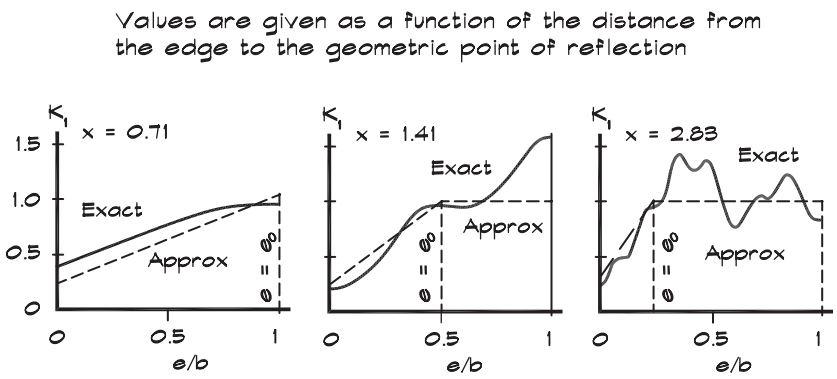
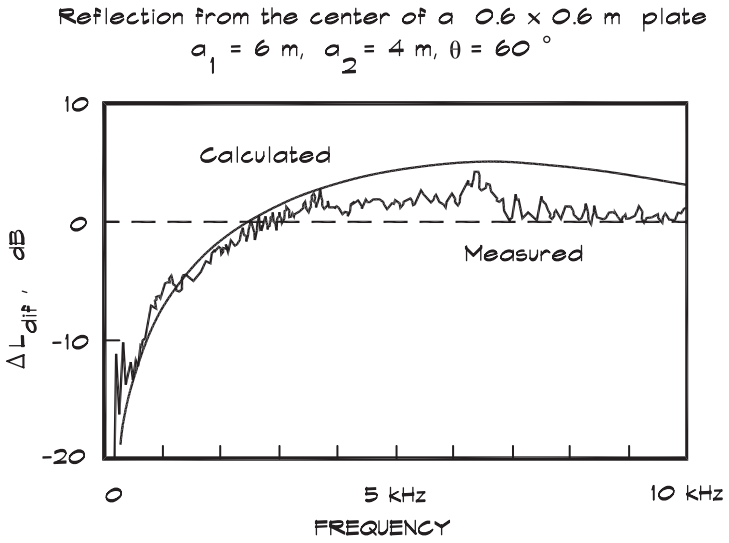


FIGURE 7.9 Measured and Calculated Attenuation of a Sound Reflection from a Square Surface (Rindel, 1986)



x greater than 0.7, edge diffraction is of minor importance. This corresponds to a limiting frequency

$$f_g > \frac{c\, a^*}{2\, S\, \cos\theta} \tag{7.31}$$

where S is the panel area. For a 2 m square panel the limiting frequency is about 360 Hz for a 45° angle of incidence and a characteristic distance of 6 m, typical of suspended reflectors.

Panel Arrays

When reflecting panels are arrayed as in Fig. 7.10, the diffusion coefficients must account for multipanel scattering. The coefficient in the direction shown is (Finne, 1987 and Rindel, 1990)

$$K_1 = \frac{1}{2}\left\{\sum_{i=1}^{I}\left[C\,(v_{1,i}) - C\,(v_{2,i})\right]^2 + \sum_{i=1}^{I}\left[S\,(v_{1,i}) - S\,(v_{2,i})\right]^2\right\} \tag{7.32}$$

FIGURE 7.10 Section through a Reflector Array with Five Rows of Reflectors (Rindel, 1990)

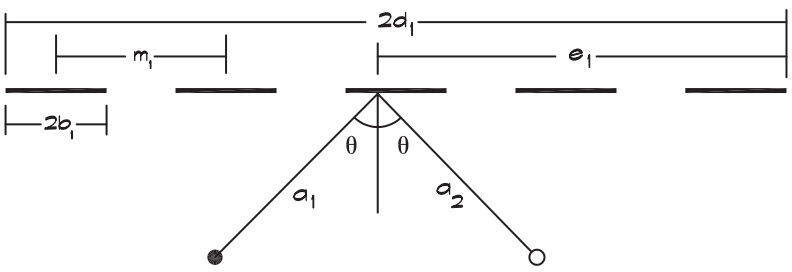
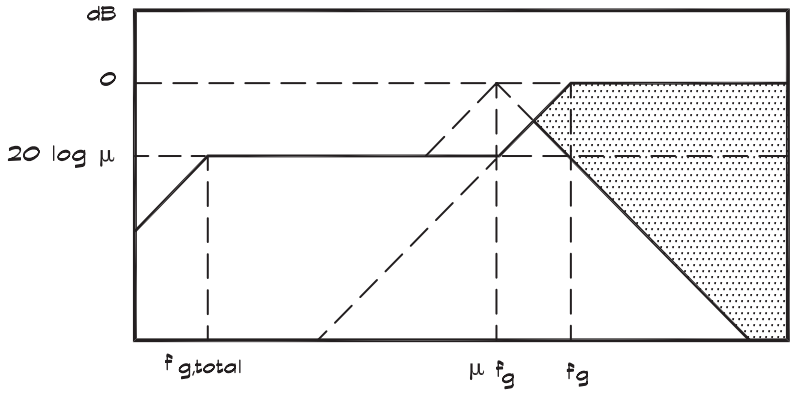


FIGURE 7.11 Simplified Illustration of the Attenuation of Reflections from an Array with Relative Density μ (Rindel, 1990)



where

$$v_{1,i} = \frac{2}{\sqrt{\lambda\, a^*}}(e_1 - (i-1)\, m_1)\cos\,\theta \tag{7.33}$$

and

$$v_{2,i} = \frac{2}{\sqrt{\lambda\, a^*}}(e_1 - 2b_1 - (i-1)\, m_1)\cos\,\theta \tag{7.34}$$

where i is the running row number and I is the total number of rows in the x-direction.

At high frequencies the v values increase and the reflection is dominated by an individual panel. The single-panel limiting frequency from Eq. 7.31 sets the upper limit for this dependence. At low frequencies the v values decrease, but the reflected vectors combine in phase. The diffusion attenuation becomes dependent on the relative panel area density, μ, (the total array area divided by the total panel area), not the size of the individual reflectors. Figure 7.11 shows a design guide.

The K values are approximately

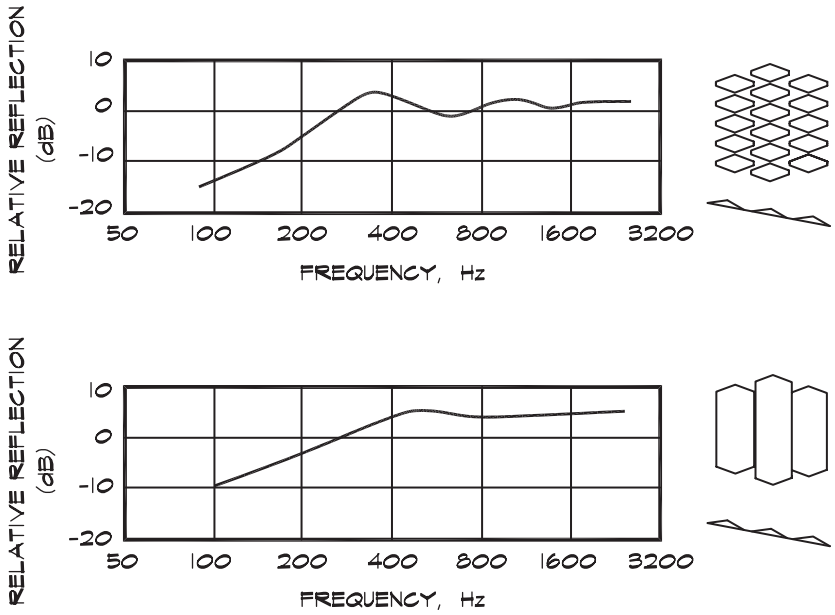
$$K \cong \mu^2 \quad \text{in the frequency range } f_{g,total} \le f \le \mu f_g \tag{7.35}$$

where the limiting frequency for the total array is given by Eq. 7.31, with the total area of the array used instead of the individual panel area. The shaded area indicates the possible variation depending on whether the sound ray strikes a panel or empty space. Beranek (1992) published relative reflection data based on laboratory tests by Watters et al., (1963), which are shown in Fig. 7.12. In general a large number of small panels is preferable to a few large ones.

Bragg Imaging

Since individual reflectors have to be relatively large to reflect bass frequencies, they are used in groups to improve their low-frequency response (see Leonard, Delsasso, and Knudsen, 1964; or Beranek, 1992). This can be tricky because if they are not arranged in a single plane there can be destructive interference at certain combinations of frequency and angle

FIGURE 7.12 Scattering from Panel Arrays (Beranek, 1992)

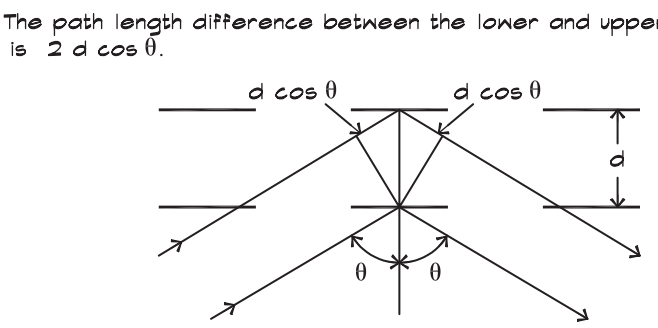


of incidence. When two planes of reflectors are employed, for a given separation distance there is a relationship between the angle of incidence and the frequency of cancellation of the reflected sound. This effect was used by Bragg to study the crystal structure of materials with x-rays. When sound is scattered from two reflecting planes, certain frequencies are missing in the reflected sound. This was one cause of the problems in Philharmonic Hall in New York (Beranek, 1996).

An illustration of this phenomenon, known as Bragg imaging, is shown in Fig. 7.13. When two rows of reflecting panels are placed one above the other, there is destructive interference between reflected sound waves when the combined path-length difference has the relationship

$$2\,\mathrm{d}\,\cos\,\theta = \frac{(2\,\mathrm{n}-1)\,\lambda}{2} \tag{7.36}$$

FIGURE 7.13 Geometry of Bragg Scattering from Rows of Parallel Reflectors



where λ = wavelength of the incident sound (m or ft)
d = perpendicular spacing between rows of reflectors (m or ft)
n = positive integer 1, 2, 3, . . . etc.
θ = angle of incidence and reflection with respect to the normal (rad or deg)

The use of slightly convex reflectors can help diffuse the sound energy and smooth out the interferences; however, stacked planes of reflecting panels can produce a loss of bass energy in localized areas of the audience.

Scattering from Curved Surfaces

When sound is scattered from a curved surface, the curvature induces diffusion of the reflected energy when the surface is convex, or focusing when it is concave. The attenuation associated with the curvature can be calculated using the geometry shown in Fig. 7.14. If we consider a rigid cylinder having a radius R, the loss in intensity is proportional to the ratio of the incident-to-reflected beam areas (Rindel, 1986). At the receiver, M, the sound energy is proportional to the width of the reflected beam tube $(a + a_2)\, d\beta$. If there were no curvature the beam width would be $(a_1 + a_2)\, d\beta_1$ at the image point M_1.

Accordingly the attenuation due to the curvature is

$$\Delta L_{curv} = -10 \log \frac{(a + a_2)\, d\beta}{(a_1 + a_2)\, d\beta_1} = -10 \log \frac{(a + a_2)(d\beta/d\beta_1)}{(a_1 + a_2)} \tag{7.37}$$

Using Fig. 7.14 we see that $a\, d\beta = a_1\, d\beta_1 = R\, d\phi \cos\theta$, and that $d\beta = d\beta_1 + 2\, d\phi$, from which it follows that

$$\frac{d\beta}{d\beta_1} = 1 + \frac{2\, a_1}{R \cos\theta} \tag{7.38}$$

and plugging this into Eq. 7.37 yields

$$\Delta L_{curv} = -10 \log \left| 1 + \frac{a^*}{R \cos\theta} \right| \tag{7.39}$$

where a^* is given in Eq. 7.24. For concave surfaces the same equation can be used with a negative value for R. Figure 7.15 shows the results for both convex and concave surfaces.

FIGURE 7.14 Geometry of the Reflection from a Curved Surface (Rindel, 1986)

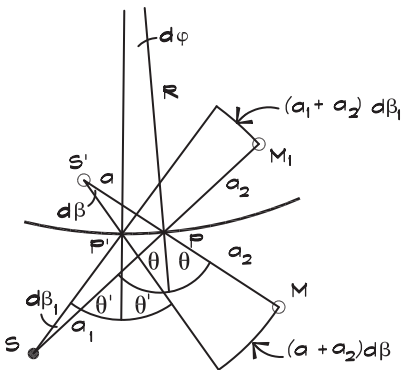


FIGURE 7.15 Attenuation or Gain Due to Curvature (Rindel, 1985)

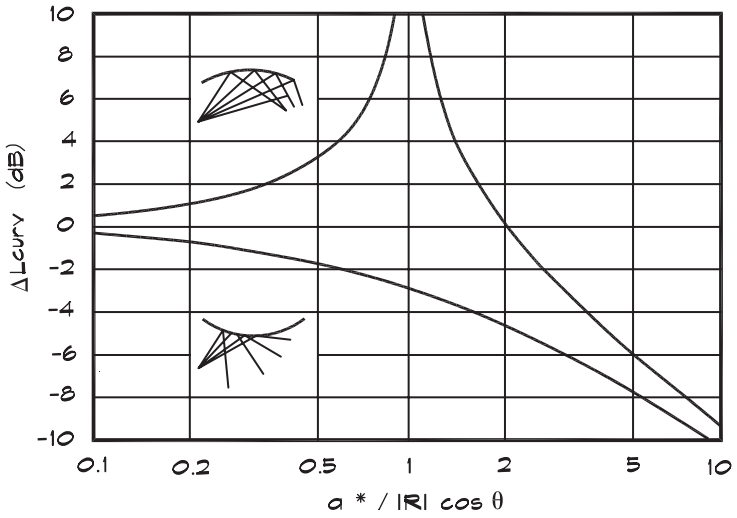


FIGURE 7.16 Calculated and Measured Values of ΔL_{curv} (Rindel, 1985)

Sketch				
Radius R	2.0 m	0.5 m	-2.0 m	-0.5 m
a * / R cos θ	0.5	2.0	0.5	-2.0
ΔL_{curv} (calc.)	-1.8 dB	-4.8 dB	3.0 dB	0.0 dB
ΔL_{curv} (meas.)	-2 ± 0.5 dB	-5 ± 1 dB	3 ± 1 dB	-1 ± 2 dB

This analysis assumes that both the source and the receiver are in a plane whose normal is the axis of the cylinder. If this is not the case, both a* and θ must be deduced from a normal projection onto that plane (Rindel, 1985). Where there is a double-curved surface with two radii of curvature, the attenuation term must be applied twice, using the appropriate projections onto the two normal planes of the surface.

Figure 7.16 gives the results of measurements using TDS on a small (1.4 m × 1.0 m) curved panel at a distance of 1 m for a zero angle of incidence over a frequency range of 3 to 19 kHz. The data also show the variation in the measured values, which Rindel (1985) attributes to diffraction effects.

Combined Effects

When sound is reflected from finite curved panels, the combined effects of distance, absorption, size, and curvature must be included. For an omnidirectional source, the level of the reflected sound relative to the direct sound is

$$L_{refl} - L_{dir} = \Delta L_{dist} + \Delta L_{abs} + \Delta L_{dif} + \Delta L_{curv} \tag{7.40}$$

where the absorption term will be addressed later in this chapter, and the distance term is

$$\Delta L_{dist} = 20 \log \frac{a_0}{a_1 + a_2} \tag{7.41}$$

The other terms have been treated earlier.

Whispering Galleries

If a source of sound is located in a circular space, very close to the outside wall, some of the sound rays strike the surface at a shallow angle and are reflected again and again, and so propagate within a narrow band completely around the room. A listener located on the opposite side of the space can clearly hear conversations that occur close to the outside wall. This phenomenon, which is called a whispering gallery since even whispered conversations are audible, occurs in circular or domed spaces such as the statuary gallery in the Capital building in Washington, DC.

7.3 ABSORPTION

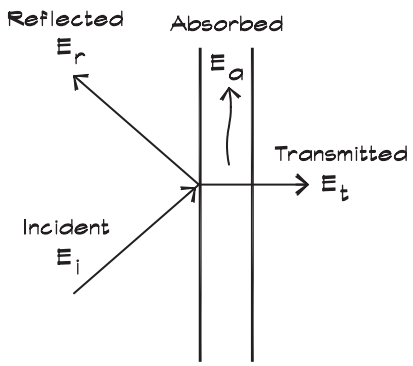
Reflection and Transmission Coefficients

When sound waves interact with real materials the energy contained in the incident wave is reflected, transmitted through the material, and absorbed within the material. The surfaces treated in this model are generally planar, although some curvature is tolerated as long as the radius of curvature is large when compared to a wavelength. The energy balance is illustrated in Fig. 7.17.

$$E_i = E_r + E_t + E_a \tag{7.42}$$

Since this analysis involves only the interaction at the boundary of the material, the difference between absorption, where energy is converted to heat, and transmission, where energy passes through the material, is not relevant. Both mechanisms are absorptive from the standpoint of the incident side because the energy is not reflected. Because we are only interested in the incident side of the boundary, we can combine the transmitted and

FIGURE 7.17 Interaction of Sound Waves with a Surface



absorbed energies. If we divide Eq. 7.42 by E_i,

$$1 = \frac{E_r}{E_i} + \frac{E_{t+a}}{E_i} \tag{7.43}$$

We can express each energy ratio as a coefficient of reflection or transmission. The fraction of the incident energy that is absorbed (or transmitted) at the surface boundary is the coefficient

$$\alpha_\theta = \frac{E_{t+a}}{E_i} \tag{7.44}$$

and the reflection coefficient is

$$\alpha_r = \frac{E_r}{E_i} \tag{7.45}$$

Substituting these coefficients into Eq. 7.43,

$$1 = \alpha_\theta + \alpha_r \tag{7.46}$$

The reflection coefficient can be expressed in terms of the complex reflection amplitude ratio $\boldsymbol{r}$ for pressure that was defined in Eq. 7.7

$$\alpha_r = \boldsymbol{r}^2 \tag{7.47}$$

and the absorption coefficient is

$$\alpha_\theta = 1 - \boldsymbol{r}^2 \tag{7.48}$$

The reflected energy is

$$E_r = (1 - \alpha_\theta)E_i \tag{7.49}$$

Impedance Tube Measurements

When a plane wave is normally incident on the boundary between two materials, 1 and 2, we can calculate the strength of the reflected wave from a knowledge of their impedances. (This solution was published by Rayleigh in 1896.) Following the approach taken in Eq. 7.3, the sound pressure from the incident and reflected waves is written as

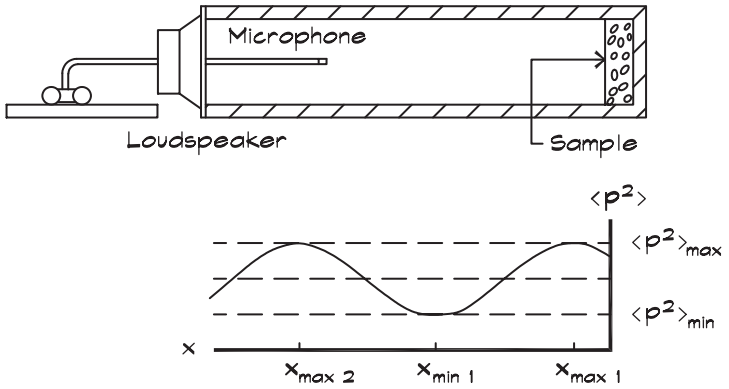
$$\mathbf{p}(x) = \mathbf{A}\, e^{j(\omega t - k x)} + \mathbf{B}\, e^{j(\omega t + k x)} \tag{7.50}$$

If we square and average this equation, we obtain the mean-squared acoustic pressure of a normally incident and reflected wave (Pierce, 1981)

$$\langle \mathbf{p}^2 \rangle = \frac{1}{2}\mathbf{A}^2\left[1 + \left|\boldsymbol{r}^2\right| + 2\left|\boldsymbol{r}\right|\cos\left(2 k x + \delta_r\right)\right] \tag{7.51}$$

where δ_r is the phase of $\boldsymbol{r}$. Equation 7.51 describes a standing wave and gives a method for measuring the normal-incidence absorption coefficient of a material placed in the end of a tube, called an *impedance tube*, pictured in Fig. 7.18.

FIGURE 7.18 Impedance Tube Measurements of the Absorption Coefficient



An incident and reflected wave combine to produce a standing wave in a tube. The pressure maximum and minimum are measured and their ratio determines the impedance.

The maximum value of the mean-squared pressure is $\frac{1}{2}\,\mathbf{A}^2\,[1 + |\,\boldsymbol{r}\,|]^2$, which occurs whenever $2\,\mathrm{k}\,\mathrm{x} + \delta_\mathrm{r}$ is an even multiple of π. The minimum is $\frac{1}{2}\,\mathbf{A}^2\,[1 - |\,\boldsymbol{r}\,|]^2$, which occurs at odd multiples of π. The ratio of the maximum-to-minimum pressures is an easily measured quantity called the *standing wave ratio*, s, which is usually obtained from its square

$$\mathrm{s}^2 = \frac{\langle \mathbf{p}^2 \rangle_{\max}}{\langle \mathbf{p}^2 \rangle_{\min}} = \left| \frac{\mathbf{A} + \mathbf{B}}{\mathbf{A} - \mathbf{B}} \right|^2 = \frac{[1 + |\,\boldsymbol{r}\,|]^2}{[1 - |\,\boldsymbol{r}\,|]^2} \tag{7.52}$$

The phase angle is

$$\delta_\mathrm{r} = -2\,\mathrm{k}\,\mathrm{x}_{\mathrm{max}\ 1} + 2\,\mathrm{m}\,\pi = -2\,\mathrm{k}\,\mathrm{x}_{\mathrm{min}\ 1} + (2\,\mathrm{n} + 1)\,\pi \tag{7.53}$$

where $\mathrm{x}_{\mathrm{min}\ 1}$ is the smallest distance to a minimum and $\mathrm{x}_{\mathrm{max}\ 1}$ is the smallest distance to a maximum, measured from the surface of the material. The numbers m and n are arbitrary integers, which do not affect the relative phase. Equation 7.52 can be solved for the magnitude and phase of the reflection amplitude ratio

$$\boldsymbol{r} = |\,\boldsymbol{r}\,|\,\mathrm{e}^{\mathrm{j}\,\delta_\mathrm{r}} \tag{7.54}$$

from which the normal incidence material impedance can be obtained.

$$\mathbf{z}_\mathrm{n} = \rho_0 \mathrm{c}_0 \frac{(1 + \boldsymbol{r})}{(1 - \boldsymbol{r})} \tag{7.55}$$

Oblique Incidence Reflections—Finite Impedance

When sound is obliquely incident on a surface having a finite impedance, the pressure of the incident and reflected waves is given by

$$\mathbf{p} = \mathbf{A}\left[\mathrm{e}^{\mathrm{j}\,\mathrm{k}\,(\mathrm{x}\,\cos\theta - \mathrm{y}\sin\theta)} + \boldsymbol{r}\,\mathrm{e}^{-\mathrm{j}\,\mathrm{k}\,(\mathrm{x}\cos\theta + \mathrm{y}\sin\theta)}\right] \tag{7.56}$$

and the velocity in the x direction at the boundary (x = 0) using Eq. 6.31 is

$$\mathbf{u}(\mathrm{x}) = \frac{\mathbf{A}}{\rho_0 \mathrm{c}_0}\left[\cos\theta\, \mathrm{e}^{-\mathrm{j\,k\,y\,sin}\,\theta} - \boldsymbol{r}\cos\theta\, \mathrm{e}^{-\mathrm{j\,k\,y\,sin}\,\theta}\right] \tag{7.57}$$

The normal specific acoustic impedance, expressed as the ratio of the pressure to the velocity at the surface is

$$\mathbf{z}_\mathrm{n} = \left(\frac{\mathbf{p}}{\mathbf{u}_\mathrm{x}}\right)_{\mathrm{x}=0} = \frac{\rho_0 \mathrm{c}_0}{\cos\theta}\frac{(1+\boldsymbol{r})}{(1-\boldsymbol{r})} \tag{7.58}$$

The reflection coefficient can then be written in terms of the boundary's specific acoustic impedance

$$\boldsymbol{r} = \frac{\mathbf{z} - \rho_0 \mathrm{c}_0}{\mathbf{z} + \rho_0 \mathrm{c}_0} \tag{7.59}$$

where $\mathbf{z} = \mathbf{z}_\mathrm{n}\cos\theta$ and $\mathbf{z}_\mathrm{n} = w_\mathrm{n} + \mathrm{j}\, x_\mathrm{n}$
$\mathbf{z}$ = complex specific acoustic impedance
= (pressure or force) / (particle or volume velocity)
w = specific acoustic resistance or real part of the impedance
x = specific acoustic reactance or the imaginary part of the impedance
- when positive it is mass like and when negative it is stiffness like
$\rho_0 \mathrm{c}_0$ = characteristic acoustic resistance of the incident medium
(about 412 Ns/m ^{3} - mks rayls in air)
$\mathrm{j} = \sqrt{-1}$

Now these relationships contain a good deal of information about the reflection process. When $|\mathbf{z}| \gg \rho_0 \mathrm{c}_0$ the reflection coefficient approaches a value of + 1, there is perfect reflection, and the reflected wave is in phase with the incident wave. If $|\mathbf{z}| \ll \rho_0 \mathrm{c}_0$, the boundary is resilient like the open end of a tube, and the value of $\boldsymbol{r}$ approaches −1. Here the reflected wave is 180° out of phase with the incident wave and there is cancellation. When $|\mathbf{z}| = \rho_0 \mathrm{c}_0$, no sound is reflected.

For any finite value of $\mathbf{z}_\mathrm{n}$, as θ approaches $\pi/2$, the incident wave grazes over the boundary and the value of $\boldsymbol{r}$ approaches − 1. Under this condition the incident and reflected waves are out of phase and interfere with one another. This is an explanation of the ground effect, which was discussed previously. Note that in both cases—that is, when $\boldsymbol{r}$ is either ± 1—there is no sound absorption by the surface. The $|\boldsymbol{r}| = -1$ case is rarely encountered in architectural acoustics and only over limited frequency ranges (Kuttruff, 1973).

Reflection and transmission coefficients can also be written in terms of the normal acoustic impedance of a material. The energy reflection coefficient using Eq. 7.59 is

$$\alpha_\mathrm{r} = \left|\frac{\left(\mathbf{z}_\mathrm{n}/\rho_0 \mathrm{c}_0\right)\cos\theta - 1}{\left(\mathbf{z}_\mathrm{n}/\rho_0 \mathrm{c}_0\right)\cos\theta + 1}\right|^2 \tag{7.60}$$

and in terms of the real and imaginary components of the impedance,

$$\alpha_r = \frac{\left(w_n \cos\theta - \rho_0 c_0\right)^2 + x_n^2 \cos^2\theta}{\left(w_n \cos\theta + \rho_0 c_0\right)^2 + x_n^2 \cos^2\theta} \tag{7.61}$$

The absorption coefficient is set equal to the absorption/transmission coefficient given in Eq. 7.44, since it is defined at a surface where it does not matter whether the energy is transmitted through the material or absorbed within the material, as long as it is not reflected back. The specular absorption coefficient is

$$\alpha_\theta = 1 - \left| \frac{\left(\mathbf{z}_n/\rho_0 c_0\right)\cos\theta - 1}{\left(\mathbf{z}_n/\rho_0 c_0\right)\cos\theta + 1} \right|^2 \tag{7.62}$$

which in terms of its components is

$$\alpha_\theta = \frac{4\,\rho_0 c_0 w_n \cos\theta}{\left(w_n \cos\theta + \rho_0 c_0\right)^2 + x_n^2 \cos^2\theta} \tag{7.63}$$

For most architectural situations the incident conducting medium is air; however, it could be any material with a characteristic resistance. Since solid surfaces such as walls or absorptive panels have a resistance $w_n \gg \rho_0 c_0$, the magnitude of the absorption coefficient yields a maximum value when $w_n \cos\theta_i = \rho_0 c_0$. For normal incidence, when $\mathbf{z}_n = \rho_0 c_0$, the transmission coefficient is unity as we would expect.

Figure 7.19 shows the behavior of a typical absorption coefficient with angle of incidence. As the angle of incidence increases, the apparent depth of the material increases, thereby increasing the absorption. At very high angles of incidence there is no longer a velocity component into the material so the coefficient drops off rapidly.

FIGURE 7.19 Absorption Coefficient as a Function of Angle of Incidence for a Porous Absorber (Benedetto and Spagnolo, 1985)

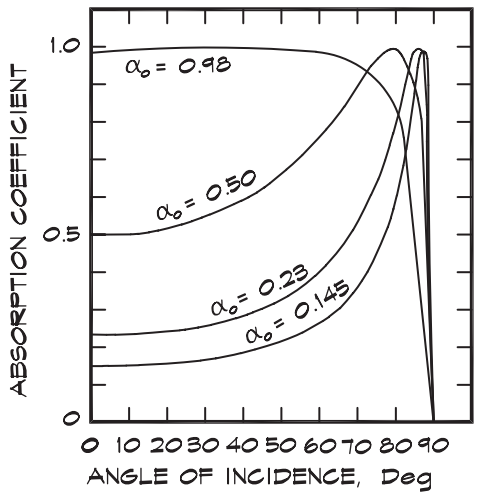
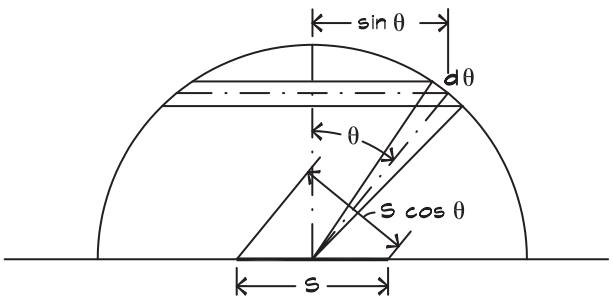


FIGURE 7.20 Geometry of the Diffuse Field Absorption Coefficient Calculation (Cremer et al., 1982)



Calculated Diffuse Field Absorption Coefficients

In Eq. 7.63 we saw that we could write the absorption coefficient as a function of the angle of incidence, in terms of the complex impedance. Although direct-field absorption coefficients are useful for gaining an understanding of the physics of the absorption process, for most architectural applications a measurement is made of the diffuse-field absorption coefficient. Recall that a diffuse field implies that incident sound waves come from any direction with equal probability. The diffuse-field absorption coefficient is the average of the coefficient α_θ, taken over all possible angles of incidence. The geometry is shown in Fig. 7.20. The energy from a uniformly radiating hemisphere that is incident on the surface **S** is proportional to the area that lies between the angle $\theta - \Delta\theta/2$ and $\theta + \Delta\theta/2$. The fraction of the total energy coming from this angle is

$$\frac{\mathrm{dE}}{\mathrm{E}} = \frac{2\pi \mathrm{r} \sin\theta \, \mathrm{r} \, \mathrm{d}\theta}{4\pi \mathrm{r}^2} = \frac{1}{2} \sin\theta \, \mathrm{d}\theta \tag{7.64}$$

and the total power sound absorbed by a projected area S cos θ is

$$\mathrm{W} = \mathrm{E}\,\mathrm{c}\,\mathrm{S} \int_0^{\pi/2} \alpha_\theta \sin\theta \quad \cos\theta \, \mathrm{d}\theta \tag{7.65}$$

The total incident power from all angles is the value of Eq. 7.65, with a perfectly absorptive material ($\alpha_\theta = 1$). The average absorption coefficient is the ratio of the absorbed to the total power

$$\alpha = \frac{\int_0^{\pi/2} \alpha_\theta \sin\theta \quad \cos\theta \, \mathrm{d}\theta}{\int_0^{\pi/2} \sin\theta \quad \cos\theta \, \mathrm{d}\theta} \tag{7.66}$$

which can be simplified to (Paris, 1928)

$$\alpha = 2 \int_0^{\pi/2} \alpha_\theta \sin\theta \quad \cos\theta \, \mathrm{d}\theta \tag{7.67}$$

Here the sine term is the probability that energy will originate at a given angle and the cosine term is the projection of the receiving area.

Measurement of Diffuse Field Absorption Coefficients

Although diffuse-field absorption coefficients can be calculated from impedance tube data, more often they are measured directly in a reverberant space. Values of α are published for a range of frequencies between 125 Hz and 4 kHz. Each coefficient represents the diffuse-field absorption averaged over a band of frequencies one octave wide. Occasionally absorption data are required for calculations beyond this range. In these cases estimates can be made from impedance tube data, by extrapolation from known data, or by calculating the values from first principles.

Some variability arises in the measurement of the absorption coefficient. The diffuse-field coefficient is, in theory, always less than or equal to a value of one. In practice, when the reverberation time method discussed in Chapt. 8, is employed, values of α greater than 1 are sometimes measured. This normally is attributed to diffraction, the lack of a perfectly diffuse field in the measuring room, and edge conditions. At low frequencies diffraction seems to be the main cause (Beranek, 1971). Since it is easier and more consistent to measure the absorption using the reverberation time method, this is the value that is found in the published literature.

Diffuse-field measurements of the absorption coefficient are carried out in a reverberation chamber, which is a room with little or no absorption. Data are taken with and without the panel under test in the room and the resulting reverberation times are used to calculate the absorptive properties of the material. The test standard, ASTM C423, specifies several mountings as shown in Fig 7.21. Mountings A, B, D, and E are used for most prefabricated products. Mounting F is for duct liners and C is used for specialized applications. When data are reported, the test mounting method must also be included since the airspace behind the material greatly affects the results. The designation E-400, for example, indicates that mounting E was used and there was a 400 mm (16”) airspace behind the test sample.

Noise Reduction Coefficient (NRC)

Absorptive materials, such as acoustical ceiling tile, wall panels, and other porous absorbers are often characterized by their noise reduction coefficient, which is the average diffuse field absorption coefficient over the speech frequencies, 250 Hz to 2 kHz, rounded to the nearest 0.05.

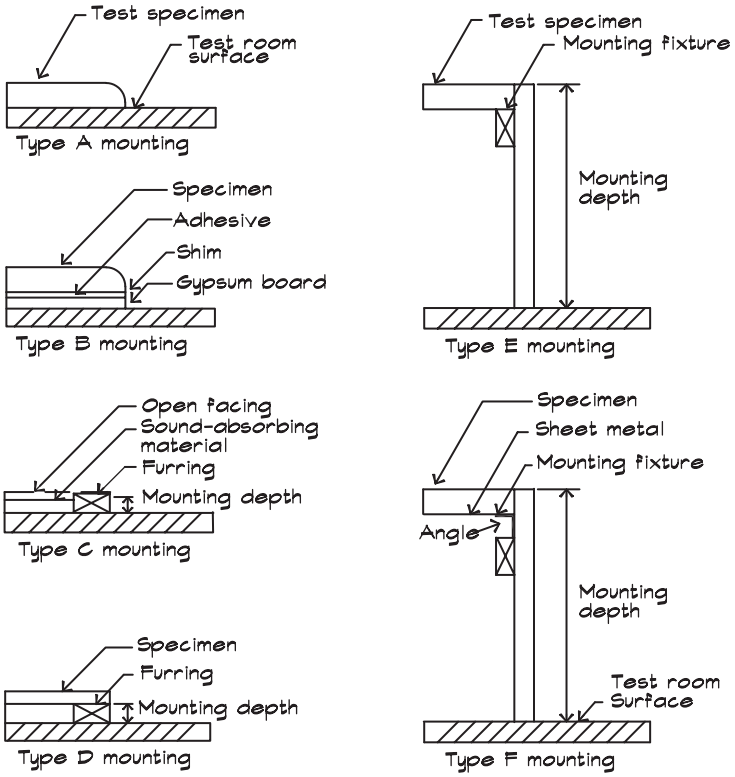
$$\mathrm{NRC} = \frac{1}{4}\left(\alpha_{250} + \alpha_{500} + \alpha_{1\,\mathrm{k}} + \alpha_{2\,\mathrm{k}}\right) \tag{7.68}$$

Although these single-number metrics are useful as a means of getting a general idea of the effectiveness of a particular material, for critical applications calculations should be carried out over the entire frequency range of interest.

Absorption Data

Table 7.1 shows a representative sample of measured absorption data. The list is by no means complete but care has been taken to include a reasonable sample of different types of materials. When layering materials or when using them in a manner that is not representative

FIGURE 7.21 Laboratory Absorption Test Mountings



of the measured data, some adjustments may have to be made to account for different air cavity depths or mounting methods.

Occasionally it is necessary to estimate the absorption of materials beyond the range of measured data. Most often this occurs in the 63 Hz octave band, but sometimes occurs at lower frequencies. Data generally are not measured in this frequency range because of the size of reverberant chamber necessary to meet the diffuse field requirements. In these cases it is particularly important to consider the contributions to the absorption of the structural elements behind any porous panels.

Layering Absorptive Materials

It is the rule rather than the exception that acoustical materials are layered in real applications. For example a 25 mm (1 in) thick cloth-wrapped fiberglass material might be applied over a 16 mm (5/8 in) thick gypsum board wall. A detailed mathematical analysis of the impedance of the composite material is beyond the scope of a typical architectural project, and when one seeks the absorption coefficient from tables such as those in Table 7.1, one finds data on the panel, tested in an A-mounting condition, and data on the gypsum board wall, but no data on the combination.

If the panel data were used without consideration of the backing, the listed value at 125 Hz would suggest that there would be a decrease in absorption from the application of the panel relative to the drywall alone. This is due to the lower absorption coefficient that comes about from the test mounting method (on concrete), rather than from the panel itself.

TABLE 7.1 Absorption Coefficients of Common Materials

Material	Mount	Frequency, Hz					
		125	250	500	1k	2k	4k
Walls							
Glass, 1/4", heavy plate		0.18	0.06	0.04	0.05	0.02	0.02
Glass, 3/32", ordinary window		0.55	0.25	0.18	0.12	0.07	0.04
Gypsum board, 1/2", on 2×4 studs		0.29	0.10	0.05	0.04	0.07	0.09
Plaster, 7/8", gypsum or lime, on brick		0.013	0.015	0.02	0.03	0.04	0.05
Plaster, on concrete block		0.12	0.09	0.07	0.05	0.05	0.04
Plaster, 7/8", on lath		0.14	0.10	0.06	0.04	0.04	0.05
Plaster, 7/8", lath on studs		0.30	0.15	0.10	0.05	0.04	0.05
Plywood, 1/4", 3" air space, 1" batt,		0.60	0.30	0.10	0.09	0.09	0.09
Soundblox, type B, painted		0.74	0.37	0.45	0.35	0.36	0.34
Wood panel, 3/8", 3-4" air space		0.30	0.25	0.20	0.17	0.15	0.10
Concrete block, unpainted		0.36	0.44	0.51	0.29	0.39	0.25
Concrete block, painted		0.10	0.05	0.06	0.07	0.09	0.08
Concrete poured, unpainted		0.01	0.01	0.02	0.02	0.02	0.03
Brick, unglazed, unpainted		0.03	0.03	0.03	0.04	0.05	0.07
Wood paneling, 1/4",		0.42	0.21	0.10	0.08	0.06	0.06
with airspace behind							
Wood, 1", paneling with airspace behind		0.19	0.14	0.09	0.06	0.06	0.05
Shredded-wood fiberboard, 2", on concrete	A	0.15	0.26	0.62	0.94	0.64	0.92
Carpet, heavy, on 5/8-in perforated mineral fiberboard		0.37	0.41	0.63	0.85	0.96	0.92
Brick, unglazed, painted	A	0.01	0.01	0.02	0.02	0.02	0.03
Light velour, 10 oz per sq yd, hung straight, in contact with wall		0.03	0.04	0.11	0.17	0.24	0.35
Medium velour, 14 oz per sq yd, draped to half area		0.07	0.31	0.49	0.75	0.70	0.60
Heavy velour, 18 oz per sq yd, draped to half area		0.14	0.35	0.55	0.72	0.70	0.65

continued

TABLE 7.1 Absorption Coefficients of Common Materials, (*Continued*)

Material	Mount	Frequency, Hz					
		125	250	500	1k	2k	4k
Floors							
Floors, concrete or terrazzo	A	0.01	0.01	0.015	0.02	0.02	0.02
Floors, linoleum, vinyl on concrete	A	0.02	0.03	0.03	0.03	0.03	0.02
Floors, linoleum, vinyl on subfloor		0.02	0.04	0.05	0.05	0.10	0.05
Floors, wooden		0.15	0.11	0.10	0.07	0.06	0.07
Floors, wooden platform w/airspace		0.40	0.30	0.20	0.17	0.15	0.10
Carpet, heavy on concrete	A	0.02	0.06	0.14	0.57	0.60	0.65
Carpet, on 40 oz (1.35 kg/sq m) pad	A	0.08	0.24	0.57	0.69	0.71	0.73
Indoor-outdoor carpet	A	0.01	0.05	0.10	0.20	0.45	0.65
Wood parquet in asphalt on concrete	A	0.04	0.04	0.07	0.06	0.06	0.07
Ceilings							
Acoustical coating K-13, 1”	A	0.08	0.29	0.75	0.98	0.93	0.96
1.5”	A	0.16	0.50	0.95	1.06	1.00	0.97
2”	A	0.29	0.67	1.04	1.06	1.00	0.97
Acoustical coating K-13 “fc” 1”	A	0.12	0.38	0.88	1.16	1.15	1.12
Glass-fiber roof fabric, 12 oz/yd		0.65	0.71	0.82	0.86	0.76	0.62
Glass-fiber roof fabric, 37.5 oz/yd		0.38	0.23	0.17	0.15	0.09	0.06
Acoustical Tile							
Standard mineral fiber, 5/8”	E400	0.68	0.76	0.60	0.65	0.82	0.76
Standard mineral fiber, 3/4”	E400	0.72	0.84	0.70	0.79	0.76	0.81
Standard mineral fiber, 1”	E400	0.76	0.84	0.72	0.89	0.85	0.81
Energy mineral fiber, 5/8”	E400	0.70	0.75	0.58	0.63	0.78	0.73
Energy mineral fiber, 3/4”	E400	0.68	0.81	0.68	0.78	0.85	0.80
Energy mineral fiber, 1”	E400	0.74	0.85	0.68	0.86	0.90	0.79
Film faced fiberglass, 1”	E400	0.56	0.63	0.69	0.83	0.71	0.55
Film faced fiberglass, 2”	E400	0.52	0.82	0.88	0.91	0.75	0.55
Film faced fiberglass, 3”	E400	0.64	0.88	1.02	0.91	0.84	0.62

continued

TABLE 7.1 Absorption Coefficients of Common Materials, (*Continued*)

Material	Mount	Frequency, Hz					
		125	250	500	1k	2k	4k
Glass Cloth Acoustical Ceiling Panels							
Fiberglass tile, 3/4”	E400	0.74	0.89	0.67	0.89	0.95	1.07
Fiberglass tile, 1”	E400	0.77	0.74	0.75	0.95	1.01	1.02
Fiberglass tile, 1 1/2”	E400	0.78	0.93	0.88	1.01	1.02	1.00
Seats and Audience							
Audience in upholstered seats		0.39	0.57	0.80	0.94	0.92	0.87
Unoccupied well-upholstered seats		0.19	0.37	0.56	0.67	0.61	0.59
Unoccupied leather covered seats		0.19	0.57	0.56	0.67	0.61	0.59
Wooden pews, occupied		0.57	0.44	0.67	0.70	0.80	0.72
Leather-covered upholstered seats, unoccupied		0.44	0.54	0.60	0.62	0.58	0.50
Congregation, seated in wooden pews		0.57	0.61	0.75	0.86	0.91	0.86
Chair, metal or wood seat, unoccupied		0.15	0.19	0.22	0.39	0.38	0.30
Students, informally dressed, seated in tablet-arm chairs		0.30	0.41	0.49	0.84	0.87	0.84
Duct Liners							
1/2”		0.11	0.51	0.48	0.70	0.88	0.98
1”		0.16	0.54	0.67	0.85	0.97	1.01
1 1/2”		0.22	0.73	0.81	0.97	1.03	1.04
2”		0.33	0.90	0.96	1.07	1.07	1.09
Aeroflex Type 150, 1”	F	0.13	0.51	0.46	0.65	0.74	0.95
Aeroflex Type 150, 2”	F	0.25	0.73	0.94	1.03	1.02	1.09
Aeroflex Type 200, 1/2”	F	0.10	0.44	0.29	0.39	0.63	0.81
Aeroflex Type 200, 1”	F	0.15	0.59	0.53	0.78	0.85	1.00
Aeroflex Type 200, 2”	F	0.28	0.81	1.04	1.10	1.06	1.09
Aeroflex Type 300, 1/2”	F	0.09	0.43	0.31	0.43	0.66	0.98
Aeroflex Type 300, 1”	F	0.14	0.56	0.63	0.82	0.99	1.04
Aeroflex Type 150, 1”	A	0.06	0.24	0.47	0.71	0.85	0.97
Aeroflex Type 150, 2”	A	0.20	0.51	0.88	1.02	0.99	1.04
Aeroflex Type 300, 1”	A	0.08	0.28	0.65	0.89	1.01	1.04

continued

TABLE 7.1 Absorption Coefficients of Common Materials, (*Continued*)

Material	Mount	Frequency, Hz					
		125	250	500	1k	2k	4k
Building Insulation - Fiberglass							
3.5" (R-11) (insulation exposed to sound)	A	0.34	0.85	1.09	0.97	0.97	1.12
6" (R-19) (insulation exposed to sound)	A	0.64	1.14	1.09	0.99	1.00	1.21
3.5" (R11) (FRK facing exposed to sound)	A	0.56	1.11	1.16	0.61	0.40	0.21
6" (R-19) (FRK facing exposed to sound)	A	0.94	1.33	1.02	0.71	0.56	0.39
Fiberglass Board (FB)							
FB, 3lb/ft^3, 1" thick	A	0.03	0.22	0.69	0.91	0.96	0.99
FB, 3 lb/ft^3, 2" thick	A	0.22	0.82	1.21	1.10	1.02	1.05
FB, 3 lb/ft^3, 3" thick	A	0.53	1.19	1.21	1.08	1.01	1.04
FB, 3 lb/ft^3, 4" thick	A	0.84	1.24	1.24	1.08	1.00	0.97
FB, 3 lb/ft^3, 1" thick	E400	0.65	0.94	0.76	0.98	1.00	1.14
FB, 3 lb/ft^3, 2" thick	E400	0.66	0.95	1.06	1.11	1.09	1.18
FB, 3 lb/ft^3, 3" thick	E400	0.66	0.93	1.13	1.10	1.11	1.14
FB, 3 lb/ft^3, 4" thick	E400	0.65	1.01	1.20	1.14	1.10	1.16
FB, 6 lb/ft^3, 1" thick	A	0.08	0.25	0.74	0.95	0.97	1.00
FB, 6 lb/ft^3, 2" thick	A	0.19	0.74	1.17	1.11	1.01	1.01
FB, 6 lb/ft^3, 3" thick	A	0.54	1.12	1.23	1.07	1.01	1.05
FB, 6 lb/ft^3, 4" thick	A	0.75	1.19	1.17	1.05	0.97	0.98
FB, 6 lb/ft^3, 1" thick	E400	0.68	0.91	0.78	0.97	1.05	1.18
FB, 6 lb/ft^3, 2" thick	E400	0.62	0.95	0.98	1.07	1.09	1.22
FB, 6 lb/ft^3, 3" thick	E400	0.66	0.92	1.11	1.12	1.10	1.19
FB, 6 lb/ft^3, 4" thick	E400	0.59	0.91	1.15	1.11	1.11	1.19
FB, FRK faced, 1" thick	A	0.12	0.74	0.72	0.68	0.53	0.24
FB, FRK faced, 2" thick	A	0.51	0.65	0.86	0.71	0.49	0.26
FB, FRK faced, 3" thick	A	0.84	0.88	0.86	0.71	0.52	0.25
FB, FRK faced, 4" thick	A	0.88	0.90	0.84	0.71	0.49	0.23
FB, FRK faced, 1" thick	E400	0.48	0.60	0.80	0.82	0.52	0.35
FB, FRK faced, 2" thick	E400	0.50	0.61	0.99	0.83	0.51	0.35
FB, FRK faced, 3" thick	E400	0.59	0.64	1.09	0.81	0.50	0.33
FB, FRK faced, 4" thick	E400	0.61	0.69	1.08	0.81	0.48	0.34
Miscellaneous							
Musician (per person), with instrument		4.0	8.5	11.5	14.0	15.0	12.0
Air, Sabins per 1000 cubic feet @ 50% RH					0.9	2.3	7.2

In cases where materials are applied in ways that differ from the manner in which they were tested, estimates must be made based on the published absorptive properties of the individual elements. For example, a drywall wall has an absorptive coefficient of 0.29 at 125 Hz since it is acting as a panel absorber, having a resonant frequency of about 55 Hz. A one-inch (25 mm) thick fiberglass panel has an absorption coefficient of 0.03 at 125 Hz since it is measured in the A-mounting position. When a panel is mounted on drywall, the low-frequency sound passes through the fiberglass panel and interacts with the drywall surface. Assuming the porous material does not significantly increase the mass of the wall surface, the absorption at 125 Hz should be at least 0.29, and perhaps a little more due to the added flow resistance of the fiberglass. Consequently when absorptive materials are layered we must consider the combined result, rather than the absorption coefficient of only the surface material alone.

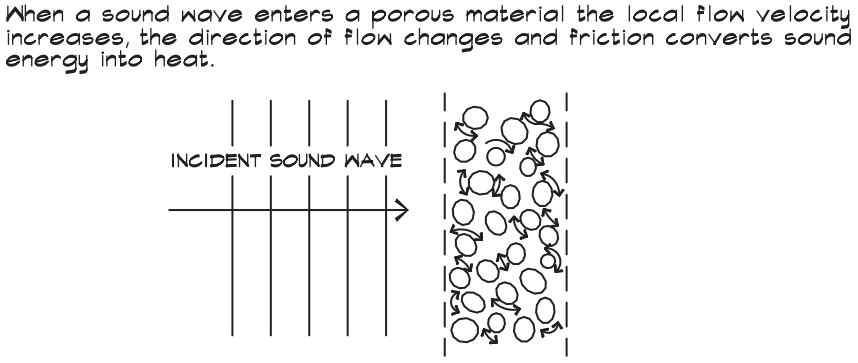
7.4 ABSORPTION MECHANISMS

Absorptive materials used in architectural applications tend to fall into three categories: porous absorbers, panel absorbers, and resonant absorbers. Of these, the porous absorbers are the most frequently encountered and include fiberglass, mineral fiber products, fiberboard, pressed wood shavings, cotton, felt, open-cell neoprene foam, carpet, sintered metal, and many other products. Panel absorbers are nonporous lightweight sheets, solid or perforated, that have an air cavity behind them, which may be filled with an absorptive material such as fiberglass. Resonant absorbers can be lightweight partitions vibrating at their mass-air-mass resonance or they can be Helmholtz resonators or other similar enclosures, which absorb sound in the frequency range around their resonant frequency. They also may be filled with absorbent porous materials.

Porous Absorbers

Several mechanisms contribute to the absorption of sound by porous materials. Air motion induced by the sound wave occurs in the interstices between fibers or particles. The movement of the air through narrow constrictions, as illustrated in Fig. 7.22, produces losses of momentum due to viscous drag (friction) as well as changes in direction. This accounts for most of the high-frequency losses. At low frequencies absorption occurs because fibers are relatively efficient conductors of heat. Fluctuations in pressure and density are isothermal, since thermal equilibrium is restored so rapidly. Temperature increases in the gas cause heat

FIGURE 7.22 Viscous Drag Mechanism of Absorption in Porous Materials



to be transported away from the interaction site to dissipate. Little attenuation seems to occur as a result of induced motion of the fibers (Mechel and Ver, 1992).

A lower (isothermal) sound velocity within a porous material also contributes to absorption. Friction forces and direction changes slow down the passage of the wave, and the isothermal nature of the process leads to a different equation of state. When sound waves travel parallel to the plane of the absorber some of the wave motion occurs within the absorber. Waves near the surface are diffracted, drawn into the material, due to the lower sound velocity.

In general, porous absorbers are too complicated for their precise impedances to be predicted from first principles. Rather, it is customary to measure the flow resistance, r_f, of the bulk material to determine the resistive component of the impedance. The bulk flow resistance is defined as the ratio between the pressure drop ΔP across the absorbing material and the steady velocity u_s of the air passing through the material.

$$r_f = -\frac{\Delta P}{u_s} \tag{7.69}$$

Since this is dependent on the thickness of the absorber it is not a fundamental property of the material. The material property is the *specific flow resistance*, r_s, which is independent of the thickness.

$$r_s = -\frac{1}{u_s}\frac{\Delta P}{\Delta x} = \frac{r_f}{\Delta x} \tag{7.70}$$

Flow resistance can be measured (Ingard, 1994) using a weighted piston as in Fig. 7.23. When the piston reaches its steady velocity, the resistance can be determined by measuring the time it takes for the piston to travel a given distance and the mass of the piston. Flow resistance is expressed in terms of the pressure drop in Newtons per square meter divided by the velocity in meters per second and is given in units of mks rayls, the same unit as the specific acoustic impedance. The specific flow resistance has units of mks rayls/m.

FIGURE 7.23 A Simple Device to Measure the Flow Resistance of a Porous Material (Ingard, 1994)

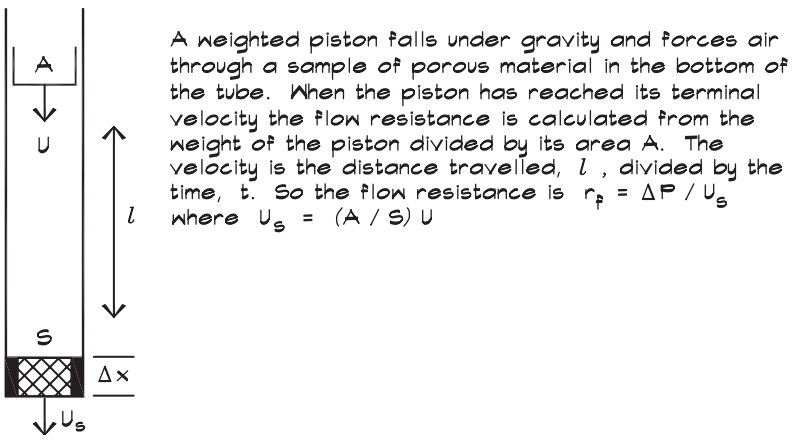
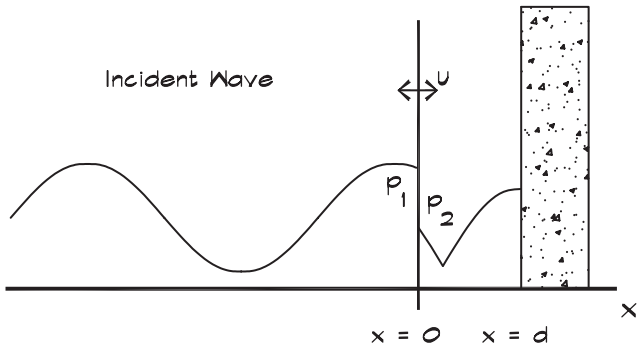


FIGURE 7.24 Geometry of a Spaced Porous Absorber (Kuttruff, 1973)



The porous absorber does not move but air flows through it, creating a standing wave in the cavity behind.

Spaced Porous Absorbers—Normal Incidence, Finite Impedance

If a thin porous absorber is positioned such that it has an airspace behind it, the composite impedance at the surface of the material and thus the absorption coefficient, is influenced by the backing. Figure 7.24 shows a porous absorber located a distance d away from a solid wall. The flow resistance, which is approximately the resistive component of the impedance, is the difference in pressure across the material divided by the velocity through the material.

$$r_f = \frac{\mathbf{p}_1 - \mathbf{p}_2}{\mathbf{u}} \tag{7.71}$$

where $\mathbf{p}_1$ is the pressure on the left side of the sheet and $\mathbf{p}_2$ is the pressure just to the right of the sheet. In this analysis it is assumed that the resistance is the same for steady and alternating flow. The velocity on either side of the sheet is the same due to conservation of mass. Equation 7.71 can be written in terms of the impedance at the surface of the absorber (at $x = 0$) on either side of the sheet.

$$r_f = \mathbf{z}_1 - \mathbf{z}_2 \tag{7.72}$$

which implies that the impedance of the composite sheet plus the air backing is the sum of the sheet flow resistance and the impedance of the cavity behind the absorber.

$$\mathbf{z}_1 = r_f + \mathbf{z}_2 \tag{7.73}$$

To calculate the impedance of the air cavity for a normally incident sound wave we write the equations for a rightward moving wave and the reflected leftward moving wave, assuming perfect reflection from the wall.

$$\begin{aligned} \mathbf{p}_2(x) &= \mathbf{A}\left[e^{-j\,k(x-d)} + e^{j\,k(x-d)}\right] \\ &= 2\,\mathbf{A}\cos\left[k\,(x-d)\right] \end{aligned} \tag{7.74}$$

The velocity in the cavity is

$$\begin{aligned}\mathbf{u}_2(x) &= \frac{\mathbf{A}}{\rho_0 c_0}\left[e^{-j\,k\,(x-d)} - e^{j\,k\,(x-d)}\right] \\ &= -\frac{2\,j\,\mathbf{A}}{\rho_0 c_0}\sin\left[k\,(x-d)\right]\end{aligned} \tag{7.75}$$

The ratio of the pressure to the velocity at the sheet surface (x = 0) is the impedance of the cavity

$$\mathbf{z}_2 = \left(\frac{\mathbf{p}_2}{\mathbf{u}_2}\right)_{x=0} = -j\,\rho_0 c_0 \cot(k\,d) \tag{7.76}$$

The normal impedance of the porous absorber and the air cavity is

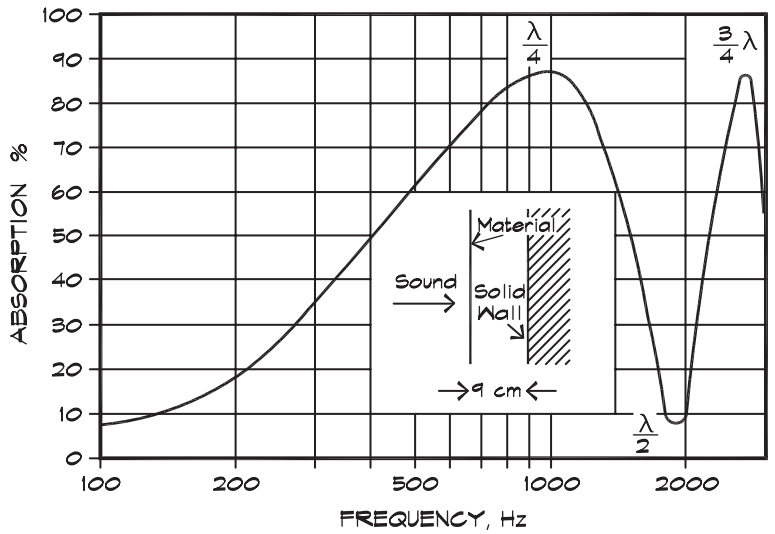
$$\mathbf{z}_n = r_f - j\,\rho_0 c_0 \cot(k\,d) \tag{7.77}$$

By plugging this expression into Eq. 7.63 we get the absorption coefficient for normal incidence (Kuttruff, 1973)

$$\alpha_n = 4\left\{\left[\sqrt{\frac{r_f}{\rho_0 c_0}} + \sqrt{\frac{\rho_0 c_0}{r_f}}\right]^2 + \frac{\rho_0 c_0}{r_f}\cot^2\left(\frac{2\pi\,f\,d}{c_0}\right)\right\}^{-1} \tag{7.78}$$

Figure 7.25 (Ginn, 1978) shows the behavior of this equation with frequency for a flow resistance $r_f \cong 2\rho_0 c_0$.

FIGURE 7.25 Absorptive Material Near a Hard Surface (Ginn, 1978)



Absorption of freely suspended, thin porous material arranged parallel to a plane hard surface.

A thin porous absorber located at multiples of a quarter wavelength from a reflecting surface is in an optimum position to absorb sound since the particle velocity is at a maximum at this point. An absorber located at multiples of one-half wavelength from a wall is not particularly effective since the particle velocity is low. Thin curtains are not good broadband absorbers unless there is considerable (usually 100%) gather or unless there are several curtains hung one behind another. Note that in real rooms it is rare to encounter a condition of purely normal incidence. For diffuse fields the phase interference patterns are much less pronounced than those shown in Fig. 7.25.

Porous Absorbers with Internal Losses—Normal Incidence

When there are internal losses that attenuate the sound as it passes through a material, the attenuation can be written as an exponentially decaying sinusoid

$$\mathbf{p}\,(\mathrm{x}) = \mathbf{A}\,\mathrm{e}^{\mathrm{j}\omega\mathrm{t}}\,\mathrm{e}^{-\mathrm{j}\boldsymbol{q}\mathrm{x}} \tag{7.79}$$

where $\boldsymbol{q} = \delta + \mathrm{j}\,\beta$ is the complex *propagation constant* within the absorbing material. It is much like the wave number in that its real part, δ, is close to ω/c. However, it has an imaginary part, β, which is the attenuation constant, in nepers/meter, of the sound passing through an absorber. To convert nepers per meter to dB/meter, multiply nepers by 8.69.

For a thick porous absorber with a characteristic wave impedance $\mathbf{z}_\mathrm{w}$, we write the equations for a normally incident and reflected plane wave with losses

$$\mathbf{p} = \mathbf{A}\,\mathrm{e}^{\mathrm{j}\boldsymbol{q}\mathrm{x}} + \mathbf{B}\,\mathrm{e}^{-\mathrm{j}\boldsymbol{q}\mathrm{x}} \tag{7.80}$$

and the particle velocity is

$$\mathbf{u} = \frac{1}{\mathbf{z}_\mathrm{w}}\left[\mathbf{A}\,\mathrm{e}^{\mathrm{j}\boldsymbol{q}\mathrm{x}} - \mathbf{B}\,\mathrm{e}^{-\mathrm{j}\boldsymbol{q}\mathrm{x}}\right] \tag{7.81}$$

We use the indices 1 and 2 for the left- and right-hand sides of the material and make the end of the material $\mathrm{x} = 0$ and the beginning of the material $\mathrm{x} = -\mathrm{d}$. The incident wave is moving in the positive x direction. At $\mathrm{x} = 0$,

$$\mathbf{p}_2 = \mathbf{A} + \mathbf{B} \tag{7.82}$$

and

$$\mathbf{u}_2 = \frac{1}{\mathbf{z}_\mathrm{w}}(\mathbf{A} - \mathbf{B}) \tag{7.83}$$

Solving for the amplitudes A and B,

$$\mathbf{A} = (\mathbf{p}_2 + \mathbf{z}_\mathrm{w}\,\mathbf{u}_2)\,/\,2 \tag{7.84}$$

$$\mathbf{B} = (\mathbf{p}_2 - \mathbf{z}_\mathrm{w}\,\mathbf{u}_2)\,/\,2 \tag{7.85}$$

Plugging these into Eqs. 7.80 and 7.81 at $\mathrm{x} = -\mathrm{d}$,

$$\mathbf{p}_1 = \mathbf{p}_2 \cos(\boldsymbol{q}\,\mathrm{d}) - \mathrm{j}\,\mathbf{z}_\mathrm{w}\,\mathbf{u}_2 \sin(\boldsymbol{q}\,\mathrm{d}) \tag{7.86}$$

and

$$\mathbf{u}_1 = \left(\frac{-\mathrm{j}}{\mathbf{z}_\mathrm{w}}\right) \mathbf{p}_2 \sin(\boldsymbol{q}\,\mathrm{d}) + \mathbf{u}_2 \cos(\boldsymbol{q}\,\mathrm{d}) \tag{7.87}$$

The ratio of these two equations yields the input impedance of the absorbing surface in terms of the characteristic wave impedance of the material and its propagation constant, and the back impedance $\mathbf{z}_2$ of the surface behind the absorber.

$$\mathbf{z}_1 = \mathbf{z}_\mathrm{w} \left(\frac{\mathbf{z}_2 \coth(\boldsymbol{q}\,\mathrm{d}) + \mathbf{z}_\mathrm{w}}{\mathbf{z}_2 + \mathbf{z}_\mathrm{w} \coth(\boldsymbol{q}\,\mathrm{d})}\right) \tag{7.88}$$

When the material is backed by a rigid wall ($\mathbf{z}_2 = \infty$), then we obtain

$$\mathbf{z}_1 = \mathbf{z}_\mathrm{w} \coth(\boldsymbol{q}\,\mathrm{d}) \tag{7.89}$$

Empirical Formulas for the Impedance of Porous Materials

It is difficult to predict the complex characteristic impedance of a material from the flow resistance based on theory alone, so empirical formulas have been developed that give good results. Delany and Bazley (1969) published a useful relationship for the wave impedance of a porous material such as fiberglass

$$\mathbf{z}_\mathrm{w} = w + \mathrm{j}\,x \tag{7.90}$$

$$w = \rho_0\,\mathrm{c}_0 \left[1 + 0.0571 \left(\rho_0\,\mathrm{f}/\mathrm{r}_\mathrm{s}\right)^{-0.754}\right] \tag{7.91}$$

$$x = -\rho_0\,\mathrm{c}_0 \left[0.0870 \left(\rho_0\,\mathrm{f}/\mathrm{r}_\mathrm{s}\right)^{-0.732}\right] \tag{7.92}$$

and the propagation constant is

$$\boldsymbol{q} = \delta + \mathrm{j}\,\beta \tag{7.93}$$

$$\delta \cong \frac{\omega}{\mathrm{c}_0} \left[1 + 0.0978 \left(\rho_0\,\mathrm{f}/\mathrm{r}_\mathrm{s}\right)^{-0.700}\right] \tag{7.94}$$

$$\beta = \frac{\omega}{\mathrm{c}_0} \left[0.189 \left(\rho_0\,\mathrm{f}/\mathrm{r}_\mathrm{s}\right)^{-0.595}\right] \tag{7.95}$$

where $\mathbf{z}_\mathrm{w}$ = complex characteristic impedance of the material
w = resistance or real part of the wave impedance
x = reactance or imaginary part of the wave impedance
$\boldsymbol{q}$ = complex propagation constant
δ = real part of the propagation constant $\cong \omega/\mathrm{c}_0$
β = imaginary part of the propagation constant
= attenuation (nepers/m)

$\rho_0 c_0$ = characteristic acoustic resistance of air (about 412 Ns/m^3 – mks rayls)

r_s = specific flow resistance (mks rayls)
d = thickness of the material (m)
f = frequency (Hz)
$j = \sqrt{-1}$

Figure 7.26 shows measured absorption data versus frequency compared to calculated data using the relationships just shown for two different flow resistance and thickness values. Note that manufacturers usually give the specific flow resistance in cgs rayls/cm (1 cgs rayl = 10 mks rayls). The shape of the curve is determined by the total flow resistance, and the thickness sets the cutoff point for low-frequency absorption.

Diffuse-field absorption coefficients show a similar behavior with thickness. For oblique incidence there is a component of the velocity parallel to the surface so there is some absorption, even near the wall. The thickness and spacing of a porous absorber such as a pressed-fiberglass panel, mounted on a concrete wall or other highly reflecting surface, still determines the frequency range of its absorption characteristics. Figure 7.27 shows the measured diffuse-field absorption coefficients of various thicknesses of felt panel.

Thick Porous Materials with an Air Cavity Backing

When a thick porous absorber is backed by an air cavity and then a rigid wall, the back impedance behind the absorber is given in Eq. 7.76. This value can be inserted into Eq. 7.63 to get the overall impedance of the composite. The absorption coefficient is plotted in Fig. 7.28 for several thicknesses. In each case the total depth and total flow resistance are the same. Note that the specific flow resistance has been changed to offset the changes in thickness. When materials are spaced away from the wall, they should have a higher characteristic resistance. It is interesting that, as the material thickness decreases, the effect of the quarter-wave spacing becomes more noticeable since its behavior approaches that of a thin resistive absorber.

FIGURE 7.26 Normal Absorption Coefficient vs Frequency for Pressed Fiberglass Board (Hamet, 1997)

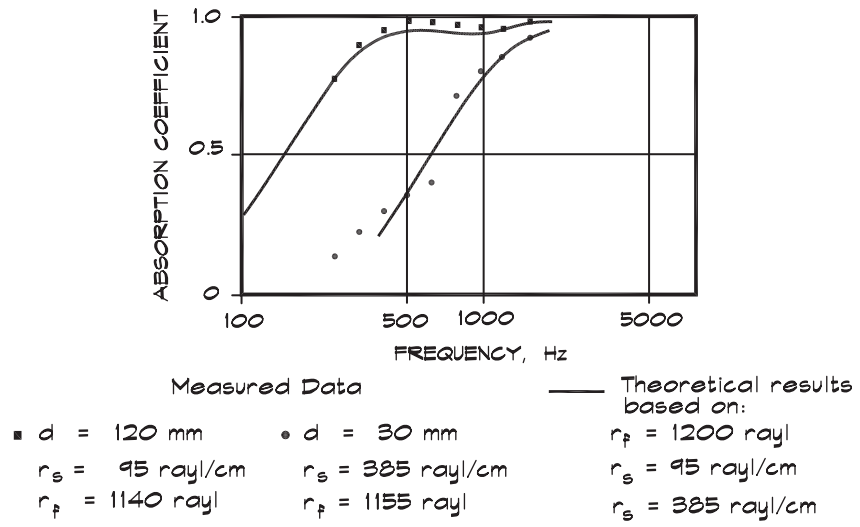


FIGURE 7.27 Dependence of Absorption on Thickness (Ginn, 1978)

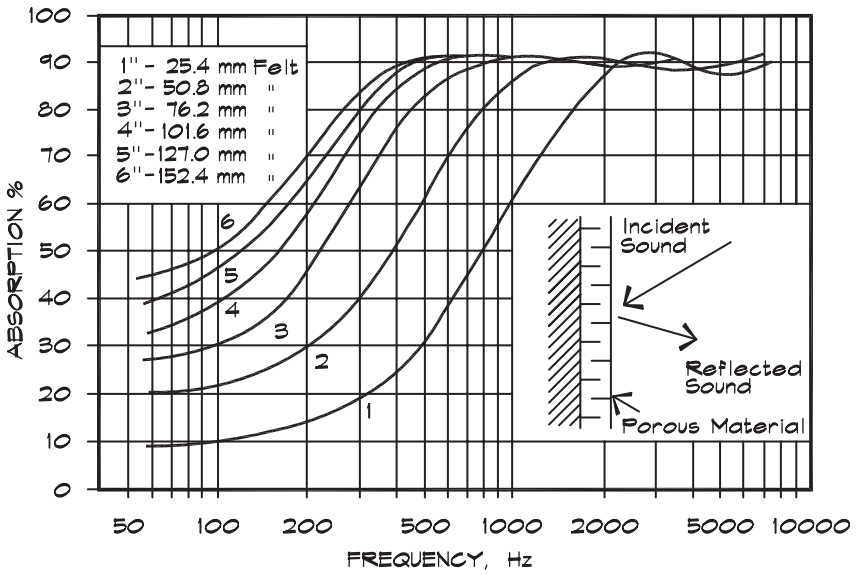
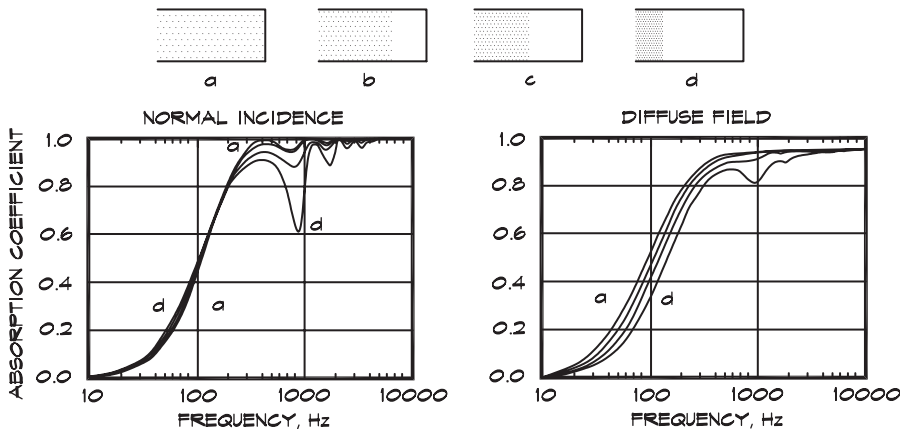


FIGURE 7.28 Absorption of Thick Materials with Air Backing (Ingard, 1994)

The absorption coefficient of a nonlocally reacting porous layer backed by an air cavity and a rigid wall. The overall thickness of the layer-cavity combination is 20 cm (8") and the different layer thicknesses are 5 cm (2"), 10 cm (4"), 15 cm (6") and 20 cm (8"). The total flow resistance in each case is $2\rho_0 c_0$.



Practical Considerations in Porous Absorbers

For most architectural applications, a 1" (25 mm) thick absorbent fiberglass panel applied over a hard surface is the minimum necessary to control reverberation for speech intelligibility. Some localized effects such as high-frequency flutter echoes can be reduced using thinner materials such as 3/16" (5 mm) wall fabric or 1/4" (6 mm) carpet, but these materials are not thick enough for general applications. If low-frequency energy in the 125 Hz. octave band is of concern, then at least 2" (50 mm) panels are necessary. At even lower frequencies, 63 Hz and below, panel absorbers such as a gypboard wall, or Helmholtz bass traps are required.

FIGURE 7.29 Diffuse Field Absorption Coefficient (Ingard, 1994)

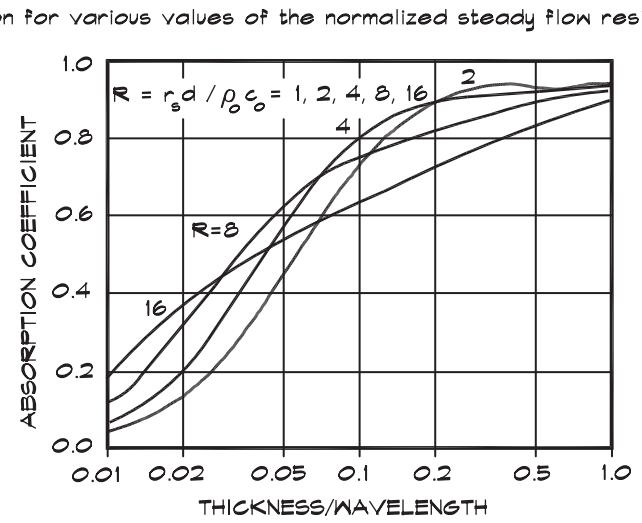


Figure 7.29 shows the absorption of materials of the same thickness but having different flow resistances. Normally a value around $2\,\rho_0 c_0$ of total flow resistance is optimal at the mid and high frequencies for a wall-mounted absorber (Ingard, 1994). At lower frequencies, or when there is an air cavity backing, higher resistances are better.

When a relatively dense material such as acoustical tile is suspended over an airspace, it can be an effective broadband absorber. Figure 7.30 shows the difference in low-frequency performance for acoustical tiles applied with adhesive directly to a reflecting surface and those supported in a suspension system. The thickness of the material is still important so that the absorption coefficient does not exhibit the high-frequency dependence shown earlier. In general, fiberglass tiles are more effective at high frequencies than mineral-fiber tiles since their characteristic resistance is lower.

Wrapping materials with a porous cloth covering has little effect on the absorption coefficient. The flow resistance of the cloth must be low. If it is easy to blow through it there is little change in the absorption. Paper or vinyl backings raise the resistance and lower the high-frequency absorption. Small perforations made in a vinyl fabric can reduce the flow resistance while delivering a product that is washable.

Screened Porous Absorbers

Absorptive materials can be overlaid by a porous screen with little effect on their properties, so long as the covering is sufficiently open. Slats of wood or metal are commonly used to protect these soft absorbers from wear and to improve their appearance. Perforated metals and wire mesh screens are also employed and can be effective as long as there is sufficient open area and the hole sizes are not so large that the spaces between the holes become reflecting surfaces, or so small that they become clogged with dirt or paint. Figure 7.31 (Doelle, 1972) shows the behavior of a porous absorber covered with a perforated facing. If there is at least 15 to 20% open area, the material works as if it were unfaced. Several examples of spaced facings are shown in Fig. 7.32.

FIGURE 7.30 Average Absorption of Acoustical Tiles (Doelle, 1972)

(A) Applied with adhesive, (B) In a suspended grid

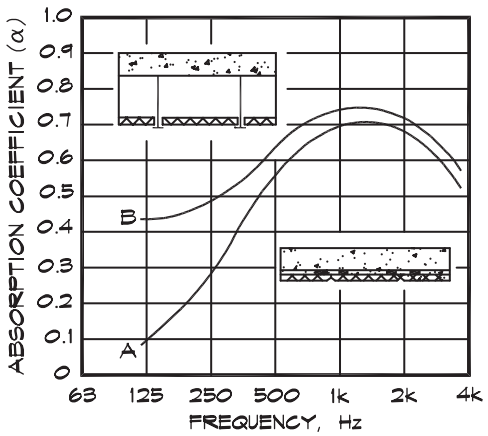
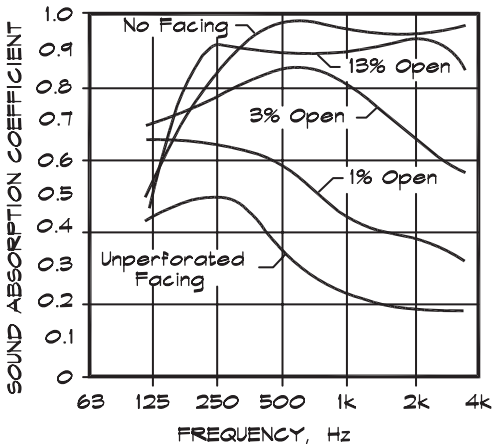


FIGURE 7.31 Sound Absorption of Perforated Panels (Doelle, 1972)

Sound absorption of perforated panel resonators with an isolation blanket in the air space. The open area (sound transparency) of the perforated facing has a considerable effect on the absorption.



The effects of painting tiles are shown in Fig. 7.33 (Doelle, 1972). When a porous absorber is painted, its effectiveness can drop dramatically if the passage of air through its surface is impeded. This is especially true of acoustical tiles, which rely on holes or perforations to achieve their porosity. Unfaced fiberglass and duct liner boards that have a cloth face can be painted once with a light spray coat of nonbridging (water-base) paint without undue degradation. Multiple coats progressively reduce the high-frequency absorption. Clearly there are marked differences in absorptivity attributable to the thickness and number of coats of paint. Porous materials such as concrete block or certain types of stone need to be coated with paint or sealer to decrease their absorptivity (and increase their transmission loss) when they are used in churches or other spaces where a long reverberation time is desired.

FIGURE 7.32 Various Configurations of Wood Slats (Doelle, 1972)

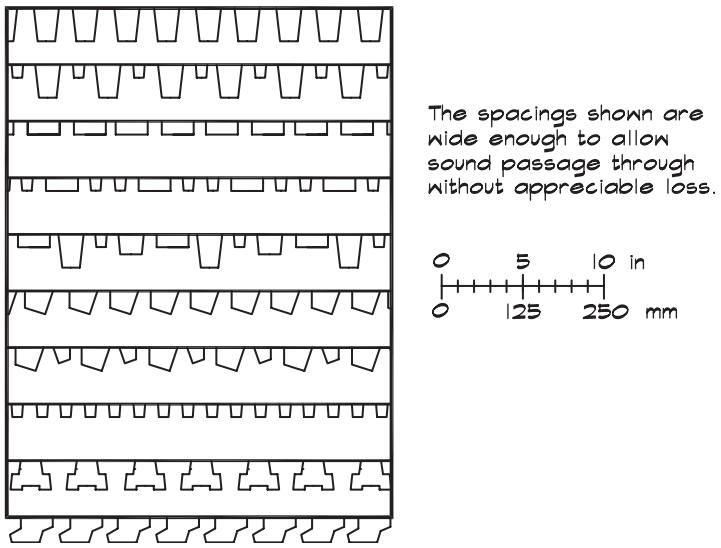
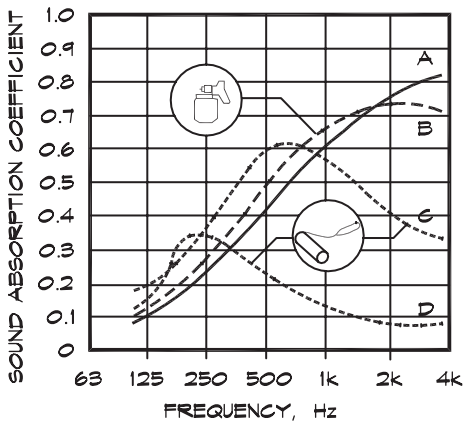


FIGURE 7.33 Effect of Paint on Absorptive Panels (Doelle, 1972)

The effect of paint on porous prefabricated acoustical units: (A) untreated surface (B) one coat of paint applied with a spray gun (C) one coat of paint applied with a brush (D) two coats of paint applied with a brush.

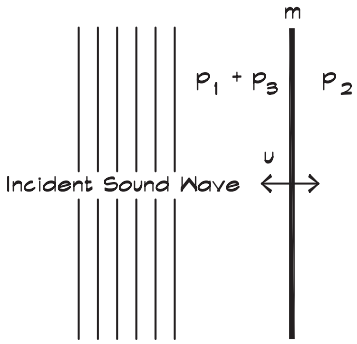


7.5 ABSORPTION BY NONPOROUS ABSORBERS

Unbacked Panel Absorbers

A freely suspended nonporous panel can absorb sound simply due to its mass reactance: that is, its induced motion. For this reason even solid walls provide some residual absorption, which may be only a few percentage points. Figure 7.34 shows the geometry of a normally incident sound wave impacting a solid plate. On the source side we have the incident pressure $\mathbf{p}_1$ and the reflected pressure $\mathbf{p}_3$; on the opposite side we have the transmitted pressure $\mathbf{p}_2$. The total pressure acting on the wall, $\mathbf{p}_1 + \mathbf{p}_3 - \mathbf{p}_2$, induces a motion in the panel according to Newton's law, $\mathbf{p}_1 + \mathbf{p}_2 - \mathbf{p}_3 = \mathrm{j}\,\omega\,\mathrm{m}\,\mathbf{u}$. Since the sound wave on the right side is radiating

FIGURE 7.34 Geometry of an Unbacked Panel Absorber



into free space we can write the pressure in terms of the particle velocity $\mathbf{p}_2 = \rho_0 \, c_0 \, \mathbf{u}$ to obtain the relationship $\mathbf{p}_1 + \mathbf{p}_3 - \rho_0 \, c_0 \, \mathbf{u} = j \, \omega \, m \, \mathbf{u}$ and from this, the impedance of the panel is (Kuttruff, 1963)

$$\mathbf{z} = j \, \omega \, m + \rho_0 \, c_0 \tag{7.96}$$

Now this expression can be inserted into Eq. 7.63 to calculate the normal incidence absorption coefficient

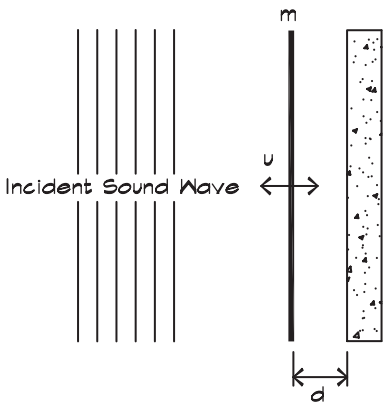
$$\alpha_n = \left[1 + \left(\frac{\omega \, m}{2 \, \rho_0 \, c_0} \right)^2 \right]^{-1} \tag{7.97}$$

From this expression the absorption of windows and solid walls can be calculated; however, their residual absorption is small and only significant at low frequencies.

Air Backed Panel Absorbers

When a nonporous panel, as in Fig. 7.35, is placed in front of a solid surface with no contact between the panel and the surface, the panel can move back and forth, but is resisted by the air spring force.

FIGURE 7.35 Geometry of an Air Backed Panel Absorber



When there is a pressure differential across the panel Newton's law governs the motion

$$\Delta \mathbf{p} = \mathrm{m}\frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}\,\mathrm{t}} = \mathrm{j}\,\omega\,\mathrm{m}\,\mathbf{u} \tag{7.98}$$

where m is the mass of the panel per unit area. Using the same notation as before, with $\mathbf{p}_1 + \mathbf{p}_3$ being the pressure in front of the panel and $\mathbf{p}_2$ the pressure behind the absorber, we obtain

$$\frac{\mathbf{p}_1 + \mathbf{p}_3 - \mathbf{p}_2}{\mathbf{u}} = \mathrm{r_f} + \mathrm{j}\,\omega\,\mathrm{m} \tag{7.99}$$

and in a similar manner the composite assembly impedance is

$$\mathbf{z} = \mathrm{r_f} + \mathrm{j}\left[\omega\,\mathrm{m} - \rho_0\,\mathrm{c}_0\,\cot(\mathrm{k\,d})\right] \tag{7.100}$$

When the depth, d, of the airspace behind the sheet is small compared to a wavelength, we can use the approximation $\cot(\mathrm{k\,d}) \cong (\mathrm{k\,d})^{-1}$ so that

$$\mathbf{z} \cong \mathrm{r_f} + \mathrm{j}\left[\omega\,\mathrm{m} - \frac{\rho_0 \mathrm{c}_0^2}{\omega\,\mathrm{d}}\right] \tag{7.101}$$

As before, we can insert this expression into Eq. 7.63 to obtain the normal-incidence absorption coefficient (Kuttruff, 1973)

$$\alpha_\mathrm{n} = \frac{4\,\mathrm{r_f}\,\rho_0\,\mathrm{c}_0}{\left(\mathrm{r_f} + \rho_0 \mathrm{c}_0\right)^2 + \left[(\mathrm{m}/\omega)\left(\omega^2 - \omega_0^2\right)\right]^2} \tag{7.102}$$

where we have used the resonant frequency from the bracketed term in Eq. 7.101, whose terms are equal at resonance.

$$\omega_\mathrm{r} = \left(\frac{\rho_0\,\mathrm{c}_0^2}{\mathrm{m\,d}}\right)^{\frac{1}{2}} \tag{7.103}$$

A simpler version of Eq. 7.103 is

$$\mathrm{f_r} = \frac{600}{\sqrt{\mathrm{m\,d}}} \tag{7.104}$$

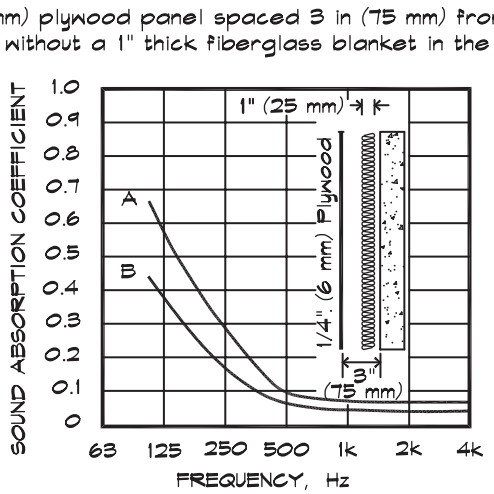
where m is the panel mass in kg / m^2 and d is the thickness of the airspace in cm. When the airspace is filled with batt insulation the resonant frequency is reduced to

$$\mathrm{f_r} = \frac{500}{\sqrt{\mathrm{m\,d}}} \tag{7.105}$$

due to the change in sound velocity.

If the panel is impervious to flow, the flow resistance is infinite, and the absorption is theoretically infinite at resonance. Above and below resonance the absorption coefficient falls

FIGURE 7.36 Sound Absorption of a Suspended Panel (Doelle, 1972)



off. In this model the sharpness of the peak is determined by the amount of flow resistance provided by the panel. When damping is added to the cavity the propagation constant in the airspace becomes complex and adds a real part to the impedance, which broadens the resonance. The damping is provided by fiberglass boards or batting suspended in the airspace behind the panel. Figure 7.36 shows the typical behavior of a panel absorber with and without insulation.

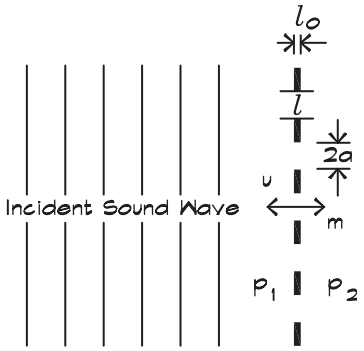
Panel absorbers of this type that are tuned to a low resonant frequency are used as bass traps in studios and control rooms. Thin wood panels, mounted over an air cavity, produce considerable low-frequency absorption and, when there is little or no absorptive treatment behind the panel, this absorption is frequently manifest in narrow bands. As a result wood panels are a serious detriment to adequate bass response in concert halls. It is a common misunderstanding, particularly among musicians, that wood, and in particular thin wood panels that vibrate, contribute to good acoustical qualities of the hall. This no doubt arises from the connection in the mind between a musical instrument such as a violin and the hall itself. In fact, vibrating components in a hall tend to remove energy at their natural frequencies and return some of it back to the hall at a later time.

Perforated Panel Absorbers

If a perforated plate is suspended in a sound field, there is absorption due to the mass reactance of the plate itself, the mass of the air moving through the perforations, and the flow resistance of the material. If the perforated plate is mounted over an air cavity, there is also its impedance to be considered. There are quite a number of details in the treatment of this subject, whose consideration extends past the scope of this book. The goal here is to present enough detail to give an understanding of the phenomena without undue mathematical complication.

In a perforated plate the perforations form small tubes of air, which have a mass and thus a mass reactance to the sound wave. Figure 7.37 illustrates the geometry of a perforated plate. The holes in the plate have a radius a, and are spaced a distance e apart. The fluid velocity $\mathbf{u}_e$ on the exterior of the plate is raised to a higher interior velocity $\mathbf{u}_i$ as the fluid

FIGURE 7.37 Geometry of a Perforated Panel Absorber



is forced through the holes. The ratio of velocities can be written in terms of the ratio of the areas, which is the porosity

$$\frac{\mathbf{u}_e}{\mathbf{u}_i} = \frac{\pi\, a^2}{e^2} = \sigma \tag{7.106}$$

The impedance due to the inertial mass of the air moving through the pores is

$$\frac{\mathbf{p}_1 - \mathbf{p}_2}{\mathbf{u}_e} = \frac{j\,\omega\,\pi\, a^2\,\rho_0\, l}{e^2} = \frac{j\,\omega\,\rho_0\, l}{\sigma} \tag{7.107}$$

so the effective mass per unit area is

$$m = \frac{\rho_0\, l}{\sigma} \tag{7.108}$$

The effective length of the tube made by the perforated hole in Eq. 7.107 is slightly longer than the actual thickness of the panel. This is because the air in the tube does not instantaneously accelerate from the exterior velocity to the interior velocity. There is an area on either side of the plate that contains a region of higher velocity and thus a slightly longer length. This correction is written in terms of the effective length

$$l = l_0 + 2\,(0.8\,a) \tag{7.109}$$

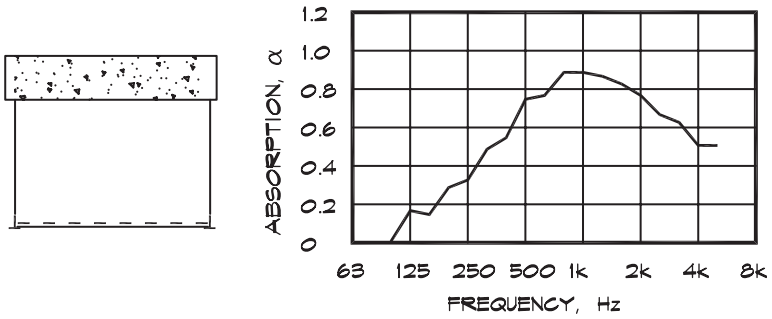
Usually the air mass is very small compared to the mass of the panel itself so that the panel is not affected by the motion of the fluid. If the area of the perforations is large and the panel mass M is small, the combined mass of the air and the panel must be used.

$$m_c = m\,\frac{M}{M + m} \tag{7.110}$$

When M is large compared with the mass of the air, m, then the combined mass is just the air mass.

The absorption coefficient for a perforated panel is obtained from Eq. 7.97, using the combined mass of the air and the panel. In commercially available products, perforated

FIGURE 7.38 Absorption of a Coated Perforated Panel (Wilhelmi Corp. Data, 2000)



metal sheets are available with an air resistant coating, which adds flow resistance to the mass reactance of the air. These materials can be supported by T-bar systems and are effective absorbers. Absorption data for a typical product are shown in Fig. 7.38.

Perforated Metal Grilles

When a perforated panel is being used as a grille to provide a transparent cover for a porous absorber, it is important that there is sufficient open area that the sound passage is not blocked. In these cases the sound "absorbed" by the panel is actually the sound transmitted through the grille into the space beyond. The normal incidence absorption coefficient in this case is the same as the normal incidence transmissivity. Thus we can set it equal to the expression shown in Eq. 7.97

$$\tau = \left[1 + \left(\frac{\omega \, m_c}{2 \, \rho_0 \, c_0}\right)^2\right]^{-1} \tag{7.111}$$

If the loss through a perforated panel is to be less than about 0.5 dB, the transmission coefficient should be greater than about 0.9, and the term inside the parenthesis becomes 0.33. At 1000 Hz the combined panel mass should be 0.04 kg / m^2. For a 2 mm thick (.079") thick panel with 3 mm (.125") diameter holes, the effective length is about 4 mm and the required porosity calculates out to about 11%. This compares well with the data shown in Fig. 7.38, even though the calculation done here is for normal incidence and the data are for a diffuse-field measurement.

If a perforated panel is to be used as a loudspeaker grille, the open area should be greater, on the order of 30 to 40%. Increasing the porosity preserves more of the off-axis directional character of the loudspeaker's sound. Porosities beyond 40% are difficult to achieve in a perforated panel while still retaining structural integrity.

Air Backed Perforated Panels

When a perforated panel is backed with an airspace of a given depth, the back impedance of the airspace must be considered. The three contributing factors are the mass reactance of the air/panel system, the flow resistance of any filler material, and the stiffness of the air cavity behind the panel. The overall impedance is the sum of these three contributors, which at low

frequencies was given in Eq. 7.102

$$\mathbf{z} = \mathrm{j}\,\omega\,\mathrm{m_c} + \mathrm{r_f} - \frac{\mathrm{j}\,\rho_0\,\mathrm{c_0^2}}{\omega\,\mathrm{d}} \tag{7.112}$$

In the case of a perforated panel the combined mass of the panel and the air through the pores is used. We obtain the same result as Eq. 7.102, which we used for the absorption coefficient of a closed panel, and it is expressed in terms of the resonant frequency of the panel-cavity-spring-mass system.

$$\omega_0 = \left(\frac{\rho_0\,\mathrm{c_0^2}}{\mathrm{m_c\,d}}\right)^{\frac{1}{2}} \tag{7.113}$$

When we substitute the mass of the moving air in terms of the length of the tube and the porosity we get a familiar result—the Helmholtz resonator natural frequency. Here we have assumed that the mass of the air is much smaller than the panel mass

$$\mathrm{m_c} = \frac{\rho_0\,l}{\sigma} = \frac{\rho_0\,l\,\mathrm{e}^2}{\pi\,\mathrm{a}^2} = \frac{\rho_0\,l\,\mathrm{V}}{\mathrm{S\,d}} \tag{7.114}$$

and

$$\omega_0 = \left(\frac{\rho_0\,\mathrm{c_0^2}}{\rho_0\,l\,\mathrm{V/S}}\right)^{\frac{1}{2}} = \mathrm{c_0}\sqrt{\frac{\mathrm{S}}{l\,\mathrm{V}}} \tag{7.115}$$

It is apparent that a perforated panel with an air backing is acting like a Helmholtz resonator absorber and will exhibit similar characteristics, just as the solid panel did. The major difference is that the moving mass, in this case the air in the holes, is much lighter than the panel and thus the resonant frequency is much higher. These perforated absorbers are mainly utilized where mid-frequency absorption is needed.

The flow resistance of perforated panels can be measured directly or can be calculated from empirical formulas, such as that given by Cremer and Muller (1982)

$$\mathrm{r_f} \cong 0.53\,\frac{\mathrm{e}^2\,l_0}{\mathrm{a}^3}\,\sqrt{\mathrm{f}}\cdot 10^{-2}\ \text{mks rayls} \tag{7.116}$$

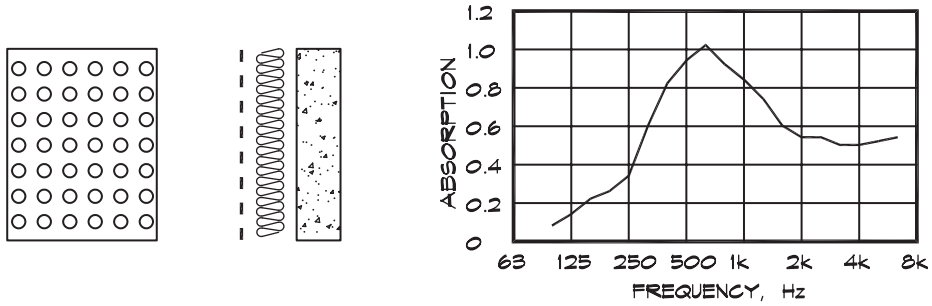
Figure 7.39 shows the absorption coefficient of a perforated plate in front of an airspace filled with absorptive material.

7.6 ABSORPTION BY RESONANT ABSORBERS

Helmholtz Resonator Absorbers

When a series of Helmholtz resonators is used as an absorbing surface, the absorption coefficient can be calculated in a manner similar to that used for a perforated plate.

FIGURE 7.39 Absorption of a Perforated Panel (Cremer and Muller, 1982)



The resonant frequency is given by

$$f_0 = \frac{c_0}{2\pi}\sqrt{\frac{\pi a^2}{(l_0 + 1.7a)V}} \tag{7.117}$$

where a is the radius of the resonator neck, l_0 is its length, and V its volume. The question then is how to calculate the depth, d, of the cavity. With a perforated plate the volume of the airspace was

$$d = \frac{V}{e^2} \tag{7.118}$$

where e is the spacing between perforations. Using V as the volume of the Helmholtz resonator the normal-incidence absorption coefficient for a series of resonators is (Cremer and Muller, 1982)

$$\alpha_n = \frac{4 r_f}{\rho_0 c_0}\left[\left(1 + \frac{r_f}{\rho_0 c_0}\right)^2 + \left(\frac{c_0 e^2}{2\pi f_0 V}\right)^2\left(\frac{f_0}{f} - \frac{f}{f_0}\right)^2\right]^{-1} \tag{7.119}$$

Products based on the Helmholtz resonator principle are commercially available. Some are constructed as concrete masonry units with slotted openings having a fibrous or metallic septum interior fill. Absorption data on typical units are given in Fig. 7.40.

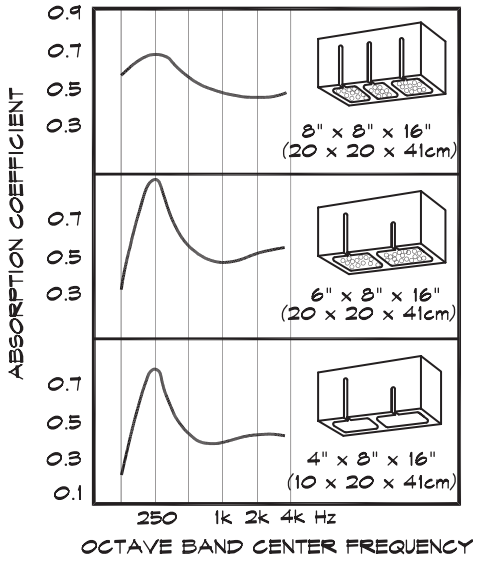
Mass-Air-Mass Resonators

A mass-air-mass resonant system is one in which two free masses are separated by an air cavity that provides the spring force between them. A typical example is a drywall stud wall, which acts much like the resonant panel absorber, except that both sides are free to move. The equation is similar to that used in the single-panel equation (Eq. 7.104) except that both masses are included. The resonant frequency is

$$f_{mam} = 600\sqrt{\frac{m_1 + m_2}{d\, m_1\, m_2}} \tag{7.120}$$

where m_1 and m_2 are the surface mass densities of the two surfaces in kg / m^2 and d is the separation distance between the sides in cm. When the cavity is filled with batt insulation,

FIGURE 7.40 Helmholtz Resonator Absorbers (Doelle, 1972)



the constant changes from 600 to 500 because the sound velocity goes from adiabatic to isothermal. Away from resonance the absorption coefficient follows the relationship (Bradley, 1997)

$$\alpha(\mathrm{f}) = \alpha_{\mathrm{mam}} \left(\frac{\mathrm{f}_{\mathrm{mam}}}{\mathrm{f}} \right)^2 + \alpha_{\mathrm{s}} \tag{7.121}$$

where $\alpha(\mathrm{f})$ = diffuse field absorption coefficient
α_{mam} = maximum absorption coefficient at $\mathrm{f}_{\mathrm{mam}}$
α_{s} = residual surface absorption coefficient at high frequencies
f = frequency (Hz)
$\mathrm{f}_{\mathrm{mam}}$ = mass-air-mass resonant frequency (Hz)

Figure 7.41 shows the absorption coefficient for a single and double-layer drywall stud wall using $\alpha_{\mathrm{mam}} = 0.44$ and $\alpha_{\mathrm{s}} = 0.045$ for the single-layer wall and $\alpha_{\mathrm{mam}} = 0.44$ and $\alpha_{\mathrm{s}} = 0.06$ in the double-layer case. The same constants are used for batt-filled stud walls. The agreement shown in the figure between measured and predicted values is quite good.

Quarter-Wave Resonators

A hard surface having a well of depth d and diameter 2 a can provide absorption through reradiation of sound that is out of phase with the incident sound. These wells, shown in Fig. 7.42, are known as quarter-wave resonators because a wave reflected from the bottom returns a half wavelength or 180° out of phase with the wave reflected from the surface. When the length of the tube is an odd-integer multiple of a quarter wavelength it is out of phase with the incident wave and perfectly absorbing. The tube acts as a small resonant radiator, which can have both absorptive and diffusive properties.

As was the case in Eq. 7.89, the interaction impedance of a tube having a depth d is

$$\mathbf{z}_{\mathrm{t}} = -\mathrm{j}\,\rho_0\,\mathrm{c}_0\,\cot(\boldsymbol{q}\,\mathrm{d}) \tag{7.122}$$

FIGURE 7.41 Resonant Absorption by a Stud Wall (Bradley, 1997)

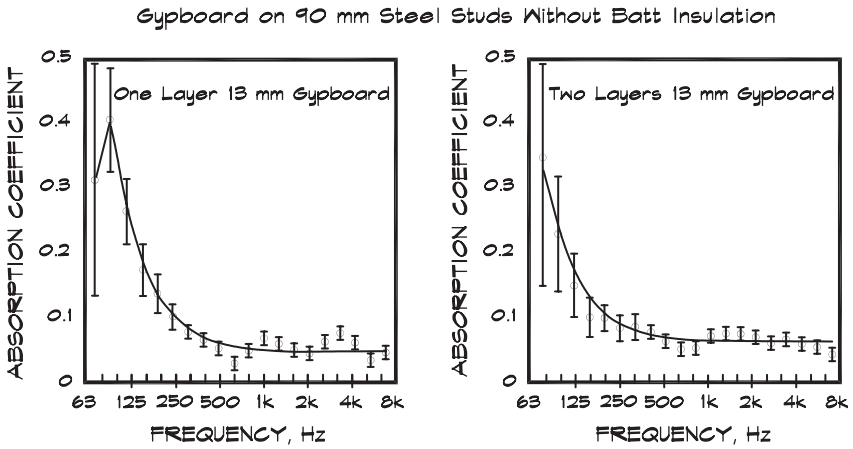
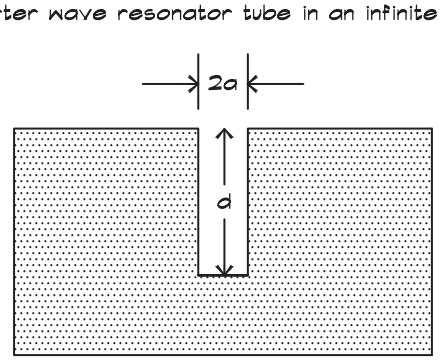


FIGURE 7.42 Quarter Wave Resonator



where $\boldsymbol{q}$ is the propagation constant. The tube has small viscous and thermal loss components and the imaginary part of the propagation constant, from an approximation originally due to Kirchoff, can be used to account for them

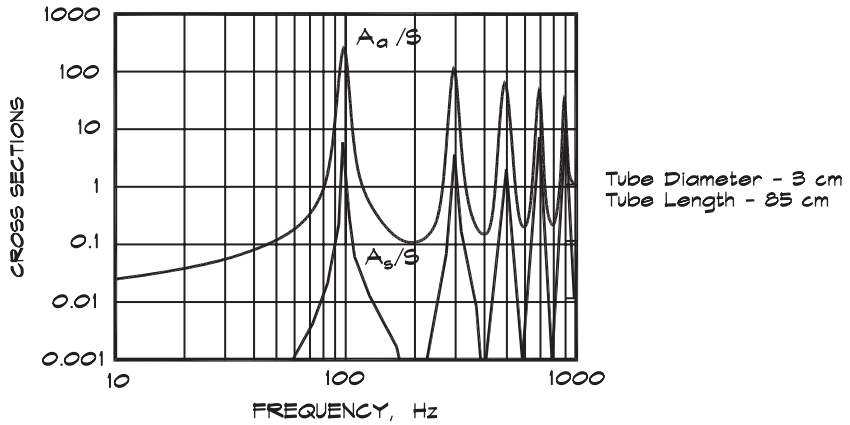
$$\boldsymbol{q} \cong \mathrm{k}\left[1 + \frac{0.31\,\mathrm{j}}{2\,\mathrm{a}\sqrt{\mathrm{f}}}\right] \tag{7.123}$$

There are two impedances to be included in the analysis: one having to do with the interaction between the incoming wave and the end of the tube, which was given in Eq. 7.122; and the other having to do with the radiation of sound back out of the tube. The radiation impedance of the tube is that of a piston in a baffle and was examined in Eq. 6.67 and 6.69 in the near field. For low frequencies, where the width of the opening is much smaller than a wavelength, the radiation impedance is approximately (Morse, 1948)

$$\mathbf{z}_\mathrm{r} \cong \rho_0 \mathrm{c}_0\left[\frac{1}{2}(\mathrm{k\,a})^2 + \frac{2\,\mathrm{j}}{\pi\,\mathrm{k\,a}}\right] \tag{7.124}$$

The pressure just outside the opening to the tube is the pressure radiated by the tube, plus twice the incident pressure, which is doubled due to its reflection off the rest of the hard

FIGURE 7.43 Absorption and Scattering Cross Sections of a Tube Resonator in a Wall (Ingard, 1994)



surface, $\mathbf{p}_{\text{outside}} = 2\,\mathbf{p}_{\text{i}} + \mathbf{p}_{\text{r}}$ where $\mathbf{p}_{\text{r}} = \mathbf{u}\,\mathbf{z}_{\text{r}}$. The pressure just outside the opening must match the pressure just inside the end of the tube, which is $\mathbf{p}_{\text{inside}} = -\mathbf{u}\,\mathbf{z}_{\text{t}}$. At the surface the pressures and the velocities must match, which leads to

$$\mathbf{u} = \frac{2\,\mathbf{p}_{\text{i}}}{\mathbf{z}_{\text{t}} + \mathbf{z}_{\text{r}}} \tag{7.125}$$

The absorption of a well in a surface can be expressed in terms of a cross section, defined (Ingard, 1994) as the power absorbed by the well divided by the intensity of the incident wave. The power absorbed by the resonator tube is

$$\mathrm{W_a} = \mathrm{S}\,|\,\mathbf{u}\,|^2 \;\; w_{\text{t}} = \mathrm{S}\,\frac{|\,\mathbf{p}_{\text{i}}\,|^2}{(\rho_0\,\mathrm{c}_0)}\,\frac{4\,\rho_0\,\mathrm{c}_0 w_{\text{t}}}{|\,\mathbf{z}_{\text{t}} + \mathbf{z}_{\text{r}}\,|^2} \tag{7.126}$$

where $\mathbf{z} = w + \mathrm{j}\,x$ and $\mathrm{S} = \pi\mathrm{a}^2$. Since $\mathrm{I_i} = |\,\mathbf{p}_{\text{i}}\,|^2 / (\rho_0\,\mathrm{c}_0)$ is the intensity of the incident wave, the power absorbed can be expressed as $\mathrm{W_a} = \mathrm{A_a}\,\mathrm{I_i}$, where $\mathrm{A_a}$ is the absorption cross section.

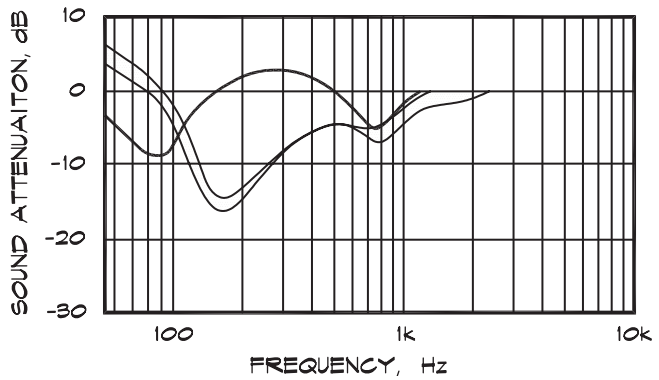
$$\mathrm{A_a} = \mathrm{S}\,\frac{4\,\rho_0\,\mathrm{c}_0\,w_{\text{t}}}{|\mathbf{z}_{\text{t}} + \mathbf{z}_{\text{r}}|^2} \tag{7.127}$$

A typical result, given in Fig. 7.43, shows strong peaks at the minima of the tube impedance. When the cross section is 100, it means that the tube is acting as a perfect absorber equal to 100 times its open area. Note that although a tube can be very effective at a given frequency, its bandwidth is very narrow.

The same figure shows the cross section of the power scattered back by the tube. The tube behaves like a piston in a baffle when it radiates sound back out. It continues to resonate even after the initial wave has been reflected and emits sound at its resonant frequency for a short period of time. When the tube mouth dimension is small compared with a wavelength, it acts as an omnidirectional source that diffuses sound at that frequency.

FIGURE 7.44 Seat Absorption (Schultz and Watters, 1964)

Anomalous sound attenuation (after correction for spherical divergence) for the early sound pressure level above the audience seating in a concert hall, as a function of frequency. —— Various seats on the main floor. —— First row in the first balcony.



Absorption by Seats

It has been recognized for some time that theater seating, both occupied and unoccupied, produces excess attenuation of the direct sound coming from a stage, primarily at about 150 Hz, in much the same way as soft earth or vegetation contributes to excess ground attenuation. Padded opera chairs in a theater subdivide the floor into a regular lattice having a particular depth and spacing. As such they are like an array of quarter-wave resonators over which a sound wave, coming from the stage, grazes. In addition, the porous padding that covers them adds a resistive component to the impedance they present to a wave. A measurement of the excess attenuation due to theater seating, is shown in Fig. 7.44. Note that the dip is broader than the behavior predicted in Fig. 7.43 due to the resistive padding and the fact that the higher modes are not as prevalent. This may be due in part to the fact that the seat spacing is no longer small compared with a wavelength.

Ando (1985) published a detailed theoretical study of the absorptive properties of different chair-shaped periodic structures. Although the absorption varies somewhat with the precise shape selected, the basic pattern of the excess attenuation exhibits a steep dip at the frequency whose quarter wavelength is equal to the chair-back height above the floor. This agrees well with measurements made in concert halls. The excess grazing attenuation contributes to decreased bass response particularly in the orchestra seating section on the first floor of a hall. To help offset the extra attenuation, overhead reflectors can be used, which increase the angle of grazing incidence.

Quadratic-Residue Diffusers

One particular type of resonant tube absorber originally suggested by Schroeder (1979) uses a series of wells of different depths in a particular sequential order. Since each well is a small narrow-band omnidirectional radiator, a series of wells of different depths can cover a range of frequencies and provide diffusion over a reasonable bandwidth. The depth d_n of the nth well is chosen such that

$$d_n = \left(\frac{\lambda}{2N}\right) s_n \tag{7.128}$$

where the sequence $s_n = (n^2 \bmod N)$ for $n = 0, 1, 2, \ldots$ and N is an odd prime number. For example, for $N = 11$ starting with $n = 0$, the sequence is [0, 1, 4, 9, 5, 3, 3, 5, 9, 4, 1] and then it repeats so that the period is N numbers long. This sequence of wells produces an essentially hemispheric polar reflection pattern within certain frequency limits. For a more detailed treatment, refer to Ando (1985).

The design process for a quadratic-residue diffuser is as follows:

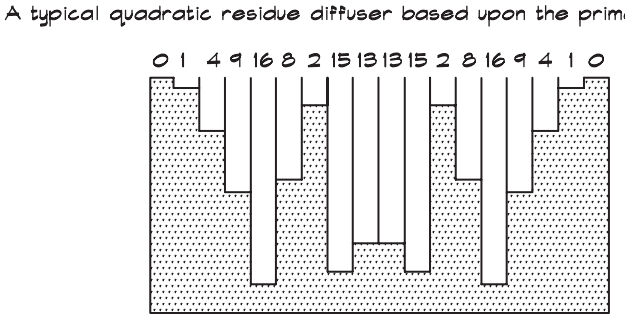
1. Determine the frequency range f_{high} to f_{low} for the diffuser. The period N is given by the ratio f_{high} / f_{low}.
2. The width w of each well must be small compared with the wavelength of the highest frequency.

$$w \leq \frac{c_0}{2 f_{high}} \tag{7.129}$$

3. Calculate the depth of each well using Eq. 7.128, where $\lambda = \frac{c_0}{f_{low}}$ is called the design wavelength.

Figure 7.45 shows a side view of a quadratic-residue diffuser. One feature of quadratic-residue diffusers is that they absorb sound at low frequencies as might be expected from our previous analysis.

FIGURE 7.45 Quadratic Residue Diffusers



This page intentionally left blank

8 SOUND IN ENCLOSED SPACES

8.1 STANDING WAVES IN PIPES AND TUBES

Resonances in Closed Tubes

When a sound is generated within a solid enclosure such as a pipe, tube, or room, it expands naturally to fill the space. If the lateral dimensions of the space are small compared with a wavelength, as is the case with a tube, duct, or organ pipe, the sound propagates along the tube as a plane wave until it encounters an impedance boundary. If the tube is closed, the impedance at the end is very high, assumed to be infinite in a simple model, and the wave reflects back along the tube in the direction from which it came. If both ends are closed, the wave can reflect back and forth many times with little attenuation. The length of the tube in the direction of wave propagation determines the frequencies of the sound waves that persist under these conditions. These resonant frequencies are self-reinforcing since they combine in phase and continue for a long time after an exciting source is turned off. Other frequencies may be present initially; however, because of the geometry of the tube, they tend to cancel each other and average out to zero.

The behavior of the pressure in a one-dimensional plane wave propagating along a tube can be described using the formula

$$\mathbf{p} = \mathbf{A} \cos (\mathrm{k\,x} + \phi) \tag{8.1}$$

where x is the distance from the end of the tube. If both ends of the tube are rigid this fact establishes the mathematical boundary conditions at the end points. At $\mathrm{x} = 0$, the rigid boundary condition requires that the change in pressure with distance be zero at the boundary,

$$\left(\frac{\partial \mathbf{p}}{\partial \mathrm{x}}\right)_{\mathrm{x}=0} = 0 \quad \text{consequently } \phi = 0 \tag{8.2}$$

If the same condition is applied at the other termination

$$\left(\frac{\partial \mathbf{p}}{\partial \mathrm{x}}\right)_{\mathrm{x}=l} = \sin \mathrm{k}\, l = 0 \tag{8.3}$$

which is satisfied for $k\ l = n\ \pi$. The resonant frequencies of the sound wave in a closed tube are given by

$$f_n = \frac{n\ c}{2\ l} \tag{8.4}$$

where n is an integer. The lowest frequency is called the fundamental mode, where $n = 1$, and the length of the tube is half of the fundamental wavelength.

Standing Waves in Closed Tubes

There are two possible solutions for plane wave propagation in a closed tube, one for sound moving in each direction. They take the form of two traveling waves

$$\mathbf{p} = \mathbf{A}\ \cos\ (-k\ x + \omega\ t) + \mathbf{A}\ \cos\ (k\ x + \omega\ t) \tag{8.5}$$

Using a trigonometric identity for the sum of two cosine waves, the equation can be written as

$$\mathbf{p} = 2\ \mathbf{A}\ \cos\ (k\ x)\ \cos\ (\omega\ t) \tag{8.6}$$

which is a standing wave. At any particular point a particle vibrates back and forth in simple harmonic motion at a particular frequency; however, its amplitude is larger or smaller depending on its location along the x axis.

The locations of the values of maximum pressure, called *antinodes*, occur at positions where

$$k\ x = 0, \pi,\ 2\pi,\ 3\pi, \ldots \tag{8.7}$$

which means

$$x = 0,\ \frac{\lambda}{2},\ \frac{2\ \lambda}{2},\ \frac{3\ \lambda}{2}, \ldots = \frac{n\ \lambda}{2} \tag{8.8}$$

These pressure antinodes occur at the rigid boundaries and have an amplitude that is twice the amplitude of a free traveling wave.

If we compare the behavior of a sound wave near a solid boundary to that of a water wave there is considerable similarity. When an ocean wave washes up against a sea wall the water rises to a height twice that of a freely propagating surface wave and falls by a similar amount. A water particle close to the wall experiences the maximum displacement up and down in response to the impact of the wave. With a sound wave there is a pressure doubling at a solid boundary resulting in a 6 dB increase in sound pressure level.

Minimum values of the sound pressure are called *nodes* and occur at positions where

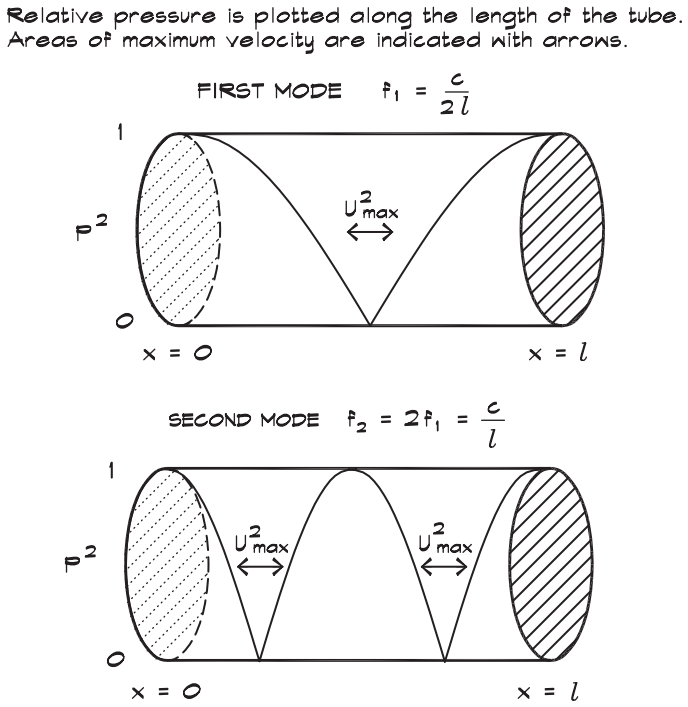
$$k\ x = \frac{\pi}{2},\ \frac{3\ \pi}{2},\ \frac{5\ \pi}{2}, \ldots \tag{8.9}$$

and

$$x = \frac{\lambda}{4},\ \frac{2\ \lambda}{4},\ \frac{3\ \lambda}{4}, \ldots = \frac{n\ \lambda}{4} \tag{8.10}$$

The pressure node in the fundamental occurs at the quarter-wavelength point, which is at the midpoint of the enclosure. This leads to a phenomenon in studios, graphically described as

FIGURE 8.1 Standing Waves in a Closed-Closed Tube



bass suckout—a lack of low-frequency energy at the mixer position near the center of the room, which is accentuated by the placement of bass loudspeakers at the ends of the room where the fundamental mode is easily excited.

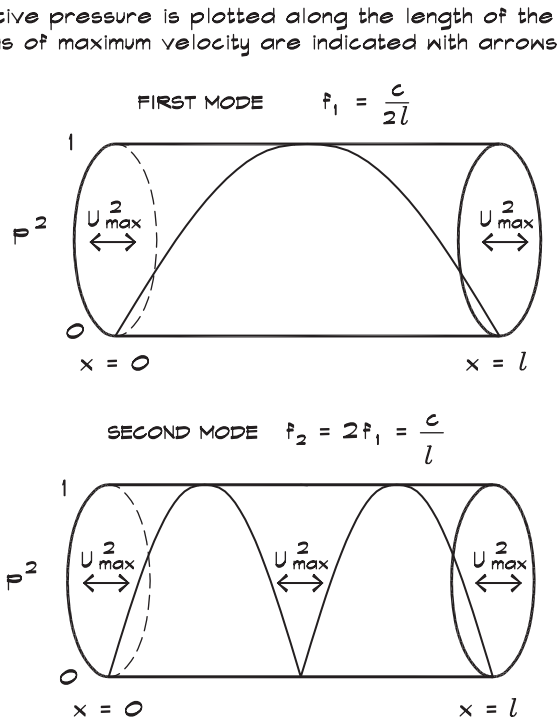
An example of standing waves is shown in Fig. 8.1. These are constructed from continuously propagating rightward and leftward traveling waves. As they move past one another and combine, their sum produces the nodes and antinodes shown in the figure.

Standing Waves in Open Tubes

When a plane wave propagates along a tube that is open at the end, some of the sound energy is reflected from the open boundary. The reflection at this pressure release surface comes about due to the mass and springiness of the air column. An analogy may be drawn using the example of a toy paddle ball, where a rubber ball is attached to the paddle by means of an elastic band. When the ball strikes the paddle it is like a sound wave reflecting off the closed end of the tube. There is a pressure maximum, corresponding to squeezing the ball together, which forces a rebound. When the ball reaches the end of the elastic tether, it acts like the open end of the tube. A mass of air moves beyond the end of the tube and is pulled back by the elastic-spring force caused by the low-pressure region behind it. Air that is confined by the boundaries of the tube acts like a spring since the pressure cannot equalize in directions normal to the direction of wave propagation.

When both ends of the tube are open, the boundary condition requires that the acoustic pressure goes to zero at each end. The allowed solutions for the pressure take the form of a

FIGURE 8.2 Standing Waves in an Open-Open Tube



sine wave

$$\mathbf{p} = \mathbf{A}\ \sin\ \mathrm{k\,x} = \mathbf{A} \sin \frac{\mathrm{n}\,\pi\,\mathrm{x}}{l} \tag{8.11}$$

having the same resonant frequencies as those given in Eq. 8.4. The difference between a totally open pipe and a totally closed pipe is that, with an open pipe, the pressure maximum (antinode) in the fundamental frequency is at the center of the pipe, rather than at the ends, as shown in Fig. 8.2. This is to be expected since a sine wave is simply a cosine wave shifted by 90°.

In a real pipe the boundary condition is not as simple as a perfect pressure null. Instead there is a finite impedance at the end, which introduces a length correction much like that discussed in Chapt. 7 for perforated plates. For long pipes, this correction is generally small. Refer to Kinsler et al., (1982) for a more detailed treatment.

Combined Open and Closed Tubes

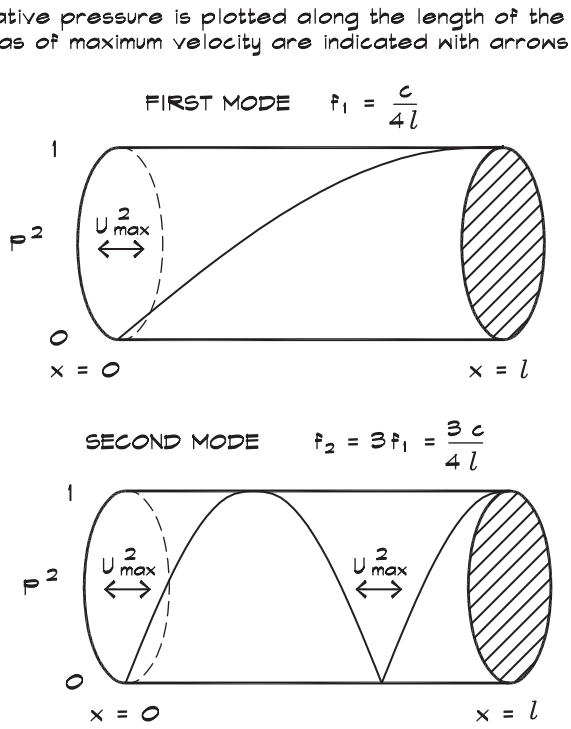
When a tube has one open end and one closed end, the boundary conditions can be applied to a sine wave to obtain

$$\mathbf{p} = \mathbf{A}\ \sin\ \mathrm{k\,x} = \mathbf{A} \sin \frac{(2\,\mathrm{n} - 1)\,\pi\,\mathrm{x}}{2} \tag{8.12}$$

where n = 1, 2, and so on, and the resonant frequencies are

$$\mathrm{f_n} = \frac{(2\,\mathrm{n} - 1)\,\mathrm{c_0}}{4\,l} \tag{8.13}$$

FIGURE 8.3 Standing Waves in a Closed-Open Tube



An organ pipe is open at the source end and can be either open or closed at the opposite end. For a given length the fundamental is one octave lower in a pipe, closed at one end, compared to one that is open. The even harmonics are missing; however, the density of modes remains the same as in the previous two examples. Fig. 8.3 shows the fundamental mode shapes.

8.2 SOUND PROPAGATION IN DUCTS

Rectangular Ducts

Sound waves that can propagate down a duct take on a form that is controlled by the duct dimensions relative to a wavelength. If a long duct is rectangular in cross section, standing waves can form in the lateral directions and a traveling wave in the third direction. The equation for the pressure takes the form

$$\mathbf{p} = \mathbf{A} \cos(k_x x) \cos(k_y y)\, e^{-j k_z z}\, e^{j \omega t} \tag{8.14}$$

where, in the lateral directions, there are allowed values of the wave number

$$k_x = m\pi/a \quad m = 0, 1, 2, \ldots \tag{8.15}$$

and

$$k_y = n\pi/b \quad n = 0, 1, 2, \ldots \tag{8.16}$$

where a and b are the lateral dimensions of the duct. The wave number k is related to its components

$$k^2 = k_x^2 + k_y^2 + k_z^2 \tag{8.17}$$

and to the frequency

$$k = \frac{2\pi f}{c} = \frac{2\pi}{\lambda} \tag{8.18}$$

The z-component of the wave number can be written as

$$k_z = \sqrt{k^2 - (k_x^2 + k_y^2)} \tag{8.19}$$

and inserting Eq. 8.15 and Eq. 8.16 into Eq. 8.19 we obtain

$$k_z = k\sqrt{1 - (f_{mn}/f)^2} \qquad f \geq f_{mn} \tag{8.20}$$

where

$$f_{m,n} = \frac{c}{2\pi}\sqrt{k_x^2 + k_y^2} = \frac{c}{2}\sqrt{\left(\frac{m}{a}\right)^2 + \left(\frac{n}{b}\right)^2} \tag{8.21}$$

is known as the *cutoff frequency* for a given mode (m,n). The indices take on integer values m, n = 0, 1, 2, and so on. By examining Eq. 8.20 it is clear that when $f = f_{m,n}$, $k_z(m, n) = 0$, and there is no wave propagation in the z direction for that particular mode. If the frequency is above cutoff, then $k_z(m, n)$ is real and positive and the mode m,n is called a propagating mode. Consequently plane waves will not be formed when the lateral dimensions of the duct are wider than half a wavelength.

When the frequency is below cutoff for a particular mode, the wave number is imaginary and the mode, which dies out exponentially, is called *evanescent*.

$$\mathbf{k}_z = -j\,\mathbf{k}\sqrt{(f_{mn}/f)^2 - 1} \qquad f < f_{mn} \tag{8.22}$$

Note that when m and n are zero, the wave is planar, and the lower cutoff frequency is zero. Thus there is no lower cutoff frequency for plane waves. The phenomenon of cutoff does not mean that the propagation of sound is cut off. It only means that below the cutoff frequency only plane waves propagate, and above the cutoff frequency only nonplane waves can be formed.

The propagation angle is the angle that a particular mode makes with the z axis. It is determined by using the relationship

$$k_z = k\cos\theta_{m,n} \tag{8.23}$$

where the propagation angle is defined as

$$\theta_{m,n} = \cos^{-1}\left(\sqrt{1 - \left(f_{mn}/f\right)^2}\right) \qquad f \geq f_{mn} \tag{8.24}$$

For plane waves the propagation direction is along the z axis since the cutoff frequency is zero. For a nonzero cutoff frequency, the propagation direction at cutoff for a particular mode is perpendicular to the z axis, so the wave does not propagate down the duct.

At high frequencies the propagation angle is small and the propagation direction approaches the z axis, which means that these modes tend to beam and have very little interaction with the duct walls. We will see the effects of this behavior in a later chapter. Lined ducts provide little attenuation both at very low frequencies, where the thickness of the lining is small compared with a wavelength, and at very high frequencies, where the wave does not interact with the lining due to beaming.

For a circular duct the cutoff frequency for the lowest mode is given by

$$f_{co} = 0.586 \frac{c_0}{d} \tag{8.25}$$

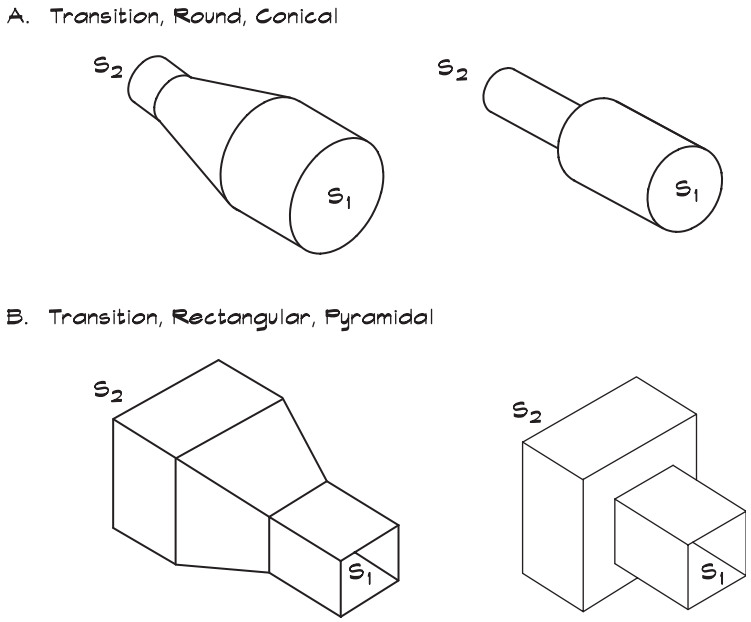
where d is the duct diameter. Above that frequency, modes can be formed across the duct, which combine with waves moving down the duct to generate cross or spinning modes depending on their shape.

Changes in Duct Area

If a plane wave, below the cutoff frequency, propagates down a duct that contains an abrupt change in area, energy will be reflected from the discontinuity. In Fig. 8.4 the area of the rectangular or conical duct changes abruptly from S_1 to S_2 at a point in the duct that marks the transition from region 1 to 2. At this point the pressure and volume velocity amplitudes must be continuous.

$$\mathbf{p}_i + \mathbf{p}_r = \mathbf{p}_t \quad \text{and} \quad S_1 \left(\mathbf{u}_i - \mathbf{u}_r\right) = S_2 \, \mathbf{u}_t \tag{8.26}$$

FIGURE 8.4 Area Changes in Various Types of Ducts



where the subscript i refers to the incident wave, r to the reflected wave, and t to the transmitted wave. The boundary conditions yield

$$\frac{\mathbf{p}_i + \mathbf{p}_r}{\mathbf{u}_i - \mathbf{u}_r} = \frac{S_1}{S_2}\left(\frac{\mathbf{p}_t}{\mathbf{u}_t}\right) \tag{8.27}$$

and using the plane wave relationships $\mathbf{u} = \mathbf{p}/\rho_0 c_0$

$$\frac{\rho_0 c_0 (\mathbf{p}_i + \mathbf{p}_r)}{\mathbf{p}_i - \mathbf{p}_r} = \frac{S_1}{S_2}\left(\frac{\mathbf{p}_t}{\mathbf{p}_t/\rho_0 c_0}\right) = \frac{\rho_0 c_0 S_1}{S_2} \tag{8.28}$$

The reflected amplitude coefficient is

$$\mathbf{r} = \frac{\mathbf{p}_r}{\mathbf{p}_i} = \frac{S_1 - S_2}{S_1 + S_2} \tag{8.29}$$

and the sound energy reflection coefficient is

$$\alpha_r = \left[\frac{S_1 - S_2}{S_1 + S_2}\right]^2 \tag{8.30}$$

and the transmission coefficient is

$$\tau = 1 - \left[\frac{S_1 - S_2}{S_1 + S_2}\right]^2 \tag{8.31}$$

The change in level in decibels, experienced by a plane wave passing through the area change boundary, is

$$\Delta L_a = 10 \log \tau = 10 \log \left[1 - \left(\frac{S_1 - S_2}{S_1 + S_2}\right)^2\right] \tag{8.32}$$

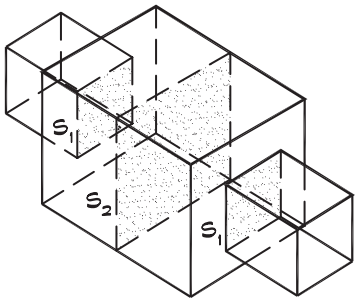
Clearly when the areas are equal there is no loss. For a 50% area reduction the loss is about 1.8 dB. Note that the formula can be used for an area increase or decrease with the same result. The loss due to changes in area contributes to the low-frequency attenuation in duct silencers as well as to the loss in plenums and expansion chambers.

Expansion Chambers and Mufflers

If a plane wave propagates down a duct and enters an area expansion for a certain distance followed by a contraction, there is an acoustical loss. This fundamental shape change is the basis of a reactive (nonlined) muffler used to quiet internal combustion engines and other noise sources, where a dissipative interior liner is not practical. It also furnishes most of the low-frequency loss in small plenums. Figure 8.5 illustrates the configuration.

When a plane wave encounters the changes in area, the pressure and volume velocities must be matched at the two boundaries. This leads to four boundary conditions similar to

FIGURE 8.5 Expansion Chamber Muffler



those in Eq. 8.26, which must be solved for the coefficients (Davis, 1957). At the first boundary

$$\mathbf{p}_{\mathrm{i}\,1} + \mathbf{p}_{\mathrm{r}\,1} = \mathbf{p}_{\mathrm{t}\,2} + \mathbf{p}_{\mathrm{r}\,2} \tag{8.33}$$

$$\left(\mathbf{u}_{\mathrm{i}\,1} - \mathbf{u}_{\mathrm{r}\,1}\right) = m\left(\mathbf{u}_{\mathrm{t}\,2} - \mathbf{u}_{\mathrm{r}\,2}\right) \tag{8.34}$$

where $\mathrm{m} = \mathrm{S}_2/\mathrm{S}_1$. At the second boundary

$$\mathbf{p}_{\mathrm{t}\,2}\, \mathrm{e}^{-\mathrm{j}\,\mathrm{k}\,l} + \mathbf{p}_{\mathrm{r}\,2}\, \mathrm{e}^{\mathrm{j}\,\mathrm{k}\,l} = \mathbf{p}_{\mathrm{t}\,3} \tag{8.35}$$

$$m\left[\mathbf{p}_{\mathrm{t}\,2}\, \mathrm{e}^{-\mathrm{j}\,\mathrm{k}\,l} - \mathbf{p}_{\mathrm{r}\,2}\, \mathrm{e}^{\mathrm{j}\,\mathrm{k}\,l}\right] = \mathbf{p}_{\mathrm{t}\,3} \tag{8.36}$$

When these four equations are solved simultaneously, the transmission coefficient can be obtained from $\boldsymbol{\tau} = \dfrac{\mathbf{p}_{\mathrm{t}\,3}}{\mathbf{p}_{\mathrm{i}\,1}}$

$$\boldsymbol{\tau} = \left[\cos \mathrm{k}\,l + \mathrm{j}\,\frac{1}{2}\left(m + \frac{1}{m}\right)\sin \mathrm{k}\,l\right]^{-1} \tag{8.37}$$

and the transmission loss through the expansion is

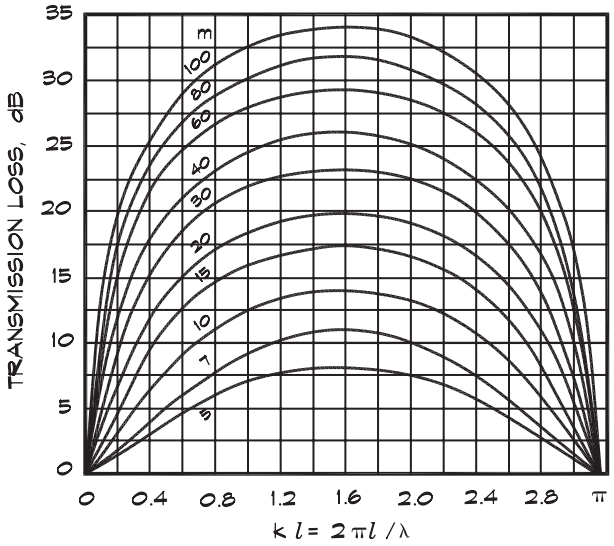
$$\Delta\,\mathrm{L_m} = 10 \log\left[1 + \frac{1}{4}\left(m - \frac{1}{m}\right)^2 \sin^2 \mathrm{k}\,l\right] \tag{8.38}$$

Recall that these equations were obtained by assuming plane waves in all three sections of the duct. For this equation to hold, the lateral dimensions of the chamber must be smaller than $0.8\,\lambda$ (Beranek, 1992). A graph of the attenuation versus normalized length is shown in Fig. 8.6. Notice that the result repeats each time $\mathrm{k}\,l = \pi$.

8.3 SOUND IN ROOMS

The analysis of sound in rooms falls into regions according to the frequency (wavelength) of the sound under consideration. At low frequencies, where the wavelength is greater than twice the length of the longest dimension of the room, only plane waves can be formed and the room behaves like a duct. This condition can occur in very small rooms. Above the cutoff

FIGURE 8.6 Transmission Loss of a Single Expansion Chamber (Beranek, 1971)



Transmission loss of a single expansion chamber of length l and $S_2/S_1 = m$. For values of kl above π use kl mod π.

frequency of a room, normal modes are formed, which are manifest as standing waves having localized regions of high and low pressure. At still higher frequencies the density of modes is so great that there is a virtual continuum in each frequency range and it becomes more useful to model room behavior based on the energy density or other statistical considerations.

Normal Modes in Rectangular Rooms

Let us consider a rectangular room having dimensions l_x, l_y, and l_z. When the room is ensonified and then the sound source removed, certain frequencies persist, much like those that remained in the case of a closed tube. In this case, however, the modes may develop in several directions, since no room dimension is small compared with a wavelength. If we apply the three-dimensional wave equation in rectangular coordinates given as Eq. 6.32 and write a general solution, it takes the form of

$$\mathbf{p} = \mathbf{A}\, e^{j(\omega t - k_x x - k_y y - k_z z)} \tag{8.39}$$

If this expression is substituted into Eq. 6.32, the values for the wave numbers k_x, k_y, and k_z must satisfy the relationship

$$k = \frac{\omega}{c} = \sqrt{k_x^2 + k_y^2 + k_z^2} \tag{8.40}$$

We can replace the negative signs in Eq. 8.39 with one or more positive signs to obtain seven additional equations, which represent the group of waves moving about the room and reflecting off the boundaries. When we apply the boundary conditions as we did for a closed tube, we find that the allowed values of the wave number are

$$k_i = \frac{n_i \pi}{l_i} \tag{8.41}$$

where i refers to the x, y, and z directions. The equation for the sound-pressure standing wave in the room, which has the same form as Eq. 8.6, is separable into three components

$$\mathbf{p} = 8\,\mathbf{A}\cos\left(\frac{n_x\,\pi x}{l_x}\right)\cos\left(\frac{n_y\,\pi y}{l_y}\right)\cos\left(\frac{n_z\,\pi z}{l_z}\right)e^{j\omega t} \tag{8.42}$$

The natural frequencies are

$$f_{\ell\,m\,n} = \frac{c_0}{2}\left[\left(\frac{\ell}{l_x}\right)^2 + \left(\frac{m}{l_y}\right)^2 + \left(\frac{n}{l_z}\right)^2\right]^{\frac{1}{2}} \tag{8.43}$$

where the ℓ, m, and n are integers that indicate the number of nodal planes perpendicular to the x, y, and z axes.

The normal modes of a rectangular room are referenced by whole number indices represented by the three letters ℓ, m, and n. The 1,0,0 mode, for example, would be the fundamental frequency in the x direction. The 2,1,0 mode is a tangential mode in the x and y directions as in Fig. 8.7. If we take a room having dimensions $7 \times 5 \times 3$ m high ($23 \times 16.4 \times 9.8$ ft) and calculate the first few modes, the results would be those in Table 8.1.

Several things are important to notice. If the room dimensions are a low integer multiple of one another, then modal frequencies will coincide. Under these conditions the energy in

FIGURE 8.7 Standing Waves in a Rectangular Room (Bruel and Kjaer, 1978)

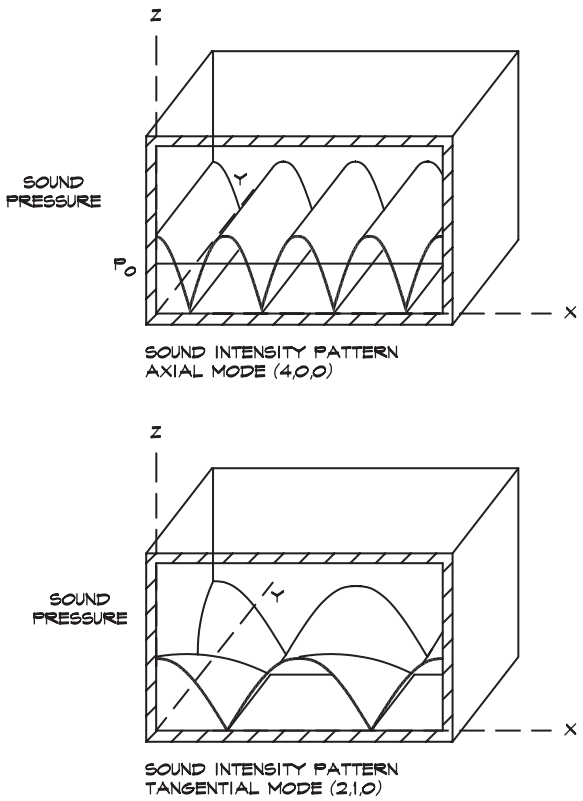


TABLE 8.1 Normal Modes of a Rectangular Room

Number	n_x	n_y	n_z	f_n (Hz)
1	1	0	0	24.6
2	0	1	0	34.5
3	1	1	0	42.4
4	2	0	0	49.2
5	0	0	1	57.4
6	2	1	0	60.1

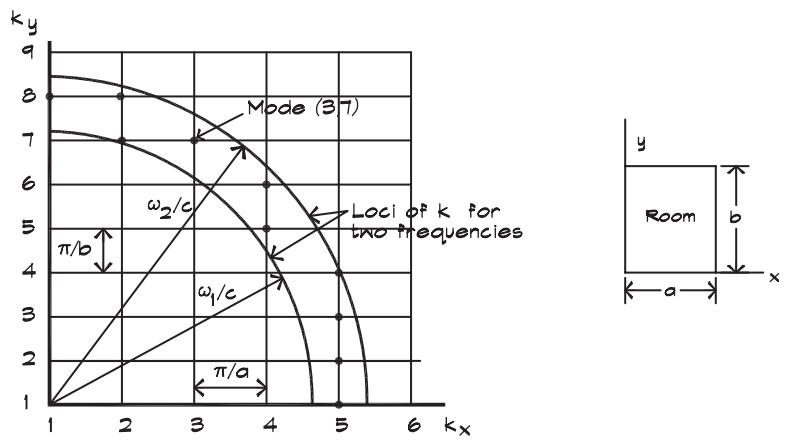
the room will tend to coalesce into a few modes, which will strongly color the sound. Note also that as the frequency increases, the resonances move closer and closer together. They segue from a discrete set of identifiable frequencies into a continuum of modes.

The number of normal modes in a given frequency range can be calculated by plotting the allowed wave numbers in a three-dimensional graph, known as k-space (Fig. 8.8), having dimensions of wave number in the x, y, and z directions. A given value of k in Eq. 8.40 is represented as a point at the intersection of three lattice lines. The total number of frequencies is contained in a sphere having radius k in the positive octant of the sphere divided by the unit volume per k point. The distance between each k_i value is $\frac{\pi}{l_x}$, $\frac{\pi}{l_y}$, or $\frac{\pi}{l_z}$ and the unit volume per k point is $\frac{\pi^3}{V}$, where $V = l_x\, l_y\, l_z$ is the volume of the room. The number of allowed k values between 0 and a given value of k is

$$N_f = \frac{\pi k^3/6}{\pi^3/V} = \frac{4\pi}{3}\left(\frac{f}{c_0}\right)^3 \tag{8.44}$$

In calculating this number we have left out all k values outside of the positive octant, so we need to add back a correction for frequencies located on the axis planes, which are counted

FIGURE 8.8 Normal Modes Falling between Two Frequencies



as one half, and on the axes themselves, which are counted as one quarter of their actual value. This yields

$$N_f = \frac{4\pi}{3} V \left(\frac{f}{c_0}\right)^3 + \frac{\pi}{4} S \left(\frac{f}{c_0}\right)^2 + \frac{L}{8}\left(\frac{f}{c_0}\right) \tag{8.45}$$

where S is the area of all the walls and L is the sum of all the edge lengths.

The number of modes in a given frequency range can be determined by taking the derivative of Eq. 8.45 with respect to frequency,

$$\frac{d\,N_f}{d\,f} = 4\pi V \frac{f^2}{c_0^3} + \frac{\pi}{2} S \frac{f}{c_0^2} + \frac{L}{8\,c_0} \tag{8.46}$$

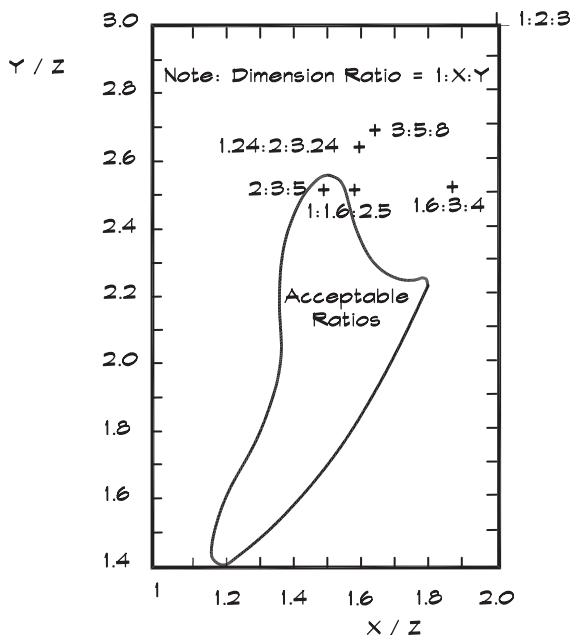
At high frequencies the density of modes is extremely large—for all practical purposes, a continuum. For the room dimensions given in Fig. 8.8, at 1000 Hz, the modal density is 34 modes per Hertz.

Preferred Room Dimensions

A number of authors, including Bolt, Rettinger, and others have offered suggestions on preferred room dimensions for listening rooms and studios. These are given in terms of the ratios of the lengths of the sides of a rectangular room. The one published by Bolt is shown in Fig. 8.9.

FIGURE 8.9 Preferred Dimensions of a Rectangular Room (Bolt, 1946)

The curve encloses dimension ratios of width to length of a rectangular room having a unity ceiling height to give a smooth response at low frequencies.



Recommendations such as those shown in Fig. 8.9 are most useful in designing reverberation chambers for acoustical-testing purposes, when a rectangular room is desired. They could be useful in the design of small studios; however, sound studios are rarely built in a rectangular shape. Normal-mode calculations for nonrectangular rooms are more difficult and can be done using finite element methods.

8.4 DIFFUSE-FIELD MODEL OF ROOMS

In a room whose dimensions are large enough that there is a sufficient density of modes, it is customary to describe the space in terms of a statistical model known as a diffuse field. A diffuse field is one in which there is an equal energy density at all points in the room. This implies that there is an equal probability that sound will arrive from any direction.

Schroeder Frequency

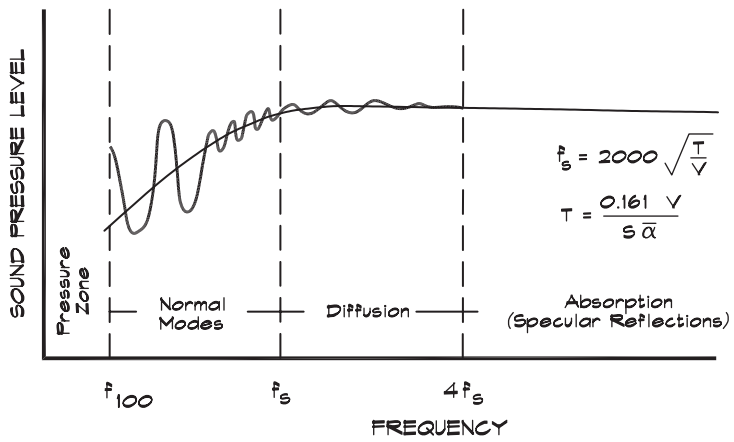
The transition between the normal-mode model and the statistical model is not a bright line, but is generally taken to occur at a modal spacing that has at least three modes within a given mode's half-power bandwidth. This point is marked by the so-called *Schroeder frequency* (Schroeder, 1954 and 1996), which is defined in metric units as

$$f_s = 2000\sqrt{\frac{T}{V}} \tag{8.47}$$

where V is the volume of the room and T is the reverberation time. In FP units, where V is in cubic feet, the multiplication constant is 12,000. Above the Schroeder frequency it is appropriate to analyze the room without having to take into account the behavior of its normal modes.

The Schroeder frequency allows us to subdivide the room behavior into regions. Figure 8.10 (Davis and Davis, 1987) shows a plot of the type of behavior we can expect from the sound pressure level in a room plotted against frequency. It also indicates the techniques that can be used to control the steady state room response. In small rooms, where the normal mode region can extend to several hundred Hertz, a statistical model can only be used at relatively high frequencies.

FIGURE 8.10 Controllers in Steady State Room Response (Davis and Davis, 1987)



Mean Free Path

Above the Schroeder frequency, sound waves in a room can be treated as rays or particles in terms of their reflections off the room's surfaces. This can be done by following the ray path around a room and studying its interaction with the walls, or by constructing an image source for each reflecting surface and summing their contributions at the receiver. If we follow a sound ray around a room it will travel until it encounters a surface. The sound particle travels a distance (c_0 t) in time t and if it undergoes N collisions during that time, the average distance between collisions is

$$\Lambda = \frac{c_0\ t}{N} = \frac{c_0}{n} \tag{8.48}$$

where n is the average number of collisions per unit time and Λ is the *mean free path* between collisions. Knudsen (1932) determined the mean free path experimentally for a number of differently shaped rectangular rooms, and others (Kuttruff, 1973) have derived the equation from first principles. The result is

$$\Lambda = \frac{4\ V}{S_T} \tag{8.49}$$

where V is the volume and S_T is the total surface area of the room. Here both Λ and n are averages and Eq. 8.49 holds only for diffuse-field conditions. Using Eq. 8.48 the average collision frequency can be obtained

$$n = \frac{c_0\ S_T}{4\ V} \tag{8.50}$$

Decay Rate of Sound in a Room

The reciprocal of the average collision frequency is the mean time between collisions, which for a diffuse field is

$$t = \frac{4\ V}{c_0\ S_T} \tag{8.51}$$

If we ride along with the sound ray, the energy density in the vicinity of the ray after each reflection, shown in Fig. 8.11, is

$$\begin{aligned} D\ (t) &= D_0\ (1 - \overline{\alpha}) \\ D\ (2\,t) &= D_0\ (1 - \overline{\alpha})^2 \\ D\ (\mathrm{n}\,t) &= D_0\ (1 - \overline{\alpha})^{\mathrm{n}} \end{aligned} \tag{8.52}$$

The total number of reflections n is the total time divided by the mean time between reflections, so we can write

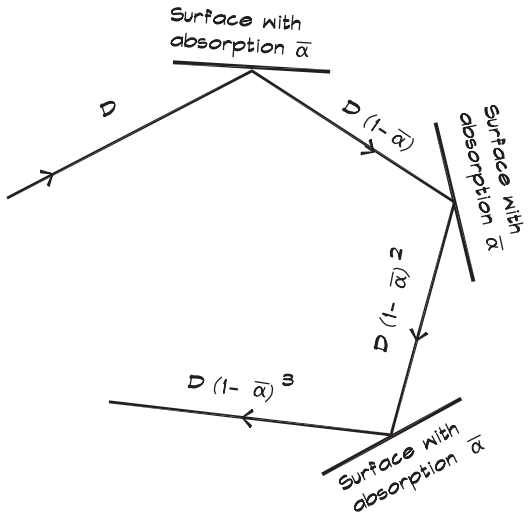
$$D\ (\mathrm{t}) = D_0\ (1 - \overline{\alpha})^{(c_0\ S_T / 4\ V)\ t} \tag{8.53}$$

Now using the identity

$$(1 - \overline{\alpha}) = \mathrm{e}^{\ln\ (1\ -\ \overline{\alpha})} \tag{8.54}$$

FIGURE 8.11 Path of a Sound Ray Having Energy Density D

Path of a sound wave modelled as a ray as it reflects off of surfaces having an average absorption coefficient $\overline{\alpha}$. The surfaces are spaced a distance equal to the mean free path $l = 4\ V/S_T$ apart.



we obtain

$$D\,(\mathrm{t}) = D_0\, \mathrm{e}^{-(c_0\ S_T/4\ V)\,[-\ln\,(1-\overline{\alpha})]\,t} \tag{8.55}$$

Multiplying each side by the speed of sound, we can convert this equation into a sound pressure level as a function of time

$$L_p(t) = L_p\ (t = 0) - 4.34\ \frac{c_0\ S_T}{4V}\ [-\ln\,(1-\overline{\alpha})]\,t \tag{8.56}$$

which is the rate of decay of sound in a room under diffuse-field conditions. Note that the average absorption coefficient can be expressed as a weighted average of the individual coefficients for each of the surfaces of the room.

$$\overline{\alpha} = \frac{S_1\ \alpha_1 + S_2\ \alpha_2 + S_3\ \alpha_3 + \cdots + S_n\alpha_n}{S_T} \tag{8.57}$$

Sabine Reverberation Time

The idea that there exists a characteristic time for sound to die out in a room originated with Wallace Clement Sabine. When he undertook the task of correcting a problem of unintelligible speech in the Fogg Art Museum lecture hall at Harvard College; the sound in the room would persist for over 5 seconds. Because an English-speaking person can complete about 15 syllables in that time, most of the words were impossible to understand (Egan, 1988). Sabine measured the *reverberation time*, the time it took for the sound level to drop 60 dB, for varying amounts of absorptive materials. He borrowed 3-inch thick seat cushions from the nearby Sanders Theater and found that the more cushions he placed around the room, the more quickly the sound would die out. When 550 cushions were arranged in the

space on the platform, seats, aisles and the rear wall, the reverberation time had decreased to about 1 second. The empirical formula he discovered, now called the Sabine reverberation time, is

$$T_{60} = .049 \frac{V}{A} \tag{8.58}$$

where T_{60} = reverberation time,or the time it takes for sound to decrease by 60 dB in a room (s)
V = volume of the room (cu ft)
A = total area of absorption in the room (sabins)
$= S_1 \alpha_1 + S_2 \alpha_2 + S_3 \alpha_3 + \cdots + S_n \alpha_n$

The standard unit of absorption, now called the sabin in his honor, has units of sq ft. The metric sabin has units of sq m. In metric units the Sabine formula is

$$T_{60} = 0.161 \frac{V}{A} \tag{8.59}$$

Norris Eyring Reverberation Time

Carl Eyring published (1930) a theory of reverberation time in rooms based on an idea that was attributed to R. F. Norris (Andree, 1932). Using the arguments that lead to Eq. 8.56, he set the difference in level to 60 dB and calculated the resulting decay time

$$T_{60} = \frac{4\,V\,(60)}{-4.34\,c_0\,S_T\,\ln(1-\overline{\alpha})} = \frac{0.161\,V}{-S_T\,\ln(1-\overline{\alpha})} \tag{8.60}$$

where the volume is in metric units. In FP units the equation is

$$T_{60} = \frac{.049\,V}{-S_T\,\ln(1-\overline{\alpha})} \tag{8.61}$$

This equation is more accurate in dead (very absorptive) rooms than the Sabine equation. For example, in a perfectly absorbing room with a given area of absorption, the Sabine equation will give a nonzero result whereas the Eyring equation will correctly give zero. Care must be exercised in using the Norris Eyring equation with absorption coefficients measured with the Sabine equation, since occasionally an absorption coefficient greater than one is obtained. All Norris Eyring coefficients must be less than one.

Derivation of the Sabine Equation

When value of the average absorption coefficient is small, that is when we have an acoustically live space, we can use a series expansion for the natural logarithm,

$$\ln(x) = (x-1) - \frac{1}{2}(x-1)^2 + \frac{1}{3}(x-1)^3 - \ldots$$

to obtain an approximation for small ($\overline{\alpha} < 0.2$) values of the average absorption coefficient,

$$-\ln(1-\overline{\alpha}) \cong \overline{\alpha} \tag{8.62}$$

This leads us back to the Sabine equation, which is accurate under these conditions. The Sabine equation is the preferred formula for use in normal rooms and auditoria.

Millington Sette Equation

The average absorption coefficient for a given room is usually not readily available, but may be calculated from the absorption coefficients of the individual surfaces (Millington, 1932 and Sette, 1933)

$$S_T\,\overline{\alpha} = \sum_{i=1}^{n} S_i\,\alpha_i \tag{8.63}$$

which was adopted by Eyring.

$$T_{60} = \frac{0.161\,V}{-S_T\,\ln\left[1 - \sum\left(S_i\,\alpha_i / S_T\right)\right]} \tag{8.64}$$

Highly Absorptive Rooms

When the average absorption coefficient for a room is large ($\overline{\alpha} > 0.5$), another series expansion for the natural logarithm can be applied to the Norris Eyring equation, namely

$$\ln(x) = \frac{x-1}{x} + \frac{1}{2}\left(\frac{x-1}{x}\right)^2 + \frac{1}{3}\left(\frac{x-1}{x}\right)^3 + \ldots$$

to obtain

$$-\ln\left(1 - \overline{\alpha}\right) \cong \frac{\overline{\alpha}}{1 - \overline{\alpha}} \tag{8.65}$$

which yields the absorbent room approximation for the reverberation time

$$T_{60} \cong \frac{0.161V}{S_T\left(\dfrac{\overline{\alpha}}{1 - \overline{\alpha}}\right)} \tag{8.66}$$

This equation is limited to relatively high values of the average absorption coefficient. Neither Eq. 8.66 or the Norris Eyring equation can be used when average absorption coefficients exceed a value of one—a relatively common occurrence since we use the Sabine formula to measure absorption.

Air Attenuation in Rooms

In Chapt. 4 we discussed how sound is attenuated as it moves through the atmosphere due to its interaction with the air molecules. If we assume there is an attenuation constant that characterizes the atmospheric loss in terms of so much per distance, we can write an equation in terms of the loss, m, in energy density per meter. After a given time, say the mean time between collisions, the ray has traveled a certain distance, in this case the mean free path, and the resulting energy density has the form

$$D(t) = D_0\,e^{-m\Lambda} \tag{8.67}$$

By inserting this relationship into Eq. 8.53 and recalculating Eq. 8.55 we get

$$D(t) = D_0 e^{-(c_0 S_T / 4V)\left[-\ln(1-\bar{\alpha}) - 4mV/S_T\right]t} \tag{8.68}$$

and the Norris Eyring reverberation time with air attenuation becomes

$$T_{60} = \frac{0.161\ V}{-S_T \ln(1-\bar{\alpha}) + 4mV} \tag{8.69}$$

in metric units and

$$T_{60} = \frac{.049\ V}{-S_T \ln(1-\bar{\alpha}) + 4mV} \tag{8.70}$$

in FP units. In terms of loss in dB/m or dB/ft the relationship is

$$m = \frac{\Delta L_{air}}{4.34} \tag{8.71}$$

where m has units of inverse meters or feet.

Air losses can be included in the Sabine reverberation time formulas, which devolve from the Eqs. 8.68 and 8.69, in the limit of small values of the average absorption coefficient. In metric

$$T_{60} = \frac{0.161\ V}{A + 4mV} \tag{8.72}$$

and in FP units,

$$T_{60} = \frac{.049\ V}{A + 4mV} \tag{8.73}$$

The total absorption in a room, including the air absorption, is called the room constant and is given the designation R

$$R = A + 4mV \tag{8.74}$$

The units are in sabins or metric sabins.

Laboratory Measurement of the Absorption Coefficient

It is standard practice (ASTM C423) to measure the absorption of an architectural material in a reverberant test chamber using the reverberation time method. A reverberation chamber is a hard room with concrete surfaces and a long reverberation time, with sufficient volume to have an adequate density of modes at the frequency of interest. Since the average absorption coefficient in the room is quite small under these conditions, the Sabine equation can be used. The reverberation time of the empty chamber is

$$T_{60}(1) = \frac{0.161\ V}{S_T\ \bar{\alpha}} \tag{8.75}$$

If a sample of absorptive material having an area S_1 is placed on the floor and the test repeated, the new reverberation time is

$$T_{60}(2) = \frac{0.161\ V}{S_T\ \overline{\alpha} - S_1\alpha_0 + S_1\alpha_1} \tag{8.76}$$

where α_0 is the absorption coefficient of the covered portion of the floor and α_1 is the absorption coefficient of the sample material under test. Combining Eqs. 8.75 and 8.76 we obtain the desired coefficient

$$\alpha_1 = \alpha_0 + \frac{0.161\ V}{S_1}\left(\frac{1}{T_{60}(2)} - \frac{1}{T_{60}(1)}\right) \tag{8.77}$$

In these tests there is some dependence on the position of the sample in the room. Materials placed in the center of a surface are more effective absorbers, and yield higher absorption coefficients, than materials located in the corners. This is because the average particle velocity is higher there. There are also diffraction effects and edge absorption attributable to the sides of the sample. For these reasons Sabine absorption coefficients that are greater than one sometimes are obtained and must be used with caution in the Norris Eyring equation.

8.5 REVERBERANT FIELD EFFECTS

Energy Density and Intensity

We have seen that, as the modal spacing gets closer and closer together, it becomes less useful to consider individual modes and we must seek other ways of describing the behavior of sound in a room. One concept is the energy density . A plane wave moves a distance c_0 in one second and carries an energy per unit area equal to its intensity, I. The direct-field energy density D_d per unit volume is

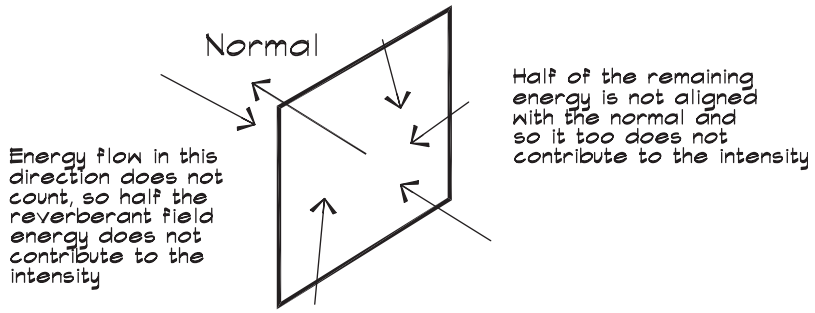
$$D_d = \frac{I_d}{c_0} = \frac{p^2}{\rho_0\, c_0^2} \tag{8.78}$$

where p^2 is the rms acoustic pressure.

The energy density in a diffuse field has the same relationship to the pressure squared, which is not a vector quantity, but a different relationship to the intensity. In a diffuse field the sound energy can be coming from any direction. The intensity is defined as the power passing through an area in a given direction. In a diffuse field, half the energy is passing through the area plane in the opposite direction to the one of interest. When we integrate the energy incident on the area in the remaining half sphere, the cosine term reduces the intensity by another factor of two. Thus in a reverberant field the intensity is only a quarter of the total power passing through the area. This is shown in Fig. 8.12.

$$I_r = \frac{1}{4}\left(\frac{p^2}{\rho_0\, c_0}\right) \tag{8.79}$$

FIGURE 8.12 Intensity in a Reverberant Field



Semireverberant Fields

Occasionally we encounter a semireverberant field, where energy falls onto one side of a plane with equal probability from any direction. Most often this occurs when sound is propagating from a reverberant field through an opening in a surface of the room. Under these conditions the power passing through the plane of an opening having area S_w is given by

$$W_{sr} = \frac{S_w}{2}\left(\frac{p^2}{\rho_0 c_0}\right) \tag{8.80}$$

Room Effect

When a sound source that emits a sound power W_S is placed in a room, the energy density will rise until the energy flow is balanced between the energy being created by the source and the energy removed from the room due to absorption. After a long time the total energy in a room having a volume V due to a source having a sound power W_S is

$$V D_r = \frac{W_S \Lambda}{c_0}\left[1 + (1-\overline{\alpha}) + (1-\overline{\alpha})^2 + \cdots\right] = \frac{W_S \Lambda}{c_0 \overline{\alpha}} \tag{8.81}$$

which has been simplified using the limit of a power series for $\overline{\alpha}^2 < 1$

$$V D_r = \frac{4 W_S V}{c_0 S_T \overline{\alpha}} \tag{8.82}$$

and the sound pressure in the room will be

$$\frac{p^2}{\rho_0 c_0} = \frac{4 W_S}{S_T \overline{\alpha}} = \frac{4 W_S}{R} \tag{8.83}$$

Equation 8.83 is the reverberant-field contribution to the sound pressure measured in a room and can be combined with the direct-field contribution to obtain

$$\frac{p^2}{\rho_0 c_0} = \frac{Q W_S}{4\pi r^2} + \frac{4 W_S}{R} \tag{8.84}$$

Taking the logarithm of each side we can express this equation as a level

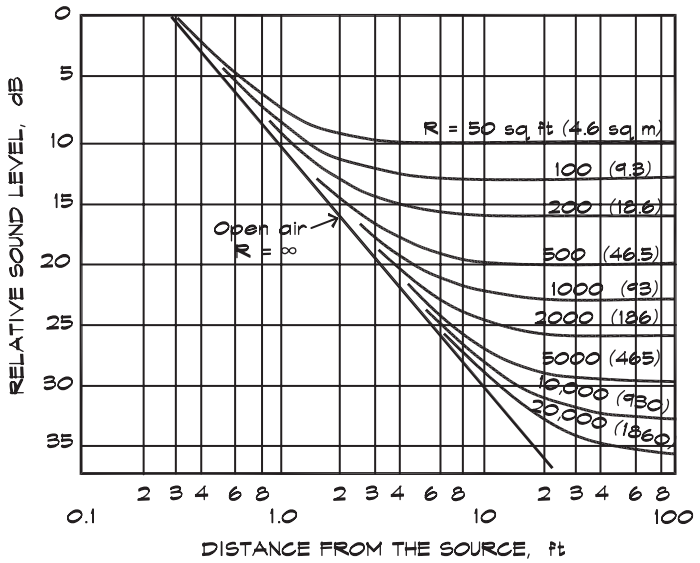
$$L_p = L_W + 10 \log \left[\frac{Q}{4 \pi r^2} + \frac{4}{R} \right] + K \tag{8.85}$$

where K is 0.1 for metric units and 10.5 for FP units. The numerical constants follow from the reasoning given in Eq. 2.67.

Equation 8.85 is based on Sabine's theory and was published in 1948 by Hopkins and Stryker. It is a useful workhorse for the calculation of the sound level in a room given the sound power level of one or more sources. It holds reasonably well where the diffuse field condition exists; that is, in relatively large rooms with adequate diffusion if we are not too close (usually within $\frac{\lambda}{2}$) to reflecting surfaces. The increase in sound pressure level due to the reverberant field over that which we would expect from free field falloff is called the room effect.

Figure 8.13 gives the result from Eq. 8.85 for various values of the room constant. Near the source the direct-field contribution is larger than the reverberant-field contribution and the falloff behavior is that of a point source in a free field. In the far field the direct-field contribution has dropped below the reverberant-field energy, and the sound pressure level is constant throughout the space. The level in the reverberant field can be reduced only by adding more absorption to the room. According to this theory, only the total amount of absorption is important, not where it is placed in the room. In practice absorption placed where the particle velocity is the highest has the greatest effect. Thus absorption mounted in a corner, where the pressure has a maximum and the velocity a minimum, would be less effective than absorption placed in the middle of a wall or other surface. Absorption, which is hung in the center of a space, has the greatest effect but this is not a practical location.

FIGURE 8.13 Difference between Sound Power and Pressure Level in a Diffuse Room Due to an Omnidirectional Source



At a given distance, known as the *critical distance*, the direct-field level equals the reverberant-field level. We can solve for the distance by setting the direct and reverberant contributions equal.

$$r_c = \sqrt{\frac{Q\,R}{16\,\pi}} \tag{8.86}$$

Beyond the critical distance the reverberant field predominates.

Radiation from Large Sources

When the source of sound is physically large, such as the wall of a room, it can radiate energy over its entire surface area. The idea of a displaced center was introduced in Eq. 2.91 to relate the sound power to the sound pressure level in free space for a receiver located close to a large radiating surface. Similarly in a reverberant space the direct and reverberant contributions are combined

$$L_p = L_W + 10\,\log\left[\frac{Q}{4\,\pi\left[z + \sqrt{\frac{S\,Q}{4\,\pi}}\right]^2} + \frac{4}{R}\right] + K \tag{8.87}$$

where K is 0.5 for metric and 10.5 for FP units.

As the distance z, between the surface of the source and the receiver, is reduced to zero, Eq. 8.87 can be simplified to

$$L_p \cong L_W + 10\log\left[\frac{1}{S} + \frac{4}{R}\right] + K \tag{8.88}$$

where S is the surface area of the source. When the receiver is far from the source the area contribution is small and the distance to the surface of the source and to its acoustic center are nearly equal ($z \cong r$). The equation then reverts to its previous form

$$L_p = L_W + 10\log\left[\frac{Q}{4\,\pi\,r^2} + \frac{4}{R}\right] + K \tag{8.89}$$

Departure from Diffuse Field Behavior

In the power-pressure conversion, when we do not measure the sound pressure level close to the reflecting surfaces, we neglect some energy near the boundary given in Eq. 8.74. Waterhouse (1955) has investigated this energy and has suggested the addition of a correction term to the room constant, which is only significant at low frequencies.

$$R = A\left(1 + \frac{S_T\,\lambda}{8\,V}\right) + 4\,m\,V \tag{8.90}$$

where S_T is the total surface area and V the volume of the room. The correction is used in certain test procedures (e.g., ISO 3741 and ASTM E336).

FIGURE 8.14 Measured (Power – Pressure) Level Differences (Davis and Davis, 1978)

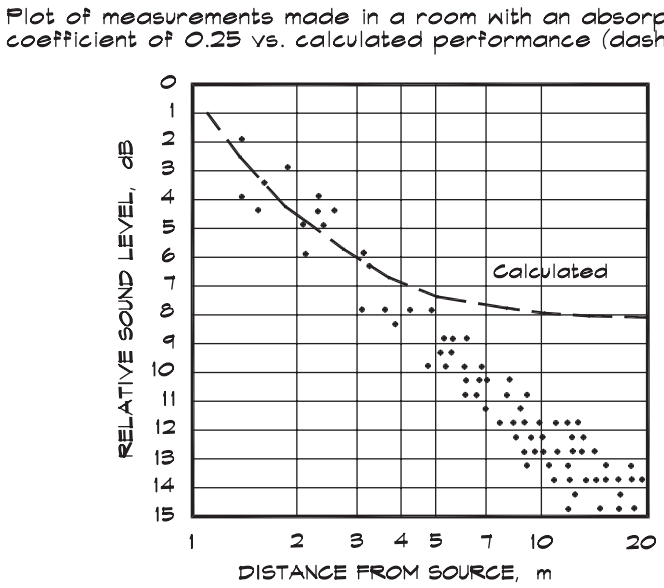
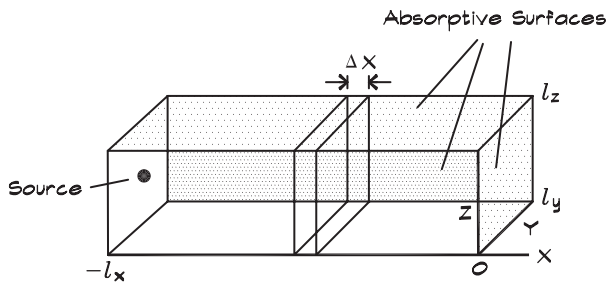


FIGURE 8.15 Rectangular Room with Source Wall ($x = -\ell_x$), Absorbing End-Wall ($x = 0$), and Absorbing Side Surfaces ($y = \ell_y$ and $z = \ell_z$) (Franzoni, 2001)



When rooms have a significant dimensional variation in different directions, particularly where there are low ceilings with a large amount of absorption, there is a departure from the behavior predicted by the Hopkins Stryker equation. Figure 8.14 shows measurements taken by Ogawa (1965) in Japan. A number of authors have attempted to account for this behavior by adding additional empirical terms or multipliers to the equation. Hodgson (1998) has published a review of several of these methods.

Franzoni and Labrozzi (1999) developed an empirical formula that applies to long, narrow, rectangular rooms, when the absorption is not uniformly distributed on all surfaces. For a source positioned near one wall and the geometry shown in Fig. 8.15,

$$\overline{p}^2_{rev} = \frac{4\,\rho_0\,c_0\,W}{A}\left[\frac{(1-\overline{\alpha}_{total})(1-\overline{\alpha}_{total}/2)}{(1-\overline{\alpha}_w\,\overline{S}/2)}\right] e^{-(1/2)\,\overline{\alpha}_w\,\overline{S}\overline{x}} \tag{8.91}$$

where $\overline{p}^2_{rev}$ = cross-sectionally averaged mean square acoustic pressure (Pa^2) at a distance x from the origin (not the source)
A = total area of absorption in the room (sabins)
$= S_1 \alpha_1 + S_2 \alpha_2 + S_3 \alpha_3 + \cdots + S_n \alpha_n$
$\overline{\alpha}_{total} = A / S_{total}$
$\overline{\alpha}_w$ = total absorption of the side surfaces divided by the area of the side surfaces
$\overline{x} = x / l_x$
$\overline{S}$ = ratio of the side wall surface area to the cross sectional surface area

Reverberant Falloff in Long Narrow Rooms

Franzoni (2001) also published a theoretical treatment of the long-narrow room problem by considering an energy balance for diffuse-field components traveling to the right and to the left using the geometry in Fig. 8.15. She assumes that there is a locally diffuse condition, where energy incidence is equally probable in all directions from a hemisphere at a planar slice across the room, but the rightward energy does not necessarily equal the leftward energy. The total energy at a point is taken to be uncorrelated and can be expressed as the sum of the two directional components

$$\overline{p}^{\,2} = \overline{p}^{\,2}_{+x} + \overline{p}^{\,2}_{-x} \tag{8.92}$$

At a given slice the reverberant intensity, due to rightward moving waves, is

$$I_{+x} = \frac{\overline{p}^{\,2}_{+x}}{\rho_0 c_0} \tag{8.93}$$

and similarly for the leftward moving waves.

To evaluate the effect of reflections from the side surfaces we write the mean square pressure near the wall as the sum of the incident and reflected components interacting with the sides

$$\overline{p}^{\,2}_{+x} = \overline{p}^{\,2}_{+x_{incident}} + \overline{p}^{\,2}_{+x_{incident}}(1 - \alpha_w) = (2 - \alpha_w)\overline{p}^{\,2}_{+x_{incident}} \tag{8.94}$$

The incident intensity into the side wall boundary (y or z) is

$$I_{sidewall} = I_s = I_y = I_z = \frac{\overline{p}^{\,2}_{+x_{incident}}}{2\rho_0 c_0} = \frac{\overline{p}^{\,2}_{+x}}{2\rho_0 c_0}\left[\frac{1}{2 - \alpha_w}\right] \tag{8.95}$$

where $\overline{p}^{\,2}_{+x}$ = mean square pressure associated with rightward traveling waves, incident plus reflected.

If we define β as the fraction of the surface area at a cross section, covered with an absorbing material having a random incidence absorption coefficient α_w, and l_p and S as the perimeter and area of the cross section, we can write a power balance relation equating the power in to the power out of the cross section.

$$I_x S = \left(I_x + \frac{d I_x}{d x}\Delta x\right) S + \alpha_w \beta l_p \Delta x I_s \tag{8.96}$$

This can be written as a differential equation

$$\frac{\mathrm{d}\left(\overline{\mathrm{p}}_{+\mathrm{x}}^{2}\right)}{\mathrm{d}\,\mathrm{x}} + \frac{\alpha_{\mathrm{w}}\,\beta\,l_{\mathrm{p}}}{(2-\alpha_{\mathrm{w}})\,S}\;\overline{\mathrm{p}}_{+\mathrm{x}}^{2} = 0 \tag{8.97}$$

which has a solution for right-running waves

$$\overline{\mathrm{p}}_{+\mathrm{x}}^{2} = \mathrm{P}_{+\mathrm{x}}\mathrm{e}^{-(\alpha_{\mathrm{w}}\,\beta\,l_{\mathrm{p}})/((2-\alpha_{\mathrm{w}})\,S)\,\mathrm{x}} \tag{8.98}$$

and another for left-running waves

$$\overline{\mathrm{p}}_{-\mathrm{x}}^{2} = \mathrm{P}_{-\mathrm{x}}\mathrm{e}^{+(\alpha_{\mathrm{w}}\,\beta\,l_{\mathrm{p}})/((2-\alpha_{\mathrm{w}})\,S)\,\mathrm{x}} \tag{8.99}$$

where $\mathrm{P}_{-\mathrm{x}}$ and $\mathrm{P}_{+\mathrm{x}}$ are coefficients to be determined by the boundary conditions at each end. At the absorbing end ($\mathrm{x} = 0$) the right and left intensities are related

$$\mathrm{I}_{-\mathrm{x}}(0) = (1-\alpha_{\mathrm{b}})\,\mathrm{I}_{+\mathrm{x}}(0) \tag{8.100}$$

with α_{b} being the end wall random incidence absorption coefficient. The coefficients in Eqs. 8.98 and 8.99 are related

$$\mathrm{P}_{-\mathrm{x}} = (1-\alpha_{\mathrm{b}})\,\mathrm{P}_{+\mathrm{x}} \tag{8.101}$$

At the source-end wall, the power of the sources is equal to the power difference in right and left traveling waves

$$\mathrm{W} = S\left[\mathrm{I}_{+\mathrm{x}}(-l_{\mathrm{x}}) - \mathrm{I}_{-\mathrm{x}}(-l_{\mathrm{x}})\right] \tag{8.102}$$

Plugging in the mean square pressure terms and using Eq. 8.101 (Franzoni, 2001),

$$\overline{\mathrm{p}}^{\,2}(\mathrm{x}) = \frac{2\,\rho_0\,\mathrm{c}_0\,\mathrm{W}}{S}\left[\frac{\cosh(\gamma\mathrm{x}) - \dfrac{1}{2}\,\alpha_{\mathrm{b}}\,\mathrm{e}^{+\gamma\mathrm{x}}}{\sinh(\gamma\,l_{\mathrm{x}}) + \dfrac{1}{2}\,\alpha_{\mathrm{b}}\,\mathrm{e}^{-\gamma l_{\mathrm{x}}}}\right] \tag{8.103}$$

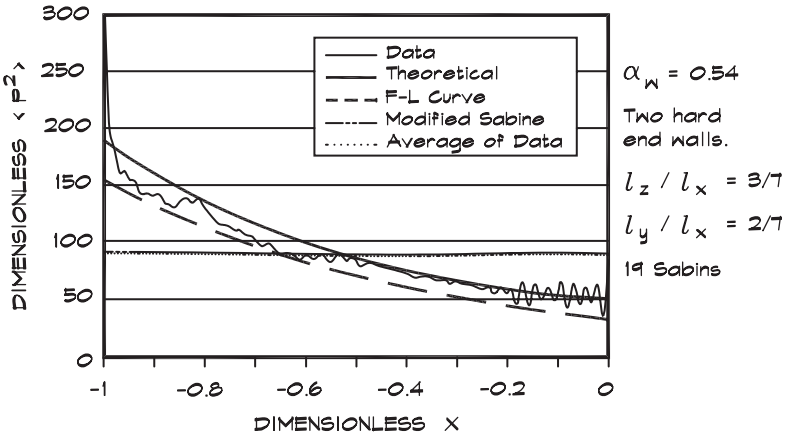
where $\gamma = \alpha_{\mathrm{w}}\,\beta\,l_{\mathrm{p}}/\left[(2-\alpha_{\mathrm{w}})\,S\right]$. Although this formula is somewhat more complicated than Eq. 8.91 it is still straightforward to use.

The result given by Eqs. 8.91 and 8.103 can be compared to more detailed calculations in Fig. 8.16. The agreement is good for both equations. Franzoni (2001) gives several other examples for different absorption coefficients, which also yield good agreement.

Reverberant Energy Balance in Long Narrow Rooms

An energy balance must still be maintained, where the energy produced by the source is absorbed by the materials in the room. In the Sabine theory, the balance is expressed as Eq. 8.83 and the reverberant field energy is assumed to be equally distributed throughout the room. In Franzoni's modified Sabine approach, the average reverberant field energy is the same as Sabine's, but the distribution is uneven. The average energy can be obtained either by integrating Eq. 8.103 over the length of the room or from the following arguments.

FIGURE 8.16 Comparison of Falloff Data—Empirical Fit and Theoretical (Franzoni, 2001)



The power removed from the room is

$$W_{out} = \sum_{\substack{\text{absorbing} \\ \text{surfaces, i}}} I_{\substack{\text{into} \\ \text{surface}}} \alpha_i S_i \tag{8.104}$$

The intensity incident on a surface is due to both the direct and reverberant-field components. From Eq. 8.94 the reverberant energy into a boundary surface is

$$I_r = \frac{\overline{p}^2_{\text{incident}}}{2\rho_0 c_0} = \frac{\overline{p}^2}{2\rho_0 c_0} \frac{1}{(2-\alpha_i)} \tag{8.105}$$

and the average direct-field energy is

$$I_d = \frac{W_{in}}{S_{total}} \tag{8.106}$$

The power removed by the absorbing surfaces is

$$W_{out} = \sum_i \frac{\overline{p}^2}{2\rho_0 c_0 (2-\alpha_i)} \alpha_i S_i + \sum_i \frac{W_{in}}{S_{total}} \alpha_i S_i \tag{8.107}$$

which in terms of the average mean square pressure is the modified Sabine equation (Franzoni, 2001)

$$\overline{p}^2_{\substack{\text{spatial} \\ \text{average}}} = \sum_i \frac{4 W_{in} \rho_0 c_0}{\sum_i \alpha_i S_i/(1-\alpha_i/2)} \left(1 - \sum_i \alpha_i S_i / S_{total}\right) \tag{8.108}$$

When the same absorption coefficient applies to all surfaces this simplifies to

$$\overline{p}^{\,2}_{\substack{\text{spatial}\\ \text{average}}} = \frac{4\,W_{in}\,\rho_0\,c_0}{A}(1-\alpha/2)\,(1-A/S_{total}) \tag{8.109}$$

The first term in the parentheses is a correction to the Sabine formula for the difference between the incoming and outgoing waves, and the second term is the power removed by the first reflection. Figure 8.16 also shows the results to be quite close to exact numerical simulations of the sound field.

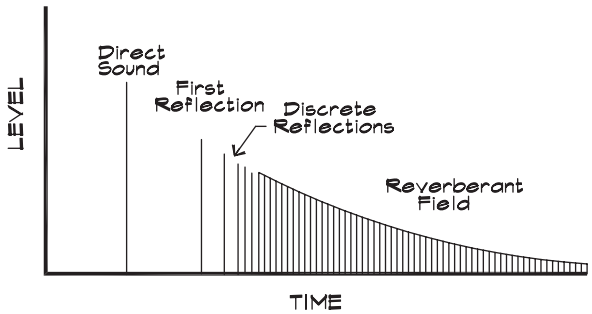
Fine Structure of the Sound Decay

When an impulsive source such as a gunshot, bursting balloon, or electronically induced pulse excites a room with a brief impulsive sound, the room response contains a great deal of information about the acoustic properties of the space. First there is the initial sound decay in the first 10 to 20 msec of drop after the initial burst. The reverberation time based on this region is called the early decay time (EDT) and it is the time we react to. After the first impulse there is a string of pulses, which are the reflections from surfaces nearest the source and receiver. Thereafter follows a complicated train of pulses, which are the first few orders of reflections from the room surfaces. In this region the acoustical defects present in the room begin to appear. Long-delayed reflections show up as isolated pulses. Flutter echoes appear as repeated reflections that do not die out as quickly as the normal reverberant tail. Focusing can cause sound concentrations, which increase the reflected sound above the initial impulse. If the energy-time behavior of the room is filtered, it can be used to explore regions where modal patterns have formed and can contribute to coloration. A typical graph is shown in Fig. 8.17.

When two rooms are acoustically coupled the reverberation pattern in one room affects the sound in the other. When one has a longer reverberation time it may lead to a dual-slope reverberation pattern in the other. Consequently it is good practice to match the decay patterns of adjacent rooms unless it is the purpose to use one to augment the reverberant tail of the other.

FIGURE 8.17 Energy vs Time for an Impulsive Source

Sound pressure level at a point in a room for an impulsive sound. The direct sound arrives first followed by discrete reflections separated in time. Multiple reflections merge to become the reverberant field.



The room resonances that we examined in earlier sections contribute to the long-term behavior of the sound field. If an initial source of sound is turned on, the direct-field energy reaches a listener first, followed closely by the early reflections and lastly by the reverberant field. In the low frequencies the reverberant field is colored by the room modes where energy is preferentially stored. These modes build up and persist longer than nonresonant sound fields. The early reflections are determined by the position and orientation of reflecting surfaces near the source, whereas the reverberant field is defined by the total absorption and position of materials in the room, by the presence of diffusion in the space, and the room modes by the room size and surface orientation. By controlling these variables we can shape the room response according to its use.

This page intentionally left blank

9 SOUND TRANSMISSION LOSS

9.1 TRANSMISSION LOSS

Sound Transmission Between Reverberant Spaces

The transmission of sound from one space to another through a partition is a subject of some complexity. In the simplest case, there are two rooms separated by a common wall having area S_w, as in Fig. 9.1. If we have a diffuse sound field in the source room that produces a sound pressure p_s and a corresponding intensity

$$I_s = \frac{p_s^2}{4\,\rho_0\,c_0} \tag{9.1}$$

which is incident on the transmitting surface, a fraction τ of the incident power is transmitted into the receiving room through the wall

$$W_r = I_s\,S_w\,\tau = \frac{p_s^2\,S_w\,\tau}{4\,\rho_0\,c_0} \tag{9.2}$$

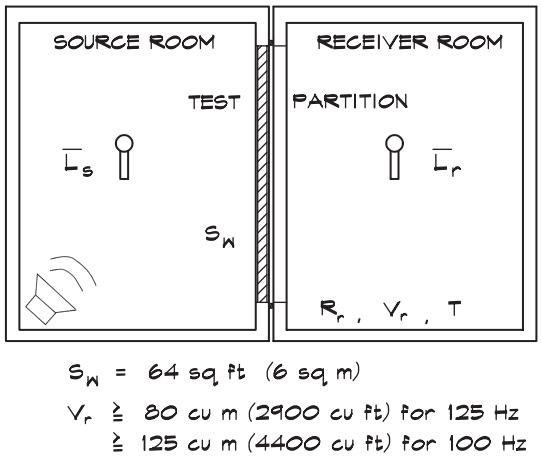
where it generates a sound pressure level. If the receiving room is highly reverberant, the sound field there also will be dominated by the diffuse field component. We use Eq. 8.83 for the reverberant-field contribution to the energy and obtain the mean square pressure in the receiving room

$$\frac{p_r^2}{\rho_0\,c_0} = \frac{p_s^2\,S_w\,\tau}{R_r\,\rho_0\,c_0} \tag{9.3}$$

We can express this as a level by taking 10 log of each side and using the definition of the *transmission loss*

$$\Delta L_{TL} = -10\ \log\ \tau \tag{9.4}$$

FIGURE 9.1 Laboratory Measurements of the Transmission Loss



we obtain the equation for the transmission of sound between two reverberant spaces

$$\overline{L}_r = \overline{L}_s - \Delta L_{TL} + 10 \log \left(\frac{S_w}{R_r} \right) \tag{9.5}$$

where $\overline{L}_r$ = spatial average sound pressure level in the receiver room (dB)
$\overline{L}_s$ = spatial average sound pressure level in the source room (dB)
ΔL_{TL} = reverberant field transmission loss (dB)
S_w = area of the transmitting surface (m^2 or ft^2)
R_r = room constant in the receiving room (m^2 or ft^2 sabins)

Measurement of the Transmission Loss

Under laboratory conditions, both the source and receiving rooms are highly reverberant and the transmission loss of the common partition is given by

$$\Delta L_{TL} = \overline{L}_s - \overline{L}_r + 10 \log S_w - 10 \log R_r \tag{9.6}$$

where the bars over the source and receiver room levels indicate a spatial average in the reverberant-field portion of the rooms. Formal procedures have been established for laboratory (ASTM E90 and ISO 140/III) and field (ASTM E336 and ISO 140/IV) measurements of the transmission loss of partitions, which establish the partition size, the minimum room volume, the method of determining of the room constant, and the appropriate measurement techniques. Loudspeakers are used to generate a sound field in the source room and are positioned in the corners of the room far enough from the transmitting partition that a diffuse field is produced. Sound levels are measured at least a distance $r \geq 0.63 \sqrt{R}$ from the source and 1 m (3 ft) from large reflecting surfaces.

Two methods are used for the determination of the room constant. The reverberation time method measures the value of R_r using the Sabine equation (Eq. 8.72). A second method, called the source substitution method, uses a calibrated noise source having a known sound power level to determine R_r by means of Eq. 8.83. The standard source is usually an unhoused centrifugal fan, which produces a relatively flat noise spectrum.

Transmission loss measurements are done in third-octave bands over a standard range of frequencies, from 125 Hz to 4000 Hz. Below 125 Hz, the size of the test room necessary to achieve the diffuse-field condition becomes large and many labs do not meet this requirement. Data are sometimes taken below the 125 Hz third-octave band for research or other specialized purposes. When measured low-frequency data are unavailable, they can be calculated based on theoretical models. When the room size does not meet the minimum volume requirements for a diffuse field, the noise reduction is used instead of the transmission loss in standard tests and a notation to that effect is included in the test report.

Sound Transmission Class (STC)

Although transmission loss data in third-octave or full-octave bands are used for the calculation of sound transmission between adjacent spaces, it is convenient to have a single-number rating system to characterize the properties of a construction element. The *Sound Transmission Class* is such a system and is calculated in accordance with ASTM E413 and ISO/R 717. It begins with a plot of the third-octave transmission loss data versus frequency. The three-segment STC curve, shown in Fig. 9.2, is compared to the measured data by sliding it vertically until certain criteria are met: 1) no single transmission loss may fall below the curve by more than 8 dB and 2) the sum of all deficiencies (the difference between the curve value and the transmission loss falling below it) may not exceed 32 dB. When the curve is positioned at its highest point consistent with these criteria, as in Fig. 9.3, the STC rating is the transmission loss value at the point where the curve crosses the 500 Hz frequency line. The shape of the STC curve is based on a speech spectrum on the source-room side so this rating system is most useful for evaluating the audibility of conversations, television, and radio receivers. It is less accurate for low-frequency sounds such as music or industrial noise, where energy in the bass frequencies may predominate.

FIGURE 9.2 Reference Contour for Calculating Sound Transmission Class and Other Ratings

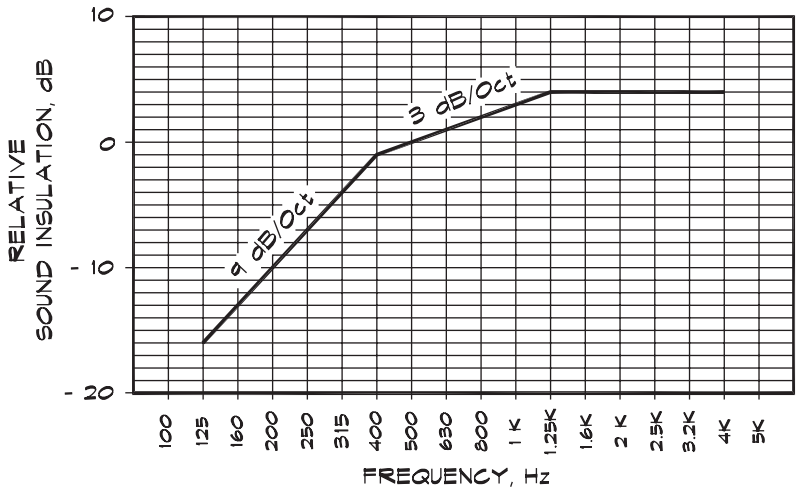
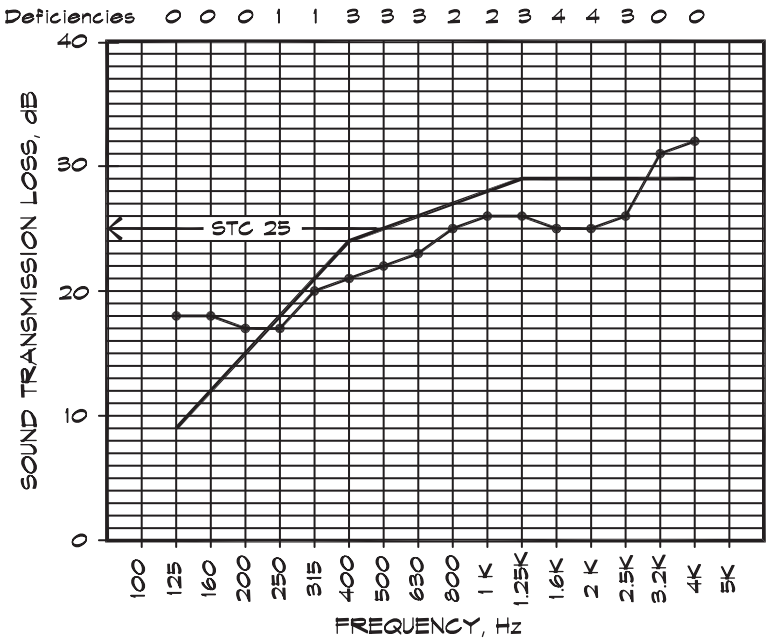


FIGURE 9.3 Example of the Reference Contour Fitted to Transmission Loss Data (STC 25)



Field Sound Transmission Class (FSTC)

Field measurements of the STC can be made in existing buildings and are designated FSTC. Care must be exercised to minimize flanking, where sound is transmitted by paths other than directly through the test partition. The FSTC rating is about five points lower than the STC rating for a given partition, due to flanking and direct-field contributions. It applies only to the partition on which it is measured, but it can serve as an example of the rating of other partitions in a group of similarly constructed structures. It is not generally applicable to a construction type as a laboratory test would be. In building codes, if a given STC rating is required, an FSTC test that is five points lower is sufficient to demonstrate compliance.

Noise Reduction and Noise Isolation Class (NIC)

The arithmetic difference between the sound pressure levels in adjacent spaces is called the *noise reduction*.

$$\Delta L_{NR} = \overline{L}_s - \overline{L}_r \tag{9.7}$$

At frequencies where rooms do not meet the minimum volume requirements necessary to establish the required modal density for a diffuse field, the noise reduction is used instead of the transmission loss to calculate the Sound Transmission Class.

A *Noise Isolation Class* (NIC) can be calculated from noise reduction values by comparing the measured data to the standard reference contour, using the STC calculation criteria (ASTM E413). The NIC is measured in the field, since a laboratory measurement would require knowledge of the transmitting area and the absorption in the receiving room to be useful. A field NIC rating is not applicable to a type of partition since it relates only to the unique combination of partition type, partition area, and the amount of absorption present in

the receiving room at the time of the measurement. Thus, it is not appropriate to assign an NIC rating to a specific construction element or to use it in place of an FSTC value.

9.2 SINGLE PANEL TRANSMISSION LOSS THEORY

Free Single Panels

When a sound wave strikes a freely suspended solid panel, there is a movement imparted, which in turn transmits its motion to the air on the opposite side. Figure 9.4 shows the geometry. The total pressure acting on the panel along its normal is

$$\Delta \mathbf{p} = \mathbf{p}_i + \mathbf{p}_r - \mathbf{p}_t \tag{9.8}$$

The velocity of a normally reacting panel can be calculated using Newton's law

$$\Delta \mathbf{p} = m_s \frac{d\,\mathbf{u}_p}{d\,t} = m_s\,(j\,\omega\,\mathbf{u}_p) \tag{9.9}$$

Note that the normal panel impedance in this model, $j\,\omega\,m_s$, is only due to the panel mass.

If a plane wave approaches at an angle θ to the surface normal and is specularly reflected,

$$\mathbf{u}_i = \frac{j\,k\,\cos\,\theta}{j\,\omega\,\rho_0}\,\mathbf{p}_i = \frac{\cos\theta}{\rho_0\,c_0}\,\mathbf{p}_i \tag{9.10}$$

and

$$\mathbf{u}_r = -\frac{\cos\theta}{\rho_0\,c_0}\,\mathbf{p}_r \quad \text{and} \quad \mathbf{u}_t = \frac{\cos\theta}{\rho_0\,c_0}\,\mathbf{p}_t \tag{9.11}$$

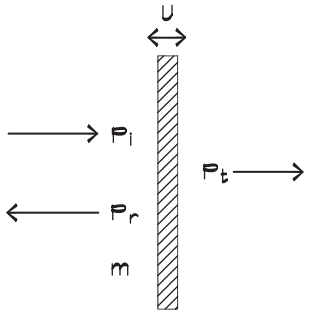
Substituting Eq. 9.10 and 9.11 into 9.9, we obtain the force balance equation along the normal

$$\frac{\rho_0\,c_0\,\mathbf{u}_i}{\cos\theta} - \frac{\rho_0\,c_0\,\mathbf{u}_r}{\cos\theta} = \frac{\rho_0\,c_0\,\mathbf{u}_t}{\cos\theta} + m_s\,(j\,\omega\,\mathbf{u}_t) \tag{9.12}$$

At the surface, the particle velocities on both sides of the plate are the same as the plate velocity, so

$$\mathbf{u}_i + \mathbf{u}_r = \mathbf{u}_t = \mathbf{u}_p \tag{9.13}$$

FIGURE 9.4 Pressure on a Plate Having a Mass, m, Per Unit Area



Substituting these into Eq. 9.9 to eliminate the $\mathbf{u}_r$ term we get the ratio of the transmitted to incident pressures, which is

$$\frac{\mathbf{p_t}}{\mathbf{p_i}} = \frac{1}{1 + \dfrac{j\,\omega\,m_s\,\cos\,\theta}{2\,\rho_0\,c_0}} = \frac{1}{1 + \dfrac{\mathbf{z}_n \cos\theta}{2\rho_0\,c_0}} \tag{9.14}$$

If we use $\zeta_n = \dfrac{\mathbf{z}_n}{\rho_0\,c_0}$ for the normalized impedance, and define the transmissivity as the ratio of the transmitted to the incident power, the square of Eq. 9.14

$$\boldsymbol{\tau}_\theta = \left[\frac{\mathbf{p}_t}{\mathbf{p}_i}\right]^2 = \frac{1}{\left|1 + \dfrac{\zeta_n \cos\,\theta}{2}\right|^2} \tag{9.15}$$

The generalized transmission loss is

$$\Delta\,L_{TL}(\theta) = 10\,\log\left|1 + \frac{\zeta_n \cos\,\theta}{2}\right|^2 \tag{9.16}$$

Mass Law

The limp mass approximation for the normalized panel impedance $\zeta_n = \dfrac{\mathbf{z}_n}{\rho_0 c_0} \cong \dfrac{j\,\omega\,m_s}{\rho_0\,c_0}$ holds for thin walls or heavy membranes in the low-frequency limit, where the panel acts as one mass, moving along its normal, and bending stiffness is not a significant contributor. Using this impedance and recalling that the square of an imaginary number is the sum of the squares of its real and imaginary parts, we can write the *transmissivity* as

$$\tau_\theta = \left[1 + \left(\frac{\omega\,m_s\,\cos\,\theta}{2\,\rho_0\,c_0}\right)^2\right]^{-1} \tag{9.17}$$

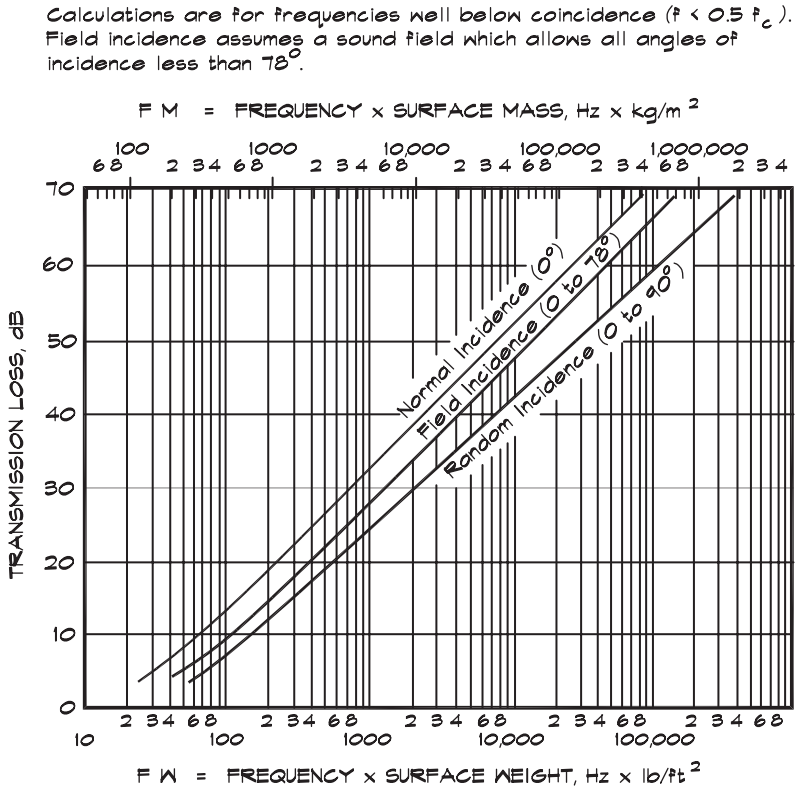
where τ_θ = transmissivity or the fraction of the incident energy transmitted through the panel as a function of incident angle θ
ω = radial frequency (rad / s)
m_s = surface mass density (kg / m^2 or lbs / ft^2)
ρ_0 = density of air (1.18 kg / m^3 or 0.0745 lbs / ft^3)
c_0 = speed of sound in air (344 m / s or 1128 ft / s)

By taking 10 log of Eq. 9.17, we obtain the transmission loss of a panel for a plane wave incident at an angle θ to the normal. A graph of the behavior of the mass law transmission loss as a function of the angle of incidence of the sound wave is given in Fig. 9.5.

$$\Delta L_{TL}(\theta) = 10\,\log\left[1 + \left(\frac{\omega\,m_s\,\cos\,\theta}{2\,\rho_0 c_0}\right)^2\right] \tag{9.18}$$

For most architectural materials, the normalized mass impedance is much greater than one so that the term on the right-hand side of the bracket dominates.

FIGURE 9.5 Theoretical Sound Transmission Loss of Panels (Beranek, 1971)



When the incident sound field is diffuse, there is an equal probability that sound will come from any direction. The diffuse incidence transmissivity is obtained by integrating Eq. 9.17 over all angles of incidence up to a limiting value of θ_{Max}.

$$\tau = \frac{\int_0^{\theta_{\text{Max}}} \tau_\theta \cos\theta \sin\theta \, \mathrm{d}\theta}{\int_0^{\theta_{\text{Max}}} \cos\theta \sin\theta \, \mathrm{d}\theta} \tag{9.19}$$

The reason for this approach is that the elementary theory of Eq. 9.18 predicts a transmission loss of zero for grazing incidence, so if the limiting angle is 90°, we obtain a result that does not occur in practice. It has become standard procedure to select a maximum angle that yields the best fit to the measured data. This turns out to be about 78°, and gives what is known as the field-incidence transmission loss

$$\Delta L_{TL} = 10 \log \left[1 + \left(\frac{\omega m_s}{3.6 \rho_0 c_0} \right)^2 \right] \tag{9.20}$$

or

$$\Delta L_{TL} = 20 \log \left(f m_s \right) - K_{TL} \tag{9.21}$$

where ΔL_{TL} = diffuse field transmission loss (dB)
f = frequency (Hz)
m_s = surface mass density of the panel material (kg/m^2 or lbs/ft^2)
K_{TL} = numerical constant
= 47.3 dB in metric units and 33.5 dB in FP units

Equation 9.21 is known as the mass law since the transmission loss at a given frequency is only dependent on the surface mass of the panel. The transmission loss increases six dB for each doubling of surface mass or frequency. Under field-incidence conditions, the integration over the angle of incidence results in an effective mass that is lower by a factor 1.8 than the actual mass. Comparing Eq. 9.18 with Eq. 9.20, we see that this is the same as a difference between the field and normal incidence transmission losses of 5 dB.

$$\Delta L_{TL} \cong \Delta L_{TL}\,(\theta = 0) - 5 \tag{9.22}$$

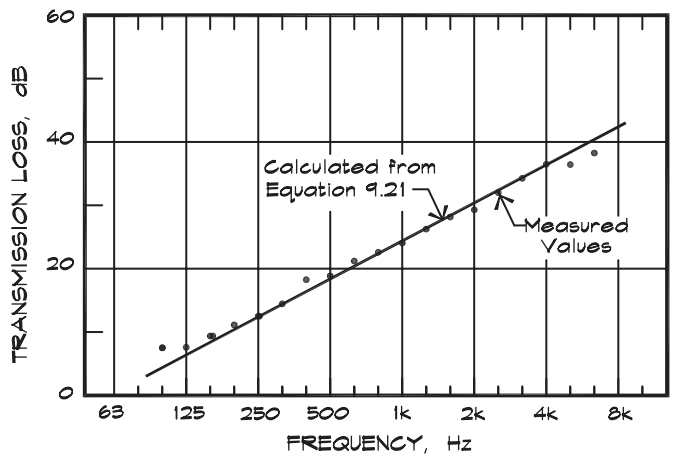
Figure 9.6 shows a graph of the diffuse field transmission loss measured for a 3-mm (1/8") hardboard panel compared with the calculated value. In this frequency range, the agreement with measured data for thin panels is quite good.

Large Panels—Bending and Shear

Panels that are large compared to a wavelength, react to an applied pressure not only as a limp mass but also as a plate, which can bend or shear. As such, they have an impedance that is more complicated than that previously assumed. When the possibility of bending or shear is present, the two transmission mechanisms act much like resistors in an electric circuit. The composite panel impedance is treated mathematically like two resistors in parallel, which are in series with the mass impedance

$$\mathbf{z} \cong j\,\omega\,m_s + \frac{\mathbf{z}_B\,\mathbf{z}_s}{\mathbf{z}_B + \mathbf{z}_s} \tag{9.23}$$

FIGURE 9.6 Transmission Loss of 3 mm (1/8") Hardboard (Sharp, 1973)



For an isotropic plate, the bending impedance (Sharp, 1973; Cremer, Heckel, and Ungar, 1973; or Fahy, 1985) is given by

$$\mathbf{z}_B \cong -\frac{j\,\omega^3\,B}{c_0^4}\,\sin^4\,\theta \tag{9.24}$$

and the shear impedance (Mindlin, 1951; Cremer, Heckel, and Ungar, 1973; or Beranek and Ver, 1992) is

$$\mathbf{z}_s = -j\,\frac{G\,h\,\omega\sin^2\,\theta}{c_0^2} \tag{9.25}$$

where ω = radial frequency = $2\,\pi\,f$ (rad/s)
m_s = surface mass density of the panel (kg / m^2 or lbs / ft^2)
$j = \sqrt{-1}$
E = Young's modulus of elasticity (N / m^2 or lb / ft^2)

$$B = \frac{E\,h^3}{12\,(1-\sigma^2)} = \text{bending stiffness (N m or ft lbs)}$$

$$G = \frac{E}{2\,(1+\sigma)} = \text{shear modulus (N / m}^2 \text{ or lbs / ft}^2\text{)}$$

c_0 = speed of sound in air (m / s or ft / s)
σ = Poisson's ratio
h = panel thickness (m or ft)

Thin Panels—Bending Waves and the Coincidence Effect

In Eq. 9.23, the impedance is composed of three imaginary terms: the inertial mass, the bending stiffness, and the shear impedance. At low frequencies, the mass term predominates; at high frequencies it is the combination of bending and shear that determines the composite impedance. Thin panels are easier to bend than to shear, so more of the energy will flow into this mode. The resulting plate impedance becomes

$$\mathbf{z} \cong j\,\omega\,m_s - \frac{j\,\omega^3\,B}{c_0^4}\,\sin^4\,\theta \tag{9.26}$$

At one frequency, called the *coincidence frequency*, the mass and bending impedance terms are equal and, since they have opposite signs, the composite impedance is zero. Figure 9.7 illustrates the crossover point for this condition.

Coincidence can be understood by realizing that the velocity of bending waves in a panel is a function of frequency. At the coincidence frequency, the bending wave velocity is the same as the trace velocity of the airborne sound moving along the panel. Since the pressure maxima and minima are spatially matched, as in Fig. 9.8, energy is easily transmitted from the air into the panel and vice versa. The frequency at which coincidence occurs varies with the angle of incidence and is obtained by setting Eq. 9.26 to zero

$$f_{co}\,(\theta) = \frac{c_0^2}{2\,\pi\,\sin^2\,\theta}\,\sqrt{\frac{m_s}{B}} \tag{9.27}$$

FIGURE 9.7 Imaginary Part of the Thin Panel Impedance (Sharp, 1973)

The imaginary part of the transmission impedance of a thin panel for grazing incidence ($\theta = \pi / 2$) showing the effect of coincidence.

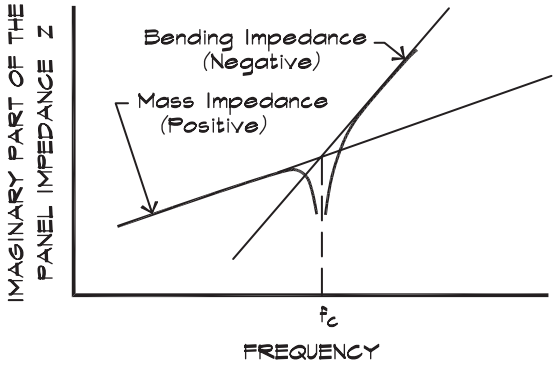
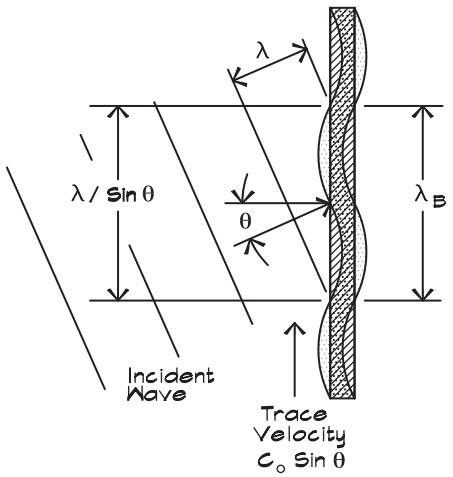


FIGURE 9.8 Coincidence Effect

The coincidence effect matches the trace of the airborne wavelength and the bending wavelength in the panel.



For normal incidence, the coincidence frequency is infinite, so there is no effect. The minimum value of the matching frequency occurs at grazing incidence and is called the *critical frequency*.

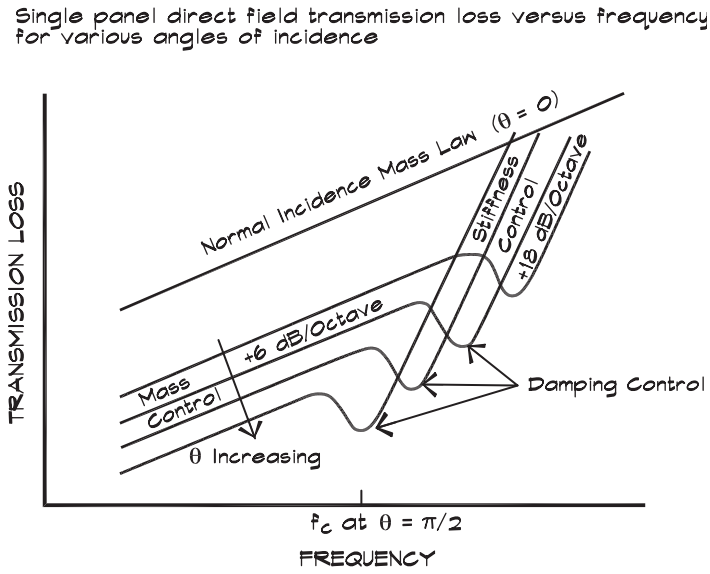
$$f_c = \frac{c_0^2}{2\pi}\sqrt{\frac{m_s}{B}} = \frac{c_0^2}{2\pi h}\sqrt{\frac{12\,(1-\sigma^2)\,\rho_m}{E}} \tag{9.28}$$

where ρ_m = bulk density of the panel material (kg/m^3 or lbs/ft^3).

The two frequencies are related by means of

$$f_{co}\,(\theta) = \frac{f_c}{\sin^2\theta} \tag{9.29}$$

FIGURE 9.9 Single Panel Direct Field Transmission Loss (Fahy, 1985)



which implies that the *coincidence effect* always occurs at or above the critical frequency. Figure 9.9 gives examples in which the position of the coincidence dip varies with the angle of incidence.

Above coincidence the bending impedance term increasingly dominates and the transmission loss becomes (Fahy, 1985)

$$\Delta L_{TL}(\theta) \cong 10 \log \left[1 + \left(\frac{B k^4 \sin^4 \theta \cos \theta}{2 \rho_0 c_0 \omega} \right)^2 \right] \tag{9.30}$$

The bending transmission loss is stiffness controlled in this region and has an 18 dB per octave slope as well as a strong angular dependence.

At the critical frequency, the transmission loss does not fall to zero because internal damping prevents it. To treat this theoretically, a complex bending stiffness $\mathbf{B} = B(1 + j\eta)$ is introduced with a damping term η having a value less than one. Since the mass term and the bending term cancel out at coincidence, damping is left. Using Eq. 9.27, Eq. 9.30 can be written as an inertial term with a damping coefficient (Fahy, 1985)

$$\Delta L_{TL}(\theta) \cong 10 \log \left(1 + \frac{\eta \omega_{co} m_s \cos \theta}{2 \rho_0 c_0} \right)^2 \tag{9.31}$$

At coincidence, the transmission loss is expressed as the direct-field mass law plus a damping term $20 \log \eta$. Figure 9.9 also shows where internal damping becomes important.

The diffuse-field transmission loss is normally calculated by integrating the direct-field transmission loss expression over all relevant angles of incidence. The integration in the coincidence region is difficult because different equations apply, depending on the angle and the frequency. Although the coincidence frequency varies with angle of incidence,

the diffuse-field transmission loss still has a minimum at the critical frequency, since the dominant path is that having the lowest loss.

Several authors have developed approximate relations to use in this region. Fahy (1985) gives an equation for the transmission loss at the diffuse-field coincidence point, which is written in terms of a combination of the normal-incidence mass law and a damping term, dependent on the bandwidth

$$\Delta L_{TL}\ (f = f_c) \cong 20\ \log\left(\frac{\omega_c\, m_s}{2\,\rho_0\, c_0}\right) + 10\ \log\left\{\frac{2\,\eta}{\pi}\left(\frac{\Delta f}{f_c}\right)\right\} \tag{9.32}$$

where Δf = bandwidth (Hz) - - typically one - third octave or one octave wide
f_c = critical frequency (Hz)
η = damping coefficient ($\eta < 1$)

When the transmission loss is calculated just below coincidence, a line is drawn between the field-incidence mass law value at $f_c/2$ and the critical-frequency transmission loss from Eq. 9.32. Above the coincidence frequency, Eq. 9.30 predicts an 18 dB per octave increase in transmission loss for a given angle of incidence. Measured diffuse-field data yield a slope that is about half that (Sharp, 1973), due to the shifting location of the coincidence dip with angle. Cremer (1942) has derived an approximate diffuse-field equation for use above the coincidence frequency that combines the normal-incidence mass law and a frequency-dependent damping term.

$$\Delta L_{TL}\ (f > f_c) \cong 20\ \log\left(\frac{\omega\, m_s}{2\,\rho_0\, c_0}\right) + 10\ \log\left\{\frac{2\eta}{\pi}\left(\frac{f}{f_c} - 1\right)\right\} \tag{9.33}$$

Above coincidence, Eq. 9.33 yields a transmission loss that increases 9 dB per octave for a single panel, which agrees with measured values.

Table 9.1 and Fig. 9.10 give a compilation of material properties: the product of the critical frequency and the panel thickness along with the damping coefficients for use in these equations. Figure 9.11 plots the measured transmission loss for a sheet of 5/8" drywall, which exhibits a coincidence dip around 2500 Hz. Data calculated from Eqs. 9.32 and 9.33 are also shown.

Thick Panels

As panel thickness increases, the composite panel impedance utilized in thin panel theory is no longer accurate. At high frequencies a shear wave can develop and propagate in a thick panel. When shear presents a lower impedance than bending, it will become the predominant transmission mode. The composite impedance is given by Eq. 9.25.

The crossover point between bending and shear occurs where the thickness of the plate is equal to a bending wavelength in the plate material. When the panel is thicker than a wavelength, shear predominates, as illustrated in Fig. 9.12. The shear limiting frequency is (Sharp, 1973)

$$f_s = \frac{c_0^2\,(1 - \sigma)}{59\,h^2\,f_c} \tag{9.34}$$

TABLE 9.1 Product of Plate Thickness and Critical Frequency in Air (20° C) (Beranek, 1971; Cremer et al., 1973; Fahy, 1985)

Material	f_c (m sec^{-1})	h f_c (ft sec^{-1})
Steel	**12.4**	**41**
Aluminum	**12.0**	**39**
Brass	**17.8**	**58**
Copper	**16.3**	**54**
Glass or Sand	**12.7**	**42**
Plaster		
Gypsum Board	**38**	**125**
Chipboard	**23**	**75**
Plywood or Brick	**20**	**66**
Asbestos Cement	**17**	**56**
Concrete		
Dense	**19**	**62**
Porous	**33**	**108**
Light	**34**	**112**
Lead	**55**	**180**

Note that variations of 10% are not uncommon.

FIGURE 9.10 Values of Material Loss Factors (Beranek and Ver 1992)

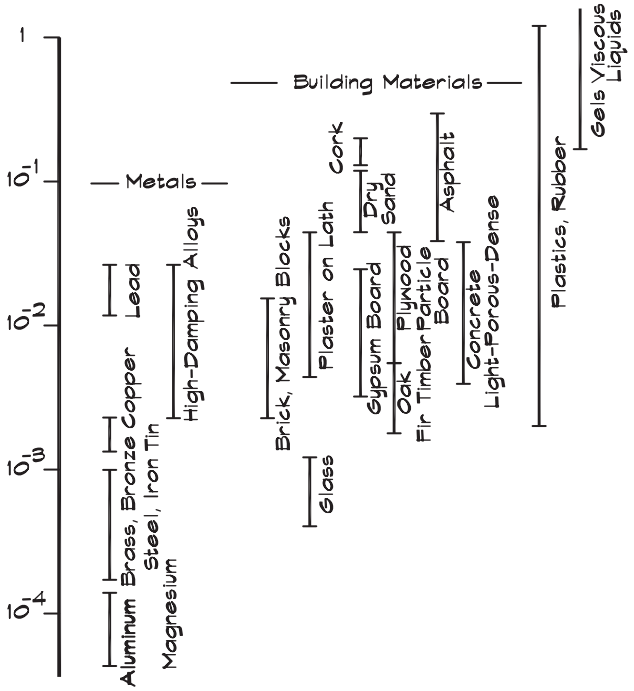


FIGURE 9.11 Transmission Loss of 16 mm (5/8") Gypboard (Sharp, 1973)

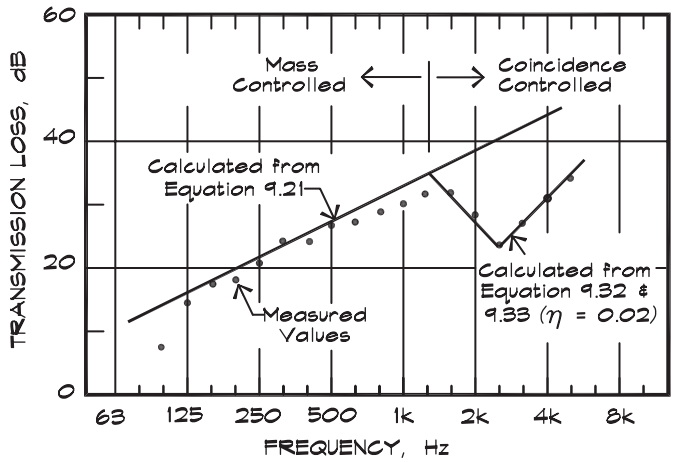
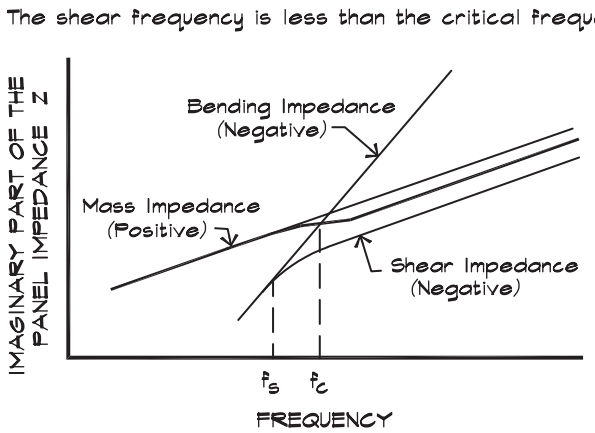


FIGURE 9.12 Transmission Impedance of Thick Panels (Sharp, 1973)



When the shear frequency falls below the critical frequency, as it can with thick panels such as concrete slabs and brick or masonry walls, there is no coincidence dip and the shear mechanism lowers the transmission loss even below that which might be expected from purely mass law considerations. This can be very important since in Fig. 9.13, it occurs in the frequency range around 200 Hz for a 15 cm (6") concrete slab. A good estimate of the transmission loss can be made above this point by using 6 dB less than the diffuse field mass law. For concrete or solid brick structures this is equivalent to assuming that there is half the actual mass.

If the shear frequency is greater than the coincidence frequency, the shear wave impedance eventually becomes lower than the bending impedance. All materials appear thick at a high enough frequency. The shear-wave impedance limits the slope of the transmission loss line above the shear-bending frequency to 6 dB per octave. Figure 9.14 gives

FIGURE 9.13 Transmission Loss of a 6" Concrete Panel (Sharp, 1973)

The measured values of transmission loss for a 6-inch (150 mm) concrete panel compared to values predicted by means of the thin and thick panel theories.

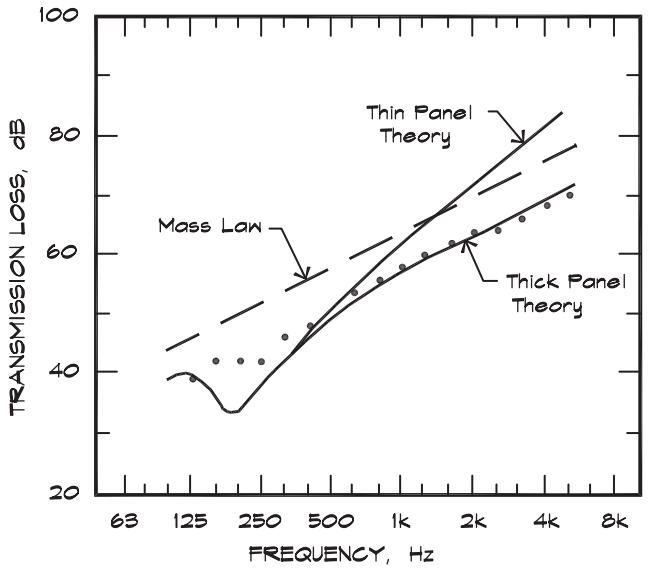
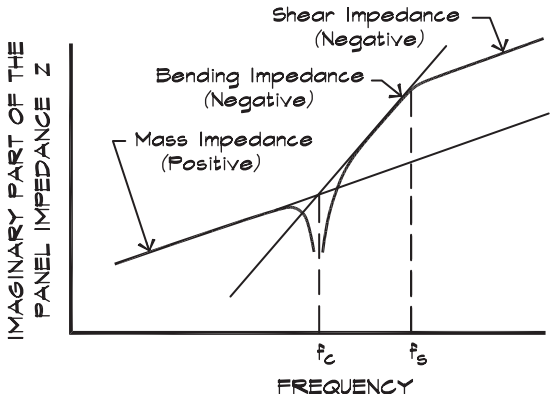


FIGURE 9.14 Imaginary Part of the Transmission Impedance (Sharp, 1973)

The imaginary part of the transmission impedance of a thin panel for grazing incidence ($\theta = \pi/2$) showing the effect of coincidence. The shear frequency is greater than the critical frequency.



an example. Since coincidence in lightweight construction materials such as sheet metal, gypsum board, or thin wood panels occurs at a relatively high frequency (e.g., above 2000 Hz), there is little practical impact from shearing effects since there is little sound transmission in this region.

Finite Panels—Resonance and Stiffness Considerations

For a finite-sized panel, an additional term must be added to the impedance at very low frequencies, to account for panel bending resonances. This term is given by (Leissa, 1969; Sharp, 1973)

$$\mathbf{z}_P = -j\,\frac{K_P}{\omega} \tag{9.35}$$

where

$$K_P = \pi^4 B \left[\frac{1}{a^2} + \frac{1}{b^2}\right]^2 \tag{9.36}$$

and a and b are the dimensions of the panel. In this frequency range, the overall impedance is the sum of the panel impedance and the mass impedance. When the two terms are equal, a panel resonant frequency is produced

$$f_P = \frac{\pi}{2}\sqrt{\frac{B}{m_s}}\left[\frac{1}{a^2} + \frac{1}{b^2}\right] \tag{9.37}$$

Typically, the dimensions of a panel such as a wall are large enough that the panel resonance is quite low, on the order of 10 Hz or less. A 2 m $\times$ 3 m sheet of gypboard, for example, has a panel resonance of about 5 Hz. Note that Eq. 9.37 is based on the fundamental resonant frequency of the panel. There are also higher panel modes; however, due to damping these rarely contribute to the transmission loss. The frequency appears in the denominator in Eq. 9.35, so below the panel resonance the transmission loss increases with decreasing frequency at 6 dB per octave. Sharp (1973) has given an approximate relationship for this region.

$$\Delta L_{TL} \cong 20\,\log(f\,m_s) - K_{TL} + 40\,\log\left(\frac{f_P}{f}\right) \tag{9.38}$$

for $f < f_P$ the lowest panel resonance frequency.

Design of Single Panels

For single panels the transmission loss is influenced by four factors: 1) size, 2) stiffness, 3) mass, and 4) damping. Josse and Lamure (1964) have developed a comprehensive, albeit somewhat more complex, formulation for the region below the critical frequency

that includes all four components

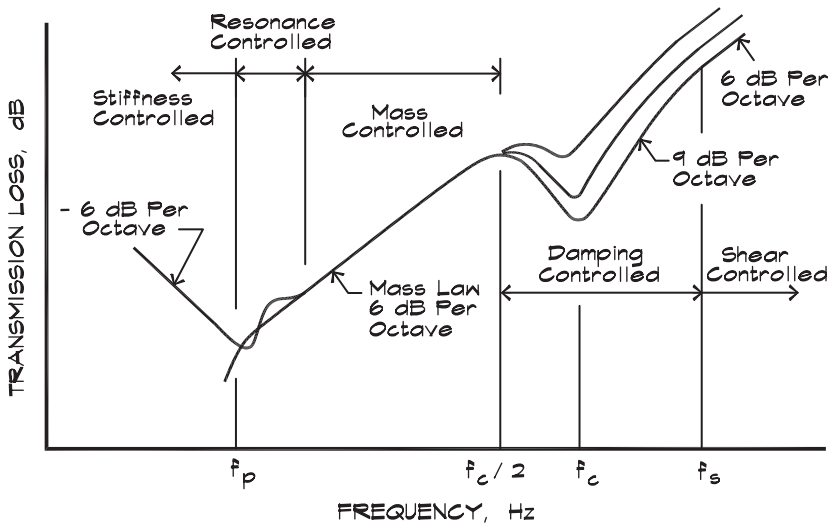
$$\Delta L_{TL} = 20 \log\left(\frac{\omega m_s}{2\rho_0 c_0}\right) - 10 \log\left\{\left[\frac{3}{2} + \ln\left(\frac{2\omega}{\Delta\omega}\right)\right] + \frac{16 c_0^2}{\eta \omega_c} \frac{1}{(\omega \omega_c)^{1/2}} \left(\frac{a^2 + b^2}{a^2 b^2}\right)\left[1 + \frac{2\omega}{\omega_c} + 3\left(\frac{\omega}{\omega_c}\right)^2\right]\right\} \tag{9.39}$$

For typical panel sizes, in the mass-law region Eq. 9.39 gives about the same result as Eq. 9.21. Above the critical frequency, Eq. 9.32 or Eq. 9.33 can be used to predict the transmission loss. For thick panels, the mass law holds up to the point where shear predominates. Above this point, mass law less 6 dB gives a good estimate.

The transmission loss of a thin panel falls into five frequency regions illustrated in Fig. 9.15. At very low frequencies the transmission loss is stiffness controlled. The greater the bending stiffness and the shorter the span, the higher the transmission loss. These considerations become important in low-frequency sound transmission problems in long-span floor structures, particularly in lightweight wood construction, where the bending stiffness of the structure is not great. Although floors are a composite structure, they can be thought of as a single bending element at these low frequencies.

Above the fundamental panel mode, the transmission loss of a thin single panel is primarily a function of its mass. Techniques such as mass loading can be used to increase the intrinsic panel mass, by adding asphalt roofing paper, built-up roofing, sand, gravel, lightweight concrete, gypboard, or discrete masses. Lead-loaded gypsum board panels are available for critical applications. Mass loading increases the weight of the panel without increasing the panel stiffness. Loading materials need not cover the entire panel but can be located at regular intervals over the surface. In existing wooden floor structures mass can be added by screwing sheets of gypboard to the underside of wood subfloors between the

FIGURE 9.15 General Form of the Diffuse Field Transmission Loss vs Frequency Curve for a Thin Panel



joists, a technique that also helps close off any cracks in the subflooring. Although a high mass is desirable, a high thickness is not, due to its effect on the coincidence frequency. For example, 9 mm (3/8") glass has a higher STC rating (STC 34) than a normal 44 mm (1 3/4") solid core door (STC 30) even though it weighs less per unit area, due to the fact that its coincidence frequency is much higher.

In the region around coincidence, damping controls the depth of the notch. Several techniques can be used to increase the transmission loss. In windows and glazed doors, a resin interlayer of 0.7 to 1.4 mm (30 to 60 mils) thickness can be introduced between two layers of glass to increase damping. Products of this type are called laminated glass or sometimes acoustical glass. Sheets of 6 mm (1/4 in) laminated glass make good acoustical windows and can achieve an STC rating of 34 in fixed stops or high-quality frames.

Damping compounds are also commercially available. Products come in sheet or paste form and add both mass and damping. They are primarily used to treat thin materials such as sheet-metal panels on vibrating equipment. Asphalt materials such as undercoat on car bodies serve the same function and reduce the vibration amplitudes of these surfaces.

Spot Laminating

Spot laminating is a technique used to increase transmission loss near coincidence. In this region, it is important not to thicken the panel, which decreases the critical frequency. If the mass is increased through the addition of an extra layer, it may decrease the transmission loss in the frequency range of interest. One way to achieve a combination of high mass and low stiffness is to laminate panels together using dabs (spots) of panel adhesive at regular intervals. At low frequencies, the panels act as one and their composite bending stiffness is greatly increased. At high frequencies the shearing effect of the adhesive acts to reduce the bending stiffness to that of an individual panel. The panels act individually and the critical frequency remains high, since it is that of a single panel in bending.

When using the spot laminating technique the spacing of the adhesive dots determines the frequency at which the two panels begin to decouple. Decoupling begins when the dot spacing is equal to the wavelength of bending waves. The bending wavelength of a single panel at a frequency f is given by (Sharp, 1973) as

$$\lambda_B = \frac{c_0}{\sqrt{f\, f_c}} \tag{9.40}$$

At low frequencies, the critical frequency is that of the composite panel since the bending wavelength is much greater than the spacing of the adhesive. If two panels of identical materials are joined together, the coincidence frequency of each sheet is about twice that of the composite. The decoupling frequency can be written as

$$f_d = \frac{2\, c_0^2}{a^2\, f_c} \tag{9.41}$$

where a is the adhesive spacing and f_c is the critical frequency of a single sheet of material. If the two panels are 1/2" gypboard, which has a coincidence frequency of about 3000 Hz, and the adhesive dot spacing is 24", then the decoupling frequency is 210 Hz. This is well below the composite panel's coincidence frequency, which would be about 1500 Hz.

9.3 DOUBLE-PANEL TRANSMISSION LOSS THEORY

Free Double Panels

The transmission loss of a double panel system is based on a theory that first considers panels of infinite extent, not structurally connected, and subsequently addresses the effects of various types of attachments. The treatment here follows that first developed by London (1950) and later by Sharp (1973). It is assumed that the air cavity is filled with batt insulation, which damps the wave motion parallel to the wall, so we are mainly concerned with internal wave motion normal to the surface. The airspace separating the panels acts as a spring and at a given frequency, a mass-air-mass resonance occurs. The resonance is the same as that given in Eq. 7.103, which was developed in the discussion of air-backed panel absorbers, but the mass has been modified to account for the diffuse field and for the fact that both panels can move

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{3.6\,\rho_0\,c_0^2}{m'\,d}} \tag{9.42}$$

where $m' = \dfrac{2\,m_1\,m_2}{m_1 + m_2}$ = effective mass per unit area of the construction (kg/m^2 or lbs/ft^2)

d = panel spacing (m or ft)

For a double panel wall, consisting of 16 mm (5/8") gypboard separated by a stud space of 9 cm (3.5"), the mass-air-mass resonance occurs at about 66 Hz. At low frequencies, below the resonant frequency, the two panels act as one mass. If the individual panels are mass controlled then the transmission loss follows the mass law of the composite structure.

At frequencies above the mass-air-mass resonance, the effect of the air cavity is to increase the transmission loss significantly. In fact for N separate panels, the ideal theoretical transmission loss increases 6(2N-1) dB per octave. At high frequencies, constructions having multiple panels with intervening air spaces can provide significant increases in transmission loss over that achieved by a single panel. Since it is rarely practical to construct walls or floors with more than about three panels, consideration of large numbers of panels is generally not useful.

A plane wave encountering a double-panel system sees the impedance of the nearest panel, the impedance of the airspace, the impedance of the second panel, and finally the impedance of the air beyond. The transmissivity has been given by London (1950) as

$$\tau_\theta = \left[1 + (X_1 + X_2) + X_1 X_2 \left(1 - e^{-j\sigma}\right)\right]^{-2} \tag{9.43}$$

where $X = \dfrac{z_n \cos\theta}{2\,\rho_0\,c_0}$

z_n = normal impedance of a panel (rayls)

$\sigma = 2\,k\,d\cos\theta$

$k = \dfrac{2\pi f}{c_0}$ = wave number (m^{-1})

Below the critical frequency the panel impedance is equal to its mass reactance and, for a diffuse field, the transmission loss of an ideal double panel system of infinite extent, attached

only through the air spring coupling, is

$$\Delta L_{TL} \cong 10 \log \left\{ 1 + \left[\frac{\omega M}{3.6 \rho_0 c_0} - \frac{\omega^2 m_1 m_2}{(3.6 \rho_0 c_0)^2} \left(1 - e^{-2jkd} \right) \right]^2 \right\} \tag{9.44}$$

where m = mass per unit area of an individual panel (kg/m^2)
M = total mass per unit area of the construction (kg/m^2)

At low frequencies, below the mass-air-mass resonance, the panel spacing is small compared with a wavelength so the rightmost term in Eq. 9.44 approaches zero and the transmission loss can be approximated by

$$\Delta L_{TL} \cong 10 \log \left\{ 1 + \left[\frac{\omega M}{3.6 \rho_0 c_0} \right]^2 \right\} \tag{9.45}$$

which is just the mass law for the composite panel.

$$\Delta L_{TL} \cong 20 \log (f M) - K_{TL} \qquad f < f_0 \tag{9.46}$$

At frequencies above f_0, but still below the point where the wavelength is comparable to the panel separation, the rightmost term in Eq. 9.44 begins to dominate. In this region the wavelength is still larger than the panel spacing, so (2 k d) is small and $e^{-2jkd} \cong 1 - 2jkd$. The transmission loss can be approximated by

$$\Delta L_{TL} \cong 20 \log \left[\frac{\omega^2 m_1 m_2}{(3.6 \rho_0 c_0)^2} 2kd \right] \tag{9.47}$$

which can be written as

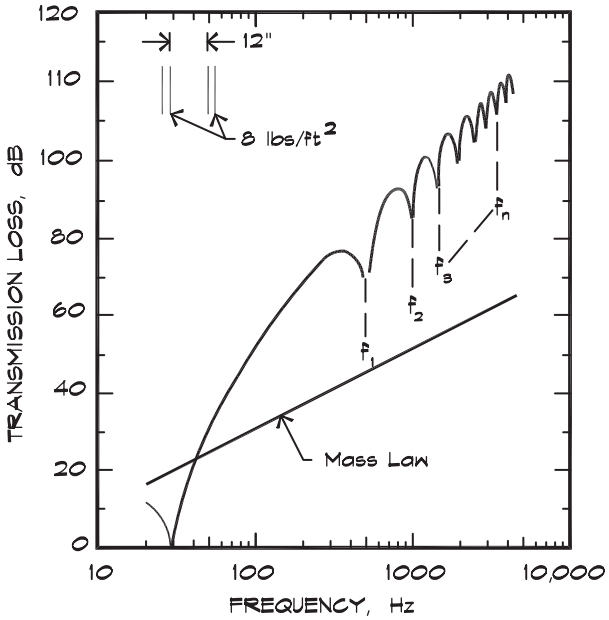
$$\Delta L_{TL} \cong \Delta L_{TL1} + \Delta L_{TL2} + 20 \log (2kd) \qquad f_0 > f > f_\ell \tag{9.48}$$

where ΔL_{TL1} and ΔL_{TL2} are the mass law transmission losses for each panel. Each term in Eq. 9.48 increases 6 dB per octave so the overall transmission loss rises 18 dB per octave in this frequency range. It also increases 6 dB for each doubling of the separation distance between the panels. At still higher frequencies resonant modes can be sustained in the airspace between the panels and there arises a series of closed tube resonances, normal to the surfaces, having frequencies

$$f_n = \frac{n c_0}{2d} \quad \text{for } n = 1, 2, 3 \ldots \tag{9.49}$$

These aid in the transfer of sound energy between the panels and result in a series of dips in the double-panel transmission loss, shown in Fig. 9.16. Due to the damping provided by batt insulation, we usually do not see this pattern in actual transmission loss measurements, which are carried out in third-octave bands. Batt flattens out the slope of the transmission

FIGURE 9.16 Theoretical Transmission Loss of an Ideal Double Panel (Sharp, 1973)



loss curve from 18 dB/octave down to 12 dB/octave in this region. The transmission loss behavior above a limiting frequency is given by

$$\Delta L_{TL} \cong \Delta L_{TL1} + \Delta L_{TL2} + 6 \qquad f_\ell < f \tag{9.50}$$

The crossover frequency, which can be obtained by setting Eq. 9.48 equal to Eq. 9.50, is equal to the first cavity resonance frequency divided by π.

$$f_l = \frac{c_0}{2\pi d} = \frac{f_1}{\pi} \tag{9.51}$$

Equations 9.46, 9.48, and 9.50 give the transmission loss of an ideal double panel system in three frequency ranges separated by the frequencies in Eqs. 9.42 and 9.51. Measurements by Sharp (1973) have shown that there is good agreement between theoretical and measured data for both the mass-reactance region (Fig. 9.17) and the critical region (Fig. 9.18). The panel transmission loss values in the critical region were used instead of simple mass law predictions.

Cavity Insulation

The theory that has been developed here has assumed that the air cavity is well damped. When there is no absorptive material between the panels, cavity resonances contribute to the transmission of sound from one side to the other in much the same way as a mechanical coupling would. The addition of damping material such as fiberglass batt insulation attenuates these modes. Figure 9.19 (Sharp, 1973) shows the effects of a fully isolated double-panel system, with and without batt insulation. The panels used were 1/4" and 1/8" hardboard so that the coincidence frequencies were above the frequency range of interest. Below the first

FIGURE 9.17 Measured and Calculated Transmission Loss Values for a Separated Double Panel (Sharp, 1973)

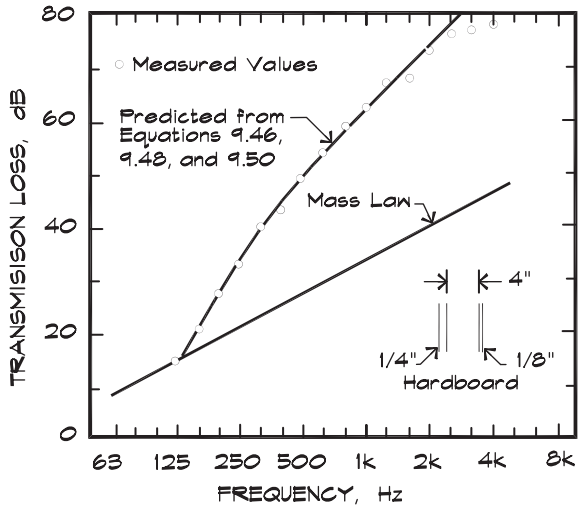


FIGURE 9.18 Transmission Loss Values for a Separated Double Panel Including Coincidence (Sharp, 1973)

Measured and calculated values of the transmission loss of 5/8-inch gypsum board wall including the coincidence region.

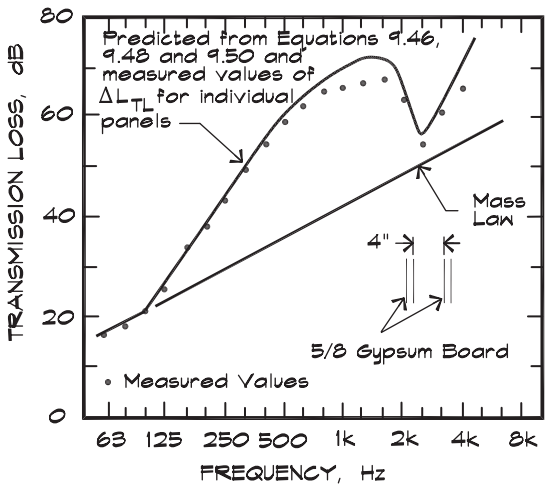
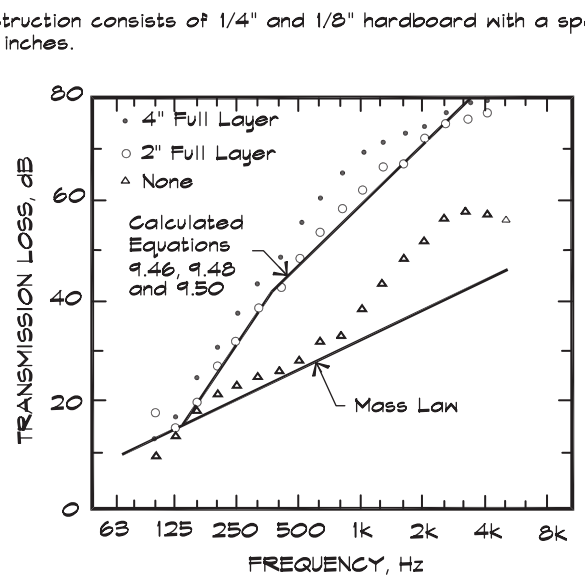


FIGURE 9.19 Transmission Loss of an Isolated Double Panel Construction with and without Insulation (Sharp, 1973)



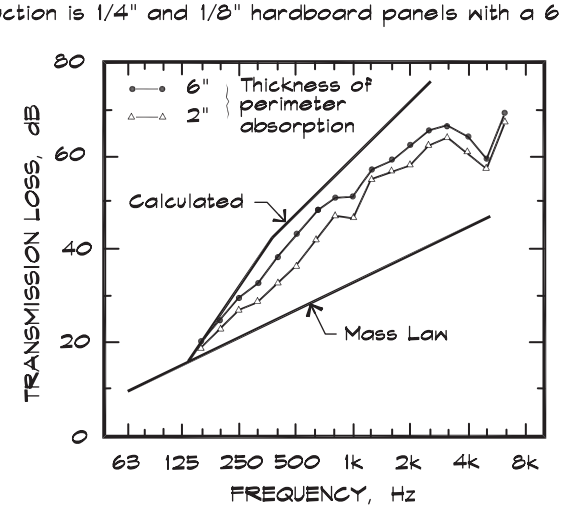
cavity resonance (1100 Hz), the panels are so coupled by the cavity resonances that the transmission loss follows the mass law. Above the first resonance, the phase varies over the depth of the cavity and the coupling is weaker. When 2" insulation is introduced into the cavity the wall begins to act like a double-panel system. With 4" of insulation, the mass of the insulation is comparable to that of the panel and transmission losses are greater than those predicted using simple theory.

Table 9.2 shows measured increases in STC values for various thicknesses of insulation between separately supported 16-mm (5/8") gypboard wall panels. The effectiveness of batt insulation in a wall cavity is almost entirely dependent on its thickness. The density of the fill for a given thickness makes only a small difference at relatively high frequencies, above 1000 Hz. The presence of paper backing on the fiberglass makes no difference for gypboard partitions, nor does the position of the partial thickness within the cavity, including contact with either side. Weaving a blanket of insulation around staggered studs is no more effective than the simpler method of placing the batt vertically between the studs. The

TABLE 9.2 Measured Effects of Fiberglass Batt in Completely Separated Double Panel Walls (Owens Corning Fiberglass, 1970)

Thickness of Cavity Fill	Average Increase in STC (dB)
1 1/2"	10
2"	11
3"	12
4"	13
6"	15
8"	16

FIGURE 9.20 Transmission Loss of an Isolated Double Panel Construction with Perimeter Insulation (Sharp, 1973)



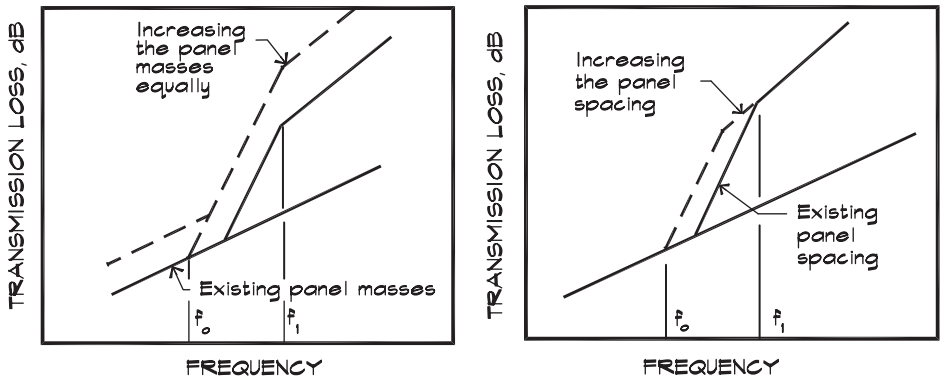
effectiveness of insulation in staggered stud and single stud walls is much less than in ideal walls due to the high degree of mechanical coupling between the two sides. Improvements in real (with studs, etc.) walls of from 3 to 7 dB were obtained, with insulation thicknesses of 2 to 6 inches. For most walls and floor-ceilings, where the two sides are mechanically coupled together in some way, the addition of batt insulation increases the STC from 3 to 5 dB for the first three inches of material and about a dB per inch above that point. Thicknesses of batt insulation beyond the stud spacing are not helpful since the compressed batt can form a structural bridge between the panels.

Higher order cavity modes, which are not necessarily perpendicular to the panels, can contribute to the sound transmission in the mid and high frequencies. Therefore, absorptive material is quite effective even if it is placed only around the perimeter of the cavity. This is a common technique used in the construction of double-glazed sound-rated windows for studios. Figure 9.20 illustrates the effect of insulation placed only around the perimeter of the cavity. At low frequencies, the transmission loss increases with the thickness of the insulation. In studio window construction it is helpful to leave an opening into the wall cavity, which acts as a Helmholtz resonator absorber, tuned to the mass-air-mass resonance frequency.

Double-Panel Design Techniques

At frequencies below the mass-air-mass (m-a-m) resonance, a double-panel wall acts as a single unit and the stiffness and mass are the most important contributors to transmission loss. Above the m-a-m resonance, both mass and panel spacing are important as shown in Fig. 9.21. Equation 9.47 holds that, for a given total mass, the transmission loss is greatest if the mass is distributed equally on both sides of the wall. This is generally true although it is a commonly held belief that unbalanced construction is preferred because of the desirability of mismatching the critical frequencies of the panels. Recent studies (Uris et al., 2000) have

FIGURE 9.21 Effect of Mass and Spacing on Transmission Loss Ideal Double Panel Construction (Sharp, 1973)



shown that, for a given mass, there is little difference between balanced and unbalanced gypsum board wall constructions, even at coincidence. The coincidence effect can be controlled by adding damping or by using different thicknesses of gypboard, while maintaining an overall equality of mass on each side (Green and Sherry, 1982).

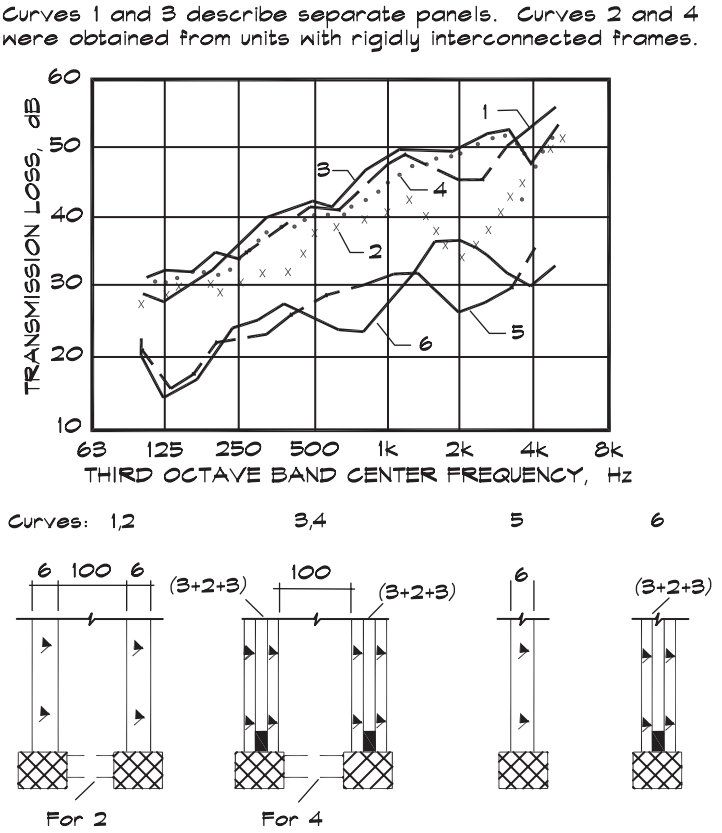
Because of the m-a-m resonance, a certain minimum air space is necessary before double-panel construction becomes significantly better than a single panel. Figure 9.22 shows a series of transmission loss tests (Vinokur, 1996) for single-glazed and various configurations of dual-glazed windows. The large dip at about 700 Hz for curve 6 corresponds to the m-a-m resonant frequency. Curves 2 and 4 illustrate the effects of mechanical coupling through direct connection between the window frames. Bridging provides a short circuit across the main air gap and magnifies the importance of the coincidence effects.

Additional tests by Vinokur using 3 mm (1/8") laminated and plate glass are summarized in Fig. 9.23. An STC of 34 was achieved with the two sheets of glass laminated together, and this result was not bettered for double-glazed windows until the airspace exceeded 25 mm (1"). The lesson here is that not all double-glazed windows are better than single-glazed even when the total overall thickness of glass is the same. When the spacing between panels is small, it is not uncommon to find that a double-panel construction has a lower transmission loss than that of the two individual single panels sandwiched together. This accounts for the failure of thin double-panel windows, sometimes referred to as thermopane windows, to provide appreciable noise reduction.

In the region around coincidence, panels may be deadened using a sandwiched damping compound. In the Vinokur tests, glass panels were laminated together using 0.7 or 1.4 mm thick polyvinylbutyral. The coincidence dip for a laminated panel (curve 6) is significantly smaller than for single (curve 5) or double (curve 2) plate glass, where there is structural bridging. Harris (1992) reports that a sandwiched viscoelastic layer contained between two layers of 5/8" plywood can improve the transmission loss of lightweight wood floor systems. Gluing plywood floor sheathing can also be helpful.

Increasing the spacing between the partitions drives down the m-a-m resonance frequency and increases the transmission loss above that point. Measured window data show that above a spacing of about 20 mm the improvement is about 3.5 STC points per doubling

FIGURE 9.22 Transmission Loss of Various Glazing Configurations (Vinokur, 1996)

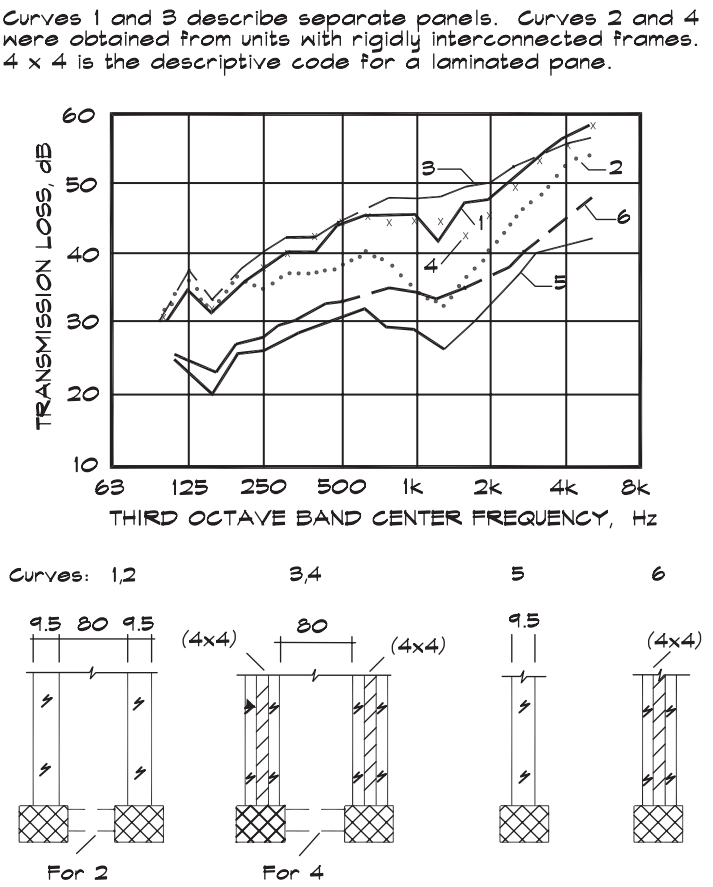


of distance between the panes and about the same increase per doubling of mass. Figure 9.24 gives laboratory data taken by Quirt (1982) for various thicknesses and spacings of window glazing. The upward trend for both mass and spacing is clear.

In the case of gypboard panels, the coincidence frequency can be kept high by using multiple sheets of differing thicknesses that are spot laminated together. In other cases, panels can be designed so that the coincidence frequency is greater than the frequency range of interest. Where thick panels such as concrete slabs or grouted concrete masonry units are used, an additional separately supported panel of gypsum board can be used to offset the low shear frequency of the concrete with a high coincidence frequency in the gypboard. In the coincidence region, the ideal mismatch is on the order of a factor of two; however, more modest ratios can provide several dB of benefit.

The coincidence effect is emphasized, perhaps overemphasized, by the STC test procedure since the knee in the standard STC curve falls at about half the critical frequency of 16-mm (5/8") drywall. Due to this fact double walls with single 13 mm (1/2") gypboard often test higher than 16 mm (5/8") gypboard and 10 mm (3/8") higher than 13 mm (1/2") (California Office of Noise Control, 1981). For a given mass of panel the thinner material is more effective due to the higher critical frequency. A higher STC value in these cases does not necessarily mean that a given construction is more effective at reducing noise, since

FIGURE 9.23 Transmission Loss Tests on Laminated and Double Glazed Windows (Vinokur, 1996)

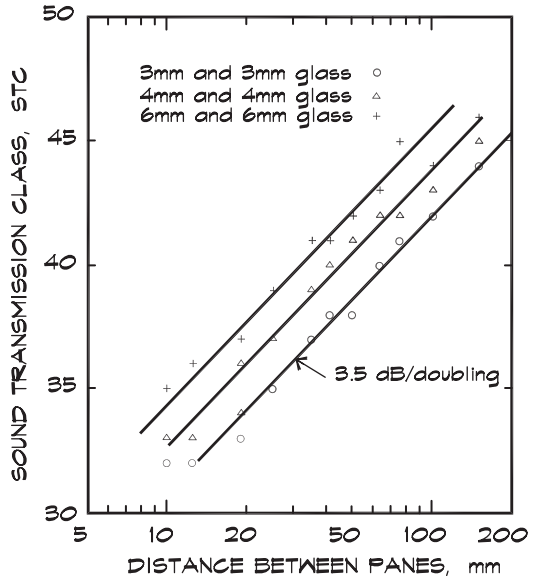


noise reduction depends on the source spectrum. At low frequencies, the heavier panels are more effective regardless of their STC rating.

There is a common misunderstanding, particularly in the design of windows in sound studios, that the slanting of one of the panes of glass with respect to the other provides some acoustical benefit. The mass-air-mass resonance is a bulk effect and is shifted only slightly by the use of a slanted panel. A careful study of window design effects, undertaken by Quirt (1982), showed that slanting had no appreciable effect on the transmission loss of double pane windows. However the technique of slanting one sheet of glass in studio control room windows is useful in reducing the ghost images from light reflections between the panes.

The introduction into the airspace of a gas such as argon, carbon dioxide, or SF_6, which have a sound velocity lower than that of air, has much the same effect as batt insulation in the cavity. Gasses that have a higher velocity than air, such as helium, can also be effective. These gasses are used in hermetically sealed dual-paned windows. The mismatch in velocity reduces the coincidence effect since, when the interior wave matches the bending velocity, the exterior wave does not.

FIGURE 9.24 STC vs Interpane Spacing for Separately Supported Double Glazing (Quirt, 1982)



9.4 TRIPLE-PANEL TRANSMISSION LOSS THEORY

Free Triple Panels

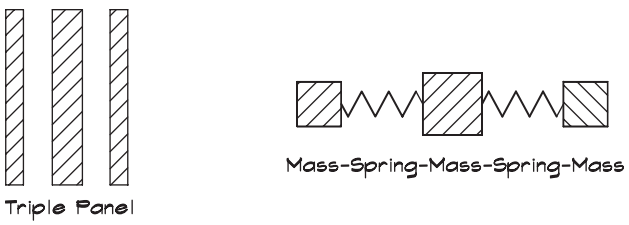
The impedance of ideal triple panels, with no mechanical connections, can be calculated in a similar fashion to that used to obtain the double panel result. The system is pictured in Fig. 9.25. It consists of three masses and two air springs and is a two-degree of freedom system. As such it has two resonant frequencies given by the equation (Vinokur, 1996)

$$f_{\alpha,\beta} = \frac{1}{2\pi}\sqrt{3.6\,\rho_0\,c_0^2}\,\sqrt{a \pm \sqrt{a^2 - b}} \qquad \left(f_\beta > f_\alpha\right) \tag{9.52}$$

where $$a = \frac{1}{2\,m_2}\left(\frac{m_1 + m_2}{m_1\,d_1} + \frac{m_2 + m_3}{m_3\,d_2}\right)$$

$$b = \frac{M}{m_1\,m_2\,m_3\,d_1\,d_2}$$

FIGURE 9.25 General Model of a Triple Panel System



m_i = mass of the i th panel (kg / m^2 or lbs / ft^2)
d_j = spacing of the j th air gap between the layers (m or ft)
$M = m_1 + m_2 + m_3$

The resulting transmission loss can be simplified (Sharp,1973) to provide approximations, which fall into three frequency ranges

$$\Delta L_{TL} \cong \begin{cases} 20 \log (M f) - K_{TL} & f < f_\alpha \\ \Delta L_{TL1} + \Delta L_{TL2} + \Delta L_{TL3} + 20 \log (2 k d_1) + 20 \log (2 k d_2) & \\ & f_\beta < f < f_l \\ \Delta L_{TL1} + \Delta L_{TL2} + \Delta L_{TL3} + 12 & f_l > f \end{cases} \quad (9.53)$$

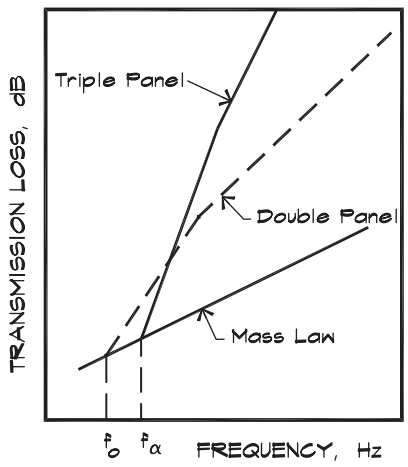
Below the lower mass-air-mass resonance, the triple-panel construction reverts to mass-law behavior. Above the higher mass-air-mass resonance, but lower than the lowest cavity resonance divided by π (as defined by Eq. 9.51), the transmission loss increases at a rate of 30 dB per octave compared with 18 dB per octave for the double panel. In this frequency range the transmission loss is increased 18 dB per doubling of mass.

Comparison of Double and Triple-Panel Partitions

It is useful to compare the behavior of double- and triple-panel walls, which are available for the construction of critical separations such as those found in sound studios and high-quality residential construction. At low frequencies, below f_0, the mass law applies to both types of construction, so for a given mass and thickness the performance is the same. Figure 9.26 shows a graph of the transmission loss achieved by the two types. Here the triple panel has a symmetric construction with the center panel having twice the mass of the outer panels and equal air spaces on either side. For this configuration, the lowest mass-air-mass resonant frequency, f_α, is twice that of the double construction frequency f_0. The two transmission

FIGURE 9.26 Double and Triple Panel ΔL_{TL} Comparison (Sharp, 1973)

A comparison of the transmission loss provided by double and triple panel constructions of equal total mass and overall thickness.

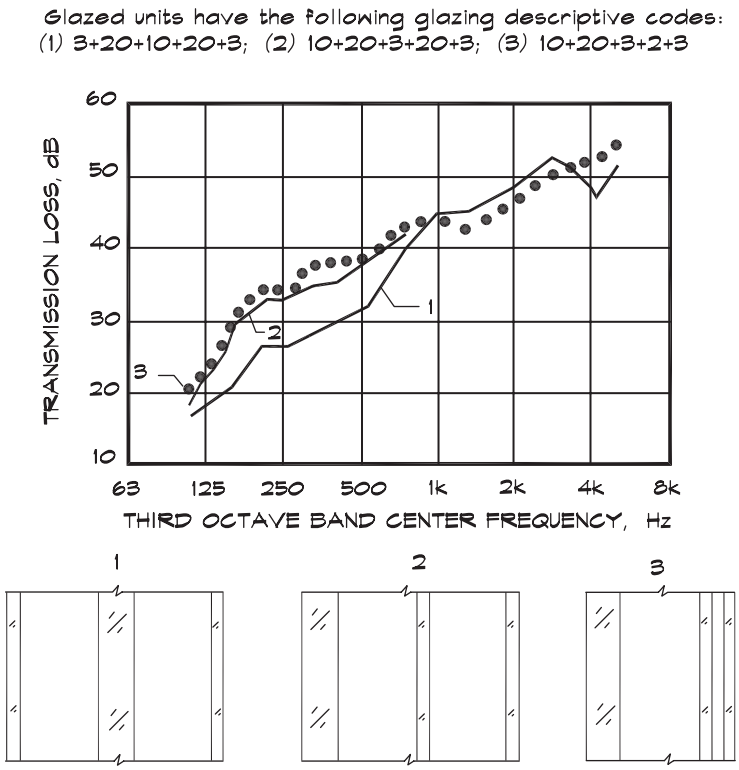


loss curves cross at a frequency that is four times f_0. Thus, assuming no connections between the panels, the double panel system is better below $4f_0$ and the triple panel is better above $4f_0$. For a wall consisting of one layer of 16 mm (5/8") gypboard on the outer panels and two layers on the inside, separated by two 90 mm (3.5") airspaces, the crossover point $4f_0$ is about 260 Hz.

In general, walls with double panels are more efficient for the isolation of music, where bass is the main concern, and triple panels for the isolation of speech. Triple panels, however, are more forgiving of poor construction practice, such as improperly isolated electrical boxes and noisy plumbing, since they provide an additional layer in the middle of the wall or floor. These practical effects in many cases may be more important than a theoretically higher transmission loss.

For a given total mass and thickness, triple-glazed windows are not as effective as double glazing, since the separation between panes in triple-glazed windows is not large. They are most effective when the heaviest sheet of glass is an outer layer and not in the center, and when the airspace dimensions are not the same. Figure 9.27 illustrates this point through a series of tests carried out by Vinokur (1996). Example 2 shows the effect of the transposition of position of the heavy and light sheets of glass. Example 3 shows that it is effective to closely space the two light panes of glass even when the larger airspace does not increase. This configuration acts more like a double-glazed unit. At the higher frequencies, the primary transmission path in windows is from pane to pane through the edge connections at the frame.

FIGURE 9.27 Measurements of Triple Panel Glass Units (Vinokur, 1996)



9.5 STRUCTURAL CONNECTIONS

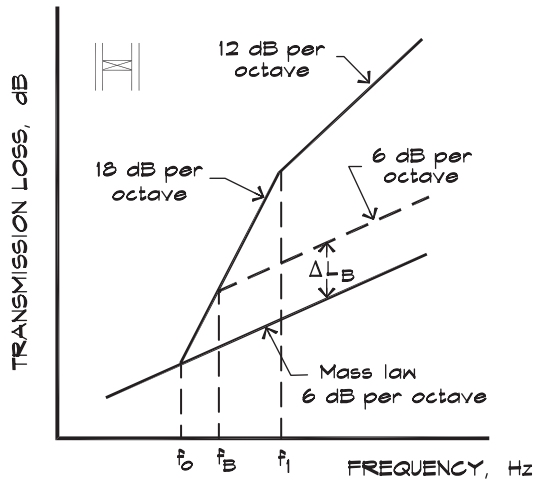
Point and Line Connections

In the previous theoretical analysis, it has been assumed that there were no structural connections linking the panels, in either the double or the triple constructions. The only means of connection between the panels was through the coupling provided by the air space. In actual building construction, there are studs, joists, and other elements that provide the structural support system and physically connect panels together. These produce a parallel means of sound transmission, by way of mechanical connections from one panel to another. In general there are two types of connections: a line connection, such as that provided by wood or metal studs, which touch the panels continuously along their length; and a point connection, in which the contact is made over a relatively small area that approximates a point.

The theory published by Sharp (1973) for the transmission of sound through mechanical connections assumes that the power transmitted by the panel on the receiving side is proportional to the square of the velocity of the panel and to its radiating area. It further assumes that, in the local area around the connection, the panel velocity on the receiving side is the same as on the sending side, and that the sending side panel is unaffected by the presence of the receiving side panel. He develops the equivalent radiating area for point and line connections, since these areas act as localized sources, which transmit energy in addition to that transmitted by the air-gap transmission path. The bridging energy transmitted is proportional to the transmitting area, which decreases with frequency since its effective radius is about one-quarter of the bending wavelength. The power radiated by this area, however, increases with frequency so that the total bridging energy transmitted stays constant.

The effect of sound bridges in double-panel construction is shown in Fig. 9.28. The normal double-panel transmission loss is truncated by the extra energy transmitted through the solid connections above the bridging frequency. The limiting value of the transmission

FIGURE 9.28 Transmission Loss of a Double Panel with Structural Bridges (Sharp, 1973)



loss above the composite mass law is given for point connections (Sharp, 1973)

$$\Delta L_{BP} = 20 \log (e f_c) + 20 \log \left(\frac{m_1}{m_1 + m_2} \right) - K_{BP} \tag{9.54}$$

and for line connections

$$\Delta L_{BL} = 20 \log (b f_c) + 20 \log \left(\frac{m_1}{m_1 + m_2} \right) - K_{BL} \tag{9.55}$$

where e = point lattice spacing (m or ft)
b = line stud separation (m or ft)
m_1 = mass per unit area of the panel (kg / m^2 or lb / ft^2)
m_2 = mass per unit area on the side supported by point or line connections (kg / m^2 or lb / ft^2)
f_c = critical frequency of the panel supported by point connections or, in the case of line connections, the higher critical frequency of the two panels (Hz)
K_{BP} = constant for point connections
= − 44.7 for metric units
= − 55 for FP units
K_{BL} = constant for line connections
= − 22.8 for metric units
= − 28 for FP units

The bridging frequency for point connections is

$$f_{BP} = f_0 \left(\frac{e}{\lambda_c} \right)^{1/2} \tag{9.56}$$

and for line connections

$$f_{BL} = f_0 \left(\frac{\pi b}{8 \lambda_c} \right)^{1/4} \tag{9.57}$$

Note that e^2 = the area associated with each point connection (m^2 or ft^2)
f_0 = fundamental resonance of the double panel (Hz)
The overall transmission loss for a bridged double panel is given by

$$\Delta L_{TL} \cong \left\{ \begin{array}{ll} 20 \log (M f) - K_{TL} & f < f_0 \\ \Delta L_{TL1} + \Delta L_{TL2} + 20 \log (2 k d) & f_0 < f < f_B \\ 20 \log (M f) - K_{TL} + \Delta L_B & f_B < f < 0.5 f_c \end{array} \right\} \tag{9.58}$$

FIGURE 9.29 ΔL_{TL} for a Point Mounted Double Panel Sharp (1973)

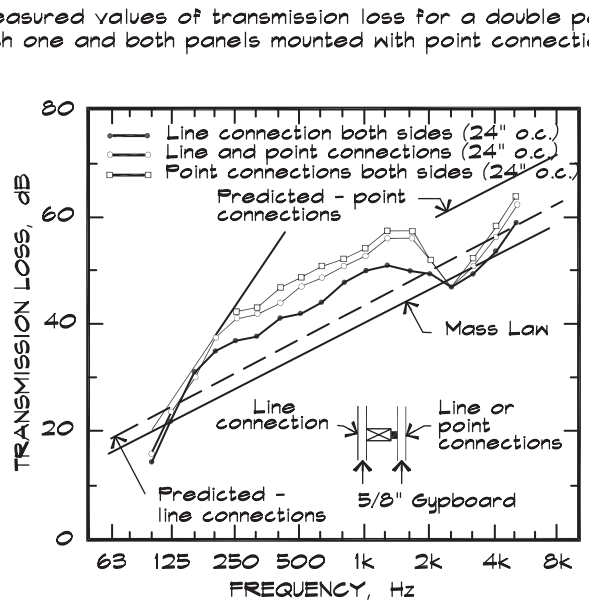


Figure 9.29 shows the results of measurements of bridged double panels done by Sharp (1973). The figure illustrates that there is little advantage to mounting both sides on point connections. Note that Eq. 9.58 does not account for the behavior at the critical frequency where the energy is transmitted by an airborne path. At the critical frequency, the various connection methods yield the same results.

When making a comparison between theoretical and laboratory transmission loss data, calculated data yield a higher result. Better agreement is obtained by assuming line bridging at a spacing based on edge connections for the panel area specified in the test standard.

Transmission Loss of Apertures

The sound transmission through apertures has been treated in some detail. Wilson and Soroka (1965) utilize an approach that analyzes a normally incident plane wave falling on a circular opening of radius a and depth h. They assume massless rigid pistons of negligible thickness at the entrance and exit of the aperture, which simulate the air motion. The radiation impedance is assumed to be that of a rigid piston in a baffle discussed in Chapt. 6. The transmission coefficient is

$$\tau = 4\,w_r \left\{ \begin{array}{l} 4\,w_r^2(\cos k\,h - x_r\ \sin k\,h)^2 + \\ \left[(w_r^2 - x_r^2 + 1)\ \sin k\,h + 2\,x_r\ \cos k\,h\right]^2 \end{array} \right\}^{-1} \tag{9.59}$$

where w_r and x_r are the resistive and reactive components of the mechanical radiation impedance of the piston radiator from Eq. 6.65. The real and imaginary parts are given in Eqs. 6.66 and 6.68.

FIGURE 9.30 Transmission Loss through a 1" (25 mm) Diameter 4"(100 mm) Long Hole (Gibbs and Balilah, 1989)

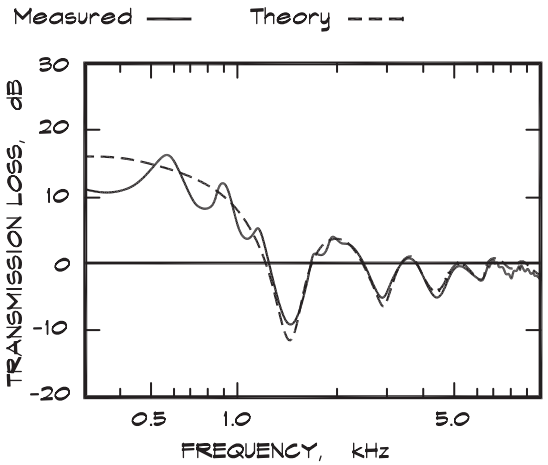
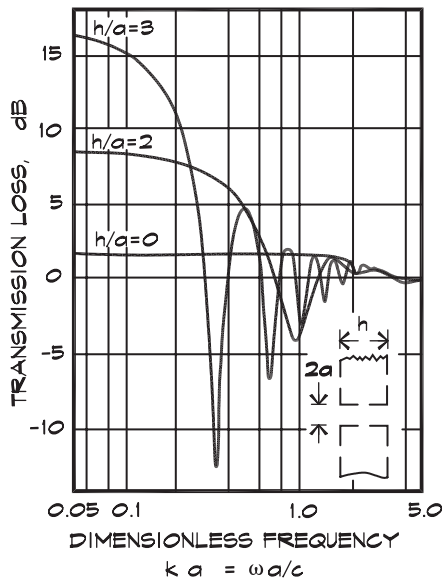


FIGURE 9.31 Transmission Loss through a Circular Aperture vs Frequency for Various h/a Ratios (Wilson and Soroka, 1965)



Apertures in enclosures can create Helmholtz resonator cavities, which increase the noise emitted from a source housed within them in much the same way that bass-reflex enclosures are used with cone loudspeakers to augment the low-frequency response. Figure 9.30 (Gibbs and Balilah, 1989) plots the theoretical and measured transmission loss of a 1" (25 mm) diameter hole having a 4" (100 mm) length. At low frequencies there is

about a 16 dB (theoretical) loss in this example, which drops sharply at the first open-open tube resonance. The transmission loss is negative in the vicinity of the tube resonant frequencies and gradually goes to zero as the wavelength approaches the hole diameter. Similar results were obtained by Sauter and Soroka (1970) for rectangular apertures. The loss at low frequencies depends on the ratio of the depth, h, to the hole radius, a. Figure 9.31 illustrates this effect.

This page intentionally left blank

10 SOUND TRANSMISSION IN BUILDINGS

10.1 DIFFUSE FIELD SOUND TRANSMISSION

Reverberant Source Room

The problem of sound transmission between rooms is one of considerable interest in architectural acoustics. When two rooms are separated by a common wall having an area S_w, as shown in Fig. 10.1, we model (Long, 1987) the behavior by first assuming that there is a diffuse field in the source room that produces a sound pressure p_s and a corresponding intensity

$$I_s = \frac{p_s^2}{4\,\rho_0\,c_0} \tag{10.1}$$

that is incident on the intervening partition. A fraction τ of the incident energy is transmitted into the receiving room

$$W_r = I_s\,S_w\,\tau = \frac{p_s^2\,S_w\,\tau}{4\,\rho_0\,c_0} \tag{10.2}$$

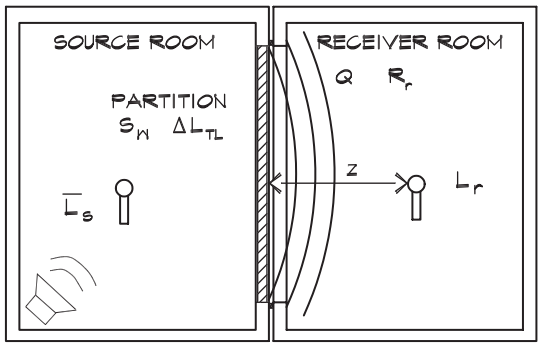
where it generates a sound pressure due to both the direct and the reverberant field contributions. Since the partition is a planar surface, we use Eq. 2.91 for the direct field and Eq. 8.83 for the reverberant field portion of the energy. The receiving room sound energy is

$$\frac{p_r^2}{\rho_0 c_0} = \frac{p_s^2\,S_w\,\tau\,Q}{16\,\pi\,\rho_0\,c_0\left[z+\sqrt{\dfrac{S_w\,Q}{4\,\pi}}\right]^2} + \frac{p_s^2\,S_w\,\tau}{R_r\,\rho_0\,c_0} \tag{10.3}$$

which we can convert to a level relationship by taking 10 log of each side and using the definition of the transmission loss

$$\Delta L_{TL} = -10\log\tau \tag{10.4}$$

FIGURE 10.1 Sound Transmission between Rooms



to obtain the expression for the transmission between two rooms for a diffuse source field and a combination of a direct and diffuse receiving room sound field

$$L_r = \bar{L}_s - \Delta L_{TL} + 10 \log \left[\frac{S_w Q}{16 \pi \left[z + \sqrt{\frac{S_w Q}{4 \pi}} \right]^2} + \frac{S_w}{R_r} \right] \tag{10.5}$$

where L_r = sound pressure level at a point in the receiver room (dB)
$\bar{L}_s$ = diffuse sound pressure level in the source room (dB)
the line over the L denotes a spatial average throughout the room
ΔL_{TL} = transmission loss (dB)
S_w = area of the transmitting surface (m^2 or ft^2)
R_r = room constant in the receiving room (m^2 or ft^2 sabins)
z = distance from the surface of the source to the receiver (m or ft)
Q = directivity of the wall (usually 2)

If the receiving room is very reverberant, the S_w/R_r term is larger than the direct field term and Eq. 10.5 can be simplified to the equation we obtained in Chapt. 9 for the transmission loss between two reverberant rooms

$$\bar{L}_r \cong \bar{L}_s - \Delta L_{TL} + 10 \log \left[\frac{S_w}{R_r} \right] \tag{10.6}$$

It is important to realize that although Eq. 10.6 is accurate for reverberant spaces with good diffusion, it is not accurate when the receiver is close to a transmitting surface or when the absorption in the receiving space is large. For example, if the receiving space is outdoors where the room constant is infinite this equation predicts that no sound will be transmitted.

Sound Propagation through Multiple Partitions

When two reverberant rooms are separated by a partition consisting of two separate components, such as a wall with a window in it, each having a different transmission loss, a *composite transmission loss* may be calculated based on Eqs. 10.4 and 10.6

$$\Delta L_{TL} = 10 \log \left(\frac{S_w}{S_1 \tau_1 + S_2 \tau_2 + \cdots + S_n \tau_n} \right) \tag{10.7}$$

Using this expression, it soon becomes clear that the component having the lowest transmission loss will control the process. It is much like having a bucket full of water with several holes in it. The largest hole (lowest transmission loss) controls the rate at which water flows out. Let us take, for example, the case where a $3\overset{0}{-} \times 4\overset{0}{-}$ (915 mm × 1220 mm) window having a 25 dB transmission loss occupies part of a $20\overset{0}{-} \times 8\overset{0}{-}$ (6.1 m × 2.4 m) gypboard and stud wall having a transmission loss of 45 dB. The composite transmission loss may be calculated

$$\Delta L_{TL} = 10 \log \left(\frac{160}{12(.0034) + 148(.00003)} \right) = 35.5 \text{ dB}$$

Thus, although the window has a much smaller area than the wall, it significantly reduces the overall transmission loss of the composite structure.

Composite Transmission Loss with Leaks

An even more dramatic example of a reduction in composite transmission loss is that produced by a zero transmission loss path such as an opening under a door. Using a $3\overset{0}{-} \times 6\overset{8}{-}$ (0.9 m × 2 m) solid core door having a transmission loss of 30 dB and a 1/2" (13 mm) high opening under the door with a transmission loss of zero dB (at high frequencies), we obtain an overall loss of

$$\Delta L_{TL} = 10 \log \left(\frac{20.125}{20(.001) + .125(1)} \right) = 21.4 \text{ dB}$$

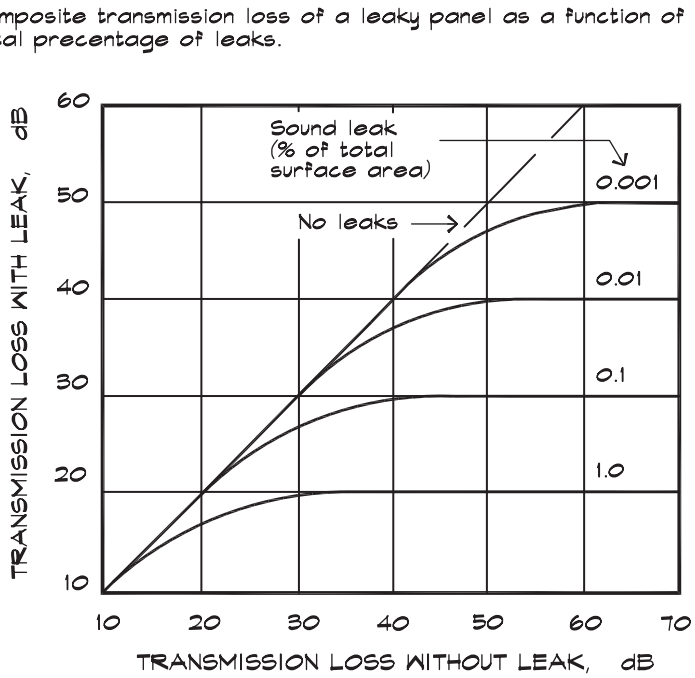
In this case, 8 dB, more sound energy comes through the slot under the door than through the remainder of the door. Figure 10.2 shows the effects leaks on the overall transmission loss of a structure. The relative area of the leak when compared with the overall area of the partition determines the composite transmission loss of the structure in the diffuse field model.

Transmission into Absorptive Spaces

Where sound is transmitted from a reverberant space, through a partition, and into an absorbent space, we can no longer use the approximations given in Eqs. 10.6 and 10.7. Instead, we must use Eq. 10.5, which includes consideration of the direct field contribution in the receiving space. Under these conditions, the composite transmission loss equation becomes inaccurate and we must calculate the energy contribution from each transmitting surface separately, and combine the levels in the receiving space. The results are then dependent on the physical proximity of the receiver to each transmitting surface.

If we repeat the calculation we just did using the half-inch crack under a solid core door, we will get a different answer for different distances between the observer and the

FIGURE 10.2 Composite Transmission Loss with Leaks (Reynolds, 1981)



transmitting surfaces. Let us assume that the receiver is located 2 feet (0.6 m) away from the door in a space having a room constant of 1000 sq. ft. (93 sq. m) sabins, and that there is an 80 dB sound pressure level in the source room. A computation of the level through the door yields a 39 dB level in the receiving room through this path. We then perform a separate calculation for the hole. If the observer is kneeling 2 feet from the hole, the resulting level is 51 dB and the combined (door + hole) level is also 51 dB. If instead, he is standing 2 feet from the door and 6 feet from the hole, the resultant level is 44 dB and the combined level is 45 dB, which is significantly less. Note that for the same conditions Eq. 10.6 and 10.7 would predict 42 dB—less than either of the other two answers since there is no direct-field contribution.

For sound that is radiated from an enclosed reverberant space into the outdoors there is no longer a reverberant field in the receiving space so the room constant goes to infinity. Equation 10.5 then reduces to

$$L_r = \overline{L}_S - \Delta L_{TL} + 10 \log \left[\frac{S_w Q}{16 \pi \left[z + \sqrt{\frac{S_w Q}{4 \pi}} \right]^2} \right] \tag{10.8}$$

If we use Eq. 10.8 to calculate the expected level for a receiver in the free field, close to a radiating surface, where z is nearly zero, we obtain

$$L_r \cong \overline{L}_S - \Delta L_{TL} - 6 \tag{10.9}$$

When the distance between the surface and the receiver is large, Eq. 10.8 becomes

$$L_r \cong \overline{L}_S - \Delta L_{TL} + 10 \log \left(S_w\, Q/16\, \pi\, z^2\right) \tag{10.10}$$

A similar approximation can be made in an enclosed receiving space when the receiver is sufficiently far from the transmitting surface that the receiving area is large compared with the area of the surface

$$L_r \cong \overline{L}_S - \Delta L_{TL} + 10 \log \left[\frac{S_w\, Q}{16\, \pi\, z^2} + \frac{S_w}{R_r} \right] \tag{10.11}$$

When there are multiple transmitting surfaces, the energy contribution through each surface must be calculated and the energies added to obtain the overall receiver level.

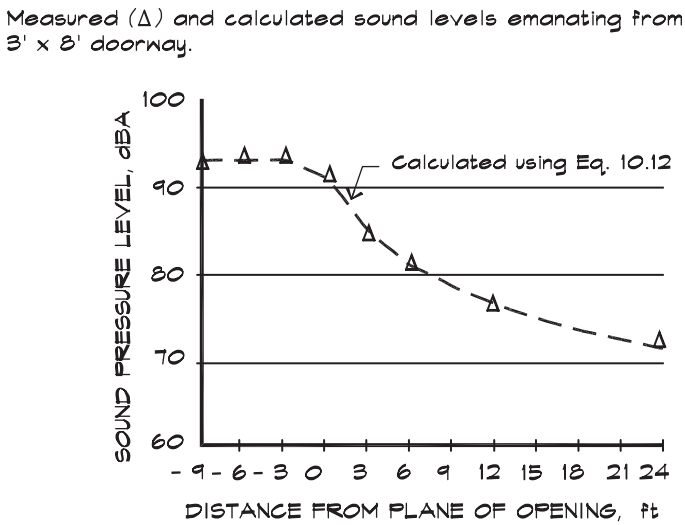
Transmission through Large Openings

When sound is transmitted through an opening in a wall that is large compared with a wavelength, an adjustment must be made to Eq. 10.8. When the formula for transmission loss is derived, it is based on the intensity passing through an area, a fraction of which is transmitted into a partition. The reason the intensity, a vector quantity, is used is that for the mass law model a panel moves as a monolithic object along one axis, normal to its surface. The only forces that move it are those with components along the normal. When reverberant sound energy passes through a large opening, all the energy falling on the opening passes through, not just the components normal to the surface. Consequently the energy transmitted is twice the reverberant-field intensity times the area as in Eq. 8.80. The difference lies in the fact that there is no longer a cosine term to integrate in the conversion from energy into intensity. If we use Eq. 10.8, whose derivation was based on intensity, the calculated value underpredicts the actual result. For transmission through a large opening in a wall, such as a door or window having a zero transmission loss,

$$L_r = \overline{L}_S + 10 \log \left[\frac{S_w\, Q}{8\, \pi \left[z + \sqrt{\dfrac{S_w\, Q}{4\, \pi}} \right]^2} \right] \tag{10.12}$$

Hessler and Sharp (1992) have tested this relationship by measuring the sound pressure level, generated by a reciprocating compressor in a reverberant concrete-and-steel mechanical equipment room, passing through a doorway into an open yard. The results are shown in Fig. 10.3 along with calculated levels for a Q of 4. The measured values closely match the predicted levels, indicating baffling by both the wall and the ground. It is interesting to note in the figure that the measured level at $z = 0$ is 3 dB below the interior level. This transitional behavior can be expected to occur not only for an opening, but also with any porous material, whose impedance was primarily due to flow resistance rather than mass. If the transmission loss of such a material were measured in the laboratory, the effect would be included in the measured data so the normal equation could be used, although negative transmission loss values might be found for light materials.

FIGURE 10.3 Sound Level Falloff from an Open Doorway (Hessler and Sharp, 1992)



Noise Transmission Calculations

Calculations of sound propagation between spaces are done with diffuse-field transmission loss data, in individual octave or third-octave bands, and the receiving levels are combined to obtain an overall result. Most transmission loss data on walls and other components are measured in third octaves and source data (either sound power levels or sound pressure levels) are available in octave bands. Absorption coefficient data also are published primarily in octave bands. If third octave transmission loss data are used in an octave-band calculation, a composite value should be calculated.

$$\Delta L_{TL} = -10 \log \frac{1}{3} \sum_{i=1}^{3} 10^{-0.1 \Delta L_{TL_i}} \tag{10.13}$$

The exact value of course depends on the actual source spectrum and cannot be determined a priori. The same equation can be applied to the composite dynamic insertion loss of silencers.

When laboratory transmission loss data are used to predict sound levels under field conditions, the standard expectation is about a 5 dB underestimation of the receiver levels when Eq. 10.6 is used. This is acknowledged in many standards. For example, the California Noise Insulation Standards (1974) require the use of minimum STC 50 rated walls and floor-ceiling systems between dwelling units, but allow an FSTC rating of 45 under a field test. The normal reason given for this difference is the care taken in the construction of laboratory test partitions as compared to typical construction practice. A portion of the difference between laboratory and field test results may be due to the lack of consideration of the direct field contribution as well as the lack of the purely diffuse field in the receiving space assumed in the standard formula.

10.2 STC RATINGS OF VARIOUS WALL TYPES

Laboratory vs Field Measurements

The sound transmission class (STC) ratings of various construction elements are of considerable interest to architects and acoustical engineers. Although these ratings are in general use, it is also important to examine the third octave band transmission loss data that form the basis for the rating and to compare it to the theoretical predictions discussed previously. Several sets of measured data are included, which are based in part on the State of California compendium of STC and IIC ratings. It is important to note that an STC rating should not be the sole basis on which a decision to use a particular construction is made. This is particularly true in the case of floor-ceilings, where the length of the structural span and floor coverings play a major role.

Measurements of the sound transmission class in the field, as contrasted to the laboratory, are designated FSTC ratings. It is generally agreed that these ratings are five or more points lower than the laboratory ratings. The reason for this difference is attributed to the extra care in blocking the various flanking paths associated with laboratory tests and the lack of electrical outlets and other paths. It may also be in part attributable to more absorption in a typical receiving space and to the fact that the direct field transmission is not considered in the standard equation. In either case, it is prudent to design critical partitions with a margin of safety, which takes into account the expected in-field performance.

Single Wood Stud Partitions

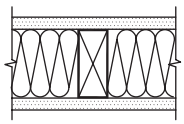
Several single wood stud walls are shown in Fig. 10.4. Note that the effectiveness of batt insulation is much less than the ideal improvement from Fig. 9.19. This is because much of the sound is transmitted through structural coupling by the studs. Nevertheless, it is important to include batt insulation for sound control even in single stud walls.

Single Metal Stud Partitions

Examples of metal stud partitions are shown in Fig. 10.5. Single lightweight (26 Ga) metal studs are more effective than wood studs since they are inherently flexible. The studs themselves act as vibration isolators and decouple one side from another, thereby reducing structureborne noise transmission. Consequently, it is of little value to add resilient channel or other flexible mounts to nonstructural metal studs. The method of attachment also affects the transmission loss. Panels that are glued continuously to studs yield lower transmission loss values than panels that are screw attached. The gluing apparently increases the stiffness of the stud flange, which then increases transmission via the studs (Green and Sherry, 1982).

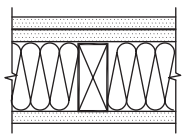
Gypsum board panels are lifted into place during construction using a spacer under their bottom edge, so there is a 3 mm to 6 mm (1/8" to 1/4") gap at the bottom of the sheet. Holes such as these must be sealed if the transmission loss of the construction is to be maintained. Closing off openings in partitions is critical to acoustical performance, particularly for the case of high transmission loss partitions. Gaps are closed off with a nonhardening caulk so the acoustical rating of the wall can be maintained. Figure 10.6 shows the effect of gaps on the STC ratings. Similar openings can be left by electrical box penetrations, pipe penetrations, cutouts for medicine cabinets, light fixtures, and duct openings. Caulk should not be used

FIGURE 10.4 Transmission Loss of Single Wood Stud Walls (California Office of Noise Control, 1981)



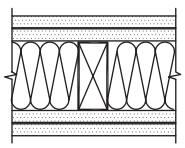
Single Wood Stud Wall - 2 x 4 Stud
1 Layer 5/8" Drywall Each Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1 V	1.3 V	1.6 V	2 V	2.5 V	3.2 V	4 V	STC
A	15	15	20	24	27	29	32	36	39	41	41	40	37	36	39	43	34
B	17	15	25	35	34	32	37	40	40	43	44	45	37	34	41	47	36



Single Wood Stud Wall - 2 x 4 Stud
1 Layer 5/8" Drywall One Side
Two Layers Opposite Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1 V	1.3 V	1.6 V	2 V	2.5 V	3.2 V	4 V	STC
A	19	19	23	26	28	28	34	36	39	38	38	41	43	44	49	52	36
B	19	21	27	30	32	32	36	40	42	40	40	43	44	46	50	52	39



Single Wood Stud Wall - 2 x 4 Stud
2 Layers 1/2" Drywall Each Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1 V	1.3 V	1.6 V	2 V	2.5 V	3.2 V	4 V	STC
A	15	19	35	35	34	42	43	43	47	48	50	52	53	48	46	50	39
B	21	24	40	37	36	42	45	46	48	50	51	54	55	50	47	51	45

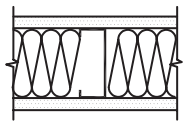
to span more than a 6 mm (1/4") gap. Larger openings should be filled with drywall mud or gypboard.

Resilient Channel

Resilient channel is a flexible strip of metal designed to support layers of gypboard, while providing a measure of mechanical isolation against structure borne vibrations. A group of wall constructions is shown in Fig. 10.7. Note that the channel is attached to the studs only on one side. Resilient channel should be installed with the open side up so that the weight of the gypboard tends to open the gap between the stud and the board. A filler strip is used at the base plate as a solid protection against impact. The gypboard is attached with drywall screws to the channel and care must be exercised to avoid screwing through the channel and into the studs, which short-circuits the isolation. When there are bookcases or other heavy objects that must be wall mounted, resilient channel is not a good choice since these items must be bolted through into the structure.

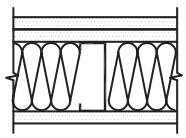
There are a number of products called resilient channel on the market. Some are more effective than others. A type that is spoon-shaped and can be attached only on one side is preferable to the furring channel type, which is hat-shaped and may be attached on both sides. The latter is often improperly installed, rendering it ineffective.

FIGURE 10.5 Transmission Loss of Single Metal Stud Walls (California Office of Noise Control, 1981)



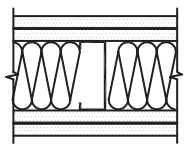
Single Metal Stud Wall - 3 5/8" Stud
1 Layer 5/8" Drywall Each Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1 ⅓	1.3 ⅓	1.6 ⅓	2 ⅓	2.5 ⅓	3.2 ⅓	4 ⅓	STC
A	20	23	30	31	30	34	40	41	45	48	49	47	39	38	40	46	39
B	29	35	40	42	43	48	52	55	57	57	57	54	45	40	44	50	44



Single Metal Stud Wall - 3 5/8" Stud
1 Layer 5/8" Drywall One Side
Two Layers Opposite Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1 ⅓	1.3 ⅓	1.6 ⅓	2 ⅓	2.5 ⅓	3.2 ⅓	4 ⅓	STC
A	29	31	35	38	41	44	48	47	49	51	53	56	53	44	41	46	45
B	35	37	40	39	40	42	44	47	50	54	56	55	50	44	46	50	47



Single Metal Stud Wall - 3 5/8" Stud
2 Layers 5/8" Drywall Each Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1 ⅓	1.3 ⅓	1.6 ⅓	2 ⅓	2.5 ⅓	3.2 ⅓	4 ⅓	STC
A	27	31	38	41	42	44	47	51	52	52	53	52	49	46	46	49	48
B	32	37	42	45	48	50	51	52	52	52	51	49	49	48	48	50	50

FIGURE 10.6 Dependence of the STC Rating on Caulking (Ihrig, Wilson. 1976)

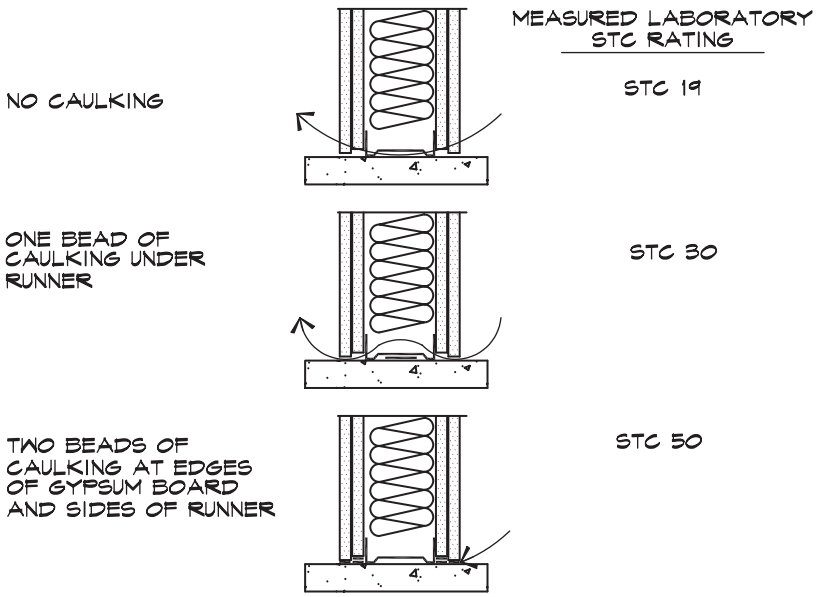
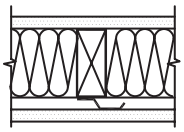
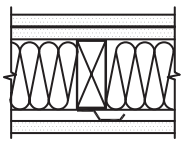


FIGURE 10.7 Transmission Loss of Single Stud Resilient Walls (California Office of Noise Control, 1981)



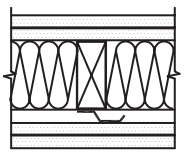
Single Wood Stud Wall - 2 x 4 Stud
1 Layer 5/8" Drywall Each Side
One Side on Resilient Channel
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1 ¼	1.3 ¼	1.6 ¼	2 ¼	2.5 ¼	3.2 ¼	4 ¼	STC
A	16	19	26	26	32	33	37	40	44	45	47	46	40	40	44	48	38
B	29	33	37	40	45	50	51	53	53	54	54	53	47	42	51	57	46



Single Wood Stud Wall - 2 x 4 Stud
1 Layer 5/8" Drywall on Resilient Channel One Side
Two Layers Opposite Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1 ¼	1.3 ¼	1.6 ¼	2 ¼	2.5 ¼	3.2 ¼	4 ¼	STC
A	27	28	34	33	34	37	40	42	44	45	46	49	49	49	53	54	43
B	30	38	41	43	46	47	49	50	51	49	49	51	52	53	55	56	50



Single Wood Stud Wall - 2 x 4 Stud
2 Layers 1/2" Drywall Each Side
One Side on Resilient Channel
A - Without Insulation B - R-11 Batt Insulation

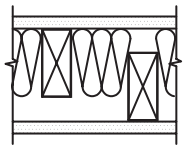
	125	160	200	250	315	400	500	630	800	1 ¼	1.3 ¼	1.6 ¼	2 ¼	2.5 ¼	3.2 ¼	4 ¼	STC
A	28	33	41	43	44	51	51	53	56	56	60	60	61	55	52	57	52
B	33	39	43	46	50	56	58	59	62	61	62	63	62	59	57	58	56

The purpose of resilient channel is to provide a flexible connection to mechanically decouple the partitions on either side of the framing. When the panels are already separately or flexibly supported, the addition of resilient channel does little to improve the transmission loss. Thus there is little or no advantage in adding channel to double stud, staggered stud, or single lightweight metal stud construction.

Resilient channel is not effective when it is installed between two layers of gypboard, since the air gap is small (typically 13 mm or 1/2") and the trapped air creates an air spring, which makes an additional mass air mass resonance. If a single metal stud wall with batt insulation has drywall on each side and another layer is added on resilient channel, the result is worse than without the additional layer (Green and Sherry, 1982).

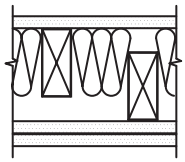
Resilient channel is utilized in floor-ceilings more often than in walls since it is compromised by mounting bookcases or cabinets to the supporting studs. It is only effective in isolating small-amplitude vibrations that are much less than the structural deflection under load. It is generally not effective in preventing the transmission of low-frequency sound created by the large-scale deflection of long-span joists under a dynamic load. However, it can provide an improvement at high frequencies to both the STC and IIC ratings in floor-ceiling systems. In floor-ceilings, it should be installed so that the ceiling gypboard is butted

FIGURE 10.8 Transmission Loss of Staggered Stud Walls (California Office of Noise Control, 1981)



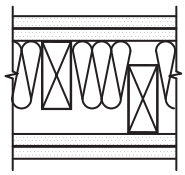
Staggered Wood Stud Wall
2 x 4 Studs on a 2 x 6 Plate
1 Layer 5/8" Drywall Each Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
A	29	30	33	36	39	43	44	44	46	46	50	50	43	39	46	51	43
B	34	36	40	43	45	45	45	46	47	49	50	49	45	42	46	53	46



Staggered Wood Stud Wall
2 x 4 Studs on a 2 x 6 Plate
1 Layer 1/2" Drywall One Side
Two Layers Opposite Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
A	32	32	35	36	39	46	46	49	50	53	54	55	47	44	50	55	47
B	38	41	41	44	47	51	51	53	54	55	56	57	58	54	50	55	53



Staggered Wood Stud Wall
2 x 4 Studs on a 2 x 6 Plate
2 Layers 5/8" Drywall Each Side
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
A	31	35	36	42	45	47	49	51	54	54	56	56	53	51	55	59	51
B	35	39	45	45	49	49	52	53	54	55	56	57	54	50	53	57	53

against the wall gypboard, leaving the resiliently supported surface free to move. If the ceiling surface rests on the wall gypboard, the mechanical isolation is compromised at the edges.

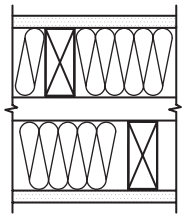
Staggered-Stud Construction

Staggered stud wall construction represents a compromise between single-stud and double-stud construction. The use of staggered wood studs on a common 2 × 6 (38 mm × 140 mm) base plate, which is shown in Fig. 10.8, can provide some mechanical decoupling between the panels on either side of a wall, but is limited by the flanking transmission through the plates. It produces transmission loss values that are comparable to resilient channel and, since a solid stud is used, this wall construction will support bookcases and the like and is thus preferred. It is difficult to use a staggered stud configuration with metal studs because at the top and bottom plates, a continuous runner cannot be used. A 3 5/8” (92 mm) 26 Ga metal stud has significant decoupling, due to its inherent softness, so there is little advantage to staggering metal studs. If a higher transmission loss is required and the width is limited, a double 2 1/2” (64 mm) metal stud with a 1/2” (13 mm) air gap will yield the same wall thickness as a staggered wood stud system and better isolation.

Double-Stud Construction

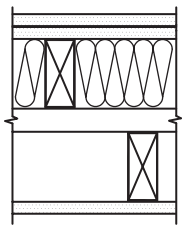
Where high transmission loss values are desired, double-stud construction, with multiple layers of gypboard or heavy plaster, is preferred. The losses are limited by the flanking transmission through the structure, which can be improved by setting one or both sides of the wall on a floating floor or isolated stud supports in specialized applications such as studios. Typical double-stud constructions are given in Fig. 10.9. There is no appreciable difference

FIGURE 10.9 Transmission Loss of Double-Stud Walls (California Office of Noise Control, 1981)



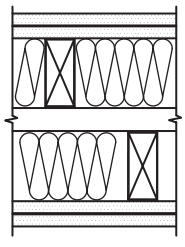
Double Wood Stud Wall
2 x 4 Studs Separated by a 1" Gap
1 Layer 5/8" Drywall Each Side
A - Without Insulation B - R-11 Batt Insulation
C - 2 Layers R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
A	31	34	37	42	43	51	47	50	51	52	54	51	41	41	46	52	45
B	36	41	44	44	51	56	56	58	61	62	63	62	53	52	58	62	56
C	35	39	42	49	52	53	55	56	59	62	64	65	62	59	62	66	57



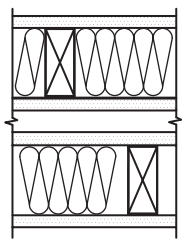
Double Wood Stud Wall
2 x 4 Studs Separated by a 1" Gap
1 Layer 5/8" Drywall One Side - 2 Layers Opposite
A - Without Insulation B - R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
A	33	37	40	47	50	55	53	56	57	56	59	57	48	48	53	58	52
B	38	45	48	49	54	60	60	60	62	62	63	62	55	57	61	63	59



Double Wood Stud Wall
2 x 4 Studs Separated by a 1" Gap
2 Layers 5/8" Drywall Each Side
A - Without Insulation B - R-11 Batt Insulation
C - 2 Layers R-11 Batt Insulation

	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
A	35	39	42	45	48	50	52	55	55	54	56	54	53	53	53	55	54
B	37	38	43	48	51	54	57	59	61	65	69	70	70	67	68	70	58
C	41	46	50	55	58	59	61	62	64	67	69	70	67	66	68	69	63



Double Wood Stud Wall
2 x 4 Studs Separated by a 2 1/4" Gap
1 Layer 5/8" Drywall Each Side of Both Studs
2 Layers R-11 Batt Insulation

125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
22	23	34	45	46	44	49	57	61	65	67	67	60	57	63	68	44

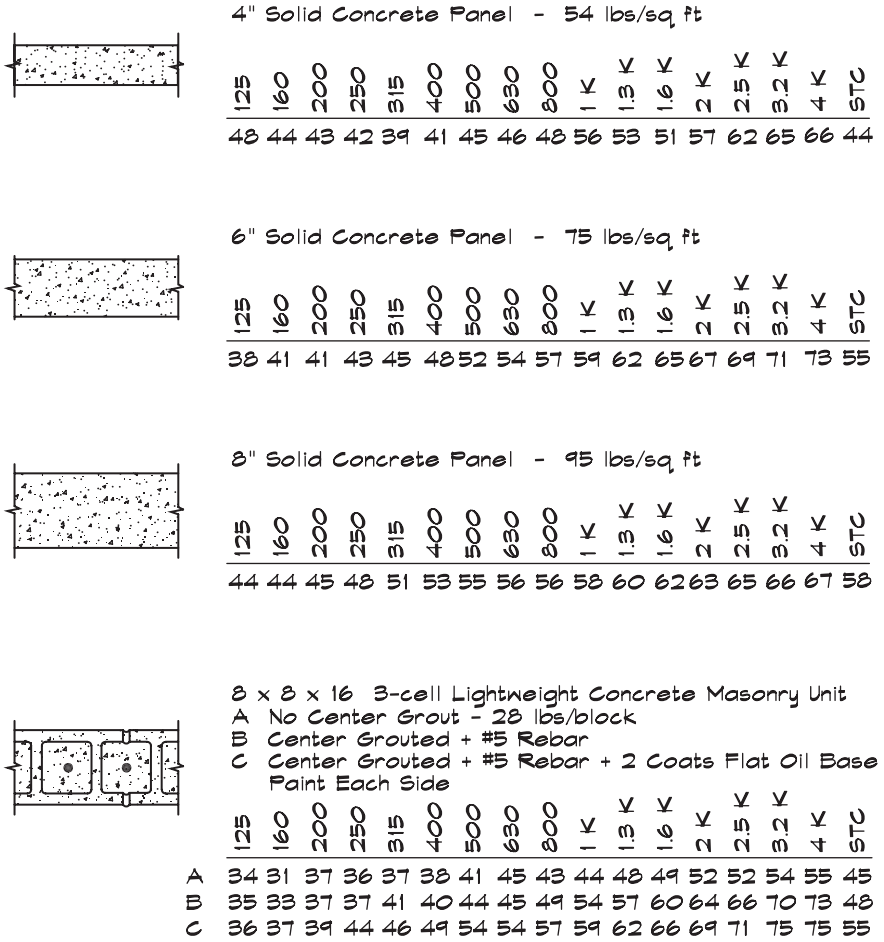
in the performance of wood and metal double studs, since there is no additional decoupling due to the intrinsic stiffness of the stud.

Gypboard layers placed in the air gap between the studs reduce the transmission loss because a bridging air pocket is formed. For a given number of layers it is most effective to place them on the outside faces of the double studs. For example, the last wall shown in Fig. 10.9 rates an STC 44 with inner drywall layers as compared to a 63 rating for the same number of layers on the outside. As the air gap increases this disadvantage is offset by the effectiveness of the separation. If the distance between the studs is several feet, such as two stud walls separated by a corridor, the mass-air-mass resonance is so low that it would have no appreciable effect.

High-Mass Constructions

Heavy materials such as concrete, grouted cmu blocks, concrete-filled metal decking, and similar products can provide substantial transmission loss due to their intrinsic mass. Although a single panel structure is less efficient in the loss per mass than a multiple layer construction, in many cases there is no viable substitute. Figure 10.10 shows the measured

FIGURE 10.10 High Mass Wall Construction (California Office of Noise Control, 1981)

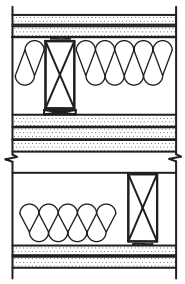


transmission loss values of concrete panels used in wall or floor construction. Note the difference in the grouted block data between painted and nonpainted conditions. Blocks are intrinsically porous and must be sealed with a bridging (oil-based) paint to achieve their full potential.

High Transmission Loss Constructions

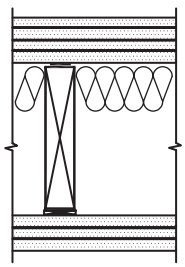
An important study was undertaken by Sharp (1973) to try to develop construction methods that would achieve transmission loss ratings 20 dB or more above the mass law. In this work several techniques were utilized, not normally seen in standard construction practice but which could easily be implemented. These included spot lamination, which has been previously discussed, and point mounting. The point mounting technique he devised was to use 1/4" (6 mm) thick foam tape squares between the gypboard and the stud and then to attach the sheet with drywall screws through the tape into the stud. This technique resulted in panel isolation that approaches the theoretical point mounting discussed in Eq. 9.54. A triple panel wall having an STC of 76 utilizing these techniques is shown in Fig. 10.11. This wall has a relatively low transmission loss value of 33 dB in the 80 Hz band.

FIGURE 10.11 High Transmission Loss Wall Construction (Sharp, 1973)



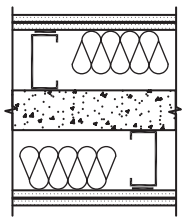
Double Wood Stud Wall
2 x 4 Studs Separated by a 3" Gap
Outside Layers 3/8" + 1/2" Drywall on 1/4" Foam Tape
Inside 5/8" + 1/2" + 5/8" Drywall on 1/4" Wood Squares
All Layers Spot Laminated at 24" OC and Screwed
Through Mounts - 2" Batt Insulation in all Cavities

125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
53	56	60	66	70	75	78	80	78	79	83	83	84	87	88	89	76



Single Wood Stud Wall - 2 x 8 Studs
1/2" + 3/8" + 1/2" + 3/8" Drywall on 1/4" Foam Tape
5/8" + 1/2" + 5/8" Drywall on 1/4" Wood Squares
All Layers Spot Laminated at 24" OC and Screwed
Through Mounts - 2" Batt Insulation in all Cavities

125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
49	53	54	57	60	62	64	66	68	74	79	80	82	83	84	85	69



Concrete and Metal Double Stud Wall
2" Reinforced Concrete Panel
2 Layers 1/4" Drywall On 1/4" Foam Tape Each Side
2 1/2" Metal Studs - All Panels Spot Laminated at 24"
OC and Screwed Throught the Mounts
2" Batt Insulation in the Airspace

125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
49	53	60	63	66	68	71	73	76	78	77	78	81	83	87	88	72

Figure 10.11 also shows a double panel wall having the same mass as the previous construction. This has a lower rating (STC 69) due to reduced performance in the mid-frequencies but much better performance at low frequencies (41 dB at 80 Hz). The multiple layers of spot laminated drywall significantly reduce the coincidence effects. Similar performance should be obtained using separate double stud construction assuming that flanking paths have been controlled.

10.3 DIRECT FIELD SOUND TRANSMISSION

Direct Field Sources

We previously examined the transmission of sound through partitions for a diffuse or reverberant source field. For a direct source field the behavior of the transmission loss is somewhat different. A direct field consists of a plane or nearly plane wave that proceeds unimpeded from the source to the transmitting surface. The energy density and thus the relationship between the sound pressure levels and the sound intensity levels for a plane wave differs by 6 dB from the relationship for a diffuse field. The transmission loss of a surface is also dependent on the angle of incidence, and this must be taken into account in any comprehensive theory.

In a plane wave the power transmitted through a surface is simply related to the intensity incident on the surface

$$W = I\, S \cos\theta\ \tau(\theta) \tag{10.14}$$

where W = power transmitted through a surface (W)
I = direct field intensity incident on the surface(W/m^2)
S = area of the surface(m^2)
θ = angle of incidence with the normal to the surface (rad)
$\tau(\theta)$ = transmissivity of the surface for angle θ

For an exposed surface and an interior observer Eq. 10.14 can be inserted into Eq. 8.87 to obtain

$$L_r = L_s - \Delta L_{TL}(\theta) + 10 \log (4 \cos\theta) + 10 \log\left[\frac{S_w\, Q}{16\,\pi\left[z + \sqrt{\frac{S_w\, Q}{4\,\pi}}\right]^2} + \frac{S_w}{R_r}\right] \tag{10.15}$$

where L_s = direct field sound pressure level near, but in the absence of, reflections from the transmitting surface (dB)
L_r = direct plus reverberant field sound pressure level in the receiving space (dB)
$\Delta L_{TL}(\theta)$ = direct field transmission loss of a partition for a given angle θ, (dB)

Let us define a receiver correction C, such that

$$C = 10 \log \left[\frac{S_w \, Q}{16 \, \pi \left[z + \sqrt{\frac{S_w \, Q}{4 \, \pi}} \right]^2} + \frac{S_w}{R_r} \right] \tag{10.16}$$

Then for normal incidence

$$L_r = L_s - \Delta L_{TL}(\theta = 0) + C + 6 \tag{10.17}$$

This is the same form as Eq. A3 in ASTM Standard E336 for normal incidence sound transmission loss. Note that if $z = 0$ and $\Delta L_{TL}(\theta = 0)$ at the center of an open window, then $L_r = L_s$. This is the correct result since the transmission loss of large openings for plane waves is zero.

Direct Field Transmission Loss

Transmission loss measurements are conducted in two highly reverberant laboratory test rooms. On the source side, by control of the absorption in the room and the number and orientation of the loudspeakers, a diffuse (reverberant) field is achieved at the test partition. Under these conditions, Eq. 10.6 holds and defines the diffuse field transmission loss. The bulk of the transmission loss data are measured in this manner. There is some difficulty in applying these data to direct-field calculations, since there is no specific angular dependence in the laboratory data. We can return to the fundamental mass law relationship given in Eq. 9.18

$$\Delta L_{TL}(\theta) = 10 \log \left[1 + \left(\frac{\omega \, m_s \cos \theta}{2 \, \rho_0 \, c_0} \right)^2 \right] \tag{10.18}$$

where $\Delta L_{TL}(\theta)$ = direct field transmission loss of a partition for a given angle θ, (dB)
ω = radial frequency (rad / s)
ρ_s = surface mass density (kg / m^2)
ρ_0 = density of air (kg / m^3)
c_0 = velocity of sound in air (m / s^2)

In Chapt. 9 this equation was integrated over values of θ between 0° and about 78° to obtain agreement with the measured results. The laboratory transmission loss data are found to be some 5 dB below the $\Delta L_{TL}(\theta = 0)$ data (Ver and Holmer, 1971). For this treatment we assume that the angular dependence of the transmission loss is given by Eq. 10.18. This does not preclude the use of actual measured transmission loss data, but only means that this angular dependence is assumed. We also assume that the density of most walls is large so that $\left(\omega \, m_s \cos \theta / 2 \, \rho_0 c_0\right)^2 >> 1$ for angles less than 78°. Under these conditions the angular dependence can be written as

$$\Delta L_{TL}(\theta) \cong \Delta L_{TL}(\theta = 0) + 20 \log (\cos \theta) \tag{10.19}$$

and substituting in Eq. 9.22

$$\Delta L_{TL}(\theta) \cong \Delta L_{TL} + 5 + 20 \log (\cos \theta) \tag{10.20}$$

Note that while this equation is used in subsequent calculations, if the transmission loss is zero, then we must revert to Eq. 10.15 to obtain accurate results.

Using these components we can assess the sound transmission due to an exterior plane wave passing through the structure of a building.

$$L_r = L_s - \Delta L_{TL} - \Delta L_{SH} + C + G \tag{10.21}$$

where ΔL_{SH} = correction for self shielding, (dB)
G = geometrical factor, which includes the orientation of the source relative to the transmitting surface, (dB)
$G = 10 \log (4 \cos \theta) + [-5 - 20 \log (\cos \theta)]$
$= 10 \log(1.26/\cos \theta) = 1 - 10 \log (\cos \theta)$

Equation 10.21 includes the diffuse field transmission loss measured in a laboratory with the angular behavior included in the G term. The other terms are defined in Eq. 10. 15.

Free Field—Normal Incidence

When a plane wave is normally incident on a transmitting surface,

$$L_r = L_s - \Delta L_{TL} + C + 1 \tag{10.22}$$

Free Field—Non-normal Incidence

For angles of incidence other than zero the value of G is shown in Table 10.1.

Line Source—Exposed Surface Parallel to It

The line source G factor for an exposed surface, whose normal is perpendicular to the line source, can be determined by energy averaging the G term over all values of θ. The G factors at 0° and 80° are single counted, and all others are double counted. The result is

$$G = 3.6 - 10 \log (\cos \phi) \tag{10.23}$$

where ϕ is the angle between the normal to the transmitting surface and the normal to the line source that intersects the center of the surface. The geometry is shown in Fig. 10.12.

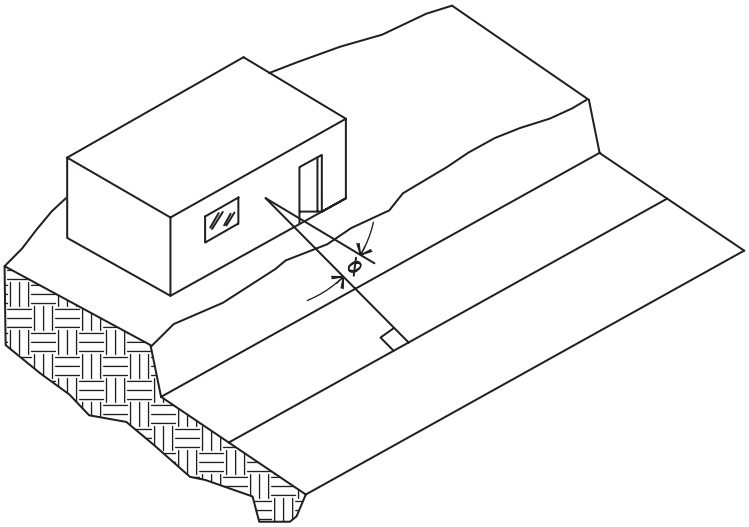
Since the transmission loss is highest when the sound is normally incident, there is often an increase in noise level in high rise buildings with height of the floor above the

TABLE 10.1 Geometrical (G) Factor

	Angle θ, Degrees								
	0	10	20	30	40	50	60	70	80
G (dB)	1.0	1.1	1.3	1.6	2.2	2.9	4.0	5.7	8.6

It is common practice not to include angles above 78°.

FIGURE 10.12 Angle between a Plane and a Line Source



street. As one goes from floor to floor the distance from the street increases so the noise level decreases. The G factor, however, increases as well since the angle ϕ is increasing. This effect offsets the distance loss. The result is that frequently, the loudest interior sound levels occur, not on the first floor, but on about the third floor above street level.

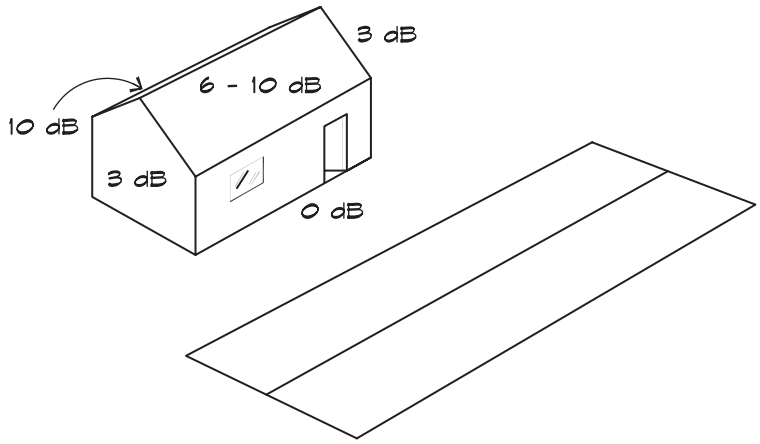
Self Shielding and G Factor Corrections

When a building element is exposed to sound emanating from a point source, the interior level may be calculated using the G factors given in Table 10.1. If the source is shielded by the side of a building it will be attenuated by an amount that can be calculated from the barrier loss relationships previously discussed.

In the case of a line source, where the transmitting surface is a side wall, parallel to the normal, the G factor is theoretically the same as for a wall perpendicular to the normal since for a line source equal energy is radiated from equal angular segments. There is a difference in the self shielding factor, which arises from the fact that the building cuts off half of the line source as seen by the side wall. Figure 10.13 shows ground level self shielding factors for various surface orientations. Both the self shielding and the changes in the G factor are most conveniently subsumed into the self shielding correction.

The difficulty in accurately assessing the geometrical and self shielding corrections for all site configurations is apparent. For odd orientations relative to a line source there is always a tradeoff between the two. For practical calculations shielding is more important than orientation, but it can be influenced by reflections from other structures. If the primary transmitting surface is not facing the roadway, but is within 30° or so, it makes little difference in the G factor while making about a dB difference in the shielding. In general, changes in the two factors due to surface orientation offset one another. For aircraft and other elevated sources, roofs are given a zero shielding factor. Side walls facing the direction of takeoff are considered unshielded, but walls on the approach side are given a 3 dB shielding factor. Surfaces on the side opposite the line of travel are given a 10 dB shielding factor so long as there is no significant sound reflection from nearby structures.

FIGURE 10.13 Line Source Self Shielding Factors



10.4 EXTERIOR TO INTERIOR NOISE TRANSMISSION

As was the case for room to room transmission loss, exterior to interior noise transmission depends on the weakest link in the chain, which in most cases is either the windows or the doors. Where a site is located in a noisy area and a quiet interior noise environment is desired, windows and doors that have a high transmission loss values are critical. Unless exterior levels are quite high, standard California building practices, including stucco exterior walls on wood studs with R-11 (3 1/2" or 90 mm) batt insulation, and 5/8" interior drywall, are adequate to obtain STC ratings that exceed those available in heavy double paned glass windows by a large margin. Thus the doors, both wood and glass, and windows are the main transmission path.

Exterior Walls

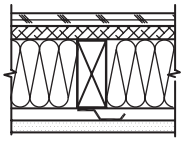
The sound transmission characteristics of several types of exterior walls have been measured by the National Bureau of Standards (Sabine et al., 1975) and are summarized in Fig. 10.14. Where the exterior surface is a lightweight material such as wood or aluminum siding, thin sheet metal or skim coat plaster over Styrofoam, a layer of 5/8" plywood against the stud is usually necessary to bring the mass up to satisfactory levels. It can be seen from Fig. 10.14 that most windows and doors have STC ratings that fall well below the ratings of the commonly used exterior walls. Thus it is necessary to use resilient mounts or separate stud construction on exterior walls only when there are no windows or doors on the wall or where the ratings of these penetrating elements are higher than standard construction will produce.

Windows

Sound transmission through windows depends on the intrinsic rating of the glazing itself and on the treatment of cracks or openings in the window frame. For single paned sealed glazing, the STC ratings are primarily dependent on the thickness of the glass and somewhat dependent on damping provided by a sandwiched interlayer. Figure 10.15 gives transmission loss ratings of various thicknesses of fixed glazing.

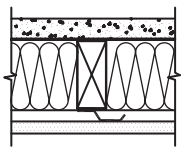
Laminated glazing can provide improved transmission loss performance, especially around the critical frequency. Recognize that although a thinner sheet of laminated glass

FIGURE 10.14 Transmission Loss of Exterior Walls (National Bureau of Standards, 1975)



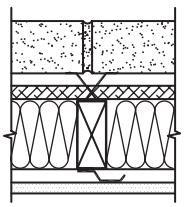
Single Wood Stud Wall - 2 x 4 Stud
1 Layer 5/8" Redwood Siding
3/4" Insulation Board Sheathing
1 Layer 5/8" Drywall
A - Without Insulation B - R-11 Batt Insulation
C - No Insulation - With Resilient Channel
D - R-11 Batt Insulation - With Resilient Channel

	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
A	16	16	22	26	33	35	36	39	43	46	50	51	52	51	48	49	37
B	18	18	28	30	33	34	40	42	45	47	50	50	51	50	50	53	39
C	24	22	26	32	35	38	43	45	50	51	56	57	58	56	54	57	43
D	26	27	30	35	41	43	49	52	56	58	59	60	61	60	58	60	47



Single Wood Stud Wall - 2 x 4 Stud
1 Layer 7/8" Stucco
1 Layer 5/8" Drywall
A - R-11 Batt Insulation
B - No Insulation - With Resilient Channel
C - R-11 Batt Insulation - With Resilient Channel

	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
A	25	30	42	41	44	43	45	45	46	45	46	48	50	50	50	55	46
B	28	32	34	37	41	46	50	50	50	51	54	55	57	55	55	58	49
C	35	41	50	49	53	55	58	58	58	59	59	60	58	57	60	64	57



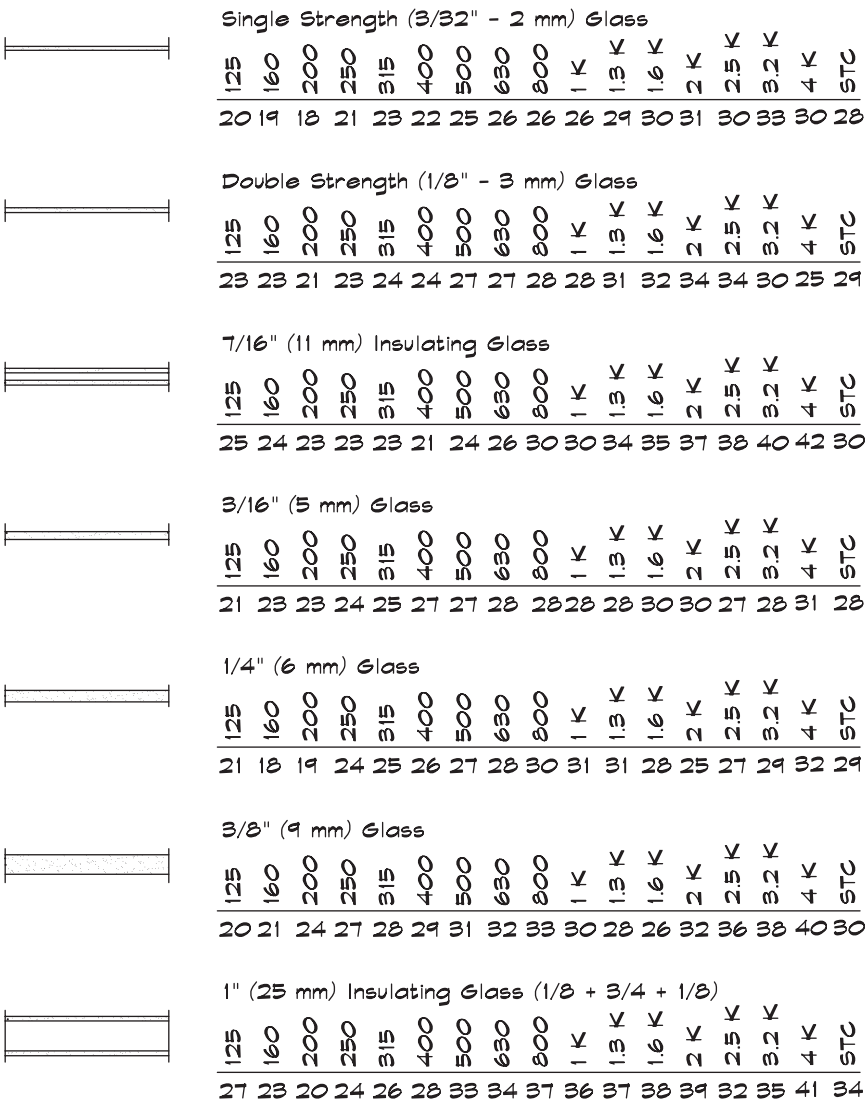
Red Brick Wire Tied to 2 X 4 Studs
1/2" Airspace
3/4" Insulation Board Sheathing
1/2" Drywall
A - R-11 Batt Insulation - No Resilient Channel
B - No Insulation - With Resilient Channel
C - R-11 Batt Insulation - With Resilient Channel

	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	4	STC
A	39	36	42	46	49	51	54	56	58	61	67	67	69	69	70	73	56
B	33	34	41	41	47	50	52	55	59	61	65	66	68	68	69	72	54
C	38	39	46	47	52	54	57	58	60	61	69	68	71	71	72	74	58

may have a higher STC rating, it may be less effective than a heavier sheet of plate glass at low frequencies. The selected product should be based on the actual noise spectrum and the transmission losses in all bands. Transmission loss values for several thicknesses of laminated glass are shown in Fig. 10.16.

When sealed insulating glass is used, the STC rating depends both on the thickness of the glass and the interior air space thickness. Double wall transmission loss theory predicts that a double panel system has a higher transmission loss than a single panel of the same surface weight. This only occurs above the mass-air-mass resonant frequency, which is determined by the weight of each layer and the separation. At or near the resonant frequency the transmission loss of a double panel system is lower than that of a single panel. Even above

FIGURE 10.15 Transmission Loss of Window Glass (National Bureau of Standards, 1975)



this frequency the improvement is limited by mechanical coupling between the two sides. In general it is not effective to use thin double paned windows with air spaces of less than about 3/4” (19 mm) for noise control. Typical transmission loss data are given in Fig. 10.17 for these types of windows.

If window glass is installed in an operable frame there can be a significant degradation in the transmission loss performance due to leakage of air through the seals as well as direct transmission through the frame itself. If we examine the performance of single strength (3/32” or 2.4 mm) glass in various types of frames we find (NBS, 1975) the results shown in Table 10.2. In the case of an aluminum sliding frame, there is a drop of 4 to 5 STC points from the sealed condition.

FIGURE 10.16 Transmission Loss of Laminated Glass



3/16" (5 mm) Safety Glass

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
23	24	25	26	27	28	30	30	31	32	34	34	31	27	30	34	31



1/4" (6 mm) Laminated Glass

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
27	25	26	28	30	30	33	33	33	33	34	35	36	36	38	41	34



3/8" (9 mm) Laminated Glass

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
29	30	31	34	32	33	35	35	35	35	34	35	34	38	42	45	36



1/2" (13mm) Laminated Glass

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
29	30	31	34	33	34	35	35	35	35	33	35	39	42	45	46	37



5/8" (16 mm) Laminated Glass

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
29	30	31	36	35	36	35	35	35	35	37	40	43	45	47	49	38



3/4" (9 mm) Laminated Glass

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
35	33	33	34	36	36	37	36	35	34	39	41	46	48	50	52	39



3/4" (9 mm) Laminated Glass (3 Layer)

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
33	30	33	36	38	37	39	38	37	37	39	42	44	47	49	53	41

Similar behavior is given in Table 10.3 for insulating glass in various types of frames (NBS, 1975).

When operable frames are part of the window assembly the losses are typically lower than those of the glass alone and can be manufacturer dependent. In critical locations laboratory test data should be obtained from a prospective window supplier and calculations performed using these data.

Doors

Like windows, exterior and interior doors are a major source of sound leakage in critical applications. Unlike windows, doors are frequently opened and closed and it is the gaps at the joints and at the threshold that present the greatest problem in controlling noise. The standard exterior door thickness in the United States is 1 3/4” (44 mm), and a solid core

FIGURE 10.17 Transmission Loss of Double Glazed Windows

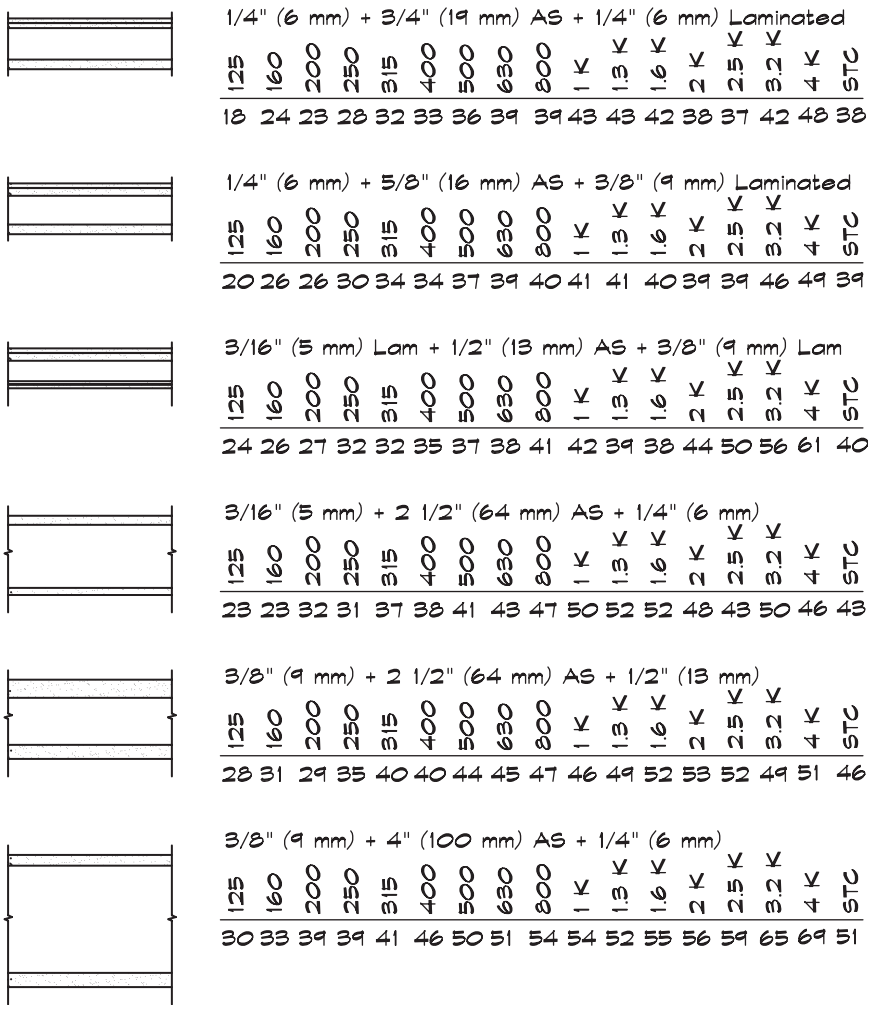


TABLE 10.2 STC Ratings of Single Strength Glass in Various Frames

Frame Configuration	STC Rating
Sealed (Average of 5 Tests)	**28–29**
Wood Double-hung, Locked	**26**
Wood Double-hung, Unlocked	**26**
Aluminum Sliding, Latched	**24**

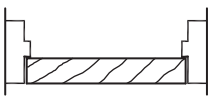
wood door typically weighs about 4–5 lbs/sq ft (20–25 kg/sq m). Based on the mass law one would expect a transmission loss at 500 Hz of about 32 dB for a sealed door. Figure 10.18 shows that this is about what we measure.

In field installations there can be considerable leakage through a door seal at the jamb, head, and threshold. These seals tend to degrade in time due to wear and mechanical failure.

TABLE 10.3 STC Ratings of 7/16" Insulating Glass in Frames

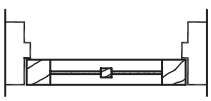
Frame Configuration	STC Rating
Sealed (Average of 2 Tests)	28–30
Wood Double-hung, Locked	26
Wood Double-hung, Unlocked	22
Aluminum Single-hung, Locked	27
Aluminum Single-hung, Unlocked	25

FIGURE 10.18 Transmission Loss of Openable Doors (National Bureau of Standards, 1975)



1 3/4" (45 mm) Hollow Core Door
A - Sealed into Frame
B - Mounted in Frame with Brass Weatherstrip

	125	160	200	250	315	400	500	630	800	1 k	1.3 k	1.6 k	2 k	2.5 k	3.2 k	4 k	STC
A	14	14	14	15	15	17	17	17	15	18	22	22	22	24	27	29	21
B	16	15	15	15	16	17	18	17	15	18	24	24	27	28	30	32	20



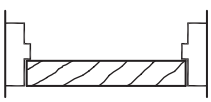
1 3/4" (45 mm) French Door - Single Strength Glass
A - Sealed into Frame
B - Mounted in Frame with Brass Weatherstrip

	125	160	200	250	315	400	500	630	800	1 k	1.3 k	1.6 k	2 k	2.5 k	3.2 k	4 k	STC
A	23	20	24	23	24	24	26	29	30	30	32	32	34	35	36	35	31
B	20	20	24	25	23	24	28	29	28	28	25	25	23	25	27	28	26



1 3/4" (45 mm) Hollow Metal Door
A - Sealed into Frame
B - Mounted in Frame with Magnetic Weatherstrip

	125	160	200	250	315	400	500	630	800	1 k	1.3 k	1.6 k	2 k	2.5 k	3.2 k	4 k	STC
A	22	21	27	28	28	29	30	31	32	33	33	27	24	34	35	37	28
B	23	21	25	26	26	27	28	29	31	31	32	27	24	32	36	39	28



1 3/4" (45 mm) Solid Core Door
A - Sealed into Frame
B - Mounted in Frame with Brass Weatherstrip

	125	160	200	250	315	400	500	630	800	1 k	1.3 k	1.6 k	2 k	2.5 k	3.2 k	4 k	STC
A	14	14	14	15	15	17	17	17	15	18	22	22	22	24	27	29	30
B	21	24	27	27	27	27	30	30	28	26	25	25	25	27	28	29	27

The most common types of threshold in residential doors are a bulb seal, brass v-shaped strips, and a brush seal. Of these the bulb seal is probably the most effective. All weather stripping in order to be effective must seal against a solid threshold of wood, metal, or smooth concrete or vinyl tile. Carpet is ineffective since the sound passes through it under the door.

For the head and jamb, weather stripping is commercially available as foam tape, bulb, or neoprene seals. Steel door frames are also available with a bulb seal built into the frame. This type of device is very effective since it gives the bulb an area to move into when the door is closed. Seals that are located between the door and the jamb can become crushed over time and lose their effectiveness. Note that all seals must be used in compression, rather than in shear if they are to perform effectively.

In moderately critical applications such as a private office, drop closures can be used. These are mechanical devices that are spring loaded and drop down when a latch pin is activated by the closure of the door. They may be mortised into the bottom of the door or surface applied. When they are mortised the appearance of the door is more pleasing but they are more difficult to adjust and maintain. Over time drop closures can malfunction and leave a gap under the door so that periodic maintenance and adjustment is required.

Commercially available sound rated doors are the most effective choice in highly critical applications. STC ratings from 45 to 53 are available in steel doors and from about 40 to 49 in wood doors. The most effective seals are made using a cam lift hinge that lifts the door as it is opened. The bottom of the door incorporates a piece of hard rubber with no moving parts to go out of adjustment. These doors are provided with custom steel frames and adjustable head and jamb seals to close off these paths.

Some sound rated doors are available with flexible magnetic strips that are attracted to the metal surface. These require less maintenance than compression seals and do not cause the door to warp in time. A compression seal requires constant pressure to maintain closure. Most of the force is provided by the latch at the door knob near the center of the panel. In time the top and bottom corners can be pushed out by the force of the seals, which can cause leakage. Since the magnetic seals do not depend on a constant compressive force there is no pressure on the corners.

Where there is a pair of doors in an opening, one of the leaves should be fixed and held into place with a sliding bolt at the top and bottom. At the center the doors should overlap with a dadoed joint or a separate astrigal so that the two leaves do not have a butt joint, which is difficult to seal.

Other transmission paths in doors include louvered openings, undercut thresholds, and lightweight vision panels. Return air paths under or through doors generally preclude effective sound isolation. When these paths are closed off an alternate route for the return air flow must be provided. When vision panels are included in sound rated doors, they require a transmission loss equivalent to that of the door itself.

Electrical Boxes

In many buildings flanking paths between rooms occur through electrical boxes. Nightingale and Quirt (1998) have investigated the phenomenon in gypsum board walls in some detail. They tested boxes in various locations built into a double stud double drywall wall, as illustrated in Fig. 10.19.

FIGURE 10.19 Electrical Box Test Configuration (Nightingale and Quirt, 1998)

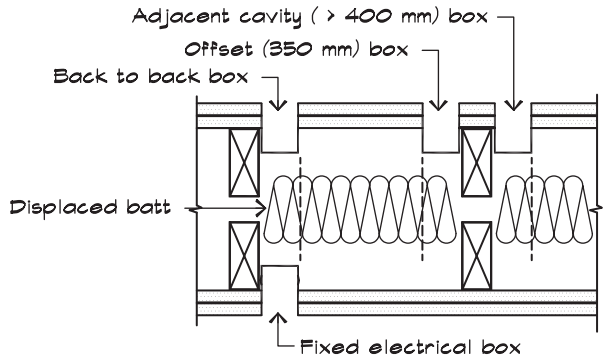


TABLE 10.4 STC Ratings of Walls with Electrical Boxes in Various Locations (Nightingale and Quirt, 1998)

Untreated Metal Boxes			Electrical Box Location		
Wood stud framing	Cavity absorption	Reference case	Back to back no offset	Same cavity offset 350 mm	Adjacent cavity
Double	None	55	51	49	53
	90 mm displaced	61	55	60	61
	90 mm	62	61	61	61
Single	90 mm displaced	55	50	54	54

The technique they used was to construct the boxes, as shown in Fig. 10.19, and to cover the unused ones with two sheets of drywall. The wall was tested with selected boxes exposed. Three insulation configurations were used: 1) no insulation, 2) insulation displaced around the box, and 3) insulation filling the cavities. The test results are summarized in Table 10.4. When the boxes are offset by a stud space (> 400 mm) the ratings are virtually unchanged, particularly with insulation in the cavity.

Figure 10.20 shows the transmission loss data for double stud walls with back-to-back boxes. At low frequencies the transmission loss through the box is high enough, due to the impedance mismatch, that there is little effect. At mid and high frequencies the flanking transmission becomes apparent through the boxes both with and without insulation.

Transmission through the boxes can be blocked by adding a drywall baffle to the inside face of the studs on the box side. Baffles in this research covered one stud cavity, extending from the sole plate to 300 mm (1 ft) above the top of the electrical box. The results are summarized in Table 10.5 and Fig. 10.21. The baffle solution is quite effective and simpler to construct than wrapping the box with drywall. Blocking the back of the box with mastic was shown in this study to be less effective than a baffle. The sides of the box still need to be caulked at the penetration through the drywall surface.

FIGURE 10.20 Measured Transmission Loss of a Double Wood Stud Wall with Cavity Absorption Displaced around the Back-to-Back Metal Boxes (Nightingale and Quirt, 1998)

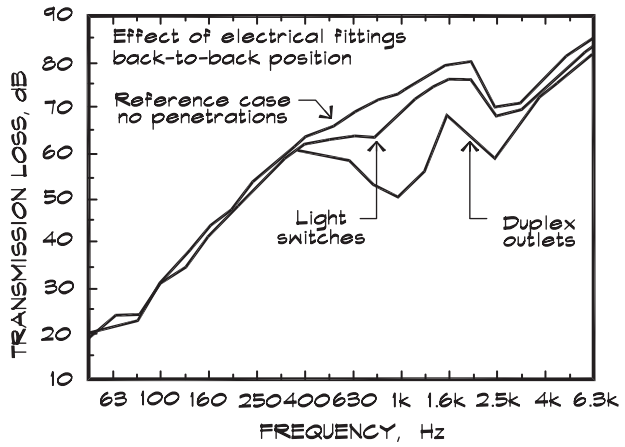
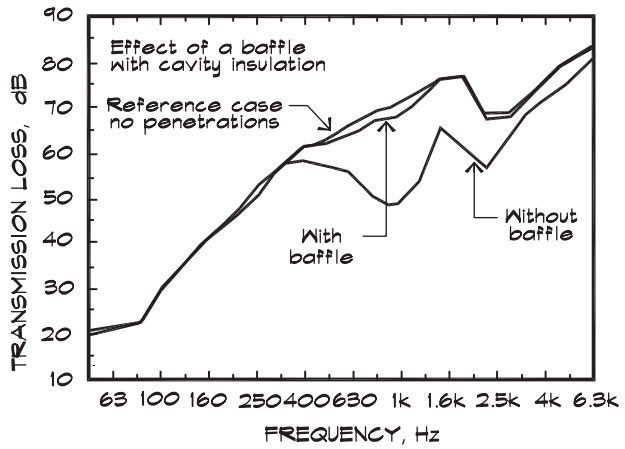


TABLE 10.5 STC Ratings of Walls with Electrical Boxes with Baffles (Nightingale and Quirt, 1998)

	Untreated Metal Boxes		Electrical Box Location		
Cavity Absorption	Reference Case	Back to back No treatment	Same cavity offset 350 mm Baffle	No treatment	Baffle
None	55	51	52	49	52
90 mm displaced	61	55	62	60	61

FIGURE 10.21 Effect of a Baffle Separating Back-to-Back Electrical Boxes in a Double Wood Stud Wall that Has 90 mm Glass Fiber Cavity Absorption (Nightingale and Quirt, 1998)



Aircraft Noise Isolation

With structures located in high noise level areas adjacent to major commercial or military airports, particular care must be exercised to insure a comfortable interior noise environment. In a study at LAX (Long, 1980) Ldn levels were near 80 dBA and the maximum allowable level was an Ldn 45 in the bedrooms. This required an A-weighted noise reduction of 35 dB. A 2 dB safety factor was included, which meant that the design was carried out based on a noise reduction of 37 dB. In homes in this area STC 38 double glazed windows were used along with heavy solid core doors, which were shielded by the structure and a roof overhang and alcove.

For control of aircraft noise the roof is the most critical parameter. Ceiling roofs are generally the largest exposed area, the most complicated structure, and acoustically the least well known. If the roof is a concrete slab or steel deck with a lightweight concrete fill, the problem of sufficient mass usually is ameliorated.

In wood structures, roofs must be solid sheeted with plywood and coverings added to increase the mass to the design level. An inexpensive way of increasing the roof mass is by using layers of 90 lb (0.9 lb/sq ft or 4.4 kg/sq m) felt roofing paper with a cap sheet and shingles or built-up roofing over it. With gravel, concrete tile, or mission tile roofs, the weight is significantly increased and additional layers of roofing paper are not required.

Estimation of the transmission loss of roofs is particularly difficult since there is no directly measured data for peaked roofs and no single separation distance. Flat roof data can be measured in a laboratory or approximated using floor-ceiling data. In the LAX study (Long, 1980), roof transmission losses were estimated using the mass law value of the heavier of the roof or ceiling panel plus two-thirds of the mass law value of the lighter panel. All roofs had solid plywood sheathing with wood shingles over. Eave vents were baffled with lined sheet-metal elbows.

Blocking, where the roofs meet the outside wall, is particularly difficult to control. In plaster homes the most practical solution is to stucco under the eaves to avoid having to caulk the blocking. Since attics must be ventilated, openings are required that must be acoustically treated—usually with a lined sheet metal duct having at least one 90° bend, located in the gable end.

Ceilings are one to two layers of gypsum board. In flat roofs resilient channel can be helpful. Where an open beam look is desired the ceiling can span between the beams but this reduces the airspace dimension and increases the length of joint.

Windows are generally heavy double glazed in noisy sites, although 1/4" laminated glass can be used up to about a 30 dB noise reduction. Highly rated French or sliding glass doors are difficult to find, although some manufacturers can provide a separate storm window or door that can be helpful.

HVAC outside air requirements can be met by providing a sheet metal duct with a commercial silencer. Where bathrooms require an exhaust fan, it too must have a treated duct with either a silencer or an appropriate length of lined duct. For noise reductions on the order of 30 dB, an 8' length of nonmetallic flex duct nested in a fiberglass-filled cavity between two joists will usually provide sufficient loss.

Traffic Noise Isolation

Control of interior noise levels from traffic is much the same as with aircraft noise. The major difference is that, when residences are located above the roadways, ceiling-roofs play a less significant part and windows a more significant part in the overall transmission path. Roofs or

patios that overhang a window or sliding glass door can reflect the sound down toward these surfaces and offset shielding that might otherwise have reduced the exterior sound pressure level. In areas of significant truck traffic, exterior windows should be heavy single glazed or double glazed with a wide airspace between panes. Trucks generate significant energy in the 125 and 250 Hz octave bands so the mass-air-mass resonance should be positioned below these bands. Where barrier shielding is present it is important to remember to use the shielded noise spectrum, which will contain a greater contribution from the lower bands than the unshielded spectrum.

This page intentionally left blank

11

VIBRATION AND VIBRATION ISOLATION

11.1 SIMPLE HARMONIC MOTION

Units of Vibration

In most vibration problems we are dealing with harmonic motion, where the quantities can be expressed as sine or cosine functions. The general formula for the harmonic displacement of a body is given by

$$x = X \sin \omega t \tag{11.1}$$

The velocity can be calculated by differentiating the displacement with respect to time

$$\dot{x} = \frac{d\,x}{d\,t} = X\,\omega \cos \omega t = V \sin (\omega t + \frac{\pi}{2}) \tag{11.2}$$

and the acceleration by differentiating the velocity

$$\ddot{x} = \frac{d\,v}{d\,t} = \frac{d^2\,x}{d\,t^2} = -X\,\omega^2 \sin \omega t = -A \sin (\omega t) \tag{11.3}$$

These lead to simple relationships between the amplitudes

$$A = \omega\, V = \omega^2\, X \tag{11.4}$$

Displacement, velocity, and acceleration are vector quantities that have a fixed angular relationship with each other, as the vector plot in Fig. 11.1 illustrates. Each vector rotates counterclockwise in time about the origin at the radial frequency, ω. Velocity leads displacement by 90° and acceleration leads displacement by 180°.

The units used in vibration measurements are more varied than those for sound level measurements. Amplitudes can be expressed in terms of displacement, velocity, acceleration, and jerk (the rate of change of acceleration). Accelerations are given not only in terms of length per time squared but also in terms of the standard gravitational acceleration, g. The peak amplitudes are simply coefficients such as those shown in Eq. 11.4. The root mean

FIGURE 11.1 Vector Representation of Harmonic Displacement, Velocity, and Acceleration

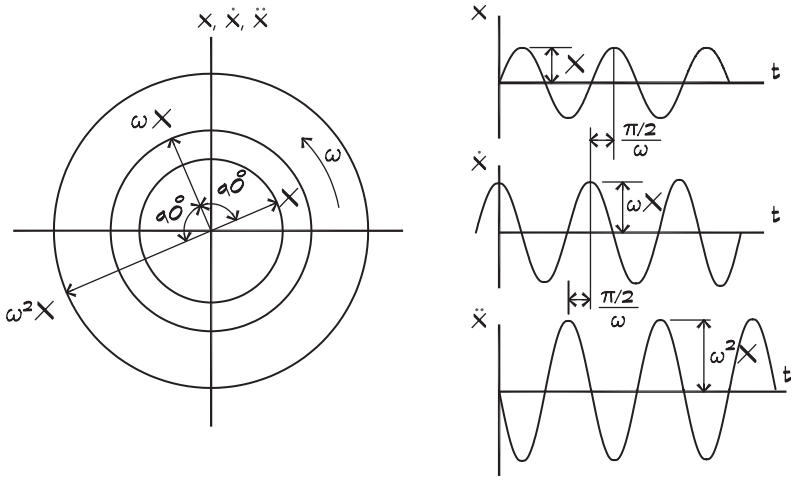


TABLE 11.1 Reference Quantities for Vibration Levels (Beranek and Ver, 1992)

Level (dB)	Formula	Reference (SI)
Acceleration	$L_a = 20 \log (a / a_o)$	$a_o = 10\ \mu m / s^2$
		$a_o = 10^{-5}\ m / s^2$
		$a_o = 1\ g$
		$a_o = 9.8\ m / s^2$
Velocity	$L_v = 20 \log (v / v_o)$	$v_o = 10\ n\ m / s$
		$v_o = 10^{-8}\ m / s$
Displacement	$L_d = 20 \log (d / d_o)$	$d_o = 10\ p\ m$
		$d_o = 10^{-11}\ m$

Note: Decimal multiples are 10^{-1} = deci (d), 10^{-2} = centi (c), 10^{-3} = milli (m), 10^{-6} = micro (μ), 10^{-9} = nano (*n*), and 10^{-12} = pico (*p*).

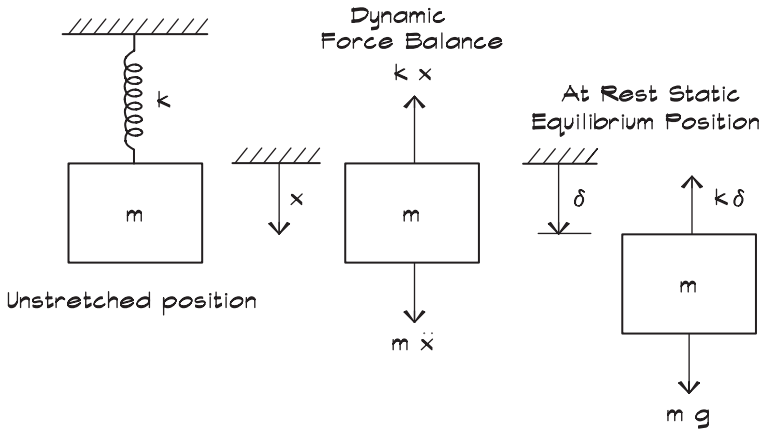
square (rms) value is the square root of the average of the square of a sine wave over a complete cycle, which is $\left(\sqrt{2}\right)^{-1}$ or .707 times the peak amplitude. Vibration amplitudes also can be expressed in decibels and Table 11.1 shows the preferred reference quantities.

11.2 SINGLE DEGREE OF FREEDOM SYSTEMS

Free Oscillators

In its simplest form a vibrating system can be represented as a spring mass, shown in Fig. 11.2. Such a system is said to have a single degree of freedom, since its motion can be described with a knowledge of only one variable, in this case its displacement.

FIGURE 11.2 Free Body Diagram of a Spring Mass System



In general if a system requires n numbers to describe its motion it is said to have n degrees of freedom. A completely free mass has six degrees of freedom: three orthogonal displacement directions and three rotations, one about each axis. A stretched string or a flexible beam has an infinite number of degrees of freedom, since there are an infinite number of possible vibration shapes. These can be analyzed in a regular manner using a superposition of all possible vibrational modes added together; however, to do so exactly requires an infinite number of constants, one for each mode. This mathematical construct, called a Fourier series, is a useful tool even if it is not carried out to infinity.

The forces on a simple spring mass system are the spring force, which depends on the displacement away from the equilibrium position, and the inertial force of the accelerating mass. The equation of motion was discussed in Chapt. 6 and is simply a summation of the forces on the body

$$\mathrm{m}\,\ddot{\mathrm{x}} + k\,\mathrm{x} = 0 \tag{11.5}$$

which has a general solution

$$\mathrm{x} = \mathrm{X}\sin\left(\omega_{\mathrm{n}}\,\mathrm{t} + \phi\right) \tag{11.6}$$

where $\omega_{\mathrm{n}} = \sqrt{k/\mathrm{m}}$ = undamped natural frequency (rad / s)
k = spring constant (N / m)
m = mass (kg)
ϕ = phase angle at time t = 0 (rad)
X = maximum displacement amplitude (m)

Although the spring mass model is simple, it is applicable as an approximation to many complicated structures. Building elements such as beams, wood or concrete floors, high-rise buildings, and towers can be modeled as spring mass systems and in more complex structures as series of connected elements, each having mass and stiffness.

Damped Oscillators

In vibrating systems, when bodies are set into motion, dissipative forces arise that damp or resist the movement. These are viscous forces that are proportional to the velocity of the

body; however, not all types of damping are velocity dependent. Coulomb damping due to sliding friction, for example, is a constant force. To model viscous damping, such as that provided by a shock absorber, we refer to the spring mass system shown in Fig 11.3. Here the damping force is proportional to the velocity and is negative because the force opposes the direction of motion.

$$\mathrm{F_r} = -c\,\dot{\mathrm{x}} \tag{11.7}$$

where $\mathrm{F_r}$ = viscous damping force, (N)

c = resistance damping coefficient (N s / m)

$\dot{\mathrm{x}} = \dfrac{\mathrm{d\,x}}{\mathrm{d\,t}}$ = first time derivative of the displacement

= velocity (m / s)

If we gather together all forces operating on the mass on the left-hand side, and equate it to the mass times the acceleration on the right-hand side in accordance with Newton's law, and rearrange the terms, we get

$$\mathrm{m}\,\ddot{\mathrm{x}} + c\,\dot{\mathrm{x}} + k\,\mathrm{x} = 0 \tag{11.8}$$

The general solution has the form $\mathrm{x} = \mathrm{e}^{\mathrm{a\,t}}$, where a is a constant to be determined. Substituting into Eq. 11.8 we obtain

$$\left(\mathrm{a}^2 + \frac{c}{\mathrm{m}}\,\mathrm{a} + \frac{k}{\mathrm{m}}\right)\mathrm{e}^{\mathrm{a\,t}} = 0 \tag{11.9}$$

which holds for all t when

$$\left(\mathrm{a}^2 + \frac{c}{\mathrm{m}}\,\mathrm{a} + \frac{k}{\mathrm{m}}\right) = 0 \tag{11.10}$$

This equation, known as the characteristic equation, has two roots

$$\mathrm{a}_{1,2} = -\frac{c}{2\,\mathrm{m}} \pm \sqrt{\left(\frac{c}{2\,\mathrm{m}}\right)^2 - \frac{k}{\mathrm{m}}} \tag{11.11}$$

from which we can construct a general steady-state solution in the underdamped condition, where the term under the radical is negative.

$$\mathrm{x} = \mathrm{X}\,\mathrm{e}^{\frac{-c\,\mathrm{t}}{2\,\mathrm{m}}}\,\sin\left(\omega_\mathrm{n}\,\mathrm{t} + \phi\right) \tag{11.12}$$

The damped natural frequency of vibration is given by

$$\omega_\mathrm{d} = 2\pi\,\mathrm{f_d} = \sqrt{\omega_\mathrm{n}^2 - \left(\frac{c}{2\,\mathrm{m}}\right)^2} \tag{11.13}$$

The damping coefficient, c, influences both the amplitude and the damped natural frequency of oscillation, ω_d, by slowing it down slightly.

An example of a damped oscillation is shown in Fig. 11.4. The envelope of the decay is controlled by the damping coefficient. One measure of the degree of damping is the decay

FIGURE 11.3 A Spring Mass System with Viscous Damping (Thomson, 1965)

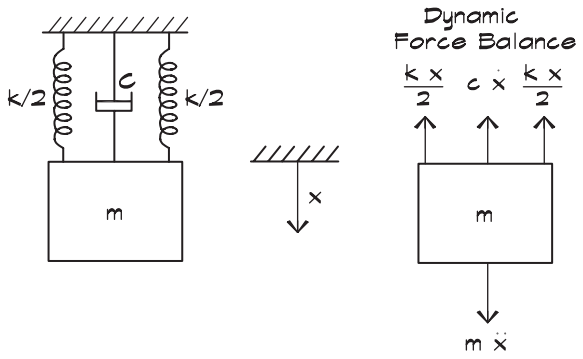
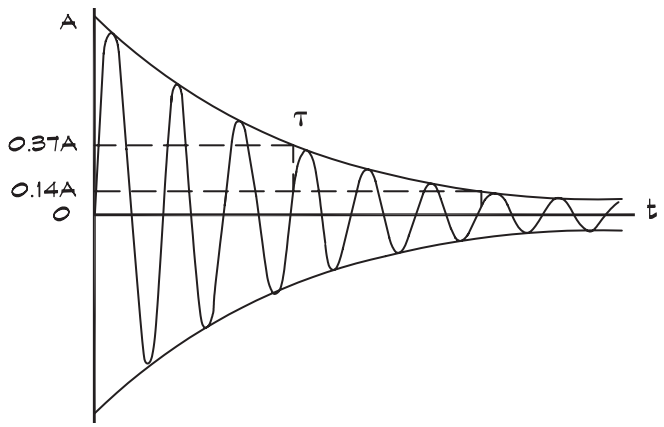


FIGURE 11.4 Response of a Damped Oscillator to an Impulse (Rossing and Fletcher,1995)

When a damped oscillator is struck or plucked it responds in an exponentially decaying envelop which falls to an amplitude of 1/e (37%) of the initial value in a decay time T .



time, $\tau = \dfrac{2\,\mathrm{m}}{\mathrm{c}}$, which is the time it takes for the amplitude of the envelope to fall to 1/e (37%) of its initial value. It can be seen from Eq. 11.13 that, when one over the decay time is equal to the undamped natural frequency, the term under the radical is zero and the system does not oscillate. Such a system is said to be critically damped. The value of the damping coefficient at this point is given the symbol $c_c = 2\,\mathrm{m}\,\omega_\mathrm{n}$, and the degree of damping is expressed in terms of the ratio of the damping coefficient to the critical damping coefficient $\eta = \dfrac{c}{c_c}$, which is called the damping ratio, and is expressed as a percentage of critical damping.

Damping Properties of Materials

All materials have a certain amount of intrinsic internal damping, which depends on the internal structure of the substance. Figure 9.10 showed the damping coefficients for a number of common construction materials, which range from extremely low values in steel and other

metals to very high values in resins and viscous liquids. These latter materials are used in laminated glass specifically for their damping characteristics. In laminated glass a resin is sandwiched between the two layers. This is called a constrained layer damper. Damping compounds are commercially available in bulk and can be trowelled directly onto lightweight metal panels. In order to be effective they should be applied thickly–to at least the thickness of the vibrating panel.

In wood floor systems panel adhesive can help provide damping when applied between sheets of flooring, between wood joists and plywood subfloors, and to stepped blocking installed within the floor joists. In concrete floor systems the thickness and density of the concrete determines the amount of damping. Additional damping can be provided by plates welded to the joist webs and by lightweight interior partitions attached either above or below the floor. Even if partitions are not load bearing, they can contribute significantly to damping.

Driven Oscillators and Resonance

When a spring mass system is driven by a periodic force, it will respond in a predictable manner, which depends on the frequency of the driving force. A familiar example is a child's swing. If a child pumps the swing by kicking his legs out at the proper moment, he can increase the amplitude of the swing oscillation. The swing responds at the frequency of the driving force but its amplitude increases substantially only when the period of the driving force matches the natural period of vibration. Thus the child soon learns that he must kick out his legs at the proper time if he is to increase his swing's height.

There are many examples of resonant systems in architecture, including sound waves in rectangular rooms, organ pipes, and other open or closed tubes; and structural systems including floors, walls and wall panels, piping, and mechanical equipment. Each of these can act as an oscillator and be driven into resonance by a periodic force. The equation describing the motion of a forced oscillator with damping is

$$\mathrm{m}\,\ddot{\mathrm{x}} + c\,\dot{\mathrm{x}} + k\,\mathrm{x} = \mathrm{F}_0 \sin(\omega\,\mathrm{t}) \tag{11.14}$$

The general solution has the form

$$\mathrm{x} = \mathrm{X} \sin(\omega\,\mathrm{t} - \phi) \tag{11.15}$$

By substituting into Eq. 11.14 we obtain

$$\begin{aligned} \mathrm{m}\,\omega^2\,\mathrm{X} \sin(\omega\,\mathrm{t} - \phi) - c\,\omega\,\mathrm{X} \sin\left(\omega\,\mathrm{t} - \phi + \frac{\pi}{2}\right) \\ - k\,\mathrm{X} \sin(\omega\,\mathrm{t} - \phi) + \mathrm{F}_0 \sin(\omega\,\mathrm{t}) = 0 \end{aligned} \tag{11.16}$$

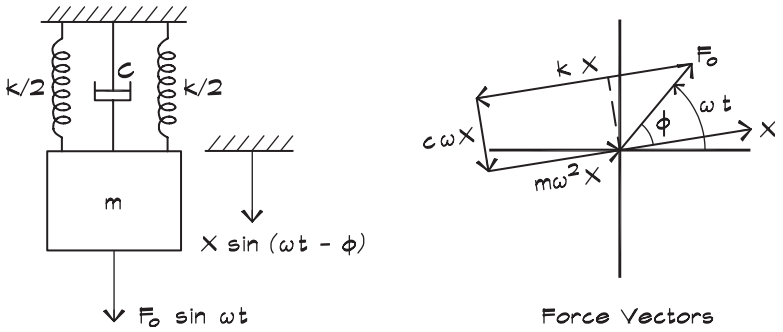
The relationship among all the forces acting on the mass is shown in Fig. 11.5, and from the geometry of the force triangle we can solve for the amplitude X

$$\mathrm{X} = \frac{\mathrm{F}_0}{\sqrt{\left(k - \mathrm{m}\,\omega^2\right)^2 + (c\,\omega)^2}} \tag{11.17}$$

and

$$\tan\phi = \frac{c\,\omega}{k - \mathrm{m}\,\omega^2} \tag{11.18}$$

FIGURE 11.5 Forced Response of a Spring Mass System with Viscous Damping (Thomson, 1965)



We can use more general notation as follows

$\omega_n = \sqrt{k/m}$ = undamped natural frequency (rad / s)
$c_c = 2\ m\ \omega_n$ = critical damping coefficient (N s / m)
$\eta = c/c_c$ = damping factor
$X_0 = F_0 / k$ = static deflection of the spring mass under the steady force F_0 (m)

and write Eq. 11.17 as

$$\frac{X}{X_0} = \frac{1}{\sqrt{\left[1 - (\omega/\omega_n)^2\right]^2 + \left[2\eta\,(\omega/\omega_n)\right]^2}} \tag{11.19}$$

and Eq. 11.18 as

$$\tan\phi = \frac{2\,\eta\,(\omega/\omega_n)}{1 - (\omega/\omega_n)^2} \tag{11.20}$$

Looking at Eqs. 11.15, 11.17, and 11.19 we see that the mass vibrates at the driving frequency ω, but the amplitude of vibration depends on the ratio of the squares of the resonant and driving frequencies. When the driving frequency matches the resonant frequency a maximum in the displacement occurs. Note that the damping term $2\ \eta\ (\omega/\omega_n)$ keeps the denominator from vanishing and limits the excursion at resonance.

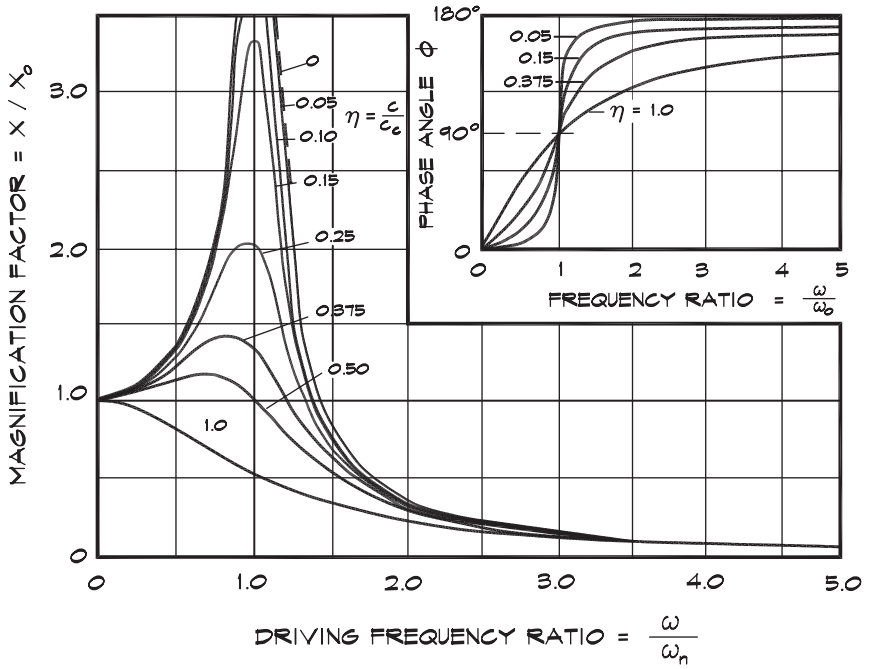
Figure 11.6 shows a plot of the response of the system. As the driving frequency moves toward the resonant frequency the output increases—theoretically reaching infinity at resonance for zero damping. The damping not only limits the maximum excursion at resonance but also shifts the resonant peak downward in frequency.

Vibration Isolation

When a simple harmonic force is applied to a spring mass system, it induces a response that reaches a maximum at the resonant frequency of the system. If we ask what force is transmitted to the foundation through the spring mass support we can refer again to Fig. 11.5.

The forces are transmitted to the support structure through the spring and shock absorber system. The formulas remain the same whether the mass is resting on springs or hung

FIGURE 11.6 Normalized Excursion vs Frequency for a Forced Simple Harmonic System with Damping (Thomson, 1965)



from springs. The balance of dynamic forces is shown, and using this geometry we can resolve the force on the support system as

$$F_t = \sqrt{(k\,X)^2 + (c\,\omega\,X)^2} = X\sqrt{k^2 + c^2\,\omega^2} \tag{11.21}$$

Using the expression given in Eq. 11.19 for the relationship between the applied force and the displacement amplitude, we can solve for the ratio of the impressed and transmitted forces

$$F_t = \frac{F_0\sqrt{1 + \left(\frac{c\,\omega}{k}\right)^2}}{\sqrt{\left[1 - \frac{m\,\omega^2}{k}\right]^2 + \left(\frac{c\,\omega}{k}\right)^2}} \tag{11.22}$$

which can be written as

$$\tau = \frac{F_t}{F_0} = \frac{\sqrt{1 + \left(2\,\eta\,\frac{\omega}{\omega_n}\right)^2}}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left(2\,\eta\,\frac{\omega}{\omega_n}\right)^2}} \tag{11.23}$$

FIGURE 11.7 Transmissibility of a Viscous Damped System
The Force Transmissibility and Motion Transmissibility of a Viscous Damped Single Degree of Freedom are Numerically Identical

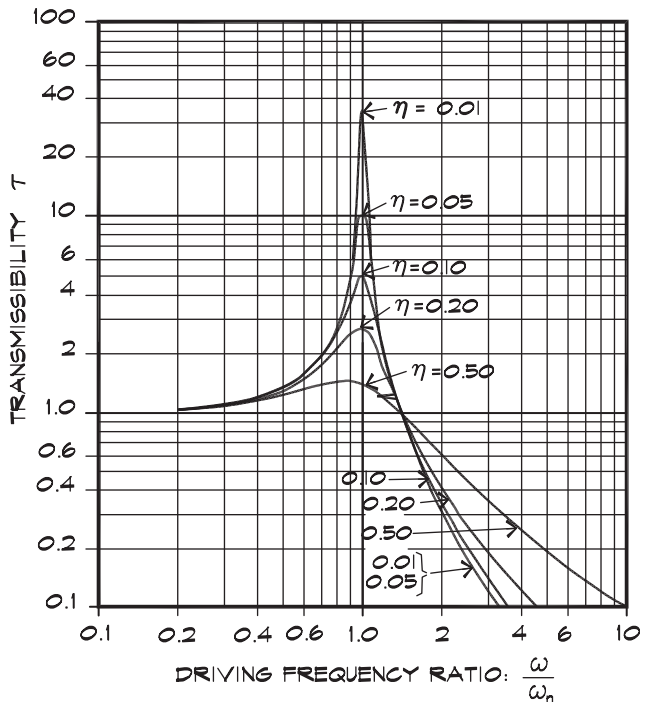


Figure 11.7 shows a plot of this expression in terms of the transmissibility, which is the ratio of the transmitted to the imposed force. We can see that above a given frequency $\left(\sqrt{2}\,\mathrm{f_n}\right)$, as the frequency of the driving force increases, the transmissibility decreases and we achieve a decrease in the transmitted force. This is the fundamental principle behind vibration isolation.

Since the isolation is dependent on frequency ratio, the lower the resonant frequency, the greater the isolation for a given excitation frequency. The natural frequency of the spring mass system is

$$\mathrm{f_n} = \frac{\omega_\mathrm{n}}{2\,\pi} = \frac{1}{2\,\pi}\sqrt{k/\mathrm{m}} = \frac{1}{2\,\pi}\sqrt{k\ \mathrm{g/m\ g}} \tag{11.24}$$

which can be written in terms of the static deflection of the vibration isolator under the weight of the supported object,

$$\mathrm{f_n} = \frac{1}{2\,\pi}\sqrt{g/\delta} = \frac{3.13}{\sqrt{\delta_i}}\ (\mathrm{Hz},\ \delta_i \text{ in inches}) \tag{11.25}$$

or

$$\mathrm{f_n} = \frac{1}{2\,\pi}\sqrt{g/\delta} = \frac{5}{\sqrt{\delta_{cm}}}\ (\mathrm{Hz},\ \delta_{cm} \text{ in centimeters}) \tag{11.26}$$

A fundamental principle for effective isolation is that the greater the deflection of the isolator, the lower the resonant frequency of the spring mass system, and the greater the vibration isolation. We must counterbalance this against the mechanical stability of the isolated object since very soft mounts are generally less stable than stiff ones. To increase the deflection, we must increase the load on each isolator, so a few point-mount isolators are preferable to a continuous mat or sheet. Thick isolators are generally more effective than thin isolators since thick isolators can deflect more than thin ones. Finally, trapped air spaces under isolated objects should be avoided and, if unavoidable, then wide spaces are better than narrow spaces, because the trapped air acts like another spring. Note that the greater the damping, the less the vibration isolation, but the lower the vibration amplitude near resonance. This leads to a second important point, which is that damping is incorporated into vibration isolators, not to increase the isolation, but to limit the amplitude at resonance. An example might be a machine that starts from a standstill (zero frequency), goes through the isolator resonance, and onto its operating point frequency. If this happens slowly we may be willing to trade off isolation efficiency at the eventual operating point for amplitude limitation at resonance.

If there is zero damping Eq. 11.23 can be simplified further. Assuming that the frequency ratio is greater than $\sqrt{2}$, the transmissibility is given by

$$\tau \cong \left[\left(\frac{\omega}{\omega_n} \right)^2 - 1 \right]^{-1} \tag{11.27}$$

We substitute $\omega_n^2 = g/\delta$, where g is the acceleration due to gravity and δ is the static deflection of the spring under the load of the supported mass, and the transmissibility becomes

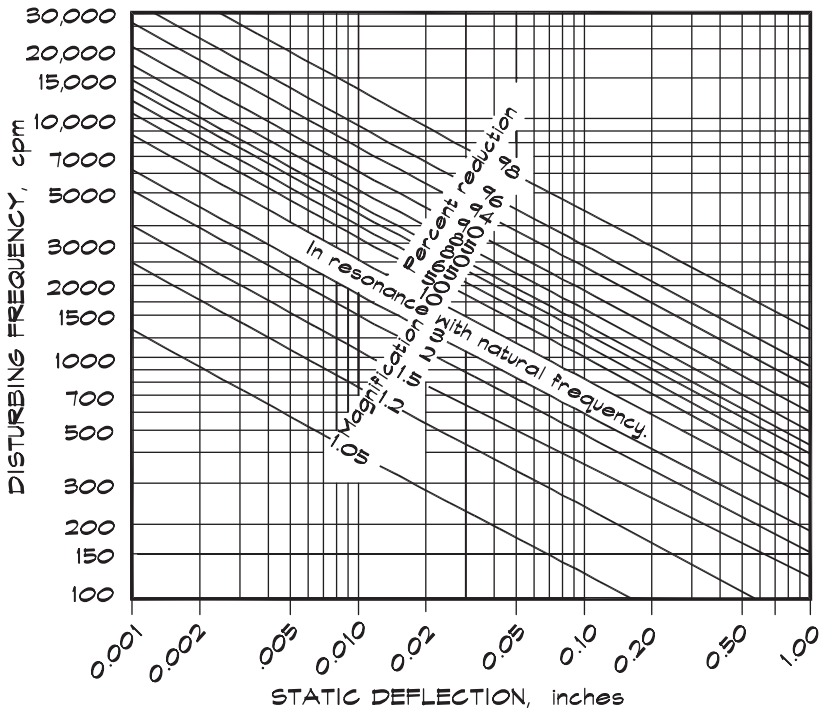
$$\tau \cong \left[\frac{(2\,\pi\,\mathrm{f})^2\,\delta}{\mathrm{g}} - 1 \right]^{-1} \tag{11.28}$$

which is sometimes expressed as an isolation efficiency or percent reduction in vibration in Fig. 11.8. This simplification is occasionally encountered in vibration isolation specifications that call for a given percentage of isolation at the operating point. It is better to specify the degree of isolation indirectly by calling out the deflection of the isolator, which is directly measurable by the installing contractor, rather than an efficiency that is abstract and difficult to measure in the field.

It is important to recall that these simple relationships only hold for single degree of freedom systems. If we are talking about a piece of mechanical equipment located on a slab the deflection of the slab under the weight of the isolated equipment must be very low—typically 8 to 10 times less than the deflection of the isolator for this approximation to hold. As the stiffness of the slab decreases, softer vibration isolators must be used to compensate.

When the excitation force is applied directly to the supported object or when it is self excited through eccentric motion, vibration isolators do not decrease the amplitude of the driven object but only the forces transmitted to the support system. When the supported object is excited by the motion of the support base, there is a similar reduction in the forces transmitted to the object. For a given directly applied excitation force, an inertial base consisting of a large mass, such as a concrete slab placed between the vibrating equipment and the support system, can decrease the amplitude of the supported equipment, but interestingly

FIGURE 11.8 Isolation Efficiency for a Flexible Mount



not the amplitude of the transmitted force. Inertial bases are very helpful in attenuating the motion of mechanical equipment such as pumps, large compressors, and fans, which can have eccentric loads that are large compared to their intrinsic mass.

Isolation of Sensitive Equipment

Frequently there are requirements to isolate a piece of sensitive equipment from floor-induced vibrations. The geometry is that shown in Fig. 11.9. Since the spring supports are in their linear region the relations are the same for equipment hung from above or supported

FIGURE 11.9 Force Vectors of a Spring Mass System with Viscous Damping for a Moving Support

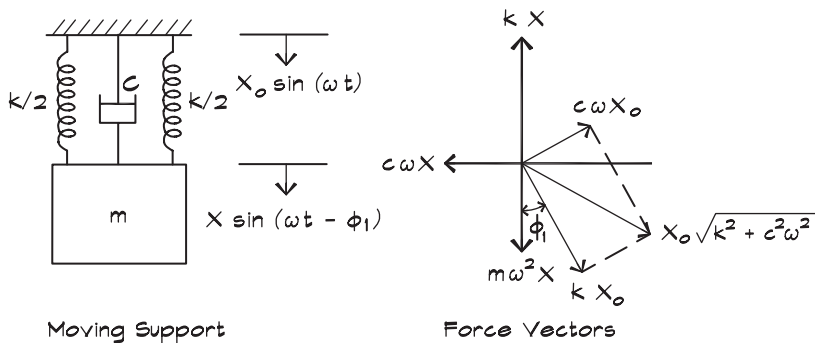
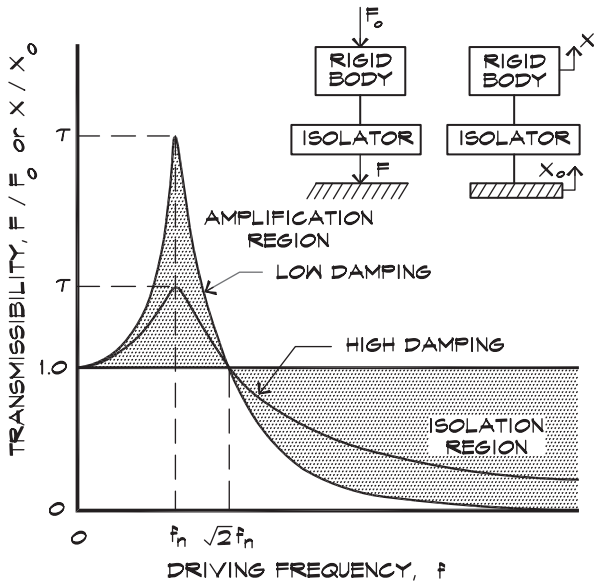


FIGURE 11.10 Transmissibility Curves for Vibration Isolation (Ruzicka, 1971)



from below. The transmissibility is the same as that given in Eq. 11.23. In the case of isolated equipment, instead of the force being generated by a vibrating machine, a displacement is created by the motion of the supporting foundation. In Eq. 11.23 the terms for force amplitudes are replaced by displacement amplitudes.

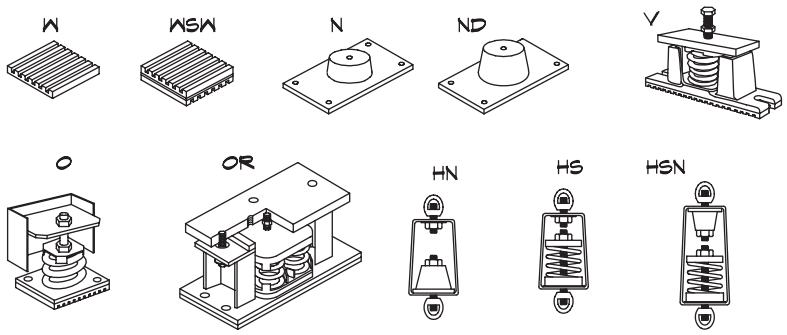
Summary of the Principles of Isolation

Figure 11.10 shows the result of this analysis for both self-excited sources and sensitive receivers. The transmission equation is the same in both cases, differing only in the definition of transmissibility, which for an imposed driving force is the force ratio and for base motion is the displacement ratio. Above the resonant frequency of the spring mass system the response to the driving function decreases until, at a frequency just over 40% above resonance, the response amplitude is less than the imposed amplitude. At higher driving frequencies the response is further decreased. The lower the natural frequency of the isolator—that is, the greater its deflection under the load of the equipment—the greater the isolation.

11.3 VIBRATION ISOLATORS

Commercially available vibration isolators fall into several general categories: resilient pads, neoprene mounts, and a combination of a steel spring and neoprene pad (Fig. 11.11). An isolator is listed by the manufacturer with a range of rated loads and a static deflection, which is the deflection under the maximum rated load. Most isolators will tolerate some loading beyond their rated capacity, often as much as 50%; however, it is good practice to check the published load versus deflection curve to be sure. An isolator must be sufficiently loaded to achieve its rated deflection, but it must also remain in the linear range of the load versus deflection curve and not bottom out.

FIGURE 11.11 Types of Vibration Isolators



Isolation Pads (Type W, WSW)

Isolation pads of felt, cork, neoprene impregnated fiberglass, or ribbed neoprene sometimes sandwiched by steel plates usually have about a .05 inch (1 mm) deflection ($f_n = 14$ Hz) and are used in noncritical or high-frequency applications. Typically these products are supplied in small squares, which are placed under vibrating equipment or piping. Depending on the stiffness of the product, they are designed to be loaded to a particular weight per unit area of pad. For 40 durometer neoprene pads, for example, the usual load recommendation is about 50 lbs/sq in. Where higher deflections are desired or where there is a need to spread the load, pads are sandwiched with thin steel plates. Such pads are designated WSW or WSWSW depending on the number of pads and plates.

Neoprene Mounts (Type N, ND)

Neoprene isolators are available in the form of individual mounts, which have about a 0.25 inch (6 mm) rated deflection, or as double deflection mounts having a 0.4 inch (10 mm) deflection. These products frequently have integral steel plates, sometimes with tapped holes, that allow them to be bolted to walls or floors. They are available in neoprene of various durometers from 30 to 60, and are color-coded for ease of identification in the field. The double deflection isolators can be used to support floating floors in critical applications such as recording studios.

Steel Springs (Type V, O, OR)

A steel spring is the most commonly used vibration isolator for large equipment. Steel springs alone can be effective for low-frequency isolation; however, for broadband isolation they must be used in combination with neoprene pads to stop high frequencies. Otherwise these vibrations will be transmitted down the spring. Springs having up to 5 inches (13 cm) static deflection are available, but it is unusual to see deflections greater than 3 inches (8 cm) due to their lateral instability. Unhoused open-spring mounts (Type O) must have a large enough diameter (at least 0.8 times the compressed height) to provide a lateral stiffness equal to the vertical stiffness. Housed springs have the advantage of providing a stop for lateral (Type V) or vertical motion and an integral support (Type OR) for installing the equipment at or near its eventual height, but are more prone to ground out when improperly positioned. These stops are useful during the installation process since the load of the equipment or piping may vary; particularly if it can be filled with water or oil. Built in limit stops are not

the same as earthquake restraints, which must resist motion in any direction. Threaded rods, allowing the height of the equipment to be adjusted and locked into place with double nuts, are also part of the isolator assembly.

Spring isolators must be loaded sufficiently to produce the design deflection, but not so much that the springs bottom out coil to coil. A properly isolated piece of equipment will move freely if one stands on the base, and should not be shorted out by solid electrical or plumbing connections.

Hanger Isolators (Type HN, HS, HSN)

Hanger isolators contain a flexible element, either neoprene (Type HN) or a steel spring (Type HS), or a combination of the two (Type HSN), which supports equipment from above. Spring hangers, like free standing springs, must have a neoprene pad as part of the assembly. Hangers should allow for some misalignment between the housing and the support rod (30°) without shorting out and be free to rotate 360° without making contact with another object. Threaded height-adjusting rods are usually part of these devices.

Air Mounts (AS)

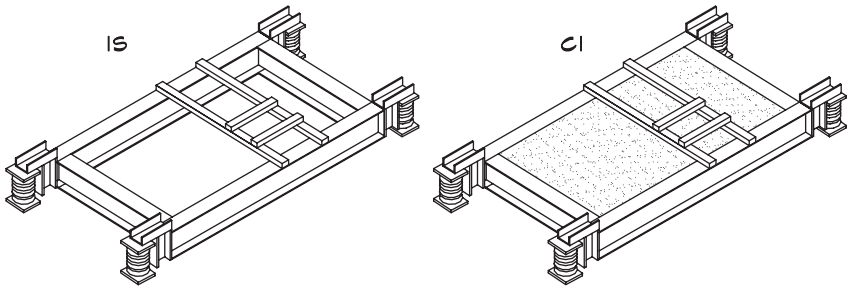
Air springs consisting of a neoprene bladder filled with compressed air are also available. These have the disadvantage of requiring an air source to maintain adequate pressure along with periodic maintenance to assure that there is no leakage. The advantage is that they allow easy level adjustment and can provide larger static deflections than spring isolators for critical applications.

Support Frames (Type IS, CI, R)

Since the lower the natural frequency of vibration the greater the vibration isolation, it is advantageous to maximize the deflection of the isolation system consistent with constraints imposed by stability requirements. If the support system is a neoprene mount—for example, under a vibrating object of a given mass—it is generally best to use the fewest number of isolators possible consistent with other constraints. It is less effective to use a continuous sheet of neoprene, cork, flexible mesh, or other similar material to isolate a piece of equipment or floating floor since the load per unit area and thus the isolator deflection is relatively low. Rather, it is better to space the mounts under the isolated equipment so that the load on each mount is maximized and the lowest possible natural frequency is obtained. A structural frame may have to be used to support the load of the equipment if its internal frame is not sufficient to take a point load. Integral steel (IS) or concrete inertial (CI) or rail frame (R) bases (Fig. 11.12) are used in these cases. A height-saving bracket that lowers the bottom of the frame to 25 to 50 mm (1” to 2”) above the floor is typically part of an IS or CI frame. Brackets allow the frame to be placed on the floor and the equipment mounted to it before the springs are slid into place and adjusted.

When equipment is mounted on isolators the load is more concentrated than with equipment set directly on a floor. The structure beneath the isolators must be capable of supporting the point load and may require a 100 to 150 mm (4” to 6”) housekeeping pad to help spread the load. Equipment such as small packaged air handlers mounted on a lightweight roof can be supported on built up platforms that incorporate a thin (3”) concrete pad. Lighter platforms may be used if they are located directly above heavy structural elements such as steel beams or columns. In all cases the ratio of structural deflection to spring deflection must be less than 1:8 under the equipment load.

FIGURE 11.12 Vibration Isolation Bases



Isolator Selection

A number of manufacturers, as well as ASHRAE, publish recommendations on the selection of vibration isolators. By and large these recommendations assume that the building structure consists of concrete slabs having a given span between columns. One of the most useful is that published by Vibron Ltd. (Allen, 1989). This particular guide is reproduced as Tables 11.2 through 11.4. To use it, first determine the sensitivity of the receiving space, the floor thickness, and span. The longer the span, the more the deflection of the floor, the lower its resonant frequency, and the harder it is to isolate mechanical equipment that it supports. From step one we obtain an isolation category, a number from 1 to 6, which is a measure of the difficulty of successfully isolating the equipment. We then enter the charts in Tables 11.3 or 11.4 and pick out the base type and isolator deflection appropriate to the type of equipment and the isolation category.

When a concrete inertial (type CI) base is required, we can calculate its thickness from the nomographs given in Fig. 11.13. Using such a table is a practical way of selecting an appropriate isolator for a given situation. Although these tabular design methods are simple in practice, there is a great deal of calculating and experience that goes into their creation.

11.4 SUPPORT OF VIBRATING EQUIPMENT

Structural Support

A spring mass system, used to isolate vibrating equipment from its support structure, is based on a theory that assumes that the support system is very stiff. In practice it is important to construct support systems that are stiff, compared to the deflection of the isolators, and to minimize radiation from lightweight diaphragms. Where the support structure is very light—which can be the case for roof-mounted units—mechanical equipment is best supported on a separate system of steel beams that in turn are supported on columns down to a footing. A lightweight roof or similar structure can radiate sound like a driven loudspeaker, so mechanical equipment should not be located directly on lightweight roof panels. Where there is no other choice, and the roof slab is less than 4.5” (11 cm) of concrete, a localized concrete housekeeping pad should be used, having a thickness of 4” (10 cm) to 6” (15 cm) and a length 12” (30 cm) longer and wider than the supported equipment. These pads help spread the load and provide some inertial mass to increase the impedance of the support. Where it is not possible to locate equipment above a column, it should be located over one or more heavy structural members. Where supporting structures are less than 3.5” of solid concrete, use one isolation category above that determined from Table 11.2 along with the concrete subbase.

TABLE 11.2 Vibration Isolation Selection Guide (Vibron, 1989)

LOCATION OF VIBRATION ISOLATION		ISOLATION CATEGORY		
TYPE OF BUILDING	FLOOR SPAN (ft)	BASEMENT BELOW GRADE	UPPER CONCRETE FLOORS - OVER AND ADJACENT TO OCCUPIED AREAS	
			6" THICK & UP	TO 6" THICK
AUDITORIUMS CHURCHES - CRITICAL AREAS HOSPITALS - CRITICAL AREAS MUSIC HALLS OFFICE BUILDINGS - CRITICAL CONDOMINUIMS RADIO AND TV STATIONS	ON GRADE - ALL SPANS THROUGH 20 21 - 30 31 - 40 41 - 50 51 & UP	— 1 1 1 1 1	2 2 3 4 5 6	2 3 4 5 6 6
APARTMENT BUILDINGS CHURCHES HOSPITALS HOTELS OFFICE BUILDINGS PUBLIC BUILDINGS SCHOOLS	ON GRADE - ALL SPANS THROUGH 20 21 - 30 31 - 40 41 - 50 51 & UP	— 1 1 1 1 1	2 2 3 3 4 5	2 3 3 4 5 6
INDUSTRIAL BUILDINGS - NEAR OFFICE AREAS SPORTS ACTIVITY AREAS - INDOORS RESTAURANTS TRANSPORTATION - AIRPORTS AND BUS TERMINALS	ON GRADE - ALL SPANS THROUGH 20 21 - 30 31 - 40 41 - 50 51 & UP	— 1 1 1 1 1	2 2 2 3 3 4	2 2 3 3 5 5

TYPES OF ISOLATORS AND BASES USED IN THE TABLES

W	3/8" THICK, 40 DUROMETER RIBBED NEOPRENE PADS
WSW	3/8" THICK, 40 DUROMETER RIBBED NEOPRENE PADS SEPARATED BY A 10 GAUGE STEEL SHEET - LOADED TO 50 PSI
N	NEOPRENE IN SHEAR MOUNTS HAVING A 0.25" RATED DEFLECTION WITH INTEGRAL STEEL PLATES AND TAPPED HOLES FOR BOLTING EQUIPMENT
ND	DOUBLE DEFLECTION NEOPRENE MOUNTS HAVING A 0.4" RATED DEFLECTION WITH STEEL PLATES AND TAPPED HOLES FOR BOLTING EQUIPMENT
V	HOUSED STEEL SPRING AND 3/8" NEOPRENE ISOLATORS WITH RUBBER SNUBBERS SEPARATING THE TOP AND BOTTOM HOUSINGS
O	OPEN FREE STANDING STEEL SPRING AND 3/8" NEOPRENE ISOLATORS WITH EQUAL STIFFNESS IN THE HORIZONTAL AND VERTICAL DIRECTIONS
OR	OPEN RESTRICTED STEEL SPRING AND 3/8" NEOPRENE ISOLATORS WITH EQUAL STIFFNESS IN THE HORIZONTAL AND VERTICAL DIRECTIONS AND LIMIT STOPS TO PREVENT LIFTING AND STRESSING OF EQUIPMENT
HN	NEOPRENE MOUNT HANGERS FOR SUSPENDED EQUIPMENT - HANGERS SHALL ACCOMMODATE A 30 DEG. MISALIGNMENT WITHOUT SHORTING OUT
HS	SPRING AND NEOPRENE PAD HANGERS FOR SUSPENDED EQUIPMENT HANGERS SHALL ACCOMMODATE A 30 DEG. MISALIGNMENT WITHOUT SHORTING OUT
HSN	SPRING AND NEOPRENE MOUNT HANGERS FOR SUSPENDED EQUIPMENT HANGERS SHALL ACCOMMODATE A 30 DEG. MISALIGNMENT WITHOUT SHORTING OUT
R	SPRING MOUNT SYSTEM WHICH INCORPORATES AN INTEGRAL RAIL FRAME
IS	INTEGRAL STEEL FRAME DESIGNED TO SUPPORT THE EQUIPMENT AND MOTOR
CI	CONCRETE INERTIAL BASE WITH FULL DEPTH PERIMETER STEEL FRAMES

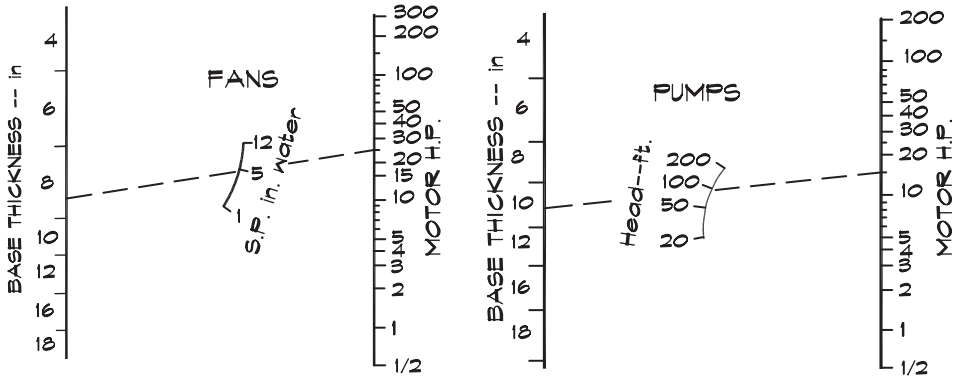
TABLE 11.3 Vibration Isolation Selection Guide (Vibron, 1989)

TYPE OF EQUIPMENT			VIBRATION ISOLATION CATEGORY																	
			1			2			3			4			5			6		
DESIGNATION	DESCRIPTION		BASE	ISLTR	DEFL	BASE	ISLTR	DEFL	BASE	ISLTR	DEFL	BASE	ISLTR	DEFL	BASE	ISLTR	DEFL	BASE	ISLTR	DEFL
CENTRIFUGAL FANS	THROUGH 40 HP	TO 275 RPM	IS	O	3.0	IS	O	3.0	IS	O	3.0	IS	O	3.0	IS	O	3.0	IS	O	3.0
		276-600 RPM	IS	O	1.5	IS	O	1.75	IS	O	2.0	IS	O	2.0	IS	O	2.0	IS	O	2.5
		601 RPM & UP	IS	O	1.0	IS	O	1.0	IS	O	1.0	IS	O	1.75	IS	O	1.75	IS	O	2.0
	50 HP & UP	TO 275 RPM	CI	O	3.0	CI	O	3.0	CI	O	3.5	CI	O	4.0	CI	O	4.0	CI	O	4.0
		276-600 RPM	CI	O	1.5	CI	O	1.75	CI	O	2.0	CI	O	2.5	CI	O	3.0	CI	O	3.0
		601 RPM & UP	CI	O	1.0	CI	O	1.0	CI	O	1.0	CI	O	1.75	CI	O	2.0	CI	O	3.0
TUBULAR CENTRIFUGAL AND AXIAL FANS	ALL SIZES SUSPENDED	TO 750 RPM	-	HS	1.0	-	HS	1.1	-	O	1.5	-	O	2.0	-	O	2.75	-	O	3.0
		751 RPM & UP	-	HS	0.7	-	HS	1.0	-	O	1.1	-	O	1.5	-	O	2.0	-	O	2.75
	ALL SIZES FLOOR MOUNTED	TO 750 RPM	IS	O	0.7	IS	O	1.1	IS	O	1.5	CI	O	2.0	CI	O	2.75	CI	O	3.0
		751 RPM & UP	IS	O	0.7	IS	O	1.0	IS	O	1.1	CI	O	1.5	CI	O	2.0	CI	O	2.75
PACKAGED AH, AC AND H & V UNITS (SUSPENDED)	THRU 7.5 HP	ALL RPM	-	HS	0.7	-	HS	0.7	-	HS	0.7	-	HS	1.0	-	HSN	1.0	-	HSN	1.0
	10 HP & UP AND TO 5" S.P.	TO 600 RPM	-	HS	0.7	-	HS	0.7	-	HS	1.0	-	HS	1.0	-	HSN	1.5	-	HSN	1.75
		601-1000 RPM	-	HS	0.7	-	HS	0.7	-	HS	1.0	-	HS	1.0	-	HSN	1.0	-	HSN	1.5
		1001 RPM & UP	-	HS	0.7	-	HS	0.7	-	HS	0.7	-	HS	0.7	-	HSN	1.0	-	HSN	1.0
PACKAGED AH, AC AND H & V UNITS (FLOOR MOUNTED)	THRU 7.5 HP	ALL RPM	-	ND	0.25	-	V	0.75	-	V	0.75	-	V	1.0	-	V	1.0	-	V	1.0
	10 HP & UP AND TO 5" S.P.	TO 600 RPM	-	V	0.7	-	V	0.7	-	V	1.0	-	O	1.0	-	O	1.5	-	O	1.75
		601-1000 RPM	-	V	0.7	-	V	0.7	-	V	1.0	-	O	1.0	-	O	1.0	-	O	1.5
		1001 RPM & UP	-	V	0.7	-	V	0.7	-	V	0.7	-	O	1.0	-	O	1.0	-	O	1.0
UTILITY SETS	SUSPENDED	TO 500 RPM	-	HS	0.35	-	HS	1.0	-	HS	1.0	-	HS	1.5	-	HS	2.0	-	HS	2.5
		501 RPM & UP	-	HS	0.35	-	HS	1.0	-	HS	1.0	-	HS	1.0	-	HS	1.5	-	HS	2.0
	FLOOR MOUNTED	TO 500 RPM	R	ND	0.35	IS	O	1.0	IS	O	1.5	IS	O	2.0	IS	O	2.5	IS	O	2.75
		501 RPM & UP	R	ND	0.35	IS	O	1.0	IS	O	1.0	IS	O	1.5	IS	O	2.0	IS	O	2.5
AIR COOLED CONDENSERS AND CLOSED CIRCUIT COOLING TOWERS COOLERS	CENTRIFUGAL UNITS	TO 400 RPM	-	OR	2.0	-	OR	2.0	-	OR	2.0	-	OR	2.5	-	OR	3.0	-	OR	3.0
		401 RPM & UP	-	OR	1.0	-	OR	1.0	-	OR	1.5	-	OR	1.5	-	OR	2.0	-	OR	2.0
	AXIAL FLOW UNITS	TO 300 RPM	-	OR	2.0	-	OR	2.0	-	OR	2.0	-	OR	3.0	-	OR	3.0	-	OR	3.5
		301-700 RPM	-	OR	1.5	-	OR	1.5	-	OR	1.5	-	OR	2.0	-	OR	2.0	-	OR	3.0
		701 RPM & UP	-	OR	1.0	-	OR	1.0	-	OR	1.5	-	OR	1.5	-	OR	2.0	-	OR	3.0
PACKAGED STEAM GENERATORS	ALL SIZES		-	OR	1.0	-	OR	1.5	-	OR	1.75	-	OR	2.0	-	OR	2.5	-	OR	2.5

TABLE 11.4 Vibration Isolation Selection Guide (Vibron, 1989)

TYPE OF EQUIPMENT			VIBRATION ISOLATION CATEGORY																	
			1			2			3			4			5			6		
DESIGNATION	DESCRIPTION		BASE	ISLTR	DEFL	BASE	ISLTR	DEFL	BASE	ISLTR	DEFL	BASE	ISLTR	DEFL	BASE	ISLTR	DEFL	BASE	ISLTR	DEFL
CENTRIFUGAL PUMPS	CLOSE COUPLED ALL RPM	THRU 5 HP	CI	O	1.0	CI	O	1.0	CI	O	1.0	CI	O	1.0	CI	O	1.5	CI	O	1.5
		7.5 HP & UP	CI	O	1.0	CI	O	1.0	CI	O	1.0	CI	O	1.5	CI	O	2.0	CI	O	2.0
	BASE MOUNTED ALL RPM	THRU 5 HP	CI	O	1.0	CI	O	1.0	CI	O	1.0	CI	O	1.5	CI	O	2.0	CI	O	2.0
		7.5 HP - 75 HP	CI	O	1.0	CI	O	1.0	CI	O	1.0	CI	O	1.5	CI	O	2.0	CI	O	3.5
		75 HP & UP	CI	O	1.0	CI	O	1.0	CI	O	1.5	CI	O	2.0	CI	O	2.5	CI	O	3.0
INLINE PUMPS	THRU 40 HP		-	WSW	0.15	-	V	0.7	CI	O	1.0	-	O	1.0	CI	O	1.5	CI	O	2.0
	40 HP & UP		-	WSW	0.15	-	V	1.0	CI	O	1.0	CI	O	1.5	CI	O	2.0	CI	O	2.5
RECIPROCATING AIR OR REFRIGERATION COMPRESSORS	THRU 15 HP	TO 500 RPM	CI	O	1.0	CI	O	1.0	CI	O	1.5	CI	O	1.75	CI	O	2.0	CI	O	2.5
		501-750 RPM	CI	O	0.75	CI	O	1.0	CI	O	1.5	CI	O	1.5	CI	O	2.0	CI	O	2.0
		751 RPM & UP	CI	O	0.75	CI	O	1.0	CI	O	1.5	CI	O	1.5	CI	O	2.0	CI	O	2.0
	20 HP AND UP	TO 500 RPM	CI	O	1.0	CI	O	1.5	CI	O	1.75	CI	O	2.0	CI	O	2.0	CI	O	1.75
		501-750 RPM	CI	O	1.0	CI	O	1.25	CI	O	1.5	CI	O	1.75	CI	O	1.75	CI	O	1.5
		751 RPM & UP	CI	O	0.75	CI	O	1.0	CI	O	1.25	CI	O	1.5	CI	O	1.5	CI	O	1.0
CENTRIFUGAL & REFRIGERATION MACHINES	ALL SIZES		IS	OR	1.0	IS	OR	1.5	IS	OR	2.0	IS	OR	2.0	IS	OR	2.5	IS	OR	2.5
ABSORPTION REFRIGERATION MACHINES	ALL SIZES		IS	OR	1.0	IS	OR	1.0	IS	OR	1.5	IS	OR	1.5	IS	OR	2.0	IS	OR	2.0
ELECTRICAL DISTRIBUTION TRANSFORMERS	0 - 50 KVA		-	W	0.06	-	WSW	0.12	-	WSW	0.12	-	WSW	0.12	-	V	0.7	-	V	0.7
	51 - 250 KVA		-	N	0.25	-	N	0.25	-	ND	0.35	-	ND	0.4	-	ND	0.4	-	ND	0.4
	251 KVA & LARGER		-	ND	0.35	-	V	0.7	-	V	0.7	-	V	0.7	-	V	1.0	-	V	1.0

FIGURE 11.13 The Thickness of Concrete Inertial Bases (Vibron, 1989)



AIR AND REFRIGERATION COMPRESSORS	
Motor H.P.	Base Thickness -- in.
through 2	8
3 & 5	10
71/2 to 20	12
25 to 40	14
50 to 75	16
100& 125	20
125 up	24

Examples of various recommendations on the support of rooftop equipment are shown in Fig. 11.14 (Schaffer, 1991).

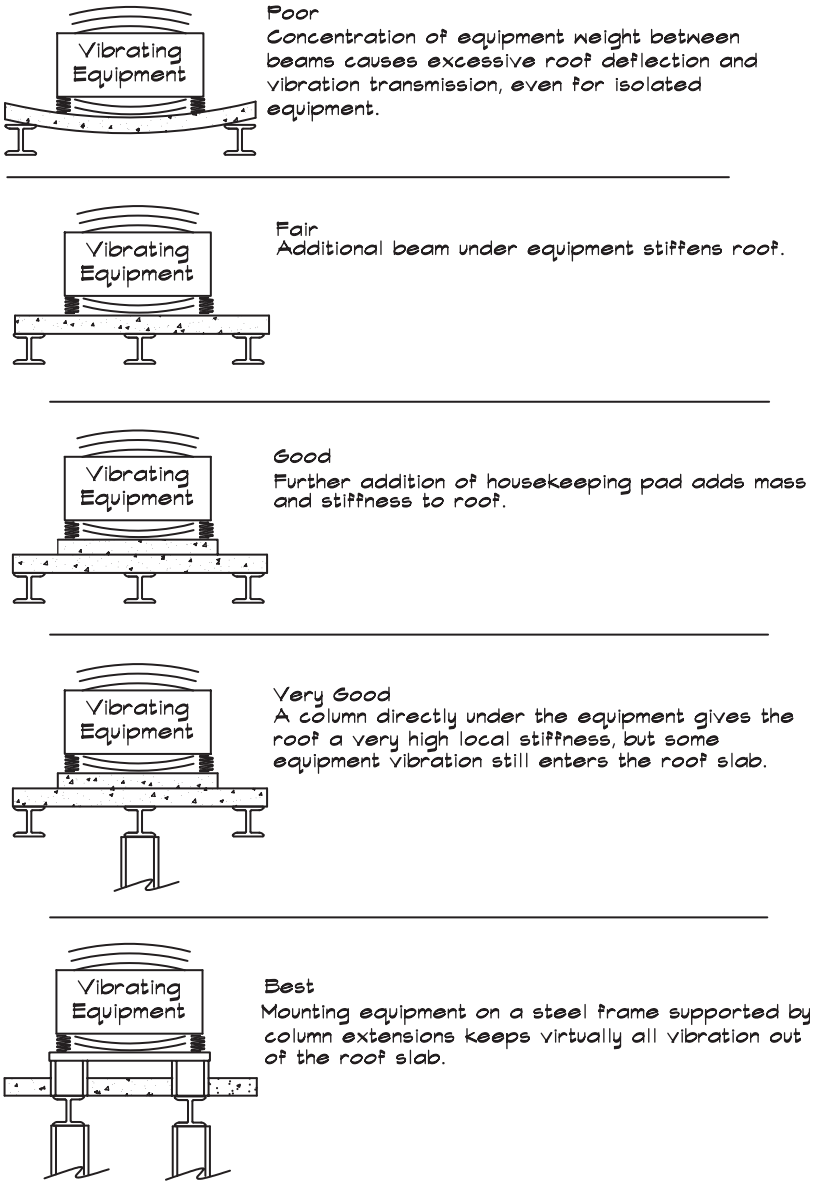
Inertial Bases

When the source of vibration is a piece of mechanical equipment with a large rotating mass or a high initial torque, it is good practice to mount it on a concrete base that is itself supported on spring isolators. The additional mass does not increase the isolation efficiency since the springs must be selected to support both the equipment and the base, and the overall spring deflection will probably not change appreciably. The advantage of having the base is that for a given driving force, such as the eccentricity of a rotating part, there is a lower overall displacement due to the extra mass of the combined base plus equipment. Inertial bases also aid in the stabilization of tall pieces of equipment, equipment with a large rocking component, and equipment requiring thrust restraint.

Concrete inertial bases are used in the isolation of pumps and provide additional frame stiffness, which a pump frequently requires. Pump bases are sized so that their weight is about two to three times that of the supported equipment. Any piping, attached to a pump mounted on an isolated base, must be supported from the inertial base or by overhead spring hangers. It must not be rigidly supported from a wall, floor, or roof slab unless it is in a noncritical location.

Where unbalanced equipment, such as single- or double-cylinder low-speed air compressors are to be isolated, the weight of the inertial base is calculated from the unbalanced

FIGURE 11.14 Structural Support of Rooftop Equipment (Schaffer, 1991)



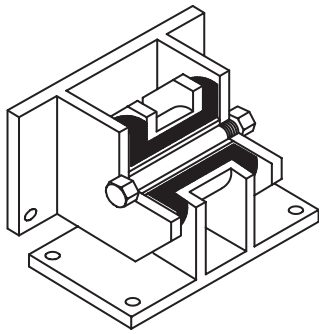
force, which can be obtained from the manufacturer. These bases frequently must be five to seven times the weight of the compressor to control the motion.

Concrete bases also offer resistance to induced forces such as fan thrust. Isolation manufacturers (Mason, 1968) recommend that a base weighing from one to three times the fan weight be used to control thrust for fans above 6" of static pressure.

Earthquake Restraints

In areas of high seismic activity, vibration isolated equipment must be constrained from moving during an earthquake. The seismic restraint system must not degrade the performance

FIGURE 11.15 Earthquake Restraint (Mason Industries, 1998)



of the vibration isolation. Some specialized isolators incorporate seismic restraints, but most vibration isolators do not since a restraint device must control motion in any direction. A standard method of providing three-dimensional restraint is shown in Fig. 11.15 using a commercial three-axis restraint system. Lightweight hanger-supported equipment can be restrained by means of several slack braided-steel cables. Any earthquake restraint system must comply with local codes and should be reviewed by a structural engineer.

Pipe Isolation

Piping can conduct noise and vibration generated through fluid motion and by being connected to vibrating equipment. Fluid flow in piping generates sound power levels that are dependent on the flow velocity. Pipes and electrical conduits that are attached directly to vibrating equipment and to a supporting structure serve as a transmission path, which short circuits otherwise adequate vibration isolation. Any rigid piping attached to isolated equipment such as pumps, refrigeration machines, and condensers must be separately vibration isolated, typically at the first three points of support, which for large pipe is about 15 m (50 ft). It should be suspended by means of an isolator having a deflection that is at least that of the supported equipment or 3/4", whichever is greater.

There is a significant difference in the weight of a large water pipe, depending on whether it is empty or filled. Isolated equipment will move up when the pipe system is drained, and in doing so, will stress elbows and joints. The suspension system should allow for normal motion of the pipe under these conditions. Risers and other long pipe runs will expand and contract as they are heated and cooled and should be resiliently mounted. Even when fluid is not flowing, a popping noise can be generated as the pipe slides past a stud or other support point during heating or cooling.

In critical applications such as condominiums, water, waste, and refrigeration pipes should be isolated from making contact with structural elements for their entire length. Table 11.5 gives typical recommendations on the types of materials used for the isolation of plumbing and piping. These recommendations also apply to the support of piping at points where it penetrates a floor.

Several examples of proper isolation of piping connected to pumps are shown in Fig. 11.16. On all piping greater than 5" (13 cm) diameter, flexible pipe couplings are necessary between the pump outlet and the pipe run. Even with smaller diameter pipes they can be very helpful in decreasing downstream vibrations and associated noise. They act as vibration isolators by breaking the mechanical coupling between the pump and the pipe, and they

TABLE 11.5 Typical Plumbing Isolation Materials

NOMINAL PIPE SIZE (Dia, in)	REQUIRED ISOLATION HORIZONTAL	REQUIRED ISOLATION VERTICAL
1/2	1/4" FELT	1/4" FELT
3/4	3/8" FELT	3/8" FELT
1	3/8" FELT	3/8" FELT
1 1/2	3/8" FELT	3/8" FELT
2	1/2" FELT	WSW PADS
3	HN ISOLATORS	ND MOUNTS
4	HN ISOLATORS	ND MOUNTS
> 5	HS ISOLATORS 3/4" DEFL	V ISOLATORS 3/4" DEFL
* NOT REQUIRED FOR VENT STACKS, FIRE SPRINKLERS, OR GAS PIPING.		

help compensate for pipe misalignment and thermal expansion. Flexible pipe connections alone are usually not sufficient to isolate pipe transmitted vibrations but are part of an overall control strategy, which includes vibration isolation of the mechanical equipment and piping.

In high pressure hydraulic systems much of the vibration can be transmitted through the fluid so that pulse dampeners inserted in the pipe run can be helpful. These consist of a gas filled bladder, surrounding the fluid, into which the pressure pulse can expand and dissipate.

Where pipes are located in rated construction elements, closing off leaks at structural penetrations is critical to maintain the acoustical rating. Here the normal order of construction dictates the method of isolation. In concrete and steel structures, slabs are poured and then cored to accommodate pipe runs. In wood construction, piping is installed along with the framing, often preceding the pouring of any concrete fill. In both building types holes should be oversized by 1" (25 mm) more than the pipe diameter to insure that the pipe does not make direct structural contact. They are then stuffed with insulation, safing, or fire stop, and sealed. In slab construction the sealant can be a heavy mastic. With walls, the holes are covered with drywall leaving a 1/8" (3 mm) gap that is caulked. Pipe sleeves, which wrap the pipe at the penetration, are also commercially available. Details are shown in Fig. 11.17.

Electrical Connections

Where electrical connections are made to isolated equipment, the conduit must not short out the vibration isolation. If rigid conduit is used it should include a flexible section to isolate this path. The section should be long enough and slack enough that a 360° loop can be made in it.

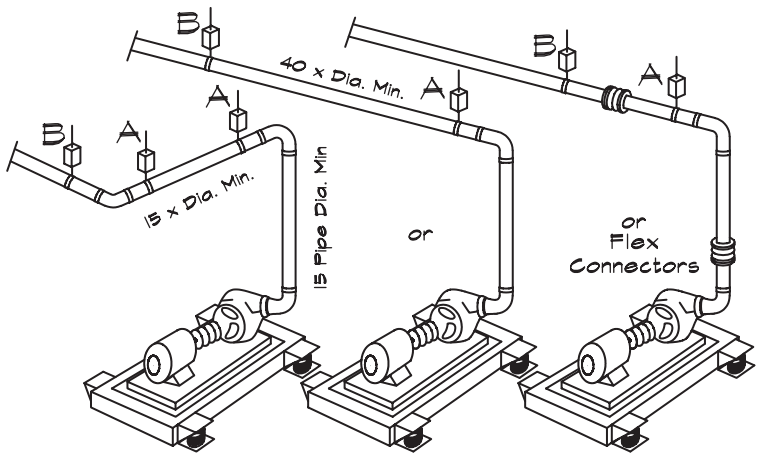
Duct Isolation

High-pressure ductwork having a static pressure of 4" (10 cm) or greater should be isolated for a distance of 30 ft (10 m) from the fan. Ducts are suspended on spring hangers with a minimum static deflection of 0.75" (19 mm), which should be spaced 10 ft (3 m) or less apart.

Roof-mounted sheet metal ductwork, located above sensitive occupancies such as studios, should be supported on vibration isolators having a deflection equal to that of the

FIGURE 11.16 Vibration Isolation of Piping and Ductwork (Vibron, 1989)

Vibration Isolation of Piping Connected to Isolated Equipment



Each of the above alternatives will result in equivalent piping isolation from a flexibly supported pump if the following suggestions are implemented.

(1) The pump inertia bases should be at least 2.5 times the unit weight.

(2) Piping isolators should have the following static deflections:
Points A and B - Equal to the pump base static deflection
Beyond Point B - At least 0.7" for 20 ft. from point B with isolators spaced at 10 ft. intervals. For critical locations the complete pipe run should be isolated, with isolator static deflection of at least 0.7" and spaced at 10 ft. intervals.

(3) For pipe diameters above 5", flexible connections are necessary.

Vibration Isolation of High Pressure Ductwork

High pressure (above 4" static pressure) duct runs should be isolated for a distance of 30 ft. from the fan. Ducts supported by hanger isolators should have a static deflection of at least 3/4" and be spaced 10 ft. apart.

FIGURE 11.17 Pipe or Duct Penetration

The opening is oversized to allow for the penetration and covered with the same number of layers as are on the wall.

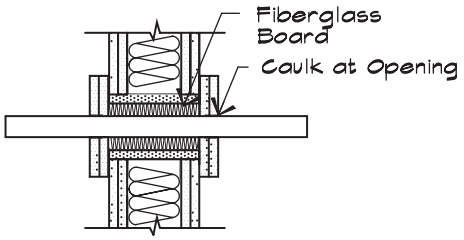
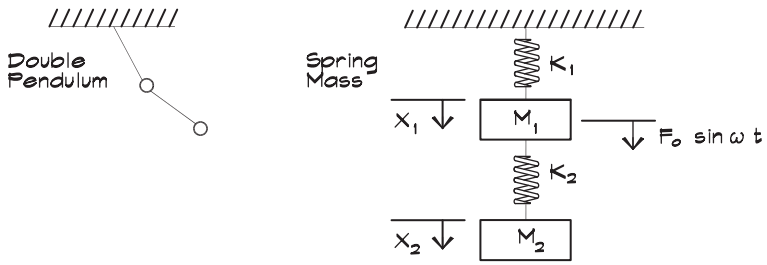


FIGURE 11.18 Forced Excitation of an Undamped Two Degree of Freedom System (Ruzicka, 1971)



isolated equipment to which they are attached, for the first three points of support. Beyond that point the ducts can be supported on mounts having half that deflection.

11.5 TWO DEGREE OF FREEDOM SYSTEMS

Two Undamped Oscillators

Although the one degree of freedom model is the most commonly utilized system for most vibration analysis problems, often situations arise that exhibit more complex motion. A model of a two degree of freedom system is shown in Fig. 11.18. This system consists of two masses and two springs with a sinusoidal force applied to one of the masses. The equations of motion can be written as

$$m_1 \ddot{x}_1 = k_2 (x_2 - x_1) - k_1 x_1 + F_0 \sin \omega t \tag{11.29}$$

$$m_2 \ddot{x}_2 = -k_2 (x_2 - x_1) \tag{11.30}$$

If we make the following substitutions

$$\omega_1 = \sqrt{k_1 / m_1} \qquad X_0 = F_0 / k_1$$

$$\omega_2 = \sqrt{k_2 / m_2}$$

and write the solution in terms of sinusoidal functions of displacement

$$x_1 = X_1 \sin \omega t$$

and

$$x_2 = X_2 \sin \omega t$$

Substituting these expressions into Eqs. 11.29 and 11.30, we obtain an expression for the relationship between the amplitude displacements

$$\left[1 + \frac{k_2}{k_1} - \left(\frac{\omega}{\omega_1}\right)^2\right] X_1 - \left(\frac{k_2}{k_1}\right) X_2 = X_0 \tag{11.31}$$

and

$$-X_1 + \left[1 - \left(\frac{\omega}{\omega_2}\right)^2\right] X_2 = 0 \tag{11.32}$$

We can then study the system behavior by looking at the expressions for the ratio of the two amplitudes

$$\frac{X_1}{X_0} = \frac{\left[1 - (\omega/\omega_2)^2\right]}{\left[1 + \frac{k_2}{k_1} - \left(\frac{\omega}{\omega_1}\right)^2\right]\left[1 - \left(\frac{\omega}{\omega_2}\right)^2\right] - \frac{k_2}{k_1}} \tag{11.33}$$

$$\frac{X_2}{X_0} = \frac{1}{\left[1 + \frac{k_2}{k_1} - \left(\frac{\omega}{\omega_1}\right)^2\right]\left[1 - \left(\frac{\omega}{\omega_2}\right)^2\right] - \frac{k_2}{k_1}} \tag{11.34}$$

Now there are two resonant frequencies of the spring mass system, ω_1 and ω_2. From Eq. 11.33 we see that when the natural frequency of the second spring mass system matches the driving frequency of the impressed force, the numerator, and thus the amplitude X_1, goes to zero. At this frequency the amplitude of the second mass is

$$X_2 = -\frac{k_1}{k_2} X_0 = -\frac{F_0}{k_2} \tag{11.35}$$

where the minus sign indicates that the motion is out of phase with, and just counterbalances, the driving force. This is the principal behind a second form of vibration isolation known as mass absorption or mass damping. The absorber mass must be selected so as to match the applied force, taking into consideration the allowable spring deflection.

Two Damped Oscillators

Figure 11.19 gives the results of an imposed force on a damped two-degree of freedom spring mass system. The two resonant peaks are at different frequencies, with $\omega_2 > \omega_1$. In this example there is a relatively narrow frequency range where the second mass provides appreciable mass damping. Indeed it may generate an unwelcome resonant peak, slightly above the fundamental frequency of the second mass.

A mass absorber is most effective when it is used to damp the natural resonant frequency of the first spring mass system. If the ω_2 is selected to match ω_1, then the two resonant peaks coincide. When a broadband vibration or an impulsive load is applied to the system, the zero in the numerator in Eq. 11.33 smothers the resonant peaks and mass damping occurs. Figure 11.20 illustrates this case.

In long-span floor systems the floor itself acts like a spring mass system. A weight, suspended by isolator springs below a floor at a point of maximum amplitude, can be used as a dynamic absorber. These weights, which are usually 1% to 2% of the weight of the

FIGURE 11.19 Forced Response of a Two Degree of Freedom System (Ruzicka, 1971)

Transmissibility curves for a vibration isolation system to which is attached a passive vibration absorber tuned to a high frequency component of vibration excitation.

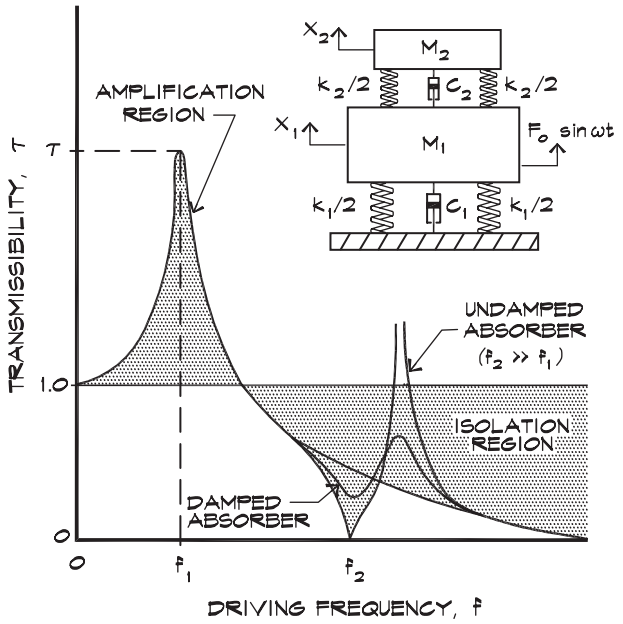
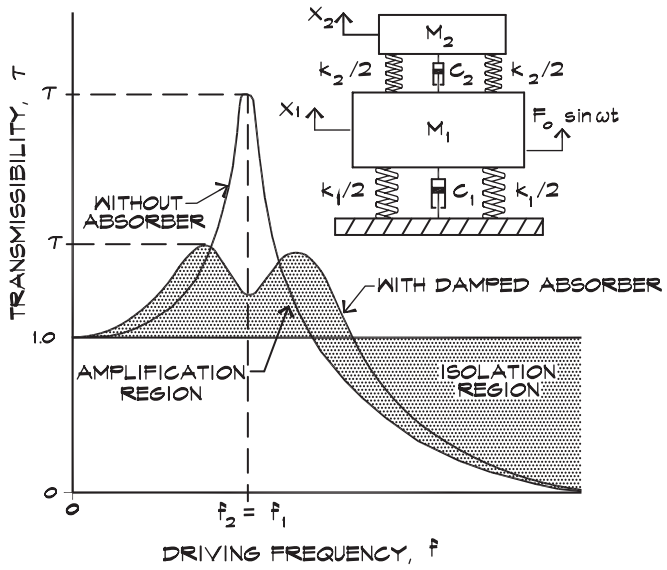


FIGURE 11.20 Forced Response of a Two Degree of Freedom System Near Resonance (Ruzicka, 1971)

Transmissibility curve for a spring mass system to which is attached a passive vibration absorber tuned so as to suppress the main system resonant response.



relevant floor area, are hung between the ceiling and the slab. It is not advisable to use the ceiling itself as the dynamic absorber, since mass damping works to minimize floor motion by maximizing the motion of the suspended mass. If the ceiling motion is maximized, it will radiate a high level of noise at the floor resonance.

Mass absorbers have also been used to damp the natural swaying motion of large towers such as the CN Tower in Toronto, Canada, using a dynamic pendulum. The double pendulum is another two degree of freedom system whose behavior is similar to that of a double spring mass. In this example the tower is encircled with a donut-shaped mass that is suspended as a pendulum. The mass is located at the point of maximum displacement of the normal modes of the structure. In the case of tall towers, the second and third modes are usually damped. The maximum displacement of the first mode occurs at the top of the tower and practical considerations prevent the suspension of a pendulum from this point. Two donut-shaped pendulums were used at the 1/3 and 1/2 points of the structure where they counter the second and third modes of vibration.

11.6 FLOOR VIBRATIONS

The vibration of floors due to motions induced by walking or mechanical equipment can be a source of complaints in modern building structures, particularly where lightweight construction such as concrete on steel deck, steel joists, or concrete on wood joist construction is used. Usually the vibration is a transient flexural motion of the floor system in response to impact loading from human activity (Allen and Swallow, 1975), which can be walking, jumping, or continuous mechanical excitation. The induced amplitudes are seldom enough to be of structural consequence; however, in extreme cases they may cause movement in light fixtures or other suspended items. The effects of floor vibrations are not limited to receivers located immediately below. With the advent of fitness centers, which feature aerobics, induced vibrations can be felt laterally 100 feet away on the same slab as well as up to 10 stories below (Allen, 1997).

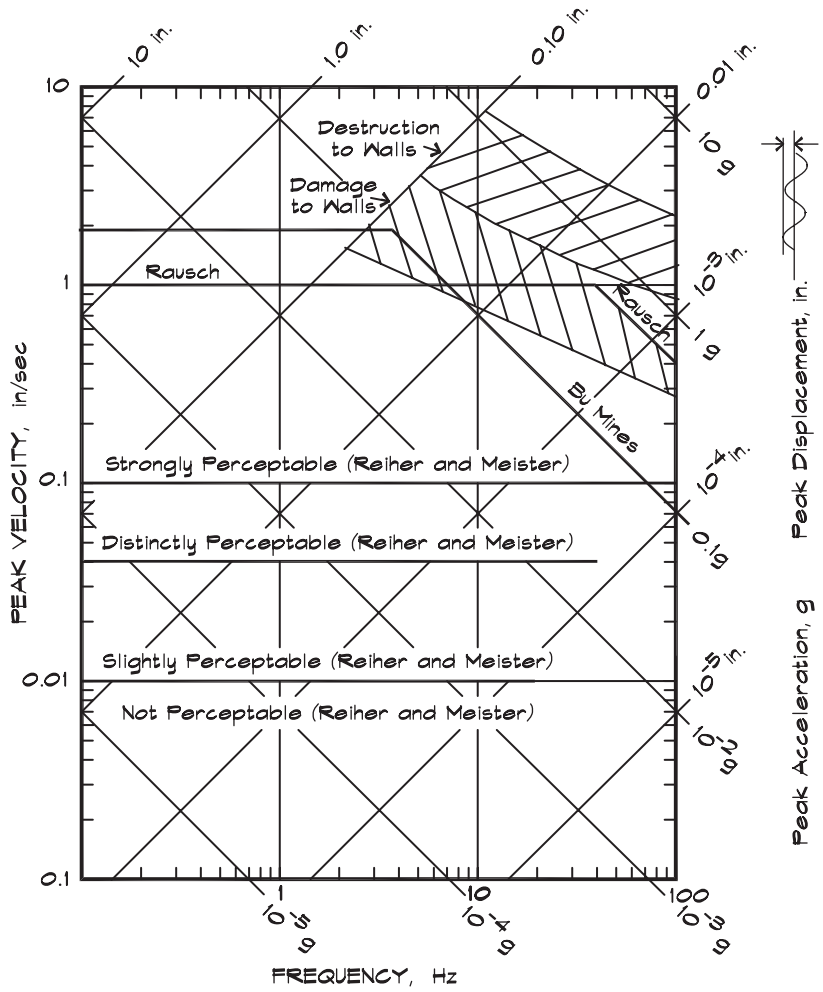
Sensitivity to Steady Floor Vibrations

People, equipment, and sophisticated manufacturing processes, such as computer chip production, are sensitive to floor vibrations. The degree of sensitivity varies with the process and various authors have published recommendations. One of the earliest was documented by Reiher and Meister (1931) and is shown in Fig. 11.21. These were human responses determined by standing subjects on a shaker table and subjecting them to continuous vertical motion. Subjects react more vigorously to higher velocities, and for high amplitudes, awareness increases with frequency. Also shown are the Rausch (1943) limits for machines and machine foundations and the US Bureau of Mines criteria for structural safety against damage from blasting.

Sensitivity to Transient Floor Vibrations

Vibrational excitation of floor systems may be steady or transient; however, it is usually the case that steady sources of vibration can be isolated. Transient vibrations due to footfall or other impulsive loads are absorbed principally by the damping of the floor. Damping provides a function somewhat akin to absorption in the control of reverberant sound in a room. People react, not only to the initial amplitude of the vibration, but also to its duration.

FIGURE 11.21 Response Spectra for Continuous Vibration (Richart et al., 1970; Reiher and Meister, 1931)



Investigators use tapping machines, walking at a normal pace (about 2 steps per second), and a heel drop test, where a subject raises up on his toes and drops his full weight back on his heels, as impulsive sources. This latter test represents a nearly worst-case scenario for human induced vibration, with aerobic studios and judo dojos being the exception.

After studying a number of steel-joist concrete-slab structures, Lenzen (1966) suggested that the original Reiher-Meister scale could be applied to floor systems having less than 5% of critical damping, if the amplitude scale were increased by a factor of 10. This means that we are less sensitive to floor vibration when it is sufficiently damped, in this case when only 20% of the initial amplitude remains after five cycles. He further suggested that if a vibration persists 12 cycles in reaching 20% of the initial amplitude, human response is the same as to steady vibration. Allen (1974), using his own experimental data along with observations of Goldman, suggested a series of annoyance thresholds for different levels of damping. This work, along with that of Allen and Rainer (1976), was adopted as a Canadian National Standard, which is shown in Fig. 11.22.

FIGURE 11.22 Annoyance Thresholds for Vibrations (Allen, 1974)

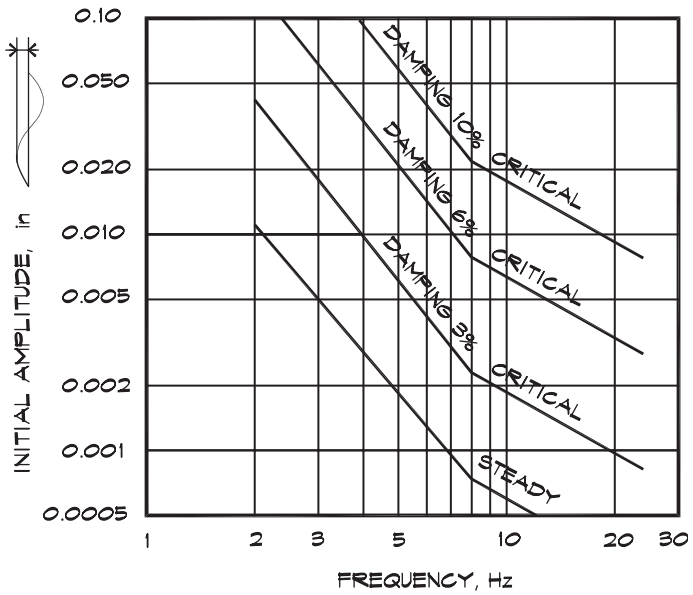
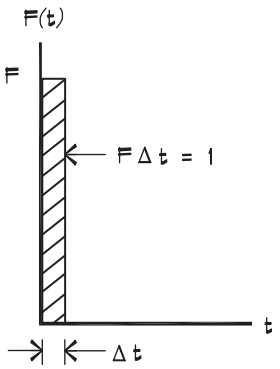


FIGURE 11.23 Impulsive Force



Vibrational Response to an Impulsive Force

When a linear system, such as a spring mass damper, is driven by an impulsive force we can calculate the overall response. For the study of vibrations in buildings the system of interest here is a floor and the impulsive force is a footfall generated by someone walking. An impulse force is one in which the force acts over a very short period of time. An impulse can be defined as

$$\hat{F} = \int_{t}^{t+\Delta t} F\, dt \cong F\, \Delta t \tag{11.36}$$

Figure 11.23 shows an example of an impulsive force, having a magnitude F and a duration Δt. An impulsive force, such as a hammer blow, can be very large; however, since it occurs over a rather short period of time, the impulse is finite. When the impulse is normalized to 1 it is called a unit impulse.

FIGURE 11.24 Response of a Damped System to a Delta Function Impulse $\widehat{F}$ (Thomson, 1965)

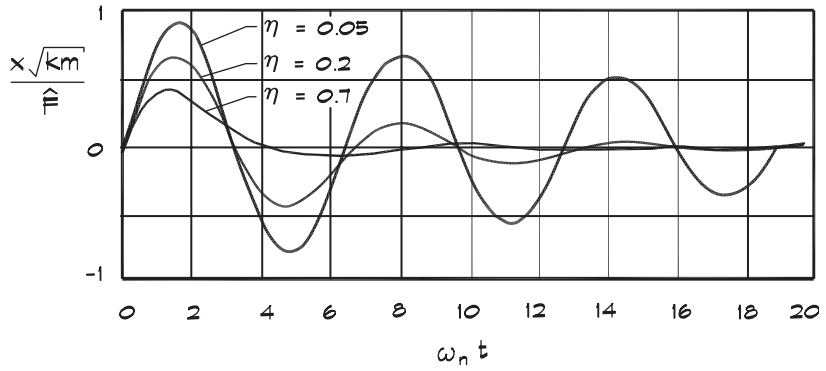


Figure 11.24 illustrates the response of a damped spring mass system under an impulse force for various values of the damping coefficient. From Newton's law, $F\,\Delta t = m\,\dot{x}_2 - m\,\dot{x}_1$. When an impulsive force is applied to a mass for a short time the response is a change in velocity without an appreciable change in displacement. The velocity changes rapidly from zero to an initial value of $\hat{F}/m$. We can use this as the initial boundary condition, assuming an initial displacement of zero, by plugging into the general undamped solution (Eq. 11.6). We get the response to the impulse force

$$x = \frac{\hat{F}}{m\,\omega_n} \sin \omega_n t \tag{11.37}$$

where ω_n is the undamped natural frequency of the spring mass system. If the system is damped, we can use the same procedure to calculate the response by plugging into Eq. 11.12.

$$x = \frac{\hat{F}}{m\,\omega_n \sqrt{1 - \eta^2}} e^{-\eta\,\omega_n t} \sin\left(\sqrt{1 - \eta^2}\,\omega_n t\right) \tag{11.38}$$

Response to an Arbitrary Force

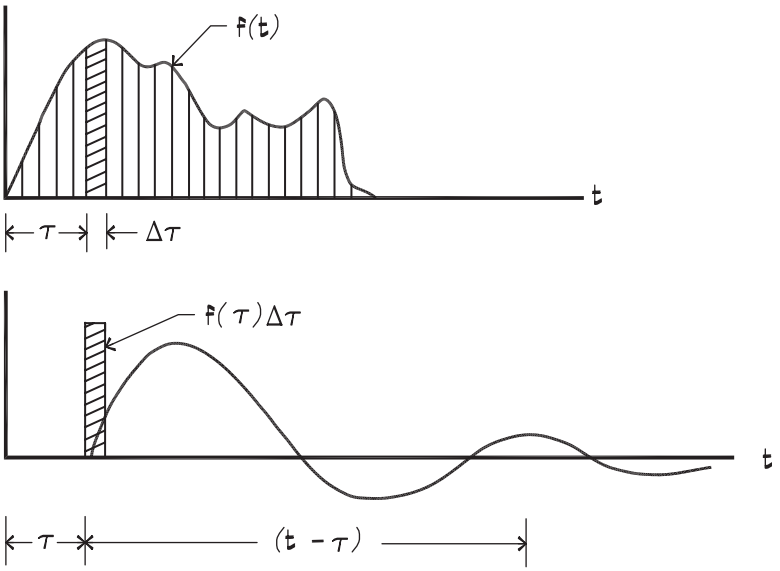
The impulse response in Eq. 11.38 is a fundamental property of the system. It is given a special designation, $g(t)$, where $x = \hat{F}\,g(t)$. Once the system response to a unit impulse (sometimes called a delta function) has been determined, it is possible to calculate the response to an arbitrary force $f(t)$ by integrating (summing) the effects of a series of impulses as illustrated in Fig. 11.25.

At a particular time τ, the force function has a value, which can be described by an impulse $\hat{F} = f(\tau)\,\Delta\,\tau$. The contribution of this slice of the force function on the system response at some elapsed time $t - \tau$ after the beginning of that particular pulse is given by

$$x = f(\tau)\,\Delta\tau\;\, g\,(t - \tau) \tag{11.39}$$

and the response to all the small force pulses is given by integrating over the total time, t_p, the force is applied. If the time of interest is less than t_p, the limit of integration becomes the

FIGURE 11.25 An Arbitrary Pulse as a Series of Impulses (Thomson, 1965)



time of interest.

$$x(t) = \int_0^{t_p} f(\tau)\ g(t-\tau)\, d\tau \tag{11.40}$$

This integral is known by various names including the Duhamel integral, the summation integral, and the convolution integral. It says that if we know the system impulse response, we can obtain the system response for any other type of input by performing the integration. This has profound implications for the modeling of concert halls and other spaces since the impulse response of a room can be modeled and the driving force can be music. Thus we can listen to the sound of a concert hall before it is built.

Response to a Step Function

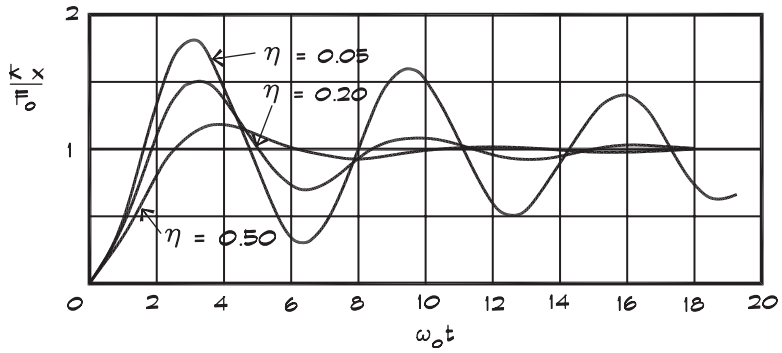
If the shape of a force applied to a spring mass system consists of a constant force that is instantaneously applied, we can substitute the force time behavior, $f(t) = F_0$, into Eq. 11.40 along with the system response to obtain the response behavior. For an undamped spring mass system the result is

$$x(t) = \int_0^{t} \frac{F_0}{m\,\omega_n} \sin\omega_n\,(t-\tau)\, d\tau \tag{11.41}$$

which is

$$x(t) = \frac{F_0}{k}\,(1 - \cos\omega_n\, t) \tag{11.42}$$

FIGURE 11.26 Response of a Damped System to a Unit Step Function (Thomson, 1965)



and for the damped system the result is (see Harris and Crede, 1961; or Thomson, 1965)

$$x = \frac{F_0}{k}\left[1 - \frac{e^{-\eta\,\omega_n\,t}}{\sqrt{1-\eta^2}}\cos\left(\sqrt{1-\eta^2}\,\omega_n\,t - \psi\right)\right] \tag{11.43}$$

where $\tan\psi = \dfrac{\eta}{\sqrt{1-\eta^2}}$

Figure 11.26 shows the system response for a damped spring mass as a function of damping. When the damping is zero the maximum amplitude is twice the displacement that the system would experience if the load were applied slowly.

Vibrational Response of a Floor to Footfall

A footstep consists of two step functions, one when the load is applied and one when it is released. Ungar and White (1979) have modeled this behavior using a versed sine pulse in Fig. 11.27, and have calculated the envelope for the dynamic amplification, defined as the ratio of the maximum dynamic amplitude divided by the static deflection obtained under the load, F_m.

$$A_m = \frac{X_{max}}{X_{static}} = \frac{\sqrt{2\left(1 + \cos 2\pi\, f_n\, t_0\right)}}{\left[1 - \left(2\, f_n\, t_0\right)^2\right]} \tag{11.44}$$

FIGURE 11.27 Idealized Footstep Force Pulse (Ungar and White, 1979)

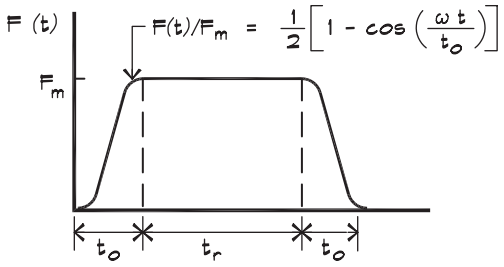
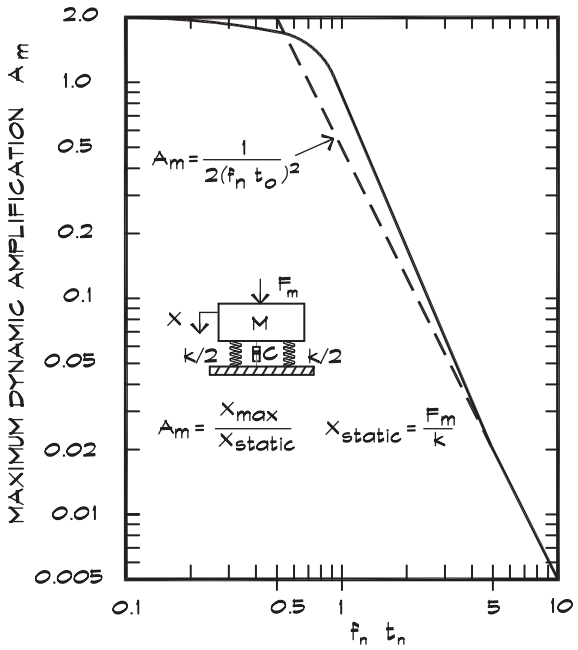


FIGURE 11.28 Maximum Dynamic Deflection Due to a Footstep Pulse (Ungar and White, 1979)



where $f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}}$ and t_0 is the rise time of the pulse. Note that k is the stiffness at the point where the footstep is taken. This equation does not give us the detailed behavior of the motion but gives us the envelope of the maximum deflection with resonant frequency, which is often sufficient for design purposes. For values of $f_n\ t_0$ that are small when compared to 1, the maximum dynamic amplification $A_m \cong 2$. For large values of $f_n\ t_0$, the amplification becomes $A_m \cong a / (2\ f_n\ t_0)^2$, where a varies between 0 and 2, so that under these conditions $A_m \leq 1 / \left[2\ (f_n\ t_0)^2\right]$. Figure 11.28 gives a plot of the upper bound envelope for A_m. In Eq. 11.44 we note that the product $f_n\ t_0$ is equal to t_0 / t_n, the ratio of the pulse rise time to the natural period of floor vibration.

Figure 11.29 shows published data on footstep forces generated by a 150 lb (68 kg) male walker, and Fig. 11.30 shows the dependence of the rise time and force on walking speed. The figures allow us to estimate the maximum deflection of a floor system for various values of the resonant floor frequency.

While floors have a multitude of vibrational modes, the fundamental is usually the most important. It exhibits the lowest resonant frequency, is the most directly excitable structural motion, and has the softest (lowest impedance) point at its antinode. Some measured results are shown in Fig. 11.31 for a concrete I-beam structure. Although only two floor modes have been predicted, and floors are not pure undamped spring mass systems, the curve neatly bounds the remainder of the modes.

Control of Floor Vibrations

When it is desirable to control floor vibration for human comfort, it is important to limit the maximum amplitude as well as increase the damping. If the driving force is footfall, we

FIGURE 11.29 The Footstep Force Pulse Produced by a 150 lb (68 kg) Male Walker (Ungar and White, 1979)

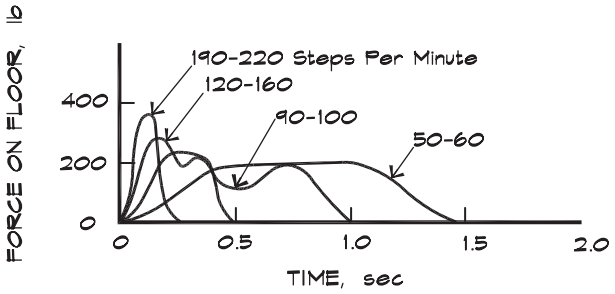
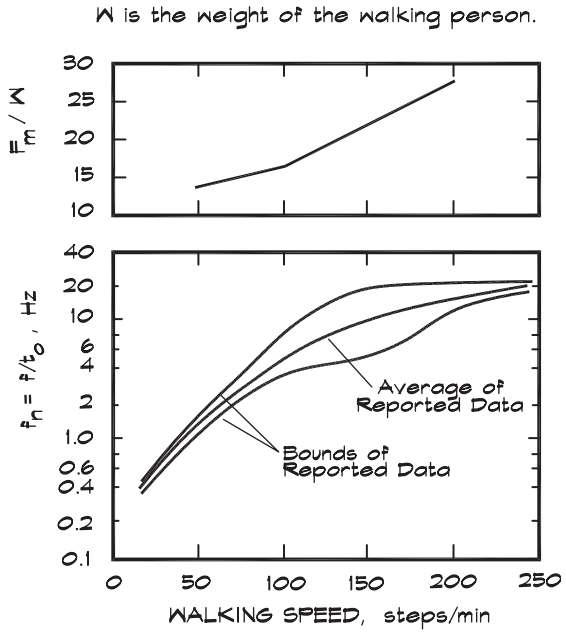


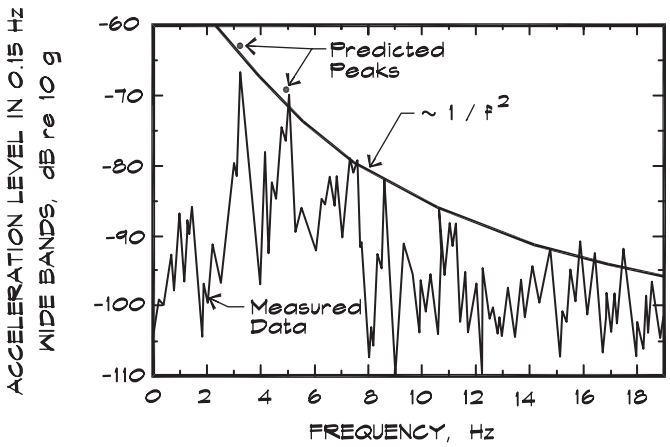
FIGURE 11.30 Dependence of the Maximum Force F and the Rise Time t of a Footstep Pulse on the Walking Speed (Ungar and White, 1979)



can use the amplification factor rise time t_0 to the natural period t_n of the structural mode. When the pulse rise time is a small fraction of the natural period we might expect a different behavior than for cases where the rise time is a large multiple of the period. This is illustrated in Fig. 11.28. From the graph it is reasonable to take the value of $f_n\ t_0 = 0.5$ as the dividing point between these two regions. From Fig. 11.29, the rise time for a typical rapid walker is about a tenth of a second, which means that the dividing point corresponds to a floor resonance of about 5 Hz. The fundamental resonances of most concrete floor systems fall into the region between 5 and 8 Hz, so that rapid walking on these structures corresponds to the region where $f_n\ t_0 \geq 0.5$. For this region,

$$x_{max} = F_m / 2\,k\,(f_n\ t_0)^2 \cong 2\,\pi^2\,F_m\,M / {t_0}^2\,k^2 \tag{11.45}$$

FIGURE 11.31 Footfall Response of a Concrete I-Beam Floor Structure (Ungar and White, 1979)



and

$$a_{max} \cong \left(2\,\pi\,f_n\right)^2\, x_{max} = 2\,\pi^2\, F_m\,/ = t_0{}^2\, k \tag{11.46}$$

where a_{max} represents the maximum floor acceleration, k the local modal stiffness, and M the corresponding mass. It is clear that the structural stiffness is the most important component in decreasing both the maximum amplitude and the maximum acceleration. The floor mass does not appear in the equation for acceleration. The maximum displacement increases with mass, unless the mass increases the stiffness.

In the region where $f_n\, t_0 \leq 0.5$, which would correspond to a very long span floor, we find that

$$x_{max} \cong F_m\,/\,k \tag{11.47}$$

and

$$a_{max} \cong \left(2\,\pi\,f_n\right)^2\, x_{max} = 2\,F_m\,/\,M \tag{11.48}$$

Here only the stiffness affects the maximum displacement and only the mass affects the maximum acceleration.

Allen and Swallow (1974) have addressed the design of concrete floors for vibration control. It is difficult to change the fundamental resonant frequency. A concrete floor might weigh 200,000 lbs (91,000 kg) and changing the gross physical properties requires major structural changes. Damping, however, is a factor that produces significant results and may be easier to control. These authors make the following preconstruction design considerations:

1. Cross bracing in steel structures has little effect (Moderow, 1970).
2. Noncomposite construction tends to increase damping by 1 to 2% over composite construction (Moderow, 1970).
3. Concrete added to the lower cord of the structural steel can increase damping of a completed floor by 2%.

4. Increasing the thickness of the concrete slab decreases the maximum amplitude and the natural frequency and increases the damping.
5. Cover plates on the joists increase the natural frequency and decrease amplitudes, due to the increased stiffness of the floor. When the data are plotted to determine human response it is found that the change moves downward with frequency, essentially paralleling the human response curve, so little is gained.

After construction, there are still some therapeutic measures available, principally to increase damping. Partitions are very effective in adding damping to an existing structure and can increase the overall damping to 14% of critical. Even lightweight low partitions, planter boxes, and the like can increase damping to 10% of critical. Partitions may be attached to a slab either above or below. Damping posts at critical locations can improve damping somewhat, but they may interfere with the decor. A dynamic absorber can be hung from a floor and can include a damper as part of the design. Allen and Swallow (1975) report that a mass damper tuned to 0.9 of the fundamental frequency and with 10% of critical damping reduced the floor amplitude by 50% and increased floor damping from 3 to 15% of critical. The added mass was 1 percent of the total floor mass.

12 NOISE TRANSMISSION IN FLOOR SYSTEMS

12.1 TYPES OF NOISE TRANSMISSION

The noise and vibration problems encountered in real floor-ceilings generally fall into four categories: airborne, footfall, structural deflection, and floor squeak. Each is a distinct class of problem with unique solutions.

Airborne Noise Isolation

Airborne noise isolation in floors follows the same principles and is tested in the same manner as airborne noise isolation in walls. STC tests are done by placing the noise source in the downstairs room to insure vibrational decoupling between the loudspeakers and the floor-ceiling system being tested.

As was the case with wall transmission, the isolation of airborne noise such as speech is well characterized by the STC rating. The best floor systems combine a high-mass floor slab with a large separation between the floor and ceiling. The two panels should be vibrationally decoupled either by means of a separate structure or by a resilient support. At low frequencies a high structural stiffness is desirable, to minimize the floor deflection. In all cases at least 3" (75mm) of batt insulation should be placed in the air cavity and openings and joints must be sealed air tight.

Footfall

The act of walking across a floor generates noise due to two mechanisms: footfall and structural deflection. Footfall noise is created by the impact of a hard object, such as a heel, striking the surface of a floor. The heel is relatively lightweight and the noise associated with its fall is considered separately from the transfer of weight due to walking. Impact noise can be measured using a standard tapping machine as a source, which leads to an Impact Insulation Class (IIC) rating. The IIC test measures the reaction of a floor system to a series of small hammers dropped from a standard height. Although this may accurately characterize the noise of a heel tap against the floor surface, it does not measure the effect of loading and unloading under the full weight of a walker. Thus the achievement of a particular IIC rating in a given floor-ceiling system does not guarantee that footfall noise will not be a problem, or that the sound of walking will not be audible in the spaces below. The level of impact

noise in the receiving space is primarily dependent on the softness of the floor covering, and is best attenuated using a thick carpet and pad.

Structural Deflection

When a person walks or bounces up and down, a floor will deflect under the static and dynamic load of his weight. Under these conditions the floor acts like a large spring mass system, which responds to a periodic or impulsive force. If the underside of the moving structure is exposed to the room below, the low-frequency sound generated by the movement will radiate directly into the receiving space. Noise created by structural deflection sounds like low-frequency thumps similar to the sound of a very large bass drum, whereas footfall sounds like a high-frequency click. Noise associated with floor deflection is more difficult to correct than footfall noise since the joist system must be stiffened and damped and the ceiling must be physically decoupled from it. Where resilient ceiling supports are used to provide effective vibration isolation, their static deflection must be much greater than the potential structural deflection under the excitation force.

Squeak

Floor squeak is a phenomenon found in wood structures, which is most often caused by the rubbing of nails against wood framing members or metal hangers. It is high pitched and localized to the area around the point of contact. It can occur when there are gaps between the floor diaphragm and the supporting joists and when no glue has been used, or when joists are supported by metal hangers. It can also be exacerbated by the use of wood products having a high glue content, which do not allow the nail to grip the wood.

To control squeak all nonbedded nails (sometimes called shiners) that can rub against a joist or other wood or metal members should be removed and wood diaphragms should be glued to their supporting joists. Shiners must be removed before any concrete topping is poured. When the squeak originates at a joist hanger, it can be caused by inconsistencies in the joist size, and can be treated with small wood shims inserted between the joist and the hanger.

12.2 AIRBORNE NOISE TRANSMISSION

Concrete Floor Slabs

The transmission loss in thick, monolithic concrete floor slabs can be modeled by subtracting 6 dB from the mass law relationship previously developed. In thick panels the shear wave impedance is below the bending impedance so there is no coincidence effect. Typical examples of measured transmission loss data are given in Fig. 10.10. Because of the mass law, we quickly reach a point of diminishing returns if we wish to increase the transmission loss by thickening the slab. A thickness of 8 to 10 inches, which rates around an STC 58, is usually the practical limit for multistory buildings.

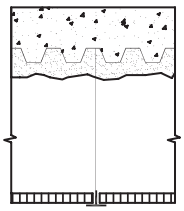
In order to achieve significantly higher STC ratings we must use double panel or other compound construction techniques. In many situations, a concrete floor slab is an excellent choice. Although it may not by itself provide an extremely high STC rating, it has many other advantages. Its low-frequency performance is excellent. If the spans are controlled it can be relatively stiff, and there are no squeak problems.

FIGURE 12.1 Transmission Loss of Metal Deck Floor-Ceilings (National Research Council Canada, 1966)



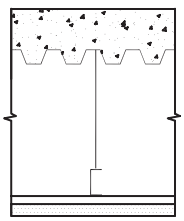
5" Thick Concrete Slab on Metal Decking (42 lbs/sq ft)
3/4" Fireproofing

125	160	200	250	315	400	500	630	800	1 V	1.3 V	1.6 V	2 V	2.5 V	3.2 V	4 V	STC
34	37	38	38	39	41	42	50	54	55	54	54	54	56	60	65	50



5" Thick Concrete Slab on Metal Decking (42 lbs/sq ft)
3/4" Fireproofing
16" Airspace
5/8" Mineral Tile in T-Bar

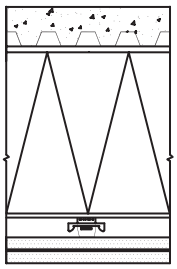
125	160	200	250	315	400	500	630	800	1 V	1.3 V	1.6 V	2 V	2.5 V	3.2 V	4 V	STC
41	47	50	52	53	53	52	62	67	71	72	75	75	76	77	78	60



5" Thick Concrete Slab on Metal Decking (42 lbs/sq ft)
16" Airspace
5/8" Gypsum Board on Resilient Channel
NIOSH, 1980

125	160	200	250	315	400	500	630	800	1 V	1.3 V	1.6 V	2 V	2.5 V	3.2 V	4 V	STC
35	37	39	44	48	50	53	55	58	60	63	62	59	61	65	67	55

(PAC International, 2001)



3.5" Thick Concrete Slab on Metal Decking (39 lbs/sq ft)
9.6" Steel Joists at 24" O.C.
RSIC-1 Neoprene Clips Screwed to Joists at 24" O.C.
2 X 5/8" Gypsum Board on 7/8" Hat Channel
Riverbank, TL01-220

125	160	200	250	315	400	500	630	800	1 V	1.3 V	1.6 V	2 V	2.5 V	3.2 V	4 V	STC
42	42	44	48	52	57	59	61	66	70	73	76	76	78	82	84	60

Concrete on Metal Pans

Concrete poured into a sheet metal pan supported on metal joists can deliver reasonable sound isolation if it is combined with a suspended drywall ceiling. Figure 12.1 gives several examples. The ceiling is supported from hanger wires at 4'-0" (1.2 m) on center that are tied to a 1.5" (38 mm) carrying channel (called black iron). A 7/8" (22 mm) thick hat channel is wire tied or clipped to the black iron and the gypsum board is screwed to it. STC ratings for this construction vary with the airspace depth and the softness of the ceiling support system. Since the slab is relatively stiff, good transmission loss values can be achieved using a neoprene mount, or a neoprene or spring hanger in the ceiling support wires.

Wood Floor Construction

Wood floor construction consists of separate floor and ceiling panels, which may be supported from the same joists or supported separately. Wood construction is lightweight and, if limited to short spans, relatively stiff, compared to long span concrete and steel floors. There is considerable damping in wood structures, so that vibrations do not propagate laterally as easily as in steel framing. Being lightweight, high transmission loss values are achieved only in compound, vibrationally isolated structures. The advantage of isolated construction is that it can approach ideal double panel performance if the floor and ceiling are sufficiently decoupled. Examples of lightweight wood and gypsum board floor-ceiling systems are shown in Fig. 12.2. In several of these the ceiling is attached directly to the underside of the wood joists, so the sound isolation ratings are relatively poor. In others the ceiling is mounted on resilient channels, which results in a modest improvement in STC rating. Resilient channel helps to provide a degree of vibration isolation between the joist system and the ceiling. The preferred type of channel is z-shaped rather than hat-shaped and can be attached only on one side. In order for a hat-shaped channel to be effective, one side of the flange and then the other must be alternately screwed to each joist. Both sides must never be screwed to the same joist. This is an important installation detail, which is rarely implemented correctly in the field, and renders the channel ineffective if not properly done. With all resilient channel, the length of the drywall screws must be controlled so that when the gypsum board is attached to the channel, the screws do not penetrate the joist and short out the isolation.

The third floor in Fig. 12.2 shows an example of adding mass to the underside of the diaphragm. This technique can be used to make improvements to existing construction. It has the advantage of adding mass without adding thickness to the diaphragm and consequently the coincidence frequency of the floor panel is not lowered. The addition of an extra layer of gypsum board on resilient channel over an existing layer is not effective due to the coupling through the air spring between the layers. Improvements of 1 dB or less are the usual result. In cases where there is an existing ceiling and substantial improvement is desired it is most effective to remove the ceiling drywall and add mass, batt insulation, and stepped blocking between the joists before resiliently supporting a double layer gypsum board ceiling.

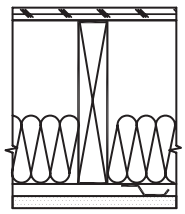
Resiliently Supported Ceilings

Supporting ceilings on resilient mounts can increase the sound transmission class. Resilient channels are one such support system. Working as a spring isolator they rarely achieve a deflection of more than about 1/8" (3 mm). Thus they do not have the softness required to isolate noise due to structural deflection. They are, however, helpful in providing a degree of decoupling of airborne or footfall vibrations transmitted through the structure. In Fig. 12.2 we can see an example in the difference between constructions B and D in the second example.

Where a moderate degree of decoupling is desired, the ceiling can be suspended from neoprene mounts. These can be cut into the hanger wire or screwed directly to the support framing. Test ratings are included in Fig. 12.1 in the last example. This system has the advantage of being able to support multiple layers of drywall in critical applications.

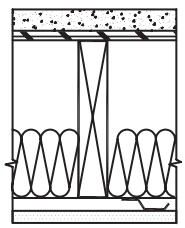
If high STC ratings are desired the ceiling must be independently supported on a separate joist system or suspended from springs, having a deflection of one-inch (25 mm)

FIGURE 12.2 Transmission Loss of Floor-Ceilings (California Office of Noise Control, 1981)



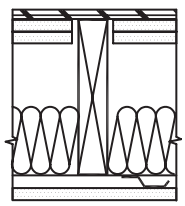
Double Layer Plywood - 1" Thick
2 x 10 Wood Joists
1 Layer 1/2" Gypsum Board
A - No Batt Insulation - No Resilient Channel
B - R-11 Batt Insulation - No Resilient Channel
C - No Insulation - With Resilient Channel
D - R-11 Batt Insulation - With Resilient Channel

	125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
A	23	23	21	27	31	31	31	35	37	41	42	45	44	41	42	45	37
B	20	23	29	32	38	41	44	41	43	45	47	50	51	50	51	54	43
C	27	29	31	33	38	41	43	45	47	48	49	52	54	52	51	54	45
D	27	33	36	35	38	41	44	47	51	51	54	57	59	61	57	60	47



1 1/2" Lightweight Concrete (110 lbs/cu ft)
1 Layer 5/8" Plywood
2 x 10 Wood Joists
1 Layer 1/2" Gypsum Board
A - No R-11 Batt Insulation - No Resilient Channel
B - R-11 Batt Insulation - No Resilient Channel
C - No Insulation - With Resilient Channel
D - R-11 Batt Insulation - With Resilient Channel

	125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
A	26	30	35	36	41	43	46	43	43	47	52	55	57	57	57	60	46
B	28	32	34	37	41	46	50	50	50	51	54	55	57	55	55	58	49
C	36	38	34	38	41	44	51	53	54	57	61	63	64	63	60	62	52
D	40	40	37	41	45	49	51	51	52	54	59	61	63	62	60	63	53

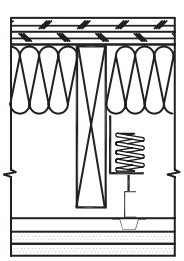


1 Layer 5/8" Plywood
2 x 5/8" Gypsum Board Screwed to the Subfloor
2 x 10 Wood Joists
1 Layer 5/8" Gypsum Board
A - R-11 Batt Insulation - With Resilient Channel

	125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
A	37	40	44	46	51	51	54	56	57	58	63	63	62	58	59	61	56

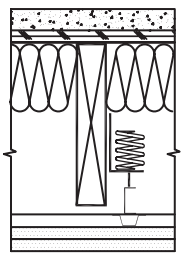
or greater. In these cases the floor and ceiling panels begin to act separately and we gain the advantages of double panel construction, which were discussed in Chapt. 9. With a spring-supported ceiling the transmission loss behavior approaches, but does not reach, the ideal double panel values. In practice, with one-inch deflection isolators, transmission losses are approximately the simple sum of the mass law values above the mass-air-mass resonance or about 6 dB below the ideal behavior. Several examples are given in Fig. 12.3 for suspended gypsum ceilings.

FIGURE 12.3 Transmission Loss of Floors and Spring Supported Ceilings (Kinetics Corporation Test Data)



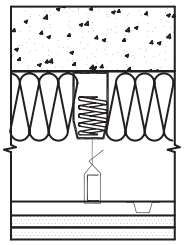
3/4" Thick Hardwood Flooring
3/4" Plywood
2 x 10 Wood Joists
R-11 Batt Insulation
1" Deflection Spring Isolators
2 Layers 5/8" Gypsum Board

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	FSTC
42	39	38	40	44	42	47	48	51	53	55	54	56	59	67	66	51



1 1/2" Lightweight Concrete (110 lbs/cu ft)
1 Layer 5/8" Plywood
2 x 10 Wood Joists
R-11 Batt Insulation
1" Deflection Spring Isolators
2 Layers 5/8" Gypsum Board

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
50	56	58	59	62	70	74	79	84	86	90	94	97	10^0	10^0	95	73



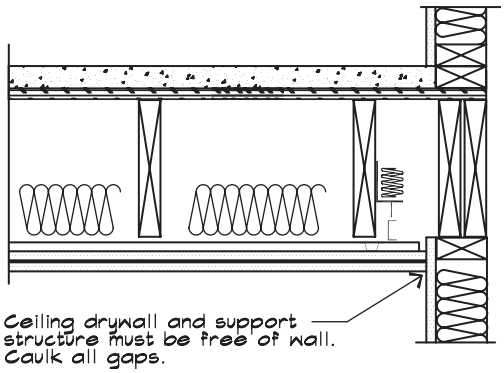
6" Concrete Slab
R-11 Batt Insulation
1" Deflection Spring Isolators
2 Layers 5/8" Gypsum Board

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
63	64	66	72	73	83	84	86	91	92	96	10^4	10^4	10^5	10^5	10^5	84

A number of details are important to the successful installation of hanger-supported ceilings. When sheets of drywall are applied to a ceiling, supported from a series of springs, the weight due to each additional layer will cause the ceiling to drop. As it drops it is important that neither the drywall nor the hat or carrying channel above it rests on the side wall structure or gypsum board. If it does an uneven load distribution will result and the ceiling will bow. The ceiling should be free to move so that the isolators can work effectively. This is accomplished most easily by building the ceiling within the confines of the side wall finish material as in Fig. 12.4. Joints between the two may then be caulked and molding can be applied slightly below the finish ceiling.

The second detail has to do with the load carried by each spring hanger. Typically hangers are located at 4'-0" (1.2 m) on center so that each spring supports 16 sq. ft. (1.5 sq m) of ceiling material. For a ceiling constructed of double 5/8" (16 mm) drywall the total weight along with the support framing works out to be about 100 lbs (45 kg) per isolator. When a spring is located near a wall it may support as little as half the load of a center

FIGURE 12.4 Resiliently Suspended Ceiling Detail



spring and when it is in a corner, as little as one-quarter. If a hanger supports a vertical portion of a soffit and therefore more than its normal load, the stiffness of each spring must be matched to the load it carries so that the deflection across the ceiling is uniform. This is done by calculating the load on each hanger and by having springs of varying stiffness (usually color-coded) available at the construction site to insert into a hanger. The deflection can also be adjusted by means of a threaded cap screw on top of each spring. This is often required in corner springs supporting more than a quarter load.

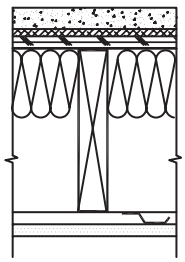
Floating Floors

Floating floors, which are resiliently supported panels located above the structural system, can be used to attenuate vertical as well as lateral noise and vibration transmission. They are usually heavier than resiliently hung ceilings and thus have the attendant advantages of mass. This is offset somewhat by the narrow air gap and low deflections inherent in the neoprene or pressed fiberglass isolators that normally are employed. Where low-deflection (< 0.1" or 3 mm) continuous support systems such as sheets of neoprene, pressed fiberglass board, pressed paper, or wire mesh materials are used, the degree of decoupling is much less, due to the low deflection as well as the additional stiffness attributable to the trapped air. Some of these materials can become overloaded and crush over time, further reducing their effectiveness. Any concrete, which flows into the space beneath the floating floor and shorts out the resilient support, also severely reduces its effectiveness.

Since the floating floor supports are acting as vibration isolators it is desirable to reduce their natural frequency by maximizing the static deflection under load. Consequently a grid of regularly spaced individual mounts is much more effective than a continuous material since the loading on each isolator is much greater.

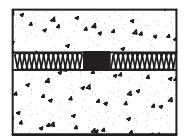
Where very high transmission loss values are required, such as in the construction of sound studios, floating floors in combination with resiliently supported ceilings can yield very good results. Figure 12.5 shows an example of a continuous floor support system and two point-mounted floor systems. Note that the floating floor is most effective if it is heavy and point mounted. The weight is important for several reasons. First, it provides the additional mass for sound attenuation. Second, it yields good isolator deflection without softening the floating floor. A high stiffness in the floating floor is important since it should not deflect appreciably between the mounts. Lightweight floating floors, such as those used

FIGURE 12.5 Transmission Loss of Floating Floors (Kinetics Corporation Test Data)



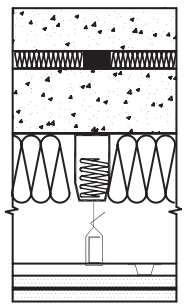
1 1/2" Lightweight Concrete (110 lbs/cu ft)
5/8" Continuous Resilient Underlayerment
1 Layer 3/4" Wood Subfloor
2 x 10 Wood Joists
R-11 Batt Insulation
Resilient Channel
1 Layer 5/8" Gypsum Board

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	FSTC
37	35	42	45	47	49	52	52	54	55	59	60	61	60	64	68	54



4" Concrete Slab (144 lbs/cu ft)
2" Thick Pressed Fiberglass Isolators at 16" on Center
R-11 Batt Insulation
6" Concrete Structural Slab (144 lbs/cu ft)

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
53	54	55	57	63	70	72	78	83	86	91	93	98	10^{1}	10^{1}	10^{0}	72



4" Concrete Slab (144 lbs/cu ft)
2" Thick Pressed Fiberglass Isolators at 16" on Center
R-11 Batt Insulation
6" Concrete Structural Slab (144 lbs/cu ft)
R-11 Batt Insulation
1" Deflection Spring Isolators
2 Layers 5/8" Gypsum Board

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
71	74	81	84	86	96	99	10^{1}	10^{1}	10^{0}	10^{0}	10^{0}	10^{0}	10^{0}	10^{1}	10^{1}	94

as gymnasium floors, tend to be springy and walkers can perceive a noticeable movement. Thus concrete floating floors are preferred for residential and studio applications.

12.3 FOOTFALL NOISE

Impact Insulation Class—IIC

The Impact Insulation Class (IIC) is a laboratory rating much like the Sound Transmission Class; however, it represents the isolation provided by a floor system subjected to a controlled impulsive load. Since there is no standard footstep, the impulsive loads are generated by a tapping machine pictured in Fig. 12.6. The machine consists of a frame supporting a row of five cylindrical hammers, each weighing a half-kilogram (1.1 lbs), which are raised by a cam mechanism and dropped sequentially from a height of 4 cm (1.6 in) onto the surface of the floor. The cam is driven by an electric motor that is set to deliver 10 impacts per second at equal intervals.

There are test standards in the United States and Europe that regulate the laboratory (ASTM E 492 and ISO 140/6) as well as field (ASTM E 1007 and ISO 140/7) test methodologies. The test is performed by placing the tapping machine near the center of the

FIGURE 12.6 Tapping Machine Showing Inner Workings

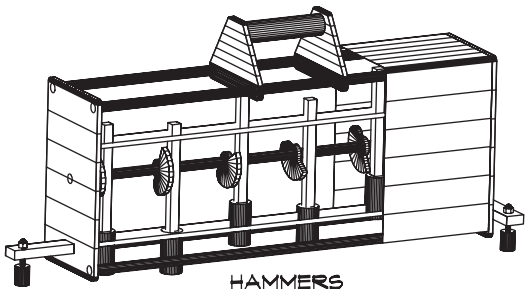
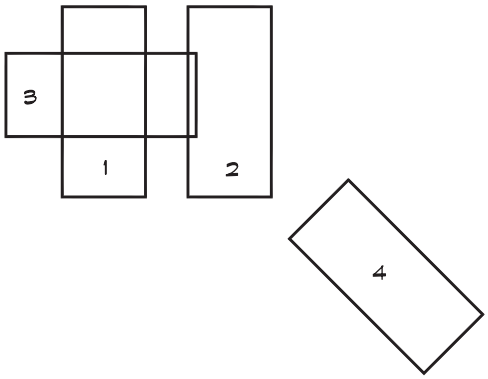


FIGURE 12.7 IIC Tapping Machine Positions



floor under test. Spatially averaged sound pressure levels are then measured in the room below in third-octave bands ranging from 100 through 3150 Hz. The readings are done for four specified tapping machine positions illustrated in Fig. 12.7. A normalized impact sound pressure level in the receiving room is then obtained from the spatial average sound pressure levels

$$L_n = \overline{L}_p - 10 \log (A_0 / R) \tag{12.1}$$

The absorption in the receiving room is measured either by using the reverberant field approximation (Eq. 12.2) and a source of known sound power, or by measuring the reverberation time, from which the total absorption is obtained using the Sabine equation.

$$\overline{L}_p \cong L_w + 10 \log (4 / R) + K \tag{12.2}$$

where $\overline{L}_p$ = average one - third octave sound pressure level measured in the receiving room (dB)

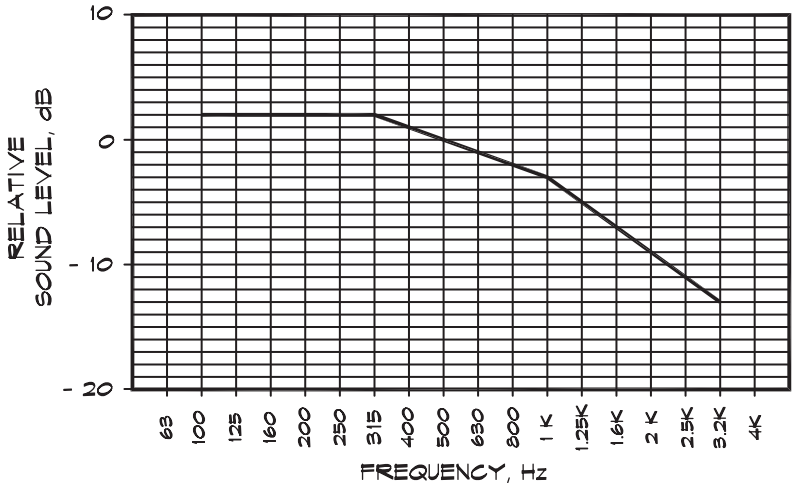
L_w = one - third octave sound power level of the reference source (dB)

R = sound absorption in the receiving room (m^2 or ft^2)

A_0 = reference absorption in the same units as R (either 10 metric sabins or 108 sabins)

K = 0.1 for SI units or 10.5 for FP units

FIGURE 12.8 Reference Contour for Calculating Impact Insulation Class and Other Ratings (ASTM E989, 1989)



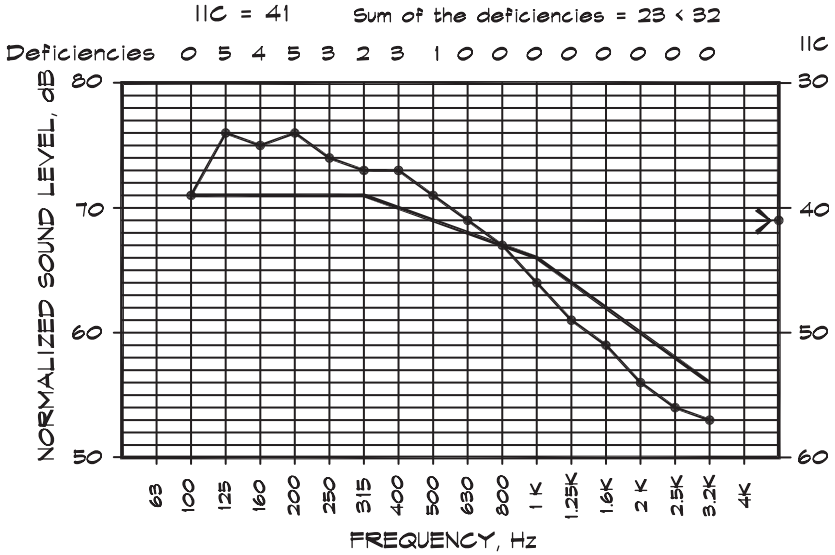
Note that $\overline{L}_p$ in Eq. 12.1 is measured using the tapping machine as the noise source, whereas in Eq. 12.2 a standard reference noise source such as a calibrated fan or loudspeaker is used.

Once the normalized levels have been obtained, they are compared to the standard IIC curve (ASTM E 989) in Fig. 12.8 by taking the deficiencies (differences) at each third-octave frequency band. The standard curve is shifted vertically relative to the test data until two conditions are fulfilled: 1) the sum of the deficiencies above the contour does not exceed 32 dB, and 2) the maximum deficiency at a single frequency band does not exceed 8 dB. Once these criteria are satisfied, the normalized sound pressure level at the intersection of the standard curve and the 500 Hz ordinate is subtracted from 110 to obtain the Impact Insulation Class. A typical example is given in Fig. 12.9.

Field tests of the IIC rating also may be carried out after a building has been constructed. These are designated FIIC, and like the FSTC tests fall about five points below the laboratory test for the same construction. They apply only to the room in which they are measured and are not generally applicable to a type of construction. Test standards set minimum limits on the volume of the receiving space at each third-octave frequency. Receiving rooms are required to meet minimum volume requirements such that there are at least 10 room modes at 125 Hz and the same modal spacing at 100 and 160 Hz. Minimum room volumes are 2100 cu ft (60 cu m) at 100 Hz, 1400 cu ft (40 cu m) at 125 Hz, and 800 cu ft (25 cu m) at 160 Hz. If room volumes fall below these limits the field report must include a notation to that effect.

The State of California has modified the standard FSTC and FIIC procedures to make them less strict by dropping the room constant term. In the revised code only the raw receiving levels are used to compute $L_n \cong \overline{L}_p$. Nonnormalized field tests generally produce ratings that are 3 to 5 dB higher than tests properly done, in accordance with the accepted ASTM standards. They yield an FIIC rating that is not normalized to the absorption in the receiving room and thus may vary from room to room or for the same room if the receiving space contains different amounts of furniture or other absorptive materials. Field tests made under these requirements are representative only of the rooms and conditions under which they

FIGURE 12.9 Calculation of an IIC Rating



were measured and are not generally applicable to other rooms, even those having the same nominal construction.

Impact Insulation Class Ratings

The IIC rating reflects the softness of the floor covering including any resilient support system. IIC ratings are shown for various floor coverings in Fig. 12.10 for a concrete slab and in Figs. 12.11 and 12.12 for wood floor-ceiling systems. The Uniform Building Code (UBC) and the State of California set minimum standards of 50 IIC (laboratory) and 45 FIIC (field) ratings in multifamily dwellings. At this rating footfall noise from a person walking on a floor above is clearly audible, and it is possible to follow the progress of the walker around the room. Thus these building code standards do not represent good building practice. Rather they represent minimums below which it is illegal to build. It is only when the ratings are above an IIC 65 that heel clicks begin to become inaudible (Kopec, 1990).

Vibrationally Induced Noise

To construct a theoretical model of vibration transmitted through floor systems we can assume that a mechanical force is applied to one or more points on the floor, which induces a motion in the ceiling below. Clearly if the ceiling vibrates, an airborne sound will be radiated. Assuming that the ceiling is a flat plate moving vertically, the intensity near its surface is the same as that radiated by a plane wave.

$$W_{rad} = I\,S = \frac{p^2}{\rho_0\, c_0} S \tag{12.3}$$

The radiated acoustic power can be written in terms of the surface velocity and a radiation efficiency, which is an empirical constant expressing the ratio of the actual power emitted

FIGURE 12.10 Impact Insulation Class of Concrete Floors (Kinetics Corporation Test Data)

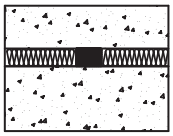


6" Concrete Slab (144 lbs/cu ft)

A 1/2" Wood Fiber Board glued to Concrete
44 oz Carpet on 40 oz Pad

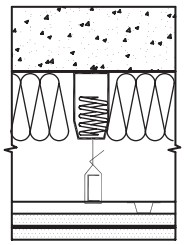
B No Floor Covering

	100	125	160	200	250	315	400	500	630	800	1 ⅓	1.3 ⅓	1.6 ⅓	2 ⅓	2.5 ⅓	3.2 ⅓	IIC
A	39	36	34	28	29	29	25	18	14	11	0	0	0	0	0	0	81
B	62	63	67	73	71	74	73	74	74	75	74	74	73	73	73	72	28



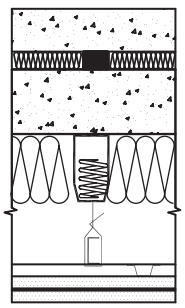
4" Concrete Slab (144 lbs/cu ft)
2" Thick Pressed Fiberglass Isolators at 16" on Center
R-11 Batt Insulation
6" Concrete Structural Slab (144 lbs/cu ft)
R-11 Batt Insulation
No Floor Covering

100	125	160	200	250	315	400	500	630	800	1 ⅓	1.3 ⅓	1.6 ⅓	2 ⅓	2.5 ⅓	3.2 ⅓	IIC
50	52	53	60	58	52	48	47	46	45	45	41	40	41	34	36	60



6" Concrete Structural Slab (144 lbs/cu ft)
R-11 Batt Insulation
1" Deflection Spring Isolators
2 Layers 5/8" Gypsum Board
No Floor Covering

100	125	160	200	250	315	400	500	630	800	1 ⅓	1.3 ⅓	1.6 ⅓	2 ⅓	2.5 ⅓	3.2 ⅓	IIC
42	42	43	44	41	45	37	39	38	39	41	36	29	29	32	30	70



4" Concrete Slab (144 lbs/cu ft)
2" Thick Pressed Fiberglass Isolators at 16" on Center
R-11 Batt Insulation
6" Concrete Structural Slab (144 lbs/cu ft)
R-11 Batt Insulation
1" Deflection Spring Isolators
2 Layers 5/8" Gypsum Board
No Floor Covering

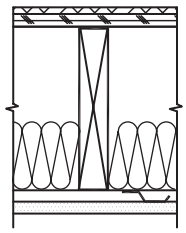
100	125	160	200	250	315	400	500	630	800	1 ⅓	1.3 ⅓	1.6 ⅓	2 ⅓	2.5 ⅓	3.2 ⅓	IIC
38	37	27	27	27	26	24	16	15	13	12	13	7	4	4	4	82

by a source compared with that emitted by an ideal radiating surface.

$$W_{rad} = \sigma\, u^2\, \rho_0\, c_0\, S \tag{12.4}$$

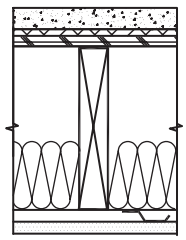
where W_{rad} = radiated sound power, (W)
σ = radiation efficiency (usually ≤ 1)
u = rms velocity of the radiating surface, (m / s)

FIGURE 12.11 Impact Insulation Class of Wood Framed Floors (California Office of Noise Control, 1981)



1/4" Particle Board Glued
5/8" Plywood Glued and Nailed
2 x 10 Wood Joists
R-11 Batt Insulation
1 Layer 1/2" Gypsum Board on Resilient Channel
A - 76 oz Carpet on 50 oz Hair Pad
B - 65 oz Carpet on 30 oz Foam Rubber Pad
C - 50 oz Carpet on 24 oz Hair Pad
D - Cushioned Vinyl Tile
E - 1/16" Vinyl Asbestos Tile
F - 1/2" Wood Parquet Flooring
G - No Floor Covering

	100	125	160	200	250	315	400	500	630	800	1 V	1.3 V	1.6 V	2 V	2.5 V	3.2 V	11C
A	50	43	42	37	34	34	33	30	31	31	32	29	0	0	0	0	70
B	49	42	39	35	34	32	32	28	28	29	29	0	0	0	0	0	71
C	62	57	56	52	47	42	38	33	34	34	34	32	0	0	0	0	58
D	68	66	68	67	65	61	53	47	45	44	41	37	33	34	0	0	51
E	69	65	69	69	69	69	67	65	64	64	60	56	50	48	48	48	47
F	65	64	64	65	64	63	63	60	60	59	57	55	49	45	45	49	50
G	66	64	68	67	66	67	65	62	62	61	58	54	49	47	46	47	49



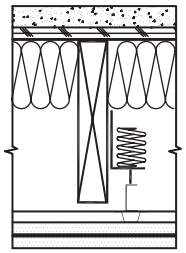
1 5/8" Lightweight Concrete (110 lbs/cu ft)
1/4" Particle Board Glued to Plywood
5/8" Plywood Glued and Nailed to Joists
2 x 10 Wood Joists
R-11 Batt Insulation
1 Layer 1/2" Gypsum Board on Resilient Channel
A - 76 oz Carpet on 50 oz Hair Pad
B - 65 oz Carpet on 30 oz Foam Rubber Pad
C - 50 oz Carpet on 24 oz Hair Pad
D - Cushioned Vinyl Tile
E - 1/16" Vinyl Asbestos Tile
F - 1/2" Wood Parquet Flooring
G - No Floor Covering

	100	125	160	200	250	315	400	500	630	800	1 V	1.3 V	1.6 V	2 V	2.5 V	3.2 V	11C
A	47	42	39	34	34	29	30	26	25	27	28	23	22	21	0	0	73
B	44	38	35	31	32	27	28	24	24	25	27	22	19	19	0	0	76
C	56	54	50	46	43	37	35	31	30	31	32	29	0	0	0	0	64
D	64	61	62	60	58	52	48	40	36	37	39	34	33	34	0	0	56
E	62	59	60	60	61	60	62	62	62	65	66	64	62	60	60	55	43
F	64	61	63	63	63	60	61	60	59	58	55	50	46	42	40	40	52
G	62	61	60	60	61	59	62	63	63	65	66	65	62	61	61	62	38

ρ_0 = density of air, (kg/m^3)
c_0 = speed of sound in air, (m/s)
S = area of the radiating surface, (m^2)

Radiation efficiencies are helpful in describing the process since real sources do not move in a perfectly planar manner and are not infinite. Sound radiation by heavy thick materials such as brick or concrete, where shear waves predominate at high frequencies, usually can be

FIGURE 12.12 Impact Insulation Class of Floors with Resiliently Supported Ceilings

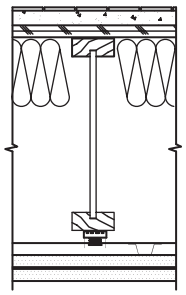


(Kinetics Corporation Test Data)

1 1/2" (40 mm) Lightweight Concrete
(12.8 lbs/sq ft or 63.4 kg/sq m)
1 Layer 5/8" (16 mm) Plywood
2 x 10 (240 mm) Wood Joists
R-11 Batt Insulation
1" (25 mm) Deflection Spring Isolators
2 Layers 5/8" (16 mm) Gypsum Board
A - Carpet on 3/8" (10 mm) Hair Pad
B - 5/16" (8 mm) Parquet Flooring

	100	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	IIC
A	31	29	27	22	19	16	14	10	10	6	6	5	6	7	7	9	89
B	56	56	53	53	55	54	50	48	46	44	43	40	36	34	31	24	62

(PAC International Test Data)



1/4" (13 mm) Ceramic Tile
1" (25 mm) Lightweight Concrete
(130 lb/cu ft or 2,083 kg/cu m)
1 Layer 3/4" (18 mm) Plywood
18" (457 mm) Truss Wood Joist
R-11 Batt Insulation
RSIC-1 Isolation Clips with 7/8" (22 mm) Hat Channel
2 Layers 5/8" (16 mm) Gypsum Board
A - As Shown
B - 3/8" (10 mm) Carpet on 3/8" (10 mm) Pad on Tile
with 1 x 5/8" (16 mm) Gypsum Board Ceiling

	100	125	160	200	250	315	400	500	630	800	1	1.3	1.6	2	2.5	3.2	IIC
A	64	55	57	57	54	52	53	54	57	55	53	51	50	52	50	45	53
B	46	32	32	28	28	29	22	13	9	8	8	5	5	5	4	4	74

assumed to have a radiation efficiency of one. If bending waves are part of the transmission mechanism they result in a pronounced increase in the radiation efficiency around the critical frequency and a decrease below it, as shown in Fig. 12.13. For finite-sized panels there are also contributions due to edge effects, so that there are additional terms contributing to Eq. 12.4.

Mechanical Impedance of a Spring Mass System

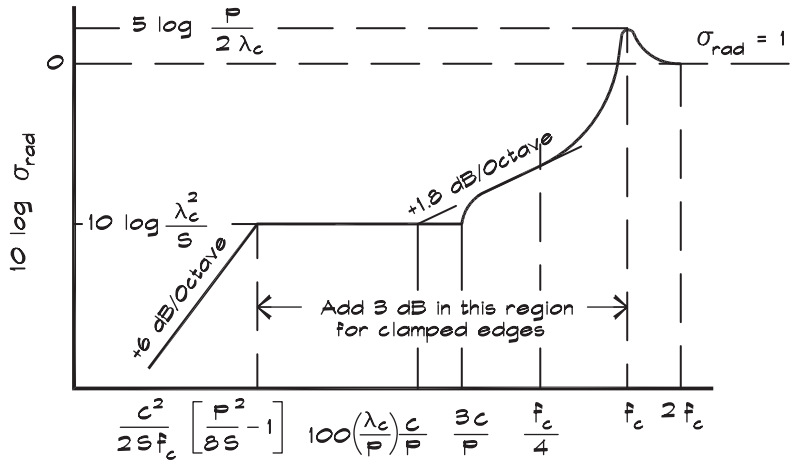
A vibrationally driven system such as a floor can be analyzed in terms of its mechanical impedance, which is the ratio of an applied force to the induced velocity. This is somewhat different from the specific acoustic impedance in which the force was expressed as a pressure. The mechanical or driving point impedance is given by

$$\mathbf{z}_m = \frac{\mathbf{F}}{\mathbf{u}} \tag{12.5}$$

The model of vibrational point excitation of structures is analogous to the use of point sources in the study of sound propagation. They allow us to deal with complex force distributions by integrating (summing) over a distribution of point forces. Point impedances can be measured

FIGURE 12.13 Design Curve for Approximating the Radiation Efficiency (Beranek, 1971)

Design curve for a finite panel having perimeter P and area S with simply supported or clamped edges. (Note that $\lambda_c = c/f_c$ = wavelength in the panel or in air at the critical frequency.)



in the laboratory under controlled conditions and are a useful tool in characterizing complex systems.

If a force is applied to a simple spring mass system the mechanical impedance is the sum of the impedances of the individual components, namely the mass, spring, and damping.

$$\mathbf{z}_{\mathrm{m}} = c + \mathrm{j}\,\omega\,\mathrm{m} + \frac{k}{\mathrm{j}\,\omega} \tag{12.6}$$

Note that at resonance the mechanical impedance is zero, since a large velocity is produced by a very small force. If we set the impedance to zero and solve Eq. 12.6 for the resonant frequency, we get Eq. 6.4 for zero damping.

For a sinusoidal force $\mathbf{F} = \mathrm{F}_0\,\mathrm{e}^{\mathrm{j}\,\omega\,\mathrm{t}}$ applied to a spring-mass system, the induced velocity is

$$\mathbf{u} = \frac{\mathrm{F}_0\,\mathrm{e}^{\mathrm{j}\,\omega\,\mathrm{t}}}{\mathbf{z}_{\mathrm{m}}} = \frac{\mathrm{F}_0\,\mathrm{e}^{\mathrm{j}\,\omega\,\mathrm{t}}}{c + \mathrm{j}\,\omega\,\mathrm{m} + \dfrac{k}{\mathrm{j}\,\omega}} \tag{12.7}$$

It is clear that for low induced velocities we want high mass, high stiffness, and high damping. Although floors are not pure spring-mass systems, the model is a helpful analogy. An examination of Eq. 12.7 reveals that at high frequencies the mass and damping terms predominate, while at very low frequencies the stiffness and damping terms are more important.

In a generalized system having a complex impedance the power is

$$W = \frac{\omega}{2\pi} \int_0^{2\pi/\omega} \mathbf{F}(t)\, \mathbf{u}(t)\, dt = \frac{1}{2} \left|F_0\right| \left|u_0\right| \cos\phi \tag{12.8}$$

so

$$W = \frac{1}{2} \left|F_0\right|^2 \mathrm{Re}\left\{\frac{1}{\mathbf{z}_m}\right\} = \frac{1}{2} \left|u_0\right|^2 \mathrm{Re}\left\{\mathbf{z}_m\right\} \tag{12.9}$$

where F_0 is the peak force amplitude, u_0 is the peak velocity amplitude, and ϕ is the relative phase between the force and the velocity. The bracketed terms are the real parts of the complex impedance or its reciprocal. If a sinusoidal force is applied to a spring mass system the steady state energy is dissipated in the dashpot. The dashpot impedance c is real and the velocity of the mass is in phase with the resistance force. The power expended is

$$W = \frac{\left|F_0^2\right|}{2c} \tag{12.10}$$

Spring mass models are useful abstractions when the structural wavelengths are large compared to the dimensions of the system (Ver, 1992). When the wavelengths are small compared to the dimensions of the floor, we use another approximation, the driving point impedance. In this model we assume that the panel is infinite so we can ignore the reflection of structural waves from the plate boundaries. We then use the infinite plate model to approximate the result for a finite structure.

Driving Point Impedance

The driving point impedance of a structural system can be measured directly or calculated from its mass and bending stiffness. This exercise has been carried out by a number of authors. Results have been published by Cremer (1973), Pinnington (1988), and Beranek and Ver (1992). The structural configuration that is generally of the greatest interest in architectural acoustics is a flat plate. The point impedance of an infinite thin plate in bending is

$$z_m = 8\sqrt{D\,\rho_m\,h} \cong 2.3\, c_L\, \rho_m\, h^2 \tag{12.11}$$

where D = flexural rigidity per unit length $= \dfrac{E\,h^3}{12\,(1-\mu^2)}$ (N m)

ρ_m = density of the plate (kg / m^3)

c_L = longitudinal sound velocity in the plate $= \sqrt{\dfrac{E}{\rho_m}}$ (m / s)

h = plate thickness (m)

μ = Poisson's ratio $\cong 0.3$

E = Young's modulus (N / m^2)

Cremer (1973, pp. 260–264) gives the derivation of this relationship.

Power Transmitted through a Plate

For vibrational energy transmitted into and out of a plate, there is an energy balance whereby the energy entering the plate is either dissipated within the plate or radiated away as sound.

$$W_{in} = W_{dis} + W_{rad} \tag{12.12}$$

There are several possible energy dissipation mechanisms. The energy may be transmitted away as a bending wave in an infinite plate, or in a finite plate it may end up exciting the normal modes of vibration, which decay out due to internal losses. In both cases we can assume there is internal damping, which attenuates energy by some amount over time.

$$E_{dis}\,(t) = E_0\,(1 - e^{-\eta\,\omega\,t}) \tag{12.13}$$

where $E_{dis}\,(t)$ = energy dissipated as a function of time (W)
E_0 = initial vibrational energy (W)
η = damping constant
ω = radial frequency (s^{-1})
t = time (s)

We can calculate the energy lost in one period ($T = 2\pi / \omega$) assuming a damping constant much less than one

$$E_{dis} \cong 2\pi\,\eta\,E_0 \tag{12.14}$$

Now we examine a small element of plate, which is vibrating. The initial energy in that element is

$$E_0 = \frac{1}{2}\rho_s\,u_0^2\,dx\,dz \tag{12.15}$$

where $\rho_s = \rho_m\,h$ is the surface density of the plate. Dividing the energy by the period $T = 1 / f$, we obtain the energy dissipated per unit time

$$E_{dis} = \frac{1}{2}\,\omega\,\eta\,\rho_s\,u_0^2\,dx\,dz \tag{12.16}$$

so the power converted to heat for a total plate area S is

$$W_{dis} = \frac{1}{2}\,\omega\,\eta \int_S \rho_s\,u_0^2\,dx\,dz = \frac{1}{2}\,\omega\,\eta\,\rho_s\,S\,u_0^2 \tag{12.17}$$

Therefore the power balance equation can be written as

$$\frac{1}{2}\,F_0^2\,Re\left\{\frac{1}{\mathbf{z}_m}\right\} = u^2\,S\left[\omega\,\eta\,\rho_s + 2\,\rho_0\,c_0\,\sigma\right] \tag{12.18}$$

and using the mechanical impedance given in Eq. 12.11, the mean square velocity in a thin infinite plate that results from an imposed force having an amplitude F_0 is

$$u^2 = \frac{F_0^2}{4.6\,\rho_S^2\,c_L\,h\,\omega\,\eta\,S\,(1 + 2\,\rho_0\,c_0\,\sigma/\omega\,\eta\,\rho_s)} \tag{12.19}$$

When $\omega\,\eta\,\rho_s >> 2\,\rho_0\,c_0\,\sigma$, Eq. 12.19 simplifies to

$$u^2 \cong \frac{F_0^2}{4.6\,\rho_S^2\,c_L\,h\,\omega\,\eta\,S} \tag{12.20}$$

Thus the transmission mechanism for impact noise in a thin plate is primarily due to bending waves.

Impact Generated Noise

When a standard tapping machine is used to test the Impact Insulation Class of a floor, a series of blows is created by releasing hammers timed to fall on the floor 10 times per second. This generates a train of force pulses that can be analyzed in terms of a Fourier series (Ver, 1971). The reason we use this methodology is that we wish to be able to calculate noise generated in each frequency band and thus we must determine the vibrational frequency spectrum of the exciting force. The Fourier series describes any repeated wave form using an infinite sum having the form

$$F(t) = \sum_{n=1}^{\infty} F_n \cos(n\,\omega\,t) \tag{12.21}$$

where the amplitudes are given by

$$F_n = \frac{2}{T_r}\int_0^{T_r} f(t)\,\cos\left(\frac{2\,\pi\,n}{T_r}\,t\right)dt \tag{12.22}$$

and f(t) is the shape of a typical force pulse, T_r is the time period between hammer blows, and $n = 1, 2, 3\ldots$ The duration of the force impulse for a hard concrete slab is small compared to the period of the highest frequency of interest. Thus the term

$$\cos\left(\frac{2\,\pi\,n}{T_r}\,t\right) \cong 1 \tag{12.23}$$

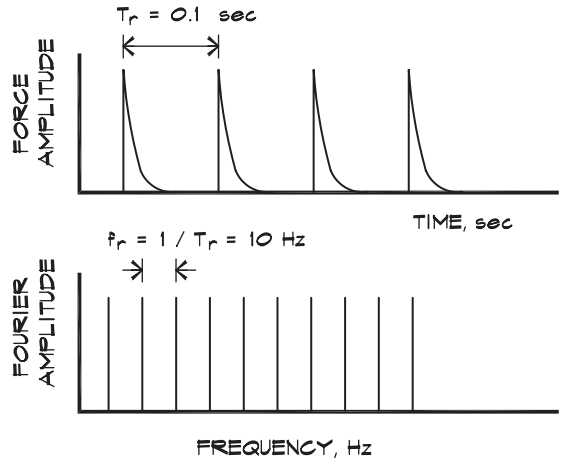
and the Fourier amplitude of the pulse train can be simplified to

$$F_n \cong \frac{2}{T_r}\int_0^{T_r} f(t)\,dt = 2\,f_r\,m\,v_0 = 2\,f_r\,m\sqrt{2\,g\,d} \tag{12.24}$$

where f_r = repetition frequency, (Hz)
m = mass of the hammer, (kg)
v_0 = velocity of the hammer at impact, (m / s)
d = drop distance, (m)
g = gravitational acceleration (9.8 m / s^2)

Figure 12.14 shows a pulse train and its spectrum. The frequency spectrum is a series of lines of the same length that are separated by the frequency interval f_r.

FIGURE 12.14 Time Dependence and Frequency Spectrum of Tapping Machine Noise (Cremer and Heckl, 1973)



This relationship holds for ideal impacts. For real impacts, which have a finite duration, the length of the lines decreases slowly at high frequencies (Lange, 1953).

To determine the force spectrum at a given frequency, we define a mean square force spectrum density, S_f, which when multiplied by the bandwidth yields the mean square force in the same bandwidth (Ver, 1971 and Cremer, 1973)

$$S_f = \frac{1}{2} T_r F_n^2 = 4 f_r m^2 g d \quad (N^2 / Hz) \tag{12.25}$$

For a standard tapping machine the value of S_f can be calculated from the fixed mass, the drop frequency, and the drop distance to be 4 N^2 / Hz. The mean square force in a standard octave band is the spectrum density times the octave bandwidth or

$$F_{rms}^2 \text{ (oct)} = 4 \frac{f}{\sqrt{2}} \quad (N^2) \tag{12.26}$$

Using Eqs. 12.3 and 12.18 we can calculate the sound power level generated by a tapping machine in a given frequency range

$$L_w \text{ (oct)} = 10 \log \left(\frac{\rho_0 c_0 \sigma_{rad}}{5.1 \rho_m^2 c_L \eta h^3} \right) + 120 \tag{12.27}$$

and using Eqs. 12.1 and 8.83, the normalized impact sound level in each octave band is

$$L_n \text{ (oct)} = 10 \log \left(\frac{4}{5.1} \frac{(\rho_0 c_0)^2 \sigma_{rad}}{p_{ref}^2 A_0 \rho_m^2 c_L \eta h^3} \right) \tag{12.28}$$

where $p_{ref} = 2 \text{ x } 10^{-5}$ (N / m^2) = 0.0002 μ bar. Note that in Eq. 12.28, the transmitted sound level follows the normal mass law, increasing 6 dB per doubling of density; however, it decreases 9 dB per doubling of slab thickness. The normalized level, L_n, decreases with

increasing damping and is independent of the frequency, as long as the radiation efficiency and damping are frequency independent. To calculate the spatial average diffuse field sound level in the room below the tapping machine we use

$$\overline{L}_p = L_n\,(\text{oct}) + 10 \log \frac{A_o}{S\,\overline{\alpha}} \tag{12.29}$$

To estimate the sound levels radiated by structural concrete or lightweight concrete floors we can use the material constants for the propagation speed of longitudinal waves and the characteristic density, which are: 1) for dense concrete $\rho_m = 2.3 \times 10^3$ (kg / m^3) and $c_L = 3.4 \times 10^3$ (m/s), and 2) for lightweight concrete $\rho_m = 6 \times 10^2$ (kg / m^3) and $c_L = 1.7 \times 10^3$ (m/s). Using these constants the expected levels for dense concrete are

$$L_n\,(\text{oct}) = 32.5 - 30 \log h_m + 10 \log (\sigma_{rad} / \eta) \tag{12.30}$$

or

$$L_n\,(\text{oct}) = 80.5 - 30 \log h_{in} + 10 \log (\sigma_{rad} / \eta) \tag{12.31}$$

and for lightweight concrete

$$L_n\,(\text{oct}) = 47 - 30 \log h_m + 10 \log (\sigma_{rad} / \eta) \tag{12.32}$$

or

$$L_n\,(\text{oct}) = 95 - 30 \log h_{in} + 10 \log (\sigma_{rad} / \eta) \tag{12.33}$$

where h_m is the thickness of the floor slab in meters and h_{in} is the thickness in inches.

Ver (1971) published a plot of the calculated levels, which is reproduced as Fig. 12.15. The quantity $L_n\,(\text{oct}) + 10 \log (\eta/\sigma_{rad})$ is shown on the left-hand scale as a function of slab thickness. On the right-hand scale L_n (oct) is given for the typical values, $\eta = 0.01$

FIGURE 12.15 Normalized Impact Sound Level as a Function of Slab Thickness for Lightweight and Dense Concrete Floor Slabs (Ver, 1971)

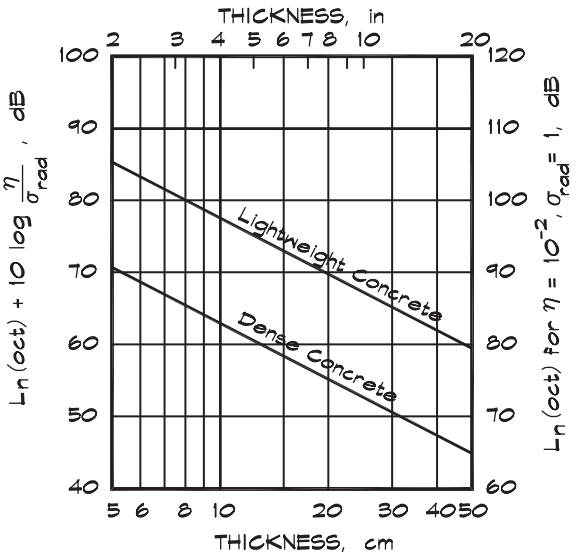
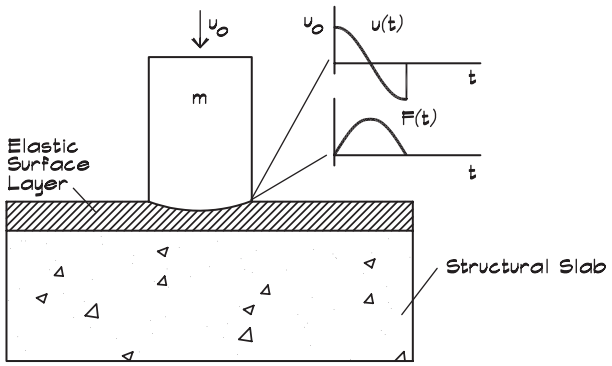


FIGURE 12.16 Forces and Velocities for Soft Floor Covering (Ver, 1971)



and $\sigma_{rad} = 1$. If we assume a 15 cm (6 in) dense concrete slab and use the figure, we get about 78 dB, and for a lightweight concrete structure of the same thickness we get about 92 dB with no surface covering. Human heel drops are not this loud, but the numbers are close to the measured levels for a tapping machine test (Fig. 12.18). Since it is usually impractical to increase the slab thickness and density enough to make a significant change, we turn to the floor surface covering for improvement.

Improvement Due to Soft Surfaces

Ver (1971) and Cremer (1973) have analyzed the impact of a carpet or other similar elastic surfaces on tapping machine noise transmitted through a floor. An illustration of Ver's model is given in Fig. 12.16.

The falling weight strikes a surface, whose stiffness is the elasticity of the carpet. In this model damping is ignored and the weight is assumed to strike the surface and recoil elastically once without multiple bounces. The equation of motion of the spring mass system is

$$m \frac{d^2 x}{d t^2} - k x = 0 \tag{12.34}$$

and

$$m \frac{d u}{d t} + \frac{k}{j \omega_0} u = 0 \tag{12.35}$$

where the natural frequency of the spring mass system is

$$\omega_n = \sqrt{\frac{k}{m}} \tag{12.36}$$

The spring constant is given by

$$k = \frac{E_d S_h}{h} \tag{12.37}$$

where E_d = dynamic Young's modulus of elasticity (N/m^2)
which is about twice the static modulus
S_h = area of the striking surface of the hammer = 0.0007 m^2
h = thickness of the elastic layer (m)
m = mass of the hammer = 0.5 kg

When the hammer is dropped it strikes the elastic layer with a velocity u_0 at time t = 0 and its subsequent motion can be calculated from Eq. 12.34 to be

$$u(t) = u_0 \cos(\omega_n t) \qquad \text{for } 0 < t < \pi/\omega_n \tag{12.38}$$

and

$$u(t) = 0 \qquad \text{for } t < 0 \text{ and } t > \pi/\omega_n \tag{12.39}$$

Figure 12.16 also shows this velocity function. The force is given by

$$F(t) = m\frac{du}{dt} = u_0\,\omega_n\, m \sin(\omega_n t) \quad \text{for } 0 < t < \pi/\omega_n \tag{12.40}$$

$$F(t) = 0 \qquad \text{for } t < 0 \text{ and } t > \pi/\omega_n \tag{12.41}$$

Now the Fourier amplitude of the tapping machine pulse train as given in Eq. 12.22

$$F_n' = 2 f_r \int_0^{1/(2f_0)} u_0\, 2\pi\, f_0\ m \sin(2\pi f_0 t) \cos(2\pi n f_r t)\, dt \tag{12.42}$$

where n = 1, 2, 3, . . . This yields the force coefficients of the Fourier series

$$F_n' = F_n \frac{\pi}{4}\left(\frac{\sin\alpha}{\alpha} + \frac{\sin\beta}{\beta}\right) \tag{12.43}$$

in terms of the coefficients given in Eq. 12.43 and

$$\alpha = \frac{\pi}{2}\left(1 - n\frac{f_r}{f_0}\right) \quad \text{and} \quad \beta = \frac{\pi}{2}\left(1 + n\frac{f_r}{f_0}\right) \tag{12.44}$$

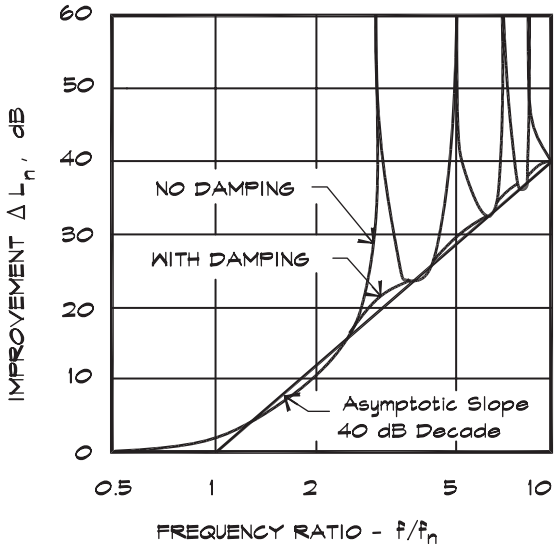
The improvement due to the elastic surface in the impact noise isolation is given in terms of a level

$$\Delta L_n = 20 \log \frac{F_n}{F_n'} = 20 \log\left(\frac{4/\pi}{\sin\alpha/\alpha + \sin\beta/\beta}\right) \tag{12.45}$$

which is shown graphically in Fig. 12.17.

Note that in this model we have ignored the contribution of the floor impedance and have assumed that it is very stiff compared with the elastic covering. At very low frequencies—that is, below the spring mass resonance—the improvement due the covering is zero. Above this frequency the surface covering becomes quite effective, giving a 12 dB per

FIGURE 12.17 Improvement in Impact Noise Isolation by an Elastic Surface (Ver, 1971)



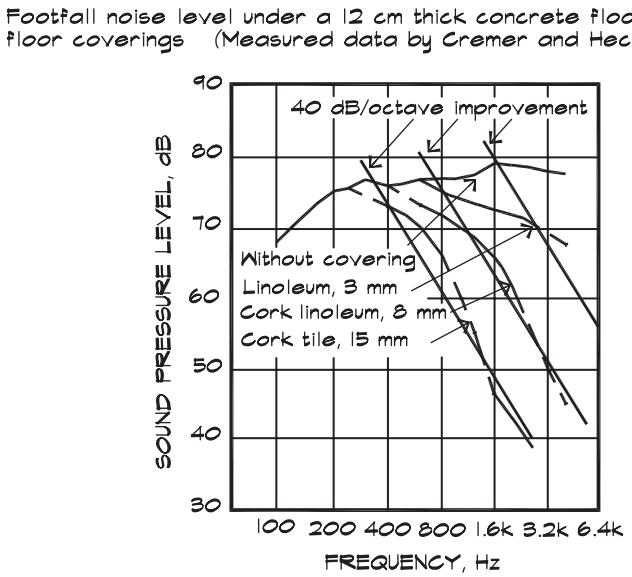
octave attenuation. The resonant peaks, which occur at odd multiples of the fundamental resonance, are diminished in actual field conditions by the damping in the surface treatment.

The normalized impact sound level for composite floor systems is given by

$$L_{n\ comp} = L_{n\ bare} - \Delta L_n \tag{12.46}$$

which is shown in Fig. 12.18 and compared with measured data. Note that while carpet and other elastic surface treatments improve the high-frequency loss for tapping noise, they

FIGURE 12.18 Calculated and Measured Impact Noise Levels



do not change the low-frequency transmission due to the weight of the walker, which is controlled by the stiffness and damping of the structural system. Likewise, they have little effect on the sound transmission loss of the structure since they do not alter the mass or stiffness of the floor system.

Improvement Due to Locally Reacting Floating Floors

A number of systems have been developed to provide noise and vibration isolation through the use of resilient floor or ceiling supports. These include continuous and point-supported floating floors, which are built on top of the structural floor system, as well as resiliently hung or separately supported ceilings. Each of these systems can provide improved isolation for footfall noise and some for walking noise, which as we will see, will depend on the softness of the mounts.

A locally reacting floating floor is one in which the influence of the initial force impulse is confined to the region around the point of impact. There is no vibrational wave in the upper slab, which is considered highly damped. An example might be a single layer of plywood without structural support other than the resilient mounts. Since the upper surface is hard the Fourier amplitude coefficients of the force pulse given in Eq. 12.24 can still be used. Ver (1971), citing Cremer (1952), has published the expected improvement

$$\Delta L_n = 20 \log \left[1 + \left(\frac{f}{f_1} \right)^2 \right] \cong 20 \log \left(\frac{f}{f_1} \right)^2 \tag{12.47}$$

where f_1 is the resonant frequency of the floating floor system is

$$f_1 = \frac{1}{2\pi} \sqrt{\frac{k'}{\rho_{S1}}} \tag{12.48}$$

and k' is the dynamic stiffness per unit area of the resilient layer between the floors, including the stiffness of the trapped air, and ρ_{S1} is the mass per unit area of the upper floor. As with a resilient surface, the result is a 12 dB per octave decrease in level above the resonant frequency. Below resonance there is no improvement.

Improvement Due to Resonantly Reacting Floating Floors

A resonantly reacting floor is a rigid, lightly damped floating slab in which bending waves are generated in response to the impulsive load. The input power into the upper slab is given by Eq. 12.9. The power balance equation for the upper slab is

$$W_{in} = W_{dis(1)} + W_{12} - W_{21} \tag{12.49}$$

and for the lower slab is

$$W_{12} = W_{dis(2)} + W_{21} \tag{12.50}$$

where powers with the subscript "dis" are the dissipated powers in slabs 1 and 2. The numerical subscripts indicate the direction of travel of the particular transmitted power

through the mounts. Three equations permit the calculation of the transmitted power through the mount

$$\begin{aligned} F &= (u_1 - u_2)\, z_m \\ u_1 &= u_0 - \frac{F}{z_1} \\ u_2 &= \frac{F}{z_2} \end{aligned} \tag{12.51}$$

where u_n = peak velocity amplitude in slab n (m / s)
z_n = point impedance of slab n (Ns / m)
z_m = point impedance of the support system (Ns / m)

When these equations are solved (Beranek and Ver, 1992) for the transmitted forces in terms of the point impedances, the improvement in the high-frequency limit, given in terms of the ratio of the initial to the transmitted forces is

$$\Delta L_n(\omega) \cong 10 \log \left[2.3\, c_{L1}\, h_1\, \eta_1\, n' \left(\frac{\omega^3}{\omega_1^4} \right) \right] \tag{12.52}$$

where c_{L1} = speed of longitudinal waves in the floating slab (m / s)
h_1 = thickness of the floating slab (m)
η_1 = damping factor in the floating slab
n' = number of resilient mounts per unit area (m^{-2})
$\omega_1 = \sqrt{\dfrac{k_m\, n'}{\rho_{s1}}}$ = resonant frequency of the floating slab (s^{-1})
k_m = dynamic stiffness of an individual mount (N / m)
ρ_{m1} = surface mass density per unit area of the floating slab (kg / m^2)

In obtaining this relationship we have assumed that the power is transmitted only through the mounts, which can be represented by a spring constant, and that the point impedance of the slab is that of an infinite thin plate. Typical material constants are given in Table 12.1 and calculated data are shown in Fig. 12.19, along with measured results published by Josse and Drouin (1969). The improvement follows a 9 dB/octave slope above the resonant frequency.

12.4 STRUCTURAL DEFLECTION

Sound can be produced by the deflection of the structure in a gross way under the weight of a walker or other moving object. When a load is applied to a floor, the structure will deflect and transmit the movement to the space below. The weight may be statically or dynamically applied. We have discussed the vibrations induced in a floor system due to a walker in Chapt. 11. Where the floor system is a slab or a nonisolated structure the motion imparted to the floor will be faithfully reproduced on the ceiling side. Where the ceiling is decoupled from the floor support system considerable improvement can be expected.

Floor Deflection

If a person stands in the center of an upper-story floor, the structure will deflect under the concentrated load of his weight. This is a static, as contrasted to a dynamic, effect since it

TABLE 12.1 Speed of Longitudinal Waves, Density, and Internal Damping Factors for Common Building Materials (Beranek and Ver, 1992)

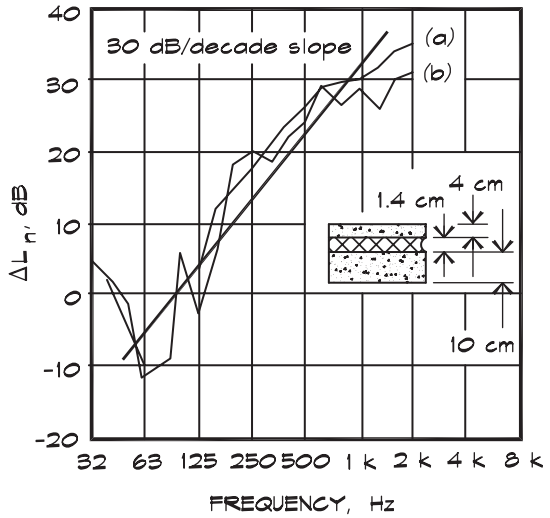
Material	c m/sec	ρ lb/ft^3	ρ kg/m^3	Damping Factor η*
Aluminum	5,150	170	2,700	10^{-4}- 10^{-2}
Brick	.....	120-140	1,900-2,300	0.01
Concrete, poured	3,400	150	2,300	0.005-0.02
Masonry block				
Hollow cinder (nominal 6 in. thick)	.....	50	750	0.005-0.02
Hollow cinder 5/8 in. sand plaster each side (nominal 6 in. thick)	.....	60	900	0.005-0.02
Hollow dense concrete (nominal 6 in. thick)	.....	70	1,100	0.007-0.02
Hollow dense concrete, sand-filled (6 in. thick)	.....	108	1,700	Varies with frequency
Solid dense concrete block (4 in. thick)	.....	110	1,700	0.012
Fir timber	3,800	40	550	0.04
Glass	5,200	156	2,500	0.001-0.01†
Lead:				
Chemical or tellurium	1,200	700	11,000	0.015
Antimonial (hard)	1,200	700	11,000	0.002
Plaster solid, on metal or gypsum lath	.....	108	1,700	0.005 - 0.01
Plexiglas or Lucite	1,800	70	1,150	0.002
Steel	5,050	480	7,700	10^{-4}- 10^{-2}
Gypboard (0.5 to 2 in)	6,800	43	650	0.01 - 0.03
Plywood (0.25 to 1.25 in)	.....	40	600	0.01 - 0.04
Wood chip board, 5 lb/ft^2	.....	48	750	0.005 - 0.01

*The range in values of η are based on limited data at 1000 Hz. The lower values are typical for the material alone.

† The loss factor for structures of these materials is very sensitive to construction techniques and edge conditions.

FIGURE 12.19 Improvement in Impact Noise Isolation for a Resonantly Reacting Floating Floor (after Beranek and Ver, 1992)

a) Standard tapping machine b) High heeled shoes
Note the negative ΔL_n in the vicinity of the resonant frequency.



takes place without any repetitive motion—at zero frequency. The deflection of the floor can be modeled in a number of ways, but let us take a simple example using standard structural equations (Roark and Young, 1975). If we assume that the load is supported entirely by a simply supported beam, the midpoint deflection, ignoring the weight of the beam, is

$$\delta = -\frac{\mathrm{W}\,L^3}{48\,\mathrm{E\,I}} \tag{12.53}$$

where δ = deflection at midspan (in)
W = weight of the impressed load (lbf)
L = length of the beam (in)
E = modulus of elasticity ($\mathrm{lb/in^2}$)
$= 1.92 \times 10^6\ \mathrm{lb/in^2}$ for douglas fir
I = moment of inertia ($\mathrm{in^4}$) = $\mathrm{b\,d^3/12}$
b = beam width and d = beam depth (in)

If the beam is a wood 2 × 12 that is 20 feet long and a 200 lb man is standing at the midpoint, the deflection is about 0.16 inch and the natural frequency, using a simple spring mass relationship is about 8 Hz.

If the floor is 5/8" plywood with a 1.5" lightweight concrete topping, each beam (assuming 16" centers) bears a distributed load of about 20 lbs/ft (w = 1.7 lbs/in). The midspan deflection is

$$\delta = -\frac{5\,\mathrm{w}\,L^4}{384\,\mathrm{E\,I}} \tag{12.54}$$

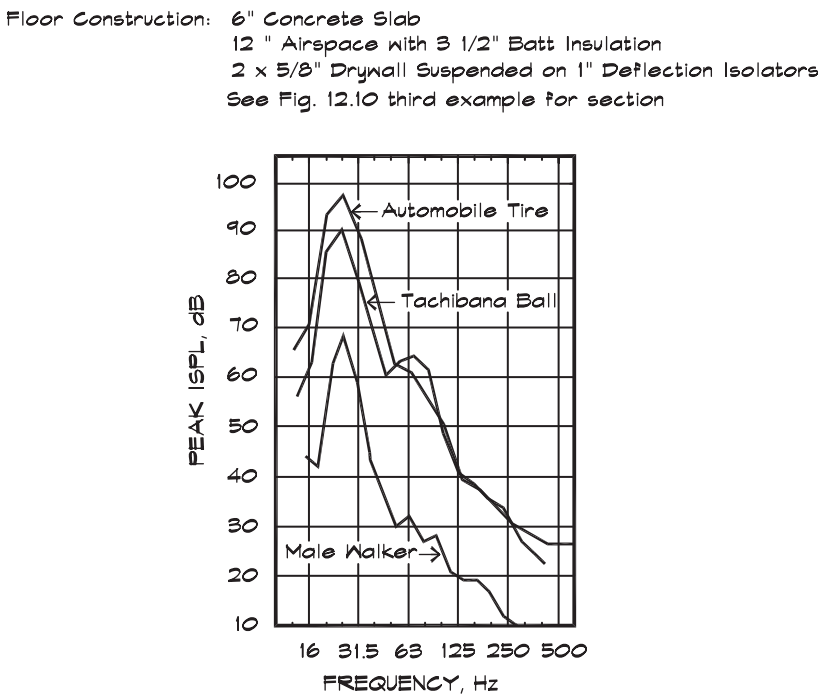
For the same beam the deflection is about 0.2" with no concentrated load, and adding the deflection due to the weight of a person it is about 0.36". The natural frequency of this system, modelled as a spring mass, is 5 Hz.

Now this is a very simple model. The plywood flooring can increase the moment of inertia of the beam somewhat. The point load may be distributed over more than one beam. The beams may not be simply supported. So we may get a bit higher resonant frequency. The result, however, will be a relatively low frequency, less than 10 to 15 Hz at these spans, and somewhat higher frequencies at shorter spans. When the floor resonance is excited by a walker we can get a high ceiling deflection unless the floor is stiffened, damped, and decoupled from the ceiling support structure.

Low-Frequency Tests

There are a few tests available to measure the results of noise generated by floor deflection. A few years ago ASTM committee E33 proposed a single microphone located 1 m below the midpoint of the ceiling. The receiving room is deadened by placing absorbing material in it. Three types of sources are used: a male walker, a heavy ball, and an automobile tire. The Japanese measurement standard JIS 1418 specifies an automobile tire mounted on an arm attached to a motor. The motor arm lifts the tire and drops it on the floor. A cam system catches the tire on the rebound before it can strike the floor again and lifts it again to the proper height. The standard specifies many drop positions and several microphone positions; however, since the fundamental resonance is excited, only a few drop positions and a single mic position are adequate for comparative measurements. Test results are shown in Fig. 12.20

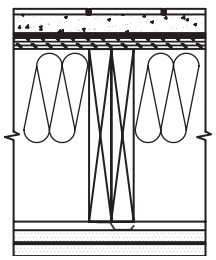
FIGURE 12.20 Peak Impact Sound Level Measurements Using Various Excitation Sources (Kinetics, 1990)



for a 6" thick concrete slab floor with a resiliently suspended drywall ceiling below. Note the peak impulse response is around 25 Hz, which is characteristic of the short span concrete floor used in these tests. The Tachibana ball, used in a Japanese test, is 180 mm in diameter and weighs 2.5 kg and is dropped from a height of 900 mm, and produces a very similar curve. A male walker yields similar results although the absolute levels vary.

Blazier and DuPree (1994) published measurements of the impact sound pressure level, taken on the wood floor system shown in Fig. 12.21, using a standard tapping machine as a source. Part of their study sought to quantify the importance of structural flanking in floating floors; however, these authors also extended their measurements to very low frequencies. In Fig 12.22 we see a comparison of a tapping test done on carpeted floor, a floating tile floor, and a partially floating tile floor. Below the 63 Hz band, there is little difference between surface treatments and we see two resonant peaks associated with the structural modes of the floor. Note that the difference between the unflanked and the partially flanked floating floor does not exceed 5 to 6 dB until the frequency is around 500 Hz, indicating a very

FIGURE 12.21 Wood Framed Floating Floor



10 mm (3/8") Ceramic Tile (or Carpet and Pad)
32 mm (1 1/4") Mortar Bed
10 mm (3/8") Mesh Mat
19 mm (3/4") Plywood
Double 38 x 292 mm (2 x 12) Wood Joists
165 mm (6 1/2" in) Batt Insulation
25 Ga Resilient Channels @ 16" O.C.
2 x 13 mm (1/2") Gypsum Board

FIGURE 12.22 Impact Noise Spectrum of ISO Tapping Machine on a Floated Floor vs Percent of Floor Area Flanked and Type of Surface Covering (Blazier and DuPree, 1994)

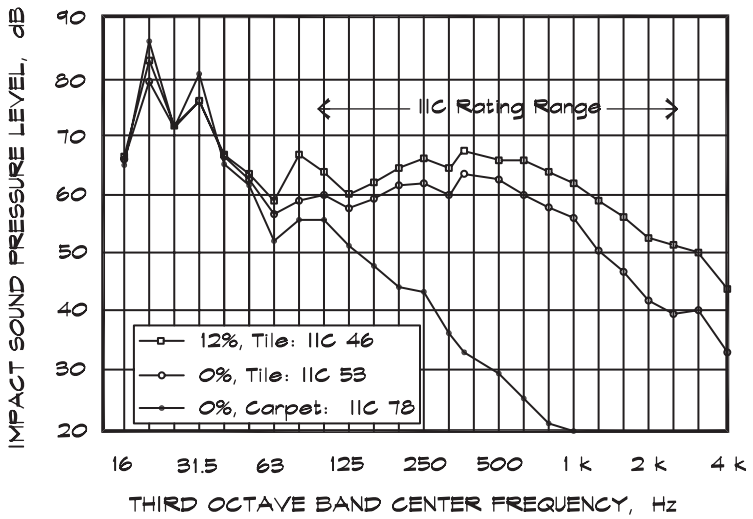
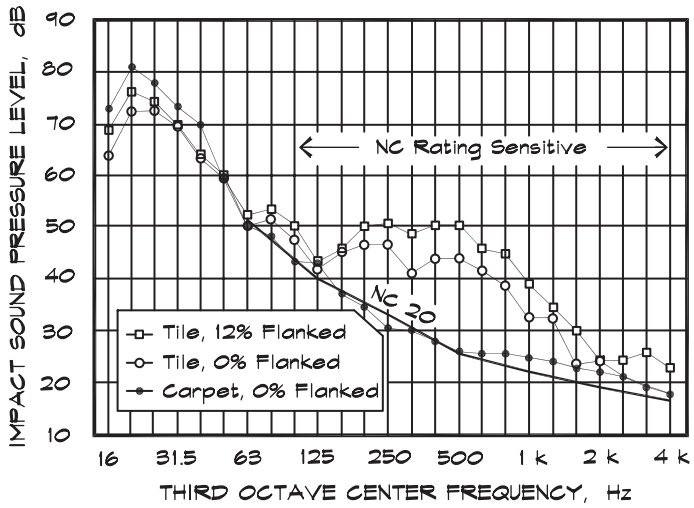


FIGURE 12.23 Impact Noise Spectrum of a Male Walker on a Floated Floor vs Percent of Surface Area Flanked and Type of Surface Covering (Blazier and DuPree, 1994)



low deflection support system. The structural flanking referred to in Fig. 12.22 was due to concrete in the mortar bed flowing around the pour dam and under the matting.

Figure 12.23 shows additional measurements from the same study on the noise produced by normal walking compared with a standard NC curve. Even with carpet the levels are audible at low frequencies and very audible for tile floors. At these very low frequencies, it is the structural resonance that is being excited by the walker. Since the effect is a gross property of the structure, it is unaffected by the surface covering. For this construction there is little decoupling between the floor and the ceiling, so relatively poor isolation results.

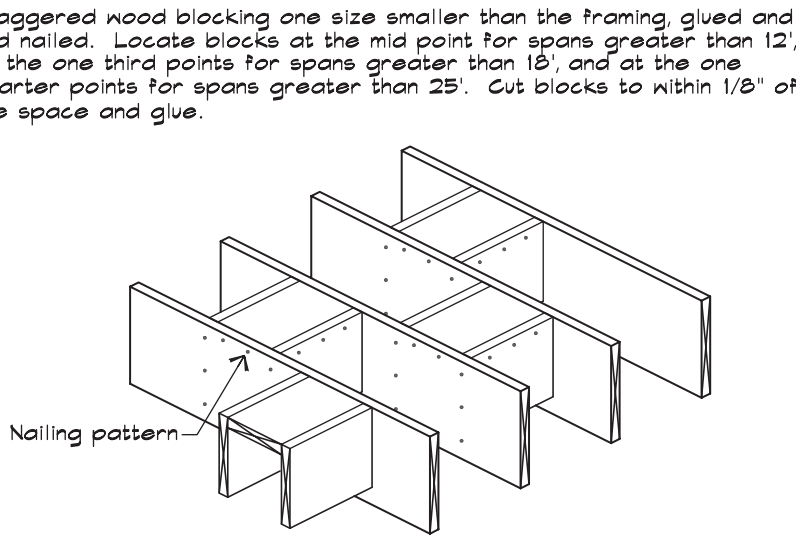
It is interesting to compare the data from the walking test in Fig. 12.20 for a concrete slab with similar tests done on the Fig. 12.23 construction, which is poorly isolated. The difference above 20 Hz is at least 10 dB and as much as 30 dB quieter in the concrete structure.

Structural Isolation of Floors

Three mechanisms are available to improve low-frequency sound transmission: 1) increase the stiffness of the floor support system, 2) increase the structural damping, and 3) increase the vibrational decoupling between the floor and the ceiling. In concrete structures both the stiffness and the damping increase with slab thickness. In Chapt. 11 we discussed the treatment of vibrations in concrete slab floors.

In wood floor structures both stiffness and damping can be increased by using stepped blocking, shown in Fig. 12.24. Blocking, using 2 × lumber one size smaller than the joist material, is installed in a series of inverted U shapes, glued and end-nailed into place. The next set of blocks is stepped (i.e., installed in a position that is offset relative to the first set) so that it can also be end-nailed. Both careful trimming and liberal application of glue are important to the installation. Blocks must be trimmed so that no more than a 1/8” gap is left between the block and the joist. The object of the blocking is to build an additional beam at right angles to the joists near the midpoint of the span and to provide additional damping.

FIGURE 12.24 Stepped Blocking in 2 × Wood Framing



Stepped blocking is used at the midspan in wood floors having a joist span of more than 12 feet (3.7 m) and less than 18 feet (5.5 m), and at the one-third points in spans 18 feet or greater. This type of blocking is most effective when it is installed before the floor is covered with a diaphragm and concrete, since the additional loads cause the structure to deflect and distributes some of the static load to the blocking. It can also be used as a retrofit; however, the joists have already deflected and the static load is not distributed as efficiently.

Figure 12.25 shows the results of tapping machine tests done on the floor system, drawn in Fig. 12.26, that incorporates stepped blocking, compared with the mesh mat construction. Note that the low-frequency transmission loss is much better than the Blazier and DuPree results.

12.5 FLOOR SQUEAK

Shiners

Floor squeak is generated most often in wood construction by the rubbing of a joist or panel on a nail that is not completely embedded. Called *shiners*, these nails occur when framers use nail guns to secure the plywood diaphragms to the joists and miss their target. If the nail is not centered it will pass through the plywood and lay alongside the joist as in Fig. 12.27. When the floor deflects due to the passage of a walker, the joist moves and the nail rubs against the wood, creating a high-pitched squeak much like a bird call.

If shiners are found in the field they should be removed by pounding them up from the bottom and pulling them out from the top. It is critical to listen below to the floor response, while someone walks over each portion of the floor, to locate nonbedded nails before any lightweight concrete or other flooring is applied. Since squeak is not dependent on mass or damping in the floor structure, it is not affected by the addition of lightweight concrete, and the presence of these materials make the nails much more difficult to remove.

FIGURE 12.25 Impact Noise Spectrum of ISO Tapping Machine on Two Types of Floors, Both Carpeted

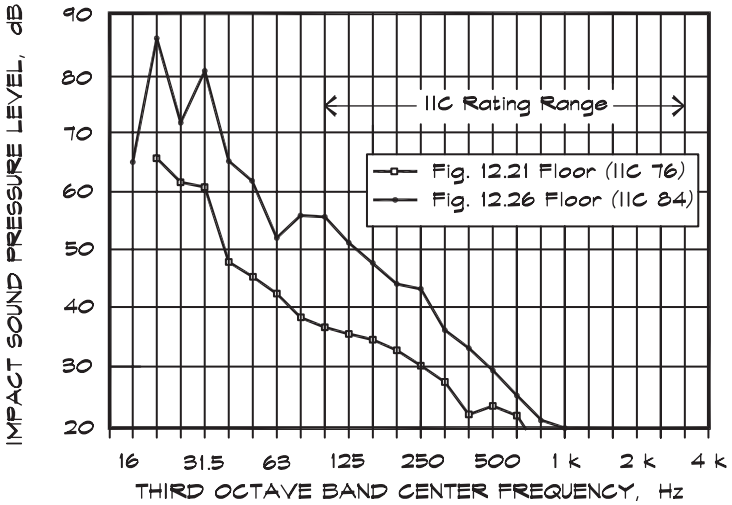
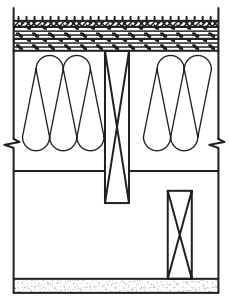


FIGURE 12.26 Separately Supported Wood Floor



44 oz Carpet on 40 oz Pad
2 x 19 mm (3/4") Plywood Glued and Screwed
38 x 204 mm (2 x 10) Wood Studs @ 16" O.C.
38 x 190 mm (2 x 8) Stepped Mid Span Blocking
150 mm (6" in) Batt Insulation
38 x 140 mm (2 x 6) Wood Joists
22 mm (7/8") Plaster

FIGURE 12.27 Sources of Floor Squeak

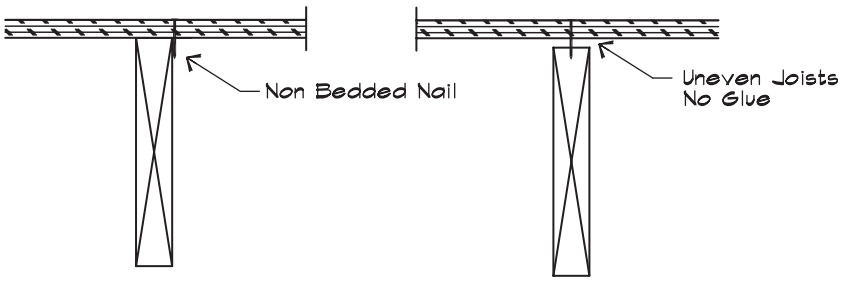
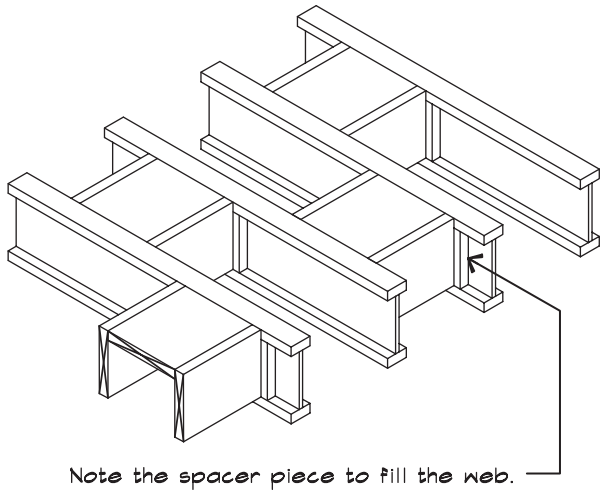


FIGURE 12.28 Truss Joists with Stepped Blocking



Uneven Joists

Nail squeak also can occur in a wood floor when the joists are of an uneven height. In these cases, the diaphragm does not make contact with the top of the joist and, in time, can move up and down on a nail, even one embedded in the joist. These conditions are particularly difficult to locate and remedy after the fact. Liberal application of panel adhesive to the top side of the joist before the plywood is installed will bond the subfloor to the joist and help fill in gaps that may be present. Factory manufactured truss joists can provide a better size consistency, which helps problems due to the variability in lumber. With truss joists, however, it is more difficult to construct stepped blocking since a spacer piece is required to fill the webbing, as illustrated in Fig. 12.28.

Hangers

Squeak can also occur when metal hangers are used to support the joists. In these cases the nails securing the hanger to the joist may rub on the hanger as the joist deflects. If the lumber varies in size, the diaphragm may not make contact with the joist and a gap will result. The best solution in these cases is to shim the joist so the top is even, and to glue the plywood down before nailing. It is preferable to frame the joists on the top plate of the bearing wall rather than being carried on wall-mounted metal hangers.

Nailing

A smooth nail, which does not grip the wood, is more prone to squeak than a ribbed nail. Ribbed or ring shank nails are helpful in preventing squeak since the wood is less likely to move vertically. Floor panel materials fabricated from strands of wood glued together are more prone to squeak than plywood since the high glue content material abrades and leaves a small hole where the nail can rub. Panel screws along with glue can give additional protection against squeak since they grip the wood more firmly. Drywall screws can be used; however, they are thinner than panel screws and more prone to break off.

This page intentionally left blank

13 NOISE IN MECHANICAL SYSTEMS

13.1 MECHANICAL SYSTEMS

Occupied spaces need the continuous delivery of the requirements for the human habitat—air, water, power, a controlled thermal environment—and the return of the waste products back to the surroundings in the form of carbon dioxide, waste water, sewage, refuse, and heat. To carry out these functions specialized machines are included in every building. The delivery of air and environmental control is provided by a heating, ventilating, and air conditioning (HVAC) system, water is circulated by pumps, and waste is removed through piping.

Fresh air is delivered by electric fans, most often centrifugal, but occasionally a plug, vane axial, or propeller type. Since thermal requirements necessitate the movement of more air than the oxygen requirements, most of the air in a room is recirculated to add or remove heat, and a portion of it is replaced with fresh air from the outside. Since the building is slightly pressurized by this process, some air leaks out through openings and the rest is removed by the mechanical system.

Temperature is controlled by blowing room air over a heat exchanger, a series of tubes, like the radiator in a car, through which a heated or cooled liquid is circulated. Heat, which is produced by electric resistive elements or a gas-fired boiler, is much easier to generate than cooling. Cooling is created by forcing a pressurized liquid or gas through an orifice, where it expands and some of the liquid changes to a gas, thereby absorbing heat from the surroundings. In the heat exchanger the cooled refrigerant is evaporated by taking heat from the circulating air. Once this process is complete the gas is recycled back to the condenser, where it is pressurized and converted back to a liquid, thereby releasing heat to the atmosphere. Figure 13.1 illustrates the process in a packaged air handler, which contains a compressor, a fan to exhaust the heat given off by the compressor, a pump to circulate the cooling fluid, a heat exchanger coil, and another fan to circulate the air in the room. When the compressor is physically separated from the fan coil unit as in Fig. 13.2, it is called a split system and the refrigerant is circulated through pipes connecting the two components.

Noise and vibration are often the byproducts of these mechanical processes. Figure 13.3 shows an example of an HVAC unit and several of the most common structural and airborne noise transmission paths. Each of the paths must be treated to assure overall noise control.

FIGURE 13.1 Heat Flow in a Package Air Handler

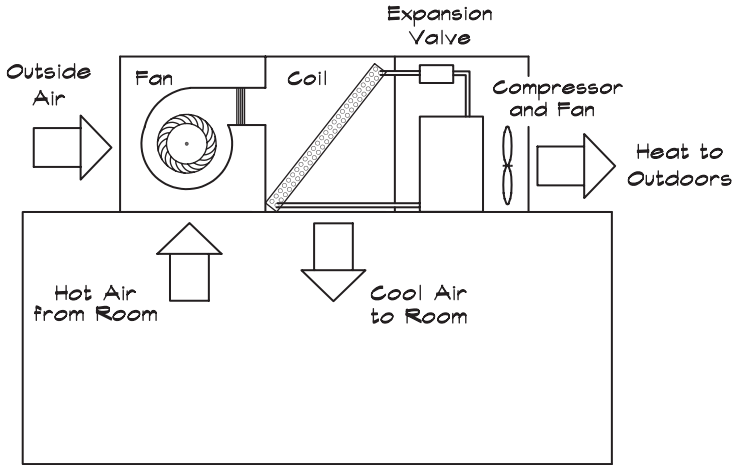


FIGURE 13.2 Air and Heat Flow in a Split System Air Conditioner

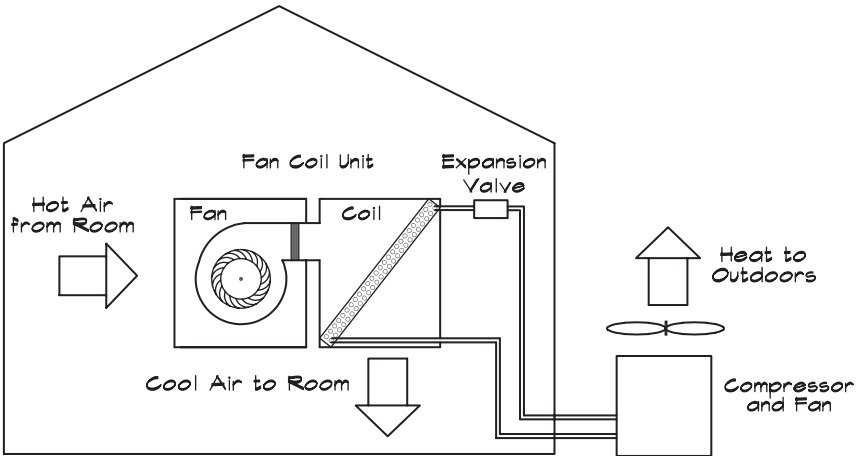
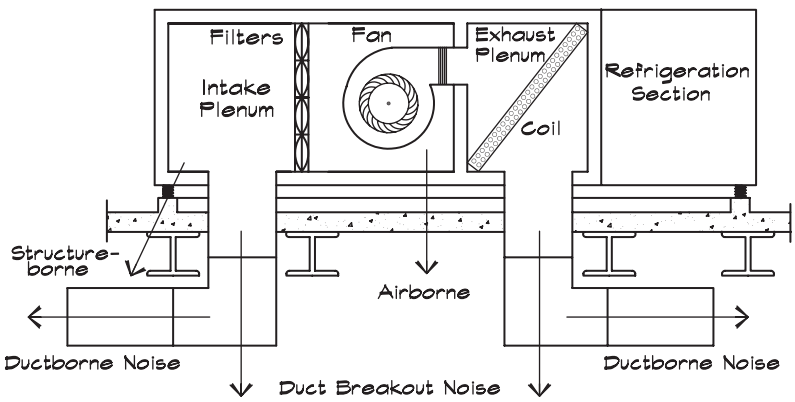


FIGURE 13.3 Rooftop Air Handling Unit Showing Noise Transmission Paths



If a generalization can be made, first it is important to insure that the structure-borne path is isolated. This may be accomplished by locating equipment away from sensitive receivers, and by vibration isolating it, along with all solid connections to it. Once the vibrational path is controlled, then the airborne path can be treated.

Manufacturer Supplied Data

An airborne sound transmission calculation begins with the sound power level generated by a piece of equipment. Often the manufacturer can provide measured sound power level data in octave bands or sound pressure levels at a known distance. The fact that data are available, however, should not lead to the suspension of disbelief about their accuracy. It is important to ferret out the origin of the measurements, and then to compare manufacturer-supplied data with data calculated from standard equations to see if the two are in general agreement. If they are not, further inquiries may be necessary to clarify the reason for the difference.

It seems logical that manufacturers would simply measure the noise their equipment makes and publish the data. Logic does not always prevail and companies may rely instead on calculations or other methods to determine noise level data. Although calculated data are better than nothing, the user does not necessarily know if the data are measured or calculated, and if calculated, which equations were used. In some instances, manufacturers will measure sound data on one unit and will publish data for other models, sizes, or speeds based on scaling relationships. This is standard practice among silencer manufacturers. The precise methodology and measurement techniques are important to learn, to confirm the appropriateness and applicability of the data. Occasionally manufacturers publish data with substantial errors. Here again a comparison to generic formulas can help uncover inconsistencies. Even when data have been measured directly on one unit, there can be some variation in levels due to the production process and details of the installation. The actual sound power level, based on carefully measured data, can still vary by a few dB in a given band from unit to unit, even under ideal conditions.

Airborne Calculations

If we have the sound power level of a source we can calculate the sound pressure level at the location of interest. When we are inside a room and sufficiently far from the source the reverberant field will predominate and we can proceed using Eq. 8.83 to predict the airborne sound pressure level in a space. If there is a significant contribution from a direct field component, such as when equipment is located outdoors, a separate calculation should be carried out using Eq. 2.74. If both types of field contribute, the levels from each should be combined.

When the sound wave cannot expand, such as when it is contained within a duct, there is no attenuation due to geometrical spreading. Instead the attenuation due to the duct lining or other elements in the ductwork is subtracted from the overall power, in each band, before the sound is introduced into a room. There it is analyzed as any other source would be.

13.2 NOISE GENERATED BY HVAC EQUIPMENT

Compilations of sound power level data radiated by mechanical equipment have been published and are sometimes available from equipment manufacturers. One of the best was developed in the 1960s by Laymon Miller (1968) for the U.S. Army. He continued this work through the 1980s and his studies are excellent references. The American Society of

TABLE 13.1 Sound Pressure Levels at 3 ft from Packaged Chillers, dB (Miller, 1980)

Freq	Rotary Screw Compressor	Reciprocating 10-50 Tons	Reciprocating 51-200 Tons	Centrifugal < 500 Tons	Centrifugal > 500 Tons
31 Hz	70	79	81	92	92
63 Hz	76	83	86	93	93
125 Hz	80	84	87	94	94
250 Hz	92	85	90	95	95
500 Hz	89	86	91	91	93
1 k Hz	85	84	90	91	98
2 k Hz	80	82	87	91	98
4 k Hz	75	78	83	87	93
8 k Hz	73	72	78	80	87
A-Weighted	90	89	94	97	103

Heating Refrigeration and Air Conditioning Engineers, ASHRAE, also publishes data on fans, pumps, and air handlers, and is another good source. Other manufacturers and trade associations make available data on specific pieces of equipment.

Refrigeration Equipment

Miller (1980) has collected and studied noise data on nearly 40 packaged chillers and reciprocating compressors. These units ranged in size from 15 tons to more than 500 tons of cooling capacity. A ton of refrigeration capacity is defined as the amount of heat removal required to produce one ton of ice from water at 32° F (0° C), 288,000 Btu (84.5 kW), in 24 hours or 12,000 Btuh (3.52 kWh). In air handling systems, fans generally are sized to provide about 400 cfm/ton of refrigeration. Sound data are given in terms of the sound pressure level at 3 ft. (1 m) from the equipment. No information on the physical size of the equipment is available. Several types of packaged chillers were investigated, differing primarily in the type of compressor. Table 13.1 shows the sound pressure levels at 3 ft. (1 m) for each type.

Cooling Towers and Evaporative Condensers

Cooling towers serve to cool water by using the latent heat absorbed during the process of evaporation. Water is introduced at the top of a cooling tower and falls to the bottom. Simultaneously air is blown or drawn upward through the falling water to aid in the mixing and increase evaporation. Noise is generated primarily by the fans; however, in certain cases the water itself can also contribute. Figure 13.4 shows examples of various types of cooling towers.

Overall sound power levels for each type are listed in Table 13.2 along with corrections to be subtracted from the overall level to obtain the level for each octave band. Cooling towers have a definite directivity, which depends on the type of fan, its location, and the side in question. Table 13.3 gives the approximate directional corrections to be added to the sound pressure levels calculated from the sound power levels in Table 13.2.

Note that these sound power level data have been calculated from sound pressure level measurements that are taken sufficiently far away from the unit that consideration of the size

FIGURE 13.4 Principal Types of Cooling Towers (Miller, 1980)

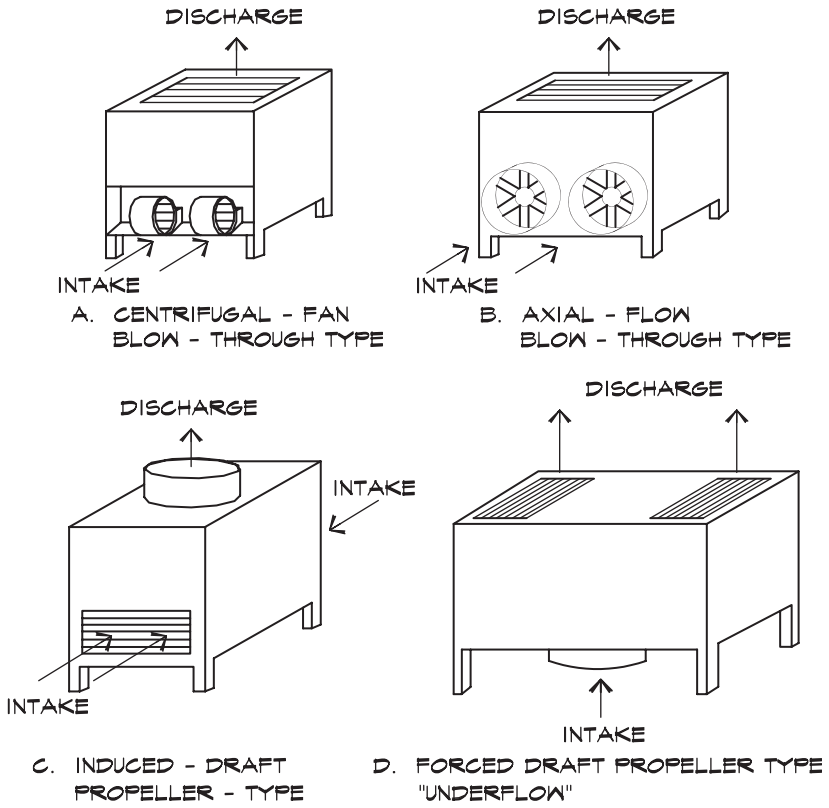


TABLE 13.2 Sound Power Levels of Cooling Towers, dB (Miller,1980)

	Propeller Type	Centrifugal Type
	$L_W = 95 + 10 \log (\text{fan hp}) - \text{Corr}$	$L_W = 85 + 10 \log (\text{fan hp}) - \text{Corr}$
31 Hz	8	6
63 Hz	5	6
125 Hz	5	8
250 Hz	8	10
500 Hz	11	11
1 k Hz	15	13
2 k Hz	18	12
4 k Hz	21	18
8 k Hz	29	25

TABLE 13.3 Corrections to Average Sound Pressure Levels for the Directivity of Cooling Towers, dB (Miller, 1980)

Octave Band (Hz)	31	63	125	250	500	1k	2k	4k	8k
Centrifugal Fan Blow-through Type									
Front (Fan Inlet)	3	3	2	3	4	3	2	2	2
Side (Enclosed)	0	0	0	−2	−3	−4	−5	−5	−5
Rear (Enclosed)	0	0	−1	−2	−3	−4	−5	−6	−6
Top (Discharge)	−3	−3	−2	0	1	2	3	4	5
Axial Flow Blow-through Type									
Front (Fan Inlet)	2	2	4	6	6	5	5	5	5
Side (Enclosed)	1	1	1	−2	−5	−5	−5	−5	−4
Rear (Enclosed)	−3	−3	−4	−7	−7	−7	−8	−11	−8
Top (Discharge)	−5	−5	−5	−5	−2	0	0	2	1
Induced Draft Propeller Type									
Front (Air Inlet)	0	0	0	1	2	2	2	3	3
Side (Enclosed)	−3	−3	−3	−3	−3	−3	−4	−5	−6
Top (Discharge)	3	3	3	3	3	4	4	3	3
Underflow Forced Draft Propeller Type									
Any Side	−1	−1	−1	−2	−2	−3	−3	−4	−4
Top	2	2	2	3	3	4	4	5	5

was unnecessary for the pressure to power conversion. If near field sound pressure levels are needed, then the physical size of the source must be taken into account by using Eq. 2.91.

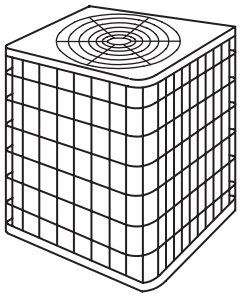
Air Cooled Condensers

In single or multifamily residences, air cooled condensers are used in place of the larger cooling towers or evaporative condensers. The noise from these units is due to the fan, usually a propeller type, with a small contribution from the air flow through the condenser coil decks. Figure 13.5 shows a sketch and measured data. These data are for a 3–5 ton residential unit based on sound pressure levels measured at 6 ft from the center of the fan and at 90° to the direction of airflow, which is out the top of the unit.

Pumps

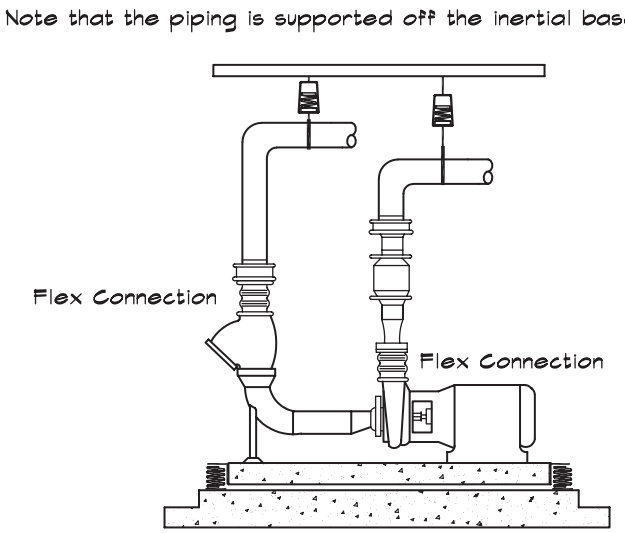
Pumps are found in virtually every building. When they are located above grade they are best mounted on an inertial base and housekeeping pad, such as that in Fig. 13.6. Pipe elbows that are connected to the pump should be supported from the isolation base. Flexible couplings are used to compensate for pipe misalignment and to provide structural decoupling. Piping should be resiliently supported in accordance with the recommendations given in Chapt. 11.

FIGURE 13.5 Sound Power Levels from a 3–5 Ton Air Cooled Compressor



Octave Band (Hz)	63	125	250	500	1k	2k	4k	8k
Level	87	84	75	75	74	71	64	56

FIGURE 13.6 Water Pump Installation with Inertial Base



Sound pressure level data at a distance of 3 ft have been published by Miller (1980) and are reproduced in Table 13.4. Also shown in the table are the corrections to be subtracted from the overall level to obtain the octave band values. A-weighted levels are 2 dB lower than the overall levels.

13.3 NOISE GENERATION IN FANS

Noise in HVAC systems is created both actively by mechanical equipment, primarily fans, and passively by static components in the air stream, which can create flow-generated noise. Nearly every component in an HVAC system, no matter how benign, can contribute to noise creation. It is critical to be aware of the noise generating mechanisms and their relative impact on the overall system.

TABLE 13.4 Overall Sound Pressure Levels at 3 ft for Pumps (Miller, 1980)

Speed Range (rpm)	Drive Motor Nameplate Power	
	Under 100 hp	Above 100 hp
	Overall Sound Pressure Level, dB	
3000–3600	71 + 10 log (hp)	85 + 3 log (hp)
1600–1800	74 + 10 log (hp)	88 + 3 log (hp)
1000–1500	69 + 10 log (hp)	83 + 3 log (hp)
450–900	67 + 10 log (hp)	81 + 3 log (hp)

Corrections to Overall SPL for Pumps, dB

Octave Band (Hz)	31	63	125	250	500	1k	2k	4k	8k
Level Subtracted	13	12	11	9	9	6	9	13	19

Fans

All buildings have fans of one sort or another for air circulation. Fans are typed according to the mechanism used to propel the air in Fig. 13.7, and further subdivided according to the type of blade in Fig. 13.8. The basic types are axial and centrifugal. Axial fans are the simplest to understand; they have a fixed-pitch multiple-bladed rotor. Propeller fans are unhoused, whereas vane axial and tube axial fans include a shroud or housing around the impeller.

FIGURE 13.7 Types of Fans

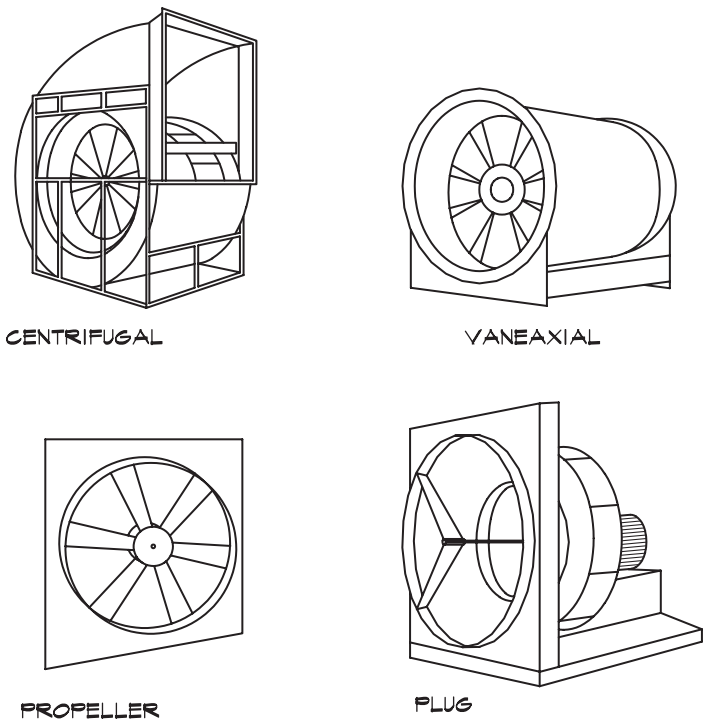
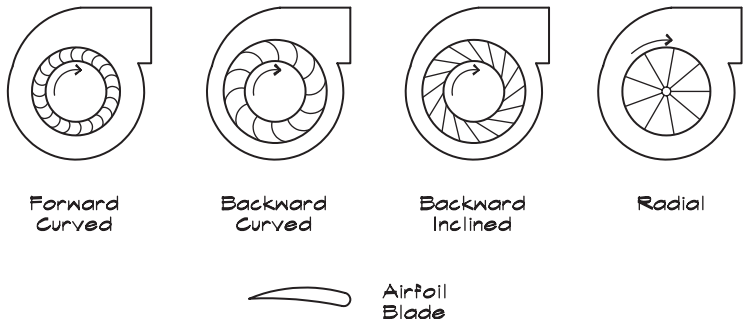


FIGURE 13.8 Types of Centrifugal Fans



Vane axial fans have fixed stator blades to straighten the flow after it passes through the rotor blades; tube axial fans do not.

Centrifugal fans consist of a series of blades, arranged at even intervals around a circle like a waterwheel, that throw the air from the inside to the outside of the circle as they rotate. Forward curved blades push the air out much like a jai alai racket. The air leaves the fan blade at a velocity higher than that of the blade tip. In backward-curved or backward-inclined blades the air velocity is lower than the tip velocity, so a lower noise level is generated. The forward-curved blades can generate the same air volume at a lower rotational speed, which means that the peak in their spectrum occurs at a lower frequency.

Fan noise is generated by several mechanisms, including the surge of the air pressure and velocity each time a blade passes, turbulent airflow in the air stream, and physical movement of the fan casing or enclosure. The noise emitted by each fan type follows a series of generalized laws called scaling laws, originally developed by Beranek, which have the general form (Graham, 1975 as given in ASHRAE, 1987)

$$L_w = K_F + 10 \log Q_F / Q_{REF} + 10 \log P_F / P_{REF} + C_{EFF} + C_{BFI} \tag{13.1}$$

where L_W = sound power level (dB re 10^{-12} Watts)
K_F = spectral constant which depends on the type of fan (dB) shown in Fig. 13.8
Q_F = volume of air per time passing through the fan (cfm or L/s)
Q_{REF} = reference volume (1 for cfm or 0.472 for L/s)

P_F = static pressure produced by the fan (in of water or Pa, gage)
P_{REF} = reference pressure (1 for in of water or 249 for Pa)

C_{EFF} = efficiency correction factor (dB)
C_{BFI} = blade frequency increment correction (dB)

Table 13.6 gives the off-peak the efficiency correction factor for fans running at less than peak efficiency. Equation 13.2 gives the method for calculating the fan's efficiency in FP units

$$\eta = \frac{100\, Q_F\, P_F}{6356\, W_{hp}} \tag{13.2}$$

TABLE 13.5 Level Correction K_F for Total Sound Power of Fans, dB (ASHRAE, 1987)

Fan Type	Octave Band Center Frequency (Hz)							
Centrifugal	63	125	250	500	1k	2k	4k	8k
Airfoil, Backwards Curved, Backward Inclined Wheel Diameter (inches)								
> 36 in	40	40	39	34	30	23	19	17
< 36 in	45	45	43	39	34	28	24	19
Forward Curved								
All	53	53	43	36	36	31	26	21
Radial Total Pressure (in. w.g.)								
Low 4–10	56	47	43	39	37	32	29	26
Med 6–15	58	54	45	42	38	33	29	26
High 15–60	61	58	53	48	46	44	41	38
Vaneaxial Hub Ratio								
0.3–0.4	49	43	43	48	47	45	38	34
0.4–0.6	49	43	46	43	41	36	30	28
0.6–0.8	53	52	51	51	49	47	43	40
Tubeaxial								
Wheel Diameter (inches)								
> 40 in	51	46	47	49	47	46	39	37
< 40 in	48	47	49	53	52	51	43	40
Propeller								
General ventilation and Cooling towers								
All	48	51	58	56	55	52	46	42

TABLE 13.6 Efficiency Corrections, C_{EFF} (ASHRAE, 1987)

Static Efficiency (% of Peak)	Correction Factor (dB)
90–100	0
85–89	3
75–85	6
65–74	9
55–64	12
50–54	15
below 50	16

TABLE 13.7 Blade Frequency Increment Correction, C_{BFI} (ASHRAE, 1987)

Fan Type	Blade Passing Octave, f_{bp}	C_{BFI}
Centrifugal		
Airfoil, backward curved, backward inclined	**250 Hz**	**3**
Forward curved	**500 Hz**	**2**
Radial blade pressure blower	**125 Hz**	**8**
Vaneaxial	**125 Hz**	**6**
Tubeaxial	**63 Hz**	**7**
Propeller		
Cooling Tower	**63 Hz**	**5**

where W_{hp} = power rating of the fan in horse power. The relative fan efficiency expressed as a percentage is

$$\eta_{rel} = 100\left[\frac{\eta}{\eta_{peak}}\right] \tag{13.3}$$

If the peak efficiency is not known, it is normal to assume a relative efficiency of about 80% , a value that adds about 6 dB to the data. If the peak efficiency is available from the manufacturer usually the actual sound power levels are as well.

The additional factor known as the blade frequency increment correction C_{BFI}, shown in Table 13.7, is a number to be added to the overall level in the octave band containing the blade passing frequency, f_{bp}

$$f_{bp} = \frac{\text{fan rpm} \times \text{number of blades}}{60} \tag{13.4}$$

The sound is radiated from both the fan intake and discharge. The formula assumes ideal inlet and outlet flow conditions and operation of the fan at a given efficiency.

Fans can also radiate noise through their enclosures and into the surrounding space. This is referred to as casing radiation and may be calculated by subtracting a factor for the insertion loss of the casing. Insertion losses are very dependent on the gauge and construction of the fan housing and those cited in Table 13.8 are only approximate.

Housing attenuation values are subtracted from the fan sound power level data to obtain a rough estimate of the power levels radiated by the fan through the housing when the fan is attached to ductwork. At low frequencies the power is unaffected by the casing since the enclosure vibration radiates as much noise as the unhoused fan would. The attenuations reflect the assumption that there is no separate enclosure around the fan housing and no absorption inside the housing, but that there is a silencer or lining in the ductwork close to the fan.

TABLE 13.8 Adjustments for the Attenuation of the Fan Housing, dB (Miller, 1980)

	Octave Band Center Frequency (Hz)								
	31	63	125	250	500	1k	2k	4k	8k
Attenuation	0	0	0	5	10	15	20	22	25

Fan Coil Units and Heat Pumps

In small offices and residential installations split HVAC systems are often used. These consist of an air-cooled condenser outdoors and a fan coil indoors as illustrated in Fig. 13.2. A refrigerant is circulated between the two devices moving under pressure in mostly liquid form between the condenser and the fan coil, and returning as a gas. The high-pressure liquid is forced through an expansion valve and thence into a cooling coil, where heat is removed from the room air.

The main source of noise in a fan coil is the small fan that circulates air through the coil and into the conditioned space. Figure 13.9 gives measured sound power levels at several rates of flow. When a fan coil unit is located above a T-bar ceiling the noise it generates is difficult to control even with lined ductwork or silencers. Wherever possible fan coils should be installed in closets or above drywall ceilings, which provide the necessary transmission loss. Lined ducts or silencers are usually required.

The compression expansion cycle can be used to heat as well as cool a space. The process is reversible in a device known as a heat pump, which literally can carry heat into or out of a building. Figure 13.10 illustrates this process. In the winter or heating mode, at the top of the figure, refrigerant is circulated through a refrigerant-to-water heat exchanger, where it absorbs heat from water that is colder than the exterior environment but warmer than the refrigerant. The heat absorbed warms the refrigerant and converts it to a gas. It then

FIGURE 13.9 Discharge Noise Levels of Fan Coil Units (Fry, 1988)

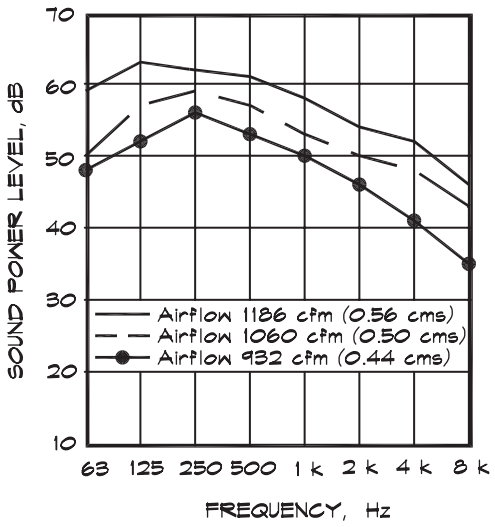
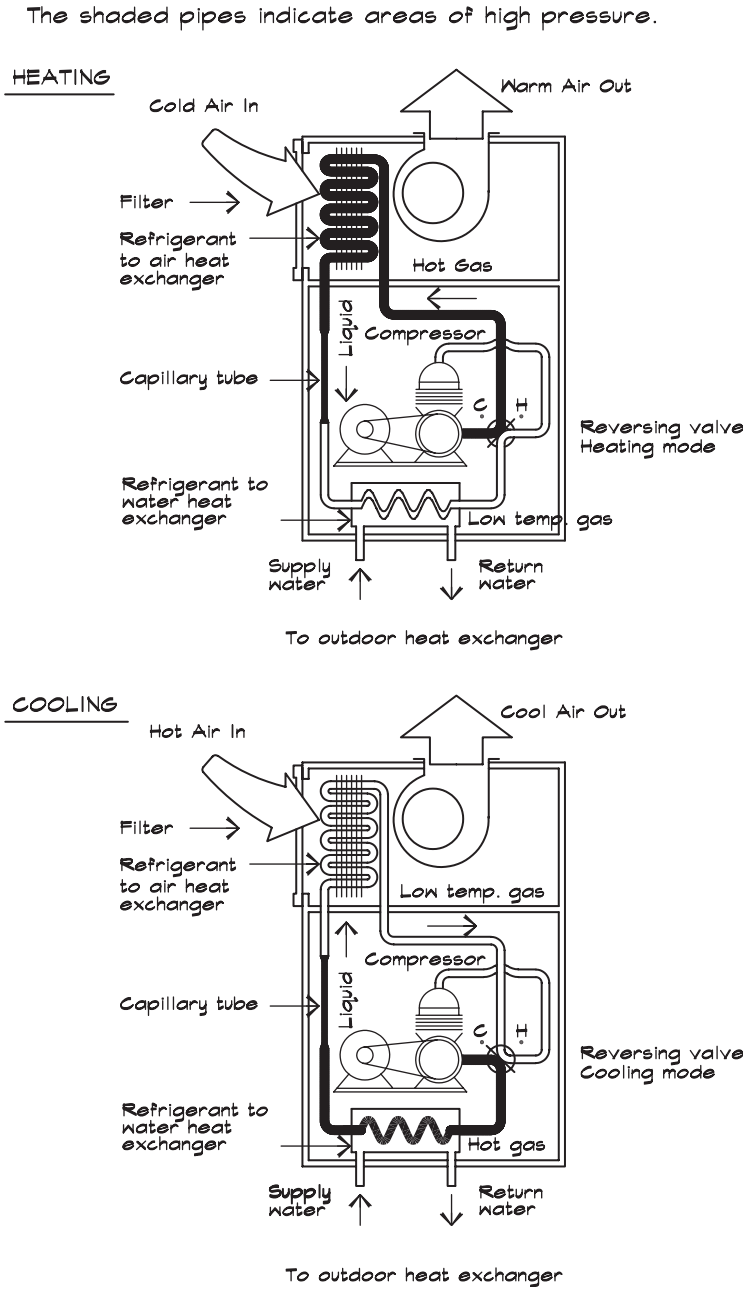


FIGURE 13.10 Air and Heat Flow in a Heat Pump System (California Heat Pump)



flows to the compressor, which further warms and pressurizes it by performing mechanical work on it. The hot gas then flows through a coil where a fan blows air over it and into the occupied space. The gas gives off heat in the exchange and condenses into a liquid. The liquid is then forced under pressure through the capillary tube where it expands and is returned to the heat exchanger. The cooling cycle is just the reverse of this process and is enabled by changing the direction of flow of the reversing valve. Noise generated by heat pump units can be greater than fan coils since the compressor is located in the same unit as the coil.

VAV Units and Mixing Boxes

In recent years, due to the emphasis being placed on energy conservation, the variable air volume or VAV system has become a commonly used design. A VAV system consists of a fan operating at a constant velocity that pressurizes a series of valves, which in turn feed a network of diffusers, distributed throughout the occupied space. The valves consist of remotely controlled dampers, which regulate the airflow to each space. A bypass duct is used to route the unused air back to the inlet side of the air handler in order to maintain a constant volume through the fan. A VAV unit, pictured in Fig. 13.11, must be capable of regulating the airflow from the full design capacity down to a very small flow, usually by means of butterfly dampers.

Blazier (1981) has published discharge (in Fig. 13.12) and radiated (in Fig. 13.13) sound power levels generated by VAV units for two rates of flow. The noise is generated by disturbed flow around the dampers.

Many manufacturers publish data on both discharge and casing radiated sound from VAV units. The most useful data are given in terms of sound power levels; however, some manufacturers list data in terms of NC levels, which are obtained by assuming a certain

FIGURE 13.11 A Variable Air Volume Unit

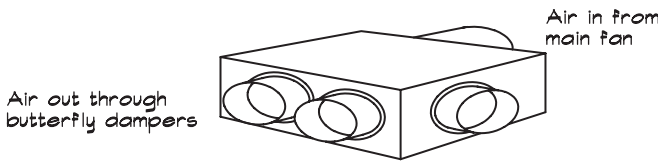


FIGURE 13.12 Range of Discharge VAV Noise Levels at Two Operating Points (Blazier, 1981)

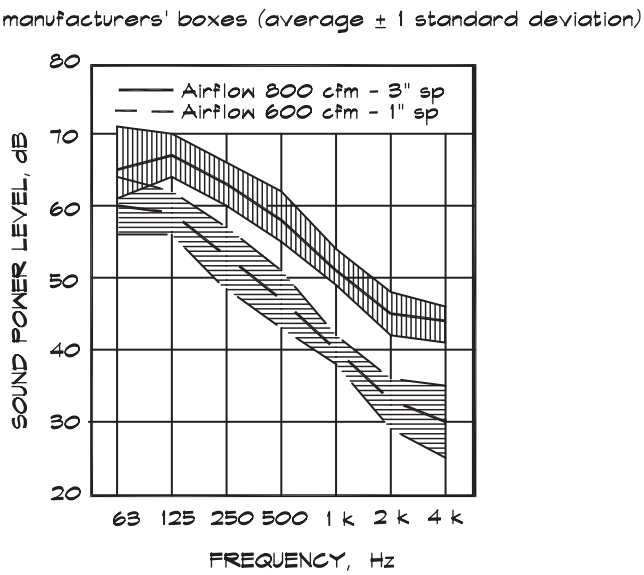


FIGURE 13.13 Range of Radiated VAV Noise Levels at Two Operating Points (Blazier, 1981)

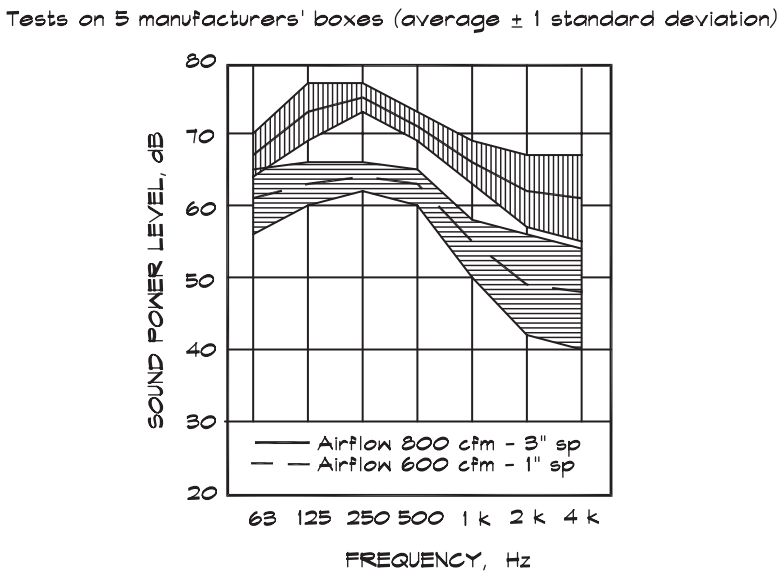
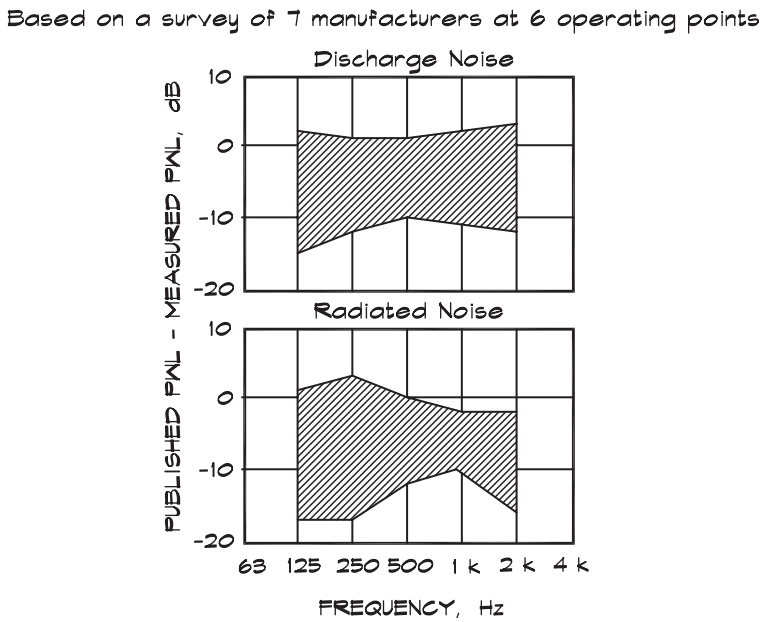
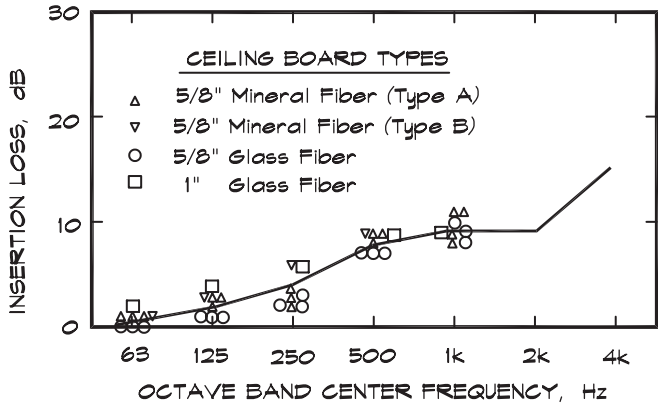


FIGURE 13.14 Comparison between Published and Measured Sound Power Levels (Blazier, 1981)



power-to-pressure conversion in the receiving room (usually −10 dB) and sometimes an additional noise reduction (also −10 dB) due to the ceiling tile. Blazier (1981) has measured the noise generated by VAV units as they compare to data published by the manufacturers. He lists in Fig. 13.14 the difference between published and measured sound power levels citing a 6 to 10 dB understatement of the noise furnished by manufacturers. Some of the

FIGURE 13.15 Measured Insertion Loss of Typical Lay-in Acoustical Ceiling Systems (Blazier, 1981)



discrepancy may be due to the difficulty in duplicating in the field the smooth entry and exit flow conditions under which the laboratory data are taken. Blazier (1981) has also measured the insertion loss due to acoustical tile ceilings given in Fig 13.15. Note that the loss approaches 10 dB only at high frequencies.

13.4 NOISE GENERATION IN DUCTS

Flow Noise in Straight Ducts

Once air has been set in motion it can generate noise by creating pressure fluctuations through turbulence, vortex shedding, mixing, and other mechanisms. Steady flow in a straight duct does not generate appreciable noise, when compared with other sources such as abrupt transitions in the air path, takeoffs, and elbows. Figure 13.16 (Fry, 1988) shows sound power

FIGURE 13.16 Sound Power Spectra of 600 mm (24 in) × 600 mm Straight Steel Duct for Various Air Velocities (Fry, 1988)

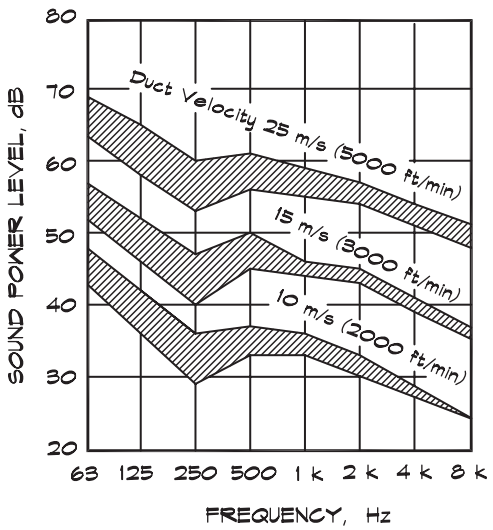
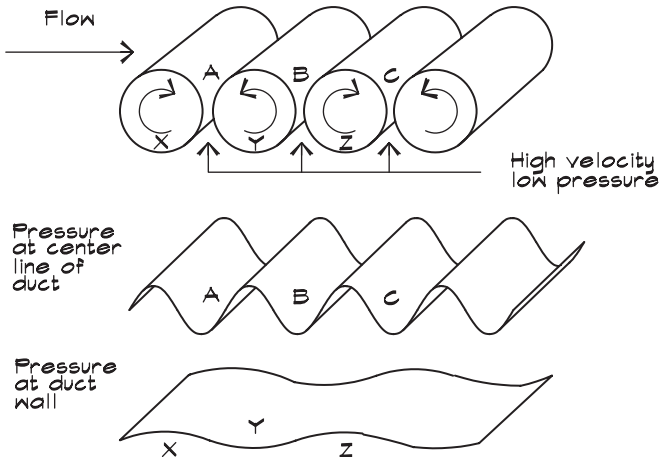


FIGURE 13.17 Buffeting in Rectangular Ducts (Fry, 1988)



level data on noise generated in straight duct runs for straightened flow. Levels generally follow an 18 dB per doubling of velocity scaling law.

The cited data show a significant rise in level when the cross sectional duct dimension is equal to a wavelength, which for this example is about 500 Hz. This bump coincides with the establishment of full cross duct turbulent eddies illustrated in Fig. 13.17. Eddies form downstream of disturbing elements such as rods or dampers. By themselves eddies are not particularly efficient sound radiators; however, they can generate noise when they impinge on a flat plate or other low-frequency radiator. In an open duct, eddies cause the flow to alternately speed up and slow down, producing pressure maxima at points X and Z and a pressure minimum at point Y.

In air distribution design it is prudent to control the duct velocity by increasing the cross sectional area of the duct, thus slowing the flow as it approaches the space served. A duct layout is pictured in Fig. 13.18.

FIGURE 13.18 Typical Duct Run in an HVAC System

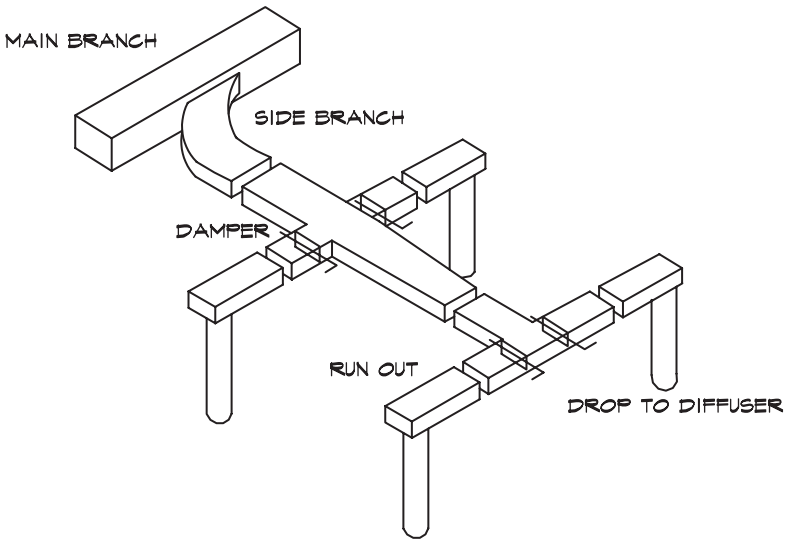


TABLE 13.9 Velocity Criteria for Air Distribution Systems

Maximum Air Velocities (ft/min)							
Description / NC Criteria	Slot Speed at Termination	Distance from Termination					
		< 10 ft (3 m) No Lining	> 5 ft (1.5 m)	> 10 ft (3 m)		> 20 ft (6 m)	
		< 5 ft (1.5 m) Lining*	Lining*	No Lining	Lining*	No Lining	Lining*
NC 15 supply	250	300	500	350	800	425	1000
NC 15 return	300	350	600	350	950	500	1200
NC 20 supply	300	350	600	425	950	550	1200
NC 20 return	350	425	725	500	1150	650	1450
NC 25 supply	350	425	725	500	1150	700	1450
NC 25 return	425	500	875	650	1375	800	1725
NC 30 supply	425	500	875	700	1375	850	1725
NC 30 return	500	600	1050	800	1650	950	2075
NC 35 supply	500	600	1050	800	1650	1000	2075
NC 35 return	600	700	1250	900	2000	1150	2500
NC 40 supply	600	700	1250	900	2000	1150	2500
NC 40 return	725	850	1500	1075	2400	1380	3000
NC 45 supply	725	850	1500	1075	2400	1375	3000
NC 45 return	875	1000	1800	1300	2875	1675	3575

* Duct must be lined with 1" fiberglass duct liner or flexible duct from this point to termination.

1 m/sec = 196.8 ft/min

FIGURE 13.19 Sound Power Levels of Abrupt and Gradual Area Transitions from 600 mm × 600 mm to 200 mm × 200 mm Duct Cross Sections (Fry, 1988)

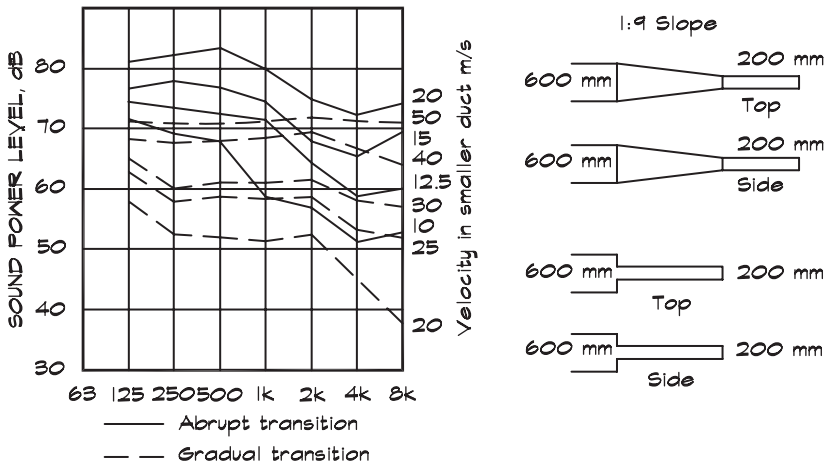


Table 13.9 gives velocity recommendations appropriate for various NC levels in the receiving space. These include consideration of the attenuation of downstream sections of lined duct. For the lined duct velocities, it is assumed that the remaining ductwork is covered on the inside with a 1" (25 mm) thick fiberglass duct liner or is flex duct.

Noise Generated by Transitions

Aerodynamic noise is generated at both gradual and abrupt changes in duct area. Gradual transitions and low velocities generate less turbulence than abrupt transitions and high velocities. Beyond these generalizations there are few models to use for sound power level prediction. Fry (1988) has published measured data for several expansion ratios in rectangular ducts, which are reproduced in Figs. 13.19 and 13.20.

Air Generated Noise in Junctions and Turns

Noise generated in transition elements such as turns, elbows, junctions, and takeoffs can run 10 to 20 dB higher than the sound power levels generated in straight duct runs. Ducts having radiused bends, with an aspect ratio of 1:3 generated, no more noise than a straight duct (Fry, 1988). Elbows having a 90° bend are about 10 dB noisier than straight duct. One or more turning vanes can reduce the noise 8 to 10 dB at high and low frequencies while increasing it 3 to 4 dB in the mid frequencies.

Ver (1984) published an empirical equation for the sound power levels given off by various fittings, which was reproduced in the 1987 ASHRAE guide. For branches, turns (including elbows without turning vanes), and junctions such as those pictured in Fig. 13.21, it is

$$L_{W\ OCT}(f_0) = K_J + 10 \log (f_0/63) + 50 \log (U_B) + 10 \log (S_B) + 10 \log (D_B) + C_B + \Delta r + \Delta T \tag{13.5}$$

FIGURE 13.20 Sound Power Levels of Abrupt and Gradual Area Transitions from 600 mm × 600 mm to 600 mm × 200 mm Duct Cross Sections (Fry, 1988)

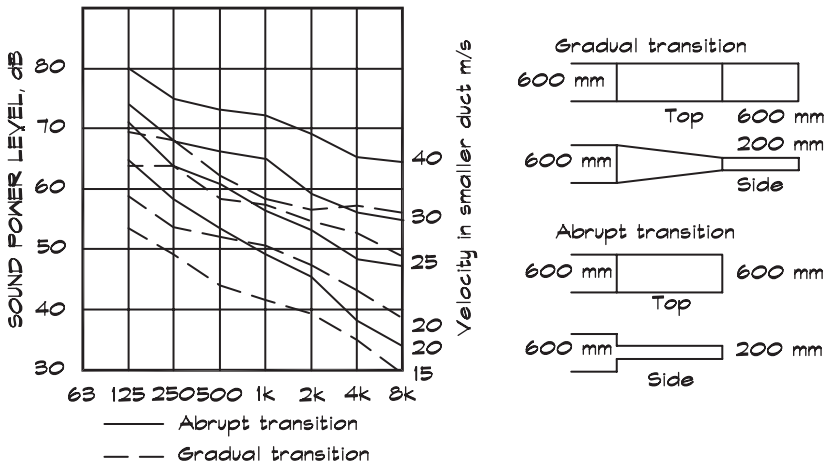
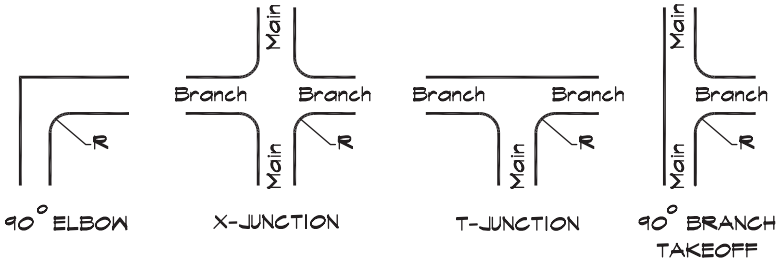


FIGURE 13.21 Elbows, Junctions, and Branch Takeoffs



where $L_{W\ OCT}$ = octave band sound power level (dB re 10^{-12} Watts)
f_0 = center frequency of the octave band (Hz)
K_J = characteristic spectrum of the junction or turn, based on the Strouhal number
U_B = velocity in the branch duct (ft/s)
S_B = cross sectional area of the branch duct (ft^2)
D_B = equivalent diameter of the branch duct or in the case of junctions, $D_B = \sqrt{4\ S_B/\pi}$ (ft)
C_B = a constant which depends on the type of branch or junction

The flow velocity in the branch is calculated using

$$U_B = Q_B/(60\ S_B)\ (ft/s) \tag{13.6}$$

where Q_B = flow volume in the branch, (cfm)

The term Δr is a correction for the roundness of the bend or elbow associated with the turn or junction (Reynolds, 1990)

$$\Delta r = \left\{1.0 - \frac{R_D}{0.15}\right\}\left[6.793 - 1.86 \log (S_t)\right] \tag{13.7}$$

where R_D is the rounding parameter $R_D = \dfrac{R}{12\ D_B}$ for the radius, R (inches), of the inside edge of the bend, and $S_t = f_0\ D_B/U_B$ is the Strouhal number.

The term ΔT is a correction for upstream turbulence, which is applied only when there are dampers, elbows, or branch takeoffs upstream, and within five main duct diameters, of the turn or junction under consideration.

$$\Delta T = -1.667 + 1.8\ m - 0.133\ m^2 \tag{13.8}$$

$$m = U_M/U_B \tag{13.9}$$

where U_M = velocity in the main duct (ft/min)
U_B = velocity in the branch duct (ft/min)

The characteristic spectrum K_J in Eq. 13.5 may be calculated (Reynolds, 1990) using

$$\begin{aligned} K_J = &-21.61 + 12.388\ m^{0.673} \\ &- 16.482\ m^{-0.303} \log (S_t) \\ &- 5.047\ m^{-0.254} \log (S_t)^2 \end{aligned} \tag{13.10}$$

Finally the correction term C_B depends on the type of junction in Fig. 13.21.

For X - junctions

$$C_B = 20 \log \left[\frac{D_M}{D_B}\right] + 3 \tag{13.11}$$

and for T junctions

$$C_B = 3 \tag{13.12}$$

For 90° elbows without turning vanes,

$$C_B = 0 \tag{13.13}$$

For a 90° branch takeoff,

$$C_B = 20 \log \left[\frac{D_M}{D_B}\right] \tag{13.14}$$

Air Generated Noise in Dampers

Damper noise follows the same general spectrum equation (Eq. 13.5) as was used for branches and turns, although the terms are defined somewhat differently

$$L_{W\ OCT}(f_0) = K_D + 10 \log (f_0/63) + 50 \log (U_C) + 10 \log (S) + 10 \log (D_H) + C_D \tag{13.15}$$

where $L_{W\ OCT}$ = octave band sound power level (dB re 10^{-12} Watts)
f_0 = center frequency of the octave band (Hz)
K_D = characteristic spectrum of the damper, based on the pressure loss factor and the Strouhal number
U_C = flow velocity in the constricted part of the duct (ft/s)
S = cross sectional area of the branch duct (ft^2)
D_H = duct height normal to the damper axis (ft)

Before solving Eq. 13.15, several preliminary calculations must be undertaken. The characteristic spectrum is determined from the Strouhal number, which depends on the velocity, the blockage factor, and the pressure loss coefficient, C.

The pressure loss coefficient in FP units is

$$C = 15.9 \cdot 10^6 \frac{\Delta P}{(Q/S)^2} \tag{13.16}$$

where ΔP = pressure drop across the fitting (in. w.g.)
Q = flow volume, (cfm)

The blockage factor, B, for multiblade dampers and elbows with turning vanes is

$$B = \frac{\sqrt{C} - 1}{C - 1} \quad \text{for } C \neq 1 \tag{13.17}$$

or

$$B = 0.5 \quad \text{if } C = 1 \tag{13.18}$$

For single blade dampers, it is

$$B = \frac{\sqrt{C} - 1}{C - 1} \quad \text{if } C < 4 \tag{13.19}$$

or

$$B = 0.68\, C^{-0.15} - 0.22 \quad \text{if } C > 4 \tag{13.20}$$

Next the constricted velocity is calculated using

$$U_c = Q/(60\ SB) \quad \text{(ft/sec)} \tag{13.21}$$

which gives the Strouhal number

$$S_t = f_0 \, D / U_c \tag{13.22}$$

Having calculated these numbers for the particular fitting we can find the characteristic spectrum for dampers (Reynolds, 1990)

$$\begin{aligned} K_D &= -36.6 - 10.7 \log (S_t) \quad \text{for } S_t \leq 25 \\ K_D &= -1.1 - 35.9 \log (S_t) \quad \text{for } S_t > 25 \end{aligned} \tag{13.23}$$

Air Noise Generated by Elbows with Turning Vanes

For elbows with turning vanes we use Ver's equation with a slightly different definition of the terms

$$\begin{aligned} L_{W\ OCT}(f_0) = K_T &+ 10 \log (f_0/63) + 50 \log (U_C) \\ &+ 10 \log (S) + 10 \log (D_C) + 10 \log n \end{aligned} \tag{13.24}$$

where $L_{W\ OCT}$ = octave band sound power level (dB re 10^{-12} Watts)
f_0 = center frequency of the octave band (Hz)
K_T = characteristic spectrum of an elbow with turning vanes
U_C = flow velocity in the constricted part of the flow field (ft/s)
S = cross sectional area of the elbow (ft^2)
D_C = chord length of a typical vane (in)
n = number of turning vanes

Figure 13.22 shows the definition of the chord length.

The characteristic spectrum (Reynolds, 1990) is

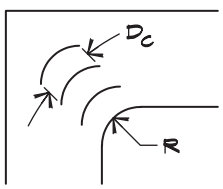
$$K_T = -47.5 - 7.69 \, [\log (S_t)]^{2.5} \tag{13.25}$$

where the Strouhal number in Eq. 13.22 is calculated from the pressure loss coefficient in Eq. 13.16, the blockage factor,

$$B = \frac{\sqrt{C} - 1}{C - 1} \tag{13.26}$$

and the constricted velocity in Eq. 13.21.

FIGURE 13.22 Ninety Degree Elbow with Turning Vanes



Grilles, Diffusers, and Integral Dampers

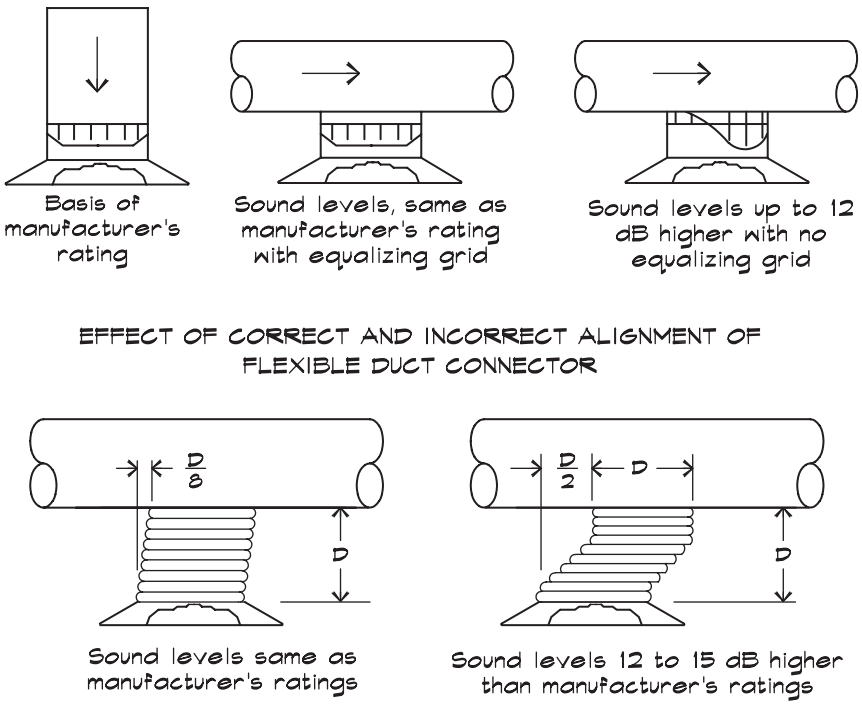
Diffuser generated noise is of paramount importance in HVAC noise control since it cannot be attenuated by the addition of downstream devices. Since diffuser noise is primarily dependent on the air velocity through the device, the only method for attenuating it is to reduce it, either by adding additional diffusers, or by increasing the size of the existing diffusers. Often diffuser noise is influenced by the upstream flow conditions that can be modified. Pressure equalizing grilles at the entry to the diffuser can help reduce the contribution due to turbulence.

Sound data on diffuser noise is published by manufacturers in terms of NC levels; however, these are only valid for ideal flow conditions. To achieve ideal conditions, flexible ducts must be straight for at least one duct diameter before the connection to the diffuser and must not be pinched or constricted. Figure 13.23 shows examples of correct and incorrect flexible duct connections.

It should be noted that manufacturer published data are given for one diffuser or, in the case of linear diffusers, for one four-foot long segment, with a power-to-pressure conversion of 10 dB, which corresponds to a room absorption of about 400 sabins and a distance of about 12 ft. The actual power-to-pressure conversion factor should be calculated for the specific room in question. Where there are multiple diffusers in a space, a factor of 10 log n, where n is the number of diffusers or the total number of four-foot segments of linear diffuser, must be added to the published noise levels.

When sound levels from diffusers are not available, they can be approximated using a general equation (Reynolds, 1990). It relates the diffuser noise to the sixth power of the flow

FIGURE 13.23 Correct and Incorrect Diffuser Installation (Fry, 1988)



velocity and the third power of the pressure drop.

$$L_w = 10 \log S_G + 30 \log \xi + 60 \log U_G - 31.3 \tag{13.27}$$

where L_W = overall sound power level (dB re 10^{-12} Watts)
S_G = cross sectional face area of the grille or diffuser (ft^2)
U_G = flow velocity prior to the diffuser (ft/s)
ξ = normalized pressure drop coefficient

It is clear that the noise emitted by diffusers is very dependent on the flow velocity and the formula yields an 18 dB per doubling of velocity relationship. For a given flow volume a doubling of grill area will reduce noise by 15 dB.

The normalized pressure drop is

$$\xi = 334.9 \ \frac{\Delta P}{\rho_0 \ U_G^2} \tag{13.28}$$

where ΔP = pressure drop across the diffuser (in w.g.)
ρ_0 = density of air (0.075 lb/ft^3)
U_G = flow velocity prior to the diffuser (ft/min)
$= \dfrac{Q}{60 \ S_G}$ (for Q in cfm)

The octave band sound power levels can be calculated from the overall level by adding a correction factor to Eq. 13.27

$$L_{W \ OCT} = L_W + C_D \tag{13.29}$$

The correction term for round diffusers is

$$C_D = -5.82 - 0.15A - 1.13 \ A^2 \tag{13.30}$$

and for rectangular (including slot) diffusers,

$$C_D = -11.82 - 0.15A - 1.13 \ A^2 \tag{13.31}$$

and is normalized to a peak frequency

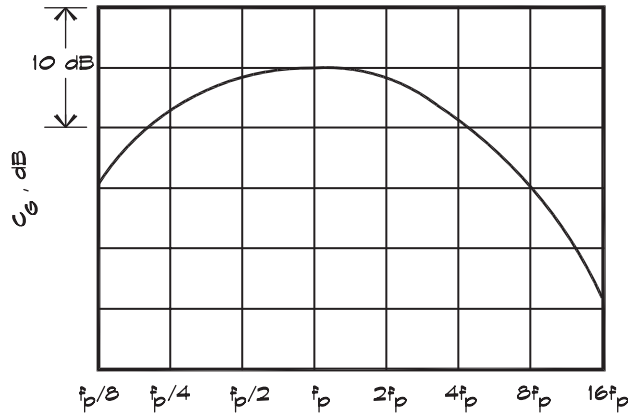
$$f_P = 48.8 \ U_G \tag{13.32}$$

The term A is

$$A = N_B \ (f_P) - N_B \ (f) \tag{13.33}$$

where $N_B(f)$ is the band number of the octave of interest and $N_B(f_P)$ is the band number of the octave where the peak frequency occurs. Octave band numbers are 0 for 32 Hz, 1 for 63 Hz, 2 for 125 Hz, and so forth. So, for example, if the peak frequency falls within the 125 Hz octave band, the band number is 2. The value of A for the 63 Hz octave band would be $A = 2 - 1 = 1$ and would decrease by 1 for each octave above that. The shape of the diffuser spectrum curve is given in Fig. 13.24.

FIGURE 13.24 Generalized Shape of the Diffuser Spectrum (ASHRAE, 1995)



Dampers located close to an outlet diffuser can add appreciably to the noise generated by the termination. First the damper generates vortex shedding in its wake, which is a source of noise. Second, downstream turbulence increases the noise generated by the grille. Manufacturers of these devices can provide sound power levels for a given flow volume. If these are not available it can be assumed that levels will increase 5 dB in all bands with the dampers in the fully open position. As the dampers are closed, there is an increase in the pressure drop across the damper, which restricts the flow. The overall sound power level increases approximately as (Fry, 1988)

$$\Delta L_w = 33 \log (\Delta P/\Delta P_0) \tag{13.34}$$

where ΔL_W = increase in sound power level radiated by the diffuser (dB)
ΔP = new static pressure drop across the unit (in. w.g.)
ΔP_0 = initial static pressure drop with the damper in place but with the vanes fully open with the same flow volume (in. w.g.)

Figure 13.25 shows the effect of integral dampers on noise radiated by ceiling diffusers for various settings. For dampers and grills to act as separate sources they should be located at least 4 duct diameters apart.

13.5 NOISE FROM OTHER MECHANICAL EQUIPMENT

Air Compressors

Air compressors in buildings generally fall into two categories: small units, under 5 hp, used to provide high pressure air for pneumatically controlled HVAC systems; and large units of up to 100 hp, which provide shop air to machine shops, laboratories, and maintenance areas. Miller (1980) has published sound pressure levels at 3 ft based on measurements of nine machines. Seven of these were reciprocating with motors ranging from 1 to 75 hp, and two were centrifugal, one of 10 hp and another 20 hp. Figure 13.26 reproduces his results. The principal source of air compressor noise is the air intake, which can be treated using a muffler between the filter and the intake manifold.

FIGURE 13.25 Ceiling Diffuser Damper—Sound Power Spectra (Fry, 1988)

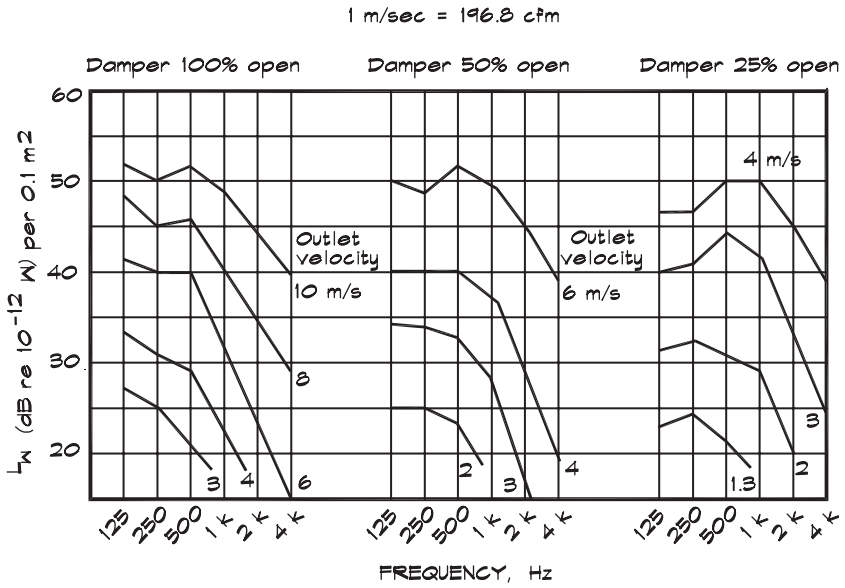
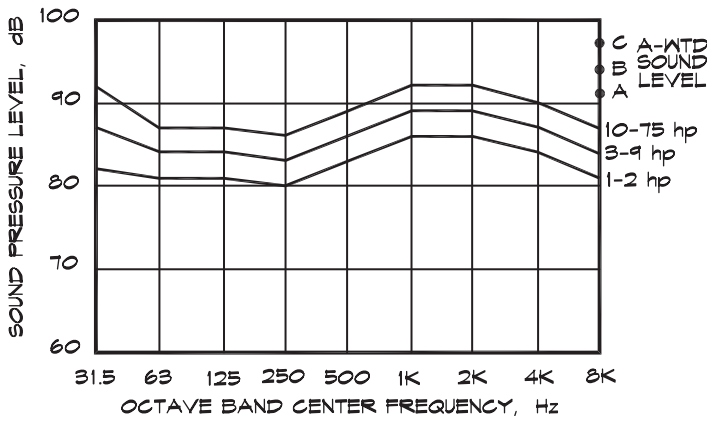


FIGURE 13.26 Sound Pressure Levels of Air Compressors at a Distance of 3 Feet (Miller, 1980)



Transformers

Transformers are usually located in an electrical equipment room, where they are sequestered from sensitive receivers. Ideally these rooms do not have common walls or floor-ceiling separations. Transformer noise is created through a process of magnetostriction, an expansion and contraction due to a magnetic field, caused by current within the coils. For a sinusoidal input voltage this phenomenon occurs twice every cycle, at 120 Hz, in single phase units and at harmonics of this frequency. Miller (1980) has published a conversion equation to obtain the sound power spectrum from the manufacturer's NEMA rating (the average of A-weighted sound pressure levels taken at a distance of 1 ft from an imaginary vertical surface passing

TABLE 13.10 Level Adjustments for the NEMA Rating of a Transformer, dB (Miller, 1980)

	Octave Band Center Frequency, Hz								
	31	63	125	250	500	1k	2k	4k	8k
C_T	-1	5	7	2	2	-4	-9	-14	-21

through a string tied around the unit), and the correction term, which includes the 10.5 dB adjustment for the power to pressure conversion. Table 13.10 gives the correction term for an unenclosed transformer.

$$L_w = \text{NEMA rating} + 10 \log S_T + C_T \tag{13.35}$$

where $L_{w\ OCT}$ = octave band sound pressure level (dB)
S_T = surface area of the transformer (ft^2)
C_T = octave band correction (dB)

Over time, transformers can grow noisier as their laminations and tie bolts become loose. Miller cites increases as large as 5 dB at the fundamental and 10 dB in the second and third harmonic frequencies. When transformers are enclosed in small vaults they can induce standing wave patterns in the room, which have the effect of increasing the transmitted power by 6 dB in the same bands.

Transformers that are directly tied to a wall can induce structure-borne noise. Jones (1984) recommends isolation techniques shown in Fig. 13.27 to prevent this. In areas of seismic activity, one or more sway braces may be necessary to provide stability at the top of the unit.

Reciprocating Engines and Emergency Generators

Most large buildings have emergency generators to provide power when the normal sources fail. It is often argued that noise control of emergency generators is unnecessary since they would be used only in an emergency, when noise is a secondary concern. Although this is probably true, generators must be tested periodically, an hour a month, and during these test periods the building functions normally and noise is still a concern. During power outages generators can be needed over longer periods of time.

Generator sets are powered by a diesel, methane, or propane fuel reciprocating engine and radiate sound from their casing, intake, and exhaust. Miller (1980) measured the casing radiated power levels, which followed the relationship

$$L_w = 93 + 10 \log (\text{rated hp}) + A + B + C + D \tag{13.36}$$

where L_w = overall sound power level (dB)
rated hp = engine manufacturer's continuous full load rating for the engine, (horse power)
A, B, C, D = correction terms given in Table 13.11 (dB)

Octave-band casing-radiated noise can be obtained from the overall sound power level spectrum by subtracting the levels given in Table 13.12.

FIGURE 13.27 Vibration Isolation of Floor-Mounted Transformers (Jones, 1984)

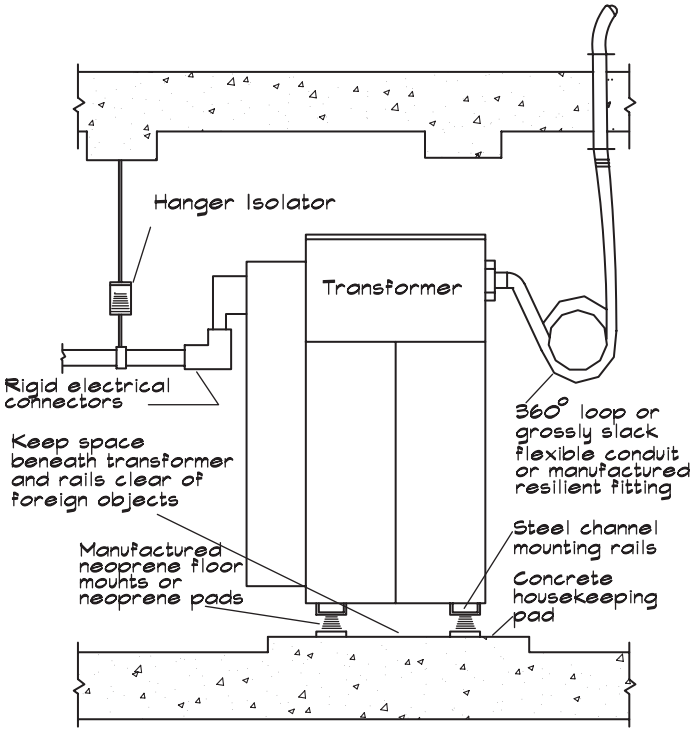


TABLE 13.11 Level Adjustments for Engine Casing Radiated Noise, dB (Miller, 1980)

Speed Correction Term, A	
Under 600 rpm	**−5**
600 - 1500 rpm	**−2**
Above 1500 rpm	**0**
Fuel Correction Term, B	
Diesel fuel only	**0**
Diesel and/or natural gas	**0**
Natural gas only (may have small amounts of "pilot oil")	**−3**
Cylinder Arrangement Term, C	
In-line	**0**
V-type	**−1**
Radial	**−1**
Air Intake Correction Term, D	
Unducted air inlet to unmuffled Roots Blower	**+3**
Ducted air from outside the room or into muffled Roots Blower	**0**
All other inlets to engine (with or without turbochargers)	**0**

Noise radiated from the inlet is usually the same as the casing radiation unless there is a separate ducted inlet to a turbocharger. In these cases the inlet noise is given by

$$L_w = 94 + 5 \log (\text{rated hp}) \tag{13.37}$$

TABLE 13.12 Frequency Adjustments for Casing Radiated Noise of Reciprocating Engines (Miller, 1980)

Octave Frequency Band (Hz)	Value to be Subtracted from Sound Power Level, dB			
	Engine Speed Under 600 rpm	Engine Speed 600-1500 rpm		Engine Speed Over 1500 rpm
		Without Roots Blower	With Roots Blower	
31	12	14	22	22
63	12	9	16	14
125	6	7	18	7
250	5	8	14	7
500	7	7	3	8
1000	9	7	4	6
2000	12	9	10	7
4000	18	13	15	13
8000	28	19	26	20
A	4	3	1	2

TABLE 13.13 Level Adjustments for Turbocharger Air Inlet, dB (Miller, 1980)

	Octave Band Center Frequency (Hz)									
	31	63	125	250	500	1k	2k	4k	8k	A
Correction	4	11	13	13	12	9	8	9	17	3

Any losses due to inlet ductwork or silencers must be subtracted from the octave band sound power levels. The corrections for each octave band are given in Table 13.13 and are subtracted from the overall sound power level.

The exhaust is the loudest source. The overall sound power level for noise radiated from an unmuffled engine exhaust is

$$L_w = 119 + 10 \log (\text{rated hp}) - T \tag{13.38}$$

where the factor T is the turbocharger correction term (T = 0 dB for no turbocharger and T = 6 dB for an engine with a turbocharger). The effects of any downstream exhaust piping or mufflers must be subtracted from the sound power level in each band. Octave-band adjustments to be subtracted form the overall sound power level are shown in Table 13.14.

TABLE 13.14 Level Adjustments for Engine Exhaust, dB (Miller, 1980)

	Octave Band Center Frequency (Hz)									
	31	63	125	250	500	1k	2k	4k	8k	A
Correction	5	9	3	7	15	19	25	35	43	12

14 SOUND ATTENUATION IN DUCTS

14.1 SOUND PROPAGATION THROUGH DUCTS

When sound propagates through a duct system it encounters various elements that provide sound attenuation. These are lumped into general categories, including ducts, elbows, plenums, branches, silencers, end effects, and so forth. Other elements such as tuned stubs and Helmholtz resonators can also produce losses; however, they rarely are encountered in practice. Each of these elements attenuates sound by a quantifiable amount, through mechanisms that are relatively well understood and lead to a predictable result.

Theory of Propagation in Ducts with Losses

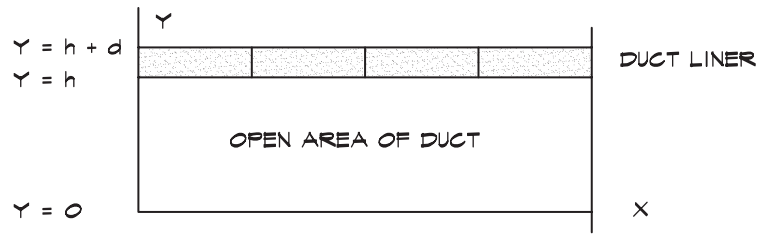
Noise generated by fans and other devices is transmitted, often without appreciable loss, from the source down an unlined duct and into an occupied space. Since ducts confine the naturally expanding acoustical wave, little attenuation occurs due to geometric spreading. So efficient are pipes and ducts in delivering a sound signal in its original form, that they are still used on board ships as a conduit for communications. To obtain appreciable attenuation, we must apply materials such as a fiberglass liner to the duct's inner surfaces to create a loss mechanism by absorbing sound incident upon it.

In Chapt. 8, we examined the propagation of sound waves in ducts without resistance and the phenomenon of cutoff. Recall that cutoff does not imply that all sound energy is prevented from being transmitted along a duct. Rather, it means that only particular waveforms propagate at certain frequencies. Below the cutoff frequency only plane waves are allowed, and above that frequency, only multimodal waves propagate.

In analyzing sound propagation in ducts it is customary to simplify the problem into one having only two dimensions. A duct, shown in Fig. 14.1, is assumed to be infinitely wide (in the x dimension) and to have a height in the y dimension equal to h. The sound wave travels along the z direction (out of the page) and its sound pressure can be written as (Ingard, 1994)

$$\mathbf{p}\,(y, z, \omega) = A\,\cos\,(\mathbf{q}_y\,y)\,e^{j\,\mathbf{q}_z\,z} \tag{14.1}$$

FIGURE 14.1 Coordinate System for Duct Analysis



where $\mathbf{p}$ = complex sound pressure (Pa)
A = pressure amplitude (Pa)
$\mathbf{q}_z$ and $\mathbf{q}_y$ = complex propagation constant in the z and y directions (m^{-1})

$j = \sqrt{-1}$
$\omega = 2\pi f$ (rad/s)

The propagation constants have real and imaginary parts as we saw in Eq. 7.79, which can be written as

$$\mathbf{q} = \delta + j\beta \tag{14.2}$$

where the z-axis subscript has been dropped. The value of the imaginary part of the propagation constant, β, in the z direction is dependent on the propagation constant in the y direction, and the normal acoustic impedance of the side wall of the duct or any liner material attached to it. The propagation constants are complex wave numbers and are related vectorially in the same way wave numbers are. As we found in Eq. 8.19,

$$\mathbf{q}_z = \sqrt{(\omega/c)^2 - \mathbf{q}_y^2} \tag{14.3}$$

At the side wall boundary the amplitude of the velocity in the y direction can be obtained from Eq. 14.1

$$\mathbf{u}_y = \frac{1}{-j\omega\rho_0}\frac{\partial \mathbf{p}}{\partial y} = \frac{A}{-j\omega\rho_0}\mathbf{q}_y \sin(\mathbf{q}_y y)\, e^{j q_z z} \tag{14.4}$$

The boundary condition at the surface of the absorptive material at y = h is

$$\frac{\mathbf{u}_y}{\mathbf{p}} = \frac{1}{\mathbf{z}_n} \tag{14.5}$$

where $\mathbf{z}_n$ is the normal specific acoustic impedance of the side wall panel material. Substituting Eqs. 14.1 and 14.4 into 14.5 we obtain

$$\mathbf{q}_y h \tan(\mathbf{q}_y h) = \frac{-j k h \rho_0 c_0}{\mathbf{z}_n} \tag{14.6}$$

where $k = \omega/c$. Values of the normal impedance for fiberglass materials were given in Chapt. 7 by the Delany and Blazey (1969) equations. Once the material impedance $\mathbf{z}_n$ has

been obtained, the propagation constant $\mathbf{q}_y$ can be extracted numerically from Eq. 14.6. Equation 14.3 then gives us a value for $\mathbf{q}_z$, from which we can solve for its imaginary part, β, in nepers/ft.

The ratio of the pressure amplitudes at two values of z is obtained from Eq. 14.1

$$\left|\frac{\mathbf{p}(0)}{\mathbf{p}(z)}\right| = e^{\beta z} \tag{14.7}$$

from which we obtain the loss in decibels over a given distance l

$$\Delta L_{duct} = 20 \log \left|\frac{\mathbf{p}(0)}{\mathbf{p}(z)}\right| = 20(\beta l) \log(e) \cong 8.68 \beta l \tag{14.8}$$

In this way we can calculate the attenuation from the physical properties of the duct liner. The lined duct configurations shown in Fig. 14.2 yield equal losses in the lowest mode. The splitters shown on the right of the figure are representative of the configuration found in a duct silencer.

In practice, lined ducts and silencers are tested in a laboratory by substituting the test specimen for an unlined sheet metal duct having the same face dimension. Figure 14.3 shows measured losses for a lined rectangular duct. The data take on a haystack shape that shifts slightly with flow velocity. At low frequencies the lining is too thin, compared with a wavelength, to have much effect. At high frequencies the sound waves beam and the interaction with the lining at the sides of the duct is minimal. The largest losses are at the mid frequencies, as evidenced by the peak in the data.

Air flow affects the attenuation somewhat. When the sound propagates in the direction of flow, it spends slightly less time in the duct so the low-frequency losses are slightly less. The high frequencies are influenced by the velocity profile, which is higher in the center

FIGURE 14.2 Equivalent Duct Configurations (Ingard, 1994)

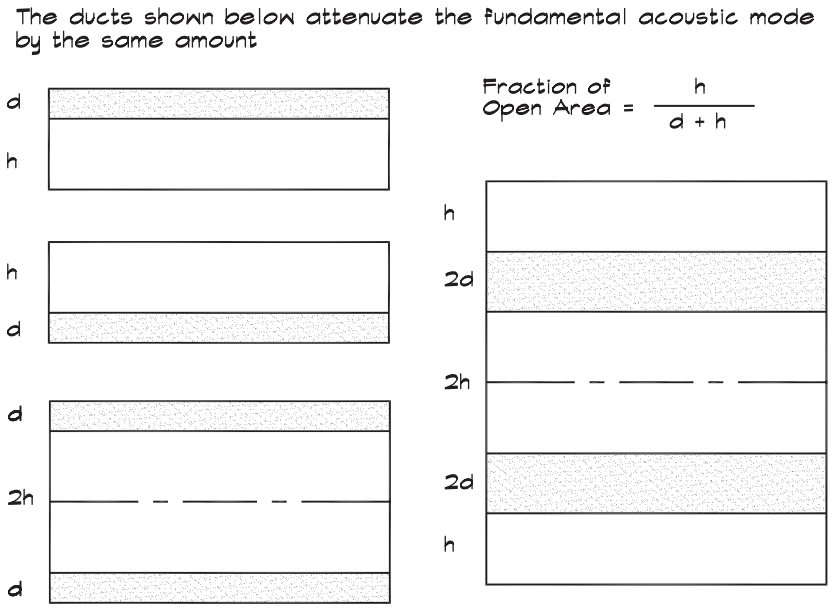
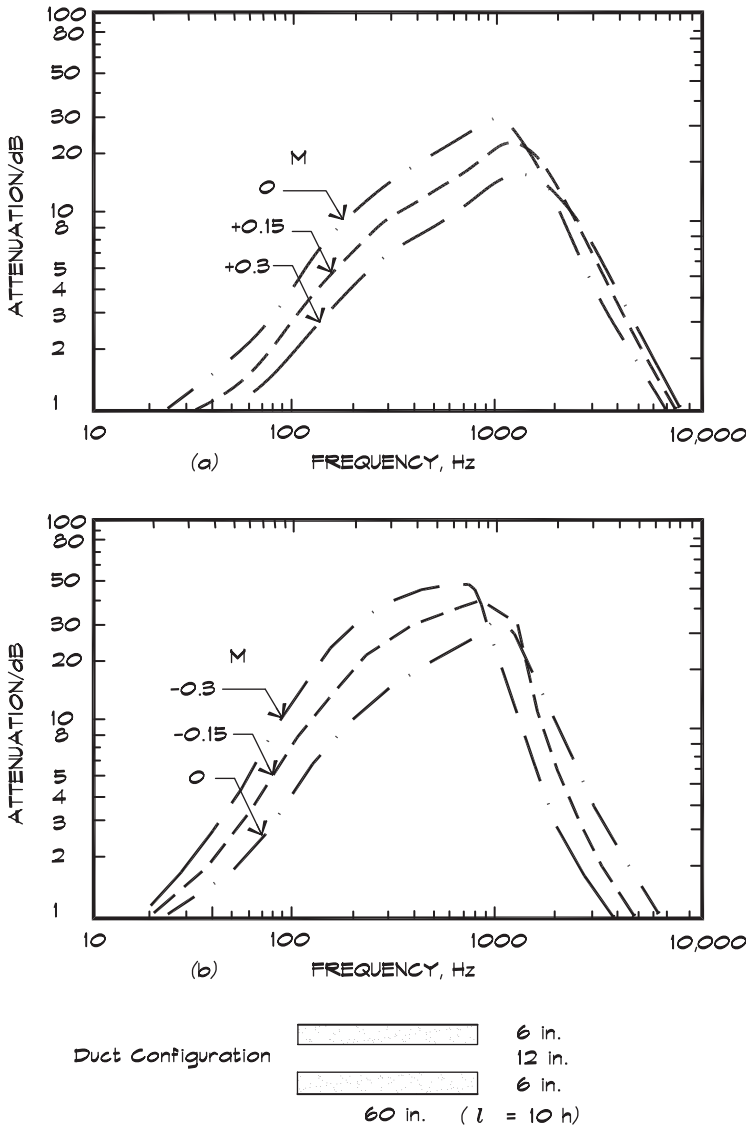


FIGURE 14.3 Attenuation in a Lined Duct (Beranek and Ver, 1992)

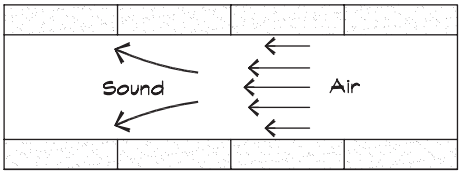
The effect of flow on the sound attenuation for Mach numbers M = 0, 0.15, 0.3 for a) sound propagation in the direction of flow and b) sound propagation against the direction of flow



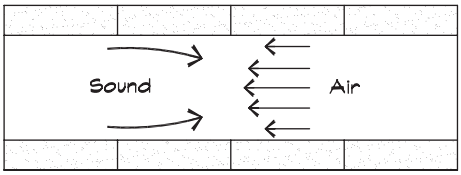
of the duct. The gradient refracts the high frequency energy toward the duct walls yielding somewhat greater losses for propagation in the downstream direction. Figure 14.4 shows this effect.

Ingard (1994) published a generalized design chart in Fig. 14.5 for lined rectangular ducts giving the maximum attainable attenuation in terms of the percentage of the open area of the duct mouth and the resistivity of the liner. The chart is useful for visualizing the effectiveness of lined duct, as well as for doing a calculation of attenuation. The peak in the

FIGURE 14.4 Influence of Air Velocity on Attenuation (IAC Corp., 1989)

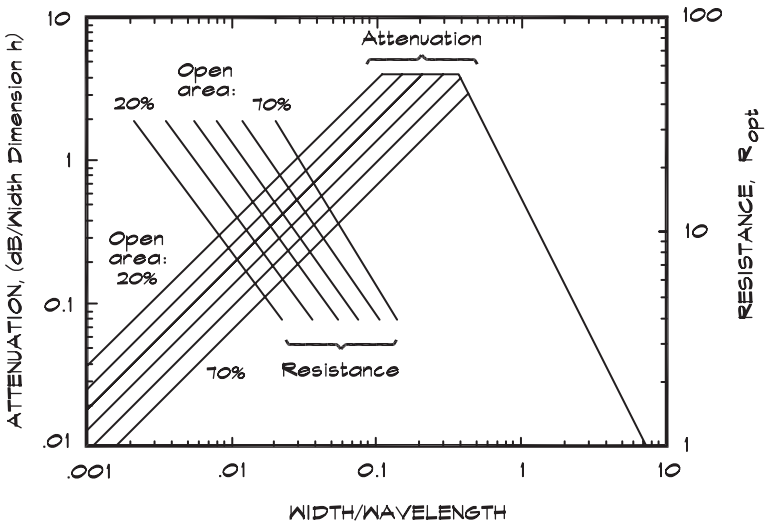


Under FORWARD FLOW conditions, high-frequency sound is refracted toward the duct walls.



Under REVERSE FLOW conditions, high-frequency sound is refracted away from the walls and toward the center of the duct.

FIGURE 14.5 Approximate Relationship between Optimized Liner Resistance and the Corresponding Maximum Attenuation (Ingard, 1994)



Relationship between the optimized liner resistance and the maximum attenuation that can be achieved for the fundamental mode in a rectangular duct with one side lined with a locally reacting porous liner. The left axis is the attenuation in a length of duct equal to the air channel width h and the right axis is the optimum value of the normalized flow resistance.

curve is extended downward in frequency with lower open area percentages. Thus there is a tradeoff between low-frequency attenuation and back pressure.

Attenuation in Unlined Rectangular Ducts

As a sound wave propagates down an unlined duct, its energy is reduced through induced motion of the duct walls. The surface impedance is due principally to the wall mass, and the duct loss calculation goes much like the derivation of the transmission loss. Circular sheet-metal ducts are much stiffer than rectangular ducts at low frequencies, particularly in their first mode of vibration, called the breathing mode, and therefore are much more difficult to excite. As a consequence, sound is attenuated in unlined rectangular ducts to a much greater degree than in circular ducts.

Since a calculation of the attenuation from the impedance of the liner is complicated, measured values or values calculated from simple empirical relationships are used. Empirical equations for the attenuation can be written in terms of a duct perimeter to area ratio. A large P/S ratio means that the duct is wide in one dimension and narrow in the other, which implies relatively flexible side walls. The attenuation of rectangular ducts in the 63 Hz to 250 Hz octave frequency bands can be approximated by using an equation by Reynolds (1990)

$$\Delta L_{\text{duct}} = 17.0 \left(\frac{P}{S}\right)^{-0.25} f^{-0.85} l \tag{14.9}$$

for $\frac{P}{S} \geq 3$ and

$$\Delta L_{\text{duct}} = 1.64 \left(\frac{P}{S}\right)^{0.73} f^{-0.58} l \tag{14.10}$$

for $\frac{P}{S} < 3$.

Note that these formulas are unit sensitive. The perimeter must be in feet, the area in square feet, and the length, l, in feet. Above 250 Hz the loss is approximately

$$\Delta L_{\text{duct}} = 0.02 \left(\frac{P}{S}\right)^{0.8} l \tag{14.11}$$

When the duct is externally wrapped with a fiberglass blanket the surface mass is increased, and so is the low-frequency attenuation. Under this condition, the losses given in Eqs. 14.9 and 14.10 are multiplied by a factor of two.

Attenuation in Unlined Circular Ducts

Unlined circular ducts have about a tenth the loss of rectangular ducts. Typical losses are given in Table 14.1

TABLE 14.1 Losses in Unlined Circular Ducts

Frequency (Hz)	63	125	250	500	1000	2000	4000
Loss (dB/ft)	0.03	0.03	0.03	0.05	0.07	0.07	0.07

TABLE 14.2 Constants Used in Eq. 14.12 (Reynolds, 1990)

	Octave Band Center Frequency (Hz)							
	63	125	250	500	1000	2000	4000	8000
B	0.0133	0.0574	0.2710	1.0147	1.7700	1.3920	1.5180	1.5810
C	1.959	1.410	0.824	0.500	0.695	0.802	0.451	0.219
D	0.917	0.941	1.079	1.087	0.000	0.000	0.000	0.000

Attenuation in Lined Rectangular Ducts

When a duct is lined with an absorbent material such as a treated fiberglass board, sound propagating in the duct is attenuated through its interaction with the material as discussed earlier. A regression equation for the insertion loss of rectangular ducts has been published by Reynolds (1990).

$$\Delta L_{duct} = B \left(\frac{P}{S}\right)^{C} t^{D} l \tag{14.12}$$

where P = perimeter of the duct (ft)
S = area of the duct (sq ft)
l = length of the duct (ft)
t = thickness of the lining (inches)

Table 14.2 lists the constants B, C, and D.

Reynolds' equation was based on data using a 1 to 2 inch (25 mm to 51 mm) thick liner having a density of 1.5 to 3 lbs / ft^3 (24 to 48 kg / m^3). Linings less than 1 inch (25 mm) thick are generally ineffective. The P/S ratios ranged from 1.1667 to 6, in units of feet. The equation is valid within these ranges.

The insertion loss of ducts is measured by substituting a lined section for an unlined section and reporting the difference. Since there may be a significant contribution to the overall attenuation furnished by the induced motion of the side walls, the unlined attenuation should be added to the lined attenuation to obtain an overall value.

Attenuation of Lined Circular Ducts

An empirical equation for the losses in lined circular ducts has been developed in the form of a third order polynomial regression by Reynolds (1990).

$$\Delta L_{d} = \left[A + B\,t + C\,t^2 + D\,d + E\,d^2 + F\,d^3\right] l \tag{14.13}$$

where t = thickness of the lining (inches)
d = interior diameter of the duct (inches)
l = length of the duct (ft)

The constants are given in Table 14.3

Reynolds developed this relationship for spiral ducts having 0.75 lb/cu ft (12 kg/cu m) density fiberglass lining in thicknesses ranging from 1 to 3 inches (25 to 76 mm) thick with a 25% open perforated metal facing. The inside diameters of the tested ducts ranged from 6 to 60 inches (0.15 to 1.5 m).

TABLE 14.3 Constants Used in Eq. 14.13 (Reynolds, 1990)

Freq., Hz	A	B	C	D	E	F
63	0.2825	0.3447	−5.251E-2	−3.837E-2	9.132E-4	−8.294E-6
125	0.5237	0.2234	−4.936E-3	−2.724E-2	3.377E-4	−2.490E-6
250	0.3652	0.7900	−0.1157	−1.834E-2	−1.211E-4	2.681E-6
500	0.1333	1.8450	−0.3735	−1.293E-2	8.624E-5	−4.986E-6
1000	1.9330	0.0000	0.0000	6.135E-2	−3.891E-3	3.934E-5
2000	2.7300	0.0000	0.0000	−7.341E-2	4.428E-4	1.006E-6
4000	2.8000	0.0000	0.0000	−0.1467	3.404E-3	−2.851E-5
8000	1.5450	0.0000	0.0000	−5.452E-2	1.290E-3	−1.318E-5

Because of flanking paths, the duct attenuation in both round and rectangular ducts is limited to 40 dB. As with rectangular ducts, the unlined attenuation may be added to the lined attenuation. For circular ducts it is such a small contribution that it is usually ignored.

Flexible and Fiberglass Ductwork

It is frequently the case that the last duct run in a supply branch is made with a round flexible duct with a lightweight fiberglass fill, surrounded on the outside with a light plastic membrane, and lined on the inside with a fabric liner. The published insertion losses of these flexible ducts are quite high, sometimes as much as 2 to 3 dB per foot or more. Table 14.4 is based on data published in ASHRAE (1995).

Since the insertion loss testing is done by replacing a section of unlined sheet-metal duct with the test specimen, some of the low-frequency loss obtained from flexible duct occurs due to breakout. This property can be used to advantage, since in tight spaces where there is little room for a sound trap, flexible duct surrounded with fiberglass batt can be used to construct a breakout silencer. Such a silencer can be built between joists in a floor-ceiling to isolate exterior noise that might otherwise enter a dwelling through an exhaust duct attached to a bathroom fan. A serpentine arrangement of flexible duct 6 to 8 feet in length in an attic can often control the noise from a fan coil unit located in this space, so long as there is a drywall or plaster ceiling beneath it.

The transmission loss properties of flexible duct are not well documented; however, a conservative approach is to assume that the flex duct is not present and to calculate the insertion loss of the ceiling material. When the attenuation of the flexible duct is greater than the insertion loss of the ceiling the latter is used.

End Effect in Ducts

When a sound wave propagates down a duct and encounters a large area expansion, such as that provided by a room, there is a loss due to the area change known as the end effect. The end effect does not always follow the simple relationship shown in Chapt. 8, which was derived assuming that the lateral dimensions of both ducts were small compared with a wavelength. At low frequencies sound waves expand to the boundaries of the duct. At very high frequencies the sound entering a room from a duct tends to radiate like a piston

TABLE 14.4 Lined Flexible Duct Insertion Loss, dB (ASHRAE, 1995)

Diameter (in/mm)	Length (ft/m)	Octave Band Center Frequency—Hz						
		63	125	250	500	1000	2000	4000
4/100	12/3.7	6	11	12	31	37	42	27
	9/2.7	5	8	9	23	28	32	20
	6/1.8	3	6	6	16	19	21	14
	3/0.9	2	3	3	8	9	11	7
5/127	12/3.7	7	12	14	32	38	41	26
	9/2.7	5	9	11	24	29	31	20
	6/1.8	4	6	7	16	19	21	13
	3/0.9	2	3	4	8	10	10	7
6/152	12/3.7	8	12	17	33	38	40	26
	9/2.7	6	9	13	25	29	30	20
	6/1.8	4	6	9	17	19	20	13
	3/0.9	2	3	4	8	10	10	7
7/178	12/3.7	8	12	19	33	37	38	25
	9/2.7	6	9	14	25	28	29	19
	6/1.8	4	6	10	17	19	19	13
	3/0.9	2	3	5	8	9	10	6
8/203	12/3.7	8	11	21	33	37	36	22
	9/2.7	6	8	16	25	28	28	18
	6/1.8	4	6	11	17	19	19	12
	3/0.9	2	3	5	8	9	9	6
9/229	12/3.7	8	11	22	33	37	36	22
	9/2.7	6	8	17	25	28	27	17
	6/1.8	4	6	11	17	19	18	11
	3/0.9	2	3	6	8	9	9	6
10/254	12/3.7	8	10	22	32	36	34	21
	9/2.7	6	8	17	24	27	26	16
	6/1.8	4	5	11	16	18	17	11
	3/0.9	2	3	6	8	9	9	5
12/305	12/3.7	7	9	20	30	34	31	18
	9/2.7	5	7	15	23	26	23	14
	6/1.8	3	5	10	15	17	16	9
	3/0.9	2	2	5	8	9	8	5
14/356	12/3.7	5	7	16	27	31	27	14
	9/2.7	4	5	12	20	23	20	11
	6/1.8	3	4	8	14	16	14	7
	3/0.9	1	2	4	7	8	7	4
16/406	12/3.7	2	4	9	23	28	23	9
	9/2.7	2	3	7	17	21	17	7
	6/1.8	1	2	5	12	14	12	5
	3/0.9	1	1	2	6	7	6	2

in a baffle and forms a beam. Therefore it does not interact with the sides of the duct and is relatively unaffected by the end effect. An empirical formula for calculating end effect has been published by Reynolds (1990). Its magnitude depends on the size of the duct, measured in wavelengths. This is expressed in the formula as a frequency-width product. The attenuation associated with a duct terminated in free space is

$$\Delta L_{end} = 10 \log \left[1 + \left(\frac{c_0}{\pi f d} \right)^{1.88} \right] \tag{14.14}$$

and for a duct terminated flush with a wall

$$\Delta L_{end} = 10 \log \left[1 + \left(\frac{0.8\ c_0}{\pi f d} \right)^{1.88} \right] \tag{14.15}$$

where d is the diameter of the duct in units consistent with those of the sound velocity. If the duct is rectangular the effective diameter is

$$d = \sqrt{\frac{4S}{\pi}} \tag{14.16}$$

where S is the area of the duct. End effect attenuation does not occur when the duct is terminated in a diffuser, since these devices smooth the impedance transition between the duct and the room.

Split Losses

When there is a division of the duct into several smaller ducts there is a distribution of the sound energy among the various available paths. The loss is derived in much the same way as was Eq. 8.32; however, multiple areas are taken into account. The split loss in propagating from a main duct into the ith branch is

$$\Delta L_{split} = 10 \log \left[1 - \left(\frac{\sum S_i - S_m}{\sum S_i + S_m} \right)^2 \right] + 10 \log \left[\left(\frac{S_i}{\sum S_i} \right) \right] \tag{14.17}$$

where S_m = area of the main feeder duct (ft^2 or m^2)
S_i = area of the i th branch (ft^2 or m^2)
$\sum S_i$ = total area of the individual branches that continue on from the main duct (ft^2 or m^2)

The first term in Eq. 14.17 comes from reflection, which occurs from the change in area, when the total area of the branches is not the same as the area of the main duct and the frequency is below cutoff. The second term comes from the division of acoustic power among the individual branches, which is based on the ratio of their areas.

Elbows

A sharp bend or elbow can provide significant high-frequency attenuation, particularly if it is lined. In order for a bend to be treated as an elbow its turn angle must be greater than 60°.

TABLE 14.5 Insertion Loss of Unlined and Lined Square Elbows without Turning Vanes

	Insertion Loss, dB	
f w	**Unlined**	**Lined**
f w < 1.9	**0**	**0**
1.9 < f w < 3.8	**1**	**1**
3.8 < f w < 7.5	**5**	**6**
7.5 < f w < 15	**8**	**11**
15 < f w < 30	**4**	**10**
f w > 30	**3**	**10**

The term f w = f times w, where f is the octave-band center frequency (kHz) and w is the width of the elbow (in).

TABLE 14.6 Insertion Loss of Unlined and Lined Square Elbows with Turning Vanes

	Insertion Loss, dB	
f w	**Unlined**	**Lined**
f w < 1.9	**0**	**0**
1.9 < f w < 3.8	**1**	**1**
3.8 < f w < 7.5	**4**	**4**
7.5 < f w < 15	**6**	**7**
f w > 15	**4**	**7**

The losses in unlined elbows are minimal, particularly if the duct is circular.

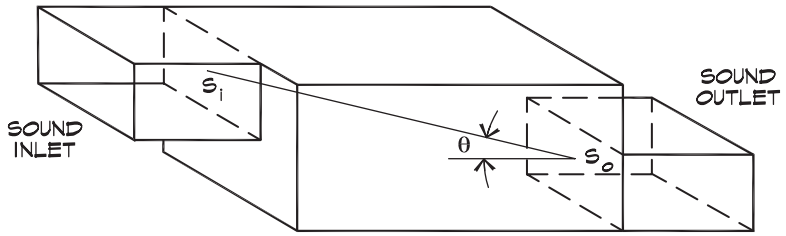
TABLE 14.7 Insertion Loss of Round Elbows

f w	**Insertion Loss, dB**
f w < 1.9	**0**
1.9 < f w < 3.8	**1**
3.8 < f w < 7.5	**2**
f w > 15	**3**

To be considered a lined elbow, the lining must extend two duct widths (in the plane of the turn) beyond the outside of the turn, and the total thickness of both sides must be at least 10% of the duct width. Reynolds (1990) has published data on unlined rectangular elbows, given in Tables 14.5 and 14.6 and for round elbows, shown in Table 14.7.

Lined round elbow losses can be calculated using an empirical regression formula published by Reynolds (1990). The testing was done on double-wall circular ducts having a perforated inner wall, with an open area of 25%, and the space between filled with 0.75 lb/cu ft (12 kg/cu m) fiberglass batt, between 1 to 3 inches (25–75 mm) thick. The ducts ranged

FIGURE 14.6 Schematic of a Plenum Chamber



from 6 inches to 60 inches (150–1500 mm) in diameter. For elbows where $6 \leq d \leq 18$ inches (150–750 mm),

$$\Delta L_e \left(\frac{d}{r}\right)^2 = 0.485 + 2.094 \log(fd) + 3.172 [\log(fd)]^2 - 1.578 [\log(fd)]^4 + 0.085 [\log(fd)]^7 \quad (14.18)$$

and for elbows where $18 < d \leq 60$ inches (750–1500 mm),

$$\Delta L_e \left(\frac{d}{r}\right)^2 = -1.493 + 0.538\, t + 1.406 \log(fd) + 2.779 [\log(fd)]^2 - 0.662 [\log(fd)]^4 + 0.016 [\log(fd)]^7 \quad (14.19)$$

where ΔL_e = attenuation due to the elbow (dB)
d = diameter of the duct (in)
r = radius of the elbow at its centerline (in)
t = thickness of the liner (in)
f = center frequency of the octave band (kHz)

Note that if calculated values are negative, the loss is set to zero. In the ducts tested, the elbow radius geometry followed the relationship $r = 1.5\, d + 3\, t$.

14.2 SOUND PROPAGATION THROUGH PLENUMS

A plenum is an enclosed space that has a well-defined entrance and exit, which is part of the air path, and that includes an increase and then a decrease in cross-sectional area. The geometry is shown in Fig. 14.6. A return-air plenum located above a ceiling may or may not be an acoustical plenum. If it is bounded by a drywall or plaster ceiling, it can be modeled as an acoustic plenum; however, if the ceiling is constructed of acoustical tile, it is usually not. Rooms that form part of the air passageway are modeled as plenums. For example a mechanical equipment room can be a plenum when the return air circulates through it. In this case the intake air opening on the fan is the plenum entrance.

Plenum Attenuation—Low-Frequency Case

Plenum attenuation depends on the relationship between the size of the cavity and the wavelength of the sound passing through it. When the wavelength is large compared with the cross-sectional dimension—that is, below the duct cutoff frequency—a plenum is modeled as a muffler, using plane wave analysis. This approach follows the same methodology

used in Chapt. 8 for plane waves incident on an expansion and contraction, which was treated in Eqs. 8.35 and 8.36. The transmissivity can be written in terms of the area ratio $m = S_2/S_1$

$$\alpha_t = \frac{4}{4\cos^2 k l + \left(m + \frac{1}{m}\right)^2 \sin^2 k l} \tag{14.20}$$

When the plenum is a lined chamber having a certain duct loss per unit length, the wave number k within that space becomes a complex propagation constant **q**, having an imaginary term j β. The plenum attenuation is then given by (Davis et al., 1954)

$$\Delta L_p = 10 \log \left[\begin{array}{l} \left(\cosh[\beta l] + \frac{1}{2}\left[m + \frac{1}{m}\right] \sinh[\beta l]\right)^2 \\ \times \cos^2\left(\frac{2\pi f l}{c_0}\right) \\ + \left(\sinh[\beta l] + \frac{1}{2}\left[m + \frac{1}{m}\right] \cosh[\beta l]\right)^2 \\ \times \sin^2\left(\frac{2\pi f l}{c_0}\right) \end{array} \right] \tag{14.21}$$

The loss term, βl, due to the plenum liner, can be calculated using empirical equations for a lined rectangular duct (Reynolds, 1990)

$$\begin{aligned} 63\,\text{Hz} \quad & \beta l = \left[0.00153\ (P/S)^{1.959}\ t^{0.917}\right] l \\ 125\,\text{Hz} \quad & \beta l = \left[0.00662\ (P/S)^{1.410}\ t^{0.941}\right] l \\ 250\,\text{Hz} \quad & \beta l = \left[0.03122\ (P/S)^{0.824}\ t^{1.079}\right] l \\ 500\,\text{Hz} \quad & \beta l = \left[0.11690\ (P/S)^{0.500}\ t^{1.087}\right] l \end{aligned} \tag{14.22}$$

where β = attenuation in the open area of the plenum (nepers/ft or dB/8.68 ft)
P/S = perimeter of the cross - section of the plenum divided by the area (ft^{-1})
t = thickness of the fiberglass liner (in)
l = length of the plenum (ft)

Plenum Attenuation—High Frequency Case

When the wavelength is not large compared with the dimensions of the central cross section, the plane wave model is no longer appropriate, since the plenum behaves more like a room than a duct. Under these conditions we return to the methodology previously developed for the behavior of sound in rooms. First, we assume that the sound propagating down a duct and into a plenum is nearly plane, so the energy entering the plenum is

$$W_i = S_i I_i \tag{14.23}$$

and using Eq. 2.74, the direct field intensity at the outlet is

$$I_o = \frac{S_i I_i Q_i}{4\pi \left[r + \sqrt{\frac{S_i Q_i}{4\pi}} \right]^2} \tag{14.24}$$

The direct field energy leaving the plenum is

$$W_o = S_o \cos\theta \, I_o \tag{14.25}$$

and the ratio of the direct field outlet energy to the inlet energy is

$$\frac{W_o}{W_i} = \frac{Q_i S_o \cos\theta}{4\pi \left[r + \sqrt{\frac{S_i Q_i}{4\pi}} \right]^2} \tag{14.26}$$

A similar treatment can be done for the reverberant energy, with the intensity in a reverberant field, from Eqs. 8.79 and 8.83

$$I_o = \frac{W_i}{R} \tag{14.27}$$

so that the reverberant field power out is

$$W_o = S_o I_o = \frac{W_i S_o}{R} \tag{14.28}$$

Combining the direct and reverberant field contributions the overall transmission loss is

$$\Delta L_p = 10 \log \frac{W_o}{W_i} = 10 \log \left\{ \frac{Q_i S_o \cos\theta}{4\pi \left[r + \sqrt{\frac{S_i Q_i}{4\pi}} \right]^2} + \frac{S_o}{R} \right\} \tag{14.29}$$

where ΔL_p = attenuation due to the plenum (dB)
S_i = sound inlet area of the plenum (m^2 or ft^2)
Q_i = directivity of the inlet
S_o = sound outlet area of the plenum (m^2 or ft^2)
R = room constant of the plenum
$= S_p \overline{\alpha}/(1 - \overline{\alpha})$ (m^2 or ft^2)
S_p = interior surface area of the plenum (m^2 or ft^2)
θ = angle between the inlet and the outlet
r = distance between the inlet and the outlet (m or ft)

When the characteristic entrance dimension is small compared with the inlet to outlet distance, $\sqrt{\frac{S_i Q_i}{4\pi}} \ll r$, Eq. 14.28 can be simplified to (Wells, 1958)

$$\Delta L_p = 10 \log \left\{ \frac{S_o \cos\theta}{4\pi r^2} + \frac{S_o}{R} \right\} \tag{14.30}$$

which assumes that the inlet directivity is one.

These plenum equations begin with the assumption that the inlet and outlet waveforms are planar. At high frequencies the field at the exit can be semidiffuse rather than planar, particularly above the cutoff frequency. This is similar to the situation encountered in the transmission from a reverberant space through an open door, which was discussed in Chapt. 10. When the inlet condition is semidiffuse and the outlet condition is planar, the plenum is 3 dB more effective than Eq. 14.28 predicts, since there is added attenuation through the conversion of the waveform. If the outlet condition is semidiffuse and the inlet planar, the plenum is 3 dB less effective since there is more energy leaving than predicted by the plane wave relationship. Usually semidiffuse conditions occur when the inlet and outlet openings are large, so that the frequencies are above cutoff, and the upstream and downstream duct lengths are short. If both inlet and outlet conditions are semidiffuse, these relations still hold since the extra energy is passed along from the inlet to the outlet.

Sometimes, the Sabine absorption coefficients of plenum materials are greater than one at certain frequencies, and in many instances a large fraction of the plenum surface is treated with such a material. In these cases the average absorption coefficient may calculate out greater than one, and the room constant is not defined. As a practical guide, when the Norris Eyring room constant is employed a limiting value of the average absorption coefficient should be established, on the order of 0.98.

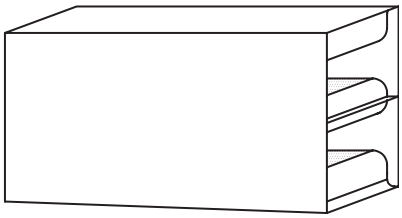
As was discussed previously, a mechanical plenum is not always an acoustic plenum. For example, if air is returned through the space above an acoustical tile ceiling, the return-air plenum is not an acoustic plenum since the noise breaks out of the space through the acoustical tile ceiling, which has a transmission loss lower than the theoretical plenum loss. The problem is treated as if the return-air duct entering the plenum were the source, and the insertion loss of the acoustical tile is subtracted from the sound power level along with the room correction factor to obtain the sound pressure level in the space. Figure 13.15 gives the insertion loss of acoustical tile materials (Blazier, 1981).

In other cases a duct may act as a plenum. If a flexible duct is enclosed in an attic filled with batt insulation, the sound breaks out of the duct and enters the attic plenum space. At the opposite end of the duct it breaks in again, completing the plenum path. This effect can provide significant low-frequency loss in a relatively short distance, particularly when the length of the duct is maximized by snaking. In this manner flexible ducts can be made into quasi-silencers by locating them in joist or attic spaces that are filled with batt.

14.3 SILENCERS

Silencers are commercially available attenuators specifically manufactured to replace a section of duct. They are available in standard lengths in one-foot increments between 3 and 10 feet, and sometimes in an elbow configuration. They consist of baffles of perforated

FIGURE 14.7 Duct Silencer Construction



metal filled with fiberglass, which alternate with open-air passage ways. An example is shown in Fig. 14.7.

Dynamic Insertion Loss

Silencer manufacturers publish dynamic insertion loss (DIL) data on their products. This is the attenuation achieved when a given length of unlined duct is replaced with a silencer. Insertion loss data are measured in both the upstream and downstream directions at various air velocities. As with lined ducts, silencer losses in the upstream direction are greater at low frequencies and less at high frequencies.

Insertion loss values are measured in third-octave bands and published as octave-band data. At very low frequencies, below 63 Hz, there are significant comb filtering effects, probably due to the silencer acting as a tuned pipe. In these regions it is more accurate to perform calculations in third-octave bands rather than in octaves. Figure 14.8 gives an example of measured data.

Self Noise

The flow of air through a silencer can generate *self noise*, and sound power level data are published by silencer manufcturers. Self-noise levels are measured on a 24" × 24" (600 × 600 mm) inlet area silencer, and a factor of 10 log (S/S_o) must be added to account

FIGURE 14.8 Silencer Dynamic Insertion Loss Data (PCI Industries, 1999)

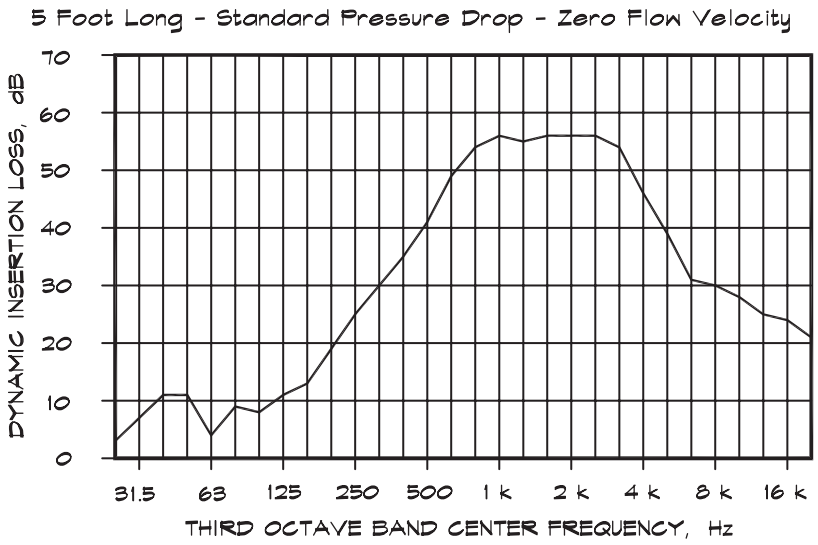


TABLE 14.8 Silencer Self Noise Octave Band Corrections (dB)

Freq. (Hz)	63	125	250	500	1k	2k	4k	8k
Correction	4	4	6	8	13	18	23	28

for the actual area of the silencer being used. In most cases $S_o = 4\,ft^2$ (0.37 sq m), but when the measurements were made on a different unit, the actual face area must be utilized. Self noise is the power radiated from the receiver end of the silencer and is combined with the sound power levels from other sources exiting the silencer. Most of the high-frequency self noise is generated at the air inlet, so it is attenuated in its passage through the silencer in the downstream direction but not in the upstream direction. Hence high-frequency (> 1k Hz) self noise levels are greater on the return-air side of an HVAC system. Low-frequency self-noise levels do not vary significantly with flow direction.

When self-noise data are not available, they can be estimated using (Fry, 1988)

$$L_w \cong 55 \log \frac{V}{V_0} + 10 \log N + 10 \log \frac{H}{H_0} - 45 \tag{14.31}$$

where L_w = sound power level generated by the silencer (dB)
V = velocity in the splitter airway (m/s or ft/min)
V_0 = reference velocity (1 for m/s and 196.8 for ft/min)
N = number of air passages
H = height or circumfrence (round) of the silencer (mm or in)
H_0 = reference height (1 for mm or 0.0394 for in)

The spectrum of noise generated by the silencer is calculated by subtracting octave band corrections given in Table 14.8 from the overall sound power level.

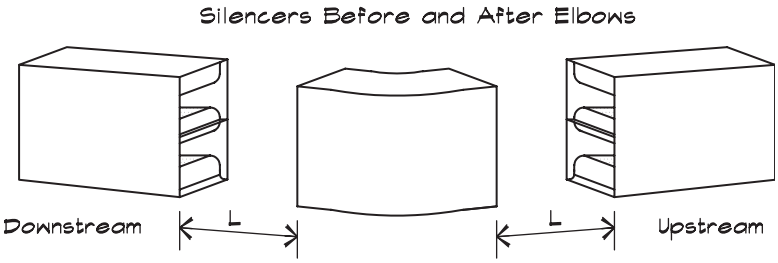
Back Pressure

Silencers create some additional back pressure or flow resistance due to the constriction they present. Silencers that minimize this pressure loss are available but there is generally a trade off between back pressure and low-frequency attenuation. Sometimes it is necessary to expand the duct to increase the silencer face area and reduce the pressure loss. It is desirable to minimize the silencer back pressure, usually limiting it to less than 10% of the total rated fan pressure. The position of the silencer in the duct, relative to other components, also affects the back pressure. Figure 14.9 shows published data (IAC Corp.) that give the multiplier of the standard back pressure for various silencer positions.

14.4 BREAKOUT

The phenomenon known as breakout describes the transmission of sound energy from the interior of the duct out through its walls and into an occupied space. The analysis of the process combines elements of duct attenuation as well as the transmission loss through the duct walls.

FIGURE 14.9 Duct Silencer Back Pressure Multipliers (Industrial Acoustics Corp., 1989)



Pressure Drop = Rated Pressure Drop x Multiplier

Elbows (without turning vanes)	Multiplier
L = D x 3	1.0
L = D x 2	1.5
L = D x 1	2.0
Elbows (with turning vanes)	
L = D x 3	1.0
L = D x 2	1.2
L = D x 1	1.75
L = D x 0.5	3.0
L = 0	4.0 (Not advised downstream)

Note: Silencer baffles should be parallel to the plane of the turn. Check manufacturer recommendations for multipliers associated with individual products.
D = Duct Diameter or Equivalent from Eq. 14.16

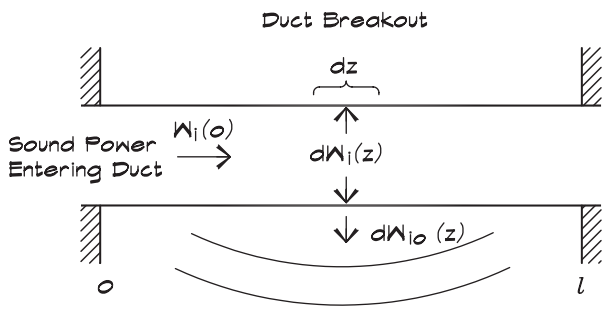
Transmission Theory

The breakout transmission loss defines the relationship between the sound power level entering an incremental slice of the duct at position z and that radiating out through the walls of that slice. Figure 14.10 illustrates the geometry (following Ver, 1983).

The breakout transmission loss at a given point is

$$\Delta L_{TLio} = 10 \log \left[\frac{d W_i (z)}{d W_{io} (z)} \right] \tag{14.32}$$

FIGURE 14.10 Duct Breakout Geometry



where the sound power incident on the increment of duct length, dz is

$$d\,W_i\,(z) = \left|I_i\,(z)\right|\,P\,d\,z = W_i\,(z)\,\frac{P}{S}\,d\,z \tag{14.33}$$

and the radiated power emanating from this slice on the outside of the duct is

$$d\,W_{io}\,(z) = d\,W_i\,(z)\,10^{-0.1\,\Delta\,L_{TLio}} = \frac{W_i\,(z)}{S}\,10^{-0.1\,\Delta\,L_{TLio}}\,P\,d\,z \tag{14.34}$$

If the internal sound power decreases with distance along the duct due to radiation through the duct walls and interaction with the interior surface, according to the relationship

$$W_i\,(z) = W_i\,(0)\,e^{-(\tau+2\beta)\,z} \tag{14.35}$$

the sound power radiated by a length l of duct is given by

$$\begin{aligned} W_{io}\,(z) &= \int_0^l d\,W_i\,(z)\,d\,z \\ &= W_i\,(0)\,\frac{P}{S}\,10^{-0.1\,\Delta\,L_{TLio}} \int_0^l e^{-(\tau+2\beta)\,z}\,d\,z \end{aligned} \tag{14.36}$$

which yields

$$W_{io}\,(z) = W_i\,(0)\,\frac{P\,\ell}{S}\,10^{-0.1\,\Delta\,L_{TLio}}\left[\frac{1-e^{-(\tau+2\beta)\,l}}{(\tau+2\beta)\,l}\right] \tag{14.37}$$

and converting to levels

$$L_{wio} = L_{wi} - \Delta\,L_{TLio} + 10\,\log\,\frac{P\,l}{S} + D \tag{14.38}$$

where L_{wio} = sound power radiated out of the duct (dB)
L_{wi} = sound power level entering the duct (dB)
ΔL_{TLio} = sound transmission loss from the inside to the outside of the duct (dB)
P = perimeter of the duct (m or ft)
l = length of the duct (m or ft)
S = cross sectional area of the duct (m^2 or ft^2)

D, the duct loss term, is defined as

$$D = 10\,\log\left\{\frac{1-e^{-(\tau+2\beta)\,l}}{(\tau+2\beta)\,l}\right\} \tag{14.39}$$

where $\beta = \dfrac{\Delta\,L_{duct}}{8.68}$ (Nepers / ft or Nepers / m)
ΔL_{duct} = attenuation per unit length inside the duct (dB)
$\tau = \dfrac{P}{S}\,10^{-0.1\,\Delta L_{TLio}}$

The D term in Eq. 14.39 can be ignored in short sections of duct, particularly when the duct is unlined and unwrapped. However, for lined duct it should be included. In internally lined ducts the attenuation term is usually larger than the breakout term. In flex or fiberglass ducts the breakout term may dominate, though transmission loss data for these products is difficult to obtain. The breakout sound power can never exceed the internal sound power.

Once the sound has penetrated the duct walls it radiates into the room at high frequencies as a normal line source. Equation 8.85 can be used to predict the expected sound pressure level in the room. Alternatively if the duct is long and unlined it radiates like a line source at high frequencies and the sound pressure level is given by

$$L_p = L_{wio} + 10 \log \left[\frac{Q}{2\pi r l} + \frac{4}{R} \right] + K \tag{14.40}$$

where K is 0.1 for SI units and 10.5 for FP units.

At low frequencies if the duct is oriented perpendicular to two parallel walls it may excite resonant modes in the room, in which case the simple diffuse field condition does not exist (Ver, 1984).

Transmission Loss of Rectangular Ducts

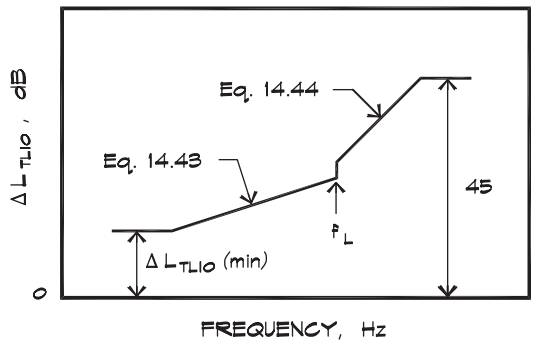
The duct transmission loss for breakout of rectangular ducts is divided into regions by frequency that are similar to those discussed in Chapt. 9 for flat panels (ASHRAE, 1987). The transmission loss behavior with frequency is first stiffness controlled, then mass controlled, and finally coincidence controlled. Figure 14.11 shows the general structure of the loss for rectangular ducts.

For all but very small ducts the fundamental wall resonance falls below the frequency range of interest. In this region there is a minimum transmission loss that is dependent on the duct dimensions (a, b in inches and *l* in feet)

$$\Delta L_{TLio}(\min) = 10 \log \left[24 l \left(\frac{1}{a} + \frac{1}{b} \right) \right] \tag{14.41}$$

At higher frequencies, in the mass controlled region, there is a crossover frequency, below which the transmission loss is affected by the duct dimensions, and above which it follows

FIGURE 14.11 Interior to Exterior Transmission Loss for Rectangular Ducts (ASHRAE, 1987)



normal mass law. The crossover frequency is given by

$$f_L = \frac{24120}{\sqrt{a\,b}} \tag{14.42}$$

where a is the larger and b the smaller duct dimension in inches. Below this frequency the transmission loss is given by

$$\Delta L_{TLio} = 10 \log\left[\frac{f\,m_s^2}{a+b}\right] + 17 \tag{14.43}$$

where f is the frequency, and m_s is the duct wall surface mass in lbs/sq ft.

Above the crossover frequency where the normal mass law holds, the transmission loss for steel ducts is given (as in Eq. 9.21) by

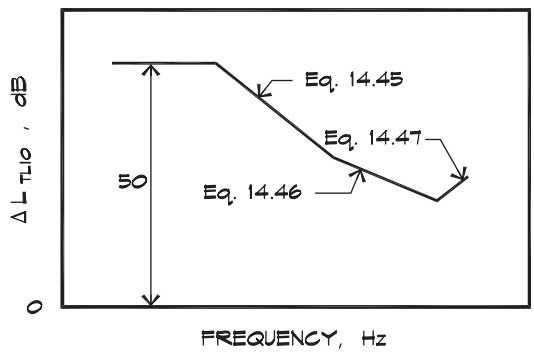
$$\Delta L_{TLio} = 20 \log\left[f\,m_s\right] - K_{TL} \tag{14.44}$$

where $K_{TL} = 47.3$ in SI units and $= 33.5$ in FP units. At very high frequencies side walls exhibit the normal behavior at the coincidence frequency, but for thin sheet metal this lies above 10 kHz and is of little practical interest.

Transmission Loss of Round Ducts

When sound breaks out of round ducts the definitions in Eq. 14.38 are the same as those used in rectangular ducts. The inside area is $S = \frac{\pi\,d^2}{4}$ and the outside transmitting area is $P\,l = 12\,\pi\,d\,l$, where d is in inches and l is in feet. The behavior of round ducts is less well understood and more complicated than with rectangular ducts; however, if the analysis is confined to octave bands, the transmission loss can be approximated by a curve, shown in Fig. 14.12. The low-frequency transmission loss is theoretically quite high because the hoop strength of the ducts in their fundamental breathing mode is very large. In practice we do not achieve the predicted theoretical maximum, which may be as high as 80 dB, and a practical limit of 50 dB is used. As the frequency increases the transmission loss is dependent on the localized bending of the duct walls.

FIGURE 14.12 Interior to Exterior Transmission Loss for Round Ducts (Reynolds, 1990)



Reynolds (1990) has given three formulas to approximate the curve segments shown in Fig. 14.12. The transmission loss is given by the larger of the following two formulas:

$$\Delta L_{TLio} = 17.6 \log (m_s) - 49.8 \log (f) - 55.3 \log (d) + C_o \tag{14.45}$$

$$\Delta L_{TLio} = 17.6 \log (m_s) - 6.6 \log (f) - 36.9 \log (d) + 97.4 \tag{14.46}$$

where ΔL_{TLio} = sound transmission loss from the inside to the outside of the duct (dB)
m_s = mass/unit area (lb / sq ft)
d = inside diameter of the duct (in)
C_o = 230.4 for long - seam ducts or 232.9 for spiral - wound ducts

In the special case where the frequency is 4000 Hz and the duct is greater than or equal to 26 inches in diameter there is a coincidence effect and

$$\Delta L_{TLio} = 17.6 \log (m_s) - 36.9 \log (f) + 90.6 \tag{14.47}$$

Since the maximum allowable level is 50 dB, if the calculated level exceeds this limit the transmission loss is set to 50.

Transmission Loss of Flat Oval Ducts

The transmission loss of flat oval or obround ducts falls in between the behavior of square and rectangular ducts. The lower limit of the transmission loss is given by

$$\Delta L_{TLio} = 10 \log \left[\frac{P\,l}{S} \right] \tag{14.48}$$

since if it were any less, according to the definition of transmission loss in Eq. 14.38, it would imply amplification. For obround ducts the areas are given by

$$S = b\,(a - b) + \frac{\pi b^2}{4} \tag{14.49}$$

and

$$P\,l = 12\,l\,[2\,(a - b) + \pi b] \tag{14.50}$$

At low to mid frequencies the wall strength is close to a rectangular duct because of bending of the flat sides. Assuming the radiation is entirely through the flat sides the transmission loss is given by (Reynolds, 1990)

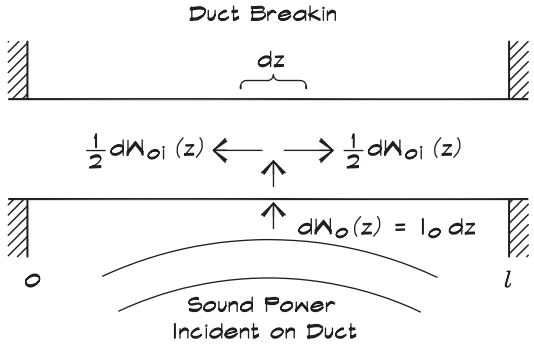
$$\Delta L_{TLio} = 10 \log \left[\frac{f\,m_s^2}{\delta^2\,P} \right] + 20 \tag{14.51}$$

where ΔL_{TLio} = sound transmission loss from the inside to the outside of the duct (dB)
m_s = mass/unit area (lbs / sq ft)
f = octave band center frequency (Hz)
P = perimeter (in) = $2\,(a - b) + \pi b$
δ = fraction of the perimeter taken up by the flat sides

$$\delta = \frac{1}{\left[1 + \dfrac{\pi b}{2\,(a - b)} \right]}$$

Eq. 14.51 holds up to a limiting frequency $f_L = \dfrac{8115}{b}$.

FIGURE 14.13 Duct Break-in Geometry



14.5 BREAK-IN

The phenomenon known as break-in encompasses the transmission of sound energy from the outside of a duct to the inside. The approach is quite similar to that applied to breakout.

Theoretical Approach

The geometry is shown in Fig. 14.13 and the definition of the transmission loss is similar to that given for breakout.

$$\Delta L_{TLoi} = 10 \log \left[\frac{d W_o (z)}{d W_{oi} (z)} \right] \tag{14.52}$$

where $d W_o (z) = I_o P \, d z$ and I_o is the intensity of the diffuse sound field incident on the exterior of the duct. The quantity $d W_{oi} (z)$ is the sound power transmitted from the outside to the inside of the duct by the segment of duct dz, located a distance z from the reference end. The power is given by

$$d W_{oi} (z) = d W_o (z) \, 10^{-0.1 \, \Delta L_{TLoi}} = I_o P \, dz \, 10^{-0.1 \, \Delta L_{TLoi}} \tag{14.53}$$

The sound then travels down the duct toward the reference end and is attenuated as it is carried along. A factor of two is included since the energy is split with half traveling in each direction. Adding the contributions from all the incremental lengths dz from $z = 0$ to $z = l$

$$W_{oi} (z) = \int_0^l \frac{1}{2} d W_{oi} (z) \, e^{-(\tau + 2\beta) z} \, d z \tag{14.54}$$

which is

$$W_{oi} (z) = \frac{I_o P l}{2} 10^{-0.1 \, \Delta L_{TLio}} \left[\frac{1 - e^{-(\tau + 2\beta) l}}{(\tau + 2\beta) l} \right] \tag{14.55}$$

and simplifying, we obtain the sound power level at the reference end in terms of the break-in transmission loss

$$L_{woi} = L_{wo} - \Delta L_{TLoi} - 3 + D \tag{14.56}$$

where L_{woi} = sound power breaking into the duct at a given point (dB)
L_{wo} = sound power level incident on the outside of the duct (dB)
ΔL_{TLoi} = sound transmission loss from the outside to the inside of the duct (dB)
D = duct loss correction term (dB)

Ver (1983) has developed simple relationships between the breakout and break-in transmission loss values based on reciprocity. Above cutoff where higher order modes can propagate,

$$\Delta L_{TLoi} = \Delta L_{TLio} - 3 \quad \text{for } f > f_{co} \tag{14.57}$$

and below cutoff,

$$\Delta L_{TLoi} = \text{the larger of} \begin{cases} \Delta L_{TLio} - 4 + 10 \log \dfrac{a}{b} + 20 \log \dfrac{f}{f_{co}} \\ 10 \log \dfrac{P l}{2 S} \end{cases} \tag{14.58}$$

Note that for round and square ducts, the duct dimensions a and b are equal. The lowest cutoff frequency is given in Eq. 8.21, in terms of the larger duct dimension, a

$$f_{co} = \frac{c_o}{2a} \tag{14.59}$$

The sound power impacting the exterior of the duct will depend on the type of sound field present in the space. Where the reverberant field predominates, the sound power level incident on the exterior is

$$L_{wo} = L_p + 10 \log P l - 14.5 \tag{14.60}$$

where L_p is the sound pressure level measured in the reverberant field and the dimensions are in feet.

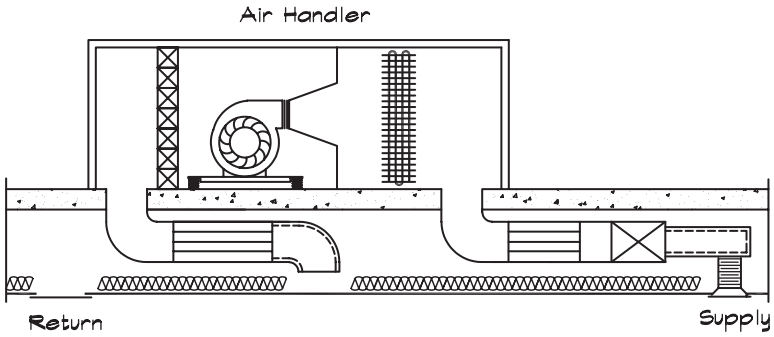
14.6 CONTROL OF DUCT BORNE NOISE

Duct Borne Calculations

A typical duct borne noise transmission problem is illustrated in Fig. 14.14. A fan is located in a mechanical enclosure and transmits noise down a supply duct and into an occupied space. On the return side the ceiling space acts as a plenum for return air, which enters through a lined elbow. There could well be more paths to analyze, such as breakout from the side of a supply or return elbow before the silencer; however, for purposes of this example, we limit it to these two.

The starting point is the sound power level emitted by the fan, which we calculate from the operating point conditions. In this example the fan has a forward curved blade, producing 5000 cfm at 2" of static pressure.

FIGURE 14.14 Roof-mounted Built-up Air Handler



Using the fan equations, we can calculate the sound power in octave bands as shown in Table 14.9. We then follow along each path, subtracting the attenuation due to each element and then adding back the sound power that each generates. The computer program used to generate these numbers uses a 0 dB self-noise sound power level as the default value or when calculated levels are negative. This has a slight effect on the very low levels but is of no practical consequence.

TABLE 14.9 HVAC System Loss Calculations, dB

No.	Description	Octave Band Center Frequency, Hz							
		63	125	250	500	1000	2000	4000	8000
Supply									
1	**Fan, Centrifugal, FC—5000 cfm, 2" s.p.**								
		90	86	82	79	77	75	71	61
2	**Elbow—36" × 24", Unlined**								
		0	−1	−2	−3	−3	−3	−3	−3
	Sum	90	85	80	76	74	72	68	58
	Self Noise—0.05" pd								
		41	39	36	29	20	6	0	0
	Combined	90	85	80	76	74	72	68	58
3	**Silencer, Standard Pressure Drop Type—3' long, 36" × 24"**								
		−7	−12	−16	−28	−35	−35	−28	−17
	Sum	83	73	64	49	39	37	40	41
	Self Noise—0.25" pd								
		49	43	44	42	42	45	35	24
	Combined	83	73	64	49	44	45	41	41

continued

TABLE 14.9 HVAC System Loss Calculations, dB (Continued)

No.	Description	Octave Band Center Frequency, Hz							
		63	125	250	500	1000	2000	4000	8000
4	Duct, Rectangular Sheet Metal—36” × 24”, 5’ long, 1” lining								
		−2	−2	−3	−7	−15	−12	−11	−9
	Sum	81	71	61	42	29	33	30	32
	Self Noise								
		0	0	0	0	0	0	0	0
	Combined	81	71	61	42	29	33	30	32
5	Split, 25%								
		−6	−6	−6	−6	−6	−6	−6	−6
	Sum	75	65	55	36	23	27	24	26
	Self Noise								
		0	0	0	0	0	0	0	0
	Combined	75	65	55	36	23	27	24	26
6	Duct, Rectangular Sheet Metal—18” × 12”, 6’ long, 1” lining								
		−3	−3	−5	−11	−25	−22	−16	−13
	Sum	72	62	50	25	−2	5	8	13
	Self Noise								
		0	0	0	0	0	0	0	0
	Combined	72	62	50	25	2	6	9	13
7	Duct, Round Flex Duct—12” diameter, 6’ long								
		−14	−14	−16	−15	−17	−22	−16	−13
	Sum	58	48	34	10	−15	−16	−7	0
	Self Noise								
		0	0	0	0	0	0	0	0
	Combined	58	48	34	10	0	0	1	3
8	Rectangular Diffuser, 312 cfm—0.05” pd, 6’ to receiver								
		0	0	0	0	0	0	0	0
	Sum	58	48	34	10	0	0	1	3
	Self Noise								
		33	32	29	23	15	4	0	0
	Combined	58	48	35	23	15	5	4	5

continued

TABLE 14.9 HVAC System Loss Calculations, dB (Continued)

No.	Description	Octave Band Center Frequency, Hz							
		63	125	250	500	1000	2000	4000	8000
9	Room Effect—20' × 20' × 8' Room, Drywall Walls, Carpeted Floor								
		−6	−6	−5	−5	−6	−7	−6	−6
	Sum	52	42	30	18	9	−2	−2	−1
Return									
1	Fan, Centrifugal, FC—5000 cfm, 2" s.p.								
		90	86	82	79	77	75	71	61
2	Elbow—36" × 24", Unlined								
		0	−1	−2	−3	−3	−3	−3	−3
	Sum	90	85	80	76	74	72	68	58
	Self Noise 0.05" pd - 4500 cfm								
		43	42	39	33	24	12	0	0
	Combined	90	85	80	76	74	72	68	58
3	Silencer, Low-frequency Standard Pressure Drop Type—5' long, 36" × 24"								
		−16	−21	−35	−41	−41	−28	−21	−15
	Sum	74	64	45	35	33	44	47	43
	Self Noise 0.3" pd - 4500 cfm								
		51	49	53	56	56	59	60	53
	Combined	74	64	54	56	56	59	60	53
4	Elbow—36" × 24", Lined, 1"								
		−1	−2	−3	−4	−5	−6	−8	−10
	Sum	73	62	51	52	51	53	52	43
	Self Noise 0.05" pd - 4500 cfm								
		39	38	34	28	18	4	0	0
	Combined	73	62	51	52	51	53	52	43
5	Plenum, 2" Duct Liner on Gypboard—800 sq ft, 50% Lined, 8 ft @ 85°								
		−12	−13	−19	−20	−20	−20	−21	−21
	Sum	61	49	32	32	31	33	31	22
	Self Noise								
		0	0	0	0	0	0	0	0
	Combined	61	49	32	32	31	33	31	22

continued

TABLE 14.9 HVAC System Loss Calculations, dB (Continued)

No.	Description	Octave Band Center Frequency, Hz							
		63	125	250	500	1000	2000	4000	8000
6	Rectangular Grille—24" × 24", 563 cfm, 0.05" pd, 6' to receiver								
		0	0	0	0	0	0	0	0
	Sum	61	49	32	32	31	33	31	22
	Self Noise								
		30	29	26	20	12	1	0	0
	Combined	61	49	33	33	31	33	31	22
7	Room Effect—20' × 20' × 8' Room, Drywall Walls, Carpeted Floor								
		−9	−8	−6	−8	−8	−8	−9	−10
	Sum	52	41	27	25	23	25	22	12
Combined Supply and Return									
	Supply	52	42	30	18	9	−2	−2	−1
	Return	52	41	27	25	23	25	22	12
	Combined	55	45	32	26	23	25	22	12
	NC 30	57	48	41	35	31	29	28	27

By making the comparison to the room criterion we surmise that the design is satisfactory. We have not checked the breakout level in the plenum through the walls of the first elbow, which should be done. Breakout levels through the walls of a silencer are on the same order as the low-frequency levels passing through the silencer and are not a concern at high frequencies.

Calculations such as these are routine in new construction. They are also useful in trouble shooting existing installations. Low-frequency noise can be generated by duct rumble or by the fans themselves. Mid frequency noise is often due to excessive duct velocities and high frequency noise to diffusers. When the measured levels do not agree with the calculated values, other causes such as flanking paths and duct velocity problems should be examined.

15 DESIGN AND CONSTRUCTION OF MULTIFAMILY DWELLINGS

As the growing population is extruded into urbanized areas, multifamily dwellings must be constructed to accommodate the higher densities. Pressures of population and cost force people together, and noise and noise transmission between occupied spaces become significant concerns. People want their homes to be quiet and free from intrusions, like a single-family residence. The most common complaints are about noise transmission through floor-ceilings, footfall noise, and noise due to the movement of people on floors and stairways. The next most common problem is audibility of plumbing and piping. Airborne noise transmission can also be problematic, but is encountered less frequently.

The design and construction of multifamily dwellings must include consideration of privacy, which in many cases is legally mandated; even if it is not controlled by a building code or property line ordinance, it nevertheless forms part of the basis of the home buyer's or occupant's reasonable expectation of quality. If the dwelling, by dint of its construction, does not meet this expectation there may be sufficient cause for the finding of a construction defect in the building for which the developer and his design team may be held liable. As the perceived quality of a residence increases, so too do the expectations for a quiet environment. This perception of quality may be based on cost, location, sales information provided to the buyer, or due to the fact that a person is purchasing a permanent home rather than renting an apartment.

Since buildings increasingly are constructed from lightweight materials, the sound transmission between spaces increases. In the older masonry and concrete structures, the mass law insured that isolation would be quite high. The exigencies of cost and time have pushed building construction toward lighter and lighter materials, and hence to greater sound transmission. Given these very real constraints it is incumbent upon architects and engineers to find ways of providing adequate sound isolation in residential structures using the commonly available materials. Where dwelling units are separated by design, good results can be achieved without heroic measures. For example, in multifamily dwellings a townhouse plan is preferred over stacked units to avoid common floor-ceilings. When multistory units are necessary, a plan that stacks similar rooms, one above another, avoids incompatible uses such as a bathroom located above a bedroom. Closets and other nonsensitive spaces can be located on party walls to provide additional shielding.

15.1 CODES AND STANDARDS

Sound Transmission Class—STC

In Chapt. 9 we discussed the formal procedures for the measurement of the airborne sound transmission loss and the determination of the sound transmission class (STC) of a partition. The STC is a weighted average of the transmission loss values at 16 third-octave band frequencies, which is normalized using the area of the common partition and the absorption in the receiving room. Many cities and states have adopted minimum code standards for the STC ratings in multifamily dwellings and these can be used to develop prudent design objectives for various levels of construction quality.

The legally mandated minimum STC ratings are usually set to 50 (State of California, 1974; the Uniform Building Code or UBC, Appendix Chapter 35, 1982); however, in some cases stricter standards have been adopted. For example, the City of Redondo Beach, CA requires a minimum STC of 55 in condominium homes. Under field conditions the measured FSTC rating is about five points lower than the laboratory rating, and this difference is acknowledged in the building code. Thus if a field test is performed to check the rating of a separation after the building has been completed, an FSTC 45 is the minimum allowed under UBC requirements.

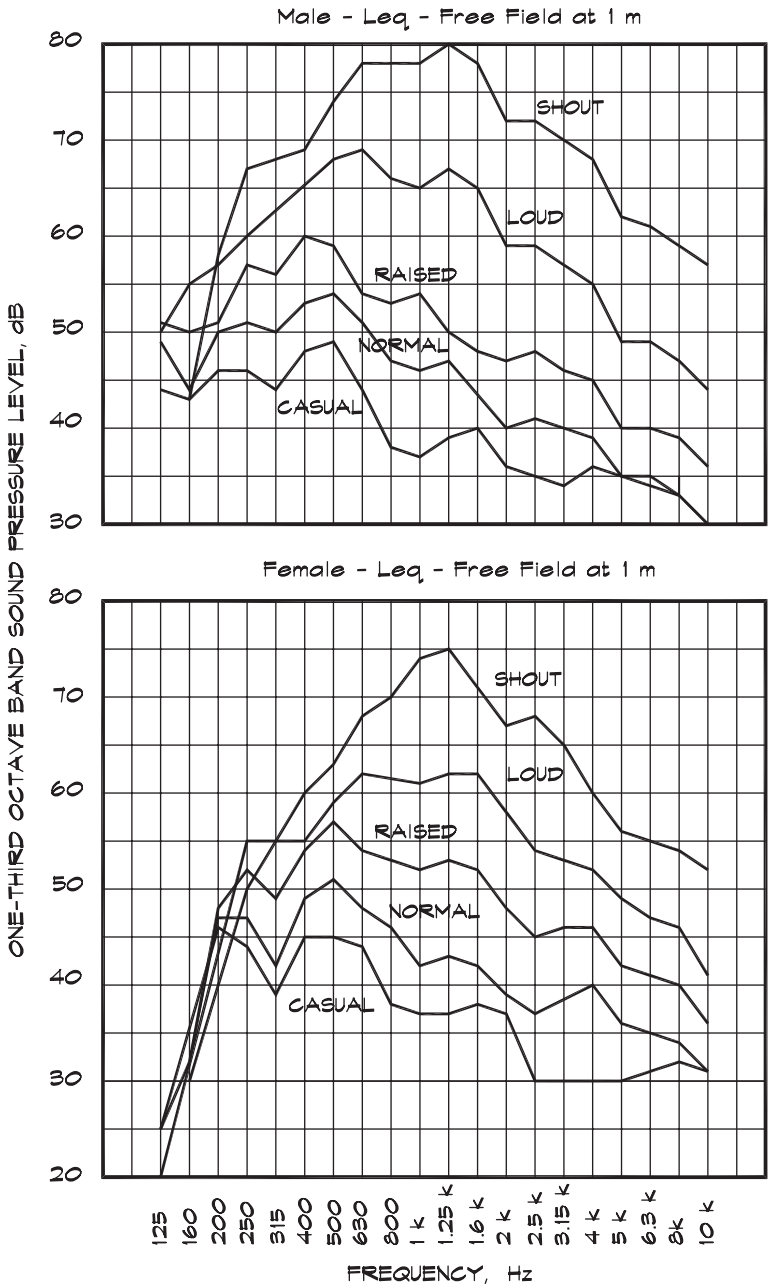
In California (Title 24 CAC, 1990), the NIC, which is an FSTC measurement based on the noise reduction without normalization, is allowed to be measured in lieu of the FSTC. The NIC, defined in Chapt. 9, is generally three to five points higher than the FSTC, and varies from room to room depending on the absorption in the receiving space. Thus this provision not only introduces a substantial weakening of the FSTC 45 minimum code standard, but also presents a standard, which may not be representative of the type of partition being tested.

An STC 50 may be the lowest allowable laboratory rating for a given partition. This does not necessarily represent a level of quality that guarantees owner satisfaction with the dwelling or acoustical privacy between units. Rather, it is the minimum level of quality legally acceptable; it is illegal to build a building any worse. The degree of isolation for airborne noise transmission depends not only on the building construction but also on the type of source, the level of the noise, and on the background noise in the receiving space. Table 15.1 and Fig. 15.1 give the assumptions used in a hypothetical sound transmission calculation. The music spectrum is taken to be flat between 125 Hz and 1000 Hz and rolls off 3 dB per octave above and below these limits. Calculations lead us to the levels shown in Table 15.2. The background level is typical of that found in a quiet bedroom at night.

TABLE 15.1 Source and Background Level Assumptions

Source Level	Receiver Level
Normal Voice = 58 dBA at 3'	Understandable => 30 dBA
Raised Voice = 65 dBA at 3'	Plainly Audible => 25 dBA
Loud Voice = 75 dBA at 3'	Background = 25 dBA
Shouting Voice = 88 dBA at 3'	Audible => 20 dBA
Loud Stereo = 95 dBA at 3'	Not Audible < 20 dBA

FIGURE 15.1 Male and Female Speech Spectra (Pearsons et al., 1977)



The calculations in Tables 15.1 and 15.2 are meant to be illustrative rather than being a result that holds for all sources and wall types. They give a portrait of the transmission of various source levels between spaces and demonstrate that not all sources, even if they are voice, generate the same level, and that minimum code compliance is not necessarily sufficient for adequate acoustical isolation. At the upper end of the level range the numbers show the problems encountered in the design of recording studios where high acoustic levels occur near very quiet recording spaces.

TABLE 15.2 Sound Transmission Class vs Expected Field Result

STC	FSTC	Expected Field Result
80	75	Very loud music audible
75	70	Very loud music plainly audible
70	65	Very loud music understandable
		No unamplified voice audible
65	60	Shouting audible
		Loud voice not audible
60	55	Shouting plainly audible
		Loud voice audible
55	50	Shouting voice understandable
		Loud voice plainly audible
50	45	Loud voice understandable
		Raised voice not audible
45	40	Raised voice plainly audible
		Normal voice not audible

Reasonable Expectation of the Buyer

In selecting the appropriate design criterion for a given level of quality the designer should consider the type of building and the reasonable expectation of quality of the buyer. Unfortunately, too often builders put money into the appearance of a residential building but little into noise isolation. The words *luxury* or *high quality* or *soundproof* are sometimes used to describe projects that barely meet minimum code requirements. If a builder or sales broker is going to characterize the product in this manner, he is well advised to provide a level of noise control commensurate with the description.

Multifamily dwellings can be grouped into three quality categories as shown in Table 15.3. Table 15.4 shows general guidelines according to the level of quality, which assumes a minimum code standard of STC 50.

TABLE 15.3 Level of Quality vs Type of Use

Classification	Residential Use
Minimum Quality	Normal Apartments
	Hotels and Motels
	Nursing Homes
	Hospitals
Medium Quality	Good Apartments
	Normal Condominiums
High Quality	High Quality Condominiums

TABLE 15.4 Sound Transmission Class vs Level of Quality for Party Wall and Floor-Ceiling Construction

Classification	STC	FSTC
Minimum Code	**50**	**45**
Minimum Quality	**55**	**50**
Medium Quality	**60**	**55**
High Quality	**65**	**60**

To achieve a minimum code standard of STC 50, one should design to the code plus a reasonable safety factor. Generally low-cost rental property, subsistence housing, and temporary housing such as hotels and motels would be designed to the minimum-quality level. Note that the minimum-quality design level is not the same as minimum-code level, since there must be a certain safety factor included to assure code compliance. If one were to design exactly to the code minimum it would mean that the selected construction would have a 50% probability of passing a field test. This is not considered good design practice, and a 3–5 dB minimum margin of safety is recommended. In practice, published test results for a given wall or floor will vary by a few points. It is prudent to examine the range of test results for a given configuration and to expect the lowest values in the test range rather than the highest.

The medium-quality level is appropriate for use for high-quality apartments and normal condominiums. In general, any condominium should be designed to at least the medium quality standard. If noise problems arise, the owner of a condominium does not have the freedom of movement of an apartment dweller. Under California law the seller must reveal any known defects to a potential buyer, including any problems associated with noise transmission. A first-time condominium purchaser may be moving from a single family home and have an expectation of quality based on his previous housing experience.

Into the high-quality category fall those condominiums where there is a level of isolation similar to that found in a single family home. In these cases owners may complain if they can hear any activities in adjacent dwelling units. They are particularly sensitive to footfall and plumbing noise since these may occur relatively frequently. Even for this type of structure the ratings given in Table 15.4 will not guarantee isolation of every noise, as illustrated in Table 15.2.

Impact Insulation Class—IIC

Minimum IIC ratings are set to 50 in the UBC with a minimum field tested FIIC of 45 allowed. At this rating, footfall noise is quite pronounced and very audible in the unit below. In response some cities and condominium associations have adopted more stringent laws. The City of Redondo Beach, for example, sets a minimum IIC rating of 65 in condominiums. Other cities such as Beverly Hills control noise through a property line ordinance as discussed next. The point at which footfall-generated impact noise becomes inaudible is closer to an IIC of 75, as shown in Fig. 12.23. The level of quality due a buyer in the control of impact-generated noise is numerically higher than that for airborne noise.

TABLE 15.5 Impact Insulation Class vs Level of Construction for Party Floor-Ceiling Construction

Classification	IIC	FIIC
Minimum Code	**50**	**45**
Minimum Quality	**55**	**50**
Medium Quality	**65**	**60**
High Quality	**75**	**70**

Property Line Ordinances

Cities and counties have ordinances restricting the levels of noise that are allowed within a real property boundary. These are known as property line ordinances and are tied to the zoning of the receiving property. Allowable levels are based on a measured or assumed ambient level within the receiving property, since a noise maker cannot be expected to be responsible for noise generated by other sources. A law usually establishes an absolute maximum at a level 5 dBA above the higher of the measured or assumed ambient. Normally the assumed ambient is reduced at night (before 7 AM and after 10 PM) by 10 dB to account for our increased sensitivity. Assumed ambient background levels provide the basis for the lowest level to which a standard may fall. For example in the City of Los Angeles, the nighttime assumed ambient is 40 dBA, so a noise maker would be allowed to create 45 dBA, even if the actual ambient background were below 40 dBA. This approach gives the noise maker a clear numeric design target even if the actual ambient falls below the assumed ambient.

The wording of these ordinances varies. Some are stated in terms of an absolute level such as the US EPA Model Noise Ordinance (1973), which reads:

> "No person shall operate or cause to be operated on private property any source of sound in such a manner as to create a sound level, which exceeds the limits set forth for the receiving land use category in Table __ when measured at or within the property boundary of the receiving land use."

In this case the ambient must still be taken into consideration since the ordinance applies only to the intrusive source.

The specific wording of an ordinance, particularly the definition section, is important. In many cases they set maximum limits on interior noise levels, which apply within the property boundaries of a dwelling unit. Thus if a person in one unit walks across a floor or turns on a tub faucet and the occupant of another unit is subjected to a noise in excess of the property line ordinance, there may be a cause of action against the builder who caused the condition to exist. It is not an action that is taken against the occupant unless the noise-making activity is unusual or excessive, such as playing a stereo too loudly, or unless the owner has changed the construction so as to worsen its sound reduction capability, for example by replacing a carpeted floor with a wood or tile floor. If an occupant runs the bath or shower, this would not be considered an unusual activity. If, however, he is doing midnight body slams, the resulting noise could not necessarily be blamed entirely on the developer.

TABLE 15.6 Recommended Maximum Interior Day Night Noise Levels from Exterior Sources

Classification	L_{dn} (dBA)
Minimum Code	45
Minimum Quality	40
Medium Quality	35
High Quality	30

Under California law developers are liable for the cost of testing and repair if party wall or floor-ceiling separations do not meet the minimum codes. Architects and engineers who design multifamily dwellings are well advised to consult local property line ordinances to make sure they are in compliance. The City of Beverly Hills, for example, limits interior noise levels to no more than 5 dB over the actual interior ambient. This is not an unusual provision; however, Beverly Hills defines the ambient as the quietest level present at any time of day at a given location with no minimum. An ordinance of this type can put a greater restriction on intrusive noise levels than a minimum STC or IIC rating does.

Exterior to Interior Noise Standards

The State of California (CAC Title 25, 1974) and other regulating bodies set maximum allowable interior levels generated by exterior noise sources such as street traffic. Housing and Urban Development (HUD) financed housing projects are subject to the same interior requirements (24 CFR 51B, 1979) as well as additional exterior requirements. Under both California law and the HUD regulations the limit is an interior L_{dn} of 45 dBA. In much the same way as the limits on STC and IIC were minimum code levels, so too are the allowable interior noise levels. Occupants are seldom happy with the minimum code noise level. Table 15.6 lists recommended interior levels for different types of construction quality. The interior standards assume that all windows and doors are closed so that adequate mechanical ventilation must be provided.

15.2 PARTY WALL CONSTRUCTION

General Principles

In actual building practice there are relatively few construction materials that are utilized, and a knowledge of transmission loss theory is most helpful in properly applying them. The most common materials are concrete, concrete masonry units (cmu), stucco, gypsum plaster, gypboard, and wood or metal sheets in various combinations. The structural supports are wood or metal studs for walls, and concrete, steel, or wood-joist systems for floors.

At low frequencies providing adequate stiffness and mass are the most important factors in achieving high transmission loss values. Stiffness can be increased by decreasing the support span and by increasing the bending stiffness. Short-span concrete slabs have both high mass and a large intrinsic stiffness and thus give excellent low-frequency transmission loss. Fully grouted cmu blocks or brick can provide nearly as much mass and stiffness; however, concrete blocks have significant porosity and must be sealed with an oil-based paint or plaster to realize the full effectiveness of their mass. Double concrete or cmu walls

can be used, but the spacing between panels must be sufficiently large that the two are not coupled through the air gap. Usually at least 10 to 15 cm (4" – 6") of spacing is required. Details of the STC ratings were discussed in Chapt. 10.

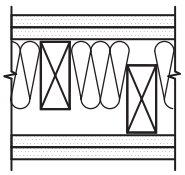
At higher frequencies, separately supported gypboard partitions, which have a high critical frequency and a large air space, are a good choice. If the separation distance is large enough, these can be more effective than a single concrete panel. In separately supported structures, either wood or metal studs yield the same results. Offsetting or staggering studs, which are already on separate base plates, is not necessary.

In single-stud construction there is a more limited range of options available. With wood studs the two panels are rigidly attached by means of the line connections. The addition of multiple layers of drywall is only somewhat effective. A resilient attachment provides some decoupling, though not as much as a separated stud. Metal studs, because they are inherently flexible, can also provide significant decoupling. Resilient supports can be helpful in decoupling gypboard layers on either side of a wood stud or floor joist. Resilient channel must be properly installed so that the screws do not short circuit. Channels applied directly over layers of gypboard or other panel materials are ineffective due to bridging by the trapped air pocket. Products that can be attached only on one side are preferred over hat-shaped channels, which can be attached on both sides. Resilient channels are not recommended for party walls since they are not suitable for the mounting of bookshelves or heavy pictures. When applied to double stud and lightweight metal stud walls, resilient supports do not significantly increase the sound transmission loss since the structures are already isolated.

Party Walls

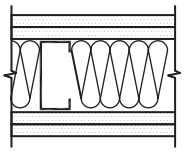
The selection of a party-wall construction should be based on the level of quality and the ultimate use of a given development. An example of party wall construction for minimum-quality construction is shown in Fig. 15.2. The minimum-quality construction consists of two layers of 5/8" (16 mm) drywall on each side of staggered 2 × 4 (38 × 89 mm) studs set

FIGURE 15.2 Minimum-Quality Party Walls



Staggered Wood Stud Wall
2 x 4 (50 mm x 100 mm) Studs on a 2 x 6
(50 mm x 150 mm) Plate
2 Layers 5/8" (19 mm) Drywall Each Side
R-11 Batt Insulation

125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
35	39	45	45	49	49	52	53	54	55	56	57	54	50	53	57	53



Metal Stud Wall
3 5/8" (90 mm) Metal Stud
2 Layers 1/2" (13 mm) Drywall Each Side
R-11 Batt Insulation

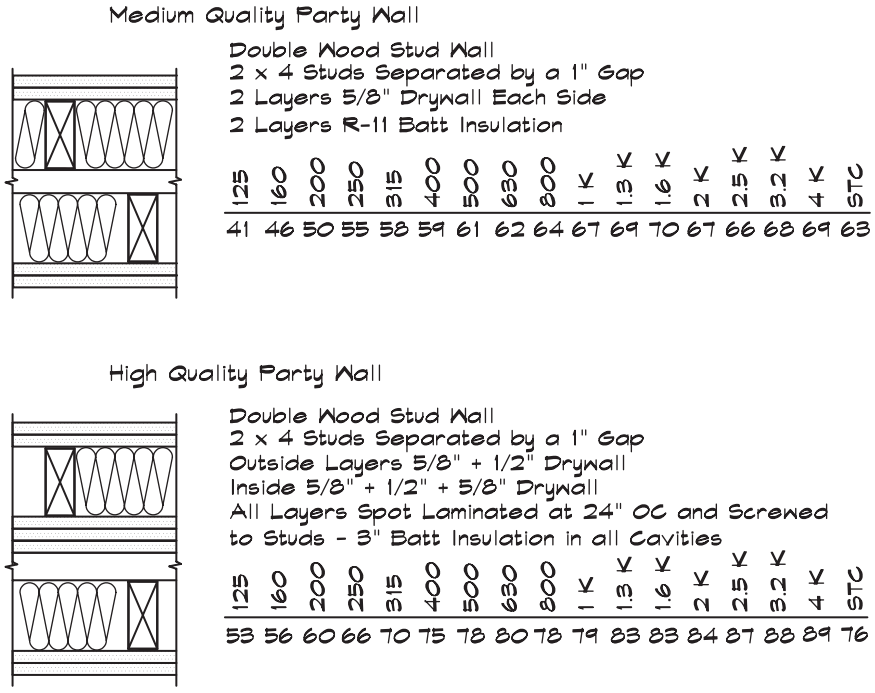
125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	4 K	STC
40	44	46	48	52	56	59	58	59	59	62	63	62	58	52	57	56

on a 2 × 6 (38 × 140 mm) wood plate. An alternative to staggered studs would be the use of a light-gauge 3 5/8" (92 mm) metal stud. The wall should have 3 1/2" (90 mm) fiberglass batt insulation having a thermal rating of R-11 in the air space. The staggered-stud wall rates an STC 53 while the metal-stud wall rates an STC 56 (CONC, 1981) with double 1/2" (13 mm) drywall. Generally a wall with two layers of drywall each side is preferable to one with only one layer even when the stud configuration yields comparable STC ratings. The small advantage in using mismatched drywall thicknesses in low-rated party walls (< 55 STC) is probably not worth the confusion it produces in having two thicknesses of drywall on the job site.

For medium-quality construction the first wall shown in Fig. 15.3 is a good choice. It consists of two layers of 5/8" (16 mm) drywall on each side of separate 2 × 4 (38 mm × 89 mm) wood or 3 5/8" (92 mm) metal studs separated by at least 1" (25 mm). There are two layers of R-11 fiberglass batt in the airspace. This wall has achieved an STC 63 in a laboratory test (CONC, 1981).

In high-quality construction projects the triple-panel wall shown in Fig. 15.3 has been used successfully. It consists of two layers of drywall, one 1/2" (13 mm) and the other 5/8" (16 mm) thick in the outside of separate 2 × 4 (38 mm × 89 mm) studs. On the inside of one set of studs are three layers of drywall: two 5/8" (16 mm) and one 1/2" (13 mm) thick. The layers are spot laminated and screwed together as described in Chapt. 9. There is R-11 fiberglass insulation in the air cavities. The STC test data shown are for a similar wall cited in Fig. 10.11, which was tested by Sharp (1973). The wall shown here will test a few points lower since the panels are not point mounted.

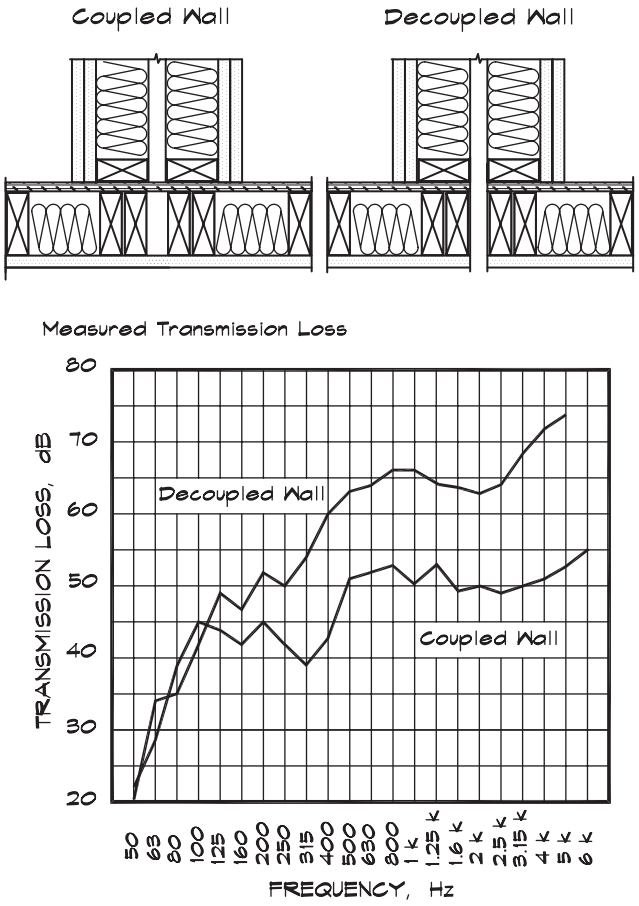
FIGURE 15.3 Medium- and High-Quality Party Walls



Structural Floor Connections

In certain cases there can be significant flanking of a wall through structural connections such as a floor diaphragm. In wood-frame buildings there are requirements that there be a fire stop between wall studs or joists supporting a floor. In earthquake country there are also requirements that the building have adequate stability to withstand vibration-induced lateral motion. In many cases this stability is provided through a plywood diaphragm, which runs continuously from floor to floor beneath a sound rated partition such as a party wall in a multifamily residence. Craik, Nightingale, and Steel (1997) published a study of the flanking due to the presence of several types of fire stops: wood, metal, drywall, and safing (no connection). Their calculations (Craik, 1996), which used statistical energy analysis, assumed that the floor and wall could be modeled using four plates, one for each side of the wall and one for the floor on each side, with only a moment connection between them. Both calculated and measured results were reported. The measured results are summarized in Fig. 15.4. Note that the wall has a double stud with double 1/2" (13 mm) gypsum board, a 1" (25 mm) air gap, and two layers of batt insulation. The floor was constructed of a single layer of 5/8" (16 mm) plywood.

FIGURE 15.4 Effects of Structural Decoupling (Craik, Nightingale, and Steel, 1997)



The results show that there can be significant flanking due to the structural path through the floor when there is continuous plywood. In fact, the calculations indicated that for coupled structures the most important noise path is from the source room into the floor, through the diaphragm, into the adjacent floor, and into the receiving room. Improvements can be obtained by separating the two sides and by using a thin sheet metal or safing fire stop and by increasing the mass of the floor structure through the use of a concrete topping layer or by the installation of a floating floor system. Horizontal fire stops in double stud party walls can be achieved with drywall, which is attached only to one side. On the opposite side the gap between the drywall and the stud is minimized and stuffed with safing.

In concrete-slab construction the high mass of the floor helps block the room-floor-floor-room path. In wood construction a continuous diaphragm may be required for structural reasons, but here concrete topping slabs increase the floor mass and help decrease the floor transmission path. Metal straps may provide the coupling required by structural or earthquake requirements, while still providing a significant impedance mismatch.

Flanking Paths

In party-wall construction there can also be nonstructural flanking paths. When double drywall is used as a surface material, the joints on the second layer should be staggered with respect to the first layer. At the corners, layers should be overlapped and all joints must be premudded before taping. When this is not done a gap can remain between the two layers of drywall, which is covered over only with drywall tape.

At the base plate, the gap between the drywall and the floor should be caulked with a nonhardening caulk. Base plates should not be completely sawed through to accommodate piping. This is important because the drywall needs a continuous backing to seal against. It is more important to caulk under the drywall than it is to caulk under the bottom plate. The principal reason for caulk under the bottom plate is to provide blockage when the lumber is warped; however, this path should also be blocked by caulking the drywall. The addition of a caulked wood base strip along the bottom of the drywall helps to close off the gaps under the drywall.

Blocking headers, located above the top plates of the wall framing, should be doubled and joints between the headers and the joists should be kept tight. This prevents sound, which makes its way into the ceiling cavity, from migrating into an adjacent unit by way of the joist space.

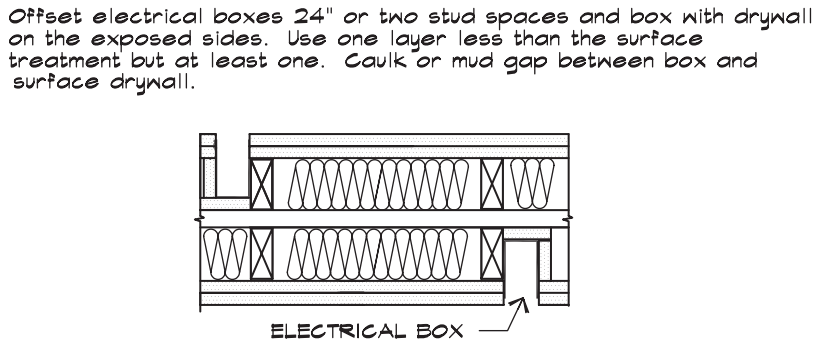
Where bathtubs and showers are located on a party wall they must be installed so that the integrity of the wall is not compromised. This means that the drywall (or green board) must be continuous behind tubs, showers, and other wall-mounted fixtures such as lavatories. Party walls should not be cut out to accommodate medicine cabinets or other surface-mounted millwork.

Drywall must also be continuous behind stairwells. Party wall framing should not provide structural support for stair risers. Even with double stud walls and carpeted stair treads, footfall noise on stairways can be audible when the stair framing is structurally attached.

Electrical Boxes

Once a decision has been made on the construction of the separating partition, care must be exercised to insure that the rating of the partition is maintained. One example, which we have already discussed, is through a hole or other area of reduced transmission loss in a wall.

FIGURE 15.5 Treatment of Electrical Boxes in Rated Walls



These can be gaps in and around electrical boxes or simple cutouts in walls for telephone, computer or television cabling. Too often wiring is run freely in walls and not contained in conduit and metal boxes. When this occurs the plastic electrical wall plate becomes the weak part of the structure and can lead to a degradation of the performance of the partition. Even a small opening such as a 1/2" (13 mm) emt conduit between two 2 × 4 electrical boxes can significantly degrade the transmission loss.

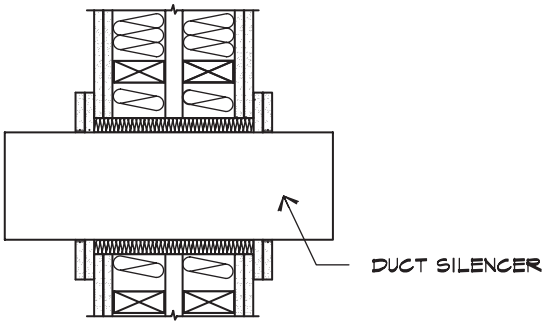
It is good practice to enclose all wiring, including low voltage computer, telephone, and cable TV wiring, that is located in sound-rated partitions, in metal boxes and conduit. Where these boxes penetrate the wall surface the openings between the drywall and the box should be sealed with caulk or plaster. Electrical boxes should be offset 24" (0.6 m) or two studs when they are located on opposite sides of a wall. Figure 15.5 illustrates this principle. The backs and sides of the boxes should be buttered with plaster, wrapped with drywall, or sealed with clay pads to attenuate sound penetration out of the back of the box. Wrapping with drywall is preferred since clay pads can peel away from the box over time. A 1/2" (13 mm) sheet of drywall, which spans the stud bay containing the electrical box from the base plate to a height 12" (300 mm) above the electrical box, can be used in place of wrapping. Batt insulation must be placed within and behind the drywall cavity. At the point where the electrical boxes penetrate the drywall, all gaps between the outside of the box and the drywall must be sealed with drywall mud or caulk.

Wall Penetrations

Where plumbing pipes are located in party walls (this is not recommended), penetrations should be avoided. If a pipe must penetrate a party wall a resilient escutcheon or caulked opening should be used; however, these openings can compromise the sound isolation over time. When water flows through a pipe it can move the pipe around, due to the forces produced by the fluid as well as thermal expansion and contraction. Pipe movement tends to open up the hole at the penetration even when it is caulked. Piping penetrations on opposite sides of a party wall should be offset by 24" (610 mm). Piping to adjacent units should be separate and the piping should only be supported on the studs on the side of the wall whose unit it serves. Plumbing piping or rigid conduit connected to the structure on both sides will short circuit the stud separation. Sufficient space must be allowed for the passage of waste piping so that it does not make contact with the panels or support structure on either side. When there are back to back pipe penetrations in a double stud party wall, a layer of drywall should

FIGURE 15.6 Treatment of Rated Wall Penetrations

When a large duct or pipe penetrates a rated wall, the opening is oversized by about 1" (25mm) and filled with fiberglass board. The ends are capped with drywall closely fitted to the penetrating body and caulked. Use the same number of layers as are on the wall.



be installed on the inside face of a stud behind one penetration, a stud space wide extending 18" (457 mm) above and below the penetration.

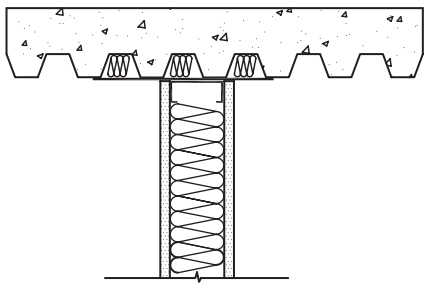
When a duct or pipe penetrates a rated separation the penetration must be treated so as to retain the rating of the partition. If the penetrating element is a large pipe or duct, as shown in Fig. 15.6, a hole in the wall is cut, leaving about a 1" (25 mm) gap that is filled with fiberglass board. The opening is then sealed off with the same number of layers of drywall as the original wall surface and the remaining gap is caulked. In the case of a pipe penetrating a concrete floor, the opening above and below the fiberglass can be filled with a sealant.

Holes

At the top of a wall the attachment of a stud wall to a metal deck can be tricky. Sheet metal plates or rubber filler strips can be used to close off the openings above the top track as shown in Fig. 15.7. Where nested tracks are employed to allow for floor movement, the outer layers of drywall should overlap the inner track and be caulked against the floor plate. When a wall

FIGURE 15.7 Wall Connection at a Metal Deck

A wall butting up against a metal deck should be attached to a sheet metal plate, typically 16 ga. The webs are stuffed with safing or prefabricated neoprene strips.



parallels the deck webs, a plate of 16 Ga sheet metal with safing in the cavities will close off the path above the wall. A wall that runs perpendicular to the webbing can be topped with a wide sheet metal plate, neoprene inserts, or cut drywall. Walls that lie at an oblique angle to the webbing can also be sealed with a wide plate with the cavities stuffed with safing.

15.3 PARTY FLOOR-CEILING SEPARATIONS

Noise and vibration problems encountered in floor-ceiling systems fall into the four categories discussed in Chapt. 12: airborne noise, structural deflection, footfall, and floor squeak. Floor vibration and vibration isolation of mechanical equipment are separate topics, which were discussed in Chapt. 11.

Airborne Noise Isolation

Airborne noise isolation in floors follows the same principles and is tested in the same manner as airborne isolation in walls. As was the case with wall transmission, the isolation of airborne noise such as speech is well characterized by the STC rating. STC tests are done by placing the noise source in the downstairs room to insure vibrational decoupling between the loudspeakers and the floor-ceiling being tested. The ratings shown in Table 15.4 apply to floor-ceiling separations just as they applied to walls, but the choice of floor systems is greater.

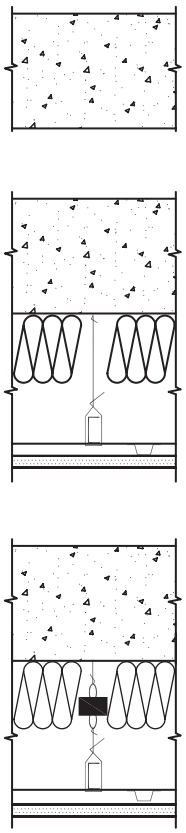
Highly rated floor-ceilings combine a high-mass floor with a large separation between panels as in a double panel wall system. Ideally the two panels should be structurally decoupled either by separate structural supports or by means of a resiliently hung ceiling or floating floor. At low frequencies a high structural stiffness is desirable to minimize the plate deflection.

A simple concrete slab of sufficient thickness can provide a good floor-ceiling. A 6” (152 mm) thick slab has an STC rating of 55 and is sufficient by itself for a minimum quality floor. Six-inch concrete slabs with a wire-hung drywall ceiling can provide sufficient isolation for airborne noise to be used in medium-quality construction. For high-quality construction even with concrete slabs a drywall ceiling suspended from neoprene isolators is preferred. Figure 15.8 shows some examples of concrete floor-ceiling systems.

In wood construction the structures are light and stiff. The problem with wood floors for airborne noise isolation is in achieving sufficient mass. Lightweight-concrete fill weighs 110 to 115 lbs/cu ft (540–560 kg/sq m) and should be poured to a thickness of at least 1.5” (38 mm). A hard concrete fill (140–150 lbs/cu ft or 685–735 kg/sq m) is preferred; however, the structural system must be designed to accommodate the additional weight. Figure 15.9 gives examples of wood floor-ceiling systems suitable for various levels of quality in multifamily dwellings. Note the increasing thickness of plywood subflooring.

Composite floor-ceiling systems fall somewhere between wood and concrete. A composite floor can be constructed using a 5” (127 mm) lightweight concrete fill poured into a webbed sheet metal deck with a suspended ceiling below. With this configuration a drywall ceiling is required even for the minimum design standard. Several designs are shown in Fig. 15.10. Note that there is no structural support provided by the lightweight concrete or the sheet metal pan. The structural floor stiffness is due to the supporting beams. When these beams have a long span, floor deflection can be a greater problem than noise transmission.

FIGURE 15.8 Concrete Floor-Ceiling Assemblies

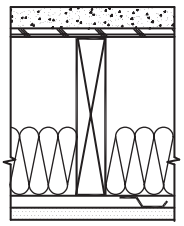


Structural Stiffness

The achievement of a high IIC rating in a given floor-ceiling system does not guarantee that noise will not be a problem or that the sound of walking will not be audible in the units below. The IIC test measures the reaction of a floor system to the impact of a series of 1.1 lb (0.5 kg) weights dropped on the surface. Although this may model the noise of a heel tap, it does not represent the full effect of the loading and unloading under the weight of a walker. When a person steps or even stands on a floor, it will deflect under the static and dynamic load of his weight, as we discussed in Chapt. 12. If the underside of the floor is exposed to the room below, a sound generated by this motion will radiate directly into the receiving space. Noise generated by floor deflection sounds like low-frequency thumps, whereas heel clicks have a spectral character largely dominated by the high frequencies.

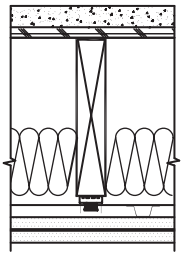
Three mechanisms are available to improve this condition: 1) increase the stiffness of the floor system, 2) increase the structural damping, and 3) increase the vibrational decoupling between the floor and the ceiling. In concrete structures both the stiffness and damping increase with slab thickness. For the 6” (152 mm) concrete slab required to achieve an STC of 53-55 structural deflection is rarely a problem for moderate spans. In wood structures the most common type of minimum quality construction consists of 1.5” (38 mm) lightweight concrete on plywood on joists with ceilings of drywall on resilient channel. This construction

FIGURE 15.9 Wood Floor-Ceiling Constructions



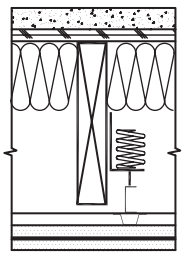
Minimum Quality

1 1/2" (38 mm) Lightweight Concrete
(110 lbs/cu ft or 1760 kg/cu m)
1 Layer 5/8" (16 mm) Plywood Glued and Screw Nailed
2 (50 mm) x Wood Joists
R-11 (3.5" or 90 mm) Batt Insulation
Resilient Channel
1 Layer 1/2" (13 mm) Gypsum Board



Medium Quality

1 1/2" (38 mm) Lightweight Concrete
(110 lbs/cu ft or 1760 kg/cu m)
1 Layer 3/4" (19 mm) Plywood Glued and Screw Nailed
2 (50 mm) x Wood Joists
R-11 (3.5" or 90 mm) Batt Insulation
RSIC-1 Isolators with 7/8" Hat Channel
2 Layers 5/8" (16 mm) Gypsum Board



High Quality

1 1/2" (38 mm) Lightweight Concrete
(110 lbs/cu ft or 1760 kg/cu m)
1 Layer 3/4" (19 mm) Plywood Glued and Screw Nailed
2 (50 mm) x Wood Joists
R-11 (3.5" or 90 mm) Batt Insulation
1" (25 mm) Deflection Spring Isolators
2 Layers 5/8" (16 mm) Gypsum Board

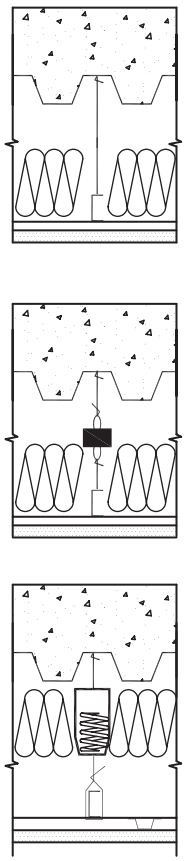
can transmit considerable low-frequency noise, since for normal joist lengths the deflection of the resilient channel is not sufficient to overcome the deflection of the joists.

In wood construction both stiffness and damping can be increased by using the stepped blocking shown in Fig. 12.24. The blocking works for several reasons. The first is the damping added by the moment connection provided by the glued faces and end nailing. Second the stiffness is increased by building the equivalent of another beam in the middle of the joist system. The third effect is additional load spreading, which distributes a point load over several joists and helps increase the composite floor stiffness. Stepped blocking is more effective than doubling joists or reducing joist spacing, although the two can be combined to good effect. When prefabricated truss joists are used, a spacer plate must be installed as in Fig. 12.28. Stepped blocking should be located at the mid-span in joists having a length of between 12 to 18 feet (3.7—5.5 m) and at the one-third points in joists greater than 18 feet.

Structural Decoupling

If a floor-ceiling system is not a monolithic slab, it generally includes an independently supported ceiling, which may be isolated vibrationally from the structure. In concrete

FIGURE 15.10 Steel Deck and Concrete Floor-Ceiling Constructions



Minimum Quality
5" (125 mm) Thick Lightweight Concrete on Metal Decking
(42 lbs/sq ft or 200 kg/sq m)
16" (400 mm) Airspace
R-11 Batt (3.5" or 90 mm) Insulation
5/8" (16 mm) Gypsum Board on Hat Channel
and Carrying Channel

Medium Quality
5" (125 mm) Thick Lightweight Concrete on Metal Decking
(42 lbs/sq ft or 200 kg/sq m)
16" (400 mm) Airspace
R-11 Batt (3.5" or 90 mm) Insulation
0.15" Deflection Neoprene Isolators
5/8" (16 mm) Gypsum Board on Hat Channel
and Carrying Channel

High Quality
5" (125 mm) Thick Lightweight Concrete on Metal Decking
(42 lbs/sq ft or 200 kg/sq m)
16" (400 mm) Airspace
R-11 Batt (3.5" or 90 mm) Insulation
1" (25 mm) Deflection Spring Isolators
5/8" (16 mm) Gypsum Board on Hat Channel
and Carrying Channel

construction the most common support system is hanger wires at 4' (1.2 m) on center wrapped around 1 1/2" (38 mm) carrying channel (black iron) to which 7/8" (22 mm) metal furring channels (hat channels) are wire-tied. This system provides some isolation because it uses a point connection (Chapt. 9) rather than a line connection. It can be improved further by utilizing vibration isolators either in the form of neoprene hangers or steel spring isolators cut into the hanger wires.

In wood structures a common type of structural decoupling is resilient channel. At high frequencies resilient channel can provide some improvement to the structural isolation; at very low frequencies, however, it is not particularly effective. Several different kinds of resilient channel are available on the market, some of which are shown in Fig. 15.11.

When resilient channel is installed improperly, it is ineffective, so the manufacturer's installation instructions must be followed closely. A z-shaped channel is one of the easiest to install, but screws that are too long can still short out the decoupling as shown in Fig. 15.12. Z-shaped resilient channels should be installed with the open side up when they are attached to studs so that the weight of the applied drywall pulls the channel open and away from the stud.

Hat-shaped resilient channels must be installed so that there is an attachment screw on only one side of the flange. Each screw attachment alternates from one side to the next.

FIGURE 15.11 Types of Resilient Channel

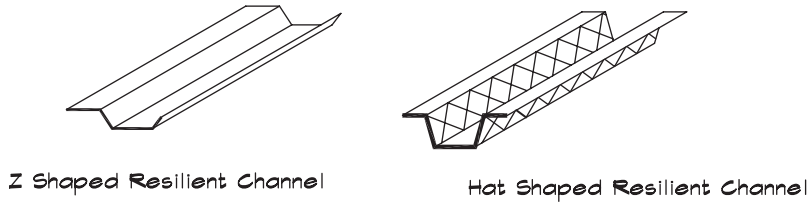
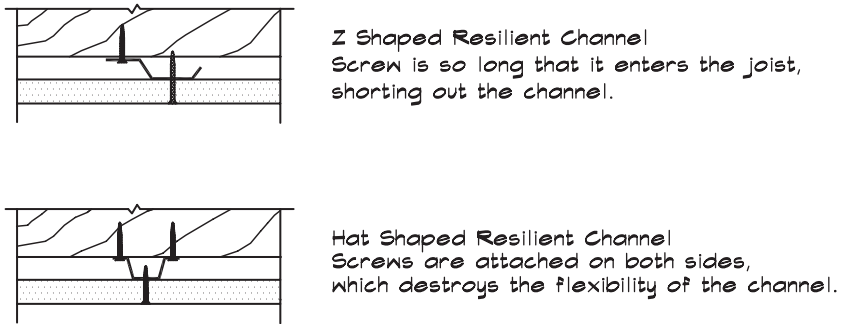


FIGURE 15.12 Improperly Attached Resilient Channel

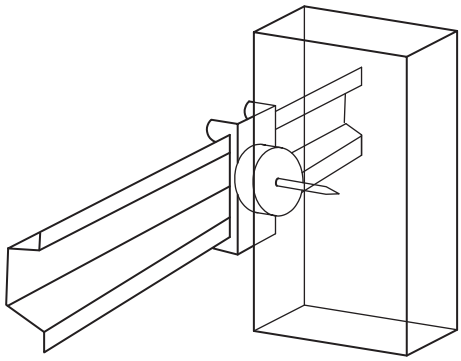


If it is screwed to a joist on both sides the hat channel is not free to deflect and is ineffective. The side webbing of hat-shaped resilient channels have openings to reduce their stiffness; however, they are still too stiff to provide appreciable isolation if they are screwed to the joists on both sides.

Neoprene mounts, which include a clip to support hat channels, recently have become available. These give somewhat better floor isolation than resilient channels (STC 61) and can support a double layer of drywall. They are installed on 24" (0.6 m) centers in one direction and at twice the joist spacing (typically 32" or 0.8 m) in the other. They provide the advantages of a resilient point-mount support along with ease of installation. A sketch is included in Fig. 15.13.

The most effective structural decoupling in wood floor-ceiling systems is a resiliently supported ceiling hung from spring hangers shown in Fig. 12.3 (STC 73). Note that the hangers are located high on the joist to preserve as much ceiling height as possible. Spring hangers are more effective than a ceiling supported on separate joists since with the latter system there is still the possibility of structural transmission through the joist supports. When a spring-hung ceiling is installed, unless the springs are precompressed, it will drop by the amount of the hanger isolator deflection. Hence the ceiling drywall must not extend beyond the top of the wall drywall, or else its weight will be supported by the walls and the ceiling will bow. An example of a proper installation is given in Fig. 12.4. Once the ceiling has come to its final elevation the gap between the ceiling and wall material may be caulked. Molding or other trim pieces then can be added since they are nonbearing.

FIGURE 15.13 RSIC Resilient Point Mount Supports (Pac International)



Spring precompression can minimize the actual deflection; in practice, however, this is somewhat tricky since the final load must be determined carefully. Springs are located at 4' (1.2 m) on center, and if they support 16 sq. ft. (1.5 sq m) of ceiling, at 5.5 lbs/sq ft (27 kg/sq m), they will each carry about 90 lbs (41 kg). A spring located along an edge will carry a little more than half that load, and one in a corner somewhat more than one quarter. In irregularly shaped ceilings or one with coffers and light fixtures the loading is more complex. It is prudent to have springs of several different sizes at a job site in case the odd hanger is needed. When stepped blocking and a resiliently hung ceiling are used in combination, the black iron can run parallel to the joists just below the blocking. The hat channels run perpendicular to the joists just below them. When the drywall is installed its weight will pull the hat channel away from the joists so it does not touch.

Floors should be structurally decoupled laterally as well as vertically. Joists should not be run continuously across a party wall separation but should be supported on the nearest side of the party wall framing.

Floor Squeak

Creaking floors are caused by the relative motion of wood on wood or nails rubbing against diaphragms, joists, or metal joist hangers. One common cause are shiners, as they are called—nonbedded nails that lay alongside a joist and rub as the floor structure deflects. These must be removed before any lightweight or other concrete fill is poured.

Another cause is unevenness in the top surface of the joists, either due to imperfections in the wood or in the case of joist hangers, to differences in the joist level, which allows motion of the floor diaphragm against the nails. Gluing diaphragms to the joists, prevents much of the panel motion and increases damping. Joists can also be shimmed at the hanger to assure even floor support. In tongue and groove flooring the individual planks can move relative to one another. Gluing or applying paraffin to the plank edges helps prevent this cause of squeak.

In some cases subflooring, made of wood strands bonded together with a resin material, has been found to contribute to floor squeak. When these materials deflect they rub against the nails, which powders the binder and opens up a small hole around the nail, which in

turn loosens the grip of the nail on the board. This effect can be offset somewhat by gluing under the flooring and using a gripping ring shank nail. Ring shank nails are recommended for nailing all wood diaphragms since they provide some additional grip on the plywood. To repair existing wood floors, screws can be added to cinch down the flooring to the joists and reduce panel movement. Glue should be applied from below along the top edges on both sides of the joists.

Floor Coverings

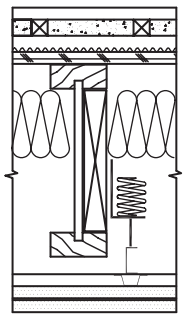
It is relatively easy to achieve high impact insulation class ratings by using carpet and pad. Medium quality ratings are achievable with a cushioned or padded vinyl floor surface. It is where hard materials such as quarry tile, marble, sheet vinyl, or hardwood floors are used that low impact ratings are encountered. As an example Fig. 12.11 shows the results of several IIC tests carried out on a minimum quality lightweight concrete and wood floor-ceiling construction. The ratings range from an IIC 76 for a heavy carpet and pad to an IIC 38 for exposed concrete.

Where hard surface floors are desired a spring isolated ceiling or concrete floating floor are usually required to achieve medium quality. Figure 12.12 gives an example. Thin layers of resilient material such as fiberglass board, cardboard-like materials, and wire mesh mats can raise the IIC ratings modestly, three to five points, but seldom provide sufficient deflection to achieve the impact isolation necessary for condominium construction. A number of products are commercially available, which are intended as a resilient underlayerment for hard-surfaced flooring. A typical construction consists of a plywood diaphragm on joists with the underlayerment above followed by a spanning cementatious board with tile set into a mortar bed on top of that. When combined with a drywall ceiling supported on resilient channel these constructions can provide IIC ratings a few points above minimum code. At this rating footfall noise is still clearly audible, and although the system may pass the minimum code requirements, it is unlikely to meet a condominium buyer's reasonable expectation of quality. The general failing of these materials is that they are so thin (1/4–3/8" thick) that they provide very little deflection. Hence their natural frequency of vibration is relatively high. When combined with the reduced impedance of the support system below due to the lack of lightweight concrete, their overall effectiveness is modest. Figure 12.22 shows the test results using a mesh mat underlayerment below a tile floor, which has an IIC 53 rating, barely above the minimum code.

The preferred floor covering in multifamily dwellings is carpet and pad. Hard surface floor coverings such as marble tile, quarry tile, and hardwood flooring are not recommended unless a spring-supported ceiling and a resilient floating floor are used. Even in these cases high quality IIC ratings are difficult to achieve. Figure 15.14 gives an example of a hardwood floor, which has achieved a field IIC rating of 69. The floating floor rests on 1" (25 mm) thick, 20 durometer dimpled rubber mats. The ceiling is supported on 1" deflection spring isolators. A similar installation with tile would likely have a lower rating. Note that the overall thickness of this construction is 18" (460 mm).

Where the appearance of wood or tile is desired, a hard surface can be used in non-walking areas such as within 1 to 2 feet (0.3–0.6 m) of a wall with carpet installed where walking traffic occurs. In kitchen and bathroom areas vinyl tile over a soft backing material such as a 20 durometer, 11/16" (17 mm) thick, dimpled rubber mat can provide reasonable IIC ratings, particularly when combined with a point-mounted resiliently suspended ceiling.

FIGURE 15.14 Double Isolated Wood Floor –Ceiling (Adams, 2002)



3/4" (19 mm) Wood Flooring on 1" (25 mm) Wood Sleepers
1" (25 mm) Lightweight Concrete Between the Sleepers
(110 lbs/cu ft or 1760 kg/cu m)
1" (25 mm) Thick Regupol QT2025.8 Underlayerment
3/4" (19 mm) Plywood Subfloor Glued and Screw Nailed
12" (300 mm) TJI Wood Joists
R-19 (6" or 150 mm) Batt Insulation
1" (25 mm) Deflection Spring Isolators
1 1/2" (38 mm) Carrying Channel
2 Layers 5/8" (16 mm) Gypsum Board on Hat Channel

100	125	160	200	250	315	400	500	630	800	1 K	1.3 K	1.6 K	2 K	2.5 K	3.2 K	FIC
47	49	48	48	49	47	41	38	36	33	30	29	28	27	25	20	69

15.4 PLUMBING AND PIPING NOISE

Noise from plumbing and piping is one of the most important causes of dissatisfaction in residential structures. It has become much more noticeable with the unfortunate use of plastic pipe in waste stacks, but it can also be caused by excessive flow velocities in supply pipes and be exacerbated by poor isolation. Plumbing and piping noise frequently originates with turbulent flow in pipes and fixtures and is transmitted primarily through vibrational coupling to the building structure and into the occupied spaces. Several other noise generation mechanisms are present in plumbing including cavitation, water hammer, vibrational transmission of pump or other mechanical noise, and water impact or splash noise; however, noise produced by turbulent flow is the primary source.

Supply Pipe

For normal velocities, the flow of water in straight residential supply pipes can be considered to be turbulent. In turbulent flow, regions of highly varying pressure are created, which transfer to the walls of the pipe and from there to the structure. Although turbulence is present in straight pipe, it is mainly caused by valves, fixtures, elbows, and constrictions. Several factors influence the noise generated by a supply pipe. The first is the velocity of flow within the pipe itself, which affects the amount of noise created by any valves or fixtures. The second is the way in which the pipe is attached to the structural framing, both to the structure and to the surface material.

Measurements have been made by Van Houten (1979), who investigated both these factors. Figure 15.15 shows his experimental setup. The test apparatus consisted of a double stud wall with one layer of drywall on each side and supply piping through the studs on the far side of the wall. Pressure was regulated with a valve and the pipe was terminated in a flexible hose. This is perhaps not the most realistic test condition since in a real situation the flow regulating fixture would probably be remotely located.

The attachment methods are sketched in Fig. 15.16. When pipes are routed through holes drilled in a series of studs it is rare that each hole is perfectly aligned with its neighbor. Consequently the pipe, which courses through these holes, will lie closer to the stud on one side or the other at each penetration. Isolators that are wedged into the holes will be pinched on the close side or will be loose if the holes are oversized. The preferred mounting method is to wrap felt around the pipe, outside the hole, and to band it with plumbers tape. This

FIGURE 15.15 Piping Noise Test Configuration (Van Houten, 1979)

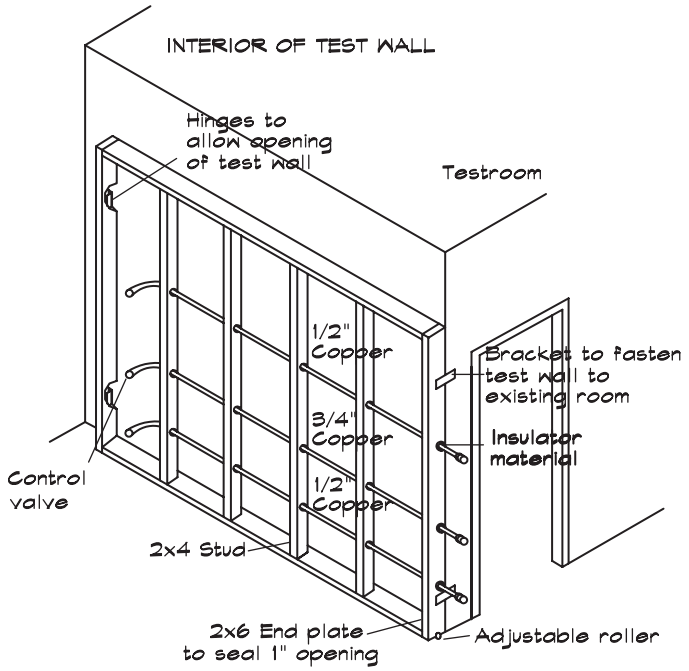


FIGURE 15.16 Pipe Mounting Methods and Descriptions (Van Houten, 1979)

MOUNTING	METHOD OF PIPE MOUNTING	ILLUSTRATION
A	Wood wedges, two wood wedges forced between the pipe and stud	Wood
B	"J" Hook nail ons, Type 3/4" E.M.T. nail on at each pipe penetratio of stud	Nail-on
C	Felt packing, 1/2" thick carpet felt packed between the pipe and stud	Felt
D	Pipe insulator, insulator without foam insert	Insulator
E	Pipe insulator, with polyethylene foam insert	Polyethylene foam
F	Rubber bands	Rubber bands

TABLE 15.7 Sound Levels Transmitted by Supply Piping, dBA (Van Houten, 1979)

Mounting Description		Supply Pressure			
1/2" Copper Water Pipe		30 psi	45 psi	60 psi	70 psi
A	Wood Wedges	50	54.5	56	57
B	J Hooks	50	54.5	56	
C	Felt Packing	44.5	48.5	49.5	
D	Plastic Pipe Insulator	51.5	55.5	57.5	
E	Foam Pipe Insulator	40.5	43.5	44	
F	Rubber Bands	35.5	35.5	31	
3/4" Copper Water Pipe					
A	Wood Wedges	47.5	51.5	52.5	
C	Felt Packing	42.5	45.5	46	
E	Foam Pipe Insulator	39.5	42.5	43.5	

allows the pipe to move without making contact with the stud. In general piping should be free to move slightly but must not touch the structure or the drywall. When a pipe cannot move, it indicates that it is being rigidly constrained, either by the structure or by an isolator that is overly compressed.

For a specific diameter pipe, the noise generated at a given back pressure is illustrated in Table 15.7. The data show the dependence of the noise level on pressure and mounting for 1/2" (13 mm) and 3/4" (19 mm) diameter pipe. Clearly there is substantial benefit to lower pressures; however in practice, pressures of 45 psi are usually required to maintain adequate flow. High pressures, on the order of 60 to 70 psi, are not necessary and should be avoided.

The data shown in Table 15.7 illustrate that there is little difference in noise level among the different types of rigid connections. Wood wedges, J hooks, and hard plastic pipe supports all lead to high levels even when contained in a double stud wall with batt insulation. There is substantial improvement afforded by resilient supports and the softer the mount the greater the isolation, although considerable effort is required to achieve satisfactory levels. This may be due in part to the setup of this experiment with the control valve upstream of the test section located close to the wall. It emphasizes the importance of keeping plumbing fixtures off party walls.

Table 15.7 lists the pipe diameter, mounting method, and back pressure as independent variables. In most real situations the flow volume is fixed by the downstream conditions. For example, if a pipe serves a shower nozzle the conditions at the outlet determine the flow rate. Upstream the piping can be sized to control the fluid velocity and thus the local turbulent noise. Careful selection of valves and nozzles can help reduce noise generation at the termination.

Intrusive noise due to plumbing and piping should be limited to the levels set forth in Table 15.8. These apply to any occupied space within a dwelling unit, including bathrooms. As we have previously discussed, as the level of quality increases the tolerance for neighbor-generated intrusive noise decreases.

In order to control turbulence-generated noise in supply piping a combination of several steps is required. First, the line pressure in the supply pipes should be below 60 psi. Second,

TABLE 15.8 Maximum Intrusive Noise Levels Due to Plumbing

Classification	SPL (dBA)
Minimum Quality	35
Medium Quality	30
High Quality	25

the pipes must be sized large enough that flow generated noise is kept to a minimum. Table 15.9 shows recommended pipe sizes and the maximum flow velocities and flow rates (Wilson, Ihrig & Associates, 1976).

A tub fills at a rate of about 8 gal/min so that a 1" (25 mm) pipe would be appropriate. A low-flow shower supplies water at about 3 gal/min so a minimum 3/4" (19 mm) pipe could be used. Where multiple fixtures are served the supply pipes should be scaled up accordingly. The larger pipe diameters allow local flow velocities to remain low even when it is necessary to neck down a pipe size to accommodate a valve or fixture. The third factor is wall construction. In walls containing piping that serves another unit, the wall surface should be a minimum of two layers of 5/8" (16 mm) drywall. There should be a full layer of batt insulation in all walls where supply, waste, or other fluid-carrying pipes are located.

The fourth factor, mechanical decoupling, is perhaps the most important. Piping must be physically decoupled from both support structure and from the wall surface. In party walls the piping should be supported only on the side of the wall that is served by the pipe. The type of resilient support is summarized in Table 15.10. Note that the isolation shown above is for water supply pipes. It is not required for vent stacks, fire sprinkler pipes, or gas pipes, although when vent stacks are rigidly attached to waste stacks they too should be isolated.

TABLE 15.9 Maximum Flow Velocities in Supply Pipe (Wilson, Ihrig & Associates, 1976)

Nominal Pipe Size (inches)	Max. Velocity (ft/sec)	Max. Flow Rate (gal/min)
1/2	1	1
3/4	2	3
1	3	8
1 1/4	3.2	15
1 1/2	3.5	22
2	4	42
2 1/2	5	74
3	6	138
3 1/2	6.5	200
4	7	277
5	7.5	467
6	8	720

TABLE 15.10 Vibration Isolation of Supply Piping

Nominal Pipe Size	Required Isolation Type	
(inches)	Horizontal	Vertical
1/2	1/4" Felt	1/4"Felt
3/4	3/8" Felt	3/8"Felt
1	3/8" Felt	3/8" Felt
1 1/4	3/8" Felt	3/8" Felt
1 1/2	3/8" Felt	3/8" Felt
2	1/2" Felt	WSW Pads
2 1/2	HN Hangers	WSW Pads
3	HN Hangers	ND Mounts
3 1/2	HN Hangers	ND Mounts
4	HS Hangers*	V Mounts*
5	HS Hangers*	V Mounts*
6	HS Hangers*	V Mounts*

*Nominal 1" deflection isolators

The details of the pipe isolation vary somewhat. For small diameters (< 2") the pipe can be secured with a full wrap of plumbers tape around a 1/4" (6 mm) felt pad. Foam and plastic isolators are commercially available, but they must contain a sufficient thickness (1/4–3/8") of foam or other isolating material to be effective. Hard-plastic mounts without soft inserts are ineffective. Neoprene in shear isolators can be effective in securing shower supply pipe as shown in Fig. 15.17.

With larger pipe sizes it may be more convenient to mount a number of pipes to a common structural frame, which is itself supported on neoprene mounts as Fig. 15.18 shows. Vertical riser pipes are held in place with a pipe clamp, which then rests on neoprene mounts or pads, as in Fig. 15.19.

In critical applications such as high-quality construction, it is sometimes necessary to wrap supply pipes with foam lead sheets where the pipes are located within walls or floor-ceilings of other occupied spaces, or where they are contained in the wall of a bedroom. Foam lead sheets consist of 1/4" (6 mm) foam adhesively attached to both sides of a 2 lb/sq ft (9.8 kg/sq m) lead sheet approximately 1/16" (1.6 mm) thick, which provides both vibration isolation and noise reduction. They can be cut to the appropriate size with shears and tied

FIGURE 15.17 Felt Isolated Shower Attachment

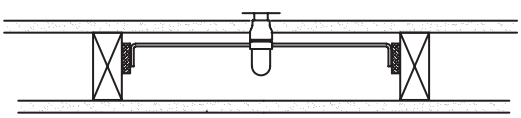


FIGURE 15.18 Resilient Multiple Pipe Support

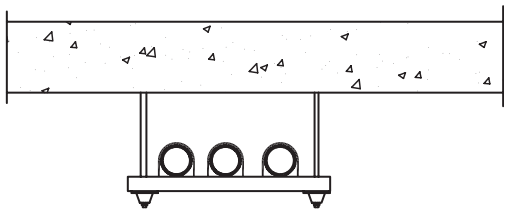
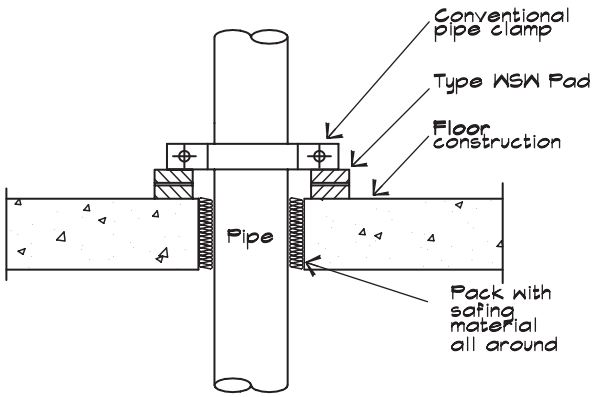


FIGURE 15.19 Floor Penetration Detail for Pipe Risers



with wires so that they completely encircle the pipe and overlap at least an inch (2.5 cm). Additional felt at points of support is not required for foam lead wrapped pipe.

Water Hammer

Water hammer is a shock wave, usually generated by the rapid closure of a valve, but occasionally caused by a pressure wave resonance within a pipe. In the case of an isolated occurrence, it produces a violent slamming, which physically moves the pipe and causes a loud banging noise. In the case of a resonance the passage of a shock wave back and forth results in a very loud low-frequency buzzing sound.

Water hammer creates a dramatic effect, which if allowed to continue, can result in damage to the pipe. Control is effected by exposing the fluid stream to a confined air pocket, which allows the high pressure shock to expand into a pressure release surface. These devices, called water hammer arrestors, contain a gas trapped in a flexible bladder that encircles the fluid or is contained in a separate branch on a threaded T-fixture. In residential structures

such devices are very effective when mounted to the outlet pipe of the water heater and near any quick-closing valves, such as automatic sprinkler systems. Occasionally they may also be necessary to control resonances in long runs of pipe or near a particularly troublesome valve.

Waste Stacks

Waste stacks, which carry water and effluent from toilets and other fixtures, are a frequent cause of noise complaints, particularly when plastic (PVC or ABS) pipe is used. This lightweight pipe transmits annoying levels of noise when water flows through it. When a toilet is flushed upstairs and the water rushes through a plastic pipe, levels as high as 60 to 65 dBA have been measured in downstairs units. This is loud enough that it makes normal conversations difficult and is certainly a show-stopper for the party guests. The cause is the lightweight plastic material, which vibrates due to the turbulent flow created by the passage of the waste water, and any direct coupling of the pipe to the structure. Consequently plastic waste stacks should never be used in residential buildings or sensitive commercial structures such as offices, classrooms, hospitals, studios, or theaters even for short runs such as P-traps.

Instead of plastic, cast iron is the preferred material for waste stacks due to its mass. Even with cast iron problems may arise if the pipe is not properly isolated. Waste stacks must be vibrationally isolated just as supply pipes were isolated. The inherent mass of cast iron and the relatively slow fluid velocity makes the isolation requirements less stringent than similarly sized supply pipes. Vertical risers should be supported on type WSW neoprene waffle pads as shown in Fig. 15.19. Horizontal runs may be isolated with 1/2" (13 mm) felt or neoprene pads. Frequently waste stacks exceed 2.5" (64 mm) in diameter, and when they are wrapped with a felt isolation material the combination may not fit within a single stud space without binding. Thus it is prudent to allow at least a 2 × 6 stud for all walls housing waste stacks. This permits a neatly drilled round hole of sufficient size to allow the pipe to pass through the base plate or stud without touching the structure.

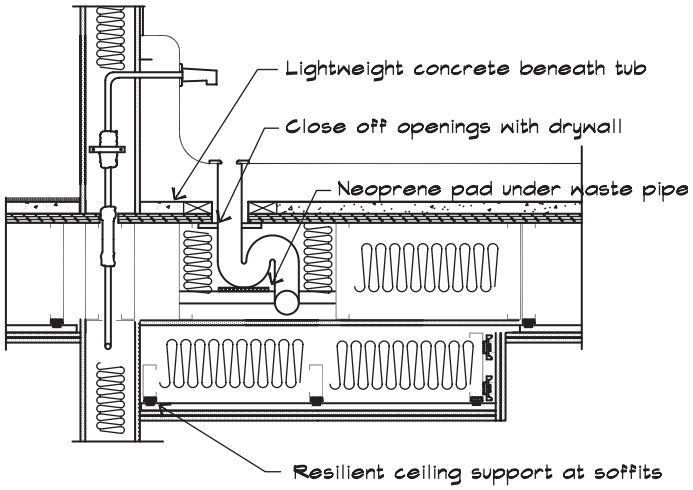
When treating an existing condition where plastic pipe has been used as a waste stack, noise levels have been reduced to inaudibility by decoupling the pipe from the structure, wrapping it with 2 lb/sq ft (9.8 kg/sq m) foam lead sheets, installing R-11 batt insulation in the airspace, and applying two layers of 5/8" (16 mm) drywall on resilient channel to the stud. In some cases the replacement of plastic pipe with cast iron pipe may be less expensive. Note that if the plastic pipe is replaced it should be removed completely for its entire length.

Tubs, Toilets, and Showers

Tubs, toilets, and showers should be installed over a floor-ceiling, which is equivalent to that constructed in the rest of the residence. In wood construction if the floor has lightweight concrete it should extend under the tub or shower. Where fiberglass tubs are installed, it is good practice to pour additional lightweight concrete around the bottom of the tub to add mass and damping.

Below the tub, where the piping penetrates the floor, builders sometimes leave a sizable opening in the floor for ease of installation. This must be closed for both acoustical and fire

FIGURE 15.20 Bathtub Installation Detail



code reasons using at least two layers of 5/8" (16 mm) drywall as in Fig. 15.20. Where the P-trap is supported on a crosspiece the pipe should rest on a neoprene waffle pad isolator consistent with its size.

Since the noise generated by supply piping is velocity dependent, low-flow devices like an aerated showerhead can reduce supply-pipe noise. Upstream noise from the pipe serving a toilet can be reduced by restricting the flow locally at the wall valve. Careful selection of quiet fixtures such as ball valves can prevent noise creation at a critical location.

Pump and Piping Vibrations

When a pump is located in a residential or other noise-sensitive structure the pulsations imparted to both the pipe and the driven fluid can create noise both by vibrational and airborne transmission paths. Control of pump vibrations begins with the location of the pump in a nonsensitive area such as a basement; however, it is not uncommon to find rooftop installations in large condominium complexes. Both the pump and the pipe must be properly isolated, beginning with a concrete housekeeping pad and inertial base such as that shown in Fig. 13.6. Any piping attached to isolated equipment, which includes not only pumps but cooling towers, compressors, boilers, refrigeration equipment, and fan coil units, must itself be isolated with an isolator having the same deflection as the isolated equipment for a distance of 50 feet or 50 pipe diameters, whichever is greater. At the first point of support an isolator having twice the pump deflection can be used, since when the pipe is drained it weighs considerably less than when it is full. The pump isolators are installed when the pipe is empty, and when it is filled the pump and pipe assembly drops down under the increased load. This motion may crack the pipe unless it is given the freedom to move. All piping must be resiliently supported either from the pump's inertial base or with separate vibration isolators. In addition to isolating the pipe, flexible couplings can be installed near the pump. These serve to compensate for minor pipe misalignments and provide some vibrational decoupling for impulses transmitted along the pipe.

Fluid Pulsations

Occasionally the pulsations induced by a pump impeller will travel from the pump along the pipe and into a structure, through mechanical coupling between the pipe and a wall or floor. In existing buildings some reduction can be obtained by inserting a flexible coupling in the pipe between the pump and the receiver.

In high-pressure lines the impeller impulses may also be transmitted within the fluid so that a pulse dampener is recommended on the supply side of the pump after the flexible coupling. Pulse dampeners work in much the same way as a water hammer arrestor with an encapsulated gas-filled bladder. The pressure surge is allowed to expand into the bladder, which removes it from the liquid in the pipe. These devices are commercially available and are quite effective for attenuating vibrations in high-pressure lines including hydraulic lines.

15.5 MECHANICAL EQUIPMENT

Split Systems

Mechanical systems in residential structures provide heating, ventilating, and air conditioning (HVAC) services to the building. The thermodynamic principles were discussed in Chapt. 13. A typical system consists of an external air-cooled condenser, which gives up heat to the outdoors in the process of compressing a circulating refrigerant. The refrigerant is pumped through copper pipes to a fan coil unit where it is forced through an expansion valve and a coil, where it changes state from a liquid to a gas and absorbs heat from its surroundings. Air is blown over the cooling coil through which the refrigerant passes and heat is transferred to the coils and then to the refrigerant. Heat can be generated by an electric coil or separate gas-fired boiler.

Roof-mounted compressors, unless properly configured, can be a source of annoying vibration. Units should be installed as in Fig. 15.21. In wood buildings compressors are set on small platforms built of 3” (75 mm) lightweight concrete pads extending at least 12” (150 mm) beyond the outside of the unit. The unit itself is supported on 1” (25 mm) deflection spring isolators. Any electrical connections to the compressor are made using a minimum of 3’ (1 m) flexible wiring, which allows the unit to move freely back and forth and up and

FIGURE 15.21 Rooftop Condenser Installation

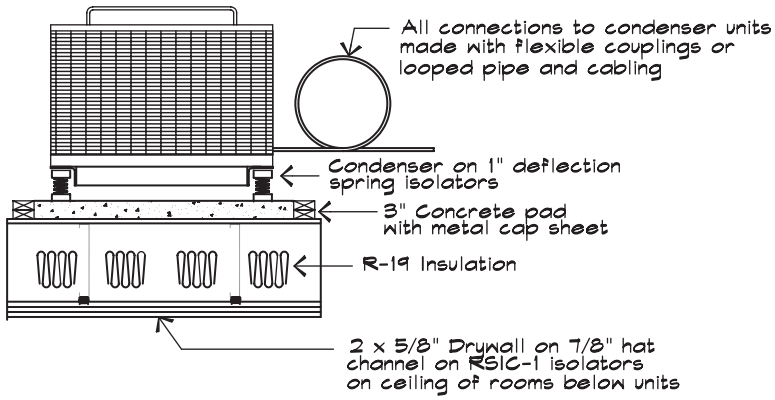
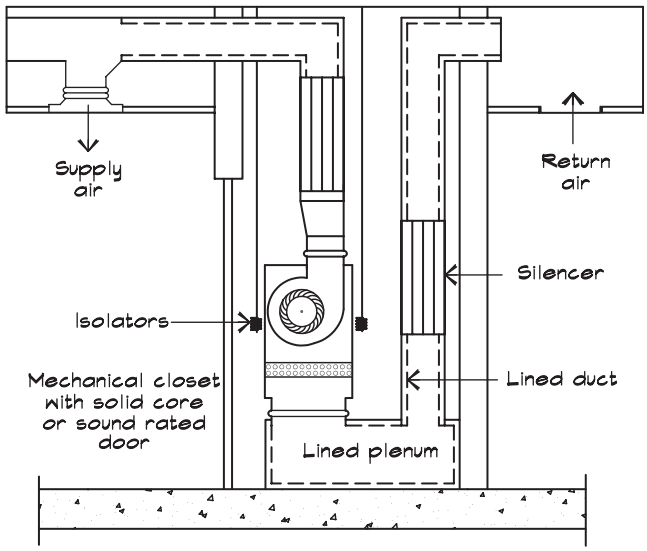


FIGURE 15.22 Vertical Fan Coil Unit



down on its isolators. Since the refrigerant lines are connected rigidly to the compressor it is good practice to wrap them one full turn around the unit before they penetrate the roof. This reduces the transmitted vibration that travels along the tubing. Refrigerant lines can create noise in the 35 to 40 dBA range if improperly isolated. They should be wrapped with foam insulation and be routed through a chase or within stacked fan coil closets. Where refrigerant lines are located in walls adjacent to critical areas such as bedrooms they should be wrapped with foam lead sheets. In any case they must not make direct contact with the structure or drywall.

Fan coil units are also a source of noise in residential buildings. They can be located in an attic in single-family residences or in a closet in multifamily dwellings and usually require a 3' (1 m) medium pressure drop silencer or equivalent loss in lined duct and elbows on the supply and return ducts to control duct borne noise. One effective mounting method is to arrange the fan coil vertically in a closet with a ducted supply and return as in Fig. 15.22.

Where the air is returned to the closet through a louvered panel located under the fan coil, noise levels are generally unacceptable, ranging from 45 to 55 dBA in the room containing the intake. Lining the area beneath the fan coil is of marginal value. Generally this space is too small to act as a plenum and at best becomes a lined elbow, which provides only a few dB of loss. Figure 15.23 shows a way of installing a fan coil or heat pump without a ducted return so that the return-air noise transmission path is attenuated by using a transfer-duct silencer. A heavy solid-core or sound-rated door with air-tight weather stripping, along with 2" (50 mm) fiberglass duct-liner board and 0.4" (10 mm) deflection neoprene isolators are necessary to properly isolate a vertical fan coil unit. Detailed calculations based on the casing radiated sound power levels of the unit can clarify the level of isolation required.

Packaged Units

Small packaged air handling units can be rooftop-mounted and contain a compressor, fan coil, and cooling fan in a single unit. These units are available in a side-discharge or

FIGURE 15.23 Fan Coil Unit with a Transfer Duct

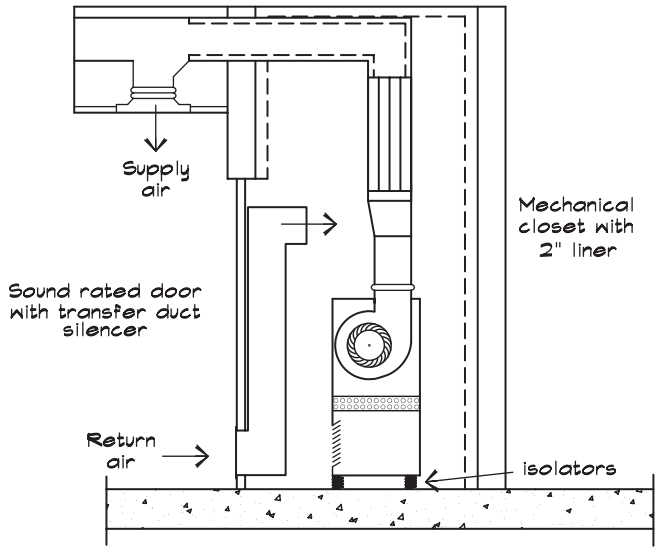
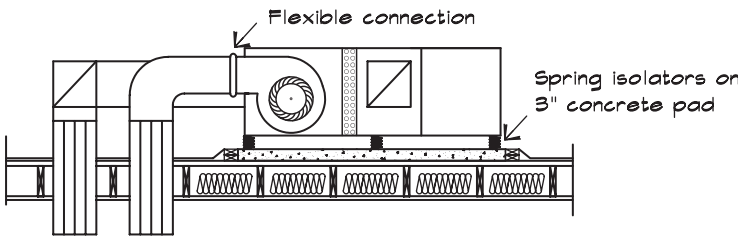


FIGURE 15.24 Roof-Mounted Packaged Air Handlers



Duct silencer at the roof penetration

down-discharge configuration. For purposes of noise control, the side-discharge units are much preferred since this allows space for the installation of silencers in a horizontal run of duct. In down-blast units, the air and noise issue straight down out of the unit, allowing very little room for a silencer or other attenuating mechanism. Figure 15.24 shows the preferred configuration for side-discharge units with silencers installed in the vertical drop. This location is optimal for the attenuation of both duct borne noise and exterior noise breaking into the duct.

Sound power levels associated with small packaged units vary according to the fan scaling laws discussed in Chapt. 13. They usually require some lined duct and either a 3’ (1 m) or a 5’ (1.5 m) medium pressure drop silencer to reduce levels to below an NC 30. An analysis along the lines of the calculation set forth in Chapt. 14 can yield the required attenuation.

Vibration isolation for these packaged units is similar to that for compressors. Rooftop units should be set on 3” (150 mm) lightweight concrete housekeeping pads and supported on 1” (25 mm) deflection spring isolators. In areas requiring seismic restraint separate limit

stops are installed. Electrical connections are by means of 3' (1 m) long flexible cables. A flexible connection is used to couple the duct to the package unit. When they are located above critical areas, roof-mounted ductwork should be supported on neoprene isolators having a 0.25" (6 mm) deflection. Any condensate lines should be supported from the isolated equipment and not from the roof deck.

Roof-mounted boilers and associated piping must also be vibration isolated. Boilers with integral pumps need 1" (25 mm) deflection isolators and those with separate pumps should have 0.25" (6 mm) deflection neoprene mounts. Gas lines, which cannot include flexible connections, attached to isolated equipment should have four or more 90° bends separated by 3' (1 m) of straight pipe and be supported on 0.25" (6 mm) deflection neoprene isolators.

15.6 APPLIANCES AND OTHER SOURCES OF NOISE

Stairways

Footfall impacts on stairways can be conducted structurally through the studs, which support the stair risers. This, in turn, is radiated by the panels attached to the wall studs. In townhouse (two-story) condominiums and apartments it is good practice to locate the stairways on the interior of the unit away from the party walls. When this is not possible the stairways should be supported separately and the risers must not be physically attached to the wall framing or drywall. The party wall surface must be continuous behind the stairs and the treads, and risers and framing must not touch the drywall. Separate pipe columns may be used to support the stair landing and the diagonal supports should be attached only at the landing and the floor.

Appliances

In critical residential structures such as condominium units, free-standing appliances such as washers, dryers, and dishwashers should be isolated from the floor by using neoprene mounts. When this arrangement affects the lateral stability of the unit such as during a spin cycle in a washing machine, a lateral stabilizer may be necessary. This, too, should be in the form of a neoprene isolator having the same stiffness as the vertical one. The lateral stabilizer can be bolted to a crosspiece of plywood, which in turn is bolted to the washer. Quick acting valves in washing machines can cause water hammer in certain cases. Generally the hoses used to connect to the supply pipes have sufficient flexibility to dissipate the pressure pulse. Where this is not the case a water hammer arrestor located near the valve on the supply pipe will solve the problem and extend the life of the hose.

Garbage disposals present a unique set of problems for vibration isolation. Internally isolated units have become available, which have a shell within a shell construction and internal neoprene mounts. These are much preferred over the standard models, but they must be installed with a piece of flexible hose approximately 18" (450 mm) long to decouple the waste pipe from the disposal.

Jacuzzis

Jacuzzi tubs have an aerator, which mixes air with the tub water, and a circulation pump forces the mixture back into the tub in a closed-loop cycle. The combination can produce relatively high noise levels in multifamily dwellings both directly under the tub as well as in rooms adjacent to the epicenter. Noise levels as high as 50 to 60 dBA have been measured

directly beneath a Jacuzzi tub in wood structures with lightweight concrete floors, and in the 40 to 50 dBA range in nearby rooms one floor below. Resiliently mounting the motor and pump provide some relief, on the order of 5 dB, but not enough to meet the levels set forth in Table 15.8.

In concrete structures Jacuzzi tubs have been installed on 4" (100 mm) concrete floating floors with 1" (25 mm) deflection spring isolators on a 6" (150 mm) slab. This approach addresses the sound and vibration problem but introduces a significant floor height change up to the base of the tub. The variation in floor height with the addition of water and bathers must also be accommodated. In general Jacuzzi tubs should not be installed in multifamily dwellings above the first floor.

Trash Chutes

Trash chutes are constructed of sheet metal tubes mounted in shafts, which carry trash from upper floors to a receptacle in the basement. They are best located in a masonry shaft, which is not on a wall of a dwelling unit. Shafts can be constructed of grouted concrete block units. Where they are located on common walls an additional batted stud wall with two layers of drywall, which does not make physical contact with the shaft wall, should be used to reduce their impact. Some authors (Harris, 1994) recommend that the steel trash chutes themselves be isolated and that the trash receptacles be covered with damping material.

Elevator Shafts and Equipment Rooms

Elevator shafts in multifamily dwellings and in office complexes are best located in a separate utility core, which is not contiguous to occupied spaces. Even if an elevator equipment room is remotely located, the elevator passage itself can cause noise levels in the 50 dBA range due to the vibration transmitted through the vertical guide rails. Where elevator shafts are located adjacent to occupied spaces a double stud double drywall wall has been used. Even in these cases vibration may still be transmitted into the adjacent space via the direct coupling of the rails to the slab or floor system.

Elevator equipment is either hydraulic or electric. Hydraulic equipment is located near the bottom of the shaft and consists of a pump, associated piping, and a thrust cylinder. The pump and piping is of greatest concern whereas the cylinder is rarely a factor. When the pump is located on a structural slab on grade below occupied areas it rarely presents a problem. Hydraulic pumps above grade should be vibration isolated using neoprene mounts. Code requirements prohibit flexible hose from being used as hydraulic lines so solid piping should be separated from the structure using isolators.

Electric winches control the elevator height by a cable wound on a drum located at the top of the shaft. In critical applications the drum and winch can be isolated by mounting the entire assembly on a concrete slab, which is set on isolators. Due to safety considerations, the mechanism must be designed so as to tolerate a failure or bottoming out of the isolation system.

Garage Doors

Where parking is provided below a multifamily dwelling or office building steel gates sometimes are installed for security. In an overhead installation the gate is supported at the sides on two vertical pipe columns attached top and bottom to the slabs. An electric motor hung from the slab above drives a chain or gear mechanism, which raises and lowers the gate.

When these gates are attached in this manner the vibration from the opening and closing of the gate can produce considerable annoyance to occupants of the floor above.

A preferable alternative to the overhead attachment is a side-opening gate, which is supported from below. Here the gate rides on a V rail and rolls to the side, driven by a chain drive motor bolted to the floor. In these installations a simple neoprene isolator supporting the track attached to the upper slab is usually sufficient to control the transmitted vibration. The important feature is that the upper rail system must not be directly coupled to the slab above. Where there is insufficient space for a side-opening gate an overhead gate can be supported using an independent structure on steel columns that do not make contact with the floor above.

16 DESIGN AND CONSTRUCTION OF OFFICE BUILDINGS

Commercial buildings present a set of design issues that differ somewhat from those encountered in residential structures. In critical spaces such as classrooms, theaters, and studios, much of the information presented in Chapt. 15 is still applicable. In commercial office buildings the uses can vary, and may include less sensitive spaces. Structures are likely to be multistory, with an air handling unit on each floor located in a central core, along with other services such as elevator shafts, stairwells, and bathrooms. Mechanical equipment is also located on the roof, sometimes directly above the most prestigious and expensive floor space. The main air ducts are sized, based on the clearance afforded by the ceiling heights, and velocities may be relatively high. Air is likely to be returned through a common plenum, which complicates the room-to-room noise transmission problem.

In high-rise office towers exterior walls can be continuously glazed curtain walls, framed outside the flooring system, with narrow mullions, leaving little opportunity for closure of gaps at the interior walls and floors. Floor coverings may not be selected based on considerations of footfall noise. Conflicting uses may abut one another. These and many more details, which are critical to a satisfactory work environment, may not be totally under the control of the acoustical engineer, but nevertheless should be addressed.

16.1 SPEECH PRIVACY IN OPEN OFFICES

Privacy

When the intelligibility of speech is low, not surprisingly it follows that the privacy is high. Table 16.1 (ANSI S3.5) shows the relationship between Articulation Index, signal-to-noise ratio, and privacy. Written characterizations of these degrees of privacy were given in Table. 3.6.

In an office environment, where speech privacy is the goal, we can strive to achieve the required signal-to-noise ratio in several ways. The general form of the intelligibility equation is

$$\begin{bmatrix}\text{Source}\\ \text{Sound}\\ \text{Level}\end{bmatrix} - \begin{bmatrix}\text{Sound}\\ \text{Attenuation}\\ \text{(Reduction)}\end{bmatrix} - \begin{bmatrix}\text{Masking}\\ \text{Sound}\\ \text{Level}\end{bmatrix} = \begin{bmatrix}\text{Signal to}\\ \text{Noise Ratio}\\ \text{(Privacy)}\end{bmatrix} \tag{16.1}$$

TABLE 16.1 The Relationship between Intelligibility and Privacy

Articulation Index	Signal to Noise	% Sentences Understood	Intelligibility	Privacy
> 0.4	> 0 dB	> 90	Very Good	None
0.3	−3 dB	80	Good	Poor
0.2	−6 dB	50	Fair	Transitional
0.1	−9 dB	20	Poor	Normal
< 0.05	−12 dB	0	Very Poor	Confidential

Clearly we have three possible ways of influencing the signal-to-noise ratio: 1) control the sound source, 2) increase the path attenuation, and 3) raise the masking sound level. The source can be oriented to take advantage of the natural directivity of the human voice. The direct sound path can be controlled by using full or partial height barriers. Reflections can be attenuated with absorptive materials, and we can electronically generate background noise to mask intrusive speech. The details of each of these features depend on the office configuration and other design considerations.

Privacy Calculations

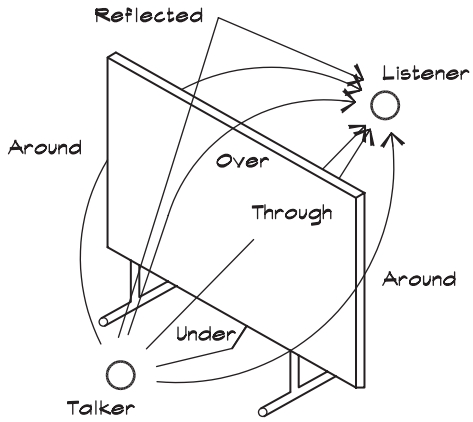
Calculations based on Eq. 16.1 can be carried out in individual third-octave or full-octave bands. The most accurate method is to use third octaves and the Articulation Index to determine intelligibility; however, there are also useful composite methods. Articulation calculations begin with the layout of the working environment.

Offices are configured either as separate rooms with full or partial height walls or as an open plan layout. Open-office plan refers to a system of workstations, distributed about an open floor, separated by partial height barriers. This design approach yields a flexible and relatively inexpensive work space, which if properly designed can furnish a degree of privacy for telephone and other conversations. Originally developed in the 1960s, its effectiveness unfortunately was oversold at first and its reputation subsequently suffered from unfulfilled expectations. Problems also arose when the system was only partially implemented.

Privacy in an open-office work environment can be achieved only when all the critical components are present and properly implemented. These include: 1) careful arrangement of the furniture and occupants, including the orientation of both talkers and listeners; 2) partial height barriers of the correct type, height, and location; 3) highly absorbent ceiling and wall panels; and 4) masking sound having the proper spectral content and level.

Chanaud (1983) and others have developed detailed methodologies for evaluating speech privacy in open offices by calculating the speech intelligibility between workstations. All calculations are based on Eq. 16.1; in the office environment, however, there is a large number of potential sound paths to be considered. Figure 16.1 shows several, each of which must be evaluated to arrive at the final signal-to-noise ratio at a receiver. Figure 16.2 presents a diagram of the separate calculations broken down into individual steps. The final result is a composite signal-to-noise ratio, which leads to the Articulation Index at the receiver. Note that each of these calculations is based on direct-field transmission, where the sound wave has undergone at most one or two reflections. In open-office environments a classic

FIGURE 16.1 Panels and Screens as Speech Barriers



reverberant field does not exist, particularly with intervening barriers and highly absorptive ceilings.

The Articulation Index (AI) is calculated from partial articulation indices (PAI), which are signal-to-noise ratios in each third-octave band, weighted according to the importance of the band for the understanding of speech. The Articulation Index is the sum of these individual weighted signal-to-noise ratios

$$\mathrm{AI} = \sum_{i=200}^{5000} \mathrm{PAI}_i \tag{16.2}$$

The signal-to-noise ratio in a particular third-octave band, between 200 Hz and 5000 Hz, is calculated from the source strength, the particular path attenuation (which is called the speech reduction), and the masking spectrum present at the receiver location. Each partial Articulation Index is determined from the signal-to-noise ratio multiplied by the weighting factor particular to a given band

$$\mathrm{PAI}_i = (\mathrm{VS}_i - \mathrm{SR}_i - \mathrm{MS}_i)\mathrm{WF}_i \tag{16.3}$$

where PAI = partial Articulation Index in a given third-octave band

VS = average peak voice spectrum, in a given third-octave band, of the male voice at one meter, on axis (dB)

SR = speech reduction—difference between the VS level and the level at the receiver in a given third-octave band (dB)

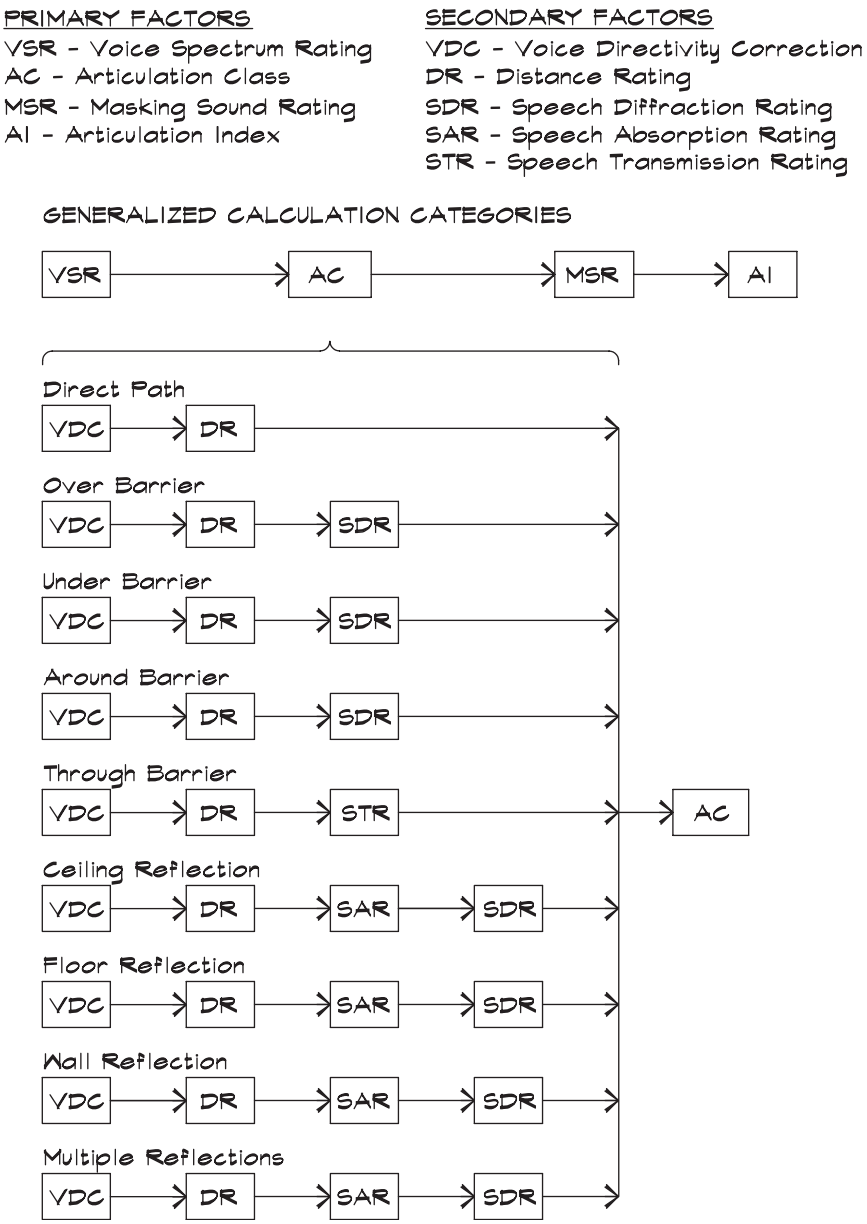
MS = third-octave masking sound spectrum (dB)

WF = third-octave Articulation Index weighting factor

All calculations are limited by conditions on the partial AI components such that

$$\begin{aligned} &\text{If } \mathrm{PAI}_i < 0 \quad \text{then set} \quad \mathrm{PAI}_i = 0 \\ &\mathrm{PAI}_i > 30\ \mathrm{WF}_i \quad \text{then} \quad \mathrm{PAI}_i = 30\ \mathrm{WF}_i \end{aligned} \tag{16.4}$$

FIGURE 16.2 Open Plan Speech Rating Diagram (Chanaud, 1983)



We begin with the sound pressure level and spectrum generated by a male voice adjusted for voice effort and weight it according to the way the ear hears. The voice spectrum (VS) in Fig. 16.3 is the third-octave sound pressure level of the male voice peaks, which differs somewhat from the energy average levels cited previously. The levels in each band are weighted with the AI speech weighting factors, WF, given in Table 16.2. For simplicity we can use the Speech Rating Factor (SRF), which is 30 times the AI weighting factor (WF) and sums to one.

FIGURE 16.3 Average Peak Male Speech Spectra (ANSI S3.5, 1997)

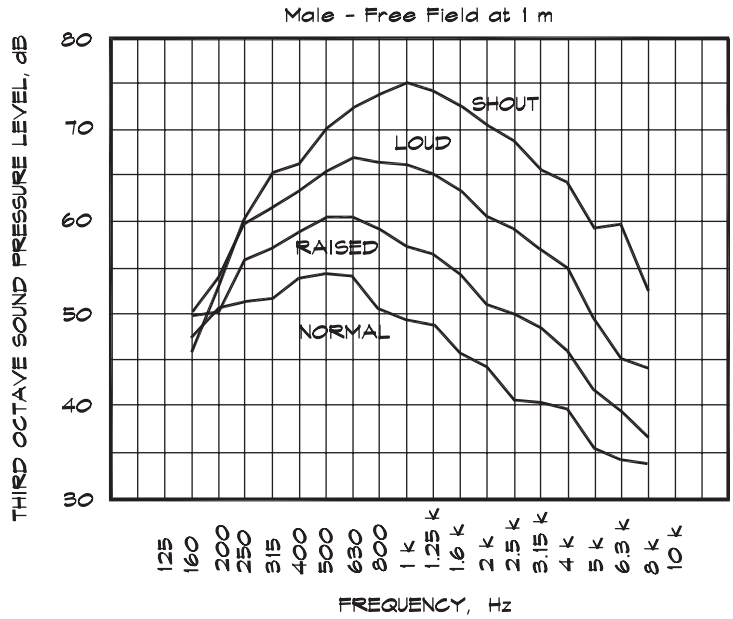


TABLE 16.2 Speech Weighting Factors (ASTM E1110-86)

	One Third Octave		One Octave		
Freq.	**SRF**	**WF**	**SRF**	**WF**	**A-Weighting**
200	.012	.0004			−11
250	.030	.0010	.072	.0024	−9
315	.030	.0010			−7
400	.042	.0014			−5
500	.042	.0014	.144	.0048	−3
630	.060	.0020			−2
800	.060	.0020			−1
1000	.072	.0024	.222	.0074	0
1250	.090	.0030			1
1600	.111	.0037			1
2000	.114	.0038	.328	.0109	1
2500	.103	.0034			1
3150	.102	.0034			1
4000	.072	.0024	.234	.0078	1
5000	.060	.0020			1

The speech reduction (SR) is calculated from the directivity relative to the on-axis level associated with the listener direction, the loss due to distance, and the attenuation due to the interaction with objects associated with the particular sound path, including diffraction over barriers, transmission through barriers, or absorption due to reflections. When each path level is computed the overall level in each third-octave band is determined and we compare the composite sound pressure level with the masking sound spectrum $(MS)_i$ to calculate the Articulation Index and the degree of privacy.

To make the AI easier to understand, a metric called the Privacy Index (Chanaud, 1983) was introduced, which increases as the isolation increases. The Privacy Index (PI) is defined as the percentage of privacy in terms of the Articulation Index as

$$PI = 100\,(1 - AI) \tag{16.5}$$

The Privacy Index is easier to understand for many users. For example, a 95% Privacy Index is excellent privacy, and 60% is poor.

Articulation Weighted Ratings

The Articulation Index can be estimated (Chanaud, 1983) from three ratings instead of being calculated individually in each third-octave band.

$$AI \cong \frac{1}{30}\,[VSR - AC - MSR] \tag{16.6}$$

where the AI weightings have been included in the individual rating terms. The ratings are the voice spectrum rating (VSR), which is based on voice level; the speech reduction rating (SRR), originally developed by Chanaud (1983), which has been standardized (ASTM E1110-86) and renamed the Articulation Class (AC); and the masking spectrum rating (MSR) defined in Eq. 16.9.

$$VSR = \sum_{i=200}^{5000} (VS_i)\,(SRF_i) \tag{16.7}$$

$$AC = \sum_{i=200}^{5000} (SR_i)\,(SRF_i) \tag{16.8}$$

$$MSR = \sum_{i=200}^{5000} (MS_i)\,(SRF_i) \tag{16.9}$$

The speech reduction calculation varies depending on the attenuation mechanism associated with a particular path. Each path attenuation can also be discussed in terms of a composite rating. The Articulation Class (AC), as contrasted to the STC rating, is a weighted measure of the noise reduction between two given positions. It can be determined by calculating the losses associated with each sound path.

For example, if the sound is diffracted over a barrier, we can calculate the barrier loss and the associated speech diffraction rating (SDR).

$$\mathrm{SDR} = \sum_{i=200}^{5000} (\Delta L_B)_i \mathrm{SRF}_i \tag{16.10}$$

For transmission through a barrier, the sound attenuation is the point-to-point direct-field noise reduction, based on the direct-field transmission loss at a given angle and frequency. The speech transmission rating (STR) is written as

$$\mathrm{STR} = \sum_{i=200}^{5000} \left\{ \left[\Delta L_{TL}(\theta) \right]_i \mathrm{SRF}_i \right\} \tag{16.11}$$

For surface reflections the speech absorption rating (SAR) is

$$\mathrm{SAR} = \sum_{i=200}^{5000} \left\{ \left[\Delta L_s \right]_i \mathrm{SRF}_i \right\} \tag{16.12}$$

where ΔL_s is the attenuation due to a reflection, based on the specular absorption coefficient given in Eq. 16.13.

These rating simplifications have the advantage of allowing a general discussion of the components, while retaining most of the original accuracy. The differences between the results using the separate calculation method and the one-number ratings are shown in Fig. 16.4, where Eq. 16.5 has been inserted into Eq. 16.6 for the estimation of the Privacy Index. The only thing that is dropped in this generalization is the limitation imposed under Eq. 16.4, which leads to a slight over-design at high isolation values.

Experience has shown that there is no single masking spectrum, which provides the most privacy at the lowest overall sound level, and is best for every situation. The shape of the spectrum depends critically on the details of the sound attenuation. Chanaud (2002)

FIGURE 16.4 Privacy Index vs Articulation Class Rating (Chanaud, 2000)

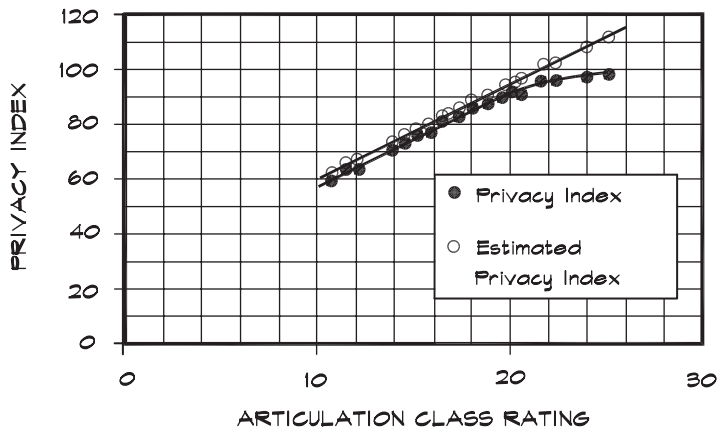


TABLE 16.3 Masking Sound Spectra (Chanaud, 2002)

	One-Third Octave		One Octave	
Freq.	**Open Office**	**Closed Office**	**Open Office**	**Closed Office**
200	**43**	**43**		
250	**43**	**41**	**47**	**46**
315	**42**	**40**		
400	**41**	**39**		
500	**41**	**37**	**45**	**42**
630	**40**	**35**		
800	**39**	**34**		
1000	**38**	**32**	**43**	**37**
1250	**36**	**30**		
1600	**34**	**29**		
2000	**32**	**27**	**37**	**32**
2500	**30**	**25**		
3150	**28**	**23**		
4000	**26**	**21**	**31**	**26**
5000	**24**	**19**		
dBA	**47**	**43**		
MSR	**41**	**37**		

suggests two good compromise spectra, one for open offices and another in closed offices, in Table 16.3.

Speech Reduction Rating and Privacy

In a calculation of the Articulation Index, based on the geometry sketched in Fig. 16.5, we begin with the source level and associated spectrum, and calculate the attenuation due to the directivity of the talker, the intervening distance, and barriers or absorbers, finally comparing the level in third octaves to the background level. The masking spectrum (MS) is the actual background noise in the receiving space.

For purposes of this analysis it has been assumed to be the open-office spectrum from Table 16.3. The worst case (least private) condition, at a given distance, would be represented by a face-to-face orientation with no intervening barriers. Table 16.4 shows the results of such a calculation, giving a compilation of the privacy expected from various speech reduction ratings, for a person talking at a normal voice level based on face-to-face orientation and an open-office masking sound level.

What we learn from this figure is that the entire span of privacy ratings falls within a 10 dB range of levels, and that a change of 1 dB in a rating results in a change of .03 in the Articulation Index. Thus even a one-point difference can have a noticeable effect on the degree of privacy. Given these relationships, we can refer to this chart when discussing speech privacy in terms of calculated AC values in more complicated configurations.

FIGURE 16.5 Noise Reduction Ratings by Path

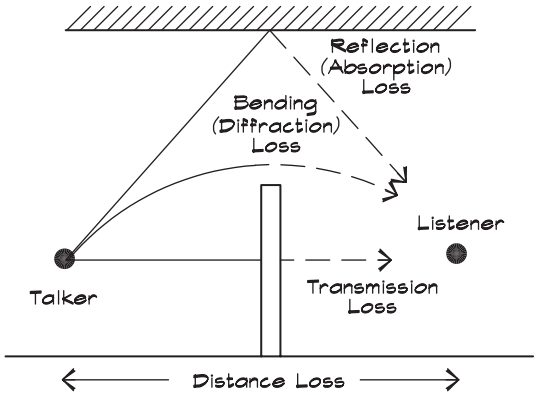


TABLE 16.4 Influence of Sound Attenuation on Speech Privacy (Chanaud, 1983)

Articulation Class	Articulation Index	Privacy Index	Degree of Privacy
16	.37	63	None
17	.33	67	None
18	.30	70	Poor
19	.27	73	Poor
20	.23	77	Poor
21	.20	80	Transitional
22	.17	83	Transitional
23	.13	87	Normal
24	.10	90	Normal
25	.07	93	Normal
26	.05	95	Confidential

Source Control

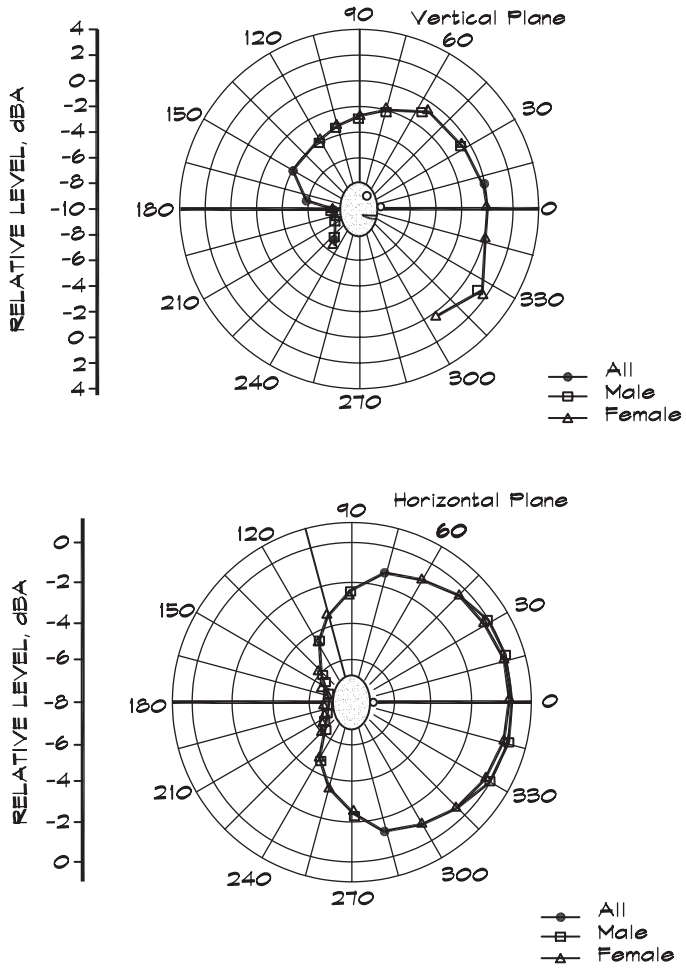
In open plan analyses the source model is the male voice, which has a characteristic spectrum, level, and directivity. In the analysis of speech, average voice peaks from Fig. 16.3 are used. The overall changes can be approximated by the vocal effort table in Table 16.5.

There are characteristic directivities associated with the human voice, which are shown in Fig. 16.6, relative to on-axis. It is assumed that the receiver has no directivity. From this table we see that a degree of natural attenuation may be achieved by taking advantage of the directivity of the human voice and the orientation of the talker and listener. If workstations are arranged so that the line between two conversing individuals in one workstation is at right angles to the direction of the listener in the next workstation, the directivity correction is maximized.

TABLE 16.5 Voice Level Corrections

Vocal Effort	Correction to Voice Spectrum
Normal	0 dB
Raised	+ 6
Loud	+ 10
Shouting	+ 20

FIGURE 16.6 Comparison of the Relative A-Weighted Levels in the Frontal Vertical and Horizontal Planes for Male and Female Talkers (Chu and Warnock, 2002)



Outside noise sources should also be controlled in open-plan offices. HVAC noise should be limited to NC 35 in open-office areas and to NC 30 in private offices. HVAC noise is not helpful as a masking source, since its spectrum is rarely the proper shape and, in any case, it is not adjustable or uniformly distributed throughout the space. Noisy office equipment should be located in separate rooms. Offices should be carpeted to reduce walking

and furniture noise. The use of intercoms, personal radios, pagers, speaker phones, and other extraneous noise sources should be discouraged. Telephone rings should be set to their minimum volume.

Partial Height Panels

Although source orientation and control are helpful, barriers are always necessary. Partial height barriers include walls, which extend up to or even beyond the acoustical tile ceiling, but not to the slab or roof above. Prefabricated furniture panels can be used as speech barriers, but their inherent transmission loss limits their ability to perform this function. All the potential transmission paths must be considered, namely over, under, around, and through the barrier.

The total direct-field attenuation of a panel is the composite of the paths in Fig. 16.2. First, the barrier must block the direct path between talker and listener. If the listener is within 12 m (40 ft) of the talker, as in Fig. 16.7, the barrier must overlap the line of sight by at least 0.3 m (1 ft) to be effective. At distances of less than 4.5 m (15 ft) the sound path should have to bend at least 90° or more, as in Fig. 16.8, to be sufficiently attenuated.

Panel height is also an important factor. Barrier attenuation for both over and around paths may be calculated using Maekawa's relationship from Eqs. 5.10 and 5.11. The attenuation depends on the relative heights of the talker, barrier, and listener, as well as the frequency, and approximate results are given in Fig. 16.9.

FIGURE 16.7 Determining the Line of Sight for Distant People (Chanaud, 1983)

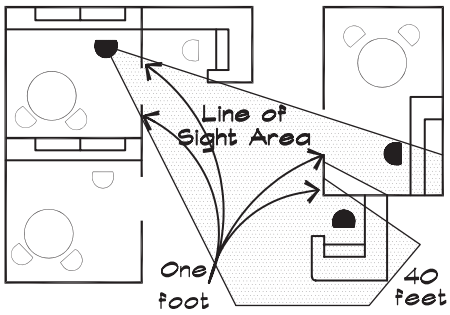


FIGURE 16.8 Determining the Line of Sight for Near People (Chanaud, 1983)

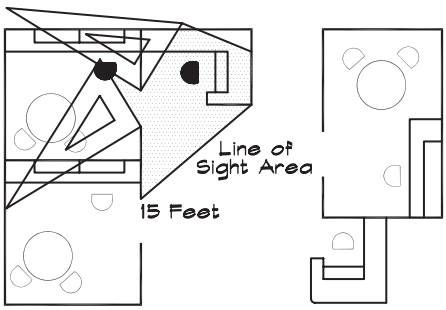


FIGURE 16.9 Influence of Panel Height on Speech Diffraction (Chanaud, 1983)

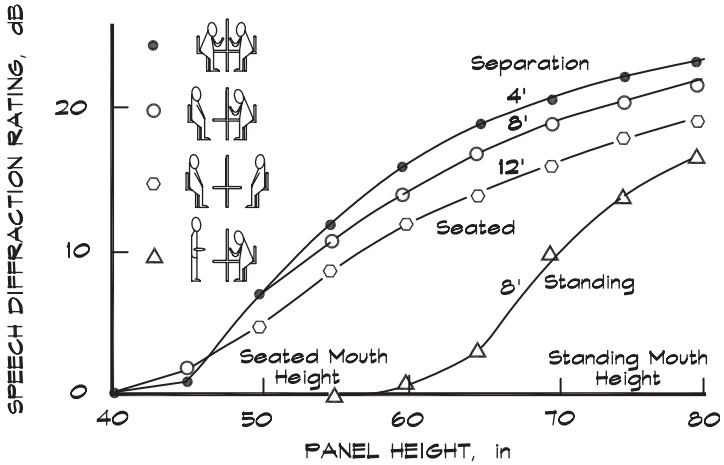
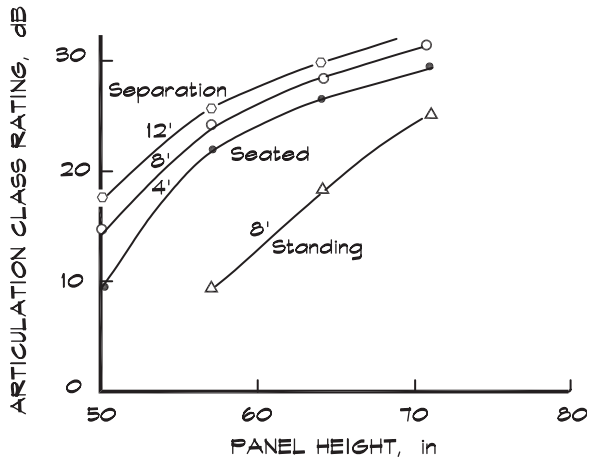


FIGURE 16.10 Influence of Panel Height and Distance on Speech Reduction (Chanaud, 1983)



When the distance and directivity attenuations are taken into account, the results are the speech reductions shown in Fig. 16.10. For seated occupants, at least a 1.8 m (70 in) panel height should be used, and if there are a significant number of standing conversations, a 2 m (80 in) panel height is recommended. In some situations it is desirable to be able to look around the room or over individual panels. In these cases panels may incorporate sections of glass without significant degradation.

In all cases it is also important that partial height barriers extend down to the floor to seal off the transmission path under the panel. A well-designed system includes blockage of this reflection. Panels should leave no more than a 25 mm (1 in) gap at the floor since carpet yields relatively low absorption values in the speech frequencies.

FIGURE 16.11 Influence of Panel Width on Speech Reduction (Chanaud, 1983)

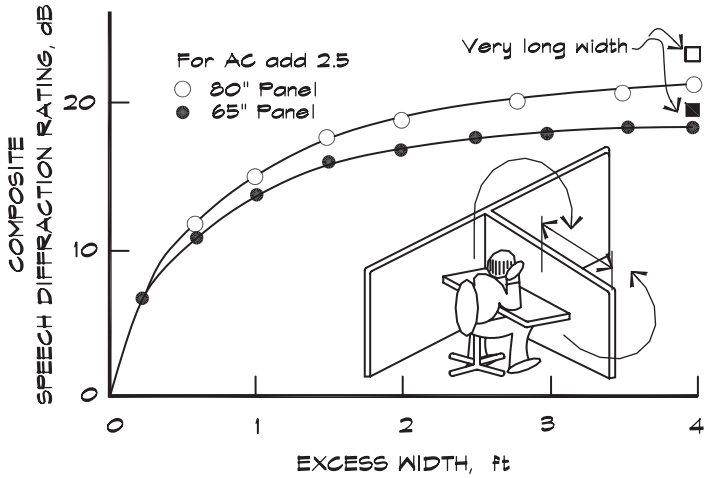
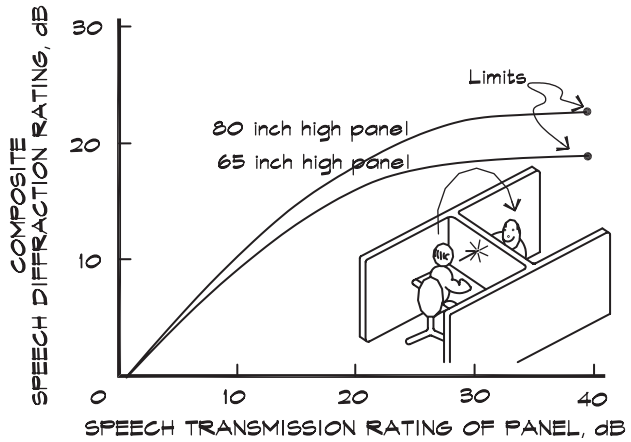


FIGURE 16.12 Influence of Transmission Loss on Speech Reduction (Chanaud, 1983)



Diffraction around the end of a panel must also be controlled. Figure 16.11 shows the effect of excess width on the speech diffraction rating of a side panel. Free-ended panel runs should be avoided and the ends capped with a right-angle piece. For 1.8 m (70 in) high panels the free end should extend at least 1.2 m (4 ft) beyond a seated worker. For 2 m (80 in) high barriers, no free-end conditions should be allowed. Free-ended panels are a problem, not only because of the diffraction, but also because they allow easy transmission of reflected sound.

Transmission of sound directly through a panel is also a concern. Since there is a natural limit to the effectiveness of a panel for diffraction losses, this sets a maximum on the necessary transmission loss. Figure 16.12 gives the diffraction limits for two heights of panel. The corresponding STC limits are 20 for 1.65 m (65") panels and 24 for 2 m (80") panels.

To achieve an STC rating of 20 to 24 a surface mass density of about 5 kg/sq m (1 lb/sq ft) is required. This can be accomplished using 3/8" plywood, 1/4" gypboard, or a 22 Ga steel sheet sandwiched between two absorbent fiberglass boards. Manufacturers of office furniture sometimes refer to this type of panel as "acoustical" or "high performance" in their literature, although the presence of these descriptors does not guarantee this rating. Manufacturers must be contacted for the actual STC ratings of a prospective material.

Absorptive and Reflective Surfaces

Absorptive surfaces should be used in locations where they will prevent reflected sound from flanking the main barrier, including ceilings, rear, and sometimes side walls. The sound reflecting from a surface is attenuated by an amount given by

$$\Delta L_s = 10 \log (1 - \alpha_s) \tag{16.13}$$

where α_s is the specular absorption coefficient.

Although the NRC ratings are useful for a general discussion of the properties of a given panel, if calculations are to be undertaken more detailed data are necessary. Random-incidence absorption coefficients are measured in third-octave bands, although they are published in octaves. Due to the details of the testing process, values of these coefficients sometimes exceed 1. In the case of ceilings and other reflecting surfaces the specular absorption coefficients are of more interest but are rarely available. Chanaud (2000) has developed an empirical relationship relating the diffuse (α) and specular (α_s) absorption coefficients for open-office calculations. He suggests

$$\begin{aligned} \alpha_s &\cong \alpha \quad \text{for } \alpha < 0.5 \\ \alpha_s &\cong .092 + .82\,\alpha \quad \text{for } 0.5 < \alpha \leq 1.1 \\ \alpha_s &\cong 0.999 \quad \text{for } \alpha > 1.1 \end{aligned} \tag{16.14}$$

Since most reflections between workstations are within 20° to 30° of the normal, there is little change in absorption with angle, as we saw in Fig. 7.19. When losses from reflections are combined with the directivity and the distance attenuation, we obtain the speech reduction for a reflected path

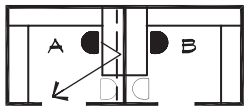
$$SR = \Delta L_\theta + \Delta L_s + 20 \log (D_s) \tag{16.15}$$

where D_s is the total reflected path length in meters.

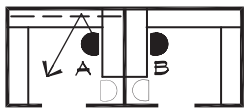
Figure 16.13 shows several examples of absorptive panel placement. If the separating panel, located directly along the line of sight between the source and receiver, is absorptive, its first reflection does not go toward the receiver, so the absorption is not in the most beneficial position. A side-panel reflection may not go directly to the receiver and so here, too, absorption may not be effective. An absorbing ceiling and rear panels decrease the sound transmission reflecting to the receiver and are effective. We therefore arrive at the conclusion that we place absorptive panels overhead and in the direction opposite the person to be given the privacy. Where there are people in all directions the application of this rule will result in several panels being covered with soft material.

Absorption should also be placed where a reflection would create a flanking path around the barrier panel. Reflections can occur from other panels, or from the walls and windows of

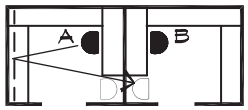
FIGURE 16.13 Absorbing Panels in Workstations (Chanaud, 1983)



Case 1 Separating panel is absorptive. B gets no benefit.



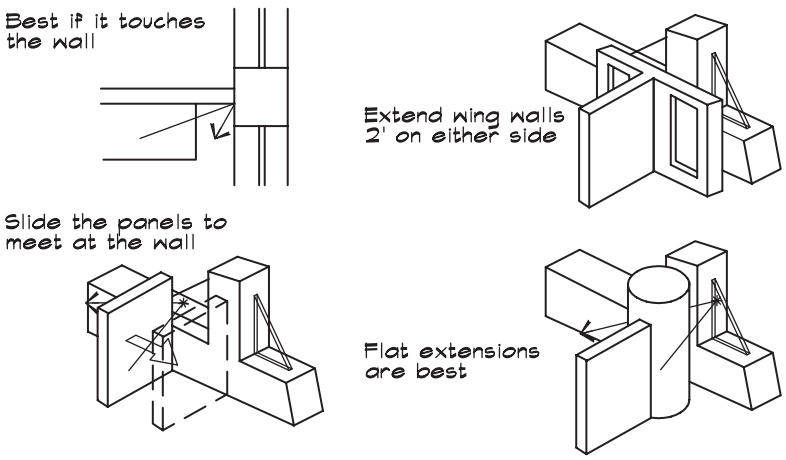
Case 2 Side panel is absorptive. Again B gets no benefit.



Case 3 Rear panel is absorptive. B gets some benefit.

Panel Type	Typical NRC	Reflection Attenuation
Hard Surface	0.05	0 dB
Acoustical	0.80	13 dB
Sound Screens	0.90	15 dB

FIGURE 16.14 Open Plan Panel to Window Connections



the building. If the reflecting surface is within 2 m (6.6 ft) of the talker the noise reduction coefficient (NRC) of the panel must be at least 0.9. For each additional 0.3 m (1 foot) of separation the NRC of the panel may be reduced by 0.05. Absorbent panels placed on potential reflecting surfaces must be long enough to cover all possible talker and listener mirror locations within their respective workstations. This is illustrated in Fig. 16.14.

Carpeted floors do little to improve speech privacy when panels extend all the way down to the floor. They do help control extraneous noise from sliding furniture, walking, and general reverberant noise at high frequencies. Their use is recommended in open-office areas for these reasons; however, they do not have to be chosen based on their NRC ratings.

Open-Plan Ceilings

The ceiling is a potential reflecting surface, which must be treated in open offices. The choice of ceiling materials was once limited to plaster, gypsum board, and mineral or fiberglass tile. Recently these traditional materials have been supplemented with many additional products including perforated and linear metal shapes, wire mesh, and sintered metal products. The traditional acoustical tile products have become more interesting visually and may be wrapped with cloth, perforated vinyl, or other porous materials.

An absorptive ceiling tile consists of three parts: backing, core, and facing. The most commonly available combinations are listed in Table 16.6. For open-plan offices, ceilings must be highly absorptive. Fiberglass tiles with a cloth facing have the highest ratings. A fiberglass tile having a 19 mm (3/4") thickness will have an NRC rating of about 0.90 and at a 38 mm (1.5") thickness its rating can sometimes exceed 1.0. Mineral-tile ceilings have NRC ratings, which range from 0.55 to 0.65 depending on thickness and surface treatment, whereas a gypboard ceiling has an NRC of about 0.05. Metal-foil backings should be used where the ceiling plenum is also a return air conduit. Table 16.7 (Chanaud, 1983) gives the speech reduction rating or Articulation Class for various ceiling absorption coefficients at an 8-foot talker-to-listener separation distance.

A comparison with the privacy ratings in Table 16.4 reveals that Normal Privacy will not be achieved if an acoustically hard ceiling is present. Transitional Privacy can be achieved with a mineral tile ceiling, but Normal Privacy can only be achieved by using a fiberglass ceiling tile in conjunction with 2 m (80 inch) high panels with masking. Figure 16.15 shows the variation of Articulation Index with ceiling material in the presence of masking sound.

The ceiling height can also have an effect on the degree of privacy. As the ceiling height increases the distance loss for reflected sounds becomes greater. Since speech reduction is a combination of distance attenuation and the absorption of the ceiling panels, a higher ceiling may allow the selection of a less expensive ceiling material. Table 16.8 (Chanaud, 1983) shows a comparison of the speech reduction rating for various ceiling heights.

A mineral tile ceiling must be 4.6 m (15 ft) high if it is to result in the same attenuation as a fiberglass tile ceiling at 2.6 m (8.5 ft). There is little advantage to be gained from raising

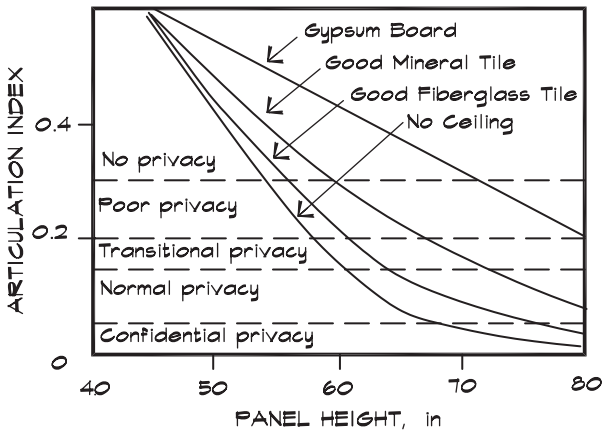
TABLE 16.6 Ceiling Tile Combinations

Backing	Core	Facing
None	Mineral Tile	Glass Cloth
Metal Foil	Fiberglass	Vinyl
		Perforated Vinyl
		Painted Cloth
		Metal Pan

TABLE 16.7 Speech Reduction Caused by Ceiling Absorption (Chanaud, 1983) 2.5 m (8 Foot) Separation

Ceiling Type	NRC	AC (dB)
1.65 m (65 Inch) Panels		
None		27
Ideal Material	1.15	26.5
Good Fiberglass Tile	1.00	25
Standard Mineral Tile	0.65	21.5
Gypsum Board	0.05	17
2 m (80 Inch) Panels		
None		33
Ideal Material	1.15	31.5
Good Fiberglass Tile	1.00	28
Standard Mineral Tile	0.65	22.5
Gypsum Board	0.05	17

FIGURE 16.15 Factors Influencing Speech Privacy with Masking Sound (Chanaud, 1983)



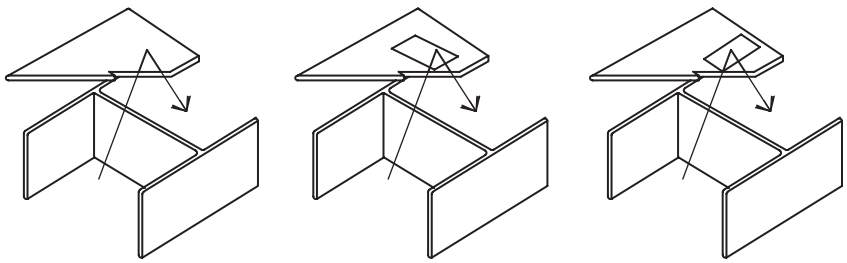
a ceiling, if it is already highly absorbent. Conversely, if a ceiling is quite high, there is little privacy improvement to be gained by changing to a more absorbent material.

Reflective light fixtures and metallic diffusers should not be located where a reflection will cause a degradation of the ceiling absorption. A flat reflective light fixture, such as those in Fig. 16.16, located above a separating partition, can produce a reflected level equal to the effect of changing the entire ceiling from fiberglass to mineral tile.

TABLE 16.8 Influence of Ceiling Height on Speech Reduction (Chanaud, 1983) 1.3 m (4 foot) Separation—1.35 m (65 Inch) Panels

Ceiling Height (m/ft)	AC (dB) Fiberglass	Mineral Tile
2.4 (8)	23	19
2.6 (8.5)	23	19.5
2.7 (9)	23.5	20
3.0 (10)	23.5	21
3.7 (12)	24	22
4.6 (15)	24	23
6.1 (20)	24.5	23.5
30 (100)	24.5	24.5

FIGURE 16.16 Influence of Light Fixtures on Ceiling Reflections (Chanaud, 1983)



Privacy for 65 inch panels on three sides, fiberglass ceiling, and masking sound

Condition	Privacy Index	Degree of Privacy
No light fixture	84	Normal
Fixture in short direction	69	Poor
Fixture in long direction	67	Poor

Masking Sound

The addition of background or masking sound to an open-office environment is a critical component to obtaining overall privacy in the workplace. When properly done, masking sound raises the noise level in an unobtrusive way that increases privacy without being noticed by the occupants. It can be used in open-office plans with partial-height barriers and in private offices with walls that are either full or partial height. Even when a private office with a full-height wall is adjacent to an open-office area, it is still a good idea to introduce masking sound into the private office. This smoothes the transition between the spaces and makes the noise in the open area much less noticeable. Masking sound is produced by using an electronic pink noise generator and a filter that results in a spectrum that falls off at about

FIGURE 16.17 Influence of Masking Sound on Speech Privacy (Chanaud, 1983)

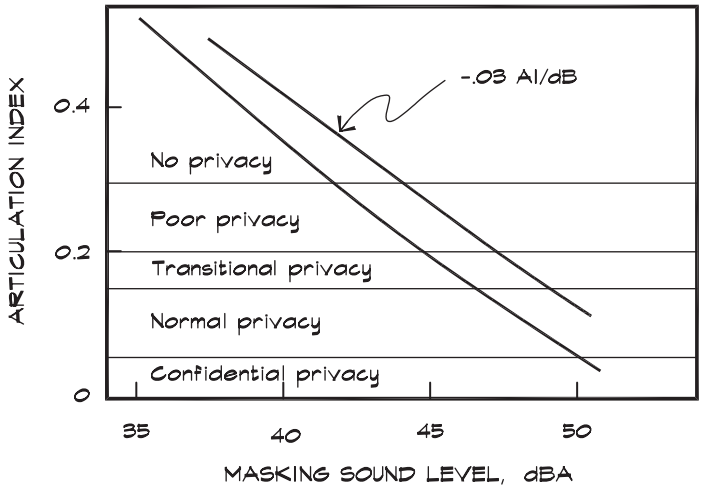
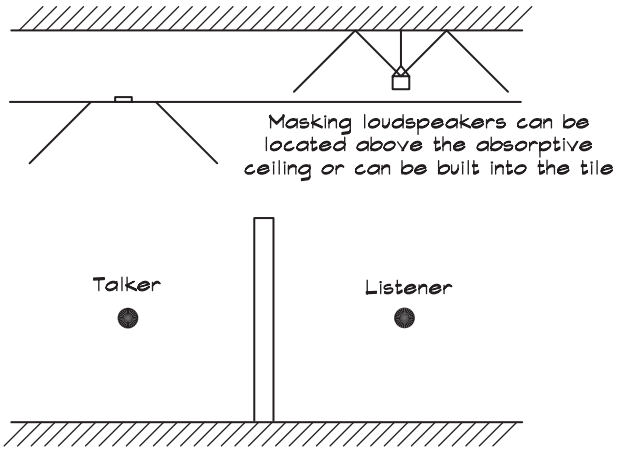


FIGURE 16.18 Masking Sound Loudspeaker Locations



5 to 6 dB per octave at the receiver. A possible spectrum is shown in Table 16.3; however, there is no single ideal curve. The overall level is set between 43 and 49 dBA. Though this seems like a relatively narrow range, below 43 dBA masking sound has little effect, and above 49 dBA it is an annoyance. Most systems end up being adjusted to somewhere around 47 dBA in open offices and 44 dBA in closed offices. The effect of masking sound on privacy for a typical workstation is shown in Fig. 16.17.

Loudspeakers can be located above the ceiling tiles, out of sight, and pointed up to increase their effective distance and widen their coverage pattern. They may also be built into the tiles themselves, which decreases the installation cost. Figure 16.18 gives two examples. The introduction of masking sound above a workstation provides privacy for a listener located within the sound field of the loudspeaker. It does not provide masking of the sound made by

a talker located there. Hence, loudspeakers must be distributed through all areas where there are potential receivers. As an employee walks around the workspace, the sound level should not change drastically. He should not be aware of the presence of masking, so any changes in level or spectrum should be gradual. Work areas should be zoned so that levels may be adjusted according to the needs of a particular area. Masking systems can be controlled using a timer that varies the level in a given zone during the workday; however, the system should not be switched off during working hours.

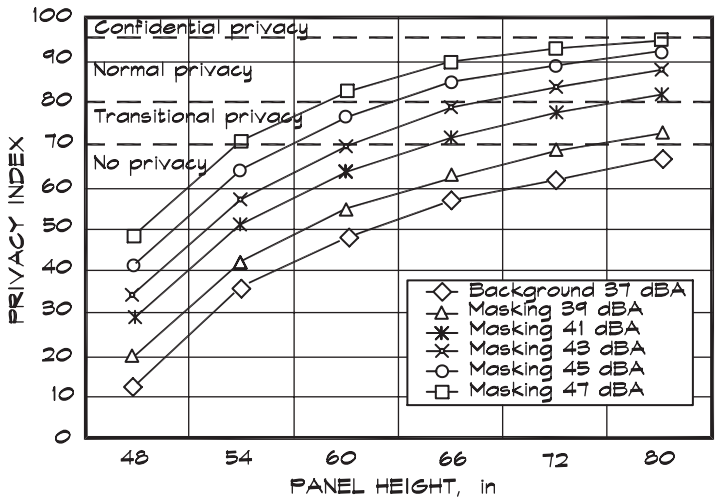
Music and paging may be mixed into the system, but the masking sound should only be muted for emergency pages. Music is not a good source of masking sound by itself, but it can provide a pleasant background environment for certain tasks. Music is not appropriate for areas where difficult tasks are performed. When music is introduced the masking sound must remain on.

Degrees of Privacy

We now have in place the tools to provide various degrees of privacy in an open (or private) office environment. As was emphasized previously, all component parts must be in place to achieve privacy. Figure 16.19 shows the influence of panel height and masking sound on levels of speech privacy. Clearly neither high panels nor masking sound alone can yield good privacy.

A successful open-office design must include four key elements and fails if any of the four is missing. The elements are: 1) partial height barriers at least 1.65 m (65 in) high, having sufficient transmission loss, provided by a 3/8" plywood or other interior panel; 2) absorptive material typically 25 mm (1 in) fiberglass panels on the reflecting walls or additional barriers to prevent flanking; 3) a highly absorbent ceiling (NRC > .85); and 4) an electronic sound-masking system with loudspeakers located above the acoustical tile ceiling set to emit a particular spectrum in the range of 45 to 49 dBA.

FIGURE 16.19 Privacy Index with Ceiling NRC = 0.9 and a 6-Foot Separation (Chanaud, 1983)



16.2 SPEECH PRIVACY IN CLOSED OFFICES

Private Offices

In a closed office the assessment of privacy is expressed in terms of STC values between adjoining spaces. Closed offices have the advantage of providing privacy throughout the enclosed space for a standing or seated occupant. The disadvantage is that the normal background level is lower, even when masking is included, and conversations may take place at raised voice levels even when confidential matters are being discussed. Specialized areas such as psychologists' offices, spaces used for conflict resolution, rooms with audio visual systems, lecture rooms, and classrooms, are all likely to have a need for extra isolation and should be identified as part of the initial planning process.

The analysis of sound transmission for private offices is similar to that for open offices. In critical locations detailed calculations using voice spectra, transmission loss, and background noise levels should be done. Figure 16.20 shows the calculated Articulation Index for various FSTC values at three voice levels, and Table 16.9 includes the corresponding sound attenuation requirements for various degrees of privacy. Chanaud (1983) recommends that the FSTC of a given structural component be 6 dB greater than the desired composite FSTC.

Full-Height Walls

The traditional approach to closed-office privacy is to use full-height walls and weatherstripped solid core doors. When the walls are full height, the total FSTC value is obtained. Sketches of several wall configurations are given in Fig. 16.21, along with estimates of their composite FSTC values. When the wall does not extend full height on both sides various plenum barrier materials can be used to make up the difference.

For Confidential Privacy with a normal voice talker, a good separation wall between rooms is a single 3 5/8" (90 mm) metal stud with 5/8" (16 mm) drywall each side and R-11 batt in the airspace. This wall rates an STC 44 (FSTC 39), and for voice provides a calculated noise reduction of approximately 42 dB. In terms of a rough calculation, a 60 dBA

FIGURE 16.20 Required FSTC for Speech Privacy for Various Voice Levels (Chanaud, 1983)

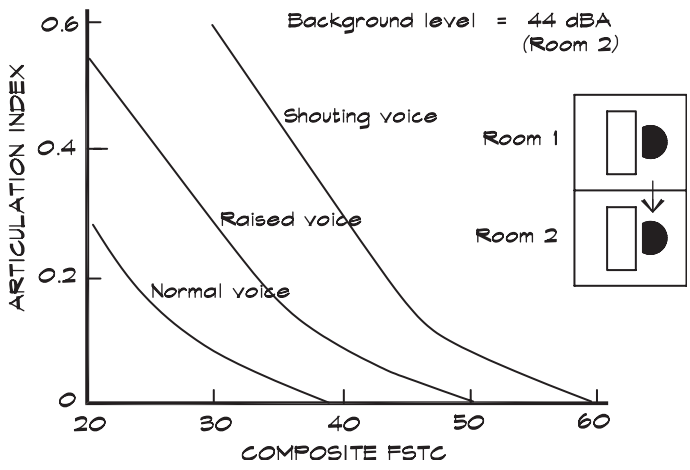


TABLE 16.9 Required Composite and Component FSTC for Various Voice Levels (Chanaud, 1983)

Background Noise = 44 dBA

Voice Level	Minimum Composite FSTC for Normal Privacy	Minimum Composite FSTC for Confidential Privacy
Normal	26	32
Raised	36	42
Shouting	46	52

Voice Level	Recommended Component FSTC for Normal Privacy	Recommended Component FSTC for Confidential Privacy
Normal	32	38
Raised	42	48
Shouting	52	58

sound pressure level due to a normal voice in one room would generate about 18 dBA sound pressure level in the adjacent room. If the masking level in the receiving space were 35 dBA, we could achieve Confidential Privacy since the signal-to-noise ratio is less than –12 dB. This describes a typical private office with a background level of about NC 30 although the NC spectrum is not identical with the masking spectrum.

Plenum Flanking

The choice of an appropriate wall type is not the end of the design exercise, since there are many other routes that the sound can take from one room to another. If the wall does not extend from slab to slab, then the sound can travel over the top of the wall, passing through the plenum above the T-bar ceiling on either side. Since the mass of acoustical tile is quite low, it does little to attenuate noise from passing through it. Blazier (1981) has published measured values of the noise reduction of an acoustical tile ceiling, which are shown in Fig. 13.15. Even if these are doubled and increased by 6 dB, following Eq. 9.50 they still represent a significant degradation in transmission loss relative to the wall performance. Grille openings for the return air plenum can further degrade these ratings.

Figure 16.22 gives several examples of FSTC values for partial-height walls flanked by plenums. In this figure there are substantial differences in FSTC values depending on the ceiling material. Manufacturers of acoustical ceilings have sought to develop so-called high transmission loss tiles, which combine both absorption and transmission loss into a single product. These tiles have a modest absorption coefficient, between 0.5 to 0.7, and a relatively low effective STC rating (as measured with a wall in the STC 30 to 35 range). Even though they are better than lighter acoustical tiles, they can be compromised by openings for return air grilles.

FIGURE 16.21 Full Height Separation Wall Configurations

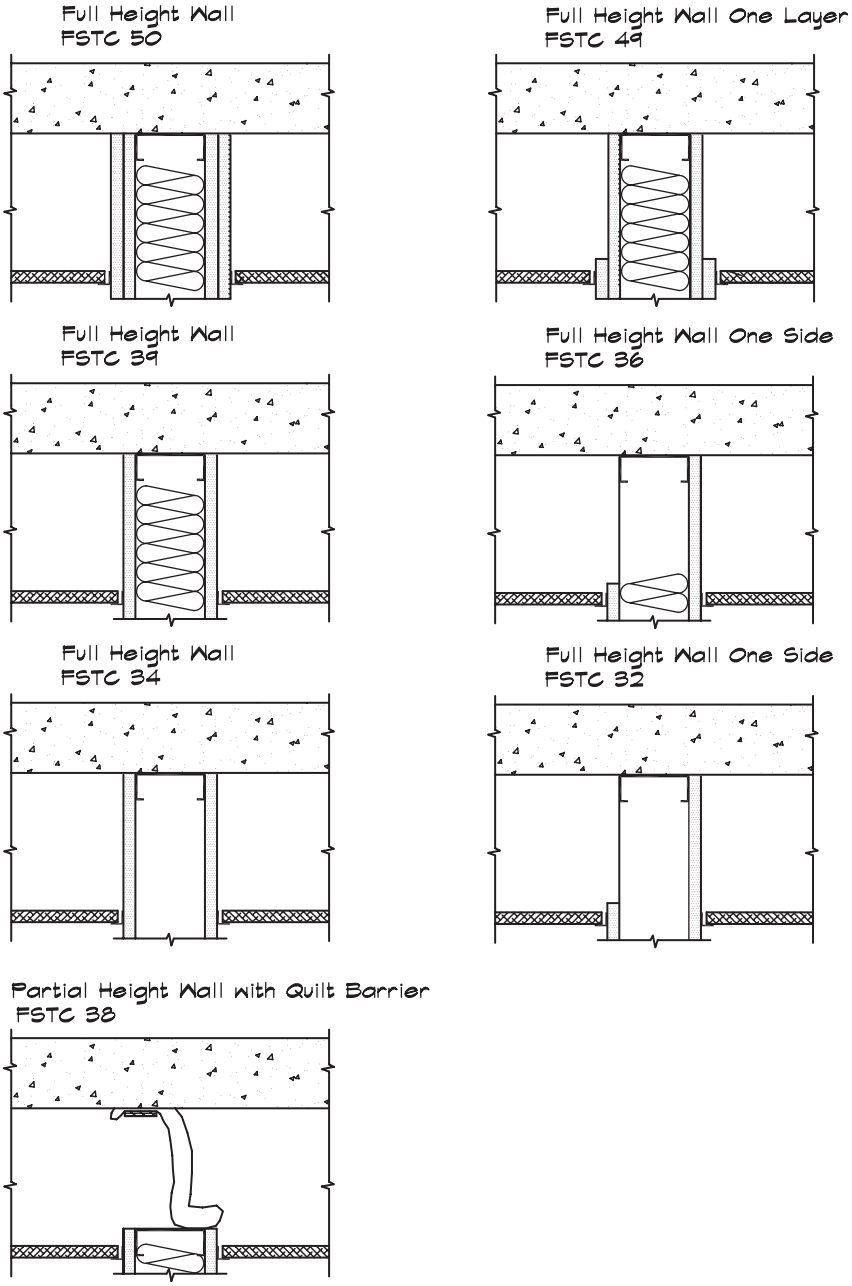


FIGURE 16.22 Wall Plenums with Estimated FSTC Values

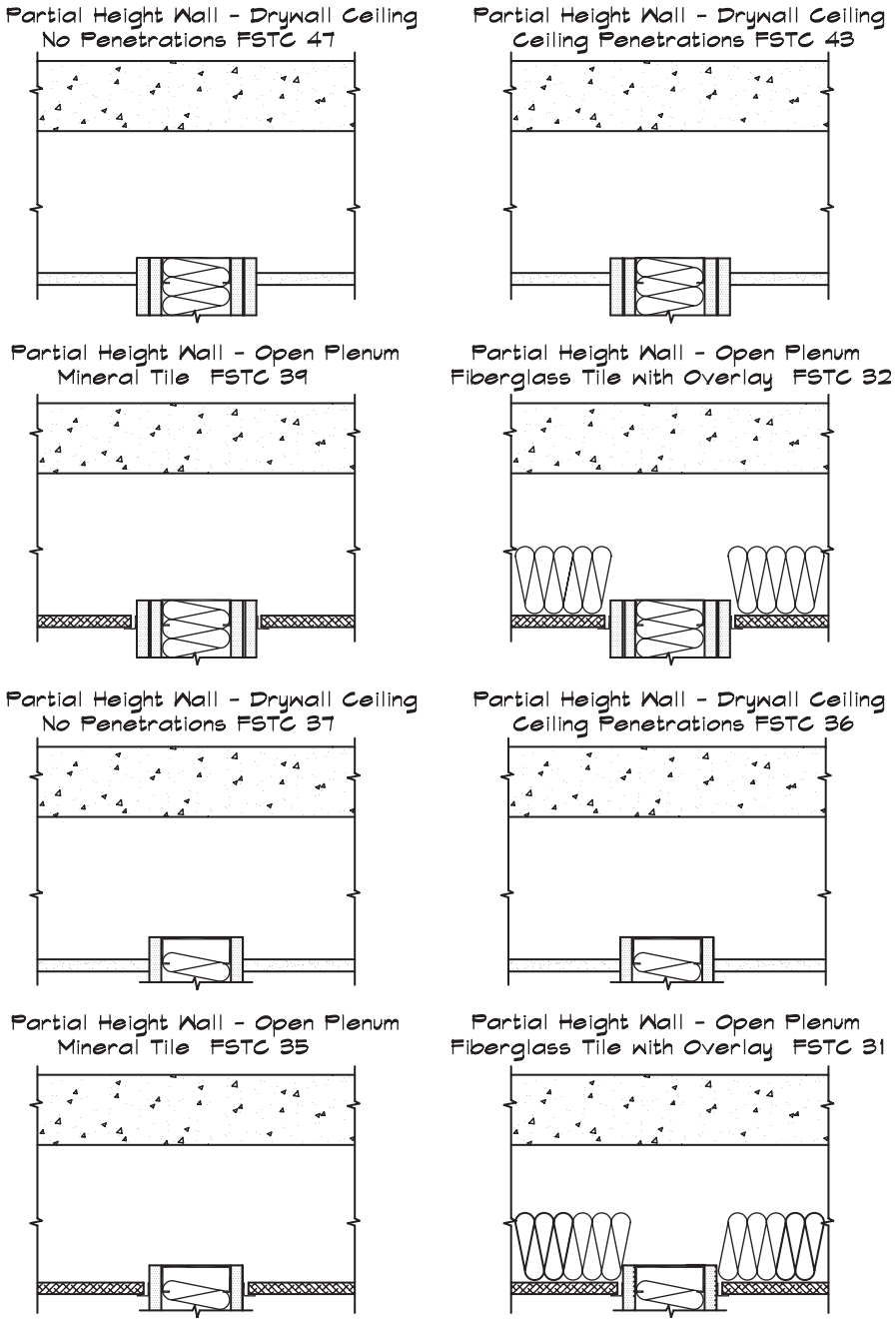
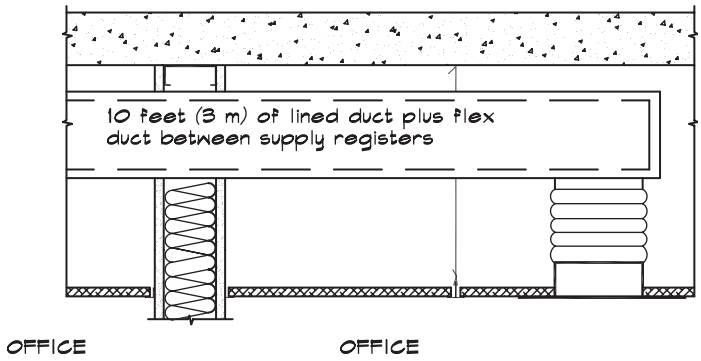


FIGURE 16.23 Flanking through Ductwork



The addition of batt insulation on top of the T-bar ceiling can provide some improvement for fiberglass tile ceilings, which have relatively low transmission loss values. Batt is not appropriate in return air plenums and instead a duct liner should be used.

Duct Flanking

HVAC ducts can also serve as a conduit for sound transmission between adjacent rooms, particularly if they are unlined. If an unlined duct directly connects diffusers in adjacent offices, sound can propagate along the duct and be heard clearly in the receiving room. To calculate this effect, the duct attenuation becomes the transmission loss and the cross sectional area of the duct becomes the transmitting area in Eq. 10.5. Where the supply or return air is ducted directly between sensitive spaces, a 3 foot (1 m) silencer or a total of 10 feet (3 m) of duct, lined with 1" (25 mm) duct liner, is required. If the ceilings are drywall, two 4-foot (1.2 m) lengths of flex duct can be substituted for the lined duct. A detail is shown in Fig. 16.23.

Where walls are constructed full height to the floor or roof above, they cut off the free circulation of air from the plenum to the rest to the return-air system. An opening in one of the walls, with 5 ft (1.5 m) of lined duct penetrating the plenum wall, is normally sufficient to isolate the two spaces and provide a low-pressure path for the air to return. Where the return air can be transported in the space above the corridors in an office complex, it provides for additional separation by forcing the noise to traverse two segments of lined duct. Figure 16.24 shows a typical arrangement.

Exterior Curtain Walls

Interior partitions separating adjacent offices will eventually meet an exterior wall. Ideally, the interior wall should join the exterior surface at an area of solid construction, so that acoustical isolation is maintained. When the walls intersect at a window mullion or at the glazing there can be a flanking path at the junction, particularly if the window is double glazed. Exterior window mullions are constructed from thin (typically 0.090" or 2 mm), hollow, rectangular aluminum extrusions with very little mass. Where the junction with an interior wall falls on the mullion the wall can be attached as shown in Fig. 16.25. For additional isolation an extra piece of drywall or heavy sheet metal can be attached to each side.

FIGURE 16.24 Return Air Plenum at a Full Height Wall

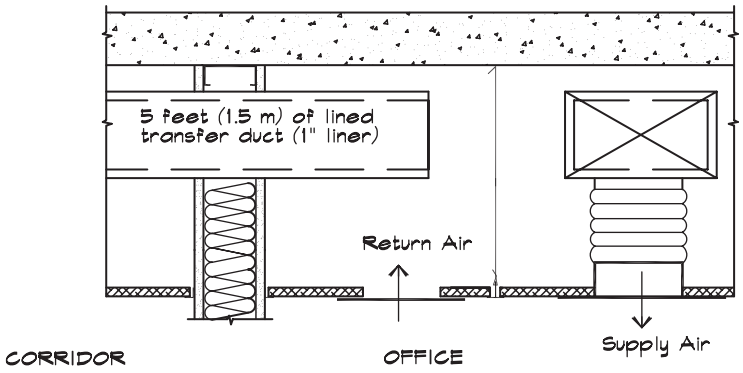
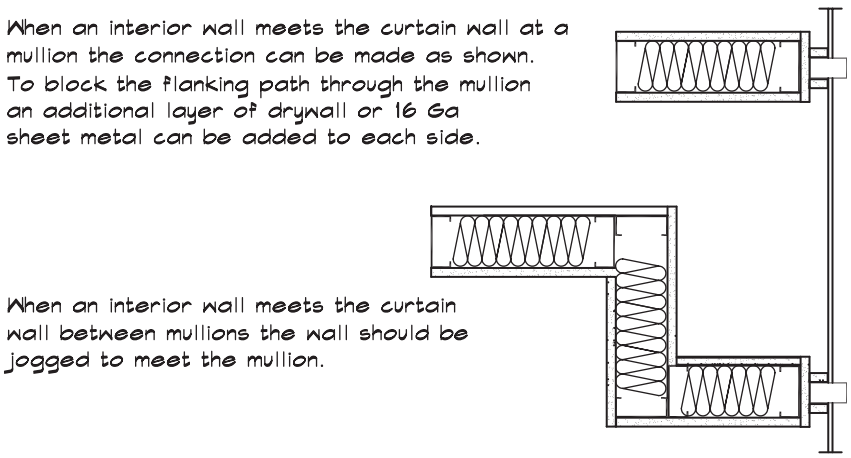


FIGURE 16.25 Flanking around Walls at Mullions

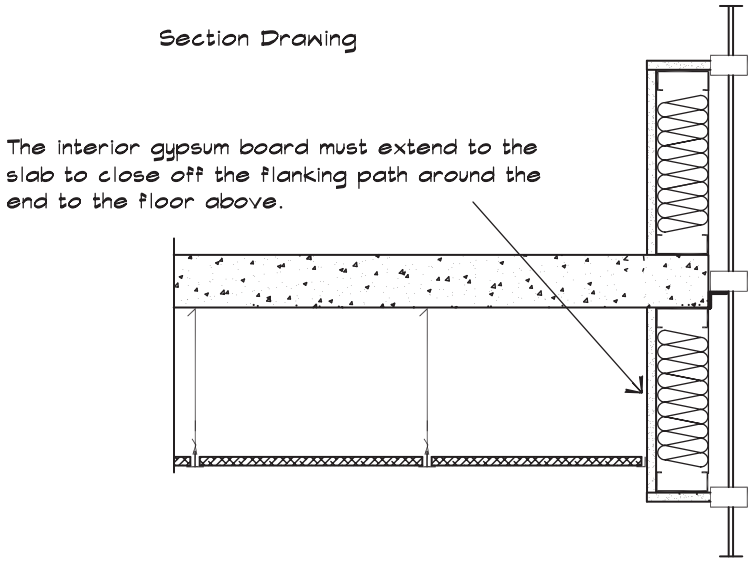


This thickens the appearance of the mullion as seen from the outside but provides much better isolation than stopping the wall short of the inner surface of the mullion.

Where the end of the wall falls between two mullions the wall should be jogged over until it falls directly on the mullion. Occasionally there is a continuous air bar around the outside of the building that penetrates the dividing wall. This can severely compromise acoustical isolation between spaces and must be closed off at the wall. Lined sections of duct can carry the air through the wall.

Not infrequently in high-rise construction the exterior curtain walls are supported from the edge of the floor slab and a gap is left between the slab and the glazing. Where there is spandrel glass and an interior wall, it is a good idea to continue the wall up past the ceiling as shown in Fig. 16.26. In other cases sheet-metal plates can be installed to bridge the gap between the slab and horizontal mullions both above and below the slab. The airspace should be filled with safing.

FIGURE 16.26 Flanking around Slabs at Curtain Walls



Flanking paths sometimes occur at column penetrations of slabs, particularly where the lower floor space has an open ceiling. In each of these conditions the openings around the column must be closed off with a material that has a sound transmission loss equivalent to the floor system.

Divisible Rooms

Moveable walls, used as dividers between meeting rooms, are commercially available with STC ratings as high as 50 or more. To achieve these ratings the panels weigh as much as 10 lbs/sq ft (49 kg/sq m). They are supported from a structural framework above that must be sized to carry the load without undue deflection. Unless this is taken into consideration the moveable walls can drag on the floor when they are moved into place. Some deflection is inevitable, and the panels are positioned and jacked up so that they support their own weight from the floor below. Moveable walls are the same as permanent walls in that the plenum flanking path must be blocked off. Figure 16.27 shows a design. Any stem wall must also accommodate structural deflection due to the weight of the panel.

Masking in Closed Offices

Where walls do not extend to the structural deck above the T-bar ceiling, it is not possible to achieve Confidential Privacy without the addition of masking sound. An office separation might consist of a single metal-stud wall with single layers of 1/2" gypsum board extending up to or slightly above the T-bar and a mineral-tile ceiling. Return-air grilles should be baffled with a lined sheet-metal enclosure illustrated in Fig. 16.28, with the open sides facing away from the receiver room. Table 16.10 shows the resulting degrees of privacy with this configuration for several levels of masking sound.

FIGURE 16.27 Installation of Moveable Wall Partitions

A wall of equivalent transmission loss should be built above the moveable wall. Studs and batt not shown for clarity.

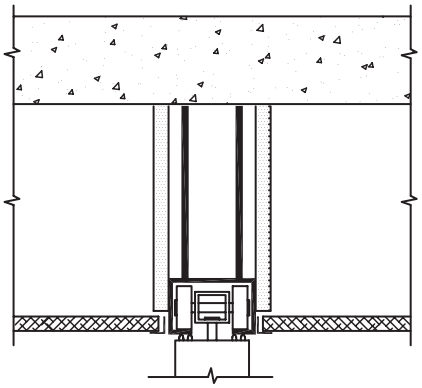


FIGURE 16.28 Return Air Boots

A sheet metal return air boot built over the return air grille. Orient the opening away from the receiver. Nominal dimensions 24" x 24" x 8" high.

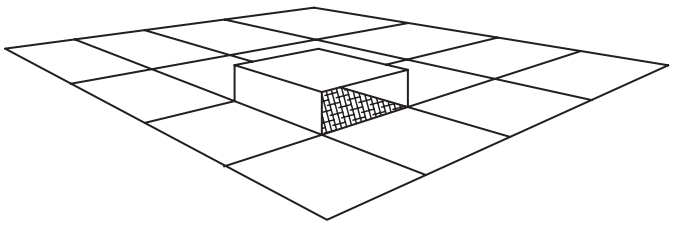


TABLE 16.10 Closed Office Speech Privacy for Mineral Tile Ceilings Normal Voice Levels (Chanaud, 1983)

	No Masking	Masking
Attenuation (NIC)	**33**	**33**
Masking Level (dBA)	**35**	**39**
Privacy Index	**89**	**95**
Degree of Privacy	**Normal**	**Confidential**

TABLE 16.11 Closed Office Speech Privacy for Fiberglass Tile Ceilings: Normal Voice Levels (Chanaud, 1983)

	Fiberglass Tile	Fiberglass with Overlay
Attenuation (NIC)	**27**	**33**
PI with 47 dBA	**> 95**	**> 95**
PI with 42 dBA	**91**	**> 95**
PI with 37 dBA	**83**	**91**

Note that in closed offices lower levels of masking sound are required to achieve Confidential Privacy than were necessary in open offices.

When fiberglass tile ceilings are present the low transmission loss values limit the attenuation between spaces. An overlay of 1/2" drywall on top of the tile can be helpful for improving the ceiling transmission loss. Table 16.11 summarizes the results of measurements for these conditions with booted return air grilles.

This solution is helpful in cases where the designer prefers the same style of ceiling tile in the open- and closed-office areas. Even with the drywall overlay the masking sound penetrates the ceiling, although the level must be increased. Loudspeakers should not be located near the return air grilles.

16.3 MECHANICAL EQUIPMENT

Mechanical systems in commercial spaces tend to be larger and more centralized than those in residential buildings. With larger units it becomes critical to address the vibration isolation aspects of the transmission problem and to enclose the mechanical spaces with heavy walls or buffer zones to protect the adjacent occupancies. Given the additional space, oft times these units are easier to treat, since there is room for silencers, lined plenums, and other conventional treatments.

System Layout

Architects and engineers can reduce the amount of treatment necessary to control mechanical noise in office spaces by shielding mechanical equipment rooms from sensitive spaces with intervening rooms such as bathrooms, storage rooms, and corridors. Bathrooms are particularly useful in this regard, since they include drywall or plaster ceilings, which can serve as return air plenums and provide additional space for the location of silencers.

Simply providing adequate space for mechanical equipment in a location that is isolated from potential receivers can be of great benefit. Many noise problems occur when this planning is not done. All too frequently air handlers or heat pumps are shoehorned into the available space in ceiling plenums above sensitive areas. Even when space is provided for the equipment, space is not left for a silencer, plenum, or other attenuating devices. This is often the case with down-shot air handlers, where the ducts emerge straight down from the underside of the unit, leaving little room for silencers and gradual turns. Where these devices are located above sensitive receptors it is very difficult to correct noise problems after installation.

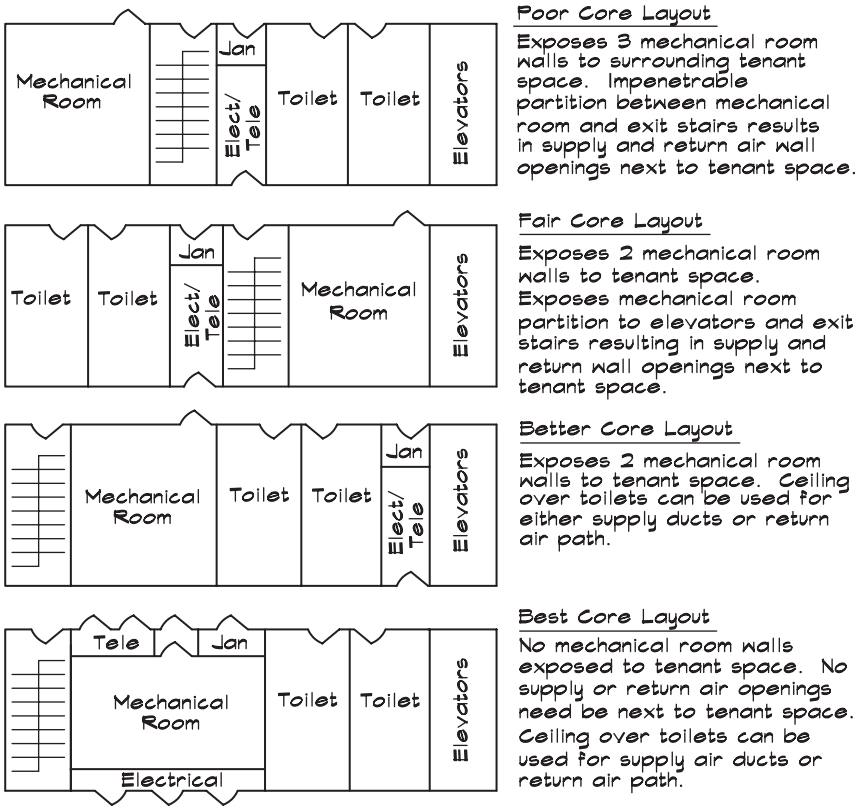
Mechanical Equipment Rooms

In a typical multistory office complex there is a central core where a mechanical equipment room is located. This is an excellent design configuration since it allows the necessary space for buffers, silencers, lined duct, or the other treatments. Several possible core layouts are shown in Fig. 16.29. The designs illustrate increasingly isolated arrangements with storage and acoustically benign equipment rooms acting as buffers.

From the central mechanical equipment room the supply air is fed into a duct loop serving the tenant spaces. The return air can be ducted or drawn from the ceiling plenum or from a plenum located above the central corridor. When the space is leased by a single tenant, there is often no reason for the corridor, and the lessor builds out the whole space to his liking. The full built-out condition is more difficult to treat acoustically than the subdivided condition since there is no buffering provided by the corridor.

In a central equipment room the air handler is located (Fig. 16.30) so that the supply ducts rise vertically to an elbow and pass horizontally through the walls of the mechanical equipment room to the duct loop. The air is returned through an opening in the mechanical equipment room wall and into the side of the air handler. Makeup air is supplied to the mechanical equipment room by means of a separate fan located on the roof.

FIGURE 16.29 Comparison of Various Building Core Layouts (after Schaffer, 1991)



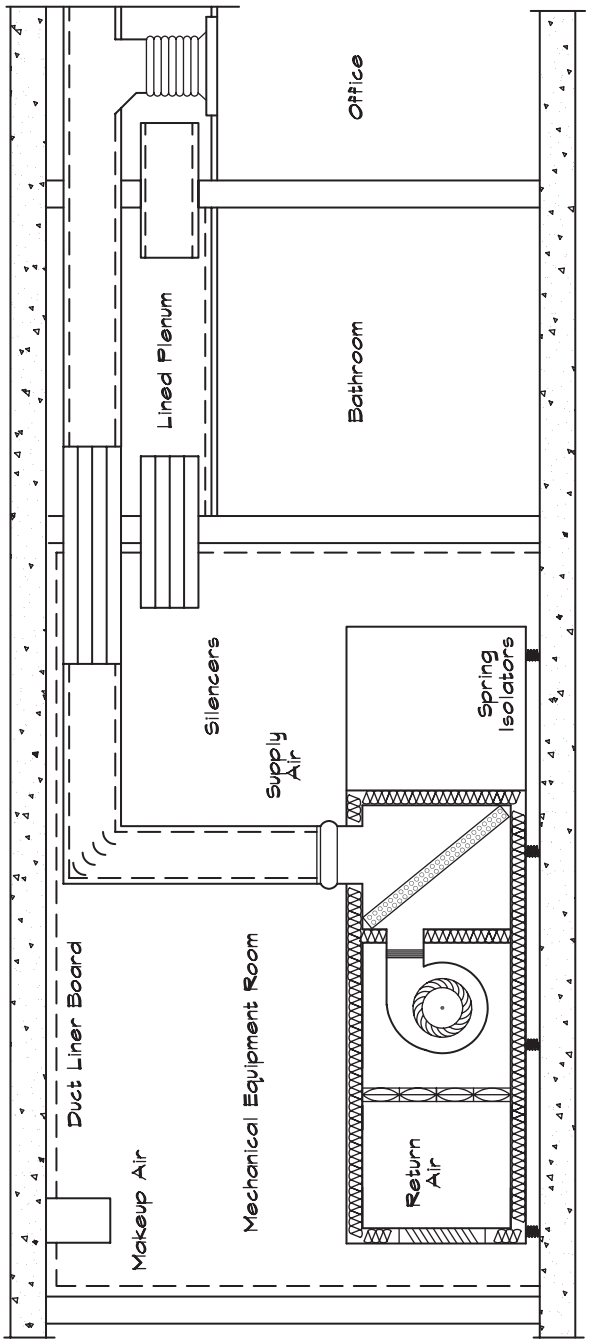


FIGURE 16.30 Central Core Air Handling Unit

Given the sound power level data on HVAC units at the specified operating point, a calculation can be carried out on both the supply side and return sides of the equipment as outlined in Chapt. 14. The final receiver sound pressure levels are compared with the interior noise level criteria to determine if any remedial steps need to be taken. In this case a silencer is added to the supply duct to provide the necessary attenuation.

On the return side the mechanical equipment room acts as an acoustical plenum with the return air opening on the air handler being the plenum entrance. If the radiating area of the air handler is not included as the entrance area, it is common to overestimate the attenuation due to the plenum effect. When the ceiling space above the bathroom is available it is useful to create an additional plenum and to use return-air silencers as well. Assuming that the exit from the plenum opens into the area above the acoustical tile ceiling we calculate the loss through the ceiling as an insertion loss to be subtracted from the sound power level before converting it to a sound pressure level using the appropriate room constant. An additional calculation of the sound radiated through the walls of the mechanical equipment room should also be done.

Roof-Mounted Air Handlers

In single story and low multistory projects packaged air handling equipment is frequently located on the roof. A packaged unit contains its own refrigerating condensing section along with the fan coil heat exchangers, filters, and one or more circulating fans. These units should be isolated with external spring isolation, on a series of open springs or curb rails, because internal isolation provided by the manufacturers may be inadequate and does not isolate casing vibration. The unit should be located over a stiff region of the roof having a deflection under the load of the equipment of no more than 1/6 to 1/8 of the spring isolator deflection. A housekeeping pad of at least 3" (75 mm) thickness should be provided for units up to 10 tons and 6" (150 mm) thickness above that. Pads must extend at least twice their thickness beyond the isolated equipment. Pads are strongly recommended, even when units are rail mounted on grade beams.

Side inlet and discharge are much preferred over down-shot units since they afford the space to install silencers. Down-shot units are often specified by mechanical engineers and contractors since they are self flashing, less prone to leaking, and less unsightly. Schaffer (1991) has published a series of figures showing various configurations of rooftop units, which are reproduced in Figs. 16.31 through 16.34.

Figure 16.31 shows a down-shot air handling unit with a large opening in the lightweight roof beneath. The opening allows the passage of noise through this space. To correct this condition, the unit must be picked up and mounted on rails to stiffen the structure and the opening must be boxed in from below with a metal stud wall with 2 layers of 5/8" (16 mm) drywall. Silencers are then installed at the points where the ducts penetrate the box. The details of the enclosure design must be determined by doing calculations using the sound power level emanating from the underside of the unit.

Figure 16.32 illustrates a configuration where some steps have been taken to control the noise. This arrangement is not sufficient above occupied office areas; however, with the addition of return-air silencers and a drywall box around the ductwork out to the silencers, it can be made to work in nonsensitive areas. Some manufacturers can provide high transmission loss ductwork of heavy (14–16 gauge) metal, which can help reduce breakout problems.

FIGURE 16.31 A Very Noisy Rooftop Air Handling Unit (Schaffer, 1991)

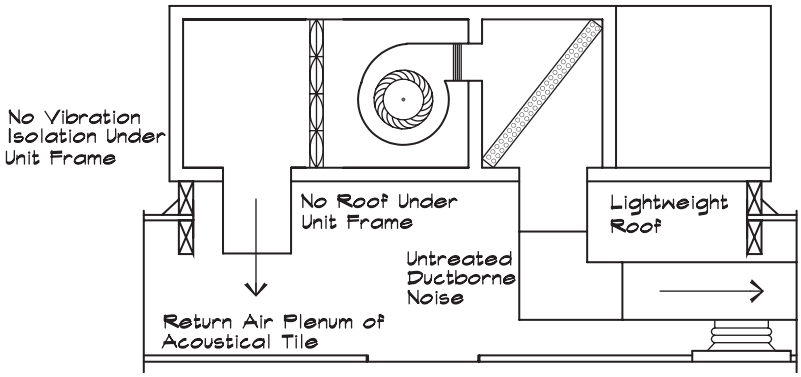


FIGURE 16.32 A Noisy Rooftop Air Handling Unit (Schaffer, 1991)

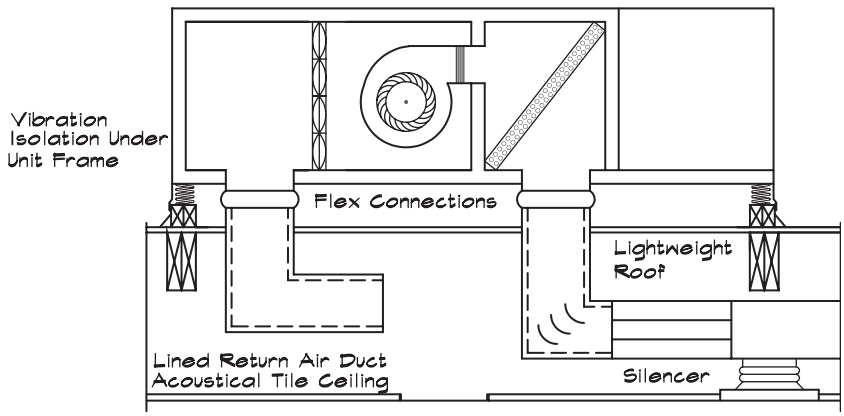


FIGURE 16.33 A Moderately Quiet Side Discharge Rooftop Air Handling Unit (Schaffer, 1991)

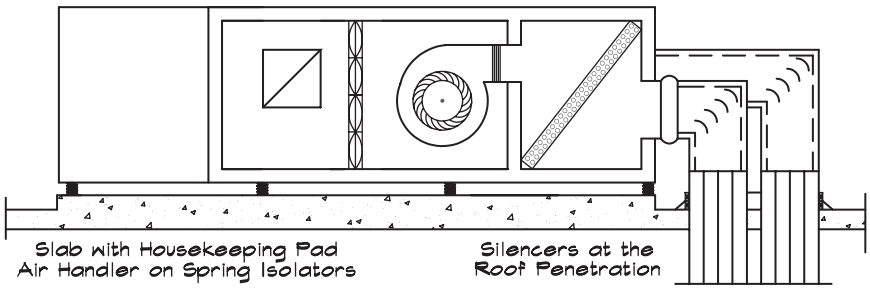


FIGURE 16.34 Quiet Rooftop Air Handling Unit (Schaffer, 1991)

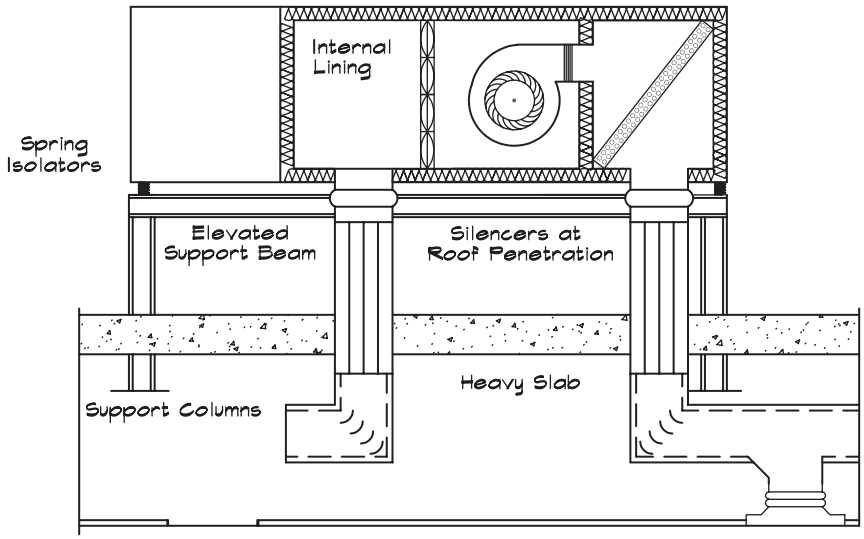


Figure 16.33 shows a side discharge configuration, which is the preferred arrangement, particularly for small installations. Where possible the ducts should penetrate the roof at a duct shaft or other nonsensitive area, since there is noise generated by the branch takeoffs. Where the building is located in a high exterior noise environment, the silencers can be located vertically at the roof penetration to reduce break-in problems.

Figure 16.34 illustrates a rooftop unit in which considerable effort has been expended to address the noise problems before the ducts penetrate the roof. The equipment support system is mounted on columns. The air handling unit is raised to allow the silencers to be installed vertically. Round ducts are used to control breakout and rumble.

Fan Coil and Heat Pump Units

In buildings such as hotels, where there is a need for temperature control in individual zones, a split system is used, which consists of an exterior condensing unit and cooling system, and an interior fan coil or heat pump unit located near the conditioned space. The fan in these units is the source of noise; as a general rule, fan coil units should not be located above or adjacent to occupied spaces when there is only an acoustical-tile or louvered-grille separation.

Fan coil units are available in two configurations, horizontal and vertical. The sound power levels vary with the fan capacity but in general a 3' (1 m) medium pressure drop silencer or equivalent is required on both the supply and return to reduce noise levels to an NC 30. Horizontal units can be located above a closet, which allows access through the closet ceiling for maintenance. Vertical units can also be placed in a closet with a silencer in the supply duct. The return can be ducted through a return air plenum in the ceiling or through a transfer duct silencer mounted to the back of a solid core or rated door (depending on the sound power level of the unit). Lined plenum returns located beneath vertical units are usually not sufficient to achieve an NC 30, but can achieve an NC 40 to 45 in the region adjacent to the return air grille. This may be sufficient for nonsensitive spaces but is not

FIGURE 16.35 Vertical Fan Coil Units

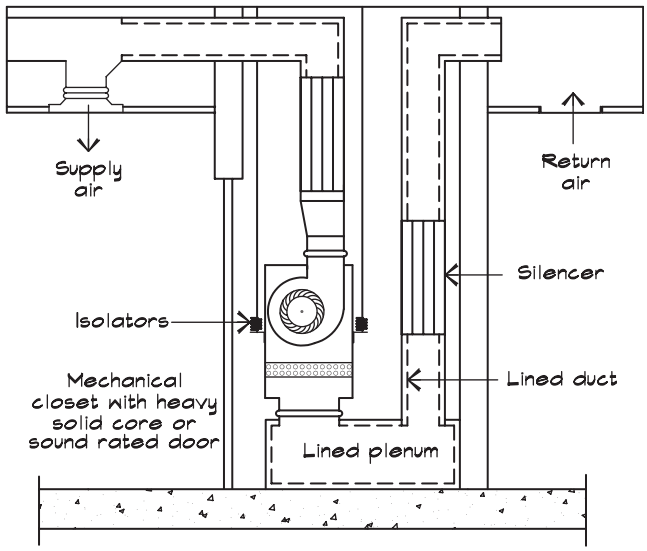
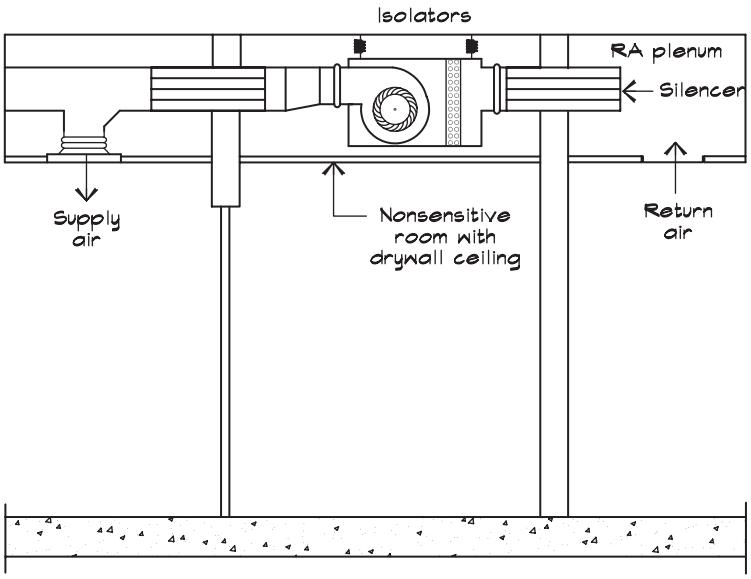


FIGURE 16.36 Horizontal Fan Coil Units

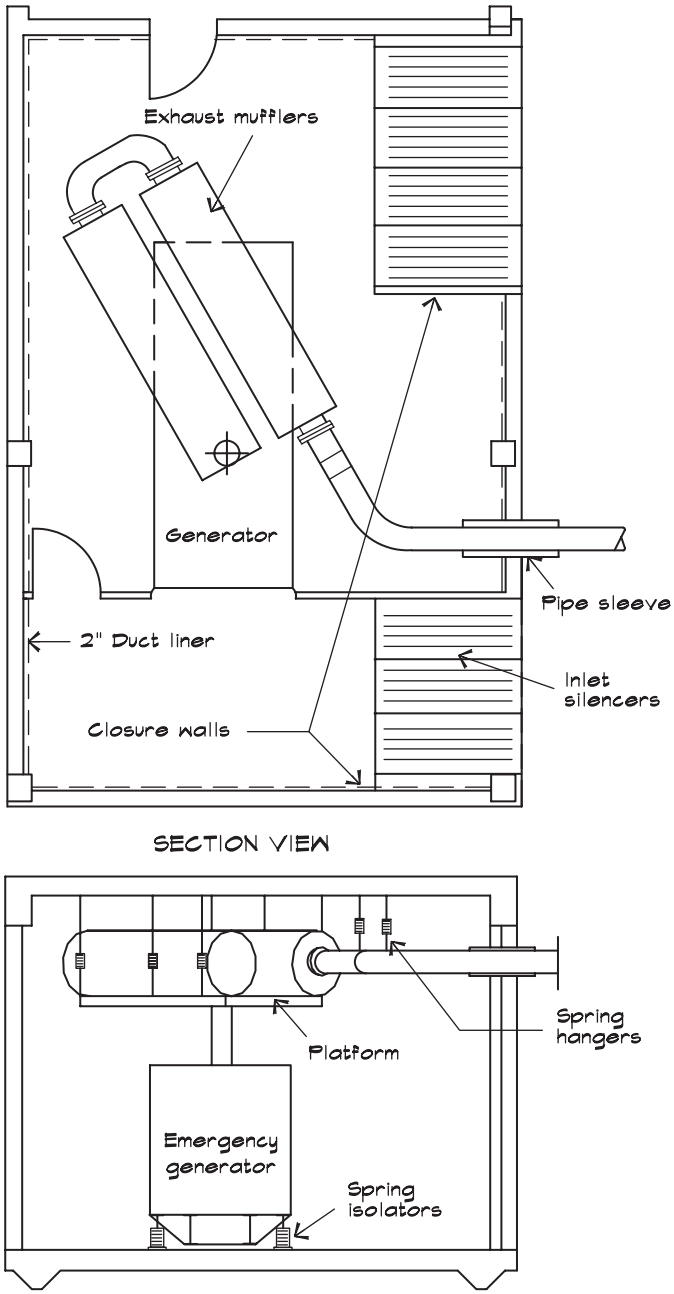


recommended for residential installations. Figures 16.35 and 16.36 illustrate these two types of fan coil installations.

Emergency Generators

Emergency generators are included in large buildings to supply power to selected equipment when the main power is lost. Although they are used infrequently, they must be tested

FIGURE 16.37 Plan of an Emergency Generator Room



periodically, typically an hour per month, so they need to be acoustically isolated. This requires treatment of both the exhaust and the inlet/cooling air. Inlet air is drawn in through the radiator by a fan and is used to cool the engine as well as to provide the combustion air. It circulates through the generator room and exhausts out again through silencers. The air intake has a large open area since the fan can accommodate only a small back pressure. The exhaust passes through one or more large mufflers. A design is shown in Fig. 16.37.

17 DESIGN OF ROOMS FOR SPEECH

17.1 GENERAL ACOUSTICAL REQUIREMENTS

General Considerations

Intelligibility depends on the masking effects of extraneous sounds on the speech we hear. Masking can be caused by noise from background sources or by reflections of the original spoken words. Speech combines the quick high-frequency sounds of consonants with the broader tones of the vowels. It is the recognition of consonants that correlates most closely with speech intelligibility, so the transmission of undistorted high-frequency information is critical. Figure 17.1 gives an illustration of a level versus time plot of the spoken word "back." Since the first part of the word is louder than the rest, its reverberant tail can mask the consonant ending.

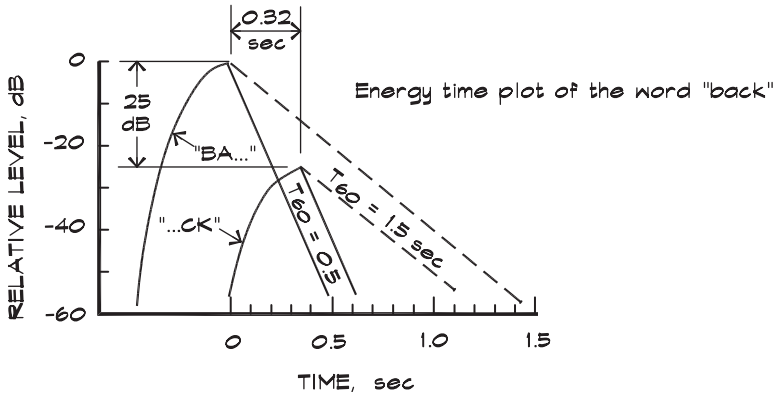
In the design of classrooms, conference rooms, and auditoria, the ability to understand speech is very important. The architectural components of these rooms—size, shape, surface orientation, and materials, as well as the background noise level—all influence intelligibility. There are several fundamental requirements in designing rooms for speech (Doelle, 1972), each of which contributes to achieving a high signal-to-noise level at the receiver:

1. There must be adequate loudness.
2. The sound level must be relatively uniform.
3. The reverberation characteristics of the room must be appropriate.
4. There must be a high signal-to-noise ratio.
5. Background noise levels must be low enough to not interfere with the listening environment.
6. The room must be free from acoustical defects such as long delayed reflections, flutter echoes, focusing, and resonance.

Adequate Loudness

For adequate loudness in a room, there must be a high direct field level. In unamplified spaces such as classrooms, the distance between the source and the receiver should be controlled.

FIGURE 17.1 An Illustration of the Effects of Reverberation on the Intelligibility of Speech (Everest, 1994)



Beyond 30 to 40 feet it is difficult to understand unreinforced speech, especially in a reverberant space. The volume per seat should be low, no more than 80 to 150 cu ft (2.3 to 4.3 cu m) per seat with an optimum value of 110 (3.1 cu m) (Doelle, 1972). By reducing the volume per seat, the loudness is increased and the reverberation time decreased for a given area of absorptive material. In general the smaller the seating capacity the larger the volume per seat can be within this range. Grazing attenuation should be controlled by raising the talker height and by sloping the floor. Beneficial reflections, preferably from overhead, should be designed in to help offset the effects of geometric spreading and grazing attenuation.

Making the audience seating area more circular minimizes the source-receiver distance. As the seating circle expands, there is a region beyond which the human voice cannot extend without physical or electronic reinforcement. These limits define the shape of a simple outdoor amphitheater and with the addition of walls and ceiling, they also contribute to the shape of a classroom or small lecture hall. For an auditorium, the semicircular seating area drives the shape toward a width that is greater than the depth as in Fig. 17.2. With a hard ceiling, the depth can be increased and the length-to-width ratio can exceed one. A balcony allows more of the audience to be seated close to the talker, as it brings the center of mass of the audience forward.

Nonacoustical considerations such as sight lines also influence the choice of room shape. The included angle between the outermost seats should be less than 140°, as shown in Fig. 17.3. This seating arrangement allows a clear view of the lecturer and writing boards. For projection screens the included angle should be limited to 125°, or about 60° from the screen centerline. Multiple angled screens can improve sight lines and reduce the effects of off-axis screen gain (loss).

Floor Slope

The floor of a large auditorium should be sloped to provide adequate sight lines. Good sight lines result in good listening conditions. Sight lines are set so that the audience can see the lowest point of interest on stage, called the arrival point of sight (APS), over the head of a person sitting in front of them. Even though it is theoretically desirable to design a theater with every-row clearance, from a practical standpoint this yields floor slopes that are too steep.

FIGURE 17.2 General Shapes of Auditoria (Doelle, 1972)

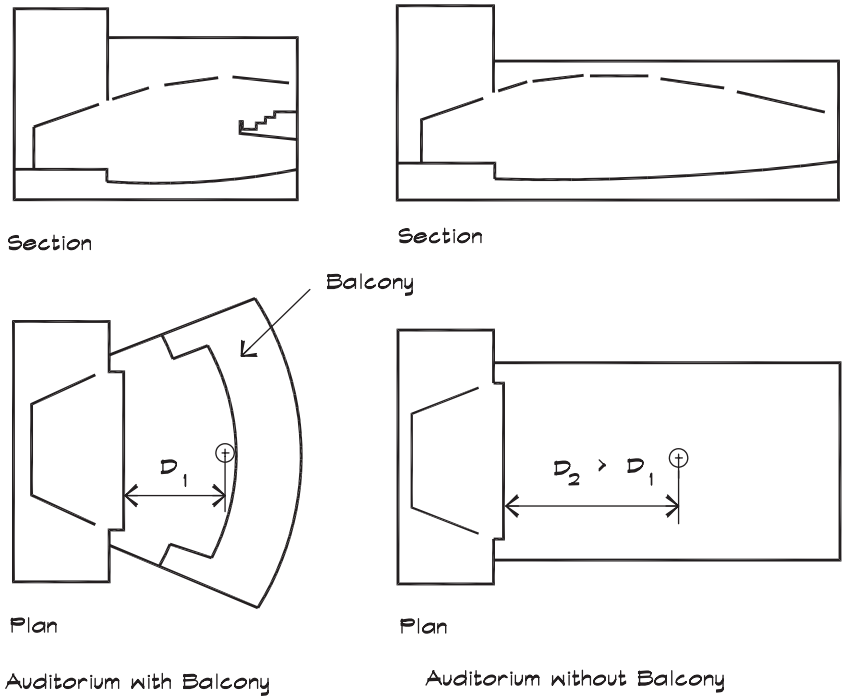
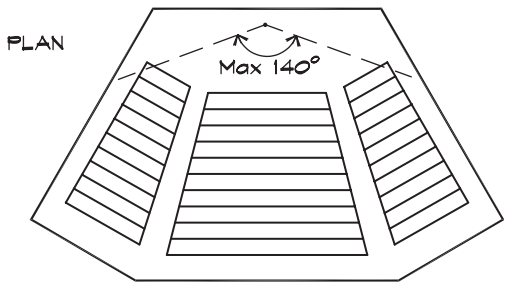


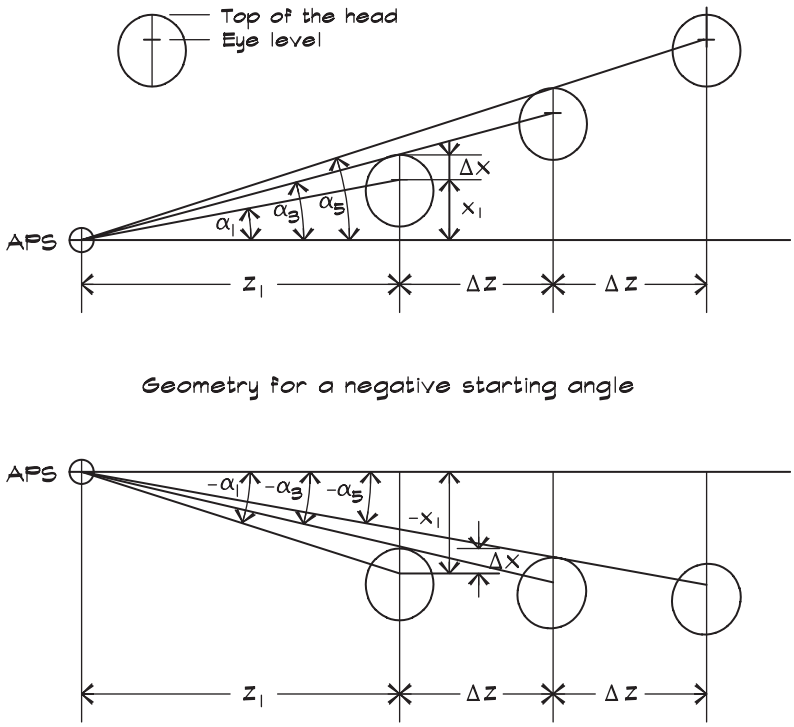
FIGURE 17.3 Seating Layout for a Lecture Hall



It is assumed that a person will adjust his position to look between the patrons seated in the next row so most theaters are designed for every-other-row visibility.

Figure 17.4 shows a typical sight-line design problem. The slope of the floor will depend dramatically on the APS that is selected. A high APS such as that found in a movie theater will still be visible, even with a relatively shallow floor slope. Stage floor heights are set low enough that a person sitting in the front row can see the actors' feet, but high enough that the APS does not force excessive floor rake. The eye height of a seated person ranges from 44 inches (1.12 m) for an adult female to 48 inches (1.22 m) for an adult male (Ramsey and Sleeper, 1970). Stage heights are fixed at between 40 and 42 inches (1.02–1.07 m) above the floor. The floor slope is determined by drawing a series of sight lines from the APS to a patron's ear level using standard anatomical data or by iteratively

FIGURE 17.4 Geometry of Theatrical Sight Line Calculations Every Other Row Sight Lines



applying a mathematical relationship

$$\tan \alpha_1 = \frac{x_1}{z_1} \tag{17.1}$$

where the index 1 refers to the first row and each subsequent index to the next row. The odd numbered rows are calculated for every-other-row sight lines. For this case the third row angle is

$$\tan \alpha_3 = \frac{x_1 + \Delta x}{z_1} \tag{17.2}$$

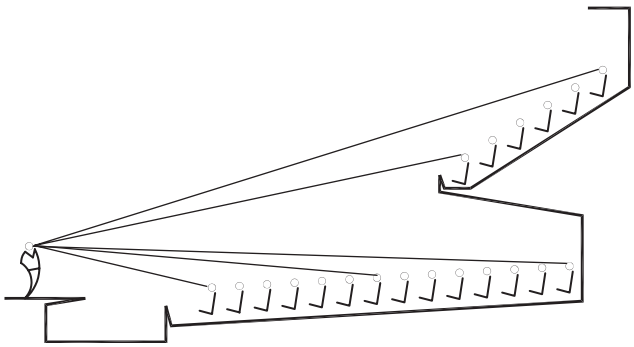
Subsequent odd numbered rows are calculated iteratively

$$\tan \alpha_n = \frac{\left\{ z_1 + \left[\frac{(n-1)}{2} - 1 \right] \Delta z \right\} \tan \alpha_{n-2} + \Delta x}{z_1 + \left[\frac{(n-1)}{2} - 1 \right] \Delta z} \tag{17.3}$$

where n = 5, 7, 9, and so forth, $\Delta x = 5$" (12.7 cm) for two rows, and $\Delta z = 2\cdot$ (row spacing).

Where there are fixed seats, grazing effect produces an attenuation that depends on the angle of incidence. The lower the angle, the greater the effect. The angle can be increased by

FIGURE 17.5 Good Sight Lines Yield Good Direct Sound



raising the talker on a platform or by raking the angle of seating as in Fig. 17.5. Seating rake is set by the sight-line requirements that are fixed by the APS and by the relative heights of each row of seats. In general the higher the APS the lower the seating rake. The rise of each row of seating can be calculated using Eq. 17.3. In the orchestra level seating a 1:9 rake for the first ten rows, and thereafter a 1:8 slope, yields a good result for a theater stage having a normal 42" (1.07 m) height. Building codes, which require no more than a 1:12 slope for handicapped access, may dictate the floor design.

In lecture halls, where the APS is selected to be at or above the waist of the lecturer or at the bottom of the writing board, the rake can be modest. In a large flat-floored classroom, a platform of 1 ft (0.3 m) height can improve the sight lines significantly. In small classrooms, seating fewer than 50 people, a platform is not necessary.

Sound Distribution

Unamplified speech can be augmented by physically placing hard surfaces in positions where they can distribute sound to the audience. Reflectors must have sufficient size that they scatter the frequencies of interest and should be close enough so that the reflection delay time is less than 30 to 50 msec. To provide this support, a hard ceiling is preferred in a lecture hall and auditorium (50–500 seats). In small classrooms (< 50 seats) the direct field, along with support from the walls, provides sufficient loudness and control of reverberant noise using an absorptive ceiling as the normal choice.

The orientation of a reflective element is determined by the required coverage area of the scattered sound. For specular reflection, the deflected angle is determined by locating the mirror image point of the sound source and by then drawing a line from the image point through the point of reflection toward the receiver. An example is given in Fig. 7.1. When the reflecting surface has a finite size, it is not an effective reflector over its entire length. Simple reflection studies illustrate the procedure. In these cases a reflection should be considered only if it occurs at a point more than one-half wavelength from the end of the reflector. Where the reflector is curved, the reflection angle is determined by mirroring the incident ray about the line connecting the center point of the curved surface with the intersection point of the ray and the surface. This is relatively straightforward in a CAD program; however, simple ray tracing does not tell the whole story since the scattered intensity falls off rapidly with increasing included angle, as was illustrated in Eq. 7.37.

The shape of the ceiling can be used to distribute sound evenly throughout an auditorium. Figure 17.7 shows a simple example of a flat ceiling. In this example the reflected rays

FIGURE 17.6 Reflection Studies from Ceiling Panels

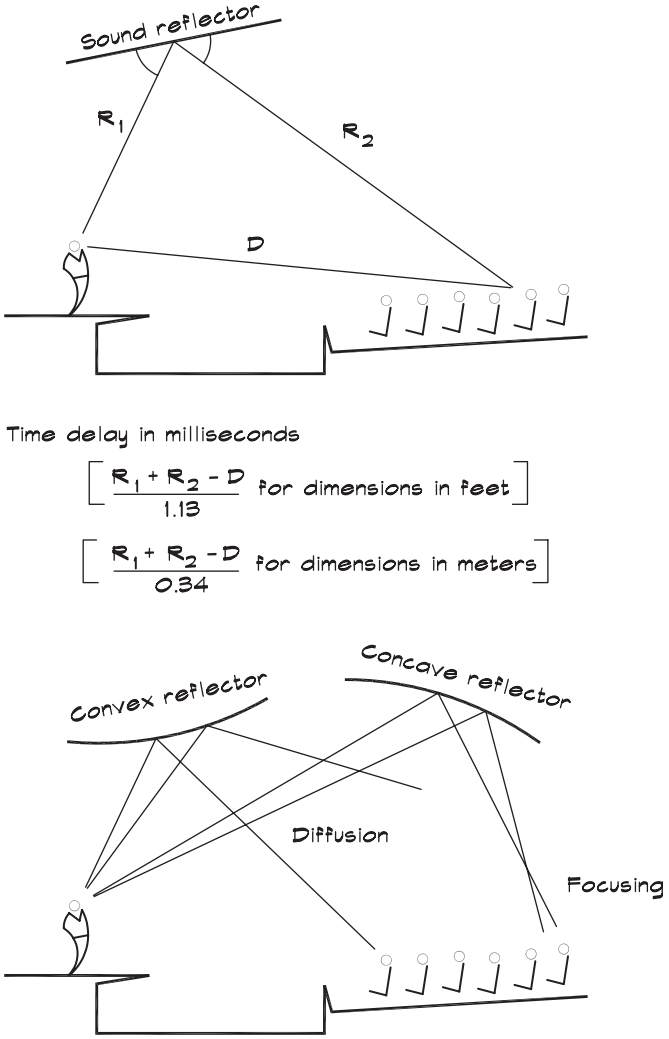


FIGURE 17.7 Reflections from a Flat Ceiling Section

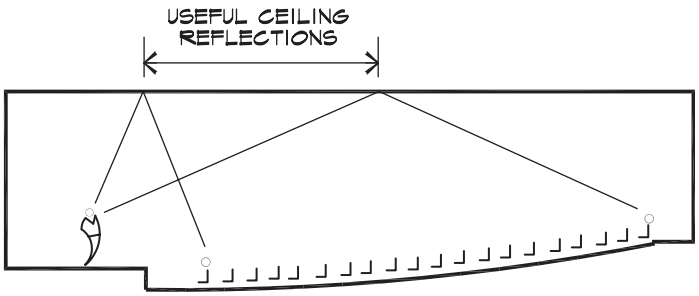


FIGURE 17.8 Reflected Sound from a Segmented Ceiling (Doelle, 1972)

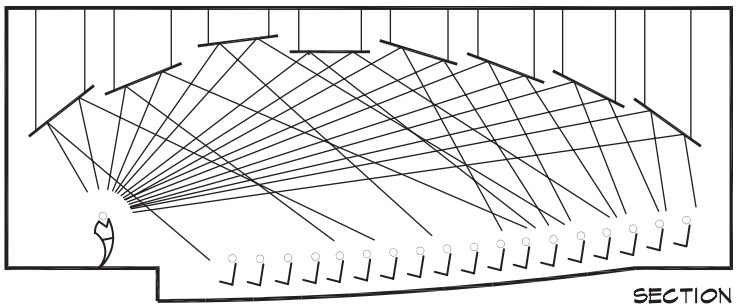
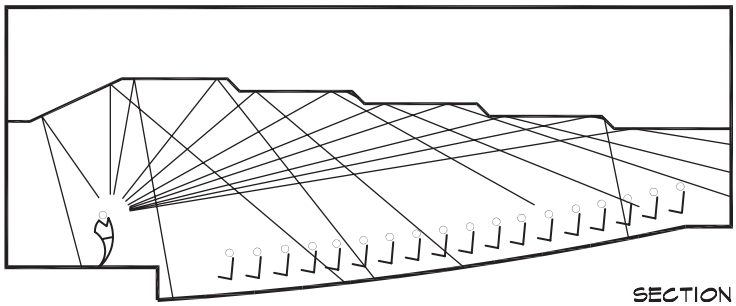


FIGURE 17.9 Reflected Sound from a Stepped Flat Ceiling



illuminate the front and middle portions of the space but much of the energy falling on the rear portion of the ceiling is grounded out on the absorptive rear wall.

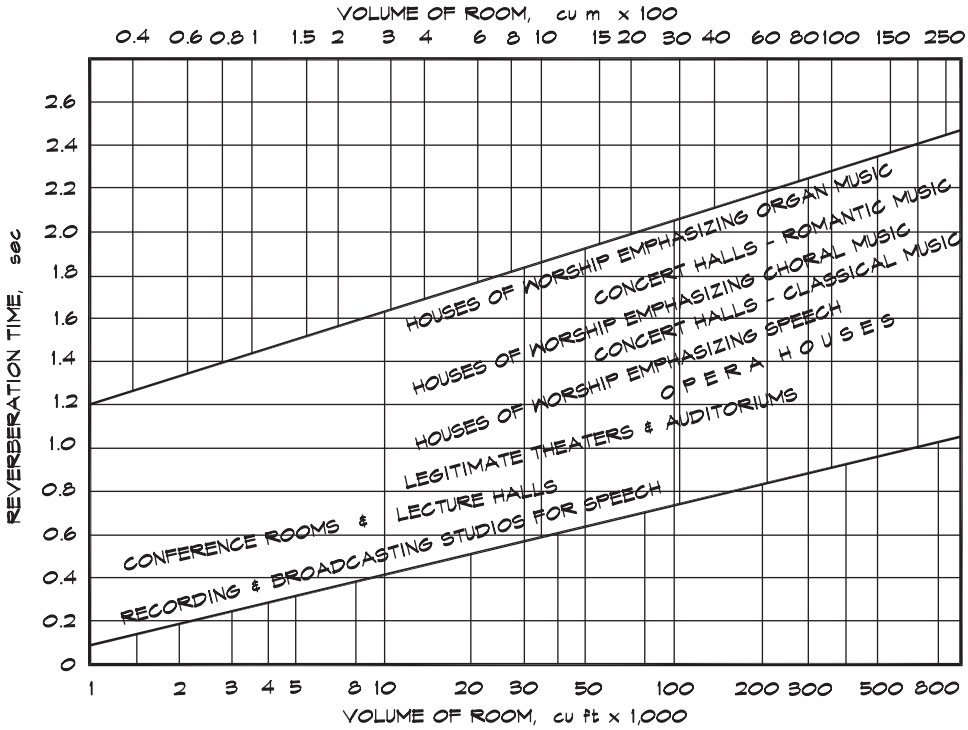
To improve the design, the ceiling can be segmented as in Fig. 17.8 or the seating raised and the ceiling stepped as in Fig. 17.9. Note that only about half of the ceiling provides useful specular reflections in both Fig. 17.7 and 17.8 since the ends of the segmented reflectors are diffusive.

The energy distribution is dependent on the location of the talker, which may vary, so slightly convex panels may be used to provide additional flexibility. Panels should not be used to reflect sound directly down, or back to the listener from behind, since this shifts the perceived source location overhead.

Reverberation

Reverberation can be the boon or the bane of the acoustical performance of a room. In general, the more speech content there is to the sound, the lower the ideal reverberation time. For classrooms and small lecture halls times at or below one second are preferred. Longer reverberation times are desirable for music; the ideal length depends on both the room size and the type of music. For light opera such as Gilbert and Sullivan, where understanding the complicated play of words is critical, a time of 1.0 to 1.2 seconds would not be too low. For a Mozart opera preferred reverberation times might range from 1.2 to 1.5 seconds. A Wagnerian opera is ideal in a 1.5 to 1.6 second room, and romantic symphonies sound best in a 1.7 to 2.1 second hall. For organ concerts and chant, reverberation times between 2.5 to 3.5 seconds

FIGURE 17.10 Reverberation Times vs Room Volume



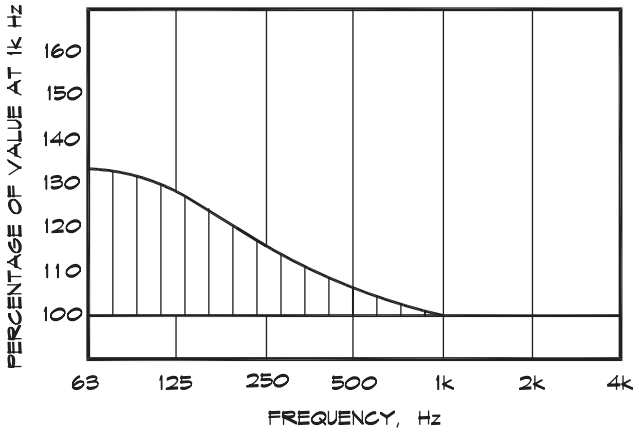
are not too long. Clearly there is no single reverberation time that is perfect for all uses of a given room. Variations of 5 to 10% from the ideal values are commonplace.

Various authors have made recommendations on ideal reverberation times for different types of spaces. Figure 17.10 shows a synthesis of a number of these graphs (e.g., Doelle, 1972; Knudsen and Harris, 1950; Long, 1999). Recent trends, particularly in the design of churches for electronically reinforced music, have driven the desirable reverberation times in large spaces downward, since reverberation can be added back electronically. Reverberation time recommendations for motion-picture theaters are given in Fig. 17.26.

Authors Knudsen and Harris (1950) and Doelle (1972) have recommended that for music the reverberation times at frequencies below 500 Hz rise to a number higher than the mid frequency value. Beranek (1996), citing measured results from various halls, recommends a factor of about 1.2 times the 500 to 1000 Hz value at 125 Hz. A recommended graph is shown in Fig. 17.11. A rising bass reverberation is good practice for performance rooms used for unreinforced music but not necessarily desirable in spaces where the low-frequency is provided by loudspeakers. As a practical matter it is difficult to achieve a rising reverberation time at very low frequencies, due to the weight and thicknesses of the materials required.

Since single-layer 16-mm (5/8”) gypsum board is nearly 30% absorptive at 125 Hz, a rising bass requires the use of multiple-layer gypboard or thick plaster construction. In the best concert halls the use of 25 to 50 mm (1”–2”) plaster is common. For speech the reverberation time behavior with frequency should be flat. In large rooms this is also difficult to achieve due to air attenuation, and the times fall off above 1 kHz. In large concert halls the HVAC system must include humidity control to reduce the high-frequency losses.

FIGURE 17.11 Ratio of the Bass to Mid-Frequency Reverberation Time



Recommended variation of T_{60} with frequency for music. At frequencies lower than 1000 Hz the T_{60} should increase according to the values inside the shaded area. For speech the T_{60} should remain flat with frequency.

Signal-to-Noise Ratio

Background noise levels in small classrooms and lecture halls are designed to an NC 30 (35 dBA) and larger auditoria to an NC 25 (30 dBA). The difference is due to the greater loss of loudness in the larger space. Some authors (Peutz and Klein, 1974) recommend that the received level be at least 25 dB higher than the background noise level for adequate intelligibility. Others (Bradley, 1986) hold that a 10–15 dB margin is a more reasonable choice. The latter value is consistent with an NC 30 background level and a direct-field level of 45 dBA, based on a speaking voice sound power level of 75 dBA and a source to receiver distance of 25 feet (7.6 m).

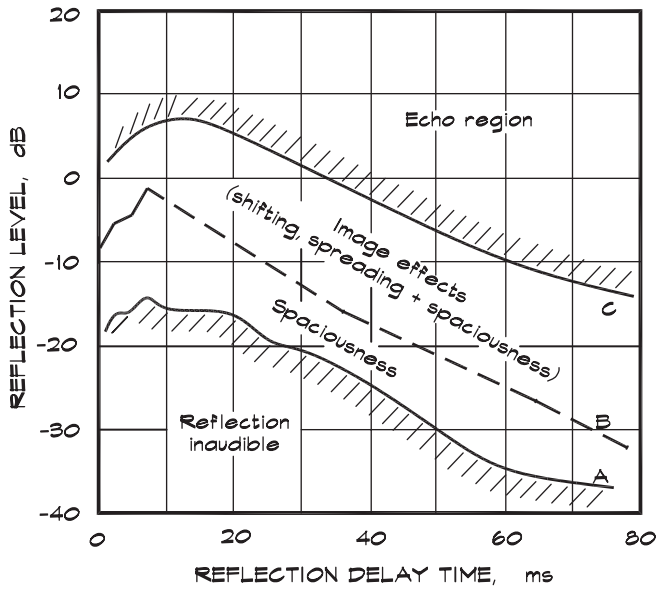
When the reverberant field is the masking noise a higher level can be tolerated. In these cases signal-to-noise ratios are rarely positive and a signal-to-reverberant-noise of −6 dB can still yield good intelligibility. This is discussed in more detail in Chapt. 18.

Acoustical Defects

The presence of acoustical defects can contribute to poor intelligibility and general discomfort in rooms. The principal defects, in addition to those already discussed, are multiple or long delayed reflections, focusing, coloration, and low-frequency phenomena such as room resonances and locally high-amplitude sound fields. In large auditoria there are also shadowed areas under balconies, coupled spaces with mismatched reverberation characteristics, and excess attenuation due to grazing incidence. Not all these defects are important in every room and some may be present without affecting the room's primary use.

There are a number of phenomena, associated with single or multiple reflections, that can detract from good intelligibility in rooms and should be avoided. These include long-delayed reflections, echoes, and flutter. Echoes occur when a sound of sufficient loudness arrives later than the direct field by more than a given time. The cause might be a single reflection from a rear wall of an auditorium, particularly if it is concave. Figure 17.12 shows

FIGURE 17.12 Perception of Lateral Speech Reflections (Everest, 1994)



the effect of reflections for various amplitudes and delay times, as simulated in an anechoic environment. Below curve B the echo increases the perception of spaciousness, while below curve A the reflected sound is reduced to inaudibility. Above curve C the reflection is perceived as an echo.

Echo and reverberation are not the same thing. Echo is a repetition of the original sound that is distinctly perceptible, whereas reverberation is a prolongation of the sound through multiple reflections, which is frequently beneficial for music. Long-delayed reflections are like echoes, but have a somewhat shorter delay time. They are not perceived as separate sounds, but blur the understanding of the original sound. Flutter echoes are sounds that persist locally due to multiple reflections between parallel planes, concave, or chevroned surfaces. They can be caused by two, three, or more reflections. Figure 17.13 gives several examples of acoustical defects.

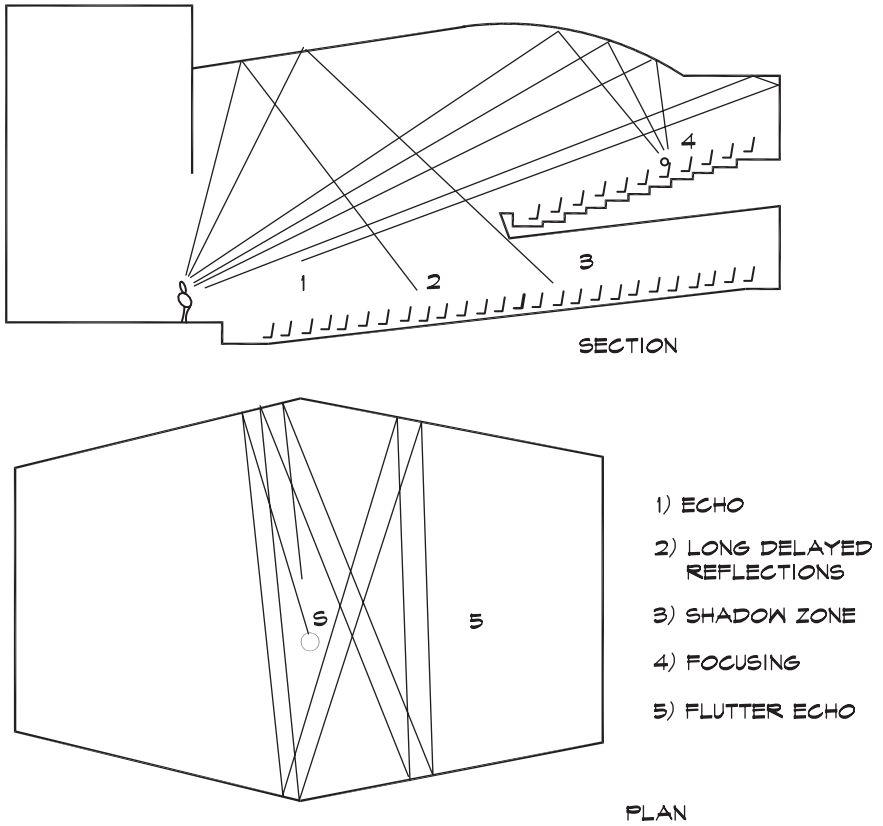
Coloration is the emphasis of certain frequencies or frequency bands over others. It can be caused by room-mode buildup or by absorptive materials that only absorb in certain frequency ranges. Focusing is the buildup of sound energy in localized regions of a room, due to concave surfaces. Shadowing is the blockage of sound traveling from the source, or from a significant reflecting surface, to the receiver. Each of these defects can detract from the overall acoustical environment in a room and each can be avoided with careful design.

17.2 SPEECH INTELLIGIBILITY

Speech-Intelligibility Tests

Speech-intelligibility tests for an unamplified talker are carried out using a single loudspeaker, ideally having a directivity similar to that of the human voice. Prerecorded words are presented in a neutral context carrier phrase such as, “Word number __ is ___”, at one or more calibrated levels, in rooms exhibiting a variety of acoustical conditions. As we gleaned

FIGURE 17.13 Examples of Acoustical Defects (Doelle, 1972)



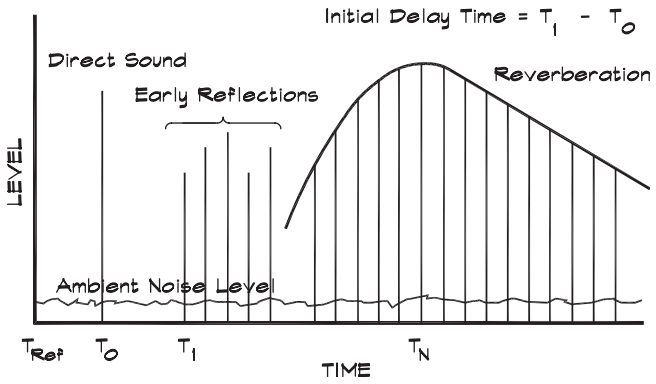
from Fig. 3.19, there is not just one test that gives the single answer for speech intelligibility. Rather there are many different results that depend on the details of the test and the type of material presented to the listener. The prediction of speech intelligibility in an enclosed space thus combines the results of listening tests with knowledge of the room's acoustical properties in such a way as to produce a predictable outcome.

Energy Buildup in a Room

When a sound is generated by a single source, the listener receives, in rapid succession, the direct-field signal followed by individual early reflections, and a rising swell of merged reflections whose sum becomes the reverberant field, which finally decays at a rate characteristic of the space. Figure 17.14 shows an example of the idealized pattern. In this figure the three temporal regions are neatly separated; in practice the divisions are not so distinct. The early reflections and the reverberation may be merged. If long-delayed reflections are present they may arrive during the reverberant decay. Sometimes individual reflections are louder than the direct sound when focusing or grazing attenuation is present. The reverberant field can be louder than the direct sound when the receiver is a relatively long way from the source.

The time between the arrival of the direct sound and the first major reflection is called the initial delay gap. If this gap is short enough, early reflections can contribute to increased

FIGURE 17.14 Idealized Acoustic Response of a Room to an Impulse Excitation



intelligibility, a broadening of the sound image, and a pleasant augmentation of the sound level. If it is too long, its effect will be to decrease intelligibility.

Background noise, along with long-delayed reflections and persistent reverberation, serve to decrease intelligibility. Background noise that interferes with the comprehension of speech can originate from many sources: people, HVAC systems, exterior noise sources such as traffic, or electronically generated masking noise, which is purposefully introduced to increase speech privacy. When words are spoken in a room, the reflections off the walls and other surfaces will eventually have a negative effect on speech intelligibility, either through long delayed individual reflections or as part of the general buildup of background noise. Thus the reverberant field of speech itself can also become the source of masking noise.

Room Impulse Response

Although it is possible to measure speech intelligibility directly in an existing room, it is also useful to have algorithms to predict it before a room is constructed. As we discussed in Chapt. 11, the impulse response completely defines the properties of a system, and we can predict the result of introducing an arbitrary forcing function (speech) by convolving (integrating) the input with the room's impulse response (Eq. 11.40). An exact formulation of a room's response is not available a priori, but it can be approximated by using the simplifying assumptions or by ray tracing.

A simple model assumes that the room is diffuse and that there exists a reverberant field characterized by a reverberation time. This model ignores common acoustical defects such as long delayed reflections, flutter echo, focusing, and the process of reverberant sound buildup. More complicated analyses utilizing ray tracing can describe these effects, but they are not expressible in a closed-form equation and are time consuming. The approximate methods yield results that are sufficiently accurate, as long as steps are taken to avoid the acoustical defects, which they do not include. The approach is to use the direct and reverberant sound energy densities previously discussed. The direct field (Eq. 2.56) energy density is given by

$$D_{\mathrm{d}} = \frac{\mathrm{E}}{\mathrm{S}\,\mathrm{c}_0\,\mathrm{t}} = \frac{\mathrm{W_S}}{\mathrm{S}\,\mathrm{c}_0} = \frac{\mathrm{p}^2}{\rho_0\,\mathrm{c}_0^2} \tag{17.4}$$

and

$$D_{\mathrm{d}} = \frac{\mathrm{Q}\,\mathrm{W_S}}{4\,\pi\,\mathrm{c_0}\,\mathrm{r}^2} \tag{17.5}$$

The steady state reverberant field (see Eq. 8.82) energy density is

$$\mathrm{V}\,D_{\mathrm{r}} = \frac{4\,\mathrm{W_S}\,\mathrm{V}}{\mathrm{c_0}\,\mathrm{S_T}\,\overline{\alpha}} \tag{17.6}$$

or

$$D_{\mathrm{r}} = \frac{\mathrm{Q}\,\mathrm{W_S}}{4\,\pi\,\mathrm{r_c^2}\,\mathrm{c_0}} \tag{17.7}$$

where $\mathrm{r_c}$ is defined as the critical distance, the point at which the direct sound pressure level is equal to the reverberant field level.

$$\mathrm{r_c} = \sqrt{\frac{\mathrm{Q\,R}}{16\,\pi}} \tag{17.8}$$

where $\mathrm{W_s}$ = source sound power (W)
V = room volume ($\mathrm{m^3}$ or $\mathrm{ft^3}$)
Q = source directivity in the receiver direction
$\mathrm{c_0}$ = speed of sound in air (m / s or ft / s)
ρ_0 = density of air ($\mathrm{kg\,/\,m^3}$ or $\mathrm{lbs\,/\,ft^3}$)
$\overline{\alpha}$ = average room absorption coefficient
R = room constant ($\mathrm{m^2}$ or $\mathrm{ft^2}$)
S = area of the control surface ($\mathrm{m^2}$ or $\mathrm{ft^2}$)
$\mathrm{S_T}$ = total surface area of the room ($\mathrm{m^2}$ or $\mathrm{ft^2}$)
$\mathrm{T_{60}}$ = reverberation time (s)

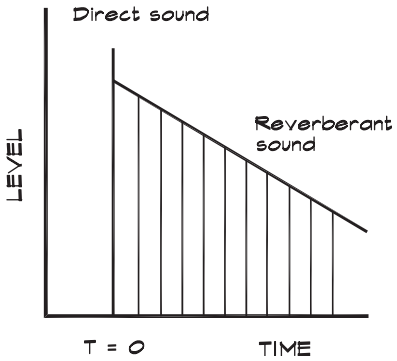
We can approximate the impulse response of a room by assuming that the sound field is made up of only a direct and perfectly reverberant field that decays at a rate defined by the reverberation time. This idealized model, illustrated in Fig. 17.15, ignores all individual reflections.

The sound power density as a function of time ($\mathrm{t} \geq 0$) is (Houtgast et al., 1980)

$$w\,(\mathrm{t}) = \frac{\mathrm{W_S}}{4\,\pi\,\mathrm{c_0}}\left[\frac{\mathrm{Q}}{\mathrm{r}^2}\,\delta\,(\mathrm{t}) + \frac{\mathrm{Q}\,\kappa}{\mathrm{r_c^2}}\,\mathrm{e}^{-\kappa \mathrm{t}}\right] \tag{17.9}$$

where $\delta(\mathrm{t})$ = Dirac delta function at $\mathrm{t} = 0$
κ = decay rate = $13.82\,/\,\mathrm{T_{60}}$ (1/s)
and the terms within the brackets are the impulse response of the room.

FIGURE 17.15 Simplified Room Impulse Response



Once the impulse response is known, the sound energy density arriving at a receiver during the time period from t = 0 to t = T is found by integrating over time

$$D_{0-\mathrm{T}} = \int_0^{\mathrm{T}} w\,(\mathrm{t})\,\mathrm{dt} \tag{17.10}$$

The steady-state energy density for a continuous driving function is found by setting $\mathrm{T} = \infty$. In this way we recover Eqs. 17.5 and 17.7.

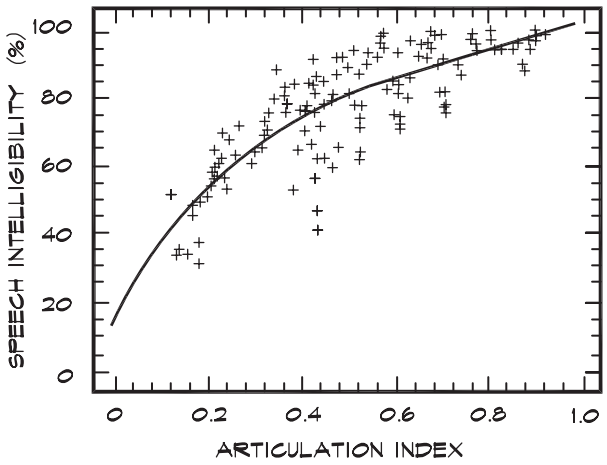
Speech-Intelligibility Metrics—Articulation Index (AI)

There are several metrics currently enjoying use for the prediction of the intelligibility of speech in rooms: the Articulation Index (AI), the Articulation Loss of Consonants ($\mathrm{AL_{cons}}$), the Speech Transmission Index (STI), and the various signal-to-noise ratios including the Useful to Detrimental Energy Ratio (U_τ) and the Useful to Late Energy Ratio (C_τ).

Much of the pioneering work in communication acoustics was done at Bell Laboratories, where engineers studied methods of improving the intelligibility of telephone conversations. Harvey Fletcher (1884–1981) was one of these early pioneers. Although Fletcher's work dates from the 1920s and 1930s, much of it was not revealed publicly until the publication of later papers and his classic book in 1953.

Fletcher (1921) proposed to quantify the speech distortion in telephone systems by relating it to articulation scores. He defined the "articulation," which ranged from 0 to 1, as an overall measure of the intelligibility of speech transmitted through a system. One of Fletcher's contributions was the discovery of the probabilistic nature of intelligibility, and indeed the definition of articulation is the probability of understanding an individual sound. If, for example, a syllable consists of a consonant-vowel-consonant (cvc) sequence, the probability of understanding the whole sequence would be the product of the probabilities of understanding each separate consonant or vowel. When this was combined with the realization that the probabilistic approach carried over into the analysis of separate frequency bands (published later in Fletcher and Galt, 1950), the basis for the Articulation Index was established. French and Steinberg (1947) formalized the method of measurement and Kryter (1970) published a method of calculating the expected speech intelligibility in rooms using the sum of weighted signal-to-noise ratios in third-octave frequency bands (Chapt. 3).

FIGURE 17.16 Measured Intelligibility vs Articulation Index (Bradley, 1986)



In 1986 Bradley published a study comparing the accuracy of various articulation metrics, and Figure 17.16 shows his result for Articulation Index.

Articulation Index (AI) is like virtually all other intelligibility prediction schemes in that it uses a signal-to-noise ratio as part of the calculation. The differences among the various schemes are how the terms "signal" and "noise" are defined. In AI calculations, the signal is the long-term rms average speech level (direct + reverberant) plus 12 dB, and the noise is the steady background noise level in each frequency band. AI is difficult to use as an intelligibility prediction methodology since it does not have a built-in way of accounting for reverberant noise. In the ANSI standard (ANSI S3.5-1969) there is an empirical correction table for reverberation time (Knudsen and Harris, 1950) but no way of dealing with the contribution of the reverberant field. Where an electronic masking system generates the steady background noise, AI yields good results in the assessment of privacy. Typical results of intelligibility scores versus AI values were given in Fig. 3.19.

Articulation Loss of Consonants ($\mathbf{AL_{cons}}$)

Early researchers (e.g., Knudsen, 1932) found that intelligibility was based on the recognition of consonants rather than vowels and developed metrics based on this concept. Maxfield and Albersheim (1947) at Bell Laboratories examined the measured articulation-loss-of-consonants data published by Steinberg (1929) and Knudsen (1932) and plotted them versus a steady-state direct-to-reverberant energy ratio. They found that the data did not lie along a straight line and subsequently developed the concept of a liveness factor, for use with microphone pickups, which they defined as

$$L = \int_0^\infty D_r(t)\,dt / D_d \tag{17.11}$$

where $D_r(t)$ is the reflected-energy density at any time, t, and D_d is the direct-field energy density. Under the assumption of a reverberant field and unit directivity, the liveness can be

written (Bistafa and Bradley, 2000) in metric units as

$$L = \frac{\mathrm{T}_{60}\, D_\mathrm{r}}{13.82\, D_\mathrm{d}} = \frac{4\,\pi}{(13.82)^2}\frac{\mathrm{c}_0\, \mathrm{T}_{60}^2\, \mathrm{r}^2}{\mathrm{V}} = 22.6\,\frac{\mathrm{T}_{60}^2\, \mathrm{r}^2}{\mathrm{V}} \tag{17.12}$$

which is the reverberant-to-direct energy ratio multiplied by the reverberation time. Bistafa and Bradley (2000) fitted a curve to the Maxfield and Albersheim plots, which related articulation loss (AL) to liveness based on Steinberg's data

$$\mathrm{AL} = 4.5\, L^{0.67} \tag{17.13}$$

In 1971 Peutz measured the speech intelligibility in rooms using cvc phonetically balanced words in Dutch. Like Knudsen, he also found that the articulation loss was much smaller for vowels than for consonants, so that consonant loss probabilities controlled cvc recognition. Unlike Maxfield and Albersheim, however, he found a linear relationship between the measured articulation loss of consonants and liveness. His relationship, in metric units, is shown in Eq. 17.14, and assumes a directivity of one and negligible background noise.

$$\mathrm{AL}_\mathrm{cons} = 8.9\, L + a = 200\,\frac{\mathrm{T}_{60}^2\, \mathrm{r}^2}{\mathrm{V}} + a \tag{17.14}$$

where a is a correction factor that can vary from 1.5% for a "good" listener, to 12.5% for a "bad" listener. This equation is said to hold as long as the listener is no more than a limiting distance r_ℓ away, where

$$\mathrm{r}_\ell = 0.21\,\sqrt{\mathrm{V}/\mathrm{T}_{60}} \quad (\mathrm{m}) \tag{17.15}$$

This is the distance at which the direct-field level is 10 dB below the reverberant-field level for a directivity of one. Beyond that point Peutz states that the articulation loss is given by

$$\mathrm{AL}_\mathrm{cons} = 9\,\mathrm{T}_{60} + a \quad \text{for } \mathrm{r} \geq \mathrm{r}_\ell \tag{17.16}$$

In terms of the limiting distance, the equation is

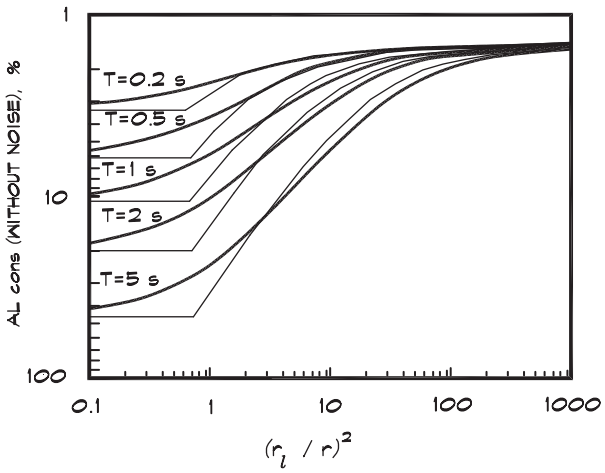
$$\mathrm{AL}_\mathrm{cons} = 9\,\mathrm{T}_{60}\,\frac{\mathrm{r}^2}{\mathrm{r}_\ell^2} + a \quad \text{for } \mathrm{r} < \mathrm{r}_\ell \tag{17.17}$$

Equations 17.16 and 17.17 are known as the architectural versions of the Peutz equations. They do not account for early reflections, discrete echoes, background noise, or frequency dependence of the variables.

Bistafa and Bradley (2000) published a continuous version of the noiseless equations based on work by Peutz (1974) and Peutz and Kok (1984).

$$\mathrm{AL}_\mathrm{cons} = 9\,\mathrm{T}_{60}\left[\frac{1}{1 + (\mathrm{r}_\ell/\mathrm{r})^2}\right] + a \tag{17.18}$$

FIGURE 17.17 ALcons–N vs $(r_\ell/r)^2$ with Reverberation Time as a Parameter (Bistafa and Bradley, 2000)



and Klein (1971) added back the directivity by defining the limiting distance in terms of the room constant.

$$r_\ell = 0.45\sqrt{QR} \quad \text{(m)} \tag{17.19}$$

A comparison of Eqs. 17.16 and 17.17 to 17.18 is shown in Fig. 17.17.

In 1974 Peutz and Klein published a graphical method of accounting for the presence of noise. This was curve fitted by Bistafa and Bradley (2000) and in its continuous form is

$$AL_{cons} = 9\,T_{60}\left\{\frac{1}{1+(r_\ell/r)^2}\right\}\left[1.071\,T_{60}^{-0.0285}\right]^{(25-L_{SN})} + a \tag{17.20}$$

where L_{SN} is the signal-to-noise ratio $L_{SN} = L_s - L_n$ in dB. According to Eq. 17.20, when the signal is less than 25 dB above the background noise there is a reduction in speech intelligibility, which becomes progressively worse as the signal level decreases. If the signal level is greater than 25 dB above the noise, there is no degradation due to background noise and the noise term is dropped. Here the signal level is the direct plus reverberant speech level, and the noise level is the steady background level having the same spectral shape as the speech level. Peutz and Klein did not include information on the spectrum of the background noise or the frequency at which the calculations are to be carried out. Standard practice is to use the 2000 Hz octave band.

Davis and Davis (1987) recommend AL_{cons} for general use in sound-system design, although in this form there is no single value of the directivity when multiple loudspeakers are used. Jacobs (1985), experimenting with single high, medium, and low-Q loudspeakers, found a poor correlation between the predicted and measured intelligibility, particularly in highly reverberant rooms. His data indicated that AL_{cons} underpredicted the speech

intelligibility for low- and medium-Q loudspeakers and overpredicted with a high-Q device. Bistafa and Bradley (2000) also found a poor correlation between AL_{cons} predictions and those based on STI and U_{50} metrics. They recommended that its use be limited to classrooms and small meeting rooms. This would seem to be a good approach. AL_{cons} includes the reverberant field as part of the signal in a signal-to-noise ratio, but switches to a different formulation when the reverberant field dominates the direct field.

Speech Transmission Index (STI)

Researchers in optics (Baker, 1970), seeking to quantify the distortion of light received from stars, developed the optical transfer function, which was based on a mathematical formulation called the modulation transfer function (MTF). Houtgast, Steeneken, and Plomp (1980) reasoned that stars are the spatial equivalent of an acoustical impulse source and this approach could be useful in evaluating distortion in rooms. As we discussed in Chapt. 4, the MTF uses a modulated sinusoidal input

$$I_{in}(t) = I_{in}(1 + \cos \omega_m t) \tag{17.21}$$

which is introduced into a room. It is convolved with the room's impulse response g (t') to obtain an output

$$I_{out}(t) = \int_0^\infty I_{in}(t - t')\ g(t')\, dt \tag{17.22}$$

which has the general form

$$I_{out}(t) = I_{out}\left\{1 + m\left[\cos \omega_m (t - \theta)\right]\right\} \tag{17.23}$$

The modulation transfer function is defined as

$$m(\omega_m) = \frac{\left| \int_0^\infty g(t)\, e^{-j\omega_m t}\, dt \right|}{\int_0^\infty g(t)\, dt} \tag{17.24}$$

and $\omega_m = 2\pi f_m$ is the modulation frequency. Schroeder (1981) pointed out that this is the normalized Fourier transform of the power density impulse response. Assuming a diffuse field, the impulse response for both the direct and reverberant field components is

$$g(t) = \frac{Q}{r^2}\, \delta(t) + \frac{Q\kappa}{r_c^2}\, e^{-\kappa t} \tag{17.25}$$

where κ is the exponential decay constant of the reverberant energy, $\kappa = 13.82 / T_{60}$.

When background noise is added to the mix the output intensity is

$$I_{sum} = I_{out}(t) + I_n \tag{17.26}$$

and using Eq. 17.23,

$$I_{sum} = I_{out}\left[1 + m\cos\,\omega_m\,(t-\theta)\right] + I_n \tag{17.27}$$

which can be written

$$I_{sum} = (I_{out} + I_n)\left[1 + m\,\frac{I_{out}}{I_{out} + I_n}\cos\,\omega_m\,(t-\theta)\right] \tag{17.28}$$

So the modulation factor due to background noise is

$$m_n = \frac{I_{out}}{I_{out} + I_n} = \left[1 + 10^{-0.1\,L_{SN}}\right]^{-1} \tag{17.29}$$

and the signal-to-noise ratio in dB is $L_{SN} = 10\,\log\,(I_{out}\,/\,I_n)$.

This yields the overall modulation transfer function including both

$$m\,(\omega_m) = \frac{\left|\int_0^\infty g\,(t)\,e^{-j\,\omega_m\,t}\,dt\right|}{\int_0^\infty g\,(t)\,dt}\left[1 + 10^{-0.1\,L_{SN}}\right]^{-1} \tag{17.30}$$

room distortion and noise. We substitute the impulse-response function (Houtgast et al., 1980)

$$m\,(\omega_m) = \frac{\left(A^2 + B^2\right)^{1/2}}{C} \tag{17.31}$$

with

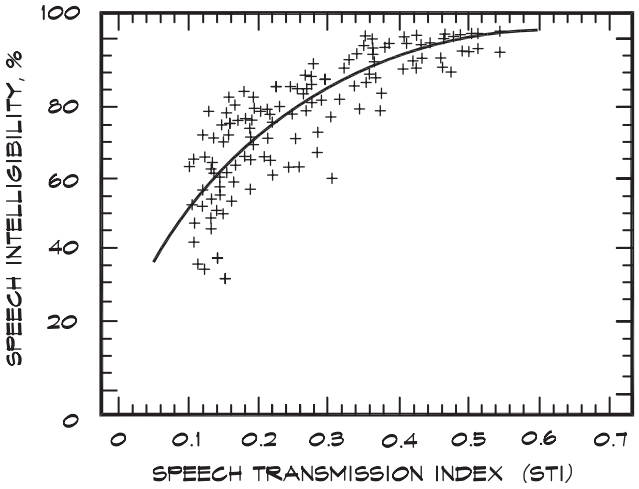
$$\begin{aligned} A &= \frac{Q}{r^2} + \frac{Q}{r_c^2}\left[1 + \left(\frac{\omega_m\,T_{60}}{13.8}\right)^2\right]^{-1} \\ B &= \frac{\omega_m\,T_{60}}{13.8}\,\frac{Q}{r_c^2}\left[1 + \left(\frac{\omega_m\,T_{60}}{13.8}\right)^2\right]^{-1} \\ C &= \frac{Q}{r^2} + \frac{Q}{r_c^2} \end{aligned} \tag{17.32}$$

and the critical distance defined in Eq. 17.8. This can be simplified in the far field, $\frac{r_c^2}{r^2} \to 0$ for unit directivity to

$$m\,(f_m) \cong \left\{1 + \left(2\,\pi\,f_m\,\frac{T_{60}}{13.8}\right)^2\right\}^{-1}\left[1 + 10^{-0.1\,L_{SN}}\right]^{-1} \tag{17.33}$$

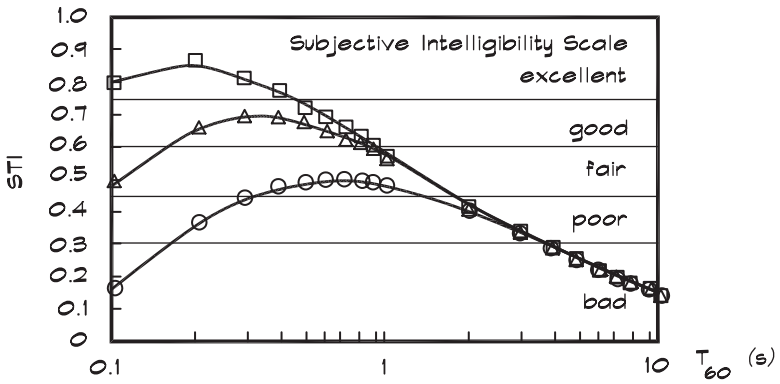
For a given modulation frequency an apparent signal-to-noise ratio and speech transmission index (STI) is calculated from the modulation index described in Eqs. 4.23 through 4.25, and from this an intelligibility can be determined.

FIGURE 17.18 Measured Speech Intelligibility vs Speech Transmission Index (Bradley, 1986)



In a study of classroom intelligibility, Bradley (1986) measured speech-intelligibility scores, including the effects of noise, and compared them to calculated STI values. The results are given in Fig. 17.18. Bistafa and Bradley (2000) plotted STI values versus reverberation time for unamplified speech in classrooms, which are reproduced as Fig. 17.19. Here we see that for a given signal-to-noise ratio the intelligibility can be maximized as a function of reverberation time.

FIGURE 17.19 Speech Transmission Index (STI) vs Reverberation Time (Bistafa and Bradley, 2000)



Measurements taken in a 300 sq m classroom for various differences between the background noise level, L_N and the long term average speech level at 1 m (65 dB), L_{Splm} at 1000 Hz. The directivity, Q, is assumed to be 2.

□ $L_N - L_{Splm} = -30$ dB

△ $L_N - L_{Splm} = -20$ dB

○ $L_N - L_{Splm} = -10$ dB

Signal-to-Noise Ratios ($\mathbf{C_t}$ *and* $\mathbf{U_t}$)

In 1935 two researchers, F. Ainger and M. J. O. Strutt, reported on the property of the ear that combines early-reflected sounds with the direct sound so as to increase the apparent strength of the whole. They suggested an energy ratio formula to quantify the effects of background noise and room acoustics on intelligibility. They called this ratio *impression*, which they defined as

$$Q = \frac{E_{\mathrm{d}} + E_{\mathrm{e}}}{E_{\ell} + E_{\mathrm{n}}} \tag{17.34}$$

where E_{d} = direct field energy (N m)
E_{e} = early part of the reflected energy (N m)
E_{ℓ} = late portion of the reflected energy (N m)
E_{n} = constant noise energy (N m)

They set the dividing line between early and late reflections at 1/16 second and set a lower limit of 1 for a satisfactory value of Q. If we write Eq. 17.34 in terms of energy densities we obtain a similar expression

$$Q = \frac{D_{\mathrm{d}} + D_{\mathrm{e}}}{D_{\ell} + D_{\mathrm{n}}} \tag{17.35}$$

where D_{d} = direct field energy density (W s / m^3)
D_{e} = early part of the reflected energy density (W s / m^3)
D_{ℓ} = late part of the reflected energy density (W s / m^3)
D_{n} = constant noise energy density $= \dfrac{\mathrm{p}^2}{\rho_0\, \mathrm{c}_0^2}$ (W s / m^3)

Using this model and the impulse response from Eq. 17.25 we can calculate the value of the impression (Bistafa and Bradley, 2000)

$$Q = \frac{1 + (\mathrm{r}_{\ell}/\mathrm{r})^2 - \mathrm{e}^{-0.86/\mathrm{T}60}}{\mathrm{e}^{-0.86/\mathrm{T}60} + 10^{0.1\,(\mathrm{L_n} - \mathrm{L_r})}} \tag{17.36}$$

where $\mathrm{L_n}$ = steady background noise level (dB)
$\mathrm{L_r}$ = steady reverberant signal level (dB)

The metric is seldom encountered now but is interesting, not only for its historical significance, but also as an introduction to more recent versions of the same concept using different cutoff times.

In the 1950s, Thiele (1953) published one of the earliest attempts at relating early to total sound energy ratio to intelligibility, which he called the definition, D. He considered the useful energy to be the direct plus the reflected energy that arrives within 50 msec of the direct sound. The definition can be written (Bistafa and Bradley, 2000) as

$$\mathrm{D}_{50} = \frac{1 + (\mathrm{r}_{\ell}/\mathrm{r})^2 - \mathrm{e}^{-0.69/\mathrm{T}60}}{1 + (\mathrm{r}_{\ell}/\mathrm{r})^2} \tag{17.37}$$

Definition does not account for the contribution of the background noise to the detrimental energy. It represents another early attempt to quantify speech intelligibility in terms of room acoustics.

Bradley (1986) used variations of the Q metric in his study of speech intelligibility in classrooms. These included the useful-to-detrimental noise ratio

$$\mathrm{U}_{\tau} = 10 \log \left[\frac{R_{\tau}}{(1 - R_{\tau}) + 10^{-0.1\,\mathrm{L_{SN}}}} \right] \tag{17.38}$$

where R_{τ} is the ratio between the early and the total energy

$$R_{\tau} = E_{\mathrm{e}} / (E_{\mathrm{e}} + E_{\ell}) \tag{17.39}$$

and the early-to-late signal-to-noise ratio

$$\mathrm{C}_{\tau} = 10 \log \left[\frac{R_{\tau}}{1 - R_{\tau}} \right] \tag{17.40}$$

which is obtained by setting the second term in the denominator of Eq. 17.38 equal to zero.

When these expressions are evaluated using the diffuse-field impulse response and a cutoff time of 50 msec we obtain

$$\mathrm{U}_{50} = 10 \log \left[\frac{1 + (\mathrm{r}_{\ell}/\mathrm{r})^2 - \mathrm{e}^{-0.69/\mathrm{T}_{60}}}{\mathrm{e}^{-0.69/\mathrm{T}_{60}} + 10^{0.1\,(\mathrm{L_n} - \mathrm{L_r})}} \right] \tag{17.41}$$

and

$$\mathrm{C}_{50} = 10 \log \left[\frac{1 + (\mathrm{r}_{\ell}/\mathrm{r})^2 - \mathrm{e}^{-0.69/\mathrm{T}_{60}}}{\mathrm{e}^{-0.69/\mathrm{T}_{60}}} \right] \tag{17.42}$$

Bradley (1986) published intelligibility versus U_{80} values in his study of classrooms, which are given in Fig. 17.20. Bradley worked with several cutoff times: 35, 50, 80, and 95 msec. He found (1998) that the differences using cutoff times between 50 and 95 msec are not great, for example, $\mathrm{C}_{80}(\mathrm{A}) \cong \mathrm{C}_{50}(\mathrm{A}) + 2$. The results are plotted in Fig. 17.21.

Weighted Signal-to-Noise Ratios ($\mathbf{C}_{\mathbf{t}}^{\alpha}$ *and* $\mathbf{U}_{\mathbf{t}}^{\alpha}$)

Early-to-late ratios were also the basis of work by Lochner and Berger (1964) in the Afrikaans language. These authors identified and separated the early sound energy, arriving at less than a certain time after the direct sound, from the later reflected sound. In their system the early arrivals are weighted and integrated over the time period and compared to the sound energy arriving after that time. They defined a useful-to-late energy ratio as

$$\mathrm{C}_{\tau}^{\alpha} = 10 \log \left[\frac{\int_{0}^{\tau} \alpha(\mathrm{t})\, w\,(\mathrm{t})\, \mathrm{dt}}{\int_{\tau}^{\infty} w\,(\mathrm{t})\, \mathrm{dt}} \right] \tag{17.43}$$

where $\alpha(\mathrm{t})$ is the average fraction of the energy of an individual reflection that is integrated into the useful early energy sum. This weighting term depends on the amplitude of the

FIGURE 17.20 Speech Intelligibility vs U_{80} Values (Bradley, 1986)

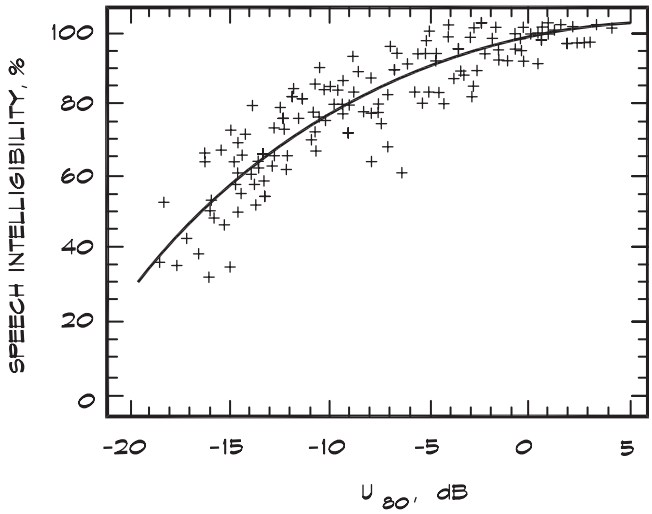
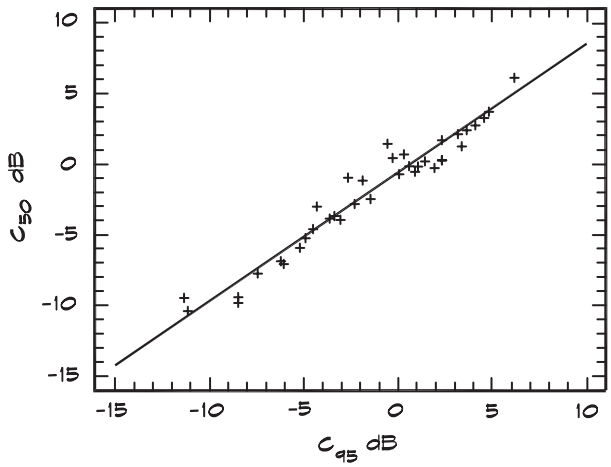


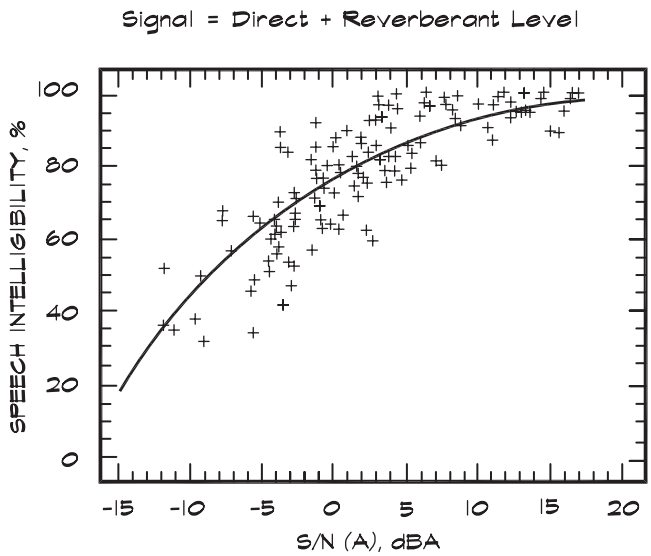
FIGURE 17.21 Measured C_{50} vs C_{95} values at 1 kHz (Bradley, 1986)



reflected energy, relative to the direct sound and the time of arrival. The $\alpha(t)$ term was included because the unweighted method proved highly sensitive to individual reflections arriving just before or just after the cutoff time. The weighting factor was set to 1 at a start time and to 0 at the finish time, and decreased linearly between them. Various algorithms have been used as a weighting function. Among them are

$$\begin{aligned} \alpha(t) &= 1 \quad \text{for } 0 \le t < t_1 \\ \alpha(t) &= \frac{t_2 - t}{t_2 - t_1} \quad \text{for } t_1 \le t \le t_2 \\ \alpha(t) &= 0 \quad \text{for } t > t_2 \end{aligned} \tag{17.44}$$

FIGURE 17.22 Measured Speech Intelligibility vs A-Weighted Signal-to Noise-Ratio (Bradley, 1986)



with $t_1 = 0.035$ s and $t_2 = 0.095$ s. For a diffuse field and a 95 ms cutoff time, Lochner and Burger's useful-to-detrimental ratio is

$$U_{95}^{\alpha} = 10 \log \left[\frac{1 + (r_{\ell}/r)^2 + 1.21\, T_{60}\, (e^{-1.31/T60} - e^{-0.48/T60})}{e^{-1.31/T60} + 10^{0.1\,(L_n - L_r)}} \right] \tag{17.45}$$

The useful-to-late ratio C_{95}^{α} can be obtained by setting the noise term in the denominator equal to zero.

A-Weighted Signal-to-Noise Ratio

Bradley (1986) also worked with a simple metric, namely the A-weighted steady-average-speech level (55 dBA at 1 m for a normal voice and 63 dBA for a raised voice in this study), based on anechoic measurements of speech. He calculated the direct plus reverberant-field level and used it to test intelligibility for various background-noise levels. The results were very similar to those found with more complicated metrics, and its ease of use makes it attractive. Figure 17.22 shows his results in terms of a signal-to-noise ratio. It is interesting to note that these data support his assertion that signal-to-noise ratios significantly less than 15 dB yield very satisfactory intelligibility.

Comparison of Speech-Intelligibility Metrics

Bradley (1986), in his comparison of several methods of predicting speech intelligibility in rooms, examined metrics in three categories: AL_{cons}, STI, and the various signal-to-noise ratios. His studies were carried out using a Fairbanks rhyme test, which gives a result similar to that obtained with nonsense syllables. He found that there was close agreement between STI and the early-to-late ratios, but poor correlation between AL_{cons} and the other metrics. Jacobs (1985), using loudspeakers of differing directivities, found a similar result with AL_{cons}, yielding errors on the order of 20% in intelligibility. In his work the use of

STI lead to a slight (5%) underprediction of intelligibility, whereas a weighted signal-to-noise ratio, similar to Eq. 17.45, yielded an overprediction of the same order of magnitude. Bistafa and Bradley (2000) found a linear relationship between STI and U_{50}

$$U_{50} \cong 31\,\mathrm{STI} - 16 \tag{17.46}$$

indicating that these metrics are essentially equivalent. A similar relation was deduced for Lochner and Burger's signal-to-noise ratio

$$U_{50}^{\alpha} \cong 1.25\,U_{50} + 3.4 \tag{17.47}$$

The research cited in this section was done with single, as opposed to distributed, loudspeakers and is best utilized in analyzing rooms with unamplified talkers or single-source reinforcement systems. The complications introduced by multiple loudspeakers with different directivity characteristics and delay times are not addressed here.

17.3 DESIGN OF ROOMS FOR SPEECH INTELLIGIBILITY

The interior design of a given room depends on the use, interior décor, and the acoustical goals for the space. In many rooms such as restaurants or private homes, the noise may be generated by conversations other than those of interest. In these cases the addition of absorbing materials can control reverberant noise but must be balanced against the interior design goals. Acousticians must be sensitive to the appearance of their work and architects must accept the fact that design is not only visual.

The Cocktail Party Effect

The cocktail party effect* is an interesting and amusing exercise in the buildup of a sound field in a room. Let us assume that we are giving a party in a relatively reverberant room and invite a number of people to come. Let us say that the room has a carpeted floor, hard walls and ceiling, and some furniture, which contribute 93 metric (1000 sq ft) sabins of absorption. Before the guests arrive two hosts are having a conversation in the living room. They are polite so that only one speaks at a time with a sound power level of 70 dB. For purposes of this calculation let us assume that the direct sound, which is transmitted between the talker (with $Q = 2$) and the listener, is the signal, and the reverberant sound reflected from the surfaces of the room is the noise. Clearly some of the reflected sound contributes to intelligibility but we are going to ignore that for this simple analysis. Using Fig. 17.23, let us say that for barely adequate (60%) intelligibility, we need a signal-to-noise ratio of at least -6 dB to understand sentences.

The reverberant field level in our living room is

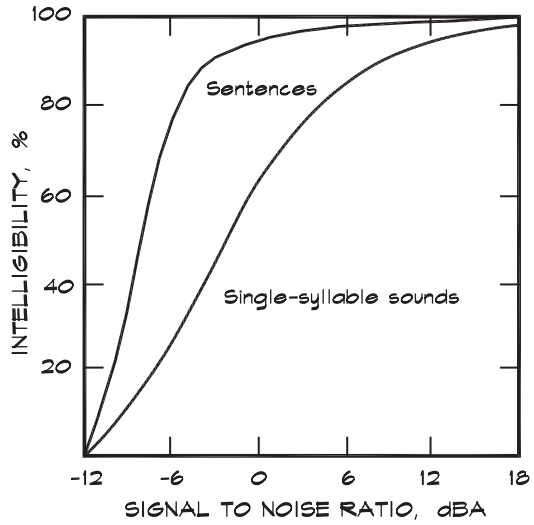
$$L_p \cong L_w + 10\,\log\,(4\,/\,R) \tag{17.48}$$

so

$$L_p = 70 + 10\,\log\,(4\,/\,93) = 56.3\,\mathrm{dB} \tag{17.49}$$

*Cocktail party effect is also used to describe our ability to understand an individual talker in the presence of a noisy combination of other conversations and background noise.

FIGURE 17.23 Percent of Words and Sentences Correctly Identified in the Presence of Background Noise (Kinsler et al., 1982)



This means that speech can be understood at a direct field level of 50.3 dB. Assuming the background noise due to other sources is low, the two people can converse comfortably at a separation distance of 3.9 m (13 ft).

Our first pair of guests arrives and the two groups begin talking, only now two people, one from each group, are talking simultaneously. The reverberant level increases by 3 dB, but the direct-field remains the same, so the minimum conversation distance drops to 2.7 m (9 ft). When two more couples arrive and pair off, the comprehension distance drops to 1.9 m (6 ft). When four more arrive the distance drops to 1.3 m (4 ft), and so forth.

In practice what happens is that people may choose not just to move closer, but also to talk louder. This raises the background noise and forces everyone to elevate their voices so at the end of the evening they all go home with sore throats—a corollary of the cocktail party effect. The point of this example is that more absorption in the room allows a higher signal-to-noise ratio and more people can talk comfortably before the increasing-volume spiral begins to kick in.

Restaurant Design

Restaurant design includes a similar problem in speech intelligibility since we want patrons to be able to talk comfortably across a table, but we do not want their conversations overheard by someone at a neighboring table. Consequently we need sufficient absorption that we do not have to raise our voices to be understood at a distance of 1 to 2 m (4 to 6 ft), but we want masking at a distance of, say, 3 m (10 ft) and beyond.

Let us imagine a restaurant that has a hard ceiling and walls and some absorption in the furniture for a total of around 20 metric sabins. A normal conversational level ($L_W = 70$ dB) will produce a direct field of 60 dB at 1.2 m (4 ft). With 20 metric sabins, our self-generated reverberant-field noise is 63 dB, our signal-to-noise ratio is -3 dB, and we achieve 75% intelligibility. If there are 20 tables in the room, with one person talking at each table, the reverberant noise level rises to 76 dB, a very uncomfortable level, and we can no longer have an intelligible conversation. This simple calculation tells us something useful—in hard-surfaced restaurants it is very difficult to have a normal conversation across a table.

People who enjoy talking to their dinner companions do not come back to these establishments and the owners ultimately suffer. Yet for some unfathomable reason countless restaurants are designed in this way.

We treat the problem by adding absorption. For example, assume that we cover the ceiling with an absorbent material. If it has an absorption coefficient of 0.9, this adds 170 metric sabins to the 13.7 × 13.7 m (45 × 45 ft) room. The 20 table reverberant noise level drops to 66 dB, which is low enough to carry on a cross-table conversation. At an adjacent table 3 m (10 ft) away, the direct field level from our conversation is about 54 dB and so it is not understandable. Off-axis directivity losses also may provide some additional isolation.

What we see from these relatively simple calculations is that unless we add absorptive treatment with an area approximately equal to the restaurant ceiling area, when the room is full of patrons, conversation across a table will be difficult and the background noise level will be uncomfortable. Second, even when we add this amount of absorption, the environment is not so dead that conversations are easily overheard at a neighboring table. More formally, these two conditions can be stated as follows.

$$L_p(\text{signal}) = L_w + 10 \log\left[\frac{Q}{4\pi r^2}\right] \tag{17.50}$$

and

$$L_p(\text{noise}) = L_w + 10 \log N + 10 \log\left[\frac{4}{N R_t}\right] \tag{17.51}$$

where N is the number of simultaneous talkers (or tables) in the room and R_t is the absorptive area per table. The signal-to-noise ratio is the difference between these two equations

$$L_{SN} = 10 \log\left[\frac{Q}{4\pi r^2}\right] + 10 \log\left[\frac{R_t}{4}\right] \tag{17.52}$$

To insure adequate communication for a cross-table distance equal to r_s we apply the condition that $L_{SN} > -6$ dB. This leads to the requirement that the amount of absorption per table in terms of the cross-table separation distance must be

$$R_t > 6.33\, r_s^2 \tag{17.53}$$

To insure privacy between tables, we apply the condition that the signal-to-noise ratio $L_{SN} < -9$ dB. This leads to the requirement that the amount of absorption per table, in terms of the separation distance r_t between tables, be limited to

$$R_t < 3.16\, r_t^2 \tag{17.54}$$

For a talker-to-listener distance of 1 m, our analysis suggests at least 6.3 or more square meters (68 sq ft) of absorption per table. If we treat the ceiling with a highly absorptive material, the minimum spacing between tables becomes about 2.5 m (8 ft), based on filling the room evenly. At that distance the maximum allowable absorption from Eq. 17.54 should be no more than 20 sq m (215 sq ft) per table. Normally we design based on Eq. 17.53 since the requirement in Eq. 17.54 is easily met.

Conference Rooms

Small conference rooms have become increasingly sophisticated primarily due to the audio-visual and computer interface requirements. Even with such systems in place face-to-face communication must still take place within a room, and the natural acoustical characteristics of the space are very important. Strong overhead reflections aid in cross-table communications so the ceiling above the table should be hard and flat. The area of reflective ceiling does not need to extend beyond the seating area. Outside this area the ceiling may be absorptive, diffuse, or recessed. In the central ceiling area acoustical diffusers are not particularly helpful. Above the conference table the ceiling should be low, preferably less than 3 m (10 ft) so the distance loss is minimized.

Most conference rooms are set up to have a table in the middle of the room with people seated around it. The shape of the conference table can help improve intelligibility. A lenticular shape allows people to see everyone seated at the table and also see plans and diagrams in the center. Horseshoe-shaped tables should be avoided, particularly if people are seated on both sides of the U, since they may face away from people on the other side.

Floors should be carpeted and absorption applied to the middle and upper portions of the walls in the form of cloth-wrapped panels, preferably with a tackable surface of 3 mm (1/8") dense fiberglass or cork between the cloth and the fiberglass. Reverberation times may be selected in accordance with the recommendations in Fig. 17.10.

Sound systems are often included in conference rooms if only to present recorded or transmitted material. Where there is a projection screen at one end of the room, loudspeakers should be located on either side. If there is also a speech-reinforcement system, loudspeakers are best located overhead with an electronic delay to maintain the correct impression of source direction. The loudspeaker system associated with the screen should not be used for speech reinforcement in order to minimize feedback.

Classrooms

The architectural design of a classroom begins with the seating layout, which is driven by the number of seats, code requirements, and the location of the audio-visual elements. Typical classrooms are relatively small, perhaps 25 feet wide by 30 feet deep, which will accommodate 30 to 40 students. Control of classroom noise, including exterior, mechanical, and reverberant, is of particular concern. For small classrooms, an NC 30 is an appropriate background level and noise from exterior sources such as traffic or aircraft should be limited to an L_{eq} of 35 dBA.

As a general rule reverberation times should be less than 0.8 seconds for good intelligibility. In his study of intelligibility in classrooms, Bradley maximized intelligibility as a function of reverberation time. The result appears in Fig. 17.19. He found that intelligibility was maximized at a reverberation time that depends on the signal-to-background-noise ratio in a range from 0.2 to 0.8 seconds. This requires a ceiling material of acoustical tile having an NRC of 0.8 or above for a ceiling height of between 9 and 12 feet. In small classrooms the ceiling is the only absorptive surface. When this is the case, the mid-frequency reverberation time can be estimated using the approximation

$$T_{60} \cong \frac{h_c}{20\,\overline{\alpha}_{NRC}} \tag{17.55}$$

where h_c is the ceiling height in feet and $\overline{\alpha}_{NRC}$ is the NRC value of the ceiling material. For a 10-foot ceiling, an NRC of 1.0 produces a half-second reverberation time.

Similar recommendations have been memorialized in an ANSI standard (ANSI, S12.60-2002) which sets background noise levels for spaces of less than 20,000 sq ft (283 cu m) to 35 dBA. Reverberation times for classroom volumes less than 10,000 sq ft are < 0.6 sec and for rooms having between 10,000 and 20,000 sq ft (566 cu m) are < 0.7 sec.

Small Lecture Halls

In a small lecture hall the choice of room shape is between a fan and a rectilinear form with a range of floor plans between the two. Fan-shaped rooms bring the seats closer to the front whereas a rectilinear shape provides a more frontal view of the display areas. Background levels due to HVAC systems and exterior sources should be limited to no more than an NC 30 or an L_{eq} of 35 dBA. Carpeted aisles are helpful in controlling the footfall noise due to latecomers. Automatic door closures without latches help to muffle the sounds of entry doors.

As a room grows larger the direct field should be augmented with early reflections from hard surfaces. Overhead reflections are preferred since the human ear is easily fooled as to the source direction, when the image source is located above (or below) the actual source. Lateral reflections smear the perceived source direction particularly when the reflection is louder than the direct sound. This condition occurs when there is grazing attenuation due to the presence of an audience that results in a direct sound being weaker than the reflected sound. In the case of large conference rooms, small auditoria, lecture halls, and legitimate theaters, a relatively low hard ceiling is preferable to an absorbent one.

A stepped or sloped floor, along with a raised platform for the talker, aids in the useful reflections and reduces grazing attenuation. Absorptive panels should be applied to the rear and side walls of the room to control reverberation and lateral reflections. The ceiling above the podium and the side walls surrounding the podium should be slanted (a 1:12 slope is sufficient) to avoid flutter echo.

A floor plan of a typical small lecture hall of about 120 seats is shown in Fig. 17.24. This hall is typical of several designed by the author and combines the audio-visual program with the acoustical requirements of the space. Moveable writing boards can be incorporated into the front walls along with projection screens for slides or video. The side walls at the front of the room are canted to accommodate the screens and to reduce the flutter echo from the side walls on either side of the lecturer.

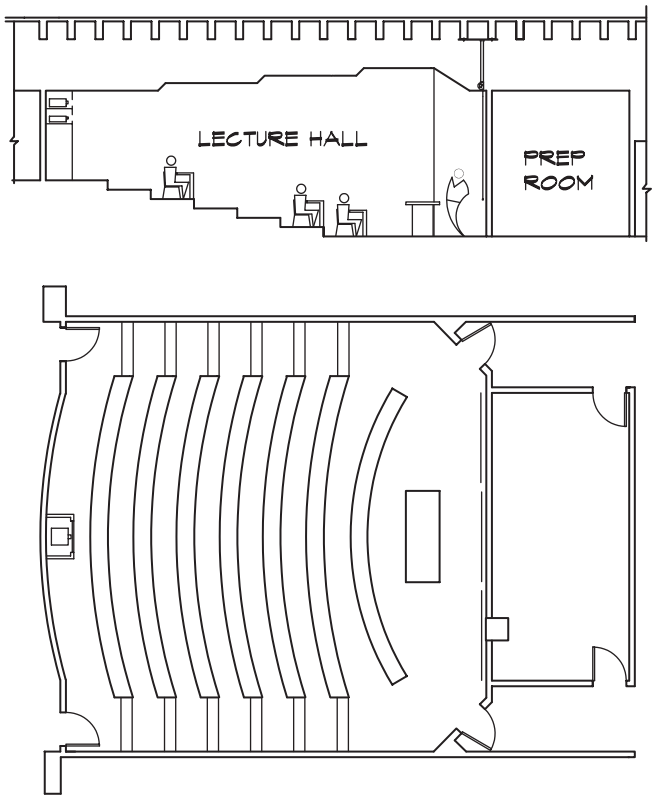
The ceiling is a series of flat-stepped elements, which provide beneficial early reflections. Flat ceiling elements are both more practical to build and better for intraclass discussions than more complicated ceiling shapes. The rear and side walls are treated with absorptive panels, which can be made tackable if classroom activities require the posting of student work.

The design of small lecture halls is increasingly influenced by the audio-visual requirements of the space. As room size increases the size of the projection screens must increase proportionately, and they tend to dominate the front surface of the room. A projection screen, which can be raised and lowered, is preferred to a fixed screen, since it discourages lecturers from writing on it, although surfaces are available that can be used for both functions.

Large Lecture Halls

In large lecture halls the design techniques are similar in principle to that of small halls. The distance from the source to the receiver should be short, which requires some widening of the seating area. Unsupported speech is not intelligible more than 30 to 40 ft (9 to 12 m)

FIGURE 17.24 Design of a Small Lecture Hall

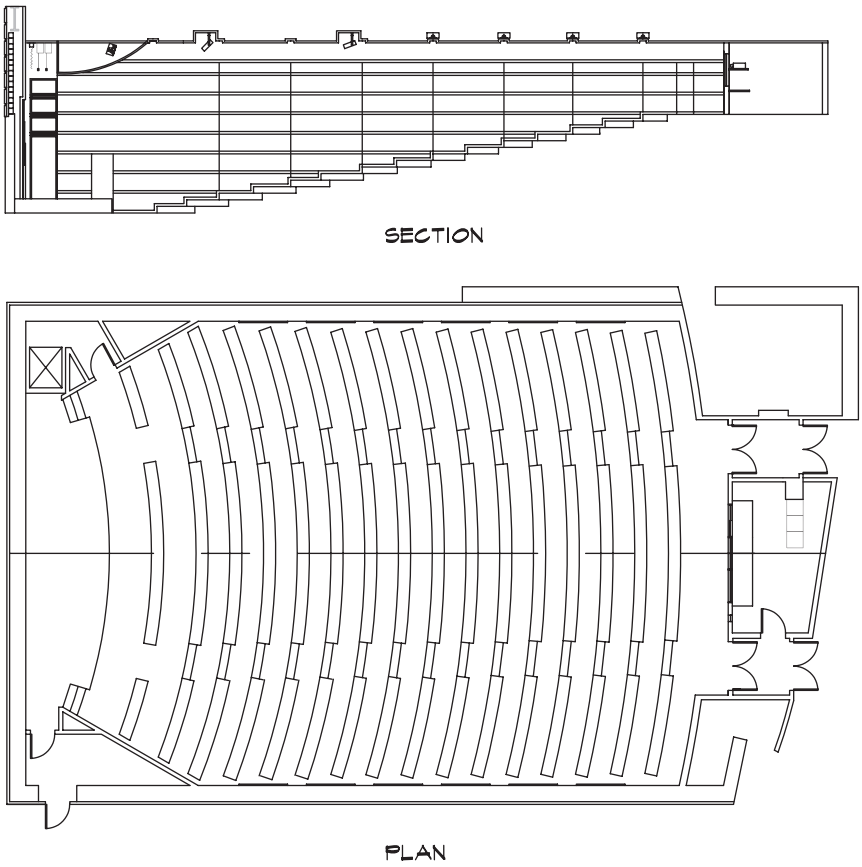


away unless considerable care is taken with the design. A fan-shaped configuration brings the audience close to the platform; however, the seating layout should be contained within a 125° maximum included angle if there is a projection screen on the front wall. The first-reflected-sound path should also be kept short. To that end the ceiling should be hard and relatively low so that the room volume (Doelle, 1972) is between 80 and 150 cu ft/seat (2.3 to 4.3 cu m/seat). Background-noise levels should be limited to no more than an NC 25 and exterior noise to an L_{eq} of 30 dBA, somewhat lower than the requirements for small halls.

A sound system should be included as part of the design and the loudspeakers should be integrated into the appearance of the room. The reverberation times can be selected from Fig. 17.10. If opera chairs are used, they should have padded seats and backs to reduce the variation in reverberation between the empty and full conditions. Reflections from the lower side walls can be helpful; however, reflections from the rear wall should be controlled with absorption.

An example of a successful lecture hall design at the Applied (Acoustics) Research Laboratory at Penn State University is shown in Fig. 17.25. This auditorium seats about 500 people and has extensive audio-visual capability. Although the room has a sound-reinforcement system, amplification is unnecessary, due to the drywall ceiling, but convenient for most lecturers. Loudspeakers are located behind a curved perforated metal screen as well as in the ceiling in the rear half of the seating area. The lower portion of the screen is backed with clear plastic to provide an overhead reflecting surface, which has the same appearance as the absorptive portion. Even though a flat ceiling yields good results when

FIGURE 17.25 Design of a Large Lecture Hall
Penn State Applied Research Laboratory (Acoustical Engineer, Marshall Long Acoustics) (Architect, The King Lindquist Partnership)

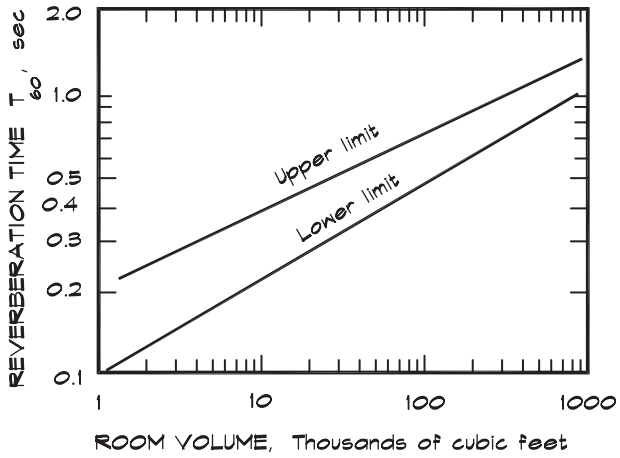


the seating is raked, a shaped ceiling is necessary in a room having a flat or shallow-angled floor. Lecture halls seating more than about 100 people should be designed with a sound reinforcement system and a hall having any type of projection or audio playback system needs to have sound reinforcement. In council chambers or courtrooms there may be a need for a recording system or for simultaneous translations. In these cases all talkers are miked and both loudspeakers and headphone feeds should be provided. The specifics of sound reinforcement design are discussed in Chapt. 18.

17.4 MOTION PICTURE THEATERS

Although motion pictures include speech and music, the design of movie theaters is driven by speech intelligibility considerations rather than by the need to provide reverberant support for unamplified music. The theater itself is an important link in the production chain since a film, as a mass-produced entertainment medium, is most effective if it is viewed in a controlled environment that yields the same auditory experience for every patron. Not all movie theaters are the same but they should be designed to achieve a consistent listening environment.

FIGURE 17.26 SMPTE Standard Reverberation Time vs Room Volume (SMPTE, 1989)



Reverberation Times

By and large, motion picture theaters are built to be acoustically dead, with absorptive material on virtually every surface except the floor, which must be washable. Ceilings are dark-colored acoustical tile and the side and rear walls are covered with minimum 1" thick cloth wrapped fiberglass panels or heavy pleated drapes. Curved rear walls should include 6" of fiberglass batt behind the panels to reduce focusing.

Recommended standards have been issued by SMPTE (Society of Motion Picture and Television Engineers) and by THX, a private company founded by George Lucas, on the preferred background noise levels, reverberation times, and sound system equalization curves. Motion picture theaters are designed to an NC 30 background-noise level and to the reverberation times shown in Fig. 17.26. THX recommends a minimum transmission loss rating (STC 65) for walls separating theaters, as well as a list of approved sound system components.

18 SOUND REINFORCEMENT SYSTEMS

Electronic sound reinforcement systems are a critical part of achieving good speech intelligibility in large rooms. When the room volume is greater than about 15,000 cu ft (425 cu m), or the seating capacity more than 100 to 150 seats, or the talker-to-listener distance greater than about 40 feet (12 m), a sound reinforcement system is necessary, even if there are no auxiliary inputs. There are many uses of electroacoustic systems beyond speech reinforcement including playback of prerecorded or transmitted material, distribution to the hearing disabled, simultaneous translation, sound effects including surround sound and bass effects in motion picture theaters, artificial reverberation, synthesized music such as electronic organs, paging, intercom, and recording. At lower seating capacities, playback requirements alone may dictate the need for a system. In this chapter, however, we will concentrate primarily on the speech reinforcement function.

The guidelines for a well-designed sound system are similar to those for general speech intelligibility:

1. There must be loudness appropriate for the program material.
2. The coverage must be uniform both spatially and over the frequency range of interest.
3. There must be adequate gain before the onset of unstable feedback.
4. The illusion that the sound is originating from the original source location should be preserved.
5. The sound system must not cause or contribute to acoustical defects such as echo, comb filtering, or long delayed reflections.
6. The sound system design should be sensitive to the architecture.

18.1 LOUDSPEAKER SYSTEMS

Loudspeaker Types

Loudspeakers in sound reinforcement systems are designed with different goals than those used in the home. Reinforcement loudspeakers emphasize efficiency, directivity control,

power handling capacity, and durability. By contrast, home listening systems stress wide-coverage angles, stereo imaging, clarity, and fidelity of reproduction. For critical listening, home systems, particularly the better quality systems, are much more accurate and much better sounding than speech reinforcement systems.

Reinforcement loudspeakers are either cones or horns or some combination of the two. Cone loudspeakers, which were discussed in Chapt. 6, are lightweight paper, metal, or composite diaphragms attached to a coil of wire immersed in a permanent magnetic field. For cone loudspeakers the greater the cone diameter and excursion, the higher the maximum sound pressure level. Large cone diameters, however, yield narrow coverage angles. Most cone loudspeakers, which span the speech frequency range and have a wide dispersion pattern, are between 5" and 8" in diameter. Large cones (having diameters greater than 10") are not used as full-range devices, but are employed in combination with a high-frequency transducer, utilizing a crossover to divide the signal at a convenient point. Coaxial loudspeakers, which include a high-frequency radiator positioned along the center axis, allow a large cone to be used for the low frequencies while maintaining the advantage of a small high-frequency transducer in a collinear position.

A horn loudspeaker, also discussed in Chapt. 6, consists of a moving coil driver coupled to a (2–4") metal or carbon-fiber diaphragm, which is coupled through a phasing plug, to the throat (usually 1", 1.5", or 2" diameter) of an expanding horn. Horns are more efficient and have a higher directivity than cones. The size of the horn's mouth opening controls the low-frequency limit of directional control. This is the most critical parameter in sound system design since it determines the visual impact as well as the degree of feedback control, and thus the gain before feedback, for a given system.

Sometimes horn and cone loudspeakers are combined into a single enclosure as in Fig. 18.1 or arranged as separate components as in Fig. 18.2. The scarcity of choices makes the commercial design problem simple when the choices are appropriate to the application but difficult when the choices are not. When the choice is between a cone and a horn for a given application, at the risk of over simplification, cone loudspeakers sound better and horn loudspeakers are louder and provide more directional control. Horns can be made to sound better by carefully matching the driver to the horn and by exercising care in the shape of the horn. Cones can be made somewhat more efficient by grouping them and can be made more directional by using line arrays over a limited bandwidth. Obviously the quality of the sound produced by an individual transducer affects the overall quality of the sound system. Cones sound better than horns since with horns there is a complex interaction between the driver and the horn, which colors the sound. This is particularly true of the small-mouth horns.

Manufacturers offer two- and three-way systems in enclosures, with the advantage of being able to match and align components so that the tonal balance is good. The disadvantage is that, due to size constraints, the directional control provided by a prepackaged box is not as good as with a large-format horn. Cabinets are available, which are trapezoidal in plan so that they can be arranged along a radius of curvature as shown in Fig. 18.3. This distributes the high-frequency components, which do not overlap, but creates a mid-frequency line array that narrows the beam width in the plane of the curve.

Loudness

Manufacturers publish data on the sensitivity of their products, along with their maximum power handling capacity and sometimes the maximum level at 1 meter from the face of

FIGURE 18.1 A Cabinet Loudspeaker

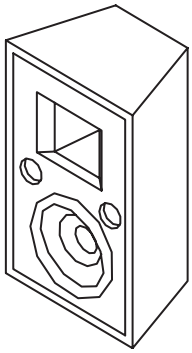


FIGURE 18.2 A Component Loudspeaker System

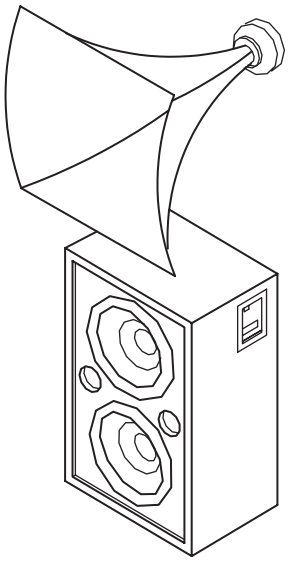


FIGURE 18.3 A Clustered Cabinet Loudspeaker System

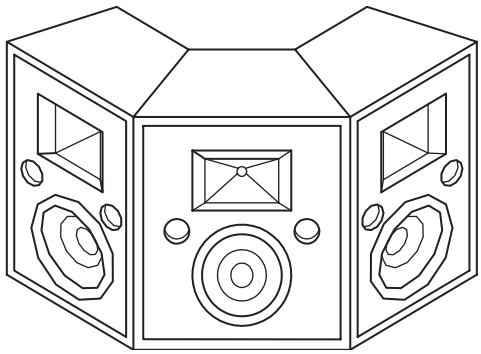


TABLE 18.1 Maximum Sound Pressure Levels for Various Uses

Type of Use	Maximum Level (dBA)
Classroom	**80**
Lecture Hall	**85**
Synagogue	**85**
Home Theater	**90**
Legitimate Theater	**90**
Church, Traditional	**90**
Movie Theater	**100**
Church, Fundamentalist	**100**
Sports Arena	**105**
Rock Concert	**110–115**

the device. These data can be used to determine the highest direct-field level achievable. It is important to establish the maximum sound pressure level required in a given venue and Table 18.1 can be helpful in this regard.

The maximum sound pressure level required by use determines the power handling capacity of the sound system and the corresponding size of the amplification system. It should be remembered that overdesign is expensive but underdesign can be fatal, particularly in live entertainment venues. Thus amplifier headroom based on the peak-to-average ratio should be included.

Bandwidth

Although less critical than loudness, bandwidth is also an important concern particularly if high levels of bass are required. For speech reinforcement systems the bandwidth ranges from about 100 Hz to 4 k Hz. For paging it might require only 200 Hz to 3000 Hz. The bandwidth of the telephone system extends from 300 Hz to 3500 Hz. A refined audiophile might demand the full range of human hearing from 20 Hz to 20 k Hz from his stereo system. As a general guideline some apply the so-called "rule of 400,000," which states that the product of the upper- and lower-frequency limits of an audio system should be equal to 400,000. This is not a physical law but more of an audio urban legend.

When assembling a multiway loudspeaker system using separate components, the bandwidth of each driver needs to be matched to its adjacent unit. Table 18.2 gives the bandwidth of the most frequently encountered loudspeakers without tuned enclosures. A carefully designed enclosure can extend the low-frequency limit by about an octave.

In full range two-way systems a dual 15" midrange/bass cabinet can be matched with a 1.5" to 2" compression driver and horn combination. For speech systems a 1" driver and horn with a dual 12" cone midrange yields better high-frequency performance. A three-way system might consist of two 15" bass loudspeakers, one or more 10" midrange cone drivers, and a high-frequency 1" compression driver and horn combination. Midrange devices can also be coupled to horns to achieve both directional control and increased efficiency.

TABLE 18.2 Bandwidth of Various Loudspeakers

Type of Loudspeaker	Bandwidth (Hz)
1" Compression Driver	1200 – 16 k
1.5" Compression Driver	1000 – 12 k
2" Compression Driver	850 – 10 k
1" Dome Tweeter	2000 – 20 k
3" Cone Driver	700 – 10 k
4" Cone Driver	500 – 7 k
5" Cone Driver	350 – 6 k
8" Cone Driver	250 – 5 k
10" Cone Driver	200 – 4k
12" Cone Driver	150 – 2k
15" Cone Driver	80 – 1250
18" Cone Driver	70 – 1000

Low-Frequency Loudspeakers

When separate subwoofers are used as part of a sound system, the loudspeakers are most efficient if they are coupled to one or more reflecting surfaces no more than one-sixth of a wavelength away. Coupling can increase the efficiency and overall output. A bass loudspeaker is most efficient when placed on a solid floor near a wall or in the corner of a room. From a corner location the loudspeaker can excite all room modes since they all have pressure maxima at the corner.

If low-frequency drivers are located away from a wall and grouped together in a cluster, additional support for the bass cabinets can be obtained by constructing a baffle wall around them. To be effective the wall must be large enough to reflect all frequencies of interest and heavy enough to minimize diaphragmatic absorption.

It is particularly important in the construction of baffle walls to seal any openings between the wall and the loudspeaker cabinets. An opening around the face of the loudspeaker, along with the cavity behind, forms a Helmholtz resonator, which will selectively absorb sound at its resonant frequency. Thus if the baffle is not properly sealed, it can do more harm than good by creating notch filters in the frequency response.

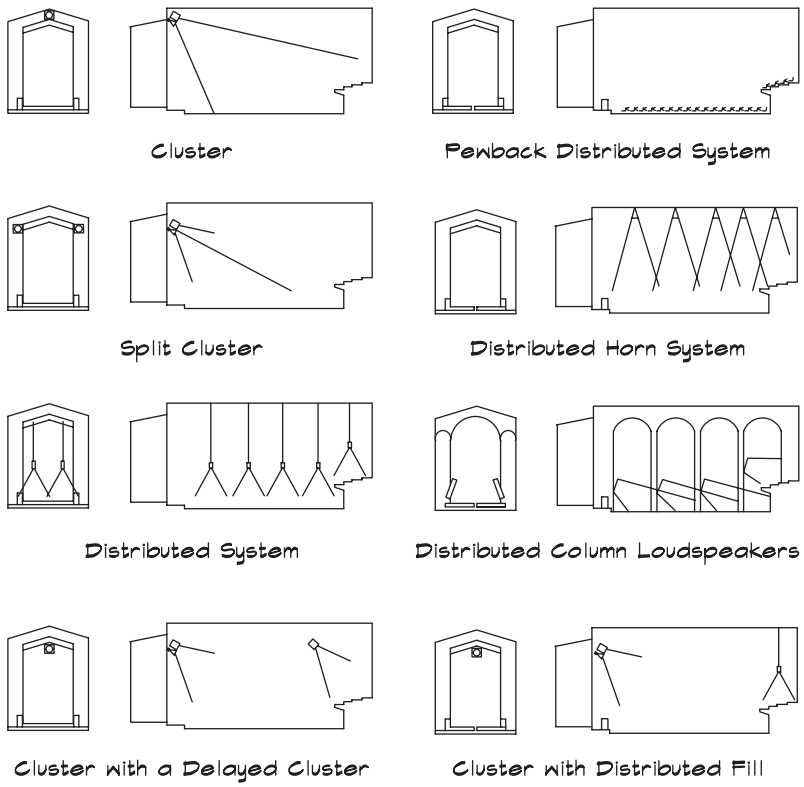
Loudspeaker Systems

A range of reinforcement systems is shown in Fig. 18.4 and includes single and multiple clusters, overhead or column-mounted distributed loudspeakers, and pewback systems. In some cases a combination of clusters, point-source and distributed loudspeakers, can be used. The choice is strongly influenced by the architecture of the space. The main design considerations include adequate signal-to-noise, maintenance of the correct image, sufficient gain before feedback, and architectural sensitivity.

Distributed Loudspeaker Systems

A distributed loudspeaker system is one in which the loudspeakers are located more or less evenly throughout the space, most often in the ceiling. The direct-field coverage can be

FIGURE 18.4 Types of Sound Systems (Klepper, 1999)



calculated; usually, for cone loudspeakers, a spacing equal to the ceiling height yields a variation of less than one dB for seated listeners. Distributed ceiling systems with cone loudspeakers can be used for ceiling heights of up to about 25 feet (7.6 m), beyond which the direct-to-reverberant ratio begins to suffer. In reverberant spaces with high ceilings, a system utilizing enclosed loudspeakers suspended in a horizontal plane can be employed, although these can be highly visible. Sometimes loudspeakers can be incorporated into light fixtures to improve their appearance, but this limits the choice to relatively small transducers. In a reverberant space such as a large airport terminal, if the system requires only a limited bandwidth, highly directional horns can be used in a distributed grid at a greater source height. In ceilings below 15 feet (4.6 m) in height, small diameter loudspeakers are preferred. A receiver standing beneath a particular loudspeaker receives the high-frequency information from the nearest source and low-frequency information from several sources. Thus even though a 5-inch (13 cm) loudspeaker may not have particularly impressive low-frequency performance, when combined with the others, it produces quite an acceptable bass output. As the ceiling height increases, the number of loudspeakers can be reduced and the aggregate contribution to the bass drops. Therefore larger transducers, having better bass response, should be used in high ceiling areas.

Distributed loudspeakers include individual transformers that increase their electrical impedance, so that several can be driven in parallel by the same amplifier. This configuration sometimes is referred to as a 70-volt or constant-voltage system, not because the impressed voltage is necessarily 70 volts, but as a shorthand. If a 70 volt (rms) signal is applied to the

input of a transformer, the rated wattage associated with each transformer tap is delivered to the loudspeaker. A constant-voltage amplifier can drive groups of loudspeakers, but they should be the same type and have the same equalization and delay. Their combined power requirements should be less than the output capacity of the amplifier.

Ceiling-mounted distributed loudspeakers can be used with or without an electronic delay system. If a delay is used, the time-base-zero source must be sufficiently loud to provide a clear directional cue to the first delay zone. Subsequent delay zones can rely in part on the zones preceding them. In small rooms the human voice is enough to provide the initial cue, but in large rooms localizing loudspeakers are required to supplement an unamplified source.

The use of electronic delays in rooms with high ceilings is a complicated problem due to the variation in delay time over the audience area. As ceiling heights increase the delay times within the area of coverage of the loudspeaker can vary enough to result in a different perceived source direction. Hence it is preferable to use delayed overhead loudspeakers in ceilings less than 20 feet (6 m) high.

Single Clusters

A loudspeaker cluster, just as the name implies, is a group of one or more loudspeakers located above the center of focus. A central cluster is one of the most effective design solutions for covering large spaces. It can deliver considerable level to the audience, while maintaining the proper relationship between the sound system and the apparent source. In a theater or auditorium this is the area above the proscenium arch. In an arena it can be the center of the room for sports events or at the end when it is used as a performance venue.

In large rooms it is difficult to design a single cluster that will provide even coverage to the whole room over the entire frequency range. In a narrow rectangular room with the focus of interest in the center of the short side, a single cluster can be successful. If the room is relatively wide, a distributed cluster is required to obtain even coverage. In a fan-shaped theater, a convex central cluster in the horizontal plane can be used. This preserves the image of sound coming from the center of the stage, even when the listener is off axis, and maintains time coincidence. The central group can be augmented with delayed clusters or distributed loudspeakers to cover hard-to-reach areas such as seats located under a balcony.

Multiple Clusters

When a single cluster is insufficient to adequately cover a space, a distributed or multiple cluster system can be used. In a wide room, loudspeakers are positioned across the front above the stage. The advantage of this type of design is that it provides excellent intelligibility and very even coverage. The disadvantage is that the apparent source is spread across the front of the room. In a long narrow room a second set of clusters located in the ceiling is sometimes necessary to cover the rear. In this type of design the second cluster must be located far enough in front of the area of coverage that it does not shift the image overhead.

If the space does not lend itself to the installation of a central cluster, other locations must be found. Loudspeakers on either side of the stage are visually benign and can yield even coverage; however, with this solution the sound image snaps to whichever side of the center aisle one is sitting on. When the talker is on the opposite side the effect is disconcerting. This may be a small price to pay for intelligibility, but it is still a design element worth consideration.

Cross-fire solutions, where one loudspeaker aims across the sound field of another, can be used as long as at the coverage overlap points, their relative levels and time delays are well matched. Cross-firing clusters can be used to preserve the source image, where there is no convenient central location, although feedback and delay times are sometimes a problem.

Other Configurations

In long narrow spaces such as cruciform-shaped churches, it is difficult to cover the entire space from the front. Supporting loudspeakers located along the sides are useful for augmenting the areas that are hard to cover. One or more small sources, built into steps or the front of the stage, should be used to assist in pulling the image back into the center.

In some churches loudspeakers are installed on the back of each pew with low-frequency augmentation under the pews. When properly designed this type of installation can provide excellent intelligibility and imaging; due to the size limitations pewback systems are limited to speech reinforcement and are not particularly successful for music.

18.2 SOUND SYSTEM DESIGN

The actual process of designing a sound system is highly iterative. It begins with an acoustical evaluation of the space, including a calculation of the reverberation times in each frequency band. The desired maximum direct-field sound pressure level should be known as well as the frequency range, type of program, location of microphones, and the preferred appearance. Based on these factors an initial choice of a system type is made. The types of reinforcement systems fall into the general categories outlined in Fig. 18.4.

The choice of loudspeaker system is based on the goal of achieving speech intelligibility through a high direct-to-reverberant ratio at the listener. This can be done in several ways: 1) by moving the sound source close to the receiver, 2) by using a loudspeaker with a high directivity, 3) by adding absorption to the space, or 4) by employing various combinations of all three.

In highly-reverberant spaces, where the signal-to-noise ratio cannot be increased by adding absorption, intelligibility can only be controlled by increasing the directivity of the loudspeakers or by decreasing the distance between the source and the receiver. This assumes that time delays and individual reflections are not a factor, which may not always be the case. The choices can be quickly narrowed to directional horns as part of one or more clusters or a distributed system. In absorptive spaces there is a greater variety of choices since the individual loudspeakers can be farther from the receiver and less directional. In rooms with low flat ceilings a distributed system is the usual selection. In long narrow rooms such as churches with high ceilings a side fill system can be used. In proscenium theaters central clusters are preferred. The system choice is largely driven by the architecture.

The second major choice is whether there will be a house mixer and where it will be located. In small conference rooms and meeting rooms an unattended automatic mixer is sufficient. As the range of program material and the number of inputs grow larger, a manual mixer becomes increasingly important. In venues such as high schools where security is a concern the mixer can be positioned in a technical booth at the rear of the auditorium with a large operable window. In legitimate theaters and concert venues where there is control over access to the facility and where the auditory experience is of the highest concern, a house mixing location in the seating area is very important. Locations may vary, however a position at the front of the balcony has many advantages including unobstructed sight lines

TABLE 18.3 Direct-to-Reverberant Levels for Speech Intelligibility in Rooms

Direct-to-Reverberant Level (dB)	Intelligibility
> –3	Excellent
–3 to –6	Very Good
–6 to –9	Good
–9 to –12	Fair
–12 to –15	Poor
< –15	Very Poor

when the audience is standing. Where there is only orchestra seating, a position in the rear of the seating area allows the mixer to stand or sit on a drafting chair to obtain a clear view of the stage.

Coverage

Once an approach has been selected, a calculation is undertaken to determine the direct sound pressure level throughout the room. Commercial computer programs are available to do these calculations or programs can be written. The design process tends to be iterative—moving loudspeakers about and determining the effect of different combinations. In a good design the predicted direct-field level should have a standard deviation of less than 2 dB in the 500 to 2k Hz octave bands throughout the seating area.

When clusters are used, the higher the elevation of the loudspeakers, the smoother the coverage but the lower the signal-to-noise ratio. For large horns, clusters should be higher than 20 feet (6 m) but lower than about 35 feet (10.6 m) above the receiver plane to achieve even coverage. For distributed ceiling systems using cone loudspeakers, the maximum practical height is around 20 to 25 feet (6 to 7.6 m). Above that height it is better to use clusters or cabinets. Cabinet systems can be very successful if there is sufficient pattern control and minimal overlap. As a general rule, the fewest number of loudspeakers possible should cover a given receiver and arrival times should be controlled so that there is no blurring or directional confusion.

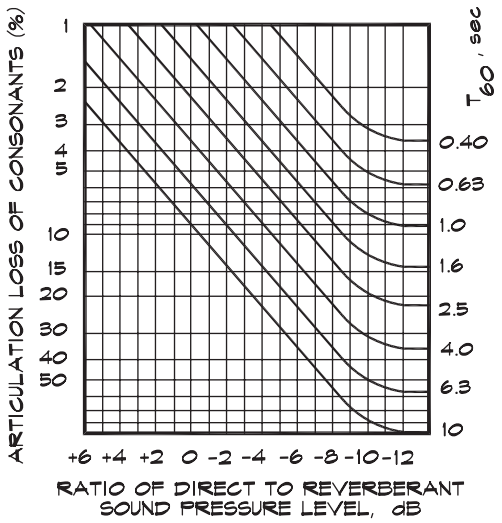
Intelligibility

Once even coverage has been achieved, the next step is to insure adequate intelligibility. The algorithms discussed in Chapt. 17 were all based on single-source tests. We have excellent techniques and well-documented results under these conditions. We have much less data on speech intelligibility in rooms having multiple loudspeakers of different types and volume settings. In part this is due to the number of variables involved.

A method used by the author for many years is to calculate a signal-to-noise ratio, where the signal is the direct-field sound pressure level from all loudspeakers, combined on an energy basis, and the noise is the reverberant-field sound pressure level. The resulting signal-to-noise ratio is compared to the table shown in Table 18.3 to determine the intelligibility.

A direct-to-reverberant ratio of 0 dB is almost never achieved indoors and even a ratio greater than –3 dB is rare. Values between –3 and –6 yield very satisfactory intelligibility. Normally it is sufficient to design a room for good intelligibility and a small number of fair

FIGURE 18.5 Alcons vs Direct-to-Reverberant Ratio (JBL, 2000)



seats, particularly at the sides and rear, are tolerable. The transition from fair to poor agrees well with Davis' (1969) limiting distance of 4 D_c for adequate intelligibility.

A chart such as that given in Table 18.3 can be very useful even though it is relatively simple. This method is similar to the useful-to-late signal-to-noise ratios used by Bradley, but uses the limiting integration time of zero. It matches up well with Fig. 17.23. A more detailed analysis would take into account early reflections from nearby surfaces and possibly a factor for the reverberation time similar to the methodologies discussed previously. In most sound system designs there is a concerted effort to point the loudspeakers at the audience and to keep reflections off of the walls and ceilings. Clearly not all the reverberant field is harmful to intelligibility, but this is accounted for in the table. As a receiver moves away from the source, and closer to side walls or other reflecting surfaces, it is more likely that the direct-field levels will be augmented by beneficial reflections and the signal-to-noise ratio will increase. Therefore receiver locations in these positions can be designed with the expectation of a bit more direct-field contribution.

Table 18.3 does not account for the noise due to other sources since sound systems can be turned up enough to overwhelm most background levels. Notable exceptions to this rule include sports stadia where crowd noise can exceed 100 dBA. The overall level must therefore be calculated separately.

Intelligibility metrics, which were not originally developed for multiple sources, have been adapted to sound system calculations. JBL (2000) issued a technical note containing equations and a graph (Fig. 18.5) for the prediction of the articulation loss of consonants in terms of direct and reverberant-field energies. The note cites work by Peutz (1971).

$$AL_{cons} = 100(10^{-2(A + BC - ABC)} + 0.15) \tag{18.1}$$

$$A = -0.32 \log \left(\frac{E_R + E_N}{10\,E_D + E_R + E_N} \right) \tag{18.2}$$

$$B = -0.32 \log \left(\frac{E_N}{10\, E_R + E_N} \right) \tag{18.3}$$

$$C = -0.5 \log \left(\frac{T_{60}}{12} \right) \tag{18.4}$$

where $E_R = 10^{0.1\, L_R}$ = reverberant - field contribution
$E_D = 10^{0.1\, L_D}$ = direct - field contribution
$E_N = 10^{0.1\, L_N}$ = noise - field contribution

Note that these terms are not simple energies but normalized intensity ratios, which are proportional to energy. This formulation is just the liveness (Eq. 17.12); that is, a constant times the reverberant-to-direct energy ratio times the reverberation time out to a direct-to-reverberant ratio of –10 dB, after which it reverts to a constant times the reverberation time. According to JBL, it is included in popular loudspeaker design programs.

When computations indicate that a loudspeaker design will not result in adequate intelligibility, the signal-to-noise ratio must be improved. There are only a few ways of doing this, namely to decrease the reverberant field or to increase the direct field. The reverberant field can be decreased by adding absorption to the room or by decreasing the sound power level of the loudspeakers. The direct field can be increased by using more directional loudspeakers or by moving the loudspeakers closer to the receiver. If highly directional loudspeakers are needed, the face area of these loudspeakers is large, and this will have architectural ramifications.

Amplifier Power Handling

The sensitivity, along with the power handling capacity, determines the maximum output level available from a loudspeaker. Horns, because they confine the energy to a narrow beam, have a higher sensitivity than cones. For example, a 1-inch driver attached to a $90^\circ \times 40^\circ$ horn might have a sensitivity of 102 dB at 1000 Hz. Small cones tend to be more sensitive than large cones. An 8" cone driver has a sensitivity of about 96 dB, whereas for a 15" cone it might be 90 dB. The decreased sensitivity of large cones is offset by the fact that they will handle a higher input power.

In designing high-powered systems it is important to match the maximum output of each component so as not to waste power or overdesign in a given frequency range. If the maximum output level for a given cabinet size is the design goal, as is often the case in road systems, then component matching is very important. Amplifiers should be sized large enough to handle the maximum power that a transducer is likely to encounter. If the low-frequency input signal drives the amplifier into clipping, high-frequency components necessary to generate the clipped wave shape are produced. These components can be passed through a crossover and overdrive the high-frequency unit. Digital distortion can also generate some nasty waveforms, which are not necessarily stopped by digital limiters within the same processor system.

To obtain the amplifier power required for a given level at a receiver we use

$$10 \log J = L_p(r,\ 0,\ 0) - L_s + 20 \log r + 2\, \Delta\, L_\sigma + 10 \tag{18.5}$$

where $L_p(r, 0, 0)$ = on - axis direct - field level produced by the loudspeaker at the aim point (dB)
L_s = loudspeaker sensitivity (dB re 1 W at 1 m)
J = electrical power applied to the loudspeaker (W)
r = distance between the loudspeaker and the receiver (m)
ΔL_σ = standard deviation of the direct - field level (dB)

The factor of 10 at the end of Eq. 18.5 represents the peak-to-average ratio for uncompressed speech since amplifiers must have enough power to handle the peaks. The factor for the standard deviation accounts for the lower off-axis level.

Electrical Power Requirements

The electrical power required by the amplifier is based on its output power, efficiency, and duty cycle. The rated output power is available from the amplifier manufacturer in the published specifications. It will depend on the load impedance, with lower impedances drawing the greater power. The amplifier efficiency is the ratio of the electrical energy, powering the loudspeakers, to the amount of electrical power supplied to the amplifier. Most A-B amplifiers are about 65% efficient. Some, however, can be as high as 95% efficient depending on design. The duty cycle is a measure of the peak-to-average ratio of the input signal over time. Amplifiers must be sized to provide an output power high enough to handle the maximum signal level, but that power is not always required. The duty cycle for various types of source material is given in Table 18.4.

Using the duty cycle data, the electrical power requirements can be determined

$$J_e = \frac{N_c\, J\, \xi}{\eta} + \frac{N_c\, J_q}{2} \tag{18.6}$$

where J_e = electrical power from the AC main (W)
J = rated amplifier output power for one channel (W)
J_q = quiescent power for zero input voltage (W)
$\cong$ 90 W (typical value)
ξ = duty cycle
η = amplifier efficiency
N_c = number of amplifier channels (assuming 2 channels per amplifier)

TABLE 18.4 Duty Cycles for Various Sources (Crown, 1995)

Source Material	Duty Cycle
Pink Noise	**50%**
Highly Compressed Rock	**40%**
Rock Music	**30%**
Background Music	**20%**
Continuous Speech	**10%**
Infrequent Short Duration Paging	**1%**

To convert the power draw into current,

$$I_e = \frac{J_e}{V_e\, f} \tag{18.7}$$

where I_e = electrical current from the AC main (A)
V_e = electrical voltage from the AC main (V)
f = power factor = 0.83

The power factor accounts for the difference in phase between the voltage and current.

Heat Load

Electronic equipment is mounted in standard rack frames located in an equipment room. Cool air must be supplied to these rooms in sufficient quantities to stabilize the room temperature so the equipment does not overheat. The heat to be removed is the waste heat generated by the electronics. The heat load for processing and line-level devices can be calculated directly from their power requirements since no power leaves the room. For amplifiers it is based on the power that is not delivered to the loudspeakers (Crown, 1993).

$$\frac{dQ}{dt} = K_h \left[\frac{N_c\, J\, \xi(1-\eta)}{\eta} + \frac{N_c\, J_q}{2} \right] \tag{18.8}$$

where Q = heat load in BTU or Joules
K_h = 0.3415 for BTU / hr
= 360 for J / hr

Heat is removed through convective cooling, whereby hot air is exhausted from the room and is replaced by refrigerated air. The rate of heat removal must match the rate of heat generation for temperature stability. The thermodynamic model of the process assumes that the air in the equipment room is heated to a temperature ΔT above the temperature of the incoming air and is then carried off through the return air system. The heat transferred to the air is

$$Q = m\, C_p\, \Delta T \tag{18.9}$$

where m = mass of air (kg or lb_m)

C_p = heat capacity of air at constant pressure
= 1006 J / kg °C

= 0.240 BTU / lb_m °F
ΔT = temperature rise (°C or °F)

The total heat removed in terms of the mass of moving air is

$$\frac{dQ}{dt} = \frac{dm}{dt} C_p\, \Delta T \tag{18.10}$$

and the required amount of cooling air for a given temperature rise and heat load is

$$\frac{dV}{dt} = \frac{1}{\rho_0\, C_p\, \Delta T}\ \frac{dQ}{dt} \tag{18.11}$$

where $\frac{dV}{d\,t}$ = volume flow of air (cu m / hr or cu ft / hr)
ρ_0 = density of air
= 1.21 kg / cu m
= 0.075 lb_m / cu ft

This yields

$$\frac{dV}{d\,t} = \frac{K_h}{\rho_0\, C_p\, \Delta T}\left[\frac{N_c\, J\, \xi(1-\eta)}{\eta} + \frac{N_c\, J_q}{2}\right] \tag{18.12}$$

For a 1000 Watt amplifier (total combined power for both channels) this works out to about 16 cfm for a 30% duty cycle and a 5° F temperature rise. For six amplifiers in a rack, the cooling requirements would be on the order of 100 cfm. It should be noted that the temperature rise is that of the air in the room, not necessarily that of the amplifiers. The amplifiers may be at a different temperature depending on the local heat transfer within the racks and the point of delivery of the refrigerated air, which depends on fans in the rack or in the amplifiers themselves.

Time Coincidence

When two sound sources are used to cover the same receiver, any physical separation between them yields a difference in arrival time and a pattern of cancellation at certain frequencies, just as it did with an image source discussed in Chapt. 7. As more sources are added, the interference pattern becomes more complex and fills in the gaps in the spectrum created by the original pair. Using this line of reasoning, Davis and Davis (1987) argue for more loudspeakers covering a given area as long as the time delays between individual loudspeakers are not excessive. Comb filtering is seldom a problem in distributed loudspeaker systems, where multiple sources cover the same area, since it is a steady-state phenomenon. Speech varies in time so rapidly that comb filtering is rarely audible.

If a cluster is used, only one or two loudspeakers will cover a given area. Arrival time differences due to loudspeaker displacements of a few inches can be perceived as a lack of speech clarity. Here it is important that the sounds from adjacent units arrive at the listener at approximately the same time, since small misalignments are more audible than large ones. For a system such as that pictured in Fig. 18.6, time coincidence can be assured on axis by aligning the drivers. If the driver diaphragms of each loudspeaker are physically aligned then the wavefronts will be coincident along the centerline. In home hi-fi systems, where the principal listener is located on the centerline, correct alignment results in a marked improvement in imaging and clarity.

An off-axis listener does not receive the same benefit from this technique since the sound paths are of different lengths. Though seldom a problem in home systems, it is an important design parameter in sound-reinforcement systems. With speech-reinforcement systems, particularly those that include horns, the physical alignment of the drivers is inconvenient and not sufficient to guarantee time alignment, because the path length difference at an off-axis receiver is not the same as that for an on-axis receiver. In these cases the faces of the individual cone loudspeakers and horns should be aligned and an electronic delay should be used to ensure arrival time coincidence, as illustrated in Fig. 18.7. Although this scheme is an excellent technique, time coincidence cannot be guaranteed for every direction, since the off-axis path lengths still vary somewhat for each component. In a two-way system where

FIGURE 18.6 Two-Way Loudspeaker with Aligned Diaphragms

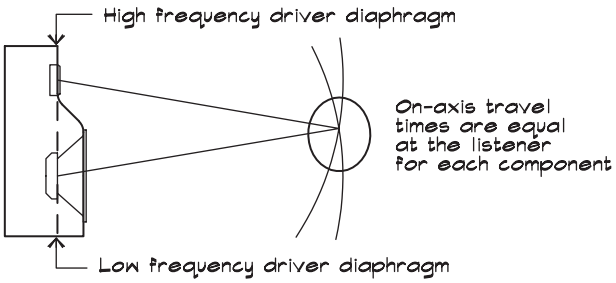
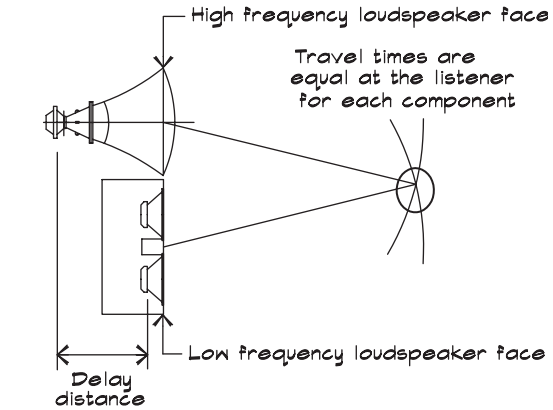


FIGURE 18.7 Two Component Loudspeaker System with Aligned Faces and an Electronic Delay



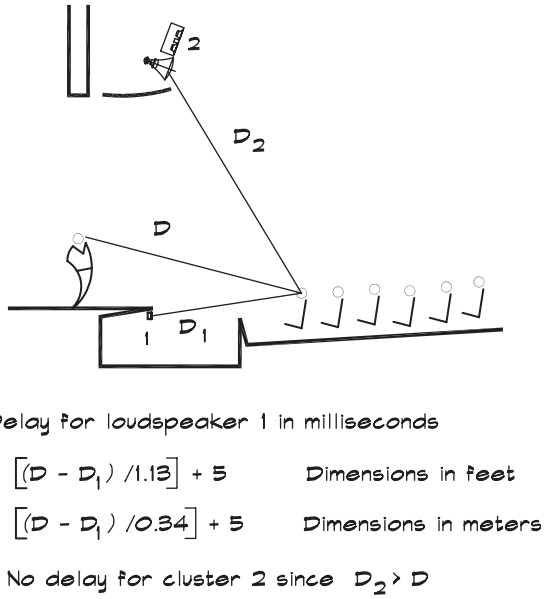
time overlap is a concern only near the crossover frequency, differences in arrival time can change the character of the sound.

Sometimes timing delay differences are introduced on purpose to achieve directivity control. In the midrange, where cone loudspeakers are used, vertical stacking of components leads to a narrowing of the coverage angle in the vertical plane. This occurs even when trapezoidal boxes with splayed sides are arrayed in a line. Just because packaged enclosures point in different directions does not mean that they do not interact in the manner of a typical line source.

Imaging

In a well-designed audio system the sound appears to be coming from the talker. This is accomplished by using the law of the first arrival, which states that the sound that reaches a listener first fixes the perceived source direction unless its level is too low. An example is pictured in Fig. 18.8. When the first-arriving sound comes from the talker or from a loudspeaker placed near him, the illusion is maintained. Since our ears are in the horizontal plane we are more sensitive to differences in azimuth than in elevation. For example, a central cluster located in front of a proscenium arch above the stage can maintain the illusion that the sound is coming from a performer on stage if the performer's voice is strong enough

FIGURE 18.8 Delay Settings for Cluster Systems

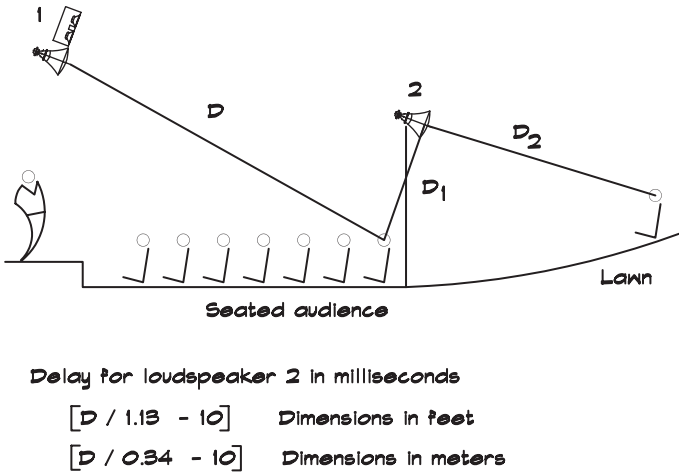


to provide the zero delay signal. Small loudspeakers located along the front of the stage can help pull the image down but they may have to be turned off when the pit orchestra is miked. The audio signal is fed into the main cluster, which may be delayed so that it arrives 5 to 10 msec later than the live sound. In the illustration, the cluster is located in front of the talker so that no additional delay is required. The actual delay setting depends on the position of the performer on stage and is a compromise since there is a range of performer locations.

In theme parks, where a live show might be combined with on-stage effects, point-source loudspeakers are positioned near an animatronic character or effect to generate the first arrival sound. This maintains the overall level and frequency balance between the live and recorded tracks and provides the physical illusion. Point-source loudspeakers do not need to have the full frequency range to preserve this imaging and can be small, as long as they produce an adequate level. If point sources are used with live microphones, the live mics must be muted during the effects to prevent feedback. In theme parks, this is done by using a show control computer to duck or gate the live mics. In live theater the ducking can be controlled either manually or automatically.

If a distributed system is used, the signal feeding the overhead loudspeakers is progressively delayed so that the illusion is maintained. Some source, either a live talker or a centrally located loudspeaker, must be used to produce the time-base-zero signal. The number of delay zones required for any given system depends on the details of the design; loudspeakers, which are no more than 10 msec apart in time delay, can be grouped into a zone. It is preferable in these systems to have only a few loudspeakers covering a given area. Thus this technique is most effective in low (< 6 m) ceiling rooms where the loudspeakers are close to the receivers. The overhead loudspeakers in the zones close to the talker help maintain the illusion for the zones farther away.

FIGURE 18.9 Delay Settings for a Lawn System



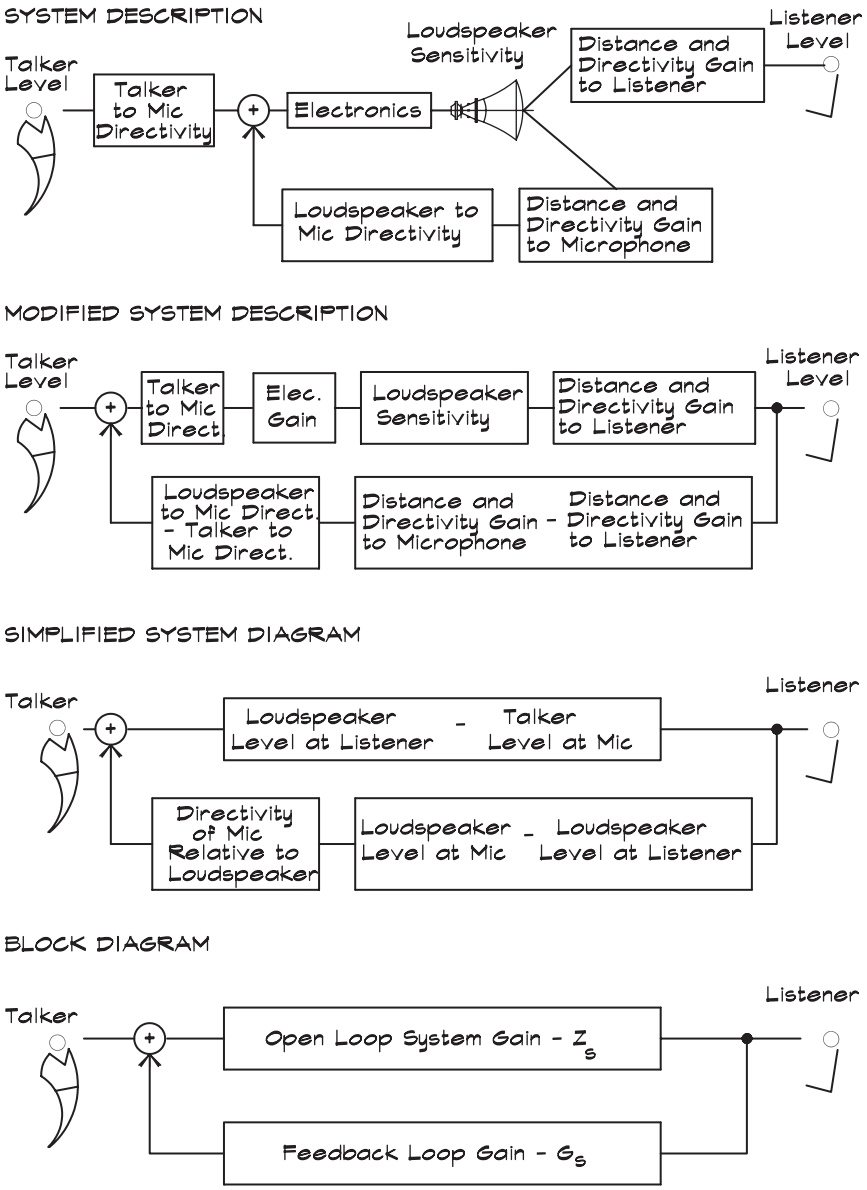
Outdoor amphitheaters often have a sound system with loudspeakers flown above a stage, with a series of loudspeakers located on poles around the rear of the fixed seating. These systems are designed to allow flexibility in servicing audiences of different sizes, with overflow patrons seated on a lawn behind the fixed seats. Setting time delays for a multiple-cluster system is difficult, particularly when the delayed cluster is elevated. In the configuration illustrated in Fig. 18.9, there are different delay times for listeners located under and behind the delayed loudspeaker. Normally the delays are set so that the time-base-zero loudspeakers lead the delayed loudspeakers by about 5–10 msec so that the illusion that the sound is coming from the stage is preserved. In this example the delayed loudspeakers are located on a tower 20 feet (6 m) in the air. If the time delay is picked to be 10 msec after the arrival of the time-base-zero signal, the sound heard at the base of the tower will lag the sound from the stage by about 30 msec and create a noticeable upward image shift. In tower systems the arrival time for the delayed loudspeakers should be set so as to lead the stage loudspeakers by about 10 msec. Then at the base of the tower the overhead sound lags the stage sound by 10 msec and the illusion is preserved. Farther back on the lawn the patrons are in line with the tower loudspeakers and even though they now lead the stage loudspeakers, the directional illusion still is preserved.

Feedback

The control of feedback and the achievement of adequate gain before the feedback becomes unstable are probably the most difficult aspects of sound system design. Figure 18.10 shows the gain structure of a sound reinforcement system. Note that gain and loss are used interchangeably. The open-loop system gain $Z_s = 20 \log z_s$ is defined as the difference between the level produced by the loudspeaker at the listener and the level produced by the talker at the microphone. The multiplier factor is 20 rather than 10 because z_s is a voltage gain. Clearly the gain must be high enough to produce an adequate level in the audience.

When there is a path for the amplified sound to travel back to the microphone, it has a transfer gain (or loss) $G_s = 20 \log g_s$, which characterizes the feedback loop. In electrical engineering terms any closed loop produces feedback, which can be stable or unstable. The transfer function for the signal moving through the entire closed loop system (with feedback)

FIGURE 18.10 Sound System Gain Block Diagrams



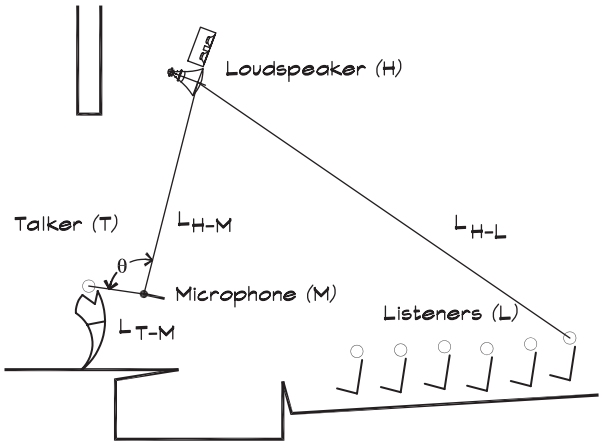
is the infinite series

$$X_S = z_s + z_s\,(z_s\,g_s) + z_s\,(z_s\,g_s)^2 + z_s\,(z_s\,g_s)^3 + \cdots \tag{18.13}$$

which has a value in the limit of

$$X_S = \frac{z_s}{1 - z_s\,g_s} \tag{18.14}$$

FIGURE 18.11 Parameters for Feedback Control



This is unstable (infinite) when the denominator goes to zero or when

$$z_s\, g_s = 1 \tag{18.15}$$

If the transfer functions are expressed in decibels (taking 20 log of Eq. 18.15) the criterion for sustained unstable feedback is

$$Z_s + G_s = 0 \tag{18.16}$$

Using Figs. 18.10 and 18.11 we can write out the gains in terms of their components. The open-loop system gain is the difference between the direct field level at an average receiver in the audience area and the direct-field level produced at the microphone by the talker

$$Z_S = L_{H\text{-}L} - L_{T\text{-}M} \tag{18.17}$$

where $L_{H\text{-}L}$ = direct field level produced by the loudspeaker system (H) at the listener (L), (dB)

$L_{T\text{-}M}$ = direct field level produced by the talker (T) at the microphone (M), (dB)

Z_S = system gain necessary to produce a given level at the listener relative to the talker level at the microphone (dB)

For a typical sound system the desired open-loop system gain might be on the order of −6 dB, which means that a comfortable communication level is achieved at a talker-to-listener distance that is twice the talker-to-microphone distance. In a rock concert, where the desired receiver sound pressure level is high, the system gain is increased as much as possible and the performer-to-microphone distance is made very short, normally less than one inch.

The feedback loop transfer function from Fig. 18.10 is

$$G_s = L_{H\text{-}M} - L_{H\text{-}L} + D_M(\theta) \tag{18.18}$$

where L_{H-M} = direct - field level produced by the loudspeaker system at the microphone, (dB)
$D_M(\theta)$ = directivity index of the microphone in the direction of the loudspeaker relative to the talker (usually negative), (dB)

We can substitute the levels into Eq. 18.16 to obtain the condition for sustained unstable feedback,

$$Z_S + L_{H-M} = L_{H-L} - D_M(\theta) \tag{18.19}$$

Now if we wish to have a stable system we must reduce the gain by a margin of safety called the feedback margin of stability, which is taken by some authors (Snow, 1955; Davis and Davis, 1987) to be between 12 dB for an unequalized system and 6 dB for a carefully equalized system. This text uses a middle value of 10 dB for an equalized system. In practice if a tone from the loudspeaker is sustained it may combine in phase at some frequency with the sound reflected from a hard surface yielding a 6 dB increase in the level seen by the microphone. The extra 4 dB is a safety factor. Thus we would expect the system to be stable if

$$Z_S + L_{H-M} \leq L_{H-L} - D_M(\theta) - 10 \tag{18.20}$$

For an open loop gain of -6 dB, typical of an auditorium or church, we obtain the condition for system stability with a closed feedback loop

$$L_{H-M} \leq L_{H-L} - D_M(\theta) - 4 \tag{18.21}$$

which states that for feedback prevention the direct-sound level at an omnidirectional microphone ($D_M(\theta) = 0$) should be 4 dB below the average direct level in the audience. With a cardioid microphone, having -2 or -3 dB of relative directivity, the stability criterion is

$$L_{H-M} \leq L_{H-L} - 2 \tag{18.22}$$

and the level at the microphone can rise to just under the average level in the receiving space. Although this formula is very simple, it is also quite useful since it is not explicitly dependent on the type, number, or characteristics of the loudspeaker system, only on the sound field it produces and the type and orientation of the microphone. The desired level in the audience area is determined independently and sets the maximum distance that the talker can be from the microphone. The coverage variability is also independently determined.

The sound level at the receiver can be increased either by increasing system gain or by decreasing the talker-to-microphone distance. Once the loudspeaker system has been set, the only way to increase the gain is to utilize a more directional microphone or to add electronic processing such as equalization, delay, or a frequency shifter to suppress the ring frequencies.

Note that the loop-gain equations include the directivity of the microphone. Directional microphones can be beneficial for increasing gain before feedback, but they discriminate against the talker when he is off-axis. They also emphasize sounds coming from a particular direction that may include sound scattered from nearby surfaces, including the talker's body. The precise calculation of their effectiveness is difficult since they are not always held at the same angle relative to the loudspeaker. It is prudent to incorporate a cardioid or

hypercardioid microphone into a system. Highly directional shotgun microphones are not always useful in preventing feedback since they can have complicated lobing patterns and can be sensitive to sound arriving from certain angles.

Equation 18.20 does not contain reverberant field information. That is included indirectly by using the ratios in Table 18.3 as part of the design process. The reverberant field can contribute to feedback; however, if it is uniform throughout the space as theory predicts, it should not matter where the microphone is located or in what direction it is pointed, unless it is close to a reflecting surface. Since the microphone location and orientation are found to be important in controlling feedback, the direct-field level is more useful as a design tool.

Multiple Open Microphones

When there are several microphones in use at the same time, the signal from the loudspeaker can be picked up by each of them and combined at the mixer. This increases the level of the signal being fed back and it is customary to add a factor into the gain equation to account for this effect. The standard adjustment for the number of open microphones (nom) is

$$\Delta L_{nom} = 10 \log N_m \tag{18.23}$$

where N_m is the number of microphones simultaneously in use. Although Eq. 18.23 is a useful generalization, it is based on the assumption that all microphones contribute equally to system gain. This requires that they be the same type, that their level and equalization settings all be the same, that they all have the same physical relationship to the loudspeakers, and that none of the signals they receive be in phase with those falling on other microphones. This is almost never the case in practice. It is much more likely that one or two microphones will contribute more than the rest and be responsible for the initiation of feedback. Nevertheless it has become accepted practice to insert the nom correction into the gain equation (Eq. 18.20) since there are too many unknowns associated with the exact multiple microphone calculation.

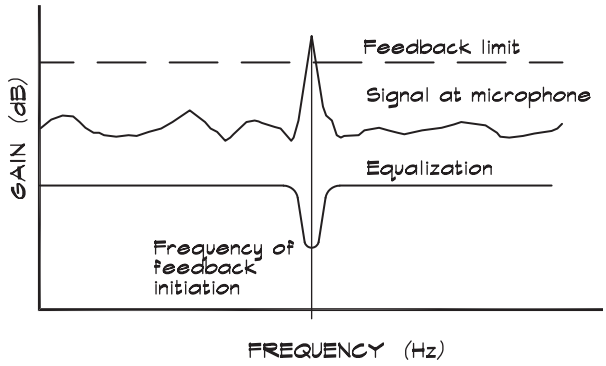
$$Z_S + L_{H-M} + \Delta L_{nom} \leq L_{H-L} - D_M(\theta) - 10 \tag{18.24}$$

It is good practice in speech systems, where there is no live mixer, to use an automatic microphone mixer, so that the nom correction is zero. These devices adjust the combined output level according to the number of microphones receiving signal, so that the system gain is theoretically the same as it would be if only one microphone were on at a given time. In music systems automatic mixers are not employed (with some exceptions) and the number of open microphones correction must be applied. We frequently encounter multiple open microphones in churches, where separate microphones are used to pick up the sound from the choir. Separately adjustable loudspeakers in the vicinity of the music microphones can help maintain a reasonable gain structure.

Equalization

An equalizer is an electronic device that allows the addition or subtraction of gain within a frequency range that is selectable by the user. Filters come in several flavors: constant bandwidth (rarely encountered); constant percentage bandwidth (such as octave or third octave); and parametric where the frequency, bandwidth, and level are all adjustable.For overall system equalization a third octave bandwidth filter is used. Parametric filters can be used where precise control is desired; however in the hands of an inexperienced user, parametric filters can be tricky.

FIGURE 18.12 Effect of Equalization on Feedback



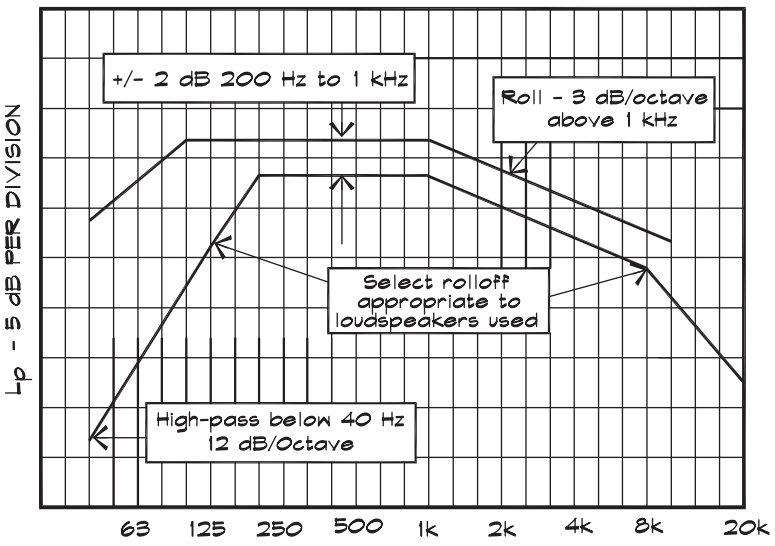
Equalizers are used to control feedback as well as even out the frequency response of a sound reinforcement system. Figure 18.12 gives an example. Since feedback begins at one particular frequency, a narrow-band filter can be used to reduce the level at the feedback frequency so that the overall gain can be increased before the instability begins. System designers debate the relative merits of adjustable-bandwidth filters versus (third-octave) constant-bandwidth filters. Either parametric or graphic equalizers can do a satisfactory job of controlling ring modes. A constant-bandwidth filter has the advantage of covering all possible frequencies within a range and providing a knowledge of where a filter is set. If digital processors are used, parametric filters require less processing real estate than the constant-bandwidth type.

Once a sound system is installed and the delay times set, it can be ensonified using a pink noise generator, which produces a random signal having equal energy per octave across the audio spectrum. In well-designed rooms, a pink noise source should yield even coverage throughout the room. Levels should vary no more than $\pm$ 2 dB, although in large rooms or in rooms having an unusual shape greater variation is sometimes encountered. A house equalization curve is introduced, which depends on the specific material to be presented. Figure 18.13 shows several examples. For speech reinforcement the equalization rolls off at about 3 dB/octave above 1 kHz.

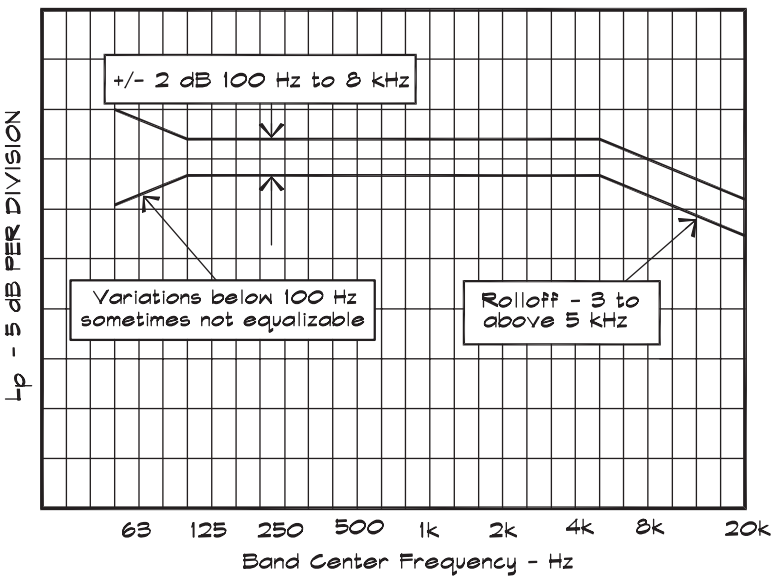
Every designer has a favorite way of doing final equalization. Some find it helpful to listen to a well-known piece of music or a familiar voice to judge the accuracy of the system. Others prefer narrow-band analyzers, which can separate out the direct-field components by time windowing. Equalization for maximum gain is done with a live microphone in its normal location. The system gain is increased until the onset of feedback occurs. A spectrum analyzer, positioned in the space, is used to read off the frequencies of the rapidly rising levels from which the equalizer can be adjusted to notch out the worst offenders. When there are different types of microphones or even microphones in different places, it is useful to use a separate equalizer for each one. Even a relatively small change in microphone location, on the order of a few feet, can sometimes shift the feedback frequencies enough to warrant resetting the equalizer. Once a feedback mode has been identified the equalizer is used to reduce the level in that frequency band until the ringing subsides. The gain is then increased until feedback occurs again and the process is repeated until there is sufficient gain in the system.

At the end of the process there needs to be a reality check on the shape of the equalization curve to insure that the notching is not so severe that it distorts the sound. If the settings are unusually dramatic from band to band, then there may be other things happening in the room causing the problem. Typical problems to avoid include loudspeaker misalignment, background or mechanical system noise, mechanical coupling to the microphone, and room modes. None of these is best treated through equalization.

FIGURE 18.13A Various Recommended Response Curves for Equalization (Foreman, 1987)

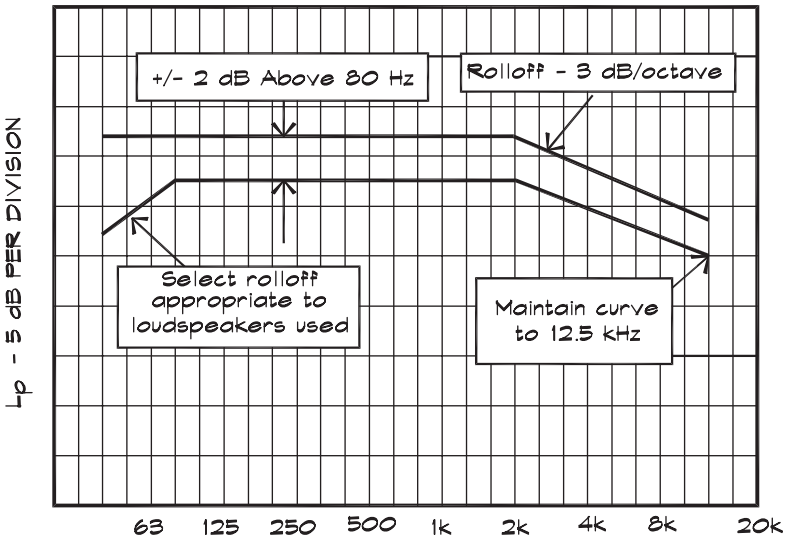


(A) Recommended response curve for speech reinforcement systems

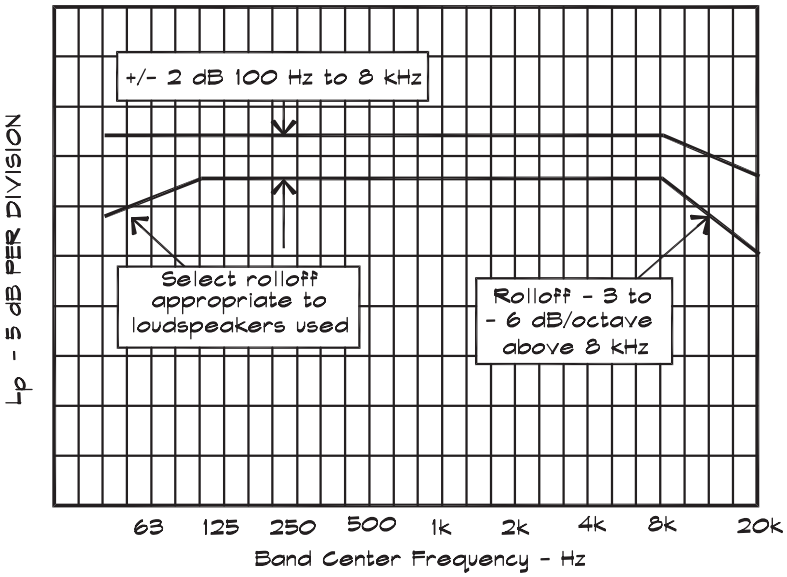


(B) Recommended response curve for studio control room monitoring systems

FIGURE 18.13B Various Recommended Response Curves for Equalization (Foreman, 1987)



(C) Standard response curve for cinema playback systems.



(D) Recommended response curve for high-level rock music reinforcement systems.

Architectural Sensitivity

Loudspeakers have the greatest architectural (visual) impact of any audio component. An early step in the sound system design process is to explore alternatives that may be used to achieve reasonable intelligibility in a space. The toolbox includes the central cluster, multiple clusters, and a combination of central cluster and delayed system, a progressively delayed system, and a distributed system. Each of these configurations has certain

advantages and disadvantages compared to the others, and each can be designed so as to provide adequate intelligibility. Not every system type, however, can be used successfully in every space.

Theaters, auditoria, churches, and concert halls all require sound systems, which can be blended into the architectural design of the space or left exposed either temporarily or permanently. Too often rooms are designed without giving consideration to the presence of the loudspeaker system that frequently must be physically large in order to provide adequate directivity control. Just as designers must accommodate mechanical and electrical systems with dedicated spaces, so too should audiovisual systems be designed into the fabric of the building, with dedicated equipment rooms and mixing locations.

Clusters can be shrouded with acoustically transparent materials such as light fabric, perforated metal screens, or wire mesh cloth. These materials should be at least 40% open to maintain acoustical transparency. Holes in solid materials must be small enough that the spacing is less than a quarter wavelength at the highest frequency of interest. This is around 0.25 inch (6 mm) apart at a maximum. They should be large enough that they are not clogged by paint, which requires a diameter greater than about 3/32 inch (2.4 mm). Hole diameters must also be larger than the thickness of the material.

Often when the loudspeaker system is not integrated, a priori, into the architecture, it is suspended as cable supported boxes or as one or more large clusters located in the center of the proscenium. This design solution may be visually acceptable; it is important to either integrate the sound system into the overall design early in the process or accept the appearance of exposed components.

18.3 CHARACTERIZATION OF TRANSDUCERS

Microphone Characterization

The ultimate goal of a sound reinforcement system is to deliver amplified speech or music in as natural sounding and intelligible form as possible. In the case of speech, which is the primary source of amplified sound, the direct-to-reverberant noise ratio determines the overall intelligibility. Thus the higher the signal-to-noise ratio of the sound gathered by the microphone, the better off we are. Even if we listen to the sound on headphones we cannot obtain a better signal-to-noise ratio than that received by the microphone, and the projection of the sound through loudspeakers can only decrease this ratio.

If we consider the reverberant sound field as the primary source of noise, the closer the microphone is to the talker and the more directional its characteristics, the higher the signal-to-noise ratio will be. From the standpoint of intelligibility, a microphone located near the source is preferable to one located farther away. Although this is true, both distance and directivity have other effects that influence the character of the sound. As we move closer to a fixed microphone, any changes in distance and head orientation can have a large influence on the level and frequency spectrum of the output signal. If the microphone is directional, the closer it is to the talker, the more severe will be the effects of head movement.

Microphones can also have an architectural impact, particularly if they are on camera. Fortunately they can be quite small with current technology. Microphones are most effective when placed close to the mouth of the user. Concert performers have come to use headset microphones, which maintain a consistent orientation and distance between the microphone and the mouth. In legitimate theater, radio microphones are located in the clothing or even the hair of the actors. In a church or synagogue the user does not have the same level of audio

sophistication or technical assistance available in a theater and choices of microphone type and location, in part, must be based on these considerations.

Omnidirectional microphones provide the most accurate reproduction of sound but are also the most prone to feedback. By using a directional microphone we can discriminate against both direct and reverberant sound coming from off-axis. A directional microphone is less likely to produce feedback, but is more sensitive to the orientation of the user's head and less natural sounding. A cardioid pattern most often represents the best compromise between feedback control and naturalness.

Unidirectional microphones are more prone to vibration and handling noise than are omnidirectional microphones. This is because the off-axis discrimination is obtained by allowing some of the signal to enter at the rear of the microphone, which decreases the net pressure on the diaphragm. At low frequencies the pressure differential, which causes the diaphragm to move, is small compared to the absolute sound pressure. Consequently the diaphragm of a unidirectional microphone must be made to move more easily in response to low-frequency sound, accomplished by reducing the damping resistance to about one-tenth that used in an omnidirectional microphone. These microphones are most sensitive to vibrations at about 150 Hz, the mechanical resonant frequency of the diaphragm and voicecoil (Ballou, 1987).

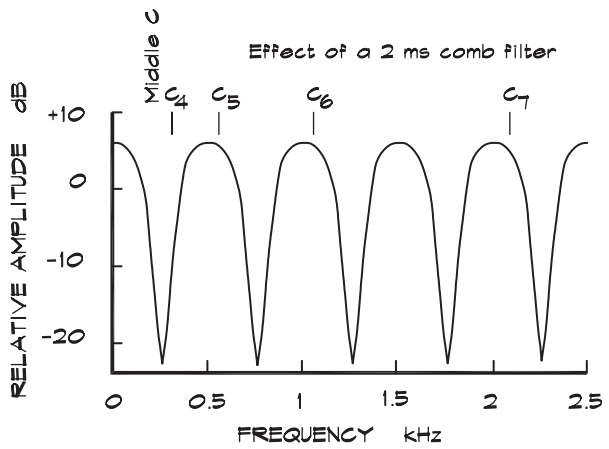
Microphones can introduce sonic artifacts due to their position and orientation. If an omnidirectional microphone is placed near a reflecting surface the signal received will include both the direct and reflected sound. The path-length difference between the direct and reflected paths will produce a series of cancellations (shown in Fig. 18.14) called comb filters which can color a steady sound.

The comb filter frequencies are given by a half wavelength path-length difference

$$f_n = \frac{(2n - 1)\, c}{2(d_2 - d_1)} \tag{18.25}$$

where n is a positive integer. If the path length difference is small, the first frequency is high and the reflection acts as a low pass filter shown in Fig. 6.16. Microphones can be placed near the floor to take advantage of this effect, but they are farther from the talker and more prone to

FIGURE 18.14 Comb Filters Plotted on a Linear Scale (Everest, 1994)



vibrational interference. Small pieces of foam located under the case can be used as vibration isolators. For large path length differences, the filter frequencies are closer together and the reflected levels are not as great as the direct level. When a microphone discriminates against off-axis sounds, the comb filtering effects are even less for the same reason. For directional microphones located on a floor-mounted stand, comb filtering is rarely a problem. Podium microphones can be positioned so that the talker's voice is not reflected directly into the capsule and these effects are minimized.

Some microphones take advantage of the reflection from a planar surface. These microphones are located close to a solid boundary where the reflected sound is in phase with the direct sound. They are generally omnidirectional, with one or two exceptions, but can be used with reflecting shields to gain some directivity. The amount of shielding is dependent on the physical size of the barrier. For small shields there is little barrier effect below a given frequency. Surface-mounted microphones placed on a table or podium are farther away from the talker than extension mounted microphones and will pick up the rustling of papers and vibrations transmitted through the mount.

In selecting a microphone several factors must be considered:

1) Directivity
2) Coloration, or changes in response with frequency
3) Ruggedness and maintainability
4) Freedom from shock and handling noise
5) Accuracy and fidelity
6) Electrical characteristics: sensitivity, phantom power requirements, impedance
7) Resistance to feedback
8) Maximum input level

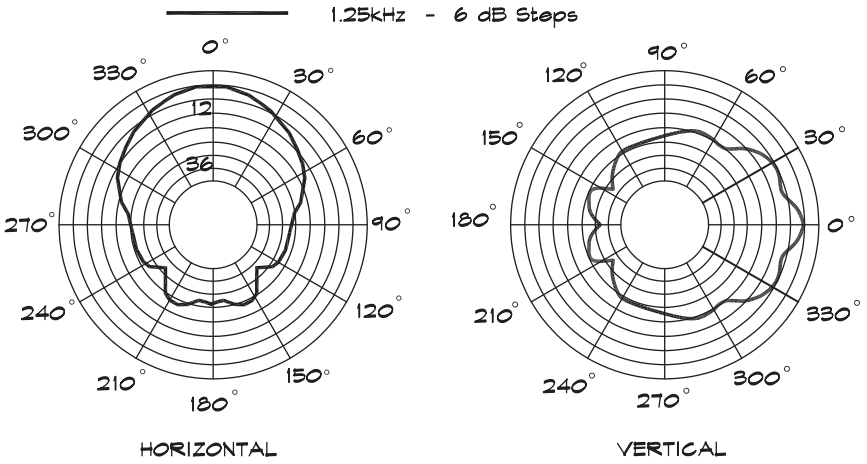
Directivity patterns are available from the manufacturer and should be studied for frequency response and lobbing. The better microphones can be handled without generating excessive noise. An external vibration isolated support is sometimes helpful, but these can be cumbersome and unattractive in sound reinforcement systems. Internal vibration isolation, which is available on some microphones, is useful, as are hum-bucker coils to reduce low-frequency hum. If phantom power is required to power internal circuitry or to provide a polarizing voltage, it is supplied by the primary mixer, which applies a DC voltage (usually from +9 to +52 volts) equally to pins 2 and 3 of the preamplifier input receptacle.

Microphones typically produce a high impedance output signal at a level of a few millivolts, which is commonly referred to as mic level. These signals may be conducted in a shielded cable over a distance of a few hundred feet without appreciable loss. Where longer distances are involved a preamplifier, with a low output impedance, can be located near the microphone to drive the line.

In a critical application, a system can be designed with a certain microphone in mind, but it is difficult to predict which microphone will sound best in a space. It is helpful to try out a number of possible microphones in the finished space and to listen to each one, preferably with the person who will ultimately be using it.

In theaters and frequently in churches a small lavaliere microphone is used either on a long cord or attached to a radio transmitter carried by the user. The alternative is one or more fixed microphones located at the principal speaking locations. Each of these solutions has

FIGURE 18.15 Horizontal and Vertical Polar Data (JBL, 1997)



advantages and disadvantages. Wireless microphones need to have their batteries periodically replaced and some have connectors that are fragile and prone to breakage. Fixed microphones are more robust but more visible and require some microphone discipline by the user.

Loudspeaker Characterization

Loudspeakers are characterized by a few simple parameters: 1) on-axis sensitivity described in Chapt. 2, 2) on-axis directivity (Q_0) and relative directivity index as a function of angle, 3) power handling capability, and 4) frequency response.

Directivity is critical to accurate system design and most manufacturers publish directivity information in terms of the difference between the on-axis level and the level for a given direction. Until recently manufacturers listed directivity data only in the form of horizontal and vertical polars as shown in Fig. 18.15, and Q_0 versus frequency data as shown in Fig. 18.16.

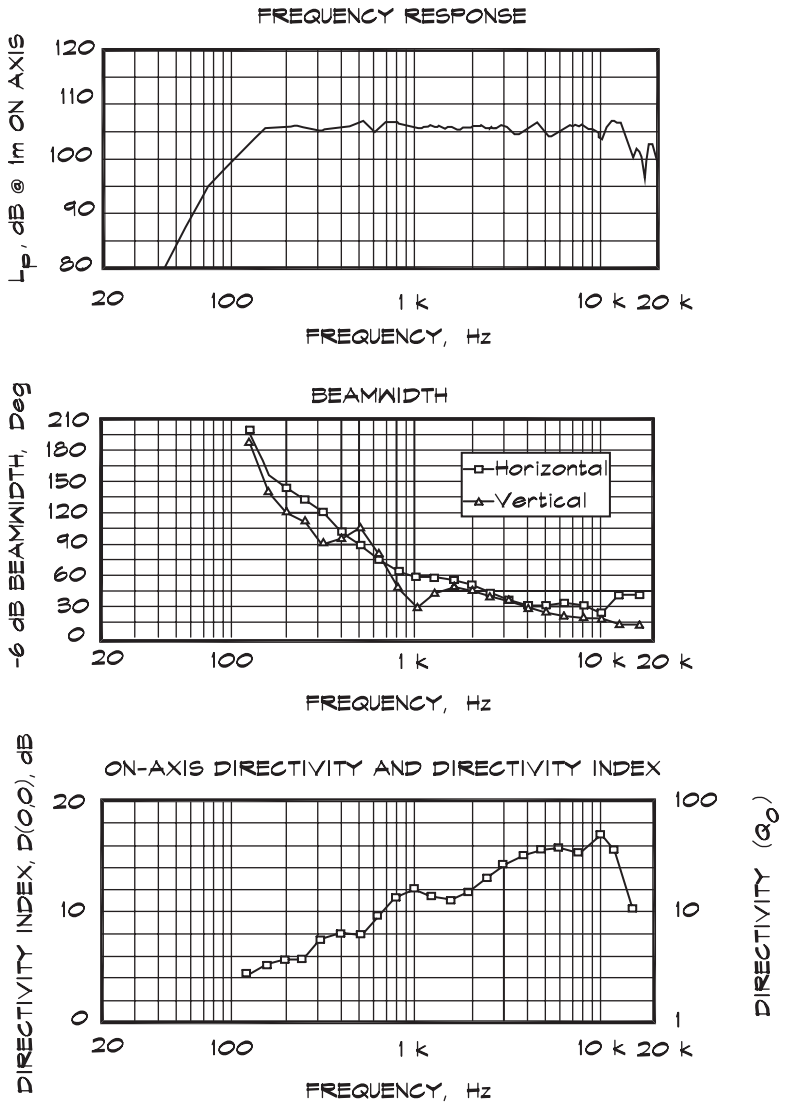
$$D_{rel}(\theta,\phi) \equiv L_p(0,0) - L_p(\theta,\phi) \tag{18.26}$$

where $D_{rel}(\theta, \phi)$ = the directivity index relative to on - axis, (dB)
taken to be positive for $L_p(0, 0) > L_p(\theta, \phi)$
$L_p(0, 0)$ = the on - axis sound pressure level at a given distance, (dB)
$L_p(\theta, \phi)$ = the sound level at a given angle for the same distance, (dB)
θ = latitude referenced to the loudspeaker centerline (deg)
ϕ = longitude from the horizontal loudspeaker plane (deg)

Since the on-axis level is almost always higher than off-axis levels, $D_{rel}(\theta,\phi)$ is generally positive.

With the development of computer modeling programs for loudspeaker coverage calculations, the horizontal and vertical polars have given way to data published in 10° increments, and there is movement toward finer increments. The normal geometry used in standard computer modeling programs is shown in Fig. 18.17.

FIGURE 18.16 Data for a Three-Way High Directivity Loudspeaker System (JBL, 1997)



All the major modeling programs use the loudspeaker centerline as the x axis (North Pole) and measure latitude relative to it. Different programs use different axes for Greenwich, the prime meridian from which longitude is measured. These choices are subsumed in the internal coding of the program; however, in generating directivity data exchange files the starting axis must be known. The CADP program measures its longitude counterclockwise beginning from the z (vertical) axis. The EASE program and the author's program, on which this text is based, use the y axis (horizontal) as the reference line.

Given the older horizontal and vertical polar data, whose coordinates are shown in Fig. 18.18, the relative directivities based on longitude and latitude around a sphere can be calculated using an interpolation formula. There is no standard method of doing this since there is no unique relationship between the polar data and the off-axis results. An approximate

FIGURE 18.17 Loudspeaker Coordinate System

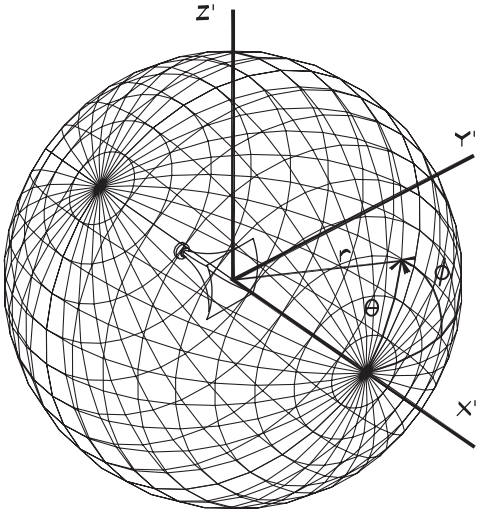
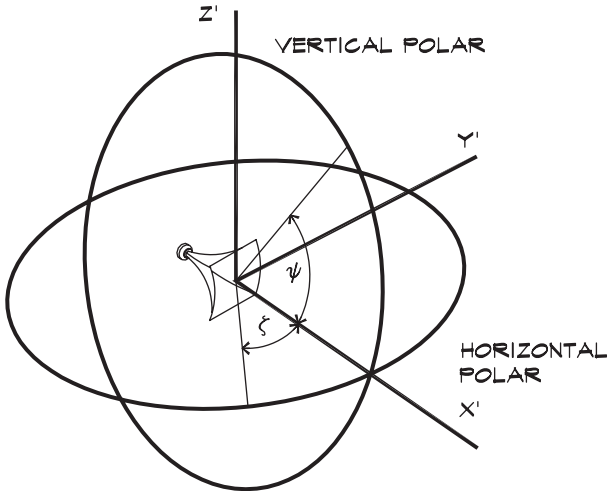


FIGURE 18.18 Horizontal and Vertical Polar Coordinates

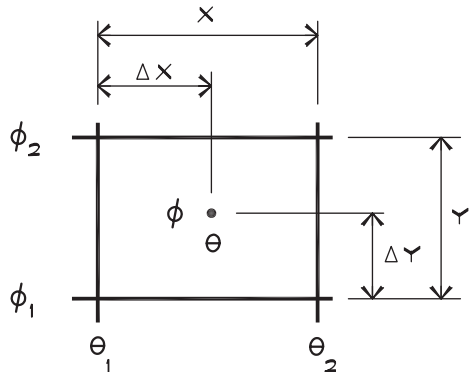


formula that can be used is

$$D_{rel}(\theta_n,\phi_n) \cong \left[\begin{array}{l} \left(D_{vert}(\psi_n = \theta_n)\sin\phi_n\right)^2 \\ +\left(D_{horiz}(\zeta_n = \theta_n)\cos\phi_n\right)^2 \end{array} \right]^{1/2} \tag{18.27}$$

where D_{rel} = the directivity index relative to on-axis for a given direction (taken to be positive for $L_p(0, 0) > L_p(\theta, \phi)$)
D_{vert} = the vertical directivity index relative to on-axis
D_{horiz} = the horizontal directivity index relative to on-axis

FIGURE 18.19 Interpolation of Data between Known Points



ς_n = polar angle in the horizontal plane in 10° steps
ψ_n = polar angle in the vertical plane in 10° steps
θ_n = latitude referenced to the speaker centerline in 10° steps
(note that $\theta_n = \alpha_n = \beta_n$)
ϕ_n = longitude from the horizontal plane in 10° steps

The vertical component of D_{rel} is zero for a vertical angle of zero, that is in the horizontal plane, and the same is true for the horizontal component of D_{rel} values in the vertical plane. The difficulty with Eq. 18.27 and other similar equations is that the real off-axis directivity values are not directly dependent on the horizontal and vertical polar values. This particular equation generates a smooth elliptical coverage pattern when viewed from the front, but does not include the complex lobing that may be present in a given direction, but may not be reflected in the polar data. Thus these types of relations are only a rough approximation of the actual off-axis behavior, but can be useful when only polar data are available.

Most manufacturers now publish the directivity index relative to the on-axis level at 10° intervals measured on a sphere of receiver locations surrounding the source. When the actual angle to a receiver falls between measured points, such as those shown in Fig. 18.19, a linear interpolation must be performed.

$$\begin{aligned} D_{rel}(\theta,\phi) = {} & D_{rel}(\theta_1,\phi_1)\left(\frac{x-\Delta x}{x}\right)\left(\frac{\Delta y}{y}\right) \\ & + D_{rel}(\theta_1,\phi_2)\left(\frac{\Delta x}{x}\right)\left(\frac{\Delta y}{y}\right) \\ & + D_{rel}(\theta_2,\phi_1)\left(\frac{x-\Delta x}{x}\right)\left(\frac{y-\Delta y}{y}\right) \\ & + D_{rel}(\theta_2,\phi_2)\left(\frac{\Delta x}{x}\right)\left(\frac{y-\Delta y}{y}\right) \end{aligned} \tag{18.28}$$

where θ_n = latitude referenced to the speaker centerline in 10° steps
ϕ_n = longitude from the horizontal plane in 10° steps

$x = \theta_2 - \theta_1$ = angular increment between measurements
$\Delta x = \theta - \theta_1$ = angular increment between receiver direction and the least measured angle
$y = \phi_2 - \phi_1$ = angular increment between measurements
$\Delta y = \phi - \phi_1$ = angular increment between receiver direction and the least measured angle

The Calculation of the On-axis Directivity

Loudspeaker sensitivity was defined in Eq. 2.83 and is the on-axis sound pressure level at one meter for a one Watt input power. Since the measurements are done on-axis, the relationship between the sensitivity and the sound power level is a function of Q_0, the on-axis Q. Once measured directivity data are available in 10° increments the directivity for a particular intermediate angle can be calculated. Recall from Chapt. 2 that $Q(\theta,\phi)$ was defined in terms of the on-axis directivity and a relative directivity.

$$Q(\theta,\phi) = Q_0\, Q_{rel}(\theta,\phi) \tag{18.29}$$

where θ = latitude relative to on - axis
ϕ = longitude relative to an arbitrary plane

The directivity Q, is defined as the ratio of the intensity in a given direction to the average intensity over the whole receiver sphere. To calculate the intensity for a given direction we must know the relative level for that particular direction. The average intensity is determined from the off-axis levels and their corresponding areas in each direction.

We begin with the on-axis directivity (Q_0), which is defined as

$$Q_0 = Q(0, 0) = \frac{I(0, 0)}{I_{ave}} \tag{18.30}$$

where

$$I_{ave} = \sum_{i=1}^{N} I_i(\theta, \phi) S_i(\theta, \phi) \tag{18.31}$$

and $S_i(\theta,\phi)$ = fractional surface area of a sphere associated with a given pair of angles θ and ϕ

Now the intensity in a given direction can also be written in terms of the directivity index relative to on-axis.

Published loudspeaker directivity indices are given relative to the on-axis directivity index, which is zero.

$$D_{rel}(\theta, \phi) = -10 \log Q_{rel}(\theta, \phi) = -10 \log \frac{I(\theta, \phi)}{I(0, 0)} \tag{18.32}$$

Thus if I_{ave} is calculated in terms of the relative directivity data,

$$I_{ave} = \sum_{i=1}^{N} I\,(0, 0)\; Q_{rel,i}(\theta, \phi)\; S_i(\theta,\phi) \tag{18.33}$$

$$I_{ave} = I\,(0, 0)\; \overline{Q}_{rel} \tag{18.34}$$

where

$$\overline{Q}_{rel} = \sum_{i=1}^{N} Q_{rel,i}(\theta, \phi)\; S_i(\theta,\phi) \tag{18.35}$$

Using Eqs. 18.30 and 18.34 it follows that

$$Q_0 = \frac{1}{\overline{Q}_{rel}} \tag{18.36}$$

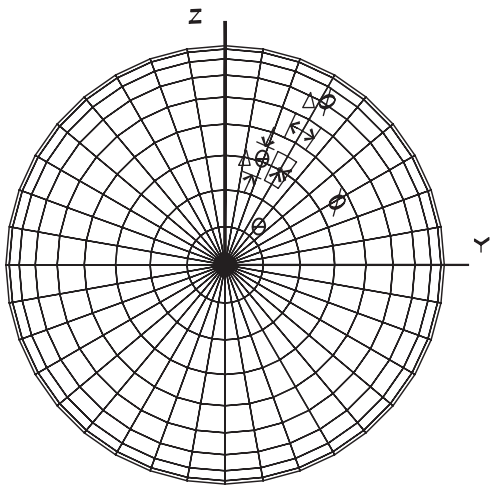
In making this calculation we subdivide a sphere into area segments. The geometry of the area of a segment of a sphere is shown in Fig. 18.20. We must then find the area of a segment associated with a given pair of angles as a fraction of the total area. For the geometry shown, the fractional area is

$$S_i(\theta,\phi) = \left(\frac{\phi_2 - \phi_1}{360}\right)\left(\frac{\cos\theta_1 - \cos\theta_2}{2}\right) \tag{18.37}$$

When the increments between angles are $\Delta\theta$ and $\Delta\phi$ (usually 10°), Eq. 18.37 can be written as

$$S_i(\theta,\phi) = \left(\frac{\Delta\phi}{360}\right)\left(\frac{\cos\,(\theta - \Delta\theta/2) - \cos\,(\theta + \Delta\theta/2)}{2}\right) \tag{18.38}$$

FIGURE 18.20 Area Associated with a Directivity Data Point



for $0 \geq \theta \geq 180$ and $(\theta - \Delta\theta/2) \geq 0$ and $(\theta + \Delta\theta/2) \leq 180$. The angle θ is incremented in 10° steps. Using these equations we can calculate the average intensity, the on-axis directivity, and from Eq. 2.83, the sound power level for a given loudspeaker using its sensitivity at each frequency.

18.4 COMPUTER MODELING OF SOUND SYSTEMS

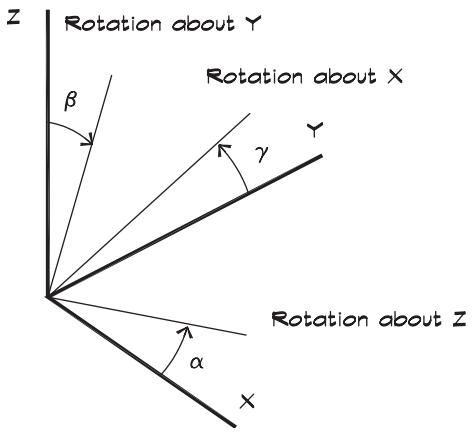
Computer modeling programs are available and can be used to predict the coverage of loudspeakers individually or in combination. When they first appeared, these programs were developed, or purchased, principally by loudspeaker manufacturers as a marketing tool to encourage the use of their products. In their earliest versions programs offered by one manufacturer did not include data on another manufacturer's devices. As other programs became available, users demanded a selection of loudspeakers from more than one manufacturer, so most programs began to include a wider range of components. When the market coalesced around a few modeling programs, manufacturers found it to their advantage to provide designers with directivity and other data on all products.

Computer programs make it possible to do rather complicated calculations of coverage and intelligibility without a detailed knowledge of the mathematics involved. This shifts the decision-making process from the sound system designer to the computer programmer on whether the coverage, intelligibility, and gain before feedback will be satisfactory for a particular design. Unfortunately vendors may not provide the user with the algorithms used in generating the code, so there is no way of back-checking the conclusions offered by the program.

Coordinate Systems and Transformation Matrices

A sound-system modeling program must describe a three-dimensional space mathematically as seen from the viewpoint of the loudspeaker. To do so a coordinate system needs to be established that defines the space in which both the loudspeakers and receivers are located. To be consistent with computer-aided design programs, the x-y axes define the plane of the building floor and the z axis is vertical. The room coordinate system is shown in Fig. 18.21. The coordinates are right-handed with the x axis aligned with the centerline of the room.

FIGURE 18.21 Room Coordinate System



When looking from a positive point on one axis toward the origin, a 90° counterclockwise rotation will transform one axis into the following one—that is, x to y and y to −x, and so forth. Rotation angles with respect to each axis are also shown.

It is convenient to use a system of homogeneous coordinates, which contains one dimension more than the space being characterized. Thus for a three-dimensional space, a four-element vector is used, where the fourth element is a scaling parameter. In this text the scaling parameter for a point is always one. The four coordinates define a particular location in space

$$\mathbf{x} = [\mathrm{x}, \mathrm{y}, \mathrm{z}, 1] \tag{18.39}$$

which is used as a column matrix when doing calculations. The effect of a translation of the coordinate system to a new location displaced by some distance $\Delta\mathrm{x}, \Delta\mathrm{y}, \Delta\mathrm{z}$ from the origin can be produced by multiplication by a translation matrix

$$\mathbf{T}(\Delta\mathrm{x},\ \Delta\mathrm{y},\ \Delta\mathrm{z}) = \begin{bmatrix} 1 & 0 & 0 & -\Delta\mathrm{x} \\ 0 & 1 & 0 & -\Delta\mathrm{y} \\ 0 & 0 & 1 & -\Delta\mathrm{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18.40}$$

The new point is

$$\mathbf{x}' = \mathbf{T}\,\mathbf{x} = [\mathrm{x} - \Delta\mathrm{x},\ \mathrm{y} - \Delta\mathrm{y},\ \mathrm{z} - \Delta\mathrm{z}, 1] \tag{18.41}$$

The convenience of this notation system is apparent since all operations, including translations, can be done with matrix multiplications.

Rotations are straightforward. A counterclockwise rotation of the axes about the z axis, shown in Fig. 18.22, is accomplished using the rotation matrix

$$\mathbf{R}_{\mathbf{z}}(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18.42}$$

The y-axis rotation matrix is

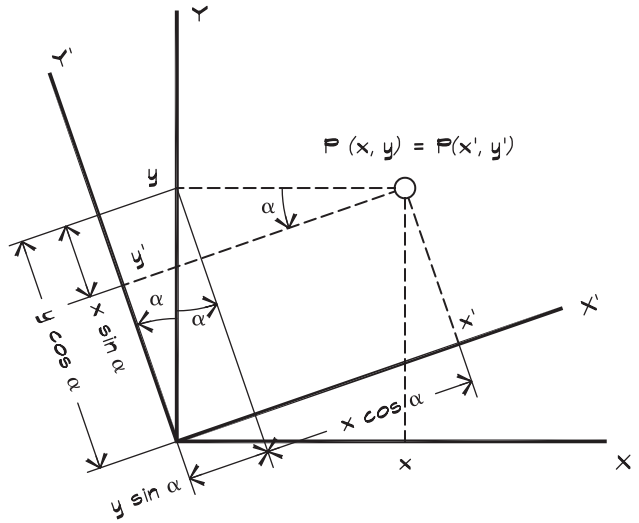
$$\mathbf{R}_{\mathbf{y}}(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18.43}$$

and the x-axis rotation matrix is

$$\mathbf{R}_{\mathbf{x}}(\gamma) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma & 0 \\ 0 & \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18.44}$$

Composite transformations may be carried out by applying the appropriate matrices sequentially.

FIGURE 18.22 Rotation of a Coordinate System



Determination of the Loudspeaker Coordinate System

If a direct-field sound pressure level is to be calculated due to a given loudspeaker, the directivity associated with the receiver must be determined. The loudspeaker is positioned in the room and aimed in a particular direction. The loudspeaker and its aim point in room coordinates are shown in Fig. 18.23.

The loudspeaker is located at

$$\mathbf{x_s} = \left[x_s, y_s, z_s, 1\right] \tag{18.45}$$

and is aimed at

$$\mathbf{x_p} = \left[x_p, y_p, z_p, 1\right] \tag{18.46}$$

FIGURE 18.23 Loudspeaker in the Room Coordinate System

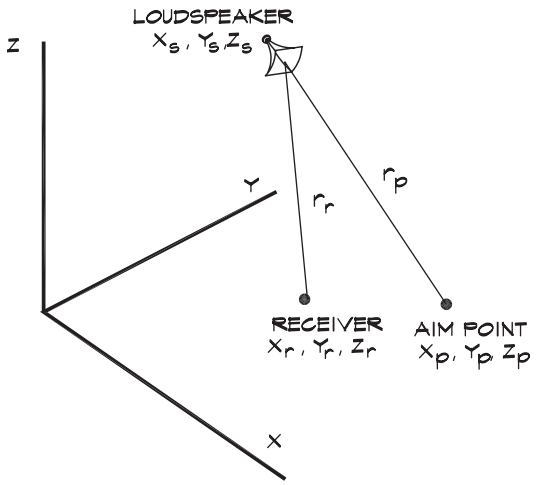
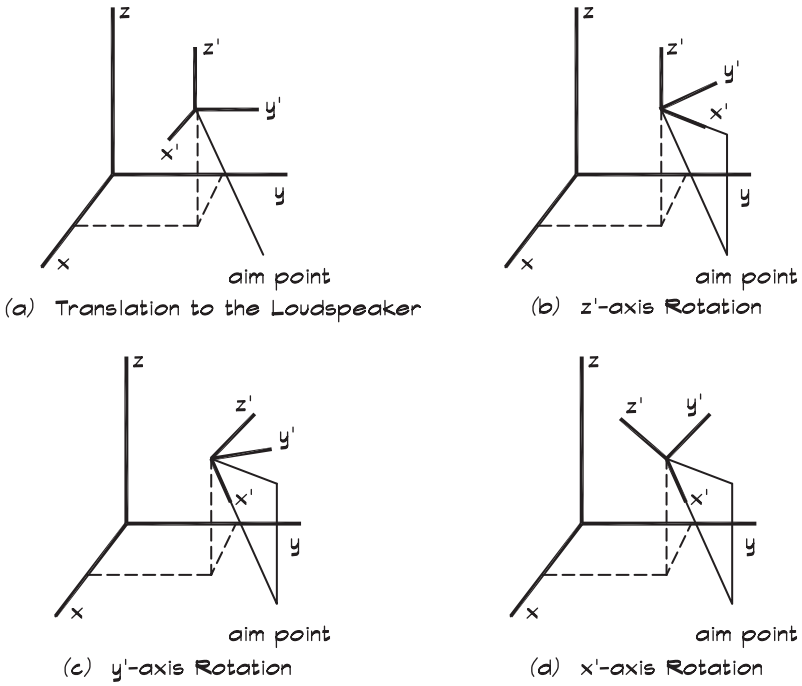


FIGURE 18.24 Four Steps in the Transformation to Loudspeaker Coordinates



The ultimate goal is to describe the receiver location in the loudspeaker coordinate system, so that the directivity associated with the vector between the loudspeaker and the receiver can be found. We align the transformed x axis with the aim point direction and the transformed y axis with the horizontal axis of the loudspeaker in accordance with the four steps outlined in Fig. 18.24. The first step in this process is to translate the room origin to the loudspeaker location. The translation matrix is

$$\mathbf{T_s} = \begin{bmatrix} 1 & 0 & 0 & -x_s \\ 0 & 1 & 0 & -y_s \\ 0 & 0 & 1 & -z_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18.47}$$

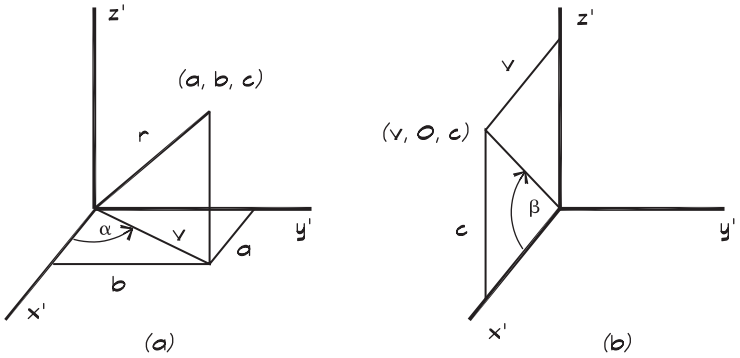
Having translated the origin, we orient the coordinate system so that its x axis is aligned with the loudspeaker centerline. Since the centerline is pointed at the aim point, we locate the loudspeaker aim point, $\mathbf{x_p}$, in room coordinates and transform it using Eq. 18.40. This yields a transformed aim point,

$$\mathbf{x'_p} = \left[x'_p,\ y'_p,\ z'_p,\ 1\right] \tag{18.48}$$

which has a unit vector

$$\frac{\mathbf{x'_p}}{r_p} = [a,\ b,\ c,\ 1] \tag{18.49}$$

FIGURE 18.25 Two Rotations Are Used to Align the X Axis with the Aim Point Unit Vector a, b, c



where

$$r_p = \sqrt{\left(x_{\mathbf{p}}{}'\right)^2 + \left(y_{\mathbf{p}}{}'\right)^2 + \left(z_{\mathbf{p}}{}'\right)^2} \tag{18.50}$$

As outlined in Fig. 18.25, we align the x' axis with the aim point unit direction by first rotating the coordinate system around the z' axis by the angle α

$$\cos\alpha = \frac{a}{v} \tag{18.51}$$

where

$$v = \sqrt{a^2 + b^2} \tag{18.52}$$

This puts the loudspeaker system's x' axis in the same plane as the loudspeaker's z' axis and the aim-point vector.

Next we rotate around the new y' axis using Eq. 18.43 by an angle

$$\sin\beta = c \tag{18.53}$$

This places the loudspeaker coordinate system collinear with the vector $\mathbf{x}'_{\mathbf{p}}$. Finally we rotate around the new loudspeaker x' axis by the loudspeaker rotation angle, γ. This last transformation establishes the loudspeaker coordinate system. The z' axis is collinear with the loudspeaker vertical, the y' axis is in the horizontal plane of the loudspeaker, and the x' axis is collinear with the aim point.

The transformation matrices may be applied sequentially so that a given receiver location, x, y, z, in room space becomes a transformed, x', y', z', in loudspeaker space according to the formula

$$\mathbf{x}' = \mathbf{R}_3\,\mathbf{R}_2\,\mathbf{R}_1\,\mathbf{T}_1\,\mathbf{x} \tag{18.54}$$

where the transformation matrices are

$$\mathbf{T_1} = \begin{bmatrix} 1 & 0 & 0 & -x_s \\ 0 & 1 & 0 & -y_s \\ 0 & 0 & 1 & -z_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R_1} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R_2} = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R_3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma & 0 \\ 0 & \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18.55}$$

Applying these transformations yields

$$\begin{aligned} x' &= (x - x_s)\cos\ \alpha\cos\beta - (y - y_s)\sin\alpha\cos\beta + (z - z_s)\sin\beta \\ y' &= (x - x_s)(+\sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma) \\ &\quad + (y - y_s)(\cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma) - (z - z_s)\cos\beta\sin\gamma \\ z' &= (x - x_s)(\sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma) \\ &\quad + (y - y_s)(+\cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma) + (z - z_s)\cos\beta\cos\gamma \end{aligned} \tag{18.56}$$

Directivity Angles in Loudspeaker Coordinates

Once the receiver is located in loudspeaker coordinates, $\mathbf{x}' = [x', y', z', 1]$, the direction angles can be calculated directly. If ϕ and θ are the longitude and latitude of the receiver then $\mathbf{x}'$ can be written

$$\begin{aligned} x' &= r_r\cos\theta \\ y' &= r_r\sin\theta\cos\phi \\ z' &= r_r\sin\theta\sin\phi \end{aligned} \tag{18.57}$$

where

$$r_r = \sqrt{(x_r')^2 + (y_r')^2 + (z_r')^2} \tag{18.58}$$

Having transformed the coordinates, the directivity angles θ and ϕ follow

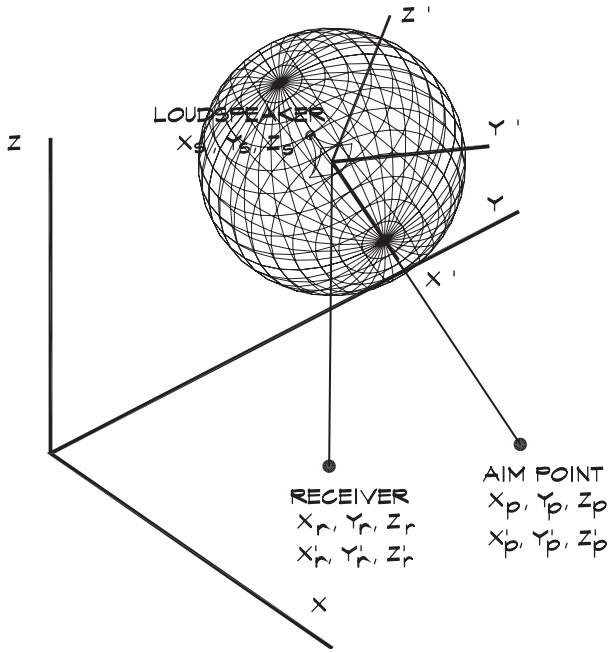
$$\theta = \cos^{-1}\left(\frac{x_r'}{r_r}\right) \tag{18.59}$$

and

$$\phi = \tan^{-1}\left(\frac{z_r'}{y_r'}\right) \tag{18.60}$$

From these angles along with the loudspeaker-directivity data and the loudspeaker-to-receiver distance we can determine the direct-field level associated with a given receiver location. A pictorial of the resultant coordinate system is given in Fig. 18.26.

FIGURE 18.26 Loudspeaker Coordinate System



Multiple Loudspeaker Contributions

If the sound field due to several loudspeakers is present at a receiver location, the direct-field contribution from all sources may be combined to produce an overall direct field level. This can be done in several ways: 1) by adding the energy from each source in accordance with Eq. 2.62, 2) by combining signals in a coherent manner that accounts for the relative phase due to the difference in time, or 3) by coherently combining the energy and integrating over the frequency bandwidth.

An incoherent (energy) combination is the simplest methodology and is sufficiently accurate for most designs. It is inaccurate in that it does not account for the phase differences, but for most applications this is not a fatal flaw. The combined pressure squared is

$$p_{rms}^2 = \sum_{i=1}^{N} p_i^2 \tag{18.61}$$

where the index i runs over all loudspeakers covering a given receiver.

Coherent addition combines the pressures from multiple loudspeakers but also includes consideration of the phase

$$p_{rms}^2 = \left[\sum_{i=1}^{N} p_i \cos \varphi_i\right]^2 + \left[\sum_{i=1}^{N} p_i \sin \varphi_i\right]^2 \tag{18.62}$$

This calculation is carried out at one frequency, representative of the band, and requires an intimate knowledge of the phase of each loudspeaker in a given direction—data that

are not available. The time difference must include the travel time from the driver to the face of the loudspeaker, which is taken to be its location. It also requires the path-length difference between the actual sound path and the distance from the center of the mouth to the receiver. In addition it requires knowledge of the time delay for each loudspeaker due to any electronic processing in the system. Some of the delay may be due to the imposition of an electronic delay line in the signal path for time coincidence. Some may also be due to the time required to perform digital electronic manipulations associated with digital equalizers, crossovers, and other components, which use microprocessors to perform these functions. These delay times are rarely published by the equipment manufacturers but can range from 20 to 80 microseconds per device. In the final analysis, a coherent signal combination is an interesting theoretical exercise. It is only valid at one frequency, however, and relies on difficult assumptions on the directional path lengths and internal time delays of each device. The results obtained from this type of summation can overstate the degree of interference.

A third method has been proposed by JBL (2000) and is called the average complex sum. It uses coherent summation, but instead of integrating over the bandwidth it employs the center frequencies of the third-octave bands within the octave of interest.

$$p_{rms}^2 = \frac{1}{N}\sum_{j=1}^{N}\left\{\left[\sum_{i=1}^{n} p_{ij}\cos\varphi_{ij}\right]^2 + \left[\sum_{i=1}^{n} p_{ij}\sin\varphi_{ij}\right]^2\right\} \tag{18.63}$$

This technique smoothes over many of the rough edges of coherent summation at single frequencies, and yields good agreement with measured results (JBL, 2000).

The phase can be defined in terms of the loudspeaker-to-receiver path length, r

$$\varphi_{ij} = 2\,\pi\ \text{trunc}\left[\frac{r_i}{\lambda_j}\right] \qquad \text{(rad)} \tag{18.64}$$

where the function "trunc" indicates a truncation of all numbers appearing to the left of the decimal point. Note that any time delays are included in the overall distance in Eq. 18.64.

This page intentionally left blank

19 DESIGN OF ROOMS FOR MUSIC

Rooms designed for unamplified music are the most visible and interesting spaces in architectural acoustics. It is here that the science of acoustics and the arts of architecture and music are blended. From the standpoint of acoustical design, the concert hall requires the most careful control, but gives the acoustician the fewest tools to accomplish the task. In contrast to sound system design, where a loudspeaker configuration can yield a very predictable result, the design of a hall for music requires the sculpting of a sound field without the ability to control the original sources, whose type, position, loudness, directivity, and number can change with every performance, or in some cases, with every note. The acoustician can only work indirectly with the room surfaces that reflect, diffuse, or absorb the primal energy. As a consequence he is left with only partial control, and general design principles to follow that have yielded good results in the past, rather than a fixed set of calculations that will produce a precise outcome.

19.1 GENERAL CONSIDERATIONS

The Language of Music

Beranek (1979, 1996), in his books on concert hall design, pointed out the importance of musicians and acousticians being able to communicate with one another. Like the British and Americans, these are two groups separated by a common language. In an attempt to provide a coherent vocabulary, Doelle (1972), Barron (1993), and Beranek (1996) have offered the definitions in Table 19.1.

Each musical term is associated with one or more acoustical properties of a performance space. Table 19.2 (Beranek, 1996; Barron, 1993) lists several of these relationships, whose properties will be explored in greater depth as we progress.

The Influence of Recording

Just as the natural acoustical properties of enclosed spaces have profoundly influenced music and the recording process, so too have electronic reinforcement and recording influenced the design of concert venues and the expectations of concertgoers. In a concert hall, the

TABLE 19.1 Definitions of Common Musical/Acoustical Terms

Term	Definition
Balance	**Equal loudness among the various orchestral and vocal participants.**
Blend	**A harmonious mixture of orchestral sounds.**
Brilliance	**A bright, clear, ringing sound, rich in harmonics, with slowly decaying high-frequency components.**
Clarity	**The degree to which rapidly occurring individual sounds are distinguishable.**
Definition	**Same as clarity.**
Dry or Dead	**Lacking reverberation.**
Dynamic Range	**The range of sound levels heard in the hall (or recording). Dependent on the difference between the loudest level and the lowest background level in a space.**
Echo	**A long delayed reflection of sufficient loudness returned to the listener.**
Ensemble	**The perception that musicians are playing in unison.**
Envelopment	**The impression that the sound is arriving from all directions and surrounding the listener.**
Glare	**High-frequency harshness, due to reflections from flat surfaces.**
Immediacy	**The sense that a hall responds quickly to a note. This depends on the early reflections returned to the musicians.**
Intimacy	**The sensation that music is being played in a small room. A short initial delay gap.**
Liveness	**The same as reverberation, above 350 Hz.**
Presence	**The sense that we are close to the source, based on a high direct-to-reverberant ratio.**
Reverberation	**The sound that remains in a room after the source is turned off. It is characterized by the reverberation time, which has been previously defined.**
Spaciousness	**The perceived widening of the source beyond its visible limits. The apparent source width (ASW) is another descriptor.**
Texture	**The subjective impression that a listener receives from the sequence of reflections returned by the hall.**
Timbre	**The quality of sound that distinguishes one instrument from another.**
Tonal Color	**The balance between tones in different frequency ranges and the balance between sections of the orchestra.**
Tonal Quality	**The beauty or fullness of tone in a space. It can be marred by unwanted noises or by resonances in the hall.**
Uniformity	**The evenness of the sound distribution.**
Warmth	**Low-frequency reverberation, between 75 Hz and 350 Hz.**

TABLE 19.2 Musical Terms and the Related Acoustical Factors

Musical Term	Acoustical Factor
Clarity	**Reverberation time** **Early-to-late energy ratio** **Speed of the music**
Intimacy	**Initial delay gap** **Proximity to the musicians**
Spaciousness	**Apparent source width of early sound** **Listener envelopment by the reverberant sound**
Timbre and Tone Color	**Frequency balance in reflection and absorption**
Color	**Richness of treble** **Tonal distortion** **Texture** **Balance** **Blend** **Diffusion** **Focusing**
Envelopment	**Lateral reflections** **Reverberant sound**
Ensemble	**Musicians' ability to hear each other**
Dynamic Range	**Level of the fortissimo minus the background noise level**
Warmth	**Low-frequency reverberation**

receiver's location determines the sound stage of his listening experience. In a rectangular room the listener is positioned facing the front of the orchestra, where the traditional arrangement of the instruments, with first violins to the left and cellos and basses on the right, is maintained. This is close to the listening experience presented by a high-quality stereo recording since the primary microphone pair is placed above the musicians near the front of the audience.

In a live performance the listener's experience is much more varied. Depending on his seat location and the shape of the hall, the orchestral sound can be quite different. Because of high-quality recordings, the audience may come with certain expectations about their musical experience. The sound of the highs in live performances is much less pronounced since close miking is employed in recordings. The concert hall designer may wish to increase the high-frequency reflections both to the audience in the center of the seating area, and to the players, to provide the extra detail that the listener has come to expect from recordings. This must be balanced against excessive brightness or glare produced by flat reflecting surfaces.

Concert Halls

Concert halls are rooms designed specifically for music in which the musicians and the audience occupy the same space. Classic concert halls are roughly rectangular and have one or more shallow balconies. In good concert halls the number of seats ranges

from 1700 to 2600, with the best halls averaging around 1850. Above 2600 seats, the chances of success are much reduced and the preferred capacity is between 1750 and 2200.

There is no division between the stage and the audience in a concert hall. The orchestra is positioned on a raised platform, sometimes with an organ and choir seating behind it. In classic rectangular halls such as Boston Symphony Hall (Fig. 19.17), there is an enclosure or shell surrounding the orchestra, which is located on one end of the room and extends nearly the full width of the room. Even in New York's Carnegie Hall (Fig. 1.23), which has the appearance of a traditional theater, the floor plan is rectangular with parallel side walls, and the orchestra shell extends out to the side wall. In surround halls, the orchestra is located toward the center of the room with the audience seating wrapped around it. There is no shell in a surround hall, although sometimes there is a wall on three sides of the orchestra and suspended reflectors overhead.

Seating in a concert hall includes a section of seats on the lowest level, about as wide as the orchestra, which runs front to back, sometimes stepping up into one or more balconies. In rectangular rooms most of the seating is located in this strip, whereas in surround halls the strip can be flanked by low side walls, which pick up the rectangular shape. In some rooms the seating rises in steps, like a vineyard cut into a hill, to minimize grazing attenuation. Rectangular rooms sometimes have one or more shallow side balconies, and surround halls can have a series of them.

Reverberation times in concert halls range from 1.5 to 2.2 seconds with preferred values between 1.8 to 2.0 seconds. Clearly the choices depend on the type of music and the volume of the space. Baroque music requires the lowest times (1.5–1.7 seconds), classical the mid-range (1.6–1.9 seconds), and romantic the longest (1.8–2.2 seconds). To attain long reverberation times, the ceilings of concert halls are high, 15 m (50 ft) or more, and diffusive, with coffered patterns having deep (15 cm or 6 in) fissures. Side walls are adorned with columns, caryatids, statuary, and convex shapes that help diffuse the reflected energy.

Opera Houses

An opera house is a mix of a legitimate theater and a concert hall, which constrains the design more than a pure concert hall. In opera, the stage performance rather than the orchestra is the main attraction. The stage house must be deep, with considerable wing space for set movement and storage. The fly tower above the stage, which accommodates the lifting of sets, is about one and a half times the height of the proscenium opening above the top of the arch.

The orchestra is seated in a pit below the stage to balance the level between the singer and the orchestra. The conductor, who must be visible to both the orchestra and the vocalists, stands with his head just at stage level. Open pits, where the orchestra is open to the audience with minimal stage overhang, give the best results in large rooms. In small auditoria some stage floor overhang is necessary to control the loudness.

Sight lines are important for theater, so opera houses tend to be shallower than concert halls. The furthest seat is no more than 30 m (98 ft) from the stage and 30° from the nearest side of the stage opening. Traditional European opera houses are horseshoe-shaped, with many tiers of box seating, stacked like a layer cake. The average seating capacity is about the same as in concert halls but smaller rooms are preferred since the vocal effort required of the singers is much less. Nevertheless a number of the most successful opera houses are relatively large, including Theatro Alla Scalla in Milan (2289 seats; Fig. 19.22), the

Metropolitan in New York (3816 seats; shown in Fig. 1.22), and the Theatro Colon in Buenos Aires (2487 seats; Fig. 19.21). Financial demands push these facilities toward larger capacities, particularly in the United States, where there is little public money available for the arts.

In opera houses most of the action takes place down stage near the center. Singers need an area where they receive adequate reflected energy returning from the room. Consequently the walls surrounding the forestage are often nearly parallel and decorated with large ornate elements such as columns and balconies. These provide diffusion and serve to create a local reverberant field where the performers stand. Reverberation times in opera houses are shorter than in concert halls, between 1.2 and 1.5 seconds. European theaters tend to fall into the low end of this range. Festspielhaus in Bayreuth, Germany (Fig. 1.16), which was built by Richard Wagner for his romantic music, is one exception with a reverberation time of 1.55 seconds occupied.

The traditional opera-house form has served us well, and opera halls have not seen the experimentation that has occurred in concert halls. The classic U-shaped Baroque theater still provides the most common model.

19.2 GENERAL DESIGN PARAMETERS

The Listening Environment

The ideal listening environment for live music depends to some degree on the type of music being played. There are, however, a number of common elements about which there is general agreement.

1. The audience should feel enveloped or surrounded by the sound. This requires strong lateral reflections with a significant fraction of the energy arriving from the side.
2. The room should support instrumental sound by providing a reverberant field, whose duration depends on the type of music being played. A reverberation time that rises with decreasing frequency below 500 Hz yields a beneficial sense of musical warmth.
3. There must be clarity and definition in the rapid musical passages so that they can be appreciated in detail. This requires reflections from supporting surfaces located close to the source or receiver so that the initial time delay gap is short.
4. Sound must have adequate loudness that is evenly distributed throughout the hall. When the number of seats becomes excessive (above 2600 seats), loudness and definition are reduced. In small auditoria the loudness must not be overbearing.
5. A wide bandwidth must be supported. Musical instruments generate sounds from 30 Hz to 12,000 Hz, which is much broader than the speech spectrum. The room must not color the natural spectrum of the music.
6. Noise from exterior sources and mechanical equipment must be controlled so that the quietest instrumental sound can be heard. Background noise levels should not exceed NC 20 in small halls and NC 15 in large concert halls.
7. The detailed reverberation characteristics of the space should be well controlled with a smooth reverberant tail and no echoes, shadowing, coloration, or other defects.

8. The performers should have the ability to hear each other clearly and to receive from the space a reverberant return that is close to that experienced by the audience.

Hall Size

If there is a generalization about concert hall seating capacity then it is the architectural homily, "less is more." Seating capacity pays the rent so there is always pressure to increase it, particularly if there is no other source of funding. The stubborn fact still remains that the best concert halls have low seating capacities and a rectangular shape. The average seating capacity for the top nine halls in Beranek's 1996 study was about 1850, with the highest and lowest not included in the average. By contrast the average seating capacity of the eight least-liked halls (also with highest and lowest number excluded) was 2800. It is far easier to design a good sounding concert hall in the 1750 to 2200 seat range than at higher numbers.

Hall Shape

The most commonly encountered shapes in concert hall design are summarized in Fig. 19.1. The most consistent performer is the rectangular or shoebox floor plan. In Beranek's (1996) review of concert halls, the three halls rated A+, and six of the top eight have this basic form. A rectangular room provides the strong lateral reflections necessary for envelopment as a natural consequence of its shape. Narrow halls also yield low delay times for early reflected sound.

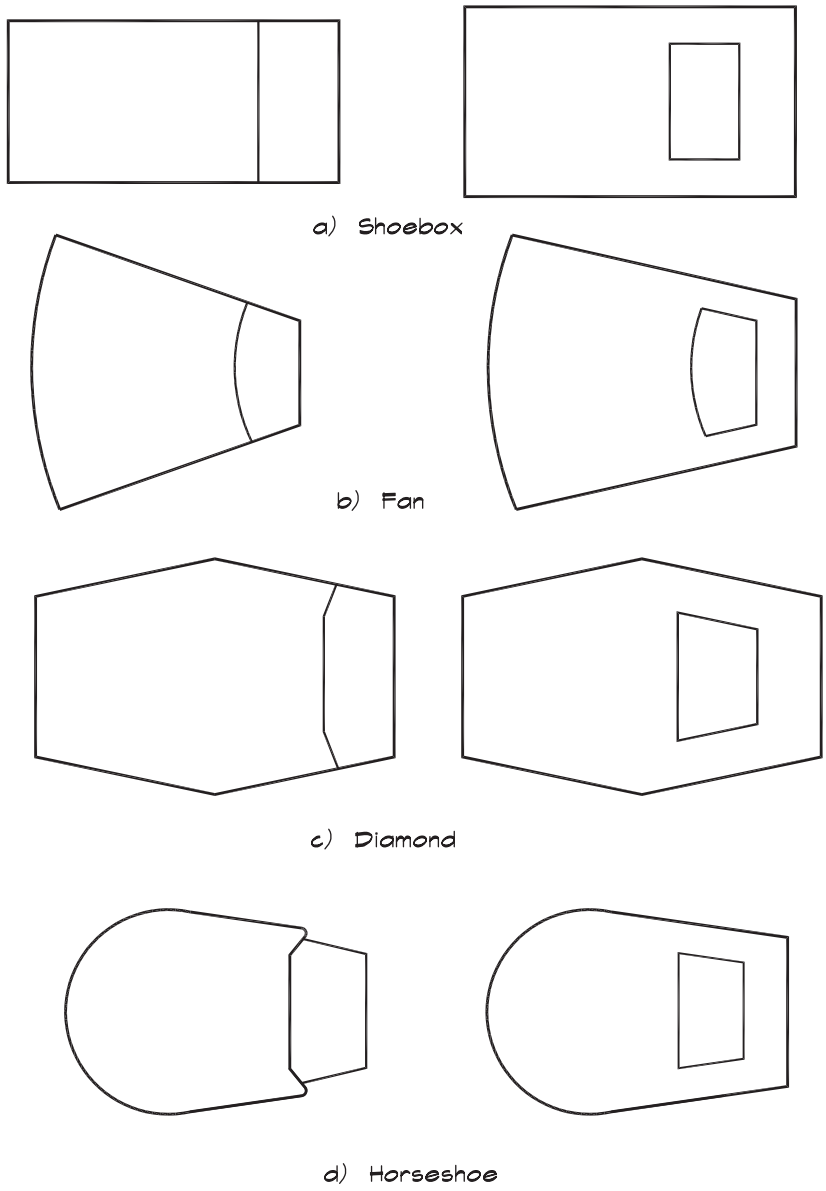
Surround halls can give the architect more freedom of expression, but less acoustical consistency than the standard forms. Any of the basic shapes can be built in a surround configuration by moving the orchestra platform toward the center of the room and arranging seating around the side and the rear. Berlin Philharmonic Hall is a typical example. It has a dazzling appearance but inconsistent acoustics (Beranek, 1994). When sitting on the side, the orchestral balance is changed and the traditional sound stage is rearranged. When the audience is behind the orchestra the spatial relationships are reversed and the balance is very uneven. French horns dominate and vocalists can be difficult to hear.

Surround halls are popular with architects and selection committees due to their dramatic appearance. Interestingly enough the overall hall shape is not infrequently chosen by the selection committee either by specifying the style directly or by selecting an acoustician based on a preference for the desired type. There are recently designed halls such as McDermott (Fig. 19.20) in Dallas, which combine traditional features such as a rectangular shape, narrow width, low seating capacity, overhead reflectors, and still provide interesting design elements.

Hall Volume

Hall volume influences both the reverberation time and room gain. At low capacities a high volume per seat is necessary to control excessive loudness. At higher capacities a low volume per seat helps preserve acoustical energy. Figure 19.2 gives the volume per seat of a number of the better-known halls. In general there is a tendency toward lower volume per seat numbers with increasing seat count; however, there is a wide range of volume per seat values that have produced good results. Each hall designer has his own preference. Those who prefer the model of Boston Symphony Hall cluster around 250 cu ft (7 cu m). Several surround halls, following the Berlin Philharmonic Hall model, are around 320 cu ft (9 cu m). Recent halls such as McDermott in Dallas, and Birmingham Symphony Hall with openable reverberation

FIGURE 19.1 Simple Plan Forms for Concert Halls in Normal and Surround Configurations

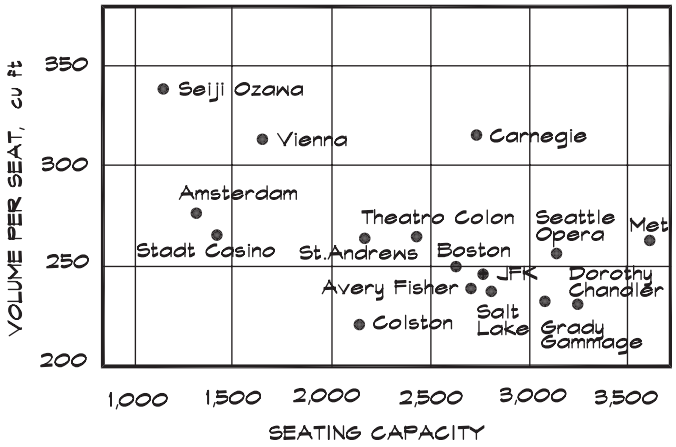


chambers, are at or above 400 cu ft (11.3 cu m); these also feature a large moveable reflector above the orchestra which offsets some of the effects of a high volume.

Surface Materials

Heavy plaster is the most commonly encountered wall and ceiling material used in concert and opera halls. In buildings more than 100 years old it was convenient since it could be worked in a semi-liquid state and would dry to a hard surface. Backings in these old buildings

FIGURE 19.2 Volume per Seat for Several Halls



were mostly brick or wood lath. In Amsterdam it is reeds, and the plaster surface sounds dull or damped when tapped (Beranek, 1996). In newer buildings the backing is metal or gypsum lath. Plaster may also be applied directly onto grouted concrete block or concrete. In a standard application it is trowelled on in three coats—called scratch, brown, and finish—to an overall thickness of 7/8" (22 mm). Each coat is allowed to dry before the next is applied. In some cases thicker plaster, up to 2" (52 mm), has been used. The thicker the plaster and the stiffer the backing, the lower the bass absorption.

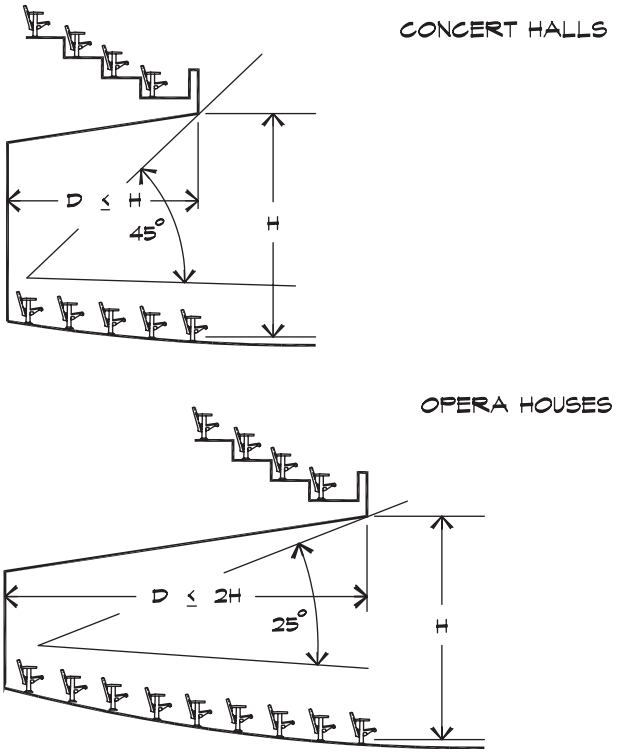
When wood is used it should be heavy, at least 1" (25 mm) thick, and be backed with a solid masonry or concrete structure. Thin wood panels over an airspace are very absorptive in the lower frequencies. Wood can be glued directly to concrete block or concrete walls or applied on furring strips. These have the advantage of allowing the surface to be faired but leave an air gap, which can be filled with pressed fiberglass board. Some designers prefer multiple wood layers of random widths, backed by randomly spaced support members. In practice the variation in spacing can be only a few inches because of structural considerations.

Floors are constructed of concrete or wood on concrete. When floors are concrete, the aisles are covered with a thin carpet walking surface. Wood floors are applied in two layers, for example a 3/4" (19 mm) finish floor on 3/4" plywood subfloor on wood sleepers over concrete. In Amsterdam the space between the sleepers is filled with 1.5" (4 cm) of sand.

Balconies and Overhangs

A balcony brings the audience closer to the musicians and improves sight lines, while dividing the room vertically into separate spaces. Under a deep balcony there is less reverberant energy and the listener has less sense of envelopment and a lower reverberation time. As a consequence under-balcony spaces should be shallow and the front openings high. Figure 19.3 gives recommendations in terms of the ratio of the height to depth ratio. Beranek (1962) suggests a D/H ratio of no more than one for concert halls and no more than two for opera. Barron (1993) recommends that there be a direct line of sight between the rearmost listener and overhead reflecting surfaces and that the angle between the listener and these surfaces not be less than 45° in concert halls and 25° in opera houses.

FIGURE 19.3 Designs for Balcony Overhangs



The greater the balcony overhang the more it cuts the listener off from the reverberant energy in the main portion of the hall, and the shorter the early decay time. Barron (1993) has measured early decay times under balconies and has compared them to those taken in the open part of the hall. His results are given in Fig. 19.4. Knudsen (1965), at Grady Gammage Hall in Tempe, Arizona, designed a flying balcony to allow reverberant energy to pass behind the upper balcony to reach those seated below. He wrote that in the rear seats the audience received the same sense of envelopment as in the orchestra seating area. A semi-cylindrical cavity under the balcony can help offset the lack of reverberant sound by creating some local reverberant focusing. A rising extension of the front part of the balcony can also be helpful for gathering acoustic energy and reflecting it to the rear.

Seating

In large auditoria the biggest absorbing surface is the seated audience. Since it is desirable to maintain a consistent acoustical environment, no matter the number of people, it is most important to use seats whose absorption characteristics closely resemble that of a seated occupant. Concert hall and opera house seats should be padded so that there is little acoustical variation between the rehearsal and the performance. The overall absorption of an audience area is based on the floor area occupied by the opera chairs, rather than the number of seats (Beranek, 1996). This area is increased by 0.5 m (or half the aisle width) on each side to account for the absorption of the exposed edges. Beranek emphasizes the importance of minimizing the area covered by the audience to provide adequate reverberation at minimum

FIGURE 19.4 Under Balcony Early Decay Time T_E (Barron, 1993)

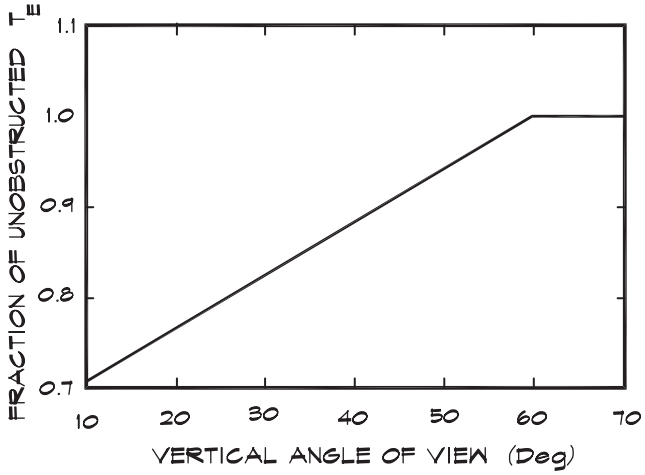


TABLE 19.3 Smoothed Average Absorption Coefficients of Seating (Beranek and Hidaka, 1998)

	Octave Band Center Frequency (Hz)					
Type of Seat	**125**	**250**	**500**	**1k**	**2k**	**4k**
Unoccupied with the weight of upholstering noted						
Heavy	**0.70**	**0.76**	**0.81**	**0.84**	**0.84**	**0.81**
Medium	**0.54**	**0.62**	**0.68**	**0.70**	**0.68**	**0.66**
Light	**0.36**	**0.47**	**0.57**	**0.62**	**0.62**	**0.60**
Occupied						
Heavy	**0.72**	**0.80**	**0.86**	**0.89**	**0.90**	**0.90**
Medium	**0.62**	**0.72**	**0.80**	**0.83**	**0.84**	**0.85**
Light	**0.51**	**0.64**	**0.75**	**0.80**	**0.82**	**0.83**

room volume. He feels that the ratio of the room volume to the seating area is a more useful figure of merit than volume per seat.

The absorption coefficients of seats show considerable variation, which depends on the thickness of the padding. Beranek and Hidaka (1998) have made a comprehensive study of the absorption coefficients of padded seats based on measurements taken in ten halls, before and after the installation of seats. Data were measured for both the occupied and unoccupied conditions. Table 19.3 gives their results grouped according to the amount of padding.

It is interesting that there are residual differences between seats having a different thickness of padding even in the occupied condition. Descriptions of the padding are given in Table 19.4.

TABLE 19.4 Upholstery Details (Beranek and Hidaka, 1998)

Type of Seat	Front Side of Seat Back	Rear Side of Seat Back	Top of Seat Bottom	Arm Rest
Heavy	7.5 cm	Sometimes	10 cm	2 cm
Medium	2.5 cm	0	5 cm	Solid
Light	1.5 cm	0	2.5 cm	Solid

As sound propagates over the seats of a theater it is found that there is an extra attenuation above that which would be expected from distance and high-frequency grazing losses. This extra attenuation is centered on 125 Hz and appears as a deep dip in the attenuation versus frequency response curve. A typical example is shown in Fig. 7.44 (Schultz and Watters, 1964). Ando (1985) has analyzed this effect and has related it to a change in the impedance due to the depth of the seats. He has proposed various solutions, including cavities in the floor and several geometrical alternatives. There is little measured field data available to determine the effectiveness of his suggestions.

One way of reducing the effect is to provide strong overhead reflections from the ceiling or from hanging reflectors. In mixed-use theaters and auditoria, low reflective ceilings are not difficult to include, however in concert halls, side wall reflections are more important and the ceiling is high and designed to diffuse sound. Overhead clouds can be helpful, however they must be of sufficient extent that they provide reflections down to 125 Hz. Another way to minimize seat dip is to group the opera chairs in clusters, which are stepped up and separated by low walls to block the reflected wave. This is called vineyard-step seating since it resembles the planting style found in hillside vineyards. A third way is to elevate the orchestra on a raised platform like those in Amsterdam or Boston to increase the grazing angle.

Platforms

Space must be provided for the musicians in a configuration where they can hear and be heard. The amount of floor space provided for the orchestra is a compromise between comfort and acoustics. This is not to say that acoustical excellence requires us to torture the musicians, but smaller stages generally yield better sound. Beranek (1962) recommends 20 sq ft (1.9 sq m) per musician, which yields a platform size of 2000 sq ft (190 sq m) for a 100 piece orchestra. Platforms in many of the best halls are considerably smaller. Vienna Musikvereinssaal has 1750 sq ft (163 sq m); Amsterdam, 1720 sq ft (160 sq m); and Boston, 1635 sq ft (152 sq m).

Barron (1993) recommends a floor area, which varies with the type of instrument:

13.5 sq ft (1.25 sq m) for upper string and wind instruments
16 sq ft (1.5 sq m) for cellos and larger wind instruments
19 sq ft (1.8 sq m) for double basses
108 sq ft (10 sq m) for timpani and up to 216 sq ft (20 sq m)
More for other percussion

He calculates that a 100-piece orchestra would require an area of 1600 sq ft (150 sq m), and suggests that Beranek's 2000 sq ft (190 sq m) also allows space for soloists, additional

percussion, losses due to risers, and a 3.3 ft (1 m) strip in front of the orchestra. Barron (1993) reckons that a choir needs 5.4 sq ft (0.5 sq m) per seated member, so a 100-voice choir would occupy 540 sq ft (50 sq m) of additional space. The choir seats can be positioned behind and on either side of the orchestra and are available to the audience when not in use. The quality of the sound in these seats is a compromise at best.

The orchestra platform should not be too deep, at most 35 to 40 ft (11–12 m), which results in a width of about 56 ft (17 m) based on Beranek's recommended area. Musicians like to spread themselves out and move forward toward the audience. Conductors frequently feel that moving an orchestra into the audience will improve the sound, but this is not always the case. It is important to keep the players close together and near the rear wall of the platform shell or enclosure, as long as it is not too deep. The double basses should be arrayed along the back wall both to enjoy the benefits of the reflection and to form a line source, which directs sound out toward the audience. Moving the orchestra against a solid reflecting surface is helpful for improving projected level, bass, and ensemble.

The platform floor material is universally wood on wood sleepers. Musicians appreciate the resonance they feel returning from the floor; however if the surface is too thin it will absorb airborne bass energy and if it is too thick it will not vibrate enough. A compromise suggested by Barron (1993) is 7/8" (22 mm) wood on wood sleepers spaced 24" (600 mm) on center. Beranek (1996), in his review of the construction of many halls, found that platform floor thickness ranged from 1/2" (13 mm) to 2" (52 mm). The depth of the airspace below the floor also varied considerably.

Risers are important to minimize shielding of the instruments in the back. Risers that are arranged in a semicircular pattern around the conductor provide an unencumbered path for the direct sound and allow good visual contact between the conductor and the players. The riser must be deep enough to accommodate both the musician and his music stand comfortably. Barron (1993) suggests a horizontal depth of 4 ft (1.25 m) for the upper strings and woodwinds, 5 ft (1.4 m) for brass and cellos, and slightly more for the brass. If the choir is placed on risers, 28" (0.7 m) is needed for a standing vocalist and 32" (0.8 m) for a seated one. The timpani and percussion need a level area about 10 ft (2.8 m) deep. Barron recommends a rise of 4" (100 mm) per step for woodwinds and slightly more further back.

The height of the stage in a theater is just below the eye level of the first row, which is about 44" (1.1 m). In a musical setting, sight lines are less critical and the platform can be higher. A higher platform helps with grazing attenuation and allows the sound to interact with the side walls in a horizontal plane. In Boston, for example, the stage is 54" (1.4 m) high and in Amsterdam it is 59" (1.5 m).

Orchestra Shells

The ability of the orchestra to hear itself is one of the most underrated factors in concert hall design. Where the orchestra is located on a platform surround by a shell such as that in Boston or New York's Avery Fisher Hall, the support provided by the ceiling and walls of the shell generates strong reflections back to the musicians, allowing them to hear themselves and each other. Therefore a goodly fraction of the overhead surface of the shell should direct the sound downward, either specularly or by means of large diffusing elements.

When the orchestra is located near the center of the room, ceiling reflections are not strong and orchestral balance is harder to achieve. In these cases panels or clouds can be

suspended below the ceiling to help generate early reflections back to the players. Individual reflectors are formed from convex sheets or discs, and are located 6 to 8 m (20 to 26 ft) above the musicians (Barron, 1993). A glass or Plexiglas surface, which produces a visible reflection of the other musicians, aids in their positioning (Nagata, 1997). When the room is narrow as in Vienna, the side walls and the overhang below the organ pipes give considerable support, which helps offset the absence of overhead reflectors. Very wide surround halls such as Berlin have low walls partially enclosing the orchestra platform on three sides, along with overhead clouds, to provide early reflections. In other surround halls like Dallas and Birmingham, the support is generated from a large overhead reflector, which can be raised or lowered according to the type of music.

In outdoor venues where the orchestra is miked, the main acoustical purpose of the shell is to provide a balanced sound onstage for the musicians. Sound that reflects out to the audience leaves the platform area and is not available to the orchestra or to the microphones. The shell at the Hollywood Bowl, a large outdoor amphitheater in Los Angeles, is a good example. It is very deep and shaped like truncated cone, which is useful for keeping the rain off, but can be an acoustical hindrance due to long delayed reflections and focusing.

A permanent shell, such as the one in Boston, is made from heavy wood or plaster, much like that of the walls and ceiling of the room. When a room has multiple uses, the shell is divided into sections, which must be light enough to be moved and stored. Permanent shells should be at least 1" (25 mm) thick wood. Portable shells can be as thin as 1/2" (13 mm) wood, but heavier materials are preferred. Where lighter materials are used it is helpful to construct shell surfaces of convex curved sheets with stiffeners in the rear to provide added structural support. Clearly the lighter materials provide less bass reflection but stiffeners help reduce panel movement and low-frequency absorption.

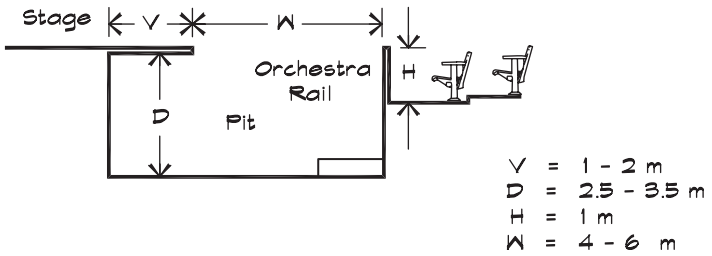
Pits

An orchestra pit is not normally part of a pure concert hall, but is quite important in opera halls and multiuse facilities. A pit functions to reduce the loudness of the orchestra relative to the singers and to provide a place where the musicians can be viewed by the conductor while being out of the sight of the audience. The musicians in the pit should be able to hear each other and the singers. The singers in turn must be able to hear the orchestra so that each can adjust their level to the other. Pit levels can be very high, particularly when the pit is partially enclosed. Absorbent materials lower the pit levels but also lessen the communication between players.

Pit design is difficult since the desired amount of level control depends on the size of the hall, the size of the orchestra, the ability of the singers, and the type of music. The space required is a little less than that necessary for a platform, about 16 to 18 sq ft (1.5 to 1.7 sq m) per musician. Pit orchestras tend to be smaller than concert orchestras with a few exceptions. The pit in the Metropolitan Opera House in New York, with an area of 1420 sq ft (132 sq m), accommodates up to 95 musicians and is approximately 25 ft (7.6 m) deep from the back wall to the front railing (Beranek, 1996).

Pits can be open or partially covered. A sketch of a basic pit configuration is given in Fig. 19.5. The top of the orchestra rail is at the same height as the stage. Where the pit floor is on a lift, it can be raised to form a thrust stage and act as a large freight elevator for sets and grand pianos. Open pits are located in front of the stage and have no stage overhang. Their depth, which must be sufficient to accommodate double basses, is between

FIGURE 19.5 Typical Opera Hall Pit Dimensions (Barron, 1993)



8 and 11.5 ft (2.5 to 3.5 m). The advantage of the open type is that the orchestral sound is unobstructed and the musicians have a better chance of hearing the singers and the sound in the hall. The disadvantages are that the orchestra can overwhelm the singers, there may be poor communication between players, and the lights on the music stands are visible to the audience. It also robs the theater of some very good seats.

A covered pit has its own set of design problems. Depending on the amount of overhang, which can vary from nothing in an open pit to as much as 16 ft (5 m) or more, the orchestra is increasingly isolated from the onstage activities and the audience. In small theaters this may not be a significant disadvantage since the orchestra must be muted, but in large houses it is problematic. With a large stage overhang the orchestra is in its own acoustic space with a separate reverberant field having a double slope. Barron (1993) estimates that the reverberation time in an orchestra pit ranges from 0.35 to 0.7 seconds in the early part of the decay curve. As the sound radiates out to the main room and returns, the late reverberation in the pit mirrors that of the hall.

The front wall of the pit enclosure acts as a barrier to the music for those seated in the first rows of the orchestra seating section. Barron (1993) has estimated the shielding based on floor slope (Fig. 19.6). He suggests that diffusing elements on the side walls and underside of the stage can help fill in the high frequencies in these areas. Overhead reflectors are helpful for the exposed players. If necessary, the orchestra can also be electronically reinforced. This is common practice in Broadway theaters, featuring musicals and light opera, but has not found its way into the classical opera house.

19.3 QUANTIFIABLE ACOUSTICAL ATTRIBUTES

Numerous attempts have been made to correlate subjective judgments of sound quality with objective measurements of acoustical attributes to establish a mathematically predictable assessment of concert halls. Opinion surveys by Beranek (1962), Barron and Lee (1988), and Haan and Fricke (1995), based on interviews with musicians, conductors, critics, and other knowledgeable concertgoers, were used to parse halls by perceived quality. Authors including Sabine (1900), Beranek (1962, 1992, 1996), Hawkes and Douglas (1971), Cremer and Muller (1982), Ando (1985), and Barron (1993) identified quantifiable attributes that contribute to the acoustical quality of concert halls. Beranek, in his early work (1962), identified eight objective factors and correlated them with the subjective judgments, assigning a weighting to the importance of each (Table 19.5).

FIGURE 19.6 Reduction of Sound Level Due to Diffraction over the Pit Rail for Two Floor Slopes (Barron, 1993)

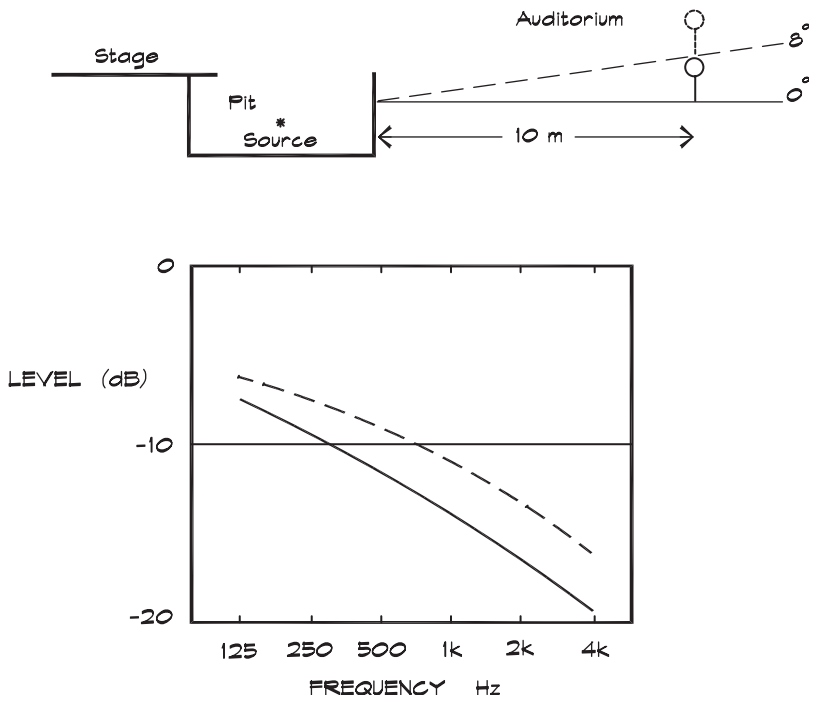


TABLE 19.5 Weightings of Acoustical Attributes (%) (Beranek, 1962)

Attribute	Concert Halls	Opera Houses
Intimacy	40	40
Liveness	15	15
Warmth	15	15
Loudness of Direct Sound	10	10
Loudness of Reverberant Sound	6	6
Balance and Blend	6	10
Diffusion	4	0
Ensemble	4	4
Total	100	100

Studies of Subjective Preference

Though studies based on recollection are useful, researchers wanted better controls and repeatability in the testing. They turned to subjective preferences, which consist of paired comparison tests carried out in a controlled listening environment. Subjects are presented

TABLE 19.6 Weightings of Acoustical Attributes (%) (Beranek, 1996)

Attribute	Concert Halls
$IACC_{E3}$	25
T_E	25
SDI	15
G_{mid}	15
t_I	10
BR	10
Total	100

where $IACC_{E3}$ = interaural cross - correlation coefficient
T_E = early decay time (sec)
SDI = surface diffusivity index
G = difference between the level at a given point and the direct field level at 10 m for an omnidirectional source on stage (dB)
t_I = initial time delay gap (sec)
BR = bass ratio

with sounds by means of binaural headphones, or a three-dimensional array of loudspeakers. The sounds presented can be based on impulse functions recorded using a dummy head in a real hall or by a synthesized simulation using direct, reflected, and reverberant fields. The variables of interest are changed and presented to listeners in pairs so that the question is always, "Which do you prefer?"

In contrast to the comprehension of speech, where intelligibility can be measured one dimensionally, the appreciation of music in a concert hall is multidimensional. Hawkes and Douglas (1971) identified five dimensions (parameters) that have been expanded and refined since that time. These were clarity, reverberation, envelopment, intimacy, and loudness. Wilkens and Plenge (1974) cited three factors: loudness (47%), definition (28%), and appreciation (14%). In that study there was agreement on the attributes, but there was a bipolar preference in loudness with some listeners choosing a fuller sound and others a lighter one.

In his book on concert hall acoustics, Ando (1985) published a study of listener preferences done by playing music through a spherical array of loudspeakers in an anechoic room. Subjects were asked to express their preferences for several variables based on two pieces of music: motif A, Royal Pavanne by Gibbons; and motif B, Sinfonetta, Opus 48, III movement by Malcom Arnold. Based on these choices, Ando identified four orthogonal (independent) variables: loudness, reverberation, envelopment, and clarity, and developed a methodology for calculating each. Later Beranek (1996) added two more to the list: warmth, expressed in terms of the bass ratio (BR); and diffusion, in terms of a surface diffusivity index (SDI). He published the composite list with weightings given in Table 19.6.

The overall assessment of concert hall quality in Ando's scheme is calculated by means of a sum of weighted preferences

$$S = \sum_{i=1}^{N} S_i \tag{19.1}$$

where

$$S_i = -a_i \left| x_i \right|^{3/2} \tag{19.2}$$

and the a_i are the weightings. Each weighted preference has a maximum value of zero. The individual acoustical attributes (Beranek, 1996) are

$$\begin{aligned}
x_1 &= \mathrm{IACC}_{E3} & a_1 &= 1.2 \\
x_2 &= 10 \log \left[t_1/(t_1)_{pref} \right] & a_2 &= 1.42 \\
x_3 &= G_{mid} - (G_{mid})_{pref} & a_3 &= 0.04 \text{ for } G_{mid} < 4.0 \\
& & a_3 &= 0.07 \text{ for } G_{mid} > 5.5 \\
x_4 &= \log \left[T_E/(T_E)_{pref} \right] & a_4 &= 9 \text{ for } T_E < 2.0 \text{ sec} \\
& & a_4 &= 12 \text{ for } T_E > 2.3 \text{ sec} \\
x_5 &= \log \left[BR/(BR)_{pref} \right] & a_5 &= 10 \text{ for } 2.2 \text{ sec} \\
x_6 &= \log \left[SDI/(SDI)_{pref} \right] & a_6 &= 1
\end{aligned} \tag{19.3}$$

and the preferred values of each attribute are

$$\begin{aligned}
(t_1)_{pref} &= 20 \text{ msec or less} \\
(G_{mid})_{pref} &= 4 \text{ to } 5.5 \text{ dB} \\
(T_E)_{pref} &= 2 \text{ to } 2.3 \text{ sec} \\
(BR)_{pref} &= 1.1 \text{ to } 1.25 \text{ for } T_{60} = 2.2 \text{ sec} \\
&\quad\ 1.1 \text{ to } 1.45 \text{ for } T_{60} = 1.8 \text{ sec} \\
(SDI)_{pref} &= 1
\end{aligned} \tag{19.4}$$

There is no preferred value for the IACC, but lower numbers are better than higher numbers. It should be noted that, with the addition of SDI to the attribute list, the quantities are no longer orthogonal. IACC is clearly related to diffusion (SDI).

The change from Beranek's 1962 assignment of 40% of a hall's total rating to intimacy, to his 1996 assignment of 40% to envelopment-related attributes, shows the subjective nature

of these rating schemes. Part of the difficulty is the assumption that subjective preference is additive. Under the presumption of linearity, theoretically a good score in one area can offset a bad score in another. It is equally arguable (Barron, 1993) that a long-delayed reflection or exterior noise can overwhelm the other good aspects of the design.

Modeling Subjective Preferences

Most of the acoustical attributes can be modeled relatively simply using direct and reverberant-field theory in a manner similar to that discussed in Chapt. 18. According to the simple two-field assumption, the relationship (metric) between sound power and sound pressure level is

$$L_p = L_w + 10 \log (Q / 4\pi r^2 + 4 T_{60} / 0.161 V) \tag{19.5}$$

The sound pressure level at a given location may be written relative to the free-field level of an omnidirectional source at 10 m instead of the sound power level. This room gain, sometimes called the total sound level or the strength factor G, is given in metric units by

$$G = L_p - L_0 = 10 \log (100/ r^2 + 31200 T_{60} / V) \tag{19.6}$$

where

$$L_0 = L_w - 10 \log (400\pi) \tag{19.7}$$

Note that these relationships are based on the Sabine reverberant field equation. In real rooms the falloff of level with distance is not always this simple. Field measurements taken in a number of halls led Barron and Lee (1988) to modify the traditional direct/reverberant field relationships since their measured data departed from predicted levels by about 2 dB on average. They found that the reverberant field levels decrease with distance from the source, although at a rate much less than in Fig. 8.14. The average falloff rate in their study was about 1 dB/10 m beyond a distance of 10 m.

Barron and Lee divide the sound fields into three categories: direct, early-reflected, and late-reflected energy, with the break point taken at 80 msec. The energy in each segment, relative to that at 10 m is

$$e_d = 100 / r^2 \tag{19.8}$$

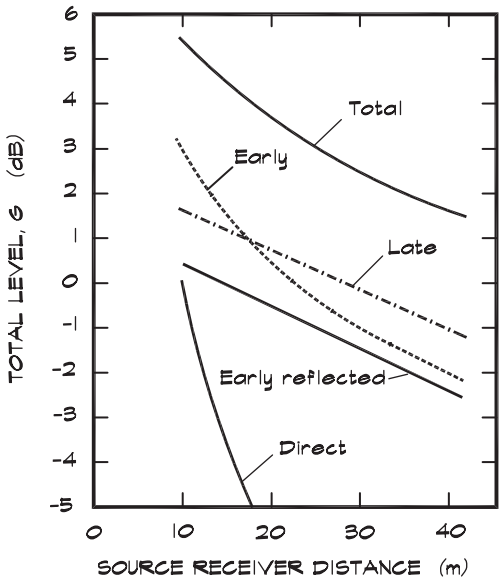
$$e_e = (31200 T_{60} / V) e^{-0.04 r / T_{60}} (1 - e^{-1.11 / T_{60}}) \tag{19.9}$$

$$e_\ell = (31200 T_{60} / V) e^{-0.04 r / T_{60}} (e^{-1.11 / T_{60}}) \tag{19.10}$$

Note that the revised reverberant-energy levels decrease linearly with distance, resulting in a 3 dB reduction at 40 m from the source. The general behavior is given in Fig. 19.7.

Figure 19.8 shows measurements by Bradley (1991) of the room gain in three unoccupied concert halls, Concertgebouw in Amsterdam, Musikvereinssaal in Vienna, and Symphony Hall in Boston, which are regarded as among the best in the world. The three relationships fit the measured data much better than the traditional two-field theory although there are some interesting variations. For Boston the theory underpredicts G at the nearer locations and overpredicts the level at the more distant locations.

FIGURE 19.7 Theoretical Behavior of Various Temporal Components of the Sound Energy (Barron and Lee, 1988)



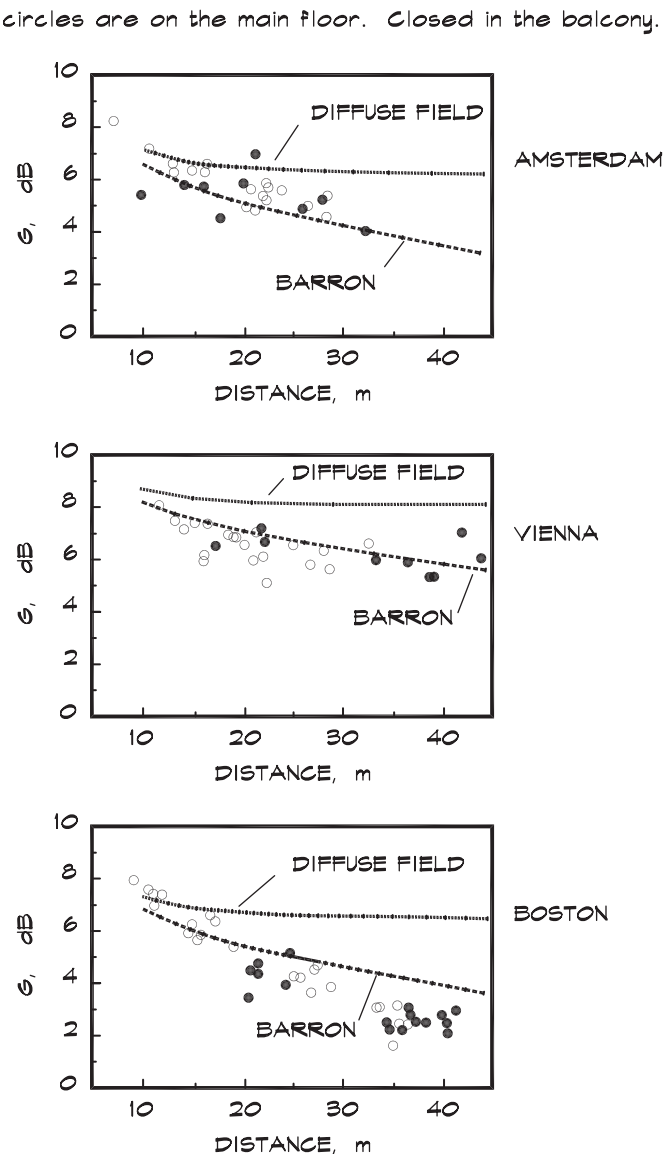
Early Reflections, Intimacy, and Clarity

The importance of early reflections has been recognized for some time. Vitruvius (1960), the Roman architect, extolled the value of what the Greeks called consonant sound or sound that is supported and strengthened by reflections from the stage and other nearby surfaces. Sabine (1964) referenced Vitruvius and also spoke of the need for reflections from surfaces that do not have too great a path length difference at the receiver. Beranek (1962) defined the initial time delay of the first reflection, t_1, as the property he calls intimacy. Barron (1993) gave a slightly different definition, which is the sense that music is being played in a small room. In his 1996 work, Beranek changed his definition, saying that it is the sensation that sound is being played in an appropriately sized room, which he characterizes primarily in terms of initial delay gap, but which also relates to loudness.

The initial delay time is the difference in milliseconds between the arrival time of the strongest reflection, minus the arrival time of the direct sound, at the center of the audience seating area. In 1962 Beranek considered this to be the most important factor in concert hall quality, assigning a maximum of 40 points to halls having an initial time delay of 20 msec or less, reducing credit linearly to zero when the gap is above 70 msec. In his later (1996) study this was reduced to 10 points maximum. He still maintained 20 msec as the threshold but held that beyond 35 msec quality suffered substantially. Ando's concert hall studies confirmed the importance of early reflections, although his test reflection was from the side, positioned at an angle about 36° from the centerline in the horizontal plane. Ando also found that the preferred initial delay time was just under 20 msec.

Barron (1993) reasons that intimacy is not precisely the same as early delay time. The sense that one is in a small room with the musicians is experienced by sitting near the players, irrespective of the delay. Indeed the gap is often greatest at locations closest to the orchestra.

FIGURE 19.8 Comparison of Measured 1-kHz G Values vs Source-Receiver Distance in Amsterdam, Vienna, and Boston (Bradley, 1991)



If intimacy were only dependent on the delay gap, he argues, the sound should always be more intimate at the rear of a hall, where the time gap is the smallest. Subjective surveys by Hawkes and Douglas (1971) and Barron (1988) have shown the opposite to be the case. Since the value of t_1 depends on where the listener is seated, Barron also asks if it is appropriate to use a single number to characterize an entire concert hall. The difference here is primarily one of definition, since Beranek (1962) and Ando (1985) define intimacy as initial delay time, and Barron as the sense that one is in a small room.

The early-to-late signal-to-noise ratio, C_{80}, which was given in Eq. 17.40 in the chapter on speech intelligibility, is also used in musical acoustics. Called the clarity factor in concert

hall analysis, it is the ratio of the sound energy arriving before, to that arriving after, the first 80 milliseconds from the arrival of the direct sound, expressed as a level. It is measured in an unoccupied room and contains more information than the initial time delay, which does not account for the strength of the early reflections.

Using the three-segment model of Barron and Lee the C_{80} can be written as

$$C_{80} = 10 \log \left[(e_d + e_e)/ e_\ell \right] \tag{19.11}$$

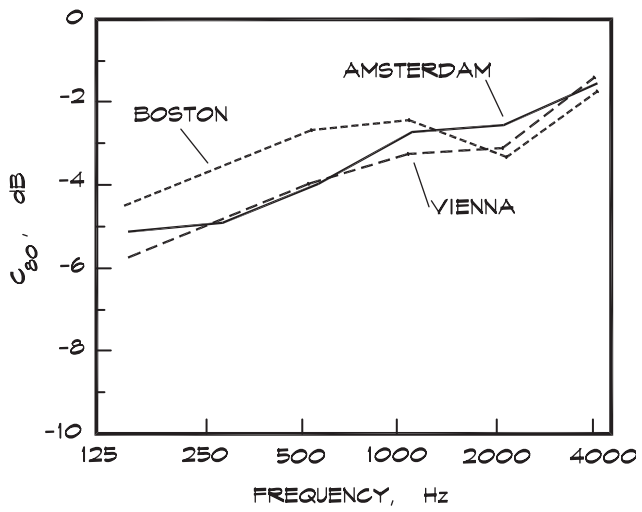
Just as with speech intelligibility, a high signal-to-noise ratio, which results from strong early reflections, yields musical clarity. Values of the C_{80}, which can range from small positive numbers, implying a dead room, to small negative values for very reverberant spaces, fall into the $\pm$ 4 dB range. The preferred values of C_{80} are between 0 and -4 dB. Several highly successful halls: Boston, Amsterdam, and Vienna, have C_{80} (3) values (Beranek, 1996) that are between -2.7 and -3.7 dB. The (3) means that three C_{80} values, those in the 500, 1k, and 2k Hz octave bands, are averaged. Bradley (1991) has calculated occupied C_{80} values, which are reproduced in Fig. 19.9 for the three halls. The calculated values are based on measured unoccupied numbers in accordance with the relationship.

$$C_{80-o} = C_{80} + 13 \log \left[T_{60} / T_{60-o} \right] \tag{19.12}$$

It is interesting to note how close the values are to one another.

Early reflections are important to the appreciation of rapid musical passages and passages played by the quieter instruments such as the harpsichord, recorder, and guitar. Convex surfaces are placed above the orchestra, at a height of around 20 feet, to provide reflections back to the musicians and the audience seated on the main floor. These surfaces should be arranged in the same plane to avoid Bragg scattering, discussed in Chapt. 7. For reinforcement over a wide frequency range, the individual panels should be relatively large, at least 6 feet (1.8 m) long, and arrayed across the entire width of the orchestra. A slight

FIGURE 19.9 Average of Measured C_{80} Values in Three Halls (Bradley, 1991)



lateral convexity helps to fill in the gaps left by the space between panels. Longitudinal convexity compensates for changes in instrument or performer location. Cremer and Muller (1982) recommend a minimum open area of about 70%, with many small clouds being preferred to a few large reflectors, to insure coupling between the upper and lower volumes of the hall.

Sidewalls provide the necessary early reflections for the audience seated toward the rear of the concert hall. In rectangular rooms these are parallel. In surround halls local reflecting surfaces must be carefully crafted to fill in the needed reflections. The high ceiling is designed to be diffuse and thus is not particularly useful for this task.

Liveness, Reverberation Time, and Early Decay Time

Liveness or reverberation is the most recognizable acoustical parameter associated with concert halls. The reverberation time has been a useful metric since its discovery by Sabine. Although it is defined as the time required for the reverberant sound to decay 60 dB below the maximum, it is measured over a range of levels between -5 and -35 dB and then doubled. If it were measured over the full 60 dB range, the reverberant tail would run into the background noise and corrupt the measurement. In Table 19.5 liveness is assigned 15 points; in Table 19.6 it increases to 25 points. As we have discussed, liveness and clarity are not independent variables, which is why the latter was dropped in Ando's study of orthogonal preferences.

The early decay time is another measure of liveness. It is the reverberation time calculated from the initial portion of the room decay curve. Formally it is the time taken for the first 10 dB of level reduction multiplied by 6, making it comparable to the reverberation time. Early decay time varies less with occupancy than does the reverberation time and is measured without an audience. Beranek's (1996) data show a difference of about 0.3 seconds between the unoccupied T_E and the occupied T_{60} for the 52 halls surveyed

$$T_E \cong T_{60-o} + 0.3 \quad \text{(sec)} \tag{19.13}$$

although the differences in several of the best halls are larger. In both of Beranek's books the preferred reverberation times were 1.8 to 2.0 seconds for concert halls and 1.3 to 1.5 seconds for opera houses, although he notes that the range for successful opera houses extends as high as 1.7 seconds in Theatro Colon in Buenos Aires. Kuhl (1954) reported differences with the type of music: 1.54 sec for early classical music (Symphony No. 41 "Jupiter" by Mozart), 2.07 sec for romantic classical music (Symphony No. 4 by Brahms), and 1.48 sec for modern music ("Le Sacré du Printemps" by Stravinsky). Figure 17.10 gives general recommendations on reverberation times versus room volumes for various types of program material. Barron (1993) also lists a range of recommended reverberation times by type of music in Table 19.7.

Early decay time can be estimated from a closed formula such as the Sabine equation and Eq. 19.13; however, this does not separate the two. In concert hall design, early decay time is determined by measurements on models or by ray tracing. It depends primarily on the interactions with reflecting surfaces located near the orchestra. For Boston Symphony Hall, with 2600 seats, a volume of 662,000 cu ft (18,750 cu m), and a reverberation time of 1.85 seconds, the first 10 dB of reverberant room decay occurs in about 300 milliseconds. Equation 8.51 and rough dimensions, 172 ft (52 m) long $\times$ 76 ft (23 m) wide $\times$ 62 ft (19 m) high, allow the calculation of a gross surface area of 48,892 sq ft (4545 sq m) and a mean time between collisions of 48 msec. Thus in this room the early decay time is based only

TABLE 19.7 Recommended Occupied Reverberation Times (Barron, 1993)

Type of Music	Reverberation Time (sec)
Organ	> 2.5
Romantic classical	1.8 – 2.2
Early classical	1.6 – 1.8
Opera	1.3 – 1.8
Chamber	1.4 – 1.7
Drama (spoken word)	0.7 – 1.0

on the first six surface reflections for a given ray. Figure 19.10 gives examples of 1000 Hz early decay times measured at various positions in our three halls. It is interesting that in all three cases the measured values are lower close to the platform and increase with distance. Once the distance is greater than about 20 m the reverberation times flatten out and are constant.

Envelopment, Lateral Reflections, and Interaural Cross-correlation

Harold Marshall (1967) and Veneklasen and Hyde (1969) identified the importance of reflections that surround or envelop the listener with reverberant sound coming from the side. Later authors quantified these effects using the lateral energy fraction and the interaural cross-correlation coefficient (IACC). Strong side reflections are generated by narrow rectangular rooms and from surfaces placed close to the listener. Not surprisingly Ando's (1985) subjective preference studies showed a strong correlation between the IACC and room width.

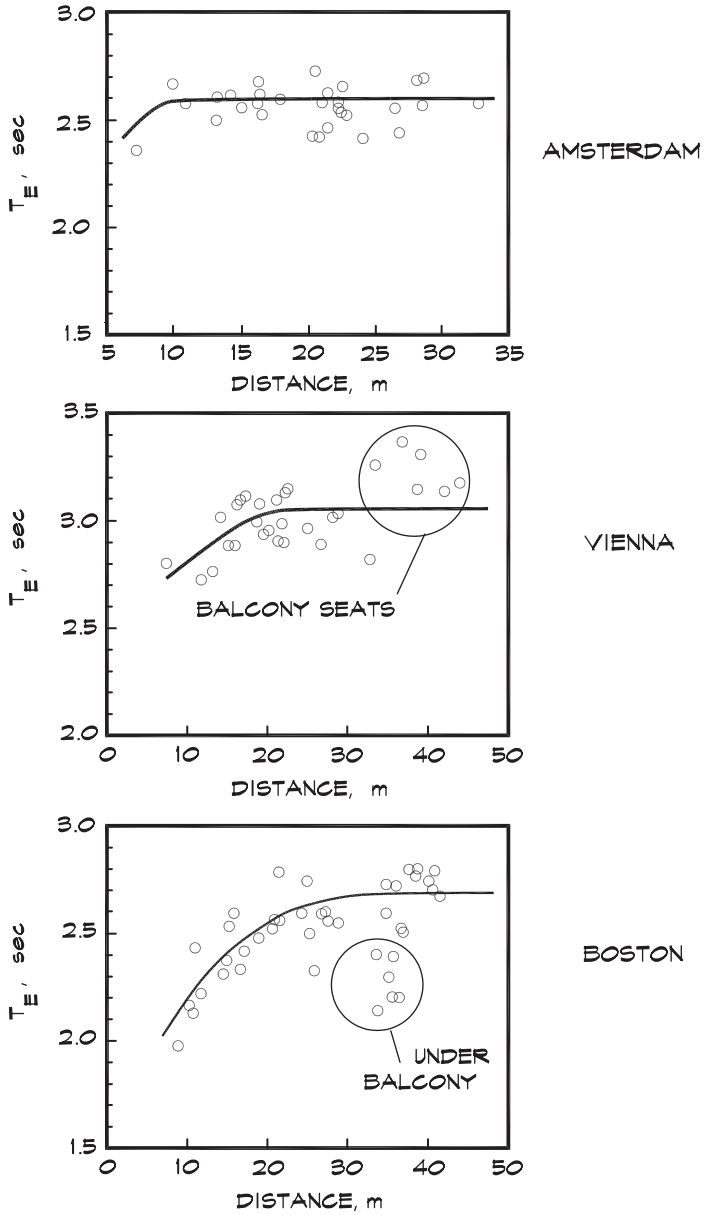
The interaural cross-correlation is a measure of the similarity of sound arriving at two points—in this case, the two ears of the listener. Mathematically it is based on the interaural cross-correlation fraction defined as

$$\mathrm{IACF}_t(\tau) = \frac{\int_{t_1}^{t_2} p_L(t)\; p_R(t+\tau)\; dt}{\left[\int_{t_1}^{t_2} p_L^2(t) \int_{t_1}^{t_2} p_R^2(t)\; dt\right]^{1/2}} \tag{19.14}$$

where the L and R refer to the entrances to the left and right ear canals. The maximum possible value for IACF_t is one, occurring when both signals are the same. The integration is done beginning from time t_1 measured from the arrival of the direct sound at one ear and ending at time t_2, which is selected arbitrarily depending on the period of interest. The variable τ accounts for the time difference between the two ears and is varied over a range from -1 to $+1$ msec from the first arrival. To obtain a single number the maximum value of Eq. 19.14 is taken

$$\mathrm{IACC}_t = \left|\mathrm{IACF}_t(\tau)\right|_{\max} \quad \text{for } -1 < \tau < +1 \tag{19.15}$$

FIGURE 19.10 Measured 1 kHz T_E Values vs Source Receiver Distance in Amsterdam, Vienna, and Boston (Bradley, 1991)



and is called the interaural cross-correlation coefficient. The integration time can be varied with different results. For $t_1 = 0$ and $t_2 = 1,000$ msec, the term is designated $IACC_A$. The early $IACC_E$ (0 to 80 msec) is a measure of the apparent source width (ASW) and the late $IACC_L$ (80 to 1,000 msec) is a measure of listener envelopment (Beranek, 1996).

Preferred values of the IACC are low, approaching 0.3, meaning that the sound arriving at each ear is only 30% correlated. The greater the difference, the higher the preference.

These coefficients are measured in the mid frequency octave bands (0.5, 1, and 2 kHz) and averaged. Data are often listed as (1 – IACC), for which higher numbers are better. Beranek (1996) lists data for 37 halls. The $(1 - \mathrm{IACC}_{E3})$ numbers for three of the best are Boston Symphony Hall, 0.65, Amsterdam Concertgebouw, 0.62, and Vienna Gr. Musikvereinssaal, 0.71.

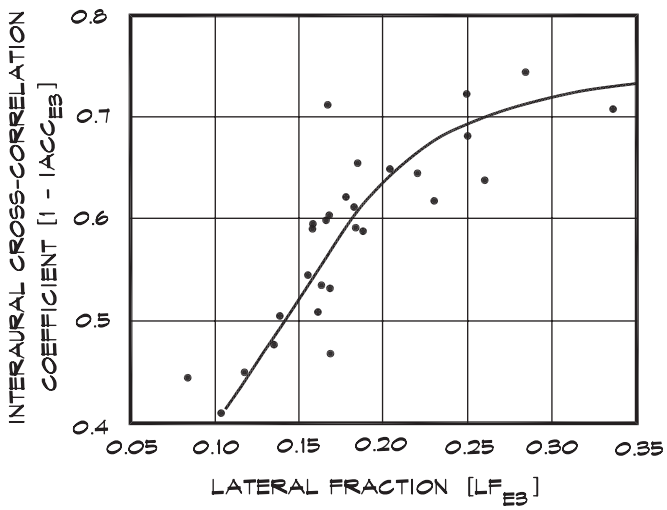
The strength of lateral reflections can be measured directly by using a second order bidirectional microphone, shown in Table 4.1, and then comparing it to an omnidirectional microphone. The figure-of-eight microphone, as it is called, is aligned so that its null direction points toward the source. The two signals are integrated out to 80 msec and compared

$$\mathrm{LF} = \frac{\int_{.005}^{.08} p^2(t) \cos^2 \theta \, dt}{\int_{0}^{.08} p^2(t) \, dt} \tag{19.16}$$

The directional integration begins at 5 msec to insure that the direct sound is not included. The sound pressures are taken in the 125–1000 Hz frequency range. Okano, Beranek, and Hidaka (1998) have published a graph shown in Fig. 19.11 giving the relationship between IACC and the LF. Since they are highly correlated, one can be estimated from the other.

Lateral reflections and envelopment come about naturally from rectangular halls having widths on the order of 75 ft (23 m). A high degree of diffusion is particularly helpful, along with a shallow rake to the main floor seating area, which allows the sound to interact with the side walls in a plane above the audience. Deep overhanging balconies produce the opposite effect, cutting the audience off from the reverberant field in the rest of the room since the sound under the overhang only comes from the front.

FIGURE 19.11 Hall-Averaged Values of $(1-\mathrm{IACC}_{E3})$ vs LF_{E3} as Determined in 28 Existing Concert Halls (Okano, Beranek, and Hidaka, 1998)



Loudness, G_{mid}, Volume, and Volume per Seat

The loudness of both the direct and the reverberant field sounds in a hall is quite important, especially in small recital halls where even a modestly sized orchestra can be overwhelming. Ando (1985), in his study of concert hall acoustics, identified loudness as one of the four orthogonal acoustical attributes. Listeners in that study preferred overall average maximum levels in the 77 to 79 dBA (slow) range for motif A and 79 to 80 dBA for motif B. Beranek (1996) recast this level preference in terms of the room gain, which is defined as the level of sound measured at any point in an empty hall due to an omnidirectional source located on stage, minus the free field (anechoic) level of the source at a distance of 10 m (33 ft).

$$G = 10 \log \frac{\int_0^{t_2} p^2(t)\, dt}{\int_0^{t_2} p_A^2(t)\, dt} \tag{19.17}$$

where the subscript A indicates the anechoic or direct field. According to diffuse-field theory, the value of G_{mid} is proportional to the log of the total absorption in the space. Since the absorption is seldom measured directly, the room gain is computed from the ratio of the reverberation time to the room volume, both easily measured quantities. Figure 19.12 shows the relationship for a number of halls. This is written in terms of the Sabine equation for steady state ($t_2 = \infty$) condition in Eq. 19.6. When the room gain is averaged over the 500 and 1000 Hz octave-band values, it is known as G_{mid}. The average of the 250 and 125 Hz bands is G_{low}. Preferred values of G_{mid} are in the 5 ± 1 dB range. Above 5 dB halls can be overpowered by an orchestra, although two of the best halls, Concertgebouw (at 6.0), and Musikvereinssaal (at 7.0) have values above 5. Below 3.5 dB halls have too little gain.

Beranek (1996) argues that once the room gain and the reverberation time are chosen the volume is set through the relations given in Fig. 19.13, and the number of people by the absorption area from Fig. 19.14. To get to this point, Beranek assumes that the audience and orchestra represent 75% of the total absorption in a hall and that the average mid-frequency absorption coefficient is 0.85. The absorption area, S_T (sq m), is the combined area of the audience and the orchestra times a factor of 1.1 to account for the side absorption. The approximate formula (metric) for the occupied reverberation time using these assumptions is

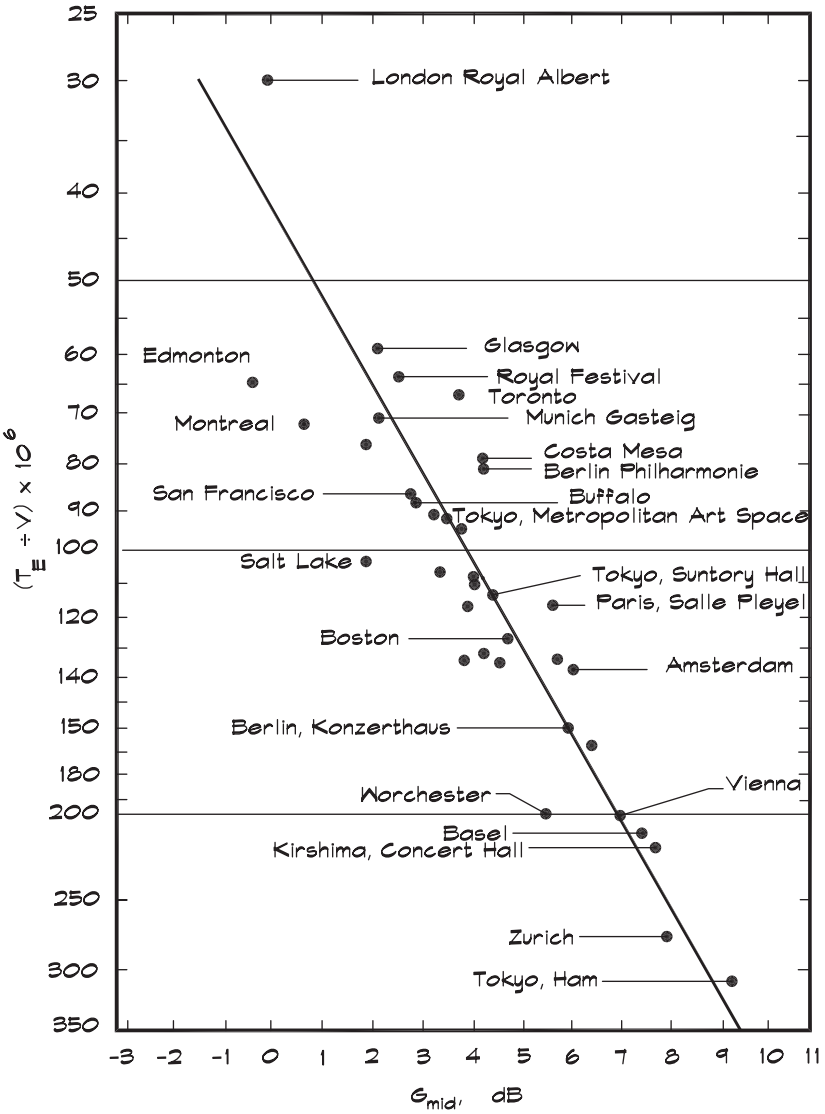
$$T_{60-o} \cong 0.14\,(V/S_T) \tag{19.18}$$

The need to control loudness is important and argues strongly for a volume per seat, which increases as the number of seats decreases. At low seating capacities a large volume per seat is necessary to control the loudness. At large capacities a low volume per seat is preferred to preserve precious acoustical energy.

Warmth and Bass Response

Bass reverberation, which is perceived as warmth, was one of the critical factors to come out of Beranek's 1962 book and was added to Ando's four orthogonal factors as part of his 1996 work. Beranek has quantified it in terms of the ratio between the average low-frequency reverberation times and those at the mid frequencies, a fraction that he defines

FIGURE 19.12 Measured Mid-Frequency Room Gain (Beranek, 1996)



as the bass ratio

$$\mathrm{BR} = \frac{\mathrm{T}_{60}(125) + \mathrm{T}_{60}(250)}{\mathrm{T}_{60}(500) + \mathrm{T}_{60}(1\,\mathrm{k})} \tag{19.19}$$

Beranek (1996) cites bass ratios from 1.1 to 1.25 as being preferred for reverberation times of 2.2 seconds and ratios from 1.1 to 1.45 for reverberation times around 1.8 seconds. These generally are comparable to the curve given in Fig. 17.11. Occupied bass ratios for our three example concert halls are 1.09 (Amsterdam), 1.11 (Vienna), and 1.03 (Boston), which illustrate the difficulty in achieving high numbers. Vienna has large windows and both Amsterdam and Boston have wood floors over deep air spaces. In Amsterdam the orchestra

FIGURE 19.13 Computational Chart for Determining the Volume of a Concert Hall (Beranek, 1996)

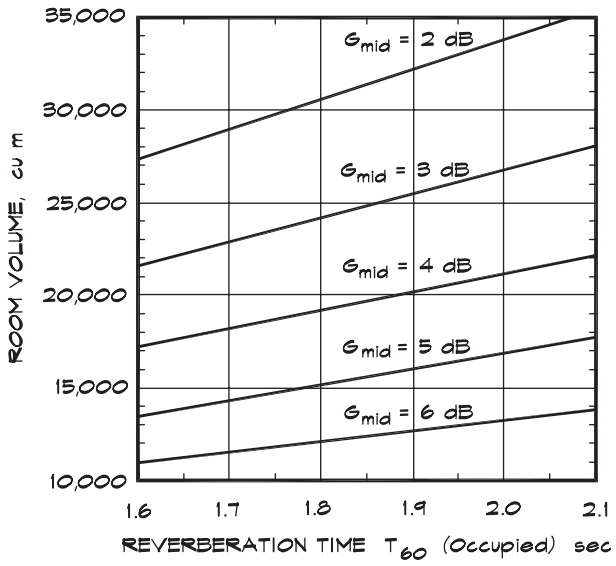
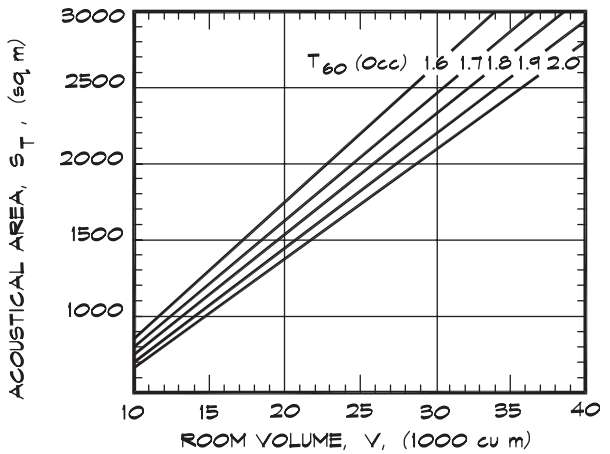


FIGURE 19.14 Computational Chart for Determining the Acoustical Area of a Concert Hall (Beranek, 1996)



is placed atop a high (1.5 m) wood platform. In Boston the seating at the rear of the main floor is mounted on a removable floor of approximately 1" thick wood over an airspace about 2 meters deep. Each of these surfaces absorbs bass.

Bass reverberation is produced by very heavy materials such as 50 mm thick wood on a concrete or plaster backing, or a similar thickness of cement plaster. In halls such as McDermott in Dallas concrete reverberation chambers are used to augment the natural reverberation of the room. Thin surfaces of wood, drywall, and glass absorb low-frequency

energy and should be avoided. Where they are used as finish surfaces they should be glued directly to a backing of concrete or heavy plaster.

Diffusion, SDI

Diffusion, which is the scattering of sound in an omnidirectional rather than a specular pattern, is created by convex or irregular surfaces. The depth of the irregularity should be equal to or greater than a quarter of a wavelength to have appreciable effect at a given frequency. Diffusion contributes to envelopment, and the even distribution of reverberant sound. Beranek (1996) assigns 15 points to it in his rating scheme. Haan and Fricke (1997) relate hall quality, in part, to diffusivity based on a visual inspection. They assign a numerical value called the surface diffusivity index (SDI) to a surface as follows:

High Diffusivity (SDI = 1)

Coffered ceiling with deep (> 100 mm or 4 in) recesses

Random diffusing elements over the whole surface (> 50 mm or 2 in deep)

Medium Diffusivity (SDI = 0.5)

Broken surfaces with shallow recesses (< 50 mm or 2 in deep)

Flat surface behind a semitransparent hard screen

Low Diffusivity (SDI = 0)

Smooth flat or curved surfaces

Absorptive surface

Once the SDI has been established for a given surface the ratings for the whole auditorium are calculated based on a weighted average according to area.

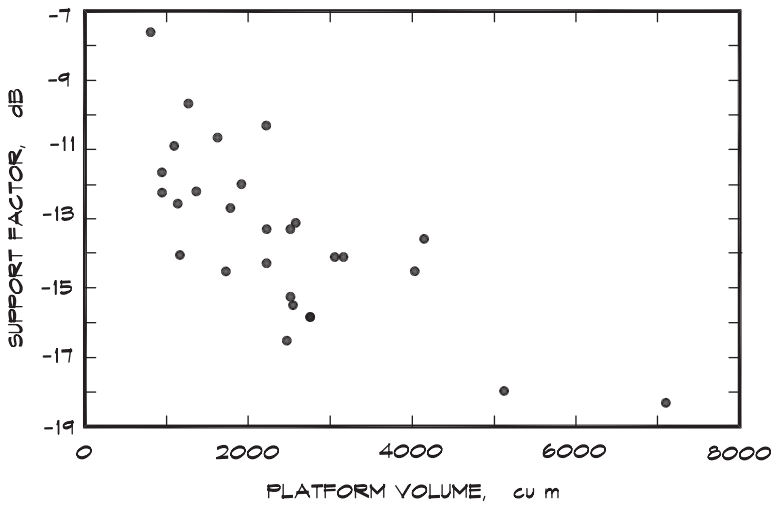
Although there is general agreement that diffusion is good, there is less certainty about the amount or type, and how to measure it. Researchers have struggled with the quantification problem, which is complicated since diffusion depends on the frequency and the angle of incidence of the incoming signal, and the receiver measurements must be performed at locations over the entire half-sphere surrounding the diffuser.

Ensemble, Blend, and Platform Acoustics

The supporting sound, that which returns to the orchestra, has been studied by Gade (1989a/b), Barron (1993), and others. Support is generated through reflections from nearby surfaces. The usual method of measurement is to use an omnidirectional source, located on the orchestra platform, and measure the return from reflecting surfaces with a microphone set at a distance of 1 m from the source. For an impulsive source such as a blank pistol shot, the energy will be returned as a series of scattered pulses. The ratio of the energy returned between 20 and 100 msec, compared to the initial level sampled between 0 and 10 msec (the direct level plus the reflection from the stage), is called the support factor (Beranek, 1996) and is a measure of the musician's ability to hear himself and others near him.

$$\mathrm{ST1} = 10 \log \frac{\int_{.02}^{0.1} p^2(t)\, dt}{\int_{0}^{0.01} p^2(t)\, dt} \tag{19.20}$$

FIGURE 19.15 Support Factor Plotted against Platform Volume in 32 British Halls (Barron, 1993)



Gade (1989b) indicates that this parameter correlates well with the responses from musicians. Barron (1993) found that preferred values of the support factor range from −11 to −13 dB.

Gade (1989a) plotted objective support against platform volume (the platform area times the height to the ceiling or to suspended reflectors) in Fig. 19.15 for a number of British halls. The preferred volumes fall into the 1000 to 2000 cu m (35,000 to 70,000 cu ft) range, which gives us an idea of the reflector heights. Using diffraction theory it is straightforward to add the energies of each reflection within the time period to calculate ST1.

19.4 CONCERT HALLS

In his two comprehensive works, Beranek (1962 and 1996) compiled a substantial record of the design, materials, and statistics of concert and opera halls throughout the world. In 1993 Barron added a detailed survey of halls in Great Britain. Any designer can learn a great deal from a close examination of the drawings and photographs in these books. In this work we will examine a few examples, relying for the most part on data that Beranek compiled. From his definitions, in the following data tables L is the room length from the front of the platform to the average back wall position, D is the distance from the front of the platform to the most distant listener measured on or near centerline, W is the overall width, and H is the height from the lowest floor level to the ceiling above the audience.

Grosser Musikvereinssaal, Vienna, Austria

Generally considered the best in the world, the Viennese concert hall, which opened in 1870, is noted for its combination of low seating capacity and long reverberation time. Having a classic shoebox shape, shown in Fig. 19.16, its narrow width and ornate ceiling along with Oscar-like caryatids supporting the single balcony, provide an abundance of diffusing surfaces. Above the balcony are rows of highly decorated doors and deeply recessed windows that provide high-frequency scattering, along with the suspended crystal chandeliers.

FIGURE 19.16 Grosser Musikvereinssaal, Vienna, Austria (Beranek, 1996)

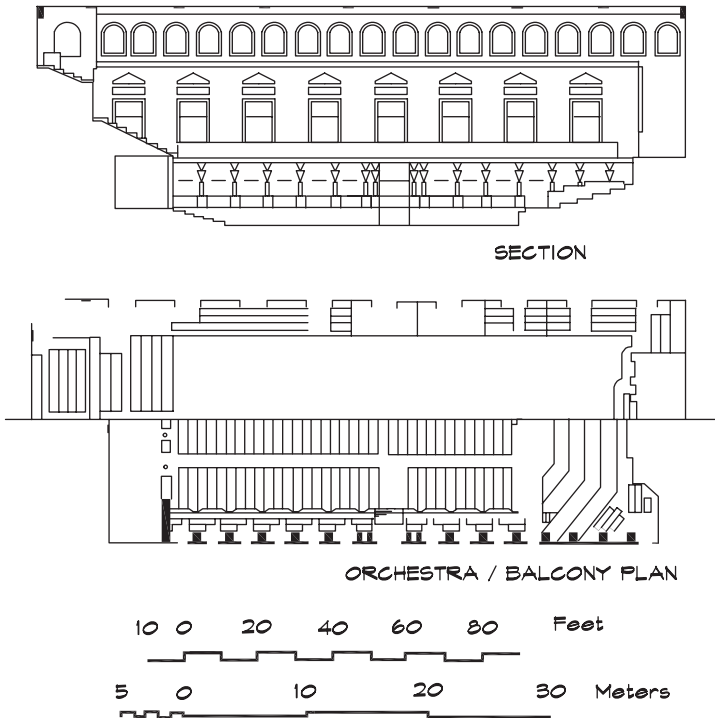


TABLE 19.8 Selected Technical Data for Grosser Musikvereinssaal (Beranek, 1996)

Attribute	Metric Units	FP Units
Volume (V)	15,000 cu m	530,000 cu ft
Dimensions (LWH)	35.7 m × 19.8 m × 17.4 m	117 ft × 65 ft × 57 ft
Furthest Seat (D)	40.2 m	132 ft
Seating Capacity (N)	1680	
Total Absorptive Area (S_T)	1,118 sq m	12,030 sq ft
V/N	8.93 cu m	315 cu ft
V/S_T	13.4 m	44 ft
T_{60}(Occ) = 2.0 s	T_E(Unocc) = 2.2 s	$(1 - IACC_{E3}) = 0.71$
t_I = 12 ms	G_{mid} = 7.1 dB	BR = 1.11
SDI = 1.0		

Architect: Theophil Ritter von Hansen

The walls and ceiling are plaster, except around the stage, where the walls are wood. The side walls and balcony overhangs furnish early reflections to maintain clarity without suspended reflectors. Its low volume and minimal absorption yield the highest room gain of the three best halls.

Boston Symphony Hall, Boston, MA, USA

Boston Symphony Hall, shown in Fig. 19.17, was designed by Wallace Clement Sabine in the late nineteenth century, and opened in 1900. It has a classic shoebox shape, with a 54" (1.37 m) high raised platform, surrounded by a shallow shell at one end. The central feature of the orchestra enclosure are gilded organ pipes that span the rear wall of the shell and provide a reflecting shelf above the backs of the musicians. Around the sides and across the rear of the hall are two narrow balconies. Above the balconies on either side are niches

FIGURE 19.17 Symphony Hall, Boston, MA (Beranek, 1996)

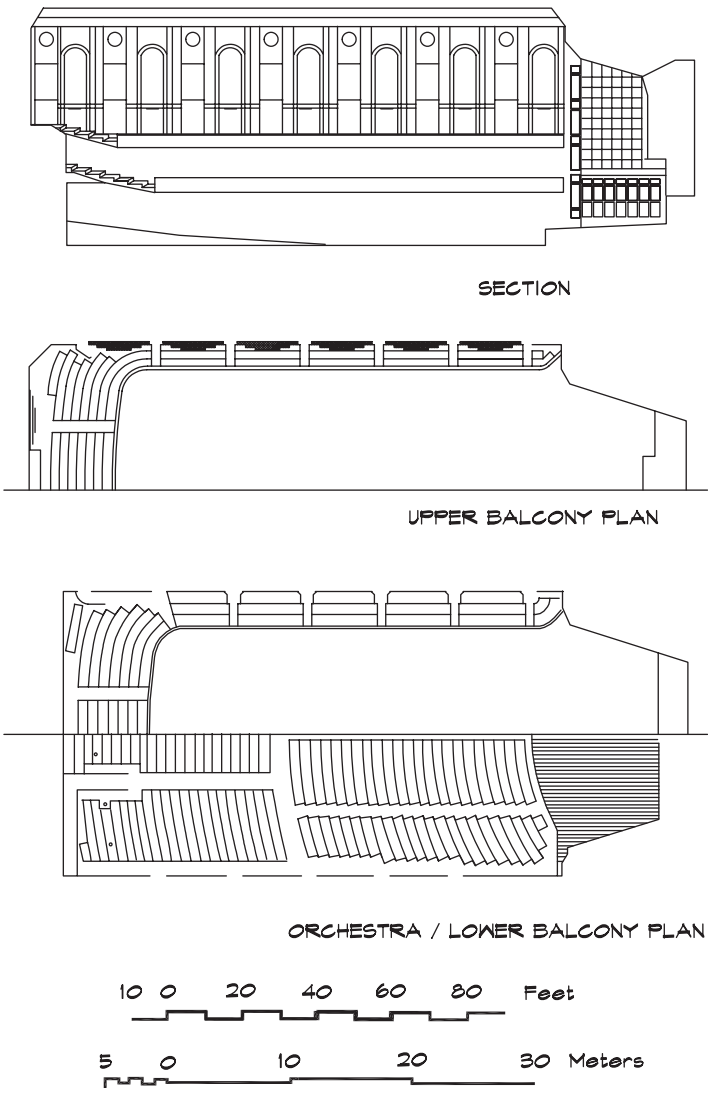


TABLE 19.9 Selected Technical Data for Boston Symphony Hall (Beranek, 1996)

Attribute	Metric Units	FP Units
Volume (V)	**18,750 cu m**	**662,000 cu ft**
Dimensions (LWH)	**39 m × 23.6 m × 18.6 m**	**128 ft × 77 ft × 61 ft**
Furthest Seat (D)	**40.5 m**	**133 ft**
Seating (N)	**2625**	
Total Absorptive Area (S_T)	**1,522 sq m**	**16,385 sq ft**
V/N	**7.14 cu m**	**252 cu ft**
V/S_T	**12.3 m**	**40.4 ft**
T_{60}(Occ) = 1.85 sec	**T_E(Unocc) = 2.1 s**	**$(1 - IACC_{E3}) = 0.65$**
t_I = 15 msec	**G_{mid} = 4.7 dB**	**BR = 1.03**
SDI = 1.0		

Acoustician: Wallace C. Sabine Architect: McKim, Mead, and White

supporting classical Greek and Roman statuary. The ceiling is built of 0.75" (19 mm) plaster formed into deeply coffered rectangles. The walls are mostly plaster on masonry (50%) or metal lath (30%) backing. The remainder, including the stage enclosure, are 0.5" (13 mm) to 1" (25 mm) wood. The floor is concrete with wood overlay, which rises gently at the rear to an elevation of 5' (1.5 m) above the slab. During the summer pops concerts are held and the sloped wooden floor is removed. Food is served and, in the tradition of the early Italian opera houses, conversations continue during the performances, which degrades the clarity in the rear of the main floor.

Boston is regarded as the finest concert hall in the United States. It has excellent clarity and loudness. Its balance is quite good and the bass is very good, particularly at the rear of the balcony. The orchestra platform is small, almost crowded, which helps the onstage ensemble. Its shallow depth improves the bass by encouraging the location of the double basses against the rear wall, under the organ pipes.

Concertgebouw, Amsterdam, Netherlands

Concertgebouw in Amsterdam, shown in Fig 19.18, opened in 1888. Like the two previous examples it has a shoebox shape, however it is much wider at 91 ft than either Vienna or Boston, and it has some seats behind the orchestra. There is no rake to the main floor but this is counterbalanced by a high platform (59 in / 1.5 m). The ceiling is coffered in the center portion with deep semicircular niches at the junction with the walls. These provide substantial diffusion over a wide range of frequencies.

The ceiling and walls are heavy (1.5 in / 4 cm) plaster applied on brick below the balconies. Floors are hardwood on 2 × 3 (5 × 7.5 cm) wood battens filled with 1.6 in (4 cm) of sand over a concrete slab. There is no orchestra enclosure or overhead reflectors, so the clarity is not quite as good as in Vienna and Boston. The sound is rich, full, and enveloping with good bass due to the heavy plaster surfaces.

FIGURE 19.18 Concertgebouw, Amsterdam, Netherlands (Beranek, 1996)

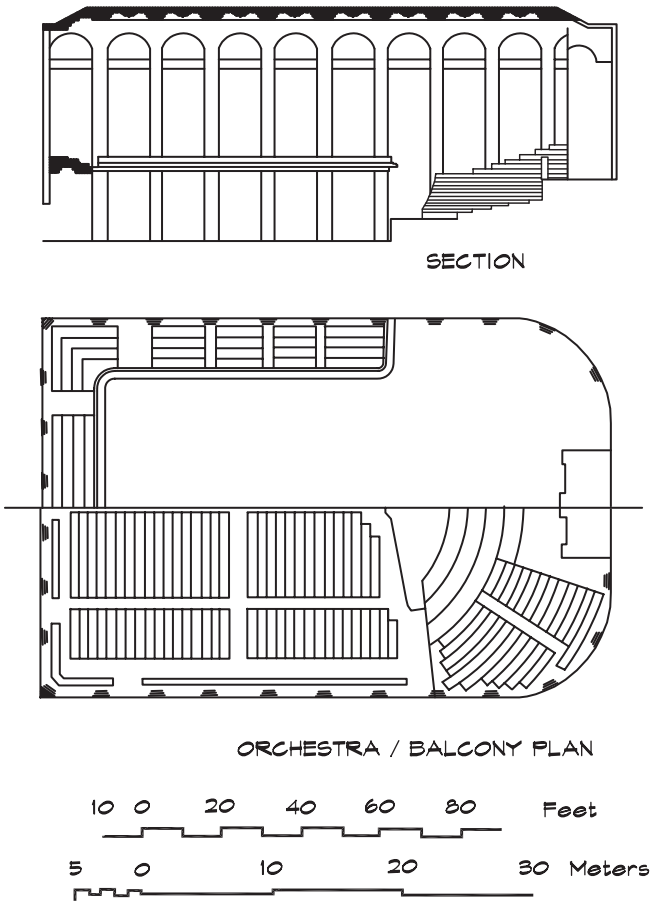


TABLE 19.10 Selected Technical Data for Concertgebouw (Beranek, 1996)

Attribute	Metric Units	FP Units
Volume (V)	18,780 cu m	663,000 cu ft
Dimensions (LWH)	26.2 m × 27.7 m × 17.1 m	86 ft × 91 ft × 56 ft
Furthest Seat (D)	25.6 m	84 ft
Seating (N)	2037	
Total Absorptive Area (S_T)	1,285 sq m	13,830 sq ft
V/N	9.2 cu m	325 cu ft
V/S_T	14.6 m	48 ft
T_{60}(Occ) = 2.0 s	T_E(Unocc) = 2.6 s	$(1 - IACC_{E3}) = 0.46$
t_I = 21 ms	G_{mid} = 5.9 dB	BR = 1.08
SDI = 1.0		
Architect: A. L. van Gendt		

Philharmonie Hall, Berlin, Germany

The Berlin Philharmonic Hall, which opened in 1963, was a dramatic departure from the traditional rectangular or fan-shaped configurations. The architect, Hans Scharoun, and the acousticians, Lothar Cremer with Joachim Nutsch, followed the design philosophy that the orchestra should be located near the middle of the room and be surrounded by the audience, generating the new type of classification—the surround hall. All surround halls sacrifice acoustical excellence mainly in the side and rear seating areas for audience proximity and visual stimulation. Figure 19.19 shows the plan and section of the Berlin hall, although a

FIGURE 19.19 Berlin Philharmonic Hall (Beranek, 1996)

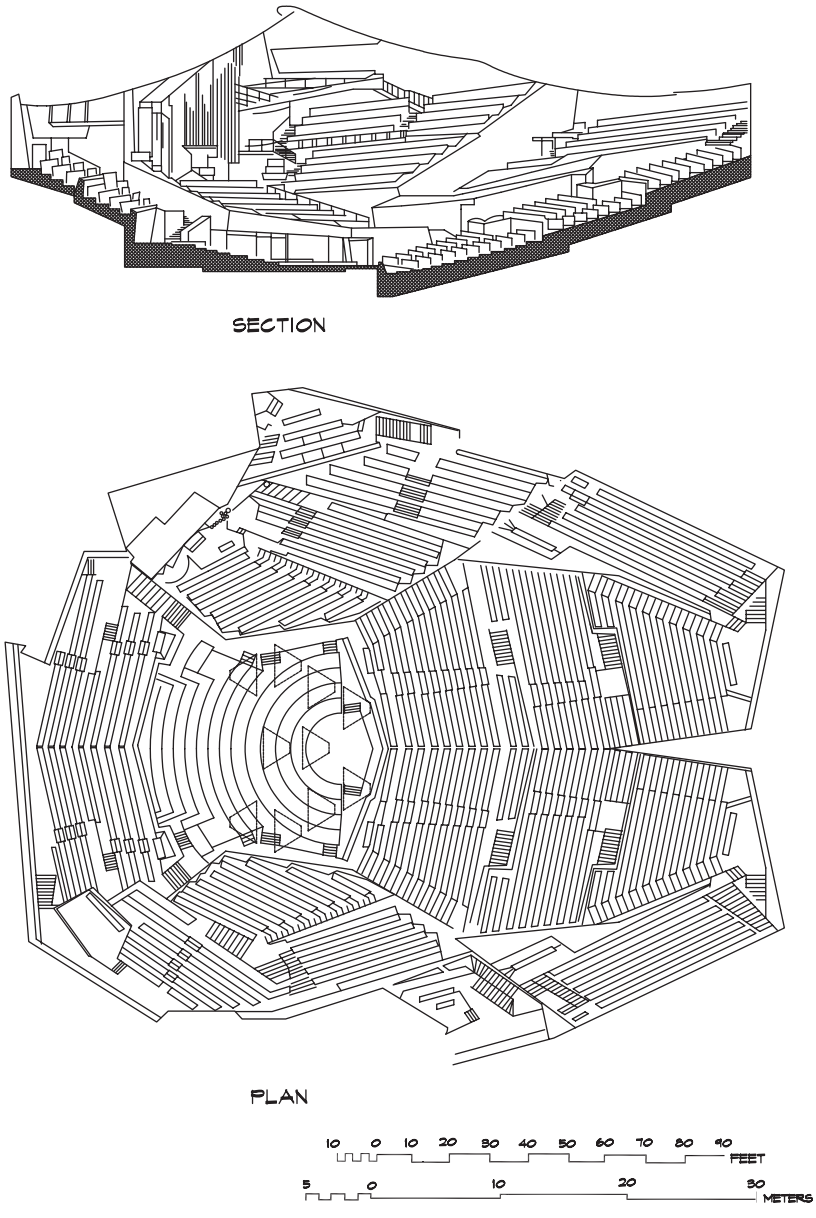


TABLE 19.11 Selected Technical Data for Berlin Philharmonic Hall (Beranek, 1996)

Attribute	Metric Units	FP Units
Volume (V)	21,000 cu m	741,300 cu ft
Dimensions (LWH)	29 m × 42.7 m × 12.8 m	95 ft × 140 ft × 42 ft
Furthest Seat (D)	30 m	98 ft
Seating (N)	2215	
Total Absorptive Area (S_T)	1,558 sq m	16,765 sq ft
V/N	9.0 cu m	317 cu ft
V/S_T	13.5 m	42.2 ft
T_{60}(Occ) = 1.9 s	T_E(Unocc) = 2.1 s	$(1 - IACC_{E3}) = 0.46$
t_I = 21 ms	G_{mid} = 4.3 dB	BR = 1.01
SDI = 0.8		
Acoustician: Lothar Cremer, Joachim Nutsch	Architect: Hans Scharoun	

feeling for the dramatic three-dimensional layout is hard to get from these two dimensional renderings.

The continuous band of seating on the main floor steps up from the orchestra platform in terraces, which are separated by low chevron-shaped walls about 5 feet (1.5 m) high. These separate the seating areas visually and provide a break from the grazing wave reflected from the seats. The upper portion of the walls, which is frequently slanted down, sends sound back to the audience seated in front of them, providing additional early reflections.

Suspended overhead clouds help the musicians onstage and give high-frequency support to the nearest seats. The convex ceiling provides useful reflections in the rear of the hall. The division of the seats into terraced sections leaves each section with a nearby side wall, so useful for envelopment. A sharp vertical wedge bifurcates the rear section of seats providing another reflecting surface in that area. The result is a balanced clear lively sound in front of the orchestra.

On the sides and rear Barron (1993) notes that the character of the sound changes with the nearest instruments audible and those further away less so. Behind the orchestra this effect is less pronounced but the French horns are prominent and the piano and vocals are difficult to hear. The ceiling is treated with small pyramid-shaped Helmholtz resonator absorbers, designed to reduce the reverberation time in the lower registers so that the bass ratio is close to one. Beranek (1996) opines that this is somewhat overdone.

Eugene McDermott Concert Hall in the Morton H. Meyerson Symphony Center, Dallas, TX, USA

Dallas's McDermott Hall (Fig. 19.20) is a good example of a recent (1989) design approach incorporating a blend of the classic shoebox and opera-house forms with interesting technological innovations. The shape of the hall is nearly rectangular with a semicircular rear wall. Three shallow balconies ring the center seating area. The lowest is subdivided

FIGURE 19.20 McDermott Hall, Dallas, Texas, USA (Beranek, 1996)

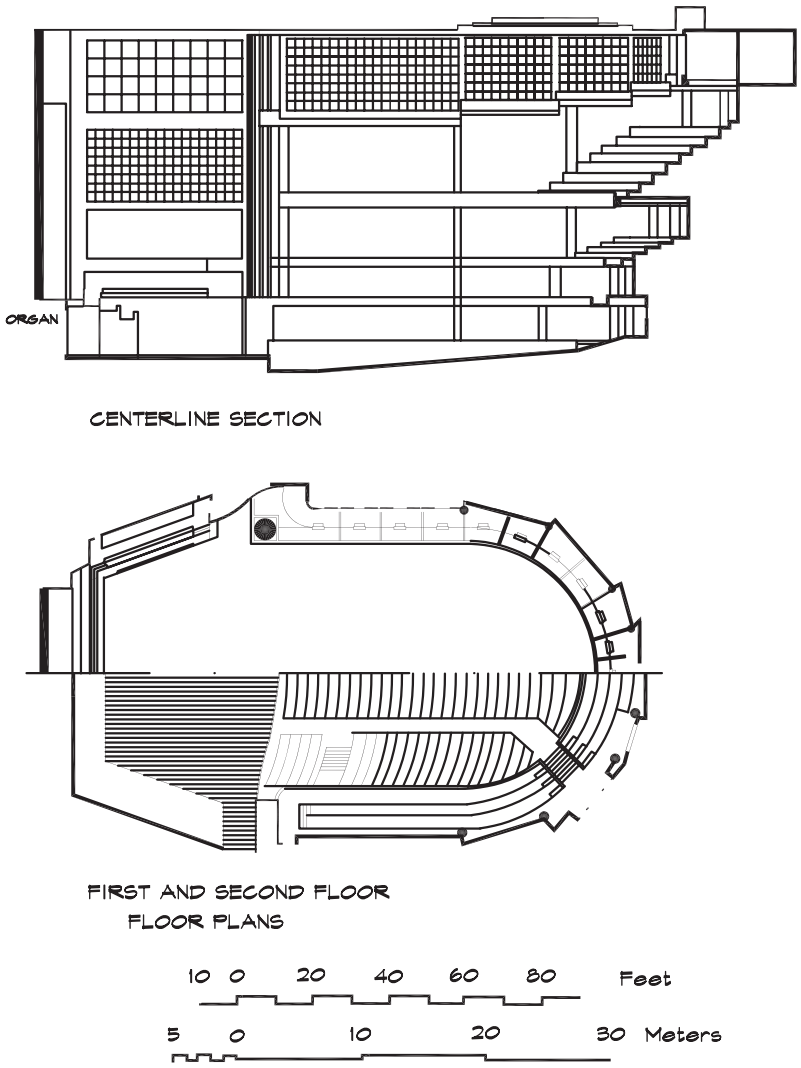
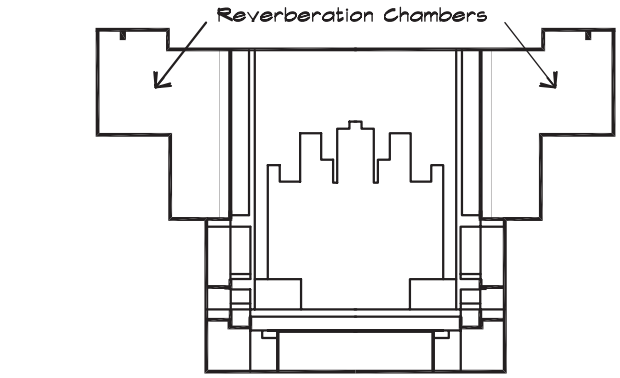
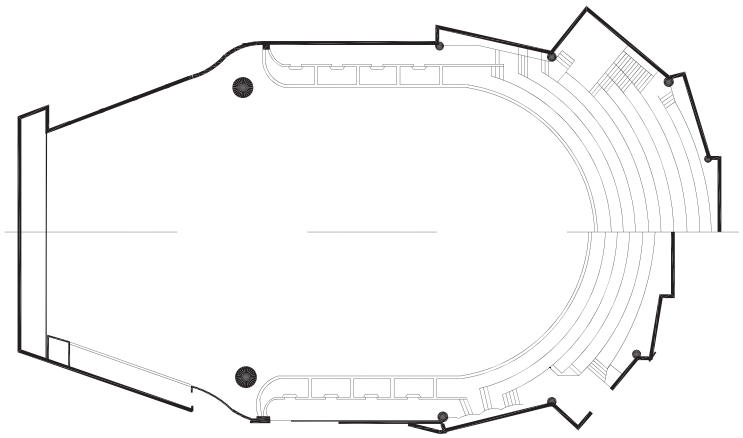


FIGURE 19.20 (Continued)



LATERAL SECTION



BALCONY PLANS

TABLE 19.12 Selected Technical Data for McDermott Hall (Beranek, 1996; Berkow, 2001)

Attribute	Metric Units	FP Units
Volume doors closed (V)	23,900 cu m	844,000 cu ft
Dimensions (LWH)	30.8 m × 25.6 m × 26.2 m	101 ft × 84 ft × 86 ft
Furthest Seat (D)	40.5 m	133 ft
Seating (N)	2065	
Total Absorptive Area (S_T)	1,460 sq m	15,718 sq ft
V/N	11.6 cu m	409 cu ft
V/S_T	16.4 m	53.7 ft
T_{60}(Occ) = 1.5 to 4.5 s	T_E(Unocc) = 0.8 to 1.5 s	

Acoustician: ARTEC Consultants; Architect: Pei Cobb Freed & Partners

into boxes, while the highest rises upward from the center band of seats. The ceiling is very high at 86 feet (26.2 m) and nearly flat. For all its size, the hall seats just over two thousand people. The volume in the main hall is augmented by reverberation chambers, attached like mouse ears out of sight on the upper sides of the auditorium. These can add an additional 254,000 cu ft (7200 cu m), or 30%, to the room by opening up to 74, 4-inch (10 cm) thick mechanized concrete doors.

The main visible feature of the hall is a flying reflector suspended on cables above the orchestra platform, reminiscent of the Starship Enterprise. This serves to provide reflections back to the orchestra from a height that is adjustable, varying between 36 and 50 feet (11–15 m). For small groups, the intimacy can be increased by lowering the canopy. For organ music, where a highly reverberant sound is required, the canopy is raised, exposing the organ pipes behind. The moveable canopy has become a trademark solution to the overhead reflection problem developed by the acoustician Russell Johnson. It is so large that it subdivides the room into three volumes: below the canopy, the main space, and the reverberation chambers. This leaves the decay pattern a complicated merger of the three coupled spaces, but offers a promising solution to the problem of simultaneously achieving clarity and reverberation. The reverberation time can be varied over a wide range, but interestingly the early reverberation time is relatively short.

For all its technical innovation most of the hall's features are nicely blended into the architecture. The floor is terrazzo tile or painted concrete, while the ceiling is 5.5 in (14 cm) concrete. The walls are wood veneer on 0.5 in (13 mm) particle board bonded to 10 in (25 cm) masonry. The 4000 sq ft (372 sq m) canopy is built from 6 in (15 cm) laminated wood. Some of it is covered with 0.06 in (1.6 mm) felt to reduce glare. The stage is 0.5 in (13 mm) tongue and groove wood on wood boards on joists. The floor under the cellos and double basses is 2 in (5 cm) wood over a 3 ft (0.9 m) airspace.

Beranek (1996) found the balance, tone, and intimacy excellent nearly everywhere, best on the main floor and in the rear. The side balconies were less well balanced. He described the effect of the reverberation chambers as "not obvious," noticeable only after stop chords but not as running reverberance. All in all this is a very interesting hall.

19.5 OPERA HALLS

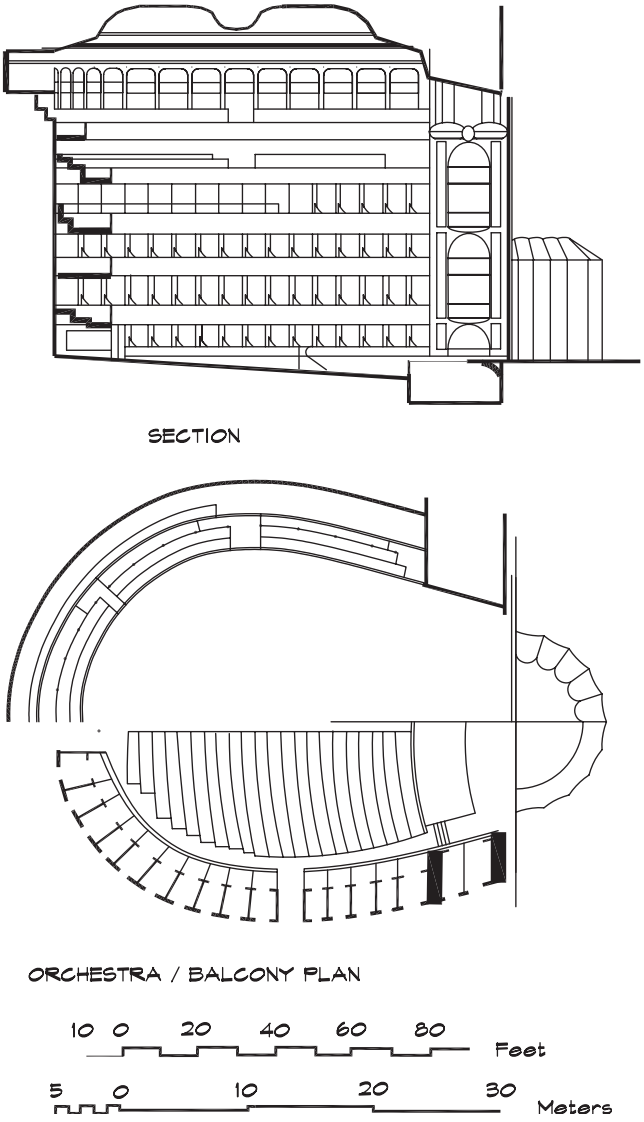
Opera halls and their history were discussed in detail by Forsyth (1985), and their acoustical attributes were surveyed and analyzed by Beranek (1962 and 1996) and Barron (1993). The following data are due primarily to the compilation of Beranek.

Theatro Colon, Buenos Aires, Argentina

Theatro Colon in Buenos Aires opened in 1908, as an opera house, but is also a highly rated concert hall. In his 1962 work Beranek found it to be in his top six concert halls. Its reverberation time is high for opera, at about 1.8 s (occupied), but quite appropriate for concerts. The shape is a classic horseshoe, seating 2487 people with six shallow balconies shown in Fig. 19.21.

As an opera house it is a favorite of singers because of the support they receive on stage. The side walls on either side of the forestage contribute early reflections as do the ceiling and vertical balcony facings. The pit is 3 m (10 ft) deep and so the best string sound is heard in

FIGURE 19.21 Theatro Colon, Buenos Aires, Argentina (Beranek, 1996)



the upper balconies. In the concert mode the pit is covered and a semicircular shell formed of convex scallops is installed. The walls and ceiling are 1" (2.5 cm) plaster on wire lath. The floors are wood with carpet in the aisles.

Theatro Alla Scala, Milan, Italy

Theatro Alla Scala, shown in Fig. 19.22, opened in 1778 and since then has been the best known opera house in the world. Although it was nearly destroyed in World War II, it was

TABLE 19.13 Selected Technical Data for Theatro Colon (Beranek, 1996)

Attribute	Metric Units	FP Units
Opera		
Volume (V)	**20,570 cu m**	**726,300 cu ft**
Dimensions (LWH)	**34.4 m × 24.4 m × 26.5 m**	**113 ft × 80 ft × 87 ft**
Furthest Seat (D)	**43 m**	**141 ft**
Seating (N)	**2487**	
Total Absorptive Area (S_T)	**2,144 sq m**	**23,077 sq ft**
V/N	**8.3 cu m**	**292 cu ft**
V/S_T	**9.6 m**	**31.5 ft**
T_{60} (Occ) = 1.88 s (est)	**t_1 = 19 ms**	
Concerts		
Volume (V)	**21,524 cu m**	**760,000 cu ft**
Total Absorptive Area (S_T)	**1,970 sq m**	**21,200 sq ft**
V/N	**8.67 cu m**	**306 cu ft**
V/S_T	**10.9 m**	**35.8 ft**

Architect: Victor Meano

rebuilt to the same design and reopened in 1946 much as before. It has the classic horseshoe shape with six balconies, subdivided into boxes. Izenour (1977) describes it as the culmination of all that is good, and bad, in Baroque theaters—"a functioning museum." The ornate balcony fronts along with the compound curve of the ceiling provide strong reflections back to the singers on stage, which can be distracting to the audience sitting in the orchestra seats. The side wall balcony facings and narrow width provide a sense of intimacy and a sound that is "clear, warm, and brilliant" (Beranek, 1962) for those in the best seats.

The boxes and horseshoe seating arrangement were designed so that both the stage and the 20-seat royal box at the rear were visible. The front faces are about 56% open presenting bands, alternately reflective and absorbent, since the boxes are deep and well upholstered and little sound emerges. The result is a rich reverberant sound (Barron, 1993) in the center portion of the rear balconies, particularly in the front row. In the side balconies and in the rear of the boxes the sound is less accessible and the sight lines from the side particularly in the upper balconies are poor.

The pit is large and open, extending across the entire front of the stage. There is a little more than 2 m (6.6 ft) of stage overhang. The orchestra rail is slightly lower than the front of the stage, which slopes upward away from the audience. The coffered proscenium is nearly

FIGURE 19.22 Theatro Alla Scalla, Milan, Italy (Beranek, 1996)

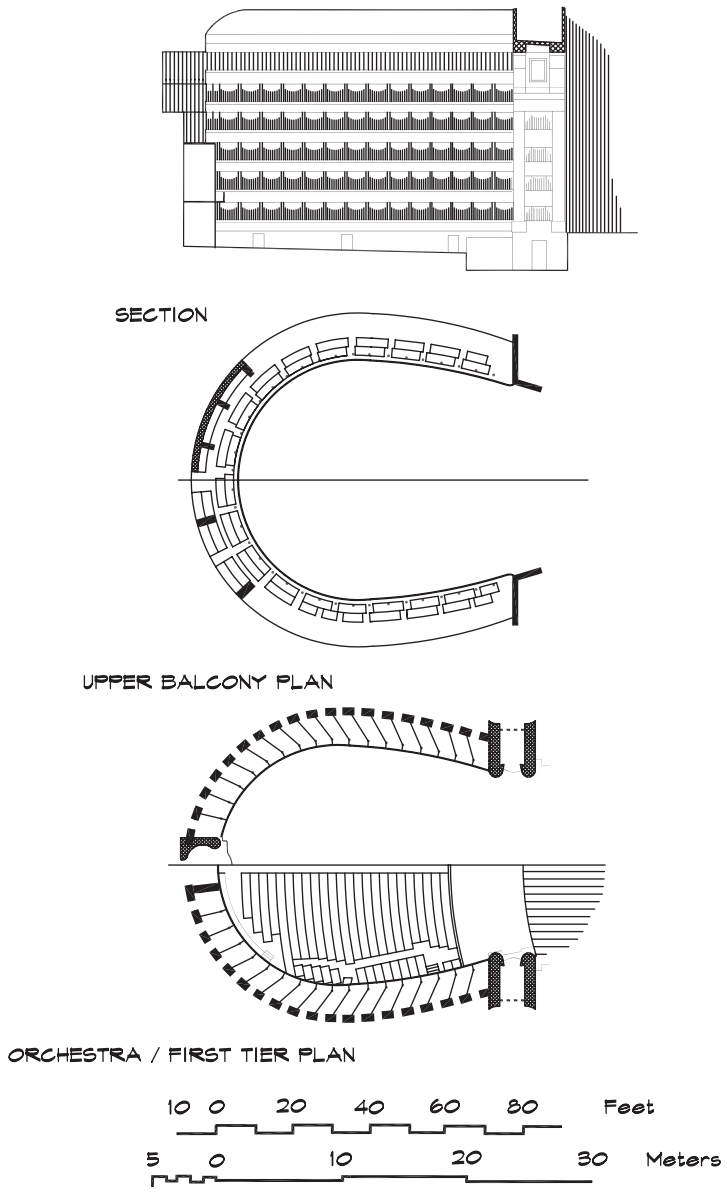


TABLE 19.14 Selected Technical Data for Theatro Alla Scala (Beranek, 1996)

Attribute	Metric Units	FP Units
Volume (V)	**11,252 cu m**	**397,300 cu ft**
Dimensions (LWH)	**30.2 m × 20.1 m × 19.2 m**	**99 ft × 66 ft × 63 ft**
Furthest Seat (D)	**32 m**	**105 ft**
Seating (N)	**2289 (including 154 sight restricted seats)**	
Total Absorptive Area (S_T)	**1,635 sq m**	**17,600 sq ft**
V/N	**4.92 cu m**	**174 cu ft**
V/S_T	**6.9 m**	**22.6 ft**
T_{60}(Occ) = 1.2 s	**T_E(Unocc) = 1.2 s**	**$(1 - IACC_{E3}) = 0.48$**
t_I = 20 ms	**$G_{mid} = -1.4$ dB**	**BR = 1.21**

Architect: Giuseppe Piermarini

horizontal above the orchestra, providing some additional support for the orchestra, though at a delay of more than 100 ms.

The walls and ceiling are plaster on wood frames. The vaulted ceiling is nearly flat with no diffusing elements. The floors are wood on a 3 ft (0.91 m) airspace on concrete and are carpeted. The stage and pit floor are wood.

This page intentionally left blank

20 DESIGN OF MULTIPURPOSE AUDITORIA AND SANCTUARIES

The acoustical properties that make a room good for speech are often the same as those that make it poor for music, and vise versa. For good speech intelligibility, room volumes and reverberation times should be low. The first reflections should be primarily from the ceiling, and there is little need for diffusion. Conversely, rooms designed for listening to unamplified music require longer reverberation times, higher volumes, lateral rather than overhead reflections, and high diffusion. Rooms designed for mixed uses require a judicious compromise between the needs of speech and music. Buildings of this type, including auditoria, theaters, churches, and synagogues, are among the most architecturally diverse of all spaces and the most challenging to design.

20.1 GENERAL DESIGN CONSIDERATIONS

Although there are design features unique to the type of room, there are also general guidelines for large mixed-use facilities that apply to all of these rooms.

1. The source should be elevated above the seated audience so that sight lines are clear of obstruction.
2. A sound system capable of reproducing the full program frequency range must be incorporated into the design. The loudspeaker layout should be arranged so that the talker is the perceived origination point.
3. Where unamplified music is a part of the program, the singers or musicians should be aided by nearby reflective surfaces both overhead and on either side. These can be integrated into the design of the structure, suspended above the players, or built into a moveable shell. Musical instruments supporting the singers should be located near them, so there is no great time difference between the two.
4. The floor should be raked to provide every other row sight lines of the APS.
5. The room volume and absorption should be controlled to achieve a reverberation time consistent with the program. Padded seating must be used to minimize the differences between the empty and occupied conditions. Room reverberation must be controlled to limit the loudness of large groups.

6. Background noise levels should be limited to NC 25 in small (< 500 seats) auditoria and to NC 20 in opera houses and drama theaters. In large concert halls it should be limited to NC 15.

Program

Architectural designs begin with a program, listing the various uses contemplated for a space and the percentage distribution of each. Program definition allows the architect and acoustician to know whether to steer the design compromise toward a speech-like or music-like solution. Not infrequently, in the design of performance spaces, there may be disagreements as to the proper direction, not only within the design team, but also among the members of the client team. It is always best to sort this out in writing early in the process so that the direction is clear.

Changes to the program, particularly late in the process, can have profound consequences. Hidden agendas are difficult to uncover. This is particularly true when there is an expectation that the room will serve a purpose different from its stated use. For example, a church may be designed as a worship space, but may also be expected to serve as a theater, performing arts center, and television studio where there are significant additional lighting, rigging, audio, and HVAC requirements. A well-crafted program can allow these issues to be sorted out and can help establish a clear design direction.

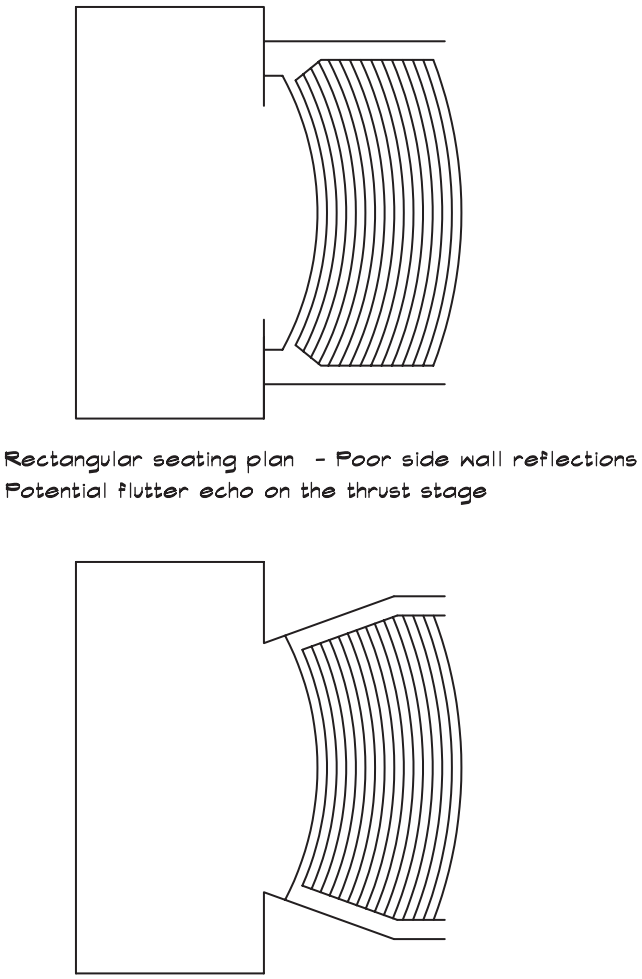
Room Shape

Floor plans are laid out on the basis of the program, which drives the seating configuration, volume, sight lines, circulation, and aesthetics. In rooms designed for speech it is good practice to bring the audience close to the talker. A fan-shaped room, particularly one with balconies, allows a larger number of people to be close to the stage, than a rectangular room. For unamplified music, where strong side reflections are important, a narrow rectangular room with a high ceiling has traditionally yielded the best result. For rooms of mixed use a compromise between the speech and music requirements must be crafted.

Small general-purpose auditoria are typically rectangular, with the walls near the front of the room angled out from the stage. For lecture halls more of a fan shape is preferred, with the maximum angle between opposing walls no more than 140°. In the case of mixed use spaces the included angle falls between 40° and 80°, with smaller values being best for music. Examples are given in Fig. 20.1. The side walls at the rear of the room are either parallel or flared out at a shallower angle than in front. This scheme provides useful early reflections from the walls near the stage or platform, while allowing regular aisles and constant width seating sections at the rear.

In large multiuse theaters it becomes increasingly difficult to accommodate the audience with good sight lines. A fan shape is the usual choice, truncated on the sides, with one or more balconies. The distance to the farthest viewer depends on the type of performance. The human eye can perceive an object as small as one minute of arc (Burris-Meyer and Cole, 1975) or about 0.35” (9 mm) at 100 ft (30 m). As the last row is pushed back to 135 feet (41 m), the smallest resolvable dimension is increased to 0.5” (13 mm) and a raised eyebrow or small gesture is lost. Thus if the subtle facial expressions of a theatrical performance are to be appreciated, the farthest patron should be seated no more than about 80 feet away.

FIGURE 20.1 Side Wall Design Near the Stage



This is a more restrictive requirement than seating for orchestral performances, and limits first floor seating to around 26 rows or 1100 seats for three banks of fourteen seats.

In very large spaces the seating fan is forced to a greater angle often with four seating banks with a center aisle or three banks with a half bank on each outside wall. In the Radio City Music Hall in New York for example there are five seating banks. The remainder of the audience is seated in the balconies, which must not cause shadowing through excessive overlap of the seats below. It is difficult to design auditoria for intimate theatrical productions larger than about 1500 seats. In large houses the types of performances must be adjusted to the scale of the space.

The shape of worship spaces is determined primarily by liturgical requirements. Churches require a focal point either at the front or in the center. A raised platform helps acoustics as well as sight lines, but if it is too high there may be unshielded sound paths to the

side or end walls, leading to long-delayed reflections and flutter echo. A cruciform shape may be highly desirable in some denominations. Medium-sized congregations, in the 500 to 1000 range, can be accommodated in a wide variety of room shapes, but care must be exercised not to let the room volumes get out of hand. Very large churches in the 2500 to 10,000 seat range are fan-shaped for the practical reason that this is the only way to accommodate the necessary seating. To maintain sight lines the included fan angle should not exceed 160°. These large venues rely on electronic sound reinforcement and video projection systems to transmit their message, so the sight lines to these sources are also important.

Religious and theatrical facilities are sometimes designed with circular or multifaceted hexagonal or octagonal floor plans. Such shapes should be avoided wherever possible since they focus sound and create localized regions of high level. Churches and theaters in the round, even with diffusive or absorptive walls, are also poor acoustical choices. At best they result in fair-to-mediocre sight and sound quality in spite of the expenditure of great effort and treasure.

Worship spaces have higher ceilings than auditoria and a stately architectural form, often with historical roots. Ceilings of religious structures can be dome-shaped, which generate focused reflections. When long delayed reflections arrive back at the talker at an elevated level, the effect can be particularly unpleasant. There is a combination of level and delay time (around 100–150 msec) so disconcerting to the talker, that, in the worst case, it can act as a severe impediment to normal speech. Large concave shapes should be avoided, particularly when the talker and listener are near the center of curvature. A calculation of the curvature gain can identify problem conditions. Curved walls can be treated with absorption, however since the normal-incidence absorption coefficients are lower than random-incidence values, there may still be reflected levels high enough to be of concern

Seating

Circulation and building code requirements influence the shape of the floor plan. In the conventional seating plan US building codes require that there be no more than 6 seats between a seated patron and an aisle, permitting a maximum of 14 seats in a section bounded by two aisles. Normal row-to-row spacing is 36 to 38" (0.92 to 0.97 m). In continental seating the larger distance between rows (40–42" or 1.01–1.07 m) allows them to be considered cross aisles and the six-seat limit no longer applies. Opera chairs are available in various widths ranging from 19 to 23" (0.48 to 0.58 m). Normal seats are about 22" wide (0.56 m) with some at 21" (0.53 m) to accommodate the stagger. A full 14-seat bank occupies 25.5 ft (7.8 m) plus the aisles.

The decision on whether to have a center aisle is influenced by the program. In religious structures a center aisle is the usual choice to provide for a processional march in traditional wedding ceremonies. In a theater the best seats are located in the center so an even number of aisles is a better alternative. Building codes in the United States specify minimum 36" (0.9 m) single-sided and 42" double-sided aisles, which widen 1.5" (38 mm) for every five feet of seating as they approach the nearest exit.

Handicapped seating requirements call for wheelchair spaces along with adjacent companion seats within the space. Recent changes to the Uniform Building Code to accommodate handicapped access can limit floor slopes to 1:12—less than ideal for every other row sight lines depending on the height of the APS. Requirements vary and enforcement of these codes differs. It is always best to get clear directions from the local building officials.

TABLE 20.1 Range of Volume per Seat by Type of Auditorium (Doelle, 1972)

Type of Auditorium	Volume per Seat cu ft (cu m) Min	Mid	Max
Rooms for Speech	80 (2.3)	110 (3.1)	150 (4.3)
Concert Halls	220 (6.2)	275 (7.8)	380 (10.8)
Opera Houses	160 (4.5)	200 (5.7)	260 (7.4)
Churches/Synagogues	180 (5.1)	255 (7.2)	320 (9.1)
Multipurpose Auditoriums	180 (5.1)	250 (7.1)	300 (8.5)
Motion-picture Theaters	100 (2.8)	125 (3.5)	180 (5.1)

Room Volume

Room volumes are selected using the volume per seat as a guideline. Since the ideal reverberation time for speech is low, the preferred volume per seat for speech is also low, ranging from 80 to 150 cu ft (2.3 to 4.3 cu m), with smaller auditoria having the larger volume per seat. For unamplified music it can vary from 160 to 400 cu ft (4.5 to 11.3 cu m), again with smaller halls having the greater volume per seat. In rooms designed for mixed uses, the volumes fall between these values, 180 to 300 cu ft (5.1 to 8.5 cu m). Table 20.1 shows typical ranges by room use.

Reverberation Time

From the program, the designer can establish the reverberation time and a preferred volume per seat for the space. The seating capacity will then fix the overall volume of the room and the circulation and other functions will lead to a preliminary floor plan. Figure 17.10 shows reverberation time versus room volume values for various types of spaces. Figure 17.11 shows the preferred behavior of reverberation time with frequency for music. In small auditoria and churches the overall volumes are rarely high enough to raise the bass response to the level shown in this figure and the alternative, which is using heavy plaster or multiple layers of drywall, is cost prohibitive except in pure concert halls. As rooms grow larger a high volume can be used with applied absorption to control the mid-frequency reverberation. The reverberation time falls off naturally at high frequencies due to air absorption and thin absorbent materials. By using wood slats over absorptive panels it is possible to offset these effects and raise the high-frequency reverberation time. This provides a favorable frequency balance as long as the absorption is needed anyway at other frequencies.

Absorption

In large auditoria and religious structures the biggest absorbing surface is the seated audience. It provides about 85% of the total absorption in concert halls and somewhat less in auditoria where applied absorptive material may be used. Since it is desirable to maintain a consistent acoustical environment no matter the number of people, it is most important to use seats, whose absorption characteristics closely resemble that of a seated occupant. This requires that the chairs or pews have thick padding on both the seats and backs. Beranek (1996) observed that the absorptive properties of a given number of seats depends not on the number of seats,

but on the area they cover. Thus reverberation calculations should be based on an absorption coefficient per unit area of seating, not on the number of sabins per seat.

As sound propagates over the seats of a theater it is found that there is an extra attenuation above that expected from distance and grazing losses. This extra attenuation is centered on 125 Hz and appears as a deep dip in the attenuation vs frequency response curve. The phenomenon has been discussed in earlier chapters and measured data are shown in Fig. 7.44. Ando (1985) analyzed the effect and related it to the quarter-wave well impedance caused by the depth of the seats. He proposed various solutions including cavities in the floor and several geometrical alternatives. Unfortunately, there is little measured data available to determine the effectiveness of his suggestions.

When the natural acoustics of a room is augmented by a sound system and design constraints allow the use of absorbent surface materials, the room volume and surface orientations are less critical than in concert hall environments. In these cases absorptive materials can be added to control unwanted reflections and reverberation since the level generated by the sound system can make up for the energy lost through absorption. In rooms accommodating both music and speech, the ideal solution is to design to the reverberation time appropriate for the music, and to manage the intelligibility using the reinforcement system. Where unamplified speech must be supported, variable absorption can be a good alternative.

Balconies

Balconies allow more of the audience to be seated close to the stage and limit the slope of the orchestra seating. Balcony floors have steeper rakes than the orchestra floor, but the slope can be calculated using the same equations. The slope should not be greater than 30° (Ramsey and Sleeper, 1970) to 26° (Egan, 1988), and the top of the balcony should not be more than 65 feet above the stage to avoid vertigo (Egan, 1988). Figure 20.2 shows a typical balcony configuration.

Balcony overhangs must be controlled to allow the reverberant sound to reach the seats beneath them. Beranek suggests a depth-to-opening ratio of one for concert halls and two for opera houses. For speech, balcony overhangs can be deeper without undue degradation. A two-to-one depth-to-opening ratio is acceptable.

A slightly convex under-balcony ceiling can help redirect sound into the shielded area. Likewise a rising leading edge at the front of the balcony is also helpful. Where the

FIGURE 20.2 Balcony Design

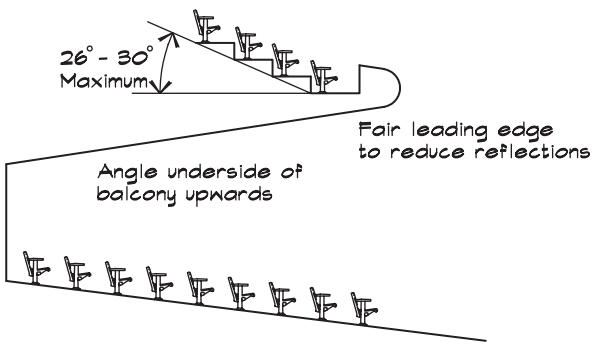
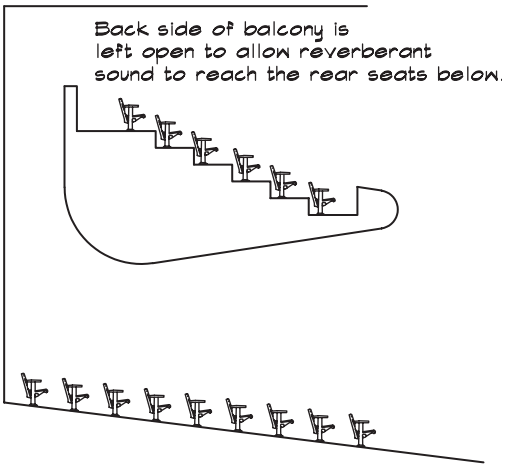


FIGURE 20.3 Flying Balcony Design



balcony overhang is very deep, both direct and reverberant sound have difficulty penetrating. A sound system can augment the direct sound, using loudspeakers located on the underside of the balcony. A concave or semicircular-ceiling cavity under the balcony can help generate a localized reverberant field to offset the effect of a deep overhang.

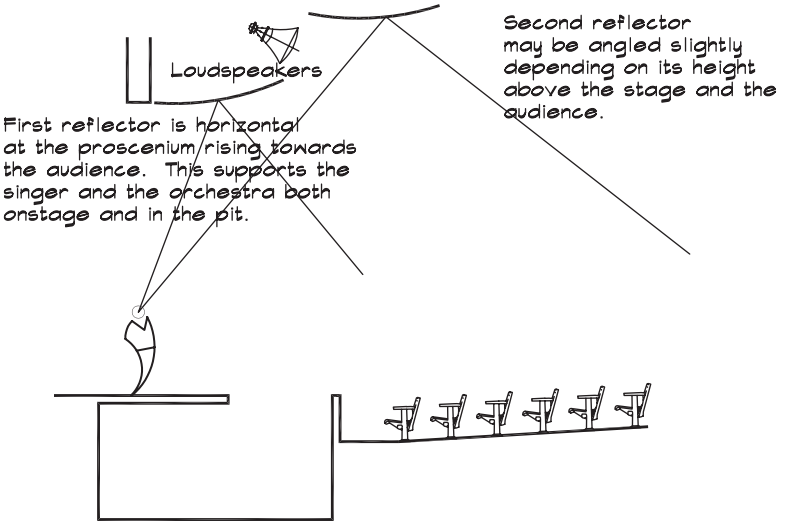
In rare instances, halls have been built with a flying balcony, separated from the rear wall of the room as in Fig. 20.3. This allows the sound to flow around the upper balcony to reach the rearmost under-balcony seats. It is an expensive solution, since it presents structural challenges, even when beams support the balcony from the rear.

Ceiling Design

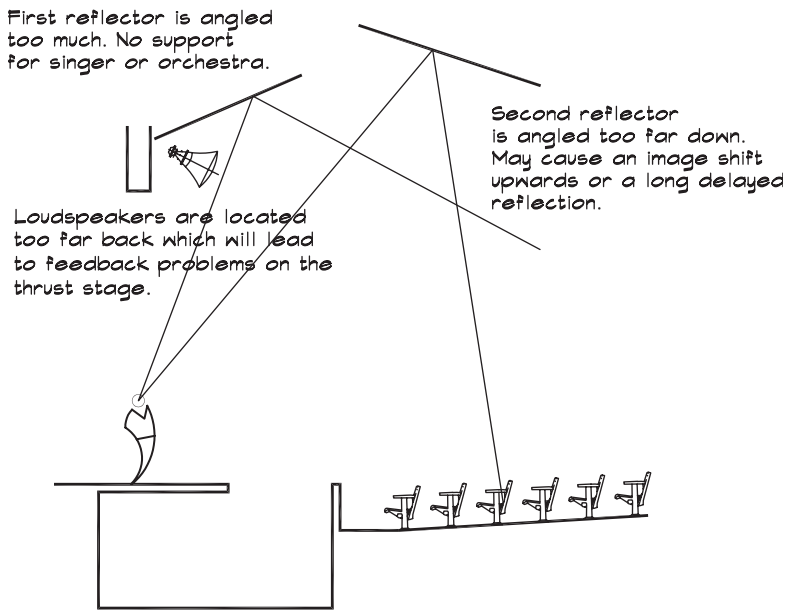
The ceiling of an auditorium or theater is more complicated than those found in a lecture hall. When auditoria are designed for speech, strong overhead reflections are preferred, whereas for music, a ceiling that diffuses the sound and aids in the sense of envelopment, or feeling surrounded by the sound, is best. A flat ceiling can yield excellent results for speech if it is not too high and if the floor rake is sufficient. In auditoria and lecture halls, particularly when the floor is flat or slightly raked, a shaped ceiling is helpful. This type can be designed by dividing the surface into angled planes or separately supported reflectors, so that there is an even distribution of reflected sound over the audience.

The surface above the proscenium arch is particularly important. Figure 20.4 shows an example. The first reflecting surface should be convex and horizontal at the proscenium wall. Such a shape aides in the reflection of sound to the seats in the center of the orchestra section, which is the most difficult to cover. It is also more forgiving of changes in the source position than a sharply slanted reflector. By controlling the curvature the clusters, loudspeakers can be nested above the reflector where the shielding helps control feedback. One or more rows of theatrical lighting must be accommodated, located at 45° and 55° from a downstage actor. Consequently there is a scramble for the ceiling space in the third of the ceiling closest to the proscenium, which must include passive acoustical reflectors, loudspeakers, catwalks, and theatrical lighting.

FIGURE 20.4 Design of Proscenium Reflectors



AN EXAMPLE OF GOOD PROSCENIUM DESIGN



AN EXAMPLE OF POOR PROSCENIUM DESIGN

Ceiling reflectors can be flat or convex shaped but are not concave or coffered. Some designers prefer to use individual panels, or clouds, since they are more easily adjusted after installation, and are in some cases less expensive. When this type of ceiling system is used, the reflectors should be arranged along a single planar or curved surface so that interference patterns due to Bragg imaging do not occur. Reflectors should be large enough that the frequencies of interest are scattered more or less specularly. Using a group or array

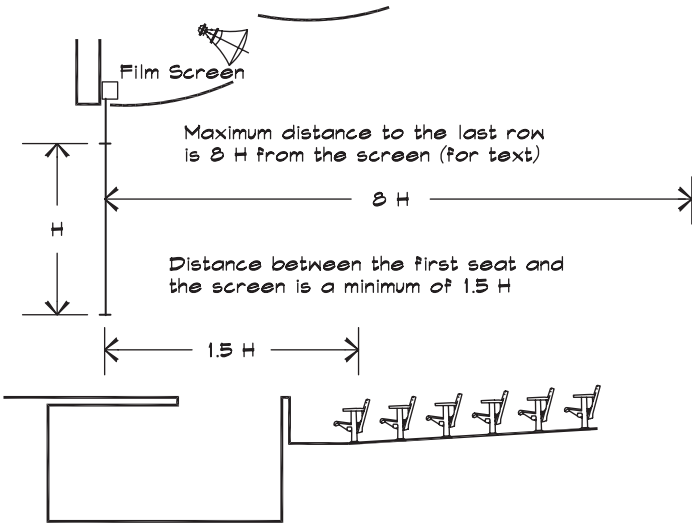
of reflecting panels can help offset this limitation and assure that low frequencies are also reflected (Leonard, Delsasso, and Knudsen, 1964). In Chapt. 7 we discussed the effects of reflector size on the frequency of the scattered sound. Where ceilings are very high or when there are ceilings at different heights, as is the case in some religious structures, the highest surfaces should be the first choice to receive absorptive treatment. These spaces sound best if the reverberation times are matched. Application of absorption to the whole area may not be required, but if some is treated the rest should maintain a consistent appearance whether treated or not. Where a highly diffuse ceiling is used, clarity is likely to suffer in the rear of the auditorium and in the balcony seats during unamplified use.

Audio Visual Considerations

As auditoria and religious structures become larger, they must rely on audio and video reinforcement systems to transmit the message to the audience. Floor layouts must accommodate sight lines, not only to the talker, but also to video screens and loudspeaker locations. If large loudspeakers are necessary to provide the directivity required for speech intelligibility, then the architectural real estate necessary to accommodate them must be included in the design. Loudspeakers should be arranged so that the source image is preserved, and frequently a central cluster above the proscenium is the best solution. When this does not match the visual expectation of the design team, enclosures can be incorporated to soften their visual impact.

If there are projected images in a theater, church, or lecture hall then the seating layout must accommodate the requirements of the screens. Figure 20.5 shows seating arrangements based on front and rear projection screens. The front row of seats should be no closer to the screen than one and one-half screen heights (1.5 H) or about one and one-eighth screen widths (1.125 D). The rearmost seat should be no more than 8H or 6D from the screen. In some cases screen sizes become very large and multiple screens must be used. This is particularly true in large fan-shaped churches where no single screen can be seen by the entire congregation.

FIGURE 20.5 Design of Projection Screens



20.2 DESIGN OF SPECIFIC ROOM TYPES

Small Auditoria

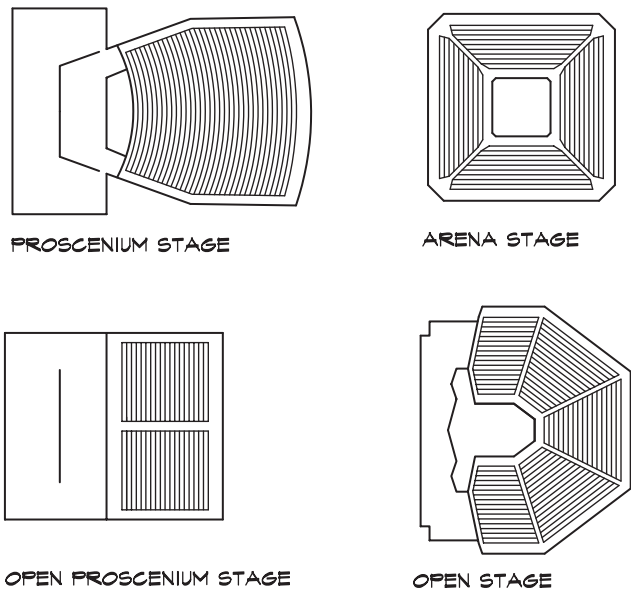
The acoustical design of an auditorium is more complicated than that of a lecture hall, however the general principles are very similar. The variety of seating layouts, wall shapes, and ceiling designs is much greater in theaters than in lecture halls. Figure 20.6 shows several theater-seating arrangements, including the traditional proscenium stage, an open stage, thrust stage, and an arena stage. Of these, the proscenium stage arrangement is the most serviceable.

In a proscenium theater the audience views the actors on stage through the picture frame of the proscenium opening. In the United States a stage must be separated from the audience area by means of a fire curtain, which descends to cover the opening in case of a fire. Stage widths vary from about 40' to 50' (12 m to 15 m), with heights ranging from a minimum of 18' to about 30' (5.5 m to 9 m). A 2:1 width-to-height ratio is fairly typical. Very high proscenium openings are trimmed down with teaser curtains.

If the auditorium is to be used for orchestral or choral concerts, a shell is important, and the side walls of the room should merge smoothly with the walls of the shell. Where there is an orchestra pit, it should be convertible into a thrust stage or a place for additional seating. Every theater should be able to accommodate some kind of thrust stage or one will eventually be built over the seats.

Side wall designs in legitimate theaters are less critical than ceiling designs, particularly when the floor plan incorporates fan-shaped, black-box, or arena seating. In proscenium theaters, the lower side walls are left reflective, and the rear wall is treated with absorption. Upper side walls can also be treated, depending on the overall reverberation time requirements. It is useful to provide reflecting surfaces on the side walls close to the proscenium so

FIGURE 20.6 Basic Stage Forms Used in Theater Design



that when the actors face away from the audience, their voices are reflected back toward the seating area.

Rear walls in theaters and small auditoria are almost always absorptively treated. This is the case even with small recital halls, where high-reverberation times are desirable and difficult to obtain, since excessive loudness results from a hard back wall. Even if the program lists only small-group performances, eventually a small orchestra will be featured, which will overwhelm the space.

In virtually all theaters of more than 150 seats, a sound-reinforcement system is used. It is not uncommon to have radio mics on actors even in middle-school productions. Live theater is best served by a mixer located within the audience space. The front of the balcony or the rear of the orchestra seats are the preferred locations. Second best is a mixer located in a booth, with a minimum 6 ft (1.8 m) wide by 4 ft (1.2 m) high open window directly in front of him. For this width, double three-panel telescoping windows work well. In areas where security is a concern, a closeable window allows the technical booth to be locked.

Figure 20.7 shows a small auditorium at the Meadows School in Las Vegas, NV. Meadows combines a number of important features previously discussed. One unique element of the design is the cross aisle, which is at the same elevation as the stage. This is a convenient solution to the handicap-access problem both for the seating area and for the stage. It also allows flexibility in the staging of productions, graduations, and lectures. The walls are angled along the sides to merge with an orchestra shell and support wood diffusing elements.

The seating is easily accessible and provides excellent sight lines. The stage is relatively large with nearby access to the catwalks and the pit. Scenery is suspended from tracks running laterally across the stage. The lighting catwalks are convenient and do not require ladders for access, an important safety feature in school facilities. Side lighting platforms are positioned above the entry doors. Tracks for absorptive curtains run along the rear of each catwalk, above the audience, making the reverberation time somewhat variable. The pit is large enough to be used as a teaching and storage space and can be closed off to provide a thrust stage. An elevator is available for freight and passenger transportation to the pit. A two-tiered technical booth houses audio and lighting control on the lower level and follow spots on the upper level. The large technical-booth window is openable for mixing. Overhead panels give good support for early reflections and are coordinated with the design of the loudspeakers and catwalks. The rear wall is covered with 2" (50 mm) cloth-covered fiberglass panels.

Another interesting school auditorium, shown in Fig. 20.8, is Rugby Hall at Harvard-Westlake School in North Hollywood, CA. It is a small theater, which was remodeled from a flat-floored assembly room, originally at the level of the lobby, into a multipurpose auditorium. During the reconstruction the floor was dug out and the front lowered to provide a sloped-seating area and stage. The side walls were slanted in at the top and facets incorporated to provide early reflected sound. The ceiling was shaped for beneficial overhead reflections, while accommodating the existing steel beams and providing for an exposed lighting catwalk. The catwalk is sufficiently open that the sound system fires through it without noticeable loss, and is accessible from ladders on the sides.

Since this facility was built within an existing building shell, on a limited budget, there was no opportunity to include an orchestra pit. The orchestra is located either on the flat floor in front of the stage or on a temporary loft constructed about 15 ft (4.6 m) above the rear of the stage. This latter approach addresses the balance problem and provides reasonable blend,

FIGURE 20.7 Meadows School Auditorium, Las Vegas, NV, USA (Acoustician: Marshall Long Acoustics) (Architect: Killefer, Flammang, Purtil)

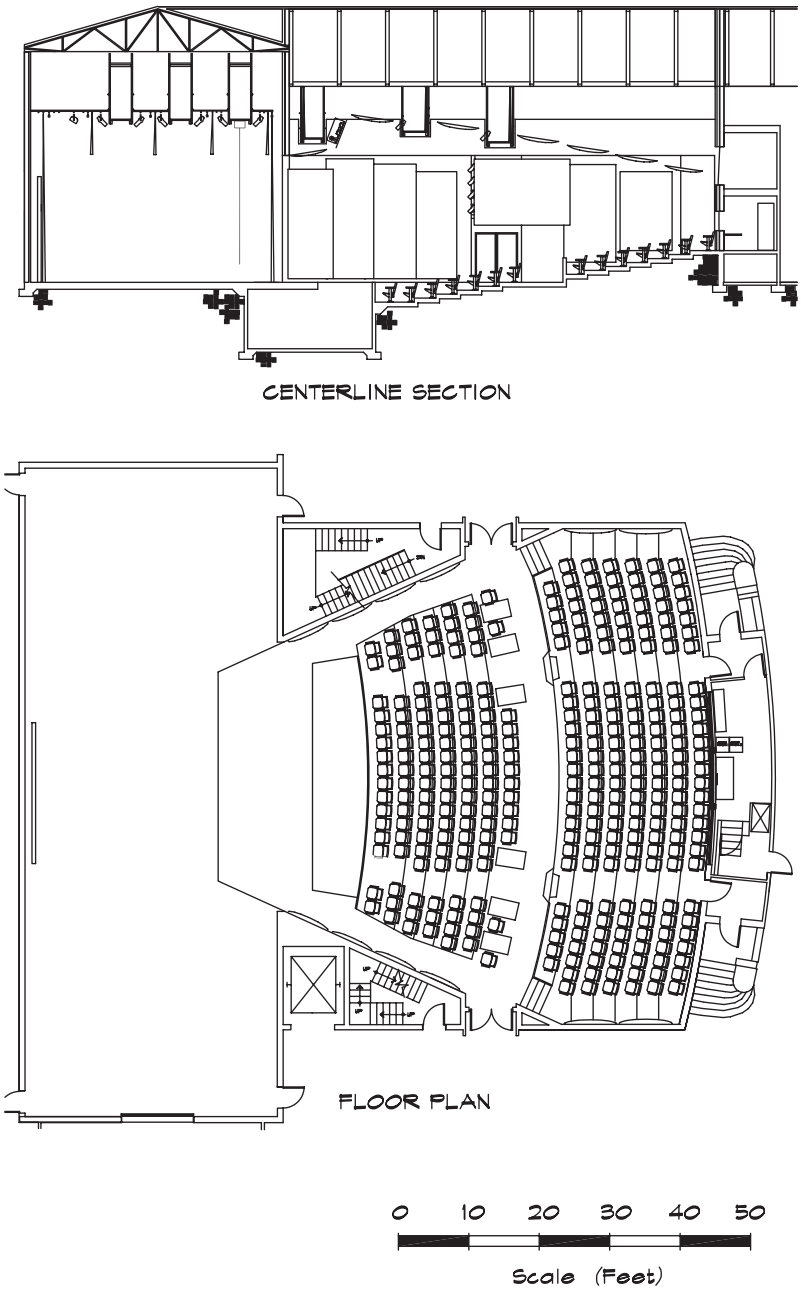
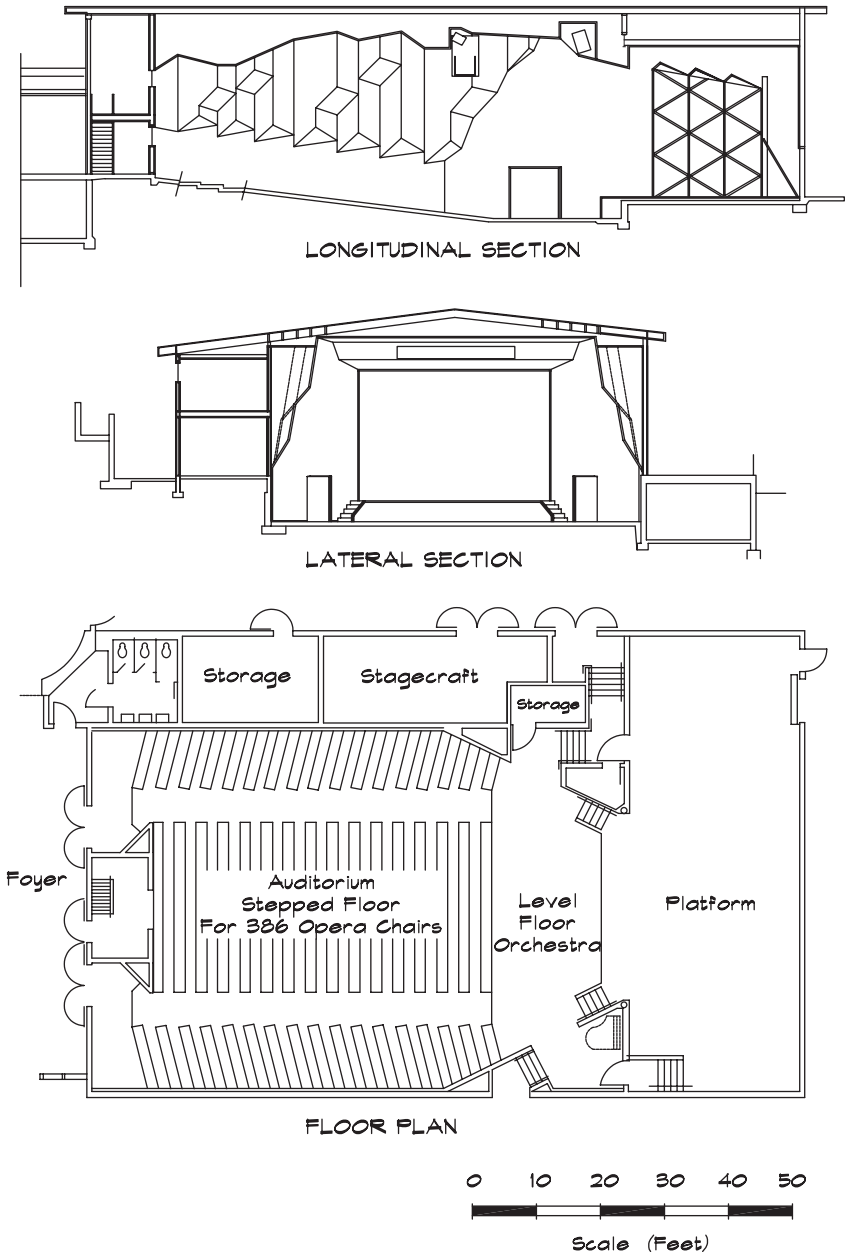


FIGURE 20.8 Rugby Auditorium, Harvard Westlake School, North Hollywood, CA, USA
(Acoustician: Marshall Long Acoustics)
(Architect: Johnson and Sylvestri)



since the ceiling of the platform is close by. It removes some of the upstage area from use, so a thrust stage is built onto the flat floor area.

The design of the side walls is interesting. The bottom portion is flat and provides good support for speech. The lower wall facets are angled to give additional early reflections, while the upper walls are highly diffusive. The result is a dramatic jewel-like appearance, which was relatively easy to build in spite of its apparent complexity. The rear wall is made of wood slats over a cloth facing and a 1" (2.5 cm) fiberglass panel. The hall provides good support for both speech and musical performances.

Mid-Sized Theaters

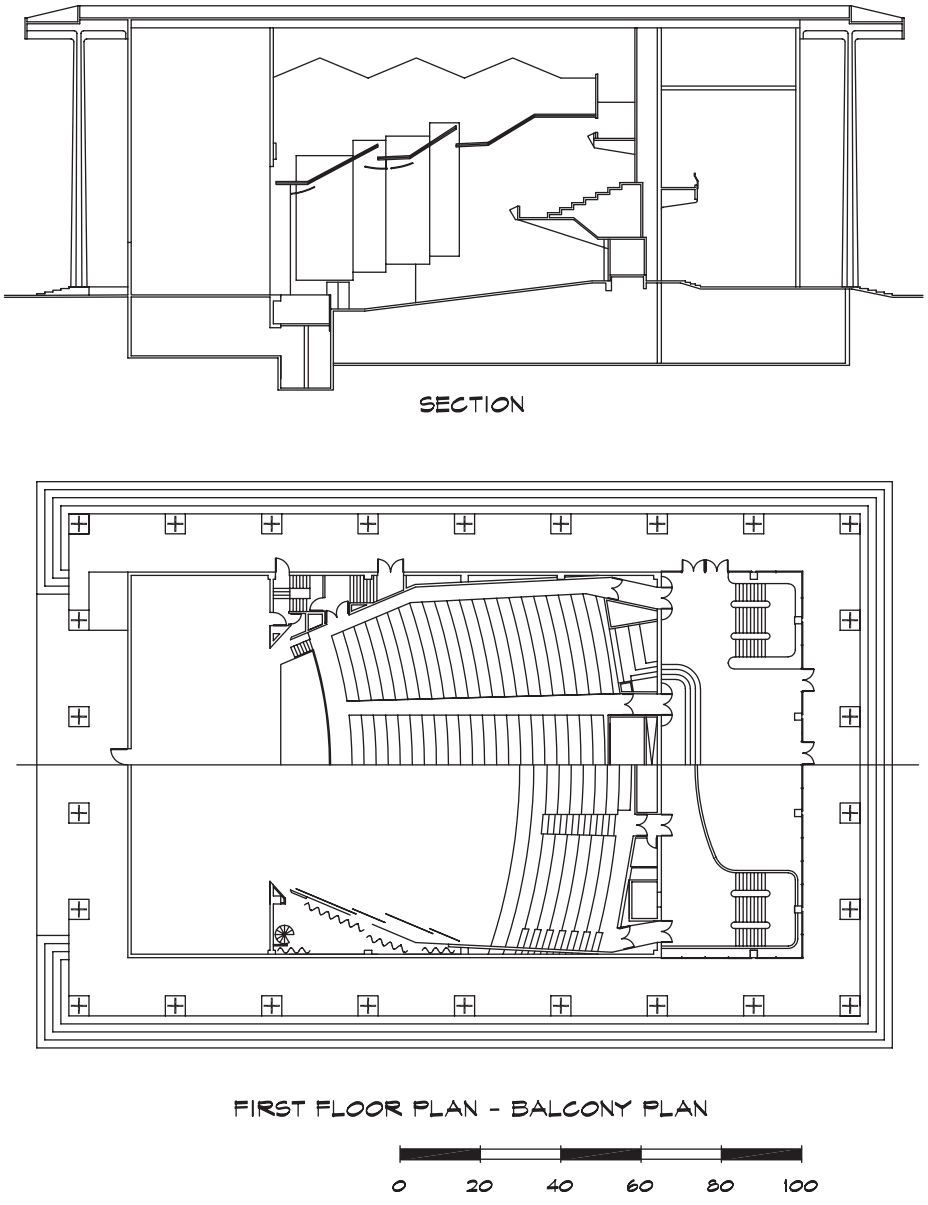
An auditorium that serves a community as legitimate theater, lecture hall, and chamber orchestra hall, should accommodate the needs of both speech and music. Mid-sized auditoria follow many of the same design guidelines as small ones. With moderate seating requirements, these facilities can be designed to provide an excellent environment for both theater and orchestra. First-floor seating can be limited to 900 to 1000 seats, yielding very acceptable sight lines, with the remainder in the balcony. A good example is Ambassador Auditorium in Pasadena, CA, which seats 1250. It is a medium-sized hall, used for chamber orchestra, dance, small musical groups, and lectures. A plan and section are shown in Fig. 20.9. The design of the hall is subtle and interesting. The side walls fan outward in two steps and are faced with a beautiful wood finish. Above the lower walls is a large perforated metal enclosure on each side, overhanging the side aisles, painted in a textured wood finish. It houses large absorbing curtains, out of sight of the audience, to provide variable control of the reverberation. The rear wall is faced with the same perforated metal, over absorptive panels.

The visual ceiling is formed by gold anodized metal bars running along the length of the auditorium. It is transondent, but visually masks an upper ceiling formed from diffusive v-shaped plaster ridges running across the hall. The lighting catwalks and ductwork are exposed between the two ceilings but are painted black. Hanging below the lower ceiling are gold-colored reflective shields in rows across the hall. Two of these rows are located permanently above the audience. Two others are hung above the orchestra, forming a canopy that becomes the open roof of the orchestra shell. These large reflectors give the sound in the hall exceptional clarity, particularly in the orchestra seats. In the balcony the sound character is not as crisp, due to the oblique angle of incidence with the reflecting panels and the reflection from the stage floor, but is still quite clear.

The spacious pit floor, supported on a hydraulic lift, can be raised to form a thrust stage. When the pit floor is set to match the height of the audience floor, it can be used as an extra seating area. It can also be lowered to the normal pit level or still further to a storage level where it can serve as a large freight elevator.

The orchestra shell is constructed from wood panels, with coffered sections faced with wood dowels approximately 1.5" (38 mm) in diameter, spaced about 12" (0.3 m) apart. These do not add appreciably to the diffusion, because of their small size, but they break up the regularity of the coffers and create a pleasant sense of artistic balance. The reflectors above the shell are stored in the fly tower when not in use. They provide good high-frequency clarity on stage; however, since they are open to the space above they do not contain and project the bass. Lack of bass reverberation and envelopment are probably the only weaknesses of this space, which is the best chamber-music hall in Los Angeles and certainly the most beautiful.

FIGURE 20.9 Ambassador College Auditorium, Pasadena, CA, USA (Acoustician: BBN; Architect: DMJM)



Large Auditoria

Large general-purpose auditoria in the 1500 to 2000 seat range must accommodate the full gamut of theatrical activities from orchestral performances to theater, ballet, lectures, and travelogues. The designers face a number of decisions about the shape, volume, reverberation time, clarity, and diffusion. Size magnifies any acoustical defects, so room volumes and reverberation times are best held to the low side of the concert hall range.

The stage should be big enough to handle a full theatrical performance with at least as much wing space as there is stage space. The width of the proscenium in a large theater can range from a minimum of 40 ft to as much as 60 or 65 feet and its height is about half its width. The full proscenium can be trimmed down with top (teaser) and side (tormentor) curtains. Ancillary rooms should include a large shop with easy access to the stage, orchestra and choir rehearsal spaces, and green and dressing rooms.

A fly tower above the stage is a necessity for legitimate theater. Fly towers are 2.5 times the proscenium opening in height so the line sets can be pulled out of sight. The scenery is raised and lowered from the side walls either at stage or catwalk level. The top of the fly tower must be accessible to service the loft blocks and gridiron. The stage catwalk system should be connected to the lighting catwalks located above the audience.

The orchestra pit can be open, if the hall is designed for opera, but is partially open in mixed-use facilities. The best pits have one or more hydraulic lifts to reconfigure the stage apron, and easy access from the practice and green rooms. The upstage edge of the pit opening must lie in front of the proscenium arch leaving four or five feet of space so that announcements and lectures can take place with the curtain drawn.

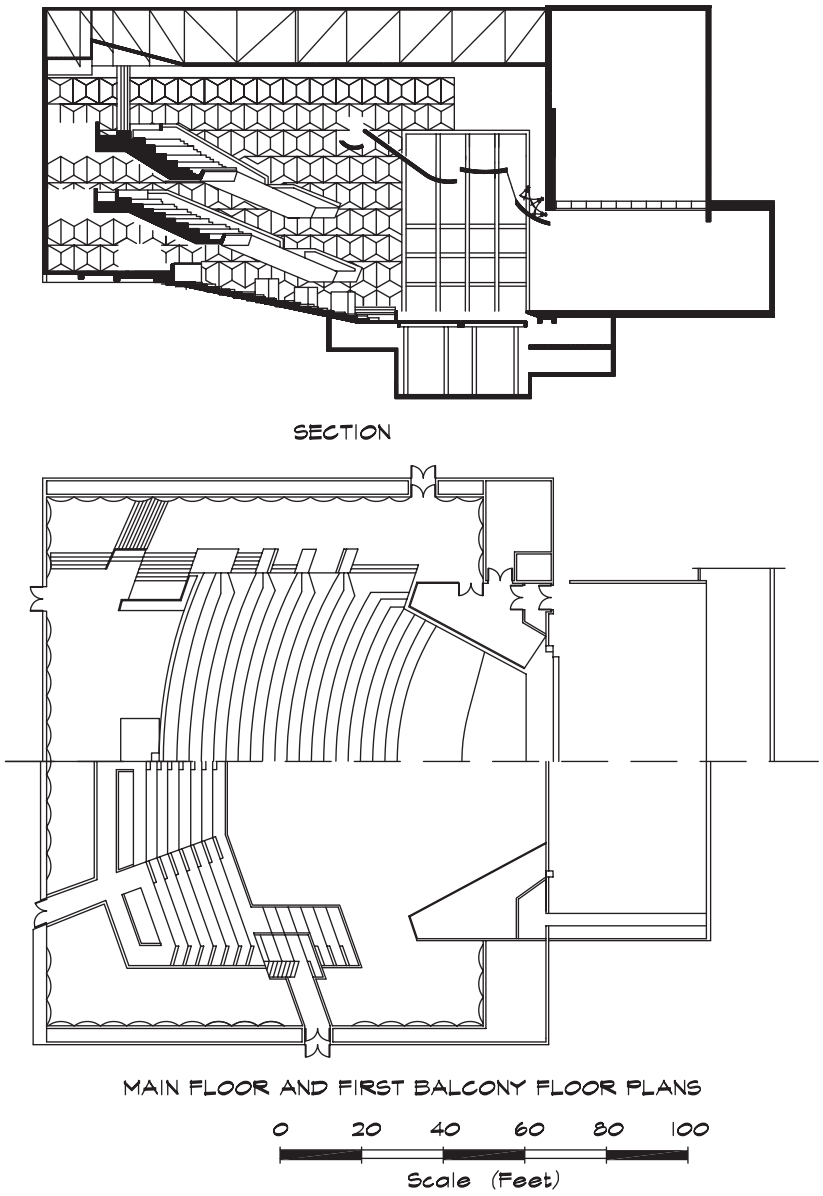
The auditorium must accommodate both theatrical and musical events with easy conversion between the two. The most common practice is to enclose the orchestra onstage within a shell either flown from the fly tower, stored in a dedicated room, or assembled from moveable towers. Another approach is to move the orchestra either partially or entirely in front of the proscenium and to close off the fly tower just behind the proscenium arch. This was the design concept used at Hamilton Place (Johnson, 2002) in Ontario, Canada. A 17,000 lb (7.7 m ton) cedar wall is lowered down to the stage floor, creating an orchestral platform in front of it, and neatly solving the acoustic problem of sealing off the fly tower. It also eliminates the need for back-stage storage of the shell and can serve as the fire curtain. The side and overhead reflectors can be designed for the same source locations irrespective of the performance configuration, and the overhead lighting requires little change. When the orchestra is on the thrust stage, the front seat wagons are lowered on the second lift and stored beneath the main floor seating area.

A sketch is given in Fig. 20.10. The flaring walls near the proscenium serve as acoustical reflectors for theater and as side walls of the orchestra shell. Simple hanging banners are used for variable absorption along the upper side walls. Both the upper and lower balconies float away from the side and rear walls, which increases the envelopment. The walls are highly diffuse and are displaced outward, a feature that increases the volume for music but still leaves space for the banners. It is a strong design, which is simple to convert from one form to another.

Traditional Churches

Church architecture has evolved out of a denomination-dependent historical context. Traditional Catholic churches look back to the cathedrals of the Middle Ages and often have high ceilings and the long reverberation times associated with Gregorian chant. Church design after Vatican II in 1969, when the saying of Mass in a language other than Latin was permitted, has moved toward an emphasis on speech and to breaking down the barriers between the laity and the clergy. Consequently, designs have tended toward lower reverberation times, smaller room volumes, padded pews, and more absorptive treatment.

FIGURE 20.10 Hamilton Place, Hamilton, Ontario, Canada (Acoustician: Russell Johnson; Architect: Trevor P. Garwood-Jones)



Episcopalian churches, reflecting their Anglican-Catholic heritage, are usually cruciform shaped, often with a double row of columns, reminiscent of the Roman basilicas. The pews are seldom padded but the room volumes tend to be lower than in Catholic churches. A raised pulpit, accessed by a steep wooden stairway, is a common feature. The choir is frequently located in a deep apse behind the altar. When monastic seating is used, choir voices do not project as well as when the choir is facing the congregation.

Other Protestant denominations place more emphasis on speech communications and have lower ceiling heights and padded pews. Their worship spaces, which evolved from

the northern European churches of the Reformation, can be rectangular-, cruciform-, or fan-shaped with a raised dais at the front. Electronic systems are becoming more central to the worship, particularly in the fundamentalist denominations. Many have multiple services on the same day, requiring different room configurations and musical setups. Since they support a variety of musical performances, stored presets for a given gain structure and a known group of microphones can save a good deal of time.

Today all religions need to preserve the dialogue between the congregation and the clergy so that speech intelligibility is critical. The overall shape of the room can assist with this. Churches in the round with the congregation surrounding a raised platform in the center of the room are difficult to design. The loudspeakers are ceiling mounted. Time delays are a challenge if there is more than one lectern position and the minister must turn his back on some of the congregation, or if the choir is located in a different part of the church. Worship spaces with both the dais and the choir at the front of the room sound much better.

Music is also important to worship, and the balance between speech and music is strongly influenced by the architecture. The choir should be located near the front, behind or on one side of the altar, so that the timing is consistent with the speech reinforcement system. Occasionally it is positioned at the rear on a dedicated balcony, in which case a separate dedicated sound system should be used. Accompanying instruments such as piano, organ, or instrumental groups should be located near the choir to help synchronize them with the singing. When there is an organ, its pipes should not be too far away from the singers or there will be a noticeable difference in the arrival time of the instrumental music and the singing. Churches having a pipe organ for instrumental concerts sometimes use a separate electronic synthesizer to support the choir.

Although there is no single best design for a worship space, several general design principles are helpful.

1. The chancel and pulpit should be elevated even if the loudspeaker is electronically reinforced.
2. Sections of the worship space that are acoustically separated should be designed to have acoustical properties similar to those in the main space. This may require the addition of absorptive treatment to high ceilings to balance out the reverberation times.
3. Solid walls or large ceiling beams should not separate seating areas, otherwise choral sound is not well distributed. Where large exposed structural members are necessary, they should be open trusses.
4. The choir should be located near the altar in an area that is open to the main part of the sanctuary, so that the singing can propagate freely throughout. The choir should be surrounded by reflective surfaces, to aid in distributing the sound to the remainder of the space. Choir enclosures should be free from flutter echoes, focusing, and long delayed reflections.
5. Pews or seats padded on the backs and sides are critical to minimize the difference between the empty and full condition.
6. An electronic reinforcement system, providing high intelligibility, even coverage, and the full range of frequencies without feedback, must be included. A mixer, located in the main space or in a nearby mix booth, should be used for singing.

7. Noise from both mechanical (interior) and environmental (exterior) sources must be controlled to a level consistent with the size of the space. Small churches are designed to an NC 30 (35 dBA) and large spaces to an NC 25 level (30 dBA).

One of the most difficult choices in church architecture is how reverberant to make the space when there is an emphasis on music. As a general rule, the reverberation time is designed to meet the musical needs of the space, and the intelligibility depends on the design of the sound system. In practice, however, the two must be designed together. If the sound system cannot overcome the reverberation, the signal-to-noise ratio must be increased either by adding absorption, by decreasing the loudspeaker-to-listener distance, or by increasing loudspeaker directivity. This does not mean that sound systems must be large and ugly to be intelligible. Intelligible sound systems can be designed without massive central clusters.

A good example of a reverberant church with excellent intelligibility is Saint Paul the Apostle Church in Chino Hills, CA, shown in Fig. 20.11. It has a high central roof supported by an intricate and beautiful truss structure. Fanning out from the raised roof area, at a lower elevation, is a semicircular roof, which drops from about 25 feet (7.6 m) to near 10 feet (3 m) at the outside walls. At the rear of the main sanctuary is a small mezzanine, where the mixer and audio equipment are located. The church seats about 1250 worshipers and was designed for both speech and music. The mid-frequency reverberation time is about 1.85 seconds even though nearly two-thirds of the high ceiling is absorbent.

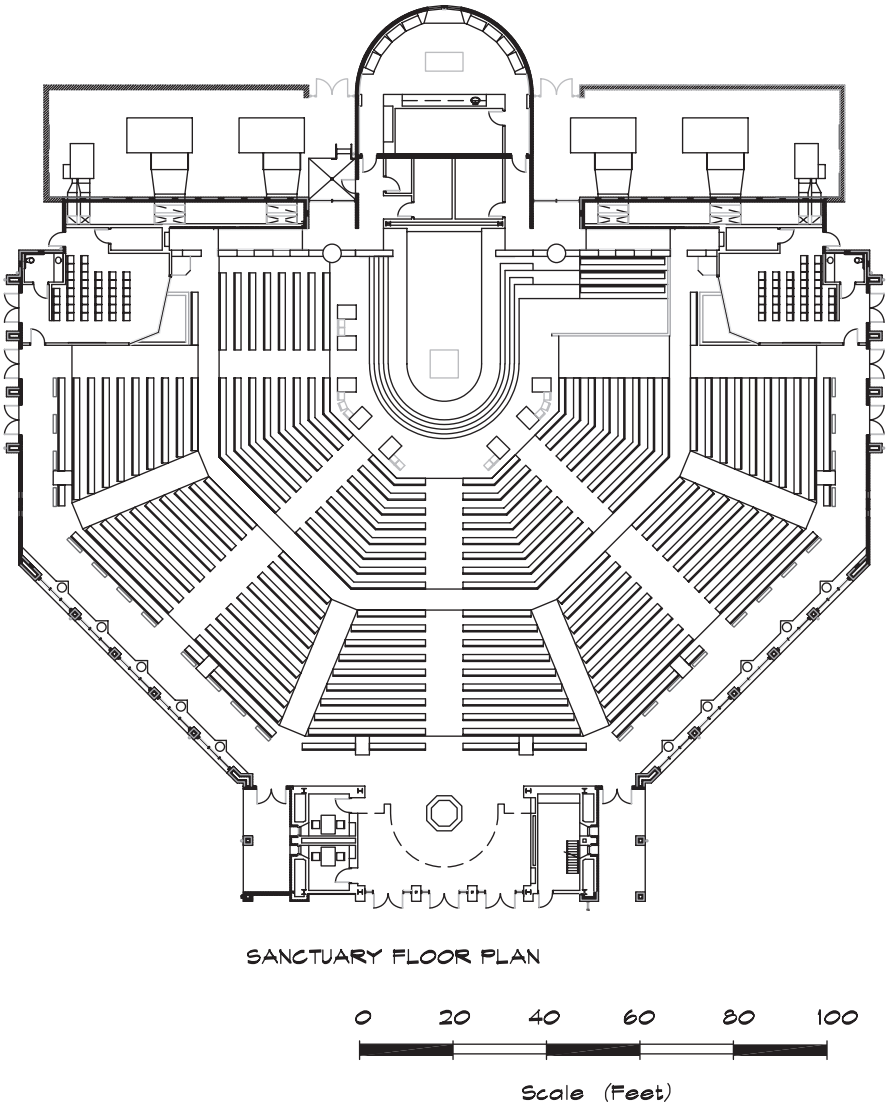
The loudspeakers are integrated into the architecture of the space. In the low-ceiling area, there are ceiling-mounted 12" (0.3 m) diameter coaxial loudspeakers. The central seating section is covered by a pair of cabinet loudspeakers, built into the framing of the truss work. Large subwoofers are supported on the catwalks near the top of the center ceiling area. To preserve the source image, 5" (13 cm) loudspeakers are incorporated into the face of the steps around the dais. These are delayed so that they lag a natural sound coming from the altar, and all other loudspeakers are delayed in rings centered on this point. The result is a sound that is clear and intelligible and, from any point in the church, appears to be coming from the center.

Large Churches

Large churches, between 2500 to 5000 seats, are increasingly common, particularly in Baptist and fundamentalist denominations. Dependent on electronic audio reinforcement systems, they often feature video screens with projected images from television cameras or recorded sources. They are designed as both theatrical and religious spaces with complex sound, lighting, and video systems. Many services include entertainment in the form of high energy music, dance, folk singers or other musical groups, and prerecorded video images. The natural acoustics of the room itself does not add appreciably to the audience experience but can detract from it. Since rooms are large, the reverberation times should be modest and the ceiling should be designed to give support to congregational singing.

Reverberation times in the 1.5 second range are a good compromise value. Hard ceilings are preferred to absorptive ones both for singing and because high levels of amplified bass will cause the T-bar ceilings to buzz. A drywall ceiling also helps redirect sound from the loudspeakers toward the rear seats. Padded pews or opera chairs are very important, since most of the absorption comes from the seating area. Since worship services can last two hours or more, congregational comfort is important.

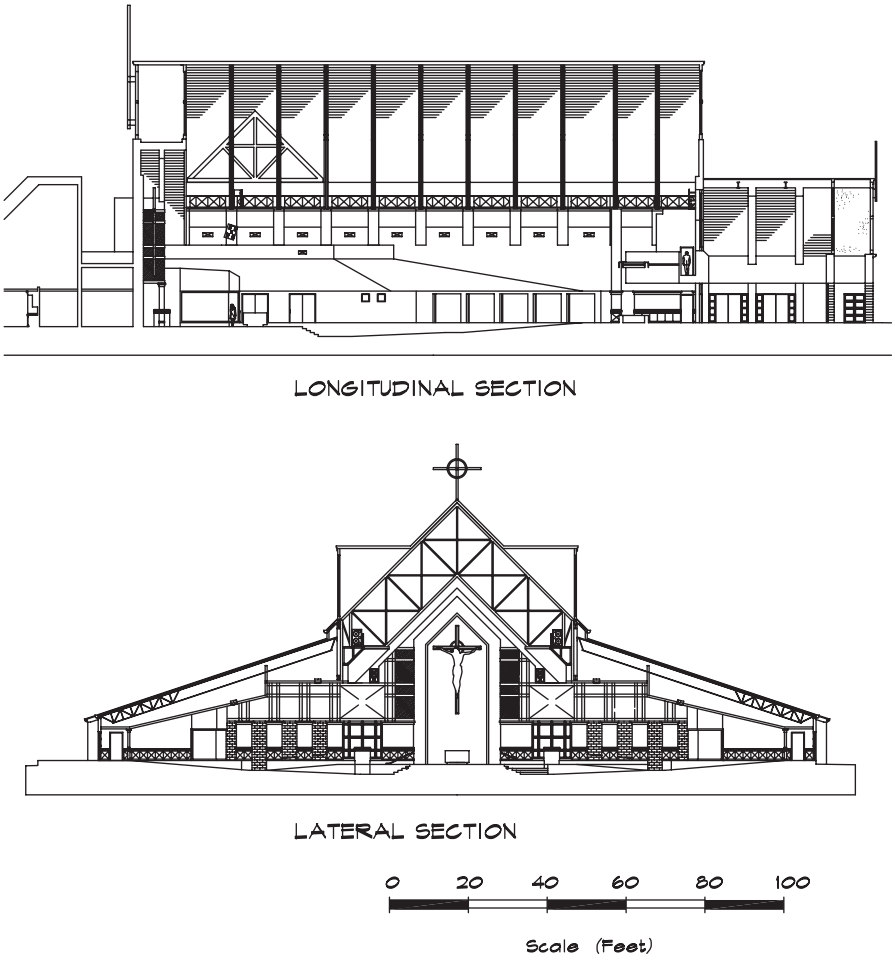
FIGURE 20.11A St. Paul the Apostle Church, Chino Hills, CA, USA (Acoustician: Marshall Long Acoustics; Architect: WLC)



Because of the large scale, the seating arrangement is invariably fan-shaped and includes a substantial balcony. An included angle of no more than 160° is recommended due to sight line considerations. Since the rear walls are concave, a reflection from this surface can result in serious focusing, they are treated with at least 2" (52 mm) cloth-wrapped fiberglass panels. A calculation of the curvature gain for both the talker and loudspeaker positions can be undertaken to determine if absorptive treatment is necessary.

Balconies should be sufficiently stiff to withstand rhythmic movement by the congregation. The highest natural frequency of rhythmic jumping is about 1.75 Hz (Allen, 1997). The fundamental frequency of a church balcony should be designed above the third mode of the excitation frequency or about 6 Hz to minimize resonant vibration.

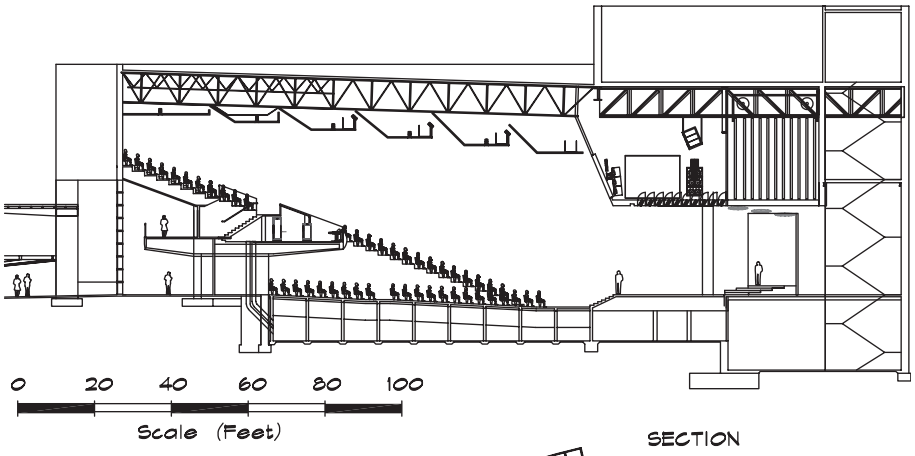
FIGURE 20.11B St. Paul the Apostle Church, Chino Hills, CA, USA

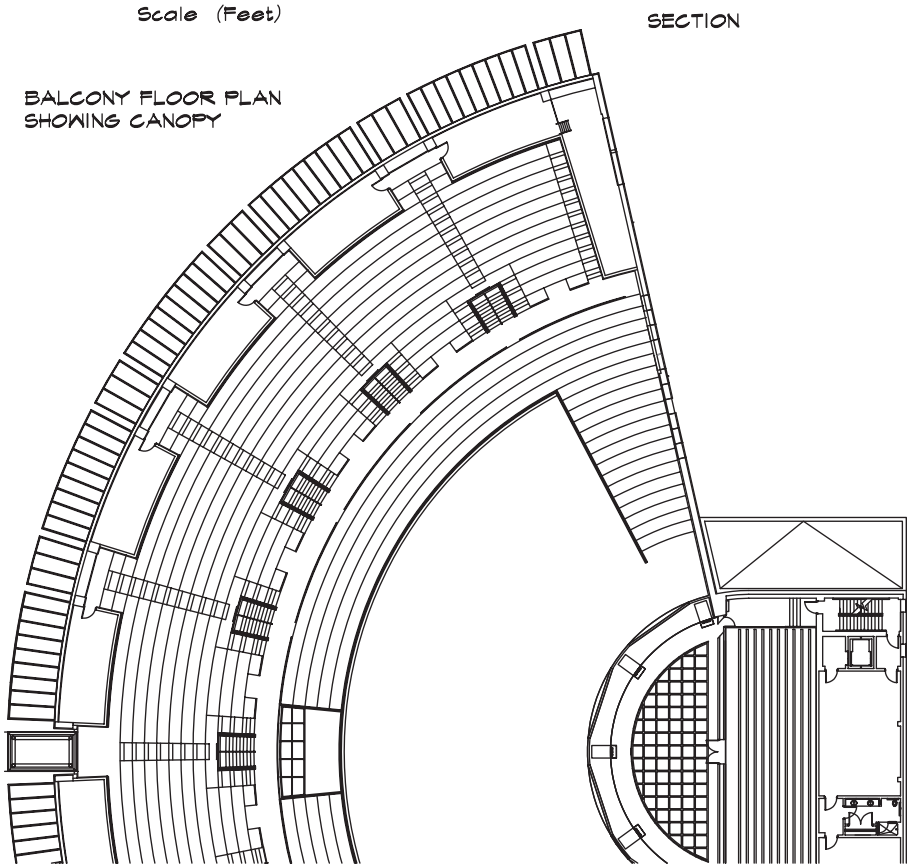


A canopy built above the platform housing the loudspeakers, video screens, and lighting instruments is the best choice. Since religious services can reach levels of 95 dBA or more, the sound system is large and requires a steel structure to provide support and access to the clusters. Amplifier racks should be located close to the loudspeakers to reduce the amount of heavy-gauge wire needed to be pulled. Ample power can be made available from nearby electrical rooms. Local step-down transformers must be vibrationally and acoustically isolated. In very large congregations (6,000 to 10,000 seats) multiple rings of loudspeakers are required to cover the whole seating area. Since clusters are more than 25 feet in the air, localizing loudspeakers around the platform are necessary to pull the image down. These can be built into the step riser using small horn or cone loudspeakers. Full-range loudspeakers are not necessary to give the illusion of direction, but the first-arriving sound must be within 10 to 12 dB of the cluster level.

A design sketch of the West Angeles Church, a large Pentecostal church in Los Angeles, is shown in Fig. 20.12. Seating 5000, its floor plan is fan-shaped with 60% of the seats in the balcony, including the long slanted areas along the side walls. The fan angle is 154°, but even at this angle there is some shielding of sight lines to choir area at the rear of the platform.

FIGURE 20.12 West Angeles Cathedral, Los Angeles, CA, USA (Acoustician: Marshall Long Acoustics; Architect: Langdon and Wilson)





The front of the platform that projects out from the front wall, is clearly visible, and can support a wide variety of performances. A canopy was designed to house the loudspeaker clusters, rear-projection video screens, and lighting fixtures, making for easy access and good coverage angles. Lighting catwalks ring the ceiling and help define its shape. The sharply defined corner of the ceiling in front of each catwalk keeps the theatrical lighting from shining on the video screens—an important concern in spaces with live TV coverage, where light levels must be kept high. The underside of the canopy is about 23 feet (7 m) high. This is high enough to get a sense of space above the platform but low enough to provide good acoustical support for the choir. The underside of the canopy is diffusive to avoid an echo return from overhead. Unfortunately the canopy was eliminated by the client in the final design.

Dedicated audio and lighting control positions are located in an open mix booth on the front edge of the balcony with an enclosed equipment booth behind. The patchbays and outboard equipment racks are in back of the mixer position. The wall above these low racks is glazed, allowing the technical director (TD) a clear view of the platform from within an acoustically isolated space. Since the congregation must be able to see over the booth, its ceiling is slanted down toward the platform. Video switching for the projection screens occurs just behind the TD position. Mixing for audio and video recording takes place in a separate studio. Audio and video recordings of the service are made available as part of the outreach ministry.

Large churches such as this one are totally dependent on electronic technology to distribute their message. Visual contact with the minister is enhanced through live television pickup of the service and projection on large video screens. Multiple video screens located in a canopy are ideal since they can be located between the loudspeaker clusters, giving good sight lines to most congregants. Screens located on either side of the choir are also a feasible alternative, however the seating fan angle has to be lower than about 130° for this to work well. Rear-projection screens are preferred to front-projection screens, since they can be angled down and they do not wash out under high background-light levels.

Synagogues

A synagogue is designed acoustically like other religious spaces emphasizing speech. Music is sung by a cantor, who is sometimes electronically reinforced but often not. There is usually a social hall adjacent to the sanctuary, separated by large doors or moveable partitions, which can be opened to combine the spaces. The social hall can serve as an extension of the main sanctuary during holidays so the audio systems must be linked. Social halls are designed primarily for speech intelligibility. Occasionally churches and synagogues will share a facility, which is used as a synagogue on Friday and Saturday and a church on Sunday, so the room must accommodate the needs of both congregations.

Orthodox synagogues have separate seating areas for men and women, divided by a freestanding wall or curtain. Orthodox Jews are proscribed from working and from doing a number of other things on the Sabbath, beginning Friday night at sundown and continuing until sundown Saturday. There is, for example, a prohibition against the lighting or extinguishing of fire, although an existing fire may be used for cooking. This teaching affects the use of electronic sound reinforcement. Each local rabbi makes his own decision on the exact interpretation and some congregations prohibit amplified sound, where there is a voltage zero crossing that is considered a lighting (or extinguishing) of fire. Others forbid the use of a dynamic microphone that initially creates the positive and negative voltage, but allow capacitive microphones because they modulate a DC voltage by varying the capacitance.

In the case of a prohibition of a zero crossing, a battery can create a voltage offset. This reduces the power output of the amplifier since the signal can only swing from zero to the positive rail, but the design complies with the teaching. David Klepper (1999) has written an interesting article on the subject.

20.3 SPECIALIZED DESIGN PROBLEMS

Once the seating capacity and floor plan layout have been established, multipurpose auditoria still have a common set of specialized design problems to be solved. These include the shape and structure of the side walls and ceiling surfaces, the configuration of the orchestra shell and how it is stored, the layout of the pit and access to it, the accommodation of variable-acoustic elements including room volume, absorption and diffusion, and the inclusion of audio/visual elements.

Wall and Ceiling Design

Side wall and ceilings should provide useful reflections to the orchestra level seating, while angling outward to accommodate fan-shaped seating layout. One approach is shown in Fig. 20.13, a design study for a large multipurpose auditorium. An interesting feature of this design is the compound-arc shape of the ceiling and wall sections. When designing for speech intelligibility, a slightly curved convex shape is ideal, whereas for high diffusion a low-radius curve is preferred. The walls and ceiling in this example are formed in a convex shape, which appears nearly flat for sound coming from the stage, but is diffuse for sound returning from the rear and sides. It provides good intelligibility, based on the first reflection, and gives good diffusion on subsequent reflections. Consequently it acts like an acoustical diode, whose properties depend on the direction of sound incidence. The side walls are angled to give good support for actors and singers onstage as well as reasonable envelopment from side reflections. When an auditorium is designed with a highly diffuse ceiling, the clarity of both speech and music can suffer in the balcony, particularly when the side walls are far apart. Both Davies Hall in San Francisco and Royce Hall at UCLA in Los Angeles share this problem.

Note that in the Fig. 20.13 design a second group of loudspeakers is used to cover the balcony seats. Delayed clusters of this type should employ drivers having similar characteristics to those used for the main cluster. They should be located in front of the seating section they are covering, so an appropriate delay time may be selected. See Chapt. 18 for a discussion of this topic.

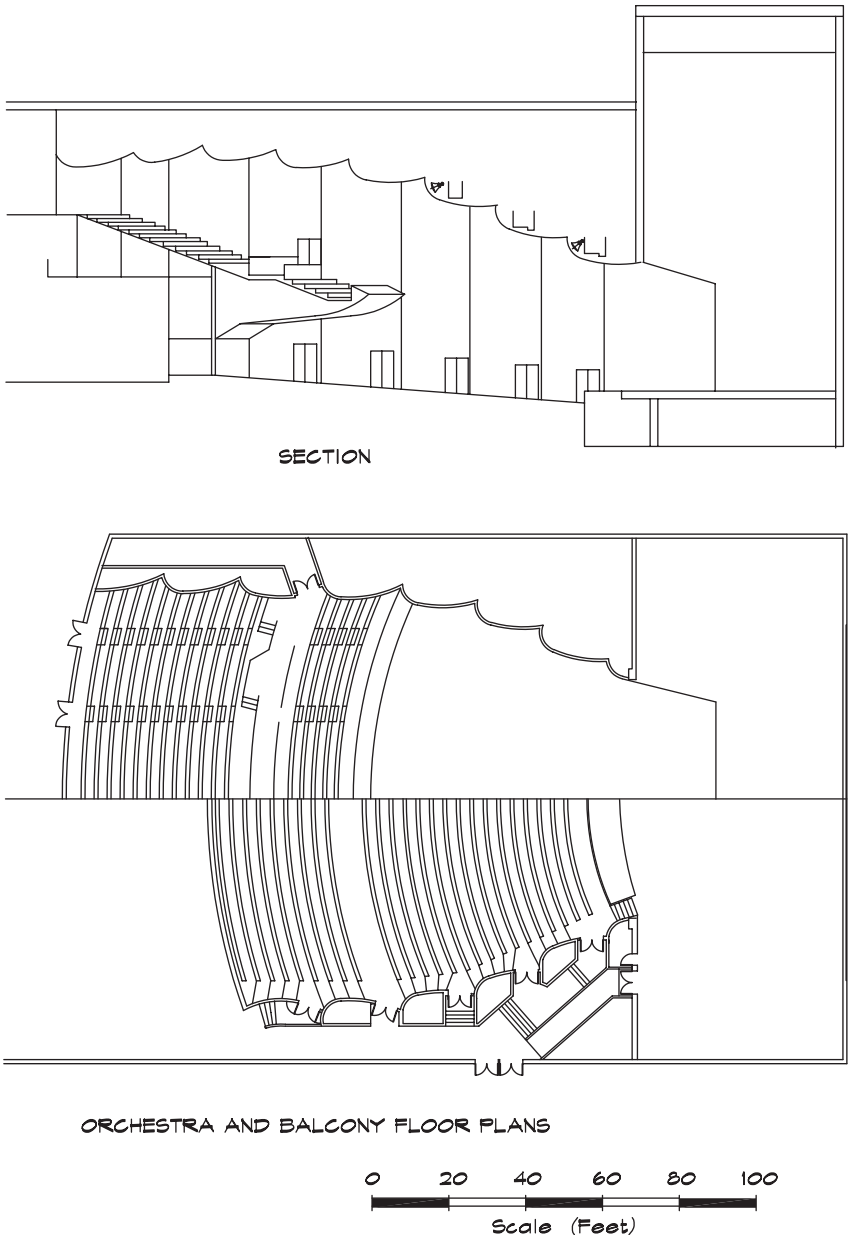
Shell Design

The design of shells and platforms is a subject meriting close attention. The general principles have been previously discussed; however, certain details are important to review. In multipurpose halls, shells are temporary structures, stored out of sight when not in use. In a concert hall the shell is a permanent construction and merges with the proscenium arch. In surround halls shells are formed from the partial-height walls around the orchestra platform.

A shell serves several functions:

1. It provides early reflections so that orchestra members can hear themselves and each other.
2. It separates the orchestra from the unfavorable acoustical environment in the fly tower and backstage.

FIGURE 20.13 Freedman Forum (Design Study) (Acoustician: Marshall Long Acoustics)



3. It helps reflect sound, primarily bass, to the audience.
4. It creates a local reverberant (diffuse) environment for the musicians.
5. It forms an attractive architectural backdrop for the orchestra, consistent with the design of the remainder of the auditorium.
6. If it is modular in construction, it can be stored efficiently when not in use.

Following these design goals we can attempt to find a Goldilocks solution—neither too deep, too high, nor too wide—just right. Depth is particularly critical. A shell should not be more than 30 to 40 feet deep from the front edge of the stage to the face of the rear wall. Boston Symphony Hall, a very good design, is about 35 ft (10.7 m) deep. Vienna is just over 30 ft (9.2 m). A shallow depth is particularly important for the reinforcement of the double basses, which should be arrayed along the rear wall of the enclosure. Not only does this arrangement benefit from a reflecting surface close by, it also forces the row of instruments to become a line source, when their notes are coherent, whose output is maximized in the direction of the audience. An overly deep enclosure allows the orchestra to move away from the rear surface, thereby giving up low-frequency gain. An overhang at the rear of the shell is also helpful. Both Boston and Vienna have excellent examples of this feature.

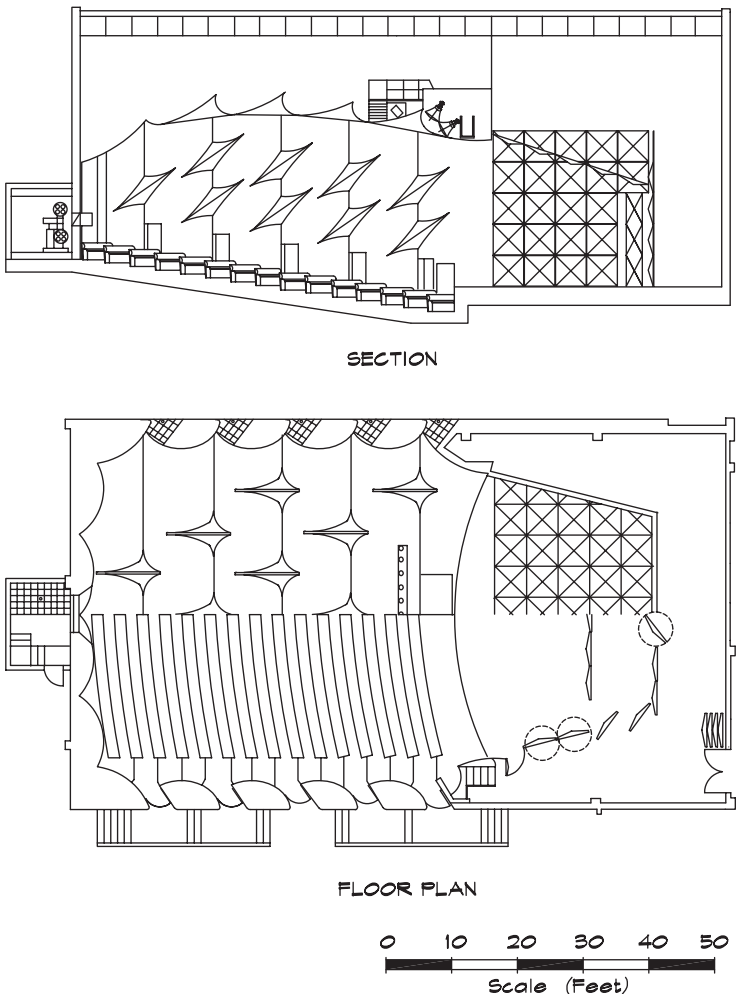
The orchestra enclosure should be narrower at the back and the shell walls should merge smoothly with the auditorium walls. The width of the platform and shell should average no more than 50 to 60 feet (15 to 18 m) at its midpoint, yielding the 2000 sq ft (186 sq m) or so of floor area recommended by Barron (1993) and Beranek (1996) for a full orchestra. When the orchestra is widely dispersed, the communication between the players is reduced. Gade (1989) found that beyond a distance of 26 ft (8 m) the delay in the direct sound can reduce the ease of ensemble playing. It is difficult to keep an orchestra this close together. Nearby reflectors either overhead or behind the orchestra can help offset distance effects.

Using Barron's (1993) recommendations for concert halls in Fig. 19.15, a platform volume of just under 70,000 cu ft (2000 cu m) yields a shell height of 35 to 45 ft (10.7 to 13.7 m). Boston's height is in this range at 43 ft (13 m). Amsterdam and Vienna have no shell. Carnegie Hall's shell is on the high side at 47 ft (14.3 m). In smaller auditoria this dimension can be reduced since proscenium heights are often no more than 20 ft (6 m). The front of the shell should match the proscenium when it is below 45 ft (13.7 m). It is also important that the ceiling of the enclosure not be too low. Low ceilings produce a sound on stage that is overly loud, and divide the hall into separate spaces with different reverberant characteristics. Barron (1993) recommends that the early decay time onstage be no less than 70% of that in the main hall.

The orientation of the shell ceiling is also important. A reasonable fraction of the energy impacting the ceiling should return to the orchestra. For an ST1 of −11 to −13 dB at a position in the center of the platform, energy must be returned from the ceiling as well as the rear and side walls. If the rear wall is 20 ft (6 m) away and perfectly reflective, it will contribute −15.7 dB (re 1 m), leaving an equal amount to be contributed from the ceiling and two side walls. If they are each 30 feet (9 m) away, at least 70% of the energy would have to be returned to the platform by a single reflection. Since this is impractical, a local reverberant field must be developed onstage to give the orchestra a sense of envelopment and balance. A combination of direct reflections and diffusive elements can be employed to keep a portion of the sound within the enclosure. Protruding elements can be used on the side walls of the shell to give additional reflecting surfaces.

An example, shown in Fig. 20.14, is a small hall in the John Arin Center at Sir Wilfrid Laurier University in Ontario, Canada, designed for recitals and lectures. The sketch is based on the author's original design. The size of the side-wall reflectors was reduced 60% by the architect in the final configuration. The shell is constructed in panels, supported by overhead rails. Each panel has a lower portion that can be swiveled about its vertical centerline to expose either its reflective or absorptive side. The angle of the side walls is set so that the panels can be shunted onto a second track to form a rear wall, positioned farther back, to accommodate a modestly sized orchestra. The panels have diffusing elements in the form of

FIGURE 20.14 Recital Hall, Wilfrid Laurier University, Ontario, Canada (Acousticians: Marshall Long Acoustics – Theater Vibron Ltd. – Mechanical Systems) (Architect: Lingwood Dubeje)

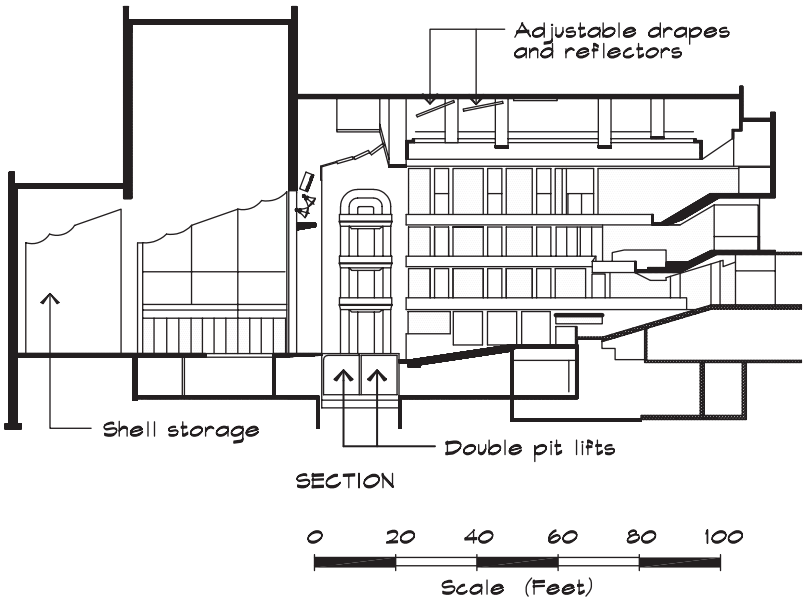


pyramids, angled so that one side is parallel to the stage floor. This provides high-frequency reflections back to the musicians.

In multipurpose auditoria the orchestra enclosure must be removable for theatrical performances. The shell ceiling can be designed to be flown from the rigging when there is a fly tower. If individually suspended clouds are used above the orchestra, bass energy will be lost into the absorptive fly tower and orchestral balance will suffer. Consequently a continuous roof is preferable to a series of open reflectors.

Modular side and rear panels can be rolled away on wheels or air casters to a storage area. There is a trade-off between ease of handling and bass absorption—lightweight materials being easier to move but more absorptive of bass. Barron (1993) recommends a material with a surface mass density of no less than 4 lbs/sq ft (20 kg/sq m), corresponding to 1" (25 mm)

FIGURE 20.15 Ordway Hall, Saint Paul, MN, USA (Acoustician: R. Lawrence Kirkegaard) (Architect: Benjamin Thompson)



gypsum board or 1 3/8" (35 mm) plywood. These widths are feasible in fixed enclosures but difficult in a modular shell, which is constrained by weight to a maximum thickness of 5/8" (16 mm). Where thin wood is used, it can be shaped into a convex curve and backed with closely spaced structural stiffeners to reduce diaphragmatic absorption. Curves are constructed by laminating several thin sheets together. Heavy shells can be accommodated using air castors or rails. Figure 20.15 shows the orchestra shell of the Ordway Theater in Minneapolis, stored in its own room adjacent to the stage.

Platform Risers

Orchestra platforms are nearly always constructed of wood over an airspace, a configuration highly favored by musicians for the warmth of its sound. Askenfeld (1986) found that wood floors provide both beneficial support as a sounding board in the low registers, as well as low-frequency absorption for airborne sound. The flooring should be sufficiently light that it responds to vibrations generated by the cellos and double basses, and provides tactile feedback to the musicians, but not so light that it absorbs significant energy. Barron (1993) suggests 7/8" (22 mm) thick wood on sleepers spaced 24" (600 mm) apart to allow for some movement, while still maintaining structural support for heavy pianos or dance troupes. The depth of the airspace below the floor varies, but a cavity of more than a few inches is not recommended.

Portable stage risers are quite helpful for projection of sound, both as a local radiator and as a way of elevating the orchestra so that the players do not shield one another. Risers also form reflecting surfaces to contribute both diffusion and early reflected energy to the audience. Since they are modular and have a relatively short structural span, they can be

constructed of lighter wood than the main floor, but should not be less than 5/8" (16 mm) thick.

A riser system should be flexible enough to accommodate a choir located behind the orchestra. When the choir is not present it should be possible to move the orchestra back against the rear wall of the shell. A permanent seating area behind and above the orchestra is useful in concert halls and there are excellent examples in Vienna and Amsterdam. These seats can be used for the audience when there is no choir, although at a sacrifice in sound quality and balance for patrons seated there.

Pit Design

Pits in auditoria and multipurpose halls are designed in much the same way as in opera houses, with some exceptions. Stage overhangs in these facilities tend to be greater than in opera houses since the singers are not as powerful and orchestras need to be muted. Multipurpose auditoria rely on electronic reinforcement for individual singers and even for the orchestra. The audience seating area, which must be sacrificed to obtain an open pit, is so valuable that it is rarely worth giving up. It is critical to be able to raise the pit floor to the level of the stage either with a hydraulic lift or by installing a pit cover to provide for a thrust stage. The open portion of the pit can serve as an extra seating area when small musical groups are performing.

In musical theater, such as that found on Broadway, the pit band almost always is amplified, and sometimes a great effort must be expended by the sound designers to overcome the "natural" acoustics of the pit. Individual and sometimes multiple microphones are used for each section. A drum set, for example, might have five microphones to itself. Each of the feeds from an instrument section should be clean—that is, it should only include signal from the source being miked. In these instances absorptive panels are hung on the walls and gobos are used to separate sections of the orchestra so that they are isolated from one another. The best design approach in these cases is to allow extra space for absorptive or diffusive panels and provide plenty of microphone jacks—at least 24 and often as many as 48 are necessary.

Since access to the pit must be provided both for freight and the handicapped, an elevator is required. Figure 20.7 shows a good example of how this can be accomplished in a small high school auditorium. Even though an elevator is more expensive than a lift its flexibility and convenience pays for itself over the long term. Flexibility in the pit acoustics is also helpful. Removable diffusive and absorptive panels can be used to vary the local reverberation in different areas of the pit according to the seating arrangement of the orchestra.

Diffusion

Adequate diffusion is critical for obtaining an even distribution of reverberant sound in a listening space. Diffusion helps absorptive materials be more effective, by scattering sound so that it is more likely to encounter these surfaces. Where there is inadequate diffusion, acoustical defects such as flutter echo or coplanar reflection patterns can develop. Lack of diffusion can produce irregularities in the slope of the reverberant decay in a room. In concert halls and other critical listening environments the listener gains a sense of envelopment, of being surrounded by the sound, from reflections that come from the side. Lateral reflections are enhanced by materials or objects that increase diffusion.

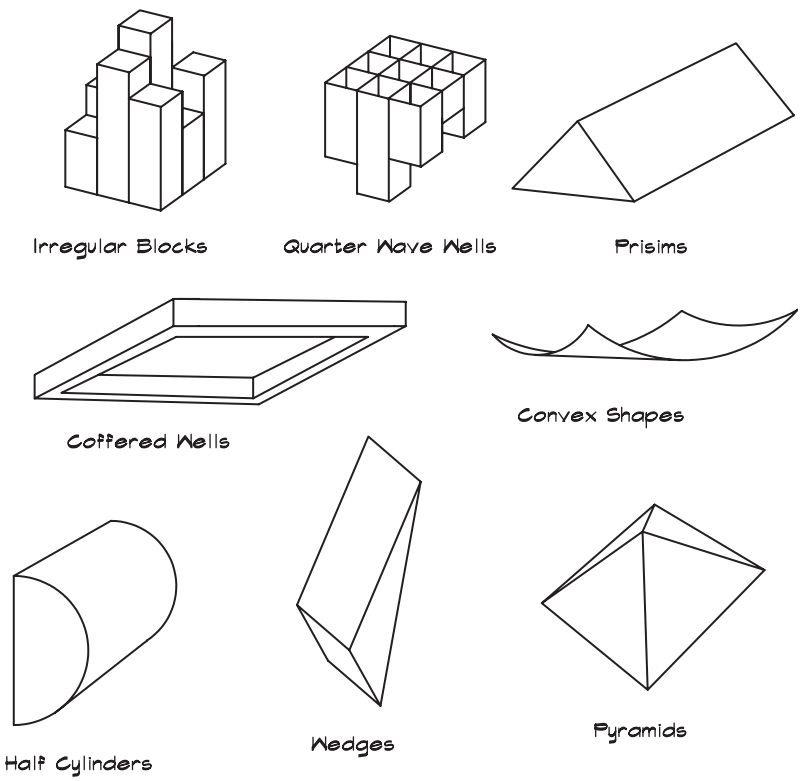
Diffusion reduces the glare caused by reflections from flat or slightly curved panels. Too much high-frequency energy reflected from walls can produce an annoying brightness or,

in the case of parallel panels, a flutter echo. This condition can be treated by roughening a surface with irregularities on the order of 2.5 cm (1 in) deep or by adding a thin, 5 mm (3/16”), layer of absorptive felt or fabric curtain. Flutter also can be treated by angling a wall about 1:12 with respect to the opposite face.

The quantification of diffusion is a subject of some theoretical interest. Several methods available to model its effects in ray tracing programs are discussed in Chapt. 22. Two separate technical problems exist—first, how to characterize the interaction of a sound wave with a surface, and second, how to incorporate this characterization into the calculation of room acoustics. In the simplest case, illustrated by a numerical measure such as SDI, a single number rating is assigned, based on visual inspection of the room surfaces (Haan and Fricke, 1997). This, however, does not provide a clear method for the calculation of the level of a reflected sound wave, let alone the more complicated metrics such as interaural cross-correlation coefficient or lateral energy fraction.

Figure 20.16 shows several examples of diffusing surfaces. Ornamentation such as columns, statuary, caryatids, railings, balustrades, steps, organ pipes, and chandeliers can all help. Prefabricated diffusing materials are becoming available in wood, FRP, and masonry units, which have the advantage of providing a predictable behavior. Diffusion should be used in areas where specular reflections are not helpful. Although this sounds obvious, it helps identify surfaces where the application of diffusion may be relatively benign. In an auditorium, likely areas include ceilings above reflective panels, the upper portion of a side

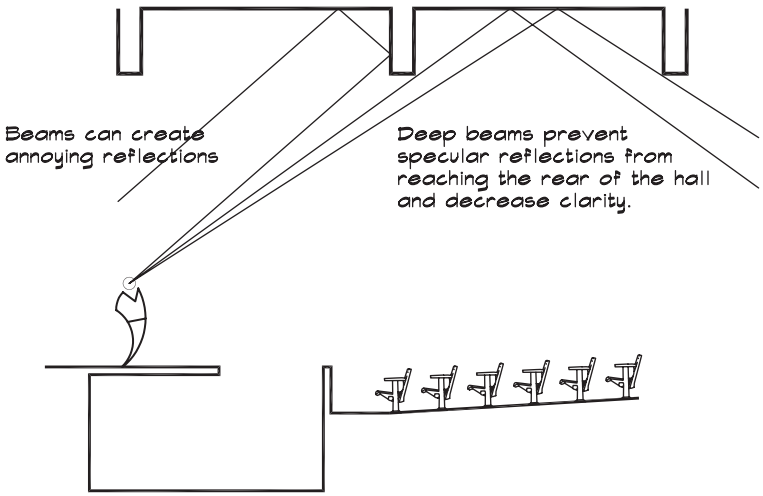
FIGURE 20.16 Examples of Diffusing Surfaces



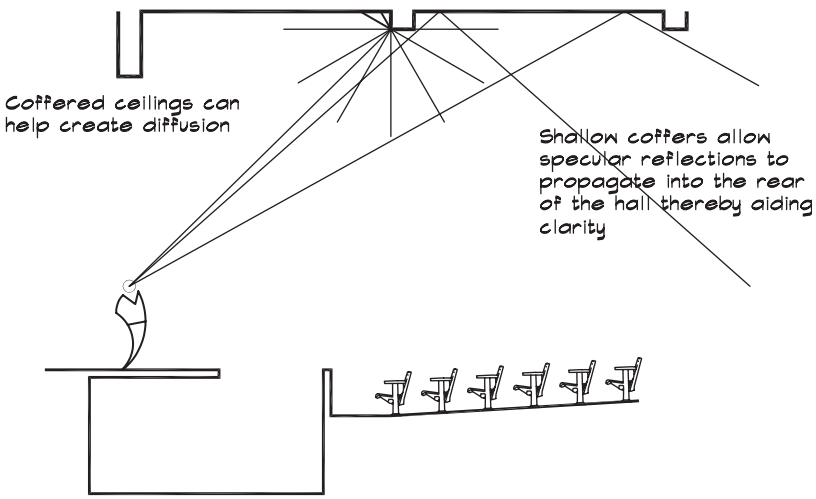
or rear wall, or on either side of the stage. Diffusion can also be helpful in orchestra pits, shell enclosures, practice rooms, and other small spaces.

Diffusing elements can include columns, statuary, and exposed structural members such as those in St. Paul the Apostle Church (Fig. 20.11B). Structural elements such as trusses or catwalks, as well as chandeliers and lighting instruments, projecting down from the ceiling, allow sound to pass beyond them and can yield excellent results. Figure 20.17 illustrates the design tradeoffs with lateral beams. If a deep beam or archway supports the ceiling, it can divide the room into separate acoustic spaces, since sound will be reflected back toward the platform and not propagate out. The combination of beam depth and ceiling height must be such that a significant fraction of the reflected energy continues to flow toward the audience.

FIGURE 20.17 Design of Coffered Ceilings



AN EXAMPLE OF POOR CEILING DESIGN



AN EXAMPLE OF GOOD CEILING DESIGN

Shallow beams, on the order of one quarter wavelength deep, still help diffuse sound while allowing specular reflections from the flat surfaces to pass by. With high ceilings, deeper beams may be used since the approach angles are more acute.

The ideal tradeoff between diffuse and specular reflection is one of the more interesting design problems. Too little diffusion can result in inadequate envelopment, glare, and other undesirable characteristics. Excessive diffusion can also a pose a problem. A highly diffuse ceiling in a concert hall may reduce the clarity in the balcony if the specular reflections from the side walls are insufficient to compensate. Too much diffusion is also expensive and may not work well with the desired architecture. Surfaces providing both specular and diffuse reflections are ideal. Both Boston and Vienna are excellent examples of this characteristic. Their coffered ceilings are nearly flat, between the dividing beams, and are relatively shallow. The flat portions allow specular reflections to continue on toward the rear seats, while a portion of the sound is scattered off the beams.

Variable Absorption

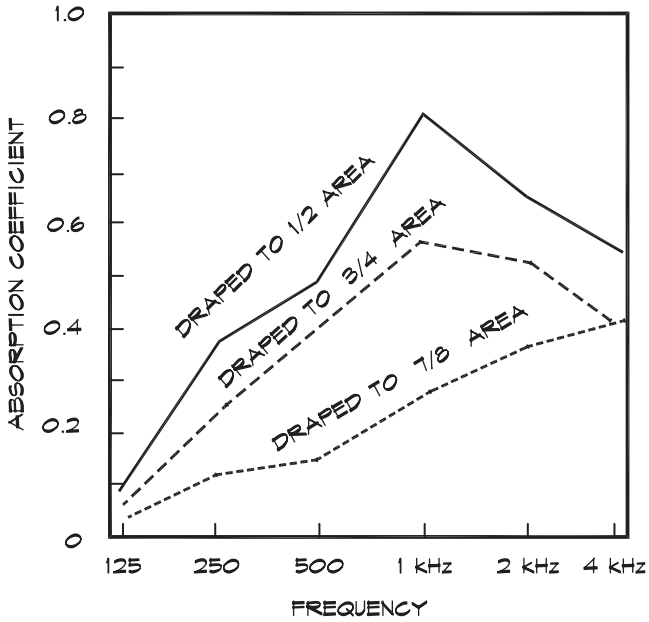
In multipurpose auditoria and even in dedicated concert halls, there are occasions when the rooms will be used for speech or when the addition of absorption to reduce reverberation is appropriate. There are many complicated ways of doing this including rotating triangular absorbers, barn door panels, and sliding mechanisms. The simplest methods feature fabric curtains or banners hung from tracks that can be trundled or lowered into place either mechanically or by hand. Large areas of curtain are required to make substantial changes in the reverberation time. For example if a 2-second reverberation time is to be changed into 1.5 seconds using absorptive material, an area equal to about one-third the audience area must be added. This could require 4000 sq ft (370 sq m) of highly absorptive material in a 2000 seat hall, a substantial amount, given the fact that it must be stored away out of sight when not in use. Absorption is most often added using heavy (doubled 20 oz/sq yard) fabric banners that are lowered through slots in the ceiling, directly into the auditorium, or behind transondent perforated metal screens. Where there is space behind a balcony, or a side seating area, banners can cover a substantial area. Curtains should be positioned so that there are several inches of airspace behind them to maintain their effectiveness at low frequencies.

Beranek (1949) published data on absorptive curtains, given in Table 7.1. Gathered curtains are difficult to store unless they are supported on horizontal tracks or large rollers. The folds help improve the low-frequency absorption mainly by increasing the thickness. The amount of gather is expressed as a percentage of the total length of the curtain divided by its finished length. Curtain absorption is a strong function of the percentage of gather as shown in Fig. 20.18.

When absorptive banners cover portions of the lower side walls, they can absorb early reflections and detract from the clarity of speech. Therefore they should not be installed immediately adjacent to seating areas where they can reduce useful reflections. If absorptive treatment is placed on side walls near listeners, local overhead reflectors extending out from the walls can be helpful in adding back early reflections to side and rear balconies, since they will not be covered by a banner.

The more difficult it is to vary the absorption, the less likely it will be done. In small auditoria curtains can be hung from the ceiling adjacent to catwalks from which they can be slid into position by hand. The range of reverberation times available through this type of adjustment is limited, but a variation on the order of 0.3 seconds in small halls is possible and

FIGURE 20.18 The Effect of Draping on the Sound Absorption of Curtains (Mankovsky, 1971)



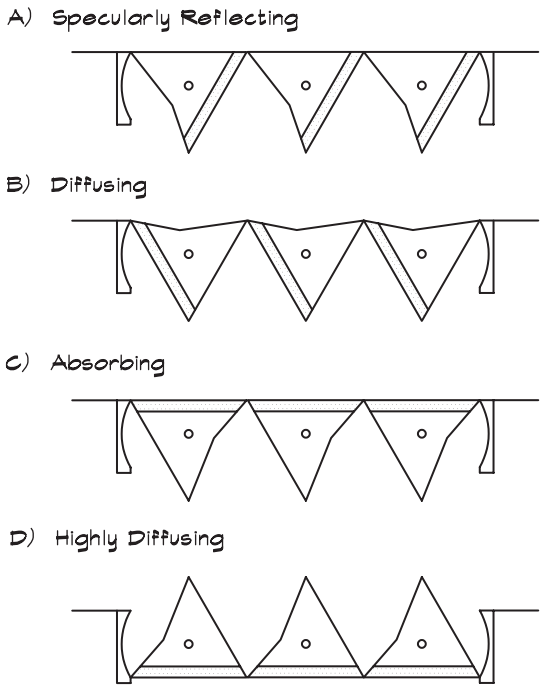
can be helpful. In large auditoria, curtains are motorized and preprogrammed presets can be used to automatically adjust the amount of exposed surface. One of the most flexible acoustic spaces is at the Institut de Recherche et Coordination Acoustique/Musique (IRCAM) in Paris. The room is used for recording, acoustic research, and as a small auditorium seating about 400. Both the volume and the absorption can be varied—the volume by raising or lowering the three-part ceiling and the absorption by rotating three-sided prisms to expose surfaces in several orientations, drawn in Fig. 20.19. It is highly flexible but not a good model for a general performance space due to its complexity.

Variable Volume

Since the reverberation time can be varied by changing either the absorption or the volume, it seems inevitable that variable volume halls would appear. Early examples include a number on which CRS Sirrine was the architect and George Izenour the theater consultant, including Jesse Jones Hall in Houston, Texas (BBN acousticians) and Edwin Thomas in Akron, Ohio (V. O. Knudsen acoustician). The method employed was to suspend ceiling panels from cables, which in the case of Jesse Jones were individual reflectors; in Edwin Thomas, the design used the catenary curve formed by the cables. The rear portion of a suspended ceiling could be lowered to the front of the balcony to reduce the volume and seating area of the space. The hardware required to carry out this transformation was substantial, and these designs have not been widely emulated. Detailed illustrations have been shown by Izenour (1977).

Another interesting design concept, first used in the Derngate Centre, Northampton, England and later in the Cerritos Performing Arts Center in California, employs mobile seating towers, supported on air castors capable of being moved about. A sketch of the

FIGURE 20.19 IRCAM Rotating Wall Prisms



seating arrangements in the Cerritos Center is given in Fig. 20.20. The rear walls of the towers are constructed of perforated metal so that absorptive banners can be lowered behind them. The envelopment in these spaces is not high, since perforated metal does not give much reflective support (Barron, 1998). The design is very flexible, however, and a clever way of allowing the rearrangement of the space into multiple configurations.

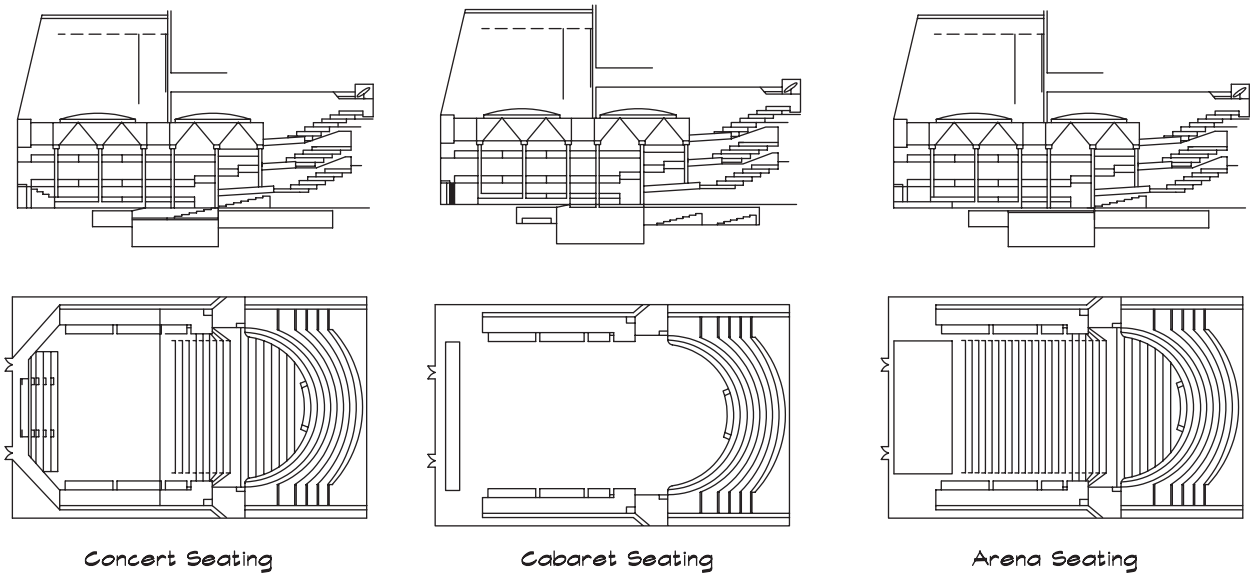
A mechanically adjustable ceiling is the most straightforward way to change room volume. A vertically adjustable ceiling has been recently installed in the Sala Sao Paulo, Sao Paulo Brazil. This space was built by converting a large outdoor courtyard, formerly a waiting area for an administration building adjacent to the railroad station, into a concert hall. The open-air plaza was bordered by two rows of columns that formed the sides of the new hall. A coffered ceiling, consisting of 30 separate sections hung from cables, can be raised and lowered to adjust the volume according to the type of music being played. A sketch of the hall is shown in Fig. 20.21.

An increase in room volume lengthens the reverberation time of a space uniformly over the entire frequency range. It is unlikely to introduce coloration in the form of normal modes associated with resonant cavities. The increased reverberation is available to the listeners immediately as running reverberance, which means that it could negatively affect clarity, but also that it provides a greater sense of diffuse envelopment.

Coupled Chambers

Another method of raising the reverberation time in a hall is by increasing the room volume using coupled chambers, located adjacent to the main space, and accessed through openable doors or moveable wall sections. A reverberation characteristic results, which is somewhat

FIGURE 20.20A Cerritos Center for the Performing Arts-Seating Configurations (Acoustician: Lawrence Kirkegaard and Associates) (Architect: Barton Meyers and Associates)



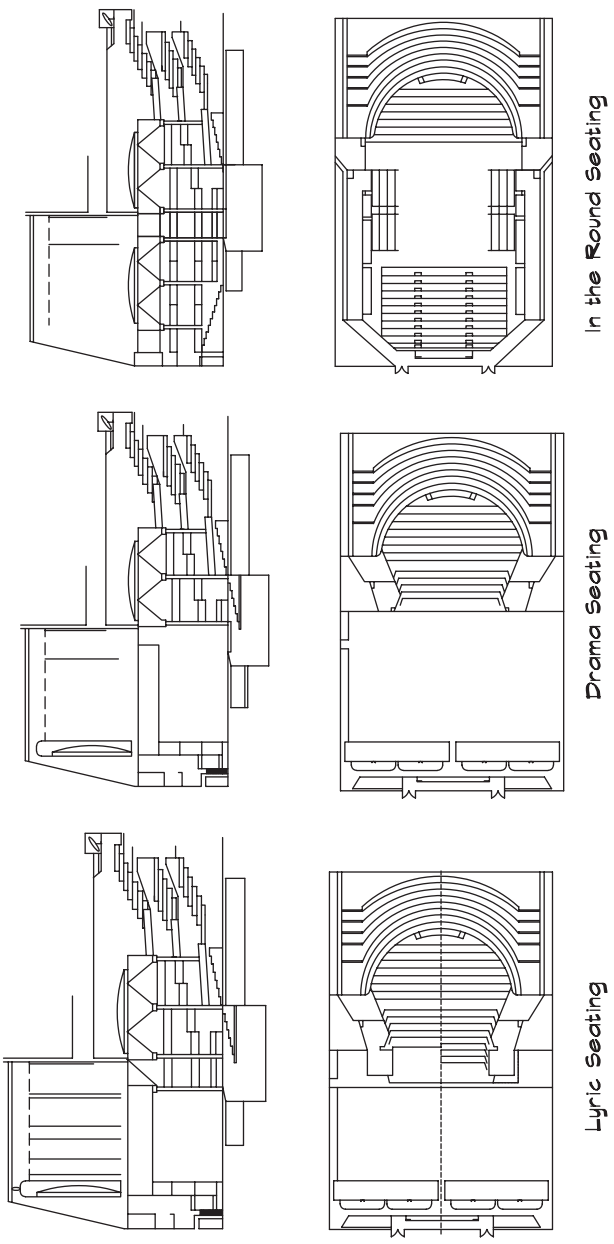


FIGURE 20.20B Cerritos Center for the Performing Arts-Seating Configurations

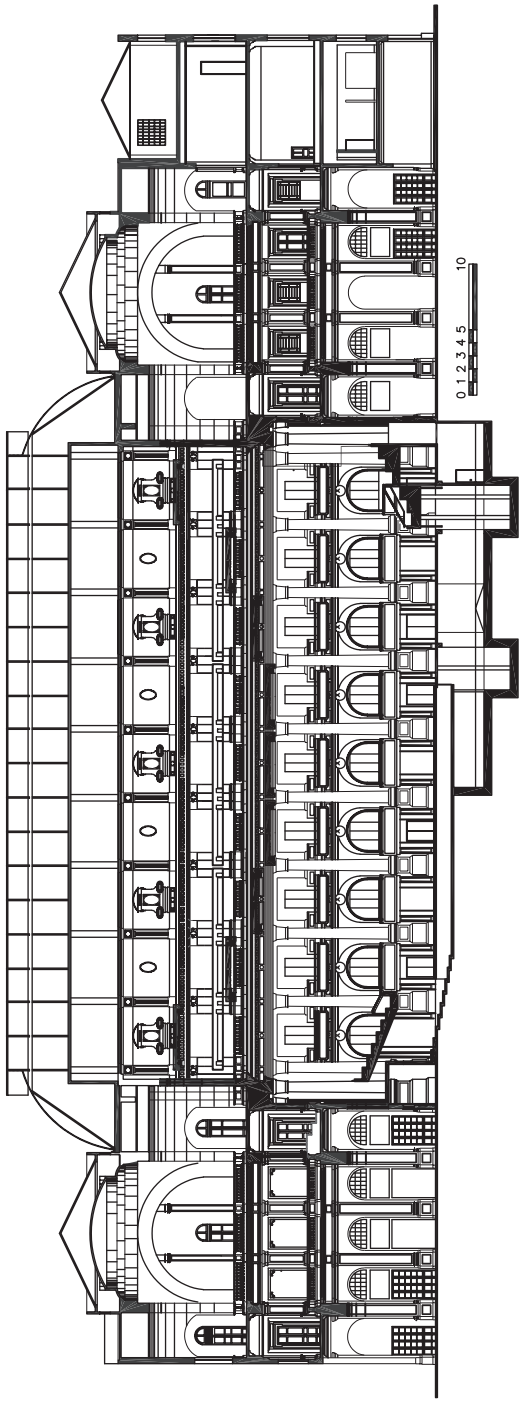
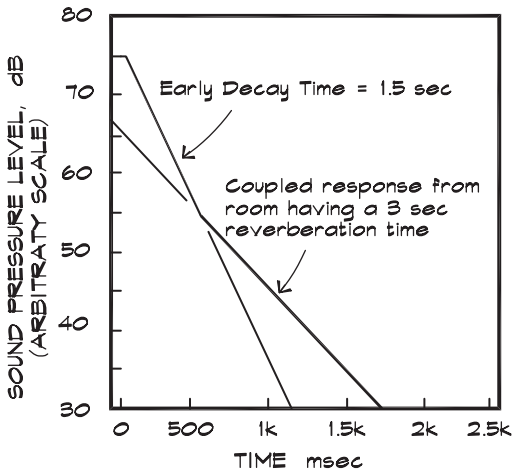


FIGURE 20.21 Sala Sao Paolo, Sao Paolo, Brazil – Longitudinal Section (Acoustician: ARTEC, Drawing reproduced with permission)

FIGURE 20.22 Dual Slope Reverberant Decay



different than that obtained by simply increasing the volume in the main hall by raising the ceiling. Coupled volumes remove some of the energy from the primary hall and return it at a later time and at a lower level yielding a segmented energy-decay slope such as that shown in Fig. 20.22. This is perceived as a reverberant tail after a stop chord rather than as running reverberance (Beranek, 1996) audible during the music.

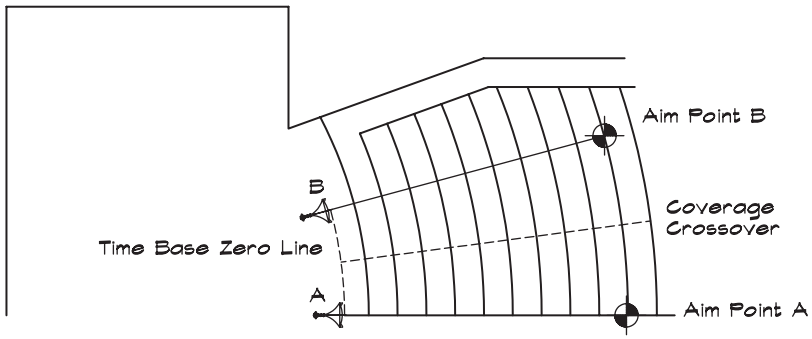
Separate hard-surfaced rooms, serving as reverberation chambers, can have characteristic resonant frequencies. A resonant device preferentially absorbs energy from the driving mechanism and returns it at a later time. Where there are characteristic modes of vibration in a coupled space these will persist longer than other frequencies. One hall, Sala Nezahualcoyotl built in Mexico City in 1970, has a concrete pit below the orchestra. Resonant chambers located directly under the orchestra are probably not a good idea since they absorb acoustic energy and in time become general storage areas. It is arguable that the energy would be more useful if it were available to the audience with the original sound rather than being banked for later distribution to the orchestra. When coupled chambers are located in the upper reaches of the auditorium the sound must first pass through the hall before entering them and is thus available to the audience. The Mexico City experiment has not been repeated.

Recently a number of halls designed by Artec in New York have included large reverberant chambers that are accessed through heavy openable doors or moveable sections of wall. The McDermott Hall in Dallas shown in Fig. 19.20 is a good example. The design maintains clarity from early reflections off the canopy, while giving reverberance to the space. The reverberation is not as strong as that associated with the classic rectangular halls such as Boston and Vienna but adds some liveness. Coupled chambers have primarily been included in orchestral halls rather than multipurpose auditoria.

Sound System Integration

When the sound system for a multipurpose hall is designed the loudspeaker layout is first modeled along the lines set forth in Chapt. 18. The most common methodology is to use a computer program to predict the loudspeaker coverage. Calculations must be undertaken in the 500, 1k, and 2kHz octave bands at a minimum to assure flat frequency response, feedback

FIGURE 20.23 Proscenium Loudspeaker Alignment



resistance, and adequate signal-to-noise ratio in the speech bands. Although equalization can be used to correct some of these problems, if the off-axis behavior of the loudspeakers is frequency dependent equalization may not produce an equal response at all locations.

In most multipurpose auditoria and churches, particularly those adopting fan-shaped seating layouts, the loudspeaker arrays required to cover the seating area include multiple horn and midrange cabinets, which are arranged across the front of the room. Since they are distributed in space there will be overlapping areas of coverage where care must be exercised in blending the sound fields from several sources. At receiver locations covered by more than one loudspeaker the arrival time from each source should be approximately the same so the transition between loudspeakers is smooth. The physical alignment of the loudspeakers is important, particularly at the overlap point. Figure 20.23 illustrates the concept. This loudspeaker arrangement is preferable to a central cluster such as that shown in Fig. 18.3. It yields better coverage and preserves the perceived source direction.

Since the directivity of high- and mid-range devices is not always the same, the number of the sources necessary to cover the audience is not always equal. For example, three horns may be required in an area that two mid-range cabinets can cover. In this case it is logical to place the mid-range cabinets between the horns. The faces of the loudspeakers should be arranged along an arc and time delays used to match the arrival time of each device. This matches the signals both in level and in time in the areas where the coverage of more than one loudspeaker occurs.

Cabinet loudspeakers are designed to match component directivity over the speech frequency range. Small horns maintain reasonable cabinet dimensions, but the directivity at the mid frequencies afforded by large horns cannot be duplicated in small boxes. Consequently cabinet loudspeakers are directional only above 2 kHz, particularly in the vertical plane. Loudspeaker manufacturers suggest that cabinets be arrayed to increase (or decrease) directivity, however an array such as that shown in Fig. 18.3 increases lateral high-frequency coverage but can decrease low-frequency coverage due to interference between the low-frequency sound fields in the horizontal plane. It may be preferable to match the directivities by using separate large-mouth horns and dual midrange boxes as shown in Fig. 18.2.

Where possible loudspeakers should be placed in a drywall enclosure, which protects the stage and provides some barrier shielding against feedback. Architectural design elements can be built around the enclosure. When a thrust stage is used, the extra shielding can be very helpful. The upper drawing in Fig. 20.4 gives a good example.

In large auditoria, a mixer should be located in the audience area, ideally on the centerline. The front of the balcony is probably the best location for a mix booth since the acoustic signal path is unobstructed. The FOH (front of house) mixer is seated at a mixing console with a clear view of the stage and with low-profile equipment racks located directly behind him. These racks house the sound sources necessary for the effects along with the wireless microphone receivers and jack bays necessary for the real-time control of the system. Amplifier racks and fixed audio processing and distribution equipment are remotely located since they do not require immediate adjustment.

In smaller facilities such as high school auditoria, where security is a concern, audio equipment is located in a lockable room or tech booth with an openable window separating the booth from the seating area. The window opening should be large enough, at least 4 ft (1.2 m) high by 6 ft (1.8 m) wide, to minimize low-frequency attenuation and give the operator a sense of the room. Eighteen-inch telescoping window panels allow a 6 ft opening in a 9 ft (2.7 m) total width. Mixing while listening to the sound using loudspeakers within a closed booth is not effective, although dummy heads with ear microphones have been used.

Electronic Augmentation

In spaces where there is insufficient natural reverberation, it is possible to introduce reverberation electronically. The idea has a long history dating back to publications by Vermuelen (1958) and Olson (1959), who each proposed room-enhancement systems using microphones, delay lines, and loudspeakers. A number of commercial systems have subsequently been developed, which fall into two groups: direct field systems and, reverberant field systems. Both types sample the natural sound in the hall and use electronic processors to increase the reverberation time and occasionally the early reflections by reintroducing the signal through loudspeakers. The first group uses microphones close to the source, within one or two times the critical distance, and feeds the signal though processors and out to loudspeakers close to the stage as well as others located in the reverberant field. The second group uses microphones and loudspeakers located in the reverberant field. These cannot generate early reflected sounds.

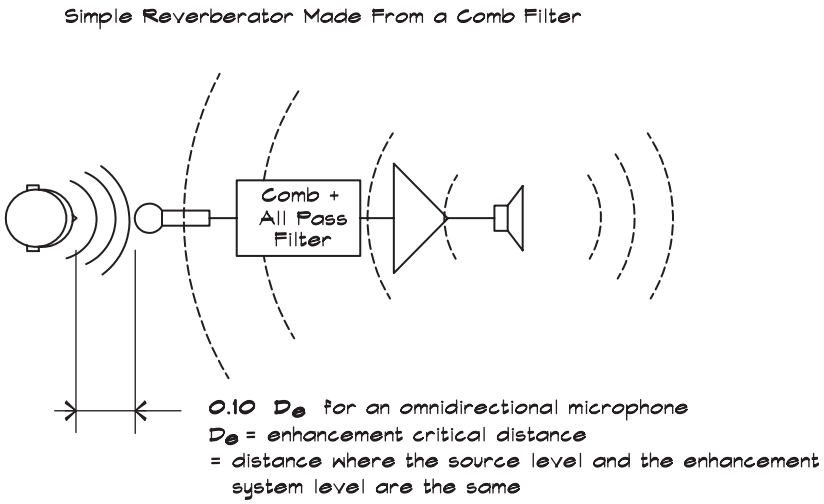
The earliest systems consisted of a microphone, amplifier, and loudspeaker with either a single or multi-tapped delay line. A representative block diagram is in Fig. 20.24. In its simplest form a single microphone picks up the sound in the auditorium, delays it, and returns it back to the room via loudspeakers. Just as was the case in a reinforcement system this process generates a feedback loop characterized by a loop gain. The loop gain is defined as

$$\text{Loop Gain} = \mu = \frac{\text{Ave Mic Output from Loudspeakers}}{\text{Ave Mic Output from Source}} \tag{20.1}$$

The system concept is the same as that shown in Fig. 18.13, except the microphones are not located as close to the talker and the feedback loop between the loudspeakers and the microphone is via the reverberant field. The loop gain is not constant with frequency. It has many bumps and dips, and varies with the position and orientation of the microphone. Clearly when the loop gain equals one we have sustained feedback and, in fact, the value must be much less than one.

The maximum value depends on the reverberation time of the room and the bandwidth of the system. For a broadband system and a 2-second room, the maximum loop gain before the onset of ringing is about -12 dB (Griesinger, 1991). In artificial reverberation

FIGURE 20.24 Single Channel Artificial Reverberator (Griesinger, 1991)

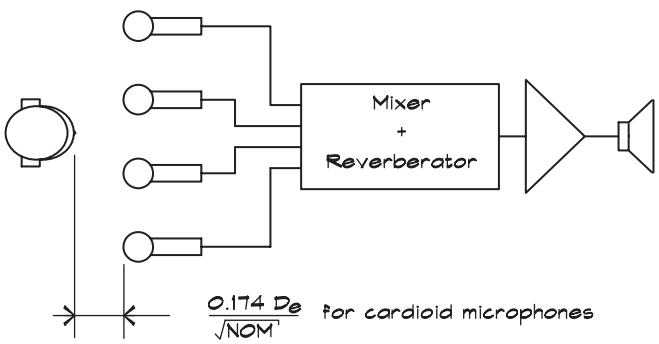


systems the loop gain should be another 8 dB lower or −20 dB to avoid coloration (Krokstad, 1991). This places rather severe constraints on the loudness of the system.

To overcome the feedback limitations the single microphone systems locate the pickup point close to the source. To meet the −20 dB limit, the ratio of the mic distance to the critical distance must be less then 1/10. In many halls the critical distance is on the order of 7 m (23 ft) (Griesinger, 1991) so an omnidirectional mic must be placed no more than 70 cm (2.3 ft) from the source. Although better for feedback, this close miking technique does not yield a representative sample of the sound in the room.

Directional mics can give some additional gain. A cardioid mic has about a factor of $\sqrt{3}$ (4.8 dB) increase in distance for the same feedback. For hypercardioids the factor is 2 (6 dB). But at 1.4 m (4.6 ft) you cannot single-mic an orchestra. Using multiple mics close to the sources overcomes some of this problem at a cost of reduced gain and considerable inconvenience. If the mics are mixed before processing as in Fig. 20.25, the nom correction reduces the minimum distance by $\sqrt{\text{nom}}$.

FIGURE 20.25 Microphone Mixing before Processing Increases Feedback Stability (Griesinger, 1991)



In an effort to achieve greater gain stability, systems with separate channels of reverberation, illustrated in Fig. 20.26, have been developed. If individual microphones and loudspeakers are separated so they do not influence each other, and if each channel meets the −20 dB gain limit, a channel will not be driven into an unstable feedback condition by the addition of another. Even if a single microphone is used to feed multiple reverberators before mixing, the same effect can be achieved. The overall reverberant level in the room and the reverberation time can be increased by 10 log (number of reverberators) and the source to microphone distance can be increased by a factor equal to the square root of that number. In practice each channel can increase the reverberation time by about 1% (Krokstad, 1991), so the total number of reverberators and loudspeakers needed may range from 50 to 100 in a typical system without additional processing.

Further improvements are possible through processing. The RODS (Reverberation on Demand System) system developed by Acoustic Management Systems uses a sophisticated gating arrangement pictured in Fig. 20.27. The logic module senses whether the sound in the room is rising or falling with time. The input to the delay is gated on when the room level is rising or constant and the output from the delay is connected to the loudspeakers only when

FIGURE 20.26 Multichannel Artificial Reverberation System (Griesinger, 1991)

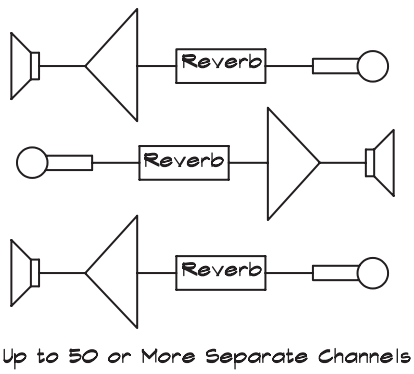


FIGURE 20.27 Reverberation on Demand System (Griesinger, 1991)

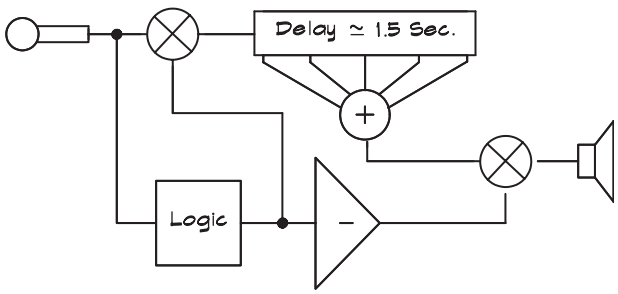
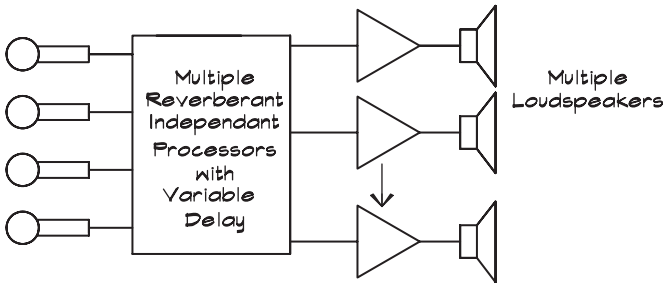


FIGURE 20.28 LARES Reverberation System (Griesinger, 1991)



the level is falling. Feedback is reduced in this manner. Lateral energy is not augmented during continuous music but only during the times when the sound is decaying.

The LARES (Lexicon Acoustic Reinforcement and Enhancement System) system developed by David Griesinger (1991) uses multiple reverberators, shown in Fig. 20.28, which change their properties as a function of time. This is done by modulating the delay lengths in the reverberation algorithm (Frenette, 2000). Griesinger notes that a delay modulation results in a pitch shift so that the modulating source must be chosen carefully. For example, if random number generators were the source of the modulation it would be possible for all reverberators to go in the same direction at the same time, producing a noticeable pitch shift in the reverberation. A traffic cop in the software can sort this out. The variation must be sufficiently rapid that the autocorrelation function is zero for any two reverberators or for an individual reverberator with itself for a delay greater than 1 second. The randomization of the reverberator behavior allows the signal from a few (2–6) microphones to feed a large number (16–64) of processors and an even larger number of loudspeakers (up to 300+). Processing in the reverberation chain yields impressive increases in gain and the microphones can be moved back to 12 m (40 ft) from the source and still provide a substantial increase in reverberation time.

The overall loop gain follows the relationship

$$\begin{aligned} G_\mu \leq & -12 \text{ dB (stability)} + 4.8 \text{ dB (cardioid microphones)} \\ & - 6 \text{ dB (safety factor)} + 6 \text{ dB (time variance)} \\ & + 10 \log (\text{number of reverberators}) \end{aligned} \tag{20.2}$$

Even for a modest number of reverberators the system loop gain can be positive and the reverberation time, which follows the loop gain

$$G_\mu = 10 \log \frac{T_{60}(\text{enhanced})}{T_{60}(\text{natural})} \tag{20.3}$$

increases accordingly.

Reinforcement systems can also control the time at which the energy is released. In rooms designed for speech sounds arriving at the listener earlier than 50 ms after the direct sound are useful for intelligibility particularly if they are in the median plane. Reflections arriving in the 50 to 150 ms range are not helpful for speech intelligibility and do not

contribute significantly to envelopment. In rooms designed for music it is desirable to be enveloped or surrounded by the sound field, particularly at frequencies below 300 Hz with sound arriving after 150 ms. In electronic enhancement systems low pass filtering can be helpful. Much of the positive effects fall into the frequency range below the 500 Hz, so more energy can be added to a space without increasing gain. A typical system attenuates signals above 500 Hz by 6 dB.

Some concert halls such as Boston enjoy early reflections from the shell or stage house, whereas others like Amsterdam have few reflections in this time period. Concertgebow is nevertheless highly intelligible since it has little masking in the middle time range, while enjoying excellent envelopment from late arriving reverberation. By using electronic enhancement, the sound energy in the time period beyond 150 ms can be altered without adding excessive energy to the 50 ms to 150 ms time period so that the extra reverberation will not degrade clarity.

21 DESIGN OF STUDIOS AND LISTENING ROOMS

21.1 SOUND RECORDING

Early Sound Recording

From the time of Thomas Edison's phonograph in 1877, up until about 1947 when practical tape recorders became available, recordings were made in real time as a single channel directly to the storage medium. The original Edison system employed a stylus attached to a diaphragm that scribed a grove into aluminum foil wrapped around a rotating cylinder. Later models used a wax cylinder, which could more easily be duplicated and stored. In 1887 Emil Berliner invented flat-disk recording that allowed straightforward manufacturing, although the initial quality was poor. Before 1925 all recording was done mechanically, with vertical hill and dale cuts in the surface. Interestingly, stereo recording was invented in 1929 by Arthur Keller of Bell Labs and later formulated by Blumlein in 1931 in its current 45°/45° groove modulation format (Alexandrovich, 1987), but the first commercial stereo recordings did not appear until 1957.

The early flat records rotated at a rate of 78 revolutions per minute (rpm), with information impressed, first on one and later on both sides of the disk. The record players were entirely mechanical. A sharpened needle stylus was attached to the center of a diaphragm, which was coupled to an expanding horn megaphone. A crank handle that wound a coiled spring provided motive power. Speed was controlled by a mechanical governor. In console units, openable doors at the end of the horn controlled the level. Since the first engineers had no temporary storage capability, there was no ability to manipulate or playback music for post production and thus no need for post-production facilities. Recordings were made with the duplicating equipment in the same room as the musicians. The sound level during a recording session was controlled by positioning a ball of yarn in a large megaphone used to concentrate the sound energy onto the pickup diaphragm (Alexandrovich, 1987).

With the invention of the audion (vacuum) tube by Lee De Forest in 1907, the audio amplifier by Harold D. Arnold in 1912, and the condenser microphone by E. C. Wente in 1917, sound could be converted to electrical signals, and used to drive a groove-cutting lathe to make a master disk from which duplicates could be pressed. Bell Telephone Laboratories

carried out experiments in "auditory perspective" and demonstrated three-channel stereo transmission on telephone lines in 1933 (SMPTE, 2001).

Sound for motion pictures had many originators. W. K. L. Dickson, working in Thomas Edison's laboratory, invented one of the early recording systems in 1895. It was offered to the public in the spring of 1895 as the Kinetophone and consisted of a recording on an Edison wax cylinder along with the Kinetoscope, a box housing a rotating strip of photographs that were observed through a peep hole. The sound was transmitted to the viewer via two rubber ear tubes.

In the silent film era of the 1910s, orchestras or theater organs played a musical score accompanying the film. Loudspeakers were later placed in the orchestra pit to replicate the musicians and still later added behind the screen for dialog—a system that required manual switching between the two by the projectionist. The vitaphone, a system of multiple long-playing (33.3 rpm) records, was introduced in 1926 as a way of recording music, but not dialog, for film. The first motion picture with dialog was *The Jazz Singer* in 1927. In 1928, Disney's *Steamboat Willie* appeared, featuring the first soundtrack created in post production including music, dialog, and sound effects.

Walt Disney's 1940 film *Fantasia* (Garity and Hawkins, 1941) was a giant technical leap forward, employing four channels of recorded information, three separate audio channels, and a tone control channel using a variable-density optical system printed on 35 mm film. The system used two linked projectors, one for the film and one for the sound. The film included an optical mono mix of the soundtrack as a backup, a scheme that is still in use today. It introduced several other innovations for the first time including multichannel surround, the pan-pot, overdubbing of orchestral parts, simultaneous multitrack recording, and three directivity-aligned loudspeakers located behind the screen (SMPTE, 2001).

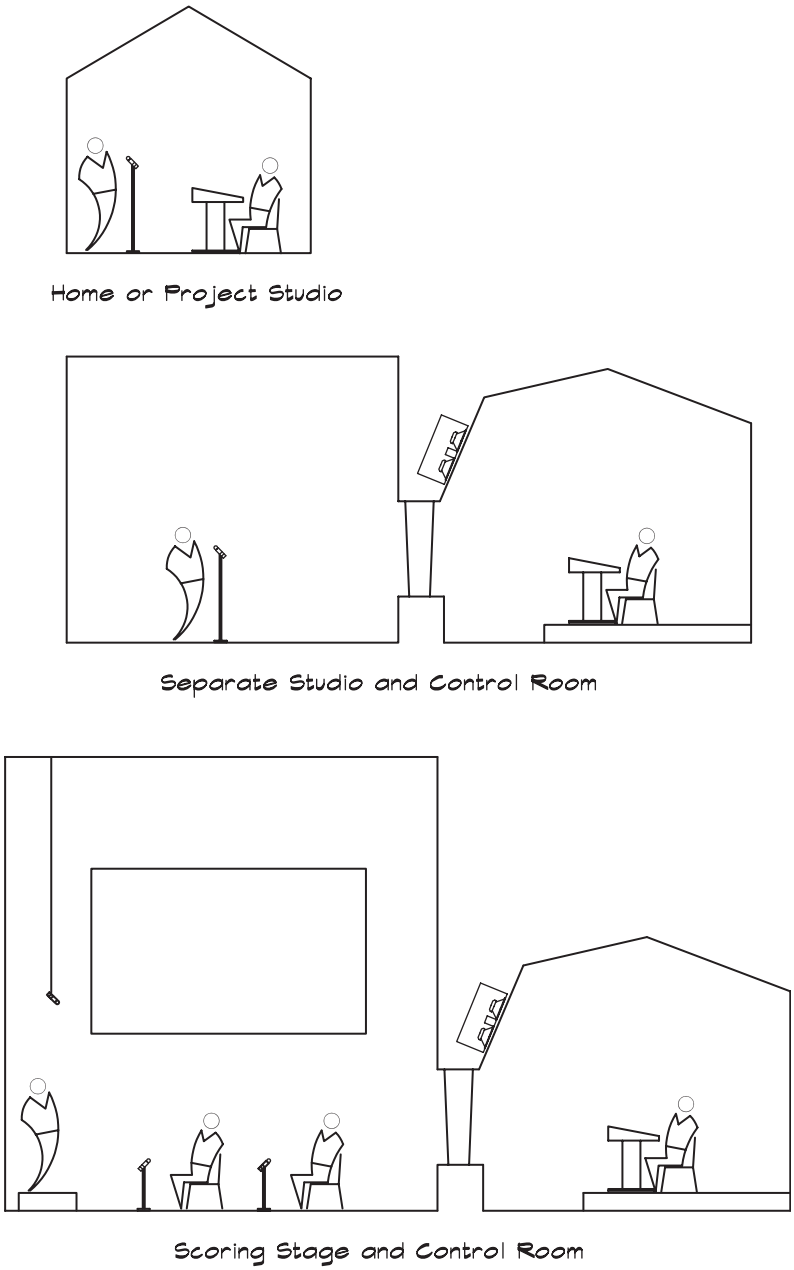
The Disney engineers also originated many new techniques including the use of multiple optical recorders called dubbers (short for doublers) to do mixing. Musicians were seated in a large stage and played the score while the film was being screened, a technique that is still used today. The conductor watched the film listening to timing cues, called a click track, through a single headphone. Sound effects, later nicknamed "Foley" after Jack Foley, a sound editor at Universal Studios, were produced using walking surfaces and clever devices manipulated by hand. Dialog was recorded separately and added in post production, a process called automatic (sometimes automated) dialog replacement (ADR). The three components of film sound—music, effects, and dialog—were combined in a large dubbing theater with mixing consoles located near the middle of the room.

With the development of magnetic tape in Germany in the 1940s, recorded sounds could be played back and manipulated after the performance. Multitrack tape recorders became available in the 1950s and artists such as Les Paul raised looping or overlaying of recorded material to a fine art. With the ability to record and erase, the need arose to listen to the material during the post-production process and the room became part of the audio chain. This led to the development of studios specifically dedicated to sound recording and playback.

Recording Process

The goal of recording and subsequent playback is to deliver an experience that accurately recreates the original performance. Although this is still the object of most recordings, some performances are never heard by an audience in their original form and exist only as electronic

FIGURE 21.1 Types of Sound Studios



signals. Their sole interaction with the architectural environment comes on playback in a listening room, and even this can be bypassed through the use of headphones. Most commercial recordings of music are done in rooms designed specifically for that purpose, carefully crafted to contribute positively to the process. Recording studios vary with the type of music; a simplified overview is given in Fig. 21.1.

In the simplest case, the performance and the recording both take place in one room. An instrument, such as a keyboard, is used to create the sounds that are monitored, either

on loudspeakers or headphones, and recorded through a small mixing console by a storage device. In this example the recorded sound never passes through a microphone and its only interaction with a room is during monitoring.

At a medium scale, studio musicians might play on acoustic (an unfortunate use of the term, which has become part of the culture) or electronic instruments, which are recorded using separate microphones or transducers built into the instrument. A recording might be made with all the musicians present at the same time, or with musicians playing their parts in separate studios, sometimes continents and weeks apart. When musicians play together, particularly loud instruments such as drum sets should be isolated in separate dedicated rooms so that their sounds do not bleed into the other microphones. Vocalists or sources needing a special environment, such as pianos, can also be placed in separate spaces.

A large symphony orchestra is recorded in a hall or scoring stage, using separate microphones for each section, plus a stereo pair on the centerline of the room, plus several more distributed throughout the room. Groups of players can be separated by portable barriers called gobos, but this affects the overall orchestral balance. Most mixers prefer to balance the orchestra using instrument placement and microphone location without resorting to isolating barriers (Murphy, 2001).

Recording Formats

The stereo format has dominated the recording industry since its commercial introduction in 1957. Since we have two ears in the horizontal plane, a pair of loudspeakers in the same plane can provide critical lateral cues for source localization. Recording techniques have improved to the extent that a phantom image can be reliably produced between or even outside the two loudspeakers. In the best combinations of recording and playback, a soundstage is created with depth as well as width, in which the listener can hear the instruments in their original positions.

Traditional film audio used right-center-left loudspeakers located behind the screen. Dialog was placed in the center and music could be distributed right and left. In 1973 a low-frequency effect called SenSurround was introduced for the Universal Studios' film Earthquake. In the earliest versions the signal was produced by a low-frequency noise generator and later by a recording on the film. At first there was a concern that this high-energy effect would damage theaters. The solution was to play the film 10 dB louder than the actual show levels, without an audience present, and see what shook loose (Stern, 1980). If nothing did, the theater was approved for showing.

Themed entertainment venues have been able to feature custom theaters with unique multitrack shows. In theme park attractions, audio animatronic (AA) characters, so named because the movements were controlled by audio tones, were designed with individual point sources for each figure. Loudspeakers were built into props and set pieces, or if the character is large enough as in the case of King Kong, into the figure itself. Multiple loudspeakers could be located throughout a theater or ride and the sound mixed down by engineers within the actual venue. Motion could be simulated using voltage controlled attenuators (VCA) under the control of a show control computer (Long, 2001). In the Sanrio Puroland Theme Park in Tokyo, Japan, a system designed by the author in the 1980s for the Time Machine of Dreams Theater utilized three loudspeaker clusters located behind the film screen with eight side and three rear loudspeakers each under individual computer control. Multiple overhead loudspeakers also were included. This system made complex motion simulation

possible, while allowing different films to be screened by changing the computer control software.

Recently the film industry has adopted a standard 5.1 system utilizing left, center, right, left surround, and right surround audio tracks (5) with a low frequency/effects (LFE) embedded bass track (.1). This is a simpler version of the multitrack formats used in theme parks and has the advantage of not requiring computer control. The sound is recorded onto the film or onto a digital disk and played by an outboard processor. Several competing systems are available. With the advent of digital projection equipment, both audio and video software in the future will be sent directly to theaters or the home via cable or wireless transmission. The 5.1 system or another similar multichannel surround system will probably become standardized for home use. There are a number of other combinations in service including five loudspeakers behind the screen, separate side and rear surrounds, one separate or two embedded bass tracks, but these have not found as wide an acceptance.

21.2 PRINCIPLES OF ROOM DESIGN

Rooms for sound recording and playback are used for three basic functions: listening, recording, and mixing. Rooms serving each function require a somewhat different environment, although there are many overlapping design considerations. All must be designed with a knowledge of the effects of scale based on the frequencies of interest. The smaller the room the greater the importance of low-frequency phenomena such as standing waves.

Standing Waves

The primary difference between the design of studios and listening rooms and that of auditoria and concert halls is in the control of standing waves. There are normal modes in all enclosed spaces but those in small rooms fall into the range that we can hear. From Chapt. 8 we recall that resonances in a closed-closed tube occur because the sound pressure is forced to a maximum at the ends, resulting in a pressure minimum in the center. The natural frequencies for the axial modes in one dimension are given by

$$\mathrm{f_n} = \frac{\mathrm{n\,c}}{2\,\ell} \tag{21.1}$$

where n = 1, 2, 3, ... In a room such as an auditorium, perhaps 60 feet (18.3 m) wide, the first mode (n = 1) will occur at about 9 Hz, well below our range of hearing. In a small studio, 15 feet (4.6 m) long, the first mode occurs at about 38 Hz, well within the audible range.

In a rectangular room the natural frequencies of the room modes were given in Eq. 8.43 and are repeated here:

$$\mathrm{f}_{l\,\mathrm{m\,n}} = \frac{\mathrm{c}_0}{2}\left[\left(\frac{1}{\ell_\mathrm{x}}\right)^2 + \left(\frac{\mathrm{m}}{\ell_\mathrm{y}}\right)^2 + \left(\frac{\mathrm{n}}{\ell_\mathrm{z}}\right)^2\right]^{\frac{1}{2}} \tag{21.2}$$

where the *l*, m, and n are integers (0, 1, 2, . . .). To calculate every possible frequency it is necessary to work through all combinations of the integers. In practice however, only the lowest few really matter since most of the energy ends up there.

When two integers are zero, the wave form, which is called an axial mode, is a one-dimensional standing wave between two parallel surfaces. When only one index is zero the wave pattern is called a tangential mode, and includes reflections from the four surfaces in a plane parallel to the remaining two. When all indices are nonzero, the mode is oblique and includes reflections from all surfaces. In practice axial modes are the most important, followed by tangential. Oblique modes are seldom highly energized.

If the room is open on one side it can be modeled as a closed-open pipe. In this case the sound pressure has a maximum at the closed end and a minimum at the open end so there is no pressure extreme in the center of the room. The resonant frequencies are given by

$$f_n = \frac{(2n - 1)\, c_0}{4\,\ell} \tag{21.3}$$

where n is an integer 1, 2, 3, . . . and so forth. Our 15 ft (4.6 m) room now has a fundamental frequency of 19 Hz, half of what we had with the closed-closed condition. The second mode is at 56 Hz as shown in Fig. 8.3, and the null is displaced away from the center. In practice, of course, we do not leave one end of the room open but we can approach this boundary condition by making the wall very absorptive at low frequencies.

This simple calculation illustrates several important points. First, if we want to avoid a low-frequency null in the center of our room, we do not build our room out of a material such as concrete, which is highly reflective at low frequencies. Second, low-frequency absorptive treatment applied to one end of a room makes the room appear acoustically longer by lowering the first-mode frequency, and moves the first node away from the center of the room. This is a good thing for a control room since the mixer sits near the middle and does not want to experience so called "bass suckout."

When the room has the same dimensions in two orthogonal directions the pressure nulls will occur at the same frequencies and in the same place in the room, that is, the center. If room dimensions are even multiples of each other, then overlapping nulls also occur in the center. For the simple example of a 2:1 length ratio, the second mode of the longer dimension coincides with the fundamental of the short dimension. Although it is unusual to have room lengths of this ratio, it is not uncommon to have a length-to-height ratio on this order of magnitude. Thus we try to select room dimensions that are not integer multiples of each other. Several authors have suggested preferred ratios, which are given in Fig. 8.9.

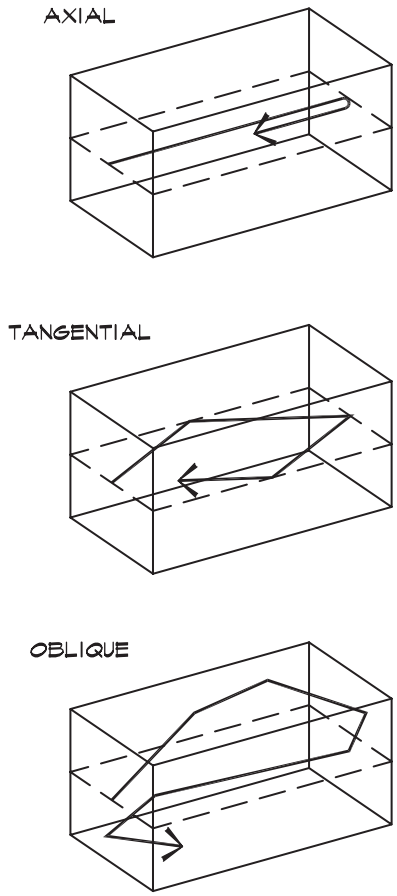
Axial modes are not the only ones of concern. Figure 21.2 shows the three types of modes that can be self reinforcing, axial, tangential, and oblique. We need to be particularly cognizant of the possibility of room modes, especially when pressure maxima or minima coincide with the loudspeaker or listener position.

Bass Control

The control of low-frequency energy in small rooms is a matter of some delicacy. There are several main techniques that are used:

1. Overall control through low-frequency panel absorbers principally on the rear and side walls
2. Application of deep layers of absorptive material sometimes used in conjunction with panel absorbers
3. Construction of Helmholtz resonator cavities

FIGURE 21.2 Visualization of Room Modes Using Rays (Everest, 1994)



4. Locating the bass loudspeakers so that they do not excite the principal modes
5. Locating the main listener position so that it does not coincide with a major low-frequency node

These techniques, which can provide low-frequency control, reveal those steps that do not. Equalization, for example, can be used to raise or lower the overall level but cannot prevent the occurrence nor shift the location of nulls and interference patterns. Diffusion has little effect on low-frequency room modes. Small free-standing absorbers or localized patches of absorbent material have a minimal effect.

A drywall partition is an effective low-frequency panel absorber, as long as the surface is not built out of too many layers. Figure 7.41 gives examples of their properties. Even walls having two layers of drywall can be absorptive below 100 Hz. Panel absorbers can be tuned to the first axial mode of the room using Eq. 7.105. Table 21.1 shows combinations of possible materials and depths based on this equation.

Broadband absorbers can be constructed by layering pressed fiberglass boards over panel absorbers or by using very deep layers of fiberglass absorption. In cases where there is sufficient depth, wedge-shaped absorbers are employed to ease the transition between the air and fiberglass. The design of wedges for anechoic chambers is discussed in detail

TABLE 21.1 Resonant Frequencies of Panel Absorbers (Hz)

Number of Layers	Depth of the Airspace with Fiberglass Batt (inches)					
(1/2" Drywall)	2	4	6	8	10	12
1	71	50	41	35	32	29
2	50	35	29	25	23	20
(1/4" Plywood)						
1	121	86	74	61	54	52

by Beranek and Sleeper (Beranek, 1946). Tapered fiberglass wedges are 99.9% (−30 dB) absorptive down to a frequency where the wedge depth is equal to a quarter wavelength of the incident sound. Most studios and listening rooms do not have sufficient depth to allow for this type of treatment but where there is the space, a deep absorber can yield impressive results (Sanicola, 1997).

Bass trapping using Helmholtz resonators can also be effective. The resonant frequency of a Helmholtz resonator, which is an enclosed volume with an open neck having a definable area and length, is calculated using Eq. 7.115. Since bass modes have pressure maxima at reflecting surfaces, where the particle velocity goes to zero, the cavity openings are most effective when located at these points—that is, at the edges and corners of a room. Volumes lined with batt insulation have a lower quality factor in their response curve and provide more broadband absorption than unlined spaces. Naturally occurring cavity spaces such as video monitor enclosures, return air plenums, soffits, raised floor platforms, and equipment enclosures can readily serve as absorptive cavities without the need for dedicated areas. Several examples are shown in Fig. 21.3.

The position of the subwoofers is another way of controlling bass buildup in a room. Subwoofers should be separated from the main LCR or stereo loudspeakers and there should be at least two of them, even if they are driven with the same signal. There are several considerations in woofer placement, some of which are conflicting. First loudspeakers located

FIGURE 21.3 Found Space Bass Traps

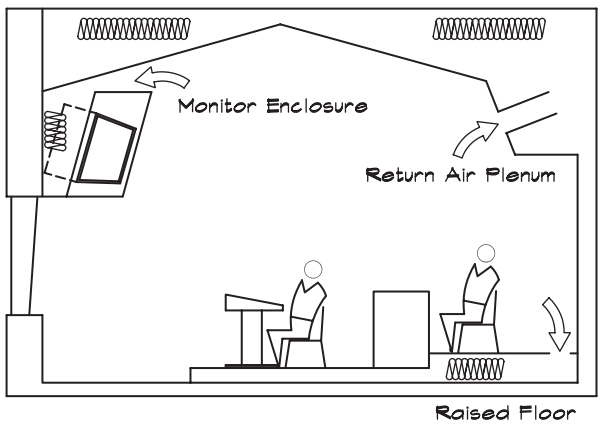
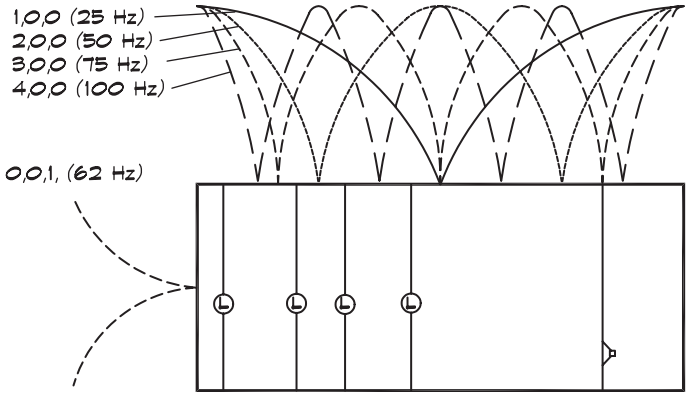


FIGURE 21.4 An Elevation View of the NRC/IEC Listening Room Showing Possible Listener and Woofer Locations (Toole, 1990)



close (within 1/6 wavelength) to a hard reflecting surface will be more efficient, because the geometrical directivity due to the reflections from the nearby surfaces increases the power output for the same voltage input. Single-layer drywall partitions do not provide much increase in sound pressure but solid floors can be helpful. Woofers placed in the corners of a rectangular room will excite the greatest number of room modes since they all have pressure maxima there. However it may not be desirable to drive all the modes, since the listener may be sitting at a major node. In this case, it may be better to position the subwoofers at an intermediate point along the floor-wall junction to avoid pouring energy into the fundamental axial modes. By using two (or more) loudspeakers that can be moved around, optimum positions may be determined experimentally. The listener position can be controlled with a knowledge of the locations of the bass nodes. Figure 21.4 gives an example of possible loudspeaker and listener positions in a rectangular room relative to the locations of the low-frequency maxima and minima.

Audible Reflections

One design objective in recording and listening spaces is to minimize the negative effects of the room on the acoustic environment. In an ideal listening room the sound is transmitted from the loudspeaker to the listener with little or no coloration. Similarly, in a studio the sound of the performer should be clearly transmitted to the microphones. Since room reflections are unavoidable, experiments have been undertaken to determine the audibility of an individual reflection. Early experiments by Haas (1951) used two loudspeakers in front of the listener at an included angle of 45° with speech as the test signal. One loudspeaker was delayed with respect to the other and the listener was instructed to adjust the level of the reference loudspeaker until the two sources were perceived to be of equal loudness. The upper curve shown in Fig. 21.5 (Toole, 1990) was the result.

Figure 21.6 contains the results of later experiments that reexamined human reaction to a combination of delay and level, again using speech as the input signal. These yielded a variety of results, indeed a nearly continuous range of reactions, including equal-loudness lines significantly different from that found by Haas. Based on these experiments it can be said that there is no one single point at which the two signals were perceived as unified, since even when the second signal was 20 dB below the first it was still perceptible. As a

FIGURE 21.5 Various Measures of Significance of a Lateral Reflection (Toole, 1990)

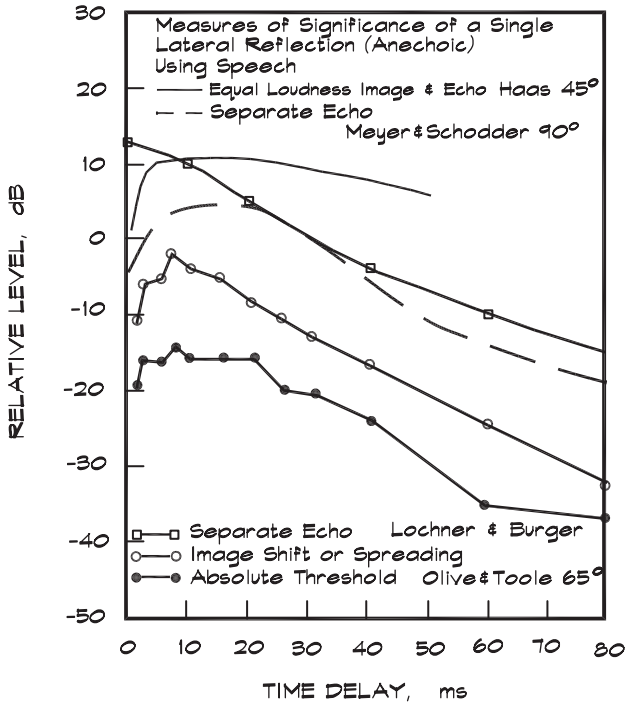
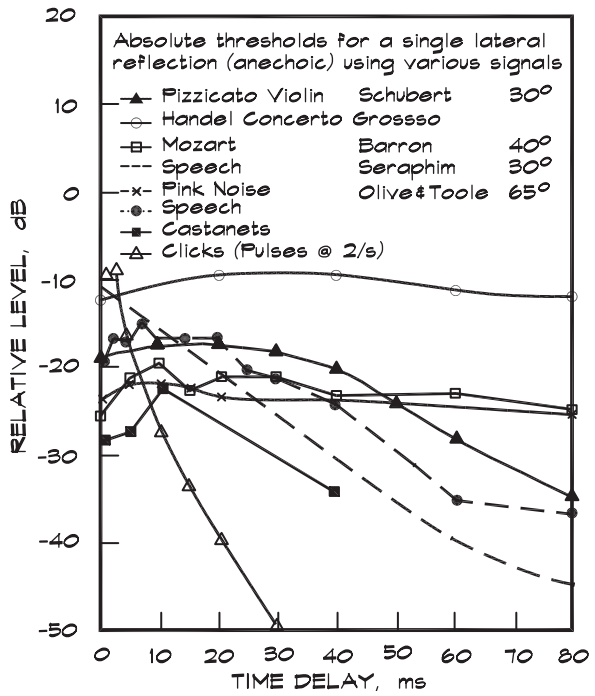


FIGURE 21.6 Absolute Thresholds for a Single Lateral Anechoic Reflection (Olive and Toole, 1989)



consequence Toole argues that the masking of a late reflection by a series of early reflections has probably been overemphasized in small listening rooms.

These results do not negate the effectiveness of delayed loudspeakers in sound reinforcement systems. Because they are A-B comparison tests, a shift in image can trigger a noticeable response. In a delayed loudspeaker system a change in image direction is one of the design objectives. Fixed installations do not feature A-B comparisons except during setup. Second, these tests were done in the horizontal plane where we are most sensitive to perceived direction. Time delays and level changes from loudspeakers located in the medial plane are less perceptible.

Separate experiments by Olive and Toole (1989) tested the influence of the source material as well as the level and delay. It was found that the audibility of reflections using short-duration signals varied greatly from those obtained with music and speech. Figure 21.7 illustrates the differences; although there was forward temporal masking, the nearly horizontal shift in level with delay time measured by Haas was observed only in the case of music. Speech thresholds dropped linearly with delay time and short-duration signals provided very little masking.

The audibility of reflections is also dependent on the reverberation time. Figure 21.8 shows the results of psychoacoustic studies by Nickson, Muncey, and Dubout, (1954). Here the source was speech and the "echo" signal was electronically delayed and presented through the same loudspeaker as the original. Note that the lower line representing 20% disturbed follows the reverberation time line quite closely. These straight lines are drawn based on the assumption of an exponential decay in the room. A reverberation time of 0.5 sec is equivalent to a 10 dB decrease in (0.5/6 = 0.083 sec). This is plotted as the top curve (Everest, 1994). The upshot is that the deader the room, the more noticeable are the individual reflections.

FIGURE 21.7 The Results of Two Psychoacoustical Listening Tests (Nickson, Muncey, and Dubout, 1954)

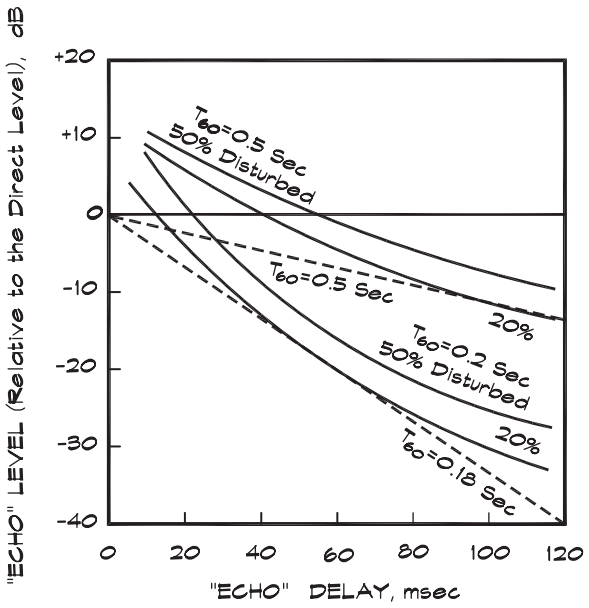
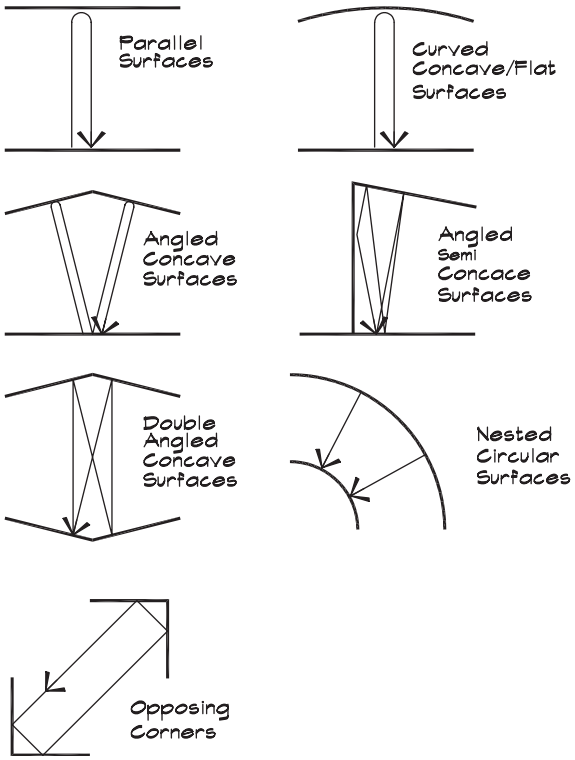


FIGURE 21.8 Configurations that Generate Flutter Echo Reverberation

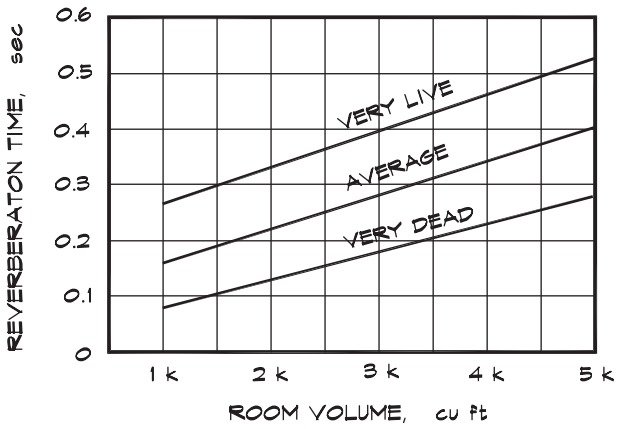


Everest (1994) divides the reaction to a reflection into four regions according to the time delay. Region 1, where directional cues are determined, is less than 1 ms; Region 2 is up to about 50 ms, where the integrating effects of the ear occur; Region 3 is an intermediate zone, which depends mostly on reverberation time, and Region 4 is that of a long-delayed reflection where the curves in Fig. 21.6 flatten out and approach the horizontal.

Clark (1983) performed an interesting series of studies to measure the effects of the audibility of time delays, phase shifts, and frequency shifts in listening rooms having a stereo pair of loudspeakers at 10' (3 m) distance and 60° included angle. This standard stereo configuration, even when a monaural signal is fed to both loudspeakers, is preferred by most listeners who describe the sound as "fuller, more solid, or having depth" in spite of the fact that the arrangement produces significant comb filtering at the listener's ear. Clark then used a single loudspeaker with a 24" × 30" reflecting surface to recreate the comb filter effect of two loudspeakers. In this case, when the reflecting surface was in the "wall" orientation little audible effect resulted, much less than the stereo loudspeaker arrangement with a monaural signal. With the reflector in the "table-top" orientation the effect was more noticeable but still small. A third experiment was done wherein an electronic filter was used to generate a comb filter, fed to one loudspeaker. Here the result was very noticeable and degrading, giving a nasal quality to voice reproduction.

In an extension of this work, Clark (1983) used electronic manipulation to introduce phase shift, time delay, and frequency shift into the signal path. Phase shifts of up to 2700° were inaudible as were time delays up to 10 msec. He speculated that time delays of up to 30 msec would also be inaudible but did not test in this range. Where all three effects are present,

FIGURE 21.9 Suggested Reverberation Times for Control Rooms (Fierstein, 1979)



he judged that frequency shifts were the most audible. Stereo reproduction was preferred to single loudspeaker, even when a wall reflector was present. This likely is due to the fact that side-wall reflections help fill in the stereo comb filtering.

It is good practice, in control room design, to minimize the energy scattered to the mix position from surfaces immediately adjacent to the main loudspeakers. Likewise, it a good idea to provide a series of reflections that transition smoothly into a reverberant decay. This can be done using diffusion, surface orientation, and absorptive treatment to avoid high-energy reflections. In most rooms the side walls and ceiling are flared outward to direct the first reflection away from the mixer position. Rear walls are treated with absorption, diffusion, or a combination of the two.

Flutter Echo

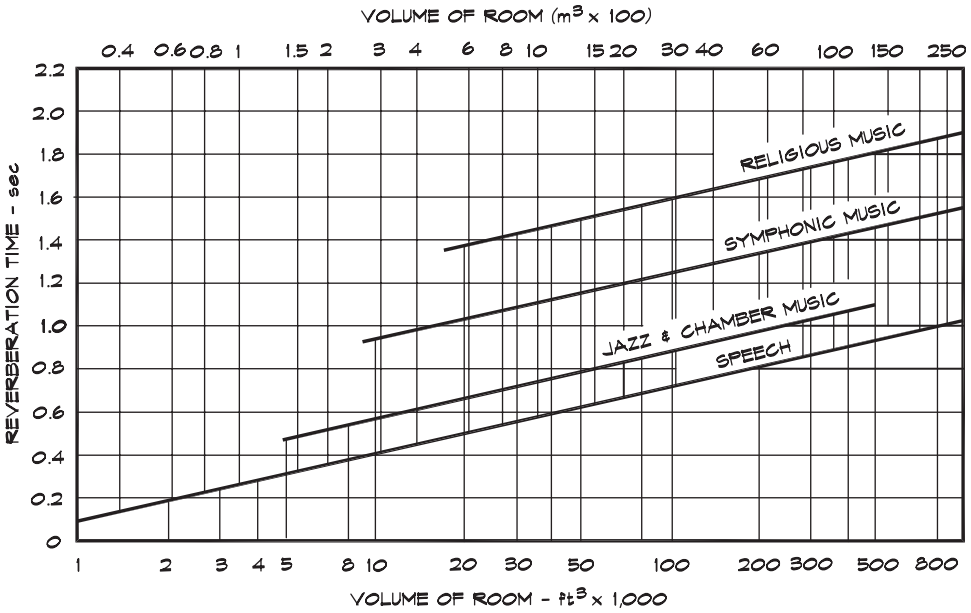
In a music practice room or recording studio, where the source and receiver are close together, repeated multipath reflections known as flutter echo can be disturbing. The most commonly encountered example is between two parallel walls, but there are numerous geometric combinations capable of producing this phenomenon. Several examples are shown in Fig. 21.8 but these are not all-inclusive. Flutter can be treated with absorbing material, diffusing elements, staggered reflectors, or by canting the reflecting surfaces. The latter approach is not always effective since a third surface may complete the flutter loop. In the design of small rooms closed reflection loops of this type can be expected and treated with surface applied materials. A surface canted 1:12 suffices to eliminate the problem as long as the loop is not closed by other surfaces.

Reverberation

The reverberation time in small rooms is still an important design parameter. Figure 21.9 shows recommended values of the reverberation time for control rooms. The use of these data should be based on the program material to be played in the space. Rooms designed for the playback or broadcast of speech should be very dead; those for music may be somewhat more reverberant.

For average control room volumes preferred reverberation times fall into the 0.3 to 0.4 second range. In recording studios the reverberation times are variable, and range

FIGURE 21.10 Reverberation Times for Studios in the 500–1000 Hz Range (Doelle, 1972)



between 0.8 and 1.5 seconds, depending on the size of the room. Figure 21.10 gives a number of values based on room volume and the type of music. For small studios and control rooms, the preferred shape of the reverberation-time-versus-frequency curve is flat.

Scoring stages, designed for orchestral music, are like concert halls with reverberation times in the 1.7 to 2.2 second range, but can be furnished with heavy drapes or other variable absorption elements to reduce the reverberation as needed. In scoring stages a reverberation time that rises at low frequencies is desirable; however, too much low-frequency reverberation can reduce the clarity. Increases below 1 kHz (about half of those shown in Fig. 17.11) are appropriate.

Diffusion

A diffusive surface reflects incident sound in several directions rather than principally in one. Its reflections are termed nonspecular as contrasted to the specular or mirror-like reflections produced by a smooth flat surface. Reflectors such as cylinders can be diffusive in one plane and specular in another. Quadratic residue or quarter-wave slot diffusers can also behave in this way. Multifaceted reflectors can generate specular reflections in several directions and contribute to diffusion in a room although their behavior is not perfectly diffuse.

Figure 21.11 shows in a qualitative way the characterization of a sound interaction with a surface. The components of the process are described using absorption, transmission, and diffusion coefficients represented here by single-number values all less than one. This is a great simplification of the actual behavior, since there is a complicated relationship that depends on the type of incident wave, its angle (both incident and reflected), frequency, and the size of the reflecting surface.

Figure 21.12 illustrates more details of the interaction process. In a specular reflection from a hard surface, most of the energy rebounds at an angle centered on the angle

FIGURE 21.11 Sound Interaction with a Rough Surface

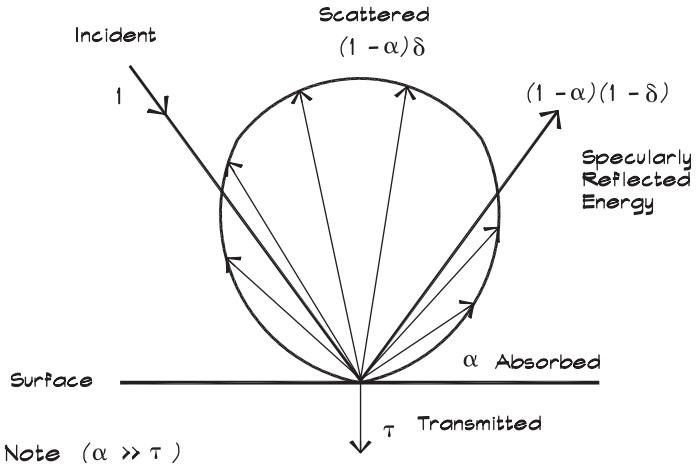
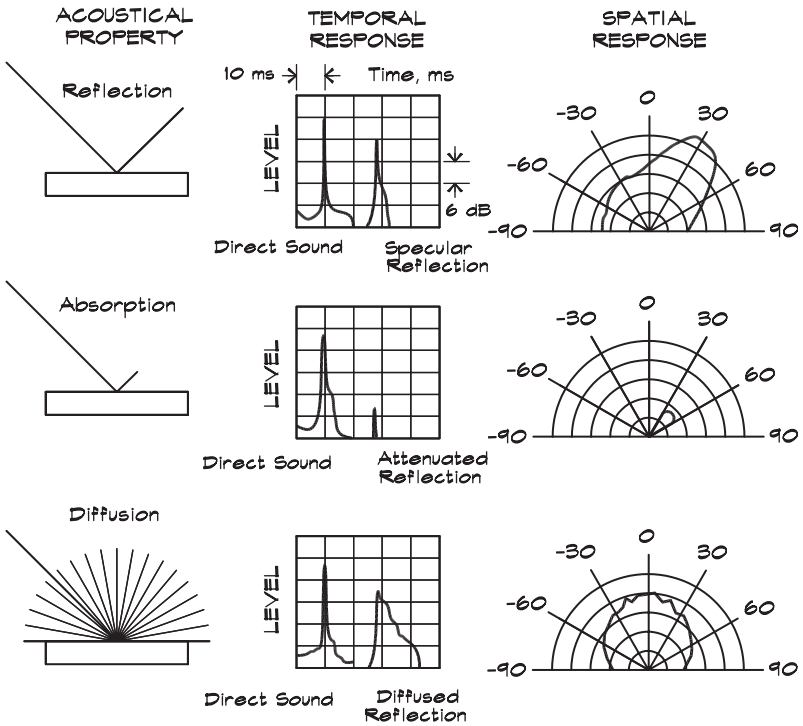


FIGURE 21.12 Comparison of the Spatio-Temporal Properties of Acoustical Interactions (D'Antonio, 1987)



of incidence. The temporal response is constrained to a narrow band determined by the path length. Some temporal smearing is present due to the after vibration of the reflecting surface and the diffuse behavior of the material.

A reflection by a purely absorptive material is also specular, however the amplitude is decreased by an amount characterized by the absorption coefficient. The temporal behavior is narrowed since there is less reradiation by a porous absorber. The behavior of panel absorbers can be expected to show more temporal smearing.

The pattern generated by an ideal diffuser is spatially uniform. There also can be temporal spreading depending on the type of process. For example with quarter-wave tube diffusers, energy is stored in the wells and released over time. A diffuser such as a half cylinder or pyramid would produce a temporal pattern much like that of a specular reflector.

Diffusion in listening rooms and studios can be helpful by contributing to a sense of spaciousness and envelopment. It can help fill in the gaps between specular reflections. Where there are flutter echo problems a diffusive surface can be used to break up the flutter loop without adding excessive absorption. Diffusion can also make absorptive materials more effective by creating a reverberant field, which assures that the acoustic energy will interact with the absorbers. Diffusers do not need to be perfect scatterers to be effective.

Imaging

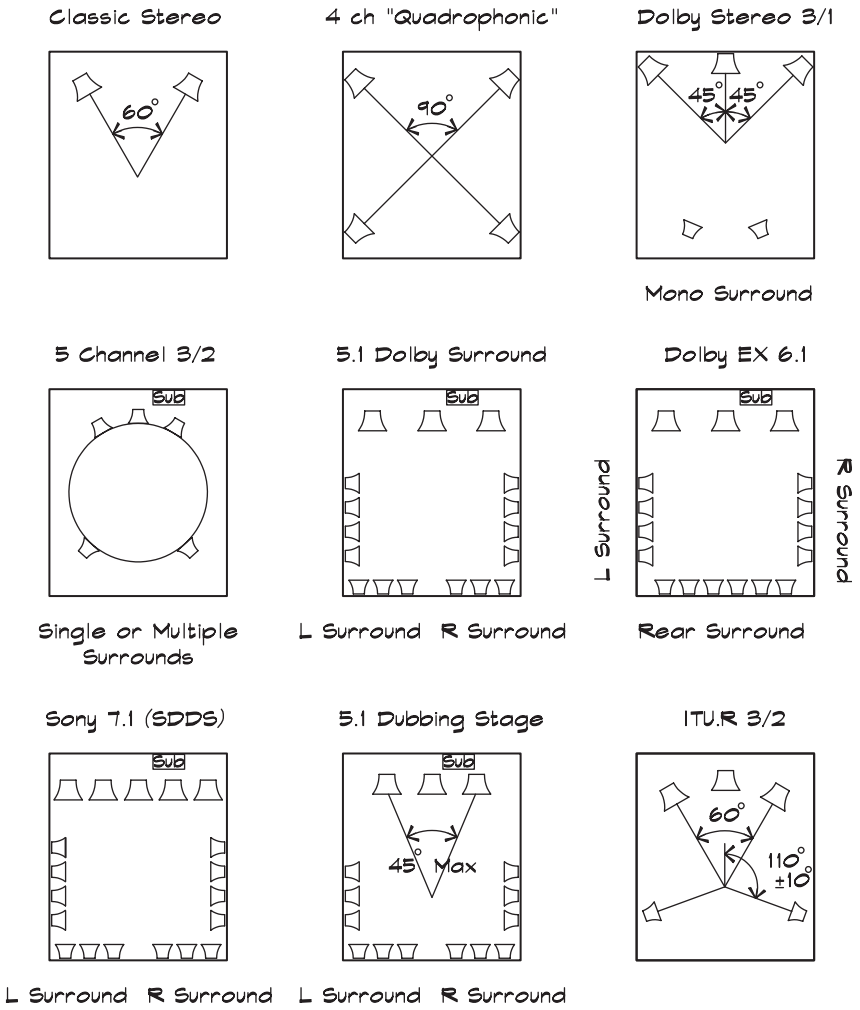
Imaging refers to the preservation of the spatial relationship in the original sound field perceived by the listener. In home listening rooms, designed for high-end audio, the stereo format with a subwoofer is the most common. Stereo loudspeakers are arranged at an included angle slightly greater than 60°, and the listener is about 10 to 12 ft (3 to 3.5 m) away, slightly closer than the separation distance between the loudspeakers. Many loudspeakers are very sensitive to placement, especially separation distance and toe-in angle. By varying these parameters one can often increase definition and detail in listening rooms. Space around the loudspeakers can be helpful in contributing to the soundstage. Where the dimensions allow it, loudspeakers should be placed at least 1.5 m (5 ft) from any walls. Reflections from a nearby wall can color the sound, particularly in the lower registers. Patches of absorptive material on side and rear walls help control rogue reflections. Transondent wall finishes such as wood slats are common in control rooms where cloth-covered panels can be located.

The design of professional listening rooms tends to be format driven. A sampling of possible loudspeaker arrangements is given in Fig. 12.13. Clearly there are so many possible arrangements that not all can be accommodated in a single room. A working room is designed for one principal format, with 5.1 being the most common for film and stereo the most common for TV and recorded music. The number and position of the surround loudspeakers is not always fixed but is fitted to the size and arrangement of the seating in the room.

The arrangement of a listening room can influence the audio image. Reflections from the side walls tend to broaden the phantom stereo image. Conversely side-wall absorption narrows it. Diffusion behind and to the side of the listener increases the sense of spaciousness. Too much diffusion behind the listener has been found to generate an acoustical fog or clutter (Toole, 1998).

Experiments were carried out in Japan by Kishinaga et al. (1979) to assess the best listening environment in which to evaluate home audio products. Four audio professionals did these experiments at Nippon-Gakki (Musical Instruments of Japan) for four different

FIGURE 21.13 Standards for Loudspeaker Placement in Studios, Listening Rooms, and Theaters



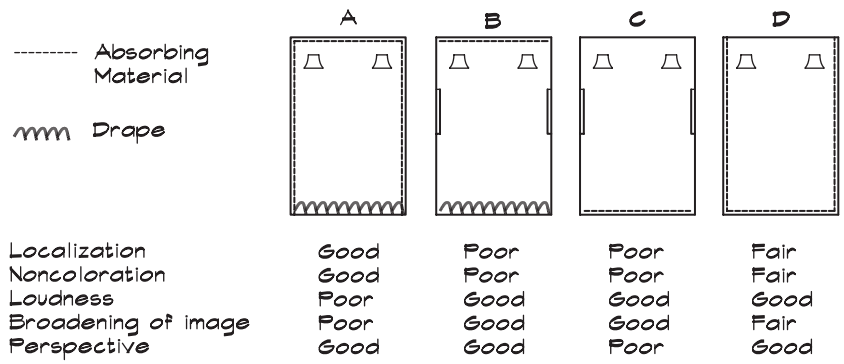
arrangements of absorption in a room. The test room had a hardwood floor and an absorptive ceiling, not a typical home environment. Since these data are over 20 years old, no diffusers were included in the mix of materials. The parameters used for judgment were:

1. Localization of the stereo image
2. Presence of coloration
3. Loudness
4. Broadening of the image
5. Perspective (probably quality of the soundstage)

(Everest, 1994)

The results are summarized in Fig. 21.14. No one arrangement of materials yielded "good" results for all parameters. The sketches in this paper do not show sufficient detail to judge what other factors may have been important. Loudspeaker type, position, as well as

FIGURE 21.14 Summary of the Results of the Nippon Gakki Listening Tests (Kishinaga et al., 1979)



listener position could have contributed to the overall impression. However, based on this study, the highly absorptive configurations seemed to generate the most favorable responses.

Imaging in theater systems, using a left-center-right loudspeaker configuration, is less complicated than in stereo systems. Ideally the three front loudspeakers are located behind a perforated screen about 5/8 of the screen height up from the bottom. The two outside loudspeakers are placed close to the outside edges of the screen. When the screen is not perforated the loudspeakers can be positioned below the screen. This is preferable to two on the side and one beneath. The arrangement of surround loudspeakers is less critical. These are placed to the side and slightly behind the listener.

Noise Control

Background-noise levels in critical listening environments must be kept very low. In studio and listening rooms background levels are a matter of personal taste, ranging from an NC 30 to the threshold of hearing. Table 21.2 gives a starting point. Levels are listed in

TABLE 21.2 Background-Noise Levels in Studios and Listening Rooms

Room Type	Noise Level
Living Room	**NC 25-30**
Music Practice Rooms	**NC 30–35**
Band/Orchestra/Choral Room	**NC 25–30**
Video Post Production	**NC 25–30**
Listening Room/Home Theater	**NC 20–25**
Project Studio	**NC 25–30**
Sound Stage (film)	**NC 20–25**
Dubbing Stage	**NC 20–25**
Control Room	**NC 15–20**
Recording Studio	**NC 10–15**
Scoring Stage	**NC 10–15**
Foley Stage	**< NC 10**

terms of the NC level. A-weighted levels are about 5 dB higher for HVAC-generated noise. Although NC 10 is not formally defined, it is taken to be approximately 4 dB below NC 15 from 63 Hz to 500 Hz and 5 dB below NC 15 at 1000 Hz and above. Note that the higher background levels in music practice rooms are desirable due to the masking they provide.

Low background-noise levels are most critical when there are microphones in use in the room. A project studio often is used to put musical ideas together and to do scratch tracks and so is less sensitive, particularly if monitoring is done on headphones. The noise requirements can also vary with the use of the room. If a final recording secession is done in a project studio it could have the same requirements as a recording studio.

The low background necessary for Foley are due to the practice of mixing at levels well above that at which the original sound was recorded. Background levels are elevated accordingly and can become noticeable unless they are carefully controlled.

Noise Isolation

Klepper, Cavanaugh, and Marshall (1980) wrote an important paper on the design of music teaching facilities. In it, they pointed out that basic planning could help avoid many of the most common problems. Most critical areas should be separated by dead spaces such as corridors, storage rooms, and other buffers. Single-story construction avoids the floor separation problem. By grouping the noisy spaces together, costs can be reduced since the expensive high TL wall does double duty.

Noise isolation between rooms in music rooms and studios can be rather complex. Airborne noise can travel through partitions, HVAC systems, conduits, doors, and windows. Structure borne noise can be carried in common slabs, structural members, piping, and rigid conduit. Footfall noise can be transmitted laterally as well as vertically, particularly when the slab is above grade. A concrete corridor on an above-grade slab, 20 feet (6 m) away from an improperly isolated studio, can render it unusable.

Klepper et al. (1980) included a table showing recommendations on a single-number degree of isolation between music rooms, reproduced as Fig. 21.15. Also included are assumptions on background noise and source level. The table is based on a single-number approximation to the detailed calculation, done by subtracting the A-weighted background-noise level in the receiving room from the C-weighted peak-noise levels in the source room. Note that the difference between the NC levels and the A-weighted background levels given by these authors is based on the assumption that the background-noise spectrum has exactly the same shape as the NC curve. This seldom occurs in practice. More often there is about a 5 dB difference between the NC level and the A-weighted level. Although there is no source spectrum presented, the authors recommend that heavy materials such as grouted concrete blocks be used as a component of the wall material to obtain the necessary low-frequency isolation. Geerdes, Watters, and Hirtle (1971) developed a chart of isolation requirements in terms of octave-band noise reduction plus background-noise levels in the receiving room for practice and teaching rooms. Figure 21.16 gives their results.

Single-number ratings such as those given in Fig. 21.15 are useful for cost estimation and preliminary guidelines, but should not replace careful engineering computations. In professional studios isolation of adjacent studios must often exceed these values, and the transmission loss requirements must be calculated using the actual source spectrum and receiver background levels.

Specific construction details for the separation of music rooms were also published by Klepper et al. (1980). They show three levels of construction, illustrated in Fig. 21.17.

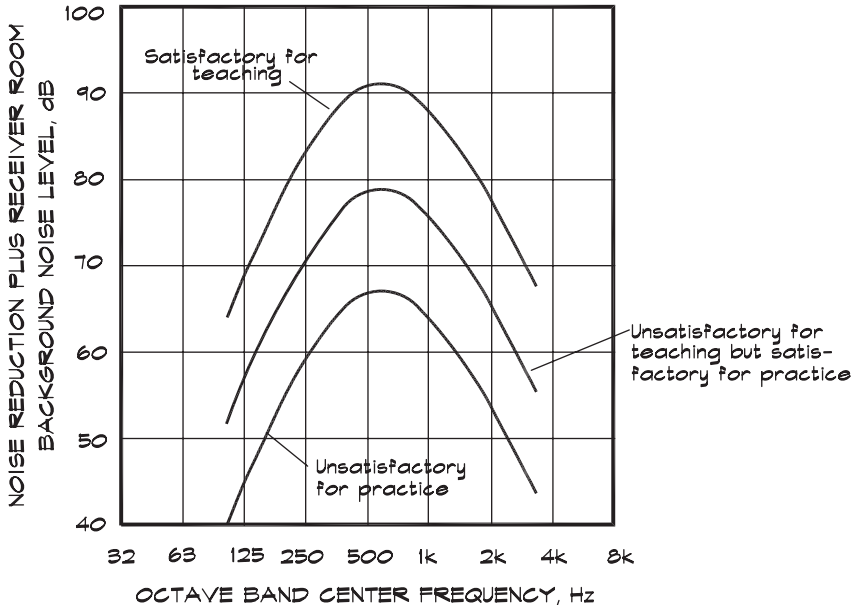
FIGURE 21.15 Acoustical Privacy Criteria for Music Buildings (Klepper, Cavanaugh, and Marshall, 1980)

	RECOMMENDED MINIMUM NOISE INSULATION CLASS (NIC)* BETWEEN MUSIC ROOMS												
	Recommended Minimum Receiving Room Background												Assumed Maximum Source Room Levels**
	NC	dB(A)	Band Room	Orchestral Room	Choral Room	Organ Practice	Music Classroom	Music Listening	Practice Room	Ensemble Room	Electronic Music	Faculty Studio	dB(C)
Band Room	25	36	65	65	65	62	65	65	57	61	62	61	103
Orchestral Room	25	36		62	62	62	62	62	56	59	62	62	100
Choral Room	25	36			59	62	59	59	56	59	62	62	97
Organ Practice	35	44				54	62	62	54	58	58	58	100
Music Classroom	25	36					59	59	56	59	62	56	97
Music Listening	25	36						59	56	59	62	56	97
Practice Room	35	44							48	52	54	52	94
Ensemble Room	30	40								55	58	55	97
Electronic Music	30	40									58	58	100
Faculty Studio	30	40										52	94

*Laboratory-derived sound transmission class (STC) values may be used with appropriate corrections (see text).

**Maximum levels as read on the C-scale of a sound level meter, exceeded only by 5% of peak readings, from several sources.

FIGURE 21.16 Music Privacy for Practice and Teaching Rooms (Geerdes, Watters and Hirtle, 1971)



Construction A rates an NIC 65 and is a completely isolated room-within-a-room construction, which could be used for back-to-back band rooms or similar occupancies. Construction B is similar to A, but the floating floors are replaced with wood floating floors of double 1.5" (38 mm) plywood on 2" (50 mm) fiberglass board, unless the slab is on grade. The drywall on one side is removed. This rates an NIC of approximately 57. Construction C yields an NIC 48 using painted grouted CMU blocks and the same wood floating floors, which can be omitted when the slab is on grade.

Flanking

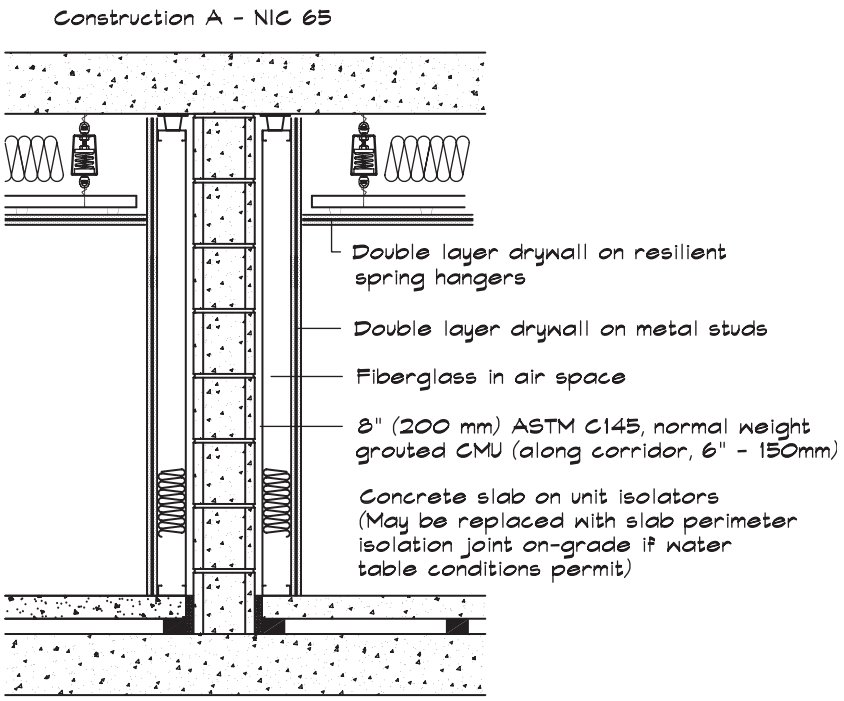
Flanking can significantly degrade carefully designed separations between studios and listening rooms. When a large conduit, duct, or plumbing pipe penetrates a sound-rated wall, as is often the case in TV studios, there can be significant sound transmission through the opening. Small openings can also make a significant difference and should be closed off.

Since there are large bundles of heavy wire in studios and since changes must be made to the cabling, a permanent seal is not possible. In permanent installations the opening is closed off with drywall and plastic pipe packed full depth with fiberglass or safing, and the cracks sealed with caulk on both sides. In flexible installations closed cell foam, fire stop compound, or other malleable materials can be used to seal the ends of a pipe. In some cases commercial isolation clamps are built into the common wall.

Where pipes, ducts, and conduits penetrate separated walls they must not form a bridge between the isolated structures. A structural break is required in these elements, particularly when studios utilize floating construction. Piping can be isolated using flexible no-hub connectors. Flexible conduit can be used and ductwork can include flexible connections.

Occasionally there are cases where a wire way is used to carry cabling between critical spaces. In these cases, it may be possible to fill the cavity with fiberglass batt or safing and

FIGURE 21.17 Separation Walls for Music Spaces to Provide a Given Noise Reduction between Adjacent Rooms (Klepper, Cavanaugh and Marshall, 1980)



Construction B - NIC 57
Remove drywall on one side. Replace floating slabs with two layers of 1 1/2" (38 mm) wood flooring on 2" (50 mm) thick 6 lb/cu ft (100 kg/cu m) pressed fiberglass board. This can usually be omitted when the slab is on grade.

Construction C - NIC 48
Remove drywall on both sides. Replace floating slabs with two layers of 1 1/2" (38 mm) wood flooring on 2" (50 mm) thick 6 lb/cu ft (100 kg/cu m) pressed fiberglass board. This can usually be omitted when the slab is on grade.

cover it with a heavy lid that can be removed to access the wire. The object of any filler material is to block off the air passage between the two spaces or to provide a long path with a high attenuation.

HVAC Noise

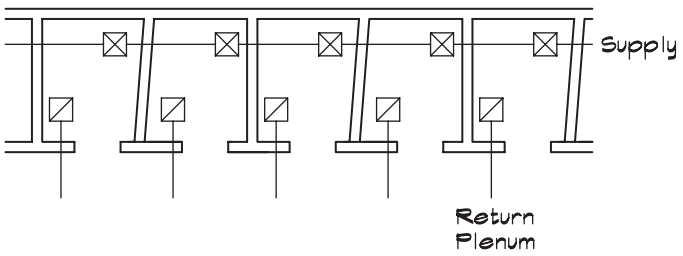
HVAC units are a major source of noise in studios and listening rooms. For low-background levels flow-generated noise is particularly important to control. Terminal devices with adjustable dampers should be avoided. Volume control units should be located well upstream of the termination. Klepper et al. (1980) recommend that for an NC 20, dampers be located at least 15 ft (4.5 m) upstream; for an NC 35, the minimum distance is 5 ft (1.5 m). This assumes flexible duct or sheet metal duct with a minimum 1” (25 mm) liner between the device and the termination.

Both supplies and returns must be ducted. The final run to the terminal unit should be made with flexible duct that is not kinked or severely bent. Lining is required but may not be necessary for the entire length of the duct. The added duct area gained by removing duct liner near the terminal can be more important than the attenuation the liner provides. Silencers, preferably low-frequency traps, are almost always necessary near the mechanical units.

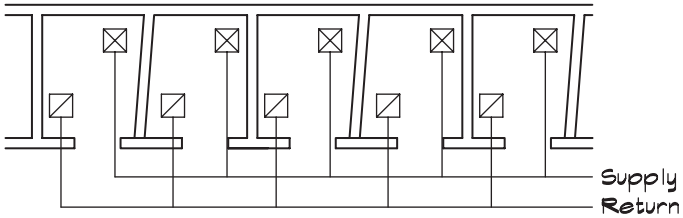
Packaged HVAC units, fan coils, and condensers should not be located directly above studios and sensitive listening rooms. In large studios and dubbing facilities this may require a separate structural support system for the equipment. In small studios it may be sufficient to locate fan coils over nonsensitive spaces such as bathrooms or closets with vibration isolation and silencers as required. Even when packaged rooftop units are remotely located there is still a possibility that low-frequency rumble can be transmitted into the roof via the ductwork. Duct runs, particularly those supported on lightweight roofs, should be vibration isolated. Even when there is a flexible connection between the ducts and the fan, low-frequency rumble in the air stream can be transmitted into the duct walls and down through the supports. Vibration isolators, having a deflection equal to that of the fan, are prudent for the first three points of support and half that thereafter. Detailed recommendations were discussed in Chapt. 11. Silencers placed at the roof penetration serve to counteract exterior noise breaking into the ducts from aircraft or other sources. Down shot units should be avoided unless there is sufficient room underneath to locate silencers and construct breakout enclosures.

Ductwork can transmit noise between rooms. HVAC ducts serving adjacent spaces must be routed so that they provide noise isolation equal to that of the other paths. Figure 21.18 gives an example of a duct layout for contiguous music practice rooms. The ducts here are assumed to be lined with 1" (25 mm) duct liner. In critical installations silencers also may be required between rooms.

FIGURE 21.18 Examples of Good and Bad Practice in Air Distribution Systems (Klepper, Cavanaugh, and Marshall, 1980)



Poorly Planned Duct System for Music Rooms



Well Planned Duct System for Music Rooms

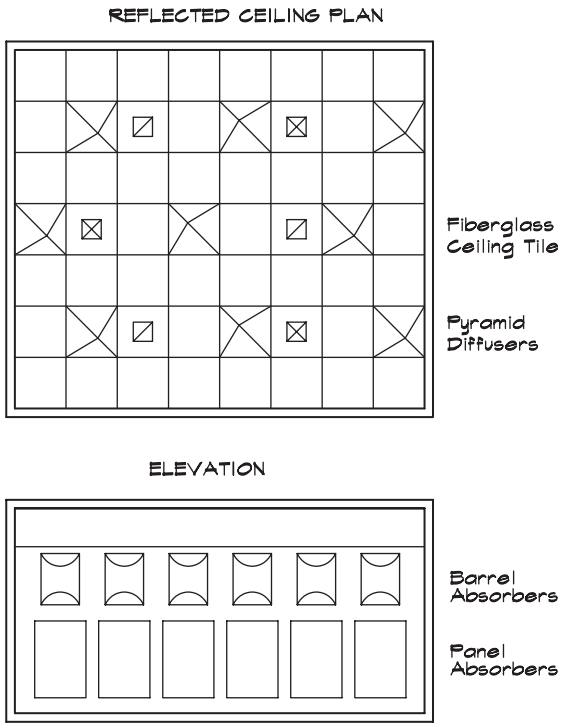
21.3 ROOMS FOR LISTENING

Music Practice Rooms

Music practice rooms range in size from single occupant spaces to rooms designed to accommodate a full orchestra. The use of a practice room differs from that of a performance space in that the sound needs to be heard only by the players and the teacher or conductor. The design objective in a practice room is not to mimic the acoustical characteristics of a performance space, but to allow musicians to play comfortably at their normal level. To the extent that a generalization can be made, the primary concern in these spaces is excessive loudness. This is certainly the most prevalent problem in band and choral practice rooms. In individual practice rooms, which can be quite small, coloration and flutter are also important factors.

Loudness is controlled by increasing the room volume and by adding absorption, on the ceiling. Room volumes of 550 to 700 cu ft (15.6 to 21.8 cu m) per instrumentalist and 350 to 500 cu ft (10.9 to 14.2 cu m) per vocalist are appropriate with ceiling heights of 18’ to 22’ (5.5 m–6.7 m) for instrumental and 16’ to 20’ (5 m to 6 m) choral spaces (Wenger, 1999). Ceilings should be 1” thick fiberglass acoustical tile. Prefabricated diffusers or inverted convex plastic skylights can be helpful, but they should cover less than about 20% of the ceiling. Wall absorption is beneficial in band rooms to address both flutter and reverberation. Barrel absorbers placed high on the walls are good for controlling excessive low-frequency reverberation. Floors are by and large hard surfaces, such as vinyl on concrete. An example of a high school practice room is shown in Fig. 21.19.

FIGURE 21.19 Design of a Band Practice Room



Small practice rooms can be designed with angled walls to help with flutter. Absorptive panels placed on contiguous walls are also helpful since repeated reflections can occur in a horizontal plane even if the bounding walls are not parallel. Where there is a hard angled ceiling it may not always eliminate flutter, so local surface-applied diffusers are appropriate. Absorptive panels are most effective when placed near the center of a wall since particle velocities are higher here than in the corners.

Listening Rooms

Listening rooms can be quite elaborate, sometimes rivaling a professional mastering or screening facility. These rooms tend to fall into three categories: pure listening rooms with stereo loudspeakers, video rooms with small monitor loudspeakers, and home theaters with surrounds.

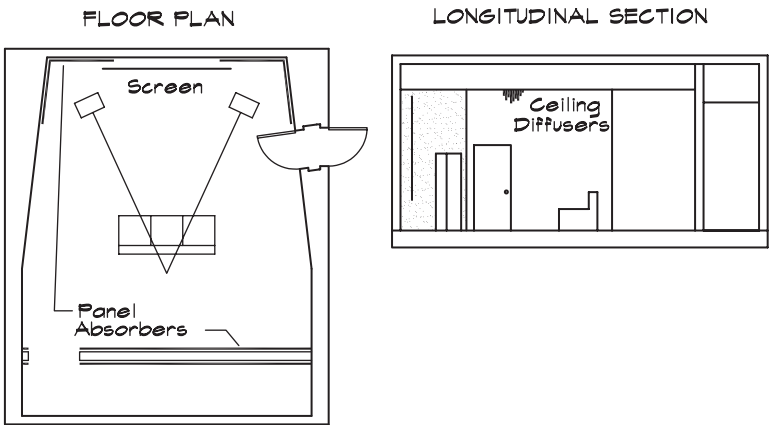
Listening rooms are often very personal and their design can revolve around the type of loudspeaker and the individual preferences of the user. The stereo format demands axial symmetry with the listener located on-axis. The so-called sweet spot, where the stereo soundstage is maintained, may be limited to a very few listeners. Traditional stereo loudspeaker placement starts with an included angle of a little more than 60°. The loudspeakers are placed about 10 to 12 feet (3 to 3.7 m) apart and the listener is slightly closer to the loudspeakers than their separation distance. The loudspeaker aim point is about 2 feet (0.6 m) behind the listener. There is some flexibility in loudspeaker positioning and manufacturers can recommend included angles as high as 100°. The high-frequency loudspeaker elements should be located in a single horizontal plane, preferably that of the listener's ears. Small differences in separation distance and toe-in angle can yield startling improvements in clarity so experimentation is encouraged.

To allow freedom of movement, the room must be designed with ample space around the loudspeakers. Accordingly the width of a dedicated listening room should be at least 16 to 18 feet (4.9 to 5.5 m) wide, leaving 3 to 4 feet (0.9 to 1.2 m) minimum between the loudspeaker and the side wall. At least one manufacturer (Dunlavy, 2001) recommends placing loudspeakers along the long wall; however, it has been the author's experience that placement along the short wall yields less low-frequency coloration and a more realistic bass response. The room depth should be on the order of 19 to 22 feet (5.8 to 6.7 m) with additional space so that some low-frequency absorption can be built into the rear walls. Ceilings should be high enough, perhaps 10 feet (3 m), that an absorbing or diffusing element can be placed at the specular reflection point without being visually oppressive.

Absorptive treatment can also be applied to the walls. If the loudspeakers are electrostatic, absorptive panels are necessary on the walls behind the loudspeakers. Some absorption, including low-frequency treatment, is recommended behind the listener and it should be in the same horizontal plane as the listener. Toole (1998) recommends some, but not continuous, diffusion on the back wall.

Where a listening room includes a video projection system in combination with the stereo audio system, the screen can be located on the front wall between and behind the loudspeakers. The two need not be coplanar. The video projector may be mounted in the ceiling, in a separate projection room, or in a wall or floor-mounted cabinet. The same design principles apply to video rooms but they are not as critical as stereo listening rooms since the quality of the soundstage is not as good as with stereo music. An example of a home listening room design is given in Fig. 21.20. This was based on a stereo audio system. The loudspeakers were electrostatic so absorption on the front walls was particularly important.

FIGURE 21.20 An Example of a Home Listening System



The storage room at the rear also helped provide bass absorption since the wall was covered with fiberglass panels over a diaphragmatic absorber.

Home theaters have become an entire industry, with publications dedicated to their design and construction. They mirror commercial theaters in that they support the 5.1 audio format as well as standard stereo format. The three front loudspeakers are best located behind a perforated projection screen with the surrounds mounted on the side walls slightly behind the listener position. If the display system is not transondent, the loudspeakers may be placed on either side and beneath the video surface. Ideally the high-frequency transducers should lie in the same horizontal plane. There is some flexibility in the placement of surrounds. When rooms can accommodate several occupants, surround loudspeakers are arranged in pairs along the side walls.

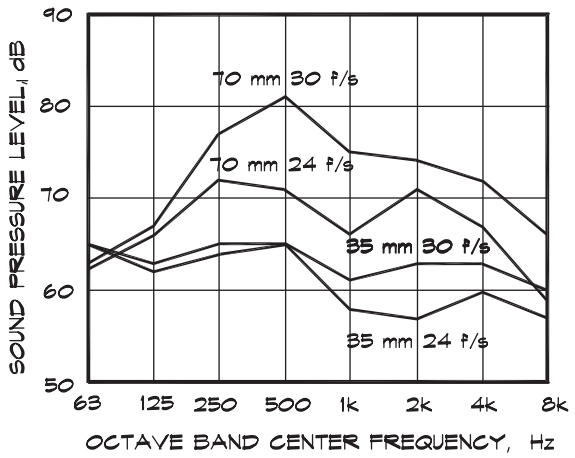
The left-center-right audio format is not as dependent on seating position to maintain the spatial relationship between the listener and the projection screen as the stereo format. Thus the area where a reasonable soundstage is maintained is larger than with simple stereo. Consequently there is a range of good seating locations. Since the seating area is larger, there are multiple specular reflection points on the side walls and a need for more absorptive material area. Home theaters, where seating capacities may vary from four to twenty, have absorptive side and rear walls that may be broken up with reflective vertical strips such as columns, lighting sconces, or other diffusive elements.

Bass buildup and rear-wall reflections should be controlled in home theaters just as it was in pure listening rooms. If loudspeakers are located behind a front-wall projection screen, the cavity can provide a space for bass absorption. Since home theaters are often located in living rooms, rather than in a dedicated space, bass trapping is more difficult and may depend on heavily upholstered furniture, thin glazing, HVAC ducts, and cabinetry.

Screening Rooms

Screening rooms are like home theaters but larger, and have the capability of film as well as video projection. Their capacity varies but falls into the 10 to 100 seat range. Since film projectors make noise, they are located in a separate booth, designed to isolate them from the theater. The level of projector noise depends on the film speed and width. Measured levels from film projectors are given in Fig. 21.21. Data are shown for both 35 and 70 mm projectors

FIGURE 21.21 Film Projector Noise at Two Feet



at 24 and 30 frames per second. Noise isolation design is straightforward. For 35 mm 24 f/s projectors the projection port can be double glazed 1/4" lam + 3" AS + 1/4" with at least one pane tilted away from vertical, not so much for sound isolation, but to prevent multiple reflected images on the screen. Laminated glass can be used in a projection port without degrading the image quality.

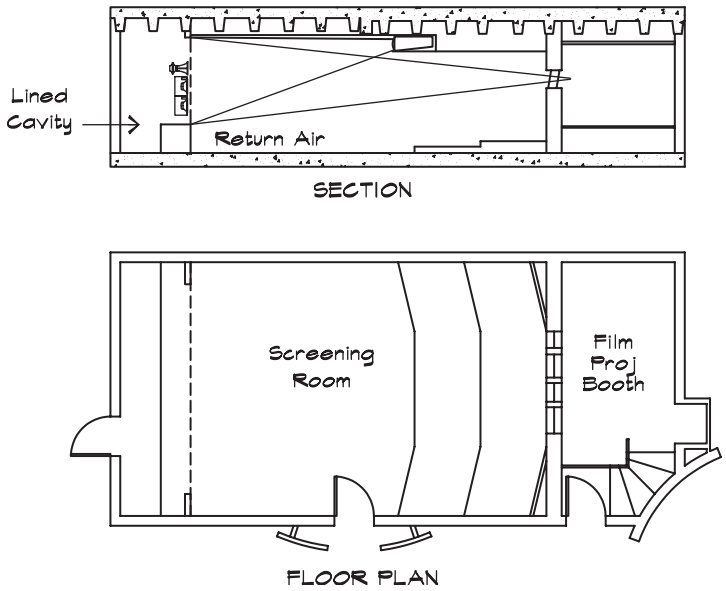
Loudspeakers located behind the projection screen can be mounted in a baffle wall or be freely suspended. Horns should be placed close to the screen to minimize back reflection and increase coupling. When a baffle wall is used, it should be at least double 5/8" (16 mm) drywall with 2" black fiberglass board on the screen side and R-11 batt in the cavity behind it. Where the loudspeakers penetrate the baffle wall, the openings must be sealed airtight so that the cavity does not become a mid-frequency Helmholtz resonator, which can color the sound.

The baffle wall cavity is a convenient location for return air HVAC ducts. If the return air grille is positioned below the screen across the front of the room it can be fashioned into a low-frequency Helmholtz resonator bass trap. An example of a professional screening room, designed for a video effects company, is given in Fig. 21.22. It can accommodate both film and video with a somewhat unusual dual three-beam video projector configuration mounted in the ceiling. In this project the surround loudspeakers were independently supported on steel columns located within the side walls to reduce vibrational transmission through the wall structure. The walls of the screening room were treated with 1" (25 mm) and the rear wall with 2" (52 mm) fiberglass panels. The cavity under the seating risers was filled with batt insulation and the front surfaces of the risers were left open, protected by a return air diffuser grille. The air was returned beneath the screen and the supply ducts were built into ceiling soffits on either side.

Video Post Production

Video post production facilities have rather modest acoustical requirements. Video post production takes place in a small office environment. Most of the audio work is done by monitoring on a pair of near-field loudspeakers placed on a desk, wall, or on stands a few feet from the listener. Soundstage and control of early reflections are not of particular concern.

FIGURE 21.22 Asylum Visual Effects Screening Room, Santa Monica, CA, USA (Acoustician: Marshall Long Acoustics) (Architect: Walter Meyer Associates)



Work spaces tend to be small, with acoustical tile ceilings and absorptive walls. Floors are not carpeted to allow rolling chairs to move about freely.

21.4 ROOMS FOR RECORDING

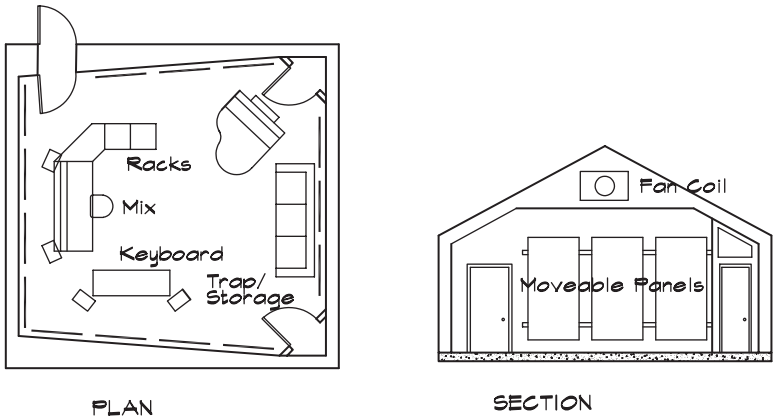
Home Studios

Home studios, sometimes known as project studios, are increasingly common as high quality recording equipment becomes smaller and more affordable. The sophistication of this electronic gear has a direct influence on the proliferation of small studios since excellent recordings can now be made in a low-cost environment. The result has been a reaction from commercial studios trying to limit their number through land-use regulations. Under pressure from the commercial studios, cities such as Los Angeles have prohibited people from using their homes to make commercial recordings. Thus the home studio builder should check local ordinances and enforcement before beginning.

The second problem confronting a home studio user is that regulations in residential neighborhoods restrict noise levels at neighboring properties. These property-line ordinances typically limit nighttime noise levels to 45 dBA or 5 dB over the existing ambient, whichever is higher, within residential properties. Property line ordinances can limit the level at which a musician can play or require substantial construction to meet the local codes. A review of the local noise ordinance is therefore prudent.

Picking a location that provides natural sound isolation is a good start. If a basement is available, it is probably the best choice; or a separate structure can be used. Probably the most common choice, and one of the most difficult, is a garage. A two-car garage is about 24 feet square, enough for a one-room studio. Garages are lightly constructed and must be heavily reinforced to achieve adequate isolation. Exterior surfaces of plaster, brick, or

FIGURE 21.23 An Example of a Garage Studio



cement board, having a surface mass density of 10 lbs/sq ft (49 kg/sq m) or greater, can be used on the exterior with double drywall interior surfaces supported on a separate framing system or resilient isolators. In most cases the garage door must be removed, although it can be retained as an applied decorative element over the exterior wall.

A garage ceiling is too low and raising it requires added structural framing and review by a structural engineer. It should be double-drywall resiliently supported. The fan coil unit can be located above the ceiling with an access panel for service or a package unit outside the building can be employed. Careful calculations are necessary to insure isolation of the fan coil supply and return from the studio. Silencers or snaked flexible duct buried in batt insulation can help provide the necessary attenuation.

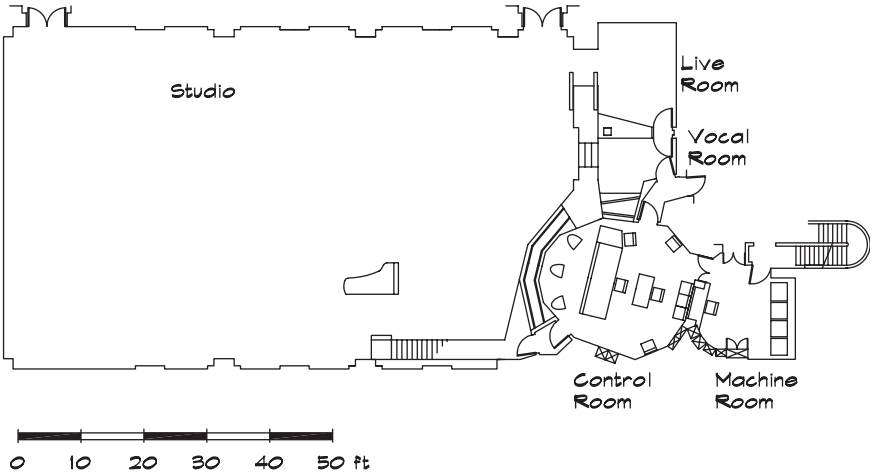
Figure 21.23 gives an example of a design for a personal studio built into a freestanding garage structure. It illustrates some of the difficulties in making a successful conversion. The layout of a successful project studio is quite personal and reflects the working habits of the user. In this example, the operator can mix and compose on a midi controller linked to a computer. A small number of musicians can be accommodated for a recording secession. Movable wall panels provide absorption and can be replaced with diffusive elements or simply removed. Storage closets double as bass traps. The floors are hardwood with throw rugs for added absorption.

Sound Stages

Sound stages are large open rooms used for indoor movie production. Acoustically they are designed to be dead with all surfaces except the floors covered with 4 to 6 inch (100 to 150 mm) deep blankets of absorptive material. The floors are smooth and flat so that cameras can be dollied. The exposed wall surfaces can be faced with commercial quilted blankets covered with hardware cloth below an elevation of 10 ft (3 m). The best rooms are built with isolated construction, floated floors, double-studded walls, and separately suspended gypboard ceilings. Access is provided via sound rated doors, which can be quite large. Some facilities have control rooms adjacent to the stage for mixing and recording.

The most difficult aspect of sound stage design is noise control. Isolation from exterior noise is a challenge because many stages are built in converted warehouses with lightweight roofs and little thought to the isolation of traffic and aircraft noise. Large air conditioning

FIGURE 21.24 Abbey Road Studio One, London, England (Abbey Road Studios, 2001)



units are required to cool the stage lighting fixtures and this equipment is often located on the roof, where it is difficult to control. It is much preferable to separately support air handlers on grade or on an elevated steel platform dedicated to that purpose. Ductwork should be isolated from the structural framework either by resilient suspension or by a separate support system. Silencers located at a roof or wall penetration provide exterior as well as equipment noise control.

Scoring Stages

Scoring stages are rooms in which the music for a film is recorded. The orchestra conductor, who is the composer, faces both the musicians and a large screen on which the film is projected. As he conducts, he may listen through a single headphone to a click track, which aids in synchronization of the film and the score. Visual cues are also projected onto the screen in the form of streamers that progress from left to right across the screen to mark the beginning of a transition or effect when they reach the right-hand side.

A scoring stage is large, almost the size of a concert hall. Like concert halls, the best ones are shoebox-shaped with high ceilings and irregularly shaped diffusers on the walls and ceiling. A very good one, Studio 1 at Abbey Road Studios in London, is shown in Fig 21.24. Its dimensions are 92.6 ft $\times$ 59.7 ft $\times$ 39.4 ft high (28.2 m $\times$ 16.1 m $\times$ 12.2 m) and its total volume of 218,000 cu ft (6172 cu m) is about one-third that of Boston Symphony Hall. On one end there is a large (44 ft or 13.4 m wide) projection screen with the control room in an opposite corner.

Scoring stages are designed much like concert halls but without the requirements for an audience. The floors are flat and the walls and ceiling surfaces feature irregular shapes for diffusion. Reverberation times can be changed using moveable curtains or panels. For film, from 5 to 8 mics are used for the right-center-left and surround signals, and another 30 to 35 mics for individual instrument groups. The high ceilings sometimes make it difficult for the musicians to hear each other so 12 to 18 foldback channels are provided from the mix board to individual players through headphones. The orchestra can be seated on risers for visual cohesion and arranged to achieve a balanced sound.

Since sound stages are smaller than concert halls the orchestra cannot play quite as loudly as they would under performance conditions. When they do, the reverberation in the room, particularly the bass, can overwhelm the direct sound and yield a muddy recording. If the balance is correct and the control room is set up properly, the recording engineer can do a live mix including surrounds if necessary, however, the recorded tracks can be remixed at a later time.

The reverberation characteristics of a scoring stage are much the same as a concert hall. Abbey Road has a mid frequency reverberation time of about 2.2 seconds, rising slightly at lower frequencies and remaining fairly constant at high frequencies. The lack of audience and seat absorption limits the falloff of the high frequencies to that due to curtains, musicians, and air absorption so these rooms can be somewhat brighter than a performance hall.

The recording of symphonic music can also be done in an empty concert hall. In these cases the room is often extensively modified to accommodate this use. For example, when Royce Hall at UCLA is used for recording, a wooden platform for the musicians is constructed over a portion of the seating area and the opera chairs in the orchestra section are covered with 3/4" (19 mm) plywood over visqueen sheets to decrease high-frequency absorption (Murphy, 2001).

Recording Studios

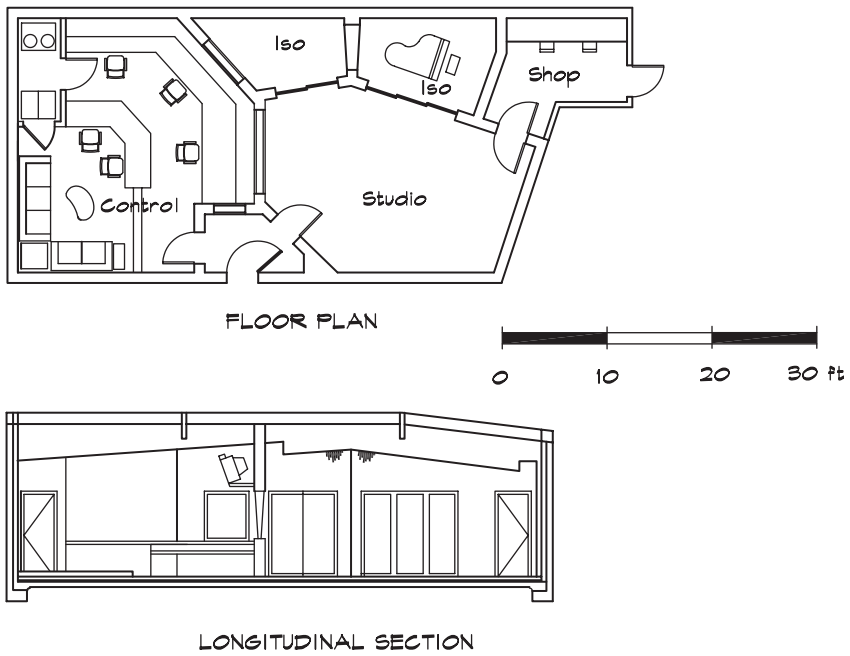
Recording studios consist of one or more rooms where music is played and recorded. The musicians may all be present at the same time or they may never see one another. With the ability to move recorded music from place to place electronically, musicians may perform in rooms a continent away and days or weeks apart. When musicians are playing simultaneously, separate rooms are desirable to isolate the instruments so that they do not bleed into other microphones. These isolation booths also can be helpful in generating different acoustic environments.

A good studio must incorporate several acoustical factors:

1. Quiet—on the order of NC 10 to 15
2. Noise isolation from other areas (including footfall)
3. Freedom from acoustical defects such as flutter
4. Adequate absorption (often variable)
5. Reasonable diffusion
6. Isolated areas for recording individual instruments
7. Visual communication between the control room and the studios
8. Control of bass reverberation and modal buildup

Studios can be generic or highly personal, based on the working preferences of an individual user. A good example of the latter is Hum Studio in Santa Monica, CA, designed for Jeff Koz, a well-known composer. Figure 21.25 shows the floor plan for the main studio and control booth. Since most of the composition work is done at a keyboard with small digital mix boards, the traditional control room layout was not used. Instead, three work stations, each with a mix board, were arrayed along the front and side walls. Each could be used simultaneously during recording and mixing sessions. The main composing station was designed around the users' equipment. Since listening is done via small near-field loudspeakers, there was no need for large stereo monitors and no need for a large loudspeaker

FIGURE 21.25 Hum Studio A, Santa Monica, CA, USA (Acoustician: Marshall Long Acoustics) (Architect: Walter Meyer Associates)



bridge above the main window. This arrangement freed up the center of the room for a client couch and social area instead of being dominated by a massive mixing console.

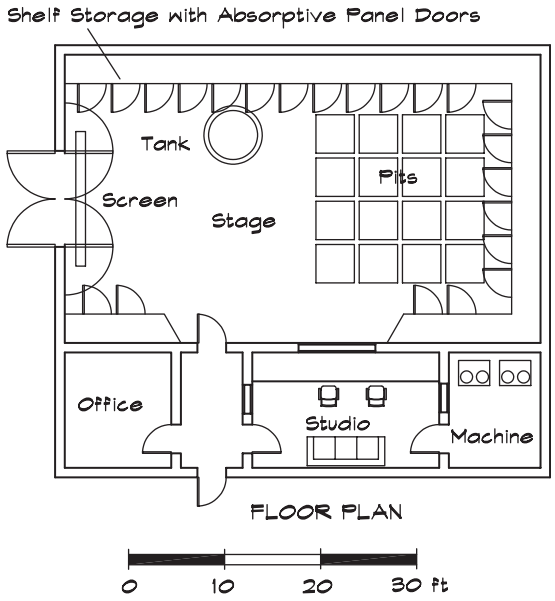
The studio consists of three rooms accessed from a small foyer separating the studio from the control room. Foyers can sometimes be used as isolation rooms particularly if there is a need for feedback such as with an electric guitar. Two isolation booths, with sliding glass doors, are available for individual instruments such as a piano or vocals. The walls and ceiling are constructed of multiple layers of drywall with a wood panel finish on the ceiling. Quilted absorbers are hung from hooks on the walls and can be removed or folded to reduce their area. The mid-frequency reverberation time is about 1.2 sec and flat with frequency. Bass trapping is done using the return-air plenum built above the ceiling. The segmented ceiling requires surface-applied wood diffusers to control flutter echo.

The control room is designed to be much deader, about 0.5 sec at mid-frequencies. The walls are faced with 2" (52 mm) cloth-wrapped fiberglass panels. The ceiling is hard—two layers of 5/8" drywall hung from springs. Bass traps are built into the space above the equipment closet and into the video monitor enclosure. Windows are arranged so that there is visual contact between the control room and any point in the studio, including the isolation booths.

Foley and ADR

Foley stages, where sound effects are generated by physical manipulation of devices, are often indistinguishable from landfills, due to the general clutter. A typical Foley stage consists of a dead room with walls and ceiling covered in broadband absorption and a hard-surface floor

FIGURE 21.26 Foley Stage



having multiple pits each 3 to 4 feet square, in which different walking surface materials are contained. The Foley artists watch the film, projected on a screen against one wall, while making the sound effects with their hands and feet and an assortment of mechanical gadgets. For example, if the film requires the sound of running along a sidewalk, the artist runs in place on a concrete slab in time with the film actor's steps, with a microphone suspended nearby. Gravel, wood, or sand may each have a separate pit. Water effects are created in a basin or large trough.

Since space is expensive, Foley pits sometimes are built into a traditional studio. This is less desirable than a dedicated space since recording studios are more reverberant than Foley stages and water is seldom available. Foley is messy and dirty and requires space around the pits for microphones and props. One approach is to build prop storage areas on the walls with absorptive panels mounted as doors. The airspace behind the panels improves bass absorption and the props are close by. Figure 21.26 gives an example of a Foley stage design based on this concept.

ADR or automatic (sometimes automated) dialog replacement is a technique using voice over, or the recording of dialog after the film has been shot. Whenever possible, film makers like to use the original sound recorded during filming but background noise or technical problems can make this impossible. In ADR the actors rerecord their parts in sync with the film. ADR stages are small, sometimes no bigger than a bathroom, and relatively dead. Low-frequency reverberation is a concern. Most have at least 2" (52 mm) thick panels on the walls. Since dialog replacement includes singing, ADR artists prefer rooms that are not completely dead (Farmer, 2001) and have a bit of volume, on the order of 8' × 12' × 9' (2.4 m × 3.7 m × 2.7 m). Larger stages can be used for recording voice over by adding absorptive panels around the actors. Flutter echo is particularly important to control so at least two nonparallel wall surfaces require treatment. ADR rooms should have a flat reverberation time versus frequency characteristic. Diffusion can be helpful and throw rugs are used to vary the room characteristics.

With animated films the dialog is recorded first and the animation created later to fit the sound. With film, the actors must watch the film and synchronize their voices to the picture. Video monitors are built into voice-over booths for this purpose. A communication system including a window between the booth and the studio and an intercom is a necessary part of the design.

21.5 ROOMS FOR MIXING

Dubbing Stages

Dubbing stages are large screening rooms, about the size of a commercial movie theater, with a full-sized projection screen and a single floor of opera chair seating. In the center of the seating is an area for three sound mixing consoles, one each for dialog, music, and effects, arrayed in a line across the theater. The film is projected on the screen using a projector fed by dubbers, which historically were mechanical transports that could be played forward or backward in synchronization with the picture. All the film sounds are combined into the final five channels that are ultimately transferred onto a CD and onto the film as a backup. Early films such as Fantasia used multiple film dubbers as storage devices for the audio tracks, which led to the current use of the word dub, meaning the replacement or recording of audio onto a film.

A dubbing stage should sound like a very good movie theater since that is where the final product will be displayed. The reverberation times given in Fig. 17.10 can be used as a guide. Side and rear walls are absorptive, occasionally broken up with reflective surfaces. The ceiling is black acoustical tile. Opera chairs are padded and the aisles are carpeted. The floor behind the mixing consoles is wood or vinyl to allow the chairs to roll easily. A series of technical racks are built into a long pedestal behind the mixers to house outboard processing gear. Producers and other executives frequently sit behind this console.

Control Rooms

Control rooms house the recording and electronic processing equipment necessary for mixing sound. The main component is the mixing console, the switchyard where the audio tracks are assembled onto a master recording device. The control room functions as a working space for mixing, as well as the social center of the studio, and the main client presentation area. Thus it must perform several different functions.

In its capacity as a stereo or multichannel mixing room the control room should incorporate a number of design features:

1. Quiet—on the order of NC 15 to 20
2. High noise isolation from the studio and machine rooms
3. Axial symmetry about the longitudinal centerline
4. Adequate absorption and reverberation control
5. Loudspeaker placement in accordance with the formats to be mixed
6. Deflection of first reflected sound away from the mix position
7. Control of bass via diaphragmatic absorbers or Helmholtz resonator trapping and loudspeaker placement
8. Convenient access to wiring and equipment racks for maintenance and setup of electronics

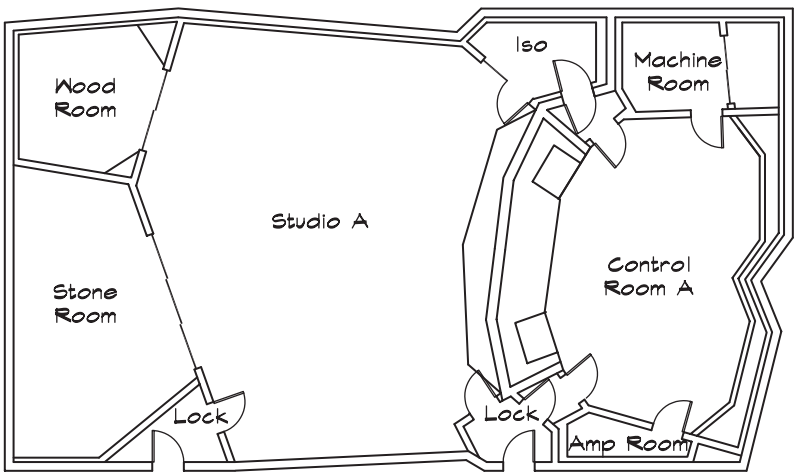
9. Good visual communication with the studio spaces
10. Some side wall and/or rear wall diffusion
11. Video monitor for coordination of audio with video playback
12. The ability to route audio and video between studios and control rooms and central storage areas
13. Storage media that do not generate noise or undue thermal loads

As a client presentation and social center it should also have:

1. Comfortable seating and good viewing angles for clients and visitors
2. Speech isolation from the main studio
3. Electronics interfaces including Internet or other high speed computer connections
4. A flat workspace such as a table or desk
5. Separate telephone room(s) for private conversations.

Figure 21.27 shows Studio A at Paisley Park Studio in Chanhassan, Minnesota. It consists of a series of recording rooms and a main control room fitted into a rectangular area 75 × 44 feet (22.9 × 13.4 m). It has a traditional design with the mixing console facing the window into the studio, flanked by stereo loudspeakers. One unique feature is the separating wall, which was cast as a monolithic concrete beam to provide structural support as well as isolation. Separate machine rooms for tape recorders and amplifiers are located on either side of the mix position. The rear wall includes a convex v-shape to direct first reflected sound away from the console with a projecting soffit above. Recently the design trend has been to lower the center window to accommodate a middle loudspeaker for mixing film or to rotate the control room so that the window is on the side. The best control rooms are relatively large, at least 20 to 25 feet (6.1 × 7.6 m) in each direction. Some designers prefer a room that is wider than it is deep as is the case at Paisley Park. With the introduction of digital consoles the space requirement for the mix board has been reduced, but most still retain at least 48 active faders. Tape machines are being replaced by digital storage units, which require less

FIGURE 21.27 Paisley Park Studio, Chanhassan, MN, USA (Acoustician: Marshall Long Acoustics) (Architect: Boto Design Associates)



space but more cooling. The cooling fans for digital storage are rack mounted and generate enough noise to require location in a remote room behind double sets of heavy glazed doors.

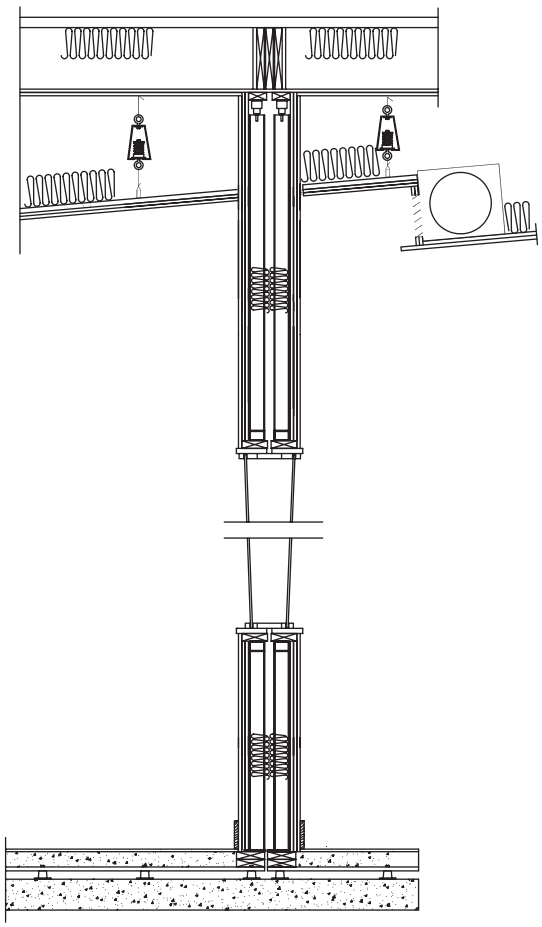
21.6 DESIGN DETAILS IN STUDIOS

Noise Isolation

For the best noise isolation, studios and control rooms are built as room-within-a-room structures, floated on an isolated concrete slab, with separately supported walls and ceiling. Floor slabs should be point mounted on neoprene isolators having a deflection of at least 0.25 to 0.4 in (6 to 10 mm). Continuous floor support systems using cork, neoprene, mineral fiber, or fiberglass sheets are much less effective since the floor load is distributed and the deflections are small.

Walls and ceilings are double (or triple) constructed on separate supports with the interior walls standing on the floated slab and the top of the studs stabilized with isolating sway braces. Ceilings are hung from spring isolators to complete the box. Penetrations of the ceiling cavity for return air openings can utilize lined sections of duct to close off the hole. Recall that there is considerable low-frequency loss due to the end effect at the opening. Figure 21.28 gives an example from the Hum Studio main control room.

FIGURE 21.28 Isolation Details from Hum Studio



Symmetry

Control rooms should be symmetric about the line through the mixer position to retain stereo imaging. The side walls and ceiling are flared out to direct the first reflection around the mix position. At one time ceilings were angled toward the mix position to "compress" the bass, in part to try to offset the null of the fundamental mode in the center of the room. This design proved difficult to control since only some frequencies had their pressure maxima there and the effect was highly dependent on the receiver position. Now most ceilings are either flat or flare away from the mix position. They can be treated with absorption or diffusion near the specular reflection points.

Loudspeaker Placement

The traditional placement of stereo loudspeakers in a control room is at an included angle of around 60° at a distance of 11 to 12 ft. (3.4 to 3.7 m) from the mixer. Loudspeakers are aimed at a point about 2 ft (0.6 m) behind the mixer's head to give a wider sweet spot. They can be built into the front wall of the studio with either a baffle wall or just a simple enclosure or can be left free standing. If a baffle wall is used it should be wrapped around the loudspeaker using a heavy material, drywall but sometimes concrete, to minimize back transmission into the studio. In some cases, free-standing loudspeakers are placed on pedestals in front of the mixing console.

In surround systems three loudspeakers are located in front, two at or slightly outside the normal stereo position. It is important that these be in the same horizontal plane, with the loudspeakers oriented so that the high-frequency drivers are closest to the horizontal plane of the mixer.

Separate moveable subwoofers can be used for bass control. They are placed on the floor, against a wall near the null point of one of the fundamental modes. Some mixers prefer full-range loudspeakers to help control room modes, particularly when they also are used for the surrounds.

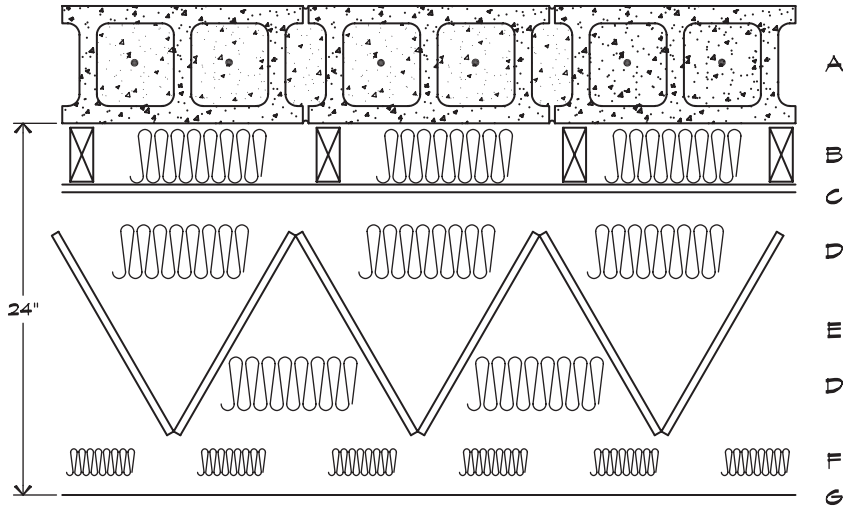
The location of surrounds is less critical than the LCR loudspeakers although they should be positioned to the side and slightly behind the mixer. The ITU-R BS775 standard calls for the placement of the front left and right loudspeakers from 2 to 4 m (6.5 to 13 ft) apart at an included angle of 60°. The rear surrounds are to be located 110° to 120° off the centerline although some mixers prefer angles as high as 125° (Mitchell, 2002). When they are located directly to the side, the directional localization is very poor and the envelopment is less effective.

Bass Control

Bass control is essential in studio and control room design. In control rooms the mixer must be confident that what he is hearing is on the audio track and not a construct of the room itself. Otherwise he cannot be assured of the accuracy of the recording and its repeatability in other rooms. In recording studios excessive bass can artificially lengthen the reverberation time in certain frequencies, causing reverberant bleed from track to track, coloration, and lack of clarity in the natural sound of the instruments.

Three physical methods are employed to achieve low-frequency absorption: 1) deep resistive materials such as fiberglass boards or batts, 2) diaphragmatic absorbers such as lightweight drywall or plywood panels, and 3) Helmholtz resonator cavities containing fiberglass batts for damping. Each of these has been discussed previously. Of the three,

FIGURE 21.29 Highly Absorptive Rear Wall



A Structural wall
B 3.5" (90 mm) Fiberglass batt insulation
C 1/2" (13 mm) Sound deadening board on 2x4
D 3 1/2" (90 mm) Fiberglass batt insulation
E 1/2" (13 mm) Pressed fiberglass sheets (3 lb/cu ft)
F 1" (25 mm) Quilt batting
G Stretched cloth surface

a deep layer of batting gives the best broadband response, but takes up considerable space. Figure 21.29 shows an example of a deep rear-wall absorber.

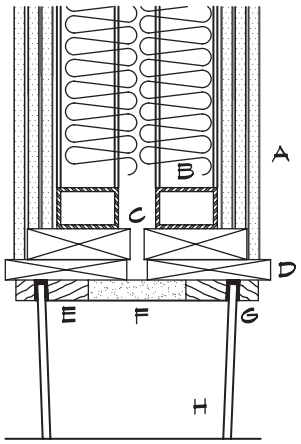
Helmholtz resonators are frequently used, but often misunderstood, bass absorbers. Their openings should be positioned at a pressure maximum such as along a surface edge or corner. Distributed openings are also effective and can be combined with a return air plenum by installing a short run of lined sheet-metal duct to act as the resonator neck. Resonator openings can be located behind porous absorbers since the fiberglass boards do not impede the flow of low-frequency energy. Cavities should be distributed around the room and filled with batt insulation to widen their bandwidth.

Studio Window Design

Large windows separating the control room from the studio are designed to provide clear visual communication between rooms while maintaining acoustical isolation. The degree of separation required is not as great as one might think since it is rare that a different musical track is being played in the control room, while mics are live in the studio. When live monitoring takes place, it is of the same material as that being recorded, so isolation is not as critical. Low-frequency separation helps keep the signal-to-noise ratio of the recorded material high. Voice privacy between the control room and the studio is critical.

An example of window design is given in Fig. 21.30. It is double-glazed construction, preferred because of its superior low-frequency performance. The edges of the cavity between the panes are lined with open cell neoprene foam or fiberglass boards, wrapped in cloth, to

FIGURE 21.30 Studio Window Detail



A Separately supported isolation wall
B Structural support frame with 2 x 3 1/2" (90 mm) batt insulation
C Open slot between 2 x nailers
D Finish wood window frame
E 1" (25 mm) Thick wood stops
F 1" (25 mm) Open cell foam or cloth wrapped fiberglass board
G 3/16" (5 mm) Rubber gasketing
H Heavy window glazing - 1/2' + 3/8" thick

provide edge absorption. The cavity between the walls can be used as a Helmholtz resonator tuned to the window mass-air-mass resonance. Each pane should be supported separately on its own stud system and isolated floor slab. Window panes are angled, not so much for acoustic reasons, but to reduce light reflection and to provide space for the monitor enclosures above.

If structural supports are necessary, they can be included via pipe columns or a bearing wall between the two studio walls. Triple-panel construction is better for high frequencies but can be worse for low frequencies. Where a triple panel is used, a high-mass bearing wall such as grouted concrete block along with heavy glass in the center can offset some of the disadvantages and provides insurance against transmission through openings. If triple-panel construction is employed there must be adequate air gaps to reduce the lower limiting frequency.

The openings for the monitors should not degrade the wall performance or color the monitor performance. It is better to extend the separation wall up past the monitors to the resilient ceiling closure rather than to try to build an enclosure around the rear of the monitors.

Diffusion

The degree to which diffusion contributes to the sound quality in studios and control rooms is a subject of some debate. In scoring stages where stereo mics are positioned to capture the overall sound of an orchestra, diffusion is important since it enhances the orchestral blend. Studio 1 at Abbey Road, for example, uses diffusive shapes on most of the wall surfaces. When separate mics are used to isolate instruments, diffusion is less critical since it contributes to reverberant bleed, although it is still helpful for breaking up flutter echoes.

In control rooms some diffusion on the side and rear walls can help with the sense of envelopment and a widening of the stereo image.

There is at present no accepted method of quantifying the proper amount of diffusion in a simple way as there is with absorption. Davis and Davis (1979) suggest that control rooms be designed with a highly diffuse rear wall (live end) and absorptive surfaces around the monitors (dead end). Toole (1990) counters that a totally diffuse rear wall creates a fog of confusing reflections and that a smaller center-section of absorption in the rear wall is preferred. The advent of surround loudspeakers, particularly when they are as large and powerful as the front ones, means that rooms have to be designed with evenly distributed absorptive and diffusive elements.

22 ACOUSTIC MODELING, RAY TRACING, AND AURALIZATION

It is highly desirable to ascertain the acoustical characteristics of a performance space before time, treasure, and emotion are expended on its construction. In spite of this fact, the ability to make even the most rudimentary predictions a priori about the acoustical behavior of spaces dates only from the work of Sabine in the early twentieth century. Before this time concert halls were designed and built empirically based on previous examples or, as in the case of Garnier, based on pure chance.

In spite of the lack of technical guidelines, some of the finest halls were constructed before the study of room acoustics was subjected to the rigors of scientific analysis. Musikvereinssaal in Austria, a hall that still ranks among the best in the world, dates from the eighteenth century. Boston Symphony Hall, which benefited from Sabine's reverberation experiments, was designed without model testing or advanced analytical techniques.

The development of small computers in the late twentieth century gave us the ability to create mathematical models of acoustical spaces and to reproduce the sound of a room before it is constructed. We must be careful, however, that we do not fool ourselves into believing that these models are perfect representations of reality. The simplifications necessary to be able to carry out the calculations in a reasonable time still leave us with an imperfect picture, but as technical sophistication and computing ability increase, the models are improving.

22.1 ACOUSTIC MODELING

Early twentieth century investigators built physical scale models to test acoustical design ideas. Several techniques were employed to construct the interior of a room and generate the sound signal. One simple method was to use a two-dimensional wave table consisting of a flat, water-filled trough into which the plan or section of a hall could be inserted. This technique has been known at least as far back as the mid nineteenth century. Davis (1926) reviewed its history and described the work that he and Kaye did at the National Physical Laboratory. Water ripples were generated using a small oscillating paddle. The interaction with the interior room surfaces was studied by observing wave patterns that formed in the tank. A light could be projected through the bottom of the tank and onto

a screen for observation. The technique was limited to long wavelengths by the speed of the driving mechanism.

W. C. Sabine (1964) adapted spark photography to study the propagation of waves in the design of several theaters on which he consulted after the construction of Boston Symphony Hall. In 1928 Knudsen and Delsasso (Knudsen, 1932) used a similar but improved technique to investigate two-dimensional plan and section models of Royce Hall at UCLA.

Vermeulen and DeBoer (1936) developed a simple method of optical model testing used by Delsasso in his work on Schoenberg Hall at UCLA. To do a reflection study, small light bulbs partially covered with tape to create directivity were used as sources. Model surfaces were constructed from thin brass sheets bent into the shapes of the walls or ceiling and then chrome-plated to act as mirrors. Opaque sheets of glass were used as receivers to study the distribution of reflections by observing the brightness patterns formed by the scattered light (Delsasso, 1967). Walls and ceilings could be shaped to yield an even intensity of reflected light, and reflection patterns from individual surfaces could be studied using this technique. Light models are a high-frequency, short-wavelength system, which may not accurately portray low-frequency phenomena such as refraction. This method has its primary application in the study of early reflections.

Testing Scale Models

Spandock (1934) reported using 1:5 scale models of rooms for acoustical testing. He ensonified them at 2500 and 4000 Hz, which corresponds to 500 and 800 Hz in the full-size rooms. He and his followers later extended their work to 1:10 scale models using tape recordings of speech, played back at ten times the recording speed (Knudsen, 1970) in the model space. A small receiver microphone was used to collect sounds in the model and convert them back to the original frequency range.

In scale modeling, material properties must also be scaled. Yamamoto and Wakuri (1968) used 1 mm (0.04") thick Plexiglas plates, backed with 10 to 15 mm (3/8–5/8") air spaces, for reflective surfaces in their models. For absorption they used 2.9 mm (0.11") thick polyurethane foam sheets or 25 mm (1") thick mineral wool boards. In some cases, the polyurethane was 50% covered with 18 mm (3/4") wide strips of 0.07 mm (0.003") thick cellophane tape. In other instances the absorptive sheets were faced with suitably dimensioned spaced wood slats.

Air absorption in scale model testing presents another interesting problem. Since tests are conducted at frequencies as high as 100,000 Hz, its effects can be substantial. Experimenters (Yamamoto and Wakuri, 1968) use very low-humidity air (1.5% relative humidity) furnished by drying equipment. Other researchers use pure nitrogen, which has lower high-frequency absorption properties than air. Because most of the interest in building models is to check on the pattern of early reflections, air absorption is often less important than one would think. A 1:10 model of a two-second hall has a scaled reverberation time of 200 ms and the critical early reflections occur in about a tenth of that time. Thus the first 20 feet of sound travel in model space is of the most interest (Grozier, 2002).

Spark Testing

Model testing requires both small sources and small receivers. In a 1:10 model an eighth-inch microphone is small enough. At 1:20 it is equivalent to a 2.5" (64 mm) diameter mic capsule, which would be a problem above 1000 Hz. For impulse testing, a spark generator is used to

produce a small, short duration sound. The time history of a received signal can be plotted or used to determine properties of materials or the impulse response of a space. Figure 22.1 shows an experimental setup to measure the absorption coefficient of a material.

A perfectly reflecting surface is used as a standard of comparison and the direct sound level is the reference calibration. By matching the direct-field levels and comparing two graphs, the difference in the reflected levels will give the specular absorption coefficient for a given angle. Figure 22.2 illustrates a result obtained from a single reflection.

A difference in level will occur even for a perfectly reflecting surface due to the path length difference. When there is more than one surface, multiple reflections can occur. Figure 22.3 shows the effect of reflections from a floor and a wall. Multiple surfaces produce more complicated patterns that mimic the behavior of a real room.

Spark testing can also be used to measure the diffusive properties of a material. Small-scale aggregate such as that used for model railroad ballast can be scattered over a flat reflective sheet. For a diffuse material the specular reflection amplitude is much reduced but lateral reflections from reflectors on either side of the specular path scatter sound toward the microphone, albeit at a later time. Figure 22.4 illustrates the effect.

FIGURE 22.1 Setup for Measuring Sound Absorption of a Flat Sheet (Grozier, 2002)

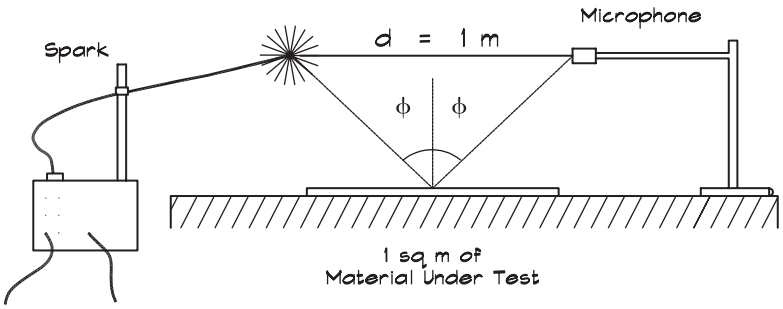


FIGURE 22.2 Reflections with the Source and Microphone over a Hard Floor (Grozier, 2002)

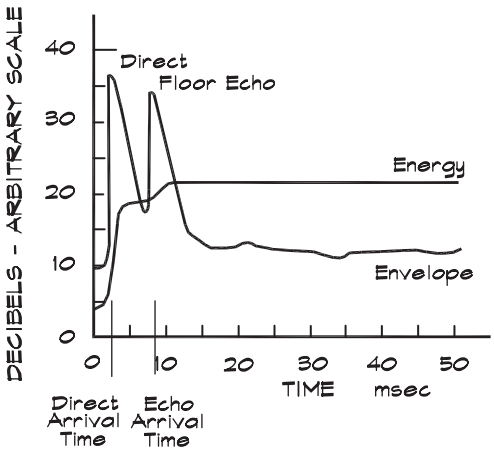


FIGURE 22.3 End Wall Added to Floor (Grozier, 2002)

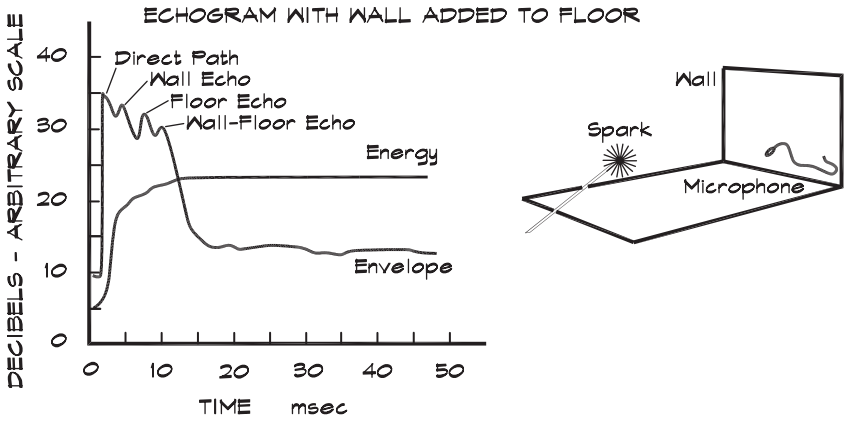
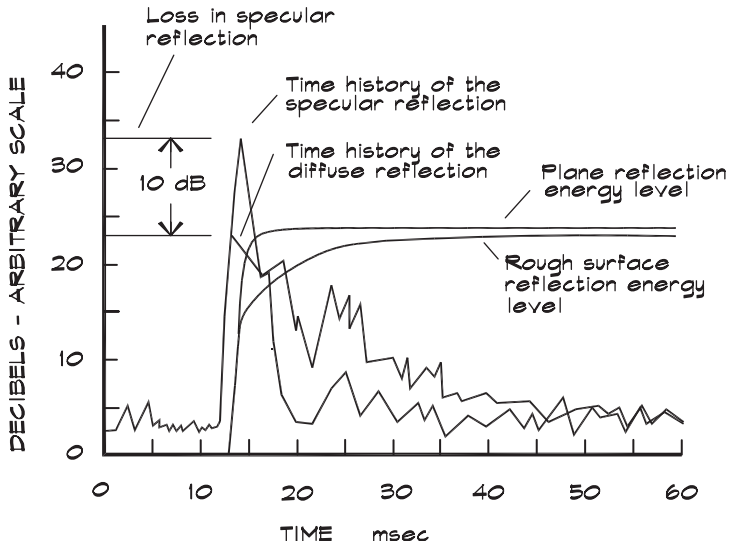


FIGURE 22.4 Comparison between Reflections from Smooth and Rough Surfaces (Grozier, 2002)

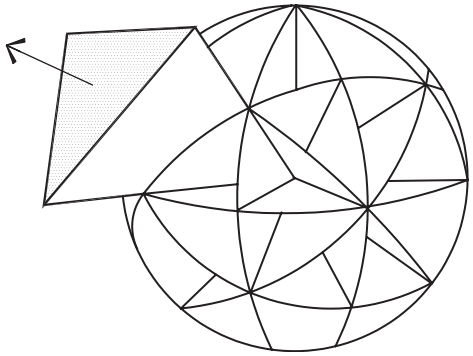


The art of model testing is unfortunately becoming lost as computer programs become more accurate. Three-dimensional physical models still serve as an important reality check on mathematical modeling. They are relatively inexpensive to build and test since they need not be as detailed as a visual model.

Ray Casting

Computers have become sufficiently powerful that mathematical modeling of three-dimensional spaces is now practical. The normal methodology is to create a surface model of an enclosed space using polygons or other easily represented surfaces. A mesh of contiguous polygons is formed and each is described by a list of vertices, connected by edges. A surface is characterized by its edges, vertices, a normal, and material properties such as absorption and diffusion coefficients.

FIGURE 22.5 Subdivision of the Source's Surface into Triangular Beams (Farina, 2002)



Rays are cast mathematically from a source and have retained properties including the original source strength and location, directivity, direction, and total distance from the source, amplitude, and bounds. Since a finite number of rays is cast, each ray actually represents a cone or pyramid-shaped beam of energy, radiating out from the source. The geometry is given in Fig. 22.5. Ray directions can be selected by using a uniform, random, or directed distribution, however a uniform pattern in which the center of each beam surface is chosen as an initial aim point is most common.

The question of how many rays to cast then arises. It is important that each room boundary be found by some ray, and that ray-surface strikes are not double counted. A ray is likely to find a surface having area S in a time t if the area of the ray wavefront is not greater than S/2. This leads to a minimum number of rays (Rindel, 2000) necessary to find a surface

$$N \geq \frac{8\pi c^2}{S} t^2 \tag{22.1}$$

For a hypothetical surface having a one square meter area and a propagation time of one second, the required number of rays is about 3×10^6. Thus, brute force ray tracing requires considerable computer time.

A receiver can be modeled as a point or sphere in space. In simple ray tracing a receiver hit occurs when a ray passes within a given radius of a listener. In beam tracing a hit is registered when the receiver falls within the pyramid beam triangle. If no receiver hit occurs, the ray is followed until it encounters a surface. When there is a surface hit, reflection calculations are performed based on the impact point using the surface normal, area, and material properties.

The output of a ray-tracing program includes the room's impulse response at a given receiver from which a number of metrics of interest can be calculated. When a receiver hit occurs, the arrival time, direction, and amplitude of the ray as a function of frequency are stored. Angles of incidence and amplitudes for each received ray allow an accurate reproduction of the sound (an auralization) of the space. Due to limits in computational capacity as well as human reaction to late arrivals, time limits are set on the propagation of rays. After a given time (or distance) a ray's remaining energy is given over to the reverberant field or simply abandoned.

FIGURE 22.6 Method of Images for a Rectangular Room

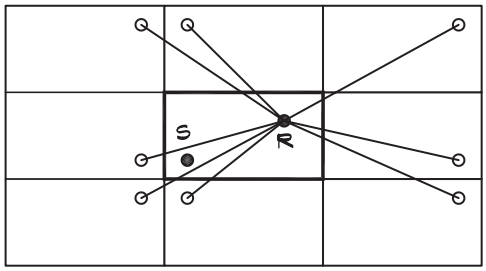


Image Source Method

The image source calculation method uses the mirror image of the source as the origination point for a new ray, drawn through the intersection point on the struck surface. Images of images are drawn until the desired order of reflection is achieved. The construction is straightforward for a simple rectangular enclosure but it becomes quite complicated if the room shape is complex. An example is given in Fig. 22.6. The number of image sources in a room of volume V for a radius $c\,t$ is (Rindel, 2000)

$$N_{refl} \geq \frac{4\pi c^3}{3\,V}\,t^3 \tag{22.2}$$

This is approximately the number of specular reflections arriving at the receiver up to a time t after the initial sound emission.

The method is quite accurate since no surfaces are missed, but the number of calculations required is very large. For n surfaces there are n first-order images, each of which can have $n-1$ second order images and so forth. For a reflection order i, the total number of sources is (Rindel, 2000)

$$N_{sou} = 1 + \frac{n}{(n-2)}\left((n-1)^i - 1\right) \cong (n-1)^i \tag{22.3}$$

Using Eq. 8.49, the mean free path in a room of 15,000 cu m (530,000 cu ft) is about 16 m (52 ft). In 600 ms about 13 reflections occur for a given ray path. Assuming a hall with 30 surfaces, the total number of image sources is $N_{sou} = 29^{13} \cong 10^{19}$, or more than we can comfortably accommodate. The procedure is also very inefficient since according to Eq. 22.2 only about 2500 of the images are valid for a given receiver.

Image-source reflection models must validate the accuracy and uniqueness of a ray path. This can be done by confirming the visibility by back tracing a ray path from the receiver to the source to make sure that the sequence of surfaces is the reverse of the original path. To avoid duplication an image tree list of the early reflection images is maintained (Rindel, 2000).

Hybrid Models

A hybrid model divides the room's time history into an early period where individual reflections are computed, and a late period where only the statistical reverberant tail remains. The division can be based on a user selection of the reflection order, after which reflections

become diffuse, or on the use of a diffusion coefficient, or both. In a diffuse reflection the impacted surface becomes a new source rather than a reflector of sound. The surface element is assumed to radiate into the hemisphere in front of it. The intensity of the scattered sound can be based on an assumed directivity pattern such as Lambert's cosine law (Rindel, 2000), where the intensity of the scattered energy is proportional to the cosine of the angle between the incoming ray and the vector to the receiver, or it can simply be uniformly distributed (Farina, 2000). Visibility of the receiver must also be checked for diffuse reflections.

The optimum transition between the specular and diffuse reflections would at first appear to be at as high a reflection order as possible. Interestingly this turns out not to be the case. The probability that an image is visible from a receiver is proportional to the size of the surface. At higher reflection orders, missed images are more likely. With more missed images the accuracy of the program decreases. As the order of reflection increases, the ratio of the cone or beam area to the reflecting surface area increases, which overstates the specular energy contribution. Rindel (2000) argues that in a hybrid model a low reflection order (2 to 3) and a relatively small number of rays (500 to 1000) should be sufficient to characterize a typical auditorium.

22.2 RAY TRACING

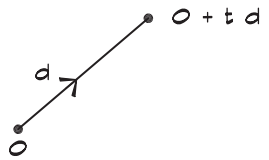
In a ray-tracing calculation the sound energy is assumed to travel from the source to the receiver like a beam or ray of light. Reflections from each surface are taken to be specular. Figure 17.8 shows a simple example done in a two-dimensional CAD drawing. A source is positioned in the space and sound rays are drawn at equal intervals (usually 5°–10°) originating from the source point. When a ray intersects a surface the reflected ray is constructed and followed until it hits the receiver. By altering the reflecting surfaces, different shapes and orientations can be tried until the rays are evenly distributed throughout the audience. At this point the surface orientation is acceptable for one source location. The process is repeated for a number of possible source locations until the design has been completed. When modeling reflections from isolated free-hanging panels, reflection points closer than a quarter wavelength from the edge of the panel should not be assumed to produce specular reflections. Two-dimensional ray tracing, even when it is done manually, can be an effective tool for designing small rooms.

Rays

The mathematical modeling of ray tracing has undergone rapid development in the last 30 years through the work of computer graphics programmers. Excellent references by Glassner (1989), Hill (1990), Shirley (2000), and Haines (2002) are available, which treat the subject in considerable detail. All these authors present treatments similar to that shown here. One must exercise care however since different authors use somewhat different (row or column) matrix formulations, and not all of the matrices in these references are directly interchangeable.

When modeling sound propagation in a three-dimensional room we begin with a default frame of reference consisting of three orthogonal basis vectors (**x**, **y**, **z**). All rays and surfaces are described mathematically relative to this coordinate system. Points are four-dimensional column vectors. Coordinate transformations are performed using 4×4 translation and rotation matrices, which were treated in Chapt. 18. Ray tracing deals with the interaction between rays and surfaces, either plane or curved. A ray is a line, most conveniently described in

FIGURE 22.7 Definition of a Ray



parametric form, using an origin (**o**), a direction vector (**d**), and a parameter (t) denoting the distance along the direction vector. Figure 22.7 gives an example.

$$\mathbf{p}(\mathrm{t}) = \mathbf{o} + \mathrm{t}\,\mathbf{d} \tag{22.4}$$

For example the line segment between points **a** and **b** is

$$\mathbf{p}(\mathrm{t}) = \mathbf{a} + \mathrm{t}\,(\mathbf{b} - \mathbf{a}) \tag{22.5}$$

where t is defined over the interval $\mathrm{t} \in [\mathrm{t_a}\,, \mathrm{t_b}]$, and in this case, $\mathrm{t} \in [0, 1]$. The direction vector is normalized to 1, and t can range to infinity. Equation 22.4 is an explicit or parametric form of the ray equation since any point on the ray can be determined simply by varying t.

Surfaces and Intersections

Surfaces are an infinite collection of points. Many of the simplest shapes can be modeled by using an implicit equation, which returns a zero value when a point (x,y,z) is on the surface (Shirley, 2000)

$$f(\mathrm{x}, \mathrm{y}, \mathrm{z}) = 0 \tag{22.6}$$

If a point is not on the surface, a nonzero value is returned. The function is called implicit because it allows a given point to be tested but does not allow the explicit calculation of a set of points. For a surface of points $\mathbf{p} = (\mathrm{x}, \mathrm{y}, \mathrm{z})$ there is an implicit function

$$f(\mathbf{p}) = 0 \tag{22.7}$$

which is satisfied for all points on the surface. If the value of the function is greater than zero, the point is "outside" the surface, and if less than zero it is "inside" the surface. For spheres and cylinders the definition of inside and outside is obvious, but for planes the definition depends on the problem context.

The intersection of a ray and a surface occurs when the equation for the ray satisfies the implicit function for the surface

$$f(\mathbf{p}(\mathrm{t})) = 0 \tag{22.8}$$

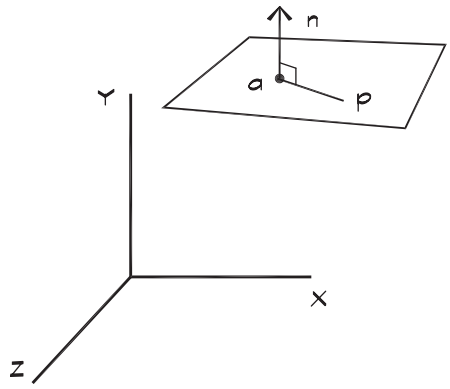
or

$$f(\mathbf{o} + \mathrm{t}\,\mathbf{d}) = 0 \tag{22.9}$$

The normal to the surface at the point of intersection, which is needed to calculate the reflected ray, is the gradient of the implicit function

$$\mathrm{n} = \nabla f(\mathbf{p}) = \left(\frac{\partial f(\mathbf{p})}{\partial \mathrm{x}}, \frac{\partial f(\mathbf{p})}{\partial \mathrm{y}}, \frac{\partial f(\mathbf{p})}{\partial z} \right) \tag{22.10}$$

FIGURE 22.8 Definition of a Plane



Planar Surfaces

To define the implicit function for an infinite plane containing point **a** and having normal **n**, as in Fig. 22.8, we write

$$(\mathbf{p} - \mathbf{a}) \cdot \mathrm{n} = 0 \tag{22.11}$$

Here **a** and **n** are known, and **p** is a sample point. When **p** lies in the plane defined by **n**, the vector drawn between **a** and **p** is normal to **n** if their dot product is zero.

Ray-Plane Intersection

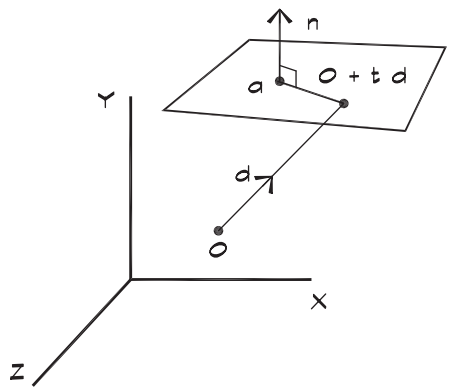
The intersection of a ray with a plane, shown in Fig. 22.9, is obtained by substituting Eq. 22.4 into Eq. 22.11

$$(\mathbf{o} + \mathrm{t}\,\mathbf{d} - \mathbf{a}) \cdot \mathbf{n} = 0 \tag{22.12}$$

and solving for t , the only unknown

$$\mathrm{t} = \frac{-(\mathbf{o} - \mathbf{a}) \cdot \mathbf{n}}{\mathbf{d} \cdot \mathbf{n}} \tag{22.13}$$

FIGURE 22.9 Ray-Plane Intersection



If the denominator is zero, the ray is parallel to the plane. If both the numerator and the denominator are zero, the ray lies in the plane.

In terms of the coefficients, a plane is defined by

$$a\,x + b\,y + c\,z + d = 0 \tag{22.14}$$

and the intersection is (Glassner, 1989)

$$t = \frac{-(a\,x_0 + b\,y_0 + c\,z_0 + d)}{a\,x_1 + b\,y_1 + c\,z_1} \tag{22.15}$$

where we have assumed that the scaling parameter is 1 in the four-element matrices, and the subscripts 0 and 1 refer to the values of the coefficients for **o** and **d**, respectively. The coefficient d is the shortest distance between the plane and the origin.

Ray-Polygon Intersections

Having determined the intersection point of a ray and an infinite plane we must see whether it lies within a polygon of interest. There are a number of techniques used to make this determination both for polygons of arbitrary shape and for the special case of triangles. The one shown here is from Haines (1989).

A polygon is defined by a list of vertices connected by edges. To see if a point on a plane lies within a polygon, an inside-outside test (known as the Jordan Curve Theorem) is performed by casting a ray, lying in the plane, in an arbitrary direction from the intersection point and counting the number of edge crossings. If the number is odd, the point is inside the polygon; otherwise it is outside. Figure 22.10 shows an example of a five-pointed star with a hollow center, and shows several scenarios. In this case horizontal rays are used to test the polygon.

The test is complicated for a plane in three-dimensional space. Fortunately it also works for the projection of the polygon onto an arbitrary two-dimensional surface. If one component of each point is removed, the remaining coordinates give the projection of the polygon onto a new **u-v** plane. The polygon can be translated so that the intersection coincides with the

FIGURE 22.10 Inside-Outside Crossing Test (Haines, 1989)

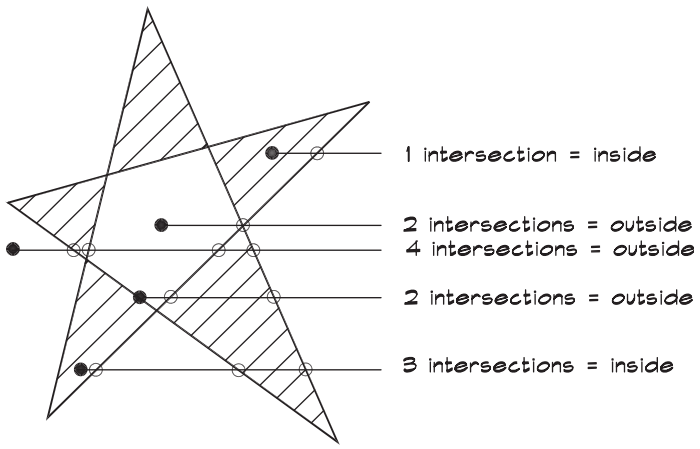
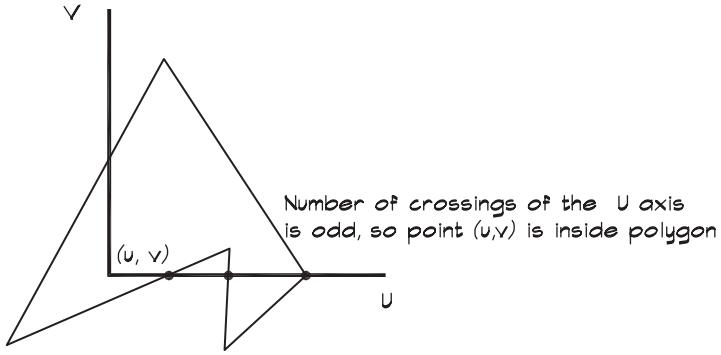


FIGURE 22.11 Polygon Crossing Test (Haines, 1989)



u-v origin. The intersection test then is performed using a ray aligned with the **u** axis as in Fig. 22.11.

If the plane of the polygon is normal to a basis vector, then the components in that direction are removed. Special care must be taken to address unique cases such as a vertex, which lies on the test ray axis. This can be done by defining a ray as incrementally above the **u** axis so that it does not coincide with the vertex (Berlin, 1985).

Ray-Sphere Intersection

A sphere, having a center $\mathbf{c} = (c_x, c_y, c_z)$ and radius R, is characterized by the second-degree polynomial (Shirley, 2000)

$$(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - R^2 = 0 \tag{22.16}$$

which in vector form is

$$(\mathbf{p} - \mathbf{c}) \cdot (\mathbf{p} - \mathbf{c}) - R^2 = 0 \tag{22.17}$$

The ray-sphere intersection is determined by plugging the ray formula into the sphere equation

$$(\mathbf{o} - t\,\mathbf{d} - \mathbf{c}) \cdot (\mathbf{o} - t\,\mathbf{d} - \mathbf{c}) - R^2 = 0 \tag{22.18}$$

By rearranging terms we obtain the solution in quadratic form

$$(\mathbf{d} \cdot \mathbf{d})\, t^2 + 2\,\mathbf{d} \cdot (\mathbf{o} - \mathbf{c})\, t + (\mathbf{o} - \mathbf{c}) \cdot (\mathbf{o} - \mathbf{c}) - R^2 = 0 \tag{22.19}$$

The equation for a sphere located at the origin is

$$x^2 + y^2 + z^2 - R^2 = 0 \tag{22.20}$$

and the point of intersection is

$$\begin{aligned} t^2 (x_1^2 + y_1^2 + z_1^2) + 2\,t\,(x_0 x_1 + y_0 y_1 + z_0 z_1) \\ + (x_0^2 + y_0^2 + z_0^2) - R^2 = 0 \end{aligned} \tag{22.21}$$

where the subscripts 0 and 1 again refer to the values of the coefficients for **o** and **d**, respectively. Note that the ray origin can be translated to maintain this location.

Equation 22.21 has the classic quadratic form

$$A\,t^2 + B\,t + C = 0 \tag{22.22}$$

that can be solved using

$$t = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{22.23}$$

Here the term under the radical, called the *discriminant,* tells how many real solutions there are. If positive, there are two intersections, one where the ray enters and the other where it leaves the sphere. If it is negative there are no intersections, and if zero the line is tangent to the sphere. It saves computing time to first check the discriminant to see if a solution exists. The general solution for t is

$$t = \frac{-\mathbf{d}\cdot(\mathbf{o}-\mathbf{c}) \pm \sqrt{(\mathbf{d}\cdot(\mathbf{o}-\mathbf{c}))^2 - (\mathbf{d}\cdot\mathbf{d})((\mathbf{o}-\mathbf{c})\cdot(\mathbf{o}-\mathbf{c}) - R^2)}}{(\mathbf{d}\cdot\mathbf{d})} \tag{22.24}$$

and the intersection point can be calculated by plugging the value of t into Eq. 22.1. If the sphere has been translated to the origin, **c** is set to zero.

The normal at the point of intersection is the gradient of Eq. 22.17

$$\mathbf{n} = 2\,(\mathbf{p} - \mathbf{c}) \tag{22.25}$$

and the unit normal is

$$\mathbf{n} = (\mathbf{p} - \mathbf{c})\,/\,R \tag{22.26}$$

Where there are two intersection points, the one having the smaller absolute value of t gives the nearer point. If the ray originates inside the sphere, the normal should be given a negative value.

Ray-Cylinder Intersection

The case of an infinite cylinder is similar to that of a sphere. A unit cylinder is formed by rotating the line $x = 1$ in the x-z plane about the z axis. For a cylinder of arbitrary radius having its center at the origin and its axis aligned with the z direction, the implicit function is

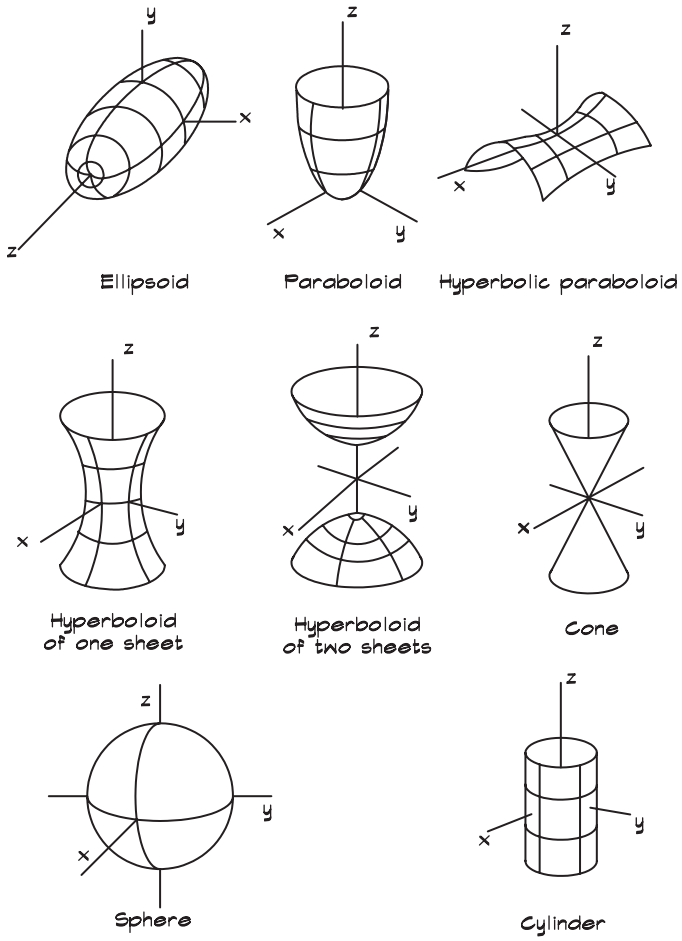
$$x^2 + y^2 - R^2 = 0 \tag{22.27}$$

and the ray intersection is given by

$$t^2\,(x_1^2 + y_1^2) + 2\,t\,(x_0x_1 + y_0y_1) + (x_0^2 + y_0^2) - R^2 = 0 \tag{22.28}$$

For a cylinder at an arbitrary location and orientation, the translation and rotation matrices necessary to move its center to the origin and to align its axis with the z coordinate are applied to the ray to maintain the proper relationship before solving for t. Once the value of t has been determined, it is substituted into Eq. 22.4 along with its direction vector to obtain the intersection point.

FIGURE 22.12 Quadric Surfaces



Ray-Quadric Intersections

Algebraic surfaces are implicit formulas of polynomials having the general form (Hanrahan, 1989)

$$\mathbf{p}\,(\mathrm{x},\ \mathrm{y},\ \mathrm{z}) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} \mathrm{a}_{ijk}\,\mathrm{x}^{i}\,\mathrm{y}^{j}\,\mathrm{z}^{k} \tag{22.29}$$

The degree of the surface is given by the maximum value of the coordinates $d = \max(l, m, n)$, and the number of terms in the equation describing a surface of degree d is $(d + 1)(d + 2)(d + 3)/6$. A group of surfaces known as quadrics, consisting of polynomials having degree 2, are shown in Fig. 22.12. The generalized implicit equation for a quadric surface has ten terms and is given by

$$\begin{aligned}\mathrm{ax}^2 + 2\mathrm{bxy} + 2\mathrm{cxz} + 2\mathrm{dxw} + \mathrm{ey}^2 + 2\mathrm{fyz} \\ + 2\mathrm{gyw} + \mathrm{hz}^2 + 2\mathrm{izw} + \mathrm{jw}^2 = 0\end{aligned} \tag{22.30}$$

This can be written in matrix form as

$$\mathbf{x}^{\mathbf{t}}\,\mathbf{Q}\,\mathbf{x} = 0 \tag{22.31}$$

where Q is a symmetric 4×4 matrix

$$\mathbf{Q} = \begin{bmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{bmatrix} \tag{22.32}$$

and $\mathbf{x}$ is a column vector and its transpose $\mathbf{x}^{\mathbf{t}}$ is a row vector. The general solution for the intersection point is determined by substituting in the ray equation (Eq. 22.4). We then obtain a general equation

$$a_2\,t^2 + a_1\,t + a_0 = 0 \tag{22.33}$$

where

$$\begin{aligned} a_2 &= \mathbf{x}_1^{\mathrm{t}}\,\mathbf{Q}\,\mathbf{x}_1 \\ a_1 &= 2\mathbf{x}_1^{\mathrm{t}}\,\mathbf{Q}\,\mathbf{x}_0 \\ a_0 &= \mathbf{x}_0^{\mathrm{t}}\,\mathbf{Q}\,\mathbf{x}_0 \end{aligned} \tag{22.34}$$

which can be solved using the quadratic equation. The subscripts 0 and 1 refer to the values for **o** and **d**, respectively.

The **Q** matrix can be diagonalized so that only the terms along its diagonal are nonzero. This is known as its canonical form since the principal axes of the surface are aligned with the canonical basis vectors (x, y, z), and its center is at the origin.

Ray-Cone Intersection

When a cone is in canonical coordinates its implicit equation is (Hanrahan, 1989)

$$x^2 + y^2 - z^2 = 0 \tag{22.35}$$

and the ray intersection is determined by solving

$$\begin{aligned} t^2\,(x_1^2 + y_1^2 - z_1^2) + 2\,t\,(x_0x_1 + y_0y_1 - z_0z_1) \\ + (x_0^2 + y_0^2 - z_0^2) = 0 \end{aligned} \tag{22.36}$$

Ray-Paraboloid Intersection

The canonical equation of a paraboloid is (Hanrahan, 1989)

$$x^2 + y^2 + z = 0 \tag{22.37}$$

and

$$\begin{aligned} t^2\,(x_1^2 + y_1^2) + 2\,t\,(x_0x_1 + y_0y_1 - z_1) \\ + (x_0^2 + y_0^2 - z_0) = 0 \end{aligned} \tag{22.38}$$

Although more complicated shapes are seldom necessary in acoustical ray tracing their implicit functions can be found in computer graphics texts such as Glassner (1989) or Hill (1990). Both quadric and superquadric shapes are also addressed in Ray Tracing News (RTNews), a web site maintained by Eric Haines.

22.3 SPECULAR REFLECTION OF RAYS FROM SURFACES

The mathematical characterization of reflections from surfaces is undergoing considerable research. The standard approach is to group the energy transfer into four components: specularly reflected, diffusely reflected, transmitted, and absorbed. Figure 21.13 shows the relationships between the components, which are characterized in terms of their coefficients. For room acoustics problems the transmitted energy is bundled with the absorbed energy and lumped into the absorption coefficient since it does not return to the originating space. In these formulas we seek useful and intuitive algorithms, which do not require massive computing power to utilize.

Specular Reflections

A specular or mirror-like reflection is pictured in Fig. 22.14. The reflected ray is given by (Shirley, 2000)

$$\mathbf{r} = \mathbf{d} + 2\,\mathbf{a} = \mathbf{d} + 2\,\frac{\mathbf{d}\cdot\mathbf{n}}{|\mathbf{n}|^2}\,\mathbf{n} \tag{22.39}$$

FIGURE 22.13 Reflection from a Surface

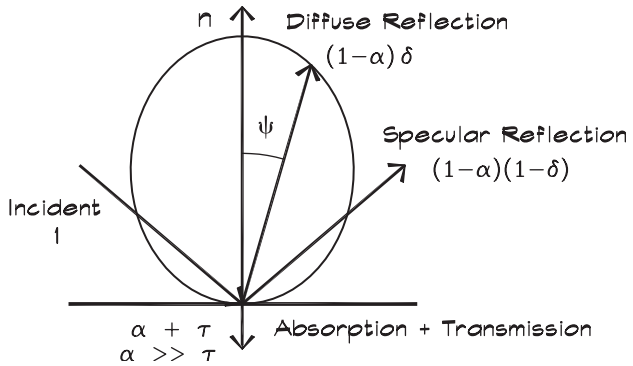
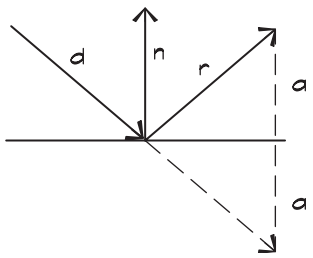


FIGURE 22.14 Specular Reflection from a Surface



If the normal vector of the reflecting plane is itself normalized, the denominator in Eq. 22.39 becomes one. Once the new ray direction has been established it can be traced until it triggers a level limit.

Specular Reflections with Absorption

The direct-field intensity at a receiver following a single specular reflection from a large absorptive surface is given by (Farina, 2000)

$$I_{spec} = \frac{W\,Q_{\phi}}{4\pi\,(r_1 + r_2)^2}\,(1-\alpha)\,e^{-\gamma(r_1 + r_2)} \tag{22.40}$$

where W = sound power of the source (W)
Q_{ϕ} = source directivity for a given angle ϕ
α = absorption coefficient (usually diffuse field)
γ = air absorption coefficient $\cong \dfrac{1.7 \cdot 10^{-8} f^2}{\varphi}$
r_1 = distance between the source and the surface (m)
r_2 = distance between the surface and the receiver (m)
φ = relative humidity (%)

The amplitude of specularly reflected rays is calculated using the absorption coefficient of the reflecting surface. Either the diffuse field absorption coefficient or a specular coefficient derived from it is used in most computer models. The standard reverberant measurement method smoothes out some of the variation and produces repeatable results. Diffuse absorption coefficients have the advantage of availability and convenience even though they only account for the angle of incidence in a statistical way.

Ideally a specular absorption coefficient, that accounts for the angle of incidence of the incoming ray, should be used, but these are rarely available from direct measurements and must be estimated from impedance data or from the diffuse coefficients. With impedance data the angular behavior of the specular absorption coefficient can be calculated from Eq. 7.63. One cannot, however, calculate the impedance from the diffuse-field absorption coefficient simply since there are an infinite number of combinations of real and imaginary parts that would give the same result.

Rindel (1993) suggested a method for estimating the angular dependence of the absorption coefficient by neglecting the imaginary part of the impedance and assuming the real part to be independent of angle of incidence; that is, the surface is locally reacting. The real part of the surface impedance is based on the diffuse-field absorption coefficient for a standard test surface of 11 sq m, assuming an angle of incidence around 55° – 60° and $\alpha_{60} \cong \alpha$.

$$r_{60} \cong \pm\sqrt{1-\alpha} \tag{22.41}$$

where r can be positive or negative. The relation for the pressure reflection factor is

$$r_{\theta} = \frac{Z_a - Z_f(\theta)}{Z_a + Z_f(\theta)} \tag{22.42}$$

where Z_a = surface impedance, which depends on the material properties
$Z_f(\theta)$ = field impedance, which depends on the angle of incidence and the panel size
α = diffuse field absorption coefficient
α_{60} = absorption coefficient at about a 60° angle of incidence
r_θ = reflection coefficient for a given angle

When the sample size is infinite the field impedance is $Z_f(\theta) \rightarrow \rho c / \cos\theta$ and is real. For a finite area it is complex. Rindel (1993) has suggested an empirical relationship for its real part

$$Z_f(\theta) \cong \rho c \left[\left(\cos^2\theta - 0.6 \frac{2\pi}{k e} \right)^2 + \pi \left(0.6 \frac{2\pi}{k e} \right)^2 + \left(\frac{2\pi}{(k e)^2} \right)^4 \right]^{-1/4} \tag{22.43}$$

where k = wave number $= \dfrac{2\pi f}{c}$ (m^{-1})
$e = \dfrac{4S}{P}$ = characteristic dimension of the test sample (m)
S = surface area of the test sample (m^2)
P = perimeter of the test sample (m)
f = frequency (Hz)
c = velocity of sound in the surface material (m/s)
ρ = density of the surface material (kg/cu m)

Values of the equivalent field impedance, Z_f^*, measured under diffuse field conditions, suggested by Rindel (1993) for a test area of 11 sq m, are given in Table 22.1. The angle used in calculating the factor varies with frequency, approaching 52.5°, above 1000 Hz.

An equivalent real part of the material impedance can be calculated using Eq. 22.41 and 22.42

$$Z_a = Z_f^* \frac{1 + r_\theta}{1 - r_\theta} = Z_f^* \frac{1 \pm \sqrt{1 - \alpha}}{1 \mp \sqrt{1 - \alpha}} \tag{22.44}$$

The sign of the radical is chosen according to the material in use. If the material is hard such as concrete, drywall, or wood, the upper sign is selected. If it is soft, having an impedance less than $2 \rho_o c_o$, then the lower signs should be used. Using Eq. 22.43 and 22.44 in Eq. 22.42 we obtain an expression for the angular dependence of the absorption coefficient in terms of the diffuse field coefficients and the size of the reflecting surface.

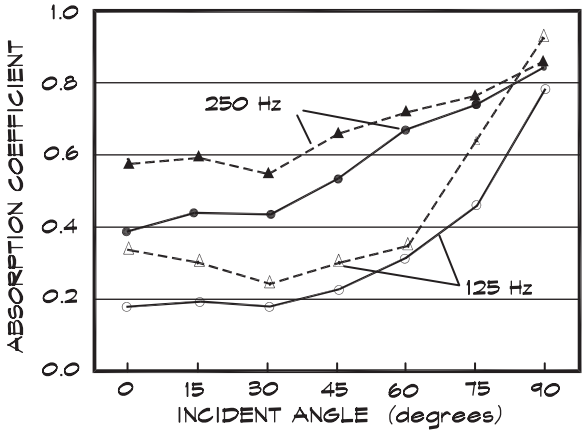
Specular Absorption by Seats

Nishihara et al. (2001) published measurements of the absorption of a seated audience in chairs with differing thicknesses of padding at varying angles of incidence. They suggest

TABLE 22.1 Suggested Field Impedance for an 11 sq m Sample (Rindel, 1993)

Frequency (Hz)	125	250	500	1k	2k	4k
$Z_f^* / \rho c$	1.04	1.35	1.53	1.62	1.64	1.64

FIGURE 22.15 Oblique Absorption Coefficients at the Hypothetical Plane of a Seated Audience (Nishihara et al., 2001)



Calculations based on measurements at Musikvereinssaal (solid line) and Burgtheater (dotted line).

that the use of this type of data is more accurate because the distribution of the angle of incidence in a concert hall is not truly uniform. Often these numbers are dependent on surface irregularities, edge conditions, sample size and orientation, and other details of the measurement methodology. The measured angular behavior of low-frequency absorption they found is shown in Fig. 22.15. Note the increase in the absorption coefficient at angles approaching 90°.

22.4 DIFFUSE REFLECTION OF RAYS FROM SURFACES

The characterization of diffusion is currently undergoing considerable study. It has been assumed that a diffusing surface can be characterized by means of a coefficient, similar to the Sabine absorption coefficient, that varies between 0 (specular reflection) and 1 (complete diffusion), which can be measured in a standard way. Just as was the case with absorption, there may be different formulations depending on whether we are in a free field, having a given angle of incidence, or a diffuse field, with a random angle of incidence. To characterize the surface diffusion a scattering coefficient and measurement methodology were proposed by Mommentz and Vorlander, (1995)

$$\delta = \frac{E_{diff}}{E_{tot}} = \frac{E_{diff}}{E_{spec} + E_{diff}} \tag{22.45}$$

The scattering coefficient is the ratio between the diffusely reflected energy and the total reflected energy. Here a diffuse reflection is defined as any nonspecular reflection, even though a perfect diffuser will radiate a portion of the incident energy in the specular direction and even though a redirection rather than perfect diffusion of the energy may be what is occurring.

Measurement of the Scattering Coefficient

The measurement of all the diffuse energy is difficult so most methods measure the specularly reflected energy and use the diffuse absorption coefficient to calculate it

$$\delta = \frac{E_{diff}}{E_{tot}} = 1 - \frac{E_{spec}}{E_{tot}} \tag{22.46}$$

where $E_{spec}/E_{inc} = 1 - \alpha_{spec} =$ normalized specularly reflected energy
$E_{tot}/E_{inc} = 1 - \alpha =$ normalized total reflected energy
so,

$$\delta = \frac{\alpha_{spec} - \alpha}{1 - \alpha} \tag{22.47}$$

which gives us a way of measuring the diffusion coefficient, δ.

Farina (2000) has suggested the technique pictured in Fig. 22.16, using a microphone 2 m above the floor that is dollied past a diffusing surface suspended 3.65 m above a loudspeaker flush mounted into the floor. Impulse response data are recorded at regular (28 mm) microphone intervals. Time windowing filters out the direct sound so that the impulse response of the reflection from the panel under test is obtained. The results from various panels are given in Fig. 22.17. Although this method yields consistent results, the technique has not been accepted as a standard. Some (e.g., D'Antonio, 1995) prefer a diffusion coefficient that characterizes the evenness of the diffused energy, based on the standard deviation of the measurements around a test surface. This approach is helpful for a comparison of products but not, as yet, particularly useful for modeling applications.

Diffuse Reflections

When a sound ray impacts a surface some of the energy is reflected specularly and some is diffused. The portion of the energy reflected specularly creates an intensity at a receiver (Naylor, 1993)

$$I_{spec} = \frac{W Q_{\phi}}{4\pi (r_1 + r_2)^2} (1 - \alpha)(1 - \delta) e^{-\gamma (r_1 + r_2)} \tag{22.48}$$

FIGURE 22.16 Measurement Methodology for the Specular Absorption Coefficient (Farina, 2000)

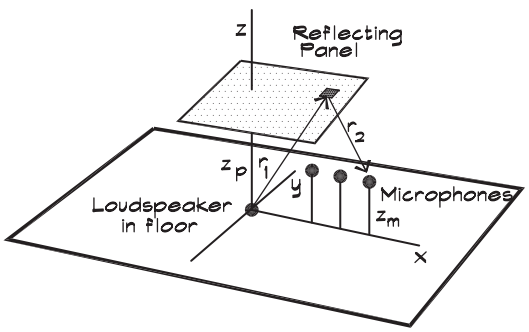
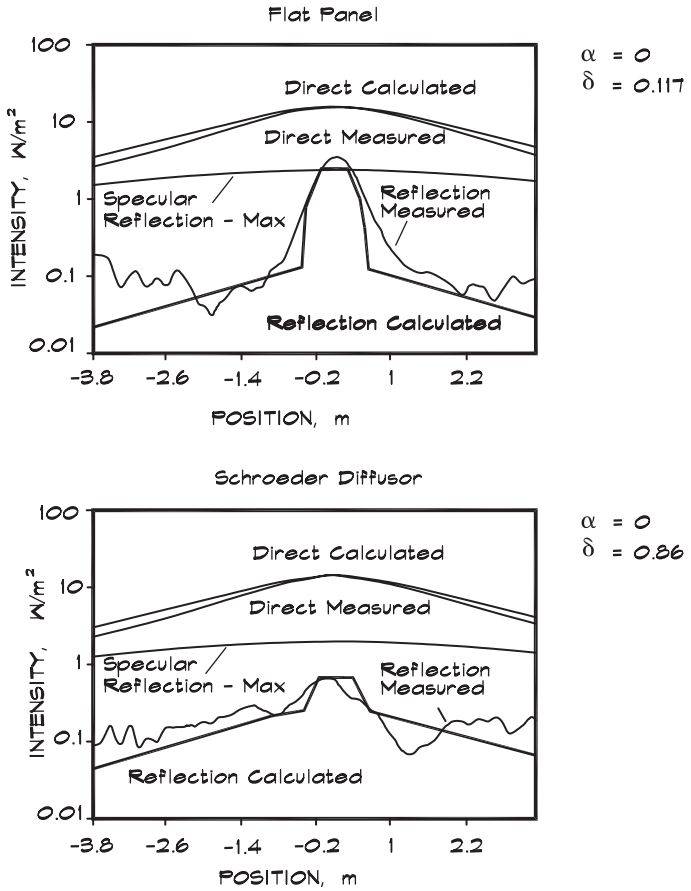
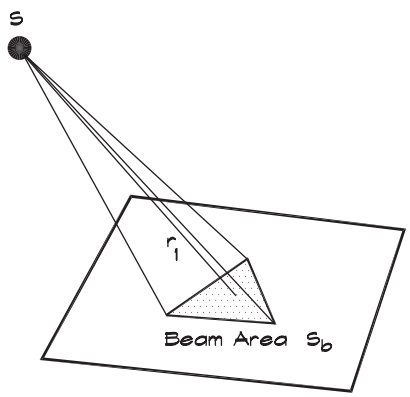


FIGURE 22.17 Comparison of Measured and Predicted Specularly Reflected Intensities (Farina, 2000)



The remaining unabsorbed energy is assumed to be scattered. Most models treat the reflecting surface as a new sound source, radiating the diffuse portion of the energy into the hemisphere in front of it. In the simplest condition, where the beam area associated with an individual ray is smaller than the area of the impacted surface, as in Fig. 22.18, the diffuse

FIGURE 22.18 Pyramid Beam Impacting a Large Surface



energy radiating from the surface is

$$W_{diff} = \frac{W Q_{\phi}}{N_r} \delta (1 - \alpha) e^{-\gamma r_1} \tag{22.49}$$

where N_r = number of equally spaced rays. Using Eq. 2.90 the intensity from a diffuse reflection at a receiver within the hemisphere is

$$I_{diff} = \frac{W Q_{\phi} Q_S \delta (1 - \alpha) e^{-\gamma (r_1 + r_2)}}{4 \pi N_r \left[r_2 + (Q_s S_b / 4 \pi)^{1/2}\right]^2} \tag{22.50}$$

where $S_b = \dfrac{4 \pi r_1^2}{N_r}$ = beam area at a distance r_1 from the source

Q_s = directivity of the reflecting surface
= 2 for an omnidirectional hemisphere
= $2 \cos \psi$ for Lambert's cosine law

ψ = angle between the scattering surface normal and the outgoing ray

Some programs (e.g., ODEON) use directional scattering of the diffuse energy according to Lambert's cosine law, which is introduced using the surface directivity Q_s, and others (e.g., RAMSETE) use omnidirectional scattering.

Multiple Reflections

When a ray undergoes multiple reflections the effect of each successive surface's absorption and diffusion coefficients must be considered. Accordingly the surface absorption coefficient and the air absorption term for a single reflection are replaced by a product

$$(1 - \alpha) e^{-\gamma (r_1 + r_2)} \Rightarrow \prod_{i=1}^{n} (1 - \alpha_i) e^{-\gamma (r_{tot} + r_2)} \tag{22.51}$$

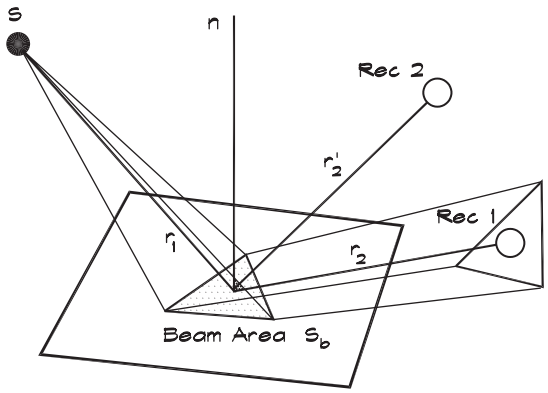
The total diffuse scattering coefficient represents the diminution in the intensity due to a succession of diffuse reflections, each of which removes some of the energy from the specular component. It is defined as

$$\delta_{tot} = \delta_{tot-1} + (1 - \delta_{tot-1}) \delta_n \tag{22.52}$$

The total diffusion coefficient increases slightly with each reflection until it approaches 1. At this point all reflections are diffuse and the sound field is totally reverberant.

A given receiver is illuminated by both specular and diffuse reflections. Figure 22.19 shows two hypothetical receivers. If a receiver lies within the specularly reflected pyramid beam, it receives both the specular and diffusely reflected intensities. If it is outside the beam it receives only the diffuse intensity, but these reflections come from every surface having a diffusion coefficient greater than zero. After each reflection the intensity of the remaining direct sound field is reduced by the amount absorbed and by the amount diffused.

FIGURE 22.19 Pyramid Beam Impacting a Surface (Farina, 2000)



When there are multiple specular reflections from a series of surfaces the received intensity after the nth reflection is (Farina, 2000)

$$I_{spec,n} = \frac{W\,Q_{\phi}}{4\pi\,(r_{tot} + r_{n+1})^2}\left[\prod_{i=1}^{n}(1-\alpha_i)\right](1-\delta_{tot})\,e^{-\gamma\,(r_{tot}\,+\,r_{n+1})} \tag{22.53}$$

where r_{tot} = total distance (including reflections) between the source and the current surface (m)

r_{n+1} = distance between the current surface and the receiver (m)

and the diffuse intensity from a large scattering surface after a series of reflections is

$$I_{diff,n} = \frac{W\,Q_{\phi}\,Q_s\left[\prod_{i=1}^{n}(1-\alpha_i)\right]\delta_{tot}\,e^{-\gamma\,(r_{tot}\,+\,r_{n+1})}}{4\pi\,N_r\left[r_{n+1} + (Q_s\,S_{tot}/4\pi)^{1/2}\right]^2} \tag{22.54}$$

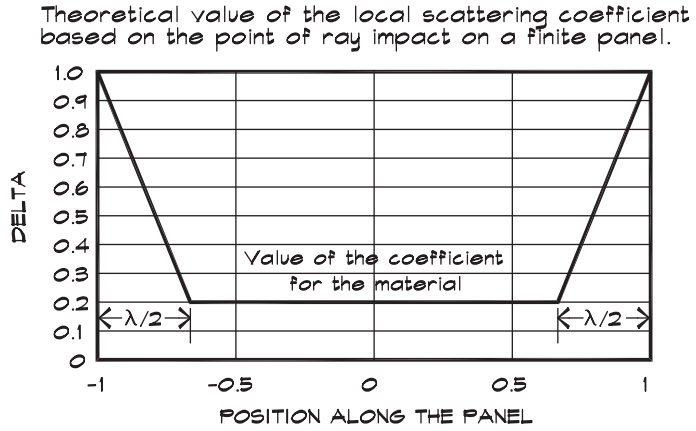
where $S_{tot} = \dfrac{4\pi\,r_{tot}^2}{N_r}$ = beam area at a distance r_{tot} from the source

Edge Effects

As the ray-panel intersection point approaches a panel edge, purely specular scattering no longer occurs. Farina (2000) has suggested that edge and size effects be accounted for by using a linear interpolation of the diffusion coefficient between the edge of a panel and a point, which is a half wavelength from the edge; at this point the panel is assumed to be characterized by the full value of the coefficient. Figure 22.20 shows an example. Between the two points a local diffusion coefficient is assumed to be in effect (Farina, 2000)

$$\delta_{loc} = \delta_0 + (1-\delta_0)\left(1 - \frac{d}{\lambda/2}\right) \tag{22.55}$$

FIGURE 22.20 Local Value of the Scattering Coefficient (Farina, 2000)



where d = distance from the edge of the surface (m or ft) $d \in [0, \lambda / 2]$
λ = wavelength of the incident sound (m or ft)
δ_0 = value of the scattering coefficient characteristic of the material

Note that even when there is a flat reflector, with a surface diffusion coefficient of 0, the diffusion due to the edge effect is addressed in Eq. 22.55. The inclusion of edge effects in this manner helps account for diffraction near the end of a freely supported reflector, but overestimates the diffuse energy when there are adjacent reflecting surfaces whose normals are close together.

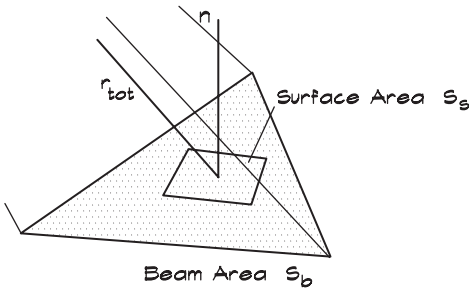
Hybrid Models and the Reverberant Tail

Every ray tracer has to deal with the transition between the early direct reflections, and the late reverberant field, and the limits imposed by computation time. Some programs (e.g., ODEON; Rindel, 2000), called hybrid models, use scattering coefficients in the normal way for the first two or three reflections, after which they abandon specular reflections and treat subsequent impacts as diffuse. The direction of the diffuse reflection can be selected at random by choosing an angle whose probability is based on Lambert's cosine law. This simplifies the calculations required and saves time but can miss multiple reflections.

Other programs (e.g., RAMSETE; Farina, 2000) remove a portion of the diffuse energy at each reflection, calculate its contribution at the receiver, and follow it no further. This simplification underestimates the diffuse-field energy, which might partake in multiple reflections after the initial one, but offsets the overestimation from the edge calculation. In this system all the energy remaining in a beam pyramid is assumed to be reflected specularly from the surface struck by the center ray even when the area of the beam pyramid is much greater than the reflecting area. The error introduced by this assumption overestimates the amount of specular energy reflected, but significantly reduces the computation time.

When the beam area is larger than the reflecting surface as in Fig. 22.21, not all the energy associated with a ray hits the surface, and a decision must be made as to how to treat the reflection. One approach is to clip the beam using the edges of the area struck, generating new beams, which continue on (Funkhouser et al., 1996). This approach is quite accurate but requires more computing power. Another approach is to reduce the specular energy by the ratio of the struck surface area to the beam area (when the beam area exceeds

FIGURE 22.21 Pyramid Beam Impacting a Small Surface



the surface area). The struck surface area would replace the beam area in Eq. 22.54 for the diffuse scattering component from the surface. The energy not striking the surface would be considered diffuse and could continue on a random or a diffused path. This would limit the specularly reflected energy to that actually hitting the surface.

The computation problem is a difficult one and is a topic of considerable interest. It may be possible some day to solve the room modeling problem exactly, but in the meantime clever models can give useful results with relatively simple approximations.

22.5 AURALIZATION

One product of a ray tracing model is the directional impulse response of a receiver in the space under study. This can be represented as a simple two-dimensional energy-time plot, which can serve as a basis for additional calculations such as reverberation time or other metrics discussed in Chapt. 19. In a more complicated form it includes directional information about each ray as well. An example based on Boston Symphony Hall is shown in Fig. 22.22. The y axis of the vertical distribution is based on the terms under the integral in the Paris equation where the probability of a ray coming from a given direction, $P(\theta)$, has replaced the diffuse-field probability, $\sin\theta$. Impulse response data can be obtained from scale model measurements as well as measurements taken in full-sized auditoria. In this manner models can be checked against actual field measurements.

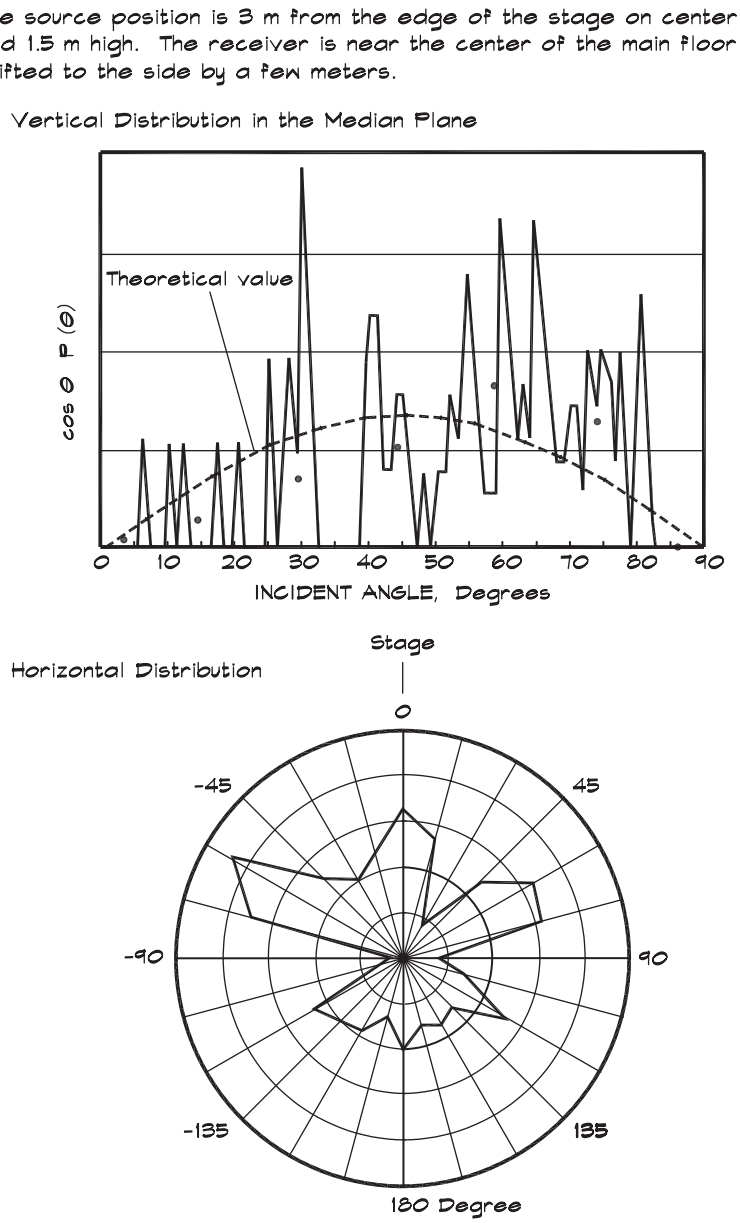
Convolution

When the room response has been determined for a source and receiver location and orientation, the resulting impulse response can be convolved with a dry (anechoic) audio signal to obtain calculated results of playing sound in the room, a process called auralization. The theory is based on the Duhamel or convolution integral, which was discussed in Chapt. 11. For a linear system there is a transfer function that yields the system output for a given input (stimulus).

$$y(t) = \int_0^{t_p} f(\tau)\, h(t-\tau)\, d\tau \tag{22.56}$$

where $y(t)$ = output response of the room
$h(t-\tau)$ = impulse response of the room
$f(\tau)$ = forcing (input) function of the system
t_p = time period of interest

FIGURE 22.22 Directional Ray Arrival Distribution for a Model of Boston Symphony Hall (Nishihara et al., 2001)

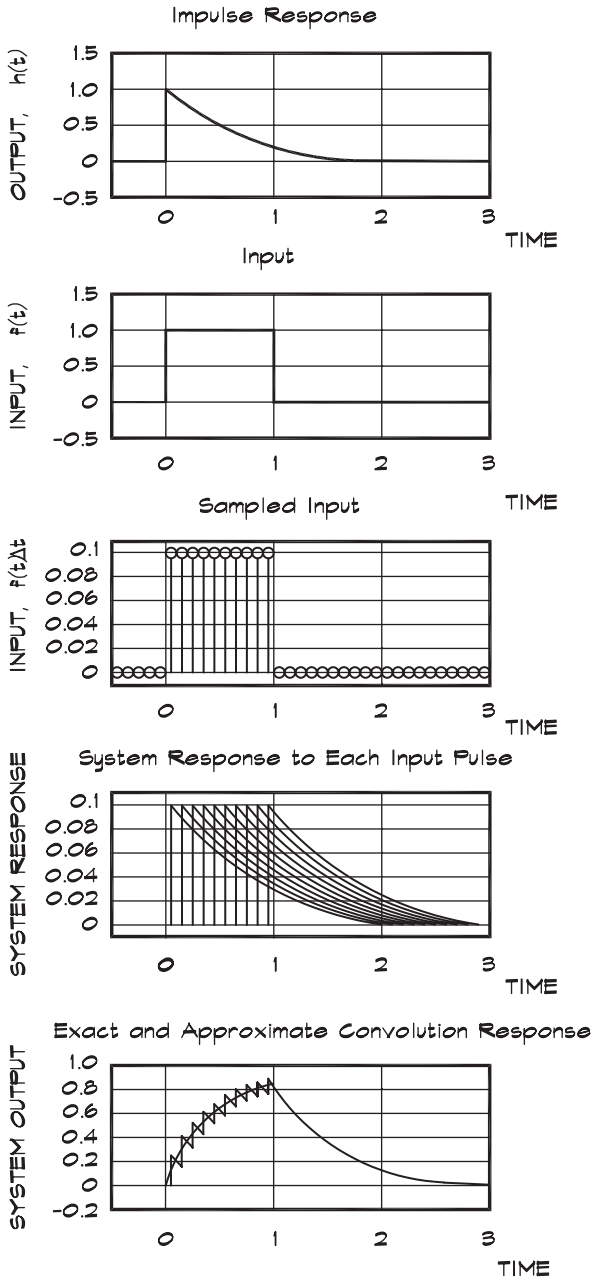


When the forcing function is a delta function, the system output is its impulse response

$$y(t) = \int_0^{t_p} \delta(\tau)\ h(t-\tau)\, d\tau = h(t) \tag{22.57}$$

Once the system response to a unit impulse (delta function) is known then the response to an arbitrary stimulus can be calculated.

FIGURE 22.23 Example of a Convolution Integral (Cheever, 2000)



We can take an example by assuming that a room can be characterized by an impulse response

$$h(t) = e^{-t/\tau 0} \tag{22.58}$$

where τ_0 = characteristic decay time. For an arbitrary value of $\tau_0 = 0.5$ s, we can calculate the system behavior for $h(t) = e^{-2t}$ and a square-wave input. Figure 22.23 shows a plot

of this system's behavior with time. The overall response is obtained from a summation of the responses from individual impulses. In practice the convolution of the signal with the room impulse function is done digitally by a series of multiply and add functions in a computer. The result in this simple example yields the room response on onset of the stimulus

$$I(t) = I_0 \, (1 - e^{-t/\tau 0}) \tag{22.59}$$

and decay after its removal

$$I(t) = I_0 \, e^{-t/\tau 0} \tag{22.60}$$

The information contained in the output of the ray tracer is not only the energy-time behavior for the space, but also the direction of the incoming reflections. A receiver is characterized by a hedgehog of rays representing specular or diffuse reflections from a scattering surface. For every ray arriving at the listener there is a delay time, a frequency-related attenuation, and a direction to be applied to the original signal.

Directional Sound Perception

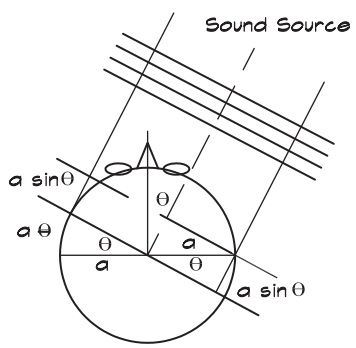
The recreation of a three-dimensional sound field relies on our ability to fool the listener into believing that the sound origination point is where we desire it to be. Accordingly we must examine the directional cues that furnish us with the information necessary to determine direction.

Rayleigh (1896) proposed a duplex theory for the perception of azimuthal sound direction. Two components were involved, the interaural time difference (ITD) and the interaural level difference (ILD) between the two ears. The ITD has a geometrical explanation, which is most simply understood by assuming a spherical shape to the head shown in Fig. 22.24.

The ITD can be calculated using this geometry for plane waves (Woodworth et al., 1954)

$$\text{ITD} = \frac{a}{c_0} \, (\theta + \sin\theta), \quad -\pi \, / \, 2 \leq \theta \leq \pi \, / \, 2 \tag{22.61}$$

FIGURE 22.24 Geometry of the Head for Azimuthal Direction Cues



When the angle of incidence is zero—that is, when the source is in the medial plane of the listener—the IDT is zero. When the source is to the side, the value of the IDT has a maximum value

$$\mathrm{ITD}_{\max} = \frac{a}{c_0}(\pi/2 + 1) \tag{22.62}$$

The measured value for a typical subject is around 0.7 ms. At high frequencies, above about 1.5 kHz, there is an aliasing problem in determining the time delay. Using the level difference resolves the ambiguity.

The second part of Raleigh's theory is due to the diffraction of the sound by the head, which reduces the sound level at the ear on the shielded side. Following the geometry in Fig. 22.24, the path-length difference for parallel sound incidence is (Blauert, 1983)

$$\Delta r = a\theta \tag{22.63}$$

and for a point source near the head assuming one unshielded ear

$$\Delta r = a\left[\left(n + \frac{1}{2}\right)\cos\varepsilon + \frac{1}{2}(\theta + \varepsilon) - \sqrt{n^2 + n + \frac{1}{2} - \left(n + \frac{1}{2}\right)\sin\theta}\right] \tag{22.64}$$

where $n = \dfrac{r - a/2}{a}$ and $\varepsilon = \arcsin\ (a/2r) = \arcsin\left(\dfrac{1}{1 + 2n}\right)$.

The shielding is then calculated using Eq. 5.11. At high frequencies the shielding may be as much as 20 dB; at low frequencies there is little level change and we rely on the time difference.

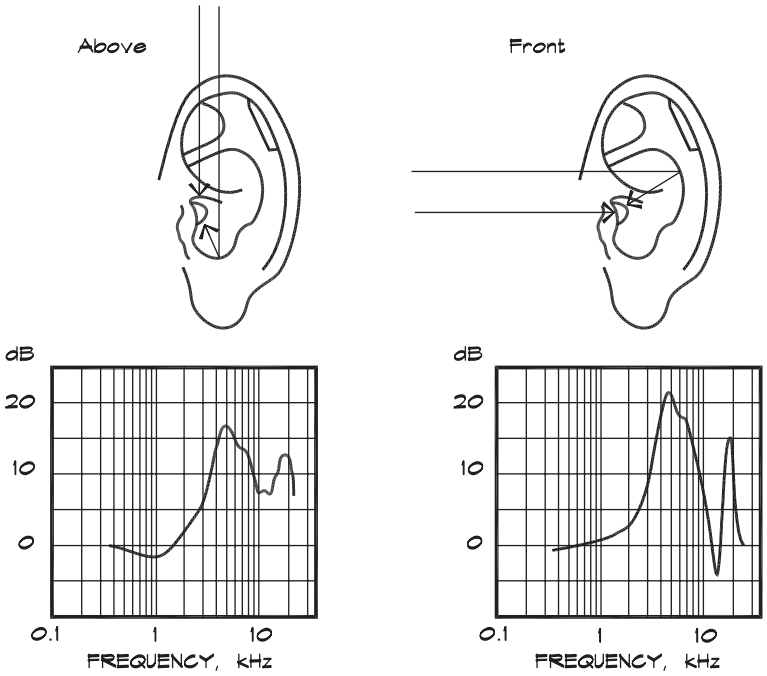
Elevation cues come from reflections from the pinna or outer ear (CIPIC, 2001). The frequency response of the ear is directional as shown in Fig. 22.25. The cues to an overhead source stem from the fact that the pinna acts like a focusing device whose frequency response is directionally dependent. The reflection from the pinna produces a second path to the ear canal and the possibility of destructive interference. The greatest effect occurs at the half-wavelength path difference, which produces a pinna notch in the 6 to 16 kHz range. Both the notch frequency and depth change with source elevation, the depth being much more pronounced for sounds coming from the front.

The perception of range is less well understood (CIPIC, 2001). It is influenced by the level of the direct sound, motion parallax, the direct-to-reverberant ratio in enclosed spaces and for sources quite close to the ear, a high degree of ILD. Motion parallax is the change in the azimuthal angle as the receiver moves, which is range dependent. The direct-to-reverberant ratio varies with the distance between the source and receiver.

Directional Reproduction

The simplest directional reproduction is accomplished by locating a loudspeaker at the desired source position or, short of that, in the source direction. This is a commonly employed

FIGURE 22.25 Elevation Cues from Pinna Reflections (CIPIC, 2001)



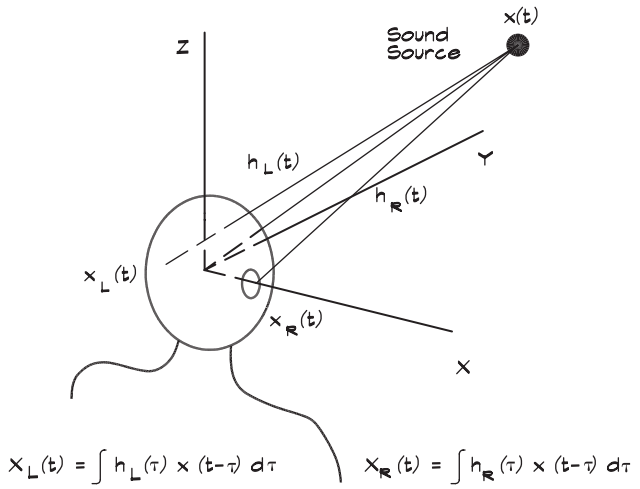
technique in theme park audio design where effects or dialog loudspeakers are located in props or set pieces constructed of perforated metal or other transondent materials. If there is more than one loudspeaker such as a central cluster with additional overhead loudspeakers, the earliest arrival time determines the perceived direction.

Stereo loudspeakers are the most commonly encountered reproduction configuration. In stereo reproduction, a phantom image can be positioned between the loudspeakers by panning or adjusting the volume of the signal sent to each one. There is a relatively small area near the centerline between the loudspeakers where an accurate stereo image is perceived. In careful recordings a sense of depth as well as width is apparent. Multichannel surround sound systems feature a series of loudspeakers in the azimuthal plane, which can act as groups of stereo pairs so that the apparent source direction can be set by panning between them.

Headphones can also be used to reproduce a stereo signal although most listeners prefer loudspeakers. Headphones interact with the pinna and alter the spatial relationships so elevation effects are not reliably reproduced (CIPIC, 2001). In a special case known as binaural recording, microphones are placed in the ear canals of a specially fitted mannequin so the listener is presented, through headphones, with the sound "heard" by the mannequin. The effect of binaural recording is startlingly real for certain source positions, mainly behind and to the side of the listener. Sources in front are difficult to localize and appear closer than they actually are. In many cases, frontal sounds appear to be located inside the listener's head.

Binaural sound can also be delivered to the listener from loudspeakers using a technique called cross-talk-canceled stereo. To do this we need to measure the effect the head has on sound coming from a given direction. Figure 22.26 illustrates the basic geometry. We measure the impulse response, h(t), of each ear for a given source position. This is called

FIGURE 22.26 Head-Related-Transfer-Function Measurements (CIPIC, 2001)



the Head-Related-Impulse-Response (HRIR) and its Fourier transform, H(f), is the Head-Related-Transfer-Function (HRTF). Once the HRTF is known for each ear we can reproduce a binaural signal from a monaural signal.

The HRTF are pairs of equalization curves (one for each ear) for every position in space. To simplify the measurement the source is positioned in the far field so the functional dependence reduces to azimuth, elevation, and frequency. Another approach is to model the HRTF based on the geometrical considerations previously discussed. A few examples are given in Fig. 22.27 and others are available online at CIPIC.com. These can be convolved with a dry signal using a special processor and presented to the listener by means of headphones or loudspeakers. Some systems also include a head tracker to account for the motion of the listener's head, to provide additional realism to the experience.

The recreation of three-dimensional cues from stereo loudspeakers, suggested by Schroeder and Atal (1963), is illustrated in Fig. 22.28. Ideally we want the left loudspeaker to deliver its signal only to the left ear and the right loudspeaker to the right ear. When a pair of loudspeakers is used, each source will deliver signal to both ears. Expressed mathematically the signal Y_1 reaching the left ear is a mixture of the pure signal S_1 and the crosstalk from S_2. In terms of the transfer functions

$$Y_1 = H_{11}\, S_1 + H_{12}\, S_2 \tag{22.65}$$

where H_{11} = HRTF between the left loudspeaker and the left ear
H_{12} = HRTF between the right loudspeaker and the left ear

and for the right ear

$$Y_2 = H_{21}\, S_1 + H_{22}\, S_2 \tag{22.66}$$

In matrix form the signals are

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{22.67}$$

FIGURE 22.27 Selected Head Related Transfer Functions (CIPIC, 2001)

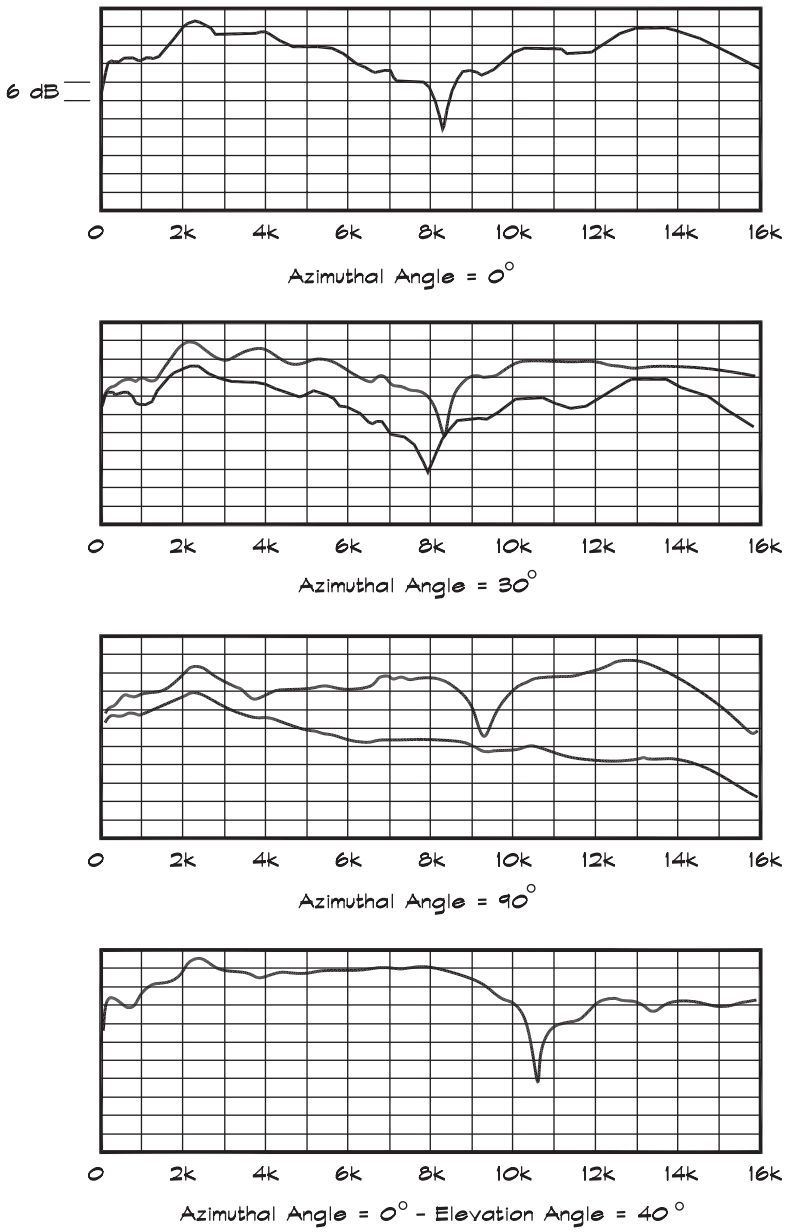
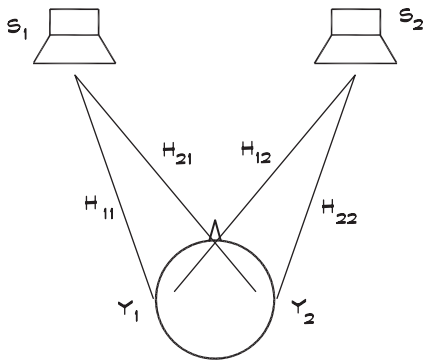


FIGURE 22.28 Binaural Reproduction Using Stereo Loudspeakers (Schroeder and Atal, 1963)



To find the proper signals we must invert the matrix

$$\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \tag{22.68}$$

which is not trivial. As the source receiver distance increases the impulse response becomes quite long. The signals are unique to the source and listener position. In practice if the listener turns his head by more than about 10° to 15°, the effect is lost.

REFERENCES

Abby Road Studios (2002). Private communication, plan reproduced with permission, 2002.

Adams, (2002). Dave L. Adams, "Firelight Lodge Sound Tests," (DLAA Project 6459). Denver, CO: Dave L. Adams Associates, 2002.

Alexandrovich (1987). George Alexandrovich, "Disk Recording and Playback," Handbook for Sound Engineers, The New Audio Cyclopedia, Glen Ballou, ed. © 1987 Indianapolis, IN: Howard W. Sams

Algazi et al. (2001). V.R. Algzi, R.O. Duda, D.M. Thompson, and C. Avendano, "The CIPIC HRTF Database." Proc. 2001 IEEE Workshop on Applications of Signal Processing to Audio and Electroacoustics, pp. 99–102.", New Paltz, NY: Mohonk Mountain House, Oct. 2001.

Allen (1974). D.L. Allen, "Vibrational Behaviour of Long-Span Floor Slabs." Canada: J. Civ. Eng., vol. 1, no. 1, 1974.

Allen (1997). D. Allen. Private Communication, 1997.

Allen (1976). D.E. Allen and J.H. Rainer, "Vibration Criteria for Long-Span Floors," Canada: Canadian Journal of Civil Engineering, vol. 3, no. 2, June 1976.

Allen (1975). D.L. Allen and John C. Swallow, "Annoying Floor Vibrations—Diagnosis and Therapy." Bay Village, OH: Sound and Vibration Magazine, 1975.

Anderson, et al. (1973). G.S. Anderson, L.N. Miller, and J.R. Shadley, "Fundamentals and Abatement of Highway Traffic Noise." Office of Environmental Policy, Federal Highway Administration, U.S. Department of Transportation, Bolt Beranek & Newman, June 1973.

Ando (1985). Y. Ando, Concert Hall Acoustics. New York, NY: Springer-Verlag, 1985.

Andree (1932). C.A. Andree, "The Effect of Position on the Absorption of Materials for the Case of a Cubical Room." from J. Acoust. Soc. Am. 3: 535–551, © 1932, Acoustical Society of America: Melville, NY.

ANSI (1971). ANSI S1.4, American National Standard Specification for Sound Level Meters. New York, NY: American National Standards Institute, 1971.

ANSI (1969). Excerpted from ANSI S3.5-1969 American National Standard Methods for the Calculation of the Articulation Index, © ANSI. This Standard has been withdrawn and replaced by, ANSI S3.5-1997 American National Standard Methods for Calculation of the Speech Intelligibility Index, which is available from the Acoustical Society of America, 3 Pinelawn Road, Suite 114E, Melville, NY 11747 <http://asa.aip.org>.

ANSI (1997). ANSI S3.5, Adapted from "ANSI S3.5-1997 American National Standard Methods for Calculation of the Speech Intelligibility Index." © 1998, with the permission of the Acoustical Society of America, 35 Pinelawn Road, Suite 114E, Melville, NY 11747.

ANSI (2002). ANSI S12.60, American National Standard Acoustical Performance Criteria, Design Requirements, and Guidlines for Schools, © 2002, with the permission of the Acoustical Society of America, 35 Pinelawn Road, Suite 114E, Melville, NY 11747

ASHRAE (1980). ASHRAE Handbook & Product Directory 1980 Systems. © 1980 Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., www.ashrae.org.

ASHRAE (1987). 1987 ASHRAE Handbook - HVAC Systems and Applications. © Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, www.ashrae.org.

ASHRAE (1995). 1995 ASHRAE Handbook, HVAC Applications. © 1995 Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., www.ashrae.org.

Askenfeld (1986). A. Askenfeld, "Stage floors and risers–supporting resonant bodies or sound traps?," Acoustics for Choir and Orchestra, Stockholm, Sweden: Royal Academy of Music.
ASTM (1997). E336-97. Standard Recommended Practice for Measurement of Airborne Sound Insulation in Buildings, 1997.
ASTM (1999). E413-87. Standard classification for rating sound insulation, 1999.
ASTM (1997). E1007-97. Standard test method for field measurement of tapping machine impact sound transmission through floor-ceiling assemblies and associated support structures, 1997.
ASTM (1966). Annual Book of ASTM Standards C423, Sound Absorption of Acoustical Materials in Reverberation Rooms. Philadelphia, PA: American Society for Testing and Materials, 1966.
ASTM (1970). Annual Book of ASTM Standards E90, Standard Method for Laboratory Measurement of Airborne Sound Transmission Loss of Building Partitions. Philadelphia, PA: American Society for Testing and Materials, 1970.
ASTM (1971). Annual Book of ASTM Standards E336, Measurement of Airborne Sound Insulation in Buildings. Philadelphia, PA: American Society for Testing and Materials, 1971.
ASTM (1973). Annual Book of ASTM Standards E492, Laboratory Measurement of Impact Sound Transmission Through Floor Ceiling Assemblies Using the Tapping Machine. Philadelphia, PA: American Society for Testing and Materials, 1973.
ASTM (1986). E1110-86. "Standard Classification for Determination of Articulation Class." Philadelphia, PA: American Society for Testing and Materials, 1986.
ASTM (1989). Annual Book of ASTM Standards E989, "Standard Classification for Determining Impact Insulation Class (IIC)." Philadelphia, PA: American Society for Testing and Materials, 1989.
ASTM (2000). E1408-91. Standard test method for laboratory measurement of the sound transmission loss of door panels and door systems, 2000.
Bagenal (1930). Hope Bagenal, "Bach's Music and Church Acoustics," Journal of the Royal Institute of British Architects 37, no. 5, Jan. 1930.
Bagenal (1931). H. Bagenal and A. Wood, Planning for Good Acoustics. Methuen, London, 1931.
Baker (1970). Optica Acta 18, (2) (1971). Special Issue of Image Evaluation by Means of Optical Transfer Functions, 1970.
Ballou (1987). Glen Ballou, ed. Handbook for Sound Engineers, The New Audio Cyclopedia. Indianapolis, IN: Howard W. Sams, 1987.
Barron (1988). M. Barron, "Subjective study of British symphony concert halls." Acustica. 66, 1988.
Barron (1993). M. Barron, "The acoustic survey of British auditoria." Reprinted with permission from J. Acoust. Soc. Am. 93, 2265 (A). © 1993, Acoustical Society of America: Melville, NY.
Barron (1995). M. Barron, "Balconies in Concert Halls." Trondheim, Norway: 15th International Congress on Acoustics, June, 1995.
Barron (1993). Michael Barron, Auditorium Acoustics and Architectural Design. London: Spon Press. Routledge, reprinted 1998.
Barron (1988). M. Barron and L.J. Lee, "Energy Relations in Concert Auditoria. I." Reprinted with permission from J. Acoust. Soc. Am., 84. © 1988, Acoustical Society of America: Melville, NY.
Barry (1978). T.M. Barry and J.A. Reagan, FHWA Highway Noise Prediction Model. U.S. Department of Transportation, Report No. FHWA-RD-77-108, 1978.
Benedetto (1985). , Giuliana Benedetto and Renato Spagnolo, "Reverberation time in enclosures: The surface reflection law and the dependence of the absorption coefficient on the angle of incidence." Reprinted with permission from J. Acoust. Soc. Am., 77(4), © 1985, Acoustical Society of America: Melville, NY.
Beranek (1947). L.L. Beranek, "The design of speech communication systems." Proceedings of the Institute of Radio Engineers, vol. 35, 1947.
Beranek (1949). L.L. Beranek, Acoustic Measurements. New York, NY: John Wiley & Sons, 1949.
Beranek (1954). Leo L. Beranek, Acoustics. New York, NY: McGraw-Hill. Reprinted (1954) New York, NY: American Institute of Physics. Courtesy Leo L. Beranek.
Beranek (1957). Leo L. Beranek, "Revised criteria for noise in buildings." Noise Control, Jan. 1957.
Beranek (1962). Leo L. Beranek, Music, Acoustics & Architecture. Original Edition, Cambridge, MA: Courtesy Leo L. Beranek, 1962.
Beranek (1971). L.L Beranek, ed., Noise and Vibration Control. Cambridge, MA: Courtesy Leo L. Beranek, 1971.
Beranek (1979) (Original Edition 1962). Leo L. Beranek, Music, Acoustics & Architecture. Cambridge, MA: Courtesy Leo L. Beranek, 1979.
Beranek (1989). Leo L. Beranek, "Balanced noise-criterion (NCB) curves." Reprinted with permission from J. Acoust. Soc. Am. vol. 86, No. 2, © 1986, Acoustical Society of America: Melville, NY, Aug. 1989.

Beranek (1992). Leo L. Beranek, "Concert Hall Acoustics." Reprinted with permission from J. Acoust. Soc. Am. vol. 92, No. 1, © 1992, Acoustical Society of America: Melville, NY, July 1992.

Beranek (1994). L.L. Beranek, "The acoustical design of concert halls." Reprinted with permission from J. Acoust. Soc. Am., 1(1), © 1994, Acoustical Society of America: Melville, NY.

Beranek (1996). Leo Beranek, How They Sound, Concert and Opera Halls. © 1996, Woodbury, NY: Acoustical Society of America.

Beranek (1998). Leo L. Beranek and Takayuki Hidaka, "Sound absorption in concert halls by seats, occupied and unoccupied, and by the hall's interior surfaces." Reprinted with permission from J. Acoust. Soc. Am., vol. 104, no. 6. © 1998, Acoustical Society of America: Melville, NY, Dec. 1998.

Beranek (1946). Leo L. Beranek and Harvey P. Sleeper, "The Design and Construction of Anechoic Sound Chambers," J. Acoust. Soc. Am., vol. 18, no. 1, pp. 140–150. © 1998, Acoustical Society of America: Melville, NY.

Beranek (1992). Leo L. Beranek and István L. Vér, Noise and Vibration Control Engineering Principles and Applications. © 1992, New York, NY: this material is used by permission of John Wiley & Sons, Inc.

Berkow (2001). Sam Berkow, Private communication, 2001.

Berlin (1985). E.P. Berlin, Jr., "Efficiency Considerations in Image Synthesis," Siggraph Course Notes, vol. 11, July 1985.

Biering (1983). H. Biering and O.Z. Pedersen, Technical Review, No. 1. System Analysis and Time Delay Spectrometry (Part 1). Marlborough, MA: Bruel & Kjaer Instruments, 1983.

Bistafa (2000). Sylvio R. Bistafa and John S. Bradley. "Reverberation time and maximum background-noise level for classrooms from a comparative study of speech intelligibility metrics." J. Acoust. Soc. Am., vol. 107, no. 2, © 2000, Acoustical Society of America: Melville, NY, Feb. 2000.

Bistafa (2000). Sylvio R. Bistafa and John S. Bradley, "Revisiting Algorihms for Predicing the Articulation Loss of Consonants Al_{cons}," Journal of the Audio Engineering Society, vol. 48, No. 6, Fig. 2, p. 537, and Figure 4, p. 539. New York, NY: Audio Engineering Society, June 2000.

Blauert (1983). Jens Blauert, Spatial Hearing, The Psychophysics of Human Sound Localization. Cambridge, MA: The MIT Press, 1983.

Blazier (1980). W.E. Blazier, Jr., "Revised Noise Criteria for Application in the Acoustical Design and Rating of HVAC Systems," Noise Control Engineering Journal, 16, 64–73.

Blazier (1981). W.E. Blazier, Jr., ed, "Noise Rating of Variable Air-Volume Terminal Devices." ASHRAE Transactions 87(1), Pt. 1, 1981.

Blazier (1994). Warren E. Blazier and Russell B. DuPree, "Investigation of low-frequency footfall noise in wood-frame, multifamily building construction." Reprinted with permission from J. Acoust. Soc. Am., vol. 96, no. 3, © 1994, Acoustical Society of America: Melville, NY, Sept. 1994.

Bolt (1946). R.H. Bolt, "Note on Normal Frequency Statistics for Rectangular Rooms." Reprinted with permission from J. Acoust. Soc. Am., Vol. XIX, no. 1, p. 130, © 1946, Acoustical Society of America: Melville, NY, July 1946.

Bradley (1986). J.S. Bradley, "Predictors of speech intelligibility in rooms." Reprinted with permission from J. Acoust. Soc. Am., 80(3), © 1986, Acoustical Society of America: Melville, NY, Sept. 1986.

Bradley (1991). J.S. Bradley, "A comparison of three classical concert halls." Reprinted with permission from J. Acoust. Soc. Am., 89, No. 3, © Mar. 1991, Acoustical Society of America: Melville, NY, Mar. 1991.

Bradley (1991). J.S. Bradley, "Some further investigations of the seat dip effect." Reprinted with permission from J. Acoust. Soc. Am., 90, © 1991, Acoustical Society of America: Melville, NY.

Bradley (1997). J.S. Bradley, "Sound Absorption of Gypsum Board Cavity Walls." New York, NY: Journal of the Audio Engineering Society, vol. 45, No. 4, Figure 1, p. 255. New York, NY: Audio Engineering Society, Apr. 1997.

Bradley (1998). J.S. Bradley, "Relationships amoung measures of speech intelligibility in rooms." Journal of the Audio Engineering Society, vol. 46. New York, NY: Audio Engineering Society, 1998.

Breton (1989). Gaelle Breton, Theaters. New York, NY: Princeton Architectural Press, Paris, France: Editions du Moniteur, 1989.

Bronsdon (1999). R.L. Bronsdon and H. Forschner, "A Propagation Model Based on Gaussian Beams that Account for Wind and Temperature Inversions," J. Noise Control Eng, 47(5) Sept.-Oct., 1999.

Brüel and Kjaer (1974). Instructions and Applications, 1974. "One-Inch Condenser Microphone Types 4144, 4545, 4146, 4161." Norcross, GA: Brüel and Kjaer, 1974.

Brüel and Kjaer (1978). Technical Review No. 4, "Reverberation Process at Low Frequencies," by Holger Larsen. Norcross, GA: Bruel and Kjaer.

Brüel and Kjaer (1979). J.R.Hassall and K. Zaveri, Acoustic Noise Measurements. 4th ed. Naerum, Denmark: Brüel & Kjaer Instruments, Jan. 1979.

Brüel and Kjaer (1986). Instruction Manual, Modular Precision Sound Level Meter, "Modular Precision Sound Level Meter Type 2231 plus Integrating SLM Module BZ 7100." Norcross, GA: Brüel and Kjaer, Feb. 1986.
Burkat (1998). Jacket notes from a classical CD.
Burris-Meyer (1975). Harold Burris-Meyer and Edward C. Cole, Theatres and Auditoriums. Huntington, NY: Robert E. Krieger Publishing Co., 1975.
Burroughs (1974). Lou Burroughs, Microphones: Design and Application, First Edition. Champaign, IL: Sagamore Publishing Co, 1974.
Cahill (1995). Thomas Cahill, How the Irish Saved Civilization: The Untold Story of Ireland's Heroic Role from the Fall of Rome to the Rise of Medieval Europe (Hinges of History, vol. 1) Anchor, 1995.
California Heat Pump (1982). Water Source Heat Pumps, CHP-A-059-82. Santa Monica, CA: California Heat Pump Co, 1982.
California, State of (1974). "California Noise Insulation Standards," California Administrative Code, Title 25, Chapter 1, Subchapter 1. © 1974 by The Bureau of National Affairs.
California, State of (1990). California Noise Insulation Standards, State Building Code (Part 2, Title 24, CCR), Appendix Chapter 35, Sound Transmission Control, 1990.
California, State of (1981). Office of Noise Control , Catalog of STC and IIC Ratings for Wall and Floor/Ceiling Assemblies. Berkeley, CA: Dept. of Health Services. Office of Noise Control, 1981.
Chanaud (1983). Robert C. Chanaud, Sound Conditioning Manual for Facilities Managers, Architects, & Design Professionals. Norcross, GA: Dyna-systems, 1983.
Chanaud (2000). Robert C. Chanaud, Private communication, 2000.
Chanaud (2002). Robert C. Chanaud. Sound Conditioning Manual, Designing Good Acoustics into the Office. Dynasound, Inc, 2002.
Cheever (2000). E. Cheever, Swarthmore College Internet. Swarthmore, PA, 2000.
Chu (2002). W.T. Chu and A.C.C. Warnock, IRC Research Report 104, "Detailed Directivity of Sound Fields around Human Talkers." Ontario, Canada: National Research Council Canada, Dec. 2002.
CIPIC (2001). "Spatial Sound: An Introduction." Center for Image Processing and Integrated Computing (CIPIC), University of California, Davis Interface Laboratory. http://interface.cipic.ucdavis.edu/CIL_tutorial/3D_psych/elev.htm, 2001.
Clark (1983). David Clark, "Measuring Audible Effects of Time Delays in Listening Rooms." AES, 74th Convention, New York, NY: AES, Oct. 1983.
Colloms (1980). Martin Colloms, High Performance Loudspeakers. © 1980, New York, NY: this material is used by permission of John Wiley & Sons, Inc.
Community (1991). Todd Rockwell, "To Smooth or Not to Smooth," Community Data Library, Volume One. Chester, PA: Community Professional Loudspeakers:
CONC (1981). California of STC and IIC Ratings for Wall and Floor/Ceiling Assemblies, Office of Noise Control, California Dept. of Health Services, 1981.
Craik (1996). R.J.M. Craik, Sound Transmission Through Buildings Using Statistical Energy Analysis. United Kingdom: Gower, Aldershot, 1996.
Craik et al. (1997). R.J.M. Craik, T.R.T. Nightingale and J.A. Steel, "Sound transmission through a double leaf partition with edge flanking." Reprinted with permission from National Research Council of Canada, Ontario, Canada and J. Acoust. Soc. Am., 101(2). © 1997, Acoustical Society of America: Melville, NY, Feb. 1997.
Cremer (1942). Theorie der Schalldämmung dünner Wände bei Schrägen einfall. Akust. Ztg, 1942.
Cremer (1952). L. Cremer, "Theorie des Klopfshalles bei Decken mit Schwimmenden Estrick." Acustica, 2, 1952.
Cremer et al. (1973). L. Cremer, M. Heckl, and E.E Ungar, Structure-Borne Sound, p. 244, Fig. IV/2 and p. 302, Fig. IV/20. © 1973, Berlin ·Heidelberg: Springer-Verlag.
Cremer (1982). Lothar Cremer and Helmut A. Müller, Principles and Applications of Room Acoustics, vol. 1. English translation with additions by T.J. Schultz, New York, NY: New York, NY: Applied Science Publishers, 1982.
Cremer (1982). Lothar Cremer and Helmut A. Müller, Principles and Applications of Room Acoustics, vol. 2. New York, NY: Applied Science Publishers, 1982.
Crown (1993). Macro-Tech 5000VZ Professional Power Amplifier Owner's Manual. © 1993, Elkhart, IN: Crown International.
Crown (1995) Micro-Tech 600/1200/2400 Professional Power Amplifiers Owner's Manual. © 1995, Elkhart, IN: Crown International.
D'Antonio (1987). Peter D'Antonio, Expand Your Acoustical Palette with the RPG Diffusor System. RPG brochure, © 1987. Marlboro, MD: RPG Diffusor Systems, Inc.

D'Antonio (1995). P. D'Antonio, "Two Decades of Diffusor Design and Development," 99th AES Convention, Invited Lecture, Preprint 4114 (P-1), New York, NY: Audio Engineering Society, Oct. 1995.

Davis (1926). A.H. Davis, Phys. Soc. Proc. 38, 1926.

Davis (1979). C. Davis and D. Davis. "(LEDE™) Live End-Dead End Control Room Acoustics . . . (TDS) Time Delay Spectrometry . . . (PZM™) Pressure Zone Microphones," Record./Eng. Prod, 1979.

Davis (1957). D.D. Davis, Jr. "Acoustical Filters and Mufflers," in C.M. Harris (ed.), Handbook of Noise Control, Chap. 21, McGraw-Hill, 1957.

Davis (1969). D. Davis, "Analyzing Loudspeaker Locations for Sound Reinforcement Systems." Journal of the Audio Engineering Society, vol. 17. New York, NY: Audio Engineering Society, Dec. 1969.

Davis (1978). Don Davis and Carolyn Davis. Sound System Engineering.First Edition, Fourth Printing, Indianapolis, IN: Howard W. Sams, 1978.

Davis (1987). Don Davis and Carolyn Davis, Sound System Engineering. 2nd ed. Indianapolis, IN: Howard W. Sams, 1987.

Davis et al. (1954). D.D. Davis, G.M. Stokes, D. Moore, and G.L. Stevens, "Theoretical and Experimental Investigation of Mufflers with Comments on Engine-Exhaust Muffler Design," Nat. Advis. Comm. Aeronaut. Rep. 1192, Washington, D.C, 1954.

de Boer (1940a). K. de Boer, Stereofonische Geluidsweergave (Stereophonic sound reproduction). Dissertation, Institute of Technology, Delf, 1940.

de Boer (1940b). K. de Boer, Plastische Klangwiedergabe (Three-dimensional sound reproduction). Philips Tech. Rev. 5, 1940.

Delany (1969). M.E. Delany and E.N. Bazley, "Acoustical properties of fibrous absorbent materials." Applied Acoustics, 3, 1969.

Delsasso (1967). Leo Delsasso, Private Communication, 1967.

DeVac Corp. (1997). "DeVAC: Master Sound Test Loss Recap Chart, 1-97." Carmichael, CA: DeVAC of California, 1997.

Dimarogonas (1990). A.D. Dimarogonas, "The Origins of Vibration Theory." Journal of Sound and Vibration, 140(2). New York, NY: Academic Press Limited, 1990.

Doelle (1972). Leslie L. Doelle, Environmental Acoustics. © 1972, New York, NY: Reprinted with permission from McGraw-Hill Book Company.

Dunlavy (2001). Internet www.soundstage.com/rev.equip/dunlavy_sciva.htm, 2001.

Egan (1988). M. David Egan, Architectural Acoustics. © 1988, New York, NY: Reprinted with permission from McGraw-Hill Book Company.

Electrovoice (1970). D.B. (Don) Keele, Jr., John Gilliom, and Ray Newman, ca. 1973, see also prosoundweb.com Jim Long interview 2004.

EPA (1974). Information on Levels of Environmental Noise Requisite to Protect Public Health and Welfare with an Adequate Margin of Safety, 550/9-74-004. Washington, D.C.: U.S. Environmental Protection Agency, Office of Noise Abatement and Control, March 1974.

Evans (1972). L.B. Evans, H.E. Bass, and L.C. Sutherland, "Atmospheric Absorption of Sound: Theoretical Predictions," J. Acoust. Soc. Am. 51 © 1972,: Melville, NY, Acoustical Society of America.

Everest (1994). F. Alton Everest, The Master Handbook of Acoustics. 3rd ed. © 1994, New York, NY: Reprinted with permission from McGraw-Hill Book Company.

Eyring (1930). Carl F. Eyring, "Reverberation Time in 'Dead' Rooms." Reprinted with permission from J. Acoust. Soc. Am. © 1930, Acoustical Society of America: Melville, NY, Jan. 1930.

Fahy (1985). Frank Fahy, Sound and Structural Vibration, Radiation, Transmission and Response. London, England: Academic Press an imprint of Elsevier Science, 1985.

Farina (2000). Angelo Farina, "Measurement of the surface scattering coefficient: comparison of the Mommertz/Vorlander approach with the new Wave Field Synthesis method." Liverpool (GB): International Symposium on Surface Diffusion in Room Acoustics, 16 April 2000.

Farina (2000). Angelo Farina, "Validation of the numerical simulation of the scattered sound field with a geometrical pyramid tracing approach." Liverpool (GB): ACOUSTICS 2000 IOA Conference, 17–18 April 2000.

Farina (2002). Angelo Farina, "Verification of the Accuracy of the Pyramid Tracing Algorithm by Comparison with Experimental Measurements of Objective Acoustic Parameters." Trondheim, Norway, ICA95 (International Conference on Acoustics), 26–30 June 1995.

Feynman (1989). Richard Feynman, Robert B. Leighton, and Matthew L. Sands, The Feynman Lectures on Physics: Commemorative Issue, Vol. 1, © 1989 California Institute of Technology, Reading, MA: Addison-Wesley Publishing.

Farmer (2001). William Farmer, Private Communication, 2001.
Fierstein (1979). Alan Fierstein, "Optimizing Control Room Reverberation," Recording Engineer/Producer Magazine, Aug. 1979.
Finne (1987). P. Finne, "Beregningsmodel for diffraktion. Akustiske anvendelser af Fresnel-Kirchhoffs approksimation." Msc thesis, The Acoustics Laboratory, Lyngby, 1987.
Flanagan (1972). J.L. Flanagan, Speech Analysis, Synthesis and Perception, 2nd ed. New York, NY: Springer-Verlag, 1972.
Fletcher (1963). Banister Fletcher, A History of Architecture on the Comparative Method. 17th ed. New York, NY: Charles Scribner's Sons, 1963.
Fletcher (1921). Harvey Fletcher, "An empirical theory of telephone quality." AT&T Internal Memorandum, 101(6). Case 211031, Report 21839, Oct 1921. Provo , UT: Brigham Young University, 1921.
Fletcher (1953). Harvey Fletcher, Speech and Hearing in Communication. ed. Jont B. Allen , 1995. Woodbury, NY: Acoustical Society of America: Melville, NY, 1953.
Fletcher (1950). H. Fletcher and R.H. Galt, "The perception of speech and its relation to telephony." Reprinted with permission from J. Acoust. Soc. Am. 22, © 1950, Acoustical Society of America: Melville, NY.
Fletcher (1933). Harvey Fletcher and W.A. Munson, "Loudness, Its Definition, Measurement and Calculation." Reprinted with permission from J. Acoust. Soc. Am. Vol. 5, © 1933, Acoustical Society of America: Melville, NY, Oct. 1933.
Fletcher (1937). Harvey Fletcher and W.A. Munson, "Relation between loudness and masking." J. Acoust. Soc. Am., 9, © 1937, Reprinted with permission from Acoustical Society of America: Melville, NY.
Foreman (1987). Chris Foreman, Handbook for Sound Engineers, The New Audio Cyclopedia edited by Glen Ballou. Indianapolis, IN: Howard W. Sams, 1987.
Forsyth (1985). Michael Forsyth, Buildings for Music. Cambridge, MA: MIT Press, 1985.
Forsyth (1987). Michael Forsyth, Auditoria. © 1987 Michael Forsyth, New York, NY: Van Nostrand Reinhold Co.
Franzoni (2001). Linda P. Franzoni, "A power conservation approach to predict the spatial variation of the cross-sectionally averaged mean-square pressure in reverberant enclosures." Reprinted with permission from J. Acoust. Soc. Am., 110 (6). © 2001, Acoustical Society of America: Melville, NY, Dec. 2001.
Franzoni (1999). Linda P. Franzoni and D.S. Labrozzi, "A study of damping effects on spatial distribution and level of reverberant sound in a rectangular acoustic cavity." J. Acoust. Soc. Am., 106(2). © 1999, Acoustical Society of America: Melville, NY.
French (1947). N.R. French and J.C. Steinberg. "Factors governing the intelligibility of speech sounds." J. Acoust. Soc. Am., 19, 90–119 © 1947, Acoustical Society of America: Melville, NY.
Frenette (2000). Jasmin Frenette, "Reducing Artificial Reverberation Algorithm Requirements Using Time-Variant Feedback Delay Networks," a research project, Dec. 2000, Music Engineering Technology. Coral Gables, FL: University of Miami, 2000.
Fry (1988). Alan Fry (ed.) Noise Control in Building Services. Pergamon Press. Sudbury, Suffolk, UK: Reprinted with permission from Sound Research Laboratories Ltd, 1988.
Funkhouser et al. (1996). Thomas Funkhouser, Ingrid Carlbom, Gary Elko, Gopal Pingalli, Mohan Sondhi, and Jim West, Bell Laboratories, "A Beam Tracing Approach to Acoustic Modeling for Interactive Virtual Environments," http://www.cs.princeton.edu/%7Efunkacoustics.html, 1996.
Gade (1989). A.C. Gade, "Investigations of Musicians' Room Acoustic Conditions in Concert Halls. II: Field Experiments and Synthesis of Results." Acustica, 69, 1989a/b.
Gade (1982). S. Gade, Technical Review, No. 3, Sound Intensity (Theory). Marlborough, MA: Brüel & Kjaer Instruments, 1982.
Garity (1941). Wm. E. Garity and J.N.A. Hawkins, "Fantasound." Journal of the Society of Motion Picture Engineers, Aug. 1941.
Garnier (1880). C. Garnier, Le nouveau Opéra de Paris, 2 vols. Ducher et Cie, Paris, 1876–1880.
Geerdes et al. (1971). H.P. Geerdes, B.G. Watters and P.W. Hirtle, "Sound Isolation for Music Activities." Washington DC: oral paper at 81st meeting of the Acoustical Society of America. Reprinted with permission from J. Acoust. Soc. Am. © 1971, Acoustical Society of America: Melville, NY, 21 April 1971.
Gibbs (1989). B.M. Gibbs and Y. Balilah, "The Measurement of Sound Transmission and Directivity of Holes by an Impulse Method." Reprinted from Journal of Sound and Vibration, Vol. 133(1), pp. 151–162. © 1989, New York, NY: Academic Press Limited with permission from Elsevier Science.
Ginn (1978). K.B. Ginn, Architectural Acoustics. Indianapolis, IN: B & K Instruments, 1978.
Glassner (1989). Andrew S. Glassner, ed., An Introduction to Ray Tracing. San Diego, CA: Academic Press, 1989.

Gordon (1978) J.E. Gordon, Structures. New York, NY & London, England: Plenum Press, 1978.
Graham (1975). J.B. Graham, "Fan Selection and Installation." ASHRAE Symposium Paper as given in ASHRAE, 1987. Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, June 1975.
Green (1982). David W. Green and Cameron W. Sherry, "Sound transmission loss of gypsum wallboard partitions, Report #1, Unfilled steel stud partitions." Reprinted with permission from J. Acoust. Soc. Am., 71(1), © 1982, Acoustical Society of America: Mellville, NY, Jan. 1982.
Griesinger (1991). David Griesinger, "Improving Room Acoustics Through Time-Variant Synthetic Reverberation," 90th Convention of the AES, Paris, 1991."
Grout (1996). Donald Jay Grout and Claude V. Palisca, A History of Western Music, 5th ed. © 1996, New York, NY: W.W. Norton & Co., Inc., 1996.
Grozier (2002). http://www.grozier.com/scattering.htm, 2002.
Guillen (1995). Michael Guillen, Five Equations that Changed the World: The Power and Poetry of Mathematics. Hyperion. ISBM 078 688 1879.
Gutenberg (1942). B. Gutenberg, "Propagation of Sound Waves in the Atmosphere," J. Acoust. Soc. Am., 14, © 1942, Acoustical Society of America: Melville, NY.
Haan (1995). C. H. Haan and F.R. Fricke. "Musician and music critic responses to concert hall acoustics." Proceedings of 15th International Congress on Acoustics, Trondheim, Norway, June 1995.
Haan (1997). C. Haan and F.R. Fricke. "An evaluation of the importance of surface diffusivity in concert halls," Applied Acoustics, vol. 51, no. 1, 1997.
Haas (1951). Helmut Haas. "The influence of a single echo on the audibility of speech," Journal of the Audio Engineering Society, 20, 2 (1972). (This is an English translation from the German by Dr. Ingr. K.P.R. Ehrenberg of Haas' original paper in Acustica, 1.2 (1951).
Haines (1989). Eric Haines, "Essential Ray Tracing Algorithms." Chapter 2 of An Introduction to Ray Tracing. ed. Andrew S. Glassner. San Diego, CA: Academic Press, 1989.
Haines (2002) Eric Haines, Ray Tracing News (RTNews) web site, 2002.
Halliday (1966). David Halliday and Robert Resnick, Physics Part I. © 1966, New York, NY: this material is used by permission of John Wiley & Sons, Inc.
Hamet (1996). J.F. Hamet, Personal Communication, 1996.
Hamet (1997). J.F. Hamet, INRETS. Cedex, France: unpublished, 1997.
Hanrahan (1989). Pat Hanrahan, "A Survey of Ray-Surface Intersection Algorithms," An Introduction to Ray Tracing. © 1989. San Diego, CA: Academic Press.
Harris (1957). Cyril M. Harris (ed.), Handbook of Noise Control. © 1957, Reprinted with permission from McGraw-Hill Book Company.
Harris (1966). C.M. Harris, "Absorption of Sound in Air versus Humidity and Temperature." Reprinted with permission from J. Acoust. Soc. Am., 40, © 1966, Acoustical Society of America: Melville, NY.
Harris (1991). C.M. Harris, ed., Handbook of Acoustical Measurements and Noise Control. 3rd ed. © 1991, New York, NY: Reprinted with permission from McGraw-Hill Book Company.
Harris (1994). Cyril M. Harris, Noise Control in Buildings, A Practical Guide for Architects and Engineers. © 1994, New York, NY: McGraw-Hill Book Company.
Harris (1961). Cyril M. Harris and Charles E. Crede. Shock & Vibration Handbook. 2nd ed. © 1961, New York, NY: Reprinted with permission from McGraw-Hill Book Company.
Harris (1992). D.A. Harris, "Quiet Lightweight Floor Systems," Bay Village, OH: Sound and Vibration Magazine, July 1992.
Hassall (1979). J.R.Hassall and K. Zaveri, Acoustic Noise Measurements. 4th ed. Naerum, Denmark: Brüel & Kjaer Instruments, Jan. 1979.
Hawkes (1971). R.J. Hawkes and H. Douglas, "Subjective acoustic experience in concert auditoria." Acustica, 24, 1971.
Hayashi et al. (1978). C. Hayashi, S. Kondo, and H. Kodama, "Psychological Assessment of Aircraft Noise Index." Reprinted with permission from J. Acoust. Soc. Am., 63, © 1978, Acoustical Society of America: Melville, NY.
Helmholtz (1877). H.L.F. Helmholtz, Lehre von den Tonempfindungen. Braunschweig: Vieweg.
Hemming (1988). Roy Hemming, Discovering Great Music. New York, NY: Newmarket Press, 1988.
Henricksen (1980). Cliff Henricksen, Technical Letter No. 237, "Directivity Response of Single Direct-Radiator Loudspeakers in Enclosures." Oklahoma City, OK: Altec Lansing Corp. Reprinted with permission from Telex Communications, 1980.
Hessler (1992). George F. Hessler, Jr. and Ben H. Sharp, "Predicting Sound Pressure Levels in the Near and Far Field from Radiating Partitions." Invited Paper. New Orleans, LA: Acoustical Society of America

Conference. Reprinted with permission from Acoustical Society of America: © 1992, Melville, NY, Nov. 1992.
Hill (1990). F.S. Hill, Jr.. Computer Graphics. © 1990, New York, NY: Macmillan Publishing Co.
Hindley (1965). Goeffrey Hindley, Larousse Encyclopedia of Music. Englewood Cliffs, NJ: Prentice Hall, 1965.
Hodgson (1998). M.R. Hodgson, "Experimental evaluation of simplified methods for predicting sound propagation in industrial workrooms," J. Acoust. Soc. Am. 103(4), 1998.
Hoover (1961). R.M. Hoover, "Tree Zones as Barriers for the Control of Noise due to Aircraft Operations," Bolt Beranek and Newman Inc. Rept. 844, February, 1961.
Hopkins (1948). Hopkins and Stryker, "A Proposed Loudness-Efficiency Rating for Loudspeakers and the Determination of System Power Requirements for Enclosures." reprinted from the March 1948 Edition of the Proceedings of the I.R.E., 1948.
Houtgast (1973). T.Houtgast and H.J.M. Steeneken, "The Modulation Transfer Function in Room Acoustics as a Predictor of Speech Intelligibility." Acustica 28, 66–73, 1973.
Houtgast et al. (1980). T. Houtgast and H.J.M. Steeneken, "A Physical Method for Measuring Speech Transmission Quality." Journal of the Audio Engineering Society, vol. 67, no. 1. New York, NY: Audio Engineering Society, 1980.
Houtgast (1983). T. Houtgast and H.J.M. Steeneken, "The Temporal Envelope Spectrum of Speech and it Significance in Room Acoustics." Proceedings of the Eleventh International Congress on Acoustics. vol. 7. GALF, Paris, France, 1983.
Houtgast (1985). T. Houtgast and H.J.M. Steeneken, Technical Review No. 3, The Modulation Transfer Function in Room Acoustics. Marlborough, MA: Brüel & Kjaer Instruments, 1985.
Houtgast (1980). T. Houtgast, H.J.M. Steeneken, and R. Plomp, "Predicting Speech Intelligibility in Rooms from the Modulation Transfer Function." Acustica, 11, 195–200, 1980.
HUD (1979). 24 CFR 51B - Environmental Criteria and Standards, U.S. Department of Housing and Urban Development, 1979.
Hudspeth (1994). A.J. Hudspeth and Vladislav S. Markin, "The Ear's Gears: Mechanoelectrical Transduction by Hair Cells." Reprinted with permission from Physics Today, © 1994, Woodbury, NY: American Institute of Physics.
Hunt (1964). Frederick V. Hunt, "Introduction to Dover Edition," Collected Papers on Acoustics by Wallace Clement Sabine. © 1964, New York, NY: Dover Publications.
Hunt (1978). F.V. Hunt, Origins in Acoustics. New Haven, CT & London, England: Yale University Press, 1978.
IAC Corp. (1989). Engineered Noise Control for Air-Handling Systems, Duct Silencers, Bulletin 1.0301.4, Applications Manual. Bronx, NY: © 1989, Industrial Acoustics Company.
Ingard (1994). K.U. Ingard, Notes on Sound Absorption Technology. ver. 94–2, © 1994, Kittery Point, ME: K.U. Ingard, 1994.
ISO (1999). 3741. "Acoustics: determination of sound power levels of noise sources. Precision methods for reverberation rooms." International Organization for Standardization (ISO), 1999.
ISO (1970). 140/III, Field and Laboratory Measurements of Airborne and Impact Sound Transmission. 3rd printing. Genève, Switzerland: International Organization for Standardization (ISO), 1970.
ISO (1998). ISO 140/IV. "Acoustics: measurement of sound insulation in buildings and of building elements. Part IV: Field measurements of airborne sound insulation between rooms. International Organization for Standardization (ISO), 1998.
ISO (1982, 1996). ISO/R 717. "Acoustics: rating of sound insulation in buildings and of building elements. Part 1: Airborne sound insulation in buildings and of interior building elements. Part 2: Impact sound insulation. Part 3: Airborne sound insulation of facade elements and facades." International Organization for Standardization (ISO), 1982, 1996.
ISO (1988). DIS 7779 (S12), "Acoustic measurement of airborne noise emitted by computer and business equipment." International Organization for Standardization (ISO), 1988.
ISO (1990). DIS 9613-1, "Acoustics - Attenuation of sound during propagation outdoors - Part 1: Method of calculation of the attenuation of sound by atmospheric absorption." Draft International Standard, International Organization for Standardization (ISO), 1990.
ISO (1994). 9613-2.2, "Acoustics - Attenuation of sound during propagation outdoors. Part 2: General method of calculation." Draft International Standard, International Organization for Standardization (ISO), 1994.
ISO (1998). 140/6. "Acoustics: measurement of sound insulation in buildings and of building elements. Part 6: Laboratory measurements of impact sound insulation of floors." International Organization for Standardization (ISO), 1998.

ISO (1998). 140/7. “Acoustics: measurement of sound insulaton in buildings and of building elements. Part 7: Field measurements of impact sound insulation of floors.” International Organization for Standardization (ISO), 1998.

Izenour (1977). George C. Izenour, Theater Design. New York, NY: McGraw-Hill. The George C. Izenour Archive at Penn State University, 1977.

Jacob (1985). K.D. Jacob, “Subjective and Predictive Measures of Speech Intelligibility - The Role of Loudspeaker Directivity.” Journal of the Audio Engineering Society, vol. 33. New York, NY: Audio Engineering Society, Dec. 1985.

JBL (1998). “JBL 2360A, 2365A, 2366A Bi-Radial® Constant Coverage Horns.” Northridge, CA: JBL Professional, Jan. 1998.

JBL Professional (1997). Catalog cut “4895 Three-Way High Directivity Horn Loaded Positionable Array Element.” Northridge, CA JBL Professional, Sept. 1997.

JBL Professional (2000). “Speech intelligibility - A JBL Professional Technical Note.” Technical Notes, Vol 1, No 26. Northridge, CA: JBL Professional, Aug. 2000.

JBL (1988) JBL and Marshall Long, “JBL and the Los Angeles Sports Arena,” A JBL Installation, Northridge, CA: JBL Incorporated, Sept. 1988.

Johnson (2002). Russell Johnson, ARTEC, Private Communication, 2002.

Jones (1984). R.S. Jones, Noise and Vibration Control in Buildings. © 1984, New York, NY: Reprinted with permission from McGraw-Hill Book Company.

Josse (1964). R. Josse and C. Lamure, Acustica 14, 266–280, 1964.

Josse (1969). R. Josse and C. Drouin, “Etude des impacts lourds a l’interieur des bâtiments d’habitation,” Rapport de fin d’étude, Centre Scientifique et Technique du Bâtiment, Paris, February 1, 1969, DS No. 1, 1.24.69.

Junger (1972). Miguel C. Junger and David Feit, Sound, Structures, and Their Interaction. 2nd ed. Cambridge, MA: MIT Press, 1972.

Keele (1983). Don Keele, Private communication, 1983.

Kinetics (1990). ELIM Shores, “Report of: Field - Impact Insulation Class (F-IIC).” St. Paul, MN: Twin City Testing. Kinetics Corporation Test Data, Dublin, OH, 1990.

Kinetics (1996) Testing performed at the National Research Council Canada, Ottawa, Ontario. Kinetics Corporation Test Data, Dublin, OH, 1996.

Kinsler (1962). L.E. Kinsler and A.R. Frey, Fundamentals of Acoustics, 2nd ed. copyright © 1962, New York, NY: this material is used by permission of John Wiley & Sons, Inc.

Kinsler et al. (1982). L.E. Kinsler, A.R. Frey, A.B. Coppens, and J.V. Sanders Fundamentals of Acoustics, 3rd ed. copyright © 1982, New York, NY: this material is used by permission of John Wiley & Sons, Inc.

Kishinaga et al. (1979). Shinji Kishinaga, Yashushi Shimizu, Shigeo Ando, and Kiminori Yamaguchi, “On The Room Acoustic Design of Listening Rooms.” Audio Eng. Soc 64th Convention, Preprint #1524. New York, NY: Audio Engineering Society, Nov. 1979.

Klark-Teknik. The Audio System Designer Technical Reference . © Klark-Teknik. Buchanan, MI: Mark IV Pro Audio Group.

Klein (1971). W. Klein, “Articulation loss of consonants as a basis for the design and judgment of sound reinforcement systems.” Cologne: AES 1st European Convention, 1971.

Klepper (1999). David L. Klepper, “A different angle.” S&VC , vol. 17, no. 1, Emeryville, CA, Jan. 1999.

Klepper et al. (1980). David L. Klepper, William J. Cavanaugh, and L. Gerald Marshall, “Noise Control in Music Teaching Facilities.” Noise Control Engineering Journal, Vol. 15 No. 2, Sept/Oct 1980.

Knudsen (1932). V.O. Knudsen, Architectural Acoustics. © 1932, New York, NY: this material is used by permission of John Wiley & Sons, Inc.

Knudsen (1970). Vern O. Knudsen, “Model Testing of Auditoriums,” S.1. J. Acoust. Soc. Am. vol. 47, no. 2 (part 1), February 1970. Acoustical Society of America: Melville, NY, 1970.

Knudsen (1965). V.O. Knudsen and L.P. Delsasso, “Acoustics of the Grady Gammage Auditorium, Arizona State University.” 5th Congres International D’Acoustique, Liege 7–14 Septembre 1965.

Knudsen (1950). V.O. Knudsen and C. Harris, Acoustical Designing in Architecture. © 1950, New York, NY: this material is used by permission of John Wiley & Sons, Inc., 190.

Kopec (1990). John Kopec, Private communication, 1990.

Krokstad (1991). A. Krokstad, “Electroacoustic Means of Controlling Auditorium Acoustics.” University of Trondheim report number 408507, The Norwegian Institute of Technology Division of Telecommunications O.S. Bragstads plass 4, N7034 Trondheim-nth, Norway, presented at the 90th Convention, 1991 February 1922, Paris, Audio Engineering Society. (AES), 1991.

Kryter (1970). Karl D. Kryter, The Effects of Noise on Man. Burlington, MA: Academic Press, 1970.

Kryter (1963 & 1964). K.D.Kryter and K.S. Pearsons, "Some effects of spectral content and duration on perceived noise level." Reprinted with permission from J. Acoust. Soc. Am. © 1963 & 1964, Acoustical Society of America: Melville, NY.

Kuhl (1954). W. Kuhl, Acustica, 4, 618–634.

Kurze (1971). U.J. Kurze and G.S. Anderson, "Sound Attenuation by Barriers." Applied Acoustics. vol. 4, 1971.

Kuttruff (1973). Heinrich Kuttruff, Room Acoustics. © Heinrich Kuttruff. London, England: Applied Science Publishers, 1973.

Landon (1995). H.C. Robbins Landon, Haydn in England 1791–1795. London: Thames and Hudson.

Lange (1953). T.H. Lange, Acustica 3, 1953.

Larsen (1978). Holger Larsen, Technical Review No. 4. "Reverberation Process at Low Frequencies." Naerum, Denmark: Brüel & Kjaer, 1978.

Leissa (1969). A.W. Leissa, "Vibration of Plates," NASA SP-160. Washington, D.C.: National Aeronautics and Space Administration, 1969.

Leizer (1966). I.G. Leizer, "Applicability of the Methods of Geometric Acoustics for the Calculation of Sound Reflection from Plane Surfaces." Soviet Physics–Acoustics, Vol. 12, No. 2, Oct.–Dec., 1966. Translated from Akusticheskii Zhurnal, vol. 12, no. 2, April–June, 1966.

Lenzen (1966). K.H. Lenzen, "Vibration of Steel Joist-Concrete Slab Floors." AISC, Engineering Journal, vol. 3, no. 1. Ames, IA: Iowa State University, 1966.

Leonard et al. (1964). R.W. Leonard, L.P. Delsasso, and V.O. Knudsen, "Diffraction of Sound by an Array of Rectangular Reflective Panels." Reprinted with permission from J. Acoust. Soc. Am., vol. 36, no. 12, © 1964, Acoustical Society of America: Melville, NY.

Levine (1948). H. Levine and J. Schwinger, "On the theory of diffraction by an aperture in an infinite screen." Physic. Rev. 74, 1948.

Lindsay (1966). Bruce Lindsay, "The Story of Acoustics." Reprinted with permission from J. Acoust. Soc. Am., 39(4), © 1966, Acoustical Society of America: Melville, NY.

Lochner (1964). J.P.A. Lochner and J.F. Berger, "The Influence of Reflections on Auditorium Acoustics." Journal of Sound and Vibration, 1, 1964.

London (1950). A. London, "Transmission of Reverberant Sound through Double Walls." Reprinted with permission from J. Acoust. Soc. Am., vol. 44, © 1950, Acoustical Society of America: Melville, NY, 1950.

Long (1980). "Design of Housing for Aircraft Noise," Presented to the 100th meeting of the Acoustical Society of America, Los Angeles, CA, 1980.

Long (1983). Marshall Long, "Constant Directivity Horns.": Sound and Video Contractor, Overland Park, KS, 1983.

Long (1987). Marshall Long, "Sound Transmission through Partitions." New York, NY: AES Journal, Jul/Aug. 1987.

Long (1993). Marshall Long, "Acoustical Ray Tracing, " National Science Foundation Phase I Final Report. Arlington, VA: National Science Foundation, 1993.

Long (1999). Marshall Long, "Getting the Message." S&VC, vol. 17, no. 1, Emeryville, CA, Sound & Video Contractor, Jan. 1999.

Long (2000). Marshall Long, "Architectural Acoustics: Theme Park Acoustics, Creativity in Theme Park Acoustics," Newport Beach, CA: Acoustical Society of America, Session 4aAA, Dec. 7, 2000.

Lukas (1975). J.S. Lukas, "Noise and Sleep." Reprinted with permission from J. Acoust. Soc. Am., 58, © 1975, Acoustical Society of America: Melville, NY.

Madsen (1970). E.R Madsen, "The Disclosure of Hidden Information in Sound Recording," a lecture given May 1970, and referenced in The Science of Sound, 2nd ed., 1990, p. 500, Fig. 25.2, written by Thomas D. Rossing. Glenview, IL: Addison-Wesley Publishing, May 1970.

Maekawa (1965). Z. Maekawa, "Noise Reduction by Screens." Memoirs of Faculty of Eng., Kobe, Japan: Kobe Univ., 1965.

Maekawa (1966). Z. Maekawa, "Noise Reduction by Screens of Finite Size." Memoirs of Faculty of Eng., Kobe, Japan: Kobe Univ., 1966.

Maekawa (1977). Z. Maekawa, "Shielding Highway Noise." Noise Control Engineering Journal. Ames, IA: Iowa State University, July–Aug. 1977.

Mankovsky (1971). V.S. Mankovsky, from Acoustics of Studios and Auditoria. by V.S. Mankovsky. Reprinted by permission of Elsevier Ltd, Oxford, UK, 1971.

Marshall (1967). A. Harold Marshall, "A note on the importance of room cross-section in concert halls." Journal of Sound and Vibration. 5, 1967.

Mason Industries (1968). Noise and Vibration Problems and Solutions, Condensed and published in Air Conditioning, Heating & Refrigeration News, 1-29-68. Hauppauge, NY: Mason Industries, 1968.
Mason Industries (1998). "Complete Seismic HVAC Engineering Specifications," SVCS Part 2, SVCS-110-3 Bulletin. Hauppauge, NY: Mason Industries, Sept. 1998.
Maxfield (1947). Maxfield and Albersheim, Bell Laboratories, 1947.
McLachlan (1934). N.W. McLachlan, Loudspeakers. New York, NY: Oxford Press, 1934.
Mechel (1992). Fridolin P. Mechel and István L. Vér, Chapter 8, "Sound-absorbing Materials and Sound Absorbers," Noise and Vibration Control Engineering Principles and Applications by Leo L. Beranek and István L. Vér, © 1980, New York, NY: this material is used by permission of John Wiley & Sons, Inc.
Meyer (1978). J. Meyer, Acoustics and the Performance of Music. Frankfurt/Main: Verlag Dis Musikinstrument, 1978.
Meyer (1952). E. Meyer and G.R. Schodder, Nachr. Akad. Wissensch. Göttingen, Math.-Phys. Kl., No. 6, 31, 1952.
Miller et al. (1951). G. A. Miller, G.A. Heise, and W. Lichten, "The intelligibility of speech as a function of the context of the test materials," J. Exp. Psychol. 41, 1951.
Miller (1968). Laymon Miller, Dept. of the Army Technical Manual, TM 5-805-9. "Power Plant Acoustics." Washington, D.C.: Headquarters, Department of the Army, 1968.
Miller (1980). Laymon N. Miller, "Noise and Vibration Control for Mechanical Equipment," Bolt Beranek and Newman, Inc. Report 4131, Project No. 08804. Huntsville, AL: Department of the Army Corps of Engineers, Jan. 1980.
Millington (1932). G. Millington, J. Acoust. Soc. Am., 4, 160, 1932.
Mindlin (1951). R.D. Mindlin, "Influence of Rotary Inertia and Shear on Flexural Motion of Elastic Plates," J. Appl. Mech., vol. 18, 1951.
Mitchell (2002). Doug Mitchell, "Surround Setup Scenarios," Nashville, TN: Audio Media Magazine, issue 49, Feb. 2002.
Moderow (1970). R.R. Moderow, "Vibration Characteristics of Steel Joist-Concrete Slab Floor Systems." M. Sc. Thesis, Dept. Civ. Eng., Lawrence, KS: Univ. Kansas, unpublished, 1970.
Mommentz (1995). E. Mommentz and M. Vorlander, "Measurement of scattering coefficients of surfaces in the reverberation chamber and in the free field," Proc. 15th Internat. Congress on Acoustics, Trondheim, Norway, 1995.
Monsanto Corp (1995). "Glass Sound Transmission Loss Data," Saflex, www.saflex.com/Acoustic/31atable.htm, 1995.
Morse (1948). Philip M. Morse, Vibration and Sound. New York, NY: McGraw-Hill. Reprinted (1981) Melville, NY: Acoustical Society of America, 1948.
Morse (1968). Philip M. Morse and K. Uno Ingard Theoretical Acoustics. New York, NY: McGraw-Hill. © 1968, New York, NY: Reprinted with permission from McGraw-Hill Book Company.
Murphy (2001). Shawn Murphy, Private Communication, 2001.
Nagata (1997). Minoru Nagata, Private Communication, 1997.
National Bureau of Standards (1975). Acoustical and Thermal Performance of Exterior Residential Walls, Doors and Windows, NBS Building Science Series 77. Washington D.C.: U.S. Depart. of Commerce, National Bureau of Standards, 1975.
National Research Council Canada (1966). "Sound transmission loss of a Q-Floor assembly with floor tile and fireproofing only." rpt. 433-PY, Ottawa, CA: National Research Council, Nov. 1966.
National Research Council Canada (1966). "Sound transmission loss of a Q-Floor assembly with floor tile, fireproofing and suspended ceiling system." rpt. 434-PY, Ottawa, CA: National Research Council, Nov. 1966.
Naylor (1993). G.M. Naylor, "ODEON - Another hybrid room acoustical model." Applied Acoustics, vol. 38 nos. 2–4, 1993.
Nickson et al. (1954). A.F.B. Nickson, R.W. Muncey, and P. Dubout, "The Acceptablity of Artificial Echoes With Reverberant Speech and Music." Acustica, Vol. 4, pp. 515–518. Oldenberg, Germany, 1954.
Nightingale (July 1998). T.R.T. Nightingale and J.D. Quirt, "Effect of electrical outlet boxes on sound insulation of a cavity wall." Reprinted with permission from J. Acoust. Soc. Am., 104(1), © 1998, Acoustical Society of America: Melville, NY, July 1998.
Nishihara et al. (2001). Noriko Nishihara, Takayuki Hidaka, and Leo L. Beranik, "Mechanism of sound absorption by seated audience in halls." Reprinted with permission from J. Acoust. Soc. Am., 110(5), Pt. 1, © 2001, Acoustical Society of America: Melville, NY, Nov. 2001.
Ogawa (1965). cited in Davis and Davis. Sound System Engineering. 1975, Fig. 4-21, Indianapolis, IN: Howard W. Sams, 1965.

Okano (1998). T. Okano, L. Beranek, and T. Hidaka, "Relations among interaural cross-correlation coefficient ($IACC_E$), lateral fraction (LF_E), and apparent source width (ASW) in concert halls," Reprinted with permission from J. Acoust. Soc. Am. 104 (1). © 1998, Acoustical Society of America: Melville, NY, July 1998.

Olive (1989). Sean E. Olive and Floyd E. Toole, "The Detection of Reflections in Typical Rooms." J. Audio Eng. Soc. 37 (7/8). New York, NY: Audio Engineering Society, July/Aug. 1989.

Ollerhead (1968). J.B. Ollerhead, "Subjective Evaluation of General Aircraft Noise," Tech. Rept. 68-35. Contract, FA67WA-1731. Washington, D.C.: Wyle Laboratories, Federal Aviation Administration, 1968.

Olson (1957). Harry F. Olson, Elements of Acoustical Engineering. New York, NY: Van Nostrand Reinhold, 1957.

Olson (1959). H.F. Olson, "Acoustoelectronic auditorium," J. Acoust. Soc. Am., vol. 31, no. 7, Melville, NY, 1959.

Olson (1991). Harry F. Olson, Acoustical Engineering. Philadelphia, PA: Professional Audio Journals, Inc., 1991.

Owens-Corning Fiberglas Corp. (1970). "Wood Stud Gypsum Board Drywalls Sound Transmission Tests." Product Testing Laboratories Report No. 32491. Toledo, OH: Owens Corning World Headquarters, Feb. 1970.

Owens-Corning Fiberglas Corp. (1971). "Sound Attenuation of Steel Stud Drywall Partitions." Toledo, OH: Owens Corning World Headquarters, May 1971.

PAC International (2001). RSIC-1 pamphlet. Tualatin, OR, 2001.

Palmer (1961). R.R. Palmer, A History of the Modern World, 2nd ed. New York, NY: Alfred A. Knopf, 1961.

Paris (1928). E.T. Paris, "On the Coefficient of Sound-Absorption Measured by the Reverberation Method," Phil. Mag., (7)5:489–497, 1928.

PCI Industries (1999). "E477-96 Silencer Test Report." Fort Worth, TX: PCI Industries Sound Testing Laboratory, 1999.

Pearsons et al. (May 1977). K.S. Pearsons, R.L. Bennett, and S. Fidell, "Speech Levels in Various Noise Environments." Report EPA-600/1-77-025, Washington, D.C.: U.S. Environmental Protection Agency, May 1977.

Peterson (1974). Arnold P.G. Peterson and Ervin E. Gross, Jr., Handbook of Noise Measurement. 7th ed. Concord, MA: General Radio, 1974.

Peutz (1971) V.M.A. Peutz, "Articulation loss of consonants as a criterion for speech transmission in a room." J. Audio Eng. Soc. 19 (11). New York, NY: Audio Engineering Society, 1971.

Peutz (1974). V.M.A. Peutz, "Designing Sound Systems for Speech Intelligibility, presented at the 48th Convention of the Audio Engineering Society," AES Journal, (Abstracts) vol. 22 (June 1974), preprint 967, New York, NY, 1974.

Peutz (1974). V.M.A. Peutz and W. Klein, "Articulation Loss of Consonants Influenced by Noise, Reverberation and Echo," (in Dutch), Acoust. Soc. Netherlands, vol. 28, 1974.

Peutz (1984). V.M.A. Peutz and B.H.M. Kok, "Speech Intelligibility," presented at the 75th Convention of the Audio Engineering Society, AES Journal, (Abstracts) vol. 32 (June 1984), preprint 2089, New York, NY, 1984.

Pierce,. (1981). Allan D. Pierce, Acoustics: An Introduction to Its Physical Principles and Applications.. © 1981, Allan D. Pierce. New York, NY: Reprinted with permission from McGraw-Hill Book Company.

Pierce (1983). John R. Pierce, from The Science of Musical Sound by John R. Pierce © 1983 by Scientific American Books. Used with permission by W.H. Freeman and Company.

Piercy (1971). J.E. Piercy, "Comparison of Standard Methods of Calculating the Attenuation of Sound in Air with Laboratory Measurements," presented orally to 82nd Meet. Acoust. Soc. Am., Denver, CO, 1971.

Pinnington (1988). R.J. Pinnington, "Approximate Mobilities of Builtup Structures," Report No. 162, Southampton, England: Institute of Sound and Vibration Research, 1988.

Plomp (1965). R. Plomp and W.J.M. Levelt, "Tonal Consonance and Critical Bandwidth," J. Acoust. Soc. Am., 38, © 1965, Acoustical Society of America: Melville, NY.

Quirt (1982). J.D. Quirt, "Sound transmission through windows I. Single and doubling glazing." Reprinted with permission from J. Acoust. Soc. Am., 72(3), © 1982, Acoustical Society of America: Melville, NY.

Ramsey (1986). Charles G. Ramsey and Harold R. Sleeper, Architectural Graphic Standards. © 1970, New York, NY: John Wiley & Sons.

Rausch (1943). E. Rausch, Maschinenfundamene und andere dynamische Bauaufgaben, Vertrieb VDI Verlag G.M.B.H. (Berlin).

Rayleigh (1894–1896). Baron Rayleigh (John William Strutt), The Theory of Sound. Vol. 1 and 2, orginally published in 1894 and 1896. Republished by Dover Publications in 1945, New York, NY.

Reif (1965). F. Reif, Fundamentals of statistical and thermal physics. New York, NY: McGraw-Hill Book Company 1965.

Reiher (1931). H. Reiher and F.J. Meister, "The Effects of Vibration on People." (in German) Forschung auf dem Gebiete des Ingenieurwesens, V.2 II, 1931. (Translation: Report No. F-TS-616-RE H.Q. Air Materials Coimmand, Wright Field, OH, 1949).

Reynolds (1981). D.D. Reynolds, Engineering Principles of Acoustics. Saddle River, NJ: © (1981) Reprinted by permission of Pearson Education, Inc.

Reynolds (1990). D.D. Reynolds, HVAC Systems Duct Design, 3rd ed., "Noise Control, Circular Ducts." Chantilly, VA: SMACNA, 1990.

Richart et al. (1970). F.E. Richart, Jr., J.R. Hall, Jr., and R.D.Woods, Vibrations of Soils and Foundations. © 1970, Reprinted by permission of Prentice-Hall, Upper Saddle River, NJ.

Rindel (1985). J.H. Rindel, "Attenuation of sound reflections from curved surfaces," in Proceedings of 24th Conference on Acoustics, The High Tatras, Czechoslovakia, October 1985.

Rindel (1986). J.H. Rindel, "Attenuation of Sound Reflections Due to Diffraction.".Nordic Acoustical Meeting, 20–22 August, 1986, Aalborg University. Aalborg, Denmark, 1986.

Rindel (1990). J.H. Rindel, "Attenuation of Sound Reflections from an Array of Reflectors." 29th Conference on Acoustics, Oct. 2–5, 1990. Strbske Pleso, 1990

Rindel (1993). J.H. Rindel, "Modelling the Angle-Dependent Pressure Reflection Factor." Applied Acoustics. England: Elsevier Science Publishers Ltd., 1993.

Rindel (2000). J.H. Rindel, "The Use of Computer Modeling in Room Acoustics." Journal of VibroEngineering, no. 3(4) /Index 41-72 Paper of the International Conference BALTIC-ACOUSTIC 2000/ISSN 1392-8716, 2000.

Rindel (2002). Jens Holger Rindel, "Computer Simulation Techniques for Acoustical Design of Rooms - How to Treat Reflections in Sound Field Simulation." proceedings, ASVA, Tokyo, 2–4 April 1997.

Roark (1975). Raymond J. Roark and Warren C. Young, Formulas for Stress and Strain, 5th ed., © 1975 by McGraw-Hill, New York, NY: McGraw-Hill.

Robinson (1956). D.W. Robinson and R.S. Dadson, "A re-determination of the equal-loudness relations for pure tones," *Br J. Appl. Phys.* 7 166–181, 1956.

Robinson (1964). D.W. Robinson and L.S. Whittle, "The loudness of octave-bands of noise." Acustica 14, 1964.

Rolland et al. (1948). Romain Rolland, Pierre Lalo, Vincent d'Indy, Pietro Mascagni, Henry T. Finck, Horatio Parker, and others. The International Library of Music for Home and Studio, vol. II. © 1948, The University Society.

Roozen et al. (1998). N.B. Roozen, M. Bockholts, P. Van Eck, and A. Hirschberg, "Vortex sound in bass-reflex ports of loudspeakers. Part I. Observation of response to harmonic excitation and remedial measures." Reprinted with permission from the author and J. Acoust. Soc. Am., vol. 104, no. 4, © 1998, Acoustical Society of America: Melville, NY, Oct. 1998.

Rosenblith (1953). W.A. Rosenblith, K.N. Stephens, and staff of Bolt Beranek and Newman, Inc., "Handbook of Acoustical Noise Control. II. Noise and Man." Rept. WADC TR 52-204. U.S. Air Force Aerospace Medical Laboratory, Wright-Patterson Air Force Base, Ohio, 1953.

Rossing (1990). T.D. Rossing, The Science of Sound. 2nd ed. Reading, MA: Addison Wesley, 1990.

Rossing (1995). Thomas D. Rossing and Neville H. Fletcher, Principles of Vibration and Sound. Reprinted with permission from the author and Springer-Verlag. © 1995, New York, NY: Springer-Verlag.

Ruzicka (July 1971). Jerome E. Ruzicka, "Fundamental Concepts of Vibration Control." Bay Village, OH: Sound and Vibration Magazine, July 1971.

Sabine (1900). Wallace Clement Sabine, "Reverberation," Chapter 1, The American Architect and The Engineering Record, from the Collected Papers on Acoustics, Wallace Clement Sabine. © 1964, New York, NY: Dover Publications, reprinted.

Sabine (1922, 1964). Wallace Clement Sabine, Collected Papers on Acoustics. New York, NY: Dover Publications, reprinted, 1964.

Sabine et al. (1975). Hale Sabine , Myron B. Lacher, Daniel R. Flynn, and Thomas L. Quindry, Acoustical and Thermal Performance of Exterior Residential Walls, Doors and Windows. National Bureau of Standards, Washington D.C, 1975.

Sandars (1968). N.K. Sandars, Prehistoric Art in Europe. London: Penguin.

Sanicola (1997). Henry Sanicola, Private Communication, 1997.

Sauter (1970). August Sauter, Jr. and Walter W. Soroka, "Sound Transmission through Rectangular Slots of Finite Depth between Reverberant Rooms," J. Acoust. Soc. Am., vol. 47, no. 1 (part 1), © 1970, Acoustical Society of America: Melville, NY.

Schaffer (1991). Mark E. Schaffer, A Practical Guide to Noise and Vibration Control for HVAC Systems. Atlanta, GA: ASHRAE, 1991.
Scharf (1970). B. Scharf, "Critical Bands," in Foundations of Modern Auditory Theory, ed. by J.V. Tobias. Burlington, MA: Academic Press, 1970.
Schneider et al. (1970). E.J. Schneider, J.E. Mutchler, H.R. Hoyle, E.H. Ode, and B.B. Holder, "The Progression of Hearing Loss from Industrial Noise Exposures." Am. Ind. Hygiene Assoc. Jour. 31: 368: Fairfax, VA, 1970.
Schroeder (1954). M.R. Schroeder, "Die statistischen Parameter der Frequenzkurven von großen Räumen," Acustica 4, 1954.
Schroeder (1979). Manfred R. Schroeder, "Binaural dissimilarity and optimum ceilings for concert halls: More lateral sound diffusion." Reprinted with permission from J. Acoust. Soc. Am., 65, 958–963, © 1979, Acoustical Society of America: Melville, NY, 1979.
Schroeder (1981). M.R. Schroeder, "Modulation transfer functions: Definition and measurement." Acustica 49, 1981.
Schroeder (May 1996). Manfred R. Schroeder, "The Schroeder frequency revisited." Physics Today. Woodbury, NY: American Institute of Physics, May 1996.
Schroeder (1963). M.R. Schroeder and B.S. Atal, "Computer simulation of sound transmission in rooms." In IEEE International Convention Record (7). New York, NY: IEEE Press, 1963.
Schuller et al. (1995). Willem M. Schuller, Fokke D. van der Ploeg, and Peter Bouter, Noise Control Engineering Journal,. "Impact of diversity in aircraft noise ratings," Ames, IA: Iowa State University, Nov–Dec 1995.
Schultz (1964). T.J. Schultz and B.G. Watters, "Propagation of sound across audience seating." Reprinted with permission from J. Acoust. Soc. Am., © 1964, Acoustical Society of America: Melville, NY, 1964.
Seraphim (1961). H.P. Seraphim, "Uber die wahrnehmbarkeir mehrerer Ruckwurfe von Sprachschall." Acustica II, 81, 1961.
Seto (1971). William W. Seto, Schaum's Outline of Theory and Problems of Acoustics. © 1971, New York, NY: Reprinted with permission from McGraw-Hill Book Company, 1971.
Sette (1933). W.J. Sette, J. Acoust. Soc. Am. 4, 160, 1933.
Sharp (1973). Ben H. Sharp, A Study of Techniques to Increase the Sound Insulation of Building Elements. WR 73-5. El Segundo, CA: Wyle Laboratories, June, 1973.
Shirley (2000). Peter Shirley, Realistic Ray Tracing. Natick, MA: A K Peters, 2000.
Shure, Inc. (2002). "Microphone directivity patterns." Evanston, IL: Shure, Incorporated, 2002.
Skudrzyk (1954). Eugen Skudrzyk, Die Grundlagen der Akustik. Wien: Springer-Verlag, 1954.
SMACNA (1990). HVAC Systems Duct Design, 3rd Edition. Chantilly, VA: © 1990, Sheet Metal and Air Conditioning Contractors' National Association, 1990.
Small (1973). Richard H. Small, "Vented-Box Loudspeaker Systems Part I, II, III & IV," New York, NY: Journal of the AES, vol. 21, no. 5, 6, 7, and 8, June, July, August, and September 1973.
SMPTE (1989). SMPTE Engineering Guideline, EG 19. "Design of Effective Cine Theaters." © 1989, White Plains, NY: Society of Motion Picture and Television Engineers.
SMPTE (2001). "Film Sound History - 30's," created by Jonathan Kay, Kimber Ghent, Brian Chumney, and Erik Lutkins. http://www.mtsu.edu/~smpte/thirties.html, 2001.
Snow (1955). William B. Snow, "Frequency Characteristics of a Sound Reinforcing System." AES Journal. New York, NY: Audio Engineering Society, Apr. 1955.
Spandock (1934). F. Spandock, "Akustische Modellversuche," Ann, der Physic, 20, 1934.
Steele (1996). James Steele, Theatre Builders. London, England: Academy Editions, 1996.
Steinberg (1929). John C. Steinberg, "Effects of Distortion Upon the Recognition of Speech Sounds." Reprinted with permission from J. Acoust. Soc. Am., © 1929, Acoustical Society of America: Melville, NY, Oct. 1929.
Stern (1980). Richard Stern, Private Communication, 1980.
Stevens (1972). J. Acoust. Soc. Am., 51, 575. © 1972, Acoustical Society of America: Melville, NY, 1972.
Stevens (1961). S.S. Stevens, "Procedure for calculating loudness." Mark IV, J. Acoust. Soc. Am., 33, © 1961, Acoustical Society of America: Melville, NY, 1961.
Stevens (1969). S.S. Stevens, "Assessment of Noise: Calculation Procedure Mark VII." Paper 355-128. Cambridge, MA: Laboratory of Psychophysics, Harvard University, December 1969.
Strayer (1955). Joseph R. Strayer, Western Europe in the Middle Ages, a Short History. New York, NY: Appleton-Century-Crofts, Inc, 1955.
Swing (1973). Jack W Swing and Donald B.Pies, Assessment of Noise Environments Around Railroad Operations. El Segundo, CA: Wyle Labs, July 1973.

Thiele (1971). A.N. Thiele, "Loudspeakers in Vented Boxes: Part I & II," New York, NY: Journal of the AES, vol. 19, no. 5 & 6, May & June 1971.
Thiele (1953). R. Thiele, "Directional Distribution and Time Sequence of Return of Sound in Rooms." Acoustica, vol. 3, no. 2, 1953.
Thiessen (1978). G.J. Thiessen, "Disturbance of Sleep by Noise." Reprinted with permission from J. Acoust. Soc. Am., 64, © 1978, Acoustical Society of America: Melville, NY.
Thiessen (1968). G.J. Thiessen and N. Olson, "Community Noise - SurfaceTransportation." Bay Village, OH: Sound and Vibration Magazine 2(4): 10, 1968.
Thomson (1965). William T. Thomson, Vibration Theory and Applications. © 1965, Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NY, 1965.
TNM (1998). FHWA Traffic Noise Model Technical Manual, ver. 1.0. U.S. Dept. of Transportation, Federal Highway Administration, Office of Environment and Planning, Washington, DC, 1998.
Toole (1990). Floyd E. Toole, "Loudspeakers and Rooms for Stereophonic Sound Reproduction." AES 8th International Conference Proceedings, National Research Council of Canada Ottawa, Ont.. New York, NY: Audio Engineering Society, 1990.
Toole (2002). Floyd E. Toole, "Loudspeakers and Rooms for Multichannel Audio Reproduction, Part One: How many loudspeakers? What kind? Where do we put them?" www.harman.com 02/12/2002.
US EPA (1973). Model Noise Ordinance. U.S. Environmental Protection Agency, Washington, D.C, 1973.
Ungar (1979). Eric E. Ungar and Robert W.White, "Footfall-Induced Vibrations of Floors Supporting Sensitive Equipment." Bay Village, OH: Sound and Vibration Magazine, 1979.
Uniform Building Code (1982). Appendix, Chapter 35. © 1982, Whittier, CA: International Conference of Building Officials, 1982.
Uris et al. (2000). Antonion Uris, Ana Liopis, and Jaime Lin, "Sound Transmission Loss." Sound Solutions Magazine, July–Dec. 2000.
Van Houten (1979). J.J. Van Houten, "Test Report No. 802-79 for Speciality Products Co.." Orange, CA: J.J. Van Houten & Associates, 1979.
Veneklasen (1996). Paul Veneklasen, Private Communication, 1996.
Veneklasen (1969). P.S. Veneklasen and J.R. Hyde, "Auditorium Synthesis - Early Results of Listener Preference." Reprinted with permission from J. Acoust. Soc. Am., 46 97, © 1969, Acoustical Society of America: Melville, NY.
Vér (1971). István Vér, "Impact Noise Isolation of Composite Floors." Reprinted with permission from J. Acoust. Soc. Am., © 1971, Acoustical Society of America: Melville, NY, 1971.
Ver (1983). Istvàn Vér, "Prediction of Sound Transmission Through Duct Walls; Breakout and Pickup," Report No. 5116, ASHRAE TRP-319, American Society of Heating, Refrigeration, and Air Conditioning Engineers.
Ver (1984b). I. Ver, "Prediction of sound transmission through duct walls: Breakout and pickup." ASHRAE Transactions 90(2A), 1984b, 1984.
Ver (1971). Istvàn Vér and Curtis I. Holmer, "Interaction of Sound Waves with Solid Structures," Noise and Vibration Control, edited by Leo L. Beranek, © 1971, New York, NY: McGraw-Hill Book Co.
Ver (1983). Istvàn L. Vér, Prediction of Sound Transmission Through Duct Walls; Breakout and Pickup, Report 5116, Prepared for American Society of Heating, Refrigeration, and Air Conditioning Engineers, 1983.
Ver (1992). Leo L. Beranek and István L. Vér, Noise and Vibration Control Engineering Principles and Applications, Ch. 9, "Interaction of Sound Waves with Solid Structures." © 1992, New York, NY: this material is used by permission of John Wiley & Sons, Inc.
Vermuelen (1958). R. Vermuelen, "Stereo-Reverberation," J. Audio Eng. Soc., vol. 6, no. 2, April 1958.
Vermeulen (1936). R. Vermeulen and J. De Boer, Philips Tech Rev., 1, 46.
Vibron (1989). "Vibration Isolation Selection Guide, Heating/Ventilating and Air Conditioning Equipment," Don Allen. Mississauga, Ontario Canada: Vibron, 1989.
Vinokur (1996). Roman Y. Vinokur, "Evaluating Sound-Transmission Effects in Multi-Layer Partitions." Bay Village, OH: Sound and Vibration Magazine, July 1996.
Vitruvius (1960). Vitruvius, the Ten Books on Architecture. translated by Morris Hicky Morgan. New York, NY: Dover Publications, 1960.
von Bekesy (1960). G. von Bekesy, Experiments in Hearing. © 1960, New York, NY: Reprinted with permission from McGraw-Hill Book Company.
von Gierke (1973). H. von Gierke, Impact Characterization of Noise Including Implications of Identifying and Achieving Levels of Cumulative Noise Exposure. Environmental Protection Agency, 1973.
Waterhouse (1955). Richard V. Waterhouse, "Interference Patterns in Reverberant Sound Fields." Reprinted with permission from J. Acoust. Soc. Am.,27(2), © 1955, Acoustical Society of America: Melville, NY, Mar. 1955.

Watters et al. (1963). B.G. Watters, L.L. Beranek, F.R. Johnson, and I. Dyer, "Reflectivity of panel arrays in concert halls," SOUND 2, 1963.

Webster (1969). John C. Webster, Effects of Noise on Speech Intelligibility. Proc. of Conference, Noise as a Public Health Hazard, Washington D.C., June 13–14, 1968, A.S.H.A. Reports 4. Washington D.C.: The American Speech and Hearing Association, 1969.

Wegel (1924). R.L. Wegel and C.E. Lane, "Auditory masking of one pure tone by another and its possible relation to the dynamics of the inner ear." Phys. Rev. 23, 1924.

Wells (1958). R.J. Wells, "Acoustical Plenum Chambers." Noise Control, July 1958.

Wells (1967). R.J. Wells, Recent research relative to perceived noise level. (A) J. Acoust. Soc. Am., 42, © 1967, Acoustical Society of America: Melville, NY.

Wendt (1963). K. Wendt, Das Richtungshören bei der überlagerung zweier Schallfelder bei Intensitäts-und Laufzeitstereophonie (Directional hearing with two superimposed sound fields in intensity-and delay-difference stereophony). Dissertation, Technische Hochschule, Aachen, 1963.

Wenger (1999). Wenger Corporation, Planning Guide for Secondary School Music Facilities. © 1999, Owatonna, MN: Wenger Corporation, 1999.

Wilhelmi Corp Data (2000). based on Riverbank Acoustical Laboratories test RAL-A85-116, 2000.

Wilkens (1974). Wilkens and Plenge, Auditorium Acoustics, ed. Robin Mackenzie, "The Correlation between Subjective and Objective Data of Concert Halls." New York, NY: John Wiley & Sons.

Wilson (1965). G.P. Wilson and W.W. Soroka, "Approximation to the diffraction of sound by a circular aperture in a rigid wall of finite aperture." Reprinted with permission from J. Acoust. Soc. Am., 37, © 1965, Acoustical Society of America: Melville, NY, Feb. 1965.

Wilson (1976). "Architectural Acoustics Design and Noise Control in Army Medical Facilities." Technical Background Report for Appendix V - "Acoustical Control" to DOA Technical Manual 5-838-2, Medical Facilities Design - Army. Oakland, CA: Wilson, Ihrig & Associates, Jan. 1976.

Woodworth et al. (1954). R.S. Woodworth and H. Schlosberg. Experimental Psychology. New York, NY: Holt, 1954.

Wyle Laboratories (1971). Transportation Noise and Noise from Equipment Powered by Internal Combustion Engines. NTID300.13. El Segundo, CA: Wyle Laboratories, Dec. 1971.

Yamamoto (1968). T. Yamamoto and T. Wakuri, "Equipment and Materials Used in Scale NHK Laboratories Note," No. 77, 1.

INDEX

A

Abel, Carl Fredrick 21
Absorption 31, 93, 146, 169, 172–173, 175, 203, 225, 248–264, 267–282, 300–306, 308–310, 312–313, 318, 352, 356–357, 366, 376–377, 405, 407, 425–426, 461, 474, 495, 510, 548, 554, 556, 558–559, 564, 591, 603–606, 608, 615, 618, 621, 655, 660–662, 665, 678, 684, 697, 700–702, 705–706, 712, 715, 720, 723–724, 728–729, 747–748, 753–754, 756–757, 764–766, 769, 771–773, 777, 779–780, 782–784, 795–799, 801
  coefficient 250–251, 253–256, 261, 264, 267, 269, 272–273, 276–279, 300–304, 310, 312, 356, 495, 549, 556, 558, 564, 591, 605, 662, 678, 700, 702, 756
  cross section 281
  diaphragmatic 615, 724, 766, 774
  diffuse-field 254–255, 267, 276, 678, 796
  porous absorbers 239, 253, 255–256, 261–268, 270–271, 276, 282, 756, 778
  panel 253, 256, 261, 267–268, 270–278
  random incidence 309–310
  resonant 261, 273–274, 277–280
  Sabine 495
  seat 282, 701–702, 771
  specular 253
Acoustical defect 579, 587–590, 611, 711, 725, 771
Acoustic Management Systems 738
Acoustic modeling 781
Active noise cancellation 210
Adaptation 77
Adiabatic
  bulk modulus 53–54, 56
  equation of state 262
ADR 742, 754, 772–773
Aeschylus 3
African music 1
Ainger, F. 599
Air Installation Compatible Land Use Zone (AICUZ) 197
Air
  attenuation 168, 170, 172, 174, 175, 181, 188, 189, 303
  spring 201, 202, 228, 334
Aircraft 158, 194–198, 368, 378
Alaric 7
Albersheim 593, 594
Aleotti, Giovanni Battista 17
Alexandrovich, George 741
Allen, Don 395, 407–409, 415, 416, 716
Altec 231, 232
Altes Gewandhaus 22, 23, 27
Amati, Nicolo 19
Ambassador Auditorium 710, 711
Ambient (background) 127–129
American Society of Heating, Refrigeration and Air Conditioning Engineers (see ASHRAE)
Amphitheater 4, 580
Amplification 608
Amplifier
  duty cycle 622, 624
  constant voltage 616, 617
  efficiency 611, 614, 615, 622
  electrical power 622
  heat load 623
  power 621
  power factor 623
Amsterdam 660, 663, 664, 670, 672, 673, 676, 677, 679, 685, 686
Anatomy of the ear 73
Anderson 165, 189
Anderson, G. S. 181
Ando, Y. 241, 282, 283
Andree, C. A. 301
Annoyance 98, 101, 102, 104
Anthemius of Tralles 9
Aperture 347, 348
Apollo 3
Appliances 540
Aristotle 4
Arnold, Harold D. 741
Arrival point of sight (APS) 580
Artec 733–734
Articulation 620
Articulation Class (AC) 548, 550, 559
Articulation Index (AI) 95–97, 151, 544–546, 548, 592–593
Articulation Loss of Consonants (AL_{cons}) 96–97, 151, 592–596, 602, 620
Artificial reverberation 736, 738
ASHRAE 87–89, 102, 395, 454, 459–461, 469, 476, 488–489, 500
Askenfeld 724

Aspendius, Turkey 4
Atal, B. S. 810, 812
Atmospheric pressure 53, 56, 59, 74, 123, 171–172, 202
Attenuation
  air (atmospheric) 127, 160, 169
  area 244–245
  atmospheric turbulenc 127
  diffraction 161, 240–243
  environmental effects 167
  ground cover 127
  grazing 282
Audio animatronic 744
Audio visual 563, 606–608, 705, 720
Auditorium 580, 583, 587, 608, 617–618, 630, 681, 691, 701, 703, 705–712, 722, 725–726, 728–729, 734, 736, 745, 787
Auditory canal (meatus) 73–74
Auditory nerve 76, 114
Auralization 781, 785, 804
Aural reflex 74–75, 105
Average complex sum 651
A-weighting 85–86, 125, 133–134, 136–137, 141, 547
Axial mode 745–747, 749

B

Bach, Carl Philipp Emanuel 21
Bach, Johann Christian 2
Bach, Johann S. 19, 21, 22, 28, 95
Background (ambient) noise 127–129, 514–515, 524, 768
Back pressure 486, 497–498, 529, 531, 578
Back to back pipe 520
Baffle 347, 376–378, 490, 495, 569, 615, 767, 777
Bagenal, Hope 19, 22, 23
Baker, 596
Balanced Noise Criterion Curves (NCB) 139–141
Balcony 580, 617–618, 653, 660–662, 682, 684–685, 691, 693, 696, 702–703, 705, 707, 710, 714, 716–717, 719–720, 728–729, 736
Balilah 348
Ballou, Glen 636
Bandwidth 40, 77–79, 84, 93, 114, 121, 125, 133, 137, 141–143, 145, 218, 239, 281–282, 298, 326, 612, 614–616, 631–632, 650–651, 657, 736, 778
Barberini Theater 18
Baroque 1, 10, 14, 16–19
  Italian opera house 17
  music 1, 18–19
Barrier
  attenuation 157, 161–167, 177, 180–181, 186, 188, 194, 199, 213, 368, 379, 544, 553, 556, 563, 637, 666
  bright zone 162
  diffraction 161
  line source 159–161, 163, 165–167, 185–186, 188, 194, 213–216, 225
  partial-height 560, 564, 720
  shadow zone 161–163, 177–178
  transition zone 162
Barron, Michael 653, 660–667, 670–675, 681, 682, 688, 691, 693, 722–724, 730
Barry, T. M. 135, 159–160, 165, 169, 183–185, 187
Base
  concrete inertial 394–395, 399, 416
  integral steel 393–394
  rail frame 394
Basilar membrane 76–77, 114
Basilican church 7–8, 10, 25
Bass
  suckout 287, 314
  trap 203, 268, 274, 748, 763, 766–767, 769, 772, 774
Batt insulation 273, 333–335, 337–338, 341, 357, 360, 369, 380, 517–518, 520, 531–532, 535, 567, 748, 767, 769, 778
Bazley, E. N. 266
Beam tracing 785
Beamwidth 210, 212–214, 220, 232–233
Beats 37, 48–50, 79, 90, 93, 238
Beethoven, Ludwig 21, 23–25
Bell Laboratories 82, 95, 97, 592–593, 741
Bell, Alexander Graham 31
Bending 240, 320, 322–323, 325–326, 328, 330–332, 341, 345
  complex stiffness 325
  impedance 323, 325, 328
  stiffness 320, 323, 325, 330–332
  wave velocity 323
Benedetti, Giovanni Battista 20
Benedetto, Giuliana 253
Beranek, Leo 23, 26–34, 54, 60–61, 85, 87, 96, 138–141, 175, 181, 224, 245–246, 255, 293–294, 323, 327, 382, 442, 459, 586, 653, 658, 660–669, 671–673, 677–695, 701–702, 722, 728, 734, 748
Berger, J. F. 600
Berlin Philharmonic Hall 658, 665, 687, 688
Berliner, Emil 741
Berlioz, Hector 25, 81
Bernini, Giovanni Lorenzo 14, 18
Bernoulli, Daniel 20
Beth Shean, Israel 6
Beverly Hills 513, 515
Binaural sound 114, 667, 809
Bistafa, Sylvio R. 594–596, 598, 599, 603
Black-box theater 706
Blade passing frequency 195
Blauert, Jens 107–108, 111
Blazier, Warren 87–88, 464–466, 495, 564
Blocking 357, 376, 378, 386, 519, 524, 527, 542
Blumlein 741
Boethius 3
Boiler 451, 536–537, 540, 542
Bolt, R. H. 297
Boston Symphony Hall 31–33, 656, 658, 663–665, 670, 672–674, 676–677, 679–680, 684–685, 722, 728, 734, 740, 770, 781–782, 804–805
Boundary condition 154, 175, 237, 239, 249–250, 252, 284
Bourges Cathedral 11
Boyle's Law 54
Bradley, J. S. 279, 587, 593–596, 598–603, 606, 620, 670, 673
Bragg, 245–246, 673, 704
Brahms, Johannes 25
Bramante, d'Agnolo 14
Breton, Gaelle 4, 15–18
Bridging 339, 345–347, 363–364
Bronsdon, Robert 181
Brownian movement 73
Brueckwald, Otto 25
Building code 509–510
Bulk modulus 53–54, 205
Burbage, James 15

Burkat 17
Burris-Meyer, Harold 698
Byzantine Empire 7, 9–10

C

CADP 639
Cahill, Thomas 7
California Environmental Quality Act (CEQA) 90
California Noise Insulation Standards 356
California Office of Noise Control (CONC) 340, 358–363
Camerata 16
Canonical coordinates 794
Cantata 16
Carnegie Hall 32–34
Carnegie, Andrew 32
Carrying channel 525
Casing radiation 461, 479
Caulk 357, 359, 376, 378, 402
Cavanaugh, William 759, 760, 762–763
Ceiling
  Acoustical 255, 258, 466, 564
  fiberglass 558–560, 567, 571, 764
  high transmission loss 515
Cent 81
Cerritos Performing Arts Center 729–732
Chaconne 16
Chamber music 19, 27
Chanaud, Robert 97–99, 544, 546, 548–551, 553–564, 570–571
Channeling 168, 181
Characteristic equation 384
Charlemagne 7, 10
Chartes Cathedral 11
Chopin, Frederic 25
Church 697–701, 705, 712–717, 719, 727, 735
CIPIC 808–811
Clarity 612, 624, 654–655, 657, 668, 671–674, 684–685, 691, 705, 710–711, 720, 728, 730, 734, 740, 754, 765, 777
Clark, David 752
Classical music 21, 24–25, 27, 29
Classroom 535, 543, 563, 579–580, 583, 585, 587, 596, 598, 600, 606–607, 614
C- weighting 85–86, 121, 126, 759
Cochlea 73–74, 76–77, 82, 91, 93, 105
Cocktail party effect 603–604
Coincidence
  Effect 323–325, 339–341, 365
  Frequency 323–326, 328, 332, 340
Cole, Edward C. 698
Coloration 120, 213, 312
Colosseum 4
Comb filter 127, 217–218, 496, 611, 624, 636–637, 737, 752, 753
Community 13, 98, 101–104, 135
Community Noise Equivalent Level (CNEL) 101–102, 125, 130–133, 142, 144
Community Noise Rating(CNR) 142
Complex
  amplitude ratio 237, 250–251
  exponential 44–45
  plane 43, 45
  propagation constant 482–483, 493
Concertgebouw 29–30, 670, 677–678, 685–686, 740
Concerto 16–17, 19, 21
Concrete 509, 515–516, 519, 521–525, 527–528, 535–537, 539, 541, 660, 680–681, 685, 691, 695, 734, 746, 759, 764, 773, 775–777, 779, 797
  on metal deck 522
  slab 395, 402, 408, 416, 515, 519, 522–523, 541, 685, 773, 776
Condenser 401, 451, 454, 456, 462, 537, 763
Condominium 401, 510, 512–513, 526, 528, 536, 540
Conduit 759, 761
Conference room 89, 579, 606–607, 618
Connection
  line 345–347
  point 345–347
Consonance 78–79, 93
Consonants 79–80, 93, 96, 151, 579, 592–594, 620, 671
Consonant-vowel-consonant (cvc) 95, 592
Constantine 7
Continuo 16
Control Room 203, 274, 341, 746, 753–754, 756, 758, 769–772, 774–780
Convolution 411, 804, 806–807
Coupled spaces 587, 691
Craik, R. J. M. 518
Crede, Charles E. 412
Cremer, Lothar 254, 277, 278, 323, 326, 327, 435, 437, 440, 666, 674, 687, 688
Critical
  band 77, 79, 91, 218
  distance 307, 591, 597, 736, 737
  frequency 324, 325, 326, 327, 328, 330, 331, 332, 333, 340, 346, 347, 360, 430, 516
  region 335
Crossover 323, 326, 344, 500–501, 612, 621, 625, 651
Crossover frequency 335, 500–501
Crown 622–623
Crystal Palace 25
Curtain 706–707, 710, 712, 719, 726, 728–729, 770–771
Curtain wall 543, 567–569
Cutoff frequency 229, 290–291
Cycle 37, 40, 46, 58–59

D

Dafne 17
Damped natural frequency 384
Damping 124, 274, 325–326, 330, 332, 334–335, 339, 369, 384–388, 390–391, 405, 407–408, 410, 412–413, 415–416
  coefficient 325–326, 384–385, 387, 410
  critical 385
  Coulomb 384
  factor 387, 408, 415
  ratio 385
  under 384, 410
  viscous 383–385, 387, 389, 391
Damrosch, Walter 32
D'Antonio, Peter 755
d'Arsinval galvanometer 124
Dauthe, Johann Carl Friedrich 22
David, Jacques Louis 21
Davies Hall 720
Davis (1957) 293
Davis, Don & Carolyn 107, 224, 298, 308, 595, 624, 630, 780

Day-night level (DNL) 101–104, 106, 132, 192, 196
De Architectura 5
de Boer, K. 112
Decay time 301, 312
De Forest, Lee 741
Degrees of freedom 383
Degree of privacy 544, 548, 550, 558, 570
Delany, M. E. 266, 482
Delsasso, Leo 245, 705, 782
Delta function 410
Demosthenes 3
Density 20, 50–54, 56–57, 59, 123, 148–149, 205, 245, 251, 289, 294, 297–300, 302–305, 318, 320, 324, 337, 365–366, 386, 475, 487, 556, 590–593, 596, 599, 624, 723, 742, 769, 797
Density of materials 54
Depth-to-opening ratio 702
Derngate Centre 729
Diaphragmatic absorber 615, 724, 766
Dickson, W. K. L. 742
Difference limen 88
Diffraction 74, 161, 240, 242, 243, 244, 249, 255, 304, 548, 555, 667, 682, 803, 808
  edge 244
Diffuse field 120, 254–256, 265, 267, 269, 276, 279, 298–300, 304, 306–307, 309, 315–318, 322, 325–326, 328, 331, 333, 351, 353, 356, 366–367, 500, 597, 600, 602, 678, 796
Diffuser 225, 282–283, 464, 474–477, 490, 506, 508, 559, 567, 606, 681, 754, 756–757, 764–765, 767, 770, 772, 798
Diffusion 28, 31–32, 170, 244–245, 247, 282, 306, 313, 352, 655, 657, 667–669, 677, 681, 685, 697, 710–711, 720, 724–728, 747, 753–754, 756, 765, 770–771, 773, 775, 777, 779–780, 784, 787, 798–799, 801–803
  Coefficient 176, 200, 241–242, 244, 249–261, 264, 267, 269, 272–273, 279, 292–293, 300–304, 309–310, 312, 325–326, 347, 356, 381, 384–385, 387, 410, 472–473, 475, 495, 549, 556–558, 564, 591, 605, 662, 668, 675–678, 700, 702, 726, 754, 756, 783–784, 787, 790, 792, 795–799, 801–803
Dimarogonas 3
Dipole 210, 228
Dirac delta function 591
Direct field 254, 304–307, 311, 313, 318, 325, 351–354, 356–357, 365–366, 453, 494, 544, 549, 553, 579, 583, 587, 589–590, 593–594, 596, 599, 603–605, 614, 615, 618–620, 629–632, 646, 650, 668, 678, 736, 783, 796
Direct field transmission loss 325, 326, 357, 365, 366, 544, 549
Directionality 67, 68
Directivity 3, 67, 68, 69, 72, 118, 120, 126, 154, 157, 210, 211, 212, 213, 214, 216, 220, 222, 223, 224, 227, 228, 230, 231, 232, 233, 235, 236, 352, 454, 456, 494, 495, 544, 548, 551, 552, 555, 556, 588, 591, 593, 594, 595, 597, 603, 605, 611, 612, 618, 625, 630, 635, 637, 638, 640, 641, 642, 643, 644, 646, 647, 649, 653, 705, 715, 735, 742, 749, 782, 785, 787, 796, 801
  Index 67, 68, 210, 212, 213, 222, 224, 630, 638, 640, 641
  on-axis (Q_0) 69, 638, 642, 644
Direct sound pressure level 591, 619
Direct-to-reverberant ratio 616, 618, 619, 808
Disney, Walt 742
Dissonance 78, 79, 93
Doelle, Leslie L. 269, 270, 271, 274, 279, 579, 580, 581, 585, 586, 589, 608, 653, 701, 754
Door 332, 353–356, 369, 372–375, 378–379, 495, 515, 538, 563, 576, 607, 682, 690–691, 707, 719, 728, 734, 741, 759, 769, 772–773, 776
  garage 541, 769
Doppler effect 182–183
Double wall 340
Double panel 333–339, 342–347, 365, 370, 522
Double-stud construction 360, 362–363, 365, 375–376, 516, 518–520, 529, 531, 541, 769
Doublet 208–211, 217
Down-shot air handler 571, 574
Drouin, C. 441
Drywall 202, 233–234, 256, 261, 278–279, 326, 340, 358, 360, 363–365, 369, 375–376, 402, 462, 488, 492, 507–508, 516–523, 525–529, 531–532, 535–536, 538, 540–541, 563, 567, 571, 574, 608, 680, 701, 715, 735, 747–749, 761, 767, 769, 772, 777, 797
Drywall screw 358, 364
Dubber 742, 774
Dubbing stage 758, 774
Dubout 751
Duct 50, 64, 210, 255, 259, 270, 285, 289–293, 357, 378, 402–403, 453, 464, 466–472, 474, 476, 481–484, 486–493, 495–504, 506–508, 521, 538–540, 567–568, 572, 574, 576, 761–763, 769, 776, 778
  branch 470–472, 488, 490, 535, 576
  breakin 503–504, 576
  breakout 488, 495, 497–498, 500–501, 503–504, 508, 574, 576, 763
  changes in area 285, 292–293
  circular(round)duct 291, 486–488, 491
  cutoff 291, 481, 490, 495, 504
  cutoff frequency 290–291, 293–294, 481, 492, 495, 504
  elbow 401, 456, 466, 469–473, 481, 492, 495, 504–505, 508, 529, 538, 572
  end effect 481, 488, 490
  fiberglass 378, 468–469, 481–482, 486–488, 491, 493, 496, 500
  lined 291, 378, 462, 468–469, 483–484, 491, 493, 496, 500, 504, 507, 538–539, 567–569, 571–572, 576, 763, 776, 778
  loop 402, 572
  obround (oval) 502
  rectangular 289, 291, 467, 469, 483–484, 486–488, 490–491, 493, 500–502, 506
  silencer 292, 356, 378, 453, 461–462, 480–481, 483, 488, 495–498, 504–505, 507–508, 538–539, 567, 571–572, 574, 576, 578, 763, 769–770
  split loss 490
  transmission loss 462, 486, 494–495, 497–504
Duhamel integral 411, 804
Dummy head 114, 668, 736
Dunlavy, John 765
DuPree, Russ 445, 446
Duration correction 133, 135, 137, 144
Dynamic insertion loss (DIL) 356, 496
Dynamic pendulum 407

E

Ear 31, 46, 50, 73–78, 82–83, 90, 93, 105, 107, 110, 218, 546, 581, 599, 607, 675–676, 736, 752, 808–810
  drum 46, 73–74
  inner ear 73–75
  middle 73–74, 76
  outer ear 73
Early decay time (EDT) 312
Early reflections 15, 110, 313, 589, 594, 607, 620, 654, 665, 671, 673–674, 684, 688, 691, 698, 707, 710, 720, 728, 734, 736, 740, 751, 767, 782
Early signal-to-noise ratio 672–673
EASE 639
Eave vent 378
Echo 579, 587–588, 590, 594, 607, 611, 654, 657, 700, 714, 719, 725–726, 751–753, 756, 772–773, 779
Echo location 51
Edison, Thomas A. 31, 228, 741–742
Edwin Thomas Hall 729
Egan, M. David 300, 702
Electrical box 357, 376–377, 520
Electrical connection 394, 402
Electrovoice 119, 230, 232
Elevator shaft 541
Emergency generator 577–578
Emergency page 562
Endolymphatic fluid 76
Energy density 59, 294, 298–300, 302, 304–305, 365, 590–593, 599
Energy-time curve (ETC) 146
Envelopment 654–655, 658, 660–661, 668–669, 675–677, 681, 688, 703, 710, 712, 720, 722, 725, 728, 730, 740, 756, 777, 780
Environmental
  Impact Report 90
  Impact Statement 90
  Protection Agency (EPA) 101–106, 138, 514
EPA Levels Document 101–106
EPA Model Noise Ordinance 514
Epidaurus Theater 2
Equalizer 631, 632
Equal-loudness contours 68, 83
Equation of state 53, 123, 201, 205, 262
  adiabatic 53
  isothermal 261–262, 279
Equation of continuity 205
Equivalent level (L_{eq}) 101, 137, 157–161, 163, 167–168, 188
Ergodic hypothesis 82
Euclid 3
Euler, Leonhard 20–21, 44
Euler's equation 148, 206
Euridice 17
Euripides 4
European music 1
Eustation tube 74
Evanescent wave 290
Evans, L. B. 175
Everest, Aton 218, 580, 588, 636, 747, 751–752, 757
Expansion chamber 292–294
Eyring, Carl 301–304

F

Fahy, Frank 323, 325–327
Fairbanks rhyme test 602
Fan 456–464
  forward curved blade 459–461, 504
Fan coil unit 451, 462, 488, 536–539, 576–577, 763, 769
Fan-shaped rooms 617, 687, 698–700, 705–706, 714, 717
Farina, A. 948, 960, 964, 967–969
Farmer, William 773
Fast Fourier transform (FFT) 145
Feedback 120, 213, 606, 611–612, 615, 618, 626–627, 629–632, 636–637, 644, 703, 714, 724, 734–739, 772
  margin of stability 630
Festspielhaus 25–26
Fiberglass batt insulation 335, 517
Fidelity 612, 637
Field Impact Insulation Class (FIIC) 513–514
Field Sound Transmission Class (FSTC) 318–319, 356–357, 510, 512–513, 563–564, 566
Film projector 766, 767
Filter 77, 84–85, 93, 121, 125, 127, 133–134, 137, 145–146, 149, 203, 217–218, 476, 560, 631–632, 637, 752
  constant bandwidth 632
  constant percentage bandwidth 631
  low-pass 636
  notch 615
  parametric 631–632
Finne, P. 244
First reflection 671, 753, 777
Flanking 318, 357, 361–362, 365, 375–376, 488, 508, 518–519, 562, 564, 567–569, 761
Fletcher, Harvey 7–9, 12–14, 82–84, 92–93, 95, 592
Fletcher Munson curves 82–84
Flexible coupling 536–537
Floating floor 202, 362, 393–394, 541, 761
  resonantly reacting 440
  locally reacting 796
Floor noise
  airborne 509–510, 513, 522
  footfall 407, 409, 412, 413, 414, 415, 509, 513, 519, 522, 528, 540, 543, 607, 665, 759, 771
  structural deflection 360, 394, 522–523
  squeak 522, 527
Floor ceiling 338, 357, 360, 378, 477, 488, 509, 513–515, 522–526, 535
Floor squeak 522, 527
Floor vibrations 407, 413
Floor wood 331, 339, 383, 386, 407, 514, 518–519, 522–529, 660, 664, 679–680, 685, 692, 724–725
Flutter 268, 312, 579, 587–588, 590, 607, 700, 714, 725–726, 752–753, 758, 764–765, 771–773, 779
Footfall 509, 513, 519, 522, 528, 540, 543, 607, 759, 771
Flow resistance 261–263, 266–267, 269, 273–274, 276–277, 355, 497
Flow velocity 401
Flutter echo 268, 312, 579, 587–588, 590, 607, 700, 714, 725–726, 752–753, 758, 764–765, 771–773, 779
Fly tower 710, 712, 720, 723
Foam lead sheet 533, 535, 538
Focusing 168, 177, 180–181, 247, 312, 579, 587–590, 610, 655, 661, 665, 714, 716, 808
Fohi 1
Foley 742, 758–759, 772–773

Footfall 407, 409, 412, 413, 414, 415, 509, 513, 519, 522, 528, 540, 543, 607, 665, 759, 771
Forsyth, Michael 17–23, 25, 32
Fourier
  analysis 145
  Joseph 145
  series 145, 383
  transform 145
Franzoni, Linda 308–311
Free field 66
French, 95, 592
Frenette, Jasmin 739
Frequency 37–42, 45–49, 51–52, 56, 64, 73, 76–79, 81–88, 90, 92–95, 99–100, 105, 110, 115–116, 118, 120–121, 125, 127–128, 133, 137–140, 143, 145–146, 149, 151, 162–163, 171–173, 175, 182–183, 188, 194–195, 199, 201–204, 206, 208, 210, 213, 216–217, 220, 222, 224, 227–233, 237, 239, 244–246, 252, 255–261, 264, 267–270, 273–274, 277–279, 281–283, 286–288, 291–299, 303, 317, 320, 322–328, 330–335, 338–340, 343–348, 360, 366, 369–371
Frequency response 35, 83, 85, 105, 115–116, 118, 120, 128, 146, 202, 204, 224, 227–228, 232, 245, 348, 615, 632, 637–638, 663, 702, 734, 808
Fresnel
  number 161–163, 165, 188
  integral 241
Frey, A. R. 54, 84
Fricke, F. R. 666, 681, 726
Fry, Alan 497
Fugue 16, 19

G

Gabrielli, A. 11, 16
Gade, A. C. 148
Gain before feedback 611–612, 615, 627, 629–632, 644, 736–738
Galileo Galilei 20
Gallini, Giovanni Andrea 21
Galt, R. H. 592
Garbage disposal 540
Garity, Wm. E. 742
Garnier, Jean Louis Charles 25
Geerdes, H. P. 759, 761
Geometrical (G) factor 367–368
Geometrical mean distance 186
Geometrical spreading 580
Germain, Sophie 21
Gewandhaus 22–23, 27–28
Gibbs, B. M. 348
Gilliom, John 230
Ginn, K. B. 264
Glare 725, 728
Glassner, Andrew S. 795
Glass 327, 332, 335, 339, 341, 344, 369–374, 377–379, 386, 554, 568, 665, 680, 767, 772, 779, 782
Globe Theater 15
Gordon, J. E. 3
Gothic cathedrals 8, 10–11
Grazing attenuation 2, 28, 175–177, 282, 580, 582, 589, 607, 656, 664
Greco-Hellenistic theater 2
Greek Chorus 3, 4
Greek period 1–5, 8, 10–11, 20, 25
Green, David W. 357, 360
Gregorian chant 1, 10
Grieg, Edvard 25
Griesinger 736–739
Gropius, Martin K. P. 27
Gross, Ervin 62
Grosser Musikvereinssaal 28–29
Ground attenuation 157, 163, 166, 175, 185, 188, 199, 282
Grout, Donald Jay 18, 21, 23
Grozier 782
Guarneri, Giuseppi 19
Guillen, Michael 20
Gutenberg, B. 180
Gypsum board 327, 330–331, 339–340, 357, 375, 378, 518, 558–559, 569, 586, 724

H

Haan, C. H. 726
Haas, Helmut 107, 109, 749, 751
  effect 107
  region 154
  zone 109
Handel, George Fredrick 22, 81
Hagia Sophia (see St. Sophia)
Hair cells (stereocilia) 73, 76, 77, 105
Haines, Eric 787, 790–791, 795
Hamet, J. F. 188
Hamilton Place 712–713
Handel, George Frederick 22, 81
Hanover Square Rooms 21–22
Hanrahan, Pat 793–794
Harmonic motion 40, 45, 381
Harmonic 23, 38, 40, 45, 79, 81, 93, 145, 201, 246, 286, 289, 381–382, 387–388, 477–478, 654, 658
Harris, D. A. 339
Harris, Cyril 52, 196, 412, 541, 586, 593
Harrison, Wallace K. 32
Harvard-Westlake School 707, 709
Hat channel 419
Hawkins, J. N. A. 742
Haydn, Franz Joseph 21–23, 27, 81
Head-Related-Transfer-Function 810–811
Hearing loss 77, 104–106
Heat pump 571, 576
Heckel, 323
Heel drop test 427
Helicotrema 76–77
Helmholtz
  resonator 203–204, 234, 261, 268, 277–279, 338, 348, 481, 615, 688, 746, 748, 767, 774, 777–779
  neckless resonator 203–204
Helmholtz, Hermann von 31, 78, 81
Hemming, Roy 14, 16
Henricksen, Cliff 222–223, 231
Henry, Joseph 107
Herodotos 3
Hertz (cycles per second) 37
Hertz, Heinrich 37
Hessler, George F. 355
Heyser, Richard 145
Hill, F. S. 787, 795
Hindley, Goeffrey 14, 16
Hirtle, P. W. 759, 761

Hodgson, M. R. 308
Holmer, Curtis I. 366
Holywell Music Room 22
Homer 3
Home studio 768
Homogeneous coordinates 645
Hooke, Robert 20
Hooke's law 199
Hoover, Robert 175
Hopkins, 306, 308
Horn
  conical throat 228–232, 234
  constant directivity 228, 230–233
  exponential 228–232
  impedance 227–229, 231
  quasi-exponential 231–232
Hourly noise level (HNL) 101
Housing and Urban Development (HUD) 515
Houtgast, T. 149–155, 591, 596–597
Hudspeth, A. J. 75–76
Human ear 73, 105, 115
Human hearing 73, 78, 82, 105
Hum Studio 771–772, 776
Hunt, Frederick V. 3, 31
HVAC 64, 67, 88, 137, 140, 172, 378, 451, 453, 457, 462, 467, 474, 476, 497, 505–508, 537, 552, 567, 574, 586, 590, 607, 698, 759, 762–763, 766–767
Hybrid model 786–787, 803
Hydraulic system 402

I

IAC Corp. 485, 497
Imaging 756, 758, 777
Image source 234, 236, 299, 607, 624, 786
Image tree 786
Impact Insulation Class (IIC) 513–514, 523, 528
Impact noise 457, 513, 523, 529, 540–541
Impedance 55–58,.61, 74, 114–115, 167, 225–229, 231, 250–256, 262–267, 272–276, 282, 285, 288, 319–320, 322–326, 330, 333, 342, 347, 355, 376, 395, 413, 482, 486, 490, 519, 528, 616, 622, 637, 663, 702, 796–797
  bending wave 55–56
  characteristic 55, 57, 61
  complex 430
  driving point 430
  mechanical 395
  normalized 229
  normal acoustic 252
  shear wave 418
  specific acoustic 55–58, 252, 262, 430
  transmission 252–253
  tube 225, 229
Implicit function 788–789, 792, 795
Impression 599, 606
Impulse force 409–410
Impulse response 410–411, 590–592, 596–597, 600, 783, 785, 799, 804–805, 810, 812
Incus (anvil) 74
Ingard, K. U. 262, 268–269, 281, 481, 483–485
Initial time delay gap 654–655, 657, 668, 671–673
Integrated Noise Model (INM) 197
Intelligibility 611, 617–621, 634–635, 644, 668, 672–673, 697, 702, 705, 714–715, 719–720, 739
Intensity 57–61, 64, 68–72, 81–83, 88, 90, 100, 106, 111, 148–149, 158, 175, 220, 224, 235, 240–241, 247, 281, 304–305, 309, 311, 315, 351, 355, 365, 494, 503, 583, 596, 821, 642, 644, 782, 787, 796, 799, 801–802
  level 60–61, 64, 70–71, 83
Interaural cross-correlation coefficient (IACC) 668–669, 675–677, 683, 685–686, 688, 695
Interaural delay time 110
Interaural level difference 807
Interaural time difference 110, 807
Ionian School 3
IRCAM, Paris 729–730
Isodorus of Miletus 9
Isotropic plate 323
Izenour, George C. 2–7, 729

J

Jacob, K. D. 595, 602
Jacuzzi tub 540–541
JBL 230, 232, 620–621, 638–639, 651
Jesse Jones Hall 729
John Arin Center 722
Johnson, Russell 712–713
Joist 386, 408, 418, 420, 495, 515–516, 519, 524, 526–527
  hanger 418
  long-span 360
  truss 449
Jongleurs 13
Jordan, Vilhelm 32
Josse, R. 330
Just noticeable differences (jnd) 88–90
Justinian 9

K

Keele, D. B. (Don), Jr. 230–232
Keller, Arthur 741
Kinetophone 742
King Louis VI 11
Kinsler, L. E. 54, 73, 78, 82, 84, 87, 124, 172, 180, 205, 217, 219–220, 225–226, 288
Kirchoff, Gustav 280
Kirchoff-Fresnel approximation 241
Kishinaga, Shinji 756
Klein, W. 587, 595
Klepper, David 720, 759–760, 762–763
Klipsch, Paul 230
Knudsen, Vern 245, 299, 586, 593–594, 661, 705, 729, 782
Kok, B. H. M. 594
Kopec, John 427
Koz, Jeff 771
Krokstad, A. 737–738
Kryter, Karl 84–85, 96, 98–100, 592
K-space 296
Kurt, 109
Kurze, U. J. 165
Kuttruff, Heinrich 108–110, 113, 252, 263–264, 272–273, 299

L

Labrozzi, D. S. 308
La Scalla (see Theatro Alla Scalla)

Lagrange, J. L. 20
Lambert's cosine law 787, 801, 803
Laminated glass 332, 339–341, 365, 369–370, 372, 378, 386
Lamure, 330
Lane, C. E. 92
Lange, 435
Laplace operator 207
Laplace, Pierre Simon 20
LARES 739
Lateral energy fraction 726
Law of the first arrival 625
Leak 353–354, 371–373, 375, 394, 402, 451
Lecture hall 580–581, 583, 585, 587, 607–609, 614, 698, 703, 705–706, 710
Leissa, A. W. 330
Leizer, I. G. 241
Lenzen, K. H. 408
Leo III 10
Leonardo da Vinci 20
Leonard, Robert 245, 705
Leonin, 11
Level
  ambient level 514
  A-weighted 121, 125, 135, 137, 141–142, 144, 188–189, 454, 457, 477, 552, 602, 759
  direct-field 579, 587, 594, 603, 605, 614, 619–620, 622, 629–631, 646, 649–650, 668, 783
  energy-average level 95, 101
  loudness 61, 68, 81–85
  sound intensity 60–61, 64, 71
  sound power 58, 60, 62–70, 72, 85, 157, 181, 209, 227, 235, 306–307, 316, 356, 401, 425, 453–455, 459–462, 464–466, 470, 472–473, 475, 478, 480, 495–498, 499, 504–505, 538–539, 574, 576, 587, 603, 621, 642, 644, 670
  sound pressure 60–61, 64–72, 85, 87–88, 94, 100, 107, 115, 118, 127, 141, 158, 181, 183, 189, 227, 235, 237, 286, 298, 300, 306–307, 315–316, 318, 352, 354–356, 365, 379, 425–426, 453–454, 456–458, 476–478, 495, 500, 504, 546, 548, 564, 574, 591, 612, 614, 616, 619, 629, 638, 642, 646, 670
Levelt, W. J. M. 79
Lexicon 739
Light fixture 559–560
Limiting distance 594–595
Line array 612, 625
Line connection 516, 525
Line source 70–71, 159–161, 163, 165–167, 185–186, 188, 194, 213–216, 225, 367–369, 500, 625, 664, 722
  partial 159–160, 165, 194
Lindsay, Bruce 20–21, 31
Listening room 741, 743, 745, 748–749, 751–752, 756–758, 761–763, 765–766
Liszt, Franz 25
Liveness 621
Locally reacting surface 440
Lochner, J. P. A. 600, 602–603
London, A. 333
Long, Marshall 71, 112, 220, 233, 351, 378, 586, 609, 708–709, 716, 718, 721, 723
Long-delayed reflections 579, 587–588, 589–590, 611, 665, 700, 714
Longitudinal waves 442, 528, 535
Long-span floor system 331, 360, 395, 405, 415, 420
Loop gain 736–737, 739
Looping 742
Loudness 3, 25–26, 61, 66, 76, 78, 81–85, 98, 113, 133–134, 137, 140–142, 579–580, 583, 587, 611–612, 614, 653–654, 656–658, 665, 667–668, 671, 678, 685, 697, 707, 737, 749, 757, 764
Loudness corrections 61, 64, 102–103, 120–121, 123–124, 132–134, 136–137, 141–142, 185, 193
  duration 136
  onset 75, 99
  pure tone content 133
  source number 133
  time of day 102, 131
  variability 135
Loudspeaker 31, 35, 50, 67–69, 109, 113, 115, 124, 145–147, 154, 203, 210, 212–214, 217, 219–220, 222, 224–228, 230, 234, 236, 276, 316, 348, 366, 395, 417, 426, 522, 561–562, 571, 586, 588, 595–596, 602–603, 606, 609, 611–627, 629–631, 633–635, 638–640, 642, 644, 646–650, 653, 667–668, 697, 703, 705, 707, 714–717, 720, 734–736, 738–739, 742, 744–749, 751–753, 756–758, 765–767, 771, 774–775, 777, 780, 799, 808–810, 812
  aim point 622, 626, 647–648
  baffle 222, 224
  bass-reflex 203–204
  central cluster 617–618, 625, 634, 705, 715, 735, 809
  cluster 613, 615, 617–619, 624–625, 626–627, 635
  coaxial 612
  cone 115, 202–203, 210, 221–222, 226–228, 612, 614–616, 619, 621, 624–625
  continuous line array 213
  coverage 611–612, 615, 617–619, 625, 630, 632, 638, 641, 644
  coverage angle 612, 625
  cross-fire cluster 618
  curved line array 214
  diaphragm 612, 615, 624–625, 636
  directional characteristic 119, 210–211, 214–216, 220–222, 224–225
  directivity 118, 120, 126, 154, 157, 210–214, 216, 220, 222, 224, 227–228, 230–233, 235–236, 595, 603, 611–612, 618, 625, 630, 635, 637–644, 646–647, 649, 653, 715, 735, 742
  distributed system 615–619, 624, 626, 634, 657, 735, 744
  efficiency 203, 228, 230, 611, 614–615, 622
  horn 213–214, 216, 220, 222, 225, 228–233, 236, 612, 614, 616, 618–619, 621, 624, 658, 688, 717, 735, 767
  latitude 638–639, 641–642, 649
  line array 212–214, 224, 233, 612
  longitude 638–639, 641–642, 649
  multiple cluster 615, 617, 627, 634
  on-axis sensitivity 67, 638, 644
  phased array 217
  phasing plug 612
  surround 742, 744–745, 756, 758, 765–767, 770–771, 777, 780, 809
  tapered line array 214
Lower gallery (scala tympani) 76
Luther, Martin 14

M

Mace, Thomas 19
Madsen, E. R. 111
Maekawa, Z. 161
Magnetic seal 375
Magnetic tape 742
Markin, Vladislav S. 75–76
Marshall, A. Harold 759–760, 762–763
Masking
  spectrum (MS) 545–546, 548–552, 560–562, 564
  spectrum rating (MSR) 548
Mason 400
Mass
  absorption 405, 407
  damping 405, 407
  law 355, 364, 366, 373, 378, 418, 421, 501, 509
  loading 331
  reactance 333, 335
Maxfield, 593–594
McDermott Hall 734
McLachlan, N. W. 224
Meadows School 707–708
Mean free path 50, 299, 302
Mean time between reflections 299
Meatus 73
Mechanical equipment room 355, 492, 541, 571–572, 574
Mechel, Fridolin P. 262
de'Medici, Maria 17
Meister, F. J. 407–408
Mels 81
Mendelssohn, Felix 22, 25
Mersenne, Marin 20
Masking 78, 91–93, 95, 109
Metal stud 345, 357, 359–361
Metropolitan Opera House 31, 33
Meyer, 113
Meyer, J. 22
Meyerbeer, Giacomo 81
Michelangelo Buonarroti 14
Microphone 31, 35, 50, 64–65, 114–122, 124, 126–128, 145–146, 148–149, 155, 210, 217, 593, 618, 626–627, 633, 635–638, 655, 665, 677, 681, 714, 719, 725, 736–739, 741, 744, 749, 759, 771, 773, 782–783, 799, 809
  cardioid 630–631, 636
  ceramic 115–117
  condenser 115–118
  directional 612, 614, 616–619, 621, 627, 630–631, 635–637, 651
  dynamic 115–116, 130, 148
  electret 115–117
  hum-bucker coil 637
  hypercardioid 631
  omnidirectional 630, 636–637
  phantom power 637
  ribbon 115–118
  sensitivity 115, 118, 120, 124
  shotgun 120
  wireless 638
Middle gallery (scala media) 76
Millington, G. 302
Millington Sette equation 302
Mindlin, R. D. 323
Minimum code standards 510–515, 528
Minnesingers 13
Mitchell, 777
Mixer
  live 631, 771, 778
  automatic 618, 631
  manual mixer 618
Mixing 707, 719, 736–738, 742, 744–745, 759, 769, 771–772, 774–775, 777
Moderow, R. R. 415
Modulation reduction factor 149–151, 153
Modulation transfer function (MTF) 149–151, 154
Mohammed 9
Mommentz, E. 798
Monastic seating 713
Monopole 208
Monteverdi, Claudio Giovanni Antonio 11, 17
Morse, P. M. 225, 280
Motion picture 31, 113, 586, 609–611, 701, 742
Mozart, Wolfgang A, 21, 23, 27, 81, 585, 674
Muffler
  reactive 292
Muller 277–278, 666, 674
Multifamily dwelling 102, 427, 509–510, 512, 515
Multi-modal wave 481
Multiuse theater 698
Muncey, R. W. 751
Munson, W. A. 82, 83–84, 93, 140
Murphy, Shawn 744, 771
Musikvereinssaal 28–29, 663, 670, 677–678, 682–683, 781

N

Nail 115, 418, 446–447, 449, 524, 527–528
Napoleon Bonaparte 21
Narrow-band 125, 142–143, 145–146, 249, 282, 632, 756
National Bureau of Standards 369–371, 374
Natural frequency 73, 81, 199, 201–204, 277, 383–385, 387, 389, 392, 394, 405, 410, 416, 423, 437, 444, 528, 716
Naylor, G. M. 799
NC level 86–87, 759
NIC 318–319, 510, 570–571, 761
Nickson, A. F. B. 751
Noise ordinance 514, 768
Normal mode 294–296, 298, 407, 433, 730, 745
Neoprene vibration isolator 261, 375, 392–394, 419–420, 423, 522, 525–526, 533, 535–536, 540–542, 776, 778
Neues Gewandhaus 23, 27–28
Newman, Ray 230
Newton, Isaac 20, 53, 56, 148, 200, 205–206, 262, 271, 273, 319, 384, 410
Nightingale, T. R. T. 375–377, 518
Nishihara, Noriko 797–798, 805
Noise
  airborne 417–418, 420, 427, 451, 453, 510, 513, 522, 536, 759
  background 3, 18, 73, 93–95, 126–128, 135, 149, 510, 514, 544, 550, 560, 563–564, 579, 587, 590, 593–597, 599, 600, 603–608, 610, 620, 622, 633, 655, 657, 674, 698, 758–759, 762, 773
  criteria (NC) 85, 89, 318, 574
  diurnal 130
  dose 106–107
  Isolation Class (NIC) 318
  masking 93, 543–545, 548, 550, 558–561, 587, 590

pink 40, 560, 622, 632
reduction 102, 149–151, 153, 166, 292, 317–318, 339, 341, 353, 378, 390, 465, 510, 514, 534, 537, 543, 545, 548–551, 554–556, 558–560, 563–564, 595, 667, 670, 674, 759, 762
reduction coefficient (NRC) 255, 557
structure-borne 432, 434
Noise Control Act of 1972 101
Noise Exposure Forecast (NEF) 142, 144
Noise metric 130
broadband 73, 85, 132, 134, 137, 217, 226, 230, 265, 269, 393, 405, 736, 747
diurnal 130
narrow band 133
Noise pollution level 135–136
Noisiness 98–100
Nom correction 631, 737
Normal acoustic impedance 18, 482
Normal distribution (Gaussian) 159
Normalized impact sound level 435–436, 439
Normal mode 298
Normal specific acoustic impedance 252, 482
Norris, R. F. 367
Norris Eyring equation 367–369, 371, 597
Notre Dame Cathedral 14–16
Noys 118
Number of open microphones (nom) 761, 762

O

Oblique mode 746
Occupational Safety and Health Administration (OSHA) 100, 106–107
Octave 40, 42, 64, 67, 77, 79, 81, 84–87, 90, 94–95, 98, 121, 125, 127, 133, 137–143, 145–146, 149, 151, 153, 155, 176, 188–189, 213–214, 228, 255–256, 268, 289, 317, 325–326, 328, 330, 333–335, 343, 356–357, 379, 425–426, 435, 439–440, 454, 456, 457–458, 460–462, 470, 472–473, 475, 478, 480, 486–487, 489, 491–492, 496–497, 501–502, 505–508, 510, 544–548, 550, 556, 561, 592, 595, 614, 619, 631–632, 651, 662, 673, 677–678, 734, 759
Odea 4
Odeon of Agrippa 4, 6
Odyssey 3
Office
building 541
closed 550, 563, 570–571
open 544, 550, 552, 556, 558, 560, 562
private 375, 560, 562, 564
Ogawa 308
Olive, Sean 109, 111, 750–751
Ollerhead, J. B. 98
Olson, Harry 211, 214–216, 221, 224, 229, 736
Olympic Academy 15
Openings 10, 76, 120, 166, 204, 278, 355, 357–358, 366, 369, 375, 378, 417, 451, 495, 520–521, 526, 564, 569, 615, 660, 706, 748, 761, 767, 776, 778–779
Open-loop gain 630
Opera 16–18, 23, 25, 585, 656, 660, 666, 675, 691, 712
Opera house 17–18, 20, 23, 25, 31, 33, 168, 175, 282, 608, 656–657, 659–661, 663, 665–667, 674, 682, 685, 688, 691–693, 698, 700–702, 715, 725, 771, 774
Oratorio 16, 18
Orchestra 4, 15, 17, 19, 22–23, 25–33, 240, 282, 583, 619, 626, 654–656, 658–659, 661, 663–666, 671, 673–674, 678–679, 681, 684–685, 687–688, 691, 693, 695, 702–703, 706–707, 710, 712, 720–725, 727, 734, 737, 742, 744, 758, 764, 770–771, 779
pit 665–666, 706–707, 712, 742
shell 33, 656, 707, 710, 712, 720, 724
Ordway Theater 724
Orfeo 17
Organ 19, 27–28, 51, 73, 76, 285, 289, 386, 585, 656, 665, 675, 684–685, 691, 714, 726
Organ of Corti 76
Organ pipe 285, 289
Organum 11–12
OSHA 100, 106–107
Oscillator 201, 203, 385–386
Oval window 74, 76–77
Overtone 79
Owens Corning 337

P

Padded seats 608, 662
Painting sound-absorbing materials 270
Paisley Park Studio 775
Palisca 21, 24
Palladio, Andrea 15
Palmer, R. 7, 14
Panel
absorber 261, 272–275, 278, 747
array 244–245
finite 240, 248
thick 261, 268, 276, 326, 328, 331, 340, 418, 610, 773
thin 30, 274, 322–324, 326, 330–331, 660
Paris formula 254, 804
Partial articulation indices (PAI) 545
Particle velocity 56, 58, 116, 148–149, 171, 207–208, 237, 239–240, 265, 272, 304, 306, 748
Party wall 513, 515–516, 518–520, 527, 540
Passacaglia 16, 19
Path length difference 162, 217, 246, 624, 636, 651, 671, 783, 808
Paul, Les 742
Paxton, Joseph 25
Peak-to-average ratio 614, 622
Pearsons, K. S. 98, 511
Penetration wall 376, 520–521
Penn State University 608–609
Perceived noise level (PNdB) 98, 142, 144
Perfect intervals 80
Perforated panel 204, 274–278
Peri 17
Pericles of Athens 4
Period
Roman 1
Christian 7
Baroque 10
Romanesque 10
Gothic 11
Rennaissance 14
Classical 21
Romantic 21
Permanent threshold shift (PTS) 105
Perotin 12

Peterson, Arnold P. G. 62–63, 124
Peutz, Victor 96, 587, 594–595, 620
Pewback system 615, 618
Phase 45–48, 59, 71, 118, 120, 146, 162, 171, 175, 177, 199, 201, 203, 208–213, 217, 219, 224–226, 235, 237, 245, 250–252, 265, 279, 285, 337, 383, 405, 477, 623, 630–631, 637, 650–651, 752
Phase angle 45–46, 48, 237, 251, 383
Philharmonic Hall 246, 658, 687–688
Phon 83–84
Phonograph 31, 228, 741
Piano 19, 25, 38, 39–40, 42, 51, 81, 90, 175, 688, 714, 772
Pierce, Allan D. 38–39, 41, 78–80, 89–90, 172–173, 250
Piercy, J. D. 173
Pies, Donald B. 190–193, 195
Pink noise 40, 560, 622, 632
Pinna 73–74, 808–809
Pinnington, R. J. 432
Pipe 3, 19–20, 50–51, 285, 288–289, 357, 386, 401–403, 451, 456, 481, 496, 520–521, 529–537, 540–541, 665, 684, 691, 714, 726, 746, 761, 779
  flexible coupling 456, 536–537
  penetration 357, 376, 402–403, 520–521, 531, 569, 576, 763
Piping 19, 386, 393, 399, 401–403, 451, 456, 480, 509, 519–520, 529–533, 535–536, 540–541, 759, 761
Piston 50, 52, 56, 123, 198, 202, 219–221, 223–226, 230–232, 262, 280, 347, 488
Pistonphone calibrator 115, 118, 122–123, 130
Pitch 20, 77–82, 418, 458, 739
Pitched-roof flutter echo 752
Plainsong 10, 12
Planar source 71, 219
Plane wave 57–59, 148, 161, 205–206, 208, 236–238, 250, 265, 285–287, 290–293, 304, 319–320, 333, 347, 365–367, 427, 481, 492–493, 495, 807
Plenum 292, 481, 492–495, 504, 507–508, 538, 543, 558, 563–564, 566–569, 571–572, 574, 576, 748, 772, 778
Plomp, R. 79, 596
Plumbing 88, 394, 401–402, 520, 531
Plumbing isolation 402
Plumbing pipe 520, 761
Plumbing noise 102, 344, 509, 513, 529, 532, 761
Point connection 345–347, 525
Point mount 347, 364, 390, 423, 517, 526–528, 776
Point source 65–67, 69, 71, 160–162, 165, 181, 185, 188, 208, 213, 222, 235, 306, 368, 430, 615, 626, 744, 808
Polygon crossing test 791
Polyvinylbutyral 339
Pope Gregory I 10
Pope Gregory II 10
Pope Julius II 14
Pope Urban VII 18
Porosity 177, 270, 275–277, 515
Porous material 430
Post production 741–742, 758, 767
Precedence effect 107–108, 111–112
Perception of direction 110
Preferred Noise Criterion (PNC) 138, 140
Preferred Speech Interference Level (PSIL) 95
Presbycusis 105
Pressure
  atmospheric 53, 56, 59, 73–74, 123, 171, 202
  drop 262, 472, 475–476, 505, 507, 538–539, 576
Privacy 90, 97–99, 166, 509, 544, 548–551, 556, 558–564, 569–571, 590, 593, 605, 760–761, 778
  Confidential 98, 544, 551, 563–564, 569–571
  Index (PI) 548–549, 562, 570–571
  Normal 98, 558
  Transitional 544, 551, 558
Product theorem 222
Program 24, 607, 611, 618, 639, 644, 674, 697–98, 700–701, 707, 734, 753, 785
Project studio 758–759, 769
Projection port 767
Projection screen 606–608
Propagation
  angle 290–291
  constant 265–266, 274, 280, 482–483, 493
Property line ordinance 90, 509, 513–514
Proscenium 33, 617–618, 625, 635, 656, 693, 703–706, 712, 720, 722, 735
Proskenion 4
Proskenium 4, 15
Pulse dampener 537
Pump 128, 399, 401, 451, 456–457, 462–463, 529, 536–538, 540–541, 576
Pyramid beam 785, 800–802, 804
Pythagoras of Athens 79
Pythagoras of Samos 3
Python 3

Q

Quadric surface 793
Quadratic-residue diffuser 283
Quality factor 748
Quirt, J. D. 340–342, 375–377

R

Radial frequency 45, 51, 206, 220, 320, 323, 366, 381, 433
Radian 43, 46
Radiation efficiency 427–428, 430–431, 436
Radio City Music Hall 699
Railroad 182, 189–190, 193–195, 730, 783
Rainer, J. H. 408
Ramsey, Charles G. 581, 702
Raphael (Raffaello Santi) 14
Rapid speech transmission index (RASTI) 149, 155
Rausch, E. 407
Ray cone intersection 794
Ray cylinder intersection 792
Ray paraboloid intersection 794
Ray polygon intersections 790
Ray sphere intersection 791
Ray plane intersection 789
Ray quadric intersection 793
Ray tracing 583, 590, 674, 726, 781, 783, 785, 787, 789, 791, 793, 795, 797, 799, 801, 803–805, 807, 809, 811
Rayleigh, Lord (See John Strutt)
RC level 87, 89
Reagan, J. A. 135, 159–160, 165, 169, 183–185, 187
Real-time analyzers (RTA) 143
Reasonable expectation of quality 509, 512, 528

Recording 31, 35, 85, 89, 112, 114, 118, 125–126, 129–130, 393, 511, 609, 611, 653–655, 719, 729, 741–745, 749, 753, 758–759, 768–769, 771, 773–775, 777, 782, 809
format 741, 744, 756, 765–766
studio 35, 753, 758–759
Redoutensaal 23
Reference quantity 60
Reflection 107, 109, 111, 146, 166–167, 175, 235, 240–247, 249–252, 263, 280, 283, 287, 292, 299, 312, 368, 490, 549, 554, 556, 559, 583–584, 587–589, 600, 607, 636–637, 654–655, 664–665, 670, 671, 681–682, 691, 703, 710, 716, 720, 722, 725, 728, 749–754, 756, 765–767, 777, 779, 782–783, 785–787, 795–799, 801–803, 808
amplitude ratio 237, 250–251
coefficient 250, 252, 292, 797
curved surface 247–248, 583, 704
finite panel 240–241
long delayed 25, 109, 312, 579, 587–590, 611, 654, 665, 670, 700, 714, 752
normal coherent 236
oblique coherent 238, 585, 726, 728, 754
specular 243, 583, 756, 765–766, 777
Reformation 14, 714
Reif, F. 170
Reiher, H. 407–408
Reiher-Meister scale 408
Reissner's membrane 76
Relative loudness 11, 84, 141
Relaxation 168–173, 175
Renaissance 14–16, 20
Resilient
Channel 202, 357–358, 360–361, 378, 420, 516, 523–526, 528, 535
mount 526–527
Resonance 73, 77, 199, 204, 227–228, 234, 261, 273–274, 279, 330, 333–335, 337–339, 341, 343, 346, 349, 360, 363, 379, 386–387, 390, 392, 406–407, 414, 421, 438–440, 444, 446, 500, 534, 579, 664, 779
mass-air-mass 261, 278–279, 333–334, 338, 341, 343, 360, 363, 370, 379, 421, 779
Resonant frequency 199, 202–204, 227–228, 261, 273–274, 277–279, 281, 330, 333, 339, 343, 370, 387, 389–390, 392, 395, 405, 413, 415, 440, 444, 615, 636, 748
Resonator
quarter-wave 279–280, 282
Restaurant 604–605
Rettinger, Michael 297
Reverberant field 163, 239, 304–307, 310–311, 313, 315–316, 351, 354–355, 365–366, 425, 453, 494, 504, 545, 587, 589–591, 593–594, 596, 602–604, 619–621, 631, 657, 666, 670, 677–678, 703, 722, 736, 756, 785, 803
Reverberation 15, 25, 27, 33, 93, 118, 149, 154, 163, 175, 255, 268, 270, 303, 312, 579, 585–590, 607–608, 611, 654–655, 657–658, 661, 666, 668, 674, 678, 680, 691, 697, 701–702, 710, 715, 725, 728, 730, 734, 736, 738, 739–740, 752–754, 764, 771, 773–774, 781
Reverberation time 19, 22–23, 26–27, 29, 31, 97, 109, 150–151, 154, 175, 255, 270, 298, 300–304, 312, 316, 425, 580, 585–587, 590–591, 593–595, 598, 606, 610, 620–621, 654–655, 657–658, 660, 666, 674–675, 678, 682, 688, 691, 697, 701–702, 706–707, 711, 715, 728–730, 736, 738–739, 751–754, 771–773, 777, 782, 804
Reynolds, Doug 471, 473–474, 486–487, 490–491, 493, 501–502
Rheims Cathedral 11
Rindel, J. H. 241–245, 247–248, 785–787, 796–797, 803
Roark, Raymond 443
Robinson-Dadson curves 82
Robinson, D. W. 141
RODS 738
Rolland, Romain 3
Roman basilica 7, 713
Roman period 1
Romanesque period 10
Romantic period 23
Roof-mounted equipment 395, 505, 537, 539–540, 574
Room Criteria (RC) 85, 87–89, 137
Room
constant 303, 306–307, 316, 352, 354, 426, 494–495, 574, 591, 595
divider 569
mode 313, 426, 588, 615, 633, 745–747, 749, 777, 804
Root mean square (rms) 58–59, 125
Roozen, 204
Rosenblith, 100
Rossing, Thomas 81
Rossini, Gioacchino 81
Royce Hall 720, 771, 782
RSIC resilient mount 527
Rugby Hall 707
Rule of 400, 614

S

Sabbioneta 15–16
Sabine equation 301, 303, 311, 316, 425, 674, 678
Sabine, Hale 369
Sabine, Wallace Clement 1, 31, 300–304, 306, 310–312, 316, 425, 495, 666, 670–671, 674, 678, 684–685, 781
Sacculus 76
Sala Nezahualcoyotl 734
Sala Sao Paolo 730, 733
Salieri, Antonio 23
Sandars 1
Sanicola, Hank 748
Sanrio Puroland 744
Sauter, Augustus 349
Scaena 4, 15
Scale
chromatic 79
diatonic 79–80
equal-tempered 80, 90
just 80
pentatonic 79
Pythagorean 80
Scale model 782, 804
Scala media 76
Scamozzi 15
Scattering coefficient 798–799, 801, 803
Schaffer, Mark 399–400, 572, 574
Scharf, B. 77
Schism 10

Schmieden, Heinrich 27
Schodder, G. R. 113
Schroeder, Manfred 282, 298–299, 596, 810, 812
Schroeder frequency 298–299
Scoring stage 744, 758, 770–771
SDI 668, 668–669, 681, 683, 685–686, 688, 726
Seismic restraint 400, 539
Self shielding 118, 367–369
Self noise 496–497, 505–508
Semicircular canals 76
Semi-diffuse field 495
Semi-reverberant field 305
Semitone 79–80
Seraphim 109
Seto, William W. 182–183
Sette, W. J. 302
Shadowing 588, 657, 699
Shakespeare, William 15
Sharp, Ben 164, 322–323, 326, 328, 330, 332–333, 335–339, 343, 345–347, 355, 364, 517
Shear 55, 322–323, 326, 328, 330–332, 340, 375, 418, 429, 533
  bending frequency 328
  frequency 328, 340
  impedance 323
  wave 55, 326, 328, 418, 429
Sherry, Cameron W. 339, 357, 360
Shiner 418, 447, 527
Shirley, Peter 787, 795
Schmieden, Heinrich 27
Seat attenuation 702
Shoebox hall 26, 29, 32
Schubert, Franz 25
Siemens, Ernst W. 31
Sight lines 2, 580–583, 618, 656, 660, 664, 693, 697–699, 700, 705, 707, 710, 717, 719
Signal-to-noise ratio 94–95, 97, 130, 149, 151, 153–155, 543–545, 564, 579, 587, 592–593, 595–600, 602–605, 618–621, 635, 672–673, 715, 735, 778
Silencer 292, 356, 378, 453, 461–462, 480–481, 483, 488, 495–498, 504–505, 507–508, 538–539, 571–572, 574, 576, 578, 763
Single event noise exposure level (SENEL) 137
Single family home 513
Single panel 245, 278, 319, 325–326, 330–333, 339, 363, 370
Skene 4
Skudrzyk, Eugen 1
Slab 55, 328, 340, 378, 390, 395, 399, 402, 407–408, 416–419, 427, 434–437, 440–441, 445–446, 515, 519, 522–524, 541–542, 553, 564, 568–569, 685, 759, 761, 773, 776, 779
Sleeper, Harold 581, 702, 748
Small, R. H. 203
Snell's law 179–180
Society of Motion Picture and Television Engineers (SMPTE) 610, 742
Soft ground falloff 185
Sonata 16, 21
Sone 84, 98
Sophocles 4
Soroka, Walter W. 347–349
Sound attenuation 164, 167–168, 170–171, 423, 481, 543, 549, 551, 563
  binaural 114, 667, 809–810, 812
  exposure level (SEL) 136, 192
  free field 66, 74, 120–122, 127, 306, 354, 367, 670, 678, 798
  masking 78, 91–93, 95, 109, 543–545, 548–550, 552, 558–564, 569–571, 579, 587, 590, 593, 604, 740, 751
  path 163, 177–179, 300, 544, 548, 553, 608, 624, 651, 699
  power 58, 60, 62, 64–65, 70, 72, 85, 214, 235, 305, 425, 428, 466, 477, 499–500, 503–505, 591, 796
  power level 60, 62–70, 72, 157, 181, 209, 227, 235, 306, 316, 356, 401, 425, 435, 453–455, 459–462, 464–465, 469–470, 473, 475–476, 478, 480, 495–499, 504–505, 538–539, 574, 576, 587, 603, 621, 642, 644, 670
  pressure 54, 56, 59–61, 72, 115, 120, 171, 175, 206, 210, 214, 219, 231, 250, 286, 295, 305, 315, 351, 481–482, 636, 677, 745, 749
  pressure doubling 146, 210, 234, 286
  pressure field 120
  pressure level 60–61, 64–72, 85, 87–88, 94, 100, 107, 115, 118, 127, 141, 158, 181, 183, 189, 227, 235, 237, 298, 300, 306–307, 315–316, 318, 352, 354–356, 365, 379, 425–426, 445, 453–454, 456–458, 476–478, 495, 500, 504, 546, 548, 564, 574, 591, 612, 614, 618–619, 629, 638, 642, 646, 670
  random incidence 120, 239, 309–310, 700
  range 73, 654
  reference value 61
  reinforcement 113, 115, 118, 120, 222, 609, 611, 624–625, 627, 632, 635, 637, 700, 707, 719, 751
  spectrum 87, 230, 545, 548
  speed of 54, 56–57, 172, 180, 206, 300, 320, 323, 429, 591
  stage 655, 658, 758, 769
  system 93, 155, 230, 595, 608, 610–612, 614–615, 617–618, 620, 627–629, 632, 634–635, 644, 653, 697, 702–703, 707, 714–715, 717, 734
  transmission 315, 317, 318, 321, 330–331, 338, 345, 347, 351–352, 357, 365–367, 369, 420, 424, 446, 453, 499, 502, 504, 509–510, 512–513, 516, 555–556, 563, 567, 569, 761
  Transmission Class (STC) 317–318, 357, 420, 424, 510, 512–513
Sounding vases 6, 7
Sound level meter 59, 84, 121, 123–124, 129
  detectors 125
  response
    fast 124
    impulse 121, 124–125
    peak 125
    slow 106–107, 124
Soundstage 744, 756–757, 765–767
Source
  moving 100, 120, 129, 157–159
Source-path-receiver model 157
Spandock, 782
Spagnolo, Renato 253
Spark testing 782
Spectrum analyzer 77, 121, 143, 632
Specular absorption coefficient 253, 549, 556, 796, 799
Speech
  absorption rating (SAR) 549
  diffraction rating (SDR) 549, 555

Speech (*continued*)
intelligibility 4, 6, 15, 86, 93–97, 149, 151, 268, 544, 579, 588–590, 592, 594–595, 598, 600–602, 603–604, 609, 611, 618–619, 672–673, 705, 714, 719–720, 739
intelligibility test 97, 588
interference level (SIL) 94–95, 139
privacy 543–544, 550–551, 558–559, 561–563, 570
reduction (SR) 514, 655, 657–659, 661, 666, 668, 671, 672
reduction rating (SRR) 548, 550, 558
Transmission Index (STI) 151, 152, 154, 592, 596–598
transmission rating (STR) 549
Rating Factor (SRF) 546
Specular reflection 243, 583, 728, 754, 765–766, 783, 795–796
Spherical coordinates 207
Spot laminating 332
Spring hanger 419, 422
Spring mass system 200–201, 227, 277, 383–395, 387, 389–392, 395, 405, 410–411, 418, 430, 437
Spring-supported ceiling 422, 528
St. Denis Cathedral 11
St. Mark Cathedral 11
St. Martial Cathedral 11
St. Paul the Apostle Church 716
St. Peter's
Basilica Church 7
Cathedral 14
St. Sophia (Hagia Sophia) 8
ST1 681, 722
Stadt Casino 26
Stage riser 664, 725
Staggered stud wall 361, 517
Staggered wood studs 361
Stairway 713
Standard deviation 135, 159–160, 619, 622, 799
Standing wave 250–251, 286, 295, 478, 746
Stapes (stirrup) 74
State of California 357, 426–427, 515
Static deflection 387, 402, 412, 418, 423
Static pressure 400, 402, 459, 476, 504
Steel, J. A. 518
Steeneken, H. J. M. 149, 150, 151, 154, 155, 596
Steinberg, John C. 95, 592, 593
Stephens, K. N. 100
Stepped blocking 386, 420, 446, 449, 524, 527
Step loudspeaker 618
Stereo 111, 113, 114, 510, 514, 612, 614, 655, 714, 741, 742, 744, 748, 752, 753, 756, 757, 758, 765, 766, 771, 774, 775, 777, 779, 809, 810, 812
cross-talk-canceled 809
image 113, 777, 780
recording 655, 741
Stereocilia 76, 77
Stern, Richard 744
Stevens, S. S. 84, 98, 141
Stiffness 56, 202, 227, 252, 276, 320, 323, 325, 330, 331, 332, 338, 357, 363, 383, 390, 393, 399, 415–417, 423, 437, 440, 446, 515, 524, 526, 540
Stradivari, Antonio 19
Strayer, Joseph R. 7, 10
Strength factor (G) 670
Structural
damping 446, 523
decoupling 456, 518, 524, 526
deflection 360, 394, 417, 418, 420, 522, 523, 569
Studio 35, 118, 338, 341, 424, 698, 719, 744, 745, 749, 753, 758, 759, 768, 769, 770, 772, 773, 774, 775, 776, 777, 778, 779
Strutt, John W. (Lord Rayleigh) 21, 31, 55, 56, 57, 219, 250, 599, 807
Stryker 306, 308
Subjective effect of reflection 588
Subjective preference 667, 670, 675
Subwoofer 234, 236, 756
Suger, Abbot 11
Superposition 20, 46, 47, 383
Surround sound 611, 742, 745, 756, 758, 766, 767, 770, 777, 780, 809
Swallow, John C. 407, 415, 416
Swing, Jack 190, 191, 192, 193, 195
Synagogue 614, 635, 719

T

Tangential mode 295, 746
Tape recorder 85, 126, 137, 741, 742, 775, 782
Tapping machine 408, 417, 424, 425, 434, 435, 436, 437, 438, 439, 445, 448
Tech booth 736
Telemachus 3
Telleman, Georg Philip 19
Temporal forward masking 109
Temporary threshold shift (TTS) 105
Thales of Miletos 3
Theater 1, 2, 4, 5, 6, 14, 15, 16, 17, 18, 19, 32, 113, 282, 300, 580, 581, 583, 609, 614, 617, 626, 635, 656, 657, 663, 664, 666, 698, 700, 702, 703, 705, 706, 707, 710, 711, 712, 723, 724, 725, 729, 742, 744, 758, 766, 774
Multiuse 665, 698
Theatro Alla Scalla 24, 656, 694
Theatro Farnese 17
Theme park 744, 809
Theophrastos of Eresos 4
Thermal inversion layer 178
Thiele, A. N. 203, 599
Third-octave 40, 42, 77, 84, 85, 95, 98, 121, 125, 133, 142, 145, 146, 151, 189, 317, 334, 356, 425, 426, 496, 510, 544, 545, 546, 548, 556, 592, 632, 651
Thirty Years War 16
Thomaskirche, Leipzig 19
Thomson, William T. 385, 387, 388, 410, 411, 412
Threshold
of human hearing 59, 61, 758
of pain 73
shift 91, 93, 105
Thrust stage 665, 706, 707, 710, 712, 725, 735
THX 610
Timbre 654, 655
Time
coincidence 617, 624, 651
delay 111, 113, 145, 217, 218, 626, 627, 651, 657, 668, 671, 763, 752, 808
delay spectrometry (TDS) 145, 146, 248
electronic delay 606, 617, 624, 625, 651
Time-base-zero 617, 626, 627

Toccata 16, 19
Toole, Floyd 109, 111, 749, 750, 751, 756, 765, 780
Townhouse 509, 540
Traffic 90, 91, 129, 130, 131, 132, 133, 135, 136, 183, 184, 186, 188, 189, 378, 379, 515, 590, 606, 769
  distribution 131, 132
  noise
    index (TNI) 136
    standard deviation 135, 136, 159, 169, 185, 188, 619, 622
Transducers 31, 76, 115, 124, 146, 217, 222, 612, 616, 621, 636, 744, 766
Transfer function 627, 629, 408, 810
Transfer gain 627
Transformer 74, 477, 478, 617
Transient vibrations 407
Transmissibility 389, 390, 392, 406
Transmission 285, 303, 304, 308, 331, 337
  coefficient 355, 357
    loss 270, 293, 294, 315, 316, 317, 318, 319, 320, 321, 322, 323, 325–349, 351–364, 366, 367, 369, 370, 371, 373, 374, 375, 376, 377, 378, 462, 486, 488, 494–495, 497–504, 510, 515–516, 519–520, 553, 555, 562–564, 567, 569, 571, 574, 610, 759
  composite 353, 354
  direct field 325, 357, 365–366, 544, 549
  field-incidence 321–322, 326
Transmissivity 276, 320–321, 333, 365, 493
Trash chute 541
Tremolo 79
Triple-panel 342–344, 364, 517, 779
Troubadours 13
Trouveres 13
Tube 20–21, 52, 56, 64, 73–74, 76, 114, 120, 203–204, 225, 229, 247, 250–252, 255, 274–275, 277, 279–282, 285–289, 294, 334, 349, 386, 451, 458–461, 463, 541, 741–742, 745, 756
  closed-closed 287, 745–746
  closed-open 289, 746
  open-open tube 288, 349
Turnhill, William B. 32
Two degrees of freedom 342, 404–407
Tympanic membrane (ear drum) 73

U

Ungar, Eric 323, 412–415
Uniform Building Code (UBC) 427, 510, 700
Upper gallery (scala vestibuli) 76
Ureda, Mark 231
Uris, 338
Useful-to-Detrimental Energy Ratio (U_τ) 592, 600, 602
Useful-to-Late Energy Ratio (C_τ) 592, 600
Useful-to-late signal-to-noise ratio 620
Utriculus 76

V

Valeria 4
Van Gendt, A. L. 29
Van Houten, John 102, 529–531
Variable volume 729
Vehicular traffic noise data
Velocity 4, 20, 37–38, 51–54, 56, 58–59, 81, 115–116, 127, 129, 148–149, 157, 170–171, 177–183, 191, 194–195, 197, 200, 207–208, 219, 225, 237, 239–240, 252–253, 262–265, 267, 272–275, 279, 291, 304, 306, 319, 323, 341, 345, 366, 381–384, 401, 410, 427–428, 430, 432, 434, 438, 441, 459, 464, 467–470, 472–473, 474–475, 482–483, 485, 490, 497, 508, 529, 531–532, 536, 748, 797
Vena contrata 203
Ver, Istan 54, 60–61, 175, 181, 262, 323, 366, 382, 434, 435–437, 439–443, 469, 498, 500, 504
Vermuelen, R. 736
Vibration
  acceleration 381–382, 384, 390, 415
  displacement 381–384, 387–388, 392, 399, 404, 407, 410, 412, 415
  isolation 202, 381, 387, 389–390, 392, 394–399, 401–403, 405–406, 418, 420, 440, 479, 522, 533–534, 539–540, 571, 637, 763
    efficiency 390–391, 399
  isolator 357, 389–390, 392–393, 395, 401–402, 423, 525, 536, 637, 763
    pads 392–393, 395, 520, 533, 535, 537, 539, 574
    neoprene 392–394, 535
    spring 420, 574
    hanger 394, 401
    air spring 201–203, 228, 272, 334, 360, 420
  jerk 381
  velocity 381–384, 401, 410
Vibron Ltd. 395–399, 403
Video projection 700, 765–766
Vinokur, Roman Y. 339–342, 344
Vitaphone 742
Vitruvius Pollio 5
Vivaldi, Antonio 19
Voice
  directivity 1, 544
  over 773–774
  spectrum (VS) 545–546, 548, 552
  spectrum rating (VSR) 548
Volume per seat 580, 658, 660, 662, 678, 701
von Bekesy, G. 77
von Gierke, H. 101–102, 106
Vorlander, M. 798

W

Wagner, Richard 25, 657
Wakuri, T. 782
Walking 407–409, 413–414, 417, 427, 440, 446, 523, 528, 552, 558, 660, 742, 773
Wall 4, 15, 30, 107, 146, 161–163, 165, 204, 210, 218–219, 231, 233–234, 237, 240, 249, 256, 261, 263, 265–268, 271, 278, 280–281, 286, 301, 306–310, 315, 330, 333, 337–339, 344, 351–353, 355, 357–358, 360–364, 368–370, 375–378, 386, 399, 403, 422, 449, 478, 482, 486, 491, 500–501, 511, 513, 515–522, 526–529, 531–533, 540–541, 544, 560, 563–569, 572, 574, 585, 587, 608, 615, 656, 659, 663–666, 682, 684–685, 688, 693, 699, 703, 706–707, 710, 712, 719–720, 722, 725–727, 730, 734, 746, 749, 752–753, 756, 759, 761, 764–767, 769–770, 773, 775, 777–780, 783–784
Wall penetration 376, 402–403, 521, 531, 770
Water hammer 529, 534–535, 537, 540
Waterhouse, Richard 237–238, 240, 307
Watters, B. G. 245, 282, 663, 759, 761

Wave
equation one-dimensional 205–206
equation three-dimensional 207, 294
number 51, 206, 209, 265, 289–290, 294, 296, 333, 493, 797
plane 57–59, 148, 161, 205–206, 208, 236–238, 250, 265, 285–287, 291–293, 304, 319–320, 333, 347, 365, 367, 427, 492–493, 495
propagation 38, 50, 177–178, 179, 285–287, 290
standing 250–251, 286, 295, 478, 746
table 781
Wave motion 55, 262, 333
longitudinal 50, 54–55, 436, 441–442
shear 55, 326, 328, 418
torsion 55
bending 55–56, 163, 240, 323, 326, 332, 341, 345, 430, 433–434, 440
Rayleigh 55–56
Wavelength 37–38, 51–52, 73, 118, 127, 162, 172, 182, 188, 199, 203, 209, 211–212, 214, 217, 220, 222, 230, 233–234, 239–240, 247, 249, 265, 273, 279–283, 285–286, 289–291, 293–294, 322, 326, 332, 334, 345, 349, 355, 467, 483, 488, 492–493, 583, 615, 635–636, 681, 728, 748–749, 782, 787, 802–803
Wave
acoustics 199, 205
plane 57–59, 148, 161, 205–206, 208, 236–238, 250, 265, 285–287, 291–293, 304, 319–320, 333, 347, 365, 367, 427, 492–493, 495
Wegel, R. L. 92
Weighting Curves, A, B, C, D, E 85
Weighting factor (WF) 142, 545–546, 601
Wells, R. J. 98, 495
Wenger Corp. 764
Wente, E. C. 741
West Angeles Cathedral 718
Whispering gallery 249
White, Robert W. 412–415
White noise 40
Whittle, L. S. 141
Wilfrid Laurier University 722–723
Willaert, Adrian 11
Wilson, C. P. 347–348
Wilson, Ihrig & Associates 532
Wind 126–127, 157, 163, 165, 168, 177–178, 180–181, 183, 663
Window 257, 338–341, 353, 355, 366, 369, 371–372, 378–379, 557, 567, 618, 707, 736, 772, 774–775, 778–779
dual-paned 341
Wood 4, 22, 29–30, 115, 166, 257–259, 261, 269, 279, 274, 330–331, 339, 345, 357–358, 361, 363, 369, 373–378, 383, 386, 402, 407, 418, 420, 427, 429, 442, 445–449, 514–519, 522–529, 531, 535, 537, 541, 660, 664–665, 679–680, 684–685, 691–692, 695, 701, 707, 710, 724–726, 756, 761, 772–774, 782, 797
Wood floor 339, 386, 427, 445–446, 449, 522, 526
Wood stud 357, 361, 516
Wyle Laboratories 130–131, 196–198

Y

Yamamoto, K. 782
Young, Warren C. 443

Z

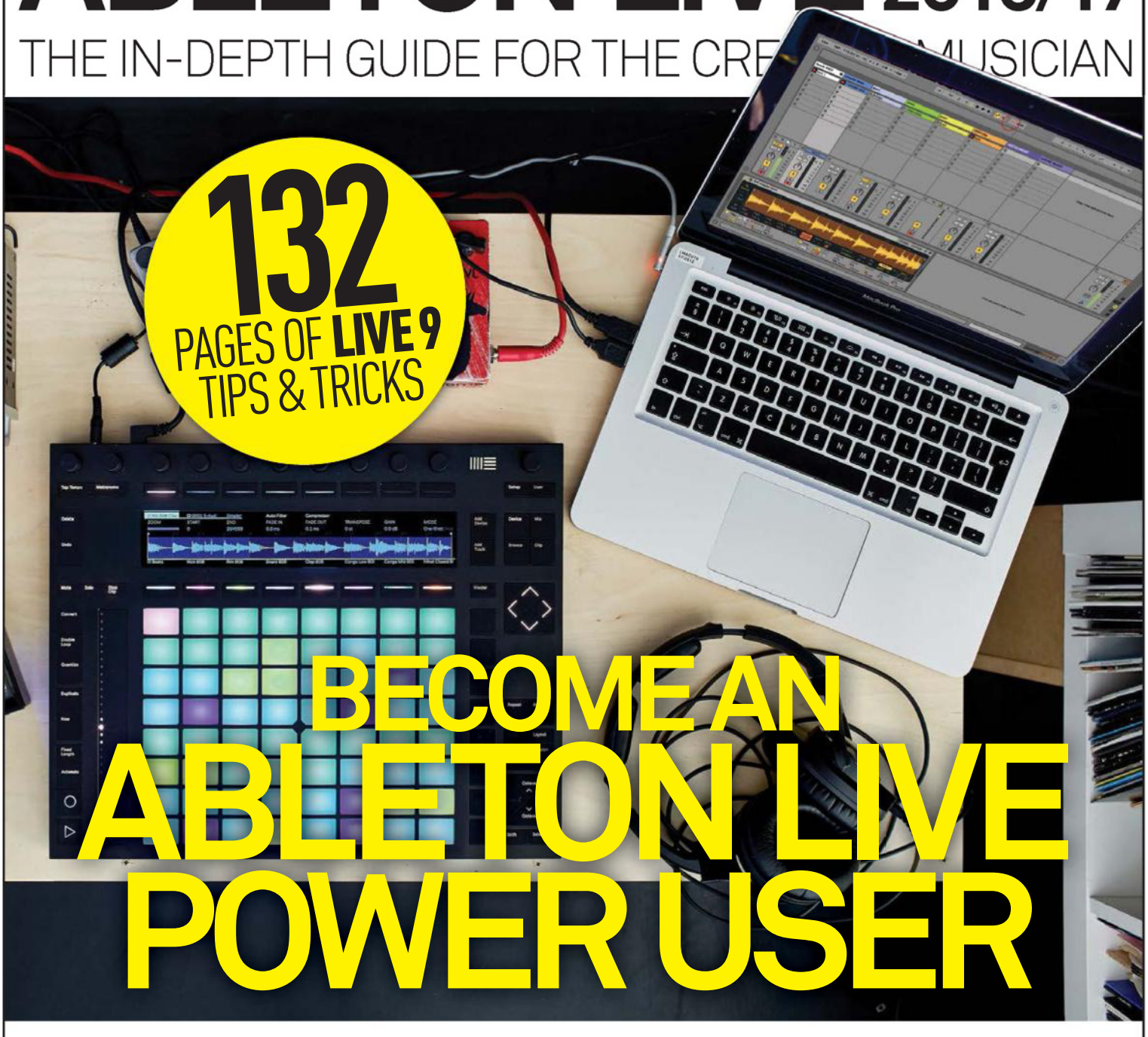
Zwicker, E. 84

MusicTech FOCUS

DVD ROM FREE 1.4GB OF SAMPLES 2 HOURS OF VIDEO

ABLETON LIVE 2016/17

THE IN-DEPTH GUIDE FOR THE CRE MUSICIAN



- **30 pages of workshops** – POWER UP in **Live now!**
- **Produce a track** from start to finish – **40 essential tips**
- The **Lazy Guide to Live!** Short cuts **to instant creativity**
- **Packed DVD! 1.4GB** of copyright-free samples **2hrs** of video
- PLUS: Ultimate **remixing** • Great **gadgets** • **40 gear tests**

Compiled by the Ableton Live experts from MusicTech

ANTHEM

9 781909 590090

Ableton Live 2016/17
£8.99

fabfilter

"Possibly the best EQ plugin on the market"

Computer Music Magazine (CM212)

Pro·Q²

An equalizer is probably the tool you use most while mixing and mastering, so you need the best of the best! With FabFilter Pro-Q 2, you get the highest possible sound quality and a gorgeous, innovative interface with unrivalled ease of use.

www.fabfilter.com



Welcome





... **to Ableton Live 2016/17!** This is a special issue of *MusicTech* where we assemble features specifically tailored to Ableton's music production DAW. This time we've really pulled out the stops with some of the finest studio articles ever: from **6 Ways To Make Your Studio Look Cool** to the **Ultimate Guide To Remixing**. In between it's everything Ableton with **five more in-depth Live tutorials** from Live guru Martin Delaney, plus a special '**Lazy Guide**' to creativity in Live, including some fantastic tips on creating variation in a loopy environment. With our **Produce A Track From Start To Finish** features (starting on p16) there's very much a Live focus and our special **Show Off Your Studio** articles (p70 and p88) all feature Live at their software hearts – and in one it's running in sync with a mountain of hardware. It all goes to show the diversity of music-making environments that Live is now central to. Perhaps the software used to be seen as a laptop-only DAW but with Push 2 and Link it is spreading into more hardware and mobile studios. Wherever and however you use Live, we hope you'll find tons here to help you. Happy music making!

However you use Live there is plenty here to help you get more from it...

Andy Jones Senior Editor, *MusicTech & MusicTech.net*
Email: andy.jones@anthem-publishing.com
(And contact me to Show Off Your Live Studios!)

MusicTech FOCUS

MUSICTECH FOCUS MAGAZINE
www.musictech.net
Anthem Publishing Ltd
Suite 6, Piccadilly House
London Road, Bath BA1 6PL
Tel +44 (0) 1225 489984
Fax +44 (0) 1225 489980
editorial@anthem-publishing.com

Senior Editor Andy Jones
andy.jones@anthem-publishing.com

Art Editor Debra Barber
debra.barber@anthem-publishing.com

Digital Editor Andy Price
andy.price@anthem-publishing.com

Multimedia Editor Alex Holmes
alex.holmes@anthem-publishing.com

Business Dev. Manager Di Marsh
di.marsh@anthem-publishing.com

Contributors
Mark Cousins, Keith Gemmell, Alex Holmes, Huw Price

Art Director Jenny Cook
jenny.cook@anthem-publishing.com

Publisher Simon Lewis
simon.lewis@anthem-publishing.com

Managing Director Jon Bickley
jon.bickley@anthem-publishing.com

Licensing enquiries Jon Bickley
+44 (0) 1225 489984
www.anthem-publishing.com

Printed by Polestar UK Print Ltd
Tel +44 (0) 1582 678900

Distributed by Marketforce (UK) Ltd
The Blue Fin Building, 110 Southwark St, London SE1 0SU Tel +44 (0) 20 3148 3300
Subscriptions to MusicTech Magazine
Tel +44 (0) 870 444 8468
Price (12 issues) £59.99
UK basic annual rate

All content copyright Anthem Publishing Ltd 2015 and 2016, all rights reserved. While we make every effort to ensure that the factual content of *MusicTech Focus* is correct we cannot take any responsibility nor be held accountable for any factual errors printed.

Please make every effort to check quoted prices and product specifications with manufacturers prior to purchase. No part of this publication may be reproduced, stored in a retrieval system or resold without the prior consent of Anthem Publishing Ltd.

MusicTech Focus recognises all copyrights contained within this issue. Where possible we acknowledge the copyright holder.

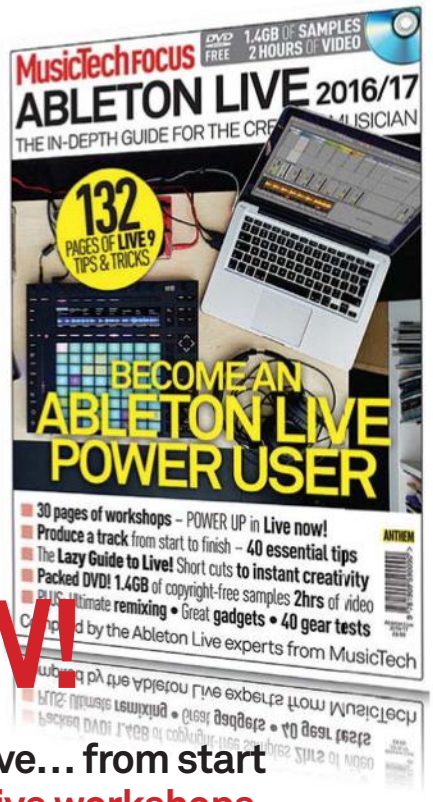


ANTHEM PUBLISHING PPA

TEXTING MUSIC TECH TERMS AND CONDITIONS
By entering *MusicTech* competitions you are agreeing to receive details of future promotions from Anthem Publishing Limited and related third parties. If you do not want to receive this information please text 'NO INFO' at the end of your message. Texts will be charged at the specified price plus one message at your standard network tariff rate.

MusicTech FOCUS

Issue 44 ▨ Ableton Live 2016/17



EVERYTHING ABLETON LIVE! MAKE MUSIC NOW!

Everything you need to produce a track in Ableton Live… from start to finish! We have six brand new series of in-depth Live workshops plus features on remixing, creating an incredible studio space, the best 12 soft synths, your Live studios on show and the best hardware and software for your Live studio on test…



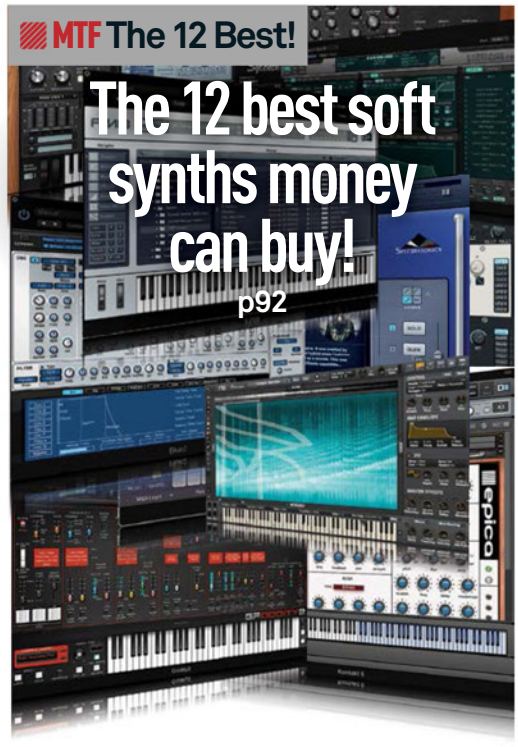














MTF44 Live 2016/17 Full listings

WORKSHOPS

006 | The lazy person's guide to Live Get creative... fast!

034 | Adding hardware to your Live rig with increased ins and outs

038 | Follow Actions and how you can make them work for you

042 | Creative clip management using Push 2

058 | What's new in Live 9.7 The latest features explained

062 | Create custom Drum Racks Flex those finger drumming fingers

TUTORIALS

016 to 126 | *Produce A Track From Start To Finish* Starting a track; arranging, mixing, finishing and finally: mastering

022 | *The Ultimate Guide To Remixing* Everything you need

032 | *6 Ways To Make Your Studio Look Cool!* The best ways to make your environment more creative

082 | *20 tips to improve your computer's performance* Beef up your mac or PC... now!

OTHER FEATURES

049 | *Interview:* Oliver Nelson

054 | *6 Of The Best:* mega expensive studio items

070 & 086 | *Show Off Your Ableton Live Studio*

072 | *Ultimate music gadget guide*

092 | *12 Of The Best:* soft synths

096 | SUBSCRIBE!

MTF REVIEWS

098 | Gothic Instruments Dronar

100 | Teenage Engineering PO-20

104 | NI Komplete Kontrol S88

107 | Zivix Jamstik+ controller

108 | Fabfilter Pro C2 EQ

109 | Sonic Academy Kick 2 inst.

110 | Zero-G Epica Bass inst.

112 | Sonimus EQ range effects

114 | Splice subscription samples

116 | NI Una Corda piano plug-in

117 | Blocs Wave app

118 | Spitfire Hans Zimmer 2&3

120-125 | *Mini Reviews*

130 | *On your MTF DVD*

Ableton Live | A lazy person's guide!

Level | Beginner

The lazy person's guide to easy clip creativity

Live is obviously excellent at looping clips – that's what it's built on, after all – but it's all too easy to get bogged down in rigid looping hell. Fortunately it's almost as easy not to! ***MusicTech Focus*** has an easy guide to creative clip use and how to get creative… fast!

We did argue over whether to call this 'The Lazy Person's Guide To Live' as we thought it could be a little too demeaning. But, let's face it, using a DAW is basically using technology as a short cut, so all of us who use DAWs are either lazy or – and we certainly prefer this option – using technology to realize our creative dreams more quickly. Yes, that's good – we'll stick to that!

So the crux of this tutorial is that the main problem with technology and DAWs is that it's very easy to fall in to some bad habits. You might, for example, just use the features available in a non-creative way and end up with something quite tedious. Live suffers especially from this largely down to its central selling point of being able to play loops in time, so that everything sounds pretty good, very quickly. This is both its main draw but also a feature which might make you think: 'Sounds ok, let's just run with that…'

We want more than 'OK'! The beauty of Live is the not-so-hidden depths that you can explore to take these looped ideas onwards and upwards. A few tweaks here and there and you suddenly introduce a dynamic to your tune that gives it life. Add a bit of variation to your clips, either by changing the notes within them, the sounds they trigger, or their start and end points and you suddenly have a lot more variation and creativity going on.

And you know what the best bit of all of this is? It's pretty easy to accomplish! It's why we've labelled this with a 'Beginner' tag although if we were betting people, we'd guess that there are people who consider themselves intermediate and expert users who choose not to use some of these short cuts – as they might not be considered clever – or they don't know about them. Either way we feel that all users should give this workshop a spin and we're sure that everyone will discover, or rediscover, at least one choice short cut…

1. The template factor

The first feature we'll cover is less about creative looping and arranging but more about making sure you have a good base to start with and covers how to set up a Default Set or Template. We can't recommend having one enough. When we started experimenting with loading in a standard Set when starting a new project, we were a little sceptical. Ten tracks later – ten times more than we'd have previously produced in the same time period – we were hooked.

The idea is that you start with a favourite set of instruments, effects and even audio, each time you start to compose. You'll end up moving between clips and scenes faster and putting tunes together faster. You might think that having the same set of sounds to start



FOCUS ON SESSION VIEW

For this tutorial we're focussing completely on working with clips within Live's Session View. You could say that is the ultimate in laziness – hey, you don't even have to switch views – but the truth is that there is so much to expand upon and create within this part of the software that we don't need to go into the Arrangement window. There are so many extras packed into Live and within these clips that we'll keep the focus here this time and save all of the Arrangement View fun for another time…

MTF Step-by-Step 1. The template factor



01 Whenever you start a new Set in Live you could, of course, hit Cmnd>N and start from scratch. It's a nice empty project that you have to start from scratch with, yet looks like a lot of effort to fill...



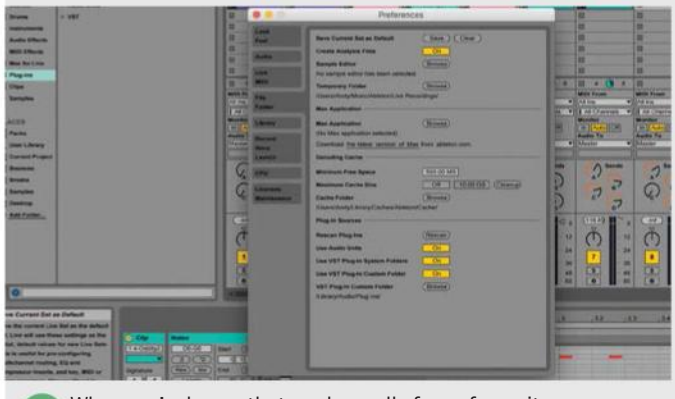
02 Instead load up a project that you have been working on and are happy with the range of sounds and effects already loaded in. With tweaking you could make this the new Default new file that loads.



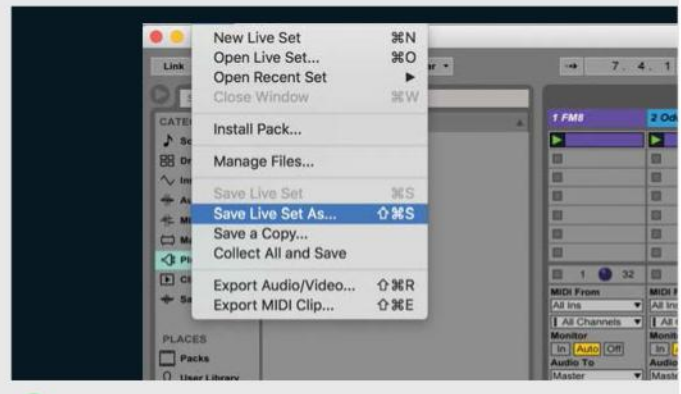
03 If you haven't got one that you are happy with simply load in one and build it up from scratch. Put your favourite instruments on different channels, line up effects, even load in some clip melodies if you like...



04 You could even load in a selection of beats, maybe some 4/4 beats if you are a house musician, so you know you've always got something reliable to underpin your new project.



05 When you're happy that you have all of your favourite instruments, effects and audio clips loaded in, click Live's Preferences/File Folder tab and the 'Save Current Set As Default'. It will now load in whenever you start a new project.



06 Make sure to Save Your Live Set As something new at this point otherwise your Default File may be altered as you explore new tangents and ideas with this notepad way of working.

with is limiting, but actually setting constraints sometimes *enhances* creativity. We can certainly tell you that our ten tracks don't sound anywhere near as similar to each other as you might think. Another way to look at it is that your Default Set is your 'scratch pad' for melodies. You can return to the ingredients later once you have some basic ideas; change sounds, change percussion loops, change everything – just use the Default Set to capture the vibe!

2. The virtual keyboard

Unless you own Push or Push 2 – in which case lucky you – you'll probably be using Live with an external keyboard. Well, as good as they are, at a very early stage of your notepad composition process we reckon that Live's Computer MIDI Keyboard option takes some beating – it does the job of auditioning and recording and again lets you jump from clip to clip quicker. It also takes Live use back to its early days of laptop-only composition. As good as Live is at breaking out of a software-only environment with Push (and our tutorials in this issue) – which seems to be something of a theme these days – what it was really good at offering in those early days was a 'laptop only' way →

MTF Step-by-Step 2. The virtual keyboard



01 Live's Computer MIDI Keyboard makes both auditioning and recording a lot more mobile and faster. Engage it with the icon shown top right or the drop down menu of Shft>Cmnd>K.



02 It turns the Q and A rows of your computer's keyboard into a virtual piano keyboard with the A row taking the white notes; Q the black; Z and X octave down and up respectively.



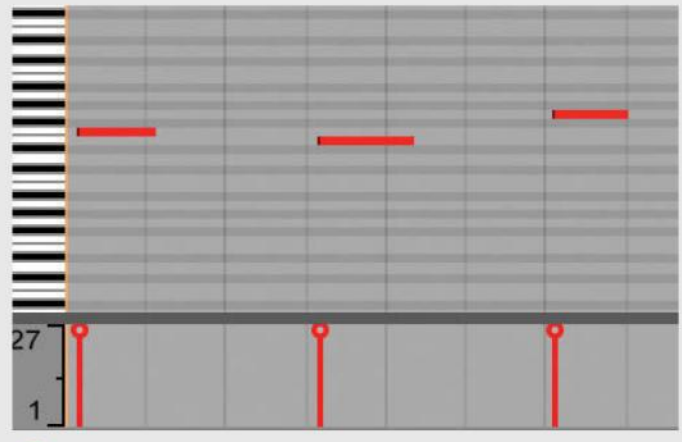
03 Load up any instrument and you'll miraculously be playing its keys, that is if you have the right track armed, of course. It makes auditioning sounds on clips a dream.



04 And recording is also something that it can take care of, but you may have to select a couple of quick edits. On the MIDI IN drop down menu select Computer Keyboard.



05 Make sure you Arm Record (top) before playing. Now you can record ideas over your existing loops allowing for solos and non looping parts to be played making your overall song structure less loop heavy.



06 We found that playing in notes like this and just jamming ideas gave us more of a freedom, to the point where we left bum notes in or at least those slightly played out of time for a more natural feel.

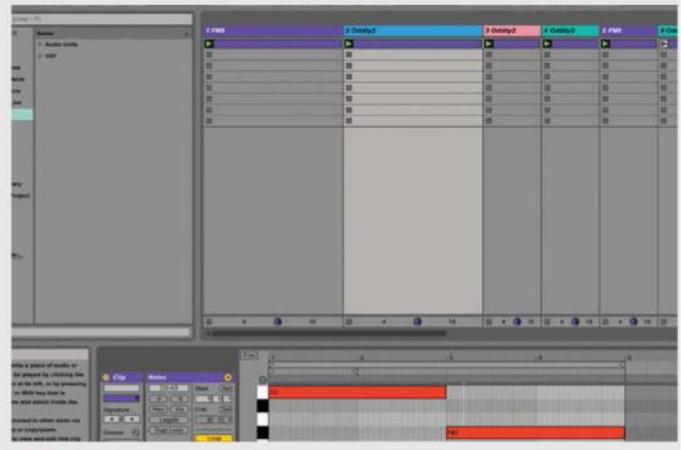
→ of putting ideas together which gave it a huge amount of appeal. You engage the computer 'keyboard' either by clicking the keyboard icon top right or hitting Shft>Cmnd>K. Play notes using your computer's A and Q rows and use Z and X to shift octaves up and down.

In order to record these, make sure you have Record Enable clicked at the top of the track. You'll also need to select the Computer Keyboard as a MIDI IN. When you record your notes you'll find that it's a fabulous and mobile way of breaking quickly out of rigid loops – the clips grow as you record – and you can easily jam ideas over existing clips. And talking of clips...

MTF Step-by-Step 3. Creative clips



01 Live is all about simple clip editing for variation. It is so easy to do this on a very basic level. Select a clip, hold down Alt>drag and copy it either to another track or another clip within the same track.



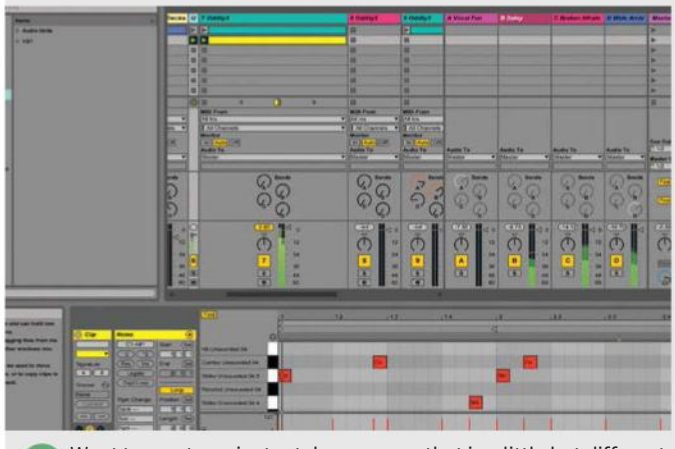
02 Copying it to another track is particularly good if you have a template arrangement set up (from two pages back) as you can then trigger more of your favourite sounds with the same MIDI clip.



03 Here we're fattening up our original bass sound from one synth (FM8) with bass sounds from another (Carbon Electra) just by copying – an obvious ploy but Live makes it so easy to do, why not?!



04 Here we've done the same thing three times. We've copied the same pad clip (FM8) to three string sounds from three instances of the same synth (Oddity) to create an instant string section!



05 Want to create an instant drum groove that is a little but different from the norm? Grab yourself an original bass line clip and copy it, but instead of copying it to a melody track... copy it to a beats track.



06 Load in one of Live's many drum kits and drop a melody clip onto it. Some simple note transposition might be needed but it will result in beats that work. Try it the other way (from beats to bass) too!

3. Creative clips

On a very basic way you can get creative with clips very quickly and very easily. Firstly there's the obvious: copying! Copying one clip to another track and listening to it trigger different sounds is one way to fatten up a bass sound or add extra padding in your upper mids. You are already in the same key as the original notes are the same and if you have a templated set up as we suggested in the first tutorial, you will quickly be triggering other sounds from your favourite synths using your initial melody clip. Then, of course, it's very easy to move a few notes around within a clip and either keep the clip on the same track to provide some melodic variation over the original track – do this as we will return to it later – or move it onto another track.

MTF Step-by-Step 3. Creative clips (cont'd)



07 It's also very easy to change a few notes around within a clip and use this as a variation. First copy the original clip to the same track (Alt drag) as shown. Double click it to edit the notes...



08 Selecting a couple of different notes to change will be enough. A couple of notes on each clip within a track will make a difference – again it's very basic but worth doing for variation.



09 Make a group of five such varied clips on a track which we'll return to for several steps in this tutorial (and the one opposite). Make sure they each have slightly different notes.



10 We'll now look at clip lengths. Live makes it very easy to keep everything rigid and perhaps set to 4- or 8-bar lengths. Everything locks in nicely but can sound dull if you're not doing dance.



11 For eachof our five clips we can set the start and end points differently to shake things up a bit which breaks out of the rigid dance music box. Here we've set the fourth clip to start and finish on the second beat.



12 Try this on larger tracks too. Here we have two identical string tracks placed on massively different clip lengths and with different start and end points. They both cycle very differently resulting in a more ambient feel even though the notes are lazily the same!

→ Another lazy trick is to move a melody clip onto a rhythm track to quickly get some beats going. These probably won't be very typical beats – as they will come from a melody MIDI clip – but will be something unusual to start with and perhaps not something you'd have come up with just programming straight beats. This also works the other way around too often with bass sounds. Try placing a programmed beat on a bass track. It might just work!

So far so simple. Copying clips and throwing different ideas at clips is what Live is all about. But now we'll try a couple of other simple clip ideas out to get things moving by using odd numbered clip lengths and different start and

MTF Step-by-Step 4. Follow Actions for easy beat creativity



01 We have a more in-depth tutorial on Follow Actions later in this issue. For now we're going to use them just on beats, specifically a loop to create easy variation within it. Load in any loop!



02 We'll now bring together everything we've done in this tutorial up until now to bring some variation in on the beats. So first copy the clip five times onto the same track as shown using Alt>drag.



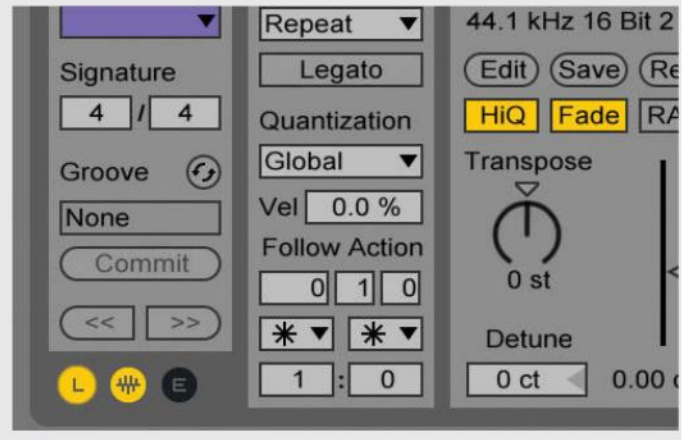
03 Now we're going to adjust the start and end points of each of the new audio clips we've just created to home in on specific areas of the drum loop. Here we'll start with the first kick drum.



04 On the second repeated clip (third clip overall) we'll home in on the snare sound plus a little bit of the percussion action. As you'll see this is a start point of 1.2 for a duration of 0.2.



05 Keep going until you have dissected the audio clip into four different sections to make up the five clips overall. You can do more clips, of course, but we'll stick to this number.



06 You can also do shorter sections but we've kept ours fairly long, each over half a bar in length (we'd even recommend going to one bar). Now go to the Follow Actions by hitting the L in the Clip area.

end points. What Live is really good at doing is playing four- or eight-bar clips together in a standard dance way, but if you break out of the mould even a little and have clips cycling around on odd bar lengths, they start to drift away from each other creating a more random song. Obviously this might not be too good for the tightness of dance music but if you want something a bit more ambient or random, try experimenting with clips that stop and start at different timings or have very different lengths. They might even have the same note content but they are playing at different times and with different sounds...

4. Follow Action for easy beat creativity

Next for clip variation it's time for Follow Actions. These →

MTF Step-by-Step 4. Follow Actions for easy beat creativity (cont'd)



07 The Follow Action section will now determine how the clips you have just set play and interact with each other. Click on each and you can determine how long each plays with bars, beats and ticks.



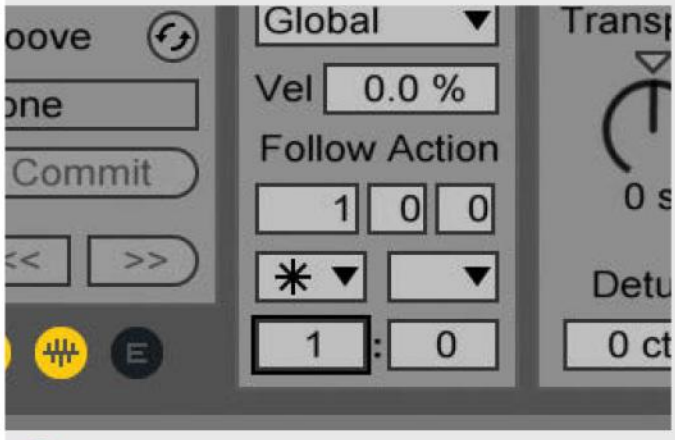
08 Click on the drop down menu and you will see a list of Follow Actions that determine what happens after the selected clip plays: from nothing to playing another selected clip at random.



09 Now you simply need to go through the five clips you have on the track and determine what you want to happen after they play by selecting one of these options from the drop down menu. Here we have the third clip selected and have selected to simply play the next clip.



10 And on the final clip we have selected 'Any' as the Follow Action so any of the selected audio clips will play at random afterwards. This is obviously great for adding a completely off-the-wall variation to proceedings, but doesn't have to be completely random!



11 A really good option within the Follow Actions section is the probability number. Set it to 1, as shown, and the probability of the next clip doing the selected Follow Action will be 100%. Set it higher and that probability drops proportionally.



12 Another cool thing is to use a second drop down menu to determine what happens the rest of the time so you could, for example, have a random other clip triggering for half the time and the first clip in the track triggering for the other half of the time.

→ might seem fairly complex at first but, believe us, they are a lazy person's dream feature! They essentially allow you to trigger clips in a more random way to bring lots of variation to a tune. In our example we're going to have variations of a clip all within the same track and we'll use Follow Actions to determine how they are played.

We go into more general detail on Follow Actions and clips in a tutorial later in this issue so for this tutorial we'll use them on an audio clip to create a variation in beats. You'll see that you never need to think of Live as a straight looping tool ever again as your loops are suddenly creatively transformed into very varied beats!

MTF Step-by-Step 4. Follow Actions for easy beat creativity (cont'd)



13 Now to hear all of your Follow Actions in, well, action! Select all of your clips as shown and press one of them to start playing. We have all of ours set to play another clip at random after each...



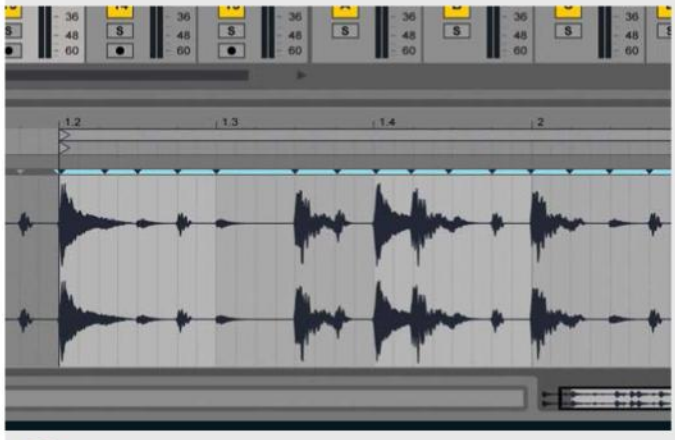
14 And it's a bit wild to start with, almost as if someone has taken control of your Live session and is triggering clips for you. The result is a bit too DJ Shadow-like on ours to start with...



15 Experimenting with the original clip length that you create (from step 3) is key here. You can either go back and adjust them from scratch or adjust the clip length in the Follow Action area for each.



16 Essentially the more stuttery beats you want, the shorter you make the clips. You can go completely wild here by slicing into beats for some old skool mid 90s Aphex beats.



17 We did find on our particular loop, that increasing the clip length to a bar for all clips gave the best results and took it far away from the original loop, but this will vary depending on your loop.

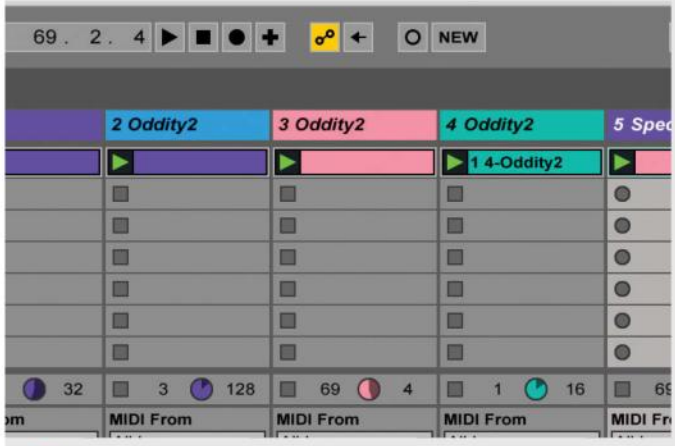


18 As if that's not enough, you can of course go into editing each of your new clips to make the variation on the original loop even greater. Here we're transposing the audio from one clip.

First we need to follow some of the clip edits we have already covered i.e. repeating and copying clips into the same track and changing the start and end points so that we home in on certain parts of the percussion loop. We are essentially cutting out half-bar chunks of the loop to play individually and using Follow Actions will determine just how. So we get four additional but different clips all from the same initial audio clip and all on the same track.

Now select Follow Actions in the clip area. You then go through all five of your clips and select an Action for each i.e. what happens after it plays. You can play another at random, or the next or previous clip – you have a choice of nine. This creates a completely new playback version of your original loop but importantly doesn't sound looped! →

MTF Step-by-Step 5. Clip automation



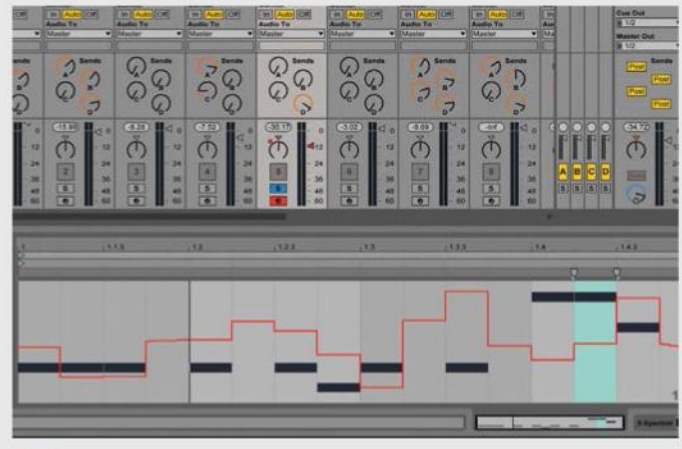
01 One final element in clip management that enables more creativity is clip automation. This means that you can record several parameters over a recorded clip to give it life and movement.



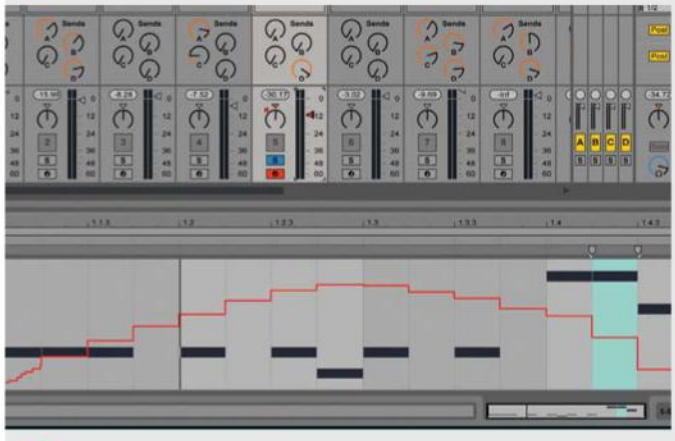
02 The yellow icon (with two arrows in the transport bar) shows that automation is enabled (if not click on it) but you will also need to arm each track...



03 All sorts of automation parameters can be recorded here including panning and volume. As you pan from left to right, for example, your movements will be recorded into the clip.



04 The automation will be shown by way of a red line along the clip as you record. Here the panning goes from left to right and back again as the clip cycles around.



05 If you want to switch to another parameter, like volume, simply move a fader and its automation will be recorded over the top. If you want to go back to the panning, simply click the pan control. Here's our volume coming in and going out over a clip.



06 If ever the automated data is greyed out – by clicking off a clip, for example – you might need to click on the orange back arrow which should be illuminated, to re-highlight the clip automation back to its red editable status.

→ 5. Clip automation

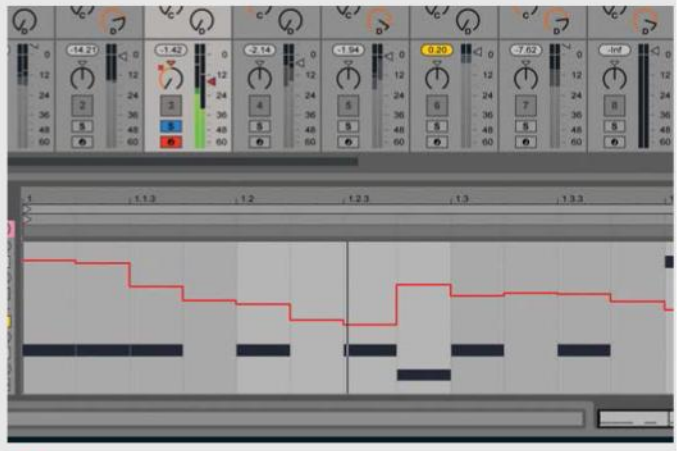
So to our final way of easily getting variation from your clips, and it's time for some automation. Again this is a subject that we could go into some depth with, but as this is 'The lazy person's guide' let's not go too deep. All you need to do to get things moving is to hit a couple of keys and record some movement and everything you record will be kept within your clip in Session View. Edit it later to make it perfect!

Simply arm the track and enable the automation. We start by showing you how simple panning and volume changes can be recorded per clip. Then we detail how you

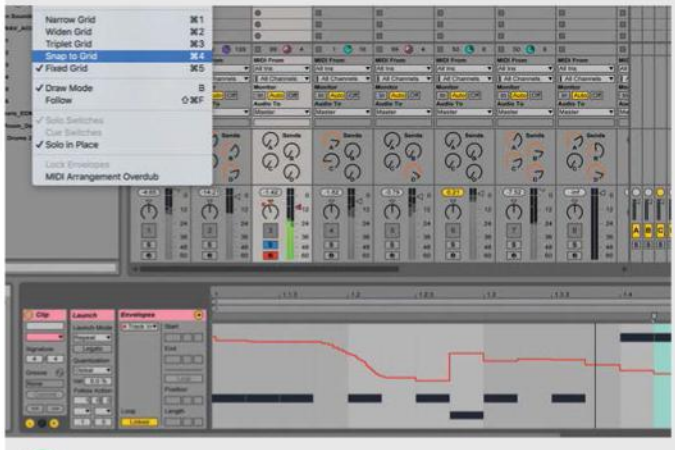
MTF Step-by-Step 5. Clip automation (cont'd)



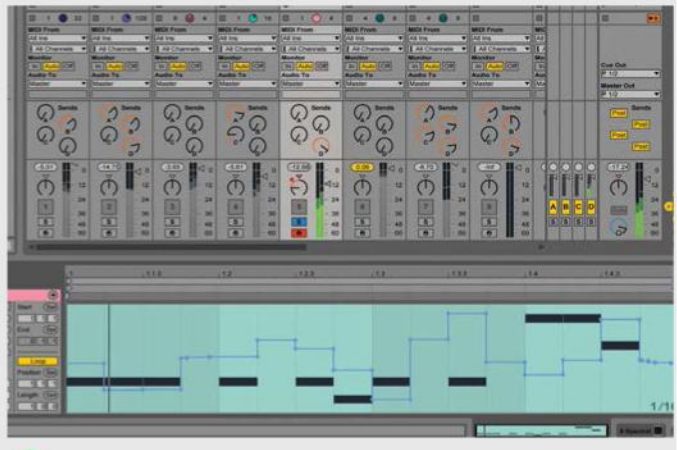
07 You can edit any automation by selecting a point with the cursor and dragging it as shown (in blue) as you would in the Arrangement view.



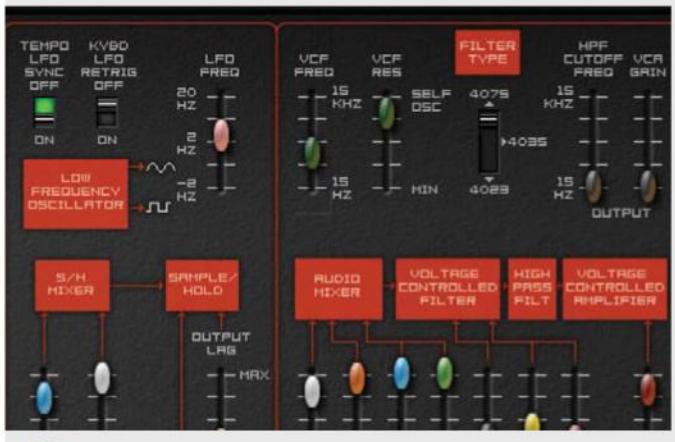
08 Better still, select the Pencil tool (top right) and draw the automation in yourself. Here we've adjust the panning position ourselves and completely redrawn it. Looks a bit chunky though...



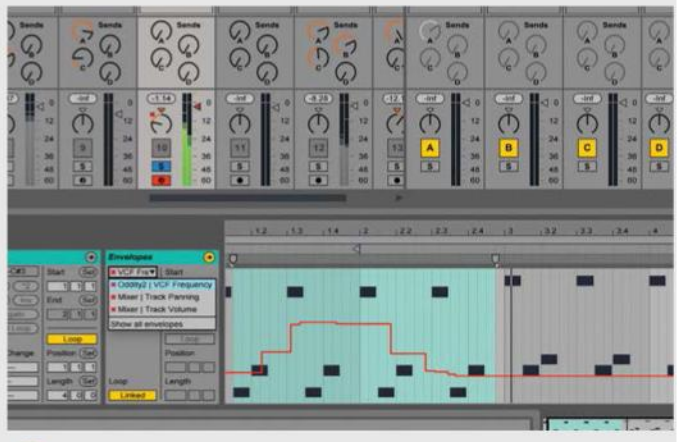
09 That's because the automation draws to the snap value so use the dropdown menu to turn the snap off and you can draw your automation in as a finer resolution (shown on the left here).



10 You might want to start from scratch – we certainly always get over excited by automation and end up doing too much – so simply select all and delete and start again. Easy.



11 So that's panning and volume, how's about we dig deeper and record some synth parameters changing into a clip? No problem! Here's the frequency opening being recorded on a synth.



12 And now as it appears as one of the envelopes of automation being recorded. Simply select from the drop down menu and you can edit it. So clip automation really can take your clips to new levels...

can edit the results or draw in completely new versions of the automation. Finally you can even record the changes you make to MIDI instruments on the track in question – we show you how to record a synth frequency fader being opened and then how to select that recorded automation (from a drop down menu) to then edit it. Automation is a key feature in any DAW and Live's clip automation takes it well beyond what is possible in some other DAWS and makes it a completely creative and immersive experience.

So we hope that this tutorial has enabled you to get more creative with your clips in Live and, importantly, to do that without spending too much time or effort getting there. Using technology to make music should make it easier after all! Happy creative clipping! **MTF**

FIND
MORE
FREE
MAGAZINES

HTTP://SOEK.IN



MTF Feature Produce a track from start to finish

PRODUCE A TRACK FROM START TO FINISH

It's time for the ultimate advice on music production: how to produce a track, from start to finish! In features spread through the issue – see the full contents below – we have top tips from the pros and advice to get you inspired, composing, mixing, arranging and mastering your music

The art of music production can, like most processes, be broken down into several stages – some creative, some perhaps more scientific than others. For this special feature on producing a complete track, we've explored the best advice ever assembled on the art of music production and broken it down into five main areas: starting a track, arranging it, mixing it, finishing it and mastering it.

We'll start with the best ways and practices to help you to get moving, to get creative and to make the most of inspirational thoughts when they strike; to capture the moment… fast! We'll tell you how to avoid using other people's ideas, and also (confusingly) when to use other people's ideas!

Then there's the all-important stage of turning those ideas or simple loops into complete works. This is often the stage you get caught up in. Like us, your hard drive might be full of great ideas which will become incredible killer tunes… one day. But for now, they are one, two, five or six sounds and melodies that might work together, but nothing more. This part of our feature will help you turn them into complete tunes.

Then we move onto the art of mixing: checking levels, checking pan positions, adjusting EQ not only to enhance certain frequency ranges, but also to prevent elements clashing – the art of mix 'breathing', and adding space to enable your main song elements to shine.

In our penultimate feature, we bring you essential advice on making the ultimate move in composition: finishing! Here, we also team up with Novation to bring you insight into the ways certain top-name producers complete their songwriting process.

Finally, it's time to tackle the mastering process, with top advice on how add a professional sheen. We'll suggest ways to add that elusive sprinkling of sophistication and magic dust to your tunes, to make them sound as good as your reference material – the track that you initially admired, which inspired you to go full circle, back to the start of your music making in the first place. Without dwelling on the clichéd 'journey' analogy, it's true that music production is a process that can be massively enjoyable, but also have the odd stumbling block or creative frustration along the way. With our feature, we're hoping to make your process smoother and your songwriting experience more fulfilling.

As ever, send us links to your resulting tunes at **editorial@anthem-publishing.com**.

MTF Features

PRODUCE A TRACK FROM START TO FINISH
Top producers offer advice in all aspects of track building…

017 | *7 ways to start a track*
Let's start with a look at ways to capture ideas, sources of inspiration and how to prepare

019 | *8 ways to arrange a track*
Going beyond the loop and turning your ideas into songs is a stumbling block, but here's how to overcome it

046 | *10 ways to mix a track*
The mixing process is where your track comes to life. Here are our top tips for perfecting the art

068 | *7 ways to finish a track*
Here are some effective strategies to make sure you actually finish your tunes

126 | *8 ways to master a track*
Mastering dos and don'ts to add a pro sheen to your productions

7 WAYS TO START A TRACK

Part 1

It's the first step: how to get those ideas you have in your head onto your virtual notepad – and also how to get more ideas into your head in the first place!



1. Turn the internet off? Really? Yes! Get rid of those distractions

2. In Live, you can create an inspirational template, then save it as your default project

01 Clear your head

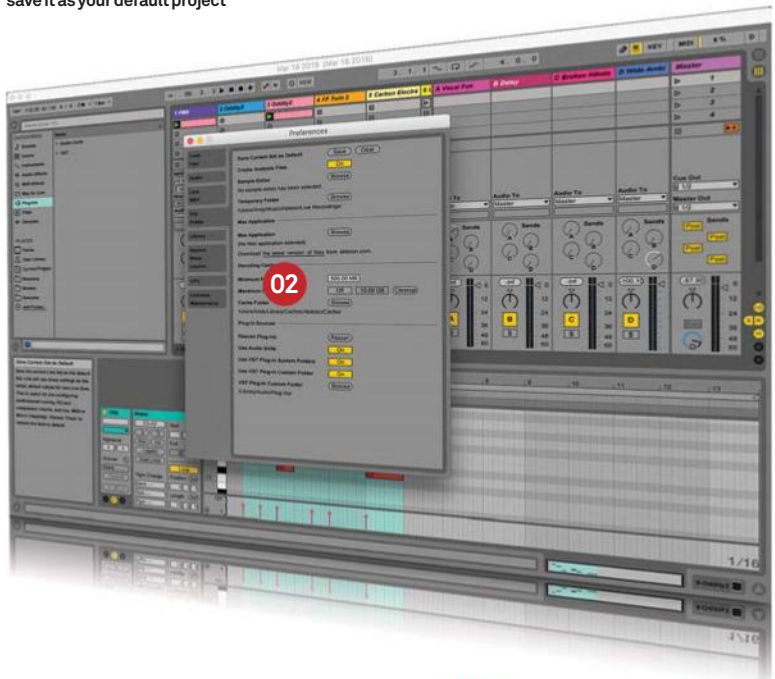
It's important to get into the right mindset when starting music. Getting into the mood helps with anything creative – if it feels like hard work, it'll end up being hard work. You have to have no distractions so switch off your internet, your doorbell, your phone – anything that could interrupt your flow.

02 Start from scratch

You've been there. You've opened up your DAW and it's loaded up an old song or some looped ideas you've been working on. Perhaps if you just spend a few minutes with it… But it's another distraction. If you're starting from scratch, start from scratch. Produce an inspirational template that your DAW loads each time you boot it up (instead of those old ideas) and make it full of your favourite synths and virtual instruments, all ready to play.

03 Favourites, favourites

And while we're at it, keep on choosing your favourite sounds and presets. Home in on a set of the best starting points for your music – bass, lead, pad, guitar, vocals, or whatever – in whatever style of music you make. Remember that most successful artists will only use a small set of tools to create their work – it's their signature set of sounds. They won't surround themselves with endless options, or they might not get anything done. Instead, they'll work within their own restrictions, so there's a compelling argument to do the same with your instruments. And as well as choosing your favourite synths for your template project, you can even home in on your favourite sounds within each one. And now go one step further and clear out that sample folder. Go on, spend an hour clearing out samples and audio that you'll never use and you'll have saved that time tenfold when it comes to choosing samples later. Too many options slow your workflow to a crawl… →





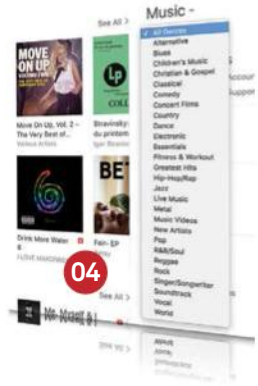
Get a style… and a plan

One massive hurdle for composition is sitting with a blank sheet of paper and expecting to fill it. You can zero in on sounds and instruments as we've suggested, but you should also have an idea of your own style from the off. You might think that this is something you have already mapped out in your music life – the kind of music that you will be making – but it's not quite as easy as that. You'll very probably like more than one kind of genre yourself, so it's all too simple to be easily led by that floozy of a breakbeat over there, or that dirty dubstep of a modulation over there. Musical temptation is everywhere and it is evil! The danger is you will end up with a hotchpotch of styles that can only lead to disaster (or, worse still, Bro Country!). So think very carefully about what you are setting out to do – have a tempo, a plan and a reference point in mind, because if you're up against a deadline, you will need a tight target and to go for it. If not, give yourself a fake deadline and go for it anyway!



05 Embrace inspiration

Always carry some kind of recorder with you (or download an app such as RecorderPro), as you never know when ideas will come to mind. Some people swear by keeping one by their beds, as ideas can come while waking or drifting off to sleep. Carry a recorder with you and speak (or sing!) those ideas into it wherever you are. You can also use one to capture unusual sounds and atmospheres for use later in your music – sometimes 'found sounds' can be the difference between your tunes and everyone else's. Take note, however! Very often, any tune that comes to you might well be someone else's. A quick way to identify a tune is to try apps like Shazam and SoundHound. These are primarily used to identify commercial music played around you in bars, but might just do the trick for new tunes. If nothing else,





3. Home in on your favourite synths and sounds

4. There are a lot of genres to be tempted by – try to settle on one!

5. Use Shazam or SoundHound to help you nail a specific style, or identify an intriguing tune you can't place

6. Borrow from the earworm that's currently in your head. Even *Agadoo* (if you feel you absolutely have to)

they are great for identifying an artist who suddenly inspires you. Give them a go anyway – there are free versions of both.

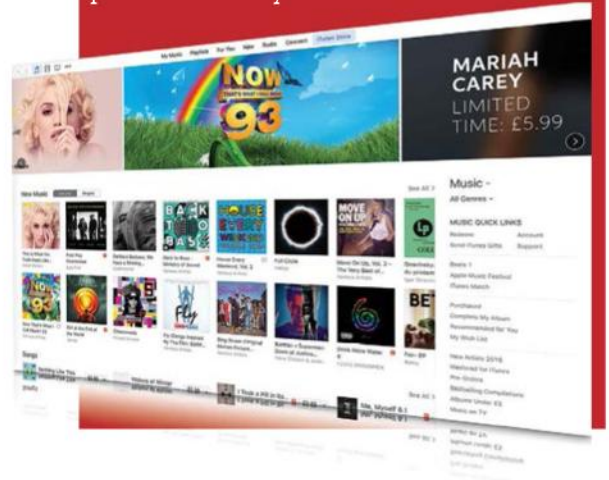
06 Borrow the earworm

Having said avoid other people's music, now we're spinning that on its head by saying… Don't be afraid to borrow ideas! Whether it be the odd sampled riff (with lots of post editing, naturally, and one that you would clear on release), or melodic progressions, you can and should use other people's music as inspiration. But let's clarify that and stress the word *inspiration*. The Blurred Lines/Robin Thicke case has, well, blurred things in music copyright and what you can and can't do. But don't be frightened to analyse that earworm to find out what makes it so 'wormy'. Is there anything you can use or learn from it to make your own creations as memorable?

Examine some of the key themes and trends from memorable songs (so long as it's not *Agadoo* in your head in the first place). Many people have borrowed ideas; you could take that argument further and say that *every* musician has borrowed from every other one at some point. Some, though, are more obvious than others: the odd riff here, the odd chord progression there. Again, use ideas carefully and edit, edit and edit some more… MTF

07 The best start

And the best tip for starting a piece of music is… To listen. Absorb. Learn. Be inspired. All of the above! Yes, you could say that the best advice we have for starting your own music is, well, *not* to. Listen to as much other music as possible: all genres, all tastes. Learn from it and really want to do it. This is probably the top tip above everything else for every step of the music-making process, including mixing and mastering (which we will also cover later). You need to have the desire to produce and this will come from being inspired by someone else and led in the direction in which you want to produce. This will only come from listening, listening and more listening, and not only to music that you like, but also to new genres and styles, stuff that you might not necessarily appreciate. Classical music will help you arrange strings, for example; other styles might inspire you not to go down certain routes. It's all about finding your way, and you'll need to be as open as possible to follow your own choices.



MTF Feature | Produce a track from start to finish

8 WAYS TO COMPOSE AND ARRANGE A TRACK

Part 2

You've got your ideas down into Live. They might be loops or simple scratchpad ideas. How do you flesh them out into full arrangements? Read on…



01

1. In Live, looping is what it's all about and it's done for you, but don't let the loop rule you.

01 Use looping creatively

You will probably find that whatever DAW you use, you will be positively encouraged to use loops at some point in the production process – usually at the start. And this, by and large, is a good thing. It gets to the heart of why you are using technology in the first place, as looping enables you to try musical ideas together very easily and very quickly. You can build all the elements of your song together using loops – bass, drums, leads, guitars – and then the main arrangement process starts. All good so far, but we'd also add the caveat that people are getting a little too reliant on loops – and looping does lead you down certain paths and towards certain genres. So, our current thinking is that looping is great at the start of your arrangement process, and we'd even encourage people who don't do it to try it.

Like we say, every DAW is a click away from a loop. Ableton dominates its part of the DAW world because of the loop, so use it, but be clever. Have them running at different lengths from each other (again, Live is great for this), have them jumping around, be less rigid with quantisation within them. Think free, be free. Don't let the loop rule you, you rule the loop. And once you have them in line, then…

02 Don't be afraid to stop looping

After decades of making music, we think we know the Big Secret, the doorway to a finished track, and this it. This is the point, the one tip that will make your purchase of this magazine, or your investment of time in reading this article on MusicTech.net, worth it. This is the Holy Grail, The Area 51, the Crystal Skull (okay, not the Crystal Skull) of music making: the 'one thing' that people need to know about DAW use and making music with technology. So… When you have a bunch of loops that work well together… here goes… Stop. Looping. Them. And Arrange. Them. Quick, while you have the motivation, while you have the time. Arrange them now!

Okay, it's obvious, right, and you might feel slightly disappointed after our big build up, but this really is important. Technology – an increasing number of grid controllers, hardware sequencers, →

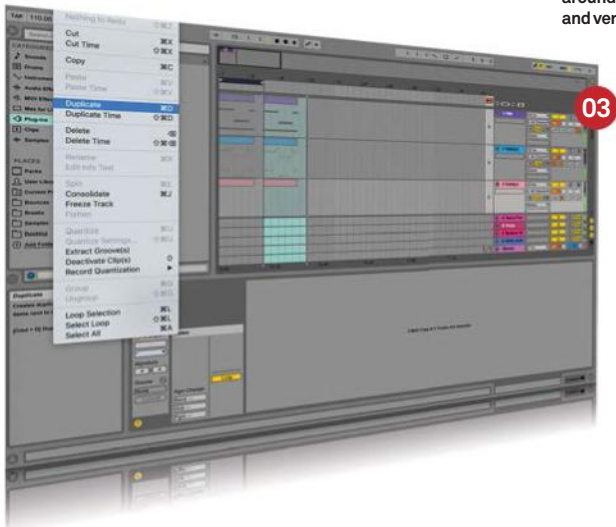




plus every software DAW – will encourage you to loop, and you will sound excellent by doing so. But the next step is the crucial one, the one that takes you away from every other bloody looper on the planet. You are the girl or the guy that *stops* looping and starts arranging. And you know what? It's…

03 Easier than you think

In certain DAWs like Live, arranging can simply be a matter of recording whatever clips you launch and stop – all movements get recorded, and you can tweak them afterwards if you make any mistakes. Lo and behold, you have an arrangement! In other DAWs, arranging is perhaps less obvious, but one sure-fire easy fix for any DAW is to copy the loops you are so happy with and repeat them for the duration of a song. Then it's simply a case of deleting certain parts of your new arrangement as the song progresses. More experienced DAW users might see this as a cop-out or a 'cheat' way to arrange, but it's what DAWs were built to do: easily repeat stuff so you don't have to play it again. So make use of it, but, like everything in the world of music production (and indeed life), use it in moderation…



2. The trick is to get to the point when you are happy with your loops, and then switch looping off. Look at all that space to fill

3. Duplicating parts in Live is also easy within an arrangement, but is arguably less used than in other DAWs

4. Starting off any dance arrangement with beats is one of the golden rules to help the DJ

5. Use the arranging features in your DAW to move big chunks around in one go, like the chorus and verse

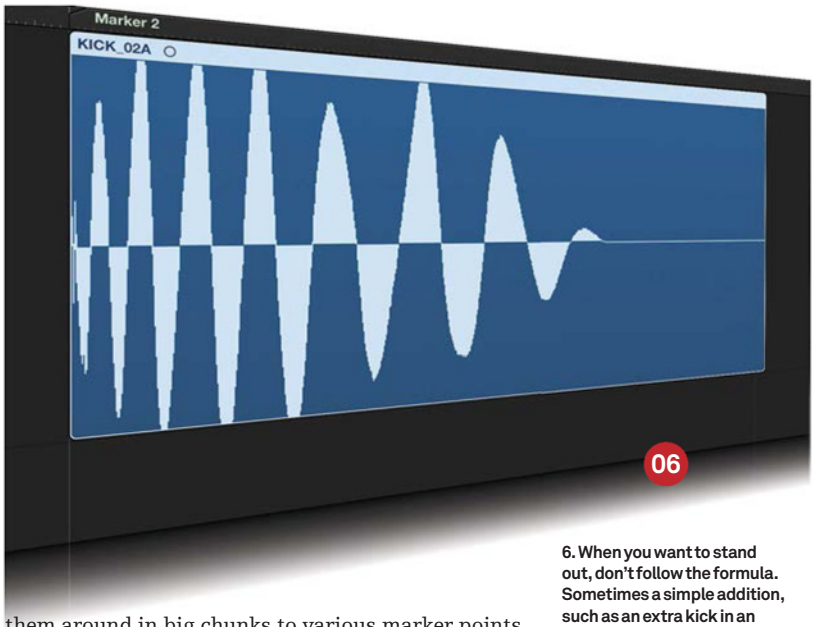
04 Follow the formula

Do dance music? Been told your music is formulaic? That's not *such* a bad comment, actually, because dance music does demand a certain amount of formula in the arrangement. We'll gloss over the arguments that say that it's the blander and formulaic styles of dance music that are killing it – people have been saying that for three decades now – and point you in the direction of some of the more obvious rules instead… Open with your beats. Obvious, yes, but DJs will want to mix your beats, not your notes. Technology such as NI's Stems is making this one a little more redundant, as DJs are getting more and more access to your song parts, but it's still safe to open on a bunch of tuneless beats than it is your breakdown. There are other arrangement tips, which will be specific to your genre of music, so a really useful and more general tip to follow here is…

05 Reference, reference, reference

As with mixing and mastering, learn from the pros with arranging. And the best way here is obviously to listen to other artists' arrangements. Consider even loading an entire song into your DAW by someone else, but one similar to the style you are producing, and noting its intro length, verse lengths, breakdowns, choruses, everything… Map out your production in the same way using your DAW's arranging tools. Many have built-in arranging features that allow you to title parts of your production as intro, or verse etc, and then move





6. When you want to stand out, don't follow the formula. Sometimes a simple addition, such as an extra kick in an arrangement, can create a whole new genre…

7. Automation can help vary your loops, from simply increasing the volume (as shown), to changing sounds

8. Live is a great DAWs for remixing. Simply load up a bunch of different beats to trial styles and genres. If it sounds good, you might have just created a remix of your own tune!

them around in big chunks to various marker points. In this way, you'll quickly produce an arrangement and wonder why you were ever stuck with just a few bars of loops in the first place!

06 Don't follow the formula

Important one this – almost as important as the 'break out of the loops' one. You could argue that by us saying 'use loops' and 'be formulaic' with your arrangements, we are contributing to the death of music. And you'd be right! Kind of. Actually, we're only advocating using these two methods up to a certain point in your productions, or during your music-making careers. Sooner or later, you will have to break the mould of whatever genre you are producing and try and make your own mark on it. There is so much music these days, in so many different areas, that in order to stand out, you have to do something a little different, and sometimes this may just be something a little different in your arrangement. So look at what everyone is doing and tweak it. An extra kick here? A few extra bars on a breakdown? More percussive elements to add swing? All these and more can be scattered across an arrangement to help take it up a level.

07 Use automation

To our minds, automation is one of the great music technology advances. It's up there with MIDI, virtual instruments, audio recording and Bro Country. It's yet another DAW feature that saves you time and enables one person to do what, at one time in our music-making history, was a job for a few people. This can be as simple as fading tracks in and out, but go a few steps further and you can automate instrument parameters so that a synth, for example, will change in sound while it's being played, or effects could be added during a chorus, or even the tempo could be slowed down or speeded up. Almost anything can be automated, so while DAWs encourage looping, they also encourage the variation of these loops with automation.



08 Consider the remix

The real beauty of the DAW – and we really are finding many things to thank our software friends for here – is that once you have a good-ish idea, you can tweak and refine it (pretty much forever if you're not careful, so read our tips on finishing!), or you can just leave it or come back to it. Consider a third option, though, one that is perhaps an extension of the 'Don't follow the formula' rule: the remix!

It's likely that up to this point, you've been working on a particular tune in a particular style. We previously told you to tweak it and break the mould, but at this point in your arrangement, it could also be time to twist it and smash the mould by remixing. You can try a couple of very quick tips here to change your tune very quickly, before we step into an entirely new tutorial on remixing (check MusicTech.net for plenty of guides and p20 of this very issue!).

First, change your beats, or lose them altogether. Stick something else – other beat loops – beneath your song and see what happens. Blend generic elements – dark trap with hip-hop, techno with country, and so on. Sounds crazy, but a thousand sub-genres were born this way. Also consider radically changing your tempo, or simply shifting your MIDI tracks up and down to hear how notes played on one instrument play on another. Beautiful accidents can happen this way, and the charts are full of them… MTF



MTF Feature

THE ULTIMATE GUIDE TO REMIXING

Fancy remixing other people's music, or having your own tunes reworked? Here's everything you need to know in the ultimate *MusicTech* guide…

Sitting down with a guitar or a computer and trying to create something from scratch can be one of the hardest things for a musician to do. Everyone has good days, of course, when the music flows easily. Other times though, the pressure we put on ourselves to write music can make things tougher. This isn't as often the case with remixing, that long-established technique of re-interpreting and reworking music made by others to create something new and sometimes even radically different. When you remix, the building blocks are handed to you and it's up to you to turn them into something. In a way, much of the legwork has already been done and all that's left to you is to interpret it in your own style.

This is why remixing is so popular and why some people even prefer it to writing original music – although some remixes depart from their source material so radically that they could arguably be described as new compositions. As well as being great fun, some remixes even end up being more commercially successful than the original song.

Starting the remix

You might tentatively suggest that bands playing covers of other bands' songs were the original remixers, but it wasn't until the advent of electronic music that the concept of the remix properly entered popular consciousness. True, early hip hop sampled a lot of funk and disco records, and dub music in the 1970s blended a lot of instrumentals with new vocals, but that wasn't quite the same thing. Official remixes really started to appear in the 1980s along with the rise of MIDI-capable equipment, early sequencers and 4/4 dance beats. These factors made syncing elements of different tracks together much simpler, and so remixing grew in popularity.

As the 90s gave way to the 2000s, computers and DAWs were becoming more powerful than anyone could have imagined in the 1980s. As it turns out, these are the perfect tools for remixing. There's no loss of quality when swapping digital files, everything can be synced precisely, and virtual instruments mean a much wider sonic palette than had ever been possible before. With the rise of the internet, remixing was no longer even constrained by where you lived. It became big business, not just in the dance world but in pop too.

Getting remixes of a track can be a great way not just to pad out a single release but also to broaden the appeal of your music, to take a great acoustic song and translate it to the dancefloor, for example. If you are a remixer, re-imagining the tracks of other artists gives you great creative freedom and also the possibility of a payday if things go well. Remember, remixing doesn't necessarily just mean sticking a 4/4 dance thud under something, even if that's still common. With the tools available to you in almost any DAW, the sky is the limit when it comes to reworking tracks for a remix. We'll show you how it all works over the following pages…

PREPARING FOR A REMIX

Whether you're handing over your stems or receiving material for a remix, here's what you'll need to do.



The ground rules for sending or receiving material for a remix are really two sides of the same coin, and rooted in basic good practice for producing music in general. They begin with the really core stuff like setting a project folder for your material, knowing what sample and bit rate your project is going to be at, recording to a click track, and setting markers correctly when exporting. Computer-based recording and DAWs make both ends of the remix process considerably easier than it would once have been, but you can help yourself by following the correct procedures.

Getting started

Let's begin with a look at how you ought to be working if you are planning on having a track remixed after you have finished it. Actually, you should always work to these rules because you never know if you might need something reworking or changing down the line. While it's possible to remix or overdub a track that's not been recorded to a click, for example, it's far easier if it has been in the first place. This is really the key: make sure you're locked to a click. Having a solid time base is essential both to editing inside your DAW since they all work on a grid system, and to someone else adapting your project to their setup.

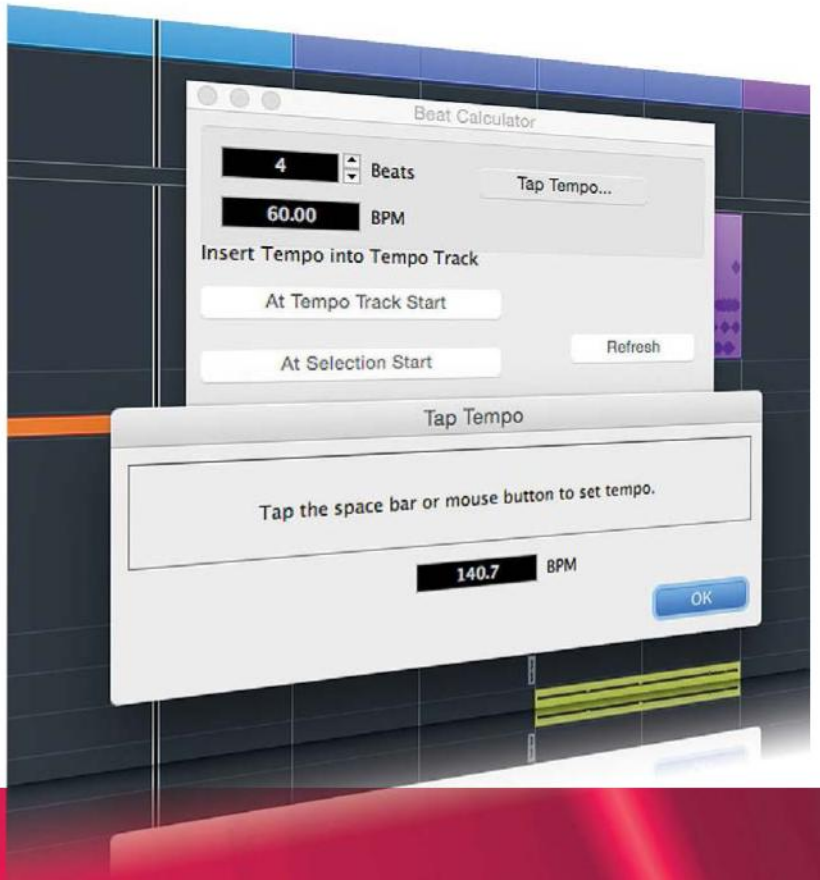
Your tempo doesn't even have to be sensible, it just has to be consistent. Most DAWs will allow a decimal point in the BPM, like 117.6BPM, for example. And while there's no real reason not to go for whole number values most of the time, you can use any tempo as long as you stick to it, make a note of it and tell the remixer so they can set their DAW to the same tempo. The same goes for formats. Pick a setup like 32-bit, 48kHz WAV files for example, and convey this to the remixer. Files can be converted on import but it's better not to have to. In the event that your track contains tempo changes, you'll want to be using a DAW that can export a tempo track or map if the change is gradual, or simply make a note of the tempo changes and where they occur so they can be recreated at the other end.

This assumes that you are both using different DAWs, though you may not be. Indeed if you are both using the latest version of Logic, just as an example, you can send a project folder with a raw project file and audio files. You'll still need to account for any difference in plug-in collections of course, as the likelihood of your having absolutely identical systems is small. In most cases you will be using different DAWs and have different plug-in collections, so it's better to take the 'all in' approach of providing properly formatted material.

Below: A beat calculator can help you sync up stems when you don't have tempo information

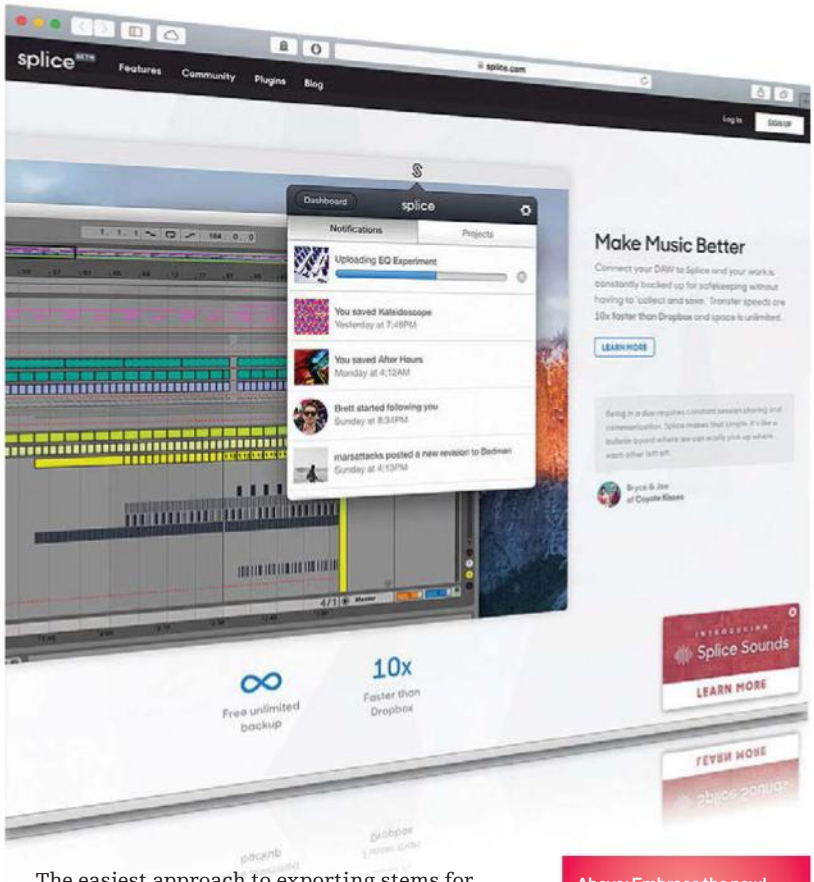
Right: Setting the left and right markers with snapping active is essential when exporting stems and importing at the other end

You should always work to these rules – you never know if you might need something reworking later



Rules for exporting

As the person providing the material for the remix, you have certain responsibilities to make your remixer's life easier. The major one is to export stems in the correct format. Begin by setting the left and right locators precisely over bar markers, either starting at zero bars, 1 bar, 5 bars or wherever as long as it's snapped. If your stems begin at 3.75 bars into a project it's going to take the other guy ages to get everything lined up, and it's an easy mistake to avoid making. If they have project BPM and stems that start on a bar marker, your remixer should be able to import everything easily and have it immediately in sync.

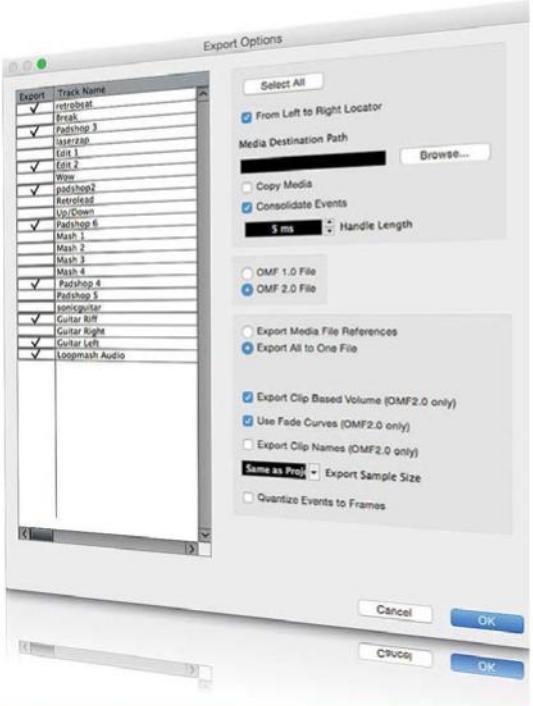


Above: Embrace the new! Use a service like Splice to upload, collaborate and remix tracks.

The easiest approach to exporting stems for remix is to just export every track as a separate, continuous stem. Yes, you might have a horn part that only comes in on the chorus, but trying to cut the stem up can just create confusion. The remixer will be quite capable of chopping that horn part out to move it around if they wish. Working with simple stems, one for each track, is the most straightforward approach.

Of course you can export loops and individual MIDI files if you prefer, it'll just take longer than batch exporting. Or, export a bunch of stems and also provide MIDI files, which gives them extra flexibility to rework your parts in MIDI. For the maximum number of options, your DAW might let you export both dry and wet versions of stems, pre and post-fader. Removing your effects processing from one version gives the remixer the choice to use dry or wet stems.

The easiest approach to exporting for a remix is to export each track as a continuous stem



Alternative formats

If you choose to send project data instead of stems, be sure to include all the samples and source audio files. If plug-ins aren't identical between systems, suggest an alternative to re-link to the MIDI tracks or mixer channels. Some DAWs also allow the use of OMF or AAF files for export and import. These are special file formats originally developed for moving between audio and video post production facilities, to cope with the different software in use. They hold certain information about a project like the positions of audio events and clip names, though they're far from comprehensive. These can save you rendering everything out, but are better suited to audio-only projects without tons of special timestretching or automation involved.

A more recent solution is to use a cloud service like www.splice.com to upload and collaborate on raw project data, removing the need to render out. At present this supports FL Studio, Logic Pro X, Ableton Live and GarageBand and it's limited, but definitely one to consider when it comes to sharing your raw material with others for them to work on. There are other online services that let people remix your music with your consent, though they tend not to support really heavyweight project data at the moment, usually just stereo mixes or loops. As things stand you're still as likely to get a DVD or a link to a cloud service to download a few gigabytes of WAV files, hopefully with some tempo information to help you set up correctly.

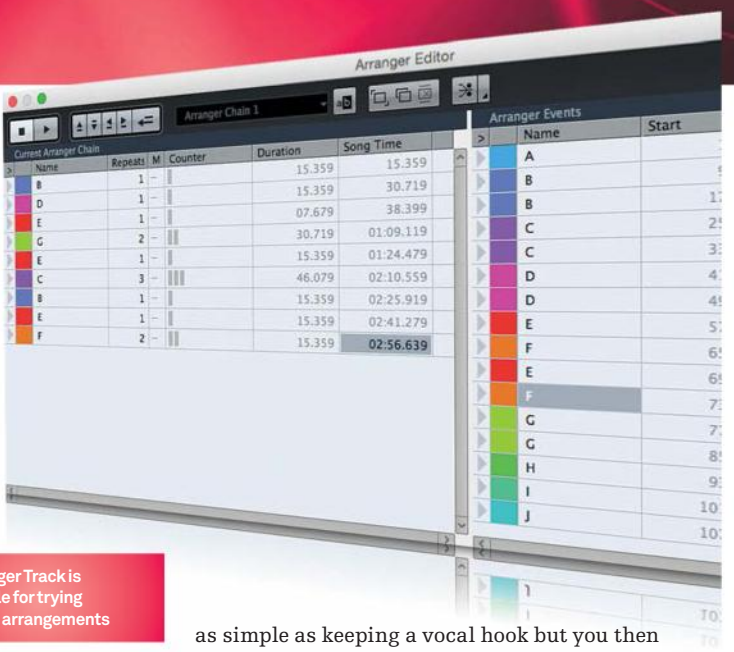
THE OMF STANDARD

The OMF 1.0 format could cope with audio clip position, project tempo and some other basic concepts, and OMF 2.0 added volume settings and fades for events as well as clip names. OMF doesn't support MIDI files (though these can be exported separately) or plug-ins, because DAWs work so differently. As such, OMF is fine for exchanging audio-based projects but certainly has its limits. Most 'full' DAWs support OMF export and import though things like advanced mixer settings, scores, surround and so forth are well beyond the scope of these kinds of file formats. OMF has been succeeded by the AAF format which is more advanced and can include more information but this is currently not as widely supported, though Logic Pro X and Pro Tools among others do support it. There's also the Mxf format for audio and video interchange, though this hasn't made much headway in the audio world yet. Different software supports different formats so it's worth investigating this before you start, as it will save you messing about later.

HOW TO APPROACH A REMIX

Remixing is all about creative decision-making. But how should you go about it?

When you remix music you may be given a brief or you may be left entirely to your own devices. This depends on who you are remixing for and what the goal of the remix is supposed to be. A club remix of a pop song, for example, is likely to need to be heavy on the beats and bass. If you've been commissioned to do one of these, you will have parameters inside which you will be required to work. For a B-side however (more likely to be billed as a bonus track in today's digital world) you may have much more freedom. A remix can be literally anything, but it will usually need to retain at least a small amount of the material from the original track in order to be recognizable as a remix and not a completely new track. This could be as simple as keeping a vocal hook but you then discard everything else. Alternatively, keep most of it but speed it up and add more drums. There are lots of examples of remixes that take wildly differing approaches and still succeed.



An Arranger Track is invaluable for trying different arrangements

A remix can be anything but will need to retain at least a small amount of the original material

Break it down

A good starting point once you have established any requirements for a remix (like it needing to be a dancefloor filler for example) is to break a track down into its constituent parts. This is why it's so important to provide the component parts of the →



Along with the Fat Bustard, this is the **best-sounding analogue summing mixer** we've heard. A world-class product — **John Pickford** *Music Tech Sept 2015*



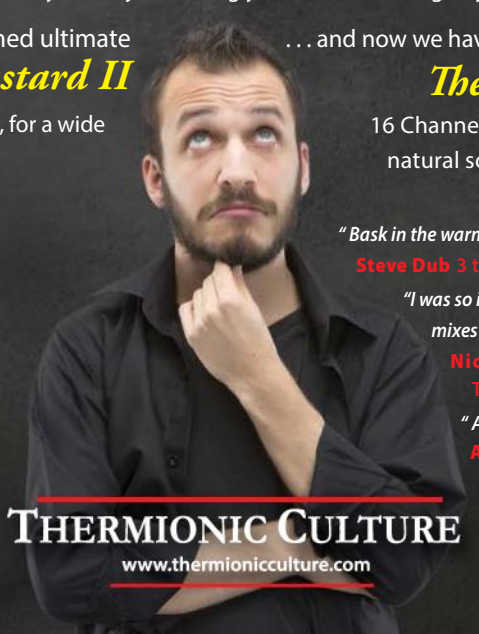
CHOOSING YOUR BUSTARD

("or maybe it is just cutting your coat according to your cloth")

On the one hand we have the long established ultimate analogue summing mixer ***The Fat Bustard II***

- Hugely versatile, 14 channels, with many extras, for a wide variety of uses.
- Pan controls on 4 channels.
- Unique Varislope Top and Bass lifts plus filters.
- 'Attitude' control for added harmonic content.
- Stereo width controls / Bass to centre and mono.
- ELMA switch to control main level (fadeable).
- Monitor outputs with level control and switching.

**** The superb album "Going Back Home" by Wilko Johnson and Roger Daltrey was Recorded at Yellow Fish and mixed through a FAT BUSTARD at Electric Davyland.*



. . . and now we have a **2015 Gear of the Year Award** Winner

The Little Red Bustard

MusicTech SILVER AWARD HIGHLY COMMENDED

16 Channel slimmed down version of the 'FAT' keeping same natural sound, 'Attitude' and with new 'Air' control added.

What the users are aleady saying . . .

" Bask in the warm Glow and Silken Top of this awesome sounding box"
Steve Dub 3 times GRAMMY Award winner. Chemical Brothers, UNKLE . . .

"I was so impressed with what the Little Red Bustard added to my new mixes that I've gone back and run all my recent projects through it!"
Nick Brine, producer/engineer Rockfield Studios. Oasis, The Darkness, Teenage Fan Club . . .

" Amazing Box-makes every sound BIGGER"
Alan Emptage owner One Louder Studios. Arcade Fire, Joss Stone . . .

" Vic you've done it again you old Bustard!"
Dave Eringa Producer, sound/mix engineer. Wilko Johnson & Roger Daltrey,*** The Who, Manic Street Preachers . . .

THERMIONIC CULTURE
www.thermionicculture.com

AN ELECTRONIC REMIX

How to turn guitar music into EDM

When taking any kind of music and creating a dance version (using the term loosely to mean anything electronic that's destined for a dancefloor) your first port of call is usually replacing or at the very least augmenting the beats with something heavy containing a 4/4 kick. That might sound clichéd but just tune into any commercial radio station and you'll hear that stuff played all day long. Heavy beats are almost a prerequisite, and these can come from a virtual instrument, a loop collection or a hardware drum machine. The importance of having original recordings properly recorded to a click becomes apparent here as you add MIDI parts to real recordings, quantize and program them to fit. From there, you'll probably dial down the guitars and maybe even replace the bassline with something electronic, or dirty up the existing bass parts. Whether or not you lose the guitars depends on you, but it's usually nice to keep a few in there. EDM prioritises beats, vocals and melodic hooks so these are the things you will probably need to keep in the forefront. A load of mushy guitars in the mid range won't be too helpful so strip it back to the hooks and then embellish with electronic elements. It doesn't hurt to juice it up too, maybe adding delays and reverbs and master pretty hot, as is the style in much electronic music.

track as stems or otherwise as separate tracks, because a remixer needs access to everything, not just a stereo mixdown. Different people take different approaches to this. Some artists will concentrate on a vocal as the key element of a track and build a new track around the vocal performance. In the case of instrumental music it might be a bassline, a riff or hook or a drum loop that particularly stands out and forms the cornerstone of the re-imagining. It's usually pretty obvious what the heart of any piece of music is, be it a vocal hook, a guitar lick or something else. In some cases, a remixer will actually add a completely new hook of their own and it's in situations like this that a remix can become more famous than the original.

It's usually pretty obvious what the heart of a track is, be it vocal hook, guitar lick, bassline or loop



Extracting audio to MIDI data is a great way of deconstructing material for a remix and recreating new parts

Modern DAWs let you manipulate time and pitch in incredible ways and these are two important things to think about before you start a remix. Would the track sound better, or at least substantially different, if it were slowed down, sped up or its pitch changed a little? Remember that tempo changes don't automatically also mean pitch changes any more, the two are exclusive. It's often better to make these kinds of choices at the start because it cuts down on the amount of stretching of individual clips and parts that you have to do. So if you've whittled it back to the drums, bass, vocal and one guitar line, perform your stretching on these early on then build the new elements around them.

Creative decisions

Just because you can chuck everything out doesn't mean you necessarily should. Every project is about judgement. Keep what works and replace or remove what doesn't in the context of your interpretation. It might be that a rock track has some synths added to it, perhaps some extra electronic beats and sounds great. Remixing can work the other way as well. Some artists prefer to take away, to strip productions back to their essence and uncover the beauty of a song or the raw power of a groove. It all depends on the ingredients you're working with.

As well as instrumentation the other main consideration is arrangement. Again this can vary from subtle modification to wholesale change. If you're treating a vocal performance as the heart of a track then it's likely that the structure isn't going to change too radically since the vocal has a set length. Of course you can mess with it a little, but it might not be dramatically different in structure. If you disregard the structure of a song or you're making it instrument-led, the rule book goes out of the window. Here, you have carte blanche to start lifting, moving, duplicating and deleting any part of a track.

Many DAWs have tools to help you with this, such as Cubase's Arranger track or Reason's Blocks mode. Others have their own methods for creating multiple arrangements within a project. You don't have to pay attention to traditional song structure, though bear in mind that most people prefer their music to have at least a passing acquaintance with structure so anything too chaotic is going to have a limited audience. Still, you can get away with a lot. Many electronic remixes really push the envelope when it comes to frenetic, cut-up structures. Plug-ins like Stutter Edit or Turnado also mean that really quite complicated mashup effects are easy to achieve without hours of programming.

You have a lot of choices when it comes to arrangement thanks to the flexibility of DAWs. You can reduce a sprawling track down to its core elements and create a tighter, more focused version by trimming the fat. Or, take a simple idea and explode it out into an epic piece of music, adding



Fatboy Slim: artist, DJ, and prolific remixer

and embellishing to create something altogether more majestic. Truthfully, the proliferation of methods by which people consume music these days means that you have more freedom. A 12-minute noise epic isn't likely to make it onto the FM radio, but it could find an audience on Soundcloud or one of the many online radio stations dedicated to niche genres. A club version of an indie song is more likely to find an audience now than when you had to fight for attention on national radio. The possibilities are not quite endless, but they're many. →

AN EXPERIMENTAL REMIX

When the going gets weird... do a remix.

Putting a thudding 4/4 beat behind a track is the stereotypical way to remix, and it is indeed still popular. You can go the other way though, taking a song or a track in a completely new and unexpected direction. Some artists specialize in this and while these projects might not get the dancefloor moving, they are often held to have greater artistic merit. You'll start as before by splitting a track into its constituent stems or parts, but this time instead of adding drums you might take them away, or add ominous electronic percussion where before there was a drum kit. Adding textures is a good way to create an atmosphere, and synths and guitars are really great for this. If there's a vocal, consider changing it in some way. Perhaps tempo or pitch stretch it, add vocoding or layer multiple versions of it on itself or add extra harmonies or backing of your own. Ghost a vocal with a distorted guitar or piano part.

While you're being more experimental you can mess with structure more as well. Reverse things, take choruses out and add new sections that weren't present in the original. If you're remixing into a different but more conventional style, give it the full treatment. Add a dubstep underpinning to an acoustic track, or an orchestral backing to a piece of dark electronica. Some of the most interesting re-versions of tracks occur when an artist goes in a completely unexpected direction. There are limits of course, so make sure it works!



REMIXING IN PRACTICE

Here's how to remix in your DAW

When it comes to remixing, your DAW already offers most of the tools you will need. Naturally they all take a slightly different approach but the basics are the same. Set up the project based on the format of the material you have received – for example 120BPM, 48kHz, 24-bit files – and then set the left marker to a definite bar marker, ensuring that snap to bar is switched on. Then import the audio stems all at once, making sure they start at the left marker. If the person supplying the material has done their job you should find that the audio tracks are all in time and perfectly synchronized. That's 'if' they have done their job…

> If the **person supplying the material** has **done their job** then you should find that **the audio tracks are all in time and perfectly synchronized.** That's **'if'…**

If this doesn't happen you might be better off asking for properly exported files. If that's not possible you can expect to have to spend some time looking at waveforms and trying to work out why the timing drifts off as the stems play back. You can use your own click track to work out how badly off the timing is if this happens.

Most DAWs are pretty good with time stretching though it can still be a pain to align entire tracks to your musical grid. In the event that you haven't been told what tempo the stems are at, a beat calculator will usually let you tap the tempo to work it out, though again you'll have to spend time realigning and stretching parts. If things get really fiddly, one trick is to cut parts up and stretch sections individually then reassemble them in the correct tempo, but it's a hassle. Like we say, don't be afraid to ask for what you need…

How you proceed depends on what your aims are for the remix. You may want to, for example, mute everything but the vocal and then bring in completely new beats underneath it to start with. If you're planning to time- or pitch-stretch the vocal track, now is the time to do it to cut down on work later. If you pull a vocal up by 10BPM or a couple of semitones, it will be easier to start building the new track around it from this base rather than changing lots of tempos later.

Some DAWs will make life a lot easier even if you do change your mind later. Some, for example, render all audio elastic, so tempo changes en masse aren't too arduous.

Much of the time you may want to keep hold of the basic character of the track by keeping certain elements in place such as a bassline, synth or guitar hook if they are particularly good. But you can mess with them for example by adding effects like delay or distortion to alter the feel of the track while maintaining its basic melody. A common technique when producing electronic remixes of rock or indie tracks is to identify the hook and then repeat it creatively. Drums might be replaced or supplemented to beef them up, and arrangements altered to change the structure of the track. Of course you could take the opposite approach and substitute new melodies, beats and basslines and simply keep a few key elements of the original – the choice is yours.

MTF Step-by-Step Remixing in Live



01 Ableton Live takes a slightly different approach and one that's useful for remixers. Here, some stems have been imported and then a loop identified. Right clicking, we are able to slice it to a new MIDI track and then trigger it as a new instrument.



02 Add a new bass instrument and either use the extracted MIDI information from the bass clip imported previously or program a new part. Use Clip view to work with audio and MIDI clips for a more DJ-style approach to remixing.



03 Use Live's great MIDI effects like arpeggiators and randomizers to help you compose new clips and parts underneath the bones of the project you have kept from the original, to make something new and unique.

MASHUP TOOLS

Modern tools can take a lot of the legwork out of the remix process...

When remixes first became a popular thing to do in music, the tools on offer were significantly less advanced than they are today. The process would involve editing tape, sampling into hardware samplers and manipulating the results, and recording overdubs largely using real or hardware electronic instruments. Even when more outlandish stuff became popular in the electronic world, a lot of it was down to many hours spent programming and automating MIDI to get the desired effects.

These days, a relatively small outlay will get you access to some mind blowing software tools for reinterpreting digital audio in ways which didn't really used to be possible. Plug-ins can now take a real-time audio signal and mangle it in amazing ways without you ever having to worry about tedious stuff like staying in sync. Most of this even remains virtual if you so choose, meaning you don't even have to render effects down if you don't want to.



Sugar Bytes' Turnado is a powerful mashup multi-effect

Mash it up

Let's look at a few of the best plug-ins for remixing, beyond the kinds of mix tools you would use as part of any music project. Izotope's Breaktweaker was designed in collaboration with electronic producer BT and is technically a virtual instrument into which you can load any samples and completely re-sequence them based on its advanced onboard →

Focal Alpha
For the producer inside you.

FOCAL PROFESSIONAL

Whether you're an, engineer, producer, writer or DJ, the most important gear in your studio is your monitors. But the good ones are usually so expensive! Introducing the new Alpha 50, 65 & 80. Designed with brand new, Focal-manufactured drivers: aluminium inverted dome tweeter c/w new 5", 6.5" & 8" polyglass woofers, all powered by dual Class AB amp's.

NEVER BEFORE HAS THIS PERFORMANCE BEEN AVAILABLE IN THIS PRICE RANGE.



SCV

www.scvdistribution.co.uk

Distributed by SCV Distribution: Call 03301 222500 for your nearest dealer

In stock now at: Absolute Music, Dawsons, Dolphin Music, Funky Junk, GAK, Gear4Music, Giraffe Audio, Guitar Guitar, Juno Records, Kazbar Systems, KMR, Music Maker (Dublin), Music Matter, Production Room, PMT, Red Dog Music, Rubadub, Scan Computers, Studioxchange, Westend DJ, Westend Production.

sequencer. Although billed as a beat making tool, in practice it can be used on anything you like. It has a MicroEdit Engine that lets you repeat or chop up any audio slice at an incredibly fine resolution so you can turn beats into melodic notes, texturize melodic sounds with rhythms or use beats as a synthesis tool by locking drum hits to individual pitches. Different internal sequencer tracks can use different tempo and step lengths so it's almost like a mashup studio inside a single instrument. Manipulating samples becomes far more flexible than anything you can achieve with regular audio edits.

Also from Izotope is Stutter Edit, an audio effect that lets you play effects like an instrument. It continuously samples live audio input into a buffer and then you use MIDI notes to trigger multiple sets of complex effects on that audio. The result is that you can create incredible stutters, runs, sweeps, repeats and much more by assigning notes to effects and then tweaking the behaviour of those effects. There are a stack of available effects and even though you'll want to invest a little time in setting it up to your liking, it's like a remix in a box. Not dissimilar is Sugar Bytes' Turnado, a multi effect that's a little more approachable and has a ton of multi fx to route your sounds through. It can do everything from slicing to distortion and filtering at the same time. To add a new dimension to your remixes it never hurts to have a good pool of samples and loops to work with such as those from Loopmasters. As well as being used straight out of the box these can be mashed up in effects like the ones mentioned above for cool results.

MTF Step-by-Step Creating new beats with Izotope Breaktweaker



01 Load an instance of Breaktweaker and load a preset. Each of the lanes represents a different sound and these can be swapped for different samples or some of your own sounds. Press a MIDI note to trigger the sequence.



02 Select any of the note events in the sequencer or add your own using the mouse, and change its position and duration. Click in the MicroEdit window and drag the mouse up or down to change the tension and behaviour of the step for cool sounding results.



03 Use the Pattern section at the bottom right to build up a series of different patterns. These can be sequenced in the main DAW sequencer or exported or bounced down to audio in order to play with them in the context of the main remix project.

RELEASING AND HOSTING YOUR REMIXES

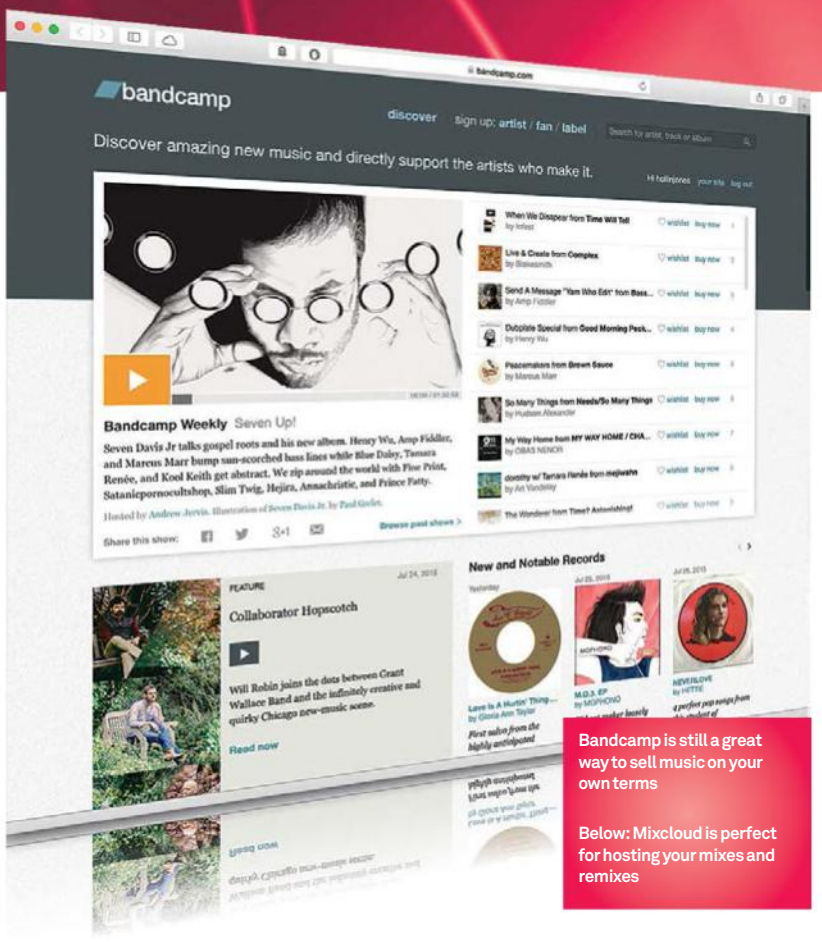
Now it's time to get your remixes heard...

The days of sending material to the CD pressing plant are more or less gone. Nowadays you're much more likely to simply send someone a WAV of a mixed and mastered remix once it's done, and they will deal with releasing it if that's the arrangement you have made. If you are releasing or hosting stuff yourself you have several options, but there are some things to be aware of first. If yours is an official remix, the person whose track you have remixed will own the copyright in at least part, if not all, of the recording. They will probably have written the track and will have the rights to the name, melody and any recorded material they supplied to you. So it's vital to be clear (preferably before you even start any work) about what the deal will be for the finished track. Maybe you will receive a fixed fee to sign over the rights to the remix entirely, or perhaps there will be a cut of sales that can be attributed to you via your membership of a collection agency like the PRS, MCPS, ASCAP or whatever your national association is. If the track contains any third party samples you need to know that the artist has cleared those too, since that adds another layer of complexity. What you shouldn't do is just start selling something yourself, unless you have explicitly agreed this first.

> The **person whose track you remixed** will probably **own the copyright of the recording**

Look to the cloud

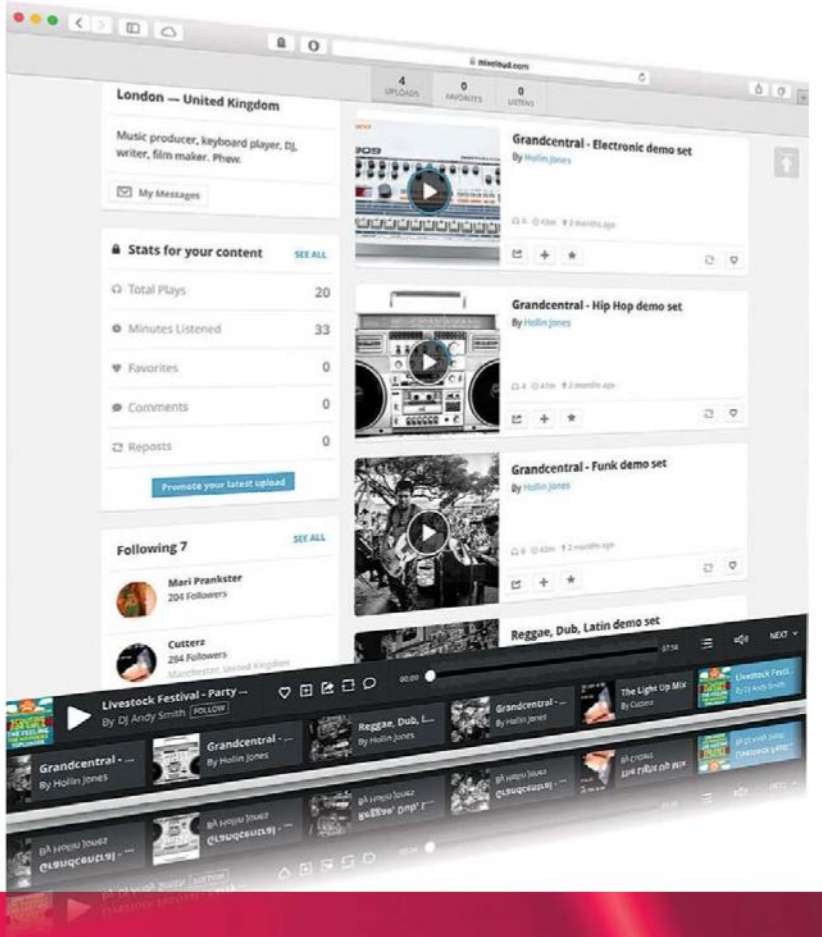
Hosting music nowadays is easier than ever, and Soundcloud is still the go-to destination for uploading your own music and embedding it on other sites. Mixcloud is great too, especially for remixes. Bandcamp is still one of the best ways to



Bandcamp is still a great way to sell music on your own terms

Below: Mixcloud is perfect for hosting your mixes and remixes

Ultimately **remixing can be a good way** to **promote both you** and the **artist you have remixed**



TECH TERMS

• STEMS
Individually exported audio files that collectively contain all the material in a project. Batch export is the quickest way of creating all of these in one go.

• TEMPO MAP
A special file or MIDI data you can export to tell the remixer about any speed changes that happen through the course of a track.

• BOUNCING
Rendering MIDI or audio down to audio files, often with effects. This provides the remixer with all relevant material even if they are running a different system.

• EXTRACT TO MIDI
Many DAWs let you analyse audio clips and convert their note and pitch info to MIDI which is great for capturing melodies or beats and reprogramming them.

• REX FILES
Combinations of MIDI data and sliced audio files that can be re-sequenced and generally manipulated. Some DAWs have dedicated REX player modules.

• OMF AND AAF FILES
Special files that contain some – but not all – content and info about a project. Useful for sending a remixer a project without having to export everything first.

• TIME STRETCHING
Altering the speed of an audio part without changing its pitch. Pitch stretching is also commonly available.

• MASHUP
Taking two or more tracks or parts of tracks and blending them together to make something new.

• THE CLOUD
Online, server-based methods for either swapping large amounts of raw project data or in some cases for collaborating on projects using web-based tools.

• MIDI SYNC
Synchronizing tracks or equipment – or both – so that the original material and the new stuff you add all stays in the correct time.

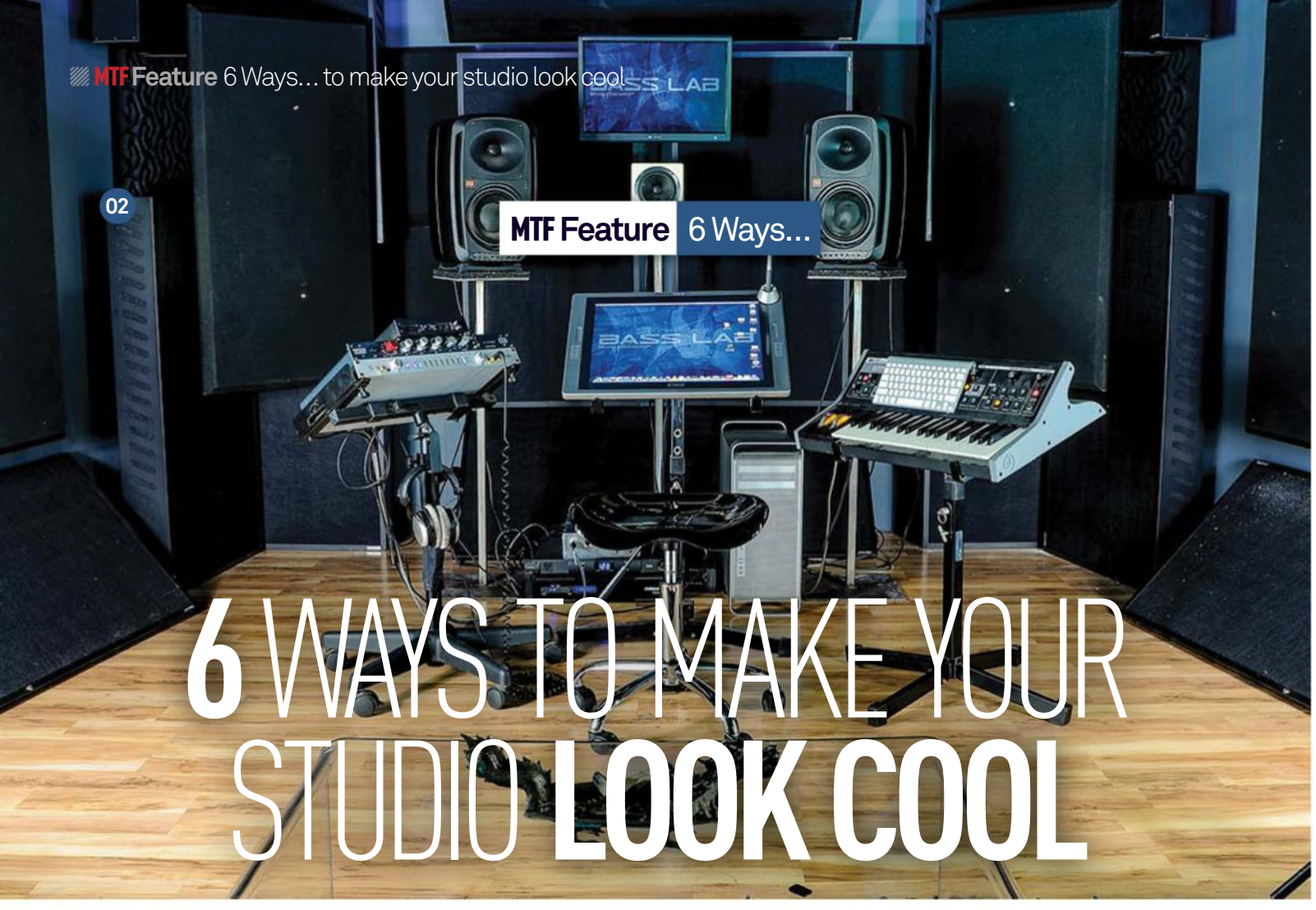
sell your own music and set your own price, with all levels of quality options including lossless. Most of these sites have decent free plans that give most people what they want though there's also usually an option to upgrade for extra features like sales stats. Beatport now only deals with labels but has an associated site called Baseware that still allows independent artists to sell music through online stores. Services like iTunes, Amazon and Spotify can also be used though you will have to go through an intermediary service such as Mondotunes since they rarely deal with people on an individual basis.

Radio and blogs are still a great avenue for promotion and all the usual rules of tenacity and casting your net wide apply here just as with promoting any music.

Of course if you have remixed an established artist or even someone who is doing their own promotion it should be easier to get them to promote your remix under the banner of the work they are already doing. Ultimately, remixing can be a good way to promote both yourself and the artist whose material you have remixed, so it's in everyone's interests to see it succeed. **MTF**

02

MTF Feature 6 Ways...



6 WAYS TO MAKE YOUR STUDIO LOOK COOL

Okay, you might not think that this is *the* most important aspect of studio life, but having a great creative environment will help your music production no end, especially when it is easy on the eye...

So should you really be concerned about how cool your studio looks? We're certainly not saying that it is as important as how it sounds – in fact, we always say that the sound *is* the most important aspect, right from getting the perfect recording through to listening to the end results on the most accurate monitors. BUT, let's not forget that music production is a *creative process*, so once you have your preamps, your monitors, your mixer, your mics and your computer sorted – and even your chair, for that matter – why not consider the environment around you? No, not that environment [*actually, yes, that environment, too* – Ed], but the room in which you sit, the room in which you create? Make it look great and you will make it a welcoming and creative place to be. So in order to do that...

01 Tidy, tidy, tidy

Got kids? You will understand this. Stuff on the floor, stuff on the desk, stuff everywhere, more stuff. LEGO. Bloody LEGO, everywhere! The easy route to a creative environment is a clean environment. Yes, it's obvious, too obvious, so obvious that you might not notice 'untidy creep', i.e. as you work within your creative space you *will* make it more untidy. In fact, as we're writing this feature within our studio setup, we've just noticed: it's untidy. Hang on... [noise of tidying...] There, that's better. Spend 10 minutes clearing your work space and your next session is guaranteed to be more productive.

02 Less is more (and, okay, more is more)

For a magazine that shows you a ton of gear pictures every month, it pains us to say: if you are working towards specific musical goals, then very often the fewer distractions you have in terms of gear, the better. And as is shown by (some of) the pictures on these pages, having a very stripped back setup can mean a very cool one. Having said that, and this is where we justify having a ton of gear, having a huge bank of outboard lights come on when you flick that master switch, can be just as inspiring. It's about a work/gear balance...

> For a mag that shows you **a ton of gear**, it pains us to say that **less is more**. Sometimes, that is...

03 Surround sound

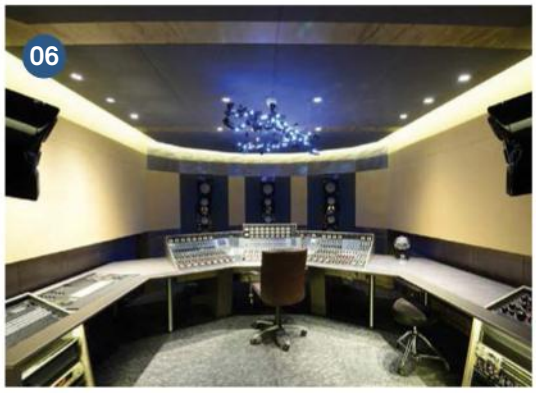
There's no doubting that there is some kind of











primal urge that kicks in when you sit down at your mixing desk or at a bank of keyboards, and surround yourself with buttons, dials, sliders and keys. It is a very controlling feeling – or is that just us? – so give in to it and make sure you immerse yourself. Have everything within reach and angled towards you; invest in proper studio furniture (or simply cheaper Ikea-style desks that have compartments and shelves) and a floor that enables you to roll across from one bank of loveliness to the next. It looks great and works great…

04 Personalise

It it *your* studio? Then, in the words of the best captain on the bridge: 'make it so'. You can have a mountain of gear, the coolest of sofas, the best of lights, but if it's not *yours*, then you might as well be anywhere. To make it personal, make sure you have a couple of reminders about whose space this is,

Is it *your* studio? Then, in the words of the best captain on the bridge: **'make it so'**

maybe who inspired you, where you came from and where you are going, the important words in that sentence all being 'you'. A stormtrooper on a monitor, a Kraftwerk traffic cone on a synth… Personalise it however you want, not just with gear but with *you*.

05 Lights, action…

We know that this has become an important one to so many of you. Good studio lighting not only looks cool, but can also be very inspiring. There's nothing worse than having the most incredible gear setup, a great-sounding acoustic room, only to turn on an artificial industrial office light to see what is going on. Invest in up lighting, strip lighting, even fairy lights if you must. Make it warm and welcoming, not cold and frightening. Green is good. As is blue. And lava lamps. Ooh, lava lamps…

06 Go wireless

As the world goes wireless, we in studio land should be rejoicing. Unfortunately, latency and other issues mean that a completely wireless recording environment is probably a few years away yet, so while you have to plug one thing into another – and let's face it, modular fans, that is still an attractive pastime in its own right – at least keep those wires out of sight as much as you can (unless you are the aforementioned modular fan, naturally). Invest in cable tidies, cable extensions and simply getting down on your hands and knees to push power leads, audio cables and computer connectors out of site but, importantly, still within reach. A good old fashioned patch-bay might be in order here.
And while we're on the subject of modular synths, yes we all know that they look cool, but the cables can look a mess. Try colour coding or even LED cables to blend points 1 and 5. Very cool… MTF

Ableton Live | Live In Depth – Aggregate audio devices

On the disc
DVD ROM
Accompanying project file included on the DVD

Beef up your Ableton Live audio ins and outs

Do you need more audio inputs and outputs for your Live rig? Before rushing out to buy another interface, make sure you're fully exploiting what you already have…

Sooner or later, we all have a moment when something changes unexpectedly in our studio or live setup; maybe we borrow a drum machine or synth from a friend, or somebody comes over for a jam, and suddenly the well-organised rig doesn't quite have enough audio inputs or outputs to go round. In the longer term, you can plan to buy another soundcard, or you can use external audio inputs on your hardware instruments, if they have any, and bodge something together that way. Either way, right there on the spot, you need something to make it work, and this is where aggregate audio devices come in, if you have an unused sound card lying around.

Under Mac OSX, we can combine two or more audio interfaces, including the one built into the computer, so that our favourite music software sees all of the connected physical inputs and outputs as one virtual device. This device will be available system-wide for any software, including Ableton live and other DAWs. In some ways, you might think we're not talking about an Ableton Live-specific topic here, but it's incredibly relevant, because Live is at the centre of so many stage and jam rigs. The understanding of aggregate audio devices fits very neatly into that world, just as you ought to know how to ReWire different DAWs, how to use MIDI Sync, and so on.

Typically these days, we'd be using USB audio interfaces for this. It's the most common form of connection for audio and MIDI peripherals, although FireWire is still around, and there are a few Thunderbolt interfaces available, like the Apogee Ensemble, and FireWire interfaces are still around – and FireWire is compatible with Thunderbolt if you have an adaptor. Just to be clear, when I talk about an audio interface, or a sound card, I'm referring to the same thing.

Combine **multiple devices** to create the **monster aggregate device** of your dreams

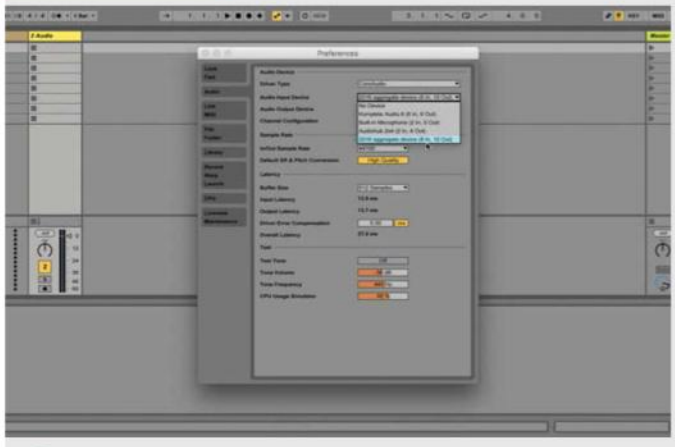
You can combine multiple devices to create the monster aggregate device of your dreams, although after a certain point, I'm not sure what'd happen regarding computer performance and latency – I don't think I've ever gone further than two interfaces plus built-in audio. I always make a point of using the mains

Workaround or long-term plan?

Is it worth buying another soundcard just so you can do this? Truthfully, I'd say no! I view aggregate audio devices strictly as a workaround for those situations where you're caught short on the input front, like when there's a gear failure in the studio, or you're having a jam with a group of players and there simply aren't enough physical inputs to do it properly. If you're looking to expand on your available inputs long-term, a better option would be to add something that connects to your existing interface via an optical cable, like Focusrite's OctoPre or RME's OctaMic.



MTF Step-by-Step Aggregate audio devices



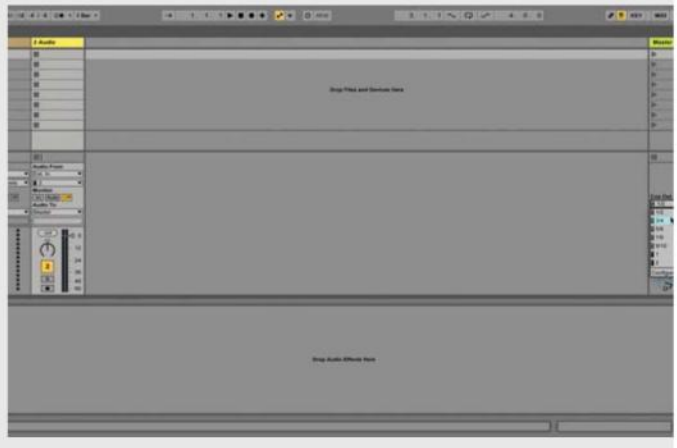
01 An aggregate audio device is a name for the connecting and configuring multiple hardware audio interfaces, so that they appear as one continuous set of inputs and outputs in software.



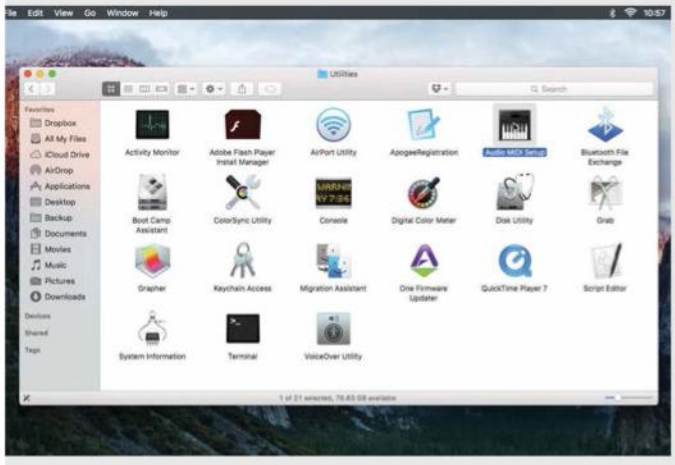
02 You'll need a Mac running Live, and one or more audio interfaces, to follow our steps. Any relatively recent version of Mac OS X should do, and I'm afraid that, yes, this is Mac-only.



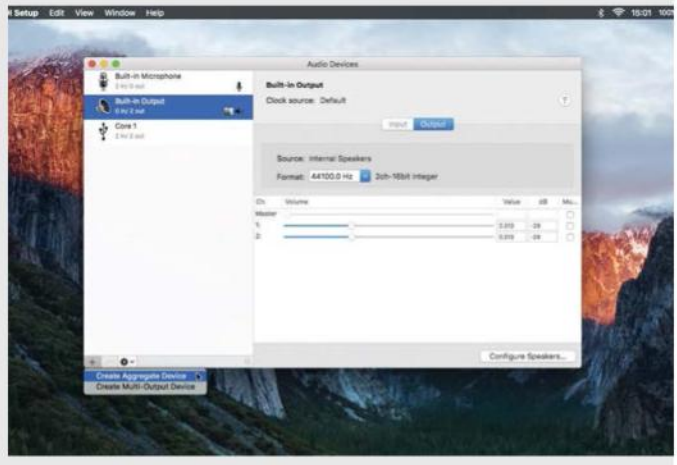
03 This can scale up from small to very big rigs. For starters, let's say you want a simple DJ cueing setup, which requires two separate stereo outputs, but you only have a basic stereo soundcard.



04 You can combine your Mac's headphone output with soundcard output to make a 'virtual' interface with two stereo outputs, and set them up so one outputs to PA/mixer, the other to your headphones.



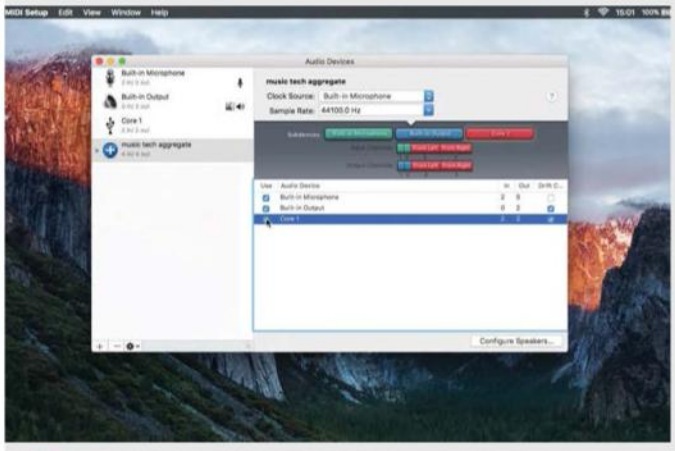
05 Make sure your soundcard is connected as usual – probably via USB. You don't want Live open yet. Go the OS X Utilities folder, which is inside the Applications folder, and open Audio MIDI Setup.



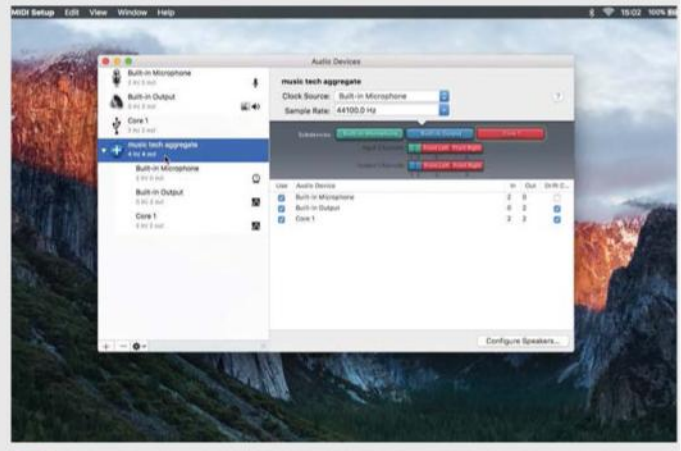
06 From the Audio Devices window, you'll see a panel that lists all recognised interfaces and any existing aggregate audio devices. Click '+' to add a new one, and rename it something sensible.

power adaptors for any interfaces in these setups as well. The ability to create aggregate audio devices is built into Mac OS X, which is why this entire tutorial is Mac-only. OS X is very convenient for handling these audio tasks, as well as having deep MIDI routing capabilities also built in – like the ability to configure wireless MIDI networks. You can use little audio interfaces like the Alesis Core 1, right up to, for example, the RME Fireface UFX. Live won't recognise a new aggregate audio device until it's relaunched, so bear that in mind if you don't see it listed in Audio Preferences. Talking of which, you'll also have to visit Audio/Input Devices/Configure and Audio/Output Devices/Configure and enable every input and output

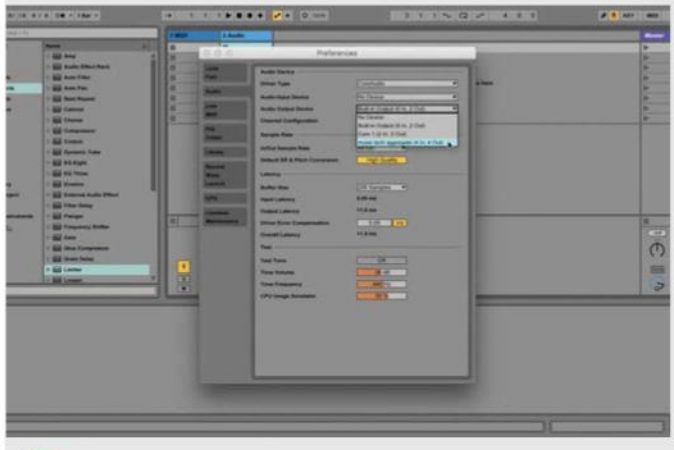
MTF Step-by-Step Aggregate audio devices (cont'd)



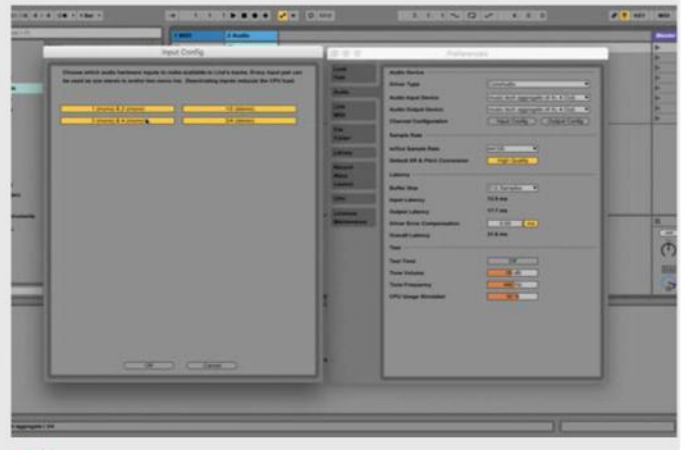
07 In the panel at the right of that, you should see a list of all available audio inputs and outputs, including Built-in Microphone and Built-in Output – which are the Mac's own hardware connections.



08 Tick all of them – you might as well include the inputs as you're not adding many, anyway. Quit Audio MIDI Setup. You can return here whenever you need to manage your aggregate audio devices.



09 Now when you open Live, go to the Audio tab in Preferences. Where you choose Audio Input Device and Audio Output Device, you can view and select your new Aggregate Audio Device.



10 Click the Input and Output Config buttons and you can enable all of the inputs and outputs in the device. From now on, Live will treat this as one device, just like any other. Close Preferences.



11 Open a Live set. Anything with some clips in Session or Arrangement View will do, though it's easier to view routings in Session. Open the In/Out View to see all four outputs available.



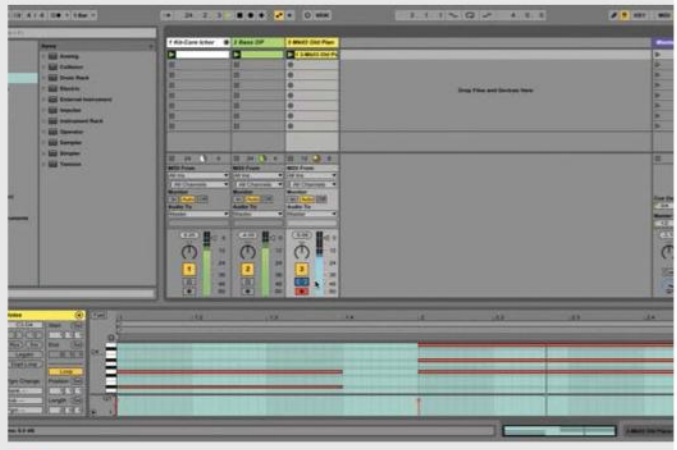
12 Under the Master Track, you can define the Master Out and Cue Out destinations – the one that will go to your headphones. By default, all tracks will go to the Master Output.

→ that you want to use – which is probably all of them! After that, of course, they'll all be available in in Live's In/Out View. How you use all those spare connections is up to you; one thing that comes to mind is routing out and back to external hardware synths and drum machines, especially if you've got something like a Machinedrum, which has six separate physical outputs...that'll plunder your cable stash in double-quick time. Same with audio effects – you can wire up a hardware reverb, or compressor, and create a convenient recall by using Live's External Audio Instrument and Effect devices, effectively letting you include your hardware in chains of presets. By the way, the OS X Audio MIDI Setup utility is the most important

MTF Step-by-Step Aggregate audio devices (cont'd)



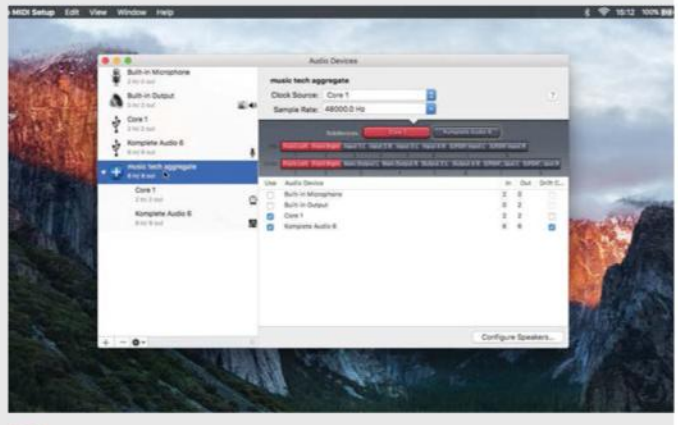
13 Click the solo/cue switch in the Master track, and the solo button in every track becomes a headphone-cue icon. Now you can use these to direct a track's audio to the cue output.



14 You may need to experiment with the In/Out View settings to find the right stereo output, and don't forget there's a cue volume control below the switch, which you could assign to a MIDI controller.



15 Useful, right? Saved yourself from buying a new interface, maybe. As I said, this scales up, so let's try with two audio interfaces. I'm using the Native Instruments Komplete Audio 6 and Alesis Core 1.



16 The procedure is similar, it's just a matter of more connections. Connect the interfaces, and use Audio MIDI Setup to configure those inputs and outputs, adding NI to the existing aggregate device.



17 View it in Live's Preferences – whoo, that's a lot of connections! Enable them all. In theory, having more enabled ins and outs here can increase CPU usage, but I've never had an issue with it.



18 If you own an interface with ADAT connections, you may not need an aggregate interface. I've got a Ferrofish A16 adding 16 inputs to an RME Fireface UFX – it simply appears as one unit.

place to view and troubleshoot your audio and MIDI routings, especially as Live users are so fond of using the IAC Driver to create weird audio setups, like sending MIDI out of Live and back in again.

A better long-term solution for this is to buy a soundcard with more inputs, and if that's still not enough, combine it with a mic pre that connects via optical cables. Naturally, you'll also have to make sure your main interface has optical connections, too. If you take a Focusrite 18i20 for example, and you want to add eight more inputs to it, get a Focusrite OctoPre and connect them via optical. The 18i20 will appear to Live as usual, without any special configuration; it'll just have more inputs. You can buy long optical cables as well, so the two units don't have to be physically together. In my studio, I have a five-metre optical cable connecting a Ferrofish A16, positioned near the hardware synths, to the UFX which is positioned near the computer.

Aggregate devices aren't always the most elegant long-term solutions. But if you have a spare interface lying around and suddenly you need more, it gets the job done. Under Mac OS X, creating and managing aggregate devices is about as easy as it gets, and Live has no problem identifying and using them. Whether you're going to use this on a day-to-day basis or not, it's something to get familiar with in advance of the day when you might need it to get you out of a hole! MTF

Ableton Live Live In Depth – Follow Actions

On the disc
DVD ROM
Accompanying project file included on the DVD

Follow actions and what they can do for you

It's so easy to let software lock everything down nice and tight and be predictable but, says **Martin Delaney**, it doesn't have to be like that…

Follow Actions are settings we can make in Ableton Live's Session View clips, that determine how and when one playing clip will automatically move on to another, and what will happen when it gets there. We can use Follow Actions for creative purposes, such as in evolving sound art installations, or to create unpredictable sequences of instrument parts, but we can also use them as purely practical tools, for example creating timed intervals in a live performance – they're very flexible. Follow Actions can be thought of as Ableton's take on generative music – music which composes or arranges itself in a semi-automatic way. Brian Eno fans will know what I'm talking about. Go back to his *Discreet Music* album from 1975, which would've been made with hardware devices and tape in those days, to the *Generative Music 1* album from 1996, which he made using Sseyo's Koan software. If you want to get some of that old-school Eno action, Koan evolved into Noatikl 3, and it's still available from www.intermorphic.com.

Follow Actions only work in Session View, and only vertically with adjoining clips in the same track. If or when the follow action clips arrive at an empty clip slot, they stop, so that's worth remembering when you're trying to prevent Follow Actions from running on into the 'wrong' areas of your set.

Follow Actions let your clips behave in a genuinely random way, or you can limit their freedom by applying two different Chance values, and combining two different behaviours, selecting from Stop, Play Again, Previous, Next, First, Last, Any, and Other. I guess those names are mostly self-explanatory, although the difference between Any and Other is that with an Any Follow Action there's a chance it will play the current one again immediately, whereas with Other it'll never repeat the current one straight after playing it once. Personally I lean towards Other and the more random selections in general, as that's what I like about them in the first place!

Just like (nearly) everything in Live, Follow Actions can be recorded, in this case, into the Arrangement View timeline. One of my favourite Follow Action moments is to trigger and record a group of clips which have short Follow Actions of 1 or 2 beats, then go to the Arrangement View, select the captured clips, and consolidate them, creating one new clip which contains all of those little parts – this works equally well with MIDI or audio clips. If you prefer to do it all inside Session View, like maybe during a live performance, you could create a new audio track, and use the resampling option to capture either the entire mix back



GOING OUT OF THE BOX

Follow Actions get even better when you integrate them into a larger setup. Why not send notes from those randomly-launching MIDI clips out to connected hardware synths and drum machines? You can randomise your hardware too – just use Live's External Instrument device in each of your Follow Action tracks. Use Program Changes in the clips to load specific hardware presets on-demand, and of course Clip Envelopes to send automated parameter changes. This is all pretty easy to organise, though you might have to check your hardware manuals to get the correct information about MIDI channels, program changes, and control messages.

MTF Step-by-Step Follow Actions



01 We'll be using Follow Actions in Session View to create random and not-so-random behaviour between adjoining clips in the same track. You should use the example Live set provided!



02 Let's use the three adjoining white clips in track 1. Click to select the top clip, then hold down Shift and click on the bottom one, this will highlight all of the clips in that block.



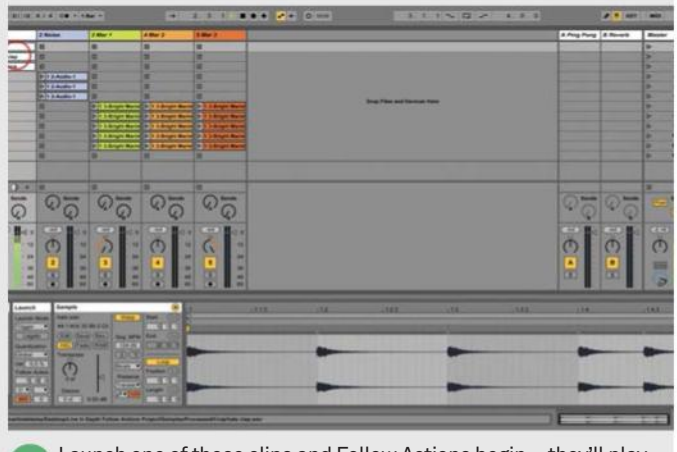
03 If you're not seeing the Clip View, click the Clip View Selector near the bottom right. Also make sure you can see the Launch Box – you might need to click the small L button at bottom left.



04 You can see a section of Follow Action controls in the Launch Box. The top three relate to timing – how long will the clip play before Follow Action occurs, in bars/beats/sixteenths. Leave that for now.



05 The middle boxes let you choose what the Follow Action will be. For now choose 'Other' from the first box and leave the second one. Finally, choose a Chance value – set it to max out at 999!



06 Launch one of those clips and Follow Actions begin – they'll play in a random order, and keep going forever unless you stop them! You could hit record and capture them into the Arrangement View.

into the new track, or else just select the source track with Follow Actions, and record it as incoming 'live' audio. These are great techniques if you're stuck on a project and you need a new sound, a new perspective, or just a bit of spontaneous reorganisation. Plus, of course (unless you choose not to let them), Follow Actions are always synced to the rest of your project.

Follow Actions are really hypnotic and totally addictive – it's as we can use an arpeggiator on Live clips. It can get even more hypnotic, though, if we use more than one track of them at the same time. Of course, two (or more) tracks of randomly launching clips have far more potential for non-repeating patterns. As stated in the tutorial, though, there is also more risk of sounds clashing then, so you can't always just dump two lots of sounds into parallel tracks and expect them to play nicely together – you might have →

MTF Step-by-Step Follow Actions



07 For bonus coolness, consolidate clips in the Arrangement View with Cmd>J, creating a new clip you can paste back into Session View. Follow Actions and Consolidate work with MIDI and audio clips.



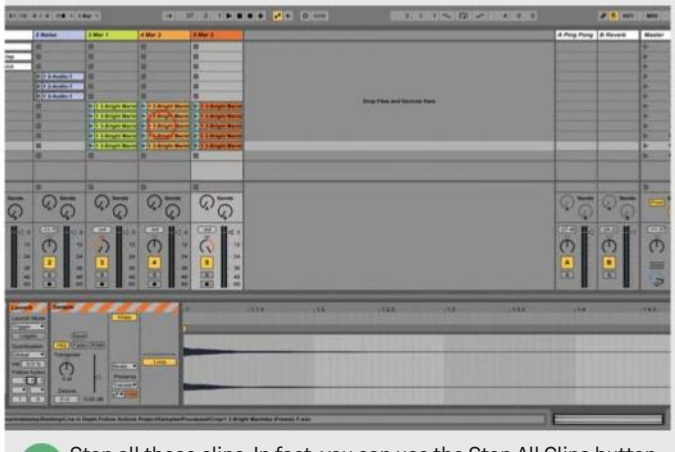
08 More? Let's use the clips in track 2. Make sure all clips in track 1 are off. Select all the clips again, as we did before, and this time choose a time value of 0/2/0.



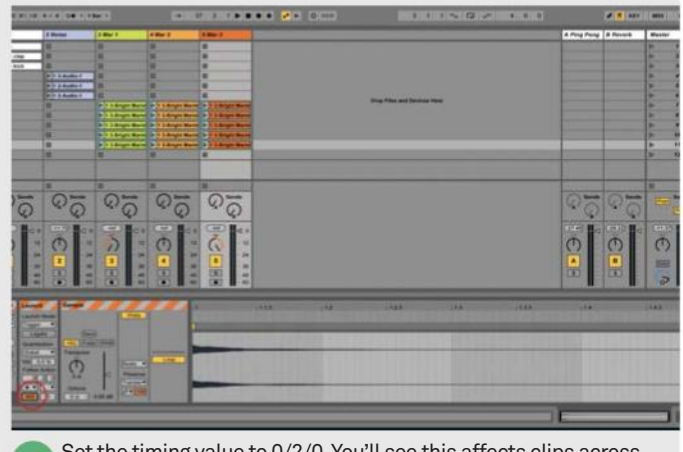
09 Choose the Next action, and set the Chance to 999. Select the last clip alone, and choose the Stop action. Launch the top clip, and they'll play in sequence, until they reach the last one and stop.



10 To make things sound more interesting, send Sends A and B to the 12 o'clock position – those are going to Delay and Reverb. I love adding Ping Pong Delay, especially, to these kinds of sounds.



11 Stop all those clips. In fact, you can use the Stop All Clips button under the master track. Now go to tracks 3, 4, and 5. Select all of those clips and go to the Launch Box.



12 Set the timing value to 0/2/0. You'll see this affects clips across all three tracks. Choose the Any action in the left hand box, and set Chance to 999 again.

→ to tinker with the timings, and adjust parameters and EQ to get a nice fit – basic mixing in other words.

You can blur the lines between organisation and chaos by adjusting the chance settings for each action - an action assigned to a value of '0' will never happen, while identical chance values will give an equal likelihood of occurrence. But going further, you can dictate specific steps and create regular, predictable, outcomes, so you can have a set of instrument parts that play and move on of their accord while you sing or play along, like a little interactive backing track. You want a clip to stop playing after 8 bars? Easy. You can always add empty MIDI clips, or silenced audio clips, to insert timed pauses or breaks into the flow as well. Taken to extremes, you could have just one track that plays

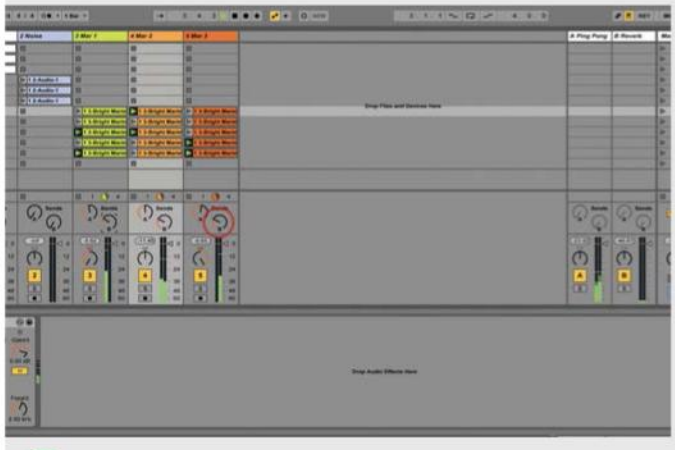
MTF Step-by-Step Follow Actions



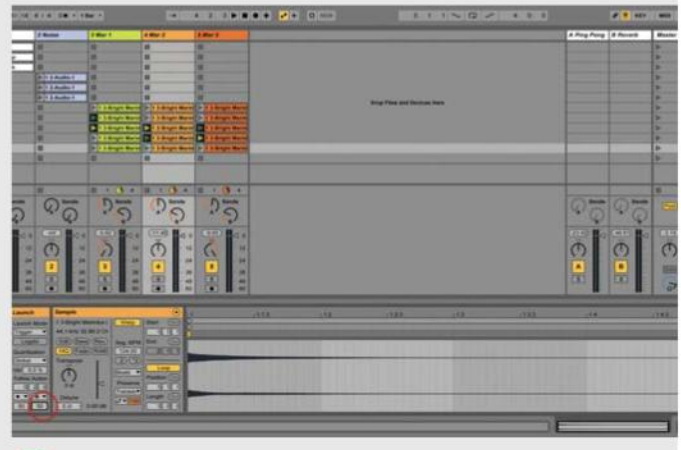
13 Instead of launching each track individually, use a scene from the Master track to fire the top clip in each track. Clip playback diverges, building a more interesting and unpredictable sequence.



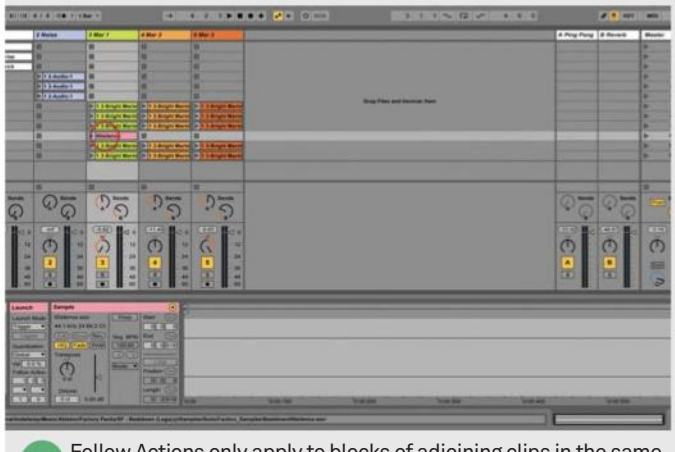
14 Sounds good, but track 4 is clashing with the others in the low end. So add an EQ Three to track 4 and roll off the low end a bit. That thins it out some more.



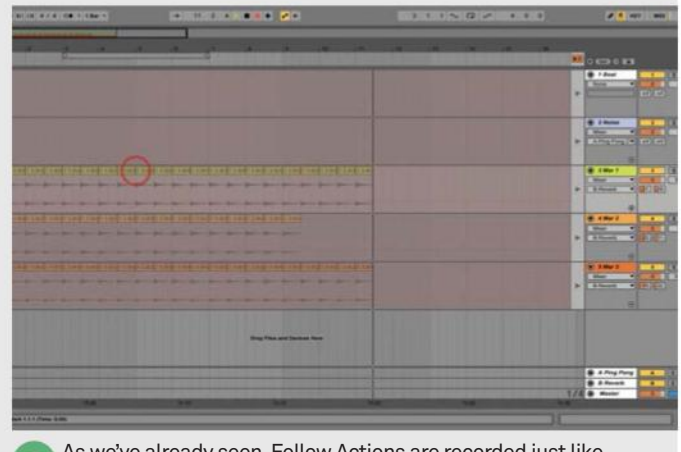
15 Having fixed that EQ issue, now you once again add delay and reverb sends to give more depth and even more rhythmic variety to the parts. Experiment with different project tempos as well.



16 Go to the bottom clip in track 4 and set one Follow Action to Stop, and the other to Any and their Chances to 50 each. Now, when the Follow Action reaches that one, the track 4 clips sometimes stop.



17 Follow Actions only apply to blocks of adjoining clips in the same track. They stop when they arrive at an empty clip slot. If you need a silence section, add a silent audio or MIDI clip!



18 As we've already seen, Follow Actions are recorded just like everything else that happens in Live, so it's easy to use them as a way of semi-randomly generating new parts and even mixes.

through an entire song structure following just one click to launch the first scene – especially if you combine with instruments racks, Simpler and the Chain Select Envelope so you can have different clips loading different instruments, audio loops (that's why I mentioned Simpler), and audio effects! It's as always with Live: the individual features are really cool, but once you get the combinations going, it comes truly alive.

You might have noticed that I'm quite partial to Follow Actions. It's true, I'm Fanboy, and not ashamed to admit it. Follow Actions are built into every version of Live, and you don't need any extra hardware or plug-ins, just some audio or MIDI clips to play with. No matter how you use Live, there's room for Follow Actions in your work flow. MTF

Ableton Live In depth

On the disc
DVD ROM
Accompanying project file included on the DVD or at www.musictech.net

Creative clip management with Push 2

Ableton's Push hardware controllers – especially Push 2 – interact with Live's Session View clips in ways that no other controller can achieve. **Martin Delaney** explains how to get creative…

It's not often we run a tutorial that's dedicated to one controller but, of course, Push isn't just any piece of hardware – it's built by Ableton itself to complement its Live software, which it does, with unmatched depth. This means that there are enough Push users and Push-curious Live users out there to justify this kind of workshop coverage. Push is primarily focused on Session View which is, of course, all about clips and we're focusing on those here. But let's not claim that this is 'everything you wanted to know about Push and clips' – that would be kind of dangerous, especially given Ableton's ongoing habit of adding new features.

As Ableton itself stated in the past: once you create your clips with Push, and browse and load instrument and effect devices, then record a little jam or performance, maybe you'll be better off putting Push to one side and working more directly with your computer. What Push does is provide a more dynamic, hands-on, 'instrument' vibe for that brainstorming creative process; that physical part of it is based around the controls – the encoders, buttons, and the control strip – and maybe most importantly, the dynamics and excitement that come from the way Push lets us get hands-on with audio and MIDI clips.

It used to be that audio clips were kind of a poor relation on Push in some ways, but not any more, as with Push 2 we can view waveforms for samples contained within Simpler and also for regular audio clips. This is a major advance in terms of getting a full 'DAW' experience from the hardware.

Our example Live set contains a selection of audio and MIDI clips for you to tinker with. You'll need Live 9.6 or later to open this set as we need to be right up to date to make the most of Push. Any necessary Push firmware updates are included within Live software updates – the Push display will let you know what's going on!

In Session Mode, Push does a good job of reflecting the contents of your Live set's Session View and you'll get clip positions and colours clearly laid out, ready to launch with a single tap. The colours should be reasonably close to what you see on the computer screen. Get familiar with what's in the example set by tapping any of the clips to start them playing. If Live isn't already running, launching a clip will start the transport. If Live is already running, the pad will flash and wait until the start of the next bar before playback begins. (If you ever want to play loops in Live without starting the transport, you can do it from the Looper and Simpler devices.)

In the walkthrough we see how launching clips from the Push pads will reflect launch mode settings you've made. Live's default launch mode is Trigger mode, which is fine



GOING OUT OF THE BOX

Push isn't the only hardware solution for Live. You can use MIDI keyboards, apps, or other pad-based controllers from people like Akai and Novation. Push is the ideal all-rounder, though, and many Live newbies buy it on day one as it's the obvious choice, and there's nothing wrong with that. However you might prefer another controller because of price, portability, or some unique feature that it has. I reckon the biggest competition is Touchable on an iPad Pro – the iPad is the only Live controller where you can see the clip names and colours on the control surface.

MTF Step-by-Step Using Push for clip management



01 This is one of our rare Push-only tutorials, so you'll need a Push to make the most of it (ideally Push 2). Load our example Live set, which has some clips ready for you.



02 Tap the Session button, so the Push pads are ready to launch the clips contained within the focus box displayed on your computer screen. You'll see the clip colours are the same as on-screen.



03 Launch some of the clips in the set – it doesn't matter which for now. They are exhibiting the default Live/Push launch behaviour. They'll flash before playback starts and flash while they're playing.



04 That's because they're following Live's global quantise settings, which you'll have to change from the software as Push can't do it – not yet anyway (you can change record quantisation from Push).



05 To stop a clip, press a pad to activate an empty clip slot in the same track or use the Clip Stop button. You could also use Stop transport or (on Push) Stop All Clips with Shift then Stop.



06 Live clips can be assigned different launch modes – behaviours such as Toggle and Repeat – and Push recognises these. Toggle's very useful because a second tap of the same pad will stop the clip.

most of the time, but it does require you to tap elsewhere or use the Stop button when you want the clip to stop but Live's transport to continue. If the clip is set to Toggle, it'll simply stop when you tap it again, which is generally a much smoother behaviour when you're using hardware with Live clips. You can even change a clip's launch mode from Push, which is great for mid-gig changes of direction. In Session Mode, there's no way for Push's pads to send velocity, so you can't use them to change how loud or quiet a clip plays upon launch. There are two ways to deal with this: one is to use another controller instead – anything with a MIDI keyboard or drum pads will work, although of course there are significant trade-offs in other areas. The other is to use Simpler, which has become arguably the most important device in Live. Not only is it a great tool for →

MTF Step-by-Step Using Push for clip management (cont'd)



07 Although audio and MIDI clips are different to work with in some ways, when it comes to launching them from Push, they're the same. But that's just like with Live itself, so you knew that already!



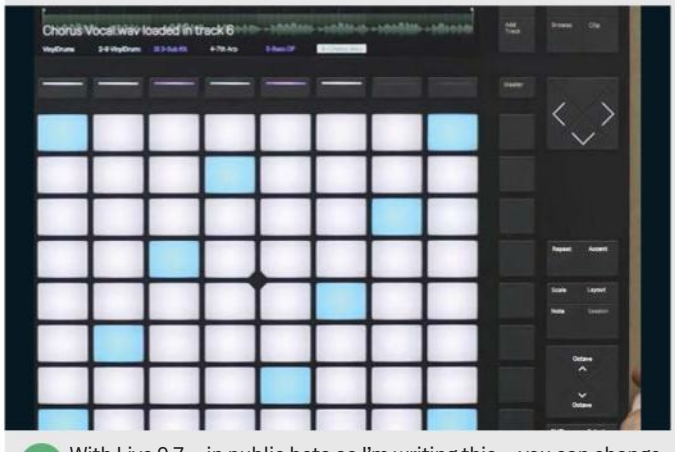
08 Push doesn't offer a relationship between pad velocity and clip volume, something that some hardware can do. The work around is to load samples into Simpler; then you can apply velocity to it.



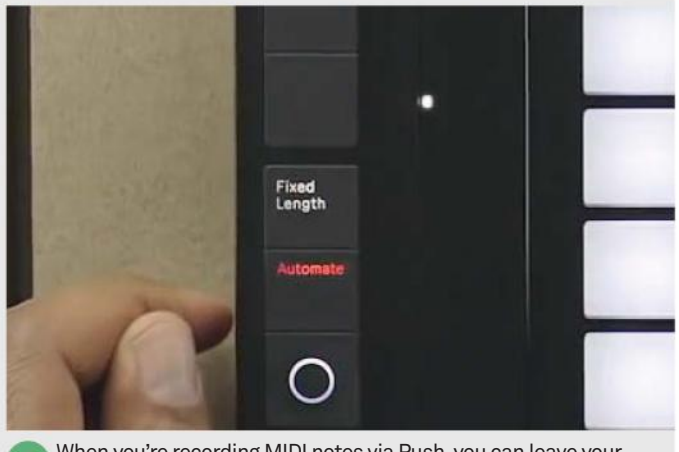
09 You can select a clip without launching it by holding down Select as you tap on it. This also works with other things such as tracks and pads in drum racks, very useful, and no noise!



10 When you're viewing the clip details, including waveform for audio clips, you can change the clip's loop length, start position, gain, and Warp modes (that'll be audio only, of course).



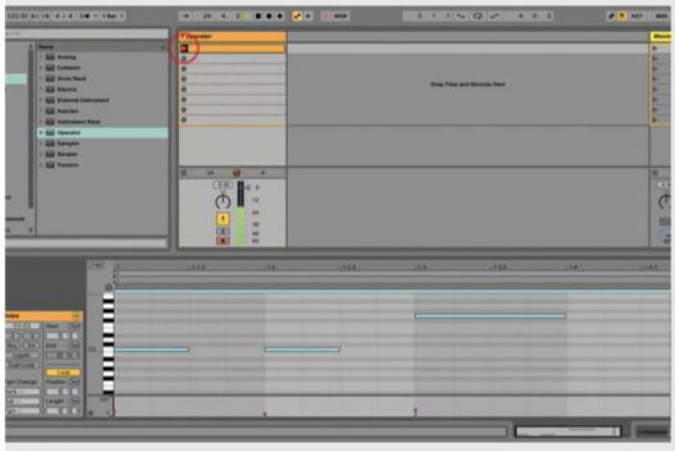
11 With Live 9.7 – in public beta as I'm writing this – you can change clip colours from Push. Hold Select and press the clip, then press and hold Shift and select a colour from the grid.



12 When you're recording MIDI notes via Push, you can leave your clip lengths open ended, so they just keep on going as you play, or use the Fixed Length button to set a bar value.

→ hosting one-shot samples or building sample-based instrument patches, it's a fantastic loop host as well. Drop an audio clip in there and you can immediately trigger that loop from the pads on Push and dial in as much velocity response as you desire. Simpler works brilliantly well with Push 2, especially as you can view the waveforms, and it could well be that you build entire Live sets where all of the audio material is contained within Simpler – working like this opens up a lot of potential for building instrument racks containing audio loops, and building your tunes in a whole new way. Personally I find I'm doing this more often. And while I'm on it,don't forget to click on Simpler's little 'expand' button when you want to get a really large scale view of

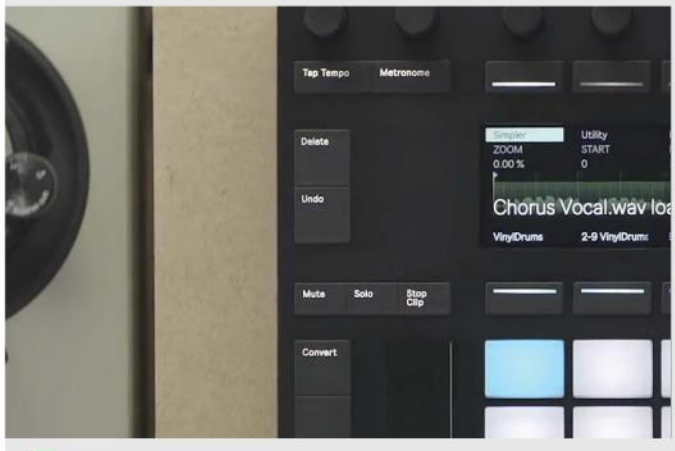
MTF Step-by-Step Using Push for clip management (cont'd)



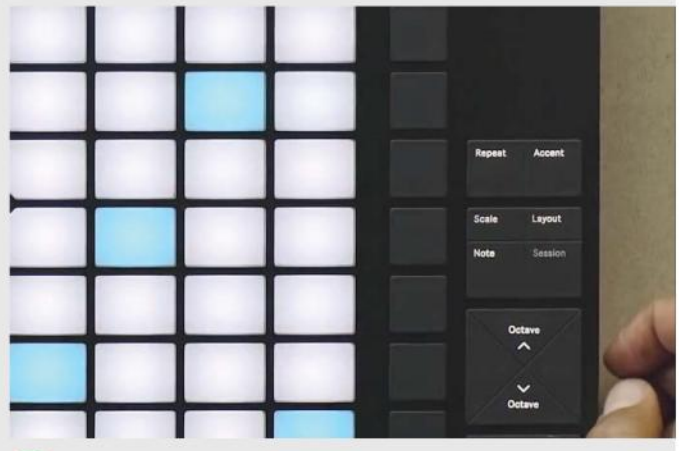
13 Once the recording reaches the limit of that bar value, the clip goes straight into playback, without interruption. This works with audio clips as well, which is great for live loopers and beatboxers.



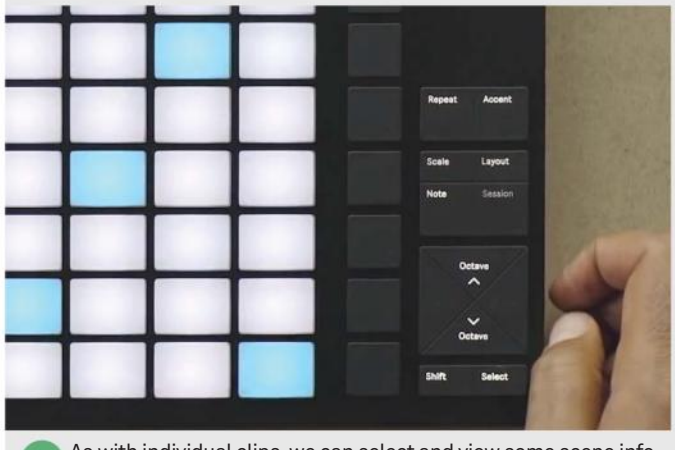
14 It's possible to copy and paste clips from Push. All you have to do is press and hold Duplicate and tap the pad you want to copy, then tap where you want it to go.



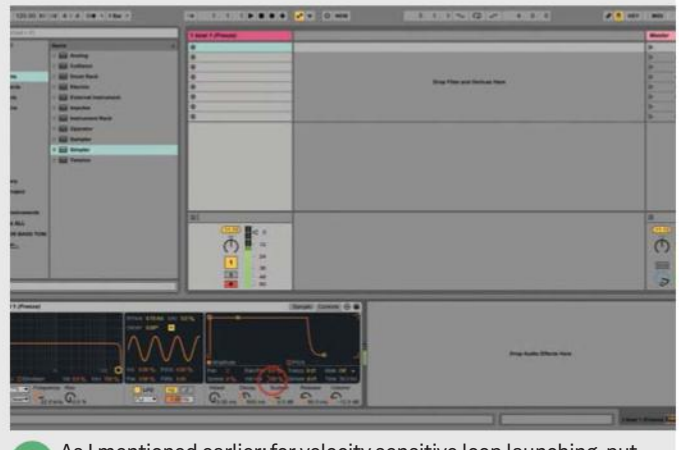
15 Likewise you can delete clips by pressing the Delete button, and then the clip you want to dump. Use Undo to bring it back if necessary! Delete also works with tracks, scenes, and devices.



16 We're allowed to mention scenes, because they're rows of clips! Launch them by tapping the Note Value buttons at the right of the pad grid. Press the New button to create a new scene.



17 As with individual clips, we can select and view some scene info without launching it. Press Select and then the Note Value button corresponding to the desired scene, you'll see the scene name.



18 As I mentioned earlier: for velocity sensitive loop launching, put your sample into Simpler and trigger it from the Notes Mode. You could build a set using Simpler for content instead of audio clips!

what's going on.

Simpler fans will love the Convert button on Push 2. This has different functions depending on what mode Simpler is in, but it is all about creating new MIDI tracks based on the currently selected clip. If you have an audio clip selected, it will build a new Simpler track containing that clip, which is a nice fast way to get that task done.

There's no such thing as the perfect Live controller as we all use Live in different ways, but what we're enjoying here are the things that Push does better than other hardware controllers (if they do them at all) – the features that best make use of Ableton's ability to develop hardware and software side-by-side. MTF

MTF Feature | Produce a track from start to finish

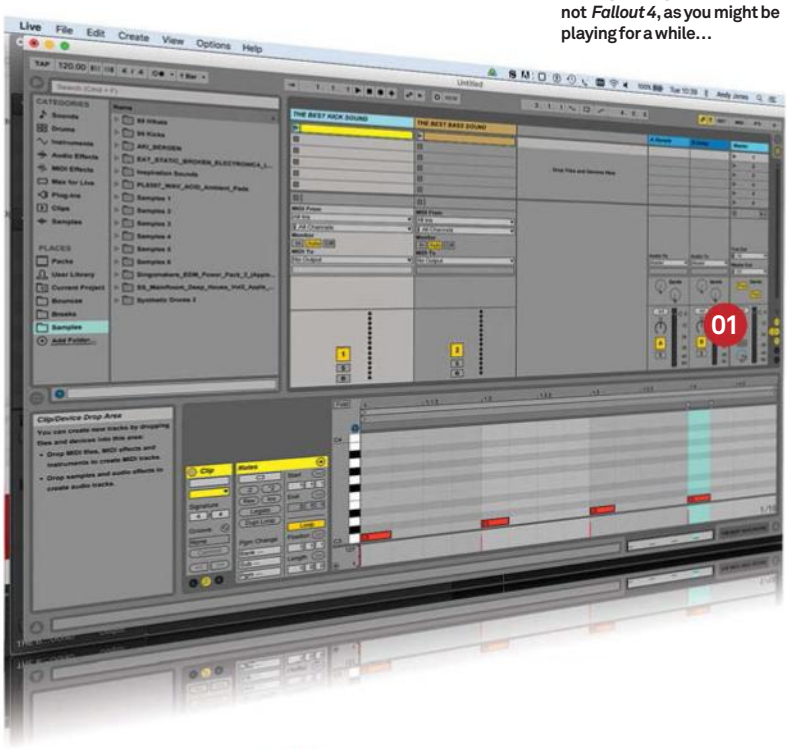
10 WAYS TO MIX A TRACK

Part 3

It's our 'produce a track special' – and it's time to mix your tune. Here are some pro ideas to take your mix up a level and let your musical ideas shine through…

01 Keep it simple

With technology comes options – many, many options. You can layer many audio tracks together, plus a whole bunch of soft synths and instruments. But do you *really* want to do that? Wouldn't your tune work better with one great sound, rather than 10 mediocre ones? Do classic pieces of popular music all rely on massive mixes? Okay, some do, but more memorable songs rely on a great hook or a well-produced individual sound, rather than walls of many sounds. With dance music, especially, if you get a great kick or bass sound, you are 90-per-cent there, and the rest is all about the arrangement. Spend time on getting your core sounds right and the rest will follow… But not too much time, mind…



1. Keep it simple, with just one or two great sounds which could make your track

2. Yes, we are recommending playing video games. Only maybe not *Fallout 4*, as you might be playing for a while…



02 Take a break

As cool as spending the whole night mixing sounds or how good a day's eight-hour session at the monitors feels, the truth is that the longer you spend listening to one mix, the more negatives will come from the experience than positives. Firstly, your ears will get tired and then so will your brain. You'll start edging volume levels up across the board until you are mixing at excessive levels and your ears will soon start ringing. Your brain will start to turn to glue and your decision making will suffer; you'll soon go from making vital and essential tweaks to undoing all the good work you put in at the start of your sessions. So, stop! Hold on a minute! While distractions can be regarded as your enemy at the start of the production and compositional process, they might well be your friends here. A good video game or pool table might be just the accessory your studio needs. Just make sure you return to your mix at some point, though…



03 Loud. Loud. Loud

While we're talking volume, you might find that you end up with the kind of 'mix creep' mentioned earlier, especially over longer sessions; but you should always try and reduce volume levels, no matter how long your sessions are. You can give your mix the odd big blast just to make sure it kicks at high levels, but always bring it back down for the main mixing session. Remember that a really well-engineered mix will sound good at any level.

04 Fill it up, but not right to the top

One of our favourite mix analogies is to consider your mix not only as a stereo left-to-right picture, but also as one going up and down. By this, we mean that when panning, think about filling the space 'left to right', and if it's a band you're mixing, consider using the positions of the band members as they would play on a stage as references for your main positions within the mix. And when EQ-ing, you could do worse than thinking 'top to bottom'. Your low-end frequencies are at ground level, and your highs are, well, at the top. In this way, you can use judicious panning and EQ to 'fill up' the entire picture. But having said that, don't fill your picture up completely…

3. Don't let those levels get too loud. Here they are in Live going into the red…

4. One simple way to mix is to consider your mix in the positions that a band traditionally takes on stage, and pan the instruments accordingly

5. If you have a vocalist like Adele in your song, then the last thing you want to do is crowd her out of the track

6. Use a compressor in your DAW mix for extra clout and power







05 Breathe more easily

Use your mixing skills to allow the most important parts of your mix to breathe. If you happen to have Adele in your studio, the last thing you want to do is mask her voice with a bunch of things that sit in the EQ range or pan position of her vocal part. Move everything away from the most important part of your mix and give it room. If necessary, don't shy away from using EQ reduction to help take other mix elements away from the most outstanding part of your tune.

06 Compress and limit

Getting your levels right is not just about adjusting faders per channel. More subtle adjustments can be made with compression, while less subtle ones can be made with limiting. In general, the entire mix is compressed and limited at the mastering stage, so don't use this method so much in mixing. However, do consider applying compression to individual parts, so that your bass can pump and your snares can have more bite. Limiting helps cut out stray levels across the board, too.

07 Use references

As well as using several different reference monitors to check your mix – see the boxout – you should remember to use musical reference material to check your mix against, too. This could be as simple as loading in a similar type of music to the one you're mixing as a reference point. You aren't so much checking the volume here, as this will be done in the mastering process; it's more the style and the mix. You can get a great idea on EQ types and levels here, too. How does their bass cut through the mix compared to yours? Does your kick have as much clout? It's a great way to learn from the pros, by just hearing their efforts side-by-side with yours. Beware, this can be a little depressing at first, but do bear in mind much of the extra shine comes from mastering, so don't get too down.

08 Effects

With effects, as with virtual instruments and synths, technology is spoiling us rotten. Back in the day (oh, →





don't you just hate it when old timers say that?), you might only have had a couple of effects to use, so you had to make do with just those and nothing more… and this was not necessarily 'a bad thing'. Today, it's tempting to smear effects across everything simply because you can, but the technology message is, once again, that it's great to have it on tap, but use it sparingly. So don't, for example, slather reverb on everything, or your mix will sound like it's recorded in a cathedral, which could be a bad thing.*

*** Unless you want it to sound like it's recorded in a cathedral, in which case this could be a good thing.**

09 Groups

Grouping is a powerful feature. Your DAW should allow you to do this, so learn how to do it now! Grouping can save a lot of time and a lot of effort as, essentially, groups allow you to process several (usually similar) mix elements at the same time. Your vocal harmonies, for example, might all use similar effects, so group them – as might your percussive elements. Then you can automate, mix, and process multiple tracks to your heart's content… MTF

7. (Above) Juxtapose some reference material with your mix so you can compare your efforts with the best music out there

8. Just because effects are available, it doesn't mean you should automatically use them



9. Grouping tracks together can save time, and allow changes to be made quicker. It's also a quick, non-destructive way to apply experimental changes to whole portions of a mix

10. You need to make your mixes sound good on any device, including mp3 players.



10 Best mixing tip ever?

And the most popular mixing tip award goes to… Listen to your mix on as many different systems as possible. And this top tip still holds true, today more than ever. People consume music on a vast array of different devices: from minute players and phones right up to high-quality recording studios, so you have to make sure your mixes sound as good as they can on as many of these as possible. This means that as well as sounding great within the confines of your own studio and on your expensive monitors, they should also sound good through the headphones of your phone, compressed down to mp3; they should also sound great on the CD player in your car; and they should also sound great on your mate's hi-fi. In the olden days, people even made them sound great in mono for radio play – it's not so vital now, but if you can, then why not?



Oliver Nelson

With a studio setup just like yours and ours, Oliver Nelson can not only offer production advice on his successful nu disco productions, but also has a lesson for all of us on how to make it in music production in the 21st century. **MusicTech** sits down with the man who has mastered the media…

The phrase 'up and coming' is often thrown lazily at any artist releasing their first recording on a major label. But while Oliver Nelson is most definitely doing the latter – his first release on Polydor, *Found Your Love*, should be out as you read this – he is not exactly new to music production, having amassed a huge social-media following with a stream of remixes and productions, way before the single came out.

Indeed, the 25-year-old Swedish producer is giving us all a lesson in how to carve a career in music making today. Before the single came out, Nelson had notched up millions of plays across various platforms. *Found Your Love* had scored over 1.5 million plays on SoundCloud, while his reworkings of tracks by Janet Jackson, The Knife, Marina And The Diamonds and Whitney Houston among others have scored nearly 100 million plays on SoundCloud and YouTube: his remix of Justin Timberlake's *Suit & Tie* garnered 16 million alone. So how has this young producer already made a name for himself before his first release? Well, let's start at the beginning…

MusicTech: How did you get into music production?
Oliver Nelson: My brother started playing drums when I was a kid. He was both my role model and inspiration, so I jumped on the drums myself and found my love for creating beats and rhythms. My parents were heavily interested in music, so in turn you could say my whole family were, and are, a huge influence. My first major step I took myself was deciding to attended a music high-school and that led me to start making my own music.

MT: From there, then, how did you break through to become successful?



ON: I met my manager and we formed a great personal and professional relationship. He helped me find remixes and I worked tirelessly to make them as perfect as I could, and then they kept coming and raised my profile. He worked for Sony for 12 years, so it's a perfect combination of our very contrasting talents. He provided the advice I needed to help me make the transition from amateur to professional. Hard work, determination and passion are my personal key drivers.

MT: Tell us about your first production setup…
ON: My good friend Jakob introduced me to my first music-software program when I was 17. It was fresh and fun; we just played around and my interest grew from there. I was amazed by the fact that making music digitally has infinite possibilities. I know electronic music production has been around for a while, but I've embraced modern techniques that come with improvements in technology and software. You have to move fast in this dynamic, competitive and rapidly paced environment.

MT: So it was really the possibilities that technology offered that drew you into production?
ON: Yes, absolutely! I couldn't do what I do without today's technology. I don't see any problem with it, I'm just happy there's an easier way today for a creative mind to do what he or she wants to do in terms of music.

Computers have really revolutionised modern music making. I do everything in front of my computer with some outboard analogue gear, but even so, I control those with plug-ins. I've always been interested in computers, so it suits me perfectly. It's also much more effective and more advanced than working with, let's say, a mixing desk and analogue recorders, even though that equipment has its own charm.

MT: How does it feel to have this much power inside a computer?
ON: When you think about it, it feels quite strange, actually. I think it's amazing what you can do on a computer nowadays. The fact that you can do what you love and make a living out of it is really cool, especially when you don't need that much. A computer, a MIDI keyboard… and you can be set for life. That's the modern age, but you need passion.

> "I am a **massive fan of Ableton Live** and I think we could well be **pals for a long time**"

The Oliver Nelson Studio

MT: So which software did you use back then compared to now?
ON: I used Reason in the beginning, but then went over to Ableton and became a massive fan. I think that as they are so dynamic and always updating, we could be pals for a long time.

MT: What else do you have in your setup?
ON: I'm working on my PC desktop in my home studio. My main synth is my Prophet 12 that goes along my other ones like the Moog Sub 37, Roland D-50, Yamaha Reface CP, Korg MS2000, Roland JV-1080, Elektron Analog Rytm, Nord Lead 2X and more, all going through my RME Fireface 802.

OLIVER'S PLUG-INS

Oliver details his favourite software instruments and effects...

PLUG-IN SYNTHS

I have four favourite 'go to' plug-in synths:



• LENNARDIGITAL SYLENTH1: The first software synth I used and I was totally blown away by how amazing it sounded. The fact is that I have never really found any other one that could match that sound quality; maybe DIVA, but for me, Sylenth1 takes the money. It also has a sentimental value for me, as it's one of my first and I use it in almost every production.



• TAL U-NO-LX: I'm a huge fan of the Roland Juno 60 and TAL Uno does a great job simulating its sound – I don't need to say any more than that.



• AAS CHROMAPHONE: I love my marimba and this is an absolutely amazing plug-in to go to for those kind of sounds. Anything plucky, realistic sounding, it's your best friend. Probably my number-one favourite.



• U-HE DIVA: This is the best soft synth for an analogue sound. It has a very rich character and amazing presets and expansion packs. It always works in any kind of production, whatever genre it might be.

PLUG-IN EFFECTS



My absolute favourites are the Soundtoys plug-ins. They are just all-around amazing-sounding effects. The Decapitator is the best saturation plug-in out there in my opinion, along with PrimalTap, which I use a lot on leads to make them more interesting. MicroShift is also a plug-in that I always use on vocals.



I also use FabFilter's Pro-Q 2, a really good EQ plug-in that has a really simple interface. For reverb, I always use Valhalla's plug-ins, especially its VintageVerb, which sounds incredible. I use it on everything from synths to percussion. It has a very warm, round sound that makes every difference.



Then there's D16 Group's Decimort. Best bitcrusher out there! I use it to make things sound more analogue, and there's a ton of ways you can use it. It works very well with vocal leads and guitars.



MT: That's a great list of hardware – both old and new – but you also use a lot of software [see boxout]. What do you like about each method?

ON: It's the simplicity with software! To be able to start your DAW and start using your favourite synth in seconds is a dream – there's always some hassle with using analogue synths, but it goes both ways, I guess. You also get the unreliability with software, not knowing when things are going to crash, and even if you save every five minutes, you never know if every little tweak is going to be the same as where you left it last time after a restart. It's happened a lot that one soft synth has a totally different preset after a restart, and hours of work is lost. That never happens when I use my external synths.

MT: Is there anything on your studio shopping list?

ON: On top of the wishlist right now is a new laptop. I'm working on my bad-boy desktop, but now when I'm touring all the time and meeting other artists and vocalists for collaborations, my desktop is useless. I do have a good laptop, but I don't find it as powerful as I need it to be. So I guess one of the things on my wishlist is the new MacBook Pro. Must work harder (laughs)!

SELECTED KIT LIST

Ampsandsound SE-84
Audeze EL-8
Audeze LCD XC
Audio Technica ATH-50x and M70x
DSI Prophet 12
Elektron Analog Rytm
Korg MS2000R
Moog Sub 37
Nord Lead 2x
RME Fireface 802
Roland D-550
Roland JV-1080 and 880
Yamaha Reface CP
...plus girlfriend, to make me have breaks

MT: What would you like to see developed in production technology?

ON: My top wish right now is a good software I can use for my Analog Rytm to drag-and-drop drum samples to it and use directly within Ableton. At the moment, it's just possible to transfer samples to it outside of the DAW, but the problem is that I have so many samples I want to use, and different samples in different projects.

MT: What about playing live? Do you take any of this tech out on the road?
ON: I've always admired people using computers live, like Madeon, playing his live sets with a controller. I've not come to that point yet and I don't know if I ever will, but I do respect people doing it.

MT: But you have dabbled in DJing…
ON: I started out with using Traktor, but to be honest, I had too many problems using it. It can be a pain with an external soundcard, all the cables etc. Especially if you are going up directly after another DJ and want to make a smooth transition between the sets. I now DJ with USB sticks; it seems to be the safest and handiest way of doing it. It is easy to travel with and using it with rekordbox is just really easy and fun.

OLIVER NELSON ADVICE

MusicTech: What advice do you have for people wanting to get into remixing/production?
Oliver Nelson: Just find software you feel comfortable with. Most programs do the same things, just in different ways and with varied designs. A very important thing is to learn stuff by yourself, and be very patient. It takes many years to learn and perfect the craft. I'm still learning new things every time I work on a new track. That is also why I love to work with electronic music, there's always something new to learn and the techniques are always improving and going forward. You need to move at the same pace and roll with it.

MT: What about advice for making a living in music production?
ON: The music industry has changed massively the last 10 years. Artists have gone from making money from selling records to earning money from streaming and having big followings on social media, etc. I believe that music should be accessible to all people, despite what their income allows them to do, but the royalties need to be fair. A lot of people all over the world might not be able to afford to subscribe to Spotify or Apple Music. There's loads of ways of making money as a musician; it might be harder making the big bucks today, as the main income will be selling out arenas. Although, as a smaller artist like me I can still make a living out of what I do, even though I'm mostly getting traction on SoundCloud, which doesn't give me any direct income, but rumour has it, that's about to change…

So just keep doing it because you love it, work hard and it will come. I didn't start producing music because I planned on making a living out of it, I just made it because it was a big passion for me. The rest is just a bonus. It takes time and you have to believe in your own talent and it will happen, sooner or later. That way of thinking can be applied to all kinds of creative work…

MT: What about using social media to help your profile – any tips?
ON: I have been a user of SoundCloud for a long time and it got me where I am today. I think it's in anyone's right to upload and market their own music, if it's not bootlegs. I'm fascinated by the way the industry has changed from consumers buying CDs and downloads to streaming music online.

MT: Finally, with such success in the area, what advice would you give to use social media to get a following and raise your profile?
ON: The foundation of any success in this industry at its core is the music, so initially, I didn't take social media into consideration as much as I do now. The plays I have gathered online have played a very big part in my career. Justin Timberlake's *Suit & Tie* and Ella Henderson's *Ghost* gave me a lot of exposure via social media and YouTube, and have opened many doors. As I've grown in popularity, it's become really important to engage with my fans and to keep them up to date.

Above all else, it has to be about the music! Focus hard on producing your best work with every track. If your music is good, then it makes creating and growing a social-media presence much easier. People will naturally want to share things with others that make them feel good, or that they genuinely enjoy. It's also important to have a solid identity and to reflect this in your brand, think about what defines you as an artist and what makes you different and promote that. Last but not least, it's crucial that you try your best to stay connected to your fans, the people that support you. Keep them updated, share pictures and updates with them, reply to comments and be yourself, be relatable.

Oliver Nelson Productions

MT: What would you say inspires a typical Oliver Nelson track?
ON: Everything I'm into at the moment, usually the tracks that stick with me scrolling through Spotify. Also, everyday things that happen and feelings affect the outcome of the music I'm making.

MT: How does a track usually come together?
ON: I usually start with chords, with a simple piano or synth around a vocal, and lay down some simple drums so you get the hang of the composition. It's like a basic skeleton that you then add things to and build, from the foundation up. The most important basic thing with a track is the chords – but with dance music, the drums obviously play an essential part as well.
MT: What do you think makes your sound and what production tricks do you use to get it?
ON: I think what has made my sound is the drums, in combination with a certain type of vibe in the chords. I tend to spend a lot of time on said drums



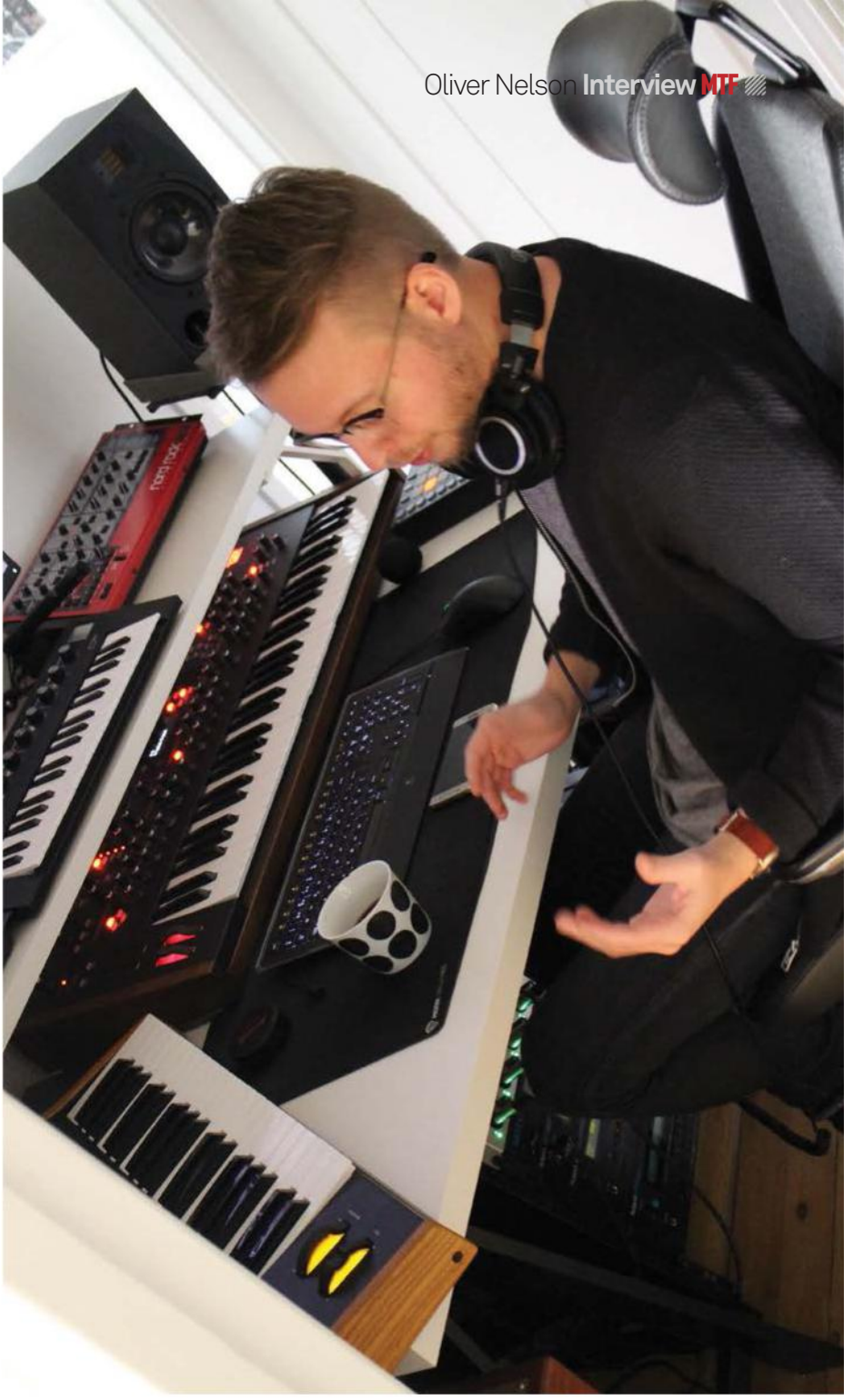
(being an old drummer myself), and can sometimes find myself sitting for hours working on small details like drum fills etc, (laughs). I'm also a big fan of a nice rhythmic groove in the chords, that's just what I like and it's always in the back of my head when I write something. I also am a perfectionist when it comes to every sound, my manager loves this about me, but also thinks I can keep going on some things just that little bit longer, when I don't need to.

MT: Tell us about the new single...
ON: *Found Your Love* features vocals from Heir aka Skogsrå. It wasn't really meant to be my first single initially, it was just some chords that I made jamming on my piano, but everyone seemed to love the chords, so I just kept working on it. The track is a funky nu disco tune with a fat beat!

My next project is a single with my good friend, and Perfect Havoc management colleague, Tobtok – which I'm working on as we speak. It's a solid mix between our unique styles and has a great pop/dance sensibility. It's a more uptempo track than my usual stuff, and It's going to rock. This year, you guys are gonna see a new side of me, with more experimental and varied productions.

MT: Is there anyone else you'd like to work with?
ON: I've been a big fan of Gallant's *Weight In Gold* which, in my opinion, was one of 2015's best tunes. I also love Honne and am a big fan of Dusky & Rüfüs' latest material.





"It **takes time** and you have to **believe in your own talent** and **it will happen**, sooner or later"

MT: Finally, then, what musical plans do you have for the future?
ON: I'm really excited about touring regularly, but it's basically a balance between that and recording. My plan is just to keep making as much music as possible and do what I love. I'm focusing on original music right now, more than remixes, and can't wait to show you what I have been doing.

Oliver Nelson Ft. Heir's *Found Your Love* is out now on Polydor Records. Visit Oliver's SoundCloud page at: soundcloud.com/olivernelson MTF

MTF Feature Spend, spend, spend!



6 PIECES OF EQUIPMENT TO BLOW A YEAR'S SALARY ON

What if money was no object and we could indulge our wildest gear-hoarding fantasies and buy whatever we wanted? **Rob Boffard** has the answer...

When it comes to buying gear, money isn't actually the primary consideration – or it shouldn't be, anyway. Buying studio kit is all about your needs, what it can offer you and how it can affect your mixes. Not that that stops us from jumping on kit websites and filtering by price, seeing just what the top end really looks like. What would we do, if money was absolutely no object? Which pieces of gear would we splurge on? Fortunately, we've saved you the trouble. Here are six pieces of hardware to spend a year's salary on – and by a year's salary, we mean a truly staggering amount of money (heads up: we've largely steered clear of vintage equipment here, as prices can often fluctuate – and we could spend a long, long time rounding up the best and most expensive vintage gear out there...).

The Slam! is terrific. It offers **superb dynamics**, lightning-fast FET and **pin-point control**





01 Avalon AD2077 Stereo Mastering EQ

Amusingly, Avalon includes a testimonial for the AD2077 on its website from a person named "Emily L, NY", who raves about the EQ's "incredible depth and elegance", "the integrity of the stereo image" and the fact that "the high-end air seems to go on forever!" We don't know whether "Emily L" exists, but unless she's got the dollar equivalent of £10,000 in the bank, we're betting she doesn't own one of these. The AD2077 is a truly marvellous bit of kit, with fantastically-made Class A amplifiers that deliver all the sound quality you'd expect. Emily L may or may not be real, but you can definitely believe her hype.

02 Manley Slam!

It's worth gracing our list with this outstanding

piece of outboard: a mic preamp/limiter combo. A very expensive one, built by a legendary manufacturer and guaranteed to give both your recordings and your studio a sheen that only lots and lots of money can buy. And, to be fair, the Slam! is terrific. It offers superb dynamics, lightning-fast FET, and pin-point control – and despite the fact that it can get bloody loud, bloody fast, the results are always butter-smooth (as you'd expect from something with a price tag of £5,000).

03 Shadow Hills Mastering Compressor

Compressor shopping is the dark underworld of studio kit. It's a realm filled with rumours, strange characters and mythical creations. The Shadow Hills Mastering Compressor is – and we use this term very deliberately – the holy grail. Even finding it on the Shadow Hills website takes time and effort, and… well, just look at the thing. Big, foreboding and very, very black, with a set of components that deliver truly breathtaking compression. No wonder it's sought-after. It's separated into discrete and optical sections, includes full sidechain filtering and is generally considered to impart shine to mixes just by being in the same studio. It also costs around £7,000 – assuming you can find one to buy in the first place.

04 Adam S6X Active Studio Monitor

When it comes to monitors, you want monsters. So why not put down £10,000 for the Adam S6X? It's a four-way monitor with front-mounted controls that is ridiculously, sublimely accurate, even at very high sound pressure levels. Each of the drivers has separate controls, allowing you to perform minute adjustments. These are the monitors you buy if neither space nor money are a problem, although you're not going to get by with just one. Nope, that £10K will get you only a single speaker, so crack open the savings account and buy two. It's the sensible thing to do.

05 Minimoog Voyager 10th Anniversary

Synths are another area where you could spend hundreds of thousands of pounds buying vintage equipment, but if we stick to new stuff, a few things become apparent. It has to be a Moog, and it has to be hand-made – also covered in gold, inlaid with awabi pearl from Japan and finished with transparent knobs. Because who wouldn't want a piece of studio equipment that not only screams the fact that your studio is probably installed above a Scrooge McDuck-like money pool, but also sounds absolutely astonishing? We'll take three, please. At £10,000 a pop, of course. Hopefully, Moog will throw in a complementary pair of white gloves.



04

The **AMS Neve Genesys G128** by definition is **an amazing console** – it is a Neve, after all


05


06

06 AMS Neve Genesys G128

Do you work in the City of London? Take home a six-figure bonus? Splendid! Here's the mixing console you need. The AMS Neve Genesys G128 is an amazing console by definition – it is a Neve, after all – but this is one that really pushes the needle into the red. Never mind the 64 channels of mic/line and 64 channels of DAW/tape monitoring, the real draw here is that this desk is modular, allowing you to mix and match, and even expand, your desk if for some ungodly reason you need more than 128 channels. Oh, and it comes with fader and button recall. You need never use the tape and sharpie again. Although if you let sticky tape come anywhere near your £104,000 beauty, you need your head examined. MTF


BANKERS
DON'T FEATURE ON
BEDROOM WALLS.
THE POWER
OF MUSIC!


th•mann
MUSIC IS OUR PASSION

Ableton Live In depth

On the disc
DVD ROM
Accompanying project file included on the DVD or at www.musictech.net

What's new in Ableton Live 9.7?

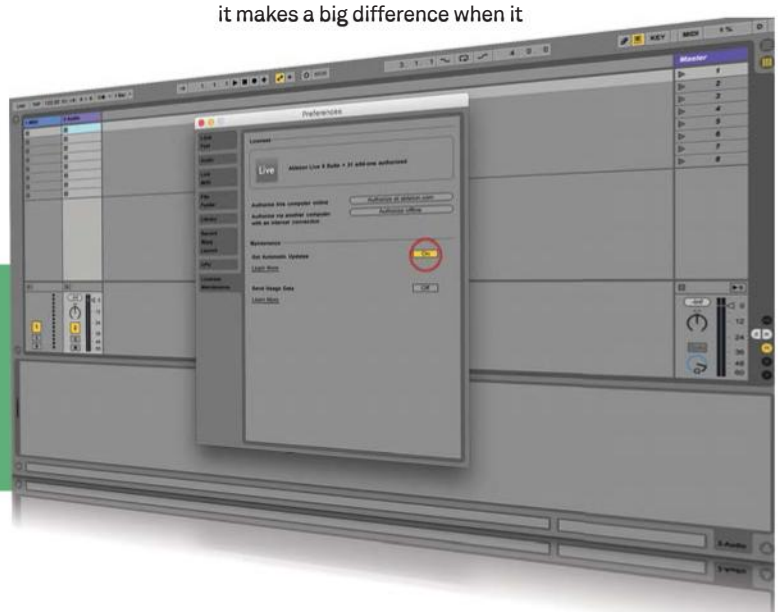
Ableton isn't rushing to release Live 10, but we can forgive the company as it's been feeding us new features for Push and Live in various 9.x updates. **Martin Delaney** details the newness in 9.7...

By the time you read this, Ableton Live 9.7 will either still be in public beta and about to get a full release, or it will already be fully released. I should mention that given this status, things may change before our publication date, so check the Ableton website for the very latest release notes. Live 9.7 will be (or is – sorry!) a free update for anybody using Live 9 – so download it as soon as you can.

You'll see that alongside the typical fixes and tweaks, there are some major new features in 9.7, especially relating to Live's interaction with Push, as that increasingly becomes the de facto control surface for our favourite DAW. In ongoing efforts to keep as up-to-date as possible, I'm using Push 2 for all of these tutorials – it probably won't make any difference in most cases, but it's something to bear in mind.

In recent times, Simpler's been getting a lot of love from Ableton, as it should, and the new slicing modes are interesting: Beat, Region, and Manual. They're really just simple variations on what's already possible but they'll speed up a lot of common slicing and sampling tasks and, like many features in Simpler, seem optimised to help make Push 2 more of a serious sampling platform/interface. That integration is working, as full control over these new features is available from Push. You can choose Mode, Slice By and then the relevant value such as Transient/Sensitivity, Beat/Note division, or Region/No of regions. For the Manual mode, you can tap a pad to specify the slices, and also tap the Delete button, then the pad to erase those slices. Genius! This is a great way to put those pads to work. The ability to access the In/Out View controls from Push is a major workflow bump as well – this goes with the existing mixer controls to make Push a more credible front end for your Live based mixing rig. The menus are divided into very logical columns and fast to navigate. This is definitely a Push function that will be quicker than using the mouse!

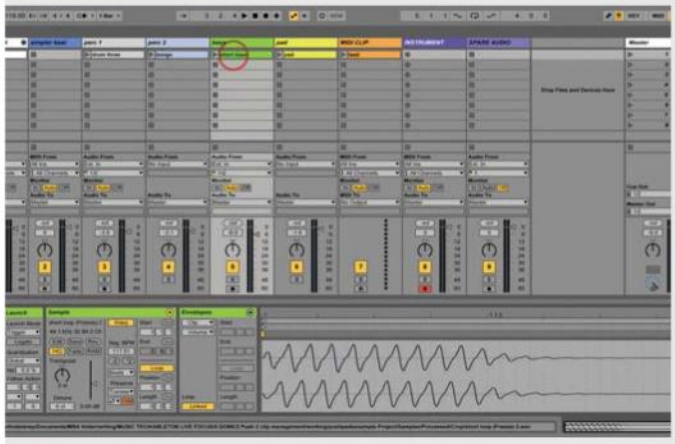
For real-time drum pad players – I hate to say 'finger drummers', but there you go, I've said it – the additional drum rack layout is a real gift. Playing beats becomes a totally different experience and gets great results. You can create really dynamic fills and breaks – it's one of those things I couldn't see the point of until I tried it! Make sure you edit your drum rack's Simpler instruments to be responsive to incoming velocity changes (if they're not already) using the Vel>Vol setting in Simpler's Controls tab; it makes a big difference when it



FOCUS ON VERSIONS

You can safely install Ableton Live betas alongside your regular version. Make sure you're always saving work on a full version though – there's no guarantee a project created and saved in a beta version will open in the future. When you install the final full release of the new version and replace the old one, your Live library should remain intact – but here I'll add the usual comments about the value of backups! And I leave Get Automatic Updates in Preferences/Licences Maintenance turned off; I don't want updates occurring the week before an important project or event!

MTF Step-by-Step What's new in Live 9.7?



01 As usual we've provided an example Live set. This one was created with Live 9.6.2, and you'll be able to open it in any versions from that update. It's unlikely to be backwards compatible.



02 Ableton have stated that because of the changes in 9.7's Simpler modes, projects created in 9.7 will not be backwards compatible with previous versions of Live. Consider yourself warned!



03 Simpler has more features and its relationship with Drum Racks and Push is more intimate. Expanding on Transient Slicing the Slice By mode has three new options: Beat, Region and Manual.



04 Slice By Beat lets you divide your sample with specific note selections rather than following the transients in the waveform, using values from as little as 1/16ths up to 4 bars.



05 With Slice By Region you can slice your sample into anything between 2 and 64 segments. These are only slight variations on the Slice By command but in different situations could be very useful.



06 Finally, Slice By Manual gives you a clean slate, where no dividers are inserted automatically and you're free to add your own by double-clicking on the Simpler sample display.

comes to personalising your racks.

There are a number of visual feedback improvements relating to Push 2, and if you're new to Live or DAWs in general, you might not appreciate how important colour coding and visual information are. They quickly become habits you depend on in the studio especially when you're working with more complex projects, where the more labelling and identification you employ, the faster you're likely to work. They are even more important in live performance situations where it's so easy to be distracted and lose focus. In my live sets I'm using renaming, colour coding, display zoom, different skins, whatever it takes! Push now provides a visual count-in – a little segmented bar across the display – so you can refer to that in situations where you don't want to hear, or can't hear,

MTF Step-by-Step What's new in Live 9.7? (Cont'd)



07 We can change track routings from Push. Select a track, press Mix, then the I/O button above the display. With audio tracks, the left column lets you select monitoring options: In/Auto/Off.



08 Next, select whether to manage inputs or outputs, then choose Sources or Destinations. For MIDI tracks choose MIDI inputs and audio outputs, or MIDI destinations if there are no audio signals.



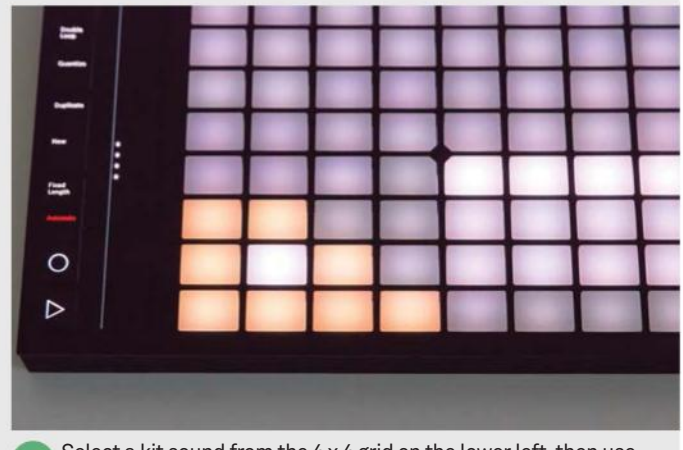
09 In an effort to keep the lists shorter and cut down on scrolling time – sensible – routings for Pre FX, Post FX, and Post Mixer appear in a separate list.



10 Drum Racks now have a third layout available. The currently selected pad can be triggered with one of 16 pre-defined velocities; a first tap of the Layout button goes into this mode.



11 The 16 pads on the right, traditionally used for loop length, are now used to activate different velocities. They're shaded in intensity from more-or-less black to bright white.



12 Select a kit sound from the 4 x 4 grid on the lower left, then use the lower right 4 x 4 grid to play it with different velocities. You can create great-sounding dynamic drum fills!

→ the usual metronome count; very useful!

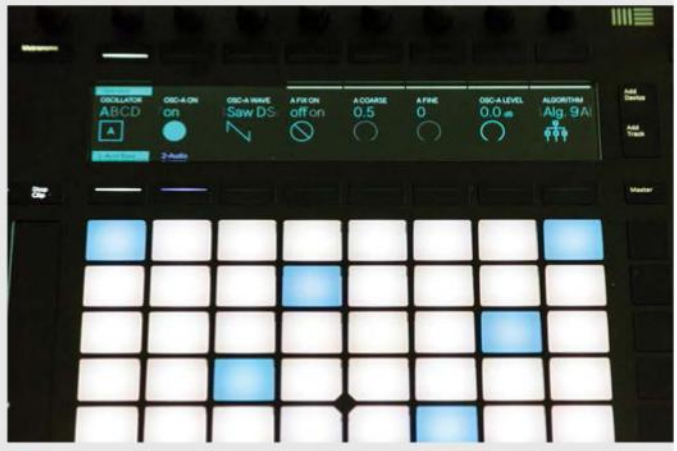
Live 9.7 also includes a small but important Link tweak – Ableton Link, of course, being the means by which Live and various iOS music apps can sync over WiFi. The timing of connected iOS apps will line up correctly when Live's transport is started with either tap tempo or the Looper audio effect device. (Looper has an option to start Live playing when loops are triggered.)

Although Push has functioned as usual with mixing and device controls, when Live's been in Arrangement View it hasn't had much else to offer for when we must work in a timeline. So not only is it good to see the changes to enable either Global or Session recording when working in either view with Push, it's now also

MTF Step-by-Step What's new in Live 9.7? (Cont'd)



13 You want pretty colours? We can change the colours of clips, tracks and drum rack pads from Push. Select the item first, then hold Shift and choose a colour from the coloured pads on the grid.



14 There are further visual feedback improvements, as you'll see if you look at the Push display when you begin recording. If you've enabled count-in, you'll see a broken bar 'count' from left to right.



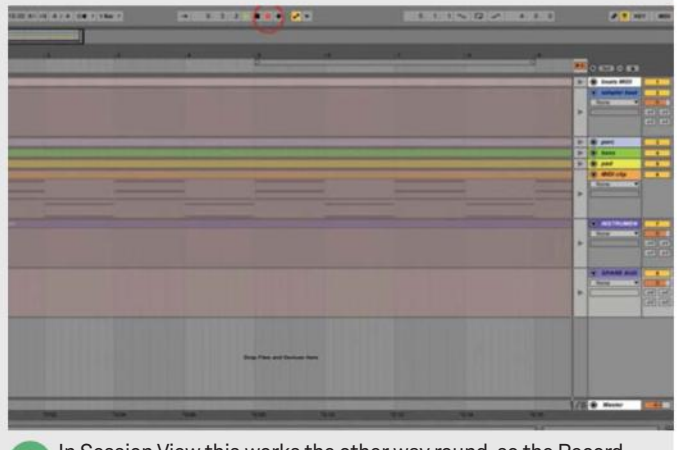
15 Similarly, any time you have clips running in your set, in any display view that includes track names, the Push display will show a small playback progress bar for each track.



16 Push has always been about the Session View, and really not so exciting when it comes to the Arrangement View. Ableton has addressed this as the Push Record button is now view-dependent.



17 So when you're in Arrangement View, the Push Record button starts global recording (into the Arrangement View timeline), which pressing Shft>Record can still initiate Session recording.



18 In Session View this works the other way round, so the Record button starts Session recording while pressing Shft>Record will stat global recording – very logical and you'll get the hang of it quickly.

possible to delete Arrangement View clips from Push – it's a small step in the right direction!

As I said, most of the Live 9.7 update is concerned with Push which I think is perfectly reasonable at the moment, especially given that it's a free update – it's not like, if you don't use Push you're being asked to pay for features that you won't use. It seems like a good use of resources to concentrate on making Push integrate as tightly as possible, if it doesn't affect Live's usability for other people – and of course everybody gets to use the new features like Simpler's enhanced slice modes. There's no sign of Live ten/10/X, whatever it will be called, so let's enjoy the little and not-so-little additions that Ableton is including along the way. **MTF**

Ableton Live In depth

On the disc
DVD ROM
Accompanying project file included on the DVD or at www.musictech.net

Create custom Drum Racks with Ableton Live

Drum Racks are one of Live's most powerful features, whether you're using them for just for drums or far more. **Martin Delaney** flexes those finger drumming fingers…

Drum Racks are the main building blocks for programming or re-sampling beats in Live. Whether you're entering notes with a mouse or trackpad, or recording in real time from pads, Push or a MIDI keyboard, Drum Racks are Ableton's most important beat making tools, as well as being useful hosts for other types of sounds and effects.

A Drum Rack is a group of Live devices which contains up to 128 drum pads, and uses Live's Simpler instrument to host the sample content for each of those pads. Alongside Simpler, you can also expect to see MIDI and audio effect devices, as well as third-party instrument or effect plug-ins. This is backed up with the potential for very sophisticated MIDI and audio routing, as well as macro controls for more accessible hardware controller assignments and, vitally, the ability to save the entire thing as a single preset. Did I mention that each pad can also contain another Rack? That means you get Racks within Racks within Racks. This is powerful stuff, but luckily it's also easy to get started with.

Our focus here's on customising existing Drum Racks, but there's an empty Drum Rack provided in the Live Library, so if you want to start from scratch, use that, and drag in your samples, letting Live put them into instances of Simpler automatically – you can also browse and load samples from Push. Drum Racks are visually complicated things at times, so it's a good idea to use the small buttons at the left to fold and unfold elements on demand, or else double click on the Rack name and fold it right up, in the usual Ableton style. As I've said, there are many ways to trigger the sounds in a Drum Rack, but I find that it usually comes down to the computer keyboard if I'm travelling, or Push if I'm at home – one extreme to the other.

I love to add synth tones to Drum Racks, whether it's to add simple synth-based drum sounds or squelches, or to incorporate some full-on instrument sounds and patterns within the Rack. Operator is my go-to synth for this – I like how it sounds and how the interface works, and I've been using for a while so I have a good idea of where I'm going when I start it up. You'll have your favourites but if you haven't tried Operator, I'd recommend giving it a go. Another angle on this would be to add the external instrument device so you can incorporate your pet hardware drum machines and synths into your Rack – of course that means you always need them connected and powered up when you use that Rack. When you start using software instruments inside Drum Racks, it's a good moment to



FOCUS ON IMPULSE

Live did drums before it had Racks, you know. Originally Live used the Impulse instrument to contain sample-based drum kits. Impulse is a very basic-looking device, but it has its fair share of sound shaping tools, and of course you could always put it in a Drum Rack. Impulse is still included with Live mainly, I guess, for legacy purposes like opening old projects. I could imagine Ableton dumping it altogether and creating a Drum Rack 'Impulse legacy mode' where it automatically creates a Rack with eight chains when you open a project with Impulse in it.

MTF Step-by-Step Create custom Drum Racks



01 The Drum Racks you'll have available in your sets will vary depending on which Live Packs you've got installed, so we've provided a Live set with a custom Rack with a mixture of samples.



02 Keep clicking on the black buttons at the left of the Rack until you've unfolded everything. Going from left to right, you'll see Macros, pads, the chain list, an instrument and audio effects (if any).



03 The effect situation is complicated by the fact that Racks can include effects that apply to individual chains or the entire Rack, and also it can include other Racks! It's deep... get used to it!



04 Trigger the sounds in the Drum Rack in several ways. Use Live's virtual MIDI keyboard – aka computer keyboard – or any MIDI keyboard or a pad-device like Novation's Launchpad or Ableton's Push.



05 Drum Racks usually default to note C1 triggering the bottom left pad, which is typically a kick drum. If you're using the computer keyboard (Shft>Cmd>k), use z and x to transpose octaves.



06 The very lowest black button is Auto Select – a very useful navigational aid. When enabled and a pad triggered, the relevant chain is highlighted. This saves a lot of scrolling in complicated Racks!

consider using MIDI effect devices as well; other than Velocity, the most obvious candidate is probably Arpeggiator, which means that when you play a pad, you'll get a pattern playing, even without a clip in the track. By combining Arpeggiator with time-based audio effects like Beat Repeat, Simple Delay, and Ping Pong Delay, you can get a lot of activity out of one note. On the same topic – sounding busy without doing much – you could be launching loops from inside Simpler instruments in the Rack at the same time, and they'll run and loop even if Live's transport isn't running, which I think is pretty damn cool. You can have a whole bunch of looping tracks as a single recallable preset. This instant recall of complex Racks goes on with audio effects as well. Not only can you add audio effects to the entire Rack or individual chains, you can use the →

niche audio

INSPIRATIONAL SOUNDS FOR CREATIVE MINDS



DYNAMIC HOUSE

DEEP BASSLINES, FUTURISTIC SYNTHS AND PUNCHY DRUMS



CHILLED TRAP



SOULFUL HIP HOP



DARK BASS HOUSE

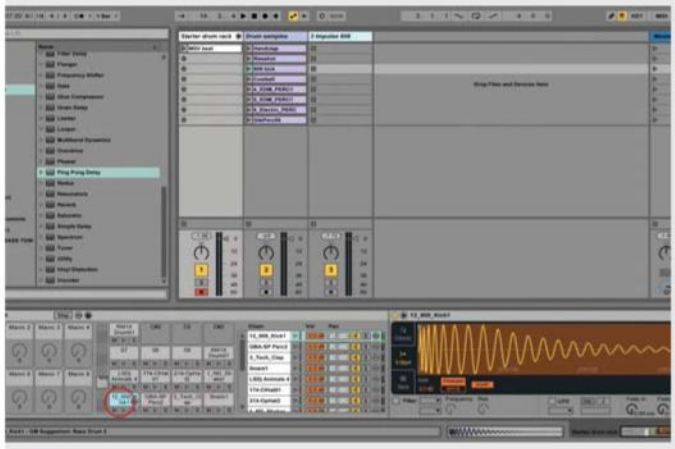


POST DUBSTEP & FUTURE GARAGE

CUSTOM KITS TO SUPERCHARGE MASCHINE AND ABLETON LIVE

AVAILABLE EXCLUSIVELY VIA WWW.LOOPMASTERS.COM

MTF Step-by-Step Create custom Drum Racks (cont'd)



07 Now, let's suppose we want to begin customising the kit, by replacing the kick. Drag the clip '909 kick' from the Drum Samples track onto the existing kick drum pad to replace it.



08 We have more sounds in Drum Samples; try dragging those into the Rack. Empty pads show their note number, occupied pads reveal their note number in the Status Bar at the bottom of the screen.



09 Our Drum Rack has 13 samples to begin, so there's room to add three more samples to the 16-pad layout. If you keep adding beyond that, no problem as Drums Racks have 128 pad slots.



10 Once you exceed 16 pads, you can navigate around them in blocks of 16. Use your mouse to click/drag in the left grid, or the control strip or octave up/down buttons on Push.



11 Push users, take note – if you're in Note mode and you're seeing the contents of a Drum Rack, press Layout twice to enter 64-pad mode, so you can access and trigger more pads at once.



12 Velocity response is a huge part of Drum Rack design, so don't forget you can go into the Simpler instruments that are hosting your samples, and edit the Vel>Vol value in the Controls tab.

Rack's built in send effects slots, then mix each chain to them just as you would with Live's return tracks. The advantage of this is that once again it's recallable – one Drum Rack preset with your sounds, MIDI effect patterns, Simpler loops, and effect sends and returns. I wish there was a way to expand the chain list upwards, so I could see more chains at once, but what you can do is unfold the tracks containing Drum Racks, and view each chain in a more typical Session View type layout. You can access this view from Push, as well. Those chains can be rearranged by dragging them around to reorder them and naturally you can also duplicate or delete them, as well as rename and colour code them. You can also drag chains out of the Rack to create new

PLUGINBOUTIQUE
STAFF PICKS



TRICKY TRAPS
FOR CREATIVE EXPLORATION

STAFF PICK

"Controlled chaos with infinite potential! I'm finding *Tricky Traps* to be intuitive and immediate for generating ideas and random variations. The melodic and rhythmic sequencers interact in a seamless manner which provides a precise programmed feel. It sounds intense as well! "

Alastair
Plugin Boutique



Musician and programmer collective Sonic Faction's new pack Tricky Traps is a toolkit that's designed to offer fresh ways of working with MIDI and modulations within Ableton; 16 creative quirky contraptions that are designed to be chained together to create unexpected rhythmic and melodic patterns – those sought-after 'happy accidents' that can bring moments of true inspiration.

SONIC FACTION

PLUGIN BOUTIQUE.COM
THE PLACE FOR MUSIC PLUGINS

MTF Step-by-Step Create custom Drum Racks (cont'd)



13 You could also add the Velocity MIDI effect device in front of the entire Rack if you needed to smooth out incoming velocities for some reason, or even go the other way, and randomise them!



14 Using instrument devices and plug-ins other than Simpler is a great way to expand the sonic range of Drum Racks. Ableton's Operator synthesizer, for example, is perfect for this.



15 Audio effects can be placed at the end of a Rack so they affect the entire kit, or on individual chains; try some different compressor settings on the kick in our example Rack.



16 We also have the in-Rack effect sends – put a Ping Pong Delay in there and send some of the percussion sounds to it. Effects in here are also saved with your Rack presets.



17 MIDI effects other than velocity can also be used inside Racks. Drag the Arpeggiator to the kick chain in our Rack and set the rate to 1/4 – when you trigger the kick pad you get patterns!



18 You can reorganise the chain list at any time by dragging them up and down to reorder them. You can also duplicate or delete, as well as rename and colour code them.

→ tracks, and drag the contents of pads to other pads.

As of Live 9.7, Drum Rack pads can be coloured individually, directly from Push. It's great if you need to highlight certain pads, or you dream of creating an 8 x 8 pixel image for your Rack. These colours are remembered by Push whenever you load the set or select the individual Rack, or reload the saved preset Rack into a new project.

There's more we could mention, like choke groups or MIDI note filtering, but that just gives us something to look forward to! The best way to get a handle on Drum Racks is to explore Live's factory Racks and then start building your own – and a hardware controller with pads will make all the difference. **MTF**

MTF Feature Produce a track from start to finish

7 WAYS TO FINISH A TRACK

Part 4

We're nearly at the end of our 'produce a track' special and now it's time for some advice on that hardest of tasks: finishing a track...

01 Learn to delete

One of the things that gets in the way of finishing a track can be having too many ideas kicking around on your hard drive. Opening up a folder crammed full of half-finished songs can be depressing and fill you with dread – when will you ever get around to completing some of those tracks that you've had for so long? One of the key secrets to composition, then, is actually learning when to discard old ideas. What is good? What is worth keeping? More importantly: what can you delete? Getting rid of old ideas can be nerve-racking at first, but actually binning rubbish can be surprisingly rewarding. Have a clearout once in a while and you will be surprised at just how focused you become on finishing what you have left.



1. Got a load of old song ideas? Maybe just get shot of them...

2. Bounce, bounce and bounce again!

02 Bounce, bounce and bounce again

No matter how far you are into making a track, it's always a good idea to bounce your work down to a stereo file so you can listen to it away from the studio. Load it on to your phone, play it in the car... you'll be inspired on where to take it and how to finish it. This is also a good exercise to do to keep focus. If you keep all the versions you have made of a song and then go back to an earlier one every so often, you might rediscover the original direction and point of the tune, which might have got lost along the way. Making music with technology can take you in all sorts of directions, very often far away from where you originally wanted to go, so using this technique can help you stay on your original path.





03 Work to a deadline

If you have someone waiting in the wings demanding you finish something, there's a much better chance that you'll actually finish it! We're not advocating employing someone to stand next to you looking at their watch and shaking their head every few minutes, but having a deadline is a good way to get focus and, as you might be realising by now, focus is the key! Set your own deadline, if you must, but try getting a track done within a certain time period and you are almost certain to get more done (whether you finish it or not).

04 Accept that it won't be perfect

Don't take this the wrong way, as it's always good to aim for perfection – the higher you set your sights, the better your results. But – and it's a big but – if your level of perfection is too high, there's a very good chance you won't hit it. If you accept your work will never be perfect, you will finish it. Why not deliberately add subtle mistakes, so you know perfection will never be achieved? Flaunt that imperfection – at least your music will be out there.

05 Get more ears… but not too many

Getting bogged down? Been working on the same loop for hours? Losing focus? Perspective gone? Get someone else in! Having a fresh set of ears to comment on what you are doing or even help it along is always a good idea but – as with everything to do with music production – don't overdo it. More ears can mean more ideas and even less focus. Did we mention the 'focus' word again?



3. Set your own deadline if you have to and try and stick to it

4. Accept that no track is going to be perfect and you might finish it. You could even consider deliberate imperfection!

5. Astronautica offers advice at www.novationmusic.com

6. Novation has created a video asking artists such as Floating Points how they finish their music, at uk.novationmusic.com/finish-something

06 Limit your options

This one could be applied to the whole process, not just finishing a track. If you limit your options, you can get things moving and get to an end point quicker. So you might want to consider keeping your track-count down – why not go old-school with eight tracks? – or keep the number of instruments and effects you use to a minimum. With far fewer options you'll not only start your productions more quickly, but you'll also finish them sooner, too.

07 Know when something *is* finished

It's the ultimate question: how do you know when a track is finished? When it's ready to be shared? When you've played it to death and not criticised it? When you play it back after a week of not hearing it and it makes the hairs on your arm stand up? The answer is all of these and more, but don't let it fester on your hard drive for too long. Finish something today!

We nicked some of these from a rather good video that the folks at Novation have put together on finishing music. You can see more here uk.novationmusic.com/finish-something. MTF







You can't beat a bit of modular. Quite a bit of modular...

Show off your Live studio

Welcome to part 1 of *MusicTech*'s **Show Off Your Studio**, where we highlight our readers' studios. Send us a picture of your studio via the *MusicTech* **Facebook** page and we'll get in touch...

Omniscient Records

Interviewee: **Josh Tonnissen (Praxis, Wave Mechanix, Konjecture)**

Contact: **omniscientrecords@gmail.com**



MT: What are the key components of your studio?
Large Eurorack modular system with 12 rows at 104hp each; Koma Elektronik Komplex Sequencer; DSI Prophet 12, DSI Tempest; Moog Voyager; Access Virus C; Roland TR-909, TR-808, TB-303, SH-101, SH-1; Sequential Circuits Pro-One; Korg Mono/Poly; Midas Venice F32.

Which DAW do you use and why?
Ableton Live. Coming from a hardware-only setup from the late nineties and into Y2K, most software applications were geared around recording guitar dude music. Because I used lots of different outboard units, my brain was wired for multiple sequencers running various blocks of sequences. When I discovered Ableton at, like, version 4 or so, I was totally blown away. Their DAW was designed almost completely around the way a hardware configuration runs. Session mode is totally brilliant, and allows for endless jamming without ever stopping the sequencer. Since I run a large hardware-based studio, Ableton offers a lot of different ways to interface all of my gear in an efficient and well-designed manner. From external MIDI sequencers, DinSync gear, voltage-controlled analogue synths and analogue clock for the modular, Ableton never skips a beat.

What is your favourite piece of gear and why?
That would have to be my modular system. Since the rebirth of modular (Eurorack), synth designers started building their gear around the way electronic musicians work. Most of the keyboard synth companies build great gear, but the Eurorack guys really focus on innovative designs and push the boundaries of synthesis. I use the modular for everything, not just synth voicing. I will run all my various other gear through the modular for filter and effect processing. Think of it as the ultimate time-synced FX unit ever made. There are also several complete voice drum modules in the

system, making for a standalone techno tool.

How often do you use the studio on average? Not nearly enough time of course! Right now about 10-16 hours per week, but I have been in there up to 35 hours a week when work doesn't get in the way.

How do you use your studio? My studio is set up as a dance music production lab, and while I occasionally collaborate with other artists, I generally work by myself. At the end of the day, I just make music for fun; however, I do release music professionally and do various remix projects for other artists and labels.

Does anything annoy you about your studio set-up? I have a love and hate relationship with my hardware overload. Since I have all of my inputs and outputs (including CV/gate) ran into five 48pt patchbays, every time I have a cable failure it is a nightmare to chase it down. There can also be the occasional sync issue, which drives me absolutely crazy, but thankfully it doesn't happen too often. I must say, though, nothing is worse than when things go wrong with my custom-built PC. That's why it has never even been on the internet!

Next on your shopping list? Probably more modular gear… but ideally a nice new set of studio monitors and acoustic treatment panels. Not sure what to get yet for monitors, but it would be ideal to have a higher-end setup to complement my production tools.

Dream piece of gear? That's a tough one. I am always lusting over various processing gear, but never pony up to buy it. I would love a Culture Vulture, and a few Distressors and valve preamps, but I will honestly probably spend the money on more modular gear…

What advice do you have on studio creation? Build a powerful desktop PC with a killer soundcard first. Then research your favorite artists and see what sort of gear they are using in their studios. Every synth had a different sweet spot, so see what your favourite artists are doing with their gear and take their results as recommendations. Buy one piece at a time and learn it front and back. Everyone says this, but it's totally true. Ditch Facebook and start reading technical documentation for your gear; you'll thank me later.



Get a good PC (as well as a fantastic Eurorack system)



There's a pretty good set of keyboards and classic Roland gear too…

Any other studio-based anecdotes? While anyone can learn the technical side of studio production, the real magic is in the moments of uninhibited inspiration, those moments in between the tech that all of your ideas hatch from. While you cannot always summon the essence of inspiration, you should always be ready for when it comes on. When I do break through the concealed barrier into full flow, I find my ideas coming at me like a kaleidoscope of butterflies, and I am poised with a net attempting to catch as many as I can. It's important to do the mundane technical studio work (running cables, testing connections, etc) when you have no flow, so that when the butterflies are upon you, you are ready to bathe in their radiance. MTF

MTF Feature Ultimate gadgets

ULTIMATE Music Gadgets

We all love a good gadget here at *MusicTech*, so how about treating yourself to some new hardware for your studio? We round up some of the best technology out there – within a variety of categories – to help you liven up both your production setup and your sound...

Computerless music production

There's a move away from relying on computers for music making and these self-contained gadgets will have you making tunes with a box (or two) – no glaring computer monitor required...

ROLAND AIRA TB-3 AND TR-8

The AIRA range was announced a couple of years ago to silly levels of hype, as it promised to be the closest range of new Roland gear to emulating classic old stuff, in this case THE Roland classics: the TR-808 and 909 drum machines and TB-303 bassline. The AIRA TB-3 is based on the classic 303 and provides all of the acid house squelch you could wish for and the TR-8 has all of the beats. Both look amazing, with great, well-lit appeal and – more importantly – both also sound as close to the real thing as anything digital can. And with that sound as well used as ever, our thoughts are very relevant... "A mere £219 for the TB-3 and the TR-8 is just £365 (both street), seriously, if you just used them as Christmas lights that would be worth it". Nothing could have matched the hype that surrounded the launch, but the TB-3 and TR-8 are both innovative and important. They undeniably look extremely cool and they sound just, well, damn close to the original. But are they as good as the originals? "They do loads more and they cost far less", we said, "so the answer is: you betcha. But their importance is that they are priced in exactly the right way to get the next generation of producers off their sorry sofas and actually performing again. So will they, as their forefathers did, kick-start another dance revolution? Well, we are due another Summer of Love..."

Roland's TR-8 (below) and TR-3 (bottom) come close to the classic sounds of the TR-808 and 909 drum machines and the TB-303 bassline





Price **TB-3 £245, TR-8 £399**
Contact **Roland on 01792 702701**
Web **www.roland.co.uk**

KORG'S VOLCA RANGE

They might be (mostly) a few years old now, but Korg's Volca range is still a great set of sound makers, and available at silly prices. In many ways, they kick-started the current shift to hardware music production and also set the form, with Beats, Bass and Keys models being responsible for all of those sounds (plus a newer sample-based one offering a little more flexibility and your own creative recording, and the even newer FM offering 'DX' type sounds). Sound-wise, they are all great with stacks of hands-on control, and when we reviewed the original three, Alex Holmes said: "If you are looking for deep sound design you probably need to look elsewhere. However, to get a true analogue synth with built-in sequencer for this price is a great deal. Korg has listened to its customers and struck the perfect balance of price and features."

Price **£119 (£99 street)**
Contact **Korg on 01908 304600**
Web **www.korg.com/uk**



Novation's Circuit (left) wowed us when it was released last summer, while Teenage Engineering's PO Series (below right) caused a stir at NAMM

NOVATION CIRCUIT

Novation's Circuit came out of nowhere just las t year and updates the Groovebox philosophy for the 21st century, making computerless music production – a trend set with the Volcas and Airas – an easy and fast reality. Indeed, it's this speed that puts Circuit in a class of its own. You can simply do everything you want it to do very quickly. Circuit's combination of great onboard sounds and pseudo keyboard playing can take what is in your head to a tune fast – something which, as we said earlier, you don't always get with many grid-based controllers and push-button sequencers. "This is a truly self-contained musical marvel," we said. "It excels so well at what it sets out to do – and the price of £250 is just ridiculous – that you can forget any shortcomings. It's the most musical fun you can have with one box and no computer, and we haven't made music so quickly in 20 years of reviewing gear. Great real-time sound mangling and creation, great fun and you will be making music in less than three minutes. If you don't, we'll come round and do it for you..."

Circuit's great onboard sounds and pseudo keyboard take what's in your head to a tune fast

Price **£249.99**
Contact **Novation on +44 1494 462246**
Web **uk.novationmusic.com**

YAMAHA REFACE

Yamaha's idea with the reface range of keyboards is to include one classic keyboard in each of four mini keyboards. You get the DX (based on the classic FM synth); the CS (based on the classic analogue); the CP (on the electric piano); and the YC (based on the 60s and 70s organ). The latter two are superb, while the DX is more fully-featured with presets and onboard saving (all use an app to do this otherwise). The CS has no presets, but sounds glorious. Reviewer Andy Jones said: "I find the range charming – carry them, play them and hear them without connecting to the mains or dragging in your entire studio. They are sufficiently different to appeal to both specialist studio-heads – who are after some light relief and instant hands-on inspiration – and to a mass market. I'm pretty sure that at least one of the range will appeal, and I'd urge you to check them out."

Price **Street £289**
Contact **Yamaha UK on 0844 811 1116**
Web **yamahaproaudio.com**

TEENAGE ENGINEERING PO SERIES

The new PO series from Teenage Engineering was the buzz of this year's NAMM show, and they certainly didn't disappoint when we reviewed them a few months back. There are now six in the range and they are all incredible (especially given their stupidly small size) and are, without a shadow of a doubt, aimed at cooler music creators, which didn't stop our editor from piling on the superlatives. The range comprises the PO-16 'factory melody micro synthesiser', the PO-12 rhythm synthesiser for beats; the PO-14 sub bassline micro synth; PO-20 Arcade; PO-24 Office and PO-28 Robot. Each has a 16-step sequencer, of which you can select 16 sounds to program up to 16 patterns, of which 16 can be chained. There's plenty you can do sound-wise and effects-wise to each and if you chain all six together you have your entire tune covered. "These units sound fantastic, they're incredibly fun and are as cheap as silicon chips," we said. We then bestowed just about every award we can muster upon the range.



Price **£49 each**
Contact **via website**
Web **www.teenage engineering.com**



Cool control – keyboards and more



There was a time when MIDI controllers comprised just a few rotaries to control a couple of key features. Now, controllers are used for everything from grid composition to production, mixing and performance. Here are some very different, cool models...

BEST **ADDED DIMENSIONS**

KEITH MCMILLEN INSTRUMENTS QUNEO



MIDI controllers such as the Keith McMillen Instruments QuNeo (left), M-Audio Trigger Finger (above right), Arturia Beatstep Pro (below) and Ableton Push 2 (bottom) now offer a wealth of functions

It's been around a while, but if you want to stand out we reckon the QuNeo is the one to beat in terms of control. Pads, sliders, buttons and rotary encoders can all output note, velocity and pressure sensitivity, earning it its name as a 3D Multi-Touch Pad Controller. The best part is each control can manipulate so many parameters, which might take you a while to comprehend. But once you do – and the QuNeo editor makes customising the device very easy – you'll discover a level of control that is unlike most other devices. As we said when we awarded it a mighty 10/10: "It's nice to find something that's so revolutionary, powerful and unique at an affordable price. We can heartily recommend it."

Price **£200**
Contact **MSL Pro: 0207 118 0133**
Web **www.keithmcmillen.com**

BEST **FOR LIVE**

ABLETON PUSH 2

Ableton Live is the music production software phenomenon of the last 10 or so years and Ableton's Push is its hardware master. If Live is Darth Vader, Push is Darth Sidious. But rather than just being a controller over the software – which it obviously can and does do, better than most – Push becomes an instrument in its own right by taking some of the hidden features of Live and bringing them to the (control) surface, so realising its creative potential on stage and in the studio. Push 2 was released last year.

Price **£499**
Contact **Ableton via website**
Web **www.ableton.com**



BEST **PAD CONTROL**

M-AUDIO TRIGGER FINGER PRO

The original Trigger Finger was one of the first controllers that dared to be a little different. Before it arrived, controllers were usually a bank of rotaries or simply keyboards. Trigger Finger Pro, a much-expanded and upgraded device that's a bit of a quantum leap from its predecessor. Designed to be used both in the studio and on stage, or even without a computer, it has quite a few potential users in its sights. Like all good controllers of this ilk, it makes beat making a breeze but also doubles, actually trebles, its control into other areas. It includes a step sequencer to put together both beats and melodies and it can control your DAW with sliders for mixing and rotaries to take care of plug-in duties. One of its best features is the software that is included with it – 5,000 samples, instruments including VST and Arsenal, which enables you to host VSTs with no DAW running.

We said: "A powerful and playable controller with some great bundled content that is equally at home on stage or in the studio. Lightweight, but well built, very playable and great hands-on control."

Price **£299.99**
Contact **InMusic via website**
Web **www.m-audio.com**



BEST **ANALOGUE AND DIGITAL**

ARTURIA BEATSTEP PRO

One of the best compact controllers out there, it's not immediately obvious what BeatStep Pro is for because, in fact, it's for pretty much everything: a controller, performance sequencer and composing tool in a single box. You get control over your instruments, be they hardware or software, and it has a more expansive selection of outputs than you might expect, with all-important CV outs to let it get to the midst of that cool modular analogue synth set-up you either have or want. The original Beatstep was good, but BeatStep Pro is a great step up from the smaller model for people who want to control different types of kit simultaneously, especially those who have at least a couple of the aforementioned CV-capable modules. Even if you don't and you're more reliant on software-triggered or MIDI hardware-triggered sounds, there's much to like. Workflow feels natural, and transport and remote control are possible. Arturia has struck the right balance of features, build and price, and proves controllers can be fun and modular synths needn't be the preserve of hardcore synthesis nuts. A fun multi-talented controller and sequencer that will tie together the old and new parts of your studio nicely.

Price **£179**
Contact **Source Distribution on 020 8962 5080**
Web **www.arturia.com**



BEST FOR GRIDS

NOVATION LAUNCHPAD PRO

The Launchpad range takes grid-based control, made so popular with Ableton Live, into new dimensions. The Pro is a tad larger than the regular Launchpad, and as Martin Delaney said: "There's plenty of newness to be explored, with velocity-sensitive full-colour RGB pads, hardware MIDI, mains power and an array of edit buttons; that's at least two items ticked off my personal wish list." He was also very positive about the new full-colour feedback from the RGB-illuminated pads, not to mention the great ease of use including fast access to commonly used controls. The unit features brilliant velocity-sensitive faders, the hardware MIDI out for synth control is dead easy to set up, and it's fun to use, what more could you ask for? "In the studio, it's not as powerful as Ableton's Push in the beat and device control departments, but for other situations, the Pro is a flexible, expressive, interface, for your clips, instruments, and just about everything else. This is a great update all-round, and takes the Launchpad to a new level of Ableton Live control and performance. If you already have a Launchpad, you'll be itching to update. If you don't have one, the Pro is the perfect excuse to get on board."

Novation's Launchpad Pro expanded on the functionality of the regular Launchpad with velocity-sensitive full-colour RGP pads, hardware MIDI and a range of edit buttons

The Launchpad range takes grid-based control, made popular with Live, into new dimensions



WHATEVER HAPPENED TO...

Jazzmutant Lemur and Dexter

French company Jazzmutant released the first multi-touch controllers especially for music producers in the forms of the Lemur and Dexter. The Lemur could control pretty much anything you defined, while Dexter was aimed at specific DAWs (Logic, Cubase, SONAR and Nuendo). These sleek, black units still look as if they are from the future, and were always a wormhole ahead of their time, not to mention expensive. Perhaps unsurprisingly, the iPad offered a similar experience for far less cash and the Dexter continues within that format. The Jazzmutant website highlights v2 software for the hardware, although we assume that hardware is no longer made. The app is well worth checking out and if you can track down the units second-hand they are still damn cool. **www.jazzmutant.com**

Price **£229**
Contact **Novation (via web)**
Web **www.novationmusic.com**

BEST COOL AESTHETIC

MONOME

Next to the other controllers in this round-up, Monome stands out rather like a geek on a catwalk. It's been around for the better part of a decade and now comes in Grid or sequencer forms. The former is a 16x8 grid



and is the most recognisable, kind of like the Yamaha Tenori-on in terms of its lights, but minus the (not so great) FM sounds. It's a USB box and you pretty much define its purpose – anything from step sequencer, to DAW controller and way beyond. We love the thinking and the open-ended nature. $700 doesn't seem too much to be part of the coolest bunch of geeks on the music tech planet. 'Isn't is just a bunch of buttons with lights?' asks a question on the website. 'Yes, isn't it wonderful?' is the response. You've gotta love that.
Price **$700**
Contact **Monome via website**
Web **http://monome.org**

BEST KEYBOARD

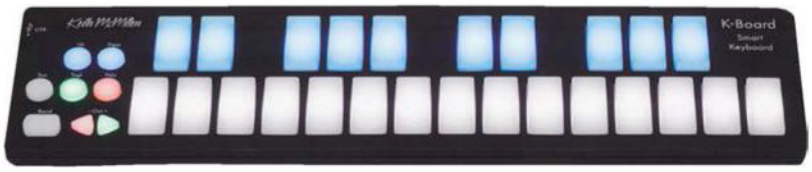
NATIVE INSTRUMENTS KOMPLETE KONTROL

Alongside the Akai Advance series, Komplete Kontrol is just about the best keyboard controller out there in terms of total software integration. The idea of a great hardware controller is you can turn away from your computer and do a lot of your noodling actions without it. Komplete Kontrol enables this more than other options by not only having classy displays, but multi-coloured, lit-up keys that replicate the coloured key zones when using NI software. The Komplete Kontrol software has also recently been updated to v1.5 and you get a bunch of free Komplete Select NI plug-ins, plus the fact there's a brand new 88-note version. That's 88 lit-up multi-coloured keys. Swoon.
Price **£369-£729**
Contact **Native Instruments on +44 207 9207500**
Web **www.native-instruments.com**



Mobile music making

There are hundreds of apps to help you make music, but what about the gadgets you need to help you get the best from them? Here are the finest keyboards, controllers, interfaces, mics and speakers...

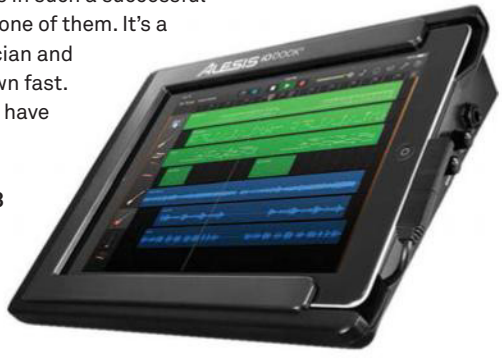


The Keith McMillen Instruments K-Board (above), Alesis I/O dock (below) and Novation iTrack Dock (bottom)

KEITH MCMILLEN INSTRUMENTS K-BOARD

Keith McMillen Instruments makes some of the coolest gadgets for music making on the planet and the K-Board is consequently not the only piece of the company's tech to appear in this feature. If you are prone to getting inspiration for music on the move, or simply don't have a lot of space for music production then look no further than this. It has touch-sensitive keys and is a plug-and-play USB MIDI keyboard that is affordable, durable and even available in different colours. It slots very nicely at the front of your laptop and the lead is long enough to play it (slightly more) remotely. We tested it with a laptop running Logic and the iPad and it rose to the occasion each time. "Once in a while, a product comes along where you think, why hasn't anyone else executed this in such a successful way before? And the K-Board is one of them. It's a must-have for any mobile musician and anyone wanting to get ideas down fast. Flashing lights and playing keys have never been so much fun."

Price **$99**
Contact **MSL Pro: 0207 118 0133**
Web **www.keithmcmillen.com**



ALESIS I/O DOCK
NOVATION ITRACK DOCK

We love the iPad but, to be honest, Apple didn't really design it completely with music production in mind, so producers need a clever way to expand their iPad into a more rounded audio and MIDI recording environment. Enter the Alesis I/O Dock and Novation iTrack Dock. Both allow you to plug iPads in to expand on their connectivity, and both therefore turn them into more usable interfaces. The Alesis is a good-all-in-one solution with great-sounding recordings and the ability to charge your iPad as you work with it. It's also the most competitively priced. The Novation is similar but fits around the iPad a little more awkwardly, but can certainly integrate easily into your setup – however you wish to use it. Both are genuinely good solutions to taking your iPad noodlings into your studio environment and getting audio or instruments from the outside world into your iOS device.



ALESIS I/O DOCK

Price **£119**
Contact **InMusic or Alesis via website**
Web **www.alesis.com**

NOVATION ITRACK DOCK

Price **£169**
Contact **+44 1494 462246**
Web **uk.focusrite.com**

FENDER FOCAL PASSPORT STUDIO MONITORS

How many portable monitors can you get with the quality monitor name that is 'Focal' written on them, not to mention the quality, er, guitar name that is Fender. OK, these aren't quite battery-mobile, nor as 'take anywhere' as the Bluetooth offerings we have here, but the Passports are portable in the sense that if you do a lot of mixing in a lot of places – if you are a flash Harry with lots of studio spaces or work on lots of other people's music, that is – then these offer the same listening experience wherever you are (bar, naturally, studio acoustics that is). Reviewer Huw Price noted: "Judged purely as studio monitors, the level of detail and solidity of the imaging is certainly commensurate with the price point. But the defining feature is that the Passport Studio system can be locked together and transported without the need to preserve the packaging or buy a flightcase. Then you simply split them, set them up and they're good to go. This is a cleverly designed, carefully thought out and

The K-Board is a must-have for any mobile musician and anyone wanting to get ideas down fast

WHATEVER HAPPENED TO...

Yamaha Tenori-on

The story of the Tenori-on follows a similar path to Jazzmutant's. Both were released to critical acclaim, both tried to do something a bit different and both were adopted by cool musicians. Both have also ended up as iOS apps with the original hardware only available second-hand. But the original Tenori-on is worth tracking down for its light show sequencing and the fact that you could quite easily put together a multi-track tune – albeit a very FM sounding one – in no time at all. It'll probably be worth a few quid too... **http://uk.yamaha.com**





affordable product that fills a niche in the market. An interesting concept that's well executed and priced – and they sound pretty good too."

Price **£490 (pair)**
Contact **Fender GBI 01342 331700**
Web **www.fender.com**

BLUETOOTH SPEAKERS

While the Focal speakers give you the studio quality you need, they're not quite as convenient as a good set of Bluetooth speakers, that are all the rage at the moment. We're including two sets – both good in their own right. The iLoud is the Bluetooth speaker of choice for the *MusicTech* office and has been providing sterling service for listening and the odd small office test for a year now. It's easy to set up, sounds great and is certainly good enough to mix on while you're away from your studio with the view to returning to your proper setup to do the main mix/master. Creative Muvo Mini, on the other hand, is smaller, cooler and redder. When we reviewed it we said: "It's portable, sounds louder than it should, and lasts for 10 hours in the rain. What more do you bloody want?"

ILOUD

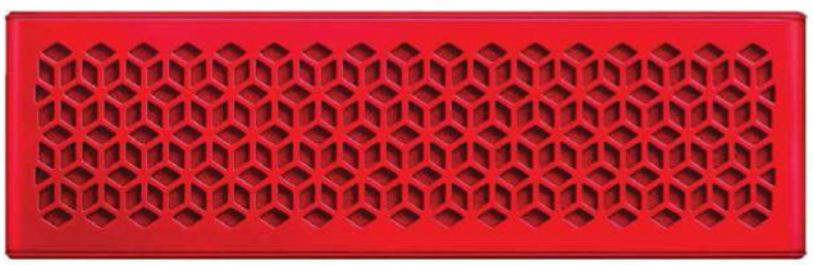
Price **€199.99**
Contact **IK Multimedia via website**
Web **www.ikmultimedia.com**

CREATIVE MUVO MINI

Price **£49.99**
Contact **Creative on 020 8208 9930**
Web **www.uk.creative.com**



The iLoud (below) has been keeping the MusicTech team happy in the office with its high-quality sound, decent volume and portability



IKM IRIG PADS

With grid-based controllers more popular than ever, there's a lot of choice for the musician: from large-scale control and instrumentation with Ableton's Push 2 and Novation's LaunchPad Pro, to these iRig Pads, one of the most portable units out there. It is brilliant for beat creation on an Apple device, and likewise with the bundled SampleTank 3SE for your Mac or PC, which offers 6.5GB (for an additional €80) to take your creativity beyond the beats. With everything programmable, it'll work with APCs, Live, Native Instruments' Maschine, GarageBand and so on, and will save up to 16 scenes to work with such software (it ships with six presets). It is hugely flexible and highly portable (its depth is particularly thin). The emphasis is definitely on grooves and dance, but if you want extra 'i-Control' and are perhaps tiring of touch, this is a worthy option. A compact, sturdy and portable unit that makes a useful, tactile addition to your i-based studio.



Price **€119.99**
Contact **IK Multimedia via website**
Web **www.ikmultimedia.com**

APOGEE ONE FOR IPAD

Apogee describes One as 'the first studio quality microphone and USB audio interface for iPad and Mac'. That's a little bold, but this is certainly one of the first proper interfaces that you can get for your iOS device, tailored for the discerning studio recordist looking for pristine audio quality. Apogee One will charge your device, connects to Mac and iPad, and has instrument and mic inputs. The Maestro software works with Mac and iOS and as we noted in our review: "the sound quality of the mic is surprisingly high and the software makes the interface very versatile. Overall, One is a very useful high-quality mobile tool. It excels at being a great travelling companion, especially with its high-quality onboard condenser microphone."

Price **£289.00**
Contact **Sound Technology on 01462 480000**
Web **www.soundtech.co.uk**



Out there

Speakers that light up, guitars with touchscreens, and beat-making clip machines – it can only be the gear that defies categorisation. Here, we round up some of the coolest, best looking and sometimes strangest gadgets out there...

NUMARK LIGHTWAVE



Before you say we'll include anything with decent lights, no we won't. But when you attach them to a pair of ruddy great big speakers that sound good and rock along with whatever you are listening to and mixing, then what's not to like? Numark's Lightwave speakers have literally lit up our studio this year, packing a volume easily good enough for a house party at half level and more than capable of holding their own outside on a medium scale. The sound is clear, too, with little distortion at high levels. We tried plugging in everything from phones and laptops to keyboards, and were pretty impressed with the results. And the lights even more so, as you get several patterns and can choose the colours and brightness of each or have rotating ones. The Meter option is the best, exhibiting the meters of a mixer, bouncing along with your music – but that might be the inner music production geek in us trying to get out. OK, these aren't mixing monitors really (although do play your mixes on them for comparison), and they're not to be taken too seriously, but they're solid, sound good and will add a bit of colour to your (probably already colourful) setup.

Price **£219.99 (per speaker, street)**
Contact **Numark via website**
Web **www.numark.com**

Manufacturers know that by adding lights to something it makes it 1,000 per cent better





EDITORS KEYS BACKLIT KEYBOARD

Well it's just a standard keyboard, right? OK, it has some labels on for your DAW shortcuts, but what's the big deal? Back-lit? Ah, now you're talking. Yes, when it comes to gadgets, manufacturers know that by adding lights to something it makes that something about 1,000 per cent better! So the Editors Keys Backlit Keyboard might not be the most essential gadget you'll ever need, but it could be the most essential gadget you'll ever need. Martin Delaney said in his review of the Live version: "There's a lot of visual feedback – each key has an icon and a text label on a coloured background – the colours are grouped according to the type of function they represent, which is a good idea. During the review period, I discovered that this keyboard divides Live users into two camps: namely, 'I don't need this!', and 'Wow, this is awesome!', so I understand there's an audience for this. There's a big cool factor that'll sway a lot of people; having a backlit-dedicated Ableton Live keyboard will impress your Live-lovin' friends. If you're one of the many Live users excited by this idea, you won't be disappointed!" There are now versions for Logic and Cubase too.

Price **£99.99**
Contact **via website**
Web **www.editorskeys.com**

KORG CLIPHIT

It looks like a large egg, but you take the three clips that attach to this Korg percussion gadget – plus a footswitch – and attach them to any beatable surface (think table top, item of clothing and so on). Striking that surface then triggers one of the onboard sounds for instant beats and jamming mayhem. The speaker and resulting sound isn't going to win too many pro-quality awards, but the Cliphit is easy to use, great to practise or busk with and tremendous fun. "The kits are good enough, ranging from standard, through rock and pop," we said. "It's another great curveball from Korg. We love the idea, and we love Korg for coming up with this leftfield gear. It's the ideal device to unleash your finger and toe beats."

Price **£99**
Contact **Korg on 01908 304600**
Web **www.korg.com/uk**

KORG MINI KAOSS PAD 2S

The number of Korg's Kaoss gadgets is becoming a tad confusing to us, we must admit, but the KP2S represents a refinement in a technology that has been

WHATEVER HAPPENED TO...

Misa Digital Kitara

The shape was most definitely guitar-based, although the Kitara also had a touchscreen slap bang in the middle of it and used an open source software platform to make sound. This included onboard synth tones and could be played by way of 144 buttons along its neck. You'll notice that we are using the past tense as, according to Misa Digital: "The Misa digital guitar was an experimental instrument, similar to an electric guitar in shape, but with an interface designed to easily and intuitively control digital audio. It is now discontinued." However, the replacement Tri-Bass, which is more of a MIDI controller with touchscreen is still available, although it's sold out at the time of writing.
www.misidigital.com





The JamHub GrenRoom may look a little like the Millennium Falcon from Star Wars, but it's actually a highly useful rehearsal tool for bands

around for more than 15 years now. It is described as 'a powered-up Mini Kaoss Pad 2, the Mini Kaoss Pad 2S is a powerful DJ effect unit that fits in the palm of your hand'. It includes a touchpad in the centre, and retains the large number of effects of its predecessor (100 in total), but adds sampling (explaining the 'S') to the mix, so you can use the device as anything from simple recorder to mobile vocoder. When we reviewed it we said: "Using the device is tremendous fun, of course, and it is perhaps more a mobile gadget for the performer than producer. But the recording option adds to the studio experience, and the more you work with it (and the smaller your fingers) the more you will sonically reap. It's an evolution of an ageing concept, but the KP2S is small and perfect in many ways."
Price **£119**
Contact **Korg on 01908 304600**
Web **www.korg.com/uk**



JAMHUB GREENROOM

In a band? Make a racket when you rehearse? You could probably do with one of these. Essentially, it's a box that each band member plugs their instrument into, plus a set of headphones out so they can hear their playing together with the band, but the neighbours can't. A simple idea, and genius in execution. Obviously, it doesn't work so well if your drummer insists on using an acoustic kit and your singer's wailings will still get heard, but it's one of a kind, and we said: "The GreenRoom excels at providing an attractive, compact and convenient way to rehearse silently at home or on the move. It combines all the essential bits of kit necessary to enable virtually silent ensemble rehearsals in one well-designed, compact and lightweight package."
Price **£420**
Contact **SCV Distribution on 03301 222500**
Web **www.scvdistribution.co.uk**

So traumatised were we by the incident, this is the first time that we've been able to write about it

NUMARK ORBIT

The Numark Orbit has been around for a couple of years, but came to our attention only when we were chased around the NAMM show by a Numark representative demoing it live on stage earlier this year. So traumatised were we by the incident that this is the first time we've been able to write about it... It's a live performance DJ controller with several modes of operation, but think of it in terms of coloured buttons/pads and banks of operation plus motion control, which allows you to assign parameters via the unit's shoulder buttons. So, moving it here there and everywhere – rather like the scary Numark rep – will yield some interesting performance results. OK, it's not for music production, but it looks cool and could add a flourish to your live show.
See, we're over the incident already...

Out there (cont'd)



The Eigenharp range is probably the most intriguing of all the gear we have in this round-up

Price **£79**
Contact **InMusic via website**
Web **www.numark.com/product/orbit**

The Eigenharp (above) has 133 keys and two ribbon strips and is a totally new concept that we can't wait to test out

THE EIGENHARP



The Eigenharp range of instruments is probably the most intriguing of all the gear we have in this round-up, arguably the most serious – in terms of its musical aims – and certainly the most expensive and expressive. Eigenlabs formed 15 years ago, an English company with enough investment to have produced three instruments since 2009, the top-end version being the Eigenharp Alpha at £4,950. The instrument uses 133 keys and two ribbon strips to play notes and samples and 'can play and record loops, change scale and key, transpose, alter tempo, program beats, create arrangements, switch and layer multiple sounds, all while the musician is performing live on stage'. We're talking about a completely new concept here, but several players have adopted the instrument, including Scanner (we saw one when we interviewed him). The less expensive Pico, at just £459, and the latest model Tau (£2,395) expand the concept. It's the only 'gadget' (and we feel as if we're demeaning it a little by calling it that) here that we haven't tested, so we'll make amends and get one in ASAP.

Price **£459 to £4590**
Contact **Eigenlabs on +44(0)1392 823000**
Web **www.eigenlabs.com**

WHATEVER HAPPENED TO...

airpiano

Not unlike a theremin in its touch-free action, the airpiano is played by way of a user gesturing above it, with keys and faders represented according to position above a wooden plank-like instrument complete with red LEDs. "We strongly believe that the airpiano introduces a new user experience, a magical and cool performance tool and an experimental instrument to explore," so says the website, but adds "airpiano was developed in Berlin, Germany by Omer Yosha. Unfortunately, the airpiano is now sold out and is no longer in production." **airpiano.de**



MOOG THEREMINI

It's a theremin, and a cool-looking 1950s-a-like one that is affordable and, as these things go, pretty versatile too. The unit is a little plasticky, but is a dream to use. For example, when changing scales you learn more about the subject in two minutes using the Theremini – simply playing it up and down within the categories – than in a previous lifetime of listening to music theory. All manner of parameters – some obvious, such as pitch and note range – are adjustable, and there's a good delay effect to add some variation. The 32 presets take it away, but not too far away, from that eerie whistle; in truth, most are of that ilk, and we could have done with some more varied tones to really experiment with. "For those after an easy intro to the world of the theremin and the other-worldly nature of its sound (it's also worth delving deeper into the fascinating life of its inventor if you have time), there's no easier way. An easy theremin to get into, both practically and financially. It's a little lightweight, but bang for buck this thing whistles like no other."

Price **£259**
Contact **Source Distribution on 020 8962 5080**
Web **www.moogmusic.com**

GADGETS HEAR AND NOW

And so, finally, to the very latest gadgets. Here are just some of the ones we've looked at most recently in *MusicTech*...





Price **£599**
Contact **ROLI on +44 (0)20 7254 2155**
Web **www.roli.com**

ROLI SEABOARD RISE

"We didn't expect to be quite as impressed with Rise as we are. It's a genuinely new piece of hardware that will expand your performance and recording creativity. It looks stunning and, with the Equator synth thrown in, can sound that way too." We needn't say any more about ROLI's new 'keyboard' here – you can probably gather that we think that this is a gadget well worth trying...

AKAI MPC TOUCH

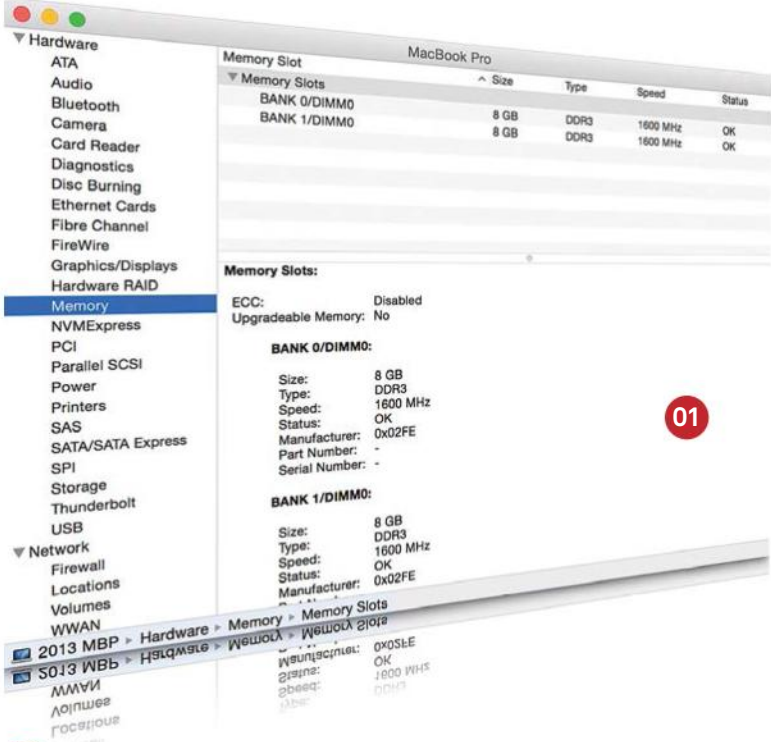
Akai is on a bit of a roll at the moment. For years, this company was on the cutting edge of both sampling and, with the MPC range, beat creation and performance. Then it went through a rather fallow period as sampling disappeared into the computer, but InMusic's acquisition of the brand has given it a new lease of life and we're now forever banging on about its Advance range of keyboards. So this new MPC – complete with a 7" colour multi-touch display – is another blinding piece of kit. It features 16 velocity-sensitive pads with RGB backlighting, a step sequencer with the touch interface, a phrase looper, XYFX real-time dynamic control, four other real-time controls and more than 20GB of content. It also acts as a 2-in/2-out interface, should you require it to. We are particularly excited about the Touch side of things, with the unit offering a new level of control over your waveforms. Akai says: "Literally grab and pinch waveforms, draw MIDI events, adjust envelopes, chop samples, add effects and precisely set your controls using your fingertips." Even more exciting is the fact that Akai could then bring this type of screen to the Advance range. Take note NI and Ableton. Something for the next version of Maschine and Push? MTF

Price **£499**
Contact **InMusic via website**
Web **www.akaipro.com**



20 Computer performance tips

If your trusty computer-based recording setup is struggling to process your tracks, don't despair, as there are lots of ways to help with frustrating glitches, latency, frozen screens or even crashes without breaking the bank...



Plenty of RAM will aid your computer's overall performance

01 UNDERSTAND ABOUT RAM

Modern audio software can be very memory-hungry and although the newer versions of Windows and Mac OSX have got much better at managing RAM, it's still one of the main performance bottlenecks on many peoples' setups. This is especially true if you use sample-based instruments, which generally have to be loaded into RAM to one extent or another. Buy a computer that's got plenty of RAM, or upgrade and expand what you already have, if possible – 4GB is an absolute minimum and 8GB or even 16GB is preferable. There's even an argument to be made for prioritising more RAM over spending money on a fractionally faster CPU, since this will offer you better performance in real world situations. Laptops, especially Apple models, have limited expansion capability and some are even non-user upgradeable so be aware of this if you're buying one to make music with.

Buy a computer that's got **plenty of RAM** or **upgrade and expand**

02 KNOW THE LIMITATIONS OF WI-FI

Although you can use wireless hard drives for storage, they don't offer quick enough performance to handle things like streaming Kontakt libraries efficiently. So as tempting as it is to offload your libraries onto a Network Attached Storage (NAS) drive, it's better to use a USB or other wired hard drive as it'll mean far less waiting when you want to call up an instrument.

03 USE SSD DRIVES WHERE POSSIBLE

Solid state hard drives (SSDs) have come down greatly in price, to the point where they're now feasible alternatives



03

It's **hard to overstate** how big a difference an SSD can make to **your life...**

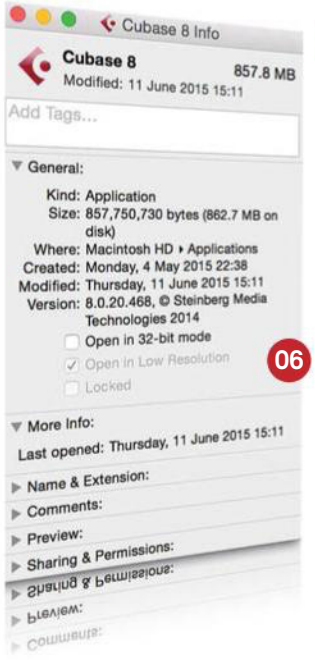
to the trad 'spinning' versions. It's true that SSDs still offer a less spectacular price-to-gigabyte ratio than conventional hard drives, but their performance is miles better and that's reason enough to use one as your boot drive. Store sizeable sample libraries on a conventional secondary drive, if possible – a terabyte-sized one will cost you very little. However, for your system drive, a 256 or 512GB SSD can be fairly affordable and you'll be astonished by the extent to which it speeds up your computer. Such system operations as app launching and memory access get much quicker with an SSD, meaning less waiting and more working. It's hard to overstate how big a difference an SSD can make to your life.

04 ONLY RUN WHAT YOU NEED

Any running application uses system resources, meaning anything that's 'open' but not in use is taking power away from your DAW. Don't have Photoshop or Excel open in the background if they aren't being used – just run the bare minimum you need, which will help with system performance.

05 PRACTICE GOOD HOUSEKEEPING

Being sensible about where you store your stuff will make the likelihood of confusion and data loss decrease. Don't keep a million files on your desktop. Set up custom folders and subfolders for DAW projects and back up regularly. This should also mean that you spend less time trying to locate 'missing' audio files.



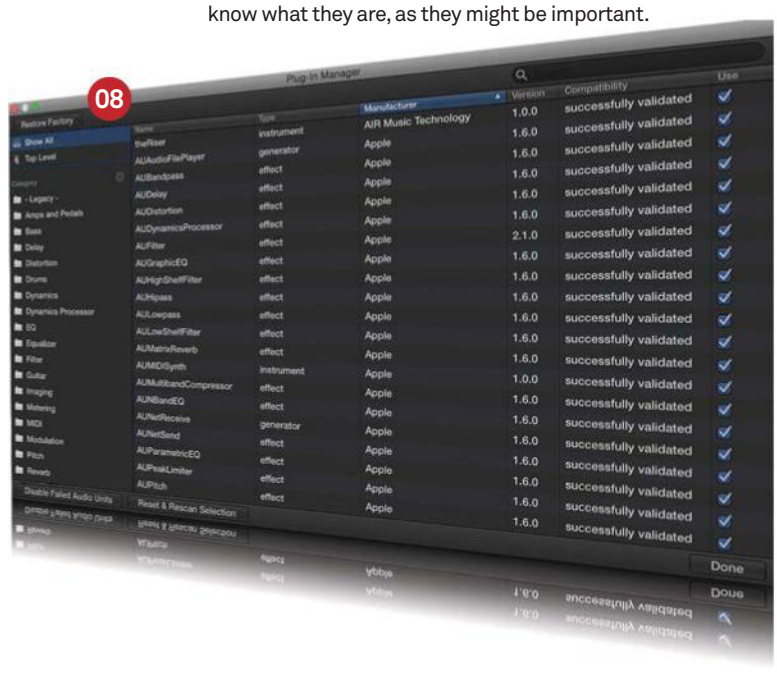
06

06 RUN IN 64-BIT MODE

Windows and OS X have been 64-bit native for some time now, and all modern desktop processors are as well. By now, the majority of DAWs are capable of running in 64-bit mode and some do so by default. You can check this on a Mac by performing 'Get Info' on an app and looking at its 'General' tab, or in the Activity Monitor. On Windows you can open the Task Manager > Details tab and choose to display the Platforms column, which will show you a list of what mode your apps are running in. The reason you should run in 64-bit mode is that it enables your apps and system to address large amounts of RAM which, in turn, improves performance and also lets apps execute certain processes more quickly. Most apps are 32-bit compatible so even where, for example, a plug-in is a little older, it should run fine but won't be able to take advantage of as much memory.

07 KEEP AN EYE ON WHAT YOU INSTALL

A lot of system slowdowns result from rogue or outdated drivers, and applications or plug-ins which are using resources, loading when they're not needed or just generally taking up space. That plug-in demo you installed which has since expired but flashes an error every time you launch Logic? That's costing you time. The control panel for a MIDI keyboard you've since sold which loads whenever you boot up your system? The same. Be ruthless, purge your system of stuff you don't need. Use uninstallers where available. On Windows use the Add/Remove software panel. On the Mac, if no uninstaller is provided, utilise something like Appzapper (**www.appzapper.com**) to get rid of junk. You want to streamline the startup of your computer and DAW, so prioritize deleting unwanted plug-ins, drivers and controller software. Of course, be careful to not delete things if you don't know what they are, as they might be important.



08

08 FAMILIARISE YOURSELF WITH THE PLUG-IN MANAGER

Most DAWs have a system for the activation of plug-ins and it's important to keep an eye on these if you experience any issues with your projects. In many cases it's no longer necessary to actually uninstall a plug-in, should you decide that you don't need it to be active, you can just tell your DAW not to load it on initial startup via the 'manager' window. However, if you have some plug-ins that you don't want to delete but also won't need to use for a while, consider deactivating them. This should speed up the startup of your DAW, simplify the plug-in 'chooser' menu and generally make life simpler. Some software now also allows you to create custom lists of plug-ins, which is another great timesaver since it stops you having to plough through endless lists of modules to find the one that you want.

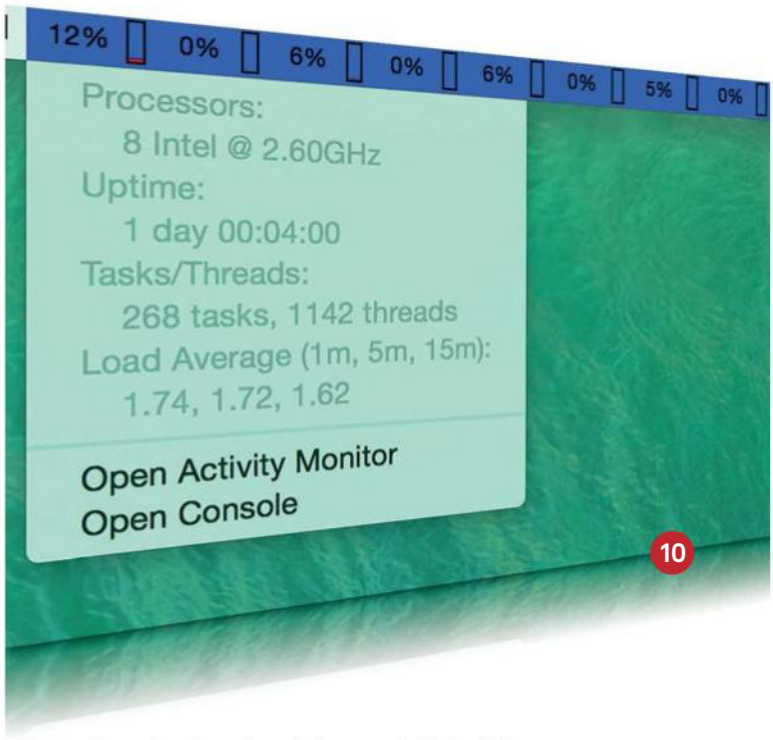
09 STAY UP TO DATE

This one is as old as the hills but it's still relevant. Generally speaking, updates to operating systems, DAWs, drivers and plug-ins make things better, faster and more stable. If someone never updates anything, or some things but not others, they don't get the benefit of everything working optimally. However, there are caveats. You should always read the system requirements of an update or a new version of →

Top: The performance of SSDs is miles better than that of 'spinning' drives

Above: Good plug-in management will reduce your DAW's startup time

Left: Improve app performance by running in 64-bit mode



software. If your hardware is at the lower end of what it'll run on, make a decision about whether you may actually lose performance by pushing it to do newer things. In the past, Microsoft has been guilty of releasing new versions of Windows which, for many users, made things worse – Windows Vista was the main culprit – but this seems to be less of an issue nowadays. Apple tends to be kinder to its older hardware and, periodically, will officially 'retire' certain models from support.

Some people have a **lax approach to 'housekeeping'** but it can be **a real barrier to performance…**

10 MONITOR SYSTEM RESOURCES

To understand what's going on inside your computer, use a system monitoring tool to look at CPU and RAM use, hard drive capacities and network performance. OS X has Activity Monitor and there are free apps like MenuMeters (**www.ragingmenace.com**). Windows has the Task Manager and many free apps, such as Rainmeter (**www.rainmeter.net**).

Above: System monitoring tools keep tabs on your computer's performance

Below: A drive that's free of clutter means smoother running

11 USE LIGHT VERSIONS OF PATCHES

Some sample-based instruments have 'light' versions of patches that use less RAM to load. They may have fewer velocity layers, for example, but sound very similar to the 'full fat' versions. Similarly, some synths and effects have different quality settings, numbers of voices and so on, and these can be switched in order to save power.

12 BE REALISTIC ABOUT WHEN TO UPGRADE

Every computer will eventually become obsolete. At some point you'll have to face the fact that your ten-year-old machine isn't going to be able to run big DAW projects any more. If component upgrades aren't a sensible option, bite the bullet and go for a new system, if you can afford it.

13 CONSIDER DSP HARDWARE

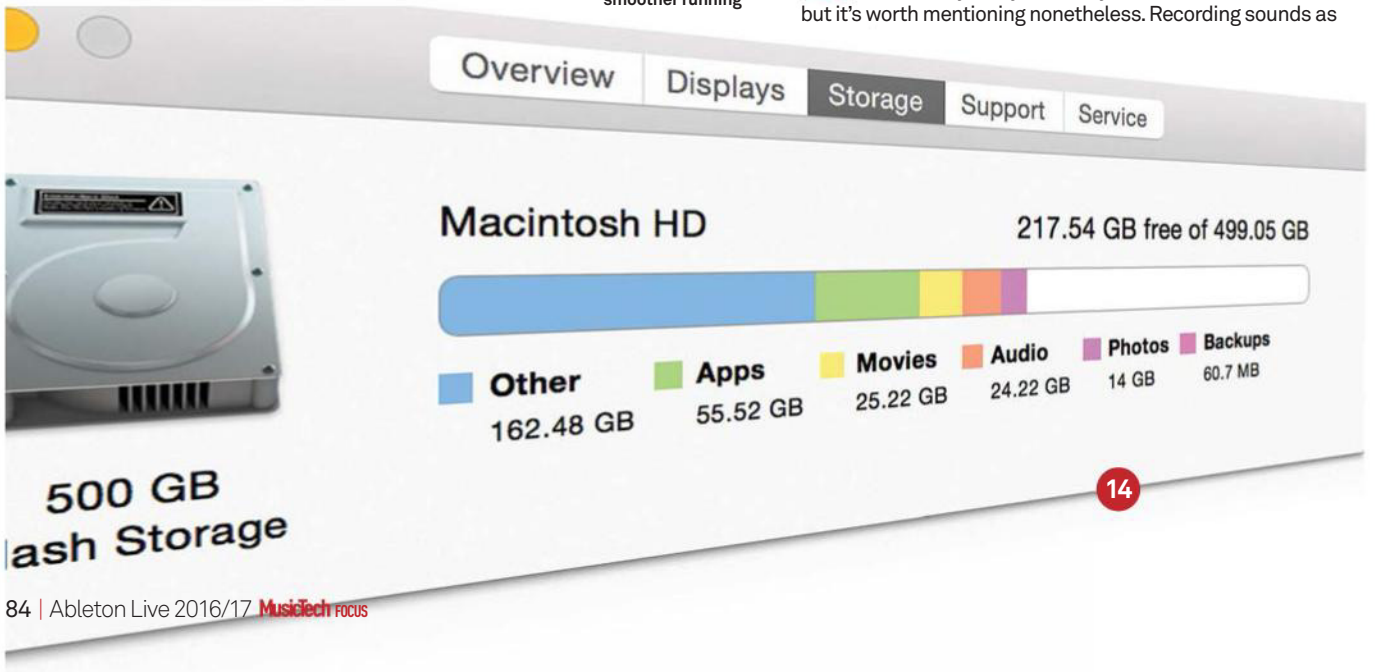
Although modern computers are very powerful, our expectations of what we should be able to achieve with them also rise constantly. Companies such as Universal Audio (**www.uaudio.com**) specialize in making specialized Digital Signal Processing (DSP) hardware add-ons for your computer which run high-end instruments and effect plug-ins independently of your CPU. These enable you to run very high numbers of these specific plug-ins without placing any strain on your host system, which remains free to deal with running your system and DAW. DSP solutions aren't usually cheap, but if you're serious about audio production they can improve the headroom on your setup massively. As a nice bonus, the plug-ins also tend to be extremely high quality and have found many high-profile fans around the world. There are fewer DSP companies around than there used to be but it's still an interesting arena.

14 CLEAR THE CLUTTER

Some people have a somewhat lax approach to 'housekeeping' but it can be a real barrier to performance. Understand that a hard drive which is rammed to bursting with movies or an iTunes library means, when you're tracking audio or loading instruments, your computer's having to search for free space much more intensively than it would if you had, say, 40% of the drive space free. This is much more of an issue with conventional spinning drives than with SSDs, which have a far faster seek time. A cluttered hard drive can also make simple tasks like browsing for files take much longer, and searches can be tedious, too, as lots of files appear which aren't actually relevant. It isn't that you necessarily have to throw stuff away, but do offload things like movie collections, which use tons of space, onto an external USB or even wireless hard drive. Space on your boot drive is precious and you should treat it as such.

15 TRACK RATHER THAN SYNTHESIZE

This one only really works if you have access to hardware but it's worth mentioning nonetheless. Recording sounds as



Mac OS X and Windows have options to help **prioritise performance** over, say, battery life, and these can be useful when you **need all the grunt you can get**

audio to your computer and playing them back uses far fewer system resources than generating them in realtime from a plug-in. For example, recording a line from a hardware synth as audio places much less strain on a CPU than generating it live. Similarly, passing audio through hardware effects uses virtually no power, but processing a track using the equivalent plug-in certainly does. A system that's dealing with 24 tracks of audio is under much less stress than one that has 24 synth and delay plug-ins loaded. Realistically, you might end up mixing live tracks with processed ones, based on what you've got access to. Remember that freezing or bouncing down tracks also frees up system resources, by creating audio versions of virtual tracks.



16 SCHEDULE BACKUPS

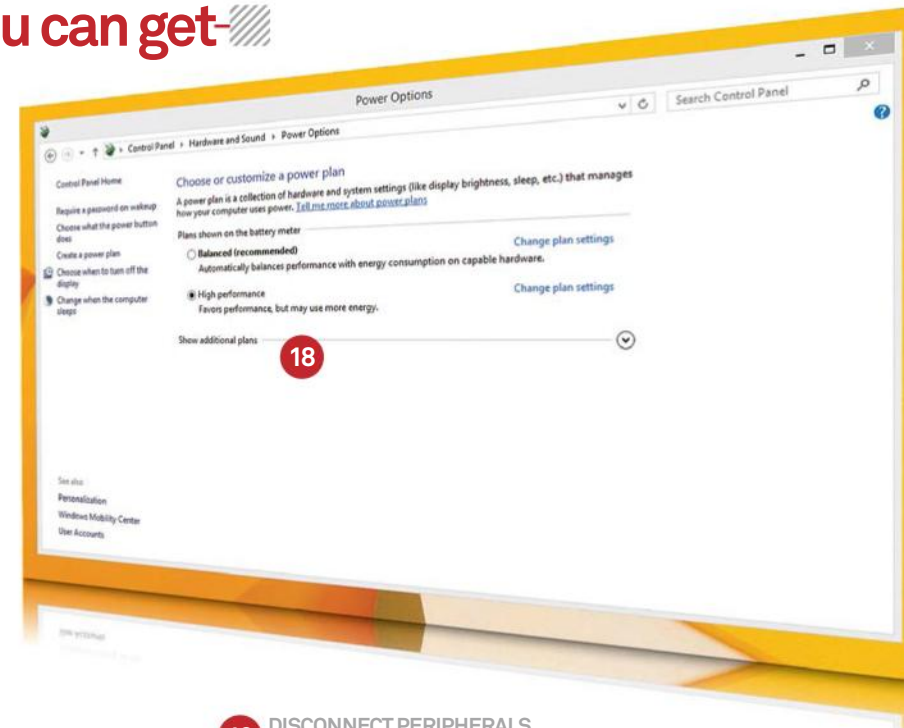
Backing up your system is essential but it shouldn't be kicking in when you're in the middle of tracking drums. Tweak your backup schedule to only start outside of working hours, or set it to manual rather than automatic. This should stop system slowdowns occurring when you least need them.

17 OVERCLOCK YOUR PC

Only do this if you're confident you're comfortable with it, but some PC systems can be 'overclocked' while remaining stable using freely available software hacks to increase the clock speed of the CPU. It can cause stability issues in some cases, though, so it may be one for the more techy user.

18 OPTIMISE THE SYSTEM FOR PERFORMANCE

Mac OS X and Windows actually have options to help you prioritise performance over, say, battery life or energy saving, and these can be useful when you need all the grunt you can get for running plug-ins or ensuring low latency during tracking. In OS X, look in System Preferences > Energy Saver and turn off the "Put hard disks to sleep when possible" option. Otherwise, OS X does most of its optimization automatically. Windows has more options and these are too numerous to list in detail here. Suffice to say, there are many online guides to optimizing Windows for performance, which include things like disabling animations and unwanted Windows Services, graphical flourishes and folder actions. There are several places to look and find options for freeing up more of your PC's resources for audio processing.



19 DISCONNECT PERIPHERALS

Having 20 things connected to your computer actually uses power, since each one has to talk to the system using a driver. Some peripherals also draw power from your machine. Cut it back to the bare minimum – an audio interface, MIDI device and maybe a dongle. Disconnect printers, scanners, webcams and the like.

20 USE FEWER SYNTH VOICES

Software synths often have configurable voicing options. More voices means a richer sound but also increased CPU usage. If your machine is struggling, consider using fewer voices as a way to save resources. Alternatively, you can always freeze or render down a virtual part. MTF

Above: Optimising your PC frees up more processing headroom

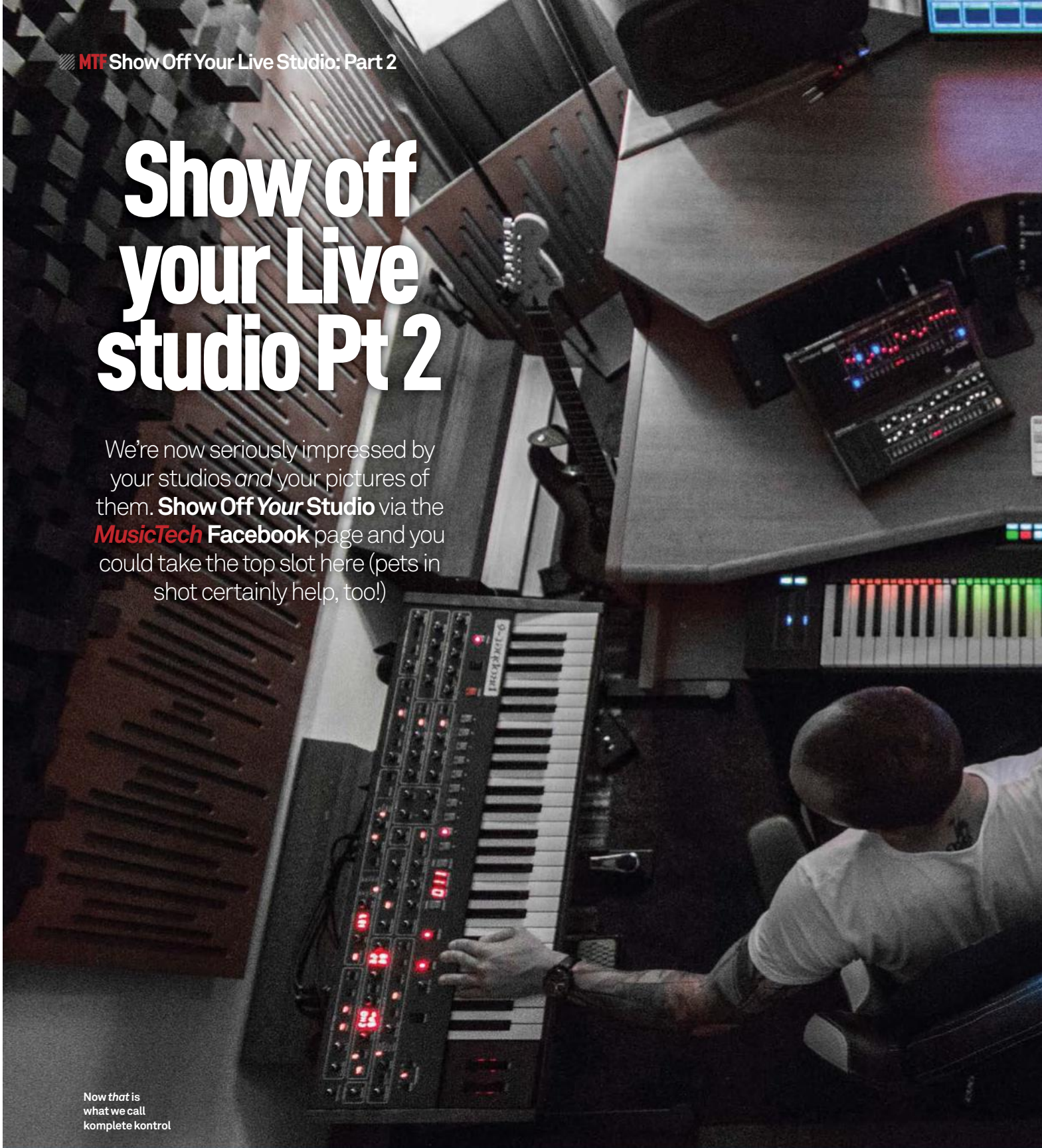
Left: Schedule backups so they take place outside 'working' hours

Below: Use fewer synth voices to gain more CPU power



Show off your Live studio Pt 2

We're now seriously impressed by your studios *and* your pictures of them. **Show Off *Your* Studio** via the ***MusicTech*** **Facebook** page and you could take the top slot here (pets in shot certainly help, too!)



Now *that* is what we call komplete kontrol

Driftmoon

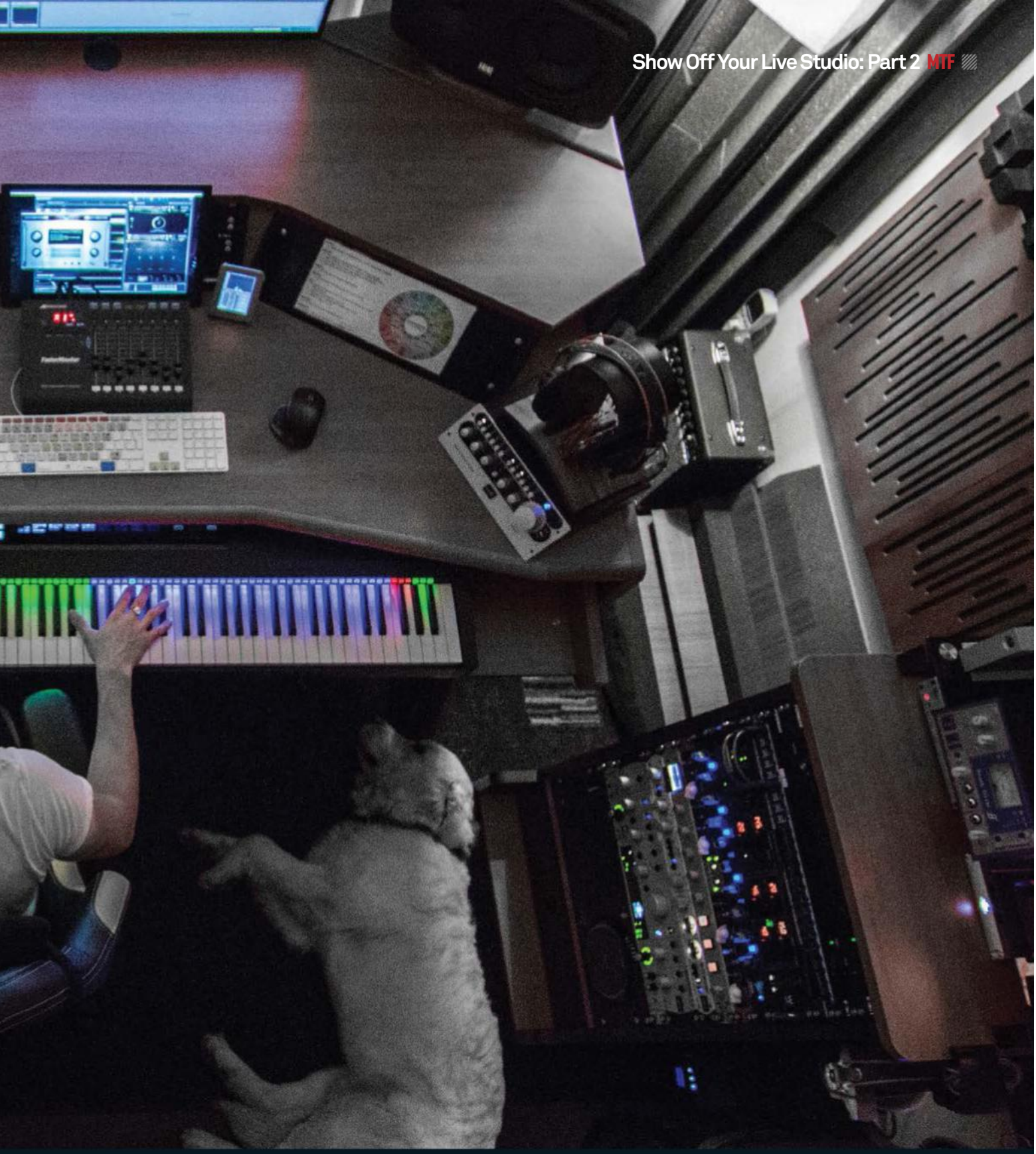
Contact: **driftmoonmusic@gmail.com**

MusicTech: **Tell us about your studio...**

Technically, my company is called Driftmoon Audio Suite, but the studio is called Double Helix Mastering. I use the studio 50 per cent for my own music composition under the Driftmoon alias and 50 per cent for mixing, producing and mastering for my clients. I didn't expect the studio to actually be so time-demanding, but it's proving to be an amazing experience. Working with so many different clients and musical genres has opened up a world of new possibilities to me.

MT: **Where, when and how?!**

I've totally refurbished and reconstructed my home to be a full-on studio. I know many people prefer to have a studio somewhere else, but I love the fact that when I have an idea for new track or melody, I can write it down immediately. Also, it's good to be available if clients need something to be done as



soon as humanly possible. The music scene is a rapid changing world, so it's good to stay ahead of any potential problems. Also, considering that I live in the heart of Prague right next to the most visited tourist locations, finding some place just for a studio setup would be extremely difficult.

MT: Kit list?
Mac Pro with 12-Core Xeon, 1TB SSD and 128GB RAM; Universal Audio Apollo Quad and UAD-2 Satellite Quad; Waves MaxxBCL; API 5500; Kush Clariphonic; SSL G Compresor; SPL M/S Master and MasterBay; Focusrite ISA One; Mutec MC-3+ Smart Clock; RTW TM3-Primus; 2x Event Opal; 2x Avantone Mixcubes; Roland JU-06 and JP-08; Dave Smith Instruments Sequential Prophet-6; Native Instruments Komplete Kontrol S88; software from Waves, UA.

MT: Which DAW?
For composing and production, I →

FIRST WITH REVIEWS
FIRST WITH NEWS
HAVE YOU CLICKED?

MusicTech net

Powered by MusicTech Magazine



www.musictech.net

Driftmoon (cont'd)

use Ableton Live. I somehow love the workflow and even for orchestral composition, it's absolutely amazing. Granted, it doesn't have as many features as, for example, Cubase when it comes to composing, but you can work around them and find your own 'way' of doing things. For mixing, I use Pro Tools and for mastering, Sequoia.

MT: What is your favourite gear?
I recently purchased a Dave Smith Prophet-6 and, oh my, it's absolutely phenomenal! I've spent days playing with the sounds and writing them down in an Excel sheet to categorise them and then I immediately started reworking my album tracks, because the sound of the Prophet is amazing. It just stands out.

MT: How often are you in your studio? I am here daily for 10 to 16 hours – it depends on how much work there is to be done. I stopped having weekends years ago, now every day is a work day, but every day is amazing and different. I somehow got used to non-linearity and it's making every day interesting.

MT: How do you use it?
My studio space consists of 'only' one room, so there isn't enough space to record a whole band: nevertheless I can record singers and instrumentalists with ease. I mostly use the studio for producing, composing, mixing and mastering.

MT: Next on your shopping list?
Right until now, I've had no need to go into the 500 Series world. But after loads of positive reviews about Meris' Mercury7 Reverb, I think I'll start collecting different 500 Series modules. I've also tested the Moog Ladder Filter module, it's incredible.

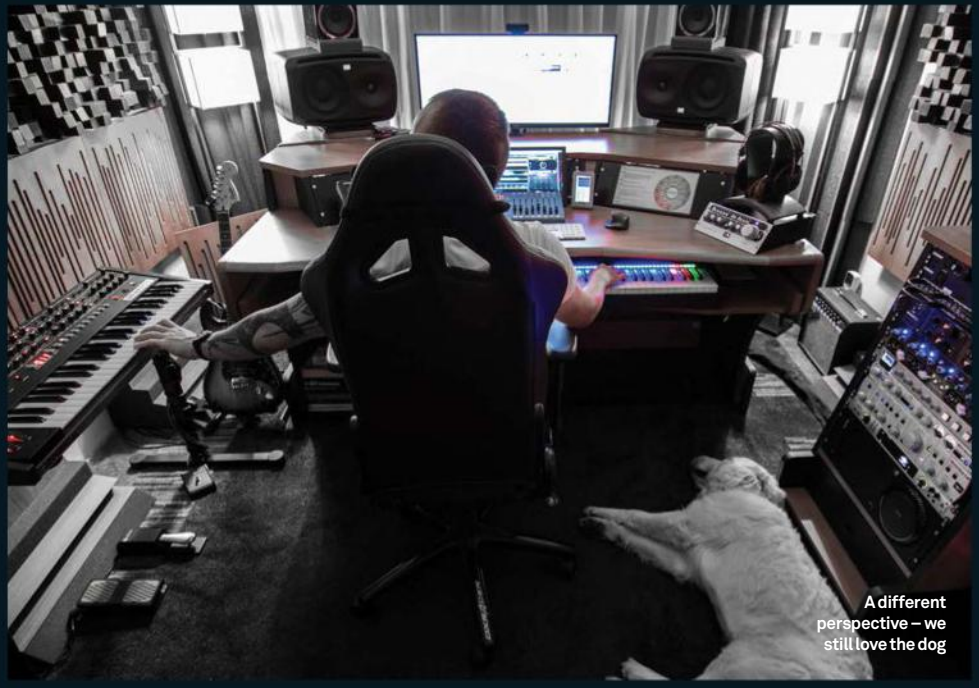
MT: Does anything annoy you about your studio setup?
Loads of time and effort (not to mention money) went into this space. I've had room to work with and I made it work. The room itself is treated to industry standards. If I could change one thing, it would be the air conditioning. However, because the studio is located in a very old building which is under UNESCO protection, I can't have an AC unit here.

MT: Any dream gear?
Barefoot monitors, I really hope one day I will be able to get a proper space which will be treated so I could buy and use these. So far, they are huge for my current space but it never hurts to dream, right?

MT: Do you have any advice for MT readers about setting up a studio?
To try and remember that we all started with something little. Whether or not you will make something great out of it depends only on you. Listen to as much music (in different genres) as possible and also to be kind and respectful to other musicians.



Driftmoon's outboard gear is impressive, particularly the SPL gear


A different perspective – we still love the dog

Read MusicTech on any device, anytime!



Download your FREE app today

FREE PREVIEW ISSUE

SINGLE ISSUES JUST £2.99 / $3.99

SUBSCRIBE FROM JUST £2.49 / $2.99 AN ISSUE

Search **MusicTech magazine** and download your FREE app today

Available on the App Store | Available on the Google play | Available on kindle fire | Available on pocketmags.com

Francesco Mulassano

Email: francesco.mulassano@gmail.com

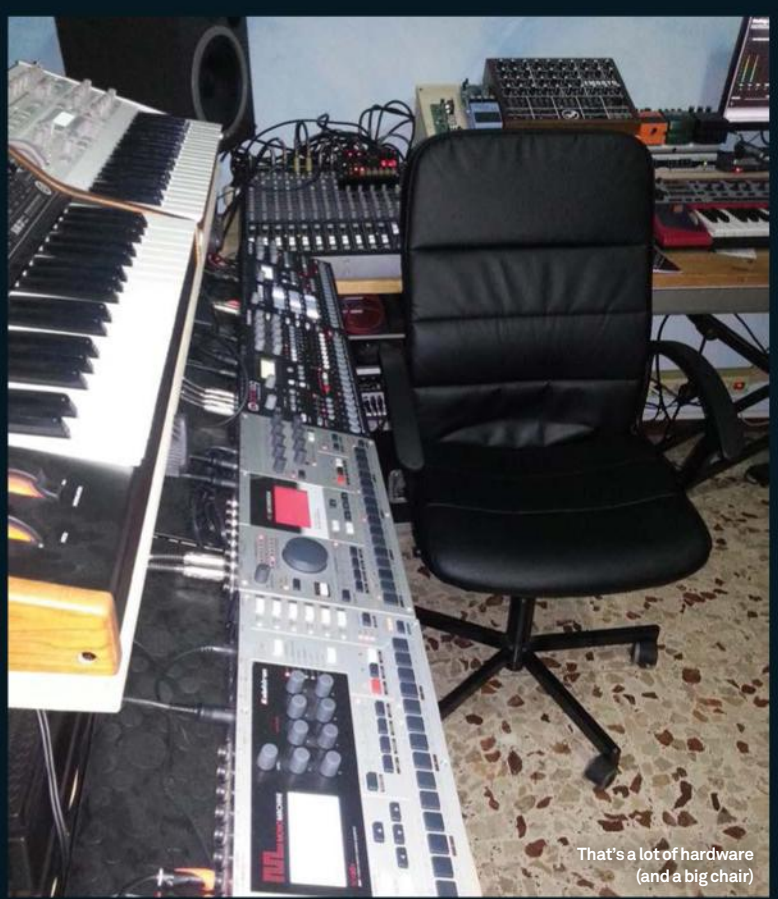
MT: Tell us the key components of your studio?
I have collected instruments for at least 25 years. I started with guitars and had a rock and metal influence. As a synth addict, I have more synthesisers than outboard. At this time, I have all the Elektron gear (minus the Octatrack), a Moog SUB 37, Access Virus TI2, Korg MS-20 (the original), a monophonic JEN SX1000 from the golden Italian synth era, one Nord Modular G2, the first Novation Bass Station Key, a beloved Roland JP-8000, Teenage Engineering OP-1 (the first one in Italy), a pair of Axoloti, Waldorf Streichfett, some custom and DIY machines, a pair of controllers (Push 2 and BeatStep Pro), a lot of software plug-ins (all original, of course). I think that's all! From the 'metal' period, only the hair and some guitars have remained.

Which DAW do you use and why?
When I switched from Windows to Mac, eight years ago, I chose Ableton Live and that's now the only DAW I use. It took a long time to go from Cubase (Win) to Ableton because the way of working is completely different, but I wouldn't go back. You can do everything with the tools offered by Ableton – it's very complete – and then with the arrival of Push and Push 2, it has become more useful and cooler.

What is your favourite gear and why? For me, it is the Access Virus TI2. I owned a Polar TI and a Polar TI2 at the same time – a very crazy time. Then I changed my TI for the G2. Access is one of the few brands that remains an investment, year after year.

What's next on your gear wish list? A Waldorf Q is one of the first on the list, but I think the next one will be the new Korg Minilogue.

How do you use your setup? My studio is located inside my house, here where I work on my solo project Urbanspaceman, and on the preproduction for other two projects: Space Aliens From Outer Space (a synth trio with alien costumes inspired by 50/60s science fiction), and Giants (a trip-hop duo with an incredible female singer).



That's a lot of hardware (and a big chair)



Got enough keyboards there, son?

What is your dream piece of gear? There is no dream synth, but if I have to choose, the Hartmann Neuron is one of the pieces of gear I've always looked at with some admiration.

Do you have any advice for MusicTech readers about setting up a studio? If you're starting from scratch, look at YouTube, talk with other musicians and focus on what you want to produce. Then read the manuals/reviews, and choose one machine only and spend all your available time on it.

Any studio anecdotes? I could tell you about when the tenant upstairs broke a water pipe and it flooded. Fortunately, though, the synths stayed dry... MTF

The 12 Best Soft Synths Money Can Buy

Fancy some synth heavy action in your Live sets? Well here's our very own rundown of the best 12 soft synths available to buy right now...

Rules

1. Only one synth per company
2. Has to (still be) available to buy separately from any DAW or bundle
3. Not freeware (that's a separate list)
4. Available to run in 64-bit
5. Why 12? Why not?
6. Vote to change this chart at www.musictech.net

Lennar Digital Sylenth1



Details
Price €139 plus tax
Developer Lennar Digital
Contact via website
Web www.lennardigital.com

Sylenth1 has been around for longer than pretty much every other synth in our Top 12 and has become a legendary synth in the world of plug-ins. Its developer is fairly quiet and updates are rare, with the 64-bit version only arriving last September, but it was well worth the wait. Sylenth1 might be old, but this four-oscillator, two-filter analogue-style synth has launched more dance tracks than most and has a sound that can be the backbone to any track in many styles. Thanks to easy-to-use effects, massive polyphony and an easy ability to tweak and refine the onboard sounds – of which there are over 1,300 – this synth was, as they say, ***designed to perform. And it certainly still does just that...***

U-He Diva

It was a close call between Hive – another great U-He synth ideal for more cutting-edge electronic music productions – and Diva, a more analogue-based synth. But this synth gets it, especially as it has a vintage synth and modular philosophy – of course, so much in vogue at the moment. You can mix and match analogue modules for some great combinations and choose from well over 1,200 presets. When we reviewed the software, we said: ***"Diva represents the current pinnacle of analogue-modelled sound. An essential synth."***



Details
Price £109 (upgrade from Blue £35)
Contact Time+Space 01837 55200
Web www.timespace.com



Details
Price $129 to $379
Contact +90 312 265 0558
Web www.synthmaster.com

KV331 Synth Master

SynthMaster – just updated to v2.8 – makes our dozen simply as it claims to be something for everyone and is one of the few synths around (next to Blue II) that actually delivers. At its heart is a synth engine (or engines) capable of a huge variety of synthesis types including additive, wavetable and vector and a huge number of modulation options. But its real beauty lies in the fact that these are all very easy to utilise. Routing is graphical and straightforward; there are more than enough presets (we recommend going for the 'Everything Bundle' for around 3,000 sounds) and you can really use the synth on whatever level you feel comfortable with. "It really does an incredible amount in a no-nonsense, almost calm way. So sit back, don't panic and enjoy the ride. ***It does what it says. It truly is a master of synths.***



Rob Papen Blue II

Details
Price £109 (upgrade from Blue £35)
Contact
Time+Space
01837 55200
Web
www.timespace.com

If there was an award for throwing synthesis engines, power and presets into one plug-in, Rob Papen could well walk away with it. Blue II is an update to Blue, released back in 2005 – but what an update. It's still a 'Cross Fusion' synth, but boasts subtractive, FM, phase distortion and wave-shaping synthesis; six oscillators; a couple of filters with 27 different filter types; a huge set of effects processors (each offering 35 FX types); and over 2,000 presets. So, quite a bit then… We said: "the synth moves from the late-80s through the 90s, noughties and very much more up-to-date, with some absolutely huge presets that would easily carry a tune from any era. ***Blue II is any synth you want it to be and a lot more."***

"Featuring **20GB of extra content** plus **400DSP waveforms** and no less than **12,000 sounds**, this could well be **the only synth you'll ever need**…"

Spectrasonics Omnisphere 2



Details
Price £285
Contact
Spectrasonics
Web
www.spectrasonics.net

Spectrasonics certainly didn't rush version 2 of this multi-award winning synth, and while the update very much keeps the best of what was good about the original, it also most definitely updates the concept. Featuring an extra 20GB download plus 400DSP waveforms and no less that 12,000 sounds to choose from, this could be the only synth you'll ever need – you'll certainly spend a large chunk of your life exploring just the sounds, let alone what you can do with them. We said: "While some have got close to Spectrasonics' greatness, there's little doubt that Omnisphere 2 will set a new gold standard for the next few years. ***It oozes class and sophistication and is an essential purchase for all."***



NI FM8

Details
Price £169
Contact
NI, via website
Web
www.native-instruments.com

It's fair to say that we could have included many NI synths in this top 12, but our own rules state that it's one synth per company, so FM8 just about snatches it from Massive. The reason? Well it's another synth that, like Oddity 2, was originally based on a classic hardware synth (Yamaha's DX7 in this case) but through the magic (aka no limits) of software, has taken on a life of its own. FM8 moves on from the original FM7 (but still includes its sounds) and well beyond the percussive, metallic and shimmering pad sounds that FM synthesis was famous for, to result in a synth that simply shines. Programming FM is now relatively easy (well, compared to the original synth, what isn't?) and sound selection is very simple. The resulting synth is big, evolving, and hugely atmospheric. ***FM8 is not Massive and is quite old, but it's still a stunning synth and one we use more than most.*** →

Linplug Spectral

Parts of Linplug's Spectral will make you think it's an out-and-out dance synth; some of the presets without a doubt do lend themselves to many dance genres. But there's both a simplicity and depth to it that allows you to take this synth pretty much anywhere you want to. We particularly love the movement you can get with sounds and how easy it is to program this via the onboard arpeggiator, but there are other highlights, too. The waveform Spectral editor is obviously intended to be one of these, as it lends its name to the synth, and here you can get to the heart of the sound to shape the harmonics. So it's as simple as you like on one level and as complex as you like on another. Dance synth? Maybe, but also far more than that. We concluded: ***"Spectral offers inspiration, loops aplenty, core wave editing and a great atmosphere…"***

Details
Price $149/€119
Developer Linplug
Contact via website
Web www.linplug.com



Why didn't you include…

1. Reaktor? Yes, it could have been in the Top 12, but it's more of a development platform than standalone synth (as good as it is)
2. Crystal – it's free! Because it's free and we'll do a freeware list separately
3. Pow3r 3Z by Anonymous Software Ltd, or some such other title? Because this is a list of titles *we have reviewed*. By all means vote for your favourites at www.musictech.net or via our Facebook page…

Details
Price £209, £89.95 (upgrade)
Contact Time+Space 01837 55200
Web www.timespace.com



"The **true joy of Iris** lies in transforming **your own sounds** into playable **music, melodies and textures**…"

iZotope Iris

Certainly one of the more unusual synths in this roundup, as it is more of a sampling resynthesiser, using audio samples as waveforms. It works by selecting parts of a sound from either a wave or spectrum, utilising iZotope's groundbreaking technology in this area – essentially the same type of code used in the high-end RX software – so you can home in on a particular part of a sound. The bottom line is not complicated at all, though: you can get some breathtaking sounds, complex as you like – and also sounds you won't get anywhere else. When we reviewed version 2 recently, we said: "The true joy is in capturing your own sounds and transforming them into playable music, melodies and textures; a task now made even more flexible with the new modulation system. ***Still a fun and incredibly creative tool to use."***

Fabfilter Twin 2

Details
Price €99
Contact Fabfilter, via website
Web www.fabfilter.com



Fabfilter makes some very special outboard and effect plug-ins – indeed, we've scored pretty much all of them a 9 or 10/10 over recent years – but before the company made any of these, it made this spectacular synth plug-in which still stands up very well today. Featuring some amazing (and very electronic and dance-based) presets, this is a virtual analogue that still cuts through any mix and still slots into many genres. Easy tweaking, routing and editing on the front panel and a sound that really works on many levels: ***one that will probably never go out of fashion. An amazing synth…***

GForce Oddity 2



Details
Price £139.99 (currently £58.33 in a sale)
Developer GForce
Contact via website
Web www.gforcesoftware.com

It was originally one of the best emulations of an ARP Odyssey you could buy in software, but Oddity 2 is now so extra-featured and powerful that it has taken on 'life after emulation' status and become an incredible-sounding synth in its own right. This is not least thanks to its ability to play in a Polyphonic Mode, meaning you have the exceptional sound of the original Odyssey multiplied into huge walls of spikey, aggressive and lush tones. There's unpredictability and a vast range of timbres across 1,000-plus presets and it's easy to understand why the original was so beloved by so many synth icons. We said in our original review: "Oddity 2 is a triumph. We love the extra mod options and using it as an effect, but best of all is Polyphonic Mode, a feature that takes Oddity to new heights. ***If ARP had carried on as a company, this is the synth it would have made now."***

We want more emulations!

We've deliberately stayed away from soft synths that specifically model hardware synths. We (mostly) favour software being used to stretch what can be done with sound rather than emulating existing hardware. The ones we have included – namely Oddity and FM8 – have, for example, extended what was done by the original hardware so much that they have become synths in their own right, but don't worry, classic-synth fans – we'll be doing a proper emulation chart full of Arturia, Roland *et al* very soon!

Details
Price €199
Contact MeldaProduction, via website
Web www.meldaproduction.com



> **"Oddity 2** has the exceptional sound of the **Odyssey** multiplied into **huge walls of spikey, aggressive and lush tones"**

Zero-G Epica

Okay, strictly speaking, it's a library of sounds and not technically a soft synth like the others in our dirty dozen, but it makes synth noises and has synth controls so we're including it, dammit! And, boy, what synth noises it makes. Epica runs as a Kontakt instrument with all of the library management that that entails. Once loaded in, though, you won't regret the hard drive space outlay because this thing sounds, well, epic. Huge sounds and enough control over them for it not to get unwieldy (and to label it a synth!). We said: "If you love synths, this could be the best £113 you will spend. ***As epic as epic can be…"***



Details
Price £113
Contact Time+Space 01837 55200
Web www.timespace.com

Melda MPower Synth

MeldaProduction might be known to you by its range of plug-ins, for which it has a great reputation among producers. MPower is its first synth and the company seems to have concentrated on the 'power' part of the name. MPower has spades of creative options and you can create sounds or take the existing presets on all sorts of journeys with them. There are 1,500 presets but, as we said in our review, it might be best to create your own to really get the best from them, as there's so much power under the hood, it would be a shame not to use it. We said: "Spend some time digging around and you will find that there is much to like. It's a serious synth that rewards some investment of effort, but, crucially, it sounds great. ***An extraordinarily powerful synth, with literally endless programmability."***

CHOOSE YOUR SUBSCRIPTION PACKAGE



THE BUNDLE SUBSCRIPTION...

- First 3 issues for £3 – print and digital editions*
- Just £20.20 every 3 issues after the trial
- Save 25% after the trial on Direct Debit
- Free UK delivery direct to your door
- Access to the app through iTunes, Kindle Fire and Android

CODE - SSDPS16B



THE PRINT SUBSCRIPTION...

- First 3 print issues for £3*
- Just £13.45 every 3 issues after the trial
- Save 25% after the trial on Direct Debit
- Free UK delivery direct to your door

CODE - SSDPS16

OVERSEAS OFFERS



PRINT + DIGITAL...

- **Europe** – 3 issues for €8 then €38.60 every 3 issues, saving 25%
- **Australia** – 3 issues for $15 then $54.75 every 3 issues, saving 15%
- **Rest of World** – 3 issues for £8 then £45 every 3 issues



PRINT ONLY...

- **Europe** – 3 issues for €8 then €28.50 every 3 issues, saving 25%
- **Australia** – 3 issues for $15 then $38.20 every 3 issues, saving 15%
- **USA** – SEE PAGE 109
- **Rest of World** – 3 issues for £8 then £20 every 3 issues

MusicTech SUMMER SALE!

3 issues for just £3!*



TWO EASY WAYS TO SUBSCRIBE

anthem.subscribeonline.co.uk/musictech

ENTER **SSDPS16B** (bundle) OR **SSDPS16** (print only) in the promo code box

0844 844 0398*

*Calls cost 7 pence per minute plus your phone company's access charge

(*Overseas readers* +44 1795 592849)

QUOTING CODE **SSDPS16B** (bundle) OR **SSDPS16** (print only)

OFFER ENDS **30 SEPTEMBER 2016** CODE **SSDPS16B / SSDPS16**

Your subscription will start with the next available issue. There are 12 issues in a year.

*UK-only saving. Get your first 3 issues for just £3 then continue on subscription at £13.45 every 3 issues unless you cancel. The print saving is calculated against the cover price x 12 issues. The bundle saving is based on buying 12 print copies in a shop plus 12 digital single copies. Details of how to access your digital editions will be sent to you with acknowledgement of your subscription order. Please provide your email address for this.

MTF Lead Review

Hardware ▨ Software ▨ Mobile tech ▨ Accessories



GOTHIC INSTRUMENTS

Dronar Hybrid Module

With Gothic Instruments' new title, we may well have a build-your-own software – not hardware – instrument. **Andy Jones** drones on. And on…

Details

Product
Dronar Hybrid Module
Price **£59.95**
Contact **via website**
Web
www.timespace.com

Features

- Atmospheric Kontakt instrument
- Multi-timbral synth/sampler with multiple arpeggiators
- 6 main controls
- 300 presets
- 16GB of data compressed to 8GB
- Produced by Hollywood trailer sound designer Alessandro Camnasio
- Live strings by players from the Royal Liverpool Philharmonic Orchestra

Every aspect of music production is being made more accessible with technology: from creating the perfect dancefloor beats to easy pro mastering. However, in more recent times, the target audience for software developers seems to have shifted to the soundtrack composer – or wannabe soundtrack composer – so we've had many libraries and instruments aimed specifically at them.

I've looked at a stack of these over the last couple of years, including fantasy favourites from Eduardo Tarilonte (Shevannai and Era) and sci-fi blockbusters from Heavyocity (Gravity) and Wide Blue Sound (Orbit).

While all very different, these collections do have one thing in common: atmosphere. They will help transport your music making to another place or time, so are ideal tools for the soundtrack composer. With the blockbuster soundtrack being so high-profile, and pop careers being so low-cash, everybody wants to be the next Hans Zimmer rather than the next Eurovision entry, so collections like these are becoming almost as ubiquitous as analogue synths.

Dronar Hybrid Module is the first release from the superbly named Gothic Instruments. It's an instrument aimed at the soundtrack artist but also one designed to stand out from the rest.

> ▨ Everyone wants to be **the next Hans Zimmer** rather than the next **Eurovision entry**… ▨

Modular software

The first reason for this is its modular approach. The instrument is being released in blocks of downloads, hence the low price, which is an interesting approach (we love modular synths, so maybe we're seeing the start of DIY software – see the box below for info). Secondly, Dronar is setting its sights on

The first of many?

The reason we're going to town on Dronar is terms of a lengthier review is that it purports to be the first of several modules under the Dronar moniker. This collection/instrument is called 'The Hybrid Module' and if you buy into it for the not unreasonable sum of £59.95, you'll get further downloads and additions cheaper. The idea is that these are released every couple of months, as Gothic Instruments head honcho Dan Graham explains: "We realised that it would take two years to create all the audio content we wanted, so we thought – why not release the project in modules? This means getting a new module out every two months until finally we can bundle it all together into a giant Master Dronar in a year or so. Until then, this approach means users don't have to wait until 2017, and also each module is quite attractively priced, making it a good chance to try it without breaking the bank."

The Expert tab

On a basic level, Dronar is very, well, basic, with six controls for big dramatic changes. Hit the Expert tab, though, and you get access to around 30 extra parameters as drop-down menus for the main controls, plus other controls like attack and decay.



The Arp tab

Like the Expert tab, the Arp tab allows you access to more parameters from the top-level controls. And like the Expert tab, you don't really need to be an expert! Here, you can adjust the Intensity, filter and arpeggiator steps and rate.



Master FX

Here, edit your effects on a deeper level with distortion and chorus (shown in this picture). With reverb, you get a drop-down menu, too.



specific soundtracks, not more general scores. Clearly all about dark, twisted atmosphere (the company's called Gothic Instruments, after all) and it's pretty obvious what you're getting. If you're doing the soundtrack to *Dumb And Dumber 3* or *Meet The Fockers 17*, you might want to turn the page now.

The library installs pretty much like any other Kontakt library, although it's the type where you don't get the fancy NI graphics – you load instruments and presets via the more standard Kontakt finder interface. The 20 categories have self-explanatory names like Alien, Low End, Pads and Chaos, plus 300 preset sounds spread between them. Auditioning is as easy as you'd expect with it being Kontakt-based, although load times are occasionally a little long.

The first few categories of Alien, Atmospheres and Dark are exactly that. You may feel slightly underwhelmed by 'otherworldliness' – it's a tad samey. But the strength is the simplicity and how a few controls on Dronar can do so much.

The producers of Dronar deliberately held back on the number of controls just to see how much control they could get from a simple set-up, and I think it's one of Dronar's main strengths. You get a couple of effects (which can be varied) plus controls for Hi, Mid and Low, which vary actual sounds within those EQ ranges rather than just the frequencies. Then there's an Intensity dial linked to your mod wheel and a Movement dial, and that is pretty much it. One or two of these controls will easily take your sound away from what you start with, and very quickly, too. So if you do think they're a bit, well, 'droney', then get hands-on – it's easy.

You can dig deeper, of course, but you'll get a good idea of the scope of Dronar with just the top level. Step down into the 'Expert' page and you can define which of 30-plus parameters can be affected by the main controls (you don't really need to be an 'expert' to be here to be fair), and you can also delve deeper into the arpgeggiation and LFO features with their own dedicated pages, too.

The later sound categories are much more musical than earlier ones, which perhaps emphasise the 'drone' part of Dronar. The strings are superb, the organs suitably dramatic and the pads as creepy and menacing as you'd expect. In fact, I'll leave it to three more category names to sum up the rest: Fear, Tension and Dark. So, if you want to give it some kind of Hans Zimmer 'rating', Dronar is definitely more *Interstellar* than *Going For Gold* [*clever Zimmer reference there for UK readers of a certain age* – Ed].

If you're doing the soundtrack to *Dumb & Dumber 3*, you might want to turn the page now...

Alternatives

I've already hinted at the collections and instruments I've used which are similar to Dronar, including Wide Blue Sound's Orbit ($150 www.widebluesound.com) and Heavyocity's Gravity ($449, www.heavyocity.com), both big, evolving sound makers and manglers, perhaps without the total darkness that this has. Elsewhere in *MusicTech*, we've also looked at Output's Signal (£150, www.outputsounds.com) which is an incredible instrument for drones and other atmospheres.

Conclusion

The Hybrid Module – the first Dronar release – didn't surprise me in what it could do in terms of atmosphere and tension, and it very much delivers its menace in spades. As an opening module of possibly many, it is very good indeed. Initially, you might think it's a bit of a one-trick pony, but there are enough strings and melodic folders to take it on more musical journeys, and it also features so many parameters just under the skin that you can easily move onto newer things.

The Movement and Intensity dials alone will have you off exploring new sonic paths and there is more than enough content to experiment with... And that may well prove to be the downfall for Gothic Instruments' modular plans – there might just be enough in this pack to suffice! Either way, with just £60 needed to start your dark and atmospheric journey, I think it's one well worth taking. **MTF**

MTF Verdict

+ Huge amount of atmosphere for the money
+ Modular upgrade path could be interesting
+ Lovely big and easy controls make dramatic changes easy
+ Lots to explore and very easy to do it (and you will!)

- It's perhaps a little samey in places, but what did you expect?
- Some hanging issues in Logic

The Hybrid Module is a great opening chapter in the Dronar series. Beautifully put together and flexes within its own constraints well, with very easy controls and more available if you want them.

9/10









Details

Price
€75 each (inc postage)
Contact
via website
Web
www.teenageengineering.com

Key features

Shared
- 16-part step sequencer with 16 patterns
- Parameter locks on sequencer
- Built-in speaker
- 3.5mm audio out/in sync
- Jam sync
- Folding stand
- Step Multiplier

PO-20 Arcade
- Beat making and chiptune improv
- 16 synthesised arcade sounds
- 128-chord and pattern chaining
- 16 punch-in effects

PO-24 Office
- Noise percussion drum machine and sequencer
- Sampled vintage h/ware and real synth engines
- 16 sounds
- Solo control
- 128-pattern chaining
- 16 punch-in effects

PO-28 Robot
- Live synth and sequencer,
- 8-bit synth engine
- 15 sounds and micro drum
- 128-pattern chaining
- Live play and sequencer

TEENAGE ENGINEERING

Pocket Operator 20 Series

Three new Pocket Operators from Teenage Engineering. Can they possibly be as good as last year's hat-trick? **Andy Jones** goes 8-bit…

Last year's NAMM show – that is to say NAMM 2015 – was stolen by Teenage Engineering, with its Pocket Operator Series. Three calculator-style sound modules for beats, sub bass and leads had everyone crowding around the TE stand and while the big companies made big announcements, three sub-€70 Euro gadgets walked away with the honours. A couple of months later, they walked off with just about every *MusicTech* award I could bestow on them, too…

Yes, the original Pocket Operators – PO-12 Rhythm, PO-14 Sub and PO-16 Factory – were (and indeed still are) incredibly fun and big-sounding melody and beat makers. Sync them together for complete tunes, admire the fancy and fantastical LCD screens, marvel at how such a small device can deliver such a big sound… They were great. So for Teenage Engineering, this year's NAMM was always going to be a case of, 'Well, how do we follow that?'

So how *do* they follow that?

The answer, of course, is with something as cool, something as geeky and something as futuristic as the first three POs – a new trio of POs known as the PO-20 Series, comprising the PO-20 Arcade, PO-24 Office and PO-28 Robot.

The names give a little away about where each of the new series is heading sonically, but it's all perhaps a little less obvious than the PO-10 Series, each of which effectively offered beats, bass, and lead sounds (actually, it was a little more complex than that, but that will do for now). So here, the Arcade focuses on beats and chip tunes; the Office on percussion and sequencing and the Robot on synth and sequencer. Three new strands of PO, then, and each promising a tantalising tangent away from the original three, already innovative, devices.

> For TE, this **year's NAMM** was always going to be a case of **'how do we follow that?'**

Shared features

Time for a quick overview of the main ways of using the POs before I get specific. With all three, it's a similar scenario and less than a year after reviewing the 10s, I am on familiar ground. So the first thing you'll do after slotting in the batteries in the 'case' around the back – actually a couple of hard wires, and I had to push the connectors to make sure they connected with the batteries on the first unit I tested – is hit the 'Play' button to hear the onboard patterns. Press the Pattern key and you will be able to choose your pattern from the 16 number keys. Hit the Sound button and, similarly, you can play one of the onboard sounds, again by hitting a number key. The two dials here typically alter volume and an effect, like filter, per sound. So you can adjust the pitch of the sound you want to record at this stage and then you can record it, either by step sequencer or by playing live over a pattern as it plays.

The former recording method is done by holding the Write button down (along with Chord) and then inputting notes of the sound you chose above on the 16-part grid. You can also play them in live by holding the Write button and



A teenager...

playing them (again, you choose the pitch before hand).

Initially, this is a bit hit and miss, but better if you make use of the screen to make sure the Record icon is highlighted, and then you choose either method. Just make sure your pitch is defined *before* you record... so a little reverse thinking is in order.

rotaries can be recorded). Then there's a Chord button that lets you choose chord changes over following patterns by selecting them in number order – again, a nice way of bringing in some variation.

Other features common to the range include tempo selection – either home in by holding the BPM button and use the dials to choose a value, or select from Hip Hop (80BPM), Disco (120) and Techno (140) at the touch of the button. The volume of each PO is set here, too: hold the BPM button and set a level, highlighted by the number of the number keys lit. And just like last time, I'm pleasantly surprised by just how loud these things go, and also by the quality of the sounds on each.

Additionally, you can chain the patterns together by holding the Pattern button down again and then hitting the pattern-number buttons in the order in which you want them to play to create complete songs. All the

There's **something of a sonic curry fest here**: a few core ingredients but **lots of results**

If you've used any of the 10s, though, you'll be at home here, as you will with other functions like adding effects live – hold the FX button and choose your effect, as you would a pattern or sound. There are lots of filtery, trigger and glitch-type of effects to choose from, which really do bring an extra edge to all pattern performances. These can also be recorded in real time as the pattern cycles (any changes using the two units can also be sync'd together, so you might use the beats from one and play live with another – and as we'll see, some do excel in certain areas for this. The manual (aka piece of paper) that comes with each unit lists some scenarios here, including syncing a PO with a phone and Volca, something that has 'achingly cool' written all over it.

Finally, there's a clock on each unit, very much reminding me of my old



If you read our review of the first three POs from Teenage Engineering, you might think we have put exactly the same pictures, albeit with different colours, in exactly the same positions. Would we really be *that* lazy?



The publicity shot from the TE website implies the Arcade model might bring you games, but sadly, it doesn't

school calculator. I'd have also wanted games – especially on the Arcade one – but that's probably just me. (Maybe we'll see some chip tuners hack these to play games as some kind of weird reverse-parallel universe stuff.)

Specifics

So that is the overview. I'll now deal with each unit in number order, so it's PO-20 Arcade first. Arcade sound effects have been used in music since Space Invaders, from the Yellow Magic Orchestra onwards via novelty hits through the 80s, where the scene flourished via Tracker software. Of course, now there's a whole community of chip tuners who have taken the guts of old gear and made new sounds from it. Arcade (and in some respects the other two new POs) is a reverential nod to that scene. Ironically, though, having a machine with the 8-bit sounds laid out so accessibly might not actually appeal to many within that scene, but for the rest of us, Arcade brings a taste of it to our fingertips.

You'll recognise the sounds in PO-20 from a hundred trips to those exciting and slightly sinister amusement arcades that still litter the coastal resorts in this country (that is if you had my kind of childhood and haven't completely had the 70s tragically erased from your mind by now).

The key sounds are there: drones, lasers, beeps and basses. And while the sounds might be based in the 70s and early 80s, like the chip-tune scene, the patterns that they produce are vibrant and fresh and surprisingly varied. This has much to do with the PO ethos, which is to shove as many features in →



Office. You see?



There's no getting away from it, they are flimsy and open to the elements. Here's what you get around the back of each PO

→ there to vary, let's face it, not that many core ingredients and patterns. So the pitch dial will radically alter your core sounds and the effects and tempos will radically alter your patterns. Throw in some chord changes and real-time effects and you will be altering these ingredients a lot more than you think. Teenage Engineering has pulled off something of a sonic curry-fest here: a few core ingredients, yes, but a huge variety of results.

I didn't expect to be blown away by Arcade – Margate was always a lot more exciting to drive to on the school bus trip than actually getting there – but I found myself charmed by it. Whether the unit has the longevity is another matter, though.

If you **made beats** from **what you hear in our office,** it would be **the stuff of nightmares**

On to Office; and while I was excited by Arcade but not expecting big things, I did struggle with the idea of this one: office noises to make beats, right? It's hardly the stuff of dream rhythms, is it? And if you worked in our office and made beats out of the noises you hear here, it would result in the stuff of nightmares, let me tell you.

Fortunately, the Teenage Engineering office seems to be a little more musical and alongside what could be photocopy, typing and general shelf/door/filing cabinet hit sounds, there are half-a-dozen decent beats and basses to bolster things up – almost like Richie Hawtin is sat your office, maybe just like your new intern, and making beats out of your hole punch and stapler. "Alright Richie, yeah two sugars in my coffee please, mate."

The resulting Office beats and patterns are surprisingly punchy and aggressive and very edgy. It's here that the PO effects really do come into full force and you'll find yourself performing full live sets without realising it, with a few button pushes and casual glances over to your imaginary crowd (of co-workers).

So, yeah, Office is surprisingly cool, and out of the three units, I'd probably pick this one to use in my studio and to extract beats and ideas from and into my main DAW set-up. I just love the overall level of control that enables you to stamp your own authority onto the existing sounds and patterns, and again, you are amazed what PO has done with so few initial sounds.

Finally, it's time for the robots. Being a certain age, I must make the link between robots, operators, and calculators and mention Kraftwerk (yes, I didn't just hang about in amusement arcades in the 70s), and perhaps here we have the whole Teenage Engineering philosophy wrapped up in one unit.

Robot is solidly electro, comes packed with a bunch of synth sounds (not so much in the way of robot voices, though) and the ability to actually play and synthesise notes (to some extent



Robot is the best for synthesis (of sorts) and melody playing – perhaps not beats

In use tips

When using the new PO series (as with the current PO-10s) it's all about what you can do with them in real time/live. The Robot might be more about playing along with patterns and Office might be more about the beats, but each one allows chaining of patterns to make songs; the addition of live effects (which can be recorded); chord changes (again which can be chained) and a Step Multiplier that allows retriggering of steps. You can even fade your performances out with a couple of key presses (FX and Play) and on the Arcade, add a Drone version of the current chord. These real-time additions are the PO's core strength.



If you don't want all six, then you're stronger willed than us...

anyway). In this way, it seems more open than the other units, as you can make more in the way of melodies and have some control over the filter and envelope of the sound (via the rotaries). The sequencer is a little more limited in the sense that the sounds you play live are not necessarily recorded when you hit Write – they're more designed to be played live – but the overall feel is more 'play me' than 'program me', which is a nice variation.

Back to the sounds and you get good leads, whistles, the odd arcade bleep and some fantastic delayed synth notes (very Kraftwerk), although I'd have wanted some vocoder action – maybe that's an idea for the PO-30 range...

Patterns are chugging electro, some interesting more glitchy, off-beat breaks and again, some arcade-type movement. Because the source sounds are more synth than beat, the underlying rhythm sounds (from a micro drum kit selected as sound number 16) are a bit less varied here, so I wasn't perhaps as blown away by the patterns, I have to say. This is probably down to the fact that each sequencer makes use of one of the lead sounds and beats. The other units have more varied sounds across their key ranges (rather than just different pitches), so their patterns sound a little more varied. But fun-wise and ideas-wise, and particularly just for playing tunes with beats – and there are additional options for vibrato and expression for playing – Robot is where it's at for me.

Conclusion

I'll leave aside any debates over chip-tune purity – whereby the ethos is to mod old gear to make sound rather than have it done for you – and attempt to sum these up as music-making devices. And it's a little harder this time around, compared to the 10s. Inevitably, this set of Operators is a little more niche – the bass, leads and beats have been done, after all. This means that some will love them even more than the first set, but they will probably appeal to fewer people overall.

I have to say, though, that Office was the biggest revelation to me, while Robot was the most amount of fun. Like the reface keyboards from Yamaha – four devices for very specific and different tasks – these are very personal, so my opinions are, as always, just a guide. But I doubt there are any other products anywhere on the planet with which you can have so much musical fun and creativity, and that cost so little cash. **MTF**

If you want to finish the job off, silicone cases are available at €39 a go



Alternatives

Korg's Volca range is still the closest thing out there to the original PO-10 series, but there's little out there that touches the new PO-20 series, not in standard hardware, anyway. I say 'standard' because, of course, there is a world of chip-tuning, DIY electronic customisation out there where people are taking everyday gadgets and making them make notes. Expect a feature on that soon. Until then, grab a Speak & Spell and a screwdriver... and experiment. (Actually, thinking about it... don't.)



MTF Verdict PO-20 Arcade

+ Surprisingly varied Patterns
+ Good sound
+ Easy to use
+ Great hands-on tweaking
+ Few core sounds = many results

- Might not last creatively over the long term and have narrow appeal
- We wanted games!

Surprisingly flexible beats and melodies take this away from just chip tune. Longevity may suffer.

8/10

MTF Verdict PO-24 Office

+ Best for beats
+ A good range of Pattern styles
+ Extra sounds on top of beats add weight to the vibe
+ Effects really do help the variation on this one

- Much less melodic possibility than the other two on test
- As with all, the unit is flimsy

A lot better than we thought it would be, Office is the one we'll be using with our DAW productions. Inspiring beats, and it begs you to perform it live (which you will do!)

9/10

MTF Verdict PO-28 Robot

+ Great for hands-on melodies
+ Nice and simple synth action
+ Great for recording melodic ideas
+ Good and varied synth sounds
+ Impressive delayed sounds

- One lead and beats make the Patterns less varied
- Could have done with robot voices

The best for getting ideas down quickly and playing live. 35 years on, Kraftwerk could use it live!

9/10



NATIVE INSTRUMENTS

Komplete Kontrol S88 & v1.5 software

Choice 9/10 MusicTech

Komplete Kontrol, the software, has been updated and the hardware gets an addition, too, in the form of an 88-note keyboard. The ultimate integration? **Andy Jones** finds out…

Details

Kit Komplete Kontrol S88 keyboard and v1.5 software
Manufacturer Native Instruments
Price £729 (S88). Other keyboards in the range are now at a lower price: S25 is £299; S49 £369; S61 £429
Contact
T: +44 207 9207500
W: www.native-instruments.com
Tech Req
PC: Windows 7 later; Intel® Core™ 2 Duo or AMD Athlon 64 X2
Mac: OS X 10.9 later Intel® Core™ 2 Duo
4 GB RAM (6 GB recommended)

Key Features

- 88-note keyboard
- Features Komplete Select bundle of 10 plug-in titles worth £850
- NKS standard opens KK up to other developers
- Features Light Guides, Native Map, Smart Play
- Size (mm): 349x126x1389
- Weight: 14.4kg

There are some products that follow trends and some that create them. Komplete Kontrol definitely sits in the latter camp, and since its release in both software form and the S-Series keyboard range, it has set the bar, not just for controller keyboards but for the very way that music production software integrates with hardware. And it's also helped the computer to be relegated to less than the main player in the recording studio.

OK, you might think that's over-stating the impact of this range, so a quick recap might be in order…

Until a few years ago, the computer was the dominant force in music making. Native Instruments itself was one of many companies producing music software that could do everything, and I mean *everything*. However, a combination of software piracy, amazing new and colourful gear, a shift back to analogue, and the sorry fact that people were starting to mix with their eyes rather than their ears has meant that hardware has made a glorious return.

Ableton's Push and NI's Maschine have married the old software to this new world and simultaneously helped pull people's faces away from their computer screens. And just over a year ago Komplete Kontrol promised the same for the keyboard player – and thus the broader playing world at large. Komplete Kontrol marries a great piece of hardware to possibly the finest music production bundle out there: NI's Komplete collection (now at v10). The Komplete Kontrol software offers a shell-like interface, while the Fatar-made keyboards replicate the software in hardware form.

I say again: *replicate the software in hardware form*. 15 years ago, I'd have been happily telling you the opposite was true in the studio world. Crazy!

So anyway, with KK software you can dial up anything in Komplete and Kontakt, all from your S-Series keyboard via preset type or list. You can step through the presets and control the most important parameters via the rather great Clear View screens. The best bit is that everything, right down to the mult-coloured key ranges that have become Kontakt's trademarks, is now replicated on the actual hardware keys of the keyboards by way of the S-Series' Light Guides.

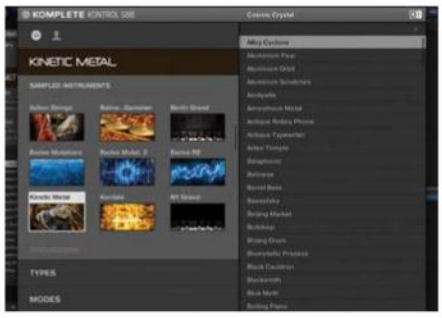
It's hardware that does software, it's multi-coloured and striking, and it keeps your head focused on what you are playing rather than what your mouse is clicking. It's *so* 2016!

Push and Maschine **pulled people away from computers.** Komplete Kontrol does **the same for keyboard players…**

It's also brilliant and Komplete Kontrol began, or at least helped chivvy along, a revolution in hardware and software integration that Akai has followed with its Advance range, Push 2 looks like maintaining for Live and Arturia, Novation and many others also have a hand in.

Which brings us nicely to…

With everyone jumping on the bandwagon, though, NI has not rested on its laurels and has made several important updates to the Komplete Kontrol philosophy over the last year.



1 MAIN BROWSE SCREEN
Press Browse in the hardware and you'll be presented with all of the plug-ins you have (and annoyingly some that you don't). Select here…



2 KOMPLETE INSTRUMENT
Load in your Komplete Select instrument – in this case Massive – and show the Plug-in panel to reveal the control assigned to the Clear View rotaries



3 OTHER PEOPLE'S PLUG-INS
Load in other plug-ins via the drop-down menu, but don't expect them all to be NKS-ready just yet. They will play, but you won't be able to control all of them via KK.

This includes third-party support, with the all-new Native Kontrol Standard (NKS) essentially meaning that the KK software and keyboards can be opened up to the VST community, so we're not talking about hardware integrated with just Komplete but potentially a lot more developers. So far, Heavyocity, ProjectSAM, Arturia, Spitfire Audio and Sonokinetic are just some of the big guns who have signed up. We'll be looking at how Komplete Kontrol works on third-party software as we review it (look out for the Arturia V collection update soon). For now, we'll cover what else you get with the software and the more immediate bonus: £850 of instruments within the all-new Komplete Select bundle.

Needless to say, this takes advantage of all of the S-Series' hardware's integration features while also offering a great selection of Native Instruments' own software line-up. It includes personal favourites Retro Machines, Monark and Massive, but there are some other choice titles here, including The Gentleman (piano), Drumlab and Vintage Organs – a little bit (actually quite a lot) of everything to show off what the Komplete Kontrol 1.5 software can do.

The hardware

So that's the software, but we're not finished. Testing the new 1.5 update required a Komplete Kontrol S-Series keyboard, so we took delivery of the brand new S88 keyboard and are including that in this test, too – to be rated separately.

Hardware-wise, the all-new keyboard is 'the biggie': 88 notes of fully-weighted, hammer action keys and touted as the ultimate keyboard for the ultimate controller range. The keyboard is identical in format to the others in the S- range in that all the action takes place in the centre bar – the two multipurpose touch strips on the left. There are Browse and Cursor buttons on the right to navigate through the KK software and eight Clear View screens and rotaries in the centre. There are performance keys, arpeggiator controls and the transpose keys to the left of these, and that is it. If you have any of the existing Komplete Kontrol S-Series, you will be at home here. If not, here's a quick recap of my previous review to explain that all important integration.

To use the S88 with Komplete Ultimate 10 is one of the best experiences in music production



Look closely and you can see which companies have already signed up to use the new NI Native Komplete Standard

Perfect integration

Within Komplete Kontrol you get access to all of your Komplete instruments (and now other plug-ins) and you can select sounds via the instrument title and then focus on preset type and select that way. The beauty is that all of this can be done via the hardware controls, so you can be away from the computer screen and more focused on your playing and recording.

You can load up the Komplete Kontrol software standalone or as a plug-in. The integration then dramatically swings into action as the most/important parameters of whatever sound you have dialled up are mapped (via Native Map) to the Clear View screens. I can't stress how cool this is. They are designed in such a way that the hardware/software combo almost mimics a hardware experience, so suddenly that FM7 preset you have always loved becomes an FM7 preset, which is easily and dramatically tweaked. The software instruments, therefore, come alive as not only preset selection is made easier but also editing, mapping, tuning and performance, all done via hardware, so negating a lot of – but not all of – your mouse clicks.

On that score, don't think that this by any means frees you of your computer completely – you will still need to load up the instruments in whatever DAW you use and, of course, work in that software, but the hardware is more than capable of handling your Komplete duties and can also load in other plug-ins. We had G-Force's excellent Oddity 2 loaded up, and while not all of the KK controls are mapped to certain plug-ins, more will doubtless sign up to →



the NKS standard. Back to the hardware, and the new S88 certainly feels its cost. It is pricey – a full £300 more than the 61 – but the keys will be attractive to proper players. They are fully-weighted, slightly heavy compared to the action on the Nord Stage 2 I also have in the studio for testing, but far more respondent than the 61-note S-Series I also have. Whether it's £300 extra worth paying is, I'm afraid, a decision you will have to make, as this depends completely on your playing ability and needs.

Konclusion

We've reviewed the new Komplete Kontrol software and S88 keyboard together, as they are obviously designed to go together, but you can, of course, choose any of the four keyboards now in the range, and anyone who has already invested in Komplete Kontrol can download Komplete Select for free, so get to it!

The new software is really all about NKS integration and the instruments you get from the off with Komplete Select. With both additions, you will be enjoying a much more integrated package and one of the best hardware and software set-ups out there. I say 'one of' because, of course, NI's own Komplete 10 software bundle is even better, and should you want the absolute ultimate in that arena – the appropriately named 'Ultimate' – it will cost you £579 over and above whichever S-Series keyboard you opt for. I have to say, though, that it is worth it, as you get hundreds of GB of the best instruments and content around – literally everything you should ever need for music production. Komplete Ultimate is also what the S-Series is designed to work with, so as good as Komplete Select is, this is what NI's designers had in mind with the S-Series keyboards – and to use the S88, Komplete Kontrol and Komplete Ultimate 10 is undeniably one of the best experiences in music production (I upgraded to v10 for this review and have to say it is staggering and very easy to install – Ultimate has never been so appropriate a name).

However, we're also going to mark the all-new S88 in its own right, as it comes with one or two caveats which we've already touched upon. It now costs a whopping £300 more than the 61-note, which puts the 88-note out on a limb a little price-wise.

So, while I love it – and I absolutely do – you may find those 61 notes enough, especially for £300 less. Personally, I think it's a bit steep a gap, or that perhaps the 61-note now looks even more attractive. Having said that, these 88 notes are weighted and seeing Light Guide across them in action in your studio is something else to behold. So if you are a player, and have a few quid, this is the one to get.

Either way, one of the range should be an even more attractive option, especially with the world and his wife now invited to the Komplete Kontrol party and £850 of software to tuck into when you arrive. **MTF**

We used the S88 as the centre-piece of the *MusicTech* Christmas party decorations…

Alternatives

Where do you start? In terms of controller keyboards, there are stacks to go for. I've had some great experience with the Alesis V and VR ranges and Korg's Taktile models, but for more integration the Arturia KeyLab range has been updated and includes a whole range of that company's classic synth heritage. The closest to the KK experience, though, is the Akai Advance range, which includes VIP software that really gives you the hardware and software integrative experience.

MTF Verdict KK S25-61

+ Great new prices
+ Komplete Select is amazing and a real bonus and advert for the rest of the Native range
+ NKS could become the new standard in integration
+ The Komplete Kontrol software opens up your instruments in a hardware-like way
+ The scale and arpeggiator sections are both excellent
+ Light Guide keys. Come on, what's not to like?!
+ Used with Komplete Ultimate, this really is the ultimate hardware software bundle

- … but it will cost you £579
- You can't do everything in hardware (just yet)

The new prices, addition of NKS and Komplete Select make the S Series even better buys.

9/10

MTF Verdict KK S88

+ Proper player keyboard makes this the ultimate Komplete Kontrol keyboard of the range
+ Just how cool are 88 Light Guide keys? Very!
+ Wider keyboard range opens up Kontakt instruments further
+ Again, used with Komplete 10 Ultimate and you have, well, the ultimate combo

- Keyboard may be a little heavy for lighter acoustic players
- Price is £300 more than an S61, so quite a bit more for the extra playing action

It's the ultimate keyboard in the Komplete Kontrol S-Series range, yes, but it will cost you. Get it with the optional Komplete Ultimate 10 and you really do have the best.

8/10



Choice 9/10 MusicTech

ZIVIX

Jamstik+

Zivix's Jamstik is billed as a guitar learning aid, but **Martin Delaney** has other ideas – it could just be a great MIDI controller too…

Details

Price **$299**

Contact **www.jamstik.com**

Features

- Guitar tuition tool/MIDI controller
- 5-fret neck, 16" long
- Works with Mac, Android, OSX
- Mute button and d-pad control
- Charges via computer USB
- 8 hours per charge
- Particularly good with Ableton Live

Us music tech types have an insatiable curiosity about alternative MIDI controllers, whether it's to replace or complement our existing setup. There's always room for one more, right? Enter the Jamstik+, a guitar tuition tool which might seem kind of toy-like, with its little plastic body (just over 16" long) and stubby five-fret neck, but has huge potential for MIDI control, putting it squarely in the sights of any free-thinking controller geek. (Yes, the '+' in the name is important – it is an update to the original Jamstik, which relied on WiFi to connect to a computer, while the Jamstik+ uses Bluetooth or USB.)

Infrared sensors in the fingerboard combine with a hexaphonic pickup to send note and string information to your Mac, Android, or iOS device, with further control messages being sent via an accelerometer. There's also an inordinately large mute button located where the bridge is on a normal guitar, and a d-pad type arrangement of buttons on the top edge of the body. The package contains the Jamstik+, a micro USB cable, guitar strap, picks, and printed setup information. The Jamstik+ charges from the computer or any standard USB charger, and should provide around 8 hours of use per charge (courtesy of a replaceable phone-style battery). It's straightforward to set up - there's a dedicated free app available on all platforms which handles the Bluetooth connection, string behaviours, velocity response, trigger sensitivity, general MIDI functions, button assignments, and MIDI channels (each string can be assigned to a different channel, which opens things up for software control).

Ableton Live is a natural partner for the device – it's rewarding to trigger soft synths like Operator from this petite stringed instrument, which feels more like a MIDI ukulele than a guitar once it's strapped on. Of course you can use any type of sound, so if you like the idea of programming beats from a guitar interface, you'll be in heaven. There were no issues using Logic instruments and third party plug-ins – at the most basic level it's just like configuring a MIDI keyboard. It doesn't feel remotely like playing a guitar with a MIDI pickup; although the Jamstik+ uses 'real' strings, they're very short, and there's no 'acoustic' resonance at all, like you'd get when you play an unplugged electric guitar. No physical tuning is needed, either – it's handled by the app, where you can specify any custom tuning you like, per string. By default the d-pad buttons provide octave +/- and capo functions. You can use a pick, or play finger-style, and there's a tap mode optimised for one/two handed tapping. The accelerometer reinforces the game controller angle, and this can send any MIDI control message, so you could be playing notes on the Jamstik+ and then give it a wiggle to change your filter frequency, for example. Setting changes are stored dynamically in the Jamstik+, but I'd like to be able to save a library of presets for different situations.

I built an Ableton set to make use of the 6 strings and 5 frets and separate MIDI channels to launch a 6x5 block of audio clips, and the d-pad buttons to activate effects, and it was as fun as it could be – which is a lot. I then routed notes and CCs from the Jamstik+, via Live, out to a Bass Station 2, using the semi-guitar vibe to play bass and lead tones; and remember, this is wireless with Bluetooth, so you can walk and play. On iOS, the Jamstik+ experience is more or less the same, using the app for configuration and then playing various soft synth apps – the guitar tutorial content is also available across all platforms if you want it. Lefties can enjoy the Jamstik too – change the orientation in the app, and flip the Jamstik+ over.

I wouldn't use the Jamstik+in any kind of guitar role – I have guitars for that. It is, however, an awesome MIDI input device and is a surprisingly expressive little instrument. If you like custom control setups and game controllers, this is one you should try. **MTF**

Alternatives

Game controllers! The good old Wii Remote still works great with MIDI, thanks to 'translator' applications like OSCulator. Livid Instruments Guitar Wing adds a MIDI controller with buttons and motion detection to your 'real' guitar. MIDI guitar pickups are the more high-end, expensive way. See Roland and Fishman as examples…

MTF Verdict

+ Excellent MIDI 'guitar' controller
+ Easy Bluetooth or USB connection
+ Accelerometer is great for effect controls
+ Portable little package
+ Works with iOS or OS X
+ Great for Ableton Live users

- Neck dives a bit
- Price will deter the merely curious
- Not necessarily a first-choice guitar learning tool

The Jamstik is quirky, it's not cheap, and I'm not convinced of its value as a guitar learning tool. But on the major 'however' front, it's a fascinating and fun MIDI device for anybody who loves new ways of communicating with their software, and particularly recommended if you like game controllers.

9/10



FABFILTER Pro-C 2

FabFilter has updated its Pro-Series compressor. **Mike Hillier** finds out if it really is fabulous

Details

Price **£114**
Contact **info@fabfilter.com**
Web **www.fabfilter.com**
Minimum System Requirements
PC Windows XP or later, AAX, RTAS or VST host
Mac Intel Processor, Mac OS X 10.5 or later, AU, AAX, RTAS or VST host

FabFilter Pro-C is now the second plug-in in the company's Pro range to get a major update, following on from the incredible update to Pro-Q last year. The new version of the FabFilter's compressor adds a considerable number of features to the already popular plug-in, including five new compression styles, a vastly improved side-chain EQ, and a new Retina-display optimised interface, which can be scaled from small to full-screen.

Eight in one

Probably the biggest single improvement in Pro-C 2 is the new user interface; even in its default setting it is larger and easier to navigate. FabFilter has embraced user interface design like no other developer, making the most use of what is possible with plug-ins instead of sticking to emulating old designs. There are no skeuomorphisms here, no rack screws, paint scratches or needle-based VU meters. Instead, the meter shows a moving waveform as it passes through the compressor up against a graph of the gain reduction, letting you see at a glance what your attack and release settings are doing to the gain reduction. Against this, another graph shows the compressor ratio and knee, again enabling you to quickly set these visually against the waveform. There will be those who say visual mixing is disabling, and that you should use your ears, which is of course true, but knowing you can get 99% of the way there visually is a great help in getting to the settings you want quickly.

Pro-C 2 has eight different compressor models, of which five are completely new in Pro-C 2. The Mastering compressor works very quickly, while producing very little harmonic, making it very useful as a clean, clear vocal compressor when artefacts have to be avoided, as well as on percussion. On the opposing front, the Pumping model has been designed with subtlety thrown out of the window. This model is great when fed a side-chain kick for those over-the-top pumping sounds most commonly associated with EDM. Perhaps two of the most useful new models, though, are the Bus and Punch models. Punch has a rich tonality, which lends weight to anything fed through it, while Bus applies that final gloss to a mix. The Vocal compressor has been designed with automatic knee and ratio, but is the only one of the new models we haven't taken to immediately, perhaps because we got great results out of the Punch, Mastering and the three old models on vocals.

Key Features

- Eight compression styles
- Sidechain EQ
- Smooth lookahead mode
- Intelligent auto-gain
- Up to 500ms Hold
- Custom knee

Alternatively

Compressors are a dime a dozen, but something with this kind of flexibility, capable of anything from transparent detailed control, through to characterful glue, is rare. Sonnox Oxford Dynamics is probably the closest competition, with three different compression types and some powerful additional features.

The best is yet to come

While the five new compressor models are the easiest new features to get your head around, the real strength of Pro-C 2 comes from the subtle touches that FabFilter has added. It might not seem like much at first, but the Hold feature added to the Attack and Release characteristics – something more commonly seen on gates than compressors – has made its way onto more tracks. In fact, we've been using it on all our recent video tutorials as a means of consistently ducking the audio track under the voiceover. We also recently used it on the hi-hat channel in a mix to both tighten the hat choke in response to the kick and to control bleed, with the end result being more controlled and groove-based than anything we could have got with a gate.

Another of the improved features is the sidechain, which previously had high- and low-pass filters. It now has a mid-band with selectable shapes, based on Pro-Q. You can, like before, add high- and low-pass filters, but you can also shape the filters as you see fit, and use the mid-band to add a bell curve, notch, shelf, or even a tilt EQ. We described the advantages of using fully-featured EQs in the sidechain in a previous workshop, "Digging Deep Into The Sidechain", but it's great to see one so prominent in a compressor like this. The actual response of the compressor once you've changed the sidechain EQ is sometimes hard to work out, but as a rule of thumb any frequency you boost in the sidechain will be reduced more by the compressor.

As always, FabFilter has managed to wrap an incredibly powerful tool into an easy-to-use and intuitive user interface, which makes getting great sounds simple. The original Pro-C got a lot of day-to-day use on our sessions, but it was always the compressor to reach for when you just needed a job doing. Pro-C 2 can handle anything you throw at it, and is already pushing far more expensive compressors aside. MTF

MTF Verdict

+ Flexible compressor
+ Easy to use
+ Great user interface

Pro-C 2 is set to become a staple of many professionals' toolkits. Everyone needs a good compressor, and there are few as good as this. A future classic.

10/10



Choice 9/10 MusicTech

SONIC ACADEMY Kick 2

Sonic Academy's kick-drum instrument gets a 2016 makeover, but is it a stairway to drum heaven, or a one-way ticket to drum hell? **Alex Holmes** finds out

Details

Price **£49.95 (£24.95 upgrade from Kick 1)**
Contact **via website**
Web **www.sonicacademy.com**
Minimum system requirements
VST, AU, AAX 32-/64-bit host
Windows 10, 8, 7, Vista (32 Bit), Intel Core Duo or AMD Athlon 64 X2, 512 MB Ram
Mac OS X 10.7 or later (32/64 Bit), Intel Core Duo, 512 MB RAM (no PPC)

Features

- Kick-drum instrument with Sub oscillator and three Click sample layers
- 230-plus genre presets
- 180 click samples
- Modelled Tube, Clip and Wave distortion
- EQ, Compressor, Limiter and Harmonics controls

It's fair to say that the kick drum is king in most modern dance music. As genres come and go, it tussles with the bassline for superiority, but in most club music, getting the right kick sound is so fundamental, it can make or break a track. We've been big fans of Sonic Academy's Kick plug-in over the last couple of years, and its punchy sounds have been used on a multitude of tracks as both kick drums and basslines. We knew Kick 2 was in development, but we're excited to finally get our hands on this new and improved drum synth/sampler instrument.

Click track

Kick 2 adds a whole slew of new features and changes, the first of which is the dropping of the Nicky Romero tag that came with the original plug-in. The interface is now larger and more polished, with splashes of colour and a waveform view that updates as you edit the various parameters. Most fundamentally, the Click area has been expanded from one to three sample layers, meaning you can create more complex textures and do more effective drum layering, choosing from either the extensive list of drum clicks, hats and noises, or loading your own with a new drag-and-drop functionality. Each Click section still has pitch and volume controls, but now adds dials for sample length, start point, pan, and phase alongside a LP/HP filter. This makes it much more flexible, as you can now load in something like a live kick sample and easily scoop out all the low end to allow space for the Sub part of the instrument. Our only gripe here is that the Pan control is so tiny that we almost missed it! All the Clicks, along with the Sub section, can be muted or solo'd, and have a button to toggle whether they respond to the keyboard, so you could have a pitched bass kick, with static transient click.

Sub focus

The main part of the plug-in is still the Sub oscillator, with a large window for drawing break points and bezier curves for the pitch and amplitude. This has been enhanced with an excellent harmonics section, that lets you add character to the waveform by dragging up eight sliders to bring out different parts of the spectrum. Other improvements include the addition of analogue-modelled Tube and Wave distortions, which are a big improvement on the (also included) Clip distortion from Kick 1. There's also a Compressor, a final output Limiter, and a Drive dial that controls how hard the signal hits the output. It's in the combination of these elements, the distortion, and the Harmonics section where the power of Kick 2 as a tone shaping tool really starts to shine. Different combinations of each now give different types of saturation and overtones, meaning you have a broader palette to play with. Other notable features include a drag-and-drop export function, and a new EQ section with a low shelf, high shelf and two notch filters for extra shaping of your overall tone. This is a nice addition, although a spectrum analyser would have been the icing on the cake to help with your tweaking. Last but not least, (as it was high on our wish list for Kick 1), the instrument is now monophonic with a new dial for Portamento. This makes it easier to have long tails without the worry of them overlapping and muddying up your low end. That said, without wanting to sound fussy, it would have been nice to be able to switch between voice modes.

Alternatives

The most obvious alternative is Big Kick from Plugin Boutique (£39.95), which doesn't quite have the depth of control of Kick 2, but has some excellent artist preset packs. If you want even more control, built-in FX and a mod matrix, then there's Vengeance Sounds' Metrum (£89), although it might be overkill for some.

New and improved

Kick 2 is a serious upgrade on the original plug-in, and is now more flexible than ever. This can be seen in the 230-plus genre presets that also include layered snares, claps and toms; a suggestion perhaps that this is much more than just a kick-drum instrument. There are also a few bass presets, although we would have liked to have seen more of them, given how surprisingly good Kick 2 is as a bass synth. Whatever the weather, this will no doubt be getting some serious use in our future tracks. MTF

MTF Verdict

+ Highly flexible tool for sculpting kicks and hits
+ Solid collection of presets
+ Excellent distortion and harmonics section
+ Crisp, clear GUI

- Kick 1 user presets not compatible with Kick 2
- Pan controls incredibly small!
- No spectrum analyser for EQ

Kick 2 is a more nuanced instrument with a more refined palette of tools for fine-tuning your kicks and sounds, and it's an essential purchase for anyone writing music destined for the club.

9/10



Excellence 10/10 MusicTech

Above: Epica Bass's main window and presets list

SAM SPACEY
Epica Bass

A sample library full of synth basses isn't something that usually comes to mind when buying your next library. **Matthew Mann** finds out if Epica Bass is bold enough to take up permanent residence on his hard drive…

Details

Price **£89 / $138**
Distributor **Time+Space**
Contact **via website**
Web **www.timespace.com**
System requirements **Kontakt 5 or Kontakt Player (included)**

I like sample libraries, but typically don't use bass-sample libraries. Why? Because I find it easier these days to create my own bass sounds with hardware or software synthesisers. But what if you don't really have any virtual or hardware synths that do bass well, or you're not a synth programmer?

> Did I say **this is all bass**? Hardly: **many of these** presets can be **equally useful as leads**

Well, mega-producer Sam Spacey, the creator of the excellent (and critically acclaimed) Epica virtual synth from Zero-G, has gone to great lengths to create the ultimate bass-synth library for you.

Enter the bass

Epica Bass is a Kontakt library of nothing but bass. Actually, it's nothing but synth bass. It works with both Kontakt and the free Kontakt Player (which is even included). It consists of a whopping 26,657 24-bit mono samples and comes in at around 7GB. All of the samples have been meticulously edited, hand-looped, optimised and organised into a 437-preset library of searing, wobbling, dance-floor pumping, brain-throbbing, gut-twisting arsenal of all-round killer bass instruments. The samples were all taken from an impressive list of analogue synths, including an original ARP Odyssey Mk III, a Sequential Pro-One, Yamaha CS-15 and CS30 synths, an Oakley Sound Systems Modular, an Analogue Solutions Telemark-K, Oberheim SEM and Matrix 6R synths. Sam processed all synths through a variety of outboard gear (including a Neve Pre and a UBK Fatso) to get the best sounds before final sampling. It was apparent from listening through the presets that he was very serious indeed, because the sounds are fantastic.

The sounds are laid out in five folders:

- Monophonic
- Arpeggio
- Multi
- Polyphonic
- Raw

The Monophonic folder is by far the largest and contains the bulk of the presets. The amount of programming involved is truly mind-numbing. The Arpeggio folder has patches that are well thought-out and useful. I found presets in this folder lent themselves not only to bass arps, but also to higher register arpeggios that fill out a track nicely. The Multi folder is full of presets that combine multiple instruments into some excellent layered presets. This is a nice change from the mono folders previously mentioned. You can even play polyphonic patches in the Polyphonic folder. Finally, the Raw folder is chock-full of unadulterated samples of the original instruments. This a great place to jump in and develop your own twisted sounds. And did I say this is all bass? Hardly: many of these presets can be equally useful as leads and more.

Sampled filters?

Sam even went so far as to sample the filters of each synth at different stages. This means some patches will move the sample start time so that you are, in essence, moving through the filters of the original instrument. I can tell you that this technique was used to great effect on these instruments. Also, the instruments were multi-sampled through every single note rather than just a few. So, there's no aliasing of stretched notes. And there are at least six 'Round Robins' for each note, so the instruments sound as they should… slightly different each time you hit a key.

Features

- 100-per-cent analogue synth sound sources
- 437 instruments
- Samples from ARP Odyssey, Sequential Pro-One, Oberheim SEM and more
- Recorded with boutique hardware
- FX rack, arp, modulation and more



This adds to the character of analogue synths – especially all those lovely vintage ones.

Sam also went into meticulous detail when programming modulation. The mod wheel itself is routed to a number of places – from FX adjustments to sample-start times.

Less is more

There are only three main pages in Epica Bass. Each page is dedicated to one or more tasks and, while options abound, the overall interface remains clean. The main pages are Main, Arp and FX. The Main page has four sub-controls: Amp, Mod, Filter and LFO. Each allows you to tailor the sound to your liking.

include a lo-fi bitcrusher, flanger, chorus, 1176-style compressor, SSL-style EQ section, a convolution reverb, and a basic delay. All the effects onboard are very good and do wonders for some of the less-animated patches. Oh, and all effects parameters are easy to tweak, with no hidden menus. The dials for each parameter are clearly labelled as to what they do… and a visual indicator at the top of the FX window indicates the exact parameter you're tweaking, and by how much.

Epica Bass's FX section is ample and well-featured, having been borrowed from Native Instruments' own line of effects. They sound good, too. The delay is very simple, but does its job well.

One more thing…

I have to mention a feature that really impressed me about Epica Bass. The Start Point Modulation mentioned earlier is brilliant. Changing the sample start time allows the user to hone in on the filter sound of the instrument. This does wonders on things like Depeche Mode-esque filtered basses and resonant leads. It really reminded me of the way Ensoniq managed the wave samples in the SQ series years ago. I really feel that this method of programming (which can be assigned to the mod wheel or other sources, such as velocity) gives a much better quality to the sounds, rather than relying on emulated filters (although some of those are included, too). The thick, warm sound of the analogue filters is genuine.



Above left: Epica Bass' sub-pages on the main page (AMP, MOD, FILTER, LFO)

Above: Epica Bass's many-faceted FX section is simple to use and adds an extra dimension to your sounds

> I've already **found use** from **so many presets**, I can **imagine using this on every track**

The Arp page is dedicated to an onboard (and well-featured) Arpeggiator as well as an Auto-scale function, which makes it easier to compose in different scales if you're not a music-theory devotee.

The FX page is packed with Native Instruments' own effects, including: bit crusher, flanger, and chorus to name a few. These effects work well to polish the final sounds to perfection.

Give my creation life!

The FX page is one of the simplest to work with. All effects are shown on the FX page, and you only have to turn on the power switch for the effect you want to use and it's ready to go. The effects

Conclusion

Epica Bass never left me wanting. I'm very impressed with the layout, the quality of the samples, and the well thought-out programming of the presets. I have found use from so many of the presets already that I can imagine using it somewhere on every track. Honestly, I'm finding it hard to find fault in this library at all.

If anything, I maybe wish the size of the library was a little smaller, but anyone with a sizable samples drive will never even bat an eyelid.

Of course, while this library may not appeal to everyone, I'm pretty much convinced that most electronic-music producers will get quite a bit of use out of this excellent selection. The tweakability and programming options onboard allow you to create anything you want and the presets are so good, you'll find it difficult not to use them just the way they are. **MTF**

Alternatives

Other bass-specific instruments include Spectrasonic's insane Trilian Bass, which comes in at a hefty 40GB. It covers electric basses as well as a host of synth basses, but also costs about twice the price. Vir2's Basis is a Kontakt library that features lots of electric and synth basses. It comes in at about 7GB, but doesn't have the level of programming that Epica Bass does… although it's also only $99.95. And, of course, Native Instruments has several bass libraries, but they are all purchased separately unless you have the dough to drop for Komplete. If you can swing the £89/$138 for Sam Spacey's Epica Bass, I'd suggest that you do, now, before he realises what he's done and doubles the price.

MTF Verdict

+ Fantastic-sounding bass-sample library in Kontakt format
+ Includes the free Kontakt player
+ All the samples are meticulously recorded and sound beautiful
+ Ideal for use in most styles of electronic music
+ Tons of options for creating your own presets
+ Start Point Modulation makes the filters sound brilliant

- Could be a slightly smaller
- Doesn't write hit singles for you, but will certainly inspire you to write your own

This is a fantastic library of bass synths. If you can't use it in most of your tracks, you're probably not writing electronic music.

10/10



Choice 9/10 MusicTech

SONIMUS

Burnley 73, SonEQ Pro and Sweetone

There is no such thing as a Holy Grail of EQs, with each one being able to bring something new. **Mike Hillier** tries out three very different vintage-EQ emulations…

Details

Manufacturer
Sonimus
Price **Burnley 73 $59, SonEQ $59, Sweetone $32**
Web
www.sonimus.com
Minimum system requirements
PC Windows XP, AAX, RTAS or VST host
Mac OSX 10.6, AAX, AU, RTAS or VST host

Having a wide selection of EQs to turn to when it comes to mixing is arguably as important as having a selection of microphones, preamps or compressors. While it may seem like a 3dB shelving boost at 12kHz should sound pretty much the same on any given EQ, in reality, there are dozens of small factors which play into the 'sound' of an EQ. From the shape of the curve to the subtle harmonic saturation it may add to the signal, each EQ has its own unique colour. When it comes to mixing, having a selection of different colours available can help you to shape the mix as you see fit.

In this review, we take a look at three new EQs from Sonimus. Each of these EQs has a unique vintage vibe, capable of bringing its own signature colour to your mix.

> From the **shape of the curve** to the **harmonic saturation**, each EQ has a **unique colour**

Made in Burnley

The first of this trio of EQs is the Burnley 73, a not-so-subtly named emulation of probably the most famous British EQ: the Neve 1073. The emulation models not only the sound of the 1073's famous EQ section, but also the two input stages, adding an output stage with +16/-24dB of gain available to make up for any level changes brought about by the EQ itself.

As has become traditional when modelling classic EQs, the Burnley 73 has attempted to model not only the sonic signature of the Neve, but also its visual aesthetic (although those with a keen eye for vintage accuracy will notice that the Marconi-style knobs have been replace with chicken-head style knobs, which while not quite authentic, are much easier to read at a glance). The dual-concentric knobs for the mid and low bands have been kept which, while in keeping with the original, are somewhat unwieldy – at least for those of us who like to work fast, without having to spend too long with our hand on the mouse.

Sonically, the Burnley 73 is a great tool. The input stages provide options for instant gratification, adding depth and drive to the sound even before we start shaping the tone with the EQ itself. Like a real Neve, the Burnley 73 is incredibly musical, capable of fairly large boosts while maintaining a natural quality to the signal.

On vocals in particular, this EQ is a real dream to work with. The high-pass filter and low-end shelf enable you to bring warmth and body to a performance without any muddiness. The high shelf adds a shine to proceedings, but it's the single parametric mid band which can really bring things together, whether that is cutting out a little nasal quality in the upper mids, or bringing out the chest cavity in the lower mids.

This band is so useful, in fact, that we think it's a shame that the 1073 gets all the attention compared to the more versatile Neve 1084. And in the world of digital, why not have the best of both?

The passive tone

The SonEQ Pro is the advanced version of a free plug-in, SonEQ, available from the Sonimus website. It is based on classic passive EQs like the Pultec EQP-1A but, unlike the Burnley 73, doesn't attempt to exactly clone the Pultec, instead taking inspiration from it and other passive EQs to create an all-new and very powerful tone-shaping tool. SonEQ Pro has four bands, with additional sweepable high- and low-pass filters and a variable drive output stage.

The low band copies the Pultec trick, with independent boost and cut knobs, enabling you to boost and cut the same frequency simultaneously. The exact result using this technique will depend if the band is set to 'shelf' or 'bell', but the principle is that, as the cut and boost have different Q settings, the end result will see either an emphasis or de-emphasis around the cut-off frequency, but the expected result at either side of the cut off. This technique can work wonders on low-frequency signals such as kick drums, toms or bass guitar, adding weight without any ringing or muddiness.

The low- and high-mid bands offer only a single gain knob with +/-10dB of control, variable frequency and a switch for narrowing or widening the Q. Finally, the high band is fixed as a shelf with +/-10dB gain at six, eight, 12 or 15kHz. This shelf is one of the standout features of the EQ, producing a silky smooth boost in the top end, which can be great on vocals or guitars for bringing out the air around the signal. It could also be just as at home across an entire mix.

The SonEQ Pro is a fantastic EQ, which offers a very different scope of tools to the Burnley 73, and therefore complementing it well.

Leaning this way or that

The final EQ available from Sonimus is the Sweetone. This is the cheapest of the three and at first glance seems the simplest, too. It has variable high- and low-pass filters, plus an additional mid-band 'Tone' knob, with two modes. Behind this veneer of apparent simplicity, however, lies a surprisingly useful tone-shaping tool.

The high-pass filter is sweepable from 10Hz to 2kHz, while the low-pass filter enables filtering from 22kHz down to 2kHz. Both filters have selectable 6dB/octave or 12dB/octave shapes, but the high-pass filter has an optional Blow mode which adds resonance to the filter. This resonance adds a boost in the low end at the cut-off point, which can be perfect for adding weight to low-end sources while simultaneously cutting any mud from below the cut-off point. This is the same trick we see employed by tools such as the Little Labs Voice Of God.

The Sweetone's high-pass filter has an optional Blow mode which adds resonance

Apart from top-and-tailing the frequency range with the filters, the main focus on the Sweetone is the Tone control. This simple dial has a number of functions, depending on how the two switches around it are set. In Tilux mode, the Tone control acts as a tilt EQ: pushing the high end up and low end down when you dial it clockwise, and the low end up and high end down when you dial it anti-clockwise. The pivot point of this tilt is controlled by the Normal/Sweet switch which, when in Normal position, is centred around 650Hz, while in Sweet, is centred around 2kHz. The Loud mode switches the tone control from a tilt to a pair of shelves. This time, dialling the control up will add both high and low end, creating a quick 'smile curve', while dialling it down will cut the top and bottom ends, creating a mid-focused sound. Again, the Normal/Sweet switch changes the behaviour, with Sweet moving the two shelf cut-off points further towards the low and high end.

In mixing, this kind of tool can be very useful for quickly altering and placing sounds in the mix. It feels less controlled than a standard parametric, but the ability to quickly get your sounds in the right ballpark cannot be underestimated. We could easily see this being placed early in the mix chain for various instruments, with any specific frequency issues cleaned up by a standard digital EQ, or one of the two other Sonimus EQs. In addition to shaping the tone of the signal, Sweetone has a preamp stage after the EQ, which can be used to further boost or cut the signal by up to +/-10dB. The preamp can be used for simple makeup gain, with no additional colouration, or switched to a more analogue-coloured stage, which can be used to add further colour with some even harmonics.

Any one of these three EQs could make a great addition to your studio plug-in collection, but they complement each other well, and provide different functionality which can be brought to bear on different instruments within your mix to produce the right result in the right context. Each is very easy to use, and has its own character. MTF

Features

BURNLEY 73
- Neve 1073 emulation
- High-pass filter
- High- and low-end shelves
- Parametric mid-band
- Models both Mic and Line inputs

SONEQ PRO
- Classic passive EQ emulation
- High- and low-pass filters
- Low-band has independent boost and cut
- Two parametric mid bands
- High-end shelf

SWEETONE
- Simple tone control
- High- and low-pass filters
- Tilt or 'Loud' tone control
- Resonant high-pass 'Blow' switch

Alternatively

The Neve 1073 is one of the most sought after, and therefore most frequently emulated EQs in the audio world. UAD owners have the legacy 1073 as well as a new version with Unison technology for Apollo owners. Waves, too, has a couple of EQs based on the Neve, the most obvious being the Scheps 73, based on Andrew Scheps' own hardware unit. Similarly passive EQ emulations are quite common, with UAD again having one of the better Pultec emulations available. Softube have an emulation of Tube-Tech EQs, which are hardware EQs themselves based on the Pultec designs. Tilt EQs are not as common. One of our favourites until now has been Softube's emulation of the Tonelux Tilt, which has similar features to the Sweetone, but without the option of switching the pivot frequency of the tilt, or the resonant high-pass filter.

Choice 9/10 MusicTech

MTF Verdict

BURNLEY 73
+ Model of the Neve 1073

SONEQ PRO
+ Pultec bottom end
+ Sweet high-end shelf

SWEETONE
+ Quick tone shaping
+ Switchable resonant high-pass filter

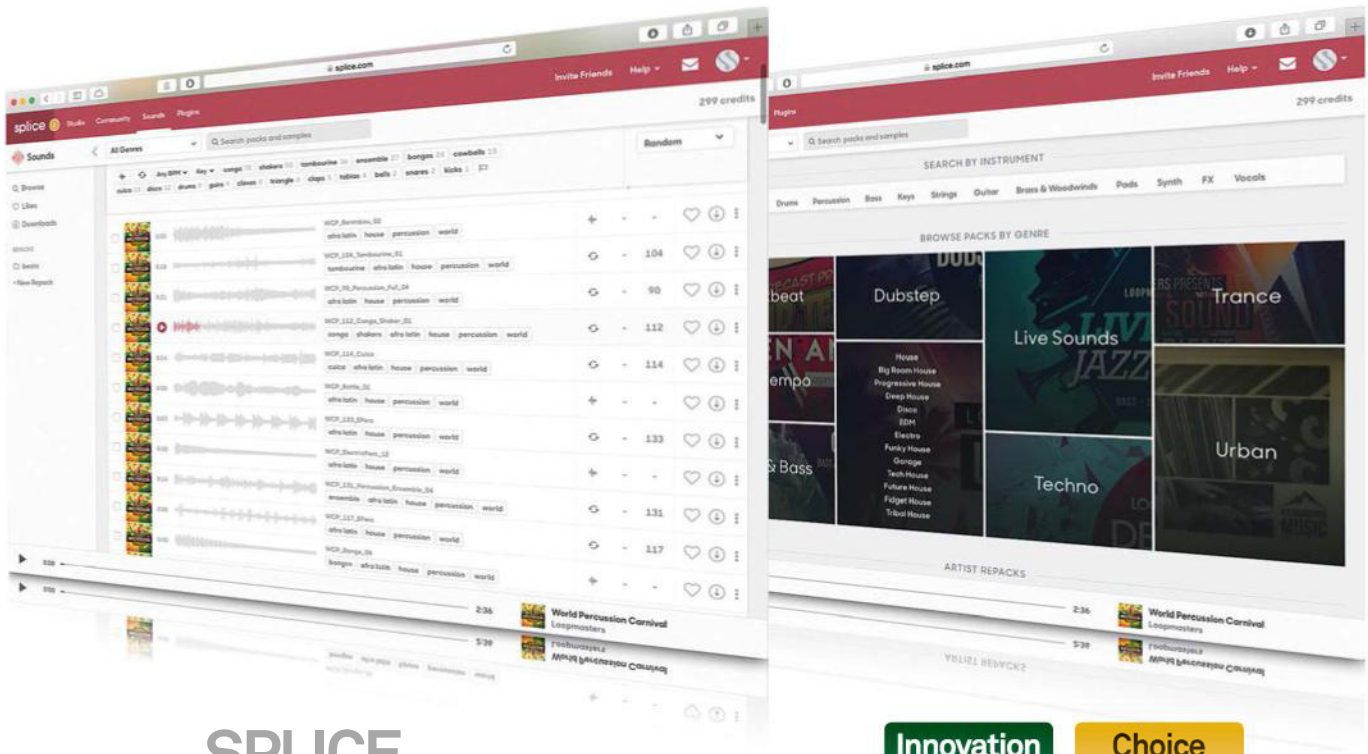
BURNLEY 73
- Fiddly skeuomorphic dual-concentric knobs

SONEQ PRO
- Limited mid-band frequency range

SWEETONE
- No resonance on the low-pass filter

Three great EQs, which between them cover almost all the ground you could need in a mix scenario.

BURNLEY 73 7/10
SONEQ PRO 8/10
SWEETONE 9/10



SPLICE

Splice Sounds

Innovation MusicTech

Choice 9/10 MusicTech

The days of paying for samples you never use could well be at an end. **Hollin Jones** explains how Splice Sounds could change your world…

Details

Price $7.99 a month
Contact Via website
Web www.splice.com

Key Features

- Access from desktop or mobile
- Audition any samples for free
- 700k samples and growing
- 100 or 300 downloads per month
- Build and share your own Repacks
- Downloads appear on your computer
- Drag and drop straight to your DAW
- Extensive searching capability
- Artist-curated collections encode

Splice started life not all that long ago as a web service that allowed users of certain DAWs to upload their raw project data to the cloud and then collaborate with others. Or, store their own work in the cloud so they could open it on any computer to keep working on it. That free service is called Splice Studio, but the company has now created a different kind of service: Splice Sounds. Claiming to be a new approach to buying samples, it's the kind of thing that has become possible only in the last couple of years with advances in web and mobile technology.

So what is it?

Splice Sounds is an online catalogue of royalty-free samples – 700,000 and counting – from some industry-leading figures, companies and artists. Where it differs from conventional sample packs is that it is entirely database-driven yet friendly to use. The developers say that they found people complaining that they would buy a sample pack and end up using only a fairly small percentage of the sounds inside it. The others were effectively useless to them. Worse, frustration with this can lead some people to pirate sample collections, which is not hugely difficult if you're determined to do it. This is similar to the challenge faced by the recorded music industry, and Splice's answer is not unlike the way that Spotify or Apple have responded to people's changing music-buying habits – not to mention a way to try to discourage piracy.

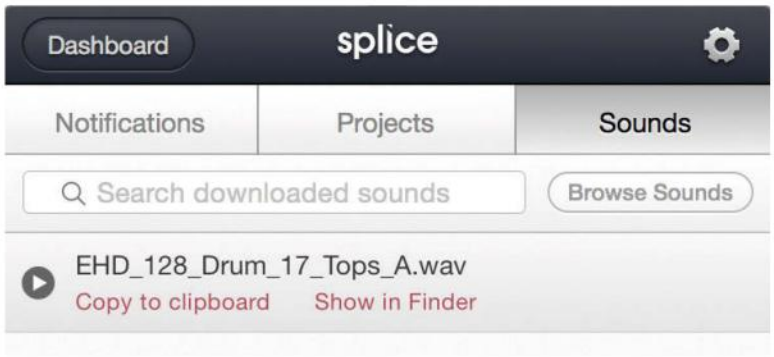
This limit is to stop people joining for a month and downloading the entire database before cancelling, which wouldn't be very fair. It will also probably have the effect of making you consider more carefully which sounds you really do want rather than just downloading thousands without thinking. If you do cancel, you keep the samples you have already downloaded but lose access to download any new ones. You can also choose to pay $13.99 a month and get 300 samples, though it's not currently possible to pay extra if you want to go beyond this in any given month.

> The idea, in short, is to **let people search**, audition and **buy only what they want**

The idea, in short, is to let people search, audition and buy only what they want. The problem of having to buy a 3GB sample pack to get the 300MB of samples you actually want is a real one, though it's not really the developers' fault. The only alternative to selling whole collections is to make a searchable, micro-payment driven online system. That requires a lot of work, but it's what Splice has done. You pay $7.99 a month, and for that you get 100 download credits, with samples costing one credit each.

The details

In order to download sounds, you have to install a small application, and this is the same one that is used for syncing DAW project data, though the two tasks are separate things. A folder is created on your computer where downloaded



The web interface is slick and user-friendly, making the search process within Splice a pleasure

files are stored, and you can audition and search them here, as well as revealing them in the file system. Files can be dragged from the Splice app straight into your DAW. You are encouraged to manage sounds in your online account via a browser and it's possible to create 'Repacks', which are essentially collections of sounds that you can manage manually: electro kicks, piano loops and so on. This works well, though collections don't seem to sync to your computer, so they're easier to manage online. The samples are downloaded when clicked, but any Repacks that you create online aren't mirrored as named folders on your computer, which would be a nice addition. Repacks can be shared online with other users, a bit like curating a playlist for Spotify, and Splice has some artists already doing this.

The process of searching for samples is really slick and benefits from a web interface that's very responsive, laid out clearly and easy to use. At the top level, you have multiple search options and you can search everything using keywords, or filter down by categories such as drums, percussion, pads and so on. Then there are genre-based categories: dubstep, house, techno and the like. Click into any of these and a selection of packs can be explored, each one curated and themed, and with a full demo track playable by clicking on it. From here, or from within a pack, you can filter down still further by selecting or deselecting tags at the top of the section. As well as the obvious tags, you get stuff such as one shots and loops, and the ability to filter by BPM or key. This all works extremely quickly and makes a lot of sense, which is vital if you're going to base your system on searching for specific sounds. The tagging seems thorough and the loading fast. This all works from mobile devices, too, and since you're signed into your account, stuff that you buy or organise on the move will be reflected on your studio machine as well.

The samples are excellent, as you'd expect with **partners like Loopmasters** and Prime Loops

For any individual sound, you get a preview button to play it, and triggering a sound will cause any other playing sample to stop, which is really helpful in saving time and confusion. You can flag samples, download them directly and add them to a Repack by using a shortcut or dragging and dropping. Selecting more than one sample illuminates a 'download all selected' type of button, and there's also a randomizer sort button that will show you a random selection of sounds if you like. The computer arrow keys can be used to move up and down sample lists and preview them automatically, which again will save you a huge amount of mouse clicking.

Sounds good

The samples are excellent, as you'd expect with partners like Loopmasters and Prime Loops providing a good chunk of the very varied and large library. The difference here is that you're much more likely to be able to find the stuff you want, and find it quickly, than if you were having to plough through hundreds of sub-folders on your computer. In the same way that you might not bother buying an album if it needed shipping from overseas, you would be much more likely to listen to it on Spotify. In the digital era, convenience is king and this applies just as much to music producers as to regular consumers.

Splice Sounds is a great idea and genuinely solves what has become a significant problem with the sheer number of sample collections on the market. Namely, how can you be sure you will use even half the sounds you're paying for? Piracy is an issue, too, and just like with online music streaming services and things such as Netflix, content providers are increasingly moving towards a subscription-for-access model. It's happening in the music tech world, too – look at EastWest's Composer Cloud, SONAR's subscription model and other developers such as Adobe's Creative Cloud. And people seem to be open to this approach, as long as the subscription price is reasonable and there are no penalties for cancellation. In the case of Splice Sounds, the pricing tiers seem very sensible when you consider the range of sounds on offer and the quality of the user experience. Expect to see more music technology go down this path. For now, Splice is leading the way for sample users. **MTF**

Alternatives

The alternative is what's always been there, which is to buy conventional sample packs. There's nothing wrong with that, and some people may prefer it. You often get multi-format samples for example, with ReFill, REX and ACID versions of the same sounds, which you don't yet get with Splice. You also get immediate access to everything you have bought, with no download limit. The two have fundamental differences of approach, however, so they're not directly comparable.

MTF Verdict

+ Excellent way to pay for just the samples you need
+ No wastage
+ Very well designed
+ Extremely slick and responsive
+ Enables work across devices
+ Easy to search
+ Tagging is great
+ Keep the samples you download
+ Library expanding constantly
+ Can curate collections online
+ Great variety and quality of samples and loops from some big names
+ Priced sensibly

- WAV format-only at present
- 300 downloads per month is the current limit
- Repacks don't sync to your computer

This could be the future of buying and using samples. Smooth and easy to use, with an ever-expanding selection of sounds.

9/10



NATIVE INSTRUMENTS Una Corda

Alex Holmes checks out a veratile sampled-piano instrument from Native Instruments that will add the ethereal to your material...

Details

Price £129
Contact via website
Web www.native-instruments.com
System Requirements Mac OS X 10.9 to 10.11, Intel Core 2 Duo, 4 GB RAM Windows 7, 8, or 10 (latest Service Pack, 32/64-bit), Intel Core 2 Duo or AMD AthlonTM 64 X2, 4 GB RAM Free Kontakt 5 Player or Kontakt 5 (version 5.5.1 or higher)

Features

- Prepared piano instrument for Kontakt 5 and Kontakt 5 Player
- 10GB of audio
- Three Kontakt Instruments, with 100 snapshot presets
- Clean, plus felt and cotton preparations
- Deeply sampled from a one-of-a-kind handcrafted instrument

Piano instruments are like drum samples for the modern producer; you'll likely have more than you need of various styles, sizes and quality that you've collected over the years. Una Corda from Native Instruments is a little different, though, and offers up a deeply sampled version of a unique piano, hand-built by David Klavins in collaboration with composer Nils Frahm. It only has one string per note, which gives a much purer sound than the usual three strings found on your average piano, and allows the user to easily slot in different prepared materials to dampen and affect the way the hammers hit the strings.

Be prepared

The technical side of this massive 10GB Kontakt Instrument has been put together by Uli Baronowsky and Galaxy Instruments, who are also responsible for The Giant, and Definitive Piano Collection for Kontakt. The one-of-a-kind original instrument was recorded using an array of mics and kit in Saal 3 at the Funkhaus in Berlin, a famous old broadcast studio in East Germany. You get three different Kontakt patches with the pure piano sound, a prepared piano with felt between the hammer and string, and one with cotton, plus 100 incredibly varied snapshot presets saved across all three versions. Essentially, the pure sound is closest to a normal piano, with a more resonant and clean tone; the felt has a more gentle, electric piano-esque sound with softer attack; and the cotton is more percussive, with hints of harpsichord.

Where things get really interesting is in the huge tweakability afforded by the Kontakt GUI. There's a basic front page with tone, dynamics and space, and three more for deeper sound design.

More than meets the eye

The first page is the Workbench, which allows you to sculpt the primary characteristics of the instrument. Here you can add harmonics, reverse samples, control the tonal depth, and add a second fabric to the prepared piano sound. You can also choose from a long list of ambient background noise, dial in mechanical and pedal noise, and even add in the sounds of the pianist moving and breathing. These interesting textures can even be brought to the forefront by removing the main piano sound. Next, you have the Response section where you can control how the instrument responds with controls for dynamics and overtones, and finally, you have the Finish section. It's here that the sound-design opportunities really open up, as you can choose from a large number of vintage-effects-chain and convolution reverbs. You can give the instrument a shimmering electric piano or toy-speaker effect, then feed it through simple reverbs or complex, swirling space convolutions to create some incredibly deep, pad-like tones. You'll also find controls for EQ, compression and saturation, transients and stereo image. It's the more ethereal textures where Una Corda really shines. There's a richness and depth to the instrument that sounds infinitely beautiful and cinematic. The flexibility of the GUI and excellent convolution options make this a great sound-design tool for soundtracking.

Alternatives

If prepared pianos are your thing, then you could check out the IRCAM Prepared Piano from UVI ($399), which is more expensive, but has 45 different preparations. Alternatively, there's VSL Prepared Piano (€65) at a much cheaper price, but neither has the deep, ethereal quality of Una Corda.

Strike a chord

Una Corda is more expensive than NI's other piano instruments, but it's also more versatile. It's been created with love and attention, and the results are often breathtaking. This is an inspiring instrument that oozes class and with very few tweaks can be morphed from a gentle, upfront piano, to a swirling, dissonant and clunking beast. **MTF**

MTF Verdict

+ Incredibly rich and beautiful sound
+ Surprisingly versatile sound sculpting
+ Well-thought-out GUI with deep-but-easy editing

- More expensive than NI's other piano instruments
- Not always completely obvious what certain parameters are doing to the sound

Effortlessly atmospheric and beautiful prepared piano instrument. It's not cheap, but there's a surprising amount of sound sculpting possibilities buried under the lid.

10/10





NOVATION/BLOCS
Wave iOS

Innovation MusicTech

Wave hello to a new kid on the iOS music-making Bloc. **Andy Jones** ponders whether Novation's new Blocs brand could usher in a fresh era of composition for Apple-based music producers…

Details

Kit **Blocs Wave iOS**
Manufacturer **Novation**
Price **£3.99/$4.99**
Distributor **Novation/Blocs**
Contact **via website**
Web **www.blocs.cc/blocswave**

Key Features

- iPhone and iPad ideas/composer
- 8 sound combinations
- includes 6 sound packs (300 sounds)
- 12 additional sound packs available (more to come)
- uses 3D Touch on iPhone 6s
- Real-time tempo and pitch change
- Low-latency recording
- Import and export audio via Mail, AudioCopy, AudioShare

Novation seems to be doing its utmost to surprise us these days. When we expect one thing to come along, we get another. Circuit was a Groovebox for the 21st century. I said *a Groovebox for the 21st century*, for pity's sake. Who thought we needed one of those 12 months ago? Turns out we did, and it's bloody amazing.

Now we have Blocs, a completely new brand and its first product: Blocs Wave for iOS. It's an arranger/composer app, of sorts. It's not quite what I thought it could be at first. Nor is it quite what I wanted it to be five minutes after that. But an hour into playing with it, it turns out to be something I didn't know I wanted it to be, but now do want it to be! Just like Circuit then. Typical Novation. Or Blocs. Whatever!

On your starting Blocs

We would dearly love something like Ableton Live for the iPad and initially, I thought that Blocs Wave might well be just that. But while it's not, it does have elements of Ableton's finest DAW that make it quite beautiful and usable in its own right.

So Live is what Blocs Wave isn't (yet) but what it *is* is an arrangement tool, and an ideas cruncher. You start with someone else's ideas, 300 of them across six sound packs, for hip hop, house, indie, dark trap and more, and you arrange whichever ones you fancy into eight 'blocks' to come up with your own tunes. So far, so sample arranger, but fortunately, there's a bit more to it than that. And potentially a lot more…

So each one of Wave's sounds will play in time and in pitch with another one, so everything will sound great when combined – indie with trap, house with hip hop and so on (okay, it won't always sound *absolutely great*, but you know what I mean). So, if nothing else, we have a great way of mixing musical ideas up in a very easy and intuitive way.

That's because arranging sounds is very easy. You use a hexagon-shaped grid (the Discover tab) with your six song elements – bass, drums, melodic, percussion, vocal and FX – and you use it to fill the eight slots in your main song/project. You can then audition each sound from the six sound packs in the next two tabs, either by sound-pack type (genre) or sound type.

Press Auto, and ideas are randomly placed together. You can easily ditch a sound by replacing it with another (or undoing in case you make a mistake), plus easily mute and unmute tracks by swiping up and down. The programmers have thought this through; the swiping and touch are so well implemented, you don't realise you're doing it after a while, as it becomes so second-nature.

However, as easy as it all is to arrange, of course we all know that the real beauty and joy of composition is crossing the line from messing with someone else's ideas to creating something that you can truly call your own. Fortunately, the wave editor makes this part a reality, as does some pretty impressive pitch-and-time adjusting, where you simply select your key, or plus and minus your tempo value to adjust. Both approaches work very well, particularly the tempo.

But wave editing is Blocs Wave's real strength, and utilises all that iOS touch power for ease. Open a sample, shorten it, increase and decrease volume – it's all possible, and everything you do just fits with what you already have. And while editing audio is the best feature, the most important one is the ability to import your own audio and export what you come up with.

Now this is key, because I like Blocs Wave and I want to work with it, but with more *of my stuff*. I want it to connect up with my Dropbox (Drop Blocs?) and to batch-add a load of my loops. (Since this review Blocs has improved the file management system in this regard). Maybe I'll also want Blocs Wave to chain to the outputs of Logic and Live and for all of my ideas to flow into Blocs, so I can experiment with them on the move? So, clearly, I want some things that it doesn't do… yet.

But this early days and I think the important part is done – the editing, the ease-of-use, the real-time pitch and tempo stuff – and the fact that I'm even thinking ahead to future versions means Novation/Blocs has done this well. Indeed the company seems to be heaping loads into Blocs as we speak with real-time recording now an option and improved importing and exporting making it an even more vital app. **MTF**

MTF Verdict

+ Easy. You don't need instructions
+ Good range of sound packs
+ Like the newsfeed that offers more of them (for free!)
+ Can import and export audio
+ Great touch design and workflow
+ Mix trap with hip hop

- Mix trap with hip hop!
- A bit dark

Blocs Wave could be what we've been waiting for and, at present, is a great ideas-mangling app. Loads of potential, get your loops in!

8/10



SPITFIRE AUDIO

HZ2 Los Angeles HZ3 London Soloists

In conjunction with Spitfire Audio, Hans Zimmer and his engineering team have produced two more cinematic percussion libraries to complete the HZ trilogy. **Keith Gemmell** gets under their skin…

Details

Product
HZ2 Los Angeles, HZ3 London Soloists
Price **£239 each**
Contact **via website**
Web **www.spitfireaudio.com**

Features

HZ2
- Jason Bonham on unique Vistalite kit
- 3 LA recording locations
- Grammy Award-winning engineers
- Numerous mic positions
- Kickstart engine

HZ3
- Frank Ricotti on percussion
- Recorded at AIR Sudios, London
- Grammy/ Oscar-winning engineers
- Several mic positions
- Numerous articulations

A while back we reviewed HZ1, a truly epic percussion library collaboration from Hans Zimmer and Spitfire Audio. It featured four percussionists and was recorded at London's AIR Studios. We now have two follow-up libraries – HZ2 and HZ3 – produced by Hans himself and the same team of engineers that worked on the original.

HZ2 features Jason Bonham, recorded at three different Los Angeles locations – 20th Century's Newman Stage, the Sony Stage and The Cathedral, a large concrete space on Hans Zimmer's Remote Control Productions lot. Hans has also contributed extra material himself, in the form of some modular-synth drum programming.

HZ3 is an expansion of the first library; a collection of Hybrid Cinematic Widescreen percussion recorded, like the first library, in London's AIR Studios, and performed by renowned percussionist Frank Ricotti.

HZ2 features **John Bonham**, and **HZ3** is performed by percussionist **Frank Ricotti**

Battery power

Looking at HZ2 first, at the top tier we have two main folders, a Stereo Mix and an Artist Elements section, each containing three folders for each engineer's specific take on Jason's unique Vistalite kit. The engineers are Alan Meyerson, Geoff Foster and Steve Lipson. Inside those, there are three more folders for each of the locations previously mentioned. And within each of those is a full performance kit with basic GM mapping and four folders containing patches for cymbals, kicks, snares and toms – quite an arsenal.

As expected, it's a superb-sounding kit and you can adjust its sound using close, overhead and room mics, plus a special gated room mic. There's also a

Beat mapping

Spitfire now features a new type of interface for percussive instruments and libraries. In HZ2, there are three 'takes' on Jason Bonham's kit, one by Alan Meyerson, one by Geoff Foster and a third stereo mix. All the kit pieces are displayed in the centre and, using a panel to the right, you can configure them individually and map them to a MIDI controller.



Sound articulation

After rummaging around in HZ2, we found a bonus folder with bass drum, toms and surdos, seen here. Note the detailed articulation (highlighted within the red triangle).



Familiar look and feel

The HZ3 (solo instruments) GUI is very similar to that of HZ1 (ensembles). An overview panel provides a quick and easy way to load articulations, mic mixes and a set of 'easy tweak' controls.



Deeper editing

A deeper HZ3 general controls panel allows tweaking of the general settings, round robins, pitch options, velocity workings and so on.



stereo mix. There are plenty of controls, filters and so on, many of which are similar to HZ1, so if you own that library you will feel pretty much at home here. Our only grumble is with the GUI – tiny black text on a grey background is tricky to read in places.

The performance kits are superb and instantly useable as everyday 'go to' rock drum kits. However, you can go much deeper than this by loading the individual kit pieces to further tailor the sounds to your own requirements. A snare patch, for example, provides a selection of sticks, bamboo sticks and mallets, all with snare on or off.

Zimmer's synth-drum patches are a real treat: powerful and dangerous sounding and bearing no resemblance to the tacky 'syn drums' of the 80s. There's plenty of variety here: anything from dry up-close hits to big roomy cinematic thunderclaps, all very exciting and immaculately recorded.

Percussive Paradise

Recorded on the opposite side of the Atlantic Ocean to HZ2, London Solos (HZ3) is an expanded selection of the solo content found in HZ1, which was mainly an ensembles library. It's structurally similar to HZ2, with two top-tier folders – Artists Elements and Additional Mics. Apart from the Bucket and Snare, the instruments might be unfamiliar, with names like, Djun, Tombek and Dohl. If you don't know them, as soon as you see the pictures that are displayed in the GUI, everything makes sense and it soon becomes obvious how and when to use them.

Mics are mostly close, front and surround perspectives, each with a very distinctive sound. If you need more, there's that Additional Mics folder. Articulations are many. For example, the Dohl has Both Sticks, Top and Bottom Sticks, Both Hands and Top and Bottom Hands, all laid out in an intuitive fashion on the keyboard, making it enjoyable to play. Again, like the Jason Bonham library, there's a wealth of controls.

As expected, there's plenty of leap-out-of-your-cinema-seat material on offer here, but it's not all blood and thunder. There's a good deal of quieter, more delicate, detailed percussion to be found if you root around a bit.

Teamwork

Both of these libraries have been beautifully recorded and immaculately produced and offer a huge array of drum hits and sound-shaping possibilities. Hans Zimmer, as we all know, is primarily an Oscar-winning composer of countless films but he also has lots of engineering experience behind him. Over the years, he has developed a truly unique approach to cinematic sound production that has been widely imitated. What you get here, though, isn't a copy. It's a recreation. He has used the same team of engineers and performers that were responsible for many of his film soundtracks, and even used many of the same recording locations. So, if you want the real thing, these two libraries will serve you well. MTF

If you want the real thing, both of these beautifully recorded libraries will serve you well

Alternatives

There are so many percussion and drum libraries out there that choosing a good alternative can become quite confusing. For creating epic cinematic percussion, Native Instrument's Damage is a good choice. Drums Of War from Cinesamples has been used in many a blockbuster, as has True Strike from ProjectSAM. If you want some of the work done for you, NI's Action Strikes has many designed rhythm sets, making it easy to drum up a thunderous cinematic atmosphere.

Do you really need this?

One good reason for buying HZ3 is that it complements HZ1 perfectly. It features solo instruments as opposed to the ensembles found in HZ1. They have the same GUI and scripting and even share the same manual. HZ2, being kit-based is a different prospect and most musicians and producers reading this will have at least one sampled kit at their disposal. That said, Jason Bonham's kit has been so well recorded and immaculately produced that we're fairly confident that if you buy HZ2, it may well become your favourite virtual drum kit.

MTF Verdict

+ Great engineering
+ Great sound
+ Outstanding drum kit (HZ2)

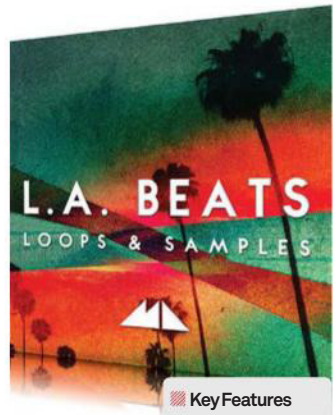
- GUI has tiny text
- Online manuals only

These two libraries are both outstanding, each in its own way. Jason Bonham's kit has been recorded as a cinematic product, but it also makes for an excellent standalone virtual drum kit. Veteran percussionist Frank Ricotti's skilful performances have been beautifully captured by an award-winning team of engineers, including Hans himself. First-class material all round.

10/10

L.A. Beats

Publisher **Mode Audio**
Price **£18**
Contact **info@modeaudio.com**
Web **www.modeaudio.com**



L.A. Beats is a crunchy new collection from Mode Audio that features resonant analogue-synth riffs and skewed, chunky beats from 70 to 86BPM. The bulk of the pack includes 156 loops covering drums, hats, percussion and fills, synth bass, glistening arps and leads, jazzy organ and keys chords, guitar licks, and vinyl noise beds. It's an eclectic mix of hip-hop and trap sounds, with an energetic and loose feel. In addition, there's a folder of tail samples that can be used with their corresponding loops, or as samples in their own right. There are also 139 MIDI files that offer a great way to use some of that live feel with your own instrument sounds, and a decent folder of 88 beefy, saturated drum, percussion and FX hits. MTF

Key Features
- 613MB of 24-bit audio
- Hip-hop and trap inspired samples
- 156 drum, bass, synth, keys, guitar and FX loops
- 139 MIDI loop files
- 88 chunky drum, percussion and FX loops

MTF Verdict

Despite some occasionally heavy-handed production finish, this is a versatile and characterful collection of loose-sounding loops and hits.

8/10

Drum Tools 02

Publisher **Wave Alchemy**
Price **£49.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Drum Tools 02 from Wave Alchemy has been two years in the making, and contains 4,025 individually processed drum hits. A range of drum machines, analogue synths, Foley recordings and acoustic sources were used for the sounds, which were then processed through boutique outboard gear. Many of the hits were recorded through an Ampex 1/2-inch tape machine, and 12-inch vinyl for added warmth. Luckily, this large collection has been sensibly organised into 36 kits, plus 43 instruments for different kicks, claps, hats, percussions and other hit types. Everything here has a vibrancy, punch and loudness that will cut through a mix like a knife. There's also a sensible mixture of solid, dependable sounds, and more experimental hits for added variety. MTF

Key Features
- 4,025 24-bit analogue and Foley drum hits
- 79 instruments for EXS24, Kontakt, NN-XT, HALion, SFZ
- Ableton Live 9 Pack
- Processed, Tape & Vinyl versions
- Perfect for techno, EDM, d'n'b, dubstep, deep house, trap, electronica

MTF Verdict

A phenomenal collection of vibrant and powerful drum hits that should be an essential purchase for anyone looking for cutting-edge, mix-ready drum sounds for electronic music.

10/10

Lost In Time Dub House

Publisher **Loopmasters**
Price **£29.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Lost In Time Dub House sees producers Ralph Lawson and Carl Finlow delve into their collection of vintage gear for an effects-heavy journey through atmospheric house. The pack contains around 777MB-worth of 24-bit audio in a range of formats, including tubby analogue bass sounds, echoing synth stabs, and some retro-sounding, if slightly lacklustre, drum loops. Although the quality of the synth sounds is excellent, it feels like a missed opportunity not to have the loops a little longer, with more evolving modulations and effects. Where this pack really shines is in the 81 sampler patches, with plenty of well-crafted analogue-sounding synth and bass instruments, plus a handful of drum and FX hits. If you explore, there are some fantastic sounds tucked away here, it's just a shame the drums don't quite rise to the same standard. MTF

Key Features
- Atmospheric house drums, bass, synths & FX
- 777MB-worth of 24-bit audio
- 81 sampler patches for EXS24, HALion, Kontakt, NN-XT, Kong, SFZ
- 98 to 130bpm
- Recorded and processed using vintage gear

MTF Verdict

Plenty of superb-sounding, dubbed-out analogue synth riffs and hits, plus some slightly average authentic vintage drum loops.

8/10

Undercover Funk Vol 2

Publisher **Loopmasters**
Price **£29.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Undercover Funk Volume 2 is a collection of royalty-free funk construction kits, in your choice or WAV and Rex2 or Apple Loops and Rex2 formats. There are 25 kits in total, in a range of tempos and keys, with an overall sound akin to a Steven Soderbergh heist-film soundtrack. From laid-back 80bpm soul grooves, to faster 140bpm bank-job funk, each folder is packed with individual instrument loops, plus files with all the musical elements combined, but no drums. There are often several variations on the beats, and main instrument parts, making it easy to build up more comprehensive arrangements. You can expect to find thumping drums, live funk bass, and reverb-drenched wah guitars, plus plenty of Rhodes, pianos, organs, percussion and more – plus an extra folder of processed beats. MTF

Key Features
- 25 funk/soul construction kits
- 361 loops, over 1GB-worth of 24-bit audio
- 80 to 140 bpm
- WAV and Rex2 or Apple Loops and Rex2
- Live drums, guitar, bass, keys and more

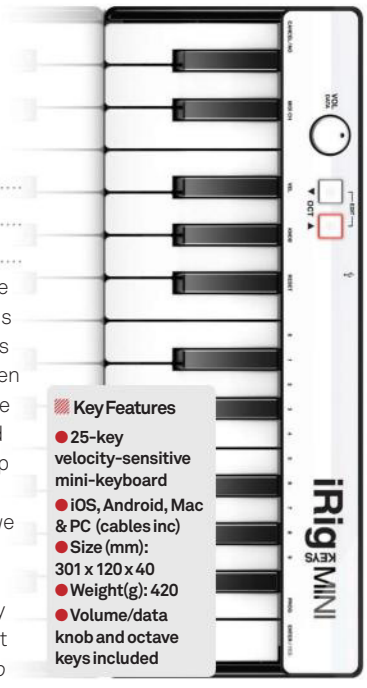
MTF Verdict

Expertly written, played, recorded and mixed construction kits, which would be a superb resource for producers and media composers looking to funk things up.

9/10

iRig Keys Mini

Manufacturer **IK Multimedia**
Price **€97.59**
Contact **via website**
Web **www.ikmultimedia.com**



Key Features
- 25-key velocity-sensitive mini-keyboard
- iOS, Android, Mac & PC (cables inc)
- Size (mm): 301 x 120 x 40
- Weight(g): 420
- Volume/data knob and octave keys included

iRig Keys Mini boasts being one of the most portable keyboards on the market. It certainly feels that way – lightweight, cheap even – but it is incredibly simple to use and plug 'n' play (almost). We had to set its MIDI channel to get it up and running (quite easy via the website instructions) and then we had it triggering Gadget and Arturia's iMini in no time. And it's surprising how much we actually used it – as good as the iPad is at playing music, sometimes you *do* want the feedback from a proper keyboard, while playing it. While this is obviously not full sized and a little too plastic-y, it does a reasonable job and also ships with a bundle of freebies that includes iOS, Mac and PC piano and SampleTank software. MTF

MTF Verdict

Feels a little cheap, but does the job and comes with a good bundle of software – a nice mobile add-on.

8/10

Record Label Marketing

Manufacturer **Focal Press**
Price **£34.99**
Contact **via website**
Web **www.routledge.com**



Key Features
- Chapters on label operations, record label finances, social media, radio
- 471 pages
- Written by Amy Macy, Clyde Rolston, Paul Allen and Tom Hutchison

It's the third edition of a book that probably needs to be updated yearly (monthly if possible), as this is a subject area that moves so fast that it must be hard for labels to pin down and follow. It has to be said, though, this book does its best... but the fact that it weighs in at 470-odd pages shows there is no clear way to make money from your music today. Subjects such as running a label, branding, radio and social media are covered in great depth, but syncing for film and TV is not – which is a shame, as it's surely one potential growing source of income for any label. It also has a US-centricity which may not help our European audience. MTF

MTF Verdict

A fast-moving subject that will inevitably quickly fall foul of time and some detail is missing on potential revenue streams, but otherwise great depth, and a glimmer of hope for musicians and labels in search of cash.

7/10

Blue Mo-Fi

Manufacturer **Blue**
Price **£274**
Contact **via website**
Web **http://blue-headphones.com**

Choice 9/10 MusicTech



Key Features
- Built-in 'audiophile' amp
- 12-hour battery life (4-hour charge)
- Three analogue amp modes
- 50mm, fibre-reinforced dynamic driver
- Over-ear design for isolation
- Multi-jointed headband design
- Imp: 42 ohms
- Multi-jointed headband design
- Frequency response: 15Hz-20kHz
- Weight: 466g

We picked the Mo-Fi headphones as one of our 'six products of NAMM that you may have missed'. If we're honest, we based that on their looks and ground-up design. For the former, you can't blame us as they look great, but for the latter, Blue has taken the headphone concept and turned it on its head – rather like it attempted to do with the microphone. The Mo-Fi headphones have been designed with maximum comfort in mind, and an audiophile experience helped by an onboard amp, which Blue says is doing to headphones what active monitors did to studio speakers. Quite a claim...

The headphones themselves are relatively heavy compared to some we've looked at of late, but they fit incredibly well around your head. You might not think so at first, as they're quite compact when you remove them from their box. However, the headband extends from a compact, sprung storage position outwards, and if this wasn't enough, both phones extend downwards, too. This is the clever bit; rather than notching them down as you possibly would with some headphones (always hit and miss, at best), you gently pull the 'phones away from you and they drop. Very cool, but hard to get across without seeing it in action!

Sound-wise, these are right up there with our reference ATs and they translated a mix we were working on very well indeed – all of the detail we expected was there. They can be loud with the amp on and this tends to crank the bass, but they are always comfortable – the extra weight not being an issue – and noise isolation is very good.

Small niggles: the amp switch crackles a little, the short cable is very short (you get a longer one, too) and you will need to charge for four hours (for active mode). But none of these are massive no-nos, especially given the great all round sound vs. comfort vs. noise isolation. Pricey? Yes, but do we have to say again about spending money on what you're listening to your music with? Thought not.

So we may have picked Mo-Fi out as one of our 'NAMM Six' based on their looks, but their performance is right up there with the design. Great cans. MTF

MTF Verdict

Look and sound great, but the onboard amp and unique design set Mo-Fi apart from most of the rest.

9/10

Ultimate Logic Pro Super Pack



Publisher **Niche Audio**
Price **£79.95 (requires Logic X 10.2.1+)**
Contact **info@loopmasters.com**
Web **www.loopmasters.com/labels/71-Niche-Audio**

Key Features
- 8 packs of production-ready cross-genre kits
- Formatted for Logic X
- 76 Projects, 29 channel strips, 347 EXS patches
- 1,936 24bit/44.1kHz samples
- Includes one-shots, plus riffs and MIDI files

Choice 9/10 MusicTech

Eight of Niche Audio's production-ready dance-music kits are now available in a bumper collection, formatted for Logic X with 76 projects, 29 channel strips, and 347 EXS patches. This includes sounds, native Logic processing, and MIDI riffs from Bass House and Garage, Dub & Reggae, Future House, Naked Techno & Sublime Techno, plus crunchy processed drums from Planet 808 and Planet 909, and effected vocal one-shots from Vital Vocals. They form a useful tool for electronic-music producers, and offer decent MIDI patterns to edit and make your own, while the analogue techno stabs and hits are particularly tasty. The content is spread across multiple packs, so you may wish to re-organise it to make browsing the EXS instruments easier. **MTF**

MTF Verdict

£200 worth of superb dance-music drum hits, basses, stabs and FX for £80. You can't argue with that!

9/10

Lynx Kinetic Drum & Bass



Publisher **Loopmasters**
Price **£24.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**

Key Features
- 624MB-worth of 24-bit audio
- 207 loops, 228 one-shots in Wav, Apple Loops, Live Pack or ReFill formats
- 56 patches for Kontakt, EXS24, NN-XT, HALion, SFZ
- Bass, drums, percussion, FX and music loops
- Written and produced by Lynx

Hospital Records' artist Lynx returns to Loopmasters for another high-octane d'n'b pack full of speaker-destroying basses, breaks and old-school inspired music loops, with a total of 207 loops, 228 one-shots, and 56 sampler patches. It's clear from the off that basslines and drums are Lynx's forte as the heavy, throbbing bass riffs are some of the best we've heard for some time. The drums, too, are expertly programmed with a good balance between originality and simplicity to make sure they work on the dancefloor. There are also some good, atmospheric music, FX and percussion loops, although these can't quite match the quality of the drums and bass. Finally, you'll find a collection of raw drum hits, bass and sub instruments, FX and music hits to round things off. **MTF**

MT Verdict

Some phenomenal, energetic-sounding bass and drum loops, backed up with a reasonable collection of music and FX riffs and one-shots.

8/10

u-he Diva Explained



Publisher **Groove 3**
Price **$15 for one-month streaming access to whole site ($25 download)**
Contact **via website**
Web **www.groove3.com**

Key Features
- Tutorial on u-he Diva
- 16 chapters, two hours, nine minutes run time
- Stream online
- Covers all areas of the synth, plus sound-design tips
- Presented by Sami Rabia

Choice 9/10 MusicTech

This new tutorial from Groove 3 and synth specialist Sami Rabia aims to uncover all areas of u-he's analogue-modelled instrument across 16 videos totalling just over two hours. Rabia begins with two in-depth chapters covering all the different oscillator and filter models, before moving on to the envelopes and LFOs. We then have a further six chapters on elements such as Trimmers, play modes, the Effects section, Modulation Sources and the Modifications Tab. Rabia then rounds things up with several videos looking at building pads, basses, leads, FX and arpeggiated sounds. Although earlier sections are more aimed at explaining the various parameters and functions, everything is still tied into how it sounds with good audio examples. The final chapters showcase some excellent sound design, using fairly advanced modulation techniques. **MTF**

MTF Verdict

A deep but easy-to-follow tutorial, with good audio examples, which left us feeling inspired to explore Diva further.

9/10

StereoSavage



Publisher **Plugin Boutique**
Price **£59.95**
Contact **via website**
Web **www.pluginboutique.com**

Key Features
- AU, VST, AAX Stereo control plug-in
- Control stereo width and pan
- Three effects for generating width from mono sources
- Syncable LFO with multiple destinations
- Bass Bypass feature to aid mono compatibility

Excellence 10/10 MusicTech

StereoSavage is a new effect from Plugin Boutique that combines multiple techniques and tools in a single plug-in. You can place sounds in the stereo field using the Width, Pan and Rotation controls, retain a solid, mono low end with the Bass Bypass dials, and control the phase and L/R of the signal via the Input Routing section. There's also an Effect section that includes an excellent Vox pitched chorus-style effect, a Haas Delay effect, and an Expander that uses early reflections to add depth. To top things off, there's a crisp goniometer and input/output meters, plus an LFO with six shapes, sync, speed and phase controls, which can be used to control an array of different parameters for subtle movement and more wild effects. The only thing missing is a mono output button to test the mono compatibility of your settings. **MTF**

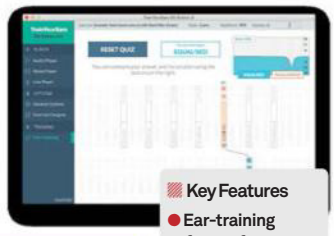
MTF Verdict

An incredibly useful tool that combines multiple stereo effects into one concise toolbox. A quick-and-easy plug-in for adding beautifully subtle or more extreme depth to your tracks.

10/10

TrainYourEars EQ Edition 2.0

Publisher **TrainYourEars**
Price **€89**
Contact **via website**
Web **www.trainyourears.com**



Choice 9/10 MusicTech

Key Features
- Ear-training software for PC and Mac
- Select from audio files, audio input or white/pink noise
- Loads WAV, AIFF, MP3, OGG and M4A files
- Easy to use layout with 13 EQ exercises
- Program your own advanced listening lessons or import new ones

TrainYourEars aims to speed up your learning process by presenting you with hundreds of EQ choices in a quiz format. You can choose to load in audio files in a variety of formats, use a live input for connecting to iTunes, internet radio or other players, or simply use white or pink noise as the source. You then choose from 13 different exercises, including things like picking out specific frequencies and Q factors, and distinguishing between shelfs and cuts. You can also design your own complex exercises, with an additional set available for free. It's not cheap, but it's slick and easy to use, and we found our scores improving after just 15 minutes. **MTF**

MTF Verdict

A simple but highly effective way to improve your EQ skills, with well-thought-out exercises that mirror production decisions encountered in the real world.

9/10

Sinister Strings

Publisher **Big Fish Audio**
Price **£46.95**
Contact **Time+Space 01837 55200**
Web **www.timespace.com**



Key Features
- Sinister-sounding live strings library
- Over 650MB of 24-bit audio
- 318 files WAV files
- Eight patches for Kontakt (four articulation patches, four slide patches)
- Violin, Viola, Cello and Ensemble performances

Sinister Strings from Big Fish Audio is a collection of creepy-sounding string effects with violin, viola, cello, and ensemble samples. There's around 650MB of audio, including raspy sustained notes, sustained notes with subtle bends and various slides, all with and without tremolo. On top of the WAV version, there's also a package with eight patches for Kontakt to aid with previewing and editing the sounds; we found the Kontakt interface to be a little clunky, as it involves a lot of different pages just to edit the effects sends and the individual effects. Not massive or especially broad, but a useful asset for adding an unnerving element to your productions. **MTF**

MTF Verdict

Clunky Kontakt interface aside, this is a well-recorded collection, which would form a useful tool for film and media composers looking to add some tense sounds.

8/10

VIBES Vol. 1 Soul Trax

Publisher **Loopmasters**
Price **£34.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Key Features
- 1.5GB worth of 24-bit audio
- WAV and Rex2, or Apple Loops and Rex2 format
- 346 loops of live drum, bass, guitar piano, keys and more
- 90 to 165BPM
- Inspired by classic Motown and artists such as Diana Ross and the Commodores

Soul Trax is the first in a new series from Loopmasters and features 20 construction kits inspired by labels like Motown, Tamla Records and Gordy Records, and by artists such as Diana Ross, Marvin Gaye and the Commodores. Each of the kits contains a number of eight-bar drum, bass, guitar and instrument loops, including welcome variations to aid in arrangement. The piano, keys, organ and horn parts work well to fill out the tracks, but it's the drums, bass, shakers and guitars that are the real highlight, with some great, energetic riffs perfect for pop productions. Production is cleaner sounding than a classic Motown record, but the kits are authentically written and carry a solid, feel-good vibe. **MTF**

MTF Verdict

A well produced, recorded and laid-out selection of soul and funk construction kits, which is packed with feel-good riffs and some excellent acoustic drum loops.

8/10

Chilled Trap

Publisher **Loopmasters**
Price **£34.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Key Features
- Chilled-out trap drum, bass and synth loops
- 274 WAV or Apple Loops
- 302 Rex2 Loops
- Over 800MB of 24-bit audio
- 130 to 140BPM

Chilled Trap is a new pack from Loopmasters that takes aim at the more ambient and downtempo side of trap music as opposed to the heavier, in-your-face variety. It's essentially 10 construction kits containing 274 24-bit loops, but is divided into folders for instruments rather than kits, which arguably encourages more experimentation. You'll find fragile pianos, chirping vocal glitches, deep pads and sweeping synths alongside booming 808 subs and tight-sounding drums with plenty of rolling hi-hat patterns. Everything is well programmed, with some intricate little riffs and fills, and the production finish is simple and clean. Although the drum, percussion and bass loops don't really push any boundaries, the music loops folder is packed with plenty of inspiring riffs, with several musically useful variations on each idea played out on different sounds and synths. **MTF**

MTF Verdict

Most of the pack is relatively simple, but some detailed production flourishes and excellent synth riffs make it well worth a look for anyone producing chilled-out trap or downtempo electronica.

7/10

Icicle

Publisher **Producertech**
Price **£29.95**
Contact **via website**
Web **www.producertech.com**



Key Features
- Drum 'n' bass synthesis, mixing and arrangement tips
- Three hours of streaming video
- Uses Ableton Live, Fab Filter plug-ins, and NI FM8 and Massive
- Includes 9 Live projects and 100MB of d'n'b samples
- Written and presented by Icicle

Drum 'n' bass production wizard Icicle has teamed up with Producertech for a tip-filled three-hour course on modern d'n'b production techniques. The tutorial is available to stream from the Producertech website and is divided into 16 chapters covering synthesising and processing drums, creating and resampling complex bass sounds, bussing and mixing techniques, and adding effects and automation. Icicle mostly uses Ableton Live, but also explores the Fab Filter plug-ins and NI's FM8 and Massive, with some great tips on FM synthesis. You also get nine Live projects, plus a varied and decent 100MB folder of samples taken from Icicle's Loopmasters pack. The quality of the video and spoken audio could be a bit better, but this is an excellent tutorial for anyone looking to learn d'n'b techniques from a pro. **MTF**

MTF Verdict

Some great synthesis and production techniques from a d'n'b master who's not afraid to explain how to break the rules to get a big, modern sound.

8/10

Terry Grant Dark Dub Odyssey

Publisher **Loopmasters**
Price **£34.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Key Features
- 2.31GB of 24-bit audio
- Over 500 loops in WAV, Apple Loops and REX2 format
- 80, 100 and 120BPM
- Live bass guitar, drums, pianos, synth bass and pads
- Produced by Terry Grant

Long-time Loopmasters collaborator Terry Grant is back with a massive collection of cinematic soundscapes and dubbed-out beats. Dark Dub Odyssey has over 2GB worth of loops divided into 80, 100 and 120BPM, with your choice of WAV and REX, and Apple Loops and REX formats. You'll find a mixture of live dub bass guitar, reverby pianos, and acoustic drums and percussion, alongside fat analogue bass riffs, electronic beats, and 206 atmospheric, ethereal pad loops. The sound design is excellent and there are some decent loops spread across this giant library. However, the drawn-out pads are a little samey and we found some drums and sounds to be a little dated. That said, some of these textures could work really well in media projects and ambient productions. **MTF**

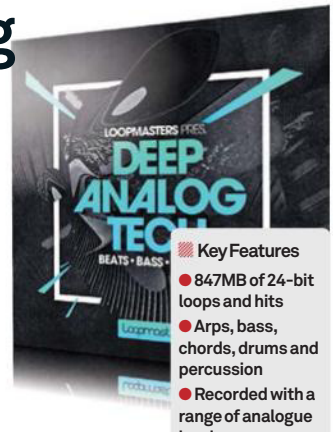
MTF Verdict

Despite some of the loops sounding a little dated, this is a large collection with some deep and atmospheric sound design.

8/10

Deep Analog Tech

Publisher **Loopmasters**
Price **£29.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**



Key Features
- 847MB of 24-bit loops and hits
- Arps, bass, chords, drums and percussion
- Recorded with a range of analogue hardware
- 123 sampler patches for NN-XT, Kontakt, EXS24 and SFZ
- Inspired by the Kompakt label

Excellence 10/10 MusicTech

Sometimes a pack will tell you it's been made on analogue hardware, but you struggle to hear it in the loops. However, Deep Analog Tech from Loopmasters wears its analogue heart on its sleeve, with a rich and varied collection of gritty loops and hits dripping with raw energy. Inspired by the Kompakt label and artists such as Gui Boratto and Agoria, sounds were created using a Roland Juno-60, MS-20 Mini, Roland RE-201 Space Echo, Elektron Analog Rytm and Korg Volca Keys. You'll find folders of twisted arps, throbbing basses, haunting chord progressions and beats and percussion, crunchy drum hits, weird FX, and fat bass and synth hits. Although this pack is aimed squarely at tech house, there's plenty of sonic variety and the quality is exceptional throughout. **MTF**

MTF Verdict

A stunning pack of rich and characterful analogue loops and sounds with interesting quirks and details in every folder.

10/10

PanShaper

Developer **Cableguys**
Price **€34**
Contact **Via website**
Web **www.cableguys.com**



Key Features
- Multiband stereo modulation plug-in
- VST and AU
- Independent curves for all three bands
- LFO with tempo sync and Hz-via-MIDI mode
- Comprehensive preset library and curves

Choice 9/10 MusicTech

PanShaper is a new plug-in from Cableguys that combines elements of its free PanCake plug-in with features from the useful VolumeShaper 4. PanShaper is a three-band stereo modulation effect that allows you to draw or load in complex pan position curves for each band, along with width and volume controls, and LFO control of the overall speed. The LFO can run freely, or sync to tempo or input MIDI for some crazy audio-range FX. An excellent GUI means it's easy to see the waveform position and edit nodes to build complex sweeping patterns, and there are also plenty of presets to try out. We found it excellent for pinpointing and panning specific drum hits in a loop, adding subtle movement to arps and pads, and for pseudo-widening without phase issues. **MTF**

MTF Verdict

A great value plug-in that's easy to use, with a clear GUI and some decent presets. PanShaper is a useful tool for controlling your stereo width in new, creative ways.

9/10

Aiaiai Modeselektor Edition

Manufacturer **Aiaiai**
Price **€250**
Contact **Aiaiai via website**
Web: **http://aiaiai.dk**



Key Features
- High isolation
- Modeselektor carry pouch
- Impedance: 32ohms
- Weight: 238g
- 1.2m cable and adaptor

We love Aiaia headphones, particularly the straight TMA-2 Studio model. The modular nature of these allows other phones of different configurations to be produced, and there are now artist editions available. As fans of uber producers Modeselektor, we like the look of this limited edition model, complete with MS logo on the earpiece as shown. They are out-and-out bass headphones, so loud and proud to be bottom-heavy – don't expect great and accurate mixing, but for listening back to your work or indeed the 'Selektor, these are choice headphones. While we don't recommend buying anything just for a cool aesthetic, we are tempted by these anyway. **MTF**

MTF Verdict

Not for accurate mixing (nor do they claim to be), but these have a great low n loud sound.

8/10

Teeel Presents – Retro Wave



Publisher **Loopmasters**
Price **£24.95**
Contact **info@loopmasters.com**
Web: **www.loopmasters.com**

Key Features
- Synthwave loops and hits
- 1.3GB+ of 24-bit audio
- 137 sampler patches for EXS24, Kontakt, NN-XT, Kong, HALion and SFZ
- 80-160bpm
- Produced by Jim Smith, aka Teeel

Value
€£$
MusicTech

Retro Wave sees producer Jim Smith (aka Teeel) delve into his large collection of analogue drum machines, including instruments from Roland, Simmons and LinnDrum, plus Yamaha DX7, Yamaha CSX1, Roland HS-60, Moog Sub Phatty and Little Phatty synths to create a large library of dreamwave and synthwave samples. There's more than 1.1GB of unison bass loops, analogue drums and bright synth lines inspired by artists such as Kavinsky, Mitch Murder and Chromatics. Although the patterns are fairly simple, the sounds are thick with analogue goodness. You get a huge collection of vintage-sounding bass, synth, drum and fx hits and there are also 18 synth and pad MIDI loops. **MTF**

MTF Verdict

A massive pack of simple but authentic-sounding analogue and digital loops and hits that would be a great resource for anyone writing in electro, synthwave, synth pop and other genres.

8/10

The Great British Recording Studios

Excellence
10/10
MusicTech



Publisher **Hal Leonard Books**
Price **$29.99**
Contact **via website**
Web **www.halleonardbooks.com**

Key Features
- Details on at least 30 studios
- In-depth on 12
- Includes tech glossary
- Layouts, gear lists, quotes, loads of additional info!

Trust the Americans to make such a great job of writing a book about the golden age of UK studios. We're somewhat annoyed that no one asked us, as we seem to have visited just about every one (of those left). There's a lot of detail – technical, floorplans, etc – on each, with more space dedicated to those you might expect (Abbey Road, Olympic, Trident, AIR, etc). We love the depth, with personnel lists, stories, gear lists (yes!), diagrams and even acoustic treatment details. There are photos, of course, and details of the famous recordings. We'd have loved local favourite Real World to have had a chapter, and it's depressing that so many are now not with us, but the effort and research that went into this is extraordinary. An excellent geek and gear-fest of a read. **MTF**

MTF Verdict

It's a book that will become one of the most used in the MusicTech library. An excellent and thorough read.

10/10

Hip Hop Shadows Vol1



Publisher **Loopmasters**
Price **£29.95**
Contact **info@loopmasters.com**
Web **www.loopmasters.com**

Key Features
- Instrumental hip hop loops and hits
- 12 construction kits from 78 to 123bpm
- 816MB worth of 24-bit audio, Wav, Apples Loops, Live Pack, ReFill, Rex2
- 12 sampler instruments for EXS24, Kontakt, NN-XT, HALion, Kong and SFZ
- Produced by Chemo

Choice
9/10
MusicTech

Instrumental hip hop master Chemo has teamed up with Loopmasters for a fresh collection of organic-sounding loops and hits, inspired by pioneering artists such as J Dilla and Flying Lotus. The pack features 12 construction kits of unquantised, live sampled drums; wonky bells; pianos and synths; warm, rounded live basses and more. The tracks themselves are atmospheric, beautiful and steeped in nostalgia, and everything has a loose feel with plenty of glorious little mistakes and human touches. Most of the loops are 8 or so bars long with little variations at the end, and are gently layered with sounds to add detail. You'll also find 12 sampler patches with instrument and drum hits taken from the pack so you can build your own beats or experiment with adding in little quirky hits to your tracks. **MTF**

MTF Verdict

A compact but beautifully crafted pack that's stuffed with character and a human, organic feel that would be great for adding life to hip hop or any other genre.

9/10

MTF Feature | Produce a track from start to finish

8 WAYS TO MASTER A TRACK

Part 5

It's the final part of our 'produce a track from start to finish' feature and it's the big one: how to give your final mix the pro sheen it deserves…



01 Zoom out

In the mixing process, you will be considering the detail: the levels, the effects, the EQ, plus perhaps some automation, including fades and other parameters. With mixing, then, you are very much looking 'inside' at every aspect you can. With mastering, you need to forget that and step back, zoom out, rise above it, look down and consider everything as a large picture with everything you were working on in the mix all joined up as one whole. You need to consider the track – a stereo mix (usually!) – as a single entity and now start to think about it in the context of other tracks on an album or reference material you are trying to put it against and to perhaps sound like (in terms of its sonic character, anyway). If, during this process, you notice some glaring things wrong with the mix, then you'll need to go back into the mix and process it again (or return it to whoever did it). Hopefully, this is something that can easily be done, either by you or the original mix engineer – so that you can consider the whole rather than having problems with the parts get in your way…

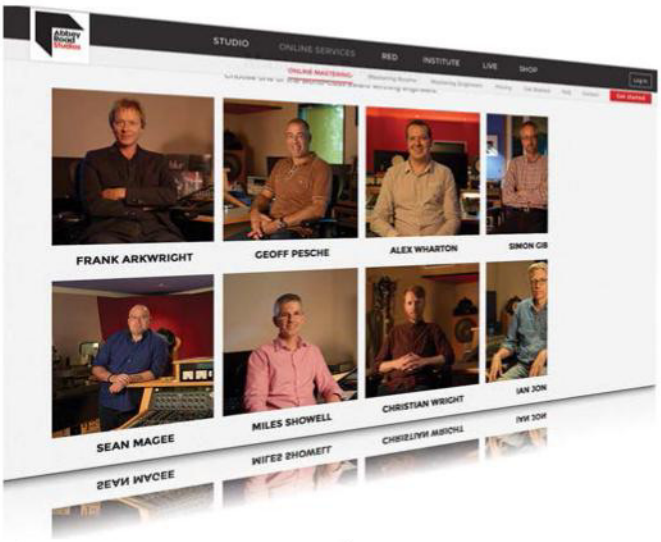
1. Trouble with your output file? You'll need to go back into the mix, so sort all of these problems out before mastering and consider the bigger picture

2. Research expert mastering services like Abbey Road and Metropolis, where a named engineer can master your music… for a fee. But do learn the process yourself, as well

3. Mastering software like IZotope's Ozone 7 can make the process easier, but don't use presets just to do the job quickly. Explore and expand…

02 Don't be afraid *not* to do it

And this leads nicely on to our next point, especially, if you have mixed your own music. You might be just like us and want to handle every single aspect of music production yourself (yes, making music does turn you into a control freak, we know!), but don't be frightened to let someone else master your mixes. There are many online companies that offer very reasonably priced mastering services for as little as £50 a track. Obviously, these do vary in terms of quality – although we've been impressed with a couple of very cheap ones that we've tested (see www.musictech.net for more), but you could consider a 'named' service like Metropolis or even Abbey Road where you can pretty much guarantee a good job if you have more budget (£110 per track for a named engineer). So why pay someone else to do it? Two main reasons: the first is that you are paying for experience. We know for a fact – because we've interviewed them all – that the mastering engineers at Metropolis, as just one example, have had their ears around just about every single piece of music you can imagine (and several you can't). So no matter



what you throw at them, they will have someone on their team who can handle it. Secondly, simply having another set of fresh ears listen to your music and can offer a new perspective. This should always be encouraged (at any step of the production process, actually). By the way, don't consider this point as us telling you *not* to give mastering a go. We would encourage you to learn the craft and to work with others doing it. Get to a point where you can master another person's mix and they master yours. Fresh perspectives all round!



4. The processes you need include compression, EQ and limiting. The smile EQ is a commonly used EQ shape

5. Limiting will help you boost your levels, but don't make mastering all about loud. That's a war you don't want to be in…

03 Don't follow a preset path

Now we can't prove this, but there's every chance that a less reputable mastering engineer could just charge you to whack your mix through a set of preset EQ, limiting and other process presets, depending on your music's style (or even not!). And you know what, this might just work! Indeed, many of the specialist mastering plug-ins we've reviewed have similar presets, where they'll 'suggest' a signal path with certain parameter values for certain styles of music. And they often *do* result in a better-sounding master. But while this can be a good way to go as a first stab at mastering, we think that the best way to master is to treat each track on its own merits – or lack of them – and not just according to a genre it is supposed to be from. On a broader level, you could probably get away with applying a very slight 'smile' EQ curve to any track to give it a bit of welly at the top and bottom and then add some width – this could also be a good start. Or you could listen well, decide what you think it needs and then do it. If





you're new to mastering, by all means use a mix of both methods; check what the presets are offering, listen to what they do and how they do it, and experiment. Can you make the presets sound better? Of course you can – just like preset sounds on a synth!

04 The actual processes are…

Here are some typical mastering processes to consider, all detailed more at www.musictech.net. **Compression** does what it says – you'll create a more even mix with louder quiet bits and fewer deafening louder parts. Think about that song which brings loads of instrumentation in after a quiet interval without you having to reach for the volume control – it's compression in action. It smooths out a mix, and **multiband compressors** allow you to smooth out certain parts of a mix. With **EQ**, it's more about a general look at the overall picture, where high end will add fizz and low end cuts will reduce some sub rumble. And the final general process is **limiting**, which brings us to…

05 Don't make it all about the volume

Volume! Or rather, limiting your volume. Don't let volume take over in any sense. By this, we mean →





6. You should reference your music against your own tracks, and those of other musicians

7. Additional mastering processes include M/S or Mid/Side processing and stereo widening. As ever, go easy on these…

8. And reference again. Did we mention referencing?

→ don't master to make your tunes louder, and while you're at it, don't listen to extreme points of view in the loudness wars, either. On a (perhaps too) simplistic level, mastering is about making your music carry more of an impact. This doesn't mean making it louder, although we do advocate brick-wall limiting at the end of the process to add a small amount of level and to control the peaks of a mix.

06 Reference, reference, reference

Have we said this in other steps of the production process? Yes, we have! But it's soooooooo important to reference your material with a track that you admire – one that has been mastered to a high standard. Load it into the software you are using to master, or your DAW, and play it side by side with your own material. As with mixing, this may be a little disappointing at first, but as you progress and progress, you will gradually step up level by level towards your reference goal. Similarly, make sure you reference other tracks that you've mastered that will be appearing on the same EP or album. You need all the tracks within a body of work to gel together, rather than having them sound as though they were recorded, mixed and mastered by different artists and engineers. It's bad enough playing tunes on Shuffle from your iPod and hearing the volume levels jump from decade to decade – you really don't want that happening on an album!

07 Go easy

Whereas compression, EQ and limiting are the mainstays of mastering, it's not all about these processes all of the time. There are plenty of other ones to get yourself involved in. **Mid/Side** processing is where the mix can be broken down into its mid and side areas, which typically allows you to process only those areas with, say, compression, EQ or reverb; **stereo widening** is just as it says, but is also one that should be considered as a process that is sorted in mixing – you can widen in mastering, and many people do, but be careful. Indeed, that is the message with all mastering processes: go easy. Each is such a subtle process that you might end up being unsubtle with certain aspects just so you can easily hear the results. But some would argue that if you can hear the results too obviously, you've overdone things (or have a very bad mix to start with!).

08 And finally… reference again

You've referenced the track you are mastering against other people's music; you've also referenced it against your own music that will be appearing on the same album, so now it's time to reference it against itself. Yes, at every stage, do reference what you are doing against the unmastered version you started with, to make sure that you're going in the right direction. If you want to get really anal about it, make and label versions at every step (with different compressors, etc), so that you're able to reference each step of the process and hear the influence it had on your original mix. Just don't get too bogged down in it… MTF





SUMMER SALE!

3 issues for $10
When you subscribe to MT

If you're reading *MusicTech* from the USA, we've got a special offer just for you! Try 3 issues for just $10 in our Summer Sale!



DIAL TOLL FREE
800 428 3003

QUOTE CODE SSDPS16

ORDER ONLINE AT
www.imsnews.com/musictech

QUOTE CODE SSDPS16

MusicTech magazine's subscriptions are now handled in North America by IMS News. Call at local rates, in your hours and pay in US/Canadian dollars.

MusicTech FOCUS DVD ROM

ON YOUR FREE DVD

The team behind Ableton is constantly tweaking and improving Live 9, with recent updates adding Link and new slicing, routing and playability features with Push. There's plenty to get stuck in to, so we've put together a wide selection of content to help guide you around Live and give you some inspiration to kick-start your music making. There's over two hours' worth of high quality videos, the latest software demos, freeware plug-ins and promotional videos showing off cutting-edge synths, effects and controllers. You'll also find plenty of royalty-free loops, samples and hits to use in your own productions...



MTF On the disc Over 2 Hours of pro video tuition



01 GROOVE 3

Groove 3 has provided six videos including getting started with Drum Racks, separating Rack sounds to different channels, working with multiple samples in Sampler, tuning samples and how to use warping techniques for sound design.
www.groove3.com



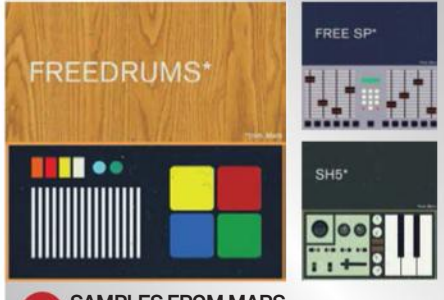
02 SUBBASS ACADEMY

SubBass Academy has provided three Live tutorials which look at using sidechain to create smoother transitions in a DJ mix, how to prep tracks and use warp markers for a mix compilation, plus an in-depth look at some of the latest features in Live.
www.subbassdj.com



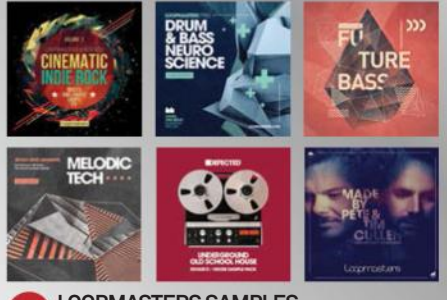
03 POINT BLANK MUSIC SCHOOL

A QuickTip on creating automatic track remixes using Follow Actions, an advanced guide to Sampler, a look at new filter types, envelopes and LFOs in Simpler, and Point Blank instructor Freddy Frogs showing how to multi-sample a keyboard.
www.pointblanklondon.com



04 SAMPLES FROM MARS

Here you'll find a whole bunch of crunchy drum hit samples and multisampled synth instruments from Samples From Mars presented in Ableton Live format, with added saturation from analogue tape and a range of outboard hardware.
www.samplesfrommars.com



05 LOOPMASTERS SAMPLES

A hand-picked library of royalty-free samples, including Cinematic Indie Rock V2, Drum & Bass Neuro Science, Future Bass, Melodic Tech, Underground Old School House, and Pete & Tim Cullen Grin Recordings. Use the code MUSICTECH10 for an exclusive discount.
www.loopmasters.com



06 MODEAUDIO

A pack of audio/MIDI loops and Massive presets taken from nine of ModeAudio's latest releases. There's neon-tinged synths, live percussion, future RnB, hazy hip hop grooves, soundscapes, synth pop and more. Use MUSTECH20 for an exclusive discount.
www.modeaudio.com

MTF Your Disc

MTF DVD 44 Ableton Live 2016/17



PROMOTIONAL VIDEOS

We've got over 750MB of videos showcasing the latest plug-ins and hardware including a huge range of cutting-edge synths and software instruments from Korg, Native Instruments, Novation, Studio Electronics, Teenage Engineering, Toontrack, Zero-G and Wave DNA. You'll also find high quality controllers, desks, headphones and interfaces from ROLI, Allen & Heath, Focusrite, and AKG, plus top of the range software effects and processors from FabFilter, iZotope, MeldaProduction, Plugin Boutique and Sonnox.

USING OUR WORKSHOPS

Whether you're a seasoned pro or totally new to Live, we've got a host of Live workshops to help you improve your programming and mixing skills. Topics include, using aggregate audio devices, creating custom Drum Racks, using Follow Actions, and clip management using Push. Where appropriate you'll find hi-res images, project files and audio for each of the workshops. *Please visit www.musictech.net to download the full list of accompanying workshop files and images.*



SOFTWARE DEMOS

Why not try out the latest plug-ins and software to supplement Live's native effects. From innovative synths, to complex, modern processors and tools, we've rounded up a range of demo and freeware software for you to try out. You'll find plenty of creative effects, filters, instruments, EQs, distortion and saturation effects, stereo tools, dynamics processors and more to help you compose and fine tune your tracks.

ROYALTY-FREE SAMPLES

We've rounded up a top-notch collection of royalty-free samples from Loopmasters, ModeAudio and Samples From Mars for you to use in your tracks. You'll find a varied array of blistering dnb beats, live percussion grooves, atmospheric ambient drones, chunky hip hop loops, twinkling arps, saturated house drums, and multisampled synth and drum instruments. All files come in 24-bit WAV, MIDI or Live formats and are ready to be imported, chopped and manipulated to your heart's content.



MusicTech FOCUS DVD ROM

YOUR DVD CONTENT FILES

ZIP FILES

To maximise the amount of content we can bring you on each DVD, the video, tutorial and samples files are supplied compressed ('zipped'). Mac users should be able to decompress ZIP files simply by double-clicking on them; PC users may need to download a utility such as WinZip (www.winzip.com).

TUTORIAL FILES

The software tutorials that feature in each issue of MTF are almost always accompanied by files and audio so you can work through them on your system. These files are zipped to reduce the space they occupy on the DVD. Download them to your hard drive and unzip them to access the individual files (remembering to eject the DVD to prevent your computer from slowing down).

WHAT IS ROYALTY-FREE?

Any MTF DVD content marked 'royalty-free' can be used in your own original compositions (even commercial ones). You may not, however, resell these samples in any other form.

DEFECTIVE DISCS

In the unlikely event that your disc is defective, please return it to: Disc Returns, Anthem Publishing, Suite 6, Piccadilly House, London, Bath BA1 6PL. We will endeavour to supply you with a replacement disc immediately. Please note that we're unable to provide technical support for the software on the MTF DVD – please check our website at www.musictech.net for any known problems.

MISSING YOUR DISC?

If your disc is missing, contact us at editorial@anthem-publishing.com with your full postal address and the issue number.

ANTHEM PUBLISHING

www.scan.co.uk

T: 01204 47 47 47

SCAN COMPUTERS



SCAN PRO AUDIO SHOW

Features the latest news and product demos, plus exclusive content & tutorials from some of the world's best producers, artists and engineers.
New show every fortnight, 1080p resolution, check them out now on the Pro Audio Homepage.

Pilot : Korg Minilogue / TMS Productions : UA Apollo Console guide
001 : Steinberg Cubase "Big Club Sound In The Box" Masterclass / Marshall Jefferson
002 : PreSonus & Studio One Expert "Migrating To Studio One" / Novation Circuit Components
003 : Arturia MatrixBrute / Vocal Production of a UK #1 Single.
004 : "Zoom ARQ"

Why not compliment your silent system with the world's first on-ear planar magnetic headphone?
The compact Audeze SINE are closed back and are perfect for both tracking and monitoring, as generous power handling, coupled with an extremely detailed transient response make these as useful for mobile production as they are for studio recording.

Supplier of performance hardware and pc audio workstations. Intel Inside®. Extraordinary Performance Outside.

intel CORE™ i7 inside™

3XS FZ170 Silent Production PC With 6th Generation Intel® Core™ Processors

Our FZ170 DAW is your silent partner in the studio built around the latest Intel Skylake CPU's complete with both Intel i5 6600 or i7 6700 options, we've taken our popular production solution and simply made it as quiet as possible. Designed around a compact case that offers a superb natural convection path, along with a passive cooler and PSU selection to take advantage of it. To finish this combination, we recommend fitting the system out with your choice from the latest M.2 & SSD storage options, to give you the very best performance for your projects, and ensure no interruptions to your recording studio environment.

- Intel® Core™ i5 Processor or
- Intel® Core™ i7 Processor
- Passive NOFAN Cooler
- Passive Seasonic PSU
- Up to 64GB Corsair LPX 2133MHz RAM
- Firewire available
- Windows 7, 8.1 & 10 Options

Starting from **£862** inc VAT

Don't buy hardware without Scansure protection. Insure yourself with 28 day installation damage www.scan.co.uk/scansure

Scan Finance. Buy now, pay 10% deposit* & spread the cost over 24, 36 or 48 months when you spend £250(Inc VAT) or more.

Classic credit options (24, 36 or 48 months). Customers will initially only have to pay a 10% deposit, with the orginal loan amount payable over 24, 36 or 48 months. *Example: £1,000.00 Deposit: £100.00 Amount of loan: £900.00 Total charge for credit: £220.68 Total payable: £1220.68 Total 36 Monthly instalments of: £31.13 APR:15.9%. Terms and Conditions Apply.

Ultrabook, Celeron, Celeron Inside, Core Inside, Intel, Intel Logo, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Inside Logo, Intel vPro, Itanium, Itanium Inside, Pentium, Pentium Inside, vPro Inside, Xeon, Xeon Phi, and Xeon Inside are trademarks of Intel Corporation in the U.S. and/or other countries.

All prices include VAT and are correct at time of print, prices are subject to change.

TEXTS IN COMPUTER SCIENCE

Computer Vision

Algorithms and Applications



Richard Szeliski

Springer

Texts in Computer Science

Editors
David Gries
Fred B. Schneider

For further volumes:
www.springer.com/series/3191

Richard Szeliski

Computer Vision

Algorithms and Applications

Springer

Dr. Richard Szeliski
Microsoft Research
One Microsoft Way
98052-6399 Redmond
Washington
USA
szeliski@microsoft.com

Series Editors
David Gries
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501, USA

Fred B. Schneider
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501, USA

ISSN 1868-0941 e-ISSN 1868-095X
ISBN 978-1-84882-934-3 e-ISBN 978-1-84882-935-0
DOI 10.1007/978-1-84882-935-0
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

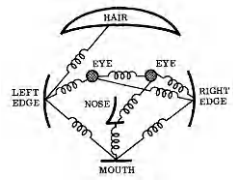
Library of Congress Control Number: 2010936817

© Springer-Verlag London Limited 2011
Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.
The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.
The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed on acid-free paper

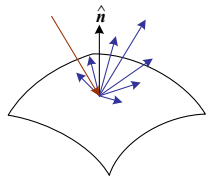
Springer is part of Springer Science+Business Media (www.springer.com)

This book is dedicated to my parents,
Zdzisław and Jadwiga,
and my family,
Lyn, Anne, and Stephen.



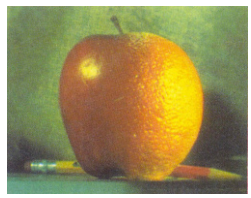
1 Introduction **1**

What is computer vision? • A brief history •
Book overview • Sample syllabus • Notation



2 Image formation **27**

Geometric primitives and transformations •
Photometric image formation •
The digital camera



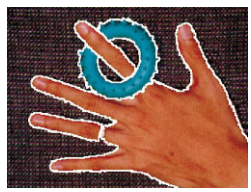
3 Image processing **87**

Point operators • Linear filtering •
More neighborhood operators • Fourier transforms •
Pyramids and wavelets • Geometric transformations •
Global optimization



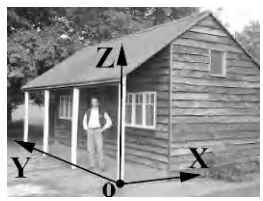
4 Feature detection and matching **181**

Points and patches •
Edges • Lines



5 Segmentation **235**

Active contours • Split and merge •
Mean shift and mode finding • Normalized cuts •
Graph cuts and energy-based methods



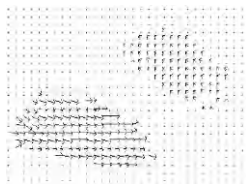
6 Feature-based alignment **273**

2D and 3D feature-based alignment •
Pose estimation •
Geometric intrinsic calibration



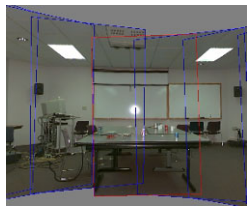
7 Structure from motion **303**

Triangulation • Two-frame structure from motion •
Factorization • Bundle adjustment •
Constrained structure and motion



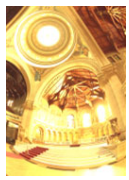
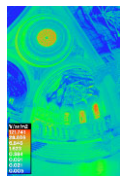
8 Dense motion estimation **335**

Translational alignment • Parametric motion • Spline-based motion • Optical flow • Layered motion



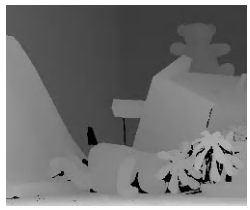
9 Image stitching **375**

Motion models • Global alignment • Compositing

10 Computational photography **409**

Photometric calibration • High dynamic range imaging • Super-resolution and blur removal • Image matting and compositing • Texture analysis and synthesis



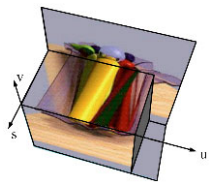
11 Stereo correspondence **467**

Epipolar geometry • Sparse correspondence • Dense correspondence • Local methods • Global optimization • Multi-view stereo



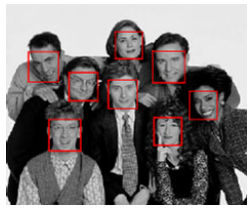
12 3D reconstruction **505**

Shape from X • Active rangefinding • Surface representations • Point-based representations • Volumetric representations • Model-based reconstruction • Recovering texture maps and albedos



13 Image-based rendering **543**

View interpolation • Layered depth images • Light fields and Lumigraphs • Environment mattes • Video-based rendering



14 Recognition **575**

Object detection • Face recognition • Instance recognition • Category recognition • Context and scene understanding • Recognition databases and test sets

Preface

The seeds for this book were first planted in 2001 when Steve Seitz at the University of Washington invited me to co-teach a course called "Computer Vision for Computer Graphics". At that time, computer vision techniques were increasingly being used in computer graphics to create image-based models of real-world objects, to create visual effects, and to merge real-world imagery using computational photography techniques. Our decision to focus on the applications of computer vision to fun problems such as image stitching and photo-based 3D modeling from personal photos seemed to resonate well with our students.

Since that time, a similar syllabus and project-oriented course structure has been used to teach general computer vision courses both at the University of Washington and at Stanford. (The latter was a course I co-taught with David Fleet in 2003.) Similar curricula have been adopted at a number of other universities and also incorporated into more specialized courses on computational photography. (For ideas on how to use this book in your own course, please see Table 1.1 in Section 1.4.)

This book also reflects my 20 years' experience doing computer vision research in corporate research labs, mostly at Digital Equipment Corporation's Cambridge Research Lab and at Microsoft Research. In pursuing my work, I have mostly focused on problems and solution techniques (algorithms) that have practical real-world applications and that work well in practice. Thus, this book has more emphasis on basic techniques that work under real-world conditions and less on more esoteric mathematics that has intrinsic elegance but less practical applicability.

This book is suitable for teaching a senior-level undergraduate course in computer vision to students in both computer science and electrical engineering. I prefer students to have either an image processing or a computer graphics course as a prerequisite so that they can spend less time learning general background mathematics and more time studying computer vision techniques. The book is also suitable for teaching graduate-level courses in computer vision (by delving into the more demanding application and algorithmic areas) and as a general reference to fundamental techniques and the recent research literature. To this end, I have attempted wherever possible to at least cite the newest research in each sub-field, even if the technical details are too complex to cover in the book itself.

In teaching our courses, we have found it useful for the students to attempt a number of small implementation projects, which often build on one another, in order to get them used to working with real-world images and the challenges that these present. The students are then asked to choose an individual topic for each of their small-group, final projects. (Sometimes these projects even turn into conference papers!) The exercises at the end of each chapter contain numerous suggestions for smaller mid-term projects, as well as more open-ended

problems whose solutions are still active research topics. Wherever possible, I encourage students to try their algorithms on their own personal photographs, since this better motivates them, often leads to creative variants on the problems, and better acquaints them with the variety and complexity of real-world imagery.

In formulating and solving computer vision problems, I have often found it useful to draw inspiration from three high-level approaches:

- **Scientific:** build detailed models of the image formation process and develop mathematical techniques to invert these in order to recover the quantities of interest (where necessary, making simplifying assumption to make the mathematics more tractable).
- **Statistical:** use probabilistic models to quantify the prior likelihood of your unknowns and the noisy measurement processes that produce the input images, then infer the best possible estimates of your desired quantities and analyze their resulting uncertainties. The inference algorithms used are often closely related to the optimization techniques used to invert the (scientific) image formation processes.
- **Engineering:** develop techniques that are simple to describe and implement but that are also known to work well in practice. Test these techniques to understand their limitation and failure modes, as well as their expected computational costs (run-time performance).

These three approaches build on each other and are used throughout the book.

My personal research and development philosophy (and hence the exercises in the book) have a strong emphasis on *testing* algorithms. It's too easy in computer vision to develop an algorithm that does something *plausible* on a few images rather than something *correct*. The best way to validate your algorithms is to use a three-part strategy.

First, test your algorithm on clean synthetic data, for which the exact results are known. Second, add noise to the data and evaluate how the performance degrades as a function of noise level. Finally, test the algorithm on real-world data, preferably drawn from a wide variety of sources, such as photos found on the Web. Only then can you truly know if your algorithm can deal with real-world complexity, i.e., images that do not fit some simplified model or assumptions.

In order to help students in this process, this books comes with a large amount of supplementary material, which can be found on the book's Web site http://szeliski.org/Book. This material, which is described in Appendix C, includes:

- pointers to commonly used data sets for the problems, which can be found on the Web
- pointers to software libraries, which can help students get started with basic tasks such as reading/writing images or creating and manipulating images
- slide sets corresponding to the material covered in this book
- a BibTeX bibliography of the papers cited in this book.

The latter two resources may be of more interest to instructors and researchers publishing new papers in this field, but they will probably come in handy even with regular students. Some of the software libraries contain implementations of a wide variety of computer vision algorithms, which can enable you to tackle more ambitious projects (with your instructor's consent).

Acknowledgements

I would like to gratefully acknowledge all of the people whose passion for research and inquiry as well as encouragement have helped me write this book.

Steve Zucker at McGill University first introduced me to computer vision, taught all of his students to question and debate research results and techniques, and encouraged me to pursue a graduate career in this area.

Takeo Kanade and Geoff Hinton, my Ph. D. thesis advisors at Carnegie Mellon University, taught me the fundamentals of good research, writing, and presentation. They fired up my interest in visual processing, 3D modeling, and statistical methods, while Larry Matthies introduced me to Kalman filtering and stereo matching.

Demetri Terzopoulos was my mentor at my first industrial research job and taught me the ropes of successful publishing. Yvan Leclerc and Pascal Fua, colleagues from my brief interlude at SRI International, gave me new perspectives on alternative approaches to computer vision.

During my six years of research at Digital Equipment Corporation's Cambridge Research Lab, I was fortunate to work with a great set of colleagues, including Ingrid Carlbom, Gudrun Klinker, Keith Waters, Richard Weiss, Stéphane Lavallée, and Sing Bing Kang, as well as to supervise the first of a long string of outstanding summer interns, including David Tonnesen, Sing Bing Kang, James Coughlan, and Harry Shum. This is also where I began my long-term collaboration with Daniel Scharstein, now at Middlebury College.

At Microsoft Research, I've had the outstanding fortune to work with some of the world's best researchers in computer vision and computer graphics, including Michael Cohen, Hugues Hoppe, Stephen Gortler, Steve Shafer, Matthew Turk, Harry Shum, Anandan, Phil Torr, Antonio Criminisi, Georg Petschnigg, Kentaro Toyama, Ramin Zabih, Shai Avidan, Sing Bing Kang, Matt Uyttendaele, Patrice Simard, Larry Zitnick, Richard Hartley, Simon Winder, Drew Steedly, Chris Pal, Nebojsa Jojic, Patrick Baudisch, Dani Lischinski, Matthew Brown, Simon Baker, Michael Goesele, Eric Stollnitz, David Nistér, Blaise Aguera y Arcas, Sudipta Sinha, Johannes Kopf, Neel Joshi, and Krishnan Ramnath. I was also lucky to have as interns such great students as Polina Golland, Simon Baker, Mei Han, Arno Schödl, Ron Dror, Ashley Eden, Jinxiang Chai, Rahul Swaminathan, Yanghai Tsin, Sam Hasinoff, Anat Levin, Matthew Brown, Eric Bennett, Vaibhav Vaish, Jan-Michael Frahm, James Diebel, Ce Liu, Josef Sivic, Grant Schindler, Colin Zheng, Neel Joshi, Sudipta Sinha, Zeev Farbman, Rahul Garg, Tim Cho, Yekeun Jeong, Richard Roberts, Varsha Hedau, and Dilip Krishnan.

While working at Microsoft, I've also had the opportunity to collaborate with wonderful colleagues at the University of Washington, where I hold an Affiliate Professor appointment. I'm indebted to Tony DeRose and David Salesin, who first encouraged me to get involved with the research going on at UW, my long-time collaborators Brian Curless, Steve Seitz, Maneesh Agrawala, Sameer Agarwal, and Yasu Furukawa, as well as the students I have had the privilege to supervise and interact with, including Fréderic Pighin, Yung-Yu Chuang, Doug Zongker, Colin Zheng, Aseem Agarwala, Dan Goldman, Noah Snavely, Rahul Garg, and Ryan Kaminsky. As I mentioned at the beginning of this preface, this book owes its inception to the vision course that Steve Seitz invited me to co-teach, as well as to Steve's encouragement, course notes, and editorial input.

I'm also grateful to the many other computer vision researchers who have given me so many constructive suggestions about the book, including Sing Bing Kang, who was my infor-

mal book editor, Vladimir Kolmogorov, who contributed Appendix B.5.5 on linear programming techniques for MRF inference, Daniel Scharstein, Richard Hartley, Simon Baker, Noah Snavely, Bill Freeman, Svetlana Lazebnik, Matthew Turk, Jitendra Malik, Alyosha Efros, Michael Black, Brian Curless, Sameer Agarwal, Li Zhang, Deva Ramanan, Olga Veksler, Yuri Boykov, Carsten Rother, Phil Torr, Bill Triggs, Bruce Maxwell, Jana Košecká, Eero Simoncelli, Aaron Hertzmann, Antonio Torralba, Tomaso Poggio, Theo Pavlidis, Baba Vemuri, Nando de Freitas, Chuck Dyer, Song Yi, Falk Schubert, Roman Pflugfelder, Marshall Tappen, James Coughlan, Sammy Rogmans, Klaus Strobel, Shanmuganathan, Andreas Siebert, Yongjun Wu, Fred Pighin, Juan Cockburn, Ronald Mallet, Tim Soper, Georgios Evangelidis, Dwight Fowler, Itzik Bayaz, Daniel O'Connor, and Srikrishna Bhat. Shena Deuchers did a fantastic job copy-editing the book and suggesting many useful improvements and Wayne Wheeler and Simon Rees at Springer were most helpful throughout the whole book publishing process. Keith Price's Annotated Computer Vision Bibliography was invaluable in tracking down references and finding related work.

If you have any suggestions for improving the book, please send me an e-mail, as I would like to keep the book as accurate, informative, and timely as possible.

Lastly, this book would not have been possible or worthwhile without the incredible support and encouragement of my family. I dedicate this book to my parents, Zdzisław and Jadwiga, whose love, generosity, and accomplishments have always inspired me; to my sister Basia for her lifelong friendship; and especially to Lyn, Anne, and Stephen, whose daily encouragement in all matters (including this book project) makes it all worthwhile.

Lake Wenatchee
August, 2010

Contents

Preface **vii**

1 Introduction **1**
1.1 What is computer vision? . . . 3
1.2 A brief history . . . 10
1.3 Book overview . . . 17
1.4 Sample syllabus . . . 23
1.5 A note on notation . . . 25
1.6 Additional reading . . . 25

2 Image formation **27**
2.1 Geometric primitives and transformations . . . 29
2.1.1 Geometric primitives . . . 29
2.1.2 2D transformations . . . 33
2.1.3 3D transformations . . . 36
2.1.4 3D rotations . . . 37
2.1.5 3D to 2D projections . . . 42
2.1.6 Lens distortions . . . 52
2.2 Photometric image formation . . . 54
2.2.1 Lighting . . . 54
2.2.2 Reflectance and shading . . . 55
2.2.3 Optics . . . 61
2.3 The digital camera . . . 65
2.3.1 Sampling and aliasing . . . 69
2.3.2 Color . . . 71
2.3.3 Compression . . . 80
2.4 Additional reading . . . 82
2.5 Exercises . . . 82

3 Image processing **87**
3.1 Point operators . . . 89
3.1.1 Pixel transforms . . . 91
3.1.2 Color transforms . . . 92
3.1.3 Compositing and matting . . . 92
3.1.4 Histogram equalization . . . 94

3.1.5 *Application*: Tonal adjustment . . . 97
3.2 Linear filtering . . . 98
3.2.1 Separable filtering . . . 102
3.2.2 Examples of linear filtering . . . 103
3.2.3 Band-pass and steerable filters . . . 104
3.3 More neighborhood operators . . . 108
3.3.1 Non-linear filtering . . . 108
3.3.2 Morphology . . . 112
3.3.3 Distance transforms . . . 113
3.3.4 Connected components . . . 115
3.4 Fourier transforms . . . 116
3.4.1 Fourier transform pairs . . . 119
3.4.2 Two-dimensional Fourier transforms . . . 123
3.4.3 Wiener filtering . . . 123
3.4.4 *Application*: Sharpening, blur, and noise removal . . . 126
3.5 Pyramids and wavelets . . . 127
3.5.1 Interpolation . . . 127
3.5.2 Decimation . . . 130
3.5.3 Multi-resolution representations . . . 132
3.5.4 Wavelets . . . 136
3.5.5 *Application*: Image blending . . . 140
3.6 Geometric transformations . . . 143
3.6.1 Parametric transformations . . . 145
3.6.2 Mesh-based warping . . . 149
3.6.3 *Application*: Feature-based morphing . . . 152
3.7 Global optimization . . . 153
3.7.1 Regularization . . . 154
3.7.2 Markov random fields . . . 158
3.7.3 *Application*: Image restoration . . . 169
3.8 Additional reading . . . 169
3.9 Exercises . . . 171

4 Feature detection and matching **181**
4.1 Points and patches . . . 183
4.1.1 Feature detectors . . . 185
4.1.2 Feature descriptors . . . 196
4.1.3 Feature matching . . . 200
4.1.4 Feature tracking . . . 207
4.1.5 *Application*: Performance-driven animation . . . 209
4.2 Edges . . . 210
4.2.1 Edge detection . . . 210
4.2.2 Edge linking . . . 215
4.2.3 *Application*: Edge editing and enhancement . . . 219
4.3 Lines . . . 220
4.3.1 Successive approximation . . . 220
4.3.2 Hough transforms . . . 221

4.3.3 Vanishing points . . . 224
4.3.4 *Application*: Rectangle detection . . . 226
4.4 Additional reading . . . 227
4.5 Exercises . . . 228

5 Segmentation 235
5.1 Active contours . . . 237
5.1.1 Snakes . . . 238
5.1.2 Dynamic snakes and CONDENSATION . . . 243
5.1.3 Scissors . . . 246
5.1.4 Level Sets . . . 248
5.1.5 *Application*: Contour tracking and rotoscoping . . . 249
5.2 Split and merge . . . 250
5.2.1 Watershed . . . 251
5.2.2 Region splitting (divisive clustering) . . . 251
5.2.3 Region merging (agglomerative clustering) . . . 251
5.2.4 Graph-based segmentation . . . 252
5.2.5 Probabilistic aggregation . . . 253
5.3 Mean shift and mode finding . . . 254
5.3.1 K-means and mixtures of Gaussians . . . 256
5.3.2 Mean shift . . . 257
5.4 Normalized cuts . . . 260
5.5 Graph cuts and energy-based methods . . . 264
5.5.1 *Application*: Medical image segmentation . . . 268
5.6 Additional reading . . . 268
5.7 Exercises . . . 270

6 Feature-based alignment 273
6.1 2D and 3D feature-based alignment . . . 275
6.1.1 2D alignment using least squares . . . 275
6.1.2 *Application*: Panography . . . 277
6.1.3 Iterative algorithms . . . 278
6.1.4 Robust least squares and RANSAC . . . 281
6.1.5 3D alignment . . . 283
6.2 Pose estimation . . . 284
6.2.1 Linear algorithms . . . 284
6.2.2 Iterative algorithms . . . 286
6.2.3 *Application*: Augmented reality . . . 287
6.3 Geometric intrinsic calibration . . . 288
6.3.1 Calibration patterns . . . 289
6.3.2 Vanishing points . . . 290
6.3.3 *Application*: Single view metrology . . . 292
6.3.4 Rotational motion . . . 293
6.3.5 Radial distortion . . . 295
6.4 Additional reading . . . 296
6.5 Exercises . . . 296

7 Structure from motion **303**
7.1 Triangulation . . . 305
7.2 Two-frame structure from motion . . . 307
7.2.1 Projective (uncalibrated) reconstruction . . . 312
7.2.2 Self-calibration . . . 313
7.2.3 *Application*: View morphing . . . 315
7.3 Factorization . . . 315
7.3.1 Perspective and projective factorization . . . 318
7.3.2 *Application*: Sparse 3D model extraction . . . 319
7.4 Bundle adjustment . . . 320
7.4.1 Exploiting sparsity . . . 322
7.4.2 *Application*: Match move and augmented reality . . . 324
7.4.3 Uncertainty and ambiguities . . . 326
7.4.4 *Application*: Reconstruction from Internet photos . . . 327
7.5 Constrained structure and motion . . . 329
7.5.1 Line-based techniques . . . 330
7.5.2 Plane-based techniques . . . 331
7.6 Additional reading . . . 332
7.7 Exercises . . . 332

8 Dense motion estimation **335**
8.1 Translational alignment . . . 337
8.1.1 Hierarchical motion estimation . . . 341
8.1.2 Fourier-based alignment . . . 341
8.1.3 Incremental refinement . . . 345
8.2 Parametric motion . . . 350
8.2.1 *Application*: Video stabilization . . . 354
8.2.2 Learned motion models . . . 354
8.3 Spline-based motion . . . 355
8.3.1 *Application*: Medical image registration . . . 358
8.4 Optical flow . . . 360
8.4.1 Multi-frame motion estimation . . . 363
8.4.2 *Application*: Video denoising . . . 364
8.4.3 *Application*: De-interlacing . . . 364
8.5 Layered motion . . . 365
8.5.1 *Application*: Frame interpolation . . . 368
8.5.2 Transparent layers and reflections . . . 368
8.6 Additional reading . . . 370
8.7 Exercises . . . 371

9 Image stitching **375**
9.1 Motion models . . . 378
9.1.1 Planar perspective motion . . . 379
9.1.2 *Application*: Whiteboard and document scanning . . . 379
9.1.3 Rotational panoramas . . . 380
9.1.4 Gap closing . . . 382

9.1.5 *Application*: Video summarization and compression . . . 383
9.1.6 Cylindrical and spherical coordinates . . . 385
9.2 Global alignment . . . 387
9.2.1 Bundle adjustment . . . 388
9.2.2 Parallax removal . . . 391
9.2.3 Recognizing panoramas . . . 392
9.2.4 Direct vs. feature-based alignment . . . 393
9.3 Compositing . . . 396
9.3.1 Choosing a compositing surface . . . 396
9.3.2 Pixel selection and weighting (de-ghosting) . . . 398
9.3.3 *Application*: Photomontage . . . 403
9.3.4 Blending . . . 403
9.4 Additional reading . . . 406
9.5 Exercises . . . 407

10 Computational photography 409
10.1 Photometric calibration . . . 412
10.1.1 Radiometric response function . . . 412
10.1.2 Noise level estimation . . . 415
10.1.3 Vignetting . . . 416
10.1.4 Optical blur (spatial response) estimation . . . 416
10.2 High dynamic range imaging . . . 419
10.2.1 Tone mapping . . . 427
10.2.2 *Application*: Flash photography . . . 434
10.3 Super-resolution and blur removal . . . 436
10.3.1 Color image demosaicing . . . 440
10.3.2 *Application*: Colorization . . . 442
10.4 Image matting and compositing . . . 443
10.4.1 Blue screen matting . . . 445
10.4.2 Natural image matting . . . 446
10.4.3 Optimization-based matting . . . 450
10.4.4 Smoke, shadow, and flash matting . . . 452
10.4.5 Video matting . . . 454
10.5 Texture analysis and synthesis . . . 455
10.5.1 *Application*: Hole filling and inpainting . . . 457
10.5.2 *Application*: Non-photorealistic rendering . . . 458
10.6 Additional reading . . . 460
10.7 Exercises . . . 461

11 Stereo correspondence 467
11.1 Epipolar geometry . . . 471
11.1.1 Rectification . . . 472
11.1.2 Plane sweep . . . 474
11.2 Sparse correspondence . . . 475
11.2.1 3D curves and profiles . . . 476
11.3 Dense correspondence . . . 477

11.3.1 Similarity measures . . . 479
11.4 Local methods . . . 480
11.4.1 Sub-pixel estimation and uncertainty . . . 482
11.4.2 *Application*: Stereo-based head tracking . . . 483
11.5 Global optimization . . . 484
11.5.1 Dynamic programming . . . 485
11.5.2 Segmentation-based techniques . . . 487
11.5.3 *Application*: Z-keying and background replacement . . . 489
11.6 Multi-view stereo . . . 489
11.6.1 Volumetric and 3D surface reconstruction . . . 492
11.6.2 Shape from silhouettes . . . 497
11.7 Additional reading . . . 499
11.8 Exercises . . . 500

12 3D reconstruction 505
12.1 Shape from X . . . 508
12.1.1 Shape from shading and photometric stereo . . . 508
12.1.2 Shape from texture . . . 510
12.1.3 Shape from focus . . . 511
12.2 Active rangefinding . . . 512
12.2.1 Range data merging . . . 515
12.2.2 *Application*: Digital heritage . . . 517
12.3 Surface representations . . . 518
12.3.1 Surface interpolation . . . 518
12.3.2 Surface simplification . . . 520
12.3.3 Geometry images . . . 520
12.4 Point-based representations . . . 521
12.5 Volumetric representations . . . 522
12.5.1 Implicit surfaces and level sets . . . 522
12.6 Model-based reconstruction . . . 523
12.6.1 Architecture . . . 524
12.6.2 Heads and faces . . . 526
12.6.3 *Application*: Facial animation . . . 528
12.6.4 Whole body modeling and tracking . . . 530
12.7 Recovering texture maps and albedos . . . 534
12.7.1 Estimating BRDFs . . . 536
12.7.2 *Application*: 3D photography . . . 537
12.8 Additional reading . . . 538
12.9 Exercises . . . 539

13 Image-based rendering 543
13.1 View interpolation . . . 545
13.1.1 View-dependent texture maps . . . 547
13.1.2 *Application*: Photo Tourism . . . 548
13.2 Layered depth images . . . 549
13.2.1 Impostors, sprites, and layers . . . 549

13.3 Light fields and Lumigraphs . . . 551
13.3.1 Unstructured Lumigraph . . . 554
13.3.2 Surface light fields . . . 555
13.3.3 *Application*: Concentric mosaics . . . 556
13.4 Environment mattes . . . 556
13.4.1 Higher-dimensional light fields . . . 558
13.4.2 The modeling to rendering continuum . . . 559
13.5 Video-based rendering . . . 560
13.5.1 Video-based animation . . . 560
13.5.2 Video textures . . . 561
13.5.3 *Application*: Animating pictures . . . 564
13.5.4 3D Video . . . 564
13.5.5 *Application*: Video-based walkthroughs . . . 566
13.6 Additional reading . . . 569
13.7 Exercises . . . 570

14 Recognition **575**
14.1 Object detection . . . 578
14.1.1 Face detection . . . 578
14.1.2 Pedestrian detection . . . 585
14.2 Face recognition . . . 588
14.2.1 Eigenfaces . . . 589
14.2.2 Active appearance and 3D shape models . . . 596
14.2.3 *Application*: Personal photo collections . . . 601
14.3 Instance recognition . . . 602
14.3.1 Geometric alignment . . . 603
14.3.2 Large databases . . . 604
14.3.3 *Application*: Location recognition . . . 609
14.4 Category recognition . . . 611
14.4.1 Bag of words . . . 612
14.4.2 Part-based models . . . 615
14.4.3 Recognition with segmentation . . . 620
14.4.4 *Application*: Intelligent photo editing . . . 621
14.5 Context and scene understanding . . . 625
14.5.1 Learning and large image collections . . . 627
14.5.2 *Application*: Image search . . . 630
14.6 Recognition databases and test sets . . . 631
14.7 Additional reading . . . 631
14.8 Exercises . . . 637

15 Conclusion **641**

A Linear algebra and numerical techniques **645**
A.1 Matrix decompositions . . . 646
A.1.1 Singular value decomposition . . . 646
A.1.2 Eigenvalue decomposition . . . 647

A.1.3 QR factorization . . . 649
A.1.4 Cholesky factorization . . . 650
A.2 Linear least squares . . . 651
A.2.1 Total least squares . . . 653
A.3 Non-linear least squares . . . 654
A.4 Direct sparse matrix techniques . . . 655
A.4.1 Variable reordering . . . 656
A.5 Iterative techniques . . . 656
A.5.1 Conjugate gradient . . . 657
A.5.2 Preconditioning . . . 659
A.5.3 Multigrid . . . 660

B Bayesian modeling and inference 661
B.1 Estimation theory . . . 662
B.1.1 Likelihood for multivariate Gaussian noise . . . 663
B.2 Maximum likelihood estimation and least squares . . . 665
B.3 Robust statistics . . . 666
B.4 Prior models and Bayesian inference . . . 667
B.5 Markov random fields . . . 668
B.5.1 Gradient descent and simulated annealing . . . 670
B.5.2 Dynamic programming . . . 670
B.5.3 Belief propagation . . . 672
B.5.4 Graph cuts . . . 674
B.5.5 Linear programming . . . 676
B.6 Uncertainty estimation (error analysis) . . . 678

C Supplementary material 679
C.1 Data sets . . . 680
C.2 Software . . . 682
C.3 Slides and lectures . . . 689
C.4 Bibliography . . . 690

References 691

Index 793

Chapter 1

Introduction

1.1 What is computer vision? 3
1.2 A brief history 10
1.3 Book overview 17
1.4 Sample syllabus 23
1.5 A note on notation 25
1.6 Additional reading 25



Figure 1.1 The human visual system has no problem interpreting the subtle variations in translucency and shading in this photograph and correctly segmenting the object from its background.

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_1, © Springer-Verlag London Limited 2011

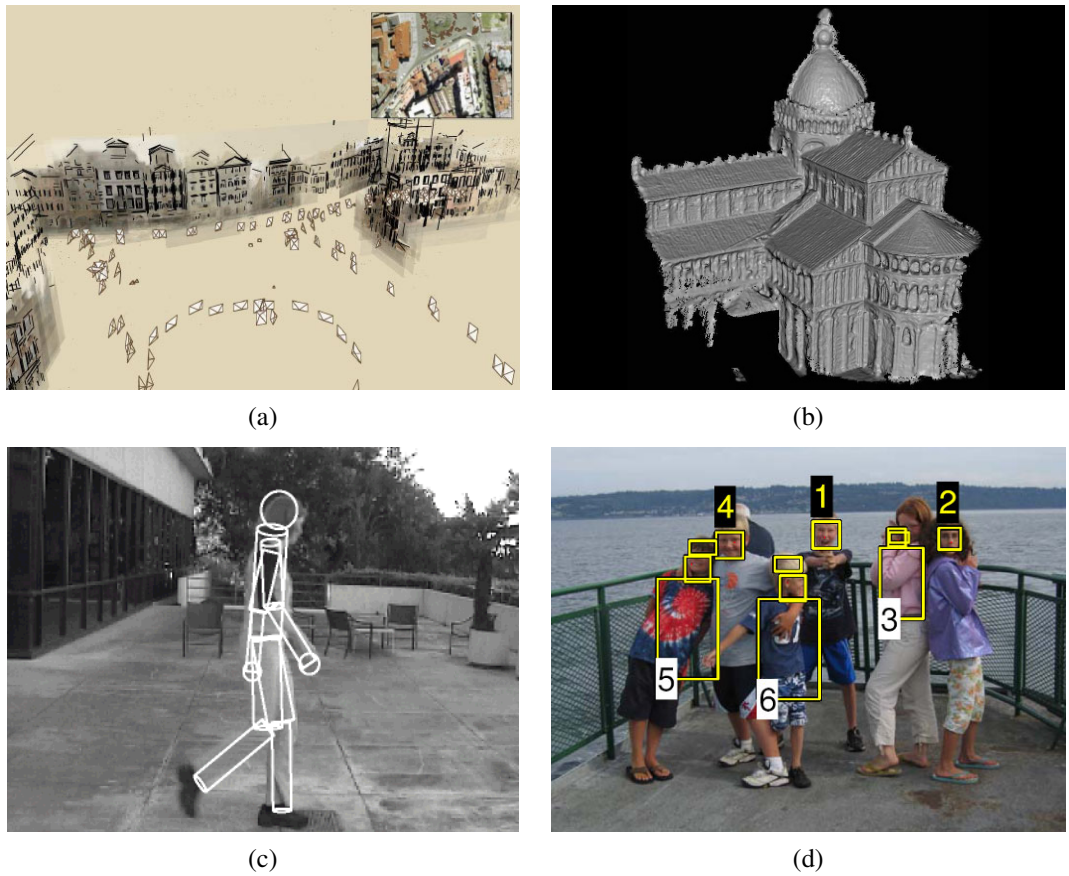


Figure 1.2 Some examples of computer vision algorithms and applications. (a) *Structure from motion* algorithms can reconstruct a sparse 3D point model of a large complex scene from hundreds of partially overlapping photographs (Snavely, Seitz, and Szeliski 2006) © 2006 ACM. (b) *Stereo matching* algorithms can build a detailed 3D model of a building façade from hundreds of differently exposed photographs taken from the Internet (Goesele, Snavely, Curless *et al.* 2007) © 2007 IEEE. (c) *Person tracking* algorithms can track a person walking in front of a cluttered background (Sidenbladh, Black, and Fleet 2000) © 2000 Springer. (d) *Face detection* algorithms, coupled with color-based clothing and hair detection algorithms, can locate and recognize the individuals in this image (Sivic, Zitnick, and Szeliski 2006) © 2006 Springer.

1.1 What is computer vision?

As humans, we perceive the three-dimensional structure of the world around us with apparent ease. Think of how vivid the three-dimensional percept is when you look at a vase of flowers sitting on the table next to you. You can tell the shape and translucency of each petal through the subtle patterns of light and shading that play across its surface and effortlessly segment each flower from the background of the scene (Figure 1.1). Looking at a framed group portrait, you can easily count (and name) all of the people in the picture and even guess at their emotions from their facial appearance. Perceptual psychologists have spent decades trying to understand how the visual system works and, even though they can devise optical illusions[1] to tease apart some of its principles (Figure 1.3), a complete solution to this puzzle remains elusive (Marr 1982; Palmer 1999; Livingstone 2008).

Researchers in computer vision have been developing, in parallel, mathematical techniques for recovering the three-dimensional shape and appearance of objects in imagery. We now have reliable techniques for accurately computing a partial 3D model of an environment from thousands of partially overlapping photographs (Figure 1.2a). Given a large enough set of views of a particular object or façade, we can create accurate dense 3D surface models using stereo matching (Figure 1.2b). We can track a person moving against a complex background (Figure 1.2c). We can even, with moderate success, attempt to find and name all of the people in a photograph using a combination of face, clothing, and hair detection and recognition (Figure 1.2d). However, despite all of these advances, the dream of having a computer interpret an image at the same level as a two-year old (for example, counting all of the animals in a picture) remains elusive. Why is vision so difficult? In part, it is because vision is an *inverse problem*, in which we seek to recover some unknowns given insufficient information to fully specify the solution. We must therefore resort to physics-based and probabilistic *models* to disambiguate between potential solutions. However, modeling the visual world in all of its rich complexity is far more difficult than, say, modeling the vocal tract that produces spoken sounds.

The *forward* models that we use in computer vision are usually developed in physics (radiometry, optics, and sensor design) and in computer graphics. Both of these fields model how objects move and animate, how light reflects off their surfaces, is scattered by the atmosphere, refracted through camera lenses (or human eyes), and finally projected onto a flat (or curved) image plane. While computer graphics are not yet perfect (no fully computer-animated movie with human characters has yet succeeded at crossing the *uncanny valley*[2] that separates real humans from android robots and computer-animated humans), in limited domains, such as rendering a still scene composed of everyday objects or animating extinct creatures such as dinosaurs, the illusion of reality *is* perfect.

In computer vision, we are trying to do the inverse, i.e., to describe the world that we see in one or more images and to reconstruct its properties, such as shape, illumination, and color distributions. It is amazing that humans and animals do this so effortlessly, while computer vision algorithms are so error prone. People who have not worked in the field often underestimate the difficulty of the problem. (Colleagues at work often ask me for software to find and name all the people in photos, so they can get on with the more "interesting" work.) This

[1] http://www.michaelbach.de/ot/sze_muelue

[2] The term *uncanny valley* was originally coined by roboticist Masahiro Mori as applied to robotics (Mori 1970). It is also commonly applied to computer-animated films such as *Final Fantasy* and *Polar Express* (Geller 2008).

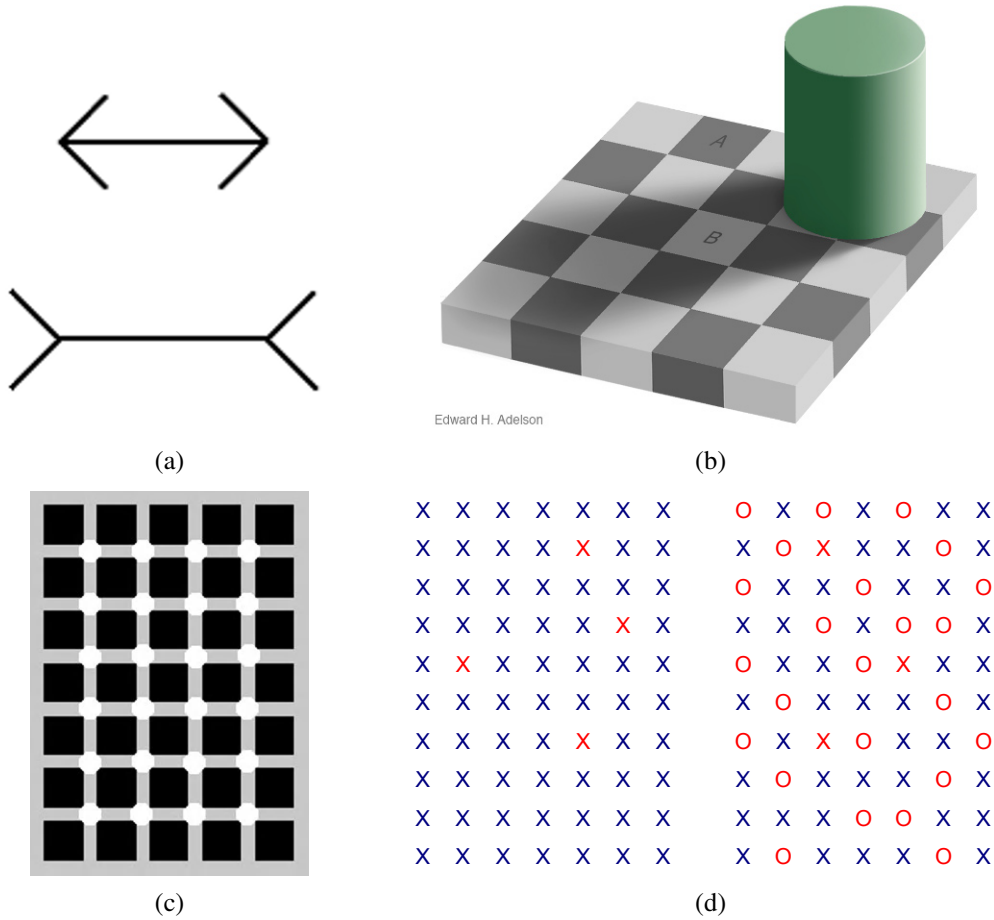


Figure 1.3 Some common optical illusions and what they might tell us about the visual system: (a) The classic Müller-Lyer illusion, where the length of the two horizontal lines appear different, probably due to the imagined perspective effects. (b) The "white" square B in the shadow and the "black" square A in the light actually have the same absolute intensity value. The percept is due to *brightness constancy*, the visual system's attempt to discount illumination when interpreting colors. Image courtesy of Ted Adelson, http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html. (c) A variation of the Hermann grid illusion, courtesy of Hany Farid, http://www.cs.dartmouth.edu/~farid/illusions/hermann.html. As you move your eyes over the figure, gray spots appear at the intersections. (d) Count the red *X*s in the left half of the figure. Now count them in the right half. Is it significantly harder? The explanation has to do with a *pop-out* effect (Treisman 1985), which tells us about the operations of parallel perception and integration pathways in the brain.

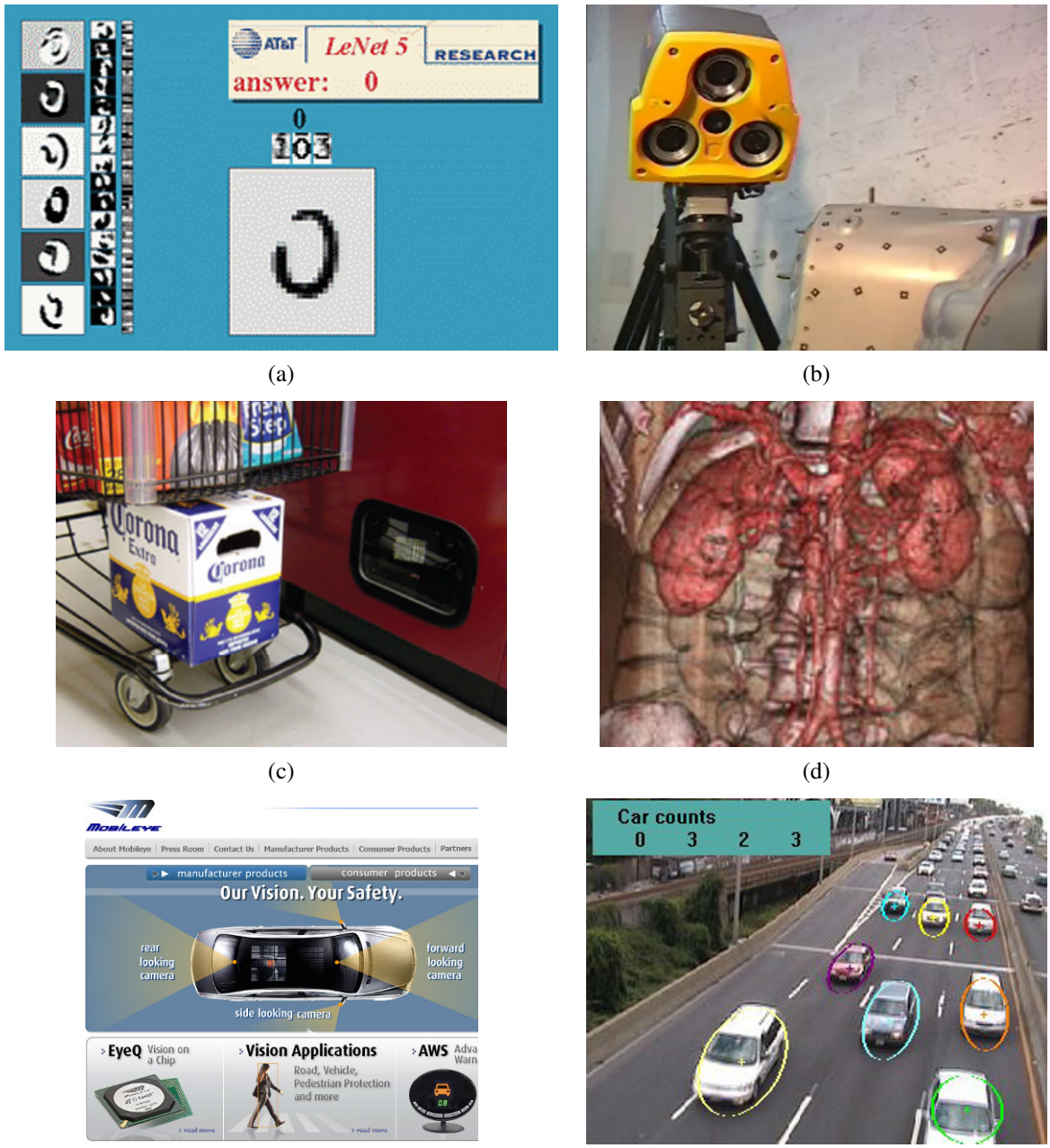
misperception that vision should be easy dates back to the early days of artificial intelligence (see Section 1.2), when it was initially believed that the *cognitive* (logic proving and planning) parts of intelligence were intrinsically more difficult than the *perceptual* components (Boden 2006).

The good news is that computer vision *is* being used today in a wide variety of real-world applications, which include:

- **Optical character recognition (OCR):** reading handwritten postal codes on letters (Figure 1.4a) and automatic number plate recognition (ANPR);
- **Machine inspection:** rapid parts inspection for quality assurance using stereo vision with specialized illumination to measure tolerances on aircraft wings or auto body parts (Figure 1.4b) or looking for defects in steel castings using X-ray vision;
- **Retail:** object recognition for automated checkout lanes (Figure 1.4c);
- **3D model building (photogrammetry):** fully automated construction of 3D models from aerial photographs used in systems such as Bing Maps;
- **Medical imaging:** registering pre-operative and intra-operative imagery (Figure 1.4d) or performing long-term studies of people's brain morphology as they age;
- **Automotive safety:** detecting unexpected obstacles such as pedestrians on the street, under conditions where active vision techniques such as radar or lidar do not work well (Figure 1.4e; see also Miller, Campbell, Huttenlocher *et al.* (2008); Montemerlo, Becker, Bhat *et al.* (2008); Urmson, Anhalt, Bagnell *et al.* (2008) for examples of fully automated driving);
- **Match move:** merging computer-generated imagery (CGI) with live action footage by tracking feature points in the source video to estimate the 3D camera motion and shape of the environment. Such techniques are widely used in Hollywood (e.g., in movies such as Jurassic Park) (Roble 1999; Roble and Zafar 2009); they also require the use of precise *matting* to insert new elements between foreground and background elements (Chuang, Agarwala, Curless *et al.* 2002).
- **Motion capture (mocap):** using retro-reflective markers viewed from multiple cameras or other vision-based techniques to capture actors for computer animation;
- **Surveillance:** monitoring for intruders, analyzing highway traffic (Figure 1.4f), and monitoring pools for drowning victims;
- **Fingerprint recognition and biometrics:** for automatic access authentication as well as forensic applications.

David Lowe's Web site of industrial vision applications (http://www.cs.ubc.ca/spider/lowe/vision.html) lists many other interesting industrial applications of computer vision. While the above applications are all extremely important, they mostly pertain to fairly specialized kinds of imagery and narrow domains.

In this book, we focus more on broader *consumer-level* applications, such as fun things you can do with your own personal photographs and video. These include:



(a) (b) (c) (d) (e) (f)

Figure 1.4 Some industrial applications of computer vision: (a) optical character recognition (OCR) http://yann.lecun.com/exdb/lenet/; (b) mechanical inspection http://www.cognitens.com/; (c) retail http://www.evoretail.com/; (d) medical imaging http://www.clarontech.com/; (e) automotive safety http://www.mobileye.com/; (f) surveillance and traffic monitoring http://www.honeywellvideo.com/, courtesy of Honeywell International Inc.

- **Stitching:** turning overlapping photos into a single seamlessly stitched panorama (Figure 1.5a), as described in Chapter 9;
- **Exposure bracketing:** merging multiple exposures taken under challenging lighting conditions (strong sunlight and shadows) into a single perfectly exposed image (Figure 1.5b), as described in Section 10.2;
- **Morphing:** turning a picture of one of your friends into another, using a seamless *morph* transition (Figure 1.5c);
- **3D modeling:** converting one or more snapshots into a 3D model of the object or person you are photographing (Figure 1.5d), as described in Section 12.6
- **Video match move and stabilization:** inserting 2D pictures or 3D models into your videos by automatically tracking nearby reference points (see Section 7.4.2)[3] or using motion estimates to remove shake from your videos (see Section 8.2.1);
- **Photo-based walkthroughs:** navigating a large collection of photographs, such as the interior of your house, by flying between different photos in 3D (see Sections 13.1.2 and 13.5.5)
- **Face detection:** for improved camera focusing as well as more relevant image searching (see Section 14.1.1);
- **Visual authentication:** automatically logging family members onto your home computer as they sit down in front of the webcam (see Section 14.2).

The great thing about these applications is that they are already familiar to most students; they are, at least, technologies that students can immediately appreciate and use with their own personal media. Since computer vision is a challenging topic, given the wide range of mathematics being covered[4] and the intrinsically difficult nature of the problems being solved, having fun and relevant problems to work on can be highly motivating and inspiring.

The other major reason why this book has a strong focus on applications is that they can be used to *formulate* and *constrain* the potentially open-ended problems endemic in vision. For example, if someone comes to me and asks for a good edge detector, my first question is usually to ask *why*? What kind of problem are they trying to solve and why do they believe that edge detection is an important component? If they are trying to locate faces, I usually point out that most successful face detectors use a combination of skin color detection (Exercise 2.8) and simple blob features Section 14.1.1; they do not rely on edge detection. If they are trying to match door and window edges in a building for the purpose of 3D reconstruction, I tell them that edges are a fine idea but it is better to tune the edge detector for long edges (see Sections 3.2.3 and 4.2) and link them together into straight lines with common vanishing points before matching (see Section 4.3).

Thus, it is better to think back from the problem at hand to suitable techniques, rather than to grab the first technique that you may have heard of. This kind of working back from

[3] For a fun student project on this topic, see the "PhotoBook" project at http://www.cc.gatech.edu/dvfx/videos/dvfx2005.html.

[4] These techniques include physics, Euclidean and projective geometry, statistics, and optimization. They make computer vision a fascinating field to study and a great way to learn techniques widely applicable in other fields.

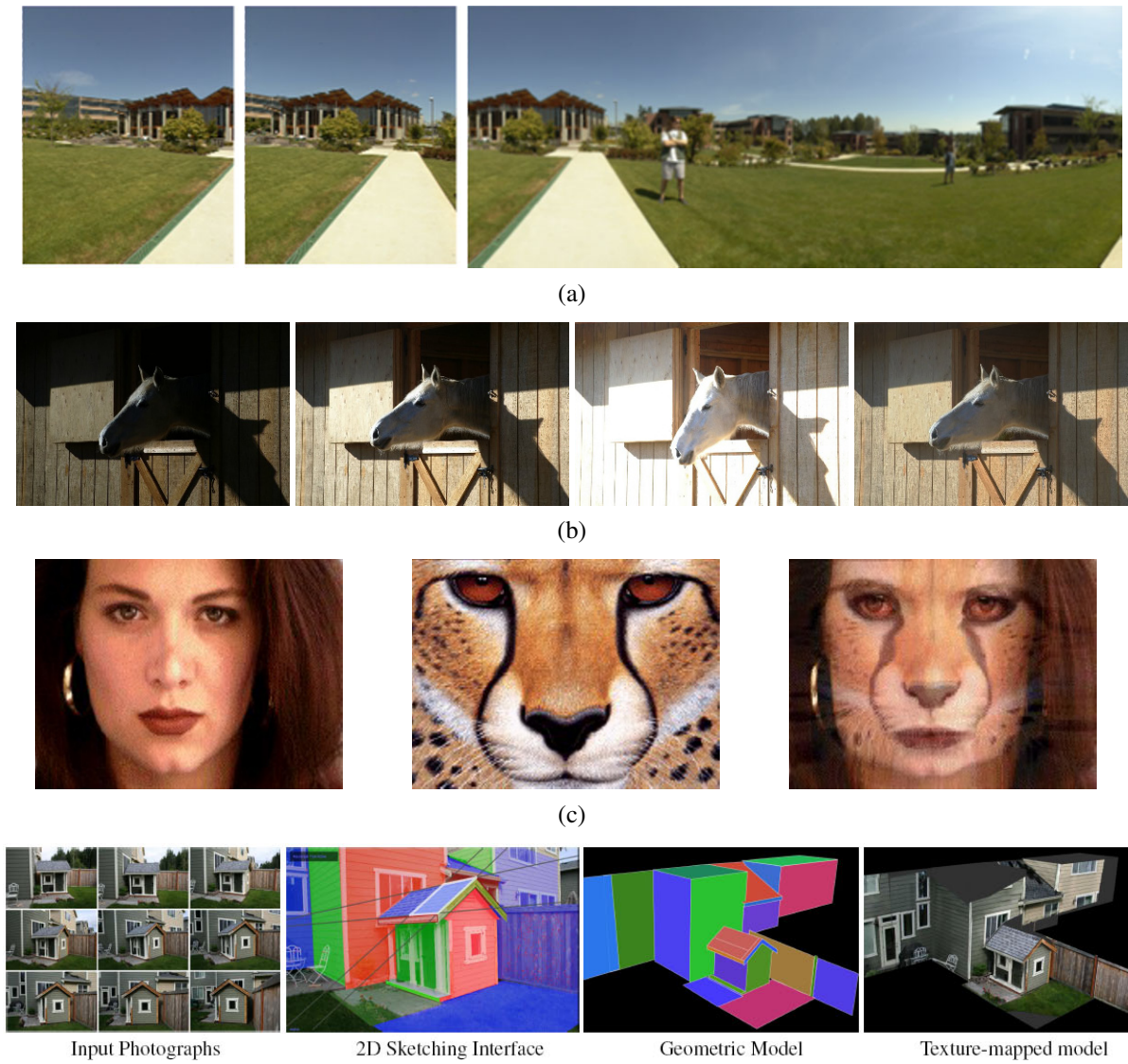


Figure 1.5 Some consumer applications of computer vision: (a) image stitching: merging different views (Szeliski and Shum 1997) © 1997 ACM; (b) exposure bracketing: merging different exposures; (c) morphing: blending between two photographs (Gomes, Darsa, Costa *et al.* 1999) © 1999 Morgan Kaufmann; (d) turning a collection of photographs into a 3D model (Sinha, Steedly, Szeliski *et al.* 2008) © 2008 ACM.

problems to solutions is typical of an **engineering** approach to the study of vision and reflects my own background in the field. First, I come up with a detailed problem definition and decide on the constraints and specifications for the problem. Then, I try to find out which techniques are known to work, implement a few of these, evaluate their performance, and finally make a selection. In order for this process to work, it is important to have realistic **test data**, both synthetic, which can be used to verify correctness and analyze noise sensitivity, and real-world data typical of the way the system will finally be used.

However, this book is not just an engineering text (a source of recipes). It also takes a **scientific** approach to basic vision problems. Here, I try to come up with the best possible models of the physics of the system at hand: how the scene is created, how light interacts with the scene and atmospheric effects, and how the sensors work, including sources of noise and uncertainty. The task is then to try to invert the acquisition process to come up with the best possible description of the scene.

The book often uses a **statistical** approach to formulating and solving computer vision problems. Where appropriate, probability distributions are used to model the scene and the noisy image acquisition process. The association of prior distributions with unknowns is often called *Bayesian modeling* (Appendix B). It is possible to associate a risk or loss function with mis-estimating the answer (Section B.2) and to set up your inference algorithm to minimize the expected risk. (Consider a robot trying to estimate the distance to an obstacle: it is usually safer to underestimate than to overestimate.) With statistical techniques, it often helps to gather lots of training data from which to learn probabilistic models. Finally, statistical approaches enable you to use proven inference techniques to estimate the best answer (or distribution of answers) and to quantify the uncertainty in the resulting estimates.

Because so much of computer vision involves the solution of inverse problems or the estimation of unknown quantities, my book also has a heavy emphasis on **algorithms**, especially those that are known to work well in practice. For many vision problems, it is all too easy to come up with a mathematical description of the problem that either does not match realistic real-world conditions or does not lend itself to the stable estimation of the unknowns. What we need are algorithms that are both **robust** to noise and deviation from our models and reasonably **efficient** in terms of run-time resources and space. In this book, I go into these issues in detail, using Bayesian techniques, where applicable, to ensure robustness, and efficient search, minimization, and linear system solving algorithms to ensure efficiency. Most of the algorithms described in this book are at a high level, being mostly a list of steps that have to be filled in by students or by reading more detailed descriptions elsewhere. In fact, many of the algorithms are sketched out in the exercises.

Now that I've described the goals of this book and the frameworks that I use, I devote the rest of this chapter to two additional topics. Section 1.2 is a brief synopsis of the history of computer vision. It can easily be skipped by those who want to get to "the meat" of the new material in this book and do not care as much about who invented what when.

The second is an overview of the book's contents, Section 1.3, which is useful reading for everyone who intends to make a study of this topic (or to jump in partway, since it describes chapter inter-dependencies). This outline is also useful for instructors looking to structure one or more courses around this topic, as it provides sample curricula based on the book's contents.

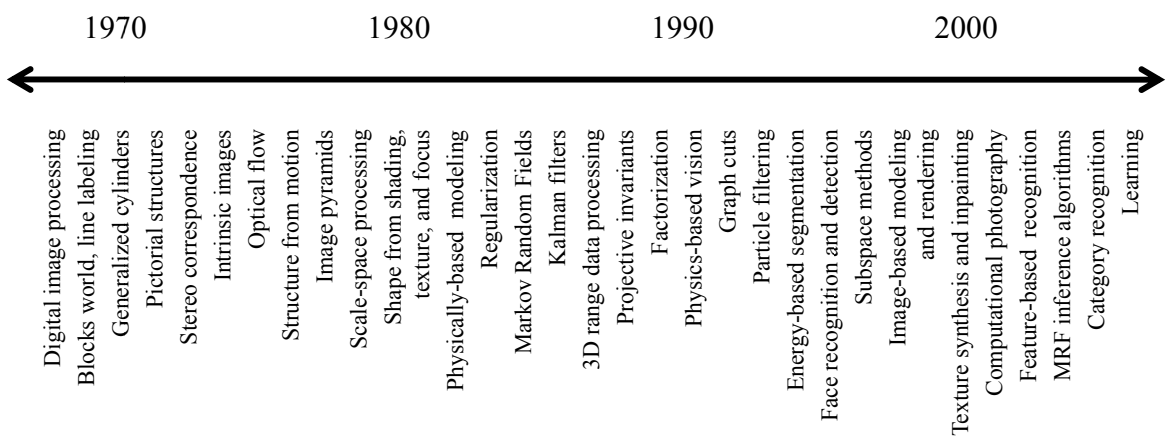


Figure 1.6 A rough timeline of some of the most active topics of research in computer vision.

1.2 A brief history

In this section, I provide a brief personal synopsis of the main developments in computer vision over the last 30 years (Figure 1.6); at least, those that I find personally interesting and which appear to have stood the test of time. Readers not interested in the provenance of various ideas and the evolution of this field should skip ahead to the book overview in Section 1.3.

1970s. When computer vision first started out in the early 1970s, it was viewed as the visual perception component of an ambitious agenda to mimic human intelligence and to endow robots with intelligent behavior. At the time, it was believed by some of the early pioneers of artificial intelligence and robotics (at places such as MIT, Stanford, and CMU) that solving the "visual input" problem would be an easy step along the path to solving more difficult problems such as higher-level reasoning and planning. According to one well-known story, in 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to "spend the summer linking a camera to a computer and getting the computer to describe what it saw" (Boden 2006, p. 781).[5] We now know that the problem is slightly more difficult than that.[6]

What distinguished computer vision from the already existing field of digital image processing (Rosenfeld and Pfaltz 1966; Rosenfeld and Kak 1976) was a desire to recover the three-dimensional structure of the world from images and to use this as a stepping stone towards full scene understanding. Winston (1975) and Hanson and Riseman (1978) provide two nice collections of classic papers from this early period.

Early attempts at scene understanding involved extracting edges and then inferring the 3D structure of an object or a "blocks world" from the topological structure of the 2D lines

[5] Boden (2006) cites (Crevier 1993) as the original source. The actual Vision Memo was authored by Seymour Papert (1966) and involved a whole cohort of students.

[6] To see how far robotic vision has come in the last four decades, have a look at the towel-folding robot at http://rll.eecs.berkeley.edu/pr/icra10/ (Maitin-Shepard, Cusumano-Towner, Lei *et al.* 2010).

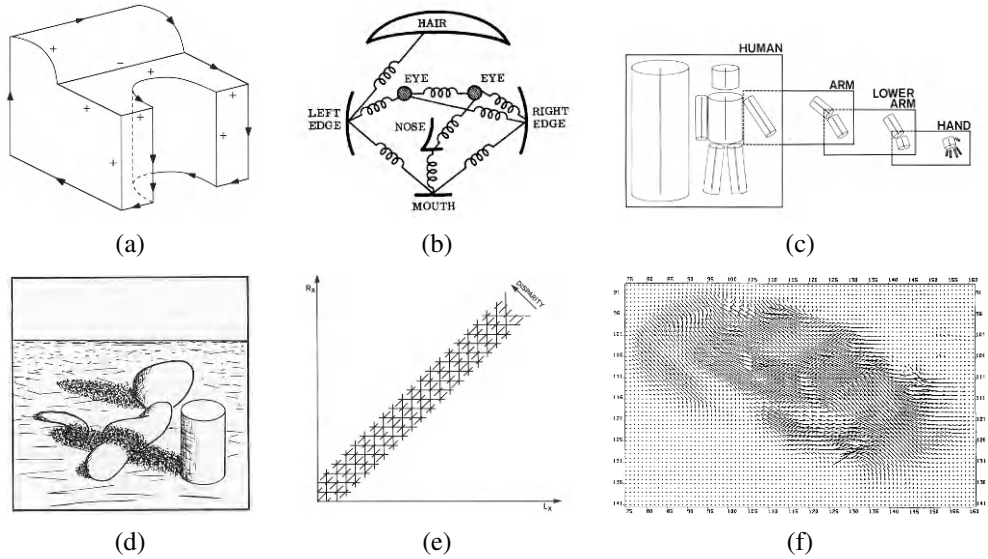


Figure 1.7 Some early (1970s) examples of computer vision algorithms: (a) line labeling (Nalwa 1993) © 1993 Addison-Wesley, (b) pictorial structures (Fischler and Elschlager 1973) © 1973 IEEE, (c) articulated body model (Marr 1982) © 1982 David Marr, (d) intrinsic images (Barrow and Tenenbaum 1981) © 1973 IEEE, (e) stereo correspondence (Marr 1982) © 1982 David Marr, (f) optical flow (Nagel and Enkelmann 1986) © 1986 IEEE.

(Roberts 1965). Several *line labeling* algorithms (Figure 1.7a) were developed at that time (Huffman 1971; Clowes 1971; Waltz 1975; Rosenfeld, Hummel, and Zucker 1976; Kanade 1980). Nalwa (1993) gives a nice review of this area. The topic of edge detection was also an active area of research; a nice survey of contemporaneous work can be found in (Davis 1975).

Three-dimensional modeling of non-polyhedral objects was also being studied (Baumgart 1974; Baker 1977). One popular approach used *generalized cylinders*, i.e., solids of revolution and swept closed curves (Agin and Binford 1976; Nevatia and Binford 1977), often arranged into parts relationships[7] (Hinton 1977; Marr 1982) (Figure 1.7c). Fischler and Elschlager (1973) called such *elastic* arrangements of parts *pictorial structures* (Figure 1.7b). This is currently one of the favored approaches being used in object recognition (see Section 14.4 and Felzenszwalb and Huttenlocher 2005).

A qualitative approach to understanding intensities and shading variations and explaining them by the effects of image formation phenomena, such as surface orientation and shadows, was championed by Barrow and Tenenbaum (1981) in their paper on *intrinsic images* (Figure 1.7d), along with the related *2½-D sketch* ideas of Marr (1982). This approach is again seeing a bit of a revival in the work of Tappen, Freeman, and Adelson (2005).

More quantitative approaches to computer vision were also developed at the time, including the first of many feature-based stereo correspondence algorithms (Figure 1.7e) (Dev 1974; Marr and Poggio 1976; Moravec 1977; Marr and Poggio 1979; Mayhew and Frisby 1981; Baker 1982; Barnard and Fischler 1982; Ohta and Kanade 1985; Grimson 1985; Pollard, Mayhew, and Frisby 1985; Prazdny 1985) and intensity-based optical flow algorithms

[7] In robotics and computer animation, these linked-part graphs are often called *kinematic chains*.

(Figure 1.7f) (Horn and Schunck 1981; Huang 1981; Lucas and Kanade 1981; Nagel 1986). The early work in simultaneously recovering 3D structure and camera motion (see Chapter 7) also began around this time (Ullman 1979; Longuet-Higgins 1981).

A lot of the philosophy of how vision was believed to work at the time is summarized in David Marr's (1982) book.[8] In particular, Marr introduced his notion of the three levels of description of a (visual) information processing system. These three levels, very loosely paraphrased according to my own interpretation, are:

- **Computational theory:** What is the goal of the computation (task) and what are the constraints that are known or can be brought to bear on the problem?
- **Representations and algorithms:** How are the input, output, and intermediate information represented and which algorithms are used to calculate the desired result?
- **Hardware implementation:** How are the representations and algorithms mapped onto actual hardware, e.g., a biological vision system or a specialized piece of silicon? Conversely, how can hardware constraints be used to guide the choice of representation and algorithm? With the increasing use of graphics chips (GPUs) and many-core architectures for computer vision (see Section C.2), this question is again becoming quite relevant.

As I mentioned earlier in this introduction, it is my conviction that a careful analysis of the problem specification and known constraints from image formation and priors (the scientific and statistical approaches) must be married with efficient and robust algorithms (the engineering approach) to design successful vision algorithms. Thus, it seems that Marr's philosophy is as good a guide to framing and solving problems in our field today as it was 25 years ago.

1980s. In the 1980s, a lot of attention was focused on more sophisticated mathematical techniques for performing quantitative image and scene analysis.

Image pyramids (see Section 3.5) started being widely used to perform tasks such as image blending (Figure 1.8a) and coarse-to-fine correspondence search (Rosenfeld 1980; Burt and Adelson 1983a,b; Rosenfeld 1984; Quam 1984; Anandan 1989). Continuous versions of pyramids using the concept of *scale-space* processing were also developed (Witkin 1983; Witkin, Terzopoulos, and Kass 1986; Lindeberg 1990). In the late 1980s, wavelets (see Section 3.5.4) started displacing or augmenting regular image pyramids in some applications (Adelson, Simoncelli, and Hingorani 1987; Mallat 1989; Simoncelli and Adelson 1990a,b; Simoncelli, Freeman, Adelson *et al.* 1992).

The use of stereo as a quantitative shape cue was extended by a wide variety of *shape-from-X* techniques, including shape from shading (Figure 1.8b) (see Section 12.1.1 and Horn 1975; Pentland 1984; Blake, Zimmerman, and Knowles 1985; Horn and Brooks 1986, 1989), photometric stereo (see Section 12.1.1 and Woodham 1981), shape from texture (see Section 12.1.2 and Witkin 1981; Pentland 1984; Malik and Rosenholtz 1997), and shape from focus (see Section 12.1.3 and Nayar, Watanabe, and Noguchi 1995). Horn (1986) has a nice discussion of most of these techniques.

[8] More recent developments in visual perception theory are covered in (Palmer 1999; Livingstone 2008).

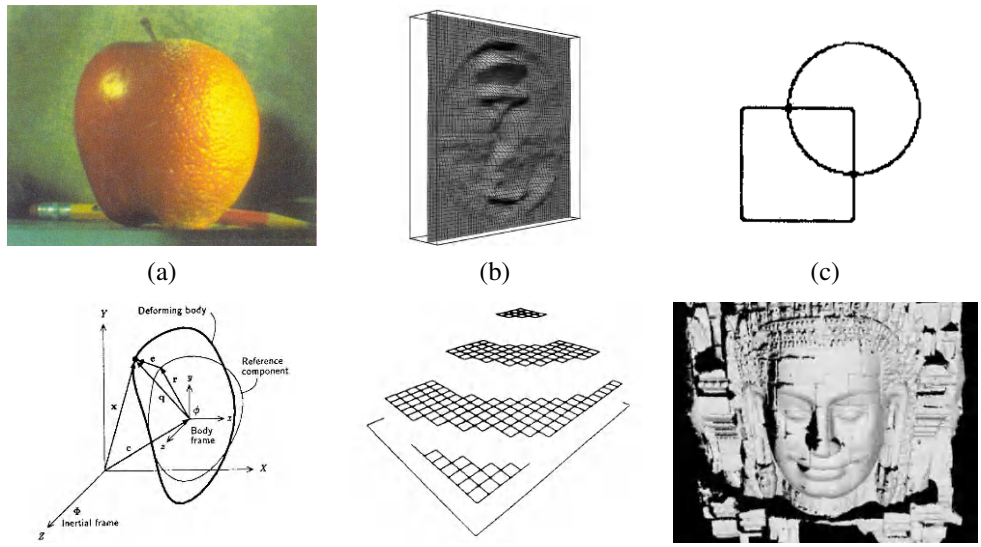


Figure 1.8 Examples of computer vision algorithms from the 1980s: (a) pyramid blending (Burt and Adelson 1983b) © 1983 ACM, (b) shape from shading (Freeman and Adelson 1991) © 1991 IEEE, (c) edge detection (Freeman and Adelson 1991) © 1991 IEEE, (d) physically based models (Terzopoulos and Witkin 1988) © 1988 IEEE, (e) regularization-based surface reconstruction (Terzopoulos 1988) © 1988 IEEE, (f) range data acquisition and merging (Banno, Masuda, Oishi *et al.* 2008) © 2008 Springer.

Research into better edge and contour detection (Figure 1.8c) (see Section 4.2) was also active during this period (Canny 1986; Nalwa and Binford 1986), including the introduction of dynamically evolving contour trackers (Section 5.1.1) such as *snakes* (Kass, Witkin, and Terzopoulos 1988), as well as three-dimensional *physically based models* (Figure 1.8d) (Terzopoulos, Witkin, and Kass 1987; Kass, Witkin, and Terzopoulos 1988; Terzopoulos and Fleischer 1988; Terzopoulos, Witkin, and Kass 1988).

Researchers noticed that a lot of the stereo, flow, shape-from-X, and edge detection algorithms could be unified, or at least described, using the same mathematical framework if they were posed as variational optimization problems (see Section 3.7) and made more robust (well-posed) using regularization (Figure 1.8e) (see Section 3.7.1 and Terzopoulos 1983; Poggio, Torre, and Koch 1985; Terzopoulos 1986b; Blake and Zisserman 1987; Bertero, Poggio, and Torre 1988; Terzopoulos 1988). Around the same time, Geman and Geman (1984) pointed out that such problems could equally well be formulated using discrete *Markov Random Field* (MRF) models (see Section 3.7.2), which enabled the use of better (global) search and optimization algorithms, such as simulated annealing.

Online variants of MRF algorithms that modeled and updated uncertainties using the Kalman filter were introduced a little later (Dickmanns and Graefe 1988; Matthies, Kanade, and Szeliski 1989; Szeliski 1989). Attempts were also made to map both regularized and MRF algorithms onto parallel hardware (Poggio and Koch 1985; Poggio, Little, Gamble *et al.* 1988; Fischler, Firschein, Barnard *et al.* 1989). The book by Fischler and Firschein (1987) contains a nice collection of articles focusing on all of these topics (stereo, flow,

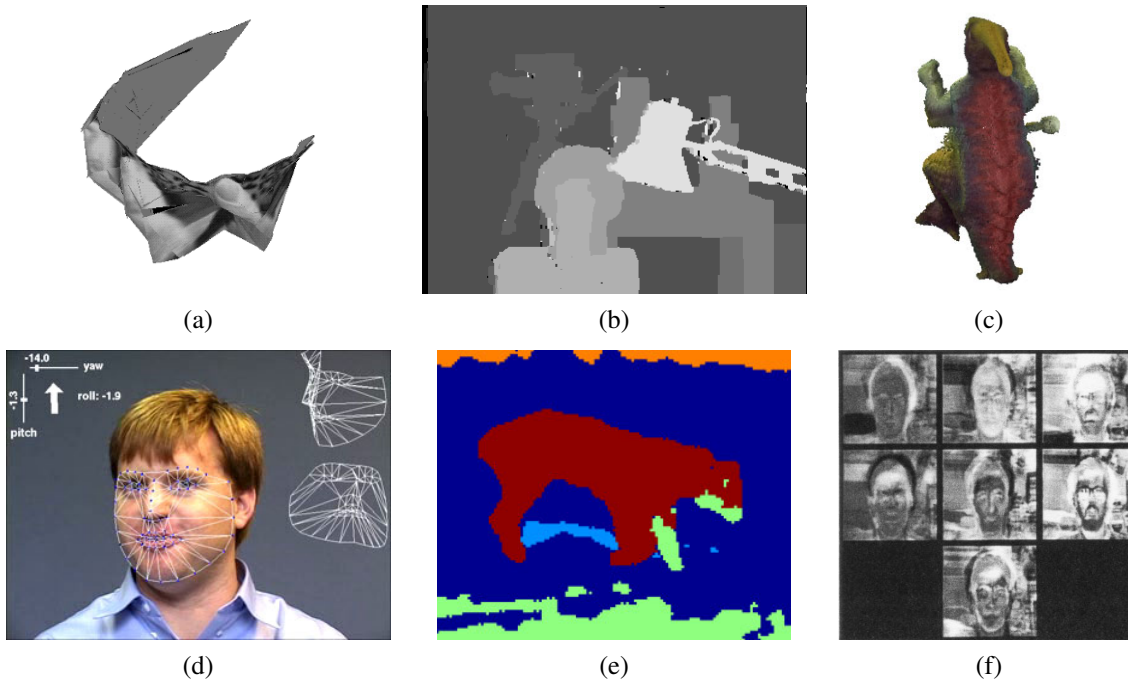


Figure 1.9 Examples of computer vision algorithms from the 1990s: (a) factorization-based structure from motion (Tomasi and Kanade 1992) © 1992 Springer, (b) dense stereo matching (Boykov, Veksler, and Zabih 2001), (c) multi-view reconstruction (Seitz and Dyer 1999) © 1999 Springer, (d) face tracking (Matthews, Xiao, and Baker 2007), (e) image segmentation (Belongie, Fowlkes, Chung *et al.* 2002) © 2002 Springer, (f) face recognition (Turk and Pentland 1991a).

regularization, MRFs, and even higher-level vision).

Three-dimensional range data processing (acquisition, merging, modeling, and recognition; see Figure 1.8f) continued being actively explored during this decade (Agin and Binford 1976; Besl and Jain 1985; Faugeras and Hebert 1987; Curless and Levoy 1996). The compilation by Kanade (1987) contains a lot of the interesting papers in this area.

1990s. While a lot of the previously mentioned topics continued to be explored, a few of them became significantly more active.

A burst of activity in using projective invariants for recognition (Mundy and Zisserman 1992) evolved into a concerted effort to solve the structure from motion problem (see Chapter 7). A lot of the initial activity was directed at *projective reconstructions*, which did not require knowledge of camera calibration (Faugeras 1992; Hartley, Gupta, and Chang 1992; Hartley 1994a; Faugeras and Luong 2001; Hartley and Zisserman 2004). Simultaneously, *factorization* techniques (Section 7.3) were developed to solve efficiently problems for which orthographic camera approximations were applicable (Figure 1.9a) (Tomasi and Kanade 1992; Poelman and Kanade 1997; Anandan and Irani 2002) and then later extended to the perspective case (Christy and Horaud 1996; Triggs 1996). Eventually, the field started using full global optimization (see Section 7.4 and Taylor, Kriegman, and Anandan 1991; Szeliski and

Kang 1994; Azarbayejani and Pentland 1995), which was later recognized as being the same as the *bundle adjustment* techniques traditionally used in photogrammetry (Triggs, McLauchlan, Hartley *et al.* 1999). Fully automated (sparse) 3D modeling systems were built using such techniques (Beardsley, Torr, and Zisserman 1996; Schaffalitzky and Zisserman 2002; Brown and Lowe 2003; Snavely, Seitz, and Szeliski 2006).

Work begun in the 1980s on using detailed measurements of color and intensity combined with accurate physical models of radiance transport and color image formation created its own subfield known as *physics-based vision*. A good survey of the field can be found in the three-volume collection on this topic (Wolff, Shafer, and Healey 1992a; Healey and Shafer 1992; Shafer, Healey, and Wolff 1992).

Optical flow methods (see Chapter 8) continued to be improved (Nagel and Enkelmann 1986; Bolles, Baker, and Marimont 1987; Horn and Weldon Jr. 1988; Anandan 1989; Bergen, Anandan, Hanna *et al.* 1992; Black and Anandan 1996; Bruhn, Weickert, and Schnörr 2005; Papenberg, Bruhn, Brox *et al.* 2006), with (Nagel 1986; Barron, Fleet, and Beauchemin 1994; Baker, Black, Lewis *et al.* 2007) being good surveys. Similarly, a lot of progress was made on dense stereo correspondence algorithms (see Chapter 11, Okutomi and Kanade (1993, 1994); Boykov, Veksler, and Zabih (1998); Birchfield and Tomasi (1999); Boykov, Veksler, and Zabih (2001), and the survey and comparison in Scharstein and Szeliski (2002)), with the biggest breakthrough being perhaps global optimization using *graph cut* techniques (Figure 1.9b) (Boykov, Veksler, and Zabih 2001).

Multi-view stereo algorithms (Figure 1.9c) that produce complete 3D surfaces (see Section 11.6) were also an active topic of research (Seitz and Dyer 1999; Kutulakos and Seitz 2000) that continues to be active today (Seitz, Curless, Diebel *et al.* 2006). Techniques for producing 3D volumetric descriptions from binary silhouettes (see Section 11.6.2) continued to be developed (Potmesil 1987; Srivasan, Liang, and Hackwood 1990; Szeliski 1993; Laurentini 1994), along with techniques based on tracking and reconstructing smooth occluding contours (see Section 11.2.1 and Cipolla and Blake 1992; Vaillant and Faugeras 1992; Zheng 1994; Boyer and Berger 1997; Szeliski and Weiss 1998; Cipolla and Giblin 2000).

Tracking algorithms also improved a lot, including contour tracking using *active contours* (see Section 5.1), such as *snakes* (Kass, Witkin, and Terzopoulos 1988), *particle filters* (Blake and Isard 1998), and *level sets* (Malladi, Sethian, and Vemuri 1995), as well as intensity-based (*direct*) techniques (Lucas and Kanade 1981; Shi and Tomasi 1994; Rehg and Kanade 1994), often applied to tracking faces (Figure 1.9d) (Lanitis, Taylor, and Cootes 1997; Matthews and Baker 2004; Matthews, Xiao, and Baker 2007) and whole bodies (Sidenbladh, Black, and Fleet 2000; Hilton, Fua, and Ronfard 2006; Moeslund, Hilton, and Krüger 2006).

Image segmentation (see Chapter 5) (Figure 1.9e), a topic which has been active since the earliest days of computer vision (Brice and Fennema 1970; Horowitz and Pavlidis 1976; Riseman and Arbib 1977; Rosenfeld and Davis 1979; Haralick and Shapiro 1985; Pavlidis and Liow 1990), was also an active topic of research, producing techniques based on minimum energy (Mumford and Shah 1989) and minimum description length (Leclerc 1989), *normalized cuts* (Shi and Malik 2000), and *mean shift* (Comaniciu and Meer 2002).

Statistical learning techniques started appearing, first in the application of principal component *eigenface* analysis to face recognition (Figure 1.9f) (see Section 14.2.1 and Turk and Pentland 1991a) and linear dynamical systems for curve tracking (see Section 5.1.1 and Blake and Isard 1998).

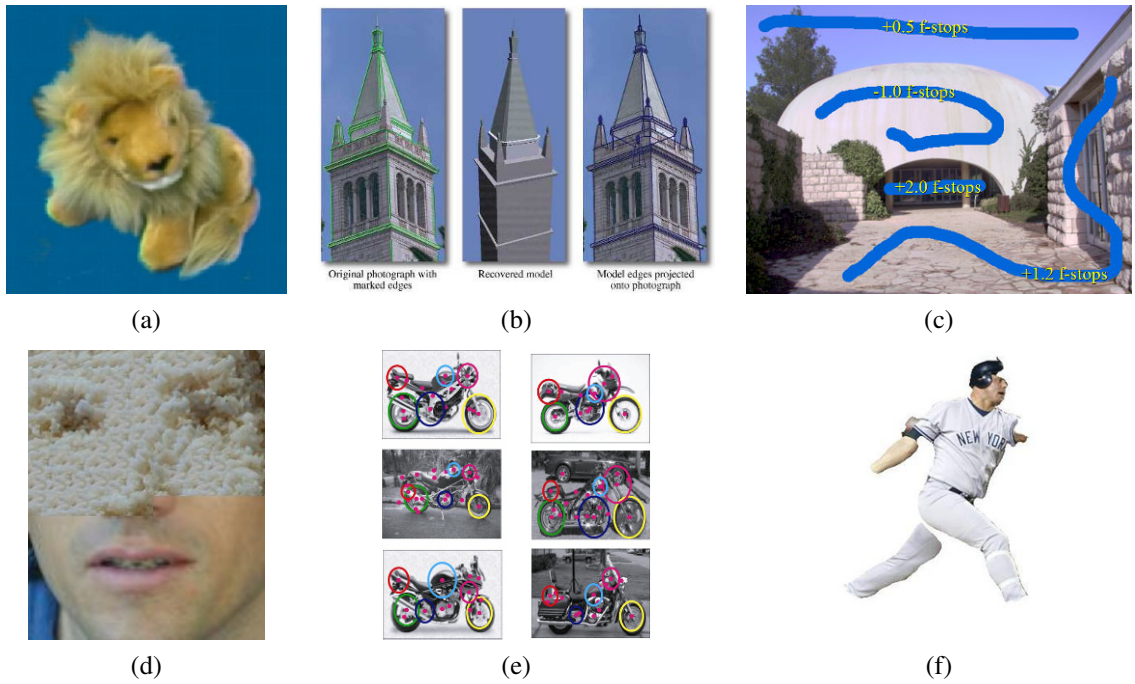


Figure 1.10 Recent examples of computer vision algorithms: (a) image-based rendering (Gortler, Grzeszczuk, Szeliski *et al.* 1996), (b) image-based modeling (Debevec, Taylor, and Malik 1996) © 1996 ACM, (c) interactive tone mapping (Lischinski, Farbman, Uyttendaele *et al.* 2006a) (d) texture synthesis (Efros and Freeman 2001), (e) feature-based recognition (Fergus, Perona, and Zisserman 2007), (f) region-based recognition (Mori, Ren, Efros *et al.* 2004) © 2004 IEEE.

Perhaps the most notable development in computer vision during this decade was the increased interaction with computer graphics (Seitz and Szeliski 1999), especially in the cross-disciplinary area of *image-based modeling and rendering* (see Chapter 13). The idea of manipulating real-world imagery directly to create new animations first came to prominence with *image morphing* techniques (Figure1.5c) (see Section 3.6.3 and Beier and Neely 1992) and was later applied to *view interpolation* (Chen and Williams 1993; Seitz and Dyer 1996), panoramic image stitching (Figure1.5a) (see Chapter 9 and Mann and Picard 1994; Chen 1995; Szeliski 1996; Szeliski and Shum 1997; Szeliski 2006a), and full light-field rendering (Figure 1.10a) (see Section 13.3 and Gortler, Grzeszczuk, Szeliski *et al.* 1996; Levoy and Hanrahan 1996; Shade, Gortler, He *et al.* 1998). At the same time, image-based modeling techniques (Figure 1.10b) for automatically creating realistic 3D models from collections of images were also being introduced (Beardsley, Torr, and Zisserman 1996; Debevec, Taylor, and Malik 1996; Taylor, Debevec, and Malik 1996).

2000s. This past decade has continued to see a deepening interplay between the vision and graphics fields. In particular, many of the topics introduced under the rubric of image-based rendering, such as image stitching (see Chapter 9), light-field capture and rendering (see Section 13.3), and *high dynamic range* (HDR) image capture through exposure bracketing (Figure1.5b) (see Section 10.2 and Mann and Picard 1995; Debevec and Malik 1997), were

re-christened as *computational photography* (see Chapter 10) to acknowledge the increased use of such techniques in everyday digital photography. For example, the rapid adoption of exposure bracketing to create high dynamic range images necessitated the development of *tone mapping* algorithms (Figure 1.10c) (see Section 10.2.1) to convert such images back to displayable results (Fattal, Lischinski, and Werman 2002; Durand and Dorsey 2002; Reinhard, Stark, Shirley *et al.* 2002; Lischinski, Farbman, Uyttendaele *et al.* 2006a). In addition to merging multiple exposures, techniques were developed to merge flash images with non-flash counterparts (Eisemann and Durand 2004; Petschnigg, Agrawala, Hoppe *et al.* 2004) and to interactively or automatically select different regions from overlapping images (Agarwala, Dontcheva, Agrawala *et al.* 2004).

Texture synthesis (Figure 1.10d) (see Section 10.5), quilting (Efros and Leung 1999; Efros and Freeman 2001; Kwatra, Schödl, Essa *et al.* 2003) and inpainting (Bertalmio, Sapiro, Caselles *et al.* 2000; Bertalmio, Vese, Sapiro *et al.* 2003; Criminisi, Pérez, and Toyama 2004) are additional topics that can be classified as computational photography techniques, since they re-combine input image samples to produce new photographs.

A second notable trend during this past decade has been the emergence of feature-based techniques (combined with learning) for object recognition (see Section 14.3 and Ponce, Hebert, Schmid *et al.* 2006). Some of the notable papers in this area include the *constellation model* of Fergus, Perona, and Zisserman (2007) (Figure 1.10e) and the *pictorial structures* of Felzenszwalb and Huttenlocher (2005). Feature-based techniques also dominate other recognition tasks, such as scene recognition (Zhang, Marszalek, Lazebnik *et al.* 2007) and panorama and location recognition (Brown and Lowe 2007; Schindler, Brown, and Szeliski 2007). And while *interest point* (patch-based) features tend to dominate current research, some groups are pursuing recognition based on contours (Belongie, Malik, and Puzicha 2002) and region segmentation (Figure 1.10f) (Mori, Ren, Efros *et al.* 2004).

Another significant trend from this past decade has been the development of more efficient algorithms for complex global optimization problems (see Sections 3.7 and B.5 and Szeliski, Zabih, Scharstein *et al.* 2008; Blake, Kohli, and Rother 2010). While this trend began with work on graph cuts (Boykov, Veksler, and Zabih 2001; Kohli and Torr 2007), a lot of progress has also been made in message passing algorithms, such as *loopy belief propagation* (LBP) (Yedidia, Freeman, and Weiss 2001; Kumar and Torr 2006).

The final trend, which now dominates a lot of the visual recognition research in our community, is the application of sophisticated machine learning techniques to computer vision problems (see Section 14.5.1 and Freeman, Perona, and Schölkopf 2008). This trend coincides with the increased availability of immense quantities of partially labelled data on the Internet, which makes it more feasible to learn object categories without the use of careful human supervision.

1.3 Book overview

In the final part of this introduction, I give a brief tour of the material in this book, as well as a few notes on notation and some additional general references. Since computer vision is such a broad field, it is possible to study certain aspects of it, e.g., geometric image formation and 3D structure recovery, without engaging other parts, e.g., the modeling of reflectance and shading. Some of the chapters in this book are only loosely coupled with others, and it is not

strictly necessary to read all of the material in sequence.

Figure 1.11 shows a rough layout of the contents of this book. Since computer vision involves going from images to a structural description of the scene (and computer graphics the converse), I have positioned the chapters horizontally in terms of which major component they address, in addition to vertically according to their dependence.

Going from left to right, we see the major column headings as Images (which are 2D in nature), Geometry (which encompasses 3D descriptions), and Photometry (which encompasses object appearance). (An alternative labeling for these latter two could also be *shape* and *appearance*—see, e.g., Chapter 13 and Kang, Szeliski, and Anandan (2000).) Going from top to bottom, we see increasing levels of modeling and abstraction, as well as techniques that build on previously developed algorithms. Of course, this taxonomy should be taken with a large grain of salt, as the processing and dependencies in this diagram are not strictly sequential and subtle additional dependencies and relationships also exist (e.g., some recognition techniques make use of 3D information). The placement of topics along the horizontal axis should also be taken lightly, as most vision algorithms involve mapping between at least two different representations.[9]

Interspersed throughout the book are sample **applications**, which relate the algorithms and mathematical material being presented in various chapters to useful, real-world applications. Many of these applications are also presented in the exercises sections, so that students can write their own.

At the end of each section, I provide a set of **exercises** that the students can use to implement, test, and refine the algorithms and techniques presented in each section. Some of the exercises are suitable as written homework assignments, others as shorter one-week projects, and still others as open-ended research problems that make for challenging final projects. Motivated students who implement a reasonable subset of these exercises will, by the end of the book, have a computer vision software library that can be used for a variety of interesting tasks and projects.

As a reference book, I try wherever possible to discuss which techniques and algorithms work well in practice, as well as providing up-to-date pointers to the latest research results in the areas that I cover. The exercises can be used to build up your own personal library of self-tested and validated vision algorithms, which is more worthwhile in the long term (assuming you have the time) than simply pulling algorithms out of a library whose performance you do not really understand.

The book begins in Chapter 2 with a review of the image formation processes that create the images that we see and capture. Understanding this process is fundamental if you want to take a scientific (model-based) approach to computer vision. Students who are eager to just start implementing algorithms (or courses that have limited time) can skip ahead to the next chapter and dip into this material later. In Chapter 2, we break down image formation into three major components. Geometric image formation (Section 2.1) deals with points, lines, and planes, and how these are mapped onto images using *projective geometry* and other models (including radial lens distortion). Photometric image formation (Section 2.2) covers *radiometry*, which describes how light interacts with surfaces in the world, and *optics*, which projects light onto the sensor plane. Finally, Section 2.3 covers how sensors work, including

[9] For an interesting comparison with what is known about the human visual system, e.g., the largely parallel *what* and *where* pathways, see some textbooks on human perception (Palmer 1999; Livingstone 2008).

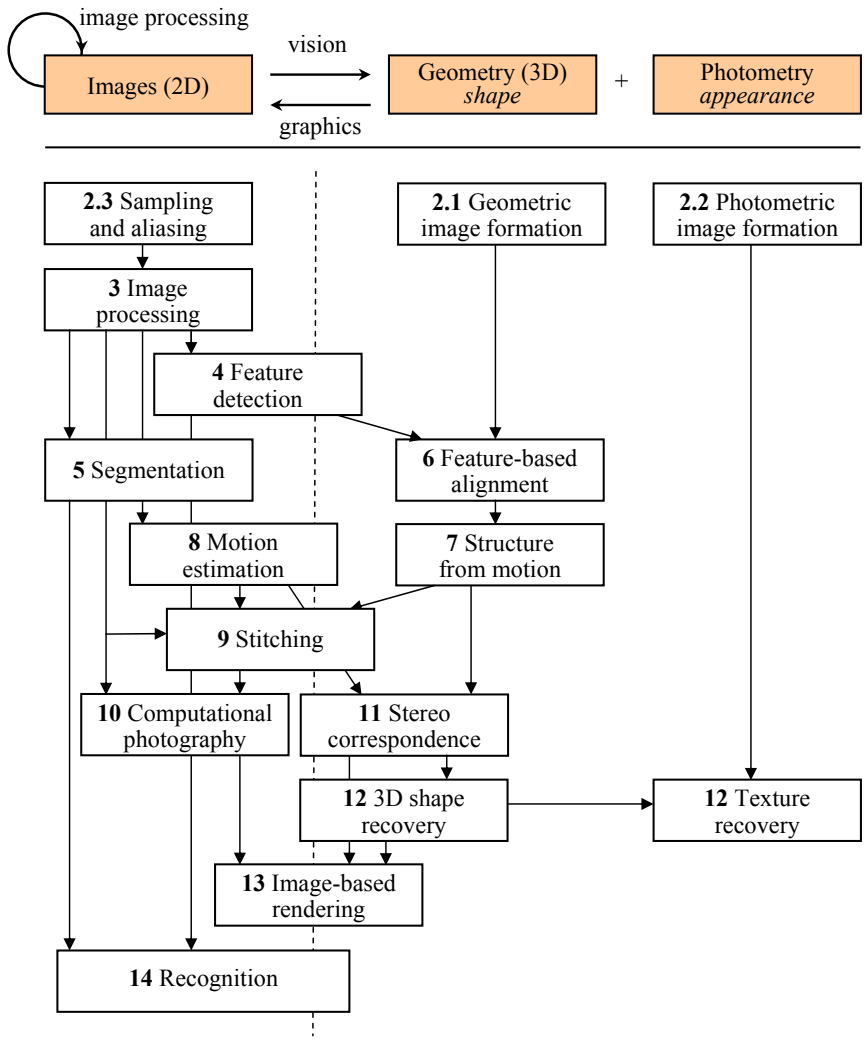
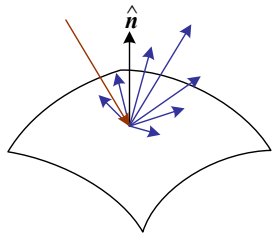
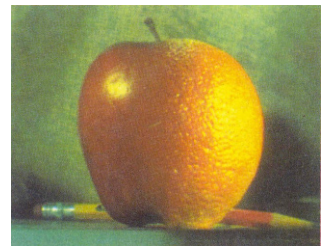


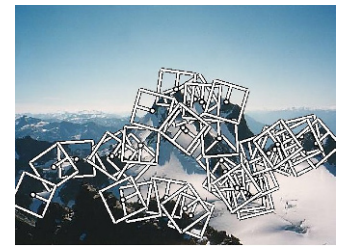
Figure 1.11 Relationship between images, geometry, and photometry, as well as a taxonomy of the topics covered in this book. Topics are roughly positioned along the left–right axis depending on whether they are more closely related to image-based (left), geometry-based (middle) or appearance-based (right) representations, and on the vertical axis by increasing level of abstraction. The whole figure should be taken with a large grain of salt, as there are many additional subtle connections between topics not illustrated here.



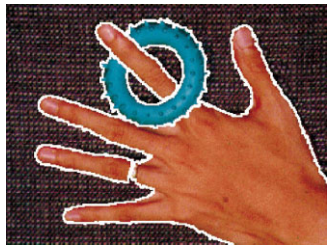
2. Image Formation



3. Image Processing



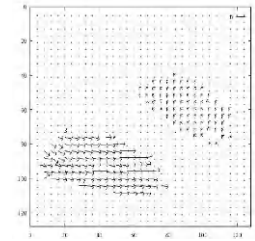
4. Features



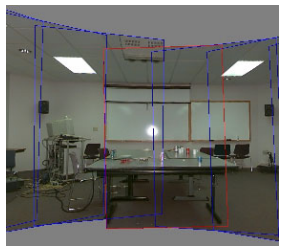
5. Segmentation



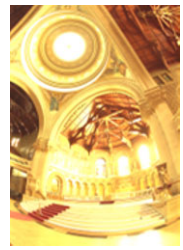
6-7. Structure from Motion

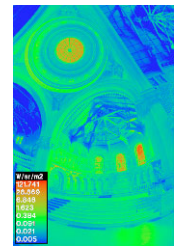


8. Motion

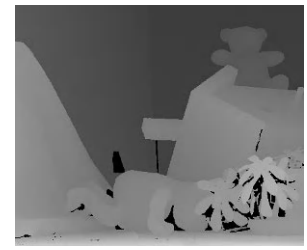


9. Stitching





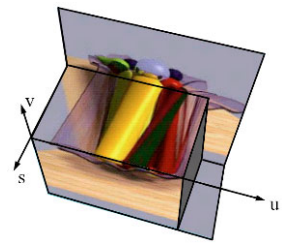
10. Computational Photography



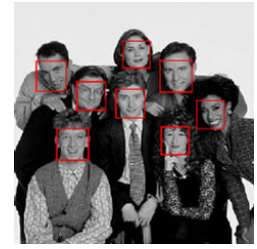
11. Stereo



12. 3D Shape



13. Image-based Rendering



14. Recognition

Figure 1.12 A pictorial summary of the chapter contents. Sources: Brown, Szeliski, and Winder (2005); Comaniciu and Meer (2002); Snavely, Seitz, and Szeliski (2006); Nagel and Enkelmann (1986); Szeliski and Shum (1997); Debevec and Malik (1997); Gortler, Grzeszczuk, Szeliski *et al.* (1996); Viola and Jones (2004)—see the figures in the respective chapters for copyright information.

topics such as sampling and aliasing, color sensing, and in-camera compression.

Chapter 3 covers image processing, which is needed in almost all computer vision applications. This includes topics such as linear and non-linear filtering (Section 3.3), the Fourier transform (Section 3.4), image pyramids and wavelets (Section 3.5), geometric transformations such as image warping (Section 3.6), and global optimization techniques such as *regularization* and *Markov Random Fields* (MRFs) (Section 3.7). While most of this material is covered in courses and textbooks on image processing, the use of optimization techniques is more typically associated with computer vision (although MRFs are now being widely used in image processing as well). The section on MRFs is also the first introduction to the use of Bayesian inference techniques, which are covered at a more abstract level in Appendix B. Chapter 3 also presents applications such as seamless image blending and image restoration.

In Chapter 4, we cover feature detection and matching. A lot of current 3D reconstruction and recognition techniques are built on extracting and matching *feature points* (Section 4.1), so this is a fundamental technique required by many subsequent chapters (Chapters 6, 7, 9 and 14). We also cover edge and straight line detection in Sections 4.2 and 4.3.

Chapter 5 covers region segmentation techniques, including active contour detection and tracking (Section 5.1). Segmentation techniques include top-down (split) and bottom-up (merge) techniques, mean shift techniques that find modes of clusters, and various graph-based segmentation approaches. All of these techniques are essential building blocks that are widely used in a variety of applications, including performance-driven animation, interactive image editing, and recognition.

In Chapter 6, we cover geometric alignment and camera calibration. We introduce the basic techniques of feature-based alignment in Section 6.1 and show how this problem can be solved using either linear or non-linear least squares, depending on the motion involved. We also introduce additional concepts, such as uncertainty weighting and robust regression, which are essential to making real-world systems work. Feature-based alignment is then used as a building block for 3D pose estimation (*extrinsic calibration*) in Section 6.2 and camera (*intrinsic*) calibration in Section 6.3. Chapter 6 also describes applications of these techniques to photo alignment for flip-book animations, 3D pose estimation from a hand-held camera, and single-view reconstruction of building models.

Chapter 7 covers the topic of *structure from motion*, which involves the simultaneous recovery of 3D camera motion and 3D scene structure from a collection of tracked 2D features. This chapter begins with the easier problem of 3D point *triangulation* (Section 7.1), which is the 3D reconstruction of points from matched features when the camera positions are known. It then describes two-frame structure from motion (Section 7.2), for which algebraic techniques exist, as well as robust sampling techniques such as RANSAC that can discount erroneous feature matches. The second half of Chapter 7 describes techniques for multi-frame structure from motion, including factorization (Section 7.3), bundle adjustment (Section 7.4), and constrained motion and structure models (Section 7.5). It also presents applications in view morphing, sparse 3D model construction, and match move.

In Chapter 8, we go back to a topic that deals directly with image intensities (as opposed to feature tracks), namely dense intensity-based motion estimation (*optical flow*). We start with the simplest possible motion models, translational motion (Section 8.1), and cover topics such as hierarchical (coarse-to-fine) motion estimation, Fourier-based techniques, and iterative refinement. We then present parametric motion models, which can be used to com-

pensate for camera rotation and zooming, as well as affine or planar perspective motion (Section 8.2). This is then generalized to spline-based motion models (Section 8.3) and finally to general per-pixel optical flow (Section 8.4), including layered and learned motion models (Section 8.5). Applications of these techniques include automated morphing, frame interpolation (slow motion), and motion-based user interfaces.

Chapter 9 is devoted to *image stitching*, i.e., the construction of large panoramas and composites. While stitching is just one example of *computation photography* (see Chapter 10), there is enough depth here to warrant a separate chapter. We start by discussing various possible motion models (Section 9.1), including planar motion and pure camera rotation. We then discuss global alignment (Section 9.2), which is a special (simplified) case of general bundle adjustment, and then present *panorama recognition*, i.e., techniques for automatically discovering which images actually form overlapping panoramas. Finally, we cover the topics of *image compositing* and *blending* (Section 9.3), which involve both selecting which pixels from which images to use and blending them together so as to disguise exposure differences.

Image stitching is a wonderful application that ties together most of the material covered in earlier parts of this book. It also makes for a good mid-term course project that can build on previously developed techniques such as image warping and feature detection and matching. Chapter 9 also presents more specialized variants of stitching such as whiteboard and document scanning, video summarization, *panography*, full 360° spherical panoramas, and interactive photomontage for blending repeated action shots together.

Chapter 10 presents additional examples of *computational photography*, which is the process of creating new images from one or more input photographs, often based on the careful modeling and calibration of the image formation process (Section 10.1). Computational photography techniques include merging multiple exposures to create *high dynamic range* images (Section 10.2), increasing image resolution through blur removal and *super-resolution* (Section 10.3), and image editing and compositing operations (Section 10.4). We also cover the topics of texture analysis, synthesis and *inpainting* (hole filling) in Section 10.5, as well as non-photorealistic rendering (Section 10.5.2).

In Chapter 11, we turn to the issue of stereo correspondence, which can be thought of as a special case of motion estimation where the camera positions are already known (Section 11.1). This additional knowledge enables stereo algorithms to search over a much smaller space of correspondences and, in many cases, to produce dense depth estimates that can be converted into visible surface models (Section 11.3). We also cover multi-view stereo algorithms that build a true 3D surface representation instead of just a single depth map (Section 11.6). Applications of stereo matching include head and gaze tracking, as well as depth-based background replacement (*Z-keying*).

Chapter 12 covers additional 3D shape and appearance modeling techniques. These include classic *shape-from-X* techniques such as shape from shading, shape from texture, and shape from focus (Section 12.1), as well as shape from smooth occluding contours (Section 11.2.1) and silhouettes (Section 12.5). An alternative to all of these *passive* computer vision techniques is to use *active rangefinding* (Section 12.2), i.e., to project patterned light onto scenes and recover the 3D geometry through triangulation. Processing all of these 3D representations often involves interpolating or simplifying the geometry (Section 12.3), or using alternative representations such as surface point sets (Section 12.4).

The collection of techniques for going from one or more images to partial or full 3D

models is often called *image-based modeling* or *3D photography*. Section 12.6 examines three more specialized application areas (architecture, faces, and human bodies), which can use *model-based reconstruction* to fit parameterized models to the sensed data. Section 12.7 examines the topic of *appearance modeling*, i.e., techniques for estimating the texture maps, albedos, or even sometimes complete *bi-directional reflectance distribution functions* (BRDFs) that describe the appearance of 3D surfaces.

In Chapter 13, we discuss the large number of image-based rendering techniques that have been developed in the last two decades, including simpler techniques such as view interpolation (Section 13.1), layered depth images (Section 13.2), and sprites and layers (Section 13.2.1), as well as the more general framework of light fields and Lumigraphs (Section 13.3) and higher-order fields such as environment mattes (Section 13.4). Applications of these techniques include navigating 3D collections of photographs using *photo tourism* and viewing 3D models as *object movies*.

In Chapter 13, we also discuss video-based rendering, which is the temporal extension of image-based rendering. The topics we cover include video-based animation (Section 13.5.1), periodic video turned into *video textures* (Section 13.5.2), and 3D video constructed from multiple video streams (Section 13.5.4). Applications of these techniques include video denoising, morphing, and tours based on 360° video.

Chapter 14 describes different approaches to recognition. It begins with techniques for detecting and recognizing faces (Sections 14.1 and 14.2), then looks at techniques for finding and recognizing particular objects (*instance recognition*) in Section 14.3. Next, we cover the most difficult variant of recognition, namely the recognition of broad *categories*, such as cars, motorcycles, horses and other animals (Section 14.4), and the role that scene context plays in recognition (Section 14.5).

To support the book's use as a textbook, the appendices and associated Web site contain more detailed mathematical topics and additional material. Appendix A covers linear algebra and numerical techniques, including matrix algebra, least squares, and iterative techniques. Appendix B covers Bayesian estimation theory, including maximum likelihood estimation, robust statistics, Markov random fields, and uncertainty modeling. Appendix C describes the supplementary material available to complement this book, including images and data sets, pointers to software, course slides, and an on-line bibliography.

1.4 Sample syllabus

Teaching all of the material covered in this book in a single quarter or semester course is a Herculean task and likely one not worth attempting. It is better to simply pick and choose topics related to the lecturer's preferred emphasis and tailored to the set of mini-projects envisioned for the students.

Steve Seitz and I have successfully used a 10-week syllabus similar to the one shown in Table 1.1 (omitting the parenthesized weeks) as both an undergraduate and a graduate-level course in computer vision. The undergraduate course[10] tends to go lighter on the mathematics and takes more time reviewing basics, while the graduate-level course[11] dives more deeply into techniques and assumes the students already have a decent grounding in either vision

[10] http://www.cs.washington.edu/education/courses/455/
[11] http://www.cs.washington.edu/education/courses/576/

Week	Material	Project
(1.)	Chapter 2 Image formation	
2.	Chapter 3 Image processing	
3.	Chapter 4 Feature detection and matching	P1
4.	Chapter 6 Feature-based alignment	
5.	Chapter 9 Image stitching	P2
6.	Chapter 8 Dense motion estimation	
7.	Chapter 7 Structure from motion	PP
8.	Chapter 14 Recognition	
(9.)	Chapter 10 Computational photography	
10.	Chapter 11 Stereo correspondence	
(11.)	Chapter 12 3D reconstruction	
12.	Chapter 13 Image-based rendering	
13.	Final project presentations	FP

Table 1.1 Sample syllabi for 10-week and 13-week courses. The weeks in parentheses are not used in the shorter version. P1 and P2 are two early-term mini-projects, PP is when the (student-selected) final project proposals are due, and FP is the final project presentations.

or related mathematical techniques. (See also the *Introduction to Computer Vision* course at Stanford,[12] which uses a similar curriculum.) Related courses have also been taught on the topics of 3D photography[13] and computational photography.[14]

When Steve and I teach the course, we prefer to give the students several small programming projects early in the course rather than focusing on written homework or quizzes. With a suitable choice of topics, it is possible for these projects to build on each other. For example, introducing feature matching early on can be used in a second assignment to do image alignment and stitching. Alternatively, direct (optical flow) techniques can be used to do the alignment and more focus can be put on either graph cut seam selection or multi-resolution blending techniques.

We also ask the students to propose a final project (we provide a set of suggested topics for those who need ideas) by the middle of the course and reserve the last week of the class for student presentations. With any luck, some of these final projects can actually turn into conference submissions!

No matter how you decide to structure the course or how you choose to use this book, I encourage you to try at least a few small programming tasks to get a good feel for how vision techniques work, and when they do not. Better yet, pick topics that are fun and can be used on your own photographs, and try to push your creative boundaries to come up with surprising results.

[12] http://vision.stanford.edu/teaching/cs223b/
[13] http://www.cs.washington.edu/education/courses/558/06sp/
[14] http://graphics.cs.cmu.edu/courses/15-463/

1.5 A note on notation

For better or worse, the notation found in computer vision and multi-view geometry textbooks tends to vary all over the map (Faugeras 1993; Hartley and Zisserman 2004; Girod, Greiner, and Niemann 2000; Faugeras and Luong 2001; Forsyth and Ponce 2003). In this book, I use the convention I first learned in my high school physics class (and later multi-variate calculus and computer graphics courses), which is that vectors $\boldsymbol{v}$ are lower case bold, matrices $\boldsymbol{M}$ are upper case bold, and scalars (T, s) are mixed case italic. Unless otherwise noted, vectors operate as column vectors, i.e., they post-multiply matrices, $\boldsymbol{Mv}$, although they are sometimes written as comma-separated parenthesized lists $\boldsymbol{x} = (x, y)$ instead of bracketed column vectors $\boldsymbol{x} = [x\ y]^T$. Some commonly used matrices are $\boldsymbol{R}$ for rotations, $\boldsymbol{K}$ for calibration matrices, and $\boldsymbol{I}$ for the identity matrix. Homogeneous coordinates (Section 2.1) are denoted with a tilde over the vector, e.g., $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\bar{\boldsymbol{x}}$ in $\mathcal{P}^2$. The cross product operator in matrix form is denoted by $[\,]_\times$.

1.6 Additional reading

This book attempts to be self-contained, so that students can implement the basic assignments and algorithms described here without the need for outside references. However, it does presuppose a general familiarity with basic concepts in linear algebra and numerical techniques, which are reviewed in Appendix A, and image processing, which is reviewed in Chapter 3.

Students who want to delve more deeply into these topics can look in (Golub and Van Loan 1996) for matrix algebra and (Strang 1988) for linear algebra. In image processing, there are a number of popular textbooks, including (Crane 1997; Gomes and Velho 1997; Jähne 1997; Pratt 2007; Russ 2007; Burger and Burge 2008; Gonzales and Woods 2008). For computer graphics, popular texts include (Foley, van Dam, Feiner *et al.* 1995; Watt 1995), with (Glassner 1995) providing a more in-depth look at image formation and rendering. For statistics and machine learning, Chris Bishop's (2006) book is a wonderful and comprehensive introduction with a wealth of exercises. Students may also want to look in other textbooks on computer vision for material that we do not cover here, as well as for additional project ideas (Ballard and Brown 1982; Faugeras 1993; Nalwa 1993; Trucco and Verri 1998; Forsyth and Ponce 2003).

There is, however, no substitute for reading the latest research literature, both for the latest ideas and techniques and for the most up-to-date references to related literature.[15] In this book, I have attempted to cite the most recent work in each field so that students can read them directly and use them as inspiration for their own work. Browsing the last few years' conference proceedings from the major vision and graphics conferences, such as CVPR, ECCV, ICCV, and SIGGRAPH, will provide a wealth of new ideas. The tutorials offered at these conferences, for which slides or notes are often available on-line, are also an invaluable resource.

[15] For a comprehensive bibliography and taxonomy of computer vision research, Keith Price's Annotated Computer Vision Bibliography http://www.visionbib.com/bibliography/contents.html is an invaluable resource.

Chapter 2

Image formation

2.1 Geometric primitives and transformations 29
2.1.1 Geometric primitives 29
2.1.2 2D transformations 33
2.1.3 3D transformations 36
2.1.4 3D rotations 37
2.1.5 3D to 2D projections 42
2.1.6 Lens distortions 52
2.2 Photometric image formation 54
2.2.1 Lighting 54
2.2.2 Reflectance and shading 55
2.2.3 Optics 61
2.3 The digital camera 65
2.3.1 Sampling and aliasing 69
2.3.2 Color 71
2.3.3 Compression 80
2.4 Additional reading 82
2.5 Exercises 82

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_2, © Springer-Verlag London Limited 2011

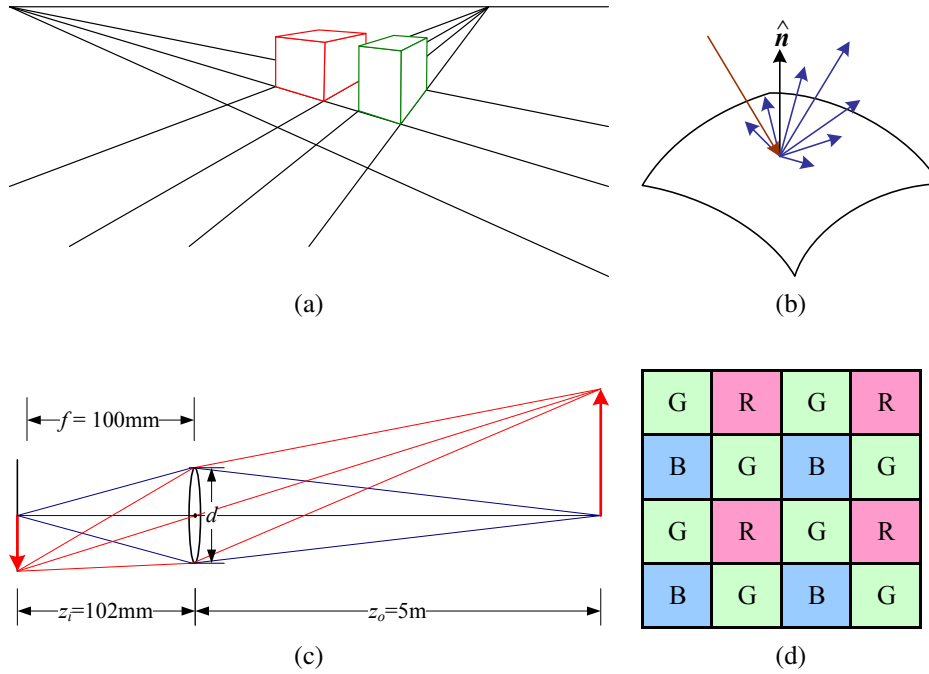


Figure 2.1 A few components of the image formation process: (a) perspective projection; (b) light scattering when hitting a surface; (c) lens optics; (d) Bayer color filter array.

Before we can intelligently analyze and manipulate images, we need to establish a vocabulary for describing the geometry of a scene. We also need to understand the image formation process that produced a particular image given a set of lighting conditions, scene geometry, surface properties, and camera optics. In this chapter, we present a simplified model of such an image formation process.

Section 2.1 introduces the basic geometric primitives used throughout the book (points, lines, and planes) and the *geometric* transformations that project these 3D quantities into 2D image features (Figure 2.1a). Section 2.2 describes how lighting, surface properties (Figure 2.1b), and camera *optics* (Figure 2.1c) interact in order to produce the color values that fall onto the image sensor. Section 2.3 describes how continuous color images are turned into discrete digital *samples* inside the image sensor (Figure 2.1d) and how to avoid (or at least characterize) sampling deficiencies, such as aliasing.

The material covered in this chapter is but a brief summary of a very rich and deep set of topics, traditionally covered in a number of separate fields. A more thorough introduction to the geometry of points, lines, planes, and projections can be found in textbooks on multi-view geometry (Hartley and Zisserman 2004; Faugeras and Luong 2001) and computer graphics (Foley, van Dam, Feiner *et al.* 1995). The image formation (synthesis) process is traditionally taught as part of a computer graphics curriculum (Foley, van Dam, Feiner *et al.* 1995; Glassner 1995; Watt 1995; Shirley 2005) but it is also studied in physics-based computer vision (Wolff, Shafer, and Healey 1992a). The behavior of camera lens systems is studied in optics (Möller 1988; Hecht 2001; Ray 2002). Two good books on color theory are (Wyszecki and Stiles 2000; Healey and Shafer 1992), with (Livingstone 2008) providing a more fun and informal introduction to the topic of color perception. Topics relating to sampling and aliasing are covered in textbooks on signal and image processing (Crane 1997; Jähne 1997; Oppenheim and Schafer 1996; Oppenheim, Schafer, and Buck 1999; Pratt 2007; Russ 2007; Burger and Burge 2008; Gonzales and Woods 2008).

A note to students: If you have already studied computer graphics, you may want to skim the material in Section 2.1, although the sections on projective depth and object-centered projection near the end of Section 2.1.5 may be new to you. Similarly, physics students (as well as computer graphics students) will mostly be familiar with Section 2.2. Finally, students with a good background in image processing will already be familiar with sampling issues (Section 2.3) as well as some of the material in Chapter 3.

2.1 Geometric primitives and transformations

In this section, we introduce the basic 2D and 3D primitives used in this textbook, namely points, lines, and planes. We also describe how 3D features are projected into 2D features. More detailed descriptions of these topics (along with a gentler and more intuitive introduction) can be found in textbooks on multiple-view geometry (Hartley and Zisserman 2004; Faugeras and Luong 2001).

2.1.1 Geometric primitives

Geometric primitives form the basic building blocks used to describe three-dimensional shapes. In this section, we introduce points, lines, and planes. Later sections of the book discuss

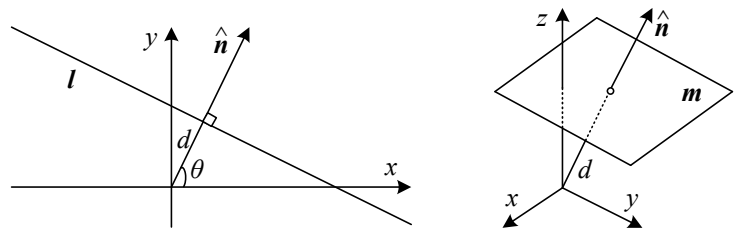


Figure 2.2 (a) 2D line equation and (b) 3D plane equation, expressed in terms of the normal $\hat{\boldsymbol{n}}$ and distance to the origin d.

curves (Sections 5.1 and 11.2), surfaces (Section 12.3), and volumes (Section 12.5).

2D points. 2D points (pixel coordinates in an image) can be denoted using a pair of values, $\boldsymbol{x} = (x, y) \in \mathcal{R}^2$, or alternatively,

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}. \tag{2.1}$$

(As stated in the introduction, we use the $(x_1, x_2, \ldots)$ notation to denote column vectors.)

2D points can also be represented using *homogeneous coordinates*, $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) \in \mathcal{P}^2$, where vectors that differ only by scale are considered to be equivalent. $\mathcal{P}^2 = \mathcal{R}^3 - (0, 0, 0)$ is called the 2D *projective space*.

A homogeneous vector $\tilde{\boldsymbol{x}}$ can be converted back into an *inhomogeneous* vector $\boldsymbol{x}$ by dividing through by the last element $\tilde{w}$, i.e.,

$$\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\bar{\boldsymbol{x}}, \tag{2.2}$$

where $\bar{\boldsymbol{x}} = (x, y, 1)$ is the *augmented vector*. Homogeneous points whose last element is $\tilde{w} = 0$ are called *ideal points* or *points at infinity* and do not have an equivalent inhomogeneous representation.

2D lines. 2D lines can also be represented using homogeneous coordinates $\tilde{\boldsymbol{l}} = (a, b, c)$. The corresponding *line equation* is

$$\bar{\boldsymbol{x}} \cdot \tilde{\boldsymbol{l}} = ax + by + c = 0. \tag{2.3}$$

We can normalize the line equation vector so that $\boldsymbol{l} = (\hat{n}_x, \hat{n}_y, d) = (\hat{\boldsymbol{n}}, d)$ with $\|\hat{\boldsymbol{n}}\| = 1$. In this case, $\hat{\boldsymbol{n}}$ is the *normal vector* perpendicular to the line and d is its distance to the origin (Figure 2.2). (The one exception to this normalization is the *line at infinity* $\tilde{\boldsymbol{l}} = (0, 0, 1)$, which includes all (ideal) points at infinity.)

We can also express $\hat{\boldsymbol{n}}$ as a function of rotation angle θ, $\hat{\boldsymbol{n}} = (\hat{n}_x, \hat{n}_y) = (\cos\theta, \sin\theta)$ (Figure 2.2a). This representation is commonly used in the *Hough transform* line-finding algorithm, which is discussed in Section 4.3.2. The combination (θ, d) is also known as *polar coordinates*.

When using homogeneous coordinates, we can compute the intersection of two lines as

$$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{l}}_1 \times \tilde{\boldsymbol{l}}_2, \tag{2.4}$$

where $\times$ is the cross product operator. Similarly, the line joining two points can be written as

$$\tilde{\boldsymbol{l}} = \tilde{\boldsymbol{x}}_1 \times \tilde{\boldsymbol{x}}_2. \tag{2.5}$$

When trying to fit an intersection point to multiple lines or, conversely, a line to multiple points, least squares techniques (Section 6.1.1 and Appendix A.2) can be used, as discussed in Exercise 2.1.

2D conics. There are other algebraic curves that can be expressed with simple polynomial homogeneous equations. For example, the *conic sections* (so called because they arise as the intersection of a plane and a 3D cone) can be written using a *quadric* equation

$$\tilde{\boldsymbol{x}}^T \boldsymbol{Q} \tilde{\boldsymbol{x}} = 0. \tag{2.6}$$

Quadric equations play useful roles in the study of multi-view geometry and camera calibration (Hartley and Zisserman 2004; Faugeras and Luong 2001) but are not used extensively in this book.

3D points. Point coordinates in three dimensions can be written using inhomogeneous coordinates $\boldsymbol{x} = (x, y, z) \in \mathcal{R}^3$ or homogeneous coordinates $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w}) \in \mathcal{P}^3$. As before, it is sometimes useful to denote a 3D point using the augmented vector $\bar{\boldsymbol{x}} = (x, y, z, 1)$ with $\tilde{\boldsymbol{x}} = \tilde{w}\bar{\boldsymbol{x}}$.

3D planes. 3D planes can also be represented as homogeneous coordinates $\tilde{\boldsymbol{m}} = (a, b, c, d)$ with a corresponding plane equation

$$\bar{\boldsymbol{x}} \cdot \tilde{\boldsymbol{m}} = ax + by + cz + d = 0. \tag{2.7}$$

We can also normalize the plane equation as $\boldsymbol{m} = (\hat{n}_x, \hat{n}_y, \hat{n}_z, d) = (\hat{\boldsymbol{n}}, d)$ with $\|\hat{\boldsymbol{n}}\| = 1$. In this case, $\hat{\boldsymbol{n}}$ is the *normal vector* perpendicular to the plane and d is its distance to the origin (Figure 2.2b). As with the case of 2D lines, the *plane at infinity* $\tilde{\boldsymbol{m}} = (0, 0, 0, 1)$, which contains all the points at infinity, cannot be normalized (i.e., it does not have a unique normal or a finite distance).

We can express $\hat{\boldsymbol{n}}$ as a function of two angles (θ, ϕ),

$$\hat{\boldsymbol{n}} = (\cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi), \tag{2.8}$$

i.e., using *spherical coordinates*, but these are less commonly used than polar coordinates since they do not uniformly sample the space of possible normal vectors.

3D lines. Lines in 3D are less elegant than either lines in 2D or planes in 3D. One possible representation is to use two points on the line, $(\boldsymbol{p}, \boldsymbol{q})$. Any other point on the line can be expressed as a linear combination of these two points

$$\boldsymbol{r} = (1 - \lambda)\boldsymbol{p} + \lambda\boldsymbol{q}, \tag{2.9}$$

as shown in Figure 2.3. If we restrict $0 \leq \lambda \leq 1$, we get the *line segment* joining $\boldsymbol{p}$ and $\boldsymbol{q}$.

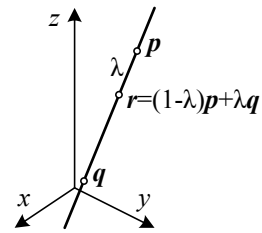


Figure 2.3 3D line equation, $\boldsymbol{r} = (1-\lambda)\boldsymbol{p} + \lambda\boldsymbol{q}$.

If we use homogeneous coordinates, we can write the line as

$$\tilde{\boldsymbol{r}} = \mu\tilde{\boldsymbol{p}} + \lambda\tilde{\boldsymbol{q}}. \tag{2.10}$$

A special case of this is when the second point is at infinity, i.e., $\tilde{\boldsymbol{q}} = (\hat{d}_x, \hat{d}_y, \hat{d}_z, 0) = (\hat{\boldsymbol{d}}, 0)$. Here, we see that $\hat{\boldsymbol{d}}$ is the *direction* of the line. We can then re-write the inhomogeneous 3D line equation as

$$\boldsymbol{r} = \boldsymbol{p} + \lambda\hat{\boldsymbol{d}}. \tag{2.11}$$

A disadvantage of the endpoint representation for 3D lines is that it has too many degrees of freedom, i.e., six (three for each endpoint) instead of the four degrees that a 3D line truly has. However, if we fix the two points on the line to lie in specific planes, we obtain a representation with four degrees of freedom. For example, if we are representing nearly vertical lines, then $z = 0$ and $z = 1$ form two suitable planes, i.e., the (x, y) coordinates in both planes provide the four coordinates describing the line. This kind of two-plane parameterization is used in the *light field* and *Lumigraph* image-based rendering systems described in Chapter 13 to represent the collection of rays seen by a camera as it moves in front of an object. The two-endpoint representation is also useful for representing line segments, even when their exact endpoints cannot be seen (only guessed at).

If we wish to represent all possible lines without bias towards any particular orientation, we can use *Plücker coordinates* (Hartley and Zisserman 2004, Chapter 2; Faugeras and Luong 2001, Chapter 3). These coordinates are the six independent non-zero entries in the 4×4 skew symmetric matrix

$$\boldsymbol{L} = \tilde{\boldsymbol{p}}\tilde{\boldsymbol{q}}^T - \tilde{\boldsymbol{q}}\tilde{\boldsymbol{p}}^T, \tag{2.12}$$

where $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{q}}$ are *any* two (non-identical) points on the line. This representation has only four degrees of freedom, since $\boldsymbol{L}$ is homogeneous and also satisfies $det(\boldsymbol{L}) = 0$, which results in a quadratic constraint on the Plücker coordinates.

In practice, the minimal representation is not essential for most applications. An adequate model of 3D lines can be obtained by estimating their direction (which may be known ahead of time, e.g., for architecture) and some point within the visible portion of the line (see Section 7.5.1) or by using the two endpoints, since lines are most often visible as finite line segments. However, if you are interested in more details about the topic of minimal line parameterizations, Förstner (2005) discusses various ways to infer and model 3D lines in projective geometry, as well as how to estimate the uncertainty in such fitted models.

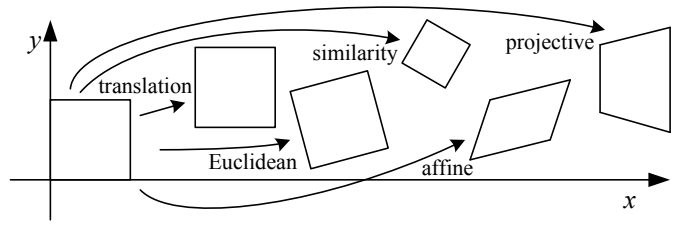


Figure 2.4 Basic set of 2D planar transformations.

3D quadrics. The 3D analog of a conic section is a quadric surface

$$\bar{\boldsymbol{x}}^T \boldsymbol{Q} \bar{\boldsymbol{x}} = 0 \tag{2.13}$$

(Hartley and Zisserman 2004, Chapter 2). Again, while quadric surfaces are useful in the study of multi-view geometry and can also serve as useful modeling primitives (spheres, ellipsoids, cylinders), we do not study them in great detail in this book.

2.1.2 2D transformations

Having defined our basic primitives, we can now turn our attention to how they can be transformed. The simplest transformations occur in the 2D plane and are illustrated in Figure 2.4.

Translation. 2D translations can be written as $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}$ or

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}} \tag{2.14}$$

where $\boldsymbol{I}$ is the (2×2) identity matrix or

$$\bar{\boldsymbol{x}}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \bar{\boldsymbol{x}} \tag{2.15}$$

where $\boldsymbol{0}$ is the zero vector. Using a 2×3 matrix results in a more compact notation, whereas using a full-rank 3×3 matrix (which can be obtained from the 2×3 matrix by appending a $[\boldsymbol{0}^T\ 1]$ row) makes it possible to chain transformations using matrix multiplication. Note that in any equation where an augmented vector such as $\bar{\boldsymbol{x}}$ appears on both sides, it can always be replaced with a full homogeneous vector $\tilde{\boldsymbol{x}}$.

Rotation + translation. This transformation is also known as *2D rigid body motion* or the *2D Euclidean transformation* (since Euclidean distances are preserved). It can be written as $\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}$ or

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}} \tag{2.16}$$

where

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.17}$$

is an orthonormal rotation matrix with $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$ and $|\boldsymbol{R}| = 1$.

Scaled rotation. Also known as the *similarity transform*, this transformation can be expressed as $\boldsymbol{x}' = s\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}$ where s is an arbitrary scale factor. It can also be written as

$$\boldsymbol{x}' = \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix} \bar{\boldsymbol{x}}, \tag{2.18}$$

where we no longer require that $a^2 + b^2 = 1$. The similarity transform preserves angles between lines.

Affine. The affine transformation is written as $\boldsymbol{x}' = \boldsymbol{A}\bar{\boldsymbol{x}}$, where $\boldsymbol{A}$ is an arbitrary 2×3 matrix, i.e.,

$$\boldsymbol{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \bar{\boldsymbol{x}}. \tag{2.19}$$

Parallel lines remain parallel under affine transformations.

Projective. This transformation, also known as a *perspective transform* or *homography*, operates on homogeneous coordinates,

$$\tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{x}}, \tag{2.20}$$

where $\tilde{\boldsymbol{H}}$ is an arbitrary 3×3 matrix. Note that $\tilde{\boldsymbol{H}}$ is homogeneous, i.e., it is only defined up to a scale, and that two $\tilde{\boldsymbol{H}}$ matrices that differ only by scale are equivalent. The resulting homogeneous coordinate $\tilde{\boldsymbol{x}}'$ must be normalized in order to obtain an inhomogeneous result $\boldsymbol{x}$, i.e.,

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \text{ and } y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}. \tag{2.21}$$

Perspective transformations preserve straight lines (i.e., they remain straight after the transformation).

Hierarchy of 2D transformations. The preceding set of transformations are illustrated in Figure 2.4 and summarized in Table 2.1. The easiest way to think of them is as a set of (potentially restricted) 3×3 matrices operating on 2D homogeneous coordinate vectors. Hartley and Zisserman (2004) contains a more detailed description of the hierarchy of 2D planar transformations.

The above transformations form a nested set of *groups*, i.e., they are closed under composition and have an inverse that is a member of the same group. (This will be important later when applying these transformations to images in Section 3.6.) Each (simpler) group is a subset of the more complex group below it.

Co-vectors. While the above transformations can be used to transform points in a 2D plane, can they also be used directly to transform a line equation? Consider the homogeneous equation $\tilde{\boldsymbol{l}} \cdot \tilde{\boldsymbol{x}} = 0$. If we transform $\boldsymbol{x}' = \tilde{\boldsymbol{H}}\boldsymbol{x}$, we obtain

$$\tilde{\boldsymbol{l}}' \cdot \tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{l}}'^T \tilde{\boldsymbol{H}}\tilde{\boldsymbol{x}} = (\tilde{\boldsymbol{H}}^T \tilde{\boldsymbol{l}}')^T \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{l}} \cdot \tilde{\boldsymbol{x}} = 0, \tag{2.22}$$

i.e., $\tilde{\boldsymbol{l}}' = \tilde{\boldsymbol{H}}^{-T}\tilde{\boldsymbol{l}}$. Thus, the action of a projective transformation on a *co-vector* such as a 2D line or 3D normal can be represented by the transposed inverse of the matrix, which is equivalent to the *adjoint* of $\tilde{\boldsymbol{H}}$, since projective transformation matrices are homogeneous. Jim

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

Table 2.1 Hierarchy of 2D coordinate transformations. Each transformation also preserves the properties listed in the rows below it, i.e., similarity preserves not only angles but also parallelism and straight lines. The 2×3 matrices are extended with a third $[\mathbf{0}^T \; 1]$ row to form a full 3×3 matrix for homogeneous coordinate transformations.

Blinn (1998) describes (in Chapters 9 and 10) the ins and outs of notating and manipulating co-vectors.

While the above transformations are the ones we use most extensively, a number of additional transformations are sometimes used.

Stretch/squash. This transformation changes the aspect ratio of an image,

$$\begin{aligned} x' &= s_x x + t_x \\ y' &= s_y y + t_y, \end{aligned}$$

and is a restricted form of an affine transformation. Unfortunately, it does not nest cleanly with the groups listed in Table 2.1.

Planar surface flow. This eight-parameter transformation (Horn 1986; Bergen, Anandan, Hanna *et al.* 1992; Girod, Greiner, and Niemann 2000),

$$\begin{aligned} x' &= a_0 + a_1 x + a_2 y + a_6 x^2 + a_7 xy \\ y' &= a_3 + a_4 x + a_5 y + a_7 x^2 + a_6 xy, \end{aligned}$$

arises when a planar surface undergoes a small 3D motion. It can thus be thought of as a small motion approximation to a full homography. Its main attraction is that it is *linear* in the motion parameters, a_k, which are often the quantities being estimated.

Bilinear interpolant. This eight-parameter transform (Wolberg 1990),

$$\begin{aligned} x' &= a_0 + a_1 x + a_2 y + a_6 xy \\ y' &= a_3 + a_4 x + a_5 y + a_7 xy, \end{aligned}$$

can be used to interpolate the deformation due to the motion of the four corner points of a square. (In fact, it can interpolate the motion of any four non-collinear points.) While

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{3\times4}$	3	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{3\times4}$	6	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{3\times4}$	7	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{3\times4}$	12	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{4\times4}$	15	straight lines	

Table 2.2 Hierarchy of 3D coordinate transformations. Each transformation also preserves the properties listed in the rows below it, i.e., similarity preserves not only angles but also parallelism and straight lines. The 3×4 matrices are extended with a fourth $[\mathbf{0}^T\ 1]$ row to form a full 4×4 matrix for homogeneous coordinate transformations. The mnemonic icons are drawn in 2D but are meant to suggest transformations occurring in a full 3D cube.

the deformation is linear in the motion parameters, it does not generally preserve straight lines (only lines parallel to the square axes). However, it is often quite useful, e.g., in the interpolation of sparse grids using splines (Section 8.3).

2.1.3 3D transformations

The set of three-dimensional coordinate transformations is very similar to that available for 2D transformations and is summarized in Table 2.2. As in 2D, these transformations form a nested set of groups. Hartley and Zisserman (2004, Section 2.4) give a more detailed description of this hierarchy.

Translation. 3D translations can be written as $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}$ or

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}} \tag{2.23}$$

where $\boldsymbol{I}$ is the (3×3) identity matrix and $\mathbf{0}$ is the zero vector.

Rotation + translation. Also known as 3D *rigid body motion* or the 3D *Euclidean transformation*, it can be written as $\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}$ or

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}} \tag{2.24}$$

where $\boldsymbol{R}$ is a 3×3 orthonormal rotation matrix with $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$ and $|\boldsymbol{R}| = 1$. Note that sometimes it is more convenient to describe a rigid motion using

$$\boldsymbol{x}' = \boldsymbol{R}(\boldsymbol{x} - \boldsymbol{c}) = \boldsymbol{R}\boldsymbol{x} - \boldsymbol{R}\boldsymbol{c}, \tag{2.25}$$

where $\boldsymbol{c}$ is the center of rotation (often the camera center).

Compactly parameterizing a 3D rotation is a non-trivial task, which we describe in more detail below.

Scaled rotation. The 3D *similarity transform* can be expressed as $\boldsymbol{x}' = s\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}$ where s is an arbitrary scale factor. It can also be written as

$$\boldsymbol{x}' = \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}}. \tag{2.26}$$

This transformation preserves angles between lines and planes.

Affine. The affine transform is written as $\boldsymbol{x}' = \boldsymbol{A}\bar{\boldsymbol{x}}$, where $\boldsymbol{A}$ is an arbitrary 3×4 matrix, i.e.,

$$\boldsymbol{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{bmatrix} \bar{\boldsymbol{x}}. \tag{2.27}$$

Parallel lines and planes remain parallel under affine transformations.

Projective. This transformation, variously known as a *3D perspective transform*, *homography*, or *collineation*, operates on homogeneous coordinates,

$$\tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{x}}, \tag{2.28}$$

where $\tilde{\boldsymbol{H}}$ is an arbitrary 4×4 homogeneous matrix. As in 2D, the resulting homogeneous coordinate $\tilde{\boldsymbol{x}}'$ must be normalized in order to obtain an inhomogeneous result $\boldsymbol{x}$. Perspective transformations preserve straight lines (i.e., they remain straight after the transformation).

2.1.4 3D rotations

The biggest difference between 2D and 3D coordinate transformations is that the parameterization of the 3D rotation matrix $\boldsymbol{R}$ is not as straightforward but several possibilities exist.

Euler angles

A rotation matrix can be formed as the product of three rotations around three cardinal axes, e.g., x, y, and z, or x, y, and x. This is generally a bad idea, as the result depends on the order in which the transforms are applied. What is worse, it is not always possible to move smoothly in the parameter space, i.e., sometimes one or more of the Euler angles change dramatically in response to a small change in rotation.[1] For these reasons, we do not even give the formula for Euler angles in this book—interested readers can look in other textbooks or technical reports (Faugeras 1993; Diebel 2006). Note that, in some applications, if the rotations are known to be a set of uni-axial transforms, they can always be represented using an explicit set of rigid transformations.

Axis/angle (exponential twist)

A rotation can be represented by a rotation axis $\hat{\boldsymbol{n}}$ and an angle θ, or equivalently by a 3D vector $\boldsymbol{\omega} = \theta\hat{\boldsymbol{n}}$. Figure 2.5 shows how we can compute the equivalent rotation. First, we project the vector $\boldsymbol{v}$ onto the axis $\hat{\boldsymbol{n}}$ to obtain

$$\boldsymbol{v}_{\parallel} = \hat{\boldsymbol{n}}(\hat{\boldsymbol{n}} \cdot \boldsymbol{v}) = (\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T)\boldsymbol{v}, \tag{2.29}$$

[1] In robotics, this is sometimes referred to as *gimbal lock*.

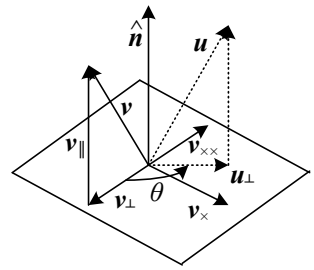


Figure 2.5 Rotation around an axis $\hat{\boldsymbol{n}}$ by an angle θ.

which is the component of $\boldsymbol{v}$ that is not affected by the rotation. Next, we compute the perpendicular residual of $\boldsymbol{v}$ from $\hat{\boldsymbol{n}}$,

$$\boldsymbol{v}_\perp = \boldsymbol{v} - \boldsymbol{v}_\parallel = (\boldsymbol{I} - \hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T)\boldsymbol{v}. \tag{2.30}$$

We can rotate this vector by 90° using the cross product,

$$\boldsymbol{v}_\times = \hat{\boldsymbol{n}} \times \boldsymbol{v} = [\hat{\boldsymbol{n}}]_\times \boldsymbol{v}, \tag{2.31}$$

where $[\hat{\boldsymbol{n}}]_\times$ is the matrix form of the cross product operator with the vector $\hat{\boldsymbol{n}} = (\hat{n}_x, \hat{n}_y, \hat{n}_z)$,

$$[\hat{\boldsymbol{n}}]_\times = \begin{bmatrix} 0 & -\hat{n}_z & \hat{n}_y \\ \hat{n}_z & 0 & -\hat{n}_x \\ -\hat{n}_y & \hat{n}_x & 0 \end{bmatrix}. \tag{2.32}$$

Note that rotating this vector by another 90° is equivalent to taking the cross product again,

$$\boldsymbol{v}_{\times\times} = \hat{\boldsymbol{n}} \times \boldsymbol{v}_\times = [\hat{\boldsymbol{n}}]_\times^2 \boldsymbol{v} = -\boldsymbol{v}_\perp,$$

and hence

$$\boldsymbol{v}_\parallel = \boldsymbol{v} - \boldsymbol{v}_\perp = \boldsymbol{v} + \boldsymbol{v}_{\times\times} = (\boldsymbol{I} + [\hat{\boldsymbol{n}}]_\times^2)\boldsymbol{v}.$$

We can now compute the in-plane component of the rotated vector $\boldsymbol{u}$ as

$$\boldsymbol{u}_\perp = \cos\theta \boldsymbol{v}_\perp + \sin\theta \boldsymbol{v}_\times = (\sin\theta [\hat{\boldsymbol{n}}]_\times - \cos\theta [\hat{\boldsymbol{n}}]_\times^2)\boldsymbol{v}.$$

Putting all these terms together, we obtain the final rotated vector as

$$\boldsymbol{u} = \boldsymbol{u}_\perp + \boldsymbol{v}_\parallel = (\boldsymbol{I} + \sin\theta [\hat{\boldsymbol{n}}]_\times + (1 - \cos\theta)[\hat{\boldsymbol{n}}]_\times^2)\boldsymbol{v}. \tag{2.33}$$

We can therefore write the rotation matrix corresponding to a rotation by θ around an axis $\hat{\boldsymbol{n}}$ as

$$\boldsymbol{R}(\hat{\boldsymbol{n}}, \theta) = \boldsymbol{I} + \sin\theta [\hat{\boldsymbol{n}}]_\times + (1 - \cos\theta)[\hat{\boldsymbol{n}}]_\times^2, \tag{2.34}$$

which is known as *Rodriguez's formula* (Ayache 1989).

The product of the axis $\hat{\boldsymbol{n}}$ and angle θ, $\boldsymbol{\omega} = \theta\hat{\boldsymbol{n}} = (\omega_x, \omega_y, \omega_z)$, is a minimal representation for a 3D rotation. Rotations through common angles such as multiples of 90° can be represented exactly (and converted to exact matrices) if θ is stored in degrees. Unfortunately,

this representation is not unique, since we can always add a multiple of 360° (2π radians) to θ and get the same rotation matrix. As well, $(\hat{\boldsymbol{n}}, \theta)$ and $(-\hat{\boldsymbol{n}}, -\theta)$ represent the same rotation.

However, for small rotations (e.g., corrections to rotations), this is an excellent choice. In particular, for small (infinitesimal or instantaneous) rotations and θ expressed in radians, Rodriguez's formula simplifies to

$$\boldsymbol{R}(\boldsymbol{\omega}) \approx \boldsymbol{I} + \sin\theta[\hat{\boldsymbol{n}}]_\times \approx \boldsymbol{I} + [\theta\hat{\boldsymbol{n}}]_\times = \begin{bmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & -\omega_x \\ -\omega_y & \omega_x & 1 \end{bmatrix}, \tag{2.35}$$

which gives a nice linearized relationship between the rotation parameters $\boldsymbol{\omega}$ and $\boldsymbol{R}$. We can also write $\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{v} \approx \boldsymbol{v} + \boldsymbol{\omega} \times \boldsymbol{v}$, which is handy when we want to compute the derivative of $\boldsymbol{Rv}$ with respect to $\boldsymbol{\omega}$,

$$\frac{\partial \boldsymbol{Rv}}{\partial \boldsymbol{\omega}^T} = -[\boldsymbol{v}]_\times = \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix}. \tag{2.36}$$

Another way to derive a rotation through a finite angle is called the *exponential twist* (Murray, Li, and Sastry 1994). A rotation by an angle θ is equivalent to k rotations through θ/k. In the limit as $k \to \infty$, we obtain

$$\boldsymbol{R}(\hat{\boldsymbol{n}}, \theta) = \lim_{k\to\infty} (\boldsymbol{I} + \frac{1}{k}[\theta\hat{\boldsymbol{n}}]_\times)^k = \exp[\boldsymbol{\omega}]_\times. \tag{2.37}$$

If we expand the matrix exponential as a Taylor series (using the identity $[\hat{\boldsymbol{n}}]_\times^{k+2} = -[\hat{\boldsymbol{n}}]_\times^k$, $k > 0$, and again assuming θ is in radians),

$$\begin{aligned} \exp[\boldsymbol{\omega}]_\times &= \boldsymbol{I} + \theta[\hat{\boldsymbol{n}}]_\times + \frac{\theta^2}{2}[\hat{\boldsymbol{n}}]_\times^2 + \frac{\theta^3}{3!}[\hat{\boldsymbol{n}}]_\times^3 + \cdots \\ &= \boldsymbol{I} + (\theta - \frac{\theta^3}{3!} + \cdots)[\hat{\boldsymbol{n}}]_\times + (\frac{\theta^2}{2} - \frac{\theta^3}{4!} + \cdots)[\hat{\boldsymbol{n}}]_\times^2 \\ &= \boldsymbol{I} + \sin\theta[\hat{\boldsymbol{n}}]_\times + (1 - \cos\theta)[\hat{\boldsymbol{n}}]_\times^2, \end{aligned} \tag{2.38}$$

which yields the familiar Rodriguez's formula.

Unit quaternions

The unit quaternion representation is closely related to the angle/axis representation. A unit quaternion is a unit length 4-vector whose components can be written as $\boldsymbol{q} = (q_x, q_y, q_z, q_w)$ or $\boldsymbol{q} = (x, y, z, w)$ for short. Unit quaternions live on the unit sphere $\|\boldsymbol{q}\| = 1$ and *antipodal* (opposite sign) quaternions, $\boldsymbol{q}$ and $-\boldsymbol{q}$, represent the same rotation (Figure 2.6). Other than this ambiguity (dual covering), the unit quaternion representation of a rotation is unique. Furthermore, the representation is *continuous*, i.e., as rotation matrices vary continuously, one can find a continuous quaternion representation, although the path on the quaternion sphere may wrap all the way around before returning to the "origin" $\boldsymbol{q}_o = (0, 0, 0, 1)$. For these and other reasons given below, quaternions are a very popular representation for pose and for pose interpolation in computer graphics (Shoemake 1985).

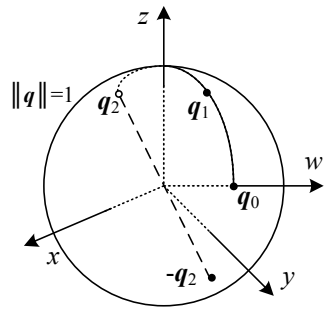


Figure 2.6 Unit quaternions live on the unit sphere $\|\boldsymbol{q}\| = 1$. This figure shows a smooth trajectory through the three quaternions $\boldsymbol{q}_0$, $\boldsymbol{q}_1$, and $\boldsymbol{q}_2$. The *antipodal* point to $\boldsymbol{q}_2$, namely $-\boldsymbol{q}_2$, represents the same rotation as $\boldsymbol{q}_2$.

Quaternions can be derived from the axis/angle representation through the formula

$$\boldsymbol{q} = (\boldsymbol{v}, w) = (\sin\frac{\theta}{2}\hat{\boldsymbol{n}}, \cos\frac{\theta}{2}), \tag{2.39}$$

where $\hat{\boldsymbol{n}}$ and θ are the rotation axis and angle. Using the trigonometric identities $\sin\theta = 2\sin\frac{\theta}{2}\cos\frac{\theta}{2}$ and $(1-\cos\theta) = 2\sin^2\frac{\theta}{2}$, Rodriguez's formula can be converted to

$$\begin{aligned}\boldsymbol{R}(\hat{\boldsymbol{n}}, \theta) &= \boldsymbol{I} + \sin\theta[\hat{\boldsymbol{n}}]_\times + (1-\cos\theta)[\hat{\boldsymbol{n}}]_\times^2 \\ &= \boldsymbol{I} + 2w[\boldsymbol{v}]_\times + 2[\boldsymbol{v}]_\times^2. \end{aligned} \tag{2.40}$$

This suggests a quick way to rotate a vector $\boldsymbol{v}$ by a quaternion using a series of cross products, scalings, and additions. To obtain a formula for $\boldsymbol{R}(\boldsymbol{q})$ as a function of (x, y, z, w), recall that

$$[\boldsymbol{v}]_\times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \text{ and } [\boldsymbol{v}]_\times^2 = \begin{bmatrix} -y^2-z^2 & xy & xz \\ xy & -x^2-z^2 & yz \\ xz & yz & -x^2-y^2 \end{bmatrix}.$$

We thus obtain

$$\boldsymbol{R}(\boldsymbol{q}) = \begin{bmatrix} 1-2(y^2+z^2) & 2(xy-zw) & 2(xz+yw) \\ 2(xy+zw) & 1-2(x^2+z^2) & 2(yz-xw) \\ 2(xz-yw) & 2(yz+xw) & 1-2(x^2+y^2) \end{bmatrix}. \tag{2.41}$$

The diagonal terms can be made more symmetrical by replacing $1-2(y^2+z^2)$ with $(x^2+w^2-y^2-z^2)$, etc.

The nicest aspect of unit quaternions is that there is a simple algebra for composing rotations expressed as unit quaternions. Given two quaternions $\boldsymbol{q}_0 = (\boldsymbol{v}_0, w_0)$ and $\boldsymbol{q}_1 = (\boldsymbol{v}_1, w_1)$, the *quaternion multiply* operator is defined as

$$\boldsymbol{q}_2 = \boldsymbol{q}_0\boldsymbol{q}_1 = (\boldsymbol{v}_0 \times \boldsymbol{v}_1 + w_0\boldsymbol{v}_1 + w_1\boldsymbol{v}_0,\ w_0w_1 - \boldsymbol{v}_0 \cdot \boldsymbol{v}_1), \tag{2.42}$$

with the property that $\boldsymbol{R}(\boldsymbol{q}_2) = \boldsymbol{R}(\boldsymbol{q}_0)\boldsymbol{R}(\boldsymbol{q}_1)$. Note that quaternion multiplication is *not* commutative, just as 3D rotations and matrix multiplications are not.

procedure *slerp*$(\boldsymbol{q}_0, \boldsymbol{q}_1, \alpha)$:

1. $\boldsymbol{q}_r = \boldsymbol{q}_1/\boldsymbol{q}_0 = (\boldsymbol{v}_r, w_r)$
2. if $w_r < 0$ then $\boldsymbol{q}_r \leftarrow -\boldsymbol{q}_r$
3. $\theta_r = 2\tan^{-1}(\|\boldsymbol{v}_r\|/w_r)$
4. $\hat{\boldsymbol{n}}_r = \mathcal{N}(\boldsymbol{v}_r) = \boldsymbol{v}_r/\|\boldsymbol{v}_r\|$
5. $\theta_\alpha = \alpha\,\theta_r$
6. $\boldsymbol{q}_\alpha = (\sin\frac{\theta_\alpha}{2}\hat{\boldsymbol{n}}_r, \cos\frac{\theta_\alpha}{2})$
7. **return** $\boldsymbol{q}_2 = \boldsymbol{q}_\alpha\boldsymbol{q}_0$

Algorithm 2.1 Spherical linear interpolation (slerp). The axis and total angle are first computed from the quaternion ratio. (This computation can be lifted outside an inner loop that generates a set of interpolated position for animation.) An incremental quaternion is then computed and multiplied by the starting rotation quaternion.

Taking the inverse of a quaternion is easy: Just flip the sign of $\boldsymbol{v}$ or w (but not both!). (You can verify this has the desired effect of transposing the $\boldsymbol{R}$ matrix in (2.41).) Thus, we can also define *quaternion division* as

$$\boldsymbol{q}_2 = \boldsymbol{q}_0/\boldsymbol{q}_1 = \boldsymbol{q}_0\boldsymbol{q}_1^{-1} = (\boldsymbol{v}_0 \times \boldsymbol{v}_1 + w_0\boldsymbol{v}_1 - w_1\boldsymbol{v}_0,\ -w_0w_1 - \boldsymbol{v}_0 \cdot \boldsymbol{v}_1). \tag{2.43}$$

This is useful when the *incremental rotation* between two rotations is desired.

In particular, if we want to determine a rotation that is partway between two given rotations, we can compute the incremental rotation, take a fraction of the angle, and compute the new rotation. This procedure is called *spherical linear interpolation* or *slerp* for short (Shoemake 1985) and is given in Algorithm 2.1. Note that Shoemake presents two formulas other than the one given here. The first exponentiates $\boldsymbol{q}_r$ by alpha before multiplying the original quaternion,

$$\boldsymbol{q}_2 = \boldsymbol{q}_r^\alpha\boldsymbol{q}_0, \tag{2.44}$$

while the second treats the quaternions as 4-vectors on a sphere and uses

$$\boldsymbol{q}_2 = \frac{\sin(1-\alpha)\theta}{\sin\theta}\boldsymbol{q}_0 + \frac{\sin\alpha\theta}{\sin\theta}\boldsymbol{q}_1, \tag{2.45}$$

where $\theta = \cos^{-1}(\boldsymbol{q}_0 \cdot \boldsymbol{q}_1)$ and the dot product is directly between the quaternion 4-vectors. All of these formulas give comparable results, although care should be taken when $\boldsymbol{q}_0$ and $\boldsymbol{q}_1$ are close together, which is why I prefer to use an arctangent to establish the rotation angle.

Which rotation representation is better?

The choice of representation for 3D rotations depends partly on the application.

The axis/angle representation is minimal, and hence does not require any additional constraints on the parameters (no need to re-normalize after each update). If the angle is expressed in degrees, it is easier to understand the pose (say, 90° twist around x-axis), and also

easier to express exact rotations. When the angle is in radians, the derivatives of $\boldsymbol{R}$ with respect to $\boldsymbol{\omega}$ can easily be computed (2.36).

Quaternions, on the other hand, are better if you want to keep track of a smoothly moving camera, since there are no discontinuities in the representation. It is also easier to interpolate between rotations and to chain rigid transformations (Murray, Li, and Sastry 1994; Bregler and Malik 1998).

My usual preference is to use quaternions, but to update their estimates using an incremental rotation, as described in Section 6.2.2.

2.1.5 3D to 2D projections

Now that we know how to represent 2D and 3D geometric primitives and how to transform them spatially, we need to specify how 3D primitives are projected onto the image plane. We can do this using a linear 3D to 2D projection matrix. The simplest model is orthography, which requires no division to get the final (inhomogeneous) result. The more commonly used model is perspective, since this more accurately models the behavior of real cameras.

Orthography and para-perspective

An orthographic projection simply drops the z component of the three-dimensional coordinate $\boldsymbol{p}$ to obtain the 2D point $\boldsymbol{x}$. (In this section, we use $\boldsymbol{p}$ to denote 3D points and $\boldsymbol{x}$ to denote 2D points.) This can be written as

$$\boldsymbol{x} = [\boldsymbol{I}_{2\times 2}|\boldsymbol{0}]\,\boldsymbol{p}. \tag{2.46}$$

If we are using homogeneous (projective) coordinates, we can write

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{\boldsymbol{p}}, \tag{2.47}$$

i.e., we drop the z component but keep the w component. Orthography is an approximate model for long focal length (telephoto) lenses and objects whose depth is *shallow* relative to their distance to the camera (Sawhney and Hanson 1991). It is exact only for *telecentric* lenses (Baker and Nayar 1999, 2001).

In practice, world coordinates (which may measure dimensions in meters) need to be scaled to fit onto an image sensor (physically measured in millimeters, but ultimately measured in pixels). For this reason, *scaled orthography* is actually more commonly used,

$$\boldsymbol{x} = [s\boldsymbol{I}_{2\times 2}|\boldsymbol{0}]\,\boldsymbol{p}. \tag{2.48}$$

This model is equivalent to first projecting the world points onto a local fronto-parallel image plane and then scaling this image using regular perspective projection. The scaling can be the same for all parts of the scene (Figure 2.7b) or it can be different for objects that are being modeled independently (Figure 2.7c). More importantly, the scaling can vary from frame to frame when estimating *structure from motion*, which can better model the scale change that occurs as an object approaches the camera.

Scaled orthography is a popular model for reconstructing the 3D shape of objects far away from the camera, since it greatly simplifies certain computations. For example, *pose* (camera

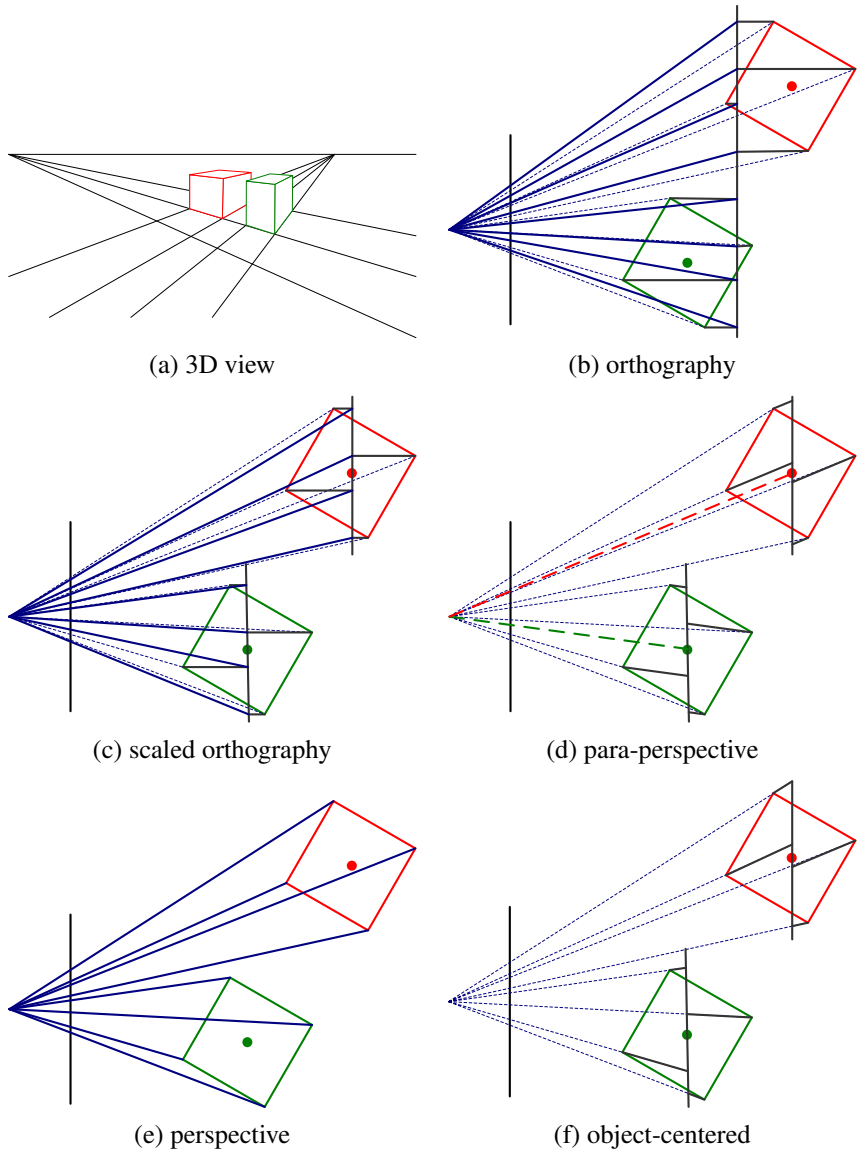


Figure 2.7 Commonly used projection models: (a) 3D view of world, (b) orthography, (c) scaled orthography, (d) para-perspective, (e) perspective, (f) object-centered. Each diagram shows a top-down view of the projection. Note how parallel lines on the ground plane and box sides remain parallel in the non-perspective projections.

orientation) can be estimated using simple least squares (Section 6.2.1). Under orthography, structure and motion can simultaneously be estimated using *factorization* (singular value decomposition), as discussed in Section 7.3 (Tomasi and Kanade 1992).

A closely related projection model is *para-perspective* (Aloimonos 1990; Poelman and Kanade 1997). In this model, object points are again first projected onto a local reference parallel to the image plane. However, rather than being projected orthogonally to this plane, they are projected *parallel* to the line of sight to the object center (Figure 2.7d). This is followed by the usual projection onto the final image plane, which again amounts to a scaling. The combination of these two projections is therefore *affine* and can be written as

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{\boldsymbol{p}}. \tag{2.49}$$

Note how parallel lines in 3D remain parallel after projection in Figure 2.7b–d. Para-perspective provides a more accurate projection model than scaled orthography, without incurring the added complexity of per-pixel perspective division, which invalidates traditional factorization methods (Poelman and Kanade 1997).

Perspective

The most commonly used projection in computer graphics and computer vision is true 3D *perspective* (Figure 2.7e). Here, points are projected onto the image plane by dividing them by their z component. Using inhomogeneous coordinates, this can be written as

$$\bar{\boldsymbol{x}} = \mathcal{P}_z(\boldsymbol{p}) = \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix}. \tag{2.50}$$

In homogeneous coordinates, the projection has a simple linear form,

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\boldsymbol{p}}, \tag{2.51}$$

i.e., we drop the w component of $\boldsymbol{p}$. Thus, after projection, it is not possible to recover the *distance* of the 3D point from the image, which makes sense for a 2D imaging sensor.

A form often seen in computer graphics systems is a two-step projection that first projects 3D coordinates into *normalized device coordinates* in the range $(x, y, z) \in [-1, -1] \times [-1, 1] \times [0, 1]$, and then rescales these coordinates to integer pixel coordinates using a *viewport* transformation (Watt 1995; OpenGL-ARB 1997). The (initial) perspective projection is then represented using a 4×4 matrix

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{\text{far}}/z_{\text{range}} & z_{\text{near}} z_{\text{far}}/z_{\text{range}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\boldsymbol{p}}, \tag{2.52}$$

where z_{near} and z_{far} are the near and far z *clipping planes* and $z_{\text{range}} = z_{\text{far}} - z_{\text{near}}$. Note that the first two rows are actually scaled by the focal length and the aspect ratio so that

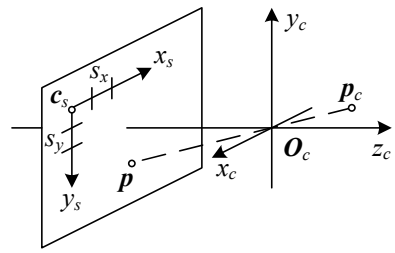


Figure 2.8 Projection of a 3D camera-centered point $\boldsymbol{p}_c$ onto the sensor planes at location $\boldsymbol{p}$. $\boldsymbol{O}_c$ is the camera center (nodal point), $\boldsymbol{c}_s$ is the 3D origin of the sensor plane coordinate system, and s_x and s_y are the pixel spacings.

visible rays are mapped to $(x, y, z) \in [-1, -1]^2$. The reason for keeping the third row, rather than dropping it, is that visibility operations, such as *z-buffering*, require a depth for every graphical element that is being rendered.

If we set $z_{\text{near}} = 1$, $z_{\text{far}} \to \infty$, and switch the sign of the third row, the third element of the normalized screen vector becomes the inverse depth, i.e., the *disparity* (Okutomi and Kanade 1993). This can be quite convenient in many cases since, for cameras moving around outdoors, the inverse depth to the camera is often a more well-conditioned parameterization than direct 3D distance.

While a regular 2D image sensor has no way of measuring distance to a surface point, *range sensors* (Section 12.2) and stereo matching algorithms (Chapter 11) can compute such values. It is then convenient to be able to map from a sensor-based depth or disparity value d directly back to a 3D location using the inverse of a 4×4 matrix (Section 2.1.5). We can do this if we represent perspective projection using a full-rank 4×4 matrix, as in (2.64).

Camera intrinsics

Once we have projected a 3D point through an ideal pinhole using a projection matrix, we must still transform the resulting coordinates according to the pixel sensor spacing and the relative position of the sensor plane to the origin. Figure 2.8 shows an illustration of the geometry involved. In this section, we first present a mapping from 2D pixel coordinates to 3D rays using a sensor homography $\boldsymbol{M}_s$, since this is easier to explain in terms of physically measurable quantities. We then relate these quantities to the more commonly used camera intrinsic matrix $\boldsymbol{K}$, which is used to map 3D camera-centered points $\boldsymbol{p}_c$ to 2D pixel coordinates $\tilde{\boldsymbol{x}}_s$.

Image sensors return pixel values indexed by integer *pixel coordinates* (x_s, y_s), often with the coordinates starting at the upper-left corner of the image and moving down and to the right. (This convention is not obeyed by all imaging libraries, but the adjustment for other coordinate systems is straightforward.) To map pixel centers to 3D coordinates, we first scale the (x_s, y_s) values by the pixel spacings (s_x, s_y) (sometimes expressed in microns for solid-state sensors) and then describe the orientation of the sensor array relative to the camera projection center $\boldsymbol{O}_c$ with an origin $\boldsymbol{c}_s$ and a 3D rotation $\boldsymbol{R}_s$ (Figure 2.8).

The combined 2D to 3D projection can then be written as

$$\boldsymbol{p} = \left[\begin{array}{c|c} \boldsymbol{R}_s & \boldsymbol{c}_s \end{array}\right] \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \boldsymbol{M}_s \bar{\boldsymbol{x}}_s. \tag{2.53}$$

The first two columns of the 3×3 matrix $\boldsymbol{M}_s$ are the 3D vectors corresponding to unit steps in the image pixel array along the x_s and y_s directions, while the third column is the 3D image array origin $\boldsymbol{c}_s$.

The matrix $\boldsymbol{M}_s$ is parameterized by eight unknowns: the three parameters describing the rotation $\boldsymbol{R}_s$, the three parameters describing the translation $\boldsymbol{c}_s$, and the two scale factors (s_x, s_y). Note that we ignore here the possibility of *skew* between the two axes on the image plane, since solid-state manufacturing techniques render this negligible. In practice, unless we have accurate external knowledge of the sensor spacing or sensor orientation, there are only seven degrees of freedom, since the distance of the sensor from the origin cannot be teased apart from the sensor spacing, based on external image measurement alone.

However, estimating a camera model $\boldsymbol{M}_s$ with the required seven degrees of freedom (i.e., where the first two columns are orthogonal after an appropriate re-scaling) is impractical, so most practitioners assume a general 3×3 homogeneous matrix form.

The relationship between the 3D pixel center $\boldsymbol{p}$ and the 3D camera-centered point $\boldsymbol{p}_c$ is given by an unknown scaling s, $\boldsymbol{p} = s\boldsymbol{p}_c$. We can therefore write the complete projection between $\boldsymbol{p}_c$ and a homogeneous version of the pixel address $\tilde{\boldsymbol{x}}_s$ as

$$\tilde{\boldsymbol{x}}_s = \alpha \boldsymbol{M}_s^{-1} \boldsymbol{p}_c = \boldsymbol{K} \boldsymbol{p}_c. \tag{2.54}$$

The 3×3 matrix $\boldsymbol{K}$ is called the *calibration matrix* and describes the camera *intrinsics* (as opposed to the camera's orientation in space, which are called the *extrinsics*).

From the above discussion, we see that $\boldsymbol{K}$ has seven degrees of freedom in theory and eight degrees of freedom (the full dimensionality of a 3×3 homogeneous matrix) in practice. Why, then, do most textbooks on 3D computer vision and multi-view geometry (Faugeras 1993; Hartley and Zisserman 2004; Faugeras and Luong 2001) treat $\boldsymbol{K}$ as an upper-triangular matrix with five degrees of freedom?

While this is usually not made explicit in these books, it is because we cannot recover the full $\boldsymbol{K}$ matrix based on external measurement alone. When calibrating a camera (Chapter 6) based on external 3D points or other measurements (Tsai 1987), we end up estimating the intrinsic $(\boldsymbol{K})$ and extrinsic $(\boldsymbol{R}, \boldsymbol{t})$ camera parameters simultaneously using a series of measurements,

$$\tilde{\boldsymbol{x}}_s = \boldsymbol{K} \left[\begin{array}{c|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right] \boldsymbol{p}_w = \boldsymbol{P} \boldsymbol{p}_w, \tag{2.55}$$

where $\boldsymbol{p}_w$ are known 3D world coordinates and

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}] \tag{2.56}$$

is known as the *camera matrix*. Inspecting this equation, we see that we can post-multiply $\boldsymbol{K}$ by $\boldsymbol{R}_1$ and pre-multiply $[\boldsymbol{R}|\boldsymbol{t}]$ by $\boldsymbol{R}_1^T$, and still end up with a valid calibration. Thus, it is impossible based on image measurements alone to know the true orientation of the sensor and the true camera intrinsics.

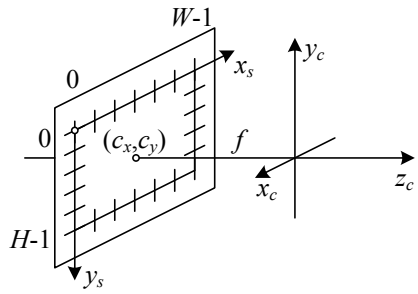


Figure 2.9 Simplified camera intrinsics showing the focal length f and the optical center (c_x, c_y). The image width and height are W and H.

The choice of an upper-triangular form for $\boldsymbol{K}$ seems to be conventional. Given a full 3×4 camera matrix $\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$, we can compute an upper-triangular $\boldsymbol{K}$ matrix using QR factorization (Golub and Van Loan 1996). (Note the unfortunate clash of terminologies: In matrix algebra textbooks, $\boldsymbol{R}$ represents an upper-triangular (right of the diagonal) matrix; in computer vision, $\boldsymbol{R}$ is an orthogonal rotation.)

There are several ways to write the upper-triangular form of $\boldsymbol{K}$. One possibility is

$$\boldsymbol{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.57}$$

which uses independent *focal lengths* f_x and f_y for the sensor x and y dimensions. The entry s encodes any possible *skew* between the sensor axes due to the sensor not being mounted perpendicular to the optical axis and (c_x, c_y) denotes the *optical center* expressed in pixel coordinates. Another possibility is

$$\boldsymbol{K} = \begin{bmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.58}$$

where the *aspect ratio* a has been made explicit and a common focal length f is used.

In practice, for many applications an even simpler form can be obtained by setting $a = 1$ and $s = 0$,

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.59}$$

Often, setting the origin at roughly the center of the image, e.g., $(c_x, c_y) = (W/2, H/2)$, where W and H are the image height and width, can result in a perfectly usable camera model with a single unknown, i.e., the focal length f.

Figure 2.9 shows how these quantities can be visualized as part of a simplified imaging model. Note that now we have placed the image plane *in front* of the nodal point (projection center of the lens). The sense of the y axis has also been flipped to get a coordinate system compatible with the way that most imaging libraries treat the vertical (row) coordinate. Certain graphics libraries, such as Direct3D, use a left-handed coordinate system, which can lead to some confusion.

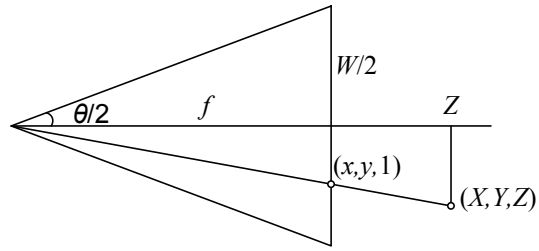


Figure 2.10 Central projection, showing the relationship between the 3D and 2D coordinates, $\boldsymbol{p}$ and $\boldsymbol{x}$, as well as the relationship between the focal length f, image width W, and the field of view θ.

A note on focal lengths

The issue of how to express focal lengths is one that often causes confusion in implementing computer vision algorithms and discussing their results. This is because the focal length depends on the units used to measure pixels.

If we number pixel coordinates using integer values, say $[0, W) \times [0, H)$, the focal length f and camera center (c_x, c_y) in (2.59) can be expressed as pixel values. How do these quantities relate to the more familiar focal lengths used by photographers?

Figure 2.10 illustrates the relationship between the focal length f, the sensor width W, and the field of view θ, which obey the formula

$$\tan\frac{\theta}{2} = \frac{W}{2f} \quad \text{or} \quad f = \frac{W}{2}\left[\tan\frac{\theta}{2}\right]^{-1}. \tag{2.60}$$

For conventional film cameras, $W = 35\text{mm}$, and hence f is also expressed in millimeters. Since we work with digital images, it is more convenient to express W in pixels so that the focal length f can be used directly in the calibration matrix $\boldsymbol{K}$ as in (2.59).

Another possibility is to scale the pixel coordinates so that they go from $[-1, 1)$ along the longer image dimension and $[-a^{-1}, a^{-1})$ along the shorter axis, where $a \geq 1$ is the *image aspect ratio* (as opposed to the *sensor cell aspect ratio* introduced earlier). This can be accomplished using *modified normalized device coordinates*,

$$x'_s = (2x_s - W)/S \;\text{ and }\; y'_s = (2y_s - H)/S, \quad \text{where} \quad S = \max(W, H). \tag{2.61}$$

This has the advantage that the focal length f and optical center (c_x, c_y) become independent of the image resolution, which can be useful when using multi-resolution, image-processing algorithms, such as image pyramids (Section 3.5).[2] The use of S instead of W also makes the focal length the same for landscape (horizontal) and portrait (vertical) pictures, as is the case in 35mm photography. (In some computer graphics textbooks and systems, normalized device coordinates go from $[-1, 1] \times [-1, 1]$, which requires the use of two different focal lengths to describe the camera intrinsics (Watt 1995; OpenGL-ARB 1997).) Setting $S = W = 2$ in (2.60), we obtain the simpler (unitless) relationship

$$f^{-1} = \tan\frac{\theta}{2}. \tag{2.62}$$

[2] To make the conversion truly accurate after a downsampling step in a pyramid, floating point values of W and H would have to be maintained since they can become non-integral if they are ever odd at a larger resolution in the pyramid.

The conversion between the various focal length representations is straightforward, e.g., to go from a unitless f to one expressed in pixels, multiply by $W/2$, while to convert from an f expressed in pixels to the equivalent 35mm focal length, multiply by $35/W$.

Camera matrix

Now that we have shown how to parameterize the calibration matrix $\boldsymbol{K}$, we can put the camera intrinsics and extrinsics together to obtain a single 3×4 *camera matrix*

$$\boldsymbol{P} = \boldsymbol{K} \left[\begin{array}{c|c} \boldsymbol{R} & \boldsymbol{t} \end{array} \right]. \tag{2.63}$$

It is sometimes preferable to use an invertible 4×4 matrix, which can be obtained by not dropping the last row in the $\boldsymbol{P}$ matrix,

$$\tilde{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} = \tilde{\boldsymbol{K}} \boldsymbol{E}, \tag{2.64}$$

where $\boldsymbol{E}$ is a 3D rigid-body (Euclidean) transformation and $\tilde{\boldsymbol{K}}$ is the full-rank calibration matrix. The 4×4 camera matrix $\tilde{\boldsymbol{P}}$ can be used to map directly from 3D world coordinates $\bar{\boldsymbol{p}}_w = (x_w, y_w, z_w, 1)$ to screen coordinates (plus disparity), $\boldsymbol{x}_s = (x_s, y_s, 1, d)$,

$$\boldsymbol{x}_s \sim \tilde{\boldsymbol{P}} \bar{\boldsymbol{p}}_w, \tag{2.65}$$

where $\sim$ indicates equality up to scale. Note that after multiplication by $\tilde{\boldsymbol{P}}$, the vector is divided by the *third* element of the vector to obtain the normalized form $\boldsymbol{x}_s = (x_s, y_s, 1, d)$.

Plane plus parallax (projective depth)

In general, when using the 4×4 matrix $\tilde{\boldsymbol{P}}$, we have the freedom to remap the last row to whatever suits our purpose (rather than just being the "standard" interpretation of disparity as inverse depth). Let us re-write the last row of $\tilde{\boldsymbol{P}}$ as $\boldsymbol{p}_3 = s_3[\hat{\boldsymbol{n}}_0 | c_0]$, where $\|\hat{\boldsymbol{n}}_0\| = 1$. We then have the equation

$$d = \frac{s_3}{z}(\hat{\boldsymbol{n}}_0 \cdot \boldsymbol{p}_w + c_0), \tag{2.66}$$

where $z = \boldsymbol{p}_2 \cdot \bar{\boldsymbol{p}}_w = \boldsymbol{r}_z \cdot (\boldsymbol{p}_w - \boldsymbol{c})$ is the distance of $\boldsymbol{p}_w$ from the camera center C (2.25) along the optical axis Z (Figure 2.11). Thus, we can interpret d as the *projective disparity* or *projective depth* of a 3D scene point $\boldsymbol{p}_w$ from the *reference plane* $\hat{\boldsymbol{n}}_0 \cdot \boldsymbol{p}_w + c_0 = 0$ (Szeliski and Coughlan 1997; Szeliski and Golland 1999; Shade, Gortler, He *et al.* 1998; Baker, Szeliski, and Anandan 1998). (The projective depth is also sometimes called *parallax* in reconstruction algorithms that use the term *plane plus parallax* (Kumar, Anandan, and Hanna 1994; Sawhney 1994).) Setting $\hat{\boldsymbol{n}}_0 = \boldsymbol{0}$ and $c_0 = 1$, i.e., putting the reference plane at infinity, results in the more standard $d = 1/z$ version of disparity (Okutomi and Kanade 1993).

Another way to see this is to invert the $\tilde{\boldsymbol{P}}$ matrix so that we can map pixels plus disparity directly back to 3D points,

$$\tilde{\boldsymbol{p}}_w = \tilde{\boldsymbol{P}}^{-1} \boldsymbol{x}_s. \tag{2.67}$$

In general, we can choose $\tilde{\boldsymbol{P}}$ to have whatever form is convenient, i.e., to sample space using an arbitrary projection. This can come in particularly handy when setting up multi-view

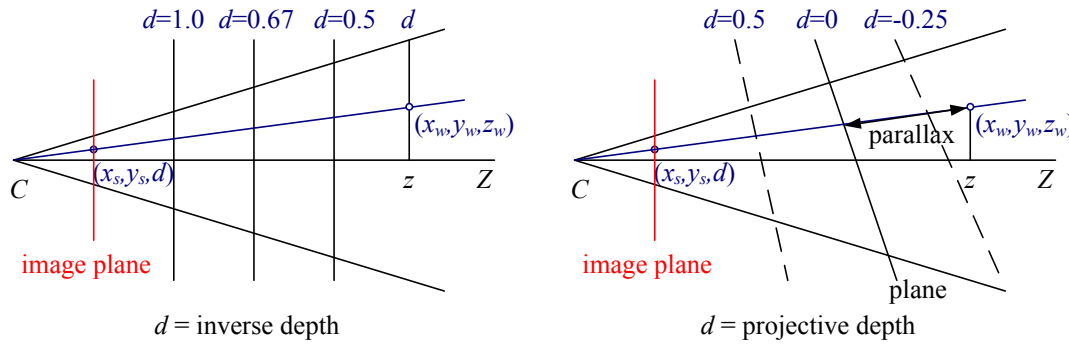


Figure 2.11 Regular disparity (inverse depth) and projective depth (parallax from a reference plane).

stereo reconstruction algorithms, since it allows us to sweep a series of planes (Section 11.1.2) through space with a variable (projective) sampling that best matches the sensed image motions (Collins 1996; Szeliski and Golland 1999; Saito and Kanade 1999).

Mapping from one camera to another

What happens when we take two images of a 3D scene from different camera positions or orientations (Figure 2.12a)? Using the full rank 4×4 camera matrix $\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{K}}\boldsymbol{E}$ from (2.64), we can write the projection from world to screen coordinates as

$$\tilde{\boldsymbol{x}}_0 \sim \tilde{\boldsymbol{K}}_0\boldsymbol{E}_0\boldsymbol{p} = \tilde{\boldsymbol{P}}_0\boldsymbol{p}. \tag{2.68}$$

Assuming that we know the z-buffer or disparity value d_0 for a pixel in one image, we can compute the 3D point location $\boldsymbol{p}$ using

$$\boldsymbol{p} \sim \boldsymbol{E}_0^{-1}\tilde{\boldsymbol{K}}_0^{-1}\tilde{\boldsymbol{x}}_0 \tag{2.69}$$

and then project it into another image yielding

$$\tilde{\boldsymbol{x}}_1 \sim \tilde{\boldsymbol{K}}_1\boldsymbol{E}_1\boldsymbol{p} = \tilde{\boldsymbol{K}}_1\boldsymbol{E}_1\boldsymbol{E}_0^{-1}\tilde{\boldsymbol{K}}_0^{-1}\tilde{\boldsymbol{x}}_0 = \tilde{\boldsymbol{P}}_1\tilde{\boldsymbol{P}}_0^{-1}\tilde{\boldsymbol{x}}_0 = \boldsymbol{M}_{10}\tilde{\boldsymbol{x}}_0. \tag{2.70}$$

Unfortunately, we do not usually have access to the depth coordinates of pixels in a regular photographic image. However, for a *planar scene*, as discussed above in (2.66), we can replace the last row of $\boldsymbol{P}_0$ in (2.64) with a general *plane equation*, $\hat{\boldsymbol{n}}_0 \cdot \boldsymbol{p} + c_0$ that maps points on the plane to $d_0 = 0$ values (Figure 2.12b). Thus, if we set $d_0 = 0$, we can ignore the last column of $\boldsymbol{M}_{10}$ in (2.70) and also its last row, since we do not care about the final z-buffer depth. The mapping equation (2.70) thus reduces to

$$\tilde{\boldsymbol{x}}_1 \sim \tilde{\boldsymbol{H}}_{10}\tilde{\boldsymbol{x}}_0, \tag{2.71}$$

where $\tilde{\boldsymbol{H}}_{10}$ is a general 3×3 homography matrix and $\tilde{\boldsymbol{x}}_1$ and $\tilde{\boldsymbol{x}}_0$ are now 2D homogeneous coordinates (i.e., 3-vectors) (Szeliski 1996).This justifies the use of the 8-parameter homography as a general alignment model for mosaics of planar scenes (Mann and Picard 1994; Szeliski 1996).

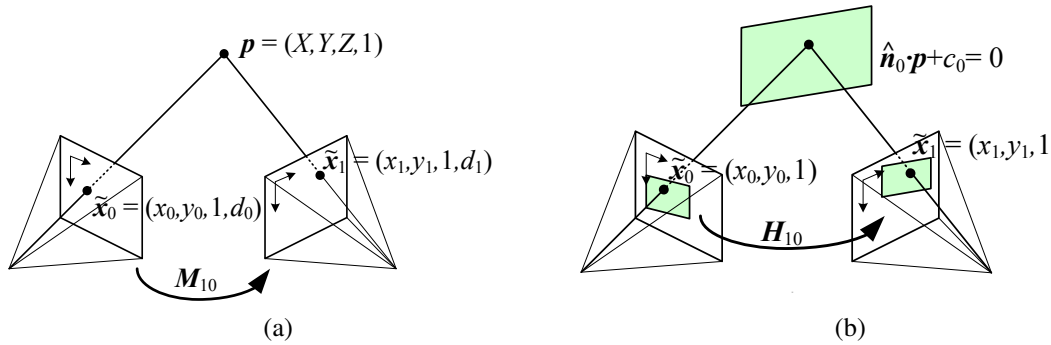


Figure 2.12 A point is projected into two images: (a) relationship between the 3D point coordinate $(X, Y, Z, 1)$ and the 2D projected point $(x, y, 1, d)$; (b) planar homography induced by points all lying on a common plane $\hat{\boldsymbol{n}}_0 \cdot \boldsymbol{p} + c_0 = 0$.

The other special case where we do not need to know depth to perform inter-camera mapping is when the camera is undergoing pure rotation (Section 9.1.3), i.e., when $\boldsymbol{t}_0 = \boldsymbol{t}_1$. In this case, we can write

$$\tilde{\boldsymbol{x}}_1 \sim \boldsymbol{K}_1 \boldsymbol{R}_1 \boldsymbol{R}_0^{-1} \boldsymbol{K}_0^{-1} \tilde{\boldsymbol{x}}_0 = \boldsymbol{K}_1 \boldsymbol{R}_{10} \boldsymbol{K}_0^{-1} \tilde{\boldsymbol{x}}_0, \tag{2.72}$$

which again can be represented with a 3×3 homography. If we assume that the calibration matrices have known aspect ratios and centers of projection (2.59), this homography can be parameterized by the rotation amount and the two unknown focal lengths. This particular formulation is commonly used in image-stitching applications (Section 9.1.3).

Object-centered projection

When working with long focal length lenses, it often becomes difficult to reliably estimate the focal length from image measurements alone. This is because the focal length and the distance to the object are highly correlated and it becomes difficult to tease these two effects apart. For example, the change in scale of an object viewed through a zoom telephoto lens can either be due to a zoom change or a motion towards the user. (This effect was put to dramatic use in some of Alfred Hitchcock's film *Vertigo*, where the simultaneous change of zoom and camera motion produces a disquieting effect.)

This ambiguity becomes clearer if we write out the projection equation corresponding to the simple calibration matrix $\boldsymbol{K}$ (2.59),

$$x_s = f \frac{\boldsymbol{r}_x \cdot \boldsymbol{p} + t_x}{\boldsymbol{r}_z \cdot \boldsymbol{p} + t_z} + c_x \tag{2.73}$$

$$y_s = f \frac{\boldsymbol{r}_y \cdot \boldsymbol{p} + t_y}{\boldsymbol{r}_z \cdot \boldsymbol{p} + t_z} + c_y, \tag{2.74}$$

where $\boldsymbol{r}_x$, $\boldsymbol{r}_y$, and $\boldsymbol{r}_z$ are the three rows of $\boldsymbol{R}$. If the distance to the object center $t_z \gg \|\boldsymbol{p}\|$ (the size of the object), the denominator is approximately t_z and the overall scale of the projected object depends on the ratio of f to t_z. It therefore becomes difficult to disentangle these two quantities.

To see this more clearly, let $\eta_z = t_z^{-1}$ and $s = \eta_z f$. We can then re-write the above equations as

$$x_s = s\frac{\boldsymbol{r}_x \cdot \boldsymbol{p} + t_x}{1 + \eta_z \boldsymbol{r}_z \cdot \boldsymbol{p}} + c_x \tag{2.75}$$

$$y_s = s\frac{\boldsymbol{r}_y \cdot \boldsymbol{p} + t_y}{1 + \eta_z \boldsymbol{r}_z \cdot \boldsymbol{p}} + c_y \tag{2.76}$$

(Szeliski and Kang 1994; Pighin, Hecker, Lischinski *et al.* 1998). The scale of the projection s can be reliably estimated if we are looking at a known object (i.e., the 3D coordinates $\boldsymbol{p}$ are known). The inverse distance η_z is now mostly decoupled from the estimates of s and can be estimated from the amount of *foreshortening* as the object rotates. Furthermore, as the lens becomes longer, i.e., the projection model becomes orthographic, there is no need to replace a perspective imaging model with an orthographic one, since the same equation can be used, with $\eta_z \to 0$ (as opposed to f and t_z both going to infinity). This allows us to form a natural link between orthographic reconstruction techniques such as factorization and their projective/perspective counterparts (Section 7.3).

2.1.6 Lens distortions

The above imaging models all assume that cameras obey a *linear* projection model where straight lines in the world result in straight lines in the image. (This follows as a natural consequence of linear matrix operations being applied to homogeneous coordinates.) Unfortunately, many wide-angle lenses have noticeable *radial distortion*, which manifests itself as a visible curvature in the projection of straight lines. (See Section 2.2.3 for a more detailed discussion of lens optics, including chromatic aberration.) Unless this distortion is taken into account, it becomes impossible to create highly accurate photorealistic reconstructions. For example, image mosaics constructed without taking radial distortion into account will often exhibit blurring due to the mis-registration of corresponding features before pixel blending (Chapter 9).

Fortunately, compensating for radial distortion is not that difficult in practice. For most lenses, a simple quartic model of distortion can produce good results. Let (x_c, y_c) be the pixel coordinates obtained *after* perspective division but *before* scaling by focal length f and shifting by the optical center (c_x, c_y), i.e.,

$$\begin{aligned} x_c &= \frac{\boldsymbol{r}_x \cdot \boldsymbol{p} + t_x}{\boldsymbol{r}_z \cdot \boldsymbol{p} + t_z} \\ y_c &= \frac{\boldsymbol{r}_y \cdot \boldsymbol{p} + t_y}{\boldsymbol{r}_z \cdot \boldsymbol{p} + t_z}. \end{aligned} \tag{2.77}$$

The radial distortion model says that coordinates in the observed images are displaced away (*barrel* distortion) or towards (*pincushion* distortion) the image center by an amount proportional to their radial distance (Figure 2.13a–b).[3] The simplest radial distortion models use low-order polynomials, e.g.,

$$\begin{aligned} \hat{x}_c &= x_c(1 + \kappa_1 r_c^2 + \kappa_2 r_c^4) \\ \hat{y}_c &= y_c(1 + \kappa_1 r_c^2 + \kappa_2 r_c^4), \end{aligned} \tag{2.78}$$

[3] Anamorphic lenses, which are widely used in feature film production, do not follow this radial distortion model. Instead, they can be thought of, to a first approximation, as inducing different vertical and horizontal scalings, i.e., non-square pixels.



(a) (b) (c)

Figure 2.13 Radial lens distortions: (a) barrel, (b) pincushion, and (c) fisheye. The fisheye image spans almost 180° from side-to-side.

where $r_c^2 = x_c^2 + y_c^2$ and κ_1 and κ_2 are called the *radial distortion parameters*.[4] After the radial distortion step, the final pixel coordinates can be computed using

$$\begin{aligned} x_s &= f x_c' + c_x \\ y_s &= f y_c' + c_y. \end{aligned} \tag{2.79}$$

A variety of techniques can be used to estimate the radial distortion parameters for a given lens, as discussed in Section 6.3.5.

Sometimes the above simplified model does not model the true distortions produced by complex lenses accurately enough (especially at very wide angles). A more complete analytic model also includes *tangential distortions* and *decentering distortions* (Slama 1980), but these distortions are not covered in this book.

Fisheye lenses (Figure 2.13c) require a model that differs from traditional polynomial models of radial distortion. Fisheye lenses behave, to a first approximation, as *equi-distance* projectors of angles away from the optical axis (Xiong and Turkowski 1997), which is the same as the *polar projection* described by Equations (9.22–9.24). Xiong and Turkowski (1997) describe how this model can be extended with the addition of an extra quadratic correction in ϕ and how the unknown parameters (center of projection, scaling factor s, etc.) can be estimated from a set of overlapping fisheye images using a direct (intensity-based) non-linear minimization algorithm.

For even larger, less regular distortions, a parametric distortion model using splines may be necessary (Goshtasby 1989). If the lens does not have a single center of projection, it may become necessary to model the 3D *line* (as opposed to *direction*) corresponding to each pixel separately (Gremban, Thorpe, and Kanade 1988; Champleboux, Lavallée, Sautot *et al.* 1992; Grossberg and Nayar 2001; Sturm and Ramalingam 2004; Tardif, Sturm, Trudeau *et al.* 2009). Some of these techniques are described in more detail in Section 6.3.5, which discusses how to calibrate lens distortions.

[4] Sometimes the relationship between x_c and $\hat{x}_c$ is expressed the other way around, i.e., $x_c = \hat{x}_c(1 + \kappa_1 \hat{r}_c^2 + \kappa_2 \hat{r}_c^4)$. This is convenient if we map image pixels into (warped) rays by dividing through by f. We can then undistort the rays and have true 3D rays in space.

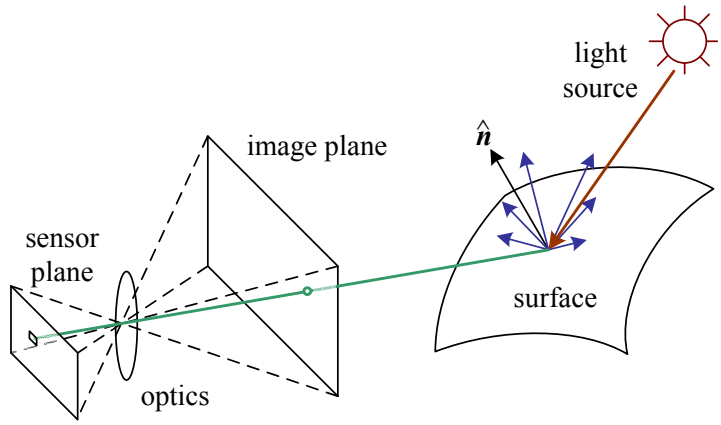


Figure 2.14 A simplified model of photometric image formation. Light is emitted by one or more light sources and is then reflected from an object's surface. A portion of this light is directed towards the camera. This simplified model ignores multiple reflections, which often occur in real-world scenes.

There is one subtle issue associated with the simple radial distortion model that is often glossed over. We have introduced a non-linearity between the perspective projection and final sensor array projection steps. Therefore, we cannot, in general, post-multiply an arbitrary 3×3 matrix $\boldsymbol{K}$ with a rotation to put it into upper-triangular form and absorb this into the global rotation. However, this situation is not as bad as it may at first appear. For many applications, keeping the simplified diagonal form of (2.59) is still an adequate model. Furthermore, if we correct radial and other distortions to an accuracy where straight lines are preserved, we have essentially converted the sensor back into a linear imager and the previous decomposition still applies.

2.2 Photometric image formation

In modeling the image formation process, we have described how 3D geometric features in the world are projected into 2D features in an image. However, images are not composed of 2D features. Instead, they are made up of discrete color or intensity values. Where do these values come from? How do they relate to the lighting in the environment, surface properties and geometry, camera optics, and sensor properties (Figure 2.14)? In this section, we develop a set of models to describe these interactions and formulate a generative process of image formation. A more detailed treatment of these topics can be found in other textbooks on computer graphics and image synthesis (Glassner 1995; Weyrich, Lawrence, Lensch *et al.* 2008; Foley, van Dam, Feiner *et al.* 1995; Watt 1995; Cohen and Wallace 1993; Sillion and Puech 1994).

2.2.1 Lighting

Images cannot exist without light. To produce an image, the scene must be illuminated with one or more light sources. (Certain modalities such as fluorescent microscopy and X-ray

tomography do not fit this model, but we do not deal with them in this book.) Light sources can generally be divided into point and area light sources.

A point light source originates at a single location in space (e.g., a small light bulb), potentially at infinity (e.g., the sun). (Note that for some applications such as modeling soft shadows (*penumbras*), the sun may have to be treated as an area light source.) In addition to its location, a point light source has an intensity and a color spectrum, i.e., a distribution over wavelengths $L(\lambda)$. The intensity of a light source falls off with the square of the distance between the source and the object being lit, because the same light is being spread over a larger (spherical) area. A light source may also have a directional falloff (dependence), but we ignore this in our simplified model.

Area light sources are more complicated. A simple area light source such as a fluorescent ceiling light fixture with a diffuser can be modeled as a finite rectangular area emitting light equally in all directions (Cohen and Wallace 1993; Sillion and Puech 1994; Glassner 1995). When the distribution is strongly directional, a four-dimensional lightfield can be used instead (Ashdown 1993).

A more complex light distribution that approximates, say, the incident illumination on an object sitting in an outdoor courtyard, can often be represented using an *environment map* (Greene 1986) (originally called a *reflection map* (Blinn and Newell 1976)). This representation maps incident light directions $\hat{\boldsymbol{v}}$ to color values (or wavelengths, λ),

$$L(\hat{\boldsymbol{v}}; \lambda), \tag{2.80}$$

and is equivalent to assuming that all light sources are at infinity. Environment maps can be represented as a collection of cubical faces (Greene 1986), as a single longitude–latitude map (Blinn and Newell 1976), or as the image of a reflecting sphere (Watt 1995). A convenient way to get a rough model of a real-world environment map is to take an image of a reflective mirrored sphere and to unwrap this image onto the desired environment map (Debevec 1998). Watt (1995) gives a nice discussion of environment mapping, including the formulas needed to map directions to pixels for the three most commonly used representations.

2.2.2 Reflectance and shading

When light hits an object's surface, it is scattered and reflected (Figure 2.15a). Many different models have been developed to describe this interaction. In this section, we first describe the most general form, the bidirectional reflectance distribution function, and then look at some more specialized models, including the diffuse, specular, and Phong shading models. We also discuss how these models can be used to compute the *global illumination* corresponding to a scene.

The Bidirectional Reflectance Distribution Function (BRDF)

The most general model of light scattering is the *bidirectional reflectance distribution function* (BRDF).[5] Relative to some local coordinate frame on the surface, the BRDF is a four-dimensional function that describes how much of each wavelength arriving at an *incident*

[5] Actually, even more general models of light transport exist, including some that model spatial variation along the surface, sub-surface scattering, and atmospheric effects—see Section 12.7.1—(Dorsey, Rushmeier, and Sillion 2007; Weyrich, Lawrence, Lensch *et al.* 2008).

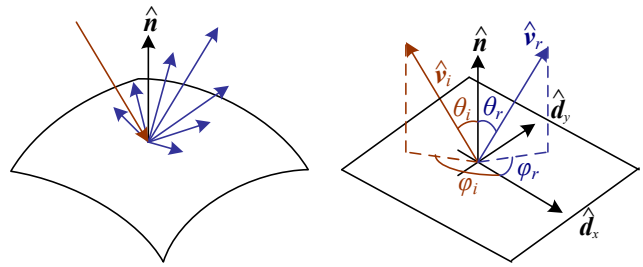


Figure 2.15 (a) Light scatters when it hits a surface. (b) The bidirectional reflectance distribution function (BRDF) $f(\theta_i, \phi_i, \theta_r, \phi_r)$ is parameterized by the angles that the incident, $\hat{\boldsymbol{v}}_i$, and reflected, $\hat{\boldsymbol{v}}_r$, light ray directions make with the local surface coordinate frame $(\hat{\boldsymbol{d}}_x, \hat{\boldsymbol{d}}_y, \hat{\boldsymbol{n}})$.

direction $\hat{\boldsymbol{v}}_i$ is emitted in a *reflected* direction $\hat{\boldsymbol{v}}_r$ (Figure 2.15b). The function can be written in terms of the angles of the incident and reflected directions relative to the surface frame as

$$f_r(\theta_i, \phi_i, \theta_r, \phi_r; \lambda). \tag{2.81}$$

The BRDF is *reciprocal*, i.e., because of the physics of light transport, you can interchange the roles of $\hat{\boldsymbol{v}}_i$ and $\hat{\boldsymbol{v}}_r$ and still get the same answer (this is sometimes called *Helmholtz reciprocity*).

Most surfaces are *isotropic*, i.e., there are no preferred directions on the surface as far as light transport is concerned. (The exceptions are *anisotropic* surfaces such as brushed (scratched) aluminum, where the reflectance depends on the light orientation relative to the direction of the scratches.) For an isotropic material, we can simplify the BRDF to

$$f_r(\theta_i, \theta_r, |\phi_r - \phi_i|; \lambda) \quad \text{or} \quad f_r(\hat{\boldsymbol{v}}_i, \hat{\boldsymbol{v}}_r, \hat{\boldsymbol{n}}; \lambda), \tag{2.82}$$

since the quantities θ_i, θ_r and $\phi_r - \phi_i$ can be computed from the directions $\hat{\boldsymbol{v}}_i$, $\hat{\boldsymbol{v}}_r$, and $\hat{\boldsymbol{n}}$.

To calculate the amount of light exiting a surface point $\boldsymbol{p}$ in a direction $\hat{\boldsymbol{v}}_r$ under a given lighting condition, we integrate the product of the incoming light $L_i(\hat{\boldsymbol{v}}_i; \lambda)$ with the BRDF (some authors call this step a *convolution*). Taking into account the *foreshortening* factor $\cos^+ \theta_i$, we obtain

$$L_r(\hat{\boldsymbol{v}}_r; \lambda) = \int L_i(\hat{\boldsymbol{v}}_i; \lambda) f_r(\hat{\boldsymbol{v}}_i, \hat{\boldsymbol{v}}_r, \hat{\boldsymbol{n}}; \lambda) \cos^+ \theta_i \, d\hat{\boldsymbol{v}}_i, \tag{2.83}$$

where

$$\cos^+ \theta_i = \max(0, \cos \theta_i). \tag{2.84}$$

If the light sources are discrete (a finite number of point light sources), we can replace the integral with a summation,

$$L_r(\hat{\boldsymbol{v}}_r; \lambda) = \sum_i L_i(\lambda) f_r(\hat{\boldsymbol{v}}_i, \hat{\boldsymbol{v}}_r, \hat{\boldsymbol{n}}; \lambda) \cos^+ \theta_i. \tag{2.85}$$

BRDFs for a given surface can be obtained through physical modeling (Torrance and Sparrow 1967; Cook and Torrance 1982; Glassner 1995), heuristic modeling (Phong 1975), or



Figure 2.16 This close-up of a statue shows both diffuse (smooth shading) and specular (shiny highlight) reflection, as well as darkening in the grooves and creases due to reduced light visibility and interreflections. (Photo courtesy of the Caltech Vision Lab, http://www.vision.caltech.edu/archive.html.)

through empirical observation (Ward 1992; Westin, Arvo, and Torrance 1992; Dana, van Ginneken, Nayar *et al.* 1999; Dorsey, Rushmeier, and Sillion 2007; Weyrich, Lawrence, Lensch *et al.* 2008).[6] Typical BRDFs can often be split into their *diffuse* and *specular* components, as described below.

Diffuse reflection

The diffuse component (also known as *Lambertian* or *matte* reflection) scatters light uniformly in all directions and is the phenomenon we most normally associate with *shading*, e.g., the smooth (non-shiny) variation of intensity with surface normal that is seen when observing a statue (Figure 2.16). Diffuse reflection also often imparts a strong *body color* to the light since it is caused by selective absorption and re-emission of light inside the object's material (Shafer 1985; Glassner 1995).

While light is scattered uniformly in all directions, i.e., the BRDF is constant,

$$f_d(\hat{\boldsymbol{v}}_i, \hat{\boldsymbol{v}}_r, \hat{\boldsymbol{n}}; \lambda) = f_d(\lambda), \tag{2.86}$$

the amount of light depends on the angle between the incident light direction and the surface normal θ_i. This is because the surface area exposed to a given amount of light becomes larger at oblique angles, becoming completely self-shadowed as the outgoing surface normal points away from the light (Figure 2.17a). (Think about how you orient yourself towards the sun or fireplace to get maximum warmth and how a flashlight projected obliquely against a wall is less bright than one pointing directly at it.) The *shading equation* for diffuse reflection can thus be written as

$$L_d(\hat{\boldsymbol{v}}_r; \lambda) = \sum_i L_i(\lambda) f_d(\lambda) \cos^+ \theta_i = \sum_i L_i(\lambda) f_d(\lambda) [\hat{\boldsymbol{v}}_i \cdot \hat{\boldsymbol{n}}]^+, \tag{2.87}$$

[6] See http://www1.cs.columbia.edu/CAVE/software/curet/ for a database of some empirically sampled BRDFs.

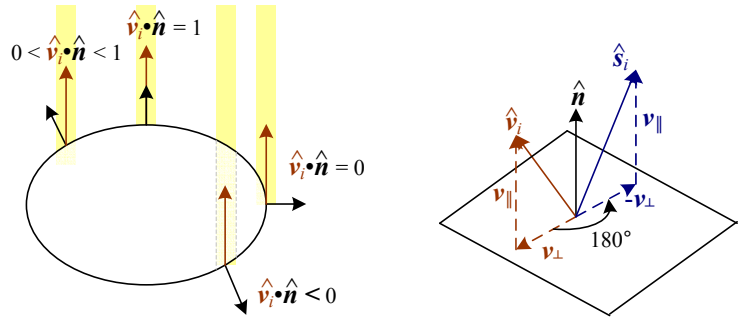


Figure 2.17 (a) The diminution of returned light caused by *foreshortening* depends on $\hat{\boldsymbol{v}}_i \cdot \hat{\boldsymbol{n}}$, the cosine of the angle between the incident light direction $\hat{\boldsymbol{v}}_i$ and the surface normal $\hat{\boldsymbol{n}}$. (b) Mirror (specular) reflection: The incident light ray direction $\hat{\boldsymbol{v}}_i$ is reflected onto the specular direction $\hat{\boldsymbol{s}}_i$ around the surface normal $\hat{\boldsymbol{n}}$.

where

$$[\hat{\boldsymbol{v}}_i \cdot \hat{\boldsymbol{n}}]^+ = \max(0, \hat{\boldsymbol{v}}_i \cdot \hat{\boldsymbol{n}}). \tag{2.88}$$

Specular reflection

The second major component of a typical BRDF is *specular* (gloss or highlight) reflection, which depends strongly on the direction of the outgoing light. Consider light reflecting off a mirrored surface (Figure 2.17b). Incident light rays are reflected in a direction that is rotated by 180° around the surface normal $\hat{\boldsymbol{n}}$. Using the same notation as in Equations (2.29–2.30), we can compute the *specular reflection* direction $\hat{\boldsymbol{s}}_i$ as

$$\hat{\boldsymbol{s}}_i = \boldsymbol{v}_\parallel - \boldsymbol{v}_\perp = (2\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T - \boldsymbol{I})\boldsymbol{v}_i. \tag{2.89}$$

The amount of light reflected in a given direction $\hat{\boldsymbol{v}}_r$ thus depends on the angle $\theta_s = \cos^{-1}(\hat{\boldsymbol{v}}_r \cdot \hat{\boldsymbol{s}}_i)$ between the view direction $\hat{\boldsymbol{v}}_r$ and the specular direction $\hat{\boldsymbol{s}}_i$. For example, the Phong (1975) model uses a power of the cosine of the angle,

$$f_s(\theta_s; \lambda) = k_s(\lambda) \cos^{k_e} \theta_s, \tag{2.90}$$

while the Torrance and Sparrow (1967) micro-facet model uses a Gaussian,

$$f_s(\theta_s; \lambda) = k_s(\lambda) \exp(-c_s^2 \theta_s^2). \tag{2.91}$$

Larger exponents k_e (or inverse Gaussian widths c_s) correspond to more specular surfaces with distinct highlights, while smaller exponents better model materials with softer gloss.

Phong shading

Phong (1975) combined the diffuse and specular components of reflection with another term, which he called the *ambient illumination*. This term accounts for the fact that objects are generally illuminated not only by point light sources but also by a general diffuse illumination corresponding to inter-reflection (e.g., the walls in a room) or distant sources, such as the

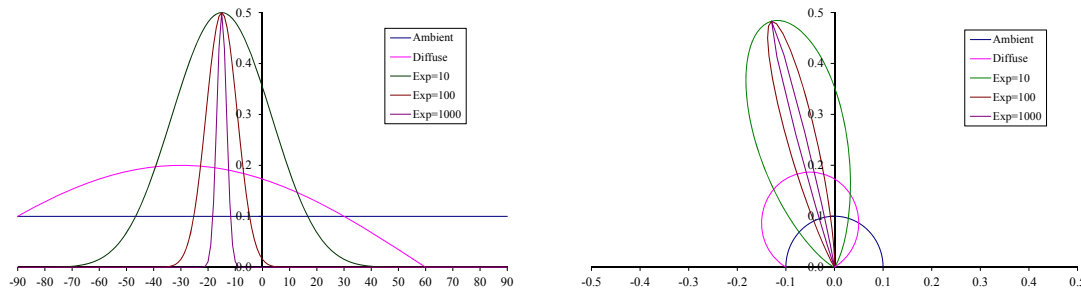


Figure 2.18 Cross-section through a Phong shading model BRDF for a fixed incident illumination direction: (a) component values as a function of angle away from surface normal; (b) polar plot. The value of the Phong exponent k_e is indicated by the "Exp" labels and the light source is at an angle of 30° away from the normal.

blue sky. In the Phong model, the ambient term does not depend on surface orientation, but depends on the color of both the ambient illumination $L_a(\lambda)$ and the object $k_a(\lambda)$,

$$f_a(\lambda) = k_a(\lambda)L_a(\lambda). \tag{2.92}$$

Putting all of these terms together, we arrive at the *Phong shading* model,

$$L_r(\hat{\boldsymbol{v}}_r;\lambda) = k_a(\lambda)L_a(\lambda) + k_d(\lambda)\sum_i L_i(\lambda)[\hat{\boldsymbol{v}}_i \cdot \hat{\boldsymbol{n}}]^+ + k_s(\lambda)\sum_i L_i(\lambda)(\hat{\boldsymbol{v}}_r \cdot \hat{\boldsymbol{s}}_i)^{k_e}. \tag{2.93}$$

Figure 2.18 shows a typical set of Phong shading model components as a function of the angle away from the surface normal (in a plane containing both the lighting direction and the viewer).

Typically, the ambient and diffuse reflection color distributions $k_a(\lambda)$ and $k_d(\lambda)$ are the same, since they are both due to sub-surface scattering (body reflection) inside the surface material (Shafer 1985). The specular reflection distribution $k_s(\lambda)$ is often uniform (white), since it is caused by interface reflections that do not change the light color. (The exception to this are *metallic* materials, such as copper, as opposed to the more common *dielectric* materials, such as plastics.)

The ambient illumination $L_a(\lambda)$ often has a different color cast from the direct light sources $L_i(\lambda)$, e.g., it may be blue for a sunny outdoor scene or yellow for an interior lit with candles or incandescent lights. (The presence of ambient sky illumination in shadowed areas is what often causes shadows to appear bluer than the corresponding lit portions of a scene). Note also that the diffuse component of the Phong model (or of any shading model) depends on the angle of the *incoming* light source $\hat{\boldsymbol{v}}_i$, while the specular component depends on the relative angle between the viewer $\boldsymbol{v}_r$ and the specular reflection direction $\hat{\boldsymbol{s}}_i$ (which itself depends on the incoming light direction $\hat{\boldsymbol{v}}_i$ and the surface normal $\hat{\boldsymbol{n}}$).

The Phong shading model has been superseded in terms of physical accuracy by a number of more recently developed models in computer graphics, including the model developed by Cook and Torrance (1982) based on the original micro-facet model of Torrance and Sparrow (1967). Until recently, most computer graphics hardware implemented the Phong model but the recent advent of programmable pixel shaders makes the use of more complex models feasible.

Di-chromatic reflection model

The Torrance and Sparrow (1967) model of reflection also forms the basis of Shafer's (1985) *di-chromatic reflection model*, which states that the apparent color of a uniform material lit from a single source depends on the sum of two terms,

$$L_r(\hat{\boldsymbol{v}}_r; \lambda) = L_i(\hat{\boldsymbol{v}}_r, \hat{\boldsymbol{v}}_i, \hat{\boldsymbol{n}}; \lambda) + L_b(\hat{\boldsymbol{v}}_r, \hat{\boldsymbol{v}}_i, \hat{\boldsymbol{n}}; \lambda) \tag{2.94}$$

$$= c_i(\lambda) m_i(\hat{\boldsymbol{v}}_r, \hat{\boldsymbol{v}}_i, \hat{\boldsymbol{n}}) + c_b(\lambda) m_b(\hat{\boldsymbol{v}}_r, \hat{\boldsymbol{v}}_i, \hat{\boldsymbol{n}}), \tag{2.95}$$

i.e., the radiance of the light reflected at the *interface*, L_i, and the radiance reflected at the *surface body*, L_b. Each of these, in turn, is a simple product between a relative power spectrum $c(\lambda)$, which depends only on wavelength, and a magnitude $m(\hat{\boldsymbol{v}}_r, \hat{\boldsymbol{v}}_i, \hat{\boldsymbol{n}})$, which depends only on geometry. (This model can easily be derived from a generalized version of Phong's model by assuming a single light source and no ambient illumination, and re-arranging terms.) The di-chromatic model has been successfully used in computer vision to segment specular colored objects with large variations in shading (Klinker 1993) and more recently has inspired local two-color models for applications such Bayer pattern demosaicing (Bennett, Uyttendaele, Zitnick *et al.* 2006).

Global illumination (ray tracing and radiosity)

The simple shading model presented thus far assumes that light rays leave the light sources, bounce off surfaces visible to the camera, thereby changing in intensity or color, and arrive at the camera. In reality, light sources can be shadowed by occluders and rays can bounce multiple times around a scene while making their trip from a light source to the camera.

Two methods have traditionally been used to model such effects. If the scene is mostly specular (the classic example being scenes made of glass objects and mirrored or highly polished balls), the preferred approach is *ray tracing* or *path tracing* (Glassner 1995; Akenine-Möller and Haines 2002; Shirley 2005), which follows individual rays from the camera across multiple bounces towards the light sources (or vice versa). If the scene is composed mostly of uniform albedo simple geometry illuminators and surfaces, *radiosity* (*global illumination*) techniques are preferred (Cohen and Wallace 1993; Sillion and Puech 1994; Glassner 1995). Combinations of the two techniques have also been developed (Wallace, Cohen, and Greenberg 1987), as well as more general *light transport* techniques for simulating effects such as the *caustics* cast by rippling water.

The basic ray tracing algorithm associates a light ray with each pixel in the camera image and finds its intersection with the nearest surface. A *primary* contribution can then be computed using the simple shading equations presented previously (e.g., Equation (2.93)) for all light sources that are visible for that surface element. (An alternative technique for computing which surfaces are illuminated by a light source is to compute a *shadow map*, or *shadow buffer*, i.e., a rendering of the scene from the light source's perspective, and then compare the depth of pixels being rendered with the map (Williams 1983; Akenine-Möller and Haines 2002).) Additional *secondary* rays can then be cast along the specular direction towards other objects in the scene, keeping track of any attenuation or color change that the specular reflection induces.

Radiosity works by associating lightness values with rectangular surface areas in the scene (including area light sources). The amount of light interchanged between any two (mutually

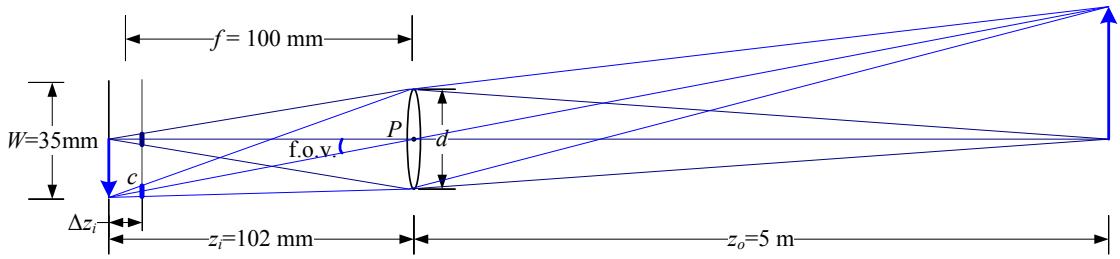


Figure 2.19 A thin lens of focal length f focuses the light from a plane a distance z_o in front of the lens at a distance z_i behind the lens, where $\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f}$. If the focal plane (vertical gray line next to c) is moved forward, the images are no longer in focus and the *circle of confusion* c (small thick line segments) depends on the distance of the image plane motion Δz_i relative to the lens aperture diameter d. The field of view (f.o.v.) depends on the ratio between the sensor width W and the focal length f (or, more precisely, the focusing distance z_i, which is usually quite close to f).

visible) areas in the scene can be captured as a *form factor*, which depends on their relative orientation and surface reflectance properties, as well as the $1/r^2$ fall-off as light is distributed over a larger effective sphere the further away it is (Cohen and Wallace 1993; Sillion and Puech 1994; Glassner 1995). A large linear system can then be set up to solve for the final lightness of each area patch, using the light sources as the forcing function (right hand side). Once the system has been solved, the scene can be rendered from any desired point of view. Under certain circumstances, it is possible to recover the global illumination in a scene from photographs using computer vision techniques (Yu, Debevec, Malik *et al.* 1999).

The basic radiosity algorithm does not take into account certain *near field* effects, such as the darkening inside corners and scratches, or the limited ambient illumination caused by partial shadowing from other surfaces. Such effects have been exploited in a number of computer vision algorithms (Nayar, Ikeuchi, and Kanade 1991; Langer and Zucker 1994).

While all of these global illumination effects can have a strong effect on the appearance of a scene, and hence its 3D interpretation, they are not covered in more detail in this book. (But see Section 12.7.1 for a discussion of recovering BRDFs from real scenes and objects.)

2.2.3 Optics

Once the light from a scene reaches the camera, it must still pass through the lens before reaching the sensor (analog film or digital silicon). For many applications, it suffices to treat the lens as an ideal pinhole that simply projects all rays through a common center of projection (Figures 2.8 and 2.9).

However, if we want to deal with issues such as focus, exposure, vignetting, and aberration, we need to develop a more sophisticated model, which is where the study of *optics* comes in (Möller 1988; Hecht 2001; Ray 2002).

Figure 2.19 shows a diagram of the most basic lens model, i.e., the *thin lens* composed of a single piece of glass with very low, equal curvature on both sides. According to the *lens law* (which can be derived using simple geometric arguments on light ray refraction), the relationship between the distance to an object z_o and the distance behind the lens at which a

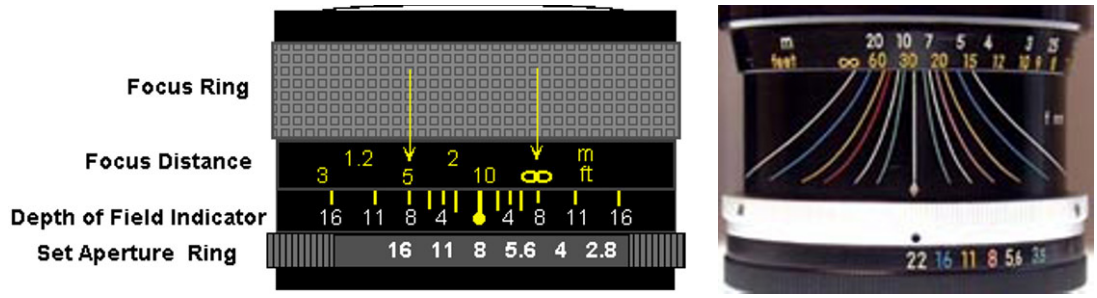


Figure 2.20 Regular and zoom lens depth of field indicators.

focused image is formed z_i can be expressed as

$$\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f}, \tag{2.96}$$

where f is called the *focal length* of the lens. If we let $z_o \to \infty$, i.e., we adjust the lens (move the image plane) so that objects at infinity are in focus, we get $z_i = f$, which is why we can think of a lens of focal length f as being equivalent (to a first approximation) to a pinhole a distance f from the focal plane (Figure 2.10), whose field of view is given by (2.60).

If the focal plane is moved away from its proper in-focus setting of z_i (e.g., by twisting the focus ring on the lens), objects at z_o are no longer in focus, as shown by the gray plane in Figure 2.19. The amount of mis-focus is measured by the *circle of confusion* c (shown as short thick blue line segments on the gray plane).[7] The equation for the circle of confusion can be derived using similar triangles; it depends on the distance of travel in the focal plane Δz_i relative to the original focus distance z_i and the diameter of the aperture d (see Exercise 2.4).

The allowable depth variation in the scene that limits the circle of confusion to an acceptable number is commonly called the *depth of field* and is a function of both the focus distance and the aperture, as shown diagrammatically by many lens markings (Figure 2.20). Since this depth of field depends on the aperture diameter d, we also have to know how this varies with the commonly displayed *f-number*, which is usually denoted as $f/\#$ or N and is defined as

$$f/\# = N = \frac{f}{d}, \tag{2.97}$$

where the focal length f and the aperture diameter d are measured in the same unit (say, millimeters).

The usual way to write the f-number is to replace the $\#$ in $f/\#$ with the actual number, i.e., $f/1.4, f/2, f/2.8, \ldots, f/22$. (Alternatively, we can say $N = 1.4$, etc.) An easy way to interpret these numbers is to notice that dividing the focal length by the f-number gives us the diameter d, so these are just formulas for the aperture diameter.[8]

[7] If the aperture is not completely circular, e.g., if it is caused by a hexagonal diaphragm, it is sometimes possible to see this effect in the actual blur function (Levin, Fergus, Durand *et al.* 2007; Joshi, Szeliski, and Kriegman 2008) or in the "glints" that are seen when shooting into the sun.

[8] This also explains why, with zoom lenses, the f-number varies with the current zoom (focal length) setting.

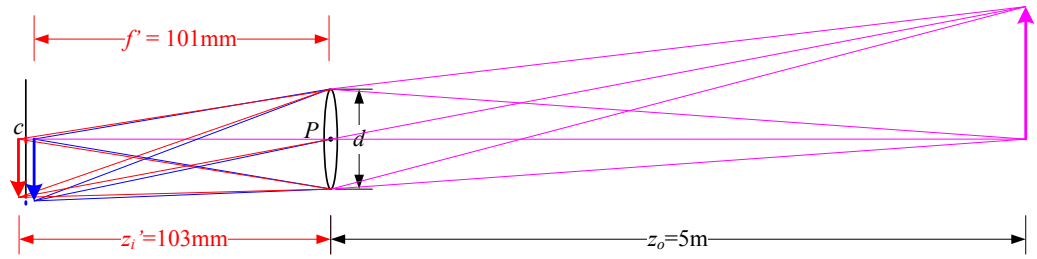


Figure 2.21 In a lens subject to *chromatic aberration*, light at different wavelengths (e.g., the red and blur arrows) is focused with a different focal length f' and hence a different depth z_i', resulting in both a geometric (in-plane) displacement and a loss of focus.

Notice that the usual progression for f-numbers is in *full stops*, which are multiples of $\sqrt{2}$, since this corresponds to doubling the area of the entrance pupil each time a smaller f-number is selected. (This doubling is also called changing the exposure by one *exposure value* or EV. It has the same effect on the amount of light reaching the sensor as doubling the exposure duration, e.g., from $1/125$ to $1/250$, see Exercise 2.5.)

Now that you know how to convert between f-numbers and aperture diameters, you can construct your own plots for the depth of field as a function of focal length f, circle of confusion c, and focus distance z_o, as explained in Exercise 2.4 and see how well these match what you observe on actual lenses, such as those shown in Figure 2.20.

Of course, real lenses are not infinitely thin and therefore suffer from geometric aberrations, unless compound elements are used to correct for them. The classic five *Seidel aberrations*, which arise when using *third-order optics*, include spherical aberration, coma, astigmatism, curvature of field, and distortion (Möller 1988; Hecht 2001; Ray 2002).

Chromatic aberration

Because the index of refraction for glass varies slightly as a function of wavelength, simple lenses suffer from *chromatic aberration*, which is the tendency for light of different colors to focus at slightly different distances (and hence also with slightly different magnification factors), as shown in Figure 2.21. The wavelength-dependent magnification factor, i.e., the *transverse chromatic aberration*, can be modeled as a per-color radial distortion (Section 2.1.6) and, hence, calibrated using the techniques described in Section 6.3.5. The wavelength-dependent blur caused by *longitudinal chromatic aberration* can be calibrated using techniques described in Section 10.1.4. Unfortunately, the blur induced by longitudinal aberration can be harder to undo, as higher frequencies can get strongly attenuated and hence hard to recover.

In order to reduce chromatic and other kinds of aberrations, most photographic lenses today are *compound lenses* made of different glass elements (with different coatings). Such lenses can no longer be modeled as having a single *nodal point* P through which all of the rays must pass (when approximating the lens with a pinhole model). Instead, these lenses have both a *front nodal point*, through which the rays enter the lens, and a *rear nodal point*, through which they leave on their way to the sensor. In practice, only the location of the front

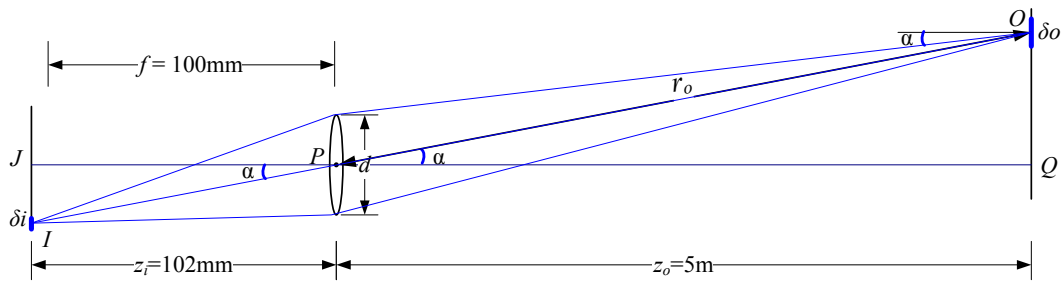


Figure 2.22 The amount of light hitting a pixel of surface area δi depends on the square of the ratio of the aperture diameter d to the focal length f, as well as the fourth power of the off-axis angle α cosine, $\cos^4 \alpha$.

nodal point is of interest when performing careful camera calibration, e.g., when determining the point around which to rotate to capture a parallax-free panorama (see Section 9.1.3).

Not all lenses, however, can be modeled as having a single nodal point. In particular, very wide-angle lenses such as fisheye lenses (Section 2.1.6) and certain *catadioptric* imaging systems consisting of lenses and curved mirrors (Baker and Nayar 1999) do not have a single point through which all of the acquired light rays pass. In such cases, it is preferable to explicitly construct a mapping function (look-up table) between pixel coordinates and 3D rays in space (Gremban, Thorpe, and Kanade 1988; Champleboux, Lavallée, Sautot *et al.* 1992; Grossberg and Nayar 2001; Sturm and Ramalingam 2004; Tardif, Sturm, Trudeau *et al.* 2009), as mentioned in Section 2.1.6.

Vignetting

Another property of real-world lenses is *vignetting*, which is the tendency for the brightness of the image to fall off towards the edge of the image.

Two kinds of phenomena usually contribute to this effect (Ray 2002). The first is called *natural vignetting* and is due to the foreshortening in the object surface, projected pixel, and lens aperture, as shown in Figure 2.22. Consider the light leaving the object surface patch of size δo located at an *off-axis angle* α. Because this patch is foreshortened with respect to the camera lens, the amount of light reaching the lens is reduced by a factor $\cos \alpha$. The amount of light reaching the lens is also subject to the usual $1/r^2$ fall-off; in this case, the distance $r_o = z_o / \cos \alpha$. The actual area of the aperture through which the light passes is foreshortened by an additional factor $\cos \alpha$, i.e., the aperture as seen from point O is an ellipse of dimensions $d \times d \cos \alpha$. Putting all of these factors together, we see that the amount of light leaving O and passing through the aperture on its way to the image pixel located at I is proportional to

$$\frac{\delta o \cos \alpha}{r_o^2} \pi \left(\frac{d}{2} \right)^2 \cos \alpha = \delta o \frac{\pi}{4} \frac{d^2}{z_o^2} \cos^4 \alpha. \tag{2.98}$$

Since triangles ΔOPQ and ΔIPJ are similar, the projected areas of of the object surface δo and image pixel δi are in the same (squared) ratio as $z_o : z_i$,

$$\frac{\delta o}{\delta i} = \frac{z_o^2}{z_i^2}. \tag{2.99}$$

Putting these together, we obtain the final relationship between the amount of light reaching pixel i and the aperture diameter d, the focusing distance $z_i \approx f$, and the off-axis angle α,

$$\delta o \frac{\pi}{4}\frac{d^2}{z_o^2}\cos^4\alpha = \delta i \frac{\pi}{4}\frac{d^2}{z_i^2}\cos^4\alpha \approx \delta i \frac{\pi}{4}\left(\frac{d}{f}\right)^2\cos^4\alpha, \tag{2.100}$$

which is called the *fundamental radiometric relation* between the scene radiance L and the light (irradiance) E reaching the pixel sensor,

$$E = L\frac{\pi}{4}\left(\frac{d}{f}\right)^2\cos^4\alpha, \tag{2.101}$$

(Horn 1986; Nalwa 1993; Hecht 2001; Ray 2002). Notice in this equation how the amount of light depends on the pixel surface area (which is why the smaller sensors in point-and-shoot cameras are so much noisier than digital single lens reflex (SLR) cameras), the inverse square of the f-stop $N = f/d$ (2.97), and the fourth power of the $\cos^4\alpha$ off-axis fall-off, which is the natural vignetting term.

The other major kind of vignetting, called *mechanical vignetting*, is caused by the internal occlusion of rays near the periphery of lens elements in a compound lens, and cannot easily be described mathematically without performing a full ray-tracing of the actual lens design.[9] However, unlike natural vignetting, mechanical vignetting can be decreased by reducing the camera aperture (increasing the f-number). It can also be calibrated (along with natural vignetting) using special devices such as integrating spheres, uniformly illuminated targets, or camera rotation, as discussed in Section 10.1.3.

2.3 The digital camera

After starting from one or more light sources, reflecting off one or more surfaces in the world, and passing through the camera's optics (lenses), light finally reaches the imaging sensor. How are the photons arriving at this sensor converted into the digital (R, G, B) values that we observe when we look at a digital image? In this section, we develop a simple model that accounts for the most important effects such as exposure (gain and shutter speed), non-linear mappings, sampling and aliasing, and noise. Figure 2.23, which is based on camera models developed by Healey and Kondepudy (1994); Tsin, Ramesh, and Kanade (2001); Liu, Szeliski, Kang *et al.* (2008), shows a simple version of the processing stages that occur in modern digital cameras. Chakrabarti, Scharstein, and Zickler (2009) developed a sophisticated 24-parameter model that is an even better match to the processing performed in today's cameras.

Light falling on an imaging sensor is usually picked up by an *active sensing area*, integrated for the duration of the exposure (usually expressed as the shutter speed in a fraction of a second, e.g., $\frac{1}{125}$, $\frac{1}{60}$, $\frac{1}{30}$), and then passed to a set of *sense amplifiers* . The two main kinds of sensor used in digital still and video cameras today are charge-coupled device (CCD) and complementary metal oxide on silicon (CMOS).

In a CCD, photons are accumulated in each active *well* during the exposure time. Then, in a *transfer* phase, the charges are transferred from well to well in a kind of "bucket brigade"

[9] There are some empirical models that work well in practice (Kang and Weiss 2000; Zheng, Lin, and Kang 2006).

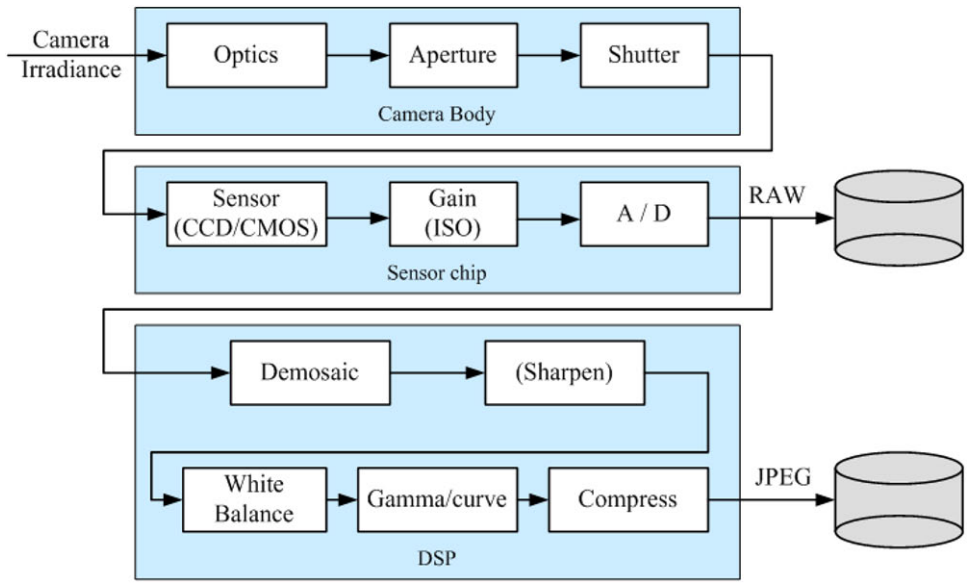


Figure 2.23 Image sensing pipeline, showing the various sources of noise as well as typical digital post-processing steps.

until they are deposited at the sense amplifiers, which amplify the signal and pass it to an analog-to-digital converter (ADC).[10] Older CCD sensors were prone to *blooming*, when charges from one over-exposed pixel spilled into adjacent ones, but most newer CCDs have anti-blooming technology ("troughs" into which the excess charge can spill).

In CMOS, the photons hitting the sensor directly affect the conductivity (or gain) of a photodetector, which can be selectively gated to control exposure duration, and locally amplified before being read out using a multiplexing scheme. Traditionally, CCD sensors outperformed CMOS in quality sensitive applications, such as digital SLRs, while CMOS was better for low-power applications, but today CMOS is used in most digital cameras.

The main factors affecting the performance of a digital image sensor are the shutter speed, sampling pitch, fill factor, chip size, analog gain, sensor noise, and the resolution (and quality) of the analog-to-digital converter. Many of the actual values for these parameters can be read from the EXIF tags embedded with digital images. while others can be obtained from the camera manufacturers' specification sheets or from camera review or calibration Web sites.[11]

Shutter speed. The shutter speed (exposure time) directly controls the amount of light reaching the sensor and, hence, determines if images are under- or over-exposed. (For bright scenes, where a large aperture or slow shutter speed are desired to get a shallow depth of field or motion blur, *neutral density filters* are sometimes used by photographers.) For dynamic scenes, the shutter speed also determines the amount of *motion blur* in the resulting picture.

[10] In digital still cameras, a complete frame is captured and then read out sequentially at once. However, if video is being captured, a *rolling shutter*, which exposes and transfers each line separately, is often used. In older video cameras, the even fields (lines) were scanned first, followed by the odd fields, in a process that is called *interlacing*.

[11] http://www.clarkvision.com/imagedetail/digital.sensor.performance.summary/ .

Usually, a higher shutter speed (less motion blur) makes subsequent analysis easier (see Section 10.3 for techniques to remove such blur). However, when video is being captured for display, some motion blur may be desirable to avoid stroboscopic effects.

Sampling pitch. The sampling pitch is the physical spacing between adjacent sensor cells on the imaging chip. A sensor with a smaller sampling pitch has a higher *sampling density* and hence provides a higher *resolution* (in terms of pixels) for a given active chip area. However, a smaller pitch also means that each sensor has a smaller area and cannot accumulate as many photons; this makes it not as *light sensitive* and more prone to noise.

Fill factor. The fill factor is the active sensing area size as a fraction of the theoretically available sensing area (the product of the horizontal and vertical sampling pitches). Higher fill factors are usually preferable, as they result in more light capture and less *aliasing* (see Section 2.3.1). However, this must be balanced with the need to place additional electronics between the active sense areas. The fill factor of a camera can be determined empirically using a photometric camera calibration process (see Section 10.1.4).

Chip size. Video and point-and-shoot cameras have traditionally used small chip areas ($\frac{1}{4}$-inch to $\frac{1}{2}$-inch sensors[12]), while digital SLR cameras try to come closer to the traditional size of a 35mm film frame.[13] When overall device size is not important, having a larger chip size is preferable, since each sensor cell can be more photo-sensitive. (For compact cameras, a smaller chip means that all of the optics can be shrunk down proportionately.) However, larger chips are more expensive to produce, not only because fewer chips can be packed into each wafer, but also because the probability of a chip defect goes up linearly with the chip area.

Analog gain. Before analog-to-digital conversion, the sensed signal is usually boosted by a *sense amplifier*. In video cameras, the gain on these amplifiers was traditionally controlled by *automatic gain control* (AGC) logic, which would adjust these values to obtain a good overall exposure. In newer digital still cameras, the user now has some additional control over this gain through the *ISO setting*, which is typically expressed in ISO standard units such as 100, 200, or 400. Since the automated exposure control in most cameras also adjusts the aperture and shutter speed, setting the ISO manually removes one degree of freedom from the camera's control, just as manually specifying aperture and shutter speed does. In theory, a higher gain allows the camera to perform better under low light conditions (less motion blur due to long exposure times when the aperture is already maxed out). In practice, however, higher ISO settings usually amplify the *sensor noise*.

[12] These numbers refer to the "tube diameter" of the old vidicon tubes used in video cameras (http://www.dpreview.com/learn/?/Glossary/Camera_System/sensor_sizes_01.htm). The 1/2.5" sensor on the Canon SD800 camera actually measures 5.76mm × 4.29mm, i.e., a sixth of the size (on side) of a 35mm full-frame (36mm × 24mm) DSLR sensor.

[13] When a DSLR chip does not fill the 35mm full frame, it results in a *multiplier effect* on the lens focal length. For example, a chip that is only 0.6 the dimension of a 35mm frame will make a 50mm lens image the same angular extent as a $50/0.6 = 50 \times 1.6 =$80mm lens, as demonstrated in (2.60).

Sensor noise. Throughout the whole sensing process, noise is added from various sources, which may include *fixed pattern noise*, *dark current noise*, *shot noise*, *amplifier noise* and *quantization noise* (Healey and Kondepudy 1994; Tsin, Ramesh, and Kanade 2001). The final amount of noise present in a sampled image depends on all of these quantities, as well as the incoming light (controlled by the scene radiance and aperture), the exposure time, and the sensor gain. Also, for low light conditions where the noise is due to low photon counts, a Poisson model of noise may be more appropriate than a Gaussian model.

As discussed in more detail in Section 10.1.1, Liu, Szeliski, Kang *et al.* (2008) use this model, along with an empirical database of camera response functions (CRFs) obtained by Grossberg and Nayar (2004), to estimate the *noise level function* (NLF) for a given image, which predicts the overall noise variance at a given pixel as a function of its brightness (a separate NLF is estimated for each color channel). An alternative approach, when you have access to the camera before taking pictures, is to pre-calibrate the NLF by taking repeated shots of a scene containing a variety of colors and luminances, such as the Macbeth Color Chart shown in Figure 10.3b (McCamy, Marcus, and Davidson 1976). (When estimating the variance, be sure to throw away or downweight pixels with large gradients, as small shifts between exposures will affect the sensed values at such pixels.) Unfortunately, the pre-calibration process may have to be repeated for different exposure times and gain settings because of the complex interactions occurring within the sensing system.

In practice, most computer vision algorithms, such as image denoising, edge detection, and stereo matching, all benefit from at least a rudimentary estimate of the noise level. Barring the ability to pre-calibrate the camera or to take repeated shots of the same scene, the simplest approach is to look for regions of near-constant value and to estimate the noise variance in such regions (Liu, Szeliski, Kang *et al.* 2008).

ADC resolution. The final step in the analog processing chain occurring within an imaging sensor is the *analog to digital conversion* (ADC). While a variety of techniques can be used to implement this process, the two quantities of interest are the *resolution* of this process (how many bits it yields) and its noise level (how many of these bits are useful in practice). For most cameras, the number of bits quoted (eight bits for compressed JPEG images and a nominal 16 bits for the RAW formats provided by some DSLRs) exceeds the actual number of usable bits. The best way to tell is to simply calibrate the noise of a given sensor, e.g., by taking repeated shots of the same scene and plotting the estimated noise as a function of brightness (Exercise 2.6).

Digital post-processing. Once the irradiance values arriving at the sensor have been converted to digital bits, most cameras perform a variety of *digital signal processing* (DSP) operations to enhance the image before compressing and storing the pixel values. These include color filter array (CFA) demosaicing, white point setting, and mapping of the luminance values through a *gamma function* to increase the perceived dynamic range of the signal. We cover these topics in Section 2.3.2 but, before we do, we return to the topic of aliasing, which was mentioned in connection with sensor array fill factors.

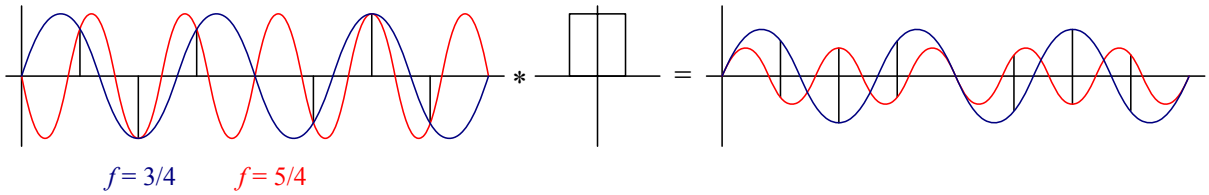


Figure 2.24 Aliasing of a one-dimensional signal: The blue sine wave at $f = 3/4$ and the red sine wave at $f = 5/4$ have the same digital samples, when sampled at $f = 2$. Even after convolution with a 100% fill factor box filter, the two signals, while no longer of the same magnitude, are still aliased in the sense that the sampled red signal looks like an inverted lower magnitude version of the blue signal. (The image on the right is scaled up for better visibility. The actual sine magnitudes are 30% and −18% of their original values.)

2.3.1 Sampling and aliasing

What happens when a field of light impinging on the image sensor falls onto the active sense areas in the imaging chip? The photons arriving at each active cell are integrated and then digitized. However, if the fill factor on the chip is small and the signal is not otherwise *band-limited*, visually unpleasing aliasing can occur.

To explore the phenomenon of aliasing, let us first look at a one-dimensional signal (Figure 2.24), in which we have two sine waves, one at a frequency of $f = {}^3/_4$ and the other at $f = {}^5/_4$. If we sample these two signals at a frequency of $f = 2$, we see that they produce the same samples (shown in black), and so we say that they are *aliased*.[14] Why is this a bad effect? In essence, we can no longer reconstruct the original signal, since we do not know which of the two original frequencies was present.

In fact, Shannon's Sampling Theorem shows that the minimum sampling (Oppenheim and Schafer 1996; Oppenheim, Schafer, and Buck 1999) rate required to reconstruct a signal from its instantaneous samples must be at least twice the highest frequency,[15]

$$f_s \geq 2 f_{\max}. \tag{2.102}$$

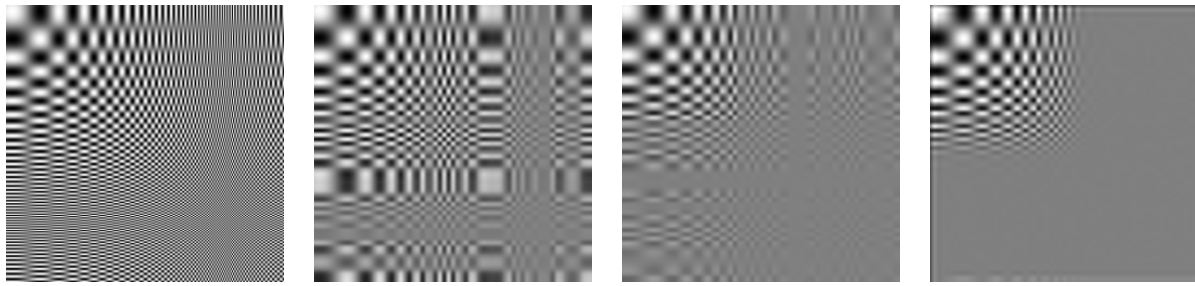
The maximum frequency in a signal is known as the *Nyquist frequency* and the inverse of the minimum sampling frequency $r_s = 1/f_s$ is known as the *Nyquist rate*.

However, you may ask, since an imaging chip actually *averages* the light field over a finite area, are the results on point sampling still applicable? Averaging over the sensor area does tend to attenuate some of the higher frequencies. However, even if the fill factor is 100%, as in the right image of Figure 2.24, frequencies above the Nyquist limit (half the sampling frequency) still produce an aliased signal, although with a smaller magnitude than the corresponding band-limited signals.

A more convincing argument as to why aliasing is bad can be seen by downsampling a signal using a poor quality filter such as a box (square) filter. Figure 2.25 shows a high-frequency *chirp* image (so called because the frequencies increase over time), along with the results of sampling it with a 25% fill-factor area sensor, a 100% fill-factor sensor, and a high-

[14] An alias is an alternate name for someone, so the sampled signal corresponds to two different *aliases*.

[15] The actual theorem states that f_s must be at least twice the signal *bandwidth* but, since we are not dealing with modulated signals such as radio waves during image capture, the maximum frequency suffices.



(a) (b) (c) (d)

Figure 2.25 Aliasing of a two-dimensional signal: (a) original full-resolution image; (b) downsampled $4\times$ with a 25% fill factor box filter; (c) downsampled $4\times$ with a 100% fill factor box filter; (d) downsampled $4\times$ with a high-quality 9-tap filter. Notice how the higher frequencies are aliased into visible frequencies with the lower quality filters, while the 9-tap filter completely removes these higher frequencies.

quality 9-tap filter. Additional examples of downsampling (*decimation*) filters can be found in Section 3.5.2 and Figure 3.30.

The best way to predict the amount of aliasing that an imaging system (or even an image processing algorithm) will produce is to estimate the *point spread function* (PSF), which represents the response of a particular pixel sensor to an ideal point light source. The PSF is a combination (convolution) of the blur induced by the optical system (lens) and the finite integration area of a chip sensor.[16]

If we know the blur function of the lens and the fill factor (sensor area shape and spacing) for the imaging chip (plus, optionally, the response of the anti-aliasing filter), we can convolve these (as described in Section 3.2) to obtain the PSF. Figure 2.26a shows the one-dimensional cross-section of a PSF for a lens whose blur function is assumed to be a disc of a radius equal to the pixel spacing s plus a sensing chip whose horizontal fill factor is 80%. Taking the Fourier transform of this PSF (Section 3.4), we obtain the *modulation transfer function* (MTF), from which we can estimate the amount of aliasing as the area of the Fourier magnitude outside the $f \leq f_s$ Nyquist frequency.[17] If we de-focus the lens so that the blur function has a radius of $2s$ (Figure 2.26c), we see that the amount of aliasing decreases significantly, but so does the amount of image detail (frequencies closer to $f = f_s$).

Under laboratory conditions, the PSF can be estimated (to pixel precision) by looking at a point light source such as a pin hole in a black piece of cardboard lit from behind. However, this PSF (the actual image of the pin hole) is only accurate to a pixel resolution and, while it can model larger blur (such as blur caused by defocus), it cannot model the sub-pixel shape of the PSF and predict the amount of aliasing. An alternative technique, described in Section 10.1.4, is to look at a calibration pattern (e.g., one consisting of slanted step edges (Reichenbach, Park, and Narayanswamy 1991; Williams and Burns 2001; Joshi, Szeliski, and Kriegman 2008)) whose ideal appearance can be re-synthesized to sub-pixel precision.

In addition to occurring during image acquisition, aliasing can also be introduced in var-

[16] Imaging chips usually interpose an optical *anti-aliasing filter* just before the imaging chip to reduce or control the amount of aliasing.

[17] The complex Fourier transform of the PSF is actually called the *optical transfer function* (OTF) (Williams 1999). Its magnitude is called the *modulation transfer function* (MTF) and its phase is called the *phase transfer function* (PTF).

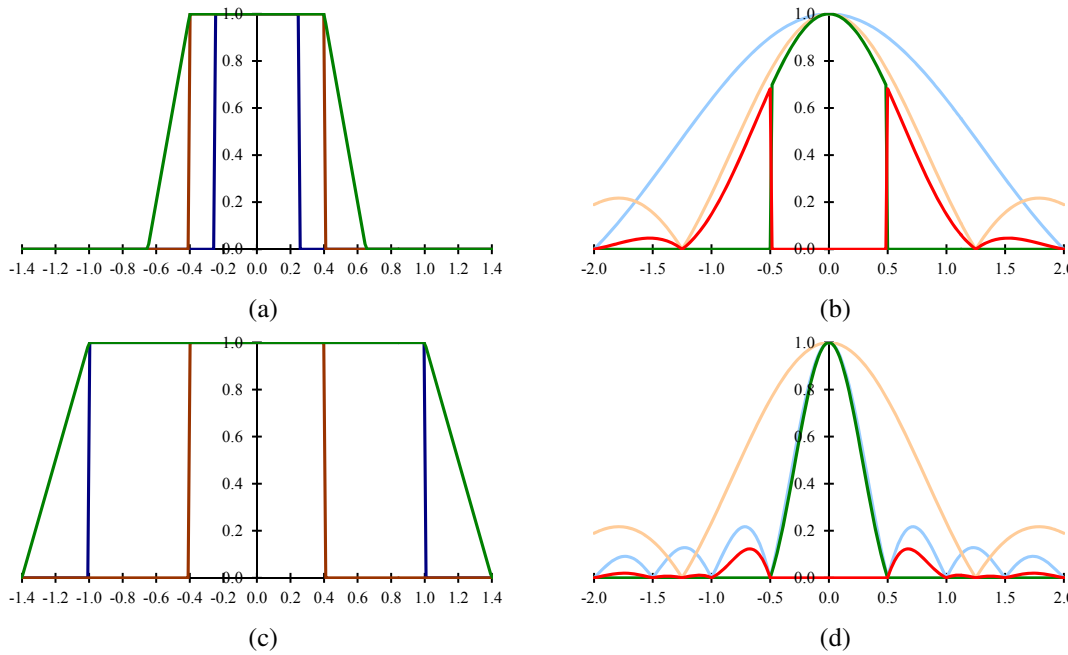


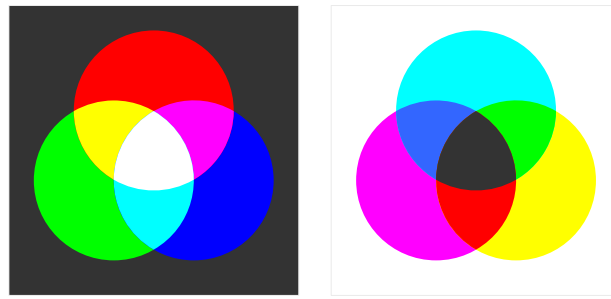
Figure 2.26 Sample point spread functions (PSF): The diameter of the blur disc (blue) in (a) is equal to half the pixel spacing, while the diameter in (c) is twice the pixel spacing. The horizontal fill factor of the sensing chip is 80% and is shown in brown. The convolution of these two kernels gives the point spread function, shown in green. The Fourier response of the PSF (the MTF) is plotted in (b) and (d). The area above the Nyquist frequency where aliasing occurs is shown in red.

ious image processing operations, such as resampling, upsampling, and downsampling. Sections 3.4 and 3.5.2 discuss these issues and show how careful selection of filters can reduce the amount of aliasing that operations inject.

2.3.2 Color

In Section 2.2, we saw how lighting and surface reflections are functions of wavelength. When the incoming light hits the imaging sensor, light from different parts of the spectrum is somehow integrated into the discrete red, green, and blue (RGB) color values that we see in a digital image. How does this process work and how can we analyze and manipulate color values?

You probably recall from your childhood days the magical process of mixing paint colors to obtain new ones. You may recall that blue+yellow makes green, red+blue makes purple, and red+green makes brown. If you revisited this topic at a later age, you may have learned that the proper *subtractive* primaries are actually cyan (a light blue-green), magenta (pink), and yellow (Figure 2.27b), although black is also often used in four-color printing (CMYK). (If you ever subsequently took any painting classes, you learned that colors can have even more fanciful names, such as alizarin crimson, cerulean blue, and chartreuse.) The subtractive colors are called subtractive because pigments in the paint absorb certain wavelengths in the color spectrum.



(a) (b)

Figure 2.27 Primary and secondary colors: (a) additive colors red, green, and blue can be mixed to produce cyan, magenta, yellow, and white; (b) subtractive colors cyan, magenta, and yellow can be mixed to produce red, green, blue, and black.

Later on, you may have learned about the *additive* primary colors (red, green, and blue) and how they can be added (with a slide projector or on a computer monitor) to produce cyan, magenta, yellow, white, and all the other colors we typically see on our TV sets and monitors (Figure 2.27a).

Through what process is it possible for two different colors, such as red and green, to interact to produce a third color like yellow? Are the wavelengths somehow mixed up to produce a new wavelength?

You probably know that the correct answer has nothing to do with physically mixing wavelengths. Instead, the existence of three primaries is a result of the *tri-stimulus* (or *tri-chromatic*) nature of the human visual system, since we have three different kinds of cone, each of which responds selectively to a different portion of the color spectrum (Glassner 1995; Wyszecki and Stiles 2000; Fairchild 2005; Reinhard, Ward, Pattanaik *et al.* 2005; Livingstone 2008).[18] Note that for machine vision applications, such as remote sensing and terrain classification, it is preferable to use many more wavelengths. Similarly, surveillance applications can often benefit from sensing in the near-infrared (NIR) range.

CIE RGB and XYZ

To test and quantify the tri-chromatic theory of perception, we can attempt to reproduce all *monochromatic* (single wavelength) colors as a mixture of three suitably chosen primaries. (Pure wavelength light can be obtained using either a prism or specially manufactured color filters.) In the 1930s, the Commission Internationale d'Eclairage (CIE) standardized the RGB representation by performing such *color matching* experiments using the primary colors of red (700.0nm wavelength), green (546.1nm), and blue (435.8nm).

Figure 2.28 shows the results of performing these experiments with a *standard observer*, i.e., averaging perceptual results over a large number of subjects. You will notice that for certain pure spectra in the blue–green range, a *negative* amount of red light has to be added, i.e., a certain amount of red has to be added to the color being matched in order to get a color match. These results also provided a simple explanation for the existence of *metamers*, which are colors with different spectra that are perceptually indistinguishable. Note that two fabrics

[18] See also Mark Fairchild's Web page, http://www.cis.rit.edu/fairchild/WhyIsColor/books_links.html.

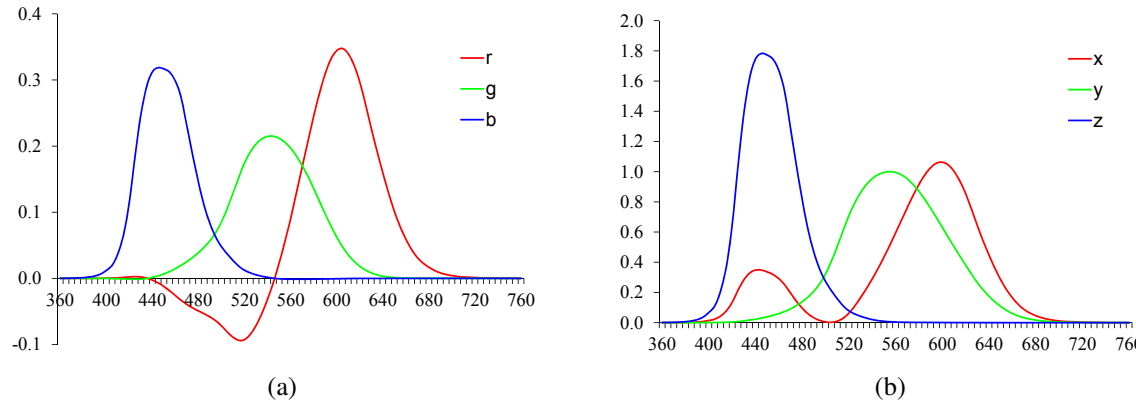


Figure 2.28 Standard CIE color matching functions: (a) $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, $\bar{b}(\lambda)$ color spectra obtained from matching pure colors to the R=700.0nm, G=546.1nm, and B=435.8nm primaries; (b) $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ color matching functions, which are linear combinations of the $(\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda))$ spectra.

or paint colors that are metamers under one light may no longer be so under different lighting.

Because of the problem associated with mixing negative light, the CIE also developed a new color space called XYZ, which contains all of the pure spectral colors within its positive octant. (It also maps the Y axis to the *luminance*, i.e., perceived relative brightness, and maps pure white to a diagonal (equal-valued) vector.) The transformation from RGB to XYZ is given by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{2.103}$$

While the official definition of the CIE XYZ standard has the matrix normalized so that the Y value corresponding to pure red is 1, a more commonly used form is to omit the leading fraction, so that the second row adds up to one, i.e., the RGB triplet $(1, 1, 1)$ maps to a Y value of 1. Linearly blending the $(\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda))$ curves in Figure 2.28a according to (2.103), we obtain the resulting $(\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda))$ curves shown in Figure 2.28b. Notice how all three spectra (color matching functions) now have only positive values and how the $\bar{y}(\lambda)$ curve matches that of the luminance perceived by humans.

If we divide the XYZ values by the sum of X+Y+Z, we obtain the *chromaticity coordinates*

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z}, \tag{2.104}$$

which sum up to 1. The chromaticity coordinates discard the absolute intensity of a given color sample and just represent its pure color. If we sweep the monochromatic color λ parameter in Figure 2.28b from $\lambda = 380$nm to $\lambda = 800$nm, we obtain the familiar *chromaticity diagram* shown in Figure 2.29. This figure shows the (x, y) value for every color value perceivable by most humans. (Of course, the CMYK reproduction process in this book does not actually span the whole gamut of perceivable colors.) The outer curved rim represents where

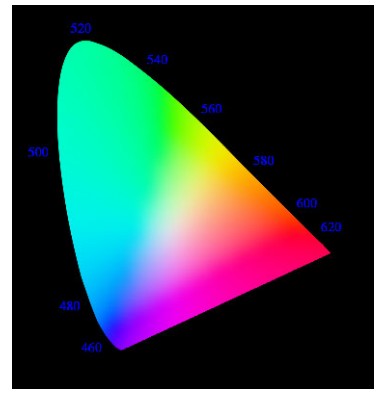


Figure 2.29 CIE chromaticity diagram, showing colors and their corresponding (x, y) values. Pure spectral colors are arranged around the outside of the curve.

all of the pure monochromatic color values map in (x, y) space, while the lower straight line, which connects the two endpoints, is known as the *purple line*.

A convenient representation for color values, when we want to tease apart luminance and chromaticity, is therefore Yxy (luminance plus the two most distinctive chrominance components).

L*a*b* color space

While the XYZ color space has many convenient properties, including the ability to separate luminance from chrominance, it does not actually predict how well humans perceive *differences* in color or luminance.

Because the response of the human visual system is roughly logarithmic (we can perceive *relative* luminance differences of about 1%), the CIE defined a non-linear re-mapping of the XYZ space called L*a*b* (also sometimes called CIELAB), where differences in luminance or chrominance are more perceptually uniform.[19]

The L* component of *lightness* is defined as

$$L^* = 116 f\left(\frac{Y}{Y_n}\right), \tag{2.105}$$

where Y_n is the luminance value for nominal white (Fairchild 2005) and

$$f(t) = \begin{cases} t^{1/3} & t > \delta^3 \\ t/(3\delta^2) + 2\delta/3 & \text{else,} \end{cases} \tag{2.106}$$

is a finite-slope approximation to the cube root with $\delta = 6/29$. The resulting $0 \ldots 100$ scale roughly measures equal amounts of lightness perceptibility.

In a similar fashion, the a* and b* components are defined as

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \quad \text{and} \quad b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right], \tag{2.107}$$

[19] Another perceptually motivated color space called L*u*v* was developed and standardized simultaneously (Fairchild 2005).

where again, (X_n, Y_n, Z_n) is the measured white point. Figure 2.32i–k show the L*a*b* representation for a sample color image.

Color cameras

While the preceding discussion tells us how we can uniquely describe the perceived tri-stimulus description of any color (spectral distribution), it does not tell us how RGB still and video cameras actually work. Do they just measure the amount of light at the nominal wavelengths of red (700.0nm), green (546.1nm), and blue (435.8nm)? Do color monitors just emit exactly these wavelengths and, if so, how can they emit negative red light to reproduce colors in the cyan range?

In fact, the design of RGB video cameras has historically been based around the availability of colored phosphors that go into television sets. When standard-definition color television was invented (NTSC), a mapping was defined between the RGB values that would drive the three color guns in the cathode ray tube (CRT) and the XYZ values that unambiguously define perceived color (this standard was called ITU-R BT.601). With the advent of HDTV and newer monitors, a new standard called ITU-R BT.709 was created, which specifies the XYZ values of each of the color primaries,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R_{709} \\ G_{709} \\ B_{709} \end{bmatrix}. \tag{2.108}$$

In practice, each color camera integrates light according to the *spectral response function* of its red, green, and blue sensors,

$$\begin{aligned} R &= \int L(\lambda) S_R(\lambda) d\lambda, \\ G &= \int L(\lambda) S_G(\lambda) d\lambda, \\ B &= \int L(\lambda) S_B(\lambda) d\lambda, \end{aligned} \tag{2.109}$$

where $L(\lambda)$ is the incoming spectrum of light at a given pixel and $\{S_R(\lambda), S_G(\lambda), S_B(\lambda)\}$ are the red, green, and blue *spectral sensitivities* of the corresponding sensors.

Can we tell what spectral sensitivities the cameras actually have? Unless the camera manufacturer provides us with this data or we observe the response of the camera to a whole spectrum of monochromatic lights, these sensitivities are *not* specified by a standard such as BT.709. Instead, all that matters is that the tri-stimulus values for a given color produce the specified RGB values. The manufacturer is free to use sensors with sensitivities that do not match the standard XYZ definitions, so long as they can later be converted (through a linear transform) to the standard colors.

Similarly, while TV and computer monitors are supposed to produce RGB values as specified by Equation (2.108), there is no reason that they cannot use digital logic to transform the incoming RGB values into different signals to drive each of the color channels. Properly calibrated monitors make this information available to software applications that perform *color management*, so that colors in real life, on the screen, and on the printer all match as closely as possible.

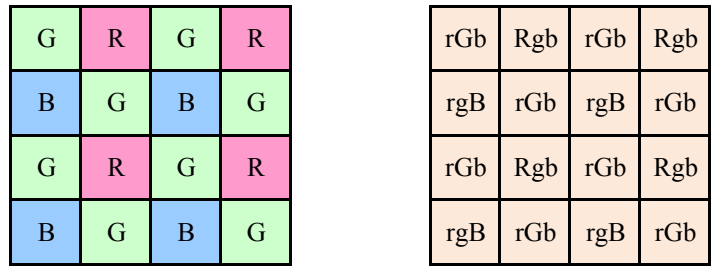


Figure 2.30 Bayer RGB pattern: (a) color filter array layout; (b) interpolated pixel values, with unknown (guessed) values shown as lower case.

Color filter arrays

While early color TV cameras used three *vidicons* (tubes) to perform their sensing and later cameras used three separate RGB sensing chips, most of today's digital still and video cameras cameras use a *color filter array* (CFA), where alternating sensors are covered by different colored filters.[20]

The most commonly used pattern in color cameras today is the *Bayer pattern* (Bayer 1976), which places green filters over half of the sensors (in a checkerboard pattern), and red and blue filters over the remaining ones (Figure 2.30). The reason that there are twice as many green filters as red and blue is because the luminance signal is mostly determined by green values and the visual system is much more sensitive to high frequency detail in luminance than in chrominance (a fact that is exploited in color image compression—see Section 2.3.3). The process of *interpolating* the missing color values so that we have valid RGB values for all the pixels is known as *demosaicing* and is covered in detail in Section 10.3.1.

Similarly, color LCD monitors typically use alternating stripes of red, green, and blue filters placed in front of each liquid crystal active area to simulate the experience of a full color display. As before, because the visual system has higher resolution (acuity) in luminance than chrominance, it is possible to digitally pre-filter RGB (and monochrome) images to enhance the perception of crispness (Betrisey, Blinn, Dresevic *et al.* 2000; Platt 2000).

Color balance

Before encoding the sensed RGB values, most cameras perform some kind of *color balancing* operation in an attempt to move the white point of a given image closer to pure white (equal RGB values). If the color system and the illumination are the same (the BT.709 system uses the daylight illuminant D_{65} as its reference white), the change may be minimal. However, if the illuminant is strongly colored, such as incandescent indoor lighting (which generally results in a yellow or orange hue), the compensation can be quite significant.

A simple way to perform color correction is to multiply each of the RGB values by a different factor (i.e., to apply a diagonal matrix transform to the RGB color space). More

[20] A newer chip design by Foveon (http://www.foveon.com) stacks the red, green, and blue sensors beneath each other, but it has not yet gained widespread adoption.

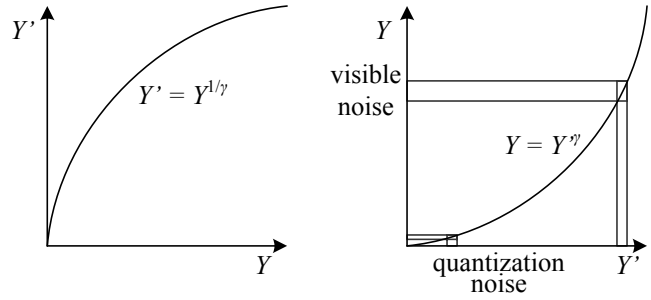


Figure 2.31 Gamma compression: (a) The relationship between the input signal luminance Y and the transmitted signal Y' is given by $Y' = Y^{1/\gamma}$. (b) At the receiver, the signal Y' is exponentiated by the factor γ, $\hat{Y} = Y'^{\gamma}$. Noise introduced during transmission is squashed in the dark regions, which corresponds to the more noise-sensitive region of the visual system.

complicated transforms, which are sometimes the result of mapping to XYZ space and back, actually perform a *color twist*, i.e., they use a general 3×3 color transform matrix.[21] Exercise 2.9 has you explore some of these issues.

Gamma

In the early days of black and white television, the phosphors in the CRT used to display the TV signal responded non-linearly to their input voltage. The relationship between the voltage and the resulting brightness was characterized by a number called *gamma* (γ), since the formula was roughly

$$B = V^{\gamma}, \tag{2.110}$$

with a γ of about 2.2. To compensate for this effect, the electronics in the TV camera would pre-map the sensed luminance Y through an inverse gamma,

$$Y' = Y^{\frac{1}{\gamma}}, \tag{2.111}$$

with a typical value of $\frac{1}{\gamma} = 0.45$.

The mapping of the signal through this non-linearity before transmission had a beneficial side effect: noise added during transmission (remember, these were analog days!) would be reduced (after applying the gamma at the receiver) in the darker regions of the signal where it was more visible (Figure 2.31).[22] (Remember that our visual system is roughly sensitive to relative differences in luminance.)

When color television was invented, it was decided to separately pass the red, green, and blue signals through the same gamma non-linearity before combining them for encoding. Today, even though we no longer have analog noise in our transmission systems, signals are still quantized during compression (see Section 2.3.3), so applying inverse gamma to sensed values is still useful.

[21] Those of you old enough to remember the early days of color television will naturally think of the *hue* adjustment knob on the television set, which could produce truly bizarre results.

[22] A related technique called *companding* was the basis of the Dolby noise reduction systems used with audio tapes.

Unfortunately, for both computer vision and computer graphics, the presence of gamma in images is often problematic. For example, the proper simulation of radiometric phenomena such as shading (see Section 2.2 and Equation (2.87)) occurs in a linear radiance space. Once all of the computations have been performed, the appropriate gamma should be applied before display. Unfortunately, many computer graphics systems (such as shading models) operate directly on RGB values and display these values directly. (Fortunately, newer color imaging standards such as the 16-bit scRGB use a linear space, which makes this less of a problem (Glassner 1995).)

In computer vision, the situation can be even more daunting. The accurate determination of surface normals, using a technique such as photometric stereo (Section 12.1.1) or even a simpler operation such as accurate image deblurring, require that the measurements be in a linear space of intensities. Therefore, it is imperative when performing detailed quantitative computations such as these to first undo the gamma and the per-image color re-balancing in the sensed color values. Chakrabarti, Scharstein, and Zickler (2009) develop a sophisticated 24-parameter model that is a good match to the processing performed by today's digital cameras; they also provide a database of color images you can use for your own testing.[23]

For other vision applications, however, such as feature detection or the matching of signals in stereo and motion estimation, this linearization step is often not necessary. In fact, determining whether it is necessary to undo gamma can take some careful thinking, e.g., in the case of compensating for exposure variations in image stitching (see Exercise 2.7).

If all of these processing steps sound confusing to model, they are. Exercise 2.10 has you try to tease apart some of these phenomena using empirical investigation, i.e., taking pictures of color charts and comparing the RAW and JPEG compressed color values.

Other color spaces

While RGB and XYZ are the primary color spaces used to describe the spectral content (and hence tri-stimulus response) of color signals, a variety of other representations have been developed both in video and still image coding and in computer graphics.

The earliest color representation developed for video transmission was the YIQ standard developed for NTSC video in North America and the closely related YUV standard developed for PAL in Europe. In both of these cases, it was desired to have a *luma* channel Y (so called since it only roughly mimics true luminance) that would be comparable to the regular black-and-white TV signal, along with two lower frequency *chroma* channels.

In both systems, the Y signal (or more appropriately, the Y' luma signal since it is gamma compressed) is obtained from

$$Y'_{601} = 0.299R' + 0.587G' + 0.114B', \tag{2.112}$$

where R'G'B' is the triplet of gamma-compressed color components. When using the newer color definitions for HDTV in BT.709, the formula is

$$Y'_{709} = 0.2125R' + 0.7154G' + 0.0721B'. \tag{2.113}$$

The UV components are derived from scaled versions of $(B'-Y')$ and $(R'-Y')$, namely,

$$U = 0.492111(B' - Y') \text{ and } V = 0.877283(R' - Y'), \tag{2.114}$$

[23] http://vision.middlebury.edu/color/.

whereas the IQ components are the UV components rotated through an angle of 33°. In composite (NTSC and PAL) video, the chroma signals were then low-pass filtered horizontally before being modulated and superimposed on top of the Y' luma signal. Backward compatibility was achieved by having older black-and-white TV sets effectively ignore the high-frequency chroma signal (because of slow electronics) or, at worst, superimposing it as a high-frequency pattern on top of the main signal.

While these conversions were important in the early days of computer vision, when frame grabbers would directly digitize the composite TV signal, today all digital video and still image compression standards are based on the newer YCbCr conversion. YCbCr is closely related to YUV (the C_b and C_r signals carry the blue and red color difference signals and have more useful mnemonics than UV) but uses different scale factors to fit within the eight-bit range available with digital signals.

For video, the Y' signal is re-scaled to fit within the $[16 \dots 235]$ range of values, while the Cb and Cr signals are scaled to fit within $[16 \dots 240]$ (Gomes and Velho 1997; Fairchild 2005). For still images, the JPEG standard uses the full eight-bit range with no reserved values,

$$\begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}, \quad (2.115)$$

where the R'G'B' values are the eight-bit gamma-compressed color components (i.e., the actual RGB values we obtain when we open up or display a JPEG image). For most applications, this formula is not that important, since your image reading software will directly provide you with the eight-bit gamma-compressed R'G'B' values. However, if you are trying to do careful image deblocking (Exercise 3.30), this information may be useful.

Another color space you may come across is *hue, saturation, value* (HSV), which is a projection of the RGB color cube onto a non-linear chroma angle, a radial saturation percentage, and a luminance-inspired value. In more detail, value is defined as either the mean or maximum color value, saturation is defined as scaled distance from the diagonal, and hue is defined as the direction around a color wheel (the exact formulas are described by Hall (1989); Foley, van Dam, Feiner *et al.* (1995)). Such a decomposition is quite natural in graphics applications such as color picking (it approximates the Munsell chart for color description). Figure 2.32l–n shows an HSV representation of a sample color image, where saturation is encoded using a gray scale (saturated = darker) and hue is depicted as a color.

If you want your computer vision algorithm to only affect the value (luminance) of an image and not its saturation or hue, a simpler solution is to use either the Yxy (luminance + chromaticity) coordinates defined in (2.104) or the even simpler *color ratios*,

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \quad (2.116)$$

(Figure 2.32e–h). After manipulating the luma (2.112), e.g., through the process of histogram equalization (Section 3.1.4), you can multiply each color ratio by the ratio of the new to old luma to obtain an adjusted RGB triplet.

While all of these color systems may sound confusing, in the end, it often may not matter that much which one you use. Poynton, in his *Color FAQ*, http://www.poynton.com/ColorFAQ.html, notes that the perceptually motivated L*a*b* system is qualitatively similar

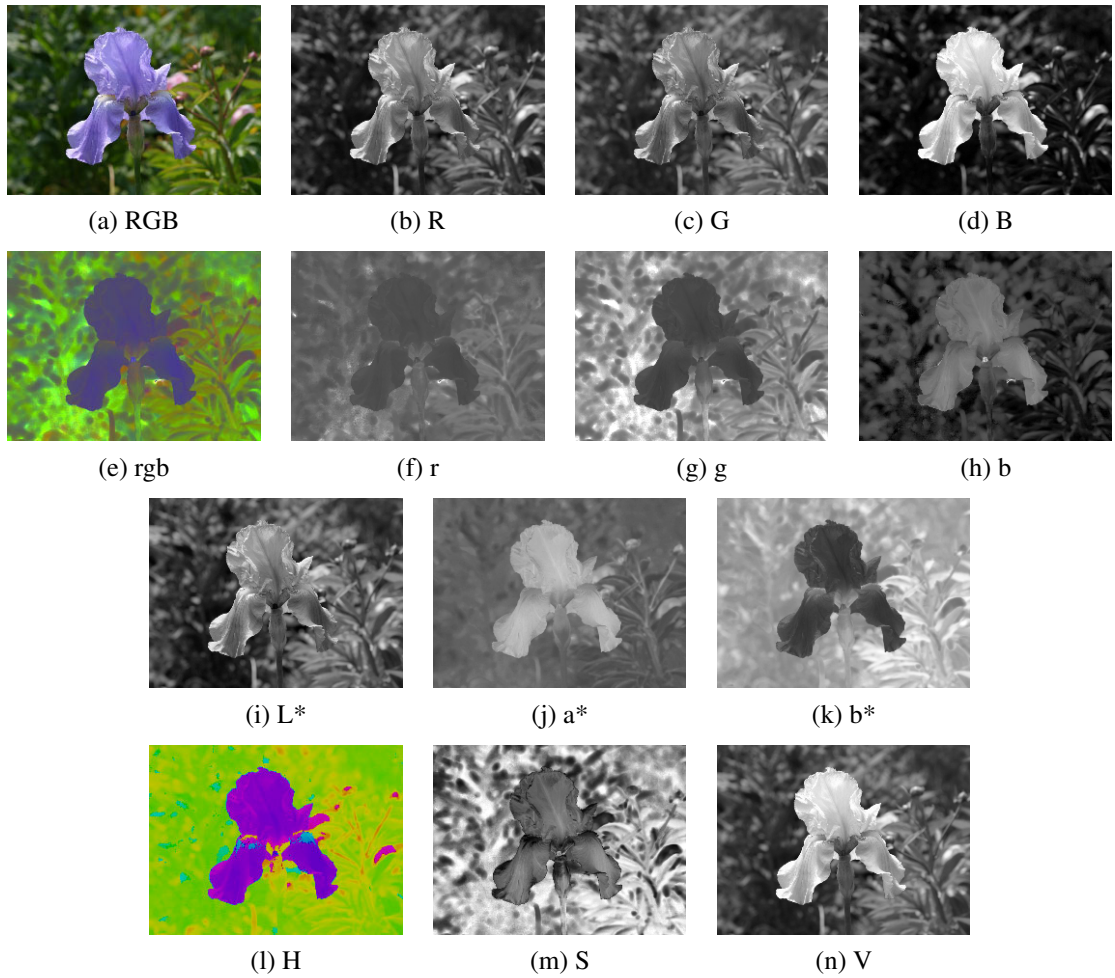


Figure 2.32 Color space transformations: (a–d) RGB; (e–h) rgb. (i–k) L*a*b*; (l–n) HSV. Note that the rgb, L*a*b*, and HSV values are all re-scaled to fit the dynamic range of the printed page.

to the gamma-compressed R'G'B' system we mostly deal with, since both have a fractional power scaling (which approximates a logarithmic response) between the actual intensity values and the numbers being manipulated. As in all cases, think carefully about what you are trying to accomplish before deciding on a technique to use.[24]

2.3.3 Compression

The last stage in a camera's processing pipeline is usually some form of image compression (unless you are using a lossless compression scheme such as camera RAW or PNG).

All color video and image compression algorithms start by converting the signal into YCbCr (or some closely related variant), so that they can compress the luminance signal with

[24] If you are at a loss for questions at a conference, you can always ask why the speaker did not use a perceptual color space, such as L*a*b*. Conversely, if they did use L*a*b*, you can ask if they have any concrete evidence that this works better than regular colors.



Figure 2.33 Image compressed with JPEG at three quality settings. Note how the amount of block artifact and high-frequency aliasing ("mosquito noise") increases from left to right.

higher fidelity than the chrominance signal. (Recall that the human visual system has poorer frequency response to color than to luminance changes.) In video, it is common to subsample Cb and Cr by a factor of two horizontally; with still images (JPEG), the subsampling (averaging) occurs both horizontally and vertically.

Once the luminance and chrominance images have been appropriately subsampled and separated into individual images, they are then passed to a *block transform* stage. The most common technique used here is the *discrete cosine transform* (DCT), which is a real-valued variant of the discrete Fourier transform (DFT) (see Section 3.4.3). The DCT is a reasonable approximation to the Karhunen–Loève or eigenvalue decomposition of natural image patches, i.e., the decomposition that simultaneously packs the most energy into the first coefficients and diagonalizes the joint covariance matrix among the pixels (makes transform coefficients statistically independent). Both MPEG and JPEG use 8×8 DCT transforms (Wallace 1991; Le Gall 1991), although newer variants use smaller 4×4 blocks or alternative transformations, such as wavelets (Taubman and Marcellin 2002) and lapped transforms (Malvar 1990, 1998, 2000) are now used.

After transform coding, the coefficient values are quantized into a set of small integer values that can be coded using a variable bit length scheme such as a Huffman code or an arithmetic code (Wallace 1991). (The DC (lowest frequency) coefficients are also adaptively predicted from the previous block's DC values. The term "DC" comes from "direct current", i.e., the non-sinusoidal or non-alternating part of a signal.) The step size in the quantization is the main variable controlled by the *quality* setting on the JPEG file (Figure 2.33).

With video, it is also usual to perform block-based *motion compensation*, i.e., to encode the difference between each block and a *predicted* set of pixel values obtained from a shifted block in the previous frame. (The exception is the *motion-JPEG* scheme used in older DV camcorders, which is nothing more than a series of individually JPEG compressed image frames.) While basic MPEG uses 16×16 motion compensation blocks with integer motion values (Le Gall 1991), newer standards use adaptively sized block, sub-pixel motions, and the ability to reference blocks from older frames. In order to recover more gracefully from failures and to allow for random access to the video stream, predicted P frames are interleaved among independently coded I frames. (Bi-directional B frames are also sometimes used.)

The quality of a compression algorithm is usually reported using its *peak signal-to-noise*

ratio (PSNR), which is derived from the average *mean square error*,

$$MSE = \frac{1}{n}\sum_{\boldsymbol{x}}\left[I(\boldsymbol{x}) - \hat{I}(\boldsymbol{x})\right]^2, \tag{2.117}$$

where $I(\boldsymbol{x})$ is the original uncompressed image and $\hat{I}(\boldsymbol{x})$ is its compressed counterpart, or equivalently, the *root mean square error* (RMS error), which is defined as

$$RMS = \sqrt{MSE}. \tag{2.118}$$

The PSNR is defined as

$$PSNR = 10\log_{10}\frac{I_{\max}^2}{MSE} = 20\log_{10}\frac{I_{\max}}{RMS}, \tag{2.119}$$

where $I_{\max}$ is the maximum signal extent, e.g., 255 for eight-bit images.

While this is just a high-level sketch of how image compression works, it is useful to understand so that the artifacts introduced by such techniques can be compensated for in various computer vision applications.

2.4 Additional reading

As we mentioned at the beginning of this chapter, it provides but a brief summary of a very rich and deep set of topics, traditionally covered in a number of separate fields.

A more thorough introduction to the geometry of points, lines, planes, and projections can be found in textbooks on multi-view geometry (Hartley and Zisserman 2004; Faugeras and Luong 2001) and computer graphics (Foley, van Dam, Feiner *et al.* 1995; Watt 1995; OpenGL-ARB 1997). Topics covered in more depth include higher-order primitives such as quadrics, conics, and cubics, as well as three-view and multi-view geometry.

The image formation (synthesis) process is traditionally taught as part of a computer graphics curriculum (Foley, van Dam, Feiner *et al.* 1995; Glassner 1995; Watt 1995; Shirley 2005) but it is also studied in physics-based computer vision (Wolff, Shafer, and Healey 1992a).

The behavior of camera lens systems is studied in optics (Möller 1988; Hecht 2001; Ray 2002).

Some good books on color theory have been written by Healey and Shafer (1992); Wyszecki and Stiles (2000); Fairchild (2005), with Livingstone (2008) providing a more fun and informal introduction to the topic of color perception. Mark Fairchild's page of color books and links[25] lists many other sources.

Topics relating to sampling and aliasing are covered in textbooks on signal and image processing (Crane 1997; Jähne 1997; Oppenheim and Schafer 1996; Oppenheim, Schafer, and Buck 1999; Pratt 2007; Russ 2007; Burger and Burge 2008; Gonzales and Woods 2008).

2.5 Exercises

A note to students: This chapter is relatively light on exercises since it contains mostly background material and not that many usable techniques. If you really want to understand

[25] http://www.cis.rit.edu/fairchild/WhyIsColor/books_links.html.

multi-view geometry in a thorough way, I encourage you to read and do the exercises provided by Hartley and Zisserman (2004). Similarly, if you want some exercises related to the image formation process, Glassner's (1995) book is full of challenging problems.

Ex 2.1: Least squares intersection point and line fitting—advanced Equation (2.4) shows how the intersection of two 2D lines can be expressed as their cross product, assuming the lines are expressed as homogeneous coordinates.

1. If you are given more than two lines and want to find a point $\tilde{\boldsymbol{x}}$ that minimizes the sum of squared distances to each line,

$$D = \sum_i (\tilde{\boldsymbol{x}} \cdot \tilde{\boldsymbol{l}}_i)^2, \tag{2.120}$$

   how can you compute this quantity? (Hint: Write the dot product as $\tilde{\boldsymbol{x}}^T \tilde{\boldsymbol{l}}_i$ and turn the squared quantity into a *quadratic form*, $\tilde{\boldsymbol{x}}^T \boldsymbol{A} \tilde{\boldsymbol{x}}$.)

2. To fit a line to a bunch of points, you can compute the *centroid* (mean) of the points as well as the *covariance matrix* of the points around this mean. Show that the line passing through the centroid along the major axis of the covariance ellipsoid (largest eigenvector) minimizes the sum of squared distances to the points.

3. These two approaches are fundamentally different, even though projective duality tells us that points and lines are interchangeable. Why are these two algorithms so apparently different? Are they actually minimizing different objectives?

Ex 2.2: 2D transform editor Write a program that lets you interactively create a set of rectangles and then modify their "pose" (2D transform). You should implement the following steps:

1. Open an empty window ("canvas").
2. Shift drag (rubber-band) to create a new rectangle.
3. Select the deformation mode (motion model): translation, rigid, similarity, affine, or perspective.
4. Drag any corner of the outline to change its transformation.

This exercise should be built on a set of pixel coordinate and transformation classes, either implemented by yourself or from a software library. Persistence of the created representation (save and load) should also be supported (for each rectangle, save its transformation).

Ex 2.3: 3D viewer Write a simple viewer for 3D points, lines, and polygons. Import a set of point and line commands (primitives) as well as a viewing transform. Interactively modify the object or camera transform. This viewer can be an extension of the one you created in (Exercise 2.2). Simply replace the viewing transformations with their 3D equivalents.

(Optional) Add a z-buffer to do hidden surface removal for polygons.

(Optional) Use a 3D drawing package and just write the viewer control.

Ex 2.4: Focus distance and depth of field Figure out how the focus distance and depth of field indicators on a lens are determined.

1. Compute and plot the focus distance z_o as a function of the distance traveled from the focal length $\Delta z_i = f - z_i$ for a lens of focal length f (say, 100mm). Does this explain the hyperbolic progression of focus distances you see on a typical lens (Figure 2.20)?

2. Compute the depth of field (minimum and maximum focus distances) for a given focus setting z_o as a function of the circle of confusion diameter c (make it a fraction of the sensor width), the focal length f, and the f-stop number N (which relates to the aperture diameter d). Does this explain the usual depth of field markings on a lens that bracket the in-focus marker, as in Figure 2.20a?

3. Now consider a zoom lens with a varying focal length f. Assume that as you zoom, the lens stays in focus, i.e., the distance from the rear nodal point to the sensor plane z_i adjusts itself automatically for a fixed focus distance z_o. How do the depth of field indicators vary as a function of focal length? Can you reproduce a two-dimensional plot that mimics the curved depth of field lines seen on the lens in Figure 2.20b?

Ex 2.5: F-numbers and shutter speeds List the common f-numbers and shutter speeds that your camera provides. On older model SLRs, they are visible on the lens and shutter speed dials. On newer cameras, you have to look at the electronic viewfinder (or LCD screen/indicator) as you manually adjust exposures.

1. Do these form geometric progressions; if so, what are the ratios? How do these relate to exposure values (EVs)?

2. If your camera has shutter speeds of $\frac{1}{60}$ and $\frac{1}{125}$, do you think that these two speeds are exactly a factor of two apart or a factor of $125/60 = 2.083$ apart?

3. How accurate do you think these numbers are? Can you devise some way to measure exactly how the aperture affects how much light reaches the sensor and what the exact exposure times actually are?

Ex 2.6: Noise level calibration Estimate the amount of noise in your camera by taking repeated shots of a scene with the camera mounted on a tripod. (Purchasing a remote shutter release is a good investment if you own a DSLR.) Alternatively, take a scene with constant color regions (such as a color checker chart) and estimate the variance by fitting a smooth function to each color region and then taking differences from the predicted function.

1. Plot your estimated variance as a function of level for each of your color channels separately.

2. Change the ISO setting on your camera; if you cannot do that, reduce the overall light in your scene (turn off lights, draw the curtains, wait until dusk). Does the amount of noise vary a lot with ISO/gain?

3. Compare your camera to another one at a different price point or year of make. Is there evidence to suggest that "you get what you pay for"? Does the quality of digital cameras seem to be improving over time?

Ex 2.7: Gamma correction in image stitching Here's a relatively simple puzzle. Assume you are given two images that are part of a panorama that you want to stitch (see Chapter 9). The two images were taken with different exposures, so you want to adjust the RGB values so that they match along the seam line. Is it necessary to undo the gamma in the color values in order to achieve this?

Ex 2.8: Skin color detection Devise a simple skin color detector (Forsyth and Fleck 1999; Jones and Rehg 2001; Vezhnevets, Sazonov, and Andreeva 2003; Kakumanu, Makrogiannis, and Bourbakis 2007) based on chromaticity or other color properties.

1. Take a variety of photographs of people and calculate the *xy chromaticity values* for each pixel.
2. Crop the photos or otherwise indicate with a painting tool which pixels are likely to be skin (e.g. face and arms).
3. Calculate a color (chromaticity) distribution for these pixels. You can use something as simple as a mean and covariance measure or as complicated as a mean-shift segmentation algorithm (see Section 5.3.2). You can optionally use non-skin pixels to model the *background distribution*.
4. Use your computed distribution to find the skin regions in an image. One easy way to visualize this is to paint all non-skin pixels a given color, such as white or black.
5. How sensitive is your algorithm to color balance (scene lighting)?
6. Does a simpler chromaticity measurement, such as a color ratio (2.116), work just as well?

Ex 2.9: White point balancing—tricky A common (in-camera or post-processing) technique for performing white point adjustment is to take a picture of a white piece of paper and to adjust the RGB values of an image to make this a neutral color.

1. Describe how you would adjust the RGB values in an image given a sample "white color" of (R_w, G_w, B_w) to make this color neutral (without changing the exposure too much).
2. Does your transformation involve a simple (per-channel) scaling of the RGB values or do you need a full 3×3 color twist matrix (or something else)?
3. Convert your RGB values to XYZ. Does the appropriate correction now only depend on the XY (or xy) values? If so, when you convert back to RGB space, do you need a full 3×3 color twist matrix to achieve the same effect?
4. If you used pure diagonal scaling in the direct RGB mode but end up with a twist if you work in XYZ space, how do you explain this apparent dichotomy? Which approach is correct? (Or is it possible that neither approach is actually correct?)

If you want to find out what your camera *actually* does, continue on to the next exercise.

Ex 2.10: In-camera color processing—challenging If your camera supports a RAW pixel mode, take a pair of RAW and JPEG images, and see if you can infer what the camera is doing when it converts the RAW pixel values to the final color-corrected and gamma-compressed eight-bit JPEG pixel values.

1. Deduce the pattern in your color filter array from the correspondence between co-located RAW and color-mapped pixel values. Use a color checker chart at this stage if it makes your life easier. You may find it helpful to split the RAW image into four separate images (subsampling even and odd columns and rows) and to treat each of these new images as a "virtual" sensor.

2. Evaluate the quality of the demosaicing algorithm by taking pictures of challenging scenes which contain strong color edges (such as those shown in in Section 10.3.1).

3. If you can take the same exact picture after changing the color balance values in your camera, compare how these settings affect this processing.

4. Compare your results against those presented by Chakrabarti, Scharstein, and Zickler (2009) or use the data available in their database of color images.[26]

[26] http://vision.middlebury.edu/color/.

Chapter 3

Image processing

3.1 Point operators . . . 89
3.1.1 Pixel transforms . . . 91
3.1.2 Color transforms . . . 92
3.1.3 Compositing and matting . . . 92
3.1.4 Histogram equalization . . . 94
3.1.5 *Application*: Tonal adjustment . . . 97
3.2 Linear filtering . . . 98
3.2.1 Separable filtering . . . 102
3.2.2 Examples of linear filtering . . . 103
3.2.3 Band-pass and steerable filters . . . 104
3.3 More neighborhood operators . . . 108
3.3.1 Non-linear filtering . . . 108
3.3.2 Morphology . . . 112
3.3.3 Distance transforms . . . 113
3.3.4 Connected components . . . 115
3.4 Fourier transforms . . . 116
3.4.1 Fourier transform pairs . . . 119
3.4.2 Two-dimensional Fourier transforms . . . 123
3.4.3 Wiener filtering . . . 123
3.4.4 *Application*: Sharpening, blur, and noise removal . . . 126
3.5 Pyramids and wavelets . . . 127
3.5.1 Interpolation . . . 127
3.5.2 Decimation . . . 130
3.5.3 Multi-resolution representations . . . 132
3.5.4 Wavelets . . . 136
3.5.5 *Application*: Image blending . . . 140
3.6 Geometric transformations . . . 143
3.6.1 Parametric transformations . . . 145
3.6.2 Mesh-based warping . . . 149
3.6.3 *Application*: Feature-based morphing . . . 152
3.7 Global optimization . . . 153
3.7.1 Regularization . . . 154
3.7.2 Markov random fields . . . 158
3.7.3 *Application*: Image restoration . . . 169
3.8 Additional reading . . . 169
3.9 Exercises . . . 171

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_3, © Springer-Verlag London Limited 2011



(a) (b) (c) (d) (e) (f)

Figure 3.1 Some common image processing operations: (a) original image; (b) increased contrast; (c) change in hue; (d) "posterized" (quantized colors); (e) blurred; (f) rotated.

Now that we have seen how images are formed through the interaction of 3D scene elements, lighting, and camera optics and sensors, let us look at the first stage in most computer vision applications, namely the use of image processing to preprocess the image and convert it into a form suitable for further analysis. Examples of such operations include exposure correction and color balancing, the reduction of image noise, increasing sharpness, or straightening the image by rotating it (Figure 3.1). While some may consider image processing to be outside the purview of computer vision, most computer vision applications, such as computational photography and even recognition, require care in designing the image processing stages in order to achieve acceptable results.

In this chapter, we review standard image processing operators that map pixel values from one image to another. Image processing is often taught in electrical engineering departments as a follow-on course to an introductory course in signal processing (Oppenheim and Schafer 1996; Oppenheim, Schafer, and Buck 1999). There are several popular textbooks for image processing (Crane 1997; Gomes and Velho 1997; Jähne 1997; Pratt 2007; Russ 2007; Burger and Burge 2008; Gonzales and Woods 2008).

We begin this chapter with the simplest kind of image transforms, namely those that manipulate each pixel independently of its neighbors (Section 3.1). Such transforms are often called *point operators* or *point processes*. Next, we examine *neighborhood* (area-based) operators, where each new pixel's value depends on a small number of neighboring input values (Sections 3.2 and 3.3). A convenient tool to analyze (and sometimes accelerate) such neighborhood operations is the *Fourier Transform*, which we cover in Section 3.4. Neighborhood operators can be cascaded to form *image pyramids* and *wavelets*, which are useful for analyzing images at a variety of resolutions (scales) and for accelerating certain operations (Section 3.5). Another important class of global operators are *geometric transformations*, such as rotations, shears, and perspective deformations (Section 3.6). Finally, we introduce *global optimization* approaches to image processing, which involve the minimization of an energy functional or, equivalently, optimal estimation using Bayesian *Markov random field* models (Section 3.7).

3.1 Point operators

The simplest kinds of image processing transforms are *point operators*, where each output pixel's value depends on only the corresponding input pixel value (plus, potentially, some globally collected information or parameters). Examples of such operators include brightness and contrast adjustments (Figure 3.2) as well as color correction and transformations. In the image processing literature, such operations are also known as *point processes* (Crane 1997).

We begin this section with a quick review of simple point operators such as brightness scaling and image addition. Next, we discuss how colors in images can be manipulated. We then present *image compositing* and *matting* operations, which play an important role in computational photography (Chapter 10) and computer graphics applications. Finally, we describe the more global process of *histogram equalization*. We close with an example application that manipulates *tonal values* (exposure and contrast) to improve image appearance.

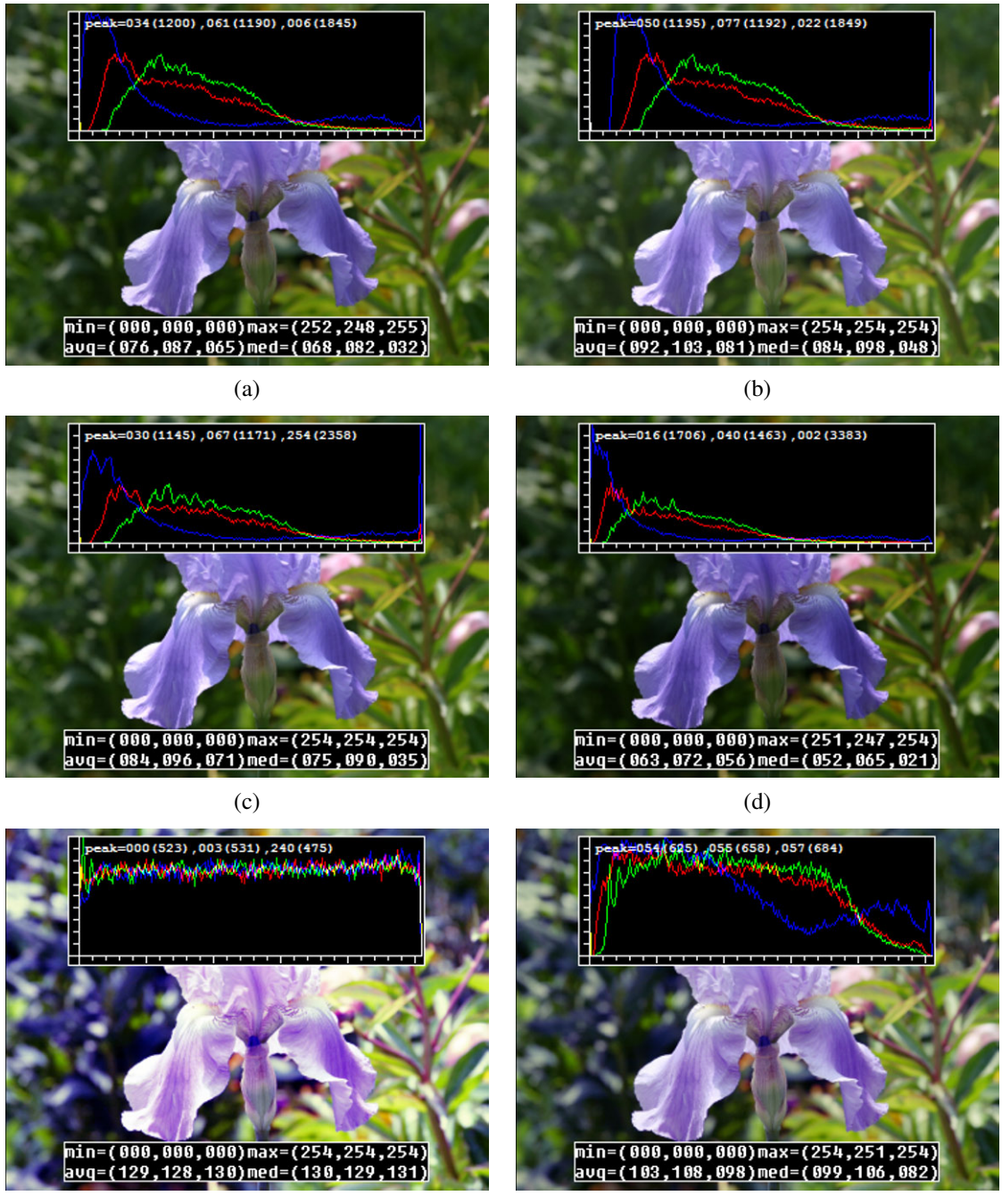
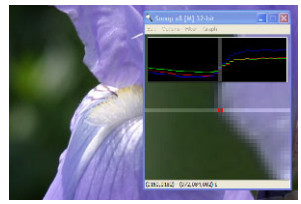


Figure 3.2 Some local image processing operations: (a) original image along with its three color (per-channel) histograms; (b) brightness increased (additive offset, $b = 16$); (c) contrast increased (multiplicative gain, $a = 1.1$); (d) gamma (partially) linearized ($\gamma = 1.2$); (e) full histogram equalization; (f) partial histogram equalization.



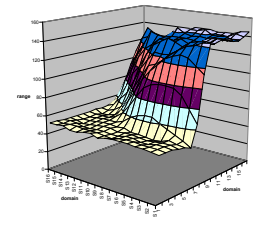
(a)



(b)

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

(c)



(d)

Figure 3.3 Visualizing image data: (a) original image; (b) cropped portion and scanline plot using an image inspection tool; (c) grid of numbers; (d) surface plot. For figures (c)–(d), the image was first converted to grayscale.

3.1.1 Pixel transforms

A general image processing *operator* is a function that takes one or more input images and produces an output image. In the continuous domain, this can be denoted as

$$g(\boldsymbol{x}) = h(f(\boldsymbol{x})) \text{ or } g(\boldsymbol{x}) = h(f_0(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})), \tag{3.1}$$

where $\boldsymbol{x}$ is in the D-dimensional *domain* of the functions (usually $D = 2$ for images) and the functions f and g operate over some *range*, which can either be scalar or vector-valued, e.g., for color images or 2D motion. For discrete (sampled) images, the domain consists of a finite number of *pixel locations*, $\boldsymbol{x} = (i, j)$, and we can write

$$g(i, j) = h(f(i, j)). \tag{3.2}$$

Figure 3.3 shows how an image can be represented either by its color (appearance), as a grid of numbers, or as a two-dimensional function (surface plot).

Two commonly used point processes are multiplication and addition with a constant,

$$g(\boldsymbol{x}) = af(\boldsymbol{x}) + b. \tag{3.3}$$

The parameters $a > 0$ and b are often called the *gain* and *bias* parameters; sometimes these parameters are said to control *contrast* and *brightness*, respectively (Figures 3.2b–c).[1] The bias and gain parameters can also be spatially varying,

$$g(\boldsymbol{x}) = a(\boldsymbol{x})f(\boldsymbol{x}) + b(\boldsymbol{x}), \tag{3.4}$$

e.g., when simulating the *graded density filter* used by photographers to selectively darken the sky or when modeling vignetting in an optical system.

Multiplicative gain (both global and spatially varying) is a *linear* operation, since it obeys the *superposition principle*,

$$h(f_0 + f_1) = h(f_0) + h(f_1). \tag{3.5}$$

(We will have more to say about linear shift invariant operators in Section 3.2.) Operators such as image squaring (which is often used to get a local estimate of the *energy* in a band-pass filtered signal, see Section 3.5) are not linear.

[1] An image's luminance characteristics can also be summarized by its *key* (average luminanance) and *range* (Kopf, Uyttendaele, Deussen *et al.* 2007).

Another commonly used *dyadic* (two-input) operator is the *linear blend* operator,

$$g(\boldsymbol{x}) = (1 - \alpha) f_0(\boldsymbol{x}) + \alpha f_1(\boldsymbol{x}). \tag{3.6}$$

By varying α from $0 \rightarrow 1$, this operator can be used to perform a temporal *cross-dissolve* between two images or videos, as seen in slide shows and film production, or as a component of image *morphing* algorithms (Section 3.6.3).

One highly used non-linear transform that is often applied to images before further processing is *gamma correction*, which is used to remove the non-linear mapping between input radiance and quantized pixel values (Section 2.3.2). To invert the gamma mapping applied by the sensor, we can use

$$g(\boldsymbol{x}) = [f(\boldsymbol{x})]^{1/\gamma}, \tag{3.7}$$

where a gamma value of $\gamma \approx 2.2$ is a reasonable fit for most digital cameras.

3.1.2 Color transforms

While color images can be treated as arbitrary vector-valued functions or collections of independent bands, it usually makes sense to think about them as highly correlated signals with strong connections to the image formation process (Section 2.2), sensor design (Section 2.3), and human perception (Section 2.3.2). Consider, for example, brightening a picture by adding a constant value to all three channels, as shown in Figure 3.2b. Can you tell if this achieves the desired effect of making the image look brighter? Can you see any undesirable side-effects or artifacts?

In fact, adding the same value to each color channel not only increases the apparent *intensity* of each pixel, it can also affect the pixel's *hue* and *saturation*. How can we define and manipulate such quantities in order to achieve the desired perceptual effects?

As discussed in Section 2.3.2, chromaticity coordinates (2.104) or even simpler color ratios (2.116) can first be computed and then used after manipulating (e.g., brightening) the luminance Y to re-compute a valid RGB image with the same hue and saturation. Figure 2.32g–i shows some color ratio images multiplied by the middle gray value for better visualization.

Similarly, color balancing (e.g., to compensate for incandescent lighting) can be performed either by multiplying each channel with a different scale factor or by the more complex process of mapping to XYZ color space, changing the nominal white point, and mapping back to RGB, which can be written down using a linear 3×3 *color twist* transform matrix. Exercises 2.9 and 3.1 have you explore some of these issues.

Another fun project, best attempted after you have mastered the rest of the material in this chapter, is to take a picture with a rainbow in it and enhance the strength of the rainbow (Exercise 3.29).

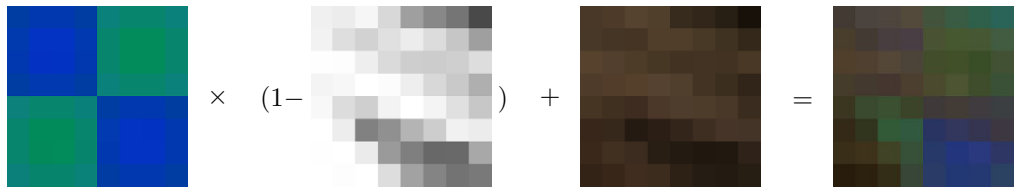
3.1.3 Compositing and matting

In many photo editing and visual effects applications, it is often desirable to cut a *foreground* object out of one scene and put it on top of a different *background* (Figure 3.4). The process of extracting the object from the original image is often called *matting* (Smith and Blinn



(a) (b) (c) (d)

Figure 3.4 Image matting and compositing (Chuang, Curless, Salesin *et al.* 2001) © 2001 IEEE: (a) source image; (b) extracted foreground object F; (c) alpha matte α shown in grayscale; (d) new composite C.



B α αF C

(a) (b) (c) (d)

Figure 3.5 Compositing equation $C = (1 - \alpha)B + \alpha F$. The images are taken from a close-up of the region of the hair in the upper right part of the lion in Figure 3.4.

1996), while the process of inserting it into another image (without visible artifacts) is called *compositing* (Porter and Duff 1984; Blinn 1994a).

The intermediate representation used for the foreground object between these two stages is called an *alpha-matted color image* (Figure 3.4b–c). In addition to the three color RGB channels, an alpha-matted image contains a fourth *alpha* channel α (or A) that describes the relative amount of *opacity* or *fractional coverage* at each pixel (Figures 3.4c and 3.5b). The opacity is the opposite of the *transparency*. Pixels within the object are fully opaque ($\alpha = 1$), while pixels fully outside the object are transparent ($\alpha = 0$). Pixels on the boundary of the object vary smoothly between these two extremes, which hides the perceptual visible *jaggies* that occur if only binary opacities are used.

To composite a new (or foreground) image on top of an old (background) image, the *over operator*, first proposed by Porter and Duff (1984) and then studied extensively by Blinn (1994a; 1994b), is used,

$$C = (1 - \alpha)B + \alpha F. \tag{3.8}$$

This operator *attenuates* the influence of the background image B by a factor $(1 - \alpha)$ and then adds in the color (and opacity) values corresponding to the foreground layer F, as shown in Figure 3.5.

In many situations, it is convenient to represent the foreground colors in *pre-multiplied* form, i.e., to store (and manipulate) the αF values directly. As Blinn (1994b) shows, the pre-multiplied RGBA representation is preferred for several reasons, including the ability to blur or resample (e.g., rotate) alpha-matted images without any additional complications (just treating each RGBA band independently). However, when matting using local color consistency (Ruzon and Tomasi 2000; Chuang, Curless, Salesin *et al.* 2001), the pure un-



Figure 3.6 An example of light reflecting off the transparent glass of a picture frame (Black and Anandan 1996) © 1996 Elsevier. You can clearly see the woman's portrait inside the picture frame superimposed with the reflection of a man's face off the glass.

multiplied foreground colors F are used, since these remain constant (or vary slowly) in the vicinity of the object edge.

The over operation is not the only kind of compositing operation that can be used. Porter and Duff (1984) describe a number of additional operations that can be useful in photo editing and visual effects applications. In this book, we concern ourselves with only one additional, commonly occurring case (but see Exercise 3.2).

When light reflects off clean transparent glass, the light passing through the glass and the light reflecting off the glass are simply added together (Figure 3.6). This model is useful in the analysis of *transparent motion* (Black and Anandan 1996; Szeliski, Avidan, and Anandan 2000), which occurs when such scenes are observed from a moving camera (see Section 8.5.2).

The actual process of *matting*, i.e., recovering the foreground, background, and alpha matte values from one or more images, has a rich history, which we study in Section 10.4. Smith and Blinn (1996) have a nice survey of traditional *blue-screen matting* techniques, while Toyama, Krumm, Brumitt *et al.* (1999) review *difference matting*. More recently, there has been a lot of activity in computational photography relating to *natural image matting* (Ruzon and Tomasi 2000; Chuang, Curless, Salesin *et al.* 2001; Wang and Cohen 2007a), which attempts to extract the mattes from a single natural image (Figure 3.4a) or from extended video sequences (Chuang, Agarwala, Curless *et al.* 2002). All of these techniques are described in more detail in Section 10.4.

3.1.4 Histogram equalization

While the brightness and gain controls described in Section 3.1.1 can improve the appearance of an image, how can we automatically determine their best values? One approach might be to look at the darkest and brightest pixel values in an image and map them to pure black and pure white. Another approach might be to find the *average* value in the image, push it towards middle gray, and expand the *range* so that it more closely fills the displayable values (Kopf, Uyttendaele, Deussen *et al.* 2007).

How can we visualize the set of lightness values in an image in order to test some of

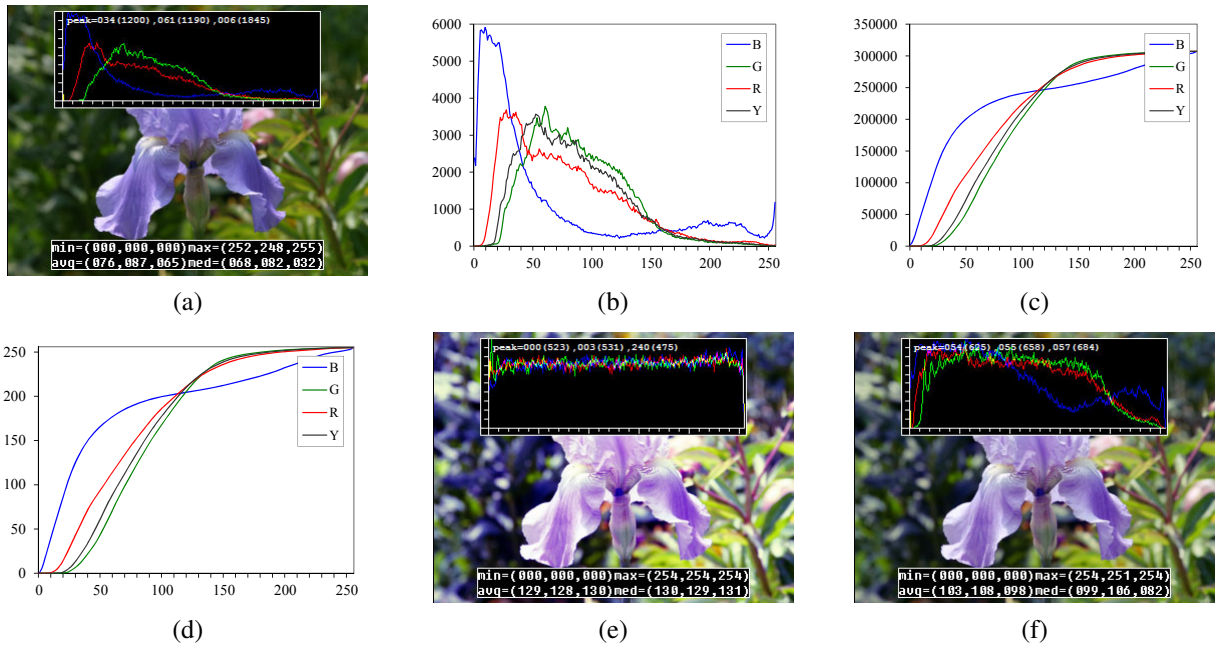


Figure 3.7 Histogram analysis and equalization: (a) original image (b) color channel and intensity (luminance) histograms; (c) cumulative distribution functions; (d) equalization (transfer) functions; (e) full histogram equalization; (f) partial histogram equalization.

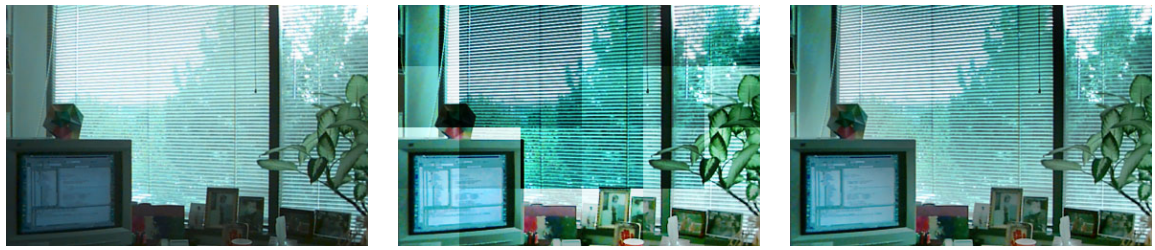
these heuristics? The answer is to plot the *histogram* of the individual color channels and luminance values, as shown in Figure 3.7b.[2] From this distribution, we can compute relevant statistics such as the minimum, maximum, and average intensity values. Notice that the image in Figure 3.7a has both an excess of dark values and light values, but that the mid-range values are largely under-populated. Would it not be better if we could simultaneously brighten some dark values and darken some light values, while still using the full extent of the available dynamic range? Can you think of a mapping that might do this?

One popular answer to this question is to perform *histogram equalization*, i.e., to find an intensity mapping function $f(I)$ such that the resulting histogram is flat. The trick to finding such a mapping is the same one that people use to generate random samples from a *probability density function*, which is to first compute the *cumulative distribution function* shown in Figure 3.7c.

Think of the original histogram $h(I)$ as the distribution of grades in a class after some exam. How can we map a particular grade to its corresponding *percentile*, so that students at the 75% percentile range scored better than $^3/_4$ of their classmates? The answer is to integrate the distribution $h(I)$ to obtain the cumulative distribution $c(I)$,

$$c(I) = \frac{1}{N}\sum_{i=0}^{I} h(i) = c(I-1) + \frac{1}{N}h(I), \tag{3.9}$$

[2] The histogram is simply the *count* of the number of pixels at each gray level value. For an eight-bit image, an accumulation table with 256 entries is needed. For higher bit depths, a table with the appropriate number of entries (probably fewer than the full number of gray levels) should be used.



(a) (b) (c)

Figure 3.8 Locally adaptive histogram equalization: (a) original image; (b) block histogram equalization; (c) full locally adaptive equalization.

where N is the number of pixels in the image or students in the class. For any given grade or intensity, we can look up its corresponding percentile $c(I)$ and determine the final value that pixel should take. When working with eight-bit pixel values, the I and c axes are rescaled from $[0, 255]$.

Figure 3.7d shows the result of applying $f(I) = c(I)$ to the original image. As we can see, the resulting histogram is flat; so is the resulting image (it is "flat" in the sense of a lack of contrast and being muddy looking). One way to compensate for this is to only *partially* compensate for the histogram unevenness, e.g., by using a mapping function $f(I) = \alpha c(I) + (1 - \alpha)I$, which is a linear blend between the cumulative distribution function and the identity transform (a straight line). As you can see in Figure 3.7e, the resulting image maintains more of its original grayscale distribution while having a more appealing balance.

Another potential problem with histogram equalization (or, in general, image brightening) is that noise in dark regions can be amplified and become more visible. Exercise 3.6 suggests some possible ways to mitigate this, as well as alternative techniques to maintain contrast and "punch" in the original images (Larson, Rushmeier, and Piatko 1997; Stark 2000).

Locally adaptive histogram equalization

While global histogram equalization can be useful, for some images it might be preferable to apply different kinds of equalization in different regions. Consider for example the image in Figure 3.8a, which has a wide range of luminance values. Instead of computing a single curve, what if we were to subdivide the image into $M \times M$ pixel blocks and perform separate histogram equalization in each sub-block? As you can see in Figure 3.8b, the resulting image exhibits a lot of blocking artifacts, i.e., intensity discontinuities at block boundaries.

One way to eliminate blocking artifacts is to use a *moving window*, i.e., to recompute the histogram for every $M \times M$ block centered at each pixel. This process can be quite slow (M^2 operations per pixel), although with clever programming only the histogram entries corresponding to the pixels entering and leaving the block (in a raster scan across the image) need to be updated (M operations per pixel). Note that this operation is an example of the *non-linear neighborhood operations* we study in more detail in Section 3.3.1.

A more efficient approach is to compute non-overlapped block-based equalization functions as before, but to then smoothly interpolate the transfer functions as we move between blocks. This technique is known as *adaptive histogram equalization* (AHE) and its contrast-

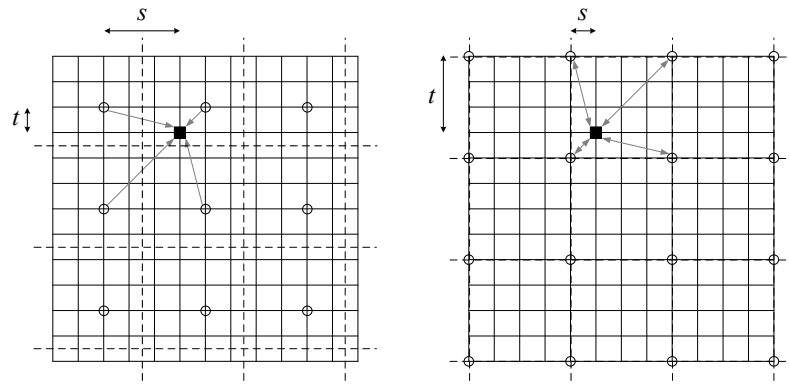


Figure 3.9 Local histogram interpolation using relative (s, t) coordinates: (a) block-based histograms, with block centers shown as circles; (b) corner-based "spline" histograms. Pixels are located on grid intersections. The black square pixel's transfer function is interpolated from the four adjacent lookup tables (gray arrows) using the computed (s, t) values. Block boundaries are shown as dashed lines.

limited (gain-limited) version is known as CLAHE (Pizer, Amburn, Austin *et al.* 1987).[3] The weighting function for a given pixel (i, j) can be computed as a function of its horizontal and vertical position (s, t) within a block, as shown in Figure 3.9a. To blend the four lookup functions $\{f_{00}, \ldots, f_{11}\}$, a *bilinear* blending function,

$$f_{s,t}(I) = (1-s)(1-t)f_{00}(I) + s(1-t)f_{10}(I) + (1-s)tf_{01}(I) + stf_{11}(I) \quad (3.10)$$

can be used. (See Section 3.5.2 for higher-order generalizations of such *spline* functions.) Note that instead of blending the four lookup tables for each output pixel (which would be quite slow), we can instead blend the results of mapping a given pixel through the four neighboring lookups.

A variant on this algorithm is to place the lookup tables at the *corners* of each $M \times M$ block (see Figure 3.9b and Exercise 3.7). In addition to blending four lookups to compute the final value, we can also *distribute* each input pixel into four adjacent lookup tables during the histogram accumulation phase (notice that the gray arrows in Figure 3.9b point both ways), i.e.,

$$h_{k,l}(I(i, j)) \mathrel{+}= w(i, j, k, l), \quad (3.11)$$

where $w(i, j, k, l)$ is the bilinear weighting function between pixel (i, j) and lookup table (k, l). This is an example of *soft histogramming*, which is used in a variety of other applications, including the construction of SIFT feature descriptors (Section 4.1.3) and vocabulary trees (Section 14.3.2).

3.1.5 *Application*: Tonal adjustment

One of the most widely used applications of point-wise image processing operators is the manipulation of contrast or *tone* in photographs, to make them look either more attractive or

[3] This algorithm is implemented in the MATLAB `adapthist` function.

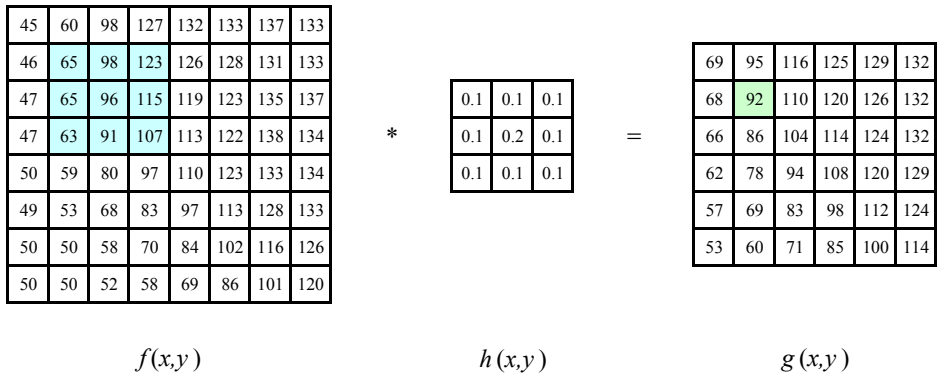


Figure 3.10 Neighborhood filtering (convolution): The image on the left is convolved with the filter in the middle to yield the image on the right. The light blue pixels indicate the source neighborhood for the light green destination pixel.

more interpretable. You can get a good sense of the range of operations possible by opening up any photo manipulation tool and trying out a variety of contrast, brightness, and color manipulation options, as shown in Figures 3.2 and 3.7.

Exercises 3.1, 3.5, and 3.6 have you implement some of these operations, in order to become familiar with basic image processing operators. More sophisticated techniques for tonal adjustment (Reinhard, Ward, Pattanaik *et al.* 2005; Bae, Paris, and Durand 2006) are described in the section on high dynamic range tone mapping (Section 10.2.1).

3.2 Linear filtering

Locally adaptive histogram equalization is an example of a *neighborhood operator* or *local operator*, which uses a collection of pixel values in the vicinity of a given pixel to determine its final output value (Figure 3.10). In addition to performing local tone adjustment, neighborhood operators can be used to *filter* images in order to add soft blur, sharpen details, accentuate edges, or remove noise (Figure 3.11b–d). In this section, we look at *linear* filtering operators, which involve weighted combinations of pixels in small neighborhoods. In Section 3.3, we look at non-linear operators such as morphological filters and distance transforms.

The most commonly used type of neighborhood operator is a *linear filter*, in which an output pixel's value is determined as a weighted sum of input pixel values (Figure 3.10),

$$g(i,j) = \sum_{k,l} f(i+k, j+l)h(k,l). \tag{3.12}$$

The entries in the weight *kernel* or *mask* $h(k,l)$ are often called the *filter coefficients*. The above *correlation* operator can be more compactly notated as

$$g = f \otimes h. \tag{3.13}$$



Figure 3.11 Some neighborhood operations: (a) original image; (b) blurred; (c) sharpened; (d) smoothed with edge-preserving filter; (e) binary image; (f) dilated; (g) distance transform; (h) connected components. For the dilation and connected components, black (ink) pixels are assumed to be active, i.e., to have a value of 1 in Equations (3.41–3.45).

$$\begin{array}{|c|c|c|c|c|} \hline 72 & 88 & 62 & 52 & 37 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 1/4 & 1/2 & 1/4 \\ \hline \end{array} \Leftrightarrow \frac{1}{4} \begin{bmatrix} 2 & 1 & . & . & . \\ 1 & 2 & 1 & . & . \\ . & 1 & 2 & 1 & . \\ . & . & 1 & 2 & 1 \\ . & . & . & 1 & 2 \end{bmatrix} \begin{bmatrix} 72 \\ 88 \\ 62 \\ 52 \\ 37 \end{bmatrix}$$

Figure 3.12 One-dimensional signal convolution as a sparse matrix-vector multiply, $\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f}$.

A common variant on this formula is

$$g(i,j) = \sum_{k,l} f(i-k, j-l)h(k,l) = \sum_{k,l} f(k,l)h(i-k, j-l), \tag{3.14}$$

where the sign of the offsets in f has been reversed. This is called the *convolution* operator,

$$g = f * h, \tag{3.15}$$

and h is then called the *impulse response function*.[4] The reason for this name is that the kernel function, h, convolved with an impulse signal, $\delta(i,j)$ (an image that is 0 everywhere except at the origin) reproduces itself, $h * \delta = h$, whereas correlation produces the reflected signal. (Try this yourself to verify that it is so.)

In fact, Equation (3.14) can be interpreted as the superposition (addition) of shifted impulse response functions $h(i-k, j-l)$ multiplied by the input pixel values $f(k,l)$. Convolution has additional nice properties, e.g., it is both commutative and associative. As well, the Fourier transform of two convolved images is the product of their individual Fourier transforms (Section 3.4).

Both correlation and convolution are *linear shift-invariant* (LSI) operators, which obey both the superposition principle (3.5),

$$h \circ (f_0 + f_1) = h \circ f_0 + h \circ f_1, \tag{3.16}$$

and the *shift invariance* principle,

$$g(i,j) = f(i+k, j+l) \quad \Leftrightarrow \quad (h \circ g)(i,j) = (h \circ f)(i+k, j+l), \tag{3.17}$$

which means that shifting a signal commutes with applying the operator ($\circ$ stands for the LSI operator). Another way to think of shift invariance is that the operator "behaves the same everywhere".

Occasionally, a shift-variant version of correlation or convolution may be used, e.g.,

$$g(i,j) = \sum_{k,l} f(i-k, j-l)h(k,l;i,j), \tag{3.18}$$

where $h(k,l;i,j)$ is the convolution kernel at pixel (i,j). For example, such a spatially varying kernel can be used to model blur in an image due to variable depth-dependent defocus.

Correlation and convolution can both be written as a matrix-vector multiply, if we first convert the two-dimensional images $f(i,j)$ and $g(i,j)$ into raster-ordered vectors $\boldsymbol{f}$ and $\boldsymbol{g}$,

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f}, \tag{3.19}$$

[4] The continuous version of convolution can be written as $g(\boldsymbol{x}) = \int f(\boldsymbol{x} - \boldsymbol{u})h(\boldsymbol{u})d\boldsymbol{u}$.

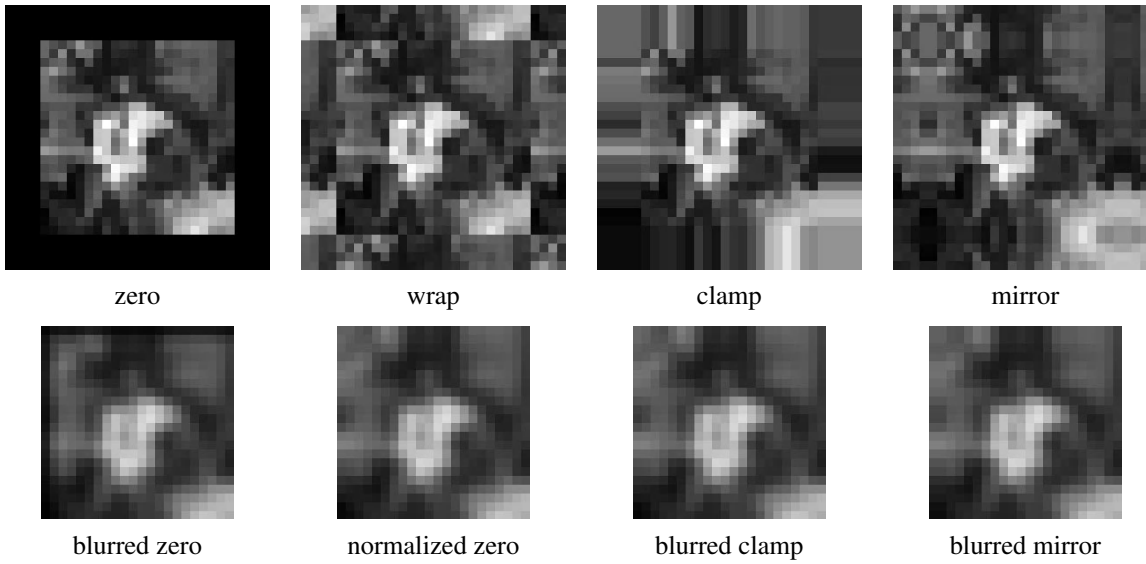


Figure 3.13 Border padding (top row) and the results of blurring the padded image (bottom row). The normalized zero image is the result of dividing (normalizing) the blurred zero-padded RGBA image by its corresponding soft alpha value.

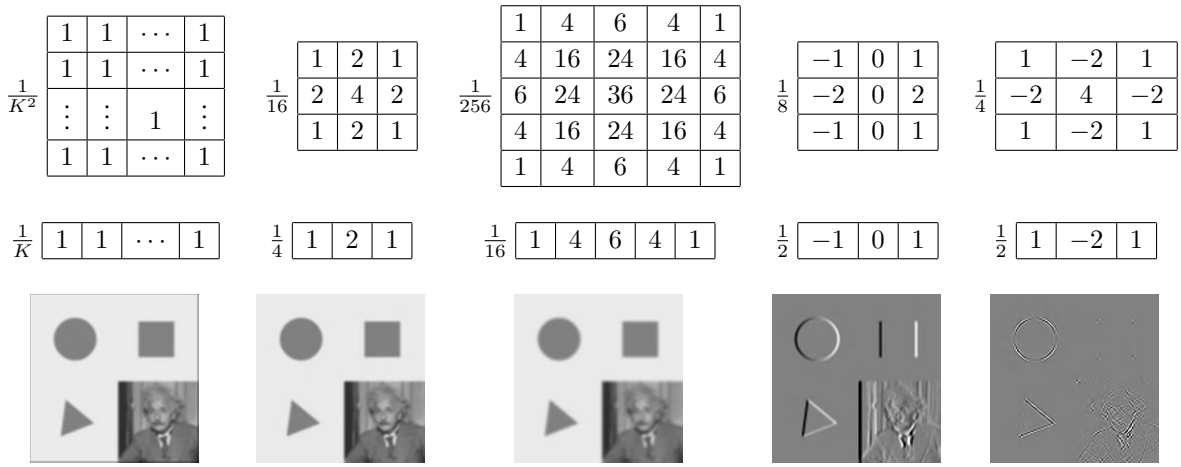
where the (sparse) $\boldsymbol{H}$ matrix contains the convolution kernels. Figure 3.12 shows how a one-dimensional convolution can be represented in matrix-vector form.

Padding (border effects)

The astute reader will notice that the matrix multiply shown in Figure 3.12 suffers from *boundary effects*, i.e., the results of filtering the image in this form will lead to a *darkening* of the corner pixels. This is because the original image is effectively being padded with 0 values wherever the convolution kernel extends beyond the original image boundaries.

To compensate for this, a number of alternative *padding* or extension modes have been developed (Figure 3.13):

- *zero*: set all pixels outside the source image to 0 (a good choice for alpha-matted cutout images);
- *constant (border color)*: set all pixels outside the source image to a specified *border* value;
- *clamp (replicate or clamp to edge)*: repeat edge pixels indefinitely;
- *(cyclic) wrap (repeat or tile)*: loop "around" the image in a "toroidal" configuration;
- *mirror*: reflect pixels across the image edge;
- *extend*: extend the signal by subtracting the mirrored version of the signal from the edge pixel value.



(a) box, $K = 5$ (b) bilinear (c) "Gaussian" (d) Sobel (e) corner

Figure 3.14 Separable linear filters: For each image (a)–(e), we show the 2D filter kernel (top), the corresponding horizontal 1D kernel (middle), and the filtered image (bottom). The filtered Sobel and corner images are signed, scaled up by $2\times$ and $4\times$, respectively, and added to a gray offset before display.

In the computer graphics literature (Akenine-Möller and Haines 2002, p. 124), these mechanisms are known as the *wrapping mode* (OpenGL) or *texture addressing mode* (Direct3D). The formulas for each of these modes are left to the reader (Exercise 3.8).

Figure 3.13 shows the effects of padding an image with each of the above mechanisms and then blurring the resulting padded image. As you can see, zero padding darkens the edges, clamp (replication) padding propagates border values inward, mirror (reflection) padding preserves colors near the borders. Extension padding (not shown) keeps the border pixels fixed (during blur).

An alternative to padding is to blur the zero-padded RGBA image and to then divide the resulting image by its alpha value to remove the darkening effect. The results can be quite good, as seen in the normalized zero image in Figure 3.13.

3.2.1 Separable filtering

The process of performing a convolution requires K^2 (multiply-add) operations per pixel, where K is the size (width or height) of the convolution kernel, e.g., the box filter in Figure 3.14a. In many cases, this operation can be significantly sped up by first performing a one-dimensional horizontal convolution followed by a one-dimensional vertical convolution (which requires a total of $2K$ operations per pixel). A convolution kernel for which this is possible is said to be *separable*.

It is easy to show that the two-dimensional kernel $\boldsymbol{K}$ corresponding to successive convolution with a horizontal kernel $\boldsymbol{h}$ and a vertical kernel $\boldsymbol{v}$ is the *outer product* of the two kernels,

$$\boldsymbol{K} = \boldsymbol{v}\boldsymbol{h}^T \tag{3.20}$$

(see Figure 3.14 for some examples). Because of the increased efficiency, the design of

convolution kernels for computer vision applications is often influenced by their separability.

How can we tell if a given kernel $\boldsymbol{K}$ is indeed separable? This can often be done by inspection or by looking at the analytic form of the kernel (Freeman and Adelson 1991). A more direct method is to treat the 2D kernel as a 2D matrix $\boldsymbol{K}$ and to take its singular value decomposition (SVD),

$$\boldsymbol{K} = \sum_i \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T \tag{3.21}$$

(see Appendix A.1.1 for the definition of the SVD). If only the first singular value σ_0 is non-zero, the kernel is separable and $\sqrt{\sigma_0}\boldsymbol{u}_0$ and $\sqrt{\sigma_0}\boldsymbol{v}_0^T$ provide the vertical and horizontal kernels (Perona 1995). For example, the Laplacian of Gaussian kernel (3.26 and 4.23) can be implemented as the sum of two separable filters (4.24) (Wiejak, Buxton, and Buxton 1985).

What if your kernel is not separable and yet you still want a faster way to implement it? Perona (1995), who first made the link between kernel separability and SVD, suggests using more terms in the (3.21) series, i.e., summing up a number of separable convolutions. Whether this is worth doing or not depends on the relative sizes of K and the number of significant singular values, as well as other considerations, such as cache coherency and memory locality.

3.2.2 Examples of linear filtering

Now that we have described the process for performing linear filtering, let us examine a number of frequently used filters.

The simplest filter to implement is the *moving average* or *box* filter, which simply averages the pixel values in a $K \times K$ window. This is equivalent to convolving the image with a kernel of all ones and then scaling (Figure 3.14a). For large kernels, a more efficient implementation is to slide a moving window across each scanline (in a separable filter) while adding the newest pixel and subtracting the oldest pixel from the running sum. This is related to the concept of *summed area tables*, which we describe shortly.

A smoother image can be obtained by separably convolving the image with a piecewise linear "tent" function (also known as a *Bartlett* filter). Figure 3.14b shows a 3×3 version of this filter, which is called the *bilinear* kernel, since it is the outer product of two linear (first-order) splines (see Section 3.5.2).

Convolving the linear tent function with itself yields the cubic approximating spline, which is called the "Gaussian" kernel (Figure 3.14c) in Burt and Adelson's (1983a) *Laplacian pyramid* representation (Section 3.5). Note that approximate Gaussian kernels can also be obtained by iterated convolution with box filters (Wells 1986). In applications where the filters really need to be rotationally symmetric, carefully tuned versions of sampled Gaussians should be used (Freeman and Adelson 1991) (Exercise 3.10).

The kernels we just discussed are all examples of blurring (smoothing) or *low-pass* kernels (since they pass through the lower frequencies while attenuating higher frequencies). How good are they at doing this? In Section 3.4, we use frequency-space Fourier analysis to examine the exact frequency response of these filters. We also introduce the *sinc* ($(\sin x)/x$) filter, which performs *ideal* low-pass filtering.

In practice, smoothing kernels are often used to reduce high-frequency noise. We have much more to say about using variants on smoothing to remove noise later (see Sections 3.3.1, 3.4, and 3.7).

Surprisingly, smoothing kernels can also be used to *sharpen* images using a process called *unsharp masking*. Since blurring the image reduces high frequencies, adding some of the difference between the original and the blurred image makes it sharper,

$$g_{\text{sharp}} = f + \gamma(f - h_{\text{blur}} * f). \tag{3.22}$$

In fact, before the advent of digital photography, this was the standard way to sharpen images in the darkroom: create a blurred ("positive") negative from the original negative by misfocusing, then overlay the two negatives before printing the final image, which corresponds to

$$g_{\text{unsharp}} = f(1 - \gamma h_{\text{blur}} * f). \tag{3.23}$$

This is no longer a linear filter but it still works well.

Linear filtering can also be used as a pre-processing stage to edge extraction (Section 4.2) and interest point detection (Section 4.1) algorithms. Figure 3.14d shows a simple 3×3 edge extractor called the Sobel operator, which is a separable combination of a horizontal *central difference* (so called because the horizontal derivative is centered on the pixel) and a vertical tent filter (to smooth the results). As you can see in the image below the kernel, this filter effectively emphasizes horizontal edges.

The simple corner detector (Figure 3.14e) looks for simultaneous horizontal and vertical second derivatives. As you can see however, it responds not only to the corners of the square, but also along diagonal edges. Better corner detectors, or at least interest point detectors that are more rotationally invariant, are described in Section 4.1.

3.2.3 Band-pass and steerable filters

The Sobel and corner operators are simple examples of band-pass and oriented filters. More sophisticated kernels can be created by first smoothing the image with a (unit area) Gaussian filter,

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3.24}$$

and then taking the first or second derivatives (Marr 1982; Witkin 1983; Freeman and Adelson 1991). Such filters are known collectively as *band-pass filters*, since they filter out both low and high frequencies.

The (undirected) second derivative of a two-dimensional image,

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 y}{\partial y^2}, \tag{3.25}$$

is known as the *Laplacian* operator. Blurring an image with a Gaussian and then taking its Laplacian is equivalent to convolving directly with the *Laplacian of Gaussian* (LoG) filter,

$$\nabla^2 G(x, y; \sigma) = \left(\frac{x^2 + y^2}{\sigma^4} - \frac{2}{\sigma^2} \right) G(x, y; \sigma), \tag{3.26}$$

which has certain nice *scale-space properties* (Witkin 1983; Witkin, Terzopoulos, and Kass 1986). The five-point Laplacian is just a compact approximation to this more sophisticated filter.

Likewise, the Sobel operator is a simple approximation to a *directional* or *oriented* filter, which can obtained by smoothing with a Gaussian (or some other filter) and then taking a



(a) (b) (c)

Figure 3.15 Second-order steerable filter (Freeman 1992) © 1992 IEEE: (a) original image of Einstein; (b) orientation map computed from the second-order oriented energy; (c) original image with oriented structures enhanced.

directional derivative $\nabla_{\hat{\boldsymbol{u}}} = \frac{\partial}{\partial \hat{\boldsymbol{u}}}$, which is obtained by taking the dot product between the gradient field ∇ and a unit direction $\hat{\boldsymbol{u}} = (\cos\theta, \sin\theta)$,

$$\hat{\boldsymbol{u}} \cdot \nabla(G * f) = \nabla_{\hat{\boldsymbol{u}}}(G * f) = (\nabla_{\hat{\boldsymbol{u}}} G) * f. \tag{3.27}$$

The smoothed directional derivative filter,

$$G_{\hat{\boldsymbol{u}}} = uG_x + vG_y = u\frac{\partial G}{\partial x} + v\frac{\partial G}{\partial y}, \tag{3.28}$$

where $\hat{\boldsymbol{u}} = (u, v)$, is an example of a *steerable* filter, since the value of an image convolved with $G_{\hat{\boldsymbol{u}}}$ can be computed by first convolving with the pair of filters (G_x, G_y) and then *steering* the filter (potentially locally) by multiplying this gradient field with a unit vector $\hat{\boldsymbol{u}}$ (Freeman and Adelson 1991). The advantage of this approach is that a whole *family* of filters can be evaluated with very little cost.

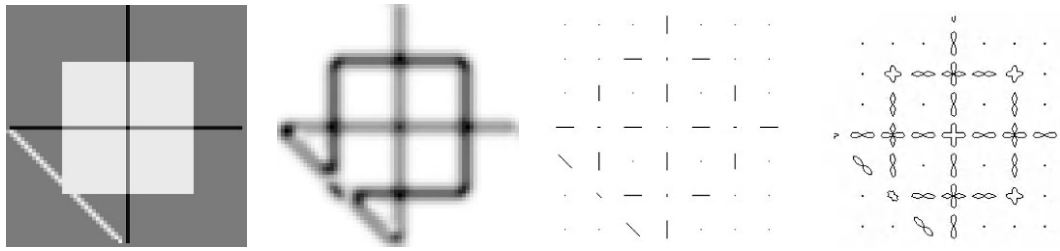
How about steering a directional second derivative filter $\nabla_{\hat{\boldsymbol{u}}} \cdot \nabla_{\hat{\boldsymbol{u}}} G_{\hat{\boldsymbol{u}}}$, which is the result of taking a (smoothed) directional derivative and then taking the directional derivative again? For example, G_{xx} is the second directional derivative in the x direction.

At first glance, it would appear that the steering trick will not work, since for every direction $\hat{\boldsymbol{u}}$, we need to compute a different first directional derivative. Somewhat surprisingly, Freeman and Adelson (1991) showed that, for directional Gaussian derivatives, it is possible to steer *any* order of derivative with a relatively small number of basis functions. For example, only three basis functions are required for the second-order directional derivative,

$$G_{\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}} = u^2 G_{xx} + 2uvG_{xy} + v^2 G_{yy}. \tag{3.29}$$

Furthermore, each of the basis filters, while not itself necessarily separable, can be computed using a linear combination of a small number of separable filters (Freeman and Adelson 1991).

This remarkable result makes it possible to construct directional derivative filters of increasingly greater *directional selectivity*, i.e., filters that only respond to edges that have strong local consistency in orientation (Figure 3.15). Furthermore, higher order steerable



(a) (b) (c) (d)

Figure 3.16 Fourth-order steerable filter (Freeman and Adelson 1991) © 1991 IEEE: (a) test image containing bars (lines) and step edges at different orientations; (b) average oriented energy; (c) dominant orientation; (d) oriented energy as a function of angle (polar plot).

filters can respond to potentially more than a single edge orientation at a given location, and they can respond to both *bar* edges (thin lines) and the classic step edges (Figure 3.16). In order to do this, however, full *Hilbert transform pairs* need to be used for second-order and higher filters, as described in (Freeman and Adelson 1991).

Steerable filters are often used to construct both feature descriptors (Section 4.1.3) and edge detectors (Section 4.2). While the filters developed by Freeman and Adelson (1991) are best suited for detecting linear (edge-like) structures, more recent work by Koethe (2003) shows how a combined 2×2 boundary tensor can be used to encode both edge and junction ("corner") features. Exercise 3.12 has you implement such steerable filters and apply them to finding both edge and corner features.

Summed area table (integral image)

If an image is going to be repeatedly convolved with different box filters (and especially filters of different sizes at different locations), you can precompute the *summed area table* (Crow 1984), which is just the running sum of all the pixel values from the origin,

$$s(i,j) = \sum_{k=0}^{i} \sum_{l=0}^{j} f(k,l). \tag{3.30}$$

This can be efficiently computed using a recursive (raster-scan) algorithm,

$$s(i,j) = s(i-1,j) + s(i,j-1) - s(i-1,j-1) + f(i,j). \tag{3.31}$$

The image $s(i,j)$ is also often called an *integral image* (see Figure 3.17) and can actually be computed using only two additions per pixel if separate row sums are used (Viola and Jones 2004). To find the summed area (integral) inside a rectangle $[i_0, i_1] \times [j_0, j_1]$, we simply combine four samples from the summed area table,

$$S(i_0 \ldots i_1, j_0 \ldots j_1) = \sum_{i=i_0}^{i_1} \sum_{j=j_0}^{j_1} s(i_1,j_1) - s(i_1,j_0-1) - s(i_0-1,j_1) + s(i_0-1,j_0-1). \tag{3.32}$$

A potential disadvantage of summed area tables is that they require $\log M + \log N$ extra bits in the accumulation image compared to the original image, where M and N are the image

3	2	7	2	3
1	5	1	3	4
5	1	**3**	5	1
4	3	2	1	6
2	4	1	4	8

(a) S = 24

3	5	12	14	17
4	*11*	**19**	24	31
9	**17**	28	38	46
13	24	37	48	62
15	30	44	59	81

(b) s = 28

3	5	12	*14*	17
4	11	19	24	31
9	17	28	38	46
13	24	37	**48**	62
15	30	44	59	81

(c) S = 24

Figure 3.17 Summed area tables: (a) original image; (b) summed area table; (c) computation of area sum. Each value in the summed area table $s(i, j)$ (red) is computed recursively from its three adjacent (blue) neighbors (3.31). Area sums S (green) are computed by combining the four values at the rectangle corners (purple) (3.32). Positive values are shown in **bold** and negative values in *italics*.

width and height. Extensions of summed area tables can also be used to approximate other convolution kernels (Wolberg (1990, Section 6.5.2) contains a review).

In computer vision, summed area tables have been used in face detection (Viola and Jones 2004) to compute simple multi-scale low-level features. Such features, which consist of adjacent rectangles of positive and negative values, are also known as *boxlets* (Simard, Bottou, Haffner *et al.* 1998). In principle, summed area tables could also be used to compute the sums in the sum of squared differences (SSD) stereo and motion algorithms (Section 11.4). In practice, separable moving average filters are usually preferred (Kanade, Yoshida, Oda *et al.* 1996), unless many different window shapes and sizes are being considered (Veksler 2003).

Recursive filtering

The incremental formula (3.31) for the summed area is an example of a *recursive filter*, i.e., one whose values depends on previous filter outputs. In the signal processing literature, such filters are known as *infinite impulse response* (IIR), since the output of the filter to an impulse (single non-zero value) goes on forever. For example, for a summed area table, an impulse generates an infinite rectangle of 1s below and to the right of the impulse. The filters we have previously studied in this chapter, which involve the image with a finite extent kernel, are known as *finite impulse response* (FIR).

Two-dimensional IIR filters and recursive formulas are sometimes used to compute quantities that involve large area interactions, such as two-dimensional distance functions (Section 3.3.3) and connected components (Section 3.3.4).

More commonly, however, IIR filters are used inside one-dimensional separable filtering stages to compute large-extent smoothing kernels, such as efficient approximations to Gaussians and edge filters (Deriche 1990; Nielsen, Florack, and Deriche 1997). Pyramid-based algorithms (Section 3.5) can also be used to perform such large-area smoothing computations.

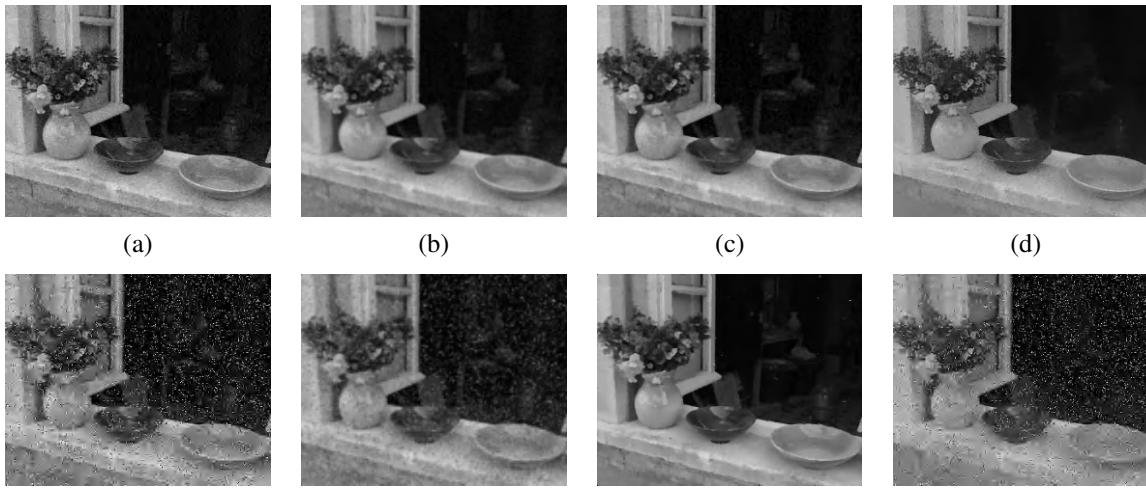


Figure 3.18 Median and bilateral filtering: (a) original image with Gaussian noise; (b) Gaussian filtered; (c) median filtered; (d) bilaterally filtered; (e) original image with shot noise; (f) Gaussian filtered; (g) median filtered; (h) bilaterally filtered. Note that the bilateral filter fails to remove the shot noise because the noisy pixels are too different from their neighbors.

3.3 More neighborhood operators

As we have just seen, linear filters can perform a wide variety of image transformations. However non-linear filters, such as edge-preserving median or bilateral filters, can sometimes perform even better. Other examples of neighborhood operators include *morphological* operators that operate on binary images, as well as *semi-global* operators that compute *distance transforms* and find *connected components* in binary images (Figure 3.11f–h).

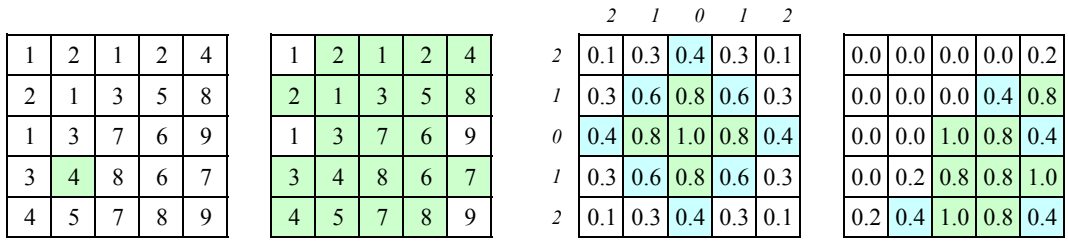
3.3.1 Non-linear filtering

The filters we have looked at so far have all been *linear*, i.e., their response to a sum of two signals is the same as the sum of the individual responses. This is equivalent to saying that each output pixel is a weighted summation of some number of input pixels (3.19). Linear filters are easier to compose and are amenable to frequency response analysis (Section 3.4).

In many cases, however, better performance can be obtained by using a *non-linear* combination of neighboring pixels. Consider for example the image in Figure 3.18e, where the noise, rather than being Gaussian, is *shot noise*, i.e., it occasionally has very large values. In this case, regular blurring with a Gaussian filter fails to remove the noisy pixels and instead turns them into softer (but still visible) spots (Figure 3.18f).

Median filtering

A better filter to use in this case is the *median* filter, which selects the median value from each pixel's neighborhood (Figure 3.19a). Median values can be computed in expected linear time using a randomized select algorithm (Cormen 2001) and incremental variants have also been



(a) median = 4 (b) α-mean= 4.6 (c) domain filter (d) range filter

Figure 3.19 Median and bilateral filtering: (a) median pixel (green); (b) selected α-trimmed mean pixels; (c) domain filter (numbers along edge are pixel distances); (d) range filter.

developed by Tomasi and Manduchi (1998) and Bovik (2000, Section 3.2). Since the shot noise value usually lies well outside the true values in the neighborhood, the median filter is able to filter away such bad pixels (Figure 3.18c).

One downside of the median filter, in addition to its moderate computational cost, is that since it selects only one input pixel value to replace each output pixel, it is not as *efficient* at averaging away regular Gaussian noise (Huber 1981; Hampel, Ronchetti, Rousseeuw *et al.* 1986; Stewart 1999). A better choice may be the α-trimmed mean (Lee and Redner 1990) (Crane 1997, p. 109), which averages together all of the pixels except for the α fraction that are the smallest and the largest (Figure 3.19b).

Another possibility is to compute a *weighted median*, in which each pixel is used a number of times depending on its distance from the center. This turns out to be equivalent to minimizing the weighted objective function

$$\sum_{k,l} w(k,l)|f(i+k,j+l) - g(i,j)|^p, \tag{3.33}$$

where $g(i,j)$ is the desired output value and $p = 1$ for the weighted median. The value $p = 2$ is the usual *weighted mean*, which is equivalent to correlation (3.12) after normalizing by the sum of the weights (Bovik 2000, Section 3.2) (Haralick and Shapiro 1992, Section 7.2.6). The weighted mean also has deep connections to other methods in robust statistics (see Appendix B.3), such as influence functions (Huber 1981; Hampel, Ronchetti, Rousseeuw *et al.* 1986).

Non-linear smoothing has another, perhaps even more important property, especially since shot noise is rare in today's cameras. Such filtering is more *edge preserving*, i.e., it has less tendency to soften edges while filtering away high-frequency noise.

Consider the noisy image in Figure 3.18a. In order to remove most of the noise, the Gaussian filter is forced to smooth away high-frequency detail, which is most noticeable near strong edges. Median filtering does better but, as mentioned before, does not do as good a job at smoothing away from discontinuities. See (Tomasi and Manduchi 1998) for some additional references to edge-preserving smoothing techniques.

While we could try to use the α-trimmed mean or weighted median, these techniques still have a tendency to round sharp corners, since the majority of pixels in the smoothing area come from the background distribution.

Bilateral filtering

What if we were to combine the idea of a weighted filter kernel with a better version of outlier rejection? What if instead of rejecting a fixed percentage α, we simply reject (in a soft way) pixels whose *values* differ too much from the central pixel value? This is the essential idea in *bilateral filtering*, which was first popularized in the computer vision community by Tomasi and Manduchi (1998). Chen, Paris, and Durand (2007) and Paris, Kornprobst, Tumblin *et al.* (2008) cite similar earlier work (Aurich and Weule 1995; Smith and Brady 1997) as well as the wealth of subsequent applications in computer vision and computational photography.

In the bilateral filter, the output pixel value depends on a weighted combination of neighboring pixel values

$$g(i,j) = \frac{\sum_{k,l} f(k,l) w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}. \tag{3.34}$$

The weighting coefficient $w(i,j,k,l)$ depends on the product of a *domain kernel* (Figure 3.19c),

$$d(i,j,k,l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2}\right), \tag{3.35}$$

and a data-dependent *range kernel* (Figure 3.19d),

$$r(i,j,k,l) = \exp\left(-\frac{\|f(i,j) - f(k,l)\|^2}{2\sigma_r^2}\right). \tag{3.36}$$

When multiplied together, these yield the data-dependent *bilateral weight function*

$$w(i,j,k,l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{\|f(i,j) - f(k,l)\|^2}{2\sigma_r^2}\right). \tag{3.37}$$

Figure 3.20 shows an example of the bilateral filtering of a noisy step edge. Note how the domain kernel is the usual Gaussian, the range kernel measures appearance (intensity) similarity to the center pixel, and the bilateral filter kernel is a product of these two.

Notice that the range filter (3.36) uses the *vector distance* between the center and the neighboring pixel. This is important in color images, since an edge in any *one* of the color bands signals a change in material and hence the need to downweight a pixel's influence.[5]

Since bilateral filtering is quite slow compared to regular separable filtering, a number of acceleration techniques have been developed (Durand and Dorsey 2002; Paris and Durand 2006; Chen, Paris, and Durand 2007; Paris, Kornprobst, Tumblin *et al.* 2008). Unfortunately, these techniques tend to use more memory than regular filtering and are hence not directly applicable to filtering full-color images.

Iterated adaptive smoothing and anisotropic diffusion

Bilateral (and other) filters can also be applied in an iterative fashion, especially if an appearance more like a "cartoon" is desired (Tomasi and Manduchi 1998). When iterated filtering is applied, a much smaller neighborhood can often be used.

[5] Tomasi and Manduchi (1998) show that using the vector distance (as opposed to filtering each color band separately) reduces color fringing effects. They also recommend taking the color difference in the more perceptually uniform CIELAB color space (see Section 2.3.2).

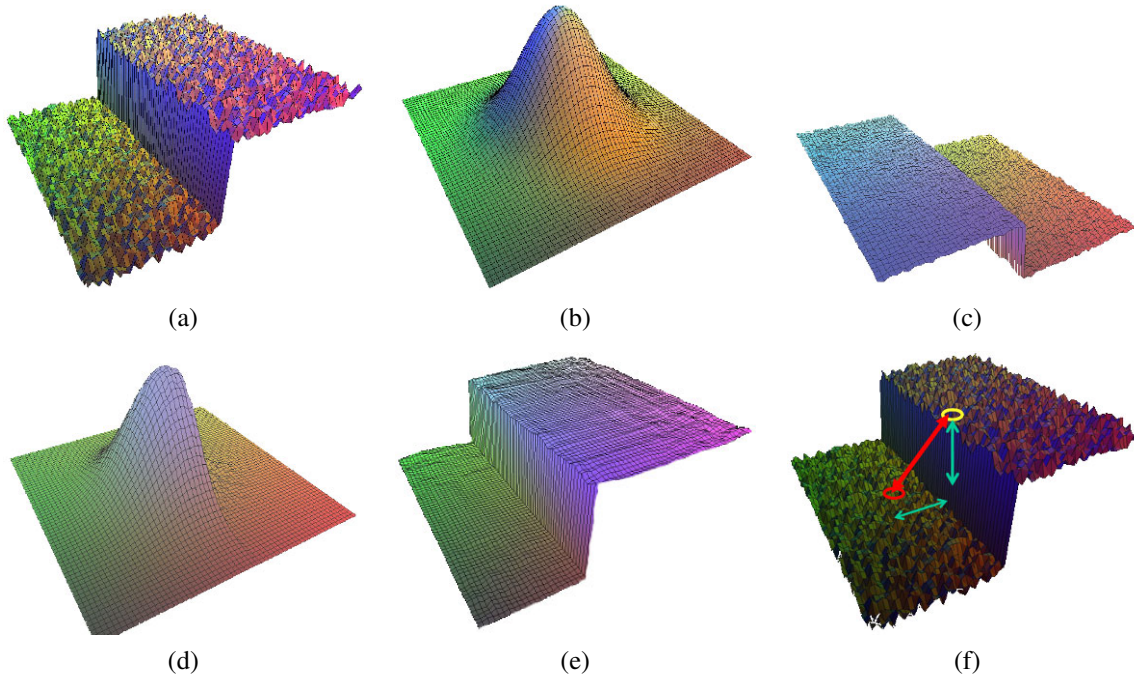


Figure 3.20 Bilateral filtering (Durand and Dorsey 2002) © 2002 ACM: (a) noisy step edge input; (b) domain filter (Gaussian); (c) range filter (similarity to center pixel value); (d) bilateral filter; (e) filtered step edge output; (f) 3D distance between pixels.

Consider, for example, using only the four nearest neighbors, i.e., restricting $|k-i|+|l-j| \leq 1$ in (3.34). Observe that

$$d(i,j,k,l) = \exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2}\right) \tag{3.38}$$

$$= \begin{cases} 1, & |k-i|+|l-j|=0, \\ \lambda = e^{-1/2\sigma_d^2}, & |k-i|+|l-j|=1. \end{cases} \tag{3.39}$$

We can thus re-write (3.34) as

$$f^{(t+1)}(i,j) = \frac{f^{(t)}(i,j)+\eta\sum_{k,l} f^{(t)}(k,l)r(i,j,k,l)}{1+\eta\sum_{k,l} r(i,j,k,l)} \tag{3.40}$$

$$= f^{(t)}(i,j) + \frac{\eta}{1+\eta R}\sum_{k,l} r(i,j,k,l)[f^{(t)}(k,l) - f^{(t)}(i,j)],$$

where $R = \sum_{(k,l)} r(i,j,k,l)$, (k,l) are the $\mathcal{N}_4$ neighbors of (i,j), and we have made the iterative nature of the filtering explicit.

As Barash (2002) notes, (3.40) is the same as the discrete *anisotropic diffusion* equation first proposed by Perona and Malik (1990b).[6] Since its original introduction, anisotropic diffusion has been extended and applied to a wide range of problems (Nielsen, Florack, and Deriche 1997; Black, Sapiro, Marimont *et al.* 1998; Weickert, ter Haar Romeny, and Viergever

[6] The $1/(1+\eta R)$ factor is not present in anisotropic diffusion but becomes negligible as $\eta \to 0$.

1998; Weickert 1998). It has also been shown to be closely related to other *adaptive smoothing* techniques (Saint-Marc, Chen, and Medioni 1991; Barash 2002; Barash and Comaniciu 2004) as well as Bayesian regularization with a non-linear smoothness term that can be derived from image statistics (Scharr, Black, and Haussecker 2003).

In its general form, the range kernel $r(i, j, k, l) = r(\|f(i, j) - f(k, l)\|)$, which is usually called the *gain* or *edge-stopping* function, or diffusion coefficient, can be any monotonically increasing function with $r'(x) \to 0$ as $x \to \infty$. Black, Sapiro, Marimont *et al.* (1998) show how anisotropic diffusion is equivalent to minimizing a robust penalty function on the image gradients, which we discuss in Sections 3.7.1 and 3.7.2). Scharr, Black, and Haussecker (2003) show how the edge-stopping function can be derived in a principled manner from local image statistics. They also extend the diffusion neighborhood from $\mathcal{N}_4$ to $\mathcal{N}_8$, which allows them to create a diffusion operator that is both rotationally invariant and incorporates information about the eigenvalues of the local structure tensor.

Note that, without a bias term towards the original image, anisotropic diffusion and iterative adaptive smoothing converge to a constant image. Unless a small number of iterations is used (e.g., for speed), it is usually preferable to formulate the smoothing problem as a joint minimization of a smoothness term and a data fidelity term, as discussed in Sections 3.7.1 and 3.7.2 and by Scharr, Black, and Haussecker (2003), which introduce such a bias in a principled manner.

3.3.2 Morphology

While non-linear filters are often used to enhance grayscale and color images, they are also used extensively to process binary images. Such images often occur after a *thresholding* operation,

$$\theta(f, t) = \begin{cases} 1 & \text{if } f \geq t, \\ 0 & \text{else,} \end{cases} \tag{3.41}$$

e.g., converting a scanned grayscale document into a binary image for further processing such as *optical character recognition*.

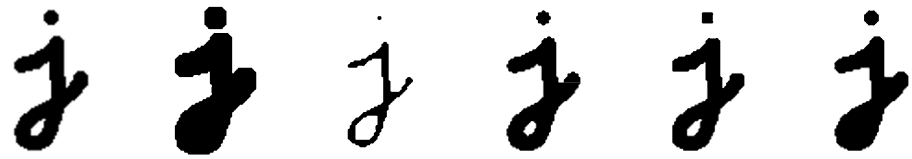
The most common binary image operations are called *morphological operations*, since they change the *shape* of the underlying binary objects (Ritter and Wilson 2000, Chapter 7). To perform such an operation, we first convolve the binary image with a binary *structuring element* and then select a binary output value depending on the thresholded result of the convolution. (This is not the usual way in which these operations are described, but I find it a nice simple way to unify the processes.) The structuring element can be any shape, from a simple 3×3 box filter, to more complicated disc structures. It can even correspond to a particular shape that is being sought for in the image.

Figure 3.21 shows a close-up of the convolution of a binary image f with a 3×3 structuring element s and the resulting images for the operations described below. Let

$$c = f \otimes s \tag{3.42}$$

be the integer-valued *count* of the number of 1s inside each structuring element as it is scanned over the image and S be the size of the structuring element (number of pixels). The standard operations used in binary morphology include:

- **dilation**: $\text{dilate}(f, s) = \theta(c, 1)$;



(a) (b) (c) (d) (e) (f)

Figure 3.21 Binary image morphology: (a) original image; (b) dilation; (c) erosion; (d) majority; (e) opening; (f) closing. The structuring element for all examples is a 5×5 square. The effects of majority are a subtle rounding of sharp corners. Opening fails to eliminate the dot, since it is not wide enough.

- **erosion**: $\text{erode}(f, s) = \theta(c, S)$;
- **majority**: $\text{maj}(f, s) = \theta(c, S/2)$;
- **opening**: $\text{open}(f, s) = \text{dilate}(\text{erode}(f, s), s)$;
- **closing**: $\text{close}(f, s) = \text{erode}(\text{dilate}(f, s), s)$.

As we can see from Figure 3.21, dilation grows (thickens) objects consisting of 1s, while erosion shrinks (thins) them. The opening and closing operations tend to leave large regions and smooth boundaries unaffected, while removing small objects or holes and smoothing boundaries.

While we will not use mathematical morphology much in the rest of this book, it is a handy tool to have around whenever you need to clean up some thresholded images. You can find additional details on morphology in other textbooks on computer vision and image processing (Haralick and Shapiro 1992, Section 5.2) (Bovik 2000, Section 2.2) (Ritter and Wilson 2000, Section 7) as well as articles and books specifically on this topic (Serra 1982; Serra and Vincent 1992; Yuille, Vincent, and Geiger 1992; Soille 2006).

3.3.3 Distance transforms

The distance transform is useful in quickly precomputing the distance to a curve or set of points using a two-pass raster algorithm (Rosenfeld and Pfaltz 1966; Danielsson 1980; Borgefors 1986; Paglieroni 1992; Breu, Gil, Kirkpatrick *et al.* 1995; Felzenszwalb and Huttenlocher 2004a; Fabbri, Costa, Torelli *et al.* 2008). It has many applications, including level sets (Section 5.1.4), fast *chamfer matching* (binary image alignment) (Huttenlocher, Klanderman, and Rucklidge 1993), feathering in image stitching and blending (Section 9.3.2), and nearest point alignment (Section 12.2.1).

The distance transform $D(i, j)$ of a binary image $b(i, j)$ is defined as follows. Let $d(k, l)$ be some *distance metric* between pixel offsets. Two commonly used metrics include the *city block* or *Manhattan* distance

$$d_1(k, l) = |k| + |l| \tag{3.43}$$

and the *Euclidean* distance

$$d_2(k, l) = \sqrt{k^2 + l^2}. \tag{3.44}$$

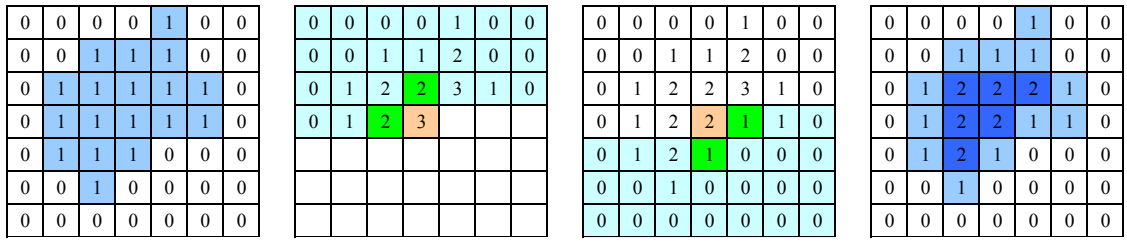


Figure 3.22 City block distance transform: (a) original binary image; (b) top to bottom (forward) raster sweep: green values are used to compute the orange value; (c) bottom to top (backward) raster sweep: green values are merged with old orange value; (d) final distance transform.

The distance transform is then defined as

$$D(i,j) = \min_{k,l:b(k,l)=0} d(i-k, j-l), \tag{3.45}$$

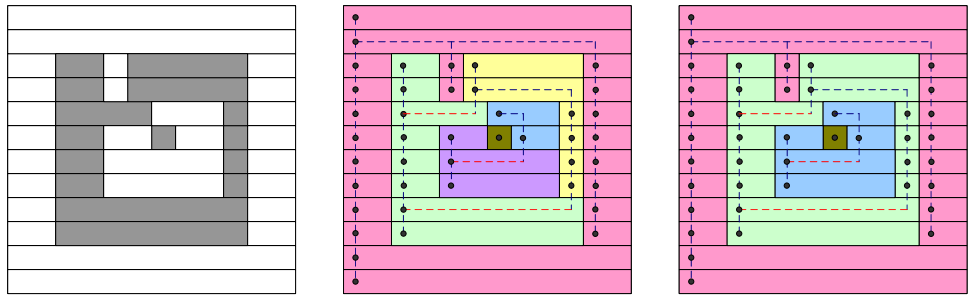
i.e., it is the distance to the *nearest* background pixel whose value is 0.

The D_1 city block distance transform can be efficiently computed using a forward and backward pass of a simple raster-scan algorithm, as shown in Figure 3.22. During the forward pass, each non-zero pixel in b is replaced by the minimum of 1 + the distance of its north or west neighbor. During the backward pass, the same occurs, except that the minimum is both over the current value D and 1 + the distance of the south and east neighbors (Figure 3.22).

Efficiently computing the Euclidean distance transform is more complicated. Here, just keeping the minimum scalar distance to the boundary during the two passes is not sufficient. Instead, a *vector-valued* distance consisting of both the x and y coordinates of the distance to the boundary must be kept and compared using the squared distance (hypotenuse) rule. As well, larger search regions need to be used to obtain reasonable results. Rather than explaining the algorithm (Danielsson 1980; Borgefors 1986) in more detail, we leave it as an exercise for the motivated reader (Exercise 3.13).

Figure 3.11g shows a distance transform computed from a binary image. Notice how the values grow away from the black (ink) regions and form ridges in the white area of the original image. Because of this linear growth from the starting boundary pixels, the distance transform is also sometimes known as the *grassfire transform*, since it describes the time at which a fire starting inside the black region would consume any given pixel, or a *chamfer*, because it resembles similar shapes used in woodworking and industrial design. The ridges in the distance transform become the *skeleton* (or *medial axis transform (MAT)*) of the region where the transform is computed, and consist of pixels that are of equal distance to two (or more) boundaries (Tek and Kimia 2003; Sebastian and Kimia 2005).

A useful extension of the basic distance transform is the *signed distance transform*, which computes distances to boundary pixels for *all* the pixels (Lavallée and Szeliski 1995). The simplest way to create this is to compute the distance transforms for both the original binary image and its complement and to negate one of them before combining. Because such distance fields tend to be smooth, it is possible to store them more compactly (with minimal loss in *relative* accuracy) using a spline defined over a quadtree or octree data structure



(a) (b) (c)

Figure 3.23 Connected component computation: (a) original grayscale image; (b) horizontal runs (nodes) connected by vertical (graph) edges (dashed blue)—runs are pseudocolored with unique colors inherited from parent nodes; (c) re-coloring after merging adjacent segments.

(Lavallée and Szeliski 1995; Szeliski and Lavallée 1996; Frisken, Perry, Rockwood *et al.* 2000). Such precomputed signed distance transforms can be extremely useful in efficiently aligning and merging 2D curves and 3D surfaces (Huttenlocher, Klanderman, and Rucklidge 1993; Szeliski and Lavallée 1996; Curless and Levoy 1996), especially if the *vectorial* version of the distance transform, i.e., a pointer from each pixel or voxel to the nearest boundary or surface element, is stored and interpolated. Signed distance fields are also an essential component of level set evolution (Section 5.1.4), where they are called *characteristic functions*.

3.3.4 Connected components

Another useful semi-global image operation is finding *connected components*, which are defined as regions of adjacent pixels that have the same input value (or label). (In the remainder of this section, consider pixels to be *adjacent* if they are immediate $\mathcal{N}_4$ neighbors and they have the same input value.) Connected components can be used in a variety of applications, such as finding individual letters in a scanned document or finding objects (say, cells) in a thresholded image and computing their area statistics.

Consider the grayscale image in Figure 3.23a. There are four connected components in this figure: the outermost set of white pixels, the large ring of gray pixels, the white enclosed region, and the single gray pixel. These are shown pseudocolored in Figure 3.23c as pink, green, blue, and brown.

To compute the connected components of an image, we first (conceptually) split the image into horizontal *runs* of adjacent pixels, and then color the runs with unique labels, re-using the labels of vertically adjacent runs whenever possible. In a second phase, adjacent runs of different colors are then merged.

While this description is a little sketchy, it should be enough to enable a motivated student to implement this algorithm (Exercise 3.14). Haralick and Shapiro (1992, Section 2.3) give a much longer description of various connected component algorithms, including ones that avoid the creation of a potentially large re-coloring (equivalence) table. Well-debugged connected component algorithms are also available in most image processing libraries.

Once a binary or multi-valued image has been segmented into its connected components,

it is often useful to compute the area statistics for each individual region $\mathcal{R}$. Such statistics include:

- the area (number of pixels);
- the perimeter (number of boundary pixels);
- the centroid (average x and y values);
- the second moments,

$$\boldsymbol{M} = \sum_{(x,y)\in\mathcal{R}} \begin{bmatrix} x-\overline{x} \\ y-\overline{y} \end{bmatrix} \begin{bmatrix} x-\overline{x} & y-\overline{y} \end{bmatrix}, \tag{3.46}$$

  from which the major and minor axis orientation and lengths can be computed using eigenvalue analysis.[7]

These statistics can then be used for further processing, e.g., for sorting the regions by the area size (to consider the largest regions first) or for preliminary matching of regions in different images.

3.4 Fourier transforms

In Section 3.2, we mentioned that Fourier analysis could be used to analyze the frequency characteristics of various filters. In this section, we explain both how Fourier analysis lets us determine these characteristics (or equivalently, the frequency *content* of an image) and how using the Fast Fourier Transform (FFT) lets us perform large-kernel convolutions in time that is independent of the kernel's size. More comprehensive introductions to Fourier transforms are provided by Bracewell (1986); Glassner (1995); Oppenheim and Schafer (1996); Oppenheim, Schafer, and Buck (1999).

How can we analyze what a given filter does to high, medium, and low frequencies? The answer is to simply pass a sinusoid of known frequency through the filter and to observe by how much it is attenuated. Let

$$s(x) = \sin(2\pi f x + \phi_i) = \sin(\omega x + \phi_i) \tag{3.47}$$

be the input sinusoid whose *frequency* is f, *angular frequency* is $\omega = 2\pi f$, and *phase* is ϕ_i. Note that in this section, we use the variables x and y to denote the spatial coordinates of an image, rather than i and j as in the previous sections. This is both because the letters i and j are used for the *imaginary* number (the usage depends on whether you are reading complex variables or electrical engineering literature) and because it is clearer how to distinguish the horizontal (x) and vertical (y) components in frequency space. In this section, we use the letter j for the imaginary number, since that is the form more commonly found in the signal processing literature (Bracewell 1986; Oppenheim and Schafer 1996; Oppenheim, Schafer, and Buck 1999).

[7] Moments can also be computed using Green's theorem applied to the boundary pixels (Yang and Albregtsen 1996).

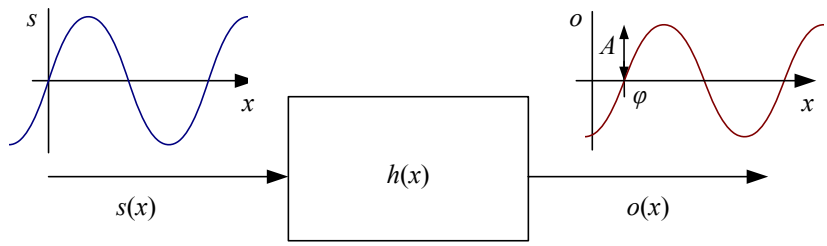


Figure 3.24 The Fourier Transform as the response of a filter $h(x)$ to an input sinusoid $s(x) = e^{j\omega x}$ yielding an output sinusoid $o(x) = h(x) * s(x) = Ae^{j\omega x+\phi}$.

If we convolve the sinusoidal signal $s(x)$ with a filter whose impulse response is $h(x)$, we get another sinusoid of the same frequency but different magnitude A and phase ϕ_o,

$$o(x) = h(x) * s(x) = A\sin(\omega x + \phi_o), \tag{3.48}$$

as shown in Figure 3.24. To see that this is the case, remember that a convolution can be expressed as a weighted summation of shifted input signals (3.14) and that the summation of a bunch of shifted sinusoids of the same frequency is just a single sinusoid at that frequency.[8] The new magnitude A is called the *gain* or *magnitude* of the filter, while the phase difference $\Delta\phi = \phi_o - \phi_i$ is called the *shift* or *phase*.

In fact, a more compact notation is to use the complex-valued sinusoid

$$s(x) = e^{j\omega x} = \cos\omega x + j\sin\omega x. \tag{3.49}$$

In that case, we can simply write,

$$o(x) = h(x) * s(x) = Ae^{j\omega x+\phi}. \tag{3.50}$$

The *Fourier transform* is simply a tabulation of the magnitude and phase response at each frequency,

$$H(\omega) = \mathcal{F}\{h(x)\} = Ae^{j\phi}, \tag{3.51}$$

i.e., it is the response to a complex sinusoid of frequency ω passed through the filter $h(x)$. The Fourier transform pair is also often written as

$$h(x) \overset{\mathcal{F}}{\leftrightarrow} H(\omega). \tag{3.52}$$

Unfortunately, (3.51) does not give an actual *formula* for computing the Fourier transform. Instead, it gives a *recipe*, i.e., convolve the filter with a sinusoid, observe the magnitude and phase shift, repeat. Fortunately, closed form equations for the Fourier transform exist both in the continuous domain,

$$H(\omega) = \int_{-\infty}^{\infty} h(x)e^{-j\omega x}dx, \tag{3.53}$$

[8] If h is a general (non-linear) transform, additional *harmonic* frequencies are introduced. This was traditionally the bane of audiophiles, who insisted on equipment with no *harmonic distortion*. Now that digital audio has introduced pure distortion-free sound, some audiophiles are buying retro tube amplifiers or digital signal processors that simulate such distortions because of their "warmer sound".

Property	Signal		Transform
superposition	$f_1(x) + f_2(x)$		$F_1(\omega) + F_2(\omega)$
shift	$f(x - x_0)$		$F(\omega)e^{-j\omega x_0}$
reversal	$f(-x)$		$F^*(\omega)$
convolution	$f(x) * h(x)$		$F(\omega)H(\omega)$
correlation	$f(x) \otimes h(x)$		$F(\omega)H^*(\omega)$
multiplication	$f(x)h(x)$		$F(\omega) * H(\omega)$
differentiation	$f'(x)$		$j\omega F(\omega)$
domain scaling	$f(ax)$		$1/aF(\omega/a)$
real images	$f(x) = f^*(x)$	$\Leftrightarrow$	$F(\omega) = F(-\omega)$
Parseval's Theorem	$\sum_x [f(x)]^2$	$=$	$\sum_\omega [F(\omega)]^2$

Table 3.1 Some useful properties of Fourier transforms. The original transform pair is $F(\omega) = \mathcal{F}\{f(x)\}$.

and in the discrete domain,

$$H(k) = \frac{1}{N}\sum_{x=0}^{N-1} h(x)e^{-j\frac{2\pi kx}{N}}, \tag{3.54}$$

where N is the length of the signal or region of analysis. These formulas apply both to filters, such as $h(x)$, and to signals or images, such as $s(x)$ or $g(x)$.

The discrete form of the Fourier transform (3.54) is known as the *Discrete Fourier Transform* (DFT). Note that while (3.54) can be evaluated for any value of k, it only makes sense for values in the range $k \in [-\frac{N}{2}, \frac{N}{2}]$. This is because larger values of k *alias* with lower frequencies and hence provide no additional information, as explained in the discussion on aliasing in Section 2.3.1.

At face value, the DFT takes $O(N^2)$ operations (multiply-adds) to evaluate. Fortunately, there exists a faster algorithm called the *Fast Fourier Transform* (FFT), which requires only $O(N \log_2 N)$ operations (Bracewell 1986; Oppenheim, Schafer, and Buck 1999). We do not explain the details of the algorithm here, except to say that it involves a series of $\log_2 N$ stages, where each stage performs small 2×2 transforms (matrix multiplications with known coefficients) followed by some semi-global permutations. (You will often see the term *butterfly* applied to these stages because of the pictorial shape of the signal processing graphs involved.) Implementations for the FFT can be found in most numerical and signal processing libraries.

Now that we have defined the Fourier transform, what are some of its properties and how can they be used? Table 3.1 lists a number of useful properties, which we describe in a little more detail below:

- **Superposition**: The Fourier transform of a sum of signals is the sum of their Fourier transforms. Thus, the Fourier transform is a linear operator.

- **Shift**: The Fourier transform of a shifted signal is the transform of the original signal multiplied by a *linear phase shift* (complex sinusoid).

- **Reversal**: The Fourier transform of a reversed signal is the complex conjugate of the signal's transform.
- **Convolution**: The Fourier transform of a pair of convolved signals is the product of their transforms.
- **Correlation**: The Fourier transform of a correlation is the product of the first transform times the complex conjugate of the second one.
- **Multiplication**: The Fourier transform of the product of two signals is the convolution of their transforms.
- **Differentiation**: The Fourier transform of the derivative of a signal is that signal's transform multiplied by the frequency. In other words, differentiation linearly emphasizes (magnifies) higher frequencies.
- **Domain scaling**: The Fourier transform of a stretched signal is the equivalently compressed (and scaled) version of the original transform and *vice versa*.
- **Real images**: The Fourier transform of a real-valued signal is symmetric around the origin. This fact can be used to save space and to double the speed of image FFTs by packing alternating scanlines into the real and imaginary parts of the signal being transformed.
- **Parseval's Theorem**: The energy (sum of squared values) of a signal is the same as the energy of its Fourier transform.

All of these properties are relatively straightforward to prove (see Exercise 3.15) and they will come in handy later in the book, e.g., when designing optimum Wiener filters (Section 3.4.3) or performing fast image correlations (Section 8.1.2).

3.4.1 Fourier transform pairs

Now that we have these properties in place, let us look at the Fourier transform pairs of some commonly occurring filters and signals, as listed in Table 3.2. In more detail, these pairs are as follows:

- **Impulse**: The impulse response has a constant (all frequency) transform.
- **Shifted impulse**: The shifted impulse has unit magnitude and linear phase.
- **Box filter**: The box (moving average) filter

$$\text{box}(x) = \begin{cases} 1 & \text{if } |x| \leq 1 \\ 0 & \text{else} \end{cases} \tag{3.55}$$

has a sinc Fourier transform,

$$\text{sinc}(\omega) = \frac{\sin \omega}{\omega}, \tag{3.56}$$

which has an infinite number of side lobes. Conversely, the sinc filter is an ideal low-pass filter. For a non-unit box, the width of the box a and the spacing of the zero crossings in the sinc $1/a$ are inversely proportional.

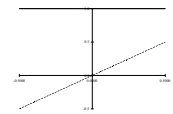
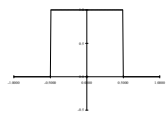
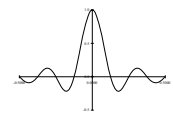
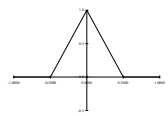
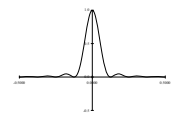
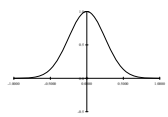
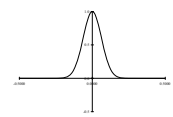
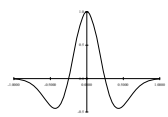
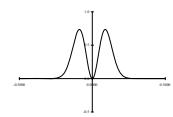
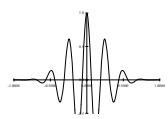
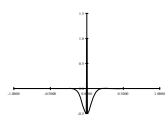
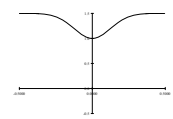
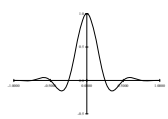
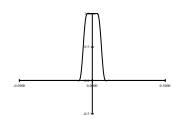
Name	Signal		Transform	
impulse		$\delta(x)$	$\Leftrightarrow$	1 
shifted impulse		$\delta(x-u)$	$\Leftrightarrow$	$e^{-j\omega u}$ 
box filter		$\text{box}(x/a)$	$\Leftrightarrow$	$a\text{sinc}(a\omega)$ 
tent		$\text{tent}(x/a)$	$\Leftrightarrow$	$a\text{sinc}^2(a\omega)$ 
Gaussian		$G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega;\sigma^{-1})$ 
Laplacian of Gaussian		$(\frac{x^2}{\sigma^4}-\frac{1}{\sigma^2})G(x;\sigma)$	$\Leftrightarrow$	$-\frac{\sqrt{2\pi}}{\sigma}\omega^2 G(\omega;\sigma^{-1})$ 
Gabor		$\cos(\omega_0 x)G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega\pm\omega_0;\sigma^{-1})$ 
unsharp mask		$(1+\gamma)\delta(x)$ $-\gamma G(x;\sigma)$	$\Leftrightarrow$	$(1+\gamma)-$ $\frac{\sqrt{2\pi}\gamma}{\sigma}G(\omega;\sigma^{-1})$ 
windowed sinc		$\text{rcos}(x/(aW))$ $\text{sinc}(x/a)$	$\Leftrightarrow$	(see Figure 3.29) 

Table 3.2 Some useful (continuous) Fourier transform pairs: The dashed line in the Fourier transform of the shifted impulse indicates its (linear) phase. All other transforms have zero phase (they are real-valued). Note that the figures are not necessarily drawn to scale but are drawn to illustrate the general shape and characteristics of the filter or its response. In particular, the Laplacian of Gaussian is drawn inverted because it resembles more a "Mexican hat", as it is sometimes called.

- **Tent**: The piecewise linear tent function,

$$\text{tent}(x) = \max(0, 1 - |x|), \tag{3.57}$$

  has a sinc^2 Fourier transform.

- **Gaussian**: The (unit area) Gaussian of width σ,

$$G(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \tag{3.58}$$

  has a (unit height) Gaussian of width σ^{-1} as its Fourier transform.

- **Laplacian of Gaussian**: The second derivative of a Gaussian of width σ,

$$LoG(x;\sigma) = (\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2})G(x;\sigma) \tag{3.59}$$

  has a band-pass response of

$$-\frac{\sqrt{2\pi}}{\sigma}\omega^2 G(\omega;\sigma^{-1}) \tag{3.60}$$

  as its Fourier transform.

- **Gabor**: The even Gabor function, which is the product of a cosine of frequency ω_0 and a Gaussian of width σ, has as its transform the sum of the two Gaussians of width σ^{-1} centered at $\omega = \pm\omega_0$. The odd Gabor function, which uses a sine, is the difference of two such Gaussians. Gabor functions are often used for oriented and band-pass filtering, since they can be more frequency selective than Gaussian derivatives.

- **Unsharp mask**: The unsharp mask introduced in (3.22) has as its transform a unit response with a slight boost at higher frequencies.

- **Windowed sinc**: The windowed (masked) sinc function shown in Table 3.2 has a response function that approximates an ideal low-pass filter better and better as additional side lobes are added (W is increased). Figure 3.29 shows the shapes of these such filters along with their Fourier transforms. For these examples, we use a one-lobe raised cosine,

$$\text{rcos}(x) = \frac{1}{2}(1 + \cos\pi x)\text{box}(x), \tag{3.61}$$

  also known as the *Hann window*, as the windowing function. Wolberg (1990) and Oppenheim, Schafer, and Buck (1999) discuss additional windowing functions, which include the *Lanczos* window, the positive first lobe of a sinc function.

We can also compute the Fourier transforms for the small discrete kernels shown in Figure 3.14 (see Table 3.3). Notice how the moving average filters do not uniformly dampen higher frequencies and hence can lead to ringing artifacts. The binomial filter (Gomes and Velho 1997) used as the "Gaussian" in Burt and Adelson's (1983a) Laplacian pyramid (see Section 3.5), does a decent job of separating the high and low frequencies, but still leaves a fair amount of high-frequency detail, which can lead to aliasing after downsampling. The Sobel edge detector at first linearly accentuates frequencies, but then decays at higher frequencies, and hence has trouble detecting fine-scale edges, e.g., adjacent black and white columns. We look at additional examples of small kernel Fourier transforms in Section 3.5.2, where we study better kernels for pre-filtering before decimation (size reduction).

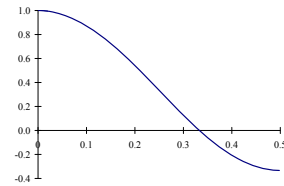
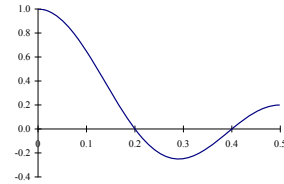
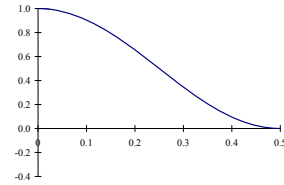
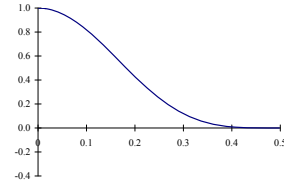
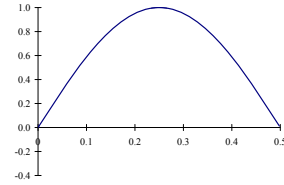
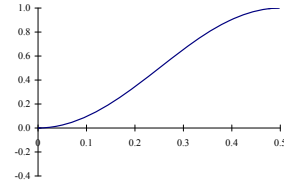
Name	Kernel	Transform	Plot
box-3	$\frac{1}{3}\begin{bmatrix}1&1&1\end{bmatrix}$	$\frac{1}{3}(1+2\cos\omega)$	
box-5	$\frac{1}{5}\begin{bmatrix}1&1&1&1&1\end{bmatrix}$	$\frac{1}{5}(1+2\cos\omega+2\cos 2\omega)$	
linear	$\frac{1}{4}\begin{bmatrix}1&2&1\end{bmatrix}$	$\frac{1}{2}(1+\cos\omega)$	
binomial	$\frac{1}{16}\begin{bmatrix}1&4&6&4&1\end{bmatrix}$	$\frac{1}{4}(1+\cos\omega)^2$	
Sobel	$\frac{1}{2}\begin{bmatrix}-1&0&1\end{bmatrix}$	$\sin\omega$	
corner	$\frac{1}{2}\begin{bmatrix}-1&2&-1\end{bmatrix}$	$\frac{1}{2}(1-\cos\omega)$	

Table 3.3 Fourier transforms of the separable kernels shown in Figure 3.14.

3.4.2 Two-dimensional Fourier transforms

The formulas and insights we have developed for one-dimensional signals and their transforms translate directly to two-dimensional images. Here, instead of just specifying a horizontal or vertical frequency ω_x or ω_y, we can create an oriented sinusoid of frequency (ω_x, ω_y),

$$s(x, y) = \sin(\omega_x x + \omega_y y). \tag{3.62}$$

The corresponding two-dimensional Fourier transforms are then

$$H(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y) e^{-j(\omega_x x + \omega_y y)} dx\, dy, \tag{3.63}$$

and in the discrete domain,

$$H(k_x, k_y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} h(x, y) e^{-j2\pi \frac{k_x x + k_y y}{MN}}, \tag{3.64}$$

where M and N are the width and height of the image.

All of the Fourier transform properties from Table 3.1 carry over to two dimensions if we replace the scalar variables x, ω, x_0 and a with their 2D vector counterparts $\boldsymbol{x} = (x, y)$, $\boldsymbol{\omega} = (\omega_x, \omega_y)$, $\boldsymbol{x}_0 = (x_0, y_0)$, and $\boldsymbol{a} = (a_x, a_y)$, and use vector inner products instead of multiplications.

3.4.3 Wiener filtering

While the Fourier transform is a useful tool for analyzing the frequency characteristics of a filter kernel or image, it can also be used to analyze the frequency spectrum of a whole *class* of images.

A simple model for images is to assume that they are random noise fields whose expected magnitude at each frequency is given by this *power spectrum* $P_s(\omega_x, \omega_y)$, i.e.,

$$\left\langle [S(\omega_x, \omega_y)]^2 \right\rangle = P_s(\omega_x, \omega_y), \tag{3.65}$$

where the angle brackets $\langle \cdot \rangle$ denote the expected (mean) value of a random variable.[9] To generate such an image, we simply create a random Gaussian noise image $S(\omega_x, \omega_y)$ where each "pixel" is a zero-mean Gaussian[10] of variance $P_s(\omega_x, \omega_y)$ and then take its inverse FFT.

The observation that signal spectra capture a first-order description of spatial statistics is widely used in signal and image processing. In particular, assuming that an image is a sample from a correlated Gaussian random noise field combined with a statistical model of the measurement process yields an optimum restoration filter known as the *Wiener filter*.[11]

To derive the Wiener filter, we analyze each frequency component of a signal's Fourier transform independently. The noisy image formation process can be written as

$$o(x, y) = s(x, y) + n(x, y), \tag{3.66}$$

[9] The notation $E[\cdot]$ is also commonly used.

[10] We set the DC (i.e., constant) component at $S(0, 0)$ to the mean grey level. See Algorithm C.1 in Appendix C.2 for code to generate Gaussian noise.

[11] Wiener is pronounced "veener" since, in German, the "w" is pronounced "v". Remember that next time you order "Wiener schnitzel".

where $s(x, y)$ is the (unknown) image we are trying to recover, $n(x, y)$ is the additive noise signal, and $o(x, y)$ is the *observed* noisy image. Because of the linearity of the Fourier transform, we can write

$$O(\omega_x, \omega_y) = S(\omega_x, \omega_y) + N(\omega_x, \omega_y), \tag{3.67}$$

where each quantity in the above equation is the Fourier transform of the corresponding image.

At each frequency (ω_x, ω_y), we know from our image spectrum that the unknown transform component $S(\omega_x, \omega_y)$ has a *prior* distribution which is a zero-mean Gaussian with variance $P_s(\omega_x, \omega_y)$. We also have noisy measurement $O(\omega_x, \omega_y)$ whose variance is $P_n(\omega_x, \omega_y)$, i.e., the power spectrum of the noise, which is usually assumed to be constant (white), $P_n(\omega_x, \omega_y) = \sigma_n^2$.

According to Bayes' Rule (Appendix B.4), the *posterior estimate* of S can be written as

$$p(S|O) = \frac{p(O|S)p(S)}{p(O)}, \tag{3.68}$$

where $p(O) = \int_S p(O|S)p(S)$ is a normalizing constant used to make the $p(S|O)$ distribution *proper* (integrate to 1). The prior distribution $p(S)$ is given by

$$p(S) = e^{-\frac{(S-\mu)^2}{2P_s}}, \tag{3.69}$$

where μ is the expected mean at that frequency (0 everywhere except at the origin) and the measurement distribution $P(O|S)$ is given by

$$p(S) = e^{-\frac{(S-O)^2}{2P_n}}. \tag{3.70}$$

Taking the negative logarithm of both sides of (3.68) and setting $\mu = 0$ for simplicity, we get

$$\begin{aligned} -\log p(S|O) &= -\log p(O|S) - \log p(S) + C & (3.71)\\ &= {}^1\!/_2 P_n^{-1}(S-O)^2 + {}^1\!/_2 P_s^{-1} S^2 + C, & (3.72) \end{aligned}$$

which is the *negative posterior log likelihood*. The minimum of this quantity is easy to compute,

$$S_{\text{opt}} = \frac{P_n^{-1}}{P_n^{-1} + P_s^{-1}} O = \frac{P_s}{P_s + P_n} O = \frac{1}{1 + P_n/P_s} O. \tag{3.73}$$

The quantity

$$W(\omega_x, \omega_y) = \frac{1}{1 + \sigma_n^2 / P_s(\omega_x, \omega_y)} \tag{3.74}$$

is the Fourier transform of the optimum *Wiener filter* needed to remove the noise from an image whose power spectrum is $P_s(\omega_x, \omega_y)$.

Notice that this filter has the right qualitative properties, i.e., for low frequencies where $P_s \gg \sigma_n^2$, it has unit gain, whereas for high frequencies, it attenuates the noise by a factor P_s/σ_n^2. Figure 3.25 shows the one-dimensional transform $W(f)$ and the corresponding filter kernel $w(x)$ for the commonly assumed case of $P(f) = f^{-2}$ (Field 1987). Exercise 3.16 has you compare the Wiener filter as a denoising algorithm to hand-tuned Gaussian smoothing.

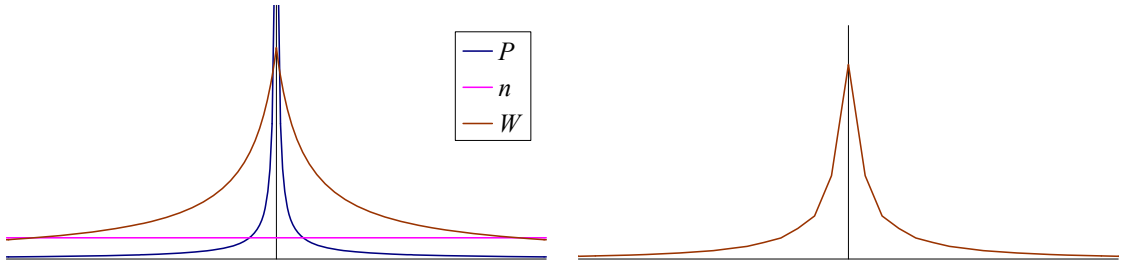


Figure 3.25 One-dimensional Wiener filter: (a) power spectrum of signal $P_s(f)$, noise level σ^2, and Wiener filter transform $W(f)$; (b) Wiener filter spatial kernel.

The methodology given above for deriving the Wiener filter can easily be extended to the case where the observed image is a noisy blurred version of the original image,

$$o(x,y) = b(x,y) * s(x,y) + n(x,y), \tag{3.75}$$

where $b(x,y)$ is the known blur kernel. Rather than deriving the corresponding Wiener filter, we leave it as an exercise (Exercise 3.17), which also encourages you to compare your de-blurring results with unsharp masking and naïve inverse filtering. More sophisticated algorithms for blur removal are discussed in Sections 3.7 and 10.3.

Discrete cosine transform

The *discrete cosine transform* (DCT) is a variant of the Fourier transform particularly well-suited to compressing images in a block-wise fashion. The one-dimensional DCT is computed by taking the dot product of each N-wide block of pixels with a set of cosines of different frequencies,

$$F(k) = \sum_{i=0}^{N-1} \cos\left(\frac{\pi}{N}(i+\frac{1}{2})k\right) f(i), \tag{3.76}$$

where k is the coefficient (frequency) index, and the $1/2$-pixel offset is used to make the basis coefficients symmetric (Wallace 1991). Some of the discrete cosine basis functions are shown in Figure 3.26. As you can see, the first basis function (the straight blue line) encodes the average DC value in the block of pixels, while the second encodes a slightly curvy version of the slope.

In turns out that the DCT is a good approximation to the optimal Karhunen–Loève decomposition of natural image statistics over small patches, which can be obtained by performing a principal component analysis (PCA) of images, as described in Section 14.2.1. The KL-transform de-correlates the signal optimally (assuming the signal is described by its spectrum) and thus, theoretically, leads to optimal compression.

The two-dimensional version of the DCT is defined similarly,

$$F(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \cos\left(\frac{\pi}{N}(i+\frac{1}{2})k\right) \cos\left(\frac{\pi}{N}(j+\frac{1}{2})l\right) f(i,j). \tag{3.77}$$

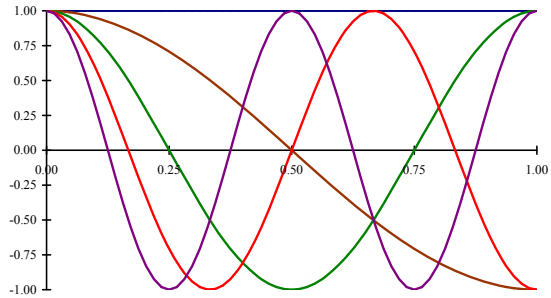


Figure 3.26 Discrete cosine transform (DCT) basis functions: The first DC (i.e., constant) basis is the horizontal blue line, the second is the brown half-cycle waveform, etc. These bases are widely used in image and video compression standards such as JPEG.

Like the 2D Fast Fourier Transform, the 2D DCT can be implemented separably, i.e., first computing the DCT of each line in the block and then computing the DCT of each resulting column. Like the FFT, each of the DCTs can also be computed in $O(N \log N)$ time.

As we mentioned in Section 2.3.3, the DCT is widely used in today's image and video compression algorithms, although it is slowly being supplanted by wavelet algorithms (Simoncelli and Adelson 1990b), as discussed in Section 3.5.4, and overlapped variants of the DCT (Malvar 1990, 1998, 2000), which are used in the new JPEG XR standard.[12] These newer algorithms suffer less from the *blocking artifacts* (visible edge-aligned discontinuities) that result from the pixels in each block (typically 8×8) being transformed and quantized independently. See Exercise 3.30 for ideas on how to remove blocking artifacts from compressed JPEG images.

3.4.4 *Application*: Sharpening, blur, and noise removal

Another common application of image processing is the enhancement of images through the use of sharpening and noise removal operations, which require some kind of neighborhood processing. Traditionally, these kinds of operation were performed using linear filtering (see Sections 3.2 and Section 3.4.3). Today, it is more common to use non-linear filters (Section 3.3.1), such as the weighted median or bilateral filter (3.34–3.37), anisotropic diffusion (3.39–3.40), or non-local means (Buades, Coll, and Morel 2008). Variational methods (Section 3.7.1), especially those using non-quadratic (robust) norms such as the L_1 norm (which is called *total variation*), are also often used. Figure 3.19 shows some examples of linear and non-linear filters being used to remove noise.

When measuring the effectiveness of image denoising algorithms, it is common to report the results as a *peak signal-to-noise ratio (PSNR)* measurement (2.119), where $I(\boldsymbol{x})$ is the original (noise-free) image and $\hat{I}(\boldsymbol{x})$ is the image after denoising; this is for the case where the noisy image has been synthetically generated, so that the clean image is known. A better way to measure the quality is to use a perceptually based similarity metric, such as the structural similarity (SSIM) index (Wang, Bovik, Sheikh *et al.* 2004; Wang, Bovik, and Simoncelli 2005).

[12] http://www.itu.int/rec/T-REC-T.832-200903-I/en.

Exercises 3.11, 3.16, 3.17, 3.21, and 3.28 have you implement some of these operations and compare their effectiveness. More sophisticated techniques for blur removal and the related task of super-resolution are discussed in Section 10.3.

3.5 Pyramids and wavelets

So far in this chapter, all of the image transformations we have studied produce output images of the same size as the inputs. Often, however, we may wish to change the resolution of an image before proceeding further. For example, we may need to interpolate a small image to make its resolution match that of the output printer or computer screen. Alternatively, we may want to reduce the size of an image to speed up the execution of an algorithm or to save on storage space or transmission time.

Sometimes, we do not even know what the appropriate resolution for the image should be. Consider, for example, the task of finding a face in an image (Section 14.1.1). Since we do not know the scale at which the face will appear, we need to generate a whole *pyramid* of differently sized images and scan each one for possible faces. (Biological visual systems also operate on a hierarchy of scales (Marr 1982).) Such a pyramid can also be very helpful in accelerating the search for an object by first finding a smaller instance of that object at a coarser level of the pyramid and then looking for the full resolution object only in the vicinity of coarse-level detections (Section 8.1.1). Finally, image pyramids are extremely useful for performing multi-scale editing operations such as blending images while maintaining details.

In this section, we first discuss good filters for changing image resolution, i.e., upsampling (*interpolation*, Section 3.5.1) and downsampling (*decimation*, Section 3.5.2). We then present the concept of multi-resolution pyramids, which can be used to create a complete hierarchy of differently sized images and to enable a variety of applications (Section 3.5.3). A closely related concept is that of *wavelets*, which are a special kind of pyramid with higher frequency selectivity and other useful properties (Section 3.5.4). Finally, we present a useful application of pyramids, namely the blending of different images in a way that hides the seams between the image boundaries (Section 3.5.5).

3.5.1 Interpolation

In order to *interpolate* (or *upsample*) an image to a higher resolution, we need to select some interpolation kernel with which to convolve the image,

$$g(i,j) = \sum_{k,l} f(k,l)h(i - rk, j - rl). \tag{3.78}$$

This formula is related to the discrete convolution formula (3.14), except that we replace k and l in $h()$ with rk and rl, where r is the upsampling rate. Figure 3.27a shows how to think of this process as the superposition of sample weighted interpolation kernels, one centered at each input sample k. An alternative mental model is shown in Figure 3.27b, where the kernel is centered at the output pixel value i (the two forms are equivalent). The latter form is sometimes called the *polyphase filter* form, since the kernel values $h(i)$ can be stored as r separate kernels, each of which is selected for convolution with the input samples depending on the *phase* of i relative to the upsampled grid.

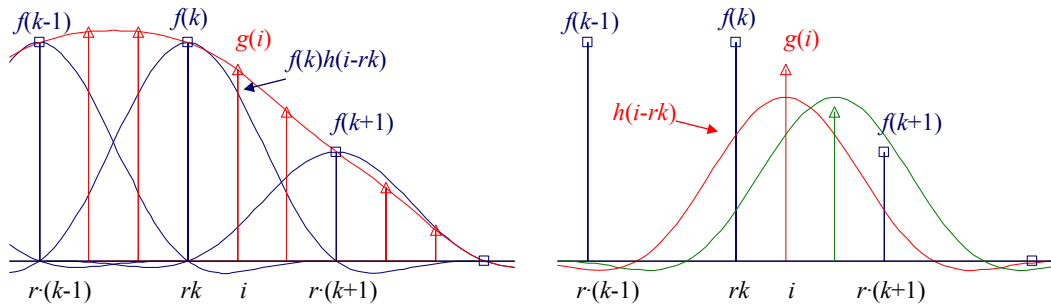


Figure 3.27 Signal interpolation, $g(i) = \sum_k f(k)h(i - rk)$: (a) weighted summation of input values; (b) polyphase filter interpretation.

What kinds of kernel make good interpolators? The answer depends on the application and the computation time involved. Any of the smoothing kernels shown in Tables 3.2 and 3.3 can be used after appropriate re-scaling.[13] The *linear* interpolator (corresponding to the tent kernel) produces interpolating piecewise linear curves, which result in unappealing *creases* when applied to images (Figure 3.28a). The cubic B-spline, whose discrete $1/2$-pixel sampling appears as the *binomial kernel* in Table 3.3, is an *approximating* kernel (the interpolated image does not pass through the input data points) that produces soft images with reduced high-frequency detail. The equation for the cubic B-spline is easiest to derive by convolving the tent function (linear B-spline) with itself.

While most graphics cards use the bilinear kernel (optionally combined with a MIP-map—see Section 3.5.3), most photo editing packages use *bicubic* interpolation. The cubic interpolant is a C^1 (derivative-continuous) piecewise-cubic *spline* (the term "spline" is synonymous with "piecewise-polynomial")[14] whose equation is

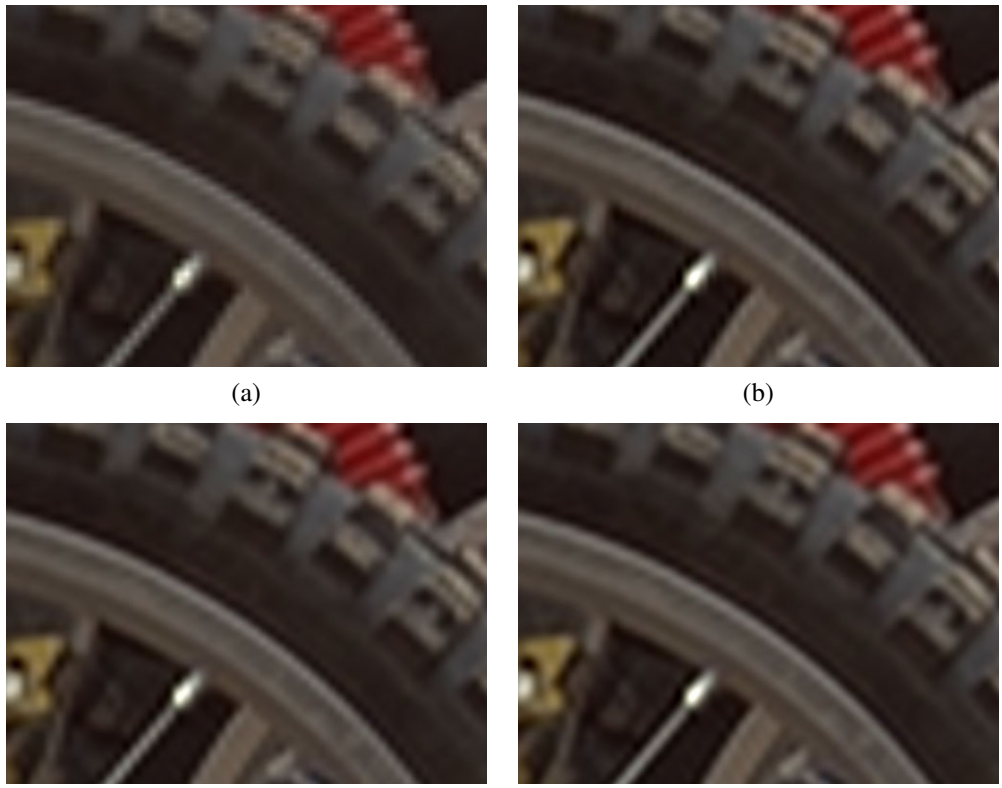
$$h(x) = \begin{cases} 1 - (a+3)x^2 + (a+2)|x|^3 & \text{if } |x| < 1 \\ a(|x|-1)(|x|-2)^2 & \text{if } 1 \le |x| < 2 \\ 0 & \text{otherwise,} \end{cases} \tag{3.79}$$

where a specifies the derivative at $x = 1$ (Parker, Kenyon, and Troxel 1983). The value of a is often set to -1, since this best matches the frequency characteristics of a sinc function (Figure 3.29). It also introduces a small amount of sharpening, which can be visually appealing. Unfortunately, this choice does not linearly interpolate straight lines (intensity ramps), so some visible ringing may occur. A better choice for large amounts of interpolation is probably $a = -0.5$, which produces a *quadratic reproducing* spline; it interpolates linear and quadratic functions exactly (Wolberg 1990, Section 5.4.3). Figure 3.29 shows the $a = -1$ and $a = -0.5$ cubic interpolating kernel along with their Fourier transforms; Figure 3.28b and c shows them being applied to two-dimensional interpolation.

Splines have long been used for function and data value interpolation because of the abil-

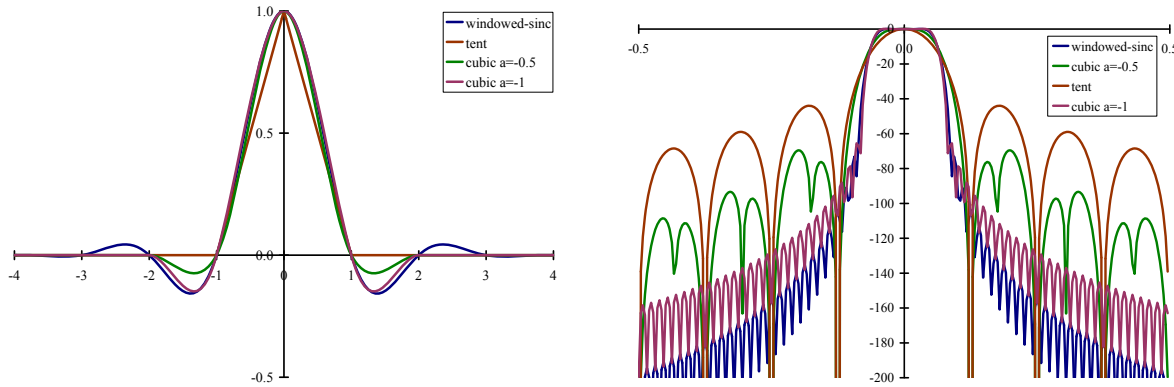
[13] The smoothing kernels in Table 3.3 have a unit area. To turn them into interpolating kernels, we simply scale them up by the interpolation rate r.

[14] The term "spline" comes from the draughtsman's workshop, where it was the name of a flexible piece of wood or metal used to draw smooth curves.



(a) (b) (c) (d)

Figure 3.28 Two-dimensional image interpolation: (a) bilinear; (b) bicubic ($a = -1$); (c) bicubic ($a = -0.5$); (d) windowed sinc (nine taps).



(a) (b)

Figure 3.29 (a) Some windowed sinc functions and (b) their log Fourier transforms: raised-cosine windowed sinc in blue, cubic interpolators ($a = -1$ and $a = -0.5$) in green and purple, and tent function in brown. They are often used to perform high-accuracy low-pass filtering operations.

ity to precisely specify derivatives at control points and efficient *incremental* algorithms for their evaluation (Bartels, Beatty, and Barsky 1987; Farin 1992, 1996). Splines are widely used in geometric modeling and computer-aided design (CAD) applications, although they have started being displaced by subdivision surfaces (Zorin, Schröder, and Sweldens 1996; Peters and Reif 2008). In computer vision, splines are often used for elastic image deformations (Section 3.6.2), motion estimation (Section 8.3), and surface interpolation (Section 12.3). In fact, it is possible to carry out most image processing operations by representing images as splines and manipulating them in a multi-resolution framework (Unser 1999).

The highest quality interpolator is generally believed to be the windowed sinc function because it both preserves details in the lower resolution image and avoids aliasing. (It is also possible to construct a C^1 piecewise-cubic approximation to the windowed sinc by matching its derivatives at zero crossing (Szeliski and Ito 1986).) However, some people object to the excessive *ringing* that can be introduced by the windowed sinc and to the repetitive nature of the ringing frequencies (see Figure 3.28d). For this reason, some photographers prefer to repeatedly interpolate images by a small fractional amount (this tends to de-correlate the original pixel grid with the final image). Additional possibilities include using the bilateral filter as an interpolator (Kopf, Cohen, Lischinski *et al.* 2007), using global optimization (Section 3.6) or hallucinating details (Section 10.3).

3.5.2 Decimation

While interpolation can be used to increase the resolution of an image, decimation (downsampling) is required to reduce the resolution.[15] To perform decimation, we first (conceptually) convolve the image with a low-pass filter (to avoid aliasing) and then keep every rth sample. In practice, we usually only evaluate the convolution at every rth sample,

$$g(i,j) = \sum_{k,l} f(k,l)h(ri-k, rj-l), \tag{3.80}$$

as shown in Figure 3.30. Note that the smoothing kernel $h(k,l)$, in this case, is often a stretched and re-scaled version of an interpolation kernel. Alternatively, we can write

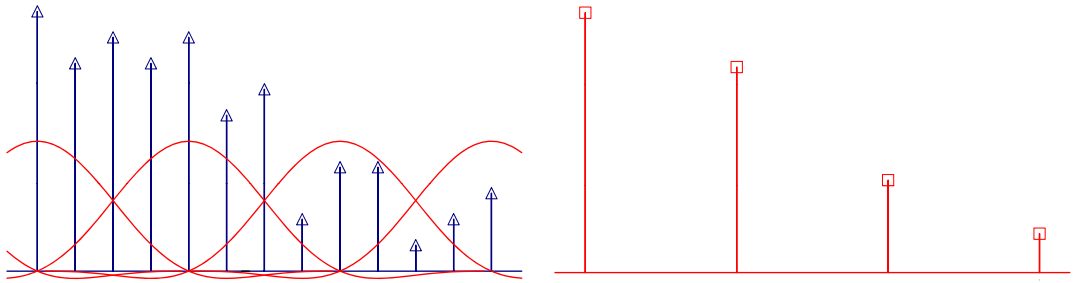
$$g(i,j) = \frac{1}{r}\sum_{k,l} f(k,l)h(i-k/r, j-l/r) \tag{3.81}$$

and keep the same kernel $h(k,l)$ for both interpolation and decimation.

One commonly used ($r = 2$) decimation filter is the *binomial* filter introduced by Burt and Adelson (1983a). As shown in Table 3.3, this kernel does a decent job of separating the high and low frequencies, but still leaves a fair amount of high-frequency detail, which can lead to aliasing after downsampling. However, for applications such as image blending (discussed later in this section), this aliasing is of little concern.

If, however, the downsampled images will be displayed directly to the user or, perhaps, blended with other resolutions (as in MIP-mapping, Section 3.5.3), a higher-quality filter is

[15] The term "decimation" has a gruesome etymology relating to the practice of killing every tenth soldier in a Roman unit guilty of cowardice. It is generally used in signal processing to mean any downsampling or rate reduction operation.



(a) (b)

Figure 3.30 Signal decimation: (a) the original samples are (b) convolved with a low-pass filter before being downsampled.

desired. For high downsampling rates, the windowed sinc pre-filter is a good choice (Figure 3.29). However, for small downsampling rates, e.g., $r = 2$, more careful filter design is required.

Table 3.4 shows a number of commonly used $r = 2$ downsampling filters, while Figure 3.31 shows their corresponding frequency responses. These filters include:

- the linear $[1, 2, 1]$ filter gives a relatively poor response;
- the binomial $[1, 4, 6, 4, 1]$ filter cuts off a lot of frequencies but is useful for computer vision analysis pyramids;
- the cubic filters from (3.79); the $a = -1$ filter has a sharper fall-off than the $a = -0.5$ filter (Figure 3.31);
- a cosine-windowed sinc function (Table 3.2);
- the QMF-9 filter of Simoncelli and Adelson (1990b) is used for wavelet denoising and aliases a fair amount (note that the original filter coefficients are normalized to $\sqrt{2}$ gain so they can be "self-inverting");
- the 9/7 analysis filter from JPEG 2000 (Taubman and Marcellin 2002).

Please see the original papers for the full-precision values of some of these coefficients.

$\|n\|$	Linear	Binomial	Cubic $a = -1$	Cubic $a = -0.5$	Windowed sinc	QMF-9	JPEG 2000
0	0.50	0.3750	0.5000	0.50000	0.4939	0.5638	0.6029
1	0.25	0.2500	0.3125	0.28125	0.2684	0.2932	0.2669
2		0.0625	0.0000	0.00000	0.0000	-0.0519	-0.0782
3			-0.0625	-0.03125	-0.0153	-0.0431	-0.0169
4					0.0000	0.0198	0.0267

Table 3.4 Filter coefficients for $2\times$ decimation. These filters are of odd length, are symmetric, and are normalized to have unit DC gain (sum up to 1). See Figure 3.31 for their associated frequency responses.

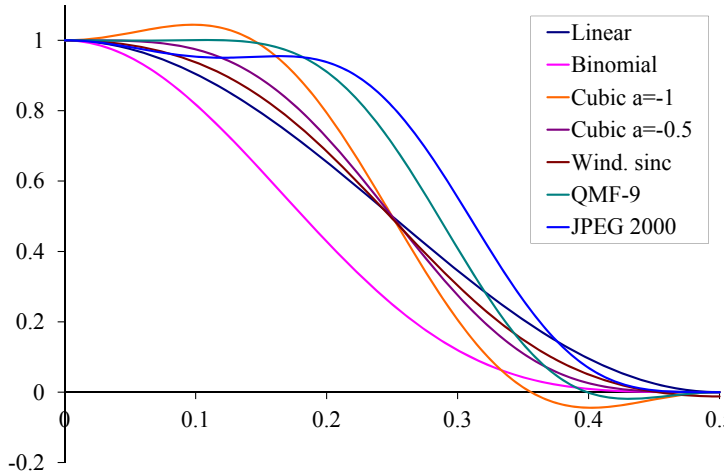


Figure 3.31 Frequency response for some $2\times$ decimation filters. The cubic $a = -1$ filter has the sharpest fall-off but also a bit of ringing; the wavelet analysis filters (QMF-9 and JPEG 2000), while useful for compression, have more aliasing.

3.5.3 Multi-resolution representations

Now that we have described interpolation and decimation algorithms, we can build a complete image pyramid (Figure 3.32). As we mentioned before, pyramids can be used to accelerate coarse-to-fine search algorithms, to look for objects or patterns at different scales, and to perform multi-resolution blending operations. They are also widely used in computer graphics hardware and software to perform fractional-level decimation using the MIP-map, which we cover in Section 3.6.

The best known (and probably most widely used) pyramid in computer vision is Burt and Adelson's (1983a) Laplacian pyramid. To construct the pyramid, we first blur and subsample the original image by a factor of two and store this in the next level of the pyramid (Figure 3.33). Because adjacent levels in the pyramid are related by a sampling rate $r = 2$, this kind of pyramid is known as an *octave pyramid*. Burt and Adelson originally proposed a five-tap kernel of the form

$$\begin{array}{|c|c|c|c|c|}\hline c & b & a & b & c \\ \hline\end{array}, \tag{3.82}$$

with $b = 1/4$ and $c = 1/4 - a/2$. In practice, $a = 3/8$, which results in the familiar binomial kernel,

$$\frac{1}{16}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline\end{array}, \tag{3.83}$$

which is particularly easy to implement using shifts and adds. (This was important in the days when multipliers were expensive.) The reason they call their resulting pyramid a *Gaussian* pyramid is that repeated convolutions of the binomial kernel converge to a Gaussian.[16]

To compute the *Laplacian* pyramid, Burt and Adelson first interpolate a lower resolution image to obtain a *reconstructed* low-pass version of the original image (Figure 3.34b). They then subtract this low-pass version from the original to yield the band-pass "Laplacian"

[16] Then again, this is true for any smoothing kernel (Wells 1986).

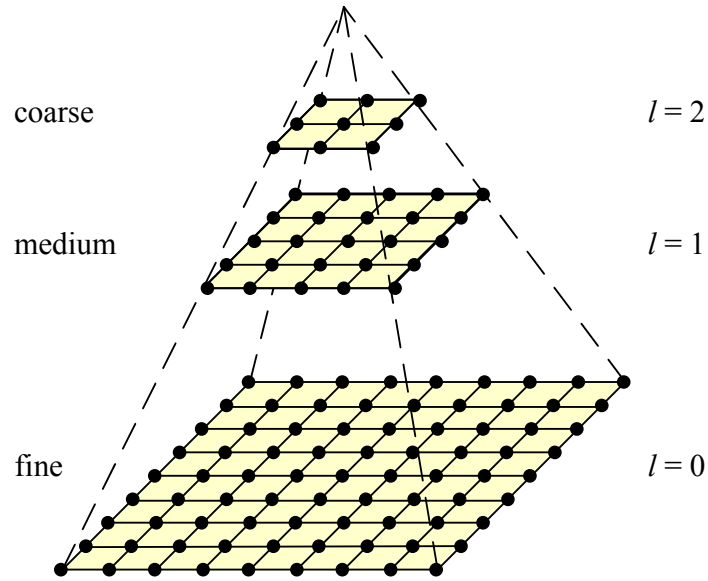


Figure 3.32 A traditional image pyramid: each level has half the resolution (width and height), and hence a quarter of the pixels, of its parent level.

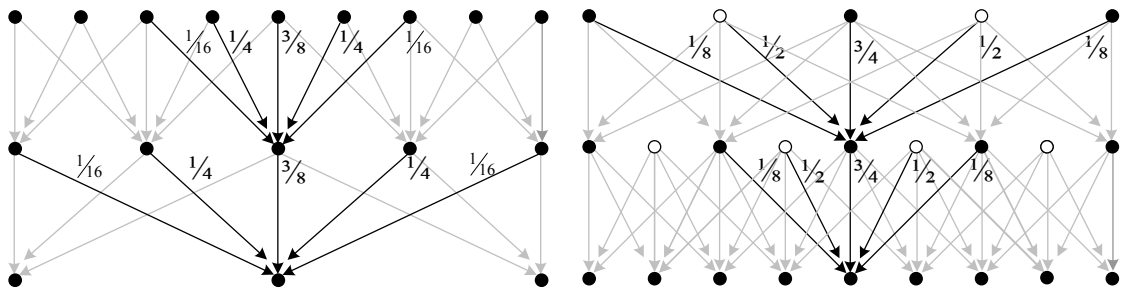


Figure 3.33 The Gaussian pyramid shown as a signal processing diagram: The (a) analysis and (b) re-synthesis stages are shown as using similar computations. The white circles indicate zero values inserted by the $\uparrow 2$ upsampling operation. Notice how the reconstruction filter coefficients are twice the analysis coefficients. The computation is shown as flowing down the page, regardless of whether we are going from coarse to fine or *vice versa*.

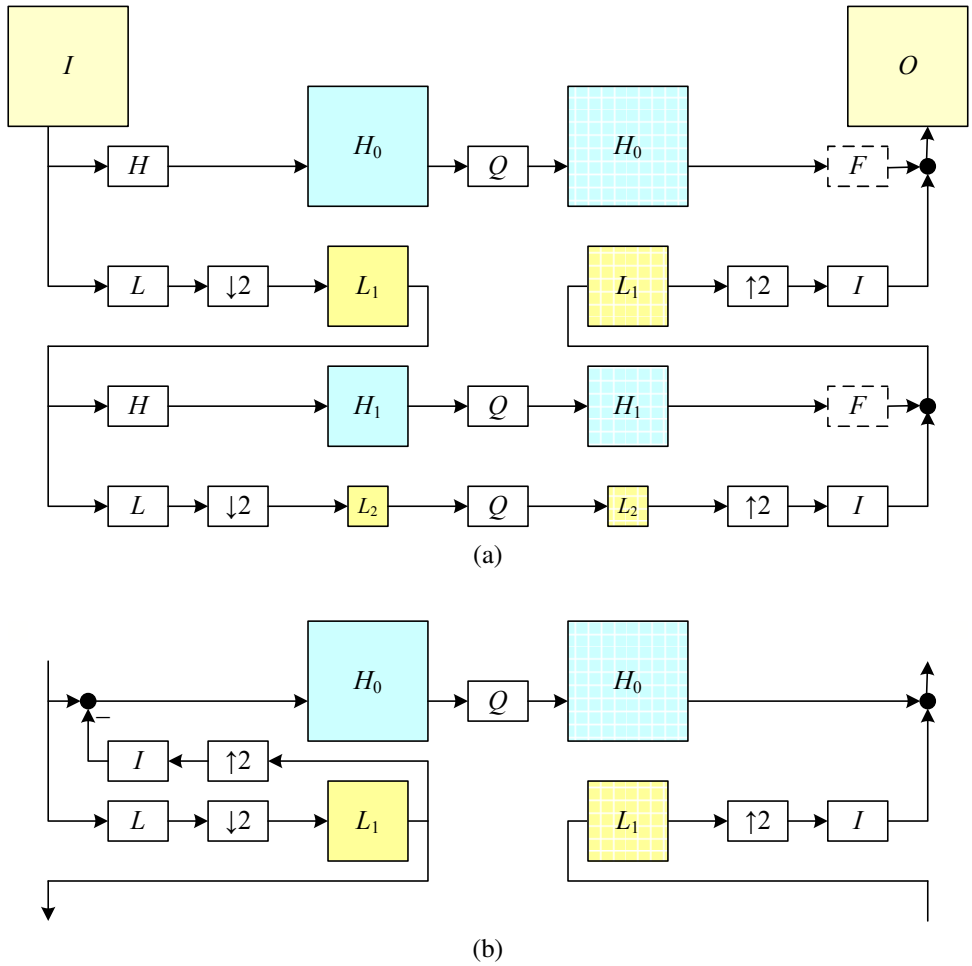


Figure 3.34 The Laplacian pyramid: (a) The conceptual flow of images through processing stages: images are high-pass and low-pass filtered, and the low-pass filtered images are processed in the next stage of the pyramid. During reconstruction, the interpolated image and the (optionally filtered) high-pass image are added back together. The Q box indicates quantization or some other pyramid processing, e.g., noise removal by *coring* (setting small wavelet values to 0). (b) The actual computation of the high-pass filter involves first interpolating the downsampled low-pass image and then subtracting it. This results in perfect reconstruction when Q is the identity. The high-pass (or band-pass) images are typically called *Laplacian* images, while the low-pass images are called *Gaussian* images.

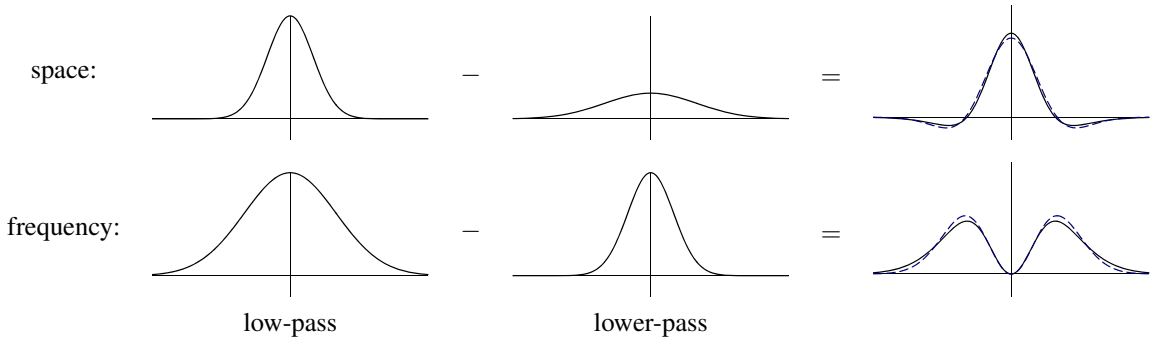


Figure 3.35 The difference of two low-pass filters results in a band-pass filter. The dashed blue lines show the close fit to a half-octave Laplacian of Gaussian.

image, which can be stored away for further processing. The resulting pyramid has *perfect reconstruction*, i.e., the Laplacian images plus the base-level Gaussian (L_2 in Figure 3.34b) are sufficient to exactly reconstruct the original image. Figure 3.33 shows the same computation in one dimension as a signal processing diagram, which completely captures the computations being performed during the analysis and re-synthesis stages.

Burt and Adelson also describe a variant on the Laplacian pyramid, where the low-pass image is taken from the original blurred image rather than the reconstructed pyramid (piping the output of the L box directly to the subtraction in Figure 3.34b). This variant has less aliasing, since it avoids one downsampling and upsampling round-trip, but it is not self-inverting, since the Laplacian images are no longer adequate to reproduce the original image.

As with the Gaussian pyramid, the term Laplacian is a bit of a misnomer, since their band-pass images are really differences of (approximate) Gaussians, or DoGs,

$$\mathrm{DoG}\{I;\sigma_1,\sigma_2\} = G_{\sigma_1} * I - G_{\sigma_2} * I = (G_{\sigma_1} - G_{\sigma_2}) * I. \tag{3.84}$$

A Laplacian of Gaussian (which we saw in (3.26)) is actually its second derivative,

$$\mathrm{LoG}\{I;\sigma\} = \nabla^2(G_\sigma * I) = (\nabla^2 G_\sigma) * I, \tag{3.85}$$

where

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{3.86}$$

is the Laplacian (operator) of a function. Figure 3.35 shows how the Differences of Gaussian and Laplacians of Gaussian look in both space and frequency.

Laplacians of Gaussian have elegant mathematical properties, which have been widely studied in the *scale-space* community (Witkin 1983; Witkin, Terzopoulos, and Kass 1986; Lindeberg 1990; Nielsen, Florack, and Deriche 1997) and can be used for a variety of applications including edge detection (Marr and Hildreth 1980; Perona and Malik 1990b), stereo matching (Witkin, Terzopoulos, and Kass 1987), and image enhancement (Nielsen, Florack, and Deriche 1997).

A less widely used variant is *half-octave pyramids*, shown in Figure 3.36a. These were first introduced to the vision community by Crowley and Stern (1984), who call them *Difference of Low-Pass* (DOLP) transforms. Because of the small scale change between adja-

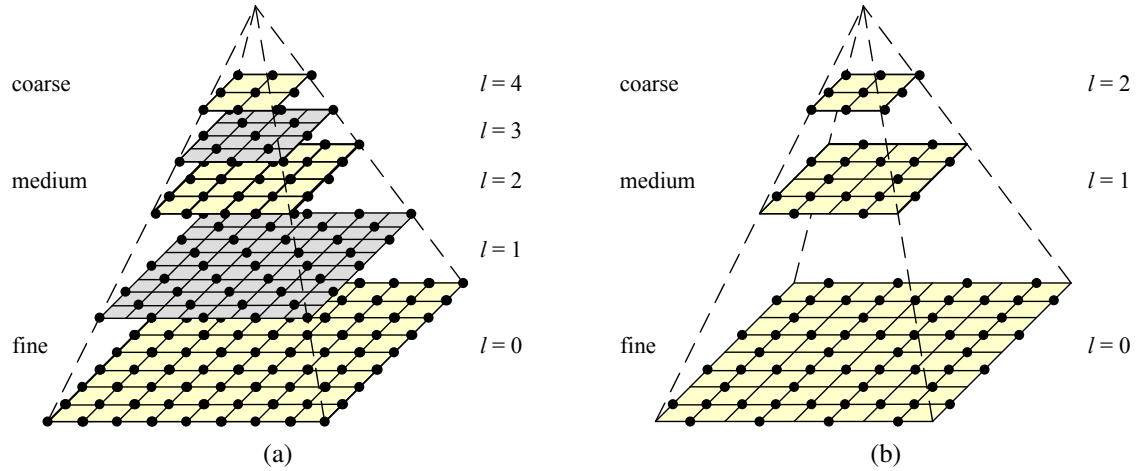


Figure 3.36 Multiresolution pyramids: (a) pyramid with half-octave (*quincunx*) sampling (odd levels are colored gray for clarity). (b) wavelet pyramid—each wavelet level stores $3/4$ of the original pixels (usually the horizontal, vertical, and mixed gradients), so that the total number of wavelet coefficients and original pixels is the same.

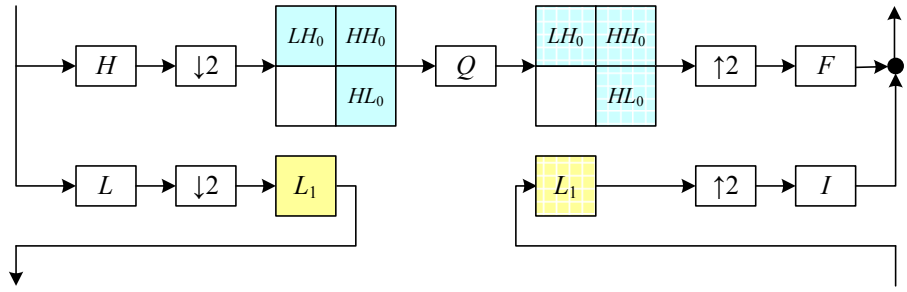
cent levels, the authors claim that coarse-to-fine algorithms perform better. In the image-processing community, half-octave pyramids combined with checkerboard sampling grids are known as *quincunx* sampling (Feilner, Van De Ville, and Unser 2005). In detecting multi-scale features (Section 4.1.1), it is often common to use half-octave or even quarter-octave pyramids (Lowe 2004; Triggs 2004). However, in this case, the subsampling only occurs at every octave level, i.e., the image is repeatedly blurred with wider Gaussians until a full octave of resolution change has been achieved (Figure 4.11).

3.5.4 Wavelets

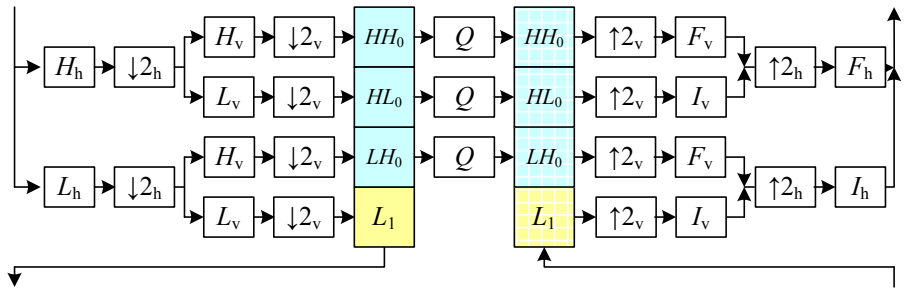
While pyramids are used extensively in computer vision applications, some people use *wavelet* decompositions as an alternative. Wavelets are filters that localize a signal in both space and frequency (like the Gabor filter in Table 3.2) and are defined over a hierarchy of scales. Wavelets provide a smooth way to decompose a signal into frequency components without blocking and are closely related to pyramids.

Wavelets were originally developed in the applied math and signal processing communities and were introduced to the computer vision community by Mallat (1989). Strang (1989); Simoncelli and Adelson (1990b); Rioul and Vetterli (1991); Chui (1992); Meyer (1993) all provide nice introductions to the subject along with historical reviews, while Chui (1992) provides a more comprehensive review and survey of applications. Sweldens (1997) describes the more recent *lifting* approach to wavelets that we discuss shortly.

Wavelets are widely used in the computer graphics community to perform multi-resolution geometric processing (Stollnitz, DeRose, and Salesin 1996) and have also been used in computer vision for similar applications (Szeliski 1990b; Pentland 1994; Gortler and Cohen 1995; Yaou and Chang 1994; Lai and Vemuri 1997; Szeliski 2006b), as well as for multi-scale oriented filtering (Simoncelli, Freeman, Adelson *et al.* 1992) and denoising (Portilla, Strela, Wainwright *et al.* 2003).



(a)



(b)

Figure 3.37 Two-dimensional wavelet decomposition: (a) high-level diagram showing the low-pass and high-pass transforms as single boxes; (b) separable implementation, which involves first performing the wavelet transform horizontally and then vertically. The I and F boxes are the interpolation and filtering boxes required to re-synthesize the image from its wavelet components.

Since both image pyramids and wavelets decompose an image into multi-resolution descriptions that are localized in both space and frequency, how do they differ? The usual answer is that traditional pyramids are *overcomplete*, i.e., they use more pixels than the original image to represent the decomposition, whereas wavelets provide a *tight frame*, i.e., they keep the size of the decomposition the same as the image (Figure 3.36b). However, some wavelet families *are*, in fact, overcomplete in order to provide better shiftability or steering in orientation (Simoncelli, Freeman, Adelson *et al.* 1992). A better distinction, therefore, might be that wavelets are more orientation selective than regular band-pass pyramids.

How are two-dimensional wavelets constructed? Figure 3.37a shows a high-level diagram of one stage of the (recursive) coarse-to-fine construction (analysis) pipeline alongside the complementary re-construction (synthesis) stage. In this diagram, the high-pass filter followed by decimation keeps $3/4$ of the original pixels, while $1/4$ of the low-frequency coefficients are passed on to the next stage for further analysis. In practice, the filtering is usually broken down into two separable sub-stages, as shown in Figure 3.37b. The resulting three wavelet images are sometimes called the high–high (HH), high–low (HL), and low–high (LH) images. The high–low and low–high images accentuate the horizontal and vertical edges and gradients, while the high–high image contains the less frequently occurring mixed derivatives.

How are the high-pass H and low-pass L filters shown in Figure 3.37b chosen and how

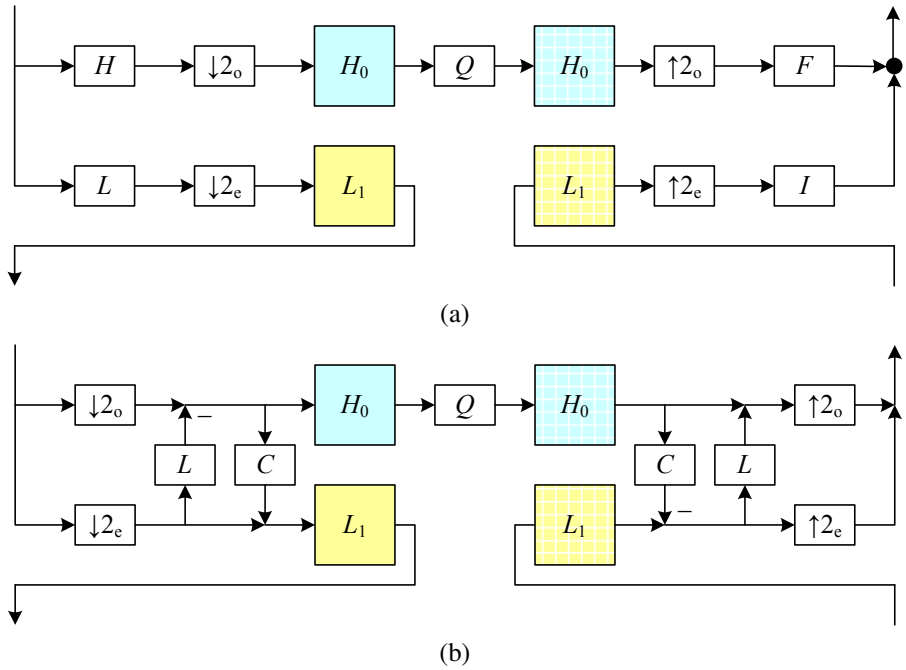
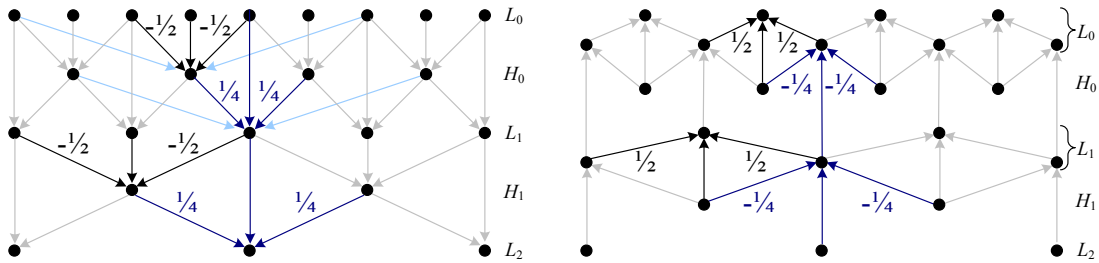


Figure 3.38 One-dimensional wavelet transform: (a) usual high-pass + low-pass filters followed by odd ($\downarrow 2_o$) and even ($\downarrow 2_e$) downsampling; (b) lifted version, which first selects the odd and even subsequences and then applies a low-pass prediction stage L and a high-pass correction stage C in an easily reversible manner.

can the corresponding reconstruction filters I and F be computed? Can filters be designed that all have finite impulse responses? This topic has been the main subject of study in the wavelet community for over two decades. The answer depends largely on the intended application, e.g., whether the wavelets are being used for compression, image analysis (feature finding), or denoising. Simoncelli and Adelson (1990b) show (in Table 4.1) some good odd-length quadrature mirror filter (QMF) coefficients that seem to work well in practice.

Since the design of wavelet filters is such a tricky art, is there perhaps a better way? Indeed, a simpler procedure is to split the signal into its even and odd components and then perform trivially reversible filtering operations on each sequence to produce what are called *lifted wavelets* (Figures 3.38 and 3.39). Sweldens (1996) gives a wonderfully understandable introduction to the *lifting scheme* for *second-generation wavelets*, followed by a comprehensive review (Sweldens 1997).

As Figure 3.38 demonstrates, rather than first filtering the whole input sequence (image) with high-pass and low-pass filters and then keeping the odd and even sub-sequences, the lifting scheme first splits the sequence into its even and odd sub-components. Filtering the even sequence with a low-pass filter L and subtracting the result from the even sequence is trivially reversible: simply perform the same filtering and then add the result back in. Furthermore, this operation can be performed in place, resulting in significant space savings. The same applies to filtering the even sequence with the correction filter C, which is used to ensure that the even sequence is low-pass. A series of such *lifting* steps can be used to create more complex filter responses with low computational cost and guaranteed reversibility.



(a) (b)

Figure 3.39 Lifted transform shown as a signal processing diagram: (a) The analysis stage first predicts the odd value from its even neighbors, stores the difference wavelet, and then compensates the coarser even value by adding in a fraction of the wavelet. (b) The synthesis stage simply reverses the flow of computation and the signs of some of the filters and operations. The light blue lines show what happens if we use four taps for the prediction and correction instead of just two.

This process can perhaps be more easily understood by considering the signal processing diagram in Figure 3.39. During analysis, the average of the even values is subtracted from the odd value to obtain a high-pass wavelet coefficient. However, the even samples still contain an aliased sample of the low-frequency signal. To compensate for this, a small amount of the high-pass wavelet is added back to the even sequence so that it is properly low-pass filtered. (It is easy to show that the effective low-pass filter is $[-1/8, 1/4, 3/4, 1/4, -1/8]$, which is indeed a low-pass filter.) During synthesis, the same operations are reversed with a judicious change in sign.

Of course, we need not restrict ourselves to two-tap filters. Figure 3.39 shows as light blue arrows additional filter coefficients that could optionally be added to the lifting scheme without affecting its reversibility. In fact, the low-pass and high-pass filtering operations can be interchanged, e.g., we could use a five-tap cubic low-pass filter on the odd sequence (plus center value) first, followed by a four-tap cubic low-pass predictor to estimate the wavelet, although I have not seen this scheme written down.

Lifted wavelets are called *second-generation wavelets* because they can easily adapt to non-regular sampling topologies, e.g., those that arise in computer graphics applications such as multi-resolution surface manipulation (Schröder and Sweldens 1995). It also turns out that lifted *weighted wavelets*, i.e., wavelets whose coefficients adapt to the underlying problem being solved (Fattal 2009), can be extremely effective for low-level image manipulation tasks and also for preconditioning the kinds of sparse linear systems that arise in the optimization-based approaches to vision algorithms that we discuss in Section 3.7 (Szeliski 2006b).

An alternative to the widely used "separable" approach to wavelet construction, which decomposes each level into horizontal, vertical, and "cross" sub-bands, is to use a representation that is more rotationally symmetric and orientationally selective and also avoids the aliasing inherent in sampling signals below their Nyquist frequency.[17] Simoncelli, Freeman, Adelson *et al.* (1992) introduce such a representation, which they call a *pyramidal radial frequency implementation* of *shiftable multi-scale transforms* or, more succinctly, *steerable pyramids*.

[17] Such aliasing can often be seen as the signal content moving between bands as the original signal is slowly shifted.

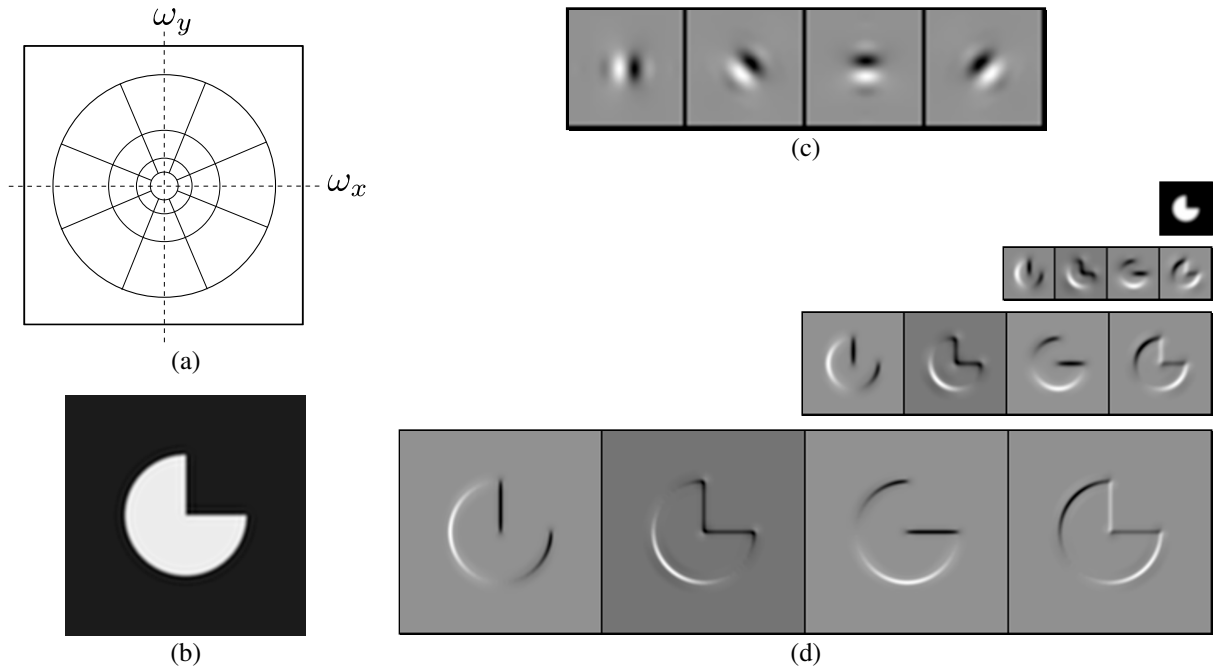


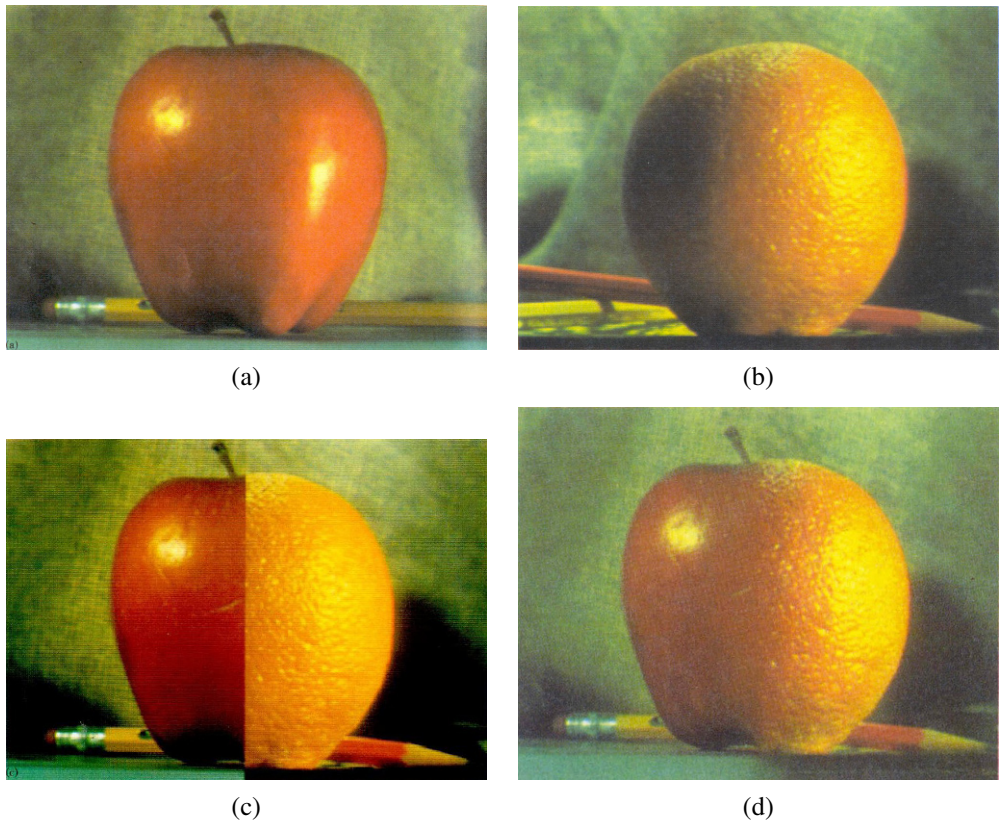
Figure 3.40 Steerable shiftable multiscale transforms (Simoncelli, Freeman, Adelson *et al.* 1992) © 1992 IEEE: (a) radial multi-scale frequency domain decomposition; (b) original image; (c) a set of four steerable filters; (d) the radial multi-scale wavelet decomposition.

Their representation is not only overcomplete (which eliminates the aliasing problem) but is also orientationally selective and has identical analysis and synthesis basis functions, i.e., it is *self-inverting*, just like "regular" wavelets. As a result, this makes steerable pyramids a much more useful basis for the structural analysis and matching tasks commonly used in computer vision.

Figure 3.40a shows how such a decomposition looks in frequency space. Instead of recursively dividing the frequency domain into 2×2 squares, which results in checkerboard high frequencies, radial arcs are used instead. Figure 3.40b illustrates the resulting pyramid sub-bands. Even through the representation is *overcomplete*, i.e., there are more wavelet coefficients than input pixels, the additional frequency and orientation selectivity makes this representation preferable for tasks such as texture analysis and synthesis (Portilla and Simoncelli 2000) and image denoising (Portilla, Strela, Wainwright *et al.* 2003; Lyu and Simoncelli 2009).

3.5.5 *Application*: Image blending

One of the most engaging and fun applications of the Laplacian pyramid presented in Section 3.5.3 is the creation of blended composite images, as shown in Figure 3.41 (Burt and Adelson 1983b). While splicing the apple and orange images together along the midline produces a noticeable cut, *splining* them together (as Burt and Adelson (1983b) called their procedure) creates a beautiful illusion of a truly hybrid fruit. The key to their approach is that the low-frequency color variations between the red apple and the orange are smoothly



(a) (b) (c) (d)

Figure 3.41 Laplacian pyramid blending (Burt and Adelson 1983b) © 1983 ACM: (a) original image of apple, (b) original image of orange, (c) regular splice, (d) pyramid blend.

blended, while the higher-frequency textures on each fruit are blended more quickly to avoid "ghosting" effects when two textures are overlaid.

To create the blended image, each source image is first decomposed into its own Laplacian pyramid (Figure 3.42, left and middle columns). Each band is then multiplied by a smooth weighting function whose extent is proportional to the pyramid level. The simplest and most general way to create these weights is to take a binary mask image (Figure 3.43c) and to construct a *Gaussian* pyramid from this mask. Each Laplacian pyramid image is then multiplied by its corresponding Gaussian mask and the sum of these two weighted pyramids is then used to construct the final image (Figure 3.42, right column).

Figure 3.43 shows that this process can be applied to arbitrary mask images with surprising results. It is also straightforward to extend the pyramid blend to an arbitrary number of images whose pixel provenance is indicated by an integer-valued label image (see Exercise 3.20). This is particularly useful in image stitching and compositing applications, where the exposures may vary between different images, as described in Section 9.3.4.

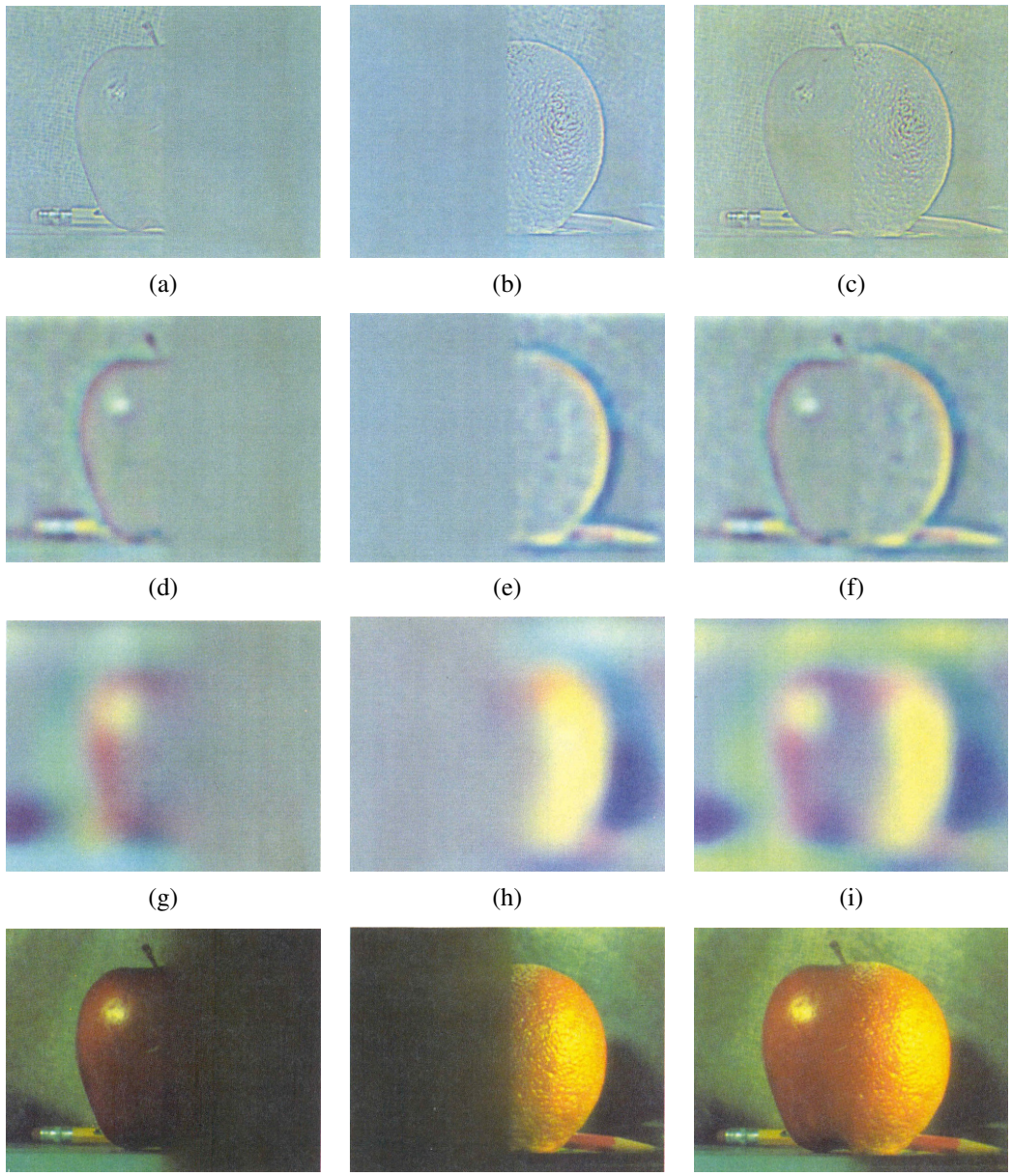
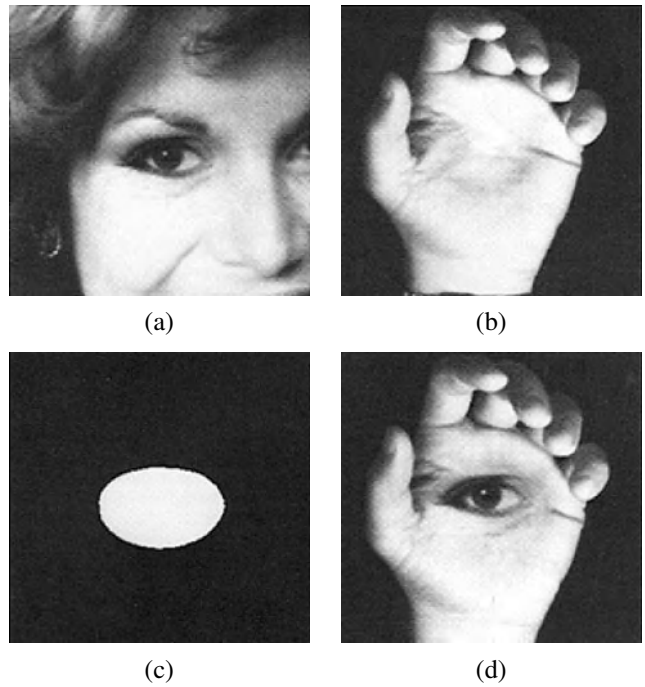


Figure 3.42 Laplacian pyramid blending details (Burt and Adelson 1983b) © 1983 ACM. The first three rows show the high, medium, and low frequency parts of the Laplacian pyramid (taken from levels 0, 2, and 4). The left and middle columns show the original apple and orange images weighted by the smooth interpolation functions, while the right column shows the averaged contributions.



(a) (b) (c) (d)

Figure 3.43 Laplacian pyramid blend of two images of arbitrary shape (Burt and Adelson 1983b) © 1983 ACM: (a) first input image; (b) second input image; (c) region mask; (d) blended image.

3.6 Geometric transformations

In the previous sections, we saw how interpolation and decimation could be used to change the *resolution* of an image. In this section, we look at how to perform more general transformations, such as image rotations or general warps. In contrast to the point processes we saw in Section 3.1, where the function applied to an image transforms the *range* of the image,

$$g(\boldsymbol{x}) = h(f(\boldsymbol{x})), \tag{3.87}$$

here we look at functions that transform the *domain*,

$$g(\boldsymbol{x}) = f(\boldsymbol{h}(\boldsymbol{x})) \tag{3.88}$$

(see Figure 3.44).

We begin by studying the global *parametric* 2D transformation first introduced in Section 2.1.2. (Such a transformation is called parametric because it is controlled by a small number of parameters.) We then turn our attention to more local general deformations such as those defined on meshes (Section 3.6.2). Finally, we show how image warps can be combined with cross-dissolves to create interesting *morphs* (in-between animations) in Section 3.6.3. For readers interested in more details on these topics, there is an excellent survey by Heckbert (1986) as well as very accessible textbooks by Wolberg (1990), Gomes, Darsa, Costa *et al.* (1999) and Akenine-Möller and Haines (2002). Note that Heckbert's survey is on *texture mapping*, which is how the computer graphics community refers to the topic of warping images onto surfaces.

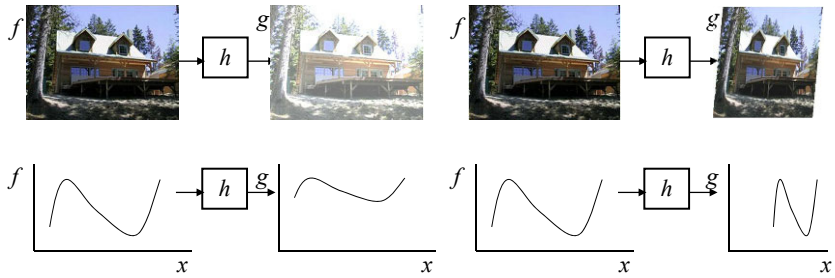


Figure 3.44 Image warping involves modifying the *domain* of an image function rather than its *range*.

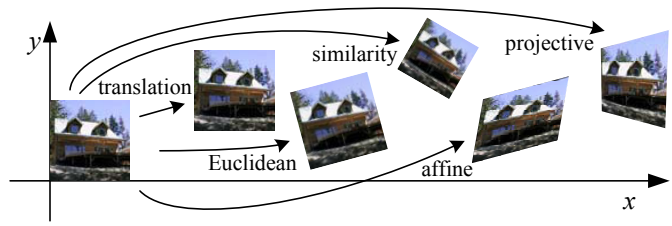


Figure 3.45 Basic set of 2D geometric image transformations.

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

Table 3.5 Hierarchy of 2D coordinate transformations. Each transformation also preserves the properties listed in the rows below it, i.e., similarity preserves not only angles but also parallelism and straight lines. The 2×3 matrices are extended with a third $[\mathbf{0}^T\ 1]$ row to form a full 3×3 matrix for homogeneous coordinate transformations.

procedure *forwardWarp*($f, \boldsymbol{h}$, **out** g):

For every pixel $\boldsymbol{x}$ in $f(\boldsymbol{x})$

1. Compute the destination location $\boldsymbol{x}' = \boldsymbol{h}(\boldsymbol{x})$.
2. Copy the pixel $f(\boldsymbol{x})$ to $g(\boldsymbol{x}')$.

Algorithm 3.1 Forward warping algorithm for transforming an image $f(\boldsymbol{x})$ into an image $g(\boldsymbol{x}')$ through the parametric transform $\boldsymbol{x}' = \boldsymbol{h}(\boldsymbol{x})$.



Figure 3.46 Forward warping algorithm: (a) a pixel $f(\boldsymbol{x})$ is copied to its corresponding location $\boldsymbol{x}' = \boldsymbol{h}(\boldsymbol{x})$ in image $g(\boldsymbol{x}')$; (b) detail of the source and destination pixel locations.

3.6.1 Parametric transformations

Parametric transformations apply a global deformation to an image, where the behavior of the transformation is controlled by a small number of parameters. Figure 3.45 shows a few examples of such transformations, which are based on the 2D geometric transformations shown in Figure 2.4. The formulas for these transformations were originally given in Table 2.1 and are reproduced here in Table 3.5 for ease of reference.

In general, given a transformation specified by a formula $\boldsymbol{x}' = \boldsymbol{h}(\boldsymbol{x})$ and a source image $f(\boldsymbol{x})$, how do we compute the values of the pixels in the new image $g(\boldsymbol{x})$, as given in (3.88)? Think about this for a minute before proceeding and see if you can figure it out.

If you are like most people, you will come up with an algorithm that looks something like Algorithm 3.1. This process is called *forward warping* or *forward mapping* and is shown in Figure 3.46a. Can you think of any problems with this approach?

In fact, this approach suffers from several limitations. The process of copying a pixel $f(\boldsymbol{x})$ to a location $\boldsymbol{x}'$ in g is not well defined when $\boldsymbol{x}'$ has a non-integer value. What do we do in such a case? What would you do?

You can round the value of $\boldsymbol{x}'$ to the nearest integer coordinate and copy the pixel there, but the resulting image has severe aliasing and pixels that jump around a lot when animating the transformation. You can also "distribute" the value among its four nearest neighbors in a weighted (bilinear) fashion, keeping track of the per-pixel weights and normalizing at the end. This technique is called *splatting* and is sometimes used for volume rendering in the graphics community (Levoy and Whitted 1985; Levoy 1988; Westover 1989; Rusinkiewicz and Levoy 2000). Unfortunately, it suffers from both moderate amounts of aliasing and a fair amount of blur (loss of high-resolution detail).

procedure *inverseWarp*($f, \boldsymbol{h}$, **out** g):

For every pixel $\boldsymbol{x}'$ in $g(\boldsymbol{x}')$

1. Compute the source location $\boldsymbol{x} = \hat{\boldsymbol{h}}(\boldsymbol{x}')$
2. Resample $f(\boldsymbol{x})$ at location $\boldsymbol{x}$ and copy to $g(\boldsymbol{x}')$

Algorithm 3.2 Inverse warping algorithm for creating an image $g(\boldsymbol{x}')$ from an image $f(\boldsymbol{x})$ using the parametric transform $\boldsymbol{x}' = \boldsymbol{h}(\boldsymbol{x})$.



Figure 3.47 Inverse warping algorithm: (a) a pixel $g(\boldsymbol{x}')$ is sampled from its corresponding location $\boldsymbol{x} = \hat{\boldsymbol{h}}(\boldsymbol{x}')$ in image $f(\boldsymbol{x})$; (b) detail of the source and destination pixel locations.

The second major problem with forward warping is the appearance of cracks and holes, especially when magnifying an image. Filling such holes with their nearby neighbors can lead to further aliasing and blurring.

What can we do instead? A preferable solution is to use *inverse warping* (Algorithm 3.2), where each pixel in the destination image $g(\boldsymbol{x}')$ is sampled from the original image $f(\boldsymbol{x})$ (Figure 3.47).

How does this differ from the forward warping algorithm? For one thing, since $\hat{\boldsymbol{h}}(\boldsymbol{x}')$ is (presumably) defined for all pixels in $g(\boldsymbol{x}')$, we no longer have holes. More importantly, resampling an image at non-integer locations is a well-studied problem (general image interpolation, see Section 3.5.2) and high-quality filters that control aliasing can be used.

Where does the function $\hat{\boldsymbol{h}}(\boldsymbol{x}')$ come from? Quite often, it can simply be computed as the inverse of $\boldsymbol{h}(\boldsymbol{x})$. In fact, all of the parametric transforms listed in Table 3.5 have closed form solutions for the inverse transform: simply take the inverse of the 3×3 matrix specifying the transform.

In other cases, it is preferable to formulate the problem of image warping as that of resampling a source image $f(\boldsymbol{x})$ given a mapping $\boldsymbol{x} = \hat{\boldsymbol{h}}(\boldsymbol{x}')$ from destination pixels $\boldsymbol{x}'$ to source pixels $\boldsymbol{x}$. For example, in optical flow (Section 8.4), we estimate the flow field as the location of the *source* pixel which produced the current pixel whose flow is being estimated, as opposed to computing the *destination* pixel to which it is going. Similarly, when correcting for radial distortion (Section 2.1.6), we calibrate the lens by computing for each pixel in the final (undistorted) image the corresponding pixel location in the original (distorted) image.

What kinds of interpolation filter are suitable for the resampling process? Any of the filters we studied in Section 3.5.2 can be used, including nearest neighbor, bilinear, bicubic, and

windowed sinc functions. While bilinear is often used for speed (e.g., inside the inner loop of a patch-tracking algorithm, see Section 8.1.3), bicubic, and windowed sinc are preferable where visual quality is important.

To compute the value of $f(\boldsymbol{x})$ at a non-integer location $\boldsymbol{x}$, we simply apply our usual FIR resampling filter,

$$g(x,y) = \sum_{k,l} f(k,l)h(x-k, y-l), \tag{3.89}$$

where (x, y) are the sub-pixel coordinate values and $h(x, y)$ is some interpolating or smoothing kernel. Recall from Section 3.5.2 that when decimation is being performed, the smoothing kernel is stretched and re-scaled according to the downsampling rate r.

Unfortunately, for a general (non-zoom) image transformation, the resampling rate r is not well defined. Consider a transformation that stretches the x dimensions while squashing the y dimensions. The resampling kernel should be performing regular interpolation along the x dimension and smoothing (to anti-alias the blurred image) in the y direction. This gets even more complicated for the case of general affine or perspective transforms.

What can we do? Fortunately, Fourier analysis can help. The two-dimensional generalization of the one-dimensional *domain scaling* law given in Table 3.1 is

$$g(\boldsymbol{A}\boldsymbol{x}) \Leftrightarrow |\boldsymbol{A}|^{-1} G(\boldsymbol{A}^{-T}\boldsymbol{f}). \tag{3.90}$$

For all of the transforms in Table 3.5 except perspective, the matrix $\boldsymbol{A}$ is already defined. For perspective transformations, the matrix $\boldsymbol{A}$ is the linearized *derivative* of the perspective transformation (Figure 3.48a), i.e., the local affine approximation to the stretching induced by the projection (Heckbert 1986; Wolberg 1990; Gomes, Darsa, Costa *et al.* 1999; Akenine-Möller and Haines 2002).

To prevent aliasing, we need to pre-filter the image $f(\boldsymbol{x})$ with a filter whose frequency response is the projection of the final desired spectrum through the $\boldsymbol{A}^{-T}$ transform (Szeliski, Winder, and Uyttendaele 2010). In general (for non-zoom transforms), this filter is non-separable and hence is very slow to compute. Therefore, a number of approximations to this filter are used in practice, include MIP-mapping, elliptically weighted Gaussian averaging, and anisotropic filtering (Akenine-Möller and Haines 2002).

MIP-mapping

MIP-mapping was first proposed by Williams (1983) as a means to rapidly pre-filter images being used for *texture mapping* in computer graphics. A MIP-map[18] is a standard image pyramid (Figure 3.32), where each level is pre-filtered with a high-quality filter rather than a poorer quality approximation, such as Burt and Adelson's (1983b) five-tap binomial. To resample an image from a MIP-map, a scalar estimate of the resampling rate r is first computed. For example, r can be the maximum of the absolute values in $\boldsymbol{A}$ (which suppresses aliasing) or it can be the minimum (which reduces blurring). Akenine-Möller and Haines (2002) discuss these issues in more detail.

Once a resampling rate has been specified, a *fractional* pyramid level is computed using the base 2 logarithm,

$$l = \log_2 r. \tag{3.91}$$

[18] The term 'MIP' stands for *multi in parvo*, meaning 'many in one'.

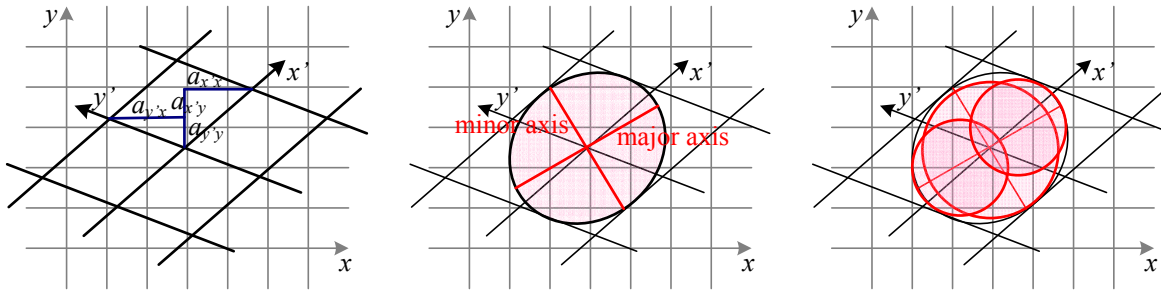


Figure 3.48 Anisotropic texture filtering: (a) Jacobian of transform $\boldsymbol{A}$ and the induced horizontal and vertical resampling rates $\{a_{x'x}, a_{x'y}, a_{y'x}, a_{y'y}\}$; (b) elliptical footprint of an EWA smoothing kernel; (c) anisotropic filtering using multiple samples along the major axis. Image pixels lie at line intersections.

One simple solution is to resample the texture from the next higher or lower pyramid level, depending on whether it is preferable to reduce aliasing or blur. A better solution is to resample *both* images and blend them linearly using the fractional component of l. Since most MIP-map implementations use bilinear resampling within each level, this approach is usually called *trilinear MIP-mapping*. Computer graphics rendering APIs, such as OpenGL and Direct3D, have parameters that can be used to select which variant of MIP-mapping (and of the sampling rate r computation) should be used, depending on the desired tradeoff between speed and quality. Exercise 3.22 has you examine some of these tradeoffs in more detail.

Elliptical Weighted Average

The Elliptical Weighted Average (EWA) filter invented by Greene and Heckbert (1986) is based on the observation that the affine mapping $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}'$ defines a skewed two-dimensional coordinate system in the vicinity of each source pixel $\boldsymbol{x}$ (Figure 3.48a). For every destination pixel $\boldsymbol{x}'$, the ellipsoidal projection of a small pixel grid in $\boldsymbol{x}'$ onto $\boldsymbol{x}$ is computed (Figure 3.48b). This is then used to filter the source image $g(\boldsymbol{x})$ with a Gaussian whose inverse covariance matrix is this ellipsoid.

Despite its reputation as a high-quality filter (Akenine-Möller and Haines 2002), we have found in our work (Szeliski, Winder, and Uyttendaele 2010) that because a Gaussian kernel is used, the technique suffers simultaneously from both blurring and aliasing, compared to higher-quality filters. The EWA is also quite slow, although faster variants based on MIP-mapping have been proposed (Szeliski, Winder, and Uyttendaele (2010) provide some additional references).

Anisotropic filtering

An alternative approach to filtering oriented textures, which is sometimes implemented in graphics hardware (GPUs), is to use anisotropic filtering (Barkans 1997; Akenine-Möller and Haines 2002). In this approach, several samples at different resolutions (fractional levels in the MIP-map) are combined along the major axis of the EWA Gaussian (Figure 3.48c).

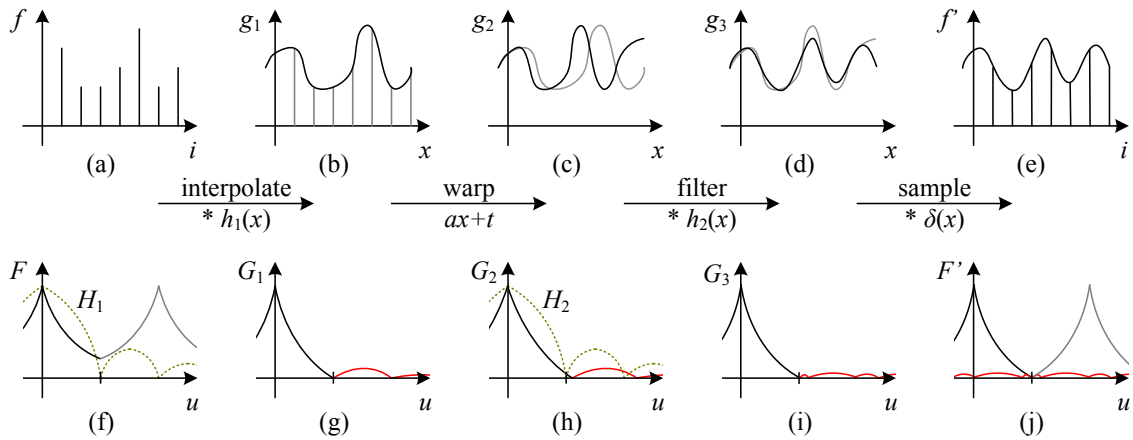


Figure 3.49 One-dimensional signal resampling (Szeliski, Winder, and Uyttendaele 2010): (a) original sampled signal $f(i)$; (b) interpolated signal $g_1(x)$; (c) warped signal $g_2(x)$; (d) filtered signal $g_3(x)$; (e) sampled signal $f'(i)$. The corresponding spectra are shown below the signals, with the aliased portions shown in red.

Multi-pass transforms

The optimal approach to warping images without excessive blurring or aliasing is to adaptively pre-filter the source image at each pixel using an ideal low-pass filter, i.e., an oriented skewed sinc or low-order (e.g., cubic) approximation (Figure 3.48a). Figure 3.49 shows how this works in one dimension. The signal is first (theoretically) interpolated to a continuous waveform, (ideally) low-pass filtered to below the new Nyquist rate, and then re-sampled to the final desired resolution. In practice, the interpolation and decimation steps are concatenated into a single *polyphase* digital filtering operation (Szeliski, Winder, and Uyttendaele 2010).

For parametric transforms, the oriented two-dimensional filtering and resampling operations can be approximated using a series of one-dimensional resampling and shearing transforms (Catmull and Smith 1980; Heckbert 1989; Wolberg 1990; Gomes, Darsa, Costa *et al.* 1999; Szeliski, Winder, and Uyttendaele 2010). The advantage of using a series of one-dimensional transforms is that they are much more efficient (in terms of basic arithmetic operations) than large, non-separable, two-dimensional filter kernels.

In order to prevent aliasing, however, it may be necessary to upsample in the opposite direction before applying a shearing transformation (Szeliski, Winder, and Uyttendaele 2010). Figure 3.50 shows this process for a rotation, where a vertical upsampling stage is added before the horizontal shearing (and upsampling) stage. The upper image shows the appearance of the letter being rotated, while the lower image shows its corresponding Fourier transform.

3.6.2 Mesh-based warping

While parametric transforms specified by a small number of global parameters have many uses, *local* deformations with more degrees of freedom are often required.

Consider, for example, changing the appearance of a face from a frown to a smile (Figure 3.51a). What is needed in this case is to curve the corners of the mouth upwards while

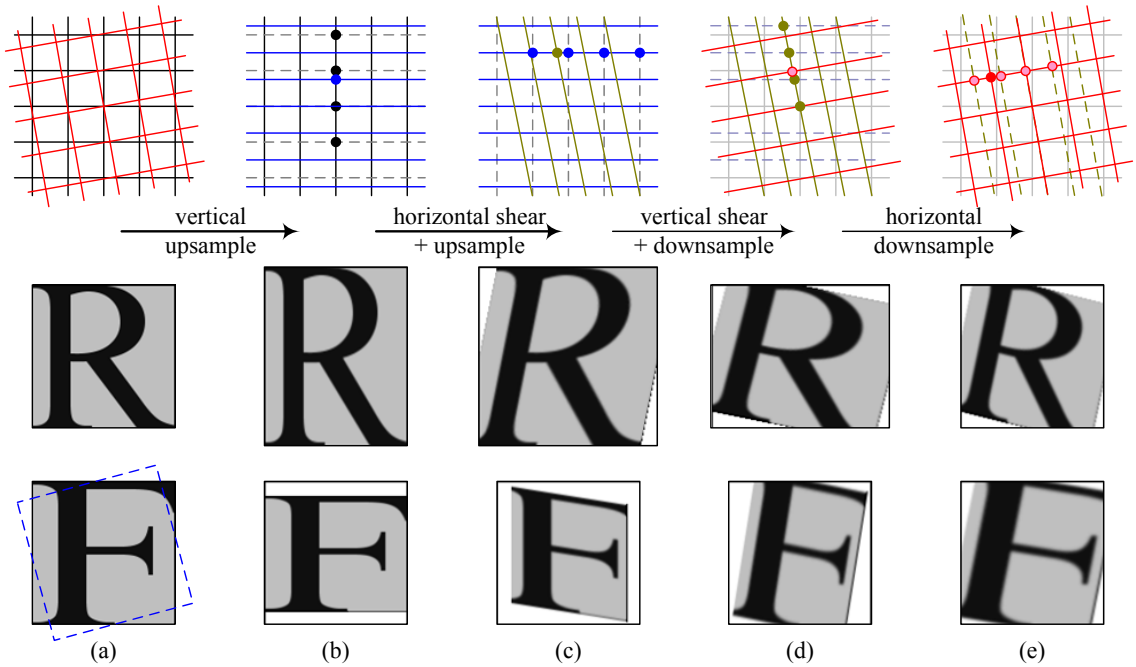


Figure 3.50 Four-pass rotation (Szeliski, Winder, and Uyttendaele 2010): (a) original pixel grid, image, and its Fourier transform; (b) vertical upsampling; (c) horizontal shear and upsampling; (d) vertical shear and downsampling; (e) horizontal downsampling. The general affine case looks similar except that the first two stages perform general resampling.

leaving the rest of the face intact.[19] To perform such a transformation, different amounts of motion are required in different parts of the image. Figure 3.51 shows some of the commonly used approaches.

The first approach, shown in Figure 3.51a–b, is to specify a *sparse* set of corresponding points. The displacement of these points can then be interpolated to a dense *displacement field* (Chapter 8) using a variety of techniques (Nielson 1993). One possibility is to *triangulate* the set of points in one image (de Berg, Cheong, van Kreveld *et al.* 2006; Litwinowicz and Williams 1994; Buck, Finkelstein, Jacobs *et al.* 2000) and to use an *affine* motion model (Table 3.5), specified by the three triangle vertices, inside each triangle. If the destination image is triangulated according to the new vertex locations, an inverse warping algorithm (Figure 3.47) can be used. If the source image is triangulated and used as a *texture map*, computer graphics rendering algorithms can be used to draw the new image (but care must be taken along triangle edges to avoid potential aliasing).

Alternative methods for interpolating a sparse set of displacements include moving nearby quadrilateral mesh vertices, as shown in Figure 3.51a, using *variational* (energy minimizing) interpolants such as regularization (Litwinowicz and Williams 1994), see Section 3.7.1, or using locally weighted (*radial basis function*) combinations of displacements (Nielson 1993). (See (Section 12.3.1) for additional *scattered data interpolation* techniques.) If quadrilateral

[19] Rowland and Perrett (1995); Pighin, Hecker, Lischinski *et al.* (1998); Blanz and Vetter (1999); Leyvand, Cohen-Or, Dror *et al.* (2008) show more sophisticated examples of changing facial expression and appearance.

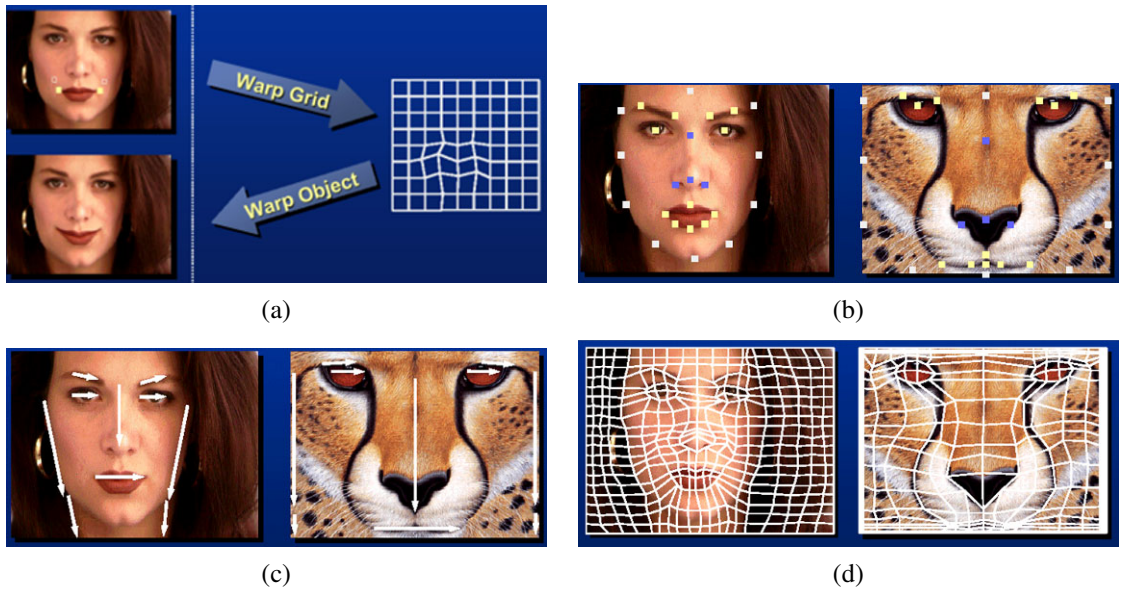


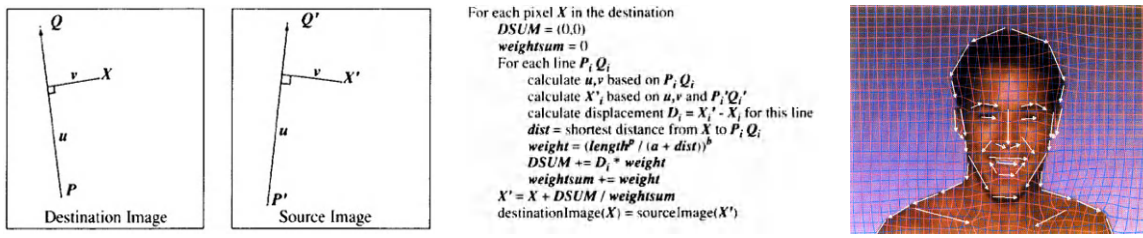
Figure 3.51 Image warping alternatives (Gomes, Darsa, Costa *et al.* 1999) © 1999 Morgan Kaufmann: (a) sparse control points ⟶ deformation grid; (b) denser set of control point correspondences; (c) oriented line correspondences; (d) uniform quadrilateral grid.

meshes are used, it may be desirable to interpolate displacements down to individual pixel values using a smooth interpolant such as a quadratic B-spline (Farin 1996; Lee, Wolberg, Chwa *et al.* 1996).[20]

In some cases, e.g., if a dense depth map has been estimated for an image (Shade, Gortler, He *et al.* 1998), we only know the forward displacement for each pixel. As mentioned before, drawing source pixels at their destination location, i.e., forward warping (Figure 3.46), suffers from several potential problems, including aliasing and the appearance of small cracks. An alternative technique in this case is to forward warp the *displacement field* (or depth map) to its new location, fill small holes in the resulting map, and then use inverse warping to perform the resampling (Shade, Gortler, He *et al.* 1998). The reason that this generally works better than forward warping is that displacement fields tend to be much smoother than images, so the aliasing introduced during the forward warping of the displacement field is much less noticeable.

A second approach to specifying displacements for local deformations is to use corresponding *oriented line segments* (Beier and Neely 1992), as shown in Figures 3.51c and 3.52. Pixels along each line segment are transferred from source to destination exactly as specified, and other pixels are warped using a smooth interpolation of these displacements. Each line segment correspondence specifies a translation, rotation, and scaling, i.e., a *similarity transform* (Table 3.5), for pixels in its vicinity, as shown in Figure 3.52a. Line segments influence the overall displacement of the image using a weighting function that depends on the minimum distance to the line segment (v in Figure 3.52a if $u \in [0, 1]$, else the shorter of the two

[20] Note that the *block-based* motion models used by many video compression standards (Le Gall 1991) can be thought of as a 0th-order (piecewise-constant) displacement field.



(a) (b) (c)

Figure 3.52 Line-based image warping (Beier and Neely 1992) © 1992 ACM: (a) distance computation and position transfer; (b) rendering algorithm; (c) two intermediate warps used for morphing.

distances to P and Q).

For each pixel X, the target location X' for each line correspondence is computed along with a weight that depends on the distance and the line segment length (Figure 3.52b). The weighted average of all target locations X'_i then becomes the final destination location. Note that while Beier and Neely describe this algorithm as a forward warp, an equivalent algorithm can be written by sequencing through the destination pixels. The resulting warps are not identical because line lengths or distances to lines may be different. Exercise 3.23 has you implement the Beier–Neely (line-based) warp and compare it to a number of other local deformation methods.

Yet another way of specifying correspondences in order to create image warps is to use snakes (Section 5.1.1) combined with B-splines (Lee, Wolberg, Chwa *et al.* 1996). This technique is used in Apple's Shake software and is popular in the medical imaging community.

One final possibility for specifying displacement fields is to use a mesh specifically *adapted* to the underlying image content, as shown in Figure 3.51d. Specifying such meshes by hand can involve a fair amount of work; Gomes, Darsa, Costa *et al.* (1999) describe an interactive system for doing this. Once the two meshes have been specified, intermediate warps can be generated using linear interpolation and the displacements at mesh nodes can be interpolated using splines.

3.6.3 *Application*: Feature-based morphing

While warps can be used to change the appearance of or to animate a *single* image, even more powerful effects can be obtained by warping and blending two or more images using a process now commonly known as *morphing* (Beier and Neely 1992; Lee, Wolberg, Chwa *et al.* 1996; Gomes, Darsa, Costa *et al.* 1999).

Figure 3.53 shows the essence of image morphing. Instead of simply cross-dissolving between two images, which leads to ghosting as shown in the top row, each image is warped toward the other image before blending, as shown in the bottom row. If the correspondences have been set up well (using any of the techniques shown in Figure 3.51), corresponding features are aligned and no ghosting results.

The above process is repeated for each intermediate frame being generated during a morph, using different blends (and amounts of deformation) at each interval. Let $t \in [0, 1]$ be

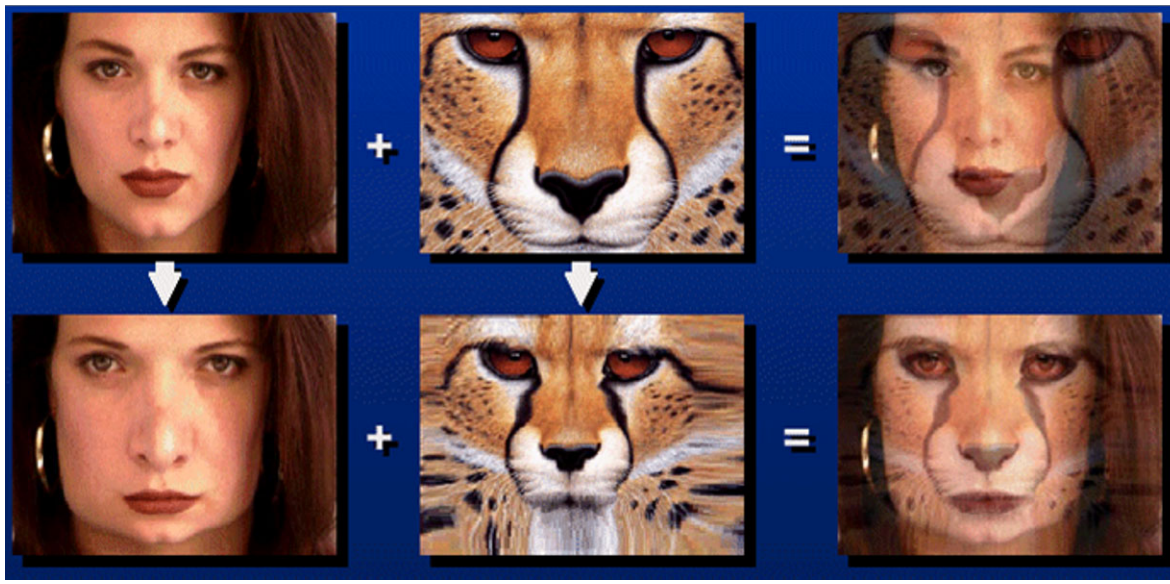


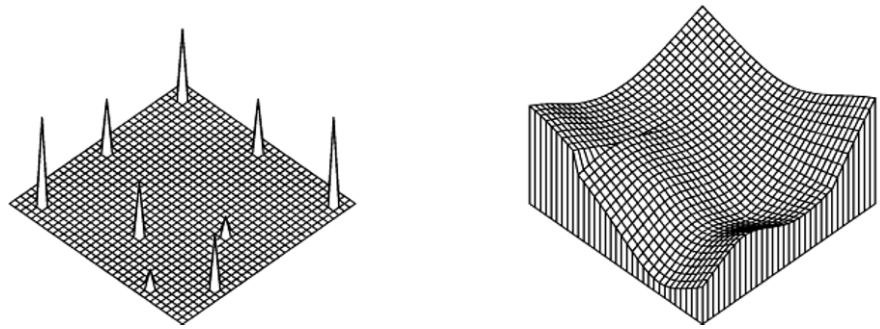
Figure 3.53 Image morphing (Gomes, Darsa, Costa *et al.* 1999) © 1999 Morgan Kaufmann. Top row: if the two images are just blended, visible ghosting results. Bottom row: both images are first warped to the same intermediate location (e.g., halfway towards the other image) and the resulting warped images are then blended resulting in a seamless morph.

the time parameter that describes the sequence of interpolated frames. The weighting functions for the two warped images in the blend go as $(1-t)$ and t. Conversely, the amount of motion that image 0 undergoes at time t is t of the total amount of motion that is specified by the correspondences. However, some care must be taken in defining what it means to partially warp an image towards a destination, especially if the desired motion is far from linear (Sederberg, Gao, Wang *et al.* 1993). Exercise 3.25 has you implement a morphing algorithm and test it out under such challenging conditions.

3.7 Global optimization

So far in this chapter, we have covered a large number of image processing operators that take as input one or more images and produce some filtered or transformed version of these images. In many applications, it is more useful to first *formulate* the goals of the desired transformation using some optimization criterion and then find or infer the solution that best meets this criterion.

In this final section, we present two different (but closely related) variants on this idea. The first, which is often called *regularization* or *variational methods* (Section 3.7.1), constructs a continuous global energy function that describes the desired characteristics of the solution and then finds a minimum energy solution using sparse linear systems or related iterative techniques. The second formulates the problem using Bayesian statistics, modeling both the noisy measurement process that produced the input images as well as *prior assumptions* about the solution space, which are often encoded using a *Markov random field*



(a) (b)

Figure 3.54 A simple surface interpolation problem: (a) nine data points of various height scattered on a grid; (b) second-order, controlled-continuity, thin-plate spline interpolator, with a tear along its left edge and a crease along its right (Szeliski 1989) © 1989 Springer.

(Section 3.7.2).

Examples of such problems include surface interpolation from scattered data (Figure 3.54), image denoising and the restoration of missing regions (Figure 3.57), and the segmentation of images into foreground and background regions (Figure 3.61).

3.7.1 Regularization

The theory of regularization was first developed by statisticians trying to fit models to data that severely underconstrained the solution space (Tikhonov and Arsenin 1977; Engl, Hanke, and Neubauer 1996). Consider, for example, finding a smooth surface that passes through (or near) a set of measured data points (Figure 3.54). Such a problem is described as *ill-posed* because many possible surfaces can fit this data. Since small changes in the input can sometimes lead to large changes in the fit (e.g., if we use polynomial interpolation), such problems are also often *ill-conditioned*. Since we are trying to recover the unknown function $f(x, y)$ from which the data point $d(x_i, y_i)$ were sampled, such problems are also often called *inverse problems*. Many computer vision tasks can be viewed as inverse problems, since we are trying to recover a full description of the 3D world from a limited set of images.

In order to quantify what it means to find a *smooth* solution, we can define a norm on the solution space. For one-dimensional functions $f(x)$, we can integrate the squared first derivative of the function,

$$\mathcal{E}_1 = \int f_x^2(x)\, dx \tag{3.92}$$

or perhaps integrate the squared second derivative,

$$\mathcal{E}_2 = \int f_{xx}^2(x)\, dx. \tag{3.93}$$

(Here, we use subscripts to denote differentiation.) Such energy measures are examples of *functionals*, which are operators that map functions to scalar values. They are also often called *variational methods*, because they measure the variation (non-smoothness) in a function.

In two dimensions (e.g., for images, flow fields, or surfaces), the corresponding smoothness functionals are

$$\mathcal{E}_1 = \int f_x^2(x,y) + f_y^2(x,y)\, dx\, dy = \int \|\nabla f(x,y)\|^2\, dx\, dy \tag{3.94}$$

and

$$\mathcal{E}_2 = \int f_{xx}^2(x,y) + 2f_{xy}^2(x,y) + f_{yy}^2(x,y)\, dx\, dy, \tag{3.95}$$

where the mixed $2f_{xy}^2$ term is needed to make the measure rotationally invariant (Grimson 1983).

The first derivative norm is often called the *membrane*, since interpolating a set of data points using this measure results in a tent-like structure. (In fact, this formula is a small-deflection approximation to the surface area, which is what soap bubbles minimize.) The second-order norm is called the *thin-plate spline*, since it approximates the behavior of thin plates (e.g., flexible steel) under small deformations. A blend of the two is called the *thin-plate spline under tension*; versions of these formulas where each derivative term is multiplied by a local weighting function are called *controlled-continuity splines* (Terzopoulos 1988). Figure 3.54 shows a simple example of a controlled-continuity interpolator fit to nine scattered data points. In practice, it is more common to find first-order smoothness terms used with images and flow fields (Section 8.4) and second-order smoothness associated with surfaces (Section 12.3.1).

In addition to the smoothness term, regularization also requires a data term (or *data penalty*). For scattered data interpolation (Nielson 1993), the data term measures the distance between the function $f(x,y)$ and a set of data points $d_i = d(x_i, y_i)$,

$$\mathcal{E}_d = \sum_i [f(x_i, y_i) - d_i]^2. \tag{3.96}$$

For a problem like noise removal, a continuous version of this measure can be used,

$$\mathcal{E}_d = \int [f(x,y) - d(x,y)]^2\, dx\, dy. \tag{3.97}$$

To obtain a global energy that can be minimized, the two energy terms are usually added together,

$$\mathcal{E} = \mathcal{E}_d + \lambda \mathcal{E}_s, \tag{3.98}$$

where $\mathcal{E}_s$ is the *smoothness penalty* ($\mathcal{E}_1$, $\mathcal{E}_2$ or some weighted blend) and λ is the *regularization parameter*, which controls how smooth the solution should be.

In order to find the minimum of this continuous problem, the function $f(x,y)$ is usually first discretized on a regular grid.[21] The most principled way to perform this discretization is to use *finite element analysis*, i.e., to approximate the function with a piecewise continuous spline, and then perform the analytic integration (Bathe 2007).

Fortunately, for both the first-order and second-order smoothness functionals, the judicious selection of appropriate finite elements results in particularly simple discrete forms

[21] The alternative of using *kernel basis functions* centered on the data points (Boult and Kender 1986; Nielson 1993) is discussed in more detail in Section 12.3.1.

(Terzopoulos 1983). The corresponding *discrete* smoothness energy functions become

$$\begin{aligned} E_1 &= \sum_{i,j} s_x(i,j)[f(i+1,j) - f(i,j) - g_x(i,j)]^2 \\ &\quad + s_y(i,j)[f(i,j+1) - f(i,j) - g_y(i,j)]^2 \end{aligned} \tag{3.99}$$

and

$$\begin{aligned} E_2 &= h^{-2} \sum_{i,j} c_x(i,j)[f(i+1,j) - 2f(i,j) + f(i-1,j)]^2 \\ &\quad + 2c_m(i,j)[f(i+1,j+1) - f(i+1,j) - f(i,j+1) + f(i,j)]^2 \\ &\quad + c_y(i,j)[f(i,j+1) - 2f(i,j) + f(i,j-1)]^2, \end{aligned} \tag{3.100}$$

where h is the size of the finite element grid. The h factor is only important if the energy is being discretized at a variety of resolutions, as in coarse-to-fine or multigrid techniques.

The optional smoothness weights $s_x(i,j)$ and $s_y(i,j)$ control the location of horizontal and vertical tears (or weaknesses) in the surface. For other problems, such as colorization (Levin, Lischinski, and Weiss 2004) and interactive tone mapping (Lischinski, Farbman, Uyttendaele *et al.* 2006a), they control the smoothness in the interpolated chroma or exposure field and are often set inversely proportional to the local luminance gradient strength. For second-order problems, the crease variables $c_x(i,j)$, $c_m(i,j)$, and $c_y(i,j)$ control the locations of creases in the surface (Terzopoulos 1988; Szeliski 1990a).

The data values $g_x(i,j)$ and $g_y(i,j)$ are gradient data terms (constraints) used by algorithms, such as photometric stereo (Section 12.1.1), HDR tone mapping (Section 10.2.1) (Fattal, Lischinski, and Werman 2002), Poisson blending (Section 9.3.4) (Pérez, Gangnet, and Blake 2003), and gradient-domain blending (Section 9.3.4) (Levin, Zomet, Peleg *et al.* 2004). They are set to zero when just discretizing the conventional first-order smoothness functional (3.94).

The two-dimensional discrete data energy is written as

$$E_d = \sum_{i,j} w(i,j)[f(i,j) - d(i,j)]^2, \tag{3.101}$$

where the local weights $w(i,j)$ control how strongly the data constraint is enforced. These values are set to zero where there is no data and can be set to the inverse variance of the data measurements when there is data (as discussed by Szeliski (1989) and in Section 3.7.2).

The total energy of the discretized problem can now be written as a *quadratic form*

$$E = E_d + \lambda E_s = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} - 2\boldsymbol{x}^T \boldsymbol{b} + c, \tag{3.102}$$

where $\boldsymbol{x} = [f(0,0) \ldots f(m-1,n-1)]$ is called the *state vector*.[22]

The sparse symmetric positive-definite matrix $\boldsymbol{A}$ is called the *Hessian* since it encodes the second derivative of the energy function.[23] For the one-dimensional, first-order problem, $\boldsymbol{A}$

[22] We use $\boldsymbol{x}$ instead of $\boldsymbol{f}$ because this is the more common form in the numerical analysis literature (Golub and Van Loan 1996).

[23] In numerical analysis, $\boldsymbol{A}$ is called the *coefficient* matrix (Saad 2003); in finite element analysis (Bathe 2007), it is called the *stiffness* matrix.

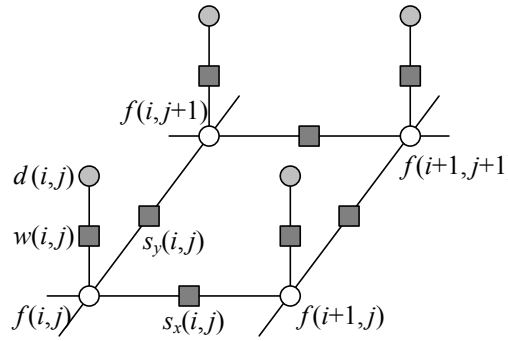


Figure 3.55 Graphical model interpretation of first-order regularization. The white circles are the unknowns $f(i,j)$ while the dark circles are the input data $d(i,j)$. In the resistive grid interpretation, the d and f values encode input and output voltages and the black squares denote resistors whose *conductance* is set to $s_x(i,j)$, $s_y(i,j)$, and $w(i,j)$. In the spring-mass system analogy, the circles denote elevations and the black squares denote springs. The same graphical model can be used to depict a first-order Markov random field (Figure 3.56).

is tridiagonal; for the two-dimensional, first-order problem, it is multi-banded with five non-zero entries per row. We call $\boldsymbol{b}$ the *weighted data vector*. Minimizing the above quadratic form is equivalent to solving the sparse linear system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{3.103}$$

which can be done using a variety of sparse matrix techniques, such as multigrid (Briggs, Henson, and McCormick 2000) and hierarchical preconditioners (Szeliski 2006b), as described in Appendix A.5.

While regularization was first introduced to the vision community by Poggio, Torre, and Koch (1985) and Terzopoulos (1986b) for problems such as surface interpolation, it was quickly adopted by other vision researchers for such varied problems as edge detection (Section 4.2), optical flow (Section 8.4), and shape from shading (Section 12.1) (Poggio, Torre, and Koch 1985; Horn and Brooks 1986; Terzopoulos 1986b; Bertero, Poggio, and Torre 1988; Brox, Bruhn, Papenberg *et al.* 2004). Poggio, Torre, and Koch (1985) also showed how the discrete energy defined by Equations (3.100–3.101) could be implemented in a resistive grid, as shown in Figure 3.55. In computational photography (Chapter 10), regularization and its variants are commonly used to solve problems such as high-dynamic range tone mapping (Fattal, Lischinski, and Werman 2002; Lischinski, Farbman, Uyttendaele *et al.* 2006a), Poisson and gradient-domain blending (Pérez, Gangnet, and Blake 2003; Levin, Zomet, Peleg *et al.* 2004; Agarwala, Dontcheva, Agrawala *et al.* 2004), colorization (Levin, Lischinski, and Weiss 2004), and natural image matting (Levin, Lischinski, and Weiss 2008).

Robust regularization

While regularization is most commonly formulated using quadratic (L_2) norms (compare with the squared derivatives in (3.92–3.95) and squared differences in (3.100–3.101)), it can also be formulated using non-quadratic *robust* penalty functions (Appendix B.3). For exam-

ple, (3.100) can be generalized to

$$E_{1r} = \sum_{i,j} s_x(i,j)\rho(f(i+1,j) - f(i,j)) + s_y(i,j)\rho(f(i,j+1) - f(i,j)), \tag{3.104}$$

where $\rho(x)$ is some monotonically increasing penalty function. For example, the family of norms $\rho(x) = |x|^p$ is called p-norms. When $p < 2$, the resulting smoothness terms become more piecewise continuous than totally smooth, which can better model the discontinuous nature of images, flow fields, and 3D surfaces.

An early example of robust regularization is the *graduated non-convexity* (GNC) algorithm introduced by Blake and Zisserman (1987). Here, the norms on the data and derivatives are clamped to a maximum value

$$\rho(x) = \min(x^2, V). \tag{3.105}$$

Because the resulting problem is highly non-convex (it has many local minima), a *continuation* method is proposed, where a quadratic norm (which is convex) is gradually replaced by the non-convex robust norm (Allgower and Georg 2003). (Around the same time, Terzopoulos (1988) was also using continuation to infer the tear and crease variables in his surface interpolation problems.)

Today, it is more common to use the L_1 ($p = 1$) norm, which is often called *total variation* (Chan, Osher, and Shen 2001; Tschumperlé and Deriche 2005; Tschumperlé 2006; Kaftory, Schechner, and Zeevi 2007). Other norms, for which the *influence* (derivative) more quickly decays to zero, are presented by Black and Rangarajan (1996); Black, Sapiro, Marimont *et al.* (1998) and discussed in Appendix B.3.

Even more recently, *hyper-Laplacian* norms with $p < 1$ have gained popularity, based on the observation that the log-likelihood distribution of image derivatives follows a $p \approx 0.5 - 0.8$ slope and is therefore a hyper-Laplacian distribution (Simoncelli 1999; Levin and Weiss 2007; Weiss and Freeman 2007; Krishnan and Fergus 2009). Such norms have an even stronger tendency to prefer large discontinuities over small ones. See the related discussion in Section 3.7.2 (3.114).

While least squares regularized problems using L_2 norms can be solved using linear systems, other p-norms require different iterative techniques, such as iteratively reweighted least squares (IRLS), Levenberg–Marquardt, or alternation between local non-linear subproblems and global quadratic regularization (Krishnan and Fergus 2009). Such techniques are discussed in Section 6.1.3 and Appendices A.3 and B.3.

3.7.2 Markov random fields

As we have just seen, regularization, which involves the minimization of energy functionals defined over (piecewise) continuous functions, can be used to formulate and solve a variety of low-level computer vision problems. An alternative technique is to formulate a *Bayesian* model, which separately models the noisy image formation (*measurement*) process, as well as assuming a statistical *prior* model over the solution space. In this section, we look at priors based on Markov random fields, whose log-likelihood can be described using local neighborhood interaction (or penalty) terms (Kindermann and Snell 1980; Geman and Geman 1984; Marroquin, Mitter, and Poggio 1987; Li 1995; Szeliski, Zabih, Scharstein *et al.* 2008).

The use of Bayesian modeling has several potential advantages over regularization (see also Appendix B). The ability to model measurement processes statistically enables us to extract the maximum information possible from each measurement, rather than just guessing what weighting to give the data. Similarly, the parameters of the prior distribution can often be *learned* by observing samples from the class we are modeling (Roth and Black 2007a; Tappen 2007; Li and Huttenlocher 2008). Furthermore, because our model is probabilistic, it is possible to estimate (in principle) complete probability *distributions* over the unknowns being recovered and, in particular, to model the *uncertainty* in the solution, which can be useful in latter processing stages. Finally, Markov random field models can be defined over *discrete* variables, such as image labels (where the variables have no proper ordering), for which regularization does not apply.

Recall from (3.68) in Section 3.4.3 (or see Appendix B.4) that, according to Bayes' Rule, the *posterior* distribution for a given set of measurements $\boldsymbol{y}$, $p(\boldsymbol{y}|\boldsymbol{x})$, combined with a prior $p(\boldsymbol{x})$ over the unknowns $\boldsymbol{x}$, is given by

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}, \tag{3.106}$$

where $p(\boldsymbol{y}) = \int_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$ is a normalizing constant used to make the $p(\boldsymbol{x}|\boldsymbol{y})$ distribution *proper* (integrate to 1). Taking the negative logarithm of both sides of (3.106), we get

$$-\log p(\boldsymbol{x}|\boldsymbol{y}) = -\log p(\boldsymbol{y}|\boldsymbol{x}) - \log p(\boldsymbol{x}) + C, \tag{3.107}$$

which is the *negative posterior log likelihood*.

To find the most likely (*maximum a posteriori* or MAP) solution $\boldsymbol{x}$ given some measurements $\boldsymbol{y}$, we simply minimize this negative log likelihood, which can also be thought of as an *energy*,

$$E(\boldsymbol{x}, \boldsymbol{y}) = E_d(\boldsymbol{x}, \boldsymbol{y}) + E_p(\boldsymbol{x}). \tag{3.108}$$

(We drop the constant C because its value does not matter during energy minimization.) The first term $E_d(\boldsymbol{x}, \boldsymbol{y})$ is the *data energy* or *data penalty*; it measures the negative log likelihood that the data were observed given the unknown state $\boldsymbol{x}$. The second term $E_p(\boldsymbol{x})$ is the *prior energy*; it plays a role analogous to the smoothness energy in regularization. Note that the MAP estimate may not always be desirable, since it selects the "peak" in the posterior distribution rather than some more stable statistic—see the discussion in Appendix B.2 and by Levin, Weiss, Durand *et al.* (2009).

For image processing applications, the unknowns $\boldsymbol{x}$ are the set of output pixels

$$\boldsymbol{x} = [f(0,0) \ldots f(m-1, n-1)],$$

and the data are (in the simplest case) the input pixels

$$\boldsymbol{y} = [d(0,0) \ldots d(m-1, n-1)]$$

as shown in Figure 3.56.

For a Markov random field, the probability $p(\boldsymbol{x})$ is a *Gibbs* or *Boltzmann distribution*, whose negative log likelihood (according to the Hammersley–Clifford theorem) can be written as a sum of pairwise interaction potentials,

$$E_p(\boldsymbol{x}) = \sum_{\{(i,j),(k,l)\} \in \mathcal{N}} V_{i,j,k,l}(f(i,j), f(k,l)), \tag{3.109}$$

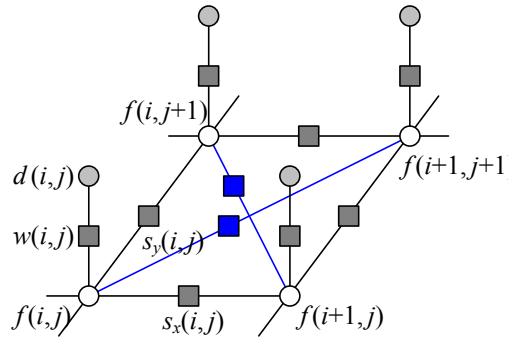


Figure 3.56 Graphical model for an $\mathcal{N}_4$ neighborhood Markov random field. (The blue edges are added for an $\mathcal{N}_8$ neighborhood.) The white circles are the unknowns $f(i,j)$, while the dark circles are the input data $d(i,j)$. The $s_x(i,j)$ and $s_y(i,j)$ black boxes denote arbitrary *interaction potentials* between adjacent nodes in the random field, and the $w(i,j)$ denote the *data penalty* functions. The same graphical model can be used to depict a discrete version of a first-order regularization problem (Figure 3.55).

where $\mathcal{N}(i,j)$ denotes the *neighbors* of pixel (i,j). In fact, the general version of the theorem says that the energy may have to be evaluated over a larger set of *cliques*, which depend on the *order* of the Markov random field (Kindermann and Snell 1980; Geman and Geman 1984; Bishop 2006; Kohli, Ladický, and Torr 2009; Kohli, Kumar, and Torr 2009).

The most commonly used neighborhood in Markov random field modeling is the $\mathcal{N}_4$ neighborhood, where each pixel in the field $f(i,j)$ interacts only with its immediate neighbors. The model in Figure 3.56, which we previously used in Figure 3.55 to illustrate the discrete version of first-order regularization, shows an $\mathcal{N}_4$ MRF. The $s_x(i,j)$ and $s_y(i,j)$ black boxes denote arbitrary *interaction potentials* between adjacent nodes in the random field and the $w(i,j)$ denote the data penalty functions. These square nodes can also be interpreted as *factors* in a *factor graph* version of the (undirected) graphical model (Bishop 2006), which is another name for interaction potentials. (Strictly speaking, the factors are (improper) probability functions whose product is the (un-normalized) posterior distribution.)

As we will see in (3.112–3.113), there is a close relationship between these interaction potentials and the discretized versions of regularized image restoration problems. Thus, to a first approximation, we can view energy minimization being performed when solving a regularized problem and the maximum a posteriori inference being performed in an MRF as equivalent.

While $\mathcal{N}_4$ neighborhoods are most commonly used, in some applications $\mathcal{N}_8$ (or even higher order) neighborhoods perform better at tasks such as image segmentation because they can better model discontinuities at different orientations (Boykov and Kolmogorov 2003; Rother, Kohli, Feng *et al.* 2009; Kohli, Ladický, and Torr 2009; Kohli, Kumar, and Torr 2009).

Binary MRFs

The simplest possible example of a Markov random field is a binary field. Examples of such fields include 1-bit (black and white) scanned document images as well as images segmented into foreground and background regions.

To denoise a scanned image, we set the data penalty to reflect the agreement between the

scanned and final images,

$$E_d(i,j) = w\delta(f(i,j), d(i,j)) \tag{3.110}$$

and the smoothness penalty to reflect the agreement between neighboring pixels

$$E_p(i,j) = E_x(i,j) + E_y(i,j) = s\delta(f(i,j), f(i+1,j)) + s\delta(f(i,j), f(i,j+1)). \tag{3.111}$$

Once we have formulated the energy, how do we minimize it? The simplest approach is to perform gradient descent, flipping one state at a time if it produces a lower energy. This approach is known as *contextual classification* (Kittler and Föglein 1984), *iterated conditional modes* (ICM) (Besag 1986), or *highest confidence first* (HCF) (Chou and Brown 1990) if the pixel with the largest energy decrease is selected first.

Unfortunately, these downhill methods tend to get easily stuck in local minima. An alternative approach is to add some randomness to the process, which is known as *stochastic gradient descent* (Metropolis, Rosenbluth, Rosenbluth *et al.* 1953; Geman and Geman 1984). When the amount of noise is decreased over time, this technique is known as *simulated annealing* (Kirkpatrick, Gelatt, and Vecchi 1983; Carnevali, Coletti, and Patarnello 1985; Wolberg and Pavlidis 1985; Swendsen and Wang 1987) and was first popularized in computer vision by Geman and Geman (1984) and later applied to stereo matching by Barnard (1989), among others.

Even this technique, however, does not perform that well (Boykov, Veksler, and Zabih 2001). For binary images, a much better technique, introduced to the computer vision community by Boykov, Veksler, and Zabih (2001) is to re-formulate the energy minimization as a *max-flow/min-cut* graph optimization problem (Greig, Porteous, and Seheult 1989). This technique has informally come to be known as *graph cuts* in the computer vision community (Boykov and Kolmogorov 2010). For simple energy functions, e.g., those where the penalty for non-identical neighboring pixels is a constant, this algorithm is guaranteed to produce the *global minimum*. Kolmogorov and Zabih (2004) formally characterize the class of binary energy potentials (*regularity conditions*) for which these results hold, while newer work by Komodakis, Tziritas, and Paragios (2008) and Rother, Kolmogorov, Lempitsky *et al.* (2007) provide good algorithms for the cases when they do not.

In addition to the above mentioned techniques, a number of other optimization approaches have been developed for MRF energy minimization, such as (loopy) belief propagation and dynamic programming (for one-dimensional problems). These are discussed in more detail in Appendix B.5 as well as the comparative survey paper by Szeliski, Zabih, Scharstein *et al.* (2008).

Ordinal-valued MRFs

In addition to binary images, Markov random fields can be applied to ordinal-valued labels such as grayscale images or depth maps. The term "ordinal" indicates that the labels have an implied ordering, e.g., that higher values are lighter pixels. In the next section, we look at unordered labels, such as source image labels for image compositing.

In many cases, it is common to extend the binary data and smoothness prior terms as

$$E_d(i,j) = w(i,j)\rho_d(f(i,j) - d(i,j)) \tag{3.112}$$



(a) (b) (c) (d)

Figure 3.57 Grayscale image denoising and inpainting: (a) original image; (b) image corrupted by noise and with missing data (black bar); (c) image restored using loopy belief propagation; (d) image restored using expansion move graph cuts. Images are from http://vision.middlebury.edu/MRF/results/ (Szeliski, Zabih, Scharstein *et al.* 2008).

and

$$E_p(i,j) = s_x(i,j)\rho_p(f(i,j) - f(i+1,j)) + s_y(i,j)\rho_p(f(i,j) - f(i,j+1)), \quad (3.113)$$

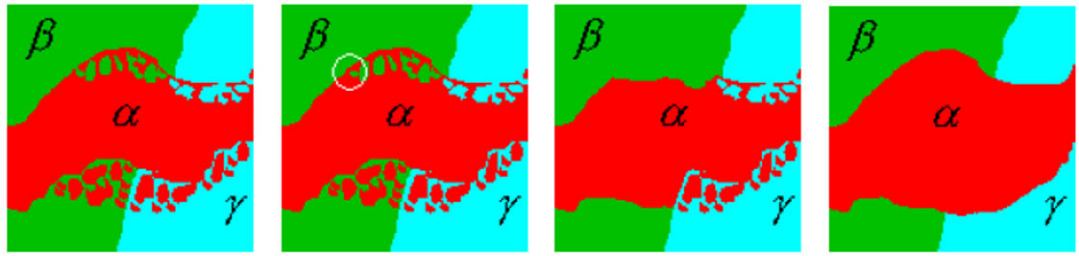
which are robust generalizations of the quadratic penalty terms (3.101) and (3.100), first introduced in (3.105). As before, the $w(i,j)$, $s_x(i,j)$ and $s_y(i,j)$ weights can be used to locally control the data weighting and the horizontal and vertical smoothness. Instead of using a quadratic penalty, however, a general monotonically increasing penalty function $\rho()$ is used. (Different functions can be used for the data and smoothness terms.) For example, ρ_p can be a hyper-Laplacian penalty

$$\rho_p(d) = |d|^p, \;\; p < 1, \quad (3.114)$$

which better encodes the distribution of gradients (mainly edges) in an image than either a quadratic or linear (total variation) penalty.[24] Levin and Weiss (2007) use such a penalty to separate a transmitted and reflected image (Figure 8.17) by encouraging gradients to lie in one or the other image, but not both. More recently, Levin, Fergus, Durand *et al.* (2007) use the hyper-Laplacian as a prior for image deconvolution (deblurring) and Krishnan and Fergus (2009) develop a faster algorithm for solving such problems. For the data penalty, ρ_d can be quadratic (to model Gaussian noise) or the log of a *contaminated Gaussian* (Appendix B.3).

When ρ_p is a quadratic function, the resulting Markov random field is called a Gaussian Markov random field (GMRF) and its minimum can be found by sparse linear system solving (3.103). When the weighting functions are uniform, the GMRF becomes a special case of Wiener filtering (Section 3.4.3). Allowing the weighting functions to depend on the input image (a special kind of conditional random field, which we describe below) enables quite sophisticated image processing algorithms to be performed, including colorization (Levin, Lischinski, and Weiss 2004), interactive tone mapping (Lischinski, Farbman, Uyttendaele *et*

[24] Note that, unlike a quadratic penalty, the sum of the horizontal and vertical derivative p-norms is not rotationally invariant. A better approach may be to locally estimate the gradient direction and to impose different norms on the perpendicular and parallel components, which Roth and Black (2007b) call a *steerable random field*.



(a) initial labeling (b) standard move (c) α-β-swap (d) α-expansion

Figure 3.58 Multi-level graph optimization from (Boykov, Veksler, and Zabih 2001) © 2001 IEEE: (a) initial problem configuration; (b) the standard move only changes one pixel; (c) the α-β-swap optimally exchanges all α and β-labeled pixels; (d) the α-expansion move optimally selects among current pixel values and the α label.

al. 2006a), natural image matting (Levin, Lischinski, and Weiss 2008), and image restoration (Tappen, Liu, Freeman *et al.* 2007).

When ρ_d or ρ_p are non-quadratic functions, gradient descent techniques such as non-linear least squares or iteratively re-weighted least squares can sometimes be used (Appendix A.3). However, if the search space has lots of local minima, as is the case for stereo matching (Barnard 1989; Boykov, Veksler, and Zabih 2001), more sophisticated techniques are required.

The extension of graph cut techniques to multi-valued problems was first proposed by Boykov, Veksler, and Zabih (2001). In their paper, they develop two different algorithms, called the *swap move* and the *expansion move*, which iterate among a series of binary labeling sub-problems to find a good solution (Figure 3.58). Note that a global solution is generally not achievable, as the problem is provably NP-hard for general energy functions. Because both these algorithms use a binary MRF optimization inside their inner loop, they are subject to the kind of constraints on the energy functions that occur in the binary labeling case (Kolmogorov and Zabih 2004). Appendix B.5.4 discusses these algorithms in more detail, along with some more recently developed approaches to this problem.

Another MRF inference technique is *belief propagation* (BP). While belief propagation was originally developed for inference over trees, where it is exact (Pearl 1988), it has more recently been applied to graphs with loops such as Markov random fields (Freeman, Pasztor, and Carmichael 2000; Yedidia, Freeman, and Weiss 2001). In fact, some of the better performing stereo-matching algorithms use loopy belief propagation (LBP) to perform their inference (Sun, Zheng, and Shum 2003). LBP is discussed in more detail in Appendix B.5.3 as well as the comparative survey paper on MRF optimization (Szeliski, Zabih, Scharstein *et al.* 2008).

Figure 3.57 shows an example of image denoising and inpainting (hole filling) using a non-quadratic energy function (non-Gaussian MRF). The original image has been corrupted by noise and a portion of the data has been removed (the black bar). In this case, the loopy belief propagation algorithm computes a slightly lower energy and also a smoother image than the alpha-expansion graph cut algorithm.

Of course, the above formula (3.113) for the smoothness term $E_p(i, j)$ just shows the simplest case. In more recent work, Roth and Black (2009) propose a *Field of Experts* (FoE) model, which sums up a large number of exponentiated local filter outputs to arrive at the

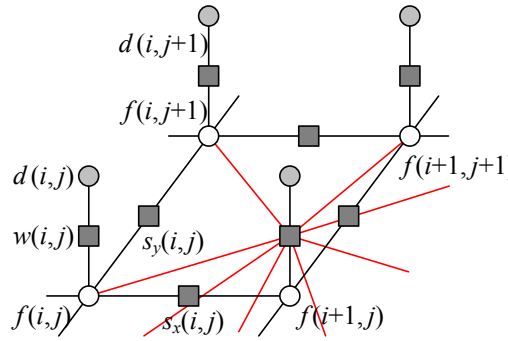


Figure 3.59 Graphical model for a Markov random field with a more complex measurement model. The additional colored edges show how combinations of unknown values (say, in a sharp image) produce the measured values (a noisy blurred image). The resulting graphical model is still a classic MRF and is just as easy to sample from, but some inference algorithms (e.g., those based on graph cuts) may not be applicable because of the increased network complexity, since state changes during the inference become more entangled and the posterior MRF has much larger cliques.

smoothness penalty. Weiss and Freeman (2007) analyze this approach and compare it to the simpler hyper-Laplacian model of natural image statistics. Lyu and Simoncelli (2009) use *Gaussian Scale Mixtures* (GSMs) to construct an inhomogeneous multi-scale MRF, with one (positive exponential) GMRF modulating the variance (amplitude) of another Gaussian MRF.

It is also possible to extend the *measurement* model to make the sampled (noise-corrupted) input pixels correspond to blends of unknown (latent) image pixels, as in Figure 3.59. This is the commonly occurring case when trying to de-blur an image. While this kind of a model is still a traditional generative Markov random field, finding an optimal solution can be difficult because the clique sizes get larger. In such situations, gradient descent techniques, such as iteratively reweighted least squares, can be used (Joshi, Zitnick, Szeliski *et al.* 2009). Exercise 3.31 has you explore some of these issues.

Unordered labels

Another case with multi-valued labels where Markov random fields are often applied are *unordered labels*, i.e., labels where there is no semantic meaning to the numerical difference between the values of two labels. For example, if we are classifying terrain from aerial imagery, it makes no sense to take the numeric difference between the labels assigned to forest, field, water, and pavement. In fact, the adjacencies of these various kinds of terrain each have different likelihoods, so it makes more sense to use a prior of the form

$$E_p(i,j) = s_x(i,j)V(l(i,j), l(i+1,j)) + s_y(i,j)V(l(i,j), l(i,j+1)), \tag{3.115}$$

where $V(l_0, l_1)$ is a general *compatibility* or *potential* function. (Note that we have also replaced $f(i,j)$ with $l(i,j)$ to make it clearer that these are labels rather than discrete function samples.) An alternative way to write this prior energy (Boykov, Veksler, and Zabih 2001;



Figure 3.60 An unordered label MRF (Agarwala, Dontcheva, Agrawala *et al.* 2004) © 2004 ACM: Strokes in each of the source images on the left are used as constraints on an MRF optimization, which is solved using graph cuts. The resulting multi-valued label field is shown as a color overlay in the middle image, and the final composite is shown on the right.

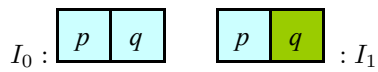
Szeliski, Zabih, Scharstein *et al.* 2008) is

$$E_p = \sum_{(p,q)\in\mathcal{N}} V_{p,q}(l_p, l_q), \tag{3.116}$$

where the (p, q) are neighboring pixels and a spatially varying potential function $V_{p,q}$ is evaluated for each neighboring pair.

An important application of unordered MRF labeling is seam finding in image compositing (Davis 1998; Agarwala, Dontcheva, Agrawala *et al.* 2004) (see Figure 3.60, which is explained in more detail in Section 9.3.2). Here, the compatibility $V_{p,q}(l_p, l_q)$ measures the quality of the visual appearance that would result from placing a pixel p from image l_p next to a pixel q from image l_q. As with most MRFs, we assume that $V_{p,q}(l, l) = 0$, i.e., it is perfectly fine to choose contiguous pixels from the same image. For different labels, however, the compatibility $V_{p,q}(l_p, l_q)$ may depend on the values of the underlying pixels $I_{l_p}(p)$ and $I_{l_q}(q)$.

Consider, for example, where one image I_0 is all sky blue, i.e., $I_0(p) = I_0(q) = B$, while the other image I_1 has a transition from sky blue, $I_1(p) = B$, to forest green, $I_1(q) = G$.



In this case, $V_{p,q}(1, 0) = 0$ (the colors agree), while $V_{p,q}(0, 1) > 0$ (the colors disagree).

Conditional random fields

In a classic Bayesian model (3.106–3.108),

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}), \tag{3.117}$$

the prior distribution $p(\boldsymbol{x})$ is independent of the observations $\boldsymbol{y}$. Sometimes, however, it is useful to modify our prior assumptions, say about the smoothness of the field we are trying to estimate, in response to the sensed data. Whether this makes sense from a probability viewpoint is something we discuss once we have explained the new model.

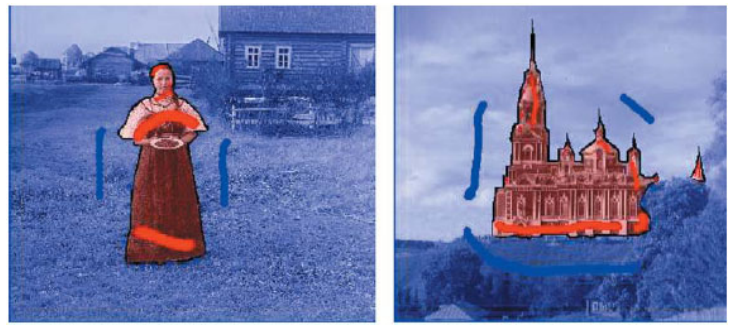


Figure 3.61 Image segmentation (Boykov and Funka-Lea 2006) © 2006 Springer: The user draws a few red strokes in the foreground object and a few blue ones in the background. The system computes color distributions for the foreground and background and solves a binary MRF. The smoothness weights are modulated by the intensity gradients (edges), which makes this a conditional random field (CRF).

Consider the interactive image segmentation problem shown in Figure 3.61 (Boykov and Funka-Lea 2006). In this application, the user draws foreground (red) and background (blue) strokes, and the system then solves a binary MRF labeling problem to estimate the extent of the foreground object. In addition to minimizing a data term, which measures the pointwise similarity between pixel colors and the inferred region distributions (Section 5.5), the MRF is modified so that the smoothness terms $s_x(x, y)$ and $s_y(x, y)$ in Figure 3.56 and (3.113) depend on the magnitude of the gradient between adjacent pixels.[25]

Since the smoothness term now depends on the data, Bayes' Rule (3.117) no longer applies. Instead, we use a direct model for the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y})$, whose negative log likelihood can be written as

$$\begin{aligned} E(\boldsymbol{x}|\boldsymbol{y}) &= E_d(\boldsymbol{x}, \boldsymbol{y}) + E_s(\boldsymbol{x}, \boldsymbol{y}) \\ &= \sum_p V_p(x_p, \boldsymbol{y}) + \sum_{(p,q)\in\mathcal{N}} V_{p,q}(x_p, x_q, \boldsymbol{y}), \end{aligned} \tag{3.118}$$

using the notation introduced in (3.116). The resulting probability distribution is called a *conditional random field* (CRF) and was first introduced to the computer vision field by Kumar and Hebert (2003), based on earlier work in text modeling by Lafferty, McCallum, and Pereira (2001).

Figure 3.62 shows a graphical model where the smoothness terms depend on the data values. In this particular model, each smoothness term depends only on its adjacent pair of data values, i.e., terms are of the form $V_{p,q}(x_p, x_q, y_p, y_q)$ in (3.118).

The idea of modifying smoothness terms in response to input data is not new. For example, Boykov and Jolly (2001) used this idea for interactive segmentation, as shown in Figure 3.61, and it is now widely used in image segmentation (Section 5.5) (Blake, Rother, Brown *et al.* 2004; Rother, Kolmogorov, and Blake 2004), denoising (Tappen, Liu, Freeman *et al.* 2007), and object recognition (Section 14.4.3) (Winn and Shotton 2006; Shotton, Winn, Rother *et al.* 2009).

[25] An alternative formulation that also uses detected edges to modulate the smoothness of a depth or motion field and hence to integrate multiple lower level vision modules is presented by Poggio, Gamble, and Little (1988).

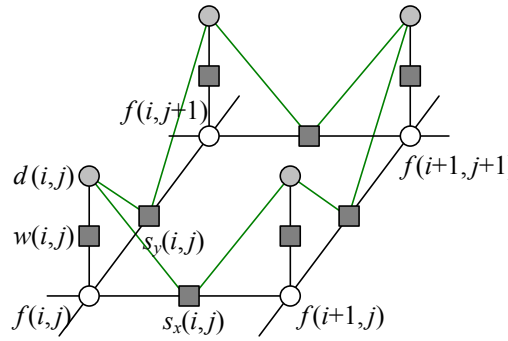


Figure 3.62 Graphical model for a conditional random field (CRF). The additional green edges show how combinations of sensed data influence the smoothness in the underlying MRF prior model, i.e., $s_x(i,j)$ and $s_y(i,j)$ in (3.113) depend on adjacent $d(i,j)$ values. These additional links (factors) enable the smoothness to depend on the input data. However, they make sampling from this MRF more complex.

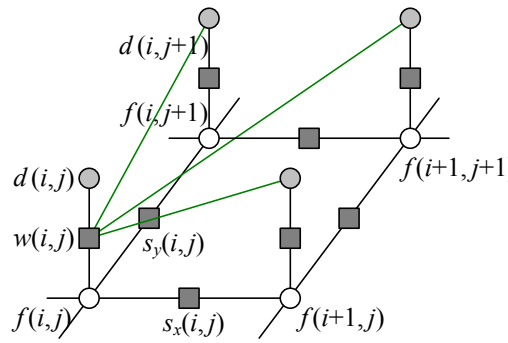


Figure 3.63 Graphical model for a discriminative random field (DRF). The additional green edges show how combinations of sensed data, e.g., $d(i,j+1)$, influence the data term for $f(i,j)$. The generative model is therefore more complex, i.e., we cannot just apply a simple function to the unknown variables and add noise.

In stereo matching, the idea of encouraging disparity discontinuities to coincide with intensity edges goes back even further to the early days of optimization and MRF-based algorithms (Poggio, Gamble, and Little 1988; Fua 1993; Bobick and Intille 1999; Boykov, Veksler, and Zabih 2001) and is discussed in more detail in (Section 11.5).

In addition to using smoothness terms that adapt to the input data, Kumar and Hebert (2003) also compute a neighborhood function over the input data for each $V_p(x_p, \boldsymbol{y})$ term, as illustrated in Figure 3.63, instead of using the classic unary MRF data term $V_p(x_p, y_p)$ shown in Figure 3.56.[26] Because such neighborhood functions can be thought of as *discriminant* functions (a term widely used in machine learning (Bishop 2006)), they call the resulting graphical model a *discriminative random field* (DRF). In their paper, Kumar and Hebert (2006) show that DRFs outperform similar CRFs on a number of applications, such as structure detection (Figure 3.64) and binary image denoising.

Here again, one could argue that previous stereo correspondence algorithms also look at

[26] Kumar and Hebert (2006) call the unary potentials $V_p(x_p, \boldsymbol{y})$ *association potentials* and the pairwise potentials $V_{p,q}(x_p, y_q, \boldsymbol{y})$ *interaction potentials*.

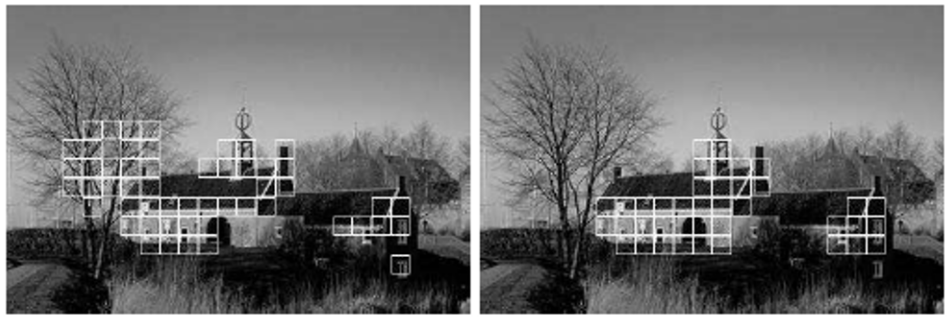


Figure 3.64 Structure detection results using an MRF (left) and a DRF (right) (Kumar and Hebert 2006) © 2006 Springer.

a neighborhood of input data, either explicitly, because they compute correlation measures (Criminisi, Cross, Blake *et al.* 2006) as data terms, or implicitly, because even pixel-wise disparity costs look at several pixels in either the left or right image (Barnard 1989; Boykov, Veksler, and Zabih 2001).

What, then are the advantages and disadvantages of using conditional or discriminative random fields instead of MRFs?

Classic Bayesian inference (MRF) assumes that the prior distribution of the data is independent of the measurements. This makes a lot of sense: if you see a pair of sixes when you first throw a pair of dice, it would be unwise to assume that they will always show up thereafter. However, if after playing for a long time you detect a statistically significant bias, you may want to adjust your prior. What CRFs do, in essence, is to select or modify the prior model based on observed data. This can be viewed as making a partial inference over additional hidden variables or correlations between the unknowns (say, a label, depth, or clean image) and the knowns (observed images).

In some cases, the CRF approach makes a lot of sense and is, in fact, the only plausible way to proceed. For example, in grayscale image colorization (Section 10.3.2) (Levin, Lischinski, and Weiss 2004), the best way to transfer the continuity information from the input grayscale image to the unknown color image is to modify local smoothness constraints. Similarly, for simultaneous segmentation and recognition (Winn and Shotton 2006; Shotton, Winn, Rother *et al.* 2009), it makes a lot of sense to permit strong color edges to influence the semantic image label continuities.

In other cases, such as image denoising, the situation is more subtle. Using a non-quadratic (robust) smoothness term as in (3.113) plays a qualitatively similar role to setting the smoothness based on local gradient information in a Gaussian MRF (GMRF) (Tappen, Liu, Freeman *et al.* 2007). (In more recent work, Tanaka and Okutomi (2008) use a larger neighborhood and full covariance matrix on a related Gaussian MRF.) The advantage of Gaussian MRFs, when the smoothness can be correctly inferred, is that the resulting quadratic energy can be minimized in a single step. However, for situations where the discontinuities are not self-evident in the input data, such as for piecewise-smooth sparse data interpolation (Blake and Zisserman 1987; Terzopoulos 1988), classic robust smoothness energy minimization may be preferable. Thus, as with most computer vision algorithms, a careful analysis of

the problem at hand and desired robustness and computation constraints may be required to choose the best technique.

Perhaps the biggest advantage of CRFs and DRFs, as argued by Kumar and Hebert (2006), Tappen, Liu, Freeman *et al.* (2007) and Blake, Rother, Brown *et al.* (2004), is that learning the model parameters is sometimes easier. While learning parameters in MRFs and their variants is not a topic that we cover in this book, interested readers can find more details in recently published articles (Kumar and Hebert 2006; Roth and Black 2007a; Tappen, Liu, Freeman *et al.* 2007; Tappen 2007; Li and Huttenlocher 2008).

3.7.3 *Application*: Image restoration

In Section 3.4.4, we saw how two-dimensional linear and non-linear filters can be used to remove noise or enhance sharpness in images. Sometimes, however, images are degraded by larger problems, such as scratches and blotches (Kokaram 2004). In this case, Bayesian methods such as MRFs, which can model spatially varying per-pixel measurement noise, can be used instead. An alternative is to use hole filling or inpainting techniques (Bertalmio, Sapiro, Caselles *et al.* 2000; Bertalmio, Vese, Sapiro *et al.* 2003; Criminisi, Pérez, and Toyama 2004), as discussed in Sections 5.1.4 and 10.5.1.

Figure 3.57 shows an example of image denoising and inpainting (hole filling) using a Markov random field. The original image has been corrupted by noise and a portion of the data has been removed. In this case, the loopy belief propagation algorithm computes a slightly lower energy and also a smoother image than the alpha-expansion graph cut algorithm.

3.8 Additional reading

If you are interested in exploring the topic of image processing in more depth, some popular textbooks have been written by Lim (1990); Crane (1997); Gomes and Velho (1997); Jähne (1997); Pratt (2007); Russ (2007); Burger and Burge (2008); Gonzales and Woods (2008). The pre-eminent conference and journal in this field are the IEEE Conference on Image Processsing and the IEEE Transactions on Image Processing.

For image compositing operators, the seminal reference is by Porter and Duff (1984) while Blinn (1994a,b) provides a more detailed tutorial. For image compositing, Smith and Blinn (1996) were the first to bring this topic to the attention of the graphics community, while Wang and Cohen (2007a) provide a recent in-depth survey.

In the realm of linear filtering, Freeman and Adelson (1991) provide a great introduction to separable and steerable oriented band-pass filters, while Perona (1995) shows how to approximate any filter as a sum of separable components.

The literature on non-linear filtering is quite wide and varied; it includes such topics as bilateral filtering (Tomasi and Manduchi 1998; Durand and Dorsey 2002; Paris and Durand 2006; Chen, Paris, and Durand 2007; Paris, Kornprobst, Tumblin *et al.* 2008), related iterative algorithms (Saint-Marc, Chen, and Medioni 1991; Nielsen, Florack, and Deriche 1997; Black, Sapiro, Marimont *et al.* 1998; Weickert, ter Haar Romeny, and Viergever 1998; Weickert 1998; Barash 2002; Scharr, Black, and Haussecker 2003; Barash and Comaniciu 2004),

and variational approaches (Chan, Osher, and Shen 2001; Tschumperlé and Deriche 2005; Tschumperlé 2006; Kaftory, Schechner, and Zeevi 2007).

Good references to image morphology include (Haralick and Shapiro 1992, Section 5.2; Bovik 2000, Section 2.2; Ritter and Wilson 2000, Section 7; Serra 1982; Serra and Vincent 1992; Yuille, Vincent, and Geiger 1992; Soille 2006).

The classic papers for image pyramids and pyramid blending are by Burt and Adelson (1983a,b). Wavelets were first introduced to the computer vision community by Mallat (1989) and good tutorial and review papers and books are available (Strang 1989; Simoncelli and Adelson 1990b; Rioul and Vetterli 1991; Chui 1992; Meyer 1993; Sweldens 1997). Wavelets are widely used in the computer graphics community to perform multi-resolution geometric processing (Stollnitz, DeRose, and Salesin 1996) and have been used in computer vision for similar applications (Szeliski 1990b; Pentland 1994; Gortler and Cohen 1995; Yaou and Chang 1994; Lai and Vemuri 1997; Szeliski 2006b), as well as for multi-scale oriented filtering (Simoncelli, Freeman, Adelson *et al.* 1992) and denoising (Portilla, Strela, Wainwright *et al.* 2003).

While image pyramids (Section 3.5.3) are usually constructed using linear filtering operators, some recent work has started investigating non-linear filters, since these can better preserve details and other salient features. Some representative papers in the computer vision literature are by Gluckman (2006a,b); Lyu and Simoncelli (2008) and in computational photography by Bae, Paris, and Durand (2006); Farbman, Fattal, Lischinski *et al.* (2008); Fattal (2009).

High-quality algorithms for image warping and resampling are covered both in the image processing literature (Wolberg 1990; Dodgson 1992; Gomes, Darsa, Costa *et al.* 1999; Szeliski, Winder, and Uyttendaele 2010) and in computer graphics (Williams 1983; Heckbert 1986; Barkans 1997; Akenine-Möller and Haines 2002), where they go under the name of *texture mapping*. Combination of image warping and image blending techniques are used to enable *morphing* between images, which is covered in a series of seminal papers and books (Beier and Neely 1992; Gomes, Darsa, Costa *et al.* 1999).

The regularization approach to computer vision problems was first introduced to the vision community by Poggio, Torre, and Koch (1985) and Terzopoulos (1986a,b, 1988) and continues to be a popular framework for formulating and solving low-level vision problems (Ju, Black, and Jepson 1996; Nielsen, Florack, and Deriche 1997; Nordström 1990; Brox, Bruhn, Papenberg *et al.* 2004; Levin, Lischinski, and Weiss 2008). More detailed mathematical treatment and additional applications can be found in the applied mathematics and statistics literature (Tikhonov and Arsenin 1977; Engl, Hanke, and Neubauer 1996).

The literature on Markov random fields is truly immense, with publications in related fields such as optimization and control theory of which few vision practitioners are even aware. A good guide to the latest techniques is the book edited by Blake, Kohli, and Rother (2010). Other recent articles that contain nice literature reviews or experimental comparisons include (Boykov and Funka-Lea 2006; Szeliski, Zabih, Scharstein *et al.* 2008; Kumar, Veksler, and Torr 2010).

The seminal paper on Markov random fields is the work of Geman and Geman (1984), who introduced this formalism to computer vision researchers and also introduced the notion of *line processes*, additional binary variables that control whether smoothness penalties are enforced or not. Black and Rangarajan (1996) showed how independent line processes

could be replaced with robust pairwise potentials; Boykov, Veksler, and Zabih (2001) developed iterative binary, graph cut algorithms for optimizing multi-label MRFs; Kolmogorov and Zabih (2004) characterized the class of binary energy potentials required for these techniques to work; and Freeman, Pasztor, and Carmichael (2000) popularized the use of loopy belief propagation for MRF inference. Many more additional references can be found in Sections 3.7.2 and 5.5, and Appendix B.5.

3.9 Exercises

Ex 3.1: Color balance Write a simple application to change the color balance of an image by multiplying each color value by a different user-specified constant. If you want to get fancy, you can make this application interactive, with sliders.

1. Do you get different results if you take out the gamma transformation before or after doing the multiplication? Why or why not?
2. Take the same picture with your digital camera using different color balance settings (most cameras control the color balance from one of the menus). Can you recover what the color balance ratios are between the different settings? You may need to put your camera on a tripod and align the images manually or automatically to make this work. Alternatively, use a color checker chart (Figure 10.3b), as discussed in Sections 2.3 and 10.1.1.
3. If you have access to the RAW image for the camera, perform the demosaicing yourself (Section 10.3.1) or downsample the image resolution to get a "true" RGB image. Does your camera perform a simple linear mapping between RAW values and the color-balanced values in a JPEG? Some high-end cameras have a RAW+JPEG mode, which makes this comparison much easier.
4. Can you think of any reason why you might want to perform a color twist (Section 3.1.2) on the images? See also Exercise 2.9 for some related ideas.

Ex 3.2: Compositing and reflections Section 3.1.3 describes the process of compositing an alpha-matted image on top of another. Answer the following questions and optionally validate them experimentally:

1. Most captured images have gamma correction applied to them. Does this invalidate the basic compositing equation (3.8); if so, how should it be fixed?
2. The additive (pure reflection) model may have limitations. What happens if the glass is tinted, especially to a non-gray hue? How about if the glass is dirty or smudged? How could you model wavy glass or other kinds of refractive objects?

Ex 3.3: Blue screen matting Set up a blue or green background, e.g., by buying a large piece of colored posterboard. Take a picture of the empty background, and then of the background with a new object in front of it. *Pull the matte* using the difference between each colored pixel and its assumed corresponding background pixel, using one of the techniques described in Section 3.1.3) or by Smith and Blinn (1996).

Ex 3.4: Difference keying Implement a difference keying algorithm (see Section 3.1.3) (Toyama, Krumm, Brumitt *et al.* 1999), consisting of the following steps:

1. Compute the mean and variance (or median and robust variance) at each pixel in an "empty" video sequence.
2. For each new frame, classify each pixel as foreground or background (set the background pixels to RGBA=0).
3. (Optional) Compute the alpha channel and composite over a new background.
4. (Optional) Clean up the image using morphology (Section 3.3.1), label the connected components (Section 3.3.4), compute their centroids, and track them from frame to frame. Use this to build a "people counter".

Ex 3.5: Photo effects Write a variety of photo enhancement or effects filters: contrast, solarization (quantization), etc. Which ones are useful (perform sensible corrections) and which ones are more creative (create unusual images)?

Ex 3.6: Histogram equalization Compute the gray level (luminance) histogram for an image and equalize it so that the tones look better (and the image is less sensitive to exposure settings). You may want to use the following steps:

1. Convert the color image to luminance (Section 3.1.2).
2. Compute the histogram, the cumulative distribution, and the compensation transfer function (Section 3.1.4).
3. (Optional) Try to increase the "punch" in the image by ensuring that a certain fraction of pixels (say, 5%) are mapped to pure black and white.
4. (Optional) Limit the local *gain* $f'(I)$ in the transfer function. One way to do this is to limit $f(I) < \gamma I$ or $f'(I) < \gamma$ while performing the accumulation (3.9), keeping any unaccumulated values "in reserve". (I'll let you figure out the exact details.)
5. Compensate the luminance channel through the lookup table and re-generate the color image using color ratios (2.116).
6. (Optional) Color values that are *clipped* in the original image, i.e., have one or more saturated color channels, may appear unnatural when remapped to a non-clipped value. Extend your algorithm to handle this case in some useful way.

Ex 3.7: Local histogram equalization Compute the gray level (luminance) histograms for each patch, but add to vertices based on distance (a spline).

1. Build on Exercise 3.6 (luminance computation).
2. Distribute values (counts) to adjacent vertices (bilinear).
3. Convert to CDF (look-up functions).
4. (Optional) Use low-pass filtering of CDFs.

5. Interpolate adjacent CDFs for final lookup.

Ex 3.8: Padding for neighborhood operations Write down the formulas for computing the padded pixel values $\tilde{f}(i,j)$ as a function of the original pixel values $f(k,l)$ and the image width and height (M,N) for *each* of the padding modes shown in Figure 3.13. For example, for replication (clamping),

$$\tilde{f}(i,j) = f(k,l), \quad \begin{matrix} k = \max(0, \min(M-1, i)), \\ l = \max(0, \min(N-1, j)), \end{matrix}$$

(Hint: you may want to use the min, max, mod, and absolute value operators in addition to the regular arithmetic operators.)

- Describe in more detail the advantages and disadvantages of these various modes.
- (Optional) Check what your graphics card does by drawing a texture-mapped rectangle where the texture coordinates lie beyond the $[0.0, 1.0]$ range and using different texture clamping modes.

Ex 3.9: Separable filters Implement convolution with a separable kernel. The input should be a grayscale or color image along with the horizontal and vertical kernels. Make sure you support the padding mechanisms developed in the previous exercise. You will need this functionality for some of the later exercises. If you already have access to separable filtering in an image processing package you are using (such as IPL), skip this exercise.

- (Optional) Use Pietro Perona's (1995) technique to approximate convolution as a sum of a number of separable kernels. Let the user specify the number of kernels and report back some sensible metric of the approximation fidelity.

Ex 3.10: Discrete Gaussian filters Discuss the following issues with implementing a discrete Gaussian filter:

- If you just sample the equation of a continuous Gaussian filter at discrete locations, will you get the desired properties, e.g., will the coefficients sum up to 0? Similarly, if you sample a derivative of a Gaussian, do the samples sum up to 0 or have vanishing higher-order moments?
- Would it be preferable to take the original signal, interpolate it with a sinc, blur with a continuous Gaussian, then pre-filter with a sinc before re-sampling? Is there a simpler way to do this in the frequency domain?
- Would it make more sense to produce a Gaussian frequency response in the Fourier domain and to then take an inverse FFT to obtain a discrete filter?
- How does truncation of the filter change its frequency response? Does it introduce any additional artifacts?
- Are the resulting two-dimensional filters as rotationally invariant as their continuous analogs? Is there some way to improve this? In fact, can any 2D discrete (separable or non-separable) filter be truly rotationally invariant?

Ex 3.11: Sharpening, blur, and noise removal Implement some softening, sharpening, and non-linear diffusion (selective sharpening or noise removal) filters, such as Gaussian, median, and bilateral (Section 3.3.1), as discussed in Section 3.4.4.

Take blurry or noisy images (shooting in low light is a good way to get both) and try to improve their appearance and legibility.

Ex 3.12: Steerable filters Implement Freeman and Adelson's (1991) steerable filter algorithm. The input should be a grayscale or color image and the output should be a multi-banded image consisting of $G_1^{0^\circ}$ and $G_1^{90^\circ}$. The coefficients for the filters can be found in the paper by Freeman and Adelson (1991).

Test the various order filters on a number of images of your choice and see if you can reliably find corner and intersection features. These filters will be quite useful later to detect elongated structures, such as lines (Section 4.3).

Ex 3.13: Distance transform Implement some (raster-scan) algorithms for city block and Euclidean distance transforms. Can you do it without peeking at the literature (Danielsson 1980; Borgefors 1986)? If so, what problems did you come across and resolve?

Later on, you can use the distance functions you compute to perform *feathering* during image stitching (Section 9.3.2).

Ex 3.14: Connected components Implement one of the connected component algorithms from Section 3.3.4 or Section 2.3 from Haralick and Shapiro's book (1992) and discuss its computational complexity.

- Threshold or quantize an image to obtain a variety of input labels and then compute the area statistics for the regions that you find.
- Use the connected components that you have found to track or match regions in different images or video frames.

Ex 3.15: Fourier transform Prove the properties of the Fourier transform listed in Table 3.1 and derive the formulas for the Fourier transforms listed in Tables 3.2 and 3.3. These exercises are very useful if you want to become comfortable working with Fourier transforms, which is a very useful skill when analyzing and designing the behavior and efficiency of many computer vision algorithms.

Ex 3.16: Wiener filtering Estimate the frequency spectrum of your personal photo collection and use it to perform Wiener filtering on a few images with varying degrees of noise.

1. Collect a few hundred of your images by re-scaling them to fit within a 512×512 window and cropping them.
2. Take their Fourier transforms, throw away the phase information, and average together all of the spectra.
3. Pick two of your favorite images and add varying amounts of Gaussian noise, $\sigma_n \in \{1, 2, 5, 10, 20\}$ gray levels.
4. For each combination of image and noise, determine by eye which width of a Gaussian blurring filter σ_s gives the best denoised result. You will have to make a subjective decision between sharpness and noise.

5. Compute the Wiener filtered version of all the noised images and compare them against your hand-tuned Gaussian-smoothed images.

6. (Optional) Do your image spectra have a lot of energy concentrated along the horizontal and vertical axes ($f_x = 0$ and $f_y = 0$)? Can you think of an explanation for this? Does rotating your image samples by $45°$ move this energy to the diagonals? If not, could it be due to edge effects in the Fourier transform? Can you suggest some techniques for reducing such effects?

Ex 3.17: Deblurring using Wiener filtering Use Wiener filtering to deblur some images.

1. Modify the Wiener filter derivation (3.66–3.74) to incorporate blur (3.75).

2. Discuss the resulting Wiener filter in terms of its noise suppression and frequency boosting characteristics.

3. Assuming that the blur kernel is Gaussian and the image spectrum follows an inverse frequency law, compute the frequency response of the Wiener filter, and compare it to the unsharp mask.

4. Synthetically blur two of your sample images with Gaussian blur kernels of different radii, add noise, and then perform Wiener filtering.

5. Repeat the above experiment with a "pillbox" (disc) blurring kernel, which is characteristic of a finite aperture lens (Section 2.2.3). Compare these results to Gaussian blur kernels (be sure to inspect your frequency plots).

6. It has been suggested that regular apertures are anathema to de-blurring because they introduce zeros in the sensed frequency spectrum (Veeraraghavan, Raskar, Agrawal *et al.* 2007). Show that this is indeed an issue if no prior model is assumed for the signal, i.e., $P_s^{-1}\boldsymbol{l}1$. If a reasonable power spectrum is assumed, is this still a problem (do we still get banding or ringing artifacts)?

Ex 3.18: High-quality image resampling Implement several of the low-pass filters presented in Section 3.5.2 and also the discussion of the windowed sinc shown in Table 3.2 and Figure 3.29. Feel free to implement other filters (Wolberg 1990; Unser 1999).

Apply your filters to continuously resize an image, both magnifying (interpolating) and minifying (decimating) it; compare the resulting animations for several filters. Use both a synthetic chirp image (Figure 3.65a) and natural images with lots of high-frequency detail (Figure 3.65b-c).[27]

You may find it helpful to write a simple visualization program that continuously plays the animations for two or more filters at once and that let you "blink" between different results.

Discuss the merits and deficiencies of each filter, as well as its tradeoff between speed and quality.

Ex 3.19: Pyramids Construct an image pyramid. The inputs should be a grayscale or color image, a separable filter kernel, and the number of desired levels. Implement at least the following kernels:

[27] These particular images are available on the book's Web site.



(a) (b) (c)

Figure 3.65 Sample images for testing the quality of resampling algorithms: (a) a synthetic chirp; (b) and (c) some high-frequency images from the image compression community.

- 2×2 block filtering;
- Burt and Adelson's binomial kernel $^1/_{16}(1, 4, 6, 4, 1)$ (Burt and Adelson 1983a);
- a high-quality seven- or nine-tap filter.

Compare the visual quality of the various decimation filters. Also, shift your input image by 1 to 4 pixels and compare the resulting decimated (quarter size) image sequence.

Ex 3.20: Pyramid blending Write a program that takes as input two color images and a binary mask image and produces the Laplacian pyramid blend of the two images.

1. Construct the Laplacian pyramid for each image.
2. Construct the Gaussian pyramid for the two mask images (the input image and its complement).
3. Multiply each Laplacian image by its corresponding mask and sum the images (see Figure 3.43).
4. Reconstruct the final image from the blended Laplacian pyramid.

Generalize your algorithm to input n images and a label image with values $1 \ldots n$ (the value 0 can be reserved for "no input"). Discuss whether the weighted summation stage (step 3) needs to keep track of the total weight for renormalization, or whether the math just works out. Use your algorithm either to blend two differently exposed image (to avoid under- and over-exposed regions) or to make a creative blend of two different scenes.

Ex 3.21: Wavelet construction and applications Implement one of the wavelet families described in Section 3.5.4 or by Simoncelli and Adelson (1990b), as well as the basic Laplacian pyramid (Exercise 3.19). Apply the resulting representations to one of the following two tasks:

- **Compression:** Compute the entropy in each band for the different wavelet implementations, assuming a given quantization level (say, $1/4$ gray level, to keep the rounding error acceptable). Quantize the wavelet coefficients and reconstruct the original images. Which technique performs better? (See (Simoncelli and Adelson 1990b) or any of the multitude of wavelet compression papers for some typical results.)

- **Denoising.** After computing the wavelets, suppress small values using *coring*, i.e., set small values to zero using a piecewise linear or other C^0 function. Compare the results of your denoising using different wavelet and pyramid representations.

Ex 3.22: Parametric image warping Write the code to do affine and perspective image warps (optionally bilinear as well). Try a variety of interpolants and report on their visual quality. In particular, discuss the following:

- In a MIP-map, selecting only the coarser level adjacent to the computed fractional level will produce a blurrier image, while selecting the finer level will lead to aliasing. Explain why this is so and discuss whether blending an aliased and a blurred image (tri-linear MIP-mapping) is a good idea.

- When the ratio of the horizontal and vertical resampling rates becomes very different (anisotropic), the MIP-map performs even worse. Suggest some approaches to reduce such problems.

Ex 3.23: Local image warping Open an image and deform its appearance in one of the following ways:

1. Click on a number of pixels and move (drag) them to new locations. Interpolate the resulting sparse displacement field to obtain a dense motion field (Sections 3.6.2 and 3.5.1).

2. Draw a number of lines in the image. Move the endpoints of the lines to specify their new positions and use the Beier–Neely interpolation algorithm (Beier and Neely 1992), discussed in Section 3.6.2, to get a dense motion field.

3. Overlay a spline control grid and move one grid point at a time (optionally select the level of the deformation).

4. Have a dense per-pixel flow field and use a soft "paintbrush" to design a horizontal and vertical velocity field.

5. (Optional): Prove whether the Beier–Neely warp does or does not reduce to a sparse point-based deformation as the line segments become shorter (reduce to points).

Ex 3.24: Forward warping Given a displacement field from the previous exercise, write a forward warping algorithm:

1. Write a forward warper using splatting, either nearest neighbor or soft accumulation (Section 3.6.1).

2. Write a two-pass algorithm, which forward warps the displacement field, fills in small holes, and then uses inverse warping (Shade, Gortler, He *et al.* 1998).

3. Compare the quality of these two algorithms.

Ex 3.25: Feature-based morphing Extend the warping code you wrote in Exercise 3.23 to import two different images and specify correspondences (point, line, or mesh-based) between the two images.

1. Create a morph by partially warping the images towards each other and cross-dissolving (Section 3.6.3).

2. Try using your morphing algorithm to perform an image rotation and discuss whether it behaves the way you want it to.

Ex 3.26: 2D image editor Extend the program you wrote in Exercise 2.2 to import images and let you create a "collage" of pictures. You should implement the following steps:

1. Open up a new image (in a separate window).

2. Shift drag (rubber-band) to crop a subregion (or select whole image).

3. Paste into the current canvas.

4. Select the deformation mode (motion model): translation, rigid, similarity, affine, or perspective.

5. Drag any corner of the outline to change its transformation.

6. (Optional) Change the relative ordering of the images and which image is currently being manipulated.

The user should see the composition of the various images' pieces on top of each other.

This exercise should be built on the image transformation classes supported in the software library. Persistence of the created representation (save and load) should also be supported (for each image, save its transformation).

Ex 3.27: 3D texture-mapped viewer Extend the viewer you created in Exercise 2.3 to include texture-mapped polygon rendering. Augment each polygon with (u, v, w) coordinates into an image.

Ex 3.28: Image denoising Implement at least two of the various image denoising techniques described in this chapter and compare them on both synthetically noised image sequences and real-world (low-light) sequences. Does the performance of the algorithm depend on the correct choice of noise level estimate? Can you draw any conclusions as to which techniques work better?

Ex 3.29: Rainbow enhancer—challenging Take a picture containing a rainbow, such as Figure 3.66, and enhance the strength (saturation) of the rainbow.

1. Draw an arc in the image delineating the extent of the rainbow.



Figure 3.66 There is a faint image of a rainbow visible in the right hand side of this picture. Can you think of a way to enhance it (Exercise 3.29)?

2. Fit an *additive* rainbow function (explain why it is additive) to this arc (it is best to work with linearized pixel values), using the spectrum as the cross section, and estimating the width of the arc and the amount of color being added. This is the trickiest part of the problem, as you need to tease apart the (low-frequency) rainbow pattern and the natural image hiding behind it.

3. Amplify the rainbow signal and add it back into the image, re-applying the gamma function if necessary to produce the final image.

Ex 3.30: Image deblocking—challenging Now that you have some good techniques to distinguish signal from noise, develop a technique to remove the *blocking artifacts* that occur with JPEG at high compression settings (Section 2.3.3). Your technique can be as simple as looking for unexpected edges along block boundaries, to looking at the quantization step as a projection of a convex region of the transform coefficient space onto the corresponding quantized values.

1. Does the knowledge of the compression factor, which is available in the JPEG header information, help you perform better deblocking?

2. Because the quantization occurs in the DCT transformed YCbCr space (2.115), it may be preferable to perform the analysis in this space. On the other hand, image priors make more sense in an RGB space (or do they?). Decide how you will approach this dichotomy and discuss your choice.

3. While you are at it, since the YCbCr conversion is followed by a chrominance subsampling stage (before the DCT), see if you can restore some of the lost high-frequency chrominance signal using one of the better restoration techniques discussed in this chapter.

4. If your camera has a RAW + JPEG mode, how close can you come to the noise-free true pixel values? (This suggestion may not be that useful, since cameras generally use reasonably high quality settings for their RAW + JPEG models.)

Ex 3.31: Inference in de-blurring—challenging Write down the graphical model corresponding to Figure 3.59 for a non-blind image deblurring problem, i.e., one where the blur kernel is known ahead of time.

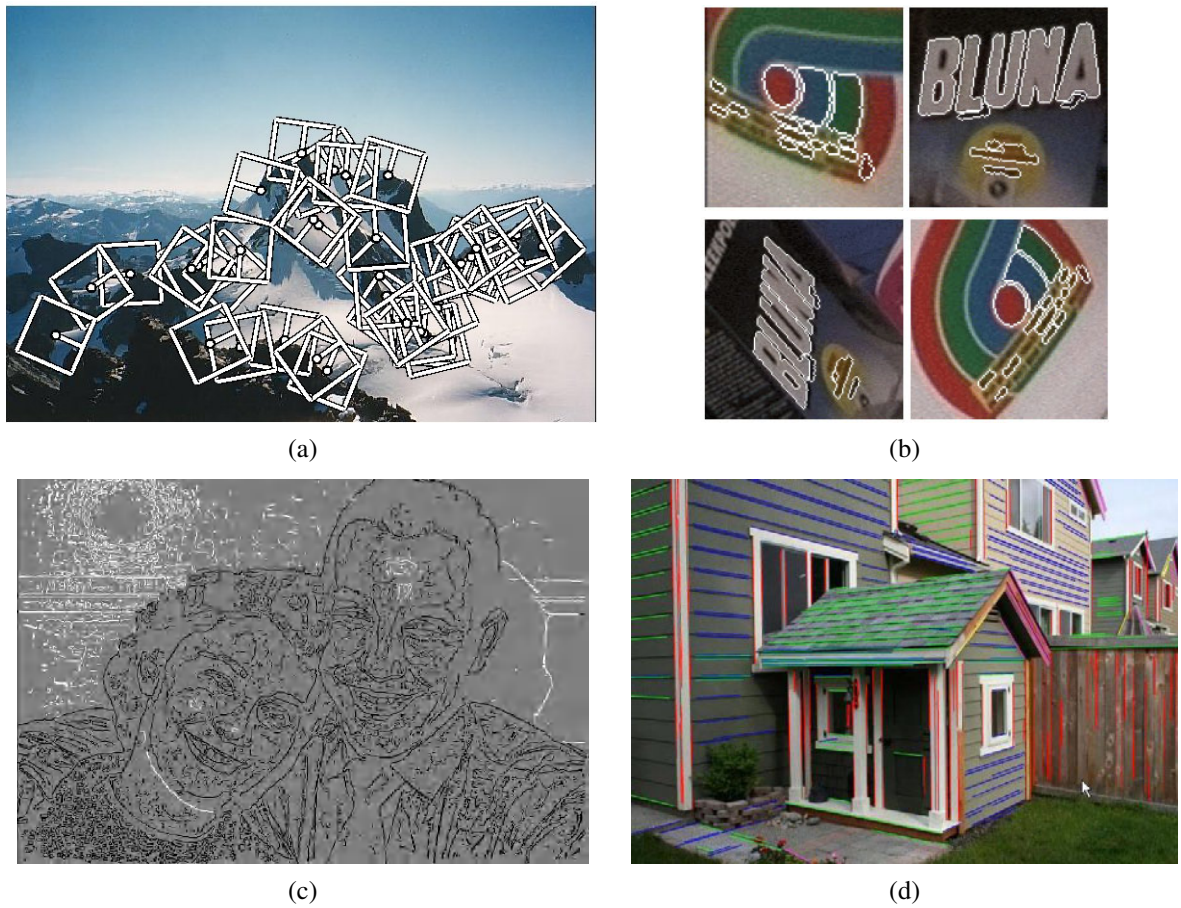
What kind of efficient inference (optimization) algorithms can you think of for solving such problems?

Chapter 4

Feature detection and matching

4.1 Points and patches . . . 183
4.1.1 Feature detectors . . . 185
4.1.2 Feature descriptors . . . 196
4.1.3 Feature matching . . . 200
4.1.4 Feature tracking . . . 207
4.1.5 *Application*: Performance-driven animation . . . 209
4.2 Edges . . . 210
4.2.1 Edge detection . . . 210
4.2.2 Edge linking . . . 215
4.2.3 *Application*: Edge editing and enhancement . . . 219
4.3 Lines . . . 220
4.3.1 Successive approximation . . . 220
4.3.2 Hough transforms . . . 221
4.3.3 Vanishing points . . . 224
4.3.4 *Application*: Rectangle detection . . . 226
4.4 Additional reading . . . 227
4.5 Exercises . . . 228

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_4, © Springer-Verlag London Limited 2011



(a) (b)

(c) (d)

Figure 4.1 A variety of feature detectors and descriptors can be used to analyze, describe and match images: (a) point-like interest operators (Brown, Szeliski, and Winder 2005) © 2005 IEEE; (b) region-like interest operators (Matas, Chum, Urban *et al.* 2004) © 2004 Elsevier; (c) edges (Elder and Goldberg 2001) © 2001 IEEE; (d) straight lines (Sinha, Steedly, Szeliski *et al.* 2008) © 2008 ACM.

Feature detection and matching are an essential component of many computer vision applications. Consider the two pairs of images shown in Figure 4.2. For the first pair, we may wish to *align* the two images so that they can be seamlessly stitched into a composite mosaic (Chapter 9). For the second pair, we may wish to establish a dense set of *correspondences* so that a 3D model can be constructed or an in-between view can be generated (Chapter 11). In either case, what kinds of *features* should you detect and then match in order to establish such an alignment or set of correspondences? Think about this for a few moments before reading on.

The first kind of feature that you may notice are specific locations in the images, such as mountain peaks, building corners, doorways, or interestingly shaped patches of snow. These kinds of localized feature are often called *keypoint features* or *interest points* (or even *corners*) and are often described by the appearance of patches of pixels surrounding the point location (Section 4.1). Another class of important features are *edges*, e.g., the profile of mountains against the sky, (Section 4.2). These kinds of features can be matched based on their orientation and local appearance (edge profiles) and can also be good indicators of object boundaries and *occlusion* events in image sequences. Edges can be grouped into longer *curves* and *straight line segments*, which can be directly matched or analyzed to find *vanishing points* and hence internal and external camera parameters (Section 4.3).

In this chapter, we describe some practical approaches to detecting such features and also discuss how feature correspondences can be established across different images. Point features are now used in such a wide variety of applications that it is good practice to read and implement some of the algorithms from (Section 4.1). Edges and lines provide information that is complementary to both keypoint and region-based descriptors and are well-suited to describing object boundaries and man-made objects. These alternative descriptors, while extremely useful, can be skipped in a short introductory course.

4.1 Points and patches

Point features can be used to find a sparse set of corresponding locations in different images, often as a pre-cursor to computing camera pose (Chapter 7), which is a prerequisite for computing a denser set of correspondences using stereo matching (Chapter 11). Such correspondences can also be used to align different images, e.g., when stitching image mosaics or performing video stabilization (Chapter 9). They are also used extensively to perform object instance and category recognition (Sections 14.3 and 14.4). A key advantage of keypoints is that they permit matching even in the presence of clutter (occlusion) and large scale and orientation changes.

Feature-based correspondence techniques have been used since the early days of stereo matching (Hannah 1974; Moravec 1983; Hannah 1988) and have more recently gained popularity for image-stitching applications (Zoghlami, Faugeras, and Deriche 1997; Brown and Lowe 2007) as well as fully automated 3D modeling (Beardsley, Torr, and Zisserman 1996; Schaffalitzky and Zisserman 2002; Brown and Lowe 2003; Snavely, Seitz, and Szeliski 2006).

There are two main approaches to finding feature points and their correspondences. The first is to find features in one image that can be accurately *tracked* using a local search technique, such as correlation or least squares (Section 4.1.4). The second is to independently



Figure 4.2 Two pairs of images to be matched. What kinds of feature might one use to establish a set of *correspondences* between these images?

detect features in all the images under consideration and then *match* features based on their local appearance (Section 4.1.3). The former approach is more suitable when images are taken from nearby viewpoints or in rapid succession (e.g., video sequences), while the latter is more suitable when a large amount of motion or appearance change is expected, e.g., in stitching together panoramas (Brown and Lowe 2007), establishing correspondences in *wide baseline stereo* (Schaffalitzky and Zisserman 2002), or performing object recognition (Fergus, Perona, and Zisserman 2007).

In this section, we split the keypoint detection and matching pipeline into four separate stages. During the *feature detection* (extraction) stage (Section 4.1.1), each image is searched for locations that are likely to match well in other images. At the *feature description* stage (Section 4.1.2), each region around detected keypoint locations is converted into a more compact and stable (invariant) *descriptor* that can be matched against other descriptors. The *feature matching* stage (Section 4.1.3) efficiently searches for likely matching candidates in other images. The *feature tracking* stage (Section 4.1.4) is an alternative to the third stage that only searches a small neighborhood around each detected feature and is therefore more suitable for video processing.

A wonderful example of all of these stages can be found in David Lowe's (2004) paper, which describes the development and refinement of his *Scale Invariant Feature Transform* (SIFT). Comprehensive descriptions of alternative techniques can be found in a series of survey and evaluation papers covering both feature detection (Schmid, Mohr, and Bauckhage 2000; Mikolajczyk, Tuytelaars, Schmid *et al.* 2005; Tuytelaars and Mikolajczyk 2007) and feature descriptors (Mikolajczyk and Schmid 2005). Shi and Tomasi (1994) and Triggs (2004) also provide nice reviews of feature detection techniques.

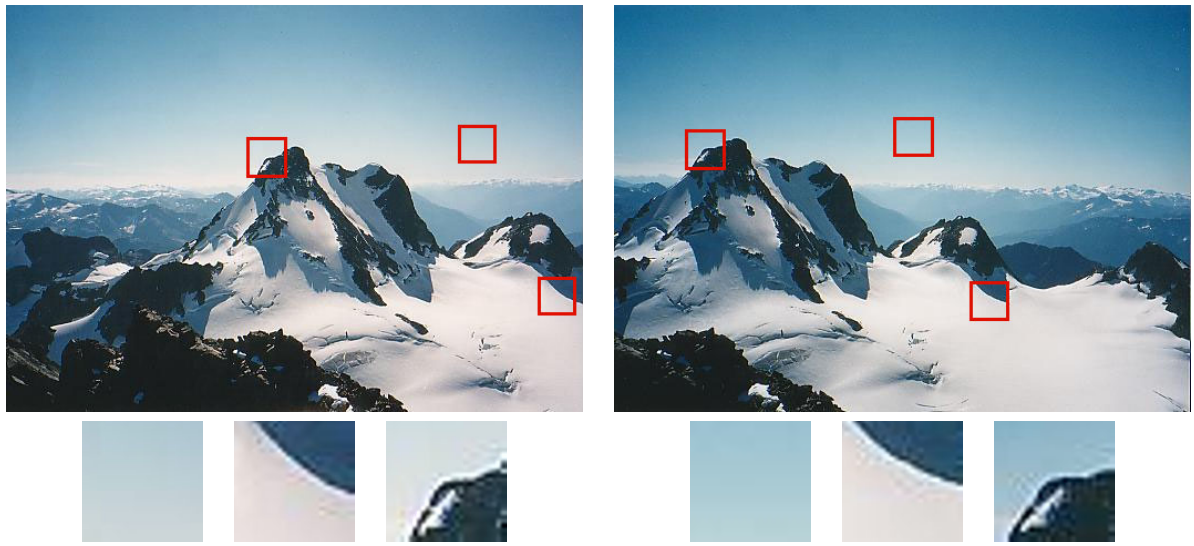


Figure 4.3 Image pairs with extracted patches below. Notice how some patches can be localized or matched with higher accuracy than others.

4.1.1 Feature detectors

How can we find image locations where we can reliably find correspondences with other images, i.e., what are good features to track (Shi and Tomasi 1994; Triggs 2004)? Look again at the image pair shown in Figure 4.3 and at the three sample *patches* to see how well they might be matched or tracked. As you may notice, textureless patches are nearly impossible to localize. Patches with large contrast changes (gradients) are easier to localize, although straight line segments at a single orientation suffer from the *aperture problem* (Horn and Schunck 1981; Lucas and Kanade 1981; Anandan 1989), i.e., it is only possible to align the patches along the direction *normal* to the edge direction (Figure 4.4b). Patches with gradients in at least two (significantly) different orientations are the easiest to localize, as shown schematically in Figure 4.4a.

These intuitions can be formalized by looking at the simplest possible matching criterion for comparing two image patches, i.e., their (weighted) summed square difference,

$$E_{\mathrm{WSSD}}(\boldsymbol{u}) = \sum_i w(\boldsymbol{x}_i)[I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2, \tag{4.1}$$

where I_0 and I_1 are the two images being compared, $\boldsymbol{u} = (u, v)$ is the *displacement* vector, $w(\boldsymbol{x})$ is a spatially varying weighting (or window) function, and the summation i is over all the pixels in the patch. Note that this is the same formulation we later use to estimate motion between complete images (Section 8.1).

When performing feature detection, we do not know which other image locations the feature will end up being matched against. Therefore, we can only compute how stable this metric is with respect to small variations in position $\Delta\boldsymbol{u}$ by comparing an image patch against

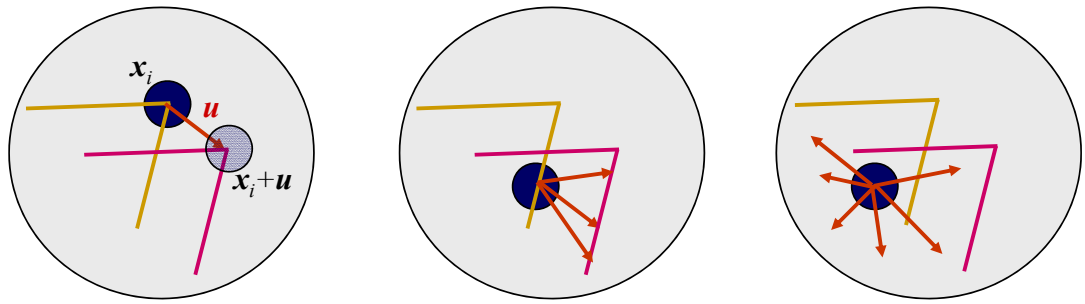


Figure 4.4 Aperture problems for different image patches: (a) stable ("corner-like") flow; (b) classic aperture problem (barber-pole illusion); (c) textureless region. The two images I_0 (yellow) and I_1 (red) are overlaid. The red vector $\boldsymbol{u}$ indicates the displacement between the patch centers and the $w(\boldsymbol{x}_i)$ weighting function (patch window) is shown as a dark circle.

itself, which is known as an *auto-correlation function* or *surface*

$$E_{\mathrm{AC}}(\Delta\boldsymbol{u}) = \sum_i w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i + \Delta\boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2 \tag{4.2}$$

(Figure 4.5).[1] Note how the auto-correlation surface for the textured flower bed (Figure 4.5b and the red cross in the lower right quadrant of Figure 4.5a) exhibits a strong minimum, indicating that it can be well localized. The correlation surface corresponding to the roof edge (Figure 4.5c) has a strong ambiguity along one direction, while the correlation surface corresponding to the cloud region (Figure 4.5d) has no stable minimum.

Using a Taylor Series expansion of the image function $I_0(\boldsymbol{x}_i+\Delta\boldsymbol{u}) \approx I_0(\boldsymbol{x}_i)+\nabla I_0(\boldsymbol{x}_i)\cdot\Delta\boldsymbol{u}$ (Lucas and Kanade 1981; Shi and Tomasi 1994), we can approximate the auto-correlation surface as

$$\begin{aligned} E_{\mathrm{AC}}(\Delta\boldsymbol{u}) &= \sum_i w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i + \Delta\boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2 && (4.3)\\ &\approx \sum_i w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i) + \nabla I_0(\boldsymbol{x}_i)\cdot\Delta\boldsymbol{u} - I_0(\boldsymbol{x}_i)]^2 && (4.4)\\ &= \sum_i w(\boldsymbol{x}_i)[\nabla I_0(\boldsymbol{x}_i)\cdot\Delta\boldsymbol{u}]^2 && (4.5)\\ &= \Delta\boldsymbol{u}^T\boldsymbol{A}\Delta\boldsymbol{u}, && (4.6) \end{aligned}$$

where

$$\nabla I_0(\boldsymbol{x}_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y}\right)(\boldsymbol{x}_i) \tag{4.7}$$

is the *image gradient* at $\boldsymbol{x}_i$. This gradient can be computed using a variety of techniques (Schmid, Mohr, and Bauckhage 2000). The classic "Harris" detector (Harris and Stephens 1988) uses a [-2 -1 0 1 2] filter, but more modern variants (Schmid, Mohr, and Bauckhage 2000; Triggs 2004) convolve the image with horizontal and vertical derivatives of a Gaussian (typically with $\sigma = 1$).

[1] Strictly speaking, a correlation is the *product* of two patches (3.12); I'm using the term here in a more qualitative sense. The weighted sum of squared differences is often called an *SSD surface* (Section 8.1).

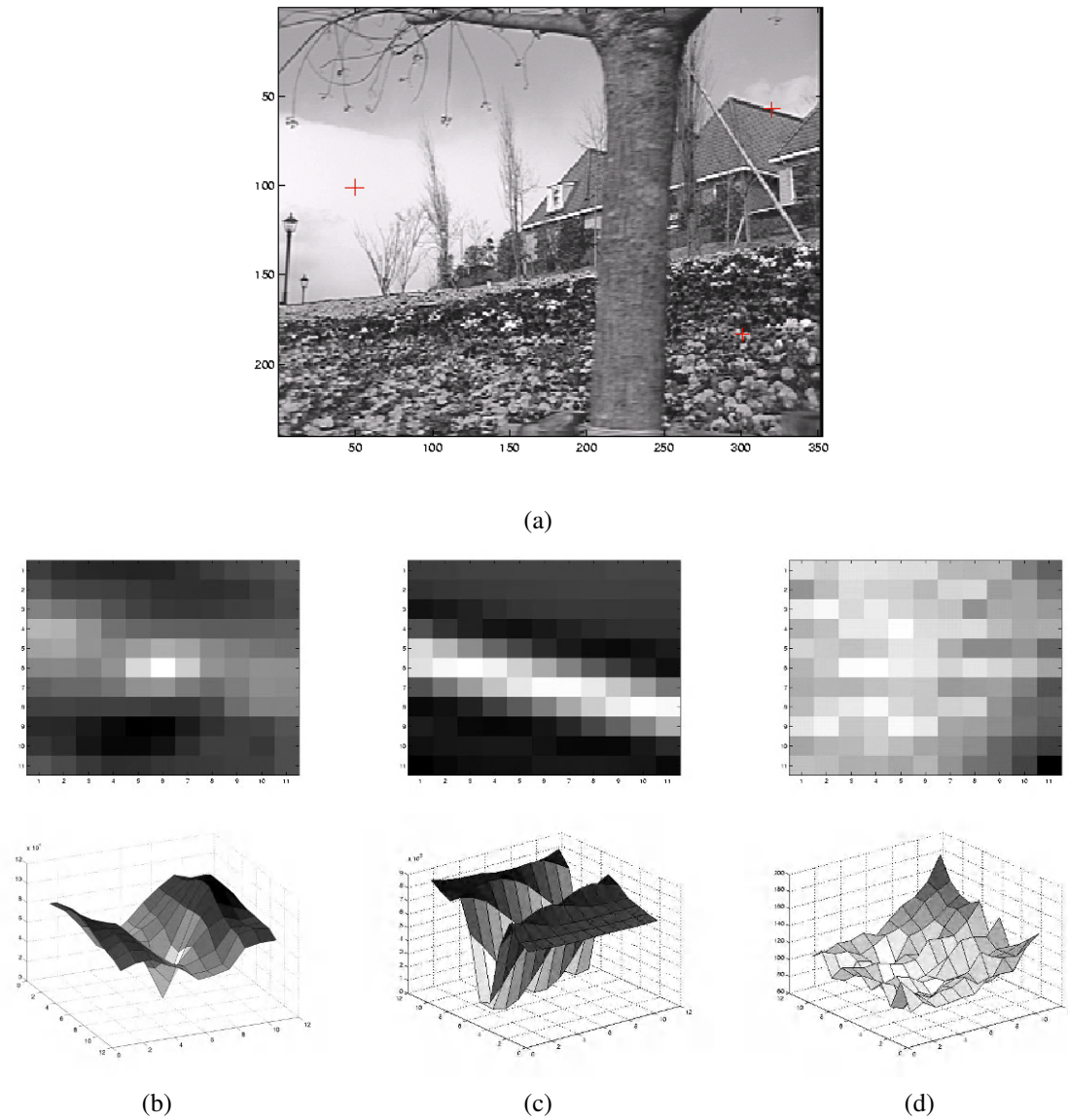


Figure 4.5 Three auto-correlation surfaces $E_{\mathrm{AC}}(\Delta \boldsymbol{u})$ shown as both grayscale images and surface plots: (a) The original image is marked with three red crosses to denote where the auto-correlation surfaces were computed; (b) this patch is from the flower bed (good unique minimum); (c) this patch is from the roof edge (one-dimensional aperture problem); and (d) this patch is from the cloud (no good peak). Each grid point in figures b–d is one value of $\Delta \boldsymbol{u}$.

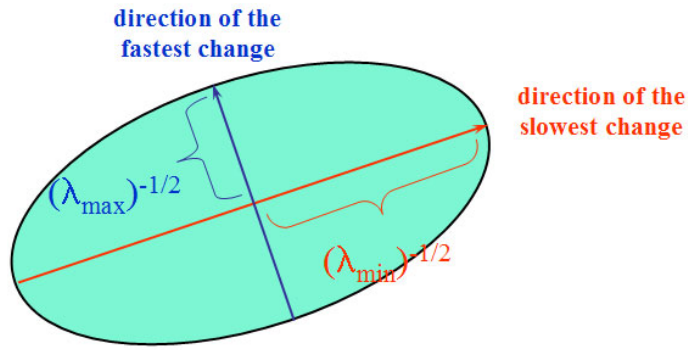


Figure 4.6 Uncertainty ellipse corresponding to an eigenvalue analysis of the auto-correlation matrix $\boldsymbol{A}$.

The auto-correlation matrix $\boldsymbol{A}$ can be written as

$$\boldsymbol{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \tag{4.8}$$

where we have replaced the weighted summations with discrete convolutions with the weighting kernel w. This matrix can be interpreted as a tensor (multiband) image, where the outer products of the gradients ∇I are convolved with a weighting function w to provide a per-pixel estimate of the local (quadratic) shape of the auto-correlation function.

As first shown by Anandan (1984; 1989) and further discussed in Section 8.1.3 and (8.44), the inverse of the matrix $\boldsymbol{A}$ provides a lower bound on the uncertainty in the location of a matching patch. It is therefore a useful indicator of which patches can be reliably matched. The easiest way to visualize and reason about this uncertainty is to perform an eigenvalue analysis of the auto-correlation matrix $\boldsymbol{A}$, which produces two eigenvalues (λ_0, λ_1) and two eigenvector directions (Figure 4.6). Since the larger uncertainty depends on the smaller eigenvalue, i.e., $\lambda_0^{-1/2}$, it makes sense to find maxima in the smaller eigenvalue to locate good features to track (Shi and Tomasi 1994).

Förstner–Harris. While Anandan and Lucas and Kanade (1981) were the first to analyze the uncertainty structure of the auto-correlation matrix, they did so in the context of associating certainties with optic flow measurements. Förstner (1986) and Harris and Stephens (1988) were the first to propose using local maxima in rotationally invariant scalar measures derived from the auto-correlation matrix to locate keypoints for the purpose of sparse feature matching. (Schmid, Mohr, and Bauckhage (2000); Triggs (2004) give more detailed historical reviews of feature detection algorithms.) Both of these techniques also proposed using a Gaussian weighting window instead of the previously used square patches, which makes the detector response insensitive to in-plane image rotations.

The minimum eigenvalue λ_0 (Shi and Tomasi 1994) is not the only quantity that can be used to find keypoints. A simpler quantity, proposed by Harris and Stephens (1988), is

$$\det(\boldsymbol{A}) - \alpha \operatorname{trace}(\boldsymbol{A})^2 = \lambda_0 \lambda_1 - \alpha(\lambda_0 + \lambda_1)^2 \tag{4.9}$$

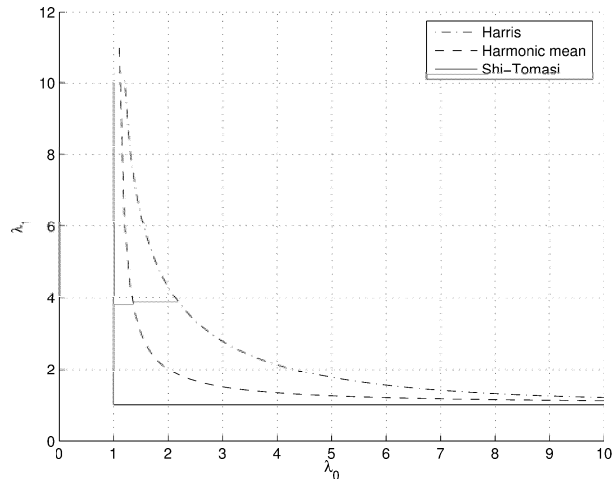


Figure 4.7 Isocontours of popular keypoint detection functions (Brown, Szeliski, and Winder 2004). Each detector looks for points where the eigenvalues λ_0, λ_1 of $\boldsymbol{A} = w * \nabla I \nabla I^T$ are both large.

with $\alpha = 0.06$. Unlike eigenvalue analysis, this quantity does not require the use of square roots and yet is still rotationally invariant and also downweights edge-like features where $\lambda_1 \gg \lambda_0$. Triggs (2004) suggests using the quantity

$$\lambda_0 - \alpha\lambda_1 \tag{4.10}$$

(say, with $\alpha = 0.05$), which also reduces the response at 1D edges, where aliasing errors sometimes inflate the smaller eigenvalue. He also shows how the basic 2×2 Hessian can be extended to parametric motions to detect points that are also accurately localizable in scale and rotation. Brown, Szeliski, and Winder (2005), on the other hand, use the harmonic mean,

$$\frac{\det \boldsymbol{A}}{\operatorname{tr} \boldsymbol{A}} = \frac{\lambda_0\lambda_1}{\lambda_0 + \lambda_1}, \tag{4.11}$$

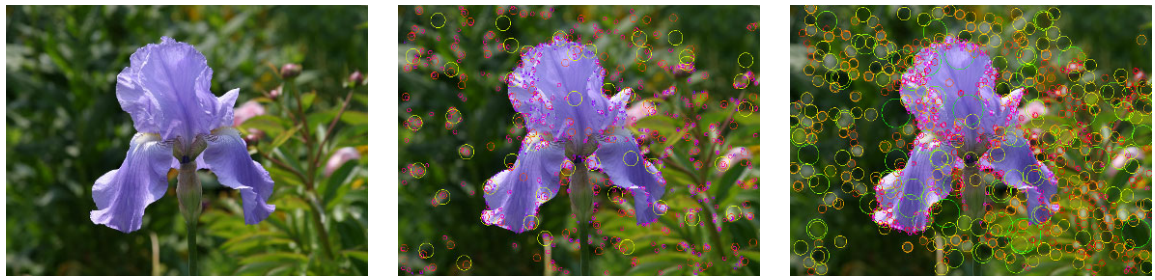
which is a smoother function in the region where $\lambda_0 \approx \lambda_1$. Figure 4.7 shows isocontours of the various interest point operators, from which we can see how the two eigenvalues are blended to determine the final interest value.

The steps in the basic auto-correlation-based keypoint detector are summarized in Algorithm 4.1. Figure 4.8 shows the resulting interest operator responses for the classic Harris detector as well as the difference of Gaussian (DoG) detector discussed below.

Adaptive non-maximal suppression (ANMS). While most feature detectors simply look for local maxima in the interest function, this can lead to an uneven distribution of feature points across the image, e.g., points will be denser in regions of higher contrast. To mitigate this problem, Brown, Szeliski, and Winder (2005) only detect features that are both local maxima and whose response value is significantly (10%) greater than that of all of its neighbors within a radius r (Figure 4.9c–d). They devise an efficient way to associate suppression radii with all local maxima by first sorting them by their response strength and then creating a second list sorted by decreasing suppression radius (Brown, Szeliski, and

1. Compute the horizontal and vertical derivatives of the image I_x and I_y by convolving the original image with derivatives of Gaussians (Section 3.2.3).
2. Compute the three images corresponding to the outer products of these gradients. (The matrix $\boldsymbol{A}$ is symmetric, so only three entries are needed.)
3. Convolve each of these images with a larger Gaussian.
4. Compute a scalar interest measure using one of the formulas discussed above.
5. Find local maxima above a certain threshold and report them as detected feature point locations.

Algorithm 4.1 Outline of a basic feature detection algorithm.

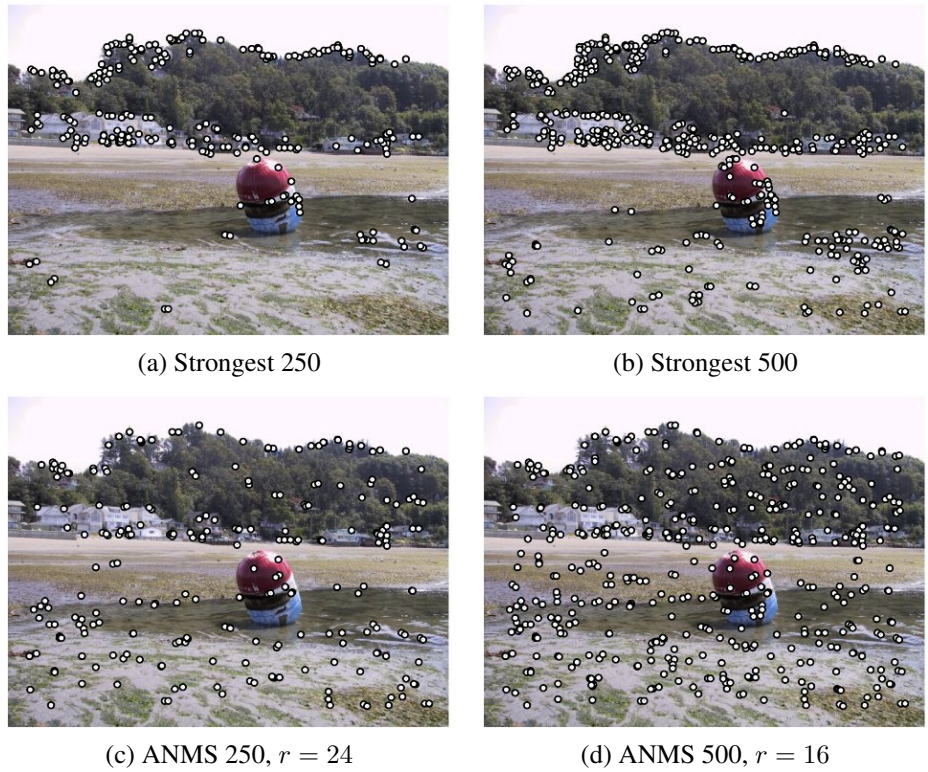


(a) (b) (c)

Figure 4.8 Interest operator responses: (a) Sample image, (b) Harris response, and (c) DoG response. The circle sizes and colors indicate the scale at which each interest point was detected. Notice how the two detectors tend to respond at complementary locations.

Winder 2005). Figure 4.9 shows a qualitative comparison of selecting the top n features and using ANMS.

Measuring repeatability. Given the large number of feature detectors that have been developed in computer vision, how can we decide which ones to use? Schmid, Mohr, and Bauckhage (2000) were the first to propose measuring the *repeatability* of feature detectors, which they define as the frequency with which keypoints detected in one image are found within ϵ (say, $\epsilon = 1.5$) pixels of the corresponding location in a transformed image. In their paper, they transform their planar images by applying rotations, scale changes, illumination changes, viewpoint changes, and adding noise. They also measure the *information content* available at each detected feature point, which they define as the entropy of a set of rotationally invariant local grayscale descriptors. Among the techniques they survey, they find that the improved (Gaussian derivative) version of the Harris operator with $\sigma_d = 1$ (scale of the derivative Gaussian) and $\sigma_i = 2$ (scale of the integration Gaussian) works best.



(a) Strongest 250 (b) Strongest 500

(c) ANMS 250, $r = 24$ (d) ANMS 500, $r = 16$

Figure 4.9 Adaptive non-maximal suppression (ANMS) (Brown, Szeliski, and Winder 2005) © 2005 IEEE: The upper two images show the strongest 250 and 500 interest points, while the lower two images show the interest points selected with adaptive non-maximal suppression, along with the corresponding suppression radius r. Note how the latter features have a much more uniform spatial distribution across the image.

Scale invariance

In many situations, detecting features at the finest stable scale possible may not be appropriate. For example, when matching images with little high frequency detail (e.g., clouds), fine-scale features may not exist.

One solution to the problem is to extract features at a variety of scales, e.g., by performing the same operations at multiple resolutions in a pyramid and then matching features at the same level. This kind of approach is suitable when the images being matched do not undergo large scale changes, e.g., when matching successive aerial images taken from an airplane or stitching panoramas taken with a fixed-focal-length camera. Figure 4.10 shows the output of one such approach, the multi-scale, oriented patch detector of Brown, Szeliski, and Winder (2005), for which responses at five different scales are shown.

However, for most object recognition applications, the scale of the object in the image is unknown. Instead of extracting features at many different scales and then matching all of them, it is more efficient to extract features that are stable in both location *and* scale (Lowe 2004; Mikolajczyk and Schmid 2004).

Early investigations into scale selection were performed by Lindeberg (1993; 1998b), who first proposed using extrema in the Laplacian of Gaussian (LoG) function as interest

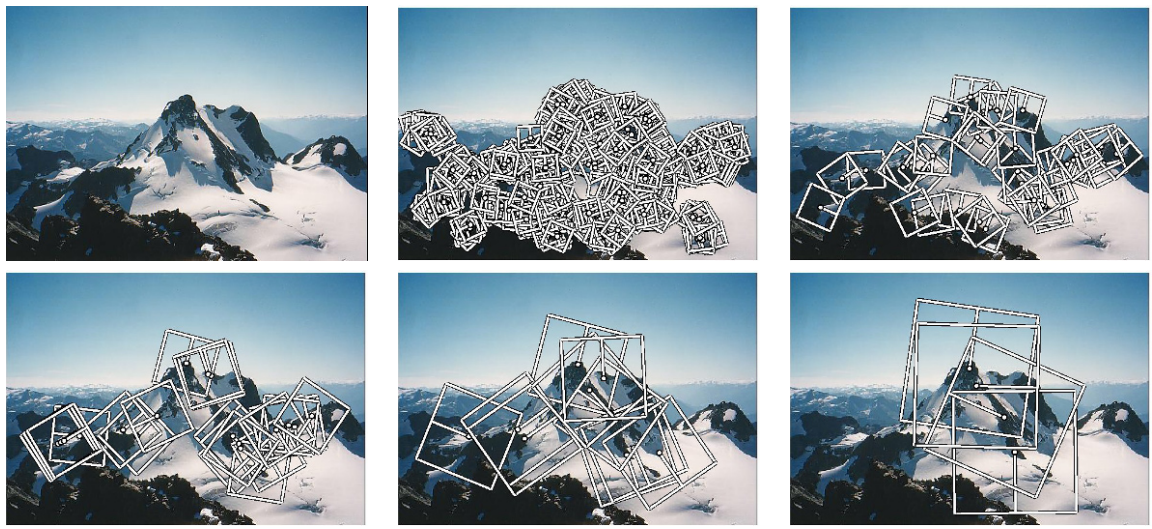


Figure 4.10 Multi-scale oriented patches (MOPS) extracted at five pyramid levels (Brown, Szeliski, and Winder 2005) © 2005 IEEE. The boxes show the feature orientation and the region from which the descriptor vectors are sampled.

point locations. Based on this work, Lowe (2004) proposed computing a set of sub-octave Difference of Gaussian filters (Figure 4.11a), looking for 3D (space+scale) maxima in the resulting structure (Figure 4.11b), and then computing a sub-pixel space+scale location using a quadratic fit (Brown and Lowe 2002). The number of sub-octave levels was determined, after careful empirical investigation, to be three, which corresponds to a quarter-octave pyramid, which is the same as used by Triggs (2004).

As with the Harris operator, pixels where there is strong asymmetry in the local curvature of the indicator function (in this case, the DoG) are rejected. This is implemented by first computing the local Hessian of the difference image D,

$$\boldsymbol{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}, \tag{4.12}$$

and then rejecting keypoints for which

$$\frac{\mathrm{Tr}(\boldsymbol{H})^2}{\mathrm{Det}(\boldsymbol{H})} > 10. \tag{4.13}$$

While Lowe's Scale Invariant Feature Transform (SIFT) performs well in practice, it is not based on the same theoretical foundation of maximum spatial stability as the auto-correlation-based detectors. (In fact, its detection locations are often complementary to those produced by such techniques and can therefore be used in conjunction with these other approaches.) In order to add a scale selection mechanism to the Harris corner detector, Mikolajczyk and Schmid (2004) evaluate the Laplacian of Gaussian function at each detected Harris point (in a multi-scale pyramid) and keep only those points for which the Laplacian is extremal (larger or smaller than both its coarser and finer-level values). An optional iterative refinement for both scale and position is also proposed and evaluated. Additional examples of scale invariant

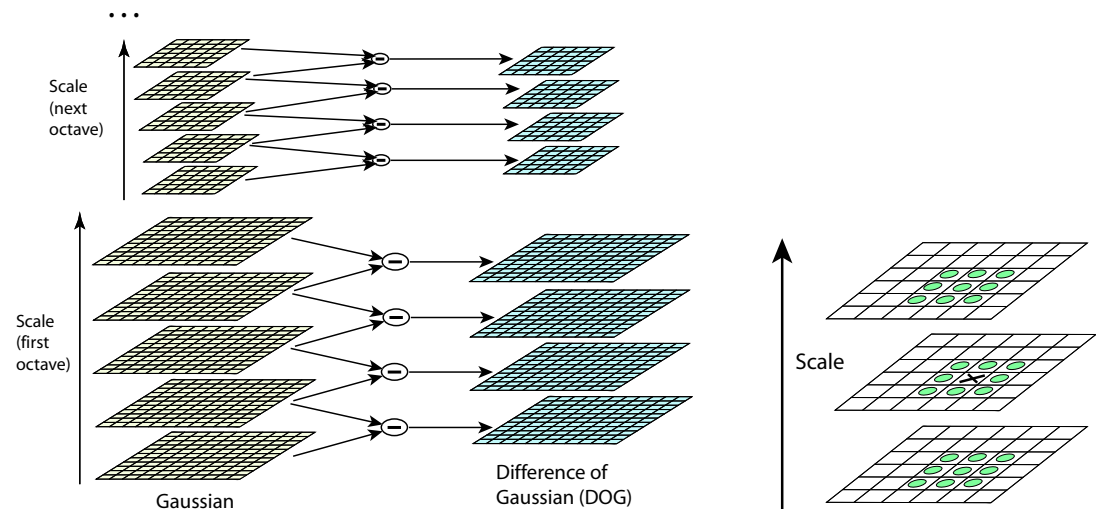


Figure 4.11 Scale-space feature detection using a sub-octave Difference of Gaussian pyramid (Lowe 2004) © 2004 Springer: (a) Adjacent levels of a sub-octave Gaussian pyramid are subtracted to produce Difference of Gaussian images; (b) extrema (maxima and minima) in the resulting 3D volume are detected by comparing a pixel to its 26 neighbors.

region detectors are discussed by Mikolajczyk, Tuytelaars, Schmid *et al.* (2005); Tuytelaars and Mikolajczyk (2007).

Rotational invariance and orientation estimation

In addition to dealing with scale changes, most image matching and object recognition algorithms need to deal with (at least) in-plane image rotation. One way to deal with this problem is to design descriptors that are rotationally invariant (Schmid and Mohr 1997), but such descriptors have poor discriminability, i.e. they map different looking patches to the same descriptor.

A better method is to estimate a *dominant orientation* at each detected keypoint. Once the local orientation and scale of a keypoint have been estimated, a scaled and oriented patch around the detected point can be extracted and used to form a feature descriptor (Figures 4.10 and 4.17).

The simplest possible orientation estimate is the average gradient within a region around the keypoint. If a Gaussian weighting function is used (Brown, Szeliski, and Winder 2005), this average gradient is equivalent to a first-order steerable filter (Section 3.2.3), i.e., it can be computed using an image convolution with the horizontal and vertical derivatives of Gaussian filter (Freeman and Adelson 1991). In order to make this estimate more reliable, it is usually preferable to use a larger aggregation window (Gaussian kernel size) than detection window (Brown, Szeliski, and Winder 2005). The orientations of the square boxes shown in Figure 4.10 were computed using this technique.

Sometimes, however, the averaged (signed) gradient in a region can be small and therefore

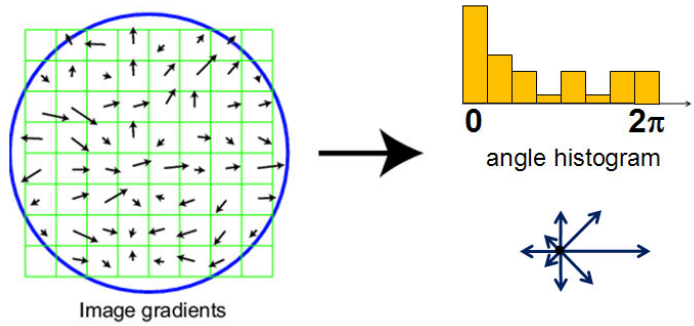


Figure 4.12 A dominant orientation estimate can be computed by creating a histogram of all the gradient orientations (weighted by their magnitudes or after thresholding out small gradients) and then finding the significant peaks in this distribution (Lowe 2004) © 2004 Springer.

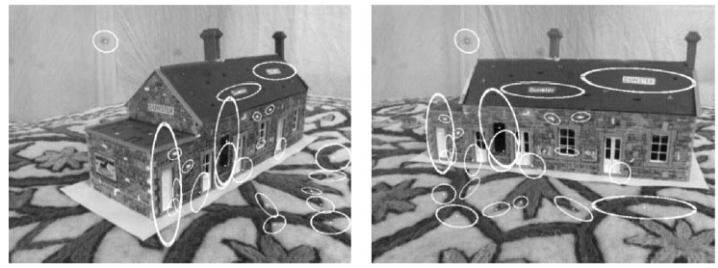


Figure 4.13 Affine region detectors used to match two images taken from dramatically different viewpoints (Mikolajczyk and Schmid 2004) © 2004 Springer.

an unreliable indicator of orientation. A more reliable technique is to look at the *histogram* of orientations computed around the keypoint. Lowe (2004) computes a 36-bin histogram of edge orientations weighted by both gradient magnitude and Gaussian distance to the center, finds all peaks within 80% of the global maximum, and then computes a more accurate orientation estimate using a three-bin parabolic fit (Figure 4.12).

Affine invariance

While scale and rotation invariance are highly desirable, for many applications such as *wide baseline stereo matching* (Pritchett and Zisserman 1998; Schaffalitzky and Zisserman 2002) or location recognition (Chum, Philbin, Sivic *et al.* 2007), full affine invariance is preferred. Affine-invariant detectors not only respond at consistent locations after scale and orientation changes, they also respond consistently across affine deformations such as (local) perspective foreshortening (Figure 4.13). In fact, for a small enough patch, any continuous image warping can be well approximated by an affine deformation.

To introduce affine invariance, several authors have proposed fitting an ellipse to the auto-correlation or Hessian matrix (using eigenvalue analysis) and then using the principal axes and ratios of this fit as the affine coordinate frame (Lindeberg and Garding 1997; Baumberg

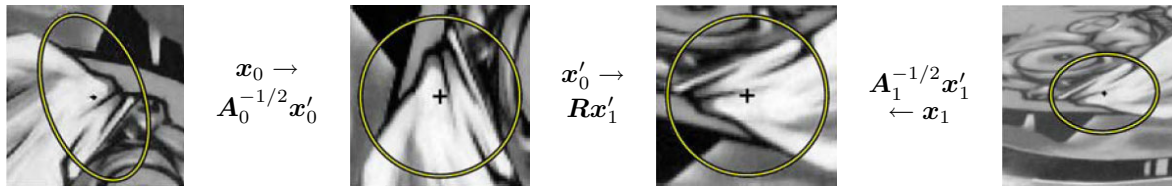


Figure 4.14 Affine normalization using the second moment matrices, as described by Mikolajczyk, Tuytelaars, Schmid *et al.* (2005) © 2005 Springer. After image coordinates are transformed using the matrices $A_0^{-1/2}$ and $A_1^{-1/2}$, they are related by a pure rotation R, which can be estimated using a dominant orientation technique.



Figure 4.15 Maximally stable extremal regions (MSERs) extracted and matched from a number of images (Matas, Chum, Urban *et al.* 2004) © 2004 Elsevier.

2000; Mikolajczyk and Schmid 2004; Mikolajczyk, Tuytelaars, Schmid *et al.* 2005; Tuytelaars and Mikolajczyk 2007). Figure 4.14 shows how the square root of the moment matrix can be used to transform local patches into a frame which is similar up to rotation.

Another important affine invariant region detector is the maximally stable extremal region (MSER) detector developed by Matas, Chum, Urban *et al.* (2004). To detect MSERs, binary regions are computed by thresholding the image at all possible gray levels (the technique therefore only works for grayscale images). This operation can be performed efficiently by first sorting all pixels by gray value and then incrementally adding pixels to each connected component as the threshold is changed (Nistér and Stewénius 2008). As the threshold is changed, the area of each component (region) is monitored; regions whose rate of change of area with respect to the threshold is minimal are defined as *maximally stable*. Such regions are therefore invariant to both affine geometric and photometric (linear bias-gain or smooth monotonic) transformations (Figure 4.15). If desired, an affine coordinate frame can be fit to each detected region using its moment matrix.

The area of feature point detectors continues to be very active, with papers appearing every year at major computer vision conferences (Xiao and Shah 2003; Koethe 2003; Carneiro and Jepson 2005; Kenney, Zuliani, and Manjunath 2005; Bay, Tuytelaars, and Van Gool 2006; Platel, Balmachnova, Florack *et al.* 2006; Rosten and Drummond 2006). Mikolajczyk, Tuytelaars, Schmid *et al.* (2005) survey a number of popular affine region detectors and provide experimental comparisons of their invariance to common image transformations such as scaling, rotations, noise, and blur. These experimental results, code, and pointers to the surveyed papers can be found on their Web site at http://www.robots.ox.ac.uk/~vgg/research/affine/.

Of course, keypoints are not the only features that can be used for registering images. Zoghlami, Faugeras, and Deriche (1997) use line segments as well as point-like features to estimate homographies between pairs of images, whereas Bartoli, Coquerelle, and Sturm (2004) use line segments with local correspondences along the edges to extract 3D structure

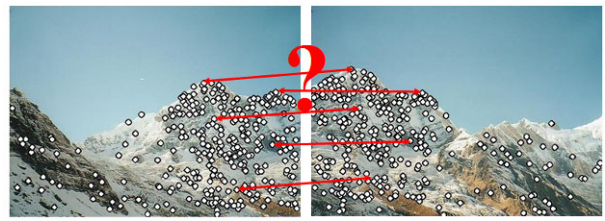


Figure 4.16 Feature matching: how can we extract local descriptors that are invariant to inter-image variations and yet still discriminative enough to establish correct correspondences?

and motion. Tuytelaars and Van Gool (2004) use affine invariant regions to detect correspondences for wide baseline stereo matching, whereas Kadir, Zisserman, and Brady (2004) detect salient regions where patch entropy and its rate of change with scale are locally maximal. Corso and Hager (2005) use a related technique to fit 2D oriented Gaussian kernels to homogeneous regions. More details on techniques for finding and matching curves, lines, and regions can be found later in this chapter.

4.1.2 Feature descriptors

After detecting features (keypoints), we must *match* them, i.e., we must determine which features come from corresponding locations in different images. In some situations, e.g., for video sequences (Shi and Tomasi 1994) or for stereo pairs that have been *rectified* (Zhang, Deriche, Faugeras *et al.* 1995; Loop and Zhang 1999; Scharstein and Szeliski 2002), the local motion around each feature point may be mostly translational. In this case, simple error metrics, such as the *sum of squared differences* or *normalized cross-correlation*, described in Section 8.1 can be used to directly compare the intensities in small patches around each feature point. (The comparative study by Mikolajczyk and Schmid (2005), discussed below, uses cross-correlation.) Because feature points may not be exactly located, a more accurate matching score can be computed by performing incremental motion refinement as described in Section 8.1.3 but this can be time consuming and can sometimes even decrease performance (Brown, Szeliski, and Winder 2005).

In most cases, however, the local appearance of features will change in orientation and scale, and sometimes even undergo affine deformations. Extracting a local scale, orientation, or affine frame estimate and then using this to resample the patch before forming the feature descriptor is thus usually preferable (Figure 4.17).

Even after compensating for these changes, the local appearance of image patches will usually still vary from image to image. How can we make image descriptors more invariant to such changes, while still preserving discriminability between different (non-corresponding) patches (Figure 4.16)? Mikolajczyk and Schmid (2005) review some recently developed view-invariant local image descriptors and experimentally compare their performance. Below, we describe a few of these descriptors in more detail.

Bias and gain normalization (MOPS). For tasks that do not exhibit large amounts of foreshortening, such as image stitching, simple normalized intensity patches perform reasonably well and are simple to implement (Brown, Szeliski, and Winder 2005) (Figure 4.17). In

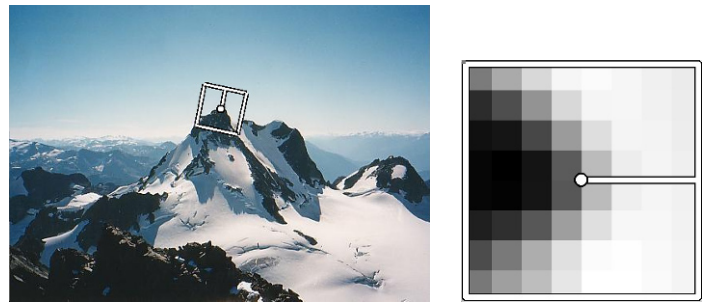


Figure 4.17 MOPS descriptors are formed using an 8×8 sampling of bias and gain normalized intensity values, with a sample spacing of five pixels relative to the detection scale (Brown, Szeliski, and Winder 2005) © 2005 IEEE. This low frequency sampling gives the features some robustness to interest point location error and is achieved by sampling at a higher pyramid level than the detection scale.

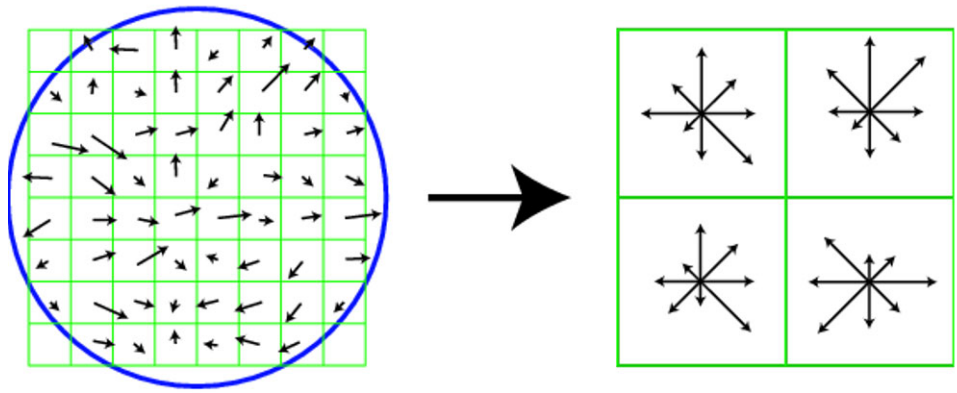
order to compensate for slight inaccuracies in the feature point detector (location, orientation, and scale), these multi-scale oriented patches (MOPS) are sampled at a spacing of five pixels relative to the detection scale, using a coarser level of the image pyramid to avoid aliasing. To compensate for affine photometric variations (linear exposure changes or bias and gain, (3.3)), patch intensities are re-scaled so that their mean is zero and their variance is one.

Scale invariant feature transform (SIFT). SIFT features are formed by computing the gradient at each pixel in a 16×16 window around the detected keypoint, using the appropriate level of the Gaussian pyramid at which the keypoint was detected. The gradient magnitudes are downweighted by a Gaussian fall-off function (shown as a blue circle in (Figure 4.18a) in order to reduce the influence of gradients far from the center, as these are more affected by small misregistrations.

In each 4×4 quadrant, a gradient orientation histogram is formed by (conceptually) adding the weighted gradient value to one of eight orientation histogram bins. To reduce the effects of location and dominant orientation misestimation, each of the original 256 weighted gradient magnitudes is softly added to $2 \times 2 \times 2$ histogram bins using trilinear interpolation. Softly distributing values to adjacent histogram bins is generally a good idea in any application where histograms are being computed, e.g., for Hough transforms (Section 4.3.2) or local histogram equalization (Section 3.1.4).

The resulting 128 non-negative values form a raw version of the SIFT descriptor vector. To reduce the effects of contrast or gain (additive variations are already removed by the gradient), the 128-D vector is normalized to unit length. To further make the descriptor robust to other photometric variations, values are clipped to 0.2 and the resulting vector is once again renormalized to unit length.

PCA-SIFT. Ke and Sukthankar (2004) propose a simpler way to compute descriptors inspired by SIFT; it computes the x and y (gradient) derivatives over a 39×39 patch and then reduces the resulting 3042-dimensional vector to 36 using principal component analysis (PCA) (Section 14.2.1 and Appendix A.1.2). Another popular variant of SIFT is SURF (Bay, Tuytelaars, and Van Gool 2006), which uses box filters to approximate the derivatives and



(a) image gradients (b) keypoint descriptor

Figure 4.18 A schematic representation of Lowe's (2004) scale invariant feature transform (SIFT): (a) Gradient orientations and magnitudes are computed at each pixel and weighted by a Gaussian fall-off function (blue circle). (b) A weighted gradient orientation histogram is then computed in each subregion, using trilinear interpolation. While this figure shows an 8×8 pixel patch and a 2×2 descriptor array, Lowe's actual implementation uses 16×16 patches and a 4×4 array of eight-bin histograms.

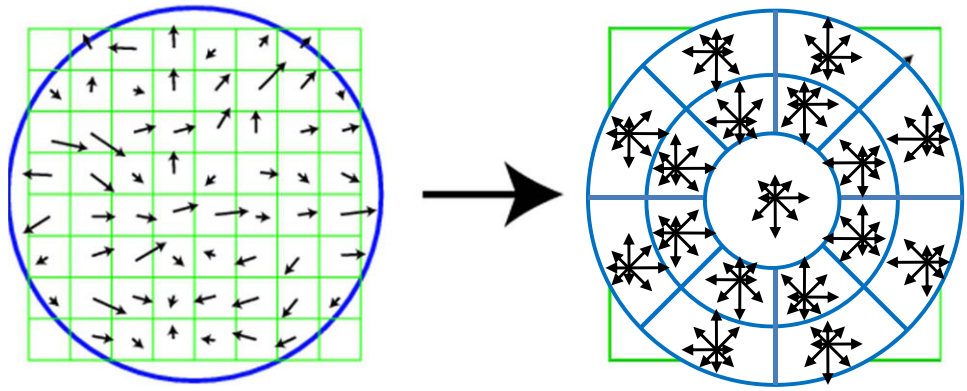
integrals used in SIFT.

Gradient location-orientation histogram (GLOH). This descriptor, developed by Mikolajczyk and Schmid (2005), is a variant on SIFT that uses a log-polar binning structure instead of the four quadrants used by Lowe (2004) (Figure 4.19). The spatial bins are of radius 6, 11, and 15, with eight angular bins (except for the central region), for a total of 17 spatial bins and 16 orientation bins. The 272-dimensional histogram is then projected onto a 128-dimensional descriptor using PCA trained on a large database. In their evaluation, Mikolajczyk and Schmid (2005) found that GLOH, which has the best performance overall, outperforms SIFT by a small margin.

Steerable filters. Steerable filters (Section 3.2.3) are combinations of derivative of Gaussian filters that permit the rapid computation of even and odd (symmetric and anti-symmetric) edge-like and corner-like features at all possible orientations (Freeman and Adelson 1991). Because they use reasonably broad Gaussians, they too are somewhat insensitive to localization and orientation errors.

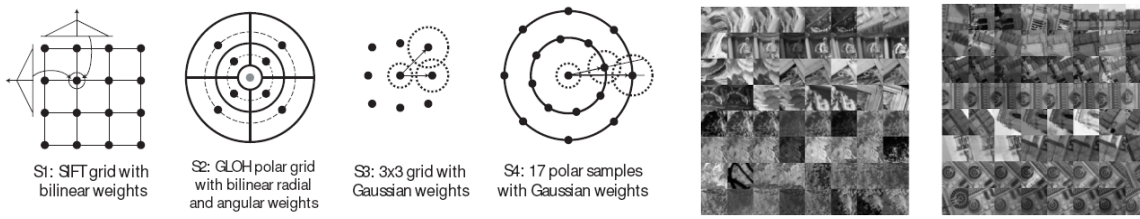
Performance of local descriptors. Among the local descriptors that Mikolajczyk and Schmid (2005) compared, they found that GLOH performed best, followed closely by SIFT (see Figure 4.25). They also present results for many other descriptors not covered in this book.

The field of feature descriptors continues to evolve rapidly, with some of the newer techniques looking at local color information (van de Weijer and Schmid 2006; Abdel-Hakim and Farag 2006). Winder and Brown (2007) develop a multi-stage framework for feature descriptor computation that subsumes both SIFT and GLOH (Figure 4.20a) and also allows



(a) image gradients (b) keypoint descriptor

Figure 4.19 The gradient location-orientation histogram (GLOH) descriptor uses log-polar bins instead of square bins to compute orientation histograms (Mikolajczyk and Schmid 2005).



(a) (b)

Figure 4.20 Spatial summation blocks for SIFT, GLOH, and some newly developed feature descriptors (Winder and Brown 2007) © 2007 IEEE: (a) The parameters for the new features, e.g., their Gaussian weights, are learned from a training database of (b) matched real-world image patches obtained from robust structure from motion applied to Internet photo collections (Hua, Brown, and Winder 2007).

them to learn optimal parameters for newer descriptors that outperform previous hand-tuned descriptors. Hua, Brown, and Winder (2007) extend this work by learning lower-dimensional projections of higher-dimensional descriptors that have the best discriminative power. Both of these papers use a database of real-world image patches (Figure 4.20b) obtained by sampling images at locations that were reliably matched using a robust structure-from-motion algorithm applied to Internet photo collections (Snavely, Seitz, and Szeliski 2006; Goesele, Snavely, Curless *et al.* 2007). In concurrent work, Tola, Lepetit, and Fua (2010) developed a similar DAISY descriptor for dense stereo matching and optimized its parameters based on ground truth stereo data.

While these techniques construct feature detectors that optimize for repeatability across *all* object classes, it is also possible to develop class- or instance-specific feature detectors that maximize *discriminability* from other classes (Ferencz, Learned-Miller, and Malik 2008).



Figure 4.21 Recognizing objects in a cluttered scene (Lowe 2004) © 2004 Springer. Two of the training images in the database are shown on the left. These are matched to the cluttered scene in the middle using SIFT features, shown as small squares in the right image. The affine warp of each recognized database image onto the scene is shown as a larger parallelogram in the right image.

4.1.3 Feature matching

Once we have extracted features and their descriptors from two or more images, the next step is to establish some preliminary feature matches between these images. In this section, we divide this problem into two separate components. The first is to select a *matching strategy*, which determines which correspondences are passed on to the next stage for further processing. The second is to devise efficient *data structures* and *algorithms* to perform this matching as quickly as possible. (See the discussion of related techniques in Section 14.3.2.)

Matching strategy and error rates

Determining which feature matches are reasonable to process further depends on the context in which the matching is being performed. Say we are given two images that overlap to a fair amount (e.g., for image stitching, as in Figure 4.16, or for tracking objects in a video). We know that most features in one image are likely to match the other image, although some may not match because they are occluded or their appearance has changed too much.

On the other hand, if we are trying to recognize how many known objects appear in a cluttered scene (Figure 4.21), most of the features may not match. Furthermore, a large number of potentially matching objects must be searched, which requires more efficient strategies, as described below.

To begin with, we assume that the feature descriptors have been designed so that Euclidean (vector magnitude) distances in feature space can be directly used for ranking potential matches. If it turns out that certain parameters (axes) in a descriptor are more reliable than others, it is usually preferable to re-scale these axes ahead of time, e.g., by determining how much they vary when compared against other known good matches (Hua, Brown, and Winder 2007). A more general process, which involves transforming feature vectors into a new scaled basis, is called *whitening* and is discussed in more detail in the context of eigenface-based face recognition (Section 14.2.1).

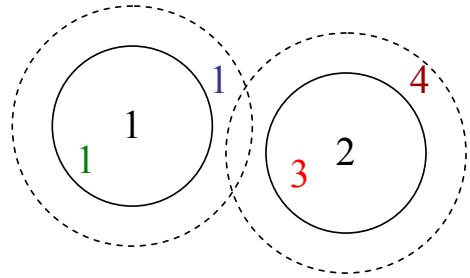


Figure 4.22 False positives and negatives: The black digits 1 and 2 are features being matched against a database of features in other images. At the current threshold setting (the solid circles), the green 1 is a *true positive* (good match), the blue 1 is a *false negative* (failure to match), and the red 3 is a *false positive* (incorrect match). If we set the threshold higher (the dashed circles), the blue 1 becomes a true positive but the brown 4 becomes an additional false positive.

	True matches	True non-matches		
Predicted matches	TP = 18	FP = 4	P' = 22	PPV = 0.82
Predicted non-matches	FN = 2	TN = 76	N' = 78	
	P = 20	N = 80	Total = 100	
	TPR = 0.90	FPR = 0.05		ACC = 0.94

Table 4.1 The number of matches correctly and incorrectly estimated by a feature matching algorithm, showing the number of true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN). The columns sum up to the actual number of positives (P) and negatives (N), while the rows sum up to the predicted number of positives (P') and negatives (N'). The formulas for the true positive rate (TPR), the false positive rate (FPR), the positive predictive value (PPV), and the accuracy (ACC) are given in the text.

Given a Euclidean distance metric, the simplest matching strategy is to set a threshold (maximum distance) and to return all matches from other images within this threshold. Setting the threshold too high results in too many *false positives*, i.e., incorrect matches being returned. Setting the threshold too low results in too many *false negatives*, i.e., too many correct matches being missed (Figure 4.22).

We can quantify the performance of a matching algorithm at a particular threshold by first counting the number of true and false matches and match failures, using the following definitions (Fawcett 2006):

- TP: true positives, i.e., number of correct matches;
- FN: false negatives, matches that were not correctly detected;
- FP: false positives, proposed matches that are incorrect;
- TN: true negatives, non-matches that were correctly rejected.

Table 4.1 shows a sample *confusion matrix* (contingency table) containing such numbers.

We can convert these numbers into *unit rates* by defining the following quantities (Fawcett 2006):

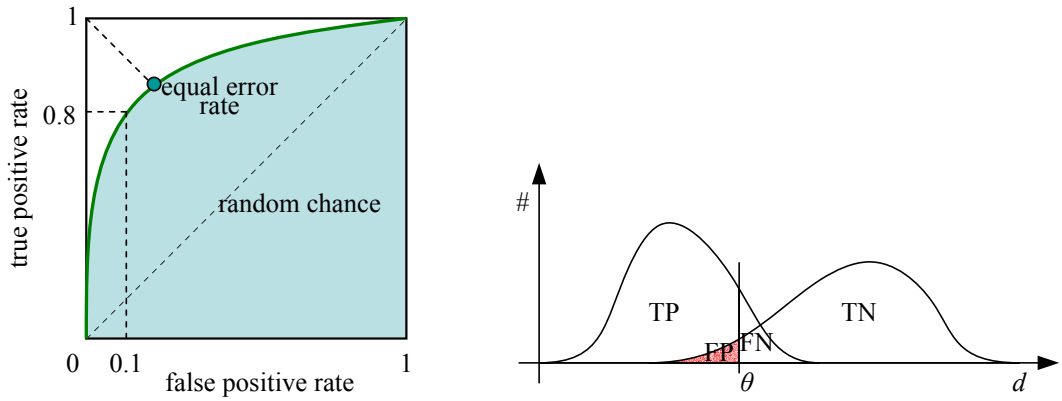


Figure 4.23 ROC curve and its related rates: (a) The ROC curve plots the true positive rate against the false positive rate for a particular combination of feature extraction and matching algorithms. Ideally, the true positive rate should be close to 1, while the false positive rate is close to 0. The area under the ROC curve (AUC) is often used as a single (scalar) measure of algorithm performance. Alternatively, the equal error rate is sometimes used. (b) The distribution of positives (matches) and negatives (non-matches) as a function of inter-feature distance d. As the threshold θ is increased, the number of true positives (TP) and false positives (FP) increases.

- true positive rate (TPR),

$$\text{TPR} = \frac{\text{TP}}{\text{TP+FN}} = \frac{\text{TP}}{\text{P}}; \tag{4.14}$$

- false positive rate (FPR),

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} = \frac{\text{FP}}{\text{N}}; \tag{4.15}$$

- positive predictive value (PPV),

$$\text{PPV} = \frac{\text{TP}}{\text{TP+FP}} = \frac{\text{TP}}{\text{P'}}; \tag{4.16}$$

- accuracy (ACC),

$$\text{ACC} = \frac{\text{TP+TN}}{\text{P+N}}. \tag{4.17}$$

In the *information retrieval* (or document retrieval) literature (Baeza-Yates and Ribeiro-Neto 1999; Manning, Raghavan, and Schütze 2008), the term *precision* (how many returned documents are relevant) is used instead of PPV and *recall* (what fraction of relevant documents was found) is used instead of TPR.

Any particular matching strategy (at a particular threshold or parameter setting) can be rated by the TPR and FPR numbers; ideally, the true positive rate will be close to 1 and the false positive rate close to 0. As we vary the matching threshold, we obtain a family of such points, which are collectively known as the *receiver operating characteristic (ROC curve)* (Fawcett 2006) (Figure 4.23a). The closer this curve lies to the upper left corner, i.e., the larger the area under the curve (AUC), the better its performance. Figure 4.23b shows how we can plot the number of matches and non-matches as a function of inter-feature distance d.

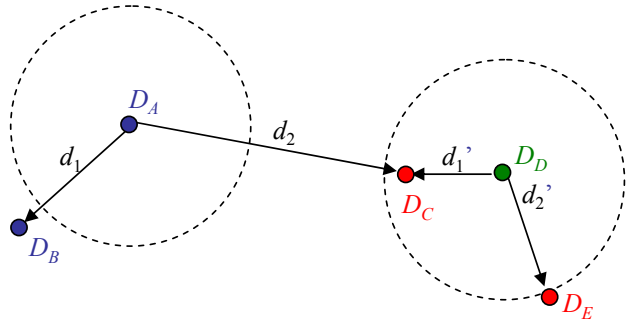


Figure 4.24 Fixed threshold, nearest neighbor, and nearest neighbor distance ratio matching. At a fixed distance threshold (dashed circles), descriptor D_A fails to match D_B and D_D incorrectly matches D_C and D_E. If we pick the nearest neighbor, D_A correctly matches D_B but D_D incorrectly matches D_C. Using nearest neighbor distance ratio (NNDR) matching, the small NNDR d_1/d_2 correctly matches D_A with D_B, and the large NNDR d_1'/d_2' correctly rejects matches for D_D.

These curves can then be used to plot an ROC curve (Exercise 4.3). The ROC curve can also be used to calculate the *mean average precision*, which is the average precision (PPV) as you vary the threshold to select the best results, then the two top results, etc.

The problem with using a fixed threshold is that it is difficult to set; the useful range of thresholds can vary a lot as we move to different parts of the feature space (Lowe 2004; Mikolajczyk and Schmid 2005). A better strategy in such cases is to simply match the *nearest neighbor* in feature space. Since some features may have no matches (e.g., they may be part of background clutter in object recognition or they may be occluded in the other image), a threshold is still used to reduce the number of false positives.

Ideally, this threshold itself will adapt to different regions of the feature space. If sufficient training data is available (Hua, Brown, and Winder 2007), it is sometimes possible to learn different thresholds for different features. Often, however, we are simply given a collection of images to match, e.g., when stitching images or constructing 3D models from unordered photo collections (Brown and Lowe 2007, 2003; Snavely, Seitz, and Szeliski 2006). In this case, a useful heuristic can be to compare the nearest neighbor distance to that of the second nearest neighbor, preferably taken from an image that is known not to match the target (e.g., a different object in the database) (Brown and Lowe 2002; Lowe 2004). We can define this *nearest neighbor distance ratio* (Mikolajczyk and Schmid 2005) as

$$\text{NNDR} = \frac{d_1}{d_2} = \frac{\|D_A - D_B\|}{\|D_A - D_C\|}, \tag{4.18}$$

where d_1 and d_2 are the nearest and second nearest neighbor distances, D_A is the target descriptor, and D_B and D_C are its closest two neighbors (Figure 4.24).

The effects of using these three different matching strategies for the feature descriptors evaluated by Mikolajczyk and Schmid (2005) are shown in Figure 4.25. As you can see, the nearest neighbor and NNDR strategies produce improved ROC curves.

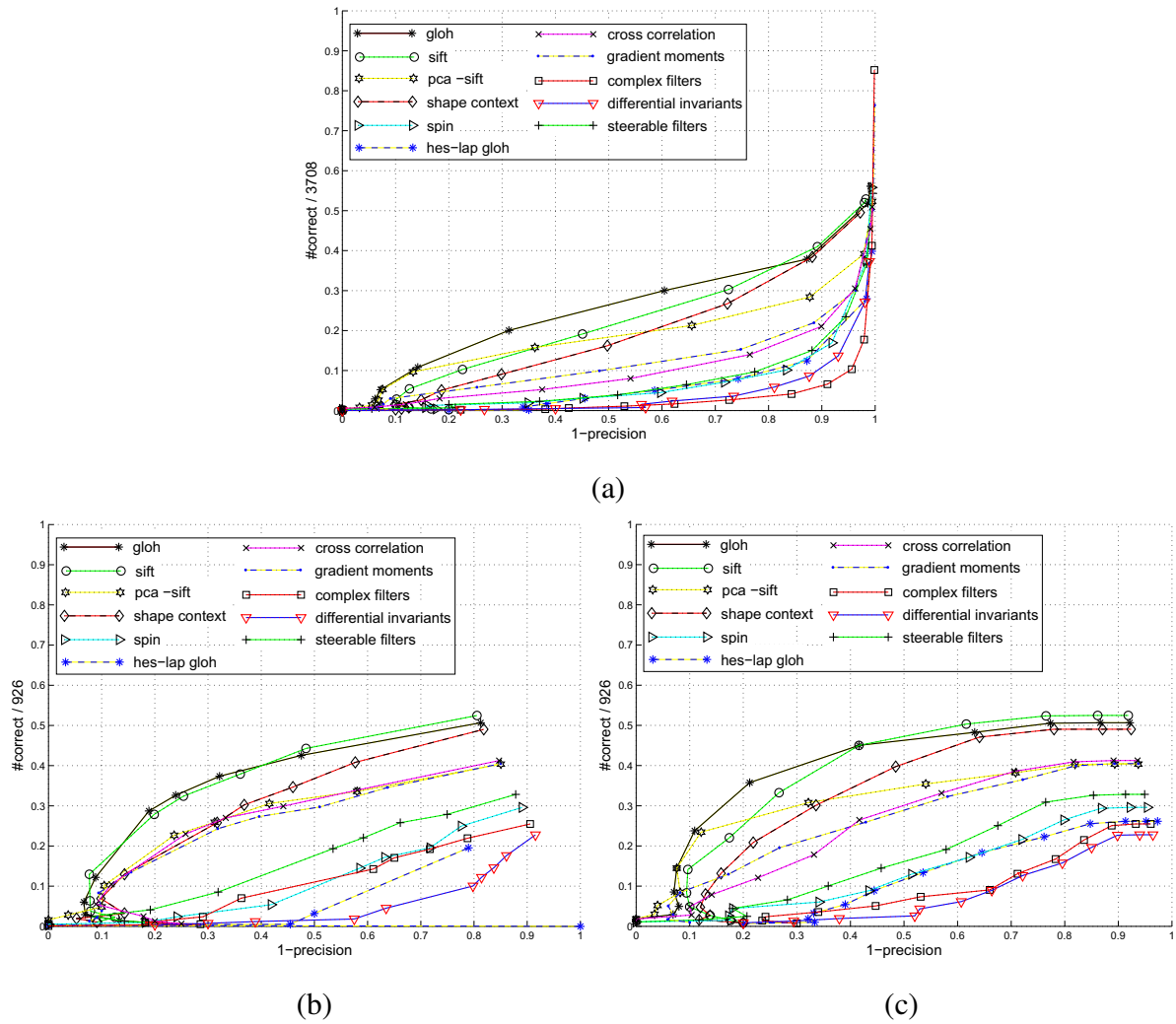


Figure 4.25 Performance of the feature descriptors evaluated by Mikolajczyk and Schmid (2005) © 2005 IEEE, shown for three matching strategies: (a) fixed threshold; (b) nearest neighbor; (c) nearest neighbor distance ratio (NNDR). Note how the ordering of the algorithms does not change that much, but the overall performance varies significantly between the different matching strategies.

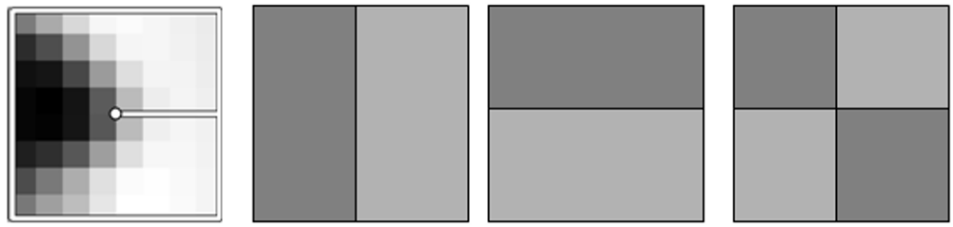


Figure 4.26 The three Haar wavelet coefficients used for hashing the MOPS descriptor devised by Brown, Szeliski, and Winder (2005) are computed by summing each 8×8 normalized patch over the light and dark gray regions and taking their difference.

Efficient matching

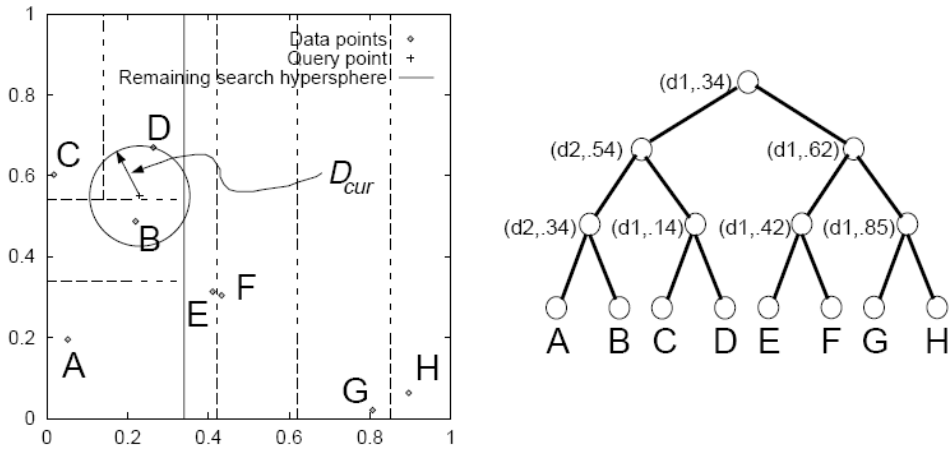
Once we have decided on a matching strategy, we still need to search efficiently for potential candidates. The simplest way to find all corresponding feature points is to compare all features against all other features in each pair of potentially matching images. Unfortunately, this is quadratic in the number of extracted features, which makes it impractical for most applications.

A better approach is to devise an *indexing structure*, such as a multi-dimensional search tree or a hash table, to rapidly search for features near a given feature. Such indexing structures can either be built for each image independently (which is useful if we want to only consider certain potential matches, e.g., searching for a particular object) or globally for all the images in a given database, which can potentially be faster, since it removes the need to iterate over each image. For extremely large databases (millions of images or more), even more efficient structures based on ideas from document retrieval (e.g., *vocabulary trees*, (Nistér and Stewénius 2006)) can be used (Section 14.3.2).

One of the simpler techniques to implement is multi-dimensional hashing, which maps descriptors into fixed size buckets based on some function applied to each descriptor vector. At matching time, each new feature is hashed into a bucket, and a search of nearby buckets is used to return potential candidates, which can then be sorted or graded to determine which are valid matches.

A simple example of hashing is the Haar wavelets used by Brown, Szeliski, and Winder (2005) in their MOPS paper. During the matching structure construction, each 8×8 scaled, oriented, and normalized MOPS patch is converted into a three-element index by performing sums over different quadrants of the patch (Figure 4.26). The resulting three values are normalized by their expected standard deviations and then mapped to the two (of $b = 10$) nearest 1D bins. The three-dimensional indices formed by concatenating the three quantized values are used to index the $2^3 = 8$ bins where the feature is stored (added). At query time, only the primary (closest) indices are used, so only a single three-dimensional bin needs to be examined. The coefficients in the bin can then be used to select k approximate nearest neighbors for further processing (such as computing the NNDR).

A more complex, but more widely applicable, version of hashing is called *locality sensitive hashing*, which uses unions of independently computed hashing functions to index the features (Gionis, Indyk, and Motwani 1999; Shakhnarovich, Darrell, and Indyk 2006). Shakhnarovich, Viola, and Darrell (2003) extend this technique to be more sensitive to the



(a) (b)

Figure 4.27 K-d tree and best bin first (BBF) search (Beis and Lowe 1999) © 1999 IEEE: (a) The spatial arrangement of the axis-aligned cutting planes is shown using dashed lines. Individual data points are shown as small diamonds. (b) The same subdivision can be represented as a tree, where each interior node represents an axis-aligned cutting plane (e.g., the top node cuts along dimension d1 at value .34) and each leaf node is a data point. During a BBF search, a query point (denoted by "+") first looks in its containing bin (D) and then in its nearest adjacent bin (B), rather than its closest neighbor in the tree (C).

distribution of points in parameter space, which they call *parameter-sensitive hashing*. Even more recent work converts high-dimensional descriptor vectors into binary codes that can be compared using Hamming distances (Torralba, Weiss, and Fergus 2008; Weiss, Torralba, and Fergus 2008) or that can accommodate arbitrary kernel functions (Kulis and Grauman 2009; Raginsky and Lazebnik 2009).

Another widely used class of indexing structures are multi-dimensional search trees. The best known of these are *k-d trees*, also often written as *k*d-trees, which divide the multi-dimensional feature space along alternating axis-aligned hyperplanes, choosing the threshold along each axis so as to maximize some criterion, such as the search tree balance (Samet 1989). Figure 4.27 shows an example of a two-dimensional k-d tree. Here, eight different data points A–H are shown as small diamonds arranged on a two-dimensional plane. The k-d tree recursively splits this plane along axis-aligned (horizontal or vertical) cutting planes. Each split can be denoted using the dimension number and split value (Figure 4.27b). The splits are arranged so as to try to balance the tree, i.e., to keep its maximum depth as small as possible. At query time, a classic k-d tree search first locates the query point (+) in its appropriate bin (D), and then searches nearby leaves in the tree (C, B, . . .) until it can guarantee that the nearest neighbor has been found. The best bin first (BBF) search (Beis and Lowe 1999) searches bins in order of their spatial proximity to the query point and is therefore usually more efficient.

Many additional data structures have been developed over the years for solving nearest neighbor problems (Arya, Mount, Netanyahu *et al.* 1998; Liang, Liu, Xu *et al.* 2001; Hjaltason and Samet 2003). For example, Nene and Nayar (1997) developed a technique they call

slicing that uses a series of 1D binary searches on the point list sorted along different dimensions to efficiently cull down a list of candidate points that lie within a hypercube of the query point. Grauman and Darrell (2005) reweight the matches at different levels of an indexing tree, which allows their technique to be less sensitive to discretization errors in the tree construction. Nistér and Stewénius (2006) use a *metric tree*, which compares feature descriptors to a small number of prototypes at each level in a hierarchy. The resulting quantized *visual words* can then be used with classical information retrieval (document relevance) techniques to quickly winnow down a set of potential candidates from a database of millions of images (Section 14.3.2). Muja and Lowe (2009) compare a number of these approaches, introduce a new one of their own (priority search on hierarchical k-means trees), and conclude that multiple randomized k-d trees often provide the best performance. Despite all of this promising work, the rapid computation of image feature correspondences remains a challenging open research problem.

Feature match verification and densification

Once we have some hypothetical (putative) matches, we can often use geometric alignment (Section 6.1) to verify which matches are *inliers* and which ones are *outliers*. For example, if we expect the whole image to be translated or rotated in the matching view, we can fit a global geometric transform and keep only those feature matches that are sufficiently close to this estimated transformation. The process of selecting a small set of seed matches and then verifying a larger set is often called *random sampling* or RANSAC (Section 6.1.4). Once an initial set of correspondences has been established, some systems look for additional matches, e.g., by looking for additional correspondences along epipolar lines (Section 11.1) or in the vicinity of estimated locations based on the global transform. These topics are discussed further in Sections 6.1, 11.2, and 14.3.1.

4.1.4 Feature tracking

An alternative to independently finding features in all candidate images and then matching them is to find a set of likely feature locations in a first image and to then *search* for their corresponding locations in subsequent images. This kind of *detect then track* approach is more widely used for video tracking applications, where the expected amount of motion and appearance deformation between adjacent frames is expected to be small.

The process of selecting good features to track is closely related to selecting good features for more general recognition applications. In practice, regions containing high gradients in both directions, i.e., which have high eigenvalues in the auto-correlation matrix (4.8), provide stable locations at which to find correspondences (Shi and Tomasi 1994).

In subsequent frames, searching for locations where the corresponding patch has low squared difference (4.1) often works well enough. However, if the images are undergoing brightness change, explicitly compensating for such variations (8.9) or using *normalized cross-correlation* (8.11) may be preferable. If the search range is large, it is also often more efficient to use a *hierarchical* search strategy, which uses matches in lower-resolution images to provide better initial guesses and hence speed up the search (Section 8.1.1). Alternatives to this strategy involve learning what the appearance of the patch being tracked should be and then searching for it in the vicinity of its predicted position (Avidan 2001; Jurie and Dhome



Figure 4.28 Feature tracking using an affine motion model (Shi and Tomasi 1994) © 1994 IEEE, Top row: image patch around the tracked feature location. Bottom row: image patch after warping back toward the first frame using an affine deformation. Even though the speed sign gets larger from frame to frame, the affine transformation maintains a good resemblance between the original and subsequent tracked frames.

2002; Williams, Blake, and Cipolla 2003). These topics are all covered in more detail in Section 8.1.3.

If features are being tracked over longer image sequences, their appearance can undergo larger changes. You then have to decide whether to continue matching against the originally detected patch (feature) or to re-sample each subsequent frame at the matching location. The former strategy is prone to failure as the original patch can undergo appearance changes such as foreshortening. The latter runs the risk of the feature drifting from its original location to some other location in the image (Shi and Tomasi 1994). (Mathematically, small misregistration errors compound to create a *Markov Random Walk*, which leads to larger drift over time.)

A preferable solution is to compare the original patch to later image locations using an *affine* motion model (Section 8.2). Shi and Tomasi (1994) first compare patches in neighboring frames using a translational model and then use the location estimates produced by this step to initialize an affine registration between the patch in the current frame and the base frame where a feature was first detected (Figure 4.28). In their system, features are only detected infrequently, i.e., only in regions where tracking has failed. In the usual case, an area around the current *predicted* location of the feature is searched with an incremental registration algorithm (Section 8.1.3). The resulting tracker is often called the Kanade–Lucas–Tomasi (KLT) tracker.

Since their original work on feature tracking, Shi and Tomasi's approach has generated a string of interesting follow-on papers and applications. Beardsley, Torr, and Zisserman (1996) use extended feature tracking combined with structure from motion (Chapter 7) to incrementally build up sparse 3D models from video sequences. Kang, Szeliski, and Shum (1997) tie together the corners of adjacent (regularly gridded) patches to provide some additional stability to the tracking, at the cost of poorer handling of occlusions. Tommasini, Fusiello, Trucco *et al.* (1998) provide a better spurious match rejection criterion for the basic Shi and Tomasi algorithm, Collins and Liu (2003) provide improved mechanisms for feature selection and dealing with larger appearance changes over time, and Shafique and Shah (2005) develop algorithms for feature matching (data association) for videos with large numbers of



Figure 4.29 Real-time head tracking using the fast trained classifiers of Lepetit, Pilet, and Fua (2004) © 2004 IEEE.

moving objects or points. Yilmaz, Javed, and Shah (2006) and Lepetit and Fua (2005) survey the larger field of object tracking, which includes not only feature-based techniques but also alternative techniques based on contour and region (Section 5.1).

One of the newest developments in feature tracking is the use of learning algorithms to build special-purpose recognizers to rapidly search for matching features anywhere in an image (Lepetit, Pilet, and Fua 2006; Hinterstoisser, Benhimane, Navab *et al.* 2008; Rogez, Rihan, Ramalingam *et al.* 2008; Özuysal, Calonder, Lepetit *et al.* 2010).[2] By taking the time to train classifiers on sample patches and their affine deformations, extremely fast and reliable feature detectors can be constructed, which enables much faster motions to be supported (Figure 4.29). Coupling such features to deformable models (Pilet, Lepetit, and Fua 2008) or structure-from-motion algorithms (Klein and Murray 2008) can result in even higher stability.

4.1.5 *Application*: Performance-driven animation

One of the most compelling applications of fast feature tracking is *performance-driven animation*, i.e., the interactive deformation of a 3D graphics model based on tracking a user's motions (Williams 1990; Litwinowicz and Williams 1994; Lepetit, Pilet, and Fua 2004).

Buck, Finkelstein, Jacobs *et al.* (2000) present a system that tracks a user's facial expressions and head motions and then uses them to morph among a series of hand-drawn sketches. An animator first extracts the eye and mouth regions of each sketch and draws control lines over each image (Figure 4.30a). At run time, a face-tracking system (Toyama 1998) determines the current location of these features (Figure 4.30b). The animation system decides

[2] See also my previous comment on earlier work in learning-based tracking (Avidan 2001; Jurie and Dhome 2002; Williams, Blake, and Cipolla 2003).



(a) (b) (c) (d)

Figure 4.30 Performance-driven, hand-drawn animation (Buck, Finkelstein, Jacobs *et al.* 2000) © 2000 ACM: (a) eye and mouth portions of hand-drawn sketch with their overlaid control lines; (b) an input video frame with the tracked features overlaid; (c) a different input video frame along with its (d) corresponding hand-drawn animation.

which input images to morph based on nearest neighbor feature appearance matching and triangular barycentric interpolation. It also computes the global location and orientation of the head from the tracked features. The resulting morphed eye and mouth regions are then composited back into the overall head model to yield a frame of hand-drawn animation (Figure 4.30d).

In more recent work, Barnes, Jacobs, Sanders *et al.* (2008) watch users animate paper cutouts on a desk and then turn the resulting motions and drawings into seamless 2D animations.

4.2 Edges

While interest points are useful for finding image locations that can be accurately matched in 2D, edge points are far more plentiful and often carry important semantic associations. For example, the boundaries of objects, which also correspond to occlusion events in 3D, are usually delineated by visible contours. Other kinds of edges correspond to shadow boundaries or crease edges, where surface orientation changes rapidly. Isolated edge points can also be grouped into longer *curves* or *contours*, as well as *straight line segments* (Section 4.3). It is interesting that even young children have no difficulty in recognizing familiar objects or animals from such simple line drawings.

4.2.1 Edge detection

Given an image, how can we find the salient edges? Consider the color images in Figure 4.31. If someone asked you to point out the most "salient" or "strongest" edges or the object boundaries (Martin, Fowlkes, and Malik 2004; Arbeláez, Maire, Fowlkes *et al.* 2010), which ones would you trace? How closely do your perceptions match the edge images shown in Figure 4.31?

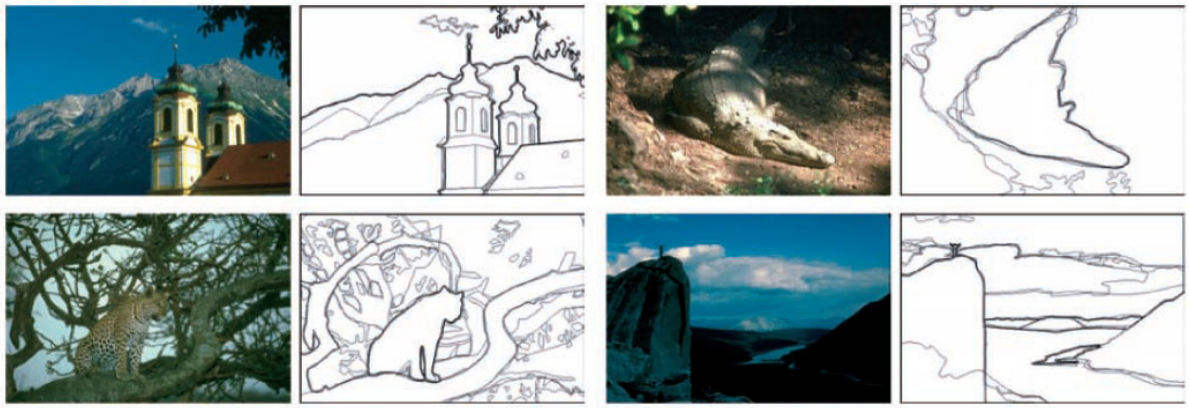


Figure 4.31 Human boundary detection (Martin, Fowlkes, and Malik 2004) © 2004 IEEE. The darkness of the edges corresponds to how many human subjects marked an object boundary at that location.

Qualitatively, edges occur at boundaries between regions of different color, intensity, or texture. Unfortunately, segmenting an image into coherent regions is a difficult task, which we address in Chapter 5. Often, it is preferable to detect edges using only purely local information.

Under such conditions, a reasonable approach is to define an edge as a location of *rapid intensity variation*.[3] Think of an image as a height field. On such a surface, edges occur at locations of *steep slopes*, or equivalently, in regions of closely packed contour lines (on a topographic map).

A mathematical way to define the slope and direction of a surface is through its gradient,

$$\boldsymbol{J}(\boldsymbol{x}) = \nabla I(\boldsymbol{x}) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)(\boldsymbol{x}). \tag{4.19}$$

The local gradient vector $\boldsymbol{J}$ points in the direction of *steepest ascent* in the intensity function. Its magnitude is an indication of the slope or strength of the variation, while its orientation points in a direction *perpendicular* to the local contour.

Unfortunately, taking image derivatives accentuates high frequencies and hence amplifies noise, since the proportion of noise to signal is larger at high frequencies. It is therefore prudent to smooth the image with a low-pass filter prior to computing the gradient. Because we would like the response of our edge detector to be independent of orientation, a circularly symmetric smoothing filter is desirable. As we saw in Section 3.2, the Gaussian is the only separable circularly symmetric filter and so it is used in most edge detection algorithms. Canny (1986) discusses alternative filters and a number of researcher review alternative edge detection algorithms and compare their performance (Davis 1975; Nalwa and Binford 1986; Nalwa 1987; Deriche 1987; Freeman and Adelson 1991; Nalwa 1993; Heath, Sarkar, Sanocki *et al.* 1998; Crane 1997; Ritter and Wilson 2000; Bowyer, Kranenburg, and Dougherty 2001; Arbeláez, Maire, Fowlkes *et al.* 2010).

Because differentiation is a linear operation, it commutes with other linear filtering oper-

[3] We defer the topic of edge detection in color images.

ations. The gradient of the smoothed image can therefore be written as

$$\boldsymbol{J}_\sigma(\boldsymbol{x}) = \nabla[G_\sigma(\boldsymbol{x}) * I(\boldsymbol{x})] = [\nabla G_\sigma](\boldsymbol{x}) * I(\boldsymbol{x}), \tag{4.20}$$

i.e., we can convolve the image with the horizontal and vertical derivatives of the Gaussian kernel function,

$$\nabla G_\sigma(\boldsymbol{x}) = (\frac{\partial G_\sigma}{\partial x}, \frac{\partial G_\sigma}{\partial y})(\boldsymbol{x}) = [-x \ -y]\frac{1}{\sigma^3}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.21}$$

(The parameter σ indicates the width of the Gaussian.) This is the same computation that is performed by Freeman and Adelson's (1991) first-order steerable filter, which we already covered in Section 3.2.3.

For many applications, however, we wish to thin such a continuous gradient image to only return isolated edges, i.e., as single pixels at discrete locations along the edge contours. This can be achieved by looking for *maxima* in the edge strength (gradient magnitude) in a direction *perpendicular* to the edge orientation, i.e., along the gradient direction.

Finding this maximum corresponds to taking a directional derivative of the strength field in the direction of the gradient and then looking for zero crossings. The desired directional derivative is equivalent to the dot product between a second gradient operator and the results of the first,

$$S_\sigma(\boldsymbol{x}) = \nabla \cdot \boldsymbol{J}_\sigma(\boldsymbol{x}) = [\nabla^2 G_\sigma](\boldsymbol{x}) * I(\boldsymbol{x})]. \tag{4.22}$$

The gradient operator dot product with the gradient is called the *Laplacian*. The convolution kernel

$$\nabla^2 G_\sigma(\boldsymbol{x}) = \frac{1}{\sigma^3}\left(2 - \frac{x^2+y^2}{2\sigma^2}\right)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.23}$$

is therefore called the *Laplacian of Gaussian* (LoG) kernel (Marr and Hildreth 1980). This kernel can be split into two separable parts,

$$\nabla^2 G_\sigma(\boldsymbol{x}) = \frac{1}{\sigma^3}\left(1 - \frac{x^2}{2\sigma^2}\right)G_\sigma(x)G_\sigma(y) + \frac{1}{\sigma^3}\left(1 - \frac{y^2}{2\sigma^2}\right)G_\sigma(y)G_\sigma(x) \tag{4.24}$$

(Wiejak, Buxton, and Buxton 1985), which allows for a much more efficient implementation using separable filtering (Section 3.2.1).

In practice, it is quite common to replace the Laplacian of Gaussian convolution with a Difference of Gaussian (DoG) computation, since the kernel shapes are qualitatively similar (Figure 3.35). This is especially convenient if a "Laplacian pyramid" (Section 3.5) has already been computed.[4]

In fact, it is not strictly necessary to take differences between adjacent levels when computing the edge field. Think about what a zero crossing in a "generalized" difference of Gaussians image represents. The finer (smaller kernel) Gaussian is a noise-reduced version of the original image. The coarser (larger kernel) Gaussian is an estimate of the average intensity over a larger region. Thus, whenever the DoG image changes sign, this corresponds to the (slightly blurred) image going from relatively darker to relatively lighter, as compared to the average intensity in that neighborhood.

[4] Recall that Burt and Adelson's (1983a) "Laplacian pyramid" actually computed differences of Gaussian-filtered levels.

Once we have computed the sign function $S(\boldsymbol{x})$, we must find its *zero crossings* and convert these into edge elements (*edgels*). An easy way to detect and represent zero crossings is to look for adjacent pixel locations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ where the sign changes value, i.e., $[S(\boldsymbol{x}_i) > 0] \neq [S(\boldsymbol{x}_j) > 0]$.

The sub-pixel location of this crossing can be obtained by computing the "x-intercept" of the "line" connecting $S(\boldsymbol{x}_i)$ and $S(\boldsymbol{x}_j)$,

$$\boldsymbol{x}_z = \frac{\boldsymbol{x}_i S(\boldsymbol{x}_j) - \boldsymbol{x}_j S(\boldsymbol{x}_i)}{S(\boldsymbol{x}_j) - S(\boldsymbol{x}_i)}. \tag{4.25}$$

The orientation and strength of such edgels can be obtained by linearly interpolating the gradient values computed on the original pixel grid.

An alternative edgel representation can be obtained by linking adjacent edgels on the dual grid to form edgels that live *inside* each square formed by four adjacent pixels in the original pixel grid.[5] The (potential) advantage of this representation is that the edgels now live on a grid offset by half a pixel from the original pixel grid and are thus easier to store and access. As before, the orientations and strengths of the edges can be computed by interpolating the gradient field or estimating these values from the difference of Gaussian image (see Exercise 4.7).

In applications where the accuracy of the edge orientation is more important, higher-order steerable filters can be used (Freeman and Adelson 1991) (see Section 3.2.3). Such filters are more selective for more elongated edges and also have the possibility of better modeling curve intersections because they can represent multiple orientations at the same pixel (Figure 3.16). Their disadvantage is that they are more expensive to compute and the directional derivative of the edge strength does not have a simple closed form solution.[6]

Scale selection and blur estimation

As we mentioned before, the derivative, Laplacian, and Difference of Gaussian filters (4.20–4.23) all require the selection of a spatial scale parameter σ. If we are only interested in detecting sharp edges, the width of the filter can be determined from image noise characteristics (Canny 1986; Elder and Zucker 1998). However, if we want to detect edges that occur at different resolutions (Figures 4.32b–c), a *scale-space* approach that detects and then selects edges at different scales may be necessary (Witkin 1983; Lindeberg 1994, 1998a; Nielsen, Florack, and Deriche 1997).

Elder and Zucker (1998) present a principled approach to solving this problem. Given a known image noise level, their technique computes, for every pixel, the minimum scale at which an edge can be reliably detected (Figure 4.32d). Their approach first computes gradients densely over an image by selecting among gradient estimates computed at different scales, based on their gradient magnitudes. It then performs a similar estimate of minimum scale for directed second derivatives and uses zero crossings of this latter quantity to robustly select edges (Figures 4.32e–f). As an optional final step, the blur width of each edge can be computed from the distance between extrema in the second derivative response minus the width of the Gaussian filter.

[5] This algorithm is a 2D version of the 3D *marching cubes* isosurface extraction algorithm (Lorensen and Cline 1987).

[6] In fact, the edge orientation can have a 180° ambiguity for "bar edges", which makes the computation of zero crossings in the derivative more tricky.

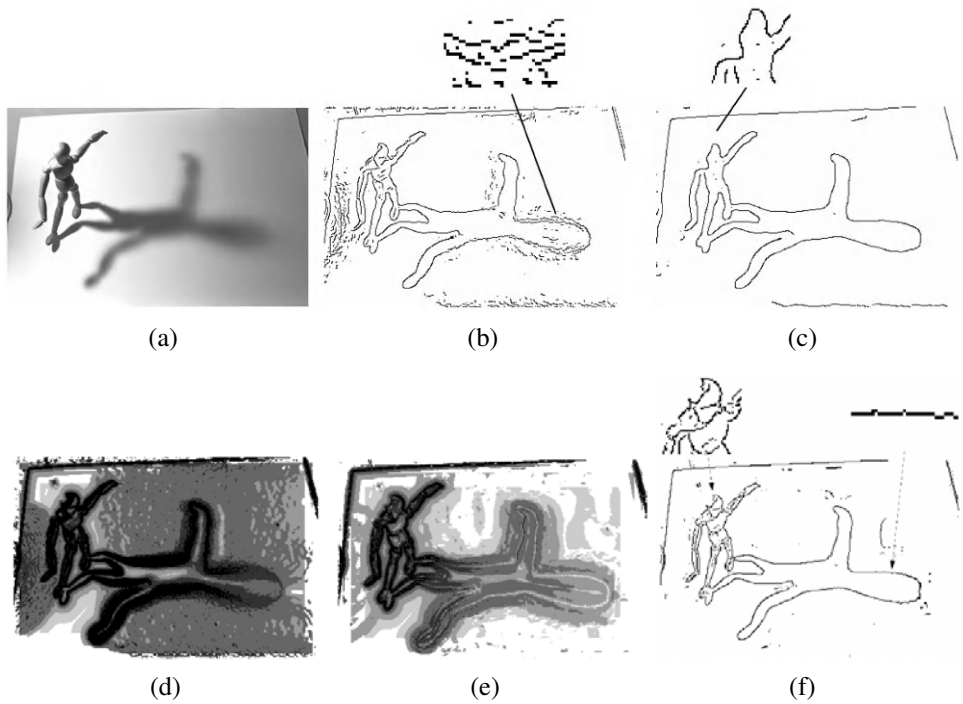


Figure 4.32 Scale selection for edge detection (Elder and Zucker 1998) © 1998 IEEE: (a) original image; (b–c) Canny/Deriche edge detector tuned to the finer (mannequin) and coarser (shadow) scales; (d) minimum reliable scale for gradient estimation; (e) minimum reliable scale for second derivative estimation; (f) final detected edges.

Color edge detection

While most edge detection techniques have been developed for grayscale images, color images can provide additional information. For example, noticeable edges between *iso-luminant* colors (colors that have the same luminance) are useful cues but fail to be detected by grayscale edge operators.

One simple approach is to combine the outputs of grayscale detectors run on each color band separately.[7] However, some care must be taken. For example, if we simply sum up the gradients in each of the color bands, the signed gradients may actually cancel each other! (Consider, for example a pure red-to-green edge.) We could also detect edges independently in each band and then take the union of these, but this might lead to thickened or doubled edges that are hard to link.

A better approach is to compute the *oriented energy* in each band (Morrone and Burr 1988; Perona and Malik 1990a), e.g., using a second-order steerable filter (Section 3.2.3) (Freeman and Adelson 1991), and then sum up the orientation-weighted energies and find their joint best orientation. Unfortunately, the directional derivative of this energy may not have a closed form solution (as in the case of signed first-order steerable filters), so a simple zero crossing-based strategy cannot be used. However, the technique described by Elder and

[7] Instead of using the raw RGB space, a more perceptually uniform color space such as L*a*b* (see Section 2.3.2) can be used instead. When trying to match human performance (Martin, Fowlkes, and Malik 2004), this makes sense. However, in terms of the physics of the underlying image formation and sensing, it may be a questionable strategy.

Zucker (1998) can be used to compute these zero crossings numerically instead.

An alternative approach is to estimate local color statistics in regions around each pixel (Ruzon and Tomasi 2001; Martin, Fowlkes, and Malik 2004). This has the advantage that more sophisticated techniques (e.g., 3D color histograms) can be used to compare regional statistics and that additional measures, such as texture, can also be considered. Figure 4.33 shows the output of such detectors.

Of course, many other approaches have been developed for detecting color edges, dating back to early work by Nevatia (1977). Ruzon and Tomasi (2001) and Gevers, van de Weijer, and Stokman (2006) provide good reviews of these approaches, which include ideas such as fusing outputs from multiple channels, using multidimensional gradients, and vector-based methods.

Combining edge feature cues

If the goal of edge detection is to match human *boundary detection* performance (Bowyer, Kranenburg, and Dougherty 2001; Martin, Fowlkes, and Malik 2004; Arbeláez, Maire, Fowlkes *et al.* 2010), as opposed to simply finding stable features for matching, even better detectors can be constructed by combining multiple low-level cues such as brightness, color, and texture.

Martin, Fowlkes, and Malik (2004) describe a system that combines brightness, color, and texture edges to produce state-of-the-art performance on a database of hand-segmented natural color images (Martin, Fowlkes, Tal *et al.* 2001). First, they construct and train[8] separate oriented half-disc detectors for measuring significant differences in brightness (luminance), color (a* and b* channels, summed responses), and texture (un-normalized filter bank responses from the work of Malik, Belongie, Leung *et al.* (2001)). Some of the responses are then sharpened using a soft non-maximal suppression technique. Finally, the outputs of the three detectors are combined using a variety of machine-learning techniques, from which logistic regression is found to have the best tradeoff between speed, space and accuracy . The resulting system (see Figure 4.33 for some examples) is shown to outperform previously developed techniques. Maire, Arbelaez, Fowlkes *et al.* (2008) improve on these results by combining the detector based on local appearance with a *spectral* (segmentation-based) detector (Belongie and Malik 1998). In more recent work, Arbeláez, Maire, Fowlkes *et al.* (2010) build a hierarchical segmentation on top of this edge detector using a variant of the watershed algorithm.

4.2.2 Edge linking

While isolated edges can be useful for a variety of applications, such as line detection (Section 4.3) and sparse stereo matching (Section 11.2), they become even more useful when linked into continuous contours.

If the edges have been detected using zero crossings of some function, linking them up is straightforward, since adjacent edgels share common endpoints. Linking the edgels into chains involves picking up an unlinked edgel and following its neighbors in both directions. Either a sorted list of edgels (sorted first by x coordinates and then by y coordinates, for example) or a 2D array can be used to accelerate the neighbor finding. If edges were not

[8] The training uses 200 labeled images and testing is performed on a different set of 100 images.



Figure 4.33 Combined brightness, color, texture boundary detector (Martin, Fowlkes, and Malik 2004) © 2004 IEEE. Successive rows show the outputs of the brightness gradient (BG), color gradient (CG), texture gradient (TG), and combined (BG+CG+TG) detectors. The final row shows human-labeled boundaries derived from a database of hand-segmented images (Martin, Fowlkes, Tal *et al.* 2001).

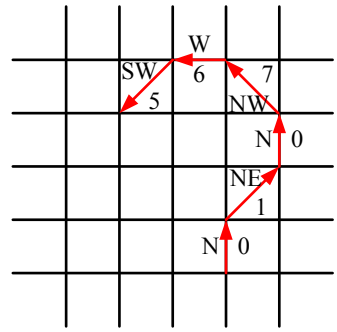


Figure 4.34 Chain code representation of a grid-aligned linked edge chain. The code is represented as a series of direction codes, e.g, 0 1 0 7 6 5, which can further be compressed using predictive and run-length coding.

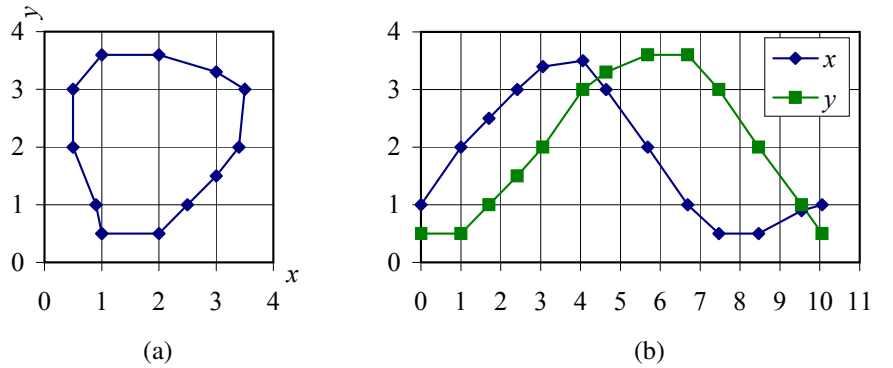


Figure 4.35 Arc-length parameterization of a contour: (a) discrete points along the contour are first transcribed as (b) (x, y) pairs along the arc length s. This curve can then be regularly re-sampled or converted into alternative (e.g., Fourier) representations.

detected using zero crossings, finding the continuation of an edgel can be tricky. In this case, comparing the orientation (and, optionally, phase) of adjacent edgels can be used for disambiguation. Ideas from connected component computation can also sometimes be used to make the edge linking process even faster (see Exercise 4.8).

Once the edgels have been linked into chains, we can apply an optional thresholding with hysteresis to remove low-strength contour segments (Canny 1986). The basic idea of hysteresis is to set two different thresholds and allow a curve being tracked above the higher threshold to dip in strength down to the lower threshold.

Linked edgel lists can be encoded more compactly using a variety of alternative representations. A *chain code* encodes a list of connected points lying on an $\mathcal{N}_8$ grid using a three-bit code corresponding to the eight cardinal directions (N, NE, E, SE, S, SW, W, NW) between a point and its successor (Figure 4.34). While this representation is more compact than the original edgel list (especially if predictive variable-length coding is used), it is not very suitable for further processing.

A more useful representation is the *arc length parameterization* of a contour, $\boldsymbol{x}(s)$, where s denotes the arc length along a curve. Consider the linked set of edgels shown in Fig-

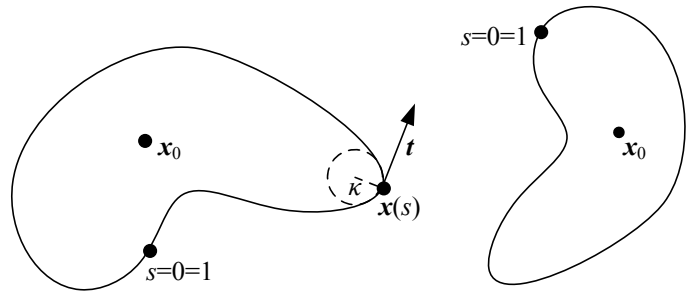


Figure 4.36 Matching two contours using their arc-length parameterization. If both curves are normalized to unit length, $s \in [0, 1]$ and centered around their centroid $\boldsymbol{x}_0$, they will have the same descriptor up to an overall "temporal" shift (due to different starting points for $s = 0$) and a phase (x-y) shift (due to rotation).

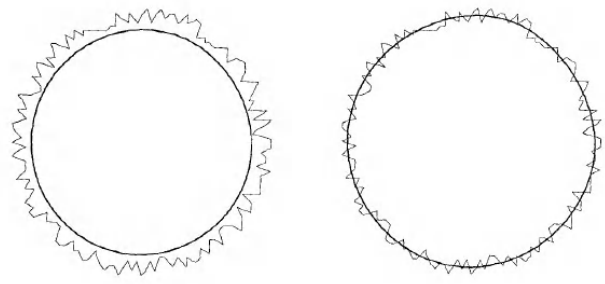


Figure 4.37 Curve smoothing with a Gaussian kernel (Lowe 1988) © 1998 IEEE: (a) without a shrinkage correction term; (b) with a shrinkage correction term.

ure 4.35a. We start at one point (the dot at $(1.0, 0.5)$ in Figure 4.35a) and plot it at coordinate $s = 0$ (Figure 4.35b). The next point at $(2.0, 0.5)$ gets plotted at $s = 1$, and the next point at $(2.5, 1.0)$ gets plotted at $s = 1.7071$, i.e., we increment s by the length of each edge segment. The resulting plot can be resampled on a regular (say, integral) s grid before further processing.

The advantage of the arc-length parameterization is that it makes matching and processing (e.g., smoothing) operations much easier. Consider the two curves describing similar shapes shown in Figure 4.36. To compare the curves, we first subtract the average values $\boldsymbol{x}_0 = \int_s \boldsymbol{x}(s)$ from each descriptor. Next, we rescale each descriptor so that s goes from 0 to 1 instead of 0 to S, i.e., we divide $\boldsymbol{x}(s)$ by S. Finally, we take the Fourier transform of each normalized descriptor, treating each $\boldsymbol{x} = (x, y)$ value as a complex number. If the original curves are the same (up to an unknown scale and rotation), the resulting Fourier transforms should differ only by a scale change in magnitude plus a constant complex phase shift, due to rotation, and a linear phase shift in the domain, due to different starting points for s (see Exercise 4.9).

Arc-length parameterization can also be used to smooth curves in order to remove digitization noise. However, if we just apply a regular smoothing filter, the curve tends to shrink

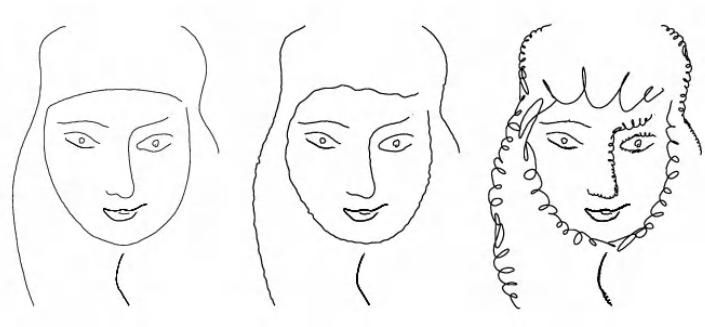


Figure 4.38 Changing the character of a curve without affecting its sweep (Finkelstein and Salesin 1994) © 1994 ACM: higher frequency wavelets can be replaced with exemplars from a style library to effect different local appearances.

on itself (Figure 4.37a). Lowe (1989) and Taubin (1995) describe techniques that compensate for this shrinkage by adding an offset term based on second derivative estimates or a larger smoothing kernel (Figure 4.37b). An alternative approach, based on selectively modifying different frequencies in a wavelet decomposition, is presented by Finkelstein and Salesin (1994). In addition to controlling shrinkage without affecting its "sweep", wavelets allow the "character" of a curve to be interactively modified, as shown in Figure 4.38.

The evolution of curves as they are smoothed and simplified is related to "grassfire" (distance) transforms and region skeletons (Section 3.3.3) (Tek and Kimia 2003), and can be used to recognize objects based on their contour shape (Sebastian and Kimia 2005). More local descriptors of curve shape such as *shape contexts* (Belongie, Malik, and Puzicha 2002) can also be used for recognition and are potentially more robust to missing parts due to occlusions.

The field of contour detection and linking continues to evolve rapidly and now includes techniques for global contour grouping, boundary completion, and junction detection (Maire, Arbelaez, Fowlkes *et al.* 2008), as well as grouping contours into likely regions (Arbeláez, Maire, Fowlkes *et al.* 2010) and wide-baseline correspondence (Meltzer and Soatto 2008).

4.2.3 *Application*: Edge editing and enhancement

While edges can serve as components for object recognition or features for matching, they can also be used directly for image editing.

In fact, if the edge magnitude and blur estimate are kept along with each edge, a visually similar image can be reconstructed from this information (Elder 1999). Based on this principle, Elder and Goldberg (2001) propose a system for "image editing in the contour domain". Their system allows users to selectively remove edges corresponding to unwanted features such as specularities, shadows, or distracting visual elements. After reconstructing the image from the remaining edges, the undesirable visual features have been removed (Figure 4.39).

Another potential application is to enhance perceptually salient edges while simplifying the underlying image to produce a cartoon-like or "pen-and-ink" stylized image (DeCarlo and Santella 2002). This application is discussed in more detail in Section 10.5.2.

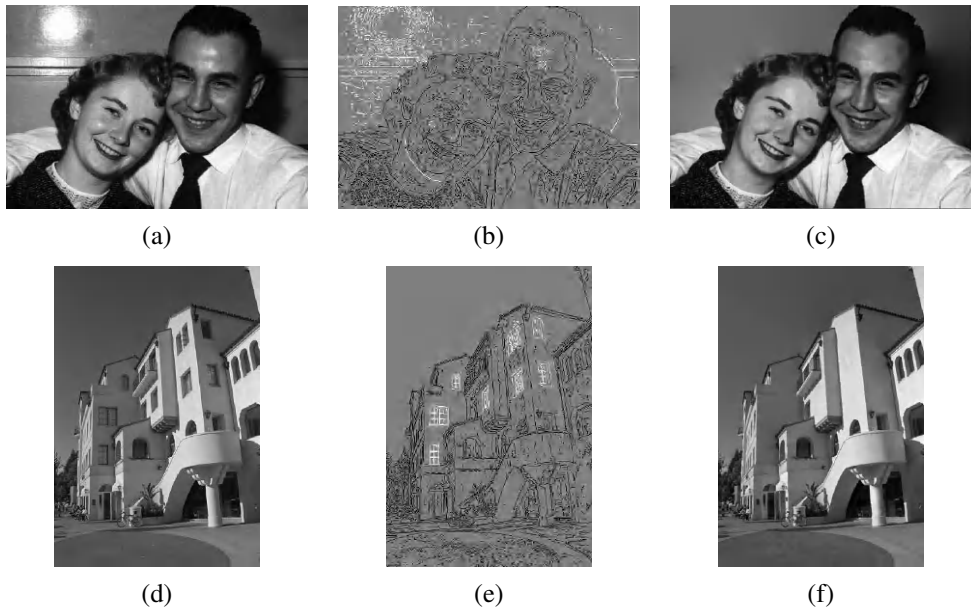


Figure 4.39 Image editing in the contour domain (Elder and Goldberg 2001) © 2001 IEEE: (a) and (d) original images; (b) and (e) extracted edges (edges to be deleted are marked in white); (c) and (f) reconstructed edited images.

4.3 Lines

While edges and general curves are suitable for describing the contours of natural objects, the man-made world is full of straight lines. Detecting and matching these lines can be useful in a variety of applications, including architectural modeling, pose estimation in urban environments, and the analysis of printed document layouts.

In this section, we present some techniques for extracting *piecewise linear* descriptions from the curves computed in the previous section. We begin with some algorithms for approximating a curve as a piecewise-linear polyline. We then describe the *Hough transform*, which can be used to group edgels into line segments even across gaps and occlusions. Finally, we describe how 3D lines with common *vanishing points* can be grouped together. These vanishing points can be used to calibrate a camera and to determine its orientation relative to a rectahedral scene, as described in Section 6.3.2.

4.3.1 Successive approximation

As we saw in Section 4.2.2, describing a curve as a series of 2D locations $\boldsymbol{x}_i = \boldsymbol{x}(s_i)$ provides a general representation suitable for matching and further processing. In many applications, however, it is preferable to approximate such a curve with a simpler representation, e.g., as a piecewise-linear polyline or as a B-spline curve (Farin 1996), as shown in Figure 4.40.

Many techniques have been developed over the years to perform this approximation, which is also known as *line simplification*. One of the oldest, and simplest, is the one proposed by Ramer (1972) and Douglas and Peucker (1973), who recursively subdivide the curve at

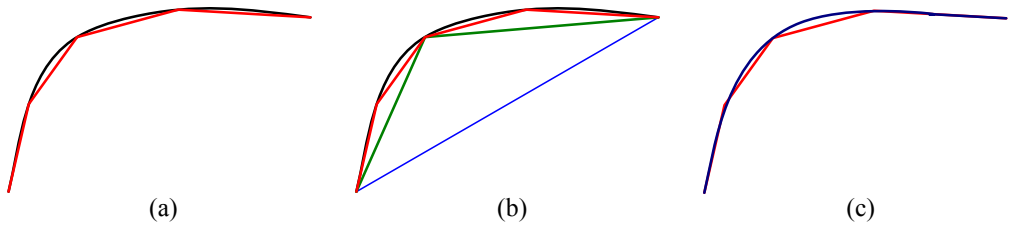


Figure 4.40 Approximating a curve (shown in black) as a polyline or B-spline: (a) original curve and a polyline approximation shown in red; (b) successive approximation by recursively finding points furthest away from the current approximation; (c) smooth interpolating spline, shown in dark blue, fit to the polyline vertices.

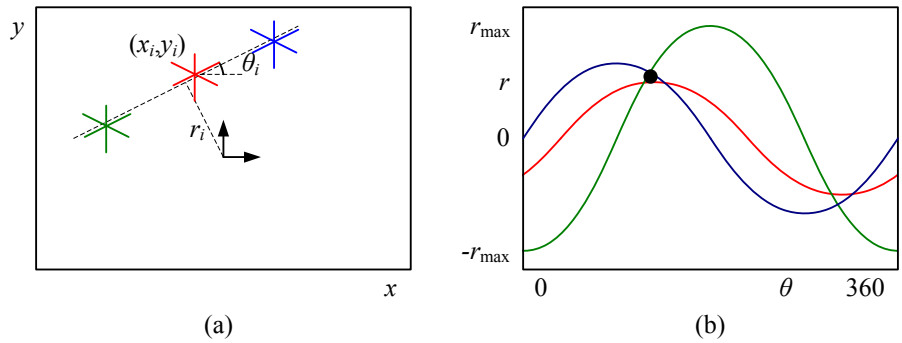


Figure 4.41 Original Hough transform: (a) each point votes for a complete family of potential lines $r_i(\theta) = x_i \cos\theta + y_i \sin\theta$; (b) each pencil of lines sweeps out a sinusoid in (r, θ); their intersection provides the desired line equation.

the point furthest away from the line joining the two endpoints (or the current coarse polyline approximation), as shown in Figure 4.40. Hershberger and Snoeyink (1992) provide a more efficient implementation and also cite some of the other related work in this area.

Once the line simplification has been computed, it can be used to approximate the original curve. If a smoother representation or visualization is desired, either approximating or interpolating splines or curves can be used (Sections 3.5.1 and 5.1.1) (Szeliski and Ito 1986; Bartels, Beatty, and Barsky 1987; Farin 1996), as shown in Figure 4.40c.

4.3.2 Hough transforms

While curve approximation with polylines can often lead to successful line extraction, lines in the real world are sometimes broken up into disconnected components or made up of many collinear line segments. In many cases, it is desirable to group such collinear segments into extended lines. At a further processing stage (described in Section 4.3.3), we can then group such lines into collections with common vanishing points.

The Hough transform, named after its original inventor (Hough 1962), is a well-known technique for having edges "vote" for plausible line locations (Duda and Hart 1972; Ballard 1981; Illingworth and Kittler 1988). In its original formulation (Figure 4.41), each edge point votes for *all* possible lines passing through it, and lines corresponding to high *accumulator* or

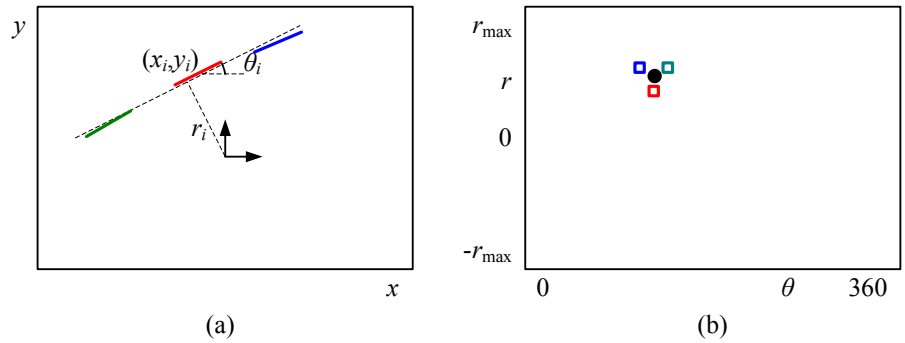


Figure 4.42 Oriented Hough transform: (a) an edgel re-parameterized in polar (r, θ) coordinates, with $\hat{\boldsymbol{n}}_i = (\cos\theta_i, \sin\theta_i)$ and $r_i = \hat{\boldsymbol{n}}_i \cdot \boldsymbol{x}_i$; (b) (r, θ) accumulator array, showing the votes for the three edgels marked in red, green, and blue.

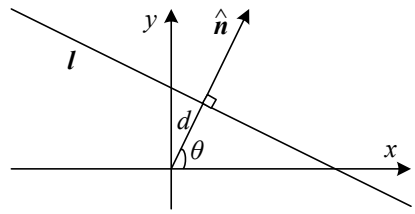


Figure 4.43 2D line equation expressed in terms of the normal $\hat{\boldsymbol{n}}$ and distance to the origin d.

bin values are examined for potential line fits.[9] Unless the points on a line are truly punctate, a better approach (in my experience) is to use the local orientation information at each edgel to vote for a *single* accumulator cell (Figure 4.42), as described below. A hybrid strategy, where each edgel votes for a number of possible orientation or location pairs centered around the estimate orientation, may be desirable in some cases.

Before we can vote for line hypotheses, we must first choose a suitable representation. Figure 4.43 (copied from Figure 2.2a) shows the normal-distance $(\hat{\boldsymbol{n}}, d)$ parameterization for a line. Since lines are made up of edge segments, we adopt the convention that the line normal $\hat{\boldsymbol{n}}$ points in the same direction (i.e., has the same sign) as the image gradient $\boldsymbol{J}(\boldsymbol{x}) = \nabla I(\boldsymbol{x})$ (4.19). To obtain a minimal two-parameter representation for lines, we convert the normal vector into an angle

$$\theta = \tan^{-1} n_y / n_x, \tag{4.26}$$

as shown in Figure 4.43. The range of possible (θ, d) values is $[-180°, 180°] \times [-\sqrt{2}, \sqrt{2}]$, assuming that we are using normalized pixel coordinates (2.61) that lie in $[-1, 1]$. The number of bins to use along each axis depends on the accuracy of the position and orientation estimate available at each edgel and the expected line density, and is best set experimentally with some test runs on sample imagery.

Given the line parameterization, the Hough transform proceeds as shown in Algorithm 4.2.

[9] The Hough transform can also be *generalized* to look for other geometric features such as circles (Ballard 1981), but we do not cover such extensions in this book.

procedure *Hough*($\{(x, y, \theta)\}$):

1. Clear the accumulator array.
2. For each detected edgel at location (x, y) and orientation $\theta = \tan^{-1} n_y/n_x$, compute the value of
$$d = x\, n_x + y\, n_y$$
and increment the accumulator corresponding to (θ, d).
3. Find the peaks in the accumulator corresponding to lines.
4. Optionally re-fit the lines to the constituent edgels.

Algorithm 4.2 Outline of a Hough transform algorithm based on oriented edge segments.

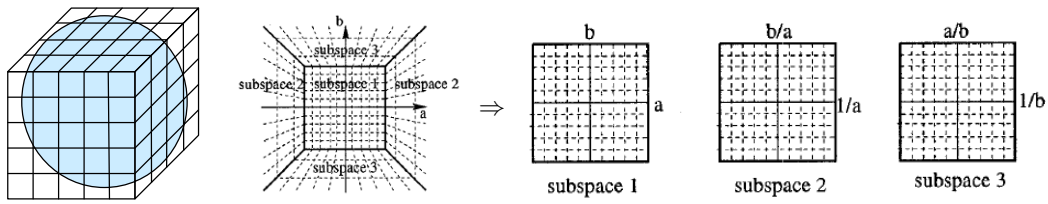


Figure 4.44 Cube map representation for line equations and vanishing points: (a) a cube map surrounding the unit sphere; (b) projecting the half-cube onto three subspaces (Tuytelaars, Van Gool, and Proesmans 1997) © 1997 IEEE.

Note that the original formulation of the Hough transform, which assumed no knowledge of the edgel orientation θ, has an additional loop inside Step 2 that iterates over all possible values of θ and increments a whole series of accumulators.

There are a lot of details in getting the Hough transform to work well, but these are best worked out by writing an implementation and testing it out on sample data. Exercise 4.12 describes some of these steps in more detail, including using edge segment lengths or strengths during the voting process, keeping a list of constituent edgels in the accumulator array for easier post-processing, and optionally combining edges of different "polarity" into the same line segments.

An alternative to the 2D polar (θ, d) representation for lines is to use the full 3D $\boldsymbol{m} = (\hat{\boldsymbol{n}}, d)$ line equation, projected onto the unit sphere. While the sphere can be parameterized using spherical coordinates (2.8),

$$\hat{\boldsymbol{m}} = (\cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi), \tag{4.27}$$

this does not uniformly sample the sphere and still requires the use of trigonometry.

An alternative representation can be obtained by using a *cube map*, i.e., projecting $\boldsymbol{m}$ onto the face of a unit cube (Figure 4.44a). To compute the cube map coordinate of a 3D vector $\boldsymbol{m}$, first find the largest (absolute value) component of $\boldsymbol{m}$, i.e., $m = \pm\max(|n_x|, |n_y|, |d|)$,

and use this to select one of the six cube faces. Divide the remaining two coordinates by m and use these as indices into the cube face. While this avoids the use of trigonometry, it does require some decision logic.

One advantage of using the cube map, first pointed out by Tuytelaars, Van Gool, and Proesmans (1997), is that all of the lines passing through a point correspond to line segments on the cube faces, which is useful if the original (full voting) variant of the Hough transform is being used. In their work, they represent the line equation as $ax + b + y = 0$, which does not treat the x and y axes symmetrically. Note that if we restrict $d \geq 0$ by ignoring the polarity of the edge orientation (gradient sign), we can use a half-cube instead, which can be represented using only three cube faces, as shown in Figure 4.44b (Tuytelaars, Van Gool, and Proesmans 1997).

RANSAC-based line detection. Another alternative to the Hough transform is the RANdom SAmple Consensus (RANSAC) algorithm described in more detail in Section 6.1.4. In brief, RANSAC randomly chooses pairs of edgels to form a line hypothesis and then tests how many other edgels fall onto this line. (If the edge orientations are accurate enough, a single edgel can produce this hypothesis.) Lines with sufficiently large numbers of *inliers* (matching edgels) are then selected as the desired line segments.

An advantage of RANSAC is that no accumulator array is needed and so the algorithm can be more space efficient and potentially less prone to the choice of bin size. The disadvantage is that many more hypotheses may need to be generated and tested than those obtained by finding peaks in the accumulator array.

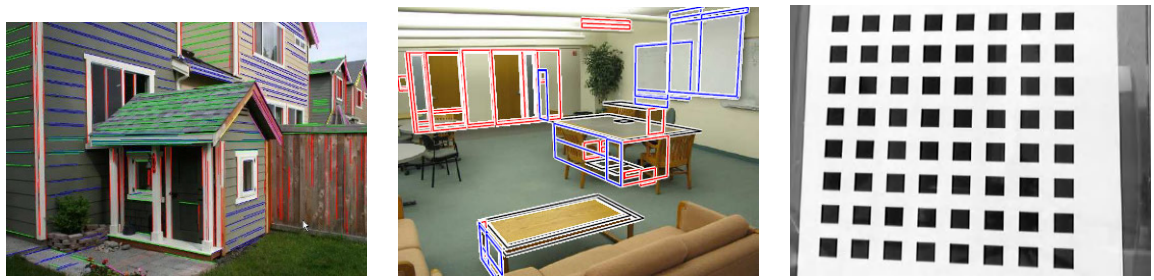
In general, there is no clear consensus on which line estimation technique performs best. It is therefore a good idea to think carefully about the problem at hand and to implement several approaches (successive approximation, Hough, and RANSAC) to determine the one that works best for your application.

4.3.3 Vanishing points

In many scenes, structurally important lines have the same vanishing point because they are parallel in 3D. Examples of such lines are horizontal and vertical building edges, zebra crossings, railway tracks, the edges of furniture such as tables and dressers, and of course, the ubiquitous calibration pattern (Figure 4.45). Finding the vanishing points common to such line sets can help refine their position in the image and, in certain cases, help determine the intrinsic and extrinsic orientation of the camera (Section 6.3.2).

Over the years, a large number of techniques have been developed for finding vanishing points, including (Quan and Mohr 1989; Collins and Weiss 1990; Brillaut-O'Mahoney 1991; McLean and Kotturi 1995; Becker and Bove 1995; Shufelt 1999; Tuytelaars, Van Gool, and Proesmans 1997; Schaffalitzky and Zisserman 2000; Antone and Teller 2002; Rother 2002; Košecká and Zhang 2005; Pflugfelder 2008; Tardif 2009)—see some of the more recent papers for additional references. In this section, we present a simple Hough technique based on having line pairs vote for potential vanishing point locations, followed by a robust least squares fitting stage. For alternative approaches, please see some of the more recent papers listed above.

The first stage in my vanishing point detection algorithm uses a Hough transform to accumulate votes for likely vanishing point candidates. As with line fitting, one possible approach



(a) (b) (c)

Figure 4.45 Real-world vanishing points: (a) architecture (Sinha, Steedly, Szeliski *et al.* 2008), (b) furniture (Mičušìk, Wildenauer, and Košecká 2008) © 2008 IEEE, and (c) calibration patterns (Zhang 2000).

is to have each line vote for *all* possible vanishing point directions, either using a cube map (Tuytelaars, Van Gool, and Proesmans 1997; Antone and Teller 2002) or a Gaussian sphere (Collins and Weiss 1990), optionally using knowledge about the uncertainty in the vanishing point location to perform a weighted vote (Collins and Weiss 1990; Brillaut-O'Mahoney 1991; Shufelt 1999). My preferred approach is to use pairs of detected line segments to form candidate vanishing point locations. Let $\hat{\boldsymbol{m}}_i$ and $\hat{\boldsymbol{m}}_j$ be the (unit norm) line equations for a pair of line segments and l_i and l_j be their corresponding segment lengths. The location of the corresponding vanishing point hypothesis can be computed as

$$\boldsymbol{v}_{ij} = \hat{\boldsymbol{m}}_i \times \hat{\boldsymbol{m}}_j \tag{4.28}$$

and the corresponding weight set to

$$w_{ij} = \|\boldsymbol{v}_{ij}\| l_i l_j. \tag{4.29}$$

This has the desirable effect of downweighting (near-)collinear line segments and short line segments. The Hough space itself can either be represented using spherical coordinates (4.27) or as a cube map (Figure 4.44a).

Once the Hough accumulator space has been populated, peaks can be detected in a manner similar to that previously discussed for line detection. Given a set of candidate line segments that voted for a vanishing point, which can optionally be kept as a list at each Hough accumulator cell, I then use a robust least squares fit to estimate a more accurate location for each vanishing point.

Consider the relationship between the two line segment endpoints $\{\boldsymbol{p}_{i0}, \boldsymbol{p}_{i1}\}$ and the vanishing point $\boldsymbol{v}$, as shown in Figure 4.46. The area A of the triangle given by these three points, which is the magnitude of their triple product

$$A_i = |(\boldsymbol{p}_{i0} \times \boldsymbol{p}_{i1}) \cdot \boldsymbol{v}|, \tag{4.30}$$

is proportional to the perpendicular distance d_1 between each endpoint and the line through $\boldsymbol{v}$ and the other endpoint, as well as the distance between $\boldsymbol{p}_{i0}$ and $\boldsymbol{v}$. Assuming that the accuracy of a fitted line segment is proportional to its endpoint accuracy (Exercise 4.13), this therefore serves as an optimal metric for how well a vanishing point fits a set of extracted lines (Leibowitz (2001, Section 3.6.1) and Pflugfelder (2008, Section 2.1.1.3)). A robustified

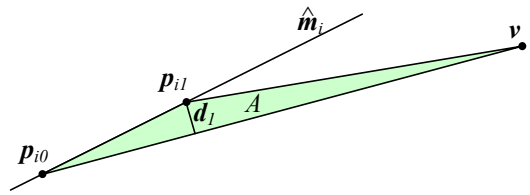


Figure 4.46 Triple product of the line segments endpoints p_{i0} and p_{i1} and the vanishing point v. The area A is proportional to the perpendicular distance d_1 and the distance between the other endpoint p_{i0} and the vanishing point.

least squares estimate (Appendix B.3) for the vanishing point can therefore be written as

$$\mathcal{E} = \sum_i \rho(A_i) = \boldsymbol{v}^T \left(\sum_i w_i(A_i) \boldsymbol{m}_i \boldsymbol{m}_i^T \right) \boldsymbol{v} = \boldsymbol{v}^T \boldsymbol{M} \boldsymbol{v}, \tag{4.31}$$

where $\boldsymbol{m}_i = \boldsymbol{p}_{i0} \times \boldsymbol{p}_{i1}$ is the segment line equation weighted by its length l_i, and $w_i = \rho'(A_i)/A_i$ is the *influence* of each robustified (reweighted) measurement on the final error (Appendix B.3). Notice how this metric is closely related to the original formula for the pairwise weighted Hough transform accumulation step. The final desired value for $\boldsymbol{v}$ is computed as the least eigenvector of $\boldsymbol{M}$.

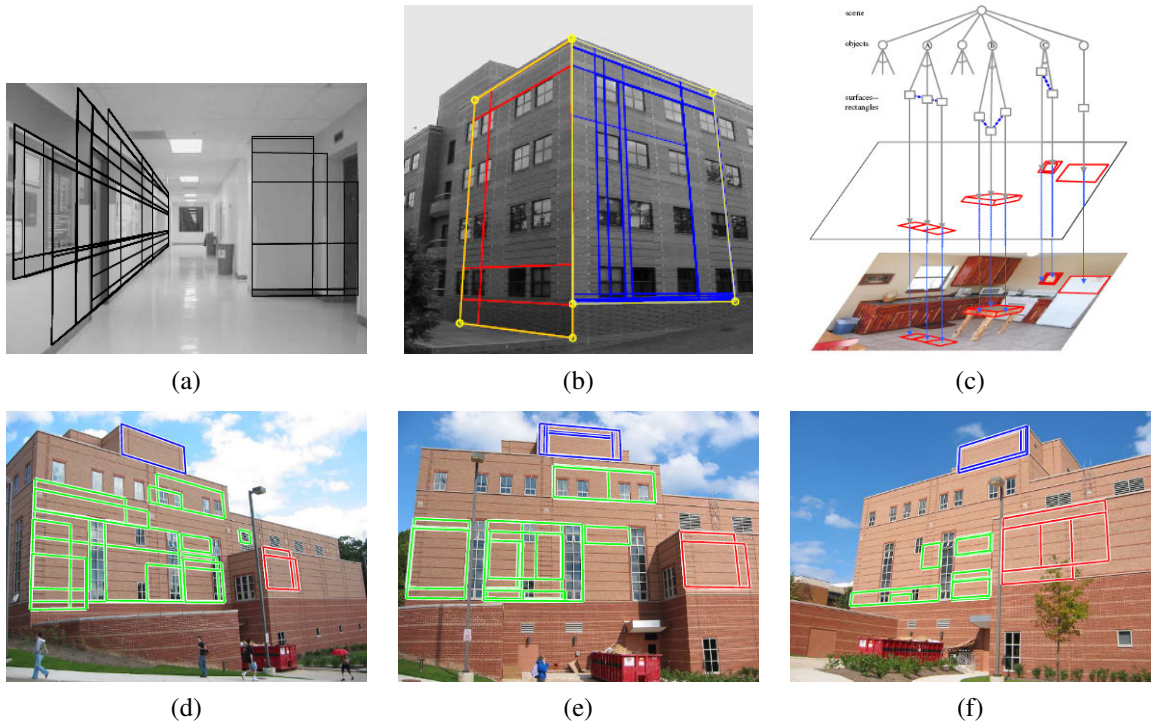
While the technique described above proceeds in two discrete stages, better results may be obtained by alternating between assigning lines to vanishing points and refitting the vanishing point locations (Antone and Teller 2002; Košecká and Zhang 2005; Pflugfelder 2008). The results of detecting individual vanishing points can also be made more robust by simultaneously searching for pairs or triplets of mutually orthogonal vanishing points (Shufelt 1999; Antone and Teller 2002; Rother 2002; Sinha, Steedly, Szeliski *et al.* 2008). Some results of such vanishing point detection algorithms can be seen in Figure 4.45.

4.3.4 *Application*: Rectangle detection

Once sets of mutually orthogonal vanishing points have been detected, it now becomes possible to search for 3D rectangular structures in the image (Figure 4.47). Over the last decade, a variety of techniques have been developed to find such rectangles, primarily focused on architectural scenes (Košecká and Zhang 2005; Han and Zhu 2005; Shaw and Barnes 2006; Mičušìk, Wildenauer, and Košecká 2008; Schindler, Krishnamurthy, Lublinerman *et al.* 2008).

After detecting orthogonal vanishing directions, Košecká and Zhang (2005) refine the fitted line equations, search for corners near line intersections, and then verify rectangle hypotheses by rectifying the corresponding patches and looking for a preponderance of horizontal and vertical edges (Figures 4.47a–b). In follow-on work, Mičušìk, Wildenauer, and Košecká (2008) use a Markov random field (MRF) to disambiguate between potentially overlapping rectangle hypotheses. They also use a plane sweep algorithm to match rectangles between different views (Figures 4.47d–f).

A different approach is proposed by Han and Zhu (2005), who use a grammar of potential rectangle shapes and nesting structures (between rectangles and vanishing points) to infer the most likely assignment of line segments to rectangles (Figure 4.47c).



(a) (b) (c)

(d) (e) (f)

Figure 4.47 Rectangle detection: (a) indoor corridor and (b) building exterior with grouped facades (Košecká and Zhang 2005) © 2005 Elsevier; (c) grammar-based recognition (Han and Zhu 2005) © 2005 IEEE; (d–f) rectangle matching using a plane sweep algorithm (Mičušìk, Wildenauer, and Košecká 2008) © 2008 IEEE.

4.4 Additional reading

One of the seminal papers on feature detection, description, and matching is by Lowe (2004). Comprehensive surveys and evaluations of such techniques have been made by Schmid, Mohr, and Bauckhage (2000); Mikolajczyk and Schmid (2005); Mikolajczyk, Tuytelaars, Schmid *et al.* (2005); Tuytelaars and Mikolajczyk (2007) while Shi and Tomasi (1994) and Triggs (2004) also provide nice reviews.

In the area of feature detectors (Mikolajczyk, Tuytelaars, Schmid *et al.* 2005), in addition to such classic approaches as Förstner–Harris (Förstner 1986; Harris and Stephens 1988) and difference of Gaussians (Lindeberg 1993, 1998b; Lowe 2004), maximally stable extremal regions (MSERs) are widely used for applications that require affine invariance (Matas, Chum, Urban *et al.* 2004; Nistér and Stewénius 2008). More recent interest point detectors are discussed by Xiao and Shah (2003); Koethe (2003); Carneiro and Jepson (2005); Kenney, Zuliani, and Manjunath (2005); Bay, Tuytelaars, and Van Gool (2006); Platel, Balmachnova, Florack *et al.* (2006); Rosten and Drummond (2006), as well as techniques based on line matching (Zoghlami, Faugeras, and Deriche 1997; Bartoli, Coquerelle, and Sturm 2004) and region detection (Kadir, Zisserman, and Brady 2004; Matas, Chum, Urban *et al.* 2004; Tuytelaars and Van Gool 2004; Corso and Hager 2005).

A variety of local feature descriptors (and matching heuristics) are surveyed and compared by Mikolajczyk and Schmid (2005). More recent publications in this area include

those by van de Weijer and Schmid (2006); Abdel-Hakim and Farag (2006); Winder and Brown (2007); Hua, Brown, and Winder (2007). Techniques for efficiently matching features include k-d trees (Beis and Lowe 1999; Lowe 2004; Muja and Lowe 2009), pyramid matching kernels (Grauman and Darrell 2005), metric (vocabulary) trees (Nistér and Stewénius 2006), and a variety of multi-dimensional hashing techniques (Shakhnarovich, Viola, and Darrell 2003; Torralba, Weiss, and Fergus 2008; Weiss, Torralba, and Fergus 2008; Kulis and Grauman 2009; Raginsky and Lazebnik 2009).

The classic reference on feature detection and tracking is (Shi and Tomasi 1994). More recent work in this field has focused on learning better matching functions for specific features (Avidan 2001; Jurie and Dhome 2002; Williams, Blake, and Cipolla 2003; Lepetit and Fua 2005; Lepetit, Pilet, and Fua 2006; Hinterstoisser, Benhimane, Navab *et al.* 2008; Rogez, Rihan, Ramalingam *et al.* 2008; Özuysal, Calonder, Lepetit *et al.* 2010).

A highly cited and widely used edge detector is the one developed by Canny (1986). Alternative edge detectors as well as experimental comparisons can be found in publications by Nalwa and Binford (1986); Nalwa (1987); Deriche (1987); Freeman and Adelson (1991); Nalwa (1993); Heath, Sarkar, Sanocki *et al.* (1998); Crane (1997); Ritter and Wilson (2000); Bowyer, Kranenburg, and Dougherty (2001); Arbeláez, Maire, Fowlkes *et al.* (2010). The topic of scale selection in edge detection is nicely treated by Elder and Zucker (1998), while approaches to color and texture edge detection can be found in (Ruzon and Tomasi 2001; Martin, Fowlkes, and Malik 2004; Gevers, van de Weijer, and Stokman 2006). Edge detectors have also recently been combined with region segmentation techniques to further improve the detection of semantically salient boundaries (Maire, Arbelaez, Fowlkes *et al.* 2008; Arbeláez, Maire, Fowlkes *et al.* 2010). Edges linked into contours can be smoothed and manipulated for artistic effect (Lowe 1989; Finkelstein and Salesin 1994; Taubin 1995) and used for recognition (Belongie, Malik, and Puzicha 2002; Tek and Kimia 2003; Sebastian and Kimia 2005).

An early, well-regarded paper on straight line extraction in images was written by Burns, Hanson, and Riseman (1986). More recent techniques often combine line detection with vanishing point detection (Quan and Mohr 1989; Collins and Weiss 1990; Brillaut-O'Mahoney 1991; McLean and Kotturi 1995; Becker and Bove 1995; Shufelt 1999; Tuytelaars, Van Gool, and Proesmans 1997; Schaffalitzky and Zisserman 2000; Antone and Teller 2002; Rother 2002; Košecká and Zhang 2005; Pflugfelder 2008; Sinha, Steedly, Szeliski *et al.* 2008; Tardif 2009).

4.5 Exercises

Ex 4.1: Interest point detector Implement one or more keypoint detectors and compare their performance (with your own or with a classmate's detector).

Possible detectors:

- Laplacian or Difference of Gaussian;
- Förstner–Harris Hessian (try different formula variants given in (4.9–4.11));
- oriented/steerable filter, looking for either second-order high second response or two edges in a window (Koethe 2003), as discussed in Section 4.1.1.

Other detectors are described by Mikolajczyk, Tuytelaars, Schmid *et al.* (2005); Tuytelaars and Mikolajczyk (2007). Additional optional steps could include:

1. Compute the detections on a sub-octave pyramid and find 3D maxima.
2. Find local orientation estimates using steerable filter responses or a gradient histogramming method.
3. Implement non-maximal suppression, such as the adaptive technique of Brown, Szeliski, and Winder (2005).
4. Vary the window shape and size (pre-filter and aggregation).

To test for repeatability, download the code from http://www.robots.ox.ac.uk/~vgg/research/affine/ (Mikolajczyk, Tuytelaars, Schmid *et al.* 2005; Tuytelaars and Mikolajczyk 2007) or simply rotate or shear your own test images. (Pick a domain you may want to use later, e.g., for outdoor stitching.)

Be sure to measure and report the stability of your scale and orientation estimates.

Ex 4.2: Interest point descriptor Implement one or more descriptors (steered to local scale and orientation) and compare their performance (with your own or with a classmate's detector).

Some possible descriptors include

- contrast-normalized patches (Brown, Szeliski, and Winder 2005);
- SIFT (Lowe 2004);
- GLOH (Mikolajczyk and Schmid 2005);
- DAISY (Winder and Brown 2007; Tola, Lepetit, and Fua 2010).

Other detectors are described by Mikolajczyk and Schmid (2005).

Ex 4.3: ROC curve computation Given a pair of curves (histograms) plotting the number of matching and non-matching features as a function of Euclidean distance d as shown in Figure 4.23b, derive an algorithm for plotting a ROC curve (Figure 4.23a). In particular, let $t(d)$ be the distribution of true matches and $f(d)$ be the distribution of (false) non-matches. Write down the equations for the ROC, i.e., TPR(FPR), and the AUC.

(Hint: Plot the cumulative distributions $T(d) = \int t(d)$ and $F(d) = \int f(d)$ and see if these help you derive the TPR and FPR at a given threshold θ.)

Ex 4.4: Feature matcher After extracting features from a collection of overlapping or distorted images,[10] match them up by their descriptors either using nearest neighbor matching or a more efficient matching strategy such as a k-d tree.

See whether you can improve the accuracy of your matches using techniques such as the nearest neighbor distance ratio.

[10] http://www.robots.ox.ac.uk/~vgg/research/affine/.

Ex 4.5: Feature tracker Instead of finding feature points independently in multiple images and then matching them, find features in the first image of a video or image sequence and then re-locate the corresponding points in the next frames using either search and gradient descent (Shi and Tomasi 1994) or learned feature detectors (Lepetit, Pilet, and Fua 2006; Fossati, Dimitrijevic, Lepetit *et al.* 2007). When the number of tracked points drops below a threshold or new regions in the image become visible, find additional points to track.

(Optional) Winnow out incorrect matches by estimating a homography (6.19–6.23) or fundamental matrix (Section 7.2.1).

(Optional) Refine the accuracy of your matches using the iterative registration algorithm described in Section 8.2 and Exercise 8.2.

Ex 4.6: Facial feature tracker Apply your feature tracker to tracking points on a person's face, either manually initialized to interesting locations such as eye corners or automatically initialized at interest points.

(Optional) Match features between two people and use these features to perform image morphing (Exercise 3.25).

Ex 4.7: Edge detector Implement an edge detector of your choice. Compare its performance to that of your classmates' detectors or code downloaded from the Internet.

A simple but well-performing sub-pixel edge detector can be created as follows:

1. Blur the input image a little,
$$B_\sigma(\boldsymbol{x}) = G_\sigma(\boldsymbol{x}) * I(\boldsymbol{x}).$$
2. Construct a Gaussian pyramid (Exercise 3.19),
$$P = \text{Pyramid}\{B_\sigma(\boldsymbol{x})\}$$
3. Subtract an interpolated coarser-level pyramid image from the original resolution blurred image,
$$S(\boldsymbol{x}) = B_\sigma(\boldsymbol{x}) - P.\text{InterpolatedLevel}(L).$$
4. For each quad of pixels, $\{(i,j),(i+1,j),(i,j+1),(i+1,j+1)\}$, count the number of zero crossings along the four edges.
5. When there are exactly two zero crossings, compute their locations using (4.25) and store these edgel endpoints along with the midpoint in the edgel structure (Figure 4.48).
6. For each edgel, compute the local gradient by taking the horizontal and vertical differences between the values of S along the zero crossing edges.
7. Store the magnitude of this gradient as the edge strength and either its orientation or that of the segment joining the edgel endpoints as the edge orientation.
8. Add the edgel to a list of edgels or store it in a 2D array of edgels (addressed by pixel coordinates).

Figure 4.48 shows a possible representation for each computed edgel.

```
struct SEdgel {
    float e[2][2];      // edgel endpoints (zero crossing)
    float x, y;         // sub-pixel edge position (midpoint)
    float n_x, n_y;     // orientation, as normal vector
    float theta;        // orientation, as angle (degrees)
    float length;       // length of edgel
    float strength;     // strength of edgel (gradient magnitude)
};

struct SLine : public SEdgel {
    float line_length;  // length of line (est. from ellipsoid)
    float sigma;        // estimated std. dev. of edgel noise
    float r;            // line equation: x * n_y - y * n_x = r
};
```

Figure 4.48 A potential C++ structure for edgel and line elements.

Ex 4.8: Edge linking and thresholding Link up the edges computed in the previous exercise into chains and optionally perform thresholding with hysteresis.

The steps may include:

1. Store the edgels either in a 2D array (say, an integer image with indices into the edgel list) or pre-sort the edgel list first by (integer) x coordinates and then y coordinates, for faster neighbor finding.

2. Pick up an edgel from the list of unlinked edgels and find its neighbors in both directions until no neighbor is found or a closed contour is obtained. Flag edgels as linked as you visit them and push them onto your list of linked edgels.

3. Alternatively, generalize a previously developed connected component algorithm (Exercise 3.14) to perform the linking in just two raster passes.

4. (Optional) Perform hysteresis-based thresholding (Canny 1986). Use two thresholds "hi" and "lo" for the edge strength. A candidate edgel is considered an edge if either its strength is above the "hi" threshold or its strength is above the "lo" threshold and it is (recursively) connected to a previously detected edge.

5. (Optional) Link together contours that have small gaps but whose endpoints have similar orientations.

6. (Optional) Find junctions between adjacent contours, e.g., using some of the ideas (or references) from Maire, Arbelaez, Fowlkes *et al.* (2008).

Ex 4.9: Contour matching Convert a closed contour (linked edgel list) into its arc-length parameterization and use this to match object outlines.

The steps may include:

1. Walk along the contour and create a list of (x_i, y_i, s_i) triplets, using the arc-length formula
$$s_{i+1} = s_i + \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|. \tag{4.32}$$
2. Resample this list onto a regular set of (x_j, y_j, j) samples using linear interpolation of each segment.
3. Compute the average values of x and y, i.e., $\overline{x}$ and $\overline{y}$ and subtract them from your sampled curve points.
4. Resample the original (x_i, y_i, s_i) piecewise-linear function onto a length-independent set of samples, say $j \in [0, 1023]$. (Using a length which is a power of two makes subsequent Fourier transforms more convenient.)
5. Compute the Fourier transform of the curve, treating each (x, y) pair as a complex number.
6. To compare two curves, fit a linear equation to the phase difference between the two curves. (Careful: phase wraps around at $360°$. Also, you may wish to weight samples by their Fourier spectrum magnitude—see Section 8.1.2.)
7. (Optional) Prove that the constant phase component corresponds to the temporal shift in s, while the linear component corresponds to rotation.

Of course, feel free to try any other curve descriptor and matching technique from the computer vision literature (Tek and Kimia 2003; Sebastian and Kimia 2005).

Ex 4.10: Jigsaw puzzle solver—challenging Write a program to automatically solve a jigsaw puzzle from a set of scanned puzzle pieces. Your software may include the following components:

1. Scan the pieces (either face up or face down) on a flatbed scanner with a distinctively colored background.
2. (Optional) Scan in the box top to use as a low-resolution reference image.
3. Use color-based thresholding to isolate the pieces.
4. Extract the contour of each piece using edge finding and linking.
5. (Optional) Re-represent each contour using an arc-length or some other re-parameterization. Break up the contours into meaningful matchable pieces. (Is this hard?)
6. (Optional) Associate color values with each contour to help in the matching.
7. (Optional) Match pieces to the reference image using some rotationally invariant feature descriptors.
8. Solve a global optimization or (backtracking) search problem to snap pieces together and place them in the correct location relative to the reference image.
9. Test your algorithm on a succession of more difficult puzzles and compare your results with those of others.

Ex 4.11: Successive approximation line detector Implement a line simplification algorithm (Section 4.3.1) (Ramer 1972; Douglas and Peucker 1973) to convert a hand-drawn curve (or linked edge image) into a small set of polylines.

(Optional) Re-render this curve using either an approximating or interpolating spline or Bezier curve (Szeliski and Ito 1986; Bartels, Beatty, and Barsky 1987; Farin 1996).

Ex 4.12: Hough transform line detector Implement a Hough transform for finding lines in images:

1. Create an accumulator array of the appropriate user-specified size and clear it. The user can specify the spacing in degrees between orientation bins and in pixels between distance bins. The array can be allocated as integer (for simple counts), floating point (for weighted counts), or as an array of vectors for keeping back pointers to the constituent edges.

2. For each detected edgel at location (x, y) and orientation $\theta = \tan^{-1} n_y/n_x$, compute the value of
$$d = xn_x + yn_y \tag{4.33}$$
and increment the accumulator corresponding to (θ, d).

   (Optional) Weight the vote of each edge by its length (see Exercise 4.7) or the strength of its gradient.

3. (Optional) Smooth the scalar accumulator array by adding in values from its immediate neighbors. This can help counteract the *discretization* effect of voting for only a single bin—see Exercise 3.7.

4. Find the largest peaks (local maxima) in the accumulator corresponding to lines.

5. (Optional) For each peak, re-fit the lines to the constituent edgels, using *total least squares* (Appendix A.2). Use the original edgel lengths or strength weights to weight the least squares fit, as well as the agreement between the hypothesized line orientation and the edgel orientation. Determine whether these heuristics help increase the accuracy of the fit.

6. After fitting each peak, zero-out or eliminate that peak and its adjacent bins in the array, and move on to the next largest peak.

Test out your Hough transform on a variety of images taken indoors and outdoors, as well as checkerboard calibration patterns.

For checkerboard patterns, you can modify your Hough transform by collapsing *antipodal* bins $(\theta \pm 180°, -d)$ with (θ, d) to find lines that do not care about polarity changes. Can you think of examples in real-world images where this might be desirable as well?

Ex 4.13: Line fitting uncertainty Estimate the uncertainty (covariance) in your line fit using uncertainty analysis.

1. After determining which edgels belong to the line segment (using either successive approximation or Hough transform), re-fit the line segment using total least squares (Van Huffel and Vandewalle 1991; Van Huffel and Lemmerling 2002), i.e., find the

mean or centroid of the edgels and then use eigenvalue analysis to find the dominant orientation.

2. Compute the perpendicular errors (deviations) to the line and robustly estimate the variance of the fitting noise using an estimator such as MAD (Appendix B.3).

3. (Optional) re-fit the line parameters by throwing away outliers or using a robust norm or influence function.

4. Estimate the error in the perpendicular location of the line segment and its orientation.

Ex 4.14: Vanishing points Compute the vanishing points in an image using one of the techniques described in Section 4.3.3 and optionally refine the original line equations associated with each vanishing point. Your results can be used later to track a target (Exercise 6.5) or reconstruct architecture (Section 12.6.1).

Ex 4.15: Vanishing point uncertainty Perform an uncertainty analysis on your estimated vanishing points. You will need to decide how to represent your vanishing point, e.g., homogeneous coordinates on a sphere, to handle vanishing points near infinity.

See the discussion of Bingham distributions by Collins and Weiss (1990) for some ideas.

Chapter 5

Segmentation

5.1 Active contours . . . 237
5.1.1 Snakes . . . 238
5.1.2 Dynamic snakes and CONDENSATION . . . 243
5.1.3 Scissors . . . 246
5.1.4 Level Sets . . . 248
5.1.5 *Application*: Contour tracking and rotoscoping . . . 249
5.2 Split and merge . . . 250
5.2.1 Watershed . . . 251
5.2.2 Region splitting (divisive clustering) . . . 251
5.2.3 Region merging (agglomerative clustering) . . . 251
5.2.4 Graph-based segmentation . . . 252
5.2.5 Probabilistic aggregation . . . 253
5.3 Mean shift and mode finding . . . 254
5.3.1 K-means and mixtures of Gaussians . . . 256
5.3.2 Mean shift . . . 257
5.4 Normalized cuts . . . 260
5.5 Graph cuts and energy-based methods . . . 264
5.5.1 *Application*: Medical image segmentation . . . 268
5.6 Additional reading . . . 268
5.7 Exercises . . . 270

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_5, © Springer-Verlag London Limited 2011



Figure 5.1 Some popular image segmentation techniques: (a) active contours (Isard and Blake 1998) © 1998 Springer; (b) level sets (Cremers, Rousson, and Deriche 2007) © 2007 Springer; (c) graph-based merging (Felzenszwalb and Huttenlocher 2004b) © 2004 Springer; (d) mean shift (Comaniciu and Meer 2002) © 2002 IEEE; (e) texture and intervening contour-based normalized cuts (Malik, Belongie, Leung *et al.* 2001) © 2001 Springer; (f) binary MRF solved using graph cuts (Boykov and Funka-Lea 2006) © 2006 Springer.

Image segmentation is the task of finding groups of pixels that "go together". In statistics, this problem is known as *cluster analysis* and is a widely studied area with hundreds of different algorithms (Jain and Dubes 1988; Kaufman and Rousseeuw 1990; Jain, Duin, and Mao 2000; Jain, Topchy, Law *et al.* 2004).

In computer vision, image segmentation is one of the oldest and most widely studied problems (Brice and Fennema 1970; Pavlidis 1977; Riseman and Arbib 1977; Ohlander, Price, and Reddy 1978; Rosenfeld and Davis 1979; Haralick and Shapiro 1985). Early techniques tend to use region splitting or merging (Brice and Fennema 1970; Horowitz and Pavlidis 1976; Ohlander, Price, and Reddy 1978; Pavlidis and Liow 1990), which correspond to *divisive* and *agglomerative* algorithms in the clustering literature (Jain, Topchy, Law *et al.* 2004). More recent algorithms often optimize some global criterion, such as intra-region consistency and inter-region boundary lengths or dissimilarity (Leclerc 1989; Mumford and Shah 1989; Shi and Malik 2000; Comaniciu and Meer 2002; Felzenszwalb and Huttenlocher 2004b; Cremers, Rousson, and Deriche 2007).

We have already seen examples of image segmentation in Sections 3.3.2 and 3.7.2. In this chapter, we review some additional techniques that have been developed for image segmentation. These include algorithms based on active contours (Section 5.1) and level sets (Section 5.1.4), region splitting and merging (Section 5.2), *mean shift* (mode finding) (Section 5.3), *normalized cuts* (splitting based on pixel similarity metrics) (Section 5.4), and binary Markov random fields solved using graph cuts (Section 5.5). Figure 5.1 shows some examples of these techniques applied to different images.

Since the literature on image segmentation is so vast, a good way to get a handle on some of the better performing algorithms is to look at experimental comparisons on human-labeled databases (Arbeláez, Maire, Fowlkes *et al.* 2010). The best known of these is the Berkeley Segmentation Dataset and Benchmark[1] (Martin, Fowlkes, Tal *et al.* 2001), which consists of 1000 images from a Corel image dataset that were hand-labeled by 30 human subjects. Many of the more recent image segmentation algorithms report comparative results on this database. For example, Unnikrishnan, Pantofaru, and Hebert (2007) propose new metrics for comparing such algorithms. Estrada and Jepson (2009) compare four well-known segmentation algorithms on the Berkeley data set and conclude that while their own SE-MinCut algorithm (Estrada, Jepson, and Chennubhotla 2004) algorithm outperforms the others by a small margin, there still exists a wide gap between automated and human segmentation performance.[2] A new database of foreground and background segmentations, used by Alpert, Galun, Basri *et al.* (2007), is also available.[3]

5.1 Active contours

While lines, vanishing points, and rectangles are commonplace in the man-made world, curves corresponding to object boundaries are even more common, especially in the natural environment. In this section, we describe three related approaches to locating such boundary curves in images.

[1] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

[2] An interesting observation about their ROC plots is that automated techniques cluster tightly along similar curves, but human performance is all over the map.

[3] http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/index.html

The first, originally called *snakes* by its inventors (Kass, Witkin, and Terzopoulos 1988) (Section 5.1.1), is an energy-minimizing, two-dimensional spline curve that evolves (moves) towards image features such as strong edges. The second, *intelligent scissors* (Mortensen and Barrett 1995) (Section 5.1.3), allow the user to sketch in real time a curve that clings to object boundaries. Finally, *level set* techniques (Section 5.1.4) evolve the curve as the zero-set of a *characteristic function*, which allows them to easily change topology and incorporate region-based statistics.

All three of these are examples of *active contours* (Blake and Isard 1998; Mortensen 1999), since these boundary detectors iteratively move towards their final solution under the combination of image and optional user-guidance forces.

5.1.1 Snakes

Snakes are a two-dimensional generalization of the 1D energy-minimizing splines first introduced in Section 3.7.1,

$$\mathcal{E}_{\text{int}} = \int \alpha(s)\|\boldsymbol{f}_s(s)\|^2 + \beta(s)\|\boldsymbol{f}_{ss}(s)\|^2\, ds, \tag{5.1}$$

where s is the arc-length along the curve $\boldsymbol{f}(s) = (x(s), y(s))$ and $\alpha(s)$ and $\beta(s)$ are first- and second-order continuity weighting functions analogous to the $s(x, y)$ and $c(x, y)$ terms introduced in (3.100–3.101). We can discretize this energy by sampling the initial curve position evenly along its length (Figure 4.35) to obtain

$$\begin{aligned} E_{\text{int}} &= \sum_i \alpha(i)\|f(i+1) - f(i)\|^2/h^2 \\ &\quad + \beta(i)\|f(i+1) - 2f(i) + f(i-1)\|^2/h^4, \end{aligned} \tag{5.2}$$

where h is the step size, which can be neglected if we resample the curve along its arc-length after each iteration.

In addition to this *internal* spline energy, a snake simultaneously minimizes external image-based and constraint-based potentials. The image-based potentials are the sum of several terms

$$\mathcal{E}_{\text{image}} = w_{\text{line}}\mathcal{E}_{\text{line}} + w_{\text{edge}}\mathcal{E}_{\text{edge}} + w_{\text{term}}\mathcal{E}_{\text{term}}, \tag{5.3}$$

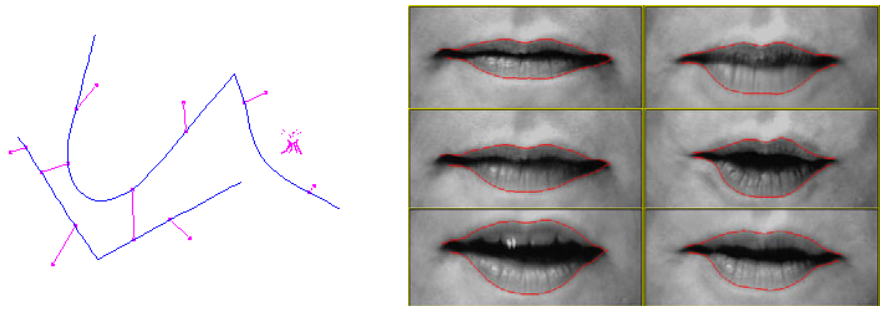
where the *line* term attracts the snake to dark ridges, the *edge* term attracts it to strong gradients (edges), and the *term* term attracts it to line terminations. In practice, most systems only use the edge term, which can either be directly proportional to the image gradients,

$$E_{\text{edge}} = \sum_i -\|\nabla I(\boldsymbol{f}(i))\|^2, \tag{5.4}$$

or to a smoothed version of the image Laplacian,

$$E_{\text{edge}} = \sum_i -|(G_\sigma * \nabla^2 I)(\boldsymbol{f}(i))|^2. \tag{5.5}$$

People also sometimes extract edges and then use a distance map to the edges as an alternative to these two originally proposed potentials.



(a) (b)

Figure 5.2 Snakes (Kass, Witkin, and Terzopoulos 1988) © 1988 Springer: (a) the "snake pit" for interactively controlling shape; (b) lip tracking.

In interactive applications, a variety of user-placed constraints can also be added, e.g., attractive (spring) forces towards anchor points $\boldsymbol{d}(i)$,

$$E_{\text{spring}} = k_i \|\boldsymbol{f}(i) - \boldsymbol{d}(i)\|^2, \tag{5.6}$$

as well as repulsive $1/r$ ("volcano") forces (Figure 5.2a). As the snakes evolve by minimizing their energy, they often "wiggle" and "slither", which accounts for their popular name. Figure 5.2b shows snakes being used to track a person's lips.

Because regular snakes have a tendency to shrink (Exercise 5.1), it is usually better to initialize them by drawing the snake outside the object of interest to be tracked. Alternatively, an expansion *ballooning* force can be added to the dynamics (Cohen and Cohen 1993), essentially moving each point outwards along its normal.

To efficiently solve the sparse linear system arising from snake energy minimization, a sparse direct solver (Appendix A.4) can be used, since the linear system is essentially pentadiagonal.[4] Snake evolution is usually implemented as an alternation between this linear system solution and the linearization of non-linear constraints such as edge energy. A more direct way to find a global energy minimum is to use dynamic programming (Amini, Weymouth, and Jain 1990; Williams and Shah 1992), but this is not often used in practice, since it has been superseded by even more efficient or interactive algorithms such as intelligent scissors (Section 5.1.3) and GrabCut (Section 5.5).

Elastic nets and slippery springs

An interesting variant on snakes, first proposed by Durbin and Willshaw (1987) and later re-formulated in an energy-minimizing framework by Durbin, Szeliski, and Yuille (1989), is the *elastic net* formulation of the Traveling Salesman Problem (TSP). Recall that in a TSP, the salesman must visit each city once while minimizing the total distance traversed. A snake that is constrained to pass through each city could solve this problem (without any optimality guarantees) but it is impossible to tell ahead of time which snake control point should be associated with each city.

[4] A closed snake has a Toeplitz matrix form, which can still be factored and solved in $O(N)$ time.

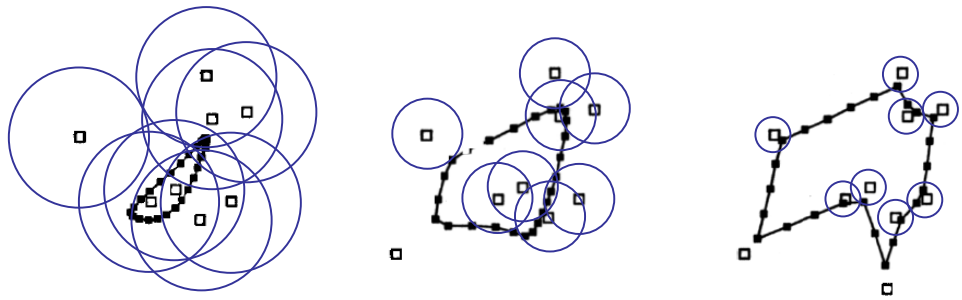


Figure 5.3 Elastic net: The open squares indicate the cities and the closed squares linked by straight line segments are the tour points. The blue circles indicate the approximate extent of the attraction force of each city, which is reduced over time. Under the Bayesian interpretation of the elastic net, the blue circles correspond to one standard deviation of the circular Gaussian that generates each city from some unknown tour point.

Instead of having a fixed constraint between snake nodes and cities, as in (5.6), a city is assumed to pass near *some* point along the tour (Figure 5.3). In a probabilistic interpretation, each city is generated as a *mixture* of Gaussians centered at each tour point,

$$p(\boldsymbol{d}(j)) = \sum_i p_{ij} \quad \text{with} \quad p_{ij} = e^{-d_{ij}^2/(2\sigma^2)} \tag{5.7}$$

where σ is the standard deviation of the Gaussian and

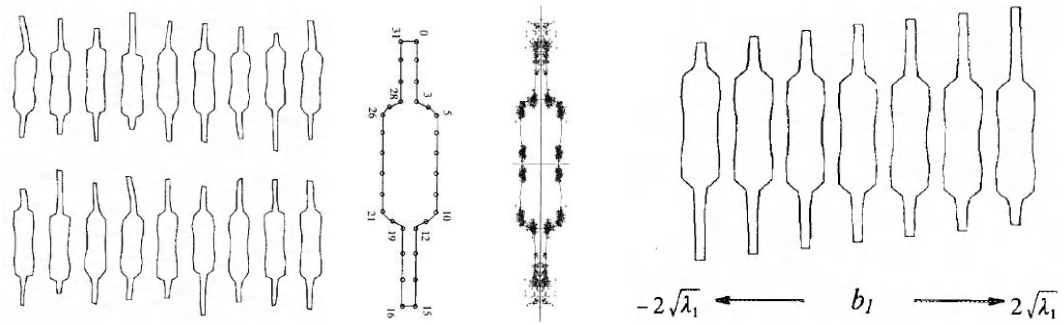
$$d_{ij} = \|\boldsymbol{f}(i) - \boldsymbol{d}(j)\| \tag{5.8}$$

is the Euclidean distance between a tour point $\boldsymbol{f}(i)$ and a city location $\boldsymbol{d}(j)$. The corresponding data fitting energy (negative log likelihood) is

$$E_{\text{slippery}} = -\sum_j \log p(\boldsymbol{d}(j)) = -\sum_j \log \left[\sum e^{-\|\boldsymbol{f}(i)-\boldsymbol{d}(j)\|^2/2\sigma^2}\right]. \tag{5.9}$$

This energy derives its name from the fact that, unlike a regular spring, which couples a given snake point to a given constraint (5.6), this alternative energy defines a *slippery spring* that allows the association between constraints (cities) and curve (tour) points to evolve over time (Szeliski 1989). Note that this is a soft variant of the popular *iterated closest point* data constraint that is often used in fitting or aligning surfaces to data points or to each other (Section 12.2.1) (Besl and McKay 1992; Zhang 1994).

To compute a good solution to the TSP, the slippery spring data association energy is combined with a regular first-order internal smoothness energy (5.3) to define the cost of a tour. The tour $\boldsymbol{f}(s)$ is initialized as a small circle around the mean of the city points and σ is progressively lowered (Figure 5.3). For large σ values, the tour tries to stay near the centroid of the points but as σ decreases each city pulls more and more strongly on its closest tour points (Durbin, Szeliski, and Yuille 1989). In the limit as $\sigma \to 0$, each city is guaranteed to capture at least one tour point and the tours between subsequent cites become straight lines.



(a) (b) (c) (d)

Figure 5.4 Point distribution model for a set of resistors (Cootes, Cooper, Taylor *et al.* 1995) © 1995 Elsevier: (a) set of input resistor shapes; (b) assignment of control points to the boundary; (c) distribution (scatter plot) of point locations; (d) first (largest) mode of variation in the ensemble shapes.

Splines and shape priors

While snakes can be very good at capturing the fine and irregular detail in many real-world contours, they sometimes exhibit too many degrees of freedom, making it more likely that they can get trapped in local minima during their evolution.

One solution to this problem is to control the snake with fewer degrees of freedom through the use of B-spline approximations (Menet, Saint-Marc, and Medioni 1990b,a; Cipolla and Blake 1990). The resulting *B-snake* can be written as

$$\boldsymbol{f}(s) = \sum_k B_k(s)\boldsymbol{x}_k \tag{5.10}$$

or in discrete form as

$$\boldsymbol{F} = \boldsymbol{B}\boldsymbol{X} \tag{5.11}$$

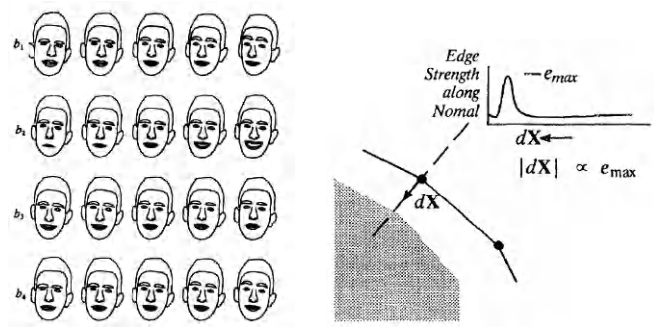
with

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{f}^T(0) \\ \vdots \\ \boldsymbol{f}^T(N) \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} B_0(s_0) & \dots & B_K(s_0) \\ \vdots & \ddots & \vdots \\ B_0(s_N) & \dots & B_K(s_N) \end{bmatrix}, \quad \text{and} \quad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^T(0) \\ \vdots \\ \boldsymbol{x}^T(K) \end{bmatrix}. \tag{5.12}$$

If the object being tracked or recognized has large variations in location, scale, or orientation, these can be modeled as an additional transformation on the control points, e.g., $\boldsymbol{x}'_k = s\boldsymbol{R}\boldsymbol{x}_k + \boldsymbol{t}$ (2.18), which can be estimated at the same time as the values of the control points. Alternatively, separate *detection* and *alignment* stages can be run to first localize and orient the objects of interest (Cootes, Cooper, Taylor *et al.* 1995).

In a B-snake, because the snake is controlled by fewer degrees of freedom, there is less need for the internal smoothness forces used with the original snakes, although these can still be derived and implemented using finite element analysis, i.e., taking derivatives and integrals of the B-spline basis functions (Terzopoulos 1983; Bathe 2007).

In practice, it is more common to estimate a set of *shape priors* on the typical distribution of the control points $\{\boldsymbol{x}_k\}$ (Cootes, Cooper, Taylor *et al.* 1995). Consider the set of resistor



(a) (b)

Figure 5.5 Active Shape Model (ASM): (a) the effect of varying the first four shape parameters for a set of faces (Cootes, Taylor, Lanitis *et al.* 1993) © 1993 IEEE; (b) searching for the strongest gradient along the normal to each control point (Cootes, Cooper, Taylor *et al.* 1995) © 1995 Elsevier.

shapes shown in Figure 5.4a. If we describe each contour with the set of control points shown in Figure 5.4b, we can plot the distribution of each point in a scatter plot, as shown in Figure 5.4c.

One potential way of describing this distribution would be by the location $\bar{\boldsymbol{x}}_k$ and 2D covariance $\boldsymbol{C}_k$ of each individual point $\boldsymbol{x}_k$. These could then be turned into a quadratic penalty (prior energy) on the point location,

$$E_{\text{loc}}(\boldsymbol{x}_k) = \frac{1}{2}(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)^T \boldsymbol{C}_k^{-1}(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k). \tag{5.13}$$

In practice, however, the variation in point locations is usually highly correlated.

A preferable approach is to estimate the joint covariance of all the points simultaneously. First, concatenate all of the point locations $\{\boldsymbol{x}_k\}$ into a single vector $\boldsymbol{x}$, e.g., by interleaving the x and y locations of each point. The distribution of these vectors across all training examples (Figure 5.4a) can be described with a mean $\bar{\boldsymbol{x}}$ and a covariance

$$\boldsymbol{C} = \frac{1}{P}\sum_p (\boldsymbol{x}_p - \bar{\boldsymbol{x}})(\boldsymbol{x}_p - \bar{\boldsymbol{x}})^T, \tag{5.14}$$

where $\boldsymbol{x}_p$ are the P training examples. Using *eigenvalue analysis* (Appendix A.1.2), which is also known as *Principal Component Analysis* (PCA) (Appendix B.1.1), the covariance matrix can be written as,

$$\boldsymbol{C} = \boldsymbol{\Phi}\ \text{diag}(\lambda_0 \ldots \lambda_{K-1})\ \boldsymbol{\Phi}^T. \tag{5.15}$$

In most cases, the likely appearance of the points can be modeled using only a few eigenvectors with the largest eigenvalues. The resulting *point distribution model* (Cootes, Taylor, Lanitis *et al.* 1993; Cootes, Cooper, Taylor *et al.* 1995) can be written as

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \hat{\boldsymbol{\Phi}}\,\boldsymbol{b}, \tag{5.16}$$

where $\boldsymbol{b}$ is an $M \ll K$ element *shape parameter* vector and $\hat{\boldsymbol{\Phi}}$ are the first m columns of $\boldsymbol{\Phi}$. To constrain the shape parameters to reasonable values, we can use a quadratic penalty of the

form

$$E_{\text{shape}} = \frac{1}{2}\boldsymbol{b}^T \text{ diag}(\lambda_0 \ldots \lambda_{M-1})\, \boldsymbol{b} = \sum_m b_m^2/2\lambda_m. \tag{5.17}$$

Alternatively, the range of allowable b_m values can be limited to some range, e.g., $|b_m| \leq 3\sqrt{\lambda_m}$ (Cootes, Cooper, Taylor *et al.* 1995). Alternative approaches for deriving a set of shape vectors are reviewed by Isard and Blake (1998).

Varying the individual shape parameters b_m over the range $-2\sqrt{\lambda_m} \leq 2\sqrt{\lambda_m}$ can give a good indication of the expected variation in appearance, as shown in Figure 5.4d. Another example, this time related to face contours, is shown in Figure 5.5a.

In order to align a point distribution model with an image, each control point searches in a direction normal to the contour to find the most likely corresponding image edge point (Figure 5.5b). These individual measurements can be combined with priors on the shape parameters (and, if desired, position, scale, and orientation parameters) to estimate a new set of parameters. The resulting *Active Shape Model* (ASM) can be iteratively minimized to fit images to non-rigidly deforming objects such as medical images or body parts such as hands (Cootes, Cooper, Taylor *et al.* 1995). The ASM can also be combined with a PCA analysis of the underlying gray-level distribution to create an *Active Appearance Model* (AAM) (Cootes, Edwards, and Taylor 2001), which we discuss in more detail in Section 14.2.2.

5.1.2 Dynamic snakes and CONDENSATION

In many applications of active contours, the object of interest is being tracked from frame to frame as it deforms and evolves. In this case, it makes sense to use estimates from the previous frame to predict and constrain the new estimates.

One way to do this is to use Kalman filtering, which results in a formulation called *Kalman snakes* (Terzopoulos and Szeliski 1992; Blake, Curwen, and Zisserman 1993). The Kalman filter is based on a linear dynamic model of shape parameter evolution,

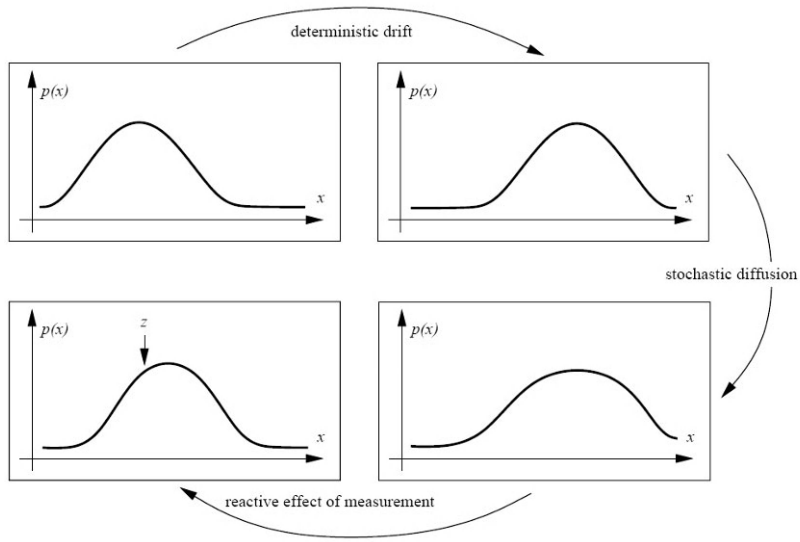
$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t, \tag{5.18}$$

where $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ are the current and previous state variables, $\boldsymbol{A}$ is the linear *transition matrix*, and $\boldsymbol{w}$ is a noise (perturbation) vector, which is often modeled as a Gaussian (Gelb 1974). The matrices $\boldsymbol{A}$ and the noise covariance can be learned ahead of time by observing typical sequences of the object being tracked (Blake and Isard 1998).

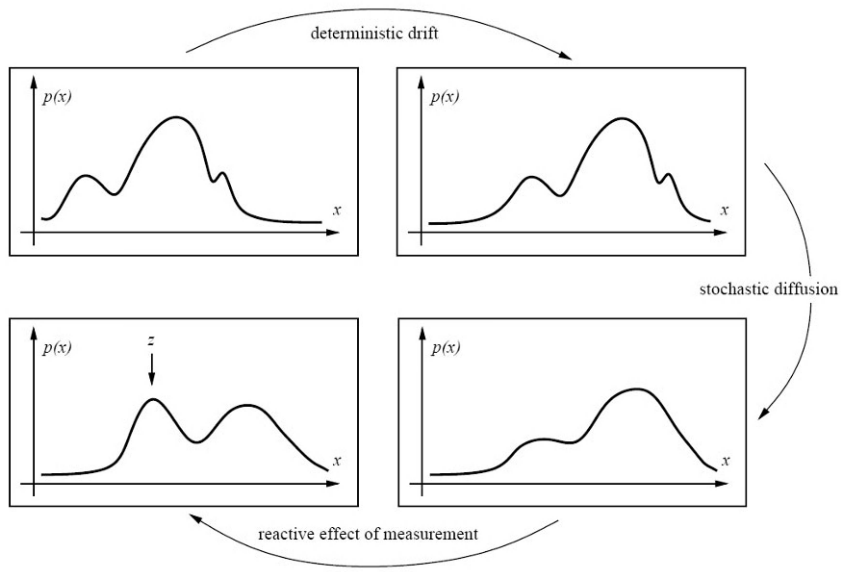
The qualitative behavior of the Kalman filter can be seen in Figure 5.6a. The linear dynamic model causes a deterministic change (drift) in the previous estimate, while the process noise (perturbation) causes a stochastic diffusion that increases the system entropy (lack of certainty). New measurements from the current frame restore some of the certainty (peakedness) in the updated estimate.

In many situations, however, such as when tracking in clutter, a better estimate for the contour can be obtained if we remove the assumptions that the distribution are Gaussian, which is what the Kalman filter requires. In this case, a general multi-modal distribution is propagated, as shown in Figure 5.6b. In order to model such multi-modal distributions, Isard and Blake (1998) introduced the use of *particle filtering* to the computer vision community.[5]

[5] Alternatives to modeling multi-modal distributions include *mixtures of Gaussians* (Bishop 2006) and *multiple hypothesis tracking* (Bar-Shalom and Fortmann 1988; Cham and Rehg 1999).



(a)



(b)

Figure 5.6 Probability density propagation (Isard and Blake 1998) © 1998 Springer. At the beginning of each estimation step, the probability density is updated according to the linear dynamic model (deterministic drift) and its certainty is reduced due to process noise (stochastic diffusion). New measurements introduce additional information that helps refine the current estimate. (a) The Kalman filter models the distributions as uni-modal, i.e., using a mean and covariance. (b) Some applications require more general multi-modal distributions.

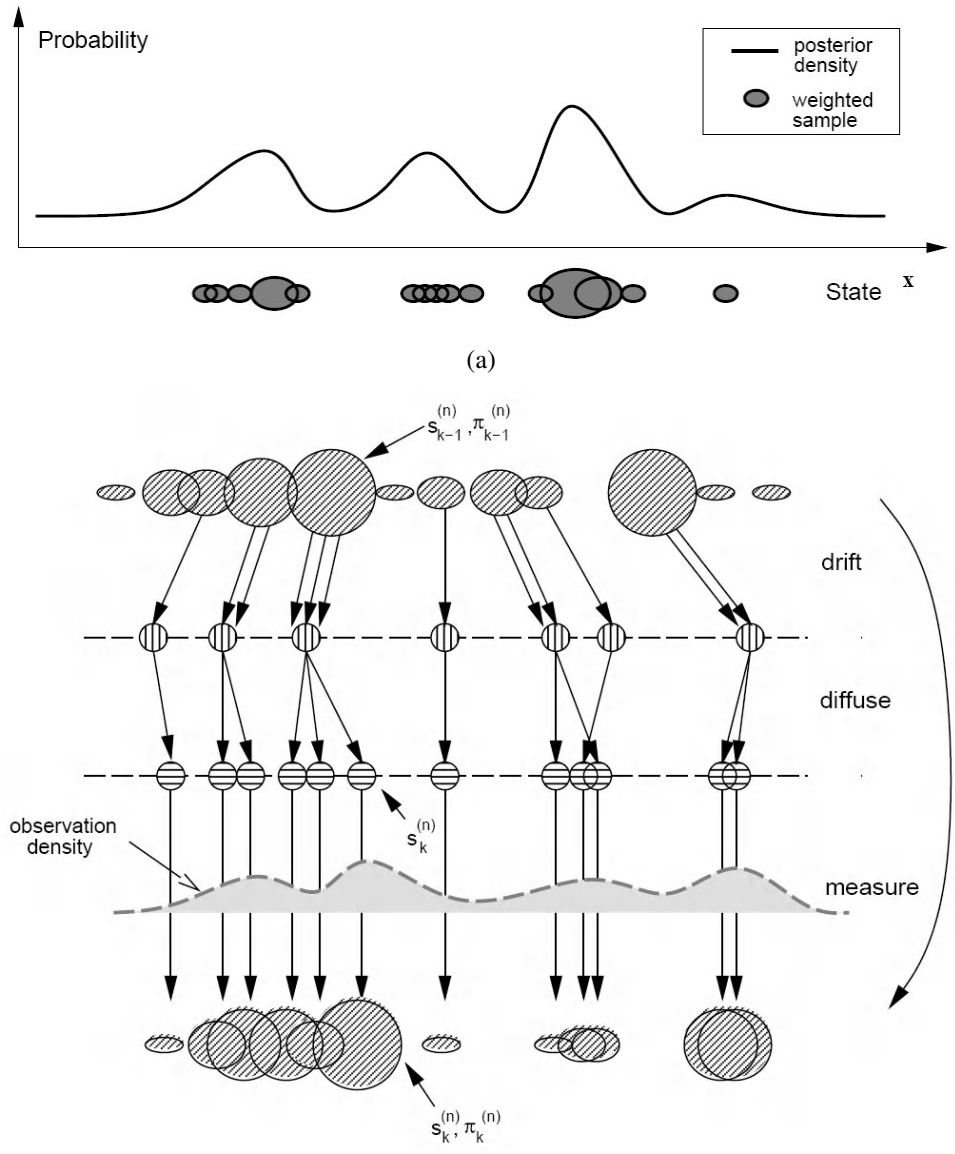
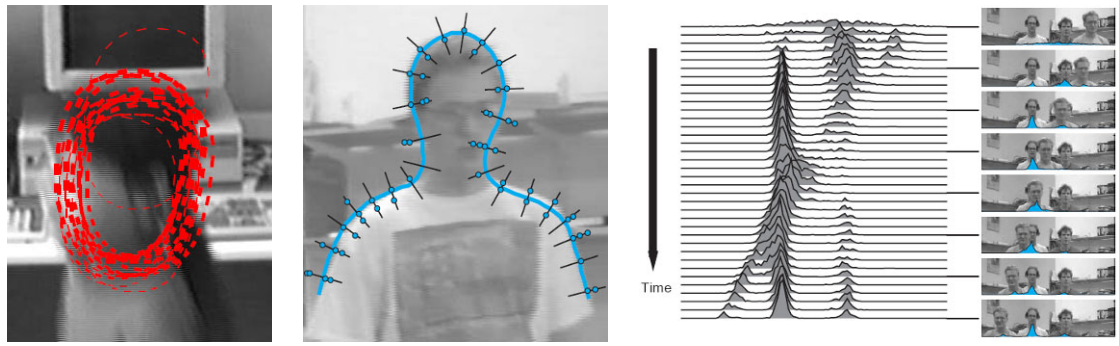


Figure 5.7 Factored sampling using particle filter in the CONDENSATION algorithm (Isard and Blake 1998) © 1998 Springer: (a) each density distribution is represented using a superposition of weighted *particles*; (b) the drift-diffusion-measurement cycle implemented using random sampling, perturbation, and re-weighting stages.



(a) (b) (c)

Figure 5.8 Head tracking using CONDENSATION (Isard and Blake 1998) © 1998 Springer: (a) sample set representation of head estimate distribution; (b) multiple measurements at each control vertex location; (c) multi-hypothesis tracking over time.

Particle filtering techniques represent a probability distribution using a collection of weighted point samples (Figure 5.7a) (Andrieu, de Freitas, Doucet *et al.* 2003; Bishop 2006; Koller and Friedman 2009). To update the locations of the samples according to the linear dynamics (deterministic drift), the centers of the samples are updated according to (5.18) and multiple samples are generated for each point (Figure 5.7b). These are then perturbed to account for the stochastic diffusion, i.e., their locations are moved by random vectors taken from the distribution of $\boldsymbol{w}$.[6] Finally, the weights of these samples are multiplied by the measurement probability density, i.e., we take each sample and measure its likelihood given the current (new) measurements. Because the point samples represent and propagate conditional estimates of the multi-modal density, Isard and Blake (1998) dubbed their algorithm CONditional DENSity propagATION or CONDENSATION.

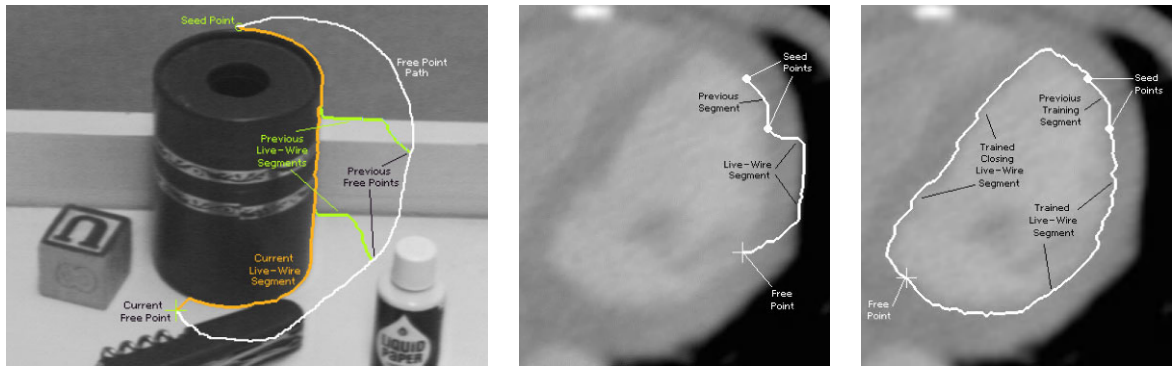
Figure 5.8a shows what a factored sample of a head tracker might look like, drawing a red B-spline contour for each of (a subset of) the particles being tracked. Figure 5.8b shows why the measurement density itself is often multi-modal: the locations of the edges perpendicular to the spline curve can have multiple local maxima due to background clutter. Finally, Figure 5.8c shows the temporal evolution of the conditional density (x coordinate of the head and shoulder tracker centroid) as it tracks several people over time.

5.1.3 Scissors

Active contours allow a user to roughly specify a boundary of interest and have the system evolve the contour towards a more accurate location as well as track it over time. The results of this curve evolution, however, may be unpredictable and may require additional user-based hints to achieve the desired result.

An alternative approach is to have the system optimize the contour in real time as the user is drawing (Mortensen 1999). The *intelligent scissors* system developed by Mortensen and Barrett (1995) does just that. As the user draws a rough outline (the white curve in Figure 5.9a), the system computes and draws a better curve that clings to high-contrast edges

[6] Note that because of the structure of these steps, non-linear dynamics and non-Gaussian noise can be used.



(a) (b) (c)

Figure 5.9 Intelligent scissors: (a) as the mouse traces the white path, the scissors follow the orange path along the object boundary (the green curves show intermediate positions) (Mortensen and Barrett 1995) © 1995 ACM; (b) regular scissors can sometimes jump to a stronger (incorrect) boundary; (c) after training to the previous segment, similar edge profiles are preferred (Mortensen and Barrett 1998) © 1995 Elsevier.

(the orange curve).

To compute the optimal curve path (*live-wire*), the image is first pre-processed to associate low costs with edges (links between neighboring horizontal, vertical, and diagonal, i.e., $\mathcal{N}_8$ neighbors) that are likely to be boundary elements. Their system uses a combination of zero-crossing, gradient magnitudes, and gradient orientations to compute these costs.

Next, as the user traces a rough curve, the system continuously recomputes the lowest-cost path between the starting *seed point* and the current mouse location using Dijkstra's algorithm, a breadth-first dynamic programming algorithm that terminates at the current target location.

In order to keep the system from jumping around unpredictably, the system will "freeze" the curve to date (reset the seed point) after a period of inactivity. To prevent the live wire from jumping onto adjacent higher-contrast contours, the system also "learns" the intensity profile under the current optimized curve, and uses this to preferentially keep the wire moving along the same (or a similar looking) boundary (Figure 5.9b–c).

Several extensions have been proposed to the basic algorithm, which works remarkably well even in its original form. Mortensen and Barrett (1999) use *tobogganing*, which is a simple form of watershed region segmentation, to pre-segment the image into regions whose boundaries become candidates for optimized curve paths. The resulting region boundaries are turned into a much smaller graph, where nodes are located wherever three or four regions meet. The Dijkstra algorithm is then run on this reduced graph, resulting in much faster (and often more stable) performance. Another extension to intelligent scissors is to use a probabilistic framework that takes into account the current trajectory of the boundary, resulting in a system called JetStream (Pérez, Blake, and Gangnet 2001).

Instead of re-computing an optimal curve at each time instant, a simpler system can be developed by simply "snapping" the current mouse position to the nearest likely boundary point (Gleicher 1995). Applications of these boundary extraction techniques to image cutting and pasting are presented in Section 10.4.

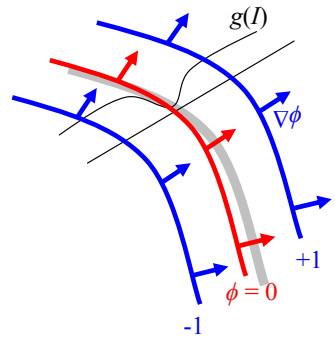


Figure 5.10 Level set evolution for a geodesic active contour. The embedding function ϕ is updated based on the curvature of the underlying surface modulated by the edge/speed function $g(I)$, as well as the gradient of $g(I)$, thereby attracting it to strong edges.

5.1.4 Level Sets

A limitation of active contours based on parametric curves of the form $\boldsymbol{f}(s)$, e.g., snakes, B-snakes, and CONDENSATION, is that it is challenging to change the topology of the curve as it evolves. (McInerney and Terzopoulos (1999, 2000) describe one approach to doing this.) Furthermore, if the shape changes dramatically, curve reparameterization may also be required.

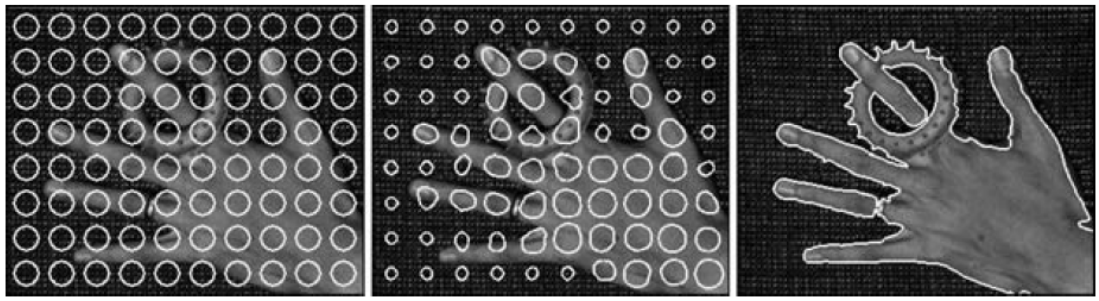
An alternative representation for such closed contours is to use a *level set*, where the *zero-crossing(s)* of a *characteristic* (or signed distance (Section 3.3.3)) function define the curve. Level sets evolve to fit and track objects of interest by modifying the underlying *embedding function* (another name for this 2D function) $\phi(x, y)$ instead of the curve $\boldsymbol{f}(s)$ (Malladi, Sethian, and Vemuri 1995; Sethian 1999; Sapiro 2001; Osher and Paragios 2003). To reduce the amount of computation required, only a small strip (frontier) around the locations of the current zero-crossing needs to updated at each step, which results in what are called *fast marching methods* (Sethian 1999).

An example of an evolution equation is the *geodesic active contour* proposed by Caselles, Kimmel, and Sapiro (1997) and Yezzi, Kichenassamy, Kumar *et al.* (1997),

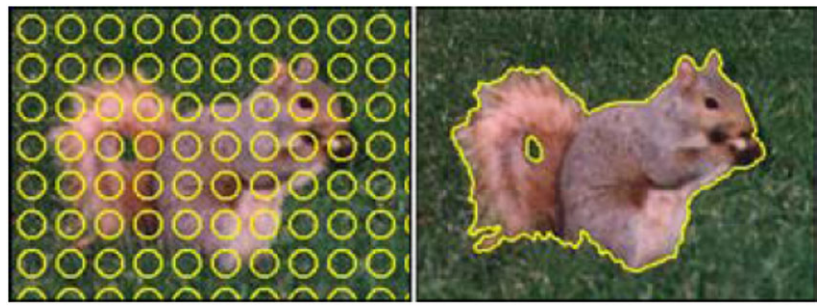
$$\begin{aligned} \frac{d\phi}{dt} &= |\nabla\phi| \text{div}\left(g(I)\frac{\nabla\phi}{|\nabla\phi|}\right) \\ &= g(I)|\nabla\phi| \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \nabla g(I) \cdot \nabla\phi, \end{aligned} \tag{5.19}$$

where $g(I)$ is a generalized version of the snake edge potential (5.5). To get an intuitive sense of the curve's behavior, assume that the embedding function ϕ is a signed distance function away from the curve (Figure 5.10), in which case $|\phi| = 1$. The first term in Equation (5.19) moves the curve in the direction of its curvature, i.e., it acts to straighten the curve, under the influence of the modulation function $g(I)$. The second term moves the curve down the gradient of $g(I)$, encouraging the curve to migrate towards minima of $g(I)$.

While this level-set formulation can readily change topology, it is still susceptible to local minima, since it is based on local measurements such as image gradients. An alternative



(a)



(b)

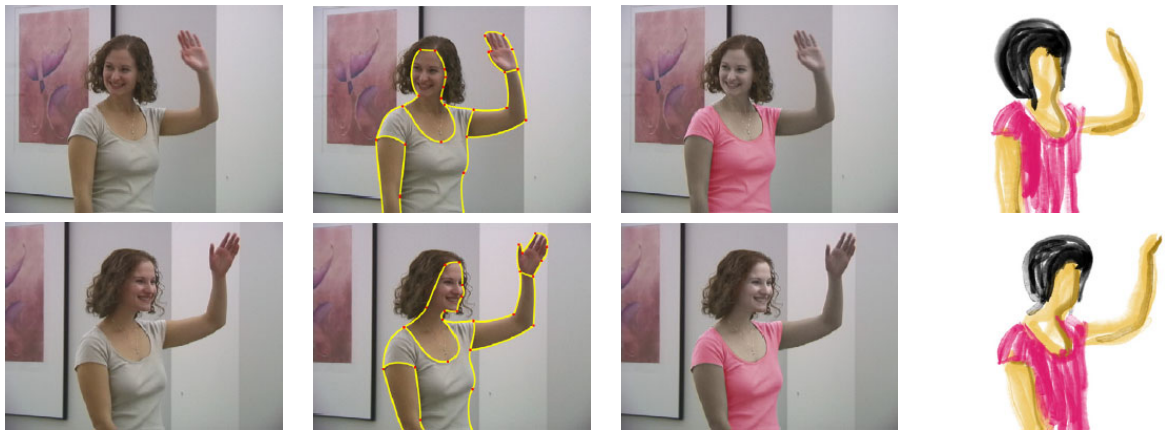
Figure 5.11 Level set segmentation (Cremers, Rousson, and Deriche 2007) © 2007 Springer: (a) grayscale image segmentation and (b) color image segmentation. Uni-variate and multi-variate Gaussians are used to model the foreground and background pixel distributions. The initial circles evolve towards an accurate segmentation of foreground and background, adapting their topology as they evolve.

approach is to re-cast the problem in a segmentation framework, where the energy measures the consistency of the image statistics (e.g., color, texture, motion) inside and outside the segmented regions (Cremers, Rousson, and Deriche 2007; Rousson and Paragios 2008; Houhou, Thiran, and Bresson 2008). These approaches build on earlier energy-based segmentation frameworks introduced by Leclerc (1989), Mumford and Shah (1989), and Chan and Vese (1992), which are discussed in more detail in Section 5.5. Examples of such level-set segmentations are shown in Figure 5.11, which shows the evolution of the level sets from a series of distributed circles towards the final binary segmentation.

For more information on level sets and their applications, please see the collection of papers edited by Osher and Paragios (2003) as well as the series of Workshops on Variational and Level Set Methods in Computer Vision (Paragios, Faugeras, Chan *et al.* 2005) and Special Issues on Scale Space and Variational Methods in Computer Vision (Paragios and Sgallari 2009).

5.1.5 *Application*: Contour tracking and rotoscoping

Active contours can be used in a wide variety of object-tracking applications (Blake and Isard 1998; Yilmaz, Javed, and Shah 2006). For example, they can be used to track facial features for performance-driven animation (Terzopoulos and Waters 1990; Lee, Terzopoulos, and Wa-



(a) (b) (c) (d)

Figure 5.12 Keyframe-based rotoscoping (Agarwala, Hertzmann, Seitz *et al.* 2004) © 2004 ACM: (a) original frames; (b) rotoscoped contours; (c) re-colored blouse; (d) rotoscoped hand-drawn animation.

ters 1995; Parke and Waters 1996; Bregler, Covell, and Slaney 1997) (Figure 5.2b). They can also be used to track heads and people, as shown in Figure 5.8, as well as moving vehicles (Paragios and Deriche 2000). Additional applications include medical image segmentation, where contours can be tracked from slice to slice in computerized tomography (3D medical imagery) (Cootes and Taylor 2001) or over time, as in ultrasound scans.

An interesting application that is closer to computer animation and visual effects is *rotoscoping*, which uses the tracked contours to deform a set of hand-drawn animations (or to modify or replace the original video frames).[7] Agarwala, Hertzmann, Seitz *et al.* (2004) present a system based on tracking hand-drawn B-spline contours drawn at selected keyframes, using a combination of geometric and appearance-based criteria (Figure 5.12). They also provide an excellent review of previous rotoscoping and image-based, contour-tracking systems.

Additional applications of rotoscoping (object contour detection and segmentation), such as cutting and pasting objects from one photograph into another, are presented in Section 10.4.

5.2 Split and merge

As mentioned in the introduction to this chapter, the simplest possible technique for segmenting a grayscale image is to select a threshold and then compute connected components (Section 3.3.2). Unfortunately, a single threshold is rarely sufficient for the whole image because of lighting and intra-object statistical variations.

In this section, we describe a number of algorithms that proceed either by recursively splitting the whole image into pieces based on region statistics or, conversely, merging pixels and regions together in a hierarchical fashion. It is also possible to combine both splitting and merging by starting with a medium-grain segmentation (in a quadtree representation) and

[7] The term comes from a device (a rotoscope) that projected frames of a live-action film underneath an acetate so that artists could draw animations directly over the actors' shapes.

then allowing both merging and splitting operations (Horowitz and Pavlidis 1976; Pavlidis and Liow 1990).

5.2.1 Watershed

A technique related to thresholding, since it operates on a grayscale image, is *watershed* computation (Vincent and Soille 1991). This technique segments an image into several *catchment basins*, which are the regions of an image (interpreted as a height field or landscape) where rain would flow into the same lake. An efficient way to compute such regions is to start flooding the landscape at all of the local minima and to label ridges wherever differently evolving components meet. The whole algorithm can be implemented using a priority queue of pixels and breadth-first search (Vincent and Soille 1991).[8]

Since images rarely have dark regions separated by lighter ridges, watershed segmentation is usually applied to a smoothed version of the gradient magnitude image, which also makes it usable with color images. As an alternative, the maximum oriented energy in a steerable filter (3.28–3.29) (Freeman and Adelson 1991) can be used as the basis of the *oriented watershed transform* developed by Arbeláez, Maire, Fowlkes *et al.* (2010). Such techniques end up finding smooth regions separated by visible (higher gradient) boundaries. Since such boundaries are what active contours usually follow, active contour algorithms (Mortensen and Barrett 1999; Li, Sun, Tang *et al.* 2004) often precompute such a segmentation using either the watershed or the related *tobogganing* technique (Section 5.1.3).

Unfortunately, watershed segmentation associates a unique region with each local minimum, which can lead to over-segmentation. Watershed segmentation is therefore often used as part of an interactive system, where the user first marks seed locations (with a click or a short stroke) that correspond to the centers of different desired components. Figure 5.13 shows the results of running the watershed algorithm with some manually placed markers on a confocal microscopy image. It also shows the result for an improved version of watershed that uses local morphology to smooth out and optimize the boundaries separating the regions (Beare 2006).

5.2.2 Region splitting (divisive clustering)

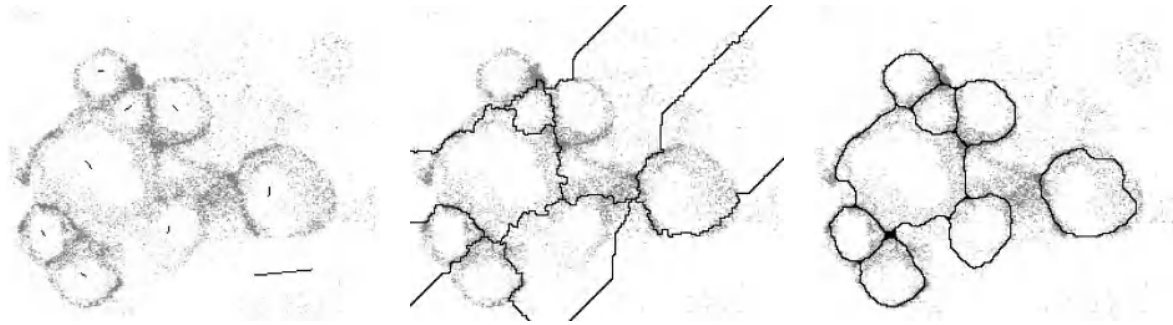
Splitting the image into successively finer regions is one of the oldest techniques in computer vision. Ohlander, Price, and Reddy (1978) present such a technique, which first computes a histogram for the whole image and then finds a threshold that best separates the large peaks in the histogram. This process is repeated until regions are either fairly uniform or below a certain size.

More recent splitting algorithms often optimize some metric of intra-region similarity and inter-region dissimilarity. These are covered in Sections 5.4 and 5.5.

5.2.3 Region merging (agglomerative clustering)

Region merging techniques also date back to the beginnings of computer vision. Brice and Fennema (1970) use a dual grid for representing boundaries between pixels and merge re-

[8] A related algorithm can be used to compute maximally stable extremal regions (MSERs) efficiently (Section 4.1.1) (Nistér and Stewénius 2008).



(a) (b) (c)

Figure 5.13 Locally constrained watershed segmentation (Beare 2006) © 2006 IEEE: (a) original confocal microscopy image with marked seeds (line segments); (b) standard watershed segmentation; (c) locally constrained watershed segmentation.

gions based on their relative boundary lengths and the strength of the visible edges at these boundaries.

In data clustering, algorithms can link clusters together based on the distance between their closest points (single-link clustering), their farthest points (complete-link clustering), or something in between (Jain, Topchy, Law *et al.* 2004). Kamvar, Klein, and Manning (2002) provide a probabilistic interpretation of these algorithms and show how additional models can be incorporated within this framework.

A very simple version of pixel-based merging combines adjacent regions whose average color difference is below a threshold or whose regions are too small. Segmenting the image into such *superpixels* (Mori, Ren, Efros *et al.* 2004), which are not semantically meaningful, can be a useful pre-processing stage to make higher-level algorithms such as stereo matching (Zitnick, Kang, Uyttendaele *et al.* 2004; Taguchi, Wilburn, and Zitnick 2008), optic flow (Zitnick, Jojic, and Kang 2005; Brox, Bregler, and Malik 2009), and recognition (Mori, Ren, Efros *et al.* 2004; Mori 2005; Gu, Lim, Arbelaez *et al.* 2009; Lim, Arbeláez, Gu *et al.* 2009) both faster and more robust.

5.2.4 Graph-based segmentation

While many merging algorithms simply apply a fixed rule that groups pixels and regions together, Felzenszwalb and Huttenlocher (2004b) present a merging algorithm that uses *relative dissimilarities* between regions to determine which ones should be merged; it produces an algorithm that provably optimizes a global grouping metric. They start with a pixel-to-pixel dissimilarity measure $w(e)$ that measures, for example, intensity differences between $\mathcal{N}_8$ neighbors. (Alternatively, they can use the *joint feature space* distances (5.42) introduced by Comaniciu and Meer (2002), which we discuss in Section 5.3.2.)

For any region R, its *internal difference* is defined as the largest edge weight in the region's minimum spanning tree,

$$Int(R) = \min_{e \in MST(R)} w(e). \tag{5.20}$$

For any two adjacent regions with at least one edge connecting their vertices, the difference



(a) (b) (c)

Figure 5.14 Graph-based merging segmentation (Felzenszwalb and Huttenlocher 2004b) © 2004 Springer: (a) input grayscale image that is successfully segmented into three regions even though the variation inside the smaller rectangle is larger than the variation across the middle edge; (b) input grayscale image; (c) resulting segmentation using an $\mathcal{N}_8$ pixel neighborhood.

between these regions is defined as the minimum weight edge connecting the two regions,

$$Dif(R_1, R_2) = \min_{e=(v_1,v_2)|v_1 \in R_1, v_2 \in R_2} w(e). \tag{5.21}$$

Their algorithm merges any two adjacent regions whose difference is smaller than the minimum internal difference of these two regions,

$$MInt(R_1, R_2) = \min(Int(R_1) + \tau(R_1), Int(R_2) + \tau(R_2)), \tag{5.22}$$

where $\tau(R)$ is a heuristic region penalty that Felzenszwalb and Huttenlocher (2004b) set to $k/|R|$, but which can be set to any application-specific measure of region goodness.

By merging regions in decreasing order of the edges separating them (which can be efficiently evaluated using a variant of Kruskal's minimum spanning tree algorithm), they provably produce segmentations that are neither too fine (there exist regions that could have been merged) nor too coarse (there are regions that could be split without being mergeable). For fixed-size pixel neighborhoods, the running time for this algorithm is $O(N \log N)$, where N is the number of image pixels, which makes it one of the fastest segmentation algorithms (Paris and Durand 2007). Figure 5.14 shows two examples of images segmented using their technique.

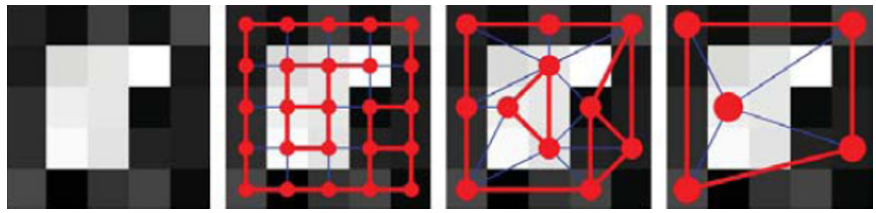
5.2.5 Probabilistic aggregation

Alpert, Galun, Basri *et al.* (2007) develop a probabilistic merging algorithm based on two cues, namely gray-level similarity and texture similarity. The gray-level similarity between regions R_i and R_j is based on the *minimal external difference* from other neighboring regions,

$$\sigma^+_{local} = \min(\Delta^+_i, \Delta^+_j), \tag{5.23}$$

where $\Delta^+_i = \min_k |\Delta_{ik}|$ and Δ_{ik} is the difference in average intensities between regions R_i and R_k. This is compared to the *average intensity difference*,

$$\sigma^-_{local} = \frac{\Delta^-_i + \Delta^-_j}{2}, \tag{5.24}$$



(a) (b) (c) (d)

Figure 5.15 Coarse to fine node aggregation in segmentation by weighted aggregation (SWA) (Sharon, Galun, Sharon *et al.* 2006) © 2006 Macmillan Publishers Ltd [Nature]: (a) original gray-level pixel grid; (b) inter-pixel couplings, where thicker lines indicate stronger couplings; (c) after one level of coarsening, where each original pixel is strongly coupled to one of the coarse-level nodes; (d) after two levels of coarsening.

where $\Delta_i^- = \sum_k (\tau_{ik}\Delta_{ik}) / \sum_k (\tau_{ik})$ and τ_{ik} is the boundary length between regions R_i and R_k. The texture similarity is defined using relative differences between histogram bins of simple oriented Sobel filter responses. The pairwise statistics σ^+_{local} and σ^-_{local} are used to compute the likelihoods p_{ij} that two regions should be merged. (See the paper by Alpert, Galun, Basri *et al.* (2007) for more details.)

Merging proceeds in a hierarchical fashion inspired by algebraic multigrid techniques (Brandt 1986; Briggs, Henson, and McCormick 2000) and previously used by Alpert, Galun, Basri *et al.* (2007) in their segmentation by weighted aggregation (SWA) algorithm (Sharon, Galun, Sharon *et al.* 2006), which we discuss in Section 5.4. A subset of the nodes $C \subset V$ that are (collectively) *strongly coupled* to all of the original nodes (regions) are used to define the problem at a coarser scale (Figure 5.15), where strong coupling is defined as

$$\frac{\sum_{j \in C} p_{ij}}{\sum_{j \in V} p_{ij}} > \phi, \tag{5.25}$$

with ϕ usually set to 0.2. The intensity and texture similarity statistics for the coarser nodes are recursively computed using weighted averaging, where the relative strengths (couplings) between coarse- and fine-level nodes are based on their merge probabilities p_{ij}. This allows the algorithm to run in essentially $O(N)$ time, using the same kind of hierarchical aggregation operations that are used in pyramid-based filtering or preconditioning algorithms. After a segmentation has been identified at a coarser level, the exact memberships of each pixel are computed by propagating coarse-level assignments to their finer-level "children" (Sharon, Galun, Sharon *et al.* 2006; Alpert, Galun, Basri *et al.* 2007). Figure 5.22 shows the segmentations produced by this algorithm compared to other popular segmentation algorithms.

5.3 Mean shift and mode finding

Mean-shift and mode finding techniques, such as k-means and mixtures of Gaussians, model the feature vectors associated with each pixel (e.g., color and position) as samples from an unknown probability density function and then try to find clusters (modes) in this distribution.

Consider the color image shown in Figure 5.16a. How would you segment this image based on color alone? Figure 5.16b shows the distribution of pixels in L*u*v* space, which is equivalent to what a vision algorithm that ignores spatial location would see. To make the

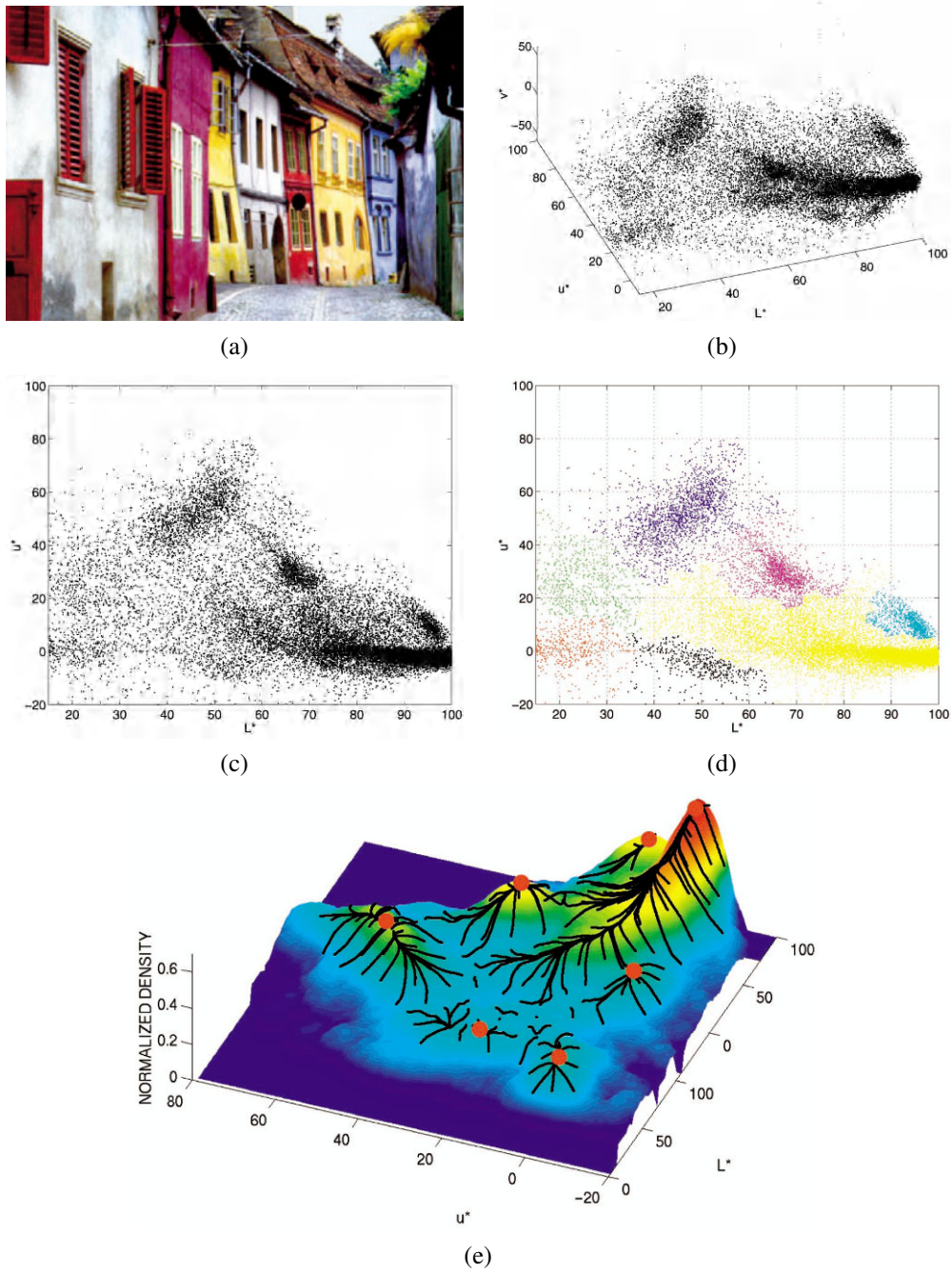


Figure 5.16 Mean-shift image segmentation (Comaniciu and Meer 2002) © 2002 IEEE: (a) input color image; (b) pixels plotted in L*u*v* space; (c) L*u* space distribution; (d) clustered results after 159 mean-shift procedures; (e) corresponding trajectories with peaks marked as red dots.

visualization simpler, let us only consider the L*u* coordinates, as shown in Figure 5.16c. How many obvious (elongated) clusters do you see? How would you go about finding these clusters?

The k-means and mixtures of Gaussians techniques use a *parametric* model of the density function to answer this question, i.e., they assume the density is the superposition of a small number of simpler distributions (e.g., Gaussians) whose locations (centers) and shape (covariance) can be estimated. Mean shift, on the other hand, smoothes the distribution and finds its peaks as well as the regions of feature space that correspond to each peak. Since a complete density is being modeled, this approach is called *non-parametric* (Bishop 2006). Let us look at these techniques in more detail.

5.3.1 K-means and mixtures of Gaussians

While k-means implicitly models the probability density as a superposition of spherically symmetric distributions, it does not require any probabilistic reasoning or modeling (Bishop 2006). Instead, the algorithm is given the number of clusters k it is supposed to find; it then iteratively updates the cluster center location based on the samples that are closest to each center. The algorithm can be initialized by randomly sampling k centers from the input feature vectors. Techniques have also been developed for splitting or merging cluster centers based on their statistics, and for accelerating the process of finding the nearest mean center (Bishop 2006).

In mixtures of Gaussians, each cluster center is augmented by a covariance matrix whose values are re-estimated from the corresponding samples. Instead of using nearest neighbors to associate input samples with cluster centers, a *Mahalanobis distance* (Appendix B.1.1) is used:

$$d(\boldsymbol{x}_i, \boldsymbol{\mu}_k; \boldsymbol{\Sigma}_k) = \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_{\boldsymbol{\Sigma}_k^{-1}} = (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k) \tag{5.26}$$

where $\boldsymbol{x}_i$ are the input samples, $\boldsymbol{\mu}_k$ are the cluster centers, and $\boldsymbol{\Sigma}_k$ are their covariance estimates. Samples can be associated with the nearest cluster center (a *hard assignment* of membership) or can be *softly assigned* to several nearby clusters.

This latter, more commonly used, approach corresponds to iteratively re-estimating the parameters for a mixture of Gaussians density function,

$$p(\boldsymbol{x}|\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}) = \sum_k \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{5.27}$$

where π_k are the *mixing coefficients*, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the Gaussian means and covariances, and

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{|\boldsymbol{\Sigma}_k|} e^{-d(\boldsymbol{x}, \boldsymbol{\mu}_k; \boldsymbol{\Sigma}_k)} \tag{5.28}$$

is the *normal* (Gaussian) distribution (Bishop 2006).

To iteratively compute (a local) maximum likely estimate for the unknown mixture parameters $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$, the *expectation maximization* (EM) algorithm (Dempster, Laird, and Rubin 1977) proceeds in two alternating stages:

1. The *expectation* stage (E step) estimates the *responsibilities*

$$z_{ik} = \frac{1}{Z_i} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad \text{with} \quad \sum_k z_{ik} = 1, \tag{5.29}$$

which are the estimates of how likely a sample $\boldsymbol{x}_i$ was generated from the kth Gaussian cluster.

2. The *maximization* stage (M step) updates the parameter values

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_i z_{ik}\boldsymbol{x}_i, \tag{5.30}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_i z_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T, \tag{5.31}$$

$$\pi_k = \frac{N_k}{N}, \tag{5.32}$$

where

$$N_k = \sum_i z_{ik}. \tag{5.33}$$

is an estimate of the number of sample points assigned to each cluster.

Bishop (2006) has a wonderful exposition of both mixture of Gaussians estimation and the more general topic of expectation maximization.

In the context of image segmentation, Ma, Derksen, Hong *et al.* (2007) present a nice review of segmentation using mixtures of Gaussians and develop their own extension based on Minimum Description Length (MDL) coding, which they show produces good results on the Berkeley segmentation database.

5.3.2 Mean shift

While k-means and mixtures of Gaussians use a parametric form to model the probability density function being segmented, mean shift implicitly models this distribution using a smooth continuous *non-parametric model*. The key to mean shift is a technique for efficiently finding peaks in this high-dimensional data distribution without ever computing the complete function explicitly (Fukunaga and Hostetler 1975; Cheng 1995; Comaniciu and Meer 2002).

Consider once again the data points shown in Figure 5.16c, which can be thought of as having been drawn from some probability density function. If we could compute this density function, as visualized in Figure 5.16e, we could find its major peaks (*modes*) and identify regions of the input space that climb to the same peak as being part of the same region. This is the inverse of the *watershed* algorithm described in Section 5.2.1, which climbs downhill to find *basins of attraction*.

The first question, then, is how to estimate the density function given a sparse set of samples. One of the simplest approaches is to just smooth the data, e.g., by convolving it with a fixed kernel of width h,

$$f(\boldsymbol{x}) = \sum_i K(\boldsymbol{x} - \boldsymbol{x}_i) = \sum_i k\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{h^2}\right), \tag{5.34}$$

where $\boldsymbol{x}_i$ are the input samples and $k(r)$ is the kernel function (or *Parzen window*).[9] This approach is known as *kernel density estimation* or the *Parzen window technique* (Duda, Hart,

[9] In this simplified formula, a Euclidean metric is used. We discuss a little later (5.42) how to generalize this to non-uniform (scaled or oriented) metrics. Note also that this distribution may not be *proper*, i.e., integrate to 1. Since we are looking for maxima in the density, this does not matter.

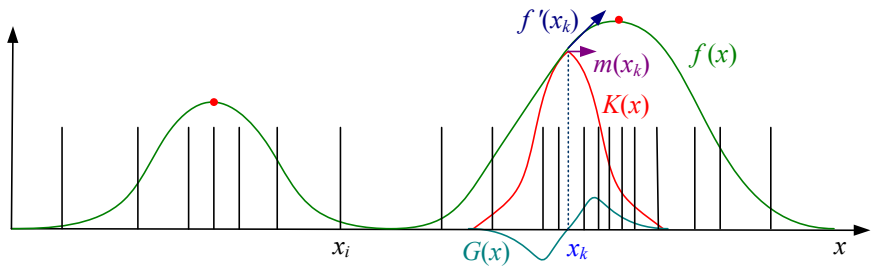


Figure 5.17 One-dimensional visualization of the kernel density estimate, its derivative, and a mean shift. The kernel density estimate $f(x)$ is obtained by convolving the sparse set of input samples x_i with the kernel function $K(x)$. The derivative of this function, $f'(x)$, can be obtained by convolving the inputs with the derivative kernel $G(x)$. Estimating the local displacement vectors around a current estimate x_k results in the mean-shift vector $m(x_k)$, which, in a multi-dimensional setting, point in the same direction as the function gradient $\nabla f(\boldsymbol{x}_k)$. The red dots indicate local maxima in $f(x)$ to which the mean shifts converge.

and Stork 2001, Section 4.3; Bishop 2006, Section 2.5.1). Once we have computed $f(\boldsymbol{x})$, as shown in Figures 5.16e and 5.17, we can find its local maxima using gradient ascent or some other optimization technique.

The problem with this "brute force" approach is that, for higher dimensions, it becomes computationally prohibitive to evaluate $f(\boldsymbol{x})$ over the complete search space.[10] Instead, mean shift uses a variant of what is known in the optimization literature as *multiple restart gradient descent*. Starting at some guess for a local maximum, $\boldsymbol{y}_k$, which can be a random input data point $\boldsymbol{x}_i$, mean shift computes the gradient of the density estimate $f(\boldsymbol{x})$ at $\boldsymbol{y}_k$ and takes an uphill step in that direction (Figure 5.17). The gradient of $f(\boldsymbol{x})$ is given by

$$\nabla f(\boldsymbol{x}) = \sum_i (\boldsymbol{x}_i - \boldsymbol{x}) G(\boldsymbol{x} - \boldsymbol{x}_i) = \sum_i (\boldsymbol{x}_i - \boldsymbol{x}) g\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{h^2}\right), \tag{5.35}$$

where

$$g(r) = -k'(r), \tag{5.36}$$

and $k'(r)$ is the first derivative of $k(r)$. We can re-write the gradient of the density function as

$$\nabla f(\boldsymbol{x}) = \left[\sum_i G(\boldsymbol{x} - \boldsymbol{x}_i)\right] \boldsymbol{m}(\boldsymbol{x}), \tag{5.37}$$

where the vector

$$\boldsymbol{m}(\boldsymbol{x}) = \frac{\sum_i \boldsymbol{x}_i G(\boldsymbol{x} - \boldsymbol{x}_i)}{\sum_i G(\boldsymbol{x} - \boldsymbol{x}_i)} - \boldsymbol{x} \tag{5.38}$$

is called the *mean shift*, since it is the difference between the weighted mean of the neighbors $\boldsymbol{x}_i$ around $\boldsymbol{x}$ and the current value of $\boldsymbol{x}$.

In the mean-shift procedure, the current estimate of the mode $\boldsymbol{y}_k$ at iteration k is replaced by its locally weighted mean,

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \boldsymbol{m}(\boldsymbol{y}_k) = \frac{\sum_i \boldsymbol{x}_i G(\boldsymbol{y}_k - \boldsymbol{x}_i)}{\sum_i G(\boldsymbol{y}_k - \boldsymbol{x}_i)}. \tag{5.39}$$

[10] Even for one dimension, if the space is extremely sparse, it may be inefficient.

Comaniciu and Meer (2002) prove that this algorithm converges to a local maximum of $f(\boldsymbol{x})$ under reasonably weak conditions on the kernel $k(r)$, i.e., that it is monotonically decreasing. This convergence is not guaranteed for regular gradient descent unless appropriate step size control is used.

The two kernels that Comaniciu and Meer (2002) studied are the Epanechnikov kernel,

$$k_E(r) = \max(0, 1 - r), \tag{5.40}$$

which is a radial generalization of a bilinear kernel, and the Gaussian (normal) kernel,

$$k_N(r) = \exp\left(-\frac{1}{2}r\right). \tag{5.41}$$

The corresponding derivative kernels $g(r)$ are a unit ball and another Gaussian, respectively. Using the Epanechnikov kernel converges in a finite number of steps, while the Gaussian kernel has a smoother trajectory (and produces better results), but converges very slowly near a mode (Exercise 5.5).

The simplest way to apply mean shift is to start a separate mean-shift mode estimate $\boldsymbol{y}$ at every input point $\boldsymbol{x}_i$ and to iterate for a fixed number of steps or until the mean-shift magnitude is below a threshold. A faster approach is to randomly subsample the input points $\boldsymbol{x}_i$ and to keep track of each point's temporal evolution. The remaining points can then be classified based on the nearest evolution path (Comaniciu and Meer 2002). Paris and Durand (2007) review a number of other more efficient implementations of mean shift, including their own approach, which is based on using an efficient low-resolution estimate of the complete multi-dimensional space of $f(\boldsymbol{x})$ along with its stationary points.

The color-based segmentation shown in Figure 5.16 only looks at pixel colors when determining the best clustering. It may therefore cluster together small isolated pixels that happen to have the same color, which may not correspond to a semantically meaningful segmentation of the image.

Better results can usually be obtained by clustering in the *joint domain* of color and location. In this approach, the spatial coordinates of the image $\boldsymbol{x}_s = (x, y)$, which are called the *spatial domain*, are concatenated with the color values $\boldsymbol{x}_r$, which are known as the *range domain*, and mean-shift clustering is applied in this five-dimensional space $\boldsymbol{x}_j$. Since location and color may have different scales, the kernels are adjusted accordingly, i.e., we use a kernel of the form

$$K(\boldsymbol{x}_j) = k\left(\frac{\|\boldsymbol{x}_r\|^2}{h_r^2}\right) k\left(\frac{\|\boldsymbol{x}_s\|^2}{h_s^2}\right), \tag{5.42}$$

where separate parameters h_s and h_r are used to control the spatial and range bandwidths of the filter kernels. Figure 5.18 shows an example of mean-shift clustering in the joint domain, with parameters $(h_s, h_r, M) = (16, 19, 40)$, where spatial regions containing less than M pixels are eliminated.

The form of the joint domain filter kernel (5.42) is reminiscent of the bilateral filter kernel (3.34–3.37) discussed in Section 3.3.1. The difference between mean shift and bilateral filtering, however, is that in mean shift the spatial coordinates of each pixel are adjusted along with its color values, so that the pixel migrates more quickly towards other pixels with similar colors, and can therefore later be used for clustering and segmentation.

Determining the best bandwidth parameters h to use with mean shift remains something of an art, although a number of approaches have been explored. These include optimizing

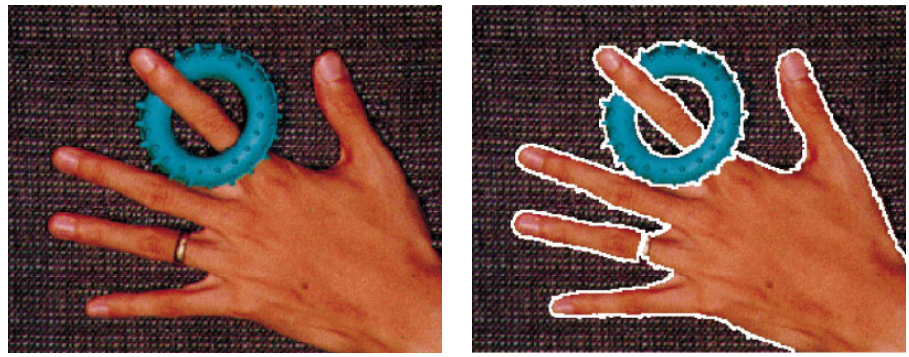


Figure 5.18 Mean-shift color image segmentation with parameters $(h_s, h_r, M) = (16, 19, 40)$ (Comaniciu and Meer 2002) © 2002 IEEE.

the bias–variance tradeoff, looking for parameter ranges where the number of clusters varies slowly, optimizing some external clustering criterion, or using top-down (application domain) knowledge (Comaniciu and Meer 2003). It is also possible to change the orientation of the kernel in joint parameter space for applications such as spatio-temporal (video) segmentations (Wang, Thiesson, Xu *et al.* 2004).

Mean shift has been applied to a number of different problems in computer vision, including face tracking, 2D shape extraction, and texture segmentation (Comaniciu and Meer 2002), and more recently in stereo matching (Chapter 11) (Wei and Quan 2004), non-photorealistic rendering (Section 10.5.2) (DeCarlo and Santella 2002), and video editing (Section 10.4.5) (Wang, Bhat, Colburn *et al.* 2005). Paris and Durand (2007) provide a nice review of such applications, as well as techniques for more efficiently solving the mean-shift equations and producing hierarchical segmentations.

5.4 Normalized cuts

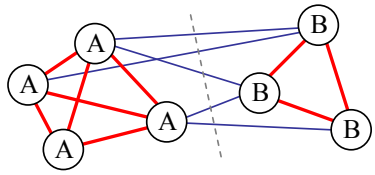
While bottom-up merging techniques aggregate regions into coherent wholes and mean-shift techniques try to find clusters of similar pixels using mode finding, the normalized cuts technique introduced by Shi and Malik (2000) examines the *affinities* (similarities) between nearby pixels and tries to separate groups that are connected by weak affinities.

Consider the simple graph shown in Figure 5.19a. The pixels in group A are all strongly connected with high affinities, shown as thick red lines, as are the pixels in group B. The connections between these two groups, shown as thinner blue lines, are much weaker. A *normalized cut* between the two groups, shown as a dashed line, separates them into two clusters.

The cut between two groups A and B is defined as the sum of all the weights being cut,

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij}, \tag{5.43}$$

where the weights between two pixels (or regions) i and j measure their similarity. Using a minimum cut as a segmentation criterion, however, does not result in reasonable clusters, since the smallest cuts usually involve isolating a single pixel.



(a)

	A	B	sum
A	$assoc(A, A)$	$cut(A, B)$	$assoc(A, V)$
B	$cut(B, A)$	$assoc(B, B)$	$assoc(B, V)$
sum	$assoc(A, V)$	$assoc(B, v)$	

(b)

Figure 5.19 Sample weighted graph and its normalized cut: (a) a small sample graph and its smallest normalized cut; (b) tabular form of the associations and cuts for this graph. The $assoc$ and cut entries are computed as area sums of the associated weight matrix $\boldsymbol{W}$ (Figure 5.20). Normalizing the table entries by the row or column sums produces normalized associations and cuts $Nassoc$ and $Ncut$.

A better measure of segmentation is the normalized cut, which is defined as

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \tag{5.44}$$

where $assoc(A, A) = \sum_{i \in A, j \in A} w_{ij}$ is the *association* (sum of all the weights) within a cluster and $assoc(A, V) = assoc(A, A) + cut(A, B)$ is the sum of *all* the weights associated with nodes in A. Figure 5.19b shows how the cuts and associations can be thought of as area sums in the weight matrix $\boldsymbol{W} = [w_{ij}]$, where the entries of the matrix have been arranged so that the nodes in A come first and the nodes in B come second. Figure 5.20 shows an actual weight matrix for which these area sums can be computed. Dividing each of these areas by the corresponding row sum (the rightmost column of Figure 5.19b) results in the normalized cut and association values. These normalized values better reflect the fitness of a particular segmentation, since they look for collections of edges that are weak relative to all of the edges both inside and emanating from a particular region.

Unfortunately, computing the optimal normalized cut is NP-complete. Instead, Shi and Malik (2000) suggest computing a real-valued assignment of nodes to groups. Let $\boldsymbol{x}$ be the *indicator vector* where $x_i = +1$ iff $i \in A$ and $x_i = -1$ iff $i \in B$. Let $\boldsymbol{d} = \boldsymbol{W}\boldsymbol{1}$ be the row sums of the symmetric matrix $\boldsymbol{W}$ and $\boldsymbol{D} = \text{diag}(\boldsymbol{d})$ be the corresponding diagonal matrix. Shi and Malik (2000) show that minimizing the normalized cut over all possible indicator vectors $\boldsymbol{x}$ is equivalent to minimizing

$$\min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y}}, \tag{5.45}$$

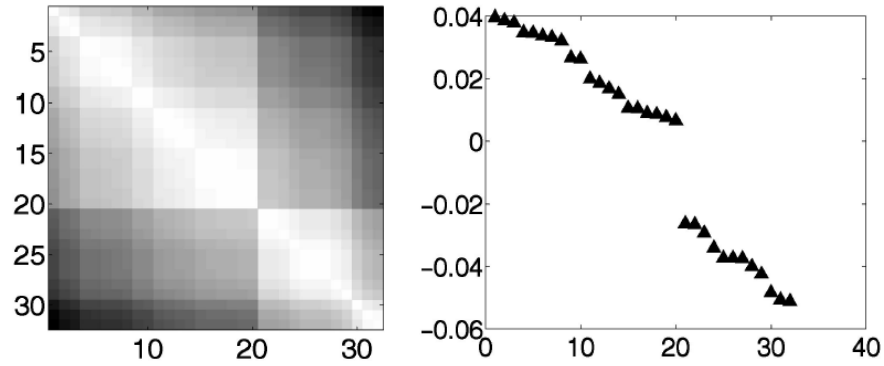
where $\boldsymbol{y} = ((\boldsymbol{1} + \boldsymbol{x}) - b(\boldsymbol{1} - \boldsymbol{x}))/2$ is a vector consisting of all 1s and $-b$s such that $\boldsymbol{y} \cdot \boldsymbol{d} = 0$. Minimizing this *Rayleigh quotient* is equivalent to solving the generalized eigenvalue system

$$(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y}, \tag{5.46}$$

which can be turned into a regular eigenvalue problem

$$(\boldsymbol{I} - \boldsymbol{N})\boldsymbol{z} = \lambda \boldsymbol{z}, \tag{5.47}$$

where $\boldsymbol{N} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ is the *normalized* affinity matrix (Weiss 1999) and $\boldsymbol{z} = \boldsymbol{D}^{1/2}\boldsymbol{y}$. Because these eigenvectors can be interpreted as the large modes of vibration in



(a) (b)

Figure 5.20 Sample weight table and its second smallest eigenvector (Shi and Malik 2000) © 2000 IEEE: (a) sample 32×32 weight matrix $\boldsymbol{W}$; (b) eigenvector corresponding to the second smallest eigenvalue of the generalized eigenvalue problem $(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y}$.

a spring-mass system, normalized cuts is an example of a *spectral method* for image segmentation.

Extending an idea originally proposed by Scott and Longuet-Higgins (1990), Weiss (1999) suggests normalizing the affinity matrix and then using the top k eigenvectors to reconstitute a $\boldsymbol{Q}$ matrix. Other papers have extended the basic normalized cuts framework by modifying the affinity matrix in different ways, finding better discrete solutions to the minimization problem, or applying multi-scale techniques (Meilă and Shi 2000, 2001; Ng, Jordan, and Weiss 2001; Yu and Shi 2003; Cour, Bénézit, and Shi 2005; Tolliver and Miller 2006).

Figure 5.20b shows the second smallest (real-valued) eigenvector corresponding to the weight matrix shown in Figure 5.20a. (Here, the rows have been permuted to separate the two groups of variables that belong to the different components of this eigenvector.) After this real-valued vector is computed, the variables corresponding to positive and negative eigenvector values are associated with the two cut components. This process can be further repeated to hierarchically subdivide an image, as shown in Figure 5.21.

The original algorithm proposed by Shi and Malik (2000) used spatial position and image feature differences to compute the pixel-wise affinities,

$$w_{ij} = \exp\left(-\frac{\|\boldsymbol{F}_i - \boldsymbol{F}_j\|^2}{\sigma_F^2} - \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma_s^2}\right), \tag{5.48}$$

for pixels within a radius $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < r$, where $\boldsymbol{F}$ is a feature vector that consists of intensities, colors, or oriented filter histograms. (Note how (5.48) is the negative exponential of the joint feature space distance (5.42).)

In subsequent work, Malik, Belongie, Leung *et al.* (2001) look for *intervening contours* between pixels i and j and define an intervening contour weight

$$w_{ij}^{IC} = 1 - \max_{\boldsymbol{x} \in l_{ij}} p_{con}(\boldsymbol{x}), \tag{5.49}$$

where l_{ij} is the image line joining pixels i and j and $p_{con}(\boldsymbol{x})$ is the probability of an intervening contour perpendicular to this line, which is defined as the negative exponential of the

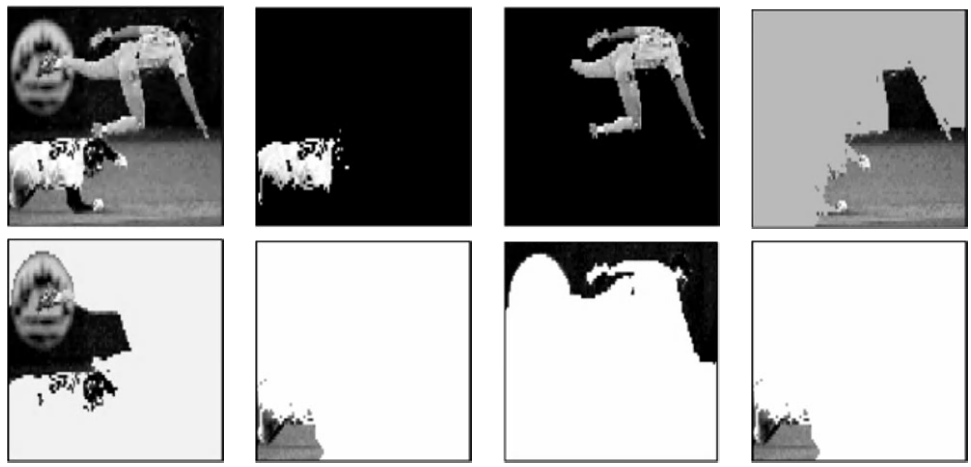


Figure 5.21 Normalized cuts segmentation (Shi and Malik 2000) © 2000 IEEE: The input image and the components returned by the normalized cuts algorithm.

oriented energy in the perpendicular direction. They multiply these weights with a texton-based texture similarity metric and use an initial over-segmentation based purely on local pixel-wise features to re-estimate intervening contours and texture statistics in a region-based manner. Figure 5.22 shows the results of running this improved algorithm on a number of test images.

Because it requires the solution of large sparse eigenvalue problems, normalized cuts can be quite slow. Sharon, Galun, Sharon *et al.* (2006) present a way to accelerate the computation of the normalized cuts using an approach inspired by algebraic multigrid (Brandt 1986; Briggs, Henson, and McCormick 2000). To coarsen the original problem, they select a smaller number of variables such that the remaining fine-level variables are *strongly coupled* to at least one coarse-level variable. Figure 5.15 shows this process schematically, while (5.25) gives the definition for strong coupling except that, in this case, the original weights w_{ij} in the normalized cut are used instead of merge probabilities p_{ij}.

Once a set of coarse variables has been selected, an inter-level interpolation matrix with elements similar to the left hand side of (5.25) is used to define a reduced version of the normalized cuts problem. In addition to computing the weight matrix using interpolation-based coarsening, additional region statistics are used to modulate the weights. After a normalized cut has been computed at the coarsest level of analysis, the membership values of finer-level nodes are computed by interpolating parent values and mapping values within $\epsilon = 0.1$ of 0 and 1 to pure Boolean values.

An example of the segmentation produced by weighted aggregation (SWA) is shown in Figure 5.22, along with the most recent probabilistic bottom-up merging algorithm by Alpert, Galun, Basri *et al.* (2007), which was described in Section 5.2. In even more recent work, Wang and Oliensis (2010) show how to estimate statistics over segmentations (e.g., mean region size) directly from the affinity graph. They use this to produce segmentations that are more *central* with respect to other possible segmentations.

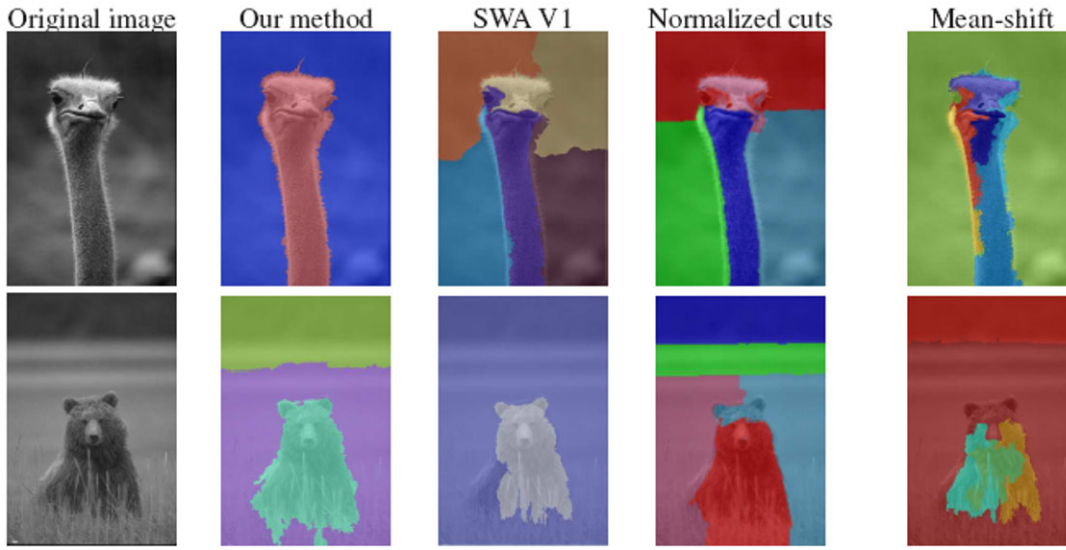


Figure 5.22 Comparative segmentation results (Alpert, Galun, Basri *et al.* 2007) © 2007 IEEE. "Our method" refers to the probabilistic bottom-up merging algorithm developed by Alpert *et al*.

5.5 Graph cuts and energy-based methods

A common theme in image segmentation algorithms is the desire to group pixels that have similar appearance (statistics) and to have the boundaries between pixels in different regions be of short length and across visible discontinuities. If we restrict the boundary measurements to be between immediate neighbors and compute region membership statistics by summing over pixels, we can formulate this as a classic pixel-based energy function using either a *variational formulation* (regularization, see Section 3.7.1) or as a binary Markov random field (Section 3.7.2).

Examples of the continuous approach include (Mumford and Shah 1989; Chan and Vese 1992; Zhu and Yuille 1996; Tabb and Ahuja 1997) along with the level set approaches discussed in Section 5.1.4. An early example of a discrete labeling problem that combines both region-based and boundary-based energy terms is the work of Leclerc (1989), who used minimum description length (MDL) coding to derive the energy function being minimized. Boykov and Funka-Lea (2006) present a wonderful survey of various energy-based techniques for binary object segmentation, some of which we discuss below.

As we saw in Section 3.7.2, the energy corresponding to a segmentation problem can be written (c.f. Equations (3.100) and (3.108–3.113)) as

$$E(f) = \sum_{i,j} E_r(i,j) + E_b(i,j), \tag{5.50}$$

where the region term

$$E_r(i,j) = E_S(I(i,j); R(f(i,j))) \tag{5.51}$$

is the negative log likelihood that pixel intensity (or color) $I(i,j)$ is consistent with the statis-

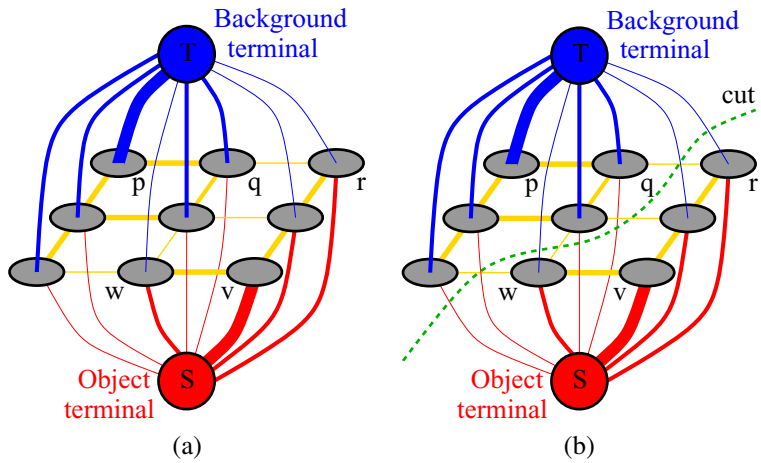


Figure 5.23 Graph cuts for region segmentation (Boykov and Jolly 2001) © 2001 IEEE: (a) the energy function is encoded as a maximum flow problem; (b) the minimum cut determines the region boundary.

tics of region $R(f(i,j))$ and the boundary term

$$E_b(i,j) = s_x(i,j)\delta(f(i,j) - f(i+1,j)) + s_y(i,j)\delta(f(i,j) - f(i,j+1)) \tag{5.52}$$

measures the inconsistency between $\mathcal{N}_4$ neighbors modulated by local horizontal and vertical smoothness terms $s_x(i,j)$ and $s_y(i,j)$.

Region statistics can be something as simple as the mean gray level or color (Leclerc 1989), in which case

$$E_S(I;\mu_k) = \|I - \mu_k\|^2. \tag{5.53}$$

Alternatively, they can be more complex, such as region intensity histograms (Boykov and Jolly 2001) or color Gaussian mixture models (Rother, Kolmogorov, and Blake 2004). For smoothness (boundary) terms, it is common to make the strength of the smoothness $s_x(i,j)$ inversely proportional to the local edge strength (Boykov, Veksler, and Zabih 2001).

Originally, energy-based segmentation problems were optimized using iterative gradient descent techniques, which were slow and prone to getting trapped in local minima. Boykov and Jolly (2001) were the first to apply the binary MRF optimization algorithm developed by Greig, Porteous, and Seheult (1989) to binary object segmentation.

In this approach, the user first delineates pixels in the background and foreground regions using a few strokes of an image brush (Figure 3.61). These pixels then become the *seeds* that tie nodes in the *S–T graph* to the source and sink labels S and T (Figure 5.23a). Seed pixels can also be used to estimate foreground and background region statistics (intensity or color histograms).

The capacities of the other edges in the graph are derived from the region and boundary energy terms, i.e., pixels that are more compatible with the foreground or background region get stronger connections to the respective source or sink; adjacent pixels with greater smoothness also get stronger links. Once the minimum-cut/maximum-flow problem has been solved using a polynomial time algorithm (Goldberg and Tarjan 1988; Boykov and Kolmogorov 2004), pixels on either side of the computed cut are labeled according to the source or sink to



(a) (b) (c)

Figure 5.24 GrabCut image segmentation (Rother, Kolmogorov, and Blake 2004) © 2004 ACM: (a) the user draws a bounding box in red; (b) the algorithm guesses color distributions for the object and background and performs a binary segmentation; (c) the process is repeated with better region statistics.

which they remain connected (Figure 5.23b). While graph cuts is just one of several known techniques for MRF energy minimization (Appendix B.5.4), it is still the one most commonly used for solving binary MRF problems.

The basic binary segmentation algorithm of Boykov and Jolly (2001) has been extended in a number of directions. The *GrabCut* system of Rother, Kolmogorov, and Blake (2004) iteratively re-estimates the region statistics, which are modeled as a mixtures of Gaussians in color space. This allows their system to operate given minimal user input, such as a single bounding box (Figure 5.24a)—the background color model is initialized from a strip of pixels around the box outline. (The foreground color model is initialized from the interior pixels, but quickly converges to a better estimate of the object.) The user can also place additional strokes to refine the segmentation as the solution progresses. In more recent work, Cui, Yang, Wen *et al.* (2008) use color and edge models derived from previous segmentations of similar objects to improve the local models used in GrabCut.

Another major extension to the original binary segmentation formulation is the addition of *directed edges*, which allows boundary regions to be oriented, e.g., to prefer light to dark transitions or *vice versa* (Kolmogorov and Boykov 2005). Figure 5.25 shows an example where the directed graph cut correctly segments the light gray liver from its dark gray surround. The same approach can be used to measure the *flux* exiting a region, i.e., the signed gradient projected normal to the region boundary. Combining oriented graphs with larger neighborhoods enables approximating continuous problems such as those traditionally solved using level sets in the globally optimal graph cut framework (Boykov and Kolmogorov 2003; Kolmogorov and Boykov 2005).

Even more recent developments in graph cut-based segmentation techniques include the addition of connectivity priors to force the foreground to be in a single piece (Vicente, Kolmogorov, and Rother 2008) and shape priors to use knowledge about an object's shape during the segmentation process (Lempitsky and Boykov 2007; Lempitsky, Blake, and Rother 2008).

While optimizing the binary MRF energy (5.50) requires the use of combinatorial optimization techniques, such as maximum flow, an approximate solution can be obtained by converting the binary energy terms into quadratic energy terms defined over a continuous $[0, 1]$ random field, which then becomes a classical membrane-based regularization problem (3.100–3.102). The resulting quadratic energy function can then be solved using standard linear system solvers (3.102–3.103), although if speed is an issue, you should use multigrid

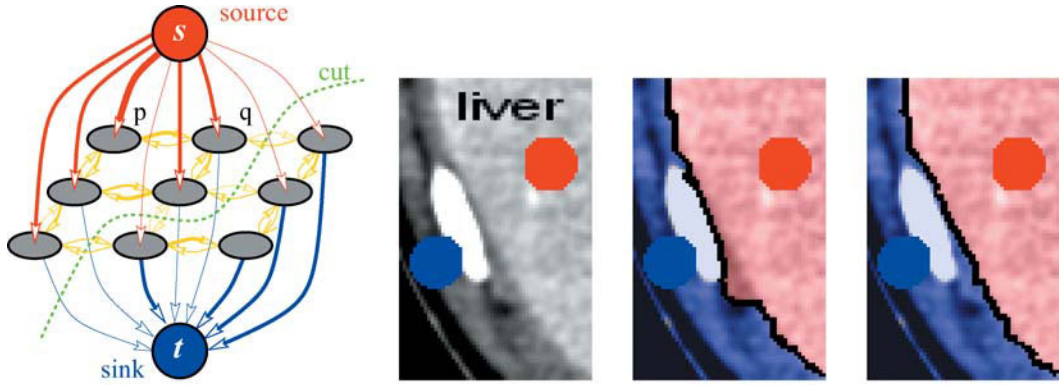
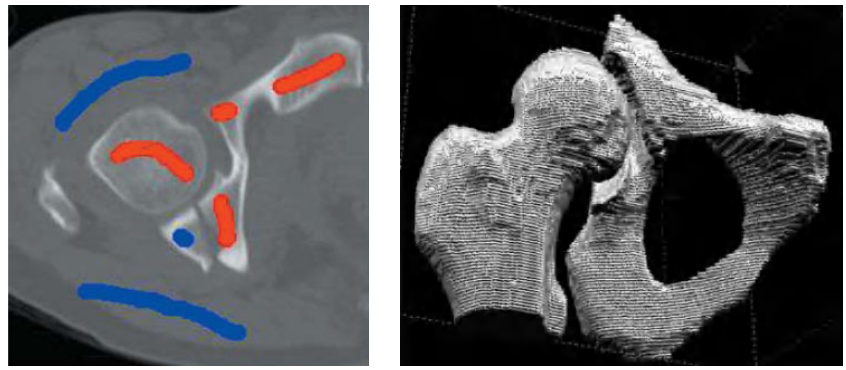


Figure 5.25 Segmentation with a directed graph cut (Boykov and Funka-Lea 2006) © 2006 Springer: (a) directed graph; (b) image with seed points; (c) the undirected graph incorrectly continues the boundary along the bright object; (d) the directed graph correctly segments the light gray region from its darker surround.

or one of its variants (Appendix A.5). Once the continuous solution has been computed, it can be thresholded at 0.5 to yield a binary segmentation.

The $[0, 1]$ continuous optimization problem can also be interpreted as computing the probability at each pixel that a *random walker* starting at that pixel ends up at one of the labeled seed pixels, which is also equivalent to computing the potential in a resistive grid where the resistors are equal to the edge weights (Grady 2006; Sinop and Grady 2007). K-way segmentations can also be computed by iterating through the seed labels, using a binary problem with one label set to 1 and all the others set to 0 to compute the relative membership probabilities for each pixel. In follow-on work, Grady and Ali (2008) use a precomputation of the eigenvectors of the linear system to make the solution with a novel set of seeds faster, which is related to the Laplacian matting problem presented in Section 10.4.3 (Levin, Acha, and Lischinski 2008). Couprie, Grady, Najman *et al.* (2009) relate the random walker to watersheds and other segmentation techniques. Singaraju, Grady, and Vidal (2008) add directed-edge constraints in order to support flux, which makes the energy piecewise quadratic and hence not solvable as a single linear system. The random walker algorithm can also be used to solve the Mumford–Shah segmentation problem (Grady and Alvino 2008) and to compute fast multigrid solutions (Grady 2008). A nice review of these techniques is given by Singaraju, Grady, Sinop *et al.* (2010).

An even faster way to compute a continuous $[0, 1]$ approximate segmentation is to compute *weighted geodesic distances* between the 0 and 1 seed regions (Bai and Sapiro 2009), which can also be used to estimate soft alpha mattes (Section 10.4.3). A related approach by Criminisi, Sharp, and Blake (2008) can be used to find fast approximate solutions to general binary Markov random field optimization problems.



(a) (b)

Figure 5.26 3D volumetric medical image segmentation using graph cuts (Boykov and Funka-Lea 2006) © 2006 Springer: (a) computed tomography (CT) slice with some seeds; (b) recovered 3D volumetric bone model (on a $256 \times 256 \times 119$ voxel grid).

5.5.1 *Application*: Medical image segmentation

One of the most promising applications of image segmentation is in the medical imaging domain, where it can be used to segment anatomical tissues for later quantitative analysis. Figure 5.25 shows a binary graph cut with directed edges being used to segment the liver tissue (light gray) from its surrounding bone (white) and muscle (dark gray) tissue. Figure 5.26 shows the segmentation of bones in a $256 \times 256 \times 119$ computed X-ray tomography (CT) volume. Without the powerful optimization techniques available in today's image segmentation algorithms, such processing used to require much more laborious manual tracing of individual X-ray slices.

The fields of medical image segmentation (McInerney and Terzopoulos 1996) and medical image registration (Kybic and Unser 2003) (Section 8.3.1) are rich research fields with their own specialized conferences, such as *Medical Imaging Computing and Computer Assisted Intervention (MICCAI)*,[11] and journals, such as *Medical Image Analysis* and *IEEE Transactions on Medical Imaging*. These can be great sources of references and ideas for research in this area.

5.6 Additional reading

The topic of image segmentation is closely related to clustering techniques, which are treated in a number of monographs and review articles (Jain and Dubes 1988; Kaufman and Rousseeuw 1990; Jain, Duin, and Mao 2000; Jain, Topchy, Law *et al.* 2004). Some early segmentation techniques include those describerd by Brice and Fennema (1970); Pavlidis (1977); Riseman and Arbib (1977); Ohlander, Price, and Reddy (1978); Rosenfeld and Davis (1979); Haralick and Shapiro (1985), while examples of newer techniques are developed by Leclerc (1989); Mumford and Shah (1989); Shi and Malik (2000); Felzenszwalb and Huttenlocher (2004b).

[11] http://www.miccai.org/.

Arbeláez, Maire, Fowlkes *et al.* (2010) provide a good review of automatic segmentation techniques and also compare their performance on the Berkeley Segmentation Dataset and Benchmark (Martin, Fowlkes, Tal *et al.* 2001).[12] Additional comparison papers and databases include those by Unnikrishnan, Pantofaru, and Hebert (2007); Alpert, Galun, Basri *et al.* (2007); Estrada and Jepson (2009).

The topic of active contours has a long history, beginning with the seminal work on snakes and other energy-minimizing variational methods (Kass, Witkin, and Terzopoulos 1988; Cootes, Cooper, Taylor *et al.* 1995; Blake and Isard 1998), continuing through techniques such as intelligent scissors (Mortensen and Barrett 1995, 1999; Pérez, Blake, and Gangnet 2001), and culminating in level sets (Malladi, Sethian, and Vemuri 1995; Caselles, Kimmel, and Sapiro 1997; Sethian 1999; Paragios and Deriche 2000; Sapiro 2001; Osher and Paragios 2003; Paragios, Faugeras, Chan *et al.* 2005; Cremers, Rousson, and Deriche 2007; Rousson and Paragios 2008; Paragios and Sgallari 2009), which are currently the most widely used active contour methods.

Techniques for segmenting images based on local pixel similarities combined with aggregation or splitting methods include watersheds (Vincent and Soille 1991; Beare 2006; Arbeláez, Maire, Fowlkes *et al.* 2010), region splitting (Ohlander, Price, and Reddy 1978), region merging (Brice and Fennema 1970; Pavlidis and Liow 1990; Jain, Topchy, Law *et al.* 2004), as well as graph-based and probabilistic multi-scale approaches (Felzenszwalb and Huttenlocher 2004b; Alpert, Galun, Basri *et al.* 2007).

Mean-shift algorithms, which find modes (peaks) in a density function representation of the pixels, are presented by Comaniciu and Meer (2002); Paris and Durand (2007). Parametric mixtures of Gaussians can also be used to represent and segment such pixel densities (Bishop 2006; Ma, Derksen, Hong *et al.* 2007).

The seminal work on spectral (eigenvalue) methods for image segmentation is the *normalized cut* algorithm of Shi and Malik (2000). Related work includes that by Weiss (1999); Meilă and Shi (2000, 2001); Malik, Belongie, Leung *et al.* (2001); Ng, Jordan, and Weiss (2001); Yu and Shi (2003); Cour, Bénézit, and Shi (2005); Sharon, Galun, Sharon *et al.* (2006); Tolliver and Miller (2006); Wang and Oliensis (2010).

Continuous-energy-based (variational) approaches to interactive segmentation include Leclerc (1989); Mumford and Shah (1989); Chan and Vese (1992); Zhu and Yuille (1996); Tabb and Ahuja (1997). Discrete variants of such problems are usually optimized using binary graph cuts or other combinatorial energy minimization methods (Boykov and Jolly 2001; Boykov and Kolmogorov 2003; Rother, Kolmogorov, and Blake 2004; Kolmogorov and Boykov 2005; Cui, Yang, Wen *et al.* 2008; Vicente, Kolmogorov, and Rother 2008; Lempitsky and Boykov 2007; Lempitsky, Blake, and Rother 2008), although continuous optimization techniques followed by thresholding can also be used (Grady 2006; Grady and Ali 2008; Singaraju, Grady, and Vidal 2008; Criminisi, Sharp, and Blake 2008; Grady 2008; Bai and Sapiro 2009; Couprie, Grady, Najman *et al.* 2009). Boykov and Funka-Lea (2006) present a good survey of various energy-based techniques for binary object segmentation.

[12] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/.

5.7 Exercises

Ex 5.1: Snake evolution Prove that, in the absence of external forces, a snake will always shrink to a small circle and eventually a single point, regardless of whether first- or second-order smoothness (or some combination) is used.

(Hint: If you can show that the evolution of the $x(s)$ and $y(s)$ components are independent, you can analyze the 1D case more easily.)

Ex 5.2: Snake tracker Implement a snake-based contour tracker:

1. Decide whether to use a large number of contour points or a smaller number interpolated with a B-spline.
2. Define your internal smoothness energy function and decide what image-based attractive forces to use.
3. At each iteration, set up the banded linear system of equations (quadratic energy function) and solve it using banded Cholesky factorization (Appendix A.4).

Ex 5.3: Intelligent scissors Implement the intelligent scissors (live-wire) interactive segmentation algorithm (Mortensen and Barrett 1995) and design a graphical user interface (GUI) to let you draw such curves over an image and use them for segmentation.

Ex 5.4: Region segmentation Implement one of the region segmentation algorithms described in this chapter. Some popular segmentation algorithms include:

- k-means (Section 5.3.1);
- mixtures of Gaussians (Section 5.3.1);
- mean shift (Section 5.3.2) and Exercise 5.5;
- normalized cuts (Section 5.4);
- similarity graph-based segmentation (Section 5.2.4);
- binary Markov random fields solved using graph cuts (Section 5.5).

Apply your region segmentation to a video sequence and use it to track moving regions from frame to frame.

Alternatively, test out your segmentation algorithm on the Berkeley segmentation database (Martin, Fowlkes, Tal *et al.* 2001).

Ex 5.5: Mean shift Develop a mean-shift segmentation algorithm for color images (Comaniciu and Meer 2002).

1. Convert your image to L*a*b* space, or keep the original RGB colors, and augment them with the pixel (x, y) locations.
2. For every pixel (L, a, b, x, y), compute the weighted mean of its neighbors using either a unit ball (Epanechnikov kernel) or finite-radius Gaussian, or some other kernel of your choosing. Weight the color and spatial scales differently, e.g., using values of $(h_s, h_r, M) = (16, 19, 40)$ as shown in Figure 5.18.

3. Replace the current value with this weighted mean and iterate until either the motion is below a threshold or a finite number of steps has been taken.

4. Cluster all final values (modes) that are within a threshold, i.e., find the connected components. Since each pixel is associated with a final mean-shift (mode) value, this results in an image segmentation, i.e., each pixel is labeled with its final component.

5. (Optional) Use a random subset of the pixels as starting points and find which component each unlabeled pixel belongs to, either by finding its nearest neighbor or by iterating the mean shift until it finds a neighboring track of mean-shift values. Describe the data structures you use to make this efficient.

6. (Optional) Mean shift divides the kernel density function estimate by the local weighting to obtain a step size that is guaranteed to converge but may be slow. Use an alternative step size estimation algorithm from the optimization literature to see if you can make the algorithm converge faster.

Chapter 6

Feature-based alignment

6.1 2D and 3D feature-based alignment 275
6.1.1 2D alignment using least squares 275
6.1.2 *Application*: Panography 277
6.1.3 Iterative algorithms 278
6.1.4 Robust least squares and RANSAC 281
6.1.5 3D alignment 283
6.2 Pose estimation 284
6.2.1 Linear algorithms 284
6.2.2 Iterative algorithms 286
6.2.3 *Application*: Augmented reality 287
6.3 Geometric intrinsic calibration 288
6.3.1 Calibration patterns 289
6.3.2 Vanishing points 290
6.3.3 *Application*: Single view metrology 292
6.3.4 Rotational motion 293
6.3.5 Radial distortion 295
6.4 Additional reading 296
6.5 Exercises 296

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_6, © Springer-Verlag London Limited 2011

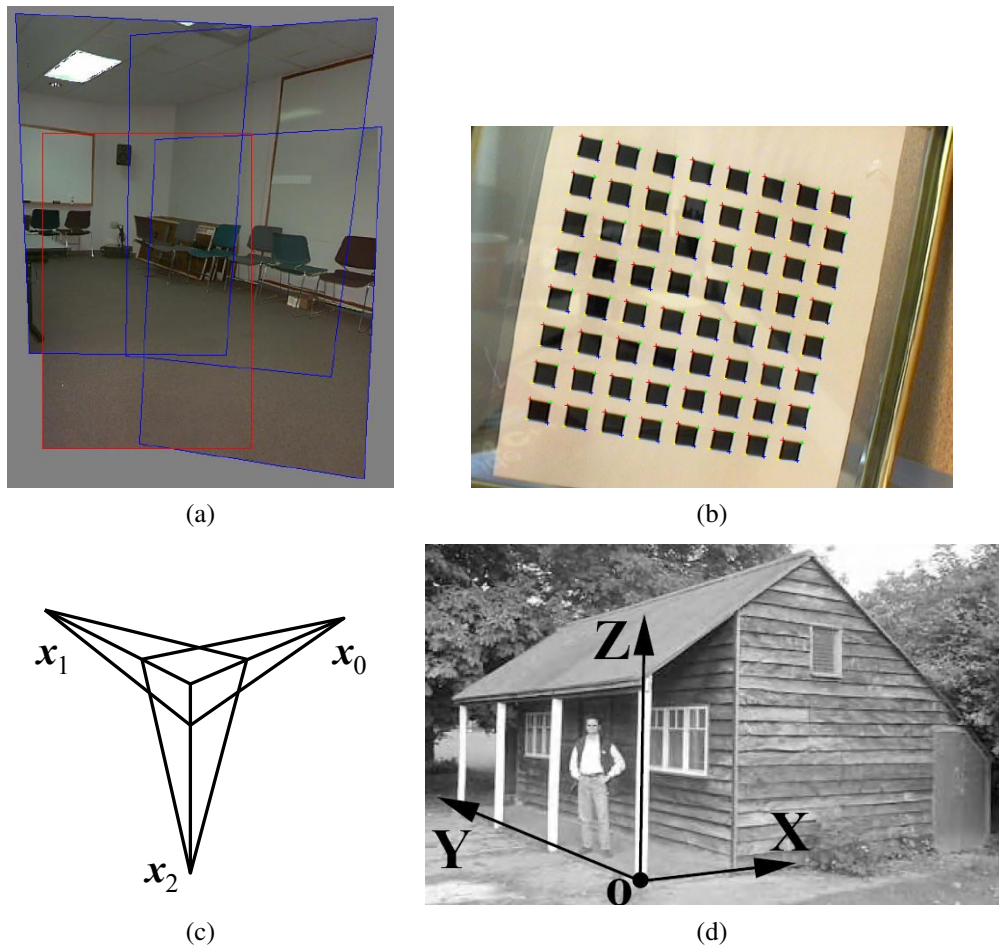


Figure 6.1 Geometric alignment and calibration: (a) geometric alignment of 2D images for stitching (Szeliski and Shum 1997) © 1997 ACM; (b) a two-dimensional calibration target (Zhang 2000) © 2000 IEEE; (c) calibration from vanishing points; (d) scene with easy-to-find lines and vanishing directions (Criminisi, Reid, and Zisserman 2000) © 2000 Springer.

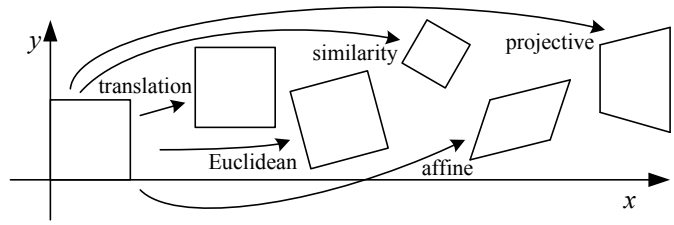


Figure 6.2 Basic set of 2D planar transformations

Once we have extracted features from images, the next stage in many vision algorithms is to match these features across different images (Section 4.1.3). An important component of this matching is to verify whether the set of matching features is geometrically consistent, e.g., whether the feature displacements can be described by a simple 2D or 3D geometric transformation. The computed motions can then be used in other applications such as image stitching (Chapter 9) or augmented reality (Section 6.2.3).

In this chapter, we look at the topic of geometric image registration, i.e., the computation of 2D and 3D transformations that map features in one image to another (Section 6.1). One special case of this problem is *pose estimation*, which is determining a camera's position relative to a known 3D object or scene (Section 6.2). Another case is the computation of a camera's *intrinsic calibration*, which consists of the internal parameters such as focal length and radial distortion (Section 6.3). In Chapter 7, we look at the related problems of how to estimate 3D point structure from 2D matches (*triangulation*) and how to simultaneously estimate 3D geometry and camera motion (*structure from motion*).

6.1 2D and 3D feature-based alignment

Feature-based alignment is the problem of estimating the motion between two or more sets of matched 2D or 3D points. In this section, we restrict ourselves to global *parametric* transformations, such as those described in Section 2.1.2 and shown in Table 2.1 and Figure 6.2, or higher order transformation for curved surfaces (Shashua and Toelg 1997; Can, Stewart, Roysam *et al.* 2002). Applications to non-rigid or elastic deformations (Bookstein 1989; Szeliski and Lavallée 1996; Torresani, Hertzmann, and Bregler 2008) are examined in Sections 8.3 and 12.6.4.

6.1.1 2D alignment using least squares

Given a set of matched feature points $\{(\boldsymbol{x}_i, \boldsymbol{x}'_i)\}$ and a planar parametric transformation[1] of the form

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{p}), \tag{6.1}$$

[1] For examples of non-planar parametric models, such as quadrics, see the work of Shashua and Toelg (1997); Shashua and Wexler (2001).

Transform	Matrix	Parameters p	Jacobian J
translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$	(t_x, t_y)	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Euclidean	$\begin{bmatrix} c_\theta & -s_\theta & t_x \\ s_\theta & c_\theta & t_y \end{bmatrix}$	(t_x, t_y, θ)	$\begin{bmatrix} 1 & 0 & -s_\theta x - c_\theta y \\ 0 & 1 & c_\theta x - s_\theta y \end{bmatrix}$
similarity	$\begin{bmatrix} 1+a & -b & t_x \\ b & 1+a & t_y \end{bmatrix}$	(t_x, t_y, a, b)	$\begin{bmatrix} 1 & 0 & x & -y \\ 0 & 1 & y & x \end{bmatrix}$
affine	$\begin{bmatrix} 1+a_{00} & a_{01} & t_x \\ a_{10} & 1+a_{11} & t_y \end{bmatrix}$	$(t_x, t_y, a_{00}, a_{01}, a_{10}, a_{11})$	$\begin{bmatrix} 1 & 0 & x & y & 0 & 0 \\ 0 & 1 & 0 & 0 & x & y \end{bmatrix}$
projective	$\begin{bmatrix} 1+h_{00} & h_{01} & h_{02} \\ h_{10} & 1+h_{11} & h_{12} \\ h_{20} & h_{21} & 1 \end{bmatrix}$	$(h_{00}, h_{01}, \ldots, h_{21})$	(see Section 6.1.3)

Table 6.1 Jacobians of the 2D coordinate transformations $\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{p})$ shown in Table 2.1, where we have re-parameterized the motions so that they are identity for $\boldsymbol{p} = 0$.

how can we produce the best estimate of the motion parameters $\boldsymbol{p}$? The usual way to do this is to use least squares, i.e., to minimize the sum of squared residuals

$$E_{\mathrm{LS}} = \sum_i \|\boldsymbol{r}_i\|^2 = \sum_i \|\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}'_i\|^2, \tag{6.2}$$

where

$$\boldsymbol{r}_i = \boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}'_i = \hat{\boldsymbol{x}}'_i - \tilde{\boldsymbol{x}}'_i \tag{6.3}$$

is the *residual* between the measured location $\hat{\boldsymbol{x}}'_i$ and its corresponding current *predicted* location $\tilde{\boldsymbol{x}}'_i = \boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p})$. (See Appendix A.2 for more on least squares and Appendix B.2 for a statistical justification.)

Many of the motion models presented in Section 2.1.2 and Table 2.1, i.e., translation, similarity, and affine, have a *linear* relationship between the amount of motion $\Delta\boldsymbol{x} = \boldsymbol{x}' - \boldsymbol{x}$ and the unknown parameters $\boldsymbol{p}$,

$$\Delta\boldsymbol{x} = \boldsymbol{x}' - \boldsymbol{x} = \boldsymbol{J}(\boldsymbol{x})\boldsymbol{p}, \tag{6.4}$$

where $\boldsymbol{J} = \partial\boldsymbol{f}/\partial\boldsymbol{p}$ is the *Jacobian* of the transformation $\boldsymbol{f}$ with respect to the motion parameters $\boldsymbol{p}$ (see Table 6.1). In this case, a simple *linear* regression (linear least squares problem) can be formulated as

$$\begin{aligned} E_{\mathrm{LLS}} &= \sum_i \|\boldsymbol{J}(\boldsymbol{x}_i)\boldsymbol{p} - \Delta\boldsymbol{x}_i\|^2 && (6.5) \\ &= \boldsymbol{p}^T \left[\sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\boldsymbol{J}(\boldsymbol{x}_i)\right] \boldsymbol{p} - 2\boldsymbol{p}^T \left[\sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\Delta\boldsymbol{x}_i\right] + \sum_i \|\Delta\boldsymbol{x}_i\|^2 && (6.6) \\ &= \boldsymbol{p}^T \boldsymbol{A}\boldsymbol{p} - 2\boldsymbol{p}^T \boldsymbol{b} + c. && (6.7) \end{aligned}$$

The minimum can be found by solving the symmetric positive definite (SPD) system of *normal equations*[2]

$$\boldsymbol{A}\boldsymbol{p} = \boldsymbol{b}, \tag{6.8}$$

where

$$\boldsymbol{A} = \sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\boldsymbol{J}(\boldsymbol{x}_i) \tag{6.9}$$

is called the *Hessian* and $\boldsymbol{b} = \sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\Delta\boldsymbol{x}_i$. For the case of pure translation, the resulting equations have a particularly simple form, i.e., the translation is the average translation between corresponding points or, equivalently, the translation of the point centroids.

Uncertainty weighting. The above least squares formulation assumes that all feature points are matched with the same accuracy. This is often not the case, since certain points may fall into more textured regions than others. If we associate a scalar variance estimate σ_i^2 with each correspondence, we can minimize the *weighted least squares* problem instead,[3]

$$E_{\mathrm{WLS}} = \sum_i \sigma_i^{-2}\|\boldsymbol{r}_i\|^2. \tag{6.10}$$

As shown in Section 8.1.3, a covariance estimate for patch-based matching can be obtained by multiplying the inverse of the *patch Hessian* $\boldsymbol{A}_i$ (8.55) with the per-pixel noise covariance σ_n^2 (8.44). Weighting each squared residual by its inverse covariance $\Sigma_i^{-1} = \sigma_n^{-2}\boldsymbol{A}_i$ (which is called the *information matrix*), we obtain

$$E_{\mathrm{CWLS}} = \sum_i \|\boldsymbol{r}_i\|^2_{\Sigma_i^{-1}} = \sum_i \boldsymbol{r}_i^T\Sigma_i^{-1}\boldsymbol{r}_i = \sum_i \sigma_n^{-2}\boldsymbol{r}_i^T\boldsymbol{A}_i\boldsymbol{r}_i. \tag{6.11}$$

6.1.2 *Application*: Panography

One of the simplest (and most fun) applications of image alignment is a special form of image stitching called *panography*. In a panograph, images are translated and optionally rotated and scaled before being blended with simple averaging (Figure 6.3). This process mimics the photographic collages created by artist David Hockney, although his compositions use an opaque overlay model, being created out of regular photographs.

In most of the examples seen on the Web, the images are aligned by hand for best artistic effect.[4] However, it is also possible to use feature matching and alignment techniques to perform the registration automatically (Nomura, Zhang, and Nayar 2007; Zelnik-Manor and Perona 2007).

Consider a simple translational model. We want all the corresponding features in different images to line up as best as possible. Let $\boldsymbol{t}_j$ be the location of the jth image coordinate frame in the global composite frame and $\boldsymbol{x}_{ij}$ be the location of the ith matched feature in the jth image. In order to align the images, we wish to minimize the least squares error

$$E_{\mathrm{PLS}} = \sum_{ij} \|(\boldsymbol{t}_j + \boldsymbol{x}_{ij}) - \boldsymbol{x}_i\|^2, \tag{6.12}$$

[2] For poorly conditioned problems, it is better to use QR decomposition on the set of linear equations $\boldsymbol{J}(\boldsymbol{x}_i)\boldsymbol{p} = \Delta\boldsymbol{x}_i$ instead of the normal equations (Björck 1996; Golub and Van Loan 1996). However, such conditions rarely arise in image registration.

[3] Problems where each measurement can have a different variance or certainty are called *heteroscedastic models*.

[4] http://www.flickr.com/groups/panography/.



Figure 6.3 A simple panograph consisting of three images automatically aligned with a translational model and then averaged together.

where $\boldsymbol{x}_i$ is the consensus (average) position of feature i in the global coordinate frame. (An alternative approach is to register each pair of overlapping images separately and then compute a consensus location for each frame—see Exercise 6.2.)

The above least squares problem is indeterminate (you can add a constant offset to all the frame and point locations $\boldsymbol{t}_j$ and $\boldsymbol{x}_i$). To fix this, either pick one frame as being at the origin or add a constraint to make the average frame offsets be 0.

The formulas for adding rotation and scale transformations are straightforward and are left as an exercise (Exercise 6.2). See if you can create some collages that you would be happy to share with others on the Web.

6.1.3 Iterative algorithms

While linear least squares is the simplest method for estimating parameters, most problems in computer vision do not have a simple linear relationship between the measurements and the unknowns. In this case, the resulting problem is called *non-linear least squares* or *non-linear regression*.

Consider, for example, the problem of estimating a rigid Euclidean 2D transformation (translation plus rotation) between two sets of points. If we parameterize this transformation by the translation amount (t_x, t_y) and the rotation angle θ, as in Table 2.1, the Jacobian of this transformation, given in Table 6.1, depends on the current value of θ. Notice how in Table 6.1, we have re-parameterized the motion matrices so that they are always the identity at the origin $\boldsymbol{p} = 0$, which makes it easier to initialize the motion parameters.

To minimize the non-linear least squares problem, we iteratively find an update $\Delta\boldsymbol{p}$ to the current parameter estimate $\boldsymbol{p}$ by minimizing

$$E_{\mathrm{NLS}}(\Delta\boldsymbol{p}) = \sum_i \|\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p} + \Delta\boldsymbol{p}) - \boldsymbol{x}'_i\|^2 \tag{6.13}$$

$$\approx \sum_i \|\boldsymbol{J}(\boldsymbol{x}_i; \boldsymbol{p})\Delta\boldsymbol{p} - \boldsymbol{r}_i\|^2 \tag{6.14}$$

$$= \Delta \boldsymbol{p}^T \left[\sum_i \boldsymbol{J}^T \boldsymbol{J} \right] \Delta \boldsymbol{p} - 2\Delta \boldsymbol{p}^T \left[\sum_i \boldsymbol{J}^T \boldsymbol{r}_i \right] + \sum_i \|\boldsymbol{r}_i\|^2 \quad (6.15)$$

$$= \Delta \boldsymbol{p}^T \boldsymbol{A} \Delta \boldsymbol{p} - 2\Delta \boldsymbol{p}^T \boldsymbol{b} + c, \quad (6.16)$$

where the "Hessian"[5] $\boldsymbol{A}$ is the same as Equation (6.9) and the right hand side vector

$$\boldsymbol{b} = \sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\boldsymbol{r}_i \quad (6.17)$$

is now a Jacobian-weighted sum of residual vectors. This makes intuitive sense, as the parameters are pulled in the direction of the prediction error with a strength proportional to the Jacobian.

Once $\boldsymbol{A}$ and $\boldsymbol{b}$ have been computed, we solve for $\Delta \boldsymbol{p}$ using

$$(\boldsymbol{A} + \lambda \text{diag}(\boldsymbol{A}))\Delta \boldsymbol{p} = \boldsymbol{b}, \quad (6.18)$$

and update the parameter vector $\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta \boldsymbol{p}$ accordingly. The parameter λ is an additional damping parameter used to ensure that the system takes a "downhill" step in energy (squared error) and is an essential component of the Levenberg–Marquardt algorithm (described in more detail in Appendix A.3). In many applications, it can be set to 0 if the system is successfully converging.

For the case of our 2D translation+rotation, we end up with a 3×3 set of normal equations in the unknowns $(\delta t_x, \delta t_y, \delta\theta)$. An initial guess for (t_x, t_y, θ) can be obtained by fitting a four-parameter similarity transform in (t_x, t_y, c, s) and then setting $\theta = \tan^{-1}(s/c)$. An alternative approach is to estimate the translation parameters using the centroids of the 2D points and to then estimate the rotation angle using polar coordinates (Exercise 6.3).

For the other 2D motion models, the derivatives in Table 6.1 are all fairly straightforward, except for the projective 2D motion (homography), which arises in image-stitching applications (Chapter 9). These equations can be re-written from (2.21) in their new parametric form as

$$x' = \frac{(1+h_{00})x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + 1} \text{ and } y' = \frac{h_{10}x + (1+h_{11})y + h_{12}}{h_{20}x + h_{21}y + 1}. \quad (6.19)$$

The Jacobian is therefore

$$\boldsymbol{J} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}} = \frac{1}{D} \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y \end{bmatrix}, \quad (6.20)$$

where $D = h_{20}x + h_{21}y + 1$ is the denominator in (6.19), which depends on the current parameter settings (as do x' and y').

An initial guess for the eight unknowns $\{h_{00}, h_{01}, \ldots, h_{21}\}$ can be obtained by multiplying both sides of the equations in (6.19) through by the denominator, which yields the linear set of equations,

$$\begin{bmatrix} \hat{x}' - x \\ \hat{y}' - y \end{bmatrix} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -\hat{x}'x & -\hat{x}'y \\ 0 & 0 & 0 & x & y & 1 & -\hat{y}'x & -\hat{y}'y \end{bmatrix} \begin{bmatrix} h_{00} \\ \vdots \\ h_{21} \end{bmatrix}. \quad (6.21)$$

[5] The "Hessian" $\boldsymbol{A}$ is not the true Hessian (second derivative) of the non-linear least squares problem (6.13). Instead, it is the approximate Hessian, which neglects second (and higher) order derivatives of $\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p} + \Delta \boldsymbol{p})$.

However, this is not optimal from a statistical point of view, since the denominator D, which was used to multiply each equation, can vary quite a bit from point to point.[6]

One way to compensate for this is to *reweight* each equation by the inverse of the current estimate of the denominator, D,

$$\frac{1}{D}\begin{bmatrix}\hat{x}'-x\\ \hat{y}'-y\end{bmatrix}=\frac{1}{D}\begin{bmatrix}x & y & 1 & 0 & 0 & 0 & -\hat{x}'x & -\hat{x}'y\\ 0 & 0 & 0 & x & y & 1 & -\hat{y}'x & -\hat{y}'y\end{bmatrix}\begin{bmatrix}h_{00}\\ \vdots\\ h_{21}\end{bmatrix}. \tag{6.22}$$

While this may at first seem to be the exact same set of equations as (6.21), because least squares is being used to solve the over-determined set of equations, the weightings *do* matter and produce a different set of normal equations that performs better in practice.

The most principled way to do the estimation, however, is to directly minimize the squared residual equations (6.13) using the Gauss–Newton approximation, i.e., performing a first-order Taylor series expansion in $\boldsymbol{p}$, as shown in (6.14), which yields the set of equations

$$\begin{bmatrix}\hat{x}'-\tilde{x}'\\ \hat{y}'-\tilde{y}'\end{bmatrix}=\frac{1}{D}\begin{bmatrix}x & y & 1 & 0 & 0 & 0 & -\tilde{x}'x & -\tilde{x}'y\\ 0 & 0 & 0 & x & y & 1 & -\tilde{y}'x & -\tilde{y}'y\end{bmatrix}\begin{bmatrix}\Delta h_{00}\\ \vdots\\ \Delta h_{21}\end{bmatrix}. \tag{6.23}$$

While these look similar to (6.22), they differ in two important respects. First, the left hand side consists of unweighted *prediction errors* rather than point displacements and the solution vector is a *perturbation* to the parameter vector $\boldsymbol{p}$. Second, the quantities inside $\boldsymbol{J}$ involve *predicted* feature locations $(\tilde{x}', \tilde{y}')$ instead of *sensed* feature locations $(\hat{x}', \hat{y}')$. Both of these differences are subtle and yet they lead to an algorithm that, when combined with proper checking for downhill steps (as in the Levenberg–Marquardt algorithm), will converge to a local minimum. Note that iterating Equations (6.22) is not guaranteed to converge, since it is not minimizing a well-defined energy function.

Equation (6.23) is analogous to the *additive* algorithm for direct intensity-based registration (Section 8.2), since the change to the full transformation is being computed. If we prepend an incremental homography to the current homography instead, i.e., we use a *compositional* algorithm (described in Section 8.2), we get $D = 1$ (since $\boldsymbol{p} = 0$) and the above formula simplifies to

$$\begin{bmatrix}\hat{x}'-x\\ \hat{y}'-y\end{bmatrix}=\begin{bmatrix}x & y & 1 & 0 & 0 & 0 & -x^2 & -xy\\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2\end{bmatrix}\begin{bmatrix}\Delta h_{00}\\ \vdots\\ \Delta h_{21}\end{bmatrix}, \tag{6.24}$$

where we have replaced $(\tilde{x}', \tilde{y}')$ with (x, y) for conciseness. (Notice how this results in the same Jacobian as (8.63).)

[6] Hartley and Zisserman (2004) call this strategy of forming linear equations from rational equations the *direct linear transform*, but that term is more commonly associated with pose estimation (Section 6.2). Note also that our definition of the h_{ij} parameters differs from that used in their book, since we define h_{ii} to be the *difference* from unity and we do not leave h_{22} as a free parameter, which means that we cannot handle certain extreme homographies.

6.1.4 Robust least squares and RANSAC

While regular least squares is the method of choice for measurements where the noise follows a normal (Gaussian) distribution, more robust versions of least squares are required when there are outliers among the correspondences (as there almost always are). In this case, it is preferable to use an *M-estimator* (Huber 1981; Hampel, Ronchetti, Rousseeuw *et al.* 1986; Black and Rangarajan 1996; Stewart 1999), which involves applying a robust penalty function $\rho(r)$ to the residuals

$$E_{\mathrm{RLS}}(\Delta \boldsymbol{p}) = \sum_i \rho(\|\boldsymbol{r}_i\|) \tag{6.25}$$

instead of squaring them.

We can take the derivative of this function with respect to $\boldsymbol{p}$ and set it to 0,

$$\sum_i \psi(\|\boldsymbol{r}_i\|) \frac{\partial \|\boldsymbol{r}_i\|}{\partial \boldsymbol{p}} = \sum_i \frac{\psi(\|\boldsymbol{r}_i\|)}{\|\boldsymbol{r}_i\|} \boldsymbol{r}_i^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{p}} = 0, \tag{6.26}$$

where $\psi(r) = \rho'(r)$ is the derivative of ρ and is called the *influence function*. If we introduce a *weight function*, $w(r) = \Psi(r)/r$, we observe that finding the stationary point of (6.25) using (6.26) is equivalent to minimizing the *iteratively reweighted least squares* (IRLS) problem

$$E_{\mathrm{IRLS}} = \sum_i w(\|\boldsymbol{r}_i\|) \|\boldsymbol{r}_i\|^2, \tag{6.27}$$

where the $w(\|\boldsymbol{r}_i\|)$ play the same local weighting role as σ_i^{-2} in (6.10). The IRLS algorithm alternates between computing the influence functions $w(\|\boldsymbol{r}_i\|)$ and solving the resulting weighted least squares problem (with fixed w values). Other incremental robust least squares algorithms can be found in the work of Sawhney and Ayer (1996); Black and Anandan (1996); Black and Rangarajan (1996); Baker, Gross, Ishikawa *et al.* (2003) and textbooks and tutorials on robust statistics (Huber 1981; Hampel, Ronchetti, Rousseeuw *et al.* 1986; Rousseeuw and Leroy 1987; Stewart 1999).

While M-estimators can definitely help reduce the influence of outliers, in some cases, starting with too many outliers will prevent IRLS (or other gradient descent algorithms) from converging to the global optimum. A better approach is often to find a starting set of *inlier* correspondences, i.e., points that are consistent with a dominant motion estimate.[7]

Two widely used approaches to this problem are called RANdom SAmple Consensus, or RANSAC for short (Fischler and Bolles 1981), and *least median of squares* (LMS) (Rousseeuw 1984). Both techniques start by selecting (at random) a subset of k correspondences, which is then used to compute an initial estimate for $\boldsymbol{p}$. The *residuals* of the full set of correspondences are then computed as

$$\boldsymbol{r}_i = \tilde{\boldsymbol{x}}_i'(\boldsymbol{x}_i; \boldsymbol{p}) - \hat{\boldsymbol{x}}_i', \tag{6.28}$$

where $\tilde{\boldsymbol{x}}_i'$ are the *estimated* (mapped) locations and $\hat{\boldsymbol{x}}_i'$ are the sensed (detected) feature point locations.

The RANSAC technique then counts the number of *inliers* that are within ϵ of their predicted location, i.e., whose $\|\boldsymbol{r}_i\| \leq \epsilon$. (The ϵ value is application dependent but is often around 1–3 pixels.) Least median of squares finds the median value of the $\|\boldsymbol{r}_i\|^2$ values. The

[7] For pixel-based alignment methods (Section 8.1.1), hierarchical (coarse-to-fine) techniques are often used to lock onto the *dominant motion* in a scene.

k	p	S
3	0.5	35
6	0.6	97
6	0.5	293

Table 6.2 Number of trials S to attain a 99% probability of success (Stewart 1999).

random selection process is repeated S times and the sample set with the largest number of inliers (or with the smallest median residual) is kept as the final solution. Either the initial parameter guess $\boldsymbol{p}$ or the full set of computed inliers is then passed on to the next data fitting stage.

When the number of measurements is quite large, it may be preferable to only score a subset of the measurements in an initial round that selects the most plausible hypotheses for additional scoring and selection. This modification of RANSAC, which can significantly speed up its performance, is called *Preemptive RANSAC* (Nistér 2003). In another variant on RANSAC called PROSAC (PROgressive SAmple Consensus), random samples are initially added from the most "confident" matches, thereby speeding up the process of finding a (statistically) likely good set of inliers (Chum and Matas 2005).

To ensure that the random sampling has a good chance of finding a true set of inliers, a sufficient number of trials S must be tried. Let p be the probability that any given correspondence is valid and P be the total probability of success after S trials. The likelihood in one trial that all k random samples are inliers is p^k. Therefore, the likelihood that S such trials will all fail is

$$1 - P = (1 - p^k)^S \tag{6.29}$$

and the required minimum number of trials is

$$S = \frac{\log(1 - P)}{\log(1 - p^k)}. \tag{6.30}$$

Stewart (1999) gives examples of the required number of trials S to attain a 99% probability of success. As you can see from Table 6.2, the number of trials grows quickly with the number of sample points used. This provides a strong incentive to use the *minimum* number of sample points k possible for any given trial, which is how RANSAC is normally used in practice.

Uncertainty modeling

In addition to robustly computing a good alignment, some applications require the computation of uncertainty (see Appendix B.6). For linear problems, this estimate can be obtained by inverting the Hessian matrix (6.9) and multiplying it by the feature position noise (if these have not already been used to weight the individual measurements, as in Equations (6.10) and 6.11)). In statistics, the Hessian, which is the inverse covariance, is sometimes called the (Fisher) *information matrix* (Appendix B.1.1).

When the problem involves non-linear least squares, the inverse of the Hessian matrix provides the *Cramer–Rao lower bound* on the covariance matrix, i.e., it provides the *minimum*

amount of covariance in a given solution, which can actually have a wider spread ("longer tails") if the energy flattens out away from the local minimum where the optimal solution is found.

6.1.5 3D alignment

Instead of aligning 2D sets of image features, many computer vision applications require the alignment of 3D points. In the case where the 3D transformations are linear in the motion parameters, e.g., for translation, similarity, and affine, regular least squares (6.5) can be used.

The case of rigid (Euclidean) motion,

$$E_{\mathrm{R3D}} = \sum_i \|\boldsymbol{x}'_i - \boldsymbol{R}\boldsymbol{x}_i - \boldsymbol{t}\|^2, \tag{6.31}$$

which arises more frequently and is often called the *absolute orientation* problem (Horn 1987), requires slightly different techniques. If only scalar weightings are being used (as opposed to full 3D per-point anisotropic covariance estimates), the weighted centroids of the two point clouds $\boldsymbol{c}$ and $\boldsymbol{c}'$ can be used to estimate the translation $\boldsymbol{t} = \boldsymbol{c}' - \boldsymbol{R}\boldsymbol{c}$.[8] We are then left with the problem of estimating the rotation between two sets of points $\{\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{c}\}$ and $\{\hat{\boldsymbol{x}}'_i = \boldsymbol{x}'_i - \boldsymbol{c}'\}$ that are both centered at the origin.

One commonly used technique is called the *orthogonal Procrustes algorithm* (Golub and Van Loan 1996, p. 601) and involves computing the singular value decomposition (SVD) of the 3×3 correlation matrix

$$\boldsymbol{C} = \sum_i \hat{\boldsymbol{x}}' \hat{\boldsymbol{x}}^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T. \tag{6.32}$$

The rotation matrix is then obtained as $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T$. (Verify this for yourself when $\hat{\boldsymbol{x}}' = \boldsymbol{R}\hat{\boldsymbol{x}}$.)

Another technique is the absolute orientation algorithm (Horn 1987) for estimating the unit quaternion corresponding to the rotation matrix $\boldsymbol{R}$, which involves forming a 4×4 matrix from the entries in $\boldsymbol{C}$ and then finding the eigenvector associated with its largest positive eigenvalue.

Lorusso, Eggert, and Fisher (1995) experimentally compare these two techniques to two additional techniques proposed in the literature, but find that the difference in accuracy is negligible (well below the effects of measurement noise).

In situations where these closed-form algorithms are not applicable, e.g., when full 3D covariances are being used or when the 3D alignment is part of some larger optimization, the incremental rotation update introduced in Section 2.1.4 (2.35–2.36), which is parameterized by an instantaneous rotation vector $\boldsymbol{\omega}$, can be used (See Section 9.1.3 for an application to image stitching.)

In some situations, e.g., when merging range data maps, the correspondence between data points is not known *a priori*. In this case, iterative algorithms that start by matching nearby points and then update the most likely correspondence can be used (Besl and McKay 1992; Zhang 1994; Szeliski and Lavallée 1996; Gold, Rangarajan, Lu *et al.* 1998; David, DeMenthon, Duraiswami *et al.* 2004; Li and Hartley 2007; Enqvist, Josephson, and Kahl 2009). These techniques are discussed in more detail in Section 12.2.1.

[8] When full covariances are used, they are transformed by the rotation and so a closed-form solution for translation is not possible.

6.2 Pose estimation

A particular instance of feature-based alignment, which occurs very often, is estimating an object's 3D pose from a set of 2D point projections. This *pose estimation* problem is also known as *extrinsic* calibration, as opposed to the *intrinsic* calibration of internal camera parameters such as focal length, which we discuss in Section 6.3. The problem of recovering pose from three correspondences, which is the minimal amount of information necessary, is known as the *perspective-3-point-problem* (P3P), with extensions to larger numbers of points collectively known as PnP (Haralick, Lee, Ottenberg *et al.* 1994; Quan and Lan 1999; Moreno-Noguer, Lepetit, and Fua 2007).

In this section, we look at some of the techniques that have been developed to solve such problems, starting with the *direct linear transform* (DLT), which recovers a 3×4 camera matrix, followed by other "linear" algorithms, and then looking at statistically optimal iterative algorithms.

6.2.1 Linear algorithms

The simplest way to recover the pose of the camera is to form a set of linear equations analogous to those used for 2D motion estimation (6.19) from the camera matrix form of perspective projection (2.55–2.56),

$$x_i = \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}} \tag{6.33}$$

$$y_i = \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}}, \tag{6.34}$$

where (x_i, y_i) are the measured 2D feature locations and (X_i, Y_i, Z_i) are the known 3D feature locations (Figure 6.4). As with (6.21), this system of equations can be solved in a linear fashion for the unknowns in the camera matrix $\boldsymbol{P}$ by multiplying the denominator on both sides of the equation.[9] The resulting algorithm is called the *direct linear transform* (DLT) and is commonly attributed to Sutherland (1974). (For a more in-depth discussion, refer to the work of Hartley and Zisserman (2004).) In order to compute the 12 (or 11) unknowns in $\boldsymbol{P}$, at least six correspondences between 3D and 2D locations must be known.

As with the case of estimating homographies (6.21–6.23), more accurate results for the entries in $\boldsymbol{P}$ can be obtained by directly minimizing the set of Equations (6.33–6.34) using non-linear least squares with a small number of iterations.

Once the entries in $\boldsymbol{P}$ have been recovered, it is possible to recover both the intrinsic calibration matrix $\boldsymbol{K}$ and the rigid transformation $(\boldsymbol{R}, \boldsymbol{t})$ by observing from Equation (2.56) that

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]. \tag{6.35}$$

Since $\boldsymbol{K}$ is by convention upper-triangular (see the discussion in Section 2.1.5), both $\boldsymbol{K}$ and $\boldsymbol{R}$ can be obtained from the front 3×3 sub-matrix of $\boldsymbol{P}$ using RQ factorization (Golub and Van Loan 1996).[10]

[9] Because $\boldsymbol{P}$ is unknown up to a scale, we can either fix one of the entries, e.g., $p_{23} = 1$, or find the smallest singular vector of the set of linear equations.

[10] Note the unfortunate clash of terminologies: In matrix algebra textbooks, $\boldsymbol{R}$ represents an upper-triangular matrix; in computer vision, $\boldsymbol{R}$ is an orthogonal rotation.

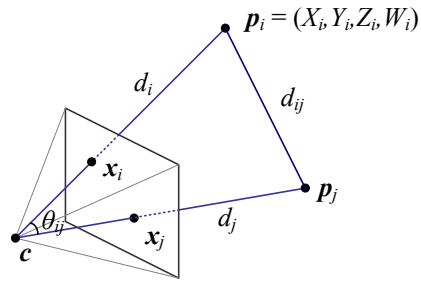


Figure 6.4 Pose estimation by the direct linear transform and by measuring visual angles and distances between pairs of points.

In most applications, however, we have some prior knowledge about the intrinsic calibration matrix $\boldsymbol{K}$, e.g., that the pixels are square, the skew is very small, and the optical center is near the center of the image (2.57–2.59). Such constraints can be incorporated into a non-linear minimization of the parameters in $\boldsymbol{K}$ and $(\boldsymbol{R}, \boldsymbol{t})$, as described in Section 6.2.2.

In the case where the camera is already calibrated, i.e., the matrix $\boldsymbol{K}$ is known (Section 6.3), we can perform pose estimation using as few as three points (Fischler and Bolles 1981; Haralick, Lee, Ottenberg *et al.* 1994; Quan and Lan 1999). The basic observation that these *linear PnP* (*perspective n-point*) algorithms employ is that the visual angle between any pair of 2D points $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{x}}_j$ must be the same as the angle between their corresponding 3D points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ (Figure 6.4).

Given a set of corresponding 2D and 3D points $\{(\hat{\boldsymbol{x}}_i, \boldsymbol{p}_i)\}$, where the $\hat{\boldsymbol{x}}_i$ are unit directions obtained by transforming 2D pixel measurements $\boldsymbol{x}_i$ to unit norm 3D directions $\hat{\boldsymbol{x}}_i$ through the inverse calibration matrix $\boldsymbol{K}$,

$$\hat{\boldsymbol{x}}_i = \mathcal{N}(\boldsymbol{K}^{-1}\boldsymbol{x}_i) = \boldsymbol{K}^{-1}\boldsymbol{x}_i / \|\boldsymbol{K}^{-1}\boldsymbol{x}_i\|, \tag{6.36}$$

the unknowns are the distances d_i from the camera origin $\boldsymbol{c}$ to the 3D points $\boldsymbol{p}_i$, where

$$\boldsymbol{p}_i = d_i\hat{\boldsymbol{x}}_i + \boldsymbol{c} \tag{6.37}$$

(Figure 6.4). The cosine law for triangle $\Delta(\boldsymbol{c}, \boldsymbol{p}_i, \boldsymbol{p}_j)$ gives us

$$f_{ij}(d_i, d_j) = d_i^2 + d_j^2 - 2d_i d_j c_{ij} - d_{ij}^2 = 0, \tag{6.38}$$

where

$$c_{ij} = \cos\theta_{ij} = \hat{\boldsymbol{x}}_i \cdot \hat{\boldsymbol{x}}_j \tag{6.39}$$

and

$$d_{ij}^2 = \|\boldsymbol{p}_i - \boldsymbol{p}_j\|^2. \tag{6.40}$$

We can take any triplet of constraints (f_{ij}, f_{ik}, f_{jk}) and eliminate the d_j and d_k using Sylvester resultants (Cox, Little, and O'Shea 2007) to obtain a quartic equation in d_i^2,

$$g_{ijk}(d_i^2) = a_4 d_i^8 + a_3 d_i^6 + a_2 d_i^4 + a_1 d_i^2 + a_0 = 0. \tag{6.41}$$

Given five or more correspondences, we can generate $\frac{(n-1)(n-2)}{2}$ triplets to obtain a linear estimate (using SVD) for the values of $(d_i^8, d_i^6, d_i^4, d_i^2)$ (Quan and Lan 1999). Estimates for

d_i^2 can computed as ratios of successive d_i^{2n+2}/d_i^{2n} estimates and these can be averaged to obtain a final estimate of d_i^2 (and hence d_i).

Once the individual estimates of the d_i distances have been computed, we can generate a 3D structure consisting of the scaled point directions $d_i\hat{\boldsymbol{x}}_i$, which can then be aligned with the 3D point cloud $\{\boldsymbol{p}_i\}$ using absolute orientation (Section 6.1.5) to obtained the desired pose estimate. Quan and Lan (1999) give accuracy results for this and other techniques, which use fewer points but require more complicated algebraic manipulations. The paper by Moreno-Noguer, Lepetit, and Fua (2007) reviews more recent alternatives and also gives a lower complexity algorithm that typically produces more accurate results.

Unfortunately, because minimal PnP solutions can be quite noise sensitive and also suffer from *bas-relief ambiguities* (e.g., depth reversals) (Section 7.4.3), it is often preferable to use the linear six-point algorithm to guess an initial pose and then optimize this estimate using the iterative technique described in Section 6.2.2.

An alternative pose estimation algorithm involves starting with a scaled orthographic projection model and then iteratively refining this initial estimate using a more accurate perspective projection model (DeMenthon and Davis 1995). The attraction of this model, as stated in the paper's title, is that it can be implemented "in 25 lines of [Mathematica] code".

6.2.2 Iterative algorithms

The most accurate (and flexible) way to estimate pose is to directly minimize the squared (or robust) reprojection error for the 2D points as a function of the unknown pose parameters in $(\boldsymbol{R}, \boldsymbol{t})$ and optionally $\boldsymbol{K}$ using non-linear least squares (Tsai 1987; Bogart 1991; Gleicher and Witkin 1992). We can write the projection equations as

$$\boldsymbol{x}_i = \boldsymbol{f}(\boldsymbol{p}_i; \boldsymbol{R}, \boldsymbol{t}, \boldsymbol{K}) \tag{6.42}$$

and iteratively minimize the robustified linearized reprojection errors

$$E_{\mathrm{NLP}} = \sum_i \rho\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{R}}\Delta\boldsymbol{R} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{t}}\Delta\boldsymbol{t} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{K}}\Delta\boldsymbol{K} - \boldsymbol{r}_i\right), \tag{6.43}$$

where $\boldsymbol{r}_i = \tilde{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_i$ is the current residual vector (2D error in predicted position) and the partial derivatives are with respect to the unknown pose parameters (rotation, translation, and optionally calibration). Note that if full 2D covariance estimates are available for the 2D feature locations, the above squared norm can be weighted by the inverse point covariance matrix, as in Equation (6.11).

An easier to understand (and implement) version of the above non-linear regression problem can be constructed by re-writing the projection equations as a concatenation of simpler steps, each of which transforms a 4D homogeneous coordinate $\boldsymbol{p}_i$ by a simple transformation such as translation, rotation, or perspective division (Figure 6.5). The resulting projection equations can be written as

$$\boldsymbol{y}^{(1)} = \boldsymbol{f}_{\mathrm{T}}(\boldsymbol{p}_i; \boldsymbol{c}_j) = \boldsymbol{p}_i - \boldsymbol{c}_j, \tag{6.44}$$

$$\boldsymbol{y}^{(2)} = \boldsymbol{f}_{\mathrm{R}}(\boldsymbol{y}^{(1)}; \boldsymbol{q}_j) = \boldsymbol{R}(\boldsymbol{q}_j)\,\boldsymbol{y}^{(1)}, \tag{6.45}$$

$$\boldsymbol{y}^{(3)} = \boldsymbol{f}_{\mathrm{P}}(\boldsymbol{y}^{(2)}) = \frac{\boldsymbol{y}^{(2)}}{z^{(2)}}, \tag{6.46}$$

$$\boldsymbol{x}_i = \boldsymbol{f}_{\mathrm{C}}(\boldsymbol{y}^{(3)}; \boldsymbol{k}) = \boldsymbol{K}(\boldsymbol{k})\,\boldsymbol{y}^{(3)}. \tag{6.47}$$

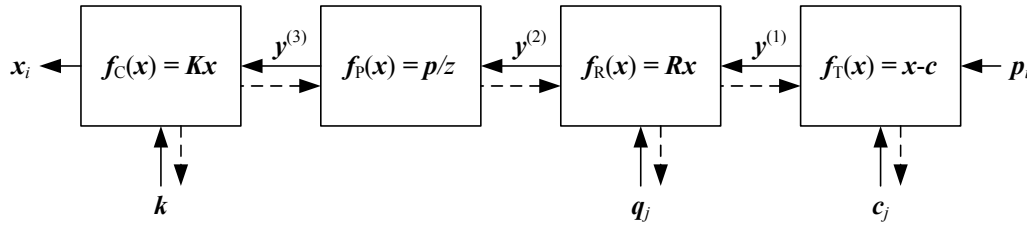


Figure 6.5 A set of chained transforms for projecting a 3D point $\boldsymbol{p}_i$ to a 2D measurement $\boldsymbol{x}_i$ through a series of transformations $\boldsymbol{f}^{(k)}$, each of which is controlled by its own set of parameters. The dashed lines indicate the flow of information as partial derivatives are computed during a backward pass.

Note that in these equations, we have indexed the camera centers $\boldsymbol{c}_j$ and camera rotation quaternions $\boldsymbol{q}_j$ by an index j, in case more than one pose of the calibration object is being used (see also Section 7.4.) We are also using the camera center $\boldsymbol{c}_j$ instead of the world translation $\boldsymbol{t}_j$, since this is a more natural parameter to estimate.

The advantage of this chained set of transformations is that each one has a simple partial derivative with respect both to its parameters and to its input. Thus, once the predicted value of $\tilde{\boldsymbol{x}}_i$ has been computed based on the 3D point location $\boldsymbol{p}_i$ and the current values of the pose parameters $(\boldsymbol{c}_j, \boldsymbol{q}_j, \boldsymbol{k})$, we can obtain all of the required partial derivatives using the chain rule

$$\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{p}^{(k)}} = \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{y}^{(k)}} \frac{\partial \boldsymbol{y}^{(k)}}{\partial \boldsymbol{p}^{(k)}}, \tag{6.48}$$

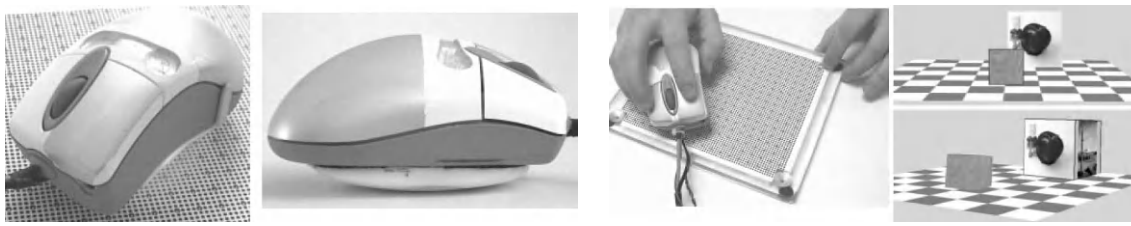
where $\boldsymbol{p}^{(k)}$ indicates one of the parameter vectors that is being optimized. (This same "trick" is used in neural networks as part of the *backpropagation* algorithm (Bishop 2006).)

The one special case in this formulation that can be considerably simplified is the computation of the rotation update. Instead of directly computing the derivatives of the 3×3 rotation matrix $\boldsymbol{R}(\boldsymbol{q})$ as a function of the unit quaternion entries, you can prepend the incremental rotation matrix $\Delta\boldsymbol{R}(\boldsymbol{\omega})$ given in Equation (2.35) to the current rotation matrix and compute the partial derivative of the transform with respect to these parameters, which results in a simple cross product of the backward chaining partial derivative and the outgoing 3D vector (2.36).

6.2.3 *Application*: Augmented reality

A widely used application of pose estimation is *augmented reality*, where virtual 3D images or annotations are superimposed on top of a live video feed, either through the use of see-through glasses (a head-mounted display) or on a regular computer or mobile device screen (Azuma, Baillot, Behringer *et al.* 2001; Haller, Billinghurst, and Thomas 2007). In some applications, a special pattern printed on cards or in a book is tracked to perform the augmentation (Kato, Billinghurst, Poupyrev *et al.* 2000; Billinghurst, Kato, and Poupyrev 2001). For a desktop application, a grid of dots printed on a mouse pad can be tracked by a camera embedded in an augmented mouse to give the user control of a full six degrees of freedom over their position and orientation in a 3D space (Hinckley, Sinclair, Hanson *et al.* 1999), as shown in Figure 6.6.

Sometimes, the scene itself provides a convenient object to track, such as the rectangle defining a desktop used in *through-the-lens camera control* (Gleicher and Witkin 1992). In



(a) (b) (c) (d)

Figure 6.6 The VideoMouse can sense six degrees of freedom relative to a specially printed mouse pad using its embedded camera (Hinckley, Sinclair, Hanson *et al.* 1999) © 1999 ACM: (a) top view of the mouse; (b) view of the mouse showing the curved base for rocking; (c) moving the mouse pad with the other hand extends the interaction capabilities; (d) the resulting movement seen on the screen.

outdoor locations, such as film sets, it is more common to place special markers such as brightly colored balls in the scene to make it easier to find and track them (Bogart 1991). In older applications, surveying techniques were used to determine the locations of these balls before filming. Today, it is more common to apply structure-from-motion directly to the film footage itself (Section 7.4.2).

Rapid pose estimation is also central to tracking the position and orientation of the hand-held remote controls used in Nintendo's Wii game systems. A high-speed camera embedded in the remote control is used to track the locations of the infrared (IR) LEDs in the bar that is mounted on the TV monitor. Pose estimation is then used to infer the remote control's location and orientation at very high frame rates. The Wii system can be extended to a variety of other user interaction applications by mounting the bar on a hand-held device, as described by Johnny Lee.[11]

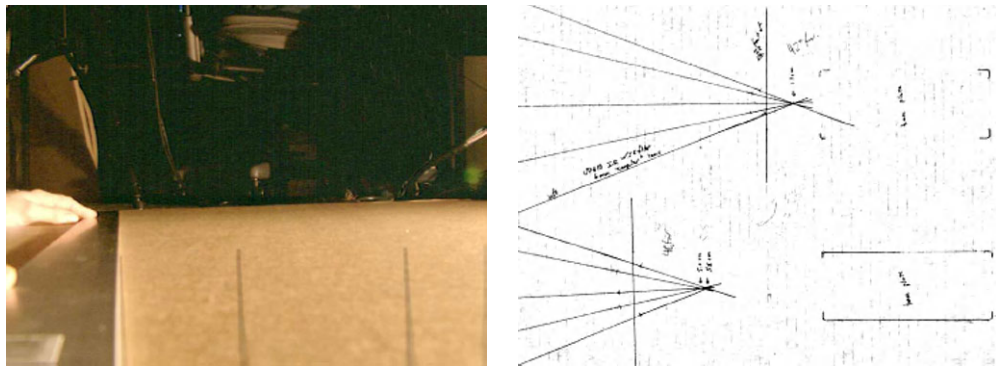
Exercises 6.4 and 6.5 have you implement two different tracking and pose estimation systems for augmented-reality applications. The first system tracks the outline of a rectangular object, such as a book cover or magazine page, and the second has you track the pose of a hand-held Rubik's cube.

6.3 Geometric intrinsic calibration

As described above in Equations (6.42–6.43), the computation of the internal (intrinsic) camera calibration parameters can occur simultaneously with the estimation of the (extrinsic) pose of the camera with respect to a known calibration target. This, indeed, is the "classic" approach to camera calibration used in both the photogrammetry (Slama 1980) and the computer vision (Tsai 1987) communities. In this section, we look at alternative formulations (which may not involve the full solution of a non-linear regression problem), the use of alternative calibration targets, and the estimation of the non-linear part of camera optics such as radial distortion.[12]

[11] http://johnnylee.net/projects/wii/.

[12] In some applications, you can use the EXIF tags associated with a JPEG image to obtain a rough estimate of a camera's focal length but this technique should be used with caution as the results are often inaccurate.



(a) (b)

Figure 6.7 Calibrating a lens by drawing straight lines on cardboard (Debevec, Wenger, Tchou *et al.* 2002) © 2002 ACM: (a) an image taken by the video camera showing a hand holding a metal ruler whose right edge appears vertical in the image; (b) the set of lines drawn on the cardboard converging on the front nodal point (center of projection) of the lens and indicating the horizontal field of view.

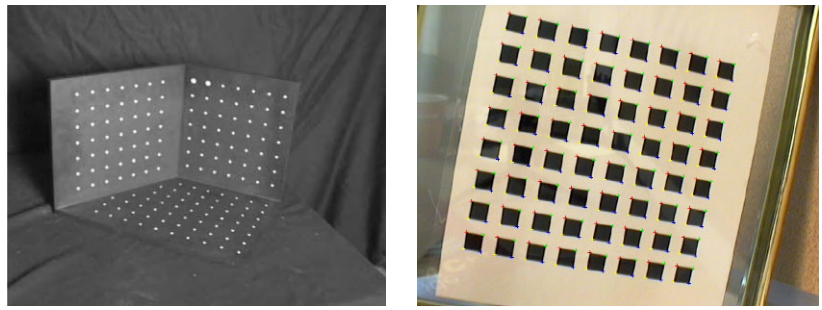
6.3.1 Calibration patterns

The use of a calibration pattern or set of markers is one of the more reliable ways to estimate a camera's intrinsic parameters. In photogrammetry, it is common to set up a camera in a large field looking at distant calibration targets whose exact location has been precomputed using surveying equipment (Slama 1980; Atkinson 1996; Kraus 1997). In this case, the translational component of the pose becomes irrelevant and only the camera rotation and intrinsic parameters need to be recovered.

If a smaller calibration rig needs to be used, e.g., for indoor robotics applications or for mobile robots that carry their own calibration target, it is best if the calibration object can span as much of the workspace as possible (Figure 6.8a), as planar targets often fail to accurately predict the components of the pose that lie far away from the plane. A good way to determine if the calibration has been successfully performed is to estimate the covariance in the parameters (Section 6.1.4) and then project 3D points from various points in the workspace into the image in order to estimate their 2D positional uncertainty.

An alternative method for estimating the focal length and center of projection of a lens is to place the camera on a large flat piece of cardboard and use a long metal ruler to draw lines on the cardboard that appear vertical in the image, as shown in Figure 6.7a (Debevec, Wenger, Tchou *et al.* 2002). Such lines lie on planes that are parallel to the vertical axis of the camera sensor and also pass through the lens' front nodal point. The location of the nodal point (projected vertically onto the cardboard plane) and the horizontal field of view (determined from lines that graze the left and right edges of the visible image) can be recovered by intersecting these lines and measuring their angular extent (Figure 6.7b).

If no calibration pattern is available, it is also possible to perform calibration simultaneously with structure and pose recovery (Sections 6.3.4 and 7.4), which is known as *self-calibration* (Faugeras, Luong, and Maybank 1992; Hartley and Zisserman 2004; Moons, Van Gool, and Vergauwen 2010). However, such an approach requires a large amount of imagery to be accurate.



(a) (b)

Figure 6.8 Calibration patterns: (a) a three-dimensional target (Quan and Lan 1999) © 1999 IEEE; (b) a two-dimensional target (Zhang 2000) © 2000 IEEE. Note that radial distortion needs to be removed from such images before the feature points can be used for calibration.

Planar calibration patterns

When a finite workspace is being used and accurate machining and motion control platforms are available, a good way to perform calibration is to move a planar calibration target in a controlled fashion through the workspace volume. This approach is sometimes called the *N-planes* calibration approach (Gremban, Thorpe, and Kanade 1988; Champleboux, Lavallée, Szeliski *et al.* 1992; Grossberg and Nayar 2001) and has the advantage that each camera pixel can be mapped to a unique 3D ray in space, which takes care of both linear effects modeled by the calibration matrix $\boldsymbol{K}$ and non-linear effects such as radial distortion (Section 6.3.5).

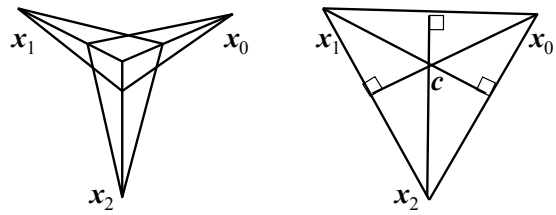
A less cumbersome but also less accurate calibration can be obtained by waving a planar calibration pattern in front of a camera (Figure 6.8b). In this case, the pattern's pose has (in principle) to be recovered in conjunction with the intrinsics. In this technique, each input image is used to compute a separate homography (6.19–6.23) $\tilde{\boldsymbol{H}}$ mapping the plane's calibration points $(X_i, Y_i, 0)$ into image coordinates (x_i, y_i),

$$\boldsymbol{x}_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \sim \boldsymbol{K} \begin{bmatrix} \boldsymbol{r}_0 & \boldsymbol{r}_1 & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \sim \tilde{\boldsymbol{H}} \boldsymbol{p}_i, \tag{6.49}$$

where the $\boldsymbol{r}_i$ are the first two columns of $\boldsymbol{R}$ and $\sim$ indicates equality up to scale. From these, Zhang (2000) shows how to form linear constraints on the nine entries in the $\boldsymbol{B} = \boldsymbol{K}^{-T}\boldsymbol{K}^{-1}$ matrix, from which the calibration matrix $\boldsymbol{K}$ can be recovered using a matrix square root and inversion. (The matrix $\boldsymbol{B}$ is known as the *image of the absolute conic* (IAC) in projective geometry and is commonly used for camera calibration (Hartley and Zisserman 2004, Section 7.5).) If only the focal length is being recovered, the even simpler approach of using vanishing points can be used instead.

6.3.2 Vanishing points

A common case for calibration that occurs often in practice is when the camera is looking at a man-made scene with strong extended rectahedral objects such as boxes or room walls. In this case, we can intersect the 2D lines corresponding to 3D parallel lines to compute their



(a) (b)

Figure 6.9 Calibration from vanishing points: (a) any pair of finite vanishing points $(\hat{x}_i, \hat{x}_j)$ can be used to estimate the focal length; (b) the orthocenter of the vanishing point triangle gives the optical center of the image $\boldsymbol{c}$.

vanishing points, as described in Section 4.3.3, and use these to determine the intrinsic and extrinsic calibration parameters (Caprile and Torre 1990; Becker and Bove 1995; Liebowitz and Zisserman 1998; Cipolla, Drummond, and Robertson 1999; Antone and Teller 2002; Criminisi, Reid, and Zisserman 2000; Hartley and Zisserman 2004; Pflugfelder 2008).

Let us assume that we have detected two or more orthogonal vanishing points, all of which are *finite*, i.e., they are not obtained from lines that appear to be parallel in the image plane (Figure 6.9a). Let us also assume a simplified form for the calibration matrix $\boldsymbol{K}$ where only the focal length is unknown (2.59). (It is often safe for rough 3D modeling to assume that the optical center is at the center of the image, that the aspect ratio is 1, and that there is no skew.) In this case, the projection equation for the vanishing points can be written as

$$\hat{\boldsymbol{x}}_i = \begin{bmatrix} x_i - c_x \\ y_i - c_y \\ f \end{bmatrix} \sim \boldsymbol{R}\boldsymbol{p}_i = \boldsymbol{r}_i, \tag{6.50}$$

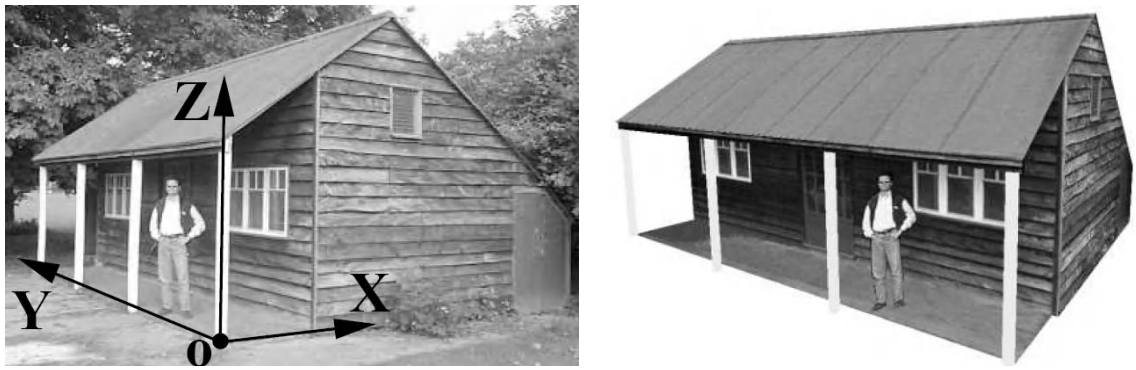
where $\boldsymbol{p}_i$ corresponds to one of the cardinal directions $(1, 0, 0)$, $(0, 1, 0)$, or $(0, 0, 1)$, and $\boldsymbol{r}_i$ is the ith column of the rotation matrix $\boldsymbol{R}$.

From the orthogonality between columns of the rotation matrix, we have

$$\boldsymbol{r}_i \cdot \boldsymbol{r}_j \sim (x_i - c_x)(x_j - c_y) + (y_i - c_y)(y_j - c_y) + f^2 = 0 \tag{6.51}$$

from which we can obtain an estimate for f^2. Note that the accuracy of this estimate increases as the vanishing points move closer to the center of the image. In other words, it is best to tilt the calibration pattern a decent amount around the 45° axis, as in Figure 6.9a. Once the focal length f has been determined, the individual columns of $\boldsymbol{R}$ can be estimated by normalizing the left hand side of (6.50) and taking cross products. Alternatively, an SVD of the initial $\boldsymbol{R}$ estimate, which is a variant on orthogonal Procrustes (6.32), can be used.

If all three vanishing points are visible and finite in the same image, it is also possible to estimate the optical center as the orthocenter of the triangle formed by the three vanishing points (Caprile and Torre 1990; Hartley and Zisserman 2004, Section 7.6) (Figure 6.9b). In practice, however, it is more accurate to re-estimate any unknown intrinsic calibration parameters using non-linear least squares (6.42).



(a) (b)

Figure 6.10 Single view metrology (Criminisi, Reid, and Zisserman 2000) © 2000 Springer: (a) input image showing the three coordinate axes computed from the two horizontal vanishing points (which can be determined from the sidings on the shed); (b) a new view of the 3D reconstruction.

6.3.3 *Application*: Single view metrology

A fun application of vanishing point estimation and camera calibration is the *single view metrology* system developed by Criminisi, Reid, and Zisserman (2000). Their system allows people to interactively measure heights and other dimensions as well as to build piecewise-planar 3D models, as shown in Figure 6.10.

The first step in their system is to identify two orthogonal vanishing points on the ground plane and the vanishing point for the vertical direction, which can be done by drawing some parallel sets of lines in the image. (Alternatively, automated techniques such as those discussed in Section 4.3.3 or by Schaffalitzky and Zisserman (2000) could be used.) The user then marks a few dimensions in the image, such as the height of a reference object, and the system can automatically compute the height of another object. Walls and other planar impostors (geometry) can also be sketched and reconstructed.

In the formulation originally developed by Criminisi, Reid, and Zisserman (2000), the system produces an *affine* reconstruction, i.e., one that is only known up to a set of independent scaling factors along each axis. A potentially more useful system can be constructed by assuming that the camera is calibrated up to an unknown focal length, which can be recovered from orthogonal (finite) vanishing directions, as we just described in Section 6.3.2. Once this is done, the user can indicate an origin on the ground plane and another point a known distance away. From this, points on the ground plane can be directly projected into 3D and points above the ground plane, when paired with their ground plane projections, can also be recovered. A fully metric reconstruction of the scene then becomes possible.

Exercise 6.9 has you implement such a system and then use it to model some simple 3D scenes. Section 12.6.1 describes other, potentially multi-view, approaches to architectural reconstruction, including an interactive piecewise-planar modeling system that uses vanishing points to establish 3D line directions and plane normals (Sinha, Steedly, Szeliski *et al.* 2008).

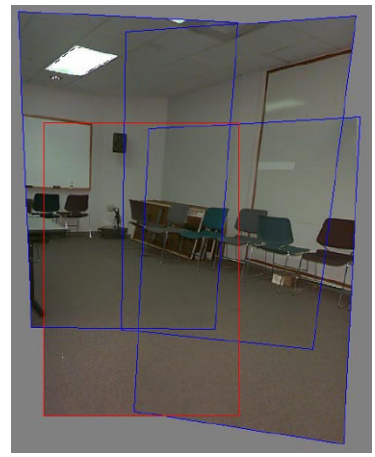


Figure 6.11 Four images taken with a hand-held camera registered using a 3D rotation motion model, which can be used to estimate the focal length of the camera (Szeliski and Shum 1997) © 2000 ACM.

6.3.4 Rotational motion

When no calibration targets or known structures are available but you can rotate the camera around its front nodal point (or, equivalently, work in a large open environment where all objects are distant), the camera can be calibrated from a set of overlapping images by assuming that it is undergoing pure rotational motion, as shown in Figure 6.11 (Stein 1995; Hartley 1997b; Hartley, Hayman, de Agapito *et al.* 2000; de Agapito, Hayman, and Reid 2001; Kang and Weiss 1999; Shum and Szeliski 2000; Frahm and Koch 2003). When a full 360° motion is used to perform this calibration, a very accurate estimate of the focal length f can be obtained, as the accuracy in this estimate is proportional to the total number of pixels in the resulting cylindrical panorama (Section 9.1.6) (Stein 1995; Shum and Szeliski 2000).

To use this technique, we first compute the homographies $\tilde{\boldsymbol{H}}_{ij}$ between all overlapping pairs of images, as explained in Equations (6.19–6.23). Then, we use the observation, first made in Equation (2.72) and explored in more detail in Section 9.1.3 (9.5), that each homography is related to the inter-camera rotation $\boldsymbol{R}_{ij}$ through the (unknown) calibration matrices $\boldsymbol{K}_i$ and $\boldsymbol{K}_j$,

$$\tilde{\boldsymbol{H}}_{ij} = \boldsymbol{K}_i \boldsymbol{R}_i \boldsymbol{R}_j^{-1} \boldsymbol{K}_j^{-1} = \boldsymbol{K}_i \boldsymbol{R}_{ij} \boldsymbol{K}_j^{-1}. \tag{6.52}$$

The simplest way to obtain the calibration is to use the simplified form of the calibration matrix (2.59), where we assume that the pixels are square and the optical center lies at the center of the image, i.e., $\boldsymbol{K}_k = \mathrm{diag}(f_k, f_k, 1)$. (We number the pixel coordinates accordingly, i.e., place pixel $(x, y) = (0, 0)$ at the center of the image.) We can then rewrite Equation (6.52) as

$$\boldsymbol{R}_{10} \sim \boldsymbol{K}_1^{-1} \tilde{\boldsymbol{H}}_{10} \boldsymbol{K}_0 \sim \begin{bmatrix} h_{00} & h_{01} & f_0^{-1} h_{02} \\ h_{10} & h_{11} & f_0^{-1} h_{12} \\ f_1 h_{20} & f_1 h_{21} & f_0^{-1} f_1 h_{22} \end{bmatrix}, \tag{6.53}$$

where h_{ij} are the elements of $\tilde{\boldsymbol{H}}_{10}$.

Using the orthonormality properties of the rotation matrix $\boldsymbol{R}_{10}$ and the fact that the right hand side of (6.53) is known only up to a scale, we obtain

$$h_{00}^2 + h_{01}^2 + f_0^{-2} h_{02}^2 = h_{10}^2 + h_{11}^2 + f_0^{-2} h_{12}^2 \tag{6.54}$$

and

$$h_{00}h_{10} + h_{01}h_{11} + f_0^{-2} h_{02}h_{12} = 0. \tag{6.55}$$

From this, we can compute estimates for f_0 of

$$f_0^2 = \frac{h_{12}^2 - h_{02}^2}{h_{00}^2 + h_{01}^2 - h_{10}^2 - h_{11}^2} \quad \text{if} \quad \mathrm{h}_{00}^2 + \mathrm{h}_{01}^2 \neq \mathrm{h}_{10}^2 + \mathrm{h}_{11}^2 \tag{6.56}$$

or

$$f_0^2 = -\frac{h_{02}h_{12}}{h_{00}h_{10} + h_{01}h_{11}} \quad \text{if} \quad \mathrm{h}_{00}\mathrm{h}_{10} \neq -\mathrm{h}_{01}\mathrm{h}_{11}. \tag{6.57}$$

(Note that the equations originally given by Szeliski and Shum (1997) are erroneous; the correct equations are given by Shum and Szeliski (2000).) If neither of these conditions holds, we can also take the dot products between the first (or second) row and the third one. Similar results can be obtained for f_1 as well, by analyzing the columns of $\tilde{\boldsymbol{H}}_{10}$. If the focal length is the same for both images, we can take the geometric mean of f_0 and f_1 as the estimated focal length $f = \sqrt{f_1 f_0}$. When multiple estimates of f are available, e.g., from different homographies, the median value can be used as the final estimate.

A more general (upper-triangular) estimate of $\boldsymbol{K}$ can be obtained in the case of a fixed-parameter camera $\boldsymbol{K}_i = \boldsymbol{K}$ using the technique of Hartley (1997b). Observe from (6.52) that $\boldsymbol{R}_{ij} \sim \boldsymbol{K}^{-1}\tilde{\boldsymbol{H}}_{ij}\boldsymbol{K}$ and $\boldsymbol{R}_{ij}^{-T} \sim \boldsymbol{K}^T \tilde{\boldsymbol{H}}_{ij}^{-T} \boldsymbol{K}^{-T}$. Equating $\boldsymbol{R}_{ij} = \boldsymbol{R}_{ij}^{-T}$ we obtain $\boldsymbol{K}^{-1}\tilde{\boldsymbol{H}}_{ij}\boldsymbol{K} \sim \boldsymbol{K}^T \tilde{\boldsymbol{H}}_{ij}^{-T} \boldsymbol{K}^{-T}$, from which we get

$$\tilde{\boldsymbol{H}}_{ij}(\boldsymbol{K}\boldsymbol{K}^T) \sim (\boldsymbol{K}\boldsymbol{K}^T)\tilde{\boldsymbol{H}}_{ij}^{-T}. \tag{6.58}$$

This provides us with some homogeneous linear constraints on the entries in $\boldsymbol{A} = \boldsymbol{K}\boldsymbol{K}^T$, which is known as the *dual of the image of the absolute conic* (Hartley 1997b; Hartley and Zisserman 2004). (Recall that when we estimate a homography, we can only recover it up to an unknown scale.) Given a sufficient number of independent homography estimates $\tilde{\boldsymbol{H}}_{ij}$, we can recover $\boldsymbol{A}$ (up to a scale) using either SVD or eigenvalue analysis and then recover $\boldsymbol{K}$ through Cholesky decomposition (Appendix A.1.4). Extensions to the cases of temporally varying calibration parameters and non-stationary cameras are discussed by Hartley, Hayman, de Agapito *et al.* (2000) and de Agapito, Hayman, and Reid (2001).

The quality of the intrinsic camera parameters can be greatly increased by constructing a full 360° panorama, since mis-estimating the focal length will result in a gap (or excessive overlap) when the first image in the sequence is stitched to itself (Figure 9.5). The resulting mis-alignment can be used to improve the estimate of the focal length and to re-adjust the rotation estimates, as described in Section 9.1.4. Rotating the camera by 90° around its optic axis and re-shooting the panorama is a good way to check for aspect ratio and skew pixel problems, as is generating a full hemi-spherical panorama when there is sufficient texture.

Ultimately, however, the most accurate estimate of the calibration parameters (including radial distortion) can be obtained using a full simultaneous non-linear minimization of the intrinsic and extrinsic (rotation) parameters, as described in Section 9.2.

6.3.5 Radial distortion

When images are taken with wide-angle lenses, it is often necessary to model *lens distortions* such as *radial distortion*. As discussed in Section 2.1.6, the radial distortion model says that coordinates in the observed images are displaced away from (*barrel* distortion) or towards (*pincushion* distortion) the image center by an amount proportional to their radial distance (Figure 2.13a–b). The simplest radial distortion models use low-order polynomials (c.f. Equation (2.78)),

$$\begin{aligned} \hat{x} &= x(1 + \kappa_1 r^2 + \kappa_2 r^4) \\ \hat{y} &= y(1 + \kappa_1 r^2 + \kappa_2 r^4), \end{aligned} \tag{6.59}$$

where $r^2 = x^2 + y^2$ and κ_1 and κ_2 are called the *radial distortion parameters* (Brown 1971; Slama 1980).[13]

A variety of techniques can be used to estimate the radial distortion parameters for a given lens.[14] One of the simplest and most useful is to take an image of a scene with a lot of straight lines, especially lines aligned with and near the edges of the image. The radial distortion parameters can then be adjusted until all of the lines in the image are straight, which is commonly called the *plumb-line method* (Brown 1971; Kang 2001; El-Melegy and Farag 2003). Exercise 6.10 gives some more details on how to implement such a technique.

Another approach is to use several overlapping images and to combine the estimation of the radial distortion parameters with the image alignment process, i.e., by extending the pipeline used for stitching in Section 9.2.1. Sawhney and Kumar (1999) use a hierarchy of motion models (translation, affine, projective) in a coarse-to-fine strategy coupled with a quadratic radial distortion correction term. They use direct (intensity-based) minimization to compute the alignment. Stein (1997) uses a feature-based approach combined with a general 3D motion model (and quadratic radial distortion), which requires more matches than a parallax-free rotational panorama but is potentially more general. More recent approaches sometimes simultaneously compute both the unknown intrinsic parameters and the radial distortion coefficients, which may include higher-order terms or more complex rational or non-parametric forms (Claus and Fitzgibbon 2005; Sturm 2005; Thirthala and Pollefeys 2005; Barreto and Daniilidis 2005; Hartley and Kang 2005; Steele and Jaynes 2006; Tardif, Sturm, Trudeau *et al.* 2009).

When a known calibration target is being used (Figure 6.8), the radial distortion estimation can be folded into the estimation of the other intrinsic and extrinsic parameters (Zhang 2000; Hartley and Kang 2007; Tardif, Sturm, Trudeau *et al.* 2009). This can be viewed as adding another stage to the general non-linear minimization pipeline shown in Figure 6.5 between the intrinsic parameter multiplication box $\boldsymbol{f}_{\mathrm{C}}$ and the perspective division box $\boldsymbol{f}_{\mathrm{P}}$. (See Exercise 6.11 on more details for the case of a planar calibration target.)

Of course, as discussed in Section 2.1.6, more general models of lens distortion, such as fisheye and non-central projection, may sometimes be required. While the parameterization of such lenses may be more complicated (Section 2.1.6), the general approach of either using calibration rigs with known 3D positions or self-calibration through the use of multiple

[13] Sometimes the relationship between x and $\hat{x}$ is expressed the other way around, i.e., using primed (final) coordinates on the right-hand side, $x = \hat{x}(1 + \kappa_1 \hat{r}^2 + \kappa_2 \hat{r}^4)$. This is convenient if we map image pixels into (warped) rays and then undistort the rays to obtain 3D rays in space, i.e., if we are using inverse warping.

[14] Some of today's digital cameras are starting to remove radial distortion using software in the camera itself.

overlapping images of a scene can both be used (Hartley and Kang 2007; Tardif, Sturm, and Roy 2007). The same techniques used to calibrate for radial distortion can also be used to reduce the amount of chromatic aberration by separately calibrating each color channel and then warping the channels to put them back into alignment (Exercise 6.12).

6.4 Additional reading

Hartley and Zisserman (2004) provide a wonderful introduction to the topics of feature-based alignment and optimal motion estimation, as well as an in-depth discussion of camera calibration and pose estimation techniques.

Techniques for robust estimation are discussed in more detail in Appendix B.3 and in monographs and review articles on this topic (Huber 1981; Hampel, Ronchetti, Rousseeuw *et al.* 1986; Rousseeuw and Leroy 1987; Black and Rangarajan 1996; Stewart 1999). The most commonly used robust initialization technique in computer vision is RANdom SAmple Consensus (RANSAC) (Fischler and Bolles 1981), which has spawned a series of more efficient variants (Nistér 2003; Chum and Matas 2005).

The topic of registering 3D point data sets is called *absolute orientation* (Horn 1987) and *3D pose estimation* (Lorusso, Eggert, and Fisher 1995). A variety of techniques has been developed for simultaneously computing 3D point correspondences and their corresponding rigid transformations (Besl and McKay 1992; Zhang 1994; Szeliski and Lavallée 1996; Gold, Rangarajan, Lu *et al.* 1998; David, DeMenthon, Duraiswami *et al.* 2004; Li and Hartley 2007; Enqvist, Josephson, and Kahl 2009).

Camera calibration was first studied in photogrammetry (Brown 1971; Slama 1980; Atkinson 1996; Kraus 1997) but it has also been widely studied in computer vision (Tsai 1987; Gremban, Thorpe, and Kanade 1988; Champleboux, Lavallée, Szeliski *et al.* 1992; Zhang 2000; Grossberg and Nayar 2001). Vanishing points observed either from rectahedral calibration objects or man-made architecture are often used to perform rudimentary calibration (Caprile and Torre 1990; Becker and Bove 1995; Liebowitz and Zisserman 1998; Cipolla, Drummond, and Robertson 1999; Antone and Teller 2002; Criminisi, Reid, and Zisserman 2000; Hartley and Zisserman 2004; Pflugfelder 2008). Performing camera calibration without using known targets is known as *self-calibration* and is discussed in textbooks and surveys on structure from motion (Faugeras, Luong, and Maybank 1992; Hartley and Zisserman 2004; Moons, Van Gool, and Vergauwen 2010). One popular subset of such techniques uses pure rotational motion (Stein 1995; Hartley 1997b; Hartley, Hayman, de Agapito *et al.* 2000; de Agapito, Hayman, and Reid 2001; Kang and Weiss 1999; Shum and Szeliski 2000; Frahm and Koch 2003).

6.5 Exercises

Ex 6.1: Feature-based image alignment for flip-book animations Take a set of photos of an action scene or portrait (preferably in motor-drive—continuous shooting—mode) and align them to make a composite or flip-book animation.

1. Extract features and feature descriptors using some of the techniques described in Sections 4.1.1–4.1.2.

2. Match your features using nearest neighbor matching with a nearest neighbor distance ratio test (4.18).
3. Compute an optimal 2D translation and rotation between the first image and all subsequent images, using least squares (Section 6.1.1) with optional RANSAC for robustness (Section 6.1.4).
4. Resample all of the images onto the first image's coordinate frame (Section 3.6.1) using either bilinear or bicubic resampling and optionally crop them to their common area.
5. Convert the resulting images into an animated GIF (using software available from the Web) or optionally implement cross-dissolves to turn them into a "slo-mo" video.
6. (Optional) Combine this technique with feature-based (Exercise 3.25) morphing.

Ex 6.2: Panography Create the kind of panograph discussed in Section 6.1.2 and commonly found on the Web.

1. Take a series of interesting overlapping photos.
2. Use the feature detector, descriptor, and matcher developed in Exercises 4.1–4.4 (or existing software) to match features among the images.
3. Turn each connected component of matching features into a *track*, i.e., assign a unique index i to each track, discarding any tracks that are inconsistent (contain two different features in the same image).
4. Compute a global translation for each image using Equation (6.12).
5. Since your matches probably contain errors, turn the above least square metric into a robust metric (6.25) and re-solve your system using iteratively reweighted least squares.
6. Compute the size of the resulting composite canvas and resample each image into its final position on the canvas. (Keeping track of bounding boxes will make this more efficient.)
7. Average all of the images, or choose some kind of ordering and implement translucent *over* compositing (3.8).
8. (Optional) Extend your parametric motion model to include rotations and scale, i.e., the similarity transform given in Table 6.1. Discuss how you could handle the case of translations and rotations only (no scale).
9. (Optional) Write a simple tool to let the user adjust the ordering and opacity, and add or remove images.
10. (Optional) Write down a different least squares problem that involves pairwise matching of images. Discuss why this might be better or worse than the global matching formula given in (6.12).

Ex 6.3: 2D rigid/Euclidean matching Several alternative approaches are given in Section 6.1.3 for estimating a 2D rigid (Euclidean) alignment.

1. Implement the various alternatives and compare their accuracy on synthetic data, i.e., random 2D point clouds with noisy feature positions.
2. One approach is to estimate the translations from the centroids and then estimate rotation in polar coordinates. Do you need to weight the angles obtained from a polar decomposition in some way to get the statistically correct estimate?
3. How can you modify your techniques to take into account either scalar (6.10) or full two-dimensional point covariance weightings (6.11)? Do all of the previously developed "shortcuts" still work or does full weighting require iterative optimization?

Ex 6.4: 2D match move/augmented reality Replace a picture in a magazine or a book with a different image or video.

1. With a webcam, take a picture of a magazine or book page.
2. Outline a figure or picture on the page with a rectangle, i.e., draw over the four sides as they appear in the image.
3. Match features in this area with each new image frame.
4. Replace the original image with an "advertising" insert, warping the new image with the appropriate homography.
5. Try your approach on a clip from a sporting event (e.g., indoor or outdoor soccer) to implement a billboard replacement.

Ex 6.5: 3D joystick Track a Rubik's cube to implement a 3D joystick/mouse control.

1. Get out an old Rubik's cube (or get one from your parents).
2. Write a program to detect the center of each colored square.
3. Group these centers into lines and then find the vanishing points for each face.
4. Estimate the rotation angle and focal length from the vanishing points.
5. Estimate the full 3D pose (including translation) by finding one or more 3×3 grids and recovering the plane's full equation from this known homography using the technique developed by Zhang (2000).
6. Alternatively, since you already know the rotation, simply estimate the unknown translation from the known 3D corner points on the cube and their measured 2D locations using either linear or non-linear least squares.
7. Use the 3D rotation and position to control a VRML or 3D game viewer.

Ex 6.6: Rotation-based calibration Take an outdoor or indoor sequence from a rotating camera with very little parallax and use it to calibrate the focal length of your camera using the techniques described in Section 6.3.4 or Sections 9.1.3–9.2.1.

1. Take out any radial distortion in the images using one of the techniques from Exercises 6.10–6.11 or using parameters supplied for a given camera by your instructor.

2. Detect and match feature points across neighboring frames and chain them into feature tracks.
3. Compute homographies between overlapping frames and use Equations (6.56–6.57) to get an estimate of the focal length.
4. Compute a full 360° panorama and update your focal length estimate to close the gap (Section 9.1.4).
5. (Optional) Perform a complete bundle adjustment in the rotation matrices and focal length to obtain the highest quality estimate (Section 9.2.1).

Ex 6.7: Target-based calibration Use a three-dimensional target to calibrate your camera.

1. Construct a three-dimensional calibration pattern with known 3D locations. It is not easy to get high accuracy unless you use a machine shop, but you can get close using heavy plywood and printed patterns.
2. Find the corners, e.g, using a line finder and intersecting the lines.
3. Implement one of the iterative calibration and pose estimation algorithms described in Tsai (1987); Bogart (1991); Gleicher and Witkin (1992) or the system described in Section 6.2.2.
4. Take many pictures at different distances and orientations relative to the calibration target and report on both your re-projection errors and accuracy. (To do the latter, you may need to use simulated data.)

Ex 6.8: Calibration accuracy Compare the three calibration techniques (plane-based, rotation-based, and 3D-target-based).

One approach is to have a different student implement each one and to compare the results. Another approach is to use synthetic data, potentially re-using the software you developed for Exercise 2.3. The advantage of using synthetic data is that you know the ground truth for the calibration and pose parameters, you can easily run lots of experiments, and you can synthetically vary the noise in your measurements.

Here are some possible guidelines for constructing your test sets:

1. Assume a medium-wide focal length (say, 50° field of view).
2. For the plane-based technique, generate a 2D grid target and project it at different inclinations.
3. For a 3D target, create an inner cube corner and position it so that it fills most of field of view.
4. For the rotation technique, scatter points uniformly on a sphere until you get a similar number of points as for other techniques.

Before comparing your techniques, predict which one will be the most accurate (normalize your results by the square root of the number of points used).

Add varying amounts of noise to your measurements and describe the noise sensitivity of your various techniques.

Ex 6.9: Single view metrology Implement a system to measure dimensions and reconstruct a 3D model from a single image of a man-made scene using visible vanishing directions (Section 6.3.3) (Criminisi, Reid, and Zisserman 2000).

1. Find the three orthogonal vanishing points from parallel lines and use them to establish the three coordinate axes (rotation matrix $\boldsymbol{R}$ of the camera relative to the scene). If two of the vanishing points are finite (not at infinity), use them to compute the focal length, assuming a known optical center. Otherwise, find some other way to calibrate your camera; you could use some of the techniques described by Schaffalitzky and Zisserman (2000).

2. Click on a ground plane point to establish your origin and click on a point a known distance away to establish the scene scale. This lets you compute the translation $\boldsymbol{t}$ between the camera and the scene. As an alternative, click on a pair of points, one on the ground plane and one above it, and use the known height to establish the scene scale.

3. Write a user interface that lets you click on ground plane points to recover their 3D locations. (Hint: you already know the camera matrix, so knowledge of a point's z value is sufficient to recover its 3D location.) Click on pairs of points (one on the ground plane, one above it) to measure vertical heights.

4. Extend your system to let you draw quadrilaterals in the scene that correspond to axis-aligned rectangles in the world, using some of the techniques described by Sinha, Steedly, Szeliski *et al.* (2008). Export your 3D rectangles to a VRML or PLY[15] file.

5. (Optional) Warp the pixels enclosed by the quadrilateral using the correct homography to produce a texture map for each planar polygon.

Ex 6.10: Radial distortion with plumb lines Implement a plumb-line algorithm to determine the radial distortion parameters.

1. Take some images of scenes with lots of straight lines, e.g., hallways in your home or office, and try to get some of the lines as close to the edges of the image as possible.

2. Extract the edges and link them into curves, as described in Section 4.2.2 and Exercise 4.8.

3. Fit quadratic or elliptic curves to the linked edges using a generalization of the successive line approximation algorithm described in Section 4.3.1 and Exercise 4.11 and keep the curves that fit this form well.

4. For each curved segment, fit a straight line and minimize the perpendicular distance between the curve and the line while adjusting the radial distortion parameters.

5. Alternate between re-fitting the straight line and adjusting the radial distortion parameters until convergence.

[15] http://meshlab.sf.net.

Ex 6.11: Radial distortion with a calibration target Use a grid calibration target to determine the radial distortion parameters.

1. Print out a planar calibration target, mount it on a stiff board, and get it to fill your field of view.
2. Detect the squares, lines, or dots in your calibration target.
3. Estimate the homography mapping the target to the camera from the central portion of the image that does not have any radial distortion.
4. Predict the positions of the remaining targets and use the differences between the observed and predicted positions to estimate the radial distortion.
5. (Optional) Fit a general spline model (for severe distortion) instead of the quartic distortion model.
6. (Optional) Extend your technique to calibrate a fisheye lens.

Ex 6.12: Chromatic aberration Use the radial distortion estimates for each color channel computed in the previous exercise to clean up wide-angle lens images by warping all of the channels into alignment. (Optional) Straighten out the images at the same time.

Can you think of any reasons why this warping strategy may not always work?

Chapter 7

Structure from motion

7.1 Triangulation . . . 305
7.2 Two-frame structure from motion . . . 307
7.2.1 Projective (uncalibrated) reconstruction . . . 312
7.2.2 Self-calibration . . . 313
7.2.3 *Application*: View morphing . . . 315
7.3 Factorization . . . 315
7.3.1 Perspective and projective factorization . . . 318
7.3.2 *Application*: Sparse 3D model extraction . . . 319
7.4 Bundle adjustment . . . 320
7.4.1 Exploiting sparsity . . . 322
7.4.2 *Application*: Match move and augmented reality . . . 324
7.4.3 Uncertainty and ambiguities . . . 326
7.4.4 *Application*: Reconstruction from Internet photos . . . 327
7.5 Constrained structure and motion . . . 329
7.5.1 Line-based techniques . . . 330
7.5.2 Plane-based techniques . . . 331
7.6 Additional reading . . . 332
7.7 Exercises . . . 332

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_7, © Springer-Verlag London Limited 2011

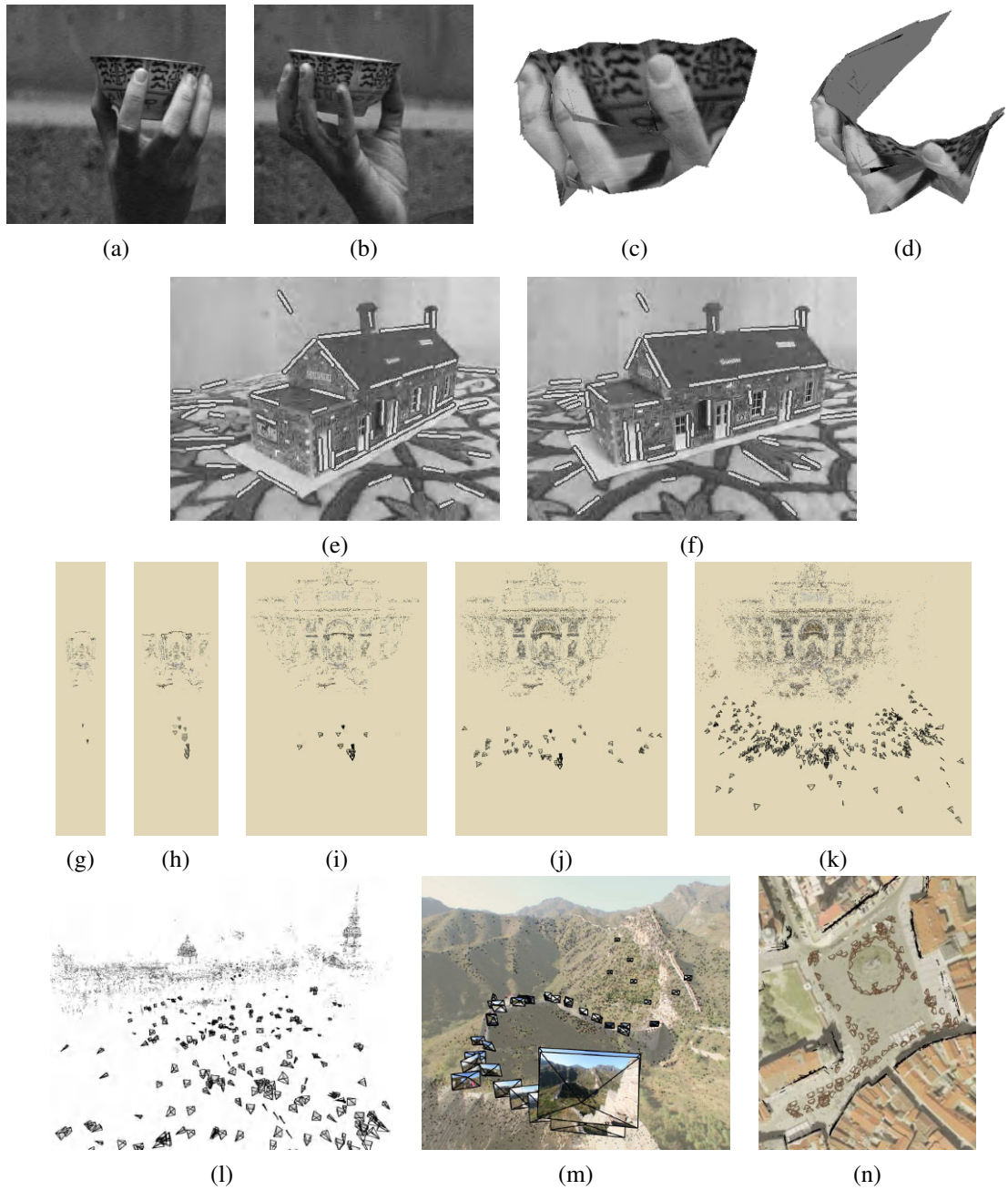


Figure 7.1 Structure from motion systems: (a–d) orthographic factorization (Tomasi and Kanade 1992) © 1992 Springer; (e–f) line matching (Schmid and Zisserman 1997) © 1997 IEEE; (g–k) incremental structure from motion (Snavely, Seitz, and Szeliski 2006); (l) 3D reconstruction of Trafalgar Square (Snavely, Seitz, and Szeliski 2006); (m) 3D reconstruction of the Great Wall of China (Snavely, Seitz, and Szeliski 2006); (n) 3D reconstruction of the Old Town Square, Prague (Snavely, Seitz, and Szeliski 2006) © 2006 ACM.

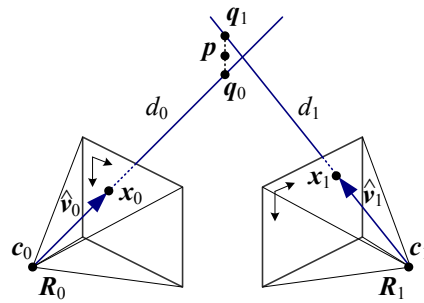


Figure 7.2 3D point triangulation by finding the point $\boldsymbol{p}$ that lies nearest to all of the optical rays $\boldsymbol{c}_j + d_j\hat{\boldsymbol{v}}_j$.

In the previous chapter, we saw how 2D and 3D point sets could be aligned and how such alignments could be used to estimate both a camera's pose and its internal calibration parameters. In this chapter, we look at the converse problem of estimating the locations of 3D points from multiple images given only a sparse set of correspondences between image features. While this process often involves simultaneously estimating both 3D geometry (structure) and camera pose (motion), it is commonly known as *structure from motion* (Ullman 1979).

The topics of projective geometry and structure from motion are extremely rich and some excellent textbooks and surveys have been written on them (Faugeras and Luong 2001; Hartley and Zisserman 2004; Moons, Van Gool, and Vergauwen 2010). This chapter skips over a lot of the richer material available in these books, such as the trifocal tensor and algebraic techniques for full self-calibration, and concentrates instead on the basics that we have found useful in large-scale, image-based reconstruction problems (Snavely, Seitz, and Szeliski 2006).

We begin with a brief discussion of *triangulation* (Section 7.1), which is the problem of estimating a point's 3D location when it is seen from multiple cameras. Next, we look at the two-frame structure from motion problem (Section 7.2), which involves the determination of the *epipolar geometry* between two cameras and which can also be used to recover certain information about the camera intrinsics using self-calibration (Section 7.2.2). Section 7.3 looks at *factorization* approaches to simultaneously estimating structure and motion from large numbers of point tracks using orthographic approximations to the projection model. We then develop a more general and useful approach to structure from motion, namely the simultaneous *bundle adjustment* of all the camera and 3D structure parameters (Section 7.4). We also look at special cases that arise when there are higher-level structures, such as lines and planes, in the scene (Section 7.5).

7.1 Triangulation

The problem of determining a point's 3D position from a set of corresponding image locations and known camera positions is known as *triangulation*. This problem is the converse of the pose estimation problem we studied in Section 6.2.

One of the simplest ways to solve this problem is to find the 3D point $\boldsymbol{p}$ that lies closest to all of the 3D rays corresponding to the 2D matching feature locations $\{\boldsymbol{x}_j\}$ observed by cam-

eras $\{\boldsymbol{P}_j = \boldsymbol{K}_j[\boldsymbol{R}_j|\boldsymbol{t}_j]\}$, where $\boldsymbol{t}_j = -\boldsymbol{R}_j\boldsymbol{c}_j$ and $\boldsymbol{c}_j$ is the jth camera center (2.55–2.56). As you can see in Figure 7.2, these rays originate at $\boldsymbol{c}_j$ in a direction $\hat{\boldsymbol{v}}_j = \mathcal{N}(\boldsymbol{R}_j^{-1}\boldsymbol{K}_j^{-1}\boldsymbol{x}_j)$. The nearest point to $\boldsymbol{p}$ on this ray, which we denote as $\boldsymbol{q}_j$, minimizes the distance

$$\|\boldsymbol{c}_j + d_j\hat{\boldsymbol{v}}_j - \boldsymbol{p}\|^2, \tag{7.1}$$

which has a minimum at $d_j = \hat{\boldsymbol{v}}_j \cdot (\boldsymbol{p} - \boldsymbol{c}_j)$. Hence,

$$\boldsymbol{q}_j = \boldsymbol{c}_j + (\hat{\boldsymbol{v}}_j\hat{\boldsymbol{v}}_j^T)(\boldsymbol{p} - \boldsymbol{c}_j) = \boldsymbol{c}_j + (\boldsymbol{p} - \boldsymbol{c}_j)_{\|}, \tag{7.2}$$

in the notation of Equation (2.29), and the squared distance between $\boldsymbol{p}$ and $\boldsymbol{q}_j$ is

$$r_j^2 = \|(\boldsymbol{I} - \hat{\boldsymbol{v}}_j\hat{\boldsymbol{v}}_j^T)(\boldsymbol{p} - \boldsymbol{c}_j)\|^2 = \|(\boldsymbol{p} - \boldsymbol{c}_j)_{\perp}\|^2. \tag{7.3}$$

The optimal value for $\boldsymbol{p}$, which lies closest to all of the rays, can be computed as a regular least squares problem by summing over all the r_j^2 and finding the optimal value of $\boldsymbol{p}$,

$$\boldsymbol{p} = \left[\sum_j (\boldsymbol{I} - \hat{\boldsymbol{v}}_j\hat{\boldsymbol{v}}_j^T)\right]^{-1}\left[\sum_j (\boldsymbol{I} - \hat{\boldsymbol{v}}_j\hat{\boldsymbol{v}}_j^T)\boldsymbol{c}_j\right]. \tag{7.4}$$

An alternative formulation, which is more statistically optimal and which can produce significantly better estimates if some of the cameras are closer to the 3D point than others, is to minimize the residual in the measurement equations

$$x_j = \frac{p_{00}^{(j)}X + p_{01}^{(j)}Y + p_{02}^{(j)}Z + p_{03}^{(j)}W}{p_{20}^{(j)}X + p_{21}^{(j)}Y + p_{22}^{(j)}Z + p_{23}^{(j)}W} \tag{7.5}$$

$$y_j = \frac{p_{10}^{(j)}X + p_{11}^{(j)}Y + p_{12}^{(j)}Z + p_{13}^{(j)}W}{p_{20}^{(j)}X + p_{21}^{(j)}Y + p_{22}^{(j)}Z + p_{23}^{(j)}W}, \tag{7.6}$$

where (x_j, y_j) are the measured 2D feature locations and $\{p_{00}^{(j)} \dots p_{23}^{(j)}\}$ are the known entries in camera matrix $\boldsymbol{P}_j$ (Sutherland 1974).

As with Equations (6.21, 6.33, and 6.34), this set of non-linear equations can be converted into a linear least squares problem by multiplying both sides of the denominator. Note that if we use homogeneous coordinates $\boldsymbol{p} = (X, Y, Z, W)$, the resulting set of equations is homogeneous and is best solved as a singular value decomposition (SVD) or eigenvalue problem (looking for the smallest singular vector or eigenvector). If we set $W = 1$, we can use regular linear least squares, but the resulting system may be singular or poorly conditioned, i.e., if all of the viewing rays are parallel, as occurs for points far away from the camera.

For this reason, it is generally preferable to parameterize 3D points using homogeneous coordinates, especially if we know that there are likely to be points at greatly varying distances from the cameras. Of course, minimizing the set of observations (7.5–7.6) using non-linear least squares, as described in (6.14 and 6.23), is preferable to using linear least squares, regardless of the representation chosen.

For the case of two observations, it turns out that the location of the point $\boldsymbol{p}$ that exactly minimizes the true reprojection error (7.5–7.6) can be computed using the solution of degree six equations (Hartley and Sturm 1997). Another problem to watch out for with triangulation is the issue of *chirality*, i.e., ensuring that the reconstructed points lie in front of all the

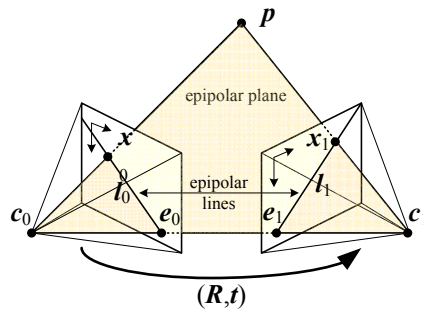


Figure 7.3 Epipolar geometry: The vectors $\boldsymbol{t} = \boldsymbol{c}_1 - \boldsymbol{c}_0$, $\boldsymbol{p} - \boldsymbol{c}_0$ and $\boldsymbol{p} - \boldsymbol{c}_1$ are co-planar and define the basic epipolar constraint expressed in terms of the pixel measurements $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$.

cameras (Hartley 1998). While this cannot always be guaranteed, a useful heuristic is to take the points that lie behind the cameras because their rays are diverging (imagine Figure 7.2 where the rays were pointing *away* from each other) and to place them on the plane at infinity by setting their W values to 0.

7.2 Two-frame structure from motion

So far in our study of 3D reconstruction, we have always assumed that either the 3D point positions or the 3D camera poses are known in advance. In this section, we take our first look at *structure from motion*, which is the simultaneous recovery of 3D structure and pose from image correspondences.

Consider Figure 7.3, which shows a 3D point $\boldsymbol{p}$ being viewed from two cameras whose relative position can be encoded by a rotation $\boldsymbol{R}$ and a translation $\boldsymbol{t}$. Since we do not know anything about the camera positions, without loss of generality, we can set the first camera at the origin $\boldsymbol{c}_0 = \boldsymbol{0}$ and at a canonical orientation $\boldsymbol{R}_0 = \boldsymbol{I}$.

Now notice that the observed location of point $\boldsymbol{p}$ in the first image, $\boldsymbol{p}_0 = d_0\hat{\boldsymbol{x}}_0$ is mapped into the second image by the transformation

$$d_1\hat{\boldsymbol{x}}_1 = \boldsymbol{p}_1 = \boldsymbol{R}\boldsymbol{p}_0 + \boldsymbol{t} = \boldsymbol{R}(d_0\hat{\boldsymbol{x}}_0) + \boldsymbol{t}, \tag{7.7}$$

where $\hat{\boldsymbol{x}}_j = \boldsymbol{K}_j^{-1}\boldsymbol{x}_j$ are the (local) ray direction vectors. Taking the cross product of both sides with $\boldsymbol{t}$ in order to annihilate it on the right hand side yields[1]

$$d_1[\boldsymbol{t}]_\times\hat{\boldsymbol{x}}_1 = d_0[\boldsymbol{t}]_\times\boldsymbol{R}\hat{\boldsymbol{x}}_0. \tag{7.8}$$

Taking the dot product of both sides with $\hat{\boldsymbol{x}}_1$ yields

$$d_0\hat{\boldsymbol{x}}_1^T([\boldsymbol{t}]_\times\boldsymbol{R})\hat{\boldsymbol{x}}_0 = d_1\hat{\boldsymbol{x}}_1^T[\boldsymbol{t}]_\times\hat{\boldsymbol{x}}_1 = 0, \tag{7.9}$$

since the right hand side is a triple product with two identical entries. (Another way to say this is that the cross product matrix $[\boldsymbol{t}]_\times$ is skew symmetric and returns 0 when pre- and post-multiplied by the same vector.)

[1] The cross-product operator $[\,]_\times$ was introduced in (2.32).

We therefore arrive at the basic *epipolar constraint*

$$\hat{\boldsymbol{x}}_1^T \boldsymbol{E} \hat{\boldsymbol{x}}_0 = 0, \tag{7.10}$$

where

$$\boldsymbol{E} = [\boldsymbol{t}]_\times \boldsymbol{R} \tag{7.11}$$

is called the *essential matrix* (Longuet-Higgins 1981).

An alternative way to derive the epipolar constraint is to notice that in order for the cameras to be oriented so that the rays $\hat{\boldsymbol{x}}_0$ and $\hat{\boldsymbol{x}}_1$ intersect in 3D at point $\boldsymbol{p}$, the vectors connecting the two camera centers $\boldsymbol{c}_1 - \boldsymbol{c}_0 = -\boldsymbol{R}_1^{-1}\boldsymbol{t}$ and the rays corresponding to pixels $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$, namely $\boldsymbol{R}_j^{-1}\hat{\boldsymbol{x}}_j$, must be co-planar. This requires that the triple product

$$(\hat{\boldsymbol{x}}_0, \boldsymbol{R}^{-1}\hat{\boldsymbol{x}}_1, -\boldsymbol{R}^{-1}\boldsymbol{t}) = (\boldsymbol{R}\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}_1, -\boldsymbol{t}) = \hat{\boldsymbol{x}}_1 \cdot (\boldsymbol{t} \times \boldsymbol{R}\hat{\boldsymbol{x}}_0) = \hat{\boldsymbol{x}}_1^T([\boldsymbol{t}]_\times \boldsymbol{R})\hat{\boldsymbol{x}}_0 = 0. \tag{7.12}$$

Notice that the essential matrix $\boldsymbol{E}$ maps a point $\hat{\boldsymbol{x}}_0$ in image 0 into a line $\boldsymbol{l}_1 = \boldsymbol{E}\hat{\boldsymbol{x}}_0$ in image 1, since $\hat{\boldsymbol{x}}_1^T\boldsymbol{l}_1 = 0$ (Figure 7.3). All such lines must pass through the second *epipole* $\boldsymbol{e}_1$, which is therefore defined as the left singular vector of $\boldsymbol{E}$ with a 0 singular value, or, equivalently, the projection of the vector $\boldsymbol{t}$ into image 1. The dual (transpose) of these relationships gives us the epipolar line in the first image as $\boldsymbol{l}_0 = \boldsymbol{E}^T\hat{\boldsymbol{x}}_1$ and $\boldsymbol{e}_0$ as the zero-value right singular vector of $\boldsymbol{E}$.

Given this fundamental relationship (7.10), how can we use it to recover the camera motion encoded in the essential matrix $\boldsymbol{E}$? If we have N corresponding measurements $\{(\boldsymbol{x}_{i0}, \boldsymbol{x}_{i1})\}$, we can form N homogeneous equations in the nine elements of $\boldsymbol{E} = \{e_{00} \dots e_{22}\}$,

$$\begin{aligned} x_{i0}x_{i1}e_{00} &+ y_{i0}x_{i1}e_{01} &+ x_{i1}e_{02} &+ \\ x_{i0}y_{i1}e_{00} &+ y_{i0}y_{i1}e_{11} &+ y_{i1}e_{12} &+ \\ x_{i0}e_{20} &+ y_{i0}e_{21} &+ e_{22} &= 0 \end{aligned} \tag{7.13}$$

where $\boldsymbol{x}_{ij} = (x_{ij}, y_{ij}, 1)$. This can be written more compactly as

$$[\boldsymbol{x}_{i1}\, \boldsymbol{x}_{i0}^T] \otimes \boldsymbol{E} = \boldsymbol{Z}_i \otimes \boldsymbol{E} = \boldsymbol{z}_i \cdot \boldsymbol{f} = 0, \tag{7.14}$$

where $\otimes$ indicates an element-wise multiplication and summation of matrix elements, and $\boldsymbol{z}_i$ and $\boldsymbol{f}$ are the rasterized (vector) forms of the $\boldsymbol{Z}_i = \hat{\boldsymbol{x}}_{i1}\hat{\boldsymbol{x}}_{i0}^T$ and $\boldsymbol{E}$ matrices.[2] Given $N \geq 8$ such equations, we can compute an estimate (up to scale) for the entries in $\boldsymbol{E}$ using an SVD.

In the presence of noisy measurements, how close is this estimate to being statistically optimal? If you look at the entries in (7.13), you can see that some entries are the products of image measurements such as $x_{i0}y_{i1}$ and others are direct image measurements (or even the identity). If the measurements have comparable noise, the terms that are products of measurements have their noise amplified by the other element in the product, which can lead to very poor scaling, e.g., an inordinately large influence of points with large coordinates (far away from the image center).

In order to counteract this trend, Hartley (1997a) suggests that the point coordinates should be translated and scaled so that their centroid lies at the origin and their variance is unity, i.e.,

$$\tilde{x}_i = s(x_i - \mu_x) \tag{7.15}$$

$$\tilde{y}_i = s(x_i - \mu_y) \tag{7.16}$$

[2] We use $\boldsymbol{f}$ instead of $\boldsymbol{e}$ to denote the rasterized form of $\boldsymbol{E}$ to avoid confusion with the epipoles $\boldsymbol{e}_j$.

such that $\sum_i \tilde{x}_i = \sum_i \tilde{y}_i = 0$ and $\sum_i \tilde{x}_i^2 + \sum_i \tilde{y}_i^2 = 2n$, where n is the number of points.[3]

Once the essential matrix $\tilde{\boldsymbol{E}}$ has been computed from the transformed coordinates $\{(\tilde{\boldsymbol{x}}_{i0}, \tilde{\boldsymbol{x}}_{i1})\}$, where $\tilde{\boldsymbol{x}}_{ij} = \boldsymbol{T}_j \hat{\boldsymbol{x}}_{ij}$, the original essential matrix $\boldsymbol{E}$ can be recovered as

$$\boldsymbol{E} = \boldsymbol{T}_1 \tilde{\boldsymbol{E}} \boldsymbol{T}_0. \tag{7.17}$$

In his paper, Hartley (1997a) compares the improvement due to his re-normalization strategy to alternative distance measures proposed by others such as Zhang (1998a,b) and concludes that his simple re-normalization in most cases is as effective as (or better than) alternative techniques. Torr and Fitzgibbon (2004) recommend a variant on this algorithm where the norm of the upper 2×2 sub-matrix of $\boldsymbol{E}$ is set to 1 and show that it has even better stability with respect to 2D coordinate transformations.

Once an estimate for the essential matrix $\boldsymbol{E}$ has been recovered, the direction of the translation vector $\boldsymbol{t}$ can be estimated. Note that the absolute distance between the two cameras can never be recovered from pure image measurements alone, regardless of how many cameras or points are used. Knowledge about absolute camera and point positions or distances, often called *ground control points* in photogrammetry, is always required to establish the final scale, position, and orientation.

To estimate this direction $\hat{\boldsymbol{t}}$, observe that under ideal noise-free conditions, the essential matrix $\boldsymbol{E}$ is singular, i.e., $\hat{\boldsymbol{t}}^T \boldsymbol{E} = 0$. This singularity shows up as a singular value of 0 when an SVD of $\boldsymbol{E}$ is performed,

$$\boldsymbol{E} = [\hat{\boldsymbol{t}}]_\times \boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \hat{\boldsymbol{t}} \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_0^T \\ \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \end{bmatrix} \tag{7.18}$$

When $\boldsymbol{E}$ is computed from noisy measurements, the singular vector associated with the smallest singular value gives us $\hat{\boldsymbol{t}}$. (The other two singular values should be similar but are not, in general, equal to 1 because $\boldsymbol{E}$ is only computed up to an unknown scale.)

Because $\boldsymbol{E}$ is rank-deficient, it turns out that we actually only need seven correspondences of the form of Equation (7.14) instead of eight to estimate this matrix (Hartley 1994a; Torr and Murray 1997; Hartley and Zisserman 2004). (The advantage of using fewer correspondences inside a RANSAC robust fitting stage is that fewer random samples need to be generated.) From this set of seven homogeneous equations (which we can stack into a 7×9 matrix for SVD analysis), we can find two independent vectors, say $\boldsymbol{f}_0$ and $\boldsymbol{f}_1$ such that $\boldsymbol{z}_i \cdot \boldsymbol{f}_j = 0$. These two vectors can be converted back into 3×3 matrices $\boldsymbol{E}_0$ and $\boldsymbol{E}_1$, which span the solution space for

$$\boldsymbol{E} = \alpha \boldsymbol{E}_0 + (1 - \alpha) \boldsymbol{E}_1. \tag{7.19}$$

To find the correct value of α, we observe that $\boldsymbol{E}$ has a zero determinant, since it is rank deficient, and hence

$$\det |\alpha \boldsymbol{E}_0 + (1 - \alpha) \boldsymbol{E}_1| = 0. \tag{7.20}$$

This gives us a cubic equation in α, which has either one or three solutions (roots). Substituting these values into (7.19) to obtain $\boldsymbol{E}$, we can test this essential matrix against other unused feature correspondences to select the correct one.

[3] More precisely, Hartley (1997a) suggests scaling the points "so that the average distance from the origin is equal to $\sqrt{2}$" but the heuristic of unit variance is faster to compute (does not require per-point square roots) and should yield comparable improvements.

Once $\hat{\boldsymbol{t}}$ has been recovered, how can we estimate the corresponding rotation matrix $\boldsymbol{R}$? Recall that the cross-product operator $[\hat{\boldsymbol{t}}]_\times$ (2.32) projects a vector onto a set of orthogonal basis vectors that include $\hat{\boldsymbol{t}}$, zeros out the $\hat{\boldsymbol{t}}$ component, and rotates the other two by 90°,

$$[\hat{\boldsymbol{t}}]_\times = \boldsymbol{S}\boldsymbol{Z}\boldsymbol{R}_{90^\circ}\boldsymbol{S}^T = \begin{bmatrix} \boldsymbol{s}_0 & \boldsymbol{s}_1 & \hat{\boldsymbol{t}} \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & \\ 1 & 0 & \\ & & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_0^T \\ \boldsymbol{s}_1^T \\ \hat{\boldsymbol{t}}^T \end{bmatrix}, \quad (7.21)$$

where $\hat{\boldsymbol{t}} = \boldsymbol{s}_0 \times \boldsymbol{s}_1$. From Equations (7.18 and 7.21), we get

$$\boldsymbol{E} = [\hat{\boldsymbol{t}}]_\times \boldsymbol{R} = \boldsymbol{S}\boldsymbol{Z}\boldsymbol{R}_{90^\circ}\boldsymbol{S}^T\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \quad (7.22)$$

from which we can conclude that $\boldsymbol{S} = \boldsymbol{U}$. Recall that for a noise-free essential matrix, ($\boldsymbol{\Sigma} = \boldsymbol{Z}$), and hence

$$\boldsymbol{R}_{90^\circ}\boldsymbol{U}^T\boldsymbol{R} = \boldsymbol{V}^T \quad (7.23)$$

and

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{R}_{90^\circ}^T\boldsymbol{V}^T. \quad (7.24)$$

Unfortunately, we only know both $\boldsymbol{E}$ and $\hat{\boldsymbol{t}}$ up to a sign. Furthermore, the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are not guaranteed to be rotations (you can flip both their signs and still get a valid SVD). For this reason, we have to generate all four possible rotation matrices

$$\boldsymbol{R} = \pm\boldsymbol{U}\boldsymbol{R}_{\pm 90^\circ}^T\boldsymbol{V}^T \quad (7.25)$$

and keep the two whose determinant $|\boldsymbol{R}| = 1$. To disambiguate between the remaining pair of potential rotations, which form a *twisted pair* (Hartley and Zisserman 2004, p. 240), we need to pair them with both possible signs of the translation direction $\pm\hat{\boldsymbol{t}}$ and select the combination for which the largest number of points is seen in front of both cameras.[4]

The property that points must lie in front of the camera, i.e., at a positive distance along the viewing rays emanating from the camera, is known as *chirality* (Hartley 1998). In addition to determining the signs of the rotation and translation, as described above, the chirality (sign of the distances) of the points in a reconstruction can be used inside a RANSAC procedure (along with the reprojection errors) to distinguish between likely and unlikely configurations.[5] Chirality can also be used to transform projective reconstructions (Sections 7.2.1 and 7.2.2) into *quasi-affine* reconstructions (Hartley 1998).

The normalized "eight-point algorithm" (Hartley 1997a) described above is not the only way to estimate the camera motion from correspondences. Variants include using seven points while enforcing the rank two constraint in $\boldsymbol{E}$ (7.19–7.20) and a five-point algorithm that requires finding the roots of a 10th degree polynomial (Nistér 2004). Since such algorithms use fewer points to compute their estimates, they are less sensitive to outliers when used as part of a random sampling (RANSAC) strategy.

[4] In the noise-free case, a single point suffices. It is safer, however, to test all or a sufficient subset of points, downweighting the ones that lie close to the plane at infinity, for which it is easy to get depth reversals.

[5] Note that as points get further away from a camera, i.e., closer toward the plane at infinity, errors in chirality become more likely.

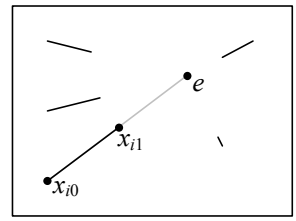


Figure 7.4 Pure translational camera motion results in visual motion where all the points move towards (or away from) a common *focus of expansion* (FOE) $\boldsymbol{e}$. They therefore satisfy the triple product condition $(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{e}) = \boldsymbol{e} \cdot (\boldsymbol{x}_0 \times \boldsymbol{x}_1) = 0$.

Pure translation (known rotation)

In the case where we know the rotation, we can pre-rotate the points in the second image to match the viewing direction of the first. The resulting set of 3D points all move towards (or away from) the *focus of expansion* (FOE), as shown in Figure 7.4.[6] The resulting essential matrix $\boldsymbol{E}$ is (in the noise-free case) skew symmetric and so can be estimated more directly by setting $e_{ij} = -e_{ji}$ and $e_{ii} = 0$ in (7.13). Two points with non-zero parallax now suffice to estimate the FOE.

A more direct derivation of the FOE estimate can be obtained by minimizing the triple product

$$\sum_i (\boldsymbol{x}_{i0}, \boldsymbol{x}_{i1}, \boldsymbol{e})^2 = \sum_i ((\boldsymbol{x}_{i0} \times \boldsymbol{x}_{i1}) \cdot \boldsymbol{e})^2, \tag{7.26}$$

which is equivalent to finding the null space for the set of equations

$$(y_{i0} - y_{i1})e_0 + (x_{i1} - x_{i0})e_1 + (x_{i0}y_{i1} - y_{i0}x_{i1})e_2 = 0. \tag{7.27}$$

Note that, as in the eight-point algorithm, it is advisable to normalize the 2D points to have unit variance before computing this estimate.

In situations where a large number of points at infinity are available, e.g., when shooting outdoor scenes or when the camera motion is small compared to distant objects, this suggests an alternative RANSAC strategy for estimating the camera motion. First, pick a pair of points to estimate a rotation, hoping that both of the points lie at infinity (very far from the camera). Then, compute the FOE and check whether the residual error is small (indicating agreement with this rotation hypothesis) and whether the motions towards or away from the epipole (FOE) are all in the same direction (ignoring very small motions, which may be noise-contaminated).

Pure rotation

The case of pure rotation results in a degenerate estimate of the essential matrix $\boldsymbol{E}$ and of the translation direction $\hat{\boldsymbol{t}}$. Consider first the case of the rotation matrix being known. The estimates for the FOE will be degenerate, since $\boldsymbol{x}_{i0} \approx \boldsymbol{x}_{i1}$, and hence (7.27), is degenerate. A similar argument shows that the equations for the essential matrix (7.13) are also rank-deficient.

[6] Fans of *Star Trek* and *Star Wars* will recognize this as the "jump to hyperdrive" visual effect.

This suggests that it might be prudent before computing a full essential matrix to first compute a rotation estimate $\boldsymbol{R}$ using (6.32), potentially with just a small number of points, and then compute the residuals after rotating the points before proceeding with a full $\boldsymbol{E}$ computation.

7.2.1 Projective (uncalibrated) reconstruction

In many cases, such as when trying to build a 3D model from Internet or legacy photos taken by unknown cameras without any EXIF tags, we do not know ahead of time the intrinsic calibration parameters associated with the input images. In such situations, we can still estimate a two-frame reconstruction, although the true metric structure may not be available, e.g., orthogonal lines or planes in the world may not end up being reconstructed as orthogonal.

Consider the derivations we used to estimate the essential matrix $\boldsymbol{E}$ (7.10–7.12). In the uncalibrated case, we do not know the calibration matrices $\boldsymbol{K}_j$, so we cannot use the normalized ray directions $\hat{\boldsymbol{x}}_j = \boldsymbol{K}_j^{-1}\boldsymbol{x}_j$. Instead, we have access only to the image coordinates $\boldsymbol{x}_j$, and so the essential matrix (7.10) becomes

$$\hat{\boldsymbol{x}}_1^T \boldsymbol{E} \hat{\boldsymbol{x}}_1 = \boldsymbol{x}_1^T \boldsymbol{K}_1^{-T} \boldsymbol{E} \boldsymbol{K}_0^{-1} \boldsymbol{x}_0 = \boldsymbol{x}_1^T \boldsymbol{F} \boldsymbol{x}_0 = 0, \tag{7.28}$$

where

$$\boldsymbol{F} = \boldsymbol{K}_1^{-T} \boldsymbol{E} \boldsymbol{K}_0^{-1} = [\boldsymbol{e}]_\times \tilde{\boldsymbol{H}} \tag{7.29}$$

is called the *fundamental matrix* (Faugeras 1992; Hartley, Gupta, and Chang 1992; Hartley and Zisserman 2004).

Like the essential matrix, the fundamental matrix is (in principle) rank two,

$$\boldsymbol{F} = [\boldsymbol{e}]_\times \tilde{\boldsymbol{H}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \boldsymbol{e}_1 \end{bmatrix} \begin{bmatrix} \sigma_0 & & \\ & \sigma_1 & \\ & & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_0^T \\ \boldsymbol{v}_1^T \\ \boldsymbol{e}_0^T \end{bmatrix}. \tag{7.30}$$

Its smallest left singular vector indicates the epipole $\boldsymbol{e}_1$ in the image 1 and its smallest right singular vector is $\boldsymbol{e}_0$ (Figure 7.3). The homography $\tilde{\boldsymbol{H}}$ in (7.29), which in principle should equal

$$\tilde{\boldsymbol{H}} = \boldsymbol{K}_1^{-T} \boldsymbol{R} \boldsymbol{K}_0^{-1}, \tag{7.31}$$

cannot be uniquely recovered from $\boldsymbol{F}$, since any homography of the form $\tilde{\boldsymbol{H}}' = \tilde{\boldsymbol{H}} + \boldsymbol{e}\boldsymbol{v}^T$ results in the same $\boldsymbol{F}$ matrix. (Note that $[\boldsymbol{e}]_\times$ annihilates any multiple of $\boldsymbol{e}$.)

Any one of these valid homographies $\tilde{\boldsymbol{H}}$ maps some plane in the scene from one image to the other. It is not possible to tell in advance which one it is without either selecting four or more co-planar correspondences to compute $\tilde{\boldsymbol{H}}$ as part of the $\boldsymbol{F}$ estimation process (in a manner analogous to guessing a rotation for $\boldsymbol{E}$) or mapping all points in one image through $\tilde{\boldsymbol{H}}$ and seeing which ones line up with their corresponding locations in the other.[7]

In order to create a *projective* reconstruction of the scene, we can pick any valid homography $\tilde{\boldsymbol{H}}$ that satisfies Equation (7.29). For example, following a technique analogous to Equations (7.18–7.24), we get

$$\boldsymbol{F} = [\boldsymbol{e}]_\times \tilde{\boldsymbol{H}} = \boldsymbol{S}\boldsymbol{Z}\boldsymbol{R}_{90^\circ}\boldsymbol{S}^T\tilde{\boldsymbol{H}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{7.32}$$

[7] This process is sometimes referred to as *plane plus parallax* (Section 2.1.5) (Kumar, Anandan, and Hanna 1994; Sawhney 1994).

and hence

$$\tilde{\boldsymbol{H}} = \boldsymbol{U}\boldsymbol{R}_{90^\circ}^T \hat{\boldsymbol{\Sigma}} \boldsymbol{V}^T, \tag{7.33}$$

where $\hat{\boldsymbol{\Sigma}}$ is the singular value matrix with the smallest value replaced by a reasonable alternative (say, the middle value).[8] We can then form a pair of camera matrices

$$\boldsymbol{P}_0 = [\boldsymbol{I}|\boldsymbol{0}] \quad \text{and} \quad \boldsymbol{P}_0 = [\tilde{\boldsymbol{H}}|\boldsymbol{e}], \tag{7.34}$$

from which a projective reconstruction of the scene can be computed using triangulation (Section 7.1).

While the projective reconstruction may not be useful in practice, it can often be *upgraded* to an affine or metric reconstruction, as detailed below. Even without this step, however, the fundamental matrix $\boldsymbol{F}$ can be very useful in finding additional correspondences, as they must all lie on corresponding epipolar lines, i.e., any feature $\boldsymbol{x}_0$ in image 0 must have its correspondence lying on the associated epipolar line $\boldsymbol{l}_1 = \boldsymbol{F}\boldsymbol{x}_0$ in image 1, assuming that the point motions are due to a rigid transformation.

7.2.2 Self-calibration

The results of structure from motion computation are much more useful (and intelligible) if a *metric* reconstruction is obtained, i.e., one in which parallel lines are parallel, orthogonal walls are at right angles, and the reconstructed model is a scaled version of reality. Over the years, a large number of *self-calibration* (or *auto-calibration*) techniques have been developed for converting a projective reconstruction into a metric one, which is equivalent to recovering the unknown calibration matrices $\boldsymbol{K}_j$ associated with each image (Hartley and Zisserman 2004; Moons, Van Gool, and Vergauwen 2010).

In situations where certain additional information is known about the scene, different methods may be employed. For example, if there are parallel lines in the scene (usually, having several lines converge on the same vanishing point is good evidence), three or more vanishing points, which are the images of points at infinity, can be used to establish the homography for the plane at infinity, from which focal lengths and rotations can be recovered. If two or more finite *orthogonal* vanishing points have been observed, the single-image calibration method based on vanishing points (Section 6.3.2) can be used instead.

In the absence of such external information, it is not possible to recover a fully parameterized independent calibration matrix $\boldsymbol{K}_j$ for each image from correspondences alone. To see this, consider the set of all camera matrices $\boldsymbol{P}_j = \boldsymbol{K}_j[\boldsymbol{R}_j|\boldsymbol{t}_j]$ projecting world coordinates $\boldsymbol{p}_i = (X_i, Y_i, Z_i, W_i)$ into screen coordinates $\boldsymbol{x}_{ij} \sim \boldsymbol{P}_j\boldsymbol{p}_i$. Now consider transforming the 3D scene $\{\boldsymbol{p}_i\}$ through an arbitrary 4×4 projective transformation $\tilde{\boldsymbol{H}}$, yielding a new model consisting of points $\boldsymbol{p}'_i = \tilde{\boldsymbol{H}}\boldsymbol{p}_i$. Post-multiplying each $\boldsymbol{P}_j$ matrix by $\tilde{\boldsymbol{H}}^{-1}$ still produces the same screen coordinates and a new set calibration matrices can be computed by applying RQ decomposition to the new camera matrix $\boldsymbol{P}'_j = \boldsymbol{P}_j\tilde{\boldsymbol{H}}^{-1}$.

For this reason, all self-calibration methods assume some restricted form of the calibration matrix, either by setting or equating some of their elements or by assuming that they do not vary over time. While most of the techniques discussed by Hartley and Zisserman (2004);

[8] Hartley and Zisserman (2004, p. 237) recommend using $\tilde{\boldsymbol{H}} = [\boldsymbol{e}]_\times \boldsymbol{F}$ (Luong and Viéville 1996), which places the camera on the plane at infinity.

Moons, Van Gool, and Vergauwen (2010) require three or more frames, in this section we present a simple technique that can recover the focal lengths (f_0, f_1) of both images from the fundamental matrix $\boldsymbol{F}$ in a two-frame reconstruction (Hartley and Zisserman 2004, p. 456).

To accomplish this, we assume that the camera has zero skew, a known aspect ratio (usually set to 1), and a known optical center, as in Equation (2.59). How reasonable is this assumption in practice? The answer, as with many questions, is "it depends".

If absolute metric accuracy is required, as in photogrammetry applications, it is imperative to pre-calibrate the cameras using one of the techniques from Section 6.3 and to use ground control points to pin down the reconstruction. If instead, we simply wish to reconstruct the world for visualization or image-based rendering applications, as in the Photo Tourism system of Snavely, Seitz, and Szeliski (2006), this assumption is quite reasonable in practice.

Most cameras today have square pixels and an optical center near the middle of the image, and are much more likely to deviate from a simple camera model due to radial distortion (Section 6.3.5), which should be compensated for whenever possible. The biggest problems occur when images have been cropped off-center, in which case the optical center will no longer be in the middle, or when perspective pictures have been taken of a different picture, in which case a general camera matrix becomes necessary.[9]

Given these caveats, the two-frame focal length estimation algorithm based on the Kruppa equations developed by Hartley and Zisserman (2004, p. 456) proceeds as follows. Take the left and right singular vectors $\{\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{v}_0, \boldsymbol{v}_1\}$ of the fundamental matrix $\boldsymbol{F}$ (7.30) and their associated singular values $\{\sigma_0, \sigma_1\}$ and form the following set of equations:

$$\frac{\boldsymbol{u}_1^T \boldsymbol{D}_0 \boldsymbol{u}_1}{\sigma_0^2 \boldsymbol{v}_0^T \boldsymbol{D}_1 \boldsymbol{v}_0} = -\frac{\boldsymbol{u}_0^T \boldsymbol{D}_0 \boldsymbol{u}_1}{\sigma_0 \sigma_1 \boldsymbol{v}_0^T \boldsymbol{D}_1 \boldsymbol{v}_1} = \frac{\boldsymbol{u}_0^T \boldsymbol{D}_0 \boldsymbol{u}_0}{\sigma_1^2 \boldsymbol{v}_1^T \boldsymbol{D}_1 \boldsymbol{v}_1}, \tag{7.35}$$

where the two matrices

$$\boldsymbol{D}_j = \boldsymbol{K}_j \boldsymbol{K}_j^T = \mathrm{diag}(f_j^2, f_j^2, 1) = \begin{bmatrix} f_j^2 & & \\ & f_j^2 & \\ & & 1 \end{bmatrix} \tag{7.36}$$

encode the unknown focal lengths. For simplicity, let us rewrite each of the numerators and denominators in (7.35) as

$$e_{ij0}(f_0^2) = \boldsymbol{u}_i^T \boldsymbol{D}_0 \boldsymbol{u}_j = a_{ij} + b_{ij} f_0^2, \tag{7.37}$$

$$e_{ij1}(f_1^2) = \sigma_i \sigma_j \boldsymbol{v}_i^T \boldsymbol{D}_1 \boldsymbol{v}_j = c_{ij} + d_{ij} f_1^2. \tag{7.38}$$

Notice that each of these is affine (linear plus constant) in either f_0^2 or f_1^2. Hence, we can cross-multiply these equations to obtain quadratic equations in f_j^2, which can readily be solved. (See also the work by Bougnoux (1998) for some alternative formulations.)

An alternative solution technique is to observe that we have a set of three equations related by an unknown scalar λ, i.e.,

$$e_{ij0}(f_0^2) = \lambda e_{ij1}(f_1^2) \tag{7.39}$$

(Richard Hartley, personal communication, July 2009). These can readily be solved to yield $(f_0^2, \lambda f_1^2, \lambda)$ and hence (f_0, f_1).

[9] In Photo Tourism, our system registered photographs of an information sign outside Notre Dame with real pictures of the cathedral.

How well does this approach work in practice? There are certain degenerate configurations, such as when there is no rotation or when the optical axes intersect, when it does not work at all. (In such a situation, you can vary the focal lengths of the cameras and obtain a deeper or shallower reconstruction, which is an example of a *bas-relief ambiguity* (Section 7.4.3).) Hartley and Zisserman (2004) recommend using techniques based on three or more frames. However, if you find two images for which the estimates of $(f_0^2, \lambda f_1^2, \lambda)$ are well conditioned, they can be used to initialize a more complete bundle adjustment of all the parameters (Section 7.4). An alternative, which is often used in systems such as Photo Tourism, is to use camera EXIF tags or generic default values to initialize focal length estimates and refine them as part of bundle adjustment.

7.2.3 *Application*: View morphing

An interesting application of basic two-frame structure from motion is *view morphing* (also known as *view interpolation*, see Section 13.1), which can be used to generate a smooth 3D animation from one view of a 3D scene to another (Chen and Williams 1993; Seitz and Dyer 1996).

To create such a transition, you must first smoothly interpolate the camera matrices, i.e., the camera positions, orientations, and focal lengths. While simple linear interpolation can be used (representing rotations as quaternions (Section 2.1.4)), a more pleasing effect is obtained by *easing in* and *easing out* the camera parameters, e.g., using a raised cosine, as well as moving the camera along a more circular trajectory (Snavely, Seitz, and Szeliski 2006).

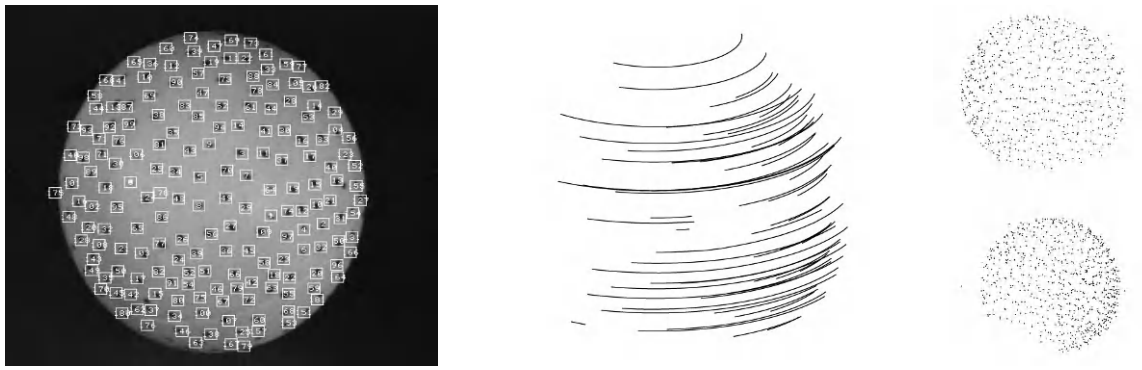
To generate in-between frames, either a full set of 3D correspondences needs to be established (Section 11.3) or 3D models (proxies) must be created for each reference view. Section 13.1 describes several widely used approaches to this problem. One of the simplest is to just triangulate the set of matched feature points in each image, e.g., using Delaunay triangulation. As the 3D points are re-projected into their intermediate views, pixels can be mapped from their original source images to their new views using affine or projective mapping (Szeliski and Shum 1997). The final image is then composited using a linear blend of the two reference images, as with usual morphing (Section 3.6.3).

7.3 Factorization

When processing video sequences, we often get extended *feature tracks* (Section 4.1.4) from which it is possible to recover the structure and motion using a process called *factorization*. Consider the tracks generated by a rotating ping pong ball, which has been marked with dots to make its shape and motion more discernable (Figure 7.5). We can readily see from the shape of the tracks that the moving object must be a sphere, but how can we infer this mathematically?

It turns out that, under orthography or related models we discuss below, the shape and motion can be recovered simultaneously using a singular value decomposition (Tomasi and Kanade 1992). Consider the orthographic and weak perspective projection models introduced in Equations (2.47–2.49). Since the last row is always $[0\ 0\ 0\ 1]$, there is no perspective division and we can write

$$\boldsymbol{x}_{ji} = \tilde{\boldsymbol{P}}_j \bar{\boldsymbol{p}}_i, \tag{7.40}$$



(a) (b) (c)

Figure 7.5 3D reconstruction of a rotating ping pong ball using factorization (Tomasi and Kanade 1992) © 1992 Springer: (a) sample image with tracked features overlaid; (b) subsampled feature motion stream; (c) two views of the reconstructed 3D model.

where $\boldsymbol{x}_{ji}$ is the location of the ith point in the jth frame, $\tilde{\boldsymbol{P}}_j$ is the upper 2×4 portion of the projection matrix $\boldsymbol{P}_j$, and $\bar{\boldsymbol{p}}_i = (X_i, Y_i, Z_i, 1)$ is the augmented 3D point position.[10]

Let us assume (for now) that every point i is visible in every frame j. We can take the *centroid* (average) of the projected point locations $\boldsymbol{x}_{ji}$ in frame j,

$$\bar{\boldsymbol{x}}_j = \frac{1}{N}\sum_i \boldsymbol{x}_{ji} = \tilde{\boldsymbol{P}}_j \frac{1}{N}\sum_i \bar{\boldsymbol{p}}_i = \tilde{\boldsymbol{P}}_j \bar{\boldsymbol{c}}, \tag{7.41}$$

where $\bar{\boldsymbol{c}} = (\bar{X}, \bar{Y}, \bar{Z}, 1)$ is the augmented 3D centroid of the point cloud.

Since world coordinate frames in structure from motion are always arbitrary, i.e., we cannot recover true 3D locations without ground control points (known measurements), we can place the origin of the world at the centroid of the points, i.e, $\bar{X} = \bar{Y} = \bar{Z} = 0$, so that $\bar{\boldsymbol{c}} = (0, 0, 0, 1)$. We see from this that the centroid of the 2D points in each frame $\bar{\boldsymbol{x}}_j$ directly gives us the last element of $\tilde{\boldsymbol{P}}_j$.

Let $\tilde{\boldsymbol{x}}_{ji} = \boldsymbol{x}_{ji} - \bar{\boldsymbol{x}}_j$ be the 2D point locations after their image centroid has been subtracted. We can now write

$$\tilde{\boldsymbol{x}}_{ji} = \boldsymbol{M}_j \boldsymbol{p}_i, \tag{7.42}$$

where $\boldsymbol{M}_j$ is the upper 2×3 portion of the projection matrix $\boldsymbol{P}_j$ and $\boldsymbol{p}_i = (X_i, Y_i, Z_i)$. We can concatenate all of these measurement equations into one large matrix

$$\hat{\boldsymbol{X}} = \begin{bmatrix} \tilde{\boldsymbol{x}}_{11} & \cdots & \tilde{\boldsymbol{x}}_{1i} & \cdots & \tilde{\boldsymbol{x}}_{1N} \\ \vdots & & \vdots & & \vdots \\ \tilde{\boldsymbol{x}}_{j1} & \cdots & \tilde{\boldsymbol{x}}_{ji} & \cdots & \tilde{\boldsymbol{x}}_{jN} \\ \vdots & & \vdots & & \vdots \\ \tilde{\boldsymbol{x}}_{M1} & \cdots & \tilde{\boldsymbol{x}}_{Mi} & \cdots & \tilde{\boldsymbol{x}}_{MN} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_1 \\ \vdots \\ \boldsymbol{M}_j \\ \vdots \\ \boldsymbol{M}_M \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 & \cdots & \boldsymbol{p}_i & \cdots & \boldsymbol{p}_N \end{bmatrix} = \hat{\boldsymbol{M}}\hat{\boldsymbol{S}}. \tag{7.43}$$

$\hat{\boldsymbol{X}}$ is called the *measurement* matrix and $\hat{\boldsymbol{M}}$ and ($\hat{\boldsymbol{S}}$ are the *motion*) and *structure* matrices, respectively (Tomasi and Kanade 1992).

[10] In this section, we index the 2D point positions as $\boldsymbol{x}_{ji}$ instead of $\boldsymbol{x}_{ij}$, since this is the convention adopted by factorization papers (Tomasi and Kanade 1992) and is consistent with the factorization given in (7.43).

Because the motion matrix $\hat{\boldsymbol{M}}$ is $2M \times 3$ and the structure matrix $\hat{\boldsymbol{S}}$ is $3 \times N$, an SVD applied to $\hat{\boldsymbol{X}}$ has only three non-zero singular values. In the case where the measurements in $\hat{\boldsymbol{X}}$ are noisy, SVD returns the rank-three factorization of $\hat{\boldsymbol{X}}$ that is the closest to $\hat{\boldsymbol{X}}$ in a least squares sense (Tomasi and Kanade 1992; Golub and Van Loan 1996; Hartley and Zisserman 2004).

It would be nice if the SVD of $\hat{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ directly returned the matrices $\hat{\boldsymbol{M}}$ and $\hat{\boldsymbol{S}}$, but it does not. Instead, we can write the relationship

$$\hat{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = [\boldsymbol{U}\boldsymbol{Q}][\boldsymbol{Q}^{-1}\boldsymbol{\Sigma}\boldsymbol{V}^T] \tag{7.44}$$

and set $\hat{\boldsymbol{M}} = \boldsymbol{U}\boldsymbol{Q}$ and $\hat{\boldsymbol{S}} = \boldsymbol{Q}^{-1}\boldsymbol{\Sigma}\boldsymbol{V}^T$.[11]

How can we recover the values of the 3×3 matrix $\boldsymbol{Q}$? This depends on the motion model being used. In the case of orthographic projection (2.47), the entries in $\boldsymbol{M}_j$ are the first two rows of rotation matrices $\boldsymbol{R}_j$, so we have

$$\begin{aligned} \boldsymbol{m}_{j0} \cdot \boldsymbol{m}_{j0} &= \boldsymbol{u}_{2j}\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{u}_{2j}^T &&= 1, \\ \boldsymbol{m}_{j0} \cdot \boldsymbol{m}_{j1} &= \boldsymbol{u}_{2j}\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{u}_{2j+1}^T &&= 0, \\ \boldsymbol{m}_{j1} \cdot \boldsymbol{m}_{j1} &= \boldsymbol{u}_{2j+1}\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{u}_{2j+1}^T &&= 1, \end{aligned} \tag{7.45}$$

where $\boldsymbol{u}_k$ are the 3×1 *rows* of the matrix $\boldsymbol{U}$. This gives us a large set of equations for the entries in the matrix $\boldsymbol{Q}\boldsymbol{Q}^T$, from which the matrix $\boldsymbol{Q}$ can be recovered using a matrix square root (Appendix A.1.4). If we have scaled orthography (2.48), i.e., $\boldsymbol{M}_j = s_j\boldsymbol{R}_j$, the first and third equations are equal to s_j and can be set equal to each other.

Note that even once $\boldsymbol{Q}$ has been recovered, there still exists a bas-relief ambiguity, i.e., we can never be sure if the object is rotating left to right or if its depth reversed version is moving the other way. (This can be seen in the classic rotating Necker Cube visual illusion.) Additional cues, such as the appearance and disappearance of points, or perspective effects, both of which are discussed below, can be used to remove this ambiguity.

For motion models other than pure orthography, e.g., for scaled orthography or paraperspective, the approach above must be extended in the appropriate manner. Such techniques are relatively straightforward to derive from first principles; more details can be found in papers that extend the basic factorization approach to these more flexible models (Poelman and Kanade 1997). Additional extensions of the original factorization algorithm include multi-body rigid motion (Costeira and Kanade 1995), sequential updates to the factorization (Morita and Kanade 1997), the addition of lines and planes (Morris and Kanade 1998), and re-scaling the measurements to incorporate individual location uncertainties (Anandan and Irani 2002).

A disadvantage of factorization approaches is that they require a complete set of tracks, i.e., each point must be visible in each frame, in order for the factorization approach to work. Tomasi and Kanade (1992) deal with this problem by first applying factorization to smaller denser subsets and then using known camera (motion) or point (structure) estimates to *hallucinate* additional missing values, which allows them to incrementally incorporate more features and cameras. Huynh, Hartley, and Heyden (2003) extend this approach to view missing data as special cases of outliers. Buchanan and Fitzgibbon (2005) develop fast iterative algorithms for performing large matrix factorizations with missing data. The general topic of

[11] Tomasi and Kanade (1992) first take the square root of $\boldsymbol{\Sigma}$ and distribute this to $\boldsymbol{U}$ and $\boldsymbol{V}$, but there is no particular reason to do this.

principal component analysis (PCA) with missing data also appears in other computer vision problems (Shum, Ikeuchi, and Reddy 1995; De la Torre and Black 2003; Gross, Matthews, and Baker 2006; Torresani, Hertzmann, and Bregler 2008; Vidal, Ma, and Sastry 2010).

7.3.1 Perspective and projective factorization

Another disadvantage of regular factorization is that it cannot deal with perspective cameras. One way to get around this problem is to perform an initial affine (e.g., orthographic) reconstruction and to then correct for the perspective effects in an iterative manner (Christy and Horaud 1996).

Observe that the object-centered projection model (2.76)

$$x_{ji} = s_j \frac{\boldsymbol{r}_{xj} \cdot \boldsymbol{p}_i + t_{xj}}{1 + \eta_j \boldsymbol{r}_{zj} \cdot \boldsymbol{p}_i} \tag{7.46}$$

$$y_{ji} = s_j \frac{\boldsymbol{r}_{yj} \cdot \boldsymbol{p}_i + t_{yj}}{1 + \eta_j \boldsymbol{r}_{zj} \cdot \boldsymbol{p}_i} \tag{7.47}$$

differs from the scaled orthographic projection model (7.40) by the inclusion of the denominator terms $(1 + \eta_j \boldsymbol{r}_{zj} \cdot \boldsymbol{p}_i)$.[12]

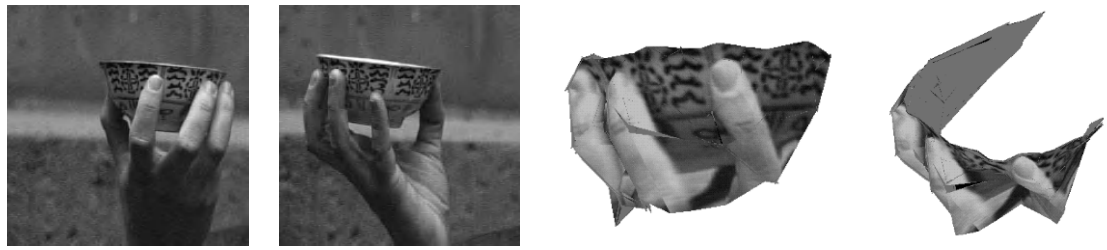
If we knew the correct values of $\eta_j = t_{zj}^{-1}$ and the structure and motion parameters $\boldsymbol{R}_j$ and $\boldsymbol{p}_i$, we could cross-multiply the left hand side (visible point measurements x_{ji} and y_{ji}) by the denominator and get corrected values, for which the bilinear projection model (7.40) is exact. In practice, after an initial reconstruction, the values of η_j can be estimated independently for each frame by comparing reconstructed and sensed point positions. (The third row of the rotation matrix $\boldsymbol{r}_{zj}$ is always available as the cross-product of the first two rows.) Note that since the η_j are determined from the image measurements, the cameras do not have to be pre-calibrated, i.e., their focal lengths can be recovered from $f_j = s_j/\eta_j$.

Once the η_j have been estimated, the feature locations can then be corrected before applying another round of factorization. Note that because of the initial depth reversal ambiguity, both reconstructions have to be tried while calculating η_j. (The incorrect reconstruction will result in a negative η_j, which is not physically meaningful.) Christy and Horaud (1996) report that their algorithm usually converges in three to five iterations, with the majority of the time spent in the SVD computation.

An alternative approach, which does not assume partially calibrated cameras (known optical center, square pixels, and zero skew) is to perform a fully *projective* factorization (Sturm and Triggs 1996; Triggs 1996). In this case, the inclusion of the third row of the camera matrix in (7.40) is equivalent to multiplying each reconstructed measurement $\boldsymbol{x}_{ji} = \boldsymbol{M}_j \boldsymbol{p}_i$ by its inverse (projective) depth $\eta_{ji} = d_{ji}^{-1} = 1/(\boldsymbol{P}_{j2}\boldsymbol{p}_i)$ or, equivalently, multiplying each measured position by its projective depth d_{ji},

$$\hat{\boldsymbol{X}} = \begin{bmatrix} d_{11}\tilde{\boldsymbol{x}}_{11} & \cdots & d_{1i}\tilde{\boldsymbol{x}}_{1i} & \cdots & d_{1N}\tilde{\boldsymbol{x}}_{1N} \\ \vdots & & \vdots & & \vdots \\ d_{j1}\tilde{\boldsymbol{x}}_{j1} & \cdots & d_{ji}\tilde{\boldsymbol{x}}_{ji} & \cdots & d_{jN}\tilde{\boldsymbol{x}}_{jN} \\ \vdots & & \vdots & & \vdots \\ d_{M1}\tilde{\boldsymbol{x}}_{M1} & \cdots & d_{Mi}\tilde{\boldsymbol{x}}_{Mi} & \cdots & d_{MN}\tilde{\boldsymbol{x}}_{MN} \end{bmatrix} = \hat{\boldsymbol{M}}\hat{\boldsymbol{S}}. \tag{7.48}$$

[12] Assuming that the optical center (c_x, c_y) lies at $(0, 0)$ and that pixels are square.



(a) (b) (c) (d)

Figure 7.6 3D teacup model reconstructed from a 240-frame video sequence (Tomasi and Kanade 1992) © 1992 Springer: (a) first frame of video; (b) last frame of video; (c) side view of 3D model; (d) top view of 3D model.

In the original paper by Sturm and Triggs (1996), the projective depths d_{ji} are obtained from two-frame reconstructions, while in later work (Triggs 1996; Oliensis and Hartley 2007), they are initialized to $d_{ji} = 1$ and updated after each iteration. Oliensis and Hartley (2007) present an update formula that is guaranteed to converge to a fixed point. None of these authors suggest actually estimating the third row of $\boldsymbol{P}_j$ as part of the projective depth computations. In any case, it is unclear when a fully projective reconstruction would be preferable to a partially calibrated one, especially if they are being used to initialize a full bundle adjustment of all the parameters.

One of the attractions of factorization methods is that they provide a "closed form" (sometimes called a "linear") method to initialize iterative techniques such as bundle adjustment. An alternative initialization technique is to estimate the homographies corresponding to some common plane seen by all the cameras (Rother and Carlsson 2002). In a calibrated camera setting, this can correspond to estimating consistent rotations for all of the cameras, for example, using matched vanishing points (Antone and Teller 2002). Once these have been recovered, the camera positions can then be obtained by solving a linear system (Antone and Teller 2002; Rother and Carlsson 2002; Rother 2003).

7.3.2 *Application*: Sparse 3D model extraction

Once a multi-view 3D reconstruction of the scene has been estimated, it then becomes possible to create a texture-mapped 3D model of the object and to look at it from new directions.

The first step is to create a denser 3D model than the sparse point cloud that structure from motion produces. One alternative is to run dense multi-view stereo (Sections 11.3–11.6). Alternatively, a simpler technique such as 3D triangulation can be used, as shown in Figure 7.6, in which 207 reconstructed 3D points are triangulated to produce a surface mesh.

In order to create a more realistic model, a *texture map* can be extracted for each triangle face. The equations to map points on the surface of a 3D triangle to a 2D image are straightforward: just pass the local 2D coordinates on the triangle through the 3×4 camera projection matrix to obtain a 3×3 homography (planar perspective projection). When multiple source images are available, as is usually the case in multi-view reconstruction, either the closest and most fronto-parallel image can be used or multiple images can be blended in to deal with view-dependent foreshortening (Wang, Kang, Szeliski *et al.* 2001) or to obtain super-resolved results (Goldluecke and Cremers 2009) Another alternative is to create a sep-

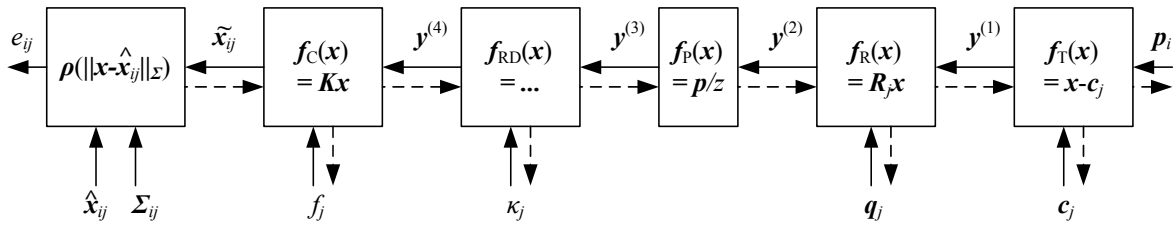


Figure 7.7 A set of chained transforms for projecting a 3D point $\boldsymbol{p}_i$ into a 2D measurement $\boldsymbol{x}_{ij}$ through a series of transformations $\boldsymbol{f}^{(k)}$, each of which is controlled by its own set of parameters. The dashed lines indicate the flow of information as partial derivatives are computed during a backward pass. The formula for the radial distortion function is $\boldsymbol{f}_{\mathrm{RD}}(\boldsymbol{x}) = (1 + \kappa_1 r^2 + \kappa_2 r^4)\boldsymbol{x}$.

arate texture map from each reference camera and to blend between them during rendering, which is known as *view-dependent texture mapping* (Section 13.1.1) (Debevec, Taylor, and Malik 1996; Debevec, Yu, and Borshukov 1998).

7.4 Bundle adjustment

As we have mentioned several times before, the most accurate way to recover structure and motion is to perform robust non-linear minimization of the measurement (re-projection) errors, which is commonly known in the photogrammetry (and now computer vision) communities as *bundle adjustment*.[13] Triggs, McLauchlan, Hartley *et al.* (1999) provide an excellent overview of this topic, including its historical development, pointers to the photogrammetry literature (Slama 1980; Atkinson 1996; Kraus 1997), and subtle issues with gauge ambiguities. The topic is also treated in depth in textbooks and surveys on multi-view geometry (Faugeras and Luong 2001; Hartley and Zisserman 2004; Moons, Van Gool, and Vergauwen 2010).

We have already introduced the elements of bundle adjustment in our discussion on iterative pose estimation (Section 6.2.2), i.e., Equations (6.42–6.48) and Figure 6.5. The biggest difference between these formulas and full bundle adjustment is that our feature location measurements $\boldsymbol{x}_{ij}$ now depend not only on the point (track index) i but also on the camera pose index j,

$$\boldsymbol{x}_{ij} = \boldsymbol{f}(\boldsymbol{p}_i, \boldsymbol{R}_j, \boldsymbol{c}_j, \boldsymbol{K}_j), \tag{7.49}$$

and that the 3D point positions $\boldsymbol{p}_i$ are also being simultaneously updated. In addition, it is common to add a stage for radial distortion parameter estimation (2.78),

$$\boldsymbol{f}_{\mathrm{RD}}(\boldsymbol{x}) = (1 + \kappa_1 r^2 + \kappa_2 r^4)\boldsymbol{x}, \tag{7.50}$$

if the cameras being used have not been pre-calibrated, as shown in Figure 7.7.

While most of the boxes (transforms) in Figure 7.7 have previously been explained (6.47), the leftmost box has not. This box performs a robust comparison of the predicted and mea-

[13] The term "bundle" refers to the bundles of rays connecting camera centers to 3D points and the term "adjustment" refers to the iterative minimization of re-projection error. Alternative terms for this in the vision community include *optimal motion estimation* (Weng, Ahuja, and Huang 1993) and *non-linear least squares* (Appendix A.3) (Taylor, Kriegman, and Anandan 1991; Szeliski and Kang 1994).

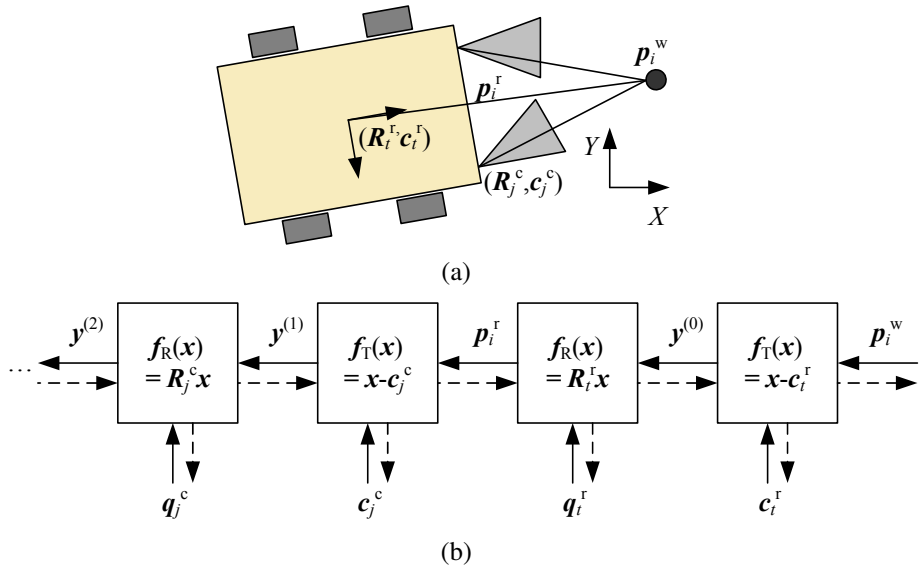


Figure 7.8 A camera rig and its associated transform chain. (a) As the mobile rig (robot) moves around in the world, its pose with respect to the world at time t is captured by $(\boldsymbol{R}_t^{\mathrm{r}}, \boldsymbol{c}_t^{\mathrm{r}})$. Each camera's pose with respect to the rig is captured by $(\boldsymbol{R}_j^{\mathrm{c}}, \boldsymbol{c}_j^{\mathrm{c}})$. (b) A 3D point with world coordinates $\boldsymbol{p}_i^{\mathrm{w}}$ is first transformed into rig coordinates $\boldsymbol{p}_i^{\mathrm{r}}$, and then through the rest of the camera-specific chain, as shown in Figure 7.7.

sured 2D locations $\hat{\boldsymbol{x}}_{ij}$ and $\tilde{\boldsymbol{x}}_{ij}$ after re-scaling by the measurement noise covariance Σ_{ij}. In more detail, this operation can be written as

$$\boldsymbol{r}_{ij} = \tilde{\boldsymbol{x}}_{ij} - \hat{\boldsymbol{x}}_{ij}, \tag{7.51}$$

$$s_{ij}^2 = \boldsymbol{r}_{ij}^T \Sigma_{ij}^{-1} \boldsymbol{r}_{ij}, \tag{7.52}$$

$$e_{ij} = \hat{\rho}(s_{ij}^2), \tag{7.53}$$

where $\hat{\rho}(r^2) = \rho(r)$. The corresponding Jacobians (partial derivatives) can be written as

$$\frac{\partial e_{ij}}{\partial s_{ij}^2} = \hat{\rho}'(s_{ij}^2), \tag{7.54}$$

$$\frac{\partial s_{ij}^2}{\partial \tilde{\boldsymbol{x}}_{ij}} = \Sigma_{ij}^{-1} \boldsymbol{r}_{ij}. \tag{7.55}$$

The advantage of the chained representation introduced above is that it not only makes the computations of the partial derivatives and Jacobians simpler but it can also be adapted to any camera configuration. Consider for example a pair of cameras mounted on a robot that is moving around in the world, as shown in Figure 7.8a. By replacing the rightmost two transformations in Figure 7.7 with the transformations shown in Figure 7.8b, we can simultaneously recover the position of the robot at each time and the calibration of each camera with respect to the rig, in addition to the 3D structure of the world.

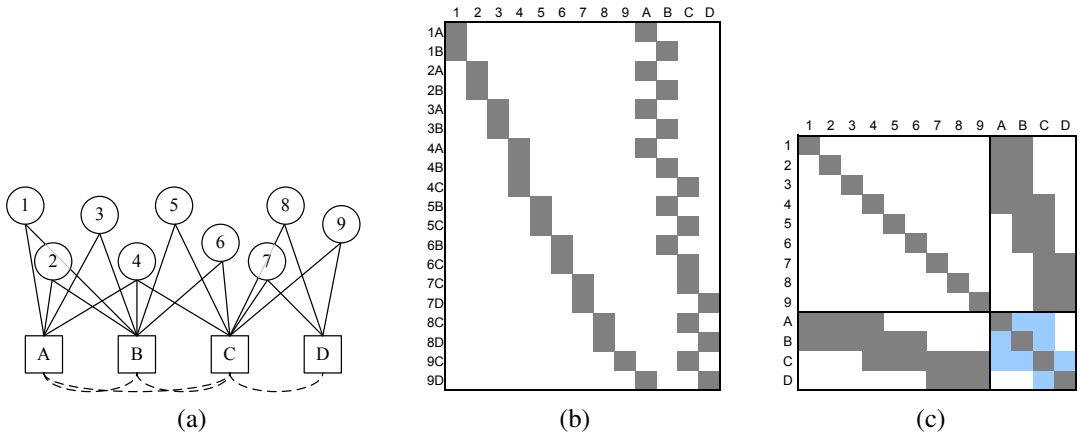


Figure 7.9 (a) Bipartite graph for a toy structure from motion problem and (b) its associated Jacobian $\boldsymbol{J}$ and (c) Hessian $\boldsymbol{A}$. Numbers indicate 3D points and letters indicate cameras. The dashed arcs and light blue squares indicate the fill-in that occurs when the structure (point) variables are eliminated.

7.4.1 Exploiting sparsity

Large bundle adjustment problems, such as those involving reconstructing 3D scenes from thousands of Internet photographs (Snavely, Seitz, and Szeliski 2008b; Agarwal, Snavely, Simon *et al.* 2009; Agarwal, Furukawa, Snavely *et al.* 2010; Snavely, Simon, Goesele *et al.* 2010), can require solving non-linear least squares problems with millions of measurements (feature matches) and tens of thousands of unknown parameters (3D point positions and camera poses). Unless some care is taken, these kinds of problem can become intractable, since the (direct) solution of dense least squares problems is cubic in the number of unknowns.

Fortunately, structure from motion is a *bipartite* problem in structure and motion. Each feature point $\boldsymbol{x}_{ij}$ in a given image depends on one 3D point position $\boldsymbol{p}_i$ and one 3D camera pose $(\boldsymbol{R}_j, \boldsymbol{c}_j)$. This is illustrated in Figure 7.9a, where each circle (1–9) indicates a 3D point, each square (A–D) indicates a camera, and lines (edges) indicate which points are visible in which cameras (2D features). If the values for all the points are known or fixed, the equations for all the cameras become independent, and vice versa.

If we order the structure variables before the motion variables in the Hessian matrix $\boldsymbol{A}$ (and hence also the right hand side vector $\boldsymbol{b}$), we obtain a structure for the Hessian shown in Figure 7.9c.[14] When such a system is solved using sparse Cholesky factorization (see Appendix A.4) (Björck 1996; Golub and Van Loan 1996), the *fill-in* occurs in the smaller motion Hessian A_{cc} (Szeliski and Kang 1994; Triggs, McLauchlan, Hartley *et al.* 1999; Hartley and Zisserman 2004; Lourakis and Argyros 2009; Engels, Stewénius, and Nistér 2006). Some recent papers by (Byröd and øAström 2009), Jeong, Nistér, Steedly *et al.* (2010) and (Agarwal, Snavely, Seitz *et al.* 2010) explore the use of iterative (conjugate gradient) techniques for the solution of bundle adjustment problems.

[14] This ordering is preferable when there are fewer cameras than 3D points, which is the usual case. The exception is when we are tracking a small number of points through many video frames, in which case this ordering should be reversed.

In more detail, the *reduced* motion Hessian is computed using the *Schur complement*,

$$\boldsymbol{A}'_{cc} = \boldsymbol{A}_{cc} - \boldsymbol{A}_{pc}^T \boldsymbol{A}_{pp}^{-1} \boldsymbol{A}_{pc}, \tag{7.56}$$

where $\boldsymbol{A}_{pp}$ is the point (structure) Hessian (the top left block of Figure 7.9c), $\boldsymbol{A}_{pc}$ is the point-camera Hessian (the top right block), and $\boldsymbol{A}_{cc}$ and $\boldsymbol{A}'_{cc}$ are the motion Hessians before and after the point variable elimination (the bottom right block of Figure 7.9c). Notice that $\boldsymbol{A}'_{cc}$ has a non-zero entry between two cameras if they see any 3D point in common. This is indicated with dashed arcs in Figure 7.9a and light blue squares in Figure 7.9c.

Whenever there are global parameters present in the reconstruction algorithm, such as camera intrinsics that are common to all of the cameras, or camera rig calibration parameters such as those shown in Figure 7.8, they should be ordered last (placed along the right and bottom edges of $\boldsymbol{A}$) in order to reduce fill-in.

Engels, Stewénius, and Nistér (2006) provide a nice recipe for sparse bundle adjustment, including all the steps needed to initialize the iterations, as well as typical computation times for a system that uses a fixed number of backward-looking frames in a real-time setting. They also recommend using homogeneous coordinates for the structure parameters $\boldsymbol{p}_i$, which is a good idea, since it avoids numerical instabilities for points near infinity.

Bundle adjustment is now the standard method of choice for most structure-from-motion problems and is commonly applied to problems with hundreds of weakly calibrated images and tens of thousands of points, e.g., in systems such as Photosynth. (Much larger problems are commonly solved in photogrammetry and aerial imagery, but these are usually carefully calibrated and make use of surveyed ground control points.) However, as the problems become larger, it becomes impractical to re-solve full bundle adjustment problems at each iteration.

One approach to dealing with this problem is to use an incremental algorithm, where new cameras are added over time. (This makes particular sense if the data is being acquired from a video camera or moving vehicle (Nistér, Naroditsky, and Bergen 2006; Pollefeys, Nistér, Frahm *et al.* 2008).) A Kalman filter can be used to incrementally update estimates as new information is acquired. Unfortunately, such sequential updating is only statistically optimal for linear least squares problems.

For non-linear problems such as structure from motion, an extended Kalman filter, which linearizes measurement and update equations around the current estimate, needs to be used (Gelb 1974; Viéville and Faugeras 1990). To overcome this limitation, several passes can be made through the data (Azarbayejani and Pentland 1995). Because points disappear from view (and old cameras become irrelevant), a *variable state dimension filter* (VSDF) can be used to adjust the set of state variables over time, for example, by keeping only cameras and point tracks seen in the last k frames (McLauchlan 2000). A more flexible approach to using a fixed number of frames is to propagate corrections backwards through points and cameras until the changes on parameters are below a threshold (Steedly and Essa 2001). Variants of these techniques, including methods that use a fixed window for bundle adjustment (Engels, Stewénius, and Nistér 2006) or select keyframes for doing full bundle adjustment (Klein and Murray 2008) are now commonly used in real-time tracking and augmented-reality applications, as discussed in Section 7.4.2.

When maximum accuracy is required, it is still preferable to perform a full bundle adjustment over all the frames. In order to control the resulting computational complexity, one

approach is to lock together subsets of frames into locally rigid configurations and to optimize the relative positions of these cluster (Steedly, Essa, and Dellaert 2003). A different approach is to select a smaller number of frames to form a *skeletal set* that still spans the whole dataset and produces reconstructions of comparable accuracy (Snavely, Seitz, and Szeliski 2008b). We describe this latter technique in more detail in Section 7.4.4, where we discuss applications of structure from motion to large image sets.

While bundle adjustment and other robust non-linear least squares techniques are the methods of choice for most structure-from-motion problems, they suffer from initialization problems, i.e., they can get stuck in local energy minima if not started sufficiently close to the global optimum. Many systems try to mitigate this by being conservative in what reconstruction they perform early on and which cameras and points they add to the solution (Section 7.4.4). An alternative, however, is to re-formulate the problem using a norm that supports the computation of global optima.

Kahl and Hartley (2008) describe techniques for using L_∞ norms in geometric reconstruction problems. The advantage of such norms is that globally optimal solutions can be efficiently computed using second-order cone programming (SOCP). The disadvantage is that L_∞ norms are particularly sensitive to outliers and so must be combined with good outlier rejection techniques before they can be used.

7.4.2 *Application*: Match move and augmented reality

One of the neatest applications of structure from motion is to estimate the 3D motion of a video or film camera, along with the geometry of a 3D scene, in order to superimpose 3D graphics or computer-generated images (CGI) on the scene. In the visual effects industry, this is known as the *match move* problem (Roble 1999), since the motion of the synthetic 3D camera used to render the graphics must be *matched* to that of the real-world camera. For very small motions, or motions involving pure camera rotations, one or two tracked points can suffice to compute the necessary visual motion. For planar surfaces moving in 3D, four points are needed to compute the homography, which can then be used to insert planar overlays, e.g., to replace the contents of advertising billboards during sporting events.

The general version of this problem requires the estimation of the full 3D camera pose along with the focal length (zoom) of the lens and potentially its radial distortion parameters (Roble 1999). When the 3D structure of the scene is known ahead of time, pose estimation techniques such as *view correlation* (Bogart 1991) or *through-the-lens camera control* (Gleicher and Witkin 1992) can be used, as described in Section 6.2.3.

For more complex scenes, it is usually preferable to recover the 3D structure simultaneously with the camera motion using structure-from-motion techniques. The trick with using such techniques is that in order to prevent any visible jitter between the synthetic graphics and the actual scene, features must be tracked to very high accuracy and ample feature tracks must be available in the vicinity of the insertion location. Some of today's best known match move software packages, such as the *boujou* package from 2d3,[15] which won an Emmy award in 2002, originated in structure-from-motion research in the computer vision community (Fitzgibbon and Zisserman 1998).

[15] http://www.2d3.com/.



(a) (b)

Figure 7.10 3D augmented reality: (a) Darth Vader and a horde of Ewoks battle it out on a table-top recovered using real-time, keyframe-based structure from motion (Klein and Murray 2007) © 2007 IEEE; (b) a virtual teapot is fixed to the top of a real-world coffee cup, whose pose is re-recognized at each time frame (Gordon and Lowe 2006) © 2007 Springer.

Closely related to the match move problem is robotics navigation, where a robot must estimate its location relative to its environment, while simultaneously avoiding any dangerous obstacles. This problem is often known as *simultaneous localization and mapping* (SLAM) (Thrun, Burgard, and Fox 2005) or *visual odometry* (Levin and Szeliski 2004; Nistér, Naroditsky, and Bergen 2006; Maimone, Cheng, and Matthies 2007). Early versions of such algorithms used range-sensing techniques, such as ultrasound, laser range finders, or stereo matching, to estimate local 3D geometry, which could then be fused into a 3D model. Newer techniques can perform the same task based purely on visual feature tracking, sometimes not even requiring a stereo camera rig (Davison, Reid, Molton *et al.* 2007).

Another closely related application is *augmented reality*, where 3D objects are inserted into a video feed in real time, often to annotate or help users understand a scene (Azuma, Baillot, Behringer *et al.* 2001). While traditional systems require prior knowledge about the scene or object being visually tracked (Rosten and Drummond 2005), newer systems can simultaneously build up a model of the 3D environment and then track it, so that graphics can be superimposed.

Klein and Murray (2007) describe a *parallel tracking and mapping* (PTAM) system, which simultaneously applies full bundle adjustment to keyframes selected from a video stream, while performing robust real-time pose estimation on intermediate frames. Figure 7.10a shows an example of their system in use. Once an initial 3D scene has been reconstructed, a dominant plane is estimated (in this case, the table-top) and 3D animated characters are virtually inserted. Klein and Murray (2008) extend their previous system to handle even faster camera motion by adding edge features, which can still be detected even when interest points become too blurred. They also use a direct (intensity-based) rotation estimation algorithm for even faster motions.

Instead of modeling the whole scene as one rigid reference frame, Gordon and Lowe (2006) first build a 3D model of an individual object using feature matching and structure from motion. Once the system has been initialized, for every new frame, they find the object and its pose using a 3D instance recognition algorithm, and then superimpose a graphical

object onto that model, as shown in Figure 7.10b.

While reliably tracking such objects and environments is now a well-solved problem, determining which pixels should be occluded by foreground scene elements still remains an open problem (Chuang, Agarwala, Curless *et al.* 2002; Wang and Cohen 2007a).

7.4.3 Uncertainty and ambiguities

Because structure from motion involves the estimation of so many highly coupled parameters, often with no known "ground truth" components, the estimates produced by structure from motion algorithms can often exhibit large amounts of uncertainty (Szeliski and Kang 1997). An example of this is the classic *bas-relief ambiguity*, which makes it hard to simultaneously estimate the 3D depth of a scene and the amount of camera motion (Oliensis 2005).[16]

As mentioned before, a unique coordinate frame and scale for a reconstructed scene cannot be recovered from monocular visual measurements alone. (When a stereo rig is used, the scale can be recovered if we know the distance (baseline) between the cameras.) This seven-degree-of-freedom *gauge ambiguity* makes it tricky to compute the covariance matrix associated with a 3D reconstruction (Triggs, McLauchlan, Hartley *et al.* 1999; Kanatani and Morris 2001). A simple way to compute a covariance matrix that ignores the gauge freedom (indeterminacy) is to throw away the seven smallest eigenvalues of the information matrix (inverse covariance), whose values are equivalent to the problem Hessian $\boldsymbol{A}$ up to noise scaling (see Section 6.1.4 and Appendix B.6). After we do this, the resulting matrix can be inverted to obtain an estimate of the parameter covariance.

Szeliski and Kang (1997) use this approach to visualize the largest directions of variation in typical structure from motion problems. Not surprisingly, they find that (ignoring the gauge freedoms), the greatest uncertainties for problems such as observing an object from a small number of nearby viewpoints are in the depths of the 3D structure relative to the extent of the camera motion.[17]

It is also possible to estimate *local* or *marginal* uncertainties for individual parameters, which corresponds simply to taking block sub-matrices from the full covariance matrix. Under certain conditions, such as when the camera poses are relatively certain compared to 3D point locations, such uncertainty estimates can be meaningful. However, in many cases, individual uncertainty measures can mask the extent to which reconstruction errors are correlated, which is why looking at the first few modes of greatest joint variation can be helpful.

The other way in which gauge ambiguities affect structure from motion and, in particular, bundle adjustment is that they make the system Hessian matrix $\boldsymbol{A}$ rank-deficient and hence impossible to invert. A number of techniques have been proposed to mitigate this problem (Triggs, McLauchlan, Hartley *et al.* 1999; Bartoli 2003). In practice, however, it appears that simply adding a small amount of the Hessian diagonal $\lambda \text{diag}(\boldsymbol{A})$ to the Hessian $\boldsymbol{A}$ itself, as is done in the Levenberg–Marquardt non-linear least squares algorithm (Appendix A.3), usually works well.

[16] Bas-relief refers to a kind of sculpture in which objects, often on ornamental friezes, are sculpted with less depth than they actually occupy. When lit from above by sunlight, they appear to have true 3D depth because of the ambiguity between relative depth and the angle of the illuminant (Section 12.1.1).

[17] A good way to minimize the amount of such ambiguities is to use wide field of view cameras (Antone and Teller 2002; Levin and Szeliski 2006).

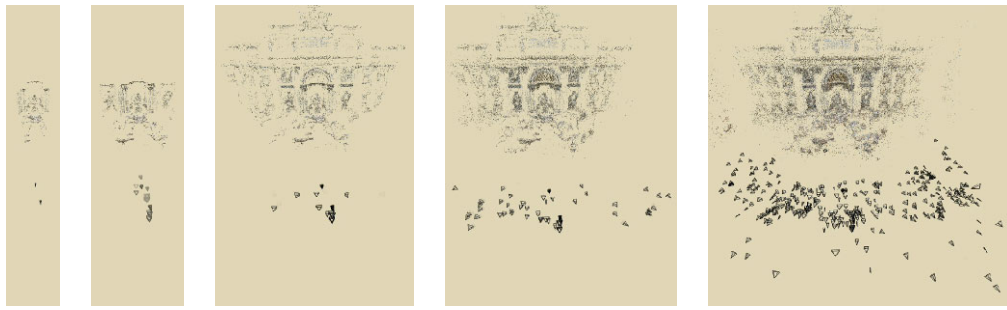


Figure 7.11 Incremental structure from motion (Snavely, Seitz, and Szeliski 2006) © 2006 ACM: Starting with an initial two-frame reconstruction of Trevi Fountain, batches of images are added using pose estimation, and their positions (along with the 3D model) are refined using bundle adjustment.

7.4.4 *Application*: Reconstruction from Internet photos

The most widely used application of structure from motion is in the reconstruction of 3D objects and scenes from video sequences and collections of images (Pollefeys and Van Gool 2002). The last decade has seen an explosion of techniques for performing this task automatically without the need for any manual correspondence or pre-surveyed ground control points. A lot of these techniques assume that the scene is taken with the same camera and hence the images all have the same intrinsics (Fitzgibbon and Zisserman 1998; Koch, Pollefeys, and Van Gool 2000; Schaffalitzky and Zisserman 2002; Tuytelaars and Van Gool 2004; Pollefeys, Nistér, Frahm *et al.* 2008; Moons, Van Gool, and Vergauwen 2010). Many of these techniques take the results of the sparse feature matching and structure from motion computation and then compute dense 3D surface models using multi-view stereo techniques (Section 11.6) (Koch, Pollefeys, and Van Gool 2000; Pollefeys and Van Gool 2002; Pollefeys, Nistér, Frahm *et al.* 2008; Moons, Van Gool, and Vergauwen 2010).

The latest innovation in this space has been the application of structure from motion and multi-view stereo techniques to thousands of images taken from the Internet, where very little is known about the cameras taking the photographs (Snavely, Seitz, and Szeliski 2008a). Before the structure from motion computation can begin, it is first necessary to establish sparse correspondences between different pairs of images and to then link such correspondences into *feature tracks*, which associate individual 2D image features with global 3D points. Because the $O(N^2)$ comparison of all pairs of images can be very slow, a number of techniques have been developed in the recognition community to make this process faster (Section 14.3.2) (Nistér and Stewénius 2006; Philbin, Chum, Sivic *et al.* 2008; Li, Wu, Zach *et al.* 2008; Chum, Philbin, and Zisserman 2008; Chum and Matas 2010).

To begin the reconstruction process, it is important to to select a good pair of images, where there are both a large number of consistent matches (to lower the likelihood of incorrect correspondences) and a significant amount of out-of-plane parallax,[18] to ensure that a stable reconstruction can be obtained (Snavely, Seitz, and Szeliski 2006). The EXIF tags associated with the photographs can be used to get good initial estimates for camera focal lengths, although this is not always strictly necessary, since these parameters are re-adjusted

[18] A simple way to compute this is to robustly fit a homography to the correspondences and measure reprojection errors.



(a) (b) (c)

Figure 7.12 3D reconstructions produced by the incremental structure from motion algorithm developed by Snavely, Seitz, and Szeliski (2006) © 2006 ACM: (a) cameras and point cloud from Trafalgar Square; (b) cameras and points overlaid on an image from the Great Wall of China; (c) overhead view of a reconstruction of the Old Town Square in Prague registered to an aerial photograph.

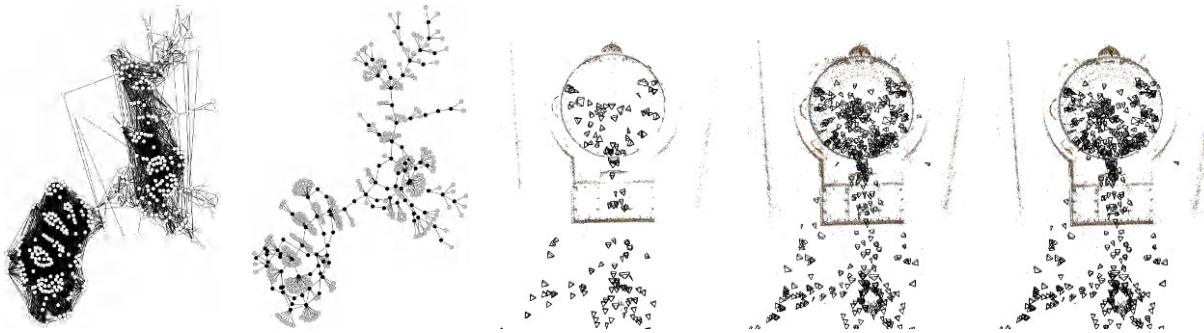
as part of the bundle adjustment process.

Once an initial pair has been reconstructed, the pose of cameras that see a sufficient number of the resulting 3D points can be estimated (Section 6.2) and the complete set of cameras and feature correspondences can be used to perform another round of bundle adjustment. Figure 7.11 shows the progression of the incremental bundle adjustment algorithm, where sets of cameras are added after each successive round of bundle adjustment, while Figure 7.12 shows some additional results. An alternative to this kind of *seed and grow* approach is to first reconstruct triplets of images and then hierarchically merge triplets into larger collections, as described by Fitzgibbon and Zisserman (1998).

Unfortunately, as the incremental structure from motion algorithm continues to add more cameras and points, it can become extremely slow. The direct solution of a dense system of $O(N)$ equations for the camera pose updates can take $O(N^3)$ time; while structure from motion problems are rarely dense, scenes such as city squares have a high percentage of cameras that see points in common. Re-running the bundle adjustment algorithm after every few camera additions results in a quartic scaling of the run time with the number of images in the dataset. One approach to solving this problem is to select a smaller number of images for the original scene reconstruction and to fold in the remaining images at the very end.

Snavely, Seitz, and Szeliski (2008b) develop an algorithm for computing such a *skeletal set* of images, which is guaranteed to produce a reconstruction whose error is within a bounded factor of the optimal reconstruction accuracy. Their algorithm first evaluates all pairwise uncertainties (position covariances) between overlapping images and then chains them together to estimate a lower bound for the relative uncertainty of any distant pair. The skeletal set is constructed so that the maximal uncertainty between any pair grows by no more than a constant factor. Figure 7.13 shows an example of the skeletal set computed for 784 images of the Pantheon in Rome. As you can see, even though the skeletal set contains just a fraction of the original images, the shapes of the skeletal set and full bundle adjusted reconstructions are virtually indistinguishable.

The ability to automatically reconstruct 3D models from large, unstructured image collections has opened a wide variety of additional applications, including the ability to automat-



(a) (b) (c) (d) (e)

Figure 7.13 Large scale structure from motion using skeletal sets (Snavely, Seitz, and Szeliski 2008b) © 2008 IEEE: (a) original match graph for 784 images; (b) skeletal set containing 101 images; (c) top-down view of scene (Pantheon) reconstructed from the skeletal set; (d) reconstruction after adding in the remaining images using pose estimation; (e) final bundle adjusted reconstruction, which is almost identical.

ically find and label locations and regions of interest (Simon, Snavely, and Seitz 2007; Simon and Seitz 2008; Gammeter, Bossard, Quack *et al.* 2009) and to cluster large image collections so that they can be automatically labeled (Li, Wu, Zach *et al.* 2008; Quack, Leibe, and Van Gool 2008). Some of these application are discussed in more detail in Section 13.1.2.

7.5 Constrained structure and motion

The most general algorithms for structure from motion make no prior assumptions about the objects or scenes that they are reconstructing. In many cases, however, the scene contains higher-level geometric primitives, such as lines and planes. These can provide information complementary to interest points and also serve as useful building blocks for 3D modeling and visualization. Furthermore, these primitives are often arranged in particular relationships, i.e., many lines and planes are either parallel or orthogonal to each other. This is particularly true of architectural scenes and models, which we study in more detail in Section 12.6.1.

Sometimes, instead of exploiting regularity in the scene structure, it is possible to take advantage of a constrained motion model. For example, if the object of interest is rotating on a turntable (Szeliski 1991b), i.e., around a fixed but unknown axis, specialized techniques can be used to recover this motion (Fitzgibbon, Cross, and Zisserman 1998). In other situations, the camera itself may be moving in a fixed arc around some center of rotation (Shum and He 1999). Specialized capture setups, such as mobile stereo camera rigs or moving vehicles equipped with multiple fixed cameras, can also take advantage of the knowledge that individual cameras are (mostly) fixed with respect to the capture rig, as shown in Figure 7.8.[19]

[19] Because of mechanical compliance and jitter, it may be prudent to allow for a small amount of individual camera rotation around a nominal position.

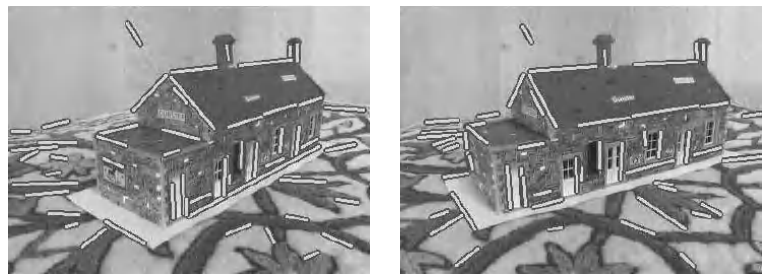


Figure 7.14 Two images of a toy house along with their matched 3D line segments (Schmid and Zisserman 1997) © 1997 Springer.

7.5.1 Line-based techniques

It is well known that pairwise epipolar geometry cannot be recovered from line matches alone, even if the cameras are calibrated. To see this, think of projecting the set of lines in each image into a set of 3D planes in space. You can move the two cameras around into any configuration you like and still obtain a valid reconstruction for 3D lines.

When lines are visible in three or more views, the trifocal tensor can be used to transfer lines from one pair of images to another (Hartley and Zisserman 2004). The trifocal tensor can also be computed on the basis of line matches alone.

Schmid and Zisserman (1997) describe a widely used technique for matching 2D lines based on the average of 15×15 pixel correlation scores evaluated at all pixels along their common line segment intersection.[20] In their system, the epipolar geometry is assumed to be known, e.g., computed from point matches. For wide baselines, all possible homographies corresponding to planes passing through the 3D line are used to warp pixels and the maximum correlation score is used. For triplets of images, the trifocal tensor is used to verify that the lines are in geometric correspondence before evaluating the correlations between line segments. Figure 7.14 shows the results of using their system.

Bartoli and Sturm (2003) describe a complete system for extending three view relations (trifocal tensors) computed from manual line correspondences to a full bundle adjustment of all the line and camera parameters. The key to their approach is to use the Plücker coordinates (2.12) to parameterize lines and to directly minimize reprojection errors. It is also possible to represent 3D line segments by their endpoints and to measure either the reprojection error perpendicular to the detected 2D line segments in each image or the 2D errors using an elongated uncertainty ellipse aligned with the line segment direction (Szeliski and Kang 1994).

Instead of reconstructing 3D lines, Bay, Ferrari, and Van Gool (2005) use RANSAC to group lines into likely coplanar subsets. Four lines are chosen at random to compute a homography, which is then verified for these and other plausible line segment matches by evaluating color histogram-based correlation scores. The 2D intersection points of lines belonging to the same plane are then used as virtual measurements to estimate the epipolar geometry, which is more accurate than using the homographies directly.

[20] Because lines often occur at depth or orientation discontinuities, it may be preferable to compute correlation scores (or to match color histograms (Bay, Ferrari, and Van Gool 2005)) separately on each side of the line.

An alternative to grouping lines into coplanar subsets is to group lines by parallelism. Whenever three or more 2D lines share a common vanishing point, there is a good likelihood that they are parallel in 3D. By finding multiple vanishing points in an image (Section 4.3.3) and establishing correspondences between such vanishing points in different images, the relative rotations between the various images (and often the camera intrinsics) can be directly estimated (Section 6.3.2).

Shum, Han, and Szeliski (1998) describe a 3D modeling system which first constructs calibrated panoramas from multiple images (Section 7.4) and then has the user draw vertical and horizontal lines in the image to demarcate the boundaries of planar regions. The lines are initially used to establish an absolute rotation for each panorama and are later used (along with the inferred vertices and planes) to infer a 3D structure, which can be recovered up to scale from one or more images (Figure 12.15).

A fully automated approach to line-based structure from motion is presented vy Werner and Zisserman (2002). In their system, they first find lines and group them by common vanishing points in each image (Section 4.3.3). The vanishing points are then used to calibrate the camera, i.e., to performa a "metric upgrade" (Section 6.3.2). Lines corresponding to common vanishing points are then matched using both appearance (Schmid and Zisserman 1997) and trifocal tensors. The resulting set of 3D lines, color coded by common vanishing directions (3D orientations) is shown in Figure 12.16a. These lines are then used to infer planes and a block-structured model for the scene, as described in more detail in Section 12.6.1.

7.5.2 Plane-based techniques

In scenes that are rich in planar structures, e.g., in architecture and certain kinds of manufactured objects such as furniture, it is possible to directly estimate homographies between different planes, using either feature-based or intensity-based methods. In principle, this information can be used to simultaneously infer the camera poses and the plane equations, i.e., to compute plane-based structure from motion.

Luong and Faugeras (1996) show how a fundamental matrix can be directly computed from two or more homographies using algebraic manipulations and least squares. Unfortunately, this approach often performs poorly, since the algebraic errors do not correspond to meaningful reprojection errors (Szeliski and Torr 1998).

A better approach is to *hallucinate* virtual point correspondences within the areas from which each homography was computed and to feed them into a standard structure from motion algorithm (Szeliski and Torr 1998). An even better approach is to use full bundle adjustment with explicit plane equations, as well as additional constraints to force reconstructed co-planar features to lie exactly on their corresponding planes. (A principled way to do this is to establish a coordinate frame for each plane, e.g., at one of the feature points, and to use 2D in-plane parameterizations for the other points.) The system developed by Shum, Han, and Szeliski (1998) shows an example of such an approach, where the directions of lines and normals for planes in the scene are pre-specified by the user.

7.6 Additional reading

The topic of structure from motion is extensively covered in books and review articles on multi-view geometry (Faugeras and Luong 2001; Hartley and Zisserman 2004; Moons, Van Gool, and Vergauwen 2010). For two-frame reconstruction, Hartley (1997a) wrote a highly cited paper on the "eight-point algorithm" for computing an essential or fundamental matrix with reasonable point normalization. When the cameras are calibrated, the five-point algorithm of Nistér (2004) can be used in conjunction with RANSAC to obtain initial reconstructions from the minimum number of points. When the cameras are uncalibrated, various self-calibration techniques can be found in work by Hartley and Zisserman (2004); Moons, Van Gool, and Vergauwen (2010)—I only briefly mention one of the simplest techniques, the Kruppa equations (7.35).

In applications where points are being tracked from frame to frame, factorization techniques, based on either orthographic camera models (Tomasi and Kanade 1992; Poelman and Kanade 1997; Costeira and Kanade 1995; Morita and Kanade 1997; Morris and Kanade 1998; Anandan and Irani 2002) or projective extensions (Christy and Horaud 1996; Sturm and Triggs 1996; Triggs 1996; Oliensis and Hartley 2007), can be used.

Triggs, McLauchlan, Hartley *et al.* (1999) provide a good tutorial and survey on bundle adjustment, while Lourakis and Argyros (2009) and Engels, Stewénius, and Nistér (2006) provide tips on implementation and effective practices. Bundle adjustment is also covered in textbooks and surveys on multi-view geometry (Faugeras and Luong 2001; Hartley and Zisserman 2004; Moons, Van Gool, and Vergauwen 2010). Techniques for handling larger problems are described by Snavely, Seitz, and Szeliski (2008b); Agarwal, Snavely, Simon *et al.* (2009); Jeong, Nistér, Steedly *et al.* (2010); Agarwal, Snavely, Seitz *et al.* (2010). While bundle adjustment is often called as an inner loop inside incremental reconstruction algorithms (Snavely, Seitz, and Szeliski 2006), hierarchical (Fitzgibbon and Zisserman 1998; Farenzena, Fusiello, and Gherardi 2009) and global (Rother and Carlsson 2002; Martinec and Pajdla 2007) approaches for initialization are also possible and perhaps even preferable.

As structure from motion starts being applied to dynamic scenes, the topic of non-rigid structure from motion (Torresani, Hertzmann, and Bregler 2008), which we do not cover in this book, will become more important.

7.7 Exercises

Ex 7.1: Triangulation Use the calibration pattern you built and tested in Exercise 6.7 to test your triangulation accuracy. As an alternative, generate synthetic 3D points and cameras and add noise to the 2D point measurements.

1. Assume that you know the camera pose, i.e., the camera matrices. Use the 3D distance to rays (7.4) or linearized versions of Equations (7.5–7.6) to compute an initial set of 3D locations. Compare these to your known ground truth locations.

2. Use iterative non-linear minimization to improve your initial estimates and report on the improvement in accuracy.

3. (Optional) Use the technique described by Hartley and Sturm (1997) to perform two-frame triangulation.

4. See if any of the failure modes reported by Hartley and Sturm (1997) or Hartley (1998) occur in practice.

Ex 7.2: Essential and fundamental matrix Implement the two-frame $\boldsymbol{E}$ and $\boldsymbol{F}$ matrix estimation techniques presented in Section 7.2, with suitable re-scaling for better noise immunity.

1. Use the data from Exercise 7.1 to validate your algorithms and to report on their accuracy.
2. (Optional) Implement one of the improved $\boldsymbol{F}$ or $\boldsymbol{E}$ estimation algorithms, e.g., using renormalization (Zhang 1998b; Torr and Fitzgibbon 2004; Hartley and Zisserman 2004), RANSAC (Torr and Murray 1997), least media squares (LMS), or the five-point algorithm developed by Nistér (2004).

Ex 7.3: View morphing and interpolation Implement automatic view morphing, i.e., compute two-frame structure from motion and then use these results to generate a smooth animation from one image to the next (Section 7.2.3).

1. Decide how to represent your 3D scene, e.g., compute a Delaunay triangulation of the matched point and decide what to do with the triangles near the border. (Hint: try fitting a plane to the scene, e.g., behind most of the points.)
2. Compute your in-between camera positions and orientations.
3. Warp each triangle to its new location, preferably using the correct perspective projection (Szeliski and Shum 1997).
4. (Optional) If you have a denser 3D model (e.g., from stereo), decide what to do at the "cracks".
5. (Optional) For a non-rigid scene, e.g., two pictures of a face with different expressions, not all of your matched points will obey the epipolar geometry. Decide how to handle them to achieve the best effect.

Ex 7.4: Factorization Implement the factorization algorithm described in Section 7.3 using point tracks you computed in Exercise 4.5.

1. (Optional) Implement uncertainty rescaling (Anandan and Irani 2002) and comment on whether this improves your results.
2. (Optional) Implement one of the perspective improvements to factorization discussed in Section 7.3.1 (Christy and Horaud 1996; Sturm and Triggs 1996; Triggs 1996). Does this produce significantly lower reprojection errors? Can you upgrade this reconstruction to a metric one?

Ex 7.5: Bundle adjuster Implement a full bundle adjuster. This may sound daunting, but it really is not.

1. Devise the internal data structures and external file representations to hold your camera parameters (position, orientation, and focal length), 3D point locations (Euclidean or homogeneous), and 2D point tracks (frame and point identifier as well as 2D locations).

2. Use some other technique, such as factorization, to initialize the 3D point and camera locations from your 2D tracks (e.g., a subset of points that appears in all frames).

3. Implement the code corresponding to the forward transformations in Figure 7.7, i.e., for each 2D point measurement, take the corresponding 3D point, map it through the camera transformations (including perspective projection and focal length scaling), and compare it to the 2D point measurement to get a residual error.

4. Take the residual error and compute its derivatives with respect to all the unknown motion and structure parameters, using backward chaining, as shown, e.g., in Figure 7.7 and Equation (6.47). This gives you the sparse Jacobian $\boldsymbol{J}$ used in Equations (6.13–6.17) and Equation (6.43).

5. Use a sparse least squares or linear system solver, e.g., MATLAB, SparseSuite, or SPARSKIT (see Appendix A.4 and A.5), to solve the corresponding linearized system, adding a small amount of diagonal preconditioning, as in Levenberg–Marquardt.

6. Update your parameters, make sure your rotation matrices are still orthonormal (e.g., by re-computing them from your quaternions), and continue iterating while monitoring your residual error.

7. (Optional) Use the "Schur complement trick" (7.56) to reduce the size of the system being solved (Triggs, McLauchlan, Hartley *et al.* 1999; Hartley and Zisserman 2004; Lourakis and Argyros 2009; Engels, Stewénius, and Nistér 2006).

8. (Optional) Implement your own iterative sparse solver, e.g., conjugate gradient, and compare its performance to a direct method.

9. (Optional) Make your bundle adjuster robust to outliers, or try adding some of the other improvements discussed in (Engels, Stewénius, and Nistér 2006). Can you think of any other ways to make your algorithm even faster or more robust?

Ex 7.6: Match move and augmented reality Use the results of the previous exercise to superimpose a rendered 3D model on top of video. See Section 7.4.2 for more details and ideas. Check for how "locked down" the objects are.

Ex 7.7: Line-based reconstruction Augment the previously developed bundle adjuster to include lines, possibly with known 3D orientations.

Optionally, use co-planar sets of points and lines to hypothesize planes and to enforce co-planarity (Schaffalitzky and Zisserman 2002; Robertson and Cipolla 2002)

Ex 7.8: Flexible bundle adjuster Design a bundle adjuster that allows for arbitrary chains of transformations and prior knowledge about the unknowns, as suggested in Figures 7.7–7.8.

Ex 7.9: Unordered image matching Compute the camera pose and 3D structure of a scene from an arbitrary collection of photographs (Brown and Lowe 2003; Snavely, Seitz, and Szeliski 2006).

Chapter 8

Dense motion estimation

8.1 Translational alignment 337
8.1.1 Hierarchical motion estimation 341
8.1.2 Fourier-based alignment 341
8.1.3 Incremental refinement 345
8.2 Parametric motion 350
8.2.1 *Application*: Video stabilization 354
8.2.2 Learned motion models 354
8.3 Spline-based motion 355
8.3.1 *Application*: Medical image registration 358
8.4 Optical flow 360
8.4.1 Multi-frame motion estimation 363
8.4.2 *Application*: Video denoising 364
8.4.3 *Application*: De-interlacing 364
8.5 Layered motion 365
8.5.1 *Application*: Frame interpolation 368
8.5.2 Transparent layers and reflections 368
8.6 Additional reading 370
8.7 Exercises 371

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_8,

© Springer-Verlag London Limited 2011

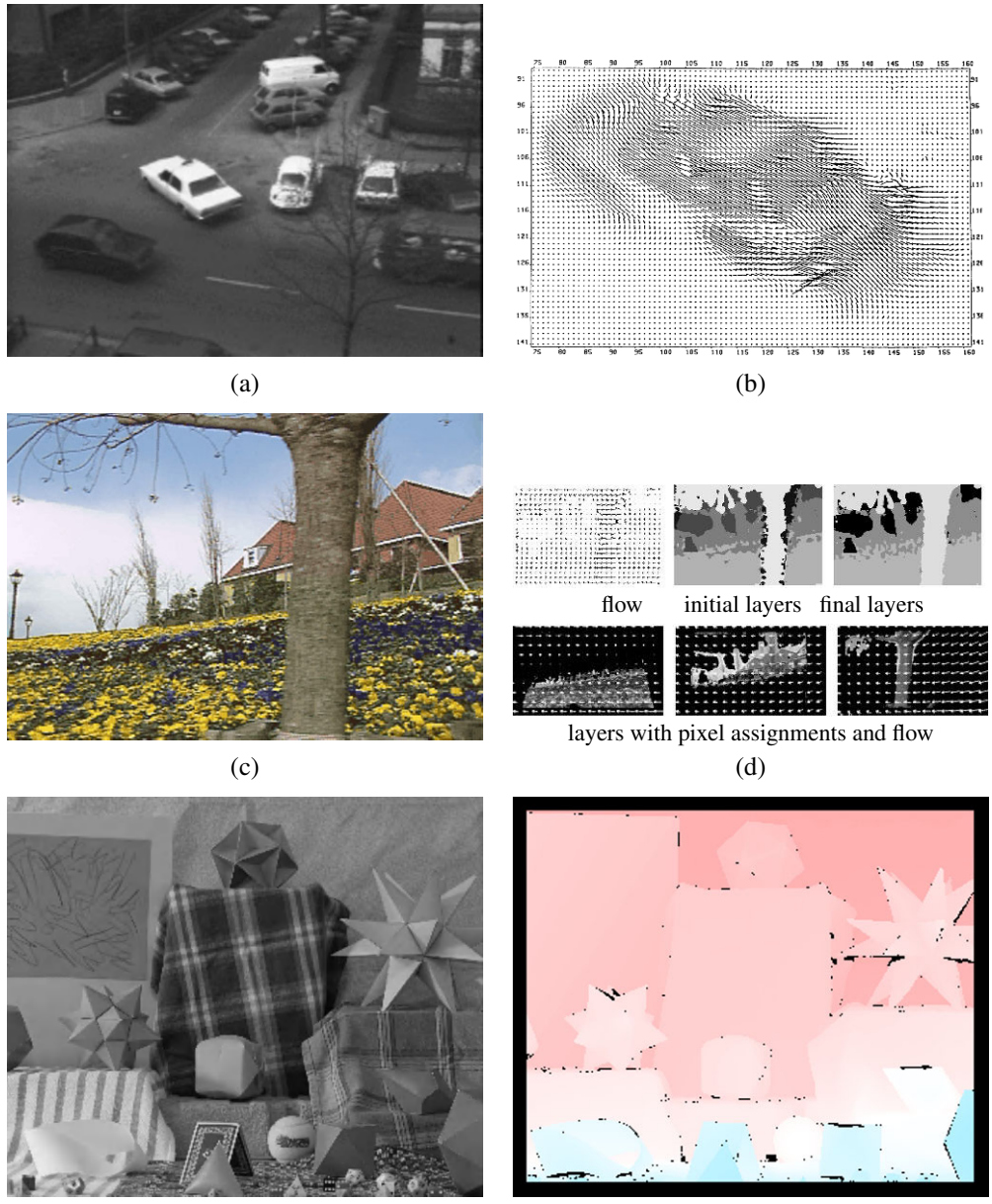


Figure 8.1 Motion estimation: (a–b) regularization-based optical flow (Nagel and Enkelmann 1986) © 1986 IEEE; (c–d) layered motion estimation (Wang and Adelson 1994) © 1994 IEEE; (e–f) sample image and ground truth flow from evaluation database (Baker, Black, Lewis *et al.* 2007) © 2007 IEEE.

Algorithms for aligning images and estimating motion in video sequences are among the most widely used in computer vision. For example, frame-rate image alignment is widely used in camcorders and digital cameras to implement their image stabilization (IS) feature.

An early example of a widely used image registration algorithm is the patch-based translational alignment (optical flow) technique developed by Lucas and Kanade (1981). Variants of this algorithm are used in almost all motion-compensated video compression schemes such as MPEG and H.263 (Le Gall 1991). Similar parametric motion estimation algorithms have found a wide variety of applications, including video summarization (Teodosio and Bender 1993; Irani and Anandan 1998), video stabilization (Hansen, Anandan, Dana *et al.* 1994; Srinivasan, Chellappa, Veeraraghavan *et al.* 2005; Matsushita, Ofek, Ge *et al.* 2006), and video compression (Irani, Hsu, and Anandan 1995; Lee, ge Chen, lung Bruce Lin *et al.* 1997). More sophisticated image registration algorithms have also been developed for medical imaging and remote sensing. Image registration techniques are surveyed by Brown (1992), Zitov'aa and Flusser (2003), Goshtasby (2005), and Szeliski (2006a).

To estimate the motion between two or more images, a suitable *error metric* must first be chosen to compare the images (Section 8.1). Once this has been established, a suitable *search* technique must be devised. The simplest technique is to exhaustively try all possible alignments, i.e., to do a *full search*. In practice, this may be too slow, so *hierarchical* coarse-to-fine techniques (Section 8.1.1) based on image pyramids are normally used. Alternatively, Fourier transforms (Section 8.1.2) can be used to speed up the computation.

To get sub-pixel precision in the alignment, *incremental* methods (Section 8.1.3) based on a Taylor series expansion of the image function are often used. These can also be applied to *parametric motion models* (Section 8.2), which model global image transformations such as rotation or shearing. Motion estimation can be made more reliable by *learning* the typical dynamics or motion statistics of the scenes or objects being tracked, e.g., the natural gait of walking people (Section 8.2.2). For more complex motions, piecewise parametric *spline motion models* (Section 8.3) can be used. In the presence of multiple independent (and perhaps non-rigid) motions, general-purpose *optical flow* (or *optic flow*) techniques need to be used (Section 8.4). For even more complex motions that include a lot of occlusions, *layered motion models* (Section 8.5), which decompose the scene into coherently moving layers, can work well.

In this chapter, we describe each of these techniques in more detail. Additional details can be found in review and comparative evaluation papers on motion estimation (Barron, Fleet, and Beauchemin 1994; Mitiche and Bouthemy 1996; Stiller and Konrad 1999; Szeliski 2006a; Baker, Black, Lewis *et al.* 2007).

8.1 Translational alignment

The simplest way to establish an alignment between two images or image patches is to shift one image relative to the other. Given a *template* image $I_0(\boldsymbol{x})$ sampled at discrete pixel locations $\{\boldsymbol{x}_i = (x_i, y_i)\}$, we wish to find where it is located in image $I_1(\boldsymbol{x})$. A least squares solution to this problem is to find the minimum of the *sum of squared differences* (SSD) function

$$E_{\text{SSD}}(\boldsymbol{u}) = \sum_i [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2 = \sum_i e_i^2, \tag{8.1}$$

where $\boldsymbol{u} = (u, v)$ is the *displacement* and $e_i = I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)$ is called the *residual error* (or the *displaced frame difference* in the video coding literature).[1] (We ignore for the moment the possibility that parts of I_0 may lie outside the boundaries of I_1 or be otherwise not visible.) The assumption that corresponding pixel values remain the same in the two images is often called the *brightness constancy constraint*.[2]

In general, the displacement $\boldsymbol{u}$ can be fractional, so a suitable interpolation function must be applied to image $I_1(\boldsymbol{x})$. In practice, a bilinear interpolant is often used but bicubic interpolation can yield slightly better results (Szeliski and Scharstein 2004). Color images can be processed by summing differences across all three color channels, although it is also possible to first transform the images into a different color space or to only use the luminance (which is often done in video encoders).

Robust error metrics. We can make the above error metric more robust to outliers by replacing the squared error terms with a robust function $\rho(e_i)$ (Huber 1981; Hampel, Ronchetti, Rousseeuw *et al.* 1986; Black and Anandan 1996; Stewart 1999) to obtain

$$E_{\mathrm{SRD}}(\boldsymbol{u}) = \sum_i \rho(I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)) = \sum_i \rho(e_i). \tag{8.2}$$

The robust norm $\rho(e)$ is a function that grows less quickly than the quadratic penalty associated with least squares. One such function, sometimes used in motion estimation for video coding because of its speed, is the *sum of absolute differences* (SAD) metric[3] or L_1 norm, i.e.,

$$E_{\mathrm{SAD}}(\boldsymbol{u}) = \sum_i |I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)| = \sum_i |e_i|. \tag{8.3}$$

However, since this function is not differentiable at the origin, it is not well suited to gradient-descent approaches such as the ones presented in Section 8.1.3.

Instead, a smoothly varying function that is quadratic for small values but grows more slowly away from the origin is often used. Black and Rangarajan (1996) discuss a variety of such functions, including the *Geman–McClure* function,

$$\rho_{\mathrm{GM}}(x) = \frac{x^2}{1 + x^2/a^2}, \tag{8.4}$$

where a is a constant that can be thought of as an *outlier threshold*. An appropriate value for the threshold can itself be derived using robust statistics (Huber 1981; Hampel, Ronchetti, Rousseeuw *et al.* 1986; Rousseeuw and Leroy 1987), e.g., by computing the *median absolute deviation*, $MAD = \mathrm{med}_i|e_i|$, and multiplying it by 1.4 to obtain a robust estimate of the standard deviation of the inlier noise process (Stewart 1999).

[1] The usual justification for using least squares is that it is the optimal estimate with respect to Gaussian noise. See the discussion below on robust error metrics as well as Appendix B.3.

[2] Brightness constancy (Horn 1974) is the tendency for objects to maintain their perceived brightness under varying illumination conditions.

[3] In video compression, e.g., the H.264 standard (http://www.itu.int/rec/T-REC-H.264), the sum of absolute transformed differences (SATD), which measures the differences in a frequency transform space, e.g., using a Hadamard transform, is often used since it more accurately predicts quality (Richardson 2003).

Spatially varying weights. The error metrics above ignore that fact that for a given alignment, some of the pixels being compared may lie outside the original image boundaries. Furthermore, we may want to partially or completely downweight the contributions of certain pixels. For example, we may want to selectively "erase" some parts of an image from consideration when stitching a mosaic where unwanted foreground objects have been cut out. For applications such as background stabilization, we may want to downweight the middle part of the image, which often contains independently moving objects being tracked by the camera.

All of these tasks can be accomplished by associating a spatially varying per-pixel weight value with each of the two images being matched. The error metric then becomes the weighted (or *windowed*) SSD function,

$$E_{\mathrm{WSSD}}(\boldsymbol{u}) = \sum_i w_0(\boldsymbol{x}_i) w_1(\boldsymbol{x}_i + \boldsymbol{u}) [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2, \tag{8.5}$$

where the weighting functions w_0 and w_1 are zero outside the image boundaries.

If a large range of potential motions is allowed, the above metric can have a bias towards smaller overlap solutions. To counteract this bias, the windowed SSD score can be divided by the overlap area

$$A = \sum_i w_0(\boldsymbol{x}_i) w_1(\boldsymbol{x}_i + \boldsymbol{u}) \tag{8.6}$$

to compute a *per-pixel* (or mean) squared pixel error E_{WSSD}/A. The square root of this quantity is the *root mean square* intensity error

$$RMS = \sqrt{E_{\mathrm{WSSD}}/A} \tag{8.7}$$

often reported in comparative studies.

Bias and gain (exposure differences). Often, the two images being aligned were not taken with the same exposure. A simple model of linear (affine) intensity variation between the two images is the *bias and gain* model,

$$I_1(\boldsymbol{x} + \boldsymbol{u}) = (1 + \alpha) I_0(\boldsymbol{x}) + \beta, \tag{8.8}$$

where β is the *bias* and α is the *gain* (Lucas and Kanade 1981; Gennert 1988; Fuh and Maragos 1991; Baker, Gross, and Matthews 2003; Evangelidis and Psarakis 2008). The least squares formulation then becomes

$$E_{\mathrm{BG}}(\boldsymbol{u}) = \sum_i [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - (1 + \alpha) I_0(\boldsymbol{x}_i) - \beta]^2 = \sum_i [\alpha I_0(\boldsymbol{x}_i) + \beta - e_i]^2. \tag{8.9}$$

Rather than taking a simple squared difference between corresponding patches, it becomes necessary to perform a *linear regression* (Appendix A.2), which is somewhat more costly. Note that for color images, it may be necessary to estimate a different bias and gain for each color channel to compensate for the automatic *color correction* performed by some digital cameras (Section 2.3.2). Bias and gain compensation is also used in video codecs, where it is known as *weighted prediction* (Richardson 2003).

A more general (spatially varying, non-parametric) model of intensity variation, which is computed as part of the registration process, is used in (Negahdaripour 1998; Jia and Tang

2003; Seitz and Baker 2009). This can be useful for dealing with local variations such as the *vignetting* caused by wide-angle lenses, wide apertures, or lens housings. It is also possible to pre-process the images before comparing their values, e.g., using band-pass filtered images (Anandan 1989; Bergen, Anandan, Hanna *et al.* 1992), gradients (Scharstein 1994; Papenberg, Bruhn, Brox *et al.* 2006), or using other local transformations such as histograms or rank transforms (Cox, Roy, and Hingorani 1995; Zabih and Woodfill 1994), or to maximize *mutual information* (Viola and Wells III 1997; Kim, Kolmogorov, and Zabih 2003). Hirschmüller and Scharstein (2009) compare a number of these approaches and report on their relative performance in scenes with exposure differences.

Correlation. An alternative to taking intensity differences is to perform *correlation*, i.e., to maximize the *product* (or *cross-correlation*) of the two aligned images,

$$E_{\mathrm{CC}}(\boldsymbol{u}) = \sum_i I_0(\boldsymbol{x}_i) I_1(\boldsymbol{x}_i + \boldsymbol{u}). \tag{8.10}$$

At first glance, this may appear to make bias and gain modeling unnecessary, since the images will prefer to line up regardless of their relative scales and offsets. However, this is actually not true. If a very bright patch exists in $I_1(\boldsymbol{x})$, the maximum product may actually lie in that area.

For this reason, *normalized cross-correlation* is more commonly used,

$$E_{\mathrm{NCC}}(\boldsymbol{u}) = \frac{\sum_i [I_0(\boldsymbol{x}_i) - \overline{I_0}]\,[I_1(\boldsymbol{x}_i + \boldsymbol{u}) - \overline{I_1}]}{\sqrt{\sum_i [I_0(\boldsymbol{x}_i) - \overline{I_0}]^2}\sqrt{\sum_i [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - \overline{I_1}]^2}}, \tag{8.11}$$

where

$$\overline{I_0} = \frac{1}{N}\sum_i I_0(\boldsymbol{x}_i) \quad \text{and} \tag{8.12}$$

$$\overline{I_1} = \frac{1}{N}\sum_i I_1(\boldsymbol{x}_i + \boldsymbol{u}) \tag{8.13}$$

are the *mean images* of the corresponding patches and N is the number of pixels in the patch. The normalized cross-correlation score is always guaranteed to be in the range $[-1, 1]$, which makes it easier to handle in some higher-level applications, such as deciding which patches truly match. Normalized correlation works well when matching images taken with different exposures, e.g., when creating high dynamic range images (Section 10.2). Note, however, that the NCC score is undefined if either of the two patches has zero variance (and, in fact, its performance degrades for noisy low-contrast regions).

A variant on NCC, which is related to the bias–gain regression implicit in the matching score (8.9), is the *normalized SSD* score

$$E_{\mathrm{NSSD}}(\boldsymbol{u}) = \frac{1}{2}\frac{\sum_i \left[[I_0(\boldsymbol{x}_i) - \overline{I_0}] - [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - \overline{I_1}]\right]^2}{\sqrt{\sum_i [I_0(\boldsymbol{x}_i) - \overline{I_0}]^2 + [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - \overline{I_1}]^2}} \tag{8.14}$$

recently proposed by Criminisi, Shotton, Blake *et al.* (2007). In their experiments, they find that it produces comparable results to NCC, but is more efficient when applied to a large number of overlapping patches using a moving average technique (Section 3.2.2).

8.1.1 Hierarchical motion estimation

Now that we have a well-defined alignment cost function to optimize, how can we find its minimum? The simplest solution is to do a *full search* over some range of shifts, using either integer or sub-pixel steps. This is often the approach used for *block matching* in *motion compensated video compression*, where a range of possible motions (say, ± 16 pixels) is explored.[4]

To accelerate this search process, *hierarchical motion estimation* is often used: an image pyramid (Section 3.5) is constructed and a search over a smaller number of discrete pixels (corresponding to the same range of motion) is first performed at coarser levels (Quam 1984; Anandan 1989; Bergen, Anandan, Hanna *et al.* 1992). The motion estimate from one level of the pyramid is then used to initialize a smaller *local* search at the next finer level. Alternatively, several seeds (good solutions) from the coarse level can be used to initialize the fine-level search. While this is not guaranteed to produce the same result as a full search, it usually works almost as well and is much faster.

More formally, let

$$I_k^{(l)}(\boldsymbol{x}_j) \leftarrow \tilde{I}_k^{(l-1)}(2\boldsymbol{x}_j) \tag{8.15}$$

be the *decimated* image at level l obtained by subsampling (*downsampling*) a smoothed version of the image at level $l-1$. See Section 3.5 for how to perform the required downsampling (pyramid construction) without introducing too much aliasing.

At the coarsest level, we search for the best displacement $\boldsymbol{u}^{(l)}$ that minimizes the difference between images $I_0^{(l)}$ and $I_1^{(l)}$. This is usually done using a full search over some range of displacements $\boldsymbol{u}^{(l)} \in 2^{-l}[-S, S]^2$, where S is the desired *search range* at the finest (original) resolution level, optionally followed by the incremental refinement step described in Section 8.1.3.

Once a suitable motion vector has been estimated, it is used to *predict* a likely displacement

$$\hat{\boldsymbol{u}}^{(l-1)} \leftarrow 2\boldsymbol{u}^{(l)} \tag{8.16}$$

for the next finer level.[5] The search over displacements is then repeated at the finer level over a much narrower range of displacements, say $\hat{\boldsymbol{u}}^{(l-1)} \pm 1$, again optionally combined with an incremental refinement step (Anandan 1989). Alternatively, one of the images can be *warped* (resampled) by the current motion estimate, in which case only small incremental motions need to be computed at the finer level. A nice description of the whole process, extended to parametric motion estimation (Section 8.2), is provided by Bergen, Anandan, Hanna *et al.* (1992).

8.1.2 Fourier-based alignment

When the search range corresponds to a significant fraction of the larger image (as is the case in image stitching, see Chapter 9), the hierarchical approach may not work that well, since

[4] In stereo matching (Section 11.1.2), an explicit search over all possible disparities (i.e., a *plane sweep*) is almost always performed, since the number of search hypotheses is much smaller due to the 1D nature of the potential displacements.

[5] This doubling of displacements is only necessary if displacements are defined in integer *pixel* coordinates, which is the usual case in the literature (Bergen, Anandan, Hanna *et al.* 1992). If *normalized device coordinates* (Section 2.1.5) are used instead, the displacements (and search ranges) need not change from level to level, although the step sizes will need to be adjusted, to keep search steps of roughly one pixel.

it is often not possible to coarsen the representation too much before significant features are blurred away. In this case, a Fourier-based approach may be preferable.

Fourier-based alignment relies on the fact that the Fourier transform of a shifted signal has the same magnitude as the original signal but a linearly varying phase (Section 3.4), i.e.,

$$\mathcal{F}\{I_1(\boldsymbol{x}+\boldsymbol{u})\} = \mathcal{F}\{I_1(\boldsymbol{x})\}\, e^{-ju\cdot\boldsymbol{\omega}} = \mathcal{I}_1(\boldsymbol{\omega})e^{-ju\cdot\boldsymbol{\omega}}, \tag{8.17}$$

where $\boldsymbol{\omega}$ is the vector-valued angular frequency of the Fourier transform and we use calligraphic notation $\mathcal{I}_1(\boldsymbol{\omega}) = \mathcal{F}\{I_1(\boldsymbol{x})\}$ to denote the Fourier transform of a signal (Section 3.4).

Another useful property of Fourier transforms is that convolution in the spatial domain corresponds to multiplication in the Fourier domain (Section 3.4).[6] Thus, the Fourier transform of the cross-correlation function E_{CC} can be written as

$$\mathcal{F}\{E_{\mathrm{CC}}(\boldsymbol{u})\} = \mathcal{F}\left\{\sum_i I_0(\boldsymbol{x}_i)I_1(\boldsymbol{x}_i+\boldsymbol{u})\right\} = \mathcal{F}\{I_0(\boldsymbol{u})\bar{*}I_1(\boldsymbol{u})\} = \mathcal{I}_0(\boldsymbol{\omega})\mathcal{I}_1^*(\boldsymbol{\omega}), \tag{8.18}$$

where

$$f(\boldsymbol{u})\bar{*}g(\boldsymbol{u}) = \sum_i f(\boldsymbol{x}_i)g(\boldsymbol{x}_i+\boldsymbol{u}) \tag{8.19}$$

is the *correlation* function, i.e., the convolution of one signal with the reverse of the other, and $\mathcal{I}_1^*(\boldsymbol{\omega})$ is the *complex conjugate* of $\mathcal{I}_1(\boldsymbol{\omega})$. This is because convolution is defined as the summation of one signal with the reverse of the other (Section 3.4).

Thus, to efficiently evaluate E_{CC} over the range of all possible values of $\boldsymbol{u}$, we take the Fourier transforms of both images $I_0(\boldsymbol{x})$ and $I_1(\boldsymbol{x})$, multiply both transforms together (after conjugating the second one), and take the inverse transform of the result. The Fast Fourier Transform algorithm can compute the transform of an $N \times M$ image in $\mathrm{O}(NM\log NM)$ operations (Bracewell 1986). This can be significantly faster than the $\mathrm{O}(N^2M^2)$ operations required to do a full search when the full range of image overlaps is considered.

While Fourier-based convolution is often used to accelerate the computation of image correlations, it can also be used to accelerate the sum of squared differences function (and its variants). Consider the SSD formula given in (8.1). Its Fourier transform can be written as

$$\begin{aligned}\mathcal{F}\{E_{\mathrm{SSD}}(\boldsymbol{u})\} &= \mathcal{F}\left\{\sum_i [I_1(\boldsymbol{x}_i+\boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2\right\} \\ &= \delta(\boldsymbol{\omega})\sum_i [I_0^2(\boldsymbol{x}_i) + I_1^2(\boldsymbol{x}_i)] - 2\mathcal{I}_0(\boldsymbol{\omega})\mathcal{I}_1^*(\boldsymbol{\omega}).\end{aligned} \tag{8.20}$$

Thus, the SSD function can be computed by taking twice the correlation function and subtracting it from the sum of the energies in the two images.

Windowed correlation. Unfortunately, the Fourier convolution theorem only applies when the summation over $\boldsymbol{x}_i$ is performed over *all* the pixels in both images, using a circular shift of the image when accessing pixels outside the original boundaries. While this is

[6] In fact, the Fourier shift property (8.17) derives from the convolution theorem by observing that shifting is equivalent to convolution with a displaced delta function $\delta(\boldsymbol{x}-\boldsymbol{u})$.

acceptable for small shifts and comparably sized images, it makes no sense when the images overlap by a small amount or one image is a small subset of the other.

In that case, the cross-correlation function should be replaced with a *windowed* (weighted) cross-correlation function,

$$E_{\mathrm{WCC}}(\boldsymbol{u}) = \sum_i w_0(\boldsymbol{x}_i) I_0(\boldsymbol{x}_i)\, w_1(\boldsymbol{x}_i + \boldsymbol{u}) I_1(\boldsymbol{x}_i + \boldsymbol{u}), \tag{8.21}$$

$$= [w_0(\boldsymbol{x}) I_0(\boldsymbol{x})] \bar{*} [w_1(\boldsymbol{x}) I_1(\boldsymbol{x})] \tag{8.22}$$

where the weighting functions w_0 and w_1 are zero outside the valid ranges of the images and both images are padded so that circular shifts return 0 values outside the original image boundaries.

An even more interesting case is the computation of the *weighted* SSD function introduced in Equation (8.5),

$$E_{\mathrm{WSSD}}(\boldsymbol{u}) = \sum_i w_0(\boldsymbol{x}_i) w_1(\boldsymbol{x}_i + \boldsymbol{u}) [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2. \tag{8.23}$$

Expanding this as a sum of correlations and deriving the appropriate set of Fourier transforms is left for Exercise 8.1.

The same kind of derivation can also be applied to the bias–gain corrected sum of squared difference function E_{BG} (8.9). Again, Fourier transforms can be used to efficiently compute all the correlations needed to perform the linear regression in the bias and gain parameters in order to estimate the exposure-compensated difference for each potential shift (Exercise 8.1).

Phase correlation. A variant of regular correlation (8.18) that is sometimes used for motion estimation is *phase correlation* (Kuglin and Hines 1975; Brown 1992). Here, the spectrum of the two signals being matched is *whitened* by dividing each per-frequency product in (8.18) by the magnitudes of the Fourier transforms,

$$\mathcal{F}\{E_{\mathrm{PC}}(\boldsymbol{u})\} = \frac{\mathcal{I}_0(\boldsymbol{\omega}) \mathcal{I}_1^*(\boldsymbol{\omega})}{\|\mathcal{I}_0(\boldsymbol{\omega})\| \|\mathcal{I}_1(\boldsymbol{\omega})\|} \tag{8.24}$$

before taking the final inverse Fourier transform. In the case of noiseless signals with perfect (cyclic) shift, we have $I_1(\boldsymbol{x} + \boldsymbol{u}) = I_0(\boldsymbol{x})$ and hence, from Equation (8.17), we obtain

$$\mathcal{F}\{I_1(\boldsymbol{x} + \boldsymbol{u})\} = \mathcal{I}_1(\boldsymbol{\omega}) e^{-2\pi j u \cdot \boldsymbol{\omega}} = \mathcal{I}_0(\boldsymbol{\omega}) \text{ and}$$

$$\mathcal{F}\{E_{\mathrm{PC}}(\boldsymbol{u})\} = e^{-2\pi j u \cdot \boldsymbol{\omega}}. \tag{8.25}$$

The output of phase correlation (under ideal conditions) is therefore a single spike (impulse) located at the correct value of $\boldsymbol{u}$, which (in principle) makes it easier to find the correct estimate.

Phase correlation has a reputation in some quarters of outperforming regular correlation, but this behavior depends on the characteristics of the signals and noise. If the original images are contaminated by noise in a narrow frequency band (e.g., low-frequency noise or peaked frequency "hum"), the whitening process effectively de-emphasizes the noise in these regions. However, if the original signals have very low signal-to-noise ratio at some frequencies (say, two blurry or low-textured images with lots of high-frequency noise), the whitening process can actually decrease performance (see Exercise 8.1).

Recently, gradient cross-correlation has emerged as a promising alternative to phase correlation (Argyriou and Vlachos 2003), although further systematic studies are probably warranted. Phase correlation has also been studied by Fleet and Jepson (1990) as a method for estimating general optical flow and stereo disparity.

Rotations and scale. While Fourier-based alignment is mostly used to estimate translational shifts between images, it can, under certain limited conditions, also be used to estimate in-plane rotations and scales. Consider two images that are related *purely* by rotation, i.e.,

$$I_1(\hat{\boldsymbol{R}}\boldsymbol{x}) = I_0(\boldsymbol{x}). \tag{8.26}$$

If we re-sample the images into *polar coordinates*,

$$\tilde{I}_0(r,\theta) = I_0(r\cos\theta, r\sin\theta) \;\text{ and }\; \tilde{I}_1(r,\theta) = I_1(r\cos\theta, r\sin\theta), \tag{8.27}$$

we obtain

$$\tilde{I}_1(r,\theta+\hat{\theta}) = \tilde{I}_0(r,\theta). \tag{8.28}$$

The desired rotation can then be estimated using a Fast Fourier Transform (FFT) shift-based technique.

If the two images are also related by a scale,

$$I_1(e^{\hat{s}}\hat{\boldsymbol{R}}\boldsymbol{x}) = I_0(\boldsymbol{x}), \tag{8.29}$$

we can re-sample into *log-polar coordinates*,

$$\tilde{I}_0(s,\theta) = I_0(e^s\cos\theta, e^s\sin\theta) \;\text{ and }\; \tilde{I}_1(s,\theta) = I_1(e^s\cos\theta, e^s\sin\theta), \tag{8.30}$$

to obtain

$$\tilde{I}_1(s+\hat{s},\theta+\hat{\theta}) = I_0(s,\theta). \tag{8.31}$$

In this case, care must be taken to choose a suitable range of s values that reasonably samples the original image.

For images that are also translated by a small amount,

$$I_1(e^{\hat{s}}\hat{\boldsymbol{R}}\boldsymbol{x}+\boldsymbol{t}) = I_0(\boldsymbol{x}), \tag{8.32}$$

De Castro and Morandi (1987) propose an ingenious solution that uses several steps to estimate the unknown parameters. First, both images are converted to the Fourier domain and only the magnitudes of the transformed images are retained. In principle, the Fourier magnitude images are insensitive to translations in the image plane (although the usual caveats about border effects apply). Next, the two magnitude images are aligned in rotation and scale using the polar or log-polar representations. Once rotation and scale are estimated, one of the images can be de-rotated and scaled and a regular translational algorithm can be applied to estimate the translational shift.

Unfortunately, this trick only applies when the images have large overlap (small translational motion). For more general motion of patches or images, the parametric motion estimator described in Section 8.2 or the feature-based approaches described in Section 6.1 need to be used.

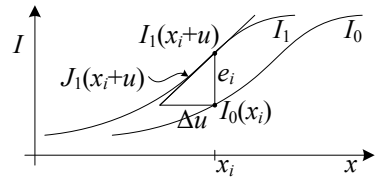


Figure 8.2 Taylor series approximation of a function and the incremental computation of the optical flow correction amount. $\boldsymbol{J}_1(\boldsymbol{x}_i + \boldsymbol{u})$ is the image gradient at $(\boldsymbol{x}_i + \boldsymbol{u})$ and e_i is the current intensity difference.

8.1.3 Incremental refinement

The techniques described up till now can estimate alignment to the nearest pixel (or potentially fractional pixel if smaller search steps are used). In general, image stabilization and stitching applications require much higher accuracies to obtain acceptable results.

To obtain better *sub-pixel* estimates, we can use one of several techniques described by Tian and Huhns (1986). One possibility is to evaluate several discrete (integer or fractional) values of (u, v) around the best value found so far and to *interpolate* the matching score to find an analytic minimum.

A more commonly used approach, first proposed by Lucas and Kanade (1981), is to perform *gradient descent* on the SSD energy function (8.1), using a Taylor series expansion of the image function (Figure 8.2),

$$E_{\mathrm{LK-SSD}}(\boldsymbol{u} + \Delta\boldsymbol{u}) = \sum_i [I_1(\boldsymbol{x}_i + \boldsymbol{u} + \Delta\boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2 \tag{8.33}$$

$$\approx \sum_i [I_1(\boldsymbol{x}_i + \boldsymbol{u}) + \boldsymbol{J}_1(\boldsymbol{x}_i + \boldsymbol{u})\Delta\boldsymbol{u} - I_0(\boldsymbol{x}_i)]^2 \tag{8.34}$$

$$= \sum_i [\boldsymbol{J}_1(\boldsymbol{x}_i + \boldsymbol{u})\Delta\boldsymbol{u} + e_i]^2, \tag{8.35}$$

where

$$\boldsymbol{J}_1(\boldsymbol{x}_i + \boldsymbol{u}) = \nabla I_1(\boldsymbol{x}_i + \boldsymbol{u}) = (\frac{\partial I_1}{\partial x}, \frac{\partial I_1}{\partial y})(\boldsymbol{x}_i + \boldsymbol{u}) \tag{8.36}$$

is the *image gradient* or *Jacobian* at $(\boldsymbol{x}_i + \boldsymbol{u})$ and

$$e_i = I_1(\boldsymbol{x}_i + \boldsymbol{u}) - I_0(\boldsymbol{x}_i), \tag{8.37}$$

first introduced in (8.1), is the current intensity error.[7] The gradient at a particular sub-pixel location $(\boldsymbol{x}_i + \boldsymbol{u})$ can be computed using a variety of techniques, the simplest of which is to simply take the horizontal and vertical differences between pixels $\boldsymbol{x}$ and $\boldsymbol{x} + (1, 0)$ or $\boldsymbol{x} + (0, 1)$. More sophisticated derivatives can sometimes lead to noticeable performance improvements.

The linearized form of the incremental update to the SSD error (8.35) is often called the *optical flow constraint* or *brightness constancy constraint* equation

$$I_x u + I_y v + I_t = 0, \tag{8.38}$$

[7] We follow the convention, commonly used in robotics and by Baker and Matthews (2004), that derivatives with respect to (column) vectors result in row vectors, so that fewer transposes are needed in the formulas.

where the subscripts in I_x and I_y denote spatial derivatives, and I_t is called the *temporal derivative*, which makes sense if we are computing instantaneous velocity in a video sequence. When squared and summed or integrated over a region, it can be used to compute optic flow (Horn and Schunck 1981).

The above least squares problem (8.35) can be minimized by solving the associated *normal equations* (Appendix A.2),

$$\boldsymbol{A}\Delta\boldsymbol{u} = \boldsymbol{b} \tag{8.39}$$

where

$$\boldsymbol{A} = \sum_i \boldsymbol{J}_1^T(\boldsymbol{x}_i + \boldsymbol{u})\boldsymbol{J}_1(\boldsymbol{x}_i + \boldsymbol{u}) \tag{8.40}$$

and

$$\boldsymbol{b} = -\sum_i e_i \boldsymbol{J}_1^T(\boldsymbol{x}_i + \boldsymbol{u}) \tag{8.41}$$

are called the (Gauss–Newton approximation of the) *Hessian* and *gradient-weighted residual vector*, respectively.[8] These matrices are also often written as

$$\boldsymbol{A} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \text{ and } \boldsymbol{b} = -\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}. \tag{8.42}$$

The gradients required for $\boldsymbol{J}_1(\boldsymbol{x}_i + \boldsymbol{u})$ can be evaluated at the same time as the image warps required to estimate $I_1(\boldsymbol{x}_i + \boldsymbol{u})$ (Section 3.6.1 (3.89)) and, in fact, are often computed as a side-product of image interpolation. If efficiency is a concern, these gradients can be replaced by the gradients in the *template* image,

$$\boldsymbol{J}_1(\boldsymbol{x}_i + \boldsymbol{u}) \approx \boldsymbol{J}_0(\boldsymbol{x}_i), \tag{8.43}$$

since near the correct alignment, the template and displaced target images should look similar. This has the advantage of allowing the pre-computation of the Hessian and Jacobian images, which can result in significant computational savings (Hager and Belhumeur 1998; Baker and Matthews 2004). A further reduction in computation can be obtained by writing the warped image $I_1(\boldsymbol{x}_i + \boldsymbol{u})$ used to compute e_i in (8.37) as a convolution of a sub-pixel interpolation filter with the discrete samples in I_1 (Peleg and Rav-Acha 2006). Precomputing the inner product between the gradient field and shifted version of I_1 allows the iterative re-computation of e_i to be performed in constant time (independent of the number of pixels).

The effectiveness of the above incremental update rule relies on the quality of the Taylor series approximation. When far away from the true displacement (say, 1–2 pixels), several iterations may be needed. It is possible, however, to estimate a value for $\boldsymbol{J}_1$ using a least squares fit to a series of larger displacements in order to increase the range of convergence (Jurie and Dhome 2002) or to "learn" a special-purpose recognizer for a given patch (Avidan 2001; Williams, Blake, and Cipolla 2003; Lepetit, Pilet, and Fua 2006; Hinterstoisser, Benhimane, Navab *et al.* 2008; Özuysal, Calonder, Lepetit *et al.* 2010) as discussed in Section 4.1.4.

A commonly used stopping criterion for incremental updating is to monitor the magnitude of the displacement correction $\|\boldsymbol{u}\|$ and to stop when it drops below a certain threshold (say,

[8] The true Hessian is the full second derivative of the error function E, which may not be positive definite—see Section 6.1.3 and Appendix A.3.

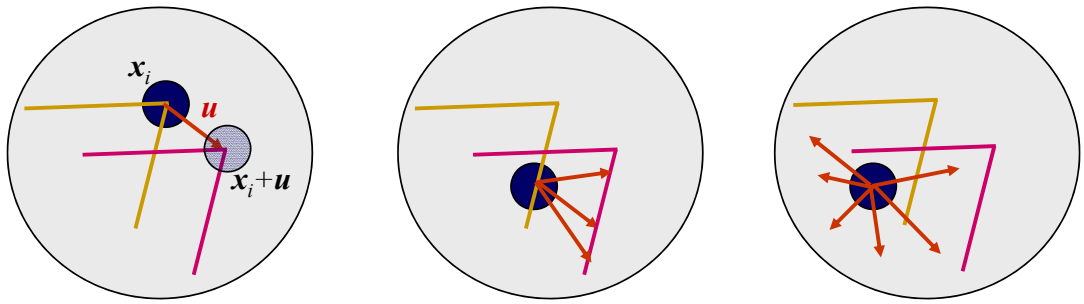


Figure 8.3 Aperture problems for different image regions, denoted by the orange and red L-shaped structures, overlaid in the same image to make it easier to diagram the flow. (a) A window $w(\boldsymbol{x}_i)$ centered at $\boldsymbol{x}_i$ (black circle) can uniquely be matched to its corresponding structure at $\boldsymbol{x}_i + \boldsymbol{u}$ in the second (red) image. (b) A window centered on the edge exhibits the classic aperture problem, since it can be matched to a 1D family of possible locations. (c) In a completely textureless region, the matches become totally unconstrained.

$1/10$ of a pixel). For larger motions, it is usual to combine the incremental update rule with a hierarchical coarse-to-fine search strategy, as described in Section 8.1.1.

Conditioning and aperture problems. Sometimes, the inversion of the linear system (8.39) can be poorly conditioned because of lack of two-dimensional texture in the patch being aligned. A commonly occurring example of this is the *aperture problem*, first identified in some of the early papers on optical flow (Horn and Schunck 1981) and then studied more extensively by Anandan (1989). Consider an image patch that consists of a slanted edge moving to the right (Figure 8.3). Only the *normal* component of the velocity (displacement) can be reliably recovered in this case. This manifests itself in (8.39) as a *rank-deficient* matrix $\boldsymbol{A}$, i.e., one whose smaller eigenvalue is very close to zero.[9]

When Equation (8.39) is solved, the component of the displacement along the edge is very poorly conditioned and can result in wild guesses under small noise perturbations. One way to mitigate this problem is to add a *prior* (soft constraint) on the expected range of motions (Simoncelli, Adelson, and Heeger 1991; Baker, Gross, and Matthews 2004; Govindu 2006). This can be accomplished by adding a small value to the diagonal of $\boldsymbol{A}$, which essentially biases the solution towards smaller $\Delta\boldsymbol{u}$ values that still (mostly) minimize the squared error.

However, the pure Gaussian model assumed when using a simple (fixed) quadratic prior, as in (Simoncelli, Adelson, and Heeger 1991), does not always hold in practice, e.g., because of aliasing along strong edges (Triggs 2004). For this reason, it may be prudent to add some small fraction (say, 5%) of the larger eigenvalue to the smaller one before doing the matrix inversion.

Uncertainty modeling. The reliability of a particular patch-based motion estimate can be captured more formally with an *uncertainty model*. The simplest such model is a *covariance matrix*, which captures the expected variance in the motion estimate in all possible directions.

[9] The matrix $\boldsymbol{A}$ is by construction always guaranteed to be symmetric positive semi-definite, i.e., it has real non-negative eigenvalues.

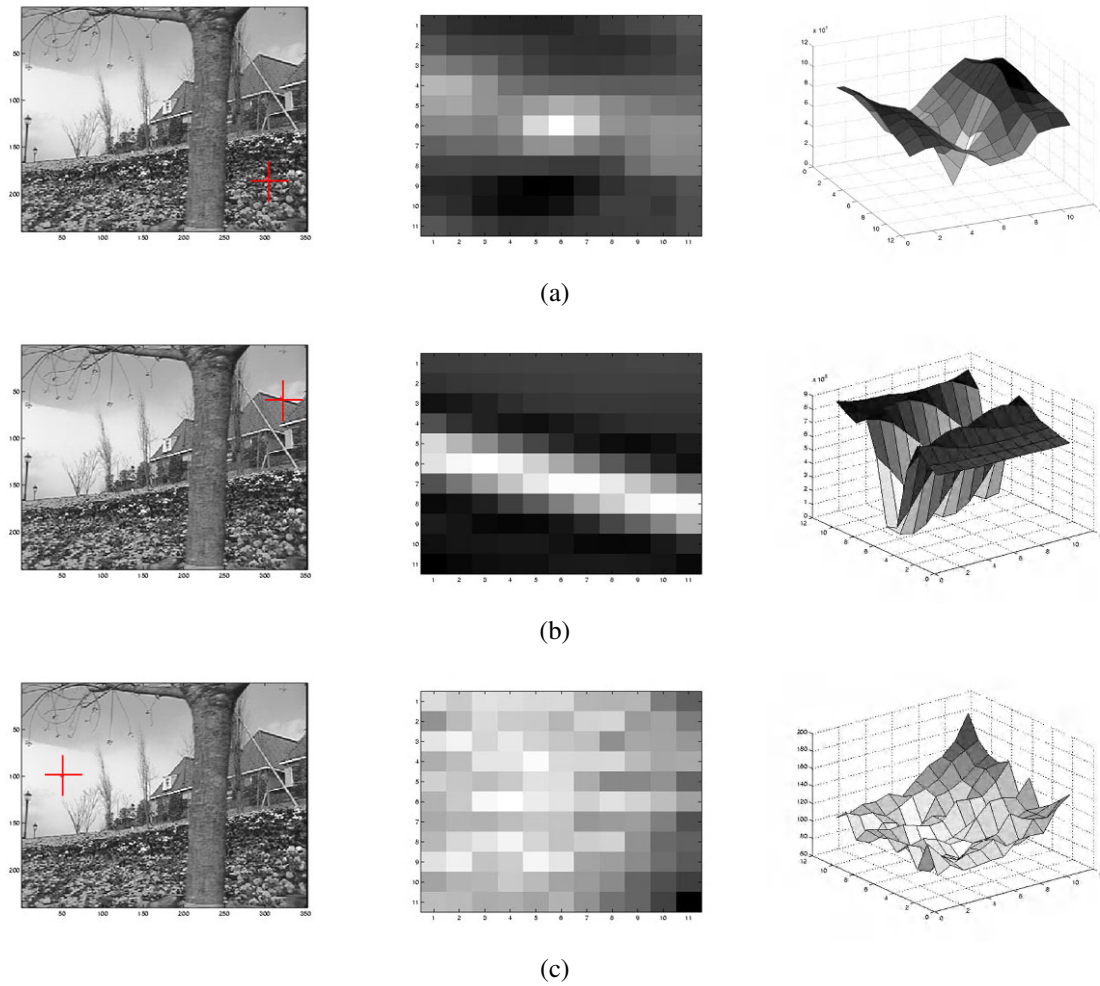


Figure 8.4 SSD surfaces corresponding to three locations (red crosses) in an image: (a) highly textured area, strong minimum, low uncertainty; (b) strong edge, aperture problem, high uncertainty in one direction; (c) weak texture, no clear minimum, large uncertainty.

As discussed in Section 6.1.4 and Appendix B.6, under small amounts of additive Gaussian noise, it can be shown that the covariance matrix $\Sigma_{\boldsymbol{u}}$ is proportional to the inverse of the Hessian $\boldsymbol{A}$,

$$\Sigma_{\boldsymbol{u}} = \sigma_n^2 \boldsymbol{A}^{-1}, \tag{8.44}$$

where σ_n^2 is the variance of the additive Gaussian noise (Anandan 1989; Matthies, Kanade, and Szeliski 1989; Szeliski 1989).

For larger amounts of noise, the linearization performed by the Lucas–Kanade algorithm in (8.35) is only approximate, so the above quantity becomes a *Cramer–Rao lower bound* on the true covariance. Thus, the minimum and maximum eigenvalues of the Hessian $\boldsymbol{A}$ can now be interpreted as the (scaled) inverse variances in the least-certain and most-certain directions of motion. (A more detailed analysis using a more realistic model of image noise is given by Steele and Jaynes (2005).) Figure 8.4 shows the local SSD surfaces for three different pixel locations in an image. As you can see, the surface has a clear minimum in the highly textured region and suffers from the aperture problem near the strong edge.

Bias and gain, weighting, and robust error metrics. The Lucas–Kanade update rule can also be applied to the bias–gain equation (8.9) to obtain

$$E_{\mathrm{LK-BG}}(\boldsymbol{u} + \Delta\boldsymbol{u}) = \sum_i [\boldsymbol{J}_1(\boldsymbol{x}_i + \boldsymbol{u})\Delta\boldsymbol{u} + e_i - \alpha I_0(\boldsymbol{x}_i) - \beta]^2 \tag{8.45}$$

(Lucas and Kanade 1981; Gennert 1988; Fuh and Maragos 1991; Baker, Gross, and Matthews 2003). The resulting 4×4 system of equations can be solved to simultaneously estimate the translational displacement update $\Delta\boldsymbol{u}$ and the bias and gain parameters β and α.

A similar formulation can be derived for images (templates) that have a *linear appearance variation*,

$$I_1(\boldsymbol{x} + \boldsymbol{u}) \approx I_0(\boldsymbol{x}) + \sum_j \lambda_j B_j(\boldsymbol{x}), \tag{8.46}$$

where the $B_j(\boldsymbol{x})$ are the *basis images* and the λ_j are the unknown coefficients (Hager and Belhumeur 1998; Baker, Gross, Ishikawa *et al.* 2003; Baker, Gross, and Matthews 2003). Potential linear appearance variations include illumination changes (Hager and Belhumeur 1998) and small non-rigid deformations (Black and Jepson 1998).

A weighted (windowed) version of the Lucas–Kanade algorithm is also possible:

$$E_{\mathrm{LK-WSSD}}(\boldsymbol{u} + \Delta\boldsymbol{u}) = \sum_i w_0(\boldsymbol{x}_i) w_1(\boldsymbol{x}_i + \boldsymbol{u})[\boldsymbol{J}_1(\boldsymbol{x}_i + \boldsymbol{u})\Delta\boldsymbol{u} + e_i]^2. \tag{8.47}$$

Note that here, in deriving the Lucas–Kanade update from the original weighted SSD function (8.5), we have neglected taking the derivative of the $w_1(\boldsymbol{x}_i + \boldsymbol{u})$ weighting function with respect to $\boldsymbol{u}$, which is usually acceptable in practice, especially if the weighting function is a binary mask with relatively few transitions.

Baker, Gross, Ishikawa *et al.* (2003) only use the $w_0(\boldsymbol{x})$ term, which is reasonable if the two images have the same extent and no (independent) cutouts in the overlap region. They also discuss the idea of making the weighting proportional to $\nabla I(\boldsymbol{x})$, which helps for very noisy images, where the gradient itself is noisy. Similar observations, formulated in terms of *total least squares* (Van Huffel and Vandewalle 1991; Van Huffel and Lemmerling 2002),

have been made by other researchers studying optical flow (Weber and Malik 1995; Bab-Hadiashar and Suter 1998b; Mühlich and Mester 1998). Lastly, Baker, Gross, Ishikawa *et al.* (2003) show how evaluating Equation (8.47) at just the *most reliable* (highest gradient) pixels does not significantly reduce performance for large enough images, even if only 5–10% of the pixels are used. (This idea was originally proposed by Dellaert and Collins (1999), who used a more sophisticated selection criterion.)

The Lucas–Kanade incremental refinement step can also be applied to the robust error metric introduced in Section 8.1,

$$E_{\mathrm{LK-SRD}}(\boldsymbol{u}+\Delta\boldsymbol{u}) = \sum_i \rho(\boldsymbol{J}_1(\boldsymbol{x}_i+\boldsymbol{u})\Delta\boldsymbol{u} + e_i), \tag{8.48}$$

which can be solved using the *iteratively reweighted least squares* technique described in Section 6.1.4.

8.2 Parametric motion

Many image alignment tasks, for example image stitching with handheld cameras, require the use of more sophisticated motion models, as described in Section 2.1.2. Since these models, e.g., affine deformations, typically have more parameters than pure translation, a full search over the possible range of values is impractical. Instead, the incremental Lucas–Kanade algorithm can be generalized to parametric motion models and used in conjunction with a hierarchical search algorithm (Lucas and Kanade 1981; Rehg and Witkin 1991; Fuh and Maragos 1991; Bergen, Anandan, Hanna *et al.* 1992; Shashua and Toelg 1997; Shashua and Wexler 2001; Baker and Matthews 2004).

For parametric motion, instead of using a single constant translation vector $\boldsymbol{u}$, we use a spatially varying *motion field* or *correspondence map*, $\boldsymbol{x}'(\boldsymbol{x};\boldsymbol{p})$, parameterized by a low-dimensional vector $\boldsymbol{p}$, where $\boldsymbol{x}'$ can be any of the motion models presented in Section 2.1.2. The parametric incremental motion update rule now becomes

$$\begin{aligned} E_{\mathrm{LK-PM}}(\boldsymbol{p}+\Delta\boldsymbol{p}) &= \sum_i [I_1(\boldsymbol{x}'(\boldsymbol{x}_i;\boldsymbol{p}+\Delta\boldsymbol{p})) - I_0(\boldsymbol{x}_i)]^2 && (8.49)\\ &\approx \sum_i [I_1(\boldsymbol{x}'_i) + \boldsymbol{J}_1(\boldsymbol{x}'_i)\Delta\boldsymbol{p} - I_0(\boldsymbol{x}_i)]^2 && (8.50)\\ &= \sum_i [\boldsymbol{J}_1(\boldsymbol{x}'_i)\Delta\boldsymbol{p} + e_i]^2, && (8.51) \end{aligned}$$

where the Jacobian is now

$$\boldsymbol{J}_1(\boldsymbol{x}'_i) = \frac{\partial I_1}{\partial \boldsymbol{p}} = \nabla I_1(\boldsymbol{x}'_i)\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{p}}(\boldsymbol{x}_i), \tag{8.52}$$

i.e., the product of the image gradient ∇I_1 with the Jacobian of the correspondence field, $\boldsymbol{J}_{x'} = \partial \boldsymbol{x}'/\partial \boldsymbol{p}$.

The motion Jacobians $\boldsymbol{J}_{\boldsymbol{x}'}$ for the 2D planar transformations introduced in Section 2.1.2 and Table 2.1 are given in Table 6.1. Note how we have re-parameterized the motion matrices so that they are always the identity at the origin $\boldsymbol{p} = 0$. This becomes useful later, when we talk about the compositional and inverse compositional algorithms. (It also makes it easier to impose priors on the motions.)

For parametric motion, the (Gauss–Newton) *Hessian* and *gradient-weighted residual vector* become

$$\boldsymbol{A} = \sum_i \boldsymbol{J}_{\boldsymbol{x}'}^T(\boldsymbol{x}_i)[\nabla I_1^T(\boldsymbol{x}_i')\nabla I_1(\boldsymbol{x}_i')]\boldsymbol{J}_{\boldsymbol{x}'}(\boldsymbol{x}_i) \tag{8.53}$$

and

$$\boldsymbol{b} = -\sum_i \boldsymbol{J}_{\boldsymbol{x}'}^T(\boldsymbol{x}_i)[e_i \nabla I_1^T(\boldsymbol{x}_i')]. \tag{8.54}$$

Note how the expressions inside the square brackets are the same ones evaluated for the simpler translational motion case (8.40–8.41).

Patch-based approximation. The computation of the Hessian and residual vectors for parametric motion can be significantly more expensive than for the translational case. For parametric motion with n parameters and N pixels, the accumulation of $\boldsymbol{A}$ and $\boldsymbol{b}$ takes $\mathrm{O}(n^2 N)$ operations (Baker and Matthews 2004). One way to reduce this by a significant amount is to divide the image up into smaller sub-blocks (patches) P_j and to only accumulate the simpler 2×2 quantities inside the square brackets at the pixel level (Shum and Szeliski 2000),

$$\boldsymbol{A}_j = \sum_{i \in P_j} \nabla I_1^T(\boldsymbol{x}_i')\nabla I_1(\boldsymbol{x}_i') \tag{8.55}$$

$$\boldsymbol{b}_j = \sum_{i \in P_j} e_i \nabla I_1^T(\boldsymbol{x}_i'). \tag{8.56}$$

The full Hessian and residual can then be approximated as

$$\boldsymbol{A} \approx \sum_j \boldsymbol{J}_{\boldsymbol{x}'}^T(\hat{\boldsymbol{x}}_j)[\sum_{i \in P_j} \nabla I_1^T(\boldsymbol{x}_i')\nabla I_1(\boldsymbol{x}_i')]\boldsymbol{J}_{\boldsymbol{x}'}(\hat{\boldsymbol{x}}_j) = \sum_j \boldsymbol{J}_{\boldsymbol{x}'}^T(\hat{\boldsymbol{x}}_j)\boldsymbol{A}_j\boldsymbol{J}_{\boldsymbol{x}'}(\hat{\boldsymbol{x}}_j) \tag{8.57}$$

and

$$\boldsymbol{b} \approx -\sum_j \boldsymbol{J}_{\boldsymbol{x}'}^T(\hat{\boldsymbol{x}}_j)[\sum_{i \in P_j} e_i \nabla I_1^T(\boldsymbol{x}_i')] = -\sum_j \boldsymbol{J}_{\boldsymbol{x}'}^T(\hat{\boldsymbol{x}}_j)\boldsymbol{b}_j, \tag{8.58}$$

where $\hat{\boldsymbol{x}}_j$ is the *center* of each patch P_j (Shum and Szeliski 2000). This is equivalent to replacing the true motion Jacobian with a piecewise-constant approximation. In practice, this works quite well. The relationship of this approximation to feature-based registration is discussed in Section 9.2.4.

Compositional approach. For a complex parametric motion such as a homography, the computation of the motion Jacobian becomes complicated and may involve a per-pixel division. Szeliski and Shum (1997) observed that this can be simplified by first warping the target image I_1 according to the current motion estimate $\boldsymbol{x}'(\boldsymbol{x};\boldsymbol{p})$,

$$\tilde{I}_1(\boldsymbol{x}) = I_1(\boldsymbol{x}'(\boldsymbol{x};\boldsymbol{p})), \tag{8.59}$$

and then comparing this *warped* image against the template $I_0(\boldsymbol{x})$,

$$E_{\mathrm{LK-SS}}(\Delta\boldsymbol{p}) = \sum_i [\tilde{I}_1(\tilde{\boldsymbol{x}}(\boldsymbol{x}_i;\Delta\boldsymbol{p})) - I_0(\boldsymbol{x}_i)]^2 \tag{8.60}$$

$$\approx \sum_i [\tilde{J}_1(\boldsymbol{x}_i)\Delta\boldsymbol{p} + e_i]^2 \tag{8.61}$$

$$= \sum_i [\nabla \tilde{I}_1(\boldsymbol{x}_i)\boldsymbol{J}_{\tilde{\boldsymbol{x}}}(\boldsymbol{x}_i)\Delta\boldsymbol{p} + e_i]^2. \tag{8.62}$$

Note that since the two images are assumed to be fairly similar, only an *incremental* parametric motion is required, i.e., the incremental motion can be evaluated around $\boldsymbol{p} = 0$, which can lead to considerable simplifications. For example, the Jacobian of the planar projective transform (6.19) now becomes

$$\boldsymbol{J}_{\tilde{\boldsymbol{x}}} = \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{p}}\bigg|_{\boldsymbol{p}=0} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x^2 & -xy \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2 \end{bmatrix}. \tag{8.63}$$

Once the incremental motion $\tilde{\boldsymbol{x}}$ has been computed, it can be *prepended* to the previously estimated motion, which is easy to do for motions represented with transformation matrices, such as those given in Tables 2.1 and 6.1. Baker and Matthews (2004) call this the *forward compositional* algorithm, since the target image is being re-warped and the final motion estimates are being composed.

If the appearance of the warped and template images is similar enough, we can replace the gradient of $\tilde{I}_1(\boldsymbol{x})$ with the gradient of $I_0(\boldsymbol{x})$, as suggested previously (8.43). This has potentially a big advantage in that it allows the pre-computation (and inversion) of the Hessian matrix $\boldsymbol{A}$ given in Equation (8.53). The residual vector $\boldsymbol{b}$ (8.54) can also be partially precomputed, i.e., the *steepest descent images* $\nabla I_0(\boldsymbol{x}) \boldsymbol{J}_{\tilde{\boldsymbol{x}}}(\boldsymbol{x})$ can precomputed and stored for later multiplication with the $e(\boldsymbol{x}) = \tilde{I}_1(\boldsymbol{x}) - I_0(\boldsymbol{x})$ error images (Baker and Matthews 2004). This idea was first suggested by Hager and Belhumeur (1998) in what Baker and Matthews (2004) call a *inverse additive* scheme.

Baker and Matthews (2004) introduce one more variant they call the *inverse compositional* algorithm. Rather than (conceptually) re-warping the warped target image $\tilde{I}_1(\boldsymbol{x})$, they instead warp the template image $I_0(\boldsymbol{x})$ and minimize

$$E_{\mathrm{LK-BM}}(\Delta \boldsymbol{p}) = \sum_i [\tilde{I}_1(\boldsymbol{x}_i) - I_0(\tilde{\boldsymbol{x}}(\boldsymbol{x}_i; \Delta \boldsymbol{p}))]^2 \tag{8.64}$$

$$\approx \sum_i [\nabla I_0(\boldsymbol{x}_i) \boldsymbol{J}_{\tilde{\boldsymbol{x}}}(\boldsymbol{x}_i) \Delta \boldsymbol{p} - e_i]^2. \tag{8.65}$$

This is identical to the forward warped algorithm (8.62) with the gradients $\nabla \tilde{I}_1(\boldsymbol{x})$ replaced by the gradients $\nabla I_0(\boldsymbol{x})$, except for the sign of e_i. The resulting update $\Delta \boldsymbol{p}$ is the *negative* of the one computed by the modified Equation (8.62) and hence the *inverse* of the incremental transformation must be prepended to the current transform. Because the inverse compositional algorithm has the potential of pre-computing the inverse Hessian and the steepest descent images, this makes it the preferred approach of those surveyed by Baker and Matthews (2004). Figure 8.5 (Baker, Gross, Ishikawa *et al.* 2003) beautifully shows all of the steps required to implement the inverse compositional algorithm.

Baker and Matthews (2004) also discuss the advantage of using Gauss–Newton iteration (i.e., the first-order expansion of the least squares, as above) compared to other approaches such as steepest descent and Levenberg–Marquardt. Subsequent parts of the series (Baker, Gross, Ishikawa *et al.* 2003; Baker, Gross, and Matthews 2003, 2004) discuss more advanced topics such as per-pixel weighting, pixel selection for efficiency, a more in-depth discussion of robust metrics and algorithms, linear appearance variations, and priors on parameters. They make for invaluable reading for anyone interested in implementing a highly tuned implementation of incremental image registration. Evangelidis and Psarakis (2008) provide some detailed experimental evaluations of these and other related approaches.

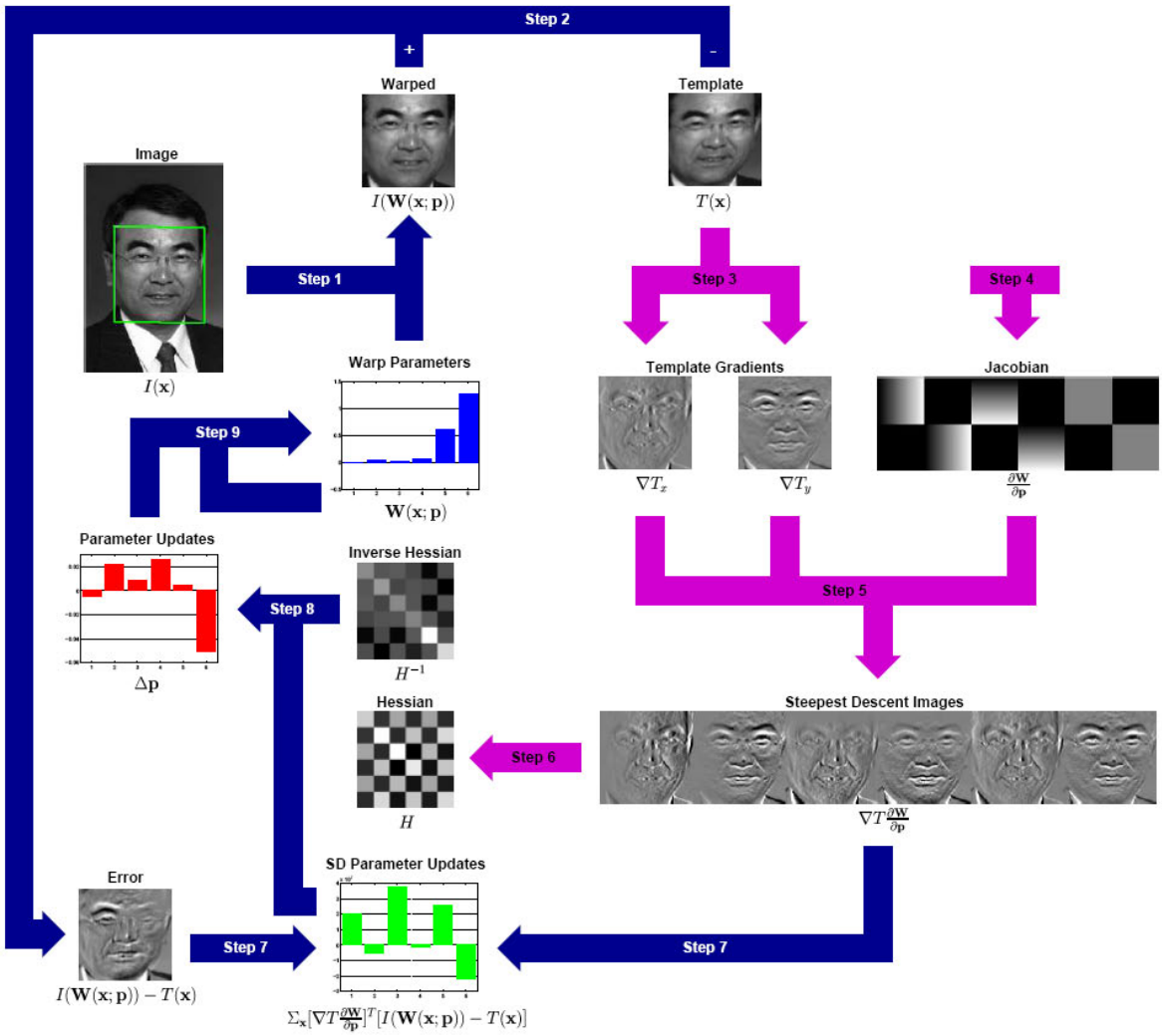


Figure 8.5 A schematic overview of the inverse compositional algorithm (copied, with permission, from (Baker, Gross, Ishikawa *et al.* 2003)). Steps 3–6 (light-colored arrows) are performed once as a pre-computation. The main algorithm simply consists of iterating: image warping (Step 1), image differencing (Step 2), image dot products (Step 7), multiplication with the inverse of the Hessian (Step 8), and the update to the warp (Step 9). All of these steps can be performed efficiently.

8.2.1 *Application*: Video stabilization

Video stabilization is one of the most widely used applications of parametric motion estimation (Hansen, Anandan, Dana *et al.* 1994; Irani, Rousso, and Peleg 1997; Morimoto and Chellappa 1997; Srinivasan, Chellappa, Veeraraghavan *et al.* 2005). Algorithms for stabilization run inside both hardware devices, such as camcorders and still cameras, and software packages for improving the visual quality of shaky videos.

In their paper on full-frame video stabilization, Matsushita, Ofek, Ge *et al.* (2006) give a nice overview of the three major stages of stabilization, namely motion estimation, motion smoothing, and image warping. Motion estimation algorithms often use a similarity transform to handle camera translations, rotations, and zooming. The tricky part is getting these algorithms to lock onto the background motion, which is a result of the camera movement, without getting distracted by independent moving foreground objects. Motion smoothing algorithms recover the low-frequency (slowly varying) part of the motion and then estimate the high-frequency shake component that needs to be removed. Finally, image warping algorithms apply the high-frequency correction to render the original frames as if the camera had undergone only the smooth motion.

The resulting stabilization algorithms can greatly improve the appearance of shaky videos but they often still contain visual artifacts. For example, image warping can result in missing borders around the image, which must be cropped, filled using information from other frames, or hallucinated using inpainting techniques (Section 10.5.1). Furthermore, video frames captured during fast motion are often blurry. Their appearance can be improved either using deblurring techniques (Section 10.3) or stealing sharper pixels from other frames with less motion or better focus (Matsushita, Ofek, Ge *et al.* 2006). Exercise 8.3 has you implement and test some of these ideas.

In situations where the camera is translating a lot in 3D, e.g., when the videographer is walking, an even better approach is to compute a full structure from motion reconstruction of the camera motion and 3D scene. A smooth 3D camera path can then be computed and the original video re-rendered using view interpolation with the interpolated 3D point cloud serving as the proxy geometry while preserving salient features (Liu, Gleicher, Jin *et al.* 2009). If you have access to a camera array instead of a single video camera, you can do even better using a light field rendering approach (Section 13.3) (Smith, Zhang, Jin *et al.* 2009).

8.2.2 Learned motion models

An alternative to parameterizing the motion field with a geometric deformation such as an affine transform is to learn a set of basis functions tailored to a particular application (Black, Yacoob, Jepson *et al.* 1997). First, a set of dense motion fields (Section 8.4) is computed from a set of training videos. Next, singular value decomposition (SVD) is applied to the stack of motion fields $\boldsymbol{u}_t(\boldsymbol{x})$ to compute the first few singular vectors $\boldsymbol{v}_k(\boldsymbol{x})$. Finally, for a new test sequence, a novel flow field is computed using a coarse-to-fine algorithm that estimates the unknown coefficient a_k in the parameterized flow field

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_k a_k \boldsymbol{v}_k(\boldsymbol{x}). \tag{8.66}$$

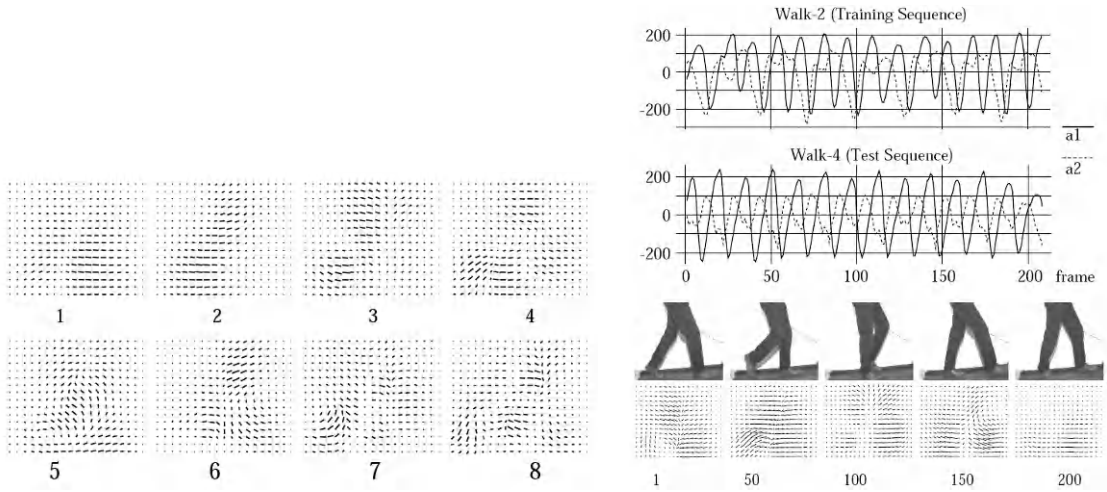


Figure 8.6 Learned parameterized motion fields for a walking sequence (Black, Yacoob, Jepson *et al.* 1997) © 1997 IEEE: (a) learned basis flow fields; (b) plots of motion coefficients over time and corresponding estimated motion fields.

Figure 8.6a shows a set of basis fields learned by observing videos of walking motions. Figure 8.6b shows the temporal evolution of the basis coefficients as well as a few of the recovered parametric motion fields. Note that similar ideas can also be applied to feature tracks (Torresani, Hertzmann, and Bregler 2008), which is a topic we discuss in more detail in Sections 4.1.4 and 12.6.4.

8.3 Spline-based motion

While parametric motion models are useful in a wide variety of applications (such as video stabilization and mapping onto planar surfaces), most image motion is too complicated to be captured by such low-dimensional models.

Traditionally, optical flow algorithms (Section 8.4) compute an independent motion estimate for each pixel, i.e., the number of flow vectors computed is equal to the number of input pixels. The general optical flow analog to Equation (8.1) can thus be written as

$$E_{\mathrm{SSD-OF}}(\{\boldsymbol{u}_i\}) = \sum_i [I_1(\boldsymbol{x}_i + \boldsymbol{u}_i) - I_0(\boldsymbol{x}_i)]^2. \tag{8.67}$$

Notice how in the above equation, the number of variables $\{\boldsymbol{u}_i\}$ is twice the number of measurements, so the problem is underconstrained.

The two classic approaches to this problem, which we study in Section 8.4, are to perform the summation over overlapping regions (the *patch-based* or *window-based* approach) or to add smoothness terms on the $\{\boldsymbol{u}_i\}$ field using *regularization* or *Markov random fields* (Section 3.7). In this section, we describe an alternative approach that lies somewhere between general optical flow (independent flow at each pixel) and parametric flow (a small number of global parameters). The approach is to represent the motion field as a two-dimensional *spline*

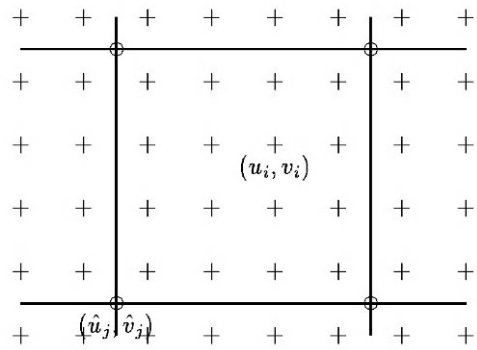


Figure 8.7 Spline motion field: the displacement vectors $\boldsymbol{u}_i = (u_i, v_i)$ are shown as pluses (+) and are controlled by the smaller number of control vertices $\hat{\boldsymbol{u}}_j = (\hat{u}_i, \hat{v}_j)$, which are shown as circles (○).

controlled by a smaller number of *control vertices* $\{\hat{\boldsymbol{u}}_j\}$ (Figure 8.7),

$$\boldsymbol{u}_i = \sum_j \hat{\boldsymbol{u}}_j B_j(\boldsymbol{x}_i) = \sum_j \hat{\boldsymbol{u}}_j w_{i,j}, \tag{8.68}$$

where the $B_j(\boldsymbol{x}_i)$ are called the *basis functions* and are only non-zero over a small *finite support* interval (Szeliski and Coughlan 1997). We call the $w_{ij} = B_j(\boldsymbol{x}_i)$ *weights* to emphasize that the $\{\boldsymbol{u}_i\}$ are known linear combinations of the $\{\hat{\boldsymbol{u}}_j\}$. Some commonly used spline basis functions are shown in Figure 8.8.

Substituting the formula for the individual per-pixel flow vectors $\boldsymbol{u}_i$ (8.68) into the SSD error metric (8.67) yields a parametric motion formula similar to Equation (8.50). The biggest difference is that the Jacobian $\boldsymbol{J}_1(\boldsymbol{x}'_i)$ (8.52) now consists of the sparse entries in the weight matrix $\boldsymbol{W} = [w_{ij}]$.

In situations where we know something more about the motion field, e.g., when the motion is due to a camera moving in a static scene, we can use more specialized motion models. For example, the *plane plus parallax* model (Section 2.1.5) can be naturally combined with a spline-based motion representation, where the in-plane motion is represented by a homography (6.19) and the out-of-plane parallax d is represented by a scalar variable at each spline control point (Szeliski and Kang 1995; Szeliski and Coughlan 1997).

In many cases, the small number of spline vertices results in a motion estimation problem that is well conditioned. However, if large textureless regions (or elongated edges subject to the aperture problem) persist across several spline patches, it may be necessary to add a *regularization* term to make the problem well posed (Section 3.7.1). The simplest way to do this is to directly add squared difference penalties between adjacent vertices in the spline control mesh $\{\hat{\boldsymbol{u}}_j\}$, as in (3.100). If a multi-resolution (coarse-to-fine) strategy is being used, it is important to re-scale these smoothness terms while going from level to level.

The linear system corresponding to the spline-based motion estimator is sparse and regular. Because it is usually of moderate size, it can often be solved using direct techniques such as Cholesky decomposition (Appendix A.4). Alternatively, if the problem becomes too large and subject to excessive fill-in, iterative techniques such as hierarchically preconditioned conjugate gradient (Szeliski 1990b, 2006b) can be used instead (Appendix A.5).

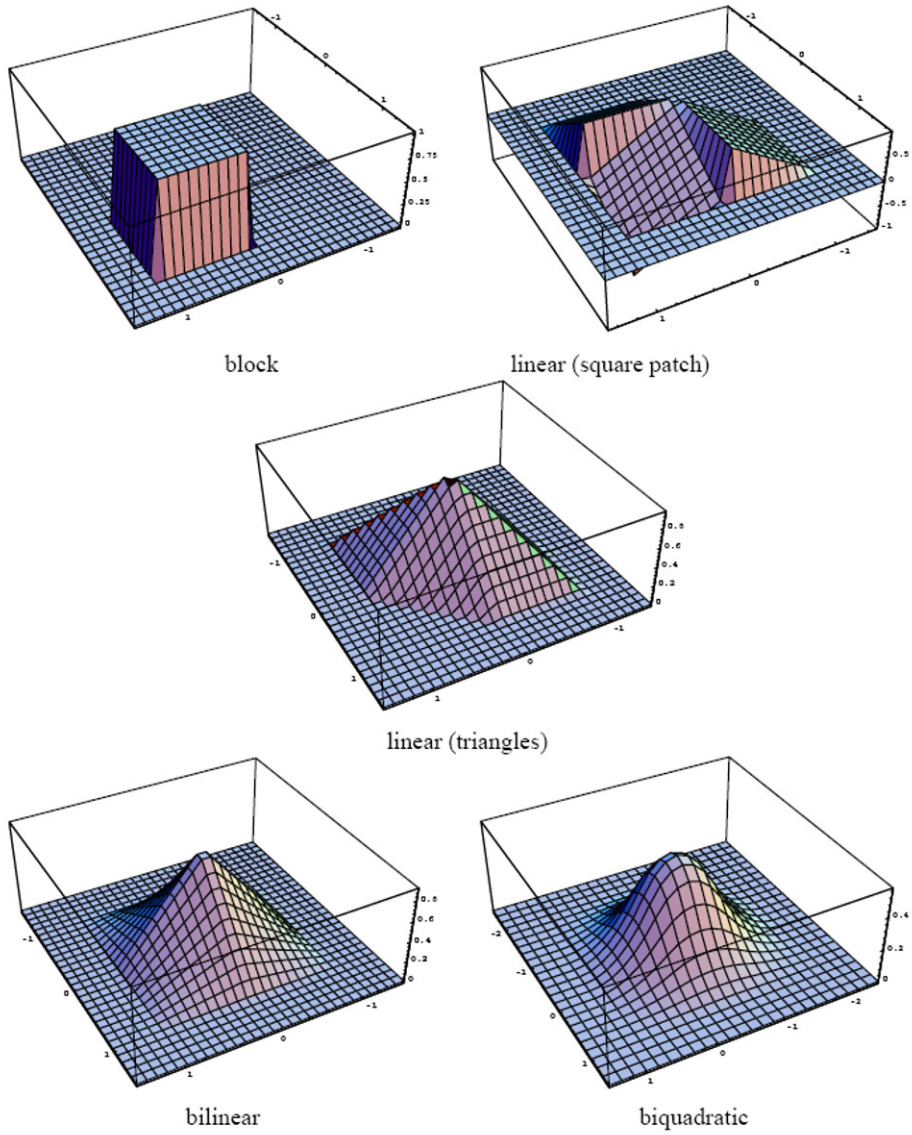
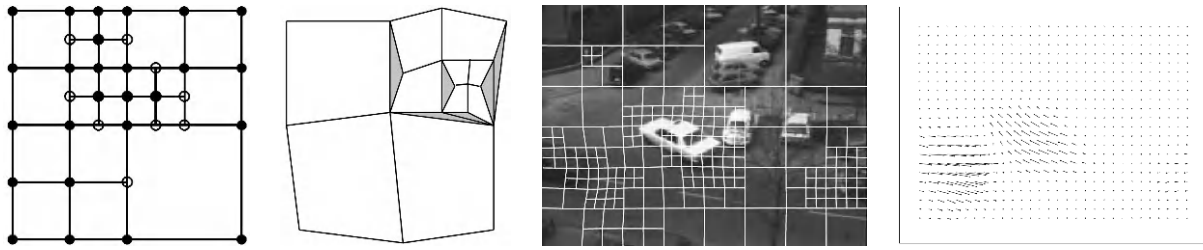


Figure 8.8 Sample spline basis functions (Szeliski and Coughlan 1997) © 1997 Springer. The block (constant) interpolator/basis corresponds to block-based motion estimation (Le Gall 1991). See Section 3.5.1 for more details on spline functions.



(a) (b) (c) (d)

Figure 8.9 Quadtree spline-based motion estimation (Szeliski and Shum 1996) © 1996 IEEE: (a) quadtree spline representation, (b) which can lead to *cracks*, unless the white nodes are constrained to depend on their parents; (c) deformed quadtree spline mesh overlaid on grayscale image; (d) flow field visualized as a needle diagram.

Because of its robustness, spline-based motion estimation has been used for a number of applications, including visual effects (Roble 1999) and medical image registration (Section 8.3.1) (Szeliski and Lavallée 1996; Kybic and Unser 2003).

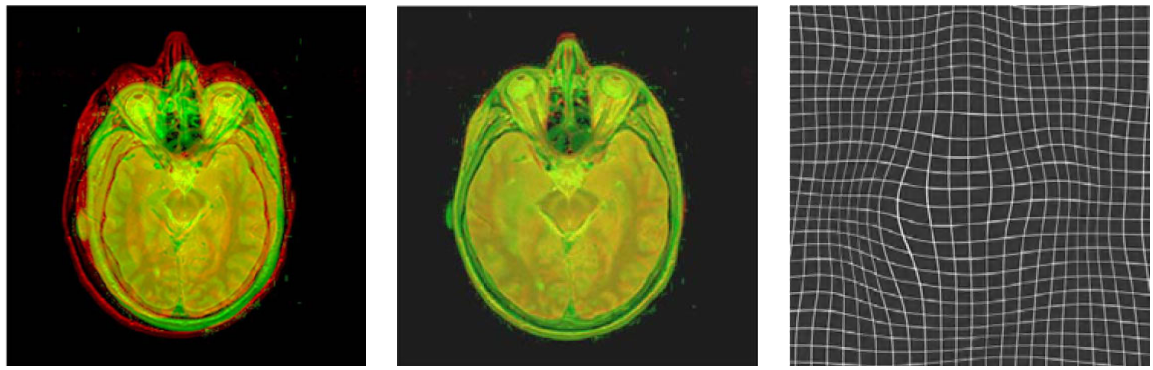
One disadvantage of the basic technique, however, is that the model does a poor job near motion discontinuities, unless an excessive number of nodes is used. To remedy this situation, Szeliski and Shum (1996) propose using a *quadtree* representation embedded in the spline control grid (Figure 8.9a). Large cells are used to present regions of smooth motion, while smaller cells are added in regions of motion discontinuities (Figure 8.9c).

To estimate the motion, a coarse-to-fine strategy is used. Starting with a regular spline imposed over a lower-resolution image, an initial motion estimate is obtained. Spline patches where the motion is inconsistent, i.e., the squared residual (8.67) is above a threshold, are subdivided into smaller patches. In order to avoid *cracks* in the resulting motion field (Figure 8.9b), the values of certain nodes in the refined mesh, i.e., those adjacent to larger cells, need to be *restricted* so that they depend on their parent values. This is most easily accomplished using a hierarchical basis representation for the quadtree spline (Szeliski 1990b) and selectively setting some of the hierarchical basis functions to 0, as described in (Szeliski and Shum 1996).

8.3.1 *Application*: Medical image registration

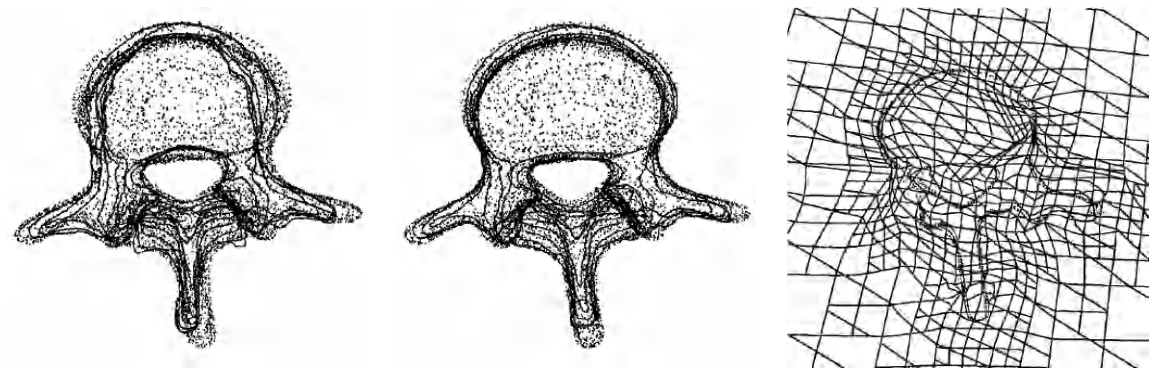
Because they excel at representing smooth *elastic* deformation fields, spline-based motion models have found widespread use in medical image registration (Bajcsy and Kovacic 1989; Szeliski and Lavallée 1996; Christensen, Joshi, and Miller 1997).[10] Registration techniques can be used both to track an individual patient's development or progress over time (a *longitudinal* study) or to match different patient images together to find commonalities and detect variations or pathologies (*cross-sectional* studies). When different imaging *modalities* are being registered, e.g., computed tomography (CT) scans and magnetic resonance images (MRI), *mutual information* measures of similarity are often necessary (Viola and Wells III 1997; Maes, Collignon, Vandermeulen *et al.* 1997).

[10] In computer graphics, such elastic volumetric deformation are known as *free-form deformations* (Sederberg and Parry 1986; Coquillart 1990; Celniker and Gossard 1991).



(a) (b) (c)

Figure 8.10 Elastic brain registration (Kybic and Unser 2003) © 2003 IEEE: (a) original brain atlas and patient MRI images overlaid in red–green; (b) after elastic registration with eight user-specified landmarks (not shown); (c) a cubic B-spline deformation field, shown as a deformed grid.



(a) (b) (c)

Figure 8.11 Octree spline-based image registration of two vertebral surface models (Szeliski and Lavallée 1996) © 1996 Springer: (a) after initial rigid alignment; (b) after elastic alignment; (c) a cross-section through the adapted octree spline deformation field.

Kybic and Unser (2003) provide a nice literature review and describe a complete working system based on representing both the images and the deformation fields as multi-resolution splines. Figure 8.10 shows an example of the Kybic and Unser system being used to register a patient's brain MRI with a labeled brain atlas image. The system can be run in a fully automatic mode but more accurate results can be obtained by locating a few key *landmarks*. More recent papers on deformable medical image registration, including performance evaluations, include (Klein, Staring, and Pluim 2007; Glocker, Komodakis, Tziritas *et al.* 2008).

As with other applications, regular volumetric splines can be enhanced using selective refinement. In the case of 3D volumetric image or surface registration, these are known as *octree splines* (Szeliski and Lavallée 1996) and have been used to register medical surface models such as vertebrae and faces from different patients (Figure 8.11).

8.4 Optical flow

The most general (and challenging) version of motion estimation is to compute an independent estimate of motion at *each* pixel, which is generally known as *optical* (or *optic*) *flow*. As we mentioned in the previous section, this generally involves minimizing the brightness or color difference between corresponding pixels summed over the image,

$$E_{\text{SSD-OF}}(\{\boldsymbol{u}_i\}) = \sum_i [I_1(\boldsymbol{x}_i + \boldsymbol{u}_i) - I_0(\boldsymbol{x}_i)]^2. \tag{8.69}$$

Since the number of variables $\{\boldsymbol{u}_i\}$ is twice the number of measurements, the problem is underconstrained. The two classic approaches to this problem are to perform the summation *locally* over overlapping regions (the *patch-based* or *window-based* approach) or to add smoothness terms on the $\{\boldsymbol{u}_i\}$ field using regularization or Markov random fields (Section 3.7) and to search for a global minimum.

The patch-based approach usually involves using a Taylor series expansion of the displaced image function (8.35) in order to obtain sub-pixel estimates (Lucas and Kanade 1981). Anandan (1989) shows how a series of local discrete search steps can be interleaved with Lucas–Kanade incremental refinement steps in a coarse-to-fine pyramid scheme, which allows the estimation of large motions, as described in Section 8.1.1. He also analyzes how the *uncertainty* in local motion estimates is related to the eigenvalues of the local Hessian matrix $\boldsymbol{A}_i$ (8.44), as shown in Figures 8.3–8.4.

Bergen, Anandan, Hanna *et al.* (1992) develop a unified framework for describing both parametric (Section 8.2) and patch-based optic flow algorithms and provide a nice introduction to this topic. After each iteration of optic flow estimation in a coarse-to-fine pyramid, they re-warp one of the images so that only incremental flow estimates are computed (Section 8.1.1). When overlapping patches are used, an efficient implementation is to first compute the outer products of the gradients and intensity errors (8.40–8.41) at every pixel and then perform the overlapping window sums using a moving average filter.[11]

Instead of solving for each motion (or motion update) independently, Horn and Schunck (1981) develop a regularization-based framework where (8.69) is simultaneously minimized over all flow vectors $\{\boldsymbol{u}_i\}$. In order to constrain the problem, smoothness constraints, i.e., squared penalties on flow derivatives, are added to the basic per-pixel error metric. Because the technique was originally developed for small motions in a variational (continuous function) framework, the linearized *brightness constancy constraint* corresponding to (8.35), i.e., (8.38), is more commonly written as an analytic integral

$$E_{\text{HS}} = \int (I_x u + I_y v + I_t)^2 \, dx \, dy, \tag{8.70}$$

where $(I_x, I_y) = \nabla I_1 = \boldsymbol{J}_1$ and $I_t = e_i$ is the *temporal derivative*, i.e., the brightness change between images. The Horn and Schunck model can also be viewed as the limiting case of spline-based motion estimation as the splines become 1x1 pixel patches.

It is also possible to combine ideas from local and global flow estimation into a single framework by using a locally aggregated (as opposed to single-pixel) Hessian as the brightness constancy term (Bruhn, Weickert, and Schnörr 2005). Consider the discrete analog

[11] Other smoothing or aggregation filters can also be used at this stage (Bruhn, Weickert, and Schnörr 2005).

(8.35) to the analytic global energy (8.70),

$$E_{\text{HSD}} = \sum_i \boldsymbol{u}_i^T [\boldsymbol{J}_i \boldsymbol{J}_i^T] \boldsymbol{u}_i + 2e_i \boldsymbol{J}_i^T \boldsymbol{u}_i + e_i^2. \tag{8.71}$$

If we replace the per-pixel (rank 1) Hessians $\boldsymbol{A}_i = [\boldsymbol{J}_i \boldsymbol{J}_i^T]$ and residuals $\boldsymbol{b}_i = \boldsymbol{J}_i e_i$ with area-aggregated versions (8.40–8.41), we obtain a global minimization algorithm where region-based brightness constraints are used.

Another extension to the basic optic flow model is to use a combination of global (parametric) and local motion models. For example, if we know that the motion is due to a camera moving in a static scene (rigid motion), we can re-formulate the problem as the estimation of a per-pixel depth along with the parameters of the global camera motion (Adiv 1989; Hanna 1991; Bergen, Anandan, Hanna *et al.* 1992; Szeliski and Coughlan 1997; Nir, Bruckstein, and Kimmel 2008; Wedel, Cremers, Pock *et al.* 2009). Such techniques are closely related to stereo matching (Chapter 11). Alternatively, we can estimate either per-image or per-segment affine motion models combined with per-pixel *residual* corrections (Black and Jepson 1996; Ju, Black, and Jepson 1996; Chang, Tekalp, and Sezan 1997; Mémin and Pérez 2002). We revisit this topic in Section 8.5.

Of course, image brightness may not always be an appropriate metric for measuring appearance consistency, e.g., when the lighting in an image is varying. As discussed in Section 8.1, matching gradients, filtered images, or other metrics such as image Hessians (second derivative measures) may be more appropriate. It is also possible to locally compute the *phase* of steerable filters in the image, which is insensitive to both bias and gain transformations (Fleet and Jepson 1990). Papenberg, Bruhn, Brox *et al.* (2006) review and explore such constraints and also provide a detailed analysis and justification for iteratively re-warping images during incremental flow computation.

Because the brightness constancy constraint is evaluated at each pixel independently, rather than being summed over patches where the constant flow assumption may be violated, global optimization approaches tend to perform better near motion discontinuities. This is especially true if robust metrics are used in the smoothness constraint (Black and Anandan 1996; Bab-Hadiashar and Suter 1998a).[12] One popular choice for robust metrics in the L_1 norm, also known as *total variation* (TV), which results in a convex energy whose global minimum can be found (Bruhn, Weickert, and Schnörr 2005; Papenberg, Bruhn, Brox *et al.* 2006). Anisotropic smoothness priors, which apply a different smoothness in the directions parallel and perpendicular to the image gradient, are another popular choice (Nagel and Enkelmann 1986; Sun, Roth, Lewis *et al.* 2008; Werlberger, Trobin, Pock *et al.* 2009). It is also possible to learn a set of better smoothness constraints (derivative filters and robust functions) from a set of paired flow and intensity images (Sun, Roth, Lewis *et al.* 2008). Additional details on some of these techniques are given by Baker, Black, Lewis *et al.* (2007) and Baker, Scharstein, Lewis *et al.* (2009).

Because of the large, two-dimensional search space in estimating flow, most algorithms use variations of gradient descent and coarse-to-fine continuation methods to minimize the global energy function. This contrasts starkly with stereo matching (which is an "easier" one-dimensional disparity estimation problem), where combinatorial optimization techniques have been the method of choice for the last decade.

[12] Robust brightness metrics (Section 8.1, (8.2)) can also help improve the performance of window-based approaches (Black and Anandan 1996).

Optical flow evaluation results

Statistics: Average SD R0.5 R1.0 R2.0 A50 A75 A95
Error type: endpoint angle interpolation normalized interpolation

Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Adaptive [20]	4.4	**0.09** 1	**0.26** 1	**0.06** 1	0.23 5	0.78 4	0.18 8	0.54 8	1.7 10	0.21 3	**0.18** 1	0.91 3	**0.10** 1	0.88 3	1.25 3	0.73 5	0.50 3	1.28 3	0.31 3	0.14 10	0.16 12	0.22 10	0.65 3	1.37 3	0.79 4
Complementary OF [21]	5.7	0.11 5	0.28 3	0.10 9	**0.18** 1	**0.63** 1	**0.12** 1	0.31 3	0.75	**0.18** 1	0.19 2	0.97 5	0.12 3	0.97 10	1.31 6	1.00 11	1.78 20	1.73 7	0.87 14	0.11 4	0.12 2	0.22 10	0.68 4	1.48 4	0.95 8
Aniso. Huber-L1 [22]	5.8	0.10 3	0.28 3	0.08 3	0.31 11	0.88	0.28 12	0.4	1.13	0.29 12	0.20 4	0.92 4	0.13 5	0.84 2	1.20 2	0.70 2	**0.39** 1	**1.23** 1	**0.28** 1	0.17 15	0.15 9	0.27 16	0.64 2	1.36 2	0.79 4
DPOF [18]	6.1	0.13 12	0.35 12	0.09 4	0.25 6	0.79 5	0.1	**0.24**	0.49	0.21 3	0.19 2	**0.62** 1	0.15 11	**0.74** 1	**1.09** 1	**0.49** 1	0.66 7	1.80 10	0.63 8	0.19 17	0.17 14	0.35 20	**0.50** 1	**1.08** 1	**0.55** 1
TV-L1-improved [17]	7.2	**0.09** 1	**0.26** 1	0.07 2	0.20 3	0.71 3	0.16 2	0.53 7	1.18 9	0.22 5	0.21 7	1.24 11	0.11 2	0.90 4	1.31 6	0.72 3	1.51 14	1.93 11	0.84 11	0.18 16	0.17 14	0.31 17	0.73 8	1.62 9	0.87 7
CBF [12]	7.8	0.10 3	0.28 3	0.09 4	0.34 12	0.80 6	0.37 13	0.43 5	0.95 5	0.26 8	0.21 7	1.14 8	0.13 5	0.90 4	1.27 4	0.82 7	0.41 2	**1.23** 1	0.30 2	0.23 22	0.19 20	0.39 21	0.76 9	1.56 6	1.02 9
Brox et al. [5]	8.4	0.11 5	0.32 8	0.11 12	0.27 9	0.93 10	0.22 9	0.39 4	0.94 4	0.24 7	0.24 9	1.25 12	0.13 5	1.10 13	1.39 12	1.43 17	0.89 8	1.77 8	0.55 7	0.10 2	0.13 4	**0.11** 1	0.91 11	1.83 12	1.13 12
Rannacher [23]	8.5	0.11 5	0.31 6	0.09 4	0.25 6	0.84 7	0.21 8	0.57 12	1.27 15	0.26 8	0.24 9	1.32 14	0.13 5	0.91 7	1.33 8	0.72 3	1.49 13	1.95 13	0.78 9	0.15 12	0.14 7	0.26 13	0.69 6	1.58 8	0.86 6
F-TV-L1 [15]	8.8	0.14 13	0.35 12	0.14 15	0.34 12	0.98 12	0.26 11	0.59 14	1.19 10	0.26 8	0.27 13	1.36 15	0.16 12	0.90 4	1.30 5	0.76 6	0.54 4	1.62 6	0.36 4	0.13 6	0.15 9	0.20 9	0.68 4	1.56 6	0.66 2
Second-order prior [8]	9.0	0.11 5	0.31 6	0.09 4	0.26 8	0.93 10	0.20 7	0.57 12	1.25 14	0.26 8	0.20 4	1.04 6	0.12 3	0.94 8	1.34 9	0.83 9	0.61 6	1.93 11	0.47 6	0.20 18	0.16 12	0.34 19	0.77 10	1.64 10	1.07 10
Fusion [6]	9.4	0.11 5	0.34 10	0.10 9	0.19 2	0.69 2	0.16 2	0.29 2	0.66 2	0.23 6	0.20 4	1.19 10	0.14 9	1.07 11	1.42 13	1.22 13	1.35 10	1.49 5	0.86 13	0.20 18	0.20 21	0.26 13	1.07 14	2.07 16	1.39 16
Dynamic MRF [7]	11.1	0.12 11	0.34 10	0.11 12	0.22 4	0.89 9	0.16 2	0.44 6	1.13 7	0.20 2	0.24 9	1.29 13	0.14 9	1.11 14	1.52 17	1.13 12	1.54 15	2.37 20	0.93 15	0.13 6	0.12 2	0.31 17	1.27 18	2.33 20	1.66 17
SegOF [10]	11.7	0.15 14	0.36 14	0.10 9	0.57 15	1.16 15	0.59 19	0.68 15	1.24 12	0.64 14	0.32 15	0.86 2	0.26 15	1.18 17	1.50 16	1.47 18	1.63 18	2.09 14	0.96 16	**0.08** 1	0.13 4	0.12 2	0.70 7	1.50 5	0.69 3
Learning Flow [11]	13.3	0.11 5	0.32 8	0.09 4	0.29 10	0.99 13	0.23 10	0.55 9	1.24 12	0.29 12	0.36 16	1.56 17	0.25 14	1.25 19	1.64 21	1.41 16	1.55 17	2.32 19	0.85 12	0.14 10	0.18 18	0.24 12	1.09 15	2.09 18	1.27 13
Filter Flow [19]	14.3	0.17 16	0.39 16	0.13 14	0.43 14	1.09 14	0.38 14	0.75 16	1.34 16	0.78 19	0.70 19	1.54 16	0.68 19	1.13 16	1.38 11	1.51 19	0.57 5	1.32 4	0.44 5	0.22 20	0.23 23	0.26 13	0.96 12	1.66 11	1.12 11
GraphCuts [14]	14.5	0.16 15	0.38 15	0.14 15	0.59 18	1.36 19	0.46 15	0.56 10	1.07 6	0.64 14	0.26 12	1.14 8	0.17 13	0.96 9	1.35 10	0.84 10	2.25 23	1.79 9	1.22 21	0.22 20	0.17 14	0.43 22	1.22 17	2.05 15	1.78 19
Black & Anandan [4]	15.0	0.18 17	0.42 17	0.19 18	0.58 17	1.31 17	0.50 16	0.95 19	1.58 18	0.70 16	0.49 17	1.59 18	0.45 17	1.08 12	1.42 13	1.22 13	1.43 11	2.28 17	0.83 10	0.15 12	0.17 14	0.17 6	1.11 16	1.98 14	1.30 14
SPSA-learn [13]	15.7	0.18 17	0.45 18	0.17 17	0.57 15	1.32 18	0.51 17	0.84 17	1.50 17	0.72 17	0.52 18	1.64 19	0.49 18	1.12 15	1.42 13	1.39 15	1.75 19	2.14 15	1.06 20	0.13 6	0.13 4	0.19 7	1.32 19	2.08 17	1.73 18
GroupFlow [9]	15.9	0.21 19	0.51 19	0.21 19	0.79 21	1.69 21	0.72 21	0.86 18	1.64 19	0.74 18	0.30 14	1.07 7	0.26 15	1.29 22	1.81 22	0.82 7	1.94 21	2.30 18	1.36 22	0.11 4	0.14 7	0.19 7	1.06 13	1.96 13	1.35 15
2D-CLG [1]	17.4	0.28 21	0.62 22	0.21 19	0.67 20	1.21 16	0.70 20	1.12 21	1.80 21	0.99 22	1.07 22	2.06 21	1.12 22	1.23 18	1.52 17	1.62 22	1.54 15	2.15 16	0.96 16	0.10 2	**0.11** 1	0.16 4	1.38 20	2.26 19	1.83 20
Horn & Schunck [3]	18.6	0.22 20	0.55 20	0.22 21	0.61 19	1.53 20	0.52 18	1.01 20	1.73 20	0.80 20	0.78 20	2.02 20	0.77 20	1.26 20	1.58 19	1.55 20	1.43 11	2.59 22	1.00 18	0.16 14	0.18 18	0.15 3	1.51 21	2.50 21	1.88 21
TI-DOFE [24]	19.6	0.38 23	0.64 23	0.47 23	1.16 22	1.72 22	1.26 22	1.39 23	2.06 24	1.17 23	1.29 23	2.21 23	1.41 23	1.27 21	1.61 20	1.57 21	1.28 9	2.57 21	1.01 19	0.13 6	0.15 9	0.16 4	1.87 22	2.71 22	2.53 22
FOLKI [16]	22.6	0.29 22	0.73 24	0.33 22	1.52 23	1.96 24	1.80 23	1.23 22	2.04 23	0.95 21	0.99 21	2.20 22	1.08 21	1.53 23	1.85 23	2.07 23	2.14 22	3.23 24	1.60 23	0.26 23	0.21 22	0.68 23	2.67 23	3.27 23	4.32 23
Pyramid LK [2]	23.7	0.39 24	0.61 21	0.61 24	1.67 24	1.78 23	2.00 24	1.50 24	1.97 22	1.38 24	1.57 24	2.39 24	1.78 24	2.94 24	3.72 24	2.98 24	3.33 24	2.74 23	2.43 24	0.30 24	0.24 24	0.73 24	3.80 24	5.08 24	4.88 24

Move the mouse over the numbers in the table to see the corresponding images. Click to compare with the ground truth.

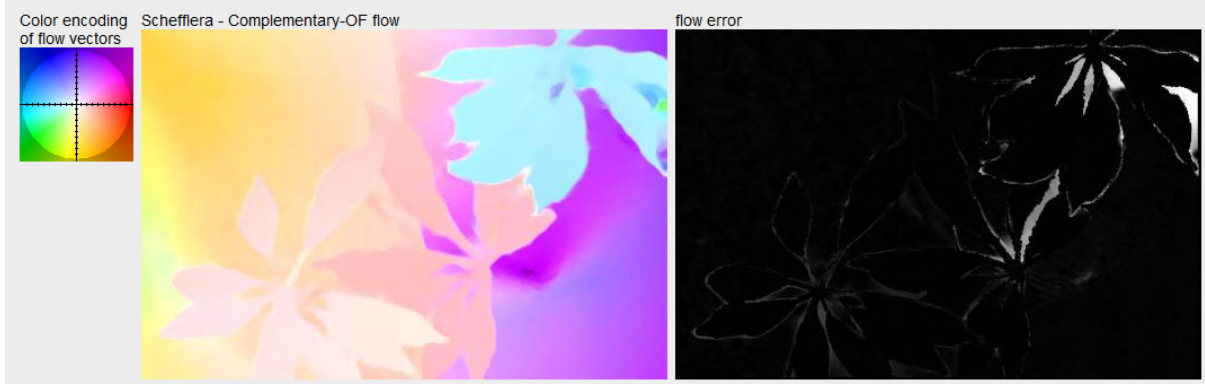


Figure 8.12 Evaluation of the results of 24 optical flow algorithms, October 2009, http://vision.middlebury.edu/flow/, (Baker, Scharstein, Lewis *et al.* 2009). By moving the mouse pointer over an underlined performance score, the user can interactively view the corresponding flow and error maps. Clicking on a score toggles between the computed and ground truth flows. Next to each score, the corresponding rank in the current column is indicated by a smaller blue number. The minimum (best) score in each column is shown in boldface. The table is sorted by the average rank (computed over all 24 columns, three region masks for each of the eight sequences). The average rank serves as an *approximate* measure of performance *under the selected metric/statistic*.

Fortunately, combinatorial optimization methods based on Markov random fields are beginning to appear and tend to be among the better-performing methods on the recently released optical flow database (Baker, Black, Lewis *et al.* 2007).[13]

Examples of such techniques include the one developed by Glocker, Paragios, Komodakis *et al.* (2008), who use a coarse-to-fine strategy with per-pixel 2D uncertainty estimates, which are then used to guide the refinement and search at the next finer level. Instead of using gradient descent to refine the flow estimates, a combinatorial search over discrete displacement labels (which is able to find better energy minima) is performed using their Fast-PD algorithm (Komodakis, Tziritas, and Paragios 2008).

Lempitsky, Roth, and Rother. (2008) use fusion moves (Lempitsky, Rother, and Blake 2007) over proposals generated from basic flow algorithms (Horn and Schunck 1981; Lucas and Kanade 1981) to find good solutions. The basic idea behind fusion moves is to replace portions of the current best estimate with hypotheses generated by more basic techniques (or their shifted versions) and to alternate them with local gradient descent for better energy minimization.

The field of accurate motion estimation continues to evolve at a rapid pace, with significant advances in performance occurring every year. The optical flow evaluation Web site (http://vision.middlebury.edu/flow/) is a good source of pointers to high-performing recently developed algorithms (Figure 8.12).

8.4.1 Multi-frame motion estimation

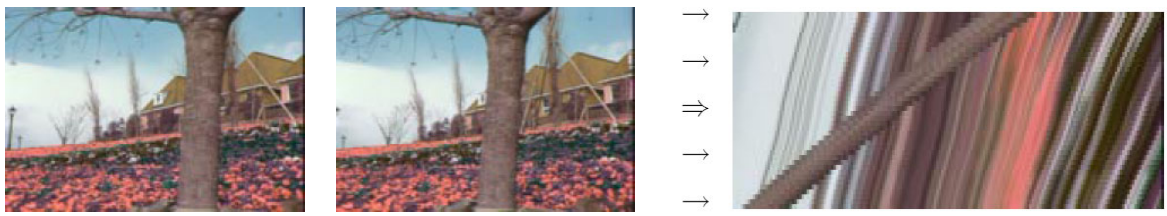
So far, we have looked at motion estimation as a two-frame problem, where the goal is to compute a motion field that aligns pixels from one image with those in another. In practice, motion estimation is usually applied to video, where a whole sequence of frames is available to perform this task.

One classic approach to multi-frame motion is to *filter* the spatio-temporal volume using oriented or steerable filters (Heeger 1988), in a manner analogous to oriented edge detection (Section 3.2.3). Figure 8.13 shows two frames from the commonly used *flower garden* sequence, as well as a horizontal slice through the spatio-temporal volume, i.e., the 3D volume created by stacking all of the video frames together. Because the pixel motion is mostly horizontal, the slopes of individual (textured) pixel tracks, which correspond to their horizontal velocities, can clearly be seen. Spatio-temporal filtering uses a 3D volume around each pixel to determine the best orientation in space–time, which corresponds directly to a pixel's velocity.

Unfortunately, in order to obtain reasonably accurate velocity estimates everywhere in an image, spatio-temporal filters have moderately large extents, which severely degrades the quality of their estimates near motion discontinuities. (This same problem is endemic in 2D window-based motion estimators.) An alternative to full spatio-temporal filtering is to estimate more local spatio-temporal derivatives and use them inside a global optimization framework to fill in textureless regions (Bruhn, Weickert, and Schnörr 2005; Govindu 2006).

Another alternative is to simultaneously estimate multiple motion estimates, while also optionally reasoning about occlusion relationships (Szeliski 1999). Figure 8.13c shows schematically one potential approach to this problem. The horizontal arrows show the locations of

[13] http://vision.middlebury.edu/flow/.



(a) (b) (c)

Figure 8.13 Slice through a spatio-temporal volume (Szeliski 1999) © 1999 IEEE: (a–b) two frames from the *flower garden* sequence; (c) a horizontal slice through the complete spatio-temporal volume, with the arrows indicating locations of potential key frames where flow is estimated. Note that the colors for the flower garden sequence are incorrect; the correct colors (yellow flowers) are shown in Figure 8.15.

keyframes s where motion is estimated, while other slices indicate video frames t whose colors are matched with those predicted by interpolating between the keyframes. Motion estimation can be cast as a global energy minimization problem that simultaneously minimizes brightness compatibility and flow compatibility terms between keyframes and other frames, in addition to using robust smoothness terms.

The multi-view framework is potentially even more appropriate for rigid scene motion (multi-view stereo) (Section 11.6), where the unknowns at each pixel are disparities and occlusion relationships can be determined directly from pixel depths (Szeliski 1999; Kolmogorov and Zabih 2002). However, it may also be applicable to general motion, with the addition of models for object accelerations and occlusion relationships.

8.4.2 *Application*: Video denoising

Video denoising is the process of removing noise and other artifacts such as scratches from film and video (Kokaram 2004). Unlike single image denoising, where the only information available is in the current picture, video denoisers can average or borrow information from adjacent frames. However, in order to do this without introducing blur or jitter (irregular motion), they need accurate per-pixel motion estimates.

Exercise 8.7 lists some of the steps required, which include the ability to determine if the current motion estimate is accurate enough to permit averaging with other frames. Gai and Kang (2009) describe their recently developed restoration process, which involves a series of additional steps to deal with the special characteristics of vintage film.

8.4.3 *Application*: De-interlacing

Another commonly used application of per-pixel motion estimation is video de-interlacing, which is the process of converting a video taken with alternating fields of even and odd lines to a non-interlaced signal that contains both fields in each frame (de Haan and Bellers 1998). Two simple de-interlacing techniques are *bob*, which copies the line above or below the missing line from the same field, and *weave*, which copies the corresponding line from the field before or after. The names come from the visual artifacts generated by these two simple techniques: bob introduces an up-and-down bobbing motion along strong horizontal

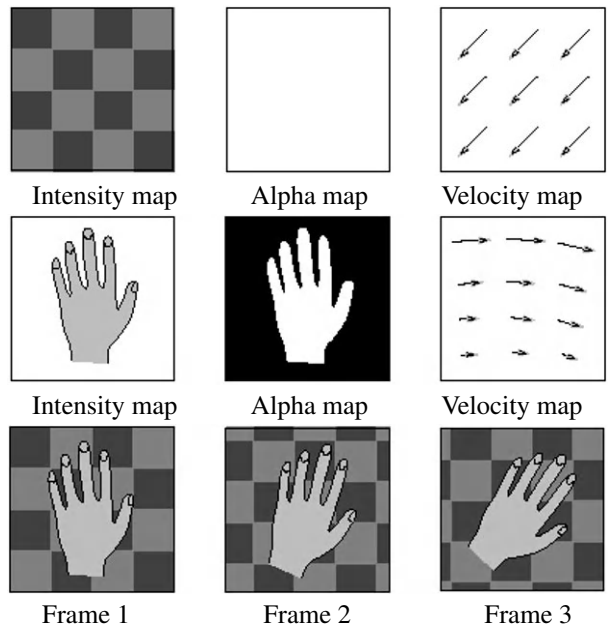


Figure 8.14 Layered motion estimation framework (Wang and Adelson 1994) © 1994 IEEE: The top two rows describe the two layers, each of which consists of an intensity (color) image, an alpha mask (black=transparent), and a parametric motion field. The layers are composited with different amounts of motion to recreate the video sequence.

lines; weave can lead to a "zippering" effect along horizontally translating edges. Replacing these copy operators with averages can help but does not completely remove these artifacts.

A wide variety of improved techniques have been developed for this process, which is often embedded in specialized DSP chips found inside video digitization boards in computers (since broadcast video is often interlaced, while computer monitors are not). A large class of these techniques estimates local per-pixel motions and interpolates the missing data from the information available in spatially and temporally adjacent fields. Dai, Baker, and Kang (2009) review this literature and propose their own algorithm, which selects among seven different interpolation functions at each pixel using an MRF framework.

8.5 Layered motion

In many situation, visual motion is caused by the movement of a small number of objects at different depths in the scene. In such situations, the pixel motions can be described more succinctly (and estimated more reliably) if pixels are grouped into appropriate objects or *layers* (Wang and Adelson 1994).

Figure 8.14 shows this approach schematically. The motion in this sequence is caused by the translational motion of the checkered background and the rotation of the foreground hand. The complete motion sequence can be reconstructed from the appearance of the foreground and background elements, which can be represented as alpha-matted images (*sprites* or *video objects*) and the parametric motion corresponding to each layer. Displacing and compositing

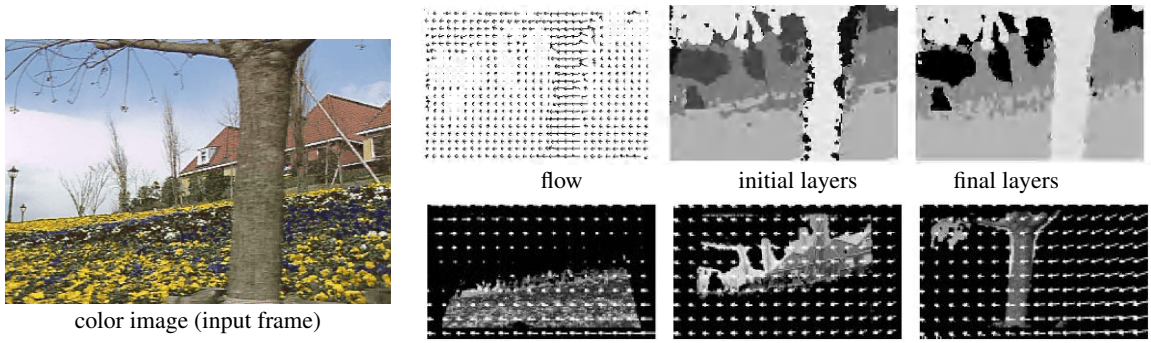


Figure 8.15 Layered motion estimation results (Wang and Adelson 1994) © 1994 IEEE.

these layers in back to front order (Section 3.1.3) recreates the original video sequence.

Layered motion representations not only lead to compact representations (Wang and Adelson 1994; Lee, ge Chen, lung Bruce Lin *et al.* 1997), but they also exploit the information available in multiple video frames, as well as accurately modeling the appearance of pixels near motion discontinuities. This makes them particularly suited as a representation for image-based rendering (Section 13.2.1) (Shade, Gortler, He *et al.* 1998; Zitnick, Kang, Uyttendaele *et al.* 2004) as well as object-level video editing.

To compute a layered representation of a video sequence, Wang and Adelson (1994) first estimate affine motion models over a collection of non-overlapping patches and then cluster these estimates using k-means. They then alternate between assigning pixels to layers and recomputing motion estimates for each layer using the assigned pixels, using a technique first proposed by Darrell and Pentland (1991). Once the parametric motions and pixel-wise layer assignments have been computed for each frame independently, layers are constructed by warping and merging the various layer pieces from all of the frames together. Median filtering is used to produce sharp composite layers that are robust to small intensity variations, as well as to infer occlusion relationships between the layers. Figure 8.15 shows the results of this process on the *flower garden* sequence. You can see both the initial and final layer assignments for one of the frames, as well as the composite flow and the alpha-matted layers with their corresponding flow vectors overlaid.

In follow-on work, Weiss and Adelson (1996) use a formal probabilistic mixture model to infer both the optimal number of layers and the per-pixel layer assignments. Weiss (1997) further generalizes this approach by replacing the per-layer affine motion models with smooth regularized per-pixel motion estimates, which allows the system to better handle curved and undulating layers, such as those seen in most real-world sequences.

The above approaches, however, still make a distinction between estimating the motions and layer assignments and then later estimating the layer colors. In the system described by Baker, Szeliski, and Anandan (1998), the generative model illustrated in Figure 8.14 is generalized to account for real-world rigid motion scenes. The motion of each frame is described using a 3D camera model and the motion of each layer is described using a 3D plane equation plus per-pixel residual depth offsets (the *plane plus parallax* representation (Section 2.1.5)). The initial layer estimation proceeds in a manner similar to that of Wang and Adelson (1994),

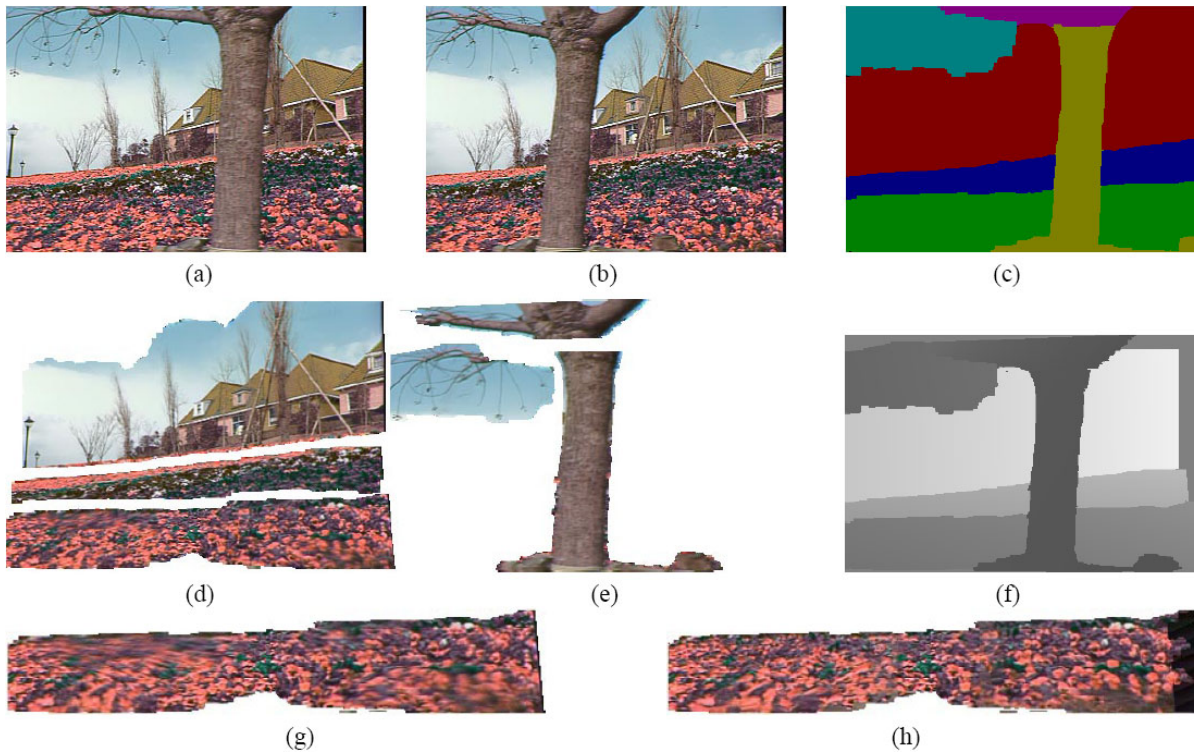


Figure 8.16 Layered stereo reconstruction (Baker, Szeliski, and Anandan 1998) © 1998 IEEE: (a) first and (b) last input images; (c) initial segmentation into six layers; (d) and (e) the six layer sprites; (f) depth map for planar sprites (darker denotes closer); front layer (g) before and (h) after residual depth estimation. Note that the colors for the flower garden sequence are incorrect; the correct colors (yellow flowers) are shown in Figure 8.15.

o

except that rigid planar motions (homographies) are used instead of affine motion models. The final model refinement, however, jointly re-optimizes the layer pixel color and opacity values L_l and the 3D depth, plane, and motion parameters z_l, $\boldsymbol{n}_l$, and $\boldsymbol{P}_t$ by minimizing the discrepancy between the re-synthesized and observed motion sequences (Baker, Szeliski, and Anandan 1998).

Figure 8.16 shows the final results obtained with this algorithm. As you can see, the motion boundaries and layer assignments are much crisper than those in Figure 8.15. Because of the per-pixel depth offsets, the individual layer color values are also sharper than those obtained with affine or planar motion models. While the original system of Baker, Szeliski, and Anandan (1998) required a rough initial assignment of pixels to layers, Torr, Szeliski, and Anandan (2001) describe automated Bayesian techniques for initializing this system and determining the optimal number of layers.

Layered motion estimation continues to be an active area of research. Representative papers in this area include (Sawhney and Ayer 1996; Jojic and Frey 2001; Xiao and Shah 2005; Kumar, Torr, and Zisserman 2008; Thayananthan, Iwasaki, and Cipolla 2008; Schoenemann and Cremers 2008).

Of course, layers are not the only way to introduce segmentation into motion estimation.

A large number of algorithms have been developed that alternate between estimating optic flow vectors and segmenting them into coherent regions (Black and Jepson 1996; Ju, Black, and Jepson 1996; Chang, Tekalp, and Sezan 1997; Mémin and Pérez 2002; Cremers and Soatto 2005). Some of the more recent techniques rely on first segmenting the input color images and then estimating per-segment motions that produce a coherent motion field while also modeling occlusions (Zitnick, Kang, Uyttendaele *et al.* 2004; Zitnick, Jojic, and Kang 2005; Stein, Hoiem, and Hebert 2007; Thayananthan, Iwasaki, and Cipolla 2008).

8.5.1 *Application*: Frame interpolation

Frame interpolation is another widely used application of motion estimation, often implemented in the same circuitry as de-interlacing hardware required to match an incoming video to a monitor's actual refresh rate. As with de-interlacing, information from novel in-between frames needs to be interpolated from preceding and subsequent frames. The best results can be obtained if an accurate motion estimate can be computed at each unknown pixel's location. However, in addition to computing the motion, occlusion information is critical to prevent colors from being contaminated by moving foreground objects that might obscure a particular pixel in a preceding or subsequent frame.

In a little more detail, consider Figure 8.13c and assume that the arrows denote keyframes between which we wish to interpolate additional images. The orientations of the streaks in this figure encode the velocities of individual pixels. If the same motion estimate $\boldsymbol{u}_0$ is obtained at location $\boldsymbol{x}_0$ in image I_0 as is obtained at location $\boldsymbol{x}_0 + \boldsymbol{u}_0$ in image I_1, the flow vectors are said to be *consistent*. This motion estimate can be transferred to location $\boldsymbol{x}_0 + t\boldsymbol{u}_0$ in the image I_t being generated, where $t \in (0, 1)$ is the time of interpolation. The final color value at pixel $\boldsymbol{x}_0 + t\boldsymbol{u}_0$ can be computed as a linear blend,

$$I_t(\boldsymbol{x}_0 + t\boldsymbol{u}_0) = (1 - t)I_0(\boldsymbol{x}_0) + tI_1(\boldsymbol{x}_0 + \boldsymbol{u}_0). \tag{8.72}$$

If, however, the motion vectors are different at corresponding locations, some method must be used to determine which is correct and which image contains colors that are occluded. The actual reasoning is even more subtle than this. One example of such an interpolation algorithm, based on earlier work in depth map interpolation (Shade, Gortler, He *et al.* 1998; Zitnick, Kang, Uyttendaele *et al.* 2004) which is the one used in the flow evaluation paper of Baker, Black, Lewis *et al.* (2007); Baker, Scharstein, Lewis *et al.* (2009). An even higher-quality frame interpolation algorithm, which uses gradient-based reconstruction, is presented by Mahajan, Huang, Matusik *et al.* (2009).

8.5.2 Transparent layers and reflections

A special case of layered motion that occurs quite often is transparent motion, which is usually caused by reflections seen in windows and picture frames (Figures 8.17 and 8.18).

Some of the early work in this area handles transparent motion by either just estimating the component motions (Shizawa and Mase 1991; Bergen, Burt, Hingorani *et al.* 1992; Darrell and Simoncelli 1993; Irani, Rousso, and Peleg 1994) or by assigning individual pixels to competing motion layers (Darrell and Pentland 1995; Black and Anandan 1996; Ju, Black, and Jepson 1996), which is appropriate for scenes partially seen through a fine occluder (e.g., foliage). However, to accurately separate truly transparent layers, a better model for



Figure 8.17 Light reflecting off the transparent glass of a picture frame: (a) first image from the input sequence; (b) dominant motion layer *min-composite*; (c) secondary motion residual layer *max-composite*; (d–e) final estimated picture and reflection layers The original images are from Black and Anandan (1996), while the separated layers are from Szeliski, Avidan, and Anandan (2000) © 2000 IEEE.

motion due to reflections is required. Because of the way that light is both reflected from and transmitted through a glass surface, the correct model for reflections is an *additive* one, where each moving layer contributes some intensity to the final image (Szeliski, Avidan, and Anandan 2000).

If the motions of the individual layers are known, the recovery of the individual layers is a simple constrained least squares problem, with the individual layer images are constrained to be positive. However, this problem can suffer from extended low-frequency ambiguities, especially if either of the layers lacks dark (black) pixels or the motion is uni-directional. In their paper, Szeliski, Avidan, and Anandan (2000) show that the simultaneous estimation of the motions and layer values can be obtained by alternating between robustly computing the motion layers and then making conservative (upper- or lower-bound) estimates of the layer intensities. The final motion and layer estimates can then be polished using gradient descent on a joint constrained least squares formulation similar to (Baker, Szeliski, and Anandan 1998), where the *over* compositing operator is replaced with addition.

Figures 8.17 and 8.18 show the results of applying these techniques to two different picture frames with reflections. Notice how, in the second sequence, the amount of reflected light is quite low compared to the transmitted light (the picture of the girl) and yet the algorithm is still able to recover both layers.

Unfortunately, the simple parametric motion models used in (Szeliski, Avidan, and Anandan 2000) are only valid for planar reflectors and scenes with shallow depth. The extension of these techniques to curved reflectors and scenes with significant depth has also been studied (Swaminathan, Kang, Szeliski *et al.* 2002; Criminisi, Kang, Swaminathan *et al.* 2005), as has the extension to scenes with more complex 3D depth (Tsin, Kang, and Szeliski 2006).

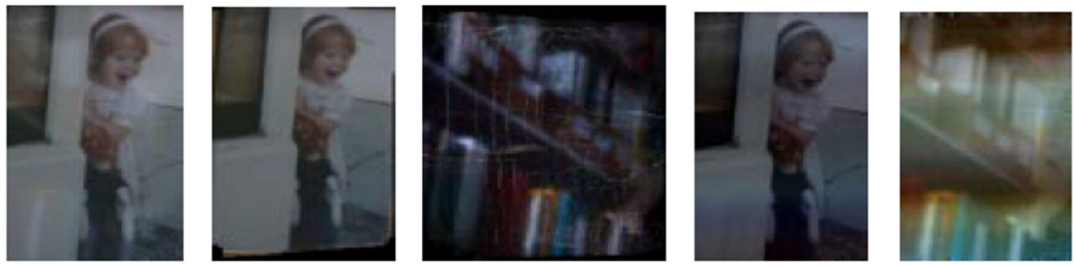


Figure 8.18 Transparent motion separation (Szeliski, Avidan, and Anandan 2000) © 2000 IEEE: (a) first image from input sequence; (b) dominant motion layer *min-composite*; (c) secondary motion residual layer *max-composite*; (d–e) final estimated picture and reflection layers. Note that the reflected layers in (c) and (e) are doubled in intensity to better show their structure.

8.6 Additional reading

Some of the earliest algorithms for motion estimation were developed for motion-compensated video coding (Netravali and Robbins 1979) and such techniques continue to be used in modern coding standards such as MPEG, H.263, and H.264 (Le Gall 1991; Richardson 2003).[14] In computer vision, this field was originally called *image sequence analysis* (Huang 1981). Some of the early seminal papers include the variational approaches developed by Horn and Schunck (1981) and Nagel and Enkelmann (1986), and the patch-based translational alignment technique developed by Lucas and Kanade (1981). Hierarchical (coarse-to-fine) versions of such algorithms were developed by Quam (1984), Anandan (1989), and Bergen, Anandan, Hanna *et al.* (1992), although they have also long been used in motion estimation for video coding.

Translational motion models were generalized to affine motion by Rehg and Witkin (1991), Fuh and Maragos (1991), and Bergen, Anandan, Hanna *et al.* (1992) and to quadric reference surfaces by Shashua and Toelg (1997) and Shashua and Wexler (2001)—see Baker and Matthews (2004) for a nice review. Such parametric motion estimation algorithms have found widespread application in video summarization (Teodosio and Bender 1993; Irani and Anandan 1998), video stabilization (Hansen, Anandan, Dana *et al.* 1994; Srinivasan, Chellappa, Veeraraghavan *et al.* 2005; Matsushita, Ofek, Ge *et al.* 2006), and video compression (Irani, Hsu, and Anandan 1995; Lee, ge Chen, lung Bruce Lin *et al.* 1997). Surveys of parametric image registration include those by Brown (1992), Zitov'aa and Flusser (2003), Goshtasby (2005), and Szeliski (2006a).

Good general surveys and comparisons of optic flow algorithms include those by Aggarwal and Nandhakumar (1988), Barron, Fleet, and Beauchemin (1994), Otte and Nagel (1994), Mitiche and Bouthemy (1996), Stiller and Konrad (1999), McCane, Novins, Crannitch *et al.* (2001), Szeliski (2006a), and Baker, Black, Lewis *et al.* (2007). The topic of matching primitives, i.e., pre-transforming images using filtering or other techniques before matching, is treated in a number of papers (Anandan 1989; Bergen, Anandan, Hanna *et al.* 1992; Scharstein 1994; Zabih and Woodfill 1994; Cox, Roy, and Hingorani 1995; Viola and

[14] http://www.itu.int/rec/T-REC-H.264.

Wells III 1997; Negahdaripour 1998; Kim, Kolmogorov, and Zabih 2003; Jia and Tang 2003; Papenberg, Bruhn, Brox *et al.* 2006; Seitz and Baker 2009). Hirschmüller and Scharstein (2009) compare a number of these approaches and report on their relative performance in scenes with exposure differences.

The publication of a new benchmark for evaluating optical flow algorithms (Baker, Black, Lewis *et al.* 2007) has led to rapid advances in the quality of estimation algorithms, to the point where new datasets may soon become necessary. According to their updated technical report (Baker, Scharstein, Lewis *et al.* 2009), most of the best performing algorithms use robust data and smoothness norms (often L_1 TV) and continuous variational optimization techniques, although some techniques use discrete optimization or segmentations (Papenberg, Bruhn, Brox *et al.* 2006; Trobin, Pock, Cremers *et al.* 2008; Xu, Chen, and Jia 2008; Lempitsky, Roth, and Rother. 2008; Werlberger, Trobin, Pock *et al.* 2009; Lei and Yang 2009; Wedel, Cremers, Pock *et al.* 2009).

8.7 Exercises

Ex 8.1: Correlation Implement and compare the performance of the following correlation algorithms:

- sum of squared differences (8.1)
- sum of robust differences (8.2)
- sum of absolute differences (8.3)
- bias–gain compensated squared differences (8.9)
- normalized cross-correlation (8.11)
- windowed versions of the above (8.22–8.23)
- Fourier-based implementations of the above measures (8.18–8.20)
- phase correlation (8.24)
- gradient cross-correlation (Argyriou and Vlachos 2003).

Compare a few of your algorithms on different motion sequences with different amounts of noise, exposure variation, occlusion, and frequency variations (e.g., high-frequency textures, such as sand or cloth, and low-frequency images, such as clouds or motion-blurred video). Some datasets with illumination variation and ground truth correspondences (horizontal motion) can be found at http://vision.middlebury.edu/stereo/data/ (the 2005 and 2006 datasets).

Some additional ideas, variants, and questions:

1. When do you think that phase correlation will outperform regular correlation or SSD? Can you show this experimentally or justify it analytically?
2. For the Fourier-based masked or windowed correlation and sum of squared differences, the results should be the same as the direct implementations. Note that you will have to expand (8.5) into a sum of pairwise correlations, just as in (8.22). (This is part of the exercise.)

3. For the bias–gain corrected variant of squared differences (8.9), you will also have to expand the terms to end up with a 3×3 (least squares) system of equations. If implementing the Fast Fourier Transform version, you will need to figure out how all of these entries can be evaluated in the Fourier domain.

4. (Optional) Implement some of the additional techniques studied by Hirschmüller and Scharstein (2009) and see if your results agree with theirs.

Ex 8.2: Affine registration Implement a coarse-to-fine direct method for affine and projective image alignment.

1. Does it help to use lower-order (simpler) models at coarser levels of the pyramid (Bergen, Anandan, Hanna *et al.* 1992)?

2. (Optional) Implement patch-based acceleration (Shum and Szeliski 2000; Baker and Matthews 2004).

3. See the Baker and Matthews (2004) survey for more comparisons and ideas.

Ex 8.3: Stabilization Write a program to *stabilize* an input video sequence. You should implement the following steps, as described in Section 8.2.1:

1. Compute the translation (and, optionally, rotation) between successive frames with robust outlier rejection.

2. Perform temporal high-pass filtering on the motion parameters to remove the low-frequency component (smooth the motion).

3. Compensate for the high-frequency motion, zooming in slightly (a user-specified amount) to avoid missing edge pixels.

4. (Optional) Do not zoom in, but instead borrow pixels from previous or subsequent frames to fill in.

5. (Optional) Compensate for images that are blurry because of fast motion by "stealing" higher frequencies from adjacent frames.

Ex 8.4: Optical flow Compute optical flow (spline-based or per-pixel) between two images, using one or more of the techniques described in this chapter.

1. Test your algorithms on the motion sequences available at http://vision.middlebury.edu/flow/ or http://people.csail.mit.edu/celiu/motionAnnotation/ and compare your results (visually) to those available on these Web sites. If you think your algorithm is competitive with the best, consider submitting it for formal evaluation.

2. Visualize the quality of your results by generating in-between images using frame interpolation (Exercise 8.5).

3. What can you say about the relative efficiency (speed) of your approach?

Ex 8.5: Automated morphing / frame interpolation Write a program to automatically morph between pairs of images. Implement the following steps, as sketched out in Section 8.5.1 and by Baker, Scharstein, Lewis *et al.* (2009):

1. Compute the flow both ways (previous exercise). Consider using a multi-frame ($n > 2$) technique to better deal with occluded regions.
2. For each intermediate (morphed) image, compute a set of flow vectors and which images should be used in the final composition.
3. Blend (cross-dissolve) the images and view with a sequence viewer.

Try this out on images of your friends and colleagues and see what kinds of morphs you get. Alternatively, take a video sequence and do a high-quality slow-motion effect. Compare your algorithm with simple cross-fading.

Ex 8.6: Motion-based user interaction Write a program to compute a low-resolution motion field in order to interactively control a simple application (Cutler and Turk 1998). For example:

1. Downsample each image using a pyramid and compute the optical flow (spline-based or pixel-based) from the previous frame.
2. Segment each training video sequence into different "actions" (e.g., hand moving inwards, moving up, no motion) and "learn" the velocity fields associated with each one. (You can simply find the mean and variance for each motion field or use something more sophisticated, such as a support vector machine (SVM).)
3. Write a recognizer that finds successive actions of approximately the right duration and hook it up to an interactive application (e.g., a sound generator or a computer game).
4. Ask your friends to test it out.

Ex 8.7: Video denoising Implement the algorithm sketched in Application 8.4.2. Your algorithm should contain the following steps:

1. Compute accurate per-pixel flow.
2. Determine which pixels in the reference image have good matches with other frames.
3. Either average all of the matched pixels or choose the sharpest image, if trying to compensate for blur. Don't forget to use regular single-frame denoising techniques as part of your solution, (see Section 3.4.4, Section 3.7.3, and Exercise 3.11).
4. Devise a fall-back strategy for areas where you don't think the flow estimates are accurate enough.

Ex 8.8: Motion segmentation Write a program to segment an image into separately moving regions or to reliably find motion boundaries.

Use the human-assisted motion segmentation database at http://people.csail.mit.edu/celiu/motionAnnotation/ as some of your test data.

Ex 8.9: Layered motion estimation Decompose into separate layers (Section 8.5) a video sequence of a scene taken with a moving camera:

1. Find the set of dominant (affine or planar perspective) motions, either by computing them in blocks or finding a robust estimate and then iteratively re-fitting outliers.
2. Determine which pixels go with each motion.
3. Construct the layers by blending pixels from different frames.
4. (Optional) Add per-pixel residual flows or depths.
5. (Optional) Refine your estimates using an iterative global optimization technique.
6. (Optional) Write an interactive renderer to generate in-between frames or view the scene from different viewpoints (Shade, Gortler, He *et al.* 1998).
7. (Optional) Construct an *unwrap mosaic* from a more complex scene and use this to do some video editing (Rav-Acha, Kohli, Fitzgibbon *et al.* 2008).

Ex 8.10: Transparent motion and reflection estimation Take a video sequence looking through a window (or picture frame) and see if you can remove the reflection in order to better see what is inside.

The steps are described in Section 8.5.2 and by Szeliski, Avidan, and Anandan (2000). Alternative approaches can be found in work by Shizawa and Mase (1991), Bergen, Burt, Hingorani *et al.* (1992), Darrell and Simoncelli (1993), Darrell and Pentland (1995), Irani, Rousso, and Peleg (1994), Black and Anandan (1996), and Ju, Black, and Jepson (1996).

Chapter 9

Image stitching

9.1 Motion models . . . 378
9.1.1 Planar perspective motion . . . 379
9.1.2 *Application*: Whiteboard and document scanning . . . 379
9.1.3 Rotational panoramas . . . 380
9.1.4 Gap closing . . . 382
9.1.5 *Application*: Video summarization and compression . . . 383
9.1.6 Cylindrical and spherical coordinates . . . 385
9.2 Global alignment . . . 387
9.2.1 Bundle adjustment . . . 388
9.2.2 Parallax removal . . . 391
9.2.3 Recognizing panoramas . . . 392
9.2.4 Direct vs. feature-based alignment . . . 393
9.3 Compositing . . . 396
9.3.1 Choosing a compositing surface . . . 396
9.3.2 Pixel selection and weighting (de-ghosting) . . . 398
9.3.3 *Application*: Photomontage . . . 403
9.3.4 Blending . . . 403
9.4 Additional reading . . . 406
9.5 Exercises . . . 407

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_9,

© Springer-Verlag London Limited 2011



(a)



(b)



(c)



(d)



(e)

Figure 9.1 Image stitching: (a) portion of a cylindrical panorama and (b) a spherical panorama constructed from 54 photographs (Szeliski and Shum 1997) © 1997 ACM; (c) a multi-image panorama automatically assembled from an unordered photo collection; a multi-image stitch (d) without and (e) with moving object removal (Uyttendaele, Eden, and Szeliski 2001) © 2001 IEEE.

Algorithms for aligning images and stitching them into seamless photo-mosaics are among the oldest and most widely used in computer vision (Milgram 1975; Peleg 1981). image stitching algorithms create the high-resolution photo-mosaics used to produce today's digital maps and satellite photos. They also come bundled with most digital cameras and can be used to create beautiful ultra wide-angle panoramas.

image stitching originated in the photogrammetry community, where more manually intensive methods based on surveyed *ground control points* or manually registered *tie points* have long been used to register aerial photos into large-scale photo-mosaics (Slama 1980). One of the key advances in this community was the development of *bundle adjustment* algorithms (Section 7.4), which could simultaneously solve for the locations of all of the camera positions, thus yielding globally consistent solutions (Triggs, McLauchlan, Hartley *et al.* 1999). Another recurring problem in creating photo-mosaics is the elimination of visible seams, for which a variety of techniques have been developed over the years (Milgram 1975, 1977; Peleg 1981; Davis 1998; Agarwala, Dontcheva, Agrawala *et al.* 2004)

In film photography, special cameras were developed in the 1990s to take ultra-wide-angle panoramas, often by exposing the film through a vertical slit as the camera rotated on its axis (Meehan 1990). In the mid-1990s, image alignment techniques started being applied to the construction of wide-angle seamless panoramas from regular hand-held cameras (Mann and Picard 1994; Chen 1995; Szeliski 1996). More recent work in this area has addressed the need to compute globally consistent alignments (Szeliski and Shum 1997; Sawhney and Kumar 1999; Shum and Szeliski 2000), to remove "ghosts" due to parallax and object movement (Davis 1998; Shum and Szeliski 2000; Uyttendaele, Eden, and Szeliski 2001; Agarwala, Dontcheva, Agrawala *et al.* 2004), and to deal with varying exposures (Mann and Picard 1994; Uyttendaele, Eden, and Szeliski 2001; Levin, Zomet, Peleg *et al.* 2004; Agarwala, Dontcheva, Agrawala *et al.* 2004; Eden, Uyttendaele, and Szeliski 2006; Kopf, Uyttendaele, Deussen *et al.* 2007).[1] These techniques have spawned a large number of commercial stitching products (Chen 1995; Sawhney, Kumar, Gendel *et al.* 1998), of which reviews and comparisons can be found on the Web.[2]

While most of the earlier techniques worked by directly minimizing pixel-to-pixel dissimilarities, more recent algorithms usually extract a sparse set of features and match them to each other, as described in Chapter 4. Such feature-based approaches to image stitching have the advantage of being more robust against scene movement and are potentially faster, if implemented the right way. Their biggest advantage, however, is the ability to "recognize panoramas", i.e., to automatically discover the adjacency (overlap) relationships among an unordered set of images, which makes them ideally suited for fully automated stitching of panoramas taken by casual users (Brown and Lowe 2007).

What, then, are the essential problems in image stitching? As with image alignment, we must first determine the appropriate mathematical model relating pixel coordinates in one image to pixel coordinates in another; Section 9.1 reviews the basic models we have studied and presents some new motion models related specifically to panoramic image stitching. Next, we must somehow estimate the correct alignments relating various pairs (or collections) of images. Chapter 4 discussed how distinctive features can be found in each image and then

[1] A collection of some of these papers was compiled by Benosman and Kang (2001) and they are surveyed by Szeliski (2006a).

[2] The Photosynth Web site, http://photosynth.net, allows people to create and upload panoramas for free.



(a) translation [2 dof] (b) affine [6 dof] (c) perspective [8 dof] (d) 3D rotation [3+ dof]

Figure 9.2 Two-dimensional motion models and how they can be used for image stitching.

efficiently matched to rapidly establish correspondences between pairs of images. Chapter 8 discussed how direct pixel-to-pixel comparisons combined with gradient descent (and other optimization techniques) can also be used to estimate these parameters. When multiple images exist in a panorama, bundle adjustment (Section 7.4) can be used to compute a globally consistent set of alignments and to efficiently discover which images overlap one another. In Section 9.2, we look at how each of these previously developed techniques can be modified to take advantage of the imaging setups commonly used to create panoramas.

Once we have aligned the images, we must choose a final compositing surface for warping the aligned images (Section 9.3.1). We also need algorithms to seamlessly cut and blend overlapping images, even in the presence of parallax, lens distortion, scene motion, and exposure differences (Section 9.3.2–9.3.4).

9.1 Motion models

Before we can register and align images, we need to establish the mathematical relationships that map pixel coordinates from one image to another. A variety of such *parametric motion models* are possible, from simple 2D transforms, to planar perspective models, 3D camera rotations, lens distortions, and mapping to non-planar (e.g., cylindrical) surfaces.

We already covered several of these models in Sections 2.1 and 6.1. In particular, we saw in Section 2.1.5 how the parametric motion describing the deformation of a planar surfaced as viewed from different positions can be described with an eight-parameter homography (2.71) (Mann and Picard 1994; Szeliski 1996). We also saw how a camera undergoing a pure rotation induces a different kind of homography (2.72).

In this section, we review both of these models and show how they can be applied to different stitching situations. We also introduce spherical and cylindrical compositing surfaces and show how, under favorable circumstances, they can be used to perform alignment using pure translations (Section 9.1.6). Deciding which alignment model is most appropriate for a given situation or set of data is a *model selection* problem (Hastie, Tibshirani, and Friedman 2001; Torr 2002; Bishop 2006; Robert 2007), an important topic we do not cover in this book.

9.1.1 Planar perspective motion

The simplest possible motion model to use when aligning images is to simply translate and rotate them in 2D (Figure 9.2a). This is exactly the same kind of motion that you would use if you had overlapping photographic prints. It is also the kind of technique favored by David Hockney to create the collages that he calls *joiners* (Zelnik-Manor and Perona 2007; Nomura, Zhang, and Nayar 2007). Creating such collages, which show visible seams and inconsistencies that add to the artistic effect, is popular on Web sites such as Flickr, where they more commonly go under the name *panography* (Section 6.1.2). Translation and rotation are also usually adequate motion models to compensate for small camera motions in applications such as photo and video stabilization and merging (Exercise 6.1 and Section 8.2.1).

In Section 6.1.3, we saw how the mapping between two cameras viewing a common plane can be described using a 3×3 homography (2.71). Consider the matrix $\boldsymbol{M}_{10}$ that arises when mapping a pixel in one image to a 3D point and then back onto a second image,

$$\tilde{\boldsymbol{x}}_1 \sim \tilde{\boldsymbol{P}}_1 \tilde{\boldsymbol{P}}_0^{-1} \tilde{\boldsymbol{x}}_0 = \boldsymbol{M}_{10} \tilde{\boldsymbol{x}}_0. \tag{9.1}$$

When the last row of the $\boldsymbol{P}_0$ matrix is replaced with a plane equation $\hat{\boldsymbol{n}}_0 \cdot \boldsymbol{p} + c_0$ and points are assumed to lie on this plane, i.e., their disparity is $d_0 = 0$, we can ignore the last column of $\boldsymbol{M}_{10}$ and also its last row, since we do not care about the final z-buffer depth. The resulting homography matrix $\tilde{\boldsymbol{H}}_{10}$ (the upper left 3×3 sub-matrix of $\boldsymbol{M}_{10}$) describes the mapping between pixels in the two images,

$$\tilde{\boldsymbol{x}}_1 \sim \tilde{\boldsymbol{H}}_{10} \tilde{\boldsymbol{x}}_0. \tag{9.2}$$

This observation formed the basis of some of the earliest automated image stitching algorithms (Mann and Picard 1994; Szeliski 1994, 1996). Because reliable feature matching techniques had not yet been developed, these algorithms used direct pixel value matching, i.e., direct parametric motion estimation, as described in Section 8.2 and Equations (6.19–6.20).

More recent stitching algorithms first extract features and then match them up, often using robust techniques such as RANSAC (Section 6.1.4) to compute a good set of inliers. The final computation of the homography (9.2), i.e., the solution of the least squares fitting problem given pairs of corresponding features,

$$x_1 = \frac{(1+h_{00})x_0 + h_{01}y_0 + h_{02}}{h_{20}x_0 + h_{21}y_0 + 1} \quad \text{and} \quad y_1 = \frac{h_{10}x_0 + (1+h_{11})y_0 + h_{12}}{h_{20}x_0 + h_{21}y_0 + 1}, \tag{9.3}$$

uses iterative least squares, as described in Section 6.1.3 and Equations (6.21–6.23).

9.1.2 *Application*: Whiteboard and document scanning

The simplest image-stitching application is to stitch together a number of image scans taken on a flatbed scanner. Say you have a large map, or a piece of child's artwork, that is too large to fit on your scanner. Simply take multiple scans of the document, making sure to overlap the scans by a large enough amount to ensure that there are enough common features. Next, take successive pairs of images that you know overlap, extract features, match them up, and estimate the 2D rigid transform (2.16),

$$\boldsymbol{x}_{k+1} = \boldsymbol{R}_k \boldsymbol{x}_k + \boldsymbol{t}_k, \tag{9.4}$$

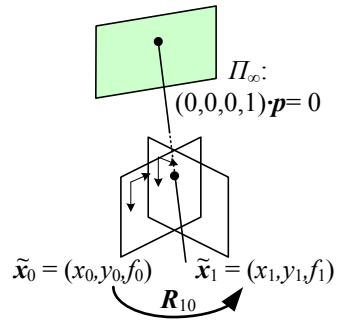


Figure 9.3 Pure 3D camera rotation. The form of the homography (mapping) is particularly simple and depends only on the 3D rotation matrix and focal lengths.

that best matches the features, using two-point RANSAC, if necessary, to find a good set of inliers. Then, on a final compositing surface (aligned with the first scan, for example), resample your images (Section 3.6.1) and average them together. Can you see any potential problems with this scheme?

One complication is that a 2D rigid transformation is non-linear in the rotation angle θ, so you will have to either use non-linear least squares or constrain $\boldsymbol{R}$ to be orthonormal, as described in Section 6.1.3.

A bigger problem lies in the pairwise alignment process. As you align more and more pairs, the solution may drift so that it is no longer globally consistent. In this case, a global optimization procedure, as described in Section 9.2, may be required. Such global optimization often requires a large system of non-linear equations to be solved, although in some cases, such as linearized homographies (Section 9.1.3) or similarity transforms (Section 6.1.2), regular least squares may be an option.

A slightly more complex scenario is when you take multiple overlapping handheld pictures of a whiteboard or other large planar object (He and Zhang 2005; Zhang and He 2007). Here, the natural motion model to use is a homography, although a more complex model that estimates the 3D rigid motion relative to the plane (plus the focal length, if unknown), could in principle be used.

9.1.3 Rotational panoramas

The most typical case for panoramic image stitching is when the camera undergoes a pure rotation. Think of standing at the rim of the Grand Canyon. Relative to the distant geometry in the scene, as you snap away, the camera is undergoing a pure rotation, which is equivalent to assuming that all points are very far from the camera, i.e., on the *plane at infinity* (Figure 9.3). Setting $\boldsymbol{t}_0 = \boldsymbol{t}_1 = 0$, we get the simplified 3×3 homography

$$\tilde{\boldsymbol{H}}_{10} = \boldsymbol{K}_1 \boldsymbol{R}_1 \boldsymbol{R}_0^{-1} \boldsymbol{K}_0^{-1} = \boldsymbol{K}_1 \boldsymbol{R}_{10} \boldsymbol{K}_0^{-1}, \tag{9.5}$$

where $\boldsymbol{K}_k = \mathrm{diag}(f_k, f_k, 1)$ is the simplified camera intrinsic matrix (2.59), assuming that $c_x = c_y = 0$, i.e., we are indexing the pixels starting from the optical center (Szeliski 1996).

This can also be re-written as

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \sim \begin{bmatrix} f_1 & & \\ & f_1 & \\ & & 1 \end{bmatrix} \boldsymbol{R}_{10} \begin{bmatrix} f_0^{-1} & & \\ & f_0^{-1} & \\ & & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{9.6}$$

or

$$\begin{bmatrix} x_1 \\ y_1 \\ f_1 \end{bmatrix} \sim \boldsymbol{R}_{10} \begin{bmatrix} x_0 \\ y_0 \\ f_0 \end{bmatrix}, \tag{9.7}$$

which reveals the simplicity of the mapping equations and makes all of the motion parameters explicit. Thus, instead of the general eight-parameter homography relating a pair of images, we get the three-, four-, or five-parameter *3D rotation* motion models corresponding to the cases where the focal length f is known, fixed, or variable (Szeliski and Shum 1997).[3] Estimating the 3D rotation matrix (and, optionally, focal length) associated with each image is intrinsically more stable than estimating a homography with a full eight degrees of freedom, which makes this the method of choice for large-scale image stitching algorithms (Szeliski and Shum 1997; Shum and Szeliski 2000; Brown and Lowe 2007).

Given this representation, how do we update the rotation matrices to best align two overlapping images? Given a current estimate for the homography $\tilde{\boldsymbol{H}}_{10}$ in (9.5), the best way to update $\boldsymbol{R}_{10}$ is to prepend an *incremental* rotation matrix $\boldsymbol{R}(\boldsymbol{\omega})$ to the current estimate $\boldsymbol{R}_{10}$ (Szeliski and Shum 1997; Shum and Szeliski 2000),

$$\tilde{\boldsymbol{H}}(\boldsymbol{\omega}) = \boldsymbol{K}_1 \boldsymbol{R}(\boldsymbol{\omega}) \boldsymbol{R}_{10} \boldsymbol{K}_0^{-1} = [\boldsymbol{K}_1 \boldsymbol{R}(\boldsymbol{\omega}) \boldsymbol{K}_1^{-1}][\boldsymbol{K}_1 \boldsymbol{R}_{10} \boldsymbol{K}_0^{-1}] = \boldsymbol{D} \tilde{\boldsymbol{H}}_{10}. \tag{9.8}$$

Note that here we have written the update rule in the *compositional* form, where the incremental update $\boldsymbol{D}$ is *prepended* to the current homography $\tilde{\boldsymbol{H}}_{10}$. Using the small-angle approximation to $\boldsymbol{R}(\boldsymbol{\omega})$ given in (2.35), we can write the incremental update matrix as

$$\boldsymbol{D} = \boldsymbol{K}_1 \boldsymbol{R}(\boldsymbol{\omega}) \boldsymbol{K}_1^{-1} \approx \boldsymbol{K}_1 (\boldsymbol{I} + [\boldsymbol{\omega}]_\times) \boldsymbol{K}_1^{-1} = \begin{bmatrix} 1 & -\omega_z & f_1 \omega_y \\ \omega_z & 1 & -f_1 \omega_x \\ -\omega_y / f_1 & \omega_x / f_1 & 1 \end{bmatrix}. \tag{9.9}$$

Notice how there is now a nice one-to-one correspondence between the entries in the $\boldsymbol{D}$ matrix and the $h_{00}, \ldots, h_{21}$ parameters used in Table 6.1 and Equation (6.19), i.e.,

$$(h_{00}, h_{01}, h_{02}, h_{00}, h_{11}, h_{12}, h_{20}, h_{21}) = (0, -\omega_z, f_1 \omega_y, \omega_z, 0, -f_1 \omega_x, -\omega_y / f_1, \omega_x / f_1). \tag{9.10}$$

We can therefore apply the chain rule to Equations (6.24 and 9.10) to obtain

$$\begin{bmatrix} \hat{x}' - x \\ \hat{y}' - y \end{bmatrix} = \begin{bmatrix} -xy/f_1 & f_1 + x^2/f_1 & -y \\ -(f_1 + y^2/f_1) & xy/f_1 & x \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \tag{9.11}$$

which give us the linearized update equations needed to estimate $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$.[4] Notice that this update rule depends on the focal length f_1 of the *target* view and is independent

[3] An initial estimate of the focal lengths can be obtained using the intrinsic calibration techniques described in Section 6.3.4 or from EXIF tags.

[4] This is the same as the rotational component of instantaneous rigid flow (Bergen, Anandan, Hanna *et al.* 1992) and the update equations given by Szeliski and Shum (1997) and Shum and Szeliski (2000).

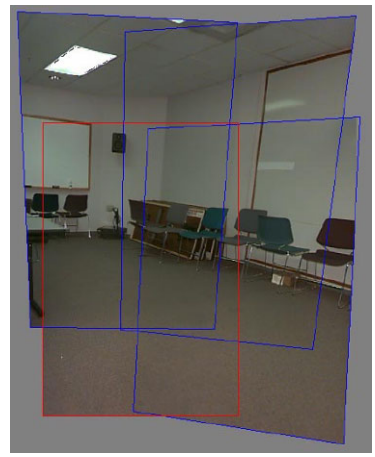


Figure 9.4 Four images taken with a hand-held camera registered using a 3D rotation motion model (Szeliski and Shum 1997) © 1997 ACM. Notice how the homographies, rather than being arbitrary, have a well-defined keystone shape whose width increases away from the origin.

of the focal length f_0 of the *template* view. This is because the compositional algorithm essentially makes small perturbations to the target. Once the incremental rotation vector $\boldsymbol{\omega}$ has been computed, the $\boldsymbol{R}_1$ rotation matrix can be updated using $\boldsymbol{R}_1 \leftarrow \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{R}_1$.

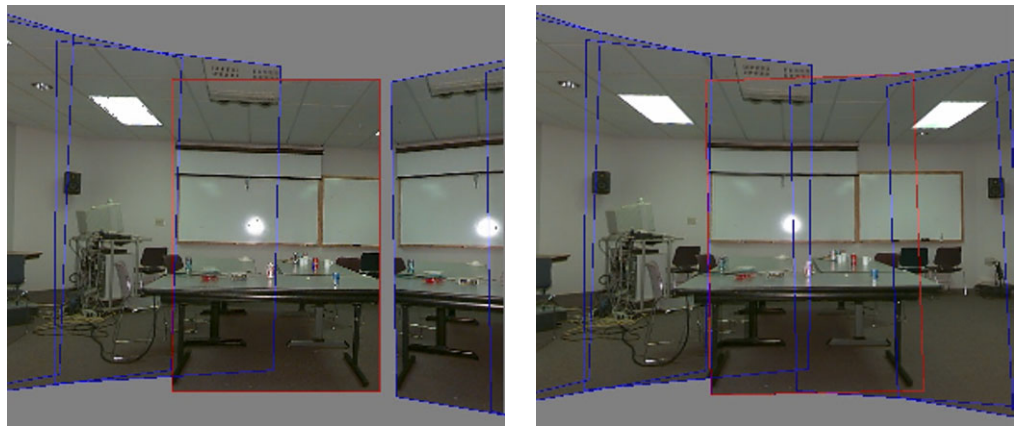
The formulas for updating the focal length estimates are a little more involved and are given in (Shum and Szeliski 2000). We will not repeat them here, since an alternative update rule, based on minimizing the difference between back-projected 3D rays, is given in Section 9.2.1. Figure 9.4 shows the alignment of four images under the 3D rotation motion model.

9.1.4 Gap closing

The techniques presented in this section can be used to estimate a series of rotation matrices and focal lengths, which can be chained together to create large panoramas. Unfortunately, because of accumulated errors, this approach will rarely produce a closed 360° panorama. Instead, there will invariably be either a gap or an overlap (Figure 9.5).

We can solve this problem by matching the first image in the sequence with the last one. The difference between the two rotation matrix estimates associated with the repeated first indicates the amount of misregistration. This error can be distributed evenly across the whole sequence by taking the quotient of the two quaternions associated with these rotations and dividing this "error quaternion" by the number of images in the sequence (assuming relatively constant inter-frame rotations). We can also update the estimated focal length based on the amount of misregistration. To do this, we first convert the error quaternion into a *gap angle*, θ_g and then update the focal length using the equation $f' = f(1 - \theta_g/360^\circ)$.

Figure 9.5a shows the end of registered image sequence and the first image. There is a big gap between the last image and the first which are in fact the same image. The gap is 32° because the wrong estimate of focal length ($f = 510$) was used. Figure 9.5b shows the registration after closing the gap with the correct focal length ($f = 468$). Notice that both



(a) (b)

Figure 9.5 Gap closing (Szeliski and Shum 1997) © 1997 ACM: (a) A gap is visible when the focal length is wrong ($f = 510$). (b) No gap is visible for the correct focal length ($f = 468$).

mosaics show very little visual misregistration (except at the gap), yet Figure 9.5a has been computed using a focal length that has 9% error. Related approaches have been developed by Hartley (1994b), McMillan and Bishop (1995), Stein (1995), and Kang and Weiss (1997) to solve the focal length estimation problem using pure panning motion and cylindrical images.

Unfortunately, this particular gap-closing heuristic only works for the kind of "one-dimensional" panorama where the camera is continuously turning in the same direction. In Section 9.2, we describe a different approach to removing gaps and overlaps that works for arbitrary camera motions.

9.1.5 *Application*: Video summarization and compression

An interesting application of image stitching is the ability to summarize and compress videos taken with a panning camera. This application was first suggested by Teodosio and Bender (1993), who called their mosaic-based summaries *salient stills*. These ideas were then extended by Irani, Hsu, and Anandan (1995), Kumar, Anandan, Irani *et al.* (1995), and Irani and Anandan (1998) to additional applications, such as video compression and video indexing. While these early approaches used affine motion models and were therefore restricted to long focal lengths, the techniques were generalized by Lee, ge Chen, lung Bruce Lin *et al.* (1997) to full eight-parameter homographies and incorporated into the MPEG-4 video compression standard, where the stitched background layers were called *video sprites* (Figure 9.6).

While video stitching is in many ways a straightforward generalization of multiple-image stitching (Steedly, Pal, and Szeliski 2005; Baudisch, Tan, Steedly *et al.* 2006), the potential presence of large amounts of independent motion, camera zoom, and the desire to visualize dynamic events impose additional challenges. For example, moving foreground objects can often be removed using *median filtering*. Alternatively, foreground objects can be extracted into a separate layer (Sawhney and Ayer 1996) and later composited back into the stitched panoramas, sometimes as multiple instances to give the impressions of a "Chronophotograph" (Massey and Bender 1996) and sometimes as video overlays (Irani and Anandan 1998).

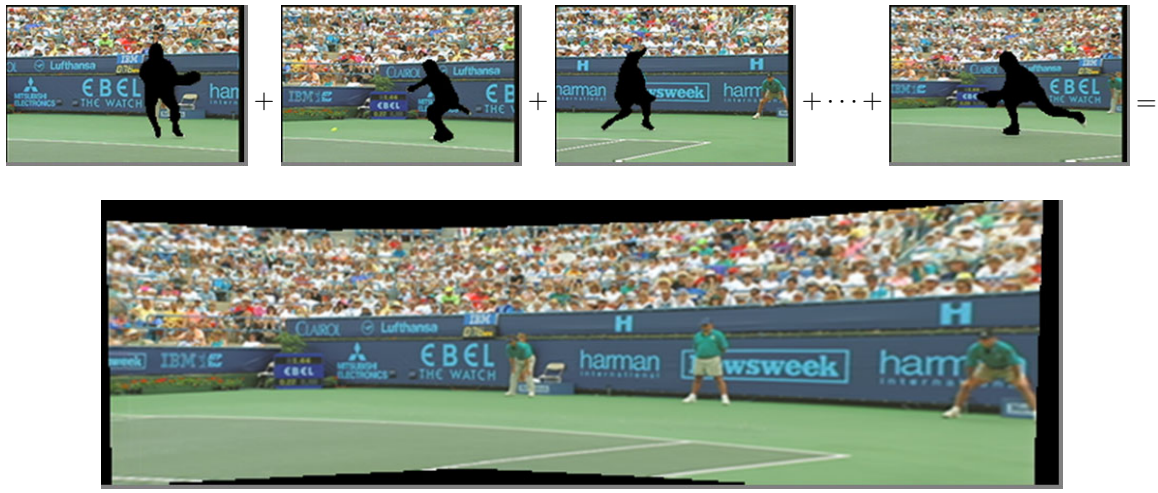


Figure 9.6 Video stitching the background scene to create a single *sprite* image that can be transmitted and used to re-create the background in each frame (Lee, ge Chen, lung Bruce Lin *et al.* 1997) © 1997 IEEE.

Videos can also be used to create animated *panoramic video textures* (Section 13.5.2), in which different portions of a panoramic scene are animated with independently moving video loops (Agarwala, Zheng, Pal *et al.* 2005; Rav-Acha, Pritch, Lischinski *et al.* 2005), or to shine "video flashlights" onto a composite mosaic of a scene (Sawhney, Arpa, Kumar *et al.* 2002).

Video can also provide an interesting source of content for creating panoramas taken from moving cameras. While this invalidates the usual assumption of a single point of view (optical center), interesting results can still be obtained. For example, the VideoBrush system of Sawhney, Kumar, Gendel *et al.* (1998) uses thin strips taken from the center of the image to create a panorama taken from a horizontally moving camera. This idea can be generalized to other camera motions and compositing surfaces using the concept of mosaics on adaptive manifold (Peleg, Rousso, Rav-Acha *et al.* 2000), and also used to generate panoramic stereograms (Peleg, Ben-Ezra, and Pritch 2001). Related ideas have been used to create panoramic matte paintings for multi-plane cel animation (Wood, Finkelstein, Hughes *et al.* 1997), for creating stitched images of scenes with parallax (Kumar, Anandan, Irani *et al.* 1995), and as 3D representations of more complex scenes using *multiple-center-of-projection images* (Rademacher and Bishop 1998) and *multi-perspective panoramas* (Román, Garg, and Levoy 2004; Román and Lensch 2006; Agarwala, Agrawala, Cohen *et al.* 2006).

Another interesting variant on video-based panoramas are *concentric mosaics* (Section 13.3.3) (Shum and He 1999). Here, rather than trying to produce a single panoramic image, the complete original video is kept and used to re-synthesize views (from different camera origins) using ray remapping (light field rendering), thus endowing the panorama with a sense of 3D depth. The same data set can also be used to explicitly reconstruct the depth using multi-baseline stereo (Peleg, Ben-Ezra, and Pritch 2001; Li, Shum, Tang *et al.* 2004; Zheng, Kang, Cohen *et al.* 2007).

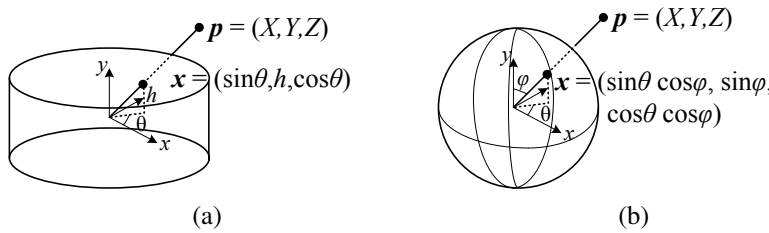


Figure 9.7 Projection from 3D to (a) cylindrical and (b) spherical coordinates.

9.1.6 Cylindrical and spherical coordinates

An alternative to using homographies or 3D motions to align images is to first warp the images into *cylindrical* coordinates and then use a pure translational model to align them (Chen 1995; Szeliski 1996). Unfortunately, this only works if the images are all taken with a level camera or with a known tilt angle.

Assume for now that the camera is in its canonical position, i.e., its rotation matrix is the identity, $\boldsymbol{R} = \boldsymbol{I}$, so that the optical axis is aligned with the z axis and the y axis is aligned vertically. The 3D ray corresponding to an (x, y) pixel is therefore (x, y, f).

We wish to project this image onto a *cylindrical surface* of unit radius (Szeliski 1996). Points on this surface are parameterized by an angle θ and a height h, with the 3D cylindrical coordinates corresponding to (θ, h) given by

$$(\sin\theta, h, \cos\theta) \propto (x, y, f), \tag{9.12}$$

as shown in Figure 9.7a. From this correspondence, we can compute the formula for the *warped* or *mapped* coordinates (Szeliski and Shum 1997),

$$x' = s\theta = s\tan^{-1}\frac{x}{f}, \tag{9.13}$$

$$y' = sh = s\frac{y}{\sqrt{x^2+f^2}}, \tag{9.14}$$

where s is an arbitrary scaling factor (sometimes called the *radius* of the cylinder) that can be set to $s = f$ to minimize the distortion (scaling) near the center of the image.[5] The inverse of this mapping equation is given by

$$x = f\tan\theta = f\tan\frac{x'}{s}, \tag{9.15}$$

$$y = h\sqrt{x^2+f^2} = \frac{y'}{s}f\sqrt{1+\tan^2 x'/s} = f\frac{y'}{s}\sec\frac{x'}{s}. \tag{9.16}$$

Images can also be projected onto a *spherical surface* (Szeliski and Shum 1997), which is useful if the final panorama includes a full sphere or hemisphere of views, instead of just a cylindrical strip. In this case, the sphere is parameterized by two angles (θ, ϕ), with 3D spherical coordinates given by

$$(\sin\theta\cos\phi, \sin\phi, \cos\theta\cos\phi) \propto (x, y, f), \tag{9.17}$$

[5] The scale can also be set to a larger or smaller value for the final compositing surface, depending on the desired output panorama resolution—see Section 9.3.



(a) (b)

Figure 9.8 A cylindrical panorama (Szeliski and Shum 1997) © 1997 ACM: (a) two cylindrically warped images related by a horizontal translation; (b) part of a cylindrical panorama composited from a sequence of images.

as shown in Figure 9.7b.[6] The correspondence between coordinates is now given by (Szeliski and Shum 1997):

$$x' = s\theta = s\tan^{-1}\frac{x}{f}, \tag{9.18}$$

$$y' = s\phi = s\tan^{-1}\frac{y}{\sqrt{x^2+f^2}}, \tag{9.19}$$

while the inverse is given by

$$x = f\tan\theta = f\tan\frac{x'}{s}, \tag{9.20}$$

$$y = \sqrt{x^2+f^2}\tan\phi = \tan\frac{y'}{s}f\sqrt{1+\tan^2 x'/s} = f\tan\frac{y'}{s}\sec\frac{x'}{s}. \tag{9.21}$$

Note that it may be simpler to generate a scaled (x, y, z) direction from Equation (9.17) followed by a perspective division by z and a scaling by f.

Cylindrical image stitching algorithms are most commonly used when the camera is known to be level and only rotating around its vertical axis (Chen 1995). Under these conditions, images at different rotations are related by a pure horizontal translation.[7] This makes it attractive as an initial class project in an introductory computer vision course, since the full complexity of the perspective alignment algorithm (Sections 6.1, 8.2, and 9.1.3) can be avoided. Figure 9.8 shows how two cylindrically warped images from a leveled rotational panorama are related by a pure translation (Szeliski and Shum 1997).

Professional panoramic photographers often use pan-tilt heads that make it easy to control the tilt and to stop at specific *detents* in the rotation angle. Motorized rotation heads are also sometimes used for the acquisition of larger panoramas (Kopf, Uyttendaele, Deussen *et al.* 2007).[8] Not only do they ensure a uniform coverage of the visual field with a desired amount of image overlap but they also make it possible to stitch the images using cylindrical or spherical coordinates and pure translations. In this case, pixel coordinates (x, y, f) must first

[6] Note that these are not the usual spherical coordinates, first presented in Equation (2.8). Here, the y axis points at the north pole instead of the z axis, since we are used to viewing images taken horizontally, i.e., with the y axis pointing in the direction of the gravity vector.

[7] Small vertical tilts can sometimes be compensated for with vertical translations.

[8] See also http://gigapan.org.



Figure 9.9 A spherical panorama constructed from 54 photographs (Szeliski and Shum 1997) © 1997 ACM.

be rotated using the known tilt and panning angles before being projected into cylindrical or spherical coordinates (Chen 1995). Having a roughly known panning angle also makes it easier to compute the alignment, since the rough relative positioning of all the input images is known ahead of time, enabling a reduced search range for alignment. Figure 9.9 shows a full 3D rotational panorama unwrapped onto the surface of a sphere (Szeliski and Shum 1997).

One final coordinate mapping worth mentioning is the *polar* mapping, where the north pole lies along the optical axis rather than the vertical axis,

$$(\cos\theta \sin\phi, \sin\theta \sin\phi, \cos\phi) = s\,(x, y, z). \tag{9.22}$$

In this case, the mapping equations become

$$x' = s\phi\cos\theta = s\frac{x}{r}\tan^{-1}\frac{r}{z}, \tag{9.23}$$

$$y' = s\phi\sin\theta = s\frac{y}{r}\tan^{-1}\frac{r}{z}, \tag{9.24}$$

where $r = \sqrt{x^2 + y^2}$ is the *radial distance* in the (x, y) plane and $s\phi$ plays a similar role in the (x', y') plane. This mapping provides an attractive visualization surface for certain kinds of wide-angle panoramas and is also a good model for the distortion induced by *fisheye lenses*, as discussed in Section 2.1.6. Note how for small values of (x, y), the mapping equations reduce to $x' \approx sx/z$, which suggests that s plays a role similar to the focal length f.

9.2 Global alignment

So far, we have discussed how to register pairs of images using a variety of motion models. In most applications, we are given more than a single pair of images to register. The goal is then to find a globally consistent set of alignment parameters that minimize the mis-registration between all pairs of images (Szeliski and Shum 1997; Shum and Szeliski 2000; Sawhney and Kumar 1999; Coorg and Teller 2000).

In this section, we extend the pairwise matching criteria (6.2, 8.1, and 8.50) to a global energy function that involves all of the per-image pose parameters (Section 9.2.1). Once we have computed the global alignment, we often need to perform *local adjustments*, such as *parallax removal*, to reduce double images and blurring due to local mis-registrations (Section 9.2.2). Finally, if we are given an unordered set of images to register, we need to discover which images go together to form one or more panoramas. This process of *panorama recognition* is described in Section 9.2.3.

9.2.1 Bundle adjustment

One way to register a large number of images is to add new images to the panorama one at a time, aligning the most recent image with the previous ones already in the collection (Szeliski and Shum 1997) and discovering, if necessary, which images it overlaps (Sawhney and Kumar 1999). In the case of 360° panoramas, accumulated error may lead to the presence of a gap (or excessive overlap) between the two ends of the panorama, which can be fixed by stretching the alignment of all the images using a process called *gap closing* (Szeliski and Shum 1997). However, a better alternative is to simultaneously align all the images using a least-squares framework to correctly distribute any mis-registration errors.

The process of simultaneously adjusting pose parameters for a large collection of overlapping images is called *bundle adjustment* in the photogrammetry community (Triggs, McLauchlan, Hartley *et al.* 1999). In computer vision, it was first applied to the general structure from motion problem (Szeliski and Kang 1994) and then later specialized for panoramic image stitching (Shum and Szeliski 2000; Sawhney and Kumar 1999; Coorg and Teller 2000).

In this section, we formulate the problem of global alignment using a feature-based approach, since this results in a simpler system. An equivalent direct approach can be obtained either by dividing images into patches and creating a virtual feature correspondence for each one (as discussed in Section 9.2.4 and by Shum and Szeliski (2000)) or by replacing the per-feature error metrics with per-pixel metrics.

Consider the feature-based alignment problem given in Equation (6.2), i.e.,

$$E_{\text{pairwise-LS}} = \sum_i \|\boldsymbol{r}_i\|^2 = \|\tilde{\boldsymbol{x}}'_i(\boldsymbol{x}_i;\boldsymbol{p}) - \hat{\boldsymbol{x}}'_i\|^2. \tag{9.25}$$

For multi-image alignment, instead of having a single collection of pairwise feature correspondences, $\{(\boldsymbol{x}_i, \hat{\boldsymbol{x}}'_i)\}$, we have a collection of n features, with the location of the ith feature point in the jth image denoted by $\boldsymbol{x}_{ij}$ and its scalar confidence (i.e., inverse variance) denoted by c_{ij}.[9] Each image also has some associated pose parameters.

In this section, we assume that this pose consists of a rotation matrix $\boldsymbol{R}_j$ and a focal length f_j, although formulations in terms of homographies are also possible (Szeliski and Shum 1997; Sawhney and Kumar 1999). The equation mapping a 3D point $\boldsymbol{x}_i$ into a point $\boldsymbol{x}_{ij}$ in frame j can be re-written from Equations (2.68) and (9.5) as

$$\tilde{\boldsymbol{x}}_{ij} \sim \boldsymbol{K}_j \boldsymbol{R}_j \boldsymbol{x}_i \;\text{ and }\; \boldsymbol{x}_i \sim \boldsymbol{R}_j^{-1} \boldsymbol{K}_j^{-1} \tilde{\boldsymbol{x}}_{ij}, \tag{9.26}$$

[9] Features that are not seen in image j have $c_{ij} = 0$. We can also use 2×2 inverse covariance matrices Σ_{ij}^{-1} in place of c_{ij}, as shown in Equation (6.11).

where $\boldsymbol{K}_j = \mathrm{diag}(f_j, f_j, 1)$ is the simplified form of the calibration matrix. The motion mapping a point $\boldsymbol{x}_{ij}$ from frame j into a point $\boldsymbol{x}_{ik}$ in frame k is similarly given by

$$\tilde{\boldsymbol{x}}_{ik} \sim \tilde{\boldsymbol{H}}_{kj}\tilde{\boldsymbol{x}}_{ij} = \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{R}_j^{-1}\boldsymbol{K}_j^{-1}\tilde{\boldsymbol{x}}_{ij}. \tag{9.27}$$

Given an initial set of $\{(\boldsymbol{R}_j, f_j)\}$ estimates obtained from chaining pairwise alignments, how do we refine these estimates?

One approach is to directly extend the pairwise energy $E_{\mathrm{pairwise-LS}}$ (9.25) to a multiview formulation,

$$E_{\mathrm{all-pairs-2D}} = \sum_i \sum_{jk} c_{ij}c_{ik}\|\tilde{\boldsymbol{x}}_{ik}(\hat{\boldsymbol{x}}_{ij}; \boldsymbol{R}_j, f_j, \boldsymbol{R}_k, f_k) - \hat{\boldsymbol{x}}_{ik}\|^2, \tag{9.28}$$

where the $\tilde{\boldsymbol{x}}_{ik}$ function is the *predicted* location of feature i in frame k given by (9.27), $\hat{\boldsymbol{x}}_{ij}$ is the *observed* location, and the "2D" in the subscript indicates that an image-plane error is being minimized (Shum and Szeliski 2000). Note that since $\tilde{\boldsymbol{x}}_{ik}$ depends on the $\hat{\boldsymbol{x}}_{ij}$ observed value, we actually have an *errors-in-variable* problem, which in principle requires more sophisticated techniques than least squares to solve (Van Huffel and Lemmerling 2002; Matei and Meer 2006). However, in practice, if we have enough features, we can directly minimize the above quantity using regular non-linear least squares and obtain an accurate multi-frame alignment.

While this approach works well in practice, it suffers from two potential disadvantages. First, since a summation is taken over all pairs with corresponding features, features that are observed many times are overweighted in the final solution. (In effect, a feature observed m times gets counted $\binom{m}{2}$ times instead of m times.) Second, the derivatives of $\tilde{\boldsymbol{x}}_{ik}$ with respect to the $\{(\boldsymbol{R}_j, f_j)\}$ are a little cumbersome, although using the incremental correction to $\boldsymbol{R}_j$ introduced in Section 9.1.3 makes this more tractable.

An alternative way to formulate the optimization is to use true bundle adjustment, i.e., to solve not only for the pose parameters $\{(\boldsymbol{R}_j, f_j)\}$ but also for the 3D point positions $\{\boldsymbol{x}_i\}$,

$$E_{\mathrm{BA-2D}} = \sum_i \sum_j c_{ij}\|\tilde{\boldsymbol{x}}_{ij}(\boldsymbol{x}_i; \boldsymbol{R}_j, f_j) - \hat{\boldsymbol{x}}_{ij}\|^2, \tag{9.29}$$

where $\tilde{\boldsymbol{x}}_{ij}(\boldsymbol{x}_i; \boldsymbol{R}_j, f_j)$ is given by (9.26). The disadvantage of full bundle adjustment is that there are more variables to solve for, so each iteration and also the overall convergence may be slower. (Imagine how the 3D points need to "shift" each time some rotation matrices are updated.) However, the computational complexity of each linearized Gauss–Newton step can be reduced using sparse matrix techniques (Section 7.4.1) (Szeliski and Kang 1994; Triggs, McLauchlan, Hartley *et al.* 1999; Hartley and Zisserman 2004).

An alternative formulation is to minimize the error in 3D projected ray directions (Shum and Szeliski 2000), i.e.,

$$E_{\mathrm{BA-3D}} = \sum_i \sum_j c_{ij}\|\tilde{\boldsymbol{x}}_i(\hat{\boldsymbol{x}}_{ij}; \boldsymbol{R}_j, f_j) - \boldsymbol{x}_i\|^2, \tag{9.30}$$

where $\tilde{\boldsymbol{x}}_i(\boldsymbol{x}_{ij}; \boldsymbol{R}_j, f_j)$ is given by the second half of (9.26). This has no particular advantage over (9.29). In fact, since errors are being minimized in 3D ray space, there is a bias towards estimating longer focal lengths, since the angles between rays become smaller as f increases.

However, if we eliminate the 3D rays $\boldsymbol{x}_i$, we can derive a pairwise energy formulated in 3D ray space (Shum and Szeliski 2000),

$$E_{\mathrm{all-pairs-3D}} = \sum_i \sum_{jk} c_{ij} c_{ik} \|\tilde{\boldsymbol{x}}_i(\hat{\boldsymbol{x}}_{ij}; \boldsymbol{R}_j, f_j) - \tilde{\boldsymbol{x}}_i(\hat{\boldsymbol{x}}_{ik}; \boldsymbol{R}_k, f_k)\|^2. \tag{9.31}$$

This results in the simplest set of update equations (Shum and Szeliski 2000), since the f_k can be folded into the creation of the homogeneous coordinate vector as in Equation (9.7). Thus, even though this formula over-weights features that occur more frequently, it is the method used by Shum and Szeliski (2000) and Brown, Szeliski, and Winder (2005). In order to reduce the bias towards longer focal lengths, we multiply each residual (3D error) by $\sqrt{f_j f_k}$, which is similar to projecting the 3D rays into a "virtual camera" of intermediate focal length.

Up vector selection. As mentioned above, there exists a global ambiguity in the pose of the 3D cameras computed by the above methods. While this may not appear to matter, people prefer that the final stitched image is "upright" rather than twisted or tilted. More concretely, people are used to seeing photographs displayed so that the vertical (gravity) axis points straight up in the image. Consider how you usually shoot photographs: while you may pan and tilt the camera any which way, you usually keep the horizontal edge of your camera (its x-axis) parallel to the ground plane (perpendicular to the world gravity direction).

Mathematically, this constraint on the rotation matrices can be expressed as follows. Recall from Equation (9.26) that the 3D to 2D projection is given by

$$\tilde{\boldsymbol{x}}_{ik} \sim \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{x}_i. \tag{9.32}$$

We wish to post-multiply each rotation matrix $\boldsymbol{R}_k$ by a global rotation $\boldsymbol{R}_{\mathrm{g}}$ such that the projection of the global y-axis, $\hat{\boldsymbol{\jmath}} = (0, 1, 0)$ is perpendicular to the image x-axis, $\hat{\boldsymbol{\imath}} = (1, 0, 0)$.[10]

This constraint can be written as

$$\hat{\boldsymbol{\imath}}^T \boldsymbol{R}_k \boldsymbol{R}_{\mathrm{g}} \hat{\boldsymbol{\jmath}} = 0 \tag{9.33}$$

(note that the scaling by the calibration matrix is irrelevant here). This is equivalent to requiring that the first row of $\boldsymbol{R}_k$, $\boldsymbol{r}_{k0} = \hat{\boldsymbol{\imath}}^T \boldsymbol{R}_k$ be perpendicular to the second column of $\boldsymbol{R}_{\mathrm{g}}$, $\boldsymbol{r}_{\mathrm{g}1} = \boldsymbol{R}_{\mathrm{g}} \hat{\boldsymbol{\jmath}}$. This set of constraints (one per input image) can be written as a least squares problem,

$$\boldsymbol{r}_{\mathrm{g}1} = \arg\min_{\boldsymbol{r}} \sum_k (\boldsymbol{r}^T \boldsymbol{r}_{k0})^2 = \arg\min_{\boldsymbol{r}} \boldsymbol{r}^T \left[\sum_k \boldsymbol{r}_{k0} \boldsymbol{r}_{k0}^T \right] \boldsymbol{r}. \tag{9.34}$$

Thus, $\boldsymbol{r}_{\mathrm{g}1}$ is the smallest eigenvector of the *scatter* or *moment* matrix spanned by the individual camera rotation x-vectors, which should generally be of the form $(c, 0, s)$ when the cameras are upright.

To fully specify the $\boldsymbol{R}_{\mathrm{g}}$ global rotation, we need to specify one additional constraint. This is related to the *view selection* problem discussed in Section 9.3.1. One simple heuristic is to prefer the average z-axis of the individual rotation matrices, $\overline{\boldsymbol{k}} = \sum_k \hat{\boldsymbol{k}}^T \boldsymbol{R}_k$ to be close to the world z-axis, $\boldsymbol{r}_{\mathrm{g}2} = \boldsymbol{R}_{\mathrm{g}} \hat{\boldsymbol{k}}$. We can therefore compute the full rotation matrix $\boldsymbol{R}_{\mathrm{g}}$ in three steps:

[10] Note that here we use the convention common in computer graphics that the vertical world axis corresponds to y. This is a natural choice if we wish the rotation matrix associated with a "regular" image taken horizontally to be the identity, rather than a 90° rotation around the x-axis.

1. $\boldsymbol{r}_{\mathrm{g}1} = \min$ eigenvector $(\sum_k \boldsymbol{r}_{k0}\boldsymbol{r}_{k0}^T)$;
2. $\boldsymbol{r}_{\mathrm{g}0} = \mathcal{N}((\sum_k \boldsymbol{r}_{k2}) \times \boldsymbol{r}_{\mathrm{g}1})$;
3. $\boldsymbol{r}_{\mathrm{g}2} = \boldsymbol{r}_{\mathrm{g}0} \times \boldsymbol{r}_{\mathrm{g}1}$,

where $\mathcal{N}(\boldsymbol{v}) = \boldsymbol{v}/\|\boldsymbol{v}\|$ normalizes a vector $\boldsymbol{v}$.

9.2.2 Parallax removal

Once we have optimized the global orientations and focal lengths of our cameras, we may find that the images are still not perfectly aligned, i.e., the resulting stitched image looks blurry or ghosted in some places. This can be caused by a variety of factors, including unmodeled radial distortion, 3D parallax (failure to rotate the camera around its optical center), small scene motions such as waving tree branches, and large-scale scene motions such as people moving in and out of pictures.

Each of these problems can be treated with a different approach. Radial distortion can be estimated (potentially ahead of time) using one of the techniques discussed in Section 2.1.6. For example, the *plumb-line method* (Brown 1971; Kang 2001; El-Melegy and Farag 2003) adjusts radial distortion parameters until slightly curved lines become straight, while mosaic-based approaches adjust them until mis-registration is reduced in image overlap areas (Stein 1997; Sawhney and Kumar 1999).

3D parallax can be handled by doing a full 3D bundle adjustment, i.e., by replacing the projection equation (9.26) used in Equation (9.29) with Equation (2.68), which models camera translations. The 3D positions of the matched feature points and cameras can then be simultaneously recovered, although this can be significantly more expensive than parallax-free image registration. Once the 3D structure has been recovered, the scene could (in theory) be projected to a single (central) viewpoint that contains no parallax. However, in order to do this, dense *stereo* correspondence needs to be performed (Section 11.3) (Li, Shum, Tang *et al.* 2004; Zheng, Kang, Cohen *et al.* 2007), which may not be possible if the images contain only partial overlap. In that case, it may be necessary to correct for parallax only in the overlap areas, which can be accomplished using a *multi-perspective plane sweep* (MPPS) algorithm (Kang, Szeliski, and Uyttendaele 2004; Uyttendaele, Criminisi, Kang *et al.* 2004).

When the motion in the scene is very large, i.e., when objects appear and disappear completely, a sensible solution is to simply *select* pixels from only one image at a time as the source for the final composite (Milgram 1977; Davis 1998; Agarwala, Dontcheva, Agrawala *et al.* 2004), as discussed in Section 9.3.2. However, when the motion is reasonably small (on the order of a few pixels), general 2D motion estimation (optical flow) can be used to perform an appropriate correction before blending using a process called *local alignment* (Shum and Szeliski 2000; Kang, Uyttendaele, Winder *et al.* 2003). This same process can also be used to compensate for radial distortion and 3D parallax, although it uses a weaker motion model than explicitly modeling the source of error and may, therefore, fail more often or introduce unwanted distortions.

The local alignment technique introduced by Shum and Szeliski (2000) starts with the global bundle adjustment (9.31) used to optimize the camera poses. Once these have been estimated, the *desired* location of a 3D point $\boldsymbol{x}_i$ can be estimated as the *average* of the back-



(a) (b) (c)

Figure 9.10 Deghosting a mosaic with motion parallax (Shum and Szeliski 2000) © 2000 IEEE: (a) composite with parallax; (b) after a single deghosting step (patch size 32); (c) after multiple steps (sizes 32, 16 and 8).

projected 3D locations,

$$\bar{\boldsymbol{x}}_i \sim \sum_j c_{ij} \tilde{\boldsymbol{x}}_i(\hat{\boldsymbol{x}}_{ij}; \boldsymbol{R}_j, f_j) \Big/ \sum_j c_{ij} \ , \tag{9.35}$$

which can be projected into each image j to obtain a *target location* $\bar{\boldsymbol{x}}_{ij}$. The difference between the target locations $\bar{\boldsymbol{x}}_{ij}$ and the original features $\boldsymbol{x}_{ij}$ provide a set of local motion estimates

$$\boldsymbol{u}_{ij} = \bar{\boldsymbol{x}}_{ij} - \boldsymbol{x}_{ij}, \tag{9.36}$$

which can be interpolated to form a dense correction field $\boldsymbol{u}_j(\boldsymbol{x}_j)$. In their system, Shum and Szeliski (2000) use an *inverse warping* algorithm where the sparse $-\boldsymbol{u}_{ij}$ values are placed at the new target locations $\bar{\boldsymbol{x}}_{ij}$, interpolated using bilinear kernel functions (Nielson 1993) and then added to the original pixel coordinates when computing the warped (corrected) image. In order to get a reasonably dense set of features to interpolate, Shum and Szeliski (2000) place a feature point at the center of each patch (the patch size controls the smoothness in the local alignment stage), rather than relying of features extracted using an interest operator (Figure 9.10).

An alternative approach to motion-based de-ghosting was proposed by Kang, Uyttendaele, Winder *et al.* (2003), who estimate dense optical flow between each input image and a central *reference* image. The accuracy of the flow vector is checked using a photo-consistency measure before a given warped pixel is considered valid and is used to compute a high dynamic range radiance estimate, which is the goal of their overall algorithm. The requirement for a reference image makes their approach less applicable to general image mosaicing, although an extension to this case could certainly be envisaged.

9.2.3 Recognizing panoramas

The final piece needed to perform fully automated image stitching is a technique to recognize which images actually go together, which Brown and Lowe (2007) call *recognizing panoramas*. If the user takes images in sequence so that each image overlaps its predecessor and also specifies the first and last images to be stitched, bundle adjustment combined with the process of *topology inference* can be used to automatically assemble a panorama (Sawhney and Kumar 1999). However, users often jump around when taking panoramas, e.g., they may start a new row on top of a previous one, jump back to take a repeat shot, or create

360° panoramas where end-to-end overlaps need to be discovered. Furthermore, the ability to discover multiple panoramas taken by a user over an extended period of time can be a big convenience.

To recognize panoramas, Brown and Lowe (2007) first find all pairwise image overlaps using a feature-based method and then find connected components in the overlap graph to "recognize" individual panoramas (Figure 9.11). The feature-based matching stage first extracts scale invariant feature transform (SIFT) feature locations and feature descriptors (Lowe 2004) from all the input images and places them in an indexing structure, as described in Section 4.1.3. For each image pair under consideration, the nearest matching neighbor is found for each feature in the first image, using the indexing structure to rapidly find candidates and then comparing feature descriptors to find the best match. RANSAC is used to find a set of *inlier* matches; pairs of matches are used to hypothesize similarity motion models that are then used to count the number of inliers. (A more recent RANSAC algorithm tailored specifically for rotational panoramas is described by Brown, Hartley, and Nistér (2007).)

In practice, the most difficult part of getting a fully automated stitching algorithm to work is deciding which pairs of images actually correspond to the same parts of the scene. Repeated structures such as windows (Figure 9.12) can lead to false matches when using a feature-based approach. One way to mitigate this problem is to perform a direct pixel-based comparison between the registered images to determine if they actually are different views of the same scene. Unfortunately, this heuristic may fail if there are moving objects in the scene (Figure 9.13). While there is no magic bullet for this problem, short of full scene understanding, further improvements can likely be made by applying domain-specific heuristics, such as priors on typical camera motions as well as machine learning techniques applied to the problem of match validation.

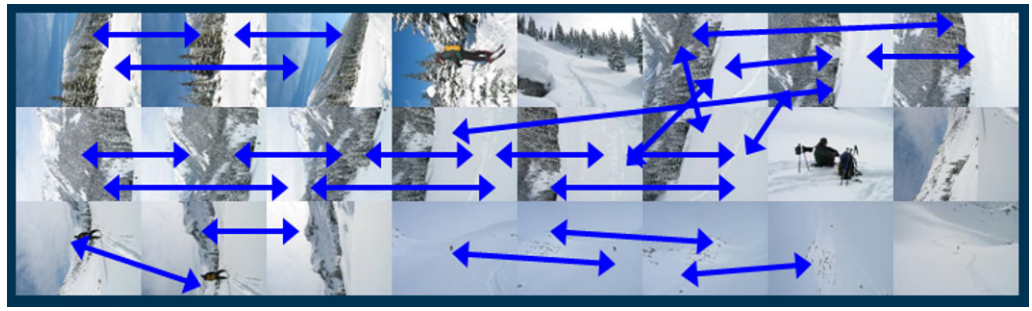
9.2.4 Direct vs. feature-based alignment

Given that there exist these two approaches to aligning images, which is preferable?

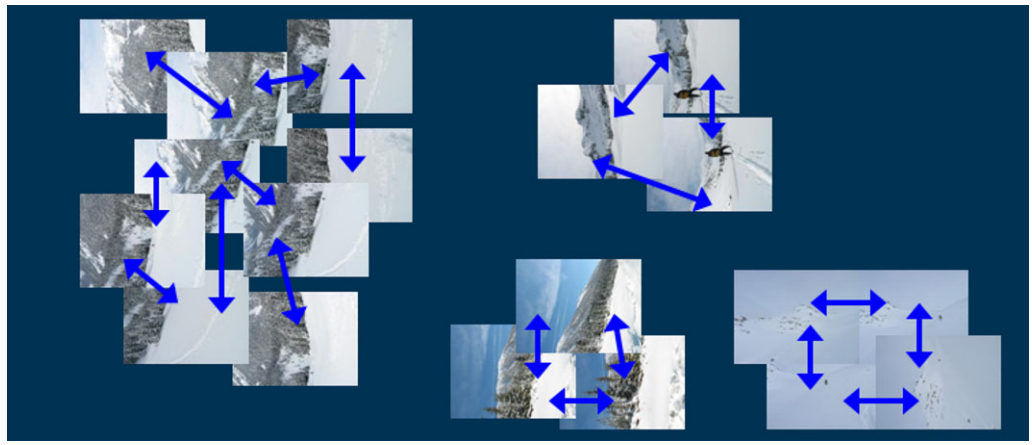
Early feature-based methods would get confused in regions that were either too textured or not textured enough. The features would often be distributed unevenly over the images, thereby failing to match image pairs that should have been aligned. Furthermore, establishing correspondences relied on simple cross-correlation between patches surrounding the feature points, which did not work well when the images were rotated or had foreshortening due to homographies.

Today, feature detection and matching schemes are remarkably robust and can even be used for known object recognition from widely separated views (Lowe 2004). Features not only respond to regions of high "cornerness" (Förstner 1986; Harris and Stephens 1988) but also to "blob-like" regions (Lowe 2004), and uniform areas (Matas, Chum, Urban *et al.* 2004; Tuytelaars and Van Gool 2004). Furthermore, because they operate in scale-space and use a dominant orientation (or orientation invariant descriptors), they can match images that differ in scale, orientation, and even foreshortening. Our own experience in working with feature-based approaches is that if the features are well distributed over the image and the descriptors reasonably designed for repeatability, enough correspondences to permit image stitching can usually be found (Brown, Szeliski, and Winder 2005).

The biggest disadvantage of direct pixel-based alignment techniques is that they have a limited range of convergence. Even though they can be used in a hierarchical (coarse-to-



(a)



(b)



(c)

Figure 9.11 Recognizing panoramas (Brown, Szeliski, and Winder 2005), figures courtesy of Matthew Brown: (a) input images with pairwise matches; (b) images grouped into connected components (panoramas); (c) individual panoramas registered and blended into stitched composites.



Figure 9.12 Matching errors (Brown, Szeliski, and Winder 2004): accidental matching of several features can lead to matches between pairs of images that do not actually overlap.



Figure 9.13 Validation of image matches by direct pixel error comparison can fail when the scene contains moving objects (Uyttendaele, Eden, and Szeliski 2001) © 2001 IEEE.

fine) estimation framework, in practice it is hard to use more than two or three levels of a pyramid before important details start to be blurred away.[11] For matching sequential frames in a video, direct approaches can usually be made to work. However, for matching partially overlapping images in photo-based panoramas or for image collections where the contrast or content varies too much, they fail too often to be useful and feature-based approaches are therefore preferred.

9.3 Compositing

Once we have registered all of the input images with respect to each other, we need to decide how to produce the final stitched mosaic image. This involves selecting a final compositing surface (flat, cylindrical, spherical, etc.) and view (reference image). It also involves selecting which pixels contribute to the final composite and how to optimally blend these pixels to minimize visible seams, blur, and ghosting.

In this section, we review techniques that address these problems, namely compositing surface parameterization, pixel and seam selection, blending, and exposure compensation. My emphasis is on fully automated approaches to the problem. Since the creation of high-quality panoramas and composites is as much an artistic endeavor as a computational one, various interactive tools have been developed to assist this process (Agarwala, Dontcheva, Agrawala *et al.* 2004; Li, Sun, Tang *et al.* 2004; Rother, Kolmogorov, and Blake 2004). Some of these are covered in more detail in Section 10.4.

9.3.1 Choosing a compositing surface

The first choice to be made is how to represent the final image. If only a few images are stitched together, a natural approach is to select one of the images as the *reference* and to then warp all of the other images into its reference coordinate system. The resulting composite is sometimes called a *flat* panorama, since the projection onto the final surface is still a perspective projection, and hence straight lines remain straight (which is often a desirable attribute).[12]

For larger fields of view, however, we cannot maintain a flat representation without excessively stretching pixels near the border of the image. (In practice, flat panoramas start to look severely distorted once the field of view exceeds 90° or so.) The usual choice for compositing larger panoramas is to use a cylindrical (Chen 1995; Szeliski 1996) or spherical (Szeliski and Shum 1997) projection, as described in Section 9.1.6. In fact, any surface used for *environment mapping* in computer graphics can be used, including a *cube map*, which represents the full viewing sphere with the six square faces of a cube (Greene 1986; Szeliski and Shum 1997). Cartographers have also developed a number of alternative methods for representing the globe (Bugayevskiy and Snyder 1995).

The choice of parameterization is somewhat application dependent and involves a trade-off between keeping the local appearance undistorted (e.g., keeping straight lines straight)

[11] Fourier-based correlation (Szeliski 1996; Szeliski and Shum 1997) can extend this range but requires cylindrical images or motion prediction to be useful.

[12] Recently, some techniques have been developed to straighten curved lines in cylindrical and spherical panoramas (Carroll, Agrawala, and Agarwala 2009; Kopf, Lischinski, Deussen *et al.* 2009).

and providing a reasonably uniform sampling of the environment. Automatically making this selection and smoothly transitioning between representations based on the extent of the panorama is an active area of current research (Kopf, Uyttendaele, Deussen *et al.* 2007).

An interesting recent development in panoramic photography has been the use of stereographic projections looking down at the ground (in an outdoor scene) to create "little planet" renderings.[13]

View selection. Once we have chosen the output parameterization, we still need to determine which part of the scene will be *centered* in the final view. As mentioned above, for a flat composite, we can choose one of the images as a reference. Often, a reasonable choice is the one that is geometrically most central. For example, for rotational panoramas represented as a collection of 3D rotation matrices, we can choose the image whose z-axis is closest to the average z-axis (assuming a reasonable field of view). Alternatively, we can use the average z-axis (or quaternion, but this is trickier) to define the reference rotation matrix.

For larger, e.g., cylindrical or spherical, panoramas, we can use the same heuristic if a subset of the viewing sphere has been imaged. In the case of full 360° panoramas, a better choice might be to choose the middle image from the sequence of inputs, or sometimes the first image, assuming this contains the object of greatest interest. In all of these cases, having the user control the final view is often highly desirable. If the "up vector" computation described in Section 9.2.1 is working correctly, this can be as simple as panning over the image or setting a vertical "center line" for the final panorama.

Coordinate transformations. After selecting the parameterization and reference view, we still need to compute the mappings between the input and output pixels coordinates.

If the final compositing surface is flat (e.g., a single plane or the face of a cube map) and the input images have no radial distortion, the coordinate transformation is the simple homography described by (9.5). This kind of warping can be performed in graphics hardware by appropriately setting texture mapping coordinates and rendering a single quadrilateral.

If the final composite surface has some other analytic form (e.g., cylindrical or spherical), we need to convert every pixel in the final panorama into a viewing ray (3D point) and then map it back into each image according to the projection (and optionally radial distortion) equations. This process can be made more efficient by precomputing some lookup tables, e.g., the partial trigonometric functions needed to map cylindrical or spherical coordinates to 3D coordinates or the radial distortion field at each pixel. It is also possible to accelerate this process by computing exact pixel mappings on a coarser grid and then interpolating these values.

When the final compositing surface is a texture-mapped polyhedron, a slightly more sophisticated algorithm must be used. Not only do the 3D and texture map coordinates have to be properly handled, but a small amount of *overdraw* outside the triangle footprints in the texture map is necessary, to ensure that the texture pixels being interpolated during 3D rendering have valid values (Szeliski and Shum 1997).

[13] These are inspired by *The Little Prince* by Antoine De Saint-Exupery. Go to http://www.flickr.com and search for "little planet projection".

Sampling issues. While the above computations can yield the correct (fractional) pixel addresses in each input image, we still need to pay attention to sampling issues. For example, if the final panorama has a lower resolution than the input images, pre-filtering the input images is necessary to avoid aliasing. These issues have been extensively studied in both the image processing and computer graphics communities. The basic problem is to compute the appropriate pre-filter, which depends on the distance (and arrangement) between neighboring samples in a source image. As discussed in Sections 3.5.2 and 3.6.1, various approximate solutions, such as MIP mapping (Williams 1983) or elliptically weighted Gaussian averaging (Greene and Heckbert 1986) have been developed in the graphics community. For highest visual quality, a higher order (e.g., cubic) interpolator combined with a spatially adaptive pre-filter may be necessary (Wang, Kang, Szeliski *et al.* 2001). Under certain conditions, it may also be possible to produce images with a higher resolution than the input images using the process of *super-resolution* (Section 10.3).

9.3.2 Pixel selection and weighting (de-ghosting)

Once the source pixels have been mapped onto the final composite surface, we must still decide how to blend them in order to create an attractive-looking panorama. If all of the images are in perfect registration and identically exposed, this is an easy problem, i.e., any pixel or combination will do. However, for real images, visible seams (due to exposure differences), blurring (due to mis-registration), or ghosting (due to moving objects) can occur.

Creating clean, pleasing-looking panoramas involves both deciding which pixels to use and how to weight or blend them. The distinction between these two stages is a little fluid, since per-pixel weighting can be thought of as a combination of selection and blending. In this section, we discuss spatially varying weighting, pixel selection (seam placement), and then more sophisticated blending.

Feathering and center-weighting. The simplest way to create a final composite is to simply take an *average* value at each pixel,

$$C(\boldsymbol{x}) = \sum_k w_k(\boldsymbol{x}) \tilde{I}_k(\boldsymbol{x}) \Big/ \sum_k w_k(\boldsymbol{x}) \ , \tag{9.37}$$

where $\tilde{I}_k(\boldsymbol{x})$ are the *warped* (re-sampled) images and $w_k(\boldsymbol{x})$ is 1 at valid pixels and 0 elsewhere. On computer graphics hardware, this kind of summation can be performed in an *accumulation buffer* (using the A channel as the weight).

Simple averaging usually does not work very well, since exposure differences, mis-registrations, and scene movement are all very visible (Figure 9.14a). If rapidly moving objects are the only problem, taking a *median* filter (which is a kind of pixel selection operator) can often be used to remove them (Figure 9.14b) (Irani and Anandan 1998). Conversely, center-weighting (discussed below) and *minimum likelihood* selection (Agarwala, Dontcheva, Agrawala *et al.* 2004) can sometimes be used to retain multiple copies of a moving object (Figure 9.17).

A better approach to averaging is to weight pixels near the center of the image more heavily and to down-weight pixels near the edges. When an image has some cutout regions,

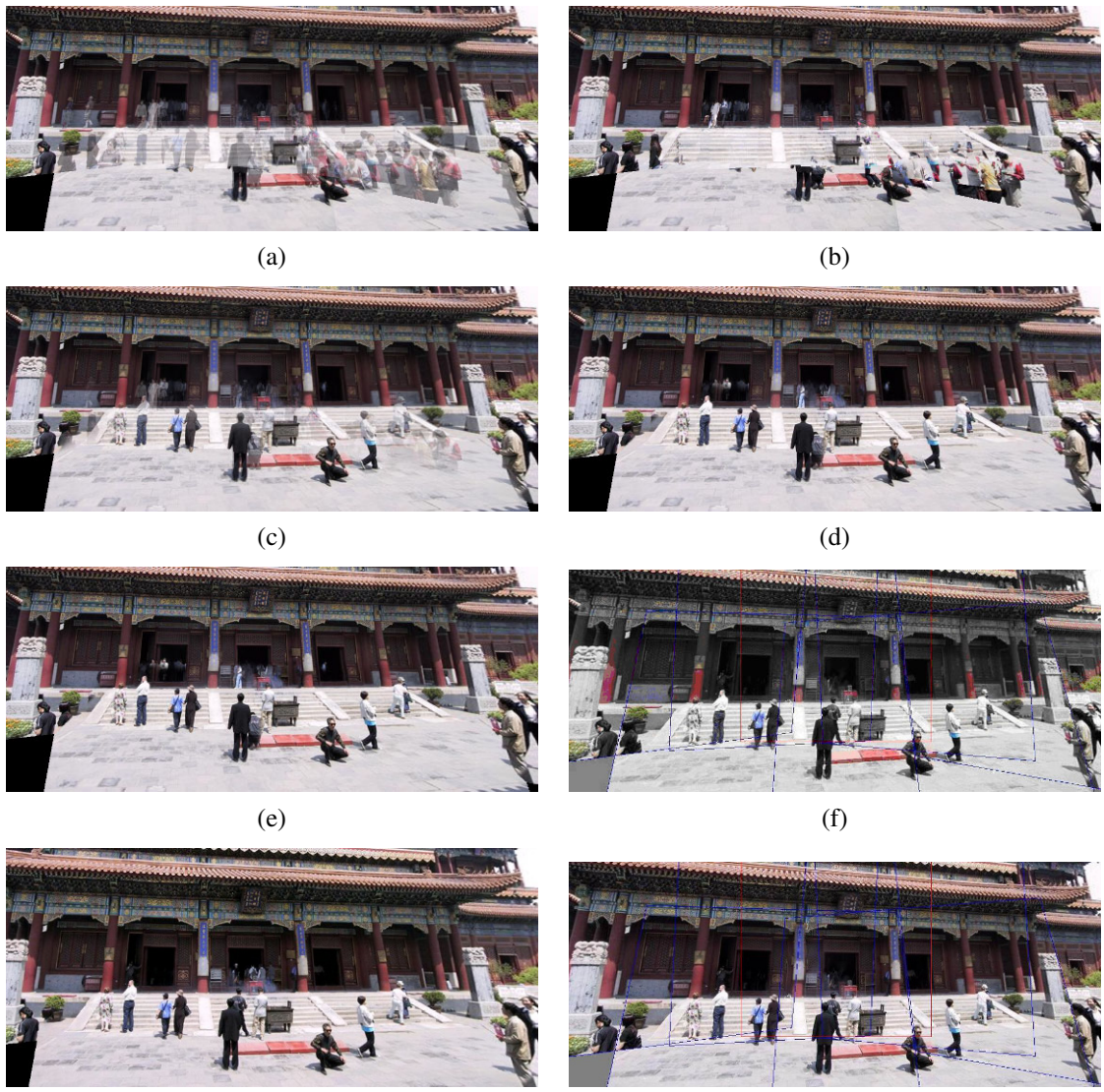


Figure 9.14 Final composites computed by a variety of algorithms (Szeliski 2006a): (a) average, (b) median, (c) feathered average, (d) *p-norm* $p = 10$, (e) Voronoi, (f) weighted ROD vertex cover with feathering, (g) graph cut seams with Poisson blending and (h) with pyramid blending.

down-weighting pixels near the edges of both cutouts and the image is preferable. This can be done by computing a *distance map* or *grassfire transform*,

$$w_k(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} \{\|\boldsymbol{y}\| \mid \tilde{I}_k(\boldsymbol{x}+\boldsymbol{y}) \text{ is invalid } \}, \tag{9.38}$$

where each valid pixel is tagged with its Euclidean distance to the nearest invalid pixel (Section 3.3.3). The Euclidean distance map can be efficiently computed using a two-pass raster algorithm (Danielsson 1980; Borgefors 1986).

Weighted averaging with a distance map is often called *feathering* (Szeliski and Shum 1997; Chen and Klette 1999; Uyttendaele, Eden, and Szeliski 2001) and does a reasonable job of blending over exposure differences. However, blurring and ghosting can still be problems (Figure 9.14c). Note that weighted averaging is *not* the same as compositing the individual images with the classic *over* operation (Porter and Duff 1984; Blinn 1994a), even when using the weight values (normalized to sum up to one) as *alpha* (translucency) channels. This is because the over operation attenuates the values from more distant surfaces and, hence, is not equivalent to a direct sum.

One way to improve feathering is to raise the distance map values to some large power, i.e., to use $w_k^p(\boldsymbol{x})$ in Equation (9.37). The weighted averages then become dominated by the larger values, i.e., they act somewhat like a *p-norm*. The resulting composite can often provide a reasonable tradeoff between visible exposure differences and blur (Figure 9.14d).

In the limit as $p \to \infty$, only the pixel with the maximum weight is selected,

$$C(\boldsymbol{x}) = \tilde{I}_{l(\boldsymbol{x})}(\boldsymbol{x}), \tag{9.39}$$

where

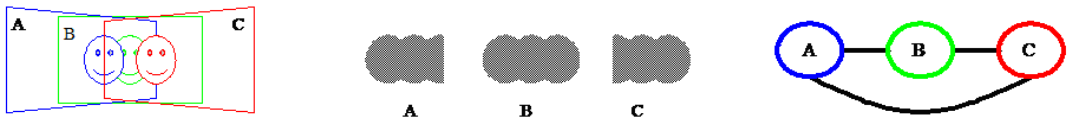
$$l = \arg\max_k w_k(\boldsymbol{x}) \tag{9.40}$$

is the *label assignment* or *pixel selection* function that selects which image to use at each pixel. This hard pixel selection process produces a visibility mask-sensitive variant of the familiar *Voronoi diagram*, which assigns each pixel to the nearest image center in the set (Wood, Finkelstein, Hughes *et al.* 1997; Peleg, Rousso, Rav-Acha *et al.* 2000). The resulting composite, while useful for artistic guidance and in high-overlap panoramas (*manifold mosaics*) tends to have very hard edges with noticeable seams when the exposures vary (Figure 9.14e).

Xiong and Turkowski (1998) use this Voronoi idea (local maximum of the grassfire transform) to select seams for Laplacian pyramid blending (which is discussed below). However, since the seam selection is performed sequentially as new images are added in, some artifacts can occur.

Optimal seam selection. Computing the Voronoi diagram is one way to select the *seams* between regions where different images contribute to the final composite. However, Voronoi images totally ignore the local image structure underlying the seam.

A better approach is to place the seams in regions where the images agree, so that transitions from one source to another are not visible. In this way, the algorithm avoids "cutting through" moving objects where a seam would look unnatural (Davis 1998). For a pair of images, this process can be formulated as a simple dynamic program starting from one edge



(a) (b) (c)

Figure 9.15 Computation of regions of difference (RODs) (Uyttendaele, Eden, and Szeliski 2001) © 2001 IEEE: (a) three overlapping images with a moving face; (b) corresponding RODs; (c) graph of coincident RODs.

of the overlap region and ending at the other (Milgram 1975, 1977; Davis 1998; Efros and Freeman 2001).

When multiple images are being composited, the dynamic program idea does not readily generalize. (For square texture tiles being composited sequentially, Efros and Freeman (2001) run a dynamic program along each of the four tile sides.)

To overcome this problem, Uyttendaele, Eden, and Szeliski (2001) observed that, for well-registered images, moving objects produce the most visible artifacts, namely translucent looking *ghosts*. Their system therefore decides which objects to keep and which ones to erase. First, the algorithm compares all overlapping input image pairs to determine *regions of difference* (RODs) where the images disagree. Next, a graph is constructed with the RODs as vertices and edges representing ROD pairs that overlap in the final composite (Figure 9.15). Since the presence of an edge indicates an area of disagreement, vertices (regions) must be removed from the final composite until no edge spans a pair of remaining vertices. The smallest such set can be computed using a *vertex cover* algorithm. Since several such covers may exist, a *weighted vertex cover* is used instead, where the vertex weights are computed by summing the feather weights in the ROD (Uyttendaele, Eden, and Szeliski 2001). The algorithm therefore prefers removing regions that are near the edge of the image, which reduces the likelihood that partially visible objects will appear in the final composite. (It is also possible to infer which object in a region of difference is the foreground object by the "edginess" (pixel differences) across the ROD boundary, which should be higher when an object is present (Herley 2005).) Once the desired excess regions of difference have been removed, the final composite can be created by feathering (Figure 9.14f).

A different approach to pixel selection and seam placement is described by Agarwala, Dontcheva, Agrawala *et al.* (2004). Their system computes the label assignment that optimizes the sum of two objective functions. The first is a per-pixel *image objective* that determines which pixels are likely to produce good composites,

$$C_D = \sum_{\boldsymbol{x}} D(\boldsymbol{x}, l(\boldsymbol{x})), \tag{9.41}$$

where $D(\boldsymbol{x}, l)$ is the *data penalty* associated with choosing image l at pixel $\boldsymbol{x}$. In their system, users can select which pixels to use by "painting" over an image with the desired object or appearance, which sets $D(\boldsymbol{x}, l)$ to a large value for all labels l other than the one selected by the user (Figure 9.16). Alternatively, automated selection criteria can be used, such as *maximum likelihood*, which prefers pixels that occur repeatedly in the background (for object removal), or *minimum likelihood* for objects that occur infrequently, i.e., for moving object retention. Using a more traditional center-weighted data term tends to favor objects that are



Figure 9.16 Photomontage (Agarwala, Dontcheva, Agrawala *et al.* 2004) © 2004 ACM. From a set of five source images (of which four are shown on the left), Photomontage quickly creates a composite family portrait in which everyone is smiling and looking at the camera (right). Users simply flip through the stack and coarsely draw strokes using the designated source image objective over the people they wish to add to the composite. The user-applied strokes and computed regions (middle) are color-coded by the borders of the source images on the left.

centered in the input images (Figure 9.17).

The second term is a *seam objective* that penalizes differences in labelings between adjacent images,

$$\mathcal{C}_S = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{N}} S(\boldsymbol{x},\boldsymbol{y},l(\boldsymbol{x}),l(\boldsymbol{y})), \tag{9.42}$$

where $S(\boldsymbol{x},\boldsymbol{y},l_x,l_y)$ is the image-dependent *interaction penalty* or *seam cost* of placing a seam between pixels $\boldsymbol{x}$ and $\boldsymbol{y}$, and $\mathcal{N}$ is the set of $\mathcal{N}_4$ neighboring pixels. For example, the simple color-based seam penalty used in (Kwatra, Schödl, Essa *et al.* 2003; Agarwala, Dontcheva, Agrawala *et al.* 2004) can be written as

$$S(\boldsymbol{x},\boldsymbol{y},l_x,l_y) = \|\tilde{I}_{l_x}(\boldsymbol{x}) - \tilde{I}_{l_y}(\boldsymbol{x})\| + \|\tilde{I}_{l_x}(\boldsymbol{y}) - \tilde{I}_{l_y}(\boldsymbol{y})\|. \tag{9.43}$$

More sophisticated seam penalties can also look at image gradients or the presence of image edges (Agarwala, Dontcheva, Agrawala *et al.* 2004). Seam penalties are widely used in other computer vision applications such as stereo matching (Boykov, Veksler, and Zabih 2001) to give the labeling function its *coherence* or *smoothness*. An alternative approach, which places seams along strong consistent edges in overlapping images using a watershed computation is described by Soille (2006).

The sum of these two objective functions gives rise to a *Markov random field* (MRF), for which good optimization algorithms are described in Sections 3.7.2 and 5.5 and Appendix B.5. For label computations of this kind, the α-*expansion* algorithm developed by Boykov, Veksler, and Zabih (2001) works particularly well (Szeliski, Zabih, Scharstein *et al.* 2008).

For the result shown in Figure 9.14g, Agarwala, Dontcheva, Agrawala *et al.* (2004) use a large data penalty for invalid pixels and 0 for valid pixels. Notice how the seam placement algorithm avoids regions of difference, including those that border the image and that might result in objects being cut off. Graph cuts (Agarwala, Dontcheva, Agrawala *et al.* 2004) and vertex cover (Uyttendaele, Eden, and Szeliski 2001) often produce similar looking results, although the former is significantly slower since it optimizes over all pixels, while the latter is more sensitive to the thresholds used to determine regions of difference.



Figure 9.17 Set of five photos tracking a snowboarder's jump stitched together into a seamless composite. Because the algorithm prefers pixels near the center of the image, multiple copies of the boarder are retained.

9.3.3 *Application*: Photomontage

While image stitching is normally used to composite partially overlapping photographs, it can also be used to composite repeated shots of a scene taken with the aim of obtaining the best possible composition and appearance of each element.

Figure 9.16 shows the *Photomontage* system developed by Agarwala, Dontcheva, Agrawala *et al.* (2004), where users draw strokes over a set of pre-aligned images to indicate which regions they wish to keep from each image. Once the system solves the resulting multi-label graph cut (9.41–9.42), the various pieces taken from each source photo are blended together using a variant of Poisson image blending (9.44–9.46). Their system can also be used to automatically composite an all-focus image from a series of bracketed focus images (Hasinoff, Kutulakos, Durand *et al.* 2009) or to remove wires and other unwanted elements from sets of photographs. Exercise 9.10 has you implement this system and try out some of its variants.

9.3.4 Blending

Once the seams between images have been determined and unwanted objects removed, we still need to blend the images to compensate for exposure differences and other mis-alignments. The spatially varying weighting (feathering) previously discussed can often be used to accomplish this. However, it is difficult in practice to achieve a pleasing balance between smoothing out low-frequency exposure variations and retaining sharp enough transitions to prevent blurring (although using a high exponent in feathering can help).

Laplacian pyramid blending. An attractive solution to this problem is the Laplacian pyramid blending technique developed by Burt and Adelson (1983b), which we discussed in Section 3.5.5. Instead of using a single transition width, a frequency-adaptive width is used by creating a band-pass (Laplacian) pyramid and making the transition widths within each level



(a) (b) (c)

Figure 9.18 Poisson image editing (Pérez, Gangnet, and Blake 2003) © 2003 ACM: (a) The dog and the two children are chosen as source images to be pasted into the destination swimming pool. (b) Simple pasting fails to match the colors at the boundaries, whereas (c) Poisson image blending masks these differences.

a function of the level, i.e., the same width in pixels. In practice, a small number of levels, i.e., as few as two (Brown and Lowe 2007), may be adequate to compensate for differences in exposure. The result of applying this pyramid blending is shown in Figure 9.14h.

Gradient domain blending. An alternative approach to multi-band image blending is to perform the operations in the *gradient domain*. Reconstructing images from their gradient fields has a long history in computer vision (Horn 1986), starting originally with work in brightness constancy (Horn 1974), shape from shading (Horn and Brooks 1989), and photometric stereo (Woodham 1981). More recently, related ideas have been used for reconstructing images from their edges (Elder and Goldberg 2001), removing shadows from images (Weiss 2001), separating reflections from a single image (Levin, Zomet, and Weiss 2004; Levin and Weiss 2007), and *tone mapping* high dynamic range images by reducing the magnitude of image edges (gradients) (Fattal, Lischinski, and Werman 2002).

Pérez, Gangnet, and Blake (2003) show how gradient domain reconstruction can be used to do seamless object insertion in image editing applications (Figure 9.18). Rather than copying pixels, the *gradients* of the new image fragment are copied instead. The actual pixel values for the copied area are then computed by solving a *Poisson equation* that locally matches the gradients while obeying the fixed *Dirichlet* (exact matching) conditions at the seam boundary. Pérez, Gangnet, and Blake (2003) show that this is equivalent to computing an additive *membrane* interpolant of the mismatch between the source and destination images along the boundary.[14] In earlier work, Peleg (1981) also proposed adding a smooth function to enforce consistency along the seam curve.

Agarwala, Dontcheva, Agrawala *et al.* (2004) extended this idea to a multi-source formulation, where it no longer makes sense to talk of a destination image whose exact pixel values must be matched at the seam. Instead, *each* source image contributes its own gradient field and the Poisson equation is solved using *Neumann* boundary conditions, i.e., dropping any

[14] The membrane interpolant is known to have nicer interpolation properties for arbitrary-shaped constraints than frequency-domain interpolants (Nielson 1993).

equations that involve pixels outside the boundary of the image.

Rather than solving the Poisson partial differential equations, Agarwala, Dontcheva, Agrawala *et al.* (2004) directly minimize a *variational problem*,

$$\min_{C(\boldsymbol{x})} \|\nabla C(\boldsymbol{x}) - \nabla \tilde{I}_{l(\boldsymbol{x})}(\boldsymbol{x})\|^2. \tag{9.44}$$

The discretized form of this equation is a set of gradient constraint equations

$$C(\boldsymbol{x}+\hat{\boldsymbol{\imath}}) - C(\boldsymbol{x}) = \tilde{I}_{l(\boldsymbol{x})}(\boldsymbol{x}+\hat{\boldsymbol{\imath}}) - \tilde{I}_{l(\boldsymbol{x})}(\boldsymbol{x}) \text{ and} \tag{9.45}$$

$$C(\boldsymbol{x}+\hat{\boldsymbol{\jmath}}) - C(\boldsymbol{x}) = \tilde{I}_{l(\boldsymbol{x})}(\boldsymbol{x}+\hat{\boldsymbol{\jmath}}) - \tilde{I}_{l(\boldsymbol{x})}(\boldsymbol{x}), \tag{9.46}$$

where $\hat{\boldsymbol{\imath}} = (1, 0)$ and $\hat{\boldsymbol{\jmath}} = (0, 1)$ are unit vectors in the x and y directions.[15] They then solve the associated sparse least squares problem. Since this system of equations is only defined up to an additive constraint, Agarwala, Dontcheva, Agrawala *et al.* (2004) ask the user to select the value of one pixel. In practice, a better choice might be to weakly bias the solution towards reproducing the original color values.

In order to accelerate the solution of this sparse linear system, Fattal, Lischinski, and Werman (2002) use multigrid, whereas Agarwala, Dontcheva, Agrawala *et al.* (2004) use hierarchical basis preconditioned conjugate gradient descent (Szeliski 1990b, 2006b) (Appendix A.5). In subsequent work, Agarwala (2007) shows how using a quadtree representation for the solution can further accelerate the computation with minimal loss in accuracy, while Szeliski, Uyttendaele, and Steedly (2008) show how representing the per-image offset fields using even coarser splines is even faster. This latter work also argues that blending in the log domain, i.e., using multiplicative rather than additive offsets, is preferable, as it more closely matches texture contrasts across seam boundaries. The resulting seam blending works very well in practice (Figure 9.14h), although care must be taken when copying large gradient values near seams so that a "double edge" is not introduced.

Copying gradients directly from the source images after seam placement is just one approach to gradient domain blending. The paper by Levin, Zomet, Peleg *et al.* (2004) examines several different variants of this approach, which they call *Gradient-domain Image STitching* (GIST). The techniques they examine include feathering (blending) the gradients from the source images, as well as using an L1 norm in performing the reconstruction of the image from the gradient field, rather than using an L2 norm as in Equation (9.44). Their preferred technique is the L1 optimization of a feathered (blended) cost function on the original image gradients (which they call GIST1-l_1). Since L1 optimization using linear programming can be slow, they develop a faster iterative median-based algorithm in a multigrid framework. Visual comparisons between their preferred approach and what they call *optimal seam on the gradients* (which is equivalent to the approach of Agarwala, Dontcheva, Agrawala *et al.* (2004)) show similar results, while significantly improving on pyramid blending and feathering algorithms.

Exposure compensation. Pyramid and gradient domain blending can do a good job of compensating for moderate amounts of exposure differences between images. However, when the exposure differences become large, alternative approaches may be necessary.

[15] At seam locations, the right hand side is replaced by the average of the gradients in the two source images.

Uyttendaele, Eden, and Szeliski (2001) iteratively estimate a local correction between each source image and a blended composite. First, a block-based quadratic transfer function is fit between each source image and an initial feathered composite. Next, transfer functions are averaged with their neighbors to get a smoother mapping and per-pixel transfer functions are computed by *splining* (interpolating) between neighboring block values. Once each source image has been smoothly adjusted, a new feathered composite is computed and the process is repeated (typically three times). The results shown by Uyttendaele, Eden, and Szeliski (2001) demonstrate that this does a better job of exposure compensation than simple feathering and can handle local variations in exposure due to effects such as lens vignetting.

Ultimately, however, the most principled way to deal with exposure differences is to stitch images in the radiance domain, i.e., to convert each image into a radiance image using its exposure value and then create a stitched, high dynamic range image, as discussed in Section 10.2 (Eden, Uyttendaele, and Szeliski 2006).

9.4 Additional reading

The literature on image stitching dates back to work in the photogrammetry community in the 1970s (Milgram 1975, 1977; Slama 1980). In computer vision, papers started appearing in the early 1980s (Peleg 1981), while the development of fully automated techniques came about a decade later (Mann and Picard 1994; Chen 1995; Szeliski 1996; Szeliski and Shum 1997; Sawhney and Kumar 1999; Shum and Szeliski 2000). Those techniques used direct pixel-based alignment but feature-based approaches are now the norm (Zoghlami, Faugeras, and Deriche 1997; Capel and Zisserman 1998; Cham and Cipolla 1998; Badra, Qumsieh, and Dudek 1998; McLauchlan and Jaenicke 2002; Brown and Lowe 2007). A collection of some of these papers can be found in the book by Benosman and Kang (2001). Szeliski (2006a) provides a comprehensive survey of image stitching, on which the material in this chapter is based.

High-quality techniques for optimal seam finding and blending are another important component of image stitching systems. Important developments in this field include work by Milgram (1977), Burt and Adelson (1983b), Davis (1998), Uyttendaele, Eden, and Szeliski (2001),Pérez, Gangnet, and Blake (2003), Levin, Zomet, Peleg *et al.* (2004), Agarwala, Dontcheva, Agrawala *et al.* (2004), Eden, Uyttendaele, and Szeliski (2006), and Kopf, Uyttendaele, Deussen *et al.* (2007).

In addition to the merging of multiple overlapping photographs taken for aerial or terrestrial panoramic image creation, stitching techniques can be used for automated whiteboard scanning (He and Zhang 2005; Zhang and He 2007), scanning with a mouse (Nakao, Kashitani, and Kaneyoshi 1998), and retinal image mosaics (Can, Stewart, Roysam *et al.* 2002). They can also be applied to video sequences (Teodosio and Bender 1993; Irani, Hsu, and Anandan 1995; Kumar, Anandan, Irani *et al.* 1995; Sawhney and Ayer 1996; Massey and Bender 1996; Irani and Anandan 1998; Sawhney, Arpa, Kumar *et al.* 2002; Agarwala, Zheng, Pal *et al.* 2005; Rav-Acha, Pritch, Lischinski *et al.* 2005; Steedly, Pal, and Szeliski 2005; Baudisch, Tan, Steedly *et al.* 2006) and can even be used for video compression (Lee, ge Chen, lung Bruce Lin *et al.* 1997).

9.5 Exercises

Ex 9.1: Direct pixel-based alignment Take a pair of images, compute a coarse-to-fine affine alignment (Exercise 8.2) and then blend them using either averaging (Exercise 6.2) or a Laplacian pyramid (Exercise 3.20). Extend your motion model from affine to perspective (homography) to better deal with rotational mosaics and planar surfaces seen under arbitrary motion.

Ex 9.2: Featured-based stitching Extend your feature-based alignment technique from Exercise 6.2 to use a full perspective model and then blend the resulting mosaic using either averaging or more sophisticated distance-based feathering (Exercise 9.9).

Ex 9.3: Cylindrical strip panoramas To generate cylindrical or spherical panoramas from a horizontally panning (rotating) camera, it is best to use a tripod. Set your camera up to take a series of 50% overlapped photos and then use the following steps to create your panorama:

1. Estimate the amount of radial distortion by taking some pictures with lots of long straight lines near the edges of the image and then using the plumb-line method from Exercise 6.10.
2. Compute the focal length either by using a ruler and paper, as in Figure 6.7 (Debevec, Wenger, Tchou *et al.* 2002) or by rotating your camera on the tripod, overlapping the images by exactly 0% and counting the number of images it takes to make a 360° panorama.
3. Convert each of your images to cylindrical coordinates using (9.12–9.16).
4. Line up the images with a translational motion model using either a direct pixel-based technique, such as coarse-to-fine incremental or an FFT, or a feature-based technique.
5. (Optional) If doing a complete 360° panorama, align the first and last images. Compute the amount of accumulated vertical mis-registration and re-distribute this among the images.
6. Blend the resulting images using feathering or some other technique.

Ex 9.4: Coarse alignment Use FFT or phase correlation (Section 8.1.2) to estimate the initial alignment between successive images. How well does this work? Over what range of overlaps? If it does not work, does aligning sub-sections (e.g., quarters) do better?

Ex 9.5: Automated mosaicing Use feature-based alignment with four-point RANSAC for homographies (Section 6.1.3, Equations (6.19–6.23)) or three-point RANSAC for rotational motions (Brown, Hartley, and Nistér 2007) to match up all pairs of overlapping images.

Merge these pairwise estimates together by finding a spanning tree of pairwise relations. Visualize the resulting global alignment, e.g., by displaying a blend of each image with all other images that overlap it.

For greater robustness, try multiple spanning trees (perhaps randomly sampled based on the confidence in pairwise alignments) to see if you can recover from bad pairwise matches (Zach, Klopschitz, and Pollefeys 2010). As a measure of fitness, count how many pairwise estimates are consistent with the global alignment.

Ex 9.6: Global optimization Use the initialization from the previous algorithm to perform a full bundle adjustment over all of the camera rotations and focal lengths, as described in Section 7.4 and by Shum and Szeliski (2000). Optionally, estimate radial distortion parameters as well or support fisheye lenses (Section 2.1.6).

As in the previous exercise, visualize the quality of your registration by creating composites of each input image with its neighbors, optionally blinking between the original image and the composite to better see mis-alignment artifacts.

Ex 9.7: De-ghosting Use the results of the previous bundle adjustment to predict the location of each feature in a consensus geometry. Use the difference between the predicted and actual feature locations to correct for small mis-registrations, as described in Section 9.2.2 (Shum and Szeliski 2000).

Ex 9.8: Compositing surface Choose a compositing surface (Section 9.3.1), e.g., a single reference image extended to a larger plane, a sphere represented using cylindrical or spherical coordinates, a stereographic "little planet" projection, or a cube map.

Project all of your images onto this surface and blend them with equal weighting, for now (just to see where the original image seams are).

Ex 9.9: Feathering and blending Compute a feather (distance) map for each warped source image and use these maps to blend the warped images.

Alternatively, use Laplacian pyramid blending (Exercise 3.20) or gradient domain blending.

Ex 9.10: Photomontage and object removal Implement a "PhotoMontage" system in which users can indicate desired or unwanted regions in pre-registered images using strokes or other primitives (such as bounding boxes).

(Optional) Devise an automatic moving objects remover (or "keeper") by analyzing which inconsistent regions are more or less typical given some consensus (e.g., median filtering) of the aligned images. Figure 9.17 shows an example where the moving object was kept. Try to make this work for sequences with large amounts of overlaps and consider averaging the images to make the moving object look more ghosted.

Chapter 10

Computational photography

10.1 Photometric calibration 412
10.1.1 Radiometric response function 412
10.1.2 Noise level estimation 415
10.1.3 Vignetting 416
10.1.4 Optical blur (spatial response) estimation 416
10.2 High dynamic range imaging 419
10.2.1 Tone mapping 427
10.2.2 *Application*: Flash photography 434
10.3 Super-resolution and blur removal 436
10.3.1 Color image demosaicing 440
10.3.2 *Application*: Colorization 442
10.4 Image matting and compositing 443
10.4.1 Blue screen matting 445
10.4.2 Natural image matting 446
10.4.3 Optimization-based matting 450
10.4.4 Smoke, shadow, and flash matting 452
10.4.5 Video matting 454
10.5 Texture analysis and synthesis 455
10.5.1 *Application*: Hole filling and inpainting 457
10.5.2 *Application*: Non-photorealistic rendering 458
10.6 Additional reading 460
10.7 Exercises 461

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_10, © Springer-Verlag London Limited 2011



(a)



(b)



(c)



(d)

Figure 10.1 Computational photography: (a) merging multiple exposures to create high dynamic range images (Debevec and Malik 1997) © 1997 ACM; (b) merging flash and non-flash photographs; (Petschnigg, Agrawala, Hoppe *et al.* 2004) © 2004 ACM; (c) image matting and compositing; (Chuang, Curless, Salesin *et al.* 2001) © 2001 IEEE; (d) hole filling with inpainting (Criminisi, Pérez, and Toyama 2004) © 2004 IEEE.

Stitching multiple images into wide field of view panoramas, which we covered in Chapter 9, allows us create photographs that could not be captured with a regular camera. This is just one instance of *computational photography*, where image analysis and processing algorithms are applied to one or more photographs to create images that go beyond the capabilities of traditional imaging systems. Some of these techniques are now being incorporated directly into digital still cameras. For example, some of the newer digital still cameras have sweep panorama modes and take multiple shots in low-light conditions to reduce image noise.

In this chapter, we cover a number of additional computational photography algorithms. We begin with a review of photometric image calibration (Section 10.1), i.e., the measurement of camera and lens responses, which is a prerequisite for many of the algorithms we describe later. We then discuss *high dynamic range imaging* (Section 10.2), which captures the full range of brightness in a scene through the use of multiple exposures (Figure 10.1a). We also discuss *tone mapping operators*, which map rich images back into regular display devices, such as screens and printers, as well as algorithms that merge flash and regular images to obtain better exposures (Figure 10.1b).

Next, we discuss how the resolution of images can be improved either by merging multiple photographs together or using sophisticated image priors (Section 10.3). This includes algorithms for extracting full-color images from the patterned Bayer mosaics present in most cameras.

In Section 10.4, we discuss algorithms for cutting pieces of images from one photograph and pasting them into others (Figure 10.1c). In Section 10.5, we describe how to generate novel textures from real-world samples for applications such as filling holes in images (Figure 10.1d). We close with a brief overview of *non-photorealistic rendering* (Section 10.5.2), which can turn regular photographs into artistic renderings that resemble traditional drawings and paintings.

One topic that we do not cover extensively in this book is novel computational sensors, optics, and cameras. A nice survey can be found in an article by Nayar (2006), a recently published book by Raskar and Tumblin (2010), and more recent research papers (Levin, Fergus, Durand *et al.* 2007). Some related discussion can also be found in Sections 10.2 and 13.3.

A good general-audience introduction to computational photography can be found in the article by Hayes (2008) as well as survey papers by Nayar (2006), Cohen and Szeliski (2006), Levoy (2006), and Debevec (2006).[1] Raskar and Tumblin (2010) give extensive coverage of topics in this area, with particular emphasis on computational cameras and sensors. The sub-field of high dynamic range imaging has its own book discussing research in this area (Reinhard, Ward, Pattanaik *et al.* 2005), as well as a wonderful book aimed more at professional photographers (Freeman 2008).[2] A good survey of image matting is provided by Wang and Cohen (2007a).

There are also several courses on computational photography where the instructors have provided extensive on-line materials, e.g., Frédo Durand's Computation Photography course at MIT,[3] Alyosha Efros' class at Carnegie Mellon,[4] Marc Levoy's class at Stanford,[5] and a

[1] See also the two special issue journals edited by Bimber (2006) and Durand and Szeliski (2007).
[2] Gulbins and Gulbins (2009) discuss related photographic techniques.
[3] MIT 6.815/6.865, http://stellar.mit.edu/S/course/6/sp08/6.815/materials.html.
[4] CMU 15-463, http://graphics.cs.cmu.edu/courses/15-463/.
[5] Stanford CS 448A, http://graphics.stanford.edu/courses/cs448a-10/.

series of SIGGRAPH courses on Computational Photography.[6]

10.1 Photometric calibration

Before we can successfully merge multiple photographs, we need to characterize the functions that map incoming irradiance into pixel values and also the amounts of noise present in each image. In this section, we examine three components of the imaging pipeline (Figure 10.2) that affect this mapping.

The first is the *radiometric response function* (Mitsunaga and Nayar 1999), which maps photons arriving at the lens into digital values stored in the image file (Section 10.1.1). The second is *vignetting*, which darkens pixel values near the periphery of images, especially at large apertures (Section 10.1.3). The third is the *point spread function*, which characterizes the blur induced by the lens, anti-aliasing filters, and finite sensor areas (Section 10.1.4).[7] The material in this section builds on the image formation processes described in Sections 2.2.3 and 2.3.3, so if it has been a while since you looked at those sections, please go back and review them.

10.1.1 Radiometric response function

As we can see in Figure 10.2, a number of factors affect how the intensity of light arriving at the lens ends up being mapped into stored digital values. Let us ignore for now any non-uniform attenuation that may occur inside the lens, which we cover in Section 10.1.3.

The first factors to affect this mapping are the aperture and shutter speed (Section 2.3), which can be modeled as global multipliers on the incoming light, most conveniently measured in *exposure values* ($\log_2$ brightness ratios). Next, the analog to digital (A/D) converter on the sensing chip applies an electronic gain, usually controlled by the ISO setting on your camera. While in theory this gain is linear, as with any electronics non-linearities may be present (either unintentionally or by design). Ignoring, for now, photon noise, on-chip noise, amplifier noise, and quantization noise, which we discuss shortly, you can often assume that the mapping between incoming light and the values stored in a RAW camera file (if your camera supports this) is roughly linear.

If images are being stored in the more common JPEG format, the camera's digital signal processor (DSP) next performs Bayer pattern demosaicing (Sections 2.3.2 and 10.3.1), which is a mostly linear (but often non-stationary) process. Some sharpening is also often applied at this stage. Next, the color values are multiplied by different constants (or sometimes a 3×3 color twist matrix) to perform color balancing, i.e., to move the white point closer to pure white. Finally, a standard gamma is applied to the intensities in each color channel and the colors are converted into YCbCr format before being transformed by a DCT, quantized, and then compressed into the JPEG format (Section 2.3.3). Figure 10.2 shows all of these steps in pictorial form.

Given the complexity of all of this processing, it is difficult to model the camera response function (Figure 10.3a), i.e., the mapping between incoming irradiance and digital RGB val-

[6] http://web.media.mit.edu/~raskar/photo/.

[7] Additional photometric camera and lens effects include sensor glare, blooming, and chromatic aberration, which can also be thought of as a spectrally varying form of geometric aberration (Section 2.2.3).

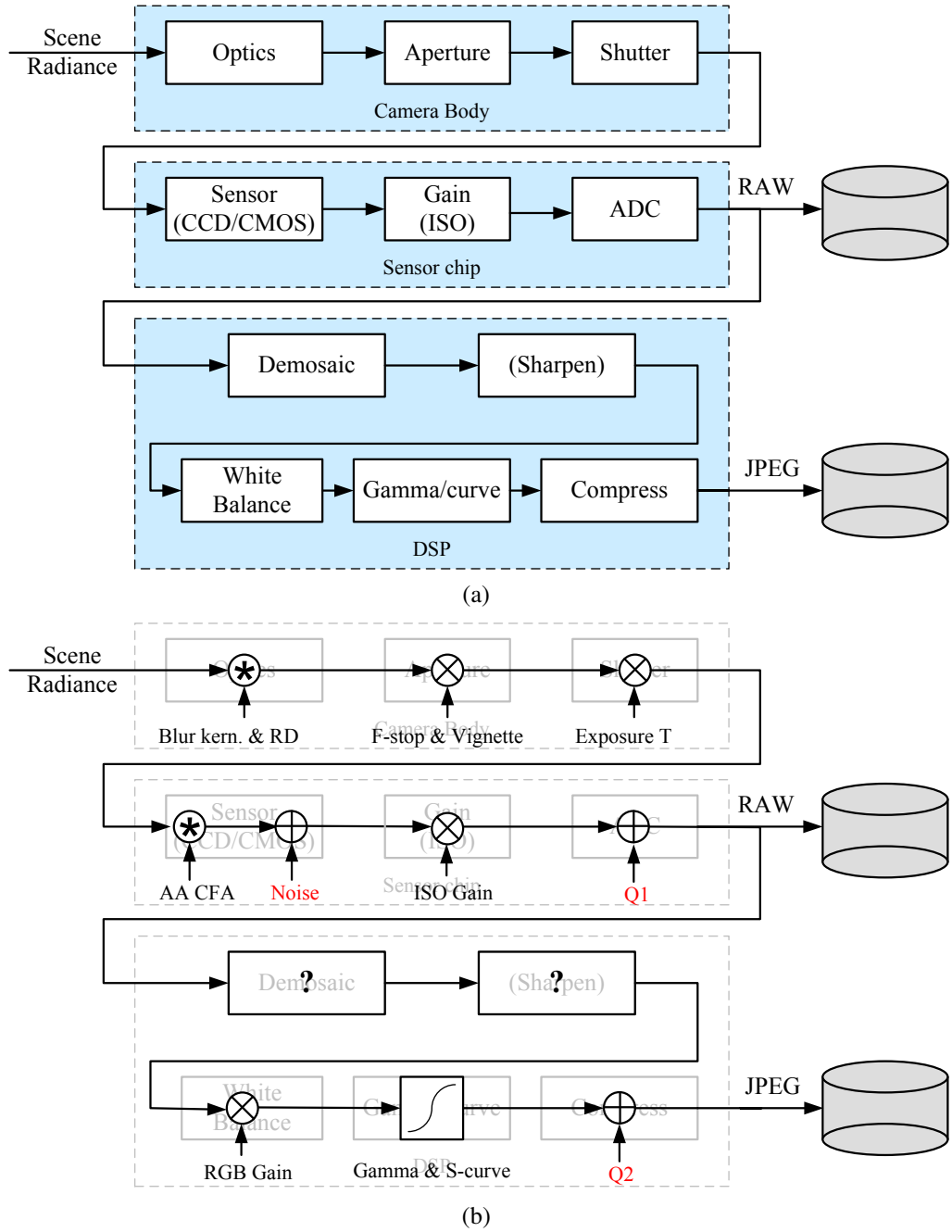
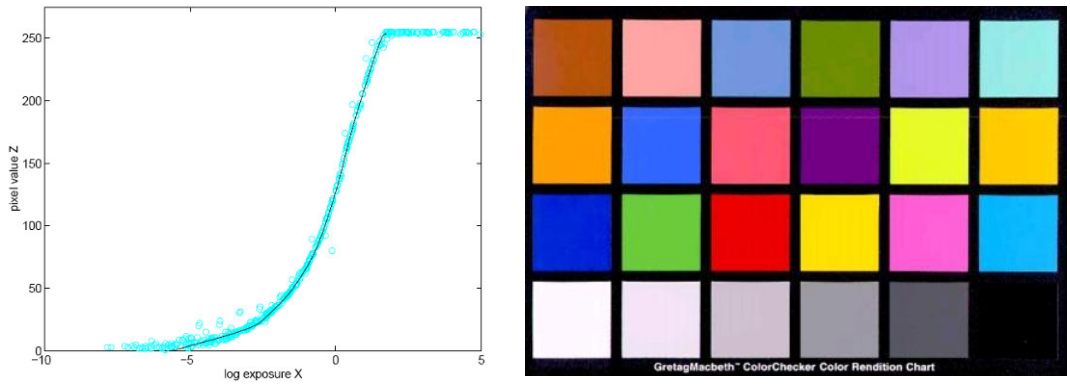


Figure 10.2 Image sensing pipeline: (a) block diagram showing the various sources of noise as well as the typical digital post-processing steps; (b) equivalent signal transforms, including convolution, gain, and noise injection. The abbreviations are: RD = radial distortion, AA = anti-aliasing filter, CFA = color filter array, Q1 and Q2 = quantization noise.



(a) (b)

Figure 10.3 Radiometric response calibration: (a) typical camera response function, showing the mapping between incoming log irradiance (exposure) and output eight-bit pixel values, for one color channel (Debevec and Malik 1997) © 1997 ACM; (b) color checker chart.

ues, from first principles. A more practical approach is to calibrate the camera by measuring correspondences between incoming light and final values.

The most accurate, but most expensive, approach is to use an *integrating sphere*, which is a large (typically 1m diameter) sphere carefully painted on the inside with white matte paint. An accurately calibrated light at the top controls the amount of radiance inside the sphere (which is constant everywhere because of the sphere's radiometry) and a small opening at the side allows for a camera/lens combination to be mounted. By slowly varying the current going into the light, an accurate correspondence can be established between incoming radiance and measured pixel values. The vignetting and noise characteristics of the camera can also be simultaneously determined.

A more practical alternative is to use a calibration chart (Figure 10.3b) such as the Macbeth or Munsell ColorChecker Chart.[8] The biggest problem with this approach is to ensure uniform lighting. One approach is to use a large dark room with a high-quality light source far away from (and perpendicular to) the chart. Another is to place the chart outdoors away from any shadows. (The results will differ under these two conditions, because the color of the illuminant will be different).

The easiest approach is probably to take multiple exposures of the same scene while the camera is on a tripod and to recover the response function by simultaneously estimating the incoming irradiance at each pixel and the response curve (Mann and Picard 1995; Debevec and Malik 1997; Mitsunaga and Nayar 1999). This approach is discussed in more detail in Section 10.2 on high dynamic range imaging.

If all else fails, i.e., you just have one or more unrelated photos, you can use an International Color Consortium (ICC) profile for the camera (Fairchild 2005).[9] Even more simply, you can just assume that the response is linear if they are RAW files and that the images have a $\gamma = 2.2$ non-linearity (plus clipping) applied to each RGB channel if they are JPEG images.

[8] http://www.xrite.com.

[9] See also the ICC *Information on Profiles*, http://www.color.org/info_profiles2.xalter.

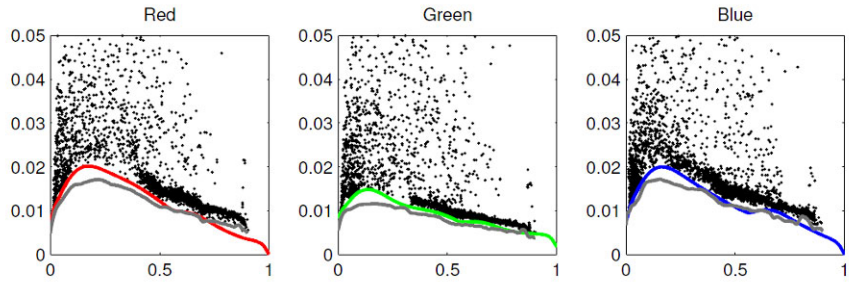


Figure 10.4 Noise level function estimates obtained from a single color photograph (Liu, Szeliski, Kang *et al.* 2008) © 2008 IEEE. The colored curves are the estimated NLF fit as the probabilistic lower envelope of the measured deviations between the noisy piecewise-smooth images. The ground truth NLFs obtained by averaging 29 images are shown in gray.

10.1.2 Noise level estimation

In addition to knowing the camera response function, it is also often important to know the amount of noise being injected under a particular camera setting (e.g., ISO/gain level). The simplest characterization of noise is a single standard deviation, usually measured in gray levels, independent of pixel value. A more accurate model can be obtained by estimating the noise level as a function of pixel value (Figure 10.4), which is known as the *noise level function* (Liu, Szeliski, Kang *et al.* 2008).

As with the camera response function, the simplest way to estimate these quantities is in the lab, using either an integrating sphere or a calibration chart. The noise can be estimated either at each pixel independently, by taking repeated exposures and computing the temporal variance in the measurements (Healey and Kondepudy 1994), or over regions, by assuming that pixel values should all be the same within some region (e.g., inside a color checker square) and computing a spatial variance.

This approach can be generalized to photos where there are regions of constant or slowly varying intensity (Liu, Szeliski, Kang *et al.* 2008). First, segment the image into such regions and fit a constant or linear function inside each region. Next, measure the (spatial) standard deviation of the differences between the noisy input pixels and the smooth fitted function away from large gradients and region boundaries. Plot these as a function of output level for each color channel, as shown in Figure 10.4. Finally, fit a lower envelope to this distribution in order to ignore pixels or deviations that are outliers. A fully Bayesian approach to this problem that models the statistical distribution of each quantity is presented by (Liu, Szeliski, Kang *et al.* 2008). A simpler approach, which should produce useful results in most cases, is to fit a low-dimensional function (e.g., positive valued B-spline) to the lower envelope (see Exercise 10.2).

In more recent work, Matsushita and Lin (2007) present a technique for simultaneously estimating a camera's response and noise level functions based on skew (asymmetries) in level-dependent noise distributions. Their paper also contains extensive references to previous work in these areas.



Figure 10.5 Single image vignetting correction (Zheng, Yu, Kang *et al.* 2008) © 2008 IEEE: (a) original image with strong visible vignetting; (b) vignetting compensation as described by Zheng, Zhou, Georgescu *et al.* (2006); (c–d) vignetting compensation as described by Zheng, Yu, Kang *et al.* (2008).

10.1.3 Vignetting

A common problem with using wide-angle and wide-aperture lenses is that the image tends to darken in the corners (Figure 10.5a). This problem is generally known as *vignetting* and comes in several different forms, including natural, optical, and mechanical vignetting (Section 2.2.3) (Ray 2002). As with radiometric response function calibration, the most accurate way to calibrate vignetting is to use an integrating sphere or a picture of a uniformly colored and illuminated blank wall.

An alternative approach is to stitch a panoramic scene and to assume that the true radiance at each pixel comes from the central portion of each input image. This is easier to do if the radiometric response function is already known (e.g., by shooting in RAW mode) and if the exposure is kept constant. If the response function, image exposures, and vignetting function are unknown, they can still be recovered by optimizing a large least squares fitting problem (Litvinov and Schechner 2005; Goldman 2011). Figure 10.6 shows an example of simultaneously estimating the vignetting, exposure, and radiometric response function from a set of overlapping photographs (Goldman 2011). Note that unless vignetting is modeled and compensated, regular gradient-domain image blending (Section 9.3.4) will not create an attractive image.

If only a single image is available, vignetting can be estimated by looking for slow consistent intensity variations in the radial direction. The original algorithm proposed by Zheng, Lin, and Kang (2006) first pre-segmented the image into smoothly varying regions and then performed an analysis inside each region. Instead of pre-segmenting the image, Zheng, Yu, Kang *et al.* (2008) compute the radial gradients at all the pixels and use the asymmetry in this distribution (since gradients away from the center are, on average, slightly negative) to estimate the vignetting. Figure 10.5 shows the results of applying each of these algorithms to an image with a large amount of vignetting. Exercise 10.3 has you implement some of the above techniques.

10.1.4 Optical blur (spatial response) estimation

One final characteristic of imaging systems that you should calibrate is the spatial response function, which encodes the optical blur that gets convolved with the incoming image to produce the point-sampled image. The shape of the convolution kernel, which is also known as *point spread function (PSF)* or *optical transfer function*, depends on several factors, including lens blur and radial distortion (Section 2.2.3), anti-aliasing filters in front of the sensor, and



Figure 10.6 Simultaneous estimation of vignetting, exposure, and radiometric response (Goldman 2011) © 2011 IEEE: (a) original average of the input images; (b) after compensating for vignetting; (c) using gradient domain blending only (note the remaining mottled look); (d) after both vignetting compensation and blending.

the shape and extent of each active pixel area (Section 2.3) (Figure 10.2). A good estimate of this function is required for applications such as multi-image super-resolution and de-blurring (Section 10.3).

In theory, one could estimate the PSF by simply observing an infinitely small point light source everywhere in the image. Creating an array of samples by drilling through a dark plate and backlighting with a very bright light source is difficult in practice.

A more practical approach is to observe an image composed of long straight lines or bars, since these can be fitted to arbitrary precision. Because the location of a horizontal or vertical edge can be *aliased* during acquisition, slightly slanted edges are preferred. The profile and locations of such edges can be estimated to sub-pixel precision, which makes it possible to estimate the PSF at sub-pixel resolutions (Reichenbach, Park, and Narayanswamy 1991; Burns and Williams 1999; Williams and Burns 2001; Goesele, Fuchs, and Seidel 2003). The thesis by Murphy (2005) contains a nice survey of all aspects of camera calibration, including the spatial frequency response (SFR), spatial uniformity, tone reproduction, color reproduction, noise, dynamic range, color channel registration, and depth of field. It also includes a description of a slant-edge calibration algorithm called `sfrmat2`.

The slant-edge technique can be used to recover a 1D projection of the 2D PSF, e.g., slightly vertical edges are used to recover the horizontal *line spread function* (LSF) (Williams 1999). The LSF is then often converted into the Fourier domain and its magnitude plotted as a one-dimensional *modulation transfer function* (MTF), which indicates which image frequencies are lost (blurred) and aliased during the acquisition process (Section 2.3.1). For most computational photography applications, it is preferable to directly estimate the full 2D PSF, since it can be hard to recover from its projections (Williams 1999).

Figure 10.7 shows a pattern containing edges at all orientations, which can be used to directly recover a two-dimensional PSF. First, corners in the pattern are located by extracting edges in the sensed image, linking them, and finding the intersections of the circular arcs. Next, the ideal pattern, whose analytic form is known, is warped (using a homography) to fit the central portion of the input image and its intensities are adjusted to fit the ones in

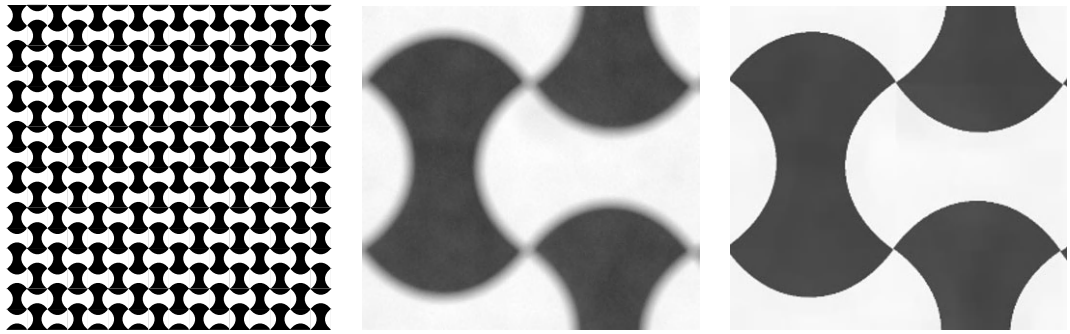


Figure 10.7 Calibration pattern with edges equally distributed at all orientations that can be used for PSF and radial distortion estimation (Joshi, Szeliski, and Kriegman 2008) © 2008 IEEE. A portion of an actual sensed image is shown in the middle and a close-up of the ideal pattern is on the right.

the sensed image. If desired, the pattern can be rendered at a higher resolution than the input image, which enables the estimation of the PSF to sub-pixel resolution (Figure 10.8a). Finally a large linear least squares system is solved to recover the unknown PSF kernel K,

$$K = \arg\min_K \|B - D(I * K)\|^2, \tag{10.1}$$

where B is the sensed (blurred) image, I is the predicted (sharp) image, and D is an optional downsampling operator that matches the resolution of the ideal and sensed images (Joshi, Szeliski, and Kriegman 2008). In terms of the notation (3.75) introduced in Section 3.4.3, this could also be written as

$$b = \arg\min_b \|o - D(s * b)\|^2, \tag{10.2}$$

where o is the observed image, s is the sharp image, and b is the blur kernel.

If the process of estimating the PSF is done locally in overlapping patches of the image, it can also be used to estimate the radial distortion and chromatic aberration induced by the lens (Figure 10.8b). Because the homography mapping the ideal target to the sensed image is estimated in the central (undistorted) part of the image, any (per-channel) shifts induced by the optics manifest themselves as a displacement in the PSF centers.[10] Compensating for these shifts eliminates both the achromatic radial distortion and the inter-channel shifts that result in visible chromatic aberration. The color-dependent blurring caused by chromatic aberration (Figure 2.21) can also be removed using the de-blurring techniques discussed in Section 10.3. Figure 10.8b shows how the radial distortion and chromatic aberration manifest themselves as elongated and displaced PSFs, along with the result of removing these effects in a region of the calibration target.

The local 2D PSF estimation technique can also be used to estimate vignetting. Figure 10.8c shows how the mechanical vignetting manifests itself as clipping of the PSF in the corners of the image. In order for the overall dimming associated with vignetting to be properly captured, the modified intensities of the ideal pattern need to be extrapolated from the center, which is best done with a uniformly illuminated target.

[10] This process confounds the distinction between geometric and photometric calibration. In principle, any geometric distortion could be modeled by spatially varying displaced PSFs. In practice, it is easier to fold any large shifts into the geometric correction component.

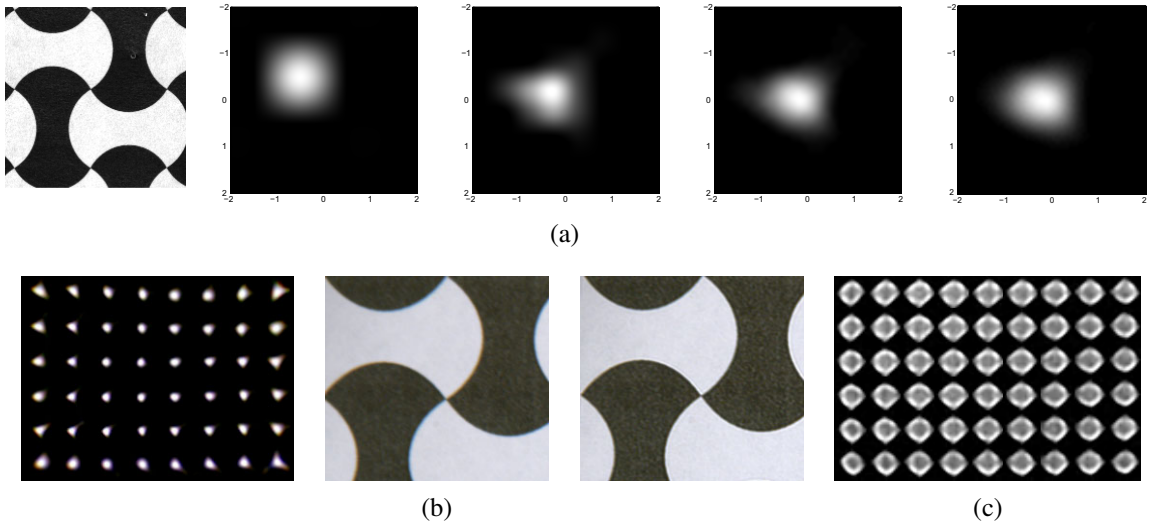


Figure 10.8 Point spread function estimation using a calibration target (Joshi, Szeliski, and Kriegman 2008) © 2008 IEEE. (a) Sub-pixel PSFs at successively higher resolutions (note the interaction between the square sensing area and the circular lens blur). (b) The radial distortion and chromatic aberration can also be estimated and removed. (c) PSF for a mis-focused (blurred) lens showing some diffraction and vignetting effects in the corners.

When working with RAW Bayer-pattern images, the correct way to estimate the PSF is to only evaluate the least squares terms in (10.1) at sensed pixel values, while interpolating the ideal image to all values. For JPEG images, you should linearize your intensities first, e.g., remove the gamma and any other non-linearities in your estimated radiometric response function.

What if you have an image that was taken with an uncalibrated camera? Can you still recover the PSF an use it to correct the image? In fact, with a slight modification, the previous algorithms still work.

Instead of assuming a known calibration image, you can detect strong elongated edges and fit ideal step edges in such regions (Figure 10.9b), resulting in the sharp image shown in Figure 10.9d. For every pixel that is surrounded by a complete set of valid estimated neighbors (green pixels in Figure 10.9c), apply the least squares formula (10.1) to estimate the kernel K. The resulting locally estimated PSFs can be used to correct for chromatic aberration (since the relative displacements between per-channel PSFs can be computed), as shown by Joshi, Szeliski, and Kriegman (2008).

Exercise 10.4 provides some more detailed instructions for implementing and testing edge-based PSF estimation algorithms. An alternative approach, which does not require the explicit detection of edges but uses image statistics (gradient distributions) instead, is presented by Fergus, Singh, Hertzmann *et al.* (2006).

10.2 High dynamic range imaging

As we mentioned earlier in this chapter, registered images taken at different exposures can be used to calibrate the radiometric response function of a camera. More importantly, they can

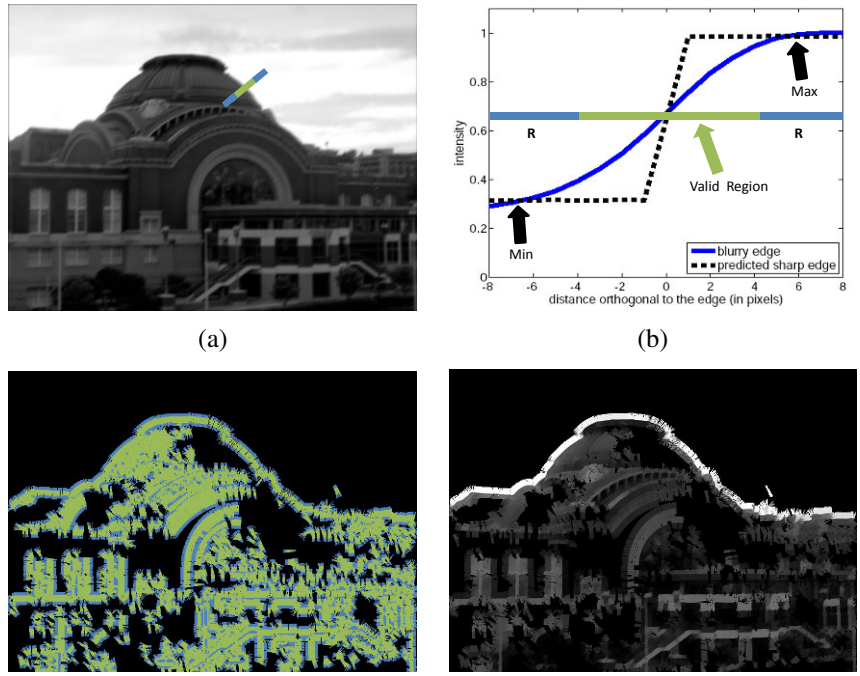


Figure 10.9 Estimating the PSF without using a calibration pattern (Joshi, Szeliski, and Kriegman 2008) © 2008 IEEE: (a) Input image with blue cross-section (profile) location, (b) Profile of sensed and predicted step edges, (c–d) Locations and values of the predicted colors near the edge locations.

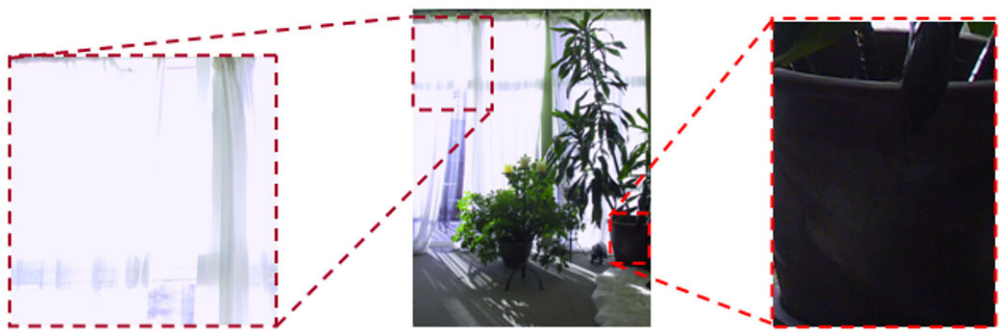


Figure 10.10 Sample indoor image where the areas outside the window are overexposed and inside the room are too dark.



1 1,500 25,000 400,000 2,000,000

Figure 10.11 Relative brightness of different scenes, ranging from 1 inside a dark room lit by a monitor to 2,000,000 looking at the sun. Photos courtesy of Paul Debevec.



Figure 10.12 A bracketed set of shots (using the camera's automatic exposure bracketing (AEB) mode) and the resulting high dynamic range (HDR) composite.

help you create well-exposed photographs under challenging conditions, such as brightly lit scenes where any single exposure contains saturated (overexposed) and dark (underexposed) regions (Figure 10.10). This problem is quite common, because the natural world contains a range of radiance values that is far greater than can be captured with any photographic sensor or film (Figure 10.11). Taking a set of *bracketed exposures* (exposures taken by a camera in automatic exposure bracketing (AEB) mode to deliberately under- and over-expose the image) gives you the material from which to create a properly exposed photograph, as shown in Figure 10.12 (Reinhard, Ward, Pattanaik *et al.* 2005; Freeman 2008; Gulbins and Gulbins 2009; Hasinoff, Durand, and Freeman 2010).

While it is possible to combine pixels from different exposures directly into a final composite (Burt and Kolczynski 1993; Mertens, Kautz, and Reeth 2007), this approach runs the risk of creating contrast reversals and halos. Instead, the more common approach is to proceed in three stages:

1. Estimate the radiometric response function from the aligned images.
2. Estimate a *radiance map* by selecting or blending pixels from different exposures.
3. Tone map the resulting high dynamic range (HDR) image back into a displayable gamut.

The idea behind estimating the radiometric response function is relatively straightforward (Mann and Picard 1995; Debevec and Malik 1997; Mitsunaga and Nayar 1999; Reinhard, Ward, Pattanaik *et al.* 2005). Suppose you take three sets of images at different exposures (shutter speeds), say at ± 2 exposure values.[11] If we were able to determine the irradiance

[11] Changing the shutter speed is preferable to changing the aperture, as the latter can modify the vignetting and focus. Using ± 2 "f-stops" (technically, exposure values, or EVs, since f-stops refer to apertures) is usually the right compromise between capturing a good dynamic range and having properly exposed pixels everywhere.

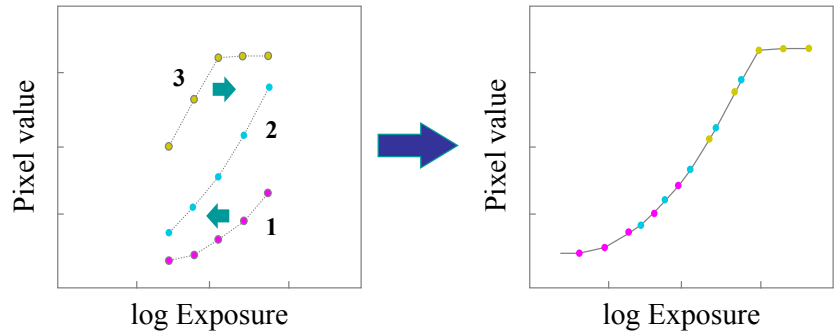


Figure 10.13 Radiometric calibration using multiple exposures (Debevec and Malik 1997). Corresponding pixel values are plotted as functions of log exposures (irradiance). The curves on the left are shifted to account for each pixel's unknown radiance until they all line up into a single smooth curve.

(exposure) E_i at each pixel (2.101), we could plot it against the measured pixel value z_{ij} for each exposure time t_j, as shown in Figure 10.13.

Unfortunately, we do not know the irradiance values E_i, so these have to be estimated at the same time as the radiometric response function f, which can be written (Debevec and Malik 1997) as

$$z_{ij} = f(E_i\, t_j), \tag{10.3}$$

where t_j is the exposure time for the jth image. The inverse response curve f^{-1} is given by

$$f^{-1}(z_{ij}) = E_i\, t_j. \tag{10.4}$$

Taking logarithms of both sides (base 2 is convenient, as we can now measure quantities in EVs), we obtain

$$g(z_{ij}) = \log f^{-1}(z_{ij}) = \log E_i + \log t_j, \tag{10.5}$$

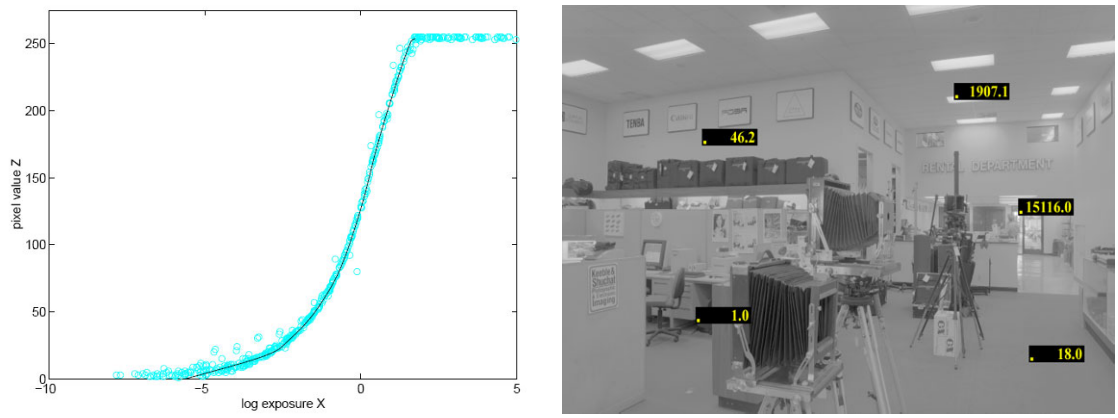
where $g = \log f^{-1}$ (which maps pixel values z_{ij} into log irradiance) is the curve we are estimating (Figure 10.13 turned on its side).

Debevec and Malik (1997) assume that the exposure times t_j are known. (Recall that these can be obtained from a camera's EXIF tags, but that they actually follow a power of 2 progression $\ldots$, $1/128$, $1/64$, $1/32$, $1/16$, $1/8$, $\ldots$ instead of the marked $\ldots$, $1/125$, $1/60$, $1/30$, $1/15$, $1/8$, $\ldots$ values—see Exercise 2.5.) The unknowns are therefore the per-pixel exposures E_i and the response values $g_k = g(k)$, where g can be discretized according to the 256 pixel values commonly observed in eight-bit images. (The response curves are calibrated separately for each color channel.)

In order to make the response curve smooth, Debevec and Malik (1997) add a second-order smoothness constraint

$$\lambda \sum_k g''(k)^2 = \lambda \sum [g(k-1) - 2g(k) + g(k+1)]^2, \tag{10.6}$$

which is similar to the one used in snakes (5.3). Since pixel values are more reliable in the middle of their range (and the g function becomes singular near saturation values), they also



(a) (b)

Figure 10.14 Recovered response function and radiance image for a real digital camera (DCS460) (Debevec and Malik 1997) © 1997 ACM.

add a weighting (hat) function $w(k)$ that decays to zero at both ends of the pixel value range,

$$w(z) = \begin{cases} z - z_{\min} & z \leq (z_{\min} + z_{\max})/2 \\ z_{\max} - z & z > (z_{\min} + z_{\max})/2. \end{cases} \tag{10.7}$$

Putting all of these terms together, they obtain a least squares problem in the unknowns $\{g_k\}$ and $\{E_i\}$,

$$E = \sum_i \sum_j w(z_{i,j})[g(z_{i,j}) - \log E_i - \log t_j]^2 + \lambda \sum_k w(k) g''(k)^2. \tag{10.8}$$

(In order to remove the overall shift ambiguity in the response curve and irradiance values, the middle of the response curve is set to 0.) Debevec and Malik (1997) show how this can be implemented in 21 lines of MATLAB code, which partially accounts for the popularity of their technique.

While Debevec and Malik (1997) assume that the exposure times t_j are known exactly, there is no reason why these additional variables cannot be thrown into the least squares problem, constraining their final estimated values to lie close to their nominal values $\hat{t}_j$ with an extra term $\eta \sum_j (t_j - \hat{t}_j)^2$.

Figure 10.14 shows the recovered radiometric response function for a digital camera along with select (relative) radiance values in the overall radiance map. Figure 10.15 shows the bracketed input images captured on color film and the corresponding radiance map.

While Debevec and Malik (1997) use a general second-order smooth curve g to parameterize their response curve, Mann and Picard (1995) use a three-parameter function

$$f(E) = \alpha + \beta E^{\gamma}, \tag{10.9}$$

while Mitsunaga and Nayar (1999) use a low-order ($N \leq 10$) polynomial for the inverse response function g. Pal, Szeliski, Uyttendaele *et al.* (2004) derive a Bayesian model that estimates an independent smooth response function for each image, which can better model

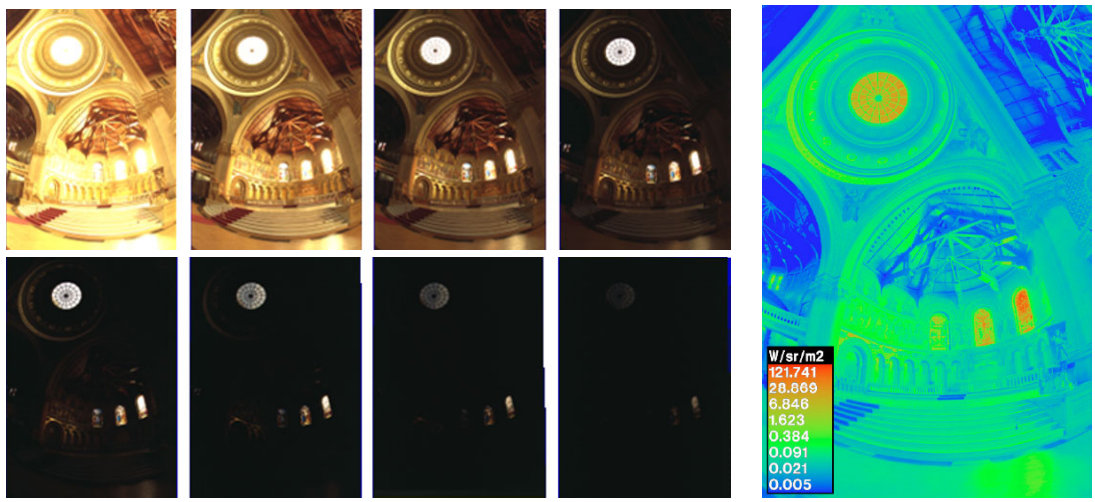


Figure 10.15 Bracketed set of exposures captured with a film camera and the resulting radiance image displayed in pseudocolor (Debevec and Malik 1997) © 1997 ACM.

the more sophisticated (and hence less predictable) automatic contrast and tone adjustment performed in today's digital cameras.

Once the response function has been estimated, the second step in creating high dynamic range photographs is to merge the input images into a composite *radiance map*. If the response function and images were known exactly, i.e., if they were noise free, you could use any non-saturated pixel value to estimate the corresponding radiance by mapping it through the inverse response curve $E = g(z)$.

Unfortunately, pixels are noisy, especially under low-light conditions when fewer photons arrive at the sensor. To compensate for this, Mann and Picard (1995) use the derivative of the response function as a weight in determining the final radiance estimate, since "flatter" regions of the curve tell us less about the incoming irradiance. Debevec and Malik (1997) use a hat function (10.7) which accentuates mid-tone pixels while avoiding saturated values. Mitsunaga and Nayar (1999) show that in order to maximize the signal-to-noise ratio (SNR), the weighting function must emphasize both higher pixel values and larger gradients in the transfer function, i.e.,

$$w(z) = g(z)/g'(z), \tag{10.10}$$

where the weights w are used to form the final irradiance estimate

$$\log E_i = \frac{\sum_j w(z_{ij})[g(z_{ij}) - \log t_j]}{\sum_j w(z_{ij})}. \tag{10.11}$$

Exercise 10.1 has you implement one of the radiometric response function calibration techniques and then use it to create radiance maps.

Under real-world conditions, casually acquired images may not be perfectly registered and may contain moving objects. Ward (2003) uses a global (parametric) transform to align the input images, while Kang, Uyttendaele, Winder *et al.* (2003) present an algorithm that combines global registration with local motion estimation (optical flow) to accurately align

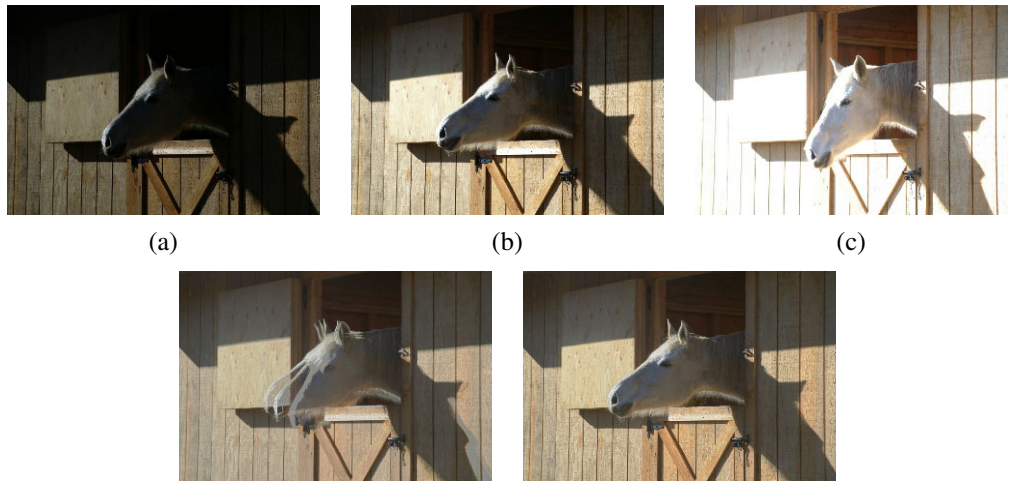


Figure 10.16 Merging multiple exposures to create a high dynamic range composite (Kang, Uyttendaele, Winder *et al.* 2003): (a–c) three different exposures; (d) merging the exposures using classic algorithms (note the ghosting due to the horse's head movement); (e) merging the exposures with motion compensation.

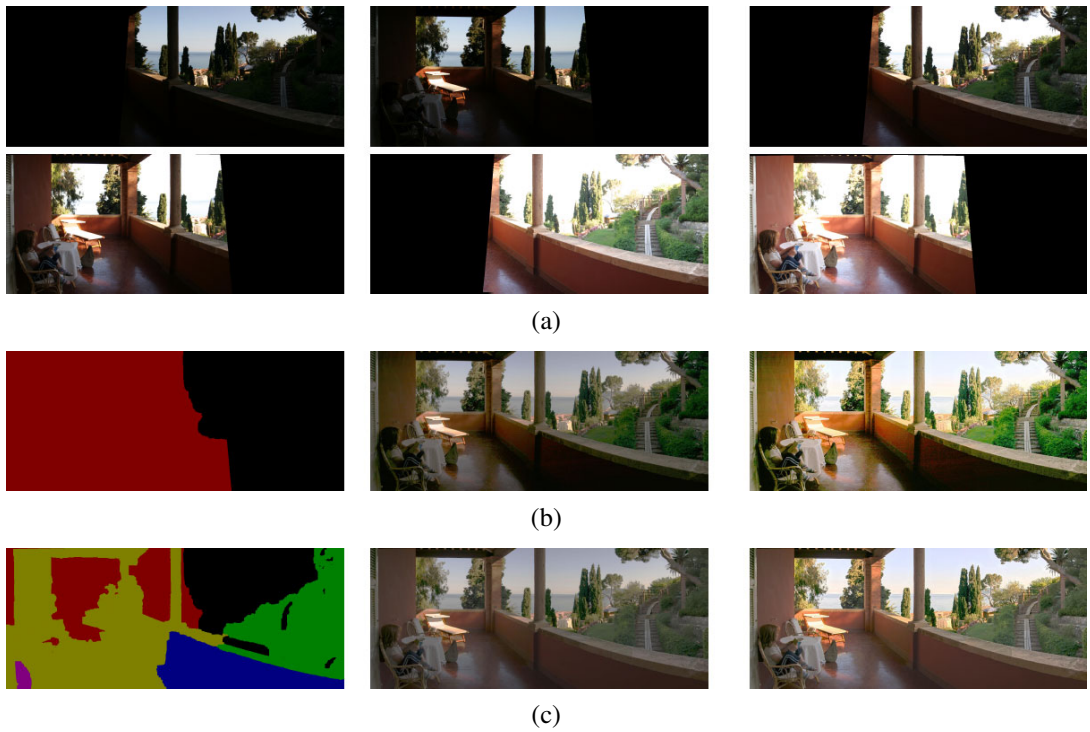


Figure 10.17 HDR merging with large amounts of motion (Eden, Uyttendaele, and Szeliski 2006) © 2006 IEEE: (a) registered bracketed input images; (b) results after the first pass of image selection: reference labels, image, and tone-mapped image; (c) results after the second pass of image selection: final labels, compressed HDR image, and tone-mapped image

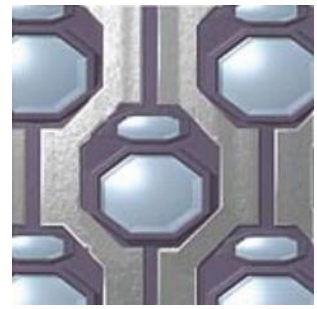


Figure 10.18 Fuji SuperCCD high dynamic range image sensor. The paired large and small active areas provide two different effective exposures.

the images before blending their radiance estimates (Figure 10.16). Since the images may have widely different exposures, care must be taken when estimating the motions, which must themselves be checked for consistency to avoid the creation of ghosts and object fragments.

Even this approach, however, may not work when the camera is simultaneously undergoing large panning motions and exposure changes, which is a common occurrence in casually acquired panoramas. Under such conditions, different parts of the image may be seen at one or more exposures. Devising a method to blend all of these different sources while avoiding sharp transitions and dealing with scene motion is a challenging problem. One approach is to first find a consensus mosaic and to then selectively compute radiances in under- and over-exposed regions (Eden, Uyttendaele, and Szeliski 2006), as shown in Figure 10.17.

Recently, some cameras, such as the Sony α550 and Pentax K-7, have started integrating multiple exposure merging and tone mapping directly into the camera body. In the future, the need to compute high dynamic range images from multiple exposures may be eliminated by advances in camera sensor technology (Figure 10.18) (Yang, El Gamal, Fowler *et al.* 1999; Nayar and Mitsunaga 2000; Nayar and Branzoi 2003; Kang, Uyttendaele, Winder *et al.* 2003; Narasimhan and Nayar 2005; Tumblin, Agrawal, and Raskar 2005). However, the need to blend such images and to tone map them to lower-gamut displays is likely to remain.

HDR image formats. Before we discuss techniques for mapping HDR images back to a displayable gamut, we should discuss the commonly used formats for storing HDR images.

If storage space is not an issue, storing each of the R, G, and B values as a 32-bit IEEE float is the best solution. The commonly used Portable PixMap (.ppm) format, which supports both uncompressed ASCII and raw binary encodings of values, can be extended to a Portable FloatMap (.pfm) format by modifying the header. TIFF also supports full floating point values.

A more compact representation is the Radiance format (.pic, .hdr) (Ward 1994), which uses a single common exponent and per-channel mantissas (10.19b). An intermediate encoding, OpenEXR from ILM,[12] uses 16-bit floats for each channel (10.19c), which is a format supported natively on most modern GPUs. Ward (2004) describes these and other data formats such as LogLuv (Larson 1998) in more detail, as do the books by Reinhard, Ward,

[12] http://www.openexr.net/.

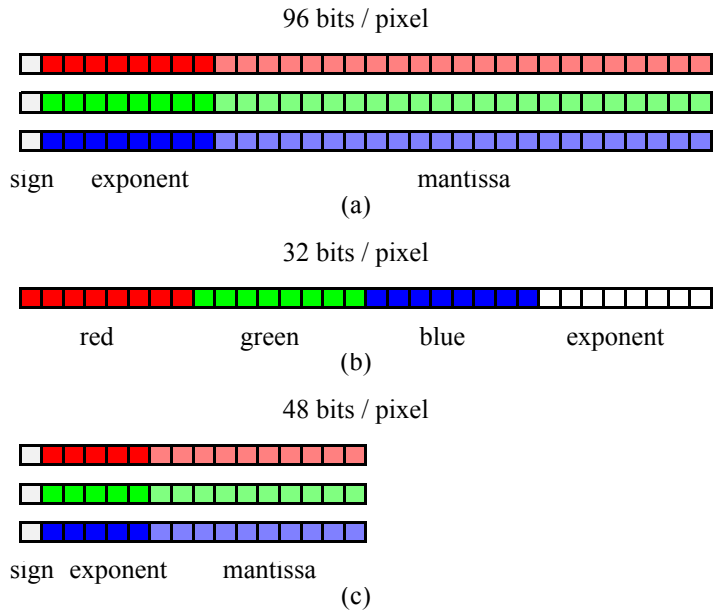


Figure 10.19 HDR image encoding formats: (a) Portable PixMap (.ppm); (b) Radiance (.pic, .hdr); (c) OpenEXR (.exr).

Pattanaik *et al.* (2005) and Freeman (2008). An even more recent HDR image format is the JPEG XR standard.[13]

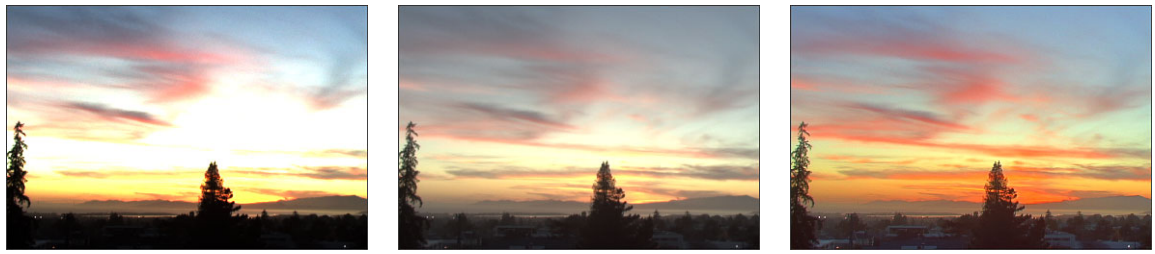
10.2.1 Tone mapping

Once a radiance map has been computed, it is usually necessary to display it on a lower gamut (i.e., eight-bit) screen or printer. A variety of *tone mapping* techniques has been developed for this purpose, which involve either computing spatially varying transfer functions or reducing image gradients to fit the available dynamic range (Reinhard, Ward, Pattanaik *et al.* 2005).

The simplest way to compress a high dynamic range radiance image into a low dynamic range gamut is to use a global transfer curve (Larson, Rushmeier, and Piatko 1997). Figure 10.20 shows one such example, where a gamma curve is used to map an HDR image back into a displayable gamut. If gamma is applied separately to each channel (Figure 10.20b), the colors become muted (less saturated), since higher-valued color channels contribute less (proportionately) to the final color. Splitting the image up into its luminance and chrominance (say, L*a*b*) components (Section 2.3.2), applying the global mapping to the luminance channel, and then reconstituting a color image works better (Figure 10.20c).

Unfortunately, when the image has a really wide range of exposures, this global approach still fails to preserve details in regions with widely varying exposures. What is needed, instead, is something akin to the dodging and burning performed by photographers in the darkroom. Mathematically, this is similar to dividing each pixel by the *average* brightness in a region around that pixel.

[13] http://www.itu.int/rec/T-REC-T.832-200903-I/en.



(a) (b) (c)

Figure 10.20 Global tone mapping: (a) input HDR image, linearly mapped; (b) gamma applied to each color channel independently; (c) gamma applied to intensity (colors are less washed out). Original HDR image courtesy of Paul Debevec, http://ict.debevec.org/~debevec/Research/HDR/. Processed images courtesy of Frédo Durand, MIT 6.815/6.865 course on Computational Photography.

Figure 10.21 shows how this process works. As before, the image is split into its luminance and chrominance channels. The log luminance image

$$H(x, y) = \log L(x, y) \tag{10.12}$$

is then low-pass filtered to produce a *base layer*

$$H_{\mathrm{L}}(x, y) = B(x, y) * H(x, y), \tag{10.13}$$

and a high-pass *detail layer*

$$H_{\mathrm{H}}(x, y) = H(x, y) - H_{\mathrm{L}}(x, y). \tag{10.14}$$

The base layer is then contrast reduced by scaling to the desired log-luminance range,

$$H'_{\mathrm{H}}(x, y) = s\, H_{\mathrm{H}}(x, y) \tag{10.15}$$

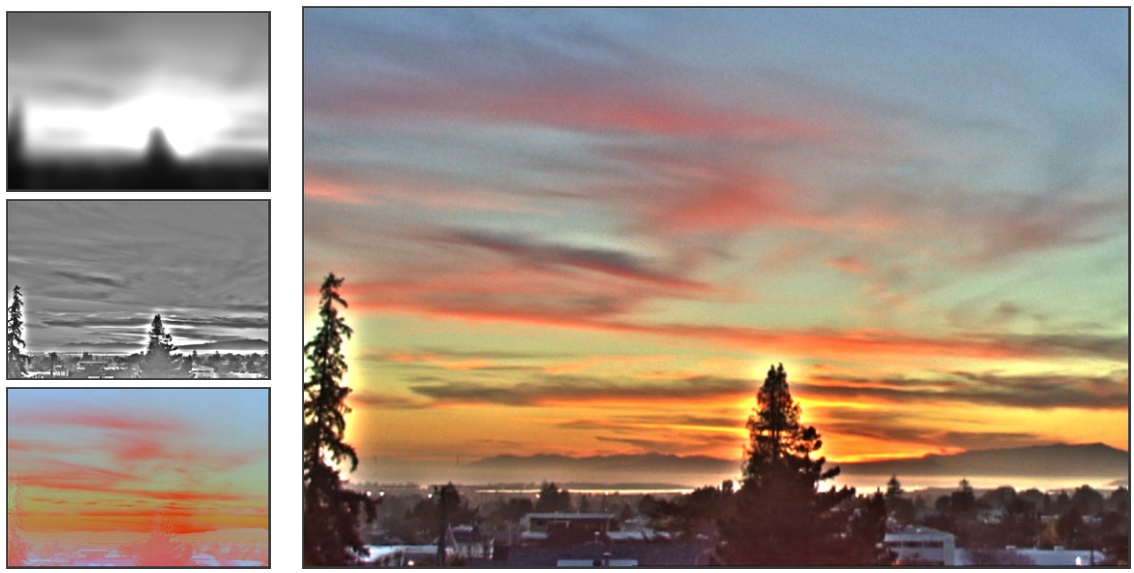
and added to the detail layer to produce the new log-luminance image

$$I(x, y) = H'_{\mathrm{H}}(x, y) + H_{\mathrm{L}}(x, y), \tag{10.16}$$

which can then be exponentiated to produce the tone-mapped (compressed) luminance image. Note that this process is equivalent to dividing each luminance value by (a monotonic mapping of) the average log-luminance value in a region around that pixel.

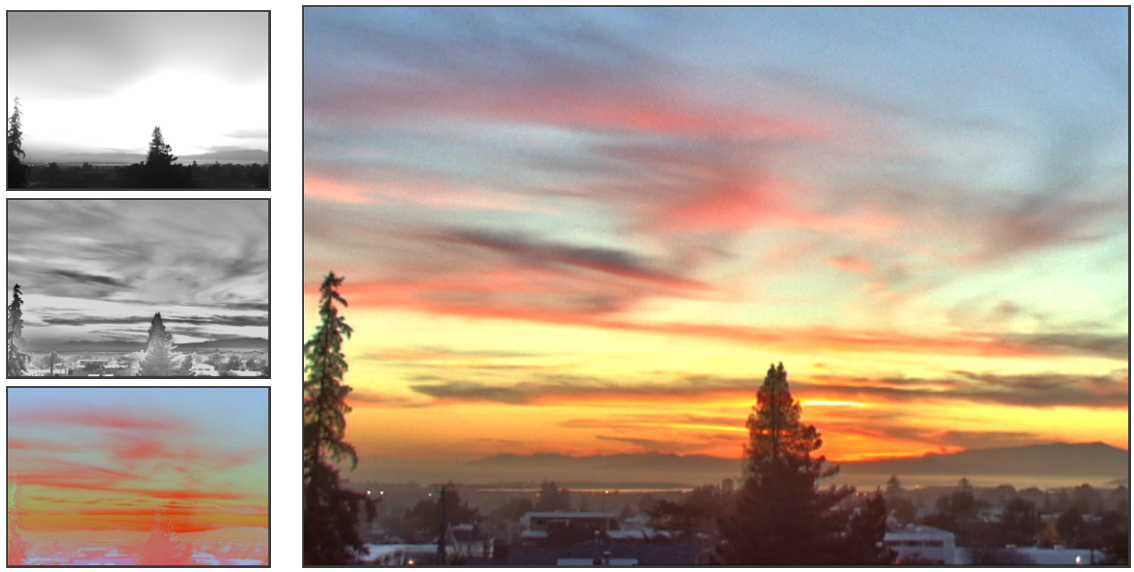
Figure 10.21 shows the low-pass and high-pass log luminance image and the resulting tone-mapped color image. Note how the detail layer has visible *halos* around the high-contrast edges, which are visible in the final tone-mapped image. This is because linear filtering, which is not edge preserving, produces halos in the detail layer (Figure 10.23).

The solution to this problem is to use an edge-preserving filter to create the base layer. Durand and Dorsey (2002) study a number of such edge-preserving filters, including anisotropic and robust anisotropic diffusion, and select bilateral filtering (Section 3.3.1) as their edge-preserving filter. (A more recent paper by Farbman, Fattal, Lischinski *et al.* (2008) argues in favor of using a weighted least squares (WLF) filter as an alternative to the bilateral filter and Paris, Kornprobst, Tumblin *et al.* (2008) reviews bilateral filtering and its applications



(a) (b)

Figure 10.21 Local tone mapping using linear filters: (a) low-pass and high-pass filtered log luminance images and color (chrominance) image; (b) resulting tone-mapped image (after attenuating the low-pass log luminance image) shows visible halos around the trees. Processed images courtesy of Frédo Durand, MIT 6.815/6.865 course on Computational Photography.



(a) (b)

Figure 10.22 Local tone mapping using bilateral filter (Durand and Dorsey 2002): (a) low-pass and high-pass bilateral filtered log luminance images and color (chrominance) image; (b) resulting tone-mapped image (after attenuating the low-pass log luminance image) shows no halos. Processed images courtesy of Frédo Durand, MIT 6.815/6.865 course on Computational Photography.

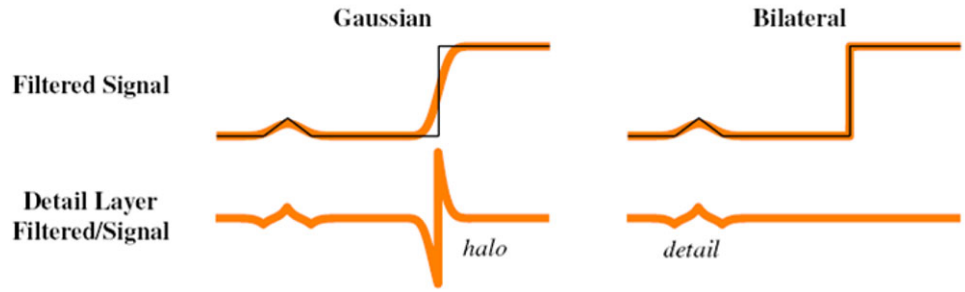


Figure 10.23 Gaussian vs. bilateral filtering (Petschnigg, Agrawala, Hoppe *et al.* 2004) © 2004 ACM: A Gaussian low-pass filter blurs across all edges and therefore creates strong peaks and valleys in the detail image that cause halos. The bilateral filter does not smooth across strong edges and thereby reduces halos while still capturing detail.

in computer vision and computational photography.) Figure 10.22 shows how replacing the linear low-pass filter with a bilateral filter produces tone-mapped images with no visible halos. Figure 10.24 summarizes the complete information flow in this process, starting with the decomposition into log luminance and chrominance images, bilateral filtering, contrast reduction, and re-composition into the final output image.

An alternative to compressing the base layer is to compress its *derivatives*, i.e., the gradient of the log-luminance image (Fattal, Lischinski, and Werman 2002). Figure 10.25 illustrates this process. The log-luminance image is differentiated to obtain a gradient image

$$H'(x,y) = \nabla H(x,y). \tag{10.17}$$

This gradient image is then attenuated by a spatially varying attenuation function $\Phi(x,y)$,

$$G(x,y) = H'(x,y)\,\Phi(x,y). \tag{10.18}$$

The attenuation function $I(x,y)$ is designed to attenuate large-scale brightness changes (Figure 10.26a) and is designed to take into account gradients at different spatial scales (Fattal, Lischinski, and Werman 2002).

After attenuation, the resulting gradient field is re-integrated by solving a first-order variational (least squares) problem,

$$\min \int\int \|\nabla I(x,y) - G(x,y)\|^2 dx\,dy \tag{10.19}$$

to obtain the compressed log-luminance image $I(x,y)$. This least squares problem is the same that was used for Poisson blending (Section 9.3.4) and was first introduced in our study of regularization (Section 3.7.1, 3.100). It can efficiently be solved using techniques such as multigrid and hierarchical basis preconditioning (Fattal, Lischinski, and Werman 2002; Szeliski 2006b; Farbman, Fattal, Lischinski *et al.* 2008). Once the new luminance image has been computed, it is combined with the original color image using

$$C_{\text{out}} = \left(\frac{C_{\text{in}}}{L_{\text{in}}}\right)^s L_{\text{out}}, \tag{10.20}$$

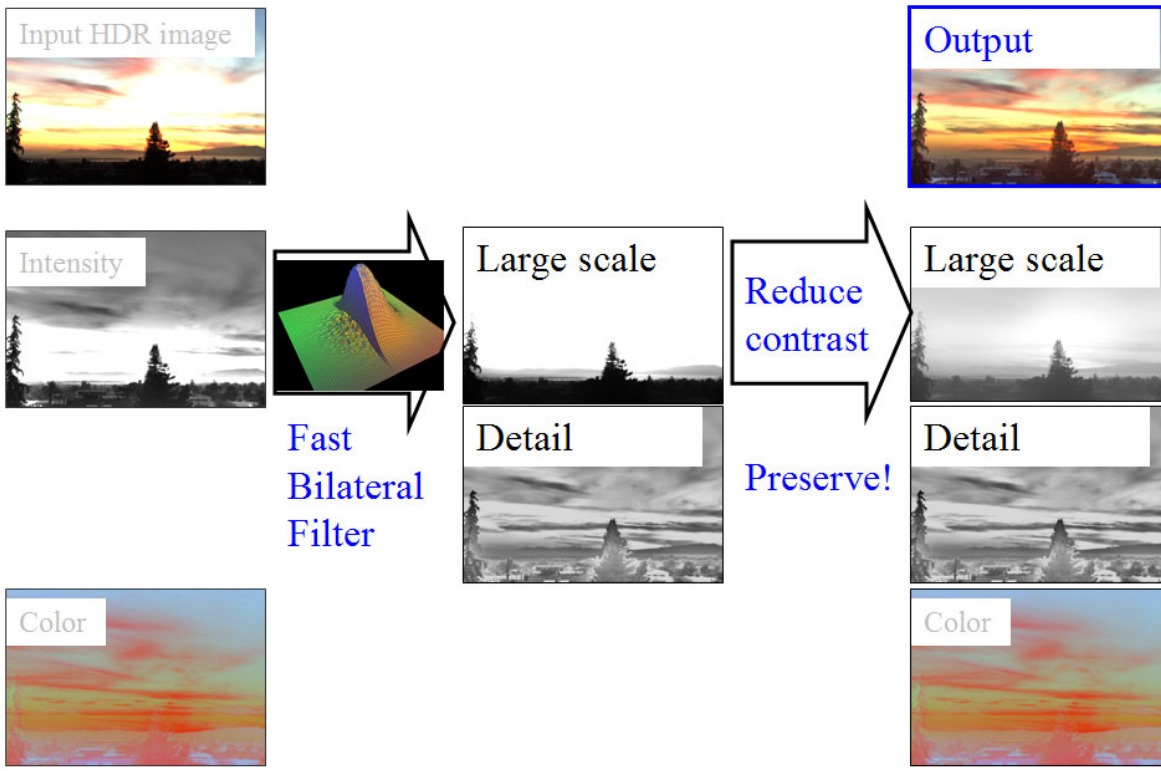


Figure 10.24 Local tone mapping using bilateral filter (Durand and Dorsey 2002): summary of algorithm workflow. Images courtesy of Frédo Durand, MIT 6.815/6.865 course on Computational Photography.

where $C = (R, G, B)$ and L_{in} and L_{out} are the original and compressed luminance images. The exponent s controls the saturation of the colors and is typically in the range $s \in [0.4, 0.6]$. Figure 10.26b shows the final tone-mapped color image, which shows no visible halos despite the extremely large variation in input radiance values.

Yet another alternative to these two approaches is to perform the local dodging and burning using a locally scale-selective operator (Reinhard, Stark, Shirley *et al.* 2002). Figure 10.27 shows how such a scale selection operator can determine a radius (scale) that only includes similar color values within the inner circle while avoiding much brighter values in the surrounding circle. In practice, a difference of Gaussians normalized by the inner Gaussian response is evaluated over a range of scales, and the largest scale whose metric is below a threshold is selected (Reinhard, Stark, Shirley *et al.* 2002).

What all of these techniques have in common is that they adaptively attenuate or brighten different regions of the image so that they can be displayed in a limited gamut without loss of contrast. Lischinski, Farbman, Uyttendaele *et al.* (2006b) introduce an *interactive* technique that performs this operation by interpolating a set of sparse user-drawn adjustments (strokes and associated exposure value corrections) to a piecewise-continuous exposure correction map (Figure 10.28). The interpolation is performed by minimizing a locally weighted least

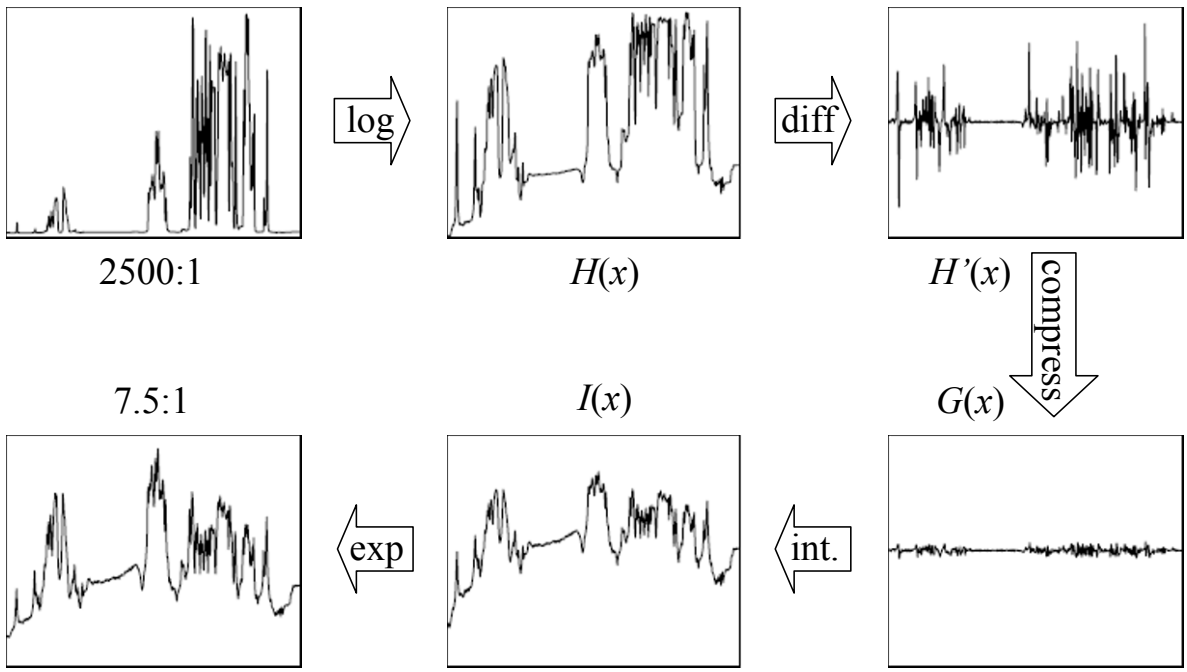


Figure 10.25 Gradient domain tone mapping (Fattal, Lischinski, and Werman 2002) © 2002 ACM. The original image with a dynamic range of 2415:1 is first converted into the log domain, $H(x)$, and its gradients are computed, $H'(x)$. These are attenuated (compressed) based on local contrast, $G(x)$, and integrated to produce the new logarithmic exposure image $I(x)$, which is exponentiated to produce the final intensity image, whose dynamic range is 7.5:1.



(a) (b)

Figure 10.26 Gradient domain tone mapping (Fattal, Lischinski, and Werman 2002) © 2002 ACM: (a) attenuation map, with darker values corresponding to more attenuation; (b) final tone-mapped image.

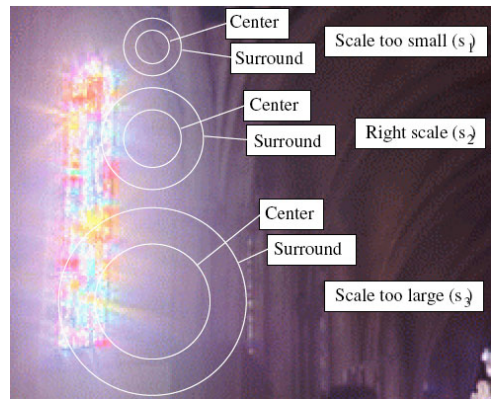


Figure 10.27 Scale selection for tone mapping (Reinhard, Stark, Shirley *et al.* 2002) © 2002 ACM.

square (WLS) variational problem,

$$\min \int\int w_{\mathrm{d}}(x,y)\|f(x,y)-g(x,y)\|^2 dx\,dy + \lambda \int\int w_{\mathrm{s}}(x,y)\|\nabla f(x,y)\|^2 dx\,dy, \tag{10.21}$$

where $g(x,y)$ and $f(x,y)$ are the input and output log exposure (attenuation) maps (Figure 10.28). The data weighting term $w_{\mathrm{d}}(x,y)$ is 1 at stroke locations and 0 elsewhere. The smoothness weighting term $w_{\mathrm{s}}(x,y)$ is inversely proportional to the log-luminance gradient,

$$w_{\mathrm{s}} = \frac{1}{\|\nabla H\|^{\alpha}+\epsilon} \tag{10.22}$$

and hence encourages the $f(x,y)$ map to be smoother in low-gradient areas than along high-gradient discontinuities.[14] The same approach can also be used for fully automated tone mapping by setting target exposure values at each pixel and allowing the weighted least squares to convert these into piecewise smooth adjustment maps.

The weighted least squares algorithm, which was originally developed for image colorization applications (Levin, Lischinski, and Weiss 2004), has recently been applied to general edge-preserving smoothing in applications such as contrast enhancement (Bae, Paris, and Durand 2006) and tone mapping (Farbman, Fattal, Lischinski *et al.* 2008) where the bilateral filtering was previously used. It can also be used to perform HDR merging and tone mapping simultaneously (Raman and Chaudhuri 2007, 2009).

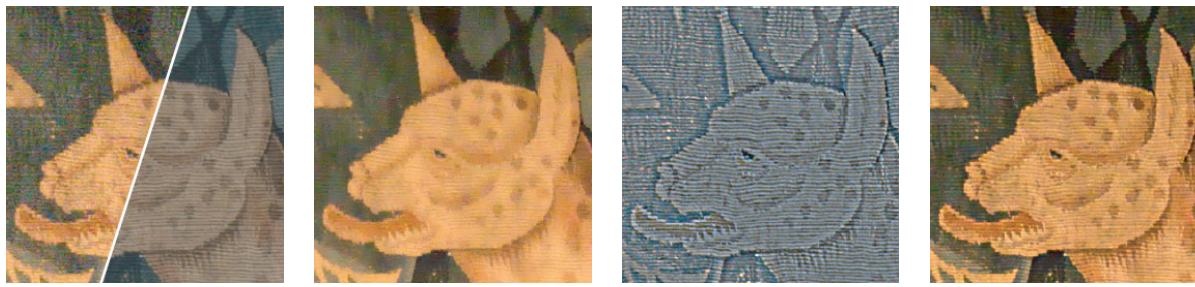
Given the wide range of locally adaptive tone mapping algorithms that have been developed, which ones should be used in practice? Freeman (2008) provides a great discussion of commercially available algorithms, their artifacts, and the parameters that can be used to control them. He also has a wealth of tips for HDR photography and workflow. I highly recommend his book for anyone contemplating additional research (or personal photography) in this area.

[14] In practice, the x and y discrete derivatives are weighted separately (Lischinski, Farbman, Uyttendaele *et al.* 2006b). Their default parameter settings are $\lambda = 0.2$, $\alpha = 1$, and $\epsilon = 0.0001$.



(a) (b)

Figure 10.28 Interactive local tone mapping (Lischinski, Farbman, Uyttendaele *et al.* 2006b) © 2006 ACM: (a) user-drawn strokes with associated exposure values $g(x, y)$ (b) corresponding piecewise-smooth exposure adjustment map $f(x, y)$.



(a) (b) (c) (d)

Figure 10.29 Detail transfer in flash/no-flash photography (Petschnigg, Agrawala, Hoppe *et al.* 2004) © 2004 ACM: (a) details of input ambient A and flash F images; (b) joint bilaterally filtered no-flash image A^{NR}; (c) detail layer F^{Detail} computed from the flash image F; (d) final merged image A^{Final}.

10.2.2 *Application*: Flash photography

While high dynamic range imaging combines images of a scene taken at different exposures, it is also possible to combine flash and non-flash images to achieve better exposure and color balance and to reduce noise (Eisemann and Durand 2004; Petschnigg, Agrawala, Hoppe *et al.* 2004).

The problem with flash images is that the color is often unnatural (it fails to capture the ambient illumination), there may be strong shadows or specularities, and there is a radial falloff in brightness away from the camera (Figures 10.1b and 10.29a). Non-flash photos taken under low light conditions often suffer from excessive noise (because of the high ISO gains and low photon counts) and blur (due to longer exposures). Is there some way to combine a non-flash photo taken just before the flash goes off with the flash photo to produce

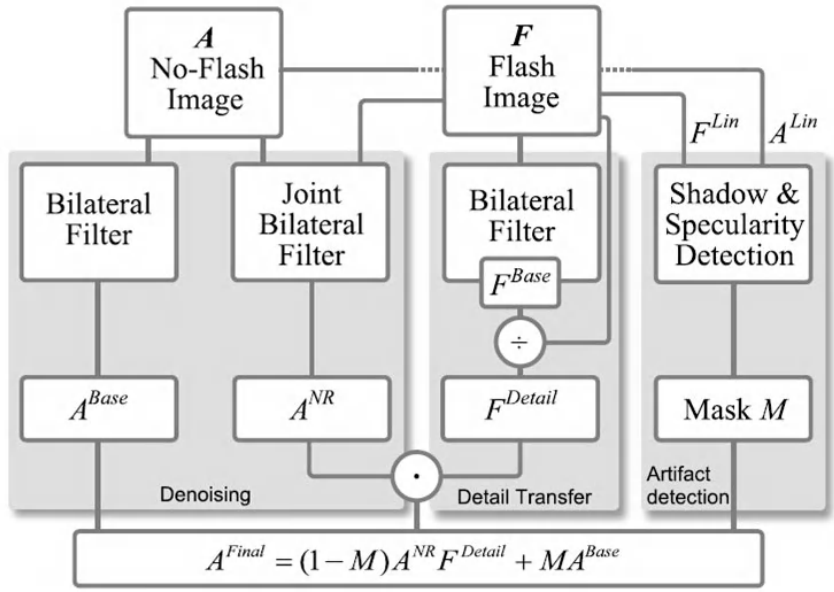


Figure 10.30 Flash/no-flash photography algorithm (Petschnigg, Agrawala, Hoppe *et al.* 2004) © 2004 ACM. The ambient (no-flash) image A is filtered with a regular bilateral filter to produce A^{Base}, which is used in shadow and specularity regions, and a joint bilaterally filtered noise reduced image A^{NR}. The flash image F is bilaterally filtered to produce a base image F^{Base} and a detail (ratio) image F^{Detail}, which is used to modulate the de-noised ambient image. The shadow/specularity mask M is computed by comparing linearized versions of the flash and no-flash images.

an image with good color values, sharpness, and low noise?[15]

Petschnigg, Agrawala, Hoppe *et al.* (2004) approach this problem by first filtering the no-flash (ambient) image A with a variant of the bilateral filter called the *joint bilateral filter*[16] in which the range kernel (3.36)

$$r(i,j,k,l) = \exp\left(-\frac{\|f(i,j) - f(k,l)\|^2}{2\sigma_r^2}\right) \tag{10.23}$$

is evaluated on the flash image F instead of the ambient image A, since the flash image is less noisy and hence has more reliable edges (Figure 10.29b). Because the contents of the flash image can be unreliable inside and at the boundaries of shadows and specularities, these are detected and a regular bilaterally filtered image A^{Base} is used instead (Figure 10.30).

The second stage of their algorithm computes a flash detail image

$$F^{Detail} = \frac{F + \epsilon}{F^{\text{Base}} + \epsilon}, \tag{10.24}$$

where F^{Base} is a bilaterally filtered version of the flash image F and $\epsilon = 0.02$. This detail image (Figure 10.29c) encodes details that may have been filtered away from the noise-reduced

[15] In fact, the discontinued FujiFilm FinePix F40fd camera takes a pair of flash and no flash images in quick succession; however, it only lets you decide to keep one of them.

[16] Eisemann and Durand (2004) call this the *cross bilateral filter*.

no-flash image A^{NR}, as well as additional details created by the flash camera, which often add crispness. The detail image is used to modulate the noise-reduced ambient image A^{NR} to produce the final results

$$A^{Final} = (1 - M)A^{NR}F^{Detail} + MA^{Base} \tag{10.25}$$

shown in Figures 10.1b and 10.29d.

Eisemann and Durand (2004) present an alternative algorithm that shares some of the same basic concepts. Both papers are well worth reading and contrasting (Exercise 10.6).

Flash images can also be used for a variety of additional applications such as extracting more reliable foreground mattes of objects (Raskar, Tan, Feris *et al.* 2004; Sun, Li, Kang *et al.* 2006). Flash photography is just one instance of the more general topic of *active illumination*, which is discussed in more detail by Raskar and Tumblin (2010).

10.3 Super-resolution and blur removal

While high dynamic range imaging enables us to obtain an image with a larger dynamic range than a single regular image, super-resolution enables us to create images with higher *spatial* resolution and less noise than regular camera images (Chaudhuri 2001; Park, Park, and Kang 2003; Capel and Zisserman 2003; Capel 2004; van Ouwerkerk 2006). Most commonly, super-resolution refers to the process of aligning and combining several input images to produce such high-resolution composites (Irani and Peleg 1991; Cheeseman, Kanefsky, Hanson *et al.* 1993; Pickup, Capel, Roberts *et al.* 2009). However, some newer techniques can super-resolve a single image (Freeman, Jones, and Pasztor 2002; Baker and Kanade 2002; Fattal 2007) and are hence closely related to techniques for removing blur (Sections 3.4.3 and 3.4.4).

The most principled way to formulate the super-resolution problem is to write down the stochastic image formation equations and image priors and to then use Bayesian inference to recover the super-resolved (original) sharp image. We can do this by generalizing the image formation equations (3.75) used for image deblurring (Section 3.4.3), which we also used in (10.2) for blur kernel (PSF) estimation (Section 10.1.4). In this case, we have several observed images $\{o_k(\boldsymbol{x})\}$, as well as an image warping function $\hat{\boldsymbol{h}}_k(\boldsymbol{x})$ for each observed image (Figure 3.47). Combining all of these elements, we get the (noisy) observation equations[17]

$$o_k(\boldsymbol{x}) = D\{b(\boldsymbol{x}) * s(\hat{\boldsymbol{h}}_k(\boldsymbol{x}))\} + n_k(\boldsymbol{x}), \tag{10.26}$$

where D is the downsampling operator, which operates *after* the super-resolved (sharp) warped image $s(\hat{\boldsymbol{h}}_k(\boldsymbol{x}))$ has been convolved with the blur kernel $b(\boldsymbol{x})$. The above image formation equations lead to the following least squares problem,

$$\sum_k \|o_k(\boldsymbol{x}) - D\{b_k(\boldsymbol{x}) * s(\hat{\boldsymbol{h}}_k(\boldsymbol{x}))\}\|^2. \tag{10.27}$$

In most super-resolution algorithms, the alignment (warping) $\hat{\boldsymbol{h}}_k$ is estimated using one of the input frames as the *reference frame*; either feature-based (Section 6.1.3) or direct (image-based) (Section 8.2) parametric alignment techniques can be used. (A few algorithms, such

[17] It is also possible to add an unknown bias–gain term to each observation (Capel 2004), as was done for motion estimation in (8.8).

as those described by Schultz and Stevenson (1996) or Capel (2004) use dense (per-pixel flow) estimates.) A better approach is to re-compute the alignment by directly minimizing (10.27) once an initial estimate of $s(\boldsymbol{x})$ has been computed (Hardie, Barnard, and Armstrong 1997) or to *marginalize* out the motion parameters altogether (Pickup, Capel, Roberts *et al.* 2007)—see also the work of Protter and Elad (2009) for some related video super-resolution work.

The point spread function (blur kernel) b_k is either inferred from knowledge of the image formation process (e.g., the amount of motion or defocus blur and the camera sensor optics) or calibrated from a test image or the observed images $\{o_k\}$ using one of the techniques described in Section 10.1.4. The problem of simultaneously inferring the blur kernel and the sharp image is known as *blind image deconvolution* (Kundur and Hatzinakos 1996; Levin 2006).[18]

Given an estimate of $\hat{\boldsymbol{h}}_k$ and $b_k(\boldsymbol{x})$, (10.27) can be re-written using matrix/vector notation as a large sparse least squares problem in the unknown values of the super-resolved pixels $\boldsymbol{s}$,

$$\sum_k \|\boldsymbol{o}_k - \boldsymbol{D}\boldsymbol{B}_k\boldsymbol{W}_k\boldsymbol{s}\|^2. \tag{10.28}$$

(Recall from (3.89) that once the warping function $\hat{\boldsymbol{h}}_k$ is known, values of $s(\hat{\boldsymbol{h}}_k(\boldsymbol{x}))$ depend linearly on those in $s(\boldsymbol{x})$.) An efficient way to solve this least squares problem is to use preconditioned conjugate gradient descent (Capel 2004), although some earlier algorithms, such as the one developed by Irani and Peleg (1991), used regular gradient descent (also known as iterative back projection (IBP), in the computed tomography literature).

The above formulation assumes that warping can be expressed as a simple (sinc or bicubic) interpolated resampling of the super-resolved sharp image, followed by a stationary (spatially invariant) blurring (PSF) and area integration process. However, if the surface is severely foreshortened, we have to take into account the spatially varying filtering that occurs during the image warping (Section 3.6.1), before we can then model the PSF induced by the optics and camera sensor (Wang, Kang, Szeliski *et al.* 2001; Capel 2004).

How well does this least squares (MLE) approach to super-resolution work? In practice, this depends a lot on the amount of blur and aliasing in the camera optics, as well as the accuracy in the motion and PSF estimates (Baker and Kanade 2002; Jiang, Wong, and Bao 2003; Capel 2004). Less blurring and more aliasing means that there is more (aliased) high frequency information available to be recovered. However, because the least squares (maximum likelihood) formulation uses no image prior, a lot of high-frequency noise can be introduced into the solution (Figure 10.31c).

For this reason, most super-resolution algorithms assume some form of image prior. The simplest of these is to place a penalty on the image derivatives similar to Equations (3.105 and 3.113), e.g.,

$$\sum_{(i,j)} \rho_p(s(i,j) - s(i+1,j)) + \rho_p(s(i,j) - s(i,j+1)). \tag{10.29}$$

[18] Notice that there is a chicken-and-egg problem if both the blur kernel and the super-resolved image are unknown. This can be "broken" either using structural assumptions about the sharp image, e.g., the presence of edges (Joshi, Szeliski, and Kriegman 2008) or prior models for the image, such as edge sparsity (Fergus, Singh, Hertzmann *et al.* 2006).

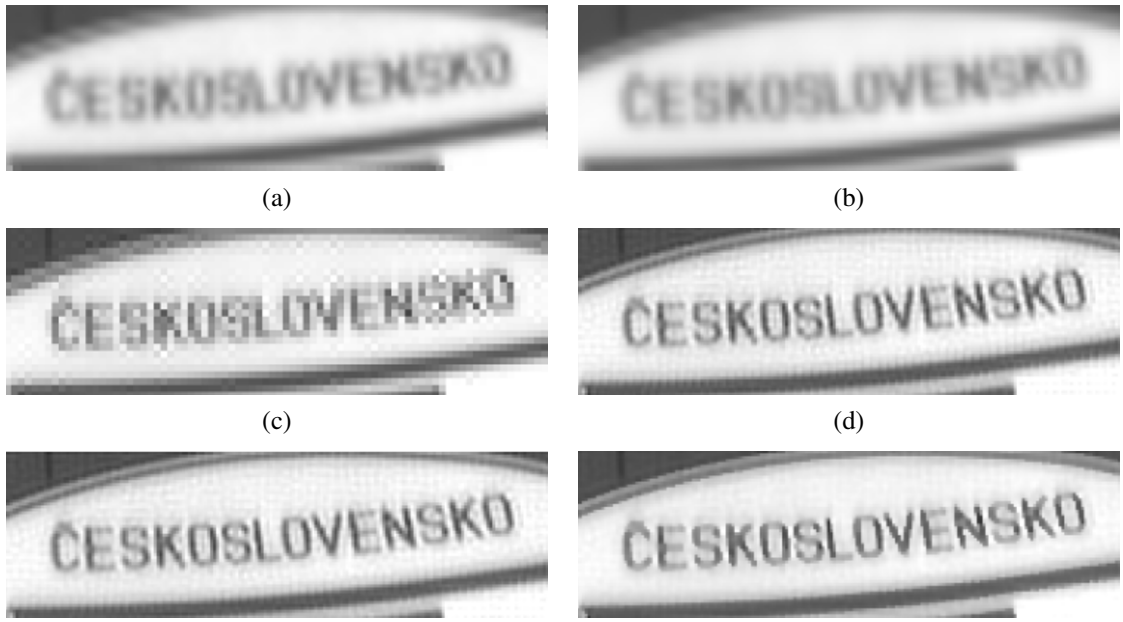


Figure 10.31 Super-resolution results using a variety of image priors (Capel 2001): (a) Low-res ROI (bicubic $3\times$ zoom); (b) average image; (c) MLE @ $1.25\times$ pixel-zoom; (d) simple $\|x\|^2$ prior ($\lambda = 0.004$); (e) GMRF ($\lambda = 0.003$); (f) HMRF ($\lambda = 0.01$, $\alpha = 0.04$). 10 images are used as input and a $3\times$ super-resolved image is produced in each case, except for the MLE result in (c).

As discussed in Section 3.7.2, when ρ_p is quadratic, this is a form of Tikhonov regularization (Section 3.7.1), and the overall problem is still linear least squares. The resulting prior image model is a Gaussian Markov random field (GMRF), which can be extended to other (e.g., diagonal) differences, as in (Capel 2004) (Figure 10.31).

Unfortunately, GMRFs tend to produce solutions with visible ripples, which can also be interpreted as increased noise sensitivity in middle frequencies (Exercise 3.17). A better image prior is a robust prior that encourages piecewise continuous solutions (Black and Rangarajan 1996), see Appendix B.3. Examples of such priors include the Huber potential (Schultz and Stevenson 1996; Capel and Zisserman 2003), which is a blend of a Gaussian with a longer-tailed Laplacian, and the even sparser (heavier-tailed) hyper-Laplacians used by Levin, Fergus, Durand *et al.* (2007) and Krishnan and Fergus (2009). It is also possible to learn the parameters for such priors using cross-validation (Capel 2004; Pickup 2007).

While sparse (robust) derivative priors can reduce rippling effects and increase edge sharpness, they cannot *hallucinate* higher-frequency texture or details. To do this, a training set of sample images can be used to find plausible mappings between low-frequency originals and the missing higher frequencies. Inspired by some of the example-based texture synthesis algorithms we discuss in Section 10.5, the *example-based super-resolution* algorithm developed by Freeman, Jones, and Pasztor (2002) uses training images to *learn* the mapping between local texture patches and missing higher-frequency details. To ensure that overlapping patches are similar in appearance, a Markov random field is used and optimized



(a) (b) (c)

Figure 10.32 Example-based super-resolution: (a) original 32×32 low-resolution image; (b) example-based super-resolved 256×256 image (Freeman, Jones, and Pasztor 2002) © 2002 IEEE; (c) upsampling via imposed edge statistics (Fattal 2007) © 2007 ACM.

using either belief propagation (Freeman, Pasztor, and Carmichael 2000) or a raster-scan deterministic variant (Freeman, Jones, and Pasztor 2002). Figure 10.32 shows the results of hallucinating missing details using this approach and compares these results to a more recent algorithm by Fattal (2007). This latter algorithm learns to predict oriented gradient magnitudes in the finer resolution image based on a pixel's location relative to the nearest detected edge along with the corresponding edge statistics (magnitude and width). It is also possible to combine sparse (robust) derivative priors with example-based super-resolution, as shown by Tappen, Russell, and Freeman (2003).

An alternative (but closely related) form of hallucination is to *recognize* the parts of a training database of images to which a low-resolution pixel might correspond. In their work, Baker and Kanade (2002) use local derivative-of-Gaussian filter responses as features and then match *parent structure* vectors in a manner similar to De Bonet (1997).[19] The high-frequency gradient at each recognized training image location is then used as a constraint on the super-resolved image, along with the usual reconstruction (prediction) equation (10.27). Figure 10.33 shows the result of hallucinating higher-resolution faces from lower-resolution inputs; Baker and Kanade (2002) also show examples of super-resolving known-font text. Exercise 10.7 gives more details on how to implement and test one or more of these super-resolution techniques.

Under favorable conditions, super-resolution and related upsampling techniques can increase the resolution of a well-photographed image or image collection. When the input images are blurry to start with, the best one can often hope for is to reduce the amount of blur. This problem is closely related super-resolution, with the biggest differences being that the blur kernel b is usually much larger and the downsampling factor D is unity. A large literature on image deblurring exists; some of the more recent publications with nice literature reviews include those by Fergus, Singh, Hertzmann *et al.* (2006), Yuan, Sun, Quan *et al.* (2008), and Joshi, Zitnick, Szeliski *et al.* (2009). It is also possible to reduce blur by combining sharp (but noisy) images with blurrier (but cleaner) images (Yuan, Sun, Quan *et al.* 2007), take lots of

[19] For face super-resolution, where all the images are pre-aligned, only corresponding pixels in different images are examined.

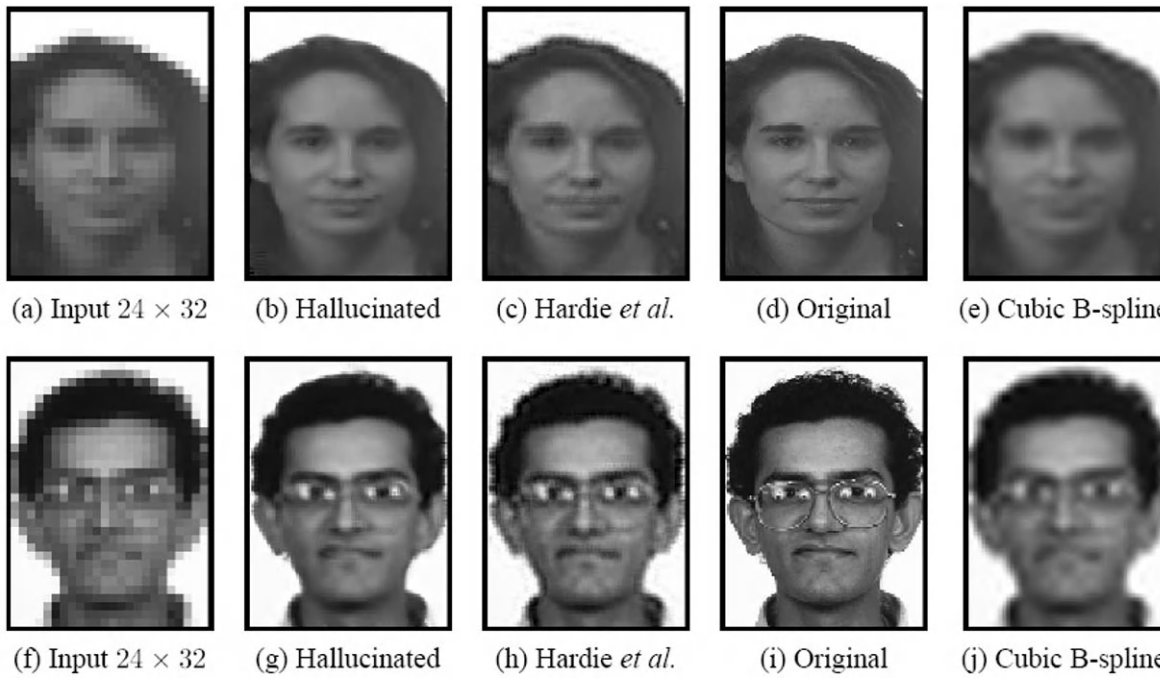


Figure 10.33 Recognition-based super-resolution (Baker and Kanade 2002) © 2002 IEEE. The *Hallucinated* column shows the results of the recognition-based algorithm compared to the regularization-based approach of Hardie, Barnard, and Armstrong (1997).

quick exposures[20] (Hasinoff and Kutulakos 2008; Hasinoff, Kutulakos, Durand *et al.* 2009; Hasinoff, Durand, and Freeman 2010), or use *coded aperture* techniques to simultaneously estimate depth and reduce blur (Levin, Fergus, Durand *et al.* 2007; Zhou, Lin, and Nayar 2009).

10.3.1 Color image demosaicing

A special case of super-resolution, which is used daily in most digital still cameras, is the process of *demosaicing* samples from a color filter array (CFA) into a full-color RGB image. Figure 10.34 shows the most commonly used CFA known as the *Bayer pattern*, which has twice as many green (G) sensors as red and blue sensors.

The process of going from the known CFA pixels values to the full RGB image is quite challenging. Unlike regular super-resolution, where small errors in guessing unknown values usually show up as blur or aliasing, demosaicing artifacts often produce spurious colors or high-frequency patterned *zippering*, which are quite visible to the eye (Figure 10.35b).

Over the years, a variety of techniques have been developed for image demosaicing (Kimmel 1999). Bennett, Uyttendaele, Zitnick *et al.* (2006) present a recently developed algorithm along with some good references, while Longere, Delahunt, Zhang *et al.* (2002) and Tappen, Russell, and Freeman (2003) compare some previously developed techniques using perceptually motivated metrics. To reduce the zippering effect, most techniques use the edge or

[20] The SONY DSC-WX1 takes multiple shots to produce better low-light photos.

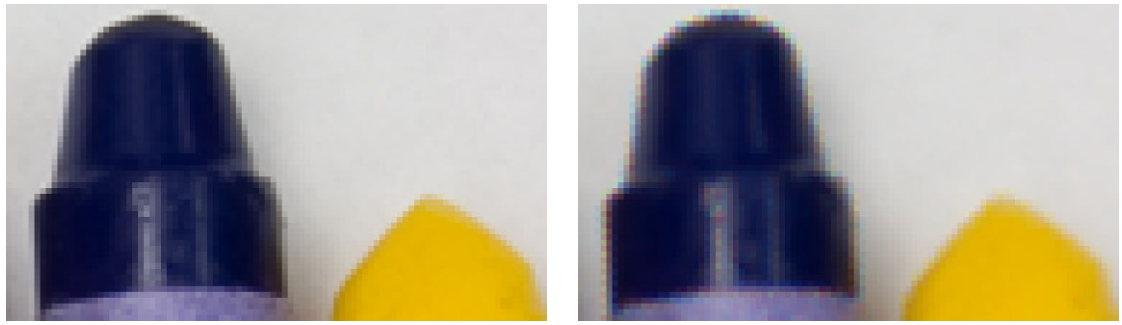
G	R	G	R
B	G	B	G
G	R	G	R
B	G	B	G

(a)

rGb	Rgb	rGb	Rgb
rgB	rGb	rgB	rGb
rGb	Rgb	rGb	Rgb
rgB	rGb	rgB	rGb

(b)

Figure 10.34 Bayer RGB pattern: (a) color filter array layout; (b) interpolated pixel values, with unknown (guessed) values shown as lower case.



(a) (b)

(c) (d)

Figure 10.35 CFA demosaicing results (Bennett, Uyttendaele, Zitnick *et al.* 2006) © 2006 Springer: (a) original full-resolution image (a color subsampled version is used as the input to the algorithms); (b) bilinear interpolation results, showing color fringing near the tip of the blue crayon and zippering near its left (vertical) edge; (c) the high-quality linear interpolation results of Malvar, He, and Cutler (2004) (note the strong halo/checkerboard artifacts on the yellow crayon); (d) using the local two-color prior of Bennett, Uyttendaele, Zitnick *et al.* (2006).

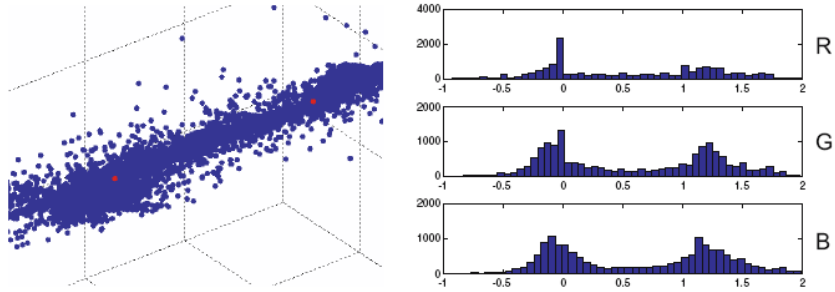


Figure 10.36 Two-color model computed from a collection of local 5×5 neighborhoods (Bennett, Uyttendaele, Zitnick *et al.* 2006) © 2006 Springer. After two-means clustering and reprojection along the line joining the two dominant colors (red dots), the majority of the pixels fall near the fitted line. The distribution along the line, projected along the RGB axes, is peaked at 0 and 1, the two dominant colors.

gradient information from the green channel, which is more reliable because it is sampled more densely, to infer plausible values for the red and blue channels, which are more sparsely sampled.

To reduce color fringing, some techniques perform a color space analysis, e.g., using median filtering on color opponent channels (Longere, Delahunt, Zhang *et al.* 2002). The approach of Bennett, Uyttendaele, Zitnick *et al.* (2006) locally forms a two-color model from an initial demosaicing result, using a moving 5×5 window to find the two dominant colors (Figure 10.36).[21]

Once the local color model has been estimated at each pixel, a Bayesian approach is then used to encourage pixel values to lie along each color line and to cluster around the dominant color values, which reduces halos (Figure 10.35d). The Bayesian approach also supports the simultaneous application of demosaicing and super-resolution, i.e., multiple CFA inputs can be merged into a higher-quality full-color image, which becomes more important as additional processing becomes incorporated into today's cameras.

10.3.2 *Application*: Colorization

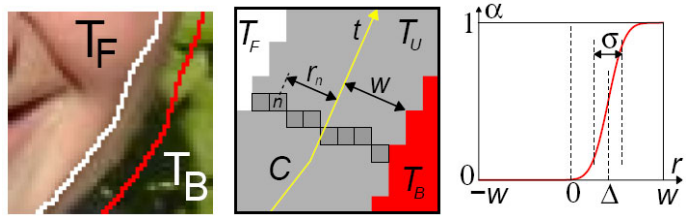
Although not strictly an example of super-resolution, the process of *colorization*, i.e., manually adding colors to a "black and white" (grayscale) image, is another example of a sparse interpolation problem. In most applications of colorization, the user draws some scribbles indicating the desired colors in certain regions (Figure 10.37a) and the system interpolates the specified chrominance (u, v) values to the whole image, which are then re-combined with the input luminance channel to produce a final colorized image, as shown in Figure 10.37b. In the system developed by Levin, Lischinski, and Weiss (2004), the interpolation is performed using locally weighted regularization (3.100), where the local smoothness weights are inversely proportional to luminance gradients. This approach to locally weighted regularization has inspired later algorithms for high dynamic range tone mapping (Lischinski, Farbman, Uyttendaele *et al.* 2006a), see Section 10.2.1, as well as other applications of the weighted least

[21] Previous work on locally linear color models (Klinker, Shafer, and Kanade 1990; Omer and Werman 2004) focuses on color and illumination variation within a single material, whereas Bennett, Uyttendaele, Zitnick *et al.* (2006) use the two-color model to describe variations across color (material) edges.



(a) (b) (c)

Figure 10.37 Colorization using optimization (Levin, Lischinski, and Weiss 2004) © 2004 ACM: (a) grayscale image some color scribbles overlaid; (b) resulting colorized image; (c) original color image from which the grayscale image and the chrominance values for the scribbles were derived. Original photograph by Rotem Weiss.



(a) (b) (c)

Figure 10.38 Softening a hard segmentation boundary (border matting) (Rother, Kolmogorov, and Blake 2004) © 2004 ACM: (a) the region surrounding a segmentation boundary where pixels of mixed foreground and background colors are visible; (b) pixel values along the boundary are used to compute a soft alpha matte; (c) at each point along the curve t, a displacement Δ and a width σ are estimated.

squares (WLS) formulation (Farbman, Fattal, Lischinski *et al.* 2008). An alternative approach to performing the sparse chrominance interpolation based on geodesic (edge-aware) distance functions has been developed by Yatziv and Sapiro (2006).

10.4 Image matting and compositing

Image matting and compositing is the process of cutting a foreground object out of one image and pasting it against a new background (Smith and Blinn 1996; Wang and Cohen 2007a). It is commonly used in television and film production to composite a live actor in front of computer-generated imagery such as weather maps or 3D virtual characters and scenery (Wright 2006; Brinkmann 2008).

We have already seen a number of tools for interactively segmenting objects in an image, including snakes (Section 5.1.1), scissors (Section 5.1.3), and GrabCut segmentation (Section 5.5). While these techniques can generate reasonable pixel-accurate segmentations, they fail to capture the subtle interplay of foreground and background colors at *mixed pixels* along the boundary (Szeliski and Golland 1999) (Figure 10.38a).

In order to successfully copy a foreground object from one image to another without

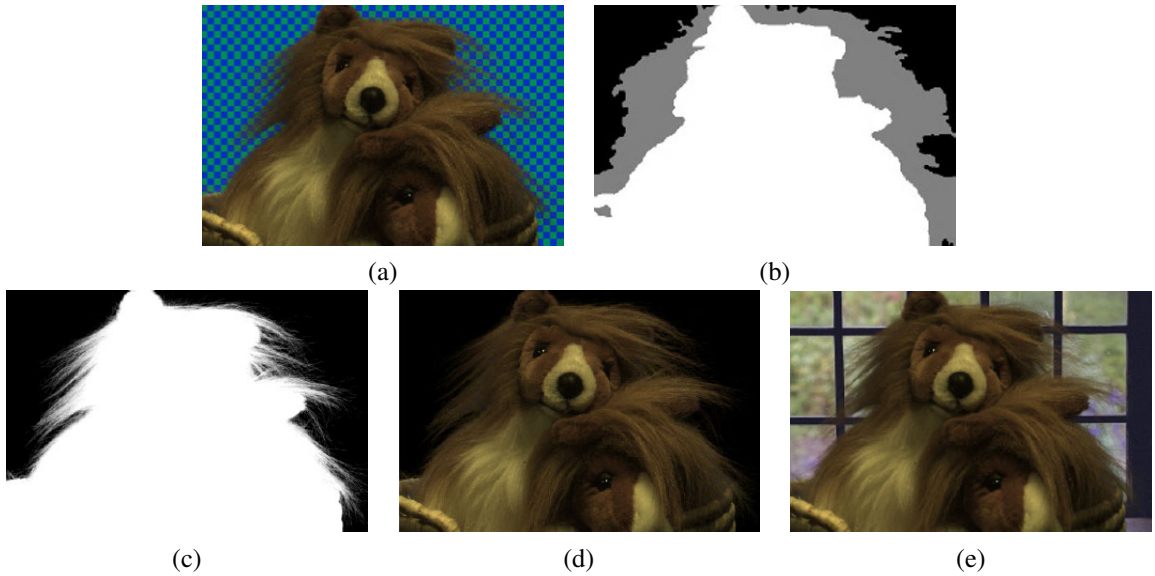


Figure 10.39 Natural image matting (Chuang, Curless, Salesin *et al.* 2001) © 2001 IEEE: (a) input image with a "natural" (non-constant) background; (b) hand-drawn trimap—gray indicates unknown regions; (c) extracted alpha map; (d) extracted (premultiplied) foreground colors; (e) composite over a new background.

visible discretization artifacts, we need to *pull a matte*, i.e., to estimate a soft opacity channel α and the uncontaminated foreground colors F from the input composite image C. Recall from Section 3.1.3 (Figure 3.4) that the compositing equation (3.8) can be written as

$$C = (1 - \alpha)B + \alpha F. \tag{10.30}$$

This operator attenuates the influence of the background image B by a factor $(1 - \alpha)$ and then adds in the (partial) color values corresponding to the foreground element F.

While the compositing operation is easy to implement, the reverse *matting* operation of estimating F, α, and B given an input image C is much more challenging (Figure 10.39). To see why, observe that while the composite pixel color C provides three measurements, the F, α, and B unknowns have a total of seven degrees of freedom. Devising techniques to estimate these unknowns despite the underconstrained nature of the problem is the essence of image matting.

In this section, we review a number of image matting techniques. We begin with *blue screen matting*, which assumes that the background is a constant known color, and discuss its variants, two-screen matting (when multiple backgrounds can be used) and difference matting (where the known background is arbitrary). We then discuss local variants of *natural image matting*, where both the foreground and background are unknown. In these applications, it is usual to first specify a *trimap*, i.e., a three-way labeling of the image into foreground, background, and unknown regions (Figure 10.39b). Next, we present some global optimization approaches to natural image matting. Finally, we discuss variants on the matting problem, including shadow matting, flash matting, and environment matting.

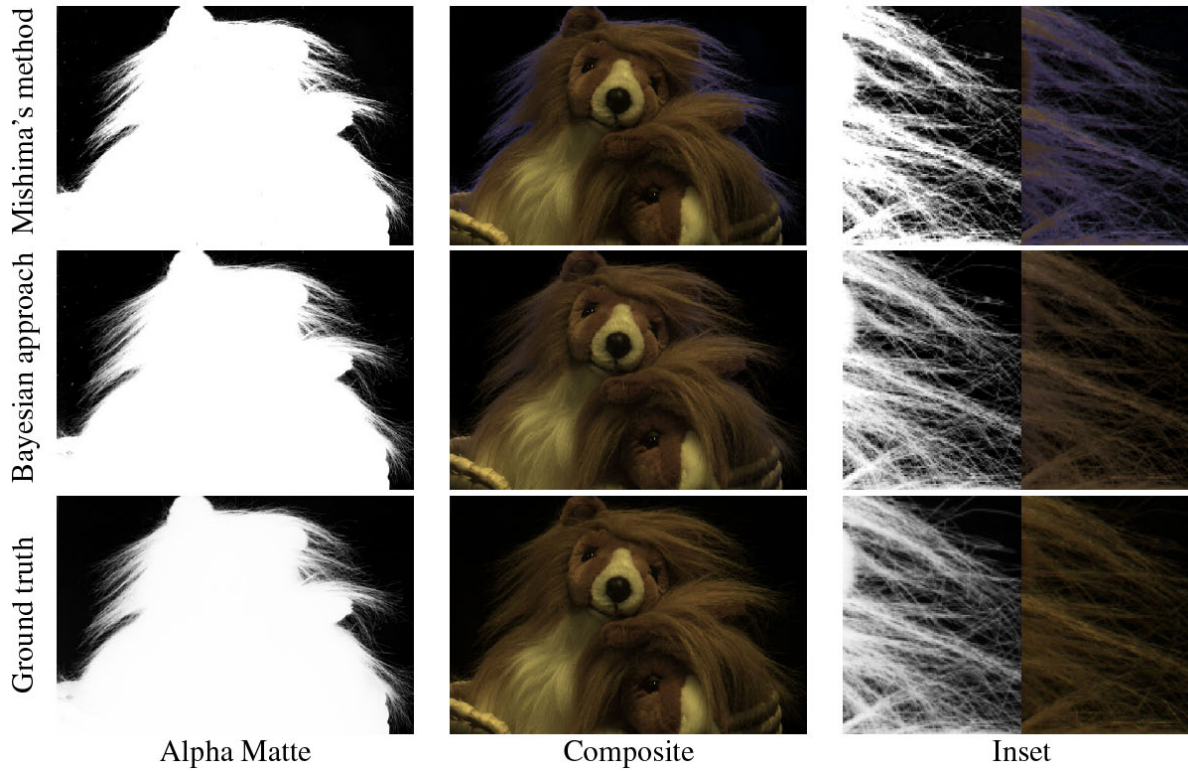


Figure 10.40 Blue-screen matting results (Chuang, Curless, Salesin *et al.* 2001) © 2001 IEEE. Mishima's method produces visible blue spill (color fringing in the hair), while Chuang's Bayesian matting approach produces accurate results.

10.4.1 Blue screen matting

Blue screen matting involves filming an actor (or object) in front of a constant colored background. While originally bright blue was the preferred color, bright green is now more commonly used (Wright 2006; Brinkmann 2008). Smith and Blinn (1996) discuss a number of techniques for blue screen matting, which are mostly described in patents rather than in the open research literature. Early techniques used linear combination of object color channels with user-tuned parameters to estimate the opacity α.

Chuang, Curless, Salesin *et al.* (2001) describe a newer technique called Mishima's algorithm, which involves fitting two polyhedral surfaces (centered at the mean background color), separating the foreground and background color distributions and then measuring the relative distance of a novel color to these surfaces to estimate α (Figure 10.41e). While this technique works well in many studio settings, it can still suffer from *blue spill*, where translucent pixels around the edges of an object acquire some of the background blue coloration (Figure 10.40).

Two-screen matting. In their paper, Smith and Blinn (1996) also introduce an algorithm called *triangulation matting* that uses more than one known background color to overconstrain the equations required to estimate the opacity α and foreground color F.

For example, consider in the compositing equation (10.30) setting the background color to black, i.e., $B = 0$. The resulting composite image C is therefore equal to αF. Replacing the background color with a different known non-zero value B now results in

$$C - \alpha F = (1 - \alpha)B, \tag{10.31}$$

which is an overconstrained set of (color) equations for estimating α. In practice, B should be chosen so as not to saturate C and, for best accuracy, several values of B should be used. It is also important that colors be linearized before processing, which is the case for *all* image matting algorithms. Papers that generate ground truth alpha mattes for evaluation purposes normally use these techniques to obtain accurate matte estimates (Chuang, Curless, Salesin *et al.* 2001; Wang and Cohen 2007b; Levin, Acha, and Lischinski 2008; Rhemann, Rother, Rav-Acha *et al.* 2008; Rhemann, Rother, Wang *et al.* 2009).[22] Exercise 10.8 has you do this as well.

Difference matting. A related approach when the background is irregular but known is called *difference matting* (Wright 2006; Brinkmann 2008). It is most commonly used when the actor or object is filmed against a static background, e.g., for office videoconferencing, person tracking applications (Toyama, Krumm, Brumitt *et al.* 1999), or to produce silhouettes for volumetric 3D reconstruction techniques (Section 11.6.2) (Szeliski 1993; Seitz and Dyer 1997; Seitz, Curless, Diebel *et al.* 2006). It can also be used with a panning camera where the background is composited from frames where the foreground has been removed using a *garbage matte* (Section 10.4.5) (Chuang, Agarwala, Curless *et al.* 2002). Another recent application is the detection of visual continuity errors in films, i.e., differences in the background when a shot is re-taken at later time (Pickup and Zisserman 2009).

In the case where the foreground and background motions can both be specified with parametric transforms, high-quality mattes can be extracted using a generalization of triangulation matting (Wexler, Fitzgibbon, and Zisserman 2002). When frames need to be processed independently, however, the results are often of poor quality (Figure 10.42). In such cases, using a pair of stereo cameras as input can dramatically improve the quality of the results (Criminisi, Cross, Blake *et al.* 2006; Yin, Criminisi, Winn *et al.* 2007).

10.4.2 Natural image matting

The most general version of image matting is when nothing is known about the background except, perhaps, for a rough segmentation of the scene into foreground, background, and unknown regions, which is known as the *trimap* (Figure 10.39b). Some recent techniques, however, relax this requirement and allow the user to just draw a few strokes or scribbles in the image, see Figures 10.45 and 10.46 (Wang and Cohen 2005; Wang, Agrawala, and Cohen 2007; Levin, Lischinski, and Weiss 2008; Rhemann, Rother, Rav-Acha *et al.* 2008; Rhemann, Rother, and Gelautz 2008). Fully automated single image matting results have also been reported (Levin, Acha, and Lischinski 2008; Singaraju, Rother, and Rhemann 2009). The survey paper by Wang and Cohen (2007a) has detailed descriptions and comparisons of all of these techniques, a selection of which are described briefly below.

[22] See the alpha matting evaluation Web site at http://alphamatting.com/.

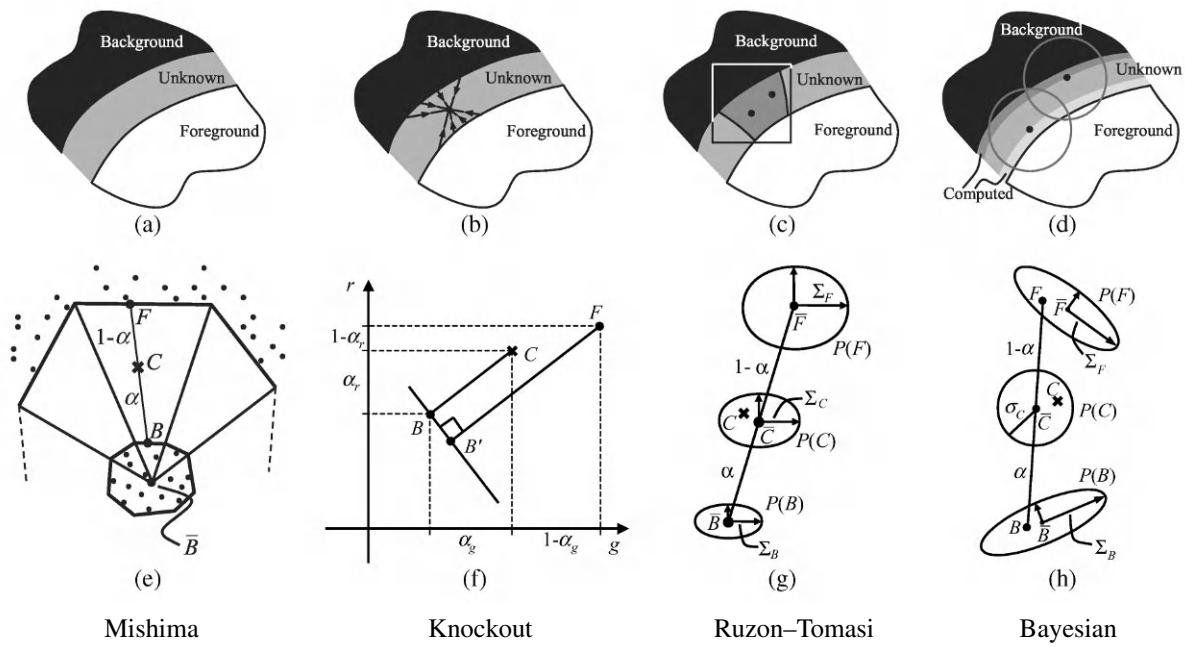


Figure 10.41 Image matting algorithms (Chuang, Curless, Salesin *et al.* 2001) © 2001 IEEE. Mishima's algorithm models global foreground and background color distribution as polyhedral surfaces centered around the mean background (blue) color. Knockout uses a local color estimate of foreground and background for each pixel and computes α along each color axis. Ruzon and Tomasi's algorithm locally models foreground and background colors and variances. Chuang *et al.*'s Bayesian matting approach computes a MAP estimate of (fractional) foreground color and opacity given the local foreground and background distributions.

A relatively simple algorithm for performing natural image matting is Knockout, as described by Chuang, Curless, Salesin *et al.* (2001) and illustrated in Figure 10.41f. In this algorithm, the nearest known foreground and background pixels (in image space) are determined and then blended with neighboring known pixels to produce a per-pixel foreground F and background B color estimate. The background color is then adjusted so that the measured color C lies on the line between F and B. Finally, opacity α is estimated on a per-channel basis, and the three estimates are combined based on per-channel color differences. (This is an approximation to the least squares solution for α.) Figure 10.42 shows that Knockout has problems when the background consists of more than one dominant local color.

More accurate matting results can be obtained if we treat the foreground and background colors as distributions sampled over some region (Figure 10.41g–h). Ruzon and Tomasi (2000) model local color distributions as mixtures of (uncorrelated) Gaussians and compute these models in strips. They then find the pairing of mixture components F and B that best describes the observed color C, compute the α as the relative distance between these means, and adjust the estimates of F and B so they are collinear with C.

Chuang, Curless, Salesin *et al.* (2001) and Hillman, Hannah, and Renshaw (2001) use full 3×3 color covariance matrices to model mixtures of correlated Gaussians, and compute estimates independently for each pixel. Matte extraction proceeds in strips starting from known color values growing into the unknown regions, so that recently computed F and B

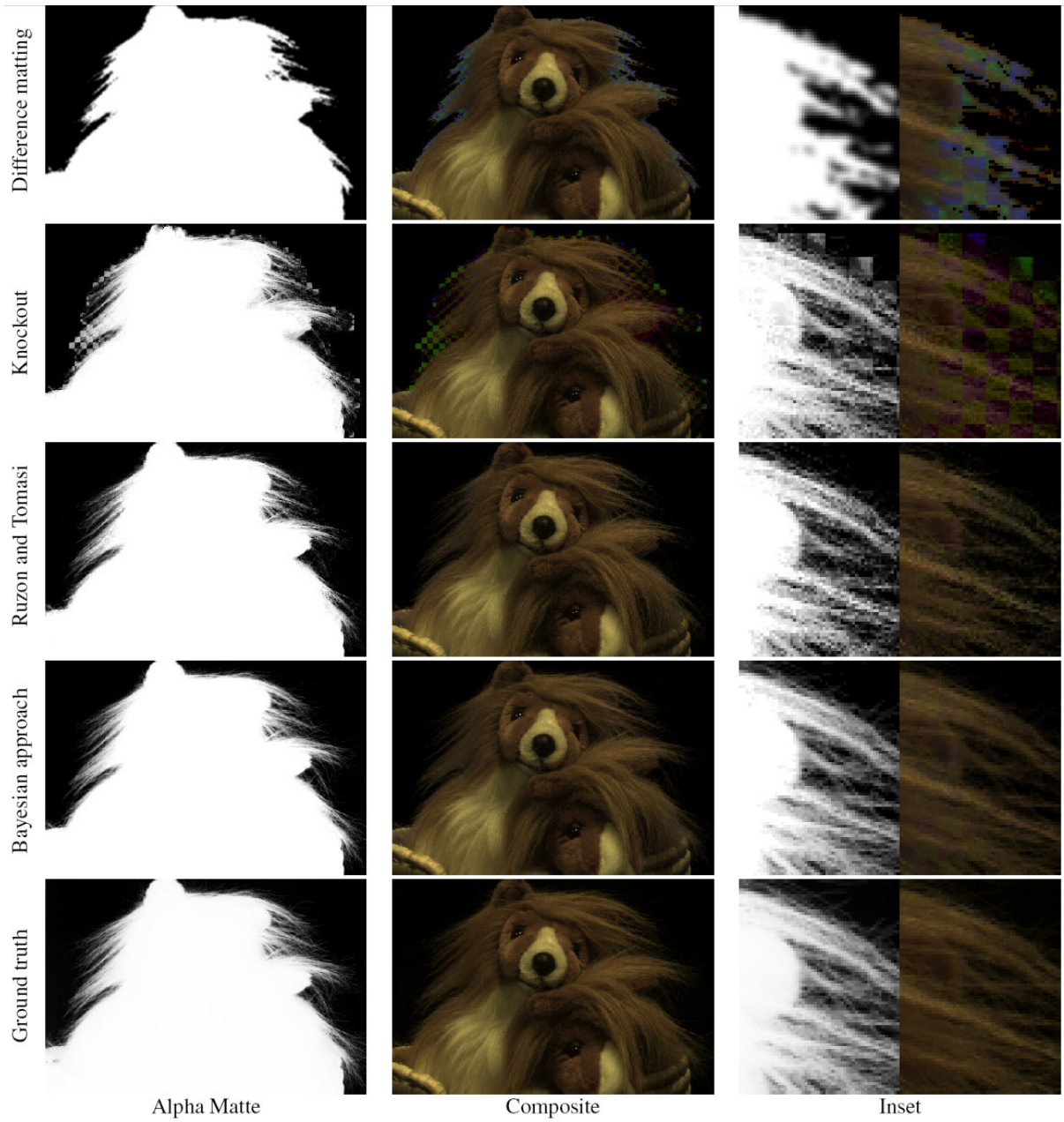


Figure 10.42 Natural image matting results (Chuang, Curless, Salesin *et al.* 2001) © 2001 IEEE. Difference matting and Knockout both perform poorly on this kind of background, while the more recent natural image matting techniques perform well. Chuang *et al.*'s results are slightly smoother and closer to the ground truth.

colors can be used in later stages.

To estimate the most likely value of an unknown pixel's opacity and (unmixed) foreground and background colors, Chuang *et al.* use a fully Bayesian formulation that maximizes

$$P(F, B, \alpha|C) = P(C|F, B, \alpha)P(F)P(B)P(\alpha)/P(C). \tag{10.32}$$

This is equivalent to minimizing the negative log likelihood

$$L(F, B, \alpha|C) = L(C|F, B, \alpha) + L(F) + L(B) + L(\alpha) \tag{10.33}$$

(dropping the $L(C)$ term since it is constant).

Let us examine each of these terms in turn. The first, $L(C|F, B, \alpha)$, is the likelihood that pixel color C was observed given values for the unknowns (F, B, α). If we assume Gaussian noise in our observation with variance σ_C^2, this negative log likelihood (data term) is

$$L(C) = \tfrac{1}{2}\|C - [\alpha F + (1 - \alpha)B]\|^2/\sigma_C^2, \tag{10.34}$$

as illustrated in Figure 10.41h.

The second term, $L(F)$, corresponds to the likelihood that a particular foreground color F comes from the mixture of Gaussians distribution. After partitioning the sample foreground colors into clusters, a weighted mean and covariance is computed, where the weights are proportional to a given foreground pixel's opacity and distance from the unknown pixel. The negative log likelihood for each cluster is thus given by

$$L(F) = (F - \overline{F})^T \Sigma_F^{-1} (F - \overline{F}). \tag{10.35}$$

A similar method is used to estimate unknown background color distributions. If the background is already known, i.e., for blue screen or difference matting applications, its measured color value and variance are used instead.

An alternative to modeling the foreground and background color distributions as mixtures of Gaussians is to keep around the original color samples and to compute the most likely pairings that explain the observed color C (Wang and Cohen 2005, 2007b). These techniques are described in more detail in (Wang and Cohen 2007a).

In their Bayesian matting paper, Chuang, Curless, Salesin *et al.* (2001) assume a constant (non-informative) distribution for $L(\alpha)$. More recent papers assume this distribution to be more peaked around 0 and 1, or sometimes use Markov random fields (MRFs) to define a global correlated prior on $P(\alpha)$ (Wang and Cohen 2007a).

To compute the most likely estimates for (F, B, α), the Bayesian matting algorithm alternates between computing (F, B) and α, since each of these problems is quadratic and hence can be solved as a small linear system. When several color clusters are estimated, the most likely pairing of foreground and background color clusters is used.

Bayesian image matting produces results that improve on the original natural image matting algorithm by Ruzon and Tomasi (2000), as can be seen in Figure 10.42. However, compared to more recent techniques (Wang and Cohen 2007a), its performance is not as good for complex background or inaccurate trimaps (Figure 10.44).

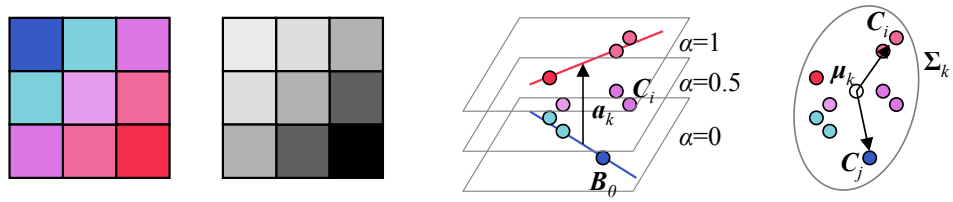


Figure 10.43 Color line matting (Levin, Lischinski, and Weiss 2008): (a) local 3×3 patch of colors; (b) potential assignment of α values; (c) foreground and background color lines, the vector $\boldsymbol{a}_k$ joining their closest points of intersection, and the family of parallel planes of constant α values, $\alpha_i = \boldsymbol{a}_k \cdot (\boldsymbol{C}_i - \boldsymbol{B}_0)$; (d) a scatter plot of sample colors and the deviations from the mean μ_k for two sample colors $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$.

10.4.3 Optimization-based matting

An alternative to estimating each pixel's opacity and foreground color independently is to use global optimization to compute a matte that takes into account correlations between neighboring α values. Two examples of this are border matting in the GrabCut interactive segmentation system (Rother, Kolmogorov, and Blake 2004) and Poisson Matting (Sun, Jia, Tang *et al.* 2004).

Border matting first dilates the region around the binary segmentation produced by GrabCut (Section 5.5) and then solves for a sub-pixel boundary location Δ and a blur width σ for every point along the boundary (Figure 10.38). Smoothness in these parameters along the boundary is enforced using regularization and the optimization is performed using dynamic programming. While this technique can obtain good results for smooth boundaries, such as a person's face, it has difficulty with fine details, such as hair.

Poisson matting (Sun, Jia, Tang *et al.* 2004) assumes a known foreground and background color for each pixel in the trimap (as with Bayesian matting). However, instead of independently estimating each α value, it assumes that the gradient of the alpha matte and the gradient of the color image are related by

$$\nabla\alpha = \frac{F - B}{\|F - B\|^2} \cdot \nabla C, \tag{10.36}$$

which can be derived by taking gradients of both sides of (10.30) and assuming that the foreground and background vary slowly. The per-pixel gradient estimates are then integrated into a continuous $\alpha(\boldsymbol{x})$ field using the regularization (least squares) technique first described in Section 3.7.1 (3.100) and subsequently used in Poisson blending (Section 9.3.4, 9.44) and gradient-based dynamic range compression mapping (Section 10.2.1, 10.19). This technique works well when good foreground and background color estimates are available and these colors vary slowly.

Instead of computing per-pixel foreground and background colors, Levin, Lischinski, and Weiss (2008) assume only that these color distribution can locally be well approximated as mixtures of two colors, which is known as the *color line model* (Figure 10.43a–c). Under this assumption, a closed-form estimate for α at each pixel i in a (say, 3×3) window W_k is given by

$$\alpha_i = \boldsymbol{a}_k \cdot (\boldsymbol{C}_i - \boldsymbol{B}_0) = \boldsymbol{a}_k \cdot \boldsymbol{C} + b_k, \tag{10.37}$$

where $\boldsymbol{C}_i$ is the pixel color treated as a three-vector, $\boldsymbol{B}_0$ is any pixel along the background color line, and $\boldsymbol{a}_k$ is the vector joining the two closest points on the foreground and background color lines, as shown in Figure 10.43c. (Note that the geometric derivation shown in this figure is an alternative to the algebraic derivation presented by Levin, Lischinski, and Weiss (2008).) Minimizing the deviations of the alpha values α_i from their respective color line models (10.37) over all overlapping windows W_k in the image gives rise to the cost

$$E_\alpha = \sum_k \left(\sum_{i \in W_k} (\alpha_i - \boldsymbol{a}_k \cdot \boldsymbol{C}_i - b_k)^2 + \epsilon \|\boldsymbol{a}_k\| \right), \tag{10.38}$$

where the ϵ term is used to regularize the value of $\boldsymbol{a}_k$ in the case where the two color distributions overlap (i.e., in constant α regions).

Because this formula is quadratic in the unknowns $\{(\boldsymbol{a}_k, b_k)\}$, they can be eliminated inside each window W_k, leading to a final energy

$$E_\alpha = \boldsymbol{\alpha}^T \boldsymbol{L} \boldsymbol{\alpha}, \tag{10.39}$$

where the entries in the $\boldsymbol{L}$ matrix are given by

$$L_{ij} = \sum_{k: i \in W_k \wedge j \in W_k} \left(\delta_{ij} - \frac{1}{M} \left(1 + (\boldsymbol{C}_i - \boldsymbol{\mu}_k)^T \hat{\boldsymbol{\Sigma}}_k^{-1} (\boldsymbol{C}_j - \boldsymbol{\mu}_k) \right) \right), \tag{10.40}$$

where $M = |W_k|$ is the number of pixels in each (overlapping) window, $\boldsymbol{\mu}_k$ is the mean color of the pixels in window W_k, and $\hat{\boldsymbol{\Sigma}}_k$ is the 3×3 covariance of the pixel colors plus $\epsilon/M \boldsymbol{I}$.

Figure 10.43d shows the intuition behind the entries in this affinity matrix, which is called the *matting Laplacian*. Note how when two pixels $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ in W_k point in opposite directions away from the mean $\boldsymbol{\mu}_k$, their weighted dot product is close to -1, and so their affinity becomes close to 0. Pixels close to each other in color space (and hence with similar expected α values) will have affinities close to $-2/M$.

Minimizing the quadratic energy (10.39) constrained by the known values of $\alpha = \{0, 1\}$ at scribbles only requires the solution of a sparse set of linear equations, which is why the authors call their technique a *closed-form solution* to natural image matting. Once α has been computed, the foreground and background colors are estimated using a least squares minimization of the compositing equation (10.30) regularized with a spatially varying first-order smoothness,

$$E_{B,F} = \sum_i \|C_i - [\alpha + F_i + (1 - \alpha_i) B_i]\|^2 + \lambda |\nabla \alpha_i| (\|\nabla F_i\|^2 + \|\nabla B_i\|^2), \tag{10.41}$$

where the $|\nabla \alpha_i|$ weight is applied separately for the x and y components of the F and B derivatives (Levin, Lischinski, and Weiss 2008).

Laplacian (closed-form) matting is just one of many optimization-based techniques surveyed and compared by Wang and Cohen (2007a). Some of these techniques use alternative formulations for the affinities or smoothness terms on the α matte, alternative estimation techniques such as belief propagation, or alternative representations (e.g., local histograms) for modeling local foreground and background color distributions (Wang and Cohen 2005, 2007b,c). Some of these techniques also provide real-time results as the user draws a contour

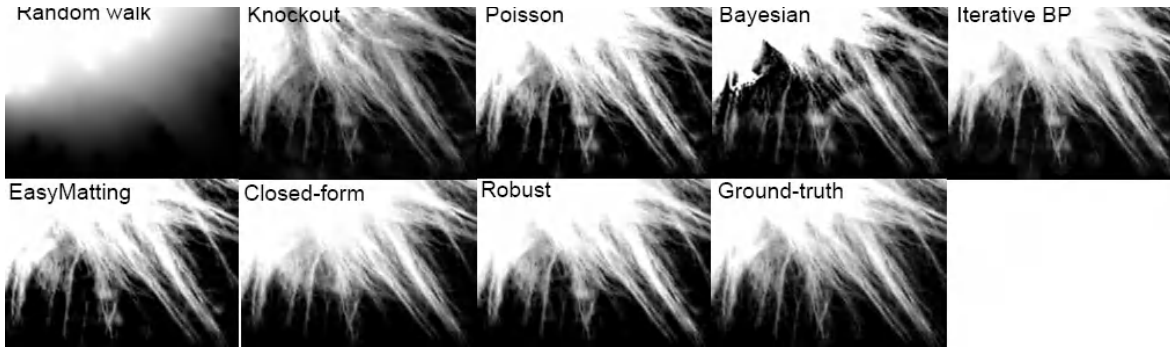


Figure 10.44 Comparative matting results for a medium accuracy trimap. Wang and Cohen (2007a) describe the individual techniques being compared.

line or sparse set of scribbles (Wang, Agrawala, and Cohen 2007; Rhemann, Rother, Rav-Acha *et al.* 2008) or even pre-segment the image into a small number of mattes that the user can select with simple clicks (Levin, Acha, and Lischinski 2008).

Figure 10.44 shows the results of running a number of the surveyed algorithms on a region of toy animal fur where a trimap has been specified, while Figure 10.45 shows results for techniques that can produce mattes with only a few scribbles as input. Figure 10.46 shows a result for an even more recent algorithm (Rhemann, Rother, Rav-Acha *et al.* 2008) that claims to outperform all of the techniques surveyed by Wang and Cohen (2007a).

Pasting. Once a matte has been pulled from an image, it is usually composited directly over the new background, unless the seams between the cutout and background regions are to be hidden, in which case Poisson blending (Pérez, Gangnet, and Blake 2003) can be used (Section 9.3.4).

In the latter case, it is helpful if the matte boundary passes through regions that either have little texture or look similar in the old and new images. Papers by Jia, Sun, Tang *et al.* (2006) and Wang and Cohen (2007c) explain how to do this.

10.4.4 Smoke, shadow, and flash matting

In addition to matting out solid objects with fractional boundaries, it is also possible to matte out translucent media such as smoke (Chuang, Agarwala, Curless *et al.* 2002). Starting with a video sequence, each pixel is modeled as a linear combination of its (unknown) background color and a constant foreground (smoke) color that is common to all pixels. Voting in color space is used to estimate this foreground color and the distance along each color line is used to estimate the per-pixel temporally varying alpha (Figure 10.47).

Extracting and re-inserting shadows is also possible using a related technique (Chuang, Goldman, Curless *et al.* 2003). Here, instead of assuming a constant foreground color, each pixel is assumed to vary between its fully lit and fully shadowed colors, which can be estimated by taking (robust) minimum and maximum values over time as a shadow passes over the scene (Exercise 10.9). The resulting fractional *shadow matte* can be used to re-project the shadow into a new scene. If the destination scene has a non-planar geometry, it can be

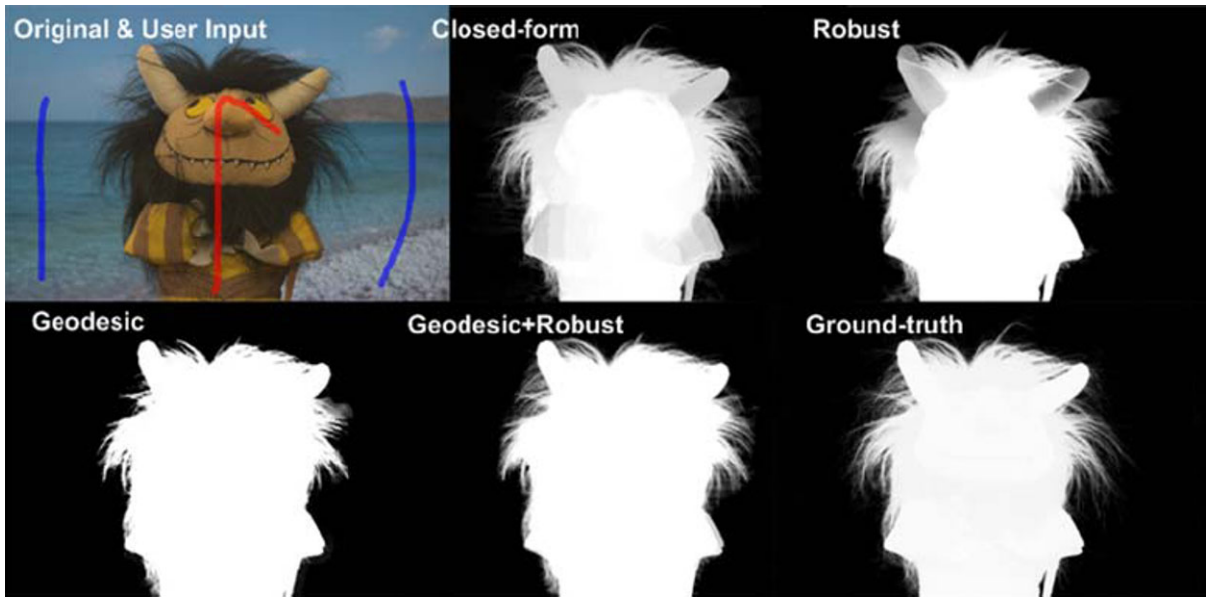


Figure 10.45 Comparative matting results with scribble-based inputs. Wang and Cohen (2007a) describe the individual techniques being compared.

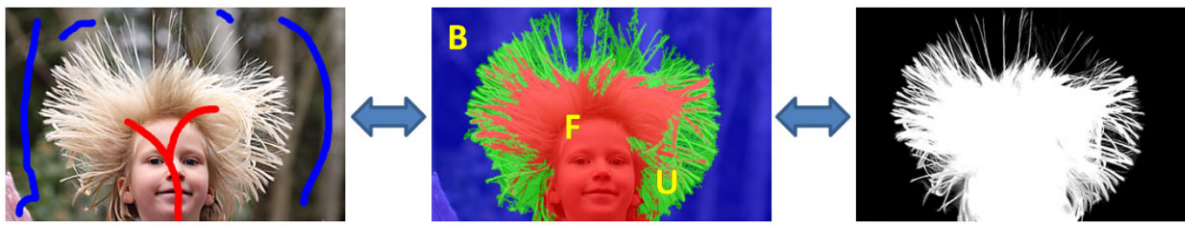


Figure 10.46 Stroke-based segmentation result (Rhemann, Rother, Rav-Acha *et al.* 2008) © 2008 IEEE.



(a) (b) (c) (d)

Figure 10.47 Smoke matting (Chuang, Agarwala, Curless *et al.* 2002) © 2002 ACM: (a) input video frame; (b) after removing the foreground object; (c) estimated alpha matte; (d) insertion of new objects into the background.



Figure 10.48 Shadow matting (Chuang, Goldman, Curless *et al.* 2003) © 2003 ACM. Instead of simply darkening the new scene with the shadow (c), shadow matting correctly dims the lit scene with the new shadow and drapes the shadow over 3D geometry (d).

scanned by waving a straight stick shadow across the scene. The new shadow matte can then be warped with the computed deformation field to have it drape correctly over the new scene (Figure 10.48).

The quality and reliability of matting algorithms can also be enhanced using more sophisticated acquisition systems. For example, taking a flash and non-flash image pair supports the reliable extraction of foreground mattes, which show up as regions of large illumination change between the two images (Sun, Li, Kang *et al.* 2006). Taking simultaneous video streams focused at different distances (McGuire, Matusik, Pfister *et al.* 2005) or using multi-camera arrays (Joshi, Matusik, and Avidan 2006) are also good approaches to producing high-quality mattes. These techniques are described in more detail in (Wang and Cohen 2007a).

Lastly, photographing a refractive object in front of a number of patterned backgrounds allows the object to be placed in novel 3D environments. These environment matting techniques (Zongker, Werner, Curless *et al.* 1999; Chuang, Zongker, Hindorff *et al.* 2000) are discussed in Section 13.4.

10.4.5 Video matting

While regular single-frame matting techniques such as blue or green screen matting (Smith and Blinn 1996; Wright 2006; Brinkmann 2008) can be applied to video sequences, the presence of moving objects can sometimes make the matting process easier, as portions of the background may get revealed in preceding or subsequent frames.

Chuang, Agarwala, Curless *et al.* (2002) describe a nice approach to this *video matting* problem, where foreground objects are first removed using a conservative *garbage matte* and the resulting *background plates* are aligned and composited to yield a high-quality background estimate. They also describe how trimaps drawn at sparse keyframes can be interpolated to in-between frames using bi-direction optic flow. Alternative approaches to video matting, such as rotoscoping, which involves drawing and tracking curves in video sequences (Agarwala, Hertzmann, Seitz *et al.* 2004), are discussed in the matting survey paper by Wang and Cohen (2007a).



Figure 10.49 Texture synthesis: (a) given a small patch of texture, the task is to synthesize (b) a similar-looking larger patch; (c) other semi-structured textures that are challenging to synthesize. (Images courtesy of Alyosha Efros.)

10.5 Texture analysis and synthesis

While texture analysis and synthesis may not at first seem like computational photography techniques, they are, in fact, widely used to repair defects, such as small holes, in images or to create non-photorealistic painterly renderings from regular photographs.

The problem of texture synthesis can be formulated as follows: given a small sample of a "texture" (Figure 10.49a), generate a larger similar-looking image (Figure 10.49b). As you can imagine, for certain sample textures, this problem can be quite challenging.

Traditional approaches to texture analysis and synthesis try to match the spectrum of the source image while generating shaped noise. Matching the frequency characteristics, which is equivalent to matching spatial correlations, is in itself not sufficient. The distributions of the responses at different frequencies must also match. Heeger and Bergen (1995) develop an algorithm that alternates between matching the histograms of multi-scale (steerable pyramid) responses and matching the final image histogram. Portilla and Simoncelli (2000) improve on this technique by also matching pairwise statistics across scale and orientations. De Bonet (1997) uses a coarse-to-fine strategy to find locations in the source texture with a similar *parent structure*, i.e., similar multi-scale oriented filter responses, and then randomly chooses one of these matching locations as the current sample value.

More recent texture synthesis algorithms sequentially generate texture pixels by looking for neighborhoods in the source texture that are similar to the currently synthesized image (Efros and Leung 1999). Consider the (as yet) unknown pixel $\boldsymbol{p}$ in the partially constructed texture on the left side of Figure 10.50. Since some of its neighboring pixels have been

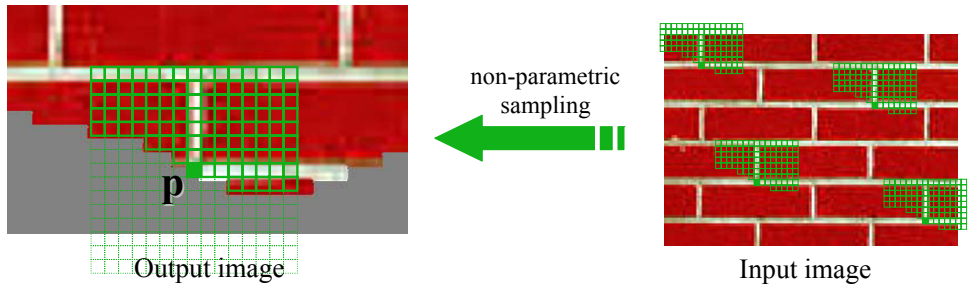


Figure 10.50 Texture synthesis using non-parametric sampling (Efros and Leung 1999). The value of the newest pixel $\boldsymbol{p}$ is randomly chosen from similar local (partial) patches in the source texture (input image). (Figure courtesy of Alyosha Efros.)

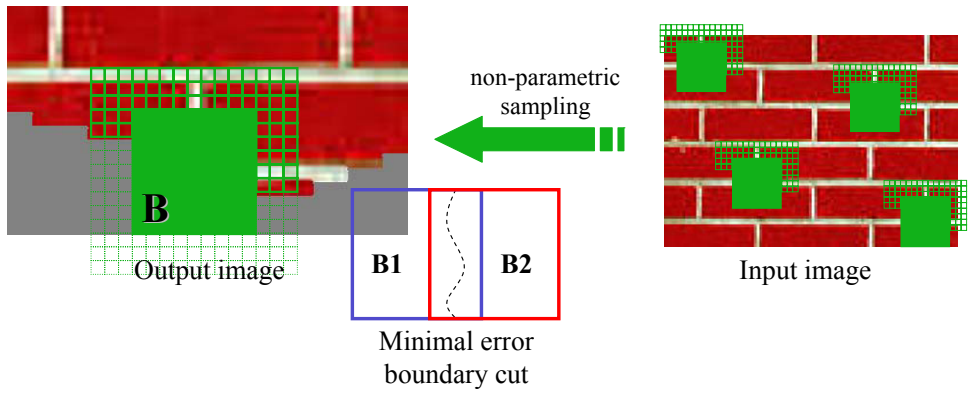


Figure 10.51 Texture synthesis by image quilting (Efros and Freeman 2001). Instead of generating a single pixel at a time, larger blocks are copied from the source texture. The transitions in the overlap regions between the selected blocks are then optimized using dynamic programming. (Figure courtesy of Alyosha Efros.)

already been synthesized, we can look for similar partial neighborhoods in the sample texture image on the right and randomly select one of these as the new value of $\boldsymbol{p}$. This process can be repeated down the new image either in a raster fashion or by scanning around the periphery ("onion peeling") when filling holes, as discussed in (Section 10.5.1). In their actual implementation, Efros and Leung (1999) find the most similar neighborhood and then include all other neighborhoods within a $d = (1 + \epsilon)$ distance, with $\epsilon = 0.1$. They also optionally weight the random pixel selections by the similarity metric d.

To accelerate this process and improve its visual quality, Wei and Levoy (2000) extend this technique using a coarse-to-fine generation process, where coarser levels of the pyramid, which have already been synthesized, are also considered during the matching (De Bonet 1997). To accelerate the nearest neighbor finding, tree-structured vector quantization is used.

Efros and Freeman (2001) propose an alternative acceleration and visual quality improvement technique. Instead of synthesizing a single pixel at a time, overlapping square blocks are selected using similarity with previously synthesized regions (Figure 10.51). Once the appropriate blocks have been selected, the seam between newly overlapping blocks is determined using dynamic programming. (Full graph cut seam selection is not required, since only



(a) (b) (c) (d)

Figure 10.52 Image inpainting (hole filling): (a–b) propagation along isophote directions (Bertalmio, Sapiro, Caselles *et al.* 2000) © 2000 ACM; (c–d) exemplar-based inpainting with confidence-based filling order (Criminisi, Pérez, and Toyama 2004).

one seam location per row is needed for a vertical boundary.) Because this process involves selecting small patches and them stitching them together, Efros and Freeman (2001) call their system *image quilting*. Komodakis and Tziritas (2007b) present an MRF-based version of this block synthesis algorithm that uses a new, efficient version of loopy belief propagation they call "Priority-BP".

10.5.1 *Application*: Hole filling and inpainting

Filling holes left behind when objects or defects are excised from photographs, which is known as *inpainting*, is one of the most common applications of texture synthesis. Such techniques are used not only to remove unwanted people or interlopers from photographs (King 1997) but also to fix small defects in old photos and movies (*scratch removal*) or to remove wires holding props or actors in mid-air during filming (*wire removal*). Bertalmio, Sapiro, Caselles *et al.* (2000) solve the problem by propagating pixel values along isophote (constant-value) directions interleaved with some anisotropic diffusion steps (Figure 10.52a–b). Telea (2004) develops a faster technique that uses the fast marching method from level sets (Section 5.1.4). However, these techniques will not hallucinate texture in the missing regions. Bertalmio, Vese, Sapiro *et al.* (2003) augment their earlier technique by adding synthetic texture to the infilled regions.

The example-based (non-parametric) texture generation techniques discussed in the previous section can also be used by filling the holes from the outside in (the "onion-peel" ordering). However, this approach may fail to propagate strong oriented structures. Criminisi, Pérez, and Toyama (2004) use exemplar-based texture synthesis where the order of synthesis is determined by the strength of the gradient along the region boundary (Figures 10.1d and 10.52c–d). Sun, Yuan, Jia *et al.* (2004) present a related approach where the user draws interactive lines to indicate where structures should be preferentially propagated. Additional techniques related to these approaches include those developed by Drori, Cohen-Or, and Yeshurun (2003), Kwatra, Schödl, Essa *et al.* (2003), Kwatra, Essa, Bobick *et al.* (2005), Wilczkowiak, Brostow, Tordoff *et al.* (2005), Komodakis and Tziritas (2007b), and Wexler, Shechtman, and Irani (2007).

Most hole filling algorithms borrow small pieces of the original image to fill in the holes. When a large database of source images is available, e.g., when images are taken from a



(a) (b) (c)

Figure 10.53 Texture transfer (Efros and Freeman 2001) © 2001 ACM: (a) reference (target) image; (b) source texture; (c) image (partially) rendered using the texture.

photo sharing site or the Internet, it is sometimes possible to copy a single contiguous image region to fill the hole. Hays and Efros (2007) present such a technique, which uses image context and boundary compatibility to select the source image, which is then blended with the original (holey) image using graph cuts and Poisson blending. This technique is discussed in more detail in Section 14.4.4 and Figure 14.46.

10.5.2 *Application*: Non-photorealistic rendering

Two more applications of the exemplar-based texture synthesis ideas are texture transfer (Efros and Freeman 2001) and image analogies (Hertzmann, Jacobs, Oliver *et al.* 2001), which are both examples of non-photorealistic rendering (Gooch and Gooch 2001).

In addition to using a source texture image, texture transfer also takes a reference (or target) image, and tries to match certain characteristics of the target image with the newly synthesized image. For example, the new image being rendered in Figure 10.53c not only tries to satisfy the usual similarity constraints with the source texture in Figure 10.53b, but it also tries to match the luminance characteristics of the reference image. Efros and Freeman (2001) mention that blurred image intensities or local image orientation angles are alternative quantities that could be matched.

Hertzmann, Jacobs, Oliver *et al.* (2001) formulate the following problem:

> Given a pair of images A and A' (the unfiltered and filtered source images, respectively), along with some additional unfiltered target image B, synthesize a new filtered target image B' such that
>
> $$A : A' :: B : B'.$$

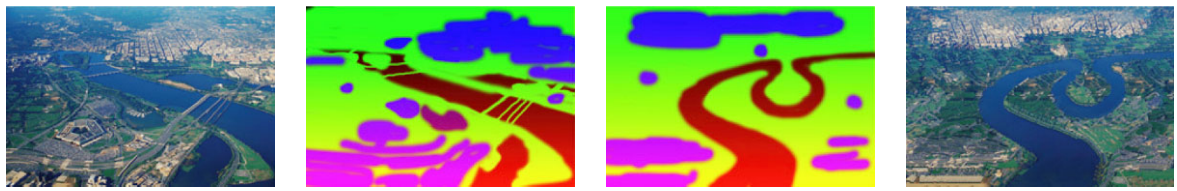
Instead of having the user program a certain non-photorealistic rendering effect, it is sufficient to supply the system with examples of before and after images, and let the system synthesize the novel image using exemplar-based synthesis, as shown in Figure 10.54.

The algorithm used to solve image analogies proceeds in a manner analogous to the texture synthesis algorithms of (Efros and Leung 1999; Wei and Levoy 2000). Once Gaussian pyramids have been computed for all of the source and reference images, the algorithm



A A' B B'

Figure 10.54 Image analogies (Hertzmann, Jacobs, Oliver *et al.* 2001) © 2001 ACM. Given an example pair of a source image A and its rendered (filtered) version A', generate the rendered version B' from another unfiltered source image B.



Original A' Painted A Novel painted B Novel textured B'

Figure 10.55 Texture-by-numbers (Hertzmann, Jacobs, Oliver *et al.* 2001) © 2001 ACM. Given a textured image A' and a hand-labeled (painted) version A, synthesize a new image B' given just the painted version B.

looks for neighborhoods in the source filtered pyramids generated from A' that are similar to the partially constructed neighborhood in B', while at the same time having similar multi-resolution appearances at corresponding locations in A and B. As with texture transfer, appearance characteristics can include not only (blurred) color or luminance values but also orientations.

This general framework allows image analogies to be applied to a variety of rendering tasks. In addition to exemplar-based non-photorealistic rendering, image analogies can be used for traditional texture synthesis, super-resolution, and texture transfer (using the same textured image for both A and A'). If only the filtered (rendered) image A' is available, as is the case with paintings, the missing reference image A can be hallucinated using a smart (edge preserving) blur operator. Finally, it is possible to train a system to perform *texture-by-numbers* by manually painting over a natural image with pseudocolors corresponding to pixels' semantic meanings, e.g., water, trees, and grass (Figure 10.55a–b). The resulting system can then convert a novel sketch into a fully rendered synthetic photograph (Figure 10.55c–d). In more recent work, Cheng, Vishwanathan, and Zhang (2008) add ideas from image quilting (Efros and Freeman 2001) and MRF inference (Komodakis, Tziritas, and Paragios 2008) to the basic image analogies algorithm, while Ramanarayanan and Bala (2007) recast this process as energy minimization, which means it can also be viewed as a conditional random field (Section 3.7.2), and devise an efficient algorithm to find a good minimum.

More traditional filtering and feature detection techniques can also be used for non-photorealistic rendering.[23] For example, pen-and-ink illustration (Winkenbach and Salesin

[23] For a good selection of papers, see the Symposia on Non-Photorealistic Animation and Rendering (NPAR) at http://www.npar.org/.



(a) (b)

Figure 10.56 Non-photorealistic abstraction of photographs: (a) DeCarlo and Santella (2002) © 2002 ACM and (b) Farbman, Fattal, Lischinski *et al.* (2008) © 2008 ACM.

1994) and painterly rendering techniques (Litwinowicz 1997) use local color, intensity, and orientation estimates as an input to their procedural rendering algorithms. Techniques for stylizing and simplifying photographs and video (DeCarlo and Santella 2002; Winnemöller, Olsen, and Gooch 2006; Farbman, Fattal, Lischinski *et al.* 2008), as in Figure 10.56, use combinations of edge-preserving blurring (Section 3.3.1) and edge detection and enhancement (Section 4.2.3).

10.6 Additional reading

A good overview of computational photography can be found in the book by Raskar and Tumblin (2010), survey articles by Nayar (2006), Cohen and Szeliski (2006), Levoy (2006), Debevec (2006), and Hayes (2008), as well as two special journal issues edited by Bimber (2006) and Durand and Szeliski (2007). Notes from the courses on computational photography mentioned at the beginning of this chapter are another great source of material and references.[24]

The sub-field of high dynamic range imaging has its own book discussing research in this area (Reinhard, Ward, Pattanaik *et al.* 2005), as well as some books describing related photographic techniques (Freeman 2008; Gulbins and Gulbins 2009). Algorithms for calibrating the radiometric response function of a camera can be found in articles by Mann and Picard (1995), Debevec and Malik (1997), and Mitsunaga and Nayar (1999).

The subject of tone mapping is treated extensively in (Reinhard, Ward, Pattanaik *et al.* 2005). Representative papers from the large volume of literature on this topic include those by Tumblin and Rushmeier (1993), Larson, Rushmeier, and Piatko (1997), Pattanaik, Ferwerda, Fairchild *et al.* (1998), Tumblin and Turk (1999), Durand and Dorsey (2002), Fattal, Lischinski, and Werman (2002), Reinhard, Stark, Shirley *et al.* (2002), Lischinski, Farbman, Uyttendaele *et al.* (2006b), and Farbman, Fattal, Lischinski *et al.* (2008).

[24] MIT 6.815/6.865, http://stellar.mit.edu/S/course/6/sp08/6.815/materials.html, CMU 15-463, http://graphics.cs.cmu.edu/courses/15-463/2008_fall/, Stanford CS 448A, http://graphics.stanford.edu/courses/cs448a-08-spring/, and SIGGRAPH courses, http://web.media.mit.edu/~raskar/photo/.

The literature on super-resolution is quite extensive (Chaudhuri 2001; Park, Park, and Kang 2003; Capel and Zisserman 2003; Capel 2004; van Ouwerkerk 2006). The term super-resolution usually describes techniques for aligning and merging multiple images to produce higher-resolution composites (Keren, Peleg, and Brada 1988; Irani and Peleg 1991; Cheeseman, Kanefsky, Hanson *et al.* 1993; Mann and Picard 1994; Chiang and Boult 1996; Bascle, Blake, and Zisserman 1996; Capel and Zisserman 1998; Smelyanskiy, Cheeseman, Maluf *et al.* 2000; Capel and Zisserman 2000; Pickup, Capel, Roberts *et al.* 2009; Gulbins and Gulbins 2009). However, single-image super-resolution techniques have also been developed (Freeman, Jones, and Pasztor 2002; Baker and Kanade 2002; Fattal 2007).

A good survey on image matting is given by Wang and Cohen (2007a). Representative papers, which include extensive comparisons with previous work, include those by Chuang, Curless, Salesin *et al.* (2001), Wang and Cohen (2007b), Levin, Acha, and Lischinski (2008), Rhemann, Rother, Rav-Acha *et al.* (2008), and Rhemann, Rother, Wang *et al.* (2009).

The literature on texture synthesis and hole filling includes traditional approaches to texture synthesis, which try to match image statistics between source and destination images (Heeger and Bergen 1995; De Bonet 1997; Portilla and Simoncelli 2000), as well as newer approaches, which search for matching neighborhoods or patches inside the source sample (Efros and Leung 1999; Wei and Levoy 2000; Efros and Freeman 2001). In a similar vein, traditional approaches to hole filling involve the solution of local variational (smooth continuation) problems (Bertalmio, Sapiro, Caselles *et al.* 2000; Bertalmio, Vese, Sapiro *et al.* 2003; Telea 2004). More recent techniques use data-driven texture synthesis approaches (Drori, Cohen-Or, and Yeshurun 2003; Kwatra, Schödl, Essa *et al.* 2003; Criminisi, Pérez, and Toyama 2004; Sun, Yuan, Jia *et al.* 2004; Kwatra, Essa, Bobick *et al.* 2005; Wilczkowiak, Brostow, Tordoff *et al.* 2005; Komodakis and Tziritas 2007b; Wexler, Shechtman, and Irani 2007).

10.7 Exercises

Ex 10.1: Radiometric calibration Implement one of the multi-exposure radiometric calibration algorithms described in Section 10.2 (Debevec and Malik 1997; Mitsunaga and Nayar 1999; Reinhard, Ward, Pattanaik *et al.* 2005). This calibration will be useful in a number of different applications, such as stitching images or stereo matching with different exposures and shape from shading.

1. Take a series of bracketed images with your camera on a tripod. If your camera has an automatic exposure bracketing (AEB) mode, taking three images may be sufficient to calibrate most of your camera's dynamic range, especially if your scene has a lot of bright and dark regions. (Shooting outdoors or through a window on a sunny day is best.)

2. If your images are not taken on a tripod, first perform a global alignment (similarity transform).

3. Estimate the radiometric response function using one of the techniques cited above.

4. Estimate the high dynamic range radiance image by selecting or blending pixels from

different exposures (Debevec and Malik 1997; Mitsunaga and Nayar 1999; Eden, Uyttendaele, and Szeliski 2006).

5. Repeat your calibration experiments under different conditions, e.g., indoors under incandescent light, to get a sense for the range of color balancing effects that your camera imposes.

6. If your camera supports RAW and JPEG mode, calibrate both sets of images simultaneously and to each other (the radiance at each pixel will correspond). See if you can come up with a model for what your camera does, e.g., whether it treats color balance as a diagonal or full 3×3 matrix multiply, whether it uses non-linearities in addition to gamma, whether it sharpens the image while "developing" the JPEG image, etc.

7. Develop an interactive viewer to change the exposure of an image based on the average exposure of a region around the mouse. (One variant is to show the adjusted image inside a window around the mouse. Another is to adjust the complete image based on the mouse position.)

8. Implement a tone mapping operator (Exercise 10.5) and use this to map your radiance image to a displayable gamut.

Ex 10.2: Noise level function Determine your camera's noise level function using either multiple shots or by analyzing smooth regions.

1. Set up your camera on a tripod looking at a calibration target or a static scene with a good variation in input levels and colors. (Check your camera's histogram to ensure that all values are being sampled.)

2. Take repeated images of the same scene (ideally with a remote shutter release) and average them to compute the variance at each pixel. Discarding pixels near high gradients (which are affected by camera motion), plot for each color channel the standard deviation at each pixel as a function of its output value.

3. Fit a lower envelope to these measurements and use this as your noise level function. How much variation do you see in the noise as a function of input level? How much of this is significant, i.e., away from flat regions in your camera response function where you do not want to be sampling anyway?

4. (Optional) Using the same images, develop a technique that segments the image into near-constant regions (Liu, Szeliski, Kang *et al.* 2008). (This is easier if you are photographing a calibration chart.) Compute the deviations for each region from a *single* image and use them to estimate the NLF. How does this compare to the multi-image technique, and how stable are your estimates from image to image?

Ex 10.3: Vignetting Estimate the amount of vignetting in some of your lenses using one of the following three techniques (or devise one of your choosing):

1. Take an image of a large uniform intensity region (well-illuminated wall or blue sky—but be careful of brightness gradients) and fit a radial polynomial curve to estimate the vignetting.

2. Construct a center-weighted panorama and compare these pixel values to the input image values to estimate the vignetting function. Weight pixels in slowly varying regions more highly, as small misalignments will give large errors at high gradients. Optionally estimate the radiometric response function as well (Litvinov and Schechner 2005; Goldman 2011).

3. Analyze the radial gradients (especially in low-gradient regions) and fit the robust means of these gradients to the derivative of the vignetting function, as described by Zheng, Yu, Kang *et al.* (2008).

For the parametric form of your vignetting function, you can either use a simple radial function, e.g.,

$$f(r) = 1 + \alpha_1 r + \alpha_2 r^2 + \cdots \tag{10.42}$$

or one of the specialized equations developed by Kang and Weiss (2000) and Zheng, Lin, and Kang (2006).

In all of these cases, be sure that you are using linearized intensity measurements, by using either RAW images or images linearized through a radiometric response function, or at least images where the gamma curve has been removed.

(Optional) What happens if you forget to undo the gamma before fitting a (multiplicative) vignetting function?

Ex 10.4: Optical blur (PSF) estimation Compute the optical PSF either using a known target (Figure 10.7) or by detecting and fitting step edges (Section 10.1.4) (Joshi, Szeliski, and Kriegman 2008).

1. Detect strong edges to sub-pixel precision.

2. Fit a local profile to each oriented edge and fill these pixels into an ideal target image, either at image resolution or at a higher resolution (Figure 10.9c–d).

3. Use least squares (10.1) at valid pixels to estimate the PSF kernel K, either globally or in locally overlapping sub-regions of the image.

4. Visualize the recovered PSFs and use them to remove chromatic aberration or de-blur the image.

Ex 10.5: Tone mapping Implement one of the tone mapping algorithms discussed in Section 10.2.1 (Durand and Dorsey 2002; Fattal, Lischinski, and Werman 2002; Reinhard, Stark, Shirley *et al.* 2002; Lischinski, Farbman, Uyttendaele *et al.* 2006b) or any of the numerous additional algorithms discussed by Reinhard, Ward, Pattanaik *et al.* (2005) and http://stellar.mit.edu/S/course/6/sp08/6.815/materials.html.

(Optional) Compare your algorithm to local histogram equalization (Section 3.1.4).

Ex 10.6: Flash enhancement Develop an algorithm to combine flash and non-flash photographs to best effect. You can use ideas from Eisemann and Durand (2004) and Petschnigg, Agrawala, Hoppe *et al.* (2004) or anything else you think might work well.

Ex 10.7: Super-resolution Implement one or more super-resolution algorithms and compare their performance.

1. Take a set of photographs of the same scene using a hand-held camera (to ensure that there is some jitter between the photographs).
2. Determine the PSF for the images you are trying to super-resolve using one of the techniques in Exercise 10.4.
3. Alternatively, simulate a collection of lower-resolution images by taking a high-quality photograph (avoid those with compression artifacts) and applying your own pre-filter kernel and downsampling.
4. Estimate the relative motion between the images using a parametric translation and rotation motion estimation algorithm (Sections 6.1.3 or 8.2).
5. Implement a basic least squares super-resolution algorithm by minimizing the difference between the observed and downsampled images (10.27–10.28).
6. Add in a gradient image prior, either as another least squares term or as a robust term that can be minimized using iteratively reweighted least squares (Appendix A.3).
7. (Optional) Implement one of the example-based super-resolution techniques, where matching against a set of exemplar images is used either to infer higher-frequency information to be added to the reconstruction (Freeman, Jones, and Pasztor 2002) or higher-frequency gradients to be matched in the super-resolved image (Baker and Kanade 2002).
8. (Optional) Use local edge statistic information to improve the quality of the super-resolved image (Fattal 2007).

Ex 10.8: Image matting Develop an algorithm for pulling a foreground matte from natural images, as described in Section 10.4.

1. Make sure that the images you are taking are linearized (Exercise 10.1 and Section 10.1) and that your camera exposure is fixed (full manual mode), at least when taking multiple shots of the same scene.
2. To acquire ground truth data, place your object in front of a computer monitor and display a variety of solid background colors as well as some natural imagery.
3. Remove your object and re-display the same images to acquire known background colors.
4. Use triangulation matting (Smith and Blinn 1996) to estimate the ground truth opacities α and pre-multiplied foreground colors αF for your objects.
5. Implement one or more of the natural image matting algorithms described in Section 10.4 and compare your results to the ground truth values you computed. Alternatively, use the matting test images published on http://alphamatting.com/.
6. (Optional) Run your algorithms on other images taken with the same calibrated camera (or other images you find interesting).

Ex 10.9: Smoke and shadow matting Extract smoke or shadow mattes from one scene and insert them into another (Chuang, Agarwala, Curless *et al.* 2002; Chuang, Goldman, Curless *et al.* 2003).

1. Take a still or video sequence of images with and without some intermittent smoke and shadows. (Remember to linearize your images before proceeding with any computations.)

2. For each pixel, fit a line to the observed color values.

3. If performing smoke matting, robustly compute the intersection of these lines to obtain the smoke color estimate. Then, estimate the background color as the other extremum (unless you already took a smoke-free background image).

   If performing shadow matting, compute robust shadow (minimum) and lit (maximum) values for each pixel.

4. Extract the smoke or shadow mattes from each frame as the fraction between these two values (background and smoke or shadowed and lit).

5. Scan a new (destination) scene or modify the original background with an image editor.

6. Re-insert the smoke or shadow matte, along with any other foreground objects you may have extracted.

7. (Optional) Using a series of cast stick shadows, estimate the deformation field for the destination scene in order to correctly warp (drape) the shadows across the new geometry. (This is related to the shadow scanning technique developed by Bouguet and Perona (1999) and implemented in Exercise 12.2.)

8. (Optional) Chuang, Goldman, Curless *et al.* (2003) only demonstrated their technique for planar source geometries. Can you extend their technique to capture shadows acquired from an irregular source geometry?

9. (Optional) Can you change the direction of the shadow, i.e., simulate the effect of changing the light source direction?

Ex 10.10: Texture synthesis Implement one of the texture synthesis or hole filling algorithms presented in Section 10.5. Here is one possible procedure:

1. Implement the basic Efros and Leung (1999) algorithm, i.e., starting from the outside (for hole filling) or in raster order (for texture synthesis), search for a similar neighborhood in the source texture image, and copy that pixel.

2. Add in the Wei and Levoy (2000) extension of generating the pixels in a coarse-to-fine fashion, i.e., generate a lower-resolution synthetic texture (or filled image), and use this as a guide for matching regions in the finer resolution version.

3. Add in the Criminisi, Pérez, and Toyama (2004) idea of prioritizing pixels to be filled by some function of the local structure (gradient or orientation strength).

4. Extend any of the above algorithms by selecting sub-blocks in the source texture and using optimization to determine the seam between the new block and the existing image that it overlaps (Efros and Freeman 2001).

5. (Optional) Implement one of the isophote (smooth continuation) inpainting algorithms (Bertalmio, Sapiro, Caselles *et al.* 2000; Telea 2004).

6. (Optional) Add the ability to supply a target (reference) image (Efros and Freeman 2001) or to provide sample filtered or unfiltered (reference and rendered) images (Hertzmann, Jacobs, Oliver *et al.* 2001), see Section 10.5.2.

Ex 10.11: Colorization Implement the Levin, Lischinski, and Weiss (2004) colorization algorithm that is sketched out in Section 10.3.2 and Figure 10.37. Find some historic monochrome photographs and some modern color ones. Write an interactive tool that lets you "pick" colors from a modern photo and paint over the old one. Tune the algorithm parameters to give you good results. Are you pleased with the results? Can you think of ways to make them look more "antique", e.g., with softer (less saturated and edgy) colors?

Chapter 11

Stereo correspondence

11.1 Epipolar geometry . . . 471
11.1.1 Rectification . . . 472
11.1.2 Plane sweep . . . 474
11.2 Sparse correspondence . . . 475
11.2.1 3D curves and profiles . . . 476
11.3 Dense correspondence . . . 477
11.3.1 Similarity measures . . . 479
11.4 Local methods . . . 480
11.4.1 Sub-pixel estimation and uncertainty . . . 482
11.4.2 *Application*: Stereo-based head tracking . . . 483
11.5 Global optimization . . . 484
11.5.1 Dynamic programming . . . 485
11.5.2 Segmentation-based techniques . . . 487
11.5.3 *Application*: Z-keying and background replacement . . . 489
11.6 Multi-view stereo . . . 489
11.6.1 Volumetric and 3D surface reconstruction . . . 492
11.6.2 Shape from silhouettes . . . 497
11.7 Additional reading . . . 499
11.8 Exercises . . . 500

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_11, © Springer-Verlag London Limited 2011

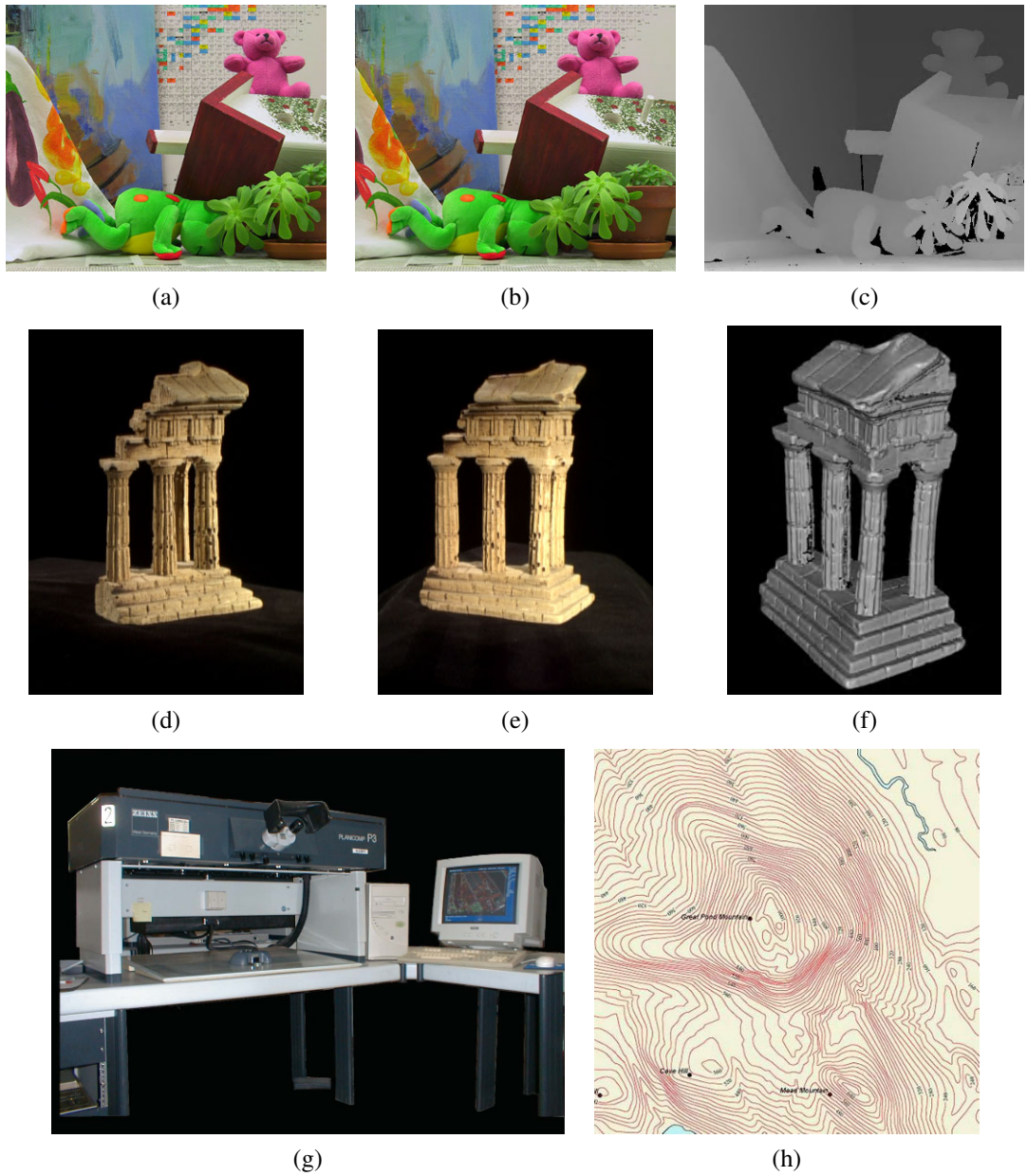


Figure 11.1 Stereo reconstruction techniques can convert (a–b) a pair of images into (c) a depth map (http://vision.middlebury.edu/stereo/data/scenes2003/) or (d–e) a sequence of images into (f) a 3D model (http://vision.middlebury.edu/mview/data/). (g) An analytical stereo plotter, courtesy of Kenney Aerial Mapping, Inc., can generate (h) contour plots.

Stereo matching is the process of taking two or more images and estimating a 3D model of the scene by finding matching pixels in the images and converting their 2D positions into 3D depths. In Chapters 6–7, we described techniques for recovering camera positions and building sparse 3D models of scenes or objects. In this chapter, we address the question of how to build a more complete 3D model, e.g., a sparse or dense *depth map* that assigns relative depths to pixels in the input images. We also look at the topic of *multi-view stereo* algorithms that produce complete 3D volumetric or surface-based object models.

Why are people interested in stereo matching? From the earliest inquiries into visual perception, it was known that we perceive depth based on the differences in appearance between the left and right eye.[1] As a simple experiment, hold your finger vertically in front of your eyes and close each eye alternately. You will notice that the finger jumps left and right relative to the background of the scene. The same phenomenon is visible in the image pair shown in Figure 11.1a–b, in which the foreground objects shift left and right relative to the background.

As we will shortly see, under simple imaging configurations (both eyes or cameras looking straight ahead), the amount of horizontal motion or *disparity* is inversely proportional to the distance from the observer. While the basic physics and geometry relating visual disparity to scene structure are well understood (Section 11.1), automatically measuring this disparity by establishing dense and accurate inter-image *correspondences* is a challenging task.

The earliest stereo matching algorithms were developed in the field of *photogrammetry* for automatically constructing topographic elevation maps from overlapping aerial images. Prior to this, operators would use photogrammetric stereo plotters, which displayed shifted versions of such images to each eye and allowed the operator to float a dot cursor around constant elevation contours (Figure 11.1g). The development of fully automated stereo matching algorithms was a major advance in this field, enabling much more rapid and less expensive processing of aerial imagery (Hannah 1974; Hsieh, McKeown, and Perlant 1992).

In computer vision, the topic of stereo matching has been one of the most widely studied and fundamental problems (Marr and Poggio 1976; Barnard and Fischler 1982; Dhond and Aggarwal 1989; Scharstein and Szeliski 2002; Brown, Burschka, and Hager 2003; Seitz, Curless, Diebel *et al.* 2006), and continues to be one of the most active research areas. While photogrammetric matching concentrated mainly on aerial imagery, computer vision applications include modeling the human visual system (Marr 1982), robotic navigation and manipulation (Moravec 1983; Konolige 1997; Thrun, Montemerlo, Dahlkamp *et al.* 2006), as well as view interpolation and image-based rendering (Figure 11.2a–d), 3D model building (Figure 11.2e–f and h–j), and mixing live action with computer-generated imagery (Figure 11.2g).

In this chapter, we describe the fundamental principles behind stereo matching, following the general taxonomy proposed by Scharstein and Szeliski (2002). We begin in Section 11.1 with a review of the *geometry* of stereo image matching, i.e., how to compute for a given pixel in one image the range of possible locations the pixel might appear at in the other image, i.e., its *epipolar line*. We describe how to pre-warp images so that corresponding epipolar lines are coincident (*rectification*). We also describe a general resampling algorithm called *plane sweep* that can be used to perform multi-image stereo matching with arbitrary camera configurations.

[1] The word *stereo* comes from the Greek for *solid*; stereo vision is how we perceive solid shape (Koenderink 1990).

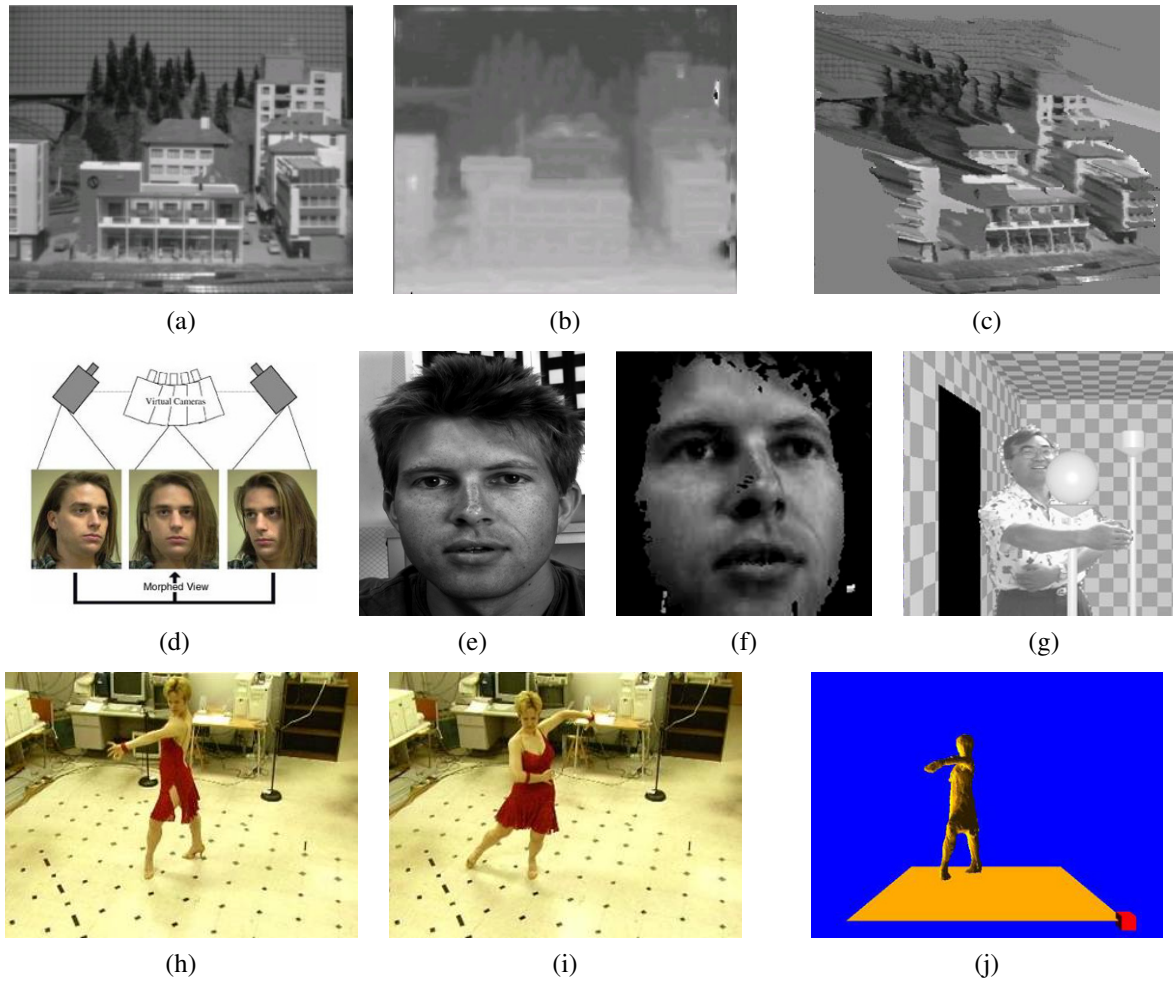


Figure 11.2 Applications of stereo vision: (a) input image, (b) computed depth map, and (c) new view generation from multi-view stereo (Matthies, Kanade, and Szeliski 1989) © 1989 Springer; (d) view morphing between two images (Seitz and Dyer 1996) © 1996 ACM; (e–f) 3D face modeling (images courtesy of Frédéric Devernay); (g) *z-keying* live and computer-generated imagery (Kanade, Yoshida, Oda *et al.* 1996) © 1996 IEEE; (h–j) building 3D surface models from multiple video streams in Virtualized Reality (Kanade, Rander, and Narayanan 1997).

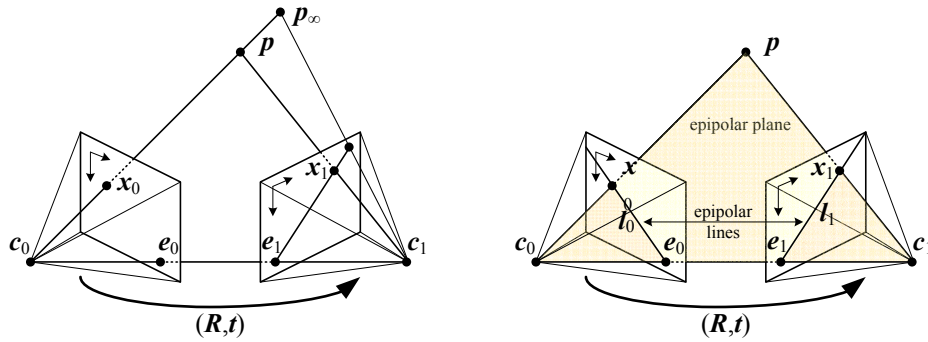


Figure 11.3 Epipolar geometry: (a) epipolar line segment corresponding to one ray; (b) corresponding set of epipolar lines and their epipolar plane.

Next, we briefly survey techniques for the *sparse* stereo matching of interest points and edge-like features (Section 11.2). We then turn to the main topic of this chapter, namely the estimation of a *dense* set of pixel-wise correspondences in the form of a *disparity map* (Figure 11.1c). This involves first selecting a pixel matching criterion (Section 11.3) and then using either local area-based aggregation (Section 11.4) or global optimization (Section 11.5) to help disambiguate potential matches. In Section 11.6, we discuss *multi-view stereo* methods that aim to reconstruct a complete 3D model instead of just a single disparity image (Figure 11.1d–f).

11.1 Epipolar geometry

Given a pixel in one image, how can we compute its correspondence in the other image? In Chapter 8, we saw that a variety of search techniques can be used to match pixels based on their local appearance as well as the motions of neighboring pixels. In the case of stereo matching, however, we have some additional information available, namely the positions and calibration data for the cameras that took the pictures of the same static scene (Section 7.2).

How can we exploit this information to reduce the number of potential correspondences, and hence both speed up the matching and increase its reliability? Figure 11.3a shows how a pixel in one image $\boldsymbol{x}_0$ projects to an *epipolar line segment* in the other image. The segment is bounded at one end by the projection of the original viewing ray at infinity $\boldsymbol{p}_\infty$ and at the other end by the projection of the original camera center $\boldsymbol{c}_0$ into the second camera, which is known as the *epipole* $\boldsymbol{e}_1$. If we project the epipolar line in the second image back into the first, we get another line (segment), this time bounded by the other corresponding epipole $\boldsymbol{e}_0$. Extending both line segments to infinity, we get a pair of corresponding *epipolar lines* (Figure 11.3b), which are the intersection of the two image planes with the *epipolar plane* that passes through both camera centers $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$ as well as the point of interest $\boldsymbol{p}$ (Faugeras and Luong 2001; Hartley and Zisserman 2004).

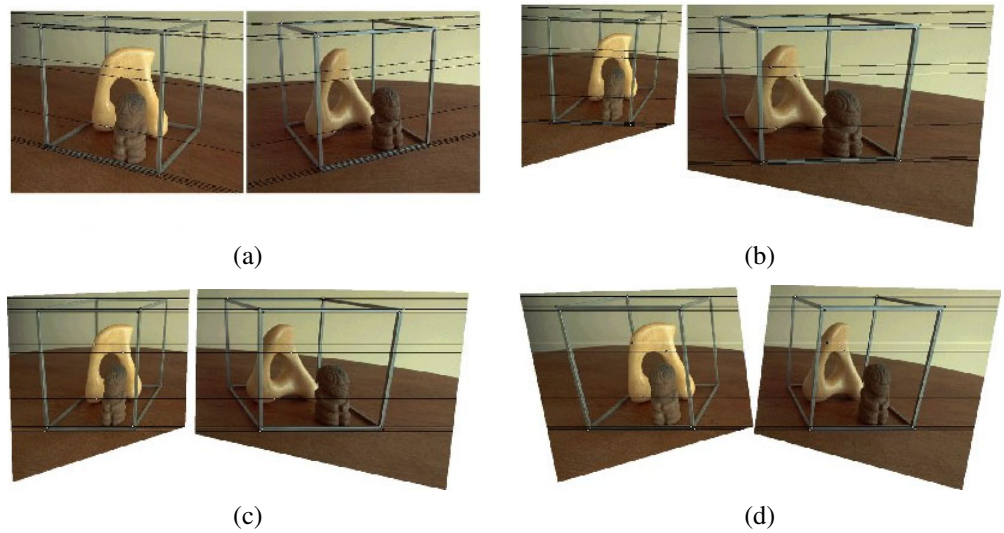


Figure 11.4 The multi-stage stereo rectification algorithm of Loop and Zhang (1999) © 1999 IEEE. (a) Original image pair overlaid with several epipolar lines; (b) images transformed so that epipolar lines are parallel; (c) images rectified so that epipolar lines are horizontal and in vertial correspondence; (d) final rectification that minimizes horizontal distortions.

11.1.1 Rectification

As we saw in Section 7.2, the epipolar geometry for a pair of cameras is implicit in the relative pose and calibrations of the cameras, and can easily be computed from seven or more point matches using the fundamental matrix (or five or more points for the calibrated essential matrix) (Zhang 1998a,b; Faugeras and Luong 2001; Hartley and Zisserman 2004). Once this geometry has been computed, we can use the epipolar line corresponding to a pixel in one image to constrain the search for corresponding pixels in the other image. One way to do this is to use a general correspondence algorithm, such as optical flow (Section 8.4), but to only consider locations along the epipolar line (or to project any flow vectors that fall off back onto the line).

A more efficient algorithm can be obtained by first *rectifying* (i.e, warping) the input images so that corresponding horizontal scanlines are epipolar lines (Loop and Zhang 1999; Faugeras and Luong 2001; Hartley and Zisserman 2004).[2] Afterwards, it is possible to match horizontal scanlines independently or to shift images horizontally while computing matching scores (Figure 11.4).

A simple way to rectify the two images is to first rotate both cameras so that they are looking perpendicular to the line joining the camera centers $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$. Since there is a degree of freedom in the *tilt*, the smallest rotations that achieve this should be used. Next, to determine the desired twist around the optical axes, make the *up vector* (the camera y axis) perpendicular to the camera center line. This ensures that corresponding epipolar lines are

[2] This makes most sense if the cameras are next to each other, although by rotating the cameras, rectification can be performed on any pair that is not *verged* too much or has too much of a scale change. In those latter cases, using plane sweep (below) or hypothesizing small planar patch locations in 3D (Goesele, Snavely, Curless *et al.* 2007) may be preferable.

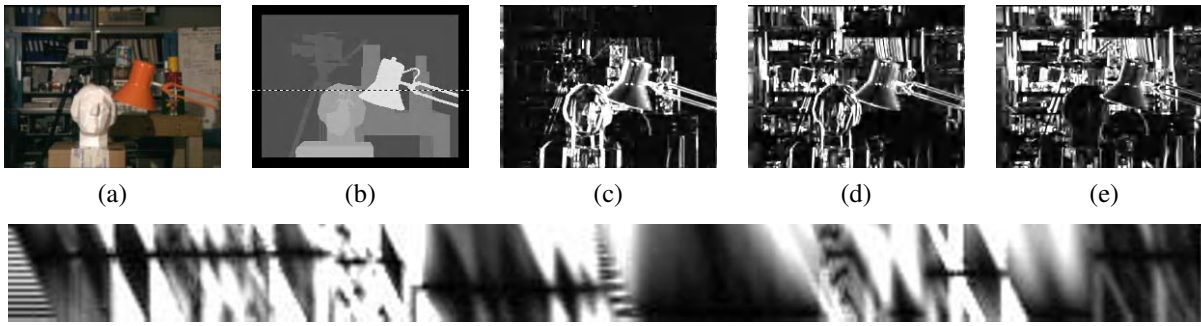


Figure 11.5 Slices through a typical disparity space image (DSI) (Scharstein and Szeliski 2002) © 2002 Springer: (a) original color image; (b) ground truth disparities; (c–e) three (x, y) slices for $d = 10, 16, 21$; (f) an (x, d) slice for $y = 151$ (the dashed line in (b)). Various dark (matching) regions are visible in (c–e), e.g., the bookshelves, table and cans, and head statue, and three disparity levels can be seen as horizontal lines in (f). The dark bands in the DSIs indicate regions that match at this disparity. (Smaller dark regions are often the result of textureless regions.) Additional examples of DSIs are discussed by Bobick and Intille (1999).

horizontal and that the disparity for points at infinity is 0. Finally, re-scale the images, if necessary, to account for different focal lengths, magnifying the smaller image to avoid aliasing. (The full details of this procedure can be found in Fusiello, Trucco, and Verri (2000) and Exercise 11.1.) Note that in general, it is not possible to rectify an arbitrary collection of images simultaneously unless their optical centers are collinear, although rotating the cameras so that they all point in the same direction reduces the inter-camera pixel movements to scalings and translations.

The resulting *standard rectified geometry* is employed in a lot of stereo camera setups and stereo algorithms, and leads to a very simple inverse relationship between 3D depths Z and disparities d,

$$d = f\frac{B}{Z}, \tag{11.1}$$

where f is the focal length (measured in pixels), B is the baseline, and

$$x' = x + d(x, y), \quad y' = y \tag{11.2}$$

describes the relationship between corresponding pixel coordinates in the left and right images (Bolles, Baker, and Marimont 1987; Okutomi and Kanade 1993; Scharstein and Szeliski 2002).[3] The task of extracting depth from a set of images then becomes one of estimating the *disparity map* $d(x, y)$.

After rectification, we can easily compare the similarity of pixels at corresponding locations (x, y) and $(x', y') = (x + d, y)$ and store them in a *disparity space image* (DSI) $C(x, y, d)$ for further processing (Figure 11.5). The concept of the disparity space (x, y, d) dates back to early work in stereo matching (Marr and Poggio 1976), while the concept of a disparity space image (volume) is generally associated with Yang, Yuille, and Lu (1993) and Intille and Bobick (1994).

[3] The term *disparity* was first introduced in the human vision literature to describe the difference in location of corresponding features seen by the left and right eyes (Marr 1982). Horizontal disparity is the most commonly studied phenomenon, but vertical disparity is possible if the eyes are verged.

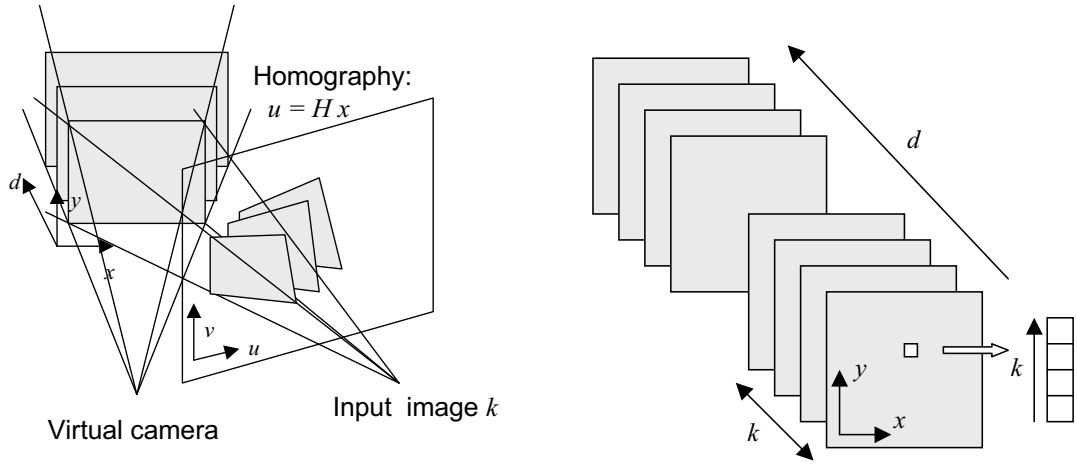


Figure 11.6 Sweeping a set of planes through a scene (Szeliski and Golland 1999) © 1999 Springer: (a) The set of planes seen from a virtual camera induces a set of homographies in any other source (input) camera image. (b) The warped images from all the other cameras can be stacked into a generalized disparity space volume $\tilde{I}(x, y, d, k)$ indexed by pixel location (x, y), disparity d, and camera k.

11.1.2 Plane sweep

An alternative to pre-rectifying the images before matching is to sweep a set of planes through the scene and to measure the *photoconsistency* of different images as they are re-projected onto these planes (Figure 11.6). This process is commonly known as the *plane sweep* algorithm (Collins 1996; Szeliski and Golland 1999; Saito and Kanade 1999).

As we saw in Section 2.1.5, where we introduced projective depth (also known as *plane plus parallax* (Kumar, Anandan, and Hanna 1994; Sawhney 1994; Szeliski and Coughlan 1997)), the last row of a full-rank 4×4 projection matrix $\tilde{\boldsymbol{P}}$ can be set to an arbitrary plane equation $\boldsymbol{p}_3 = s_3[\hat{\boldsymbol{n}}_0|c_0]$. The resulting four-dimensional projective transform (*collineation*) (2.68) maps 3D world points $\boldsymbol{p} = (X, Y, Z, 1)$ into screen coordinates $\boldsymbol{x}_s = (x_s, y_s, 1, d)$, where the *projective depth* (or *parallax*) d (2.66) is 0 on the reference plane (Figure 2.11).

Sweeping d through a series of disparity hypotheses, as shown in Figure 11.6a, corresponds to mapping each input image into the *virtual camera* $\tilde{\boldsymbol{P}}$ defining the disparity space through a series of homographies (2.68–2.71),

$$\tilde{\boldsymbol{x}}_k \sim \tilde{\boldsymbol{P}}_k \tilde{\boldsymbol{P}}^{-1} \boldsymbol{x}_s = \tilde{\boldsymbol{H}}_k \tilde{\boldsymbol{x}} + \boldsymbol{t}_k d = (\tilde{\boldsymbol{H}}_k + \boldsymbol{t}_k [0\ 0\ d]) \tilde{\boldsymbol{x}}, \tag{11.3}$$

as shown in Figure 2.12b, where $\tilde{\boldsymbol{x}}_k$ and $\tilde{\boldsymbol{x}}$ are the homogeneous pixel coordinates in the source and virtual (reference) images (Szeliski and Golland 1999). The members of the family of homographies $\tilde{\boldsymbol{H}}_k(d) = \tilde{\boldsymbol{H}}_k + \boldsymbol{t}_k[0\ 0\ d]$, which are parametererized by the addition of a rank-1 matrix, are related to each other through a *planar homology* (Hartley and Zisserman 2004, A5.2).

The choice of virtual camera and parameterization is application dependent and is what gives this framework a lot of its flexibility. In many applications, one of the input cameras (the *reference* camera) is used, thus computing a depth map that is registered with one of the

input images and which can later be used for image-based rendering (Sections 13.1 and 13.2). In other applications, such as view interpolation for gaze correction in video-conferencing (Section 11.4.2) (Ott, Lewis, and Cox 1993; Criminisi, Shotton, Blake *et al.* 2003), a camera centrally located between the two input cameras is preferable, since it provides the needed per-pixel disparities to hallucinate the virtual middle image.

The choice of disparity sampling, i.e., the setting of the zero parallax plane and the scaling of integer disparities, is also application dependent, and is usually set to bracket the range of interest, i.e., the *working volume*, while scaling disparities to sample the image in pixel (or sub-pixel) shifts. For example, when using stereo vision for obstacle avoidance in robot navigation, it is most convenient to set up disparity to measure per-pixel elevation above the ground (Ivanchenko, Shen, and Coughlan 2009).

As each input image is warped onto the current planes parameterized by disparity d, it can be stacked into a *generalized disparity space image* $\tilde{I}(x, y, d, k)$ for further processing (Figure 11.6b) (Szeliski and Golland 1999). In most stereo algorithms, the photoconsistency (e.g., sum of squared or robust differences) with respect to the reference image I_r is calculated and stored in the DSI

$$C(x, y, d) = \sum_k \rho(\tilde{I}(x, y, d, k) - I_r(x, y)). \tag{11.4}$$

However, it is also possible to compute alternative statistics such as robust variance, focus, or entropy (Section 11.3.1) (Vaish, Szeliski, Zitnick *et al.* 2006) or to use this representation to reason about occlusions (Szeliski and Golland 1999; Kang and Szeliski 2004). The generalized DSI will come in particularly handy when we come back to the topic of multi-view stereo in Section 11.6.

Of course, planes are not the only surfaces that can be used to define a 3D sweep through the space of interest. Cylindrical surfaces, especially when coupled with panoramic photography (Chapter 9), are often used (Ishiguro, Yamamoto, and Tsuji 1992; Kang and Szeliski 1997; Shum and Szeliski 1999; Li, Shum, Tang *et al.* 2004; Zheng, Kang, Cohen *et al.* 2007). It is also possible to define other manifold topologies, e.g., ones where the camera rotates around a fixed axis (Seitz 2001).

Once the DSI has been computed, the next step in most stereo correspondence algorithms is to produce a univalued function in disparity space $d(x, y)$ that best describes the shape of the surfaces in the scene. This can be viewed as finding a surface embedded in the disparity space image that has some optimality property, such as lowest cost and best (piecewise) smoothness (Yang, Yuille, and Lu 1993). Figure 11.5 shows examples of slices through a typical DSI. More figures of this kind can be found in the paper by Bobick and Intille (1999).

11.2 Sparse correspondence

Early stereo matching algorithms were *feature-based*, i.e., they first extracted a set of potentially matchable image locations, using either interest operators or edge detectors, and then searched for corresponding locations in other images using a patch-based metric (Hannah 1974; Marr and Poggio 1979; Mayhew and Frisby 1980; Baker and Binford 1981; Arnold 1983; Grimson 1985; Ohta and Kanade 1985; Bolles, Baker, and Marimont 1987; Matthies, Kanade, and Szeliski 1989; Hsieh, McKeown, and Perlant 1992; Bolles, Baker, and Hannah

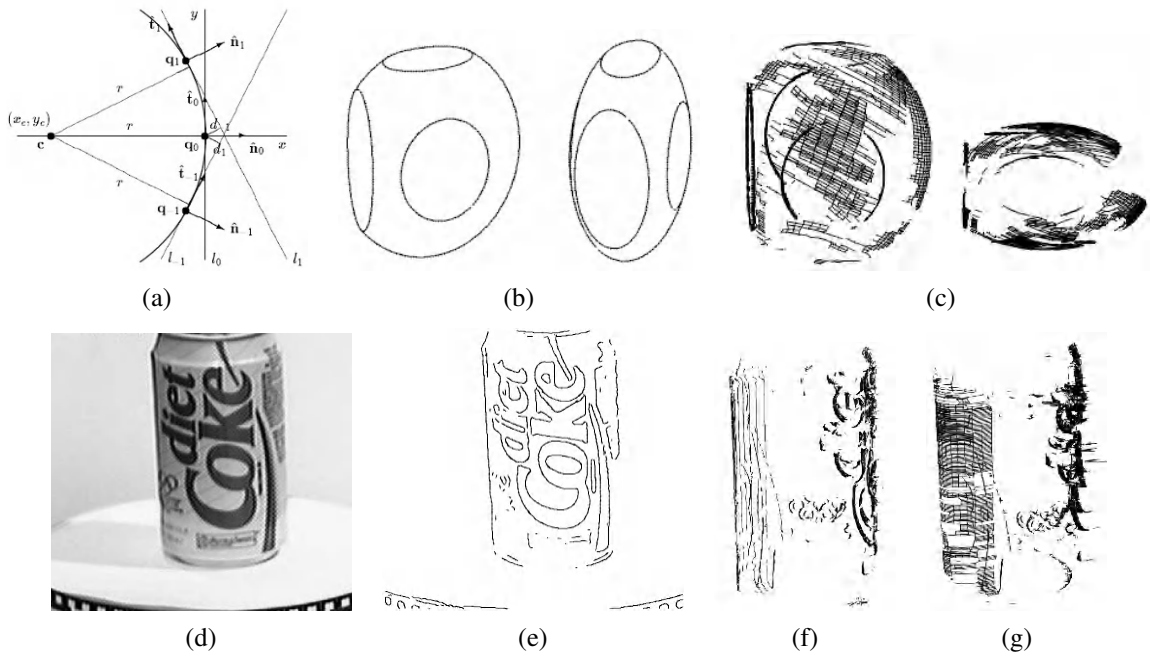


Figure 11.7 Surface reconstruction from occluding contours (Szeliski and Weiss 1998) © 2002 Springer: (a) circular arc fitting in the epipolar plane; (b) synthetic example of an ellipsoid with a truncated side and elliptic surface markings; (c) partially reconstructed surface mesh seen from an oblique and top-down view; (d) real-world image sequence of a soda can on a turntable; (e) extracted edges; (f) partially reconstructed profile curves; (g) partially reconstructed surface mesh. (Partial reconstructions are shown so as not to clutter the images.)

1993). This limitation to sparse correspondences was partially due to computational resource limitations, but was also driven by a desire to limit the answers produced by stereo algorithms to matches with high certainty. In some applications, there was also a desire to match scenes with potentially very different illuminations, where edges might be the only stable features (Collins 1996). Such sparse 3D reconstructions could later be interpolated using surface fitting algorithms such as those discussed in Sections 3.7.1 and 12.3.1.

More recent work in this area has focused on first extracting highly reliable features and then using these as *seeds* to grow additional matches (Zhang and Shan 2000; Lhuillier and Quan 2002). Similar approaches have also been extended to wide baseline multi-view stereo problems and combined with 3D surface reconstruction (Lhuillier and Quan 2005; Strecha, Tuytelaars, and Van Gool 2003; Goesele, Snavely, Curless *et al.* 2007) or free-space reasoning (Taylor 2003), as described in more detail in Section 11.6.

11.2.1 3D curves and profiles

Another example of sparse correspondence is the matching of *profile curves* (or *occluding contours*), which occur at the boundaries of objects (Figure 11.7) and at interior self occlusions, where the surface curves away from the camera viewpoint.

The difficulty in matching profile curves is that in general, the locations of profile curves vary as a function of camera viewpoint. Therefore, matching curves directly in two images

and then triangulating these matches can lead to erroneous shape measurements. Fortunately, if three or more closely spaced frames are available, it is possible to fit a local circular arc to the locations of corresponding edgels (Figure 11.7a) and therefore obtain semi-dense curved surface meshes directly from the matches (Figures 11.7c and g). Another advantage of matching such curves is that they can be used to reconstruct surface shape for untextured surfaces, so long as there is a visible difference between foreground and background colors.

Over the years, a number of different techniques have been developed for reconstructing surface shape from profile curves (Giblin and Weiss 1987; Cipolla and Blake 1992; Vaillant and Faugeras 1992; Zheng 1994; Boyer and Berger 1997; Szeliski and Weiss 1998). Cipolla and Giblin (2000) describe many of these techniques, as well as related topics such as inferring camera motion from profile curve sequences. Below, we summarize the approach developed by Szeliski and Weiss (1998), which assumes a discrete set of images, rather than formulating the problem in a continuous differential framework.

Let us assume that the camera is moving smoothly enough that the local epipolar geometry varies slowly, i.e., the epipolar planes induced by the successive camera centers and an edgel under consideration are nearly co-planar. The first step in the processing pipeline is to extract and link edges in each of the input images (Figures 11.7b and e). Next, edgels in successive images are matched using pairwise epipolar geometry, proximity and (optionally) appearance. This provides a linked set of edges in the spatio-temporal volume, which is sometimes called the *weaving wall* (Baker 1989).

To reconstruct the 3D location of an individual edgel, along with its local in-plane normal and curvature, we project the viewing rays corresponding to its neighbors onto the instantaneous epipolar plane defined by the camera center, the viewing ray, and the camera velocity, as shown in Figure 11.7a. We then fit an *osculating circle* to the projected lines, parameterizing the circle by its centerpoint $\boldsymbol{c} = (x_c, y_c)$ and radius r,

$$c_i x_c + s_i y_c + r = d_i, \tag{11.5}$$

where $c_i = \hat{\boldsymbol{t}}_i \cdot \hat{\boldsymbol{t}}_0$ and $s_i = -\hat{\boldsymbol{t}}_i \cdot \hat{\boldsymbol{n}}_0$ are the cosine and sine of the angle between viewing ray i and the central viewing ray 0, and $d_i = (\boldsymbol{q}_i - \boldsymbol{q}_0) \cdot \hat{\boldsymbol{n}}_0$ is the perpendicular distance between viewing ray i and the local origin $\boldsymbol{q}_0$, which is a point chosen on the central viewing ray close to the line intersections (Szeliski and Weiss 1998). The resulting set of linear equations can be solved using least squares, and the quality of the solution (residual error) can be used to check for erroneous correspondences.

The resulting set of 3D points, along with their spatial (in-image) and temporal (between-image) neighbors, form a 3D surface mesh with local normal and curvature estimates (Figures 11.7c and g). Note that whenever a curve is due to a surface marking or a sharp crease edge, rather than a smooth surface profile curve, this shows up as a 0 or small radius of curvature. Such curves result in isolated 3D space curves, rather than elements of smooth surface meshes, but can still be incorporated into the 3D surface model during a later stage of surface interpolation (Section 12.3.1).

11.3 Dense correspondence

While sparse matching algorithms are still occasionally used, most stereo matching algorithms today focus on dense correspondence, since this is required for applications such as

image-based rendering or modeling. This problem is more challenging than sparse correspondence, since inferring depth values in textureless regions requires a certain amount of guesswork. (Think of a solid colored background seen through a picket fence. What depth should it be?)

In this section, we review the taxonomy and categorization scheme for dense correspondence algorithms first proposed by Scharstein and Szeliski (2002). The taxonomy consists of a set of algorithmic "building blocks" from which a large set of algorithms can be constructed. It is based on the observation that stereo algorithms generally perform some subset of the following four steps:

1. matching cost computation;
2. cost (support) aggregation;
3. disparity computation and optimization; and
4. disparity refinement.

For example, *local* (window-based) algorithms (Section 11.4), where the disparity computation at a given point depends only on intensity values within a finite window, usually make implicit smoothness assumptions by aggregating support. Some of these algorithms can cleanly be broken down into steps 1, 2, 3. For example, the traditional sum-of-squared-differences (SSD) algorithm can be described as:

1. The matching cost is the squared difference of intensity values at a given disparity.
2. Aggregation is done by summing the matching cost over square windows with constant disparity.
3. Disparities are computed by selecting the minimal (winning) aggregated value at each pixel.

Some local algorithms, however, combine steps 1 and 2 and use a matching cost that is based on a support region, e.g. normalized cross-correlation (Hannah 1974; Bolles, Baker, and Hannah 1993) and the rank transform (Zabih and Woodfill 1994) and other ordinal measures (Bhat and Nayar 1998). (This can also be viewed as a preprocessing step; see (Section 11.3.1).)

Global algorithms, on the other hand, make explicit smoothness assumptions and then solve a a global optimization problem (Section 11.5). Such algorithms typically do not perform an aggregation step, but rather seek a disparity assignment (step 3) that minimizes a global cost function that consists of data (step 1) terms and smoothness terms. The main distinctions among these algorithms is the minimization procedure used, e.g., simulated annealing (Marroquin, Mitter, and Poggio 1987; Barnard 1989), probabilistic (mean-field) diffusion (Scharstein and Szeliski 1998), expectation maximization (EM) (Birchfield, Natarajan, and Tomasi 2007), graph cuts (Boykov, Veksler, and Zabih 2001), or loopy belief propagation (Sun, Zheng, and Shum 2003), to name just a few.

In between these two broad classes are certain iterative algorithms that do not explicitly specify a global function to be minimized, but whose behavior mimics closely that of iterative optimization algorithms (Marr and Poggio 1976; Zitnick and Kanade 2000). Hierarchical (coarse-to-fine) algorithms resemble such iterative algorithms, but typically operate on an

image pyramid where results from coarser levels are used to constrain a more local search at finer levels (Witkin, Terzopoulos, and Kass 1987; Quam 1984; Bergen, Anandan, Hanna *et al.* 1992).

11.3.1 Similarity measures

The first component of any dense stereo matching algorithm is a similarity measure that compares pixel values in order to determine how likely they are to be in correspondence. In this section, we briefly review the similarity measures introduced in Section 8.1 and mention a few others that have been developed specifically for stereo matching (Scharstein and Szeliski 2002; Hirschmüller and Scharstein 2009).

The most common pixel-based matching costs include sums of *squared intensity differences* (SSD) (Hannah 1974) and *absolute intensity differences* (SAD) (Kanade 1994). In the video processing community, these matching criteria are referred to as the *mean-squared error* (MSE) and *mean absolute difference* (MAD) measures; the term *displaced frame difference* is also often used (Tekalp 1995).

More recently, robust measures (8.2), including truncated quadratics and contaminated Gaussians, have been proposed (Black and Anandan 1996; Black and Rangarajan 1996; Scharstein and Szeliski 1998). These measures are useful because they limit the influence of mismatches during aggregation. Vaish, Szeliski, Zitnick *et al.* (2006) compare a number of such robust measures, including a new one based on the entropy of the pixel values at a particular disparity hypothesis (Zitnick, Kang, Uyttendaele *et al.* 2004), which is particularly useful in multi-view stereo.

Other traditional matching costs include normalized cross-correlation (8.11) (Hannah 1974; Bolles, Baker, and Hannah 1993; Evangelidis and Psarakis 2008), which behaves similarly to sum-of-squared-differences (SSD), and binary matching costs (i.e., match or no match) (Marr and Poggio 1976), based on binary features such as edges (Baker and Binford 1981; Grimson 1985) or the sign of the Laplacian (Nishihara 1984). Because of their poor discriminability, simple binary matching costs are no longer used in dense stereo matching.

Some costs are insensitive to differences in camera gain or bias, for example gradient-based measures (Seitz 1989; Scharstein 1994), phase and filter-bank responses (Marr and Poggio 1979; Kass 1988; Jenkin, Jepson, and Tsotsos 1991; Jones and Malik 1992), filters that remove regular or robust (bilaterally filtered) means (Ansar, Castano, and Matthies 2004; Hirschmüller and Scharstein 2009), dense feature descriptor (Tola, Lepetit, and Fua 2010), and non-parametric measures such as rank and census transforms (Zabih and Woodfill 1994), ordinal measures (Bhat and Nayar 1998), or entropy (Zitnick, Kang, Uyttendaele *et al.* 2004; Zitnick and Kang 2007). The census transform, which converts each pixel inside a moving window into a bit vector representing which neighbors are above or below the central pixel, was found by Hirschmüller and Scharstein (2009) to be quite robust against large-scale, non-stationary exposure and illumination changes.

It is also possible to correct for differing global camera characteristics by performing a preprocessing or iterative refinement step that estimates inter-image bias–gain variations using global regression (Gennert 1988), histogram equalization (Cox, Roy, and Hingorani 1995), or mutual information (Kim, Kolmogorov, and Zabih 2003; Hirschmüller 2008). Local, smoothly varying compensation fields have also been proposed (Strecha, Tuytelaars, and Van Gool 2003; Zhang, McMillan, and Yu 2006).

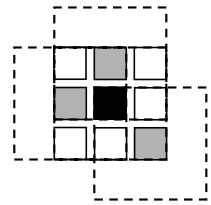


Figure 11.8 Shiftable window (Scharstein and Szeliski 2002) © 2002 Springer. The effect of trying all 3×3 shifted windows around the black pixel is the same as taking the minimum matching score across all *centered* (non-shifted) windows in the same neighborhood. (For clarity, only three of the neighboring shifted windows are shown here.)

In order to compensate for sampling issues, i.e., dramatically different pixel values in high-frequency areas, Birchfield and Tomasi (1998) proposed a matching cost that is less sensitive to shifts in image sampling. Rather than just comparing pixel values shifted by integral amounts (which may miss a valid match), they compare each pixel in the reference image against a linearly interpolated function of the other image. More detailed studies of these and additional matching costs are explored in (Szeliski and Scharstein 2004; Hirschmüller and Scharstein 2009). In particular, if you expect there to be significant exposure or appearance variation between images that you are matching, some of the more robust measures that performed well in the evaluation by Hirschmüller and Scharstein (2009), such as the census transform (Zabih and Woodfill 1994), ordinal measures (Bhat and Nayar 1998), bilateral subtraction (Ansar, Castano, and Matthies 2004), or hierarchical mutual information (Hirschmüller 2008), should be used.

11.4 Local methods

Local and window-based methods aggregate the matching cost by summing or averaging over a *support region* in the DSI $C(x, y, d)$.[4] A support region can be either two-dimensional at a fixed disparity (favoring fronto-parallel surfaces), or three-dimensional in x-y-d space (supporting slanted surfaces). Two-dimensional evidence aggregation has been implemented using square windows or Gaussian convolution (traditional), multiple windows anchored at different points, i.e., shiftable windows (Arnold 1983; Fusiello, Roberto, and Trucco 1997; Bobick and Intille 1999), windows with adaptive sizes (Okutomi and Kanade 1992; Kanade and Okutomi 1994; Kang, Szeliski, and Chai 2001; Veksler 2001, 2003), windows based on connected components of constant disparity (Boykov, Veksler, and Zabih 1998), or the results of color-based segmentation (Yoon and Kweon 2006; Tombari, Mattoccia, Di Stefano *et al.* 2008). Three-dimensional support functions that have been proposed include limited disparity difference (Grimson 1985), limited disparity gradient (Pollard, Mayhew, and Frisby 1985), Prazdny's coherence principle (Prazdny 1985), and the more recent work (which includes visibility and occlusion reasoning) by Zitnick and Kanade (2000).

[4] For two recent surveys and comparisons of such techniques, please see the work of Gong, Yang, Wang *et al.* (2007) and Tombari, Mattoccia, Di Stefano *et al.* (2008).



(a) (b) (c) (d)

Figure 11.9 Aggregation window sizes and weights adapted to image content (Tombari, Mattoccia, Di Stefano *et al.* 2008) © 2008 IEEE: (a) original image with selected evaluation points; (b) variable windows (Veksler 2003); (c) adaptive weights (Yoon and Kweon 2006); (d) segmentation-based (Tombari, Mattoccia, and Di Stefano 2007). Notice how the adaptive weights and segmentation-based techniques adapt their support to similarly colored pixels.

Aggregation with a fixed support region can be performed using 2D or 3D convolution,

$$C(x, y, d) = w(x, y, d) * C_0(x, y, d), \tag{11.6}$$

or, in the case of rectangular windows, using efficient moving average box-filters (Section 3.2.2) (Kanade, Yoshida, Oda *et al.* 1996; Kimura, Shinbo, Yamaguchi *et al.* 1999). Shiftable windows can also be implemented efficiently using a separable sliding min-filter (Figure 11.8) (Scharstein and Szeliski 2002, Section 4.2). Selecting among windows of different shapes and sizes can be performed more efficiently by first computing a *summed area table* (Section 3.2.3, 3.30–3.32) (Veksler 2003). Selecting the right window is important, since windows must be large enough to contain sufficient texture and yet small enough so that they do not straddle depth discontinuities (Figure 11.9). An alternative method for aggregation is *iterative diffusion*, i.e., repeatedly adding to each pixel's cost the weighted values of its neighboring pixels' costs (Szeliski and Hinton 1985; Shah 1993; Scharstein and Szeliski 1998).

Of the local aggregation methods compared by Gong, Yang, Wang *et al.* (2007) and Tombari, Mattoccia, Di Stefano *et al.* (2008), the fast variable window approach of Veksler (2003) and the locally weighting approach developed by Yoon and Kweon (2006) consistently stood out as having the best tradeoff between performance and speed.[5] The local weighting technique, in particular, is interesting because, instead of using square windows with uniform weighting, each pixel within an aggregation window influences the final matching cost based on its color similarity and spatial distance, just as in bilinear filtering (Figure 11.9c). (In fact, their aggregation step is closely related to doing a joint bilateral filter on the color/disparity image, except that it is done symmetrically in both reference and target images.) The segmentation-based aggregation method of Tombari, Mattoccia, and Di Stefano (2007) did even better, although a fast implementation of this algorithm does not yet exist.

In local methods, the emphasis is on the matching cost computation and cost aggregation steps. Computing the final disparities is trivial: simply choose at each pixel the disparity associated with the minimum cost value. Thus, these methods perform a local "winner-take-all" (WTA) optimization at each pixel. A limitation of this approach (and many other

[5] More recent and extensive results from Tombari, Mattoccia, Di Stefano *et al.* (2008) can be found at http://www.vision.deis.unibo.it/spe/SPEHome.aspx.

correspondence algorithms) is that uniqueness of matches is only enforced for one image (the *reference image*), while points in the other image might match multiple points, unless cross-checking and subsequent hole filling is used (Fua 1993; Hirschmüller and Scharstein 2009).

11.4.1 Sub-pixel estimation and uncertainty

Most stereo correspondence algorithms compute a set of disparity estimates in some discretized space, e.g., for integer disparities (exceptions include continuous optimization techniques such as optical flow (Bergen, Anandan, Hanna *et al.* 1992) or splines (Szeliski and Coughlan 1997)). For applications such as robot navigation or people tracking, these may be perfectly adequate. However for image-based rendering, such quantized maps lead to very unappealing view synthesis results, i.e., the scene appears to be made up of many thin shearing layers. To remedy this situation, many algorithms apply a sub-pixel refinement stage after the initial discrete correspondence stage. (An alternative is to simply start with more discrete disparity levels (Szeliski and Scharstein 2004).)

Sub-pixel disparity estimates can be computed in a variety of ways, including iterative gradient descent and fitting a curve to the matching costs at discrete disparity levels (Ryan, Gray, and Hunt 1980; Lucas and Kanade 1981; Tian and Huhns 1986; Matthies, Kanade, and Szeliski 1989; Kanade and Okutomi 1994). This provides an easy way to increase the resolution of a stereo algorithm with little additional computation. However, to work well, the intensities being matched must vary smoothly, and the regions over which these estimates are computed must be on the same (correct) surface.

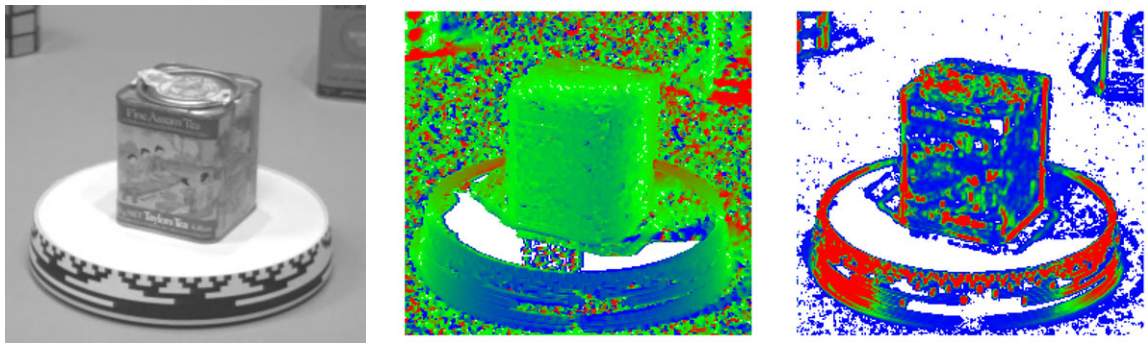
Recently, some questions have been raised about the advisability of fitting correlation curves to integer-sampled matching costs (Shimizu and Okutomi 2001). This situation may even be worse when sampling-insensitive dissimilarity measures are used (Birchfield and Tomasi 1998). These issues are explored in more depth by Szeliski and Scharstein (2004).

Besides sub-pixel computations, there are other ways of post-processing the computed disparities. Occluded areas can be detected using cross-checking, i.e., comparing left-to-right and right-to-left disparity maps (Fua 1993). A median filter can be applied to clean up spurious mismatches, and holes due to occlusion can be filled by surface fitting or by distributing neighboring disparity estimates (Birchfield and Tomasi 1999; Scharstein 1999; Hirschmüller and Scharstein 2009).

Another kind of post-processing, which can be useful in later processing stages, is to associate *confidences* with per-pixel depth estimates (Figure 11.10), which can be done by looking at the curvature of the correlation surface, i.e., how strong the minimum in the DSI image is at the winning disparity. Matthies, Kanade, and Szeliski (1989) show that under the assumption of small noise, photometrically calibrated images, and densely sampled disparities, the variance of a local depth estimate can be estimated as

$$Var(d) = \frac{\sigma_I^2}{a}, \tag{11.7}$$

where a is the curvature of the DSI as a function of d, which can be measured using a local parabolic fit or by squaring all the horizontal gradients in the window, and σ_I^2 is the variance of the image noise, which can be estimated from the minimum SSD score. (See also Section 6.1.4, (8.44), and Appendix B.6.)



(a) (b) (c)

Figure 11.10 Uncertainty in stereo depth estimation (Szeliski 1991b): (a) input image; (b) estimated depth map (blue is closer); (c) estimated confidence(red is higher). As you can see, more textured areas have higher confidence.

11.4.2 *Application*: Stereo-based head tracking

A common application of real-time stereo algorithms is for tracking the position of a user interacting with a computer or game system. The use of stereo can dramatically improve the reliability of such a system compared to trying to use monocular color and intensity information (Darrell, Gordon, Harville *et al.* 2000). Once recovered, this information can be used in a variety of applications, including controlling a virtual environment or game, correcting the apparent gaze during video conferencing, and background replacement. We discuss the first two applications below and defer the discussion of background replacement to Section 11.5.3.

The use of head tracking to control a user's virtual viewpoint while viewing a 3D object or environment on a computer monitor is sometimes called *fish tank virtual reality*, since the user is observing a 3D world as if it were contained inside a fish tank (Ware, Arthur, and Booth 1993). Early versions of these systems used mechanical head tracking devices and stereo glasses. Today, such systems can be controlled using stereo-based head tracking and stereo glasses can be replaced with autostereoscopic displays. Head tracking can also be used to construct a "virtual mirror", where the user's head can be modified in real-time using a variety of visual effects (Darrell, Baker, Crow *et al.* 1997).

Another application of stereo head tracking and 3D reconstruction is in gaze correction (Ott, Lewis, and Cox 1993). When a user participates in a desktop video-conference or video chat, the camera is usually placed on top of the monitor. Since the person is gazing at a window somewhere on the screen, it appears as if they are looking down and away from the other participants, instead of straight at them. Replacing the single camera with two or more cameras enables a virtual view to be constructed right at the position where they are looking resulting in virtual eye contact. Real-time stereo matching is used to construct an accurate 3D head model and view interpolation (Section 13.1) is used to synthesize the novel in-between view (Criminisi, Shotton, Blake *et al.* 2003).

11.5 Global optimization

Global stereo matching methods perform some optimization or iteration steps after the disparity computation phase and often skip the aggregation step altogether, because the global smoothness constraints perform a similar function. Many global methods are formulated in an energy-minimization framework, where, as we saw in Sections 3.7 (3.100–3.102) and 8.4, the objective is to find a solution d that minimizes a global energy,

$$E(d) = E_d(d) + \lambda E_s(d). \tag{11.8}$$

The data term, $E_d(d)$, measures how well the disparity function d agrees with the input image pair. Using our previously defined disparity space image, we define this energy as

$$E_d(d) = \sum_{(x,y)} C(x, y, d(x, y)), \tag{11.9}$$

where C is the (initial or aggregated) matching cost DSI.

The smoothness term $E_s(d)$ encodes the smoothness assumptions made by the algorithm. To make the optimization computationally tractable, the smoothness term is often restricted to measuring only the differences between neighboring pixels' disparities,

$$E_s(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1)), \tag{11.10}$$

where ρ is some monotonically increasing function of disparity difference. It is also possible to use larger neighborhoods, such as $\mathcal{N}_8$, which can lead to better boundaries (Boykov and Kolmogorov 2003), or to use second-order smoothness terms (Woodford, Reid, Torr *et al.* 2008), but such terms require more complex optimization techniques. An alternative to smoothness functionals is to use a lower-dimensional representation such as splines (Szeliski and Coughlan 1997).

In standard regularization (Section 3.7.1), ρ is a quadratic function, which makes d smooth everywhere and may lead to poor results at object boundaries. Energy functions that do not have this problem are called *discontinuity-preserving* and are based on robust ρ functions (Terzopoulos 1986b; Black and Rangarajan 1996). The seminal paper by Geman and Geman (1984) gave a Bayesian interpretation of these kinds of energy functions and proposed a discontinuity-preserving energy function based on Markov random fields (MRFs) and additional *line processes*, which are additional binary variables that control whether smoothness penalties are enforced or not. Black and Rangarajan (1996) show how independent line process variables can be replaced by robust pairwise disparity terms.

The terms in E_s can also be made to depend on the intensity differences, e.g.,

$$\rho_d(d(x, y) - d(x + 1, y)) \cdot \rho_I(\|I(x, y) - I(x + 1, y)\|), \tag{11.11}$$

where ρ_I is some monotonically decreasing function of intensity differences that lowers smoothness costs at high-intensity gradients. This idea (Gamble and Poggio 1987; Fua 1993; Bobick and Intille 1999; Boykov, Veksler, and Zabih 2001) encourages disparity discontinuities to coincide with intensity or color edges and appears to account for some of the good performance of global optimization approaches. While most researchers set these functions

heuristically, Scharstein and Pal (2007) show how the free parameters in such *conditional random fields* (Section 3.7.2, (3.118)) can be learned from ground truth disparity maps.

Once the global energy has been defined, a variety of algorithms can be used to find a (local) minimum. Traditional approaches associated with regularization and Markov random fields include continuation (Blake and Zisserman 1987), simulated annealing (Geman and Geman 1984; Marroquin, Mitter, and Poggio 1987; Barnard 1989), highest confidence first (Chou and Brown 1990), and mean-field annealing (Geiger and Girosi 1991).

More recently, *max-flow* and *graph cut* methods have been proposed to solve a special class of global optimization problems (Roy and Cox 1998; Boykov, Veksler, and Zabih 2001; Ishikawa 2003). Such methods are more efficient than simulated annealing and have produced good results, as have techniques based on loopy belief propagation (Sun, Zheng, and Shum 2003; Tappen and Freeman 2003). Appendix B.5 and a recent survey paper on MRF inference (Szeliski, Zabih, Scharstein *et al.* 2008) discuss and compare such techniques in more detail.

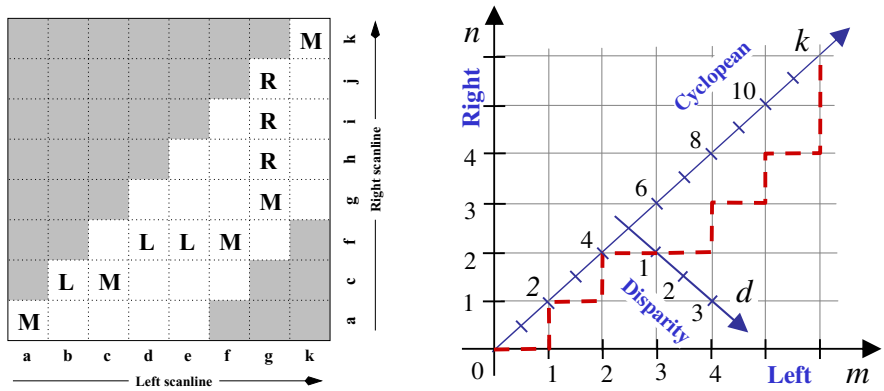
While global optimization techniques currently produce the best stereo matching results, there are some alternative approaches worth studying.

Cooperative algorithms. Cooperative algorithms, inspired by computational models of human stereo vision, were among the earliest methods proposed for disparity computation (Dev 1974; Marr and Poggio 1976; Marroquin 1983; Szeliski and Hinton 1985; Zitnick and Kanade 2000). Such algorithms iteratively update disparity estimates using non-linear operations that result in an overall behavior similar to global optimization algorithms. In fact, for some of these algorithms, it is possible to explicitly state a global function that is being minimized (Scharstein and Szeliski 1998).

Coarse-to-fine and incremental warping. Most of today's best algorithms first enumerate all possible matches at all possible disparities and then select the best set of matches in some way. Faster approaches can sometimes be obtained using methods inspired by classic (infinitesimal) optical flow computation. Here, images are successively warped and disparity estimates incrementally updated until a satisfactory registration is achieved. These techniques are most often implemented within a coarse-to-fine hierarchical refinement framework (Quam 1984; Bergen, Anandan, Hanna *et al.* 1992; Barron, Fleet, and Beauchemin 1994; Szeliski and Coughlan 1997).

11.5.1 Dynamic programming

A different class of global optimization algorithm is based on *dynamic programming*. While the 2D optimization of Equation (11.8) can be shown to be NP-hard for common classes of smoothness functions (Veksler 1999), dynamic programming can find the global minimum for independent scanlines in polynomial time. Dynamic programming was first used for stereo vision in sparse, edge-based methods (Baker and Binford 1981; Ohta and Kanade 1985). More recent approaches have focused on the dense (intensity-based) scanline matching problem (Belhumeur 1996; Geiger, Ladendorf, and Yuille 1992; Cox, Hingorani, Rao *et al.* 1996; Bobick and Intille 1999; Birchfield and Tomasi 1999). These approaches work by computing the minimum-cost path through the matrix of all pairwise matching costs between two corresponding scanlines, i.e., through a horizontal slice of the DSI. Partial occlusion is



(a) (b)

Figure 11.11 Stereo matching using dynamic programming, as illustrated by (a) Scharstein and Szeliski (2002) © 2002 Springer and (b) Kolmogorov, Criminisi, Blake *et al.* (2006). © 2006 IEEE. For each pair of corresponding scanlines, a minimizing path through the matrix of all pairwise matching costs (DSI) is selected. Lowercase letters (a–k) symbolize the intensities along each scanline. Uppercase letters represent the selected path through the matrix. Matches are indicated by M, while partially occluded points (which have a fixed cost) are indicated by L or R, corresponding to points only visible in the left or right images, respectively. Usually, only a limited disparity range is considered (0–4 in the figure, indicated by the non-shaded squares). The representation in (a) allows for diagonal moves while the representation in (b) does not. Note that these diagrams, which use the *Cyclopean* representation of depth, i.e., depth relative to a camera between the two input cameras, show an "unskewed" x-d slice through the DSI.

handled explicitly by assigning a group of pixels in one image to a single pixel in the other image. Figure 11.11 schematically shows how DP works, while Figure 11.5f shows a real DSI slice over which the DP is applied.

To implement dynamic programming for a scanline y, each entry (state) in a 2D cost matrix $D(m, n)$ is computed by combining its DSI value

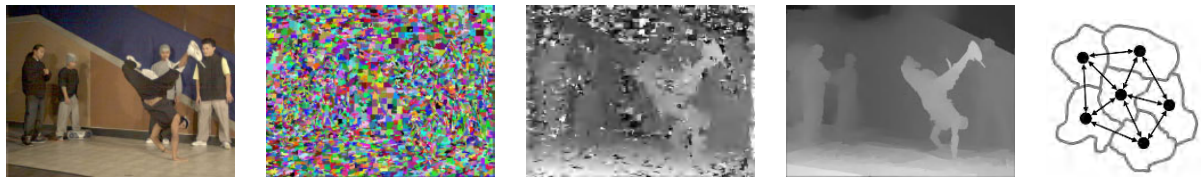
$$C'(m, n) = C(m + n, m - n, y) \tag{11.12}$$

with one of its predecessor cost values. Using the representation shown in Figure 11.11a, which allows for "diagonal" moves, the aggregated match costs can be recursively computed as

$$\begin{aligned} D(m, n, M) &= \min(D(m-1, n-1, M), D(m-1, n, L), D(m-1, n-1, R)) \\ &\quad + C'(m, n) \\ D(m, n, L) &= \min(D(m-1, n-1, M), D(m-1, n, L)) + O \\ D(m, n, R) &= \min(D(m, n-1, M), D(m, n-1, R)) + O, \end{aligned} \tag{11.13}$$

where O is a per-pixel occlusion cost. The aggregation rules corresponding to Figure 11.11b are given by Kolmogorov, Criminisi, Blake *et al.* (2006), who also use a two-state foreground–background model for bi-layer segmentation.

Problems with dynamic programming stereo include the selection of the right cost for occluded pixels and the difficulty of enforcing inter-scanline consistency, although several



(a) (b) (c) (d) (e)

Figure 11.12 Segmentation-based stereo matching (Zitnick, Kang, Uyttendaele *et al.* 2004) © 2004 ACM: (a) input color image; (b) color-based segmentation; (c) initial disparity estimates; (d) final piecewise-smoothed disparities; (e) MRF neighborhood defined over the segments in the disparity space distribution (Zitnick and Kang 2007) © 2007 Springer.

methods propose ways of addressing the latter (Ohta and Kanade 1985; Belhumeur 1996; Cox, Hingorani, Rao *et al.* 1996; Bobick and Intille 1999; Birchfield and Tomasi 1999; Kolmogorov, Criminisi, Blake *et al.* 2006). Another problem is that the dynamic programming approach requires enforcing the *monotonicity* or *ordering constraint* (Yuille and Poggio 1984). This constraint requires that the relative ordering of pixels on a scanline remain the same between the two views, which may not be the case in scenes containing narrow foreground objects.

An alternative to traditional dynamic programming, introduced by Scharstein and Szeliski (2002), is to neglect the vertical smoothness constraints in (11.10) and simply optimize independent scanlines in the global energy function (11.8), which can easily be done using a recursive algorithm,

$$D(x, y, d) = C(x, y, d) + \min_{d'} \{D(x - 1, y, d') + \rho_d(d - d')\} . \tag{11.14}$$

The advantage of this *scanline optimization* algorithm is that it computes the same representation and minimizes a reduced version of the same energy function as the full 2D energy function (11.8). Unfortunately, it still suffers from the same streaking artifacts as dynamic programming.

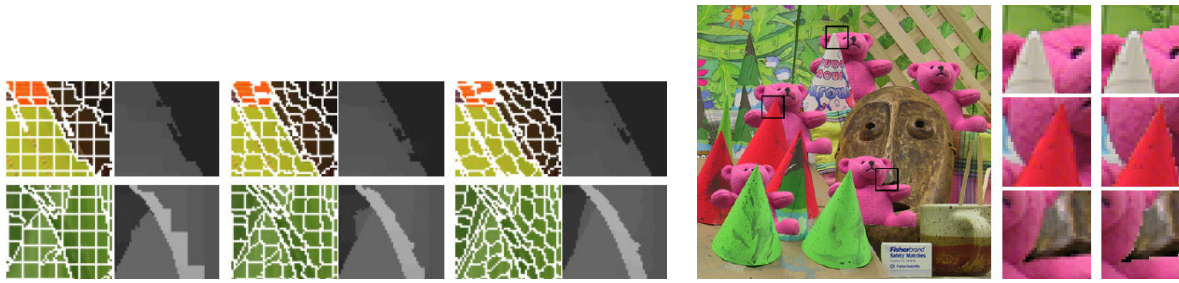
A much better approach is to evaluate the cumulative cost function (11.14) from multiple directions, e.g, from the eight cardinal directions, N, E, W, S, NE, SE, SW, NW (Hirschmüller 2008). The resulting *semi-global* optimization performs quite well and is extremely efficient to implement.

Even though dynamic programming and scanline optimization algorithms do not generally produce *the* most accurate stereo reconstructions, when combined with sophisticated aggregation strategies, they can produce very fast and high-quality results.

11.5.2 Segmentation-based techniques

While most stereo matching algorithms perform their computations on a per-pixel basis, some of the more recent techniques first segment the images into regions and then try to label each region with a disparity.

For example, Tao, Sawhney, and Kumar (2001) segment the reference image, estimate per-pixel disparities using a local technique, and then do local plane fits inside each segment



(a) (b)

Figure 11.13 Stereo matching with adaptive over-segmentation and matting (Taguchi, Wilburn, and Zitnick 2008) © 2008 IEEE: (a) segment boundaries are refined during the optimization, leading to more accurate results (e.g., the thin green leaf in the bottom row); (b) alpha mattes are extracted at segment boundaries, which leads to visually better compositing results (middle column).

before applying smoothness constraints between neighboring segments. Zitnick, Kang, Uyttendaele *et al.* (2004) and Zitnick and Kang (2007) use over-segmentation to mitigate initial bad segmentations. After a set of initial cost values for each segment has been stored into a *disparity space distribution* (DSD), iterative relaxation (or loopy belief propagation, in the more recent work of Zitnick and Kang (2007)) is used to adjust the disparity estimates for each segment, as shown in Figure 11.12. Taguchi, Wilburn, and Zitnick (2008) refine the segment shapes as part of the optimization process, which leads to much improved results, as shown in Figure 11.13.

Even more accurate results are obtained by Klaus, Sormann, and Karner (2006), who first segment the reference image using mean shift, run a small (3×3) SAD plus gradient SAD (weighted by cross-checking) to get initial disparity estimates, fit local planes, re-fit with global planes, and then run a final MRF on plane assignments with loopy belief propagation. When the algorithm was first introduced in 2006, it was the top ranked algorithm on the evaluation site at http://vision.middlebury.edu/stereo; in early 2010, it still had the top rank on the new evaluation datasets.

The highest ranked algorithm, by Wang and Zheng (2008), follows a similar approach of segmenting the image, doing local plane fits, and then performing cooperative optimization of neighboring plane fit parameters. Another highly ranked algorithm, by Yang, Wang, Yang *et al.* (2009), uses the color correlation approach of Yoon and Kweon (2006) and hierarchical belief propagation to obtain an initial set of disparity estimates. After left–right consistency checking to detect occluded pixels, the data terms for low-confidence and occluded pixels are recomputed using segmentation-based plane fits and one or more rounds of hierarchical belief propagation are used to obtain the final disparity estimates.

Another important ability of segmentation-based stereo algorithms, which they share with algorithms that use explicit layers (Baker, Szeliski, and Anandan 1998; Szeliski and Golland 1999) or boundary extraction (Hasinoff, Kang, and Szeliski 2006), is the ability to extract fractional pixel alpha mattes at depth discontinuities (Bleyer, Gelautz, Rother *et al.* 2009). This ability is crucial when attempting to create virtual view interpolation without clinging boundary or tearing artifacts (Zitnick, Kang, Uyttendaele *et al.* 2004) and also to seamlessly insert virtual objects (Taguchi, Wilburn, and Zitnick 2008), as shown in Figure 11.13b.



Figure 11.14 Background replacement using z-keying with a bi-layer segmentation algorithm (Kolmogorov, Criminisi, Blake *et al.* 2006) © 2006 IEEE.

Since new stereo matching algorithms continue to be introduced every year, it is a good idea to periodically check the Middlebury evaluation site at http://vision.middlebury.edu/stereo for a listing of the most recent algorithms to be evaluated.

11.5.3 *Application*: Z-keying and background replacement

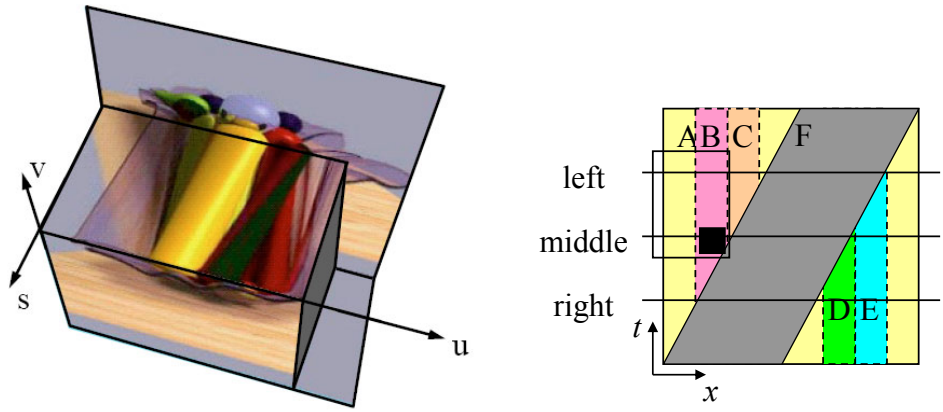
Another application of real-time stereo matching is *z-keying*, which is the process of segmenting a foreground actor from the background using depth information, usually for the purpose of replacing the background with some computer-generated imagery, as shown in Figure 11.2g.

Originally, z-keying systems required expensive custom-built hardware to produce the desired depth maps in real time and were, therefore, restricted to broadcast studio applications (Kanade, Yoshida, Oda *et al.* 1996; Iddan and Yahav 2001). Off-line systems were also developed for estimating 3D multi-viewpoint geometry from video streams (Section 13.5.4) (Kanade, Rander, and Narayanan 1997; Carranza, Theobalt, Magnor *et al.* 2003; Zitnick, Kang, Uyttendaele *et al.* 2004; Vedula, Baker, and Kanade 2005). Recent advances in highly accurate real-time stereo matching, however, now make it possible to perform z-keying on regular PCs, enabling desktop videoconferencing applications such as those shown in Figure 11.14 (Kolmogorov, Criminisi, Blake *et al.* 2006).

11.6 Multi-view stereo

While matching pairs of images is a useful way of obtaining depth information, matching more images can lead to even better results. In this section, we review not only techniques for creating complete 3D object models, but also simpler techniques for improving the quality of depth maps using multiple source images.

As we saw in our discussion of plane sweep (Section 11.1.2), it is possible to resample all neighboring k images at each disparity hypothesis d into a generalized disparity space



(a) (b)

Figure 11.15 Epipolar plane image (EPI) (Gortler, Grzeszczuk, Szeliski *et al.* 1996) © 1996 ACM and a schematic EPI (Kang, Szeliski, and Chai 2001) © 2001 IEEE. (a) The Lumigraph (light field) (Section 13.3) is the 4D space of all light rays passing through a volume of space. Taking a 2D slice results in all of the light rays embedded in a plane and is equivalent to a scanline taken from a stacked EPI volume. Objects at different depths move sideways with velocities (slopes) proportional to their inverse depth. Occlusion (and translucency) effects can easily be seen in this representation. (b) The EPI corresponding to Figure 11.16 showing the three images (middle, left, and right) as slices through the EPI volume. The spatially and temporally shifted window around the black pixel is indicated by the rectangle, showing the right image is not being used in matching.

volume $\tilde{I}(x, y, d, k)$. The simplest way to take advantage of these additional images is to sum up their differences from the reference image I_r as in (11.4),

$$C(x, y, d) = \sum_k \rho(\tilde{I}(x, y, d, k) - I_r(x, y)). \tag{11.15}$$

This is the basis of the well-known sum of summed-squared-difference (SSSD) and SSAD approaches (Okutomi and Kanade 1993; Kang, Webb, Zitnick *et al.* 1995), which can be extended to reason about likely patterns of occlusion (Nakamura, Matsuura, Satoh *et al.* 1996). More recent work by Gallup, Frahm, Mordohai *et al.* (2008) show how to adapt the baselines used to the expected depth in order to get the best tradeoff between geometric accuracy (wide baseline) and robustness to occlusion (narrow baseline). Alternative multi-view cost metrics include measures such as synthetic focus sharpness and the entropy of the pixel color distribution (Vaish, Szeliski, Zitnick *et al.* 2006).

A useful way to visualize the multi-frame stereo estimation problem is to examine the *epipolar plane image* (EPI) formed by stacking corresponding scanlines from all the images, as shown in Figures 8.13c and 11.15 (Bolles, Baker, and Marimont 1987; Baker and Bolles 1989; Baker 1989). As you can see in Figure 11.15, as a camera translates horizontally (in a standard horizontally rectified geometry), objects at different depths move sideways at a rate inversely proportional to their depth (11.1).[6] Foreground objects occlude background objects,

[6] The four-dimensional generalization of the EPI is the *light field*, which we study in Section 13.3. In principle, there is enough information in a light field to recover both the shape and the BRDF of objects (Soatto, Yezzi, and Jin 2003), although relatively little progress has been made to date on this topic.

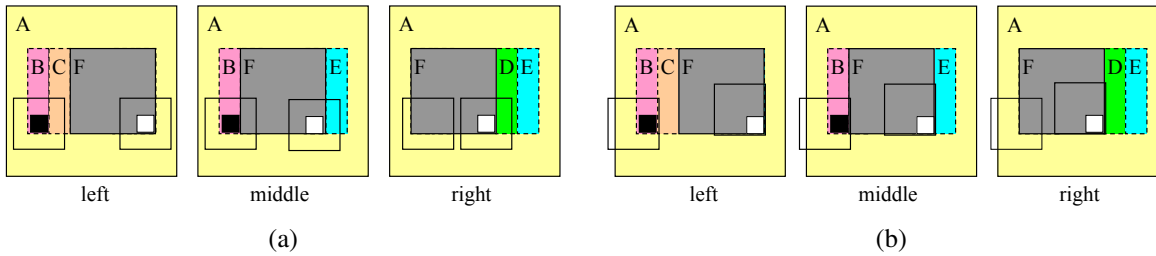


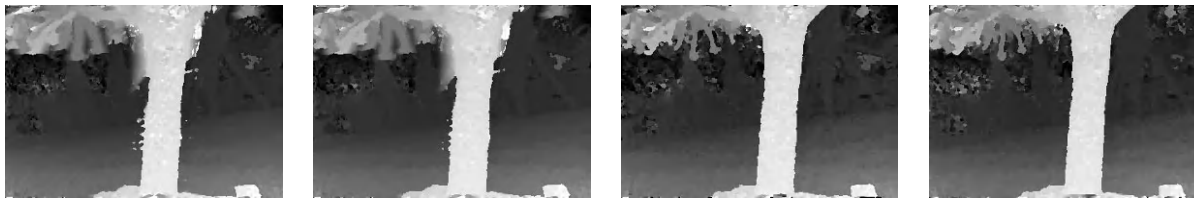
Figure 11.16 Spatio-temporally shiftable windows (Kang, Szeliski, and Chai 2001) © 2001 IEEE: A simple three-image sequence (the middle image is the reference image), which has a moving frontal gray square (marked F) and a stationary background. Regions B, C, D, and E are partially occluded. (a) A regular SSD algorithm will make mistakes when matching pixels in these regions (e.g. the window centered on the black pixel in region B) and in windows straddling depth discontinuities (the window centered on the white pixel in region F). (b) Shiftable windows help mitigate the problems in partially occluded regions and near depth discontinuities. The shifted window centered on the white pixel in region F matches correctly in all frames. The shifted window centered on the black pixel in region B matches correctly in the left image, but requires temporal selection to disable matching the right image. Figure 11.15b shows an EPI corresponding to this sequence and describes in more detail how temporal selection works.

which can be seen as *EPI-strips* (Criminisi, Kang, Swaminathan *et al.* 2005) occluding other strips in the EPI. If we are given a dense enough set of images, we can find such strips and reason about their relationships in order to both reconstruct the 3D scene and make inferences about translucent objects (Tsin, Kang, and Szeliski 2006) and specular reflections (Swaminathan, Kang, Szeliski *et al.* 2002; Criminisi, Kang, Swaminathan *et al.* 2005). Alternatively, we can treat the series of images as a set of sequential observations and merge them using Kalman filtering (Matthies, Kanade, and Szeliski 1989) or maximum likelihood inference (Cox 1994).

When fewer images are available, it becomes necessary to fall back on aggregation techniques such as sliding windows or global optimization. With additional input images, however, the likelihood of occlusions increases. It is therefore prudent to adjust not only the best window locations using a shiftable window approach, as shown in Figure 11.16a, but also to optionally select a subset of neighboring frames in order to discount those images where the region of interest is occluded, as shown in Figure 11.16b (Kang, Szeliski, and Chai 2001). Figure11.15b shows how such spatio-temporal selection or shifting of windows corresponds to selecting the most likely un-occluded volumetric region in the epipolar plane image volume.

The results of applying these techniques to the multi-frame *flower garden* image sequence are shown in Figure 11.17, which compares the results of using regular (non-shifted) SSSD with spatially shifted windows and full spatio-temporal window selection. (The task of applying stereo to a rigid scene filmed with a moving camera is sometimes called *motion stereo*). Similar improvements from using spatio-temporal selection are reported by (Kang and Szeliski 2004) and are evident even when local measurements are combined with global optimization.

While computing a depth map from multiple inputs outperforms pairwise stereo matching, even more dramatic improvements can be obtained by estimating multiple depth maps



(a) (b) (c) (d)

Figure 11.17 Local (5×5 window-based) matching results (Kang, Szeliski, and Chai 2001) © 2001 IEEE: (a) window that is not spatially perturbed (centered); (b) spatially perturbed window; (c) using the best five of 10 neighboring frames; (d) using the better half sequence. Notice how the results near the tree trunk are improved using temporal selection.

simultaneously (Szeliski 1999; Kang and Szeliski 2004). The existence of multiple depth maps enables more accurate reasoning about occlusions, as regions which are occluded in one image may be visible (and matchable) in others. The multi-view reconstruction problem can be formulated as the simultaneous estimation of depth maps at key frames (Figure 8.13c) while maximizing not only photoconsistency and piecewise disparity smoothness but also the consistency between disparity estimates at different frames. While Szeliski (1999) and Kang and Szeliski (2004) use soft (penalty-based) constraints to encourage multiple disparity maps to be consistent, Kolmogorov and Zabih (2002) show how such consistency measures can be encoded as hard constraints, which guarantee that the multiple depth maps are not only similar but actually identical in overlapping regions. Newer algorithms that simultaneously estimate multiple disparity maps include papers by Maitre, Shinagawa, and Do (2008) and Zhang, Jia, Wong *et al.* (2008).

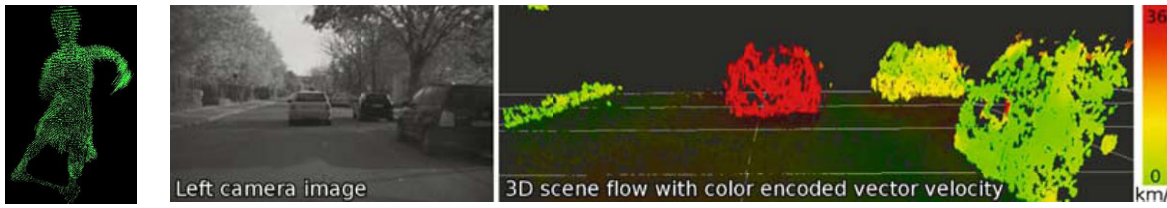
A closely related topic to multi-frame stereo estimation is *scene flow*, in which multiple cameras are used to capture a dynamic scene. The task is then to simultaneously recover the 3D shape of the object at every instant in time and to estimate the full 3D motion of every surface point between frames. Representative papers in this area include those by Vedula, Baker, Rander *et al.* (2005), Zhang and Kambhamettu (2003), Pons, Keriven, and Faugeras (2007), Huguet and Devernay (2007), and Wedel, Rabe, Vaudrey *et al.* (2008). Figure 11.18a shows an image of the 3D scene flow for the tango dancer shown in Figure 11.2h–j, while Figure 11.18b shows 3D scene flows captured from a moving vehicle for the purpose of obstacle avoidance. In addition to supporting mensuration and safety applications, scene flow can be used to support both spatial and temporal view interpolation (Section 13.5.4), as demonstrated by Vedula, Baker, and Kanade (2005).

11.6.1 Volumetric and 3D surface reconstruction

According to Seitz, Curless, Diebel *et al.* (2006):

> *The goal of multi-view stereo is to reconstruct a complete 3D object model from a collection of images taken from known camera viewpoints.*

The most challenging but potentially most useful variant of multi-view stereo reconstruction is to create globally consistent 3D models. This topic has a long history in computer vision, starting with surface mesh reconstruction techniques such as the one developed by



(a) (b)

Figure 11.18 Three-dimensional scene flow: (a) computed from a multi-camera dome surrounding the dancer shown in Figure 11.2h–j (Vedula, Baker, Rander *et al.* 2005) © 2005 IEEE; (b) computed from stereo cameras mounted on a moving vehicle (Wedel, Rabe, Vaudrey *et al.* 2008) © 2008 Springer.

Fua and Leclerc (1995) (Figure 11.19a). A variety of approaches and representations have been used to solve this problem, including 3D voxel representations (Seitz and Dyer 1999; Szeliski and Golland 1999; De Bonet and Viola 1999; Kutulakos and Seitz 2000; Eisert, Steinbach, and Girod 2000; Slabaugh, Culbertson, Slabaugh *et al.* 2004; Sinha and Pollefeys 2005; Vogiatzis, Hernandez, Torr *et al.* 2007; Hiep, Keriven, Pons *et al.* 2009), level sets (Faugeras and Keriven 1998; Pons, Keriven, and Faugeras 2007), polygonal meshes (Fua and Leclerc 1995; Narayanan, Rander, and Kanade 1998; Hernandez and Schmitt 2004; Furukawa and Ponce 2009), and multiple depth maps (Kolmogorov and Zabih 2002). Figure 11.19 shows representative examples of 3D object models reconstructed using some of these techniques.

In order to organize and compare all these techniques, Seitz, Curless, Diebel *et al.* (2006) developed a six-point taxonomy that can help classify algorithms according to the *scene representation*, *photoconsistency measure*, *visibility model*, *shape priors*, *reconstruction algorithm*, and *initialization requirements* they use. Below, we summarize some of these choices and list a few representative papers. For more details, please consult the full survey paper (Seitz, Curless, Diebel *et al.* 2006) and the evaluation Web site, http://vision.middlebury.edu/mview/, which contains pointers to even more recent papers and results.

Scene representation. One of the more popular 3D representations is a uniform grid of 3D voxels,[7] which can be reconstructed using a variety of carving (Seitz and Dyer 1999; Kutulakos and Seitz 2000) or optimization (Sinha and Pollefeys 2005; Vogiatzis, Hernandez, Torr *et al.* 2007; Hiep, Keriven, Pons *et al.* 2009) techniques. Level set techniques (Section 5.1.4) also operate on a uniform grid but, instead of representing a binary occupancy map, they represent the signed distance to the surface (Faugeras and Keriven 1998; Pons, Keriven, and Faugeras 2007), which can encode a finer level of detail. Polygonal meshes are another popular representation (Fua and Leclerc 1995; Narayanan, Rander, and Kanade 1998; Isidoro and Sclaroff 2003; Hernandez and Schmitt 2004; Furukawa and Ponce 2009; Hiep, Keriven, Pons *et al.* 2009). Meshes are the standard representation used in computer graphics and also readily support the computation of visibility and occlusions. Finally, as we discussed in the previous section, multiple depth maps can also be used (Szeliski 1999; Kolmogorov and Zabih 2002; Kang and Szeliski 2004). Many algorithms also use more than a single representation, e.g., they may start by computing multiple depth maps and then merge

[7] For outdoor scenes that go to infinity, a non-uniform gridding of space may be preferable (Slabaugh, Culbertson, Slabaugh *et al.* 2004).

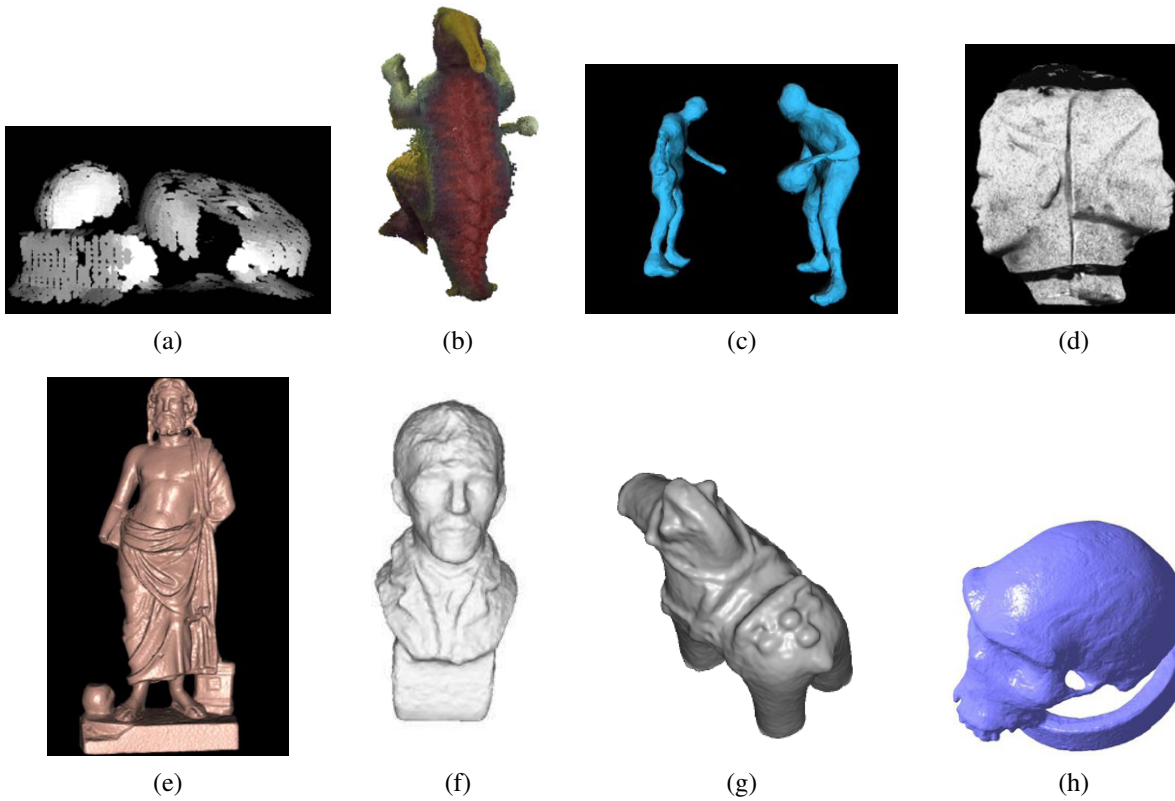


Figure 11.19 Multi-view stereo algorithms: (a) surface-based stereo (Fua and Leclerc 1995); (b) voxel coloring (Seitz and Dyer 1999) © 1999 Springer; (c) depth map merging (Narayanan, Rander, and Kanade 1998); (d) level set evolution (Faugeras and Keriven 1998) © 1998 IEEE; (e) silhouette and stereo fusion (Hernandez and Schmitt 2004) © 2004 Elsevier; (f) multi-view image matching (Pons, Keriven, and Faugeras 2005) © 2005 IEEE; (g) volumetric graph cut (Vogiatzis, Torr, and Cipolla 2005) © 2005 IEEE; (h) carved visual hulls (Furukawa and Ponce 2009) © 2009 Springer.

them into a 3D object model (Narayanan, Rander, and Kanade 1998; Furukawa and Ponce 2009; Goesele, Curless, and Seitz 2006; Goesele, Snavely, Curless *et al.* 2007; Furukawa, Curless, Seitz *et al.* 2010).

Photoconsistency measure. As we discussed in (Section 11.3.1), a variety of similarity measures can be used to compare pixel values in different images, including measures that try to discount illumination effects or be less sensitive to outliers. In multi-view stereo, algorithms have a choice of computing these measures directly on the surface of the model, i.e., in *scene space*, or projecting pixel values from one image (or from a textured model) back into another image, i.e., in *image space*. (The latter corresponds more closely to a Bayesian approach, since input images are noisy measurements of the colored 3D model.) The geometry of the object, i.e., its distance to each camera and its local surface normal, when available, can be used to adjust the matching windows used in the computation to account for foreshortening and scale change (Goesele, Snavely, Curless *et al.* 2007).

Visibility model. A big advantage that multi-view stereo algorithms have over single-depth-map approaches is their ability to reason in a principled manner about visibility and occlusions. Techniques that use the current state of the 3D model to predict which surface pixels are visible in each image (Kutulakos and Seitz 2000; Faugeras and Keriven 1998; Vogiatzis, Hernandez, Torr *et al.* 2007; Hiep, Keriven, Pons *et al.* 2009) are classified as using *geometric visibility models* in the taxonomy of Seitz, Curless, Diebel *et al.* (2006). Techniques that select a neighboring subset of image to match are called *quasi-geometric* (Narayanan, Rander, and Kanade 1998; Kang and Szeliski 2004; Hernandez and Schmitt 2004), while techniques that use traditional robust similarity measures are called *outlier-based*. While full geometric reasoning is the most principled and accurate approach, it can be very slow to evaluate and depends on the evolving quality of the current surface estimate to predict visibility, which can be a bit of a chicken-and-egg problem, unless conservative assumptions are used, as they are by Kutulakos and Seitz (2000).

Shape priors. Because stereo matching is often underconstrained, especially in textureless regions, most matching algorithms adopt (either explicitly or implicitly) some form of prior model for the expected shape. Many of the techniques that rely on optimization use a 3D smoothness or area-based photoconsistency constraint, which, because of the natural tendency of smooth surfaces to shrink inwards, often results in a *minimal surface* prior (Faugeras and Keriven 1998; Sinha and Pollefeys 2005; Vogiatzis, Hernandez, Torr *et al.* 2007). Approaches that carve away the volume of space often stop once a photoconsistent solution is found (Seitz and Dyer 1999; Kutulakos and Seitz 2000), which corresponds to a *maximal surface* bias, i.e., these techniques tend to over-estimate the true shape. Finally, multiple depth map approaches often adopt traditional *image-based* smoothness (regularization) constraints.

Reconstruction algorithm. The details of how the actual reconstruction algorithm proceeds is where the largest variety—and greatest innovation—in multi-view stereo algorithms can be found.

Some approaches use global optimization defined over a three-dimensional photoconsistency volume to recover a complete surface. Approaches based on graph cuts use polynomial complexity binary segmentation algorithms to recover the object model defined on the voxel grid (Sinha and Pollefeys 2005; Vogiatzis, Hernandez, Torr *et al.* 2007; Hiep, Keriven, Pons *et al.* 2009). Level set approaches use a continuous surface evolution to find a good minimum in the configuration space of potential surfaces and therefore require a reasonably good initialization (Faugeras and Keriven 1998; Pons, Keriven, and Faugeras 2007). In order for the photoconsistency volume to be meaningful, matching costs need to be computed in some robust fashion, e.g., using sets of limited views or by aggregating multiple depth maps.

An alternative approach to global optimization is to sweep through the 3D volume while computing both photoconsistency and visibility simultaneously. The *voxel coloring* algorithm of Seitz and Dyer (1999) performs a front-to-back plane sweep. On every plane, any voxels that are sufficiently photoconsistent are labeled as part of the object. The corresponding pixels in the source images can then be "erased", since they are already accounted for, and therefore do not contribute to further photoconsistency computations. (A similar approach, albeit without the front-to-back sweep order, is used by Szeliski and Golland (1999).) The resulting 3D volume, under noise- and resampling-free conditions, is guaranteed to produce



(a) (b) (c) (d) (e) (f)

Figure 11.20 The multi-view stereo data sets captured by Seitz, Curless, Diebel *et al.* (2006) © 2006 Springer. Only (a) and (b) are currently used for evaluation.

both a photoconsistent 3D model and to enclose whatever true 3D object model generated the images.

Unfortunately, voxel coloring is only guaranteed to work if all of the cameras lie on the same side of the sweep planes, which is not possible in general ring configurations of cameras. Kutulakos and Seitz (2000) generalize voxel coloring to *space carving*, where subsets of cameras that satisfy the voxel coloring constraint are iteratively selected and the 3D voxel grid is alternately carved away along different axes.

Another popular approach to multi-view stereo is to first independently compute multiple depth maps and then merge these partial maps into a complete 3D model. Approaches to depth map merging, which are discussed in more detail in Section 12.2.1, include signed distance functions (Curless and Levoy 1996), used by Goesele, Curless, and Seitz (2006), and Poisson surface reconstruction (Kazhdan, Bolitho, and Hoppe 2006), used by Goesele, Snavely, Curless *et al.* (2007). It is also possible to reconstruct sparser representations, such as 3D points and lines, and to interpolate them to full 3D surfaces (Section 12.3.1) (Taylor 2003).

Initialization requirements. One final element discussed by Seitz, Curless, Diebel *et al.* (2006) is the varying degrees of initialization required by different algorithms. Because some algorithms refine or evolve a rough 3D model, they require a reasonably accurate (or overcomplete) initial model, which can often be obtained by reconstructing a volume from object silhouettes, as discussed in Section 11.6.2. However, if the algorithm performs a global optimization (Kolev, Klodt, Brox *et al.* 2009; Kolev and Cremers 2009), this dependence on initialization is not an issue.

Empirical evaluation. In order to evaluate the large number of design alternatives in multi-view stereo, Seitz, Curless, Diebel *et al.* (2006) collected a dataset of calibrated images using a spherical gantry. A representative image from each of the six datasets is shown in Figure 11.20, although only the first two datasets have as yet been fully processed and used for evaluation. Figure 11.21 shows the results of running seven different algorithms on the *temple* dataset. As you can see, most of the techniques do an impressive job of capturing the fine details in the columns, although it is also clear that the techniques employ differing amounts of smoothing to achieve these results.

Since the publication of the survey by Seitz, Curless, Diebel *et al.* (2006), the field of

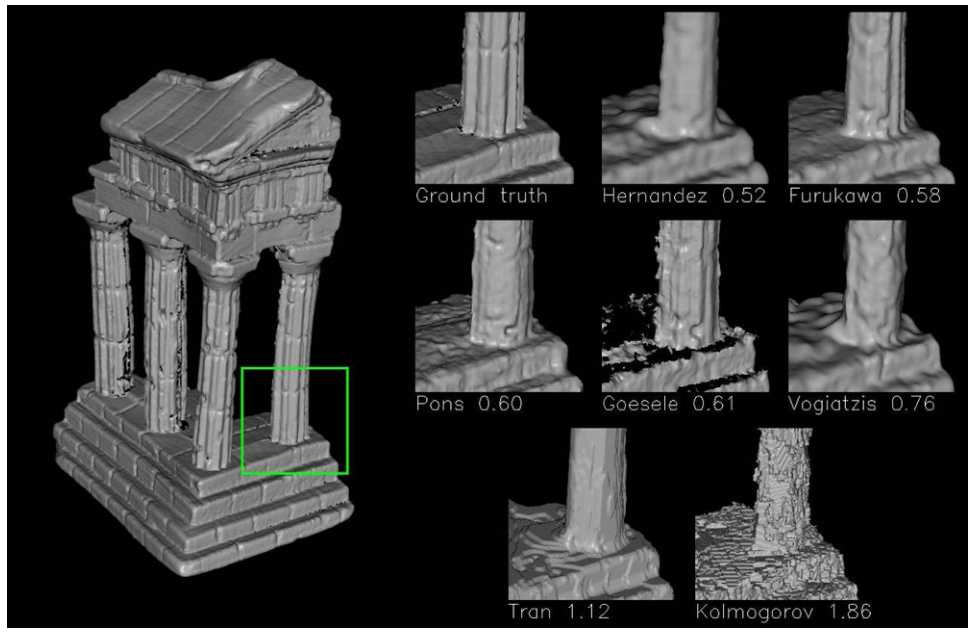


Figure 11.21 Reconstruction results (details) for seven algorithms (Hernandez and Schmitt 2004; Furukawa and Ponce 2009; Pons, Keriven, and Faugeras 2005; Goesele, Curless, and Seitz 2006; Vogiatzis, Torr, and Cipolla 2005; Tran and Davis 2002; Kolmogorov and Zabih 2002) evaluated by Seitz, Curless, Diebel *et al.* (2006) on the 47-image Temple Ring dataset. The numbers underneath each detail image are the accuracy of each of these techniques measured in millimeters.

multi-view stereo has continued to advance at a rapid pace (Strecha, Fransens, and Van Gool 2006; Hernandez, Vogiatzis, and Cipolla 2007; Habbecke and Kobbelt 2007; Furukawa and Ponce 2007; Vogiatzis, Hernandez, Torr *et al.* 2007; Goesele, Snavely, Curless *et al.* 2007; Sinha, Mordohai, and Pollefeys 2007; Gargallo, Prados, and Sturm 2007; Merrell, Akbarzadeh, Wang *et al.* 2007; Zach, Pock, and Bischof 2007b; Furukawa and Ponce 2008; Hornung, Zeng, and Kobbelt 2008; Bradley, Boubekeur, and Heidrich 2008; Zach 2008; Campbell, Vogiatzis, Hernández *et al.* 2008; Kolev, Klodt, Brox *et al.* 2009; Hiep, Keriven, Pons *et al.* 2009; Furukawa, Curless, Seitz *et al.* 2010). The multi-view stereo evaluation site, http://vision.middlebury.edu/mview/, provides quantitative results for these algorithms along with pointers to where to find these papers.

11.6.2 Shape from silhouettes

In many situations, performing a foreground–background segmentation of the object of interest is a good way to initialize or fit a 3D model (Grauman, Shakhnarovich, and Darrell 2003; Vlasic, Baran, Matusik *et al.* 2008) or to impose a convex set of constraints on multi-view stereo (Kolev and Cremers 2008). Over the years, a number of techniques have been developed to reconstruct a 3D volumetric model from the intersection of the binary silhouettes projected into 3D. The resulting model is called a *visual hull* (or sometimes a *line hull*), analogous with the convex hull of a set of points, since the volume is maximal with respect

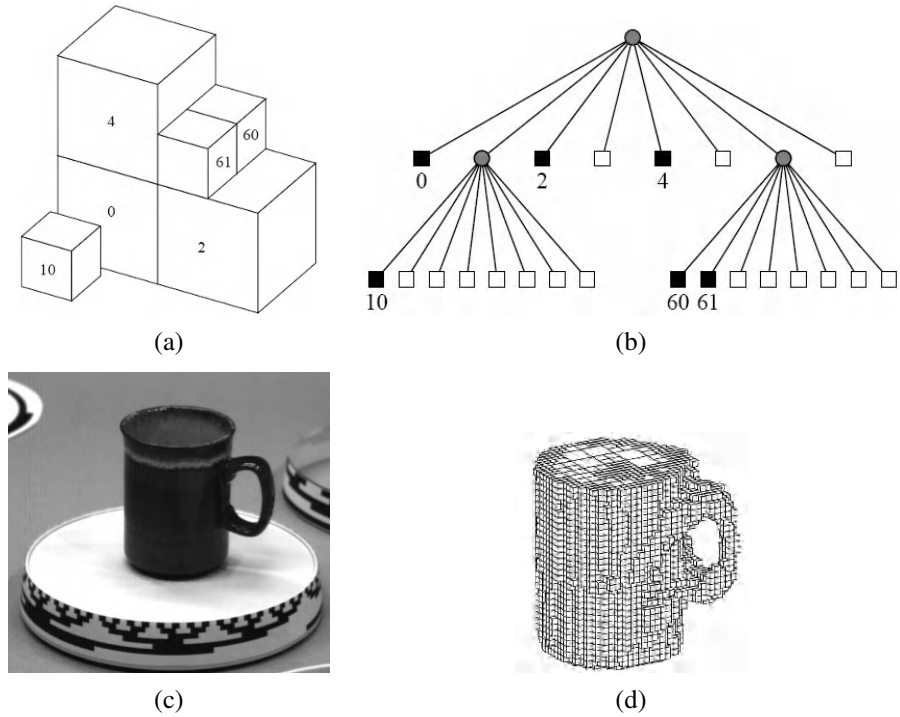


Figure 11.22 Volumetric octree reconstruction from binary silhouettes (Szeliski 1993) © 1993 Elsevier: (a) octree representation and its corresponding (b) tree structure; (c) input image of an object on a turntable; (d) computed 3D volumetric octree model.

to the visual silhouettes and surface elements are tangent to the viewing rays (lines) along the silhouette boundaries (Laurentini 1994). It is also possible to carve away a more accurate reconstruction using multi-view stereo (Sinha and Pollefeys 2005) or by analyzing cast shadows (Savarese, Andreetto, Rushmeier *et al.* 2007).

Some techniques first approximate each silhouette with a polygonal representation and then intersect the resulting faceted conical regions in three-space to produce polyhedral models (Baumgart 1974; Martin and Aggarwal 1983; Matusik, Buehler, and McMillan 2001), which can later be refined using triangular splines (Sullivan and Ponce 1998). Other approaches use voxel-based representations, usually encoded as octrees (Samet 1989), because of the resulting space–time efficiency. Figures 11.22a–b show an example of a 3D octree model and its associated colored tree, where black nodes are interior to the model, white nodes are exterior, and gray nodes are of mixed occupancy. Examples of octree-based reconstruction approaches include those by Potmesil (1987), Noborio, Fukada, and Arimoto (1988), Srivasan, Liang, and Hackwood (1990), and Szeliski (1993).

The approach of Szeliski (1993) first converts each binary silhouette into a one-sided variant of a distance map, where each pixel in the map indicates the largest square that is completely inside (or outside) the silhouette. This makes it fast to project an octree cell into the silhouette to confirm whether it is completely inside or outside the object, so that it can be colored black, white, or left as gray (mixed) for further refinement on a smaller grid. The octree construction algorithm proceeds in a coarse-to-fine manner, first building an

octree at a relatively coarse resolution, and then refining it by revisiting and subdividing all the input images for the gray (mixed) cells whose occupancy has not yet been determined. Figure 11.22d shows the resulting octree model computed from a coffee cup rotating on a turntable.

More recent work on visual hull computation borrows ideas from image-based rendering, and is hence called an *image-based visual hull* (Matusik, Buehler, Raskar *et al.* 2000). Instead of precomputing a global 3D model, an image-based visual hull is recomputed for each new viewpoint, by successively intersecting viewing ray segments with the binary silhouettes in each image. This not only leads to a fast computation algorithm but also enables fast texturing of the recovered model with color values from the input images. This approach can also be combined with high-quality deformable templates to capture and re-animate whole body motion (Vlasic, Baran, Matusik *et al.* 2008).

11.7 Additional reading

The field of stereo correspondence and depth estimation is one of the oldest and most widely studied topics in computer vision. A number of good surveys have been written over the years (Marr and Poggio 1976; Barnard and Fischler 1982; Dhond and Aggarwal 1989; Scharstein and Szeliski 2002; Brown, Burschka, and Hager 2003; Seitz, Curless, Diebel *et al.* 2006) and they can serve as good guides to this extensive literature.

Because of computational limitations and the desire to find appearance-invariant correspondences, early algorithms often focused on finding *sparse correspondences* (Hannah 1974; Marr and Poggio 1979; Mayhew and Frisby 1980; Baker and Binford 1981; Arnold 1983; Grimson 1985; Ohta and Kanade 1985; Bolles, Baker, and Marimont 1987; Matthies, Kanade, and Szeliski 1989; Hsieh, McKeown, and Perlant 1992; Bolles, Baker, and Hannah 1993).

The topic of computing epipolar geometry and pre-rectifying images is covered in Sections 7.2 and 11.1 and is also treated in textbooks on multi-view geometry (Faugeras and Luong 2001; Hartley and Zisserman 2004) and articles specifically on this topic (Torr and Murray 1997; Zhang 1998a,b). The concepts of the *disparity space* and *disparity space image* are often associated with the seminal work by Marr (1982) and the papers of Yang, Yuille, and Lu (1993) and Intille and Bobick (1994). The plane sweep algorithm was first popularized by Collins (1996) and then generalized to a full arbitrary projective setting by Szeliski and Golland (1999) and Saito and Kanade (1999). Plane sweeps can also be formulated using cylindrical surfaces (Ishiguro, Yamamoto, and Tsuji 1992; Kang and Szeliski 1997; Shum and Szeliski 1999; Li, Shum, Tang *et al.* 2004; Zheng, Kang, Cohen *et al.* 2007) or even more general topologies (Seitz 2001).

Once the topology for the cost volume or DSI has been set up, we need to compute the actual photoconsistency measures for each pixel and potential depth. A wide range of such measures have been proposed, as discussed in Section 11.3.1. Some of these are compared in recent surveys and evaluations of matching costs (Scharstein and Szeliski 2002; Hirschmüller and Scharstein 2009).

To compute an actual depth map from these costs, some form of optimization or selection criterion must be used. The simplest of these are sliding windows of various kinds, which are discussed in Section 11.4 and surveyed by Gong, Yang, Wang *et al.* (2007) and Tombari,

Mattoccia, Di Stefano *et al.* (2008). More commonly, global optimization frameworks are used to compute the best disparity field, as described in Section 11.5. These techniques include dynamic programming and truly global optimization algorithms, such as graph cuts and loopy belief propagation. Because the literature on this is so extensive, it is described in more detail in Section 11.5. A good place to find pointers to the latest results in this field is the Middlebury Stereo Vision Page at http://vision.middlebury.edu/stereo.

Algorithms for multi-view stereo typically fall into two categories. The first include algorithms that compute traditional depth maps using several images for computing photoconsistency measures (Okutomi and Kanade 1993; Kang, Webb, Zitnick *et al.* 1995; Nakamura, Matsuura, Satoh *et al.* 1996; Szeliski and Golland 1999; Kang, Szeliski, and Chai 2001; Vaish, Szeliski, Zitnick *et al.* 2006; Gallup, Frahm, Mordohai *et al.* 2008). Optionally, some of these techniques compute multiple depth maps and use additional constraints to encourage the different depth maps to be consistent (Szeliski 1999; Kolmogorov and Zabih 2002; Kang and Szeliski 2004; Maitre, Shinagawa, and Do 2008; Zhang, Jia, Wong *et al.* 2008).

The second category consists of papers that compute true 3D volumetric or surface-based object models. Again, because of the large number of papers published on this topic, rather than citing them here, we refer you to the material in Section 11.6.1, the survey by Seitz, Curless, Diebel *et al.* (2006), and the on-line evaluation Web site at http://vision.middlebury.edu/mview/.

11.8 Exercises

Ex 11.1: Stereo pair rectification Implement the following simple algorithm (Section 11.1.1):

1. Rotate both cameras so that they are looking perpendicular to the line joining the two camera centers $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$. The smallest rotation can be computed from the cross product between the original and desired optical axes.

2. Twist the optical axes so that the horizontal axis of each camera looks in the direction of the other camera. (Again, the cross product between the current x-axis after the first rotation and the line joining the cameras gives the rotation.)

3. If needed, scale up the smaller (less detailed) image so that it has the same resolution (and hence line-to-line correspondence) as the other image.

Now compare your results to the algorithm proposed by Loop and Zhang (1999). Can you think of situations where their approach may be preferable?

Ex 11.2: Rigid direct alignment Modify your spline-based or optical flow motion estimator from Exercise 8.4 to use epipolar geometry, i.e. to only estimate disparity.

(Optional) Extend your algorithm to simultaneously estimate the epipolar geometry (without first using point correspondences) by estimating a base homography corresponding to a reference plane for the dominant motion and then an epipole for the residual parallax (motion).

Ex 11.3: Shape from profiles Reconstruct a surface model from a series of edge images (Section 11.2.1).

1. Extract edges and link them (Exercises 4.7–4.8).
2. Based on previously computed epipolar geometry, match up edges in triplets (or longer sets) of images.
3. Reconstruct the 3D locations of the curves using osculating circles (11.5).
4. Render the resulting 3D surface model as a sparse mesh, i.e., drawing the reconstructed 3D profile curves and links between 3D points in neighboring images with similar osculating circles.

Ex 11.4: Plane sweep Implement a plane sweep algorithm (Section 11.1.2).

If the images are already pre-rectified, this consists simply of shifting images relative to each other and comparing pixels. If the images are not pre-rectified, compute the homography that resamples the target image into the reference image's coordinate system for each plane.

Evaluate a subset of the following similarity measures (Section 11.3.1) and compare their performance by visualizing the disparity space image (DSI), which should be dark for pixels at correct depths:

- squared difference (SD);
- absolute difference (AD);
- truncated or robust measures;
- gradient differences;
- rank or census transform (the latter usually performs better);
- mutual information from a pre-computed joint density function.

Consider using the Birchfield and Tomasi (1998) technique of comparing ranges between neighboring pixels (different shifted or warped images). Also, try pre-compensating images for bias or gain variations using one or more of the techniques discussed in Section 11.3.1.

Ex 11.5: Aggregation and window-based stereo Implement one or more of the matching cost aggregation strategies described in Section 11.4:

- convolution with a box or Gaussian kernel;
- shifting window locations by applying a min filter (Scharstein and Szeliski 2002);
- picking a window that maximizes some match-reliability metric (Veksler 2001, 2003);
- weighting pixels by their similarity to the central pixel (Yoon and Kweon 2006).

Once you have aggregated the costs in the DSI, pick the winner at each pixel (winner-take-all), and then optionally perform one or more of the following post-processing steps:

1. compute matches both ways and pick only the reliable matches (draw the others in another color);
2. tag matches that are unsure (whose confidence is too low);

3. fill in the matches that are unsure from neighboring values;

4. refine your matches to sub-pixel disparity by either fitting a parabola to the DSI values around the winner or by using an iteration of Lukas–Kanade.

Ex 11.6: Optimization-based stereo Compute the disparity space image (DSI) volume using one of the techniques you implemented in Exercise 11.4 and then implement one (or more) of the global optimization techniques described in Section 11.5 to compute the depth map. Potential choices include:

- dynamic programming or scanline optimization (relatively easy);
- semi-global optimization (Hirschmüller 2008), which is a simple extension of scanline optimization and performs well;
- graph cuts using alpha expansions (Boykov, Veksler, and Zabih 2001), for which you will need to find a max-flow or min-cut algorithm (http://vision.middlebury.edu/stereo);
- loopy belief propagation (Appendix B.5.3).

Evaluate your algorithm by running it on the Middlebury stereo data sets.

How well does your algorithm do against local aggregation (Yoon and Kweon 2006)? Can you think of some extensions or modifications to make it even better?

Ex 11.7: View interpolation, revisited Compute a dense depth map using one of the techniques you developed above and use it (or, better yet, a depth map for each source image) to generate smooth in-between views from a stereo data set.

Compare your results against using the ground truth depth data (if available).

What kinds of artifacts do you see? Can you think of ways to reduce them?

More details on implementing such algorithms can be found in Section 13.1 and Exercises 13.1–13.4.

Ex 11.8: Multi-frame stereo Extend one of your previous techniques to use multiple input frames (Section 11.6) and try to improve the results you obtained with just two views.

If helpful, try using temporal selection (Kang and Szeliski 2004) to deal with the increased number of occlusions in multi-frame data sets.

You can also try to simultaneously estimate multiple depth maps and make them consistent (Kolmogorov and Zabih 2002; Kang and Szeliski 2004).

Test your algorithms out on some standard multi-view data sets.

Ex 11.9: Volumetric stereo Implement voxel coloring (Seitz and Dyer 1999) as a simple extension to the plane sweep algorithm you implemented in Exercise 11.4.

1. Instead of computing the complete DSI all at once, evaluate each plane one at a time from front to back.

2. Tag every voxel whose photoconsistency is below a certain threshold as being part of the object and remember its average (or robust) color (Seitz and Dyer 1999; Eisert, Steinbach, and Girod 2000; Kutulakos 2000; Slabaugh, Culbertson, Slabaugh *et al.* 2004).

3. Erase the input pixels corresponding to tagged voxels in the input images, e.g., by setting their alpha value to 0 (or to some reduced number, depending on occupancy).

4. As you evaluate the next plane, use the source image alpha values to modify your photoconsistency score, e.g., only consider pixels that have full alpha or weight pixels by their alpha values.

5. If the cameras are not all on the same side of your plane sweeps, use space carving (Kutulakos and Seitz 2000) to cycle through different subsets of source images while carving away the volume from different directions.

Ex 11.10: Depth map merging Use the technique you developed for multi-frame stereo in Exercise 11.8 or a different technique, such as the one described by Goesele, Snavely, Curless *et al.* (2007), to compute a depth map for every input image.

Merge these depth maps into a coherent 3D model, e.g., using Poisson surface reconstruction (Kazhdan, Bolitho, and Hoppe 2006).

Ex 11.11: Shape from silhouettes Build a silhouette-based volume reconstruction algorithm (Section 11.6.2). Use an octree or some other representation of your choosing.

Chapter 12

3D reconstruction

12.1 Shape from X 508
12.1.1 Shape from shading and photometric stereo 508
12.1.2 Shape from texture 510
12.1.3 Shape from focus 511
12.2 Active rangefinding 512
12.2.1 Range data merging 515
12.2.2 *Application*: Digital heritage 517
12.3 Surface representations 518
12.3.1 Surface interpolation 518
12.3.2 Surface simplification 520
12.3.3 Geometry images 520
12.4 Point-based representations 521
12.5 Volumetric representations 522
12.5.1 Implicit surfaces and level sets 522
12.6 Model-based reconstruction 523
12.6.1 Architecture 524
12.6.2 Heads and faces 526
12.6.3 *Application*: Facial animation 528
12.6.4 Whole body modeling and tracking 530
12.7 Recovering texture maps and albedos 534
12.7.1 Estimating BRDFs 536
12.7.2 *Application*: 3D photography 537
12.8 Additional reading 538
12.9 Exercises 539

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_12, © Springer-Verlag London Limited 2011

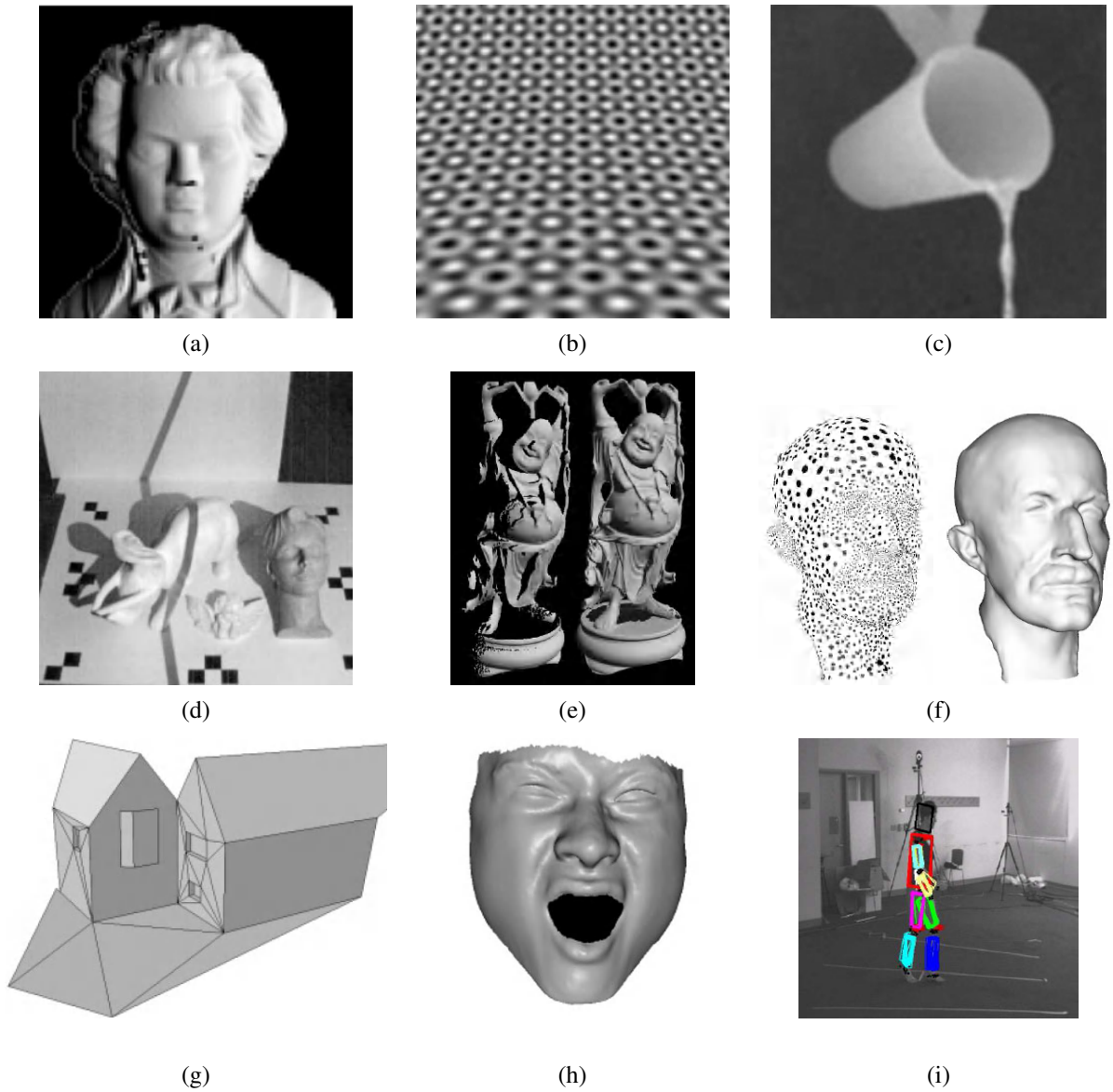


Figure 12.1 3D shape acquisition and modeling techniques: (a) shaded image (Zhang, Tsai, Cryer *et al.* 1999) © 1999 IEEE; (b) texture gradient (Garding 1992) © 1992 Springer; (c) real-time depth from focus (Nayar, Watanabe, and Noguchi 1996) © 1996 IEEE; (d) scanning a scene with a stick shadow (Bouguet and Perona 1999) © 1999 Springer; (e) merging range maps into a 3D model (Curless and Levoy 1996) © 1996 ACM; (f) point-based surface modeling (Pauly, Keiser, Kobbelt *et al.* 2003) © 2003 ACM; (g) automated modeling of a 3D building using lines and planes (Werner and Zisserman 2002) © 2002 Springer; (h) 3D face model from spacetime stereo (Zhang, Snavely, Curless *et al.* 2004) © 2004 ACM; (i) person tracking (Sigal, Bhatia, Roth *et al.* 2004) © 2004 IEEE.

As we saw in the previous chapter, a variety of stereo matching techniques have been developed to reconstruct high quality 3D models from two or more images. However, stereo is just one of the many potential cues that can be used to infer shape from images. In this chapter, we investigate a number of such techniques, which include not only visual cues such as shading and focus, but also techniques for merging multiple range or depth images into 3D models, as well as techniques for reconstructing specialized models, such as heads, bodies, or architecture.

Among the various cues that can be used to infer shape, the shading on a surface (Figure 12.1a) can provide a lot of information about local surface orientations and hence overall surface shape (Section 12.1.1). This approach becomes even more powerful when lights shining from different directions can be turned on and off separately (*photometric stereo*). Texture gradients (Figure 12.1b), i.e., the foreshortening of regular patterns as the surface slants or bends away from the camera, can provide similar cues on local surface orientation (Section 12.1.2). Focus is another powerful cue to scene depth, especially when two or more images with different focus settings are used (Section 12.1.3).

3D shape can also be estimated using active illumination techniques such as light stripes (Figure 12.1d) or time of flight range finders (Section 12.2). The partial surface models obtained using such techniques (or passive image-based stereo) can then be merged into more coherent 3D surface models (Figure 12.1e), as discussed in Section 12.2.1. Such techniques have been used to construct highly detailed and accurate models of cultural heritage such as historic sites (Section 12.2.2). The resulting surface models can then be simplified to support viewing at different resolutions and streaming across the Web (Section 12.3.2). An alternative to working with continuous surfaces is to represent 3D surfaces as dense collections of 3D oriented points (Section 12.4) or as volumetric primitives (Section 12.5).

3D modeling can be more efficient and effective if we know something about the objects we are trying to reconstruct. In Section 12.6, we look at three specialized but commonly occurring examples, namely architecture (Figure 12.1g), heads and faces (Figure 12.1h), and whole bodies (Figure 12.1i). In addition to modeling people, we also discuss techniques for tracking them.

The last stage of shape and appearance modeling is to extract some textures to paint onto our 3D models (Section 12.7). Some techniques go beyond this and actually estimate full BRDFs (Section 12.7.1).

Because there exists such a large variety of techniques to perform 3D modeling, this chapter does not go into detail on any one of these. Readers are encouraged to find more information in the cited references or more specialized publications and conferences devoted to these topics, e.g., the International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT), the International Conference on 3D Digital Imaging and Modeling (3DIM), the International Conference on Automatic Face and Gesture Recognition (FG), the IEEE Workshop on Analysis and Modeling of Faces and Gestures, and the International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS).

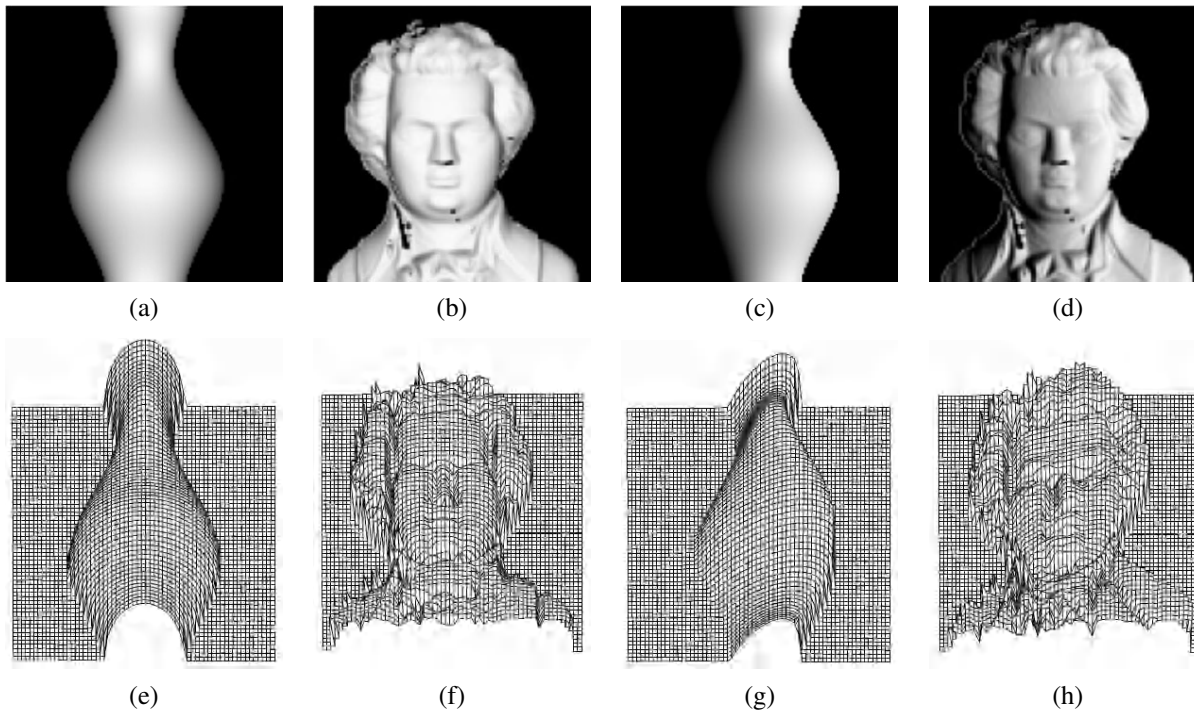


Figure 12.2 Synthetic shape from shading (Zhang, Tsai, Cryer *et al.* 1999) © 1999 IEEE: shaded images, (a–b) with light from in front $(0, 0, 1)$ and (c–d) with light the front right $(1, 0, 1)$; (e–f) corresponding shape from shading reconstructions using the technique of Tsai and Shah (1994).

12.1 Shape from X

In addition to binocular disparity, shading, texture, and focus all play a role in how we perceive shape. The study of how shape can be inferred from such cues is sometimes called *shape from X*, since the individual instances are called *shape from shading*, *shape from texture*, and *shape from focus*.[1] In this section, we look at these three cues and how they can be used to reconstruct 3D geometry. A good overview of all these topics can be found in the collection of papers on physics-based shape inference edited by Wolff, Shafer, and Healey (1992b).

12.1.1 Shape from shading and photometric stereo

When you look at images of smooth shaded objects, such as the ones shown in Figure 12.2, you can clearly see the shape of the object from just the shading variation. How is this possible? The answer is that as the surface normal changes across the object, the apparent brightness changes as a function of the angle between the local surface orientation and the incident illumination, as shown in Figure 2.15 (Section 2.2.2).

The problem of recovering the shape of a surface from this intensity variation is known as

[1] We have already seen examples of shape from stereo, shape from profiles, and shape from silhouettes in Chapter 11.

shape from shading and is one of the classic problems in computer vision (Horn 1975). The collection of papers edited by Horn and Brooks (1989) is a great source of information on this topic, especially the chapter on variational approaches. The survey by Zhang, Tsai, Cryer *et al.* (1999) not only reviews more recent techniques, but also provides some comparative results.

Most shape from shading algorithms assume that the surface under consideration is of a uniform albedo and reflectance, and that the light source directions are either known or can be calibrated by the use of a reference object. Under the assumptions of distant light sources and observer, the variation in intensity (*irradiance equation*) become purely a function of the local surface orientation,

$$I(x, y) = R(p(x, y), q(x, y)), \tag{12.1}$$

where $(p, q) = (z_x, z_y)$ are the depth map derivatives and $R(p, q)$ is called the *reflectance map*. For example, a diffuse (Lambertian) surface has a reflectance map that is the (non-negative) dot product (2.88) between the surface normal $\hat{\boldsymbol{n}} = (p, q, 1)/\sqrt{1 + p^2 + q^2}$ and the light source direction $\boldsymbol{v} = (v_x, v_y, v_z)$,

$$R(p, q) = \max\left(0, \rho \frac{pv_x + qv_y + v_z}{\sqrt{1 + p^2 + q^2}}\right), \tag{12.2}$$

where ρ is the surface reflectance factor (albedo).

In principle, Equations (12.1–12.2) can be used to estimate (p, q) using non-linear least squares or some other method. Unfortunately, unless additional constraints are imposed, there are more unknowns per pixel (p, q) than there are measurements (I). One commonly used constraint is the smoothness constraint,

$$\mathcal{E}_s = \int p_x^2 + p_y^2 + q_x^2 + q_y^2 \, dx \, dy = \int \|\nabla p\|^2 + \|\nabla q\|^2 \, dx \, dy, \tag{12.3}$$

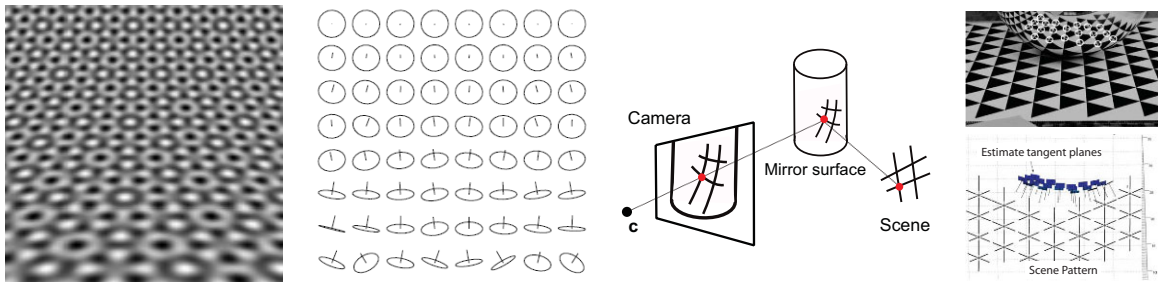
which we already saw in Section 3.7.1 (3.94). The other is the *integrability constraint*,

$$\mathcal{E}_i = \int (p_y - q_x)^2 \, dx \, dy, \tag{12.4}$$

which arises naturally, since for a valid depth map $z(x, y)$ with $(p, q) = (z_x, z_y)$, we have $p_y = z_{xy} = z_{yx} = q_x$.

Instead of first recovering the orientation fields (p, q) and integrating them to obtain a surface, it is also possible to directly minimize the discrepancy in the image formation equation (12.1) while finding the optimal depth map $z(x, y)$ (Horn 1990). Unfortunately, shape from shading is susceptible to local minima in the search space and, like other variational problems that involve the simultaneous estimation of many variables, can also suffer from slow convergence. Using multi-resolution techniques (Szeliski 1991a) can help accelerate the convergence, while using more sophisticated optimization techniques (Dupuis and Oliensis 1994) can help avoid local minima.

In practice, surfaces other than plaster casts are rarely of a single uniform albedo. Shape from shading therefore needs to be combined with some other technique or extended in some way to make it useful. One way to do this is to combine it with stereo matching (Fua and Leclerc 1995) or known texture (surface patterns) (White and Forsyth 2006). The stereo and texture components provide information in textured regions, while shape from shading helps fill in the information across uniformly colored regions and also provides finer information about surface shape.



(a) (b) (c) (d)

Figure 12.3 Synthetic shape from texture (Garding 1992) © 1992 Springer: (a) regular texture wrapped onto a curved surface and (b) the corresponding surface normal estimates. Shape from mirror reflections (Savarese, Chen, and Perona 2005) © 2005 Springer: (c) a regular pattern reflecting off a curved mirror gives rise to (d) curved lines, from which 3D point locations and normals can be inferred.

Photometric stereo. Another way to make shape from shading more reliable is to use multiple light sources that can be selectively turned on and off. This technique is called *photometric stereo*, since the light sources play a role analogous to the cameras located at different locations in traditional stereo (Woodham 1981).[2] For each light source, we have a different reflectance map, $R_1(p, q)$, $R_2(p, q)$, etc. Given the corresponding intensities I_1, I_2, etc. at a pixel, we can in principle recover both an unknown albedo ρ and a surface orientation estimate (p, q).

For diffuse surfaces (12.2), if we parameterize the local orientation by $\hat{\boldsymbol{n}}$, we get (for non-shadowed pixels) a set of linear equations of the form

$$I_k = \rho \hat{\boldsymbol{n}} \cdot \boldsymbol{v}_k, \tag{12.5}$$

from which we can recover $\rho\hat{\boldsymbol{n}}$ using linear least squares. These equations are well conditioned as long as the (three or more) vectors $\boldsymbol{v}_k$ are linearly independent, i.e., they are not along the same azimuth (direction away from the viewer).

Once the surface normals or gradients have been recovered at each pixel, they can be integrated into a depth map using a variant of regularized surface fitting (3.100). (Nehab, Rusinkiewicz, Davis *et al.* (2005) and Harker and O'Leary (2008) have produced some recent work in this area.)

When surfaces are specular, more than three light directions may be required. In fact, the irradiance equation given in (12.1) not only requires that the light sources and camera be distant from the surface, it also neglects inter-reflections, which can be a significant source of the shading observed on object surfaces, e.g., the darkening seen inside concave structures such as grooves and crevasses (Nayar, Ikeuchi, and Kanade 1991).

12.1.2 Shape from texture

The variation in foreshortening observed in regular textures can also provide useful information about local surface orientation. Figure 12.3 shows an example of such a pattern, along

[2] An alternative to turning lights on-and-off is to use three colored lights (Woodham 1994; Hernandez, Vogiatzis, Brostow *et al.* 2007; Hernandez and Vogiatzis 2010).

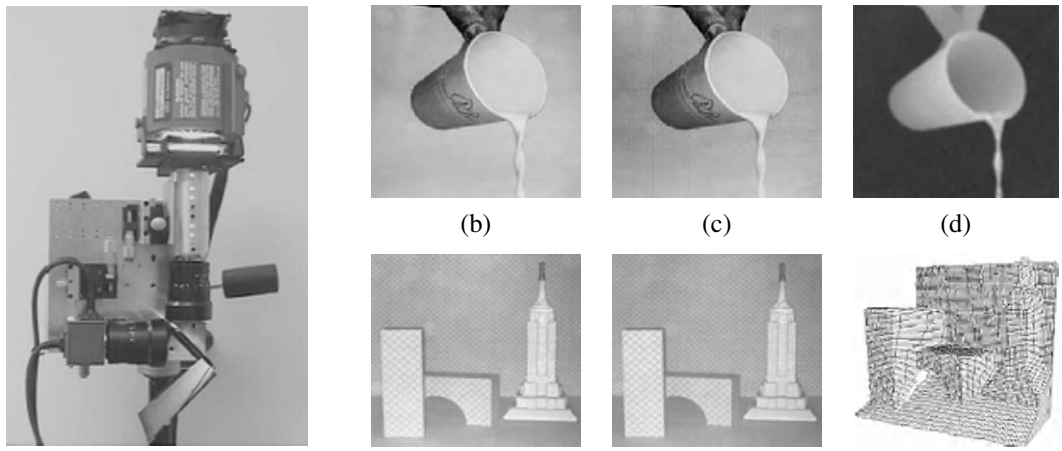


Figure 12.4 Real time depth from defocus (Nayar, Watanabe, and Noguchi 1996) © 1996 IEEE: (a) the real-time focus range sensor, which includes a half-silvered mirror between the two telecentric lenses (lower right), a prism that splits the image into two CCD sensors (lower left), and an edged checkerboard pattern illuminated by a Xenon lamp (top); (b–c) input video frames from the two cameras along with (d) the corresponding depth map; (e–f) two frames (you can see the texture if you zoom in) and (g) the corresponding 3D mesh model.

with the estimated local surface orientations. Shape from texture algorithms require a number of processing steps, including the extraction of repeated patterns or the measurement of local frequencies in order to compute local affine deformations, and a subsequent stage to infer local surface orientation. Details on these various stages can be found in the research literature (Witkin 1981; Ikeuchi 1981; Blostein and Ahuja 1987; Garding 1992; Malik and Rosenholtz 1997; Lobay and Forsyth 2006).

When the original pattern is regular, it is possible to fit a regular but slightly deformed grid to the image and use this grid for a variety of image replacement or analysis tasks (Liu, Collins, and Tsin 2004; Liu, Lin, and Hays 2004; Hays, Leordeanu, Efros *et al.* 2006; Lin, Hays, Wu *et al.* 2006; Park, Brocklehurst, Collins *et al.* 2009). This process becomes even easier if specially printed textured cloth patterns are used (White and Forsyth 2006; White, Crane, and Forsyth 2007).

The deformations induced in a regular pattern when it is viewed in the reflection of a curved mirror, as shown in Figure 12.3c–d, can be used to recover the shape of the surface (Savarese, Chen, and Perona 2005; Rozenfeld, Shimshoni, and Lindenbaum 2007). It is is also possible to infer local shape information from *specular flow*, i.e., the motion of specularities when viewed from a moving camera (Oren and Nayar 1997; Zisserman, Giblin, and Blake 1989; Swaminathan, Kang, Szeliski *et al.* 2002).

12.1.3 Shape from focus

A strong cue for object depth is the amount of blur, which increases as the object's surface moves away from the camera's focusing distance. As shown in Figure 2.19, moving the object surface away from the focus plane increases the circle of confusion, according to a formula that is easy to establish using similar triangles (Exercise 2.4).

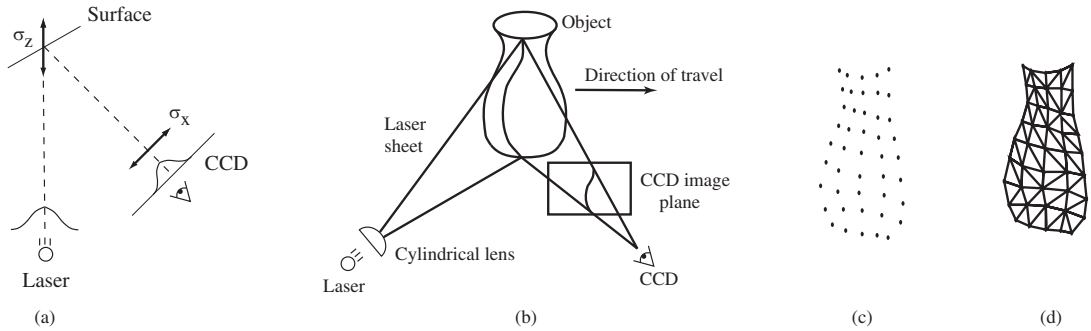


Figure 12.5 Range data scanning (Curless and Levoy 1996) © 1996 ACM: (a) a laser dot on a surface is imaged by a CCD sensor; (b) a laser stripe (sheet) is imaged by the sensor (the deformation of the stripe encodes the distance to the object); (c) the resulting set of 3D points are turned into (d) a triangulated mesh.

A number of techniques have been developed to estimate depth from the amount of defocus (*depth from defocus*) (Pentland 1987; Nayar and Nakagawa 1994; Nayar, Watanabe, and Noguchi 1996; Watanabe and Nayar 1998; Chaudhuri and Rajagopalan 1999; Favaro and Soatto 2006). In order to make such a technique practical, a number issues need to be addressed:

- The amount of blur increase in *both* directions as you move away from the focus plane. Therefore, it is necessary to use two or more images captured with different focus distance settings (Pentland 1987; Nayar, Watanabe, and Noguchi 1996) or to translate the object in depth and look for the point of maximum sharpness (Nayar and Nakagawa 1994).
- The magnification of the object can vary as the focus distance is changed or the object is moved. This can be modeled either explicitly (making correspondence more difficult) or using *telecentric optics*, which approximate an orthographic camera and require an aperture in front of the lens (Nayar, Watanabe, and Noguchi 1996).
- The amount of defocus must be reliably estimated. A simple approach is to average the squared gradient in a region but this suffers from several problems, including the image magnification problem mentioned above. A better solution is to use carefully designed *rational filters* (Watanabe and Nayar 1998).

Figure 12.4 shows an example of a real-time depth from defocus sensor, which employs two imaging chips at slightly different depths sharing a common optical path, as well as an active illumination system that projects a checkerboard pattern from the same direction. As you can see in Figure 12.4b–g, the system produces high-accuracy real-time depth maps for both static and dynamic scenes.

12.2 Active rangefinding

As we have seen in the previous section, actively lighting a scene, whether for the purpose of estimating normals using photometric stereo or for adding artificial texture for shape from



(a) (b) (c)

Figure 12.6 Shape scanning using cast shadows (Bouguet and Perona 1999) © 1999 Springer: (a) camera setup with a point light source (a desk lamp without its reflector), a hand-held stick casting a shadow, and (b) the objects being scanned in front of two planar backgrounds. (c) Real-time depth map using a pulsed illumination system (Iddan and Yahav 2001) © 2001 SPIE.

defocus, can greatly improve the performance of vision systems. This kind of *active illumination* has been used from the earliest days of machine vision to construct highly reliable sensors for estimating 3D depth images using a variety of *rangefinding* (or *range sensing*) techniques (Besl 1989; Curless 1999; Hebert 2000).

One of the most popular active illumination sensors is a laser or light stripe sensor, which sweeps a plane of light across the scene or object while observing it from an offset viewpoint, as shown in Figure 12.5b (Rioux and Bird 1993; Curless and Levoy 1995). As the stripe falls across the object, it deforms its shape according to the shape of the surface it is illuminating. It is then a simple matter of using *optical triangulation* to estimate the 3D locations of all the points seen in a particular stripe. In more detail, knowledge of the 3D plane equation of the light stripe allows us to infer the 3D location corresponding to each illuminated pixel, as previously discussed in (2.70–2.71). The accuracy of light striping techniques can be improved by finding the exact temporal peak in illumination for each pixel (Curless and Levoy 1995). The final accuracy of a scanner can be determined using slant edge modulation techniques, i.e., by imaging sharp creases in a calibration object (Goesele, Fuchs, and Seidel 2003).

An interesting variant on light stripe rangefinding is presented by Bouguet and Perona (1999). Instead of projecting a light stripe, they simply wave a stick casting a shadow over a scene or object illuminated by a point light source such as a lamp or the sun (Figure 12.6a). As the shadow falls across two background planes whose orientation relative to the camera is known (or inferred during pre-calibration), the plane equation for each stripe can be inferred from the two projected lines, whose 3D equations are known (Figure 12.6b). The deformation of the shadow as it crosses the object being scanned then reveals its 3D shape, as with regular light stripe rangefinding (Exercise 12.2). This technique can also be used to estimate the 3D geometry of a background scene and how its appearance varies as it moves into shadow, in order to cast new shadows onto the scene (Chuang, Goldman, Curless *et al.* 2003) (Section 10.4.3).

The time it takes to scan an object using a light stripe technique is proportional to the number of depth planes used, which is usually comparable to the number of pixels across an image. A much faster scanner can be constructed by turning different projector pixels on



(a) (b)

Figure 12.7 Real-time dense 3D face capture using spacetime stereo (Zhang, Snavely, Curless *et al.* 2004) © 2004 ACM: (a) set of five consecutive video frames from one of two stereo cameras (every fifth frame is free of stripe patterns, in order to extract texture); (b) resulting high-quality 3D surface model (depth map visualized as a shaded rendering).

and off in a structured manner, e.g., using a binary or Gray code (Besl 1989). For example, let us assume that the LCD projector we are using has 1024 columns of pixels. Taking the 10-bit binary code corresponding to each column's address $(0 \ldots 1023)$, we project the first bit, then the second, etc. After 10 projections (e.g., a third of a second for a synchronized 30Hz camera-projector system), each pixel in the camera knows which of the 1024 columns of projector light it is seeing. A similar approach can also be used to estimate the refractive properties of an object by placing a monitor behind the object (Zongker, Werner, Curless *et al.* 1999; Chuang, Zongker, Hindorff *et al.* 2000) (Section 13.4). Very fast scanners can also be constructed with a single laser beam, i.e., a real-time *flying spot* optical triangulation scanner (Rioux, Bechthold, Taylor *et al.* 1987).

If even faster, i.e., frame-rate, scanning is required, we can project a single textured pattern into the scene. Proesmans, Van Gool, and Defoort (1998) describe a system where a checkerboard grid is projected onto an object (e.g., a person's face) and the deformation of the grid is used to infer 3D shape. Unfortunately, such a technique only works if the surface is continuous enough to link all of the grid points.

A much better system can be constructed using high-speed custom illumination and sensing hardware. Iddan and Yahav (2001) describe the construction of their 3DV Zcam video-rate depth sensing camera, which projects a pulsed plane of light onto the scene and then integrates the returning light for a short interval, essentially obtaining time-of-flight measurement for the distance to individual pixels in the scene. A good description of earlier time-of-flight systems, including amplitude and frequency modulation schemes for LIDAR, can be found in (Besl 1989).

Instead of using a single camera, it is also possible to construct an active illumination range sensor using stereo imaging setups. The simplest way to do this is to just project random stripe patterns onto the scene to create synthetic texture, which helps match textureless surfaces (Kang, Webb, Zitnick *et al.* 1995). Projecting a known series of stripes, just as in coded pattern single-camera rangefinding, makes the correspondence between pixels unambiguous and allows for the recovery of depth estimates at pixels only seen in a single camera (Scharstein and Szeliski 2003). This technique has been used to produce large numbers of highly accurate registered multi-image stereo pairs and depth maps for the purpose of evaluating stereo correspondence algorithms (Scharstein and Szeliski 2002; Hirschmüller and

Scharstein 2009) and learning depth map priors and parameters (Scharstein and Pal 2007).

While projecting multiple patterns usually requires the scene or object to remain still, additional processing can enable the production of real-time depth maps for dynamic scenes. The basic idea (Davis, Ramamoorthi, and Rusinkiewicz 2003; Zhang, Curless, and Seitz 2003) is to assume that depth is nearly constant within a 3D space–time window around each pixel and to use the 3D window for matching and reconstruction. Depending on the surface shape and motion, this assumption may be error-prone, as shown in (Davis, Nahab, Ramamoorthi *et al.* 2005). To model shapes more accurately, Zhang, Curless, and Seitz (2003) model the linear disparity variation within the space–time window and show that better results can be obtained by globally optimizing disparity and disparity gradient estimates over video volumes (Zhang, Snavely, Curless *et al.* 2004). Figure 12.7 shows the results of applying this system to a person's face; the frame-rate 3D surface model can then be used for further model-based fitting and computer graphics manipulation (Section 12.6.2).

12.2.1 Range data merging

While individual range images can be useful for applications such as real-time z-keying or facial motion capture, they are often used as building blocks for more complete 3D object modeling. In such applications, the next two steps in processing are the registration (alignment) of partial 3D surface models and their integration into coherent 3D surfaces (Curless 1999). If desired, this can be followed by a model fitting stage using either parametric representations such as generalized cylinders (Agin and Binford 1976; Nevatia and Binford 1977; Marr and Nishihara 1978; Brooks 1981), superquadrics (Pentland 1986; Solina and Bajcsy 1990; Terzopoulos and Metaxas 1991), or non-parametric models such as triangular meshes (Boissonat 1984) or physically-based models (Terzopoulos, Witkin, and Kass 1988; Delingette, Hebert, and Ikeuichi 1992; Terzopoulos and Metaxas 1991; McInerney and Terzopoulos 1993; Terzopoulos 1999). A number of techniques have also been developed for segmenting range images into simpler constituent surfaces (Hoover, Jean-Baptiste, Jiang *et al.* 1996).

The most widely used 3D registration technique is the *iterated closest point* (ICP) algorithm, which alternates between finding the closest point matches between the two surfaces being aligned and then solving a 3D *absolute orientation* problem (Section 6.1.5, (6.31–6.32) (Besl and McKay 1992; Chen and Medioni 1992; Zhang 1994; Szeliski and Lavallée 1996; Gold, Rangarajan, Lu *et al.* 1998; David, DeMenthon, Duraiswami *et al.* 2004; Li and Hartley 2007; Enqvist, Josephson, and Kahl 2009).[3] Since the two surfaces being aligned usually only have partial overlap and may also have outliers, robust matching criteria (Section 6.1.4 and Appendix B.3) are typically used. In order to speed up the determination of the closest point, and also to make the distance-to-surface computation more accurate, one of the two point sets (e.g., the current merged model) can be converted into a *signed distance function*, optionally represented using an *octree spline* for compactness (Lavallée and Szeliski 1995). Variants on the basic ICP algorithm can be used to register 3D point sets under non-rigid deformations, e.g., for medical applications (Feldmar and Ayache 1996; Szeliski and Lavallée 1996). Color values associated with the points or range measurements can also be used as part of the registration process to improve robustness (Johnson and Kang 1997; Pulli 1999).

[3] Some techniques, such as the one developed by Chen and Medioni (1992), use local surface tangent planes to make this computation more accurate and to accelerate convergence.

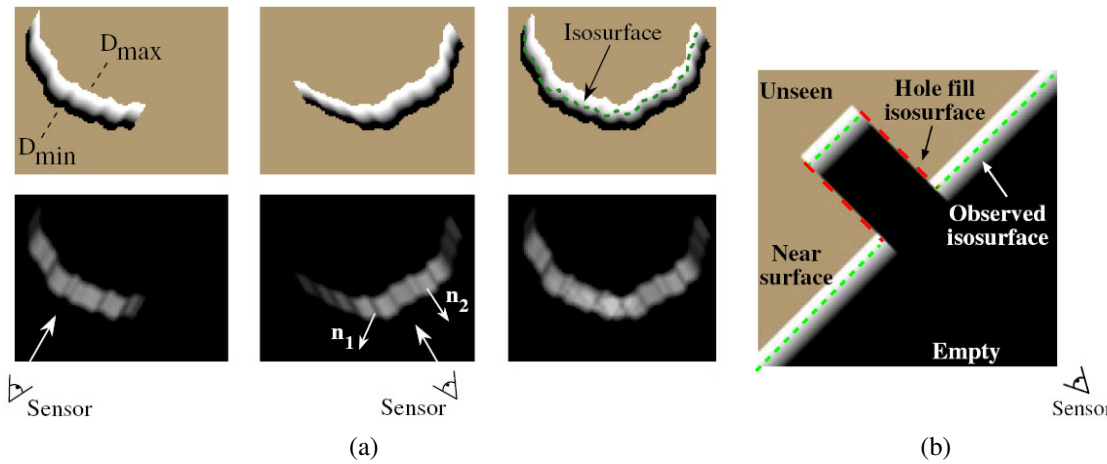


Figure 12.8 Range data merging (Curless and Levoy 1996) © 1996 ACM: (a) two signed distance functions (top left) are merged with their (weights) bottom left to produce a combined set of functions (right column) from which an isosurface can be extracted (green dashed line); (b) the signed distance functions are combined with empty and unseen space labels to fill holes in the isosurface.

Unfortunately, the ICP algorithm and its variants can only find a locally optimal alignment between 3D surfaces. If this is not known *a priori*, more global correspondence or search techniques, based on local descriptors invariant to 3D rigid transformations, need to be used. An example of such a descriptor is the *spin image*, which is a local circular projection of a 3D surface patch around the local normal axis (Johnson and Hebert 1999). Another (earlier) example is the *splash* representation introduced by Stein and Medioni (1992).

Once two or more 3D surfaces have been aligned, they can be merged into a single model. One approach is to represent each surface using a triangulated mesh and combine these meshes using a process that is sometimes called *zippering* (Soucy and Laurendeau 1992; Turk and Levoy 1994). Another, now more widely used, approach is to compute a signed distance function that fits all of the 3D data points (Hoppe, DeRose, Duchamp *et al.* 1992; Curless and Levoy 1996; Hilton, Stoddart, Illingworth *et al.* 1996; Wheeler, Sato, and Ikeuchi 1998).

Figure 12.8 shows one such approach, the *volumetric range image processing* (VRIP) technique developed by Curless and Levoy (1996), which first computes a weighted signed distance function from each range image and then merges them using a weighted averaging process. To make the representation more compact, run-length coding is used to encode the empty, seen, and varying (signed distance) voxels, and only the signed distance values near each surface are stored.[4] Once the merged signed distance function has been computed, a zero-crossing surface extraction algorithm, such as *marching cubes* (Lorensen and Cline 1987), can be used to recover a meshed surface model. Figure 12.9 shows an example of the complete range data merging and isosurface extraction pipeline.

Volumetric range data merging techniques based on signed distance or characteristic (inside–outside) functions are also widely used to extract smooth well-behaved surfaces from oriented or unoriented sets of points (Hoppe, DeRose, Duchamp *et al.* 1992; Ohtake, Belyaev,

[4] An alternative, even more compact, representation could be to use octrees (Lavallée and Szeliski 1995).



(a) (b) (c) (d) (e)

Figure 12.9 Reconstruction and hardcopy of the "Happy Buddha" statuette (Curless and Levoy 1996) © 1996 ACM: (a) photograph of the original statue after spray painting with matte gray; (b) partial range scan; (c) merged range scans; (d) colored rendering of the reconstructed model; (e) hardcopy of the model constructed using stereolithography.

Alexa *et al.* 2003; Kazhdan, Bolitho, and Hoppe 2006; Lempitsky and Boykov 2007; Zach, Pock, and Bischof 2007b; Zach 2008), as discussed in more detail in Section 12.5.1.

12.2.2 *Application*: Digital heritage

Active rangefinding technologies, combined with surface modeling and appearance modeling techniques (Section 12.7), are widely used in the fields of archeological and historical preservation, which often also goes under the name *digital heritage* (MacDonald 2006). In such applications, detailed 3D models of cultural objects are acquired and later used for applications such as analysis, preservation, restoration, and the production of duplicate artwork (Rioux and Bird 1993).

A more recent example of such an endeavor is the Digital Michelangelo project of Levoy, Pulli, Curless *et al.* (2000), which used Cyberware laser stripe scanners and high-quality digital SLR cameras mounted on a large gantry to obtain detailed scans of Michelangelo's David and other sculptures in Florence. The project also took scans of the *Forma Urbis Romae*, an ancient stone map of Rome that had shattered into pieces, for which new matches were obtained using digital techniques. The whole process, from initial planning, to software development, acquisition, and post-processing, took several years (and many volunteers), and produced a wealth of 3D shape and appearance modeling techniques as a result.

Even larger-scale projects are now being attempted, for example, the scanning of complete temple sites such as Angkor-Thom (Ikeuchi and Sato 2001; Ikeuchi and Miyazaki 2007; Banno, Masuda, Oishi *et al.* 2008). Figure 12.10 shows details from this project, including a sample photograph, a detailed 3D (sculptural) head model scanned from ground level, and an aerial overview of the final merged 3D site model, which was acquired using a balloon.



(a) (b) (c)

Figure 12.10 Laser range modeling of the Bayon temple at Angkor-Thom (Banno, Masuda, Oishi *et al.* 2008) © 2008 Springer: (a) sample photograph from the site; (b) a detailed head model scanned from the ground; (c) final merged 3D model of the temple scanned using a laser range sensor mounted on a balloon.

12.3 Surface representations

In previous sections, we have seen different representations being used to integrate 3D range scans. We now look at several of these representations in more detail. Explicit surface representations, such as triangle meshes, splines (Farin 1992, 1996), and subdivision surfaces (Stollnitz, DeRose, and Salesin 1996; Zorin, Schröder, and Sweldens 1996; Warren and Weimer 2001; Peters and Reif 2008), enable not only the creation of highly detailed models but also processing operations, such as interpolation (Section 12.3.1), fairing or smoothing, and decimation and simplification (Section 12.3.2). We also examine discrete point-based representations (Section 12.4) and volumetric representations (Section 12.5).

12.3.1 Surface interpolation

One of the most common operations on surfaces is their reconstruction from a set of sparse data constraints, i.e. *scattered data interpolation*. When formulating such problems, surfaces may be parameterized as height fields $f(\boldsymbol{x})$, as 3D parametric surfaces $\boldsymbol{f}(\boldsymbol{x})$, or as non-parametric models such as collections of triangles.

In the section on image processing, we saw how two-dimensional function interpolation and approximation problems $\{d_i\} \rightarrow f(\boldsymbol{x})$ could be cast as energy minimization problems using regularization (Section 3.7.1 (3.94–3.98).[5] Such problems can also specify the locations of discontinuities in the surface as well as local orientation constraints (Terzopoulos 1986b; Zhang, Dugas-Phocion, Samson *et al.* 2002).

One approach to solving such problems is to discretize both the surface and the energy on a discrete grid or mesh using finite element analysis (3.100–3.102) (Terzopoulos 1986b). Such problems can then be solved using sparse system solving techniques, such as multigrid (Briggs, Henson, and McCormick 2000) or hierarchically preconditioned conjugate gradient (Szeliski 2006b). The surface can also be represented using a hierarchical combination of multilevel B-splines (Lee, Wolberg, and Shin 1996).

[5] The difference between interpolation and approximation is that the former requires the surface or function to pass through the data while the latter allows the function to pass near the data, and can therefore be used for surface smoothing as well.

An alternative approach is to use *radial basis* (or *kernel*) functions (Boult and Kender 1986; Nielson 1993). To interpolate a field $\boldsymbol{f}(\boldsymbol{x})$ through (or near) a number of data values $\boldsymbol{d}_i$ located at $\boldsymbol{x}_i$, the *radial basis function* approach uses

$$\boldsymbol{f}(\boldsymbol{x}) = \frac{\sum_i w_i(\boldsymbol{x})\boldsymbol{d}_i}{\sum_i w_i(\boldsymbol{x})}, \tag{12.6}$$

where the weights,

$$w_i(\boldsymbol{x}) = K(\|\boldsymbol{x} - \boldsymbol{x}_i\|), \tag{12.7}$$

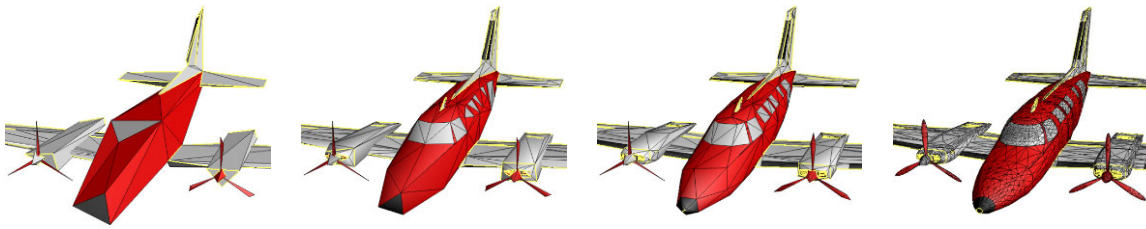
are computed using a *radial basis* (spherically symmetrical) function $K(r)$.

If we want the function $\boldsymbol{f}(\boldsymbol{x})$ to exactly interpolate the data points, the kernel functions must either be singular at the origin, $\lim_{r\to 0} K(r) \to \infty$ (Nielson 1993), or a dense linear system must be solved to determine the magnitude associated with each basis function (Boult and Kender 1986). It turns out that, for certain regularized problems, e.g., (3.94–3.96), there exist radial basis functions (kernels) that give the same results as a full analytical solution (Boult and Kender 1986). Unfortunately, because the dense system solving is cubic in the number of data points, basis function approaches can only be used for small problems such as feature-based image morphing (Beier and Neely 1992).

When a three-dimensional *parametric surface* is being modeled, the vector-valued function $\boldsymbol{f}$ in (12.6) or (3.94–3.102) encodes 3D coordinates (x, y, z) on the surface and the domain $\boldsymbol{x} = (s, t)$ encodes the surface parameterization. One example of such surfaces are symmetry-seeking parametric models, which are elastically deformable versions of *generalized cylinders*[6] (Terzopoulos, Witkin, and Kass 1987). In these models, s is the parameter *along* the spine of the deformable tube and t is the parameter *around* the tube. A variety of smoothness and radial symmetry forces are used to constrain the model while it is fitted to image-based silhouette curves.

It is also possible to define *non-parametric* surface models such as general triangulated meshes and to equip such meshes (using finite element analysis) with both internal smoothness metrics and external data fitting metrics (Sander and Zucker 1990; Fua and Sander 1992; Delingette, Hebert, and Ikeuichi 1992; McInerney and Terzopoulos 1993). While most of these approaches assume a standard *elastic* deformation model, which uses quadratic internal smoothness terms, it is also possible to use sub-linear energy models in order to better preserve surface creases (Diebel, Thrun, and Brünig 2006). Triangle meshes can also be augmented with either spline elements (Sullivan and Ponce 1998) or subdivision surfaces (Stollnitz, DeRose, and Salesin 1996; Zorin, Schröder, and Sweldens 1996; Warren and Weimer 2001; Peters and Reif 2008) to produce surfaces with better smoothness control.

Both parametric and non-parametric surface models assume that the topology of the surface is known and fixed ahead of time. For more flexible surface modeling, we can either represent the surface as a collection of oriented points (Section 12.4) or use 3D implicit functions (Section 12.5.1), which can also be combined with elastic 3D surface models (McInerney and Terzopoulos 1993).



(a) (b) (c) (d)

Figure 12.11 Progressive mesh representation of an airplane model (Hoppe 1996) © 1996 ACM: (a) base mesh M^0 (150 faces); (b) mesh M^{175} (500 faces); (c) mesh M^{425} (1000 faces); (d) original mesh $M = M^n$ (13,546 faces).

12.3.2 Surface simplification

Once a triangle mesh has been created from 3D data, it is often desirable to create a hierarchy of mesh models, for example, to control the displayed *level of detail* (LOD) in a computer graphics application. (In essence, this is a 3D analog to image pyramids (Section 3.5).) One approach to doing this is to approximate a given mesh with one that has *subdivision connectivity*, over which a set of triangular wavelet coefficients can then be computed (Eck, DeRose, Duchamp *et al.* 1995). A more continuous approach is to use sequential *edge collapse* operations to go from the original fine-resolution mesh to a coarse base-level mesh (Hoppe 1996). The resulting *progressive mesh* (PM) representation can be used to render the 3D model at arbitrary levels of detail, as shown in Figure 12.11.

12.3.3 Geometry images

While multi-resolution surface representations such as (Eck, DeRose, Duchamp *et al.* 1995; Hoppe 1996) support level of detail operations, they still consist of an irregular collection of triangles, which makes them more difficult to compress and store in a cache-efficient manner.[7]

To make the triangulation completely regular (uniform and gridded), Gu, Gortler, and Hoppe (2002) describe how to create *geometry images* by cutting surface meshes along well-chosen lines and "flattening" the resulting representation into a square. Figure 12.12a shows the resulting (x, y, z) values of the surface mesh mapped over the unit square, while Figure 12.12b shows the associated (n_x, n_y, n_z) *normal map*, i.e., the surface normals associated with each mesh vertex, which can be used to compensate for loss in visual fidelity if the original geometry image is heavily compressed.

[6] A generalized cylinder (Brooks 1981) is a *solid of revolution*, i.e., the result of rotating a (usually smooth) curve around an axis. It can also be generated by sweeping a slowly varying circular cross-section along the axis. (These two interpretations are equivalent.)

[7] Subdivision triangulations, such as those in (Eck, DeRose, Duchamp *et al.* 1995), are *semi-regular*, i.e., regular (ordered and nested) within each subdivided base triangle.

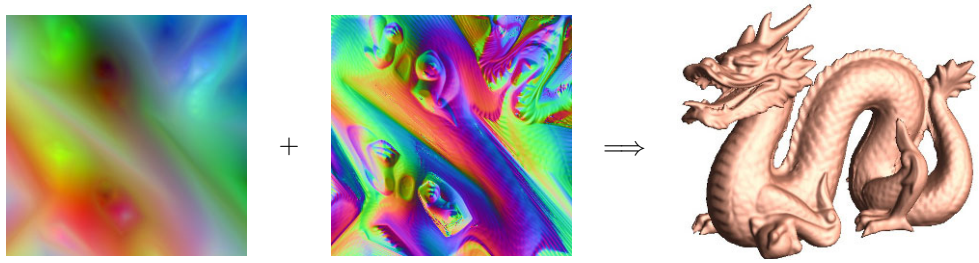


Figure 12.12 Geometry images (Gu, Gortler, and Hoppe 2002) © 2002 ACM: (a) the 257×257 geometry image defines a mesh over the surface; (b) the 512×512 normal map defines vertex normals; (c) final lit 3D model.

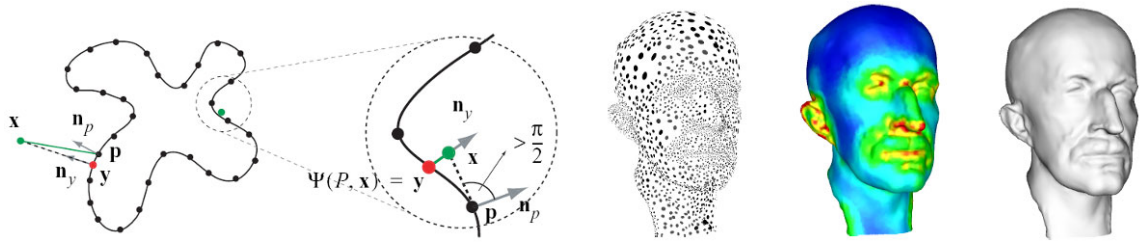


Figure 12.13 Point-based surface modeling with moving least squares (MLS) (Pauly, Keiser, Kobbelt *et al.* 2003) © 2003 ACM: (a) a set of points (black dots) is turned into an implicit inside–outside function (black curve); (b) the signed distance to the nearest oriented point can serve as an approximation to the inside–outside distance; (c) a set of oriented points with variable sampling density representing a 3D surface (head model); (d) local estimate of sampling density, which is used in the moving least squares; (e) reconstructed continuous 3D surface.

12.4 Point-based representations

As we mentioned previously, triangle-based surface models assume that the topology (and often the rough shape) of the 3D model is known ahead of time. While it is possible to re-mesh a model as it is being deformed or fitted, a simpler solution is to dispense with an explicit triangle mesh altogether and to have triangle vertices behave as *oriented points*, or particles, or *surface elements* (surfels) (Szeliski and Tonnesen 1992).

In order to endow the resulting particle system with internal smoothness constraints, pairwise interaction potentials can be defined that approximate the equivalent elastic bending energies that would be obtained using local finite-element analysis.[8] Instead of defining the finite element neighborhood for each particle (vertex) ahead of time, a soft influence function is used to couple nearby particles. The resulting 3D model can change both topology and particle density as it evolves and can therefore be used to interpolate partial 3D data with holes (Szeliski, Tonnesen, and Terzopoulos 1993b). Discontinuities in both the surface orientation and crease curves can also be modeled (Szeliski, Tonnesen, and Terzopoulos 1993a).

[8] As mentioned before, an alternative is to use *sub-linear* interaction potentials, which encourage the preservation of surface creases (Diebel, Thrun, and Brünig 2006).

To render the particle system as a continuous surface, local dynamic triangulation heuristics (Szeliski and Tonnesen 1992) or direct surface element *splatting* (Pfister, Zwicker, van Baar *et al.* 2000) can be used. Another alternative is to first convert the point cloud into an implicit signed distance or inside–outside function, using either minimum signed distances to the oriented points (Hoppe, DeRose, Duchamp *et al.* 1992) or by interpolating a characteristic (inside–outside) function using radial basis functions (Turk and O'Brien 2002; Dinh, Turk, and Slabaugh 2002). Even greater precision over the implicit function fitting, including the ability to handle irregular point densities, can be obtained by computing a *moving least squares* (MLS) estimate of the signed distance function (Alexa, Behr, Cohen-Or *et al.* 2003; Pauly, Keiser, Kobbelt *et al.* 2003), as shown in Figure 12.13. Further improvements can be obtained using local sphere fitting (Guennebaud and Gross 2007), faster and more accurate re-sampling (Guennebaud, Germann, and Gross 2008), and kernel regression to better tolerate outliers (Oztireli, Guennebaud, and Gross 2008).

12.5 Volumetric representations

A third alternative for modeling 3D surfaces is to construct 3D volumetric inside–outside functions. We already saw examples of this in Section 11.6.1, where we looked at voxel coloring (Seitz and Dyer 1999), space carving (Kutulakos and Seitz 2000), and level set (Faugeras and Keriven 1998; Pons, Keriven, and Faugeras 2007) techniques for stereo matching, and Section 11.6.2, where we discussed using binary silhouette images to reconstruct volumes.

In this section, we look at continuous *implicit* (inside–outside) functions to represent 3D shape.

12.5.1 Implicit surfaces and level sets

While polyhedral and voxel-based representations can represent three-dimensional shapes to an arbitrary precision, they lack some of the intrinsic smoothness properties available with continuous implicit surfaces, which use an *indicator function* (*characteristic function*) $F(x, y, z)$ to indicate which 3D points are inside $F(x, y, z) < 0$ or outside $F(x, y, z) > 0$ the object.

An early example of using implicit functions to model 3D objects in computer vision are superquadrics, which are a generalization of quadric (e.g., ellipsoidal) parametric volumetric models,

$$F(x, y, z) = \left(\left(\frac{x}{a_1} \right)^{2/\epsilon_2} + \left(\frac{y}{a_2} \right)^{2/\epsilon_2} \right)^{\epsilon_2/\epsilon_1} + \left(\frac{x}{a_1} \right)^{2/\epsilon_1} - 1 = 0 \tag{12.8}$$

(Pentland 1986; Solina and Bajcsy 1990; Waithe and Ferrie 1991; Leonardis, Jaklič, and Solina 1997). The values of (a_1, a_2, a_3) control the extent of model along each (x, y, z) axis, while the values of (ϵ_1, ϵ_2) control how "square" it is. To model a wider variety of shapes, superquadrics are usually combined with either rigid or non-rigid deformations (Terzopoulos and Metaxas 1991; Metaxas and Terzopoulos 2002). Superquadric models can either be fit to range data or used directly for stereo matching.

A different kind of implicit shape model can be constructed by defining a *signed distance function* over a regular three-dimensional grid, optionally using an octree spline to represent

this function more coarsely away from its surface (zero-set) (Lavallée and Szeliski 1995; Szeliski and Lavallée 1996; Frisken, Perry, Rockwood *et al.* 2000; Ohtake, Belyaev, Alexa *et al.* 2003). We have already seen examples of signed distance functions being used to represent distance transforms (Section 3.3.3), level sets for 2D contour fitting and tracking (Section 5.1.4), volumetric stereo (Section 11.6.1), range data merging (Section 12.2.1), and point-based modeling (Section 12.4). The advantage of representing such functions directly on a grid is that it is quick and easy to look up distance function values for any (x, y, z) location and also easy to extract the isosurface using the marching cubes algorithm (Lorensen and Cline 1987). The work of Ohtake, Belyaev, Alexa *et al.* (2003) is particularly notable since it allows for several distance functions to be used simultaneously and then combined locally to produce sharp features such as creases.

Poisson surface reconstruction (Kazhdan, Bolitho, and Hoppe 2006) uses a closely related volumetric function, namely a smoothed 0/1 inside–outside (characteristic) function, which can be thought of as a clipped signed distance function. The gradients for this function are set to lie along oriented surface normals near known surface points and 0 elsewhere. The function itself is represented using a quadratic tensor-product B-spline over an octree, which provides a compact representation with larger cells away from the surface or in regions of lower point density, and also admits the efficient solution of the related Poisson equations (3.100–3.102), see Section 9.3.4 (Pérez, Gangnet, and Blake 2003).

It is also possible to replace the quadratic penalties used in the Poisson equations with L_1 (total variation) constraints and still obtain a convex optimization problem, which can be solved using either continuous (Zach, Pock, and Bischof 2007b; Zach 2008) or discrete graph cut (Lempitsky and Boykov 2007) techniques.

Signed distance functions also play an integral role in level-set evolution equations ((Sections 5.1.4 and 11.6.1), where the values of distance transforms on the mesh are updated as the surface evolves to fit multi-view stereo photoconsistency measures (Faugeras and Keriven 1998).

12.6 Model-based reconstruction

When we know something ahead of time about the objects we are trying to model, we can construct more detailed and reliable 3D models using specialized techniques and representations. For example, architecture is usually made up of large planar regions and other parametric forms (such as surfaces of revolution), usually oriented perpendicular to gravity and to each other (Section 12.6.1). Heads and faces can be represented using low-dimensional, non-rigid shape models, since the variability in shape and appearance of human faces, while extremely large, is still bounded (Section 12.6.2). Human bodies or parts, such as hands, form highly articulated structures, which can be represented using kinematic chains of piecewise rigid skeletal elements linked by joints (Section 12.6.4).

In this section, we highlight some of the main ideas, representations, and modeling algorithms used for these three cases. Additional details and references can be found in specialized conferences and workshops devoted to these topics, e.g., the International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT), the International Conference on 3D Digital Imaging and Modeling (3DIM), the International Conference on Automatic Face and Gesture Recognition (FG), the IEEE Workshop on Analysis and Modeling of Faces



Figure 12.14 Interactive architectural modeling using the Façade system (Debevec, Taylor, and Malik 1996) © 1996 ACM: (a) input image with user-drawn edges shown in green; (b) shaded 3D solid model; (c) geometric primitives overlaid onto the input image; (d) final view-dependent, texture-mapped 3D model.

and Gestures, and the International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS).

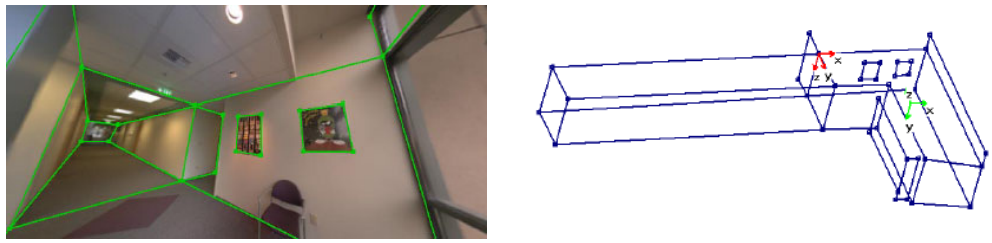
12.6.1 Architecture

Architectural modeling, especially from aerial photography, has been one of the longest studied problems in both photogrammetry and computer vision (Walker and Herman 1988). Recently, the development of reliable image-based modeling techniques, as well as the prevalence of digital cameras and 3D computer games, has spurred renewed interest in this area.

The work by Debevec, Taylor, and Malik (1996) was one of the earliest hybrid geometry- and image-based modeling and rendering systems. Their Façade system combines an interactive image-guided geometric modeling tool with model-based (local plane plus parallax) stereo matching and view-dependent texture mapping. During the interactive photogrammetric modeling phase, the user selects block elements and aligns their edges with visible edges in the input images (Figure 12.14a). The system then automatically computes the dimensions and locations of the blocks along with the camera positions using constrained optimization (Figure 12.14b–c). This approach is intrinsically more reliable than general feature-based structure from motion, because it exploits the strong geometry available in the block primitives. Related work by Becker and Bove (1995), Horry, Anjyo, and Arai (1997), and Criminisi, Reid, and Zisserman (2000) exploits similar information available from vanishing points. In the interactive, image-based modeling system of Sinha, Steedly, Szeliski *et al.* (2008), vanishing point directions are used to guide the user drawing of polygons, which are then automatically fitted to sparse 3D points recovered using structure from motion.

Once the rough geometry has been estimated, more detailed offset maps can be computed for each planar face using a local plane sweep, which Debevec, Taylor, and Malik (1996) call *model-based stereo*. Finally, during rendering, images from different viewpoints are warped and blended together as the camera moves around the scene, using a process (related to light field and Lumigraph rendering, see Section 13.3) called *view-dependent texture mapping* (Figure 12.14d).

For interior modeling, instead of working with single pictures, it is more useful to work



(a) (b)

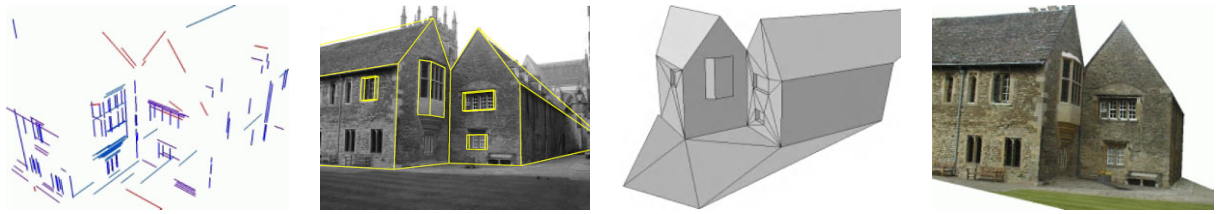
Figure 12.15 Interactive 3D modeling from panoramas (Shum, Han, and Szeliski 1998) © 1998 IEEE: (a) wide-angle view of a panorama with user-drawn vertical and horizontal (axis-aligned) lines; (b) single-view reconstruction of the corridors.

with panoramas, since you can see larger extents of walls and other structures. The 3D modeling system developed by Shum, Han, and Szeliski (1998) first constructs calibrated panoramas from multiple images (Section 7.4) and then has the user draw vertical and horizontal lines in the image to demarcate the boundaries of planar regions. The lines are initially used to establish an absolute rotation for each panorama and are later used (along with the inferred vertices and planes) to optimize the 3D structure, which can be recovered up to scale from one or more images (Figure 12.15). 360° high dynamic range panoramas can also be used for outdoor modeling, since they provide highly reliable estimates of relative camera orientations as well as vanishing point directions (Antone and Teller 2002; Teller, Antone, Bodnar *et al.* 2003).

While earlier image-based modeling systems required some user authoring, Werner and Zisserman (2002) present a fully automated line-based reconstruction system. As described in Section 7.5.1, they first detect lines and vanishing points and use them to calibrate the camera; then they establish line correspondences using both appearance matching and trifocal tensors, which enables them to reconstruct families of 3D line segments, as shown in Figure 12.16a. They then generate plane hypotheses, using both co-planar 3D lines and a plane sweep (Section 11.1.2) based on cross-correlation scores evaluated at interest points. Intersections of planes are used to determine the extent of each plane, i.e., an initial coarse geometry, which is then refined with the addition of rectangular or wedge-shaped indentations and extrusions (Figure 12.16c). Note that when top-down maps of the buildings being modeled are available, these can be used to further constrain the 3D modeling process (Robertson and Cipolla 2002, 2009). The idea of using matched 3D lines for estimating vanishing point directions and dominant planes continues to be used in a number of recent fully automated image-based architectural modeling systems (Zebedin, Bauer, Karner *et al.* 2008; Mičušík and Košecká 2009; Furukawa, Curless, Seitz *et al.* 2009b; Sinha, Steedly, and Szeliski 2009).

Another common characteristic of architecture is the repeated use of primitives such as windows, doors, and colonnades. Architectural modeling systems can be designed to search for such repeated elements and to use them as part of the structure inference process (Dick, Torr, and Cipolla 2004; Mueller, Zeng, Wonka *et al.* 2007; Schindler, Krishnamurthy, Lublinerman *et al.* 2008; Sinha, Steedly, Szeliski *et al.* 2008).

The combination of all these techniques now makes it possible to reconstruct the structure of large 3D scenes (Zhu and Kanade 2008). For example, the *Urbanscan* system of Polle-



(a) (b) (c) (d)

Figure 12.16 Automated architectural reconstruction using 3D lines and planes (Werner and Zisserman 2002) © 2002 Springer: (a) reconstructed 3D lines, color coded by their vanishing directions; (b) wire-frame model superimposed onto an input image; (c) triangulated piecewise-planar model with windows; (d) final texture-mapped model.

feys, Nistér, Frahm *et al.* (2008) reconstructs texture-mapped 3D models of city streets from videos acquired with a GPS-equipped vehicle. To obtain real-time performance, they use both optimized on-line structure-from-motion algorithms, as well as GPU implementations of plane-sweep stereo aligned to dominant planes and depth map fusion. Cornelis, Leibe, Cornelis *et al.* (2008) present a related system that also uses plane-sweep stereo (aligned to vertical building façades) combined with object recognition and segmentation for vehicles. Mičušík and Košecká (2009) build on these results using omni-directional images and super-pixel-based stereo matching along dominant plane orientations. Reconstruction directly from active range scanning data combined with color imagery that has been compensated for exposure and lighting variations is also possible (Chen and Chen 2008; Stamos, Liu, Chen *et al.* 2008; Troccoli and Allen 2008).

12.6.2 Heads and faces

Another area in which specialized shape and appearance models are extremely helpful is in the modeling of heads and faces. Even though the appearance of people seems at first glance to be infinitely variable, the actual shape of a person's head and face can be described reasonably well using a few dozen parameters (Pighin, Hecker, Lischinski *et al.* 1998; Guenter, Grimm, Wood *et al.* 1998; DeCarlo, Metaxas, and Stone 1998; Blanz and Vetter 1999; Shan, Liu, and Zhang 2001).

Figure 12.17 shows an example of an image-based modeling system, where user-specified keypoints in several images are used to fit a generic head model to a person's face. As you can see in Figure 12.17c, after specifying just over 100 keypoints, the shape of the face has become quite adapted and recognizable. Extracting a texture map from the original images and then applying it to the head model results in an animatable model with striking visual fidelity (Figure 12.18a).

A more powerful system can be built by applying *principal component analysis* (PCA) to a collection of 3D scanned faces, which is a topic we discuss in Section 12.6.3. As you can see in Figure 12.19, it is then possible to fit morphable 3D models to single images and to use such models for a variety of animation and visual effects (Blanz and Vetter 1999). It is also possible to design stereo matching algorithms that optimize directly for the head model parameters (Shan, Liu, and Zhang 2001; Kang and Jones 2002) or to use the output of real-

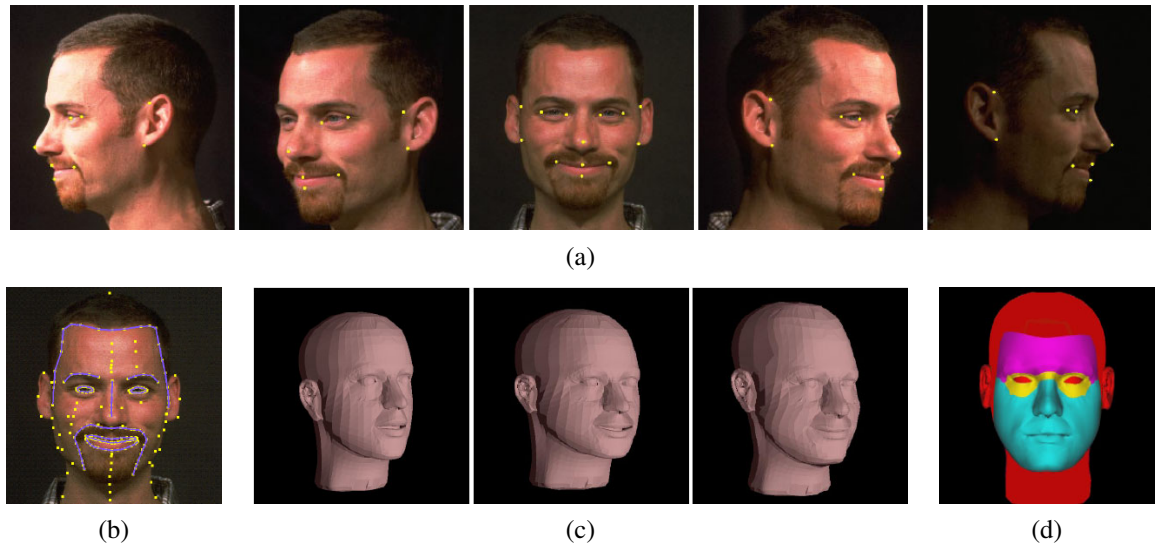


Figure 12.17 3D model fitting to a collection of images: (Pighin, Hecker, Lischinski *et al.* 1998) © 1998 ACM: (a) set of five input images along with user-selected keypoints; (b) the complete set of keypoints and curves; (c) three meshes—the original, adapted after 13 keypoints, and after an additional 99 keypoints; (d) the partition of the image into separately animatable regions.

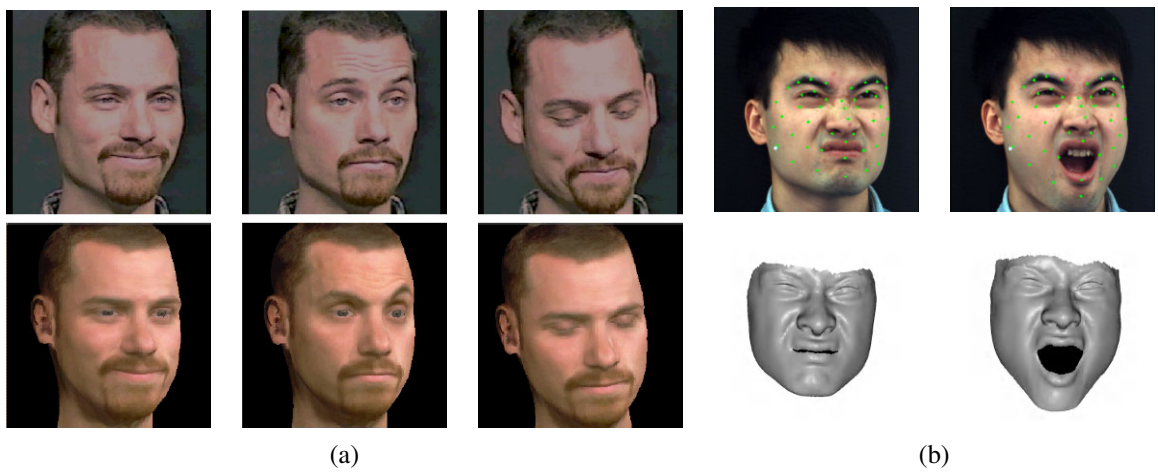


Figure 12.18 Head and expression tracking and re-animation using deformable 3D models. (a) Models fit directly to five input video streams (Pighin, Szeliski, and Salesin 2002) © 2002 Springer: The bottom row shows the results of re-animating a synthetic texture-mapped 3D model with pose and expression parameters fitted to the input images in the top row. (b) Models fit to frame-rate spacetime stereo surface models (Zhang, Snavely, Curless *et al.* 2004) © 2004 ACM: The top row shows the input images with synthetic green markers overlaid, while the bottom row shows the fitted 3D surface model.

time stereo with active illumination (Zhang, Snavely, Curless *et al.* 2004) (Figures 12.7 and 12.18b).

As the sophistication of 3D facial capture systems evolves, so does the detail and realism in the reconstructed models. Newer systems can capture (in real-time) not only surface details such as wrinkles and creases, but also accurate models of skin reflection, translucency, and sub-surface scattering (Weyrich, Matusik, Pfister *et al.* 2006; Golovinskiy, Matusik, ster *et al.* 2006; Bickel, Botsch, Angst *et al.* 2007; Igarashi, Nishino, and Nayar 2007).

Once a 3D head model has been constructed, it can be used in a variety of applications, such as head tracking (Toyama 1998; Lepetit, Pilet, and Fua 2004; Matthews, Xiao, and Baker 2007), as shown in Figures 4.29 and 14.24, and face transfer, i.e., replacing one person's face with another in a video (Bregler, Covell, and Slaney 1997; Vlasic, Brand, Pfister *et al.* 2005). Additional applications include face beautification by warping face images toward a more attractive "standard" (Leyvand, Cohen-Or, Dror *et al.* 2008), face de-identification for privacy protection (Gross, Sweeney, De la Torre *et al.* 2008), and face swapping (Bitouk, Kumar, Dhillon *et al.* 2008).

12.6.3 *Application*: Facial animation

Perhaps the most widely used application of 3D head modeling is facial animation. Once a parameterized 3D model of shape and appearance (surface texture) has been constructed, it can be used directly to track a person's facial motions (Figure 12.18a) and to animate a different character with these same motions and expressions (Pighin, Szeliski, and Salesin 2002).

An improved version of such a system can be constructed by first applying principal component analysis (PCA) to the space of possible head shapes and facial appearances. Blanz and Vetter (1999) describe a system where they first capture a set of 200 colored range scans of faces (Figure 12.19a), which can be represented as a large collection of (X, Y, Z, R, G, B) samples (vertices).[9] In order for 3D morphing to be meaningful, corresponding vertices in different people's scans must first be put into correspondence (Pighin, Hecker, Lischinski *et al.* 1998). Once this is done, PCA can be applied to more naturally parameterize the 3D morphable model. The flexibility of this model can be increased by performing separate analyses in different subregions, such as the eyes, nose, and mouth, just as in modular eigenspaces (Moghaddam and Pentland 1997).

After computing a subspace representation, different directions in this space can be associated with different characteristics such as gender, facial expressions, or facial features (Figure 12.19a). As in the work of Rowland and Perrett (1995), faces can be turned into caricatures by exaggerating their displacement from the mean image.

3D morphable models can be fitted to a single image using gradient descent on the error between the input image and the re-synthesized model image, after an initial manual placement of the model in an approximately correct pose, scale, and location (Figures 12.19b–c). The efficiency of this fitting process can be increased using inverse compositional image alignment (8.64–8.65), as described by Romdhani and Vetter (2003).

The resulting texture-mapped 3D model can then be modified to produce a variety of vi-

[9] A cylindrical coordinate system provides a natural two-dimensional embedding for this collection, but such an embedding is not necessary to perform PCA.

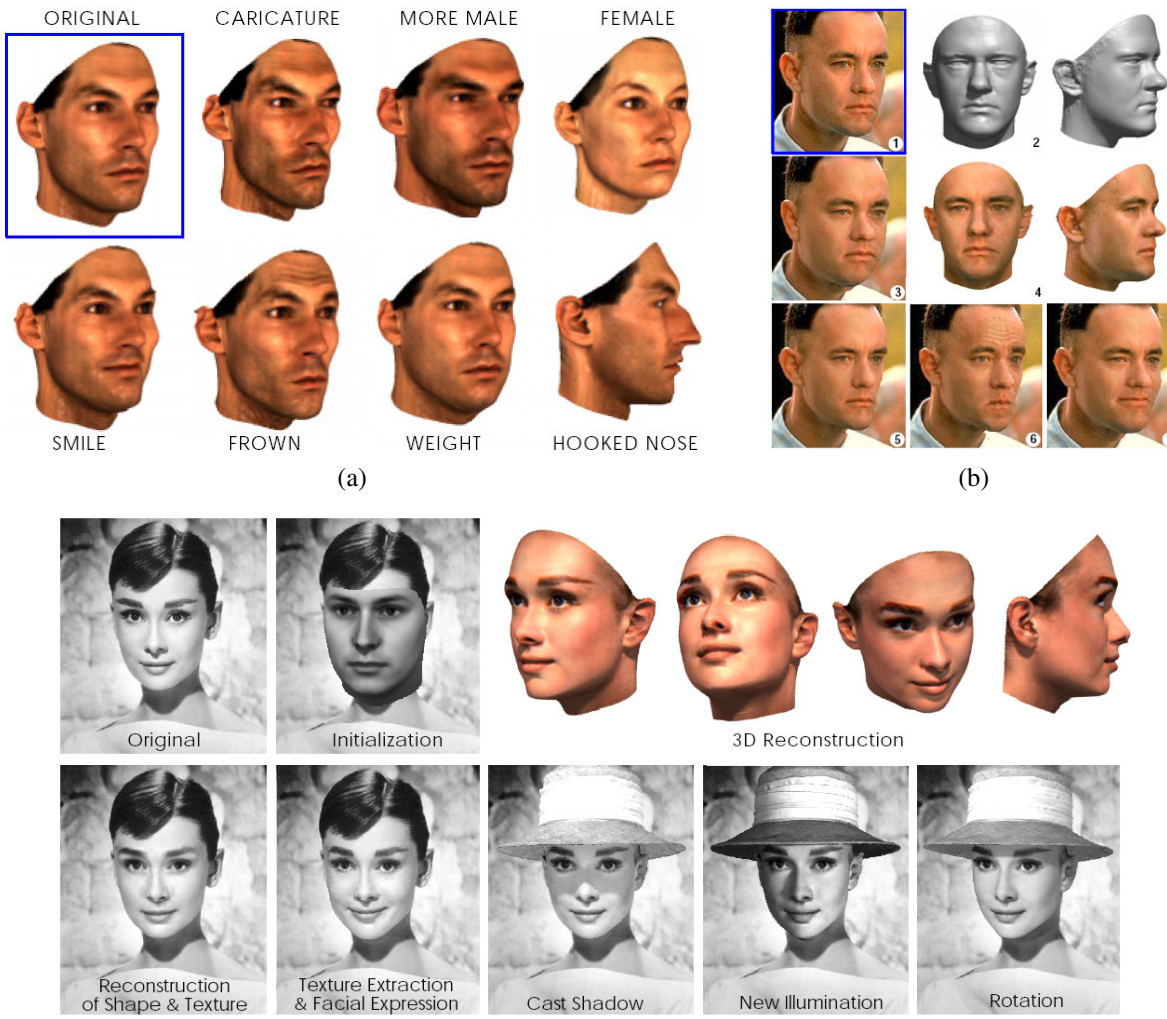


Figure 12.19 3D morphable face model (Blanz and Vetter 1999) © 1999 ACM: (a) original 3D face model with the addition of shape and texture variations in specific directions: deviation from the mean (caricature), gender, expression, weight, and nose shape; (b) a 3D morphable model is fit to a single image, after which its weight or expression can be manipulated; (c) another example of a 3D reconstruction along with a different set of 3D manipulations such as lighting and pose change.

sual effects, including changing a person's weight or expression, or three-dimensional effects such as re-lighting or 3D video-based animation (Section 13.5.1). Such models can also be used for video compression, e.g., by only transmitting a small number of facial expression and pose parameters to drive a synthetic avatar (Eisert, Wiegand, and Girod 2000; Gao, Chen, Wang *et al.* 2003).

3D facial animation is often matched to the performance of an actor, in what is known as *performance-driven animation* (Section 4.1.5) (Williams 1990). Traditional performance-driven animation systems use marker-based motion capture (Ma, Jones, Chiang *et al.* 2008), while some newer systems use video footage to control the animation (Buck, Finkelstein, Jacobs *et al.* 2000; Pighin, Szeliski, and Salesin 2002; Zhang, Snavely, Curless *et al.* 2004; Vlasic, Brand, Pfister *et al.* 2005).

An example of the latter approach is the system developed for the film *Benjamin Button*, in which Digital Domain used the CONTOUR system from Mova[10] to capture actor Brad Pitt's facial motions and expressions (Roble and Zafar 2009). CONTOUR uses a combination of phosphorescent paint and multiple high-resolution video cameras to capture real-time 3D range scans of the actor. These 3D models were then translated into Facial Action Coding System (FACS) shape and expression parameters (Ekman and Friesen 1978) to drive a different (older) synthetically animated computer-generated imagery (CGI) character.

12.6.4 Whole body modeling and tracking

The topics of tracking humans, modeling their shape and appearance, and recognizing their activities, are some of the most actively studied areas of computer vision. Annual conferences[11] and special journal issues (Hilton, Fua, and Ronfard 2006) are devoted to this subject, and two recent surveys (Forsyth, Arikan, Ikemoto *et al.* 2006; Moeslund, Hilton, and Krüger 2006) each list over 400 papers devoted to these topics.[12] The HumanEva database of articulated human motions[13] contains multi-view video sequences of human actions along with corresponding motion capture data, evaluation code, and a reference 3D tracker based on particle filtering. The companion paper by Sigal, Balan, and Black (2010) not only describes the database and evaluation but also has a nice survey of important work in this field.

Given the breadth of this area, it is difficult to categorize all of this research, especially since different techniques usually build on each other. Moeslund, Hilton, and Krüger (2006) divide their survey into initialization, tracking (which includes background modeling and segmentation), pose estimation, and action (activity) recognition. Forsyth, Arikan, Ikemoto *et al.* (2006) divide their survey into sections on tracking (background subtraction, deformable templates, flow, and probabilistic models), recovering 3D pose from 2D observations, and data association and body parts. They also include a section on motion synthesis, which is more widely studied in computer graphics (Arikan and Forsyth 2002; Kovar, Gleicher, and Pighin 2002; Lee, Chai, Reitsma *et al.* 2002; Li, Wang, and Shum 2002; Pullen and Bregler

[10] http://www.mova.com.

[11] International Conference on Automatic Face and Gesture Recognition (FG), IEEE Workshop on Analysis and Modeling of Faces and Gestures, and International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS).

[12] Older surveys include those by Gavrila (1999) and Moeslund and Granum (2001). Some surveys on gesture recognition, which we do not cover in this book, include those by Pavlović, Sharma, and Huang (1997) and Yang, Ahuja, and Tabb (2002).

[13] http://vision.cs.brown.edu/humaneva/.

2002), see Section 13.5.2. Another potential taxonomy for work in this field would be along the lines of whether 2D or 3D (or multi-view) images are used as input and whether 2D or 3D kinematic models are used.

In this section, we briefly review some of the more seminal and widely cited papers in the areas of background subtraction, initialization and detection, tracking with flow, 3D kinematic models, probabilistic models, adaptive shape modeling, and activity recognition. We refer the reader to the previously mentioned surveys for other topics and more details.

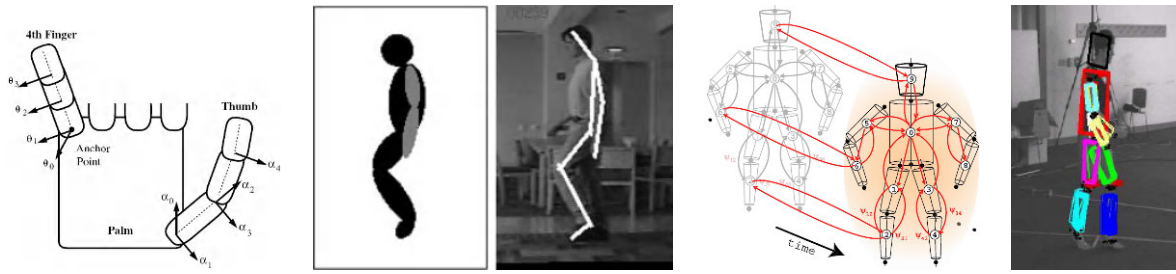
Background subtraction. One of the first steps in many (but certainly not all) human tracking systems is to model the background in order to extract the moving foreground objects (silhouettes) corresponding to people. Toyama, Krumm, Brumitt *et al.* (1999) review several *difference matting* and *background maintenance* (modeling) techniques and provide a good introduction to this topic. Stauffer and Grimson (1999) describe some techniques based on mixture models, while Sidenbladh and Black (2003) develop a more comprehensive treatment, which models not only the background image statistics but also the appearance of the foreground objects, e.g., their edge and motion (frame difference) statistics.

Once silhouettes have been extracted from one or more cameras, they can then be modeled using deformable templates or other contour models (Baumberg and Hogg 1996; Wren, Azarbayejani, Darrell *et al.* 1997). Tracking such silhouettes over time supports the analysis of multiple people moving around a scene, including building shape and appearance models and detecting if they are carrying objects (Haritaoglu, Harwood, and Davis 2000; Mittal and Davis 2003; Dimitrijevic, Lepetit, and Fua 2006).

Initialization and detection. In order to track people in a fully automated manner, it is necessary to first detect (or re-acquire) their presence in individual video frames. This topic is closely related to *pedestrian detection*, which is often considered as a kind of object recognition (Mori, Ren, Efros *et al.* 2004; Felzenszwalb and Huttenlocher 2005; Felzenszwalb, McAllester, and Ramanan 2008), and is therefore treated in more depth in Section 14.1.2. Additional techniques for initializing 3D trackers based on 2D images include those described by Howe, Leventon, and Freeman (2000), Rosales and Sclaroff (2000), Shakhnarovich, Viola, and Darrell (2003), Sminchisescu, Kanaujia, Li *et al.* (2005), Agarwal and Triggs (2006), Lee and Cohen (2006), Sigal and Black (2006), and Stenger, Thayananthan, Torr *et al.* (2006).

Single-frame human detection and pose estimation algorithms can sometimes be used by themselves to perform tracking (Ramanan, Forsyth, and Zisserman 2005; Rogez, Rihan, Ramalingam *et al.* 2008; Bourdev and Malik 2009), as described in Section 4.1.4. More often, however, they are combined with frame-to-frame tracking techniques to provide better reliability (Fossati, Dimitrijevic, Lepetit *et al.* 2007; Andriluka, Roth, and Schiele 2008; Ferrari, Marin-Jimenez, and Zisserman 2008).

Tracking with flow. The tracking of people and their pose from frame to frame can be enhanced by computing optic flow or matching the appearance of their limbs from one frame to another. For example, the *cardboard people* model of Ju, Black, and Yacoob (1996) models the appearance of each leg portion (upper and lower) as a moving rectangle, and uses optic flow to estimate their location in each subsequent frame. Cham and Rehg (1999) and Sidenbladh, Black, and Fleet (2000) track limbs using optical flow and templates, along with



(a) (b) (c) (d)

Figure 12.20 Tracking 3D human motion: (a) kinematic chain model for a human hand (Rehg, Morris, and Kanade 2003) © 2003, reprinted by permission of SAGE; (b) tracking a kinematic chain blob model in a video sequence (Bregler, Malik, and Pullen 2004) © 2004 Springer; (c–d) probabilistic loose-limbed collection of body parts (Sigal, Bhatia, Roth *et al.* 2004)

techniques for dealing with multiple hypotheses and uncertainty. Bregler, Malik, and Pullen (2004) use a full 3D model of limb and body motion, as described below. It is also possible to match the estimated motion field itself to some prototypes in order to identify the particular phase of a running motion or to match two low-resolution video portions in order to perform video replacement (Efros, Berg, Mori *et al.* 2003).

3D kinematic models. The effectiveness of human modeling and tracking can be greatly enhanced using a more accurate 3D model of a person's shape and motion. Underlying such representations, which are ubiquitous in 3D computer animation in games and special effects, is a *kinematic model* or *kinematic chain*, which specifies the length of each limb in a skeleton as well as the 2D or 3D rotation angles between the limbs or segments (Figure 12.20a–b). Inferring the values of the joint angles from the locations of the visible surface points is called *inverse kinematics* (IK) and is widely studied in computer graphics.

Figure 12.20a shows the kinematic model for a human hand used by Rehg, Morris, and Kanade (2003) to track hand motion in a video. As you can see, the attachment points between the fingers and the thumb have two degrees of freedom, while the finger joints themselves have only one. Using this kind of model can greatly enhance the ability of an edge-based tracker to cope with rapid motion, ambiguities in 3D pose, and partial occlusions.

Kinematic chain models are even more widely used for whole body modeling and tracking (O'Rourke and Badler 1980; Hogg 1983; Rohr 1994). One popular approach is to associate an ellipsoid or superquadric with each rigid limb in the kinematic model, as shown in Figure 12.20b. This model can then be fitted to each frame in one or more video streams either by matching silhouettes extracted from known backgrounds or by matching and tracking the locations of occluding edges (Gavrila and Davis 1996; Kakadiaris and Metaxas 2000; Bregler, Malik, and Pullen 2004; Kehl and Van Gool 2006). Note that some techniques use 2D models coupled to 2D measurements, some use 3D measurements (range data or multi-view video) with 3D models, and some use monocular video to infer and track 3D models directly.

It is also possible to use temporal models to improve the tracking of periodic motions, such as walking, by analyzing the joint angles as functions of time (Polana and Nelson 1997; Seitz and Dyer 1997; Cutler and Davis 2000). The generality and applicability of such tech-

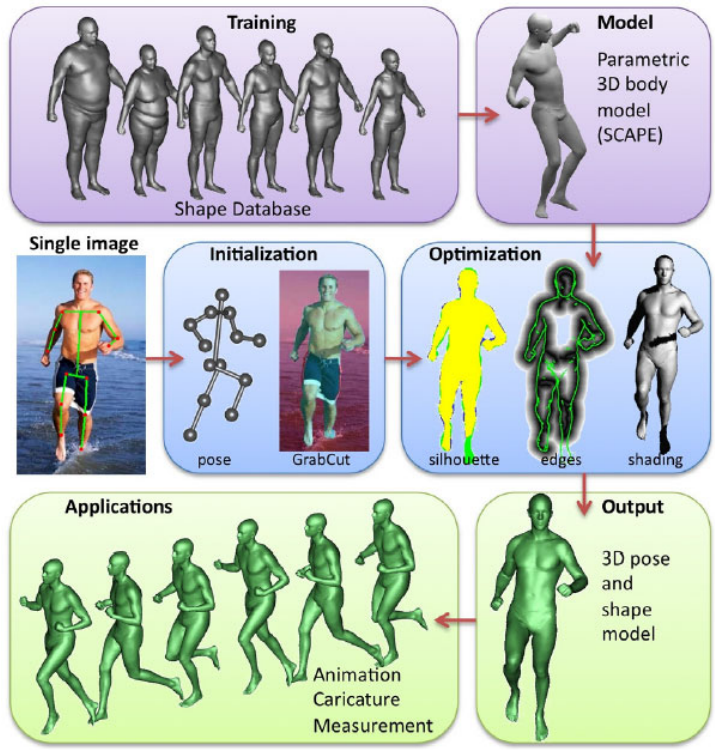


Figure 12.21 Estimating human shape and pose from a single image using a parametric 3D model (Guan, Weiss, Bălan *et al.* 2009) © 2009 IEEE.

niques can be improved by learning typical motion patterns using principal component analysis (Sidenbladh, Black, and Fleet 2000; Urtasun, Fleet, and Fua 2006).

Probabilistic models. Because tracking can be such a difficult task, sophisticated probabilistic inference techniques are often used to estimate the likely states of the person being tracked. One popular approach, called *particle filtering* (Isard and Blake 1998), was originally developed for tracking the outlines of people and hands, as described in Section 5.1.2 (Figures 5.6–5.8). It was subsequently applied to whole-body tracking (Deutscher, Blake, and Reid 2000; Sidenbladh, Black, and Fleet 2000; Deutscher and Reid 2005) and continues to be used in modern trackers (Ong, Micilotta, Bowden *et al.* 2006). Alternative approaches to handling the uncertainty inherent in tracking include multiple hypothesis tracking (Cham and Rehg 1999) and inflated covariances (Sminchisescu and Triggs 2001).

Figure 12.20c–d shows an example of a sophisticated spatio-temporal probabilistic graphical model called *loose-limbed people*, which models not only the geometric relationship between various limbs, but also their likely temporal dynamics (Sigal, Bhatia, Roth *et al.* 2004). The conditional probabilities relating various limbs and time instances are learned from training data, and particle filtering is used to perform the final pose inference.

Adaptive shape modeling. Another essential component of whole body modeling and tracking is the fitting of parameterized shape models to visual data. As we saw in Section 12.6.3 (Figure 12.19), the availability of large numbers of registered 3D range scans can be used to create *morphable models* of shape and appearance (Allen, Curless, and Popović 2003). Building on this work, Anguelov, Srinivasan, Koller *et al.* (2005) develop a sophisticated system called SCAPE (Shape Completion and Animation for PEople), which first acquires a large number of range scans of different people and of one person in different poses, and then registers these scans using semi-automated marker placement. The registered datasets are used to model the variation in shape as a function of personal characteristics and skeletal pose, e.g., the bulging of muscles as certain joints are flexed (Figure 12.21, top row). The resulting system can then be used for *shape completion*, i.e., the recovery of a full 3D mesh model from a small number of captured markers, by finding the best model parameters in both shape and pose space that fit the measured data.

Because it is constructed completely from scans of people in close-fitting clothing and uses a parametric shape model, the SCAPE system cannot cope with people wearing loose-fitting clothing. Bălan and Black (2008) overcome this limitation by estimating the body shape that fits within the visual hull of the same person observed in multiple poses, while Vlasic, Baran, Matusik *et al.* (2008) adapt an initial surface mesh fitted with a parametric shape model to better match the visual hull.

While the preceding body fitting and pose estimation systems use multiple views to estimate body shape, even more recent work by Guan, Weiss, Bălan *et al.* (2009) can fit a human shape and pose model to a single image of a person on a natural background. Manual initialization is used to estimate a rough pose (skeleton) and height model, and this is then used to segment the person's outline using the Grab Cut segmentation algorithm (Section 5.5). The shape and pose estimate are then refined using a combination of silhouette edge cues and shading information (Figure 12.21). The resulting 3D model can be used to create novel animations.

Activity recognition. The final widely studied topic in human modeling is motion, activity, and action recognition (Bobick 1997; Hu, Tan, Wang *et al.* 2004; Hilton, Fua, and Ronfard 2006). Examples of actions that are commonly recognized include walking and running, jumping, dancing, picking up objects, sitting down and standing up, and waving. Recent representative papers on these topics have been written by Robertson and Reid (2006), Sminchisescu, Kanaujia, and Metaxas (2006), Weinland, Ronfard, and Boyer (2006), Yilmaz and Shah (2006), and Gorelick, Blank, Shechtman *et al.* (2007).

12.7 Recovering texture maps and albedos

After a 3D model of an object or person has been acquired, the final step in modeling is usually to recover a *texture map* to describe the object's surface appearance. This first requires establishing a parameterization for the (u, v) texture coordinates as a function of 3D surface position. One simple way to do this is to associate a separate texture map with each triangle (or pair of triangles). More space-efficient techniques involve unwrapping the surface onto one or more maps, e.g., using a subdivision mesh (Section 12.3.2) (Eck, DeRose, Duchamp *et al.* 1995) or a geometry image (Section 12.3.3) (Gu, Gortler, and Hoppe 2002).



(a) (b) (c)

Figure 12.22 Estimating the diffuse albedo and reflectance parameters for a scanned 3D model (Sato, Wheeler, and Ikeuchi 1997) © 1997 ACM: (a) set of input images projected onto the model; (b) the complete diffuse reflection (albedo) model; (c) rendering from the reflectance model including the specular component.

Once the (u, v) coordinates for each triangle have been fixed, the perspective projection equations mapping from texture (u, v) to an image j's pixel (u_j, v_j) coordinates can be obtained by concatenating the affine $(u, v) \rightarrow (X, Y, Z)$ mapping with the perspective homography $(X, Y, Z) \rightarrow (u_j, v_j)$ (Szeliski and Shum 1997). The color values for the (u, v) texture map can then be re-sampled and stored, or the original image can itself be used as the texture source using projective texture mapping (OpenGL-ARB 1997).

The situation becomes more involved when more than one source image is available for appearance recovery, which is the usual case. One possibility is to use a *view-dependent texture map* (Section 13.1.1), in which a different source image (or combination of source images) is used for each polygonal face based on the angles between the virtual camera, the surface normals, and the source images (Debevec, Taylor, and Malik 1996; Pighin, Hecker, Lischinski *et al.* 1998). An alternative approach is to estimate a complete Surface Light Field for each surface point (Wood, Azuma, Aldinger *et al.* 2000), as described in Section 13.3.2.

In some situations, e.g., when using models in traditional 3D games, it is preferable to merge all of the source images into a single coherent texture map during pre-processing. Ideally, each surface triangle should select the source image where it is seen most directly (perpendicular to its normal) and at the resolution best matching the texture map resolution.[14] This can be posed as a graph cut optimization problem, where the smoothness term encourages adjacent triangles to use similar source images, followed by blending to compensate for exposure differences (Lempitsky and Ivanov 2007; Sinha, Steedly, Szeliski *et al.* 2008). Even better results can be obtained by explicitly modeling geometric and photometric misalignments between the source images (Shum and Szeliski 2000; Gal, Wexler, Ofek *et al.* 2010).

These kinds of approaches produce good results when the lighting stays fixed with respect to the object, i.e., when the camera moves around the object or space. When the lighting is strongly directional, however, and the object is being moved relative to this lighting, strong shading effects or specularities may be present, which will interfere with the reliable recovery of a texture (albedo) map. In this case, it is preferable to explicitly undo the shading effects (Section 12.1) by modeling the light source directions and estimating the surface reflectance properties while recovering the texture map (Sato and Ikeuchi 1996; Sato, Wheeler, and Ikeuchi 1997; Yu and Malik 1998; Yu, Debevec, Malik *et al.* 1999). Figure 12.22 shows

[14] When surfaces are seen at oblique viewing angles, it may be necessary to blend different images together to obtain the best resolution (Wang, Kang, Szeliski *et al.* 2001).

the results of one such approach, where the specularities are first removed while estimating the matte reflectance component (albedo) and then later re-introduced by estimating the specular component k_s in a Torrance–Sparrow reflection model (2.91).

12.7.1 Estimating BRDFs

A more ambitious approach to the problem of view-dependent appearance modeling is to estimate a general bidirectional reflectance distribution function (BRDF) for each point on an object's surface. Dana, van Ginneken, Nayar *et al.* (1999), Jensen, Marschner, Levoy *et al.* (2001), and Lensch, Kautz, Goesele *et al.* (2003) present different techniques for estimating such functions, while Dorsey, Rushmeier, and Sillion (2007) and Weyrich, Lawrence, Lensch *et al.* (2008) present more recent surveys of the topics of BRDF modeling, recovery, and rendering.

As we saw in Section 2.2.2 (2.81), the BRDF can be written as

$$f_r(\theta_i, \phi_i, \theta_r, \phi_r; \lambda), \tag{12.9}$$

where (θ_i, ϕ_i) and (θ_r, ϕ_r) are the angles the incident $\hat{\boldsymbol{v}}_i$ and reflected $\hat{\boldsymbol{v}}_r$ light ray directions make with the local surface coordinate frame $(\hat{\boldsymbol{d}}_x, \hat{\boldsymbol{d}}_y, \hat{\boldsymbol{n}})$ shown in Figure 2.15. When modeling the appearance of an object, as opposed to the appearance of a patch of material, we need to estimate this function at every point (x, y) on the object's surface, which gives us the *spatially varying* BRDF, or SVBRDF (Weyrich, Lawrence, Lensch *et al.* 2008),

$$f_v(x, y, \theta_i, \phi_i, \theta_r, \phi_r; \lambda). \tag{12.10}$$

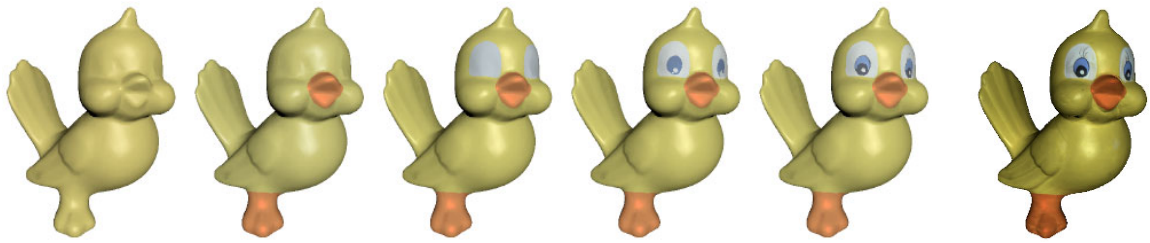
If sub-surface scattering effects are being modeled, such as the long-range transmission of light through materials such as alabaster, the eight-dimensional bidirectional scattering-surface reflectance-distribution function (BSSRDF) is used instead,

$$f_e(x_i, y_i, \theta_i, \phi_i, x_e, y_e, \theta_e, \phi_e; \lambda), \tag{12.11}$$

where the e subscript now represents the *emitted* rather than the *reflected* light directions.

Weyrich, Lawrence, Lensch *et al.* (2008) provide a nice survey of these and related topics, including basic photometry, BRDF models, traditional BRDF acquisition using *gonio reflectometry* (the precise measurement of visual angles and reflectances), multiplexed illumination (Schechner, Nayar, and Belhumeur 2009), skin modeling (Debevec, Hawkins, Tchou *et al.* 2000; Weyrich, Matusik, Pfister *et al.* 2006), and image-based acquisition techniques, which simultaneously recover an object's 3D shape and reflectometry from multiple photographs.

A nice example of this latter approach is the system developed by Lensch, Kautz, Goesele *et al.* (2003), who estimate locally varying BRDFs and refine their shape models using local estimates of surface normals. To build up their models, they first associate a *lumitexels*, which contains a 3D position, a surface normal, and a set of sparse radiance samples, with each surface point. Next, they cluster such lumitexels into materials that share common properties, using a Lafortune reflectance model (Lafortune, Foo, Torrance *et al.* 1997) and a divisive clustering approach (Figure 12.23a). Finally, in order to model detailed spatially varying appearance, each lumitexel (surface point) is projected onto the basis of clustered appearance models (Figure 12.23b).



(a) (b)

Figure 12.23 Image-based reconstruction of appearance and detailed geometry (Lensch, Kautz, Goesele *et al.* 2003) © 2003 ACM. (a) Appearance models (BRDFs) are re-estimated using divisive clustering. (b) In order to model detailed spatially varying appearance, each lumitexel is projected onto the basis formed by the clustered materials.

While most of the techniques discussed in this section require large numbers of views to estimate surface properties, a challenging future direction will be to take these techniques out of the lab and into the real world, and to combine them with regular and Internet photo image-based modeling approaches.

12.7.2 *Application*: 3D photography

The techniques described in this chapter for building complete 3D models from multiple images and then recovering their surface appearance have opened up a whole new range of applications that often go under the name *3D photography*. Pollefeys and Van Gool (2002) provide a nice introduction to this field, including the processing steps of feature matching, structure from motion recovery,[15] dense depth map estimation, 3D model building, and texture map recovery. A complete Web-based system for automatically performing all of these tasks, called ARC3D, is described by Vergauwen and Van Gool (2006) and Moons, Van Gool, and Vergauwen (2010). The latter paper provides not only an in-depth survey of this whole field but also a detailed description of their complete end-to-end system.

An alternative to such fully automated systems is to put the user in the loop in what is sometimes called *interactive computer vision*. van den Hengel, Dick, Thormhlen *et al.* (2007) describe their VideoTrace system, which performs automated point tracking and 3D structure recovery from video and then lets the user draw triangles and surfaces on top of the resulting point cloud, as well as interactively adjusting the locations of model vertices. Sinha, Steedly, Szeliski *et al.* (2008) describe a related system that uses matched vanishing points in multiple images (Figure 4.45) to infer 3D line orientations and plane normals. These are then used to guide the user drawing axis-aligned planes, which are automatically fitted to the recovered 3D point cloud. Fully automated variants on these ideas are described by Zebedin, Bauer, Karner *et al.* (2008), Furukawa, Curless, Seitz *et al.* (2009a), Furukawa, Curless, Seitz *et al.* (2009b), Mičušík and Košecká (2009), and Sinha, Steedly, and Szeliski (2009).

As the sophistication and reliability of these techniques continues to improve, we can expect to see even more user-friendly applications for photorealistic 3D modeling from images (Exercise 12.8).

[15] These earlier steps are also discussed in Section 7.4.4.

12.8 Additional reading

Shape from shading is one of the classic problems in computer vision (Horn 1975). Some representative papers in this area include those by Horn (1977), Ikeuchi and Horn (1981), Pentland (1984), Horn and Brooks (1986), Horn (1990), Szeliski (1991a), Mancini and Wolff (1992), Dupuis and Oliensis (1994), and Fua and Leclerc (1995). The collection of papers edited by Horn and Brooks (1989) is a great source of information on this topic, especially the chapter on variational approaches. The survey by Zhang, Tsai, Cryer *et al.* (1999) not only reviews more recent techniques but also provides some comparative results.

Woodham (1981) wrote the seminal paper of photometric stereo. Shape from texture techniques include those by Witkin (1981), Ikeuchi (1981), Blostein and Ahuja (1987), Garding (1992), Malik and Rosenholtz (1997), Liu, Collins, and Tsin (2004), Liu, Lin, and Hays (2004), Hays, Leordeanu, Efros *et al.* (2006), Lin, Hays, Wu *et al.* (2006), Lobay and Forsyth (2006), White and Forsyth (2006), White, Crane, and Forsyth (2007), and Park, Brocklehurst, Collins *et al.* (2009). Good papers and books on depth from defocus have been written by Pentland (1987), Nayar and Nakagawa (1994), Nayar, Watanabe, and Noguchi (1996), Watanabe and Nayar (1998), Chaudhuri and Rajagopalan (1999), and Favaro and Soatto (2006). Additional techniques for recovering shape from various kinds of illumination effects, including inter-reflections (Nayar, Ikeuchi, and Kanade 1991), are discussed in the book on shape recovery edited by Wolff, Shafer, and Healey (1992b).

Active rangefinding systems, which use laser or natural light illumination projected into the scene, have been described by Besl (1989), Rioux and Bird (1993), Kang, Webb, Zitnick *et al.* (1995), Curless and Levoy (1995), Curless and Levoy (1996), Proesmans, Van Gool, and Defoort (1998), Bouguet and Perona (1999), Curless (1999), Hebert (2000), Iddan and Yahav (2001), Goesele, Fuchs, and Seidel (2003), Scharstein and Szeliski (2003), Davis, Ramamoorthi, and Rusinkiewicz (2003), Zhang, Curless, and Seitz (2003), Zhang, Snavely, Curless *et al.* (2004), and Moons, Van Gool, and Vergauwen (2010). Individual range scans can be aligned using 3D correspondence and distance optimization techniques such as *iterated closest points* and its variants (Besl and McKay 1992; Zhang 1994; Szeliski and Lavallée 1996; Johnson and Kang 1997; Gold, Rangarajan, Lu *et al.* 1998; Johnson and Hebert 1999; Pulli 1999; David, DeMenthon, Duraiswami *et al.* 2004; Li and Hartley 2007; Enqvist, Josephson, and Kahl 2009). Once they have been aligned, range scans can be merged using techniques that model the signed distance of surfaces to volumetric sample points (Hoppe, DeRose, Duchamp *et al.* 1992; Curless and Levoy 1996; Hilton, Stoddart, Illingworth *et al.* 1996; Wheeler, Sato, and Ikeuchi 1998; Kazhdan, Bolitho, and Hoppe 2006; Lempitsky and Boykov 2007; Zach, Pock, and Bischof 2007b; Zach 2008).

Once constructed, 3D surfaces can be modeled and manipulated using a variety of three-dimensional representations, which include triangle meshes (Eck, DeRose, Duchamp *et al.* 1995; Hoppe 1996), splines (Farin 1992, 1996; Lee, Wolberg, and Shin 1996), subdivision surfaces (Stollnitz, DeRose, and Salesin 1996; Zorin, Schröder, and Sweldens 1996; Warren and Weimer 2001; Peters and Reif 2008), and geometry images (Gu, Gortler, and Hoppe 2002). Alternatively, they can be represented as collections of point samples with local orientation estimates (Hoppe, DeRose, Duchamp *et al.* 1992; Szeliski and Tonnesen 1992; Turk and O'Brien 2002; Pfister, Zwicker, van Baar *et al.* 2000; Alexa, Behr, Cohen-Or *et al.* 2003; Pauly, Keiser, Kobbelt *et al.* 2003; Diebel, Thrun, and Brünig 2006; Guennebaud and Gross

2007; Guennebaud, Germann, and Gross 2008; Oztireli, Guennebaud, and Gross 2008). They can also be modeled using implicit inside–outside characteristic or signed distance functions sampled on regular or irregular (octree) volumetric grids (Lavallée and Szeliski 1995; Szeliski and Lavallée 1996; Frisken, Perry, Rockwood *et al.* 2000; Dinh, Turk, and Slabaugh 2002; Kazhdan, Bolitho, and Hoppe 2006; Lempitsky and Boykov 2007; Zach, Pock, and Bischof 2007b; Zach 2008).

The literature on model-based 3D reconstruction is extensive. For modeling architecture and urban scenes, both interactive and fully automated systems have been developed. A special journal issue devoted to the reconstruction of large-scale 3D scenes (Zhu and Kanade 2008) is a good source of references and Robertson and Cipolla (2009) give a nice description of a complete system. Lots of additional references can be found in Section 12.6.1.

Face and whole body modeling and tracking is a very active sub-field of computer vision, with its own conferences and workshops, e.g., the International Conference on Automatic Face and Gesture Recognition (FG), the IEEE Workshop on Analysis and Modeling of Faces and Gestures, and the International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS). Recent survey articles on the topic of whole body modeling and tracking include those by Forsyth, Arikan, Ikemoto *et al.* (2006), Moeslund, Hilton, and Krüger (2006), and Sigal, Balan, and Black (2010).

12.9 Exercises

Ex 12.1: Shape from focus Grab a series of focused images with a digital SLR set to manual focus (or get one that allows for programmatic focus control) and recover the depth of an object.

1. Take some calibration images, e.g., of a checkerboard, so you can compute a mapping between the amount of defocus and the focus setting.
2. Try both a fronto-parallel planar target and one which is slanted so that it covers the working range of the sensor. Which one works better?
3. Now put a real object in the scene and perform a similar focus sweep.
4. For each pixel, compute the local sharpness and fit a parabolic curve over focus settings to find the most in-focus setting.
5. Map these focus settings to depth and compare your result to ground truth. If you are using a known simple object, such as sphere or cylinder (a ball or a soda can), it's easy to measure its true shape.
6. (Optional) See if you can recover the depth map from just two or three focus settings.
7. (Optional) Use an LCD projector to project artificial texture onto the scene. Use a pair of cameras to compare the accuracy of your shape from focus and shape from stereo techniques.
8. (Optional) Create an all-in-focus image using the technique of Agarwala, Dontcheva, Agrawala *et al.* (2004).

Ex 12.2: Shadow striping Implement the handheld shadow striping system of Bouguet and Perona (1999). The basic steps include the following.

1. Set up two background planes behind the object of interest and calculate their orientation relative to the viewer, e.g., with fiducial marks.
2. Cast a moving shadow with a stick across the scene; record the video or capture the data with a webcam.
3. Estimate each light plane equation from the projections of the cast shadow against the two backgrounds.
4. Triangulate to the remaining points on each curve to get a 3D stripe and display the stripes using a 3D graphics engine.
5. (Optional) remove the requirement for a known second (vertical) plane and infer its location (or that of the light source) using the techniques described by Bouguet and Perona (1999). The techniques from Exercise 10.9 may also be helpful here.

Ex 12.3: Range data registration Register two or more 3D datasets using either iterated closest points (ICP) (Besl and McKay 1992; Zhang 1994; Gold, Rangarajan, Lu *et al.* 1998) or octree signed distance fields (Szeliski and Lavallée 1996) (Section 12.2.1).

Apply your technique to narrow-baseline stereo pairs, e.g., obtained by moving a camera around an object, using structure from motion to recover the camera poses, and using a standard stereo matching algorithm.

Ex 12.4: Range data merging Merge the datasets that you registered in the previous exercise using signed distance fields (Curless and Levoy 1996; Hilton, Stoddart, Illingworth *et al.* 1996). You can optionally use an octree to represent and compress this field if you already implemented it in the previous registration step.

Extract a meshed surface model from the signed distance field using marching cubes and display the resulting model.

Ex 12.5: Surface simplification Use progressive meshes (Hoppe 1996) or some other technique from Section 12.3.2 to create a hierarchical simplification of your surface model.

Ex 12.6: Architectural modeler Build a 3D interior or exterior model of some architectural structure, such as your house, from a series of handheld wide-angle photographs.

1. Extract lines and vanishing points (Exercises 4.11–4.15) to estimate the dominant directions in each image.
2. Use structure from motion to recover all of the camera poses and match up the vanishing points.
3. Let the user sketch the locations of the walls by drawing lines corresponding to wall bottoms, tops, and horizontal extents onto the images (Sinha, Steedly, Szeliski *et al.* 2008)—see also Exercise 6.9. Do something similar for openings (doors and windows) and simple furniture (tables and countertops).

4. Convert the resulting polygonal meshes into a 3D model (e.g., VRML) and optionally texture-map these surfaces from the images.

Ex 12.7: Body tracker Download the video sequences from the HumanEva Web site.[16] Either implement a human motion tracker from scratch or extend the code on that Web site (Sigal, Balan, and Black 2010) in some interesting way.

Ex 12.8: 3D photography Combine all of your previously developed techniques to produce a system that takes a series of photographs or a video and constructs a photorealistic texture-mapped 3D model.

[16] http://vision.cs.brown.edu/humaneva/.

Chapter 13

Image-based rendering

13.1 View interpolation 545
13.1.1 View-dependent texture maps 547
13.1.2 *Application*: Photo Tourism 548
13.2 Layered depth images 549
13.2.1 Impostors, sprites, and layers 549
13.3 Light fields and Lumigraphs 551
13.3.1 Unstructured Lumigraph 554
13.3.2 Surface light fields 555
13.3.3 *Application*: Concentric mosaics 556
13.4 Environment mattes 556
13.4.1 Higher-dimensional light fields 558
13.4.2 The modeling to rendering continuum 559
13.5 Video-based rendering 560
13.5.1 Video-based animation 560
13.5.2 Video textures 561
13.5.3 *Application*: Animating pictures 564
13.5.4 3D Video 564
13.5.5 *Application*: Video-based walkthroughs 566
13.6 Additional reading 569
13.7 Exercises 570

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_13, © Springer-Verlag London Limited 2011



Figure 13.1 Image-based and video-based rendering: (a) a 3D view of a Photo Tourism reconstruction (Snavely, Seitz, and Szeliski 2006) © 2006 ACM; (b) a slice through a 4D light field (Gortler, Grzeszczuk, Szeliski *et al.* 1996) © 1996 ACM; (c) sprites with depth (Shade, Gortler, He *et al.* 1998) © 1998 ACM; (d) surface light field (Wood, Azuma, Aldinger *et al.* 2000) © 2000 ACM; (e) environment matte in front of a novel background (Zongker, Werner, Curless *et al.* 1999) © 1999 ACM; (f) real-time video environment matte (Chuang, Zongker, Hindorff *et al.* 2000) © 2000 ACM; (g) Video Rewrite used to re-animate old video (Bregler, Covell, and Slaney 1997) © 1997 ACM; (h) video texture of a candle flame (Schödl, Szeliski, Salesin *et al.* 2000) © 2000 ACM; (i) video view interpolation (Zitnick, Kang, Uyttendaele *et al.* 2004) © 2004 ACM.

Over the last two decades, image-based rendering has emerged as one of the most exciting applications of computer vision (Kang, Li, Tong *et al.* 2006; Shum, Chan, and Kang 2007). In image-based rendering, 3D reconstruction techniques from computer vision are combined with computer graphics rendering techniques that use multiple views of a scene to create interactive photo-realistic experiences, such as the Photo Tourism system shown in Figure 13.1a. Commercial versions of such systems include immersive street-level navigation in on-line mapping systems[1] and the creation of 3D Photosynths[2] from large collections of casually acquired photographs.

In this chapter, we explore a variety of image-based rendering techniques, such as those illustrated in Figure 13.1. We begin with *view interpolation* (Section 13.1), which creates a seamless transition between a pair of reference images using one or more pre-computed depth maps. Closely related to this idea are *view-dependent texture maps* (Section 13.1.1), which blend multiple texture maps on a 3D model's surface. The representations used for both the color imagery and the 3D geometry in view interpolation include a number of clever variants such as *layered depth images* (Section 13.2) and *sprites with depth* (Section 13.2.1).

We continue our exploration of image-based rendering with the *light field* and *Lumigraph* four-dimensional representations of a scene's appearance (Section 13.3), which can be used to render the scene from any arbitrary viewpoint. Variants on these representations include the *unstructured Lumigraph* (Section 13.3.1), *surface light fields* (Section 13.3.2), *concentric mosaics* (Section 13.3.3), and *environment mattes* (Section 13.4).

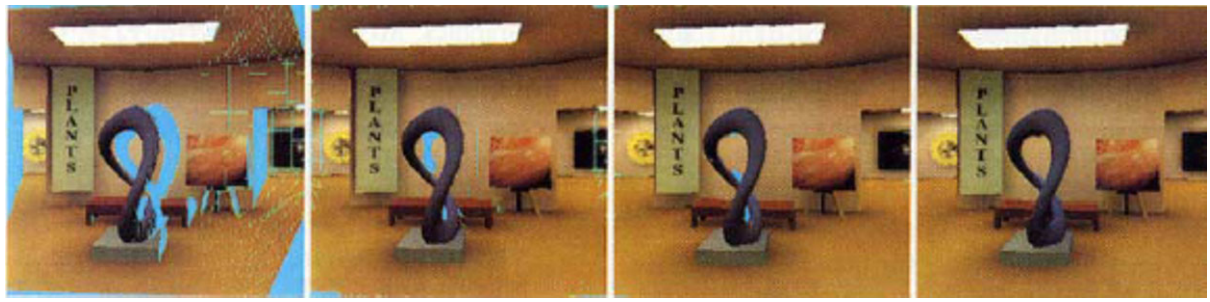
The last part of this chapter explores the topic of *video-based rendering*, which uses one or more videos in order to create novel video-based experiences (Section 13.5). The topics we cover include video-based facial animation (Section 13.5.1), as well as *video textures* (Section 13.5.2), in which short video clips can be seamlessly looped to create dynamic real-time video-based renderings of a scene. We close with a discussion of *3D videos* created from multiple video streams (Section 13.5.4), as well as *video-based walkthroughs* of environments (Section 13.5.5), which have found widespread application in immersive outdoor mapping and driving direction systems.

13.1 View interpolation

While the term *image-based rendering* first appeared in the papers by Chen (1995) and McMillan and Bishop (1995), the work on *view interpolation* by Chen and Williams (1993) is considered as the seminal paper in the field. In view interpolation, pairs of rendered color images are combined with their pre-computed depth maps to generate interpolated views that mimic what a virtual camera would see in between the two reference views.

View interpolation combines two ideas that were previously used in computer vision and computer graphics. The first is the idea of pairing a recovered depth map with the reference image used in its computation and then using the resulting texture-mapped 3D model to generate novel views (Figure 11.1). The second is the idea of *morphing* (Section 3.6.3) (Figure 3.53), where correspondences between pairs of images are used to warp each reference image to an in-between location while simultaneously cross-dissolving between the two warped images.

[1] http://maps.bing.com and http://maps.google.com.
[2] http://photosynth.net.



(a) (b) (c) (d)

Figure 13.2 View interpolation (Chen and Williams 1993) © 1993 ACM: (a) holes from one source image (shown in blue); (b) holes after combining two widely spaced images; (c) holes after combining two closely spaced images; (d) after interpolation (hole filling).

Figure 13.2 illustrates this process in more detail. First, both source images are warped to the novel view, using both the knowledge of the reference and virtual 3D camera pose along with each image's depth map (2.68–2.70). In the paper by Chen and Williams (1993), a *forward warping* algorithm (Algorithm 3.1 and Figure 3.46) is used. The depth maps are represented as quadtrees for both space and rendering time efficiency (Samet 1989).

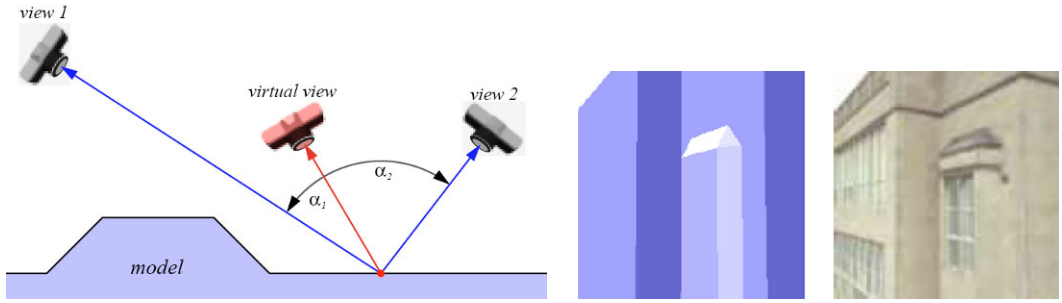
During the forward warping process, multiple pixels (which occlude one another) may land on the same destination pixel. To resolve this conflict, either a *z-buffer* depth value can be associated with each destination pixel or the images can be warped in back-to-front order, which can be computed based on the knowledge of epipolar geometry (Chen and Williams 1993; Laveau and Faugeras 1994; McMillan and Bishop 1995).

Once the two reference images have been warped to the novel view (Figure 13.2a–b), they can be merged to create a coherent composite (Figure 13.2c). Whenever one of the images has a *hole* (illustrated as a cyan pixel), the other image is used as the final value. When both images have pixels to contribute, these can be blended as in usual morphing, i.e., according to the relative distances between the virtual and source cameras. Note that if the two images have very different exposures, which can happen when performing view interpolation on real images, the hole-filled regions and the blended regions will have different exposures, leading to subtle artifacts.

The final step in view interpolation (Figure 13.2d) is to fill any remaining holes or cracks due to the forward warping process or lack of source data (scene visibility). This can be done by copying pixels from the *further* pixels adjacent to the hole. (Otherwise, foreground objects are subject to a "fattening effect".)

The above process works well for rigid scenes, although its visual quality (lack of aliasing) can be improved using a two-pass, forward–backward algorithm (Section 13.2.1) (Shade, Gortler, He *et al.* 1998) or full 3D rendering (Zitnick, Kang, Uyttendaele *et al.* 2004). In the case where the two reference images are views of a non-rigid scene, e.g., a person smiling in one image and frowning in the other, *view morphing*, which combines ideas from view interpolation with regular morphing, can be used (Seitz and Dyer 1996).

While the original view interpolation paper describes how to generate novel views based on similar pre-computed (linear perspective) images, the *plenoptic modeling* paper of McMillan and Bishop (1995) argues that cylindrical images should be used to store the pre-computed



(a) (b) (c)

Figure 13.3 View-dependent texture mapping (Debevec, Taylor, and Malik 1996) © 1996 ACM. (a) The weighting given to each input view depends on the relative angles between the novel (virtual) view and the original views; (b) simplified 3D model geometry; (c) with view-dependent texture mapping, the geometry appears to have more detail (recessed windows).

rendering or real-world images. (Chen 1995) also propose using environment maps (cylindrical, cubic, or spherical) as source images for view interpolation.

13.1.1 View-dependent texture maps

View-dependent texture maps (Debevec, Taylor, and Malik 1996) are closely related to view interpolation. Instead of associating a separate depth map with each input image, a single 3D model is created for the scene, but different images are used as texture map sources depending on the virtual camera's current position (Figure 13.3a).[3]

In more detail, given a new virtual camera position, the similarity of this camera's view of each polygon (or pixel) is compared to that of potential source images. The images are then blended using a weighting that is inversely proportional to the angles α_i between the virtual view and the source views (Figure 13.3a). Even though the geometric model can be fairly coarse (Figure 13.3b), blending between different views gives a strong sense of more detailed geometry because of the parallax (visual motion) between corresponding pixels. While the original paper performs the weighted blend computation separately at each pixel or coarsened polygon face, follow-on work by Debevec, Yu, and Borshukov (1998) presents a more efficient implementation based on precomputing contributions for various portions of viewing space and then using projective texture mapping (OpenGL-ARB 1997).

The idea of view-dependent texture mapping has been used in a large number of subsequent image-based rendering systems, including facial modeling and animation (Pighin, Hecker, Lischinski *et al.* 1998) and 3D scanning and visualization (Pulli, Abi-Rached, Duchamp *et al.* 1998). Closely related to view-dependent texture mapping is the idea of blending between light rays in 4D space, which forms the basis of the Lumigraph and unstructured Lumigraph systems (Section 13.3) (Gortler, Grzeszczuk, Szeliski *et al.* 1996; Buehler, Bosse, McMillan *et al.* 2001).

[3] The term *image-based modeling*, which is now commonly used to describe the creation of texture-mapped 3D models from multiple images, appears to have first been used by Debevec, Taylor, and Malik (1996), who also used the term *photogrammetric modeling* to describe the same process.



(a) (b)

Figure 13.4 Photo Tourism (Snavely, Seitz, and Szeliski 2006): © 2006 ACM: (a) a 3D overview of the scene, with translucent washes and lines painted onto the planar impostors; (b) once the user has selected a region of interest, a set of related thumbnails is displayed along the bottom; (c) planar proxy selection for optimal stabilization (Snavely, Garg, Seitz *et al.* 2008) © 2008 ACM.

In order to provide even more realism in their Façade system, Debevec, Taylor, and Malik (1996) also include a *model-based stereo* component, which optionally computes an offset (parallax) map for each coarse planar facet of their 3D model. They call the resulting analysis and rendering system a *hybrid geometry- and image-based* approach, since it uses traditional 3D geometric modeling to create the global 3D model, but then uses local depth offsets, along with view interpolation, to add visual realism.

13.1.2 *Application*: Photo Tourism

While view interpolation was originally developed to accelerate the rendering of 3D scenes on low-powered processors and systems without graphics acceleration, it turns out that it can be applied directly to large collections of casually acquired photographs. The *Photo Tourism* system developed by Snavely, Seitz, and Szeliski (2006) uses structure from motion to compute the 3D locations and poses of all the cameras taking the images, along with a sparse 3D point-cloud model of the scene (Section 7.4.4, Figure 7.11).

To perform an image-based exploration of the resulting *sea of images* (Aliaga, Funkhouser, Yanovsky *et al.* 2003), Photo Tourism first associates a 3D proxy with each image. While a triangulated mesh obtained from the point cloud can sometimes form a suitable proxy, e.g., for outdoor terrain models, a simple dominant plane fit to the 3D points visible in each image often performs better, because it does not contain any erroneous segments or connections that pop out as artifacts. As automated 3D modeling techniques continue to improve, however, the pendulum may swing back to more detailed 3D geometry (Goesele, Snavely, Curless *et al.* 2007; Sinha, Steedly, and Szeliski 2009).

The resulting image-based navigation system lets users move from photo to photo, either by selecting cameras from a top-down view of the scene (Figure 13.4a) or by selecting regions of interest in an image, navigating to nearby views, or selecting related thumbnails (Figure 13.4b). To create a background for the 3D scene, e.g., when being viewed from above, non-photorealistic techniques (Section 10.5.2), such as translucent color washes or highlighted 3D line segments, can be used (Figure 13.4a). The system can also be used to annotate regions of images and to automatically propagate such annotations to other photographs.

The 3D planar proxies used in Photo Tourism and the related Photosynth system from Microsoft result in non-photorealistic transitions reminiscent of visual effects such as "page flips". Selecting a stable 3D axis for all the planes can reduce the amount of swimming and enhance the perception of 3D (Figure 13.4c) (Snavely, Garg, Seitz *et al.* 2008). It is also possible to automatically detect objects in the scene that are seen from multiple views and create "orbits" of viewpoints around such objects. Furthermore, nearby images in both 3D position and viewing direction can be linked to create "virtual paths", which can then be used to navigate between arbitrary pairs of images, such as those you might take yourself while walking around a popular tourist site (Snavely, Garg, Seitz *et al.* 2008).

The spatial matching of image features and regions performed by Photo Tourism can also be used to infer more information from large image collections. For example, Simon, Snavely, and Seitz (2007) show how the match graph between images of popular tourist sites can be used to find the most *iconic* (commonly photographed) objects in the collection, along with their related tags. In follow-on work, Simon and Seitz (2008) show how such tags can be propagated to sub-regions of each image, using an analysis of which 3D points appear in the central portions of photographs. Extensions of these techniques to *all* of the world's images, including the use of GPS tags where available, have been investigated as well (Li, Wu, Zach *et al.* 2008; Quack, Leibe, and Van Gool 2008; Crandall, Backstrom, Huttenlocher *et al.* 2009; Li, Crandall, and Huttenlocher 2009; Zheng, Zhao, Song *et al.* 2009).

13.2 Layered depth images

Traditional view interpolation techniques associate a single depth map with each source or reference image. Unfortunately, when such a depth map is warped to a novel view, holes and cracks inevitably appear behind the foreground objects. One way to alleviate this problem is to keep several depth and color values (*depth pixels*) at every pixel in a reference image (or, at least for pixels near foreground–background transitions) (Figure 13.5). The resulting data structure, which is called a *layered depth image* (LDI), can be used to render new views using a back-to-front forward warping (splatting) algorithm (Shade, Gortler, He *et al.* 1998).

13.2.1 Impostors, sprites, and layers

An alternative to keeping lists of color-depth values at each pixel, as is done in the LDI, is to organize objects into different *layers* or *sprites*. The term sprite originates in the computer game industry, where it is used to designate flat animated characters in games such as Pac-Man or Mario Bros. When put into a 3D setting, such objects are often called *impostors*, because they use a piece of flat, alpha-matted geometry to represent simplified versions of 3D objects that are far away from the camera (Shade, Lischinski, Salesin *et al.* 1996; Lengyel and Snyder 1997; Torborg and Kajiya 1996). In computer vision, such representations are usually called *layers* (Wang and Adelson 1994; Baker, Szeliski, and Anandan 1998; Torr, Szeliski, and Anandan 1999; Birchfield, Natarajan, and Tomasi 2007). Section 8.5.2 discusses the topics of transparent layers and reflections, which occur on specular and transparent surfaces such as glass.

While flat layers can often serve as an adequate representation of geometry and appearance for far-away objects, better geometric fidelity can be achieved by also modeling the

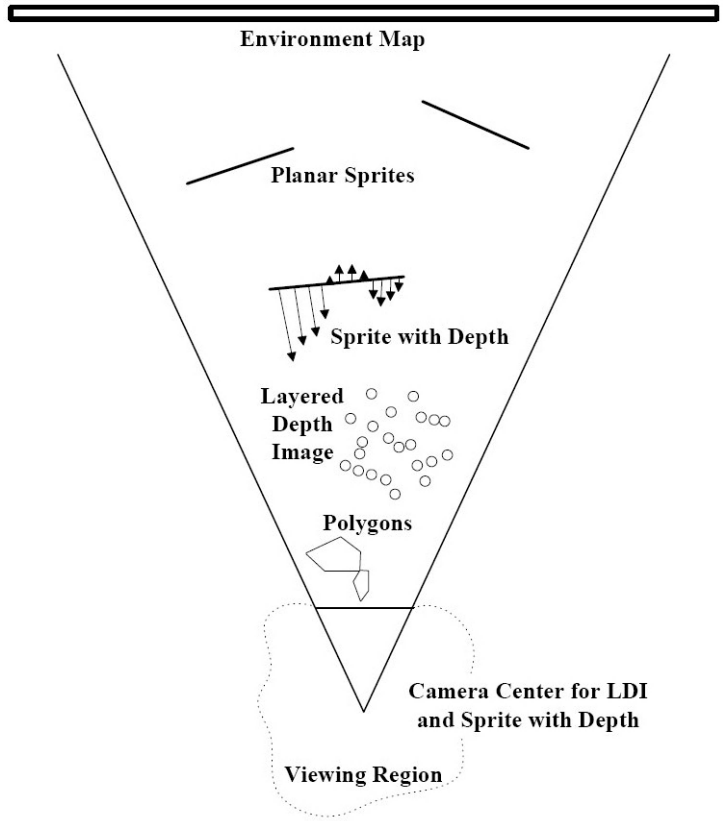


Figure 13.5 A variety of image-based rendering primitives, which can be used depending on the distance between the camera and the object of interest (Shade, Gortler, He *et al.* 1998) © 1998 ACM. Closer objects may require more detailed polygonal representations, while mid-level objects can use a layered depth image (LDI), and far-away objects can use sprites (potentially with depth) and environment maps.



(a) (b) (c) (d)

Figure 13.6 Sprites with depth (Shade, Gortler, He *et al.* 1998) © 1998 ACM: (a) alpha-matted color sprite; (b) corresponding relative depth or parallax; (c) rendering without relative depth; (d) rendering with depth (note the curved object boundaries).

per-pixel offsets relative to a base plane, as shown in Figures 13.5 and 13.6a–b. Such representations are called *plane plus parallax* in the computer vision literature (Kumar, Anandan, and Hanna 1994; Sawhney 1994; Szeliski and Coughlan 1997; Baker, Szeliski, and Anandan 1998), as discussed in Section 8.5 (Figure 8.16). In addition to fully automated stereo techniques, it is also possible to paint in depth layers (Kang 1998; Oh, Chen, Dorsey *et al.* 2001; Shum, Sun, Yamazaki *et al.* 2004) or to infer their 3D structure from monocular image cues (Section 14.4.4) (Hoiem, Efros, and Hebert 2005b; Saxena, Sun, and Ng 2009).

How can we render a sprite with depth from a novel viewpoint? One possibility, as with a regular depth map, is to just forward warp each pixel to its new location, which can cause aliasing and cracks. A better way, which we already mentioned in Section 3.6.2, is to first warp the depth (or (u, v) displacement) map to the novel view, fill in the cracks, and then use higher-quality inverse warping to resample the color image (Shade, Gortler, He *et al.* 1998). Figure 13.6d shows the results of applying such a two-pass rendering algorithm. From this still image, you can appreciate that the foreground sprites look more rounded; however, to fully appreciate the improvement in realism, you would have to look at the actual animated sequence.

Sprites with depth can also be rendered using conventional graphics hardware, as described in (Zitnick, Kang, Uyttendaele *et al.* 2004). Rogmans, Lu, Bekaert *et al.* (2009) describe GPU implementations of both real-time stereo matching and real-time forward and inverse rendering algorithms.

13.3 Light fields and Lumigraphs

While image-based rendering approaches can synthesize scene renderings from novel viewpoints, they raise the following more general question:

> *Is is possible to capture and render the appearance of a scene from all possible viewpoints and, if so, what is the complexity of the resulting structure?*

Let us assume that we are looking at a static scene, i.e., one where the objects and illuminants are fixed, and only the observer is moving around. Under these conditions, we can describe each image by the location and orientation of the virtual camera (6 dof) as well as its intrinsics (e.g., its focal length). However, if we capture a two-dimensional *spherical* image around each possible camera location, we can re-render any view from this information.[4] Thus, taking the cross-product of the three-dimensional space of camera positions with the 2D space of spherical images, we obtain the 5D *plenoptic function* of Adelson and Bergen (1991), which forms the basis of the image-based rendering system of McMillan and Bishop (1995).

Notice, however, that when there is no light dispersion in the scene, i.e., no smoke or fog, all the coincident rays along a portion of free space (between solid or refractive objects) have the same color value. Under these conditions, we can reduce the 5D plenoptic function to the 4D *light field* of all possible rays (Gortler, Grzeszczuk, Szeliski *et al.* 1996; Levoy and Hanrahan 1996; Levoy 2006).[5]

[4] Since we are counting dimensions, we ignore for now any sampling or resolution issues.

[5] Levoy and Hanrahan (1996) borrowed the term *light field* from a paper by Gershun (1939). Another name for this representation is the *photic field* (Moon and Spencer 1981).

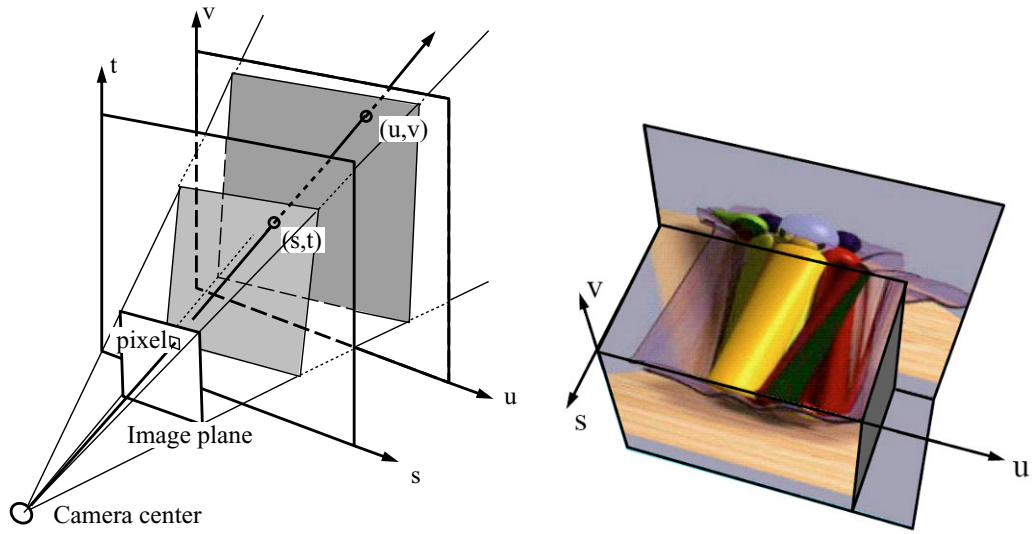


Figure 13.7 The Lumigraph (Gortler, Grzeszczuk, Szeliski *et al.* 1996) © 1996 ACM: (a) a ray is represented by its 4D two-plane parameters (s, t) and (u, v); (b) a slice through the 3D light field subset (u, v, s).

To make the parameterization of this 4D function simpler, let us put two planes in the 3D scene roughly bounding the area of interest, as shown in Figure 13.7a. Any light ray terminating at a camera that lives in front of the st plane (assuming that this space is empty) passes through the two planes at (s, t) and (u, v) and can be described by its 4D coordinate (s, t, u, v). This diagram (and parameterization) can be interpreted as describing a family of cameras living on the st plane with their image planes being the uv plane. The uv plane can be placed at infinity, which corresponds to all the virtual cameras looking in the same direction.

In practice, if the planes are of finite extent, the finite *light slab* $L(s, t, u, v)$ can be used to generate any synthetic view that a camera would see through a (finite) *viewport* in the st plane with a view frustum that wholly intersects the far uv plane. To enable the camera to move all the way around an object, the 3D space surrounding the object can be split into multiple domains, each with its own light slab parameterization. Conversely, if the camera is moving inside a bounded volume of free space looking outward, multiple cube faces surrounding the camera can be used as (s, t) planes.

Thinking about 4D spaces is difficult, so let us drop our visualization by one dimension. If we fix the row value t and constrain our camera to move along the s axis while looking at the uv plane, we can stack all of the stabilized images the camera sees to get the (u, v, s) *epipolar volume*, which we discussed in Section 11.6. A "horizontal" cross-section through this volume is the well-known *epipolar plane image* (Bolles, Baker, and Marimont 1987), which is the us slice shown in Figure 13.7b.

As you can see in this slice, each color pixel moves along a linear track whose slope is related to its depth (parallax) from the uv plane. (Pixels exactly on the uv plane appear "vertical", i.e., they do not move as the camera moves along s.) Furthermore, pixel tracks

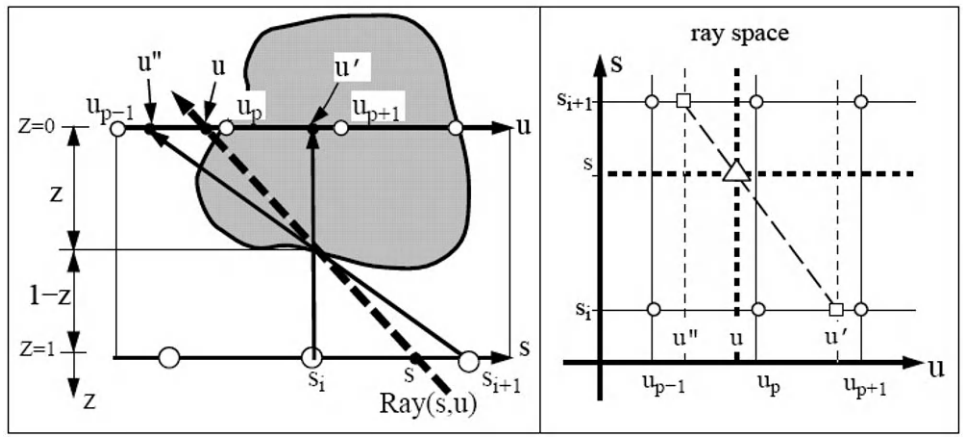


Figure 13.8 Depth compensation in the Lumigraph (Gortler, Grzeszczuk, Szeliski *et al.* 1996) © 1996 ACM. To resample the (s, u) dashed light ray, the u parameter corresponding to each discrete s_i camera location is modified according to the out-of-plane depth z to yield new coordinates u and u'; in (u, s) ray space, the original sample ($\triangle$) is resampled from the (s_i, u') and (s_{i+1}, u'') samples, which are themselves linear blends of their adjacent ($\circ$) samples.

occlude one another as their corresponding 3D surface elements occlude. Translucent pixels, however, composite *over* background pixels (Section 3.1.3, (3.8)) rather than occluding them. Thus, we can think of adjacent pixels sharing a similar planar geometry as *EPI strips* or *EPI tubes* (Criminisi, Kang, Swaminathan *et al.* 2005).

The equations mapping from pixels (x, y) in a virtual camera and the corresponding (s, t, u, v) coordinates are relatively straightforward to derive and are sketched out in Exercise 13.7. It is also possible to show that the set of pixels corresponding to a regular orthographic or perspective camera, i.e., one that has a linear projective relationship between 3D points and (x, y) pixels (2.63), lie along a two-dimensional hyperplane in the (s, t, u, v) light field (Exercise 13.7).

While a light field can be used to render a complex 3D scene from novel viewpoints, a much better rendering (with less ghosting) can be obtained if something is known about its 3D geometry. The Lumigraph system of Gortler, Grzeszczuk, Szeliski *et al.* (1996) extends the basic light field rendering approach by taking into account the 3D location of surface points corresponding to each 3D ray.

Consider the ray (s, u) corresponding to the dashed line in Figure 13.8, which intersects the object's surface at a distance z from the uv plane. When we look up the pixel's color in camera s_i (assuming that the light field is discretely sampled on a regular 4D (s, t, u, v) grid), the actual pixel coordinate is u', instead of the original u value specified by the (s, u) ray. Similarly, for camera s_{i+1} (where $s_i \leq s \leq s_{i+1}$), pixel address u'' is used. Thus, instead of using quadri-linear interpolation of the nearest sampled (s, t, u, v) values around a given ray to determine its color, the (u, v) values are modified for each discrete (s_i, t_i) camera.

Figure 13.8 also shows the same reasoning in *ray space*. Here, the original continuous-valued (s, u) ray is represented by a triangle and the nearby sampled discrete values are shown as circles. Instead of just blending the four nearest samples, as would be indicated

by the vertical and horizontal dashed lines, the modified (s_i, u') and (s_{i+1}, u'') values are sampled instead and their values are then blended.

The resulting rendering system produces images of much better quality than a proxy-free light field and is the method of choice whenever 3D geometry can be inferred. In subsequent work, Isaksen, McMillan, and Gortler (2000) show how a planar proxy for the scene, which is a simpler 3D model, can be used to simplify the resampling equations. They also describe how to create synthetic aperture photos, which mimic what might be seen by a wide-aperture lens, by blending more nearby samples (Levoy and Hanrahan 1996). A similar approach can be used to re-focus images taken with a plenoptic (microlens array) camera (Ng, Levoy, Bréedif *et al.* 2005; Ng 2005) or a light field microscope (Levoy, Ng, Adams *et al.* 2006). It can also be used to see through obstacles, using extremely large synthetic apertures focused on a background that can blur out foreground objects and make them appear translucent (Wilburn, Joshi, Vaish *et al.* 2005; Vaish, Szeliski, Zitnick *et al.* 2006).

Now that we understand how to render new images from a light field, how do we go about capturing such data sets? One answer is to move a calibrated camera with a motion control rig or *gantry*.[6] Another approach is to take handheld photographs and to determine the pose and intrinsic calibration of each image using either a calibrated stage or structure from motion. In this case, the images need to be *rebinned* into a regular 4D (s, t, u, v) space before they can be used for rendering (Gortler, Grzeszczuk, Szeliski *et al.* 1996). Alternatively, the original images can be used directly using a process called the *unstructured Lumigraph*, which we describe below.

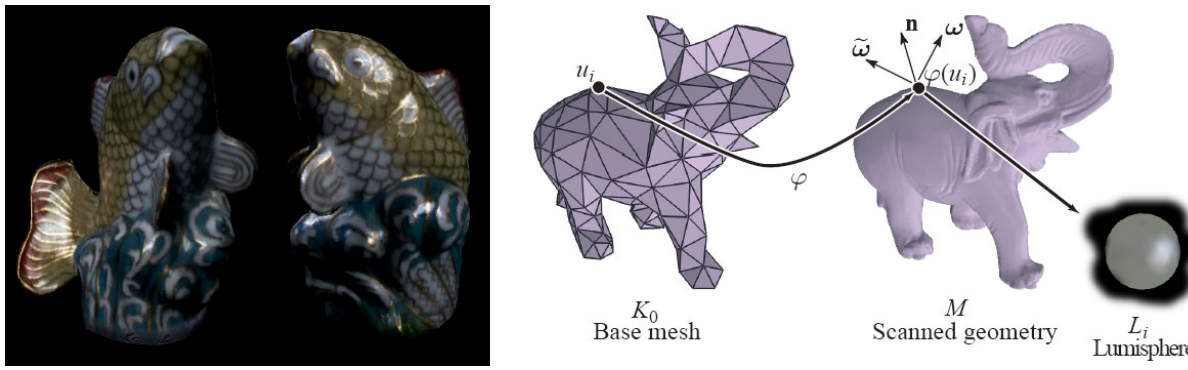
Because of the large number of images involved, light fields and Lumigraphs can be quite voluminous to store and transmit. Fortunately, as you can tell from Figure 13.7b, there is a tremendous amount of redundancy (coherence) in a light field, which can be made even more explicit by first computing a 3D model, as in the Lumigraph. A number of techniques have been developed to compress and progressively transmit such representations (Gortler, Grzeszczuk, Szeliski *et al.* 1996; Levoy and Hanrahan 1996; Rademacher and Bishop 1998; Magnor and Girod 2000; Wood, Azuma, Aldinger *et al.* 2000; Shum, Kang, and Chan 2003; Magnor, Ramanathan, and Girod 2003; Shum, Chan, and Kang 2007).

13.3.1 Unstructured Lumigraph

When the images in a Lumigraph are acquired in an unstructured (irregular) manner, it can be counterproductive to resample the resulting light rays into a regularly binned (s, t, u, v) data structure. This is both because resampling always introduces a certain amount of aliasing and because the resulting gridded light field can be populated very sparsely or irregularly.

The alternative is to render directly from the acquired images, by finding for each light ray in a virtual camera the closest pixels in the original images. The *unstructured Lumigraph* rendering (ULR) system of Buehler, Bosse, McMillan *et al.* (2001) describes how to select such pixels by combining a number of fidelity criteria, including *epipole consistency* (distance of rays to a source camera's center), *angular deviation* (similar incidence direction on the surface), *resolution* (similar sampling density along the surface), *continuity* (to nearby pixels), and *consistency* (along the ray). These criteria can all be combined to determine a weighting

[6] See http://lightfield.stanford.edu/acq.html for a description of some of the gantries and camera arrays built at the Stanford Computer Graphics Laboratory. This Web site also provides a number of light field data sets that are a great source of research and project material.



(a) (b)

Figure 13.9 Surface light fields (Wood, Azuma, Aldinger *et al.* 2000) © 2000 ACM: (a) example of a highly specular object with strong inter-reflections; (b) the surface light field stores the light emanating from each surface point in all visible directions as a "Lumisphere".

function between each virtual camera's pixel and a number of candidate input cameras from which it can draw colors. To make the algorithm more efficient, the computations are performed by discretizing the virtual camera's image plane using a regular grid overlaid with the polyhedral object mesh model and the input camera centers of projection and interpolating the weighting functions between vertices.

The unstructured Lumigraph generalizes previous work in both image-based rendering and light field rendering. When the input cameras are gridded, the ULR behaves the same way as regular Lumigraph rendering. When fewer cameras are available but the geometry is accurate, the algorithm behaves similarly to view-dependent texture mapping (Section 13.1.1).

13.3.2 Surface light fields

Of course, using a two-plane parameterization for a light field is not the only possible choice. (It is the one usually presented first since the projection equations and visualizations are the easiest to draw and understand.) As we mentioned on the topic of light field compression, if we know the 3D shape of the object or scene whose light field is being modeled, we can effectively compress the field because nearby rays emanating from nearby surface elements have similar color values.

In fact, if the object is totally diffuse, ignoring occlusions, which can be handled using 3D graphics algorithms or z-buffering, all rays passing through a given surface point will have the same color value. Hence, the light field "collapses" to the usual 2D texture-map defined over an object's surface. Conversely, if the surface is totally specular (e.g., mirrored), each surface point reflects a miniature copy of the environment surrounding that point. In the absence of inter-reflections (e.g., a convex object in a large open space), each surface point simply reflects the far-field *environment map* (Section 2.2.1), which again is two-dimensional. Therefore, is seems that re-parameterizing the 4D light field to lie on the object's surface can be extremely beneficial.

These observations underlie the *surface light field* representation introduced by Wood, Azuma, Aldinger *et al.* (2000). In their system, an accurate 3D model is built of the object

being represented. Then the *Lumisphere* of all rays emanating from each surface point is estimated or captured (Figure 13.9). Nearby Lumispheres will be highly correlated and hence amenable to both compression and manipulation.

To estimate the diffuse component of each Lumisphere, a median filtering over all visible exiting directions is first performed for each channel. Once this has been subtracted from the Lumisphere, the remaining values, which should consist mostly of the specular components, are *reflected* around the local surface normal (2.89), which turns each Lumisphere into a copy of the local environment around that point. Nearby Lumispheres can then be compressed using predictive coding, vector quantization, or principal component analysis.

The decomposition into a diffuse and specular component can also be used to perform editing or manipulation operations, such as re-painting the surface, changing the specular component of the reflection (e.g., by blurring or sharpening the specular Lumispheres), or even geometrically deforming the object while preserving detailed surface appearance.

13.3.3 *Application*: Concentric mosaics

A useful and simple version of light field rendering is a panoramic image with parallax, i.e., a video or series of photographs taken from a camera swinging in front of some rotation point. Such panoramas can be captured by placing a camera on a boom on a tripod, or even more simply, by holding a camera at arm's length while rotating your body around a fixed axis.

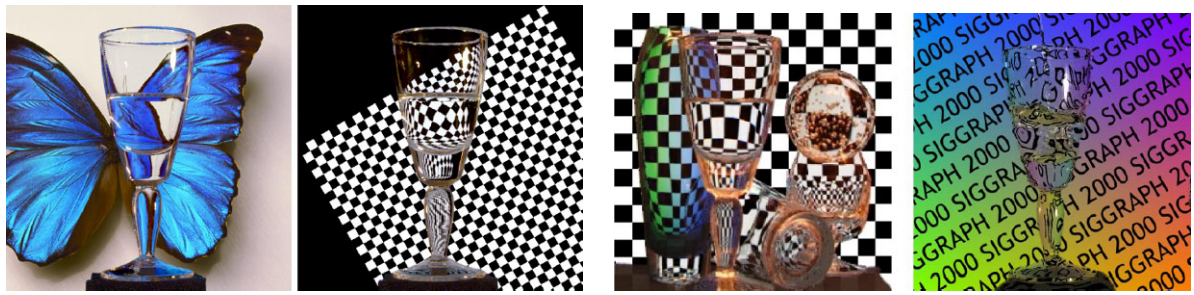
The resulting set of images can be thought of as a *concentric mosaic* (Shum and He 1999; Shum, Wang, Chai *et al.* 2002) or a *layered depth panorama* (Zheng, Kang, Cohen *et al.* 2007). The term "concentric mosaic" comes from a particular structure that can be used to re-bin all of the sampled rays, essentially associating each column of pixels with the "radius" of the concentric circle to which it is tangent (Shum and He 1999; Peleg, Ben-Ezra, and Pritch 2001).

Rendering from such data structures is fast and straightforward. If we assume that the scene is far enough away, for any virtual camera location, we can associate each column of pixels in the virtual camera with the nearest column of pixels in the input image set. (For a regularly captured set of images, this computation can be performed analytically.) If we have some rough knowledge of the depth of such pixels, columns can be stretched vertically to compensate for the change in depth between the two cameras. If we have an even more detailed depth map (Peleg, Ben-Ezra, and Pritch 2001; Li, Shum, Tang *et al.* 2004; Zheng, Kang, Cohen *et al.* 2007), we can perform pixel-by-pixel depth corrections.

While the virtual camera's motion is constrained to lie in the plane of the original cameras and within the radius of the original capture ring, the resulting experience can exhibit complex rendering phenomena, such as reflections and translucencies, which cannot be captured using a texture-mapped 3D model of the world. Exercise 13.10 has you construct a concentric mosaic rendering system from a series of hand-held photos or video.

13.4 Environment mattes

So far in this chapter, we have dealt with view interpolation and light fields, which are techniques for modeling and rendering complex static scenes seen from different viewpoints.



(a) (b) (c) (d)

Figure 13.10 Environment mattes: (a–b) a refractive object can be placed in front of a series of backgrounds and their light patterns will be correctly refracted (Zongker, Werner, Curless *et al.* 1999) (c) multiple refractions can be handled using a mixture of Gaussians model and (d) real-time mattes can be pulled using a single graded colored background (Chuang, Zongker, Hindorff *et al.* 2000) © 2000 ACM.

What if instead of moving around a virtual camera, we take a complex, refractive object, such as the water goblet shown in Figure 13.10, and place it in front of a new background? Instead of modeling the 4D space of rays emanating from a scene, we now need to model how each pixel in our view of this object refracts incident light coming from its environment.

What is the intrinsic dimensionality of such a representation and how do we go about capturing it? Let us assume that if we trace a light ray from the camera at pixel (x, y) toward the object, it is reflected or refracted back out toward its environment at an angle (ϕ, θ). If we assume that other objects and illuminants are sufficiently distant (the same assumption we made for surface light fields in Section 13.3.2), this 4D mapping $(x, y) \rightarrow (\phi, \theta)$ captures all the information between a refractive object and its environment. Zongker, Werner, Curless *et al.* (1999) call such a representation an *environment matte*, since it generalizes the process of object matting (Section 10.4) to not only cut and paste an object from one image into another but also take into account the subtle refractive or reflective interplay between the object and its environment.

Recall from Equations (3.8) and (10.30) that a foreground object can be represented by its premultiplied colors and opacities $(\alpha F, \alpha)$. Such a matte can then be composited onto a new background B using

$$C_i = \alpha_i F_i + (1 - \alpha_i) B_i, \tag{13.1}$$

where i is the pixel under consideration. In environment matting, we augment this equation with a reflective or refractive term to model indirect light paths between the environment and the camera. In the original work of Zongker, Werner, Curless *et al.* (1999), this indirect component I_i is modeled as

$$I_i = R_i \int A_i(\boldsymbol{x}) B(\boldsymbol{x}) d\boldsymbol{x}, \tag{13.2}$$

where A_i is the rectangular *area of support* for that pixel, R_i is the colored reflectance or transmittance (for colored glossy surfaces or glass), and $B(\boldsymbol{x})$ is the background (environment) image, which is integrated over the area $A_i(\boldsymbol{x})$. In follow-on work, Chuang, Zongker,

Hindorff *et al.* (2000) use a superposition of oriented Gaussians,

$$I_i = \sum_j R_{ij} \int G_{ij}(\boldsymbol{x}) B(\boldsymbol{x}) d\boldsymbol{x}, \tag{13.3}$$

where each 2D Gaussian

$$G_{ij}(\boldsymbol{x}) = G_{2\mathrm{D}}(\boldsymbol{x}; \boldsymbol{c}_{ij}, \boldsymbol{\sigma}_{ij}, \theta_{ij}) \tag{13.4}$$

is modeled by its center $\boldsymbol{c}_{ij}$, unrotated widths $\boldsymbol{\sigma}_{ij} = (\sigma_{ij}^x, \sigma_{ij}^y)$, and orientation θ_{ij}.

Given a representation for an environment matte, how can we go about estimating it for a particular object? The trick is to place the object in front of a monitor (or surrounded by a set of monitors), where we can change the illumination patterns $B(\boldsymbol{x})$ and observe the value of each composite pixel C_i.[7]

As with traditional two-screen matting (Section 10.4.1), we can use a variety of solid colored backgrounds to estimate each pixel's foreground color $\alpha_i F_i$ and partial coverage (opacity) α_i. To estimate the area of support A_i in (13.2), Zongker, Werner, Curless *et al.* (1999) use a series of periodic horizontal and vertical solid stripes at different frequencies and phases, which is reminiscent of the structured light patterns used in active rangefinding (Section 12.2). For the more sophisticated mixture of Gaussian model (13.3), Chuang, Zongker, Hindorff *et al.* (2000) sweep a series of narrow Gaussian stripes at four different orientations (horizontal, vertical, and two diagonals), which enables them to estimate multiple oriented Gaussian responses at each pixel.

Once an environment matte has been "pulled", it is then a simple matter to replace the background with a new image $B(\boldsymbol{x})$ to obtain a novel composite of the object placed in a different environment (Figure 13.10a–c). The use of multiple backgrounds during the matting process, however, precludes the use of this technique with dynamic scenes, e.g., water pouring into a glass (Figure 13.10d). In this case, a single graded color background can be used to estimate a single 2D monochromatic displacement for each pixel (Chuang, Zongker, Hindorff *et al.* 2000).

13.4.1 Higher-dimensional light fields

As you can tell from the preceding discussion, an environment matte in principle maps every pixel (x, y) into a 4D distribution over light rays and is, hence, a six-dimensional representation. (In practice, each 2D pixel's response is parameterized using a dozen or so parameters, e.g., $\{F, \alpha, B, R, A\}$, instead of a full mapping.) What if we want to model an object's refractive properties from every potential point of view? In this case, we need a mapping from every incoming 4D light ray to every potential exiting 4D light ray, which is an 8D representation. If we use the same trick as with surface light fields, we can parameterize each surface point by its 4D BRDF to reduce this mapping back down to 6D but this loses the ability to handle multiple refractive paths.

If we want to handle dynamic light fields, we need to add another temporal dimension. (Wenger, Gardner, Tchou *et al.* (2005) gives a nice example of a dynamic appearance and illumination acquisition system.) Similarly, if we want a continuous distribution over wavelengths, this becomes another dimension.

[7] If we relax the assumption that the environment is distant, the monitor can be placed at several depths to estimate a depth-dependent mapping function (Zongker, Werner, Curless *et al.* 1999).

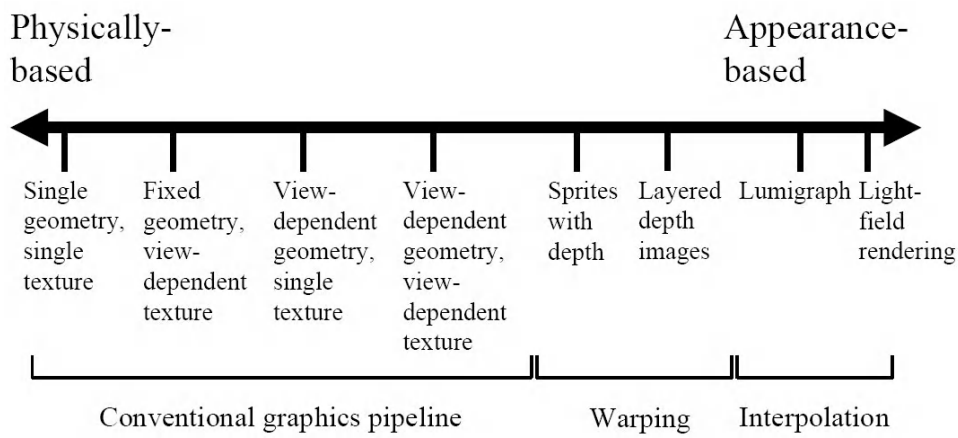


Figure 13.11 The geometry-image continuum in image-based rendering (Kang, Szeliski, and Anandan 2000) © 2000 IEEE. Representations at the left of the spectrum use more detailed geometry and simpler image representations, while representations and algorithms on the right use more images and less geometry.

These examples illustrate how modeling the full complexity of a visual scene through sampling can be extremely expensive. Fortunately, constructing specialized models, which exploit knowledge about the physics of light transport along with the natural coherence of real-world objects, can make these problems more tractable.

13.4.2 The modeling to rendering continuum

The image-based rendering representations and algorithms we have studied in this chapter span a continuum ranging from classic 3D texture-mapped models all the way to pure sampled ray-based representations such as light fields (Figure 13.11). Representations such as view-dependent texture maps and Lumigraphs still use a single global geometric model, but select the colors to map onto these surfaces from nearby images. View-dependent geometry, e.g., multiple depth maps, sidestep the need for coherent 3D geometry, and can sometimes better model local non-rigid effects such as specular motion (Swaminathan, Kang, Szeliski *et al.* 2002; Criminisi, Kang, Swaminathan *et al.* 2005). Sprites with depth and layered depth images use image-based representations of both color and geometry and can be efficiently rendered using warping operations rather than 3D geometric rasterization.

The best choice of representation and rendering algorithm depends on both the quantity and quality of the input imagery as well as the intended application. When nearby views are being rendered, image-based representations capture more of the visual fidelity of the real world because they directly sample its appearance. On the other hand, if only a few input images are available or the image-based models need to be manipulated, e.g., to change their shape or appearance, more abstract 3D representations such as geometric and local reflection models are a better fit. As we continue to capture and manipulate increasingly larger quantities of visual data, research into these aspects of image-based modeling and rendering will continue to evolve.



Figure 13.12 Video Rewrite (Bregler, Covell, and Slaney 1997) © 1997 ACM: the video frames are composed from bits and pieces of old video footage matched to a new audio track.

13.5 Video-based rendering

Since multiple images can be used to render new images or interactive experiences, can something similar be done with video? In fact, a fair amount of work has been done in the area of *video-based rendering* and *video-based animation*, two terms first introduced by Schödl, Szeliski, Salesin *et al.* (2000) to denote the process of generating new video sequences from captured video footage. An early example of such work is Video Rewrite (Bregler, Covell, and Slaney 1997), in which archival video footage is "re-animated" by having actors say new utterances (Figure 13.12). More recently, the term video-based rendering has been used by some researchers to denote the creation of virtual camera moves from a set of synchronized video cameras placed in a studio (Magnor 2005). (The terms *free-viewpoint video* and *3D video* are also sometimes used, see Section 13.5.4.)

In this section, we present a number of video-based rendering systems and applications. We start with *video-based animation* (Section 13.5.1), in which video footage is re-arranged or modified, e.g., in the capture and re-rendering of facial expressions. A special case of this are *video textures* (Section 13.5.2), in which source video is automatically cut into segments and re-looped to create infinitely long video animations. It is also possible to create such animations from still pictures or paintings, by segmenting the image into separately moving regions and animating them using stochastic motion fields (Section 13.5.3).

Next, we turn our attention to *3D video* (Section 13.5.4), in which multiple synchronized video cameras are used to film a scene from different directions. The source video frames can then be re-combined using image-based rendering techniques, such as view interpolation, to create virtual camera paths between the source cameras as part of a real-time viewing experience. Finally, we discuss capturing environments by driving or walking through them with panoramic video cameras in order to create interactive video-based walkthrough experiences (Section 13.5.5).

13.5.1 Video-based animation

As we mentioned above, an early example of video-based animation is Video Rewrite, in which frames from original video footage are rearranged in order to match them to novel spoken utterances, e.g., for movie dubbing (Figure 13.12). This is similar in spirit to the way that *concatenative speech synthesis* systems work (Taylor 2009).

In their system, Bregler, Covell, and Slaney (1997) first use speech recognition to extract phonemes from both the source video material and the novel audio stream. Phonemes are grouped into *triphones* (triplets of phonemes), since these better model the *coarticulation* effect present when people speak. Matching triphones are then found in the source footage and audio track. The mouth images corresponding to the selected video frames are then cut and pasted into the desired video footage being re-animated or dubbed, with appropriate geometric transformations to account for head motion. During the analysis phase, features corresponding to the lips, chin, and head are tracked using computer vision techniques. During synthesis, image morphing techniques are used to blend and stitch adjacent mouth shapes into a more coherent whole. In more recent work, Ezzat, Geiger, and Poggio (2002) describe how to use a *multidimensional morphable model* (Section 12.6.2) combined with regularized trajectory synthesis to improve these results.

A more sophisticated version of this system, called *face transfer*, uses a novel source video, instead of just an audio track, to drive the animation of a previously captured video, i.e., to re-render a video of a talking head with the appropriate visual speech, expression, and head pose elements (Vlasic, Brand, Pfister *et al.* 2005). This work is one of many *performance-driven animation* systems (Section 4.1.5), which are often used to animate 3D facial models (Figures 12.18–12.19). While traditional performance-driven animation systems use marker-based motion capture (Williams 1990; Litwinowicz and Williams 1994; Ma, Jones, Chiang *et al.* 2008), video footage can now often be used directly to control the animation (Buck, Finkelstein, Jacobs *et al.* 2000; Pighin, Szeliski, and Salesin 2002; Zhang, Snavely, Curless *et al.* 2004; Vlasic, Brand, Pfister *et al.* 2005; Roble and Zafar 2009).

In addition to its most common application to facial animation, video-based animation can also be applied to whole body motion (Section 12.6.4), e.g., by matching the flow fields between two different source videos and using one to drive the other (Efros, Berg, Mori *et al.* 2003). Another approach to video-based rendering is to use flow or 3D modeling to *unwrap* surface textures into stabilized images, which can then be manipulated and re-rendered onto the original video (Pighin, Szeliski, and Salesin 2002; Rav-Acha, Kohli, Fitzgibbon *et al.* 2008).

13.5.2 Video textures

Video-based animation is a powerful means of creating photo-realistic videos by re-purposing existing video footage to match some other desired activity or script. What if instead of constructing a special animation or narrative, we simply want the video to continue playing in a plausible manner? For example, many Web sites use images or videos to highlight their destinations, e.g., to portray attractive beaches with surf and palm trees waving in the wind. Instead of using a static image or a video clip that has a discontinuity when it loops, can we transform the video clip into an infinite-length animation that plays forever?

This idea is the basis of *video textures*, in which a short video clip can be arbitrarily extended by re-arranging video frames while preserving visual continuity (Schödl, Szeliski, Salesin *et al.* 2000). The basic problem in creating video textures is how to perform this re-arrangement without introducing visual artifacts. Can you think of how you might do this?

The simplest approach is to match frames by visual similarity (e.g., L_2 distance) and to jump between frames that appear similar. Unfortunately, if the motions in the two frames are different, a dramatic visual artifact will occur (the video will appear to "stutter"). For



Figure 13.13 Video textures (Schödl, Szeliski, Salesin *et al.* 2000) © 2000 ACM: (a) a clock pendulum, with correctly matched direction of motion; (b) a candle flame, showing temporal transition arcs; (c) the flag is generated using morphing at jumps; (d) a bonfire uses longer cross-dissolves; (e) a waterfall cross-dissolves several sequences at once; (f) a smiling animated face; (g) two swinging children are animated separately; (h) the balloons are automatically segmented into separate moving regions; (i) a synthetic fish tank consisting of bubbles, plants, and fish. Videos corresponding to these images can be found at http://www.cc.gatech.edu/gvu/perception/projects/videotexture/.

example, if we fail to match the motions of the clock pendulum in Figure 13.13a, it can suddenly change direction in mid-swing.

How can we extend our basic frame matching to also match motion? In principle, we could compute optic flow at each frame and match this. However, flow estimates are often unreliable (especially in textureless regions) and it is not clear how to weight the visual and motion similarities relative to each other. As an alternative, Schödl, Szeliski, Salesin *et al.* (2000) suggest matching *triplets* or larger neighborhoods of adjacent video frames, much in the same way as Video Rewrite matches triphones. Once we have constructed an $n \times n$ similarity matrix between all video frames (where n is the number of frames), a simple finite impulse response (FIR) filtering of each match sequence can be used to emphasize subsequences that match well.

The results of this match computation gives us a *jump table* or, equivalently, a transition probability between any two frames in the original video. This is shown schematically as red arcs in Figure 13.13b, where the red bar indicates which video frame is currently being displayed, and arcs light up as a forward or backward transition is taken. We can view these transition probabilities as encoding the *hidden Markov model* (HMM) that underlies a stochastic video generation process.

Sometimes, it is not possible to find exactly matching subsequences in the original video. In this case, morphing, i.e., warping and blending frames during transitions (Section 3.6.3) can be used to hide the visual differences (Figure 13.13c). If the motion is chaotic enough, as in a bonfire or a waterfall (Figures 13.13d–e), simple blending (extended cross-dissolves) may be sufficient. Improved transitions can also be obtained by performing 3D graph cuts on the spatio-temporal volume around a transition (Kwatra, Schödl, Essa *et al.* 2003).

Video textures need not be restricted to chaotic random phenomena such as fire, wind, and water. Pleasing video textures can be created of people, e.g., a smiling face (as in Figure 13.13f) or someone running on a treadmill (Schödl, Szeliski, Salesin *et al.* 2000). When multiple people or objects are moving independently, as in Figures 13.13g–h, we must first segment the video into independently moving regions and animate each region separately. It is also possible to create large panoramic video textures from a slowly panning camera (Agarwala, Zheng, Pal *et al.* 2005).

Instead of just playing back the original frames in a stochastic (random) manner, video textures can also be used to create scripted or interactive animations. If we extract individual elements, such as fish in a fishtank (Figure 13.13i) into separate *video sprites*, we can animate them along pre-specified paths (by matching the path direction with the original sprite motion) to make our video elements move in a desired fashion (Schödl and Essa 2002). In fact, work on video textures inspired research on systems that re-synthesize new motion sequences from motion capture data, which some people refer to as "mocap soup" (Arikan and Forsyth 2002; Kovar, Gleicher, and Pighin 2002; Lee, Chai, Reitsma *et al.* 2002; Li, Wang, and Shum 2002; Pullen and Bregler 2002).

While video textures primarily analyze the video as a sequence of frames (or regions) that can be re-arranged in time, *temporal textures* (Szummer and Picard 1996; Bar-Joseph, El-Yaniv, Lischinski *et al.* 2001) and *dynamic textures* (Doretto, Chiuso, Wu *et al.* 2003; Yuan, Wen, Liu *et al.* 2004; Doretto and Soatto 2006) treat the video as a 3D spatio-temporal volume with textural properties, which can be described using auto-regressive temporal models.

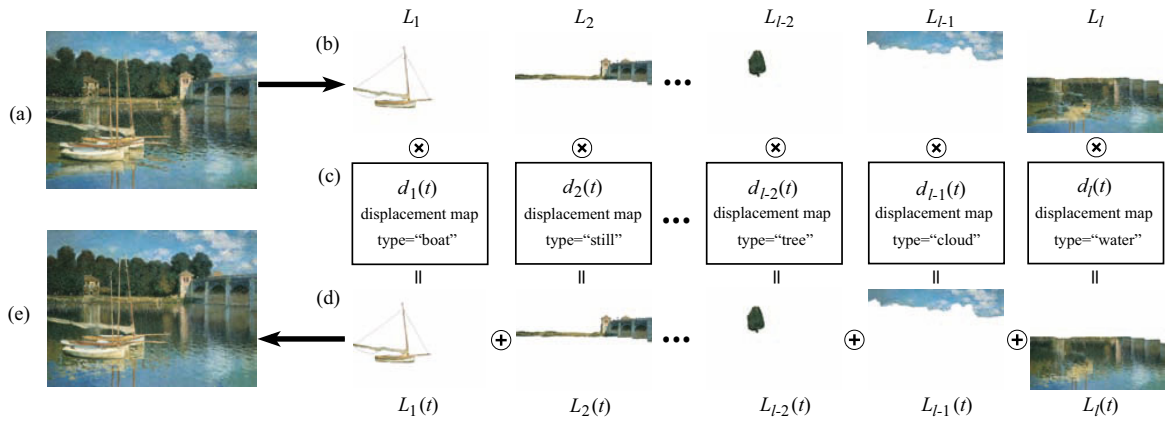


Figure 13.14 Animating still pictures (Chuang, Goldman, Zheng *et al.* 2005) © 2005 ACM. (a) The input still image is manually segmented into (b) several layers. (c) Each layer is then animated with a different stochastic motion texture (d) The animated layers are then composited to produce (e) the final animation

13.5.3 *Application*: Animating pictures

While video textures can turn a short video clip into an infinitely long video, can the same thing be done with a single still image? The answer is yes, if you are willing to first segment the image into different layers and then animate each layer separately.

Chuang, Goldman, Zheng *et al.* (2005) describe how an image can be decomposed into separate layers using interactive matting techniques. Each layer is then animated using a class-specific synthetic motion. As shown in Figure 13.14, boats rock back and forth, trees sway in the wind, clouds move horizontally, and water ripples, using a shaped noise displacement map. All of these effects can be tied to some global control parameters, such as the velocity and direction of a virtual wind. After being individually animated, the layers can be composited to create a final dynamic rendering.

13.5.4 3D Video

In recent years, the popularity of 3D movies has grown dramatically, with recent releases ranging from *Hannah Montana*, through U2's 3D concert movie, to James Cameron's *Avatar*. Currently, such releases are filmed using stereoscopic camera rigs and displayed in theaters (or at home) to viewers wearing polarized glasses.[8] In the future, however, home audiences may wish to view such movies with multi-zone auto-stereoscopic displays, where each person gets his or her own customized stereo stream and can move around a scene to see it from different perspectives.[9]

The stereo matching techniques developed in the computer vision community along with image-based rendering (view interpolation) techniques from graphics are both essential components in such scenarios, which are sometimes called *free-viewpoint video* (Carranza, Theobalt, Magnor *et al.* 2003) or *virtual viewpoint video* (Zitnick, Kang, Uyttendaele *et al.* 2004). In

[8] http://www.3d-summit.com/.
[9] http://www.siggraph.org/s2008/attendees/caf/3d/.

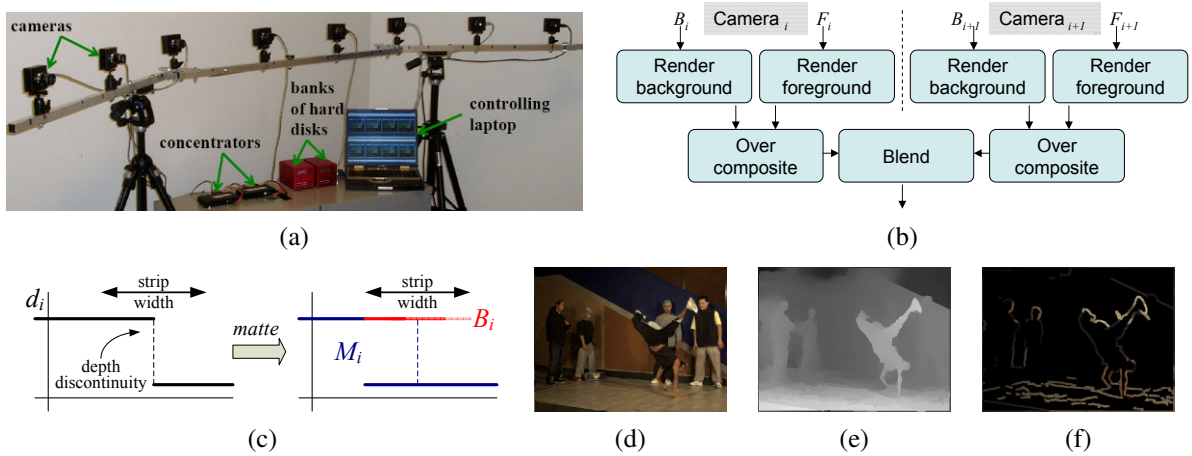


Figure 13.15 Video view interpolation (Zitnick, Kang, Uyttendaele *et al.* 2004) © 2004 ACM: (a) the capture hardware consists of eight synchronized cameras; (b) the background and foreground images from each camera are rendered and composited before blending; (c) the two-layer representation, before and after boundary matting; (d) background color estimates; (e) background depth estimates; (f) foreground color estimates.

addition to solving a series of per-frame reconstruction and view interpolation problems, the depth maps or proxies produced by the analysis phase must be temporally consistent in order to avoid flickering artifacts.

Shum, Chan, and Kang (2007) and Magnor (2005) present nice overviews of various video view interpolation techniques and systems. These include the Virtualized Reality system of Kanade, Rander, and Narayanan (1997) and Vedula, Baker, and Kanade (2005), Immersive Video (Moezzi, Katkere, Kuramura *et al.* 1996), Image-Based Visual Hulls (Matusik, Buehler, Raskar *et al.* 2000; Matusik, Buehler, and McMillan 2001), and Free-Viewpoint Video (Carranza, Theobalt, Magnor *et al.* 2003), which all use global 3D geometric models (surface-based (Section 12.3) or volumetric (Section 12.5)) as their proxies for rendering. The work of Vedula, Baker, and Kanade (2005) also computes *scene flow*, i.e., the 3D motion between corresponding surface elements, which can then be used to perform spatio-temporal interpolation of the multi-view video stream.

The Virtual Viewpoint Video system of Zitnick, Kang, Uyttendaele *et al.* (2004), on the other hand, associates a two-layer depth map with each input image, which allows them to accurately model occlusion effects such as the mixed pixels that occur at object boundaries. Their system, which consists of eight synchronized video cameras connected to a disk array (Figure 13.15a), first uses segmentation-based stereo to extract a depth map for each input image (Figure 13.15e). Near object boundaries (depth discontinuities), the background layer is extended along a strip behind the foreground object (Figure 13.15c) and its color is estimated from the neighboring images where it is not occluded (Figure 13.15d). Automated matting techniques (Section 10.4) are then used to estimate the fractional opacity and color of boundary pixels in the foreground layer (Figure 13.15f).

At render time, given a new virtual camera that lies between two of the original cameras, the layers in the neighboring cameras are rendered as texture-mapped triangles and the foreground layer (which may have fractional opacities) is then composited over the background

layer (Figure 13.15b). The resulting two images are merged and blended by comparing their respective z-buffer values. (Whenever the two z-values are sufficiently close, a linear blend of the two colors is computed.) The interactive rendering system runs in real time using regular graphics hardware. It can therefore be used to change the observer's viewpoint while playing the video or to freeze the scene and explore it in 3D. More recently, Rogmans, Lu, Bekaert *et al.* (2009) have developed GPU implementations of both real-time stereo matching and real-time rendering algorithms, which enable them to explore algorithmic alternatives in a real-time setting.

At present, the depth maps computed from the eight stereo cameras using off-line stereo matching have produced the highest quality depth maps associated with live video.[10] They are therefore often used in studies of 3D video compression, which is an active area of research (Smolic and Kauff 2005; Gotchev and Rosenhahn 2009). Active video-rate depth sensing cameras, such as the 3DV Zcam (Iddan and Yahav 2001), which we discussed in Section 12.2.1, are another potential source of such data.

When large numbers of closely spaced cameras are available, as in the Stanford Light Field Camera (Wilburn, Joshi, Vaish *et al.* 2005), it may not always be necessary to compute explicit depth maps to create video-based rendering effects, although the results are usually of higher quality if you do (Vaish, Szeliski, Zitnick *et al.* 2006).

13.5.5 *Application*: Video-based walkthroughs

Video camera arrays enable the simultaneous capture of 3D dynamic scenes from multiple viewpoints, which can then enable the viewer to explore the scene from viewpoints near the original capture locations. What if instead we wish to capture an extended area, such as a home, a movie set, or even an entire city?

In this case, it makes more sense to move the camera through the environment and play back the video as an interactive video-based walkthrough. In order to allow the viewer to look around in all directions, it is preferable to use a panoramic video camera (Uyttendaele, Criminisi, Kang *et al.* 2004).[11]

One way to structure the acquisition process is to capture these images in a 2D horizontal plane, e.g., over a grid superimposed inside a room. The resulting *sea of images* (Aliaga, Funkhouser, Yanovsky *et al.* 2003) can be used to enable continuous motion between the captured locations.[12] However, extending this idea to larger settings, e.g., beyond a single room, can become tedious and data-intensive.

Instead, a natural way to explore a space is often to just walk through it along some prespecified paths, just as museums or home tours guide users along a particular path, say down the middle of each room.[13] Similarly, city-level exploration can be achieved by driving down the middle of each street and allowing the user to branch at each intersection. This idea dates back to the Aspen MovieMap project (Lippman 1980), which recorded analog video taken from moving cars onto videodiscs for later interactive playback.

[10] http://research.microsoft.com/en-us/um/redmond/groups/ivm/vvv/.

[11] See http://www.cis.upenn.edu/~kostas/omni.html for descriptions of panoramic (omnidirectional) vision systems and associated workshops.

[12] (The Photo Tourism system of Snavely, Seitz, and Szeliski (2006) applies this idea to less structured collections.

[13] In computer games, restricting a player to forward and backward motion along predetermined paths is called *rail-based gaming*.

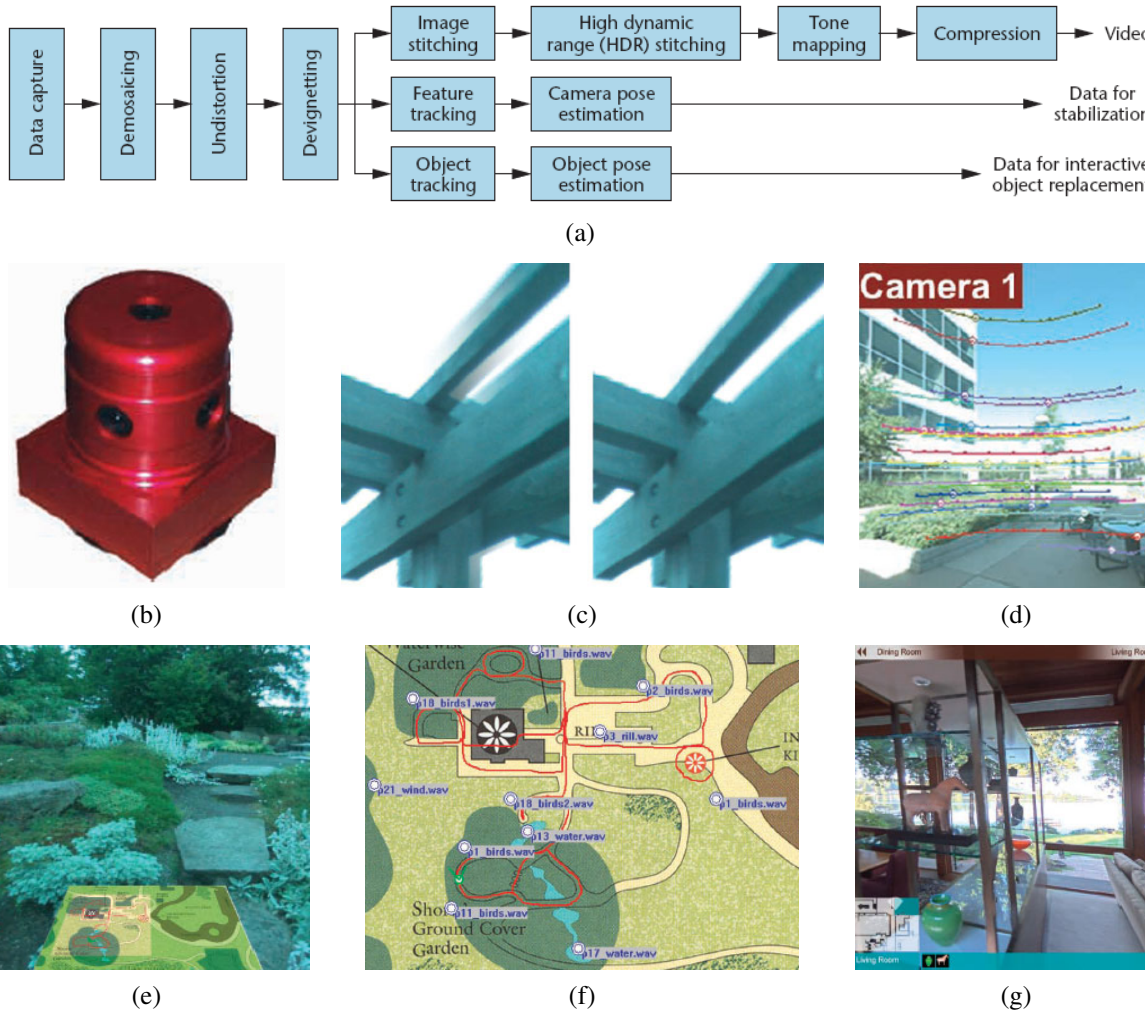


Figure 13.16 Video-based walkthroughs (Uyttendaele, Criminisi, Kang *et al.* 2004) © 2004 IEEE: (a) system diagram of video pre-processing; (b) the Point Grey Ladybug camera; (c) ghost removal using multi-perspective plane sweep; (d) point tracking, used both for calibration and stabilization; (e) interactive garden walkthrough with map below; (f) overhead map authoring and sound placement; (g) interactive home walkthrough with navigation bar (top) and icons of interest (bottom).

Recent improvements in video technology now enable the capture of panoramic (spherical) video using a small co-located array of cameras, such as the Point Grey Ladybug camera[14] (Figure 13.16b) developed by Uyttendaele, Criminisi, Kang *et al.* (2004) for their interactive video-based walkthrough project. In their system, the synchronized video streams from the six cameras (Figure 13.16a) are stitched together into 360° panoramas using a variety of techniques developed specifically for this project.

Because the cameras do not share the same center of projection, parallax between the cameras can lead to ghosting in the overlapping fields of view (Figure 13.16c). To remove this, a multi-perspective plane sweep stereo algorithm is used to estimate per-pixel depths at each column in the overlap area. To calibrate the cameras relative to each other, the camera is spun in place and a constrained structure from motion algorithm (Figure 7.8) is used to estimate the relative camera poses and intrinsics. Feature tracking is then run on the walk-through video in order to stabilize the video sequence—Liu, Gleicher, Jin *et al.* (2009) have carried out more recent work along these lines.

Indoor environments with windows, as well as sunny outdoor environments with strong shadows, often have a dynamic range that exceeds the capabilities of video sensors. For this reason, the Ladybug camera has a programmable exposure capability that enables the bracketing of exposures at subsequent video frames. In order to merge the resulting video frames into high dynamic range (HDR) video, pixels from adjacent frames need to be motion-compensated before being merged (Kang, Uyttendaele, Winder *et al.* 2003).

The interactive walk-through experience becomes much richer and more navigable if an overview map is available as part of the experience. In Figure 13.16f, the map has annotations, which can show up during the tour, and localized sound sources, which play (with different volumes) when the viewer is nearby. The process of aligning the video sequence with the map can be automated using a process called *map correlation* (Levin and Szeliski 2004).

All of these elements combine to provide the user with a rich, interactive, and immersive experience. Figure 13.16e shows a walk through the Bellevue Botanical Gardens, with an overview map in perspective below the live video window. Arrows on the ground are used to indicate potential directions of travel. The viewer simply orients his view towards one of the arrows (the experience can be driven using a game controller) and "walks" forward along the desired path.

Figure 13.16g shows an indoor home tour experience. In addition to a schematic map in the lower left corner and adjacent room names along the top navigation bar, icons appear along the bottom whenever items of interest, such as a homeowner's art pieces, are visible in the main window. These icons can then be clicked to provide more information and 3D views.

The development of interactive video tours spurred a renewed interest in 360° video-based virtual travel and mapping experiences, as evidenced by commercial sites such as Google's Street View and Bing Maps. The same videos can also be used to generate turn-by-turn driving directions, taking advantage of both expanded fields of view and image-based rendering to enhance the experience (Chen, Neubert, Ofek *et al.* 2009).

As we continue to capture more and more of our real world with large amounts of high-quality imagery and video, the interactive modeling, exploration, and rendering techniques described in this chapter will play an even bigger role in bringing virtual experiences based

[14] http://www.ptgrey.com/.

on remote areas of the world closer to everyone.

13.6 Additional reading

Two good recent surveys of image-based rendering are by Kang, Li, Tong *et al.* (2006) and Shum, Chan, and Kang (2007), with earlier surveys available from Kang (1999), McMillan and Gortler (1999), and Debevec (1999). The term *image-based rendering* was introduced by McMillan and Bishop (1995), although the seminal paper in the field is the view interpolation paper by Chen and Williams (1993). Debevec, Taylor, and Malik (1996) describe their Façade system, which not only created a variety of image-based modeling tools but also introduced the widely used technique of *view-dependent texture mapping*.

Early work on planar impostors and layers was carried out by Shade, Lischinski, Salesin *et al.* (1996), Lengyel and Snyder (1997), and Torborg and Kajiya (1996), while newer work based on *sprites with depth* is described by Shade, Gortler, He *et al.* (1998).

The two foundational papers in image-based rendering are *Light field rendering* by Levoy and Hanrahan (1996) and *The Lumigraph* by Gortler, Grzeszczuk, Szeliski *et al.* (1996). Buehler, Bosse, McMillan *et al.* (2001) generalize the Lumigraph approach to irregularly spaced collections of images, while Levoy (2006) provides a survey and more gentle introduction to the topic of light field and image-based rendering.

Surface light fields (Wood, Azuma, Aldinger *et al.* 2000) provide an alternative parameterization for light fields with accurately known surface geometry and support both better compression and the possibility of editing surface properties. Concentric mosaics (Shum and He 1999; Shum, Wang, Chai *et al.* 2002) and panoramas with depth (Peleg, Ben-Ezra, and Pritch 2001; Li, Shum, Tang *et al.* 2004; Zheng, Kang, Cohen *et al.* 2007), provide useful parameterizations for light fields captured with panning cameras. Multi-perspective images (Rademacher and Bishop 1998) and manifold projections (Peleg and Herman 1997), although not true light fields, are also closely related to these ideas.

Among the possible extensions of light fields to higher-dimensional structures, environment mattes (Zongker, Werner, Curless *et al.* 1999; Chuang, Zongker, Hindorff *et al.* 2000) are the most useful, especially for placing captured objects into new scenes.

Video-based rendering, i.e., the re-use of video to create new animations or virtual experiences, started with the seminal work of Szummer and Picard (1996), Bregler, Covell, and Slaney (1997), and Schödl, Szeliski, Salesin *et al.* (2000). Important follow-on work to these basic re-targeting approaches was carried out by Schödl and Essa (2002), Kwatra, Schödl, Essa *et al.* (2003), Doretto, Chiuso, Wu *et al.* (2003), Wang and Zhu (2003), Zhong and Sclaroff (2003), Yuan, Wen, Liu *et al.* (2004), Doretto and Soatto (2006), Zhao and Pietikäinen (2007), and Chan and Vasconcelos (2009).

Systems that allow users to change their 3D viewpoint based on multiple synchronized video streams include those by Moezzi, Katkere, Kuramura *et al.* (1996), Kanade, Rander, and Narayanan (1997), Matusik, Buehler, Raskar *et al.* (2000), Matusik, Buehler, and McMillan (2001), Carranza, Theobalt, Magnor *et al.* (2003), Zitnick, Kang, Uyttendaele *et al.* (2004), Magnor (2005), and Vedula, Baker, and Kanade (2005). 3D (multiview) video coding and compression is also an active area of research (Smolic and Kauff 2005; Gotchev and Rosenhahn 2009), with 3D Blu-Ray discs, encoded using the multiview video coding (MVC) extension to H.264/MPEG-4 AVC, expected by the end of 2010.

13.7 Exercises

Ex 13.1: Depth image rendering Develop a "view extrapolation" algorithm to re-render a previously computed stereo depth map coupled with its corresponding reference color image.

1. Use a 3D graphics mesh rendering system such as OpenGL or Direct3D, with two triangles per pixel quad and perspective (projective) texture mapping (Debevec, Yu, and Borshukov 1998).
2. Alternatively, use the one- or two-pass forward warper you constructed in Exercise 3.24, extended using (2.68–2.70) to convert from disparities or depths into displacements.
3. (Optional) Kinks in straight lines introduced during view interpolation or extrapolation are visually noticeable, which is one reason why image morphing systems let you specify line correspondences (Beier and Neely 1992). Modify your depth estimation algorithm to match and estimate the geometry of straight lines and incorporate it into your image-based rendering algorithm.

Ex 13.2: View interpolation Extend the system you created in the previous exercise to render two reference views and then blend the images using a combination of z-buffering, hole filing, and blending (morphing) to create the final image (Section 13.1).

1. (Optional) If the two source images have very different exposures, the hole-filled regions and the blended regions will have different exposures. Can you extend your algorithm to mitigate this?
2. (Optional) Extend your algorithm to perform three-way (trilinear) interpolation between neighboring views. You can triangulate the reference camera poses and use barycentric coordinates for the virtual camera in order to determine the blending weights.

Ex 13.3: View morphing Modify your view interpolation algorithm to perform morphs between views of a non-rigid object, such as a person changing expressions.

1. Instead of using a pure stereo algorithm, use a general flow algorithm to compute displacements, but separate them into a rigid displacement due to camera motion and a non-rigid deformation.
2. At render time, use the rigid geometry to determine the new pixel location but then add a fraction of the non-rigid displacement as well.
3. Alternatively, compute a stereo depth map but let the user specify additional correspondences or use a feature-based matching algorithm to provide them automatically.
4. (Optional) Take a single image, such as the Mona Lisa or a friend's picture, and create an animated 3D view morph (Seitz and Dyer 1996).
   (a) Find the vertical axis of symmetry in the image and reflect your reference image to provide a virtual pair (assuming the person's hairstyle is somewhat symmetric).
   (b) Use structure from motion to determine the relative camera pose of the pair.

(c) Use dense stereo matching to estimate the 3D shape.

(d) Use view morphing to create a 3D animation.

Ex 13.4: View dependent texture mapping Use a 3D model you created along with the original images to implement a view-dependent texture mapping system.

1. Use one of the 3D reconstruction techniques you developed in Exercises 7.3, 11.9, 11.10, or 12.8 to build a triangulated 3D image-based model from multiple photographs.
2. Extract textures for each model face from your photographs, either by performing the appropriate resampling or by figuring out how to use the texture mapping software to directly access the source images.
3. At run time, for each new camera view, select the best source image for each visible model face.
4. Extend this to blend between the top two or three textures. This is trickier, since it involves the use of texture blending or pixel shading (Debevec, Taylor, and Malik 1996; Debevec, Yu, and Borshukov 1998; Pighin, Hecker, Lischinski *et al.* 1998).

Ex 13.5: Layered depth images Extend your view interpolation algorithm (Exercise 13.2) to store more than one depth or color value per pixel (Shade, Gortler, He *et al.* 1998), i.e., a layered depth image (LDI). Modify your rendering algorithm accordingly. For your data, you can use synthetic ray tracing, a layered reconstructed model, or a volumetric reconstruction.

Ex 13.6: Rendering from sprites or layers Extend your view interpolation algorithm to handle multiple planes or sprites (Section 13.2.1) (Shade, Gortler, He *et al.* 1998).

1. Extract your layers using the technique you developed in Exercise 8.9.
2. Alternatively, use an interactive painting and 3D placement system to extract your layers (Kang 1998; Oh, Chen, Dorsey *et al.* 2001; Shum, Sun, Yamazaki *et al.* 2004).
3. Determine a back-to-front order based on expected visibility or add a z-buffer to your rendering algorithm to handle occlusions.
4. Render and composite all of the resulting layers, with optional alpha matting to handle the edges of layers and sprites.

Ex 13.7: Light field transformations Derive the equations relating regular images to 4D light field coordinates.

1. Determine the mapping between the far plane (u, v) coordinates and a virtual camera's (x, y) coordinates.
   (a) Start by parameterizing a 3D point on the uv plane in terms of its (u, v) coordinates.
   (b) Project the resulting 3D point to the camera pixels $(x, y, 1)$ using the usual 3×4 camera matrix $\boldsymbol{P}$ (2.63).
   (c) Derive the 2D homography relating (u, v) and (x, y) coordinates.

2. Write down a similar transformation for (s, t) to (x, y) coordinates.
3. Prove that if the virtual camera is actually on the (s, t) plane, the (s, t) value depends only on the camera's optical center and is independent of (x, y).
4. Prove that an image taken by a regular orthographic or perspective camera, i.e., one that has a linear projective relationship between 3D points and (x, y) pixels (2.63), samples the (s, t, u, v) light field along a two-dimensional hyperplane.

Ex 13.8: Light field and Lumigraph rendering Implement a light field or Lumigraph rendering system:

1. Download one of the light field data sets from http://lightfield.stanford.edu/.
2. Write an algorithm to synthesize a new view from this light field, using quadri-linear interpolation of (s, t, u, v) ray samples.
3. Try varying the focal plane corresponding to your desired view (Isaksen, McMillan, and Gortler 2000) and see if the resulting image looks sharper.
4. Determine a 3D proxy for the objects in your scene. You can do this by running multi-view stereo over one of your light fields to obtain a depth map per image.
5. Implement the Lumigraph rendering algorithm, which modifies the sampling of rays according to the 3D location of each surface element.
6. Collect a set of images yourself and determine their pose using structure from motion.
7. Implement the unstructured Lumigraph rendering algorithm from Buehler, Bosse, McMillan *et al.* (2001).

Ex 13.9: Surface light fields Construct a surface light field (Wood, Azuma, Aldinger *et al.* 2000) and see how well you can compress it.

1. Acquire an interesting light field of a specular scene or object, or download one from http://lightfield.stanford.edu/.
2. Build a 3D model of the object using a multi-view stereo algorithm that is robust to outliers due to specularities.
3. Estimate the Lumisphere for each surface point on the object.
4. Estimate its diffuse components. Is the median the best way to do this? Why not use the minimum color value? What happens if there is Lambertian shading on the diffuse component?
5. Model and compress the remaining portion of the Lumisphere using one of the techniques suggested by Wood, Azuma, Aldinger *et al.* (2000) or invent one of your own.
6. Study how well your compression algorithm works and what artifacts it produces.
7. (Optional) Develop a system to edit and manipulate your surface light field.

Ex 13.10: Handheld concentric mosaics Develop a system to navigate a handheld concentric mosaic.

1. Stand in the middle of a room with a camcorder held at arm's length in front of you and spin in a circle.
2. Use a structure from motion system to determine the camera pose and sparse 3D structure for each input frame.
3. (Optional) Re-bin your image pixels into a more regular concentric mosaic structure.
4. At view time, determine from the new camera's view (which should be near the plane of your original capture) which source pixels to display. You can simplify your computations to determine a source column (and scaling) for each output column.
5. (Optional) Use your sparse 3D structure, interpolated to a dense depth map, to improve your rendering (Zheng, Kang, Cohen *et al.* 2007).

Ex 13.11: Video textures Capture some videos of natural phenomena, such as a water fountain, fire, or smiling face, and loop the video seamlessly into an infinite length video (Schödl, Szeliski, Salesin *et al.* 2000).

1. Compare all the frames in the original clip using an L_2 (sum of square difference) metric. (This assumes the videos were shot on a tripod or have already been stabilized.)
2. Filter the comparison table temporally to accentuate temporal sub-sequences that match well together.
3. Convert your similarity table into a jump probability table through some exponential distribution. Be sure to modify transitions near the end so you do not get "stuck" in the last frame.
4. Starting with the first frame, use your transition table to decide whether to jump forward, backward, or continue to the next frame.
5. (Optional) Add any of the other extensions to the original video textures idea, such as multiple moving regions, interactive control, or graph cut spatio-temporal texture seaming.

Chapter 14

Recognition

14.1 Object detection 578
14.1.1 Face detection 578
14.1.2 Pedestrian detection 585
14.2 Face recognition 588
14.2.1 Eigenfaces 589
14.2.2 Active appearance and 3D shape models 596
14.2.3 *Application*: Personal photo collections 601
14.3 Instance recognition 602
14.3.1 Geometric alignment 603
14.3.2 Large databases 604
14.3.3 *Application*: Location recognition 609
14.4 Category recognition 611
14.4.1 Bag of words 612
14.4.2 Part-based models 615
14.4.3 Recognition with segmentation 620
14.4.4 *Application*: Intelligent photo editing 621
14.5 Context and scene understanding 625
14.5.1 Learning and large image collections 627
14.5.2 *Application*: Image search 630
14.6 Recognition databases and test sets 631
14.7 Additional reading 631
14.8 Exercises 637

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_14, © Springer-Verlag London Limited 2011

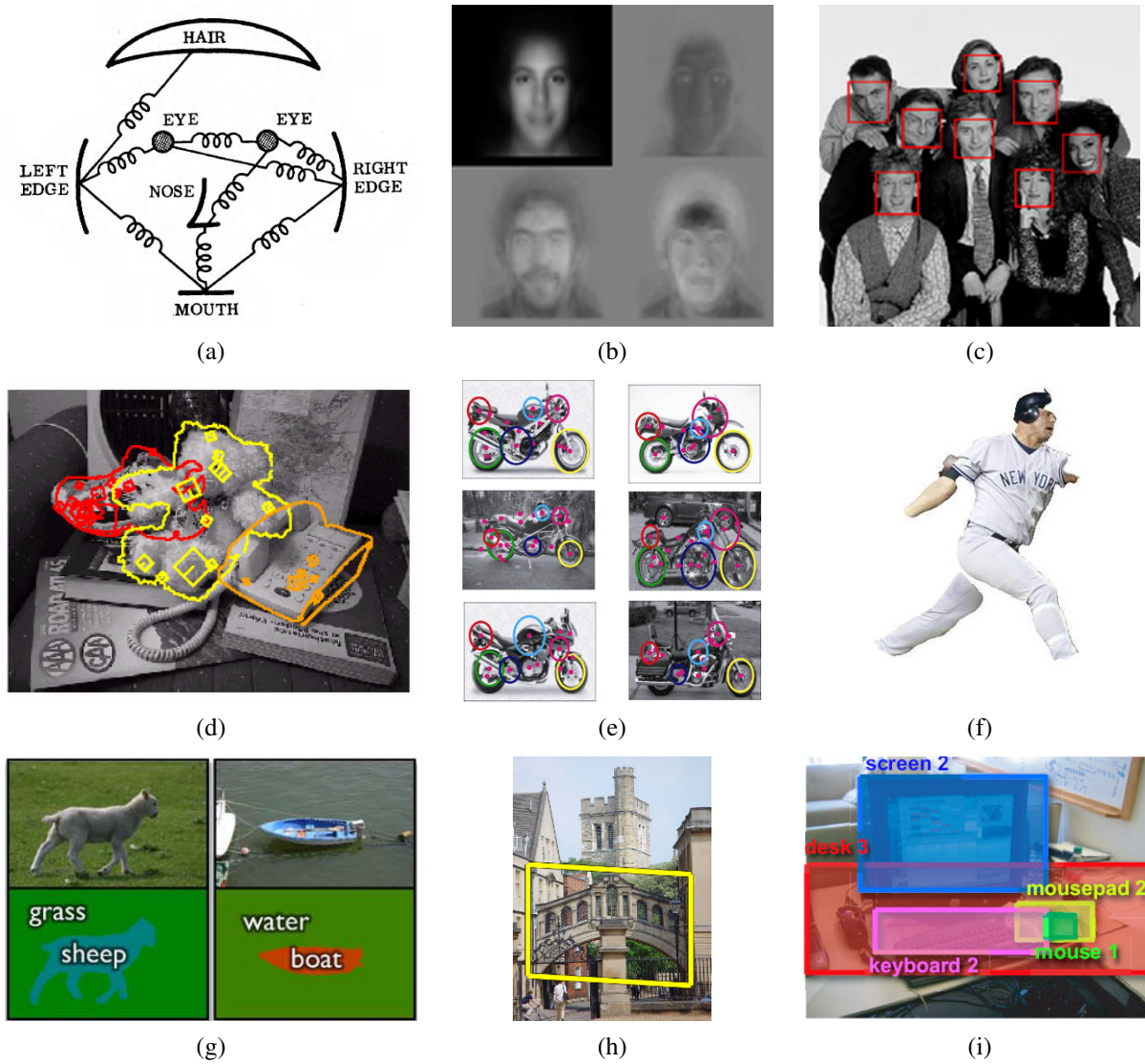


Figure 14.1 Recognition: face recognition with (a) pictorial structures (Fischler and Elschlager 1973) © 1973 IEEE and (b) eigenfaces (Turk and Pentland 1991b); (c) real-time face detection (Viola and Jones 2004) © 2004 Springer; (d) instance (known object) recognition (Lowe 1999) © 1999 IEEE; (e) feature-based recognition (Fergus, Perona, and Zisserman 2007); (f) region-based recognition (Mori, Ren, Efros *et al.* 2004) © 2004 IEEE; (g) simultaneous recognition and segmentation (Shotton, Winn, Rother *et al.* 2009) © 2009 Springer; (h) location recognition (Philbin, Chum, Isard *et al.* 2007) © 2007 IEEE; (i) using context (Russell, Torralba, Liu *et al.* 2007).

Of all the visual tasks we might ask a computer to perform, analyzing a scene and recognizing all of the constituent objects remains the most challenging. While computers excel at accurately reconstructing the 3D shape of a scene from images taken from different views, they cannot name all the objects and animals present in a picture, even at the level of a two-year-old child. There is not even any consensus among researchers on when this level of performance might be achieved.

Why is recognition so hard? The real world is made of a jumble of objects, which all occlude one another and appear in different poses. Furthermore, the variability intrinsic within a class (e.g., dogs), due to complex non-rigid articulation and extreme variations in shape and appearance (e.g., between different breeds), makes it unlikely that we can simply perform exhaustive matching against a database of exemplars.[1]

The recognition problem can be broken down along several axes. For example, if we know what we are looking for, the problem is one of *object detection* (Section 14.1), which involves quickly scanning an image to determine where a match may occur (Figure 14.1c). If we have a specific rigid object we are trying to recognize (*instance recognition*, Section 14.3), we can search for characteristic feature points (Section 4.1) and verify that they align in a geometrically plausible way (Section 14.3.1) (Figure 14.1d).

The most challenging version of recognition is general *category* (or *class*) recognition (Section 14.4), which may involve recognizing instances of extremely varied classes such as animals or furniture. Some techniques rely purely on the presence of features (known as a "bag of words" model—see Section 14.4.1), their relative positions (*part-based models* (Section 14.4.2)), Figure 14.1e, while others involve segmenting the image into semantically meaningful regions (Section 14.4.3) (Figure 14.1f). In many instances, recognition depends heavily on the *context* of surrounding objects and scene elements (Section 14.5). Woven into all of these techniques is the topic of *learning* (Section 14.5.1), since hand-crafting specific object recognizers seems like a futile approach given the complexity of the problem.

Given the extremely rich and complex nature of this topic, this chapter is structured to build from simpler concepts to more complex ones. We begin with a discussion of face and object detection (Section 14.1), where we introduce a number of machine-learning techniques such as boosting, neural networks, and support vector machines. Next, we study face recognition (Section 14.2), which is one of the more widely known applications of recognition. This topic serves as an introduction to subspace (PCA) models and Bayesian approaches to recognition and classification. We then present techniques for instance recognition (Section 14.3), building upon earlier topics in this book, such as feature detection, matching, and geometric alignment (Section 14.3.1). We introduce topics from the information and document retrieval communities, such as frequency vectors, feature quantization, and inverted indices (Section 14.3.2). We also present applications of location recognition (Section 14.3.3).

In the second half of the chapter, we address the most challenging variant of recognition, namely the problem of category recognition (Section 14.4). This includes approaches that use bags of features (Section 14.4.1), parts (Section 14.4.2), and segmentation (Section 14.4.3). We show how such techniques can be used to automate photo editing tasks, such as 3D modeling, scene completion, and creating collages (Section 14.4.4). Next, we discuss the role that context can play in both individual object recognition and more holistic scene under-

[1] However, some recent research suggests that direct image matching may be feasible for large enough databases (Russell, Torralba, Liu *et al.* 2007; Malisiewicz and Efros 2008; Torralba, Freeman, and Fergus 2008).

standing (Section 14.5). We close this chapter with a discussion of databases and test sets for constructing and evaluating recognition systems (Section 14.6).

While there is no comprehensive reference on object recognition, an excellent set of notes can be found in the ICCV 2009 short course (Fei-Fei, Fergus, and Torralba 2009), Antonio Torralba's more comprehensive MIT course (Torralba 2008), and two recent collections of papers (Ponce, Hebert, Schmid *et al.* 2006; Dickinson, Leonardis, Schiele *et al.* 2007) and a survey on object categorization (Pinz 2005). An evaluation of some of the best performing recognition algorithms can be found on the PASCAL Visual Object Classes (VOC) Challenge Web site at http://pascallin.ecs.soton.ac.uk/challenges/VOC/.

14.1 Object detection

If we are given an image to analyze, such as the group portrait in Figure 14.2, we could try to apply a recognition algorithm to every possible sub-window in this image. Such algorithms are likely to be both slow and error-prone. Instead, it is more effective to construct special-purpose *detectors*, whose job it is to rapidly find likely regions where particular objects might occur.

We begin this section with face detectors, which are some of the more successful examples of recognition. For example, such algorithms are built into most of today's digital cameras to enhance auto-focus and into video conferencing systems to control pan-tilt heads. We then look at pedestrian detectors, as an example of more general methods for object detection. Such detectors can be used in automotive safety applications, e.g., detecting pedestrians and other cars from moving vehicles (Leibe, Cornelis, Cornelis *et al.* 2007).

14.1.1 Face detection

Before face recognition can be applied to a general image, the locations and sizes of any faces must first be found (Figures 14.1c and 14.2). In principle, we could apply a face recognition algorithm at every pixel and scale (Moghaddam and Pentland 1997) but such a process would be too slow in practice.

Over the years, a wide variety of fast face detection algorithms have been developed. Yang, Kriegman, and Ahuja (2002) provide a comprehensive survey of earlier work in this field; Yang's ICPR 2004 tutorial[2] and the Torralba (2007) short course provide more recent reviews.[3]

According to the taxonomy of Yang, Kriegman, and Ahuja (2002), face detection techniques can be classified as feature-based, template-based, or appearance-based. Feature-based techniques attempt to find the locations of distinctive image features such as the eyes, nose, and mouth, and then verify whether these features are in a plausible geometrical arrangement. These techniques include some of the early approaches to face recognition (Fischler and Elschlager 1973; Kanade 1977; Yuille 1991), as well as more recent approaches based on modular eigenspaces (Moghaddam and Pentland 1997), local filter jets (Leung, Burl, and Perona 1995; Penev and Atick 1996; Wiskott, Fellous, Krüger *et al.* 1997), support

[2] http://vision.ai.uiuc.edu/mhyang/face-detection-survey.html.

[3] An alternative approach to detecting faces is to look for regions of skin color in the image (Forsyth and Fleck 1999; Jones and Rehg 2001). See Exercise 2.8 for some additional discussion and references.



Figure 14.2 Face detection results produced by Rowley, Baluja, and Kanade (1998a) © 1998 IEEE. Can you find the one false positive (a box around a non-face) among the 57 true positive results?

vector machines (Heisele, Ho, Wu *et al.* 2003; Heisele, Serre, and Poggio 2007), and boosting (Schneiderman and Kanade 2004).

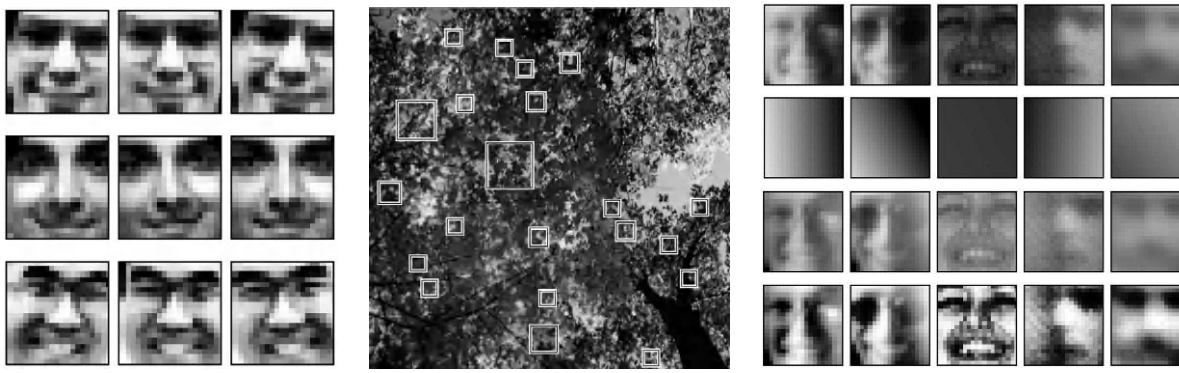
Template-based approaches, such as active appearance models (AAMs) (Section 14.2.2), can deal with a wide range of pose and expression variability. Typically, they require good initialization near a real face and are therefore not suitable as fast face detectors.

Appearance-based approaches scan over small overlapping rectangular patches of the image searching for likely face candidates, which can then be refined using a *cascade* of more expensive but selective detection algorithms (Sung and Poggio 1998; Rowley, Baluja, and Kanade 1998a; Romdhani, Torr, Schölkopf *et al.* 2001; Fleuret and Geman 2001; Viola and Jones 2004). In order to deal with scale variation, the image is usually converted into a sub-octave pyramid and a separate scan is performed on each level. Most appearance-based approaches today rely heavily on training classifiers using sets of labeled face and non-face patches.

Sung and Poggio (1998) and Rowley, Baluja, and Kanade (1998a) present two of the earliest appearance-based face detectors and introduce a number of innovations that are widely used in later work by others.

To start with, both systems collect a set of labeled face patches (Figure 14.2) as well as a set of patches taken from images that are known not to contain faces, such as aerial images or vegetation (Figure 14.3b). The collected face images are augmented by artificially mirroring, rotating, scaling, and translating the images by small amounts to make the face detectors less sensitive to such effects (Figure 14.3a).

After an initial set of training images has been collected, some optional pre-processing can be performed, such as subtracting an average gradient (linear function) from the image to compensate for global shading effects and using histogram equalization to compensate for



(a) (b) (c)

Figure 14.3 Pre-processing stages for face detector training (Rowley, Baluja, and Kanade 1998a) © 1998 IEEE: (a) artificially mirroring, rotating, scaling, and translating training images for greater variability; (b) using images without faces (looking up at a tree) to generate non-face examples; (c) pre-processing the patches by subtracting a best fit linear function (constant gradient) and histogram equalizing.

varying camera contrast (Figure 14.3c).

Clustering and PCA. Once the face and non-face patterns have been pre-processed, Sung and Poggio (1998) cluster each of these datasets into six separate clusters using k-means and then fit PCA subspaces to each of the resulting 12 clusters (Figure 14.4). At detection time, the DIFS and DFFS metrics first developed by Moghaddam and Pentland (1997) (see Figure 14.14 and (14.14)) are used to produce 24 Mahalanobis distance measurements (two per cluster). The resulting 24 measurements are input to a multi-layer perceptron (MLP), which is a neural network with alternating layers of weighted summations and sigmoidal non-linearities trained using the "backpropagation" algorithm (Rumelhart, Hinton, and Williams 1986).

Neural networks. Instead of first clustering the data and computing Mahalanobis distances to the cluster centers, Rowley, Baluja, and Kanade (1998a) apply a neural network (MLP) directly to the 20×20 pixel patches of gray-level intensities, using a variety of differently sized hand-crafted "receptive fields" to capture both large-scale and smaller scale structure (Figure 14.5). The resulting neural network directly outputs the likelihood of a face at the center of every overlapping patch in a multi-resolution pyramid. Since several overlapping patches (in both space and resolution) may fire near a face, an additional merging network is used to merge overlapping detections. The authors also experiment with training several networks and merging their outputs. Figure 14.2 shows a sample result from their face detector.

To make the detector run faster, a separate network operating on 30×30 patches is trained to detect both faces and faces shifted by ± 5 pixels. This network is evaluated at every 10th pixel in the image (horizontally and vertically) and the results of this "coarse" or "sloppy" detector are used to select regions on which to run the slower single-pixel overlap technique. To deal with in-plane rotations of faces, Rowley, Baluja, and Kanade (1998b) train a *router*

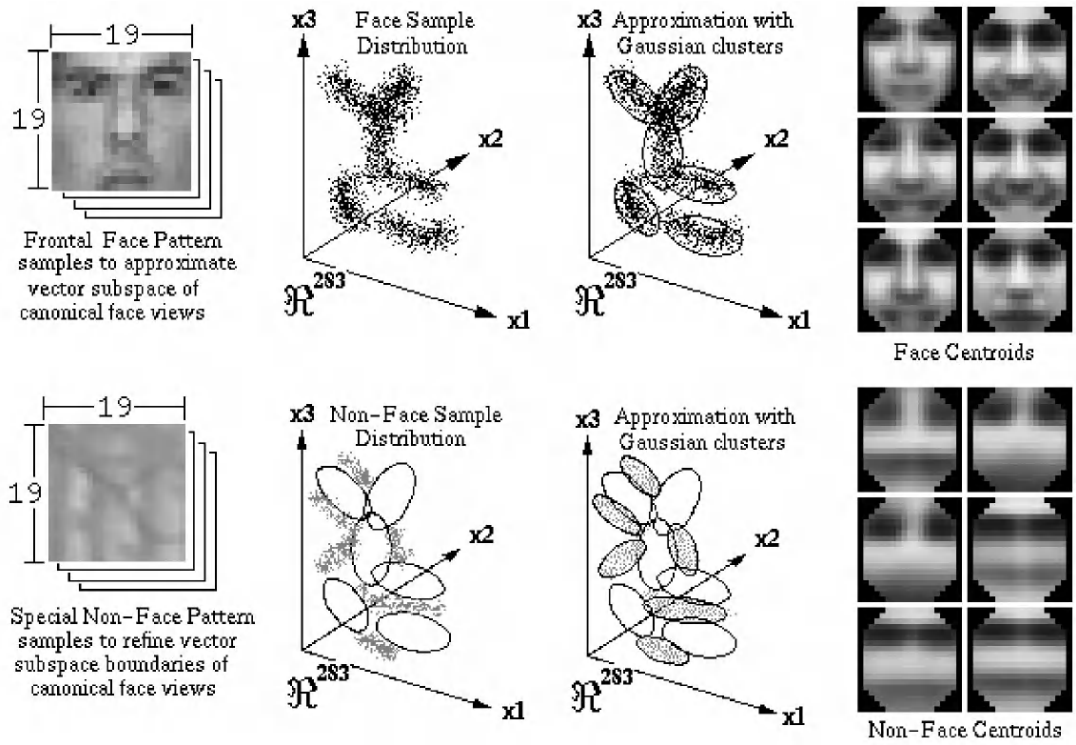


Figure 14.4 Learning a mixture of Gaussians model for face detection (Sung and Poggio 1998) © 1998 IEEE. The face and non-face images (19^2-long vectors) are first clustered into six separate clusters (each) using k-means and then analyzed using PCA. The cluster centers are shown in the right-hand columns.

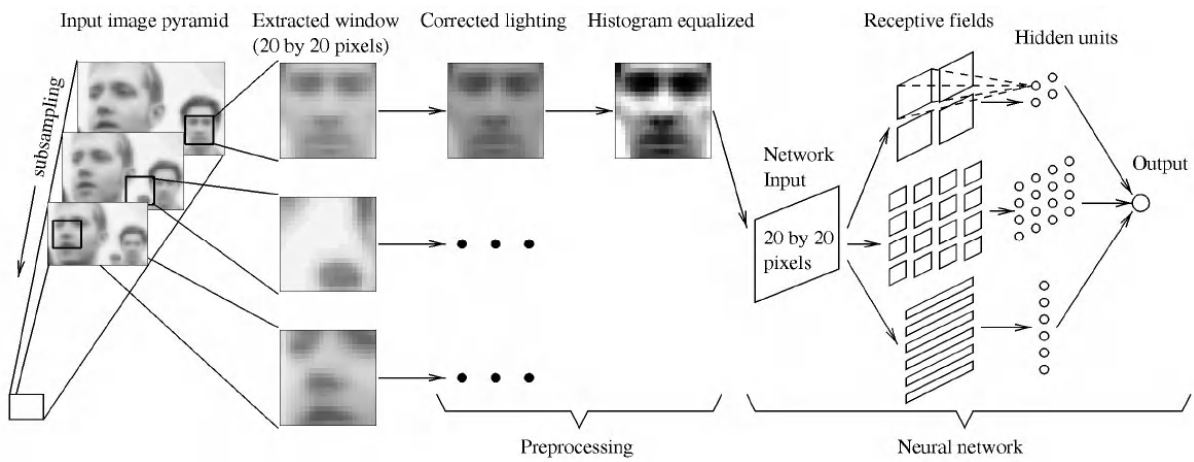
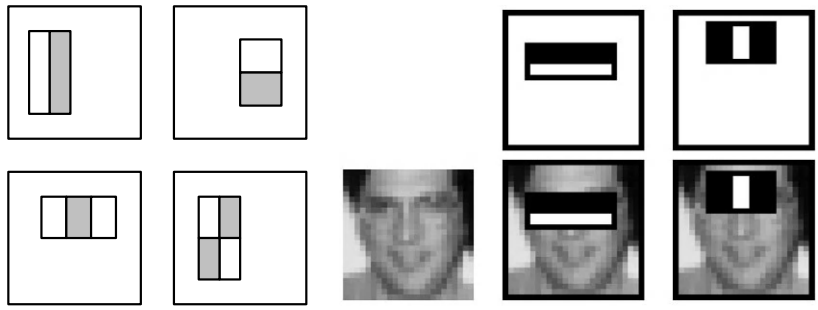


Figure 14.5 A neural network for face detection (Rowley, Baluja, and Kanade 1998a) © 1998 IEEE. Overlapping patches are extracted from different levels of a pyramid and then pre-processed as shown in Figure 14.3b. A three-layer neural network is then used to detect likely face locations.



(a) (b)

Figure 14.6 Simple features used in boosting-based face detector (Viola and Jones 2004) © 2004 Springer: (a) difference of rectangle feature composed of 2–4 different rectangles (pixels inside the white rectangles are subtracted from the gray ones); (b) the first and second features selected by AdaBoost. The first feature measures the differences in intensity between the eyes and the cheeks, the second one between the eyes and the bridge of the nose.

network to estimate likely rotation angles from input patches and then apply the estimated rotation to each patch before running the result through their upright face detector.

Support vector machines. Instead of using a neural network to classify patches, Osuna, Freund, and Girosi (1997) use a *support vector machine* (SVM) (Hastie, Tibshirani, and Friedman 2001; Schölkopf and Smola 2002; Bishop 2006; Lampert 2008) to classify the same preprocessed patches as Sung and Poggio (1998). An SVM searches for a series of *maximum margin* separating planes in feature space between different classes (in this case, face and non-face patches). In those cases where linear classification boundaries are insufficient, the feature space can be lifted into higher-dimensional features using *kernels* (Hastie, Tibshirani, and Friedman 2001; Schölkopf and Smola 2002; Bishop 2006). SVMs have been used by other researchers for both face detection and face recognition (Heisele, Ho, Wu *et al.* 2003; Heisele, Serre, and Poggio 2007) and are a widely used tool in object recognition in general.

Boosting. Of all the face detectors currently in use, the one introduced by Viola and Jones (2004) is probably the best known and most widely used. Their technique was the first to introduce the concept of *boosting* to the computer vision community, which involves training a series of increasingly discriminating simple classifiers and then blending their outputs (Hastie, Tibshirani, and Friedman 2001; Bishop 2006).

In more detail, boosting involves constructing a *classifier* $h(\boldsymbol{x})$ as a sum of simple *weak learners*,

$$h(\boldsymbol{x}) = \text{sign}\left[\sum_{j=0}^{m-1} \alpha_j h_j(\boldsymbol{x})\right], \tag{14.1}$$

where each of the weak learners $h_j(\boldsymbol{x})$ is an extremely simple function of the input, and hence is not expected to contribute much (in isolation) to the classification performance.

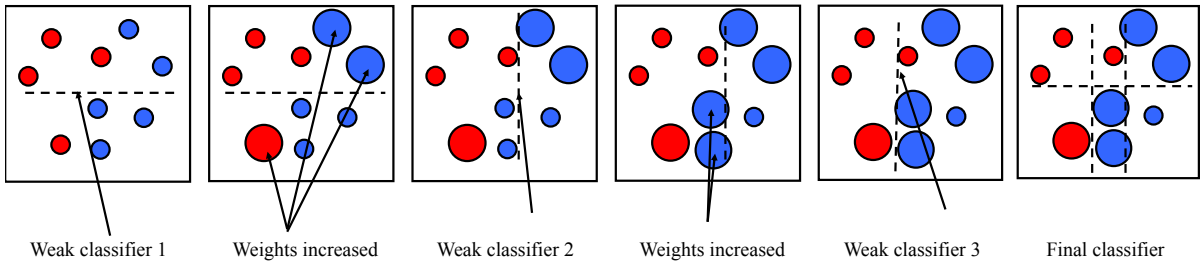


Figure 14.7 Schematic illustration of boosting, courtesy of Svetlana Lazebnik, after original illustrations from Paul Viola and David Lowe. After each weak classifier (decision stump or hyperplane) is selected, data points that are erroneously classified have their weights increased. The final classifier is a linear combination of the simple weak classifiers.

In most variants of boosting, the weak learners are threshold functions,

$$h_j(\boldsymbol{x}) = a_j[f_j < \theta_j] + b_j[f_j \geq \theta_j] = \begin{cases} a_j & \text{if } f_j < \theta_j \\ b_j & \text{otherwise,} \end{cases} \tag{14.2}$$

which are also known as *decision stumps* (basically, the simplest possible version of *decision trees*). In most cases, it is also traditional (and simpler) to set a_j and b_j to ± 1, i.e., $a_j = -s_j$, $b_j = +s_j$, so that only the feature f_j, the threshold value θ_j, and the polarity of the threshold $s_j \in \pm 1$ need to be selected.[4]

In many applications of boosting, the features are simply coordinate axes x_k, i.e., the boosting algorithm selects one of the input vector components as the best one to threshold. In Viola and Jones' face detector, the features are differences of rectangular regions in the input patch, as shown in Figure 14.6. The advantage of using these features is that, while they are more discriminating than single pixels, they are extremely fast to compute once a summed area table has been pre-computed, as described in Section 3.2.3 (3.31–3.32). Essentially, for the cost of an $O(N)$ pre-computation phase (where N is the number of pixels in the image), subsequent differences of rectangles can be computed in $4r$ additions or subtractions, where $r \in \{2, 3, 4\}$ is the number of rectangles in the feature.

The key to the success of boosting is the method for incrementally selecting the weak learners and for re-weighting the training examples after each stage (Figure 14.7). The AdaBoost (Adaptive Boosting) algorithm (Hastie, Tibshirani, and Friedman 2001; Bishop 2006) does this by re-weighting each sample as a function of whether it is correctly classified at each stage, and using the stage-wise average classification error to determine the final weightings α_j among the weak classifiers, as described in Algorithm 14.1. While the resulting classifier is extremely fast in practice, the training time can be quite slow (in the order of weeks), because of the large number of feature (difference of rectangle) hypotheses that need to be examined at each stage.

To further increase the speed of the detector, it is possible to create a *cascade* of classifiers, where each classifier uses a small number of tests (say, a two-term AdaBoost classifier) to reject a large fraction of non-faces while trying to pass through all potential face candidates

[4] Some variants, such as that of Viola and Jones (2004), use $(a_j, b_j) \in [0, 1]$ and adjust the learning algorithm accordingly.

1. Input the positive and negative training examples along with their labels $\{(\boldsymbol{x}_i, y_i)\}$, where $y_i = 1$ for positive (face) examples and $y_i = -1$ for negative examples.
2. Initialize all the weights to $w_{i,1} \leftarrow \frac{1}{N}$, where N is the number of training examples. (Viola and Jones (2004) use a separate N_1 and N_2 for positive and negative examples.)
3. For each training stage $j = 1 \ldots M$:
   (a) Renormalize the weights so that they sum up to 1 (divide them by their sum).
   (b) Select the best classifier $h_j(\boldsymbol{x}; f_j, \theta_j, s_j)$ by finding the one that minimizes the weighted classification error

$$e_j = \sum_{i=0}^{N-1} w_{i,j} e_{i,j}, \tag{14.3}$$

$$e_{i,j} = 1 - \delta(y_i, h_j(\boldsymbol{x}_i; f_j, \theta_j, s_j)). \tag{14.4}$$

   For any given f_j function, the optimal values of (θ_j, s_j) can be found in linear time using a variant of weighted median computation (Exercise 14.2).
   (c) Compute the modified error rate β_j and classifier weight α_j,

$$\beta_j = \frac{e_j}{1 - e_j} \quad \text{and} \quad \alpha_j = -\log \beta_j. \tag{14.5}$$

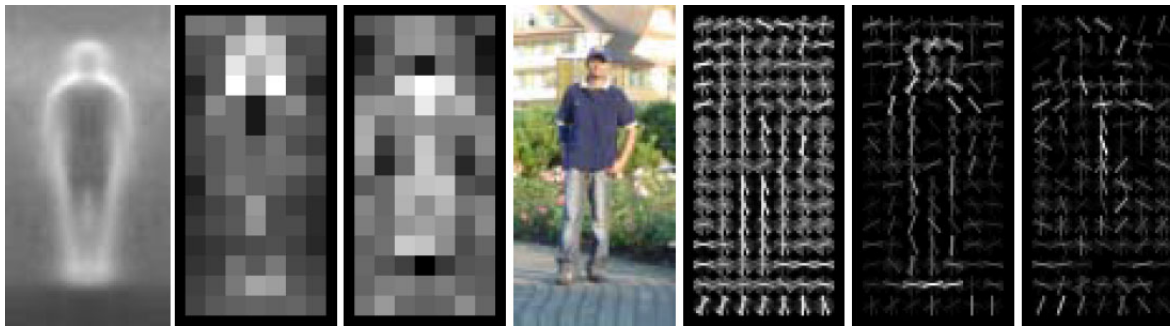
   (d) Update the weights according to the classification errors $e_{i,j}$

$$w_{i,j+1} \leftarrow w_{i,j} \beta_j^{1-e_{i,j}}, \tag{14.6}$$

   i.e., downweight the training samples that were correctly classified in proportion to the overall classification error.
4. Set the final classifier to

$$h(\boldsymbol{x}) = \text{sign}\left[\sum_{j=0}^{m-1} \alpha_j h_j(\boldsymbol{x})\right]. \tag{14.7}$$

Algorithm 14.1 The AdaBoost training algorithm, adapted from Hastie, Tibshirani, and Friedman (2001), Viola and Jones (2004), and Bishop (2006).



(a) (b) (c) (d) (e) (f) (g)

Figure 14.8 Pedestrian detection using histograms of oriented gradients (Dalal and Triggs 2005) © 2005 IEEE: (a) the average gradient image over the training examples; (b) each "pixel" shows the maximum positive SVM weight in the block centered on the pixel; (c) likewise, for the negative SVM weights; (d) a test image; (e) the computed R-HOG (rectangular histogram of gradients) descriptor; (f) the R-HOG descriptor weighted by the positive SVM weights; (g) the R-HOG descriptor weighted by the negative SVM weights.

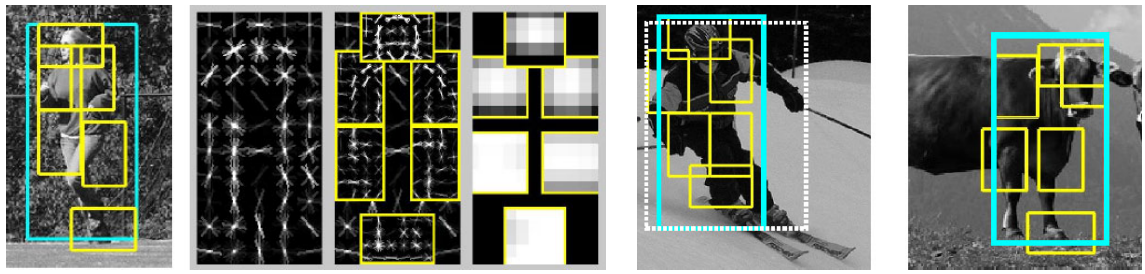
(Fleuret and Geman 2001; Viola and Jones 2004). An even faster algorithm for performing cascade learning has recently been developed by Brubaker, Wu, Sun *et al.* (2008).

14.1.2 Pedestrian detection

While a lot of the research on object detection has focused on faces, the detection of other objects, such as pedestrians and cars, has also received widespread attention (Gavrila and Philomin 1999; Gavrila 1999; Papageorgiou and Poggio 2000; Mohan, Papageorgiou, and Poggio 2001; Schneiderman and Kanade 2004). Some of these techniques maintain the same focus as face detection on speed and efficiency. Others, however, focus instead on accuracy, viewing detection as a more challenging variant of generic class recognition (Section 14.4) in which the locations and extents of objects are to be determined as accurately as possible. (See, for example, the PASCAL VOC detection challenge, http://pascallin.ecs.soton.ac.uk/challenges/VOC/.)

An example of a well-known pedestrian detector is the algorithm developed by Dalal and Triggs (2005), who use a set of overlapping *histogram of oriented gradients* (HOG) descriptors fed into a support vector machine (Figure 14.8). Each HOG has cells to accumulate magnitude-weighted votes for gradients at particular orientations, just as in the scale invariant feature transform (SIFT) developed by Lowe (2004), which we discussed in Section 4.1.2 and Figure 4.18. Unlike SIFT, however, which is only evaluated at interest point locations, HOGs are evaluated on a regular overlapping grid and their descriptor magnitudes are normalized using an even coarser grid; they are only computed at a single scale and a fixed orientation. In order to capture the subtle variations in orientation around a person's outline, a large number of orientation bins is used and no smoothing is performed in the central difference gradient computation—see the work of Dalal and Triggs (2005) for more implementation details. Figure 14.8d shows a sample input image, while Figure 14.8e shows the associated HOG descriptors.

Once the descriptors have been computed, a support vector machine (SVM) is trained



(a) (b) (c) (d)

Figure 14.9 Part-based object detection (Felzenszwalb, McAllester, and Ramanan 2008) © 2008 IEEE: (a) An input photograph and its associated person (blue) and part (yellow) detection results. (b) The detection model is defined by a coarse template, several higher resolution part templates, and a spatial model for the location of each part. (c) True positive detection of a skier and (d) false positive detection of a cow (labeled as a person).

on the resulting high-dimensional continuous descriptor vectors. Figures 14.8b–c show a diagram of the (most) positive and negative SVM weights in each block, while Figures 14.8f–g show the corresponding weighted HOG responses for the central input image. As you can see, there are a fair number of positive responses around the head, torso, and feet of the person, and relatively few negative responses (mainly around the middle and the neck of the sweater).

The fields of pedestrian and general object detection have continued to evolve rapidly over the last decade (Belongie, Malik, and Puzicha 2002; Mikolajczyk, Schmid, and Zisserman 2004; Leibe, Seemann, and Schiele 2005; Opelt, Pinz, and Zisserman 2006; Torralba 2007; Andriluka, Roth, and Schiele 2009, 2010; Dollàr, Belongie, and Perona 2010). Munder and Gavrila (2006) compare a number of pedestrian detectors and conclude that those based on local receptive fields and SVMs perform the best, with a boosting-based approach coming close. Maji, Berg, and Malik (2008) improve on the best of these results using non-overlapping multi-resolution HOG descriptors and a histogram intersection kernel SVM based on a spatial pyramid match kernel from Lazebnik, Schmid, and Ponce (2006).

When detectors for several different classes are being constructed simultaneously, Torralba, Murphy, and Freeman (2007) show that sharing features and weak learners between detectors yields better performance, both in terms of faster computation times and fewer training examples. To find the features and decision stumps that work best in a shared manner, they introduce a novel *joint boosting* algorithm that optimizes, at each stage, a summed expected exponential loss function using the "gentleboost" algorithm of Friedman, Hastie, and Tibshirani (2000).

In more recent work, Felzenszwalb, McAllester, and Ramanan (2008) extend the histogram of oriented gradients person detector to incorporate flexible parts models (Section 14.4.2). Each part is trained and detected on HOGs evaluated at two pyramid levels below the overall object model and the locations of the parts relative to the parent node (the overall bounding box) are also learned and used during recognition (Figure 14.9b). To compensate for inaccuracies or inconsistencies in the training example bounding boxes (dashed white lines in Figure 14.9c), the "true" location of the parent (blue) bounding box is considered a latent (hidden) variable and is inferred during both training and recognition. Since the locations

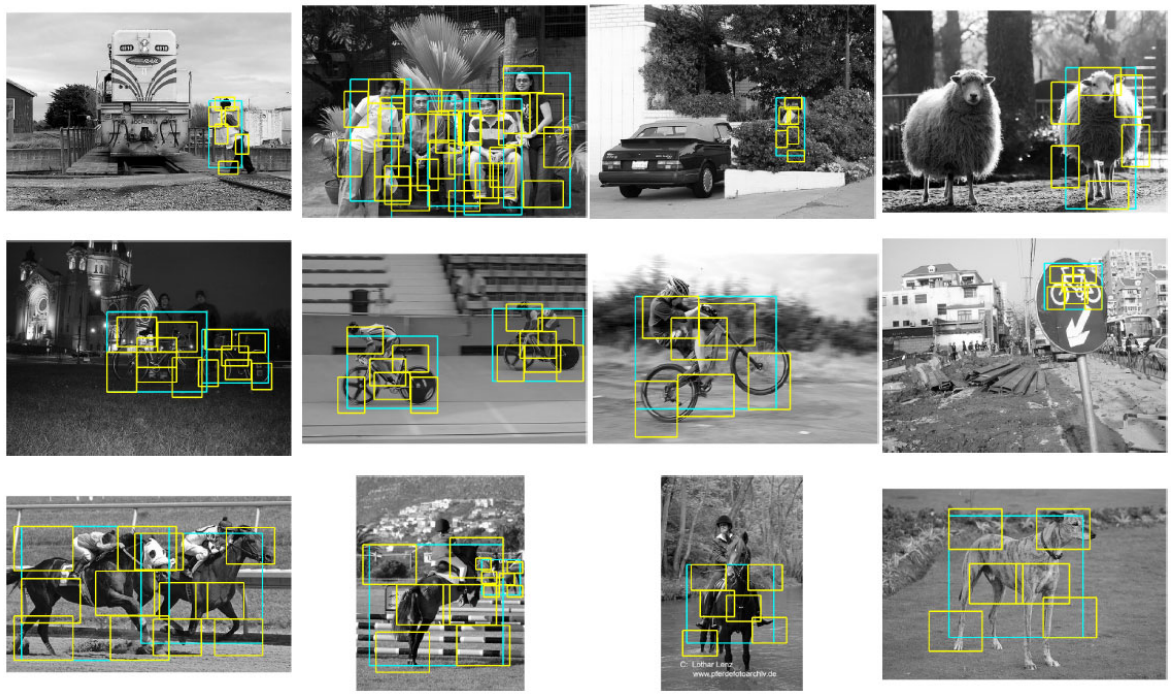


Figure 14.10 Part-based object detection results for people, bicycles, and horses (Felzenszwalb, McAllester, and Ramanan 2008) © 2008 IEEE. The first three columns show correct detections, while the rightmost column shows false positives.

of the parts are also latent, the system can be trained in a semi-supervised fashion, without needing part labels in the training data. An extension to this system (Felzenszwalb, Girshick, McAllester *et al.* 2010), which includes among its improvements a simple contextual model, was among the two best object detection systems in the 2008 Visual Object Classes detection challenge. Other recent improvements to part-based person detection and pose estimation include the work by Andriluka, Roth, and Schiele (2009) and Kumar, Zisserman, and H.S.Torr (2009).

An even more accurate estimate of a person's pose and location is presented by Rogez, Rihan, Ramalingam *et al.* (2008), who compute both the phase of a person in a walk cycle and the locations of individual joints, using random forests built on top of HOGs (Figure 14.11). Since their system produces full 3D pose information, it is closer in its application domain to 3D person trackers (Sidenbladh, Black, and Fleet 2000; Andriluka, Roth, and Schiele 2010), which we discussed in Section 12.6.4.

One final note on person and object detection. When video sequences are available, the additional information present in the optic flow and motion discontinuities can greatly aid in the detection task, as discussed by Efros, Berg, Mori *et al.* (2003), Viola, Jones, and Snow (2003), and Dalal, Triggs, and Schmid (2006).

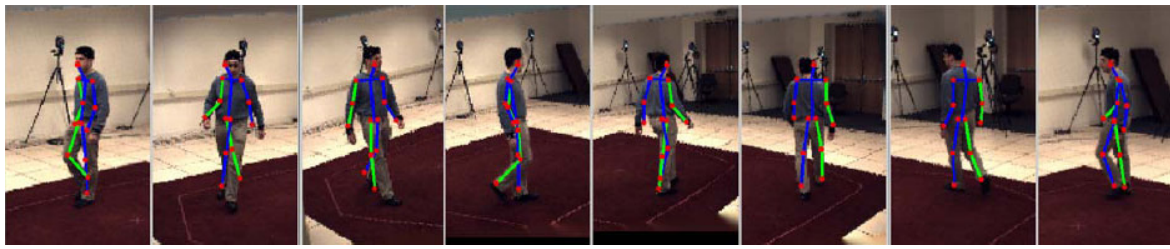


Figure 14.11 Pose detection using random forests (Rogez, Rihan, Ramalingam *et al.* 2008) © 2008 IEEE. The estimated pose (state of the kinematic model) is drawn over each input frame.



Figure 14.12 Humans can recognize low-resolution faces of familiar people (Sinha, Balas, Ostrovsky *et al.* 2006) © 2006 IEEE.

14.2 Face recognition

Among the various recognition tasks that computers might be asked to perform, face recognition is the one where they have arguably had the most success.[5] While computers cannot pick out suspects from thousands of people streaming in front of video cameras (even people cannot readily distinguish between similar people with whom they are not familiar (O'Toole, Jiang, Roark *et al.* 2006; O'Toole, Phillips, Jiang *et al.* 2009)), their ability to distinguish among a small number of family members and friends has found its way into consumer-level photo applications, such as Picasa and iPhoto. Face recognition can also be used in a variety of additional applications, including human–computer interaction (HCI), identity verification (Kirovski, Jojic, and Jancke 2004), desktop login, parental controls, and patient monitoring (Zhao, Chellappa, Phillips *et al.* 2003).

Today's face recognizers work best when they are given full frontal images of faces under relatively uniform illumination conditions, although databases that include large amounts of pose and lighting variation have been collected (Phillips, Moon, Rizvi *et al.* 2000; Sim,

[5] Instance recognition, i.e., the re-recognition of known objects such as locations or planar objects, is the other most successful application of general image recognition. In the general domain of *biometrics*, i.e., identity recognition, specialized images such as irises and fingerprints perform even better (Jain, Bolle, and Pankanti 1999; Pankanti, Bolle, and Jain 2000; Daugman 2004).



(a) (b) (c) (d)

Figure 14.13 Face modeling and compression using eigenfaces (Moghaddam and Pentland 1997) © 1997 IEEE: (a) input image; (b) the first eight eigenfaces; (c) image reconstructed by projecting onto this basis and compressing the image to 85 bytes; (d) image reconstructed using JPEG (530 bytes).

Baker, and Bsat 2003; Gross, Shi, and Cohn 2005; Huang, Ramesh, Berg *et al.* 2007; Phillips, Scruggs, O'Toole *et al.* 2010). (See Table 14.1 in Section 14.6 for more details.)

Some of the earliest approaches to face recognition involved finding the locations of distinctive image features, such as the eyes, nose, and mouth, and measuring the distances between these feature locations (Fischler and Elschlager 1973; Kanade 1977; Yuille 1991). More recent approaches rely on comparing gray-level images projected onto lower dimensional subspaces called *eigenfaces* (Section 14.2.1) and jointly modeling shape and appearance variations (while discounting pose variations) using *active appearance models* (Section 14.2.2).

Descriptions of additional face recognition techniques can be found in a number of surveys and books on this topic (Chellappa, Wilson, and Sirohey 1995; Zhao, Chellappa, Phillips *et al.* 2003; Li and Jain 2005) as well as the Face Recognition Web site.[6] The survey on face recognition by humans by Sinha, Balas, Ostrovsky *et al.* (2006) is also well worth reading; it includes a number of surprising results, such as humans' ability to recognize low-resolution images of familiar faces (Figure 14.12) and the importance of eyebrows in recognition.

14.2.1 Eigenfaces

Eigenfaces rely on the observation first made by Kirby and Sirovich (1990) that an arbitrary face image $\boldsymbol{x}$ can be compressed and reconstructed by starting with a mean image $\boldsymbol{m}$ (Figure 14.1b) and adding a small number of scaled signed images $\boldsymbol{u}_i$,[7]

$$\tilde{\boldsymbol{x}} = \boldsymbol{m} + \sum_{i=0}^{M-1} a_i \boldsymbol{u}_i, \tag{14.8}$$

where the signed basis images (Figure 14.13b) can be derived from an ensemble of training images using *principal component analysis* (also known as *eigenvalue analysis* or the *Karhunen–Loève transform*). Turk and Pentland (1991a) recognized that the coefficients a_i in the eigenface expansion could themselves be used to construct a fast image matching algorithm.

[6] http://www.face-rec.org/.

[7] In previous chapters, we used I to indicate images; in this chapter, we use the more abstract quantities $\boldsymbol{x}$ and $\boldsymbol{u}$ to indicate collections of pixels in an image turned into a vector.

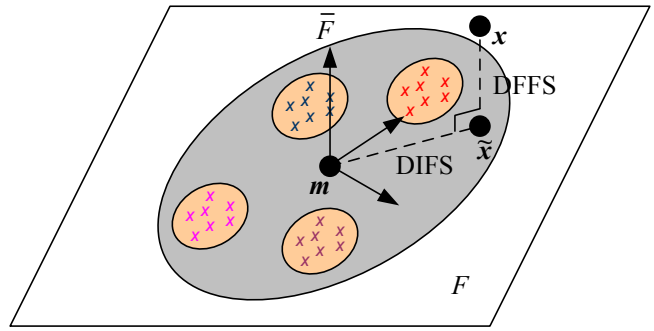


Figure 14.14 Projection onto the linear subspace spanned by the eigenface images (Moghaddam and Pentland 1997) © 1997 IEEE. The distance from face space (DFFS) is the orthogonal distance to the plane, while the distance in face space (DIFS) is the distance along the plane from the mean image. Both distances can be turned into Mahalanobis distances and given probabilistic interpretations.

In more detail, let us start with a collection of *training images* $\{\boldsymbol{x}_j\}$, from which we can compute the mean image $\boldsymbol{m}$ and a *scatter* or *covariance* matrix

$$\boldsymbol{C} = \frac{1}{N}\sum_{j=0}^{N-1}(\boldsymbol{x}_j - \boldsymbol{m})(\boldsymbol{x}_j - \boldsymbol{m})^T. \tag{14.9}$$

We can apply the eigenvalue decomposition (A.6) to represent this matrix as

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = \sum_{i=0}^{N-1}\lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T, \tag{14.10}$$

where the λ_i are the eigenvalues of $\boldsymbol{C}$ and the $\boldsymbol{u}_i$ are the *eigenvectors*. For general images, Kirby and Sirovich (1990) call these vectors *eigenpictures*; for faces, Turk and Pentland (1991a) call them *eigenfaces* (Figure 14.13b).[8]

Two important properties of the eigenvalue decomposition are that the optimal (best approximation) coefficients a_i for any new image $\boldsymbol{x}$ can be computed as

$$a_i = (\boldsymbol{x} - \boldsymbol{m}) \cdot \boldsymbol{u}_i, \tag{14.11}$$

and that, assuming the eigenvalues $\{\lambda_i\}$ are sorted in decreasing order, truncating the approximation given in (14.8) at any point M gives the best possible approximation (least error) between $\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}$. Figure 14.13c shows the resulting approximation corresponding to Figure 14.13a and shows how much better it is at compressing a face image than JPEG.

Truncating the eigenface decomposition of a face image (14.8) after M components is equivalent to projecting the image onto a linear subspace F, which we can call the *face space* (Figure 14.14). Because the eigenvectors (eigenfaces) are orthogonal and of unit norm, the

[8] In actual practice, the full $P \times P$ scatter matrix (14.9) is never computed. Instead, a smaller $N \times N$ matrix consisting of the inner products between all the signed deviations $(\boldsymbol{x}_i - \boldsymbol{m})$ is accumulated instead. See Appendix A.1.2 (A.13–A.14) for details.

distance of a projected face $\tilde{\boldsymbol{x}}$ to the mean face $\boldsymbol{m}$ can be written as

$$\text{DIFS} = \|\tilde{\boldsymbol{x}} - \boldsymbol{m}\| = \sqrt{\sum_{i=0}^{M-1} a_i^2}, \tag{14.12}$$

where DIFS stands for *distance in face space* (Moghaddam and Pentland 1997). The remaining distance between the original image $\boldsymbol{x}$ and its projection onto face space $\tilde{\boldsymbol{x}}$, i.e., the *distance from face space* (DFFS), can be computed directly in pixel space and represents the "faceness" of a particular image.[9] It is also possible to measure the distance between two different faces in face space as

$$\text{DIFS}(\boldsymbol{x}, \boldsymbol{y}) = \|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{y}}\| = \sqrt{\sum_{i=0}^{M-1} (a_i - b_i)^2}, \tag{14.13}$$

where the $b_i = (\boldsymbol{y} - \boldsymbol{m}) \cdot \boldsymbol{u}_i$ are the eigenface coefficients corresponding to $\boldsymbol{y}$.

Computing such distances in Euclidean vector space, however, does not exploit the additional information that the eigenvalue decomposition of our covariance matrix (14.10) provides. If we interpret the covariance matrix $\boldsymbol{C}$ as the covariance of a multi-variate Gaussian (Appendix B.1.1),[10] we can turn the DIFS into a log likelihood by computing the *Mahalanobis distance*

$$\text{DIFS}' = \|\tilde{\boldsymbol{x}} - \boldsymbol{m}\|_{\boldsymbol{C}^{-1}} = \sqrt{\sum_{i=0}^{M-1} a_i^2/\lambda_i^2}. \tag{14.14}$$

Instead of measuring the squared distance along each principal component in face space F, the Mahalanobis distance measures the *ratio* between the squared distance and the corresponding *variance* $\sigma_i^2 = \lambda_i$ and then sums these squared ratios (per-component log-likelihoods). An alternative way to implement this is to pre-scale each eigenvector by the inverse square root of its corresponding eigenvalue,

$$\hat{\boldsymbol{U}} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1/2}. \tag{14.15}$$

This *whitening* transformation then means that Euclidean distances in feature (face) space now correspond directly to log likelihoods (Moghaddam, Jebara, and Pentland 2000). (This same whitening approach can also be used in feature-based matching algorithms, as discussed in Section 4.1.3.)

If the distribution in eigenface space is very elongated, the Mahalanobis distance properly scales the components to come up with a sensible (probabilistic) distance from the mean. A similar analysis can be performed for computing a sensible difference from face space (DFFS) (Moghaddam and Pentland 1997) and the two terms can be combined to produce an estimate of the likelihood of being a true face, which can be useful in doing face detection (Section 14.1.1). More detailed explanations of probabilistic and Bayesian PCA can be found in textbooks on statistical learning (Hastie, Tibshirani, and Friedman 2001; Bishop 2006), which also discuss techniques for selecting the optimum number of components M to use in modeling a distribution.

[9] This can be used to form a simple face detector, as mentioned in Section 14.1.1.

[10] The ellipse shown in Figure 14.14 denotes an equi-probability contour of this multi-variate Gaussian.

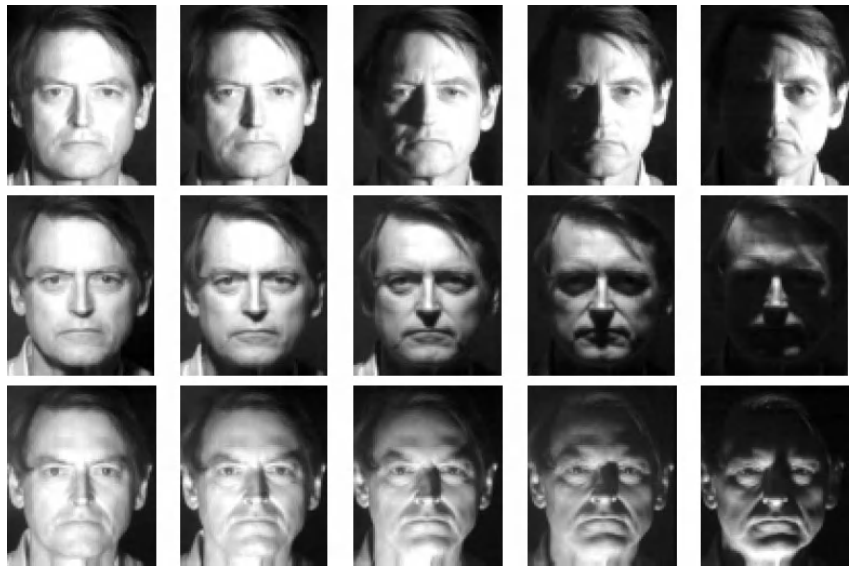


Figure 14.15 Images from the Harvard database used by Belhumeur, Hespanha, and Kriegman (1997) © 1997 IEEE. Note the wide range of illumination variation, which can be more dramatic than inter-personal variations.

One of the biggest advantages of using eigenfaces is that they reduce the comparison of a new face image $\boldsymbol{x}$ to a prototype (training) face image $\boldsymbol{x}_k$ (one of the colored xs in Figure 14.14) from a P-dimensional difference in pixel space to an M-dimensional difference in face space,

$$\|\boldsymbol{x} - \boldsymbol{x}_k\| = \|\boldsymbol{a} - \boldsymbol{a}_k\|, \tag{14.16}$$

where $\boldsymbol{a} = \boldsymbol{U}^T(\boldsymbol{x} - \boldsymbol{m})$ (14.11) involves computing a dot product between the signed difference-from-mean image $(\boldsymbol{x} - \boldsymbol{m})$ and each of the eigenfaces $\boldsymbol{u}_i$. Once again, however, this Euclidean distance ignores the fact that we have more information about face likelihoods available in the distribution of training images.

Consider the set of images of one person taken under a wide range of illuminations shown in Figure 14.15. As you can see, the *intrapersonal* variability within these images is much greater than the typical *extrapersonal* variability between any two people taken under the same illumination. Regular PCA analysis fails to distinguish between these two sources of variability and may, in fact, devote most of its principal components to modeling the intrapersonal variability.

If we are going to approximate faces by a linear subspace, it is more useful to have a space that *discriminates* between different classes (people) and is less sensitive to within-class variations (Belhumeur, Hespanha, and Kriegman 1997). Consider the three classes shown as different colors in Figure 14.16. As you can see, the distributions within a class (indicated by the tilted colored axes) are elongated and tilted with respect to the main face space PCA, which is aligned with the black x and y axes. We can compute the total *within-class* scatter matrix as

$$\boldsymbol{S}_{\mathrm{W}} = \sum_{k=0}^{K-1} \boldsymbol{S}_k = \sum_{k=0}^{K-1} \sum_{i \in C_k} (\boldsymbol{x}_i - \boldsymbol{m}_k)(\boldsymbol{x}_i - \boldsymbol{m}_k)^T, \tag{14.17}$$

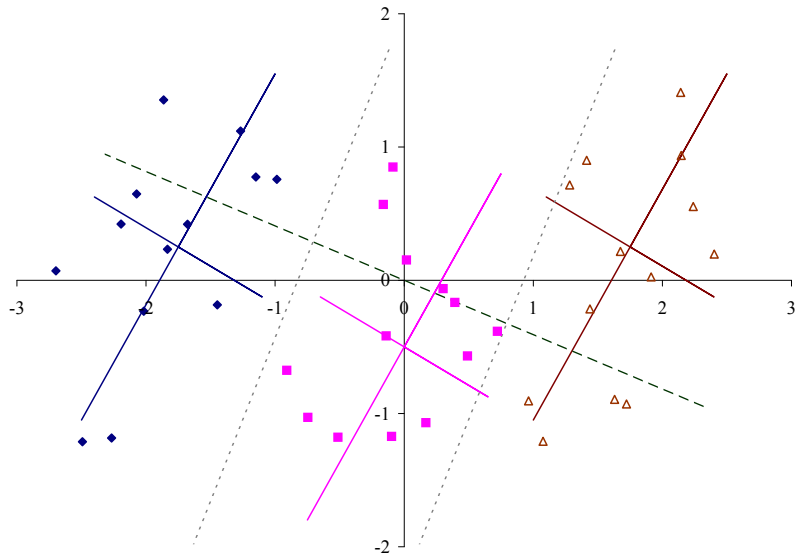


Figure 14.16 Simple example of Fisher linear discriminant analysis. The samples come from three different classes, shown in different colors along with their principal axes, which are scaled to $2\sigma_i$. (The intersections of the tilted axes are the class means $\boldsymbol{m}_k$.) The dashed line is the (dominant) Fisher linear discriminant direction and the dotted lines are the linear discriminants between the classes. Note how the discriminant direction is a blend between the principal directions of the between-class and within-class scatter matrices.

where $\boldsymbol{m}_k$ is the mean of class k and $\boldsymbol{S}_k$ is its within-class scatter matrix.[11] Similarly, we can compute the *between-class* scatter as

$$\boldsymbol{S}_{\mathrm{B}} = \sum_{k=0}^{K-1} N_k (\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^T, \tag{14.18}$$

where N_k are the number of exemplars in each class and $\boldsymbol{m}$ is the overall mean. For the three distributions shown in Figure 14.16, we have

$$\boldsymbol{S}_{\mathrm{W}} = 3N \begin{bmatrix} 0.246 & 0.183 \\ 0.183 & 0.457 \end{bmatrix} \quad \text{and} \quad \boldsymbol{S}_{\mathrm{B}} = N \begin{bmatrix} 6.125 & 0 \\ 0 & 0.375 \end{bmatrix}, \tag{14.19}$$

where $N = N_k = 13$ is the number of samples in each class.

To compute the most discriminating direction, *Fisher's linear discriminant* (FLD) (Belhumeur, Hespanha, and Kriegman 1997; Hastie, Tibshirani, and Friedman 2001; Bishop 2006), which is also known as *linear discriminant analysis* (LDA), selects the direction $\boldsymbol{u}$ that results in the largest ratio between the projected between-class and within-class variations

$$\boldsymbol{u}^* = \arg\max_{\boldsymbol{u}} \frac{\boldsymbol{u}^T \boldsymbol{S}_{\mathrm{B}} \boldsymbol{u}}{\boldsymbol{u}^T \boldsymbol{S}_{\mathrm{W}} \boldsymbol{u}}, \tag{14.20}$$

[11] To be consistent with Belhumeur, Hespanha, and Kriegman (1997), we use $\boldsymbol{S}_{\mathrm{W}}$ and $\boldsymbol{S}_{\mathrm{B}}$ to denote the scatter matrices, even though we use $\boldsymbol{C}$ elsewhere (14.9).

which is equivalent to finding the eigenvector corresponding to the largest eigenvalue of the generalized eigenvalue problem

$$\boldsymbol{S}_{\mathrm{B}}\boldsymbol{u} = \lambda \boldsymbol{S}_{\mathrm{W}}\boldsymbol{u} \quad \text{or} \quad \lambda \boldsymbol{u} = \boldsymbol{S}_{\mathrm{W}}^{-1}\boldsymbol{S}_{\mathrm{B}}\boldsymbol{u}. \tag{14.21}$$

For the problem shown in Figure 14.16,

$$\boldsymbol{S}_{\mathrm{W}}^{-1}\boldsymbol{S}_{\mathrm{B}} = \begin{bmatrix} 11.796 & -0.289 \\ -4.715 & 0.3889 \end{bmatrix} \quad \text{and} \quad \boldsymbol{u} = \begin{bmatrix} 0.926 \\ -0.379 \end{bmatrix} \tag{14.22}$$

As you can see, using this direction results in a better separation between the classes than using the dominant PCA direction, which is the horizontal axis. In their paper, Belhumeur, Hespanha, and Kriegman (1997) show that Fisherfaces significantly outperform the original eigenfaces algorithm, especially when faces have large amounts of illumination variation, as in Figure 14.15.

An alternative for modeling within-class (intrapersonal) and between-class (extrapersonal) variations is to model each distribution separately and then use Bayesian techniques to find the closest exemplar (Moghaddam, Jebara, and Pentland 2000). Instead of computing the mean for each class and then the within-class and between-class distributions, consider evaluating the difference images

$$\boldsymbol{\Delta}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j \tag{14.23}$$

between all pairs of training images $(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The differences between pairs that are in the same class (the same person) are used to estimate the intrapersonal covariance matrix $\boldsymbol{\Sigma}_I$, while differences between different people are used to estimate the extrapersonal covariance $\boldsymbol{\Sigma}_E$.[12] The principal components (eigenfaces) corresponding to these two classes are shown in Figure 14.17.

At recognition time, we can compute the distance $\boldsymbol{\Delta}_i$ between a new face $\boldsymbol{x}$ and a stored training image $\boldsymbol{x}_i$ and evaluate its intrapersonal likelihood as

$$p_I(\boldsymbol{\Delta}_i) = p_{\mathcal{N}}(\boldsymbol{\Delta}_i; \boldsymbol{\Sigma}_I) = \frac{1}{|2\pi\boldsymbol{\Sigma}_I|^{1/2}} \exp -\|\boldsymbol{\Delta}_i\|^2_{\boldsymbol{\Sigma}_I^{-1}}, \tag{14.24}$$

where $p_{\mathcal{N}}$ is a normal (Gaussian) distribution with covariance $\boldsymbol{\Sigma}_I$ and

$$|2\pi\boldsymbol{\Sigma}_I|^{1/2} = (2\pi)^{M/2} \prod_{j=1}^{M} \lambda_j^{1/2} \tag{14.25}$$

is its volume. The Mahalanobis distance

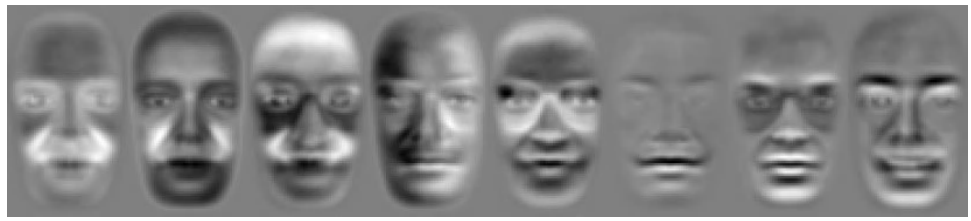
$$\|\boldsymbol{\Delta}_i\|^2_{\boldsymbol{\Sigma}_I^{-1}} = \boldsymbol{\Delta}_i^T \boldsymbol{\Sigma}_I^{-1} \boldsymbol{\Delta}_i = \|\boldsymbol{a}^I - \boldsymbol{a}_i^I\|^2 \tag{14.26}$$

can be computed more efficiently by first projecting the new image $\boldsymbol{x}$ into the whitened intrapersonal face space (14.15)

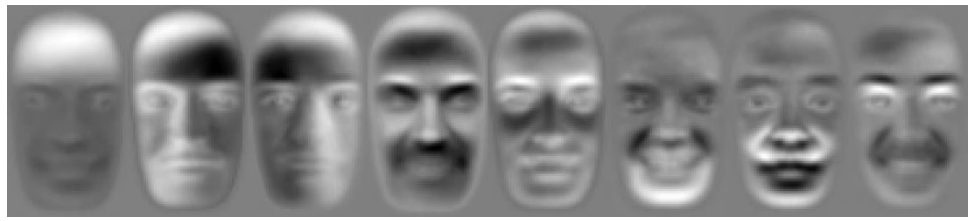
$$\boldsymbol{a}^I = \hat{\boldsymbol{U}}^I \boldsymbol{x} \tag{14.27}$$

and then computing a Euclidean distance to the training image vector $\boldsymbol{a}_i^I$, which can be precomputed offline. The extrapersonal likelihood $p_E(\boldsymbol{\Delta}_i)$ can be computed in a similar fashion.

[12] Note that the difference distributions are zero mean because for every $\boldsymbol{\Delta}_{ij}$ there corresponds a negative $\boldsymbol{\Delta}_{ji}$.



(a)



(b)

Figure 14.17 "Dual" eigenfaces (Moghaddam, Jebara, and Pentland 2000) © 2000 Elsevier: (a) intrapersonal and (b) extrapersonal.

Once the intrapersonal and extrapersonal likelihoods have been computed, we can compute the Bayesian likelihood of a new image $\boldsymbol{x}$ matching a training image $\boldsymbol{x}_i$ as

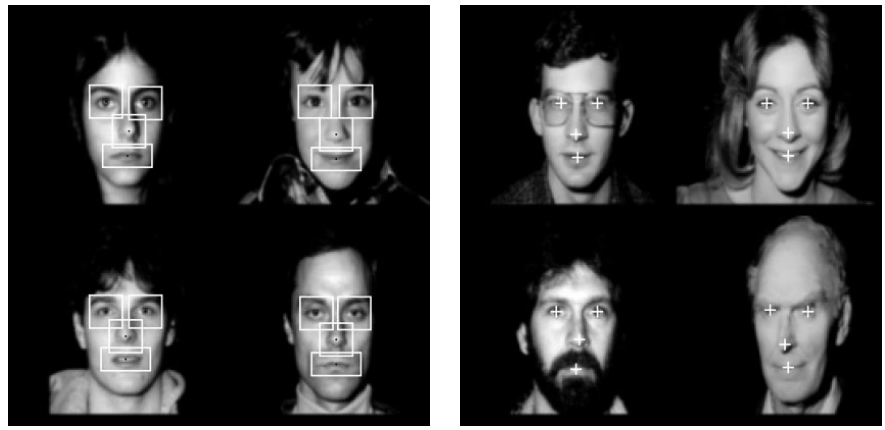
$$p(\boldsymbol{\Delta}_i) = \frac{p_I(\boldsymbol{\Delta}_i) l_I}{p_I(\boldsymbol{\Delta}_i) l_I + p_E(\boldsymbol{\Delta}_i) l_E}, \tag{14.28}$$

where l_I and l_E are the prior probabilities of two images being in the same or in different classes (Moghaddam, Jebara, and Pentland 2000). A simpler approach, which does not require the evaluation of extrapersonal probabilities, is to simply choose the training image with the highest likelihood $p_I(\boldsymbol{\Delta}_i)$. In this case, nearest neighbor search techniques in the space spanned by the precomputed $\{\boldsymbol{a}_i^I\}$ vectors could be used to speed up finding the best match.[13]

Another way to improve the performance of eigenface-based approaches is to break up the image into separate regions such as the eyes, nose, and mouth (Figure 14.18) and to match each of these *modular eigenspaces* independently (Moghaddam and Pentland 1997; Heisele, Ho, Wu *et al.* 2003; Heisele, Serre, and Poggio 2007). The advantage of such a modular approach is that it can tolerate a wider range of viewpoints, because each part can move relative to the others. It also supports a larger variety of combinations, e.g., we can model one person as having a narrow nose and bushy eyebrows, without requiring the eigenfaces to span all possible combinations of nose, mouth, and eyebrows. (If you remember the cardboard children's books where you can select different top and bottom faces, or Mr. Potato Head, you get the idea.)

Another approach to dealing with large variability in appearance is to create *view-based* (view-specific) eigenspaces, as shown in Figure 14.19 (Moghaddam and Pentland 1997). We can think of these view-based eigenspaces as local descriptors that select different axes depending on which part of the face space you are in. Note that such approaches, however,

[13] Note that while the covariance matrices $\boldsymbol{\Sigma}_I$ and $\boldsymbol{\Sigma}_E$ are computed by looking at differences between *all* pairs of images, the run-time evaluation selects the *nearest* image to determine the facial identity. Whether this is statistically correct is explored in Exercise 14.4.



(a) (b)

Figure 14.18 Modular eigenspace for face recognition (Moghaddam and Pentland 1997) © 1997 IEEE. (a) By detecting separate features in the faces (eyes, nose, mouth), separate eigenspaces can be estimated for each one. (b) The relative positions of each feature can be detected at recognition time, thus allowing for more flexibility in viewpoint and expression.

potentially require large amounts of training data, i.e., pictures of every person in every possible pose or expression. This is in contrast to the shape and appearance models we study in Section 14.2.2, which can learn deformations across all individuals.

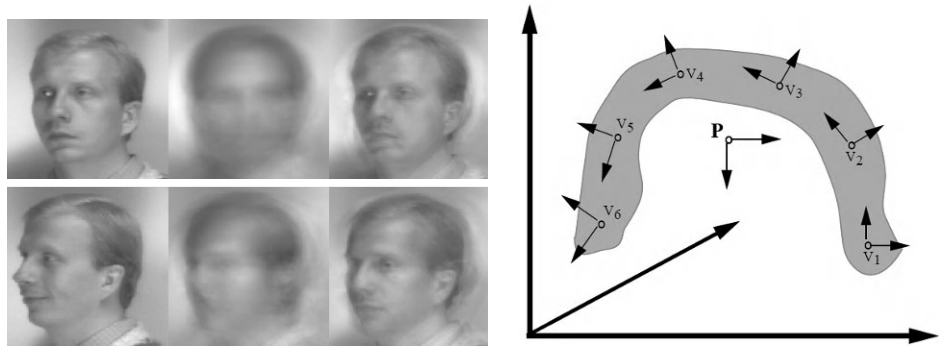
It is also possible to generalize the bilinear factorization implicit in PCA and SVD approaches to multilinear (tensor) formulations that can model several interacting factors simultaneously (Vasilescu and Terzopoulos 2007). These ideas are related to currently active topics in machine learning such as *subspace learning* (Cai, He, Hu *et al.* 2007), *local distance functions* (Frome, Singer, Sha *et al.* 2007), and *metric learning* (Ramanan and Baker 2009). Learning approaches play an increasingly important role in face recognition, e.g., in the work of Sivic, Everingham, and Zisserman (2009) and Guillaumin, Verbeek, and Schmid (2009).

14.2.2 Active appearance and 3D shape models

The need to use modular or view-based eigenspaces for face recognition is symptomatic of a more general observation, i.e., that facial appearance and identifiability depend as much on *shape* as they do on color or texture (which is what eigenfaces capture). Furthermore, when dealing with 3D head rotations, the *pose* of a person's head should be discounted when performing recognition.

In fact, the earliest face recognition systems, such as those by Fischler and Elschlager (1973), Kanade (1977), and Yuille (1991), found distinctive feature points on facial images and performed recognition on the basis of their relative positions or distances. Newer techniques such as *local feature analysis* (Penev and Atick 1996) and *elastic bunch graph matching* (Wiskott, Fellous, Krüger *et al.* 1997) combine local filter responses (jets) at distinctive feature locations together with shape models to perform recognition.

A visually compelling example of why both shape and texture are important is the work of Rowland and Perrett (1995), who manually traced the contours of facial features and then



(a) (b)

Figure 14.19 View-based eigenspace (Moghaddam and Pentland 1997) © 1997 IEEE. (a) Comparison between a regular (parametric) eigenspace reconstruction (middle column) and a view-based eigenspace reconstruction (right column) corresponding to the input image (left column). The top row is from a training image, the bottom row is from the test set. (b) A schematic representation of the two approaches, showing how each view computes its own local basis representation.



(a) (b) (c) (d) (e)

Figure 14.20 Manipulating facial appearance through shape and color (Rowland and Perrett 1995) © 1995 IEEE. By adding or subtracting gender-specific shape and color characteristics to (b) an input image, different amounts of gender variation can be induced. The amounts added (from the mean) are: (a) +50% (gender enhancement), (c) -50% (near "androgyny"), (d) -100% (gender switched), and (e) -150% (opposite gender attributes enhanced).



(a) (b) (c)

Figure 14.21 Active Appearance Models (Cootes, Edwards, and Taylor 2001) © 2001 IEEE: (a) input image with registered feature points; (b) the feature points (shape vector $\boldsymbol{s}$); (c) the shape-free appearance image (texture vector $\boldsymbol{t}$).

used these contours to normalize (warp) each image to a canonical shape. After analyzing both the shape and color images for deviations from the mean, they were able to associate certain shape and color deformations with personal characteristics such as age and gender (Figure 14.20). Their work demonstrates that both shape and color have an important influence on the perception of such characteristics.

Around the same time, researchers in computer vision were beginning to use simultaneous shape deformations and texture interpolation to model the variability in facial appearance caused by identity or expression (Beymer 1996; Vetter and Poggio 1997), developing techniques such as Active Shape Models (Lanitis, Taylor, and Cootes 1997), 3D Morphable Models (Blanz and Vetter 1999), and Elastic Bunch Graph Matching (Wiskott, Fellous, Krüger *et al.* 1997).[14]

Of all these techniques, the *active appearance models* (AAMs) of Cootes, Edwards, and Taylor (2001) are among the most widely used for face recognition and tracking. Like other shape and texture models, an AAM models both the variation in the shape of an image $\boldsymbol{s}$, which is normally encoded by the location of key feature points on the image (Figure 14.21b), as well as the variation in texture $\boldsymbol{t}$, which is normalized to a canonical shape before being analyzed (Figure 14.21c).[15]

Both shape and texture are represented as deviations from a mean shape $\bar{\boldsymbol{s}}$ and texture $\bar{\boldsymbol{t}}$,

$$\boldsymbol{s} = \bar{\boldsymbol{s}} + \boldsymbol{U}_s \boldsymbol{a} \tag{14.29}$$

$$\boldsymbol{t} = \bar{\boldsymbol{t}} + \boldsymbol{U}_t \boldsymbol{a}, \tag{14.30}$$

where the eigenvectors in $\boldsymbol{U}_s$ and $\boldsymbol{U}_t$ have been pre-scaled (whitened) so that unit vectors in $\boldsymbol{a}$ represent one standard deviation of variation observed in the training data. In addition to these principal deformations, the shape parameters are transformed by a global similarity to match the location, size, and orientation of a given face. Similarly, the texture image contains a scale and offset to best match novel illumination conditions.

As you can see, the same appearance parameters $\boldsymbol{a}$ in (14.29–14.30) simultaneously control both the shape and texture deformations from the mean, which makes sense if we believe

[14] We have already seen the application of PCA to 3D head and face modeling and animation in Section 12.6.3.

[15] When only the shape variation is being captured, such models are called *active shape models* (ASMs) (Cootes, Cooper, Taylor *et al.* 1995; Davies, Twining, and Taylor 2008). These were already discussed in Section 5.1.1 (5.13–5.17).

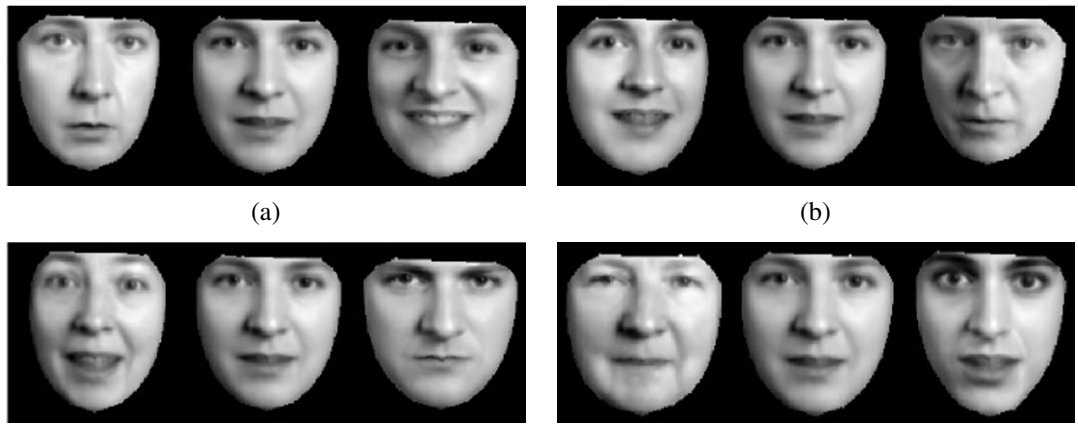


Figure 14.22 Principal modes of variation in active appearance models (Cootes, Edwards, and Taylor 2001) © 2001 IEEE. The four images show the effects of simultaneously changing the first four modes of variation in both shape and texture by $\pm\sigma$ from the mean. You can clearly see how the shape of the face and the shading are simultaneously affected.

them to be correlated. Figure 14.22 shows how moving three standard deviations along each of the first four principal directions ends up changing several correlated factors in a person's appearance, including expression, gender, age, and identity.

In order to fit an active appearance model to a novel image, Cootes, Edwards, and Taylor (2001) pre-compute a set of "difference decomposition" images, using an approach related to other fast techniques for incremental tracking, such as those we discussed in Sections 4.1.4, 8.1.3, and 8.2 (Gleicher 1997; Hager and Belhumeur 1998), which often *learn* a discriminative mapping between matching errors and incremental displacements (Avidan 2001; Jurie and Dhome 2002; Liu, Chen, and Kumar 2003; Sclaroff and Isidoro 2003; Romdhani and Vetter 2003; Williams, Blake, and Cipolla 2003).

In more detail, Cootes, Edwards, and Taylor (2001) compute the derivatives of a set of training images with respect to each of the parameters in $\boldsymbol{a}$ using finite differences and then compute a set of *displacement weight* images

$$\boldsymbol{W} = \left[\frac{\partial \boldsymbol{x}^T}{\partial \boldsymbol{a}} \; \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{a}}\right]^{-1} \frac{\partial \boldsymbol{x}^T}{\partial \boldsymbol{a}}, \tag{14.31}$$

which can be multiplied by the current error residual to produce an update step in the parameters, $\delta \boldsymbol{a} = -\boldsymbol{W}\boldsymbol{r}$. Matthews and Baker (2004) use their *inverse compositional method*, which they first developed for parametric optical flow (8.64–8.65), to further speed up active appearance model fitting and tracking. Examples of AAMs being fitted to two input images are shown in Figure 14.23.

Although active appearance models are primarily designed to accurately capture the variability in appearance and deformation that are characteristic of faces, they can be adapted to face recognition by computing an identity subspace that separates variation in identity from other sources of variability such as lighting, pose, and expression (Costen, Cootes, Edwards *et al.* 1999). The basic idea, which is modeled after similar work in eigenfaces (Belhumeur,



Figure 14.23 Multiresolution model fitting (search) in active appearance models (Cootes, Edwards, and Taylor 2001) © 2001 IEEE. The columns show the initial model, the results after 3, 8, and 11 iterations, and the final convergence. The rightmost column shows the input image.

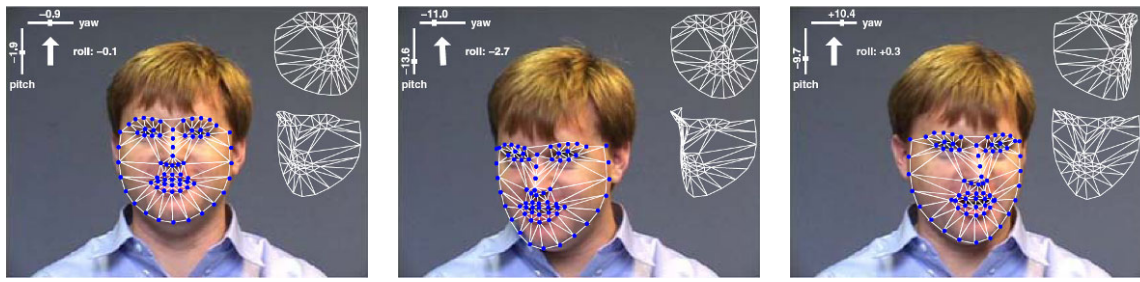
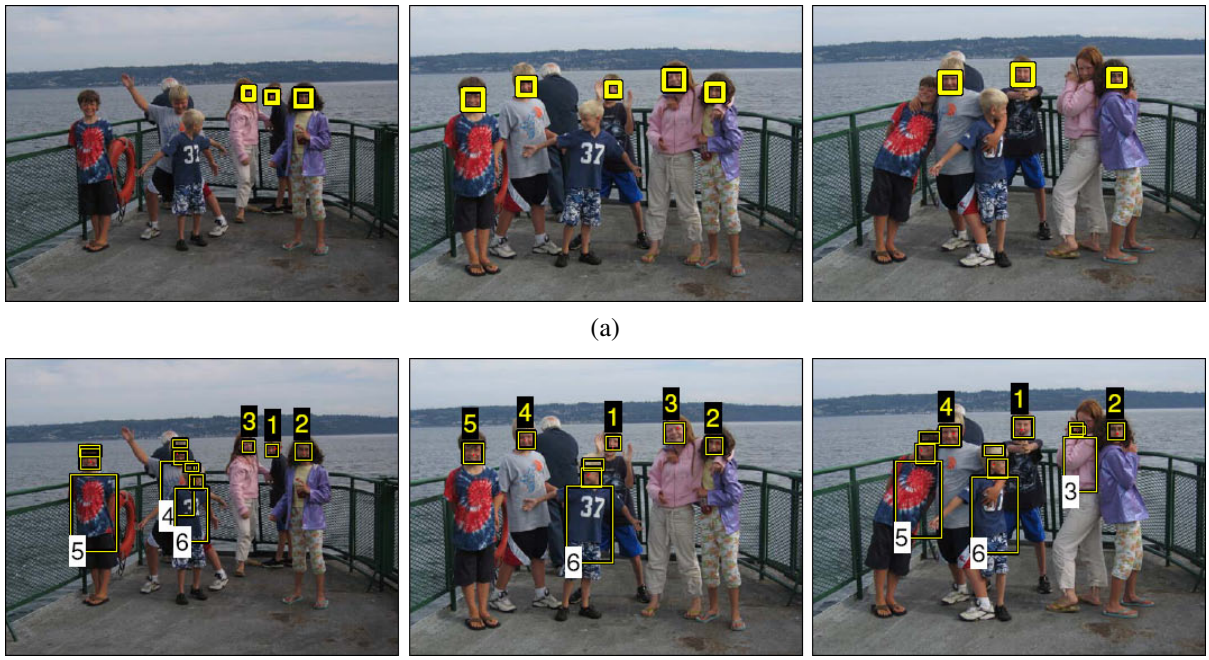


Figure 14.24 Head tracking with 3D AAMs (Matthews, Xiao, and Baker 2007) © 2007 Springer. Each image shows a video frame along with the estimate yaw, pitch, and roll parameters and the fitted 3D deformable mesh.

Hespanha, and Kriegman 1997; Moghaddam, Jebara, and Pentland 2000), is to compute separate statistics for intrapersonal and extrapersonal variation and then find discriminating directions in these subspaces. While AAMs have sometimes been used directly for recognition (Blanz and Vetter 2003), their main use in the context of recognition is to align faces into a canonical pose (Liang, Xiao, Wen *et al.* 2008) so that more traditional methods of face recognition (Penev and Atick 1996; Wiskott, Fellous, Krüger *et al.* 1997; Ahonen, Hadid, and Pietikäinen 2006; Zhao and Pietikäinen 2007; Cao, Yin, Tang *et al.* 2010) can be used. AAMs (or, actually, their simpler version, Active Shape Models (ASMs)) can also be used to align face images to perform automated morphing (Zanella and Fuentes 2004).

Active appearance models continue to be an active research area, with enhancements to deal with illumination and viewpoint variation (Gross, Baker, Matthews *et al.* 2005) as well as occlusions (Gross, Matthews, and Baker 2006). One of the most significant extensions is to construct 3D models of shape (Matthews, Xiao, and Baker 2007), which are much better at capturing and explaining the full variability of facial appearance across wide changes in pose.



(a)

(b)

Figure 14.25 Person detection and re-recognition using a combined face, hair, and torso model (Sivic, Zitnick, and Szeliski 2006) © 2006 Springer. (a) Using face detection alone, several of the heads are missed. (b) The combined face and clothing model successfully re-finds all the people.

Such models can be constructed either from monocular video sequences (Matthews, Xiao, and Baker 2007), as shown in Figure 14.24, or from multi-view video sequences (Ramnath, Koterba, Xiao *et al.* 2008), which provide even greater reliability and accuracy in reconstruction and tracking. (For a recent review of progress in head pose estimation, please see the survey paper by Murphy-Chutorian and Trivedi (2009).)

14.2.3 *Application*: Personal photo collections

In addition to digital cameras automatically finding faces to aid in auto-focusing and video cameras finding faces in video conferencing to center on the speaker (either mechanically or digitally), face detection has found its way into most consumer-level photo organization packages, such as iPhoto, Picasa, and Windows Live Photo Gallery. Finding faces and allowing users to tag them makes it easier to find photos of selected people at a later date or to automatically share them with friends. In fact, the ability to tag friends in photos is one of the more popular features on Facebook.

Sometimes, however, faces can be hard to find and recognize, especially if they are small, turned away from the camera, or otherwise occluded. In such cases, combining face recognition with person detection and clothes recognition can be very effective, as illustrated in Figure 14.25 (Sivic, Zitnick, and Szeliski 2006). Combining person recognition with other kinds of context, such as location recognition (Section 14.3.3) or activity or event recognition, can also help boost performance (Lin, Kapoor, Hua *et al.* 2010).



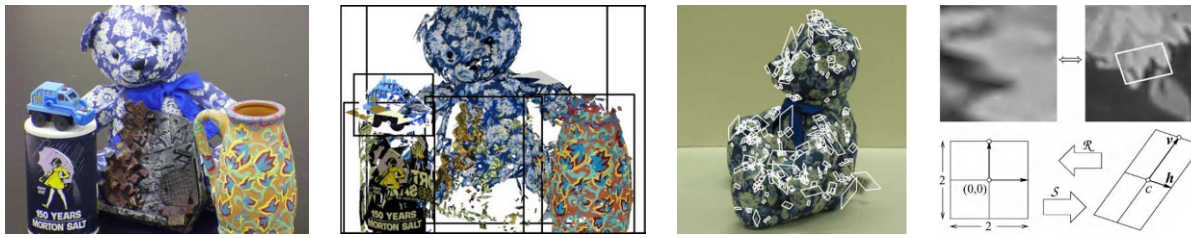
Figure 14.26 Recognizing objects in a cluttered scene (Lowe 2004) © 2004 Springer. Two of the training images in the database are shown on the left. They are matched to the cluttered scene in the middle using SIFT features, shown as small squares in the right image. The affine warp of each recognized database image onto the scene is shown as a larger parallelogram in the right image.

14.3 Instance recognition

General object recognition falls into two broad categories, namely *instance recognition* and *class recognition*. The former involves re-recognizing a known 2D or 3D rigid object, potentially being viewed from a novel viewpoint, against a cluttered background, and with partial occlusions. The latter, which is also known as *category-level* or *generic* object recognition (Ponce, Hebert, Schmid *et al.* 2006), is the much more challenging problem of recognizing any instance of a particular general class such as "cat", "car", or "bicycle".

Over the years, many different algorithms have been developed for instance recognition. Mundy (2006) surveys earlier approaches, which focused on extracting lines, contours, or 3D surfaces from images and matching them to known 3D object models. Another popular approach was to acquire images from a large set of viewpoints and illuminations and to represent them using an eigenspace decomposition (Murase and Nayar 1995). More recent approaches (Lowe 2004; Rothganger, Lazebnik, Schmid *et al.* 2006; Ferrari, Tuytelaars, and Van Gool 2006b; Gordon and Lowe 2006; Obdržálek and Matas 2006; Sivic and Zisserman 2009) tend to use viewpoint-invariant 2D features, such as those we saw in Section 4.1.2. After extracting informative sparse 2D features from both the new image and the images in the database, image features are matched against the object database, using one of the sparse feature matching strategies described in Section 4.1.3. Whenever a sufficient number of matches have been found, they are verified by finding a geometric transformation that aligns the two sets of features (Figure 14.26).

Below, we describe some of the techniques that have been proposed for representing the geometric relationships between such features (Section 14.3.1). We also discuss how to make the feature matching process more efficient using ideas from text and information retrieval (Section 14.3.2).



(a) (b) (c) (d)

Figure 14.27 3D object recognition with affine regions (Rothganger, Lazebnik, Schmid *et al.* 2006) © 2006 Springer: (a) sample input image; (b) five of the recognized (reprojected) objects along with their bounding boxes; (c) a few of the local affine regions; (d) local affine region (patch) reprojected into a canonical (square) frame, along with its geometric affine transformations.

14.3.1 Geometric alignment

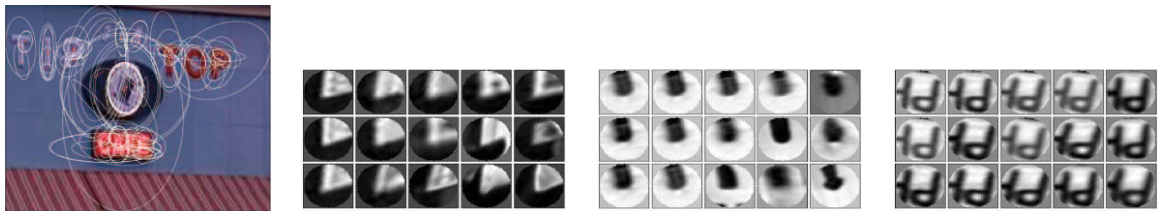
To recognize one or more instances of some known objects, such as those shown in the left column of Figure 14.26, the recognition system first extracts a set of interest points in each database image and stores the associated descriptors (and original positions) in an indexing structure such as a search tree (Section 4.1.3). At recognition time, features are extracted from the new image and compared against the stored object features. Whenever a sufficient number of matching features (say, three or more) are found for a given object, the system then invokes a *match verification* stage, whose job is to determine whether the spatial arrangement of matching features is consistent with those in the database image.

Because images can be highly cluttered and similar features may belong to several objects, the original set of feature matches can have a large number of outliers. For this reason, Lowe (2004) suggests using a Hough transform (Section 4.3.2) to accumulate votes for likely geometric transformations. In his system, he uses an affine transformation between the database object and the collection of scene features, which works well for objects that are mostly planar, or where at least several corresponding features share a quasi-planar geometry.[16]

Since SIFT features carry with them their own location, scale, and orientation, Lowe uses a four-dimensional similarity transformation as the original Hough binning structure, i.e., each bin denotes a particular location for the object center, scale, and in-plane rotation. Each matching feature votes for the nearest 2^4 bins and peaks in the transform are then selected for a more careful affine motion fit. Figure 14.26 (right image) shows three instances of the two objects on the left that were recognized by the system. Obdržálek and Matas (2006) generalize Lowe's approach to use feature descriptors with full local affine frames and evaluate their approach on a number of object recognition databases.

Another system that uses local affine frames is the one developed by Rothganger, Lazebnik, Schmid *et al.* (2006). In their system, the affine region detector of Mikolajczyk and Schmid (2004) is used to rectify local image patches (Figure 14.27d), from which both a SIFT descriptor and a 10×10 UV color histogram are computed and used for matching and recognition. Corresponding patches in different views of the same object, along with

[16] When a larger number of features is available, a full fundamental matrix can be used (Brown and Lowe 2002; Gordon and Lowe 2006). When image stitching is being performed (Brown and Lowe 2007), the motion models discussed in Section 9.1 can be used instead.



(a) (b)

Figure 14.28 Visual words obtained from elliptical normalized affine regions (Sivic and Zisserman 2009) © 2009 IEEE. (a) Affine covariant regions are extracted from each frame and clustered into visual words using k-means clustering on SIFT descriptors with a learned Mahalanobis distance. (b) The central patch in each grid shows the query and the surrounding patches show the nearest neighbors.

their local affine deformations, are used to compute a 3D affine model for the object using an extension of the factorization algorithm of Section 7.3, which can then be upgraded to a Euclidean reconstruction (Tomasi and Kanade 1992).

At recognition time, local Euclidean neighborhood constraints are used to filter potential matches, in a manner analogous to the affine geometric constraints used by Lowe (2004) and Obdržálek and Matas (2006). Figure 14.27 shows the results of recognizing five objects in a cluttered scene using this approach.

While feature-based approaches are normally used to detect and localize known objects in scenes, it is also possible to get pixel-level segmentations of the scene based on such matches. Ferrari, Tuytelaars, and Van Gool (2006b) describe such a system for simultaneously recognizing objects and segmenting scenes, while Kannala, Rahtu, Brandt *et al.* (2008) extend this approach to non-rigid deformations. Section 14.4.3 re-visits this topic of joint recognition and segmentation in the context of generic class (category) recognition.

14.3.2 Large databases

As the number of objects in the database starts to grow large (say, millions of objects or video frames being searched), the time it takes to match a new image against each database image can become prohibitive. Instead of comparing the images one at a time, techniques are needed to quickly narrow down the search to a few likely images, which can then be compared using a more detailed and conservative verification stage.

The problem of quickly finding partial matches between documents is one of the central problems in *information retrieval* (IR) (Baeza-Yates and Ribeiro-Neto 1999; Manning, Raghavan, and Schütze 2008). The basic approach in fast document retrieval algorithms is to pre-compute an *inverted index* between individual words and the documents (or Web pages or news stories) where they occur. More precisely, the *frequency* of occurrence of particular words in a document is used to quickly find documents that match a particular query.

Sivic and Zisserman (2009) were the first to adapt IR techniques to visual search. In their Video Google system, affine invariant features are first detected in all the video frames they are indexing using both *shape adapted* regions around Harris feature points (Schaffalitzky and Zisserman 2002; Mikolajczyk and Schmid 2004) and maximally stable extremal regions (Matas, Chum, Urban *et al.* 2004), (Section 4.1.1), as shown in Figure 14.28a. Next, 128-



(a) (b)

Figure 14.29 Matching based on visual words (Sivic and Zisserman 2009) © 2009 IEEE. (a) Features in the query region on the left are matched to corresponding features in a highly ranked video frame. (b) Results after removing the stop words and filtering the results using spatial consistency.

dimensional SIFT descriptors are computed from each normalized region (i.e., the patches shown in Figure 14.28b). Then, an average covariance matrix for these descriptors is estimated by accumulating statistics for features tracked from frame to frame. The feature descriptor covariance $\mathbf{\Sigma}$ is then used to define a Mahalanobis distance between feature descriptors,

$$d(\boldsymbol{x}_0, \boldsymbol{x}_1) = \|\boldsymbol{x}_0 - \boldsymbol{x}_1\|_{\mathbf{\Sigma}^{-1}} = \sqrt{(\boldsymbol{x}_0 - \boldsymbol{x}_1)^T \mathbf{\Sigma}^{-1} (\boldsymbol{x}_0 - \boldsymbol{x}_1)}. \tag{14.32}$$

In practice, feature descriptors are *whitened* by pre-multiplying them by $\mathbf{\Sigma}^{-1/2}$ so that Euclidean distances can be used.[17]

In order to apply fast information retrieval techniques to images, the high-dimensional feature descriptors that occur in each image must first be mapped into discrete *visual words*. Sivic and Zisserman (2003) perform this mapping using k-means clustering, while some of newer methods discussed below (Nistér and Stewénius 2006; Philbin, Chum, Isard *et al.* 2007) use alternative techniques, such as vocabulary trees or randomized forests. To keep the clustering time manageable, only a few hundred video frames are used to learn the cluster centers, which still involves estimating several thousand clusters from about 300,000 descriptors. At visual query time, each feature in a new query region (e.g., Figure 14.28a, which is a cropped region from a larger video frame) is mapped to its corresponding visual word. To keep very common patterns from contaminating the results, a *stop list* of the most common visual words is created and such words are dropped from further consideration.

Once a query image or region has been mapped into its constituent visual words, likely matching images or video frames must then be retrieved from the database. Information retrieval systems do this by matching word distributions (*term frequencies*) n_{id}/n_d between the query and target documents, where n_{id} is how many times word i occurs in document d, and n_d is the total number of words in document d. In order to downweight words that occur frequently and to focus the search on rarer (and hence, more informative) terms, an *inverse document frequency* weighting $\log N/N_i$ is applied, where N_i is the number of documents containing word i, and N is the total number of documents in the database. The combination of these two factors results in the *term frequency-inverse document frequency (tf-idf)* measure,

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{N_i}. \tag{14.33}$$

[17] Note that the computation of feature covariances from matched feature points is much more sensible than simply performing a PCA on the descriptor space (Winder and Brown 2007). This corresponds roughly to the *within-class* scatter matrix (14.17) we studied in Section 14.2.1.

1. **Vocabulary construction (off-line)**
   (a) Extract affine covariant regions from each database image.
   (b) Compute descriptors and optionally whiten them to make Euclidean distances meaningful (Sivic and Zisserman 2009).
   (c) Cluster the descriptors into visual words, either using k-means (Sivic and Zisserman 2009), hierarchical clustering (Nistér and Stewénius 2006), or randomized k-d trees (Philbin, Chum, Isard *et al.* 2007).
   (d) Decide which words are too common and put them in the stop list.
2. **Database construction (off-line)**
   (a) Compute term frequencies for the visual word in each image, document frequencies for each word, and normalized *tf-idf* vectors for each document.
   (b) Compute inverted indices from visual words to images (with word counts).
3. **Image retrieval (on-line)**
   (a) Extract regions, descriptors, and visual words, and compute a *tf-idf* vector for the query image or region.
   (b) Retrieve the top image candidates, either by exhaustively comparing sparse *tf-idf* vectors (Sivic and Zisserman 2009) or by using inverted indices to examine only a subset of the images (Nistér and Stewénius 2006).
   (c) Optionally re-rank or verify all the candidate matches, using either spatial consistency (Sivic and Zisserman 2009) or an affine (or simpler) transformation model (Philbin, Chum, Isard *et al.* 2007).
   (d) Optionally expand the answer set by re-submitting highly ranked matches as new queries (Chum, Philbin, Sivic *et al.* 2007).

Algorithm 14.2 Image retrieval using visual words (Sivic and Zisserman 2009; Nistér and Stewénius 2006; Philbin, Chum, Isard *et al.* 2007; Chum, Philbin, Sivic *et al.* 2007; Philbin, Chum, Sivic *et al.* 2008).

At match time, each document (or query region) is represented by its *tf-idf* vector,

$$\boldsymbol{t} = (t_1, \ldots, t_i, \ldots t_m). \tag{14.34}$$

The similarity between two documents is measured by the dot product between their corresponding normalized vectors $\hat{\boldsymbol{t}} = \boldsymbol{t}/\|\boldsymbol{t}\|$, which means that their dissimilarity is proportional to their Euclidean distance. In their journal paper, Sivic and Zisserman (2009) compare this simple metric to a dozen other metrics and conclude that it performs just about as well as more complicated metrics. Because the number of non-zero t_i terms in a typical query or document is small ($M \approx 200$) compared to the number of visual words ($V \approx 20,000$), the distance between pairs of (sparse) *tf-idf* vectors can be computed quite quickly.

After retrieving the top $N_s = 500$ documents based on word frequencies, Sivic and Zisserman (2009) re-rank these results using spatial consistency. This step involves taking every matching feature and counting the number of $k = 15$ nearest adjacent features that also match between the two documents. (This latter process is accelerated using inverted files, which we discuss in more detail below.) As shown in Figure 14.29, this step helps remove spurious false positive matches and produces a better estimate of which frames and regions in the video are actually true matches. Algorithm 14.2 summarizes the processing steps involved in image retrieval using visual words.

While this approach works well for tens of thousand of visual words and thousands of keyframes, as the size of the database continues to increase, both the time to quantize each feature and to find potential matching frames or images can become prohibitive. Nistér and Stewénius (2006) address this problem by constructing a hierarchical *vocabulary tree*, where feature vectors are hierarchically clustered into a k-way tree of prototypes. (This technique is also known as *tree-structured vector quantization* (Gersho and Gray 1991).) At both database construction time and query time, each descriptor vector is compared to several prototypes at a given level in the vocabulary tree and the branch with the closest prototype is selected for further refinement (Figure 14.30). In this way, vocabularies with millions (10^6) of words can be supported, which enables individual words to be far more discriminative, while only requiring $10 \cdot 6$ comparisons for quantizing each descriptor.

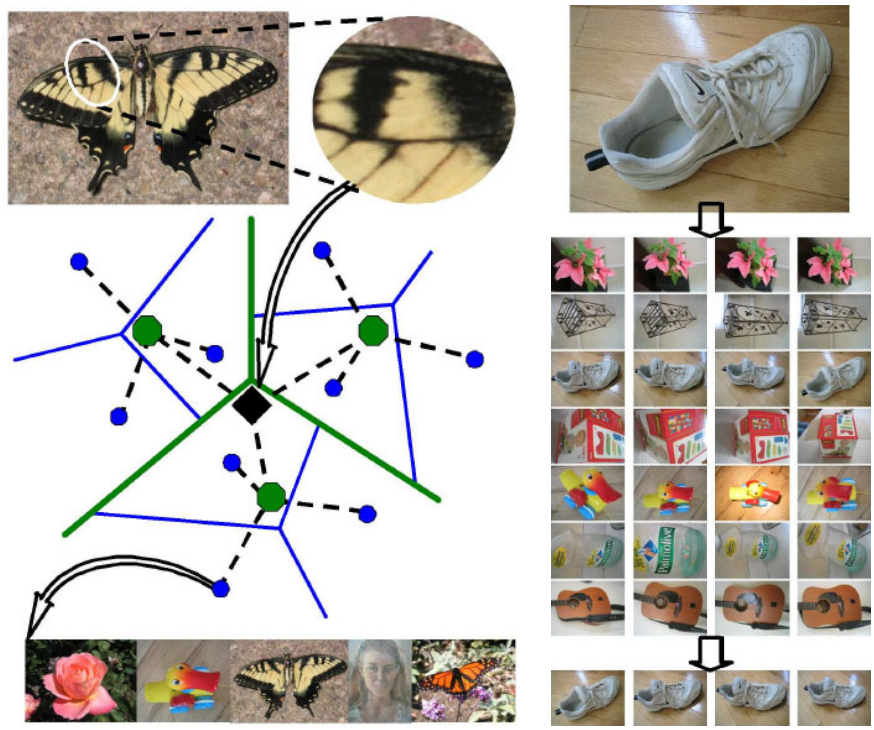
At query time, each node in the vocabulary tree keeps its own inverted file index, so that features that match a particular node in the tree can be rapidly mapped to potential matching images. (Interior leaf nodes just use the inverted indices of their corresponding leaf-node descendants.) To score a particular query *tf-idf* vector $\boldsymbol{t}_q$ against all document vectors $\{\boldsymbol{t}_j\}$ using an L_p metric,[18] the non-zero t_{iq} entries in $\boldsymbol{t}_q$ are used to fetch corresponding non-zero t_{ij} entries, and the L_p norm is efficiently computed as

$$\|\boldsymbol{t}_q - \boldsymbol{t}_j\|_p^p = 2 + \sum_{i|t_{iq}>0 \wedge t_{ij}>0} (|t_{iq} - t_{ij}|^p - |t_{iq}|^p - |t_{ij}|^p). \tag{14.35}$$

In order to mitigate quantization errors due to noise in the descriptor vectors, Nistér and Stewénius (2006) not only score leaf nodes in the vocabulary tree (corresponding to visual words), but also score interior nodes in the tree, which correspond to clusters of similar visual words.

Because of the high efficiency in both quantizing and scoring features, their vocabulary-tree-based recognition system is able to process incoming images in real time against a

[18] In their actual implementation, Nistér and Stewénius (2006) use an L_1 metric.



(a) (b)

Figure 14.30 Scalable recognition using a vocabulary tree (Nistér and Stewénius 2006) © 2006 IEEE. (a) Each MSER elliptical region is converted into a SIFT descriptor, which is then quantized by comparing it hierarchically to some prototype descriptors in a vocabulary tree. Each leaf node stores its own inverted index (sparse list of non-zero *tf-idf* counts) into images that contain that feature. (b) A recognition result, showing a query image (top row) being indexed into a database of 6000 test images and correctly finding the corresponding four images.

database of 40,000 CD covers and at 1Hz when matching a database of one million frames taken from six feature-length movies. Figure 14.30b shows some typical images from the database of objects taken under varying viewpoints and illumination that was used to train and test the vocabulary tree recognition system.

The state of the art in instance recognition continues to improve rapidly. Philbin, Chum, Isard *et al.* (2007) have shown that randomized forest of k-d trees perform better than vocabulary trees on a large location recognition task (Figure 14.31). They also compare the effects of using different 2D motion models (Section 2.1.2) in the verification stage. In follow-on work, Chum, Philbin, Sivic *et al.* (2007) apply another idea from information retrieval, namely *query expansion*, which involves re-submitting top-ranked images from the initial query as additional queries to generate additional candidate results, to further improve recognition rates for difficult (occluded or oblique) examples. Philbin, Chum, Sivic *et al.* (2008) show how to mitigate quantization problems in visual words selection using *soft assignment*, where each feature descriptor is mapped to a number of visual words based on its distance from the cluster prototypes. The soft weights derived from these distances are used, in turn, to weight the counts used in the *tf-idf* vectors and to retrieve additional images for later verification.

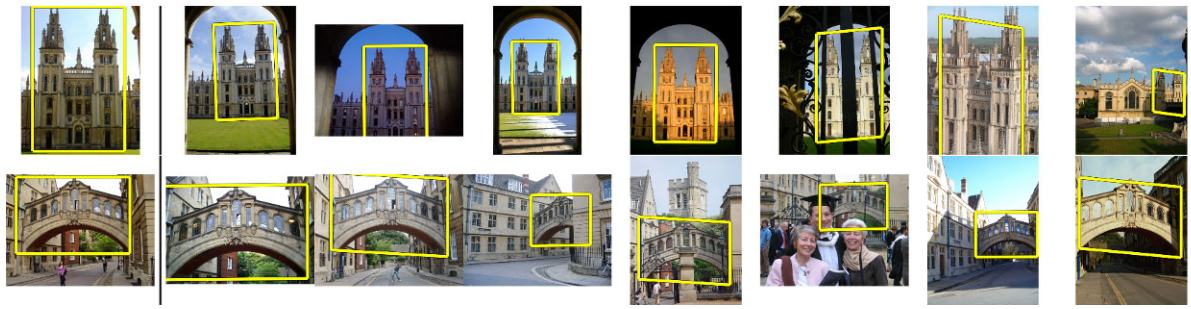


Figure 14.31 Location or building recognition using randomized trees (Philbin, Chum, Isard *et al.* 2007) © 2007 IEEE. The left image is the query, the other images are the highest-ranked results.

Taken together, these recent advances hold the promise of extending current instance recognition algorithms to performing Web-scale retrieval and matching tasks (Agarwal, Snavely, Simon *et al.* 2009; Agarwal, Furukawa, Snavely *et al.* 2010; Snavely, Simon, Goesele *et al.* 2010).

14.3.3 *Application*: Location recognition

One of the most exciting applications of instance recognition today is in the area of location recognition, which can be used both in desktop applications (where did I take this holiday snap?) and in mobile (cell-phone) applications. The latter case includes not only finding out your current location based on a cell-phone image but also providing you with navigation directions or annotating your images with useful information, such as building names and restaurant reviews (i.e., a portable form of *augmented reality*).

Some approaches to location recognition assume that the photos consist of architectural scenes for which vanishing directions can be used to pre-rectify the images for easier matching (Robertson and Cipolla 2004). Other approaches use general affine covariant interest points to perform *wide baseline matching* (Schaffalitzky and Zisserman 2002). The Photo Tourism system of Snavely, Seitz, and Szeliski (2006) (Section 13.1.2) was the first to apply these kinds of ideas to large-scale image matching and (implicit) location recognition from Internet photo collections taken under a wide variety of viewing conditions.

The main difficulty in location recognition is in dealing with the extremely large community (user-generated) photo collections on Web sites such as Flickr (Philbin, Chum, Isard *et al.* 2007; Chum, Philbin, Sivic *et al.* 2007; Philbin, Chum, Sivic *et al.* 2008; Turcot and Lowe 2009) or commercially captured databases (Schindler, Brown, and Szeliski 2007). The prevalence of commonly appearing elements such as foliage, signs, and common architectural elements further complicates the task. Figure 14.31 shows some results on location recognition from community photo collections, while Figure 14.32 shows sample results from denser commercially acquired datasets. In the latter case, the overlap between adjacent database images can be used to verify and prune potential matches using "temporal" filtering, i.e., requiring the query image to match nearby overlapping database images before accepting the match.

Another variant on location recognition is the automatic discovery of *landmarks*, i.e.,



(a) (b) (c)

Figure 14.32 Feature-based location recognition (Schindler, Brown, and Szeliski 2007) © 2007 IEEE: (a) three typical series of overlapping street photos; (b) handheld camera shots and (c) their corresponding database photos.

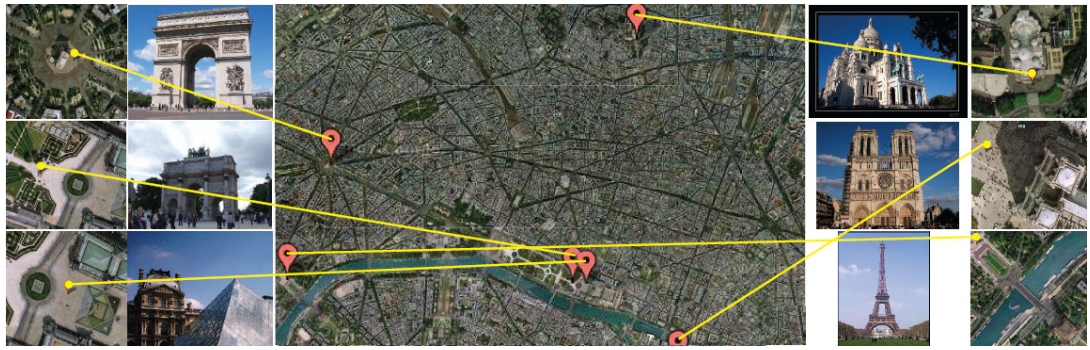
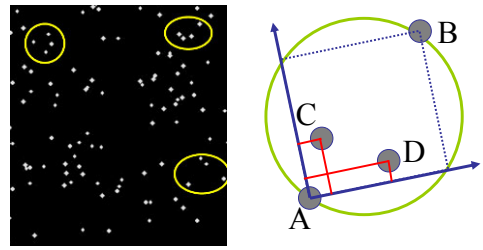


Figure 14.33 Automatic mining, annotation, and localization of community photo collections (Quack, Leibe, and Van Gool 2008) © 2008 ACM. This figure does not show the textual annotations or corresponding Wikipedia entries, which are also discovered.

frequently photographed objects and locations. Simon, Snavely, and Seitz (2007) show how these kinds of objects can be discovered simply by analyzing the matching graph constructed as part of the 3D modeling process in Photo Tourism. More recent work has extended this approach to larger data sets using efficient clustering techniques (Philbin and Zisserman 2008; Li, Wu, Zach *et al.* 2008; Chum, Philbin, and Zisserman 2008; Chum and Matas 2010) as well as combining meta-data such as GPS and textual tags with visual search (Quack, Leibe, and Van Gool 2008; Crandall, Backstrom, Huttenlocher *et al.* 2009), as shown in Figure 14.33. It is now even possible to automatically associate object tags with images based on their co-occurrence in multiple loosely tagged images (Simon and Seitz 2008; Gammeter, Bossard, Quack *et al.* 2009).

The concept of organizing the world's photo collections by location has even been recently extended to organizing all of the universe's (astronomical) photos in an application called *astrometry*, http://astrometry.net/. The technique used to match any two star fields is



(a) (b)

Figure 14.34 Locating star fields using astrometry, http://astrometry.net/. (a) Input star field and some selected star quads. (b) The 2D coordinates of stars C and D are encoded relative to the unit square defined by A and B.

to take quadruplets of nearby stars (a pair of stars and another pair inside their diameter) to form a 30-bit *geometric hash* by encoding the relative positions of the second pair of points using the inscribed square as the reference frame, as shown in Figure 14.34. Traditional information retrieval techniques (k-d trees built for different parts of a sky atlas) are then used to find matching quads as potential star field location hypotheses, which can then be verified using a similarity transform.

14.4 Category recognition

While instance recognition techniques are relatively mature and are used in commercial applications, such as Photosynth (Section 13.1.2), generic category (class) recognition is still a largely unsolved problem. Consider for example the set of photographs in Figure 14.35, which shows objects taken from 10 different visual categories. (I'll leave it up to you to name each of the categories.) How would you go about writing a program to categorize each of these images into the appropriate class, especially if you were also given the choice "none of the above"?

As you can tell from this example, visual category recognition is an *extremely* challenging problem; no one has yet constructed a system that approaches the performance level of a two-year-old child. However, the progress in the field has been quite dramatic, if judged by how much better today's algorithms are compared to those of a decade ago.

Figure 14.54 shows a sample image from each of the 20 categories used in the 2008 PASCAL Visual Object Classes Challenge. The yellow boxes represent the extent of each of the objects found in a given image. On such *closed world* collections where the task is to decide among 20 categories, today's classification algorithms can do remarkably well.

In this section, we look at a number of approaches to solving category recognition. While historically, *part-based* representations and recognition algorithms (Section 14.4.2) were the preferred approach (Fischler and Elschlager 1973; Felzenszwalb and Huttenlocher 2005; Fergus, Perona, and Zisserman 2007), we begin by describing simpler *bag-of-features* approaches (Section 14.4.1) that represent objects and images as unordered collections of feature descriptors. We then look at the problem of simultaneously segmenting images while recognizing objects (Section 14.4.3) and also present some applications of such techniques to photo manipulation (Section 14.4.4). In Section 14.5, we look at how context and scene un-

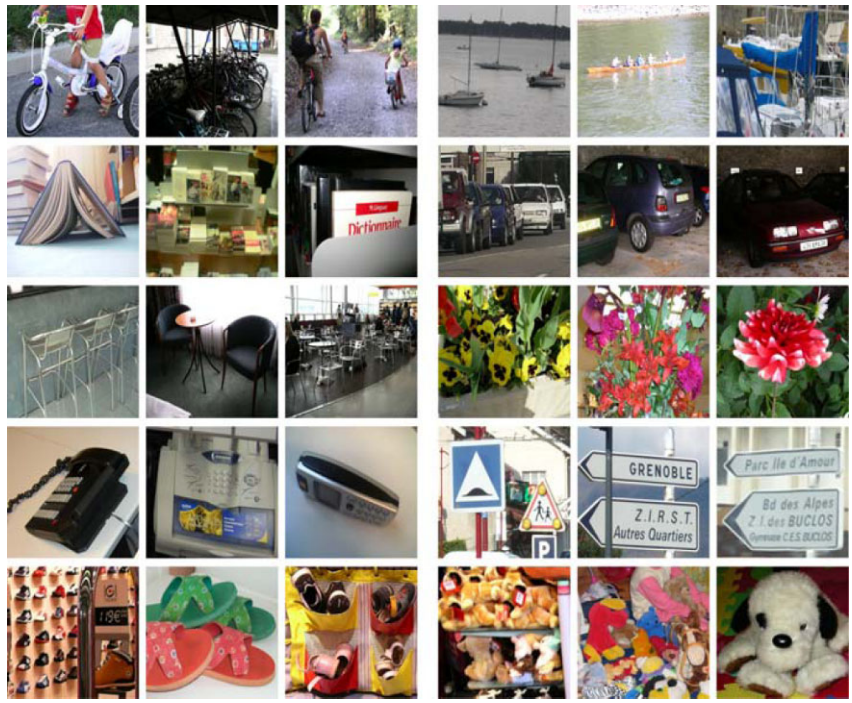


Figure 14.35 Sample images from the Xerox 10 class dataset (Csurka, Dance, Perronnin *et al.* 2006) © 2007 Springer. Imagine trying to write a program to distinguish such images from other photographs.

derstanding, as well as machine learning, can improve overall recognition results. Additional details on the techniques presented in this section can be found in (Pinz 2005; Ponce, Hebert, Schmid *et al.* 2006; Dickinson, Leonardis, Schiele *et al.* 2007; Fei-Fei, Fergus, and Torralba 2009).

14.4.1 Bag of words

One of the simplest algorithms for category recognition is the *bag of words* (also known as *bag of features* or *bag of keypoints*) approach (Csurka, Dance, Fan *et al.* 2004; Lazebnik, Schmid, and Ponce 2006; Csurka, Dance, Perronnin *et al.* 2006; Zhang, Marszalek, Lazebnik *et al.* 2007). As shown in Figure 14.36, this algorithm simply computes the distribution (histogram) of visual words found in the query image and compares this distribution to those found in the training images. We have already seen elements of this approach in Section 14.3.2, Equations (14.33–14.35) and Algorithm 14.2. The biggest difference from instance recognition is the absence of a geometric verification stage (Section 14.3.1), since individual instances of generic visual categories, such as those shown in Figure 14.35, have relatively little spatial coherence to their features (but see the work by Lazebnik, Schmid, and Ponce (2006)).

Csurka, Dance, Fan *et al.* (2004) were the first to use the term *bag of keypoints* to describe such approaches and among the first to demonstrate the utility of frequency-based techniques for category recognition. Their original system used affine covariant regions and SIFT de-

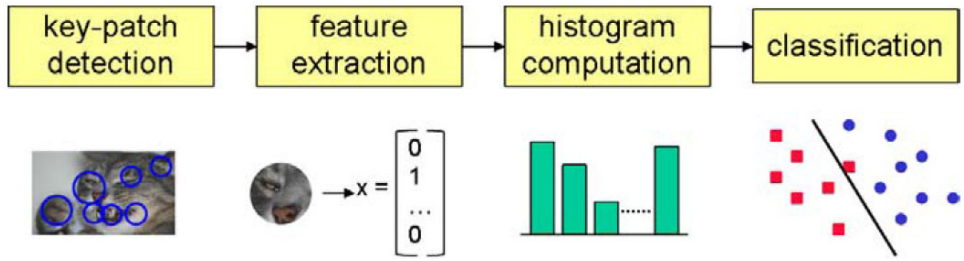


Figure 14.36 A typical processing pipeline for a bag-of-words category recognition system (Csurka, Dance, Perronnin *et al.* 2006) © 2007 Springer. Features are first extracted at keypoints and then quantized to get a distribution (histogram) over the learned *visual words* (feature cluster centers). The feature distribution histogram is used to learn a decision surface using a classification algorithm, such as a support vector machine.

scriptors, k-means visual vocabulary construction, and both a naïve Bayesian classifier and support vector machines for classification. (The latter was found to perform better.) Their newer system (Csurka, Dance, Perronnin *et al.* 2006) uses regular (non-affine) SIFT patches, boosting instead of SVMs, and incorporates a small amount of geometric consistency information.

Zhang, Marszalek, Lazebnik *et al.* (2007) perform a more detailed study of such bag of features systems. They compare a number of feature detectors (Harris–Laplace (Mikolajczyk and Schmid 2004) and Laplacian (Lindeberg 1998b)), descriptors (SIFT, RIFT, and SPIN (Lazebnik, Schmid, and Ponce 2005)), and SVM kernel functions. To estimate distances for the kernel function, they form an *image signature*

$$S = ((t_1, \boldsymbol{m}_1), \ldots, (t_m, \boldsymbol{m}_m)), \tag{14.36}$$

analogous to the *tf-idf* vector $\boldsymbol{t}$ in (14.34), where the cluster centers $\boldsymbol{m}_i$ are made explicit. They then investigate two different kernels for comparing such image signatures. The first is the *earth mover's distance* (EMD) (Rubner, Tomasi, and Guibas 2000),

$$EMD(S, S') = \frac{\sum_i \sum_j f_{ij} d(\boldsymbol{m}_i, \boldsymbol{m}'_j)}{\sum_i \sum_j f_{ij}}, \tag{14.37}$$

where f_{ij} is a *flow* value that can be computed using a linear program and $d(\boldsymbol{m}_i, \boldsymbol{m}'_j)$ is the *ground distance* (Euclidean distance) between $\boldsymbol{m}_i$ and $\boldsymbol{m}'_j$. Note that the EMD can be used to compare two signatures of different lengths, where the entries do not need to correspond. The second is a χ^2 distance

$$\chi^2(S, S') = \frac{1}{2} \sum_i \frac{(t_i - t'_i)^2}{t_i + t'_i}, \tag{14.38}$$

which measures the likelihood that the two signatures were generated from consistent random processes. These distance metrics are then converted into SVM kernels using a generalized Gaussian kernel

$$K(S, S') = \exp\left(-\frac{1}{A} D(S, S')\right), \tag{14.39}$$

where A is a scaling parameter set to the mean distance between training images. In their experiments, they find that the EMD works best for visual category recognition and the χ^2 measure is best for texture recognition.

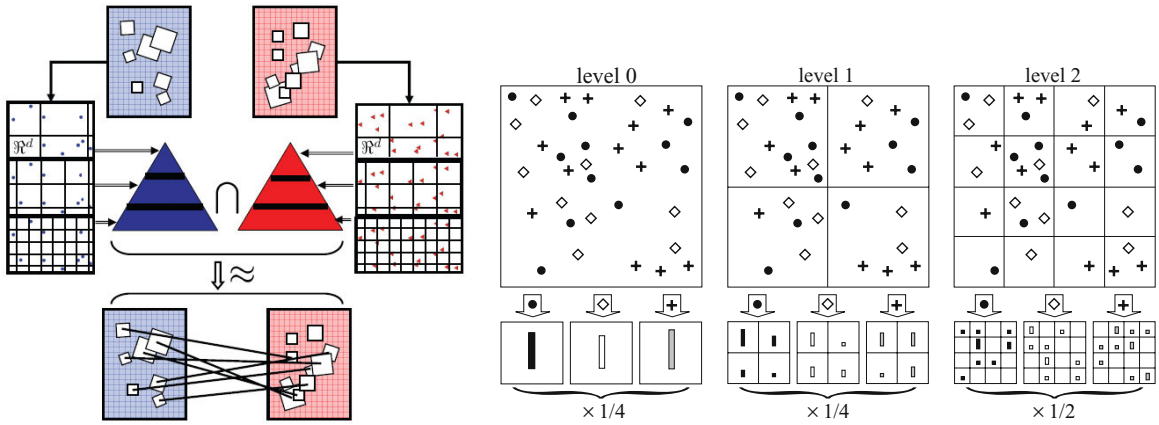


Figure 14.37 Comparing collections of feature vectors using pyramid matching. (a) The feature-space pyramid match kernel (Grauman and Darrell 2007b) constructs a pyramid in high-dimensional feature space and uses it to compute distances (and implicit correspondences) between sets of feature vectors. (b) Spatial pyramid matching (Lazebnik, Schmid, and Ponce 2006) © 2006 IEEE divides the image into a pyramid of pooling regions and computes separate visual word histograms (distributions) inside each spatial bin.

Instead of quantizing feature vectors to visual words, Grauman and Darrell (2007b) develop a technique for directly computing an approximate distance between two variably sized collections of feature vectors. Their approach is to bin the feature vectors into a multi-resolution pyramid defined in feature space (Figure 14.37a) and count the number of features that land in corresponding bins B_{il} and B'_{il} (Figure 14.38a–c). The distance between the two sets of feature vectors (which can be thought of as points in a high-dimensional space) is computed using histogram intersection between corresponding bins

$$C_l = \sum_i \min(B_{il}, B'_{il}) \tag{14.40}$$

(Figure 14.38d). These per-level counts are then summed up in a weighted fashion

$$D_\Delta = \sum_l w_l N_l \quad \text{with} \quad N_l = C_l - C_{l-1} \quad \text{and} \quad w_l = \frac{1}{d2^l} \tag{14.41}$$

(Figure 14.38e), which discounts matches already found at finer levels while weighting finer matches more heavily. (d is the dimension of the embedding space, i.e., the length of the feature vectors.) In follow-on work, Grauman and Darrell (2007a) show how an explicit construction of the pyramid can be avoided using hashing techniques.

Inspired by this work, Lazebnik, Schmid, and Ponce (2006) show how a similar idea can be employed to augment bags of keypoints with loose notions of 2D spatial location analogous to the pooling performed by SIFT (Lowe 2004) and "gist" (Torralba, Murphy, Freeman *et al.* 2003). In their work, they extract affine region descriptors (Lazebnik, Schmid, and Ponce 2005) and quantize them into visual words. (Based on previous results by Fei-Fei and Perona (2005), the feature descriptors are extracted densely (on a regular grid) over the image, which can be helpful in describing textureless regions such as the sky.) They then form

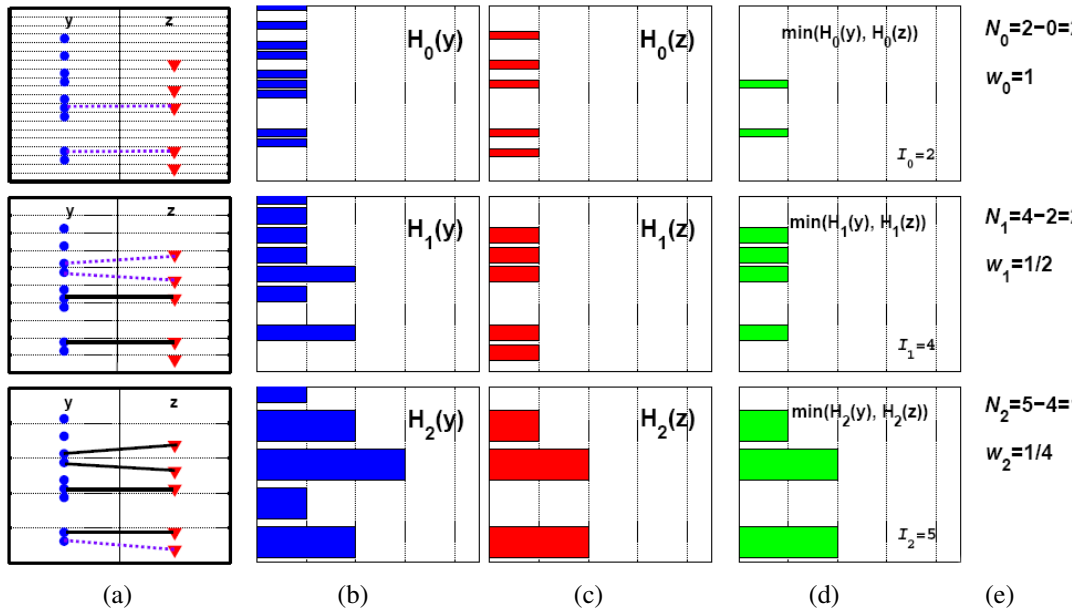


Figure 14.38 A one-dimensional illustration of comparing collections of feature vectors using the pyramid match kernel (Grauman and Darrell 2007b): (a) distribution of feature vectors (point sets) into the pyramidal bins; (b–c) histogram of point counts in bins B_{il} and B'_{il} for the two images; (d) histogram intersections (minimum values); (e) per-level similarity scores, which are weighted and summed to form the final distance/similarity metric.

a spatial pyramid of bins containing word counts (histograms), as shown in Figure 14.37b, and use a similar pyramid match kernel to combine histogram intersection counts in a hierarchical fashion.

The debate about whether to use quantized feature descriptors or continuous descriptors and also whether to use sparse or dense features continues to this day. Boiman, Shechtman, and Irani (2008) show that if query images are compared to *all* the features representing a given class, rather than just each class image individually, nearest-neighbor matching followed by a naïve Bayes classifier outperforms quantized visual words (Figure 14.39). Instead of using generic feature detectors and descriptors, some authors have been investigating *learning* class-specific features (Ferencz, Learned-Miller, and Malik 2008), often using randomized forests (Philbin, Chum, Isard *et al.* 2007; Moosmann, Nowak, and Jurie 2008; Shotton, Johnson, and Cipolla 2008) or combining the feature generation and image classification stages (Yang, Jin, Sukthankar *et al.* 2008). Others, such as Serre, Wolf, and Poggio (2005) and Mutch and Lowe (2008) use hierarchies of dense feature transforms inspired by biological (visual cortical) processing combined with SVMs for final classification.

14.4.2 Part-based models

Recognizing an object by finding its constituent parts and measuring their geometric relationships is one of the oldest approaches to object recognition (Fischler and Elschlager 1973; Kanade 1977; Yuille 1991). We have already seen examples of part-based approaches being used for face recognition (Figure 14.18) (Moghaddam and Pentland 1997; Heisele, Ho, Wu

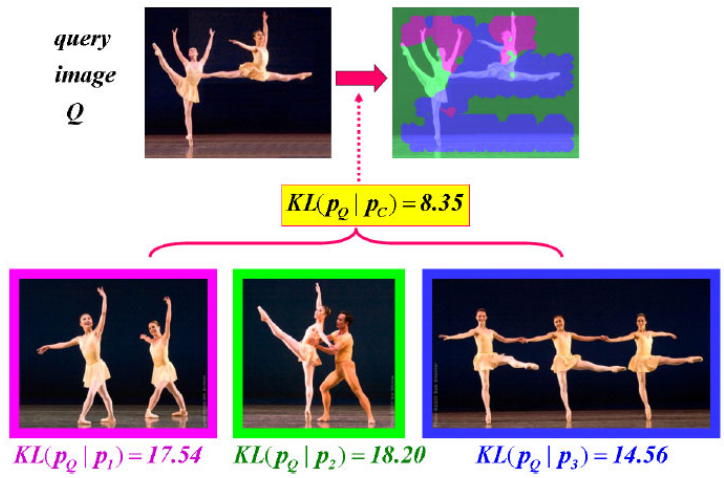


Figure 14.39 "Image-to-Image" vs. "Image-to-Class" distance comparison (Boiman, Shechtman, and Irani 2008) © 2008 IEEE. The query image on the upper left may not match the feature distribution of any of the database images in the bottom row. However, if each feature in the query is matched to its closest analog in *all* the class images, a good match can be found.

et al. 2003; Heisele, Serre, and Poggio 2007) and pedestrian detection (Figure 14.9) (Felzenszwalb, McAllester, and Ramanan 2008).

In this section, we look more closely at some of the central issues in part-based recognition, namely, the representation of geometric relationships, the representation of individual parts, and algorithms for learning such descriptions and recognizing them at run time. More details on part-based models for recognition can be found in the course notes of Fergus (2007b, 2009).

The earliest approaches to representing geometric relationships were dubbed *pictorial structures* by Fischler and Elschlager (1973) and consisted of spring-like connections between different feature locations (Figure 14.1a). To fit a pictorial structure to an image, an energy function of the form

$$E = \sum_i V_i(\boldsymbol{l}_i) + \sum_{ij \in E} V_{ij}(\boldsymbol{l}_i, \boldsymbol{l}_j) \tag{14.42}$$

is minimized over all potential part locations or poses $\{\boldsymbol{l}_i\}$ and pairs of parts (i, j) for which an edge (geometric relationship) exists in E. Note how this energy is closely related to that used with Markov random fields (3.108–3.109), which can be used to embed pictorial structures in a probabilistic framework that makes parameter learning easier (Felzenszwalb and Huttenlocher 2005).

Part-based models can have different topologies for the geometric connections between the parts (Figure 14.41). For example, Felzenszwalb and Huttenlocher (2005) restrict the connections to a tree (Figure 14.41d), which makes learning and inference more tractable. A tree topology enables the use of a recursive Viterbi (dynamic programming) algorithm (Pearl 1988; Bishop 2006), in which leaf nodes are first optimized as a function of their parents, and the resulting values are then plugged in and eliminated from the energy function—see Appendix B.5.2. The Viterbi algorithm computes an optimal match in $O(N^2|E| + NP)$ time,

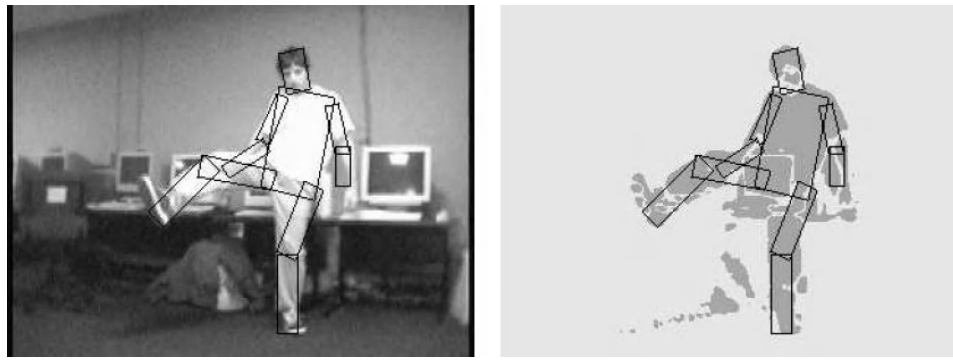


Figure 14.40 Using pictorial structures to locate and track a person (Felzenszwalb and Huttenlocher 2005) © 2005 Springer. The structure consists of articulated rectangular body parts (torso, head, and limbs) connected in a tree topology that encodes relative part positions and orientations. To fit a pictorial structure model, a binary silhouette image is first computed using background subtraction.

where N is the number of potential locations or poses for each part, $|E|$ is the number of edges (pairwise constraints), and $P = |V|$ is the number of parts (vertices in the graphical model, which is equal to $|E| + 1$ in a tree). To further increase the efficiency of the inference algorithm, Felzenszwalb and Huttenlocher (2005) restrict the pairwise energy functions $V_{ij}(\boldsymbol{l}_i, \boldsymbol{l}_j)$ to be Mahalanobis distances on functions of location variables and then use fast distance transform algorithms to minimize each pairwise interaction in time that is closer to linear in N.

Figure 14.40 shows the results of using their pictorial structures algorithm to fit an articulated body model to a binary image obtained by background segmentation. In this application of pictorial structures, parts are parameterized by the locations, sizes, and orientations of their approximating rectangles. Unary matching potentials $V_i(\boldsymbol{l}_i)$ are determined by counting the percentage of foreground and background pixels inside and just outside the tilted rectangle representing each part.

Over the last decade, a large number of different graphical models have been proposed for part-based recognition, as shown in Figure 14.41. Carneiro and Lowe (2006) discuss a number of these models and propose one of their own, which they call a *sparse flexible model*; it involves ordering the parts and having each part's location depend on at most k of its ancestor locations.

The simplest models, which we saw in Section 14.4.1, are bags of words, where there are no geometric relationships between different parts or features. While such models can be very efficient, they have a very limited capacity to express the spatial arrangement of parts. Trees and stars (a special case of trees where all leaf nodes are directly connected to a common root) are the most efficient in terms of inference and hence also learning (Felzenszwalb and Huttenlocher 2005; Fergus, Perona, and Zisserman 2005; Felzenszwalb, McAllester, and Ramanan 2008). Directed acyclic graphs (Figure 14.41f–g) come next in terms of complexity and can still support efficient inference, although at the cost of imposing a causal structure on the part model (Bouchard and Triggs 2005; Carneiro and Lowe 2006). k-fans, in which a clique of size k forms the root of a star-shaped model (Figure 14.41c) have inference complexity $O(N^{k+1})$, although with distance transforms and Gaussian priors, this can be lowered to

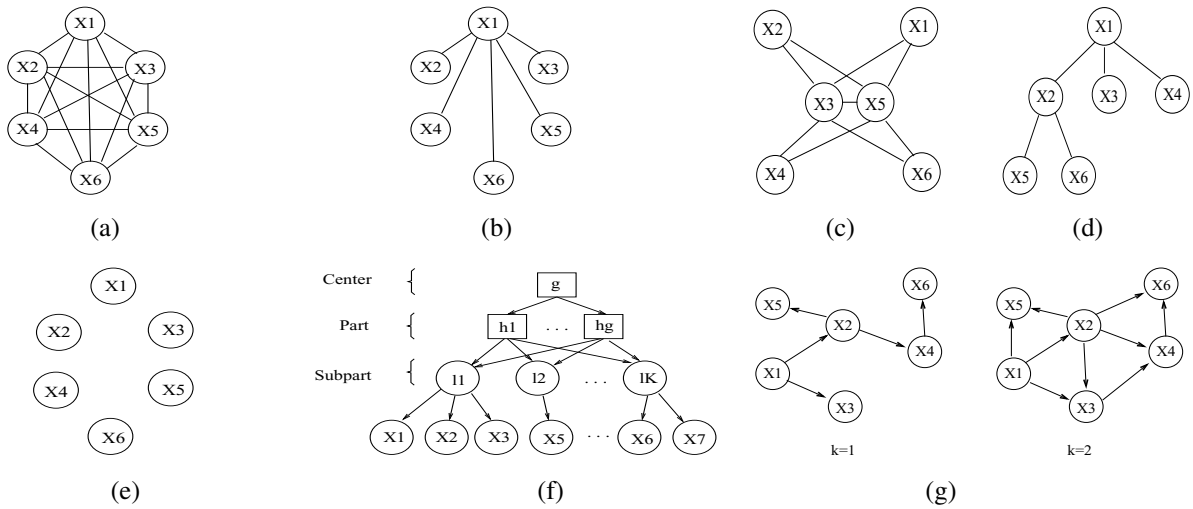


Figure 14.41 Graphical models for geometric spatial priors (Carneiro and Lowe 2006) © 2006 Springer: (a) constellation (Fergus, Perona, and Zisserman 2007); (b) star (Crandall, Felzenszwalb, and Huttenlocher 2005; Fergus, Perona, and Zisserman 2005); (c) k-fan ($k = 2$) (Crandall, Felzenszwalb, and Huttenlocher 2005); (d) tree (Felzenszwalb and Huttenlocher 2005); (e) bag of features (Csurka, Dance, Fan *et al.* 2004); (f) hierarchy (Bouchard and Triggs 2005); (g) sparse flexible model (Carneiro and Lowe 2006).

$O(N^k)$ (Crandall, Felzenszwalb, and Huttenlocher 2005; Crandall and Huttenlocher 2006). Finally, fully connected *constellation* models (Figure 14.41a) are the most general, but the assignment of features to parts becomes intractable for moderate numbers of parts P, since the complexity of such an assignment is $O(N^P)$ (Fergus, Perona, and Zisserman 2007).

The original constellation model was developed by Burl, Weber, and Perona (1998) and consists of a number of parts whose relative positions are encoded by their mean locations and a full covariance matrix, which is used to denote not only positional uncertainty but also potential correlations (covariance) between different parts (Figure 14.42a). Weber, Welling, and Perona (2000) extended this technique to a weakly supervised setting, where both the appearance of each part and its locations are automatically learned given only whole image labels. Fergus, Perona, and Zisserman (2007) further extend this approach to simultaneous learning of appearance and shape models from scale-invariant keypoint detections.

Figure 14.42a shows the shape model learned for the motorcycle class. The top figure shows the mean relative locations for each part along with their position covariances (inter-part covariances are not shown) and likelihood of occurrence. The bottom curve shows the Gaussian PDFs for the relative log-scale of each part with respect to the "landmark" feature. Figure 14.42b shows the appearance model learned for each part, visualized as the patches around detected features in the training database that best match the appearance model. Figure 14.42c shows the features detected in the test database (pink dots) along with the corresponding parts that they were assigned to (colored circles). As you can see, the system has successfully learned and then used a fairly complex model of motorcycle appearance.

The part-based approach to recognition has also been extended to learning new categories from small numbers of examples, building on recognition components developed for other classes (Fei-Fei, Fergus, and Perona 2006). More complex hierarchical part-based models can

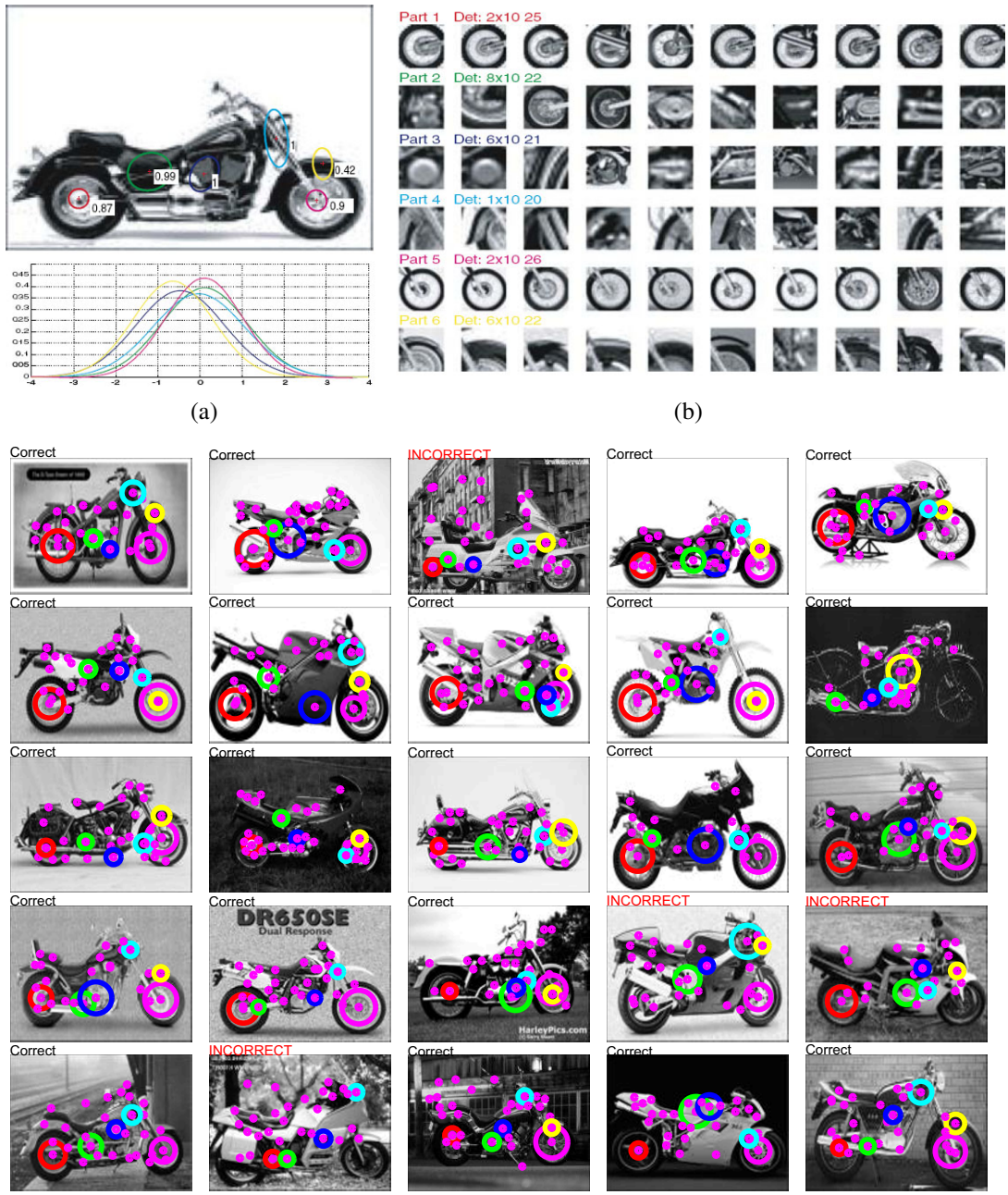


Figure 14.42 Part-based recognition (Fergus, Perona, and Zisserman 2007) © 2007 Springer: (a) locations and covariance ellipses for each part, along with their occurrence probabilities (top) and relative log-scale densities (bottom); (b) part examples drawn from the training images that best match the average appearance; (c) recognition results for the motorcycle class, showing detected features (pink dots) and parts (colored circles).

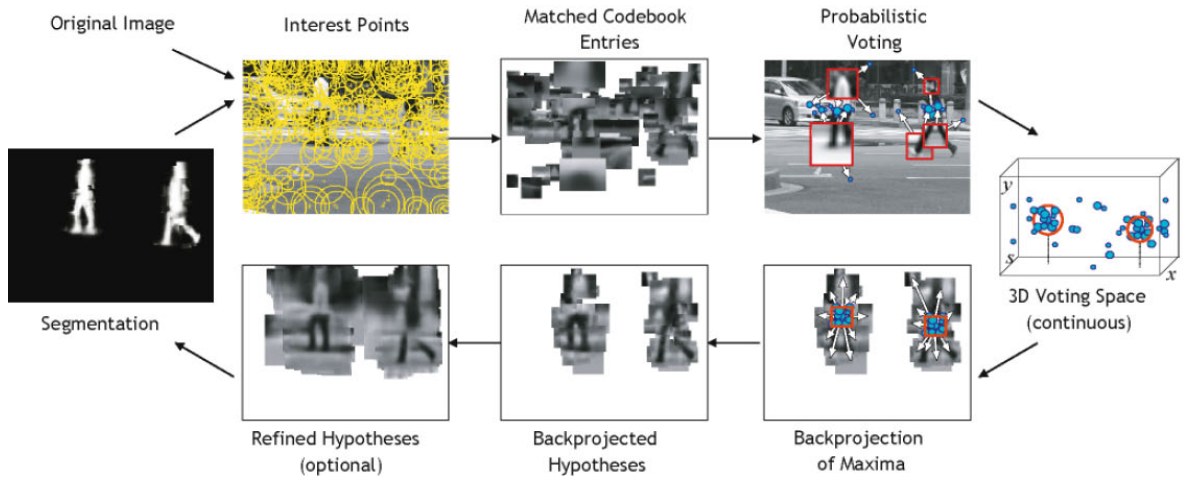


Figure 14.43 Interleaved recognition and segmentation (Leibe, Leonardis, and Schiele 2008) © 2008 Springer. The process starts by re-recognizing visual words (codebook entries) in a new image (scene) and having each part vote for likely locations and size in a 3D (x, y, s) voting space (top row). Once a maximum has been found, the parts (features) corresponding to this instance are determined by *backprojecting* the contributing votes. The foreground–background segmentation for each object can be found by backprojecting probabilistic masks associated with each codebook entry. The whole recognition and segmentation process can then be repeated.

be developed using the concept of grammars (Bouchard and Triggs 2005; Zhu and Mumford 2006). A simpler way to use parts is to have keypoints that are recognized as being part of a class vote for the estimated part locations, as shown in the top row of Figure 14.43 (Leibe, Leonardis, and Schiele 2008). (Implicitly, this corresponds to having a star-shaped geometric model.)

14.4.3 Recognition with segmentation

The most challenging version of generic object recognition is to simultaneously perform recognition with accurate boundary segmentation (Fergus 2007a). For instance recognition (Section 14.3.1), this can sometimes be achieved by backprojecting the object model into the scene (Lowe 2004), as shown in Figure 14.1d, or matching portions of the new scene to pre-learned (segmented) object models (Ferrari, Tuytelaars, and Van Gool 2006b; Kannala, Rahtu, Brandt *et al.* 2008).

For more complex (flexible) object models, such as those for humans Figure 14.1f, a different approach is to pre-segment the image into larger or smaller pieces (Chapter 5) and then match such pieces to portions of the model (Mori, Ren, Efros *et al.* 2004; Mori 2005; He, Zemel, and Ray 2006; Gu, Lim, Arbelaez *et al.* 2009).

An alternative approach by Leibe, Leonardis, and Schiele (2008), which we introduced in the previous section, votes for potential object locations and scales based on the detection of features corresponding to pre-clustered visual codebook entries (Figure 14.43). To support segmentation, each codebook entry has an associated foreground–background mask, which is learned as part of the codebook clustering process from pre-labeled object segmentation masks. During recognition, once a maximum in the voting space is found, the masks

associated with the entries that voted for this instance are combined to obtain an object segmentation, as shown on the left side of Figure 14.43.

A more holistic approach to recognition and segmentation is to formulate the problem as one of labeling every pixel in an image with its class membership, and to solve this problem using energy minimization or Bayesian inference techniques, i.e., conditional random fields (Section 3.7.2, (3.118)) (Kumar and Hebert 2006; He, Zemel, and Carreira-Perpiñán 2004). The TextonBoost system of Shotton, Winn, Rother *et al.* (2009) uses unary (pixel-wise) potentials based on image-specific color distributions (Section 5.5) (Boykov and Jolly 2001; Rother, Kolmogorov, and Blake 2004), location information (e.g., foreground objects are more likely to be in the middle of the image, sky is likely to be higher, and road is likely to be lower), and novel texture-layout classifiers trained using shared boosting. It also uses traditional pairwise potentials that look at image color gradients (Veksler 2001; Boykov and Jolly 2001; Rother, Kolmogorov, and Blake 2004). The texton-layout features first filter the image with a series of 17 oriented filter banks and then cluster the responses to classify each pixel into 30 different texton classes (Malik, Belongie, Leung *et al.* 2001). The responses are then filtered using offset rectangular regions trained with joint boosting (Viola and Jones 2004) to produce the texton-layout features used as unary potentials.

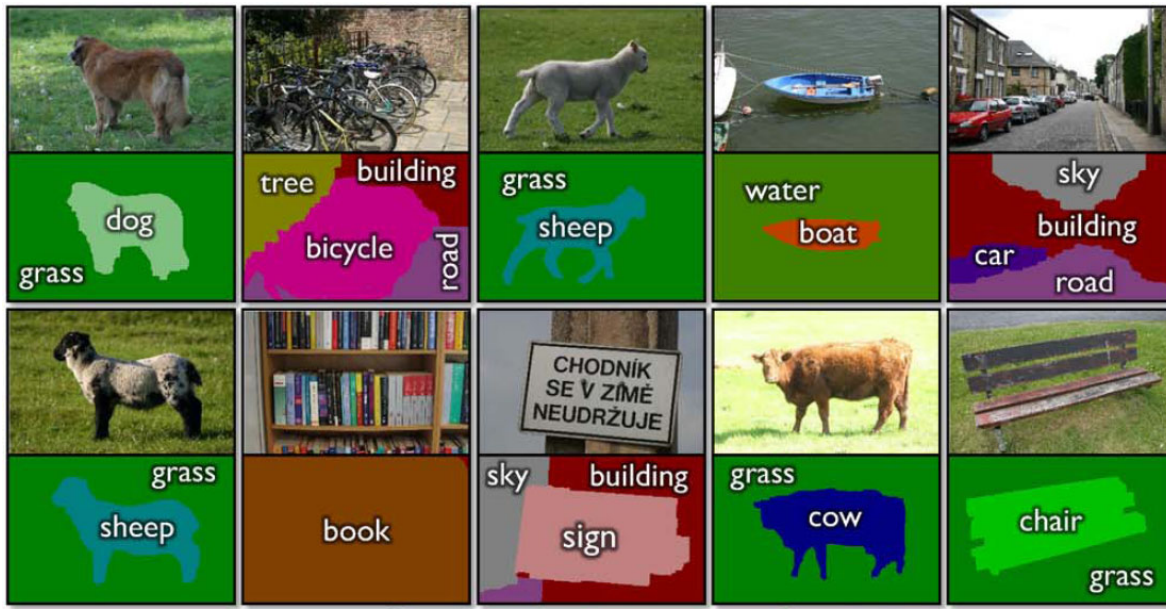
Figure 14.44a shows some examples of images successfully labeled and segmented using TextonBoost, while Figure 14.44b shows examples where it does not do as well. As you can see, this kind of semantic labeling can be extremely challenging.

The TextonBoost conditional random field framework has been extended to LayoutCRFs by Winn and Shotton (2006), who incorporate additional constraints to recognize multiple object instances and deal with occlusions (Figure 14.45), and even more recently by Hoiem, Rother, and Winn (2007) to incorporate full 3D models.

Conditional random fields continue to be widely used and extended for simultaneous recognition and segmentation applications (Kumar and Hebert 2006; He, Zemel, and Ray 2006; Levin and Weiss 2006; Verbeek and Triggs 2007; Yang, Meer, and Foran 2007; Rabinovich, Vedaldi, Galleguillos *et al.* 2007; Batra, Sukthankar, and Chen 2008; Larlus and Jurie 2008; He and Zemel 2008; Kumar, Torr, and Zisserman 2010), producing some of the best results on the difficult PASCAL VOC segmentation challenge (Shotton, Johnson, and Cipolla 2008; Kohli, Ladický, and Torr 2009). Approaches that first segment the image into unique or multiple segmentations (Borenstein and Ullman 2008; He, Zemel, and Ray 2006; Russell, Efros, Sivic *et al.* 2006) (potentially combined with CRF models) also do quite well: Csurka and Perronnin (2008) have one of the top algorithms in the VOC segmentation challenge. Hierarchical (multi-scale) and grammar (parsing) models are also sometimes used (Tu, Chen, Yuille *et al.* 2005; Zhu, Chen, Lin *et al.* 2008).

14.4.4 *Application*: Intelligent photo editing

Recent advances in object recognition and scene understanding have greatly increased the power of intelligent (semi-automated) photo editing applications. One example is the Photo Clip Art system of Lalonde, Hoiem, Efros *et al.* (2007), which recognizes and segments objects of interest, such as pedestrians, in Internet photo collections and then allows users to paste them into their own photos. Another is the scene completion system of Hays and Efros (2007), which tackles the same *inpainting* problem we studied in Section 10.5. Given an image in which we wish to erase and fill in a large section (Figure 14.46a–b), where do you



(a)



(b)

Figure 14.44 Simultaneous recognition and segmentation using TextonBoost (Shotton, Winn, Rother *et al.* 2009) © 2009 Springer: (a) successful recognition results; (b) less successful results.

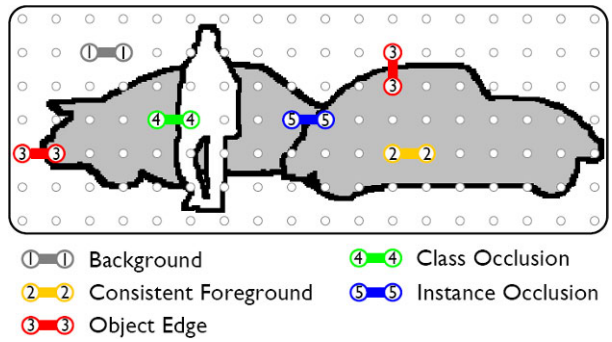


Figure 14.45 Layout consistent random field (Winn and Shotton 2006) © 2006 IEEE. The numbers indicate the kind of neighborhood relations that can exist between pixels assigned to the same or different classes. Each pairwise relationship carries its own likelihood (energy penalty).



(a) (b) (c) (d)

Figure 14.46 Scene completion using millions of photographs (Hays and Efros 2007) © 2007 ACM: (a) original image; (b) after unwanted foreground removal; (c) plausible scene matches, with the one the user selected highlighted in red; (d) output image after replacement and blending.

get the pixels to fill in the gaps in the edited image? Traditional approaches either use smooth continuation (Bertalmio, Sapiro, Caselles *et al.* 2000) or borrowing pixels from other parts of the image (Efros and Leung 1999; Criminisi, Pérez, and Toyama 2004; Efros and Freeman 2001). With the advent of huge repositories of images on the Web (a topic we return to in Section 14.5.1), it often makes more sense to find a *different* image to serve as the source of the missing pixels.

In their system, Hays and Efros (2007) compute the *gist* of each image (Oliva and Torralba 2001; Torralba, Murphy, Freeman *et al.* 2003) to find images with similar colors and composition. They then run a graph cut algorithm that minimizes image gradient differences and composite the new replacement piece into the original image using Poisson image blending (Section 9.3.4) (Pérez, Gangnet, and Blake 2003). Figure 14.46d shows the resulting image with the erased foreground rooftops region replaced with sailboats.

A different application of image recognition and segmentation is to infer 3D structure from a single photo by recognizing certain scene structures. For example, Criminisi, Reid, and Zisserman (2000) detect vanishing points and have the user draw basic structures, such as walls, in order infer the 3D geometry (Section 6.3.3). Hoiem, Efros, and Hebert (2005a) on the other hand, work with more "organic" scenes such as the one shown in Figure 14.47.



(a) (b) (c) (d) (e)

Figure 14.47 Automatic photo pop-up (Hoiem, Efros, and Hebert 2005a) © 2005 ACM: (a) input image; (b) superpixels are grouped into (c) multiple regions; (d) labelings indicating ground (green), vertical (red), and sky (blue); (e) novel view of resulting piecewise-planar 3D model.

Their system uses a variety of classifiers and statistics learned from labeled images to classify each pixel as either ground, vertical, or sky (Figure 14.47d). To do this, they begin by computing superpixels (Figure 14.47b) and then group them into plausible regions that are likely to share similar geometric labels (Figure 14.47c). After all the pixels have been labeled, the boundaries between the vertical and ground pixels can be used to infer 3D lines along which the image can be folded into a "pop-up" (after removing the sky pixels), as shown in Figure 14.47e. In related work, Saxena, Sun, and Ng (2009) develop a system that directly infers the depth and orientation of each pixel instead of using just three geometric class labels.

Face detection and localization can also be used in a variety of photo editing applications (in addition to being used in-camera to provide better focus, exposure, and flash settings). Zanella and Fuentes (2004) use active shape models (Section 14.2.2) to register facial features for creating automated morphs. Rother, Bordeaux, Hamadi *et al.* (2006) use face and sky detection to determine regions of interest in order to decide which pieces from a collection of images to stitch into a collage. Bitouk, Kumar, Dhillon *et al.* (2008) describe a system that matches a given face image to a large collection of Internet face images, which can then be used (with careful relighting algorithms) to replace the face in the original image. Applications they describe include de-identification and getting the best possible smile from everyone in a "burst mode" group shot. Leyvand, Cohen-Or, Dror *et al.* (2008) show how accurately locating facial features using an active shape model (Cootes, Edwards, and Taylor 2001; Zhou, Gu, and Zhang 2003) can be used to warp such features (and hence the image) towards configurations resembling those found in images whose facial attractiveness was highly rated, thereby "beautifying" the image without completely losing a person's identity.

Most of these techniques rely either on a set of labeled training images, which is an essential component of all learning techniques, or the even more recent explosion in images available on the Internet. The assumption in some of this work (and in recognition systems based on such very large databases (Section 14.5.1)) is that as the collection of accessible (and potentially partially labeled) images gets larger, finding a close match gets easier. As Hays and Efros (2007) state in their abstract "Our chief insight is that while the space of images is effectively infinite, the space of semantically differentiable scenes is actually not that large." In an interesting commentary on their paper, Levoy (2008) disputes this assertion, claiming that "features in natural scenes form a heavy-tailed distribution, meaning that while some features in photographs are more common than others, the relative occurrence of less common features drops slowly. In other words, there are many unusual photographs in the world." He does, however agree that in computational photography, as in many other applications such



(a) (b) (c) (d) (e)

Figure 14.48 The importance of context (images courtesy of Antonio Torralba). Can you name all of the objects in images (a–b), especially those that are circled in (c–d). Look carefully at the circled objects. Did you notice that they all have the same shape (after being rotated), as shown in column (e)?

as speech recognition, synthesis, and translation, "simple machine learning algorithms often outperform more sophisticated ones if trained on large enough databases." He also goes on to point out both the potential advantages of such systems, such as better automatic color balancing, and potential issues and pitfalls with the kind of image fakery that these new approaches enable.

For additional examples of photo editing and computational photography applications enabled by Internet computer vision, please see recent workshops on this topic,[19] as well as the special journal issue (Avidan, Baker, and Shan 2010), and the course on Internet Vision by Tamara Berg (2008).

14.5 Context and scene understanding

Thus far, we have mostly considered the task of recognizing and localizing objects in isolation from that of understanding the scene (context) in which the object occur. This is a severe limitation, as context plays a very important role in human object recognition (Oliva and Torralba 2007). As we will see in this section, it can greatly improve the performance of object recognition algorithms (Divvala, Hoiem, Hays *et al.* 2009), as well as providing useful semantic clues for general scene understanding (Torralba 2008).

Consider the two photographs in Figure 14.48a–b. Can you name all of the objects, especially those circled in images (c–d)? Now have a closer look at the circled objects. Do see any similarity in their shapes? In fact, if you rotate them by 90°, they are all the same as the "blob" shown in Figure 14.48e. So much for our ability to recognize object by their shape! Another (perhaps more artificial) example of recognition in context is shown in Figure 14.49. Try to name all of the letters and numbers, and then see if you guessed right.

Even though we have not addressed context explicitly earlier in this chapter, we have already seen several instances of this general idea being used. A simple way to incorporate spatial information into a recognition algorithm is to compute feature statistics over different regions, as in the spatial pyramid system of Lazebnik, Schmid, and Ponce (2006). Part-based models (Section 14.4.2, Figures 14.40–14.43), use a kind of local context, where various parts need to be arranged in a proper geometric relationship to constitute an object.

[19] http://www.internetvisioner.org/.



Figure 14.49 More examples of context: read the letters in the first group, the numbers in the second, and the letters and numbers in the third. (Images courtesy of Antonio Torralba.)

The biggest difference between part-based and context models is that the latter combine objects into scenes and the number of constituent objects from each class is not known in advance. In fact, it is possible to combine part-based and context models into the same recognition architecture (Murphy, Torralba, and Freeman 2003; Sudderth, Torralba, Freeman *et al.* 2008; Crandall and Huttenlocher 2007).

Consider the street and office scenes shown in Figure 14.50a–b. If we have enough training images with labeled regions, such as buildings, cars, and roads or monitors, keyboards, and mice, we can develop a geometric model for describing their relative positions. Sudderth, Torralba, Freeman *et al.* (2008) develop such a model, which can be thought of as a two-level constellation model. At the top level, the distributions of objects relative to each other (say, buildings with respect to cars) is modeled as a Gaussian (Figure 14.50c, upper right corners). At the bottom level, the distribution of parts (affine covariant features) with respect to the object center is modeled using a mixture of Gaussians (Figure 14.50c, lower two rows). However, since the number of objects in the scene and parts in each object is unknown, a *latent Dirichlet process* (LDP) is used to model object and part creation in a generative framework. The distributions for all of the objects and parts are learned from a large labeled database and then later used during inference (recognition) to label the elements of a scene.

Another example of context is in simultaneous segmentation and recognition (Section 14.4.3) (Figures 14.44–14.45), where the arrangements of various objects in a scene are used as part of the labeling process. Torralba, Murphy, and Freeman (2004) describe a conditional random field where the estimated locations of building and roads influence the detection of cars, and where boosting is used to learn the structure of the CRF. Rabinovich, Vedaldi, Galleguillos *et al.* (2007) use context to improve the results of CRF segmentation by noting that certain adjacencies (relationships) are more likely than others, e.g., a person is more likely to be on a horse than on a dog.

Context also plays an important role in 3D inference from single images (Figure 14.47), using computer vision techniques for labeling pixels as belonging to the ground, vertical surfaces, or sky (Hoiem, Efros, and Hebert 2005a,b). This line of work has been extended to a more holistic approach that simultaneously reasons about object identity, location, surface orientations, occlusions, and camera viewing parameters (Hoiem, Efros, and Hebert 2008a,b).

A number of approaches use the *gist* of a scene (Torralba 2003; Torralba, Murphy, Freeman *et al.* 2003) to determine where instances of particular objects are likely to occur. For example, Murphy, Torralba, and Freeman (2003) train a regressor to predict the vertical loca-

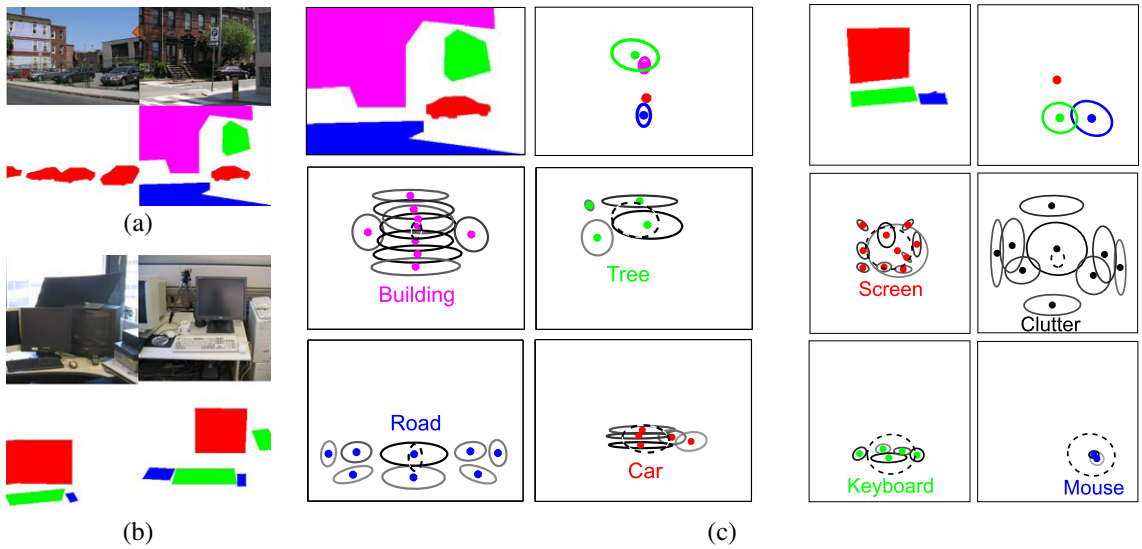


Figure 14.50 Contextual scene models for object recognition (Sudderth, Torralba, Freeman *et al.* 2008) © 2008 Springer: (a) some street scenes and their corresponding labels (magenta = buildings, red = cars, green = trees, blue = road); (b) some office scenes (red = computer screen, green = keyboard, blue = mouse); (c) learned contextual models built from these labeled scenes. The top row shows a sample label image and the distribution of the objects relative to the center red (car or screen) object. The bottom rows show the distributions of parts that make up each object.

tions of objects such as pedestrians, cars, and buildings (or screens and keyboard for indoor office scenes) based on the gist of an image. These location distributions are then used with classic object detectors to improve the performance of the detectors. Gists can also be used to directly match complete images, as we saw in the scene completion work of Hays and Efros (2007).

Finally, some of the most recent work in scene understanding exploits the existence of large numbers of labeled (or even unlabeled) images to perform matching directly against whole images, where the images themselves implicitly encode the expected relationships between objects (Figure 14.51) (Russell, Torralba, Liu *et al.* 2007; Malisiewicz and Efros 2008). We discuss such techniques in the next section, where we look at the influence that large image databases have had on object recognition and scene understanding.

14.5.1 Learning and large image collections

Given how learning techniques are widely used in recognition algorithms, you may wonder whether the topic of learning deserves its own section (or even chapter), or whether it is just part of the basic fabric of all recognition tasks. In fact, trying to build a recognition system without lots of training data for anything other than a basic pattern such as a UPC code has proven to be a dismal failure.

In this chapter, we have already seen lots of techniques borrowed from the machine learning, statistics, and pattern recognition communities. These include principal component, subspace, and discriminant analysis (Section 14.2.1) and more sophisticated discriminative clas-



(a) (b) (c)

Figure 14.51 Recognition by scene alignment (Russell, Torralba, Liu *et al.* 2007): (a) input image; (b) matched images with similar scene configurations; (c) final labeling of the input image.

sification algorithms such as neural networks, support vector machines, and boosting (Section 14.1.1). Some of the best-performing techniques on challenging recognition benchmarks (Varma and Ray 2007; Felzenszwalb, McAllester, and Ramanan 2008; Fritz and Schiele 2008; Vedaldi, Gulshan, Varma *et al.* 2009) rely heavily on the latest machine learning techniques, whose development is often being driven by challenging vision problems (Freeman, Perona, and Schölkopf 2008).

A distinction sometimes made in the recognition community is between problems where most of the variables of interest (say, parts) are already (partially) labeled and systems that learn more of the problem structure with less supervision (Fergus, Perona, and Zisserman 2007; Fei-Fei, Fergus, and Perona 2006). In fact, recent work by Sivic, Russell, Zisserman *et al.* (2008) has demonstrated the ability to learn visual hierarchies (hierarchies of object parts with related visual appearance) and scene segmentations in a totally unsupervised framework.

Perhaps the most dramatic change in the recognition community has been the appearance of very large databases of training images.[20] Early learning-based algorithms, such as those for face and pedestrian detection (Section 14.1), used relatively few (in the hundreds) labeled examples to train recognition algorithm parameters (say, the thresholds used in boosting). Today, some recognition algorithms use databases such as LabelMe (Russell, Torralba, Murphy *et al.* 2008), which contain tens of thousands of labeled examples.

The existence of such large databases opens up the possibility of matching directly against the training images rather than using them to learn the parameters of recognition algorithms. Russell, Torralba, Liu *et al.* (2007) describe a system where a new image is matched against each of the training images, from which a consensus labeling for the unknown objects in the scene can be inferred, as shown in Figure 14.51. Malisiewicz and Efros (2008) start by over-segmenting each image and then use the LabelMe database to search for similar images and configurations in order to obtain per-pixel category labelings. It is also possible to combine feature-based correspondence algorithms with large labeled databases to perform simultaneous recognition and segmentation (Liu, Yuen, and Torralba 2009).

When the database of images becomes large enough, it is even possible to directly match complete images with the expectation of finding a good match. Torralba, Freeman, and Fergus (2008) start with a database of 80 million tiny (32×32) images and compensate for the poor accuracy in their image labels, which are collected automatically from the Internet, by using

[20] We have already seen some computational photography applications of such databases in Section 14.4.4.

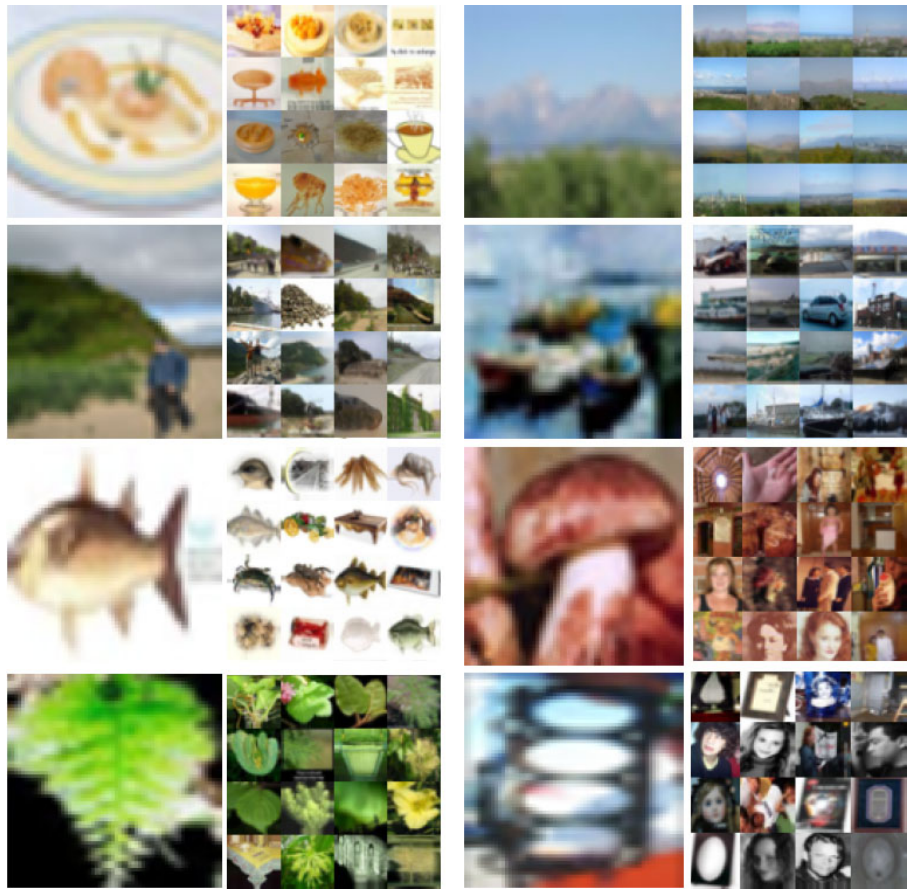


Figure 14.52 Recognition using tiny images (Torralba, Freeman, and Fergus 2008) © 2008 IEEE: columns (a) and (c) show sample input images and columns (b) and (d) show the corresponding 16 nearest neighbors in the database of 80 million tiny images.

a semantic taxonomy (Wordnet) to infer the most likely labels for a new image. Somewhere in the 80 million images, there are enough examples to associate some set of images with each of the 75,000 non-abstract nouns in Wordnet that they use in their system. Some sample recognition results are shown in Figure 14.52.

Another example of a large labeled database of images is ImageNet (Deng, Dong, Socher *et al.* 2009), which is collecting images for the 80,000 nouns (synonym sets) in WordNet (Fellbaum 1998). As of April 2010, about 500–1000 carefully vetted examples for 14841 synsets have been collected (Figure 14.53). The paper by Deng, Dong, Socher *et al.* (2009) also has a nice review of related databases.

As we mentioned in Section 14.4.3, the existence of large databases of partially labeled Internet imagery has given rise to a new sub-field of Internet computer vision, with its own workshops[21] and a special journal issue (Avidan, Baker, and Shan 2010).

[21] http://www.internetvisioner.org/.

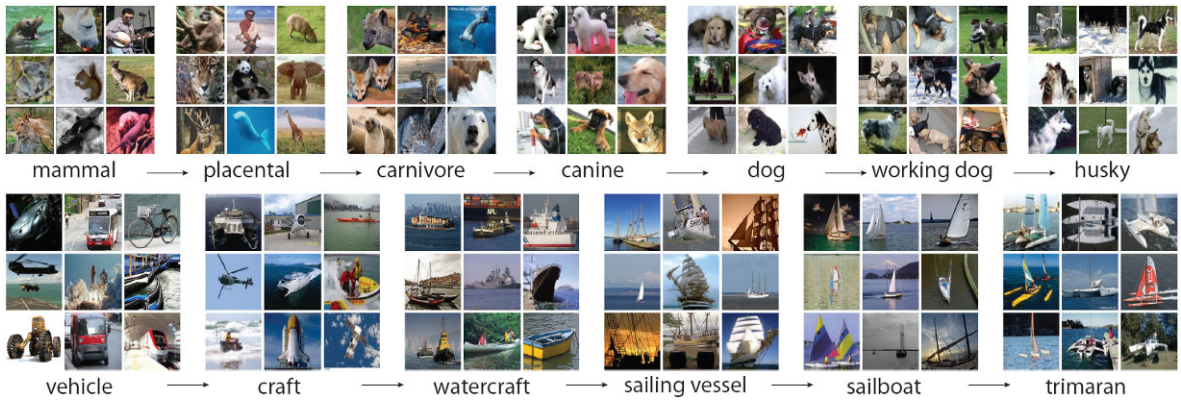


Figure 14.53 ImageNet (Deng, Dong, Socher *et al.* 2009) © 2009 IEEE. This database contains over 500 carefully vetted images for each of 14,841 (as of April, 2010) nouns from the WordNet hierarchy.

14.5.2 *Application*: Image search

Even though visual recognition algorithms are by some measures still in their infancy, they are already starting to have some impact on image search, i.e., the retrieval of images from the Web using combinations of keywords and visual similarity. Today, most image search engines rely mostly on textual keywords found in captions, nearby text, and filenames, augmented by user click-through data (Craswell and Szummer 2007). As recognition algorithms continue to improve, however, visual features and visual similarity will start being used to recognize images with missing or erroneous keywords.

The topic of searching by visual similarity has a long history and goes by a variety of names, including content-based image retrieval (CBIR) (Smeulders, Worring, Santini *et al.* 2000; Lew, Sebe, Djeraba *et al.* 2006; Vasconcelos 2007; Datta, Joshi, Li *et al.* 2008) and query by image content (QBIC) (Flickner, Sawhney, Niblack *et al.* 1995). Original publications in these fields were based primarily on simple whole-image similarity metrics, such as color and texture (Swain and Ballard 1991; Jacobs, Finkelstein, and Salesin 1995; Manjunathi and Ma 1996).

In more recent work, Fergus, Perona, and Zisserman (2004) use a feature-based learning and recognition algorithm to re-rank the outputs from a traditional keyword-based image search engine. In follow-on work, Fergus, Fei-Fei, Perona *et al.* (2005) cluster the results returned by image search using an extension of probabilistic latest semantic analysis (PLSA) (Hofmann 1999) and then select the clusters associated with the highest ranked results as the representative images for that category.

Even more recent work relies on carefully annotated image databases such as LabelMe (Russell, Torralba, Murphy *et al.* 2008). For example, Malisiewicz and Efros (2008) describe a system that, given a query image, can find similar LabelMe images, whereas Liu, Yuen, and Torralba (2009) combine feature-based correspondence algorithms with the labeled database to perform simultaneous recognition and segmentation.

14.6 Recognition databases and test sets

In addition to rapid advances in machine learning and statistical modeling techniques, one of the key ingredients in the continued improvement of recognition algorithms has been the increased availability and quality of image recognition databases.

Tables 14.1 and 14.2, which are based on similar tables in Fei-Fei, Fergus, and Torralba (2009), updated with more recent entries and URLs, show some of the mostly widely used recognition databases. Some of these databases, such as the ones for face recognition and localization, date back over a decade. The most recent ones, such as the PASCAL database, are refreshed annually with ever more challenging problems. Table 14.1 shows examples of databases used primarily for (whole image) recognition while Table 14.2 shows databases where more accurate localization or segmentation information is available and expected.

Ponce, Berg, Everingham *et al.* (2006) discuss some of the problems with earlier datasets and describe how the latest PASCAL Visual Object Classes Challenge aims to overcome these. Some examples of the 20 visual classes in the 2008 challenge are shown in Figure 14.54. The slides from the VOC workshops,[22] are a great source for pointers to the best recognition techniques currently available.

Two of the most recent trends in recognition databases are the emergence of Web-based annotation and data collection tools, and the use of search and recognition algorithms to build up databases (Ponce, Berg, Everingham *et al.* 2006). Some of the most interesting work in human annotation of images comes from a series of interactive multi-person games such as ESP (von Ahn and Dabbish 2004) and Peekaboom (von Ahn, Liu, and Blum 2006). In these games, people help each other guess the identity of a hidden image by giving textual clues as to its contents, which implicitly labels either the whole image or just regions. A more "serious" volunteer effort is the LabelMe database, in which vision researchers contribute manual polygonal region annotations in return for gaining access to the database (Russell, Torralba, Murphy *et al.* 2008).

The use of computer vision algorithms for collecting recognition databases dates back to the work of Fergus, Fei-Fei, Perona *et al.* (2005), who cluster the results returned by Google image search using an extension of PLSA and then select the clusters associated with the highest ranked results. More recent examples of related techniques include the work of Berg and Forsyth (2006) and Li and Fei-Fei (2010).

Whatever methods are used to collect and validate recognition databases, they will continue to grow in size, utility, and difficulty from year to year. They will also continue to be an essential component of research into the recognition and scene understanding problems, which remain, as always, the grand challenges of computer vision.

14.7 Additional reading

Although there are currently no specialized textbooks on image recognition and scene understanding, some surveys (Pinz 2005) and collections of papers (Ponce, Hebert, Schmid *et al.* 2006; Dickinson, Leonardis, Schiele *et al.* 2007) can be found that describe the latest approaches. Other good sources of recent research are courses on this topic, such as the ICCV

[22] http://pascallin.ecs.soton.ac.uk/challenges/VOC/.

Name / URL	Extents	Contents / Reference
Face and person recognition		
Yale face database http://www1.cs.columbia.edu/~belhumeur/	Centered face images	Frontal faces Belhumeur, Hespanha, and Kriegman (1997)
Resources for face detection http://vision.ai.uiuc.edu/mhyang/face-detection-survey.html	Various databases	Faces in various poses Yang, Kriegman, and Ahuja (2002)
FERET http://www.frvt.org/FERET	Centered face images	Frontal faces Phillips, Moon, Rizvi *et al.* (2000)
FRVT http://www.frvt.org/	Centered face images	Faces in various poses Phillips, Scruggs, O'Toole *et al.* (2010)
CMU PIE database http://www.ri.cmu.edu/projects/project_418.html	Centered face image	Faces in various poses Sim, Baker, and Bsat (2003)
CMU Multi-PIE database http://multipie.org	Centered face image	Faces in various poses Gross, Matthews, Cohn *et al.* (2010)
Faces in the Wild http://vis-www.cs.umass.edu/lfw/	Internet images	Faces in various poses Huang, Ramesh, Berg *et al.* (2007)
Consumer image person DB http://chenlab.ece.cornell.edu/people/Andy/GallagherDataset.html	Complete images	People Gallagher and Chen (2008)
Object recognition		
Caltech 101 http://www.vision.caltech.edu/Image_Datasets/Caltech101/	Segmentation masks	101 categories Fei-Fei, Fergus, and Perona (2006)
Caltech 256 http://www.vision.caltech.edu/Image_Datasets/Caltech256/	Centered objects	256 categories and clutter Griffin, Holub, and Perona (2007)
COIL-100 http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php	Centered objects	100 instances Nene, Nayar, and Murase (1996)
ETH-80 http://www.mis.tu-darmstadt.de/datasets	Centered objects	8 instances, 10 views Leibe and Schiele (2003)
Instance recognition benchmark http://vis.uky.edu/~stewe/ukbench/	Objects in various poses	2550 objects Nistér and Stewénius (2006)
Oxford buildings dataset http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/	Pictures of buildings	5062 images Philbin, Chum, Isard *et al.* (2007)
NORB http://www.cs.nyu.edu/~ylclab/data/norb-v1.0/	Bounding box	50 toys LeCun, Huang, and Bottou (2004)
Tiny images http://people.csail.mit.edu/torralba/tinyimages/	Complete images	75,000 (Wordnet) things Torralba, Freeman, and Fergus (2008)
ImageNet http://www.image-net.org/	Complete images	14,000 (Wordnet) things Deng, Dong, Socher *et al.* (2009)

Table 14.1 Image databases for recognition, adapted and expanded from Fei-Fei, Fergus, and Torralba (2009).

Name / URL	Extents	Contents / Reference
Object detection / localization		
CMU frontal faces http://vasc.ri.cmu.edu/idb/html/face/frontal_images	Patches	Frontal faces Rowley, Baluja, and Kanade (1998a)
MIT frontal faces http://cbcl.mit.edu/software-datasets/FaceData2.html	Patches	Frontal faces Sung and Poggio (1998)
CMU face detection databases http://www.ri.cmu.edu/research_project_detail.html?project_id=419	Multiple faces	Faces in various poses Schneiderman and Kanade (2004)
UIUC Image DB http://l2r.cs.uiuc.edu/~cogcomp/Data/Car/	Bounding boxes	Cars Agarwal and Roth (2002)
Caltech Pedestrian Dataset http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/	Bounding boxes	Pedestrians Dollàr, Wojek, Schiele *et al.* (2009)
Graz-02 Database http://www.emt.tugraz.at/~pinz/data/GRAZ_02/	Segmentation masks	Bikes, cars, people Opelt, Pinz, Fussenegger *et al.* (2006)
ETHZ Toys http://www.vision.ee.ethz.ch/~calvin/datasets.html	Cluttered images	Toys, boxes, magazines Ferrari, Tuytelaars, and Van Gool (2006b)
TU Darmstadt DB http://www.vision.ee.ethz.ch/~bleibe/data/datasets.html	Segmentation masks	Motorbikes, cars, cows Leibe, Leonardis, and Schiele (2008)
MSR Cambridge http://research.microsoft.com/en-us/projects/objectclassrecognition/	Segmentation masks	23 classes Shotton, Winn, Rother *et al.* (2009)
LabelMe dataset http://labelme.csail.mit.edu/	Polygonal boundary	>500 categories Russell, Torralba, Murphy *et al.* (2008)
Lotus Hill http://www.imageparsing.com/	Segmentation masks	Scenes and hierarchies Yao, Yang, Lin *et al.* (2010)
On-line annotation tools		
ESP game http://www.gwap.com/gwap/	Image descriptions	Web images von Ahn and Dabbish (2004)
Peekaboom http://www.gwap.com/gwap/	Labeled regions	Web images von Ahn, Liu, and Blum (2006)
LabelMe http://labelme.csail.mit.edu/	Polygonal boundary	High-resolution images Russell, Torralba, Murphy *et al.* (2008)
Collections of challenges		
PASCAL http://pascallin.ecs.soton.ac.uk/challenges/VOC/	Segmentation, boxes	Various Everingham, Van Gool, Williams *et al.* (2010)

Table 14.2 Image databases for detection and localization, adapted and expanded from Fei-Fei, Fergus, and Torralba (2009).

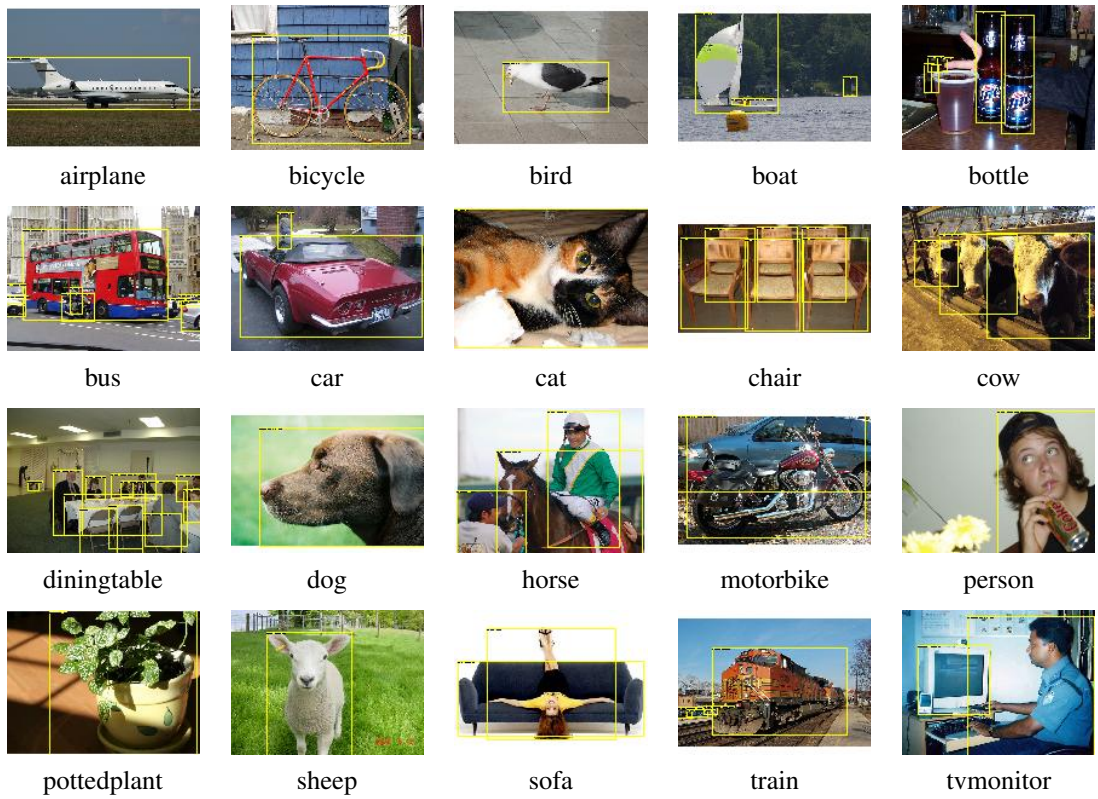


Figure 14.54 Sample images from the PASCAL Visual Object Classes Challenge 2008 (VOC2008) database (Everingham, Van Gool, Williams *et al.* 2008). The original images were obtained from flickr (http://www.flickr.com/) and the database rights are explained on http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/.

2009 short course (Fei-Fei, Fergus, and Torralba 2009) and Antonio Torralba's more comprehensive MIT course (Torralba 2008). The PASCAL VOC Challenge Web site contains workshop slides that summarize today's best performing algorithms.

The literature on face, pedestrian, car, and other object detection is quite extensive. Seminal papers in face detection include those by Osuna, Freund, and Girosi (1997), Sung and Poggio (1998), Rowley, Baluja, and Kanade (1998a), and Viola and Jones (2004), with Yang, Kriegman, and Ahuja (2002) providing a comprehensive survey of early work in this field. More recent examples include (Heisele, Ho, Wu *et al.* 2003; Heisele, Serre, and Poggio 2007).

Early work in pedestrian and car detection was carried out by Gavrila and Philomin (1999), Gavrila (1999), Papageorgiou and Poggio (2000), Mohan, Papageorgiou, and Poggio (2001), and Schneiderman and Kanade (2004). More recent examples include the work of Belongie, Malik, and Puzicha (2002), Mikolajczyk, Schmid, and Zisserman (2004), Dalal and Triggs (2005), Leibe, Seemann, and Schiele (2005), Dalal, Triggs, and Schmid (2006), Opelt, Pinz, and Zisserman (2006), Torralba (2007), Andriluka, Roth, and Schiele (2008), Felzenszwalb, McAllester, and Ramanan (2008), Rogez, Rihan, Ramalingam *et al.* (2008), Andriluka, Roth, and Schiele (2009), Kumar, Zisserman, and H.S.Torr (2009), Dollàr, Belongie, and Perona (2010). and Felzenszwalb, Girshick, McAllester *et al.* (2010).

While some of the earliest approaches to face recognition involved finding the distinc-

tive image features and measuring the distances between them (Fischler and Elschlager 1973; Kanade 1977; Yuille 1991), more recent approaches rely on comparing gray-level images, often projected onto lower dimensional subspaces (Turk and Pentland 1991a; Belhumeur, Hespanha, and Kriegman 1997; Moghaddam and Pentland 1997; Moghaddam, Jebara, and Pentland 2000; Heisele, Ho, Wu *et al.* 2003; Heisele, Serre, and Poggio 2007). Additional details on principal component analysis (PCA) and its Bayesian counterparts can be found in Appendix B.1.1 and books and articles on this topic (Hastie, Tibshirani, and Friedman 2001; Bishop 2006; Roweis 1998; Tipping and Bishop 1999; Leonardis and Bischof 2000; Vidal, Ma, and Sastry 2010). The topics of subspace learning, local distance functions, and metric learning are covered by Cai, He, Hu *et al.* (2007), Frome, Singer, Sha *et al.* (2007), Guillaumin, Verbeek, and Schmid (2009), Ramanan and Baker (2009), and Sivic, Everingham, and Zisserman (2009). An alternative to directly matching gray-level images or patches is to use non-linear local transforms such as local binary patterns (Ahonen, Hadid, and Pietikäinen 2006; Zhao and Pietikäinen 2007; Cao, Yin, Tang *et al.* 2010).

In order to boost the performance of what are essentially 2D appearance-based models, a variety of shape and pose deformation models have been developed (Beymer 1996; Vetter and Poggio 1997), including Active Shape Models (Lanitis, Taylor, and Cootes 1997; Cootes, Cooper, Taylor *et al.* 1995; Davies, Twining, and Taylor 2008), Elastic Bunch Graph Matching (Wiskott, Fellous, Krüger *et al.* 1997), 3D Morphable Models (Blanz and Vetter 1999), and Active Appearance Models (Costen, Cootes, Edwards *et al.* 1999; Cootes, Edwards, and Taylor 2001; Gross, Baker, Matthews *et al.* 2005; Gross, Matthews, and Baker 2006; Matthews, Xiao, and Baker 2007; Liang, Xiao, Wen *et al.* 2008; Ramnath, Koterba, Xiao *et al.* 2008). The topic of head pose estimation, in particular, is covered in a recent survey by Murphy-Chutorian and Trivedi (2009).

Additional information about face recognition can be found in a number of surveys and books on this topic (Chellappa, Wilson, and Sirohey 1995; Zhao, Chellappa, Phillips *et al.* 2003; Li and Jain 2005) as well as on the Face Recognition Web site.[23] Databases for face recognition are discussed by Phillips, Moon, Rizvi *et al.* (2000), Sim, Baker, and Bsat (2003), Gross, Shi, and Cohn (2005), Huang, Ramesh, Berg *et al.* (2007), and Phillips, Scruggs, O'Toole *et al.* (2010).

Algorithms for instance recognition, i.e., the detection of static man-made objects that only vary slightly in appearance but may vary in 3D pose, are mostly based on detecting 2D points of interest and describing them using viewpoint-invariant descriptors (Lowe 2004; Rothganger, Lazebnik, Schmid *et al.* 2006; Ferrari, Tuytelaars, and Van Gool 2006b; Gordon and Lowe 2006; Obdržálek and Matas 2006; Kannala, Rahtu, Brandt *et al.* 2008; Sivic and Zisserman 2009).

As the size of the database being matched increases, it becomes more efficient to quantize the visual descriptors into words (Sivic and Zisserman 2003; Schindler, Brown, and Szeliski 2007; Sivic and Zisserman 2009; Turcot and Lowe 2009), and to then use information-retrieval techniques, such as inverted indices (Nistér and Stewénius 2006; Philbin, Chum, Isard *et al.* 2007; Philbin, Chum, Sivic *et al.* 2008), query expansion (Chum, Philbin, Sivic *et al.* 2007; Agarwal, Snavely, Simon *et al.* 2009), and min hashing (Philbin and Zisserman 2008; Li, Wu, Zach *et al.* 2008; Chum, Philbin, and Zisserman 2008; Chum and Matas 2010) to perform efficient retrieval and clustering.

[23] http://www.face-rec.org/.

A number of surveys, collections of papers, and course notes have been written on the topic of category recognition (Pinz 2005; Ponce, Hebert, Schmid *et al.* 2006; Dickinson, Leonardis, Schiele *et al.* 2007; Fei-Fei, Fergus, and Torralba 2009). Some of the seminal papers on the bag of words (bag of keypoints) approach to whole-image category recognition have been written by Csurka, Dance, Fan *et al.* (2004), Lazebnik, Schmid, and Ponce (2006), Csurka, Dance, Perronnin *et al.* (2006), Grauman and Darrell (2007b), and Zhang, Marszalek, Lazebnik *et al.* (2007). Additional and more recent papers in this area include Sivic, Russell, Efros *et al.* (2005), Serre, Wolf, and Poggio (2005), Opelt, Pinz, Fussenegger *et al.* (2006), Grauman and Darrell (2007a), Torralba, Murphy, and Freeman (2007), Boiman, Shechtman, and Irani (2008), Ferencz, Learned-Miller, and Malik (2008), and Mutch and Lowe (2008). It is also possible to recognize objects based on their contours, e.g., using shape contexts (Belongie, Malik, and Puzicha 2002) or other techniques (Jurie and Schmid 2004; Shotton, Blake, and Cipolla 2005; Opelt, Pinz, and Zisserman 2006; Ferrari, Tuytelaars, and Van Gool 2006a).

Many object recognition algorithms use part-based decompositions to provide greater invariance to articulation and pose. Early algorithms focused on the relative positions of the parts (Fischler and Elschlager 1973; Kanade 1977; Yuille 1991) while newer algorithms use more sophisticated models of appearance (Felzenszwalb and Huttenlocher 2005; Fergus, Perona, and Zisserman 2007; Felzenszwalb, McAllester, and Ramanan 2008). Good overviews on part-based models for recognition can be found in the course notes of Fergus 2007b; 2009.

Carneiro and Lowe (2006) discuss a number of graphical models used for part-based recognition, which include trees and stars (Felzenszwalb and Huttenlocher 2005; Fergus, Perona, and Zisserman 2005; Felzenszwalb, McAllester, and Ramanan 2008), k-fans (Crandall, Felzenszwalb, and Huttenlocher 2005; Crandall and Huttenlocher 2006), and constellations (Burl, Weber, and Perona 1998; Weber, Welling, and Perona 2000; Fergus, Perona, and Zisserman 2007). Other techniques that use part-based recognition include those developed by Dorkó and Schmid (2003) and Bar-Hillel, Hertz, and Weinshall (2005).

Combining object recognition with scene segmentation can yield strong benefits. One approach is to pre-segment the image into pieces and then match the pieces to portions of the model (Mori, Ren, Efros *et al.* 2004; Mori 2005; He, Zemel, and Ray 2006; Russell, Efros, Sivic *et al.* 2006; Borenstein and Ullman 2008; Csurka and Perronnin 2008; Gu, Lim, Arbelaez *et al.* 2009). Another is to vote for potential object locations and scales based on object detection (Leibe, Leonardis, and Schiele 2008). One of the currently most popular approaches is to use conditional random fields (Kumar and Hebert 2006; He, Zemel, and Carreira-Perpiñán 2004; He, Zemel, and Ray 2006; Levin and Weiss 2006; Winn and Shotton 2006; Hoiem, Rother, and Winn 2007; Rabinovich, Vedaldi, Galleguillos *et al.* 2007; Verbeek and Triggs 2007; Yang, Meer, and Foran 2007; Batra, Sukthankar, and Chen 2008; Larlus and Jurie 2008; He and Zemel 2008; Shotton, Winn, Rother *et al.* 2009; Kumar, Torr, and Zisserman 2010), which produce some of the best results on the difficult PASCAL VOC segmentation challenge (Shotton, Johnson, and Cipolla 2008; Kohli, Ladický, and Torr 2009).

More and more recognition algorithms are starting to use scene context as part of their recognition strategy. Representative papers in this area include those by Torralba (2003), Torralba, Murphy, Freeman *et al.* (2003), Murphy, Torralba, and Freeman (2003), Torralba, Murphy, and Freeman (2004), Crandall and Huttenlocher (2007), Rabinovich, Vedaldi, Galleguillos *et al.* (2007), Russell, Torralba, Liu *et al.* (2007), Hoiem, Efros, and Hebert (2008a),

Hoiem, Efros, and Hebert (2008b), Sudderth, Torralba, Freeman *et al.* (2008), and Divvala, Hoiem, Hays *et al.* (2009).

Sophisticated machine learning techniques are also becoming a key component of successful object detection and recognition algorithms (Varma and Ray 2007; Felzenszwalb, McAllester, and Ramanan 2008; Fritz and Schiele 2008; Sivic, Russell, Zisserman *et al.* 2008; Vedaldi, Gulshan, Varma *et al.* 2009), as is exploiting large human-labeled databases (Russell, Torralba, Liu *et al.* 2007; Malisiewicz and Efros 2008; Torralba, Freeman, and Fergus 2008; Liu, Yuen, and Torralba 2009). Rough three-dimensional models are also making a comeback for recognition, as evidenced in some recent papers (Savarese and Fei-Fei 2007, 2008; Sun, Su, Savarese *et al.* 2009; Su, Sun, Fei-Fei *et al.* 2009). As always, the latest conferences on computer vision are your best reference for the newest algorithms in this rapidly evolving field.

14.8 Exercises

Ex 14.1: Face detection Build and test one of the face detectors presented in Section 14.1.1.

1. Download one or more of the labeled face detection databases in Table 14.2.
2. Generate your own negative examples by finding photographs that do not contain any people.
3. Implement one of the following face detectors (or devise one of your own):
   - boosting (Algorithm 14.1) based on simple area features, with an optional cascade of detectors (Viola and Jones 2004);
   - PCA face subspace (Moghaddam and Pentland 1997);
   - distances to clustered face and non-face prototypes, followed by a neural network (Sung and Poggio 1998) or SVM (Osuna, Freund, and Girosi 1997) classifier;
   - a multi-resolution neural network trained directly on normalized gray-level patches (Rowley, Baluja, and Kanade 1998a).
4. Test the performance of your detector on the database by evaluating the detector at every location in a sub-octave pyramid. Optionally retrain your detector on false positive examples you get on non-face images.

Ex 14.2: Determining the threshold for AdaBoost Given a set of function evaluations on the training examples $\boldsymbol{x}_i$, $f_i = f(\boldsymbol{x}_i) \in \pm 1$, training labels $y_i \in \pm 1$, and weights $w_i \in (0, 1)$, as explained in Algorithm 14.1, devise an efficient algorithm to find values of θ and $s = \pm 1$ that maximize

$$\sum_i w_i y_i h(s f_i, \theta), \tag{14.43}$$

where $h(x, \theta) = \text{sign}(x - \theta)$.

Ex 14.3: Face recognition using eigenfaces Collect a set of facial photographs and then build a recognition system to re-recognize the same people.

1. Take several photos of each of your classmates and store them.
2. Align the images by automatically or manually detecting the corners of the eyes and using a similarity transform to stretch and rotate each image to a canonical position.
3. Compute the average image and a PCA subspace for the face images
4. Take a new set of photographs a week later and use them as your test set.
5. Compare each new image to each database image and select the nearest one as the recognized identity. Verify that the distance in PCA space is close to the distance computed with a full SSD (sum of squared difference) measure.
6. (Optional) Compute different principal components for identity and expression, and use them to improve your recognition results.

Ex 14.4: Bayesian face recognition Moghaddam, Jebara, and Pentland (2000) compute separate covariance matrices $\mathbf{\Sigma}_I$ and $\mathbf{\Sigma}_E$ by looking at differences between *all* pairs of images. At run time, they select the *nearest* image to determine the facial identity. Does it make sense to estimate statistics for all pairs of images and use them for testing the distance to the nearest exemplar? Discuss whether this is statistically correct.

How is the all-pair intrapersonal covariance matrix $\mathbf{\Sigma}_I$ related to the within-class scatter matrix $\boldsymbol{S}_{\mathrm{W}}$? Does a similar relationship hold between $\mathbf{\Sigma}_E$ and $\boldsymbol{S}_{\mathrm{B}}$?

Ex 14.5: Modular eigenfaces Extend your face recognition system to separately match the eye, nose, and mouth regions, as shown in Figure 14.18.

1. After normalizing face images to a canonical scale and location, manually segment out some of the eye, nose, and face regions.
2. Build separate detectors for these three (or four) kinds of region, either using a subspace (PCA) approach or one of the techniques presented in Section 14.1.1.
3. For each new image to be recognized, first detect the locations of the facial features.
4. Then, match the individual features against your database and note the locations of these features.
5. Train and test a classifier that uses the individual feature matching IDs as well as (optionally) the feature locations to perform face recognition.

Ex 14.6: Recognition-based color balancing Build a system that recognizes the most important color areas in common photographs (sky, grass, skin) and color balances the image accordingly. Some references and ideas for skin detection are given in Exercise 2.8 and by Forsyth and Fleck (1999), Jones and Rehg (2001), Vezhnevets, Sazonov, and Andreeva (2003), and Kakumanu, Makrogiannis, and Bourbakis (2007). These may give you ideas for how to detect other regions or you can try more sophisticated MRF-based approaches (Shotton, Winn, Rother *et al.* 2009).

Ex 14.7: Pedestrian detection Build and test one of the pedestrian detectors presented in Section 14.1.2.

Ex 14.8: Simple instance recognition Use the feature detection, matching, and alignment algorithms you developed in Exercises 4.1–4.4 and 9.2 to find matching images given a query image or region (Figure 14.26).

Evaluate several feature detectors, descriptors, and robust geometric verification strategies, either on your own or by comparing your results with those of classmates.

Ex 14.9: Large databases and location recognition Extend the previous exercise to larger databases using quantized visual words and information retrieval techniques, as described in Algorithm 14.2.

Test your algorithm on a large database, such as the one used by Nistér and Stewénius (2006) or Philbin, Chum, Sivic *et al.* (2008), which are listed in Table 14.1. Alternatively, use keyword search on the Web or in a photo sharing site (e.g., for a city) to create your own database.

Ex 14.10: Bag of words Adapt the feature extraction and matching pipeline developed in Exercise 14.8 to category (class) recognition, using some of the techniques described in Section 14.4.1.

1. Download the training and test images from one or more of the databases listed in Tables 14.1 and 14.2, e.g., Caltech 101, Caltech 256, or PASCAL VOC.

2. Extract features from each of the training images, quantize them, and compute the *tf-idf* vectors (bag of words histograms).

3. As an option, consider not quantizing the features and using pyramid matching (14.40–14.41) (Grauman and Darrell 2007b) or using a spatial pyramid for greater selectivity (Lazebnik, Schmid, and Ponce 2006).

4. Choose a classification algorithm (e.g., nearest neighbor classification or support vector machine) and "train" your recognizer, i.e., build up the appropriate data structures (e.g., k-d trees) or set the appropriate classifier parameters.

5. Test your algorithm on the test data set using the same pipeline you developed in steps 2–4 and compare your results to the best reported results.

6. Explain why your results differ from the previously reported ones and give some ideas for how you could improve your system.

You can find a good synopsis of the best-performing classification algorithms and their approaches in the report of the PASCAL Visual Object Classes Challenge found on their Web site (http://pascallin.ecs.soton.ac.uk/challenges/VOC/).

Ex 14.11: Object detection and localization Extend the classification algorithm developed in the previous exercise to localize the objects in an image by reporting a bounding box around each detected object. The easiest way to do this is to use a sliding window approach. Some pointers to recent techniques in this area can be found in the workshop associated with the PASCAL VOC 2008 Challenge.

Ex 14.12: Part-based recognition Choose one or more of the techniques described in Section 14.4.2 and implement a part-based recognition system. Since these techniques are fairly involved, you will need to read several of the research papers in this area, select which general approach you want to follow, and then implement your algorithm. A good starting point could be the paper by Felzenszwalb, McAllester, and Ramanan (2008), since it performed well in the PASCAL VOC 2008 detection challenge.

Ex 14.13: Recognition and segmentation Choose one or more of the techniques described in Section 14.4.3 and implement a simultaneous recognition and segmentation system. Since these techniques are fairly involved, you will need to read several of the research papers in this area, select which general approach you want to follow, and then implement your algorithm. Test your algorithm on one or more of the segmentation databases in Table 14.2.

Ex 14.14: Context Implement one or more of the context and scene understanding systems described in Section 14.5 and report on your experience. Does context or whole scene understanding perform better at naming objects than stand-alone systems?

Ex 14.15: Tiny images Download the tiny images database from http://people.csail.mit.edu/torralba/tinyimages/ and build a classifier based on comparing your test images directly against all of the labeled training images. Does this seem like a promising approach?

Chapter 15

Conclusion

In this book, we have covered a broad range of computer vision topics. Starting with image formation, we have seen how images can be pre-processed to remove noise or blur, segmented into regions, or converted into feature descriptors. Multiple images can be matched and registered, with the results used to estimate motion, track people, reconstruct 3D models, or merge images into more attractive and interesting composites and renderings. Images can also be analyzed to produce semantic descriptions of their content. However, the gap between computer and human performance in this area is still large and is likely to remain so for many years.

Our study has also exposed us to a wide range of mathematical techniques. These include continuous mathematics, such as signal processing, variational approaches, three-dimensional and projective geometry, linear algebra, and least squares. We have also studied topics in discrete mathematics and computer science, such as graph algorithms, combinatorial optimization, and even database techniques for information retrieval. Since many problems in computer vision are inverse problems that involve estimating unknown quantities from noisy input data, we have also looked at Bayesian statistical inference techniques, as well as machine learning techniques to learn probabilistic models from large amounts of training data. As the availability of partially labeled visual imagery on the Internet continues to increase exponentially, this latter approach will continue to have a major impact on our field.

You may ask: why is our field so broad and aren't there any unifying principles that can be used to simplify our study? Part of the answer lies in the expansive definition of computer vision, which is the analysis of images and video, as well as the incredible complexity inherent in the formation of visual imagery. In some ways, our field is as complex as the study of automotive engineering, which requires an understanding of internal combustion, mechanics, aerodynamics, ergonomics, electrical circuitry, and control systems, among other topics. Computer vision similarly draws on a wide variety of sub-disciplines, which makes it challenging to cover in a one-semester course, let alone to achieve mastery during a course of graduate studies. Conversely, the incredible breadth and technical complexity of computer vision problems is what draws many people to this research field.

Because of this richness and the difficulty in making and measuring progress, I have attempted to instill in my students and in readers of this book a discipline founded on principles from engineering, science, and statistics.

R. Szeliski, *Computer Vision: Algorithms and Applications*, Texts in Computer Science,
DOI 10.1007/978-1-84882-935-0_15, © Springer-Verlag London Limited 2011

The engineering approach to problem solving is to first carefully define the overall problem being tackled and to question the basic assumptions and goals inherent in this process. Once this has been done, a number of alternative solutions or approaches are implemented and carefully tested, paying attention to issues such as reliability and computational cost. Finally, one or more solutions are deployed and evaluated in real-world settings. For this reason, this book contains many different alternatives for solving vision problems, many of which are sketched out in the exercises for students to implement and test on their own.

The scientific approach builds upon a basic understanding of physical principles. In the case of computer vision, this includes the physics of man-made and natural structures, image formation, including lighting and atmospheric effects, optics, and noisy sensors. The task is to then invert this formation using stable and efficient algorithms to obtain reliable descriptions of the scene and other quantities of interest. The scientific approach also encourages us to formulate and test hypotheses, which is similar to the extensive testing and evaluation inherent in engineering disciplines.

Lastly, because so much about the image formation process is inherently uncertain and ambiguous, a statistical approach that models both uncertainty in the world (e.g., the number and types of animals in a picture) and noise in the image formation process, is often essential. Bayesian inference techniques can then be used to combine prior and measurement models to estimate the unknowns and to model their uncertainty. Machine learning techniques can be used to create the probabilistic models in the first place. Efficient learning and inference algorithms, such as dynamic programming, graph cuts, and belief propagation, often play a crucial role in this process.

Given the breadth of material we have covered in this book, what new developments are we likely to see in the future? As I have mentioned before, one of the recent trends in computer vision is using the massive amounts of partially labeled visual data on the Internet as sources for learning visual models of scenes and objects. We have already seen data-driven approaches succeed in related fields such as speech recognition, machine translation, speech and music synthesis, and even computer graphics (both in image-based rendering and animation from motion capture). A similar process has been occurring in computer vision, with some of the most exciting new work occurring at the intersection of the object recognition and machine learning fields.

More traditional quantitative techniques in computer vision such as motion estimation, stereo correspondence, and image enhancement, all benefit from better prior models for images, motions, and disparities, as well as efficient statistical inference techniques such as those for inhomogeneous and higher-order Markov random fields. Some techniques, such as feature matching and structure from motion, have matured to where they can be applied to almost arbitrary collections of images of static scenes. This has resulted in an explosion of work in 3D modeling from Internet datasets, which again is related to visual recognition from massive amounts of data.

While these are all encouraging developments, the gap between human and machine performance in semantic scene understanding remains large. It may be many years before computers can name and outline all of the objects in a photograph with the same skill as a two-year-old child. However, we have to remember that human performance is often the result of many years of training and familiarity and often works best in special ecologically important situations. For example, while humans appear to be experts at face recognition, our actual

performance when shown people we do not know well is not that good. Combining vision algorithms with general inference techniques that reason about the real world will likely lead to more breakthroughs, although some of the problems may turn out to be "AI-complete", in the sense that a full emulation of human experience and intelligence may be necessary.

Whatever the outcome of these research endeavors, computer vision is already having a tremendous impact in many areas, including digital photography, visual effects, medical imaging, safety and surveillance, and Web-based search. The breadth of the problems and techniques inherent in this field, combined with the richness of the mathematics and the utility of the resulting algorithms, will ensure that this remains an exciting area of study for years to come.

Appendix A

Linear algebra and numerical techniques

A.1 Matrix decompositions . . . 646
A.1.1 Singular value decomposition . . . 646
A.1.2 Eigenvalue decomposition . . . 647
A.1.3 QR factorization . . . 649
A.1.4 Cholesky factorization . . . 650
A.2 Linear least squares . . . 651
A.2.1 Total least squares . . . 653
A.3 Non-linear least squares . . . 654
A.4 Direct sparse matrix techniques . . . 655
A.4.1 Variable reordering . . . 656
A.5 Iterative techniques . . . 656
A.5.1 Conjugate gradient . . . 657
A.5.2 Preconditioning . . . 659
A.5.3 Multigrid . . . 660

In this appendix, we introduce some elements of linear algebra and numerical techniques that are used elsewhere in the book. We start with some basic decompositions in matrix algebra, including the singular value decomposition (SVD), eigenvalue decompositions, and other matrix decompositions (factorizations). Next, we look at the problem of linear least squares, which can be solved using either the QR decomposition or normal equations. This is followed by non-linear least squares, which arise when the measurement equations are not linear in the unknowns or when robust error functions are used. Such problems require iteration to find a solution. Next, we look at direct solution (factorization) techniques for sparse problems, where the ordering of the variables can have a large influence on the computation and memory requirements. Finally, we discuss iterative techniques for solving large linear (or linearized) least squares problems. Good general references for much of this material include the work by Björck (1996), Golub and Van Loan (1996), Trefethen and Bau (1997), Meyer (2000), Nocedal and Wright (2006), and Björck and Dahlquist (2010).

A note on vector and matrix indexing. To be consistent with the rest of the book and with the general usage in the computer science and computer vision communities, I adopt a 0-based indexing scheme for vector and matrix element indexing. Please note that most mathematical textbooks and papers use 1-based indexing, so you need to be aware of the differences when you read this book.

Software implementations. Highly optimized and tested libraries corresponding to the algorithms described in this appendix are readily available and are listed in Appendix C.2.

A.1 Matrix decompositions

In order to better understand the structure of matrices and more stably perform operations such as inversion and system solving, a number of decompositions (or factorizations) can be used. In this section, we review singular value decomposition (SVD), eigenvalue decomposition, QR factorization, and Cholesky factorization.

A.1.1 Singular value decomposition

One of the most useful decompositions in matrix algebra is the *singular value decomposition* (SVD), which states that any real-valued $M \times N$ matrix $\boldsymbol{A}$ can be written as

$$\begin{aligned} \boldsymbol{A}_{M\times N} &= \boldsymbol{U}_{M\times P}\boldsymbol{\Sigma}_{P\times P}\boldsymbol{V}^T_{P\times N} \\ &= \left[\begin{array}{c|c|c} & & \\ \boldsymbol{u}_0 & \cdots & \boldsymbol{u}_{p-1} \\ & & \end{array}\right] \left[\begin{array}{ccc} \sigma_0 & & \\ & \ddots & \\ & & \sigma_{p-1} \end{array}\right] \left[\begin{array}{c} \boldsymbol{v}_0^T \\ \hline \cdots \\ \hline \boldsymbol{v}_{p-1}^T \end{array}\right], \end{aligned} \tag{A.1}$$

where $P = \min(M, N)$. The matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal, i.e., $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$, and so are their column vectors,

$$\boldsymbol{u}_i \cdot \boldsymbol{u}_j = \boldsymbol{v}_i \cdot \boldsymbol{v}_j = \delta_{ij}. \tag{A.2}$$

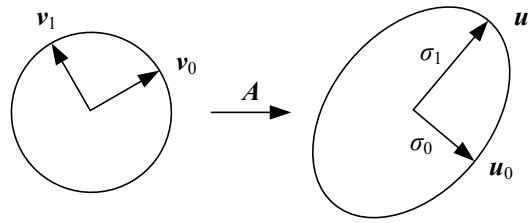


Figure A.1 The action of a matrix $\boldsymbol{A}$ can be visualized by thinking of the domain as being spanned by a set of orthonormal vectors $\boldsymbol{v}_j$, each of which is transformed to a new orthogonal vector $\boldsymbol{u}_j$ with a length σ_j. When $\boldsymbol{A}$ is interpreted as a covariance matrix and its eigenvalue decomposition is performed, each of the $\boldsymbol{u}_j$ axes denote a principal direction (component) and each σ_j denotes one standard deviation along that direction.

The singular values are all non-negative and can be ordered in decreasing order

$$\sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_{p-1} \geq 0. \tag{A.3}$$

A geometric intuition for the SVD of a matrix $\boldsymbol{A}$ can be obtained by re-writing $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ in (A.2) as

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma} \quad \text{or} \quad \boldsymbol{A}\boldsymbol{v}_j = \sigma_j \boldsymbol{u}_j. \tag{A.4}$$

This formula says that the matrix $\boldsymbol{A}$ takes any basis vector $\boldsymbol{v}_j$ and maps it to a direction $\boldsymbol{u}_j$ with length σ_j, as shown in Figure A.1

If only the first r singular values are positive, the matrix $\boldsymbol{A}$ is of *rank* r and the index p in the SVD decomposition (A.2) can be replaced by r. (In other words, we can drop the last $p - r$ columns of $\boldsymbol{U}$ and $\boldsymbol{V}$.)

An important property of the singular value decomposition of a matrix (also true for the eigenvalue decomposition of a real symmetric non-negative definite matrix) is that if we truncate the expansion

$$\boldsymbol{A} = \sum_{j=0}^{t} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^T, \tag{A.5}$$

we obtain the best possible least squares approximation to the original matrix $\boldsymbol{A}$. This is used both in eigenface-based face recognition systems (Section 14.2.1) and in the separable approximation of convolution kernels (3.21).

A.1.2 Eigenvalue decomposition

If the matrix $\boldsymbol{C}$ is symmetric ($m = n$),[1] it can be written as an eigenvalue decomposition,

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = \left[\begin{array}{c|c|c} \boldsymbol{u}_0 & \cdots & \boldsymbol{u}_{n-1} \end{array}\right] \begin{bmatrix} \lambda_0 & & \\ & \ddots & \\ & & \lambda_{n-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_0^T \\ \hline \cdots \\ \hline \boldsymbol{u}_{n-1}^T \end{bmatrix} \\ &= \sum_{i=0}^{n-1} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T. \end{aligned} \tag{A.6}$$

[1] In this appendix, we denote symmetric matrices using $\boldsymbol{C}$ and general rectangular matrices using $\boldsymbol{A}$.

(The eigenvector matrix $\boldsymbol{U}$ is sometimes written as $\boldsymbol{\Phi}$ and the eigenvectors $\boldsymbol{u}$ as $\boldsymbol{\phi}$.) In this case, the eigenvalues

$$\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{n-1} \tag{A.7}$$

can be both positive and negative.[2]

A special case of the symmetric matrix $\boldsymbol{C}$ occurs when it is constructed as the sum of a number of outer products

$$\boldsymbol{C} = \sum_i \boldsymbol{a}_i \boldsymbol{a}_i^T = \boldsymbol{A}\boldsymbol{A}^T, \tag{A.8}$$

which often occurs when solving least squares problems (Appendix A.2), where the matrix $\boldsymbol{A}$ consists of all the $\boldsymbol{a}_i$ column vectors stacked side-by-side. In this case, we are guaranteed that all of the eigenvalues λ_i are non-negative. The associated matrix $\boldsymbol{C}$ is *positive semi-definite*

$$\boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x} \geq 0, \quad \forall \boldsymbol{x}. \tag{A.9}$$

If the matrix $\boldsymbol{C}$ is of full rank, the eigenvalues are all positive and the matrix is called *symmetric positive definite* (SPD).

Symmetric positive semi-definite matrices also arise in the statistical analysis of data, since they represent the *covariance* of a set of $\{\boldsymbol{x}_i\}$ points around their mean $\bar{\boldsymbol{x}}$,

$$\boldsymbol{C} = \frac{1}{n} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T. \tag{A.10}$$

In this case, performing the eigenvalue decomposition is known as *principal component analysis* (PCA), since it models the principal directions (and magnitudes) of variation of the point distribution around their mean, as shown in Section 5.1.1 (5.13–5.15), Section 14.2.1 (14.9), and Appendix B.1.1 (B.10). Figure A.1 shows how the principal components of the covariance matrix $\boldsymbol{C}$ denote the principal axes $\boldsymbol{u}_j$ of the uncertainty ellipsoid corresponding to this point distribution and how the $\sigma_j = \sqrt{\lambda_j}$ denote the standard deviations along each axis.

The eigenvalues and eigenvectors of $\boldsymbol{C}$ and the singular values and singular vectors of $\boldsymbol{A}$ are closely related. Given

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{A.11}$$

we get

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T. \tag{A.12}$$

From this, we see that $\lambda_i = \sigma_i^2$ and that the left singular vectors of $\boldsymbol{A}$ are the eigenvectors of $\boldsymbol{C}$.

This relationship gives us an efficient method for computing the eigenvalue decomposition of large matrices that are rank deficient, such as the scatter matrices observed in computing eigenfaces (Section 14.2.1). Observe that the covariance matrix $\boldsymbol{C}$ in (14.9) is exactly the same as $\boldsymbol{C}$ in (A.8). Note also that the individual difference-from-mean images $\boldsymbol{a}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}$ are long vectors of length P (the number of pixels in the image), while the total number of exemplars N (the number of faces in the training database) is much smaller. Instead of forming $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^T$, which is $P \times P$, we form the matrix

$$\hat{\boldsymbol{C}} = \boldsymbol{A}^T\boldsymbol{A}, \tag{A.13}$$

[2] Eigenvalue decompositions can be computed for non-symmetric matrices but the eigenvalues and eigenvectors can have complex entries in that case.

which is $N \times N$. (This involves taking the dot product between every pair of difference images $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$.) The eigenvalues of $\hat{\boldsymbol{C}}$ are the squared singular values of $\boldsymbol{A}$, namely $\boldsymbol{\Sigma}^2$, and are hence also the eigenvalues of $\boldsymbol{C}$. The eigenvectors of $\hat{\boldsymbol{C}}$ are the right singular vectors $\boldsymbol{V}$ of $\boldsymbol{A}$, from which the desired eigenfaces $\boldsymbol{U}$, which are the left singular vectors of $\boldsymbol{A}$, can be computed as

$$\boldsymbol{U} = \boldsymbol{A}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}. \tag{A.14}$$

This final step is essentially computing the eigenfaces as linear combinations of the difference images (Turk and Pentland 1991a). If you have access to a high-quality linear algebra package such as LAPACK, routines for efficiently computing a small number of the left singular vectors and singular values of rectangular matrices such as $\boldsymbol{A}$ are usually provided (Appendix C.2). However, if storing all of the images in memory is prohibitive, the construction of $\hat{\boldsymbol{C}}$ in (A.13) can be used instead.

How can eigenvalue and singular value decompositions actually be computed? Notice that an eigenvector is defined by the equation

$$\lambda_i \boldsymbol{u}_i = \boldsymbol{C}\boldsymbol{u}_i \quad \text{or} \quad (\lambda_i \boldsymbol{I} - \boldsymbol{C})\boldsymbol{u}_i = 0. \tag{A.15}$$

(This can be derived from (A.6) by post-multiplying both sides by $\boldsymbol{u}_i$.) Since the latter equation is *homogeneous*, i.e., it has a zero right-hand-side, it can only have a non-zero (non-trivial) solution for $\boldsymbol{u}_i$ if the system is rank deficient, i.e.,

$$|(\lambda \boldsymbol{I} - \boldsymbol{C})| = 0. \tag{A.16}$$

Evaluating this determinant yields a *characteristic* polynomial equation in λ, which can be solved for small problems, e.g., 2×2 or 3×3 matrices, in closed form.

For larger matrices, iterative algorithms that first reduce the matrix $\boldsymbol{C}$ to a real symmetric tridiagonal form using orthogonal transforms and then perform QR iterations are normally used (Golub and Van Loan 1996; Trefethen and Bau 1997; Björck and Dahlquist 2010). Since these techniques are rather involved, it is best to use a linear algebra package such as LAPACK (Anderson, Bai, Bischof *et al.* 1999)—see Appendix C.2.

Factorization with missing data requires different kinds of iterative algorithms, which often involve either hallucinating the missing terms or minimizing some weighted reconstruction metric, which is intrinsically much more challenging than regular factorization. This area has been widely studied in computer vision (Shum, Ikeuchi, and Reddy 1995; De la Torre and Black 2003; Huynh, Hartley, and Heyden 2003; Buchanan and Fitzgibbon 2005; Gross, Matthews, and Baker 2006; Torresani, Hertzmann, and Bregler 2008) and is sometimes called *generalized PCA*. However, this term is also sometimes used to denote algebraic subspace clustering techniques, which is the subject of a forthcoming monograph by Vidal, Ma, and Sastry (2010).

A.1.3 QR factorization

A widely used technique for stably solving poorly conditioned least squares problems (Björck 1996) and as the basis of more complex algorithms, such as computing the SVD and eigenvalue decompositions, is the QR factorization,

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}, \tag{A.17}$$

procedure *Cholesky*$(\boldsymbol{C}, \boldsymbol{R})$:

$\boldsymbol{R} = \boldsymbol{C}$

for $i = 0 \ldots n-1$

for $j = i+1 \ldots n-1$

$\boldsymbol{R}_{j,j:n-1} = \boldsymbol{R}_{j,j:n-1} - r_{ij} r_{ii}^{-1} \boldsymbol{R}_{i,j:n-1}$

$\boldsymbol{R}_{i,i:n-1} = r_{ii}^{-1/2} \boldsymbol{R}_{i,i:n-1}$

Algorithm A.1 Cholesky decomposition of the matrix $\boldsymbol{C}$ into its upper triangular form $\boldsymbol{R}$.

where $\boldsymbol{Q}$ is an *orthonormal* (or *unitary*) matrix $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}$ and $\boldsymbol{R}$ is upper triangular.[3] In computer vision, QR can be used to convert a camera matrix into a rotation matrix and an upper-triangular calibration matrix (6.35) and also in various self-calibration algorithms (Section 7.2.2). The most common algorithms for computing QR decompositions, modified Gram–Schmidt, Householder transformations, and Givens rotations, are described by Golub and Van Loan (1996), Trefethen and Bau (1997), and Björck and Dahlquist (2010) and are also found in LAPACK. Unlike the SVD and eigenvalue decompositions, QR factorization does not require iteration and can be computed exactly in $O(MN^2 + N^3)$ operations, where M is the number of rows and N is the number of columns (for a tall matrix).

A.1.4 Cholesky factorization

Cholesky factorization can be applied to any symmetric positive definite matrix $\boldsymbol{C}$ to convert it into a product of symmetric lower and upper triangular matrices,

$$\boldsymbol{C} = \boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{R}^T\boldsymbol{R}, \tag{A.18}$$

where $\boldsymbol{L}$ is a lower-triangular matrix and $\boldsymbol{R}$ is an upper-triangular matrix. Unlike Gaussian elimination, which may require pivoting (row and column reordering) or may become unstable (sensitive to roundoff errors or reordering), Cholesky factorization remains stable for positive definite matrices, such as those that arise from normal equations in least squares problems (Appendix A.2). Because of the form of (A.18), the matrices $\boldsymbol{L}$ and $\boldsymbol{R}$ are sometimes called *matrix square roots*.[4]

The algorithm to compute an upper triangular Cholesky decomposition of $\boldsymbol{C}$ is a straightforward symmetric generalization of Gaussian elimination and is based on the decomposition (Björck 1996; Golub and Van Loan 1996)

$$\boldsymbol{C} = \begin{bmatrix} \gamma & \boldsymbol{c}^T \\ \boldsymbol{c} & \boldsymbol{C}_{11} \end{bmatrix} \tag{A.19}$$

[3] The term "R" comes from the German name for the lower–upper (LU) decomposition, which is LR for "links" and "rechts" (left and right of the diagonal).

[4] In fact, there exists a whole family of matrix square roots. Any matrix of the form $\boldsymbol{L}\boldsymbol{Q}$ or $\boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q}$ is a unitary matrix, is a square root of $\boldsymbol{C}$.

$$= \begin{bmatrix} \gamma^{1/2} & \mathbf{0}^T \\ \boldsymbol{c}\gamma^{-1/2} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \boldsymbol{C}_{11} - \boldsymbol{c}\gamma^{-1}\boldsymbol{c}^T \end{bmatrix} \begin{bmatrix} \gamma^{1/2} & \gamma^{-1/2}\boldsymbol{c}^T \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \quad \text{(A.20)}$$

$$= \boldsymbol{R}_0^T \boldsymbol{C}_1 \boldsymbol{R}_0, \quad \text{(A.21)}$$

which, through recursion, can be turned into

$$\boldsymbol{C} = \boldsymbol{R}_0^T \dots \boldsymbol{R}_{n-1}^T \boldsymbol{R}_{n-1} \dots \boldsymbol{R}_0 = \boldsymbol{R}^T \boldsymbol{R}. \quad \text{(A.22)}$$

Algorithm A.1 provides a more procedural definition, which can store the upper-triangular matrix $\boldsymbol{R}$ in the same space as $\boldsymbol{C}$, if desired. The total operation count for Cholesky factorization is $O(N^3)$ for a dense matrix but can be significantly lower for sparse matrices with low fill-in (Appendix A.4).

Note that Cholesky decomposition can also be applied to block-structured matrices, where the term γ in (A.19) is now a square block sub-matrix and $\boldsymbol{c}$ is a rectangular matrix (Golub and Van Loan 1996). The computation of square roots can be avoided by leaving the γ on the diagonal of the middle factor in (A.20), which results in the $\boldsymbol{C} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$ factorization, where $\boldsymbol{D}$ is a diagonal matrix. However, since square roots are relatively fast on modern computers, this is not worth the bother and Cholesky factorization is usually preferred.

A.2 Linear least squares

Least squares fitting problems are pervasive in computer vision. For example, the alignment of images based on matching feature points involves the minimization of a squared distance objective function (6.2),

$$E_{\text{LS}} = \sum_i \|\boldsymbol{r}_i\|^2 = \sum_i \|\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}_i'\|^2, \quad \text{(A.23)}$$

where

$$\boldsymbol{r}_i = \boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p}) - \boldsymbol{x}_i' = \hat{\boldsymbol{x}}_i' - \tilde{\boldsymbol{x}}_i' \quad \text{(A.24)}$$

is the *residual* between the measured location $\hat{\boldsymbol{x}}_i'$ and its corresponding current *predicted* location $\tilde{\boldsymbol{x}}_i' = \boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p})$. More complex versions of least squares problems, such as large-scale structure from motion (Section 7.4), may involve the minimization of functions of thousands of variables. Even problems such as image filtering (Section 3.4.3) and regularization (Section 3.7.1) may involve the minimization of sums of squared errors.

Figure A.2a shows an example of a simple least squares line fitting problem, where the quantities being estimated are the line equation parameters (m, b). When the sampled vertical values y_i are assumed to be noisy versions of points on the line $y = mx + b$, the optimal estimates for (m, b) can be found by minimizing the squared vertical residuals

$$E_{\text{VLS}} = \sum_i |y_i - (mx_i + b)|^2. \quad \text{(A.25)}$$

Note that the function being fitted need not itself be linear to use linear least squares. All that is required is that the function be linear in the unknown parameters. For example, polynomial fitting can be written as

$$E_{\text{PLS}} = \sum_i |y_i - (\sum_{j=0}^{p} a_j x_i^j)|^2, \quad \text{(A.26)}$$

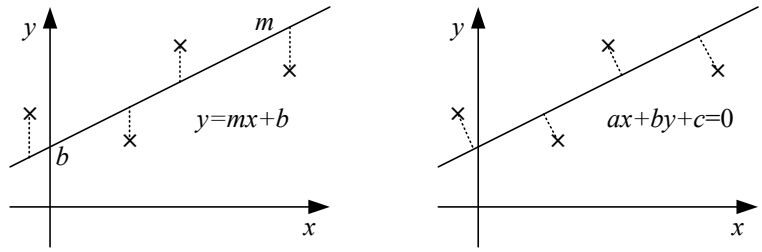


Figure A.2 Least squares regression. (a) The line $y = mx + b$ is fit to the four noisy data points, $\{(x_i, y_i)\}$, denoted by $\times$ by minimizing the squared vertical residuals between the data points and the line, $\sum_i \|y_i - (mx_i + b)\|^2$. (b) When the measurements $\{(x_i, y_i)\}$ are assumed to have noise in all directions, the sum of orthogonal squared distances to the line $\sum_i \|ax_i + by_i + c\|^2$ is minimized using total least squares.

while sinusoid fitting with unknown amplitude A and phase ϕ (but known frequency f) can be written as

$$E_{\text{SLS}} = \sum_i |y_i - A\sin(2\pi f x_i + \phi)|^2 = \sum_i |y_i - (B\sin 2\pi f x_i + C\cos 2\pi f x_i)|^2, \quad \text{(A.27)}$$

which is linear in (B, C).

In general, it is more common to denote the unknown parameters using $\boldsymbol{x}$ and to write the general form of linear least squares as[5]

$$E_{\text{LLS}} = \sum_i |\boldsymbol{a}_i \boldsymbol{x} - b_i|^2 = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2. \quad \text{(A.28)}$$

Expanding the above equation gives us

$$E_{\text{LLS}} = \boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{x} - 2\boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{b}) + \|\boldsymbol{b}\|^2, \quad \text{(A.29)}$$

whose minimum value for $\boldsymbol{x}$ can be found by solving the associated *normal equations* (Björck 1996; Golub and Van Loan 1996)

$$(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}. \quad \text{(A.30)}$$

The preferred way to solve the normal equations is to use Cholesky factorization. Let

$$\boldsymbol{C} = \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{R}^T\boldsymbol{R}, \quad \text{(A.31)}$$

where $\boldsymbol{R}$ is the upper-triangular Cholesky factor of the Hessian $\boldsymbol{C}$, and

$$\boldsymbol{d} = \boldsymbol{A}^T\boldsymbol{b}. \quad \text{(A.32)}$$

After factorization, the solution for $\boldsymbol{x}$ can be obtained as

$$\boldsymbol{R}^T\boldsymbol{z} = \boldsymbol{d}, \quad \boldsymbol{R}\boldsymbol{x} = \boldsymbol{z}, \quad \text{(A.33)}$$

[5] Be extra careful in interpreting the variable names here. In the 2D line-fitting example, x is used to denote the horizontal axis, but in the general least squares problem, $\boldsymbol{x} = (m, b)$ denotes the unknown parameter vector.

which involves the solution of two triangular systems, i.e., forward and backward substitution (Björck 1996).

In cases where the least squares problem is numerically poorly conditioned (which should generally be avoided by adding sufficient regularization or prior knowledge about the parameters, (Appendix A.3)), it is possible to use QR factorization or SVD directly on the matrix $\boldsymbol{A}$ (Björck 1996; Golub and Van Loan 1996; Trefethen and Bau 1997; Nocedal and Wright 2006; Björck and Dahlquist 2010), e.g.,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{R}\boldsymbol{x} = \boldsymbol{b} \quad \longrightarrow \quad \boldsymbol{R}\boldsymbol{x} = \boldsymbol{Q}^T\boldsymbol{b}. \tag{A.34}$$

Note that the upper triangular matrices $\boldsymbol{R}$ produced by the Cholesky factorization of $\boldsymbol{C} = \boldsymbol{A}^T\boldsymbol{A}$ and the QR factorization of $\boldsymbol{A}$ are the same, but that solving (A.34) is generally more stable (less sensitive to roundoff error) but slower (by a constant factor).

A.2.1 Total least squares

In some problems, e.g., when performing geometric line fitting in 2D images or 3D plane fitting to point cloud data, instead of having measurement error along one particular axis, the measured points have uncertainty in all directions, which is known as the *errors-in-variables* model (Van Huffel and Lemmerling 2002; Matei and Meer 2006). In this case, it makes more sense to minimize a set of homogeneous squared errors of the form

$$E_{\text{TLS}} = \sum_i (\boldsymbol{a}_i\boldsymbol{x})^2 = \|\boldsymbol{A}\boldsymbol{x}\|^2, \tag{A.35}$$

which is known as *total least squares* (TLS) (Van Huffel and Vandewalle 1991; Björck 1996; Golub and Van Loan 1996; Van Huffel and Lemmerling 2002).

The above error metric has a trivial minimum solution at $\boldsymbol{x} = 0$ and is, in fact, homogeneous in $\boldsymbol{x}$. For this reason, we augment this minimization problem with the requirement that $\|\boldsymbol{x}\|^2 = 1$. which results in the eigenvalue problem

$$\boldsymbol{x} = \arg\min_{\boldsymbol{x}} \boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{x} \quad \text{such that} \quad \|\boldsymbol{x}\|^2 = 1. \tag{A.36}$$

The value of $\boldsymbol{x}$ that minimizes this constrained problem is the eigenvector associated with the smallest eigenvalue of $\boldsymbol{A}^T\boldsymbol{A}$. This is the same as the last right singular vector of $\boldsymbol{A}$, since

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}, \tag{A.37}$$

$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}, \tag{A.38}$$

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{v}_k = \sigma_k^2, \tag{A.39}$$

which is minimized by selecting the smallest σ_k value.

Figure A.2b shows a line fitting problem where, in this case, the measurement errors are assumed to be isotropic in (x, y). The solution for the best line equation $ax + by + c = 0$ is found by minimizing

$$E_{\text{TLS}-2\text{D}} = \sum_i (ax_i + by_i + c)^2, \tag{A.40}$$

i.e., finding the eigenvector associated with the smallest eigenvalue of[6]

$$\boldsymbol{C} = \boldsymbol{A}^T\boldsymbol{A} = \sum_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}. \tag{A.41}$$

Notice, however, that minimizing $\sum_i (\boldsymbol{a}_i\boldsymbol{x})^2$ in (A.35) is only statistically optimal (Appendix B.1.1) if all of the measured terms in the $\boldsymbol{a}_i$, e.g., the $(x_i, y_i, 1)$ measurements, have equal noise. This is definitely not the case in the line-fitting example of Figure A.2b (A.40), since the 1 values are noise-free. To mitigate this, we first subtract the mean x and y values from all the measured points

$$\hat{x}_i = x_i - \bar{x} \tag{A.42}$$

$$\hat{y}_i = y_i - \bar{y} \tag{A.43}$$

and then fit the 2D line equation $a(x - \bar{x}) + b(y - \bar{y}) = 0$ by minimizing

$$E_{\mathrm{TLS-2Dm}} = \sum_i (a\hat{x}_i + b\hat{y}_i)^2. \tag{A.44}$$

The more general case where each individual measurement component can have different noise level, as is the case in estimating essential and fundamental matrices (Section 7.2), is called the *heteroscedastic* errors-in-variable (HEIV) model and is discussed by Matei and Meer (2006).

A.3 Non-linear least squares

In many vision problems, such as structure from motion, the least squares problem formulated in (A.23) involves functions $\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p})$ that are *not* linear in the unknown parameters $\boldsymbol{p}$. This problem is known as *non-linear least squares* or *non-linear regression* (Björck 1996; Madsen, Nielsen, and Tingleff 2004; Nocedal and Wright 2006). It is usually solved by iteratively re-linearizing (A.23) around the current estimate of $\boldsymbol{p}$ using the gradient derivative (Jacobian) $\boldsymbol{J} = \partial \boldsymbol{f}/\partial \boldsymbol{p}$ and computing an incremental improvement $\Delta\boldsymbol{p}$.

As shown in Equations (6.13–6.17), this results in

$$E_{\mathrm{NLS}}(\Delta\boldsymbol{p}) = \sum_i \|\boldsymbol{f}(\boldsymbol{x}_i; \boldsymbol{p} + \Delta\boldsymbol{p}) - \boldsymbol{x}_i'\|^2 \tag{A.45}$$

$$\approx \sum_i \|\boldsymbol{J}(\boldsymbol{x}_i; \boldsymbol{p})\Delta\boldsymbol{p} - \boldsymbol{r}_i\|^2, \tag{A.46}$$

where the Jacobians $\boldsymbol{J}(\boldsymbol{x}_i; \boldsymbol{p})$ and residual vectors $\boldsymbol{r}_i$ play the same role in forming the normal equations as $\boldsymbol{a}_i$ and b_i in (A.28).

Because the above approximation only holds near a local minimum or for small values of $\Delta\boldsymbol{p}$, the update $\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta\boldsymbol{p}$ may not always decrease the summed square residual error (A.45). One way to mitigate this problem is to take a smaller step,

$$\boldsymbol{p} \leftarrow \boldsymbol{p} + \alpha\Delta\boldsymbol{p}, \quad 0 < \alpha \leq 1. \tag{A.47}$$

[6] Again, be careful with the variable names here. The measurement equation is $\boldsymbol{a}_i = (x_i, y_i, 1)$ and the unknown parameters are $\boldsymbol{x} = (a, b, c)$.

A simple way to determine a reasonable value of α is to start with 1 and successively halve the value, which is a simple form of *line search* (Al-Baali and Fletcher. 1986; Björck 1996; Nocedal and Wright 2006).

Another approach to ensuring a downhill step in error is to add a diagonal damping term to the approximate Hessian

$$\boldsymbol{C} = \sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\boldsymbol{J}(\boldsymbol{x}_i), \tag{A.48}$$

i.e., to solve

$$[\boldsymbol{C} + \lambda \operatorname{diag}(\boldsymbol{C})]\Delta\boldsymbol{p} = \boldsymbol{d}, \tag{A.49}$$

where

$$\boldsymbol{d} = \sum_i \boldsymbol{J}^T(\boldsymbol{x}_i)\boldsymbol{r}_i, \tag{A.50}$$

which is called a *damped Gauss–Newton* method. The damping parameter λ is increased if the squared residual is not decreasing as fast as expected, i.e., as predicted by (A.46), and is decreased if the expected decrease is obtained (Madsen, Nielsen, and Tingleff 2004). The combination of the Newton (first-order Taylor series) approximation (A.46) and the adaptive damping parameter λ is commonly known as the Levenberg–Marquardt algorithm (Levenberg 1944; Marquardt 1963) and is an example of more general *trust region methods*, which are discussed in more detail in (Björck 1996; Conn, Gould, and Toint 2000; Madsen, Nielsen, and Tingleff 2004; Nocedal and Wright 2006).

When the initial solution is far away from its quadratic region of convergence around a local minimum, *large residual methods*, e.g., *Newton-type methods*, which add a second-order term to the Taylor series expansion in (A.46), may converge faster. Quasi-Newton methods such as BFGS, which require only gradient evaluations, can also be useful if memory size is an issue. Such techniques are discussed in textbooks and papers on numerical optimization (Toint 1987; Björck 1996; Conn, Gould, and Toint 2000; Nocedal and Wright 2006).

A.4 Direct sparse matrix techniques

Many optimization problems in computer vision, such as bundle adjustment (Szeliski and Kang 1994; Triggs, McLauchlan, Hartley *et al.* 1999; Hartley and Zisserman 2004; Snavely, Seitz, and Szeliski 2008b; Agarwal, Snavely, Simon *et al.* 2009) have Jacobian and (approximate) Hessian matrices that are extremely sparse (Section 7.4.1). For example, Figure 7.9a shows the *bipartite* model typical of structure from motion problems, in which most points are only observed by a subset of the cameras, which results in the sparsity patterns for the Jacobian and Hessian shown in Figure 7.9b–c.

Whenever the Hessian matrix is sparse enough, it is more efficient to use sparse Cholesky factorization instead of regular Cholesky factorization. In such sparse direct techniques, the Hessian matrix $\boldsymbol{C}$ and its associated Cholesky factor $\boldsymbol{R}$ are stored in *compressed form*, in which the amount of storage is proportional to the number of (potentially) non-zero entries (Björck 1996; Davis 2006).[7] Algorithms for computing the non-zero elements in $\boldsymbol{C}$ and $\boldsymbol{R}$

[7] For example, you can store a list of (i, j, c_{ij}) triples. One example of such a scheme is *compressed sparse row (CSR)* storage. An alternative storage method called *skyline*, which stores adjacent vertical spans of non-zero elements (Bathe 2007), is sometimes used in finite element analysis. Banded systems such as snakes (5.3) can store just the non-zero band elements (Björck 1996, Section 6.2) and can be solved in $O(nb^2)$, where n is the number of

from the sparsity pattern of the Jacobian matrix $\boldsymbol{J}$ are given by Björck (1996, Section 6.4), and algorithms for computing the numerical Cholesky and QR decompositions (once the sparsity pattern has been computed and storage allocated) are discussed by Björck (1996, Section 6.5).

A.4.1 Variable reordering

The key to efficiently solving sparse problems using direct (non-iterative) techniques is to determine an efficient *ordering* for the variables, which reduces the amount of *fill-in*, i.e., the number of non-zero entries in $\boldsymbol{R}$ that were zero in the original $\boldsymbol{C}$ matrix. We already saw in Section 7.4.1 how storing the more numerous 3D point parameters before the camera parameters and using the Schur complement (7.56) results in a more efficient algorithm. Similarly, sorting parameters by time in video-based reconstruction problems usually results in lower fill-in. Furthermore, any problem whose adjacency graph (the graph corresponding to the sparsity pattern) is a tree can be solved in linear time with an appropriate reordering of the variables (putting all the children before their parents). All of these are examples of good reordering techniques.

In the general case of unstructured data, there are many heuristics available to find good reorderings (Björck 1996; Davis 2006).[8] For general adjacency (sparsity) graphs, *minimum degree orderings* generally produce good results. For planar graphs, which often arise on image or spline grids (Section 8.3), *nested dissection*, which recursively splits the graph into two equal halves along a *frontier* (or boundary) of small size, generally works well. Such *domain decomposition* (or *multi-frontal*) techniques also enable the use of parallel processing, since independent sub-graphs can be processed in parallel on separate processors (Davis 2008).

The overall set of steps used to perform the direct solution of sparse least squares problems are summarized in Algorithm A.2, which is a modified version of Algorithm 6.6.1 by Björck (1996, Section 6.6)). If a series of related least squares problems is being solved, as is the case in iterative non-linear least squares (Appendix A.3), steps 1–3 can be performed ahead of time and reused for each new invocation with different $\boldsymbol{C}$ and $\boldsymbol{d}$ values. When the problem is block-structured, as is the case in structure from motion where point (structure) variables have dense 3×3 sub-entries in $\boldsymbol{C}$ and cameras have 6×6 (or larger) entries, the cost of performing the reordering computation is small compared to the actual numerical factorization, which can benefit from block-structured matrix operations (Golub and Van Loan 1996). It is also possible to apply sparse reordering and multifrontal techniques to QR factorization (Davis 2008), which may be preferable when the least squares problems are poorly conditioned.

A.5 Iterative techniques

When problems become large, the amount of memory required to store the Hessian matrix $\boldsymbol{C}$ and its factor $\boldsymbol{R}$, and the amount of time it takes to compute the factorization, can become prohibitively large, especially when there are large amounts of fill-in. This is often the case with image processing problems defined on pixel grids, since, even with the optimal reordering (nested dissection) the amount of fill can still be large.

variables and b is the bandwidth.

[8] Finding the optimal reordering with minimal fill-in is provably NP-hard.

procedure *SparseCholeskySolve*($\boldsymbol{C}, \boldsymbol{d}$):

1. Determine symbolically the structure of $\boldsymbol{C}$, i.e., the adjacency graph.
2. (Optional) Compute a reordering for the variables, taking into account any block structure inherent in the problem.
3. Determine the fill-in pattern for $\boldsymbol{R}$ and allocate the compressed storage for $\boldsymbol{R}$ as well as storage for the permuted right hand side $\hat{\boldsymbol{d}}$.
4. Copy the elements of $\boldsymbol{C}$ and $\boldsymbol{d}$ into $\boldsymbol{R}$ and $\hat{\boldsymbol{d}}$, permuting the values according to the computed ordering.
5. Perform the numerical factorization of $\boldsymbol{R}$ using Algorithm A.1.
6. Solve the factored system (A.33), i.e.,
$$\boldsymbol{R}^T \boldsymbol{z} = \hat{\boldsymbol{d}}, \quad \boldsymbol{R}\boldsymbol{x} = \boldsymbol{z}.$$
7. Return the solution $\boldsymbol{x}$, after undoing the permutation.

Algorithm A.2 Sparse least squares using a sparse Cholesky decomposition of the matrix $\boldsymbol{C}$.

A preferable approach to solving such linear systems is to use iterative techniques, which compute a series of estimates that converge to the final solution, e.g., by taking a series of downhill steps in an energy function such as (A.29).

A large number of iterative techniques have been developed over the years, including such well-known algorithms as successive overrelaxation and multi-grid. These are described in specialized textbooks on iterative solution techniques (Axelsson 1996; Saad 2003) as well as in more general books on numerical linear algebra and least squares techniques (Björck 1996; Golub and Van Loan 1996; Trefethen and Bau 1997; Nocedal and Wright 2006; Björck and Dahlquist 2010).

A.5.1 Conjugate gradient

The iterative solution technique that often performs best is conjugate gradient descent, which takes a series of downhill steps that are *conjugate* to each other with respect to the $\boldsymbol{C}$ matrix, i.e., if the $\boldsymbol{u}$ and $\boldsymbol{v}$ descent directions satisfy $\boldsymbol{u}^T\boldsymbol{C}\boldsymbol{v} = 0$. In practice, conjugate gradient descent outperforms other kinds of gradient descent algorithm because its convergence rate is proportional to the square root of the *condition number* of $\boldsymbol{C}$ instead of the condition number itself.[9] Shewchuk (1994) provides a nice introduction to this topic, with clear intuitive explanations of the reasoning behind the conjugate gradient algorithm and its performance.

Algorithm A.3 describes the conjugate gradient algorithm and its related least squares counterpart, which can be used when the original set of least squares linear equations are

[9] The condition number $\kappa(\boldsymbol{C})$ is the ratio of the largest and smallest eigenvalues of $\boldsymbol{C}$. The actual convergence rate depends on the clustering of the eigenvalues, as discussed in the references cited in this section.

ConjugateGradient$(\boldsymbol{C}, \boldsymbol{d}, \boldsymbol{x}_0)$

1. $\boldsymbol{r}_0 = \boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}_0$
2. $\boldsymbol{p}_0 = \boldsymbol{r}_0$
3. **for** $k = 0 \ldots$
4. $\quad \boldsymbol{w}_k = \boldsymbol{C}\boldsymbol{p}_k$
5. $\quad \alpha_k = \|\boldsymbol{r}_k\|^2 / (\boldsymbol{p}_k \cdot \boldsymbol{w}_k)$
6. $\quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
7. $\quad \boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{w}_k$
8.
9. $\quad \beta_{k+1} = \|\boldsymbol{r}_{k+1}\|^2 / \|\boldsymbol{r}_k\|^2$
10. $\quad \boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k$

ConjugateGradientLS$(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}_0)$

1. $\boldsymbol{q}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0, \quad \boldsymbol{r}_0 = \boldsymbol{A}^T \boldsymbol{q}_0$
2. $\boldsymbol{p}_0 = \boldsymbol{r}_0$
3. **for** $k = 0 \ldots$
4. $\quad \boldsymbol{v}_k = \boldsymbol{A}\boldsymbol{p}_k$
5. $\quad \alpha_k = \|\boldsymbol{r}_k\|^2 / \|\boldsymbol{v}_k\|^2$
6. $\quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
7. $\quad \boldsymbol{q}_{k+1} = \boldsymbol{q}_k - \alpha_k \boldsymbol{v}_k$
8. $\quad \boldsymbol{r}_{k+1} = \boldsymbol{A}^T \boldsymbol{q}_{k+1}$
9. $\quad \beta_{k+1} = \|\boldsymbol{r}_{k+1}\|^2 / \|\boldsymbol{r}_k\|^2$
10. $\quad \boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k$

Algorithm A.3 Conjugate gradient and conjugate gradient least squares algorithms. The algorithm is described in more detail in the text, but in brief, they choose descent directions $\boldsymbol{p}_k$ that are conjugate to each other with respect to $\boldsymbol{C}$ by computing a factor β by which to discount the previous search direction $\boldsymbol{p}_{k-1}$. They then find the optimal step size α and take a downhill step by an amount $\alpha_k \boldsymbol{p}_k$.

available in the form of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ (A.28). While it is easy to convince yourself that the two forms are mathematically equivalent, the least squares form is preferable if rounding errors start to affect the results because of poor conditioning. It may also be preferable if, due to the sparsity structure of $\boldsymbol{A}$, multiplies with the original $\boldsymbol{A}$ matrix are faster or more space efficient than multiplies with $\boldsymbol{C}$.

The conjugate gradient algorithm starts by computing the current residual $\boldsymbol{r}_0 = \boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}_0$, which is the direction of steepest descent of the energy function (A.28). It sets the original descent direction $\boldsymbol{p}_0 = \boldsymbol{r}_0$. Next, it multiplies the descent direction by the quadratic form (Hessian) matrix $\boldsymbol{C}$ and combines this with the residual to estimate the optimal step size α_k. The solution vector $\boldsymbol{x}_k$ and the residual vector $\boldsymbol{r}_k$ are then updated using this step size. (Notice how the least squares variant of the conjugate gradient algorithm splits the multiplication by the $\boldsymbol{C} = \boldsymbol{A}^T\boldsymbol{A}$ matrix across steps 4 and 8.) Finally, a new search direction is calculated by first computing a factor β as the ratio of current to previous residual magnitudes. The new search direction $\boldsymbol{p}_{k+1}$ is then set to the residual plus β times the old search direction $\boldsymbol{p}_k$, which keeps the directions conjugate with respect to $\boldsymbol{C}$.

It turns out that conjugate gradient descent can also be directly applied to non-quadratic energy functions, e.g., those arising from non-linear least squares (Appendix A.3). Instead of explicitly forming a local quadratic approximation $\boldsymbol{C}$ and then computing residuals $\boldsymbol{r}_k$, non-linear conjugate gradient descent computes the gradient of the energy function E (A.45) directly inside each iteration and uses it to set the search direction (Nocedal and Wright 2006). Since the quadratic approximation to the energy function may not exist or may be inaccurate,

line search is often used to determine the step size α_k. Furthermore, to compensate for errors in finding the true function minimum, alternative formulas for β_{k+1} such as Polak–Ribière,

$$\beta_{k+1} = \frac{\nabla E(\boldsymbol{x}_{k+1})[\nabla E(\boldsymbol{x}_{k+1}) - \nabla E(\boldsymbol{x}_k)]}{\|\nabla E(\boldsymbol{x}_k)\|^2} \tag{A.51}$$

are often used (Nocedal and Wright 2006).

A.5.2 Preconditioning

As we mentioned previously, the rate of convergence of the conjugate gradient algorithm is governed in large part by the condition number $\kappa(\boldsymbol{C})$. Its effectiveness can therefore be increased dramatically by reducing this number, e.g., by rescaling elements in $\boldsymbol{x}$, which corresponds to rescaling rows and columns in $\boldsymbol{C}$.

In general, preconditioning is usually thought of as a change of basis from the vector $\boldsymbol{x}$ to a new vector

$$\hat{x} = \boldsymbol{S}\boldsymbol{x}. \tag{A.52}$$

The corresponding linear system being solved then becomes

$$\boldsymbol{A}\boldsymbol{S}^{-1}\hat{x} = \boldsymbol{S}^{-1}\boldsymbol{b} \quad \text{or} \quad \hat{\boldsymbol{A}}\hat{x} = \hat{\boldsymbol{b}}, \tag{A.53}$$

with a corresponding least squares energy (A.29) of the form

$$E_{\text{PLS}} = \hat{x}^T(\boldsymbol{S}^{-T}\boldsymbol{C}\boldsymbol{S}^{-1})\hat{x} - 2\hat{x}^T(\boldsymbol{S}^{-T}\boldsymbol{d}) + \|\hat{\boldsymbol{b}}\|^2. \tag{A.54}$$

The actual preconditioned matrix $\hat{\boldsymbol{C}} = \boldsymbol{S}^{-T}\boldsymbol{C}\boldsymbol{S}^{-1}$ is usually not explicitly computed. Instead, Algorithm A.3 is extended to insert $\boldsymbol{S}^{-T}$ and $\boldsymbol{S}^T$ operations at the appropriate places (Björck 1996; Golub and Van Loan 1996; Trefethen and Bau 1997; Saad 2003; Nocedal and Wright 2006).

A good preconditioner $\boldsymbol{S}$ is easy and cheap to compute, but is also a decent approximation to a square root of $\boldsymbol{C}$, so that $\kappa(\boldsymbol{S}^{-T}\boldsymbol{C}\boldsymbol{S}^{-1})$ is closer to 1. The simplest such choice is the square root of the diagonal matrix $\boldsymbol{S} = \boldsymbol{D}^{1/2}$, with $\boldsymbol{D} = \text{diag}(\boldsymbol{C})$. This has the advantage that any scalar change in variables (e.g., using radians instead of degrees for angular measurements) has no effect on the range of convergence of the iterative technique. For problems that are naturally block-structured, e.g., for structure from motion, where 3D point positions or 6D camera poses are being estimated, a block diagonal preconditioner is often a good choice.

A wide variety of more sophisticated preconditioners have been developed over the years (Björck 1996; Golub and Van Loan 1996; Trefethen and Bau 1997; Saad 2003; Nocedal and Wright 2006), many of which can be directly applied to problems in computer vision (Byröd and øAström 2009; Jeong, Nistér, Steedly *et al.* 2010; Agarwal, Snavely, Seitz *et al.* 2010). Some of these are based on an *incomplete Cholesky* factorization of $\boldsymbol{C}$, i.e., one in which the amount of fill-in in $\boldsymbol{R}$ is strictly limited, e.g., to just the original non-zero elements in $\boldsymbol{C}$.[10] Other preconditioners are based on a sparsified, e.g., tree-based or clustered, approximation to $\boldsymbol{C}$ (Koutis 2007; Koutis and Miller 2008; Grady 2008; Koutis, Miller, and Tolliver 2009), since these are known to have efficient inversion properties.

[10] If a complete Cholesky factorization $\boldsymbol{C} = \boldsymbol{R}^T\boldsymbol{R}$ is used, we get $\hat{\boldsymbol{C}} = \boldsymbol{R}^{-T}\boldsymbol{C}\boldsymbol{R}^{-1} = \boldsymbol{I}$ and all iterative algorithms converge in a single step, thereby obviating the need to use them, but the complete factorization is often too expensive. Note that incomplete factorization can also benefit from reordering.

For grid-based image-processing applications, *parallel* or *hierarchical* preconditioners often perform extremely well (Yserentant 1986; Szeliski 1990b; Pentland 1994; Saad 2003; Szeliski 2006b). These approaches use a change of basis transformation $\boldsymbol{S}$ that resembles the pyramidal or wavelet representations discussed in Section 3.5, and are hence amenable to parallel and GPU-based implementations. Coarser elements in the new representation quickly converge to the low-frequency components in the solution, while finer-level elements encode the higher-frequency components. Some of the relationships between hierarchical preconditioners, incomplete Cholesky factorization, and multigrid techniques are explored by Saad (2003) and Szeliski (2006b).

A.5.3 Multigrid

One other class of iterative techniques widely used in computer vision is *multigrid* techniques (Briggs, Henson, and McCormick 2000; Trottenberg, Oosterlee, and Schuller 2000), which have been applied to problems such as surface interpolation (Terzopoulos 1986a), optical flow (Terzopoulos 1986a; Bruhn, Weickert, Kohlberger *et al.* 2006), high dynamic range tone mapping (Fattal, Lischinski, and Werman 2002), colorization (Levin, Lischinski, and Weiss 2004), natural image matting (Levin, Lischinski, and Weiss 2008), and segmentation (Grady 2008).

The main idea behind multigrid is to form coarser (lower-resolution) versions of the problems and use them to compute the low-frequency components of the solution. However, unlike simple coarse-to-fine techniques, which use the coarse solutions to initialize the fine solution, multigrid techniques only *correct* the low-frequency component of the current solution and use multiple rounds of coarsening and refinement (in what are often called "V" and "W" patterns of motion across the pyramid) to obtain rapid convergence.

On certain simple homogeneous problems (such as solving Poisson equations), multigrid techniques can achieve optimal performance, i.e., computation times linear in the number of variables. However, for more inhomogeneous problems or problems on irregular grids, variants on these techniques, such as *algebraic multigrid* (AMG) approaches, which look at the structure of $\boldsymbol{C}$ to derive coarse level problems, may be preferable. Saad (2003) has a nice discussion of the relationship between multigrid and parallel preconditioners and on the relative merits of using multigrid or conjugate gradient approaches.

Appendix B

Bayesian modeling and inference

B.1 Estimation theory 662
B.1.1 Likelihood for multivariate Gaussian noise 663
B.2 Maximum likelihood estimation and least squares 665
B.3 Robust statistics 666
B.4 Prior models and Bayesian inference 667
B.5 Markov random fields 668
B.5.1 Gradient descent and simulated annealing 670
B.5.2 Dynamic programming 670
B.5.3 Belief propagation 672
B.5.4 Graph cuts 674
B.5.5 Linear programming 676
B.6 Uncertainty estimation (error analysis) 678

The following problem commonly recurs in this book: Given a number of measurements (images, feature positions, etc.), estimate the values of some unknown structure or parameter (camera positions, object shape, etc.). These kinds of problems are in general called *inverse* problems because they involve estimating unknown model parameters instead of simulating the forward formation equations.[1] Computer graphics is a classic forward modeling problem (given some objects, cameras, and lighting, simulate the images that would result), while computer vision problems are usually of the inverse kind (given one or more images, recover the scene that gave rise to these images).

Given an instance of an inverse problem, there are, in general, several ways to proceed. For instance, through clever (or sometimes straightforward) algebraic manipulation, a closed form solution for the unknowns can sometimes be derived. Consider, for example, the *camera matrix calibration* problem (Section 6.2.1): given an image of a calibration pattern consisting of known 3D point positions, compute the 3×4 camera matrix $\boldsymbol{P}$ that maps these points onto the image plane.

In more detail, we can write this problem as (6.33–6.34)

$$x_i = \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}} \tag{B.1}$$

$$y_i = \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}}, \tag{B.2}$$

where (x_i, y_i) is the feature position of the ith point measured in the image plane, (X_i, Y_i, Z_i) is the corresponding 3D point position, and the p_{ij} are the unknown entries of the camera matrix $\boldsymbol{P}$. Moving the denominator over to the left hand side, we end up with a set of simultaneous linear equations,

$$x_i(p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}) = p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}, \tag{B.3}$$

$$y_i(p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}) = p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}, \tag{B.4}$$

which we can solve using linear least squares (Appendix A.2) to obtain an estimate of $\boldsymbol{P}$.

The question then arises: is this set of equations the right ones to be solving? If the measurements are totally noise-free or we do not care about getting the best possible answer, then the answer is yes. However, in general, we cannot be sure that we have a reasonable algorithm unless we make a model of the likely sources of error and devise an algorithm that performs as well as possible given these potential errors.

B.1 Estimation theory

The study of such inference problems from noisy data is often called *estimation theory* (Gelb 1974), and its extension to problems where we explicitly choose a loss function is called *statistical decision theory* (Berger 1993; Hastie, Tibshirani, and Friedman 2001; Bishop 2006; Robert 2007). We first start by writing down the forward process that leads from our unknowns (and knowns) to a set of noise-corrupted measurements. We then devise an algorithm that will give us an estimate (or set of estimates) that are both insensitive to the noise (as best they can be) and also quantify the reliability of these estimates.

[1] In machine learning, these problems are called *regression problems*, because we are trying to estimate a *continuous* quantity from noisy inputs, as opposed to a discrete *classification* task (Bishop 2006).

The specific equations above (B.1) are just a particular instance of a more general set of *measurement equations*,

$$\boldsymbol{y}_i = \boldsymbol{f}_i(\boldsymbol{x}) + \boldsymbol{n}_i. \tag{B.5}$$

Here, the $\boldsymbol{y}_i$ are the noise-corrupted *measurements*, e.g., (x_i, y_i) in Equation (B.1), and $\boldsymbol{x}$ is the unknown *state vector*.[2]

Each measurement comes with its associated *measurement model* $\boldsymbol{f}_i(\boldsymbol{x})$, which maps the unknown into that particular measurement. An alternative formulation would be to have one general function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}_i)$ and to use a per-measurement parameter vector $\boldsymbol{p}_i$ to distinguish between different measurements, e.g., (X_i, Y_i, Z_i) in Equation (B.1). Note that the use of the $\boldsymbol{f}_i(\boldsymbol{x})$ form makes it straightforward to have measurements of different dimensions, which becomes useful when we start adding in prior information (Appendix B.4).

Each measurement is also contaminated with some noise $\boldsymbol{n}_i$. In Equation (B.5), we have indicated that $\boldsymbol{n}_i$ is a zero-mean normal (Gaussian) random variable with a covariance matrix $\boldsymbol{\Sigma}_i$. In general, the noise need not be Gaussian and, in fact, it is usually prudent to assume that some measurements may be outliers. However, we defer this discussion to Appendix B.3, after we have explored the simpler Gaussian noise case more fully. We also assume that the noise vectors $\boldsymbol{n}_i$ are independent. In the case where they are not (e.g., when some constant gain or offset contaminates all of the pixels in a given image), we can add this effect as a *nuisance parameter* to our state vector $\boldsymbol{x}$ and later estimate its value (and discard it, if so desired).

B.1.1 Likelihood for multivariate Gaussian noise

Given all of the noisy measurements $\boldsymbol{y} = \{\boldsymbol{y}_i\}$, we would like to infer a probability distribution on the unknown $\boldsymbol{x}$ vector. We can write the *likelihood* of having observed the $\{\boldsymbol{y}_i\}$ given a particular value of $\boldsymbol{x}$ as

$$L = p(\boldsymbol{y}|\boldsymbol{x}) = \prod_i p(\boldsymbol{y}_i|\boldsymbol{x}) = \prod_i p(\boldsymbol{y}_i|\boldsymbol{f}_i(\boldsymbol{x})) = \prod_i p(\boldsymbol{n}_i). \tag{B.6}$$

When each noise vector $\boldsymbol{n}_i$ is a multivariate Gaussian with covariance $\boldsymbol{\Sigma}_i$,

$$\boldsymbol{n}_i \sim \mathcal{N}(0, \boldsymbol{\Sigma}_i), \tag{B.7}$$

we can write this likelihood as

$$\begin{aligned} L &= \prod_i |2\pi\boldsymbol{\Sigma}_i|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{x}))^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{x}))\right) \\ &= \prod_i |2\pi\boldsymbol{\Sigma}_i|^{-1/2} \exp\left(-\frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{x})\|^2_{\boldsymbol{\Sigma}_i^{-1}}\right), \end{aligned} \tag{B.8}$$

where the matrix norm $\|\boldsymbol{x}\|^2_{\boldsymbol{A}}$ is a shorthand notation for $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$.

The norm $\|\boldsymbol{y}_i - \overline{\boldsymbol{y}}_i\|_{\boldsymbol{\Sigma}_i^{-1}}$ is often called the *Mahalanobis distance* (5.26 and 14.14) and is used to measure the distance between a measurement and the mean of a multivariate Gaussian distribution. Contours of equal Mahalanobis distance are equi-probability contours. Note

[2] In the Kalman filtering literature (Gelb 1974), it is more common to use $\boldsymbol{z}$ instead of $\boldsymbol{y}$ to denote measurements.

that when the measurement covariance is isotropic (the same in all directions), i.e., when $\boldsymbol{\Sigma}_i = \sigma_i^2 \boldsymbol{I}$, the likelihood can be written as

$$L = \prod_i (2\pi\sigma_i^2)^{-N_i/2} \exp\left(-\frac{1}{2\sigma_i^2}\|\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{x})\|^2\right), \tag{B.9}$$

where N_i is the length of the ith measurement vector $\boldsymbol{y}_i$.

We can more easily visualize the structure of the covariance matrix and the corresponding Mahalanobis distance if we first perform an *eigenvalue* or *principal component* analysis (PCA) of the covariance matrix (A.6),

$$\boldsymbol{\Sigma} = \boldsymbol{\Phi}\, \mathrm{diag}(\lambda_0 \ldots \lambda_{N-1})\, \boldsymbol{\Phi}^T. \tag{B.10}$$

Equal-probability contours of the corresponding multi-variate Gaussian, which are also equi-distance contours in the Mahalanobis distance (Figure 14.14), are multi-dimensional ellipsoids whose axis directions are given by the columns of $\boldsymbol{\Phi}$ (the *eigenvectors*) and whose lengths are given by the $\sigma_j = \sqrt{\lambda_j}$ (Figure A.1).

It is usually more convenient to work with the negative log likelihood, which we can think of as a *cost* or *energy*

$$E = -\log L = \frac{1}{2}\sum_i (\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{x}))^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{x})) + k \tag{B.11}$$

$$= \frac{1}{2}\sum_i \|\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{x})\|^2_{\boldsymbol{\Sigma}_i^{-1}} + k, \tag{B.12}$$

where $k = \sum_i \log |2\pi\boldsymbol{\Sigma}_i|$ is a constant that depends on the measurement variances, but is independent of $\boldsymbol{x}$.

Notice that the inverse covariance $\boldsymbol{C}_i = \boldsymbol{\Sigma}_i^{-1}$ plays the role of a *weight* on each of the measurement error *residuals*, i.e., the difference between the contaminated measurement $\boldsymbol{y}_i$ and its uncontaminated (predicted) value $\boldsymbol{f}_i(\boldsymbol{x})$. In fact, the inverse covariance is often called the (Fisher) *information matrix* (Bishop 2006), since it tells us how much information is contained in a given measurement, i.e., how well it constrains the final estimate. We can also think of this matrix as denoting the amount of *confidence* to associate with each measurement (hence the letter $\boldsymbol{C}$).

In this formulation, it is quite acceptable for some information matrices to be singular (of degenerate rank) or even zero (if the measurement is missing altogether). Rank-deficient measurements often occur, for example, when using a line feature or edge to measure a 3D edge-like feature, since its exact position along the edge is unknown (of infinite or extremely large variance) §8.1.3.

In order to make the distinction between the noise contaminated measurement and its expected value for a particular setting of $\boldsymbol{x}$ more explicit, we adopt the notation $\tilde{\boldsymbol{y}}$ for the former (think of the tilde as the approximate or noisy value) and $\hat{\boldsymbol{y}} = \boldsymbol{f}_i(\boldsymbol{x})$ for the latter (think of the hat as the predicted or expected value). We can then write the negative log likelihood as

$$E = -\log L = \sum_i \|\tilde{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_i\|_{\boldsymbol{\Sigma}_i^{-1}} + k. \tag{B.13}$$

B.2 Maximum likelihood estimation and least squares

Now that we have presented the likelihood and log likelihood functions, how can we find the optimal value for our state estimate $\boldsymbol{x}$? One plausible choice might be to select the value of $\boldsymbol{x}$ that maximizes $L = p(\boldsymbol{y}|\boldsymbol{x})$. In fact, in the absence of any prior model for $\boldsymbol{x}$ (Appendix B.4), we have

$$L = p(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{y}, \boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y}).$$

Therefore, choosing the value of $\boldsymbol{x}$ that maximizes the likelihood is equivalent to choosing the maximum of our probability density estimate for $\boldsymbol{x}$.

When might this be a good idea? If the data (measurements) constrain the possible values of $\boldsymbol{x}$ so that they all cluster tightly around one value (e.g., if the distribution $p(\boldsymbol{x}|\boldsymbol{y})$ is a unimodal Gaussian), the maximum likelihood estimate is the optimal one in that it is both unbiased and has the least possible variance. In many other cases, e.g., if a single estimate is all that is required, it is still often the best estimate.[3] However, if the probability is multi-modal, i.e., it has several local minima in the log likelihood (Figure 5.7), much more care may be required. In particular, it might be necessary to defer certain decisions (such as the ultimate position of an object being tracked) until more measurements have been taken. The CONDENSATION algorithm presented in Section 5.1.2 is one possible method for modeling and updating such multi-modal distributions but is just one example of more general *particle filtering* and *Markov Chain Monte Carlo* (MCMC) techniques (Andrieu, de Freitas, Doucet *et al.* 2003; Bishop 2006; Koller and Friedman 2009).

Another possible way to choose the best estimate is to maximize the *expected utility* (or, conversely, to minimize the expected risk or loss) associated with obtaining the correct estimate, i.e., by minimizing

$$E_{\text{loss}}(\boldsymbol{x}, \boldsymbol{y}) = \int l(\boldsymbol{x} - \boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{y}) d\boldsymbol{z}. \tag{B.14}$$

For example, if a robot wants to avoid hitting a wall at all costs, the loss function will be high whenever the estimate underestimates the true distance to the wall. When $l(\boldsymbol{x} - \boldsymbol{y}) = \delta(\boldsymbol{x} - \boldsymbol{y})$, we obtain the maximum likelihood estimate, whereas when $l(\boldsymbol{x} - \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2$, we obtain the *mean square error* (MSE) or *expected value* estimate. The explicit modeling of a utility or loss function is what characterizes *statistical decision theory* (Berger 1993; Hastie, Tibshirani, and Friedman 2001; Bishop 2006; Robert 2007).

How do we find the maximum likelihood estimate? If the measurement noise is Gaussian, we can minimize the quadratic objective function (B.13). This becomes even simpler if the measurement equations are linear, i.e.,

$$\boldsymbol{f}_i(\boldsymbol{x}) = \boldsymbol{H}_i \boldsymbol{x}, \tag{B.15}$$

where $\boldsymbol{H}$ is the *measurement matrix* relating unknown state variables $\boldsymbol{x}$ to measurements $\tilde{\boldsymbol{y}}$. In this case, (B.13) becomes

$$E = \sum_i \|\tilde{\boldsymbol{y}}_i - \boldsymbol{H}_i \boldsymbol{x}\|_{\boldsymbol{\Sigma}_i^{-1}} = \sum_i (\tilde{\boldsymbol{y}}_i - \boldsymbol{H}_i \boldsymbol{x})^T \boldsymbol{C}_i (\tilde{\boldsymbol{y}}_i - \boldsymbol{H}_i \boldsymbol{x}), \tag{B.16}$$

[3] According to the Gauss-Markov theorem, least squares produces the best linear unbiased estimator (BLUE) for a linear measurement model regardless of the actual noise distribution, assuming that the noise is zero mean and uncorrelated.

which is a simple quadratic form in $\boldsymbol{x}$, which can be solved using linear least squares (Appendix A.2). When the measurements are non-linear, the system must be solved iteratively using non-linear least squares (Appendix A.3).

B.3 Robust statistics

In Appendix B.1.1, we assumed that the noise being added to each measurement (B.5) was multivariate Gaussian (B.7). This is an appropriate model if the noise is the result of lots of tiny errors being added together, e.g., from thermal noise in a silicon imager. In most cases, however, measurements can be contaminated with larger *outliers*, i.e., gross failures in the measurement process. Examples of such outliers include bad feature matches (Section 6.1.4), occlusions in stereo matching (Chapter 11), and discontinuities in an otherwise smooth image, depth map, or label image (Sections 3.7.1 and 3.7.2).

In such cases, it makes more sense to model the measurement noise with a long-tailed *contaminated* noise model such as a Laplacian. The negative log likelihood in this case, rather than being quadratic in the measurement residuals (B.12–B.16), has a slower growth in the penalty function to account for the increased likelihood of large errors.

This formulation of the inference problem is called an *M-estimator* in the robust statistics literature (Huber 1981; Hampel, Ronchetti, Rousseeuw *et al.* 1986; Black and Rangarajan 1996; Stewart 1999) and involves applying a robust penalty function $\rho(r)$ to the residuals

$$E_{\mathrm{RLS}}(\Delta \boldsymbol{p}) = \sum_i \rho(\|\boldsymbol{r}_i\|) \tag{B.17}$$

instead of squaring them.

As we mentioned in Section 6.1.4, we can take the derivative of this function with respect to $\boldsymbol{p}$ and set it to 0,

$$\sum_i \psi(\|\boldsymbol{r}_i\|) \frac{\partial \|\boldsymbol{r}_i\|}{\partial \boldsymbol{p}} = \sum_i \frac{\psi(\|\boldsymbol{r}_i\|)}{\|\boldsymbol{r}_i\|} \boldsymbol{r}_i^T \frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{p}} = 0, \tag{B.18}$$

where $\psi(r) = \rho'(r)$ is the derivative of ρ and is called the *influence function*. If we introduce a *weight function*, $w(r) = \Psi(r)/r$, we observe that finding the stationary point of (B.17) using (B.18) is equivalent to minimizing the *iteratively re-weighted least squares* (IRLS) problem

$$E_{\mathrm{IRLS}} = \sum_i w(\|\boldsymbol{r}_i\|) \|\boldsymbol{r}_i\|^2, \tag{B.19}$$

where the $w(\|\boldsymbol{r}_i\|)$ play the same local weighting role as $\boldsymbol{C}_i = \boldsymbol{\Sigma}_i^{-1}$ in (B.12). Black and Anandan (1996) describe a variety of robust penalty functions and their corresponding influence and weighting function.

The IRLS algorithm alternates between computing the influence functions $w(\|\boldsymbol{r}_i\|)$ and solving the resulting weighted least squares problem (with fixed w values). Alternative incremental robust least squares algorithms can be found in the work of Sawhney and Ayer (1996); Black and Anandan (1996); Black and Rangarajan (1996); Baker, Gross, Ishikawa *et al.* (2003) and textbooks and tutorials on robust statistics (Huber 1981; Hampel, Ronchetti, Rousseeuw *et al.* 1986; Rousseeuw and Leroy 1987; Stewart 1999). It is also possible to apply general optimization techniques (Appendix A.3) directly to the non-linear cost function given in Equation (B.19), which may sometimes have better convergence properties.

Most robust penalty functions involve a scale parameter, which should typically be set to the variance (or standard deviation, depending on the formulation) of the non-contaminated (inlier) noise. Estimating such noise levels directly from the measurements or their residuals, however, can be problematic, as such estimates themselves become contaminated by outliers. The robust statistics literature contains a variety of techniques to estimate such parameters. One of the simplest and most effective is the *median absolute deviation* (MAD),

$$MAD = \text{med}_i \|\boldsymbol{r}_i\|, \tag{B.20}$$

which, when multiplied by 1.4, provides a robust estimate of the standard deviation of the inlier noise process.

As mentioned in Section 6.1.4, it is often better to start iterative non-linear minimization techniques, such as IRLS, in the vicinity of a good solution by first randomly selecting small subsets of measurements until a good set of inliers is found. The best known of these techniques is RANdom SAmple Consensus (RANSAC) (Fischler and Bolles 1981), although even better variants such as Preemptive RANSAC (Nistér 2003) and PROgressive SAmple Consensus (PROSAC) (Chum and Matas 2005) have since been developed.

B.4 Prior models and Bayesian inference

While maximum likelihood estimation can often lead to good solutions, in some cases the range of possible solutions consistent with the measurements is too large to be useful. For example, consider the problem of image denoising (Sections 3.4.4 and 3.7.3). If we estimate each pixel separately based on just its noisy version, we cannot make any progress, as there are a large number of values that could lead to each noisy measurement.[4] Instead, we need to rely on typical properties of images, e.g., that they tend to be piecewise smooth (Section 3.7.1).

The propensity of images to be piecewise smooth can be encoded in a *prior distribution* $p(\boldsymbol{x})$, which measures the likelihood of an image being a natural image. For example, to encode piecewise smoothness, we can use a *Markov random field* model (3.109 and B.24) whose negative log likelihood is proportional to a robustified measure of image smoothness (gradient magnitudes).

Prior models need not be restricted to image processing applications. For example, we may have some external knowledge about the rough dimensions of an object being scanned, the focal length of a lens being calibrated, or the likelihood that a particular object might appear in an image. All of these are examples of prior distributions or probabilities and they can be used to produce more reliable estimates.

As we have already seen in (3.68) and (3.106), Bayes' Rule states that a *posterior* distribution $p(\boldsymbol{x}|\boldsymbol{y})$ over the unknowns $\boldsymbol{x}$ given the measurements $\boldsymbol{y}$ can be obtained by multiplying the measurement likelihood $p(\boldsymbol{y}|\boldsymbol{x})$ by the prior distribution $p(\boldsymbol{x})$,

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}, \tag{B.21}$$

where $p(\boldsymbol{y}) = \int_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$ is a normalizing constant used to make the $p(\boldsymbol{x}|\boldsymbol{y})$ distribution *proper* (integrate to 1). Taking the negative logarithm of both sides of Equation (B.21), we

[4] In fact, the maximum likelihood estimate is just the noisy image itself.

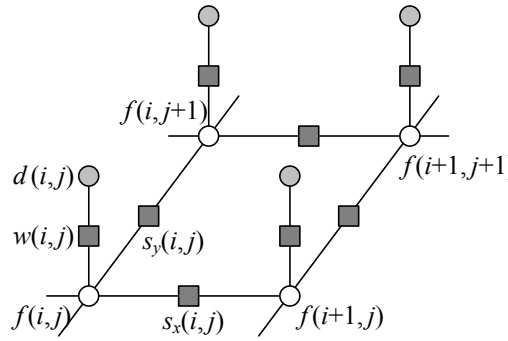


Figure B.1 Graphical model for an $\mathcal{N}_4$ neighborhood Markov random field. The white circles are the unknowns $f(i,j)$, while the dark circles are the input data $d(i,j)$. The $s_x(i,j)$ and $s_y(i,j)$ black boxes denote arbitrary *interaction potentials* between adjacent nodes in the random field, and the $w(i,j)$ denote the *data penalty* functions. They are all examples of the general potentials $V_{i,j,k,l}(f(i,j), f(k,l))$ used in Equation (B.24).

get

$$-\log p(\boldsymbol{x}|\boldsymbol{y}) = -\log p(\boldsymbol{y}|\boldsymbol{x}) - \log p(\boldsymbol{x}) + \log p(\boldsymbol{y}), \tag{B.22}$$

which is the *negative posterior log likelihood*. It is common to drop the constant $\log p(\boldsymbol{y})$ because its value does not matter during energy minimization. However, if the prior distribution $p(\boldsymbol{x})$ depends on some unknown parameters, we may wish to keep $\log p(\boldsymbol{y})$ in order to compute the most likely value of these parameters using *Occam's razor*, i.e., by maximizing the likelihood of the observations, or to select the correct number of free parameters using *model selection* (Hastie, Tibshirani, and Friedman 2001; Torr 2002; Bishop 2006; Robert 2007).

To find the most likely (*maximum a posteriori* or MAP) solution $\boldsymbol{x}$ given some measurements $\boldsymbol{y}$, we simply minimize this negative log likelihood, which can also be thought of as an *energy*,

$$E(\boldsymbol{x}, \boldsymbol{y}) = E_d(\boldsymbol{x}, \boldsymbol{y}) + E_p(\boldsymbol{x}). \tag{B.23}$$

The first term $E_d(\boldsymbol{x}, \boldsymbol{y})$ is the *data energy* or *data penalty* and measures the negative log likelihood that the measurements $\boldsymbol{y}$ were observed given the unknown state $\boldsymbol{x}$. The second term $E_p(\boldsymbol{x})$ is the *prior energy* and it plays a role analogous to the smoothness energy in regularization. Note that the MAP estimate may not always be desirable, since it selects the "peak" in the posterior distribution rather than some more stable statistic such as MSE—see the discussion in Appendix B.2 about loss functions and decision theory.

B.5 Markov random fields

Markov random fields (Blake, Kohli, and Rother 2010) are the most popular types of prior model for gridded image-like data,[5] which include not only regular natural images (Section 3.7.2) but also two-dimensional fields such as optic flow (Chapter 8) or depth maps (Chapter 11), as well as binary fields, such as segmentations (Section 5.5).

[5] Alternative formulations include power spectra (Section 3.4.3) and non-local means (Buades, Coll, and Morel 2008).

As we discussed in Section 3.7.2, the prior probability $p(\boldsymbol{x})$ for a Markov random field is a *Gibbs* or *Boltzmann distribution*, whose negative log likelihood (according to the Hammersley–Clifford Theorem) can be written as a sum of pairwise *interaction potentials*,

$$E_p(\boldsymbol{x}) = \sum_{\{(i,j),(k,l)\}\in\mathcal{N}} V_{i,j,k,l}(f(i,j), f(k,l)), \tag{B.24}$$

where $\mathcal{N}(i,j)$ denotes the *neighbors* of pixel (i,j). In the more general case, MRFs can also contain unary potentials, as well as *higher-order potentials* defined over larger cardinality *cliques* (Kindermann and Snell 1980; Geman and Geman 1984; Bishop 2006; Potetz and Lee 2008; Kohli, Kumar, and Torr 2009; Kohli, Ladický, and Torr 2009; Rother, Kohli, Feng *et al.* 2009; Alahari, Kohli, and Torr 2011). They can also contain *line processes*, i.e., additional binary variables that mediate discontinuities between adjacent elements (Geman and Geman 1984). Black and Rangarajan (1996) show how independent line process variables can be eliminated and incorporated into regular MRFs using robust pairwise penalty functions.

The most commonly used neighborhood in Markov random field modeling is the $\mathcal{N}_4$ neighborhood, where each pixel in the field $f(i,j)$ interacts only with its immediate neighbors—Figure B.1 shows such an $\mathcal{N}_4$ MRF. The $s_x(i,j)$ and $s_y(i,j)$ black boxes denote arbitrary interaction potentials between adjacent nodes in the random field and the $w(i,j)$ denote the elemental data penalty terms in E_d (B.23). These square nodes can also be interpreted as *factors* in a *factor graph* version of the undirected graphical model (Bishop 2006; Wainwright and Jordan 2008; Koller and Friedman 2009), which is another name for interaction potentials. (Strictly speaking, the factors are improper probability functions whose product is the un-normalized posterior distribution.)

More complex and higher-dimensional interaction models and neighborhoods are also possible. For example, 2D grids can be enhanced with the addition of diagonal connections (an $\mathcal{N}_8$ neighborhood) or even larger numbers of pairwise terms (Boykov and Kolmogorov 2003; Rother, Kolmogorov, Lempitsky *et al.* 2007). 3D grids can be used to compute globally optimal segmentations in 3D volumetric medical images (Boykov and Funka-Lea 2006) (Section 5.5.1). Higher-order cliques can also be used to develop more sophisticated models (Potetz and Lee 2008; Kohli, Ladický, and Torr 2009; Kohli, Kumar, and Torr 2009).

One of the biggest challenges in using MRF models is to develop efficient *inference algorithms* that will find low-energy solutions (Veksler 1999; Boykov, Veksler, and Zabih 2001; Kohli 2007; Kumar 2008). Over the years, a large variety of such algorithms have been developed, including simulated annealing, graph cuts, and loopy belief propagation. The choice of inference technique can greatly affect the overall performance of a vision system. For example, most of the top-performing algorithms on the Middlebury Stereo Evaluation page either use belief propagation or graph cuts.

In the next few subsections, we review some of the more widely used MRF inference techniques. More in-depth descriptions of most of these algorithms can be found in a recently published book on advances in MRF techniques (Blake, Kohli, and Rother 2010). Experimental comparisons, along with test datasets and reference software, are provided by Szeliski, Zabih, Scharstein *et al.* (2008).[6]

[6] http://vision.middlebury.edu/MRF/.

B.5.1 Gradient descent and simulated annealing

The simplest optimization technique is gradient descent, which minimizes the energy by changing independent subsets of nodes to take on lower-energy configurations. Such techniques go under a variety of names, including *contextual classification* (Kittler and Föglein 1984) and *iterated conditional modes* (ICM) (Besag 1986).[7] Variables can either be updated sequentially, e.g., in raster scan, or in parallel, e.g., using red–black coloring on a checkerboard. Chou and Brown (1990) suggests using highest confidence first (HCF), i.e., choosing variables based on how large a difference they make in reducing the energy.

The problem with gradient descent is that it is prone to getting stuck in local minima, which is almost always the case with MRF problems. One way around this is to use *stochastic gradient descent* or *Markov chain Monte Carlo* (MCMC) (Metropolis, Rosenbluth, Rosenbluth *et al.* 1953), i.e., to randomly take occasional uphill steps in order to get out of such minima. One popular update rule is the *Gibbs sampler* (Geman and Geman 1984); rather than choosing the lowest energy state for a variable being updated, it chooses the state with probability

$$p(\boldsymbol{x}) \propto e^{-E(\boldsymbol{x})/T}, \tag{B.25}$$

where T is called the *temperature* and controls how likely the system is to choose a more random update. Stochastic gradient descent is usually combined with *simulated annealing* (Kirkpatrick, Gelatt, and Vecchi 1983), which starts at a relatively high temperature, thereby randomly exploring a large part of the state space, and gradually cools (anneals) the temperature to find a good local minimum. During the late 1980s, simulated annealing was the method of choice for solving MRF inference problems (Szeliski 1986; Marroquin, Mitter, and Poggio 1985; Barnard 1989).

Another variant on simulated annealing is the Swendsen–Wang algorithm (Swendsen and Wang 1987; Barbu and Zhu 2003, 2005). Here, instead of "flipping" (changing) single variables, a connected subset of variables, chosen using a random walk based on MRF connectively strengths, is selected as the basic update unit. This can sometimes help make larger state changes, and hence find better-quality solutions in less time.

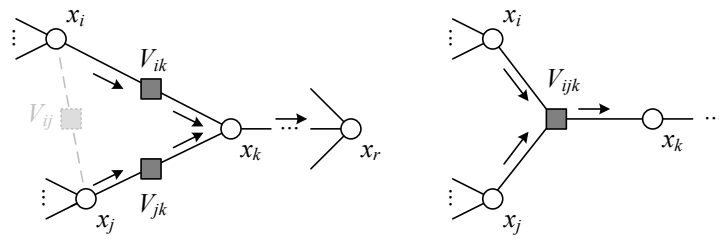
While simulated annealing has largely been superseded by the newer graph cuts and loopy belief propagation techniques, it still occasionally finds use, especially in highly connected and highly non-submodular graphs (Rother, Kolmogorov, Lempitsky *et al.* 2007).

B.5.2 Dynamic programming

Dynamic programming (DP) is an efficient inference procedure that works for any tree-structured graphical model, i.e., one that does not have any cycles. Given such a tree, pick any node as the root r and figuratively pick up the tree by its root. The depth or distance of all the other nodes from this root induces a partial ordering over the vertices, from which a total ordering can be obtained by arbitrarily breaking ties. Let us now lay out this graph as a tree with the root on the right and indices increasing from left to right, as shown in Figure B.2a.

Before describing the DP algorithm, let us re-write the potential function of Equation (B.24)

[7] The name comes from iteratively setting variables to the mode (most likely, i.e., lowest energy) state conditioned on its currently fixed neighbors.



(a) (b)

Figure B.2 Dynamic programming over a tree drawn as a factor graph. (a) To compute the lowest energy solution $\hat{E}_k(x_k)$ at node x_k conditioned on the best solutions to the left of this node, we enumerate all possible values of $\hat{E}_i(x_i) + V_{ik}(x_i, x_k)$ and pick the smallest one (and similarly for j). (b) For higher-order cliques, we need to try all combinations of (x_i, x_j) in order to select the best possible configuration. The arrows show the basic flow of the computation. The lightly shaded factor V_{ij} in (a) shows an additional connection that turns the tree into a cyclic graph, for which exact inference cannot be efficiently computed.

in a more general but succinct form,

$$E(\boldsymbol{x}) = \sum_{(i,j)\in\mathcal{N}} V_{i,j}(x_i, x_j) + \sum_i V_i(x_i), \tag{B.26}$$

where instead of using pixel indices (i, j) and (k, l), we just use scalar index variables i and j. We also replace the function value $f(i, j)$ with the more succinct notation x_i, with the $\{x_i\}$ variables making up the state vector $\boldsymbol{x}$. We can simplify this function even further by adding dummy nodes (vertices) i^- for every node that has a non-zero $V_i(x_i)$ and setting $V_{i,i^-}(x_i, x_{i^-}) = V_i(x_i)$, which lets us drop the V_i terms from (B.26).

Dynamic programming proceeds by computing partial sums in a left-to-right fashion, i.e., in order of increasing variable index. Let $\mathcal{C}_k$ be the children of k, i.e., $i < k, (i, k) \in \mathcal{N}$). Then, define

$$\tilde{E}_k(\boldsymbol{x}) = \sum_{i<k,\ j\le k} V_{i,j}(x_i, x_j) = \sum_{i\in\mathcal{C}_k} \Big[V_{i,k}(x_i, x_k) + \tilde{E}_i(\boldsymbol{x})\Big], \tag{B.27}$$

as a partial sum of (B.26) over all variables up to and including k, i.e., over all parts of the graph shown in Figure B.2a to the left of x_k. This sum depends on the state of all the unknown variables in $\boldsymbol{x}$ with $i \le k$.

Now suppose we wish to find the setting for all variables $i < k$ that minimizes this sum. It turns out that we can use a simple recursive formula

$$\hat{E}_k(x_k) = \min_{\{x_i,\ i<k\}} \tilde{E}_k(\boldsymbol{x}) = \sum_{i\in\mathcal{C}_k} \min_{x_i} \Big[V_{i,k}(x_i, x_k) + \hat{E}_i(x_i)\Big] \tag{B.28}$$

to find this minimum. Visually, this is easy to understand. Looking at Figure B.2a, associate an energy $\hat{E}_k(x_k)$ with each node k and each possible setting of its value x_k that is based on the *best* possible setting of variables to the left of that node. It is easy to convince yourself that in this figure, you only need to know $\hat{E}_i(x_i)$ and $\hat{E}_j(x_j)$ in order to compute this value.

Once the flow of information in the tree has been processed from left to right, the minimum value of $\hat{E}_r(x_r)$ at the root gives the MAP (lowest-energy) solution for $E(\boldsymbol{x})$. The

root node is set to the choice of x_r that minimizes this function, and other nodes are set in a *backward chaining* pass by selecting the values of child nodes $i \in \mathcal{C}_k$ that were minimal in the original recursion (B.28).

Dynamic programming is not restricted to trees with pairwise potentials. Figure B.2b shows an example of a three-way potential $V_{ijk}(x_i, x_j, x_k)$ inside a tree. To compute the optimum value of $\hat{E}_k(x_k)$, the recursion formula in (B.28) now has to evaluate the minimum over all combinations of possible state values leading into a factor node (gray box). For this reason, dynamic programming is normally exponential in complexity in the order of the clique size, i.e., a clique of size n with l labels at each node requires the evaluation of l^{n-1} possible states (Potetz and Lee 2008; Kohli, Kumar, and Torr 2009). However, for certain kinds of potential functions $V_{i,k}(x_i, x_k)$, including the Potts model (delta function), absolute values (total variation), and quadratic (Gaussian MRF), Felzenszwalb and Huttenlocher (2006) show how to reduce the complexity of the min-finding step (B.28) from $O(l^2)$ to $O(l)$. In Appendix B.5.3, we also discuss how Potetz and Lee (2008) reduce the complexity for special kinds of higher-order clique, i.e., linear summations followed by non-linearities.

Figure B.2a also shows what happens if we add an extra factor between nodes i and j. In this case, the graph is no longer a tree, i.e., it contains a cycle. It is no longer possible to use the recursion formula (B.28), since $\hat{E}_i(x_i)$ now appears in two different terms inside the summation, i.e., as a child of both nodes j and k, and the same setting for x_i may not minimize both. In other words, when loops exist, there is no ordering of the variables that allows the recursion (elimination) in (B.28) to be well-founded.

It is, however, possible to convert small loops into higher-order factors and to solve these as shown in Figure B.2b. However, graphs with long loops or meshes result in extremely large clique sizes and hence an amount of computation potentially exponential in the size of the graph.

B.5.3 Belief propagation

Belief propagation is an inference technique originally developed for trees (Pearl 1988) but more recently extended to "loopy" (cyclic) graphs such as MRFs (Frey and MacKay 1997; Freeman, Pasztor, and Carmichael 2000; Yedidia, Freeman, and Weiss 2001; Weiss and Freeman 2001a,b; Yuille 2002; Sun, Zheng, and Shum 2003; Felzenszwalb and Huttenlocher 2006). It is closely related to dynamic programming, in that both techniques pass messages forward and backward over a tree or graph. In fact, one of the two variants of belief propagation, the *max-product rule*, performs the exact same computation (inference) as dynamic programming, albeit using probabilities instead of energies.

Recall that the energy we are minimizing in MAP estimation (B.26) is the negative log likelihood (B.12, B.13, and B.22) of a factored Gibbs posterior distribution,

$$p(\boldsymbol{x}) = \prod_{(i,j)\in\mathcal{N}} \phi_{i,j}(x_i, x_j), \tag{B.29}$$

where

$$\phi_{i,j}(x_i, x_j) = e^{-V_{i,j}(x_i, x_j)} \tag{B.30}$$

are the pairwise *interaction potentials*. We can rewrite (B.27) as

$$\tilde{p}_k(\boldsymbol{x}) = \prod_{i<k,\ j\le k} \phi_{i,j}(x_i, x_j) = \prod_{i\in\mathcal{C}_k} \tilde{p}_{i,k}(\boldsymbol{x}), \tag{B.31}$$

where

$$\tilde{p}_{i,k}(\boldsymbol{x}) = \phi_{i,k}(x_i, x_k)\tilde{p}_i(\boldsymbol{x}). \tag{B.32}$$

We can therefore rewrite (B.28) as

$$\hat{p}_k(x_k) = \max_{\{x_i,\, i<k\}} \tilde{p}_k(\boldsymbol{x}) = \prod_{i \in \mathcal{C}_k} \hat{p}_{i,k}(\boldsymbol{x}), \tag{B.33}$$

with

$$\hat{p}_{i,k}(\boldsymbol{x}) = \max_{x_i} \phi_{i,k}(x_i, x_k)\hat{p}_i(\boldsymbol{x}). \tag{B.34}$$

Equation (B.34) is the *max* update rule evaluated at all square box factors in Figure B.2a, while (B.33) is the *product* rule evaluated at the nodes. The probability distribution $\hat{p}_{i,k}(\boldsymbol{x})$ is often interpreted as a *message* passing information about child i to parent k and is hence written as $m_{i,k}(x_k)$ (Yedidia, Freeman, and Weiss 2001) or $\mu_{i\to k}(x_k)$ (Bishop 2006).

The max-product rule can be used to compute the MAP estimate in a tree using the same kind of forward and backward sweep as in dynamic programming (which is sometimes called the *max-sum* algorithm (Bishop 2006)). An alternative rule, known as the *sum–product*, sums over all possible values in (B.34) rather than taking the maximum, in essence computing the *expected* distribution rather than the *maximum likelihood* distribution. This produces a set of probability estimates that can be used to compute the *marginal* distributions $b_i(x_i) = \sum_{\boldsymbol{x}\setminus x_i} p(\boldsymbol{x})$ (Pearl 1988; Yedidia, Freeman, and Weiss 2001; Bishop 2006).

Belief propagation may not produce optimal estimates for cyclic graphs for the same reason that dynamic programming fails to work, i.e., because a node with multiple parents may take on different optimal values for each of the parents, i.e., there is no unique elimination ordering. Early algorithms for extending belief propagation to graphs with cycles, dubbed *loopy belief propagation*, performed the updates in parallel over the graph, i.e., using *synchronous updates* (Frey and MacKay 1997; Freeman, Pasztor, and Carmichael 2000; Yedidia, Freeman, and Weiss 2001; Weiss and Freeman 2001a,b; Yuille 2002; Sun, Zheng, and Shum 2003; Felzenszwalb and Huttenlocher 2006).

For example, Felzenszwalb and Huttenlocher (2006) split an $\mathcal{N}_4$ graph into its red and black (checkerboard) components and alternate between sending messages from the red nodes to the black and vice versa. They also use multi-grid (coarser level) updates to speed up the convergence. As discussed previously, to reduce the complexity of the basic max-product update rule (B.28) from $O(l^2)$ to $O(l)$, they develop specialized update algorithms for several cost functions $V_{i,k}(x_i, x_k)$, including the Potts model (delta function), absolute values (total variation), and quadratic (Gaussian MRF). A related algorithm, *mean field diffusion* (Scharstein and Szeliski 1998), also uses synchronous updates between nodes to compute marginal distributions. Yuille (2010) discusses the relationships between mean field theory and loopy belief propagation.

More recent loopy belief propagation algorithms and their variants use sequential scans through the graph (Szeliski, Zabih, Scharstein *et al.* 2008). For example, Tappen and Freeman (2003) pass messages from left to right along each row and then reverse the direction once they reach the end. This is similar to treating each row as an independent tree (chain), except that messages from nodes above and below the row are also incorporated. They then perform similar computations along columns. These sequential updates allow the information to propagate much more quickly across the image than synchronous updates.

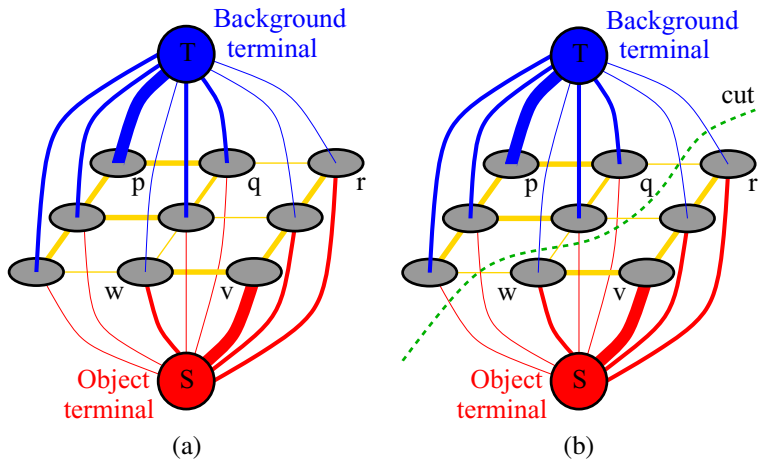


Figure B.3 Graph cuts for minimizing binary sub-modular MRF energies (Boykov and Jolly 2001) © 2001 IEEE: (a) energy function encoded as a max flow problem; (b) the minimum cut determines the region boundary.

The other belief propagation variant tested by Szeliski, Zabih, Scharstein *et al.* (2008), which they call BP-S or TRW-S, is based on Kolmogorov's (2006) sequential extension of the *tree-reweighted message passing* of Wainwright, Jaakkola, and Willsky (2005). TRW first selects a set of trees from the neighborhood graph and computes a set of probability distributions over each tree. These are then used to reweight the messages being passed during loopy belief propagation. The sequential version of TRW, called TRW-S, processed nodes in scan-line order, with a forward and backward pass. In the forward pass, each node sends messages to its right and bottom neighbors. In the backward pass, messages are sent to the left and upper neighbors. TRW-S also computes a lower bound on the energy, which is used by Szeliski, Zabih, Scharstein *et al.* (2008) to estimate how close to the best possible solution all of the MRF inference algorithms being evaluated get.

As with dynamic programming, belief propagation techniques also become less efficient as the order of each factor clique increases. Potetz and Lee (2008) shows how this complexity can be reduced back to linear in the clique order for continuous-valued problems where the factors involve linear summations followed by a non-linearity, which is typical of more sophisticated MRF models such as fields of experts (Roth and Black 2009) and steerable random fields (Roth and Black 2007b). Kohli, Kumar, and Torr (2009) and Alahari, Kohli, and Torr (2011) develop alternative ways for dealing with higher-order cliques in the context of graph cut algorithms.

B.5.4 Graph cuts

The computer vision community has adopted "graph cuts" as an informal name to describe a large family of MRF inference algorithms based on solving one or more min-cut or max-flow problems (Boykov, Veksler, and Zabih 2001; Boykov and Kolmogorov 2010; Boykov, Veksler, and Zabih 2010; Ishikawa and Veksler 2010).

The simplest example of an MRF graph cut is the polynomial-time algorithm for performing exact minimization of a binary MRF originally developed by Greig, Porteous, and Seheult

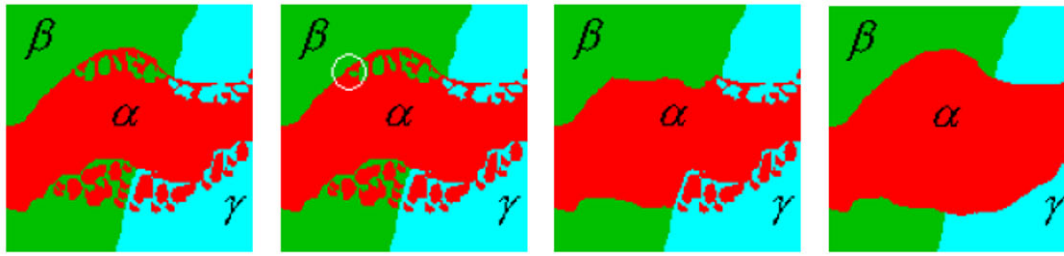
(1989) and brought to the attention of the computer vision community by Boykov, Veksler, and Zabih (2001) and Boykov and Jolly (2001). The basic construction of the min-cut graph from an MRF energy function is shown in Figure B.3 and described in Sections 3.7.2 and 5.5. In brief, the nodes in an MRF are connected to special source and sink nodes, and the minimum cut between these two nodes, whose cost is exactly that of the MRF energy under a binary assignment of labels, is computed using a polynomial-time max flow algorithm (Goldberg and Tarjan 1988; Boykov and Kolmogorov 2004).

As discussed in Section 5.5, important extensions of this basic algorithm have been made for the case of directed edges (Kolmogorov and Boykov 2005), larger neighborhoods (Boykov and Kolmogorov 2003; Kolmogorov and Boykov 2005), connectivity priors (Vicente, Kolmogorov, and Rother 2008), and shape priors (Lempitsky and Boykov 2007; Lempitsky, Blake, and Rother 2008). Kolmogorov and Zabih (2004) formally characterize the class of binary energy potentials (*regularity conditions*) for which these algorithms find the global minimum. Komodakis, Tziritas, and Paragios (2008) and Rother, Kolmogorov, Lempitsky *et al.* (2007) provide good algorithms for the cases when they do not.

Binary MRF problems can also be approximately solved by turning them into continuous $[0, 1]$ problems, solving them either as linear systems (Grady 2006; Sinop and Grady 2007; Grady and Alvino 2008; Grady 2008; Grady and Ali 2008; Singaraju, Grady, and Vidal 2008; Couprie, Grady, Najman *et al.* 2009) (the *random walker model*) or by computing geodesic distances (Bai and Sapiro 2009; Criminisi, Sharp, and Blake 2008) and then thresholding the results. More details on these techniques are provided in Section 5.5 and a nice review can be found in the work of Singaraju, Grady, Sinop *et al.* (2010). A different connection to continuous segmentation techniques, this time to the literature on level sets (Section 5.1.4), is made by Boykov, Kolmogorov, Cremers *et al.* (2006), who develop an approach to solving surface propagation PDEs based on combinatorial graph cut algorithms—Boykov and Funka-Lea (2006) discuss this and related techniques.

Multi-valued MRF inference problems usually require solving a series of related binary MRF problems (Boykov, Veksler, and Zabih 2001), although for special cases, such as some convex functions, a single graph cut may suffice (Ishikawa 2003; Schlesinger and Flach 2006). The seminal work in this area is that of Boykov, Veksler, and Zabih (2001), who introduced two algorithms, called the *swap move* and the *expansion move*, which are sketched in Figure B.4. The α–β-swap move selects two labels (usually by cycling through all possible pairings) and then formulates a binary MRF problem that allows any pixels currently labeled as either α or β to optionally switch their values to the other label. The α-expansion move allows any pixel in the MRF to take on the α label or to keep its current identity. It is easy to see by inspection that both of these moves result in binary MRFs with well-defined energy functions.

Because these algorithms use a binary MRF optimization inside their inner loop, they are subject to the constraints on the energy functions that occur in the binary labeling case (Kolmogorov and Zabih 2004). However, more recent algorithms such as those developed by Komodakis, Tziritas, and Paragios (2008) and Rother, Kolmogorov, Lempitsky *et al.* (2007) can be used to provide approximate solutions for more general energy functions. Efficient algorithms for re-using previous solutions (*flow-* and *cut-recycling*) have been developed for on-line applications such as *dynamic MRFs* (Kohli and Torr 2005; Juan and Boykov 2006; Alahari, Kohli, and Torr 2011) and coarse-to-fine banded graph cuts (Agarwala, Zheng, Pal *et*



(a) initial labeling (b) standard move (c) α-β-swap (d) α-expansion

Figure B.4 Multi-level graph optimization from (Boykov, Veksler, and Zabih 2001) © 2001 IEEE: (a) initial problem configuration; (b) the standard move changes only one pixel; (c) the α–β-swap optimally exchanges all α- and β-labeled pixels; (d) the α-expansion move optimally selects among current pixel values and the α label.

al. 2005; Lombaert, Sun, Grady *et al.* 2005; Juan and Boykov 2006). It is also now possible to minimize the number of labels used as part of the alpha-expansion process (Delong, Osokin, Isack *et al.* 2010).

In experimental comparisons, α-expansions usually converge faster to a good solution than α–β-swaps (Szeliski, Zabih, Scharstein *et al.* 2008), especially for problems that involve large regions of identical labels, such as the labeling of source imagery in image stitching (Figure 3.60). For truncated convex energy functions defined over ordinal values, more accurate algorithms that consider complete ranges of labels inside each min-cut and often produce lower energies have been developed (Veksler 2007; Kumar and Torr 2008; Kumar, Veksler, and Torr 2010). The whole field of efficient MRF inference algorithms is rapidly developing, as witnessed by a recent special journal issue (Kohli and Torr 2008; Komodakis, Tziritas, and Paragios 2008; Olsson, Eriksson, and Kahl 2008; Potetz and Lee 2008), articles (Alahari, Kohli, and Torr 2011), and a forthcoming book (Blake, Kohli, and Rother 2010).

B.5.5 Linear programming

[8] Many successful algorithms for MRF optimization are based on the *linear programming* (LP) relaxation of the energy function (Weiss, Yanover, and Meltzer 2010). For some practical MRF problems, LP-based techniques can produce globally minimal solutions (Meltzer, Yanover, and Weiss 2005), even though MRF inference is in general NP-hard. In order to describe this relaxation, let us first rewrite the energy function (B.26) as

$$E(\boldsymbol{x}) = \sum_{(i,j)\in\mathcal{N}} V_{i,j}(x_i, x_j) + \sum_i V_i(x_i) \tag{B.35}$$

$$= \sum_{i,j,\alpha,\beta} V_{i,j}(\alpha,\beta) x_{i,j;\alpha,\beta} + \sum_{i,\alpha} V_i(\alpha) x_{i;\alpha} \tag{B.36}$$

$$\text{subject to} \quad x_{i;\alpha} = \sum_\beta x_{i,j;\alpha,\beta} \quad \forall (i,j) \in \mathcal{N}, \alpha, \tag{B.37}$$

$$x_{j;\beta} = \sum_\alpha x_{i,j;\alpha,\beta} \quad \forall (i,j) \in \mathcal{N}, \beta, \quad \text{and} \tag{B.38}$$

$$x_{i,\alpha},\ x_{i,j;\alpha,\beta} \in \{0,1\}. \tag{B.39}$$

[8] This section was contributed by Vladimir Kolmogorov. Thanks!

Here, α and β range over label values and $x_{i;\alpha} = \delta(x_i - \alpha)$ and $x_{ij;\alpha\beta} = \delta(x_i - \alpha)\delta(x_j - \beta)$ are indicator variables of assignments $x_i = \alpha$ and $(x_i, x_j) = (\alpha, \beta)$, respectively. The LP relaxation is obtained by replacing the discreteness constraints (B.39) with linear constraints $x_{ij;\alpha\beta} \in [0, 1]$. It is easy to show that the optimal value of (B.36) is a lower bound on (B.26).

This relaxation has been extensively studied in the literature, starting with the work of Schlesinger (1976). An important question is how to solve this LP efficiently. Unfortunately, general-purpose LP solvers cannot handle large problems in vision (Yanover, Meltzer, and Weiss 2006). A large number of customized iterative techniques have been proposed. Most of these solve the dual problem, i.e., they formulate a lower bound on (B.36) and then try to maximize this bound. The bound is often formulated using a convex combination of trees, as proposed in (Wainwright, Jaakkola, and Willsky 2005).

The LP lower bound can be maximized via a number of techniques, such as *max-sum diffusion* (Werner 2007), *tree-reweighted message passing* (TRW) (Wainwright, Jaakkola, and Willsky 2005; Kolmogorov 2006), subgradient methods (Schlesinger and Giginyak 2007a,b; Komodakis, Paragios, and Tziritas 2007), and Bregman projections (Ravikumar, Agarwal, and Wainwright 2008). Note that the max-sum diffusion and TRW algorithms are not guaranteed to converge to a global maximum of LP—they may get stuck at a suboptimal point (Kolmogorov 2006; Werner 2007). However, in practice, this does not appear to be a problem (Kolmogorov 2006).

For some vision applications, algorithms based on relaxation (B.36) produce excellent results. However, this is not guaranteed in all cases—after all, the problem is NP-hard. Recently, researchers have investigated alternative linear programming relaxations (Sontag and Jaakkola 2007; Sontag, Meltzer, Globerson *et al.* 2008; Komodakis and Paragios 2008; Schraudolph 2010). These algorithms are capable of producing tighter bounds compared to (B.36) at the expense of additional computational cost.

LP relaxation and alpha expansion. Solving a linear program produces primal and dual solutions that satisfy *complementary slackness conditions*. In general, the primal solution of (B.36) does not have to be integer-valued so, in practice, we may have to round it to obtain a valid labeling $\boldsymbol{x}$. An alternative proposed by Komodakis and Tziritas (2007a); Komodakis, Tziritas, and Paragios (2007) is to search for primal and dual solutions such that they satisfy *approximate* complementary slackness conditions and the primal solution is already integer-valued. Several max-flow-based algorithms are proposed by (Komodakis and Tziritas 2007a; Komodakis, Tziritas, and Paragios 2007) for this purpose and the *Fast-PD* method (Komodakis, Tziritas, and Paragios 2007) is shown to perform best. In the case of metric interactions, the default version of Fast-PD produces the same primal solution as the alpha-expansion algorithm (Boykov, Veksler, and Zabih 2001). This provides an interesting interpretation of the alpha expansion algorithm as trying to approximately solve relaxation (B.36).

Unlike the standard alpha expansion algorithm, Fast-PD also maintains a dual solution and thus runs faster in practice. Fast-PD can be extended to the case of semi-metric interactions (Komodakis, Tziritas, and Paragios 2007). The primal version of such extension was also given by Rother, Kumar, Kolmogorov *et al.* (2005).

B.6 Uncertainty estimation (error analysis)

In addition to computing the most likely estimate, many applications require an estimate for the *uncertainty* in this estimate.[9] The most general way to do this is to compute a complete probability distribution over all of the unknowns but this is generally intractable. The one special case where it is easy to obtain a simple description for this distribution is linear estimation problems with Gaussian noise, where the joint energy function (negative log likelihood of the posterior estimate) is a quadratic. In this case, the posterior distribution is a multi-variate Gaussian and the covariance can be computed directly from the inverse of the problem Hessian. (Another name for the inverse covariance matrix, which is equal to the Hessian in such simple cases, is the *information matrix*.)

Even here, however, the full covariance matrix may be too large to compute and store. For example, in large structure from motion problems, a large sparse Hessian normally results in a full dense covariance matrix. In such cases, it is often considered acceptable to report only the variance in the estimated quantities or simple covariance estimates on individual parameters, such as 3D point positions or camera pose estimates (Szeliski 1990a). More insight into the problem, e.g., the dominant *modes* of uncertainty, can be obtained using eigenvalue analysis (Szeliski and Kang 1997).

For problems where the posterior energy is non-quadratic, e.g., in non-linear or robustified least squares, it is still often possible to obtain an estimate of the Hessian in the vicinity of the optimal solution. In this case, the *Cramer–Rao lower bound* on the uncertainty (covariance) can be computed as the inverse of the Hessian. Another way of saying this is that while the local Hessian can underestimate how "wide" the energy function can be, the covariance can never be smaller than the estimate based on this local quadratic approximation. It is also possible to estimate a different kind of uncertainty (min-marginal energies) in general MRFs where the MAP inference is performed using graph cuts (Kohli and Torr 2008).

While many computer vision applications ignore uncertainty modeling, it is often useful to compute these estimates just to get an intuitive feeling for the reliability of the estimates. Certain applications, such as Kalman filtering, require the computation of this uncertainty (either explicitly as posterior covariances or implicitly as inverse covariances) in order to optimally integrate new measurements with previously computed estimates.

[9] This is particularly true of classic photogrammetry applications, where the reporting of precision is almost always considered mandatory (Förstner 2005).

Appendix C

Supplementary material

C.1 Data sets 680
C.2 Software 682
C.3 Slides and lectures 689
C.4 Bibliography 690

In this final appendix, I summarize some of the supplementary materials that may be useful to students, instructors, and researchers. The book's Web site at http://szeliski.org/Book contains updated lists of datasets and software, so please check there as well.

C.1 Data sets

One of the keys to developing reliable vision algorithms is to test your procedures on challenging and representative data sets. When ground truth or other people's results are available, such test can be even more informative (and quantitative).

Over the years, a large number of datasets have been developed for testing and evaluating computer vision algorithms. A number of these datasets (and software) are indexed on the Computer Vision Homepage.[1] Some newer Web sites, such as CVonline (http://homepages.inf.ed.ac.uk/rbf/CVonline/), VisionBib.Com (http://datasets.visionbib.com/), and Computer Vision online (http://computervisiononline.com/), have more recent pointers.

Below, I list some of the more popular data sets, grouped by the book chapters to which they most closely correspond:

Chapter 2: Image formation

CUReT: Columbia-Utrecht Reflectance and Texture Database, http://www1.cs.columbia.edu/CAVE/software/curet/ (Dana, van Ginneken, Nayar *et al.* 1999).

Middlebury Color Datasets: registered color images taken by different cameras to study how they transform gamuts and colors, http://vision.middlebury.edu/color/data/ (Chakrabarti, Scharstein, and Zickler 2009).

Chapter 3: Image processing

Middlebury test datasets for evaluating MRF minimization/inference algorithms, http://vision.middlebury.edu/MRF/results/ (Szeliski, Zabih, Scharstein *et al.* 2008).

Chapter 4: Feature detection and matching

Affine Covariant Features database for evaluating feature detector and descriptor matching quality and repeatability, http://www.robots.ox.ac.uk/~vgg/research/affine/ (Mikolajczyk and Schmid 2005; Mikolajczyk, Tuytelaars, Schmid *et al.* 2005).

Database of matched image patches for learning and feature descriptor evaluation, http://cvlab.epfl.ch/~brown/patchdata/patchdata.html (Winder and Brown 2007; Hua, Brown, and Winder 2007).

Chapter 5: Segmentation

Berkeley Segmentation Dataset and Benchmark of 1000 images labeled by 30 humans, along with an evaluation, http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/ (Martin, Fowlkes, Tal *et al.* 2001).

[1] http://www.cs.cmu.edu/~cil/vision.html, although it has not been maintained since 2004.

Weizmann segmentation evaluation database of 100 grayscale images with ground truth segmentations, http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/index.html (Alpert, Galun, Basri *et al.* 2007).

Chapter 8: Dense motion estimation

The Middlebury optic flow evaluation Web site, http://vision.middlebury.edu/flow/data (Baker, Scharstein, Lewis *et al.* 2009).

The Human-Assisted Motion Annotation database, http://people.csail.mit.edu/celiu/motionAnnotation/ (Liu, Freeman, Adelson *et al.* 2008)

Chapter 10: Computational photography

High Dynamic Range radiance maps, http://www.debevec.org/Research/HDR/ (Debevec and Malik 1997).

Alpha matting evaluation Web site, http://alphamatting.com/ (Rhemann, Rother, Wang *et al.* 2009).

Chapter 11: Stereo correspondence

Middlebury Stereo Datasets and Evaluation, http://vision.middlebury.edu/stereo/ (Scharstein and Szeliski 2002).

Stereo Classification and Performance Evaluation of different aggregation costs for stereo matching, http://www.vision.deis.unibo.it/spe/SPEHome.aspx (Tombari, Mattoccia, Di Stefano *et al.* 2008).

Middlebury Multi-View Stereo Datasets, http://vision.middlebury.edu/mview/data/ (Seitz, Curless, Diebel *et al.* 2006).

Multi-view and Oxford Colleges building reconstructions, http://www.robots.ox.ac.uk/~vgg/data/data-mview.html.

Multi-View Stereo Datasets, http://cvlab.epfl.ch/data/strechamvs/ (Strecha, Fransens, and Van Gool 2006).

Multi-View Evaluation, http://cvlab.epfl.ch/~strecha/multiview/ (Strecha, von Hansen, Van Gool *et al.* 2008).

Chapter 12: 3D reconstruction

HumanEva: synchronized video and motion capture dataset for evaluation of articulated human motion, http://vision.cs.brown.edu/humaneva/ (Sigal, Balan, and Black 2010).

Chapter 13: Image-based rendering

The (New) Stanford Light Field Archive, http://lightfield.stanford.edu/ (Wilburn, Joshi, Vaish *et al.* 2005).

Virtual Viewpoint Video: multi-viewpoint video with per-frame depth maps, http://research.microsoft.com/en-us/um/redmond/groups/ivm/vvv/ (Zitnick, Kang, Uyttendaele *et al.* 2004).

Chapter 14: Recognition

For a list of visual recognition datasets, see Tables 14.1–14.2. In addition to those, there are also:

Buffy pose classes, http://www.robots.ox.ac.uk/~vgg/data/buffy_pose_classes/ and Buffy stickmen V2.1, http://www.robots.ox.ac.uk/~vgg/data/stickmen/index.html (Ferrari, Marin-Jimenez, and Zisserman 2009; Eichner and Ferrari 2009).

H3D database of pose/joint annotated photographs of humans, http://www.eecs.berkeley.edu/~lbourdev/h3d/ (Bourdev and Malik 2009).

Action Recognition Datasets, http://www.cs.berkeley.edu/projects/vision/action, has pointers to several datasets for action and activity recognition, as well as some papers. The human action database at http://www.nada.kth.se/cvap/actions/ contains more action sequences.

C.2 Software

One of the best sources for computer vision algorithms is the Open Source Computer Vision (OpenCV) library (http://opencv.willowgarage.com/wiki/), which was developed by Gary Bradski and his colleagues at Intel and is now being maintained and extended at Willow Garage (Bradsky and Kaehler 2008). A partial list of the available functions, taken from http://opencv.willowgarage.com/documentation/cpp/ includes:

- image processing and transforms (filtering, morphology, pyramids);
- geometric image transformations (rotations, resizing);
- miscellaneous image transformations (Fourier transforms, distance transforms);
- histograms;
- segmentation (watershed, mean shift);
- feature detection (Canny, Harris, Hough, MSER, SURF);
- motion analysis and object tracking (Lucas–Kanade, mean shift);
- camera calibration and 3D reconstruction;
- machine learning (k nearest neighbors, support vector machines, decision trees, boosting, random trees, expectation-maximization, and neural networks).

The Intel Performance Primitives (IPP) library, http://software.intel.com/en-us/intel-ipp/, contains highly optimized code for a variety of image processing tasks. Many of the routines in OpenCV take advantage of this library, if it is installed, to run even faster. In terms of

functionality, it has many of the same operators as those found in OpenCV, plus additional libraries for image and video compression, signal and speech processing, and matrix algebra.

The MATLAB Image Processing Toolbox, http://www.mathworks.com/products/image/, contains routines for spatial transformations (rotations, resizing), normalized cross-correlation, image analysis and statistics (edges, Hough transform), image enhancement (adaptive histogram equalization, median filtering) and restoration (deblurring), linear filtering (convolution), image transforms (Fourier and DCT), and morphological operations (connected components and distance transforms).

Two older libraries, which no longer appear to be under active development but contain many useful routines, are VXL (C++ Libraries for Computer Vision Research and Implementation, http://vxl.sourceforge.net/) and LTI-Lib 2 (http://www.ie.itcr.ac.cr/palvarado/ltilib-2/homepage/).

Photo editing and viewing packages, such as Windows Live Photo Gallery, iPhoto, Picasa, GIMP, and IrfanView, can be useful for performing common processing tasks, converting formats, and viewing your results. They can also serve as interesting reference implementations for image processing algorithms (such as tone correction or denoising) that you are trying to develop from scratch.

There are also software packages and infrastructure that can be helpful for building real-time video processing demos. Vision on Tap (http://www.visionontap.com/) provides a Web service that will process your webcam video in real time (Chiu and Raskar 2009). VideoMan (VideoManager, http://videomanlib.sourceforge.net/) can be useful for getting real-time video-based demos and applications running. You can also use `imread` in MATLAB to read directly from any URL, such as a webcam.

Below, I list some additional software that can be found on the Web, grouped by the book chapters to which they most correspond:

Chapter 3: Image processing

matlabPyrTools—MATLAB source code for Laplacian pyramids, QMF/Wavelets, and steerable pyramids, http://www.cns.nyu.edu/~lcv/software.php (Simoncelli and Adelson 1990a; Simoncelli, Freeman, Adelson *et al.* 1992).

BLS-GSM image denoising, http://decsai.ugr.es/~javier/denoise/ (Portilla, Strela, Wainwright *et al.* 2003).

Fast bilateral filtering code, http://people.csail.mit.edu/jiawen/#code (Chen, Paris, and Durand 2007).

C++ implementation of the fast distance transform algorithm, http://people.cs.uchicago.edu/~pff/dt/ (Felzenszwalb and Huttenlocher 2004a).

GREYC's Magic Image Converter, including image restoration software using regularization and anisotropic diffusion, http://gmic.sourceforge.net/gimp.shtml (Tschumperlé and Deriche 2005).

Chapter 4: Feature detection and matching

VLFeat, an open and portable library of computer vision algorithms, http://vlfeat.org/ (Vedaldi and Fulkerson 2008).

SiftGPU: A GPU Implementation of Scale Invariant Feature Transform (SIFT), http://www.cs.unc.edu/~ccwu/siftgpu/ (Wu 2010).

SURF: Speeded Up Robust Features, http://www.vision.ee.ethz.ch/~surf/ (Bay, Tuytelaars, and Van Gool 2006).

FAST corner detection, http://mi.eng.cam.ac.uk/~er258/work/fast.html (Rosten and Drummond 2005, 2006).

Linux binaries for affine region detectors and descriptors, as well as MATLAB files to compute repeatability and matching scores, http://www.robots.ox.ac.uk/~vgg/research/affine/.

Kanade–Lucas–Tomasi feature trackers: KLT, http://www.ces.clemson.edu/~stb/klt/ (Shi and Tomasi 1994); GPU-KLT, http://cs.unc.edu/~cmzach/opensource.html (Zach, Gallup, and Frahm 2008); and Lucas–Kanade 20 Years On, http://www.ri.cmu.edu/projects/project_515.html (Baker and Matthews 2004).

Chapter 5: Segmentation

Efficient graph-based image segmentation, http://people.cs.uchicago.edu/~pff/segment/ (Felzenszwalb and Huttenlocher 2004b).

EDISON, edge detection and image segmentation, http://coewww.rutgers.edu/riul/research/code/EDISON/ (Meer and Georgescu 2001; Comaniciu and Meer 2002).

Normalized cuts segmentation including intervening contours, http://www.cis.upenn.edu/~jshi/software/ (Shi and Malik 2000; Malik, Belongie, Leung *et al.* 2001).

Segmentation by weighted aggregation (SWA), http://www.cs.weizmann.ac.il/~vision/SWA/ (Alpert, Galun, Basri *et al.* 2007).

Chapter 6: Feature-based alignment and calibration

Non-iterative PnP algorithm, http://cvlab.epfl.ch/software/EPnP/ (Moreno-Noguer, Lepetit, and Fua 2007).

Tsai Camera Calibration Software, http://www-2.cs.cmu.edu/~rgw/TsaiCode.html (Tsai 1987).

Easy Camera Calibration Toolkit, http://research.microsoft.com/en-us/um/people/zhang/Calib/ (Zhang 2000).

Camera Calibration Toolbox for MATLAB, http://www.vision.caltech.edu/bouguetj/calib_doc/; a C version is included in OpenCV.

MATLAB functions for multiple view geometry, http://www.robots.ox.ac.uk/~vgg/hzbook/code/ (Hartley and Zisserman 2004).

Chapter 7: Structure from motion

SBA: A generic sparse bundle adjustment C/C++ package based on the Levenberg–Marquardt algorithm, http://www.ics.forth.gr/~lourakis/sba/ (Lourakis and Argyros 2009).

Simple sparse bundle adjustment (SSBA), http://cs.unc.edu/~cmzach/opensource.html.

Bundler, structure from motion for unordered image collections, http://phototour.cs.washington.edu/bundler/ (Snavely, Seitz, and Szeliski 2006).

Chapter 8: Dense motion estimation

Optical flow software, http://www.cs.brown.edu/~black/code.html (Black and Anandan 1996).

Optical flow using total variation and conjugate gradient descent, http://people.csail.mit.edu/celiu/OpticalFlow/ (Liu 2009).

TV-L1 optical flow on the GPU, http://cs.unc.edu/~cmzach/opensource.html (Zach, Pock, and Bischof 2007a).

elastix: a toolbox for rigid and nonrigid registration of images, http://elastix.isi.uu.nl/ (Klein, Staring, and Pluim 2007).

Deformable image registration using discrete optimization, http://www.mrf-registration.net/deformable/index.html (Glocker, Komodakis, Tziritas *et al.* 2008).

Chapter 9: Image stitching

Microsoft Research Image Compositing Editor for stitching images, http://research.microsoft.com/en-us/um/redmond/groups/ivm/ice/.

Chapter 10: Computational photography

HDRShop software for combining bracketed exposures into high-dynamic range radiance images, http://projects.ict.usc.edu/graphics/HDRShop/.

Super-resolution code, http://www.robots.ox.ac.uk/~vgg/software/SR/ (Pickup 2007; Pickup, Capel, Roberts *et al.* 2007, 2009).

Chapter 11: Stereo correspondence

StereoMatcher, standalone C++ stereo matching code, http://vision.middlebury.edu/stereo/code/ (Scharstein and Szeliski 2002).

Patch-based multi-view stereo software (PMVS Version 2), http://grail.cs.washington.edu/software/pmvs/ (Furukawa and Ponce 2011).

Chapter 12: 3D reconstruction

Scanalyze: a system for aligning and merging range data, http://graphics.stanford.edu/software/scanalyze/ (Curless and Levoy 1996).

MeshLab: software for processing, editing, and visualizing unstructured 3D triangular meshes, http://meshlab.sourceforge.net/.

VRML viewers (various) are also a good way to visualize texture-mapped 3D models.

Section 12.6.4: Whole body modeling and tracking

Bayesian 3D person tracking, http://www.cs.brown.edu/~black/code.html (Sidenbladh, Black, and Fleet 2000; Sidenbladh and Black 2003).

HumanEva: baseline code for the tracking of articulated human motion, http://vision.cs.brown.edu/humaneva/ (Sigal, Balan, and Black 2010).

Section 14.1.1: Face detection

Sample face detection code and evaluation tools, http://vision.ai.uiuc.edu/mhyang/face-detection-survey.html.

Section 14.1.2: Pedestrian detection

A simple object detector with boosting, http://people.csail.mit.edu/torralba/shortCourseRLOC/boosting/boosting.html (Hastie, Tibshirani, and Friedman 2001; Torralba, Murphy, and Freeman 2007).

Discriminatively trained deformable part models, http://people.cs.uchicago.edu/~pff/latent/ (Felzenszwalb, Girshick, McAllester *et al.* 2010).

Upper-body detector, http://www.robots.ox.ac.uk/~vgg/software/UpperBody/ (Ferrari, Marin-Jimenez, and Zisserman 2008).

2D articulated human pose estimation software, http://www.vision.ee.ethz.ch/~calvin/articulated_human_pose_estimation_code/ (Eichner and Ferrari 2009).

Section 14.2.2: Active appearance and 3D shape models

AAMtools: An active appearance modeling toolbox, http://cvsp.cs.ntua.gr/software/AAMtools/ (Papandreou and Maragos 2008).

Section 14.3: Instance recognition

FASTANN and FASTCLUSTER for approximate k-means (AKM), http://www.robots.ox.ac.uk/~vgg/software/ (Philbin, Chum, Isard *et al.* 2007).

Feature matching using fast approximate nearest neighbors, http://people.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN (Muja and Lowe 2009).

Section 14.4.1: Bag of words

Two bag of words classifiers, http://people.csail.mit.edu/fergus/iccv2005/bagwords.html (Fei-Fei and Perona 2005; Sivic, Russell, Efros *et al.* 2005).

Bag of features and hierarchical k-means, http://www.vlfeat.org/ (Nistér and Stewénius 2006; Nowak, Jurie, and Triggs 2006).

Section 14.4.2: Part-based models

A simple parts and structure object detector, http://people.csail.mit.edu/fergus/iccv2005/partsstructure.html (Fischler and Elschlager 1973; Felzenszwalb and Huttenlocher 2005).

Section 14.5.1: Machine learning software

Support vector machines (SVM) software (http://www.support-vector-machines.org/SVM_soft.html) has pointers to lots of SVM libraries, including SVM^{light}, http://svmlight.joachims.org/; LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (Fan, Chen, and Lin 2005); and LIBLINEAR, http://www.csie.ntu.edu.tw/~cjlin/liblinear/ (Fan, Chang, Hsieh *et al.* 2008).

Kernel Machines: links to SVM, Gaussian processes, boosting, and other machine learning algorithms, http://www.kernel-machines.org/software.

Multiple kernels for image classification, http://www.robots.ox.ac.uk/~vgg/software/MKL/ (Varma and Ray 2007; Vedaldi, Gulshan, Varma *et al.* 2009).

Appendix A.1–A.2: Matrix decompositions and linear least squares[2]

BLAS (Basic Linear Algebra Subprograms), http://www.netlib.org/blas/ (Blackford, Demmel, Dongarra *et al.* 2002).

LAPACK (Linear Algebra PACKage), http://www.netlib.org/lapack/ (Anderson, Bai, Bischof *et al.* 1999).

GotoBLAS, http://www.tacc.utexas.edu/tacc-projects/.

ATLAS (Automatically Tuned Linear Algebra Software), http://math-atlas.sourceforge.net/ (Demmel, Dongarra, Eijkhout *et al.* 2005).

Intel Math Kernel Library (MKL), http://software.intel.com/en-us/intel-mkl/.

AMD Core Math Library (ACML), http://developer.amd.com/cpu/Libraries/acml/Pages/default.aspx.

Robust PCA code, http://www.salle.url.edu/~ftorre/papers/rpca2.html (De la Torre and Black 2003).

Appendix A.3: Non-linear least squares

MINPACK, http://www.netlib.org/minpack/.

levmar: Levenberg–Marquardt nonlinear least squares algorithms, http://www.ics.forth.gr/~lourakis/levmar/ (Madsen, Nielsen, and Tingleff 2004).

Appendix A.4–A.5: Direct and iterative sparse matrix solvers

SuiteSparse (various reordering algorithms, CHOLMOD) and SuiteSparse QR, http://www.cise.ufl.edu/research/sparse/SuiteSparse/ (Davis 2006, 2008).

PARDISO (iterative and sparse direct solution), http://www.pardiso-project.org/.

TAUCS (sparse direct, iterative, out of core, preconditioners), http://www.tau.ac.il/~stoledo/taucs/.

HSL Mathematical Software Library, http://www.hsl.rl.ac.uk/index.html.

[2] Thanks to Sameer Agarwal for suggesting and describing most of these sites.

Templates for the solution of linear systems, http://www.netlib.org/linalg/html_templates/Templates.html (Barrett, Berry, Chan *et al.* 1994). Download the PDF for instructions on how to get the software.

ITSOL, MIQR, and other sparse solvers, http://www-users.cs.umn.edu/~saad/software/ (Saad 2003).

ILUPACK, http://www-public.tu-bs.de/~bolle/ilupack/.

Appendix B: Bayesian modeling and inference

Middlebury source code for MRF minimization, http://vision.middlebury.edu/MRF/code/ (Szeliski, Zabih, Scharstein *et al.* 2008).

C++ code for efficient belief propagation for early vision, http://people.cs.uchicago.edu/~pff/bp/ (Felzenszwalb and Huttenlocher 2006).

FastPD MRF optimization code, http://www.csd.uoc.gr/~komod/FastPD (Komodakis and Tziritas 2007a; Komodakis, Tziritas, and Paragios 2008)

Gaussian noise generation. A lot of basic software packages come with a uniform random noise generator (e.g., the `rand()` routine in Unix), but not all have a Gaussian random noise generator. To compute a normally distributed random variable, you can use the Box–Muller transform (Box and Muller 1958), whose C code is given in Algorithm C.1—note that this routine returns pairs of random variables. Alternative methods for generating Gaussian random numbers are given by Thomas, Luk, Leong *et al.* (2007).

Pseudocolor generation. In many applications, it is convenient to be able to visualize the set of labels assigned to an image (or to image features such as lines). One of the easiest ways to do this is to assign a unique color to each integer label. In my work, I have found it convenient to distribute these labels in a quasi-uniform fashion around the RGB color cube using the following idea.

For each (non-negative) label value, consider the bits as being split among the three color channels, e.g., for a nine-bit value, the bits could be labeled RGBRGBRGB. After collecting each of the three color values, *reverse* the bits so that the low-order bits vary the most quickly. In practice, for eight-bit color channels, this bit reverse can be stored in a table or a complete table mapping from labels to pseudocolors (say with 4092 entries) can be pre-computed. Figure 8.16 shows an example of such a pseudo-color mapping.

GPU implementation

The advent of programmable GPUs with capabilities such as pixel shaders and compute shaders has led to the development of fast computer vision algorithms for real-time applications such as segmentation, tracking, stereo, and motion estimation (Pock, Unger, Cremers *et al.* 2008; Vineet and Narayanan 2008; Zach, Gallup, and Frahm 2008). A good source for learning about such algorithms is the CVPR 2008 workshop on Visual Computer Vision on GPUs (CVGPU), http://www.cs.unc.edu/~jmf/Workshop_on_Computer_Vision_on_GPU.html, whose papers can be found on the CVPR 2008 proceedings DVD. Additional sources

```
double urand()
{
  return ((double) rand()) / ((double) RAND_MAX);
}
void grand(double& g1, double& g2)
{
#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif // M_PI

  double n1 = urand();
  double n2 = urand();
  double x1 = n1 + (n1 == 0); /* guard against log(0) */
  double sqlogn1 = sqrt(-2.0 * log (x1));
  double angl = (2.0 * M_PI) * n2;
  g1 = sqlogn1 * cos(angl);
  g2 = sqlogn1 * sin(angl);
}
```

Algorithm C.1 C algorithm for Gaussian random noise generation, using the Box–Muller transform.

for GPU algorithms include the GPGPU Web site and workshops, http://gpgpu.org/, and the OpenVIDIA Web site, http://openvidia.sourceforge.net/index.php/OpenVIDIA.

C.3 Slides and lectures

As I mentioned in the preface, I hope to post slides corresponding to the material in the book. Until these are ready, your best bet is to look at the slides from the courses I have co-taught at the University of Washington, as well as related courses that have used a similar syllabus. Here is a partial list of such courses:

UW 455: Undergraduate Computer Vision, http://www.cs.washington.edu/education/courses/455/.

UW 576: Graduate Computer Vision, http://www.cs.washington.edu/education/courses/576/.

Stanford CS233B: Introduction to Computer Vision, http://vision.stanford.edu/teaching/cs223b/.

MIT 6.869: Advances in Computer Vision, http://people.csail.mit.edu/torralba/courses/6.869/6.869.computervision.htm.

Berkeley CS 280: Computer Vision, http://www.eecs.berkeley.edu/~trevor/CS280.html.

UNC COMP 776: Computer Vision, http://www.cs.unc.edu/~lazebnik/spring10/.

Middlebury CS 453: Computer Vision, http://www.cs.middlebury.edu/~schar/courses/cs453-s10/.

Related courses have also been taught on the topic of Computational Photography, e.g.,

CMU 15-463: Computational Photography, http://graphics.cs.cmu.edu/courses/15-463/.

MIT 6.815/6.865: Advanced Computational Photography, http://stellar.mit.edu/S/course/6/sp09/6.815/.

Stanford CS 448A: Computational photography on cell phones, http://graphics.stanford.edu/courses/cs448a-10/.

SIGGRAPH courses on Computational Photography, http://web.media.mit.edu/~raskar/photo/.

There is also an excellent set of on-line lectures available on a range of computer vision topics, such as belief propagation and graph cuts, at the UW-MSR Course of Vision Algorithms http://www.cs.washington.edu/education/courses/577/04sp/.

C.4 Bibliography

While a bibliography (BibTex .bib file) for all of the references cited in this book is available on the book's Web site, a much more comprehensive partially annotated bibliography of nearly *all* computer vision publications is maintained by Keith Price at http://iris.usc.edu/Vision-Notes/bibliography/contents.html. There is also a searchable computer graphics bibliography at http://www.siggraph.org/publications/bibliography/. Additional good sources for technical papers are Google Scholar and CiteSeer^{X}.

References

Abdel-Hakim, A. E. and Farag, A. A. (2006). CSIFT: A SIFT descriptor with color invariant characterstics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 1978–1983, New York City, NY.

Adelson, E. H. and Bergen, J. (1991). The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pp. 3–20.

Adelson, E. H., Simoncelli, E., and Hingorani, R. (1987). Orthogonal pyramid transforms for image coding. In *SPIE Vol. 845, Visual Communications and Image Processing II*, pp. 50–58, Cambridge, Massachusetts.

Adiv, G. (1989). Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–490.

Agarwal, A. and Triggs, B. (2006). Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58.

Agarwal, S. and Roth, D. (2002). Learning a sparse representation for object detection. In *Seventh European Conference on Computer Vision (ECCV 2002)*, pp. 113–127, Copenhagen.

Agarwal, S., Snavely, N., Seitz, S. M., and Szeliski, R. (2010). Bundle adjustment in the large. In *Eleventh European Conference on Computer Vision (ECCV 2010)*, Heraklion, Crete.

Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., and Szeliski, R. (2009). Building Rome in a day. In *Twelfth IEEE International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Agarwal, S., Furukawa, Y., Snavely, N., Curless, B., Seitz, S. M., and Szeliski, R. (2010). Reconstructing Rome. *Computer*, 43(6):40–47.

Agarwala, A. (2007). Efficient gradient-domain compositing using quadtrees. *ACM Transactions on Graphics*, 26(3).

Agarwala, A., Hertzmann, A., Seitz, S., and Salesin, D. (2004). Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 23(3):584–591.

Agarwala, A., Agrawala, M., Cohen, M., Salesin, D., and Szeliski, R. (2006). Photographing long scenes with multi-viewpoint panoramas. *ACM Transactions on Graphics (Proc. SIGGRAPH 2006)*, 25(3):853–861.

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D. H., and Cohen, M. F. (2004). Interactive digital photomontage. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 23(3):292–300.

Agarwala, A., Zheng, K. C., Pal, C., Agrawala, M., Cohen, M., Curless, B., Salesin, D., and Szeliski, R. (2005). Panoramic video textures. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):821–827.

Aggarwal, J. K. and Nandhakumar, N. (1988). On the computation of motion from sequences of images—a review. *Proceedings of the IEEE*, 76(8):917–935.

Agin, G. J. and Binford, T. O. (1976). Computer description of curved objects. *IEEE Transactions on Computers*, C-25(4):439–449.

Ahonen, T., Hadid, A., and Pietikäinen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041.

Akenine-Möller, T. and Haines, E. (2002). *Real-Time Rendering*. A K Peters, Wellesley, Massachusetts, second edition.

Al-Baali, M. and Fletcher., R. (1986). An efficient line search for nonlinear least squares. *Journal Journal of Optimization Theory and Applications*, 48(3):359–377.

Alahari, K., Kohli, P., and Torr, P. (2011). Dynamic hybrid algorithms for discrete MAP MRF inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. T. (2003). Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15.

Aliaga, D. G., Funkhouser, T., Yanovsky, D., and Carlbom, I. (2003). Sea of images. *IEEE Computer Graphics and Applications*, 23(6):22–30.

Allen, B., Curless, B., and Popović, Z. (2003). The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, 22(3):587–594.

Allgower, E. L. and Georg, K. (2003). *Introduction to Numerical Continuation Methods*. Society for Industrial and Applied Mathematics.

Aloimonos, J. (1990). Perspective approximations. *Image and Vision Computing*, 8:177–192.

Alpert, S., Galun, M., Basri, R., and Brandt, A. (2007). Image segmentation by probabilistic bottom-up aggregation and cue integration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Amini, A. A., Weymouth, T. E., and Jain, R. C. (1990). Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867.

Anandan, P. (1984). Computing dense displacement fields with confidence measures in scenes containing occlusion. In *Image Understanding Workshop*, pp. 236–246, New Orleans.

Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310.

Anandan, P. and Irani, M. (2002). Factorization with uncertainty. *International Journal of Computer Vision*, 49(2-3):101–116.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J. W. *et al.* (1999). *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, 3rd edition.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43.

Andriluka, M., Roth, S., and Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Andriluka, M., Roth, S., and Schiele, B. (2010). Monocular 3d pose estimation and tracking by detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA.

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). SCAPE: Shape completion and animation of people. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):408–416.

Ansar, A., Castano, A., and Matthies, L. (2004). Enhanced real-time stereo using bilateral filtering. In *International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*.

Antone, M. and Teller, S. (2002). Scalable extrinsic calibration of omni-directional image networks. *International Journal of Computer Vision*, 49(2-3):143–174.

Arbeláez, P., Maire, M., Fowlkes, C., and Malik, J. (2010). *Contour Detection and Hierarchical Image Segmentation*. Technical Report UCB/EECS-2010-17, EECS Department, University of California, Berkeley. Submitted to PAMI.

Argyriou, V. and Vlachos, T. (2003). Estimation of sub-pixel motion using gradient cross-correlation. *Electronic Letters*, 39(13):980–982.

Arikan, O. and Forsyth, D. A. (2002). Interactive motion generation from examples. *ACM Transactions on Graphics*, 21(3):483–490.

Arnold, R. D. (1983). *Automated Stereo Perception*. Technical Report AIM-351, Artificial Intelligence Laboratory, Stanford University.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., and Wu, A. Y. (1998). An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45(6):891–923.

Ashdown, I. (1993). Near-field photometry: A new approach. *Journal of the Illuminating Engineering Society*, 22(1):163–180.

Atkinson, K. B. (1996). *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, Scotland, UK.

Aurich, V. and Weule, J. (1995). Non-linear Gaussian filters performing edge preserving diffusion. In *17th DAGM-Symposium*, pp. 538–545, Bielefeld.

Avidan, S. (2001). Support vector tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, pp. 283–290, Kauai, Hawaii.

Avidan, S., Baker, S., and Shan, Y. (2010). Special issue on Internet Vision. *Proceedings of the IEEE*, 98(8):1367–1369.

Axelsson, O. (1996). *Iterative Solution Methods*. Cambridge University Press, Cambridge.

Ayache, N. (1989). *Vision Stéréoscopique et Perception Multisensorielle*. InterEditions, Paris.

Azarbayejani, A. and Pentland, A. P. (1995). Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575.

Azuma, R. T., Baillot, Y., Behringer, R., Feiner, S. K., Julier, S., and MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47.

Bab-Hadiashar, A. and Suter, D. (1998a). Robust optic flow computation. *International Journal of Computer Vision*, 29(1):59–77.

Bab-Hadiashar, A. and Suter, D. (1998b). Robust total least squares based optic flow computation. In *Asian Conference on Computer Vision (ACCV'98)*, pp. 566–573, Hong Kong.

Badra, F., Qumsieh, A., and Dudek, G. (1998). Rotation and zooming in image mosaicing. In *IEEE Workshop on Applications of Computer Vision (WACV'98)*, pp. 50–55, Princeton.

Bae, S., Paris, S., and Durand, F. (2006). Two-scale tone management for photographic look. *ACM Transactions on Graphics*, 25(3):637–645.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley.

Bai, X. and Sapiro, G. (2009). Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International Journal of Computer Vision*, 82(2):113–132.

Bajcsy, R. and Kovacic, S. (1989). Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46(1):1–21.

Baker, H. H. (1977). Three-dimensional modeling. In *Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*, pp. 649–655.

Baker, H. H. (1982). *Depth from Edge and Intensity Based Stereo*. Technical Report AIM-347, Artificial Intelligence Laboratory, Stanford University.

Baker, H. H. (1989). Building surfaces of evolution: The weaving wall. *International Journal of Computer Vision*, 3(1):50–71.

Baker, H. H. and Binford, T. O. (1981). Depth from edge and intensity based stereo. In *IJCAI81*, pp. 631–636.

Baker, H. H. and Bolles, R. C. (1989). Generalizing epipolar-plane image analysis on the spatiotemporal surface. *International Journal of Computer Vision*, 3(1):33–49.

Baker, S. and Kanade, T. (2002). Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183.

Baker, S. and Matthews, I. (2004). Lucas-Kanade 20 years on: A unifying framework: Part 1: The quantity approximated, the warp update rule, and the gradient descent approximation. *International Journal of Computer Vision*, 56(3):221–255.

Baker, S. and Nayar, S. (1999). A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 5(2):175–196.

Baker, S. and Nayar, S. K. (2001). Single viewpoint catadioptric cameras. In Benosman, R. and Kang, S. B. (eds), *Panoramic Vision: Sensors, Theory, and Applications*, pp. 39–71, Springer, New York.

Baker, S., Gross, R., and Matthews, I. (2003). *Lucas-Kanade 20 Years On: A Unifying Framework: Part 3*. Technical Report CMU-RI-TR-03-35, The Robotics Institute, Carnegie Mellon University.

Baker, S., Gross, R., and Matthews, I. (2004). *Lucas-Kanade 20 Years On: A Unifying Framework: Part 4*. Technical Report CMU-RI-TR-04-14, The Robotics Institute, Carnegie Mellon University.

Baker, S., Szeliski, R., and Anandan, P. (1998). A layered approach to stereo reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 434–441, Santa Barbara.

Baker, S., Gross, R., Ishikawa, T., and Matthews, I. (2003). *Lucas-Kanade 20 Years On: A Unifying Framework: Part 2*. Technical Report CMU-RI-TR-03-01, The Robotics Institute, Carnegie Mellon University.

Baker, S., Black, M., Lewis, J. P., Roth, S., Scharstein, D., and Szeliski, R. (2007). A database and evaluation methodology for optical flow. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R. (2009). *A Database and Evaluation Methodology for Optical Flow*. Technical Report MSR-TR-2009-179, Microsoft Research.

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary patterns. *Pattern Recognition*, 13(2):111–122.

Ballard, D. H. and Brown, C. M. (1982). *Computer Vision*. Prentice-Hall, Englewood Cliffs, New Jersey.

Banno, A., Masuda, T., Oishi, T., and Ikeuchi, K. (2008). Flying laser range sensor for large-scale site-modeling and its applications in Bayon digital archival project. *International Journal of Computer Vision*, 78(2-3):207–222.

Bar-Hillel, A., Hertz, T., and Weinshall, D. (2005). Object class recognition by boosting a part based model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 701–708, San Diego, CA.

Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., and Werman, M. (2001). Texture mixing and texture movie synthesis using statistical learning. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):120–135.

Bar-Shalom, Y. and Fortmann, T. E. (1988). *Tracking and data association*. Academic Press, Boston.

Barash, D. (2002). A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847.

Barash, D. and Comaniciu, D. (2004). A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Vision Computing*, 22(1):73–81.

Barbu, A. and Zhu, S.-C. (2003). Graph partition by Swendsen–Wang cuts. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 320–327, Nice, France.

Barbu, A. and Zhu, S.-C. (2005). Generalizing Swendsen–Wang to sampling arbitrary posterior probabilities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1239–1253.

Barkans, A. C. (1997). High quality rendering using the Talisman architecture. In *Proceedings of the Eurographics Workshop on Graphics Hardware*.

Barnard, S. T. (1989). Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3(1):17–32.

Barnard, S. T. and Fischler, M. A. (1982). Computational stereo. *Computing Surveys*, 14(4):553–572.

Barnes, C., Jacobs, D. E., Sanders, J., Goldman, D. B., Rusinkiewicz, S., Finkelstein, A., and Agrawala, M. (2008). Video puppetry: A performative interface for cutout animation. *ACM Transactions on Graphics*, 27(5).

Barreto, J. P. and Daniilidis, K. (2005). Fundamental matrix for cameras with radial distortion. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 625–632, Beijing, China.

Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J. *et al*. (1994). *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA.

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77.

Barrow, H. G. and Tenenbaum, J. M. (1981). Computational vision. *Proceedings of the IEEE*, 69(5):572–595.

Bartels, R. H., Beatty, J. C., and Barsky, B. A. (1987). *An Introduction to Splines for use in Computer Graphics and Geeometric Modeling*. Morgan Kaufmann Publishers, Los Altos.

Bartoli, A. (2003). Towards gauge invariant bundle adjustment: A solution based on gauge dependent damping. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 760–765, Nice, France.

Bartoli, A. and Sturm, P. (2003). Multiple-view structure and motion from line correspondences. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 207–212, Nice, France.

Bartoli, A., Coquerelle, M., and Sturm, P. (2004). A framework for pencil-of-points structure-from-motion. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 28–40, Prague.

Bascle, B., Blake, A., and Zisserman, A. (1996). Motion deblurring and super-resolution from an image sequence. In *Fourth European Conference on Computer Vision (ECCV'96)*, pp. 573–582, Cambridge, England.

Bathe, K.-J. (2007). *Finite Element Procedures*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Batra, D., Sukthankar, R., and Chen, T. (2008). Learning class-specific affinities for image labelling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Baudisch, P., Tan, D., Steedly, D., Rudolph, E., Uyttendaele, M., Pal, C., and Szeliski, R. (2006). An exploration of user interface designs for real-time panoramic photography. *Australian Journal of Information Systems*, 13(2).

Baumberg, A. (2000). Reliable feature matching across widely separated views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, pp. 774–781, Hilton Head Island.

Baumberg, A. M. and Hogg, D. C. (1996). Generating spatiotemporal models from examples. *Image and Vision Computing*, 14(8):525–532.

Baumgart, B. G. (1974). *Geometric Modeling for Computer Vision*. Technical Report AIM-249, Artificial Intelligence Laboratory, Stanford University.

Bay, H., Ferrari, V., and Van Gool, L. (2005). Wide-baseline stereo matching with line segments. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 329–336, San Diego, CA.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded up robust features. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 404–417.

Bayer, B. E. (1976). Color imaging array. US Patent No. 3,971,065.

Beardsley, P., Torr, P., and Zisserman, A. (1996). 3D model acquisition from extended image sequences. In *Fourth European Conference on Computer Vision (ECCV'96)*, pp. 683–695, Cambridge, England.

Beare, R. (2006). A locally constrained watershed transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1063–1074.

Becker, S. and Bove, V. M. (1995). Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In *SPIE Vol. 2410, Visual Data Exploration and Analysis II*, pp. 447–461, San Jose.

Beier, T. and Neely, S. (1992). Feature-based image metamorphosis. *Computer Graphics (SIGGRAPH '92)*, 26(2):35–42.

Beis, J. S. and Lowe, D. G. (1999). Indexing without invariants in 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1000–1015.

Belhumeur, P. N. (1996). A Bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260.

Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720.

Belongie, S. and Malik, J. (1998). Finding boundaries in natural images: a new method using point descriptors and area completion. In *Fifth European Conference on Computer Vision (ECCV'98)*, pp. 751–766, Freiburg, Germany.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522.

Belongie, S., Fowlkes, C., Chung, F., and Malik, J. (2002). Spectral partitioning with indefinite kernels using the Nyström extension. In *Seventh European Conference on Computer Vision (ECCV 2002)*, pp. 531–543, Copenhagen.

Bennett, E., Uyttendaele, M., Zitnick, L., Szeliski, R., and Kang, S. B. (2006). Video and image Bayesian demosaicing with a two color image prior. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 508–521, Graz.

Benosman, R. and Kang, S. B. (eds). (2001). *Panoramic Vision: Sensors, Theory, and Applications*, Springer, New York.

Berg, T. (2008). Internet vision. SUNY Stony Brook Course CSE 690, http://www.tamaraberg.com/teaching/Fall_08/.

Berg, T. and Forsyth, D. (2006). Animals on the web. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 1463–1470, New York City, NY.

Bergen, J. R., Anandan, P., Hanna, K. J., and Hingorani, R. (1992). Hierarchical model-based motion estimation. In *Second European Conference on Computer Vision (ECCV'92)*, pp. 237–252, Santa Margherita Liguere, Italy.

Bergen, J. R., Burt, P. J., Hingorani, R., and Peleg, S. (1992). A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–896.

Berger, J. O. (1993). *Statistical Decision Theory and Bayesian Analysis*. Springer, New York, second edition.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 417–424.

Bertalmio, M., Vese, L., Sapiro, G., and Osher, S. (2003). Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889.

Bertero, M., Poggio, T. A., and Torre, V. (1988). Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48(3):259–302.

Besl, P. (1989). Active optical range imaging sensors. In Sanz, J. L. (ed.), *Advances in Machine Vision*, chapter 1, pp. 1–63, Springer-Verlag.

Besl, P. J. and Jain, R. C. (1985). Three-dimensional object recognition. *Computing Surveys*, 17(1):75–145.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Betrisey, C., Blinn, J. F., Dresevic, B., Hill, B., Hitchcock, G. *et al.* (2000). Displaced filtering for patterned displays. In *Society for Information Display Symposium,*, pp. 296–299.

Beymer, D. (1996). Feature correspondence by interleaving shape and texture computations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pp. 921–928, San Francisco.

Bhat, D. N. and Nayar, S. K. (1998). Ordinal measures for image correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):415–423.

Bickel, B., Botsch, M., Angst, R., Matusik, W., Otaduy, M., Pfister, H., and Gross, M. (2007). Multi-scale capture of facial geometry and motion. *ACM Transactions on Graphics*, 26(3).

Billinghurst, M., Kato, H., and Poupyrev, I. (2001). The MagicBook: a transitional AR interface. *Computers & Graphics*, 25:745–753.

Bimber, O. (2006). Computational photography—the next big step. *Computer*, 39(8):28–29.

Birchfield, S. and Tomasi, C. (1998). A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406.

Birchfield, S. and Tomasi, C. (1999). Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293.

Birchfield, S. T., Natarajan, B., and Tomasi, C. (2007). Correspondence as energy-based segmentation. *Image and Vision Computing*, 25(8):1329–1340.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York, NY.

Bitouk, D., Kumar, N., Dhillon, S., Belhumeur, P., and Nayar, S. K. (2008). Face swapping: Automatically replacing faces in photographs. *ACM Transactions on Graphics*, 27(3).

Björck, A. (1996). *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics.

Björck, A. and Dahlquist, G. (2010). *Numerical Methods in Scientific Computing*. Volume II, Society for Industrial and Applied Mathematics.

Black, M., Yacoob, Y., Jepson, A. D., and Fleet, D. J. (1997). Learning parameterized models of image motion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 561–567, San Juan, Puerto Rico.

Black, M. J. and Anandan, P. (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104.

Black, M. J. and Jepson, A. D. (1996). Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986.

Black, M. J. and Jepson, A. D. (1998). EigenTracking: robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84.

Black, M. J. and Rangarajan, A. (1996). On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91.

Black, M. J., Sapiro, G., Marimont, D. H., and Heeger, D. (1998). Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432.

Blackford, L. S., Demmel, J., Dongarra, J., Duff, I., Hammarling, S. *et al.* (2002). An updated set of basic linear algebra subprograms (BLAS). *ACM Transactions on Mathematical Software*, 28(2):135–151.

Blake, A. and Isard, M. (1998). *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer Verlag, London.

Blake, A. and Zisserman, A. (1987). *Visual Reconstruction*. MIT Press, Cambridge, Massachusetts.

Blake, A., Curwen, R., and Zisserman, A. (1993). A framework for spatio-temporal control in the tracking of visual contour. *International Journal of Computer Vision*, 11(2):127–145.

Blake, A., Kohli, P., and Rother, C. (eds). (2010). *Advances in Markov Random Fields*, MIT Press.

Blake, A., Zimmerman, A., and Knowles, G. (1985). Surface descriptions from stereo and shading. *Image and Vision Computing*, 3(4):183–191.

Blake, A., Rother, C., Brown, M., Perez, P., and Torr, P. (2004). Interactive image segmentation using an adaptive GMMRF model. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 428–441, Prague.

Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *ACM SIGGRAPH 1999 Conference Proceedings*, pp. 187–194.

Blanz, V. and Vetter, T. (2003). Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25():1063–1074.

Bleyer, M., Gelautz, M., Rother, C., and Rhemann, C. (2009). A stereo approach that handles the matting problem via image warping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Blinn, J. (1998). *Dirty Pixels*. Morgan Kaufmann Publishers, San Francisco.

Blinn, J. F. (1994a). Jim Blinn's corner: Compositing, part 1: Theory. *IEEE Computer Graphics and Applications*, 14(5):83–87.

Blinn, J. F. (1994b). Jim Blinn's corner: Compositing, part 2: Practice. *IEEE Computer Graphics and Applications*, 14(6):78–82.

Blinn, J. F. and Newell, M. E. (1976). Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–547.

Blostein, D. and Ahuja, N. (1987). Shape from texture: Integrating texture-element extraction and surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1233–1251.

Bobick, A. F. (1997). Movement, activity and action: the role of knowledge in the perception of motion. *Proceedings of the Royal Society of London*, B 352:1257–1265.

Bobick, A. F. and Intille, S. S. (1999). Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200.

Boden, M. A. (2006). *Mind As Machine: A History of Cognitive Science*. Oxford University Press, Oxford, England.

Bogart, R. G. (1991). View correlation. In Arvo, J. (ed.), *Graphics Gems II*, pp. 181–190, Academic Press, Boston.

Boiman, O., Shechtman, E., and Irani, M. (2008). In defense of nearest-neighbor based image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Boissonat, J.-D. (1984). Representing 2D and 3D shapes with the Delaunay triangulation. In *Seventh International Conference on Pattern Recognition (ICPR'84)*, pp. 745–748, Montreal, Canada.

Bolles, R. C., Baker, H. H., and Hannah, M. J. (1993). The JISCT stereo evaluation. In *Image Understanding Workshop*, pp. 263–274.

Bolles, R. C., Baker, H. H., and Marimont, D. H. (1987). Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1:7–55.

Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585.

Borenstein, E. and Ullman, S. (2008). Combined top-down/bottom-up segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2109–2125.

Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34(3):227–248.

Bouchard, G. and Triggs, B. (2005). Hierarchical part-based visual object categorization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 709–714, San Diego, CA.

Bougnoux, S. (1998). From projective to Euclidean space under any practical situation, a criticism of self-calibration. In *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 790–798, Bombay.

Bouguet, J.-Y. and Perona, P. (1999). 3D photography using shadows in dual-space geometry. *International Journal of Computer Vision*, 35(2):129–149.

Boult, T. E. and Kender, J. R. (1986). Visual surface reconstruction using sparse depth data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'86)*, pp. 68–76, Miami Beach.

Bourdev, L. and Malik, J. (2009). Poselets: Body part detectors trained using 3D human pose annotations. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Bovik, A. (ed.). (2000). *Handbook of Image and Video Processing*, Academic Press, San Diego.

Bowyer, K. W., Kranenburg, C., and Dougherty, S. (2001). Edge detector evaluation using empirical ROC curves. *Computer Vision and Image Understanding*, 84(1):77–103.

Box, G. E. P. and Muller, M. E. (1958). A note on the generation of random normal deviates. *Annals of Mathematical Statistics*, 29(2).

Boyer, E. and Berger, M. O. (1997). 3D surface reconstruction using occluding contours. *International Journal of Computer Vision*, 22(3):219–233.

Boykov, Y. and Funka-Lea, G. (2006). Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70(2):109–131.

Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 105–112, Vancouver, Canada.

Boykov, Y. and Kolmogorov, V. (2003). Computing geodesics and minimal surfaces via graph cuts. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 26–33, Nice, France.

Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137.

Boykov, Y. and Kolmogorov, V. (2010). Basic graph cut algorithms. In Blake, A., Kohli, P., and Rother, C. (eds), *Advances in Markov Random Fields*, MIT Press.

Boykov, Y., Veksler, O., and Zabih, R. (1998). A variable window approach to early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1283–1294.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

Boykov, Y., Veksler, O., and Zabih, R. (2010). Optimizing multi-label MRFs by move making algorithms. In Blake, A., Kohli, P., and Rother, C. (eds), *Advances in Markov Random Fields*, MIT Press.

Boykov, Y., Kolmogorov, V., Cremers, D., and Delong, A. (2006). An integral solution to surface evolution PDEs via Geo-cuts. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 409–422.

Bracewell, R. N. (1986). *The Fourier Transform and its Applications*. McGraw-Hill, New York, 2nd edition.

Bradley, D., Boubekeur, T., and Heidrich, W. (2008). Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Bradsky, G. and Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Sebastopol, CA.

Brandt, A. (1986). Algebraic multigrid theory: The symmetric case. *Applied Mathematics and Computation*, 19(1-4):23–56.

Bregler, C. and Malik, J. (1998). Tracking people with twists and exponential maps. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 8–15, Santa Barbara.

Bregler, C., Covell, M., and Slaney, M. (1997). Video rewrite: Driving visual speech with audio. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 353–360.

Bregler, C., Malik, J., and Pullen, K. (2004). Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194.

Breu, H., Gil, J., Kirkpatrick, D., and Werman, M. (1995). Linear time Euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529–533.

Brice, C. R. and Fennema, C. L. (1970). Scene analysis using regions. *Artificial Intelligence*, 1(3-4):205–226.

Briggs, W. L., Henson, V. E., and McCormick, S. F. (2000). *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, Philadelphia, second edition.

Brillaut-O'Mahoney, B. (1991). New method for vanishing point detection. *Computer Vision, Graphics, and Image Processing*, 54(2):289–300.

Brinkmann, R. (2008). *The Art and Science of Digital Compositing*. Morgan Kaufmann Publishers, San Francisco, 2nd edition.

Brooks, R. A. (1981). Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17:285–348.

Brown, D. C. (1971). Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866.

Brown, L. G. (1992). A survey of image registration techniques. *Computing Surveys*, 24(4):325–376.

Brown, M. and Lowe, D. (2002). Invariant features from interest point groups. In *British Machine Vision Conference*, pp. 656–665, Cardiff, Wales.

Brown, M. and Lowe, D. (2003). Unsupervised 3D object recognition and reconstruction in unordered datasets. In *International Conference on 3D Imaging and Modelling*, pp. 1218–1225, Nice, France.

Brown, M. and Lowe, D. (2007). Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73.

Brown, M., Hartley, R., and Nistér, D. (2007). Minimal solutions for panoramic stitching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Brown, M., Szeliski, R., and Winder, S. (2004). *Multi-Image Matching Using Multi-Scale Oriented Patches*. Technical Report MSR-TR-2004-133, Microsoft Research.

Brown, M., Szeliski, R., and Winder, S. (2005). Multi-image matching using multi-scale oriented patches. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 510–517, San Diego, CA.

Brown, M. Z., Burschka, D., and Hager, G. D. (2003). Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008.

Brox, T., Bregler, C., and Malik, J. (2009). Large displacement optical flow. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 25–36, Prague.

Brubaker, S. C., Wu, J., Sun, J., Mullin, M. D., and Rehg, J. M. (2008). On the design of cascades of boosted ensembles for face detection. *International Journal of Computer Vision*, 77(1-3):65–86.

Bruhn, A., Weickert, J., and Schnörr, C. (2005). Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231.

Bruhn, A., Weickert, J., Kohlberger, T., and Schnörr, C. (2006). A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *International Journal of Computer Vision*, 70(3):257–277.

Buades, A., Coll, B., and Morel, J.-M. (2008). Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139.

Bălan, A. O. and Black, M. J. (2008). The naked truth: Estimating body shape under clothing. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 15–29, Marseilles.

Buchanan, A. and Fitzgibbon, A. (2005). Damped Newton algorithms for matrix factorization with missing data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 316–322, San Diego, CA.

Buck, I., Finkelstein, A., Jacobs, C., Klein, A., Salesin, D. H., Seims, J., Szeliski, R., and Toyama, K. (2000). Performance-driven hand-drawn animation. In *Symposium on Non Photorealistic Animation and Rendering*, pp. 101–108, Annecy.

Buehler, C., Bosse, M., McMillan, L., Gortler, S. J., and Cohen, M. F. (2001). Unstructured Lumigraph rendering. In *ACM SIGGRAPH 2001 Conference Proceedings*, pp. 425–432.

Bugayevskiy, L. M. and Snyder, J. P. (1995). *Map Projections: A Reference Manual*. CRC Press.

Burger, W. and Burge, M. J. (2008). *Digital Image Processing: An Algorithmic Introduction Using Java*. Springer, New York, NY.

Burl, M. C., Weber, M., and Perona, P. (1998). A probabilistic approach to object recognition using local photometry and global geometry. In *Fifth European Conference on Computer Vision (ECCV'98)*, pp. 628–641, Freiburg, Germany.

Burns, J. B., Hanson, A. R., and Riseman, E. M. (1986). Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):425–455.

Burns, P. D. and Williams, D. (1999). Using slanted edge analysis for color registration measurement. In *IS&T PICS Conference*, pp. 51–53.

Burt, P. J. and Adelson, E. H. (1983a). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31(4):532–540.

Burt, P. J. and Adelson, E. H. (1983b). A multiresolution spline with applications to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236.

Burt, P. J. and Kolczynski, R. J. (1993). Enhanced image capture through fusion. In *Fourth International Conference on Computer Vision (ICCV'93)*, pp. 173–182, Berlin, Germany.

Byröd, M. and øAström, K. (2009). Bundle adjustment using conjugate gradients with multiscale preconditioning. In *British Machine Vision Conference (BMVC 2009)*.

Cai, D., He, X., Hu, Y., Han, J., and Huang, T. (2007). Learning a spatially smooth subspace for face recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Campbell, N. D. F., Vogiatzis, G., Hernández, C., and Cipolla, R. (2008). Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 766–779, Marseilles.

Can, A., Stewart, C., Roysam, B., and Tanenbaum, H. (2002). A feature-based, robust, hierarchical algorithm for registering pairs of images of the curved human retina. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):347–364.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

Cao, Z., Yin, Q., Tang, X., and Sun, J. (2010). Face recognition with learning-based descriptor. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA.

Capel, D. (2004). *Image Mosaicing and Super-resolution*. *Distinguished Dissertation Series, British Computer Society*, Springer-Verlag.

Capel, D. and Zisserman, A. (1998). Automated mosaicing with super-resolution zoom. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 885–891, Santa Barbara.

Capel, D. and Zisserman, A. (2000). Super-resolution enhancement of text image sequences. In *Fifteenth International Conference on Pattern Recognition (ICPR'2000)*, pp. 600–605, Barcelona, Spain.

Capel, D. and Zisserman, A. (2003). Computer vision applied to super resolution. *IEEE Signal Processing Magazine*, 20(3):75–86.

Capel, D. P. (2001). *Super-resolution and Image Mosaicing*. Ph.D. thesis, University of Oxford.

Caprile, B. and Torre, V. (1990). Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139.

Carneiro, G. and Jepson, A. (2005). The distinctiveness, detectability, and robustness of local image features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 296–301, San Diego, CA.

Carneiro, G. and Lowe, D. (2006). Sparse flexible models of local features. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 29–43.

Carnevali, P., Coletti, L., and Patarnello, S. (1985). Image processing by simulated annealing. *IBM Journal of Research and Development*, 29(6):569–579.

Carranza, J., Theobalt, C., Magnor, M. A., and Seidel, H.-P. (2003). Free-viewpoint video of human actors. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, 22(3):569–577.

Carroll, R., Agrawala, M., and Agarwala, A. (2009). Optimizing content-preserving projections for wide-angle images. *ACM Transactions on Graphics*, 28(3).

Caselles, V., Kimmel, R., and Sapiro, G. (1997). Geodesic active contours. *International Journal of Computer Vision*, 21(1):61–79.

Catmull, E. and Smith, A. R. (1980). 3-D transformations of images in scanline order. *Computer Graphics (SIGGRAPH '80)*, 14(3):279–285.

Celniker, G. and Gossard, D. (1991). Deformable curve and surface finite-elements for free-form shape design. *Computer Graphics (SIGGRAPH '91)*, 25(4):257–266.

Chakrabarti, A., Scharstein, D., and Zickler, T. (2009). An empirical camera model for internet color vision. In *British Machine Vision Conference (BMVC 2009)*, London, UK.

Cham, T. J. and Cipolla, R. (1998). A statistical framework for long-range feature matching in uncalibrated image mosaicing. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 442–447, Santa Barbara.

Cham, T.-J. and Rehg, J. M. (1999). A multiple hypothesis approach to figure tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 239–245, Fort Collins.

Champleboux, G., Lavallée, S., Sautot, P., and Cinquin, P. (1992). Accurate calibration of cameras and range imaging sensors, the NPBS method. In *IEEE International Conference on Robotics and Automation*, pp. 1552–1558, Nice, France.

Champleboux, G., Lavallée, S., Szeliski, R., and Brunie, L. (1992). From accurate range imaging sensor calibration to accurate model-based 3-D object localization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, pp. 83–89, Champaign, Illinois.

Chan, A. B. and Vasconcelos, N. (2009). Layered dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1862–1879.

Chan, T. F. and Vese, L. A. (1992). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277.

Chan, T. F., Osher, S., and Shen, J. (2001). The digital TV filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10(2):231–241.

Chang, M. M., Tekalp, A. M., and Sezan, M. I. (1997). Simultaneous motion estimation and segmentation. *IEEE Transactions on Image Processing*, 6(9):1326–1333.

Chaudhuri, S. (2001). *Super-Resolution Imaging*. Springer.

Chaudhuri, S. and Rajagopalan, A. N. (1999). *Depth from Defocus: A Real Aperture Imaging Approach*. Springer.

Cheeseman, P., Kanefsky, B., Hanson, R., and Stutz, J. (1993). *Super-Resolved Surface Reconstruction From Multiple Images*. Technical Report FIA-93-02, NASA Ames Research Center, Artificial Intelligence Branch.

Chellappa, R., Wilson, C., and Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):705–740.

Chen, B., Neubert, B., Ofek, E., Deussen, O., and Cohen, M. F. (2009). Integrated videos and maps for driving directions. In *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pp. 223–232, Victoria, BC, Canada, New York, NY, USA.

Chen, C.-Y. and Klette, R. (1999). Image stitching - comparisons and new techniques. In *Computer Analysis of Images and Patterns (CAIP'99)*, pp. 615–622, Ljubljana.

Chen, J. and Chen, B. (2008). Architectural modeling from sparsely scanned range data. *International Journal of Computer Vision*, 78(2-3):223–236.

Chen, J., Paris, S., and Durand, F. (2007). Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics*, 26(3).

Chen, S. and Williams, L. (1993). View interpolation for image synthesis. In *ACM SIGGRAPH 1993 Conference Proceedings*, pp. 279–288.

Chen, S. E. (1995). QuickTime VR – an image-based approach to virtual environment navigation. In *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 29–38, Los Angeles.

Chen, Y. and Medioni, G. (1992). Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155.

Cheng, L., Vishwanathan, S. V. N., and Zhang, X. (2008). Consistent image analogies using semi-supervised learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799.

Chiang, M.-C. and Boult, T. E. (1996). Efficient image warping and super-resolution. In *IEEE Workshop on Applications of Computer Vision (WACV'96)*, pp. 56–61, Sarasota.

Chiu, K. and Raskar, R. (2009). Computer vision on tap. In *Second IEEE Workshop on Internet Vision*, Miami Beach, Florida.

Chou, P. B. and Brown, C. M. (1990). The theory and practice of Bayesian image labeling. *International Journal of Computer Vision*, 4(3):185–210.

Christensen, G., Joshi, S., and Miller, M. (1997). Volumetric transformation of brain anatomy. *IEEE Transactions on Medical Imaging*, 16(6):864–877.

Christy, S. and Horaud, R. (1996). Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1098–1104.

Chuang, Y.-Y., Curless, B., Salesin, D. H., and Szeliski, R. (2001). A Bayesian approach to digital matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, pp. 264–271, Kauai, Hawaii.

Chuang, Y.-Y., Agarwala, A., Curless, B., Salesin, D. H., and Szeliski, R. (2002). Video matting of complex scenes. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002)*, 21(3):243–248.

Chuang, Y.-Y., Goldman, D. B., Curless, B., Salesin, D. H., and Szeliski, R. (2003). Shadow matting. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, 22(3):494–500.

Chuang, Y.-Y., Goldman, D. B., Zheng, K. C., Curless, B., Salesin, D. H., and Szeliski, R. (2005). Animating pictures with stochastic motion textures. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):853–860.

Chuang, Y.-Y., Zongker, D., Hindorff, J., Curless, B., Salesin, D. H., and Szeliski, R. (2000). Environment matting extensions: Towards higher accuracy and real-time capture. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 121–130, New Orleans.

Chui, C. K. (1992). *Wavelet Analysis and Its Applications*. Academic Press, New York.

Chum, O. and Matas, J. (2005). Matching with PROSAC—progressive sample consensus. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 220–226, San Diego, CA.

Chum, O. and Matas, J. (2010). Large-scale discovery of spatially related images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):371–377.

Chum, O., Philbin, J., and Zisserman, A. (2008). Near duplicate image detection: min-hash and tf-idf weighting. In *British Machine Vision Conference (BMVC 2008)*, Leeds, England.

Chum, O., Philbin, J., Sivic, J., Isard, M., and Zisserman, A. (2007). Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Cipolla, R. and Blake, A. (1990). The dynamic analysis of apparent contours. In *Third International Conference on Computer Vision (ICCV'90)*, pp. 616–623, Osaka, Japan.

Cipolla, R. and Blake, A. (1992). Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112.

Cipolla, R. and Giblin, P. (2000). *Visual Motion of Curves and Surfaces*. Cambridge University Press, Cambridge.

Cipolla, R., Drummond, T., and Robertson, D. P. (1999). Camera calibration from vanishing points in images of architectural scenes. In *British Machine Vision Conference (BMVC99)*.

Claus, D. and Fitzgibbon, A. (2005). A rational function lens distortion model for general cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 213–219, San Diego, CA.

Clowes, M. B. (1971). On seeing things. *Artificial Intelligence*, 2:79–116.

Cohen, L. D. and Cohen, I. (1993). Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147.

Cohen, M. and Wallace, J. (1993). *Radiosity and Realistic Image Synthesis*. Morgan Kaufmann.

Cohen, M. F. and Szeliski, R. (2006). The Moment Camera. *Computer*, 39(8):40–45.

Collins, R. T. (1996). A space-sweep approach to true multi-image matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pp. 358–363, San Francisco.

Collins, R. T. and Liu, Y. (2003). On-line selection of discriminative tracking features. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 346–352, Nice, France.

Collins, R. T. and Weiss, R. S. (1990). Vanishing point calculation as a statistical inference on the unit sphere. In *Third International Conference on Computer Vision (ICCV'90)*, pp. 400–403, Osaka, Japan.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

Comaniciu, D. and Meer, P. (2003). An algorithm for data-driven bandwidth selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):281–288.

Conn, A. R., Gould, N. I. M., and Toint, P. L. (2000). *Trust-Region Methods*. Society for Industrial and Applied Mathematics, Philadephia.

Cook, R. L. and Torrance, K. E. (1982). A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24.

Coorg, S. and Teller, S. (2000). Spherical mosaics with quaternions and dense correlation. *International Journal of Computer Vision*, 37(3):259–273.

Cootes, T., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

Cootes, T., Cooper, D., Taylor, C., and Graham, J. (1995). Active shape models—their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

Cootes, T., Taylor, C., Lanitis, A., Cooper, D., and Graham, J. (1993). Building and using flexible models incorporating grey-level information. In *Fourth International Conference on Computer Vision (ICCV'93)*, pp. 242–246, Berlin, Germany.

Cootes, T. F. and Taylor, C. J. (2001). Statistical models of appearance for medical image analysis and computer vision. In *Medical Imaging*.

Coquillart, S. (1990). Extended free-form deformations: A sculpturing tool for 3D geometric modeling. *Computer Graphics (SIGGRAPH '90)*, 24(4):187–196.

Cormen, T. H. (2001). *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts.

Cornelis, N., Leibe, B., Cornelis, K., and Van Gool, L. (2008). 3D urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision*, 78(2-3):121–141.

Corso, J. and Hager, G. (2005). Coherent regions for concise and stable image description. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 184–190, San Diego, CA.

Costeira, J. and Kanade, T. (1995). A multi-body factorization method for motion analysis. In *Fifth International Conference on Computer Vision (ICCV'95)*, pp. 1071–1076, Cambridge, Massachusetts.

Costen, N., Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1999). Simultaneous extraction of functional face subspaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 492–497, Fort Collins.

Couprie, C., Grady, L., Najman, L., and Talbot, H. (2009). Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Cour, T., Bénézit, F., and Shi, J. (2005). Spectral segmentation with multiscale graph decomposition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 1123–1130, San Diego, CA.

Cox, D., Little, J., and O'Shea, D. (2007). *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer.

Cox, I. J. (1994). A maximum likelihood N-camera stereo algorithm. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 733–739, Seattle.

Cox, I. J., Roy, S., and Hingorani, S. L. (1995). Dynamic histogram warping of image pairs for constant image brightness. In *IEEE International Conference on Image Processing (ICIP'95)*, pp. 366–369.

Cox, I. J., Hingorani, S. L., Rao, S. B., and Maggs, B. M. (1996). A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567.

Crandall, D. and Huttenlocher, D. (2007). Composite models of objects and scenes for category recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Crandall, D., Felzenszwalb, P., and Huttenlocher, D. (2005). Spatial priors for part-based recognition using statistical models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 10–17, San Diego, CA.

Crandall, D., Backstrom, L., Huttenlocher, D., and Kleinberg, J. (2009). Mapping the world's photos. In *18th Int. World Wide Web Conference*, pp. 761–770, Madrid.

Crandall, D. J. and Huttenlocher, D. P. (2006). Weakly supervised learning of part-based spatial models for visual object recognition. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 16–29.

Crane, R. (1997). *A Simplified Approach to Image Processing*. Prentice Hall, Upper Saddle River, NJ.

Craswell, N. and Szummer, M. (2007). Random walks on the click graph. In *ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 239–246, New York, NY.

Cremers, D. and Soatto, S. (2005). Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265.

Cremers, D., Rousson, M., and Deriche, R. (2007). A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215.

Crevier, D. (1993). *AI: The Tumultuous Search for Artificial Intelligence*. BasicBooks, New York, NY.

Criminisi, A., Pérez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212.

Criminisi, A., Reid, I., and Zisserman, A. (2000). Single view metrology. *International Journal of Computer Vision*, 40(2):123–148.

Criminisi, A., Sharp, T., and Blake, A. (2008). Geos: Geodesic image segmentation. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 99–112, Marseilles.

Criminisi, A., Cross, G., Blake, A., and Kolmogorov, V. (2006). Bilayer segmentation of live video. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 53–60, New York City, NY.

Criminisi, A., Shotton, J., Blake, A., and Torr, P. (2003). Gaze manipulation for one-to-one teleconferencing. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 191–198, Nice, France.

Criminisi, A., Kang, S. B., Swaminathan, R., Szeliski, R., and Anandan, P. (2005). Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer Vision and Image Understanding*, 97(1):51–85.

Criminisi, A., Shotton, J., Blake, A., Rother, C., and Torr, P. H. S. (2007). Efficient dense stereo with occlusion by four-state dynamic programming. *International Journal of Computer Vision*, 71(1):89–110.

Crow, F. C. (1984). Summed-area table for texture mapping. *Computer Graphics (SIGGRAPH '84)*, 18(3):207–212.

Crowley, J. L. and Stern, R. M. (1984). Fast computation of the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):212–222.

Csurka, G. and Perronnin, F. (2008). A simple high performance approach to semantic segmentation. In *British Machine Vision Conference (BMVC 2008)*, Leeds.

Csurka, G., Dance, C. R., Perronnin, F., and Willamowski, J. (2006). Generic visual categorization using weak geometry. In Ponce, J., Hebert, M., Schmid, C., and Zisserman, A. (eds), *Toward Category-Level Object Recognition*, pp. 207–224, Springer, New York.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, Prague.

Cui, J., Yang, Q., Wen, F., Wu, Q., Zhang, C., Van Gool, L., and Tang, X. (2008). Transductive object cutout. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Curless, B. (1999). From range scans to 3D models. *Computer Graphics*, 33(4):38–41.

Curless, B. and Levoy, M. (1995). Better optical triangulation through spacetime analysis. In *Fifth International Conference on Computer Vision (ICCV'95)*, pp. 987–994, Cambridge, Massachusetts.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 303–312, New Orleans.

Cutler, R. and Davis, L. S. (2000). Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796.

Cutler, R. and Turk, M. (1998). View-based interpretation of real-time optical flow for gesture recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 416–421, Nara, Japan.

Dai, S., Baker, S., and Kang, S. B. (2009). An MRF-based deinterlacing algorithm with exemplar-based refinement. *IEEE Transactions on Image Processing*, 18(5):956–968.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 886–893, San Diego, CA.

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 428–441.

Dana, K. J., van Ginneken, B., Nayar, S. K., and Koenderink, J. J. (1999). Reflectance and texture of real world surfaces. *ACM Transactions on Graphics*, 18(1):1–34.

Danielsson, P. E. (1980). Euclidean distance mapping. *Computer Graphics and Image Processing*, 14(3):227–248.

Darrell, T. and Pentland, A. (1991). Robust estimation of a multi-layered motion representation. In *IEEE Workshop on Visual Motion*, pp. 173–178, Princeton, New Jersey.

Darrell, T. and Pentland, A. (1995). Cooperative robust estimation using layers of support. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):474–487.

Darrell, T. and Simoncelli, E. (1993). "Nulling" filters and the separation of transparent motion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pp. 738–739, New York.

Darrell, T., Gordon, G., Harville, M., and Woodfill, J. (2000). Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175–185.

Darrell, T., Baker, H., Crow, F., Gordon, G., and Woodfill, J. (1997). Magic morphin mirror: face-sensitive distortion and exaggeration. In *ACM SIGGRAPH 1997 Visual Proceedings*, Los Angeles.

Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2).

Daugman, J. (2004). How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30.

David, P., DeMenthon, D., Duraiswami, R., and Samet, H. (2004). SoftPOSIT: Simultaneous pose and correspondence determination. *International Journal of Computer Vision*, 59(3):259–284.

Davies, R., Twining, C., and Taylor, C. (2008). *Statistical Models of Shape*. Springer-Verlag, London.

Davis, J. (1998). Mosaics of scenes with moving objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 354–360, Santa Barbara.

Davis, J., Ramamoorthi, R., and Rusinkiewicz, S. (2003). Spacetime stereo: A unifying framework for depth from triangulation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, pp. 359–366, Madison, WI.

Davis, J., Nahab, D., Ramamoorthi, R., and Rusinkiewicz, S. (2005). Spacetime stereo: A unifying framework for depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):296–302.

Davis, L. (1975). A survey of edge detection techniques. *Computer Graphics and Image Processing*, 4(3):248–270.

Davis, T. A. (2006). *Direct Methods for Sparse Linear Systems*. SIAM.

Davis, T. A. (2008). Multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Trans. on Mathematical Software*, (submitted).

Davison, A., Reid, I., Molton, N. D., and Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067.

de Agapito, L., Hayman, E., and Reid, I. (2001). Self-calibration of rotating and zooming cameras. *International Journal of Computer Vision*, 45(2):107–127.

de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2006). *Computational Geometry: Algorithms and Applications*. Springer, New York, NY, third edition.

De Bonet, J. (1997). Multiresolution sampling procedure for analysis and synthesis of texture images. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 361–368, Los Angeles.

De Bonet, J. S. and Viola, P. (1999). Poxels: Probabilistic voxelized volume reconstruction. In *Seventh International Conference on Computer Vision (ICCV'99)*, pp. 418–425, Kerkyra, Greece.

De Castro, E. and Morandi, C. (1987). Registration of translated and rotated images using finite Fourier transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):700–703.

de Haan, G. and Bellers, E. B. (1998). Deinterlacing—an overview. *Proceedings of the IEEE*, 86:1839–1857.

De la Torre, F. and Black, M. J. (2003). A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1/2/3):117–142.

Debevec, P. (1998). Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 1998 Conference Proceedings*, pp. 189–198.

Debevec, P. (2006). Virtual cinematography: Relighting through computation. *Computer*, 39(8):57–65.

Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., and Sagar, M. (2000). Acquiring the reflectance field of a human face. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 145–156.

Debevec, P., Wenger, A., Tchou, C., Gardner, A., Waese, J., and Hawkins, T. (2002). A lighting reproduction approach to live-action compositing. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002)*, 21(3):547–556.

Debevec, P. E. (1999). Image-based modeling and lighting. *Computer Graphics*, 33(4):46–50.

Debevec, P. E. and Malik, J. (1997). Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 369–378.

Debevec, P. E., Taylor, C. J., and Malik, J. (1996). Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 11–20, New Orleans.

Debevec, P. E., Yu, Y., and Borshukov, G. D. (1998). Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Rendering Workshop 1998*, pp. 105–116.

DeCarlo, D. and Santella, A. (2002). Stylization and abstraction of photographs. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002)*, 21(3):769–776.

DeCarlo, D., Metaxas, D., and Stone, M. (1998). An anthropometric face model using variational techniques. In *ACM SIGGRAPH 1998 Conference Proceedings*, pp. 67–74.

Delingette, H., Hebert, M., and Ikeuichi, K. (1992). Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, 10(3):132–144.

Dellaert, F. and Collins, R. (1999). Fast image-based tracking by selective pixel integration. In *ICCV Workshop on Frame-Rate Vision*, pp. 1–22.

Delong, A., Osokin, A., Isack, H. N., and Boykov, Y. (2010). Fast approximate energy minimization with label costs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA.

DeMenthon, D. I. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141.

Demmel, J., Dongarra, J., Eijkhout, V., Fuentes, E., Petitet, A. *et al.* (2005). Self-adapting linear algebra algorithms and software. *Proceedings of the IEEE*, 93(2):293–312.

Dempster, A., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Deriche, R. (1987). Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187.

Deriche, R. (1990). Fast algorithms for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):78–87.

Deutscher, J. and Reid, I. (2005). Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205.

Deutscher, J., Blake, A., and Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, pp. 126–133, Hilton Head Island.

Dev, P. (1974). *Segmentation Processes in Visual Perception: A Cooperative Neural Model*. COINS Technical Report 74C-5, University of Massachusetts at Amherst.

Dhond, U. R. and Aggarwal, J. K. (1989). Structure from stereo—a review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510.

Dick, A., Torr, P. H. S., and Cipolla, R. (2004). Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2):111–134.

Dickinson, S., Leonardis, A., Schiele, B., and Tarr, M. J. (eds). (2007). *Object Categorization: Computer and Human Vision Perspectives*, Cambridge University Press, New York.

Dickmanns, E. D. and Graefe, V. (1988). Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223–240.

Diebel, J. (2006). *Representing Attitude: Euler Angles, Quaternions, and Rotation Vectors*. Technical Report, Stanford University. http://ai.stanford.edu/~diebel/attitude.html.

Diebel, J. R., Thrun, S., and Brünig, M. (2006). A Bayesian method for probable surface reconstruction and decimation. *ACM Transactions on Graphics*, 25(1).

Dimitrijevic, M., Lepetit, V., and Fua, P. (2006). Human body pose detection using Bayesian spatio-temporal templates. *Computer Vision and Image Understanding*, 104(2-3):127–139.

Dinh, H. Q., Turk, G., and Slabaugh, G. (2002). Reconstructing surfaces by volumetric regularization using radial basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10):1358–1371.

Divvala, S., Hoiem, D., Hays, J., Efros, A. A., and Hebert, M. (2009). An empirical study of context in object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami, FL.

Dodgson, N. A. (1992). *Image Resampling*. Technical Report TR261, Wolfson College and Computer Laboratory, University of Cambridge.

Dollàr, P., Belongie, S., and Perona, P. (2010). The fastest pedestrian detector in the west. In *British Machine Vision Conference (BMVC 2010)*, Aberystwyth, Wales, UK.

Dollàr, P., Wojek, C., Schiele, B., and Perona, P. (2009). Pedestrian detection: A benchmark. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Doretto, G. and Soatto, S. (2006). Dynamic shape and appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2006–2019.

Doretto, G., Chiuso, A., Wu, Y. N., and Soatto, S. (2003). Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109.

Dorkó, G. and Schmid, C. (2003). Selection of scale-invariant parts for object class recognition. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 634–640, Nice, France.

Dorsey, J., Rushmeier, H., and Sillion, F. (2007). *Digital Modeling of Material Appearance*. Morgan Kaufmann, San Francisco.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122.

Drori, I., Cohen-Or, D., and Yeshurun, H. (2003). Fragment-based image completion. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, 22(3):303–312.

Duda, R. O. and Hart, P. E. (1972). Use of the Hough transform to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, New York, 2nd edition.

Dupuis, P. and Oliensis, J. (1994). An optimal control formulation and related numerical methods for a problem in shape reconstruction. *Annals of Applied Probability*, 4(2):287–346.

Durand, F. and Dorsey, J. (2002). Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002)*, 21(3):257–266.

Durand, F. and Szeliski, R. (2007). Computational photography. *IEEE Computer Graphics and Applications*, 27(2):21–22. Guest Editors' Introduction to Special Issue.

Durbin, R. and Willshaw, D. (1987). An analogue approach to the traveling salesman problem using an elastic net method. *Nature*, 326:689–691.

Durbin, R., Szeliski, R., and Yuille, A. (1989). An analysis of the elastic net approach to the travelling salesman problem. *Neural Computation*, 1(3):348–358.

Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. (1995). Multiresolution analysis of arbitrary meshes. In *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 173–182, Los Angeles.

Eden, A., Uyttendaele, M., and Szeliski, R. (2006). Seamless image stitching of scenes with large motions and exposure differences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 2498–2505, New York, NY.

Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH 2001 Conference Proceedings*, pp. 341–346.

Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *Seventh International Conference on Computer Vision (ICCV'99)*, pp. 1033–1038, Kerkyra, Greece.

Efros, A. A., Berg, A. C., Mori, G., and Malik, J. (2003). Recognizing action at a distance. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 726–733, Nice, France.

Eichner, M. and Ferrari, V. (2009). Better appearance models for pictorial structures. In *British Machine Vision Conference (BMVC 2009)*.

Eisemann, E. and Durand, F. (2004). Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics*, 23(3):673–678.

Eisert, P., Steinbach, E., and Girod, B. (2000). Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(2):261–277.

Eisert, P., Wiegand, T., and Girod, B. (2000). Model-aided coding: a new approach to incorporate facial animation into motion-compensated video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(3):344–358.

Ekman, P. and Friesen, W. V. (1978). *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists press, Palo Alto, CA.

El-Melegy, M. and Farag, A. (2003). Nonmetric lens distortion calibration: Closed-form solutions, robust estimation and model selection. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 554–559, Nice, France.

Elder, J. H. (1999). Are edges incomplete? *International Journal of Computer Vision*, 34(2/3):97–122.

Elder, J. H. and Goldberg, R. M. (2001). Image editing in the contour domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):291–296.

Elder, J. H. and Zucker, S. W. (1998). Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716.

Engels, C., Stewénius, H., and Nistér, D. (2006). Bundle adjustment rules. In *Photogrammetric Computer Vision (PCV'06)*, Bonn, Germany.

Engl, H. W., Hanke, M., and Neubauer, A. (1996). *Regularization of Inverse Problems*. Kluwer Academic Publishers, Dordrecht.

Enqvist, O., Josephson, K., and Kahl, F. (2009). Optimal correspondences from pairwise constraints. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Estrada, F. J. and Jepson, A. D. (2009). Benchmarking image segmentation algorithms. *International Journal of Computer Vision*, 85(2):167–181.

Estrada, F. J., Jepson, A. D., and Chennubhotla, C. (2004). Spectral embedding and min-cut for image segmentation. In *British Machine Vision Conference (BMVC 2004)*, pp. 317–326, London.

Evangelidis, G. D. and Psarakis, E. Z. (2008). Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2008). The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):147–168.

Ezzat, T., Geiger, G., and Poggio, T. (2002). Trainable videorealistic speech animation. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002)*, 21(3):388–398.

Fabbri, R., Costa, L. D. F., Torelli, J. C., and Bruno, O. M. (2008). 2D Euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys*, 40(1).

Fairchild, M. D. (2005). *Color Appearance Models*. Wiley, 2nd edition.

Fan, R.-E., Chen, P.-H., and Lin, C.-J. (2005). Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6:1889–1918.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Farbman, Z., Fattal, R., Lischinski, D., and Szeliski, R. (2008). Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics (Proc. SIGGRAPH 2008)*, 27(3).

Farenzena, M., Fusiello, A., and Gherardi, R. (2009). Structure-and-motion pipeline on a hierarchical cluster tree. In *IEEE International Workshop on 3D Digital Imaging and Modeling (3DIM 2009)*, Kyoto, Japan.

Farin, G. (1992). From conics to NURBS: A tutorial and survey. *IEEE Computer Graphics and Applications*, 12(5):78–86.

Farin, G. E. (1996). *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, Boston, Massachusetts, 4th edition.

Fattal, R. (2007). Image upsampling via imposed edge statistics. *ACM Transactions on Graphics*, 26(3).

Fattal, R. (2009). Edge-avoiding wavelets and their applications. *ACM Transactions on Graphics*, 28(3).

Fattal, R., Lischinski, D., and Werman, M. (2002). Gradient domain high dynamic range compression. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002)*, 21(3):249–256.

Faugeras, O. (1993). *Three-dimensional computer vision: A geometric viewpoint*. MIT Press, Cambridge, Massachusetts.

Faugeras, O. and Keriven, R. (1998). Variational principles, surface evolution, PDEs, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344.

Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images*. MIT Press, Cambridge, MA.

Faugeras, O. D. (1992). What can be seen in three dimensions with an uncalibrated stereo rig? In *Second European Conference on Computer Vision (ECCV'92)*, pp. 563–578, Santa Margherita Liguere, Italy.

Faugeras, O. D. and Hebert, M. (1987). The representation, recognition and positioning of 3-D shapes from range data. In Kanade, T. (ed.), *Three-Dimensional Machine Vision*, pp. 301–353, Kluwer Academic Publishers, Boston.

Faugeras, O. D., Luong, Q.-T., and Maybank, S. J. (1992). Camera self-calibration: Theory and experiments. In *Second European Conference on Computer Vision (ECCV'92)*, pp. 321–334, Santa Margherita Liguere, Italy.

Favaro, P. and Soatto, S. (2006). *3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*. Springer.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Fei-Fei, L. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 524–531, San Diego, CA.

Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611.

Fei-Fei, L., Fergus, R., and Torralba, A. (2009). ICCV 2009 short course on recognizing and learning object categories. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan. http://people.csail.mit.edu/torralba/shortCourseRLOC/.

Feilner, M., Van De Ville, D., and Unser, M. (2005). An orthogonal family of quincunx wavelets with continuously adjustable order. *IEEE Transactions on Image Processing*, 14(4):499–520.

Feldmar, J. and Ayache, N. (1996). Rigid, affine, and locally affine registration of free-form surfaces. *International Journal of Computer Vision*, 18(2):99–119.

Fellbaum, C. (ed.). (1998). *WordNet: An Electronic Lexical Database*, Bradford Books.

Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004a). *Distance Transforms of Sampled Functions*. Technical Report TR2004-1963, Cornell University Computing and Information Science.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004b). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2006). Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645.

Ferencz, A., Learned-Miller, E. G., and Malik, J. (2008). Learning to locate informative features for visual identification. *International Journal of Computer Vision*, 77(1-3):3–24.

Fergus, R. (2007a). Combined segmentation and recognition. In *CVPR 2007 Short Course on Recognizing and Learning Object Categories*. http://people.csail.mit.edu/torralba/shortCourseRLOC/.

Fergus, R. (2007b). Part-based models. In *CVPR 2007 Short Course on Recognizing and Learning Object Categories*. http://people.csail.mit.edu/torralba/shortCourseRLOC/.

Fergus, R. (2009). Classical methods for object recognition. In *ICCV 2009 Short Course on Recognizing and Learning Object Categories*, Kyoto, Japan. http://people.csail.mit.edu/torralba/shortCourseRLOC/.

Fergus, R., Perona, P., and Zisserman, A. (2004). A visual category filter for Google images. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 242–256, Prague.

Fergus, R., Perona, P., and Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 380–387, San Diego, CA.

Fergus, R., Perona, P., and Zisserman, A. (2007). Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3):273–303.

Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A. (2005). Learning object categories from Google's image search. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 1816–1823, Beijing, China.

Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T. (2006). Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 25(3):787–794.

Ferrari, V., Marin-Jimenez, M., and Zisserman, A. (2009). Pose search: retrieving people using their pose. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Ferrari, V., Marin-Jimenez, M. J., and Zisserman, A. (2008). Progressive search space reduction for human pose estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Ferrari, V., Tuytelaars, T., and Van Gool, L. (2006a). Object detection by contour segment networks. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 14–28.

Ferrari, V., Tuytelaars, T., and Van Gool, L. (2006b). Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision*, 67(2):159–188.

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394.

Finkelstein, A. and Salesin, D. H. (1994). Multiresolution curves. In *ACM SIGGRAPH 1994 Conference Proceedings*, pp. 261–268.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Fischler, M. A. and Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92.

Fischler, M. A. and Firschein, O. (1987). *Readings in Computer Vision*. Morgan Kaufmann Publishers, Inc., Los Altos.

Fischler, M. A., Firschein, O., Barnard, S. T., Fua, P. V., and Leclerc, Y. (1989). *The Vision Problem: Exploiting Parallel Computation*. Technical Note 458, SRI International, Menlo Park.

Fitzgibbon, A. W. and Zisserman, A. (1998). Automatic camera recovery for closed and open image sequences. In *Fifth European Conference on Computer Vision (ECCV'98)*, pp. 311–326, Freiburg, Germany.

Fitzgibbon, A. W., Cross, G., and Zisserman, A. (1998). Automatic 3D model construction for turn-table sequences. In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE)*, pp. 155–170, Freiburg.

Fleet, D. and Jepson, A. (1990). Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104.

Fleuret, F. and Geman, D. (2001). Coarse-to-fine face detection. *International Journal of Computer Vision*, 41(1/2):85–107.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q. *et al*. (1995). Query by image and video content: The QBIC system. *Computer*, 28(9):23–32.

Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1995). *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA, 2 edition.

Förstner, W. (1986). A feature-based correspondence algorithm for image matching. *Intl. Arch. Photogrammetry & Remote Sensing*, 26(3):150–166.

Förstner, W. (2005). Uncertainty and projective geometry. In Bayro-Corrochano, E. (ed.), *Handbook of Geometric Computing*, pp. 493–534, Springer, New York.

Forsyth, D. and Ponce, J. (2003). *Computer Vision: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.

Forsyth, D. A. and Fleck, M. M. (1999). Automatic detection of human nudes. *International Journal of Computer Vision*, 32(1):63–77.

Forsyth, D. A., Arikan, O., Ikemoto, L., O'Brien, J., and Ramanan, D. (2006). Computational studies of human motion: Part 1, tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Computer Vision*, 1(2/3):77–254.

Fossati, A., Dimitrijevic, M., Lepetit, V., and Fua, P. (2007). Bridging the gap between detection and tracking for 3D monocular video-based motion capture. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Frahm, J.-M. and Koch, R. (2003). Camera calibration with known rotation. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 1418–1425, Nice, France.

Freeman, M. (2008). *Mastering HDR Photography*. Amphoto Books, New York.

Freeman, W., Perona, P., and Schölkopf, B. (2008). Guest editorial: Special issue on machine learning for vision. *International Journal of Computer Vision*, 77(1-3):1.

Freeman, W. T. (1992). *Steerable Filters and Local Analysis of Image Structure*. Ph.D. thesis, Massachusetts Institute of Technology.

Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906.

Freeman, W. T., Jones, T. R., and Pasztor, E. C. (2002). Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65.

Freeman, W. T., Pasztor, E. C., and Carmichael, O. T. (2000). Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47.

Frey, B. J. and MacKay, D. J. C. (1997). A revolution: Belief propagation in graphs with cycles. In *Advances in Neural Information Processing Systems*.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 38(2):337–374.

Frisken, S. F., Perry, R. N., Rockwood, A. P., and Jones, T. R. (2000). Adaptively sampled distance fields: A general representation of shape for computer graphics. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 249–254.

Fritz, M. and Schiele, B. (2008). Decomposition, discovery and detection of visual categories using topic models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Frome, A., Singer, Y., Sha, F., and Malik, J. (2007). Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Fua, P. (1993). A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49.

Fua, P. and Leclerc, Y. G. (1995). Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal of Computer Vision*, 16(1):35–56.

Fua, P. and Sander, P. (1992). Segmenting unstructured 3D points into surfaces. In *Second European Conference on Computer Vision (ECCV'92)*, pp. 676–680, Santa Margherita Liguere, Italy.

Fuh, C.-S. and Maragos, P. (1991). Motion displacement estimation using an affine model for image matching. *Optical Engineering*, 30(7):881–887.

Fukunaga, K. and Hostetler, L. D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21:32–40.

Furukawa, Y. and Ponce, J. (2007). Accurate, dense, and robust multi-view stereopsis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Furukawa, Y. and Ponce, J. (2008). Accurate calibration from multi-view stereo and bundle adjustment. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Furukawa, Y. and Ponce, J. (2009). Carved visual hulls for image-based modeling. *International Journal of Computer Vision*, 81(1):53–67.

Furukawa, Y. and Ponce, J. (2011). Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2009a). Manhattan-world stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami, FL.

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2009b). Reconstructing building interiors from images. In *Twelfth IEEE International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2010). Towards internet-scale multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA.

Fusiello, A., Roberto, V., and Trucco, E. (1997). Efficient stereo with multiple windowing. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 858–863, San Juan, Puerto Rico.

Fusiello, A., Trucco, E., and Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22.

Gai, J. and Kang, S. B. (2009). Matte-based restoration of vintage video. *IEEE Transactions on Image Processing*, 18:2185–2197.

Gal, R., Wexler, Y., Ofek, E., Hoppe, H., and Cohen-Or, D. (2010). Seamless montage for texturing models. In *Proceedings of Eurographics 2010*.

Gallagher, A. C. and Chen, T. (2008). Multi-image graph cut clothing segmentation for recognizing people. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Gallup, D., Frahm, J.-M., Mordohai, P., and Pollefeys, M. (2008). Variable baseline/resolution stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Gamble, E. and Poggio, T. (1987). *Visual integration and detection of discontinuities: the key role of intensity edges*. A. I. Memo 970, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Gammeter, S., Bossard, L., Quack, T., and Van Gool, L. (2009). I know what you did last summer: Object-level auto-annotation of holiday snaps. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Gao, W., Chen, Y., Wang, R., Shan, S., and Jiang, D. (2003). Learning and synthesizing MPEG-4 compatible 3-D face animation from video sequence. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1119–1128.

Garding, J. (1992). Shape from texture for smooth curved surfaces in perspective projection. *Journal of Mathematical Imaging and Vision*, 2:329–352.

Gargallo, P., Prados, E., and Sturm, P. (2007). Minimizing the reprojection error in surface reconstruction from images. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Gavrila, D. M. (1999). The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98.

Gavrila, D. M. and Davis, L. S. (1996). 3D model-based tracking of humans in action: A multi-view approach. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pp. 73–80, San Francisco.

Gavrila, D. M. and Philomin, V. (1999). Real-time object detection for smart vehicles. In *Seventh International Conference on Computer Vision (ICCV'99)*, pp. 87–93, Kerkyra, Greece.

Geiger, D. and Girosi, F. (1991). Parallel and deterministic algorithms for MRFs: Surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):401–412.

Geiger, D., Ladendorf, B., and Yuille, A. (1992). Occlusions and binocular stereo. In *Second European Conference on Computer Vision (ECCV'92)*, pp. 425–433, Santa Margherita Liguere, Italy.

Gelb, A. (ed.). (1974). *Applied Optimal Estimation*. MIT Press, Cambridge, Massachusetts.

Geller, T. (2008). Overcoming the uncanny valley. *IEEE Computer Graphics and Applications*, 28(4):11–17.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Gennert, M. A. (1988). Brightness-based stereo matching. In *Second International Conference on Computer Vision (ICCV'88)*, pp. 139–143, Tampa.

Gersho, A. and Gray, R. M. (1991). *Vector Quantization and Signal Compression*. Springer.

Gershun, A. (1939). The light field. *Journal of Mathematics and Physics*, XVIII:51–151.

Gevers, T., van de Weijer, J., and Stokman, H. (2006). Color feature detection. In Lukac, R. and Plataniotis, K. N. (eds), *Color Image Processing: Methods and Applications*, CRC Press.

Giblin, P. and Weiss, R. (1987). Reconstruction of surfaces from profiles. In *First International Conference on Computer Vision (ICCV'87)*, pp. 136–144, London, England.

Gionis, A., Indyk, P., and Motwani, R. (1999). Similarity search in high dimensions via hashing. In *25th International Conference on Very Large Data Bases (VLDB'99)*, pp. 518–529.

Girod, B., Greiner, G., and Niemann, H. (eds). (2000). *Principles of 3D Image Analysis and Synthesis*, Kluwer, Boston.

Glassner, A. S. (1995). *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers, San Francisco.

Gleicher, M. (1995). Image snapping. In *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 183–190.

Gleicher, M. (1997). Projective registration with difference decomposition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 331–337, San Juan, Puerto Rico.

Gleicher, M. and Witkin, A. (1992). Through-the-lens camera control. *Computer Graphics (SIGGRAPH '92)*, 26(2):331–340.

Glocker, B., Komodakis, N., Tziritas, G., Navab, N., and Paragios, N. (2008). Dense image registration through MRFs and efficient linear programming. *Medical Image Analysis*, 12(6):731–741.

Glocker, B., Paragios, N., Komodakis, N., Tziritas, G., and Navab, N. (2008). Optical flow estimation with uncertainties through dynamic MRFs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Gluckman, J. (2006a). Higher order image pyramids. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 308–320.

Gluckman, J. (2006b). Scale variant image pyramids. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 1069–1075, New York City, NY.

Goesele, M., Curless, B., and Seitz, S. (2006). Multi-view stereo revisited. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 2402–2409, New York City, NY.

Goesele, M., Fuchs, C., and Seidel, H.-P. (2003). Accuracy of 3D range scanners by measurement of the slanted edge modulation transfer function. In *Fourth International Conference on 3-D Digital Imaging and Modeling*, Banff.

Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. M. (2007). Multi-view stereo for community photo collections. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Gold, S., Rangarajan, A., Lu, C., Pappu, S., and Mjolsness, E. (1998). New algorithms for 2D and 3D point matching: Pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031.

Goldberg, A. V. and Tarjan, R. E. (1988). A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4):921–940.

Goldluecke, B. and Cremers, D. (2009). Superresolution texture maps for multiview reconstruction. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Goldman, D. B. (2011). Vignette and exposure calibration and compensation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Golovinskiy, A., Matusik, W., ster, H. P., Rusinkiewicz, S., and Funkhouser, T. (2006). A statistical model for synthesis of detailed facial geometry. *ACM Transactions on Graphics*, 25(3):1025–1034.

Golub, G. and Van Loan, C. F. (1996). *Matrix Computation, third edition*. The John Hopkins University Press, Baltimore and London.

Gomes, J. and Velho, L. (1997). *Image Processing for Computer Graphics*. Springer-Verlag, New York.

Gomes, J., Darsa, L., Costa, B., and Velho, L. (1999). *Warping and Morphing of Graphical Objects*. Morgan Kaufmann Publishers, San Francisco.

Gong, M., Yang, R., Wang, L., and Gong, M. (2007). A performance study on different cost aggregation approaches used in realtime stereo matching. *International Journal of Computer Vision*, 75(2):283–296.

Gonzales, R. C. and Woods, R. E. (2008). *Digital Image Processing*. Prentice-Hall, Upper Saddle River, NJ, 3rd edition.

Gooch, B. and Gooch, A. (2001). *Non-Photorealistic Rendering*. A K Peters, Ltd, Natick, Massachusetts.

Gordon, I. and Lowe, D. G. (2006). What and where: 3D object recognition with accurate pose. In Ponce, J., Hebert, M., Schmid, C., and Zisserman, A. (eds), *Toward Category-Level Object Recognition*, pp. 67–82, Springer, New York.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253.

Gortler, S. J. and Cohen, M. F. (1995). Hierarchical and variational geometric modeling with wavelets. In *Symposium on Interactive 3D Graphics*, pp. 35–43, Monterey, CA.

Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. (1996). The Lumigraph. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 43–54, New Orleans.

Goshtasby, A. (1989). Correction of image deformation from lens distortion using Bézier patches. *Computer Vision, Graphics, and Image Processing*, 47(4):385–394.

Goshtasby, A. (2005). *2-D and 3-D Image Registration*. Wiley, New York.

Gotchev, A. and Rosenhahn, B. (eds). (2009). *Proceedings of the 3DTV Conference: The True Vision—Capture, Transmission and Display of 3D Video*, IEEE Computer Society Press.

Govindu, V. M. (2006). Revisiting the brightness constraint: Probabilistic formulation and algorithms. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 177–188.

Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783.

Grady, L. (2008). A lattice-preserving multigrid method for solving the inhomogeneous Poisson equations used in image analysis. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 252–264, Marseilles.

Grady, L. and Ali, S. (2008). Fast approximate random walker segmentation using eigenvector precomputation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Grady, L. and Alvino, C. (2008). Reformulating and optimizing the Mumford–Shah functional on a graph — a faster, lower energy solution. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 248–261, Marseilles.

Grauman, K. and Darrell, T. (2005). Efficient image matching with distributions of local invariant features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 627–634, San Diego, CA.

Grauman, K. and Darrell, T. (2007a). Pyramid match hashing: Sub-linear time indexing over partial correspondences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Grauman, K. and Darrell, T. (2007b). The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760.

Grauman, K., Shakhnarovich, G., and Darrell, T. (2003). Inferring 3D structure with a statistical image-based shape model. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 641–648, Nice, France.

Greene, N. (1986). Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29.

Greene, N. and Heckbert, P. (1986). Creating raster Omnimax images from multiple perspective views using the elliptical weighted average filter. *IEEE Computer Graphics and Applications*, 6(6):21–27.

Greig, D., Porteous, B., and Seheult, A. (1989). Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279.

Gremban, K. D., Thorpe, C. E., and Kanade, T. (1988). Geometric camera calibration using systems of linear equations. In *IEEE International Conference on Robotics and Automation*, pp. 562–567, Philadelphia.

Griffin, G., Holub, A., and Perona, P. (2007). *Caltech-256 Object Category Dataset*. Technical Report 7694, California Institute of Technology.

Grimson, W. E. L. (1983). An implementation of a computational theory of visual surface interpolation. *Computer Vision, Graphics, and Image Processing*, 22:39–69.

Grimson, W. E. L. (1985). Computational experiments with a feature based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(1):17–34.

Gross, R., Matthews, I., and Baker, S. (2006). Active appearance models with occlusion. *Image and Vision Computing*, 24(6):593–604.

Gross, R., Shi, J., and Cohn, J. F. (2005). Quo vadis face recognition? In *IEEE Workshop on Empirical Evaluation Methods in Computer Vision*, San Diego.

Gross, R., Baker, S., Matthews, I., and Kanade, T. (2005). Face recognition across pose and illumination. In Li, S. Z. and Jain, A. K. (eds), *Handbook of Face Recognition*, Springer.

Gross, R., Sweeney, L., De la Torre, F., and Baker, S. (2008). Semi-supervised learning of multi-factor models for face de-identification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. (2010). Multi-PIE. *Image and Vision Computing*, 28(5):807–813.

Grossberg, M. D. and Nayar, S. K. (2001). A general imaging model and a method for finding its parameters. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 108–115, Vancouver, Canada.

Grossberg, M. D. and Nayar, S. K. (2004). Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282.

Gu, C., Lim, J., Arbelaez, P., and Malik, J. (2009). Recognition using regions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Gu, X., Gortler, S. J., and Hoppe, H. (2002). Geometry images. *ACM Transactions on Graphics*, 21(3):355–361.

Guan, P., Weiss, A., Bălan, A. O., and Black, M. J. (2009). Estimating human shape and pose from a single image. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Guennebaud, G. and Gross, M. (2007). Algebraic point set surfaces. *ACM Transactions on Graphics*, 26(3).

Guennebaud, G., Germann, M., and Gross, M. (2008). Dynamic sampling and rendering of algebraic point set surfaces. *Computer Graphics Forum*, 27(2):653–662.

Guenter, B., Grimm, C., Wood, D., Malvar, H., and Pighin, F. (1998). Making faces. In *ACM SIGGRAPH 1998 Conference Proceedings*, pp. 55–66.

Guillaumin, M., Verbeek, J., and Schmid, C. (2009). Is that you? Metric learning approaches for face identification. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Gulbins, J. and Gulbins, R. (2009). *Photographic Multishot Techniques: High Dynamic Range, Super-Resolution, Extended Depth of Field, Stitching*. Rocky Nook.

Habbecke, M. and Kobbelt, L. (2007). A surface-growing approach to multi-view stereo reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Hager, G. D. and Belhumeur, P. N. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039.

Hall, R. (1989). *Illumination and Color in Computer Generated Imagery*. Springer-Verlag, New York.

Haller, M., Billinghurst, M., and Thomas, B. (2007). *Emerging Technologies of Augmented Reality: Interfaces and Design*. IGI Publishing.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics : The Approach Based on Influence Functions*. Wiley, New York.

Han, F. and Zhu, S.-C. (2005). Bottom-up/top-down image parsing by attribute graph grammar. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 1778–1785, Beijing, China.

Hanna, K. J. (1991). Direct multi-resolution estimation of ego-motion and structure from motion. In *IEEE Workshop on Visual Motion*, pp. 156–162, Princeton, New Jersey.

Hannah, M. J. (1974). *Computer Matching of Areas in Stereo Images*. Ph.D. thesis, Stanford University.

Hannah, M. J. (1988). Test results from SRI's stereo system. In *Image Understanding Workshop*, pp. 740–744, Cambridge, Massachusetts.

Hansen, M., Anandan, P., Dana, K., van der Wal, G., and Burt, P. (1994). Real-time scene stabilization and mosaic construction. In *IEEE Workshop on Applications of Computer Vision (WACV'94)*, pp. 54–62, Sarasota.

Hanson, A. R. and Riseman, E. M. (eds). (1978). *Computer Vision Systems*, Academic Press, New York.

Haralick, R. M. and Shapiro, L. G. (1985). Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132.

Haralick, R. M. and Shapiro, L. G. (1992). *Computer and Robot Vision*. Addison-Wesley, Reading, MA.

Haralick, R. M., Lee, C.-N., Ottenberg, K., and Nölle, M. (1994). Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356.

Hardie, R. C., Barnard, K. J., and Armstrong, E. E. (1997). Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Transactions on Image Processing*, 6(12):1621–1633.

Haritaoglu, I., Harwood, D., and Davis, L. S. (2000). W^4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830.

Harker, M. and O'Leary, P. (2008). Least squares surface reconstruction from measured gradient fields. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Harris, C. and Stephens, M. J. (1988). A combined corner and edge detector. In *Alvey Vision Conference*, pp. 147–152.

Hartley, R. and Kang, S. B. (2007). Parameter-free radial distortion correction with center of distortion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1309–1321.

Hartley, R., Gupta, R., and Chang, T. (1992). Estimation of relative camera positions for uncalibrated cameras. In *Second European Conference on Computer Vision (ECCV'92)*, pp. 579–587, Santa Margherita Liguere, Italy.

Hartley, R. I. (1994a). Projective reconstruction and invariants from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1036–1041.

Hartley, R. I. (1994b). Self-calibration from multiple views of a rotating camera. In *Third European Conference on Computer Vision (ECCV'94)*, pp. 471–478, Stockholm, Sweden.

Hartley, R. I. (1997a). In defense of the 8-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593.

Hartley, R. I. (1997b). Self-calibration of stationary cameras. *International Journal of Computer Vision*, 22(1):5–23.

Hartley, R. I. (1998). Chirality. *International Journal of Computer Vision*, 26(1):41–61.

Hartley, R. I. and Kang, S. B. (2005). Parameter-free radial distortion correction with centre of distortion estimation. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 1834–1841, Beijing, China.

Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry*. Cambridge University Press, Cambridge, UK.

Hartley, R. I., Hayman, E., de Agapito, L., and Reid, I. (2000). Camera calibration and the search for infinity. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, pp. 510–517, Hilton Head Island.

Hasinoff, S. W. and Kutulakos, K. N. (2008). Light-efficient photography. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 45–59, Marseilles.

Hasinoff, S. W., Durand, F., and Freeman, W. T. (2010). Noise-optimal capture for high dynamic range photography. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA.

Hasinoff, S. W., Kang, S. B., and Szeliski, R. (2006). Boundary matting for view synthesis. *Computer Vision and Image Understanding*, 103(1):22–32.

Hasinoff, S. W., Kutulakos, K. N., Durand, F., and Freeman, W. T. (2009). Time-constrained photography. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York.

Hayes, B. (2008). Computational photography. *American Scientist*, 96:94–99.

Hays, J. and Efros, A. A. (2007). Scene completion using millions of photographs. *ACM Transactions on Graphics*, 26(3).

Hays, J., Leordeanu, M., Efros, A. A., and Liu, Y. (2006). Discovering texture regularity as a higher-order correspondence problem. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 522–535.

He, L.-W. and Zhang, Z. (2005). Real-time whiteboard capture and processing using a video camera for teleconferencing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, pp. 1113–1116, Philadelphia.

He, X. and Zemel, R. S. (2008). Learning hybrid models for image annotation with partially labeled data. In *Advances in Neural Information Processing Systems*.

He, X., Zemel, R. S., and Carreira-Perpiñán, M. A. (2004). Multiscale conditional random fields for image labeling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 695–702, Washington, DC.

He, X., Zemel, R. S., and Ray, D. (2006). Learning and incorporating top-down cues in image segmentation. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 338–351.

Healey, G. E. and Kondepudy, R. (1994). Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276.

Healey, G. E. and Shafer, S. A. (1992). *Color. Physics-Based Vision: Principles and Practice*, Jones & Bartlett, Cambridge, MA.

Heath, M. D., Sarkar, S., Sanocki, T., and Bowyer, K. W. (1998). Comparison of edge detectors. *Computer Vision and Image Understanding*, 69(1):38–54.

Hebert, M. (2000). Active and passive range sensing for robotics. In *IEEE International Conference on Robotics and Automation*, pp. 102–110, San Francisco.

Hecht, E. (2001). *Optics*. Pearson Addison Wesley, Reading, MA, 4th edition.

Heckbert, P. (1986). Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67.

Heckbert, P. (1989). *Fundamentals of Texture Mapping and Image Warping*. Master's thesis, The University of California at Berkeley.

Heeger, D. J. (1988). Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1(1):279–302.

Heeger, D. J. and Bergen, J. R. (1995). Pyramid-based texture analysis/synthesis. In *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 229–238.

Heisele, B., Serre, T., and Poggio, T. (2007). A component-based framework for face detection and identification. *International Journal of Computer Vision*, 74(2):167–181.

Heisele, B., Ho, P., Wu, J., and Poggio, T. (2003). Face recognition: component-based versus global approaches. *Computer Vision and Image Understanding*, 91(1-2):6–21.

Herley, C. (2005). Automatic occlusion removal from minimum number of images. In *International Conference on Image Processing (ICIP 2005)*, pp. 1046–1049–16, Genova.

Hernandez, C. and Schmitt, F. (2004). Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, 96(3):367–392.

Hernandez, C. and Vogiatzis, G. (2010). Self-calibrating a real-time monocular 3d facial capture system. In *Fifth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'10)*, Paris.

Hernandez, C., Vogiatzis, G., and Cipolla, R. (2007). Probabilistic visibility for multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Hernandez, C., Vogiatzis, G., Brostow, G. J., Stenger, B., and Cipolla, R. (2007). Non-rigid photometric stereo with colored lights. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Hershberger, J. and Snoeyink, J. (1992). *Speeding Up the Douglas-Peucker Line-Simplification Algorithm*. Technical Report TR-92-07, Computer Science Department, The University of British Columbia.

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. H. (2001). Image analogies. In *ACM SIGGRAPH 2001 Conference Proceedings*, pp. 327–340.

Hiep, V. H., Keriven, R., Pons, J.-P., and Labatut, P. (2009). Towards high-resolution large-scale multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Hillman, P., Hannah, J., and Renshaw, D. (2001). Alpha channel estimation in high resolution images and image sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, pp. 1063–1068, Kauai, Hawaii.

Hilton, A., Fua, P., and Ronfard, R. (2006). Modeling people: Vision-based understanding of a person's shape, appearance, movement, and behaviour. *Computer Vision and Image Understanding*, 104(2-3):87–89.

Hilton, A., Stoddart, A. J., Illingworth, J., and Windeatt, T. (1996). Reliable surface reconstruction from multiple range images. In *Fourth European Conference on Computer Vision (ECCV'96)*, pp. 117–126, Cambridge, England.

Hinckley, K., Sinclair, M., Hanson, E., Szeliski, R., and Conway, M. (1999). The VideoMouse: a camera-based multi-degree-of-freedom input device. In *12th annual ACM symposium on User interface software and technology*, pp. 103–112.

Hinterstoisser, S., Benhimane, S., Navab, N., Fua, P., and Lepetit, V. (2008). Online learning of patch perspective rectification for efficient object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Hinton, G. E. (1977). *Relaxation and its Role in Vision*. Ph.D. thesis, University of Edinburgh.

Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341.

Hirschmüller, H. and Scharstein, D. (2009). Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1582–1599.

Hjaltason, G. R. and Samet, H. (2003). Index-driven similarity search in metric spaces. *ACM Transactions on Database Systems*, 28(4):517–580.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 50–57, Berkeley, CA.

Hogg, D. (1983). Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20.

Hoiem, D., Efros, A. A., and Hebert, M. (2005a). Automatic photo pop-up. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):577–584.

Hoiem, D., Efros, A. A., and Hebert, M. (2005b). Geometric context from a single image. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 654–661, Beijing, China.

Hoiem, D., Efros, A. A., and Hebert, M. (2008a). Closing the loop in scene interpretation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Hoiem, D., Efros, A. A., and Hebert, M. (2008b). Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15.

Hoiem, D., Rother, C., and Winn, J. (2007). 3D LayoutCRF for multi-view object class recognition and segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H. *et al*. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689.

Hoppe, H. (1996). Progressive meshes. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 99–108, New Orleans.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1992). Surface reconstruction from unorganized points. *Computer Graphics (SIGGRAPH '92)*, 26(2):71–78.

Horn, B. K. P. (1974). Determining lightness from an image. *Computer Graphics and Image Processing*, 3(1):277–299.

Horn, B. K. P. (1975). Obtaining shape from shading information. In Winston, P. H. (ed.), *The Psychology of Computer Vision*, pp. 115–155, McGraw-Hill, New York.

Horn, B. K. P. (1977). Understanding image intensities. *Artificial Intelligence*, 8(2):201–231.

Horn, B. K. P. (1986). *Robot Vision*. MIT Press, Cambridge, Massachusetts.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.

Horn, B. K. P. (1990). Height and gradient from shading. *International Journal of Computer Vision*, 5(1):37–75.

Horn, B. K. P. and Brooks, M. J. (1986). The variational approach to shape from shading. *Computer Vision, Graphics, and Image Processing*, 33:174–208.

Horn, B. K. P. and Brooks, M. J. (eds). (1989). *Shape from Shading*, MIT Press, Cambridge, Massachusetts.

Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

Horn, B. K. P. and Weldon Jr., E. J. (1988). Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76.

Hornung, A., Zeng, B., and Kobbelt, L. (2008). Image selection for improved multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Horowitz, S. L. and Pavlidis, T. (1976). Picture segmentation by a tree traversal algorithm. *Journal of the ACM*, 23(2):368–388.

Horry, Y., Anjyo, K.-I., and Arai, K. (1997). Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 225–232.

Hough, P. V. C. (1962). Method and means for recognizing complex patterns. *U. S. Patent*, 3,069,654.

Houhou, N., Thiran, J.-P., and Bresson, X. (2008). Fast texture segmentation using the shape operator and active contour. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Howe, N. R., Leventon, M. E., and Freeman, W. T. (2000). Bayesian reconstruction of 3D human motion from single-camera video. In *Advances in Neural Information Processing Systems*.

Hsieh, Y. C., McKeown, D., and Perlant, F. P. (1992). Performance evaluation of scene registration and stereo matching for cartographic feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):214–238.

Hu, W., Tan, T., Wang, L., and Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334–352.

Hua, G., Brown, M., and Winder, S. (2007). Discriminant embedding for local image descriptors. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Technical Report 07-49, University of Massachusetts, Amherst.

Huang, T. S. (1981). *Image Sequence Analysis*. Springer-Verlag, Berlin, Heidelberg.

Huber, P. J. (1981). *Robust Statistics*. John Wiley & Sons, New York.

Huffman, D. A. (1971). Impossible objects and nonsense sentences. *Machine Intelligence*, 8:295–323.

Huguet, F. and Devernay, F. (2007). A variational method for scene flow estimation from stereo sequences. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Huttenlocher, D. P., Klanderman, G., and Rucklidge, W. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863.

Huynh, D. Q., Hartley, R., and Heyden, A. (2003). Outlier correcton in image sequences for the affine camera. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 585–590, Nice, France.

Iddan, G. J. and Yahav, G. (2001). 3D imaging in the studio (and elsewhere...). In *Three-Dimensional Image Capture and Applications IV*, pp. 48–55.

Igarashi, T., Nishino, K., and Nayar, S. (2007). The appearance of human skin: A survey. *Foundations and Trends in Computer Graphics and Computer Vision*, 3(1):1–95.

Ikeuchi, K. (1981). Shape from regular patterns. *Artificial Intelligence*, 22(1):49–75.

Ikeuchi, K. and Horn, B. K. P. (1981). Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184.

Ikeuchi, K. and Miyazaki, D. (eds). (2007). *Digitally Archiving Cultural Objects*, Springer, Boston, MA.

Ikeuchi, K. and Sato, Y. (eds). (2001). *Modeling From Reality*, Kluwer Academic Publishers, Boston.

Illingworth, J. and Kittler, J. (1988). A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116.

Intille, S. S. and Bobick, A. F. (1994). Disparity-space images and large occlusion stereo. In *Third European Conference on Computer Vision (ECCV'94)*, Stockholm, Sweden.

Irani, M. and Anandan, P. (1998). Video indexing based on mosaic representations. *Proceedings of the IEEE*, 86(5):905–921.

Irani, M. and Peleg, S. (1991). Improving resolution by image registration. *Graphical Models and Image Processing*, 53(3):231–239.

Irani, M., Hsu, S., and Anandan, P. (1995). Video compression using mosaic representations. *Signal Processing: Image Communication*, 7:529–552.

Irani, M., Rousso, B., and Peleg, S. (1994). Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16.

Irani, M., Rousso, B., and Peleg, S. (1997). Recovery of ego-motion using image stabilization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):268–272.

Isaksen, A., McMillan, L., and Gortler, S. J. (2000). Dynamically reparameterized light fields. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 297–306.

Isard, M. and Blake, A. (1998). CONDENSATION—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28.

Ishiguro, H., Yamamoto, M., and Tsuji, S. (1992). Omni-directional stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):257–262.

Ishikawa, H. (2003). Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336.

Ishikawa, H. and Veksler, O. (2010). Convex and truncated convex priors for multi-label MRFs. In Blake, A., Kohli, P., and Rother, C. (eds), *Advances in Markov Random Fields*, MIT Press.

Isidoro, J. and Sclaroff, S. (2003). Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 1335–1342, Nice, France.

Ivanchenko, V., Shen, H., and Coughlan, J. (2009). Elevation-based stereo implemented in real-time on a GPU. In *IEEE Workshop on Applications of Computer Vision (WACV 2009)*, Snowbird, Utah.

Jacobs, C. E., Finkelstein, A., and Salesin, D. H. (1995). Fast multiresolution image querying. In *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 277–286.

Jähne, B. (1997). *Digital Image Processing*. Springer-Verlag, Berlin.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey.

Jain, A. K., Bolle, R. M., and Pankanti, S. (eds). (1999). *Biometrics: Personal Identification in Networked Society*, Kluwer.

Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.

Jain, A. K., Topchy, A., Law, M. H. C., and Buhmann, J. M. (2004). Landscape of clustering algorithms. In *International Conference on Pattern Recognition (ICPR 2004)*, pp. 260–263.

Jenkin, M. R. M., Jepson, A. D., and Tsotsos, J. K. (1991). Techniques for disparity measurement. *CVGIP: Image Understanding*, 53(1):14–30.

Jensen, H. W., Marschner, S. R., Levoy, M., and Hanrahan, P. (2001). A practical model for subsurface light transport. In *ACM SIGGRAPH 2001 Conference Proceedings*, pp. 511–518.

Jeong, Y., Nistér, D., Steedly, D., Szeliski, R., and Kweon, I.-S. (2010). Pushing the envelope of modern methods for bundle adjustment. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA.

Jia, J. and Tang, C.-K. (2003). Image registration with global and local luminance alignment. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 156–163, Nice, France.

Jia, J., Sun, J., Tang, C.-K., and Shum, H.-Y. (2006). Drag-and-drop pasting. *ACM Transactions on Graphics*, 25(3):631–636.

Jiang, Z., Wong, T.-T., and Bao, H. (2003). Practical super-resolution from dynamic video sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, pp. 549–554, Madison, WI.

Johnson, A. E. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–448.

Johnson, A. E. and Kang, S. B. (1997). Registration and integration of textured 3-D data. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pp. 234–241, Ottawa.

Jojic, N. and Frey, B. J. (2001). Learning flexible sprites in video layers. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, pp. 199–206, Kauai, Hawaii.

Jones, D. G. and Malik, J. (1992). A computational framework for determining stereo correspondence from a set of linear spatial filters. In *Second European Conference on Computer Vision (ECCV'92)*, pp. 397–410, Santa Margherita Liguere, Italy.

Jones, M. J. and Rehg, J. M. (2001). Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96.

Joshi, N., Matusik, W., and Avidan, S. (2006). Natural video matting using camera arrays. *ACM Transactions on Graphics*, 25(3):779–786.

Joshi, N., Szeliski, R., and Kriegman, D. J. (2008). PSF estimation using sharp edge prediction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Joshi, N., Zitnick, C. L., Szeliski, R., and Kriegman, D. J. (2009). Image deblurring and denoising using color priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami, FL.

Ju, S. X., Black, M. J., and Jepson, A. D. (1996). Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pp. 307–314, San Francisco.

Ju, S. X., Black, M. J., and Yacoob, Y. (1996). Cardboard people: a parameterized model of articulated image motion. In *2nd International Conference on Automatic Face and Gesture Recognition*, pp. 38–44, Killington, VT.

Juan, O. and Boykov, Y. (2006). Active graph cuts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 1023–1029, New York City, NY.

Jurie, F. and Dhome, M. (2002). Hyperplane approximation for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):996–1000.

Jurie, F. and Schmid, C. (2004). Scale-invariant shape features for recognition of object categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 90–96, Washington, DC.

Kadir, T., Zisserman, A., and Brady, M. (2004). An affine invariant salient region detector. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 228–241, Prague.

Kaftory, R., Schechner, Y., and Zeevi, Y. (2007). Variational distance-dependent image restoration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Kahl, F. and Hartley, R. (2008). Multiple-view geometry under the l_∞-norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1603–1617.

Kakadiaris, I. and Metaxas, D. (2000). Model-based estimation of 3D human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1453–1459.

Kakumanu, P., Makrogiannis, S., and Bourbakis, N. (2007). A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122.

Kamvar, S. D., Klein, D., and Manning, C. D. (2002). Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. In *International Conference on Machine Learning*, pp. 283–290.

Kanade, T. (1977). *Computer Recognition of Human Faces*. Birkhauser, Basel.

Kanade, T. (1980). A theory of the origami world. *Artificial Intelligence*, 13:279–311.

Kanade, T. (ed.). (1987). *Three-Dimensional Machine Vision*, Kluwer Academic Publishers, Boston.

Kanade, T. (1994). Development of a video-rate stereo machine. In *Image Understanding Workshop*, pp. 549–557, Monterey.

Kanade, T. and Okutomi, M. (1994). A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932.

Kanade, T., Rander, P. W., and Narayanan, P. J. (1997). Virtualized reality: constructing virtual worlds from real scenes. *IEEE MultiMedia Magazine*, 4(1):34–47.

Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M. (1996). A stereo machine for video-rate dense depth mapping and its new applications. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pp. 196–202, San Francisco.

Kanatani, K. and Morris, D. D. (2001). Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy. *IEEE Transactions on Information Theory*, 47(5):2017–2028.

Kang, S. B. (1998). *Depth Painting for Image-based Rendering Applications*. Technical Report, Compaq Computer Corporation, Cambridge Research Lab.

Kang, S. B. (1999). A survey of image-based rendering techniques. In *Videometrics VI*, pp. 2–16, San Jose.

Kang, S. B. (2001). Radial distortion snakes. *IEICE Trans. Inf. & Syst.*, E84-D(12):1603–1611.

Kang, S. B. and Jones, M. (2002). Appearance-based structure from motion using linear classes of 3-D models. *International Journal of Computer Vision*, 49(1):5–22.

Kang, S. B. and Szeliski, R. (1997). 3-D scene data recovery using omnidirectional multibaseline stereo. *International Journal of Computer Vision*, 25(2):167–183.

Kang, S. B. and Szeliski, R. (2004). Extracting view-dependent depth maps from a collection of images. *International Journal of Computer Vision*, 58(2):139–163.

Kang, S. B. and Weiss, R. (1997). Characterization of errors in compositing panoramic images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 103–109, San Juan, Puerto Rico.

Kang, S. B. and Weiss, R. (1999). Characterization of errors in compositing panoramic images. *Computer Vision and Image Understanding*, 73(2):269–280.

Kang, S. B. and Weiss, R. (2000). Can we calibrate a camera using an image of a flat, textureless Lambertian surface? In *Sixth European Conference on Computer Vision (ECCV 2000)*, pp. 640–653, Dublin, Ireland.

Kang, S. B., Szeliski, R., and Anandan, P. (2000). The geometry-image representation tradeoff for rendering. In *International Conference on Image Processing (ICIP-2000)*, pp. 13–16, Vancouver.

Kang, S. B., Szeliski, R., and Chai, J. (2001). Handling occlusions in dense multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, pp. 103–110, Kauai, Hawaii.

Kang, S. B., Szeliski, R., and Shum, H.-Y. (1997). A parallel feature tracker for extended image sequences. *Computer Vision and Image Understanding*, 67(3):296–310.

Kang, S. B., Szeliski, R., and Uyttendaele, M. (2004). *Seamless Stitching using Multi-Perspective Plane Sweep*. Technical Report MSR-TR-2004-48, Microsoft Research.

Kang, S. B., Li, Y., Tong, X., and Shum, H.-Y. (2006). Image-based rendering. *Foundations and Trends in Computer Graphics and Computer Vision*, 2(3):173–258.

Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R. (2003). High dynamic range video. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, 22(3):319–325.

Kang, S. B., Webb, J., Zitnick, L., and Kanade, T. (1995). A multibaseline stereo system with active illumination and real-time image acquisition. In *Fifth International Conference on Computer Vision (ICCV'95)*, pp. 88–93, Cambridge, Massachusetts.

Kannala, J., Rahtu, E., Brandt, S. S., and Heikkila, J. (2008). Object recognition and segmentation by non-rigid quasi-dense matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Kass, M. (1988). Linear image features in stereopsis. *International Journal of Computer Vision*, 1(4):357–368.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., and Tachibana, K. (2000). Virtual object manipulation on a table-top AR environment. In *International Symposium on Augmented Reality (ISAR 2000)*.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Hoboken.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing*, pp. 61–70.

Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: a more distinctive representation for local image descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 506–513, Washington, DC.

Kehl, R. and Van Gool, L. (2006). Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*, 104(2-3):190–209.

Kenney, C., Zuliani, M., and Manjunath, B. (2005). An axiomatic approach to corner detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 191–197, San Diego, CA.

Keren, D., Peleg, S., and Brada, R. (1988). Image sequence enhancement using sub-pixel displacements. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'88)*, pp. 742–746, Ann Arbor, Michigan.

Kim, J., Kolmogorov, V., and Zabih, R. (2003). Visual correspondence using energy minimization and mutual information. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 1033–1040, Nice, France.

Kimmel, R. (1999). Demosaicing: image reconstruction from color CCD samples. *IEEE Transactions on Image Processing*, 8(9):1221–1228.

Kimura, S., Shinbo, T., Yamaguchi, H., Kawamura, E., and Nakano, K. (1999). A convolver-based real-time stereo machine (SAZAN). In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 457–463, Fort Collins.

Kindermann, R. and Snell, J. L. (1980). *Markov Random Fields and Their Applications*. American Mathematical Society.

King, D. (1997). *The Commissar Vanishes*. Henry Holt and Company.

Kirby, M. and Sirovich, L. (1990). Application of the Karhunen–Lòeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108.

Kirkpatrick, S., Gelatt, C. D. J., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Kirovski, D., Jojic, N., and Jancke, G. (2004). Tamper-resistant biometric IDs. In *ISSE 2004 - Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe 2004 Conference*, pp. 160–175.

Kittler, J. and Föglein, J. (1984). Contextual classification of multispectral pixel data. *Image and Vision Computing*, 2(1):13–29.

Klaus, A., Sormann, M., and Karner, K. (2006). Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *International Conference on Pattern Recognition (ICPR 2006)*, pp. 15–18.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality (ISMAR 2007)*, Nara.

Klein, G. and Murray, D. (2008). Improving the agility of keyframe-based slam. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 802–815, Marseilles.

Klein, S., Staring, M., and Pluim, J. P. W. (2007). Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-splines. *IEEE Transactions on Image Processing*, 16(12):2879–2890.

Klinker, G. J. (1993). *A Physical Approach to Color Image Understanding*. A K Peters, Wellesley, Massachusetts.

Klinker, G. J., Shafer, S. A., and Kanade, T. (1990). A physical approach to color image understanding. *International Journal of Computer Vision*, 4(1):7–38.

Koch, R., Pollefeys, M., and Van Gool, L. J. (2000). Realistic surface reconstruction of 3D scenes from uncalibrated image sequences. *Journal Visualization and Computer Animation*, 11:115–127.

Koenderink, J. J. (1990). *Solid Shape*. MIT Press, Cambridge, Massachusetts.

Koethe, U. (2003). Integrated edge and junction detection with the boundary tensor. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 424–431, Nice, France.

Kohli, P. (2007). *Minimizing Dynamic and Higher Order Energy Functions using Graph Cuts*. Ph.D. thesis, Oxford Brookes University.

Kohli, P. and Torr, P. H. S. (2005). Effciently solving dynamic Markov random fields using graph cuts. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 922–929, Beijing, China.

Kohli, P. and Torr, P. H. S. (2007). Dynamic graph cuts for efficient inference in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2079–2088.

Kohli, P. and Torr, P. H. S. (2008). Measuring uncertainty in graph cut solutions. *Computer Vision and Image Understanding*, 112(1):30–38.

Kohli, P., Kumar, M. P., and Torr, P. H. S. (2009). $\mathcal{P}^3$ & beyond: Move making algorithms for solving higher order functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1645–1656.

Kohli, P., Ladický, L., and Torr, P. H. S. (2009). Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324.

Kokaram, A. (2004). On missing data treatment for degraded video and film archives: a survey and a new Bayesian approach. *IEEE Transactions on Image Processing*, 13(3):397–415.

Kolev, K. and Cremers, D. (2008). Integration of multiview stereo and silhouettes via convex functionals on convex domains. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 752–765, Marseilles.

Kolev, K. and Cremers, D. (2009). Continuous ratio optimization via convex relaxation with applications to multiview 3D reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Kolev, K., Klodt, M., Brox, T., and Cremers, D. (2009). Continuous global optimization in multiview 3D reconstruction. *International Journal of Computer Vision*, 84(1):80–96.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, Massachusetts.

Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583.

Kolmogorov, V. and Boykov, Y. (2005). What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 564–571, Beijing, China.

Kolmogorov, V. and Zabih, R. (2002). Multi-camera scene reconstruction via graph cuts. In *Seventh European Conference on Computer Vision (ECCV 2002)*, pp. 82–96, Copenhagen.

Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159.

Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., and Rother, C. (2006). Probabilistic fusion of stereo with color and contrast for bi-layer segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1480–1492.

Komodakis, N. and Paragios, N. (2008). Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 806–820, Marseilles.

Komodakis, N. and Tziritas, G. (2007a). Approximate labeling via graph cuts based on linear programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1436–1453.

Komodakis, N. and Tziritas, G. (2007b). Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing*, 29(11):2649–2661.

Komodakis, N., Paragios, N., and Tziritas, G. (2007). MRF optimization via dual decomposition: Message-passing revisited. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Komodakis, N., Tziritas, G., and Paragios, N. (2007). Fast, approximately optimal solutions for single and dynamic MRFs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Komodakis, N., Tziritas, G., and Paragios, N. (2008). Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal dual strategies. *Computer Vision and Image Understanding*, 112(1):14–29.

Konolige, K. (1997). Small vision systems: Hardware and implementation. In *Eighth International Symposium on Robotics Research*, pp. 203–212, Hayama, Japan.

Kopf, J., Cohen, M. F., Lischinski, D., and Uyttendaele, M. (2007). Joint bilateral upsampling. *ACM Transactions on Graphics*, 26(3).

Kopf, J., Uyttendaele, M., Deussen, O., and Cohen, M. F. (2007). Capturing and viewing gigapixel images. *ACM Transactions on Graphics*, 26(3).

Kopf, J., Lischinski, D., Deussen, O., Cohen-Or, D., and Cohen, M. (2009). Locally adapted projections to reduce panorama distortions. *Computer Graphics Forum (Proceedings of EGSR 2009)*, 28(4).

Koutis, I. (2007). *Combinatorial and algebraic tools for optimal multilevel algorithms*. Ph.D. thesis, Carnegie Mellon University. Technical Report CMU-CS-07-131.

Koutis, I. and Miller, G. L. (2008). Graph partitioning into isolated, high conductance clusters: theory, computation and applications to preconditioning. In *Symposium on Parallel Algorithms and Architectures*, pp. 137–145, Munich.

Koutis, I., Miller, G. L., and Tolliver, D. (2009). Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. In *5th International Symposium on Visual Computing (ISVC09)*, Las Vegas.

Kovar, L., Gleicher, M., and Pighin, F. (2002). Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482.

Košecká, J. and Zhang, W. (2005). Extraction, matching and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding*, 100(3):174–293.

Kraus, K. (1997). *Photogrammetry*. Dümmler, Bonn.

Krishnan, D. and Fergus, R. (2009). Fast image deconvolution using hyper-Laplacian priors. In *Advances in Neural Information Processing Systems*.

Kuglin, C. D. and Hines, D. C. (1975). The phase correlation image alignment method. In *IEEE 1975 Conference on Cybernetics and Society*, pp. 163–165, New York.

Kulis, B. and Grauman, K. (2009). Kernelized locality-sensitive hashing for scalable image search. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Kumar, M. P. (2008). *Combinatorial and Convex Optimization for Probabilistic Models in Computer Vision*. Ph.D. thesis, Oxford Brookes University.

Kumar, M. P. and Torr, P. H. S. (2006). Fast memory-efficient generalized belief propagation. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 451–463.

Kumar, M. P. and Torr, P. H. S. (2008). Improved moves for truncated convex models. In *Advances in Neural Information Processing Systems*.

Kumar, M. P., Torr, P. H. S., and Zisserman, A. (2008). Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76(3):301–319.

Kumar, M. P., Torr, P. H. S., and Zisserman, A. (2010). OBJCUT: Efficient segmentation using top-down and bottom-up cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3).

Kumar, M. P., Veksler, O., and Torr, P. H. S. (2010). Improved moves for truncated convex models. *Journal of Machine Learning Research*, (sumbitted).

Kumar, M. P., Zisserman, A., and H.S.Torr, P. (2009). Efficient discriminative learning of parts-based models. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Kumar, R., Anandan, P., and Hanna, K. (1994). Direct recovery of shape from multiple views: A parallax based approach. In *Twelfth International Conference on Pattern Recognition (ICPR'94)*, pp. 685–688, Jerusalem, Israel.

Kumar, R., Anandan, P., Irani, M., Bergen, J., and Hanna, K. (1995). Representation of scenes from collections of images. In *IEEE Workshop on Representations of Visual Scenes*, pp. 10–17, Cambridge, Massachusetts.

Kumar, S. and Hebert, M. (2003). Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 1150–1157, Nice, France.

Kumar, S. and Hebert, M. (2006). Discriminative random fields. *International Journal of Computer Vision*, 68(2):179–202.

Kundur, D. and Hatzinakos, D. (1996). Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64.

Kutulakos, K. N. (2000). Approximate N-view stereo. In *Sixth European Conference on Computer Vision (ECCV 2000)*, pp. 67–83, Dublin, Ireland.

Kutulakos, K. N. and Seitz, S. M. (2000). A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218.

Kwatra, V., Essa, I., Bobick, A., and Kwatra, N. (2005). Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(5):795–802.

Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A. (2003). Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, 22(3):277–286.

Kybic, J. and Unser, M. (2003). Fast parametric elastic image registration. *IEEE Transactions on Image Processing*, 12(11):1427–1442.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

Lafortune, E. P. F., Foo, S.-C., Torrance, K. E., and Greenberg, D. P. (1997). Non-linear approximation of reflectance functions. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 117–126, Los Angeles.

Lai, S.-H. and Vemuri, B. C. (1997). Physically based adaptive preconditioning for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):594–607.

Lalonde, J.-F., Hoiem, D., Efros, A. A., Rother, C., Winn, J., and Criminisi, A. (2007). Photo clip art. *ACM Transactions on Graphics*, 26(3).

Lampert, C. H. (2008). Kernel methods in computer vision. *Foundations and Trends in Computer Graphics and Computer Vision*, 4(3):193–285.

Langer, M. S. and Zucker, S. W. (1994). Shape from shading on a cloudy day. *Journal Optical Society America, A*, 11(2):467–478.

Lanitis, A., Taylor, C. J., and Cootes, T. F. (1997). Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):742–756.

Larlus, D. and Jurie, F. (2008). Combining appearance models and Markov random fields for category level object segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Larson, G. W. (1998). LogLuv encoding for full-gamut, high-dynamic range images. *Journal of Graphics Tools*, 3(1):15–31.

Larson, G. W., Rushmeier, H., and Piatko, C. (1997). A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306.

Laurentini, A. (1994). The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162.

Lavallée, S. and Szeliski, R. (1995). Recovering the position and orientation of free-form objects from image contours using 3-D distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):378–390.

Laveau, S. and Faugeras, O. D. (1994). 3-D scene representation as a collection of images. In *Twelfth International Conference on Pattern Recognition (ICPR'94)*, pp. 689–691, Jerusalem, Israel.

Lazebnik, S., Schmid, C., and Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 2169–2176, New York City, NY.

Le Gall, D. (1991). MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58.

Leclerc, Y. G. (1989). Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3(1):73–102.

LeCun, Y., Huang, F. J., and Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 97–104, Washington, DC.

Lee, J., Chai, J., Reitsma, P. S. A., Hodgins, J. K., and Pollard, N. S. (2002). Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics*, 21(3):491–500.

Lee, M.-C., ge Chen, W., lung Bruce Lin, C., Gu, C., Markoc, T., Zabinsky, S. I., and Szeliski, R. (1997). A layered video object coding system using sprite and affine motion model. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):130–145.

Lee, M. E. and Redner, R. A. (1990). A note on the use of nonlinear filtering in computer graphics. *IEEE Computer Graphics and Applications*, 10(3):23–29.

Lee, M. W. and Cohen, I. (2006). A model-based approach for estimating human 3D poses in static images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):905–916.

Lee, S., Wolberg, G., and Shin, S. Y. (1996). Data interpolation using multilevel b-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244.

Lee, S., Wolberg, G., Chwa, K.-Y., and Shin, S. Y. (1996). Image metamorphosis with scattered feature constraints. *IEEE Transactions on Visualization and Computer Graphics*, 2(4):337–354.

Lee, Y. D., Terzopoulos, D., and Waters, K. (1995). Realistic facial modeling for animation. In *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 55–62.

Lei, C. and Yang, Y.-H. (2009). Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Leibe, B. and Schiele, B. (2003). Analyzing appearance and contour based methods for object categorization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, pp. 409–415, Madison, WI.

Leibe, B., Leonardis, A., and Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289.

Leibe, B., Seemann, E., and Schiele, B. (2005). Pedestrian detection in crowded scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 878–885, San Diego, CA.

Leibe, B., Cornelis, N., Cornelis, K., and Van Gool, L. (2007). Dynamic 3D scene analysis from a moving vehicle. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Leibowitz, D. (2001). *Camera Calibration and Reconstruction of Geometry from Images*. Ph.D. thesis, University of Oxford.

Lempitsky, V. and Boykov, Y. (2007). Global optimization for shape fitting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Lempitsky, V. and Ivanov, D. (2007). Seamless mosaicing of image-based texture maps. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Lempitsky, V., Blake, A., and Rother, C. (2008). Image segmentation by branch-and-mincut. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 15–29, Marseilles.

Lempitsky, V., Roth, S., and Rother., C. (2008). FlowFusion: Discrete-continuous optimization for optical flow estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Lempitsky, V., Rother, C., and Blake, A. (2007). Logcut - efficient graph cut optimization for Markov random fields. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Lengyel, J. and Snyder, J. (1997). Rendering with coherent layers. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 233–242, Los Angeles.

Lensch, H. P. A., Kautz, J., Goesele, M., Heidrich, W., and Seidel, H.-P. (2003). Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics*, 22(2):234–257.

Leonardis, A. and Bischof, H. (2000). Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78(1):99–118.

Leonardis, A., Jaklič, A., and Solina, F. (1997). Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1289–1295.

Lepetit, V. and Fua, P. (2005). Monocular model-based 3D tracking of rigid objects. *Foundations and Trends in Computer Graphics and Computer Vision*, 1(1).

Lepetit, V., Pilet, J., and Fua, P. (2004). Point matching as a classification problem for fast and robust object pose estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 244–250, Washington, DC.

Lepetit, V., Pilet, J., and Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479.

Leung, T. K., Burl, M. C., and Perona, P. (1995). Finding faces in cluttered scenes using random labeled graph matching. In *Fifth International Conference on Computer Vision (ICCV'95)*, pp. 637–644, Cambridge, Massachusetts.

Levenberg, K. (1944). A method for the solution of certain problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168.

Levin, A. (2006). Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems*.

Levin, A. and Szeliski, R. (2004). Visual odometry and map correlation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 611–618, Washington, DC.

Levin, A. and Szeliski, R. (2006). *Motion Uncertainty and Field of View*. Technical Report MSR-TR-2006-37, Microsoft Research.

Levin, A. and Weiss, Y. (2006). Learning to combine bottom-up and top-down segmentation. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 581–594.

Levin, A. and Weiss, Y. (2007). User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1647–1654.

Levin, A., Acha, A. R., and Lischinski, D. (2008). Spectral matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1699–1712.

Levin, A., Lischinski, D., and Weiss, Y. (2004). Colorization using optimization. *ACM Transactions on Graphics*, 23(3):689–694.

Levin, A., Lischinski, D., and Weiss, Y. (2008). A closed form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242.

Levin, A., Zomet, A., and Weiss, Y. (2004). Separating reflections from a single image using local features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 306–313, Washington, DC.

Levin, A., Fergus, R., Durand, F., and Freeman, W. T. (2007). Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics*, 26(3).

Levin, A., Weiss, Y., Durand, F., and Freeman, B. (2009). Understanding and evaluating blind deconvolution algorithms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Levin, A., Zomet, A., Peleg, S., and Weiss, Y. (2004). Seamless image stitching in the gradient domain. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 377–389, Prague.

Levoy, M. (1988). Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37.

Levoy, M. (2006). Light fields and computational imaging. *Computer*, 39(8):46–55.

Levoy, M. (2008). Technical perspective: Computational photography on large collections of images. *Communications of the ACM*, 51(10):86.

Levoy, M. and Hanrahan, P. (1996). Light field rendering. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 31–42, New Orleans.

Levoy, M. and Whitted, T. (1985). *The Use of Points as a Display Primitive*. Technical Report 85-022, University of North Carolina at Chapel Hill.

Levoy, M., Ng, R., Adams, A., Footer, M., and Horowitz, M. (2006). Light field microscopy. *ACM Transactions on Graphics*, 25(3):924–934.

Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D. *et al*. (2000). The digital Michelangelo project: 3D scanning of large statues. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 131–144.

Lew, M. S., Sebe, N., Djeraba, C., and Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2(1):1–19.

Leyvand, T., Cohen-Or, D., Dror, G., and Lischinski, D. (2008). Data-driven enhancement of facial attractiveness. *ACM Transactions on Graphics*, 27(3).

Lhuillier, M. and Quan, L. (2002). Match propagation for image-based modeling and rendering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1140–1146.

Lhuillier, M. and Quan, L. (2005). A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433.

Li, H. and Hartley, R. (2007). The 3D–3D registration problem revisited. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Li, L.-J. and Fei-Fei, L. (2010). Optimol: Automatic object picture collection via incremental model learning. *International Journal of Computer Vision*, 88(2):147–168.

Li, S. (1995). *Markov Random Field Modeling in Computer Vision*. Springer-Verlag.

Li, S. Z. and Jain, A. K. (eds). (2005). *Handbook of Face Recognition*, Springer.

Li, X., Wu, C., Zach, C., Lazebnik, S., and Frahm, J.-M. (2008). Modeling and recognition of landmark image collections using iconic scene graphs. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 427–440, Marseilles.

Li, Y. and Huttenlocher, D. P. (2008). Learning for optical flow using stochastic optimization. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 379–391, Marseilles.

Li, Y., Crandall, D. J., and Huttenlocher, D. P. (2009). Landmark classification in large-scale image collections. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Li, Y., Wang, T., and Shum, H.-Y. (2002). Motion texture: a two-level statistical model for character motion synthesis. *ACM Transactions on Graphics*, 21(3):465–472.

Li, Y., Shum, H.-Y., Tang, C.-K., and Szeliski, R. (2004). Stereo reconstruction from multiperspective panoramas. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):44–62.

Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. (2004). Lazy snapping. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 23(3):303–308.

Liang, L., Xiao, R., Wen, F., and Sun, J. (2008). Face alignment via component-based discriminative search. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 72–85, Marseilles.

Liang, L., Liu, C., Xu, Y.-Q., Guo, B., and Shum, H.-Y. (2001). Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 20(3):127–150.

Liebowitz, D. and Zisserman, A. (1998). Metric rectification for perspective images of planes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 482–488, Santa Barbara.

Lim, J. (1990). *Two-Dimensional Signal and Image Processing*. Prentice-Hall, Englewood, NJ.

Lim, J. J., Arbeláez, P., Gu, C., and Malik, J. (2009). Context by region ancestry. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Lin, D., Kapoor, A., Hua, G., and Baker, S. (2010). Joint people, event, and location recognition in personal photo collections using cross-domain context. In *Eleventh European Conference on Computer Vision (ECCV 2010)*, Heraklion, Crete.

Lin, W.-C., Hays, J., Wu, C., Kwatra, V., and Liu, Y. (2006). Quantitative evaluation of near regular texture synthesis algorithms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 427–434, New York City, NY.

Lindeberg, T. (1990). Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254.

Lindeberg, T. (1993). Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318.

Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270.

Lindeberg, T. (1998a). Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):116–154.

Lindeberg, T. (1998b). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116.

Lindeberg, T. and Garding, J. (1997). Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *Image and Vision Computing*, 15(6):415–434.

Lippman, A. (1980). Movie maps: An application of the optical videodisc to computer graphics. *Computer Graphics (SIGGRAPH '80)*, 14(3):32–43.

Lischinski, D., Farbman, Z., Uyttendaele, M., and Szeliski, R. (2006a). Interactive local adjustment of tonal values. *ACM Transactions on Graphics (Proc. SIGGRAPH 2006)*, 25(3):646–653.

Lischinski, D., Farbman, Z., Uyttendaele, M., and Szeliski, R. (2006b). Interactive local adjustment of tonal values. *ACM Transactions on Graphics*, 25(3):646–653.

Litvinov, A. and Schechner, Y. Y. (2005). Radiometric framework for image mosaicking. *Journal of the Optical Society of America A*, 22(5):839–848.

Litwinowicz, P. (1997). Processing images and video for an impressionist effect. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 407–414.

Litwinowicz, P. and Williams, L. (1994). Animating images with drawings. In *ACM SIGGRAPH 1994 Conference Proceedings*, pp. 409–412.

Liu, C. (2009). *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. Ph.D. thesis, Massachusetts Institute of Technology.

Liu, C., Yuen, J., and Torralba, A. (2009). Nonparametric scene parsing: Label transfer via dense scene alignment. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Liu, C., Freeman, W. T., Adelson, E., and Weiss, Y. (2008). Human-assisted motion annotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Liu, C., Szeliski, R., Kang, S. B., Zitnick, C. L., and Freeman, W. T. (2008). Automatic estimation and removal of noise from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):299–314.

Liu, F., Gleicher, M., Jin, H., and Agarwala, A. (2009). Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics*, 28(3).

Liu, X., Chen, T., and Kumar, B. V. (2003). Face authentication for multiple subjects using eigenflow. *Pattern Recognition*, 36(2):313–328.

Liu, Y., Collins, R. T., and Tsin, Y. (2004). A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):354–371.

Liu, Y., Lin, W.-C., and Hays, J. (2004). Near-regular texture analysis and manipulation. *ACM Transactions on Graphics*, 23(3):368–376.

Livingstone, M. (2008). *Vision and Art: The Biology of Seeing*. Abrams, New York.

Lobay, A. and Forsyth, D. A. (2006). Shape from texture without boundaries. *International Journal of Computer Vision*, 67(1):71–91.

Lombaert, H., Sun, Y., Grady, L., and Xu, C. (2005). A multilevel banded graph cuts method for fast image segmentation. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 259–265, Beijing, China.

Longere, P., Delahunt, P. B., Zhang, X., and Brainard, D. H. (2002). Perceptual assessment of demosaicing algorithm performance. *Proceedings of the IEEE*, 90(1):123–132.

Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135.

Loop, C. and Zhang, Z. (1999). Computing rectifying homographies for stereo vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 125–131, Fort Collins.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics (SIGGRAPH '87)*, 21(4):163–169.

Lorusso, A., Eggert, D., and Fisher, R. B. (1995). A comparison of four algorithms for estimating 3-D rigid transformations. In *British Machine Vision Conference (BMVC95)*, pp. 237–246, Birmingham, England.

Lourakis, M. I. A. and Argyros, A. A. (2009). SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1).

Lowe, D. G. (1988). Organization of smooth image curves at multiple scales. In *Second International Conference on Computer Vision (ICCV'88)*, pp. 558–567, Tampa.

Lowe, D. G. (1989). Organization of smooth image curves at multiple scales. *International Journal of Computer Vision*, 3(2):119–130.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Seventh International Conference on Computer Vision (ICCV'99)*, pp. 1150–1157, Kerkyra, Greece.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application in stereo vision. In *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, pp. 674–679, Vancouver.

Luong, Q.-T. and Faugeras, O. D. (1996). The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–75.

Luong, Q.-T. and Viéville, T. (1996). Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2):193–229.

Lyu, S. and Simoncelli, E. (2008). Nonlinear image representation using divisive normalization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Lyu, S. and Simoncelli, E. (2009). Modeling multiscale subbands of photographic images with fields of Gaussian scale mixtures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):693–706.

Ma, W.-C., Jones, A., Chiang, J.-Y., Hawkins, T., Frederiksen, S., Peers, P., Vukovic, M., Ouhyoung, M., and Debevec, P. (2008). Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Transactions on Graphics*, 27(5).

Ma, Y., Derksen, H., Hong, W., and Wright, J. (2007). Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562.

MacDonald, L. (ed.). (2006). *Digital Heritage: Applying Digital Imaging to Cultural Heritage*, Butterworth-Heinemann.

Madsen, K., Nielsen, H. B., and Tingleff, O. (2004). Methods for non-linear least squares problems. Informatics and Mathematical Modelling, Technical University of Denmark (DTU).

Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., and Suetens, P. (1997). Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198.

Magnor, M. (2005). *Video-Based Rendering*. A. K. Peters, Wellesley, MA.

Magnor, M. and Girod, B. (2000). Data compression for light-field rendering. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(3):338–343.

Magnor, M., Ramanathan, P., and Girod, B. (2003). Multi-view coding for image-based rendering using 3-D scene geometry. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1092–1106.

Mahajan, D., Huang, F.-C., Matusik, W., Ramamoorthi, R., and Belhumeur, P. (2009). Moving gradients: A path-based method for plausible image interpolation. *ACM Transactions on Graphics*, 28(3).

Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the Mars exploration rovers. *Journal of Field Robotics*, 24(3).

Maire, M., Arbelaez, P., Fowlkes, C., and Malik, J. (2008). Using contours to detect and localize junctions in natural images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. (2010). Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *IEEE International Conference on Robotics and Automation*, Anchorage, AK.

Maitre, M., Shinagawa, Y., and Do, M. N. (2008). Symmetric multi-view stereo reconstruction from planar camera arrays. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Maji, S., Berg, A., and Malik, J. (2008). Classification using intersection kernel support vector machines is efficient. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Malik, J. and Rosenholtz, R. (1997). Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23(2):149–168.

Malik, J., Belongie, S., Leung, T., and Shi, J. (2001). Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27.

Malisiewicz, T. and Efros, A. A. (2008). Recognition by association via learning per-exemplar distances. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Malladi, R., Sethian, J. A., and Vemuri, B. C. (1995). Shape modeling with front propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–176.

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(7):674–693.

Malvar, H. S. (1990). Lapped transforms for efficient transform/subband coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(6):969–978.

Malvar, H. S. (1998). Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts. *IEEE Transactions on Signal Processing*, 46(4):1043–1053.

Malvar, H. S. (2000). Fast progressive image coding without wavelets. In *IEEE Data Compressions Conference*, pp. 243–252, Snowbird, UT.

Malvar, H. S., He, L.-W., and Cutler, R. (2004). High-quality linear interpolation for demosaicing of Bayer-patterned color images. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, pp. 485–488, Montreal.

Mancini, T. A. and Wolff, L. B. (1992). 3D shape and light source location from depth and reflectance. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, pp. 707–709, Champaign, Illinois.

Manjunathi, B. S. and Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842.

Mann, S. and Picard, R. W. (1994). Virtual bellows: Constructing high-quality images from video. In *First IEEE International Conference on Image Processing (ICIP-94)*, pp. 363–367, Austin.

Mann, S. and Picard, R. W. (1995). On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. In *IS&T's 48th Annual Conference*, pp. 422–428, Washington, D. C.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, San Francisco.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London*, B 207:187–217.

Marr, D. and Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. Roy. Soc. London, B*, 200:269–294.

Marr, D. and Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, 194:283–287.

Marr, D. C. and Poggio, T. (1979). A computational theory of human stereo vision. *Proceedings of the Royal Society of London*, B 204:301–328.

Marroquin, J., Mitter, S., and Poggio, T. (1985). Probabilistic solution of ill-posed problems in computational vision. In *Image Understanding Workshop*, pp. 293–309, Miami Beach.

Marroquin, J., Mitter, S., and Poggio, T. (1987). Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 82(397):76–89.

Marroquin, J. L. (1983). *Design of Cooperative Networks*. Working Paper 253, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Martin, D., Fowlkes, C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 416–423, Vancouver, Canada.

Martin, W. N. and Aggarwal, J. K. (1983). Volumetric description of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):150–158.

Martinec, D. and Pajdla, T. (2007). Robust rotation and translation estimation in multiview reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Massey, M. and Bender, W. (1996). Salient stills: Process and practice. *IBM Systems Journal*, 35(3&4):557–573.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767.

Matei, B. C. and Meer, P. (2006). Estimation of nonlinear errors-in-variables models for computer vision applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1537–1552.

Matsushita, Y. and Lin, S. (2007). Radiometric calibration from noise distributions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Matsushita, Y., Ofek, E., Ge, W., Tang, X., and Shum, H.-Y. (2006). Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1150–1163.

Matthews, I. and Baker, S. (2004). Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164.

Matthews, I., Xiao, J., and Baker, S. (2007). 2D vs. 3D deformable face models: Representational power, construction, and real-time fitting. *International Journal of Computer Vision*, 75(1):93–113.

Matthies, L., Kanade, T., and Szeliski, R. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–236.

Matusik, W., Buehler, C., and McMillan, L. (2001). Polyhedral visual hulls for real-time rendering. In *12th Eurographics Workshop on Rendering Techniques*, pp. 115–126, London.

Matusik, W., Buehler, C., Raskar, R., Gortler, S. J., and McMillan, L. (2000). Image-based visual hulls. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 369–374.

Mayhew, J. E. W. and Frisby, J. P. (1980). The computation of binocular edges. *Perception*, 9:69–87.

Mayhew, J. E. W. and Frisby, J. P. (1981). Psychophysical and computational studies towards a theory of human stereopsis. *Artificial Intelligence*, 17(1-3):349–408.

McCamy, C. S., Marcus, H., and Davidson, J. G. (1976). A color-rendition chart. *Journal of Applied Photogrammetric Engineering*, 2(3):95–99.

McCane, B., Novins, K., Crannitch, D., and Galvin, B. (2001). On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):126–143.

McGuire, M., Matusik, W., Pfister, H., Hughes, J. F., and Durand, F. (2005). Defocus video matting. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):567–576.

McInerney, T. and Terzopoulos, D. (1993). A finite element model for 3D shape reconstruction and nonrigid motion tracking. In *Fourth International Conference on Computer Vision (ICCV'93)*, pp. 518–523, Berlin, Germany.

McInerney, T. and Terzopoulos, D. (1996). Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2):91–108.

McInerney, T. and Terzopoulos, D. (1999). Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Transactions on Medical Imaging*, 18(10):840–850.

McInerney, T. and Terzopoulos, D. (2000). T-snakes: Topology adaptive snakes. *Medical Image Analysis*, 4:73–91.

McLauchlan, P. F. (2000). A batch/recursive algorithm for 3D scene reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, pp. 738–743, Hilton Head Island.

McLauchlan, P. F. and Jaenicke, A. (2002). Image mosaicing using sequential bundle adjustment. *Image and Vision Computing*, 20(9-10):751–759.

McLean, G. F. and Kotturi, D. (1995). Vanishing point detection by line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1090–1095.

McMillan, L. and Bishop, G. (1995). Plenoptic modeling: An image-based rendering system. In *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 39–46.

McMillan, L. and Gortler, S. (1999). Image-based rendering: A new interface between computer vision and computer graphics. *Computer Graphics*, 33(4):61–64.

Meehan, J. (1990). *Panoramic Photography*. Watson-Guptill.

Meer, P. and Georgescu, B. (2001). Edge detection with embedded confidence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1351–1365.

Meilă, M. and Shi, J. (2000). Learning segmentation by random walks. In *Advances in Neural Information Processing Systems*.

Meilă, M. and Shi, J. (2001). A random walks view of spectral segmentation. In *Workshop on Artificial Intelligence and Statistics*, pp. 177–182, Key West, FL.

Meltzer, J. and Soatto, S. (2008). Edge descriptors for robust wide-baseline correspondence. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Meltzer, T., Yanover, C., and Weiss, Y. (2005). Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 428–435, Beijing, China.

Mémin, E. and Pérez, P. (2002). Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 44(2):129–155.

Menet, S., Saint-Marc, P., and Medioni, G. (1990a). Active contour models: overview, implementation and applications. In *IEEE International Conference on Systems, Man and Cybernetics*, pp. 194–199, Los Angeles.

Menet, S., Saint-Marc, P., and Medioni, G. (1990b). B-snakes: implementation and applications to stereo. In *Image Understanding Workshop*, pp. 720–726, Pittsburgh.

Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.-M., Yang, R., Nister, D., and Pollefeys, M. (2007). Real-time visibility-based fusion of depth maps. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Mertens, T., Kautz, J., and Reeth, F. V. (2007). Exposure fusion. In *Proceedings of Pacific Graphics 2007*, pp. 382–390.

Metaxas, D. and Terzopoulos, D. (2002). Dynamic deformation of solid primitives with constraints. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002)*, 21(3):309–312.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091.

Meyer, C. D. (2000). *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, Philadephia.

Meyer, Y. (1993). *Wavelets: Algorithms and Applications*. Society for Industrial and Applied Mathematics, Philadephia.

Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.

Mikolajczyk, K., Schmid, C., and Zisserman, A. (2004). Human detection based on a probabilistic assembly of robust part detectors. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 69–82, Prague.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. J. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72.

Milgram, D. L. (1975). Computer methods for creating photomosaics. *IEEE Transactions on Computers*, C-24(11):1113–1119.

Milgram, D. L. (1977). Adaptive techniques for photomosaicking. *IEEE Transactions on Computers*, C-26(11):1175–1180.

Miller, I., Campbell, M., Huttenlocher, D., Kline, F.-R., Nathan, A. *et al*. (2008). Team Cornell's Skynet: Robust perception and planning in an urban environment. *Journal of Field Robotics*, 25(8):493–527.

Mitiche, A. and Bouthemy, P. (1996). Computation and analysis of image motion: A synopsis of current problems and methods. *International Journal of Computer Vision*, 19(1):29–55.

Mitsunaga, T. and Nayar, S. K. (1999). Radiometric self calibration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 374–380, Fort Collins.

Mittal, A. and Davis, L. S. (2003). M_2 tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203.

Mičušík, B. and Košecká, J. (2009). Piecewise planar city 3D modeling from street view panoramic sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Mičušìk, B., Wildenauer, H., and Košecká, J. (2008). Detection and matching of rectilinear structures. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Moeslund, T. B. and Granum, E. (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268.

Moeslund, T. B., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126.

Moezzi, S., Katkere, A., Kuramura, D., and Jain, R. (1996). Reality modeling and visualization from multiple video sequences. *IEEE Computer Graphics and Applications*, 16(6):58–63.

Moghaddam, B. and Pentland, A. (1997). Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710.

Moghaddam, B., Jebara, T., and Pentland, A. (2000). Bayesian face recognition. *Pattern Recognition*, 33(11):1771–1782.

Mohan, A., Papageorgiou, C., and Poggio, T. (2001). Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361.

Möller, K. D. (1988). *Optics*. University Science Books, Mill Valley, CA.

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D. *et al*. (2008). Junior: The Stanford entry in the Urban Challenge. *Journal of Field Robotics*, 25(9):569–597.

Moon, P. and Spencer, D. E. (1981). *The Photic Field*. MIT Press, Cambridge, Massachusetts.

Moons, T., Van Gool, L., and Vergauwen, M. (2010). 3D reconstruction from multiple images. *Foundations and Trends in Computer Graphics and Computer Vision*, 4(4).

Moosmann, F., Nowak, E., and Jurie, F. (2008). Randomized clustering forests for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1632–1646.

Moravec, H. (1977). Towards automatic visual obstacle avoidance. In *Fifth International Joint Conference on Artificial Intelligence (IJCAI'77)*, p. 584, Cambridge, Massachusetts.

Moravec, H. (1983). The Stanford cart and the CMU rover. *Proceedings of the IEEE*, 71(7):872–884.

Moreno-Noguer, F., Lepetit, V., and Fua, P. (2007). Accurate non-iterative $O(n)$ solution to the PnP problem. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Mori, G. (2005). Guiding model search using segmentation. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 1417–1423, Beijing, China.

Mori, G., Ren, X., Efros, A., and Malik, J. (2004). Recovering human body configurations: Combining segmentation and recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 326–333, Washington, DC.

Mori, M. (1970). The uncanny valley. *Energy*, 7(4):33–35. http://www.androidscience.com/theuncannyvalley/proceedings2005/uncannyvalley.html.

Morimoto, C. and Chellappa, R. (1997). Fast 3D stabilization and mosaic construction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 660–665, San Juan, Puerto Rico.

Morita, T. and Kanade, T. (1997). A sequential factorization method for recovering shape and motion from image streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):858–867.

Morris, D. D. and Kanade, T. (1998). A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 696–702, Bombay.

Morrone, M. and Burr, D. (1988). Feature detection in human vision: A phase dependent energy model. *Proceedings of the Royal Society of London B*, 235:221–245.

Mortensen, E. N. (1999). Vision-assisted image editing. *Computer Graphics*, 33(4):55–57.

Mortensen, E. N. and Barrett, W. A. (1995). Intelligent scissors for image composition. In *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 191–198.

Mortensen, E. N. and Barrett, W. A. (1998). Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384.

Mortensen, E. N. and Barrett, W. A. (1999). Toboggan-based intelligent scissors with a four parameter edge model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 452–458, Fort Collins.

Mueller, P., Zeng, G., Wonka, P., and Van Gool, L. (2007). Image-based procedural modeling of facades. *ACM Transactions on Graphics*, 26(3).

Mühlich, M. and Mester, R. (1998). The role of total least squares in motion analysis. In *Fifth European Conference on Computer Vision (ECCV'98)*, pp. 305–321, Freiburg, Germany.

Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, Lisbon, Portugal.

Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and variational problems. *Comm. Pure Appl. Math.*, XLII(5):577–685.

Munder, S. and Gavrila, D. M. (2006). An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1863–1868.

Mundy, J. L. (2006). Object recognition in the geometric era: A retrospective. In Ponce, J., Hebert, M., Schmid, C., and Zisserman, A. (eds), *Toward Category-Level Object Recognition*, pp. 3–28, Springer, New York.

Mundy, J. L. and Zisserman, A. (eds). (1992). *Geometric Invariance in Computer Vision*. MIT Press, Cambridge, Massachusetts.

Murase, H. and Nayar, S. K. (1995). Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24.

Murphy, E. P. (2005). *A Testing Procedure to Characterize Color and Spatial Quality of Digital Cameras Used to Image Cultural Heritage*. Master's thesis, Rochester Institute of Technology.

Murphy, K., Torralba, A., and Freeman, W. T. (2003). Using the forest to see the trees: A graphical model relating features, objects, and scenes. In *Advances in Neural Information Processing Systems*.

Murphy-Chutorian, E. and Trivedi, M. M. (2009). Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626.

Murray, R. M., Li, Z. X., and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

Mutch, J. and Lowe, D. G. (2008). Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision*, 80(1):45–57.

Nagel, H. H. (1986). Image sequences—ten (octal) years—from phenomenology towards a theoretical foundation. In *Eighth International Conference on Pattern Recognition (ICPR'86)*, pp. 1174–1185, Paris.

Nagel, H.-H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(5):565–593.

Nakamura, Y., Matsuura, T., Satoh, K., and Ohta, Y. (1996). Occlusion detectable stereo—occlusion patterns in camera matrix. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pp. 371–378, San Francisco.

Nakao, T., Kashitani, A., and Kaneyoshi, A. (1998). Scanning a document with a small camera attached to a mouse. In *IEEE Workshop on Applications of Computer Vision (WACV'98)*, pp. 63–68, Princeton.

Nalwa, V. S. (1987). Edge-detector resolution improvement by image interpolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(3):446–451.

Nalwa, V. S. (1993). *A Guided Tour of Computer Vision*. Addison-Wesley, Reading, MA.

Nalwa, V. S. and Binford, T. O. (1986). On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):699–714.

Narasimhan, S. G. and Nayar, S. K. (2005). Enhancing resolution along multiple imaging dimensions using assorted pixels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):518–530.

Narayanan, P., Rander, P., and Kanade, T. (1998). Constructing virtual worlds using dense stereo. In *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 3–10, Bombay.

Nayar, S., Watanabe, M., and Noguchi, M. (1995). Real-time focus range sensor. In *Fifth International Conference on Computer Vision (ICCV'95)*, pp. 995–1001, Cambridge, Massachusetts.

Nayar, S. K. (2006). Computational cameras: Redefining the image. *Computer*, 39(8):30–38.

Nayar, S. K. and Branzoi, V. (2003). Adaptive dynamic range imaging: Optical control of pixel exposures over space and time. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 1168–1175, Nice, France.

Nayar, S. K. and Mitsunaga, T. (2000). High dynamic range imaging: Spatially varying pixel exposures. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, pp. 472–479, Hilton Head Island.

Nayar, S. K. and Nakagawa, Y. (1994). Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831.

Nayar, S. K., Ikeuchi, K., and Kanade, T. (1991). Shape from interreflections. *International Journal of Computer Vision*, 6(3):173–195.

Nayar, S. K., Watanabe, M., and Noguchi, M. (1996). Real-time focus range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1186–1198.

Negahdaripour, S. (1998). Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):961–979.

Nehab, D., Rusinkiewicz, S., Davis, J., and Ramamoorthi, R. (2005). Efficiently combining positions and normals for precise 3d geometry. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):536–543.

Nene, S. and Nayar, S. K. (1997). A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):989–1003.

Nene, S. A., Nayar, S. K., and Murase, H. (1996). *Columbia Object Image Library (COIL-100)*. Technical Report CUCS-006-96, Department of Computer Science, Columbia University.

Netravali, A. and Robbins, J. (1979). Motion-compensated television coding: Part 1. *Bell System Tech.*, 58(3):631–670.

Nevatia, R. (1977). A color edge detector and its use in scene segmentation. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-7(11):820–826.

Nevatia, R. and Binford, T. (1977). Description and recognition of curved objects. *Artificial Intelligence*, 8:77–98.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pp. 849–854.

Ng, R. (2005). Fourier slice photography. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):735–744.

Ng, R., Levoy, M., Bréedif, M., Duval, G., Horowitz, M., and Hanrahan, P. (2005). *Light Field Photography with a Hand-held Plenoptic Camera*. Technical Report CSTR 2005-02, Stanford University.

Nielsen, M., Florack, L. M. J., and Deriche, R. (1997). Regularization, scale-space, and edge-detection filters. *Journal of Mathematical Imaging and Vision*, 7(4):291–307.

Nielson, G. M. (1993). Scattered data modeling. *IEEE Computer Graphics and Applications*, 13(1):60–70.

Nir, T., Bruckstein, A. M., and Kimmel, R. (2008). Over-parameterized variational optical flow. *International Journal of Computer Vision*, 76(2):205–216.

Nishihara, H. K. (1984). Practical real-time imaging stereo matcher. *OptEng*, 23(5):536–545.

Nistér, D. (2003). Preemptive RANSAC for live structure and motion estimation. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 199–206, Nice, France.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–777.

Nistér, D. and Stewénius, H. (2006). Scalable recognition with a vocabulary tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 2161–2168, New York City, NY.

Nistér, D. and Stewénius, H. (2008). Linear time maximally stable extremal regions. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 183–196, Marseilles.

Nistér, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20.

Noborio, H., Fukada, S., and Arimoto, S. (1988). Construction of the octree approximating three-dimensional objects by using multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(6):769–782.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, second edition.

Nomura, Y., Zhang, L., and Nayar, S. K. (2007). Scene collages and flexible camera arrays. In *Eurographics Symposium on Rendering*.

Nordström, N. (1990). Biased anisotropic diffusion: A unified regularization and diffusion approach to edge detection. *Image and Vision Computing*, 8(4):318–327.

Nowak, E., Jurie, F., and Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 490–503.

Obdržálek, S. and Matas, J. (2006). Object recognition using local affine frames on maximally stable extremal regions. In Ponce, J., Hebert, M., Schmid, C., and Zisserman, A. (eds), *Toward Category-Level Object Recognition*, pp. 83–104, Springer, New York.

Oh, B. M., Chen, M., Dorsey, J., and Durand, F. (2001). Image-based modeling and photo editing. In *ACM SIGGRAPH 2001 Conference Proceedings*, pp. 433–442.

Ohlander, R., Price, K., and Reddy, D. R. (1978). Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8(3):313–333.

Ohta, Y. and Kanade, T. (1985). Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):139–154.

Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., and Seidel, H.-P. (2003). Multi-level partition of unity implicits. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, 22(3):463–470.

Okutomi, M. and Kanade, T. (1992). A locally adaptive window for signal matching. *International Journal of Computer Vision*, 7(2):143–162.

Okutomi, M. and Kanade, T. (1993). A multiple baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363.

Okutomi, M. and Kanade, T. (1994). A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932.

Oliensis, J. (2005). The least-squares error for structure from infinitesimal motion. *International Journal of Computer Vision*, 61(3):259–299.

Oliensis, J. and Hartley, R. (2007). Iterative extensions of the Sturm/Triggs algorithm: Convergence and nonconvergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2217–2233.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175.

Oliva, A. and Torralba, A. (2007). The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527.

Olsson, C., Eriksson, A. P., and Kahl, F. (2008). Improved spectral relaxation methods for binary quadratic optimization problems. *Computer Vision and Image Understanding*, 112(1):3–13.

Omer, I. and Werman, M. (2004). Color lines: Image specific color representation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 946–953, Washington, DC.

Ong, E.-J., Micilotta, A. S., Bowden, R., and Hilton, A. (2006). Viewpoint invariant exemplar-based 3D human tracking. *Computer Vision and Image Understanding*, 104(2-3):178–189.

Opelt, A., Pinz, A., and Zisserman, A. (2006). A boundary-fragment-model for object detection. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 575–588.

Opelt, A., Pinz, A., Fussenegger, M., and Auer, P. (2006). Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):614–641.

OpenGL-ARB. (1997). *OpenGL Reference Manual: The Official Reference Document to OpenGL, Version 1.1*. Addison-Wesley, Reading, MA, 2nd edition.

Oppenheim, A. V. and Schafer, A. S. (1996). *Signals and Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 2nd edition.

Oppenheim, A. V., Schafer, R. W., and Buck, J. R. (1999). *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 2nd edition.

Oren, M. and Nayar, S. (1997). A theory of specular surface geometry. *International Journal of Computer Vision*, 24(2):105–124.

O'Rourke, J. and Badler, N. I. (1980). Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):522–536.

Osher, S. and Paragios, N. (eds). (2003). *Geometric Level Set Methods in Imaging, Vision, and Graphics*, Springer.

Osuna, E., Freund, R., and Girosi, F. (1997). Training support vector machines: An application to face detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 130–136, San Juan, Puerto Rico.

O'Toole, A. J., Jiang, F., Roark, D., and Abdi, H. (2006). Predicting human face recognition. In Zhao, W.-Y. and Chellappa, R. (eds), *Face Processing: Advanced Methods and Models*, Elsevier.

O'Toole, A. J., Phillips, P. J., Jiang, F., Ayyad, J., Pénard, N., and Abdi, H. (2009). Face recognition algorithms surpass humans matching faces over changes in illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1642–1646.

Ott, M., Lewis, J. P., and Cox, I. J. (1993). Teleconferencing eye contact using a virtual camera. In *INTERACT'93 and CHI'93 conference companion on Human factors in computing systems*, pp. 109–110, Amsterdam.

Otte, M. and Nagel, H.-H. (1994). Optical flow estimation: advances and comparisons. In *Third European Conference on Computer Vision (ECCV'94)*, pp. 51–60, Stockholm, Sweden.

Oztireli, C., Guennebaud, G., and Gross, M. (2008). Feature preserving point set surfaces. *Computer Graphics Forum*, 28(2):493–501.

Özuysal, M., Calonder, M., Lepetit, V., and Fua, P. (2010). Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3).

Paglieroni, D. W. (1992). Distance transforms: Properties and machine vision applications. *Graphical Models and Image Processing*, 54(1):56–74.

Pal, C., Szeliski, R., Uyttendaele, M., and Jojic, N. (2004). Probability models for high dynamic range imaging. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 173–180, Washington, DC.

Palmer, S. E. (1999). *Vision Science: Photons to Phenomenology*. The MIT Press, Cambridge, Massachusetts.

Pankanti, S., Bolle, R. M., and Jain, A. K. (2000). Biometrics: The future of identification. *Computer*, 21(2):46–49.

Papageorgiou, C. and Poggio, T. (2000). A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33.

Papandreou, G. and Maragos, P. (2008). Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Papenberg, N., Bruhn, A., Brox, T., Didas, S., and Weickert, J. (2006). Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158.

Papert, S. (1966). *The Summer Vision Project*. Technical Report AIM-100, Artificial Intelligence Group, Massachusetts Institute of Technology. http://hdl.handle.net/1721.1/6125.

Paragios, N. and Deriche, R. (2000). Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280.

Paragios, N. and Sgallari, F. (2009). Special issue on scale space and variational methods in computer vision. *International Journal of Computer Vision*, 84(2).

Paragios, N., Faugeras, O. D., Chan, T., and Schnörr, C. (eds). (2005). *Third International Workshop on Variational, Geometric, and Level Set Methods in Computer Vision (VLSM 2005)*, Springer.

Paris, S. and Durand, F. (2006). A fast approximation of the bilateral filter using a signal processing approach. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 568–580.

Paris, S. and Durand, F. (2007). A topological approach to hierarchical segmentation using mean shift. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Paris, S., Kornprobst, P., Tumblin, J., and Durand, F. (2008). Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Computer Vision*, 4(1):1–73.

Park, M., Brocklehurst, K., Collins, R. T., and Liu, Y. (2009). Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1804–1816.

Park, S. C., Park, M. K., and Kang, M. G. (2003). Super-resolution image reconstruction: A technical overview. *IEEE Signal Processing Magazine*, 20:21–36.

Parke, F. I. and Waters, K. (1996). *Computer Facial Animation*. A K Peters, Wellesley, Massachusetts.

Parker, J. A., Kenyon, R. V., and Troxel, D. E. (1983). Comparison of interpolating methods for image resampling. *IEEE Transactions on Medical Imaging*, MI-2(1):31–39.

Pattanaik, S. N., Ferwerda, J. A., Fairchild, M. D., and Greenberg, D. P. (1998). A multiscale model of adaptation and spatial vision for realistic image display. In *ACM SIGGRAPH 1998 Conference Proceedings*, pp. 287–298, Orlando.

Pauly, M., Keiser, R., Kobbelt, L. P., and Gross, M. (2003). Shape modeling with point-sampled geometry. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, 21(3):641–650.

Pavlidis, T. (1977). *Structural Pattern Recognition*. Springer-Verlag, Berlin; New York.

Pavlidis, T. and Liow, Y.-T. (1990). Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):225–233.

Pavlović, V., Sharma, R., and Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers, Los Altos.

Peleg, R., Ben-Ezra, M., and Pritch, Y. (2001). Omnistereo: Panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):279–290.

Peleg, S. (1981). Elimination of seams from photomosaics. *Computer Vision, Graphics, and Image Processing*, 16(1):1206–1210.

Peleg, S. and Herman, J. (1997). Panoramic mosaics by manifold projection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 338–343, San Juan, Puerto Rico.

Peleg, S. and Rav-Acha, A. (2006). Lucas-Kanade without iterative warping. In *International Conference on Image Processing (ICIP-2006)*, pp. 1097–1100, Atlanta.

Peleg, S., Rousso, B., Rav-Acha, A., and Zomet, A. (2000). Mosaicing on adaptive manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1144–1154.

Penev, P. and Atick, J. (1996). Local feature analysis: A general statistical theory for object representation. *Network Computation and Neural Systems*, 7:477–500.

Pentland, A. P. (1984). Local shading analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(2):170–179.

Pentland, A. P. (1986). Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331.

Pentland, A. P. (1987). A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):523–531.

Pentland, A. P. (1994). Interpolation using wavelet bases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):410–414.

Pérez, P., Blake, A., and Gangnet, M. (2001). JetStream: Probabilistic contour extraction with particles. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 524–531, Vancouver, Canada.

Pérez, P., Gangnet, M., and Blake, A. (2003). Poisson image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)*, 22(3):313–318.

Perona, P. (1995). Deformable kernels for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):488–499.

Perona, P. and Malik, J. (1990a). Detecting and localizing edges composed of steps, peaks and roofs. In *Third International Conference on Computer Vision (ICCV'90)*, pp. 52–57, Osaka, Japan.

Perona, P. and Malik, J. (1990b). Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Peters, J. and Reif, U. (2008). *Subdivision Surfaces*. Springer.

Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., and Toyama, K. (2004). Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 23(3):664–672.

Pfister, H., Zwicker, M., van Baar, J., and Gross, M. (2000). Surfels: Surface elements as rendering primitives. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 335–342.

Pflugfelder, R. (2008). *Self-calibrating Cameras in Video Surveillance*. Ph.D. thesis, Graz University of Technology.

Philbin, J. and Zisserman, A. (2008). Object mining using a matching graph on very large image collections. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Bhubaneswar, India.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Philbin, J., Chum, O., Sivic, J., Isard, M., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Phillips, P. J., Moon, H., Rizvi, S. A., and Rauss, P. J. (2000). The FERET evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104.

Phillips, P. J., Scruggs, W. T., O'Toole, A. J., Flynn, P. J., Bowyer, K. W. *et al*. (2010). FRVT 2006 and ICE 2006 large-scale experimental results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):831–846.

Phong, B. T. (1975). Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317.

Pickup, L. C. (2007). *Machine Learning in Multi-frame Image Super-resolution*. Ph.D. thesis, University of Oxford.

Pickup, L. C. and Zisserman, A. (2009). Automatic retrieval of visual continuity errors in movies. In *ACM International Conference on Image and Video Retrieval*, Santorini, Greece.

Pickup, L. C., Capel, D. P., Roberts, S. J., and Zisserman, A. (2007). Overcoming registration uncertainty in image super-resolution: Maximize or marginalize? *EURASIP Journal on Advances in Signal Processing*, 2010(Article ID 23565).

Pickup, L. C., Capel, D. P., Roberts, S. J., and Zisserman, A. (2009). Bayesian methods for image super-resolution. *The Computer Journal*, 52.

Pighin, F., Szeliski, R., and Salesin, D. H. (2002). Modeling and animating realistic faces from images. *International Journal of Computer Vision*, 50(2):143–169.

Pighin, F., Hecker, J., Lischinski, D., Salesin, D. H., and Szeliski, R. (1998). Synthesizing realistic facial expressions from photographs. In *ACM SIGGRAPH 1998 Conference Proceedings*, pp. 75–84, Orlando.

Pilet, J., Lepetit, V., and Fua, P. (2008). Fast non-rigid surface detection, registration, and realistic augmentation. *International Journal of Computer Vision*, 76(2).

Pinz, A. (2005). Object categorization. *Foundations and Trends in Computer Graphics and Computer Vision*, 1(4):255–353.

Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A. *et al*. (1987). Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368.

Platel, B., Balmachnova, E., Florack, L., and ter Haar Romeny, B. (2006). Top-points as interest points for image matching. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 418–429.

Platt, J. C. (2000). Optimal filtering for patterned displays. *IEEE Signal Processing Letters*, 7(7):179–180.

Pock, T., Unger, M., Cremers, D., and Bischof, H. (2008). Fast and exact solution of total variation models on the GPU. In *CVPR 2008 Workshop on Visual Computer Vision on GPUs (CVGPU)*, Anchorage, AK.

Poelman, C. J. and Kanade, T. (1997). A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218.

Poggio, T. and Koch, C. (1985). Ill-posed problems in early vision: from computational theory to analogue networks. *Proceedings of the Royal Society of London*, B 226:303–323.

Poggio, T., Gamble, E., and Little, J. (1988). Parallel integration of vision modules. *Science*, 242(4877):436–440.

Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature*, 317(6035):314–319.

Poggio, T., Little, J., Gamble, E., Gillet, W., Geiger, D. *et al*. (1988). The MIT vision machine. In *Image Understanding Workshop*, pp. 177–198, Boston.

Polana, R. and Nelson, R. C. (1997). Detection and recognition of periodic, nonrigid motion. *International Journal of Computer Vision*, 23(3):261–282.

Pollard, S. B., Mayhew, J. E. W., and Frisby, J. P. (1985). PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470.

Pollefeys, M. and Van Gool, L. (2002). From images to 3D models. *Communications of the ACM*, 45(7):50–55.

Pollefeys, M., Nistér, D., Frahm, J.-M., Akbarzadeh, A., Mordohai, P. *et al*. (2008). Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167.

Ponce, J., Hebert, M., Schmid, C., and Zisserman, A. (eds). (2006). *Toward Category-Level Object Recognition*, Springer, New York.

Ponce, J., Berg, T., Everingham, M., Forsyth, D., Hebert, M. *et al*. (2006). Dataset issues in object recognition. In Ponce, J., Hebert, M., Schmid, C., and Zisserman, A. (eds), *Toward Category-Level Object Recognition*, pp. 29–48, Springer, New York.

Pons, J.-P., Keriven, R., and Faugeras, O. (2005). Modelling dynamic scenes by registering multi-view image sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 822–827, San Diego, CA.

Pons, J.-P., Keriven, R., and Faugeras, O. (2007). Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193.

Porter, T. and Duff, T. (1984). Compositing digital images. *Computer Graphics (SIGGRAPH '84)*, 18(3):253–259.

Portilla, J. and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–71.

Portilla, J., Strela, V., Wainwright, M., and Simoncelli, E. P. (2003). Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351.

Potetz, B. and Lee, T. S. (2008). Efficient belief propagation for higher-order cliques using linear constraint nodes. *Computer Vision and Image Understanding*, 112(1):39–54.

Potmesil, M. (1987). Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics, and Image Processing*, 40:1–29.

Pratt, W. K. (2007). *Digital Image Processing*. Wiley-Interscience, Hoboken, NJ, 4th edition.

Prazdny, K. (1985). Detection of binocular disparities. *Biological Cybernetics*, 52:93–99.

Pritchett, P. and Zisserman, A. (1998). Wide baseline stereo matching. In *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 754–760, Bombay.

Proesmans, M., Van Gool, L., and Defoort, F. (1998). Reading between the lines – a method for extracting dynamic 3D with texture. In *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 1081–1086, Bombay.

Protter, M. and Elad, M. (2009). Super resolution with probabilistic motion estimation. *IEEE Transactions on Image Processing*, 18(8):1899–1904.

Pullen, K. and Bregler, C. (2002). Motion capture assisted animation: texturing and synthesis. *ACM Transactions on Graphics*, 21(3):501–508.

Pulli, K. (1999). Multiview registration for large data sets. In *Second International Conference on 3D Digital Imaging and Modeling (3DIM'99)*, pp. 160–168, Ottawa, Canada.

Pulli, K., Abi-Rached, H., Duchamp, T., Shapiro, L., and Stuetzle, W. (1998). Acquisition and visualization of colored 3D objects. In *International Conference on Pattern Recognition (ICPR'98)*, pp. 11–15.

Quack, T., Leibe, B., and Van Gool, L. (2008). World-scale mining of objects and events from community photo collections. In *Conference on Image and Video Retrieval*, pp. 47–56, Niagara Falls.

Quam, L. H. (1984). Hierarchical warp stereo. In *Image Understanding Workshop*, pp. 149–155, New Orleans.

Quan, L. and Lan, Z. (1999). Linear N-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780.

Quan, L. and Mohr, R. (1989). Determining perspective structures using hierarchical Hough transform. *Pattern Recognition Letters*, 9(4):279–286.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. (2007). Objects in context. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Rademacher, P. and Bishop, G. (1998). Multiple-center-of-projection images. In *ACM SIGGRAPH 1998 Conference Proceedings*, pp. 199–206, Orlando.

Raginsky, M. and Lazebnik, S. (2009). Locality-sensitive binary codes from shift-invariant kernels. In *Advances in Neural Information Processing Systems*.

Raman, S. and Chaudhuri, S. (2007). A matte-less, variational approach to automatic scene compositing. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Raman, S. and Chaudhuri, S. (2009). Bilateral filter based compositing for variable exposure photography. In *Proceedings of Eurographics 2009*.

Ramanan, D. and Baker, S. (2009). Local distance functions: A taxonomy, new algorithms, and an evaluation. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Ramanan, D., Forsyth, D., and Zisserman, A. (2005). Strike a pose: Tracking people by finding stylized poses. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 271–278, San Diego, CA.

Ramanarayanan, G. and Bala, K. (2007). Constrained texture synthesis via energy minimization. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):167–178.

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256.

Ramnath, K., Koterba, S., Xiao, J., Hu, C., Matthews, I., Baker, S., Cohn, J., and Kanade, T. (2008). Multi-view AAM fitting and construction. *International Journal of Computer Vision*, 76(2):183–204.

Raskar, R. and Tumblin, J. (2010). *Computational Photography: Mastering New Techniques for Lenses, Lighting, and Sensors*. A K Peters, Wellesley, Massachusetts.

Raskar, R., Tan, K.-H., Feris, R., Yu, J., and Turk, M. (2004). Non-photorealistic camera: Depth edge detection and stylized rendering using multi-flash imaging. *ACM Transactions on Graphics*, 23(3):679–688.

Rav-Acha, A., Kohli, P., Fitzgibbon, A., and Rother, C. (2008). Unwrap mosaics: A new representation for video editing. *ACM Transactions on Graphics*, 27(3).

Rav-Acha, A., Pritch, Y., Lischinski, D., and Peleg, S. (2005). Dynamosaics: Video mosaics with non-chronological time. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 58–65, San Diego, CA.

Ravikumar, P., Agarwal, A., and Wainwright, M. J. (2008). Message-passing for graph-structured linear programs: Proximal projections, convergence and rounding schemes. In *International Conference on Machine Learning*, pp. 800–807.

Ray, S. F. (2002). *Applied Photographic Optics*. Focal Press, Oxford, 3rd edition.

Rehg, J. and Kanade, T. (1994). Visual tracking of high DOF articulated structures: an application to human hand tracking. In *Third European Conference on Computer Vision (ECCV'94)*, pp. 35–46, Stockholm, Sweden.

Rehg, J. and Witkin, A. (1991). Visual tracking with deformation models. In *IEEE International Conference on Robotics and Automation*, pp. 844–850, Sacramento.

Rehg, J., Morris, D. D., and Kanade, T. (2003). Ambiguities in visual tracking of articulated objects using two- and three-dimensional models. *International Journal of Robotics Research*, 22(6):393–418.

Reichenbach, S. E., Park, S. K., and Narayanswamy, R. (1991). Characterizing digital image acquisition devices. *Optical Engineering*, 30(2):170–177.

Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J. (2002). Photographic tone reproduction for digital images. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002)*, 21(3):267–276.

Reinhard, E., Ward, G., Pattanaik, S., and Debevec, P. (2005). *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann.

Rhemann, C., Rother, C., and Gelautz, M. (2008). Improving color modeling for alpha matting. In *British Machine Vision Conference (BMVC 2008)*, Leeds.

Rhemann, C., Rother, C., Rav-Acha, A., and Sharp, T. (2008). High resolution matting via interactive trimap segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., and Rott, P. (2009). A perceptually motivated online benchmark for image matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Richardson, I. E. G. (2003). *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. Wiley.

Rioul, O. and Vetterli, M. (1991). Wavelets and signal processing. *IEEE Signal Processing Magazine*, 8(4):14–38.

Rioux, M. and Bird, T. (1993). White laser, synced scan. *IEEE Computer Graphics and Applications*, 13(3):15–17.

Rioux, M., Bechthold, G., Taylor, D., and Duggan, M. (1987). Design of a large depth of view three-dimensional camera for robot vision. *Optical Engineering*, 26(12):1245–1250.

Riseman, E. M. and Arbib, M. A. (1977). Computational techniques in the visual segmentation of static scenes. *Computer Graphics and Image Processing*, 6(3):221–276.

Ritter, G. X. and Wilson, J. N. (2000). *Handbook of Computer Vision Algorithms in Image Algebra*. CRC Press, Boca Raton, 2nd edition.

Robert, C. P. (2007). *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer-Verlag, New York.

Roberts, L. G. (1965). Machine perception of three-dimensional solids. In Tippett, J. T., Borkowitz, D. A., Clapp, L. C., Koester, C. J., and Vanderburgh Jr., A. (eds), *Optical and Electro-Optical Information Processing*, pp. 159–197, MIT Press, Cambridge, Massachusetts.

Robertson, D. and Cipolla, R. (2004). An image-based system for urban navigation. In *British Machine Vision Conference*, pp. 656–665, Kingston.

Robertson, D. P. and Cipolla, R. (2002). Building architectural models from many views using map constraints. In *Seventh European Conference on Computer Vision (ECCV 2002)*, pp. 155–169, Copenhagen.

Robertson, D. P. and Cipolla, R. (2009). Architectural modelling. In Varga, M. (ed.), *Practical Image Processing and Computer Vision*, John Wiley.

Robertson, N. and Reid, I. (2006). A general method for human activity recognition in video. *Computer Vision and Image Understanding*, 104(2-3):232–248.

Roble, D. (1999). Vision in film and special effects. *Computer Graphics*, 33(4):58–60.

Roble, D. and Zafar, N. B. (2009). Don't trust your eyes: cutting-edge visual effects. *Computer*, 42(7):35–41.

Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., and Torr, P. H. S. (2008). Randomized trees for human pose detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Rogmans, S., Lu, J., Bekaert, P., and Lafruit, G. (2009). Real-time stereo-based views synthesis algorithms: A unified framework and evaluation on commodity GPUs. *Signal Processing: Image Communication*, 24:49–64.

Rohr, K. (1994). Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics, and Image Processing*, 59(1):94–115.

Román, A. and Lensch, H. P. A. (2006). Automatic multiperspective images. In *Eurographics Symposium on Rendering*, pp. 83–92.

Román, A., Garg, G., and Levoy, M. (2004). Interactive design of multi-perspective images for visualizing urban landscapes. In *IEEE Visualization 2004*, pp. 537–544, Minneapolis.

Romdhani, S. and Vetter, T. (2003). Efficient, robust and accurate fitting of a 3D morphable model. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 59–66, Nice, France.

Romdhani, S., Torr, P. H. S., Schölkopf, B., and Blake, A. (2001). Computationally efficient face detection. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 695–700, Vancouver, Canada.

Rosales, R. and Sclaroff, S. (2000). Inferring body pose without tracking body parts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, pp. 721–727, Hilton Head Island.

Rosenfeld, A. (1980). Quadtrees and pyramids for pattern recognition and image processing. In *Fifth International Conference on Pattern Recognition (ICPR'80)*, pp. 802–809, Miami Beach.

Rosenfeld, A. (ed.). (1984). *Multiresolution Image Processing and Analysis*, Springer-Verlag, New York.

Rosenfeld, A. and Davis, L. S. (1979). Image segmentation and image models. *Proceedings of the IEEE*, 67(5):764–772.

Rosenfeld, A. and Kak, A. C. (1976). *Digital Picture Processing*. Academic Press, New York.

Rosenfeld, A. and Pfaltz, J. L. (1966). Sequential operations in digital picture processing. *Journal of the ACM*, 13(4):471–494.

Rosenfeld, A., Hummel, R. A., and Zucker, S. W. (1976). Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6:420–433.

Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 1508–1515, Beijing, China.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 430–443.

Roth, S. and Black, M. J. (2007a). On the spatial statistics of optical flow. *International Journal of Computer Vision*, 74(1):33–50.

Roth, S. and Black, M. J. (2007b). Steerable random fields. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Roth, S. and Black, M. J. (2009). Fields of experts. *International Journal of Computer Vision*, 82(2):205–229.

Rother, C. (2002). A new approach for vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10):647–656.

Rother, C. (2003). Linear multi-view reconstruction of points, lines, planes and cameras using a reference plane. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 1210–1217, Nice, France.

Rother, C. and Carlsson, S. (2002). Linear multi view reconstruction and camera recovery using a reference plane. *International Journal of Computer Vision*, 49(2/3):117–141.

Rother, C., Kolmogorov, V., and Blake, A. (2004). "GrabCut"—interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 23(3):309–314.

Rother, C., Bordeaux, L., Hamadi, Y., and Blake, A. (2006). Autocollage. *ACM Transactions on Graphics*, 25(3):847–852.

Rother, C., Kohli, P., Feng, W., and Jia, J. (2009). Minimizing sparse higher order energy functions of discrete variables. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Rother, C., Kolmogorov, V., Lempitsky, V., and Szummer, M. (2007). Optimizing binary MRFs via extended roof duality. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Rother, C., Kumar, S., Kolmogorov, V., and Blake, A. (2005). Digital tapestry. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 589–596, San Diego, CA.

Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. (2006). 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3):231–259.

Rousseeuw, P. J. (1984). Least median of squares regresssion. *Journal of the American Statistical Association*, 79:871–880.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. Wiley, New York.

Rousson, M. and Paragios, N. (2008). Prior knowledge, level set representations, and visual grouping. *International Journal of Computer Vision*, 76(3):231–243.

Roweis, S. (1998). EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems*, pp. 626–632.

Rowland, D. A. and Perrett, D. I. (1995). Manipulating facial appearance through shape and color. *IEEE Computer Graphics and Applications*, 15(5):70–76.

Rowley, H. A., Baluja, S., and Kanade, T. (1998a). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38.

Rowley, H. A., Baluja, S., and Kanade, T. (1998b). Rotation invariant neural network-based face detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 38–44, Santa Barbara.

Roy, S. and Cox, I. J. (1998). A maximum-flow formulation of the N-camera stereo correspondence problem. In *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 492–499, Bombay.

Rozenfeld, S., Shimshoni, I., and Lindenbaum, M. (2007). Dense mirroring surface recovery from 1d homographies and sparse correspondences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E., McClelland, J. L., and the PDP research group (eds), *Parallel distributed processing: Explorations in the microstructure of cognition*, pp. 318–362, Bradford Books, Cambridge, Massachusetts.

Rusinkiewicz, S. and Levoy, M. (2000). Qsplat: A multiresolution point rendering system for large meshes. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 343–352.

Russ, J. C. (2007). *The Image Processing Handbook*. CRC Press, Boca Raton, 5th edition.

Russell, B., Efros, A., Sivic, J., Freeman, W., and Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 1605–1612, New York City, NY.

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173.

Russell, B. C., Torralba, A., Liu, C., Fergus, R., and Freeman, W. T. (2007). Object recognition by scene alignment. In *Advances in Neural Information Processing Systems*.

Ruzon, M. A. and Tomasi, C. (2000). Alpha estimation in natural images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, pp. 18–25, Hilton Head Island.

Ruzon, M. A. and Tomasi, C. (2001). Edge, junction, and corner detection using color distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1281–1295.

Ryan, T. W., Gray, R. T., and Hunt, B. R. (1980). Prediction of correlation errors in stereo-pair images. *Optical Engineering*, 19(3):312–322.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition.

Saint-Marc, P., Chen, J. S., and Medioni, G. (1991). Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):514–529.

Saito, H. and Kanade, T. (1999). Shape reconstruction in projective grid space from large number of images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 49–54, Fort Collins.

Samet, H. (1989). *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, Massachusetts.

Sander, P. T. and Zucker, S. W. (1990). Inferring surface trace and differential structure from 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):833–854.

Sapiro, G. (2001). *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press.

Sato, Y. and Ikeuchi, K. (1996). Reflectance analysis for 3D computer graphics model generation. *Graphical Models and Image Processing*, 58(5):437–451.

Sato, Y., Wheeler, M., and Ikeuchi, K. (1997). Object shape and reflectance modeling from observation. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 379–387, Los Angeles.

Savarese, S. and Fei-Fei, L. (2007). 3D generic object categorization, localization and pose estimation. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Savarese, S. and Fei-Fei, L. (2008). View synthesis for recognizing unseen poses of object classes. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 602–615, Marseilles.

Savarese, S., Chen, M., and Perona, P. (2005). Local shape from mirror reflections. *International Journal of Computer Vision*, 64(1):31–67.

Savarese, S., Andreetto, M., Rushmeier, H. E., Bernardini, F., and Perona, P. (2007). 3D reconstruction by shadow carving: Theory and practical evaluation. *International Journal of Computer Vision*, 71(3):305–336.

Sawhney, H. S. (1994). Simplifying motion and structure analysis using planar parallax and image warping. In *Twelfth International Conference on Pattern Recognition (ICPR'94)*, pp. 403–408, Jerusalem, Israel.

Sawhney, H. S. and Ayer, S. (1996). Compact representation of videos through dominant multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–830.

Sawhney, H. S. and Hanson, A. R. (1991). Identification and 3D description of 'shallow' environmental structure over a sequence of images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pp. 179–185, Maui, Hawaii.

Sawhney, H. S. and Kumar, R. (1999). True multi-image alignment and its application to mosaicing and lens distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):235–243.

Sawhney, H. S., Kumar, R., Gendel, G., Bergen, J., Dixon, D., and Paragano, V. (1998). VideoBrush: Experiences with consumer video mosaicing. In *IEEE Workshop on Applications of Computer Vision (WACV'98)*, pp. 56–62, Princeton.

Sawhney, H. S., Arpa, A., Kumar, R., Samarasekera, S., Aggarwal, M., Hsu, S., Nister, D., and Hanna, K. (2002). Video flashlights: real time rendering of multiple videos for immersive model visualization. In *Proceedings of the 13th Eurographics Workshop on Rendering*, pp. 157–168, Pisa, Italy.

Saxena, A., Sun, M., and Ng, A. Y. (2009). Make3D: Learning 3D scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840.

Schaffalitzky, F. and Zisserman, A. (2000). Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing*, 18:647–658.

Schaffalitzky, F. and Zisserman, A. (2002). Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Seventh European Conference on Computer Vision (ECCV 2002)*, pp. 414–431, Copenhagen.

Scharr, H., Black, M. J., and Haussecker, H. W. (2003). Image statistics and anisotropic diffusion. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 840–847, Nice, France.

Scharstein, D. (1994). Matching images by comparing their gradient fields. In *Twelfth International Conference on Pattern Recognition (ICPR'94)*, pp. 572–575, Jerusalem, Israel.

Scharstein, D. (1999). *View Synthesis Using Stereo Vision*. Volume 1583, Springer-Verlag.

Scharstein, D. and Pal, C. (2007). Learning conditional random fields for stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Scharstein, D. and Szeliski, R. (1998). Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42.

Scharstein, D. and Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, pp. 195–202, Madison, WI.

Schechner, Y. Y., Nayar, S. K., and Belhumeur, P. N. (2009). Multiplexing for optimal lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1339–1354.

Schindler, G., Brown, M., and Szeliski, R. (2007). City-scale location recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., and Dellaert, F. (2008). Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Schlesinger, D. and Flach, B. (2006). *Transforming an arbitrary minsum problem into a binary one*. Technical Report TUD-FI06-01, Dresden University of Technology.

Schlesinger, M. I. (1976). Syntactic analysis of two-dimensional visual signals in noisy conditions. *Kibernetika*, 4:113–130.

Schlesinger, M. I. and Giginyak, V. V. (2007a). Solution to structural recognition (max,+)-problems by their equivalent transformations – part 1. *Control Systems and Computers*, 2007(1):3–15.

Schlesinger, M. I. and Giginyak, V. V. (2007b). Solution to structural recognition (max,+)-problems by their equivalent transformations – part 2. *Control Systems and Computers*, 2007(2):3–18.

Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534.

Schmid, C. and Zisserman, A. (1997). Automatic line matching across views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 666–671, San Juan, Puerto Rico.

Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172.

Schneiderman, H. and Kanade, T. (2004). Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177.

Schödl, A. and Essa, I. (2002). Controlled animation of video sprites. In *ACM Symposium on Computater Animation*, San Antonio.

Schödl, A., Szeliski, R., Salesin, D. H., and Essa, I. (2000). Video textures. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 489–498, New Orleans.

Schoenemann, T. and Cremers, D. (2008). High resolution motion layer decomposition using dual-space graph cuts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Schölkopf, B. and Smola, A. (eds). (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, Massachusetts.

Schraudolph, N. N. (2010). Polynomial-time exact inference in NP-hard binary MRFs via reweighted perfect matching. In *13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 717–724.

Schröder, P. and Sweldens, W. (1995). Spherical wavelets: Efficiently representing functions on the sphere. In *ACM SIGGRAPH 1995 Conference Proceedings*, pp. 161–172.

Schultz, R. R. and Stevenson, R. L. (1996). Extraction of high-resolution frames from video sequences. *IEEE Transactions on Image Processing*, 5(6):996–1011.

Sclaroff, S. and Isidoro, J. (2003). Active blobs: region-based, deformable appearance models. *Computer Vision and Image Understanding*, 89(2-3):197–225.

Scott, G. L. and Longuet-Higgins, H. C. (1990). Feature grouping by relocalization of eigenvectors of the proximity matrix. In *British Machine Vision Conference*, pp. 103–108.

Sebastian, T. B. and Kimia, B. B. (2005). Curves vs. skeletons in object recognition. *Signal Processing*, 85(2):246–263.

Sederberg, T. W. and Parry, S. R. (1986). Free-form deformations of solid geometric models. *Computer Graphics (SIGGRAPH '86)*, 20(4):151–160.

Sederberg, T. W., Gao, P., Wang, G., and Mu, H. (1993). 2D shape blending: An intrinsic solution to the vertex path problem. In *ACM SIGGRAPH 1993 Conference Proceedings*, pp. 15–18.

Seitz, P. (1989). Using local orientation information as image primitive for robust object recognition. In *SPIE Vol. 1199, Visual Communications and Image Processing IV*, pp. 1630–1639.

Seitz, S. (2001). The space of all stereo images. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 26–33, Vancouver, Canada.

Seitz, S. and Szeliski, R. (1999). Applications of computer vision to computer graphics. *Computer Graphics*, 33(4):35–37. Guest Editors' introduction to the Special Issue.

Seitz, S., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 519–526, New York, NY.

Seitz, S. M. and Baker, S. (2009). Filter flow. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Seitz, S. M. and Dyer, C. M. (1996). View morphing. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 21–30, New Orleans.

Seitz, S. M. and Dyer, C. M. (1997). Photorealistic scene reconstruction by voxel coloring. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 1067–1073, San Juan, Puerto Rico.

Seitz, S. M. and Dyer, C. M. (1999). Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173.

Seitz, S. M. and Dyer, C. R. (1997). View invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25(3):231–251.

Serra, J. (1982). *Image Analysis and Mathematical Morphology*. Academic Press, New York.

Serra, J. and Vincent, L. (1992). An overview of morphological filtering. *Circuits, Systems and Signal Processing*, 11(1):47–108.

Serre, T., Wolf, L., and Poggio, T. (2005). Object recognition with features inspired by visual cortex. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 994–1000, San Diego, CA.

Sethian, J. (1999). *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, 2nd edition.

Shade, J., Gortler, S., He, L., and Szeliski, R. (1998). Layered depth images. In *ACM SIGGRAPH 1998 Conference Proceedings*, pp. 231–242, Orlando.

Shade, J., Lischinski, D., Salesin, D., DeRose, T., and Snyder, J. (1996). Hierarchical images caching for accelerated walkthroughs of complex environments. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 75–82, New Orleans.

Shafer, S. A. (1985). Using color to separate reflection components. *COLOR Research and Applications*, 10(4):210–218.

Shafer, S. A., Healey, G., and Wolff, L. (1992). *Physics-Based Vision: Principles and Practice*. Jones & Bartlett, Cambridge, MA.

Shafique, K. and Shah, M. (2005). A noniterative greedy algorithm for multiframe point correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):51–65.

Shah, J. (1993). A nonlinear diffusion model for discontinuous disparity and half-occlusion in stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pp. 34–40, New York.

Shakhnarovich, G., Darrell, T., and Indyk, P. (eds). (2006). *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, MIT Press.

Shakhnarovich, G., Viola, P., and Darrell, T. (2003). Fast pose estimation with parameter-sensitive hashing. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 750–757, Nice, France.

Shan, Y., Liu, Z., and Zhang, Z. (2001). Model-based bundle adjustment with application to face modeling. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 644–641, Vancouver, Canada.

Sharon, E., Galun, M., Sharon, D., Basri, R., and Brandt, A. (2006). Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442(7104):810–813.

Shashua, A. and Toelg, S. (1997). The quadric reference surface: Theory and applications. *International Journal of Computer Vision*, 23(2):185–198.

Shashua, A. and Wexler, Y. (2001). Q-warping: Direct computation of quadratic reference surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):920–925.

Shaw, D. and Barnes, N. (2006). Perspective rectangle detection. In *Workshop on Applications of Computer Vision at ECCV'2006*.

Shewchuk, J. R. (1994). An introduction to the conjugate gradient method without the agonizing pain. Unpublished manuscript, available on author's homepage (http://www.cs.berkeley.edu/~jrs/). An earlier version appeared as a Carnegie Mellon University Technical Report, CMU-CS-94-125.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(22):888–905.

Shi, J. and Tomasi, C. (1994). Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 593–600, Seattle.

Shimizu, M. and Okutomi, M. (2001). Precise sub-pixel estimation on area-based matching. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 90–97, Vancouver, Canada.

Shirley, P. (2005). *Fundamentals of Computer Graphics*. A K Peters, Wellesley, Massachusetts, second edition.

Shizawa, M. and Mase, K. (1991). A unified computational theory of motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pp. 289–295, Maui, Hawaii.

Shoemake, K. (1985). Animating rotation with quaternion curves. *Computer Graphics (SIGGRAPH '85)*, 19(3):245–254.

Shotton, J., Blake, A., and Cipolla, R. (2005). Contour-based learning for object detection. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 503–510, Beijing, China.

Shotton, J., Johnson, M., and Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling appearance, shape and context. *International Journal of Computer Vision*, 81(1):2–23.

Shufelt, J. (1999). Performance evaluation and analysis of vanishing point detection techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):282–288.

Shum, H.-Y. and He, L.-W. (1999). Rendering with concentric mosaics. In *ACM SIGGRAPH 1999 Conference Proceedings*, pp. 299–306, Los Angeles.

Shum, H.-Y. and Szeliski, R. (1999). Stereo reconstruction from multiperspective panoramas. In *Seventh International Conference on Computer Vision (ICCV'99)*, pp. 14–21, Kerkyra, Greece.

Shum, H.-Y. and Szeliski, R. (2000). Construction of panoramic mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130. Erratum published July 2002, 48(2):151–152.

Shum, H.-Y., Chan, S.-C., and Kang, S. B. (2007). *Image-Based Rendering*. Springer, New York, NY.

Shum, H.-Y., Han, M., and Szeliski, R. (1998). Interactive construction of 3D models from panoramic mosaics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 427–433, Santa Barbara.

Shum, H.-Y., Ikeuchi, K., and Reddy, R. (1995). Principal component analysis with missing data and its application to polyhedral modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):854–867.

Shum, H.-Y., Kang, S. B., and Chan, S.-C. (2003). Survey of image-based representations and compression techniques. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1020–1037.

Shum, H.-Y., Wang, L., Chai, J.-X., and Tong, X. (2002). Rendering by manifold hopping. *International Journal of Computer Vision*, 50(2):185–201.

Shum, H.-Y., Sun, J., Yamazaki, S., Li, Y., and Tang, C.-K. (2004). Pop-up light field: An interactive image-based modeling and rendering system. *ACM Transactions on Graphics*, 23(2):143–162.

Sidenbladh, H. and Black, M. J. (2003). Learning the statistics of people in images and video. *International Journal of Computer Vision*, 54(1):189–209.

Sidenbladh, H., Black, M. J., and Fleet, D. J. (2000). Stochastic tracking of 3D human figures using 2D image motion. In *Sixth European Conference on Computer Vision (ECCV 2000)*, pp. 702–718, Dublin, Ireland.

Sigal, L. and Black, M. J. (2006). Predicting 3D people from 2D pictures. In *AMDO 2006 - IV Conference on Articulated Motion and Deformable Objects*, pp. 185–195, Mallorca, Spain.

Sigal, L., Balan, A., and Black, M. J. (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4–27.

Sigal, L., Bhatia, S., Roth, S., Black, M. J., and Isard, M. (2004). Tracking loose-limbed people. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2004)*, pp. 421–428, Washington, DC.

Sillion, F. and Puech, C. (1994). *Radiosity and Global Illumination*. Morgan Kaufmann.

Sim, T., Baker, S., and Bsat, M. (2003). The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618.

Simard, P. Y., Bottou, L., Haffner, P., and Cun, Y. L. (1998). Boxlets: a fast convolution algorithm for signal processing and neural networks. In *Advances in Neural Information Processing Systems 13*, pp. 571–577.

Simon, I. and Seitz, S. M. (2008). Scene segmentation using the wisdom of crowds. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 541–553, Marseilles.

Simon, I., Snavely, N., and Seitz, S. M. (2007). Scene summarization for online image collections. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Simoncelli, E. P. (1999). Bayesian denoising of visual images in the wavelet domain. In Müller, P. and Vidakovic, B. (eds), *Bayesian Inference in Wavelet Based Models*, pp. 291–308, Springer-Verlag, New York.

Simoncelli, E. P. and Adelson, E. H. (1990a). Non-separable extensions of quadrature mirror filters to multiple dimensions. *Proceedings of the IEEE*, 78(4):652–664.

Simoncelli, E. P. and Adelson, E. H. (1990b). Subband transforms. In Woods, J. (ed.), *Subband Coding*, pp. 143–191, Kluwer Academic Press, Norwell, MA.

Simoncelli, E. P., Adelson, E. H., and Heeger, D. J. (1991). Probability distributions of optic flow. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pp. 310–315, Maui, Hawaii.

Simoncelli, E. P., Freeman, W. T., Adelson, E. H., and Heeger, D. J. (1992). Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38(3):587–607.

Singaraju, D., Grady, L., and Vidal, R. (2008). Interactive image segmentation via minimization of quadratic energies on directed graphs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Singaraju, D., Rother, C., and Rhemann, C. (2009). New appearance models for natural image matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Singaraju, D., Grady, L., Sinop, A. K., and Vidal, R. (2010). A continuous valued MRF for image segmentation. In Blake, A., Kohli, P., and Rother, C. (eds), *Advances in Markov Random Fields*, MIT Press.

Sinha, P., Balas, B., Ostrovsky, Y., and Russell, R. (2006). Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962.

Sinha, S., Mordohai, P., and Pollefeys, M. (2007). Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Sinha, S. N. and Pollefeys, M. (2005). Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 349–356, Beijing, China.

Sinha, S. N., Steedly, D., and Szeliski, R. (2009). Piecewise planar stereo for image-based rendering. In *Twelfth IEEE International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Sinha, S. N., Steedly, D., Szeliski, R., Agrawala, M., and Pollefeys, M. (2008). Interactive 3D architectural modeling from unordered photo collections. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2008)*, 27(5).

Sinop, A. K. and Grady, L. (2007). A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 1470–1477, Nice, France.

Sivic, J. and Zisserman, A. (2009). Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–606.

Sivic, J., Everingham, M., and Zisserman, A. (2009). "Who are you?"—Learning person specific classifiers from video. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Sivic, J., Zitnick, C. L., and Szeliski, R. (2006). Finding people in repeated shots of the same scene. In *British Machine Vision Conference (BMVC 2006)*, pp. 909–918, Edinburgh.

Sivic, J., Russell, B., Zisserman, A., Freeman, W. T., and Efros, A. A. (2008). Unsupervised discovery of visual object class hierarchies. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., and Freeman, W. T. (2005). Discovering objects and their localization in images. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 370–377, Beijing, China.

Slabaugh, G. G., Culbertson, W. B., Slabaugh, T. G., Culbertson, B., Malzbender, T., and Stevens, M. (2004). Methods for volumetric reconstruction of visual scenes. *International Journal of Computer Vision*, 57(3):179–199.

Slama, C. C. (ed.). (1980). *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, Virginia, fourth edition.

Smelyanskiy, V. N., Cheeseman, P., Maluf, D. A., and Morris, R. D. (2000). Bayesian super-resolved surface reconstruction from images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, pp. 375–382, Hilton Head Island.

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. C. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):477–490.

Sminchisescu, C. and Triggs, B. (2001). Covariance scaled sampling for monocular 3D body tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, pp. 447–454, Kauai, Hawaii.

Sminchisescu, C., Kanaujia, A., and Metaxas, D. (2006). Conditional models for contextual human motion recognition. *Computer Vision and Image Understanding*, 104(2-3):210–220.

Sminchisescu, C., Kanaujia, A., Li, Z., and Metaxas, D. (2005). Discriminative density propagation for 3D human motion estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 390–397, San Diego, CA.

Smith, A. R. and Blinn, J. F. (1996). Blue screen matting. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 259–268, New Orleans.

Smith, B. M., Zhang, L., Jin, H., and Agarwala, A. (2009). Light field video stabilization. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Smith, S. M. and Brady, J. M. (1997). SUSAN—a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78.

Smolic, A. and Kauff, P. (2005). Interactive 3-D video representation and coding technologies. *Proceedings of the IEEE*, 93(1):98–110.

Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (Proc. SIGGRAPH 2006)*, 25(3):835–846.

Snavely, N., Seitz, S. M., and Szeliski, R. (2008a). Modeling the world from Internet photo collections. *International Journal of Computer Vision*, 80(2):189–210.

Snavely, N., Seitz, S. M., and Szeliski, R. (2008b). Skeletal graphs for efficient structure from motion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Snavely, N., Garg, R., Seitz, S. M., and Szeliski, R. (2008). Finding paths through the world's photos. *ACM Transactions on Graphics (Proc. SIGGRAPH 2008)*, 27(3).

Snavely, N., Simon, I., Goesele, M., Szeliski, R., and Seitz, S. M. (2010). Scene reconstruction and visualization from community photo collections. *Proceedings of the IEEE*, 98(8):1370–1390.

Soatto, S., Yezzi, A. J., and Jin, H. (2003). Tales of shape and radiance in multiview stereo. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 974–981, Nice, France.

Soille, P. (2006). Morphological image compositing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):673–683.

Solina, F. and Bajcsy, R. (1990). Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–147.

Sontag, D. and Jaakkola, T. (2007). New outer bounds on the marginal polytope. In *Advances in Neural Information Processing Systems*.

Sontag, D., Meltzer, T., Globerson, A., Jaakkola, T., and Weiss, Y. (2008). Tightening LP relaxations for MAP using message passing. In *Uncertainty in Artificial Intelligence (UAI)*.

Soucy, M. and Laurendeau, D. (1992). Multi-resolution surface modeling from multiple range views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'92)*, pp. 348–353, Champaign, Illinois.

Srinivasan, S., Chellappa, R., Veeraraghavan, A., and Aggarwal, G. (2005). Electronic image stabilization and mosaicking algorithms. In Bovik, A. (ed.), *Handbook of Image and Video Processing*, Academic Press.

Srivasan, P., Liang, P., and Hackwood, S. (1990). Computational geometric methods in volumetric intersections for 3D reconstruction. *Pattern Recognition*, 23(8):843–857.

Stamos, I., Liu, L., Chen, C., Wolberg, G., Yu, G., and Zokai, S. (2008). Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. *International Journal of Computer Vision*, 78(2-3):237–260.

Stark, J. A. (2000). Adaptive image contrast enhancement using generalizations of histogram equalization. *IEEE Transactions on Image Processing*, 9(5):889–896.

Stauffer, C. and Grimson, W. (1999). Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 246–252, Fort Collins.

Steedly, D. and Essa, I. (2001). Propagation of innovative information in non-linear least-squares structure from motion. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 223–229, Vancouver, Canada.

Steedly, D., Essa, I., and Dellaert, F. (2003). Spectral partitioning for structure from motion. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 996–1003, Nice, France.

Steedly, D., Pal, C., and Szeliski, R. (2005). Efficiently registering video into panoramic mosaics. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 1300–1307, Beijing, China.

Steele, R. and Jaynes, C. (2005). Feature uncertainty arising from covariant image noise. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 1063–1070, San Diego, CA.

Steele, R. M. and Jaynes, C. (2006). Overconstrained linear estimation of radial distortion and multi-view geometry. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 253–264.

Stein, A., Hoiem, D., and Hebert, M. (2007). Learning to extract object boundaries using motion cues. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Stein, F. and Medioni, G. (1992). Structural indexing: Efficient 3-D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145.

Stein, G. (1995). Accurate internal camera calibration using rotation, with analysis of sources of error. In *Fifth International Conference on Computer Vision (ICCV'95)*, pp. 230–236, Cambridge, Massachusetts.

Stein, G. (1997). Lens distortion calibration using point correspondences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 602–608, San Juan, Puerto Rico.

Stenger, B., Thayananthan, A., Torr, P. H. S., and Cipolla, R. (2006). Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1372–1384.

Stewart, C. V. (1999). Robust parameter estimation in computer vision. *SIAM Reviews*, 41(3):513–537.

Stiller, C. and Konrad, J. (1999). Estimating motion in image sequences: A tutorial on modeling and computation of 2D motion. *IEEE Signal Processing Magazine*, 16(4):70–91.

Stollnitz, E. J., DeRose, T. D., and Salesin, D. H. (1996). *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann, San Francisco.

Strang, G. (1988). *Linear Algebra and its Applications*. Harcourt, Brace, Jovanovich, Publishers, San Diego, 3rd edition.

Strang, G. (1989). Wavelets and dilation equations: A brief introduction. *SIAM Reviews*, 31(4):614–627.

Strecha, C., Fransens, R., and Van Gool, L. (2006). Combined depth and outlier estimation in multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 2394–2401, New York City, NY.

Strecha, C., Tuytelaars, T., and Van Gool, L. (2003). Dense matching of multiple wide-baseline views. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 1194–1201, Nice, France.

Strecha, C., von Hansen, W., Van Gool, L., Fua, P., and Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Sturm, P. (2005). Multi-view geometry for general camera models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 206–212, San Diego, CA.

Sturm, P. and Ramalingam, S. (2004). A generic concept for camera calibration. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 1–13, Prague.

Sturm, P. and Triggs, W. (1996). A factorization based algorithm for multi-image projective structure and motion. In *Fourth European Conference on Computer Vision (ECCV'96)*, pp. 709–720, Cambridge, England.

Su, H., Sun, M., Fei-Fei, L., and Savarese, S. (2009). Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Sudderth, E. B., Torralba, A., Freeman, W. T., and Willsky, A. S. (2008). Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77(1-3):291–330.

Sullivan, S. and Ponce, J. (1998). Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1091–1096.

Sun, D., Roth, S., Lewis, J. P., and Black, M. J. (2008). Learning optical flow. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 83–97, Marseilles.

Sun, J., Zheng, N., and Shum, H. (2003). Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800.

Sun, J., Jia, J., Tang, C.-K., and Shum, H.-Y. (2004). Poisson matting. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 23(3):315–321.

Sun, J., Li, Y., Kang, S. B., and Shum, H.-Y. (2006). Flash matting. *ACM Transactions on Graphics*, 25(3):772–778.

Sun, J., Yuan, L., Jia, J., and Shum, H.-Y. (2004). Image completion with structure propagation. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 24(3):861–868.

Sun, M., Su, H., Savarese, S., and Fei-Fei, L. (2009). A multi-view probabilistic model for 3D object classes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Sung, K.-K. and Poggio, T. (1998). Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51.

Sutherland, I. E. (1974). Three-dimensional data input by tablet. *Proceedings of the IEEE*, 62(4):453–461.

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7(1):11–32.

Swaminathan, R., Kang, S. B., Szeliski, R., Criminisi, A., and Nayar, S. K. (2002). On the motion and appearance of specularities in image sequences. In *Seventh European Conference on Computer Vision (ECCV 2002)*, pp. 508–523, Copenhagen.

Sweldens, W. (1996). Wavelets and the lifting scheme: A 5 minute tour. *Z. Angew. Math. Mech.*, 76 (Suppl. 2):41–44.

Sweldens, W. (1997). The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546.

Swendsen, R. H. and Wang, J.-S. (1987). Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88.

Szeliski, R. (1986). *Cooperative Algorithms for Solving Random-Dot Stereograms*. Technical Report CMU-CS-86-133, Computer Science Department, Carnegie Mellon University.

Szeliski, R. (1989). *Bayesian Modeling of Uncertainty in Low-Level Vision*. Kluwer Academic Publishers, Boston.

Szeliski, R. (1990a). Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–301.

Szeliski, R. (1990b). Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):513–528.

Szeliski, R. (1991a). Fast shape from shading. *CVGIP: Image Understanding*, 53(2):129–153.

Szeliski, R. (1991b). Shape from rotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pp. 625–630, Maui, Hawaii.

Szeliski, R. (1993). Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32.

Szeliski, R. (1994). Image mosaicing for tele-reality applications. In *IEEE Workshop on Applications of Computer Vision (WACV'94)*, pp. 44–53, Sarasota.

Szeliski, R. (1996). Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30.

Szeliski, R. (1999). A multi-view approach to motion and stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. 157–163, Fort Collins.

Szeliski, R. (2006a). Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Computer Vision*, 2(1):1–104.

Szeliski, R. (2006b). Locally adapted hierarchical basis preconditioning. *ACM Transactions on Graphics (Proc. SIGGRAPH 2006)*, 25(3):1135–1143.

Szeliski, R. and Coughlan, J. (1997). Spline-based image registration. *International Journal of Computer Vision*, 22(3):199–218.

Szeliski, R. and Golland, P. (1999). Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61. Special Issue for Marr Prize papers.

Szeliski, R. and Hinton, G. (1985). Solving random-dot stereograms using the heat equation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'85)*, pp. 284–288, San Francisco.

Szeliski, R. and Ito, M. R. (1986). New Hermite cubic interpolator for two-dimensional curve generation. *IEE Proceedings E*, 133(6):341–347.

Szeliski, R. and Kang, S. B. (1994). Recovering 3D shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28.

Szeliski, R. and Kang, S. B. (1995). Direct methods for visual scene reconstruction. In *IEEE Workshop on Representations of Visual Scenes*, pp. 26–33, Cambridge, Massachusetts.

Szeliski, R. and Kang, S. B. (1997). Shape ambiguities in structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):506–512.

Szeliski, R. and Lavallée, S. (1996). Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. *International Journal of Computer Vision*, 18(2):171–186.

Szeliski, R. and Scharstein, D. (2004). Sampling the disparity space image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):419–425.

Szeliski, R. and Shum, H.-Y. (1996). Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199–1210.

Szeliski, R. and Shum, H.-Y. (1997). Creating full view panoramic image mosaics and texture-mapped models. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 251–258, Los Angeles.

Szeliski, R. and Tonnesen, D. (1992). Surface modeling with oriented particle systems. *Computer Graphics (SIGGRAPH '92)*, 26(2):185–194.

Szeliski, R. and Torr, P. (1998). Geometrically constrained structure from motion: Points on planes. In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE)*, pp. 171–186, Freiburg, Germany.

Szeliski, R. and Weiss, R. (1998). Robust shape recovery from occluding contours using a linear smoother. *International Journal of Computer Vision*, 28(1):27–44.

Szeliski, R., Avidan, S., and Anandan, P. (2000). Layer extraction from multiple images containing reflections and transparency. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2000)*, pp. 246–253, Hilton Head Island.

Szeliski, R., Tonnesen, D., and Terzopoulos, D. (1993a). Curvature and continuity control in particle-based surface models. In *SPIE Vol. 2031, Geometric Methods in Computer Vision II*, pp. 172–181, San Diego.

Szeliski, R., Tonnesen, D., and Terzopoulos, D. (1993b). Modeling surfaces of arbitrary topology with dynamic particles. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pp. 82–87, New York.

Szeliski, R., Uyttendaele, M., and Steedly, D. (2008). *Fast Poisson Blending using Multi-Splines*. Technical Report MSR-TR-2008-58, Microsoft Research.

Szeliski, R., Winder, S., and Uyttendaele, M. (2010). *High-quality multi-pass image resampling*. Technical Report MSR-TR-2010-10, Microsoft Research.

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. (2008). A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1068–1080.

Szummer, M. and Picard, R. W. (1996). Temporal texture modeling. In *IEEE International Conference on Image Processing (ICIP-96)*, pp. 823–826, Lausanne.

Tabb, M. and Ahuja, N. (1997). Multiscale image segmentation by integrated edge and region detection. *IEEE Transactions on Image Processing*, 6(5):642–655.

Taguchi, Y., Wilburn, B., and Zitnick, C. L. (2008). Stereo reconstruction with mixed pixels using adaptive over-segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Tanaka, M. and Okutomi, M. (2008). Locally adaptive learning for translation-variant MRF image priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Tao, H., Sawhney, H. S., and Kumar, R. (2001). A global matching framework for stereo computation. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 532–539, Vancouver, Canada.

Tappen, M. F. (2007). Utilizing variational optimization to learn Markov random fields. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Tappen, M. F. and Freeman, W. T. (2003). Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 900–907, Nice, France.

Tappen, M. F., Freeman, W. T., and Adelson, E. H. (2005). Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1459–1472.

Tappen, M. F., Russell, B. C., and Freeman, W. T. (2003). Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *Third International Workshop on Statistical and Computational Theories of Vision*, Nice, France.

Tappen, M. F., Liu, C., Freeman, W., and Adelson, E. (2007). Learning Gaussian conditional random fields for low-level vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Tardif, J.-P. (2009). Non-iterative approach for fast and accurate vanishing point detection. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Tardif, J.-P., Sturm, P., and Roy, S. (2007). Plane-based self-calibration of radial distortion. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Tardif, J.-P., Sturm, P., Trudeau, M., and Roy, S. (2009). Calibration of cameras with radially symmetric distortion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1552–1566.

Taubin, G. (1995). Curve and surface smoothing without shrinkage. In *Fifth International Conference on Computer Vision (ICCV'95)*, pp. 852–857, Cambridge, Massachusetts.

Taubman, D. S. and Marcellin, M. W. (2002). Jpeg2000: standard for interactive imaging. *Proceedings of the IEEE*, 90(8):1336–1357.

Taylor, C. J. (2003). Surface reconstruction from feature based stereo. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 184–190, Nice, France.

Taylor, C. J., Debevec, P. E., and Malik, J. (1996). Reconstructing polyhedral models of architectural scenes from photographs. In *Fourth European Conference on Computer Vision (ECCV'96)*, pp. 659–668, Cambridge, England.

Taylor, C. J., Kriegman, D. J., and Anandan, P. (1991). Structure and motion in two dimensions from multiple images: A least squares approach. In *IEEE Workshop on Visual Motion*, pp. 242–248, Princeton, New Jersey.

Taylor, P. (2009). *Text-to-Speech Synthesis*. Cambridge University Press, Cambridge.

Tek, K. and Kimia, B. B. (2003). Symmetry maps of free-form curve segments via wave propagation. *International Journal of Computer Vision*, 54(1-3):35–81.

Tekalp, M. (1995). *Digital Video Processing*. Prentice Hall, Upper Saddle River, NJ.

Telea, A. (2004). An image inpainting technique based on fast marching method. *Journal of Graphics Tools*, 9(1):23–34.

Teller, S., Antone, M., Bodnar, Z., Bosse, M., Coorg, S., Jethwa, M., and Master, N. (2003). Calibrated, registered images of an extended urban area. *International Journal of Computer Vision*, 53(1):93–107.

Teodosio, L. and Bender, W. (1993). Salient video stills: Content and context preserved. In *ACM Multimedia 93*, pp. 39–46, Anaheim, California.

Terzopoulos, D. (1983). Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 24:52–96.

Terzopoulos, D. (1986a). Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):129–139.

Terzopoulos, D. (1986b). Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):413–424.

Terzopoulos, D. (1988). The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(4):417–438.

Terzopoulos, D. (1999). Visual modeling for computer animation: Graphics with a vision. *Computer Graphics*, 33(4):42–45.

Terzopoulos, D. and Fleischer, K. (1988). Deformable models. *The Visual Computer*, 4(6):306–331.

Terzopoulos, D. and Metaxas, D. (1991). Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714.

Terzopoulos, D. and Szeliski, R. (1992). Tracking with Kalman snakes. In Blake, A. and Yuille, A. L. (eds), *Active Vision*, pp. 3–20, MIT Press, Cambridge, Massachusetts.

Terzopoulos, D. and Waters, K. (1990). Analysis of facial images using physical and anatomical models. In *Third International Conference on Computer Vision (ICCV'90)*, pp. 727–732, Osaka, Japan.

Terzopoulos, D. and Witkin, A. (1988). Physically-based models with rigid and deformable components. *IEEE Computer Graphics and Applications*, 8(6):41–51.

Terzopoulos, D., Witkin, A., and Kass, M. (1987). Symmetry-seeking models and 3D object reconstruction. *International Journal of Computer Vision*, 1(3):211–221.

Terzopoulos, D., Witkin, A., and Kass, M. (1988). Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123.

Thayananthan, A., Iwasaki, M., and Cipolla, R. (2008). Principled fusion of high-level model and low-level cues for motion segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Thirthala, S. and Pollefeys, M. (2005). The radial trifocal tensor: A tool for calibrating the radial distortion of wide-angle cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 321–328, San Diego, CA.

Thomas, D. B., Luk, W., Leong, P. H., and Villasenor, J. D. (2007). Gaussian random number generators. *ACM Computing Surveys*, 39(4).

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The MIT Press, Cambridge, Massachusetts.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A. *et al*. (2006). Stanley, the robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692.

Tian, Q. and Huhns, M. N. (1986). Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35:220–233.

Tikhonov, A. N. and Arsenin, V. Y. (1977). *Solutions of Ill-Posed Problems*. V. H. Winston, Washington, D. C.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal components analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622.

Toint, P. L. (1987). On large scale nonlinear least squares calculations. *SIAM J. Sci. Stat. Comput.*, 8(3):416–435.

Tola, E., Lepetit, V., and Fua, P. (2010). DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830.

Tolliver, D. and Miller, G. (2006). Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 1053–1060, New York City, NY.

Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154.

Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 839–846, Bombay.

Tombari, F., Mattoccia, S., and Di Stefano, L. (2007). Segmentation-based adaptive support for accurate stereo correspondence. In *Pacific-Rim Symposium on Image and Video Technology*.

Tombari, F., Mattoccia, S., Di Stefano, L., and Addimanda, E. (2008). Classification and evaluation of cost aggregation methods for stereo correspondence. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Tommasini, T., Fusiello, A., Trucco, E., and Roberto, V. (1998). Making good features track better. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 178–183, Santa Barbara.

Torborg, J. and Kajiya, J. T. (1996). Talisman: Commodity realtime 3D graphics for the PC. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 353–363, New Orleans.

Torr, P. H. S. (2002). Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61.

Torr, P. H. S. and Fitzgibbon, A. W. (2004). Invariant fitting of two view geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):648–650.

Torr, P. H. S. and Murray, D. (1997). The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300.

Torr, P. H. S., Szeliski, R., and Anandan, P. (1999). An integrated Bayesian approach to layer extraction from image sequences. In *Seventh International Conference on Computer Vision (ICCV'99)*, pp. 983–990, Kerkyra, Greece.

Torr, P. H. S., Szeliski, R., and Anandan, P. (2001). An integrated Bayesian approach to layer extraction from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):297–303.

Torralba, A. (2003). Contextual priming for object detection. *International Journal of Computer Vision*, 53(2):169–191.

Torralba, A. (2007). Classifier-based methods. In *CVPR 2007 Short Course on Recognizing and Learning Object Categories*. http://people.csail.mit.edu/torralba/shortCourseRLOC/.

Torralba, A. (2008). Object recognition and scene understanding. MIT Course 6.870, http://people.csail.mit.edu/torralba/courses/6.870/6.870.recognition.htm.

Torralba, A., Freeman, W. T., and Fergus, R. (2008). 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970.

Torralba, A., Murphy, K. P., and Freeman, W. T. (2004). Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems*.

Torralba, A., Murphy, K. P., and Freeman, W. T. (2007). Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869.

Torralba, A., Weiss, Y., and Fergus, R. (2008). Small codes and large databases of images for object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. (2003). Context-based vision system for place and object recognition. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 273–280, Nice, France.

Torrance, K. E. and Sparrow, E. M. (1967). Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America A*, 57(9):1105–1114.

Torresani, L., Hertzmann, A., and Bregler, C. (2008). Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892.

Toyama, K. (1998). *Prolegomena for Robust Face Tracking*. Technical Report MSR-TR-98-65, Microsoft Research.

Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: Principles and practice of background maintenance. In *Seventh International Conference on Computer Vision (ICCV'99)*, pp. 255–261, Kerkyra, Greece.

Tran, S. and Davis, L. (2002). 3D surface reconstruction using graph cuts with surface constraints. In *Seventh European Conference on Computer Vision (ECCV 2002)*, pp. 219–231, Copenhagen.

Trefethen, L. N. and Bau, D. (1997). *Numerical Linear Algebra*. SIAM.

Treisman, A. (1985). Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31(2):156–177.

Triggs, B. (1996). Factorization methods for projective structure and motion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pp. 845–851, San Francisco.

Triggs, B. (2004). Detecting keypoints with stable position, orientation, and scale under illumination changes. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 100–113, Prague.

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (1999). Bundle adjustment — a modern synthesis. In *International Workshop on Vision Algorithms*, pp. 298–372, Kerkyra, Greece.

Trobin, W., Pock, T., Cremers, D., and Bischof, H. (2008). Continuous energy minimization via repeated binary fusion. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 677–690, Marseilles.

Troccoli, A. and Allen, P. (2008). Building illumination coherent 3D models of large-scale outdoor scenes. *International Journal of Computer Vision*, 78(2-3):261–280.

Trottenberg, U., Oosterlee, C. W., and Schuller, A. (2000). *Multigrid*. Academic Press.

Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Upper Saddle River, NJ.

Tsai, P. S. and Shah, M. (1994). Shape from shading using linear approximation. *Image and Vision Computing*, 12:487–498.

Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344.

Tschumperlé, D. (2006). Curvature-preserving regularization of multi-valued images using PDEs. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 295–307.

Tschumperlé, D. and Deriche, R. (2005). Vector-valued image regularization with PDEs: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:506–517.

Tsin, Y., Kang, S. B., and Szeliski, R. (2006). Stereo matching with linear superposition of layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):290–301.

Tsin, Y., Ramesh, V., and Kanade, T. (2001). Statistical calibration of CCD imaging process. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 480–487, Vancouver, Canada.

Tu, Z., Chen, X., Yuille, A. L., and Zhu, S.-C. (2005). Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, 63(2):113–140.

Tumblin, J. and Rushmeier, H. E. (1993). Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6):42–48.

Tumblin, J. and Turk, G. (1999). LCIS: A boundary hierarchy for detail-preserving contrast reduction. In *ACM SIGGRAPH 1999 Conference Proceedings*, pp. 83–90, Los Angeles.

Tumblin, J., Agrawal, A., and Raskar, R. (2005). Why I want a gradient camera. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 103–110, San Diego, CA.

Turcot, P. and Lowe, D. G. (2009). Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshop on Emergent Issues in Large Amounts of Visual Data (WS-LAVD)*, Kyoto, Japan.

Turk, G. and Levoy, M. (1994). Zippered polygonal meshes from range images. In *ACM SIGGRAPH 1994 Conference Proceedings*, pp. 311–318.

Turk, G. and O'Brien, J. (2002). Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics*, 21(4):855–873.

Turk, M. and Pentland, A. (1991a). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

Turk, M. and Pentland, A. (1991b). Face recognition using eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pp. 586–591, Maui, Hawaii.

Tuytelaars, T. and Mikolajczyk, K. (2007). Local invariant feature detectors. *Foundations and Trends in Computer Graphics and Computer Vision*, 3(1).

Tuytelaars, T. and Van Gool, L. (2004). Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85.

Tuytelaars, T., Van Gool, L., and Proesmans, M. (1997). The cascaded Hough transform. In *International Conference on Image Processing (ICIP'97)*, pp. 736–739.

Ullman, S. (1979). The interpretation of structure from motion. *Proceedings of the Royal Society of London*, B-203:405–426.

Unnikrishnan, R., Pantofaru, C., and Hebert, M. (2007). Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):828–944.

Unser, M. (1999). Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R. *et al*. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466.

Urtasun, R., Fleet, D. J., and Fua, P. (2006). Temporal motion models for monocular and multiview 3D human body tracking. *Computer Vision and Image Understanding*, 104(2-3):157–177.

Uyttendaele, M., Eden, A., and Szeliski, R. (2001). Eliminating ghosting and exposure artifacts in image mosaics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, pp. 509–516, Kauai, Hawaii.

Uyttendaele, M., Criminisi, A., Kang, S. B., Winder, S., Hartley, R., and Szeliski, R. (2004). Image-based interactive exploration of real-world environments. *IEEE Computer Graphics and Applications*, 24(3):52–63.

Vaillant, R. and Faugeras, O. D. (1992). Using extremal boundaries for 3-D object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):157–173.

Vaish, V., Szeliski, R., Zitnick, C. L., Kang, S. B., and Levoy, M. (2006). Reconstructing occluded surfaces using synthetic apertures: Shape from focus vs. shape from stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 2331–2338, New York, NY.

van de Weijer, J. and Schmid, C. (2006). Coloring local feature extraction. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 334–348.

van den Hengel, A., Dick, A., Thormhlen, T., Ward, B., and Torr, P. H. S. (2007). Videotrace: Rapid interactive scene modeling from video. *ACM Transactions on Graphics*, 26(3).

Van Huffel, S. and Lemmerling, P. (eds). (2002). *Total Least Squares and Errors-in-Variables Modeling*, Springer.

Van Huffel, S. and Vandewalle, J. (1991). *The Total Least Squares Problem: Computational Aspects and Analysis*. Society for Industrial and Applied Mathematics, Philadephia.

van Ouwerkerk, J. D. (2006). Image super-resolution survey. *Image and Vision Computing*, 24(10):1039–1052.

Varma, M. and Ray, D. (2007). Learning the discriminative power-invariance trade-off. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Vasconcelos, N. (2007). From pixels to semantic spaces: Advances in content-based image retrieval. *Computer*, 40(7):20–26.

Vasilescu, M. A. O. and Terzopoulos, D. (2007). Multilinear (tensor) image synthesis, analysis, and recognition. *IEEE Signal Processing Magazine*, 24(6):118–123.

Vedaldi, A. and Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/.

Vedaldi, A., Gulshan, V., Varma, M., and Zisserman, A. (2009). Multiple kernels for object detection. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Vedula, S., Baker, S., and Kanade, T. (2005). Image-based spatio-temporal modeling and view interpolation of dynamic events. *ACM Transactions on Graphics*, 24(2):240–261.

Vedula, S., Baker, S., Rander, P., Collins, R., and Kanade, T. (2005). Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):475–480.

Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., and Tumblin, J. (2007). Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Transactions on Graphics*, 26(3).

Veksler, O. (1999). *Efficient Graph-based Energy Minimization Methods in Computer Vision*. Ph.D. thesis, Cornell University.

Veksler, O. (2001). Stereo matching by compact windows via minimum ratio cycle. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 540–547, Vancouver, Canada.

Veksler, O. (2003). Fast variable window for stereo correspondence using integral images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, pp. 556–561, Madison, WI.

Veksler, O. (2007). Graph cut based optimization for MRFs with truncated convex priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Verbeek, J. and Triggs, B. (2007). Region classification with Markov field aspect models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Vergauwen, M. and Van Gool, L. (2006). Web-based 3D reconstruction service. *Machine Vision and Applications*, 17(2):321–329.

Vetter, T. and Poggio, T. (1997). Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742.

Vezhnevets, V., Sazonov, V., and Andreeva, A. (2003). A survey on pixel-based skin color detection techniques. In *GRAPHICON03*, pp. 85–92.

Vicente, S., Kolmogorov, V., and Rother, C. (2008). Graph cut based image segmentation with connectivity priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Vidal, R., Ma, Y., and Sastry, S. S. (2010). *Generalized Principal Component Analysis*. Springer.

Viéville, T. and Faugeras, O. D. (1990). Feedforward recovery of motion and structure from a sequence of 2D-lines matches. In *Third International Conference on Computer Vision (ICCV'90)*, pp. 517–520, Osaka, Japan.

Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–596.

Vineet, V. and Narayanan, P. J. (2008). CUDA cuts: Fast graph cuts on the GPU. In *CVPR 2008 Workshop on Visual Computer Vision on GPUs (CVGPU)*, Anchorage, AK.

Viola, P. and Wells III, W. (1997). Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154.

Viola, P., Jones, M. J., and Snow, D. (2003). Detecting pedestrians using patterns of motion and appearance. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 734–741, Nice, France.

Viola, P. A. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154.

Vlasic, D., Baran, I., Matusik, W., and Popović, J. (2008). Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics*, 27(3).

Vlasic, D., Brand, M., Pfister, H., and Popović, J. (2005). Face transfer with multilinear models. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):426–433.

Vogiatzis, G., Torr, P., and Cipolla, R. (2005). Multi-view stereo via volumetric graph-cuts. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 391–398, San Diego, CA.

Vogiatzis, G., Hernandez, C., Torr, P., and Cipolla, R. (2007). Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2241–2246.

von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In *CHI'04: SIGCHI Conference on Human Factors in Computing Systems*, pp. 319–326, Vienna, Austria.

von Ahn, L., Liu, R., and Blum, M. (2006). Peekaboom: A game for locating objects in images. In *CHI'06: SIGCHI Conference on Human Factors in Computing Systems*, pp. 55–64, Montréal, Québec, Canada.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305.

Wainwright, M. J., Jaakkola, T. S., and Willsky, A. S. (2005). MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717.

Waithe, P. and Ferrie, F. (1991). From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1038–1049.

Walker, E. L. and Herman, M. (1988). Geometric reasoning for constructing 3D scene descriptions from images. *Artificial Intelligence*, 37:275–290.

Wallace, G. K. (1991). The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44.

Wallace, J. R., Cohen, M. F., and Greenberg, D. P. (1987). A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *Computer Graphics (SIGGRAPH '87)*, 21(4):311–320.

Waltz, D. L. (1975). Understanding line drawings of scenes with shadows. In Winston, P. H. (ed.), *The Psychology of Computer Vision*, McGraw-Hill, New York.

Wang, H. and Oliensis, J. (2010). Shape matching by segmentation averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):619–635.

Wang, J. and Cohen, M. F. (2005). An iterative optimization approach for unified image segmentation and matting. In *Tenth International Conference on Computer Vision (ICCV 2005)*, Beijing, China.

Wang, J. and Cohen, M. F. (2007a). Image and video matting: A survey. *Foundations and Trends in Computer Graphics and Computer Vision*, 3(2).

Wang, J. and Cohen, M. F. (2007b). Optimized color sampling for robust matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Wang, J. and Cohen, M. F. (2007c). Simultaneous matting and compositing. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Wang, J., Agrawala, M., and Cohen, M. F. (2007). Soft scissors: An interactive tool for realtime high quality matting. *ACM Transactions on Graphics*, 26(3).

Wang, J., Thiesson, B., Xu, Y., and Cohen, M. (2004). Image and video segmentation by anisotropic kernel mean shift. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 238–249, Prague.

Wang, J., Bhat, P., Colburn, R. A., Agrawala, M., and Cohen, M. F. (2005). Video cutout. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):585–594.

Wang, J. Y. A. and Adelson, E. H. (1994). Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638.

Wang, L., Kang, S. B., Szeliski, R., and Shum, H.-Y. (2001). Optimal texture map reconstruction from multiple views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, pp. 347–354, Kauai, Hawaii.

Wang, Y. and Zhu, S.-C. (2003). Modeling textured motion: Particle, wave and sketch. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 213–220, Nice, France.

Wang, Z., Bovik, A. C., and Simoncelli, E. P. (2005). Structural approaches to image quality assessment. In Bovik, A. C. (ed.), *Handbook of Image and Video Processing*, pp. 961–974, Elsevier Academic Press.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Wang, Z.-F. and Zheng, Z.-G. (2008). A region based stereo matching algorithm using cooperative optimization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Ward, G. (1992). Measuring and modeling anisotropic reflection. *Computer Graphics (SIGGRAPH '92)*, 26(4):265–272.

Ward, G. (1994). The radiance lighting simulation and rendering system. In *ACM SIGGRAPH 1994 Conference Proceedings*, pp. 459–472.

Ward, G. (2003). Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *Journal of Graphics Tools*, 8(2):17–30.

Ward, G. (2004). High dynamic range image encodings. http://www.anyhere.com/gward/hdrenc/hdr_encodings.html.

Ware, C., Arthur, K., and Booth, K. S. (1993). Fish tank virtual reality. In *INTERCHI'03*, pp. 37–42, Amsterdam.

Warren, J. and Weimer, H. (2001). *Subdivision Methods for Geometric Design: A Constructive Approach*. Morgan Kaufmann.

Watanabe, M. and Nayar, S. K. (1998). Rational filters for passive depth from defocus. *International Journal of Computer Vision*, 27(3):203–225.

Watt, A. (1995). *3D Computer Graphics*. Addison-Wesley, Harlow, England, third edition.

Weber, J. and Malik, J. (1995). Robust computation of optical flow in a multi-scale differential framework. *International Journal of Computer Vision*, 14(1):67–81.

Weber, M., Welling, M., and Perona, P. (2000). Unsupervised learning of models for recognition. In *Sixth European Conference on Computer Vision (ECCV 2000)*, pp. 18–32, Dublin, Ireland.

Wedel, A., Cremers, D., Pock, T., and Bischof, H. (2009). Structure- and motion-adaptive regularization for high accuracy optic flow. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., and Cremers, D. (2008). Efficient dense scene flow from sparse or dense stereo data. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 739–751, Marseilles.

Wei, C. Y. and Quan, L. (2004). Region-based progressive stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, pp. 106–113, Washington, D. C.

Wei, L.-Y. and Levoy, M. (2000). Fast texture synthesis using tree-structured vector quantization. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 479–488.

Weickert, J. (1998). *Anisotropic Diffusion in Image Processing*. Tuebner, Stuttgart.

Weickert, J., ter Haar Romeny, B. M., and Viergever, M. A. (1998). Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410.

Weinland, D., Ronfard, R., and Boyer, E. (2006). Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257.

Weiss, Y. (1997). Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 520–526, San Juan, Puerto Rico.

Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. In *Seventh International Conference on Computer Vision (ICCV'99)*, pp. 975–982, Kerkyra, Greece.

Weiss, Y. (2001). Deriving intrinsic images from image sequences. In *Eighth International Conference on Computer Vision (ICCV 2001)*, pp. 7–14, Vancouver, Canada.

Weiss, Y. and Adelson, E. H. (1996). A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pp. 321–326, San Francisco.

Weiss, Y. and Freeman, B. (2007). What makes a good model of natural images? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Weiss, Y. and Freeman, W. T. (2001a). Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200.

Weiss, Y. and Freeman, W. T. (2001b). On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744.

Weiss, Y., Torralba, A., and Fergus, R. (2008). Spectral hashing. In *Advances in Neural Information Processing Systems*.

Weiss, Y., Yanover, C., and Meltzer, T. (2010). Linear programming and variants of belief propagation. In Blake, A., Kohli, P., and Rother, C. (eds), *Advances in Markov Random Fields*, MIT Press.

Wells, III, W. M. (1986). Efficient synthesis of Gaussian filters by cascaded uniform filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):234–239.

Weng, J., Ahuja, N., and Huang, T. S. (1993). Optimal motion and structure estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):864–884.

Wenger, A., Gardner, A., Tchou, C., Unger, J., Hawkins, T., and Debevec, P. (2005). Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):756–764.

Werlberger, M., Trobin, W., Pock, T., Bischof, H., Wedel, A., and Cremers, D. (2009). Anisotropic Huber-L1 optical flow. In *British Machine Vision Conference (BMVC 2009)*, London.

Werner, T. (2007). A linear programming approach to max-sum problem: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1165–1179.

Werner, T. and Zisserman, A. (2002). New techniques for automated architectural reconstruction from photographs. In *Seventh European Conference on Computer Vision (ECCV 2002)*, pp. 541–555, Copenhagen.

Westin, S. H., Arvo, J. R., and Torrance, K. E. (1992). Predicting reflectance functions from complex surfaces. *Computer Graphics (SIGGRAPH '92)*, 26(4):255–264.

Westover, L. (1989). Interactive volume rendering. In *Workshop on Volume Visualization*, pp. 9–16, Chapel Hill.

Wexler, Y., Fitzgibbon, A., and Zisserman, A. (2002). Bayesian estimation of layers from multiple images. In *Seventh European Conference on Computer Vision (ECCV 2002)*, pp. 487–501, Copenhagen.

Wexler, Y., Shechtman, E., and Irani, M. (2007). Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476.

Weyrich, T., Lawrence, J., Lensch, H. P. A., Rusinkiewicz, S., and Zickler, T. (2008). Principles of appearance acquisition and representation. *Foundations and Trends in Computer Graphics and Computer Vision*, 4(2):75–191.

Weyrich, T., Matusik, W., Pfister, H., Bickel, B., Donner, C. *et al*. (2006). Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics*, 25(3):1013–1024.

Wheeler, M. D., Sato, Y., and Ikeuchi, K. (1998). Consensus surfaces for modeling 3D objects from multiple range images. In *Sixth International Conference on Computer Vision (ICCV'98)*, pp. 917–924, Bombay.

White, R. and Forsyth, D. (2006). Combining cues: Shape from shading and texture. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 1809–1816, New York City, NY.

White, R., Crane, K., and Forsyth, D. A. (2007). Capturing and animating occluded cloth. *ACM Transactions on Graphics*, 26(3).

Wiejak, J. S., Buxton, H., and Buxton, B. F. (1985). Convolution with separable masks for early image processing. *Computer Vision, Graphics, and Image Processing*, 32(3):279–290.

Wilburn, B., Joshi, N., Vaish, V., Talvala, E.-V., Antunez, E. *et al*. (2005). High performance imaging using large camera arrays. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):765–776.

Wilczkowiak, M., Brostow, G. J., Tordoff, B., and Cipolla, R. (2005). Hole filling through photomontage. In *British Machine Vision Conference (BMVC 2005)*, pp. 492–501, Oxford Brookes.

Williams, D. and Burns, P. D. (2001). Diagnostics for digital capture using MTF. In *IS&T PICS Conference*, pp. 227–232.

Williams, D. J. and Shah, M. (1992). A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics, and Image Processing*, 55(1):14–26.

Williams, L. (1983). Pyramidal parametrics. *Computer Graphics (SIGGRAPH '83)*, 17(3):1–11.

Williams, L. (1990). Performace driven facial animation. *Computer Graphics (SIGGRAPH '90)*, 24(4):235–242.

Williams, O., Blake, A., and Cipolla, R. (2003). A sparse probabilistic learning algorithm for real-time tracking. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 353–360, Nice, France.

Williams, T. L. (1999). *The Optical Transfer Function of Imaging Systems*. Institute of Physics Publishing, London.

Winder, S. and Brown, M. (2007). Learning local image descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Winkenbach, G. and Salesin, D. H. (1994). Computer-generated pen-and-ink illustration. In *ACM SIGGRAPH 1994 Conference Proceedings*, pp. 91–100, Orlando, Florida.

Winn, J. and Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 37–44, New York City, NY.

Winnemöller, H., Olsen, S. C., and Gooch, B. (2006). Real-time video abstraction. *ACM Transactions on Graphics*, 25(3):1221–1226.

Winston, P. H. (ed.). (1975). *The Psychology of Computer Vision*, McGraw-Hill, New York.

Wiskott, L., Fellous, J.-M., Krüger, N., and von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779.

Witkin, A. (1981). Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17(1-3):17–45.

Witkin, A. (1983). Scale-space filtering. In *Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, pp. 1019–1022.

Witkin, A., Terzopoulos, D., and Kass, M. (1986). Signal matching through scale space. In *Fifth National Conference on Artificial Intelligence (AAAI-86)*, pp. 714–719, Philadelphia.

Witkin, A., Terzopoulos, D., and Kass, M. (1987). Signal matching through scale space. *International Journal of Computer Vision*, 1:133–144.

Wolberg, G. (1990). *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos.

Wolberg, G. and Pavlidis, T. (1985). Restoration of binary images using stochastic relaxation with annealing. *Pattern Recognition Letters*, 3:375–388.

Wolff, L. B., Shafer, S. A., and Healey, G. E. (eds). (1992a). *Radiometry*. *Physics-Based Vision: Principles and Practice*, Jones & Bartlett, Cambridge, MA.

Wolff, L. B., Shafer, S. A., and Healey, G. E. (eds). (1992b). *Shape Recovery*. *Physics-Based Vision: Principles and Practice*, Jones & Bartlett, Cambridge, MA.

Wood, D. N., Finkelstein, A., Hughes, J. F., Thayer, C. E., and Salesin, D. H. (1997). Multiperspective panoramas for cel animation. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 243–250, Los Angeles.

Wood, D. N., Azuma, D. I., Aldinger, K., Curless, B., Duchamp, T., Salesin, D. H., and Stuetzle, W. (2000). Surface light fields for 3D photography. In *ACM SIGGRAPH 2000 Conference Proceedings*, pp. 287–296.

Woodford, O., Reid, I., Torr, P. H., and Fitzgibbon, A. (2008). Global stereo reconstruction under second order smoothness priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Woodham, R. J. (1981). Analysing images of curved surfaces. *Artificial Intelligence*, 17:117–140.

Woodham, R. J. (1994). Gradient and curvature from photometric stereo including local confidence estimation. *Journal of the Optical Society of America, A*, 11:3050–3068.

Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785.

Wright, S. (2006). *Digital Compositing for Film and Video*. Focal Press, 2nd edition.

Wu, C. (2010). SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). http://www.cs.unc.edu/~ccwu/siftgpu/.

Wyszecki, G. and Stiles, W. S. (2000). *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, New York, 2nd edition.

Xiao, J. and Shah, M. (2003). Two-frame wide baseline matching. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 603–609, Nice, France.

Xiao, J. and Shah, M. (2005). Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1644–1659.

Xiong, Y. and Turkowski, K. (1997). Creating image-based VR using a self-calibrating fisheye lens. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 237–243, San Juan, Puerto Rico.

Xiong, Y. and Turkowski, K. (1998). Registration, calibration and blending in creating high quality panoramas. In *IEEE Workshop on Applications of Computer Vision (WACV'98)*, pp. 69–74, Princeton.

Xu, L., Chen, J., and Jia, J. (2008). A segmentation based variational model for accurate optical flow estimation. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 671–684, Marseilles.

Yang, D., El Gamal, A., Fowler, B., and Tian, H. (1999). A 640x512 CMOS image sensor with ultra-wide dynamic range floating-point pixel level ADC. *IEEE Journal of Solid State Circuits*, 34(12):1821–1834.

Yang, L. and Albregtsen, F. (1996). Fast and exact computation of Cartesian geometric moments using discrete Green's theorem. *Pattern Recognition*, 29(7):1061–1073.

Yang, L., Meer, P., and Foran, D. (2007). Multiple class segmentation using a unified framework over mean-shift patches. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Yang, L., Jin, R., Sukthankar, R., and Jurie, F. (2008). Unifying discriminative visual codebook generation with classifier training for object category recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Yang, M.-H., Ahuja, N., and Tabb, M. (2002). Extraction of 2D motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1061–1074.

Yang, M.-H., Kriegman, D. J., and Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58.

Yang, Q., Wang, L., Yang, R., Stewénius, H., and Nistér, D. (2009). Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):492–504.

Yang, Y., Yuille, A., and Lu, J. (1993). Local, global, and multilevel stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pp. 274–279, New York.

Yanover, C., Meltzer, T., and Weiss, Y. (2006). Linear programming relaxations and belief propagation — an empirical study. *Journal of Machine Learning Research*, 7:1887–1907.

Yao, B. Z., Yang, X., Lin, L., Lee, M. W., and Zhu, S.-C. (2010). I2T: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508.

Yaou, M.-H. and Chang, W.-T. (1994). Fast surface interpolation using multiresolution wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7):673–689.

Yatziv, L. and Sapiro, G. (2006). Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing*, 15(5):1120–1129.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2001). Understanding belief propagation and its generalization. In *International Joint Conference on Artificial Intelligence (IJCAI 2001)*.

Yezzi, Jr., A. J., Kichenassamy, S., Kumar, A., Olver, P., and Tannenbaum, A. (1997). A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16(2):199–209.

Yilmaz, A. and Shah, M. (2006). Matching actions in presence of camera motion. *Computer Vision and Image Understanding*, 104(2-3):221–231.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys*, 38(4).

Yin, P., Criminisi, A., Winn, J., and Essa, I. (2007). Tree-based classifiers for bilayer video segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Yoon, K.-J. and Kweon, I.-S. (2006). Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656.

Yserentant, H. (1986). On the multi-level splitting of finite element spaces. *Numerische Mathematik*, 49:379–412.

Yu, S. X. and Shi, J. (2003). Multiclass spectral clustering. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 313–319, Nice, France.

Yu, Y. and Malik, J. (1998). Recovering photometric properties of architectural scenes from photographs. In *ACM SIGGRAPH 1996 Conference Proceedings*, pp. 207–218, Orlando.

Yu, Y., Debevec, P., Malik, J., and Hawkins, T. (1999). Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *ACM SIGGRAPH 1999 Conference Proceedings*, pp. 215–224.

Yuan, L., Sun, J., Quan, L., and Shum, H.-Y. (2007). Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics*, 26(3).

Yuan, L., Sun, J., Quan, L., and Shum, H.-Y. (2008). Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Transactions on Graphics*, 27(3).

Yuan, L., Wen, F., Liu, C., and Shum, H.-Y. (2004). Synthesizing dynamic texture with closed-loop linear dynamic system. In *Eighth European Conference on Computer Vision (ECCV 2004)*, pp. 603–616, Prague.

Yuille, A. (1991). Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1):59–70.

Yuille, A. (2002). CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722.

Yuille, A. (2010). Loopy belief propagation, mean-field and Bethe approximations. In Blake, A., Kohli, P., and Rother, C. (eds), *Advances in Markov Random Fields*, MIT Press.

Yuille, A. and Poggio, T. (1984). *A Generalized Ordering Constraint for Stereo Correspondence*. A. I. Memo 777, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Yuille, A., Vincent, L., and Geiger, D. (1992). Statistical morphology and Bayesian reconstruction. *Journal of Mathematical Imaging and Vision*, 1(3):223–238.

Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Third European Conference on Computer Vision (ECCV'94)*, pp. 151–158, Stockholm, Sweden.

Zach, C. (2008). Fast and high quality fusion of depth maps. In *Fourth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'08)*, Atlanta.

Zach, C., Gallup, D., and Frahm, J.-M. (2008). Fast gain-adaptive KLT tracking on the GPU. In *CVPR 2008 Workshop on Visual Computer Vision on GPUs (CVGPU)*, Anchorage, AK.

Zach, C., Klopschitz, M., and Pollefeys, M. (2010). Disambiguating visual relations using loop constraints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA.

Zach, C., Pock, T., and Bischof, H. (2007a). A duality based approach for realtime TV-L1 optical flow. In *Pattern Recognition (DAGM 2007)*.

Zach, C., Pock, T., and Bischof, H. (2007b). A globally optimal algorithm for robust TV-L^1 range image integration. In *Eleventh International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil.

Zanella, V. and Fuentes, O. (2004). An approach to automatic morphing of face images in frontal view. In *Mexican International Conference on Artificial Intelligence (MICAI 2004)*, pp. 679–687, Mexico City.

Zebedin, L., Bauer, J., Karner, K., and Bischof, H. (2008). Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *Tenth European Conference on Computer Vision (ECCV 2008)*, pp. 873–886, Marseilles.

Zelnik-Manor, L. and Perona, P. (2007). Automating joiners. In *Symposium on Non Photorealistic Animation and Rendering*, Annecy.

Zhang, G., Jia, J., Wong, T.-T., and Bao, H. (2008). Recovering consistent video depth maps via bundle optimization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Zhang, J., McMillan, L., and Yu, J. (2006). Robust tracking and stereo matching under variable illumination. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 871–878, New York City, NY.

Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. (2007). Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73(2):213–238.

Zhang, L., Curless, B., and Seitz, S. (2003). Spacetime stereo: Shape recovery for dynamic scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, pp. 367–374, Madison, WI.

Zhang, L., Dugas-Phocion, G., Samson, J.-S., and Seitz, S. M. (2002). Single view modeling of free-form scenes. *Journal of Visualization and Computer Animation*, 13(4):225–235.

Zhang, L., Snavely, N., Curless, B., and Seitz, S. M. (2004). Spacetime faces: High resolution capture for modeling and animation. *ACM Transactions on Graphics*, 23(3):548–558.

Zhang, R., Tsai, P.-S., Cryer, J. E., and Shah, M. (1999). Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706.

Zhang, Y. and Kambhamettu, C. (2003). On 3D scene flow and structure recovery from multiview image sequences. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(4):592–606.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152.

Zhang, Z. (1998a). Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195.

Zhang, Z. (1998b). On the optimization criteria used in two-view motion analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):717–729.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.

Zhang, Z. and He, L.-W. (2007). Whiteboard scanning and image enhancement. *Digital Signal Processing*, 17(2):414–432.

Zhang, Z. and Shan, Y. (2000). A progressive scheme for stereo matching. In *Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE 2000)*, pp. 68–85, Dublin, Ireland.

Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119.

Zhao, G. and Pietikäinen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928.

Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–358.

Zheng, J. Y. (1994). Acquiring 3-D models from sequences of contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):163–178.

Zheng, K. C., Kang, S. B., Cohen, M., and Szeliski, R. (2007). Layered depth panoramas. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, MN.

Zheng, Y., Lin, S., and Kang, S. B. (2006). Single-image vignetting correction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, pp. 461–468, New York City, NY.

Zheng, Y., Yu, J., Kang, S.-B., Lin, S., and Kambhamettu, C. (2008). Single-image vignetting correction using radial gradient symmetry. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK.

Zheng, Y., Zhou, X. S., Georgescu, B., Zhou, S. K., and Comaniciu, D. (2006). Example based non-rigid shape detection. In *Ninth European Conference on Computer Vision (ECCV 2006)*, pp. 423–436.

Zheng, Y.-T., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Chua, T.-S., and Neven, H. (2009). Tour the world: building a web-scale landmark recognition engine. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami Beach, FL.

Zhong, J. and Sclaroff, S. (2003). Segmenting foreground objects from a dynamic, textured background via a robust Kalman filter. In *Ninth International Conference on Computer Vision (ICCV 2003)*, pp. 44–50, Nice, France.

Zhou, C., Lin, S., and Nayar, S. (2009). Coded aperture pairs for depth from defocus. In *Twelfth International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Zhou, Y., Gu, L., and Zhang, H.-J. (2003). Bayesian tangent shape model: Estimating shape and pose parameters via Bayesian inference. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2003)*, pp. 109–116, Madison, WI.

Zhu, L., Chen, Y., Lin, Y., Lin, C., and Yuille, A. (2008). Recursive segmentation and recognition templates for 2D parsing. In *Advances in Neural Information Processing Systems*.

Zhu, S.-C. and Mumford, D. (2006). A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Computer Vision*, 2(4).

Zhu, S. C. and Yuille, A. L. (1996). Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900.

Zhu, Z. and Kanade, T. (2008). Modeling and representations of large-scale 3D scenes. *International Journal of Computer Vision*, 78(2-3):119–120.

Zisserman, A., Giblin, P. J., and Blake, A. (1989). The information available to a moving observer from specularities. *Image and Vision Computing*, 7(1):38–42.

Zitnick, C. L. and Kanade, T. (2000). A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684.

Zitnick, C. L. and Kang, S. B. (2007). Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75(1):49–65.

Zitnick, C. L., Jojic, N., and Kang, S. B. (2005). Consistent segmentation for optical flow estimation. In *Tenth International Conference on Computer Vision (ICCV 2005)*, pp. 1308–1315, Beijing, China.

Zitnick, C. L., Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R. (2004). High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 23(3):600–608.

Zitov'aa, B. and Flusser, J. (2003). Image registration methods: A survey. *Image and Vision Computing*, 21:997–1000.

Zoghlami, I., Faugeras, O., and Deriche, R. (1997). Using geometric corners to build a 2D mosaic from a set of images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 420–425, San Juan, Puerto Rico.

Zongker, D. E., Werner, D. M., Curless, B., and Salesin, D. H. (1999). Environment matting and compositing. In *ACM SIGGRAPH 1999 Conference Proceedings*, pp. 205–214.

Zorin, D., Schröder, P., and Sweldens, W. (1996). Interpolating subdivision for meshes with arbitrary topology. In *ACM SIGGRAPH 1997 Conference Proceedings*, pp. 189–192, New Orleans.

Index

3D Rotations, *see* Rotations
3D alignment, 283
    absolute orientation, 283, 515
    orthogonal Procrustes, 283
3D photography, 537
3D video, 564

Absolute orientation, 283, 515
Active appearance model (AAM), 598
Active contours, 238
Active illumination, 512
Active rangefinding, 512
Active shape model (ASM), 243, 598
Activity recognition, 534
Adaptive smoothing, 111
Affine transforms, 34, 37
Affinities (segmentation), 260
    normalizing, 262
Algebraic multigrid, 254
Algorithms
    testing, viii
Aliasing, 69, 417
Alignment, *see* Image alignment
Alpha
    opacity, 93
    pre-multiplied, 93
Alpha matte, 93
Ambient illumination, 58
Analog to digital conversion (ADC), 68
Anisotropic diffusion, 111
Anisotropic filtering, 148
Anti-aliasing filter, 70, 417
Aperture, 62
Aperture problem, 347
Applications, 5
    3D model reconstruction, 319, 327
    3D photography, 537
    augmented reality, 287, 325
    automotive safety, 5
    background replacement, 489
    biometrics, 588
    colorization, 442
    de-interlacing, 364
    digital heritage, 517
    document scanning, 379
    edge editing, 219
    facial animation, 528
    flash photography, 434
    frame interpolation, 368
    gaze correction, 483
    head tracking, 483
    hole filling, 457
    image restoration, 169
    image search, 630
    industrial, 5
    intelligent photo editing, 621
    Internet photos, 327
    location recognition, 609
    machine inspection, 5
    match move, 324
    medical imaging, 5, 268, 358
    morphing, 152
    mosaic-based video compression, 383
    non-photorealistic rendering, 458
    Optical character recognition (OCR), 5
    panography, 277
    performance-driven animation, 209
    photo pop-up, 623

Photo Tourism, 548
Photomontage, 403
planar pattern tracking, 287
rotoscoping, 249
scene completion, 621
scratch removal, 457
single view reconstruction, 292
tonal adjustment, 97
video denoising, 364
video stabilization, 354
video summarization, 383
video-based walkthroughs, 566
VideoMouse, 288
view morphing, 315
visual effects, 5
whiteboard scanning, 379
z-keying, 489
Arc length parameterization of a curve, 217
Architectural reconstruction, 524
Area statistics, 115
mean (centroid), 115
perimeter, 115
second moment (inertia), 115
Aspect ratio, 47, 48
Augmented reality, 287, 298, 325
Auto-calibration, 313
Automatic gain control (AGC), 67
Axis/angle representation of rotations, 37

B-snake, 241
B-spline, 151, 152, 220, 241, 246, 359
cubic, 128
multilevel, 518
octree, 523
Background plate, 454
Background subtraction (maintenance), 531
Bag of words (keypoints), 612, 639
distance metrics, 613
Band-pass filter, 104
Bartlett filter, *see* Bilinear kernel
Bayer pattern (RGB sensor mosaic), 76
demosaicing, 76, 440
Bayes' rule, 124, 159, 667
MAP (maximum a posteriori) estimate, 668
posterior distribution, 667
Bayesian modeling, 158, 667
MAP estimate, 159, 668
matting, 449
posterior distribution, 159, 667
prior distribution, 159, 667
uncertainty, 159
Belief propagation (BP), 163, 672
update rule, 673
Bias, 91, 339
Bidirectional Reflectance Distribution Function, *see* BRDF
Bilateral filter, 110
joint, 435
range kernel, 110
tone mapping, 428
Bilinear blending, 97
Bilinear kernel, 103
Biometrics, 588
Bipartite problem, 322
Blind image deconvolution, 437
Block-based motion estimation (block matching), 341
Blocks world, 10
Blue screen matting, 94, 171, 445
Blur kernel, 62
estimation, 416, 463
Blur removal, 126, 174
Body color, 57
Boltzmann distribution, 159, 668
Boosting, 582
AdaBoost algorithm, 584
decision stump, 582
weak learner, 582
Border (boundary) effects, 101, 173
Boundary detection, 215
Box filter, 103
Boxlet, 107
BRDF, 55
anisotropic, 56
isotropic, 56
recovery, 536
spatially varying (SVBRDF), 536
Brightness, 91
Brightness constancy, 3, 338
Brightness constancy constraint, 338, 345, 360
Bundle adjustment, 320

Calibration, *see* Camera calibration
Calibration matrix, 46
Camera calibration, 45, 86
  accuracy, 299
  aliasing, 417
  extrinsic, 46, 284
  intrinsic, 45, 288
  optical blur, 416, 463
  patterns, 289
  photometric, 412
  plumb-line method, 295, 300
  point spread function, 416, 463
  radial distortion, 295
  radiometric, 412, 421, 461
  rotational motion, 293, 298
  slant edge, 417
  vanishing points, 290
  vignetting, 416
Camera matrix, 46, 49
Catadioptric optics, 64
Category-level recognition, 611
  bag of words, 612, 639
  data sets, 631
  part-based, 615
  segmentation, 620
  surveys, 635
CCD, 65
  blooming, 65
Central difference, 104
Chained transformations, 287, 321
Chamfer matching, 113
Characteristic function, 115, 248, 516, 522
Characteristic polynomial, 649
Chirality, 306, 310
Cholesky factorization, 650
  algorithm, 650
  incomplete, 659
  sparse, 657
Chromatic aberration, 63, 301
Chromaticity coordinates, 73
CIE L*a*b*, *see* Color
CIE L*u*v*, *see* Color
CIE XYZ, *see* Color
Circle of confusion, 62
CLAHE, *see* Histogram equalization
Clustering
  agglomerative, 251
  cluster analysis, 237, 268
  divisive, 251
CMOS, 66
Co-vector, 34
Coefficient matrix, 156
Collineation, 37
Color, 71
  balance, 76, 85, 171
  camera, 75
  demosaicing, 76, 440
  fringing, 442
  hue, saturation, value (HSV), 79
  L*a*b*, 74
  L*u*v*, 74, 254
  primaries, 71
  profile, 414
  ratios, 79
  RGB, 72
  transform, 92
  twist, 76, 92
  XYZ, 72
  YIQ, 78
  YUV, 78
Color filter array (CFA), 76, 440
Color line model, 450
ColorChecker chart, 414
Colorization, 442
Compositing, 92, 169, 171
  image stitching, 396
  opacity, 93
  over operator, 93
  surface, 396
  transparency, 93
Compression, 80
Computational photography, 409
  active illumination, 436
  flash and non-flash, 434
  high dynamic range, 419
  references, 411, 460
  tone mapping, 427
Concentric mosaic, 384, 556
CONDENSATION, 246
Condition number, 657

Conditional random field (CRF), 165, 484, 621
Confusion matrix (table), 201
Conic section, 31
Conjugate gradient descent (CG), 657
  algorithm, 658
  non-linear, 658
  preconditioned, 659
Connected components, 115, 174
Constellation model, 618
Content based image retrieval (CBIR), 630
Continuation method, 158
Contour
  arc length parameterization, 217
  chain code, 217
  matching, 218, 231
  smoothing, 218
Contrast, 91
Controlled-continuity spline, 155
Convolution, 100
  kernel, 98
  mask, 98
  superposition, 100
Coring, 134, 177
Correlation, 98, 340
  windowed, 342
Correspondence map, 350
Cramer–Rao lower bound, 282, 349, 678
Cube map
  Hough transform, 223
  image stitching, 396
Curve
  arc length parameterization, 217
  evolution, 218
  matching, 218
  smoothing, 218
Cylindrical coordinates, 385

Data energy (term), 159, 668
Data sets and test databases, 680
  recognition, 631
De-interlacing, 364
Decimation, 130
Decimation kernels
  bicubic, 131
  binomial, 130, 132
  QMF, 131
  windowed sinc, 130
Demosaicing (Bayer), 76, 440
Depth from defocus, 511
Depth map, *see* Disparity map
Depth of field, 62, 83
Di-chromatic reflection model, 60
Difference matting (keying), 94, 172, 446, 531
Difference of Gaussians (DoG), 135
Difference of low-pass (DOLP), 135
Diffuse reflection, 57
Diffusion
  anisotropic, 111
Digital camera, 65
  color, 75
  color filter array (CFA), 76
  compression, 80
Direct current (DC), 81
Direct linear transform (DLT), 284
Direct sparse matrix techniques, 655
Directional derivative, 105
  selectivity, 106
Discrete cosine transform (DCT), 81, 125
Discrete Fourier transform (DFT), 118
Discriminative random field (DRF), 167
Disparity, 45, 473
Disparity map, 473, 492
  multiple, 491
Disparity space image (DSI), 473
  generalized, 475
Displaced frame difference (DFD), 338
Displacement field, 150
Distance from face space (DFFS), 590
Distance in face space (DIFS), 590
Distance map, *see* Distance transform
Distance transform, 113, 174
  Euclidean, 114
  image stitching, 398
  Manhattan (city block), 113
  signed, 114
Domain (of a function), 91
Domain scaling law, 147
Downsampling, *see* Decimation
Dynamic programming (DP), 485, 670
  monotonicity, 487
  ordering constraint, 487

scanline optimization, 487
Dynamic snake, 243
Dynamic texture, 563

Earth mover's distance (EMD), 613
Edge detection, 210, 230
boundary detection, 215
Canny, 211
chain code, 217
color, 214
Difference of Gaussian, 212
edgel (edge element), 212
hysteresis, 217
Laplacian of Gaussian, 212
linking, 215, 231
marching cubes, 213
scale selection, 213
steerable filter, 213
zero crossing, 212
Eigenface, 589
Eigenvalue decomposition, 242, 589, 647
Eigenvector, 647
Elastic deformations, 358
image registration, 358
Elastic nets, 239
Elliptical weighted average (EWA), 148
Environment map, 55, 555
Environment matte, 556
Epanechnikov kernel, 259
Epipolar constraint, 307
Epipolar geometry, 307, 471
pure rotation, 311
pure translation, 311
Epipolar line, 471
Epipolar plane, 471, 477
image (EPI), 490, 552
Epipolar volume, 552
Epipole, 308, 471
Error rates
accuracy (ACC), 202
false negative (FN), 201
false positive (FP), 201
positive predictive value (PPV), 202
precision, 202
recall, 202
ROC curve, 202
true negative (TN), 201
true positive (TP), 201
Errors-in-variable model, 389, 653
heteroscedastic, 654
Essential matrix, 308
5-point algorithm, 310
eight-point algorithm, 308
re-normalization, 309
seven-point algorithm, 309
twisted pair, 310
Estimation theory, 662
Euclidean transformation, 33, 36
Euler angles, 37
Expectation maximization (EM), 256
Exponential twist, 39
Exposure bracketing, 421
Exposure value (EV), 62, 412

F-number (stop), 62, 84
Face detection, 578
boosting, 582
cascade of classifiers, 583
clustering and PCA, 580
data sets, 631
neural networks, 580
support vector machines, 582
Face modeling, 526
Face recognition, 588
active appearance model, 598
data sets, 631
eigenface, 589
elastic bunch graph matching, 596
local binary patterns (LBP), 635
local feature analysis, 596
Face transfer, 561
Facial motion capture, 528, 530, 561
Factor graph, 160, 669, 672
Factorization, 14, 315
missing data, 318
projective, 318
Fast Fourier transform (FFT), 118
Fast marching method (FMM), 248
Feature descriptor, 196, 229
bias and gain normalization, 196
GLOH, 198
patch, 196

PCA-SIFT, 197
performance (evaluation), 198
quantization, 207, 607, 612
SIFT, 197
steerable filter, 198
Feature detection, 183, 185, 228
Adaptive non-maximal suppression, 189
affine invariance, 194
auto-correlation, 185
Förstner, 188
Harris, 188
Laplacian of Gaussian, 191
MSER, 195
region, 195
repeatability, 190
rotation invariance, 193
scale invariance, 191
Feature matching, 183, 200, 229
densification, 207
efficiency, 205
error rates, 201
hashing, 205
indexing structure, 205
k-d trees, 206
locality sensitive hashing, 205
nearest neighbor, 203
strategy, 200
verification, 207
Feature tracking, 207, 230
affine, 208
learning, 209
Feature tracks, 315, 327
Feature-based alignment, 275
2D, 275
3D, 283
iterative, 278
Jacobian, 276
least squares, 275
match verification, 603
RANSAC, 281
robust, 281
Field of Experts (FoE), 163
Fill factor, 67
Fill-in, 322, 656
Filter
adaptive, 111
band-pass, 104
bilateral, 110
directional derivative, 105
edge-preserving, 109, 111
Laplacian of Gaussian, 104
median, 108
moving average, 103
non-linear, 108
separable, 102, 173
steerable, 105, 174
Filter coefficients, 98
Filter kernel, *see* Kernel
Finding faces, *see* Face detection
Finite element analysis, 155
stiffness matrix, 156
Finite impulse response (FIR) filter, 98, 107
Fisher information matrix, 277, 282, 664, 678
Fisher's linear discriminant (FLD), 593
Fisheye lens, 53
Flash and non-flash merging, 434
Flash matting, 454
Flip-book animation, 296
Flying spot scanner, 514
Focal length, 47, 48, 61
Focus, 62
shape-from, 511, 539
Focus of expansion (FOE), 311
Form factor, 60
Forward mapping, *see* Forward warping
Forward warping, 145, 177
Fourier transform, 116, 174
discrete, 118
examples, 119
magnitude (gain), 117
pairs, 119
Parseval's Theorem, 119
phase (shift), 117
power spectrum, 123
properties, 118
two-dimensional, 123
Fourier-based motion estimation, 341
rotations and scale, 344
Frame interpolation, 368
Free-viewpoint video, 564

Fundamental matrix, 312
estimation, *see* Essential matrix
Fundamental radiometric relation, 65

Gain, 91, 339
Gamma, 92
Gamma correction, 77, 85
Gap closing (image stitching), 382
Garbage matte, 454
Gaussian kernel, 103
Gaussian Markov random field (GMRF), 163, 168, 438
Gaussian mixtures, *see* Mixture of Gaussians
Gaussian pyramid, 132
Gaussian scale mixtures (GSM), 163
Gaze correction, 483
Geman–McClure function, 338
Generalized cylinders, 11, 515, 519
Geodesic active contour, 248
Geodesic distance (segmentation), 267
Geometric image formation, 29
Geometric lens aberrations, 63
Geometric primitives, 29
homogeneous coordinates, 30
lines, 30, 31
normal vector, 30
normal vectors, 31
planes, 31
points, 30, 31
Geometric transformations
2D, 33, 145
3D, 36
3D perspective, 37
3D rotations, 37
affine, 34, 37
bilinear, 35
calibration matrix, 46
collineation, 37
Euclidean, 33, 36
forward warping, 145, 177
hierarchy, 34
homography, 34, 37, 50, 379
inverse warping, 146
perspective, 34
projections, 42
projective, 34
rigid body, 33, 36
scaled rotation, 34, 37
similarity, 34, 37
translation, 33, 36
Geometry image, 520
Gesture recognition, 530
Gibbs distribution, 159, 668
Gibbs sampler, 670
Gimbal lock, 37
Gist (of a scene), 623, 626
Global illumination, 60
Global optimization, 153
GPU algorithms, 688
Gradient location-orientation histogram (GLOH), 198
Graduated non-convexity (GNC), 158
Graph cuts
MRF inference, 161, 674
normalized cuts, 260
Graph-based segmentation, 252
Grassfire transform, 114, 219, 398
Ground control points, 309, 377

Hammersley–Clifford theorem, 159, 668
Hann window, 121
Harris corner detector, *see* Feature detection
Head tracking, 483
active appearance model (AAM), 598
Helmholtz reciprocity, 56
Hessian, 156, 189, 277, 279, 282, 346, 350, 652
eigenvalues, 349
image, 346, 361
inverse, 282, 349, 352
local, 360
patch-based, 351
rank-deficient, 326
reduced motion, 322
sparse, 322, 334, 655
Heteroscedastic, 277, 654
Hidden Markov model (HMM), 563
Hierarchical motion estimation, 341
High dynamic range (HDR) imaging, 419
formats, 426
tone mapping, 427
Highest confidence first, 670
Highest confidence first (HCF), 161
Hilbert transform pair, 106
Histogram equalization, 94, 172

locally adaptive, 96, 172
Histogram intersection, 613
Histogram of oriented gradients (HOG), 585
History of computer vision, 10
Hole filling, 457
Homogeneous coordinates, 30, 306
Homography, 34, 50, 379
Hough transform, 221, 233
cascaded, 224
cube map, 223
generalized, 222
Human body shape modeling, 533
Human motion tracking, 530
activity recognition, 534
adaptive shape modeling, 533
background subtraction, 531
flow-based, 531
initialization, 531
kinematic models, 532
particle filtering, 533
probabilistic models, 533
Hyper-Laplacian, 158, 162, 164

Ideal points, 30
Ill-posed (ill-conditioned) problems, 154
Illusions, 3
Image alignment
feature-based, 275, 475
intensity-based, 337
intensity-based vs. feature-based, 393
Image analogies, 458
Image blending
feathering, 400
GIST, 405
gradient domain, 404
image stitching, 398
Poisson, 404
pyramid, 140, 403
Image compositing, *see* Compositing
Image compression, 80
Image decimation, 130
Image deconvolution, *see* Blur removal
Image filtering, 98
Image formation
geometric, 29
photometric, 54
Image gradient, 104, 112, 345
constraint, 156
Image interpolation, 127
Image matting, 443, 464
Image processing, 89
textbooks, 89, 169
Image pyramid, 127, 175
Image resampling, 145, 175
test images, 176
Image restoration, 126, 169
blur removal, 126, 174, 175
deblocking, 179
inpainting, 169
noise removal, 126, 174, 178
using MRFs, 169
Image search, 630
Image segmentation, *see* Segmentation
Image sensing, *see* Sensing
Image statistics, 115
Image stitching, 375
automatic, 392
bundle adjustment, 388
compositing, 396
coordinate transformations, 397
cube map, 396
cylindrical, 385, 407
de-ghosting, 392, 401, 408
direct vs. feature-based, 393
exposure compensation, 405
feathering, 400
gap closing, 382
global alignment, 387
homography, 379
motion models, 378
panography, 277
parallax removal, 391
photogrammetry, 377
pixel selection, 398
planar perspective motion, 379
recognizing panoramas, 392
rotational motion, 380
seam selection, 400
spherical, 385
up vector selection, 390
Image warping, 145, 177, 341

Image-based modeling, 547
Image-based rendering, 543
  concentric mosaic, 556
  environment matte, 556
  impostors, 549
  layered depth image, 549
  layers, 549
  light field, 551
  Lumigraph, 551
  modeling vs. rendering continuum, 559
  sprites, 549
  surface light field, 555
  unstructured Lumigraph, 554
  view interpolation, 545
  view-dependent texture maps, 547
Image-based visual hull, 499
ImageNet, 629
Implicit surface, 522
Impostors, *see* Sprites
Impulse response, 100
Incremental refinement
  motion estimation, 341, 345
Incremental rotation, 41
Indexing structure, 205
Indicator function, 522
Industrial applications, 5
Infinite impulse response (IIR) filter, 107
Influence function, 158, 281, 666
Information matrix, 277, 282, 326, 664, 678
Inpainting, 457
Instance recognition, 602
  algorithm, 606
  data sets, 631
  geometric alignment, 603
  inverted index, 604
  large scale, 604
  match verification, 603
  query expansion, 608
  stop list, 605
  visual words, 605
  vocabulary tree, 607
Integrability constraint, 509
Integral image, 106
Integrating sphere, 414
Intelligent scissors, 247
Interaction potential, 159, 160, 668, 672
Interactive computer vision, 537
International Color Consortium (ICC), 414
Internet photos, 327
Interpolation, 127
Interpolation kernels
  bicubic, 128
  bilinear, 127
  binomial, 127
  sinc, 130
  spline, 130
Intrinsic camera calibration, 288
Intrinsic images, 11
Inverse kinematics (IK), 532
Inverse mapping, *see* Inverse warping
Inverse problems, 3, 154
Inverse warping, 146
ISO setting, 67
Iterated closest point (ICP), 239, 283, 515
Iterated conditional modes (ICM), 161, 670
Iterative back projection (IBP), 437
Iterative feature-based alignment, 278
Iterative sparse matrix techniques, 656
  conjugate gradient, 657
Iteratively reweighted least squares
  (IRLS), 281, 286, 350, 666

Jacobian, 276, 287, 321, 345, 654
  image, 346
  motion, 350
  sparse, 322, 334, 655
Joint bilateral filter, 435
Joint domain (feature space), 259

K-d trees, 206
K-means, 256
Kalman snakes, 243
Kanade–Lucas–Tomasi (KLT) tracker, 208
Karhunen–Loève transform, 125, 589
Kernel, 103
  bilinear, 103
  Gaussian, 103
  low-pass, 103
  Sobel operator, 104
  unsharp mask, 103
Kernel basis function, 155

Kernel density estimation, 257
Keypoint detection, *see* Feature detection
Kinematic model (chain), 532
Kruppa equations, 314

L*a*b*, *see* Color
L*u*v*, *see* Color
L_1 norm, 158, 338, 361, 523
L_∞ norm, 324
Lambertian reflection, 57
Laplacian matting, 451
Laplacian of Gaussian (LoG) filter, 104
Laplacian pyramid, 135
  blending, 141, 176, 403
  perfect reconstruction, 135
Latent Dirichlet process (LDP), 626
Layered depth image (LDI), 549
Layered depth panorama, 556
Layered motion estimation, 365
  transparent, 368
Layers
  image-based rendering, 549
Layout consistent random field, 621
Learning in computer vision, 627
Least median of squares (LMS), 281
Least squares
  iterative solvers, 286, 656
  linear, 83, 275, 283, 337, 648, 651, 662, 665, 687
  non-linear, 278, 286, 306, 654, 666, 687
  robust, *see* Robust least squares
  sparse, 322, 656, 687
  total, 653
  weighted, 277, 433, 436, 443
Lens
  compound, 63
  nodal point, 63
  thin, 61
Lens distortions, 52
  calibration, 295
  decentering, 53
  radial, 52
  spline-based, 53
  tangential, 53
Lens law, 61
Level of detail (LOD), 520
Level sets, 248, 249
  fast marching method, 248
  geodesic active contour, 248
Levenberg–Marquardt, 279, 326, 334, 655, 684
Lifting, *see* Wavelets
Light field
  higher dimensional, 558
  light slab, 552
  ray space, 553
  rendering, 551
  surface, 555
Lightness, 74
Line at infinity, 30
Line detection, 220
  Hough transform, 221, 233
  RANSAC, 224
  simplification, 220, 233
  successive approximation, 221, 233
Line equation, 30, 31
Line fitting, 83, 233
  uncertainty, 233
Line hull, *see* Visual hull
Line labeling, 11
Line process, 170, 484, 669
Line spread function (LSF), 417
Line-based structure from motion, 330
Linear algebra, 645
  least squares, 651
  matrix decompositions, 646
  references, 646
Linear blend, 91
Linear discriminant analysis (LDA), 593
Linear filtering, 98
Linear operator, 91
  superposition, 91
Linear shift invariant (LSI) filter, 100
Live-wire, 247
Local distance functions, 596
Local operator, 98
Locality sensitive hashing (LSH), 205
Locally adaptive histogram equalization, 96
Location recognition, 609
Loopy belief propagation (LBP), 163, 673
Low-pass filter, 103
  sinc, 103
Lumigraph, 551

unstructured, 554
Luminance, 73
Lumisphere, 555

M-estimator, 281, 338, 666
Mahalanobis distance, 256, 591, 594, 663
Manifold mosaic, 400, 569
Markov chain Monte Carlo (MCMC), 665, 670
Markov random field, 158, 668
cliques, 160, 669
directed edges, 266
dynamic, 675
flux, 266
inference, *see* MRF inference
layout consistent, 621
learning parameters, 158
line process, 170, 484, 669
neighborhood, 160, 668
order, 160, 669
random walker, 267
stereo matching, 484
Marr's framework, 12
computational theory, 12
hardware implementation, 12
representations and algorithms, 12
Match move, 324
Matrix decompositions, 646
Cholesky, 650
eigenvalue (ED), 647
QR, 649
singular value (SVD), 646
square root, 650
Matte reflection, 57
Matting, 92, 94, 443, 464
alpha matte, 93
Bayesian, 449
blue screen, 94, 171, 445
difference, 94, 172, 446, 531
flash, 454
GrabCut, 450
Laplacian, 451
natural, 446
optimization-based, 450
Poisson, 450
shadow, 452
smoke, 452
triangulation, 445, 454
trimap, 446
two screen, 445
video, 454
Maximally stable extremal region (MSER), 195
Maximum a posteriori (MAP) estimate, 159, 668
Mean absolute difference (MAD), 479
Mean average precision, 202
Mean shift, 254, 258
bandwidth selection, 259
Mean square error (MSE), 81, 479
Measurement equation (model), 306, 662
Measurement matrix, 316
Measurement model, *see* Bayesian model
Medial axis transform (MAT), 114
Median absolute deviation (MAD), 338
Median filter, 108
weighted, 109
Medical image registration, 358
Medical image segmentation, 268
Membrane, 155
Mesh-based warping, 149, 177
Metamer, 72
Metric learning, 596
Metric tree, 207
MIP-mapping, 147
trilinear, 148
Mixture of Gaussians, 239, 243, 256
color model, 447
expectation maximization (EM), 256
mixing coefficient, 256
soft assignment, 256
Model selection, 378, 668
Model-based reconstruction, 523
architecture, 524
heads and faces, 526
human body, 530
Model-based stereo, 524, 547
Models
Bayesian, 158, 667
forward, 3
physically based, 13
physics-based, 3
probabilistic, 3
Modular eigenspace, 595

Modulation transfer function (MTF), 70, 417
Morphable model
  body, 533
  face, 528, 561
  multidimensional, 561
Morphing, 152, 178, 545, 546
  3D body, 533
  3D face, 528
  automated, 372
  facial feature, 561
  feature-based, 152, 178
  flow-based, 372
  video textures, 563
  view morphing, 546, 570
Morphological operator, 112
  closing, 112
  dilation, 112
  erosion, 112
  opening, 112
Morphology, 112
Mosaic, *see* Image stitching
Mosaics
  motion models, 378
  video compression, 383
  whiteboard and document scanning, 379
Motion compensated video compression, 341, 370
Motion compensation, 81
Motion estimation, 337
  affine, 350
  aperture problem, 347
  compositional, 351
  Fourier-based, 341
  frame interpolation, 368
  hierarchical, 341
  incremental refinement, 345
  layered, 365
  learning, 354, 361
  linear appearance variation, 349
  optical flow, 360
  parametric, 350
  patch-based, 337, 351
  phase correlation, 343
  quadtree spline-based, 358
  reflections, 369
  spline-based, 355
  translational, 337
  transparent, 368
  uncertainty modeling, 347
Motion field, 350
Motion models
  learned, 354
Motion segmentation, 373
Motion stereo, 491
Motion-based user interaction, 373
Moving least squares (MLS), 522
MRF inference, 161, 669
  alpha expansion, 163, 675
  belief propagation, 163, 672
  dynamic programming, 670
  expansion move, 163, 675
  gradient descent, 670
  graph cuts, 161, 674
  highest confidence first, 161
  highest confidence first (HCF), 670
  iterated conditional modes, 161, 670
  linear programming (LP), 676
  loopy belief propagation, 163, 673
  Markov chain Monte Carlo, 670
  simulated annealing, 161, 670
  stochastic gradient descent, 161, 670
  swap move (alpha-beta), 163, 675
  Swendsen–Wang, 670
Multi-frame motion estimation, 363
Multi-pass transforms, 149
Multi-perspective panoramas, 384
Multi-perspective plane sweep (MPPS), 391
Multi-view stereo, 489
  epipolar plane image, 490
  evaluation, 496
  initialization requirements, 496
  reconstruction algorithm, 495
  scene representation, 493
  shape priors, 495
  silhouettes, 497
  space carving, 496
  spatio-temporally shiftable window, 491
  taxonomy, 493
  visibility, 495
  volumetric, 492
  voxel coloring, 495

Multigrid, 660
algebraic (AMG), 254, 660
Multiple hypothesis tracking, 243
Multiple-center-of-projection images, 384, 569
Multiresolution representation, 132
Mutual information, 340, 358

Natural image matting, 446
Nearest neighbor
distance ratio (NNDR), 203
matching, *see* Feature matching
Negative posterior log likelihood, 159, 664, 667
Neighborhood operator, 98, 108
Neural networks, 580
Nintendo Wii, 288
Nodal point, 63
Noise
sensor, 67, 415
Noise level function (NLF), 68, 84, 415, 462
Noise removal, 126, 174, 178
Non-linear filter, 108, 169
Non-linear least squares
seeLeast squares, 278
Non-maximal suppression, *see* Feature detection
Non-parametric density modeling, 257
Non-photorealistic rendering (NPR), 458
Non-rigid motion, 332
Normal equations, 277, 346, 652, 654
Normal map (geometry image), 520
Normal vector, 31
Normalized cross-correlation (NCC), 340, 371, 479
Normalized cuts, 260
intervening contour, 262
Normalized device coordinates (NDC), 44, 48
Normalized sum of squared differences
(NSSD), 340
Norms
L_1, 158, 338, 361, 523
L_∞, 324
Nyquist rate / frequency, 69

Object detection, 578
car, 585, 634
face, 578
part-based, 586
pedestrian, 585, 601
Object-centered projection, 51
Occluding contours, 476
Octree reconstruction, 498
Octree spline, 359
Omnidirectional vision systems, 566
Opacity, 93
Operator
linearity, 91
Optic flow, *see* Optical flow
Optical center, 47
Optical flow, 360
anisotropic smoothness, 361
evaluation, 363
fusion move, 363
global and local, 360
Markov random field, 361
multi-frame, 363
normal flow, 347
patch-based, 360
region-based, 367
regularization, 360
robust regularization, 361
smoothness, 360
total variation, 361
Optical flow constraint equation, 345
Optical illusions, 3
Optical transfer function (OTF), 70, 416
Optical triangulation, 513
Optics, 61
chromatic aberration, 63
Seidel aberrations, 63
vignetting, 64, 462
Optimal motion estimation, 320
Oriented particles (points), 521
Orthogonal Procrustes, 283
Orthographic projection, 42
Osculating circle, 477
Over operator, 93
Overview, 17

Padding, 101, 173
Panography, 277, 297
Panorama, *see* Image stitching
Panorama with depth, 384, 475, 556
Para-perspective projection, 44
Parallel tracking and mapping (PTAM), 325

Parameter sensitive hashing, 205
Parametric motion estimation, 350
Parametric surface, 519
Parametric transformation, 145, 177
Parseval's Theorem, *see* Fourier transform
Part-based recognition, 615
  constellation model, 618
Particle filtering, 243, 533, 665
Parzen window, 257
PASCAL Visual Object Classes Challenge (VOC), 631
Patch-based motion estimation, 337
Peak signal-to-noise Ratio (PSNR), 81, 126
Pedestrian detection, 585
Penumbra, 55
Performance-driven animation, 209, 530, 561
Perspective n-point problem (PnP), 285
Perspective projection, 44
Perspective transform (2D), 34
Phase correlation, 343, 371
Phong shading, 58
Photo pop-up, 623
Photo Tourism, 548
Photo-mosaic, 377
Photoconsistency, 474, 494
Photometric image formation, 54
  calibration, 412
  global illumination, 60
  lighting, 54
  optics, 61
  radiosity, 60
  reflectance, 55
  shading, 58
Photometric stereo, 510
Photometry, 54
Photomontage, 403
Physically based models, 13
Physics-based vision, 15
Pictorial structures, 11, 17, 616
Pixel transform, 91
Plücker coordinates, 32
Planar pattern tracking, 287
Plane at infinity, 31
Plane equation, 31
Plane plus parallax, 49, 356, 366, 474, 549
Plane sweep, 474, 501
Plane-based structure from motion, 331
Plenoptic function, 551
Plenoptic modeling, 546
Plumb-line calibration method, 295, 300
Point distribution model, 242
Point operator, 89
Point process, 89
Point spread function (PSF), 70
  estimation, 416, 463
Point-based representations, 521
Points at infinity, 30
Poisson
  blending, 404
  equations, 523
  matting, 450
  noise, 68
  surface reconstruction, 523
Polar coordinates, 30
Polar projection, 53, 387
Polyphase filter, 127
Pop-out effect, 4
Pose estimation, 284
  iterative, 286
Power spectrum, 123
Precision, *see* Error rates
  mean average, 202
Preconditioning, 659
Principal component analysis (PCA), 242, 580, 589, 648, 664
  face modeling, 526
  generalized, 649
  missing data, 318, 649
Prior energy (term), 159, 668
Prior model, *see* Bayesian model
Profile curves, 476
Progressive mesh (PM), 520
Projections
  object-centered, 51
  orthographic, 42
  para-perspective, 44
  perspective, 44
Projective (uncalibrated) reconstruction, 312
Projective depth, 49, 474
Projective disparity, 49, 474
Projective space, 30

PROSAC (PROgressive SAmple Consensus), 282
PSNR, *see* Peak signal-to-noise ratio
Pyramid, 127, 175
  blending, 141, 176
  Gaussian, 132
  half-octave, 135
  Laplacian, 135
  motion estimation, 341
  octave, 132
  radial frequency implementation, 140
  steerable, 140
Pyramid match kernel, 613

QR factorization, 649
Quadratic form, 156
Quadrature mirror filter (QMF), 131
Quadric equation, 31, 32
Quadtree spline
  motion estimation, 358
  restricted, 358
Quaternions, 39
  antipodal, 39
  multiplication, 40
Query by image content (QBIC), 630
Query expansion, 608
Quincunx sampling, 135

Radial basis function, 151, 155, 518
Radial distortion, 52
  barrel, 52
  calibration, 295
  parameters, 52
  pincushion, 52
Radiance map, 424
Radiometric image formation, 54
Radiometric response function, 412
Radiometry, 54
Radiosity, 60
Random walker, 267, 675
Range (of a function), 91
Range data, *see* Range scan
Range image, *see* Range scan
Range scan
  alignment, 515, 540
  large scenes, 517
  merging, 516
  registration, 515, 540
  segmentation, 515
  volumetric, 516
Range sensing (rangefinding), 512
  coded pattern, 513
  light stripe, 513
  shadow stripe, 513, 540
  spacetime stereo, 515
  stereo, 514
  texture pattern (checkerboard), 514
  time of flight, 514
RANSAC
  (RAndom SAmple Consensus), 281
  inliers, 281
  preemptive, 282
  progressive (PROSAC), 282
RAW image format, 68
Ray space (light field), 553
Ray tracing, 60
Rayleigh quotient, 262
Recall, *see* Error rates
Receiver Operating Characteristic
  area under the curve (AUC), 202
  mean average precision, 202
  ROC curve, 202, 229
Recognition, 575
  3D models, 637
  category (class), 611
  color similarity, 630
  context, 625
  contour-based, 636
  data sets, 631
  face, 588
  instance, 602
  large scale, 628
  learning, 627
  part-based, 615
  scene understanding, 625
  segmentation, 620
  shape context, 636
Rectangle detection, 226
Rectification, 472, 500
  standard rectified geometry, 473
Recursive filter, 107
Reference plane, 49

Reflectance, 55
Reflectance map, 509
Reflectance modeling, 535
Reflection
  di-chromatic, 60
  diffuse, 57
  specular, 58
Region
  merging, 251
  splitting, 251
Region segmentation, *see* Segmentation
Registration, *see* Image Alignment
  feature-based, 275
  intensity-based, 337
  medical image, 358
Regularization, 154, 356
  robust, 157
Regularization parameter, 155
Residual error, 276, 281, 306, 320, 338, 346, 350, 361, 651, 658
RGB (red green blue), *see* Color
Rigid body transformation, 33, 36
Robust error metric, *see* Robust penalty function
Robust least squares, 224, 226, 281, 338, 666
  iteratively reweighted, 281, 286, 350, 666
Robust penalty function, 157, 338, 349, 437, 475, 479, 480, 484, 666
Robust regularization, 157
Robust statistics, 338, 666
  inliers, 281
  M-estimator, 281, 338, 666
Rodriguez's formula, 38
Root mean square error (RMS), 81, 339
Rotations, 37
  Euler angles, 37
  axis/angle, 37
  exponential twist, 39
  incremental, 41
  interpolation, 41
  quaternions, 39
  Rodriguez's formula, 38

Sampling, 69
Scale invariant feature transform (SIFT), 197
Scale-space, 12, 104, 135, 249
Scatter matrix, 589
  between-class, 593
  within-class, 592
Scattered data interpolation, 151, 518
Scene completion, 621
Scene flow, 492, 565
Scene understanding, 625
  gist, 623, 626
  scene alignment, 628
Schur complement, 322, 656
Scratch removal, 457
Seam selection
  image stitching, 400
Second-order cone programming (SOCP), 324
Seed and grow
  stereo, 476
  structure from motion, 328
Segmentation, 235
  active contours, 238
  affinities, 260
  binary MRF, 160, 264
  CONDENSATION, 246
  connected components, 115, 174
  energy-based, 264
  for recognition, 620
  geodesic active contour, 248
  geodesic distance, 267
  GrabCut, 266, 450
  graph cuts, 264
  graph-based, 252
  hierarchical, 251, 254
  intelligent scissors, 247
  joint feature space, 259
  k-means, 256
  level sets, 248
  mean shift, 254, 258
  medical image, 268
  merging, 251
  minimum description length (MDL), 264
  mixture of Gaussians, 256
  Mumford–Shah, 264
  non-parametric, 257
  normalized cuts, 260
  probabilistic aggregation, 253
  random walker, 267
  snakes, 238

splitting, 251
stereo matching, 487
thresholding, 112
tobogganing, 247, 251
watershed, 251
weighted aggregation (SWA), 263
Seidel aberrations, 63
Self-calibration, 313
bundle adjustment, 315
Kruppa equations, 314
Sensing, 65
aliasing, 69, 417
color, 71
color balance, 76
gamma, 77
pipeline, 66, 413
sampling, 69
sampling pitch, 67
Sensor noise, 67, 415
amplifier, 67
dark current, 67
fixed pattern, 67
shot noise, 67
Separable filtering, 102, 173
Shading, 58
equation, 57
shape-from, 508
Shadow matting, 452
Shape context, 219, 636
Shape from
focus, 511, 539
photometric stereo, 510
profiles, 476
shading, 508
silhouettes, 497
specularities, 511
stereo, 467
texture, 510
Shape parameters, 242, 598
Shape-from-X, 12
focus, 12
photometric stereo, 12
shading, 12
texture, 12
Shift invariance, 100
Shiftable multi-scale transform, 140
Shutter speed, 66
Signed distance function, 248, 515, 521, 522
Silhouette-based reconstruction, 497
octree, 498
visual hull, 497
Similarity transform, 34, 37
Simulated annealing, 161, 670
Simultaneous localization and mapping (SLAM), 324
Sinc filter
interpolation, 130
low-pass, 103
windowed, 130
Single view metrology, 292, 300
Singular value decomposition (SVD), 646
Skeletal set, 324, 328
Skeleton, 114, 219
Skew, 46, 47
Skin color detection, 85
Slant edge calibration, 417
Slippery spring, 240
Smoke matting, 452
Smoothness constraint, 155
Smoothness penalty, 155
Snakes, 238
ballooning, 239
dynamic, 243
internal energy, 238
Kalman, 243
shape priors, 241
slippery spring, 240
Soft assignment, 256
Software, 682
Space carving
multi-view stereo, 496
Spacetime stereo, 515
Sparse flexible model, 617
Sparse matrices, 655, 687
compressed sparse row (CSR), 655
skyline storage, 655
Sparse methods
direct, 655, 687
iterative, 656, 687
Spatial pyramid matching, 614
Spectral response function, 75

Spectral sensitivity, 75
Specular flow, 511
Specular reflection, 58
Spherical coordinates, 31, 223, 225, 385
Spherical linear interpolation, 41
Spin image, 515
Splatting, *see* Forward warping
    volumetric, 521
Spline
    controlled continuity, 155
    octree, 359
    quadtree, 358
    thin plate, 155
Spline-based motion estimation, 355
Splining images, *see* Laplacian pyramid blending
Sprites
    image-based rendering, 549
    motion estimation, 365
    video, 563
    video compression, 383
    with depth, 550
Statistical decision theory, 662, 665
Steerable filter, 105, 174
Steerable pyramid, 140
Steerable random field, 162
Stereo, 467
    aggregation methods, 481, 501
    coarse-to-fine, 485
    cooperative algorithms, 485
    correspondence, 469
    curve-based, 476
    dense correspondence, 477
    depth map, 469
    dynamic programming, 485
    edge-based, 475
    epipolar geometry, 471
    feature-based, 475
    global optimization, 484, 502
    graph cut, 485
    layers, 488
    local methods, 480
    model-based, 524, 547
    multi-view, 489
    non-parametric similarity measures, 479
    photoconsistency, 474
    plane sweep, 474, 501
    rectification, 472, 500
    region-based, 480
    scanline optimization, 487
    seed and grow, 476
    segmentation-based, 480, 487
    semi-global optimization, 487
    shiftable window, 491
    similarity measure, 479
    spacetime, 515
    sparse correspondence, 475
    sub-pixel refinement, 482
    support region, 480
    taxonomy, 469, 478
    uncertainty, 482
    window-based, 480, 501
    winner-take-all (WTA), 481
Stereo-based head tracking, 483
Stiffness matrix, 156
Stitching, *see* Image stitching
Stochastic gradient descent, 161
Structural Similarity (SSIM) index, 126
Structure from motion, 305
    affine, 317
    bas-relief ambiguity, 326
    bundle adjustment, 320
    constrained, 329
    factorization, 315
    feature tracks, 327
    iterative factorization, 318
    line-based, 330
    multi-frame, 315
    non-rigid, 332
    orthographic, 315
    plane-based, 319, 331
    projective factorization, 318
    seed and grow, 328
    self-calibration, 313
    skeletal set, 324, 328
    two-frame, 307
    uncertainty, 326
Subdivision surface, 519
    subdivision connectivity, 520
Subspace learning, 596
Sum of absolute differences (SAD), 338, 371, 479

Sum of squared differences (SSD), 337, 371, 479
bias and gain, 339
Fourier-based computation, 342
normalized, 340
surface, 186, 348
weighted, 339
windowed, 339
Sum of sum of squared differences (SSSD), 489
Summed area table, 106
Super-resolution, 436, 463
example-based, 438
faces, 439
hallucination, 438
prior, 437
Superposition principle, 91
Superquadric, 522
Support vector machine (SVM), 582, 585
Surface element (surfel), 521
Surface interpolation, 518
Surface light field, 555
Surface representations, 518
non-parametric, 519
parametric, 519
point-based, 521
simplification, 520
splines, 519
subdivision surface, 519
symmetry-seeking, 519
triangle mesh, 519
Surface simplification, 520
Swendsen–Wang algorithm, 670

Telecentric lens, 42, 512
Temporal derivative, 346, 360
Temporal texture, 563
Term frequency-inverse document frequency (TF-IDF), 605
Testing algorithms, viii
TextonBoost, 621
Texture
shape-from, 510
Texture addressing mode, 102
Texture map
recovery, 534
view-dependent, 535, 547
Texture mapping
anisotropic filtering, 148
MIP-mapping, 147
multi-pass, 149
trilinear interpolation, 148
Texture synthesis, 455, 465
by numbers, 459
hole filling, 457
image quilting, 456
non-parametric, 455
transfer, 458
Thin lens, 61
Thin-plate spline, 155
Thresholding, 112
Through-the-lens camera control, 287, 324
Tobogganing, 247, 251
Tonal adjustment, 97, 172
Tone mapping, 427
adaptive, 427
bilateral filter, 428
global, 427
gradient domain, 430
halos, 428
interactive, 431
local, 427
scale selection, 431
Total least squares (TLS), 233, 349, 653
Total variation, 158, 361, 523
Tracking
feature, 207
head, 483
human motion, 530
multiple hypothesis, 243
planar pattern, 287
PTAM, 325
Translational motion estimation, 337
bias and gain, 339
Transparency, 93
Travelling salesman problem (TSP), 239
Tri-chromatic sensing, 72
Tri-stimulus values, 72, 75
Triangulation, 305
Trilinear interpolation, *see* MIP-mapping
Trimap (matting), 446
Trust region method, 655
Two-dimensional Fourier transform, 123

Uncanny valley, 3
Uncertainty
    correspondence, 277
    modeling, 282, 678
    weighting, 277
Unsharp mask, 103
Upsampling, *see* Interpolation

Vanishing point
    detection, 224, 234
        Hough, 224
        least squares, 226
    modeling, 524
    uncertainty, 234
Variable reordering, 656
    minimum degree, 656
    multi-frontal, 656
    nested dissection, 656
Variable state dimension filter (VSDF), 323
Variational method, 154
Video compression
    motion compensated, 341
Video compression (coding), 370
Video denoising, 364
Video matting, 454
Video objects (coding), 365
Video sprites, 563
Video stabilization, 354, 372
Video texture, 561
Video-based animation, 560
Video-based rendering, 560
    3D video, 564
    animating pictures, 564
    sprites, 563
    video texture, 561
    virtual viewpoint video, 564
    walkthroughs, 566
VideoMouse, 288
View correlation, 324
View interpolation, 315, 545, 570
View morphing, 315, 546, 563
View-based eigenspace, 595
View-dependent texture maps, 547
Vignetting, 64, 339, 416, 462
    mechanical, 65
    natural, 64
Virtual viewpoint video, 564
Visual hull, 497
    image-based, 499
Visual illusions, 3
Visual odometry, 324
Visual words, 207, 605, 612
Vocabulary tree, 207, 607
Volumetric 3D reconstruction, 492
Volumetric range image processing (VRIP), 516
Volumetric representations, 522
Voronoi diagram, 400
Voxel coloring
    multi-view stereo, 495

Watershed, 251, 257
    basins, 251, 257
    oriented, 251
Wavelets, 136, 176
    compression, 176
    lifting, 138
    overcomplete, 137, 140
    second generation, 139
    self-inverting, 140
    tight frame, 137
    weighted, 139
Weaving wall, 477
Weighted least squares (WLS), 431, 443
Weighted prediction (bias and gain), 339
White balance, 76, 85
Whitening transform, 591
Wiener filter, 123, 124, 174
Wire removal, 457
Wrapping mode, 102

XYZ, *see* Color

Zippering, 516

THE SCIENCE AND APPLICATIONS OF ACOUSTICS

THE SCIENCE AND APPLICATIONS OF ACOUSTICS

SECOND EDITION

Daniel R. Raichel
CUNY Graduate Center
and
School of Architecture, Urban Design and Landscape Design
The City College of the City University of New York

With 253 Illustrations

Springer

Daniel R. Raichel
2727 Moore Lane
Fort Collins, CO 80526
USA
draichel@comcast.net

Library of Congress Control Number: 2005928848

ISBN-10: 0-387-26062-5 eISBN: 0-387-30089-9 Printed on acid-free paper.
ISBN-13: 978-0387-26062-4

© 2006 Springer Science+Business Media, Inc.
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, Inc., 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.
The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed in the United States of America. (TB/MVY)

9 8 7 6 5 4 3 2 1

springeronline.com

To Geri, Adam, Dina, and Madison Rose

Preface

The science of acoustics deals with the creation of sound, sound transmission through solids, and the effects of sound on both inert and living materials. As a mechanical effect, sound is essentially the passage of pressure fluctuations through matter as the result of vibrational forces acting on that medium. Sound possesses the attributes of wave phenomena, as do light and radio signals. But unlike its electromagnetic counterparts, sound cannot travel through a vacuum. In *Sylva Sylvarum* written in the early seventeenth century, Sir Francis Bacon deemed sound to be "one of the subtlest pieces of nature," but he complained, "the nature of sound in general hath been superficially observed." His accusation of superficiality from the perspective of the modern viewpoint was justified for his time, not only for acoustics, but also for nearly all branches of physical science. Frederick V. Hunt (1905–1967), one of America's greatest acoustical pioneers, pointed out that "the seeds of analytical self-consciousness were already there, however, and Bacon's libel against acoustics was eventually discharged through the flowering of a clearer comprehension of the physical nature of sound."

Modern acoustics is vastly different from the field that existed in Bacon's time and even 20 years ago. It has grown to encompass the realm of ultrasonics and infrasonics in addition to the audio range, as the result of applications in materials science, medicine, dentistry, oceanology, marine navigation, communications, petroleum and mineral prospecting, industrial processes, music and voice synthesis, animal bioacoustics, and noise cancellation. Improvements are still being made in the older domains of music and voice reproduction, audiometry, psychoacoustics, speech analysis, and environmental noise control.

This text—aimed at science and engineering majors in colleges and universities, principally undergraduates in the last year or two of their programs and graduation students, as well as practitioners in the field—was written with the assumption that the users of this text are sufficiently versed in mathematics up to and including the level of differential and partial differential equations, and that they have taken the sequence of undergraduate physics courses that satisfy engineering accreditation criteria. It is my hope that a degree of mathematical elegance has been sustained here, even with the emphasis on engineering and scientific applications. While the use of SI units is stressed, very occasional references are made to physical

parameters expressed in English (or Imperial) units. It is strenuously urged that laboratory experience be included in the course (or courses) in which this text is being used. The student of acoustics will thus obtain a far keener appreciation of the topics covered in "recitation" classes when he or she gains "hand on" experience in the use of sound—level meters, signal generators, frequency analyzers, and other measurement tools.

Many of the later chapters in the text are self-contained in the sense that an instructor may skip certain segments in order to concentrate on the agenda most appropriate to the class. However, mastery of the materials in the earlier chapters, namely, Chapters 1–6, is obviously requisite to understanding of the later chapters. Chapters such as those dealing with musical instruments or underwater sound propagation or the legal aspects of environmental noise can be skipped in order to accommodate academic schedules or to allow concentration on certain topics of greater interest to the instructor (and, hopefully, his or her class) such as ultrasound, architectural acoustics, or other topics. Problems of different levels of difficulty are included at the end of nearly all of the chapters. Many of the problems entail the theoretical aspects of acoustics, but a number of "practical" questions have also been included.

As an author, I hope that I have successfully met the challenge of providing a modern, fairly comprehensive text in the field for the benefit of both students and practitioners, whether they are scientists or engineers. In using parts of this book in prepublication editions in teaching acoustics classes, I have benefited from feedback and suggestions from my students. A number of them have proven to be quite eagle-eyed, as they have supplied a continuous stream of recommendations and corrections, even after the publication of the first edition. It is impossible to acknowledge them all, but Gregory Miller and José Sinabaldi come to mind as being among the most assiduous. A number of my colleagues and friends have gone through the chapters of the first edition. The real genesis of the first edition occurred when Harry Himmelblau saw the prepublication copy when I was a summer visiting professor at Caltech's Jet Propulsion Laboratory, and he urged me to consider publication. In particular I must acknowledge Paul Arveson, now retired from the Naval Surface Warfare Center, Carderock of Bethesda, Maryland, who went through the first three chapters with a fine-toothed comb, M. G. Prasad of Stevens Institute of Technology who made a number of extremely valuable suggestions for Chapter 9 in instrumentation, and Edith Corliss who greatly encourage me on Chapter 10 dealing with the mechanism of hearing. Dr. Zouhair Lazreq, who did his postdoctorate under my tutelage, also looked over some of the chapters, Martin Alexander has been helpful in obtaining illustrations for Chapter 9 in both editions from Brüel and Kjær; Dr. Volker Irmer of Germany's Federal Environmental Agency introduced me to the European Union's noise regulations and other international codes, and Armand Lerner arranged to have materials forwarded from Eckel Corporation of Cambridge, Massachusetts. James E. West, formerly of Lucent Bell Laboratories (and now at the Johns Hopkins University) and past president (1998–1999) of the Acoustical Society of America, was instrumental

in providing photographs of the anechoic chamber. I am also indebted to Caleb Cochran of the Boston Symphony Orchestra, Steve Lowe of the Seattle Symphony Orchestra, Elizabeth Canada of the Kennedy Center, Sandi Brown of the Minnesota Orchestral Association, Rachelle B. Roe of the Los Angeles Philharmonic, Thomas D. Rossing of Northern Illinois University, Ann C. Perlman of the American Institute of Physics, Karen Welty of Abbott Laboratories, Tom Radler of Hohner, Inc., and others, too many to list here, for their help in providing photographs, certain figures, and/or permission to reproduce the figures.

I regarded the preparation of this second edition as a splendid opportunity to update *The Science and Applications of Acoustics*. A number of features have been added to this new edition. Besides the obvious updating of information on acoustic research and applications throughout the text, a section on prosthetic hearing devices was added to Chapter 10; and the original Chapter 17 was split into two chapters, one covering music and music instrumentation and the other dealing with audio processors and sound reproduction. The topic of ultrasound has also been expanded to the extent that two chapters became necessary, with the latter chapter treating the increasingly important topic of medical and industrial applications. An introduction to nonlinear acoustics is provided in Chapter 21.

I also must take this opportunity to thank many of my fellow acousticians for their comments and suggestions for the second edition. It is hoped that all of the errors in the first edition has been weeded out and there are precious few, if any, in this volume. Suggestions for improving the text have come from M. G. Prasad, Stevens Institute of Technology; Yves Berthelot, Georgia Institute of Technology; Mark Hamilton, University of Texas at Austin; Neville H. Fletcher, Australian National University; Uwe Hansen, Indiana State; Frank J. Fahy, University of Southhampton; Carleen M. Hutchins, Violin Family Association; and others.

Springer-Verlag's Dr. Hans Koelsch and Ronald Johnson served ably as the editor and acquisitions editor, respectively. Komila Bhat supervised the editing process and Natacha Menar proved to be instrumental in expediting this publication; their contribution surely helped to improve this second edition. It was a pleasure to work with them. I am still grateful for the past contributions of Dr. Thomas von Foerster and Steven Pisano, who both worked with me at Springer-Verlag on the first edition. Dr. Robert Beyer, the editor of this AIP series dealing with acoustics, provided a great deal of encouragement and inspiration. He has my unbounded admiration (and that of virtually every acoustician) for the range of his knowledge and extraordinary wisdom. I deem it a rare privilege to know such a person.

In the preparation of the second edition, my chief source of inspiration and support continues to come from my wife, Geri. My past and present works were stimulated by the radiance of her presence.

Daniel R. Raichel
Fort Collins, Colorado

References

Bacon, Sir Francis (Lord Veralum). 1616 (published posthumously). *Sylva Sylvarum*. In *The Works of Sir Francis Bacon*, vol. 2. 1957. Spedding, Ellis, R. L., Heath, D. D., et al. (eds.). London: Longman and Co. 1957.

Hunt, Frederick Vinton. 1992. *Origins in Acoustics*. Woodbury, NY: Acoustical Society of America.

Contents

Preface **vii**

1. A Capsule History of Acoustics **1**

2. Fundamentals of Acoustics **13**

3. Sound Wave Propagation and Characteristics **31**

4. Vibrating Strings **71**

5. Vibrating Bars **89**

6. Membrane and Plates **111**

7. Pipes, Waveguides, and Resonators **131**

8. Acoustic Analogs, Ducts, and Filters **151**

9. Sound-Measuring Instrumentation **173**

10. Physiology of Hearing and Psychoacoustics **213**

11. Acoustics of Enclosed Spaces: Architectural Acoustics **243**

12. Walls, Enclosures, and Barriers **281**

13. Criteria and Regulations for Noise Control **319**

14. Machinery Noise Control **357**

15. Underwater Acoustics **409**

16. Ultrasonics **443**

17. Commercial and Medical Ultrasound Applications **479**

18. Music and Musical Instruments **509**

19. Sound Reproduction **569**

20. Vibration and Vibration Control **585**

21. Nonlinear Acoustics **617**

Appendix A. Physical Properties of Matter **629**

Appendix B. Bessel Functions **633**

Appendix C. Using Laplace Transforms to Solve Differential Equations **637**

Index **649**

1
A Capsule History of Acoustics

Of the five senses that we possess, hearing probably ranks second only to sight in regular usage. It is therefore with little wonder that human interest in acoustics would date to prehistoric times. Sound effects entailing loud clangorous noises were used to terrorize enemies in the course of heated battles; yet the gentler aspects of human nature became manifest through the evolution of music during primeval times, when it was discovered that the plucking of bow strings and the pounding of animal skins stretched taut made for rather interesting and pleasurable listening. Life in prehistoric society was fraught with emotion, just as in the present time, so music became a medium of expression. Speech enhanced by musical inflection became song. Body motion following the rhythm of accompanying music evolved into dance. Animal horns were fashioned into musical instruments (the Bible described the ancient Israelites' use of *shofarim*, made from horns of rams or gazelles, to sound alarms for the purpose of rousing warriors to battle). Ancient shepherds amused themselves during their lonely vigils playing on pipes and reeds, the precursors of modern woodwinds.

Possibly the first written set of acoustical specifications may be found in the Old Testament, Exodus XXVI:7:

> And thou shalt make curtains of goats' hair for a tent over the tabernacle . . . . The length of each curtain shall be thirty cubits and the breadth of each curtain shall be four cubits. . . .

Additional specifications are given in extreme detail for the construction and hanging of these curtains, which were to be draped over the tabernacle walls to ensure that the curtains would hang in generous sound-absorbing folds. More fine details on the construction of the tabernacle followed. Absolutely no substitution of materials nor deviation from prescribed methods was permitted.

With the advent of metal forming skills, newer wind instruments were constructed of metals. The march evolved from ceremonial processions, on grand military and ceremonial occasions. Patriotic fervor often was elevated to a state of higher pitch by the blare of martial music, indeed to the point of sheer madness on the part of the citizenry, even in modern times as epitomized during the 1930s by the grandiose thunder of Nazi goose-stepping marches through Berlin's boulevards to the accompaniment of the crowds' roar.

With sound as a major factor affecting human lives, it was only natural for interest in the science of sound, or *acoustics*, to emerge. In the twenty-seventh century BCE, Lin-lun, a minister of the Yellow Emperor Huangundi, was commissioned to establish a standard pitch for music. He cut a bamboo stem between the nodes to make his fundamental note, resulting in the "Huang-zhong pipe"; the other notes took their place in a series of twelve standard pitch pipes. He also took on the task of casting twelve bells in order to harmonize the five notes, so as to enable the composing of regal music for royalty. Archeological studies of the unearthed musical instruments attested to the high level of instrument design and the art of metallurgy in ancient China. Approximately 2000 BCE, another Chinese, the philosopher Fohi, attempted to establish a relationship between the pitch of a sound and the five elements: earth, water, air, fire, and wind. The ancient Hindus systematized music by subdividing the octave into 22 steps, with a large whole tone containing four steps, a small tone assigned three, and a half tone containing two such steps. The Arabs carried matters further by partitioning the octave into 17 divisions. But the ancient Greeks developed musical concepts similar to those of the modern Western world. Three tonal genders—the diatonic, the chromatic, and the enharmonic—were attributed to the gods.

Observation of water waves may have influenced the ancient Greeks to surmise that sound is an oscillating perturbation emanating from a source over large distances of propagation. It cannot have failed to attract notice that the vibrations of plucked strings of a lute can be seen as well as felt. The honor of being the earliest acousticians probably falls to the Greek philosopher Chrysippus (ca. 240 BCE), the Roman architect-engineer Vitruvius (also known as Marcus Vitruvius Pollio, ca. 25 BCE), and the Roman philosopher Severinus Boethius (480–524). Aristotle (384–322 BCE) stated in rather pedantic fashion that air motion is generated by a source "thrusting forward in like movement the adjoining air, so that sound travels unaltered in quality as far as the disturbance of the air manages to reach." Pythagoras (570–497 BCE) observed that "air motion generated by a vibrating body sounding a single musical note is also vibratory and of the same frequency as the body;" and it was he who successfully applied mathematics to the musical consonances described as the octave, the fifth and the fourth, and established the inverse proportionality of the length of a vibrating string with its pitch. The forerunner of the modern megaphone was used by Alexander the Great (400 BCE) to summon his troops from distances as far as 15 km.

The principal laws of sound propagation and reflection were understood by the ancient Greeks, and the echo figured prominently in a number of classical tales. Quintillianus demonstrated with small straw segments the resonance of a string in air. Vitruvius, after making use of the spread of circular waves on a water's surface as an example, went on to explain that true sound waves travel in a three-dimensional world not as circles, but rather as outwardly spreading spherical waves. He also described the placement of rows of large empty vases for the purpose of improving the acoustics of ancient theaters. While there may be some question if such vases have actually been employed in these theaters (since archeological excavations have failed to disclose their shards), it does presage

knowledge of room acoustics on Vitruvius's part. These vases would have the effect of low-frequency absorption, similar to that of special panels that are used today as absorbers. As these amphitheaters were constructed in stony recesses which provide little or no low-frequency absorption, such vases would definitely improve the acoustics of the ancient theaters. There is evidence of Lucius Mummius, who, after destroying Corinth's theater, brought its bronze vessels to Rome and made a dedicatory offering from the proceeds of their sale to the goddess Luna in her temple at Rome.

Aristotle's eschewal of experiments (which he deemed unworthy of a scientist) to establish the validity of hypotheses essentially caused the stagnation of all natural sciences, including acoustics, such was the sway of his authority until the end of the Middle Ages.

Leonardo da Vinci (1452–1519) knew as the ancients did that "there cannot be any sound when there is no movement or percussion of the air." His observations led him to correlate the waves generated by a stone cast into water with the propagation of sound waves as similar phenomena. He also ascertained that wave motion of a sound has a definite value of velocity, and he noted that "the stroke of one bell is answered by a feeble quivering and ringing of another bell nearby; a string sounding on a lute, compels to sound on another lute, nearby, a string of the same note," thus anticipating by nearly a century Galileo Galilei's discovery of sympathetic resonance.

Almost no further progress in acoustics was made until the seventeenth century when a relationship was established between pitch and frequency. Marin Mersenne (1588–1648), a French natural philosopher and Franciscan friar, may be considered to be the "father of modern acoustics." In *Harmonie universelle*, published in 1636, he rendered the first scientifically palpable description of an audible tone (84 Hz), and he demonstrated that the absolute frequency ratio of two vibrating strings, radiating a musical note and its octave, is of the frequency ratio 1:2. An analog with water waves is drawn: the belief was registered that the air motion generated by musical sounds is oscillatory in nature, and it was observed that sound travels with a finite speed. Sound is also known to bend around corners, suggestive of diffraction effects which are also commonly observed in water waves. Mersenne measured the velocity of sound by counting the number of heart beats during the interval occurring between the flash of a shot and the perception of the sound.

Independently of Mersenne, Galileo Galilei (1564–1642), in his *Mathematical Discourses Concerning the New Sciences* (1638), supplied to date the most lucid statement and discussion of frequency equivalence. It is interesting to note that the wave viewpoint was not accorded unanimous acceptance among the early scientists. Pierre Gassendi (1582–1655), a contemporary of Galileo and Mersenne, argued for a ray theory whereby sound is attributed to a stream of atoms emitted by the sounding body; the velocity of sound is the speed of atoms in motion, and the frequency is the number of atoms emitted per unit time. He also attempted to demonstrate that sound velocity was independent of pitch by comparing results of the crack of a rifle with those for the deep roar of a cannon.

Robert Boyle (1626–1691) with the help of his assistant Robert Hooke (1635–1703) performed a classic experiment (1660) on sound by placing a ticking watch in a partially evacuated glass chamber. He proved that air is necessary, either for the production or emission of sound. In this respect he disproved Athenasius Kircher's (1602–1680) negative experiment in which the latter enclosed a bell in a vacuum container and excited the bell magnetically from the exterior. Kirchner's results were erroneous because he did not take the precaution to prevent the conduction of sound through the bell's supports to the surroundings. Francis Hausksbee (1666–1713) repeated the Boyle experiment (in a modified form) before the Royal Society.

Mention should be made here of Joseph Sauveur (1653–1713) who suggested the term *acoustics* (from the Greek word for sound) for the science of sound. In describing his research on the physics of music at the College Royal in Paris, he introduced terms such as *fundamental*, *harmonics*, *node*, and *ventral segment*.[1] It is also an interesting footnote to history that Sauveur may have been born with defective hearing and speaking mechanisms; he was reported to have been a deaf–mute until the age of 7. He took an immense interest in music even though he had to rely on the help of his assistants to compensate for his lack of keen musical discernment in conducting acoustic experiments.

Franciscus Mario Grimaldi (1613–1663) published *Physicomathesis de lumine, coloribus et tride*, which dealt with experimental studies of diffraction, much of which was to apply to acoustics as well as to light, and in 1678 Hooke announced his law relating force to deformation, which established the foundations of vibration and elasticity theories.

Kircher's publication *Phomugia, die neue Hall- und Tonkunst* (*The New Art of Sound and Tone*), issued in 1680, provides us a rather amusing insight into the world of misconception, nostrums, and plain scientific hokum that were prevalent at the time. While delving into the phenomena of echoes and whispering galleries, the text recommended music as the only remedy against tarantula bites and provided a discourse on wines. In the chapter on wines, Kircher claimed that old wine has purified itself and acquired a deep soul. If old wine is poured into a glass, which is then struck, a sound will emanate. On the other hand, new wine was deemed to be "jumpy" as a child and bereft of a sound. Hence, recent wine in a glass will not sound. Another misconception widely believed at the time was that sound could be trapped in a little box and preserved indefinitely, the idea of attenuation or absorption of sound being completely alien then. It was even proposed by a Professor Hut of the music academy at Frankfurt that a communications tube be constructed to transmit speech over long distances.

Ernst F. F. Chladni (1756–1827), author of the highly acclaimed *Die Akustik*, is often credited for establishing the field of modern experimental acoustics through

[1] Nearly 20 years earlier, in 1683, Narcissus Marsh, then the Bishop of Ferns and Leighlin in the Protestant Church, published an article "An Introductory Essay to the Doctrine of Sounds, Containing Some Proposals for the Improvement of Acousticks" in the *Philosophical Transactions of the Royal Society of London*. He was using the term "acousticks" to denote direct sound as distinguished from reflected and diffracted sound.

his discovery of torsional vibrations and measurements of the velocity of sound with the aid of vibrating rods and resonating pipes. The dawn of the eighteenth century saw the birth of theoretical physics and applied mechanics, particularly under the impetus of archrivals Isaac Newton (1642–1726) and Gottfried Wilhelm Leibniz (1646–1716). Newton's theoretical derivation of the speed of sound (in the *Principia*) motivated a spate of experimental measurements by Royal Society members John Flamsteed (1646–1719), the founder of the Greenwich Observatory and the first Astronomer Royal, and his eventual successor (in 1720), Edmund Halley (1656–1742); also by Giovanni Domenico Cassini (1625–1712), Jean Picard (1620–1682), and Olof Römer (1644–1710) of the French Acadêmie des Sciences; and nearly half century later in 1738 by a team led by César François Cassini de Thury (1714–1762), a grandson of the aforementioned G. D. Cassini who headed the earlier 1677 measurement team.

Newton's estimate was found to be in error, for in his observations he had erred by assuming an isothermal (rather than an isentropic) process as being the prevalent mode for acoustic vibrations.[2] Temperature was found to influence the speed of sound in independent separate experiments by Count Giovanni Lodovico Bianoni (1717–1781) of Bologna and Charles Marie de la Condamine (1701–1773). Other acoustic developments included the evolution of the exponential horn by Richard Helsham (1680–1758); this device loads the sound source heavily, thus causing the source to concentrate its energy more than it could without the horn and directs the output more effectively. Real understanding of this phenomenon did not come about until John William Strutt, Lord Rayleigh (1845–1919) treated the problem of source loading, and Arthur Gordon Webster (1863–1923) the theory of horns.

Each of the optical phenomena of refraction, diffraction, and interference was elucidated during the seventeenth century. But all of these phenomena were soon realized to apply to acoustics as well as to light. Willbrod Snell (or Snellius) (1591–1626) composed an essay in 1620 treating the refraction of light rays in a transparent medium such as water or glass, but he somehow neglected to publish his manuscript which was later unearthed and used by Christian Huygens (1629–1695) in his own works, which secured posthumous fame for Snell, in spite of a publication of the same law by the stellar René Descartes (1596–1650) who, it turned out, had made two erroneous assumptions, which were corrected by Pierre de Fermat (1601–1665). Fermat's principle derives from the assumption that the light always travels from a source point in one medium to a receptor point in the second medium by the path of least time. Diffraction was first observed by the Jesuit mathematician Francesco Maria Grimaldi (1618–1663) of Bologna. His experiments were repeated by Newton, Hooke, and Huygens; and soon this phenomenon that light does not always travel in straight lines but can diffuse slightly around corners constituted a core issue in the controversy between the wave and corpuscular theories of light. But it took nearly 200 years following Newton's era to resolve the

[2] Actually, what Newton really did was to assume that the "elastic force" of the fluid is proportional to its condensation, which is now realized, in the context of modern thermodynamics, to be the equivalence of the isothermal process.

conflict by embracing elements of both theories. Newton essentially squelched the wave theory until its revival by Thomas Young (1773–1829) and Augustin Jean Fresnel (1788–1817), both of whom, independently of each other, elucidated the principle of interference. On his analysis of diffraction, Fresnel drew heavily on Huygen's principle in which successive positions of a wavefront are established by the envelope of secondary wavelets.

Armed with the analytical tools afforded by the advent of calculus by Newton and Leibniz, the French mathematical school treated problems of theoretical mechanics. Among the major contributors were Joseph Louis Lagrange (1736–1813), the Bernoulli brothers James (1654–1705) and Johann (1667–1748), G. F. A. l'Hôpital (Marquis de St. Mesme) (1661–1704), Gabriel Cramer (1704–1752), Leonhard Euler (1707–1783), Jean Le Rond d'Alembert (1717–1783), and Daniel Bernoulli (1700–1783). And the next generation provided a further flowering of genius: Joseph Louis Lagrange (1736–1813), Pierre Simon Laplace (1749–1827), Adrian Marie Legendre (1736–1833), Jean Baptiste Joseph Fourier (1768–1830), and Siméon Denis Poisson (1781–1840). The nineteenth century was also dominated by discoveries in electricity and magnetism by Michael Faraday (1791–1867), James Clerk Maxwell (1831–1879), Heinrich Rudolf Hertz (1857–1894), and by the theory of elasticity, principally developed by Clause L. M. Navier (1785–1836), Augustin Louis Cauchy (1789–1857), Rudolf J. E. Clausius (1822–1888), and George Gabriel Stokes (1890–1909).

These developments constituted the foundation for understanding the physical and eventually the physiologcial aspects of acoustics. In the attempt to grasp the nature of musical sound, Simon Ohm (1789–1854) advanced the hypothesis that the ear perceived only a single, pure sinusoidal vibration and that each complex sound is resolved by the ear into its fundamental frequency and its harmonics. Hermann F. L. von Helmholtz (1821–1894) arguably deserves the credit for laying the foundations of spectral analysis in his classic *Lehre von den Tonempfindungen* (*Sensation of Sound*). The monumental two-volume *Theory of Sound*, released in 1877 and 1878 by the future Nobel laureate, Lord Rayleigh, laid down in a fairly complete fashion the theoretical foundations of acoustics.

When the newly constructed Fogg Lecture Hall was opened in 1894 at Harvard University, its acoustics was found to be so atrocious so as to render that facility almost useless. This prompted Harvard's Board of Overseers to request the physics department that something be done to rectify the situation. The task was assigned to a young Harvard researcher, Wallace Clement Sabine (1868–1919), and he discovered soon enough that excessive reverberations tend to mask the lecturer's words. In a series of papers (1900–1915) evolving from his studies of the lecture hall, he almost single-handedly elevated architectural acoustics to scientific status. Sabine helped establish the Riverbank Acoustical Laboratories[3] at Geneva, Illinois. Just prior to his scheduled assumption of his duties at Riverbank, Sabine succumbed at the young age of 50 to cancer. His distant cousin, Paul Earls Sabine

[3] Riverbank is possibly the first research facility set up specifically for study and research in acoustics.

(1879–1958), also a Harvard physicist, took on the task of running the laboratory. The development of test procedures, methodology, and standardization in testing the acoustical nature of products arose from the pioneering efforts of the younger Sabine. A third member of the family, Paul Sabine's son Hale Johnson Sabine (1909–1981), began his career in architectural acoustics at the tender age of 10 by assisting his father at Riverbank, and his efforts centered on control of noise in industry and institutions. Both father and son, Paul and Hale, served terms as president of the Acoustical Society of America.

The genesis of ultrasonics occurred in the nineteenth century with James P. Joule's (1818–1889) discovery in 1847 of the *magnetostrictive effect*, the alteration of the dimensions of a magnetic material under the influence of a magnetic field, and in 1880 with the finding by the brothers Paul-Jacques (1855–1941) and Pierre (1859–1906) Curie that electric charges result on the surfaces of certain crystals subjected to pressure or tension. The Curies' discovery of the *piezoelectric electric effect* provided the means of detecting ultrasonic signals. The inverse effect, whereby a voltage impressed across two surfaces of a crystals give rise to stresses in the materials, now constitutes the principal method of generating ultrasonic energy.

The study of underwater sound stemming from the necessity for ships to avoid dangerous obstacles in water supplied the impetus for development of ultrasonic applications. Until the early part of the twentieth century ships were warned of hazardous conditions by bells suspended from lightships. Specially trained crew members listened for these bells by pressing microphones or stethoscopes against the hulls. In the effort to counteract the German submarine threat during World War I, Robert Williams Wood (1868–1955) and Gerrard in England and Paul Langevin (1872–1946) in France were assigned the task of developing counter surveillance methods.

The youthful Russian electrical engineer, Constantin Chilowsky (1880–1958), collaborated with Langevin in experiments with an electrostatic (condenser) projector and a carbon-button microphone placed at the focus of a concave mirror. In spite of troubles encountered with leakages and breakdowns due to the high voltages necessary for the operation of the projectors, Langevin and Chilowsky were able by 1916 to obtain echoes from the ocean bottom and from a sheet of armor plate at a distance of 200 m. A year later Langevin came up with the concept of using a piezoelectric receiver and employed one of the newly developed vacuum-tube amplifiers—the earliest application of electronics to underwater sound equipment—and Wood constructed the first directional hydrophone geared to locate hostile submarines. The first devices to generate directional beams of acoustic energy also constitute the first use of ultrasonics. Reginald A. Fessenden (1866–1932), a Canadian engineer, working independently, developed a moving coil transducer operating at frequencies in the range of 500–1000 Hz to generate underwater signals. In the course of their underwater sound investigations, Wood and his co-worker Alfred L. Loomis (1887–1975), who also was a trained lawyer, and Langevin observed that small water creatures could be stunned, maimed, or even destroyed by the effects of intense ultrasonic fields.

World War I ended before underwater echo-ranging could be fully deployed to meet the German U-boat threat. The years of peace following World War I witnessed a slow but nevertheless steady advance in applying underwater sound to depth-sounding by ships. Improvements in electronic amplification and processing, magnetostrictive projectors, piezoelectric transducers provided refinements in echo-ranging. The advent of World War II heightened research activity on both sides of the Atlantic, and most of the present concepts and applications of underwater acoustics traced their origins to this period. The concept of target strength, noise output of various ships at different speeds and frequencies, reverberation in the sea, and evaluation of underwater sound through spectrum analysis were quantitatively established. It was during this period that underwater acoustics became a mature branch of science and engineering, backed by vast literature and history of achievement.

The invention of the triode vacuum tube and the advent of the telephone and radio broadcasting served to intensify interest in the field of acoustics. The development of vacuum tube amplifiers and signal generators rendered feasible the design and construction of sensitive and reliable measurement instruments. The evolution of the modern telephone system in the United States was facilitated by the progress of communication acoustics, mainly through the remarkable efforts of the Bell Telephone Laboratories.

The historic invention of the transistor (1949) at the Bell Laboratories in Murray Hill, New Jersey, gave rise to a whole slew of new devices in the field of electronics, including solid state audio and video equipment, computers, spectrum analyzers, electric power conditioners, and other gear too numerous to mention here.

Experiments and development of theory in architectural acoustics were conducted during the 1930s and the 1940s at a number of major research centers, notably Harvard, MIT, and UCLA. Vern O. Knudsen (1893–1974), eventually the chancellor of UCLA, carried on Sabine's work by conducting major research on sound absorption and transmission. The most notable of his younger associates was Cyril M. Harris (b. 1917), who was to become the principal consultant on the acoustics of the Metropolitan Opera House in New York, the John F. Kennedy Center in the District of Columbia, the Powell Symphony Hall in St. Louis, and a number of other notable edifices.

Sound decay, in terms of reverberation times, was discovered to be a decisive factor in gauging the suitability of enclosed areas for use as listening chambers. The impedance method of rating acoustical materials was established to predict the radiative patterns of sonic output, and prediction of sound attenuation in ducts was established on a scientific footing. The architectural acoustician now has a wide array of acoustical materials to choose from and to tailor the walls segmentwise in order to effect the proper acoustic environment.

Acoustics also engendered the science of psychoacoustics. Harvey Fletcher (1884–1990) led the Bell Telephone Laboratories in describing and quantifying the concepts of loudness and masking, and there, many of the determinants of speech communication were also established (1920–1940). Fletcher, now regarded as "the father of psychoacoustics," worked with the physicist Robert Millikan at

the University of Chicago, on the determination of the electron charge. Fletcher indeed performed much of the famed oil drop experiment, to the extent that many physicists feel that the student should have shared the 1923 Nobel Prize in physics with his professor who received the award for this effort. At Bell Labs, Fletcher also developed the first electronic hearing aid and invented stereophonic reproduction. Sound reproduction also constituted the domain of Harry F. Olson (1902–1982), who directed the Acoustical Laboratory at RCA and developed modern versions of loudspeakers. Warren P. Mason's (1900–1986) major work in physical acoustics essentially laid down the modern foundations of ultrasonics, and Georg von Bekésy (1849–1972) earned the Nobel Prize for his research on the mechanics of human hearing. Acoustics penetrated the fields of medicine and chemistry through the medium of ultrasonics: ultrasonic diathermy became established and certain chemical reactions were found to become accelerated under acoustic conditions.

The outbreak of World War II served to greatly intensify acoustics research at major laboratories in Western Europe and in the United States, particularly in view of the demand for sonar detection of stealthily moving submarines and for reliable speech communication in cacophonous environments such as propeller aircraft and armored vehicles. This research not only has reached great proportions, it has continued unabated to this day, at major universities and government institutions, among them being the U.S. Naval Research Laboratory, Naval Surface Warfare Center, MIT, Purdue University, Georgia Institute of Technology, and Pennsylvania State University.

Prominent among the researchers were Richard Henry Bolt (1911–2002) and Leo L. Beranek (b. 1914) who teamed up after World War II to found a major research corporation, Bolt, Beranek & Newman (now BBN Technologies); Phillip M. Morse of MIT [who authored and co-authored with Karl Uno Ingard (b. 1921) major texts in physical acoustics]; R. Bruce Lindsay (1900–1985) of Brown University; and Robert T. Beyer, who contributed to nonlinear acoustics, also at Brown. In 1947 Eugen Skudrzyk (1913–1990) began research in nearly all areas of acoustics at the Technical University of Vienna and went on to Pennsylvania State University in the United States, he wrote possibly the best comprehensive text on physical acoustics since Lord Rayleigh's *Theory of Sound*.

Karl D. Kryter (b. 1914) of California dealt with the physiological effects of noise on humans, and Carleen Hutchins (b. 1911) is still providing great insight into the design and construction of musical string instruments, in her dual role as investigating acoustician and craftsperson seeking to emulate the old Cremona masters in her hometown of Montclair, New Jersey. Laser intereferometry was applied by Karl H. Steson (b. 1937) and by Lothar Cremer (1905–1990) to visualize vibrations of the violin body. Sir James Lighthill (1924–1998), who held the Lucasian chair (once occupied by Newton) in mathematics at Cambridge University, laid down the foundations of modern aeroacoustics, building on the foundations of Lord Rayleigh's earlier research. UCLA's Isadore Rudnick (1917–1997) performed major experiments in superfluid hydrodynamics, involving sound propagation in helium at cryogeneic temperatures and also conducted studies of acoustically induced streaming modes of vibrations of elastic bodies and attenuation of sound in seawater. At

the Applied Physics Laboratory at the University of Washington, Lawrence A. Crum (b. 1941) directs major research on sonofluorescence as well as the development of ultrasound diagnostic and therapeutic medical devices. Kenneth S. Suslick (b. 1952) and his co-workers at the University of Illinois are making major contributions in the field of sonochemistry. Whitlow W. L. Au at the University of Hawaii is conducting studies on the characteristics of cetacean acoustics, including the target discrimination capabilities of dolphins and whales.

With acoustic research continuing apace, the number of great acoustcians living surely exceeds that of deceased ones.

It can truly now be said that the U.S. Navy has done more (and is still doing more) than any other institution to further acoustics research at its widespread facilities, including Naval Research Laboratory (NRL) and the Naval Surface Warfare Center (NSWC). Much magnificent work was done under the cloak of security classification during the days of the Cold War, with the consequence that many deserving researchers do not bask in the glory that have been publicly accorded professional societies' medal honorees and Nobel Prize laureates.

Robert J. Bobber (b. 1918) of NRL facility in Orlando paved the way in underwater electroacoustics measurements. Acoustics radiation constituted the domain of Sam Hanish, late of the NRL in the District of Columbia. At NSWC's David Taylor Basin in Bethesda, Maryland, Murray Strausberg (b. 1917) continues to make major contributions in the field of propeller noise, which entails the study of cavitation and hydroacoustics as he did for the past three decades; David Feit (b. 1937) ranks as a leading expert in the field of structural acoustics; and William K. Blake reigned preeminent in the category of aero-hydroacoustics (Blake, 1964). Herman Medwin (b. 1920) of the Navy Postgraduate School at Monterey, California, conducted major research in acoustical oceanography. As a senior research physicist at the U.S. Naval Surface Weapons Center, headquarters in Silver Spring, Maryland, Robert Joseph Urick (1916–1996) elucidated the characteristics of underwater acoustical phenomena, including sonar effects. He later taught the principles of underwater sound at the Catholic University of America in Washington, DC.

Acoustics is no longer the esoteric domain of interest to a few specialists in the telephone and broadcasting industries, the military, and university research centers. Legislation and subsequent action have been demanded internationally to provide quiet housing, safe and comfortable work environments in the factory and the office, quieter airports and streets, and protection in general from excessive exposure to noisy appliances and equipment.

The wiser architects are increasingly using acoustical engineers to ensure environmental harmony with the esthetic aspects of their designs. Acoustic instrumentation is being used in industry to facilitate manufacturing processes and to ensure quality control. Acoustics has even invaded the living room through the medium of high fidelity reproduction, giving rise to a spate of new equipment such as Dolby processors, digital processors, compact disc (and more lately DVD) players, multi-speaker "Surround-Sound" environment conditioners, music synthesizer circuit boards for personal computers. The escalating applications of ultrasound provide new diagnostic and therapeutic tools in the medical field, more reliable

characterization of materials, better surveillance methodologies, and improved manufacturing techniques.

And what does the future hold in acoustics? The continuing miniaturization of electronic circuitry is now resulting in digitized hearing aids that can circumvent the "cocktail party effect" (the tendency of background noise to make it difficult for the sensorneurally impaired listeners to focus on a conversation). Even newer diagnostic and therapeutic processes entailing acoustical signals are being developed and tested at major medical centers. More sensitive and versatile transducers that can withstand harsher environments lead to new acoustical devices such as sonic viscometers, undersea probes, and portable voice-recognition devices. And if we can gain a greater understanding of how cetaceans make use of their natural sonars to assess the submarine environment and perhaps to communicate with each other, we could be well on the way to constructing far more sophisticated megachannel acoustical analyzers. The generation of acoustical waves in the gigahertz range can rival or exceed the optical microscope for resolution with greater penetrating power. The repertoire of what is to come should truly constitute an amazing cornucopia of beneficence to humanity.

References

Beranek, Leo J. 1995. Harvey Fletcher: Friend and scientific critic, *Journal of the Acoustical Society of America* 97(5):3357.

Beyer, Robert T. 1995. Acoustic, acoustics. *Journal of the Acoustical Society of America* 98(1): 33–34.

Beyer, Robert T. 1999. *Sounds of Our Times*. New York: Springer-Verlag. [A fascinating history of acoustics over the past 200 years, with many allusions to even earlier history. This text picks up where Frederick Vinton Hunt left off in his unfinished, meticulously researched work which was published posthumously (seen below).]

Blake, William K. 1964. *Aero-hydroacoustics for Ships*, 2 Vols. Bethesda, MD: David Taylor Basin publication DTNSRDC-84/010, June 1964.

Bobber, Robert J. 1970. *Underwater Electroacoustic Measurements*. Washington, DC: Naval Research Laboratory.

Chladni, E. F. F. 1802. *Die Acustik*. Leipzig: Breitkopf & Hartel.

Clay, Clarence S. and Medwin, Herman. 1977. *Acoustical Oceanography: Principles and Applications*. New York: John Wiley & Sons.

Fletcher, Steven Harvey. 1995. Harvey Fletcher: A son's reflections. *Journal of the Acoustical Society of America* 97(5 Pt. 2): 3356–3357.

Galileo, Galilei. 1638 (translation published in 1939). *Dialogues Concerning Two New Sciences*, Translated by Crew, H. and De Salvio, A. Evanston, IL: Northwestern University Press.

Hanish, Sam. 1981. *A Treatise on Acoustic Radiation*. Washington, DC: Naval Research Laboratory.

Harris, Cyril M. 1995. Harvey Fletcher: Some personal recollections. *Journal of the Acoustical Society of America* 97(5 Pt. 2): 3357.

Helmholtz, Hermann F. L. von. 1877. *Lehre con den Tonempfindungen*. Braunschweig, Wiesbaden: Vieweg.

Hertz, J. H. (ed.). 1987. *The Pentateuch and Haftorahs*. London: Soncino Press.

Hunt, Frederick V. 1992 (reissue). *Origins in Acoustics*. Woodbury, NY: Acoustical Society of America. (Although left incomplete by the author at the time of his death, this text is one of the most definitive accounts by one of the great modern acoustical scientists of the history of acoustics leading up to the eighteenth century.)

Junger, Miguel C., Feit, David. 1986. *Sound, Structure, and Their Interaction*. Cambridge, MA: The MIT Press.

Kopec, John W. 1994. The Sabines at Riverbank. *Proceedings, Wallace Clement Sabine Centennial Symposium*. Woodbury, NY: Acoustical Society of America, pp. 25–28.

Lindsay, R. Bruce. 1966. The story of acoustics. *Journal of the Acoustical Society of America* 39(4): 629–644.

Lindsay, R. Bruce (ed.). 1972. *Acoustics: Historical and Philosophical Development*. Benchmark Papers in Acoustics. Stroudsburg, PA: Dowden, Hutchinson & Ross, Inc. (A most interesting compendium of selected papers by major contributors to acoustical science, ranging from Aristotle to Wallace Clement Sabine. A must-read for the serious student of the history of acoustics).

Lindsay, R. Bruce. 1880. Acoustics and the Acoustical Society of America in Historical Perspective. *Journal of the Acoustical Society of America* 68(1): 2–9.

Mersenne, Marin. 1636. *Harmonie universelle*. Paris: S. Cramoisy; English translation: Hawkins, J. 1853. *General History of the Practice and Science of Music*. London: J. A. Novello, pp. 600–616, 650 ff.

Newton, Sir Isaac. 1687. *Philosophiae Naturalis Principia Mathematica*. London: Joseph Streater for the Royal Society.

Pierce, Allan D. 1989 (reissue). *Acoustics: An Introduction to its Physical Principles and Application*. Woodbury, NY: Acoustical Society of America.

Raman, V. V. 1973. Where credit is due: Sauveur, the forgotten founder of acoustics. *Physics Teacher* pp. 161–163.

Shaw, Neil A., Klapholz, Jesse, Gander, Mark R. 1994. Books and Acoustics, especially Wallace Clement Sabine's *Collected Papers on Acoustics*. *Proceedings, Wallace Clement Sabine Centennial Symposium*. Woodbury, NY: Acoustical Society of America, pp. 41–44.

Skudrzyk, Eugen. 1971. *The Foundations of Acoustics—Basic Mathematics and Basic Acoustics*. New York: Springer-Verlag. (A text of classic proportions. Nearly one quarter of this volume lays the mathematical foundations requisite to analysis of acoustical phenomena.)

Strutt, John William (Lord Rayleigh). 1877. *Theory of Sound*. London: Macmillan & Co. Ltd. 2nd edition revised and enlarged 1894, reprinted 1926, 1929. Reprinted in two volumes, New York: Dover, 1945. (These volumes should be in every acoustician's library.)

Wang, Ji-qing. 1994. Architectural Acoustics in China, Past and Present. *Proceedings, Wallace Clement Sabine Centennial Symposium*. Woodbury, NY: Acoustical Society of America, pp. 21–24.

Webster, Arthur G. 1919. *Proceedings of the National Academy of Science* 5:275.

2
Fundamentals of Acoustics

2.1 Wave Nature of Sound and the Importance of Acoustics

Acoustics refers to the study of sound, namely, its production, transmission through solid and fluid media, and any other phenomenon engendered by its propagation through media. Sound may be described as the passage of pressure fluctuations through an elastic medium as the result of a vibrational impetus imparted to that medium. An acoustic signal can arise from a number of sources, e.g., turbulence of air or any other gas, the passage of a body through a fluid, and the impact of a solid against another solid.

Because it is a phenomenon incarnating the nature of waves, sound may contain only one frequency, as in the case of a pure steady-state sine wave, or many frequency components, as in the case of noise generated by construction machinery or a rocket engine. The purest type of sound wave can be represented by a sine function (Figure 2.1) where the abscissa represents elapsed time and the ordinate represents the displacement of the molecules of the propagation medium or the deviation of pressure, density, or the aggregate speed of the disturbed molecules from the quiescent (undisturbed) state of the propagation medium.

When the ordinate represents the pressure difference from the quiescent pressure, the upper portions of the sine wave would then represent the compressive states and the lower portions the rarefaction phases of the propagation. A sine wave is generated in Figure 2.2 by the projection of the trace of a particle A traveling in a circular orbit. This projection assumes the pattern of an oscillation, in which the particle A's projection or "shadow" A' onto an abscissa moves back and forth at a specified frequency. Frequency f is the number of times the sound pressure varies from its equilibrium value through a complete cycle per unit time. Frequency is also denoted by the angular (or radian) frequency

$$\omega = 2\pi f = \frac{2\pi}{T} \tag{2.1}$$

expressed in radians per second. The period T is the amount of time for a single cycle to occur, i.e., the length of the time it takes for a tracer point on the sine curve to reach a corresponding point on the next cycle. The reciprocal of

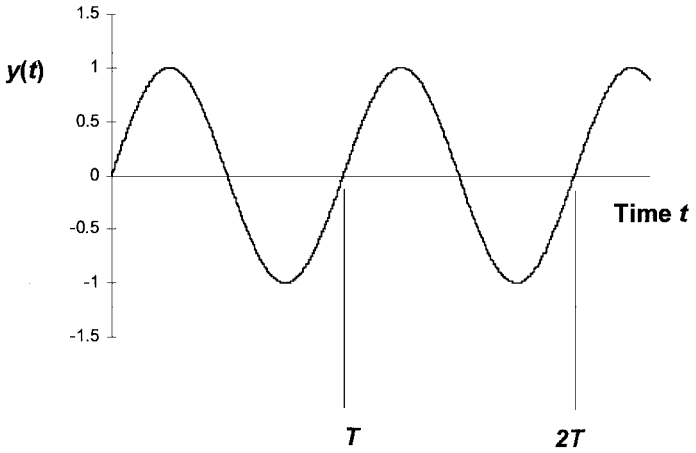


FIGURE 2.1. Plot of a sine wave $y(t) = \sin 2\pi \sin ft$ over slightly more than two periods of $T = 1/f$ s, where f is the frequency of the sine wave. $y(t)$ may be the displacement function x/x_0, velocity ratio v/v_0, pressure variation p/p_0, or condensation variation s/s_0, where the subscript 0 denotes maximum values.

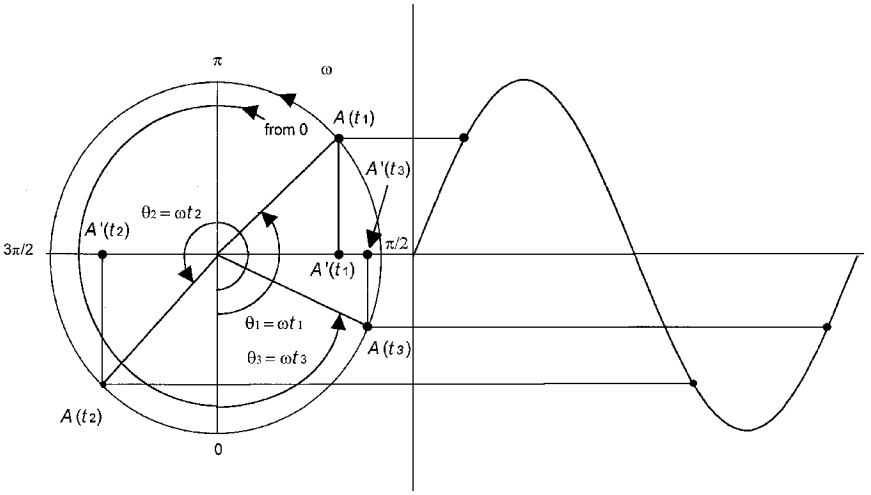


FIGURE 2.2. The oscillation of a particle A' in a sinusoidal fashion is generated by the circular motion of particle A moving in a circle with constant angular speed ω. A' is the projection of $A\cos \omega t = A\cos \theta$ onto the diameter of the circle which has a radius A. The projection of point A to the right traces a sine wave over an abscissa representing time t. The projections for three points at times t_1, t_2, and t_3 are shown here. The amplitude of the oscillation is equal to the radius of the circle, and the peak-to-peak amplitude is equal to the diameter of the circle.

period T is simply the frequency f. The most common unit of frequency used in acoustics (and electromagnetic theory) is the *hertz* (abbreviated Hz in the SI system), which is equal to one cycle per second. An acoustic signal may or may not be audible to the human ear, depending on its frequency content and intensity. If the frequencies are sufficiently high (>20 kilohertz, which can be expressed more briefly as 20 kHz), ultrasound will result, and the sound is inaudible to the human ear. This sound is said to be *ultrasonic*. Below 20 Hz, the sound becomes too low (frequency-wise) to be heard by a human. It is then considered to be *infrasonic*.

Sound in the *audio* frequency range of approximately 20 Hz–20 kHz can be heard by humans. While a degree of subjectivity is certainly entailed here, *noise* conveys the definition of *unwanted sound*. Excessive levels of sound can cause permanent hearing loss, and continued exposure can be deleterious, both physiologically and psychologically, to one's well-being.

With the advent of modern technology, our aural senses are being increasingly assailed and benumbed by noise from high-speed road traffic, passing ambulances and fire engine sirens, industrial and agricultural machinery, excessively loud radio and television receivers, recreational vehicles such as snowmobiles and unmuffled motorcycles, elevated and underground trains, jet aircraft flying at low altitudes, domestic quarrels heard through flimsy walls, and so on.

Young men and women are prematurely losing their hearing acuity as the result of sustained exposure to loud rock concerts, discotheques, use of personal cassette and compact disk players and mega-powered automobile stereo systems. In the early 1980s, during the waning days of the Cold War, the Swedish navy reported considerable difficulty in recruiting young people with hearing sufficiently keen to qualify for operating surveillance sonar equipment for tracking Soviet submarines traveling beneath Sweden's coastal waters. Oral communication can be rendered difficult or made impossible by background noise; and life-threatening situations may arise when sound that conveys information becomes masked by noise. Thus, the adverse effects of noise fall into one or more of the following categories: (1) hearing loss, (2) annoyance, and (3) speech interference.

Modern acoustical technology also brings benefits: it is quite probable that the availability (and judicious use) of audiophile equipment has enabled many of us, if we are so inclined, to hear more musical performances than Beethoven, Mozart or even the long-lived Haydn could have heard during their respective lifetimes. Ultrasonic devices are being used to: dislodge dental plaque; overcome the effects of arteriosclerosis by freeing up clogged blood vessels; provide noninvasive medical diagnoses; aid in surgical procedures; supply a means of nondestructive testing of materials; and clean nearly everything from precious stones to silted conduits. The relatively new technique of active noise cancellation utilizes computerized sensing to duplicate the histograms of offending sounds but at 180 degrees out of phase, which effectively counteracts the noise. This technique can be applied to aircraft to lessen environmental impact and to automobiles to provide quieter interiors.

2.2 Sound Generation and Propagation

Sound is a mechanical disturbance that travels through an elastic medium at a speed characteristic of that medium. Sound propagation is essentially a wave phenomenon, as with the case of a light beam. But acoustical phenomena are mechanical in nature, while light, X rays, and gamma rays occur as electromagnetic phenomena. Acoustic signals require a mechanically elastic medium for propagation and therefore cannot travel through a vacuum. On the other hand, the propagation of an electromagnetic wave can occur in empty space. Other types of wave phenomena include those of ocean movement, the oscillations of machinery, and the quantum mechanical equivalence of momenta as propounded by de Broglie.[1]

Consider sound as generated by the vibration of a plane surface at $x = 0$ as shown in Figure 2.3. The displacement of the surface to the right, in the $+x$ direction, causes a compression of a layer of air immediately adjacent to the surface, thereby causing an increase in the density of the air in that layer. Because the pressure of that layer is greater than the pressure of the undisturbed atmosphere, the air molecules in the layer tend to move in the $+x$ direction and compress the second layer which, in turn, transmits the pressure impulse to the third layer and so on. But as the plane surface reverses its direction of vibration, an opposite effect occurs. A rarefaction of the first layer now occurs, and this rarefaction decreases the pressure to a value below that of the undisturbed atmosphere. The molecules from the second layer now tend to move leftward, in the $-x$ direction, and a rarefaction impulse now follows the previously generated compression impulse.

This succession of outwardly moving rarefactions and compressions constitutes a wave motion. At a given point in the space, an alternating increase and decrease in pressure occur, with a corresponding decrease and increase in the density. The spatial distance λ from one point on the cycle to the corresponding point on the next cycle is the *wavelength*. The vibrating molecules that transmit the waves do not, on the average, change their positions, but are merely moved back and forth under the influence of the transmitted waves. The distances these particles move about their respective equilibrium positions are referred to as *displacement amplitudes*. The velocity at which the molecules move back and forth is termed *particle velocity*, which is not to be confused with the *speed of sound*, the rate at which the waves travel through the medium.

The speed of sound is a characteristic of the medium. Sound travels far more rapidly in solids than it does in gases. At a temperature of 20°C sound moves at the rate of 344 m/s (1127 ft/s) through air at the normal atmospheric pressure of

---

[1] The de Broglie theory assigns the nature of a wave to the momentum of a particle of matter in motion in the following way:

$$mv = \frac{hv}{c}$$

where mv represents the moment of the particle, h Planck's constant $= 6.625 \times 10^{-27}$ erg s, c the velocity of light $= 3 \times 10^8$ m/s, and v the radial frequency of the wave attributable to the particle.

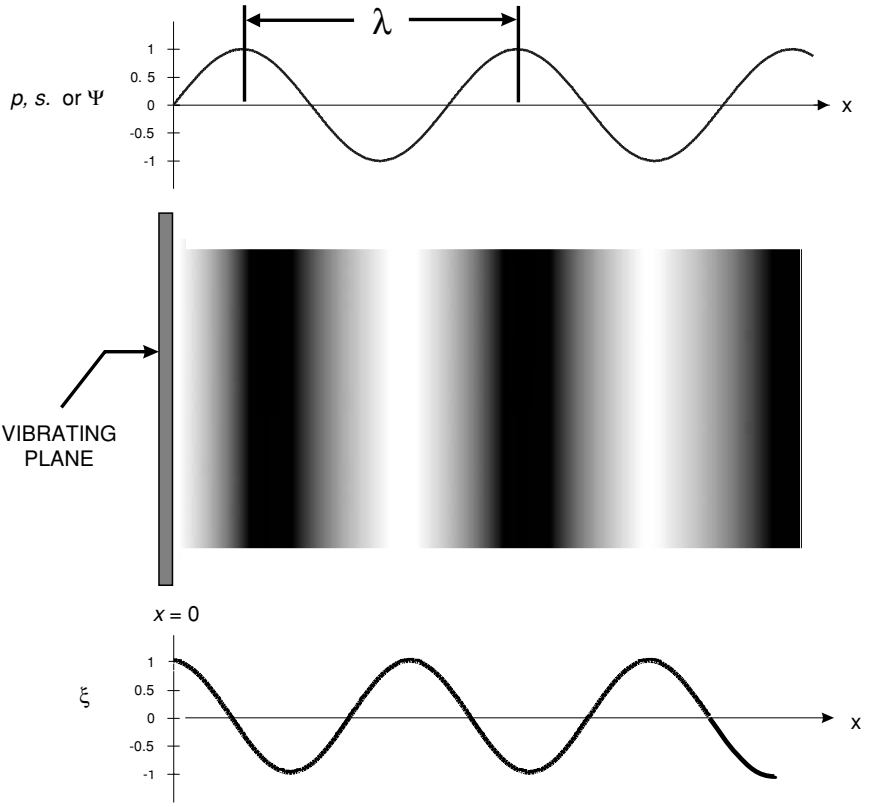


FIGURE 2.3. Depiction of rarefaction and condensation of air molecules subjected to the vibrational impact of a plane wall located at $x = 0$. The degree of darkness is proportional to the density of molecules. Lighter areas are those of rarefactions. Mini-plots of the local variations of molecular displacement Ψ, pressure p, condensation $s = (\rho - \rho_0)/\rho_0$, and particle displacement ξ are given as functions of x for a given instant of the sound propagation. Note that wavelength λ represents the distance between corresponding points of adjacent cycles.

101 kPa (14.7 psia or 760 mmHg). Sound velocities are also greater in liquids than in gases, but remain less in order of magnitude than those for solids. For an ideal gas the velocity c of a sound wave may be computed from

$$c = \sqrt{\frac{\gamma p}{\rho}} = \sqrt{\gamma RT} \tag{2.2}$$

where γ is the gas constant equivalent to the thermodynamic ratio of specific heats, c_p/c_v, p the quiescent gas pressure, and ρ the density of the gas. R is the thermodynamic constant characteristic of the gas and T is the absolute temperature

of the gas. For air at 20°C, the sound propagation speed c is found from

$$c = \sqrt{\gamma RT} = \sqrt{1.4\,[287\ \text{N} \cdot \text{m}/(\text{kg K})](20 + 273.2)\,\text{K}} = 343.2\ \text{m/s}$$

A simple relation such as Equation (2.2) does not exist for acoustic velocity in liquids, but the propagation velocity does depend on the temperature of the liquid and, to a lesser degree, on the pressure. Sound velocity is approximately 1461 m/s in deaerated water. For a solid the propagation speed can be found approximately from

$$c = \sqrt{\frac{E}{\rho}} \tag{2.3}$$

where E represents the Young's modulus (or modulus of elasticity) of the material and ρ the material density. As an example, considered cast iron with a specific gravity of 7.70 and a modulus of elasticity of 105 GPa. Applying Equation (2.3) and recalling that 1 N is equal to 1 kg m/s^2, we find that

$$c = \sqrt{\frac{105(10)^9\ \text{N/m}^2}{7700\,\text{kg/m}^2}} = 3692\,\text{m/s}$$

which does represent the propagation speed of sound in that material. Appendix A lists the speed of sound for a variety of materials.

The strength of an acoustic signal, as exemplified by *loudness* or *sound pressure level* (SPL), directly relates to the magnitudes of the displacement amplitudes and pressure and density variations, as we shall see later in Chapter 3.

When the procession of rarefactions and condensations occurs at a steady sinusoidal rate, a single constant frequency f occurs. If the sound pressure of a pure tone was plotted against distance for a given instant, the wavelength λ can be established as being the peak-to-peak distance between two successive waves. The wavelength λ is related to frequency f by:

$$\lambda = \frac{c}{f} \tag{2.4}$$

where c represents the propagation speed. From Equation (2.4), it can be seen that higher frequencies will result in shorter wavelengths in a given propagation medium.

2.3 Thermodynamic States of Fluids

In the treatment that follows this section, we eschew the details of molecular motion and intermolecular forces by describing relevant effects in terms of macroscopic thermodynamic variables: pressure p, density ρ, and absolute temperature T. These variables relate to each other through an equation of state

$$p = p(\rho, T)$$

which is usually established experimentally. The implication of the equation of state is that only two of the variables are independent; this is to say if the values of two of the independent thermodynamic variables are given for a fluid, the specific value of any other thermodynamic property is automatically established. The equation of state for an ideal gas,

$$\frac{p}{\rho} = RT \tag{2.5}$$

can be derived from simple kinetic theory. Here,

R = gas constant, energy per unit mass per degree
$R = \Re/\mathbf{M}$
$\Re$ = universal gas constant, energy per mole per degree
= 8.314.3 kJ/kg mol K = 1545.5 ft lb_f/lb mol R
= 1.986 Btu/lb_mmol R
$\mathbf{M}$ = molecular weight of gas, kg/kg mol or lb_m/lb_m mol

Each kilogram-mole of the gas contains $N_0 = 6.02 \times 10^{26}$ molecules. N_0 constitutes Avogadro's number for the MKS system of dimensional units. With $\widehat{\eta}$ representing the mass of a single-gas molecule, $\mathbf{M} = N_0\widehat{\eta}$, the number of molecules per unit volume is $N = \rho/\widehat{\eta}$. The equation of state for the ideal gas can now be rewritten as:

$$p = N\frac{\Re}{N_0}T = NkT$$

where k is the Boltzmann constant $= \Re/N_0 = 1.38 \times 10^{-26}$ kJ/K.

In liquids and gases under extreme pressures, the relationships between the thermodynamic variables p, T, ρ, X (here X is the quality or the fractional mass of gas comprising a saturated liquid–gas mixture, e.g., $X = 1.00$ represents a fully saturated gaseous state and $X = 0$ represents the fully saturated liquid state) are not so simple, but the fact remains that these parameters are fully dependent upon each other, and specifying two thermodynamic parameters (including enthalpy, entropy, etc.) will fully specify the thermodynamic state of the fluid.

2.4 Fluid Flow Equations

In the Eulerian description of fluid mechanics the field variables, such as pressure, density, momenta, and energy, are considered to be continuous functions of the spatial coordinates x, y, z and of time t. Because velocity has three components in three-dimensional space and only two independent thermodynamic variables need to be selected to fix the thermodynamic state of the fluid (we chose p and ρ), we have a total of five field variables for which we need five independent equations. We can take advantage of conservation laws to establish these equations, namely the conservation of mass, which supplies one equation; the conservation of momentum along each of the three principal axes, which provides three equations;

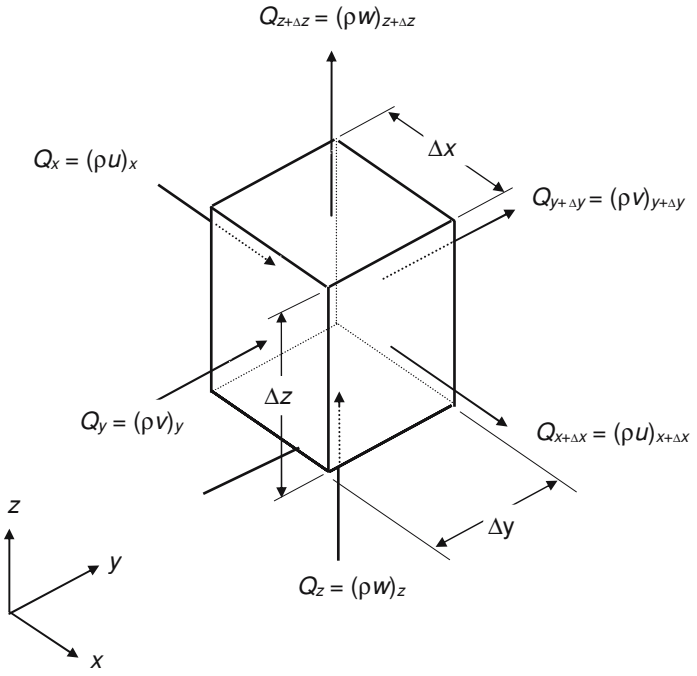


FIGURE 2.4. Flow $Q(\mathbf{v}, t)$ into and out of a control volume $\Delta V = \Delta x \Delta y \Delta z$ depicted for the derivation of the equation of continuity.

and the conservation of energy (or the equation of state, in the derivation of the actual wave equation)[2] that constitutes the fifth equation.

2.5 Conservation of Mass

In Figure 2.4, consider a parallelepiped serving as a control (or reference) volume, $dV = dx\, dy\, dz$, through which fluid flows. Conservation of matter dictates that the net flow into this volume equals the gain or loss of fluid inside the volume, i.e.,

$$m_{\text{exit}} - m_{\text{enter}} = \left(\frac{\Delta m}{\Delta t_{\text{volume}}} \right)_{\Delta V \to dv}$$

Let the velocity $\mathbf{V}$ of the fluid resolve into u, v, w, the velocity components in the x, y, and z directions, respectively. In vector terminology

$$\mathbf{V} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k}$$

[2] It can be argued that because the equation of state derives from the principles of conservation of momentum and energy in classic kinetic theory, it effectively becomes the equivalent of the energy conservation principle in the extraction of the acoustic wave equations for a fluid, in conjunction with the equations of continuity and momentum.

where **i**, **j**, **k** represent the unit vectors along the x, y, z coordinates. The mass flux $Q(x, t)$ is defined as the flow of the mass of fluid per unit time per unit area, which is represented by

$$\mathbf{Q}(\mathbf{x}, t) = \rho(\mathbf{x}, t)\, u(\mathbf{x}, t)$$

The rate of mass per unit time flowing into the control volume dV in the direction is given by

$$\dot{m}_x = Q(x, t)dA_x = (\rho u)_{x,t} dA_x = (\rho u)_{x,t} dy\, dz \tag{2.6}$$

at position x and the rate of mass per unit time $\dot{m}_{x+\Delta x}$ leaving dV at $x + \Delta x$ by

$$\dot{m}_{x+\Delta x} = Q(x + \Delta x, t)dA_x = (\rho u)_{x,t} dA_x (\rho u)_{x,t} dy dz \tag{2.7}$$

Then subtracting Equation (2.6) from Equation (2.7) yields the net flow in the x direction

$$\dot{m}_{x+\Delta x} - \dot{m}_x = dA_x[(\rho u)_{x+\Delta x} - (\rho u)_x] = dA_x \frac{\partial(\rho u)}{\partial x} dx = \frac{\partial(\rho u)}{\partial x} dV \tag{2.8}$$

Similarly for mass flow in the$y$ and z directions

$$\dot{m}_{y+\Delta y} - \dot{m}_y = \frac{\partial(\rho v)}{\partial y} dV \tag{2.9}$$

$$\dot{m}_{z+\Delta z} - \dot{m}_z = \frac{\partial(\rho w)}{\partial z} dV \tag{2.10}$$

Summing the net mass flows Equations (2.8)–(2.10) and equating them to the change of mass in the control volume:

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = -\frac{\partial \rho}{\partial t} \tag{2.11}$$

Equation (2.11) is the *equation of continuity*, a general statement of the conservation of matter for compressible fluid[3] flow. In vector notation Equation (2.11) may be written as

$$\frac{\partial \rho}{\partial t} - \nabla \cdot (\rho \mathbf{V}) = 0 \tag{2.12}$$

wherein the gradient symbol represents

$$\nabla = \mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z}$$

[3] If density ρ is constant, the fluid is said to be *incompressible*. As ρ is no longer a spatial or a time function, Equation (2.11) simplifies to:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

in the rectangular coordinate system. In the cylindrical coordinate system the gradient operator ∇ appears as

$$\nabla = \frac{\partial}{\partial r} + \frac{1}{r}\frac{\partial}{\partial \phi} + \frac{\partial}{\partial z}$$

and in the spherical coordinate system as

$$\nabla = \frac{\partial}{\partial r} + \frac{1}{r}\frac{\partial}{\partial \vartheta} + \frac{1}{r\sin\vartheta}\frac{\partial}{\partial \phi}$$

2.6 Conservation of Momentum

In order to develop the equations of momentum for a fluid, let us consider the motion of a fluid particle with a velocity field $\mathbf{V_p} = \mathbf{V}_t(x, y, z, t)$. At a later time $t + dt$, the velocity becomes $\mathbf{V_{p'}} = \mathbf{V}_{t+dt}(x + dx, y + dy, z + dz, t + dt)$. The change in velocity is given by:

$$d\mathbf{V} = \Delta(\mathbf{V}_{p'} - \mathbf{V}_p) = \frac{\partial \mathbf{V}}{\partial x}dx + \frac{\partial \mathbf{V}}{\partial y}dy + \frac{\partial \mathbf{V}}{\partial z}dz + \frac{\partial \mathbf{V}}{\partial t}dt$$

and the total acceleration of the particle is therefore expressed as

$$\mathbf{a}_p = \frac{d\mathbf{V}}{dt} = \frac{\partial \mathbf{V}}{\partial x}\frac{dx}{dt} + \frac{\partial \mathbf{V}}{\partial y}\frac{dy}{dt} + \frac{\partial \mathbf{V}}{\partial z}\frac{dz}{dt} + \frac{\partial \mathbf{V}}{\partial t} \tag{2.13}$$

But $dx/dt = u$, $dy/dt = v$, and $dz/dt = w$. Equation (2.13) now can be written as:

$$\mathbf{a}_p = \frac{d\mathbf{V}}{dt} = u\frac{\partial \mathbf{V}}{\partial x} + v\frac{\partial \mathbf{V}}{\partial y} + w\frac{\partial \mathbf{V}}{\partial z} + \frac{\partial \mathbf{V}}{\partial t} = \frac{D\mathbf{V}}{Dt} \tag{2.14}$$

Here, the operator

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z}$$

represents the *total* or *convective* derivative of fluid mechanics. While Equation (2.14) is a vector expression, we can rephrase it into scalar terms. With reference to a rectangular coordinate system the scalar components of Equation (2.14) are written as

$$a_x = \frac{Du}{Dt} = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}$$
$$a_y = \frac{Dv}{Dt} = \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}$$
$$a_z = \frac{Dw}{Dt} = \frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}$$

No acceleration or deceleration of the fluid will occur unless forces are acting upon it. Two types of forces act on the fluid element as shown in Figure 2.5, namely, body forces and surface forces. Gravity constitutes a body force that pervades throughout the volume of the fluid. Surface forces include both normal forces

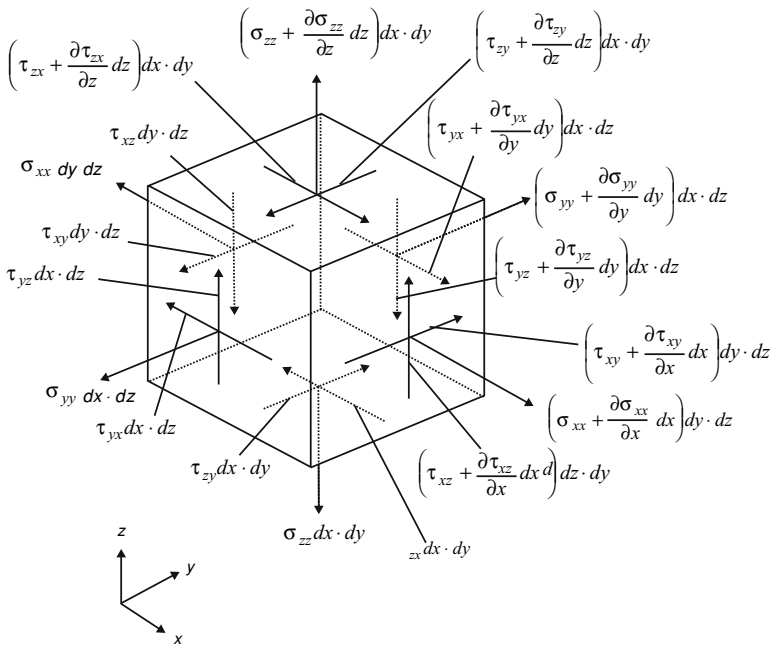


FIGURE 2.5. A fluid element acted on by normal and tangential stresses.

(pressure) and tangential (shear) forces. A normal force is denoted by the symbol σ_{mm}, where m denotes the direction of the normal. Because σ_{mm} is dimensionally expressed in force per unit area, it must be multiplied by the area normal to it in order to obtain the force.

A shear stress acts along the plane of the surface. It is represented by the symbol τ_{mn}, where the force produced by the shear is normal to coordinate m and parallel for coordinate n, and either m or n may represent the principal coordinate x, y, or z, provided that $m \neq n$. If $m = n$, then τ_{mm} really represents the normal force σ_{mm} and thus is no longer a tangential force. The shear stress is multiplied by the area it is acting on to yield the shear force. For example, a shear τ_{xy} multiplied by area ($dx\ dy$) represents the shear force normal to the x-axis and parallel to the y-axis, as shown in Figure 2.5 for a fluid element displayed in Cartesian coordinates.

In order to determine the net force F_x in the x-direction, all of the forces in the x-direction must be summed. From Figure 2.5 we can write

$$dF_x = \rho g_x dxdydz + \left(-\sigma_{xx} + \sigma_{xx} + \frac{\partial \sigma_{xx}}{\partial x} dxdydz\right) + \left(\tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} dx - \tau_{yx}\right) dxdz + \left(\tau_{zx} + \frac{\partial \tau_{zx}}{\partial z} dx - \tau_{zx}\right) dxdz$$

which simplifies to

$$dF_x = \left(\rho g_x + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}\right) dxdydz \tag{2.15}$$

We can easily apply the force summation procedure to the other two principal axes:

$$dF_y = \left(\rho g_y + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z}\right) dxdydz \tag{2.16}$$

$$dF_z = \left(\rho g_z + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z}\right) dxdydz \tag{2.17}$$

From Newton's second law of motion,

$$d\mathbf{F} = d(m\mathbf{a}) = \rho d\mathrm{V}\frac{D\mathbf{V}}{Dt}$$

we can now formulate the differential momentum equations by combining the scalar components of Equation (2.13) with Equations (2.15)–(2.17) with the following results:

$$\rho g_x + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} = \rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial u}{\partial z}\right) \tag{2.18a}$$

$$\rho g_y + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} = \rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) \tag{2.18b}$$

$$\rho g_z + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} = \rho\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) \tag{2.18c}$$

In order to use Equations (2.18a)–(2.18c), the expressions for the stresses should be stated in terms of the velocity field. If a Newtonian fluid is assumed, the viscous stresses are proportional to the rate of shearing strain (i.e., the rate of angular deformation). Without going into details, we express the stresses in terms of velocity gradients and viscosity coefficient μ as follows:

$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) \tag{2.19a}$$

$$\tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) \tag{2.19b}$$

$$\tau_{zx} = \tau_{xz} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \tag{2.19c}$$

$$\sigma_{xx} = -p - \frac{2}{3}\mu\Delta \cdot \boldsymbol{V} + 2\mu\frac{\partial u}{\partial x} \tag{2.19d}$$

$$\sigma_{yy} = -p - \frac{2}{3}\mu\Delta \cdot \boldsymbol{V} + 2\mu\frac{\partial v}{\partial y} \tag{2.19e}$$

$$\sigma_{zz} = -p - \frac{2}{3}\mu\Delta \cdot \boldsymbol{V} + 2\mu\frac{\partial w}{\partial z} \tag{2.19f}$$

Here the term p is the local thermodynamic pressure, which is essentially an isotropic parameter at any given point in the fluid. If we assume the fluid to be frictionless, then $\mu = 0$, and we are left with Equations (2.19d)–(2.19f) in the

following format:

$$\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = -p$$

and, neglecting the gravitational body force ρg_i (where $i = x, y, z$), we recast Equations (2.18a)–(2.18c) as

$$-\frac{\partial p}{\partial x} = \rho \frac{Du}{Dt}$$
$$-\frac{\partial p}{\partial y} = \rho \frac{Dv}{Dt}$$
$$-\frac{\partial p}{\partial z} = \rho \frac{Dw}{Dt}$$

2.7 Conservation of Energy

The energy content W of a fluid is the sum of the macroscopic kinetic energy $\rho|V|^2/2$ and the internal energy ρE of the fluid. In a gas, the microscopic kinetic energy (i.e., the thermal energy of the molecules) comprises the major portion of the internal energy, so the potential energy between molecules is negligible in comparison. Denoting the energy flux by S we write equation for the conservation of energy as

$$\frac{\partial W}{\partial t} + \frac{\partial S}{\partial x} = 0 \tag{2.20}$$

The internal energy of a volumetric element can be increased through heat flow from the surrounding fluid or from external sources and by the work of compression $-\int pdV$ by the surrounding fluid pressure. This energy balance and the fact that the internal energy is a thermodynamic state that can be fully specified by two independent thermodynamic variables constitute the first law of thermodynamics.

With the conservation equations discussed above and the equation of state, we have all the necessary equations to obtain solutions for the three components of velocity $\mathbf{V}$, ρ, p and absolute temperature T. Because the fluid equations are nonlinear, solutions are not easy to come by, even with the aid of supercomputers to map the complex motions of atmospheric eddies, turbulent jet flows, capillary flow, and so on. Exact solutions exist principally for a few simple problems. Nevertheless, through the derivation of these equations, we have established the foundation for the derivation of acoustic field equations for fluids.

2.8 Derivation of the Acoustic Equations

We begin with the following assumptions:

(1) the unperturbed fluid has definite values of pressure, density, temperature, and velocity, all of which are assumed to be time independent and denoted by the subscript 0.

(2) the passage of an acoustic signal through the fluid results in small perturbations of pressure, temperature, density, and velocity. These perturbations are expressed as $p_0 + p$, $\rho_0 + \rho$, u, and so on. The unperturbed velocity u_0 is set to zero; the unperturbed fluid does not undergo macroscopic motion, and u constitutes the perturbation velocity in the x-direction. Also, $p \ll p_0$, $\rho \ll \rho_0$, and $T \ll T_0$.
(3) the transmission of the sound through the fluid results in low values of spatial temperature gradients at audio frequencies, resulting in almost no heat transfer between warmer and cooler regions of the plane wave. Thus the ongoing thermodynamics process may be considered an adiabatic process (at ultrasonic frequencies there is virtually no time for heat transfer to occur).

Under the above conditions we obtain an expansion of the continuity equation in the x-direction as follows:

$$\frac{\partial[(\rho_0 + \rho)u]}{\partial x} = -\frac{\partial(\rho_0 + \rho)}{\partial t} = u\frac{\partial \rho}{\partial x} + \rho_0 \frac{\partial u}{\partial x}$$

which, by discarding second-order terms, reduces to

$$\frac{\partial \rho}{\partial t} = -\rho_0 \frac{\partial u}{\partial x} \tag{2.21}$$

Here we consider $\rho_0 \approx \rho_0 + \rho$, also recalling that the quiescent density ρ_0 does not vary in time and space. Treating in the same fashion the one-dimensional momentum equation

$$\frac{\partial(p_0 + p)}{\partial x} = (\rho_0 + \rho)\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x}\right)$$

yields

$$\frac{\partial p}{\partial x} = \rho_0 \frac{\partial u}{\partial t} \tag{2.22}$$

In an adiabatic process involving an ideal gas,

$$p\rho^{-\gamma} = \text{constant.}$$

Here γ represents a thermodynamic constant, characteristic of the gas, equal to the ratio of the specific heats c_p/c_v. The numerator of this thermodynamic ratio is the specific heat at constant pressure, and the denominator, the specific heat at constant volume. By differentiation,

$$\rho^{-\gamma} dp - \gamma p \rho^{-\gamma - 1} d\rho = 0$$

and rearranging

$$\frac{dp}{p} = \gamma \frac{d\rho}{\rho}$$

we have for this situation

$$\frac{dp}{p_0} = \gamma \frac{d\rho}{\rho_0}$$

The above expression can be differentiated with respect to time:

$$\frac{1}{p_0}\frac{\partial p}{\partial t} = \frac{\gamma}{\rho_0}\frac{\partial \rho}{\partial t} \tag{2.23}$$

Combining Equations (2.21) and (2.23),

$$\frac{\partial p}{\partial t} = \frac{\gamma p_0}{\rho_0}\frac{\partial \rho}{\partial t} = \gamma p_0 \frac{\partial u}{\partial x}$$

and then differentiating with respect to time t we obtain

$$\frac{\partial^2 p}{\partial t^2} = \gamma p_0 \frac{\partial^2 u}{\partial t \partial x}$$

Differentiating Equation (2.22) with respect to x results in

$$\frac{\partial^2 p}{\partial x^2} = \rho_0 \frac{\partial^2 u}{\partial x \partial t}$$

Equating the above two cross-differential terms to each other, as we consider them to be equivalent regardless of their order of differentiation, we obtain the result

$$\frac{\partial^2 p}{\partial x^2} = \frac{\rho_0}{\gamma p_0}\frac{\partial^2 p}{\partial t^2} = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} \tag{2.24}$$

where

$$c^2 = \frac{\gamma p_0}{\rho_0} = \gamma R T$$

Here c, R, and T are respectively the propagation velocity of sound, the gas constant, and absolute temperature of the (ideal) gas. In three-dimensional form the wave equation (2.17) appears as

$$\nabla^2 p = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} \tag{2.25}$$

We also could have eliminated p in favor of u by reversing the differentiation procedure between Equations (2.22) and (2.23), in which situation we would get

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} \tag{2.26}$$

for the one-dimensional situation, and

$$\nabla^2 \mathbf{V} = \frac{1}{c^2}\frac{\partial^2 \mathbf{V}}{\partial t^2} \tag{2.27}$$

in the three-dimensional case. It is also a straightforward matter to derive the wave equation in terms of density, resulting in the following expressions:

$$\frac{\partial^2 \rho}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 \rho}{\partial t^2} \tag{2.28a}$$

for the one-dimensional case and

$$\nabla^2 \rho = \frac{1}{c^2}\frac{\partial^2 \rho}{\partial t^2} \tag{2.28b}$$

for three dimensions.

Equations (2.24)–(2.28) are second-order partial differential equations in x and t. Ordinarily we need two initial conditions and two boundary conditions for a fully defined solution for each of the equations, but we need not define these conditions in order to ascertain the nature of the general solutions. The general solution to Equation (2.24) may be written as

$$p(x, t) = F(x - ct) + G(x + ct) \tag{2.29}$$

The function $F(x - ct)$ represents waves moving in the positive x-direction and $G(x + ct)$ represents waves moving in the opposite direction. All solutions to Equation (2.24) must be of the form represented in Equation (2.29); otherwise any p that does not adhere to this form cannot constitute a solution. Because Equations (2.26) and (2.28a) are functionally the same as Equation (2.24), their respective general solutions take on the same cast as that of Equation (2.29):

$$u(x, t) = \Phi(x - ct) + \Gamma(x + ct) \tag{2.30}$$

$$\rho(x, t) = \Theta(x - ct) + Y(x + ct) \tag{2.31}$$

The arbitrary functions $F, G, \Phi, \Gamma, \Theta, Y$ can be assumed to have continuous derivatives of the first and second order. Because of the manner in which the constant c appears in relation to x and t inside these functions, it must have the physical dimensions of x/t, so c must be a speed, which is indeed the experimentally determined rate at which the sound wave is propagated through a medium. No matter how it is shaped, the propagating wave (or its counterpart, the backward traveling wave) moves without changing its form. To prove this, consider the sound pressure level at $x = 0$ and time $t = t_1$ for a wave moving in the positive x-direction. Thus $p = f_{\hat{a}}(t_1)$. At time $= t_1 + t_2$, the sound wave will have traveled a distance $x = ct_2$. The sound pressure will now be

$$p = f_\alpha(t_1 + t_2) = f_\alpha\left(t_1 + t_2 - \frac{ct_2}{t_2}\right) = f_\alpha(t_1)$$

This means the sound pressure has propagated without change.

References

Beranek, Leo L. 1986. *Acoustics*. New York: American Institute of Physics. (An exceptionally clear text in the field.)

Crocker, Malcolm J. (ed.). 1997. *Encyclopedia of Acoustics*, Vol. 1. New York: John Wiley & Sons, Chapters 1 and 2. (Sir James Lighthill compared the significance of this four-volume compilation with that of Lord Rayleigh's *The Theory of Sound*, which is not at all far-fetched considering that this encyclopedia contains contributions from an editorial board whose members constitute a veritable *Who's Who in Acoustics*. *Handbook of Acoustics* by the same editor and publisher (1998) is a truncated version of the *Encyclopedia*, containing approximately 75 percent of the chapters. Chapters 1 and 2 are identical in both publications.)

Shapiro, Ascher H. 1953. *The Dynamics and Thermodynamics of Compressible Fluid Flow*, Vol. 1. New York: The Ronald Press Co., Chapter 1. (In spite of its venerable age, it is still one of the best works on the topic of fluid dynamics.)

Problems for Chapter 2

1. Write the expression for a simple sine wave having a frequency of 10 Hz and an amplitude of 10^{-8} cm. What is the frequency expressed in radians per second? Plot the expression on graph paper or, better yet, on a computer with the aid of a program such as Excel©, Mathcad©, MathLab©, etc.
   Repeat the process for a frequency of 20 Hz and for 50 Hz.
2. If the frequency of a pure cosine wave is 100 Hz and the velocity of the wave front is 330 m/s, what is the wavelength of this signal? Express the frequency in radians per second.
3. Air may be considered to be a nearly ideal gas with the ratio of specific heat $\gamma = 1.402$. At 0°C its density is 1.293 kg/m^3. Predict the speed of sound c for the normal atmospheric pressure of 101.2 kPa (1 Pa = 1 N/m^2).
4. Nitrogen is known to have a molecular weight of 28 kg/kg mol. Predict the speed of sound at 0°C, 20°C, and 50°C, with the assumption that nitrogen behaves as an ideal gas. Repeat the problem for pure oxygen which has a molecular weight of 32.
5. Compute the speed of sound (in ft/s) traveling through steel that has a Young's modulus of 30×10^6 psi and a specific gravity of 7.7. Why does it differ from cast iron?
6. A solid material is known to have a density of 8.5 g/cm^3. Sound velocity traveling through this material was measured as being 4000 m/s. Determine the Young's modulus in GPa for this material.
7. Find the speed of sound (in m/s) traveling through aluminum that typically has a Young's modulus of 72.4 GPa and a specific gravity of 2.7.
8. For distilled water, the speed of sound c in m/s can be predicted within 0.05% as a function of pressure P and temperature T from the experimentally determined formula

   $$c(P,t) = 1402.7 + 488t - 482t^2 + 135t^3 + (15.9 + 2.8t + 2.4t^2)(P_a/100)$$

   where P_a is the gauge pressure in bars (1 bar = 100 kPa) and $t = 0.01T$, with temperature T in degrees Celsius. Find the speed of sound for the water at 20°C and 1 bar. What will be the wavelength of a 200-Hz sine-wave signal traveling through water? If the same signal travels through air at the speed of 331 m/s, what will be the corresponding wavelength?
9. Explain why density and pressure are in phase and that both are out of phase with particle velocity.
10. When does the maximum amplitude of a pure sine wave occur with respect to the particle velocity and the instantaneous pressure?

11. A molecule exposed to a pure cosine sound wave undergoes a particle displacement y, with maximum amplitude A, according to

$$y = A\cos\omega t$$

Find the corresponding particle velocity and show that the expressions for both displacement and velocity constitute solutions to a wave equation.

12. Demonstrate that $y(x, t) = A_1\cos(x - ct) + B_1\sin(x + ct) + A_2\cos^2 2(x - ct) + B_2\sin^2 3(x + ct)$ constitutes a solution to the wave equation. Which portion of the solution represents wave travel in the $+x$ direction and which portion denotes propagation in the $-x$-direction?

13. If the density of a medium subject to wave propagation varies in the following manner:

$$\rho = \rho_0 e^{i(x-ct)}$$

express the corresponding pressure $p(x,t)$ in terms of quiescent pressure p_0 and density ρ_0.

3
Sound Wave Propagation and Characteristics

3.1 The Nature of Sound Propagation

When energy passes through a medium resulting in a wave-type motion, several different types of waves may be generated, depending upon the motion of a particle in the medium. A *transverse* wave occurs when its amplitude varies in the direction normal to the direction of the propagation. This type of wave has been used to describe the transmission of light and alternating electric current. But the situation is almost completely different in the case of sound waves, which are principally *longitudinal*, in that the particles oscillate back and forth in the direction of the wave motion, with the result the motion creates alternative compression and rarefaction of the medium particles as the sound passes a given point. The net fluid displacement over a cycle is zero, since it is the disturbance rather than the fluid that is moving at the speed of sound. The fluid molecules do not move far from their original positions.

Additionally, waves may also fall into the category of being rotational or torsional. The particles of a rotational wave rotate about a common center; the curl of an ocean wave roaring onto a beach provides a vivid example. The particles of torsional waves move in a helical fashion that could be considered a vector combination of longitudinal and transverse motions. Such waves occur in solid substances, and shear patterns often result. These are referred to as *shear waves*, which all solids support.

3.2 Forward Propagating Plane Wave

In Equation (2.29), which is the general solution to the one-dimensional wave equation, we consider only the wave moving in the $+x$ direction with the solution for a monofrequency wave represented by

$$p(x, t) = F(x - ct) = p_m \cos k(x - ct) \tag{3.1}$$

where p_m is the peak amplitude of the sound pressure; k, the wave number which equals $2\pi/\lambda$; and λ, the wavelength. Figure 3.1 shows the variation of sound

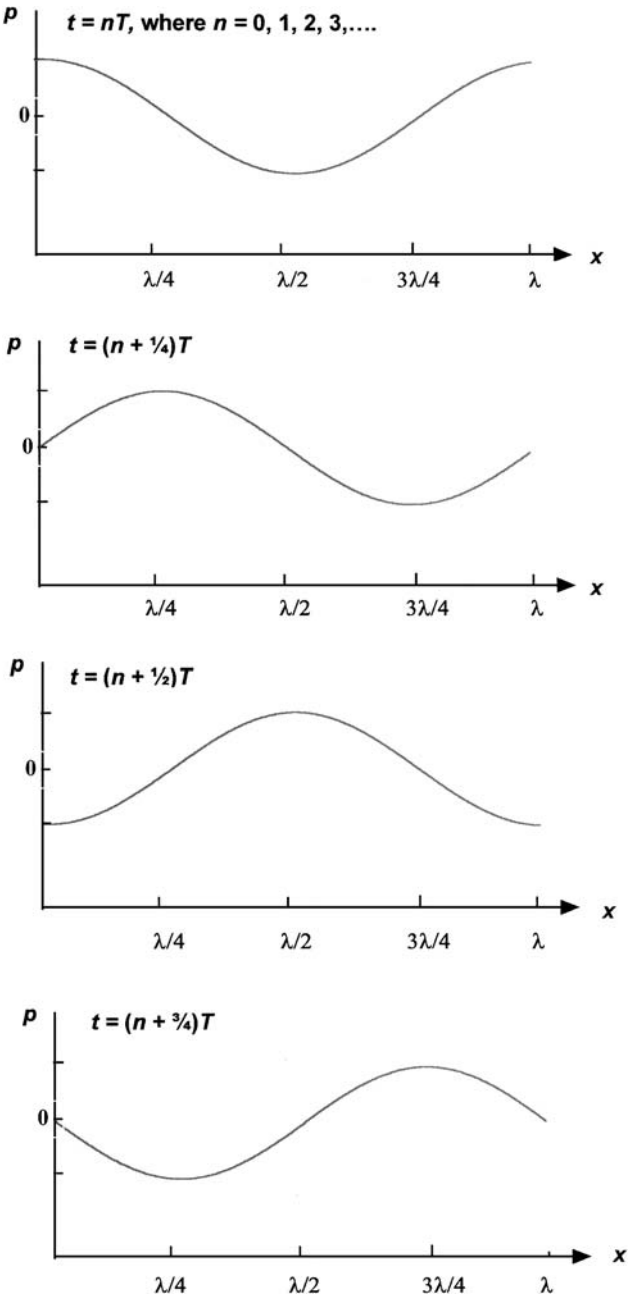


FIGURE 3.1. Variation of pressure from quiescent state along the x-axis for time instants $t = 0, t = T/4, t = T/2, t = 3T/4$, where $T = 1/f$ is the period for a complete cycle to occur.

pressure p for different time intervals $t = 0$, $t = T/4 = 1/(4f) = (1/2)\pi/\omega$, $t = T/2 = 1/(2f) = \pi/\omega$, $t = 3/(4f) = 3T/4 = (3/2)(\pi/\omega)$, and $t = T = 1/f = 2\pi/\omega$.

3.3 Complex Waves

The concept of simple sinusoidal waves lacks specificity to be of practical value in noise control, but complex periodic waveforms can be broken into two or more sinusoidal harmonically related waves. In Figure 3.2, a complex waveform resolves into a sum of harmonically related waves. The harmonic relationship in this example is such that the frequency of one harmonic is twice that of the other. The lowest-frequency sine term is the fundamental, and the next highest frequency the second harmonic, the next the third harmonic, and so on. Sound pressure waves radiating from pumps, gears, and other rotating machinery are usually complex and periodic, with distinguishable discrete tones or pure tones. These sinusoidal waves can be broken down or synthesized into simple sinusoidal terms. In the analysis of the noise emanating from rotating machinery, there are often 8 to 10 harmonics present with frequencies which are integer-ordered multiples of the fundamental frequency. Even aperiodic sounds such as the hiss of a pressure valve of an autoclave, the broadband whine of a jet engine, or the pulsating sound of a jackhammer can be resolved and described in terms of sums of simple sinusoids. Integer harmonic relationships associated with periodic sound waves do not occur in these sounds, and the composition entails more than a simple series. But the principle of synthesis still applies.

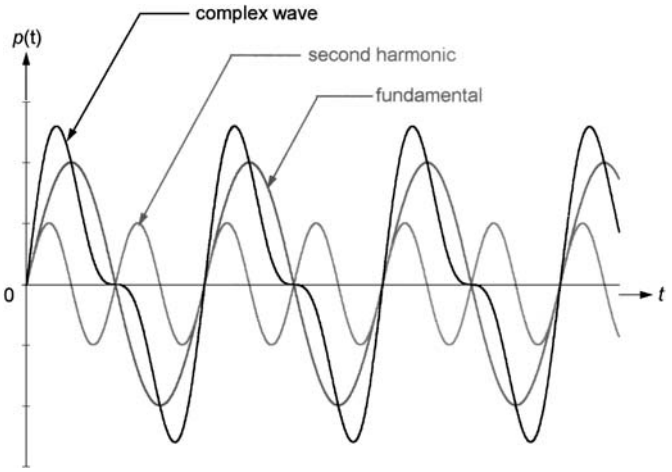


FIGURE 3.2. The resolution of a complex waveform into a set of harmonically related sinusoidal waves. The fundamental wave and the second harmonic sine wave add up algebraically to form the complex wave.

A general equation can be written to incorporate the elements of the sound pressure level associated with complex periodic noise sources as

$$\begin{aligned} p(t) &= A_1 \sin(\omega t + \phi_1) + A_2 \sin(2\omega t + \phi_2) \\ &\quad + A_3 \sin(3\omega t + \phi_3) + \cdots + A_n \sin(n\omega t + \phi_n) \\ &= \sum_1^n A_n \sin(n\omega t + \phi_n) = \sum_1^n C_n e^{i\omega t} \end{aligned} \tag{3.2}$$

where

A_n = amplitude of the nth harmonic
ϕ_n = phase angle of the nth harmonic
C_n = complex amplitude of the nth harmonic

Equation (3.2) constitutes a form of the Fourier series, an analytical tool developed nearly two centuries ago by the French physicist Jean Baptiste Joseph Fourier to characterize complex functions and used by him to predict tides. Fourier's concept of complex wave synthesis constitutes one of the most powerful analytical and diagnostic tools available to the present-day acoustician.

When two or more sound waves become superimposed upon each other, they combine in a linear manner, i.e., their amplitudes add algebraically at any point in space and time. The superposition generally results in a complex wave that can be synthesized into basic sinusoidal spectrum components. Two special phenomena resulting from superposition are of special interest, namely, *beat frequency* and *standing waves*.

Consider the superposition of two sound waves of equal amplitudes but slightly differing frequencies. With A_0 denoting the amplitude of each wave and $\omega_1 \not\simeq \omega_2$, the total superimposed pressure becomes

$$p(t) = A_0(\sin \omega_1 t + \sin \omega_2 t)$$

Applying the trigonometric identity

$$\sin \alpha + \sin \beta = 2 \cos \frac{(\alpha - \beta)}{2} \sin \frac{(\alpha + \beta)}{2}$$

the total pressure assumes the form

$$\begin{aligned} p(t) &= 2\, A_0 \cos \frac{(\omega_1 - \omega_2)t}{2} \sin \frac{(\omega_1 + \omega_2)t}{2} \\ &= 2\, A_0 \cos 2\pi \frac{(f_1 - f_2)t}{2} \sin 2\pi \frac{(f_1 + f_2)t}{2} \end{aligned} \tag{3.3}$$

where $\omega = 2\pi f$.

From Equation (3.3) the resultant wave may be considered a complex sound wave with a frequency of $(f_1 + f_2)/2$, as indicated by the sine factor and which is the average of the two superimposed waves. The amplitude is

$$p'(t) = 2\, A_0 \cos 2\pi \frac{(f_1 - f_2)t}{2}$$

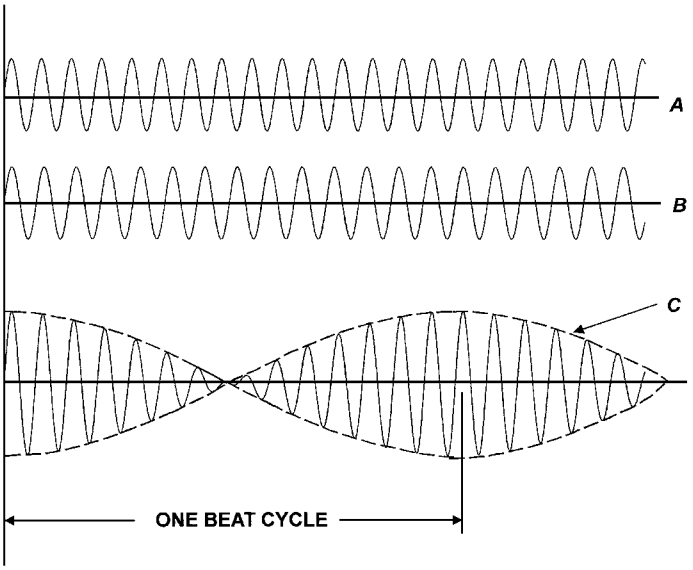


FIGURE 3.3. Addition of waves A and B with equal amplitudes but slightly differing frequencies. The sum of the two sine waves yields an envelope C which has a beat frequency equal to the difference between the frequencies of the superimposed waves A and B.

When the argument of the cosine assumes integer values of π, the amplitude of the complex wave is a maximum that is equal to $2A_0$. Continuing the reasoning further, it is established that the amplitude of the complex wave vanishes when the argument of the cosine takes on integer odd values of $\pi/2$, i.e.,

$$2\pi \frac{(f_1 - f_2)t}{2} = \frac{(2n-1)\pi}{2} \quad (n = 1, 2, 3, \ldots) \tag{3.4}$$

A graph of the envelope of this transient amplitude modulation is given in Figure 3.3. The modulation or *beat frequency* is simply the frequency difference $(f_1 - f_2)$ between the two superposed waves. To demonstrate this, let us solve Equation (3.4) for those times t_n when the amplitude of the superimposed sound pressure is zero,

$$t_n = \frac{2n-1}{2(f_1 - f_2)} \quad (n = 1, 2, 3, \ldots)$$

Now consider in a general fashion the time difference between two consecutive beats, namely, the nth and the $(n+1)$th:

$$\begin{aligned} t_{n+1} - t_n &= \frac{2(n+1)-1}{2(f_1 - f_2)} - \frac{2n-1}{2(f_1 - f_2)} \\ &= \frac{1}{f_1 - f_2} \end{aligned} \tag{3.5}$$

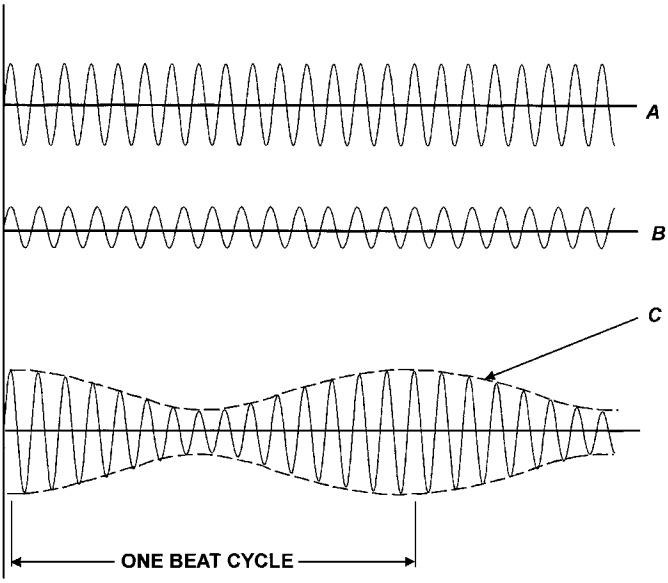


FIGURE 3.4. Addition of waves A and B with slightly different frequencies but unequal amplitudes. Envelope C that results from adding waves A and B shows beats in which the minimum strength is not zero.

The time duration between beats is, by definition, the period T_b of the beat frequency; and the reciprocal of the period defined in Equation (3.5) yields the beat frequency f_b:

$$f_b = f_1 - f_2 \tag{3.6}$$

In the more general case when the amplitudes of the superposed equations are not equal, the amplitude of the superposed wave varies between the sum and difference of the component waves, as shown in Figure 3.4. The periodic variation in amplitude generates a rhythmic pulsating sound, and when the frequency difference is only a few hertz, say 4 or 5, the human ear can readily discern the beat.

3.4 Standing Waves

When a sound wave is superposed upon another wave of the same frequency but traveling in a different direction, a standing-wave sound field is generated. As an illustration, consider the superimposition of two sound waves traveling in opposite directions as given by

$$p_1(t) + p_2(t) = A_1 \sin(2\pi ft - kx) + A_2 \sin(2\pi ft + kx) \tag{3.7}$$

The first sine term in Equation (3.7) represents a sound wave traveling in the positive x-direction with amplitude A_1 and frequency f. The second sine term

represents a sine wave traveling in the negative x-direction with amplitude A_2 and identical frequency f. Using trigonometric identities for the sum and difference of angles, we rewrite Equation (3.7) as follows:

$$p_1(t) + p_2(t) = A_1 \sin 2\pi ft \cos kx - A_1 \cos 2\pi ft \sin kx + A_2 \sin 2\pi ft \cos kx + A_2 \cos 2\pi ft \sin kx$$

For waves of equal amplitudes we obtain the following simplification:

$$p_1(t) = +p_2(t) = 2\,A_1 \cos kx \sin 2\pi ft \tag{3.8}$$

Equation (3.8) may now be considered to be a simple sinusoidal function of time whose amplitude depends on the spatial location x of the observer. When the argument of the cosine assumes odd integer values of $\pi/2$, i.e.,

$$kx = \frac{\pi}{2}, \frac{3\pi}{2}, \frac{5\pi}{2}, \ldots, \frac{(2n-1)\pi}{2} \quad (n = 1, 2, 3, \ldots)$$

the sound pressure vanishes, and there are nodal points in space where no sound occurs. Solution of the preceding equation for x_n yields the spatial locations of these nodes:

$$x_n = \frac{(2n-1)\pi}{2k} \quad (n = 1, 2, 3, \ldots)$$

and since $k = \omega/c = 2\pi/\lambda$, we obtain

$$x_n = \frac{(2n-1)\lambda}{4} \quad (n = 1, 2, 3, \ldots)$$

We thus note that the location of the nodes is simply related to the wavelength of the superimposed waves. By taking the difference between successive nodal locations, it can be demonstrated that the nodes occur every half wavelength, i.e.,

$$x_{n+1} - x_n = \frac{2[(n+1)-1]\lambda}{4} - \frac{(2n-1)\lambda}{4} = \frac{\lambda}{2}$$

We can also establish from Equation (3.8) that the antinodes or points of maximum sound pressure in the standing wave occur when the argument of the cosine assumes integers values of π, i.e.,

$$kx = n\pi \quad (n = 1, 2, 3, \ldots)$$

The amplitude of the antinodes is simply $2A_1$. These antinodes or points of maximum sound pressure are stationary, located halfway between the nodes and spaced one-half wavelength apart.

Example Problem 1

Consider a case where a hydraulic pump radiates a 600-Hz sound wave that is reflected back from a tile wall located at 1 m away from the wall. What is the spacing of the nodes or position of minimum sound for the fundamental tones and its second harmonic?

Solution

We first calculate the wavelength of the first harmonic from the relationship $\lambda = c/f$, with the speed of sound being taken at 344 m/s at room temperature:

$$\lambda = \frac{c}{f} = \frac{344}{600} = 0.573 \text{ m}$$

The spacing between the nodes occurs every half wavelength, or distance intervals of 0.287 m. With the hydraulic pump positioned in close proximity to the wall, the amplitude of the antinodes will be nearly twice as large, as a result of the summation of the radiated and the reflected waves. For the second harmonic at 1200 Hz, the spacing of the nodes will be 0.143 m, since its wavelength is half that of the fundamental. This example points out the necessity for caution in taking measurements in close proximity to highly reflecting surfaces, which can yield highly misleading results. In many situations reflective surfaces are not in close proximity and the amplitudes of reflected waves are relatively small compared to the original waves, so the variation in the amplitudes of the standing waves are correspondingly small and thus can be neglected.

3.5 Huygens' Principle

While originally conceived in the seventeenth century to explain optical phenomena, Huygens' principle applies equally to sound propagation. The principle states that advancing wavefronts can be considered to be point sources of secondary wavelets. Figure 3.5 illustrates the Huygens' construction of a wave front at time

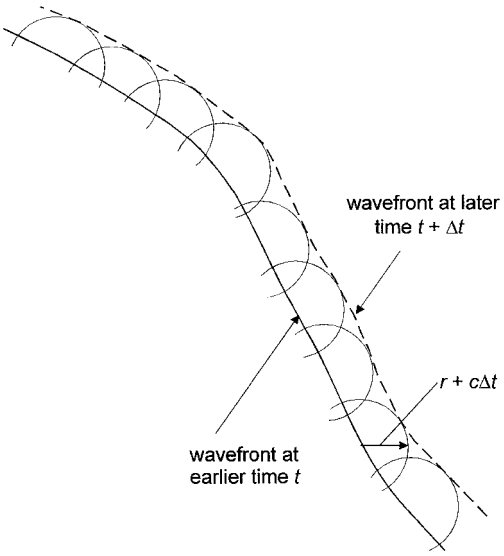


FIGURE 3.5. Construction of a wavefront at time $t + \Delta t$ from its previous state at time t, according to Huygens' principle.

$t + \Delta t$ from a wavefront at earlier time t. The new wavefront is the envelope of radii $c\Delta t$ centered at points on the preceding wavefront. Thus, a plane wave remains a plane wave and a spherical wave remains a spherical wave with ever-enlarging radius.

3.6 The Doppler Effect

When a sound source moves, the acoustic radiation pattern changes, thereby producing changes to the generated frequencies as perceived by a stationary observer. When an acoustic source generating a frequency f approaches an observer at velocity v, then during a single period T $(=1/f)$ a signal emitted at the onset of the period travels a distance cT. But the signal is emitted at the end of the period from the source that is closer to the observer by a distance vT. The distance between the crests, i.e., the wavelength λ, has been reduced to

$$\lambda = cT - vT = \frac{c - v}{f} \tag{3.9}$$

The resulting frequency heard by the observer is not the source's output frequency but that increased by the resulting drop in the wavelength, that is to say,

$$f_d = \frac{c}{\lambda} = \frac{fc}{c - v} = \frac{f}{1 - v/c} \tag{3.10}$$

Here f_d is the frequency perceived by the observer. When the source approaches at a velocity v, the observer hears a higher-frequency sound which represents the original frequency multiplied by a factor $(1 - v/c)^{-1}$. On the other hand, when v assumes a negative value, which means the source is pulling away from the observer, f_d assumes a lower value than that of the source frequency f, since the velocity v in Equation (3.10) assumes a negative value.

Example Problem 2

A train emits a 250-Hz signal while traveling at the rate of 200 km/h. What are the apparent frequencies in approaching the observer and retreating from the observer at the railroad crossing?

Solution

From Equation (3.10)

$$v = (200{,}000 \text{ m/h})/(3600 \text{ s/h}) = 55.6 \text{ m/s}$$
$$f_d = 250 \text{ Hz } (1 - 55.6/344)^{-1} = 298 \text{ Hz for approach}$$
$$f_d = 250 \text{ Hz } (1 + 55.6/344)^{-1} = 215 \text{ Hz for retreat}$$

If an observer stands on a line making an angle θ with a source's direction of motion at speed V, the approach velocity of the source is $v = V \cos\theta$, and Equation (3.10)

modifies to:

$$f_d = \frac{f}{1 - \dfrac{V \cos\theta}{c}} \tag{3.11}$$

The relative frequency increases or diminished, when θ exists as an acute or obtuse angle.

3.7 Reflection

When sound impinges upon a surface, a portion of its energy is absorbed by the surface and the remainder bounces back or becomes reflected from the surface. A perfectly hard surface will reflect back all of the energy. A classic example of the reflection phenomenon is the *echo* which has intrigued and mystified humanity for centuries.

As waves impinges on a hard, smooth surface, the waves are reflected with shape and propagation characteristics unaltered, in accordance with Huygens' principle. Consider the impingement of a series of plane waves on reflecting surface A-A' in Figure 3.6. The arrows normal to the wavefronts, or *rays*, which represent the directions of propagation, are drawn to represent the impingement and the consequent reflection of the wavefront. It follows from the application of Huygens' principle and geometry that the angle of incidence is equal to the angle of reflection, where the angles are defined between a normal to the reflecting plane and the incident and the reflected rays, respectively. In Figure 3.7, the geometric ray construction is rendered for a diverging spherical wave incident upon a plane surface. The direction of the reflected sound can at least be qualitatively determined. It should be pointed out here that *standing wave interference patterns* will occur from these reflections.

It is of interest to consider the sound field resulting from reflection. Consider the sound waves in Figure 3.6 to be sinusoidal. As the incident or reflected wavefront

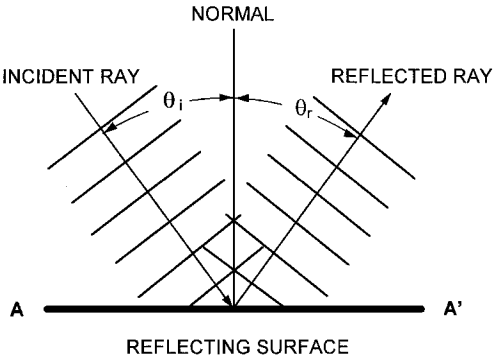


FIGURE 3.6. Geometric depiction of a plane-wave reflection. Angle of reflection θ_r is equal to angle of incidence θ_i.

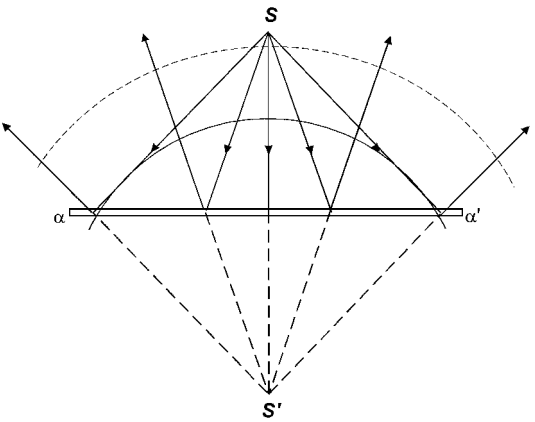


FIGURE 3.7. Construction for a spherical wave from point S incident upon plane surface α–α'. Point S' is an imaginary point that is the mirror image of point S on the other side of plane α–α'.

intersects any normal to the wavefront, a scissorlike effect occurs, not unlike ocean waves breaking obliquely along a beach. The intersection of these waves along the normals constitutes a projection of the incident and reflected waves. From the concept of wave motion, the distance between crests along the normal may be established as a projected wavelength λ', which relates to the incident wave as follows:

$$\lambda' = \frac{\lambda}{\cos\ \vartheta_1} = \lambda\ \sec\ \vartheta_1 \tag{3.12}$$

In obeying the laws of reflection, the reflected wave also scissors back along the normal in the opposite direction, producing a traveling wave with a projected wavelength also equal to λ'. Hence, there occurs along any normal line the superimposition of two waves traveling in opposite directions with wavelength λ'. From the concept of standing waves it can be inferred that nodes and antinodes occur along the normal line, and, moreover, the spacing between the nodes and antinodes needs only to be modified by the factor sec θ_1 of Equation (3.12).

Consider a complex periodic wave that impinges upon a fully reflective plane surface. A standing wave sound field will exist. The distance d' between peaks along the normal ensues from Equation (3.12) in the following manner:

$$d' = \frac{\lambda'}{2} = \lambda_n \sec \theta_1$$

where λ_n is the wavelength of the nth harmonic and θ_i the angle of incidence of the propagating wavefronts. From the last equation it will be noted for the special case of $\theta_i = 0$ (normal incidence), the nodal spacing reduces to $\lambda/2$, according to Equation (3.12). As the angle of incidence increases, the spacing between the nodes likewise increases, and in the limit $\theta_i = \pi/2$, there is no reflected wave, and thus the standing wave field vanishes.

The phenomenon of sound wave reflection finds many applications. The time it takes for a sound wave pulse to travel from a transducer at sea level to the ocean bottom and for the echo to travel back gives a measure of the depth of the water. Further, comparison of the spectral characteristics of the reflected wave with those of the generated waves provides an ample measure of the geological composition of the ocean bottom, for example, silt, rock, sand, coral, and so on. Reflected sound is also used in an analogous way by geologists to gauge the depth and composition of stratified layers in the earth crust, to locate oil, natural gas, and mineral deposits.

3.8 Refraction

A phenomenon more familiar in optics than in acoustics is that of refraction, in which the direction of the advancing wavefront is bent away from the straight line of travel. Refraction occurs as the result of the difference in the propagation velocity as the wave travels from one medium to a different medium.

In the optical situation, refraction occurs suddenly when light waves cross the sharp interface between the atmosphere and glass at the surface of a lens, because light travels more slowly in glass than it does in air. At audible frequencies of sound waves, the wavelengths are so long that the apparatus would have to be extremely large in order to render observable acoustic refractions. However, at ultrasonic frequencies, which correspond to extremely short wavelengths, refraction constitutes the operating principle of the *acoustic microscope*. The device functions as indicated in Figure 3.8. A piezoelectric transducer P_z, under the impetus

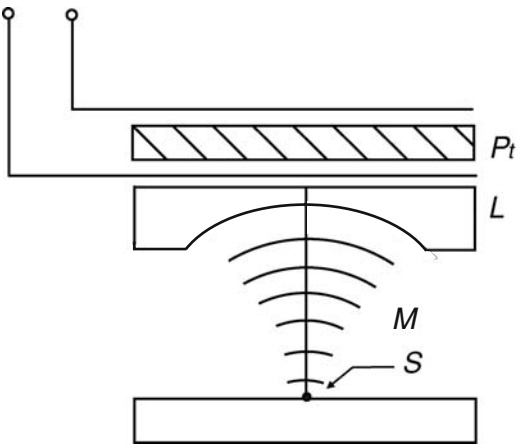


FIGURE 3.8. A schematic of an acoustic microscope. Voltage V causes the piezoelectric crystal P_t to launch a short train of waves into lens L. The propagation velocity of the waves slows down in liquid medium M, and the waves are focused toward point S on the surface of the specimen. Reflected waves follow the same paths in reverse, reaching the piezoelectric transducer that now has been switched into the detector mode.

of an input voltage V to the opposite faces of the piezoelectric crystal, delivers a short train of ultrasonic waves into the lens L, which may consist of a small block of sapphire, which incorporates a spherical hollow in the face facing liquid medium M. The waves travel much faster in the crystal than in the adjoining liquid. As a result the central portion of each wave is retarded relative to the outer parts. All of the wavefront enter the liquid at the same time, but the refractive effect causes almost all of the nearly spherical waves to focus at the central point of curvature. The strength of the reflection depends on the nature of the specimen surface at the focal point S. The operating mode of the lens and transducer now changes from the role of an emitter to that of a receptor. The lens gathers the reflected signal and the transducer detects the signal. In this fashion this device resembles radar and sonar systems. As information is obtained from only one point at a time, the specimen must be moved in a raster pattern on the focal plane of the microscope while an image is progressively accumulated in a computer memory.

If water is used as a medium, a 3-MHz signal, with a propagation velocity of 1480 m/s, would have a wavelength of 500 μm = 0.5 mm, which would amount to a rather coarse resolution. Clearly, higher-frequency signals are called for, but such signals become strongly absorbed in water. A medium with a lower value of c and, more importantly, less absorption than water constitutes another possibility. An attractive choice turned out to be liquid helium, used with instruments that generate signals up to the 8-GHz frequency range. The wavelengths are as small as 0.03 μm.

In Figure 3.9, a geometrical ray construction illustrates the refraction of sound passing from one medium to another. Application of Huygens' principle leads to

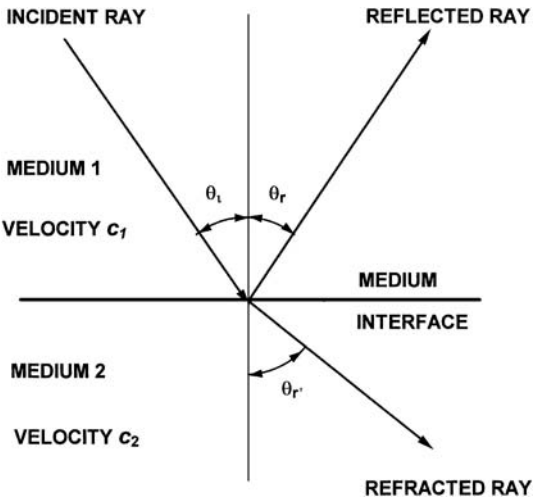


FIGURE 3.9. A sound wave passing from medium 1 to medium 2. In this case the speed of sound c_2 in medium 2 is greater than the speed of sound c_1 in medium 1.

the basic laws of refraction, the most useful of which is

$$\frac{\sin\theta_i}{c_1} = \frac{\sin\theta_r}{c_2} \tag{3.13}$$

where

θ_i = angle of incidence
θ_r = angle of refraction
c_1, c_2 = speeds of sound in medium 1 and medium 2, respectively.

Equation (3.13) should be recognized as the analog to the Snell law for light refraction. While the analysis of refraction does not figure prominently in noise control, we cannot overlook the fact that zones of severe temperature differences do occur in the atmosphere and oceans. When sound travels from zone to zone, often across regions of severe temperature gradients, the direction of propagation changes measurably to an extent that cannot be ignored.

For example, the surface of the Earth heats up more rapidly on a sunny day than the atmosphere. Due chiefly to conduction, the temperature of the air close to the ground rises correspondingly. Because the speed of sound is higher in the warmer lower layer, sound waves traveling horizontally are refracted upward. Similarly on a clear night the Earth's crust cools more quickly, and a layer of cooler air forms and bends the sound waves downward toward the surface. Thus the noise from an industrial plant would be refracted downward at night and would seem louder to a homeowner residing near the plant than during the day (when upward refraction occurs), which is often the situation.

Nonuniform sound speed also constitutes a very important factor in underwater acoustics owing to the persistent presence of temperature and salinity and pressure gradients in the ocean. It is not unusual to find a minimum in c at some depth, usually in the order of 1 km, with higher values above and below that stratum. Interesting possibilities can occur, one of which is communication through *sound channels* in which trapped signals traveling horizontally retain their strength more effectively than if they had been able to spread in all directions. Another is the existence of *shadow zones*, where sound waves from a particular source never arrive, so they provide good places for submarines to hide.

3.9 Diffraction

In Figure 3.10, sound waves impinge upon a barrier. Some of the sound is reflected back, some continues onward unimpeded, and some of the sound bends or diffracts over the top. The barrier does not provide a sharply delineated *acoustical shadow*. Another example of diffraction is bending of sound around a building corner. We usually can hear voices on the other side of a wall that is approximately 3 m high.

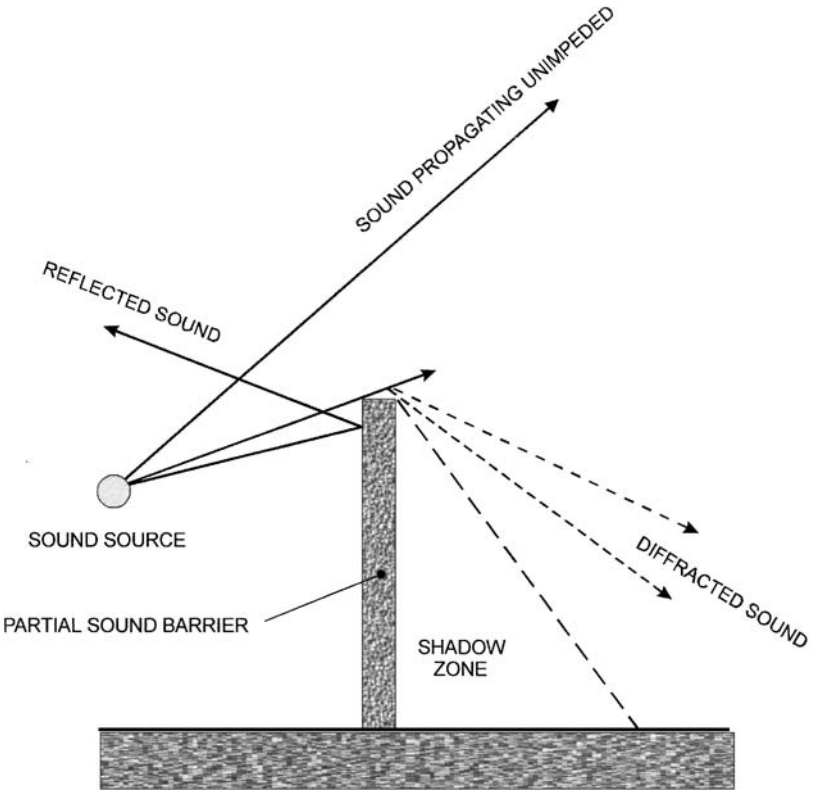


FIGURE 3.10. Impingement of sound waves on a partial barrier, with a resulting sound diffraction and a shadow zone.

The analytical treatment of sound barriers is covered in Chapter 12, but it suffices for now to say qualitatively that sound at lower frequencies tends to diffract over partial barriers more easily than sound at higher frequencies. Moreover, the sharpness and extent of the shadow zone behind the barrier depend on the relative positions of the source and receiver. The closer the source is to the barrier, the longer is the shadow zone on the other side of the barrier, i.e., the more sound reduction obtained.

3.10 Octave and One-Third Octave Bands in the Audio Range

For analytical purposes, the audio range of frequencies are divided into 10 standard octave bands with center frequencies $f_C = 31.5$, 63, 125, 250, and 500 Hz, and 1, 2, 4, 8, and 16 kHz. Each octave band-center frequency f_C is double the preceding one and each bandwidth doubles the preceding one. The lower and upper limits of

TABLE 3.1. Octave Bands.

Lower Band Limit f_L	Center Frequency f_C	Upper Band Limit f_U
22.4 Hz	31.5 Hz	45 Hz
45	63	90
90	125	180
180	250	355
355	500	710
710	1 kHz	1.4 kHz
1.4 kHz	2	2.8
2.8	4	5.6
5.6	8	11.2
11.2	16	22.4

each octave band are given by

$$f_L = \frac{f_C}{\sqrt{2}} \qquad f_U = \sqrt{2} f_C \tag{3.14}$$

Because

$$f_C = \sqrt{f_L \, f_U} \tag{3.15}$$

we see that the center frequency f_C constitutes the geometric mean of the upper band limit f_U and lower band limit f_L. The *bandwidth* **BW** for full octaves is defined by

$$\mathbf{BW} = f_U - f_L = f_C \left(\sqrt{2} - \frac{1}{\sqrt{2}} \right) = \frac{f_C}{\sqrt{2}} \tag{3.16}$$

and thus the ratio $\mathbf{BW}/f_C$ is shown to constitute a constant. In order to avoid the use of irrational numbers the octave bands have been standardized in the field of acoustics, according to Table 3.1.

One-third octave bands are formed by subdividing each octave band into three parts. The successive center frequencies increase in intervals by cube root of 2, and the upper and lower frequencies are related to the center frequency as follows:

$$f_L = \frac{f_C}{\sqrt[6]{2}}, \quad f_U = \sqrt[6]{2} f_C, \quad f_C = \sqrt{f_L \, f_U} \tag{3.17}$$

From Equations (3.16) and (3.17), the ratio $\mathbf{BW}/f_C$ is also a constant for the third-octave bands,

$$\frac{\mathbf{BW}}{f_C} = \sqrt[6]{2} - \frac{1}{\sqrt[6]{2}} \tag{3.18}$$

Table 3.2 lists the standardized one-third octave limits and center frequencies.

TABLE 3.2. One-Third Octave Band.

Lower Band Limit f_L (Hz)	Center Frequency f_C (Hz)	Upper Band Limit f_U (Hz)
18.0	20	24.4
22.4[a]	25	28.0
28.0	31.5[a]	35.5
35.5	40	45[a]
45[a]	50	56
56	63[a]	71
71	80	90[a]
90[a]	100	112
112	125[a]	140
140	160	180[a]
180[a]	200	224
224	250[a]	280
280	315	355[a]
355[a]	400	450
450	500[a]	560
560	630	710[a]
710[a]	800	900
900	1,000[a]	1,120
1,120	1,250	1,400[a]
1,400[a]	1,600	1,800
1,800	2,000[a]	2,240
2,240	2,500	2,800[a]
2,800[a]	3,150	3,550
3,550	4,000[a]	4,500
4,500	5,000	5,600[a]
5,600[a]	6,300	7,100
7,100	8,000[a]	9,000
9,000	10,000	11,200[a]
11,200[a]	12,500	14,000
14,000	16,000[a]	18,000
18,000	20,000	22,400[a]

[a] Octave marking points.

While one-third octave bands generally suffice in providing adequate information, there are cases where one-tenth and even one-hundredth octaves are applied. For 1/nth octaves, successive center frequencies are related as follows:

$$f_{n+1} = 2^{1/n} f_n \tag{3.19}$$

3.11 Root-Mean-Square Sound Pressure and the Decibel

Sound consists of small positive pressure disturbances (compression) and negative pressure disturbances (rarefaction) measured as deviations from the equilibrium or quiescent pressure value. The mean-pressure deviation from equilibrium is

always zero, since the mean rarefaction equals the mean compression. A simple way to measure the degree of disturbance is to square the values of the sound pressure disturbance over a period of time, thereby eliminating the counter-effects of negative and positive disturbances by rendering them always positive. The root-mean-square sound pressure p_{rms} can be defined by

$$p_{\text{rms}} = \sqrt{(p)^2} = \sqrt{\frac{\int_0^\tau p^2 dt}{\int_0^\tau dt}} \tag{3.20}$$

where τ is the time interval of measurement and p the instantaneous pressure. For a simple cosine wave over an interval of period $T = 2\pi/\omega$, there results

$$p_{\text{rms}} = \sqrt{\frac{\int_0^T p_m^2 \cos^2 k(x - ct)\, dt}{T}} = \frac{p_m}{\sqrt{2}}. \tag{3.21}$$

The sound pressure as portrayed by the oscillation of the pressure above and below the atmospheric pressure is detected by normal human ear at levels as low as approximately 20 μPa (the SI unit of pressure is the *pascal*, abbreviated Pa, equivalent to 1.0 N/m^2).[1] Because p_{rms} could vary over a wide range of orders of magnitude, it would be cumbersome to use it as the measure of loudness. At the threshold of pain, p_{rms} would reach approximately 40,000,000 μPa! The blast-off pressure in the vicinity of the launching pad of a Titan rocket can exceed up to a thousandfold the threshold of pain (i.e., 40 kPa). It is therefore more convenient to use the decibel as the folding-scale measure of loudness. This unit is defined by

$$L_p = 10 \log \left(\frac{p_{\text{rms}}}{p_0} \right)^2 = 20 \log \left(\frac{p_{\text{rms}}}{p_0} \right) \tag{3.22}$$

where

$$\begin{aligned} L_p &= \text{sound pressure level (dB)} \\ \log &= \text{common (base-ten) logarithm} \\ p_0 &= 20 \times 10^{-6}\ \text{Pa} = \text{the reference pressure.} \end{aligned}$$

From the context of Equation (3.22) it can be established that the doubling of a root-mean-square pressure corresponds to approximately 6 dB increase in the sound pressure level. In order to determine the sound pressure level from a given value of L_p, Equation (3.22) can be rewritten as

$$\frac{p_{\text{rms}}}{p_0} = 10^{\frac{L_p}{20}}$$

or

$$p_{\text{rms}} = 20 \times 10^{\frac{L_p}{20} - 6}$$

[1] One standard atmosphere equals 101.325 kPa.

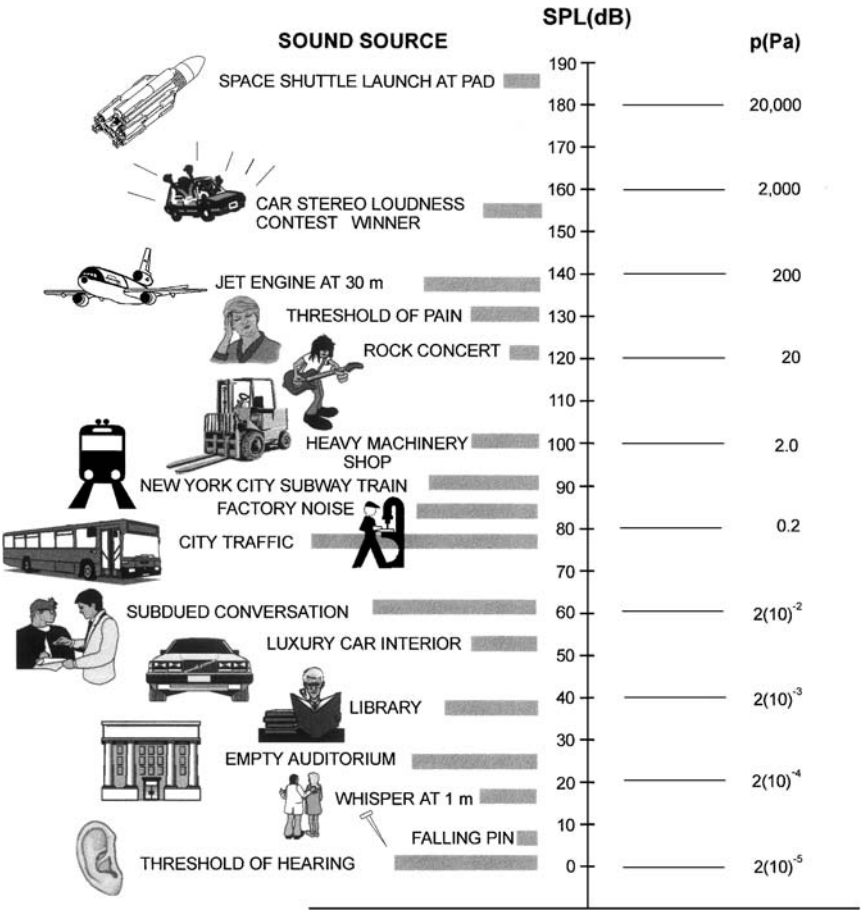


FIGURE 3.11. Sound pressure levels and corresponding pressures of various sound sources.

Figure 3.11 illustrates the range of the decibel scale in terms of measured values of common sound sources. The audible range of sound, which encompasses music and speech, is shown delineated in Figure 3.12 in terms of sound pressure levels and frequencies.

3.12 Decibel Additions, Subtracting, and Averaging

Most sound pressure levels do not arise from single sources, nor do they remain constant in time. Mathematical procedures must be used to add, subtract, and average decibels. From the definition of Equation (3.22) it is apparent that decibels from single-noise sources do not add or subtract directly. If we wish to add sound

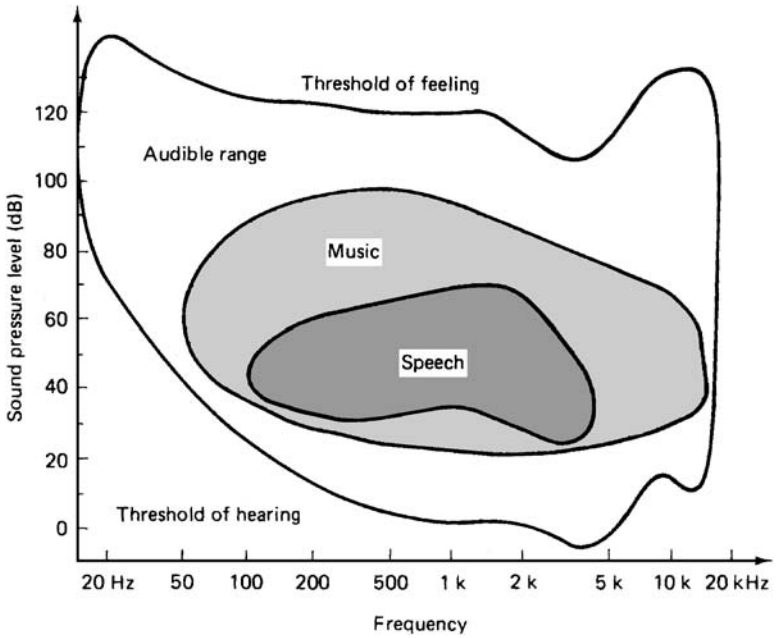


FIGURE 3.12. Sound pressure levels for audible range, music range, and the range of speech versus frequency. (*Source*: Brüel & Kjær Instruments, Inc.)

pressure levels (abbreviated to SPLs) L_{p1}, L_{p2}, L_{p3}, ... L_{pn}, we must obtain the antilogs of these SPLs to convert them into squares of rms pressures which can then be added directly to yield the square of total rms pressure, that in turn yields the total dB. Because

$$\left(\frac{p_i}{p_{\text{ref}}}\right)^2 = \log^{-1}\left(\frac{L_{p_i}}{10}\right)$$

the total sound pressure level L_{pt} becomes

$$L_{pt} = 10\log\left[\sum_{i=1}^{n}\left(\frac{p_i}{p_{\text{ref}}}\right)^2\right] \tag{3.23}$$

or, in terms of the sound pressure levels,

$$L_{pt} = 10\log\left[\sum_{i=1}^{n}\log^{-1}\left(\frac{L_{pi}}{10}\right)\right] = 10\log\left(\sum_{i=1}^{n}10^{L_{pi}/10}\right) \tag{3.24}$$

Example Problem 3

Find the total sound pressure level due to $L_{p1} = 96$ dB, $L_{p2} = 87$ dB, and $L_{p3} = 90$ dB.

Solution

Applying Equation (3.24) we have for this case

$$\begin{aligned} L_{pt} &= 10\log(10^{96/10} + 10^{87/10} + 10^{90/10}) \\ &= 97 \text{ dB} \end{aligned}$$

In certain situations, it is desirable to subtract an ambient or background noise from the total sound pressure level L_{pt} in order to establish the sound pressure level L_{ps} due to a particular source. Subtraction of decibels is analogous to the procedure for their addition. The total sound pressure level L_{pt} is converted into the mean-square pressure ratio, as in Equation (3.23), and the background noise level L_{pB} is measured simply by turning off the noise source. The mean-square pressure ratio due to the background noise is obtained from

$$L_{pB} = 10\log\left(\frac{p_B}{p_{\text{ref}}}\right)^2$$

or

$$\left(\frac{p_B}{p_{\text{ref}}}\right)^2 = \log^{-1}\left(\frac{L_{pB}}{10}\right) = 10^{L_{pB}/10}$$

The sound pressure level L_{ps} of the source is found from

$$L_{ps} = 10\log\left(10^{L_{pt}/10} - 10^{L_{pB}/10}\right) \tag{3.25}$$

Example Problem 4

Measurements indicated $L_{pt} = 93$ dB at a specific location with a lathe in operation. When the lathe is shut down the background noise measures at 85 dB. What is the sound level due to the lathe?

Solution

According to Equation (3.25), the noise level due to the lathe is

$$\begin{aligned} L_{ps} &= 10\log(10^{93/10} - 10^{85/10}) \\ &= 92\,\text{dB} \end{aligned}$$

A requirement may arise occasionally to find the average decibels in order to determine the average sound pressure level L_p. In some situations we may wish to measure the SPL at a single location several times and determine an average value for engineering evaluation purposes. The procedure for averaging decibels is based on the same premise for the summation and subtraction of decibels, namely, the application of Equation (3.22). Equation (3.24) for the addition of decibels is

modified by dividing the sum by n, the number of levels taken into consideration, in order to obtain the average value of L_p:

$$\overline{L}_p = 10 \log \left(\frac{1}{n} \sum_{n=1}^{n} 10^{\frac{L_{pi}}{10}} \right) \tag{3.26}$$

Example Problem 5

Determine the average sound pressure level L_p for a series of measurements taken at different times: 96, 88, 94, 102, and 90 dB.

Solution

Using Equation (3.26) we write

$$\begin{aligned} \overline{L}_p &= 10 \log \left[\frac{1}{5} (10^{96/10} + 10^{88/10} + 10^{94/10} + 10^{102/10} + 10^{90/10}) \right] \\ &= 97 \text{ dB} \end{aligned}$$

3.13 Weighting Curves and Associated Sound Levels

Human perception of loudness depends on the frequency of a sound. A noise having most of its energy concentrated in the middle of the audio spectrum (e.g., in the region of 1 kHz) is perceived as being louder than noise of equal energy but concentrated either in the low-frequency region (say, 40 Hz) or in the high-frequency region (near 15 kHz). This frequency effect becomes more apparent with soft sounds than is the case with loud sounds, which provides the *raison d'être* for the presence of a *loudness control* on some audio amplifiers. This control supplies a loudness contour at low volumes, which applies greater amplification to the high- and low-frequency contents of the program material relative to the middle-frequency components.

Frequency weighting takes typical human hearing response into account when the loudness generated by all of the audible frequency components present is to be represented by a single value. Rather than describing the sound level in each frequency band, we can use the A-weighted sound level to report the overall loudness. The A-weighted sound level is obtained from the conversion chart of Table 3.3, which also lists the B and C weightings in 1/3-octave bands. A-weighting is almost exclusively used in measurements that entail human response to noise. Sound level that is measured with A-weighting is reported in terms of dB(A) or simply dBA rather than the generic decibel dB. Similarly, B-weighted and C-weighted measurements are designated as dB(B) (rarely used) and dB(C), respectively. In the conversion table, it will be noted that all of the

TABLE 3.3. Conversion of Sound Levels from Flat Response to *A*, *B*, *C* Weightings.

Frequency (Hz)	*A* weighting (dB)	*B* weighting (dB)	*C* weighting (dB)
10	−70.4	−38.2	−14.3
12.5	−63.4	−33.2	−11.2
16	−56.7	−28.5	−8.5
20	−50.5	−24.2	−6.2
25	−44.7	−20.4	−4.4
31.5	−39.4	−17.1	−3.0
40	−34.6	−14.2	−2.0
50	−30.2	−11.6	−1.3
63	−26.2	−9.3	−0.8
80	−22.5	−7.4	−0.5
100	−19.1	−5.6	−0.3
125	−16.1	−4.2	−0.2
160	−13.4	−3.0	−0.1
200	−10.9	−2.0	0
250	−8.6	−1.3	0
315	−6.6	−0.8	0
400	−4.8	−0.5	0
500	−3.2	−0.3	0
630	−1.9	−0.1	0
800	−0.8	0	0
1,000	0	0	0
1,250	+0.6	0	0
1,600	+1.0	0	−0.1
2,000	+1.2	−0.1	−0.2
2,500	+1.3	−0.2	−0.3
3,150	+1.2	−0.4	−0.5
4,000	+1.0	−0.7	−0.8
5,000	+0.5	−1.2	−1.3
6,300	−0.1	−1.9	−2.0
8,000	−1.1	−2.9	−3.0
10,000	−2.5	−4.3	−4.4
12,500	−4.3	−6.1	−6.2
16,000	−6.6	−8.4	−8.5
20,000	−9.3	−11.1	−11.2

weighting curves show an adjustment of 0 dB for the 1-kHz frequency band. The *A*-weighting was introduced for sound levels below 55 dB, *B*-weighting for levels between 55 dB and 85 dB, and *C*-weighting was for levels exceeding 85 dB, all of which corresponded, respectively, to human response to low, moderate, and loud sounds. *C*-weighting is relatively "flat" in the mid-range frequencies, with less than 1 dB subtracted from actual dB levels measured in the frequency bands from 63 Hz through 4 kHz. The weighting curves are also shown graphically in Figure 3.13.

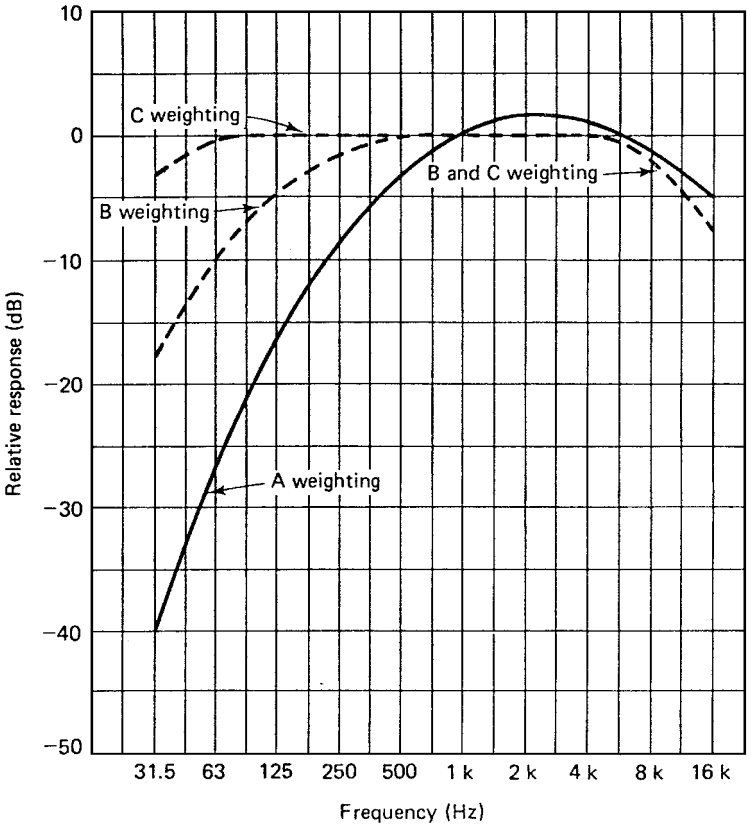


FIGURE 3.13. Frequency responses for the *A*-, *B*-, and *C*-weighting networks.

Example Problem 6

Find the total *A*-weight sound level *L* for the octave-band sound pressure levels given below:

Band-Center Frequency (Hz)	Sound Pressure Level (Hz)
31.5	73
63	68
125	72
250	68
500	80
1000	88
2000	95
4000	83
8000	97
16,000	92

Solution

Use Table 3.3 to obtain the dB conversion from a flat response to dB(A) for each of the octave bands. This results in

$$\begin{aligned}
73 \text{ dB at } 31.5 \text{ Hz} &= 73 - 39.4 = 33.6 \text{ dB}(A)\\
68 \text{ dB at } 63 \text{ Hz} &= 68 - 26.2 = 41.8 \text{ dB}(A)\\
72 \text{ dB at}125 \text{ Hz} &= 72 - 16.1 = 55.9 \text{ dB}(A)\\
68 \text{ dB at } 250 \text{ Hz} &= 68 - 8.6 = 59.4 \text{ dB}(A)\\
80 \text{ dB at } 500 \text{ Hz} &= 80 - 3.2 = 76.8 \text{ dB}(A)\\
88 \text{ dB at } 1 \text{ kHz} &= 88 - 0 = 88 \text{ dB}(A)\\
95 \text{ dB at } 2 \text{ kHz} &= 95 + 1.2 = 96.2 \text{ dB}(A)\\
83 \text{ dB at } 4 \text{ kHz} &= 83 + 1.0 = 84 \text{ dB}(A)\\
97 \text{ dB at } 8 \text{ kHz} &= 97 - 1.1 = 95.9 \text{ dB}(A)\\
92 \text{ dB at } 16 \text{ kHz} &= 92 - 6.6 = 85.4 \text{ dB}(A)
\end{aligned}$$

The dB(A) values in each of the bands can be summed up for the total sound level L_p, through the use of Equation (3.24).

3.14 Performance Indices for Environmental Noise

As the result of the passage of the Noise Control Act of 1972 by the U.S. Congress, the Environmental Protection Agency (EPA) issued two major documents published in April 1974, in accordance with Section 5 of the Act. One document dealt principally with the criteria for time-varying community noise levels and the other document is concerned with definitions of performance indices for noise levels. These indices are generally represented as single-number criteria, serving as internationally recognized, simple means of assessing the noise environment. Three performance indices are described in this section. Because they utilize A-weighted measurements, these three statistically based methods of quantifying noise exposures tend to have good correlation with human response. These indices are L_N, which presents the levels exceeded N percent of the measurement time; L_{eq}, the equivalent continuous sound pressure level in dB(A); and L_{dn}, the day–night sound level average in dB(A).

L_N may be measured with the use of an amplitude-distribution analyzer. An output of the device can provide a histogram, an example of which is shown in Figure 3.14. The time in any chosen band can be read as a percentage of the total observation time. The cumulative distribution curve in the figure indicates the probability of exceeding each range of decibel levels. In noise-abatement planning, criteria are often specified in terms of sound levels that are exceeded 10%, 50%, and 90% of the time. These levels are customarily represented as L_{10}, L_{50}, and L_{90}, respectively.

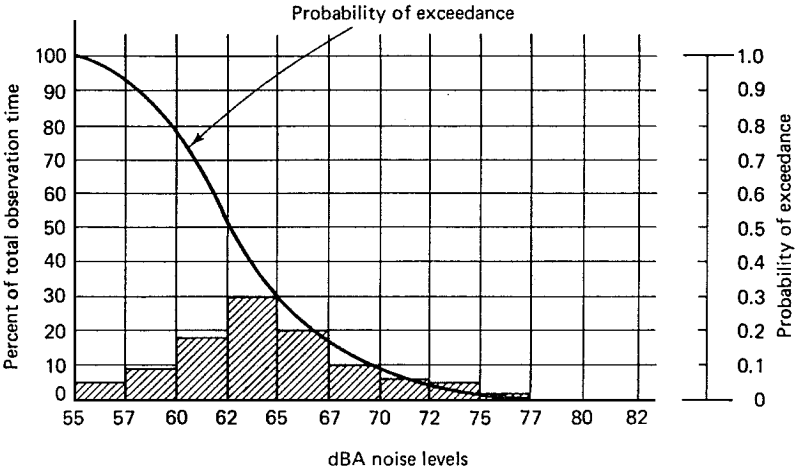


FIGURE 3.14. A histogram showing probability of exceedance for plant noise.

Example Problem 7

From perusal of Figure 3.14, estimate the sound levels that are exceeded 10%, 50%, and 90% of the time. Also establish the percentage of the total observation time that the sound was between 65 and 67 dB(A).

Solution

The probability-of-exceedance curve in Figure 3.14 is read to yield ≈69 dB(A) (exceeded 10% of the time), $L_{50} \approx 63$ dB(A) (exceeded 50% of the time), and $L_{90} \cdot \approx 58$ dB(A) (exceeded 90% of the time).

Equivalent sound level L_{eq} is the sound energy averaged over a given period of time T, i.e., it is the rms or mean level of the time-varying noise. It is defined by

$$L_{\text{eq}} = 10 \log \left(\frac{1}{T} \int_0^T \frac{p^2}{p_{\text{ref}}^2} \, dt \right) \tag{3.27}$$

where $p^2 = p^2(t)$ is the mean-square (time-varying)sound pressure and $p_{\text{ref}} = 20$ μPa. Equation (3.27) can be more conveniently rewritten in terms of sound level $L = L(t)$ using the relationship of Equation (3.22):

$$L_{\text{eq}} = 10 \log \left(\frac{1}{T} \int_0^T 10^{L_p/10} \, dt \right) \tag{3.28}$$

In order to facilitate digital processing in the measurement of L_{eq} through the use of an integrating sound level meter, the integral form of Equation (3.28) is replaced

by the equivalent summation:

$$L_{\text{eq}} = 10 \log \left(\frac{1}{N} \sum_{n=1}^{N} 10^{L_n/10} \right) \tag{3.29}$$

Here L_n is acquired instrumentally for each of N equal intervals to yield L_{eq}, in the course of digital processing of discrete samples.

Example Problem 8

Find L_{eq} in the case where $L_n = 90.5$, 95, 103, 88, and 98 dB(A) are obtained as the respective average levels for five short, equal time intervals.

Solution

From Equation (3.29)

$$\begin{aligned} L_{\text{eq}} &= \log \left[\frac{1}{5} (10^{90.5/10} + 10^{95/10} + 10^{103/10} + 10^{88/10} + 10^{98/10}) \right] \\ &= 97.9 \text{ dB}(A) \end{aligned}$$

The *day–night equivalent sound pressure level*, L_{dn}, essentially a modification of L_{eq}, was conceived for the purpose of evaluating community noise problems. The modification consists of a nighttime penalty of 10 dB imposed on measurements between 10 P.M. and 7 A.M. With time t given in hours, Equation (3.28) now becomes

$$L_{\text{dn}} = 10 \log \left[\frac{1}{24} \left(\int_{7 \text{ A.M.}}^{10 \text{ P.M.}} 10^{L/10} \, dt + \int_{10 \text{ P.M.}}^{7 \text{ A.M.}} 10^{(L+10)/10} \, dt \right) \right] \tag{3.30}$$

When the equivalent sound levels L_{eqd} and L_{eqn} are known for the day and night periods, respectively, the following version of Equation (3.30) can be used to compute the day–night sound level:

$$L_{\text{dn}} = 10 \log \left[\frac{1}{24} \left(15 \times 10^{L_{\text{eqd}}/10} + 9 \times 10^{(L_{\text{eqn}}+10)/10} \right) \right] \tag{3.31}$$

Example Problem 9

Find L_{dn} for the situation where the daytime equivalent sound level is 82 dB(A) and the nighttime equivalent sound level is 76 dB(A).

Solution

Inserting the appropriate values in Equation (3.31) we obtain

$$L_{\text{dn}} = 10 \log \left[\frac{1}{24} (15 \times 10^{\frac{82}{10}} + 9 \times 10^{\frac{(76+10)}{10}}) \right] = 84 \text{ dB}(A)$$

3.15 Particle Displacement and Velocity

Invoking Equation (2.21) and inserting the wave equation solution (3.1), we obtain the expression for the particle velocity u:

$$u = -\frac{1}{\rho}\int \frac{\partial \rho}{\partial x} dt = -\frac{p_m}{\rho c}\cos k(x - ct)$$

$$u = \frac{p}{\rho\, c} \tag{3.32}$$

where ρ = quiescent density of air = 1.18 kg/m^3 at a normal room temperature of 22°C and atmospheric pressure of 101.3 kPa and c = speed of sound = 344 m/s. The term ρc is the *characteristic* or *acoustic impedance* for a wave propagating in air in a free-field condition. The value of ρc at standard conditions of temperature and pressure is 40.7 rayls or 407 MKS rayls. The dimensional unit rayl is defined as follows:

$$1 \text{ rayl} = 1.0 \text{ dyne s/cm}^3$$

The particle displacement x for a cosine wave function can be found by simply integrating Equation (3.32) with respect to time:

$$x = \frac{p_m}{\rho c^2}\sin k(x - ct)$$

It is interesting to note that at 0 dB, the threshold of human hearing, the oscillation of an air molecule covers an rms amplitude that is approximately only one-tenth the diameter of a hydrogen atom.

The particle acceleration is obtained from the differentiation of Equation (3.32) with respect to time, and for a cosine wave function the acceleration is

$$\frac{du}{dt} = \frac{p}{\rho}\sin k(x - ct)$$

3.16 Correlated and Uncorrelated Sound

Correlated sound waves occur when they have a precise time and frequency relationship between them. An example of correlated sound waves is the output of two identical loudspeakers located in the same plane, consisting of a pure tone supplied by a single amplifier connected to both loudspeakers. Most of the sound waves that we hear are generally *uncorrelated*.

Consider two sound waves which are detected at a point in space:

$$p_1 = P_1 \cos(\omega_1 t + \phi_1) \tag{3.33a}$$

$$p_2 = P_2 \cos(\omega_2 t + \phi_2) \tag{3.33b}$$

or in terms of complex exponential functions

$$p_1 = \mathbf{Re}\left[P_1 e^{i(\omega_1 t + \phi_1)}\right] \tag{3.33c}$$

$$p_2 = \mathbf{Re}\left[P_2^{i(\omega_2 t + \phi_2)}\right] \tag{3.33d}$$

where

p = instantaneous sound pressure
P = amplitude of sound pressure
ω = angular frequency
ϕ = phase angle

The instantaneous sound pressure resulting from the superimposition of the two waves is given by the sum of the two instantaneous sound pressures, and the root-mean-square sound pressure of the combined waves can be found from

$$p_{\text{rms}}^2 = \frac{1}{T}\int_0^T (p_1 + p_2)^2\, dt \tag{3.34}$$

where T represents the averaging time, which should be an integer number of periods at both frequencies. In real measurements it suffices to have the averaging time cover many periods so that contributions from fractional periods become insignificant. This condition is met if $T \gg 1/f_{\text{lower}}$, where f_{lower} is the lower of the two frequencies. Inserting Equations (3.33a) and (3.33b) or Equations (3.33c) and (3.33d) into Equation (3.34) and integrating, we obtain

$$p_{\text{rms}}^2 = \frac{{P_1}^2 + {P_2}^2}{2} = p_{\text{rms1}}^2 + p_{\text{rms2}}^2 \qquad \text{for } \omega_1 \neq \omega_2 \tag{3.35}$$

$${p^2}_{\text{rms}} = \frac{{P_1}^2 + {P_2}^2}{2} + P_1 P_2 \cos(\phi_1\, \phi_2) \qquad \text{for } \omega_1 = \omega_2 \tag{3.36}$$

Consider the case of two 4-kHz signals that are in phase at the receiving point. Each of the signals has a sound pressure level of 60 dB. From Equation (3.36) for sound waves of the same frequency, the root-mean-square pressure is given by

$$p_{\text{rms}}^2 = \frac{P_1^2 + P_2^2}{2} + P_1 P_2 \cos 0 = 2P_1^2 = 4\, p_{\text{rms}}^2$$

The increase in sound pressure level as the result of adding an identical in-phase pure tone is

$$\begin{aligned} L_P - L_{P1} &= 10\log\left(\frac{p_{\text{rms}}^2}{p_{\text{ref}}^2}\right) - 10\log\left(\frac{p_{\text{rms1}}^2}{p_{\text{ref}}^2}\right) = 10\log\left(\frac{4p_{\text{rms1}}^2}{p_{\text{rms1}}^2}\right) \\ &= 10\log 4 \approx 6 \text{ dB} \end{aligned}$$

The combined signals result in a 4-Hz signal with an SPL of 66 dB. If the amplitudes would be out of phase by π radians (or 180°) and the frequencies are equal, the sound pressure would theoretically be zero at the observation point.

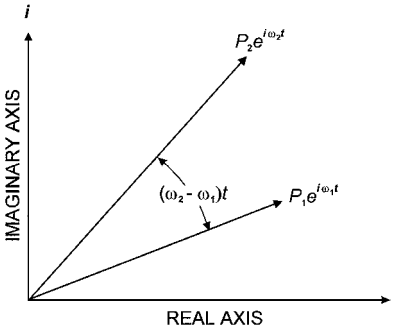


FIGURE 3.15. Complex exponentials portrayed as rotating vectors.

Let us now determine the effect of adding two 60-dB signals, which have frequencies of 1000 and 1100 Hz, respectively. Because the frequencies are not equal, the root-mean-square sound pressure is double that of one wave. The sound pressure level increases by

$$10 \log \left(\frac{p_{\text{rms}}^2}{p_{\text{rms1}}^2} \right) = 10 \log 2$$

or approximately 3 dB. The combined SPL is 63 dB.

We gain a further insight into the phenomenon of beat frequency if we visualize the complex exponential functions (3.33c) and (3.33d) as rotating vectors in Figure 3.15. Without loss of generality we can define time $t = 0$ as the instant when both vectors lie along the positive real axis, resulting in the maximum sound pressure. The two vectors will be opposed along the real axis when $(\omega_2 - \omega_1)t = 2\pi$. The envelope of the pressure–time curve yields a period $\tau = 2\pi/(\omega_2 - \omega_1) = 1/(f_2 - f_1)$. The term $(f_2 - f_1)$ is, of course, the beat frequency, which has been previously discussed in Section 3.3.

3.17 Sound Intensity

An acoustic signal emanates from a point source in a spherical pattern over an increasingly larger area. When a closed surface completely surrounding the source is defined, the sound power **W** radiated by the source can be established from

$$\mathbf{W} = \int_S I \cdot dS \tag{3.37}$$

where

$$\begin{aligned} \boldsymbol{I} &= \text{sound intensity, W/m}^2 \\ dS &= \text{element of surface area, m}^2 \\ S &= \text{surface area surrounding source} \end{aligned}$$

Here the surface integral in Equation (3.37) is the integral of the sound intensity $\boldsymbol{I}$ normal to the element dS of the surface area. The integration can be executed over a spherical or hemispherical surface enclosing the source. If the source of power W is mounted on an acoustically hard surface (i.e., a surface which is totally reflective), the sound waves expand within a hemisphere. Other surfaces, such as those of a parallelopiped (representing, for instance, the walls of a room), are often used in practical applications. When the integration is performed over a spherical surface of radius r for a nondirectional source, sound intensity is related to sound power by

$$I(r) = \frac{W}{S} = \frac{W}{4\pi r^2} \tag{3.38}$$

where S denotes the area of a sphere having radius r. Equation (3.38) constitutes the inverse-square law of sound propagation, which accounts for the fact that sound becomes weaker as it travels in open space away from the source, even if viscous effects of the medium are disregarded. For sound radiation within a hemisphere, with the sound source mounted at the origin above a totally reflective surface, Equation (3.38) becomes

$$I = \frac{W}{2\pi r^2}$$

Intensity, which represents the transfer of sound wave energy, equals the product of sound pressure and particle velocity,

$$I = p \cdot u \tag{3.39}$$

and for a simple cosine spherical wave, the pressure $p(r, t)$ is given as a solution to the spherical coordinate form of Equation (2.25)

$$p(r, t) = \frac{A}{r} \cos k(r - ct)$$

A is a constant amplitude with its physical units in N/m. From Equation (2.22) the velocity is

$$\begin{aligned} u(r, t) &= -\frac{1}{\rho} \int \left[\frac{A}{r} k \sin k(r - ct) + \frac{A}{r^2} \cos k(r - ct) \right] dt \\ &= -\frac{1}{\rho} \left[-\frac{kA}{kcr} \cos k(r - ct) - \frac{A}{r^2 kc} \sin k(r - ct) \right] \end{aligned}$$

or

$$u(r, t) = \frac{A}{\rho cr} \cos k(r - ct) \left[1 + \frac{1}{kr} \tan k(r - ct) \right] \tag{3.40}$$

At large values of kr Equation (3.40) becomes

$$u(r, t) \approx \frac{p(r, t)}{\rho c} \qquad k^2 r^2 \gg 1 \tag{3.41}$$

and for $k^2r^2 \ll 1$:

$$u(r,t) \approx \frac{A}{k\rho cr^2} \sin k(r - ct) \approx \frac{p(r,t)}{\rho ckr} \qquad \angle 90^\circ \text{ re } p(r,t) \tag{3.42}$$

The difference between Equations (3.41) and (3.42) connotes, respectively, the *far field* and *near field effects* of a spherical wave. As r approaches the center of the spherical source the sound pressure and particle velocity becomes progressively more out of phase, approaching 90° as the limit. In the near field the sound intensity is not simply related to the root-mean-square value of the sound pressure.

A sound source is generally directional, and the sound intensity does not have the same value at all points on the surface. In order to evaluate the integral of Equation (3.37) it is necessary to execute an approximation by segmenting the surface into a finite number of sub-elements, each subtending an area S_i and to establish the sound intensity on each sub-element in a direction normal to that element. A summation procedure over all of the surface sub-elements will yield the total sound power

$$W = \sum_i I_i S_i \tag{3.43}$$

where

I_i = sound intensity averaged over the ith element of area S_i, W/m^2
S_i = ith element of area, m^2

Equation (3.43) can be expressed logarithmically as

$$L_w = 10 \log \left(\frac{I}{I_0} \right) + 10 \log \left(\frac{S}{S_0} \right)$$

$$L_w = L_I + 10 \log \left(\frac{S}{S_0} \right)$$

where

L_w = sound power level, dB *re* 10^{-12} W
L_I = sound intensity level, dB *re* 10^{-12} W/m^2
S = area of surface, m^2
S_0 = reference area = 1.0 m^2
I_0 = reference sound intensity, internationally set at 10^{-12} W/m^2

3.18 The Monopole Source

A *monopole* can be described as an idealized point generating a spherical sound wave. A pulsating sphere can be considered a good approximation of a point source when its radius is small compared with the wavelength of the sound it generates.

The three-dimensional Equation (2.25) is expressed in spherical coordinates, without angular dependence, as

$$\frac{\partial^2 p}{\partial t^2} = c^2 \left(\frac{\partial^2 p}{\partial r^2} + \frac{2}{r} \frac{\partial p}{\partial r} \right) \tag{3.44}$$

The solution to Equation (3.44) must be of the form

$$p = \frac{1}{r}[F_1(ct - r) + F_2(ct + r)]$$

with the term $F_1(ct - r)$ describing waves moving away from the source. We discard the term $F_2(ct + r)$ which describes waves traveling toward the source. With time $t = 0$ in order to eliminate a phase angle, a harmonic solution to the spherical wave equation (3.44) is given by

$$p = \frac{A}{r} \cos[k(ct - r)]$$

where A is a constant and k the wave number equal to ω/c. The root-mean-square sound pressure p^2_{rms} at a distance r from the source is given by

$$p^2{}_{\text{rms}} = \frac{1}{T} \int_0^T p^2 \, dt = \frac{A^2}{r^2 T} \int_0^T \cos[k(ct - t)] dt = \frac{A^2}{2 r^2} \tag{3.45}$$

Here $T = 1/f = 2\pi/\omega$ or the period needed to complete one cycle. If many frequencies are present, p^2_{rms} and the root-mean-square pressure can be measured fairly accurately if the integration time is sufficiently large compared with the period of the lowest frequency. The sound power of the spherical source can be found by the use of Equation (3.37) as follows:

$$W = \iint_s \mathbf{I} \cdot \mathbf{n} \, dS \tag{3.46}$$

Equation (3.46) represents the sound power of a source where

$$\mathbf{I} = \frac{1}{T} \int_0^T \rho \mathbf{u} \, dt \tag{3.47}$$

constitutes the vector sound intensity; $\mathbf{u}$ represents the particle velocity vector, S any closed surface about the source, $\mathbf{n}$ the unit normal to surface S, and T the averaging time. The surface S for a spherical wave is defined by a sphere of radius r about the source, so that Equation (3.46) becomes

$$W = \iint_S I_r \, dS \tag{3.48}$$

The magnitude of the sound intensity I_r is directed radially, i.e., it runs parallel to unit normal $\mathbf{n}$. When sufficiently far from the source, the sound pressure and particle velocity are in-phase. Applying Equation (3.39) yields

$$I_r = p_{\text{rms}} \, u_{\text{rms}} = \frac{p^2{}_{\text{rms}}}{\rho c}$$

where ρ is the mass density of the propagation medium. With the sound source being isotropic (i.e., omnidirectional, with no angular-dependent variations), and integration of Equation (3.46) over 4π steradians, the sound power is given by

$$W = I_r\, S = 4\pi\, r^2\, I_r = \frac{4\pi r^2\, p_{\text{rms}}^2}{\rho c} \tag{3.49}$$

If sound power is measured in half-space, i.e., if the source lies on a reflective surface, then integration occurs over 2π steradians and

$$W = \frac{2\pi r^2 p_{\text{rms}}^2}{\rho c} \tag{3.50}$$

Here, W denotes the sound power of the source in watts, I the sound intensity (W/m^2) in the direction of wave propagation, and r the distance from the center of the source.

For a spherical wave in full space, root-mean-square sound pressure and sound intensity in the direction of wave propagation are related to the sound power of the source by

$$p_{\text{rms}}^2 = \frac{\rho c W}{4\pi\, r^2} \tag{3.51a}$$

$$I = \frac{W}{4\pi\, r^2} \tag{3.51b}$$

The tendency of the sound intensity in Equation (3.51b) to decrease with increasing distance from the source is called the *inverse square law*.

3.19 The Spherical Wave: Sound Pressure Level and Sound Intensity Level

Combining Equations (3.51a) and (3.51b) with the definition of sound pressure level given by Equation (3.22), we express the sound pressure Lp in terms of sound power W of the source and distance r from the source:

$$\begin{aligned} L_p &= 10\log\left(\frac{p_{\text{rms}}^2}{p_{\text{ref}}^2}\right) = 10\log\left(\frac{\rho c W}{4\pi r^2 p_{\text{ref}}^2}\right) \\ &= 10\log(\rho c W) - 20\log r + 83 \end{aligned} \tag{3.52}$$

where $p_{\text{ref}} = 20$ μPa. The sound intensity level L_I in the direction of spherical propagation is found from

$$\begin{aligned} L_I &= 19\log\left(\frac{I}{I_{\text{ref}}}\right) = 10\log\left(\frac{W}{4\pi r^2\, p_{\text{ref}}^2}\right) \\ &= 10\log W - 20\log r + 109 \\ &= L_W - 20\log r - 11 \end{aligned} \tag{3.53}$$

where $I_{\text{ref}} = 10^{-12}$ W/m^2 and L_W is the sound power level (dB re 1 pW).

3.20 The Hemispherical Wave

If an omnidirectional sound power source W is placed above an acoustically hard (i.e., totally reflective) surface, the sound waves expand within a half-space. For sound pressure and particle velocity in phase, sound intensity in the direction of propagation is given by

$$I = \frac{W}{2\pi r^2} \tag{3.54}$$

and the sound intensity level by

$$L_I = L_W - 20\log r - 8 \tag{3.55}$$

A *hemianechoic chamber* is constructed by installing wedges of sound-absorbing materials on the ceiling and walls of a room with an acoustically hard floor. In contrast, a full anechoic chamber is constructed with virtually *all* of its surfaces, including the floor, lined with sound absorption wedges. A mesh floor or grating above the bottom surfacing provides structural support to equipment and laboratory personnel. With either type of chamber, a free-field condition is generated within, in which the sound emitted by a source placed inside does not undergo reflection.

3.21 Energy Density

In the mathematical treatment of sound in enclosed spaces, we need to know the amount of energy per unit volume being transported from a source to different parts of the room. Both kinetic energy and internal (potential) energy are involved in sound propagation. An interchange between these two forms of energy occurs from the compression/rarefaction process and the motion of the propagation medium particles. The kinetic energy density e_k is simply $\rho u^2/2$, and when the wave amplitude is small it will be fairly legitimate to assume the quiescent value of ρ_0 instead of ρ. In general, we may write an expression for kinetic energy in terms of displacement vector $\mathbf{x}$:

$$e_k = \frac{\rho_0}{2}\left(\frac{\partial \mathbf{x}}{\partial t}\right)^2 \tag{3.56}$$

The three-dimensional version of Equation (2.24) is

$$\frac{1}{\rho_0}\nabla p = -\frac{\partial^2 \mathbf{x}}{\partial t^2}$$

which leads to

$$\frac{1}{\rho_0}\int \nabla p\, dt = \frac{\partial \mathbf{x}}{\partial t} \tag{3.57}$$

Inserting Equation (3.57) into Equation (3.56) yields

$$e_k = \frac{1}{2\,\rho_0}\left(\int \nabla p\, dt\right)^2$$

For a general case of a sine spherical wave described by

$$p = A\sin(r - ct)$$

we have

$$e_k = \frac{1}{2\,\rho_0\, c^2}\, p^2 \tag{3.58}$$

From elementary thermodynamics, the change in energy per unit volume V_0 associated with the variation of density is given for a volume V of the fluid by

$$e_p = -\frac{1}{V_0}\int_V^{V_0} p\, dV \tag{3.59}$$

The negative sign indicates that the potential energy increases when compression occurs (i.e., when density increases) and decreases with rarefaction (when density decreases) under the impetus of an acoustic signal. In order to perform the integration we need to express all variables in terms of one variable, namely, instantaneous pressure p. From conservation of mass $p_0 V_0 = pV =$ constant, and differentiating yields

$$dV = -\frac{V}{\rho}d\rho \approx -\frac{V_0}{\rho_0}d\rho$$

From Equation (2.23),

$$\frac{dp}{d\rho} = \gamma\frac{p_0}{\rho_0} = c^2$$

for an isentropic process in an ideal gas. Eliminating $d\rho$ between the preceding two equations gives

$$dV = -\frac{V_0}{\rho_0\, c^2}dp$$

which now can be inserted into Equation (3.59), which is then integrated from 0 to p to yield

$$e_p = \frac{1}{2}\frac{p^2}{\rho_0\, c^2} \tag{3.60}$$

The sum of Equations (3.58) and (3.60) constitutes the total *instantaneous energy density* denoted by e

$$e = \frac{1}{2}\,\rho_0\left(u^2 + \frac{p^2}{\rho_0^2\, c^2}\right) \tag{3.61}$$

Because the particle speed and acoustic pressure are functions of both time and space, the instantaneous energy density is not constant throughout the fluid

medium. The time average E of e provides the energy density at any point in the fluid

$$E = \frac{1}{T}\int_0^T e\,dt$$

where the time interval T represents one period of a harmonic wave. With the fact that $p = \forall\rho_0 cu$, as manifested in Equation (3.32), Equation (3.61) becomes

$$e = \rho_0 u^2 = pu/c$$

If we now let p and u represent the amplitude of the pressure and particle velocity, respectively, then the time-averaged energy E is written as

$$E = \frac{1}{2}\frac{pu}{c} = \frac{p^2}{2\,\rho_0\,c^2} = \frac{1}{2}\,\rho_0\,u^2 \tag{3.62}$$

For the cases of spherical or cylindrical waves or standing waves in a room, the pressure and particle velocity in Equation (3.61) must be the real quantities derived from the superposition of all waves present. In these more complex cases, the pressure is not necessarily in phase with the particle speed, nor is the energy density given by $pu/2c$. But $E = pu/2c$ does constitute a good approximation for progressive waves if the surfaces of constant phase approaches a radius of curvature much greater than a wavelength. This situation occurs for spherical and cylindrical waves at distances considerably far (i.e., many wavelengths) from their sources.

References

Federal Register, May 28, 1975. v. 40: 23105.
Federal Register, Oct. 29, 1974. v. 39: 38208.
Federal Register, Oct. 30, 1974. v. 39: 38338.
Harris, Cyril M. (ed.). 1991. *Handbook of Acoustical Measurements and Noise Control*, 3rd ed. New York: McGraw-Hill, Chapters 1 and 2. (A most valuable reference manual for the practicing acoustician.)
Lord, Harold, Gatley, William S., and Evensen, H. A. 1980. *Noise Control for Engineers*. New York, NY: McGraw-Hill, pp. 7–30.
Morse, Philip M. and Ingard, K. Uno K. Uno. 1968. *Theoretical Acoustics*. New York: McGraw-Hill. Chapter 1. (A classic text which probably contains more details than necessary, with the result that the mathematics tends to obscure the physics of acoustics.)
Pierce, Alan D. 1981. *Acoustics: An Introduction to Its Physical Principles and Applications*. New York: McGraw-Hill, Chapters 1 and 2. (An excellent modern text by a contemporary acoustician.)

Problems for Chapter 3

1. A signal consists of the following components:

$$y_1 = 10 \sin 4t, \quad y_2 = 6 \sin 8t, \quad y_3 = 4.3 \sin 10t$$

   Plot each component and add them up to obtain a composite wave.
2. A 300-Hz sound wave is propagating axially in a steel bar. Find the wavelength and the wave number.
3. Determine the wave number and the wavelength of a 30-Hz pure tone at 20° C.
4. Two sine waves have approximately the same amplitudes, but one is at a frequency of 135.3 Hz and the other 136.0 Hz. What is the beat frequency? What is the time duration between the beats?
5. Show mathematically how noise cancellation can be effected by duplicating an offending signal and changing its phasing. In real life situation, can the cancellation be a total one? If not, why not?
6. It is desired to place a worker at a "quieter" location near a machine that puts out a steady 400 Hz hum. Where would this location be and where are the points where the noise would be greater?
7. A train whistle is measured with its frequency at 250 Hz when the trains approaches the observer near the tracks at the rate of 125 km/h. Predict the frequency when the train is pulling away from the observer.
8. Train A traveling 60 km/h is approaching Train B traveling on a parallel track at 85 km/h. Train A blows its whistle which has a fundamental of 255 Hz. What frequency will the engine man at Train B hear? What will be the perceived frequency after the whistle on Train B passes the locomotive of Train A? Neglect the distance between the tracks.
9. An observer stands 150 km from a railroad track. A train is 300 km on the normal from the track to the observer. It is traveling at 80 km/h and approaching this normal. Its whistle emits a fundamental of 300 Hz. What will be the frequency of this signal perceived by the observer?
10. Develop an equation for the rate of change of the frequency with respect to an observer subtending an angle θ with a source's direction. The source is moving at a velocity V.
11. A boundary exists between two mediums, A and B, through which an acoustic signal travels. The signal traveling in Medium A impinges the boundary at an angle of 45°. In passing into Medium B on the other side of the boundary the signal refracts at an angle of 55°. The velocity of sound in Medium A is 450 m/s. Determine the velocity of sound in Medium B on the other side of the boundary.
12. Given a 150-Hz signal expressed as

$$p(x, t) = 35 \sin 2.5(x - 344t)$$

    determine the followings:

(a) the wave number
(b) the wavelength
(c) the root-mean-square pressure

13. Convert the following rms pressures into decibels:
    (a) 20 μPa
    (b) 150 μPa
    (c) 1 kPa
    (d) 50 kPa
14. Convert the following values expressed in dB into rms pressure:
    (a) 20 dB
    (b) 60 dB
    (c) 90 dB
    (d) 130 dB
15. Two machines are running. One machine puts out 95 dB and the other 98 dB. What will be the combined sound pressure level in dB?
16. Three machines are operating simultaneously. Their combined noise level is 115 dB. One machine is shut down and noise level drops to 110 dB. The remaining two machines have identical noise output. What is the noise output of these two machines?
17. An octave-band analysis of a machine yields the following results:

Band-center frequency	SPL (Hz)
31.5	72
63	76
125	77
250	72
500	69
1000	84
2000	92
4000	83
8000	80
16,000	78

Find the total A-weighted sound level, the total B-weighted sound level, and the total C-weighted sound level.

18. The following SPL readings were taken of a noisy electric generator:

Amount of time, s	Unweighted SPL reading, dB
15	73.4
22	79.4
20	88.9
12	91.9

Find the equivalent sound pressure level L_{eq}.

19. At a property line the noise from a nearby machine shop was found over a 24-h period to have the following averaged sound pressure levels.

Time	Noise Level, dB(A)
7 A.M.–12 noon	87.5
12 noon–4 P.M.	84.6
4 P.M.–9 P.M.	78.5
9 P.M.–3 A.M.	76.5
3 A.M.–7 A.M.	77.4

Determine the L_{eq} and L_{dn}.

20. Determine the particle velocity for air at 1 atm and 22°C.

4
Vibrating Strings

4.1 Introduction

In dealing with vibrating systems it is commonly assumed that the entire mass of the system is concentrated at a single point and the motion of the system can be described by giving the displacement as a function of time. This rather simplified approach yields approximations rather than accurate closed-form solutions. A spring, for example, certainly does not concentrate its mass at one end, nor can a loudspeaker be accurately depicted as being a massless piston engaged in an oscillating motion. The loudspeaker diaphragm consists of a considerable portion of its mass spread out over its surface, and each part of the diaphragm can vibrate with a motion that is different from those of other segments.

The vibrational modes of a loudspeaker constitute a complex affair, so it would behoove us to study simpler modes of vibration, say, those of a vibrating string or bar, so that we can readily visualize the transverse vibrations. Even in the simplest of cases, certain simplifying assumptions have to be made which cannot be fully justified in the real physical world.

4.2 The Vibrating String: Basic Assumptions

Consider a long, heavy string stretched to a moderate tension between two rigid supports. A momentary force is applied to the string that becomes displaced from its equilibrium position. The displacement does not remain in the initial position; it breaks up into two separate disturbances that propagate along the string apart from each other as shown in Figure 4.1. The propagation velocity of all *small* displacements depends only on the mass and tension of the string, not on the shape and amplitude of the initial displacement. The wave generated by such a transverse perturbation is generally known as a *transverse wave*.

4.3 Derivation of the Transverse Wave Equation

In Figure 4.2, a portion of a string under tension T and rigidly clamped at its ends is shown. The string has negligible stiffness and a uniform linear density δ.

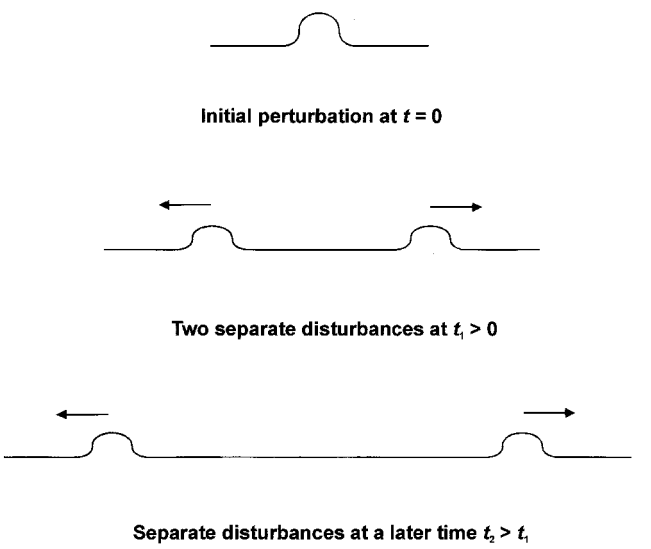


FIGURE 4.1. History of the propagation of a disturbance in a stretched string.

Dissipation of vibrational energy is neglected. We let x represent the coordinate of a point along the horizontal distance with the origin at the left clamp of the string. The y-coordinate represents the transverse displacement from the equilibrium position. As the transverse displacements are defined as being small, tension T can be considered nearly constant ($T \cos\theta$ is even more so). Let θ denote the angle between a tangent to the string and the x-axis. In the segment of the string shown

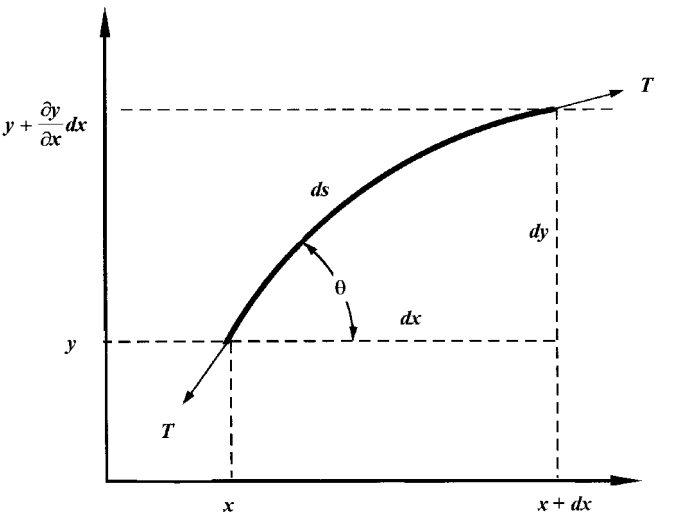


FIGURE 4.2. String element under the influence of tension force T.

in Figure 4.2, the difference between the y-components of the tension at the two ends of element ds is the net transverse force given by

$$\Delta F_y = (T \sin\theta)_{x+\Delta x} - (T \sin\theta)_x \tag{4.1}$$

Here $(T \sin\theta)_{x+\Delta x}$ is the value of $T \sin\theta$ at $x + \Delta x$, and $(T \sin\theta)_x$ is the value at x. Letting $\Delta x \to dx$ and applying the Taylor's series expansion

$$f(x + dx) = f(x) + \frac{\partial f(x)}{\partial x} dx$$

Equation (4.1) can be rewritten as

$$dF_y = (T \sin\theta)_x + \frac{\partial (T \sin\theta)}{\partial x} - (T \sin\theta)_x = \frac{\partial (T \sin\theta)}{\partial x} dx$$

As the displacement y is assumed to be small, θ will be correspondingly small and the relationship $\sin\theta \approx \tan\theta$ applies, with $\tan\theta$ equal to y/x. The net transverse force on the element ds then becomes

$$dF_y = \frac{\partial \left[\left(T \frac{\partial y}{\partial x} \right) \right]}{\partial x} dx = T \frac{\partial^2 y}{\partial x^2} dx$$

The mass of the string element is $\delta\, dx$. Applying Newton's law $F = ma$, we get:

$$T \frac{\partial^2 y}{\partial x^2} dx = \delta dx \frac{\partial^2 y}{\partial t^2}$$

Setting

$$c = \sqrt{\frac{T}{\delta}} \tag{4.2}$$

the equation of string motion becomes

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} \tag{4.3}$$

The constant c defined in Equation (4.2) represents the propagation velocity of the transverse wave. Equation (4.3) is the wave equation representing the wave disturbances propagated along the string. This equation was first derived by Leonhard Euler in 1748.

4.4 General Solution of the Wave Equation

The second-order partial differential Equation (4.3) has the general solution

$$y = f(ct - x) + g(ct + x) \tag{4.4}$$

where the functions, $f(ct - x)$ and $g(ct + x)$, are arbitrary with arguments $(ct \pm x)$. The first term of the right-hand side of Equation (4.4) represents a wave moving

to the right (in the positive x-direction) and the second term a wave moving to the left (in the negative x-direction). While each of the two wave shapes remains constant as the initial perturbation propagates along the string, the actual fact is quite the opposite since the simplifying assumptions are not fully realized in real strings. In relatively flexible strings with low damping, the rate of wave distortion is quite minimal as long as the initial perturbation is kept small. Large amplitudes, however, will result in a larger rate of change of the wave shapes.

The functions $f(ct - x)$ and $g(ct + x)$ cannot be freely arbitrary; they are constrained by initial and boundary conditions. The initial conditions, established at time $t = 0$, are dictated by the type and the location of application of the perturbing force applied to the string. To cite a musical example, the initial wave shape generated by plucking the string of a banjo or a harp will be quite different from the wave shape created by bowing a violin string. The boundary conditions extant at the ends of a string further limit the wave function. Real strings always have finite lengths and are fixed in some fashion at their ends. The displacement sum $y = f + g$ of Equation (4.4) is constrained to have a zero value at all times at the clamping points. Also, when a string is sustained in a steady-state condition by periodic external driving forces, the functions f and g will also have the same frequency as the applied forces but the amplitudes of vibration are determined by the point of application of the force and by boundary conditions at the ends of the string.

Example 1: String Clamped at Both Ends. Given a string of length P clamped rigidly at $x = 0$ and $x = L$. The solutions $y_1 = f(ct - x)$ and $y_2 = g(ct + x)$ are no longer arbitrary, and their sum must be zero at all times, i.e.,

$$f(ct - 0) + g(ct + 0) = 0$$

or

$$f(ct - 0) = -g(ct + 0) \tag{4.5}$$

The two functions must be of the same form but with opposite signs; and we can now rewrite Equation (4.5) as

$$y(x, t) = f(ct - x) - f(ct + x) \tag{4.6}$$

The first term on the right-hand side of Equation (4.6) represents a wave traveling to the right (in the positive x-direction) and the second term a wave moving leftward (in the negative x-direction).

4.5 Reflection of Waves at Boundaries

The reflection process at the boundary $x = 0$ can be viewed as one in which a second wave does not pass the boundary point but is considered to reflect back, generating a similarly shaped wave of opposite displacement traveling in the positive x-direction. The presence of a fixed point at $x = L$ results in another reflection.

In this case, the wave traveling in the positive x-direction reflects back as a similar wave of opposite displacement moving in the negative x-direction. The major result of these two reflections is that the motion of the free vibration becomes periodic. A pulse leaving $x = 0$ reaches $x = L$ after an interval of L/c seconds. There, it is reflected and returns to the origin where it again undergoes a reflection after a time lapse of $2L/c$ seconds. The shape of the pulse after its second reflection is identical with that of the original pulse. This periodicity has resulted from the specified boundary conditions, i.e., fixed points at $x = 0$ and $x = L$.

4.6 Simple Harmonic Solutions of the Wave Equation

Simple harmonic vibrations frequently occur in nature, and we shall now consider a simple harmonic motion (SHM) propagating along a string. Any vibration of the string, however complex, can be resolved into an equivalent array of simple harmonic vibrations. This resolution of complex vibration into a series of SHMs is not a mere mathematical exercise but constitutes the phenomenal principle of how the ear functions. The ear breaks down a complex sound into its simple harmonic components. This capability permits us to distinguish the differences between different voices and musical instruments. A piano sounding a note will sound differently from the same note played by an oboe. If all the frequencies present in the sound consist of a fundamental tone plus its harmonics, they will sound more harmonious than in the situation where the frequencies are not related so simply to each other.

The displacement of any point on the string exciting a SHM of angular frequency ω can be depicted by the special solution to Equation (4.3):

$$y = a_1 \sin(\omega t - kx) + a_2 \sin(\omega t + kx) + b_1 \cos(\omega t - kx) + b_2 \cos(\omega t + kx) \tag{4.7}$$

where a_1, a_2, b_1, b_2 are arbitrary constants and k is the wavelength constant given by

$$k \equiv \omega/c$$

Applying the boundary condition $y(0, t) = 0$ (i.e., $y = 0$ at $x = 0$), which describes a fixed point, Equation (4.7) reduces to

$$(a_1 + a_2) \sin \omega t = -(b_1 + b_2) \cos \omega t \tag{4.8}$$

Because this equation applies to all values of t, the following relations between the constants must exist

$$a_1 + a_2 = 0, \quad b_1 + b_2 = 0 \quad \text{or} \quad a_1 = -a_2, \quad b_1 = -b_2$$

The two limitations for the arbitrary constants of Equation (4.7) are equivalent to the single restriction of Equation (4.5) in the general solution of wave equation (4.3). The two waves must be of equal and opposite displacements and must

therefore differ in phase by π radians at $x = 0$. With these restrictions Equation (4.7) becomes

$$y = a_1 [\sin(\omega t - kx) - \sin(\omega t + kx)] + b_1 [\cos(\omega t - kx) - \cos(\omega t + kx)] \tag{4.9}$$

Making use of trigonometric transformations for the sine and cosine terms we simplify Equation (4.9) to

$$y = [-2a_1 \cos \omega t + 2b_1 \sin \omega t] \sin kx$$

Thus y is expressed as a product of a time-dependent term and a coordinate-dependent term.

Applying the boundary condition of $y(L, t) = 0$ (i.e., the displacement is zero at end point $x = L$) adds yet another restriction

$$\sin kL = \sin n\pi \quad \text{where } n = 1, 2, 3, \ldots$$

Consequently, the string cannot vibrate freely at any random frequency; it can only vibrate with a discrete set of frequencies given by

$$\omega_n = n\pi c/L$$

where $n = 1, 2, 3, \ldots$, or, in terms of frequency:

$$f_n = nc/(2L) \tag{4.10}$$

4.7 Standing Waves

The boundary conditions at $x = 0$ and at $x = L$ reduced the general SHM solution Equation (4.7) to a pattern of standing waves on the string. At the lowest or *fundamental* frequency, where $n = 1$, the displacement is given by

$$y_1 = (A_1 \cos \omega t + B_1 \sin \omega t) \sin k_1 x \tag{4.11}$$

Here $k_1 = \pi/L$, and A_1 and B_1 are arbitrary constants of which numerical values are established by the initial conditions, i.e., the type of excitation imparted to the string at $t = 0$. This *fundamental mode* of vibration is associated with the fundamental (or *first harmonic*) frequency $f_1 = c/2L$. The nth mode of vibration corresponding to the nth harmonic frequency is represented by

$$y_n = (A_n \cos \omega_n t + B_n \sin \omega t) \sin k_x \tag{4.12}$$

and the frequency is $f_n = nc/2L$, i.e., n times the fundamental frequency. The constants A_n and B_n are determined by the initial excitation.

In evaluating the term $\sin k_n x = \sin n\pi x/L$, we recognize that displacement $y_n = 0$ occurs for all values of x when $\sin n\pi x/L = 0$, i.e.,

$$n\pi x/L = m\pi \quad \text{where } m = 0, 1, 2, 3, \ldots.$$

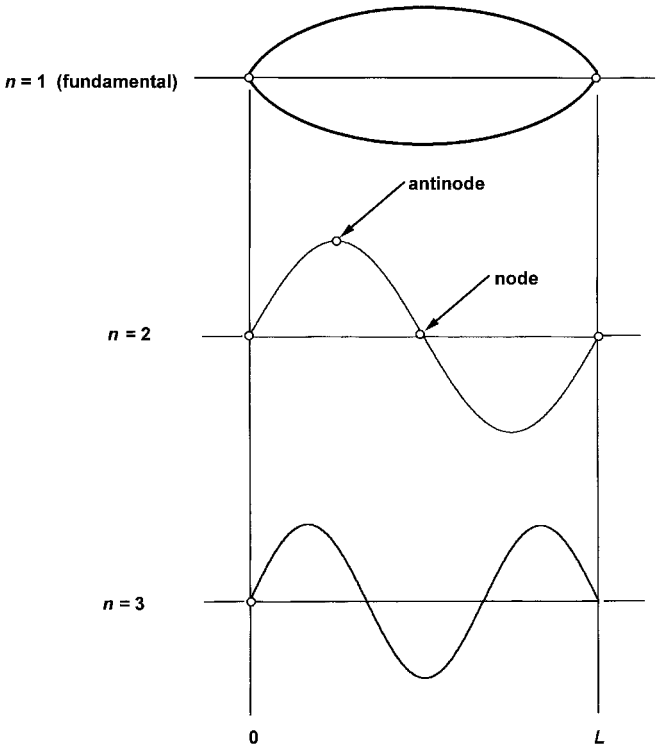


FIGURE 4.3. Different modes of vibrations for a string for the fundamental and the first two harmonics.

The cases for which $m = 0$ and m $= n$ correspond to the boundary conditions at the fixed points at the two ends of the string. However, there are additional $(n - 1)$ locations, called *nodal points* or *nodes*, where the displacement produced by the nth harmonic mode of vibration remains at zero, as illustrated in Figure 4.3. This situation may be viewed as one in which the harmonic wave moving in the positive x-direction cancels precisely at the nodal points at all times t the harmonic wave moving in the opposite direction. Because the points of zero displacement remain fixed, the resultant wave pattern constitutes what is known as *standing waves*. The distance between nodal points for the nth harmonic mode of operation is L/n,and the points of maximum vibrational amplitudes are referred to as *antinodes* or *loops*.

Let us take a "snapshot" of the vibrating string at a particular time for the sixth harmonic mode (Figure 4.4). The displacement of the string frozen in time occurs as a sinusoidal function of x.This function repeats itself every length $2L/n$ of the x-coordinate, which, in turn, is equal to the wavelength λ_n of the harmonic waves. In this case of the sixth harmonic mode, the repetition occurs every $L/3$ of the string length. The wavelength is related to the velocity of propagation by

$$\lambda_n = c/f_n$$

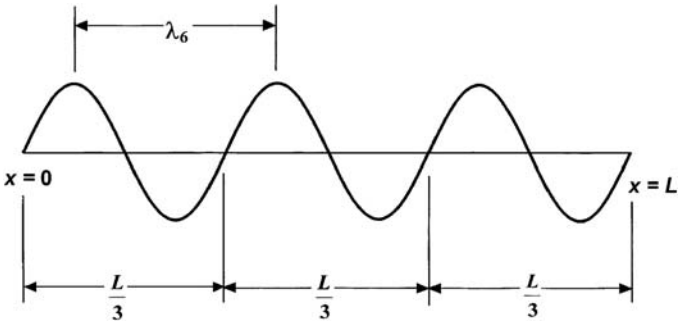


FIGURE 4.4. The sixth harmonic mode of a vibrating string stretched between $x = 0$ and $x = L$.

where f_n is the frequency of vibration in the nth mode and the wavelength constant by $k_n = 2\pi/\lambda_n$. From Equation (4.10) we derive

$$\lambda_n = 2L/n \tag{4.13}$$

that is, the wavelength is twice the nodal distance of the associated wave pattern.

4.8 The Effect of Initial Conditions

The complete general solution to the general harmonic wave equation for a freely vibrating string rigidly clamped at its ends contains all the individual modes of vibration described by Equation (4.12). It is expressed as

$$y_n = \sum_{n=1}^{\infty} (A_n \cos \omega_n t + B_n \sin \omega_n t) \sin k_n t \tag{4.14}$$

where A_n and B_n are the amplitude coefficients dependent on the method of exciting the string to vibrate. The actual amplitude of the nth mode is

$$a_n = \sqrt{A_n^2 + B_n^2}$$

Consider the initial condition at $t = 0$ at the time when the string is displaced from its normal linear configuration so that the displacement $y(x,t)$ at each point of the string is given by the function

$$y(x, 0) = y_0(x)$$

The corresponding velocity $v(x,t)$ is given for $t = 0$ by

$$v(x, 0) = \frac{\partial y(x, 0)}{\partial t} = v_0(x)$$

In order that Equation (4.14) represents the string at all times, it also must describe the displacement at $t = 0$ and therefore is written as

$$y_0(x) = y(x, 0) = \sum_{n=1}^{\infty} A_n \sin k_n x \tag{4.15}$$

The derivative of y with respect to time must also represent the velocity at $t = 0$,

$$v_0(x) = v(x, 0) = \sum_{n=1}^{\infty} \omega_n B_n \sin k_n x \tag{4.16}$$

We apply Fourier's theorem[1] to Equation (4.14) in order to obtain

$$A_n = \frac{2}{L} \int_0^L y_0(x) \sin k_n x \, dx \tag{4. 17a}$$

and then apply the theorem to Equation (4.16) to get

$$B_n = \frac{2}{\omega_n L} \int_0^L v_0(x) \sin k_n x \, dx \tag{4.17b}$$

Example 2: String Pulled and Suddenly Released. Consider a string that is plucked by pulling it at its center a distance d and then is suddenly released at instant $t = 0$. In such a case, $v_0(x) = 0$ and all the coefficients B_n will be zero. The coefficients A_n are given by

$$\begin{aligned} A_n &= \frac{2}{L}\left[\int_0^{\frac{L}{2}} \frac{2dx}{L} \sin k_n x \, dx + \int_{\frac{L}{2}}^{L} 2\frac{d}{L}(L - x) \sin k_n \, dx\right] \\ &= \frac{8d}{n^2 \pi^2} \sin \frac{n\pi}{2} \end{aligned} \tag{4.18}$$

---

[1] The theorem states that a complex vibration of period T can be represented by a displacement $x = f(t)$ written in terms of a harmonic series

$$\begin{aligned} \boldsymbol{x} = f(t) = A_0 + A_1 \cos \omega t + A_2 \cos 2\omega t + \cdots A_n \cos n\omega t + \cdots + B_1 \sin \omega t \\ + B_2 \sin 2\omega t + \cdots B_n \sin n\omega t + \ldots \end{aligned}$$

where $\omega = 2\pi/T$ and the constants are given by

$$A_0 = \frac{1}{T} \int_0^T f(t) \, dt$$

$$A_n = \frac{2}{T} \int_0^T f(t) \cos n\omega \, dt$$

$$B_n = \frac{2}{T} \int_0^T f(t) \sin n\omega \, dt$$

Therefore, all even modes $n = 0, 2, 4, \ldots$ have

$$A_2 = A_4 = A_6 = \cdots = 0$$

and the odd modes result in non-zero A_n coefficients:

$$A_1 = \frac{8d}{\pi^2}, \quad A_3 = -\frac{8d}{9\pi^2}, \quad A_5 = \frac{8d}{25\pi^2}, \quad \text{etc.} \tag{4.19}$$

The amplitudes of the various harmonic modes are given by the numerical values of A_n. In general, it may be observed that no harmonics are generated having a node at the point of the string initially plucked. As the nodal number n increases, the associated amplitudes decrease from the value of the fundamental amplitude, i.e., the fundamental A_1 is 9 times larger than A_3 and 25 times larger than A_5, and so on.

Example 3: Sharp Blow Applied to String. If the string is struck a sharp blow (as opposed to being plucked, as described above), $v_0(x,0)$ has nonzero values but no initial displacement exists. Then all the coefficients A_n are zero and the coefficients B_n are given by Equation (4.17b). A common example of a struck string is the impact of a piano hammer striking a string. It is interesting to note that pianos are designed in such a way that the impact point of the hammer is one-seventh of the distance from one end of the string, thus eliminating the seventh harmonic (which would have produced a discordant sound).

4.9 Energy of Vibrating String

In any nondissipative system the total energy content remains constant, equal to the value of the maximum kinetic energy. For the nth mode of vibration the maximum value of the kinetic energy of a segment of length dx is

$$dE_n = \frac{\omega_n \delta}{2}\left(A_n^2 + B_n^2\right)\sin^2 k_n x \, dx \tag{4.20}$$

which is established by simply applying the relation $dE = (mv)dv/2$ in conjunction with Equation (4.6) and the fact that the mass of the string element is given by

$$m = \delta \, dx$$

where δ is the linear density. With integration over the variable x from 0 to L, the maximum kinetic energy of the string is

$$E_n = \frac{\omega_n^2 \delta}{2}\left(A_n^2 + B_n^2\right)\frac{L}{2} = \frac{m}{4}\omega_n^2\left(A_n^2 + B_n^2\right)$$

Here m is the total mass of the string and $(A_n^2 + B_n^2)$ is the square of the maximum displacement of the nth harmonic. In a conservative system (which describes the dissipationless vibrating string) the maximum potential energy is also equal to the maximum kinetic energy of the system. From Equation (4.20) the energy of the

nth mode of vibration is

$$E_n = \frac{m}{4}\omega_n^2 A_n^2 = \frac{m}{4}\left(\frac{n\pi c}{L}\right)^2 \left(\frac{8d}{n^2\pi^2}\right)^2 = \frac{16md^2c^2}{n^2\pi^2L^2}$$

It becomes apparent that as n increases, the energy of the nth mode lessens. For example, the energy of the third harmonic is one-ninth that of the fundamental mode.

4.10 Forced Vibrations in an Infinite String

While this may appear to be a purely academic exercise, the simple case of a transverse sinusoidal force on an idealized string of infinite length can provide insight into the forced vibrations of finite strings and the transmission of acoustic waves.

An ideal string of infinite length subject to a tension T receives a transverse driving force $T\cos\omega t$ at the string end $x = 0$. The end at $x = 4$ is rigidly clamped but the point $x = 0$ is rigid only in the x-direction, being free to move in the y-direction, a support that can be approximated by a pivoted lever as shown in Figure 4.5. Thus, the driving force can move the lever as well as the string. We shall neglect the mechanical impedance of the pivoted lever (or hinge) that is deemed to have no friction and no stiffness. No waves are reflected from the far end $x = 4$, and hence no waves travel in the negative x-direction. The displacement of the string can now be described by the general solution containing only the expression for a harmonic wave traveling in the positive x-direction:

$$y = a_1 \sin(\omega t - kx) + b_1 \cos(\omega t - kx)$$

or in a complex format:

$$\mathbf{y} = \mathbf{A}\, e^{i(\omega t - kx)} \tag{4.21}$$

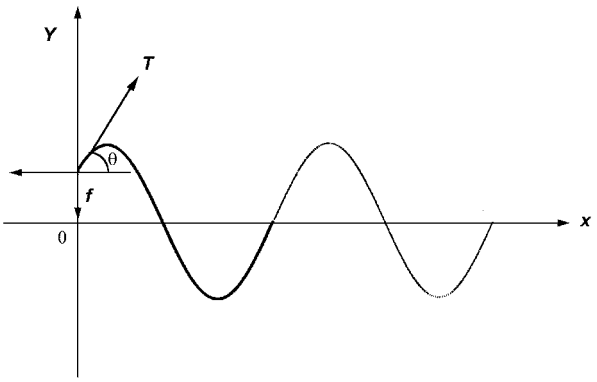


FIGURE 4.5. Forces acting at one end of an infinite string.

where $\mathbf{A}$ is a complex constant of which magnitude equals the displacement amplitude of the wave motion and whose phase angle renders the difference in phase between the motion of the string and the driving force.

In complex format the harmonic driving force can be written as

$$\mathbf{f} = \mathbf{F}e^{i\omega t} \tag{4.22}$$

In Figure 4.5, the driving force is shown being applied to the string at an angle θ the string makes with the horizontal. This angle is given by

$$\tan\theta = \left(\frac{\partial y}{\partial x}\right)_{x=0}$$

The force exerted in the horizontal direction at the support at the end of the string is $-T\cos\theta$. Because the displacements are assumed small, $\cos\theta \approx 1$, and the magnitude of this force in the horizontal direction is essentially equal to tension T in the string. From similar considerations, the transverse force exerted by the support on the string is $-T\sin\theta$, approximated by

$$\mathbf{f} = -T\sin\theta = -T\left(\frac{\partial y}{\partial x}\right)_{x=0} \tag{4.23}$$

Equation (4.23) indicates that for any applied transverse force the shape of the string at $x = 0$ will vary. Inserting $\mathbf{f}$ and $\mathbf{y}$ from Equations (4.21) and (4.22) into Equation (4.23) yields for this boundary condition

$$\mathbf{F}e^{i\omega t} = -T(-ik)\mathbf{A}e^{i[\omega t - k(0)]}$$

or

$$\mathbf{A} = \frac{\mathbf{F}}{ikT} \tag{4.24}$$

The term $\mathbf{F}/kT$ represents the magnitude of this complex amplitude $\mathbf{A}$. Inserting Equation (4.24) into Equation (4.21) and then differentiating with respect to time results in the complex velocity $\mathbf{v}$

$$\mathbf{v} = \mathbf{F}\left(\frac{c}{T}\right)e^{i(\omega t - kx)}$$

The mechanical (or wave) impedance $\mathbf{Z_s}$ of the string is defined as the ratio of the driving force to the transverse velocity of the string at $x = 0$:

$$\mathbf{Z_s} = \frac{T}{c} = \sqrt{T\delta} = \delta c$$

It turns out that $\mathbf{Z_s}$ is a real quantity, with no imaginary load. The mechanical load presented by the string to the driving force is purely resistance. The input impedance exists as a function of the linear density δ and the tension applied to the string, and it does *not* depend on the applied driving force; this means it is a property characteristic of the string, not the wave propagation in the string. The input, often termed *characteristic* or *mechanical impedance* (or *resistance*)

is analogous to the characteristic electrical impedance of an infinite transmission line.

The average power input to the string is found from the average value of the instantaneous power $\boldsymbol{W} = fv$ evaluated at $x = 0$, or

$$W = \frac{F^2}{2\delta c} = \frac{\delta c V_0^2}{2}$$

where V_0 is the velocity amplitude at $x = 0$.

4.11 Strings of Finite Lengths: Forced Vibrations

In the case of a finite string, the reflections from the far end generate frequencies that cause the input impedance to change greatly with the frequency of the driving force. If the support at finite $x = L$ is fully rigid and no dissipative forces occur in the string, the input impedance becomes a pure reactance, and no power is consumed in the string.

Because the complex expression for transverse waves on a finite string now needs a term descriptive of the reflected wave, Equation (4.21) needs to be rewritten as

$$\mathbf{y} = \mathbf{A}e^{i(\omega t - kx)} + \mathbf{B}e^{i(\omega t + kx)} \tag{4.25}$$

for all times t. The boundary condition at $x = 0$ is

$$Fe^{i\omega t} = -T\left(\frac{\partial y}{\partial x}\right)_{x=0} \tag{4.26}$$

for all values of t. Inserting Equation (4.25) into Equation (4.26) yields

$$F = -T(-ik\mathbf{A} + ik\mathbf{B}) \tag{4.27}$$

Applying $\boldsymbol{y}(\boldsymbol{L}, t) = 0$ for the rigid clamp at $x = L$, Equation (4.25) becomes

$$0 = \mathbf{A}e^{-ikL} + \mathbf{B}e^{ikL} \tag{4.28}$$

Solving the Equations (4.27) and (4.28) for $\boldsymbol{A}$ and $\boldsymbol{B}$ results in

$$\mathbf{A} = \frac{F}{ikT} \cdot \frac{e^{ikL}}{e^{ikL} + e^{-ikL}} = \frac{Fe^{ikL}}{2ikT\cos kL}$$

and

$$\mathbf{B} = -\frac{F}{ikT} \cdot \frac{e^{-ikL}}{e^{ikL} + e^{-ikL}} = -\frac{Fe^{-ikL}}{2ikT\cos kL}$$

Substituting the above constants into Equation (4.26) yields

$$y = \frac{Fe^{i\omega t}}{ikT} \cdot \frac{e^{ik(L-x)} - e^{ik(L-x)}}{2\cos kL} = \frac{Fe^{i\omega t}}{kT} \cdot \frac{\sin k(L-x)}{\cos kL} \tag{4.29}$$

The real portion of Equation (4.29) graphs a pattern of standing waves on the string with nodes occurring at those points where $\sin(L - x) = 0$, in addition to

$\boldsymbol{y}(0, L) = 0$. The displacement at $x = 0$, however, has an amplitude

$$\mathbf{y}_0 = (\mathbf{F} \tan kL)/k\boldsymbol{T} \tag{4.30}$$

Singularities of Equation (4.30) occur when $\cos kL = 0$, i.e.,

$$kL = \frac{\omega L}{c} = \frac{(2n-1)\pi}{2}, \qquad n = 1, 2, 3, \ldots$$

or

$$\omega_n = \frac{(2n-1)\pi c}{2L}$$

and

$$f_n = \frac{(2n-1)}{4L} c$$

These singularities connote infinite amplitudes which, of course, do not occur in real strings, because dissipative forces neglected in the foregoing analysis do actually exist. But the amplitudes do achieve maximum values at these frequencies. In a similar fashion we can ascertain the minimum amplitudes from the condition $kL = \pm 1$, i.e.,

$$kL = \omega_n L/c \qquad n = 1, 2, 3, \ldots.$$

and

$$\omega_n = \frac{n\pi c}{L} \qquad \text{or} \qquad f_n = \frac{nc}{2L}$$

The minimum amplitudes decrease progressively with increasing frequencies. In fact, on comparing with Equation (4.10) it is noted that the frequencies of minimum amplitudes are identical to those of the free-string vibration, and the term *antiresonance* has been applied to describe those frequencies. Differentiation of Equation (4.29) with respect to time t yields the complex velocity $\mathbf{v}$ of the string,

$$\mathbf{v} = \frac{Fe^{i\omega t}}{T/c} \cdot \frac{\sin k(L-x)}{\cos kL}$$

The input mechanical impedance then becomes

$$\mathbf{Z}_s = \frac{fe^{i\omega t}}{v} = \frac{T}{ic} \frac{\cos kL}{\sin kL} = -i\delta c \cot kL$$

which exists as a pure reactance, with no power absorbed by the string. The amplitude of vibration is a maximum at $\cot kL = 0$, which occurs at the frequency given by $f_n = nc/2L$. For extremely low frequencies the input impedance has the limits

$$\mathbf{Z}_s = -\frac{i\delta c}{kL} = -\frac{iT}{\omega L}$$

4.12 Real Strings: Free Vibration

Real strings manifest some degree of stiffness, causing the observed frequencies to be higher than the theoretical values for idealized strings. This results from the presence of elastic boundary forces augmenting the action of tensile forces considered previously, with the net effect of increased restoring forces. The presence of stiffness exerts a greater influence with increasing frequency, and the overtones of a stiff string no longer constitute an exact harmonic series.

We must also be attentive to the fact that clamping at the ends of the string may not be exactly rigid and that yielding can occur at these points. The wave impedance at the ends will constitute the transverse mechanical impedance of the supports.

Consider a case where the left end of a finite string at $x = 0$ is attached to a pivot representing the slightly loose clamp. The transverse force f_0 exerted by the string on the hinge is

$$f_0 = T \sin\theta \approx T \left(\frac{\partial \mathbf{y}}{\partial x} \right)_{x=0}$$

Here $\mathbf{y}$ is the complex expression for the transverse wave on the string, as given by Equation (4.25). Because the motion of the swivel must match that of the end of the string, the velocity is

$$v_0 = \left(\frac{\partial y}{\partial t} \right)_{x=0} = i\omega y_0$$

Let $\boldsymbol{z_0}$ denote the transverse mechanical impedance of the hinge at $x = 0$

$$z_0 = \frac{\mathbf{f}_0}{\mathbf{v}_0} = \frac{T}{i\omega} \cdot \frac{\left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right)_{x=0}}{\mathbf{y}_0}$$

and hence the boundary condition at $x = 0$ becomes

$$\mathbf{y}_0 = \frac{T(\partial y/\partial x)_{x=L}}{i\omega \mathbf{Z}_L}$$

where $\mathbf{Z}_L$ is the mechanical impedance of the swivel located at $x = L$. If the supports were truly rigid, $\mathbf{Z_0} = \mathbf{Z_1} = 4$, and the boundary conditions reduce to $\boldsymbol{y_0} = \boldsymbol{y_L} = 0$.

References

Fletcher, Neville H. and Rossing, Thomas D. 1998. *The Physics of Musical Instruments*, 2nd ed. New York: Springer-Verlag, Chapter 2. (This text provides an excellent exposition on the physical principles of musical instruments.)

Jean, Sir James. 1968. *Science and Music*. New York: Dover Publications, Chapter 3. (Although rather skimpy on the mathematical details, the exposition provides a good insight into the physics of vibrating string to the reader.)

Kinsler, Lawrence E., Frey, Austin R., Coppens, Alan B., and Sanders, James V. 1999. *Fundamentals of Acoustics*, 4th ed. New York: John Wiley & Sons, Chapter 2. (Still a good textbook which dates back to 1950.)

Morse, Philip M. 1982 (reprint from 1948 edition published by McGraw-Hill). *Vibration and Sound*. Woodbury, NY: Acoustical Society of America, Chapter 3.

Morse, Philip M. and Ingard, K. Uno. 1968. *Theoretical Acoustics*. New York: McGraw-Hill, Chapter 4.

Reynolds, Douglas R. 1981. *Engineering Principles of Acoustics, Noise and Vibration Control*. Boston: Allyn and Bacon, pp. 213–224.

Problems for Chapter 4

1. Show by direct substitution that each of the following expressions constitutes solutions of the wave equation:
   (a) $f(x - ct)$
   (b) $\ln [f(x - ct)]$
   (c) $A(ct - x)^3$
   (d) $\sin [A(ct - x)]$
2. Show which of the followings are solutions and *not* solutions to the wave equation:
   (a) $B(ct - x^2)$
   (b) $C(ct - c)t$
   (c) $\mathrm{A} + B \sin(ct + x)$
   (d) $A \cos^2(ct - x) + B \sin(ct + x)$
3. Plot (by computer if possible) the expression $y = Ae^{-B(ct-x)}$ for times $t = 0$ and $t = 1.0$, with $A = 6$ cm, $B = 4$ cm^{-1}, and $c = 3$ cm/s. Discuss the physical significance of these curves.
4. Consider a string of density 0.05 g/cm, in which a wave form $y = 4 \cos (5t - 3x)$ is propagating. x and y are expressed in centimeters, and time t in seconds.
   (a) Determine the amplitude, phase speed, frequency, wavelength, and the wave number.
   (b) Find the particle speed of the string element at $x = 0$ at time $t = 0$.
5. A string is stretched with tension T between two rigid supports located at $x = 0$ and $x = L$. It is driven at its midpoint by a force $F \cos \omega t$.
   (a) Determine the mechanical impedance at the midpoint.
   (b) Establish that the amplitude of the midpoint is given by $F \tan (kL/2)/(2kT)$.
   (c) Find the amplitude of displacement at the quarter point $x = L/4$.
6. Determine the mechanical impedance with respect to the applied force driving a semi-infinite string at a distance L from the rigid end. What is the significance of the individual terms in the expression for mechanical impedance?
7. Consider a string of density 0.02 kg/m that is stretched with a tension of 8 N from a rigid support to a device producing transverse periodic vibrations at the other end. The length of the string is 0.52 m. It is noted that for a specific driving frequency, the nodes are spaced 0.1 m apart and the maximum amplitude is 0.022 m. What are the frequency and the amplitude of the driving force?

8. A device that has a constant speed amplitude $u(0,t) = U_0 e^{i\omega t}$, where U_0 is a constant, drives a forced, fixed string.
   (a) Find the frequencies of the maximum amplitude of the standing wave.
   (b) Repeat the problem for a constant displacement amplitude $y(0, t) = Y_0 e^{i\omega t}$.
   (c) Compare the results of (a) and (b) with the frequencies of mechanical resonances for the forced fixed string. Does the mechanical amplitude coincide with the maximum amplitude of the motion?
9. Consider a string fixed at both ends, with specified values of ρ_L, c, L, f, and T. Express the phase speed c' in terms of c and the fundamental resonance f' in terms of f if another string of the same materials is used but
   (a) the length of the string is doubled.
   (b) the density per unit length is doubled.
   (c) the cross-sectional area is doubled.
   (d) the tension is reduced by half.
   (e) the diameter of the string is doubled.
10. Consider a string of length L that is plucked at the location $L/3$ by producing an initial displacement δ and then suddenly releasing the string. Find the resultant amplitudes of the fundamental and the first three harmonic overtones. Draw (through computer techniques, if possible) the wave forms of these individual waves and the shape of the string occurring from the linear combination of these waves at $t = 0$. Redo this problem for time $t = L/c$, where c represents the transverse wave velocity of the string.
11. A string of length 1.0 m and weighing 0.03 kg has a mass of 0.15 kg hanging from it.
   (a) Find the speed of transverse waves in the string (*Hint*: neglect the weight of the string in establishing the tension in string).
   (b) Determine the frequencies of the fundamental and the first overtone modes of the transverse vibrations.
   (c) For the first overtone of the string, compare the relative amplitude of the string's displacement at the antinode with that of the mass.
12. A string having a linear density of 0.02 kg/m is stretched to a tension of 12 N between rigid supports 0.25 m apart. A mass of 0.002 kg is loaded on the string at its center.
   (a) Find the fundamental frequency of the system.
   (b) Find the first overtone frequency of the system.
13. A standing wave on a fixed–fixed string is given by $y = 3 \sin(\pi x/4) \sin 2t$. The length of the string is 36 cm and its linear density 0.1 gm/cm. The units of x and y are in centimeters, and t is given in seconds.
   (a) Find the frequency, phase speed, and wave number.
   (b) Determine the amplitude of the particle displacement at the center of the string and at $x = L/4$ and $x = L/3$.
   (c) Find the energy density for those points, and determine how much energy there is in the entire string?

5
Vibrating Bars

5.1 Introduction

The theory underlying the physical operation of vibrating bars is of great interest to acousticians, because a number of acoustic devices employ longitudinal vibrations in bars and frequency standards are established by producing sounds of specific pitches in circular rods of different lengths. The analysis of vibrating bars facilitates our understanding of acoustic waves through fluids, for the mathematical expressions governing the transmission of acoustic plane waves through fluid media are similar to those describing the travel of compression waves through a bar. Moreover, if the fluid is confined inside a rigid pipe, the boundary conditions bear a close correlation to those of a vibrating bar. An example of devices falling into the category of vibrating bars include piezoelectric crystals which are cut so that the frequency of the longitudinal vibration in the direction of the major axis of the crystal may be used to monitor the frequency of an oscillating electric current or to drive an electroacoustic transducer.

The principal mode of sound transmission in bars is through the propagation of *longitudinal* waves. Here the displacement of the solid particles in the bar occurs parallel to the axis of the bar. The lateral dimensions of a bar are small compared with the length, so the cross-sectional plane can be pictured as moving as a unit. In reality, because of the Poisson effect that generally occurs in solid materials, the longitudinal expansion of the bar results in a lesser degree of lateral shrinkage and expansion; but this lateral motion can be disregarded in very thin bars.

5.2 Derivation of the Longitudinal Wave Equation for a Bar

In Figure 5.1, a bar of length L and uniform cross-sectional area $\hat{A}$ is subjected to longitudinal forces which produce a longitudinal displacement ξ of each of the molecules in the bar. This displacement in long, thin bars will be the same at each point in any specific cross section. If the applied longitudinal forces vary in a wavelike perturbative manner, the displacement ξ is a function of both x and t

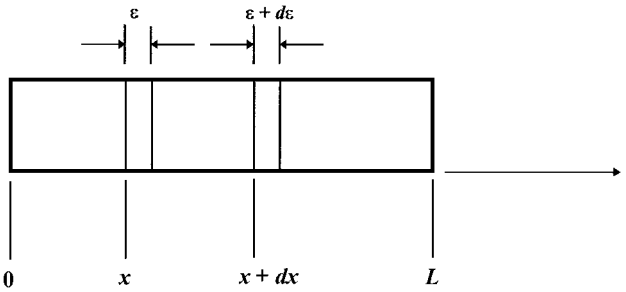


FIGURE 5.1. A bar undergoing longitudinal strain in the x-direction.

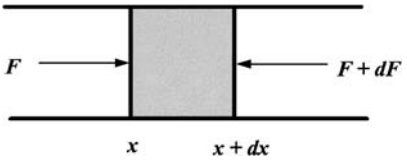


FIGURE 5.2. An element of the bar undergoing compression.

and is fairly independent of lateral coordinates y and z. Thus,

$$\xi = \xi(x, t)$$

The x-coordinate of the bar is established by placing the left end of the bar at $x = 0$, with the right end terminating at $x = L$. Consider an incremental element formed by dx of the unstrained bar positioned between x and $x + dx$ as shown in Figure 5.2. The application of a force in the positive x-direction causes a displacement of the plane at x by a distance ξ to the right and the plane at $x + dx$ by a distance $\xi + d\xi$ also to the right. A force acting in the opposite direction will likewise cause corresponding negatively valued displacements to the left. Because the element dx is small, we can represent the displacement at $x + dx$ by the first two terms of a Taylor series expansion of ξ about x:

$$\xi + d\xi = \xi + \left(\frac{\partial \xi}{\partial x}\right) dx$$

The left end of the element dx has been displaced a distance ξ and the right end a distance $\xi + d\xi$, thus yielding a net increase $d\xi$ in the length of the element given by

$$(\xi + d\xi) - \xi = d\xi = \left(\frac{\partial \xi}{\partial x}\right) dx$$

In solid mechanics the strain ε of an element is defined as the ratio of the change of its length to the original length, i.e.,

$$\varepsilon = \frac{\left(\dfrac{\partial \xi}{\partial x}\right)}{dx} dx = \frac{\partial \xi}{\partial x} \tag{5.1}$$

In the situation in which static forces are applied to a uniform bar, the strain is the same for each point and is time-independent. But we are considering a dynamic case in which the strain in the bar varies with coordinate x and with time t. This type of variation generates a longitudinal wave motion in the bar in a manner analogous to the transverse waves in a string. When a bar undergoes strain, elastic forces are generated inside the bar. These forces act across each cross-sectional plane and essentially constitute reactions to longitudinally applied forces. We let $F_x = F_x(x, t)$ denote these longitudinal forces and adopt the convention that *compressive* forces are represented by *positive* values of F_x and *tensile* forces by *negative* values of F_x. The stress σ in the bar is defined by

$$\sigma = F_x/\hat{A}$$

Here $\hat{A}$ is the cross-sectional area of the bar. We can now apply Hooke's law

$$\sigma = \frac{F_x}{\hat{A}} = -E\varepsilon = -E\frac{\partial \xi}{\partial x} \tag{5.2}$$

where E is the elastic constant or *Young's modulus*, a property characteristic of the material constituting the bar. Table A in Appendix lists the values of Young's moduli for a number of commonly used materials. Because E must always have a positive value, a negative sign is introduced in Equation (5.2) to accommodate the fact that a positive stress (compression) results in a negative strain, and a negative stress (tension) in a positive strain. We rewrite Equation (5.2) to express force F_x at point x as follows:

$$F_x = -E\hat{A}\varepsilon = -E\hat{A}\frac{\partial \xi}{\partial x} \tag{5.3}$$

Unlike the static case where the strain $\varepsilon = \partial\xi/\partial x$ and hence the force F_x, remains constant throughout the bar, both the strain and F_x vary in the dynamic case, and a net force acts on element dx. F_x represents the internal force at x, and so $F_x + (\partial F_x/\partial x)dx$ constitutes the force at $x + dx$. The net force acting to the right becomes

$$dF_x = F_x - \left(F_x + \frac{\partial F_x}{\partial x}dx\right) = -\frac{\partial F_x}{\partial x}dx \tag{5.4}$$

Combining Equations (5.3) and (5.4) results in

$$dF_x = E\hat{A}\frac{\partial^2 \xi}{\partial x^2}dx \tag{5.5}$$

The volume of the element dx is given by $\hat{A}\,dx$, and therefore the mass is $\rho\hat{A}\,dx$, where ρ denotes the density (kg/m^3) of the bar material. Applying Newton's equation of motion, with acceleration $\partial^2\xi/\partial t^2$, to Equation (5.5), we obtain

$$\rho\hat{A}\;dx\frac{\partial^2 \xi}{\partial t^2} = E\hat{A}\frac{\partial^2 \xi}{\partial x^2}dx$$

Setting

$$c^2 = E/\rho \tag{5.6}$$

we now obtain the one-dimensional longitudinal wave equation:

$$\frac{\partial^2 \xi}{\partial t^2} = c^2 \frac{\partial^2 \xi}{\partial x^2} \tag{5.7}$$

Equation (5.7) corresponds to Equation (4.3) for the transverse motion of the string, with the longitudinal displacement ξ assuming the role of the transverse displacement y. We note that Equation (5.7) is identical to Equation (2.3), and we have derived in this section the wave equation (5.7) that applies to acoustic propagation in a linearly elastic solid.

5.3 Solutions of the Longitudinal Wave Equation

The format of the general solution to Equation (5.7) is identical with that of the solution to Equation (4.3), i.e.,

$$\xi = f(ct - x) + g(ct + x) \tag{5.8}$$

The square root of Equation. (5.6) gives us the wave propagation velocity c:

$$c = \sqrt{\frac{E}{\rho}} \tag{5.9}$$

which indicates that c is a property of the bar material.

Let us write the solution (5.8) in the form of a complex harmonic solution

$$\xi = \mathbf{A}e^{i(\omega t - kx)} + \mathbf{B}e^{i(\omega t + kx)} \tag{5.10}$$

where $\mathbf{A}$ and $\mathbf{B}$ represent complex amplitude constants and $k = \omega/c$ is the wave number. We now assume that the bar is rigidly fixed at both ends; the boundary conditions $\boldsymbol{\xi}(x, t)$ becomes $\boldsymbol{\xi} = 0$ at $x = 0$ and at $x = L$ at all times t. Applying the condition $\boldsymbol{\xi}(0, t) = 0$ yields $\mathbf{A} = -\mathbf{B}$, and Equation (5.10) revises to

$$\xi = \mathbf{A}e^{i\omega t(e^{-ikx} - e^{ikx})} \tag{5.11}$$

The stipulation $\boldsymbol{\xi}(L, t) = 0$ results in

$$e^{-ikt} - e^{ikt} = \frac{2 \sin kL}{i} = 0$$

or equivalently

$$\sin kL = 0$$

which means that

$$k_n L = n\pi, \qquad n = 1, 2, 3, \ldots$$

The allowed modes of vibration possess the radial frequencies ω_n and the corresponding cyclic frequencies f_n given by

$$\omega_n = n\pi c/L \quad \text{or} \quad f_n = nc/2L, \quad n = 1, 2, 3, \ldots$$

In simplifying Equation (5.11), the complex displacement $\boldsymbol{\xi}$ for the nth mode of vibration is

$$\xi_n = i A_n e^{i\omega_n t} \sin k_n x \tag{5.12}$$

The real part of Equation (5.12) is

$$\xi_n = \sin k_n x (A_n \cos \omega_n t + B_n \sin \omega_n t) \tag{5.13}$$

where the real amplitude constants A_n and B_n are related to the complex constant $\mathbf{A}_n$ as follows:

$$2\boldsymbol{A}_n = B_n + i A_n$$

The full solution to Equation (5.7) consists of the sum of all of the individual harmonic solutions, i.e.,

$$\xi = \sum_{n=1}^{\infty} \sin k_n x (A_n \cos \omega_n t + B_n \sin \omega_n t) \tag{5.14}$$

The constants A_n and B_n can be evaluated by using the Fourier analysis described in the last chapter, provided the initial conditions are known with respect to the displacement and the velocity of the bar.

5.4 Other Boundary Conditions

It should be understood that the boundary conditions corresponding to rigid supports are difficult to realize in practice. The free-end condition, on the other hand, can be simulated by supporting the bar on extremely pliant supports placed some distance inward from the ends. The end of the bar can now move freely and no internal elastic force exists at that location. We now apply Equation (5.3), setting $F_x = 0$; this gives rise to the condition $\partial\xi/\partial x = 0$ at the free end. If the bar is free to move at both ends (this is termed the *free–free bar*), the condition $\partial\xi/\partial x = 0$ applied to $x = 0$ in the wave equation, solution (5.10) yields

$$\mathbf{A} = \mathbf{B}$$

with the result

$$\xi = Ae^{i\omega t}(e^{-ikx} + e^{ikx}) \tag{5.15}$$

Inserting the condition $\partial\xi/\partial x = 0$ into the above Equation (5.15) for the location $x = L$ yields

$$-e^{-ikL} + e^{ikL} = 0 \quad \text{or} \quad \sin kL = 0$$

The allowable frequencies for a free–free bar are the same as those for the bar fixed at both ends (*fixed–fixed bar*). There will be, however, major differences in the respective wave patterns of the free–free bar and the fixed–fixed bar. Recasting Equation (5.15) by making use of the relation

$$2\cos kx = e^{-ikL} + e^{ikL}$$

we can express the complex displacements corresponding to the nth mode of vibration as

$$\xi = 2\ \mathbf{A}_n e^{i\omega_n t}\cos k_n t$$

The real part of the preceding equation gives the tangible vibrations described by

$$\xi_n = \cos k_n x(A_n \cos \omega_n t + B_n \sin \omega_n t) \tag{5.16}$$

By comparing Equation (5.13) for the fixed–fixed bar with Equation (5.14) for the free–free bar, it will be seen that antinodes exist at the end points for the latter bar in contrast with the nodes that must exist at the end points of the fixed–fixed bar. The nodal patterns for both bars are shown in Figure 5.3. It is of interest to observe that when an antinode exists at the center of the bar, the vibrations are

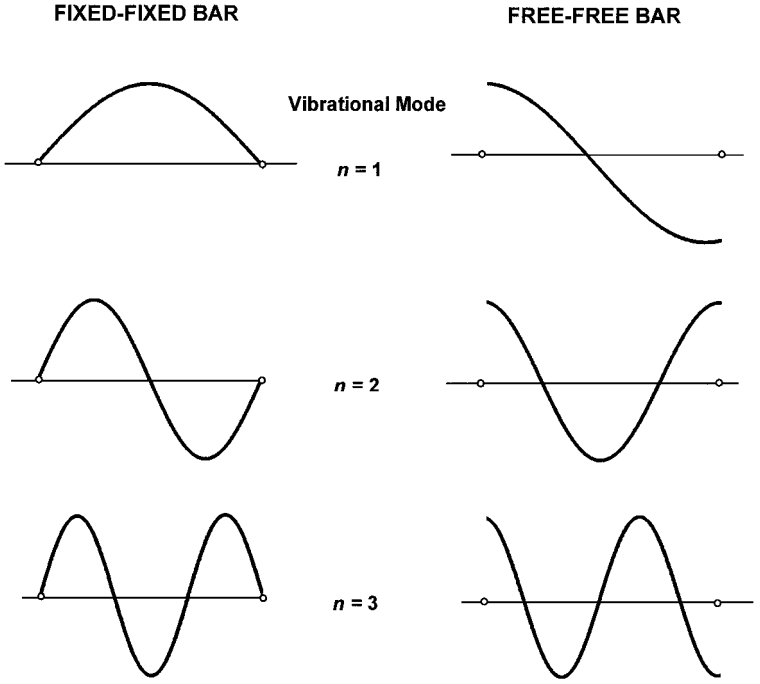


FIGURE 5.3. Typical standing waves for the first three modes of vibration in a fixed–fixed bar and in a free–free bar.

symmetrical with respect to the center; otherwise a node in the center corresponds to asymmetrical vibration.

As with case of the vibrating string, we can rigidly clamp a bar at any one of its nodal positions without affecting the modes of vibrations which have a node at this position. But the vibrations that do not have a node at this position will become suppressed. The nature of the vibration of the free–free bar is such that it is not possible to clamp it at any position that will not eliminate at least some of the allowed modes of vibrations.

The case of a free-fixed bar also makes for an interesting study. One end remains free at $x = 0$, and the other is rigidly clamped at $x = L$. The first condition $\partial\xi/\partial x = 0$ at $x = 0$ leads again to Equation (5.15), while the second condition $\xi = 0$ at $x = L$ yields

$$e^{-ikL} + e^{ikL} = 0 \quad \text{or} \quad \cos kL = 0$$

which means that the allowable frequencies must satisfy

$$k_n L = \frac{\omega_n}{c} L = \frac{\pi}{2}(2n-1), \qquad n = 1, 2, 3, \ldots$$

or

$$\omega_n = (2n-1)\pi\frac{c}{2L}, \qquad f_n = (2n-1)\frac{c}{4l}$$

The fundamental frequency is half that for an otherwise identical free–free bar, and only odd-numbered harmonic overtones exist. The quality of the sound produced by an oscillating free-fixed bar will thus differ from that of a free–free bar, because of the absence of the even harmonics.

5.5 Mass Concentrated Bars

In practical situations a vibrating bar is not truly clamped totally nor is it completely free to move at its ends. It may incorporate some type of mechanical impedance, most commonly as the result of concentrating a certain amount of mass at a certain location. An example is a diaphragm represented as a distributed mass located at one end of a vibrating tube inside a sonar transducer.

As an example let us consider a bar that is unfettered at $x = 0$ and has a loading consisting of a mass m concentrated at $x = L$. The mass is depicted as a point mass so that it does not move as a unit and thus merely sustain waves propagating through it. The boundary condition $\partial\xi/\partial x = 0$ at $x = 0$ again leads us to Equation (5.17), as the result of $\mathbf{A} = \mathbf{B}$. For the boundary condition at $x = L$ we again invoke Newton's law of motion:

$$F_x(L, t) = m\left(\frac{\partial^2\xi}{\partial t^2}\right)_{x=L} \tag{5.17}$$

A positive value of F_x, which compresses the bar, will result in acceleration of the mass in the positive x-direction. Because the mass is rigidly coupled to the bar,

the accelerations of the mass and of the end of the bar should be identical. But if the mass had been concentrated at $x = 0$, a positive (compression) force would correspond to a reaction force to the left on the mass. The appropriate boundary condition for this case would be

$$-F_x(0, t) = m\left(\frac{\partial^2 \xi}{\partial t^2}\right)_{x=0} \tag{5.18}$$

Incorporating the boundary condition Equation (5.18) into Equation (5.16) results in

$$-E\hat{A}e^{i\omega t}(-ike^{-ikt} + ike^{ikt}) = m\mathrm{A}e^{i\omega t}(-\omega^2)(e^{-ikL} + e^{ikL})$$

which rearranges to

$$kE\hat{A}\sin kL = -m\omega^2 \cos kL$$

or

$$\tan kL = -\frac{\omega mc}{E\hat{A}} \tag{5.19}$$

Because Equation (5.19) is a transcendental equation, no explicit solution exists. However, if the mass m is very small, $m \approx 0$ and hence $\tan kL \approx 0$ and $kL \approx n\omega$, both of which constitute the allowed conditions for a free–free bar. This is a result that should occur, since light loadings render a bar nearly free at both ends. At the other extreme, for very heavy mass loadings, the mass behaves very nearly like a rigid support, and the allowed frequencies will approximate those of a free-fixed bar.

In the more general case of intermediate mass loading, it is rather cumbersome to solve by hand the transcendental equation (5.19) through graphic means. However, computer programs such as Mathcad®, MathLab®, Mathematica®, or even a professional-level spread sheet for IBM compatible and Macintosh personal computers can be used to facilitate solutions. Eliminating Young's modulus in Equation (5.19) by applying $E = \rho c^2$ from Equation (5.9) and recognizing that the mass of the bar is given by $m_b = \rho\hat{A}L$, we can rewrite Equation (5.19) as

$$\frac{\tan kL}{kL} = -\frac{m}{m_b} \tag{5.20}$$

The right-hand side of Equation (5.20) is fixed by amount of mass m_b in the bar and the loading mass m located at $x = L$. An example of the solution to Equation (5.20) is given in Figure 5.4 for the case of a steel bar with a mass loading m/m_b of 20%. The longitudinal velocity of sound propagation of steel is 5050 m/s. The fundamental frequency f_1 is found from k_1L through the relation

$$f_1 = \frac{k_1 L}{2\pi}\frac{c}{L}$$

and the higher frequencies are similarly established from

$$f_n = \frac{k_n L}{2\pi}\frac{c}{L}$$

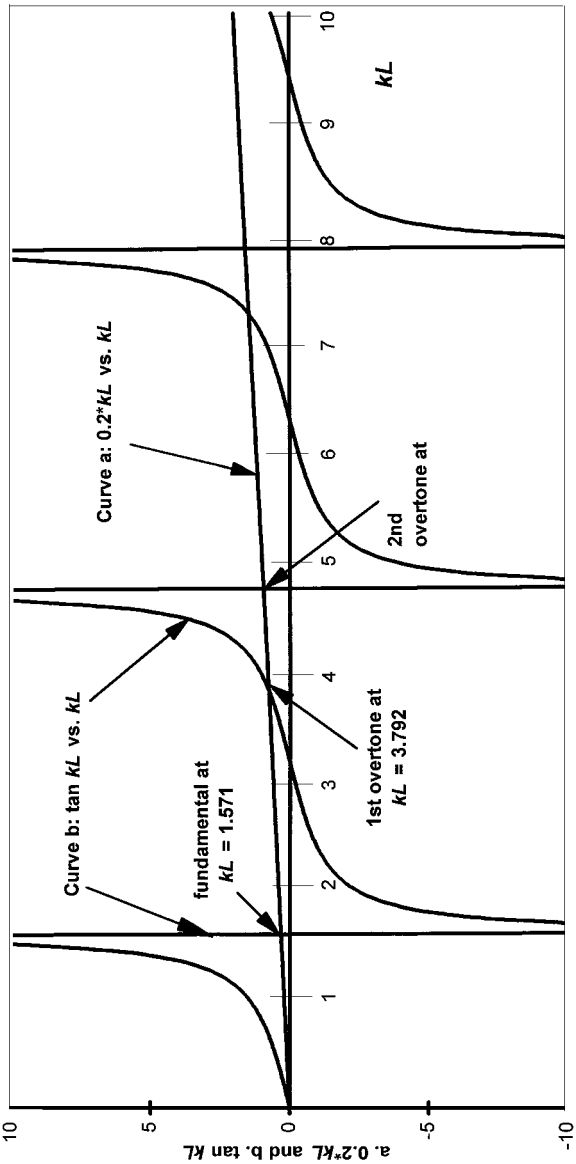


FIGURE 5.4. Plot of $0.2 \times kL$ (Curve a) and $\tan kL$ (Curve b) vs. kL. The intersections of the two sets of curves as shown above provide the first five solutions $k_n L (kL = 0$ being trivial) of the transcendental equation (5.20) for the fundamental and overtones of a vibrating bar with 20% concentrated loading at location $x = L$. The fundamental occurs at $k_1 L = 1.571$, the first overtone at $k_2 L = 3.792$, the second overtone at $k_3 L = 4.712$; and neither of the latter two are harmonics of the fundamental.

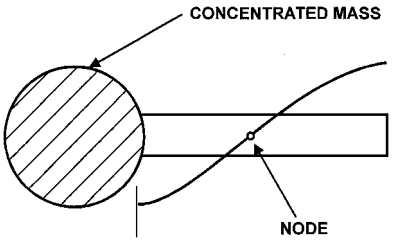


FIGURE 5.5. Location of node in a free mass-loaded bar and fundamental mode of vibration.

But overtones corresponding to the higher frequencies are not harmonics of the fundamental. In the example of Figure 5.4, in which the plots of $(m/m_b)kL$ and tan kL vs. kL yield intersections which constitute the solutions to Equation (5.20), the ratio of the first overtone to the fundamental is $3.792/1.571 = 2.414$, *not* the value of 2.0 that would have specified the overtone to be a harmonic of the fundamental. Solutions exist where the occurrence of non-harmonics overtones could be useful: for example, in a properly mass-loaded loudspeaker, a pure monofrequency input would not result in harmonics that could arise from the driving signal, if at all. The nodes of the vibration in the bar exist at locations where $\cos kx = 0$. The fundamental mode where $kL = 3.792$ engenders a node at $3.792x/L = \pi/2$, or $x = 0.414L$, not $x = 0.5L$ that would normally occur in a second harmonic. In contrast of the free–free bar, the node in a free mass-loaded bar is no longer at the center—it shifts toward the loading mass as shown in Figure 5.5. In this particular case, the bar could be supported at this nodal position without affecting the fundamental mode of vibration.

5.6 General Boundary Conditions for a Freely Vibrating Bar

Consider a freely vibrating bar with arbitrary loadings at each end. We shall establish the normal modes of vibration in terms of the mechanical impedances at both ends of the bar. Designating the mechanical impedance of the support at $x = 0$ by $\mathbf{Z}_{mi0}$ we express the force acting at this support due to the bar as

$$\mathbf{f}_0 = -\mathbf{Z}_{mi0}\,\mathbf{u}(0, t)$$

Here the minus sign is introduced to indicate that a positive (compressive) force generates an acceleration of the support in the negative x-direction. But a positive compressive force at the end $x = L$ causes an acceleration of the other support to the right, in the positive x-direction; the force acting on this support is

$$\mathbf{f}_L = +\mathbf{Z}_{miL}\,\mathbf{u}(L, t)$$

where $\mathbf{Z}_{miL}$ represents the mechanical impedance of the support at $x = L$. The preceding two equations can be restated in terms of particle displacements by

applying Equation (5.3) to supplant the compressive forces and by expressing the particle velocity as $\mathbf{u} = \partial\xi/\partial t$:

$$\left(\frac{\partial \xi}{\partial x}\right)_{x=0} = \frac{\mathbf{Z}_{mio}}{\rho_L c^2}\left(\frac{\partial \xi}{\partial t}\right)_{x=0} \tag{5.21}$$

$$\left(\frac{\partial \xi}{\partial x}\right)_{x=L} = \frac{\mathbf{Z}_{miL}}{\rho_L c^2}\left(\frac{\partial \xi}{\partial t}\right)_{x=L} \tag{5.22}$$

where $\rho_L = \rho\hat{A}$ is the linear density (kg/m) of the bar.

If the loads $\mathbf{Z}_{mio}$ and $\mathbf{Z}_{miL}$ are purely reactive, there is no transient or spatial damping, and hence no loss of acoustical energy occurs. Equation (5.16) constitutes a proper solution. And because no loss of acoustical energy occurs, a wave traveling in the $+x$-direction must equal the energy of a wave moving in the opposite direction. The absolute magnitudes of the complex wave amplitudes must therefore be equal, i.e., $|\mathbf{A}| = |\mathbf{B}|$. The boundary conditions (5.21) and (5.22) establish the phase angles of the complex amplitudes.

But if the mechanical impedances contain some measure of resistive components, a solution more general than that of Equation (5.16) needs to be applied. As in the case of a freely vibrating string terminated by a resistive support, transient (or temporal) damping has to occur in the presence of resistance. The transient behavior of the bar is characterized by a complex angular frequency $\boldsymbol{\omega} = \omega + i\beta$. The real portion of this frequency is the angular frequency ω; the imaginary part represents the transient absorption coefficient β. But no internal losses occur in the bar, so wave equation (5.7) still applies, and we infer the solution

$$\xi(x, t) = (\mathbf{A}e^{-ikx} + \mathbf{B}e^{ikx})e^{i\omega t} \tag{5.23}$$

where $\boldsymbol{\omega}^2 = c^2\mathbf{k}^2$. If the losses are quite small we can use the approximation $\boldsymbol{\omega} \approx c\mathbf{k}$ to simplify the solution. Applying boundary conditions (5.21) for $x = 0$ and (5.22) for $x = L$ to (5.23) and making use of the approximation we obtain the following pair of equations

$$\mathbf{A} - \mathbf{B} = -\frac{\mathbf{Z}_{mi0}}{\rho_L c}(\mathbf{A} + \mathbf{B})$$

$$\mathbf{A}_e^{-ikL} - \mathbf{B}_e^{ikL} = \frac{\mathbf{Z}_{miL}}{\rho_L c}\left(\mathbf{A}_e^{-ikL} + \mathbf{B}_e^{ikL}\right)$$

Solution of these preceding two equations by elimination of $\mathbf{A}$ and $\mathbf{B}$ results in the transcendental equation

$$\tan kL = i\frac{\dfrac{\mathbf{Z}_{mi0}}{\rho_L c} + \dfrac{\mathbf{Z}_{miL}}{\rho_L c}}{1 + \dfrac{\mathbf{Z}_{mi0}}{\rho_L c}\dfrac{\mathbf{Z}_{miL}}{\rho_L c}}$$

The characteristics of the vibration are determined from the complex impedances $\mathbf{Z}_{mi0}$ and $\mathbf{Z}_{miL}$. The solution of the preceding transcendental equation is rendered

more difficult by the presence of any resistive component in either $\mathbf{Z}_{mi0}$ or $\mathbf{Z}_{miL}$, which produces a complex argument of the tangent.

5.7 Transverse Vibrations of a Bar

A bar not only vibrates longitudinally; it can also vibrate transversely, which is usually the case because the strains engendered by longitudinal motion give a virtually automatic rise to transverse strains as the result of the Poisson effect. A hammer blow aimed along the axis of a long, thin bar supported at its center will usually result in principally transverse vibrations rather than the expected longitudinal effects, because it is not easy in the real world to avoid a slight eccentricity in applying the blow.

In our derivation of the transverse wave equation, consider a straight bar of length L with a uniform bilaterally symmetric cross-section $\hat{A}$. In Figure 5.6, a segment dx of the bar is shown bent (the bending is exaggerated to better illustrate the effect). The x-coordinate lies along the axis of the bar and the y-coordinate measures the transverse displacements of the bar from its unperturbed configuration. The bar behaves as a beam, i.e., the upper part of the cross section stretches under tension and lower part becomes compressed. A *neutral axis* NN', whose length remains unchanged, comprises the line of demarcation between compression and tension in the bar. If the cross section of the bar is symmetrical about a horizontal plane, the central axis of the bar will coincide with the neutral axis. The bending of the bar is gauged by the radius of curvature R of the neutral axis. Consider the length increment $\delta x = (\partial\xi/\partial x)dx$ due to the bending of a filament in the bar located at a distance r from the neutral axis. The longitudinal force df is found from

$$df = Ed\hat{A}\frac{\delta x}{dx} = Ed\hat{A}\frac{\partial\xi}{\partial x} \tag{5.24}$$

where $d\hat{A}$ is the cross-sectional area of the filament. Above the neutral axis NN' the value of δx is positive, so the force df becomes negative and thus is a tension. For filaments positioned below the neutral axis, δx is negative, resulting in a positive compressive force. From geometry we note that

$$\frac{dx + \delta x}{R + r} = \frac{dx}{R}$$

and this leads to $\delta x/dx = r/R$. Equation (5.24) now becomes

$$df = -\frac{E}{R} r\, d\hat{A}$$

The negative forces above the neutral axis cancel out the positive forces below the neutral axis, and hence the total longitudinal force $f = Idf$ equals zero. But a bending moment M occurs in the bar, i.e.,

$$M = \int r\, df = -\frac{E}{R}\int r^2\, d\hat{A}$$

From classical mechanics we recognize the radius of gyration κ of the cross-sectional area $\hat{A}$, as defined by

$$\kappa^2 = \frac{\int r^2 d\hat{A}}{\hat{A}}$$

We now obtain

$$M = \frac{\kappa^2 E \hat{A}}{R} \tag{5.25}$$

(For a bar with a rectangular cross section, $\kappa = t/\sqrt{12}$ where t denotes the thickness of the bar; for a circular rod of radius a, the radius of gyration is given by $\kappa = a/2$.)

The radius of curvature R varies along the neutral axis, but the mathematics can be simplified by assuming the displacements y of the bar to be quite small, $\partial y/\partial x \ll 1$, which permits the use of the approximation

$$R = \frac{[1 + (\partial y/\partial x)^2]^{3/2}}{\partial^2 y/\partial x^2} \approx \frac{1}{\partial^2 y/\partial x^2}$$

Equation (5.25) now modifies to

$$M = -\kappa^2 E \hat{A} \frac{\partial^2 y}{\partial x^2} \tag{5.26}$$

The curvature shown in Figure 5.5 has a negative $\partial^2 y/\partial x^2$, and the bending moment is consequently positive. In order to get this type of curvature the torque must be applied to the left end of the segment in a counterclockwise (or positive angular) direction and the torque at the right end of the segment must be clockwise (or in the negative angular direction).

Shear forces as well as bending moments arise when a bar becomes distorted. In Figure 5.7, a shear force $F_y(x)$ acts upward (in the positive sense) on the left end of the element dx. An opposing shear force $-F_y(x + \Delta x)$ acts downward at

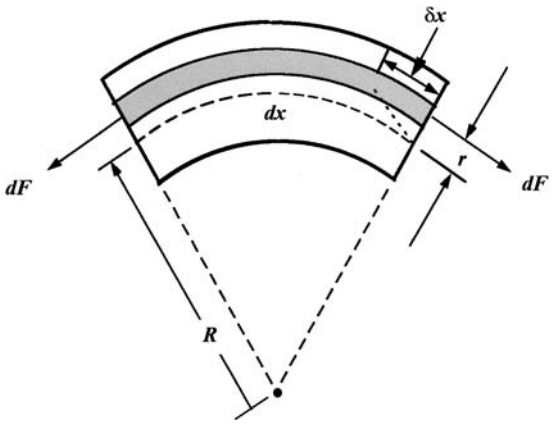


FIGURE 5.6. An element of a bar showing bending stresses and strains.

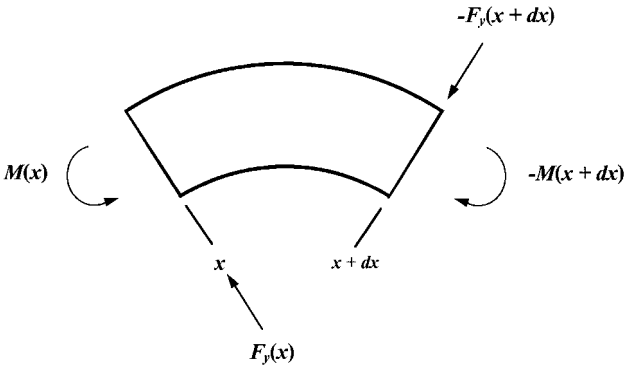


FIGURE 5.7. An element of the bar showing shear forces and bending moments.

the right end of the element. To sustain static equilibrium in the bent bar, the shear forces and torsions acting on the element must counterbalance each other so that there is no net turning momentum. As shown in Figure 5.7, taking the left end of the element as the reference pivot point we obtain

$$M(x) - M(x + dx) = F_y(x + dx) \tag{5.27}$$

The terms $M(x + \Delta x)$ and $F_y(x + \Delta x)$ are now expanded in a Taylor's series about point x, with the result that Equation (5.27) becomes

$$F_y = -\frac{\partial M}{\partial x} = \kappa^2 E \hat{A} \frac{\partial^3 y}{\partial x^3} \tag{5.28}$$

In Equation (5.28) the second-order and higher terms in dx have been discarded.

In undergoing transverse vibrations the bar is in dynamic rather than static equilibrium. This requires that the right-hand side of Equation (5.27) must equal the rate of increase of the angular momentum of the segment. But as long as the displacement and the slope of the bar remain small, the variations in angular momentum can be disregarded, and Equation (5.28) should serve as a good approximation of the correlation between the displacement y and the acting force F_y. The net upward force dF_y in element dx is given by

$$dF_y = F_y(x) - F_y(x + \Delta x) = -\frac{\partial F_y}{\partial x} = -\kappa^2 \hat{E} A \frac{\partial^4 y}{\partial x^4} dx$$

The element undergoes an upward acceleration under the impetus of the force, and the equation of motion for the mass of the element, $\rho \hat{A}\, dx$, may now be written as:

$$\rho \hat{A}\, dx \frac{\partial^2 y}{\partial t^2} = -\kappa^2 E \hat{A} \frac{\partial^4 y}{\partial x^4} dx$$

Setting $c = (E/\rho)^{1/2}$, as in the case of longitudinal waves, the last equation changes to

$$\frac{\partial^2 y}{\partial t^2} = -\kappa^2 c^2 \frac{\partial^4 y}{\partial x^4} \tag{5.29}$$

Equation (5.29) for the transverse wave differs from Equation (5.7) for the longitudinal wave principally in the presence of the fourth partial derivative with respect to x. Solutions in the functional form of $f(ct - x)$ do *not* apply to transverse waves, as can be readily proved by direct substitution into Equation (5.29). This means that transverse waves do not travel in the x-direction with constant speed c and unchanging shape.

Equation (5.28) can be solved by separation of variables by setting the complex transverse displacement $\boldsymbol{y}$ as

$$\boldsymbol{y} = \boldsymbol{\Psi}(x)e^{i\omega t} \tag{5.30}$$

and inserting into Equation (5.29). This yields a new total differential equation in which $\boldsymbol{\Psi}$ exists as a function of x only:

$$\frac{d^4\boldsymbol{\Psi}}{dx^4} = \frac{\omega^2}{\kappa^2 c^2}\boldsymbol{\Psi}$$

Setting

$$v = \sqrt{\kappa\omega c} \tag{5.31}$$

the fourth-order differential equation becomes

$$\frac{d^4\boldsymbol{\Psi}}{dx^4} = \frac{\omega^4}{v^4}\boldsymbol{\Psi} \tag{5.32}$$

The function $\boldsymbol{\Psi}$ may be assumed as an exponential of the form $\boldsymbol{\Psi}(x) = \mathbf{A}e^{\gamma\ x}$ and substituted into Equation (5.32). The result is

$$\gamma^4 = (\omega/v)^4$$

Four values of γ occur: $\pm(\omega/v)$ and $\pm(i\omega/v)$. The complete solution to Equation (5.32) consists of the sum of the four solutions:

$$\boldsymbol{\Psi} = \mathbf{A}e^{\omega x/v} + \mathbf{B}e^{-\omega x/v} + \mathbf{C}e^{i(\omega x/v)} + \mathbf{D}e^{-i(\omega x/v)}$$

wherein $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ constitute complex amplitude constants. From Equation (5.30) the solution for the displacements y can be written as

$$\mathbf{y} = e^{i\omega t}(\mathbf{A}e^{\omega x/v} + \mathbf{B}e^{-\omega x/v} + \mathbf{C}e^{i(\omega x/v)} + \mathbf{D}e^{-i(\omega x/v)}) \tag{5.33}$$

It should be noted that none of the terms in Equation (5.33) contains a wave moving with a velocity c. The third term inside the parenthesis of Equation (5.33) represents a wave disturbance moving to the left and the fourth term represents a wave moving in the positive x-direction. The *phase speed* v is itself a function of the frequency, as attested by Equation (5.31), so waves of differing frequencies will travel with different phase speeds. Higher-frequency waves will outpace the lower-frequency waves, and accordingly a complex wave containing a number of different frequencies will alter its shape along the x-axis. Each frequency component of a complex wave travels at its own speed v, which gives rise to a situation analogous to the transmission of light through glass, in which different component frequencies of

the light beam travel with different speeds thereby causing dispersion. A vibrating bar thus acts as a *dispersive* medium for transverse waves.

The real part of Equation (5.33) constitutes the actual solution of Equation (5.29). We make use of the following hyperbolic and trigonometric identities:

$$\begin{aligned} \sin y &= (e^{iy} - e^{-iy})/2, & \cos y &= (e^{iy} + e^{-iy})/2 \\ \sin(iy) &= i \sinh y, & \sinh(iy) &= i \sin y \\ \cos(iy) &= \cosh y, & \cosh(iy) &= \cos y \end{aligned}$$

to recast Equation (5.33) as

$$y = \cos(\omega t + \phi) \left(A \cosh \frac{\omega x}{v} + B \sinh \frac{\omega x}{v} + C \cos \frac{\omega x}{v} + D \sin \frac{\omega x}{v} \right) \tag{5.34}$$

Here A, B, C, D are real constants that occur from the rearrangement of the original complex constants **A**, **B**, **C**, **D**. The intricate relationships between the real set of constants and the set of complex constants are not really of much concern to us, because it is the application of the initial and boundary conditions that provides the evaluation of these constants. However, there are twice as many arbitrary constants in the transverse equation (5.34) as in the longitudinal wave equation (5.7), due to the fact that the former is of the fourth-differential order rather than the second-differential order. Therefore, twice as many boundary conditions are required, and this can be satisfied by specifying *pairs* of boundary conditions at the ends of the bars. The nature of the supports establishes the boundary conditions that generally fall into the categories of free and clamped ends.

5.8 Boundary Conditions for Transverse Vibrations

1. If the bar is rigidly clamped at one end, the both the displacement and the slope must be zero at that end at all times, and the boundary conditions are expressed as:

$$y = 0, \qquad \partial y / \partial x = 0 \tag{5.35}$$

2. On the other hand, neither an externally applied moment nor a shear force may exist at a free end of a vibrating bar. But the displacement and the slope of the bar at a free end are not constrained, excepting for the mathematical stipulation they remain small. From Equations (5.26) and (5.28) the boundary conditions become

$$\frac{\partial^2 y}{\partial x^2} = 0, \qquad \frac{\partial^3 y}{\partial x^3} = 0 \tag{5.36}$$

Case 1: Bar Clamped at One End

Consider a bar of length L that is rigidly clamped at $x = 0$ but is free at $x = L$. At $x = 0$, the two conditions of Equation (5.35) apply, so $A = -C$ and $B = -D$. The general solution (5.35) reduces to

$$y = \cos(\omega t + \phi) \left[A \left(\cosh \frac{\omega x}{v} - \cos \frac{\omega x}{v} \right) + B \left(\sinh \frac{\omega x}{v} - \sin \frac{\omega x}{v} \right) \right]$$

Applying free-end condition Equation (5.36) at $x = L$ yields the following two sets of equations:

$$A\left(\cosh\frac{\omega L}{v} + \cos\frac{\omega L}{v}\right) = -B\left(\sinh\frac{\omega L}{v} + \sin\frac{\omega L}{v}\right)$$

$$A\left(\sinh\frac{\omega L}{v} - \sin\frac{\omega L}{v}\right) = -B\left(\cosh\frac{\omega L}{v} + \cos\frac{\omega L}{v}\right)$$

Both of the preceding two equations cannot hold true for all frequencies. In order to determine the permissible frequencies, one equation is divided into the other, thus canceling out the constants A and B. Ridding the resulting equation of fractional expressions by cross-multiplication and using the identities $\cos^2\theta + \sin^2\theta = 1$ and $\cosh^2\theta + 1 = \sinh^2\theta$, we obtain

$$\cosh\frac{\omega L}{v}\cos\frac{\omega L}{v} = -1$$

We can alter the last equation by using the identities

$$\tan\frac{\theta}{2} = \sqrt{\frac{1-\cos\theta}{1+\cos\theta}}, \qquad \tanh\frac{\theta}{2} = \sqrt{\frac{\cosh\theta - 1}{\cosh\theta + 1}}$$

and we now obtain

$$\cot\frac{\omega L}{2v} = \pm\tanh\frac{\omega L}{2v} \tag{5.37}$$

The frequencies which correspond to the allowable modes of vibration can be found through the use of a microcomputer program which determines the intersections of the curves of $\cot \omega L/2v$ and $\pm\tanh \omega L/2v$, as shown in Figure 5.8. The frequencies of the permissible modes are given by

$$\frac{\omega L}{2v} = \zeta\frac{\pi}{4} \tag{5.38}$$

where $\zeta = 1.194, 2.988, 5, 7, \ldots$ with ζ approaching whole numbers for the higher allowed frequencies. Inserting $v = (\kappa\omega c)^{1/2}$ into Equation (5.38), squaring both sides, and solving for frequencies f, we obtain

$$f = \zeta\frac{\pi\kappa c}{8L^2}$$

The constraint imposed by the boundary conditions leads to a set of discrete allowable frequencies, but the overtone frequencies are not harmonics of the fundamental. When a metal bar is struck in such a manner that the amplitudes of the vibration of some of the overtones are fairly strong, the sound produced has a metallic cast. But these overtones rapidly die out, and the initial sound soon evolves into a mellower pure tone whose frequency is the fundamental. This is a characteristic of the behavior of a tuning fork that emits a short metallic sound upon being struck before emitting a pure tone.

The distribution of the nodal points along the transversely vibrating bar is quite complex, with three distinct types of nodal points being identified mathematically.

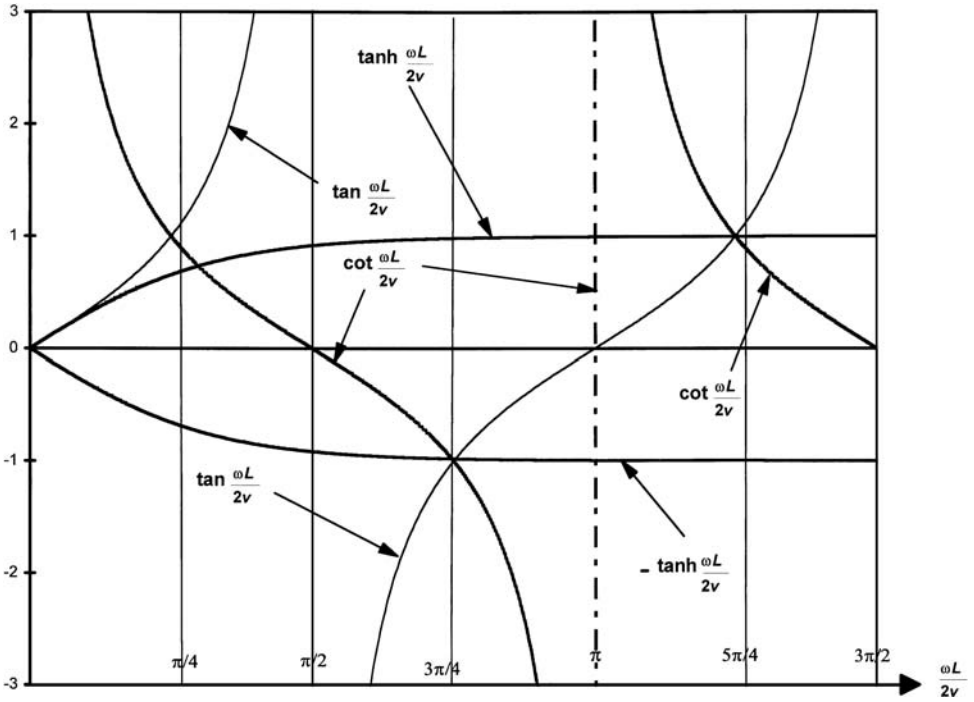


FIGURE 5.8. Trigonometric functions, used in Equations (5.37) and (5.38), plotted as functions of $(\omega L/2v)$.

The clamping point of the bar constitutes one type, with conditions $y = 0$ and $\partial y/\partial x = 0$ at all times. Another group of points called *true nodes* is characterized by $y = 0$ and $\partial y/\partial x \approx 0$, and they are found near points of inflections on the bar. The spacing between these true nodes is very nearly (but not quite) to $\lambda/2$. The third type of nodal point occurs at the node very near the free end of the bar, where $y = 0$, but the corresponding point of inflection where $\partial^2 y/\partial x^2 \approx 0$ does not coincide with that point but it is moved out to the free end. The vibrational amplitudes do not equal each other at the various antinodes but the greatest vibrational amplitude is that of the free end.

Case 2: Free–Free Bar

In the case of a bar that is free to move at both ends, the boundary conditions at $x = 0$ are satisfied by $A = C$ and $B = D$ as the result of applying Equation (5.36). The same set of boundary conditions applied to the other end $x = L$ yields

$$\tan \frac{\omega L}{2v} = +\tanh \frac{\omega L}{2v}$$

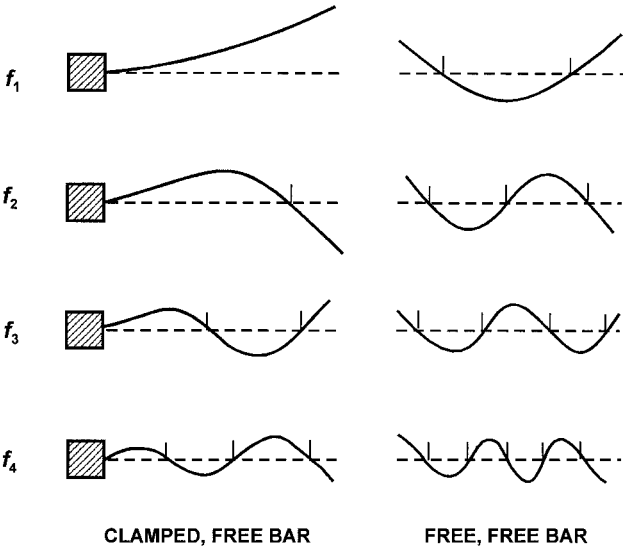


FIGURE 5.9. The first four transverse modes of a vibrating bar.

which in turn gives a discrete set of allowable frequencies of the transverse vibration. The frequencies are given by

$$f = \xi \frac{\pi \kappa c}{8L^2}$$

where $\xi = 3.0112^2, 5^2, 7^2, 9^2, \ldots$. The overtones are *not* harmonics of the fundamental.

Figure 5.9 illustrates the transverse modes of a clamped-free bar and a free–free bar. In the free–free bar the modes correspond to the fundamental frequency f_1 and all additional odd-numbered frequencies. The odd-numbered frequencies f_3, f_5, and so on, are symmetric about the center of the bar. The slope $\partial y/\partial x$ is always zero at the center, which is a true antinode. But the even-numbered frequencies f_2, f_4, $f_6, \ldots$ yield asymmetric modes of vibrations with respect to the center. In all modes the nodal points are found to be distributed symmetrically about the center. A bar may therefore be supported on a knife edge (or held by knife-edged clamps) at any nodal point without affecting the mode of vibration having a node at that point. A knife edge or a knife-edged set of clamps disallows displacement but not a change in the slope that occurs at the node.

The xylophone consists of metal bars that are supported at the nodal locations of the fundamental. But the nodes of the associated overtones are unlikely to be located at the same points, and the overtones quickly die out, leaving the pure tone of the fundamental. The concept of a free–free bar applies to a tuning fork which is essentially a U-shaped bar attached to a stem (Figure 5.10). The geometry of the fork and the mass-loading effect of the stem cause the nodes of the fundamental to be spaced closely near the stem. When the tuning fork is struck, overtones damp

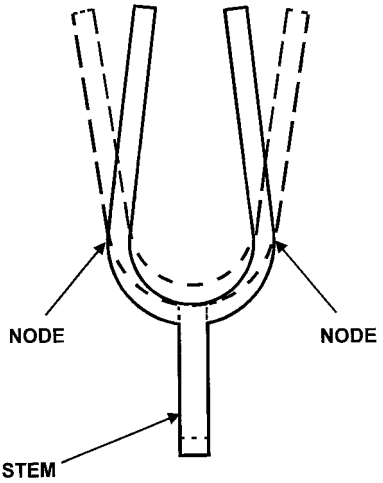


FIGURE 5.10. The tuning fork, essentially a U-shaped bar attached to a stem.

out in a very short time, leaving only the pure sinusoidal fundamental, much in the same manner as a xylophone. Because the stem shares the antinodal motion of the center of a free–free bar, the radiation efficiency of a tuning fork becomes greatly increased by touching the stem to a surface of a large area such as a counter top.

References

Fletcher, Neville H. and Rossing, Thomas D. 1991. *The Physics of Musical Instruments*. New York: Springer-Verlag, pp. 53–60.

Kinsler, Lawrence E., Frey, Austin R., Coppens, Alan B., and Sanders, James V. 1982. *Fundamentals of Acoustics*, 3rd ed. New York: John Wiley & Sons, Chapter 3.

Morse, Philip M. and Ingard., K. Uno. 1968. *Theoretical Acoustics*. New York: McGraw-Hill, Chapter 5.

Reynolds, Douglas R. 1981. *Engineering Principles of Acoustics, Noise and Vibration Control*. Boston, MA: Allyn and Bacon, pp. 224–234.

Wood, Alexander. 1966. *Acoustics*. New York: Dover Publications, pp. 384–386.

Problems for Chapter 5

1. Show that a bar of length L, that is rigidly fixed at $x = 0$ and totally free at $x = L$ will have only odd integral harmonic overtones.
2. Determine the fundamental frequency of the bar in Problem 1 if $L = 0.60$ m and the bar is made of steel. If a static force F is applied to the free end, so

that the bar displaces a distance δ, and then is suddenly released, demonstrate that the amplitudes of the subsequent longitudinal vibrations are given by

$$A_n = 8\frac{\delta}{n^2\pi^2}\ \sin\ \frac{n\pi}{2}$$

Find these amplitudes for this steel bar with a cross-sectional area of $5.0(10)^{-5}$ m^2, under the effect of a force of 4500 N.

3. A steel bar is free to move at $x = 0$. It has 0.0002 m^2 cross-sectional area and 0.35 m length. A 0.20 kg load is placed at $x = 0.35$ m.
   (a) Determine the fundamental frequency of the longitudinal vibration of the mass-loaded bar.
   (b) Establish the position at which the bar may be clamped so as to minimize interference with the fundamental mode.
   (c) Find the ratio of the displacement amplitude of the free end to that of the mass-loaded end for the first overtone of the bar.
4. A steel bar having a mass of 0.05 kg and 0.25 m length is loaded at one end with 0.028 kg and 0.056 kg at the other end. Find the fundamental frequency of the system's longitudinal vibration and determine the location of the node in the bar. Also compute the ratio of the displacement amplitudes at the two ends of the bar.
5. Redo Problem 4 for an aluminum bar of the same length but with a mass of 0.03 kg, subjected to the same end loadings.
6. Consider a thin bar of length L and mass M that is rigidly fixed at one end and free at the other. What mass m must be affixed to the free end in order to lower the fundamental frequency of longitudinal vibration by 30% from the fixed-free value?
7. A fixed-free bar of length L and mass m has a mechanical reactance, or stiffness, equal to $-is/\omega$ in the fixture. Develop an expression for the fundamental frequency of the longitudinal vibration.
8. A longitudinal force $F\cos\omega t$ drives a long thin bar at $x = 0$. The bar is free to move at $x = L$.
   (a) Obtain the equation for the amplitude of the standing waves occurring in the bar.
   (b) Obtain the expression giving the input mechanical impedance of the bar of length L.
   (c) Derive the expression for the input mechanical impedance of the same bar having an infinite length.
   (d) For the case of part (a) plot the amplitude of the driven end of the bar as a function of frequency over the range 100 Hz–3 kHz, if the material of the bar is aluminum, its length is 1.5 m, its cross-sectional area is $1.5\ (10)^{-4}$ m^2, and the amplitude of the driving force is 12 N.
9. Demonstrate that the dimensional units of $v = \sqrt{\kappa\omega c}$ are those of speed. At what frequency will the phase speed of transverse vibrations of a steel rod

of 0.005 m diameter equal the phase speed of longitudinal vibrations in the rod?

10. Given an aluminum rod of 0.010 radius and length 0.4 m, what will be the fundamental frequency of free–free vibrations? Predict the displacement amplitude of the free ends if the displacement amplitude of the rod at its center is 2.5 cm.

6
Membrane and Plates

6.1 Introduction

In this chapter we are now applying two-dimensional wave equations to membranes and plates. In this group of physical applications, three dimensions are really involved in the theory that applies to two-dimensional surfaces such as drumheads and diaphragms of microphones or loudspeakers. Two spatial coordinates are required to locate a point on a vibrating surface, but the displacement generally occurs along the third spatial coordinate. However, the expansion of the general one-dimensional solution to the wave equation for a string or a bar can easily be extended to two dimensions. Again, boundary conditions determine the discreteness of the vibrational frequencies, but the peripheral geometry of a membrane constitutes a factor additional to the effects expected for different types of support. For certain membrane or plate geometries, the choice of appropriate reference coordinate systems to match the contour of the subject surface can greatly facilitate the solution of the wave equation.

In the following sections the wave equation for a membrane under tension is derived, and solutions are developed for various geometries and supports. The classic case of a kettledrum is included as a practical example that includes the effect of damping, and forced vibrations are also taken into additional consideration.

The number of easily solvable two-dimensional problems is limited by the narrow choice of coordinate systems, and more complex cases can be treated by computerization, most notably through the application of finite-element methods.

6.2 Derivation of the Wave Equation for a Stretched Membrane

In order to develop a viably simple equation of motion for a vibrating membrane stretched under tension, the assumption is made that the extremely thin membrane is uniform and the vibrational amplitudes remain small. The membrane is also deemed to be perfectly elastic, without any damping. Designate T as the tension

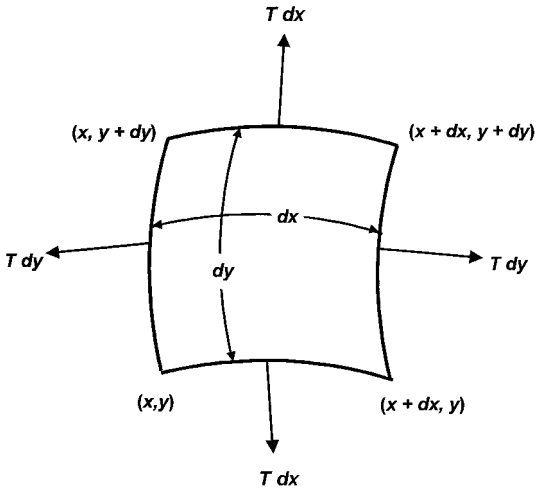


FIGURE 6.1. An element of the vibrating element.

of the membrane applied as a uniform force per unit length (e.g., in N/m) and ρ_S the surface density of the membrane as mass per unit area (kg/m^2). Thus, in Figure 6.1, an essentially two-dimensional membrane is stretched outward toward opposing sides of an element (each side of length dL) under the influence of a tension force $T\,dL$. A cartesian coordinate system is adopted, with the expanse of the quiescent membrane lying in the $x-y$ plane and the transverse displacement of a point at (x, y) occurring in the z-direction as $z = z(x, y, t)$. In Figure 6.1 an element of area $dS = dx\,dy$ undergoes the effect of transverse forces acting in the x- and y-directions along the peripheral lengths of the element. The net force in the x-direction is given by

$$T\left[\left(\frac{\partial z}{\partial x}\right)_{x+dx} - \left(\frac{\partial z}{\partial x}\right)_{x}\right] dy = T\frac{\partial^2 z}{\partial x^2}\,dx\,dy$$

and similarly, the net force in the y-direction is $T(\partial^2 z/\partial y^2)\,dx\,dy$. These two terms add up to contribute to the acceleration $\partial^2 z/\partial t^2$ of the element's mass $\rho_S\,dx\,dy$, in accordance with the Newton's law of motion:

$$T\left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2}\right) dy\,dy = \rho_S\,dx\,dy\frac{\partial^2 z}{\partial t^2}$$

and setting

$$c = \sqrt{\frac{T}{\rho_S}} \tag{6.1}$$

we obtain the classic two-dimensional wave equation

$$\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = \frac{1}{c^2}\frac{\partial^2 z}{\partial t^2} \tag{6.2}$$

We can recast the wave equation (6.2) in the more general Laplacian format:

$$\nabla^2 z = \frac{1}{c^2}\frac{\partial^2 z}{\partial t^2} \tag{6.3}$$

Equation (6.2) is suitable for treatment of rectangular membranes, but Equation (6.3) should be expressed in terms of polar coordinates to facilitate mathematical treatment of circular membranes:

$$\frac{\partial^2 z}{\partial r^2} + \frac{1}{r}\frac{\partial z}{\partial r} + \frac{1}{r^2}\frac{\partial^2 z}{\partial \theta^2} = \frac{1}{r^2}\frac{\partial^2 z}{\partial t^2} \tag{6.4}$$

For normal vibrational modes it is the standard mathematical procedure to assume that the solution to Equation (6.3) consists of a spatially dependent function $\mathbf{\Psi}$ and a strictly time-dependent function $e^{i\omega t}$ (we dispense with the other function $e^{-i\omega t}$ as being superfluous for the current physical applications):

$$z = \mathbf{\Psi} e^{i\omega t} \tag{6.5}$$

Inserting the above expression into Equation (6.3) and setting $k = \omega/c$ yields the time-independent *Helmholtz equation*:

$$\nabla^2 \mathbf{\Psi} + k^2 \mathbf{\Psi} = 0 \tag{6.6}$$

whose solutions upon insertion into Equation (6.5) yield normal modes of vibrations in a membrane of a given geometry and boundary conditions.

6.3 Rectangular Membrane with Fixed Edges

Consider a stretched rectangular membrane that is fixed at its four edges $x = 0$, $x = L_x$, $y = 0$, and $y = L_y$. The boundary conditions may be expressed as

$$z(0, y, t) = z(L_x, y, t) = z(x, 0, t) = z(x, L_y, t) = 0 \tag{6.7}$$

In the Cartesian format the solution $z(x, y, t) = \mathbf{\Psi}(x, y)e^{i\omega t}$ to Equation (6.2) must derive from the Helmholtz equation given below for Cartesian coordinates:

$$\frac{\partial^2 \mathbf{\Psi}}{\partial x^2} + \frac{\partial^2 \mathbf{\Psi}}{\partial y^2} + k^2 \mathbf{\Psi} = 0 \tag{6.8}$$

But $\mathbf{\Psi}(x, y)$ can be stated as the product of two singly dimensioned functions $X(x)$ and $Y(y)$ so that

$$\mathbf{\Psi}(x, y) = \mathbf{X}(x)\mathbf{Y}(y)$$

and then Equation (6.8) transforms to

$$\frac{1}{\mathbf{X}}\frac{\partial^2 \mathbf{X}}{\partial x^2} + \frac{1}{\mathbf{Y}}\frac{\partial^2 \mathbf{Y}}{\partial y^2} + k^2 = 0 \tag{6.9}$$

Because the three terms of Equation (6.9) cannot all sum to zero and the first and second terms are wholly independent of each other, we separate Equation (6.9) into two separate differential equations, one wholly dependent on x and the other on y:

$$\begin{aligned} \frac{1}{\mathbf{X}}\frac{\partial^2 \mathbf{X}}{\partial x^2} + k_x^2 &= 0 \\ \frac{1}{\mathbf{Y}}\frac{\partial^2 \mathbf{Y}}{\partial y^2} + k_y^2 &= 0 \end{aligned} \tag{6.10}$$

where k_x^2 and k_y^2 are constants related by

$$k_x^2 + k_y^2 = k^2$$

The solutions to the equation set (6.10) consists of sinusoids, with the result

$$\mathbf{z}(x, y, t) = \alpha \sin(k_x x + \phi_x) \sin(k_y y + \phi_y) e^{i\omega t} \tag{6.11}$$

Here α represents the maximum displacement of the membrane in the transverse direction, and ϕ_x and ϕ_y are determined by boundary conditions. With the first and the third of the boundary condition set (6.7) we find that $\phi_x = \phi_y = 0$ and the remaining conditions necessitate that $\sin k_x L_x = 0$ and $\sin k_y L_y = 0$. Therefore, the normal modes occur from

$$\mathbf{z}(x, y, t) = \alpha e^{i\omega t} \sin k_x x \sin k_y y \tag{6.12}$$

for which k_x and k_y turn out to be discrete values established by

$$\begin{aligned} k_x &= n\pi/L_x \qquad n = 1, 2, 3, \ldots \\ k_y &= m\pi/L_y \qquad m = 1, 2, 3, \ldots \end{aligned}$$

The frequencies of the allowed modes of vibrations are found from

$$f_{nm} = \frac{\omega_{nm}}{2\pi} = \frac{c}{2}\sqrt{\left(\frac{n}{L_x}\right)^2 + \left(\frac{m}{L_y}\right)^2} \tag{6.13}$$

Equation (6.13) constitutes a fairly simple extension of the allowable frequencies of a idealized free vibrating string to two-dimensional status.

The fundamental frequency is found by merely setting $n = m = 1$ in Equation (6.12). The overtones corresponding to $m = n > 1$ will be harmonics of the fundamental frequency, but those in which $m \neq n$ (with either m or $n > 1$) may not necessarily be so. A number of possible modes in a rectangular membrane are illustrated in Figure 6.2. The shaded areas vibrate π radians out of time phase with the unshaded areas. Each normal mode is designated by an ordered pair (n, m), and the nodal lines are those with zero displacement at all times. In theory, rigid supports could be placed along these lines without affecting the nodal pattern for the associated specific frequency.

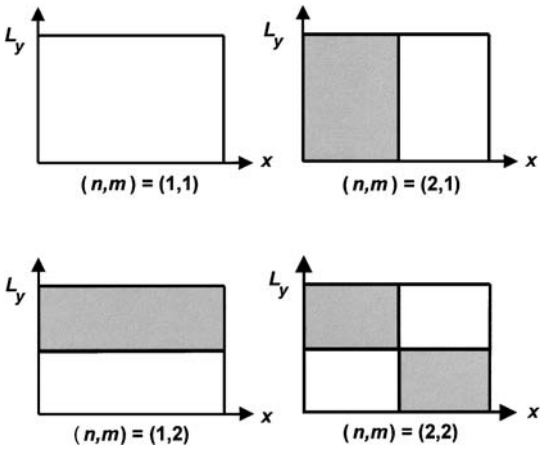


FIGURE 6.2. Four modes of a vibrating membrane. Lines located within the borders of the membranes constitute nodal loci where displacements are zero for these respective modes.

6.4 Freely Vibrating Circular Membrane with Fixed Rim

As mentioned in the foregoing it is preferable to adopt the polar coordinate version (6.4) of the wave equation (6.3) to treat the case of a circular membrane that is fixed at its rim. Accordingly, the zero displacement of the membrane's boundary at radius $r = a$ gives the boundary condition

$$z(a, \theta, t) = 0$$

The harmonic solution to Equation (6.4) can be represented as the product of three terms, each of which is functions of only one variable:

$$z(r, \theta, t) = \mathbf{R}(r)\Theta(\theta)e^{i\omega t} \tag{6.14}$$

The boundary condition stipulated at the rim of the circular membrane now becomes

$$\mathbf{R}(a) = 0$$

and insertion into Equation (6.4) yields the polar coordinate version of the Helmholtz equation:

$$\Theta\frac{d^2\mathbf{R}}{dr^2} + \frac{\Theta}{r}\frac{d\mathbf{R}}{dr} + \frac{\mathbf{R}}{r^2}\frac{d^2\Theta}{d\theta^2} + k^2\mathbf{R}\Theta = 0 \tag{6.15}$$

where $k = \omega/c$, as before. Equation (6.15) is then rearranged to effect the separation of two variables, so we obtain

$$\frac{r^2}{\mathbf{R}}\left(\frac{d^2\mathbf{R}}{dr^2} + \frac{1}{r}\frac{d\mathbf{R}}{dr}\right) + (kr)^2 = -\frac{1}{\Theta}\frac{d^2\Theta}{d\theta^2} \tag{6.16}$$

The left-hand side of Equation (6.16) is solely a function of r while the right depends only on θ. In order for the equality of Equation (6.16) to prevail, both sides of the equation must be set equal to a constant m^2. Then we obtain from the right-hand side of Equation (6.16)

$$\frac{d^2\Theta}{d\theta^2} + m^2\Theta = 0$$

which in turn yields the harmonic solution

$$\Theta(\theta) = \cos(m\theta + \varepsilon)$$

Here ε is the phase angle. The azimuthal coordinate is of periodic nature, repeating itself every 2π radians. In order that the displacement z be single-valued function of position, $z(r, \theta, t)$ must then equal itself every 2π radians, that is, $z(r, \theta, t) = z(r, \theta + 2\pi, t)$, with the result that the constant m is constrained to integral values $m = 1, 2, 3, \ldots$. The left-hand side of Equation (6.16) therefore becomes the Bessel's differential equation:

$$\frac{d^2\mathbf{R}}{dr^2} + \frac{1}{r}\frac{d\mathbf{R}}{dr} + \left(k^2 - \frac{m^2}{r^2}\right)\mathbf{R} = 0 \tag{6.17}$$

The solution of Equation (6.17) is

$$\mathbf{R}(r) = \mathbf{A}J_m(kr) + \mathbf{B}Y_m(kr) \tag{6.18}$$

where $J_m(kr)$ and $Y_m(kr)$ are, respectively, the transcendental *Bessel function*s of the *first* and the *second kind*, each of the order m. Bessel functions are oscillating functions with diminishing amplitudes for increasing kr. $Y_m(kr)$ approaches infinity as $kr \to 0$. The numerical values of Bessel functions and their properties are given in standard tables and advanced mathematical computer programs. An abbreviated set of Bessel formulas and tables art given in Appendix B. Because the circular membrane includes the origin $r = 0$ and the displacement of the membrane must remain finite at that point, it is necessary to set[1] $\mathbf{B} = 0$, reducing Equation (6.18) to

$$\mathbf{R}(r) = \mathbf{A}J_m(k)$$

Applying the boundary condition $\mathbf{R}(a) = 0$ requires that $J_m(ka) = 0$. Let us designate by q_{mn} those values of the argument ka at which the mth order Bessel function J_m equals zero. From $J_m(q_{mn}) = 0$ we can find those discrete values of k which are given by

$$k_{mn} = \frac{J_{mn}(q_{mn})}{a} = \frac{J_{mn}}{a}$$

[1] On the other hand, if an annular membrane stretched over region $a < r < b$ (thus excluding the origin $r = 0$) is considered, both Bessel functions must be retained in Equation (6.18) to provide the two arbitrary constants needed for the boundary conditions at the inner and outer borders.

A number of values of q_{mn} that yield zeros of the Bessel functions are given in Appendix B. We can therefore write the normal modes of vibration as

$$z_{mn}(r, \theta, t) = A_{mn} J_n(k_{mn} r) \cos(m\theta + \varepsilon_{mn}) e^{i\omega_{mn} t} \tag{6.19}$$

with $k_{mn} a = q_{mn}$ and the natural frequencies found from

$$f_{mn} = \frac{1}{2\pi} \frac{q_{mn} c}{a} \tag{6.20}$$

The real part of Equation (6.19) describes the physical displacement in the normal mode (m, n) as follows:

$$z_{mn}(r, \theta, t) = A_{mn} J_m(k_{mn} r) \cos(m\theta + \varepsilon_{mn}) \cos(\omega_{mn} t + \phi_{mn})$$

where $A_{mn} = \mathbf{A}_{mn} = e^{i\phi mn}$. The arbitrary constant ε_{mn} is an azimuthal phase angle. For each normal mode, this constant of integration defines the directions along which the radial nodal lines of zero displacement occur, but the value of ε_{mn} depends on the value of the azimuthal angle at which the membrane is excited at $t = 0$. A number of the first few (and simpler) modes of vibration with $\varepsilon_{mn} = 0$ are shown in Figure 6.3. Each mode is designated by the ordered pair of integers (m, n). Integer m governs the number of *radial nodal lines*, whereas integer n determines the number of *azimuthal nodal circles*. Since mode $(0, 0)$ would obviously be trivial,

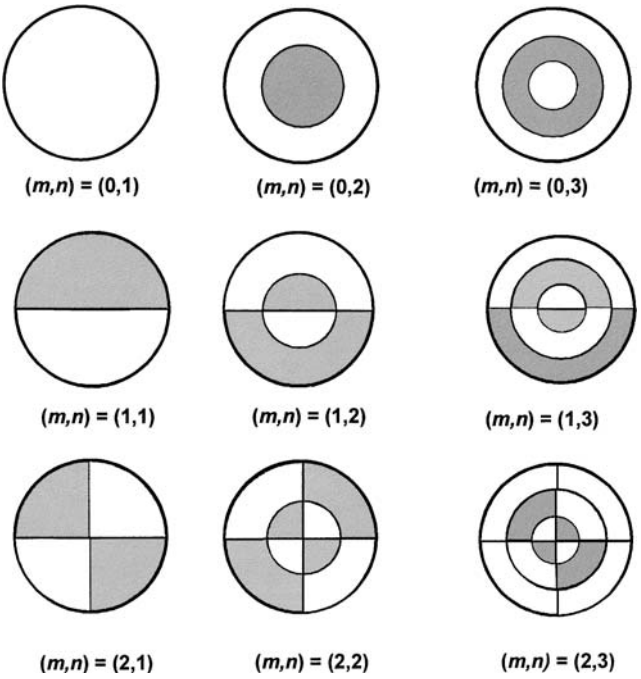


FIGURE 6.3. Vibration modes in a circular membrane fixed at its perimeter. A number of simpler modes are shown here.

TABLE 6.1. Relative Frequencies for Various Vibrational Modes.

$f_{01} = 1.0 f_{01}$	$f_{11} = 1.593 f_{01}$	$f_{21} = 2.135 f_{01}$
$f_{02} = 2.295 f_{01}$	$f_{12} = 2.917 f_{01}$	$f_{22} = 3.500 f_{01}$
$f_{03} = 3.598 f_{01}$	$f_{13} = 4.230 f_{01}$	$f_{23} = 4.832 f_{01}$

the case of $n = 1$ constitutes the least allowed value of n, and this corresponds to a mode of vibration where one azimuthal nodal circle occurs only at the rim of the membrane where $r = a$. Virtually the entire membrane vibrates in the z-direction in axisymmetric unison, with the maximum amplitude at $r = 0$ tapering off to zero at the boundary $r =$ a.

For each non-zero value of m there exists a chain of allowed radial vibration modes of increasing frequency, as illustrated in Figure 6.3. At $m = 0$, $J_0(k_{0n}a)$ equals zero, which sets the conditions for the allowed frequencies. For $m = 1$, $J_1(k_{1n}a) = 0$ provides the allowed frequencies; and for $m = 2$, $J_2(k_{2n}a) = 0$ supplies the corresponding frequencies for that radial mode. Table 6.1 lists the frequencies f_{mn} of the circular membrane relative to the fundamental frequency f_{01} obtained from Equation (6.20). From Table 6.1 it is observed that none of the overtones exists as a harmonic of the fundamental.

6.5 Case Study: Symmetric Vibrations of a Circular Membrane Fixed at its Perimeter

The case of symmetric vibrations of a circular membrane fixed at its rim has many applications, so this situation holds great interest for us. Vibrational symmetry implies a solution to the wave equation (6.4) that is independent of θ, and we limit the solutions for this case to

$$z_{0n} = A_{0n} J_0(k_{0n} r) e^{i\omega_{0n} t} \tag{6.21}$$

and because only $m = 0$ applies, we discard this subscript and retain only the index n. The fixedness of the circular perimeter forecasts the boundary condition $z = 0$ at $r = a$. Therefore, $J_0(ka) = 0$, with the zeroes occurring as $k_n a = 2.405, 5.520, 8.654, 11.792, \ldots$. The fundamental frequency is derived from

$$f_1 = \frac{\omega_1}{2\pi} = \frac{k_1 c}{2\pi} = \frac{2.405}{2\pi a}\sqrt{\frac{T}{\rho_s}} \tag{6.22}$$

with the (non-harmonic) overtones to the fundamental frequencies obtained as

$$f_2 = \left(\frac{5.520}{2.405}\right) f_1 = 2.295 f_1$$
$$f_3 = 3.58 f_1,$$
$$f_4 = 4.90 f_1, \ldots, \text{etc.}$$

For the real part of $\mathbf{z}_1 = \boldsymbol{A}_1 J_0(k_1 r) e^{i\omega t}$ the general expression for the membrane's displacement in the fundamental mode can be written as

$$z_1 = A_1 \cos(\omega_1 t + \phi_1) J_0\left(\frac{2.405r}{ra}\right) \tag{6.23}$$

where A_1 is the maximum absolute value of the complex amplitude $\mathbf{A}_1$ at the center $r = 0$ of the membrane. To completely describe the symmetric vibrations the complete solution must be expressed as

$$z = \sum A_n \cos(\omega_n t + \phi_n) J_0(k_n r) \tag{6.24}$$

For the symmetric modes of vibration above the fundamental, nodal circles will exist at the inner radii at which $J_0(k_n r)$ vanishes. For example, the first overtone $J_0(k_2 r) = 0$ when $k_2 r = 5.520\ r/a = 2.405$, or $r = 0.436a$. In Figure 6.3, this mode of vibration is shown as an annulus surrounding the central circle. When the central circle of the membrane moves up, the outer annulus moves down, and vice-versa. Because portions of the membrane are moving upward, at the same time the remaining areas move downward, the efficiency of the sound output of a drum is low for the overtone frequencies.

A measure of the sound output of each mode is the average displacement amplitude of the surface when it is vibrating in that mode. For the nth symmetric mode, we can apply Equation (6.24) to determine the average displacement amplitude $\langle \boldsymbol{\Psi}_n \rangle$ over surface S as follows:

$$\begin{aligned} \langle \boldsymbol{\Psi}_n \rangle &= \frac{1}{\pi a^2} \int_S A_n J_0(k_n r)\, dS \\ &= \frac{1}{\pi a^2} \int_0^a A_n J_0(k_n r) 2\pi r\, dr \\ &= \frac{2A_n}{k_n a} J_1(k_n a) \end{aligned} \tag{6.25}$$

For all nonsymmetric modes, we note that the angular dependence $\cos(m\theta + \varepsilon)$ ensures that the average displacement is zero. Thus, using the prior double-subscript notation, we can state that $\langle \boldsymbol{\Psi}_n \rangle = 0$ for all $m \neq 0$.

It is of interest to find the value of average displacement amplitude $\langle \boldsymbol{\Psi}_n \rangle$ for the fundamental mode and to compare it with $\langle \boldsymbol{\Psi}_n \rangle$ for the first overtone. From Equation (6.25),

$$\langle \boldsymbol{\Psi}_1 \rangle = \frac{2A_1}{k_1 a} J_1(k_1 a) = \frac{2A_1}{2.405} = 0.432 A_1$$

The motion of the membrane can be equated to the displacement of a rigid flat piston of radius a moving with an amplitude of $0.432A_1$. We can also apply Equation (6.25) to find that $\langle \boldsymbol{\Psi}_n \rangle = -0.123A_2$. The negative sign denotes that the average displacement amplitude is opposite in direction to the displacement at the center. The fundamental node of vibration is thus seen to be more than three times as effective for displacing air as for the first overtone.

In real applications such as loudspeakers, the amount of air displaced by the membranes, rather than the exact shape of the moving surface, determine the principle characteristics of generated sound waves. The radiating source can be depicted by an *equivalent simple piston* of area S_{eq}, and this piston moves through a displacement amplitude ζ_{eq} so as to sweep the volume displacement of the actual source. The volume displacement amplitude of the simple piston equivalent to the circular membrane vibrating in its fundamental mode is

$$S_{eq}\zeta_{eq} = 0.432\pi a^2 A_1$$

The nodal vibrations of actual membranes cannot be sustained with constant amplitudes because of damping forces occasioned by internal friction and external forces associated with the radiation of acoustic energy. The amplitude of each mode tends to decay exponentially with time as $e^{-\beta_n t}$, where β_n represents the damping constant for mode n. This damping constant generally increases with frequency, with the result that higher frequencies damp out more quickly than does the fundamental.

6.6 Application of Membrane Theory to the Kettledrum

In addition to the damping forces mentioned above, other forces may act on a membrane and affect its vibration. The kettledrum is an example of the case of a membrane that covers a closed space in which changes of pressures occur as the entrapped volume of air changes in pressure incurred by the vibration of the drumhead. A similar situation occurs with the air entrapped behind the diaphragm of a condenser microphone.

The kettledrum consists of a membrane stretched taut over the open end of a hemispherical shell. As this membrane (i.e., the drumhead) vibrates, the air contained inside the shell undergoes alternative compressions and rarefactions. With the radial velocity of the transverse waves being considerably less than the speed of sound in air, the pressure arising from the alternative compression and decompression of the entrapped air is fairly uniform across the entire drumhead and depends only on the average displacement $\langle z \rangle$. With the radius of the drumhead designated by a, the incremental or displaced volume of the enclosed air is given by $dV = \pi a^2 \langle z \rangle$. Let us denote by V_0 the equilibrium or quiescent volume of the air enclosed in the kettledrum. The corresponding unperturbed pressure is P_0. The vibration of the air enclosed in the kettledrum is essentially an *adiabatic* process, with the result that the instantaneous pressure P and volume V are related to the quiescent values by

$$PV^\gamma = P_0 V_0^\gamma = \text{constant} \tag{6.26}$$

Here γ is the ratio of c_p, the specific heat of the contained air at constant pressure, to c_v, the specific heat at constant volume. Differentiating Equation (6.26) yields

the pressure deviation dP:

$$dP = -\frac{\gamma P_0}{V_0} dV = -\frac{\gamma P_0}{V_0} \pi a^2 \langle z \rangle \tag{6.27}$$

This gives rise to an incremental force $dP\ dx\ dy$ over incremental area $dx\ dy$ of the membrane. Modifying Equation (6.3) to include this incremental force we obtain

$$\nabla^2 z - \frac{1}{c^2} \frac{\gamma P_0}{\rho_0 V_0} \pi a^2 \langle z \rangle = \frac{1}{c^2} \frac{\partial^2 z}{\partial t^2} \tag{6.28}$$

The term $\langle z \rangle$ is an integral function of all the permitted modes of vibration, which must also include the influences of their relative amplitudes and phases. We can greatly simplify the solution of Equation (6.28) by assuming only one mode of vibration and disregarding all of the other modes which constitute the general solution.

The average displacement is zero for all normal modes dependent upon θ; therefore, none of these modes are affected by the pressure fluctuation of the air inside the drum. We need only to consider the symmetric modes entailing the Bessel function J_0. The solution with only one frequency present is of the form depending only on the coordinate r

$$\mathbf{z} = \mathbf{\Psi} e^{i\omega t}$$

Inserting the above into Equation (6.28) yields

$$\frac{d^2 \mathbf{\Psi}}{dr^2} + \frac{1}{r} \frac{d\mathbf{\Psi}}{dr} + k^2 \mathbf{\Psi} = \frac{\gamma P_0}{T V_0} \int_0^a 2\pi r \mathbf{\Psi}\ dr \tag{6.29}$$

In order to establish the solution to the differential equation, we examine the salient features of Equation (6.29). If the right-hand integral term were not present, the solution would entail $J_0(kr)$. But the presence of this integral term involving the radius a suggests an additional term to the solution, one that is a function of a, namely $J_0(ka)$. Moreover, the assumed solution should meet the boundary condition that $\mathbf{\Psi} = 0$ at $r = a$, regardless of the value of k.

We now integrate the right-hand side of Equation (6.29) as follows

$$\begin{aligned} &\frac{2\pi \gamma P_0}{T V_0} \mathbf{A} \left[\frac{r J_1(kr)}{k} - \frac{r^2}{2} J_0(ka) \right]_0^a \\ &\quad = \frac{\gamma P_0}{T V_0} \pi a^2 \mathbf{A} \left[\frac{2 J_1(ka)}{ka} - J_0(ka) \right] = \frac{\gamma P_0}{T V_0} \pi a^2 \mathbf{A} J_2(ka) \end{aligned} \tag{6.30}$$

Insertion of Equation (6.30) into Equation (6.29) provides the condition for the viability of the assumed solution

$$-k^2 J_0(ka) = \frac{\gamma P_0}{T V_0} \pi a^2 J_2(ka)$$

TABLE 6.2. Allowed Frequencies of a Kettledrum.

B	0	1	2	5	10
k_1a	2.405	2.545	2.68	3.02	3.485
k_2a	5.520	5.54	5.55	5.59	5.87
k_3a	8.654	8.657	8.660	8.67	8.69

or

$$J_0(ka) = -\frac{BJ_2(ka)}{(ka)^2}$$

where

$$B = \frac{\gamma P_0}{TV_0}\pi a^2$$

The parameter B is a dimensionless constant that compares the relative magnitudes of the restoring forces arising from the compression effects of the air trapped inside the drum and the tension applied to the drumhead. The value of B is small if either the volume or the tension in the membrane is quite large compared with the compressive pressure acting over the area of the membrane. In the limit where B approaches zero, the allowed frequencies become those corresponding to the freely vibrating circular membrane which was described earlier in this chapter. The allowed values of ka are listed in Table 6.2 for selected values of B ranging from 0 (which corresponds to an unimpeded vibrating circular membrane) to 10 (indicative of low drum volume or light drumhead tension). The effect of the additional term in Equation (6.28), which is proportional to the displacement and therefore is indicative of membrane stiffness, is to elevate the allowed frequencies. The effect on the fundamental frequency is much more considerable than it is on the higher modes of vibration. This stems from the fact that the average displacement amplitude becomes smaller with increasingly higher modes of vibration with a consequently larger number of oppositely phased segments. It is also apparent that since pressure fluctuations inside the drum affect only the basic frequency modes z_{0n}, the tonal qualities of a kettledrum can be varied by parametric changes of the drum volume V_0 and the area πr^2 of the drumhead.

6.7 Forced Vibrations of a Membrane

Consider a circular membrane that is acted only on one side by a evenly distributed sinusoidal driving pressure $p = P\cos\omega t$. In complex notation the pressure is given by

$$\mathbf{p} = Pe^{i\omega t}$$

and the equation of motion (6.3) becomes modified as follows:

$$\frac{\partial^2 \mathrm{z}}{\partial t^2} = c^2\nabla^2 \mathrm{z} + \frac{P}{\rho_s}e^{i\omega t} \tag{6.31}$$

Assume a steady-state solution

$$\mathbf{z} = \boldsymbol{\Psi} e^{i\omega t} \tag{6.32}$$

which is then inserted into Equation (6.31), resulting in

$$\nabla^2 \boldsymbol{\Psi} + k^2 \boldsymbol{\Psi} = \frac{P}{\rho_s c^2} = -\frac{P}{T} \tag{6.33}$$

where $k = \omega/c$. In this situation of a driven membrane the angular frequency ω may have any value, and the wave number k is thus not limited to discrete sets of values which prevail in freely vibrating membranes.

The solution to Equation (6.33) consists of two parts, one being a general solution of the homogeneous equation $\Lambda^2 \boldsymbol{\Psi}_h + k^2 \boldsymbol{\Psi}_h = 0$ and the second being the particular solution $\boldsymbol{\Psi}_p = -P/(k^2 T)$. Then the complete solution can be written as

$$\boldsymbol{\Psi} = \mathbf{A} J_0(kr) - \frac{P}{k^2 T}$$

The immobility of the membrane at the rim $r = a$ provides the boundary condition $\boldsymbol{\Psi}(a) = 0$, and

$$\mathbf{A} = \frac{1}{J_0(ka)} \cdot \frac{P}{k^2 T}$$

The displacement of the membrane becomes

$$z(r, t) = \frac{P}{k^2 T} \left[\frac{J_0(kr)}{J_0(ka)} - 1 \right] e^{i\omega t} \tag{6.34}$$

with the corresponding amplitude of the displacement at any position in the membrane given by

$$\boldsymbol{\Psi}(r) = \frac{P}{T} \left[\frac{J_0(kr) - J_0(ka)}{k^2 J_0(ka)} \right] \tag{6.35}$$

From Equation (6.35) it is seen that the amplitude of the displacement is directly proportional to the driving force P and inversely proportional to the tension T. The vibrational amplitude at any location on the membrane depends on the transcendental terms enclosed by the square bracket in Equation (6.35). But if the driving frequency ω corresponds to any of the free-oscillation frequencies of Equation (6.22), the overtones, the Bessel function $J_0(ka)$ assumes zero values, presaging infinite amplitudes. But damping forces occur in real cases, and these may be represented in Equation (6.31) by a damping factor $-(R/\rho_s)(\partial z/\partial t)$ that limits the amplitudes to finite maximum values.

The average displacement $\langle z \rangle_s$ of the driven membrane is found by averaging over the surface area of the membrane:

$$\begin{aligned} \langle z \rangle_s &= \frac{2\pi e^{i\omega t} \int_0^a \frac{P}{k^2 T} \left[\frac{J_0(kr)}{J_0(ka)} - 1 \right] r \, dr}{\pi a^2} \\ &= \frac{P}{k^2 T} \frac{J_2(ka)}{J_0(ka)} e^{i\omega t} \end{aligned} \tag{6.36}$$

At low frequencies ka assumes a value less than unity and the following approximations for Bessel functions hold true:

$$J_0(ka) \approx 1 - 3(ka)^2$$
$$J_2(ka) = \frac{k^2a^2}{8}\left[1 - \frac{k^2a^2}{12}\right]$$

Introducing the above approximations into Equation (6.36) yields the following expression for the average displacement at low frequencies,

$$\langle z\rangle_s \approx \frac{Pa^2}{8T}\left(1 + \frac{k^2a^2}{6}\right) \tag{6.37}$$

If we apply the situation as represented by Equation (6.37) to the design of a condenser microphone, it is apparent that as long the driving frequency is sufficiently low, i.e., $ka \ll 1$, the output of the microphone will be virtually independent of the frequency. No resonances should occur in that frequency range. The first resonance occurs at $ka = 2.405$. Because

$$k = \frac{2\pi f}{c} = 2\pi f\left(\frac{\rho_S}{T}\right)^{-1/2}$$

and if we set the limiting frequency of the uniform microphone response to $ka < 1$, then

$$f < \frac{1}{2\pi a}\sqrt{\frac{T}{\rho_s}}$$

The upper frequency limit of the microphone can be elevated by either increasing the tension T or decreasing the radius a, all other factors being equal. But this also has the effect of lessening the amplitude of the average displacement $\langle z\rangle_s$ and, consequently, the voltage output of the microphone.

When a damping factor $-(R/\rho_s)(\partial z/\partial t)$ is included in Equation (6.31), the resulting solution does not change except that k is replaced by $\mathbf{k}$, a complex expression represented by

$$\mathbf{k}^2 = \frac{\omega^2}{c^2} - \frac{i\omega R}{T}$$

The presence of the imaginary component $-\omega R/T$ causes the average displacement to assume a finite value at resonance. Figure 6.4 displays the average displacement response $\langle\Psi\rangle_s$ of a freely vibrating dissipationless membrane, as computed through the use of Equation (6.36). The amplitude assumes a value of infinity at $ka = 2.405$. Another curve that includes the effect of damping is also plotted, and the corresponding amplitude assumes a finite value at $ka = 2.405$. Both of these curves indicate zero responses at $ka = 5.136$, for which $J_2(ka) = 0$. If the frequency is increased beyond the first resonance value to approximately 1.60 times the first resonant frequency, a circular nodal line will appear near the rim of the membrane. As the frequency is increased, the nodular line moves inward as a circle of decreasing radius. The displacement of the membrane's center is out of

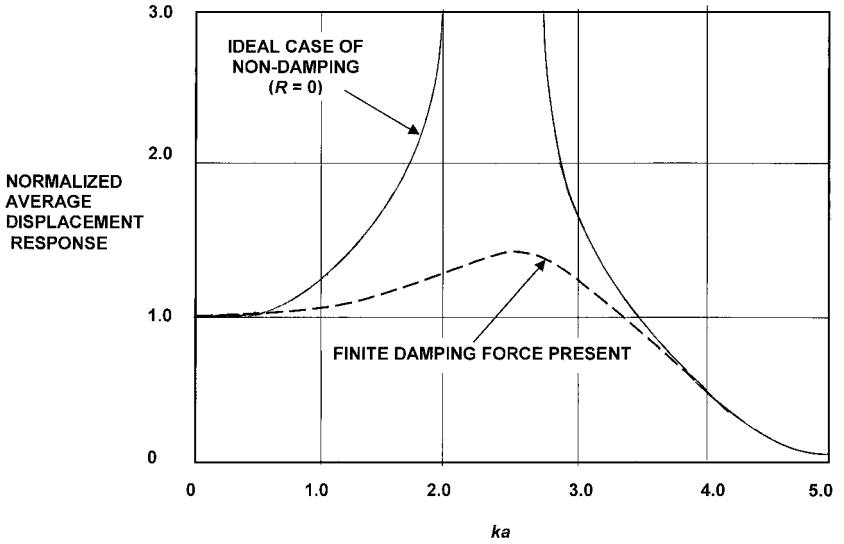


FIGURE 6.4. Plot showing the average (normalized) displacement response as a function of frequency. The effect of damping is also shown.

phase with the driving force, while that of the membrane's outer portion remains in phase. As the driving frequency increases and the nodal circle shrinks, the average displacements of the two zones tend to cancel each other out. The cancellation becomes complete at $ka = 5.136$, and no displacement occurs across the entire surface of the membrane.

6.8 Vibrating Thin Plates

The principal difference between the vibration of a membrane and a thin plate is the restoring force in the former is due entirely to the tension acting on the membrane and in the latter there is no tension applied, and the restoring force is attributed entirely to the inherent stiffness of the plate.

To keep matters simple, we consider only symmetrical vibrations of a uniform circular diaphragm. The appropriate equation, essentially equivalent to Equation (6.36) but modified to include the effect of stiffness, is

$$\frac{\partial^2 z}{\partial t^2} = \frac{\kappa^2 E}{\rho(1-\mu^2)} \nabla^2(\nabla^2 z) \tag{6.38}$$

where ρ is the density of the material, μ is the Poisson's ratio, E is Young's modulus, and κ is the surface radius of gyration. For a circular plate of uniform thickness b, the radius of gyration is given by

$$\kappa = \frac{b}{\sqrt{12}}$$

The elastic resistance to flexing provides the restoring force that acts on the circular plate. While there may be some temptation to consider the coefficient $-\kappa^2 E/[\rho(1-\mu^2)]^2$ on the right-hand side of Equation (6.38) as being analogous to $-\kappa^2 E/\rho$ in Equation (5.28) for the transverse vibration of a bar, this is not strictly true because a sheet will curl up sideways as it is bend downward along its length. This is the *Poisson's effect* in which the curling occurs from the lateral expansion as the longitudinal compression ensues from the bending of the plate. An increase in the effective stiffness is thereby produced. The Poisson's ratio μ given in Equation (6.38) is the negative ratio of the lateral strain $M\xi/My$ to $M\zeta/Mx$, i.e.,

$$\mu = -\frac{\partial \xi/\partial y}{\partial \varsigma/\partial x}$$

In order to keep the Poisson's ratio a positive number, it is necessary to introduce the minus sign to counteract the effect that a positive longitudinal strain gives rise to a negative lateral strain of compression. The value of μ, which is a property of the material, may be obtained from standard tables and is generally of the value 0.3. In Equation (6.38), the factor $(1-\mu^2)^{-1}$ embodies the effective increase in the stiffness of the plate resulting from the curling.

In solving Equation (6.38) it is assumed that

$$\mathbf{z} = \mathbf{\Psi}(r)^{i\omega t}$$

which is then substituted into that equation to give

$$\nabla^2(\nabla^2\mathbf{\Psi}) - K^4\mathbf{\Psi} = 0 \tag{6.39}$$

in which

$$K^4 = \frac{\omega^2\rho(1-\mu^2)}{\kappa^2 E}$$

The substitution of the Helmholtz equation $\Lambda^2\mathbf{\Psi} = -K^2\mathbf{\Psi}$ into Equation (6.39) indicates that if $\mathbf{\Psi}$ can satisfy the Helmholtz equation, it will also constitute a solution to Equation (6.39). The function $\mathbf{\Psi}$ in the relationship $\Lambda^2\mathbf{\Psi} = K^2\mathbf{\Psi}$ will also satisfy Equation (6.39), so the complete solution of this equation must be the sum of four independent solutions to

$$\nabla^2\mathbf{\Psi} \pm K^2\mathbf{\Psi} = 0 \tag{6.40}$$

Equation (6.40) with the positive sign is the Helmholtz equation with circular symmetry, which yields the solutions $J_0(Kr)$ and $Y_0(Kr)$. But the boundary condition that the displacement must be finite at $r = 0$ at the center of the plate requires that the latter solution must be scrapped. The solution of Equation (6.40) with the negative sign yields $J_0(iKr)$ and $Y_0(iKr)$; the latter term is also discarded. The term $J_0(iKr)$ is a modified Bessel function of the first kind, generally written as[2]

[2] The modified Bessel functions $I_n(x)$ are solutions of the modified Bessel differential equation

$$\frac{d^2y}{dx^2} + \frac{1}{x}\frac{dy}{dx} - \left(1 + \frac{n^2}{x^2}\right)y = 0$$

$I_0(Kr)$. The complete applicable solution of Equation (6.39) is

$$\Psi = \mathbf{A}J_0(Kr) + \mathbf{B}I_0(Kr) \tag{6.41}$$

The function $J_0(Kr)$ is an oscillating function that damps out with increasing r while $I_0(Kr)$ increases continuously with r.

The manner in which the plate is supported determines the conditions which are used to evaluate the constants **A** and **B**. A common type of support is one in which the circular plate is rigidly clamped at its periphery $r = a$. The boundary conditions therefore are

$$\Psi = 0 \quad \text{and} \quad \partial\Psi/\partial r = 0 \quad \text{at } r = a$$

These yield

$$\mathbf{A}J_0(Ka) = -\mathbf{B}I_0(Ka), \quad \mathbf{A}J_1(Ka) = -\mathbf{B}I_1(Ka) \tag{6.42}$$

and through elimination of the constants **A** and **B** we obtain the transcendental equation which gives the permissible values of Ka:

$$\frac{J_0(Ka)}{J_1(Ka)} = -\frac{I_0(Ka)}{I_1(Ka)} \tag{6.43}$$

Both I_0 and I_1 remain positive for all values of Ka, so solutions occur only when J_0 and J_1 have opposite signs. The sequence of solutions satisfying Equation (6.43) is

$$Ka = 3.20, 6.30, 9.44, 12.57, \ldots$$

The above can be approximated by $Ka = n\pi$, where $n = 1, 2, 3, \ldots$. This approximation improves with increasing values of n.

From the definition of K for Equation (6.39), it is apparent that the frequency can be found from

$$f = \frac{\omega}{2\pi} = \frac{\kappa K^2}{2\pi}\sqrt{\frac{E}{\rho(1-\mu^2)}}$$

By setting $K = 3.20/a$, the fundamental frequency f_1 is found to be

$$f_1 = \frac{\omega_1}{2\pi} = \frac{3.2^2}{2\pi a^2}\frac{b}{\sqrt{12}}\sqrt{\frac{E}{\rho(1-\mu^2)}} = 0.47\frac{b}{a^2}\sqrt{\frac{E}{\rho(1-\mu^2)}}$$

where b represents the thickness of the plate. The frequencies of the overtones are given by

$$f_2 = \left(\frac{6.3}{3.2}\right)^2 f_1 = 3.88 f_1$$

$$f_3 = 8.70 f_1, \text{ etc.}$$

These frequencies are spread out much further apart than those for the circular membrane.

For the fundamental mode of vibration, the displacement of a thin circular plate is given by

$$z_1 = \cos(\omega_1 t + \phi_1)\left[A_1 J_0\left(\frac{3.2}{a}r\right) + B_1 I_0\left(\frac{3.2}{a}r\right)\right]$$

From the boundary condition relationships of Equation (6.42) the last expression becomes

$$z_1 = A_1 \cos(\omega_1 t + \phi_1)\left[J_0\left(\frac{3.2}{a}r\right) + 0.555 I_0\left(\frac{3.2}{a}r\right)\right]$$

It is interesting to observe that the amplitude at the center $r = 0$ is $1.0555A_1$, not A_1. If we compare the shape function represented by the bracketed terms on the right-hand side of the last equation with the corresponding shape function $J_0(2.405r/a)$ for the fundamental mode of a similar-sized vibrating circular membrane, it will be found that the relative displacement of the plate near its edge is considerably smaller than that of the membrane. Hence, the ratio of the average amplitude to the amplitude at the center is less than that in the case for the membrane. The average displacement amplitude is given by

$$\langle \Psi_1 \rangle_s = 0.326A_1 = 0.309z_0$$

where $z_0 = 1.0555A_1$ represents the amplitude at the center $r = 0$ of the plate. In the same manner we used to represent the membrane, the circular plate can be depicted by an equivalent flat piston so that

$$S_{eq}\zeta_{eq} = 0.309\pi a^2 z_0$$

Plates can also undergo loaded and forced vibrations. The mathematical treatments of these cases are analogous to those for membranes, and the response curves are similar to those shown in Figure 6.4. Large amplitudes will also occur at resonance frequencies unless there is appreciable damping.

The most apparent use of the vibrating thin plate is that of the telephone diaphragms (both receiver and microphone). While these diaphragms do not provide the flatter frequency responses or frequency range of membranes in condenser microphones, they do provide adequate intelligibility, are generally far more rugged in their construction and cheaper to manufacture. Sonar transducers used to generate underwater sounds less than 1 kHz constitute another class of vibrating plates; the signals are produced by the variations of an electromagnetic field in an electromagnet positioned closely to a thin circular steel plate.

References

Fletcher, Neville H. and Rossing, Thomas D. 1998. *The Physics of Musical Instruments*, 2nd ed. New York: Springer-Verlag, Chapter 3.

Kinsler, Lawrence E., Frey, Austin R., Coppens, Alan B., and Sanders, James V. 1982. *Fundamentals of Acoustics*, 3rd ed. New York: John Wiley & Sons, Chapter 4.

Morse, Philip M. and Ingard, K. Uno. 1968. *Theoretical Acoustics*. New York: McGraw-Hill, Sections 5.2 and 5.3.
Reynolds, Douglas R. 1981. *Engineering Principles of Acoustics, Noise and Vibration Control*. Boston: Allyn and Bacon, pp. 247–255.
Wood, Alexander. 1966. *Acoustics*. New York: Dover Publications, pp. 429–436.

Problems for Chapter 6

All membranes described below may be assumed to be fixed at their perimeters unless otherwise indicated.

1. Consider a square membrane, having dimensions $b \times b$, vibrating at its fundamental frequency with amplitude δ at its center. Develop an expression that gives the average displacement amplitude. Obtain a general expression for points having an amplitude of $\delta/2$. Plot at least five points from this general expression. Do these points fall in a circle?
2. A rectangular membrane has width b and length $3b$. Find the ratio of the first three overtone frequencies relative to the fundamental frequency.
3. Consider a circular membrane with a free rim. Develop the general expression for the normal modes and sketch the nodal patterns for the three normal modes with the lowest natural frequencies. Express the frequencies of these normal modes in terms of tension and surface density.
4. A circular aluminum membrane of 2.5 cm radius and 0.012 cm thickness is stretched with a tension of 15,000 N/m. Find the first three frequencies of free vibration, and for each of the frequencies, determine any nodal circles.
5. Prove that the total energy of a circular membrane vibrating in its fundamental mode is equal to $0.135\pi\rho_s(a\omega A_f)^2$ where ρ_s is the area density, a the radius of the membrane, and ω the angular frequency of the vibration, and A_f the fundamental amplitude at the center.
6. Steel has a tensile strength of 1.0 GPa ($= 10^9$ Pa) and aluminum, 0.2 GPa. Using these values as the maximum tensions, what will be the maximum fundamental frequency of a 2-cm-diameter circular membrane made up of each of these materials? *Note*: for thin membranes these fundamental frequencies are independent of the thicknesses.
7. A damping force is applied uniformly over the surface of a circular membrane. This damping force per unit area $= -\Im\, \partial z/\partial t$ should be introduced into the appropriate wave equation in a manner consistent with the dimensions of the terms of the equation. Solve the equation to demonstrate that the amplitudes of the free vibrations are damped exponentially as $e^{-1/2\Im t/\rho_s}$.
8. A kettledrum consists of a circular membrane of 50 cm diameter, with an area density of 1.0 kg/m^2. The membrane is stretched under a tension of 10,000 N/m.
   (a) Determine the fundamental frequency of the membrane without a backing vessel.

(b) Determine the fundamental frequency for the membrane with a backing vessel that is a hemispherical bowl of 25 cm radius. The vessel is filled with air at a pressure of 100 kPa, and γ (the ratio of specific heats) is 1.4.

9. An undamped membrane of 4-cm radius has an area density of 1.6 kg/m^2 and is stretched to a tension of 1200 N/m. It is driven by a uniform pressure 7000 sin ωt Pa applied over the entire surface.
   (a) Determine and plot the amplitude of the displacement at the center as a function of frequency ranging 0–2 kHz.
   (b) Compute and plot the shape of the membrane when driven by the applied frequency of 600 Hz and the applied frequency of 1000 Hz.
10. A condenser microphone contains a circular aluminum diaphragm of 30-mm diameter and 0.02-mm thickness. Aluminum has a maximum tensile strength of 0.2 GPa.
   (a) What is the allowable maximum tension in N/m in the diaphragm?
   (b) What will be the fundamental frequency under these conditions?
   (c) What will be the displacement of the diaphragm at its center under the impetus of a 500-Hz sound wave having a pressure amplitude of 1.5 Pa?
   (d) What will be the average displacement under the conditions of (c)?
11. If the volume of air trapped behind the diaphragm of the condenser microphone of the preceding problem is 2.5×10^{-7}m^3, by how much will the fundamental frequency be raised? Assume the normal air pressure to be equal to 100 kPa and $\gamma = 1.4$.
12. Use integration over the surface of a circular thin plate vibrating in its fundamental mode to show that the average displacement amplitude is $0.327A$, where A denotes the displacement in the center of the plate.
13. The diaphragm of a typical telephone receiver comes in the form of a circular sheet of steel, 4 cm in diameter and 0.18 mm thick.
   (a) What is the fundamental frequency, if the diaphragm is rigidly clamped at its rim?
   (b) How will this fundamental frequency change if the diaphragm thickness is doubled?
   (c) What would happen to the fundamental frequency if the diameter of the diaphragm was increased by 50%?
14. Find the fundamental frequency of a vibrating circular steel plate which is clamped at its rim and is of 25 cm diameter and 0.55 mm thickness.

7
Pipes, Waveguides, and Resonators

7.1 Introduction

In dealing with strings, bars, and membranes in Chapters 4–6, we considered relatively simple geometric conditions. The situation becomes more complex when the sound waves are confined in a restricted amount of space. For example, when sound propagates inside a rigid-walled pipe with a wavelength that exceeds the radius of a rigid-wall pipe, the acoustic propagation inside the pipe becomes fairly planar. The resonance properties of the pipes driven at one end and closed off at other end constitute the basis for measuring acoustical impedances and absorption properties of materials. In our study of pipes we establish the models for physical analyses of wind musical instruments, organ pipes, and ventilation ducts (Fletcher and Rossing, 1991). In larger spaces, where the dimensions may exceed wavelengths, two- and three-dimensional standing waves can occur. We shall treat the simple case of a waveguide with a uniform cross section, establish the concept of group speed and phase speed which occurs with a wave propagating inside a waveguide. The acoustic waveguide is very much analogous to the electromagnetic waveguides, and it finds applications in surface-wave delay lines and in the propagation of sound in ocean and atmospheric layers. We shall also consider the physics of a Helmholtz resonator.

7.2 The Simplest Enclosed System: Infinite Cylindrical Pipe

The simplest enclosed system inside which sound propagation occurs is an infinite cylindrical pipe with its axis parallel to the direction of the propagation of the plane wave in the enclosed medium. The pipe wall is assumed to be rigid, perfectly smooth, and adiabatic (i.e., no heat transfer occurs through the wall). The pipe thus has no effect on the wave propagation. A pressure wave generated by a piston moving in the x-direction can be expressed as

$$\mathbf{p}(\mathbf{x},\mathrm{t}) = \mathbf{p}_0\, e^{i(\omega t - kx)} \tag{7.1}$$

Here $\mathbf{p}_0$ is the maximum amplitude of the pressure wave. The volume flow is given by

$$U(x,t) = \frac{p_0 S}{\rho c}\, e^{i(\omega t - kx)}$$

where ω is the angular frequency, $k = 2\pi/\lambda = \omega/c$ the wave number, S the cross sectional area of the pipe, ρ the density of the fluid inside the pipe, and c the propagation velocity.

7.3 Resonances in a Close-Ended Pipe

As shown in Figure 7.1 consider a pipe of length L and cross-sectional area S, filled with a fluid, and sealed off at one end, $x = L$. Let the fluid inside the pipe be driven by a piston at $x = 0$. The pipe has a mechanical impedance $\mathbf{Z}_{nL}$. The piston vibrates harmonically at a sufficiently low frequency so that only plane waves are considered to exist inside the pipe. The wave inside the pipe can be described by

$$\mathbf{p} = \mathbf{A}e^{i[\omega t + k(L-x)]} + \mathbf{B}e^{i[\omega t - k(L-x)]} \tag{7.2}$$

where $\mathbf{A}$ and $\mathbf{B}$ are established by the boundary conditions at $x = 0$ and $x = L$.

At $x = L$ the mechanical impedance of the wave must equal the mechanical impedance $\mathbf{Z}_{nL}$ at the termination so as to sustain the continuities of force and particles. The force of the fluid acting at the end of the pipe is $\mathbf{p}(L, t)S$, and the corresponding particle speed $u(L, t)$ derives from the integrated Equation (2.23)

$$u = -\int \frac{1}{\rho}\delta\left(\frac{\partial p}{\partial x}\right) dt$$

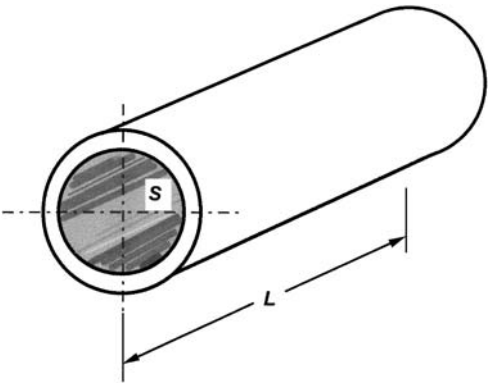


FIGURE 7.1. A pipe close-ended at $x = L$.

Mechanical impedance Z_n, expressed as

$$\mathbf{Z}_n = \frac{\mathbf{f}}{\mathbf{u}} \tag{7.3}$$

represents a complex value that is the ratio of the complex driving force **f** to the complex speed **u** at the point where the force is applied. In the case of the finite pipe, the mechanical impedance at $x = L$ is given by

$$\mathbf{Z}_{nL} = \rho_0 c S \frac{\mathbf{A} + \mathbf{B}}{\mathbf{A} - \mathbf{B}} \tag{7.4}$$

The value of the input mechanical impedance at $x = 0$ is expressed as

$$\mathbf{Z}_{n0} = \rho_0 c S \frac{\mathbf{A}e^{ikL} + \mathbf{B}e^{-ikL}}{1 + \dfrac{iZ_{nL}}{\rho_0 c S}\tan kL} \tag{7.5}$$

Eliminating **A** and **B** by combining Equations (7.4) and (7.5) yields

$$\frac{\mathbf{Z}_{n0}}{\rho_0 c S} = \frac{\dfrac{Z_{nL}}{\rho_0 c S} + i\tan kL}{1 + \dfrac{Z_{nL}}{\rho_0 c S}\tan kL} \tag{7.6}$$

The term $\rho_0 c S$ is the *characteristic mechanical impedance* of the fluid. The complex quantity $\mathbf{Z}_{n0}$ can be recast in terms of real and imaginary components, r and ψ, respectively,

$$\frac{\mathbf{Z}_{nL}}{\rho_0 c S} = r + i\psi \tag{7.7}$$

The ratio on the left-hand side of Equation (7.7) constitutes a *normalized* impedance. Inserting Equation (7.7) into Equation (7.6) yields

$$\begin{aligned}\frac{\mathbf{Z}_{n0}}{\rho_0 c S} &= \frac{(r + i\psi) + i\tan kL}{1 + i(r + i\psi)\tan kL} \\ &= \frac{r(\tan^2 kL + 1) - i[\psi\tan^2 kL + (r^2 + \psi^2 - 1)\tan kL - \psi]}{(\psi^2 + r^2)\tan^2 kL - 2\psi\tan kL + 1}\end{aligned} \tag{7.8}$$

When $r = 0$, the input impedance $\mathbf{Z}_{n0}$ vanishes when the reactance vanishes, i.e.,

$$\frac{-i[\psi\tan^2 kL + (r^2 + \psi^2 - 1)\tan kL - \psi]}{(\psi^2 + r^2)\tan^2 kL - 2\psi\tan kL + 1} = 0 \tag{7.9}$$

and this results in

$$\psi\tan^2 kL + (\psi^2 - 1)\tan kL - \psi = 0 \tag{7.10}$$

that is,

$$\psi = -\tan kL \tag{7.11}$$

The impedance becomes infinite when

$$\psi \tan^2 kL + (\psi^2 - 1) \tan kL - \psi = 0$$

or

$$\psi = \cot kL$$

Let us briefly examine the situation for a constant driving force at $x = 0$. The vanishing of $\mathbf{Z}_{n0} = \mathbf{f}/\mathbf{u}_0$ connotes that the speed amplitude at the point of force application ($x = 0$) is infinite, the condition for mechanical resonance. Conversely the input impedance reaching infinity means that the speed amplitude approaches zero, which describes the condition of *antiresonance.*

To obtain the condition of resonance in a pipe driven at $x = 0$ and sealed with a rigid cap at $x = L$, we let $|\mathbf{Z}_{nL}/\rho_0 cS|$ approach infinity in Equation (7.4), giving

$$\frac{\mathbf{Z}_{n0}}{\rho_0 cS} = -i \cot kL$$

The reactance becomes zero when $\cot kL = 0$,

$$k_n L - (2n - 1)\pi/2 \qquad n = 1, 2, 3, \ldots$$

and so we obtain the set of resonant frequencies as for the forced-fixed string:

$$f_n = \frac{2n - 1}{4} \frac{c}{L}$$

With the odd harmonics of the fundamental constituting the resonance frequencies, the driven closed pipe contains a pressure antinode at $x = L$ and a pressure node at $x = 0$. This means that the driver must present a vanishing mechanical impedance to the tube.

7.4 The Open-Ended Pipe

Consider a pipe driven at $x = 0$ but open-ended at the other end $x = L$. The assumption that $\mathbf{Z}_{nL} = 0$ at $x = L$ (which would lead to resonances at $f_n = {}^1\!/_2\, nc/L$) is not valid, because the open end of the pipe radiates into the surrounding air. The appropriate condition is $\mathbf{Z}_{nL} = \mathbf{Z}_r$, where $\mathbf{Z}_r$ is the radiation impedance at the open end of the tube. Also, the presence of a flange at the open end affects the exit impedance. Consider the case of a flange at the open end of a circular pipe of radius a. The flange is large with respect to the wavelength of the sound, which, in turn, is considerably larger than the tube radius ($\lambda \gg a$). This situation resembles a baffled piston in the low-frequency limit. From theory (Kinsler and Frey, 1962)

$$\frac{\mathbf{Z}_{nL}}{\rho_0 cS} = \frac{(ka)^2}{2} + i \frac{8}{3} \frac{ka}{\pi} \tag{7.12}$$

where the real component $r = (ka)^2/2$ and the imaginary component $\psi = 8ka/3\pi$ are both much less than unity and $r \ll \psi$. Under these conditions the solution to

Equation (7.9) yields $\tan kL = -\psi$ in order for resonance frequencies to occur. With the assumption $\psi \ll 1$, we obtain

$$\tan(n\pi - k_n L) = \frac{8ka}{3\pi} \approx \tan \frac{8ka}{3\pi} \tag{7.13}$$

where $n = 1, 2, 3, \ldots$. Hence

$$n\pi = k_n L + \frac{8k_n a}{3\pi} \tag{7.14}$$

The resonance frequencies therefore are

$$f_n = \frac{n}{2} \frac{c}{L + \frac{8}{3\pi} a} \tag{7.15}$$

All of the resonance frequencies are harmonics of the fundamentals. We also note that the denominator $L + 8a/(3\pi)$ constitutes the effective length of the pipe rather than actual length L.

In the case of the *unflanged* pipe, the radiation impedance, indicated by both theory and experiment is given approximately by

$$\frac{\mathbf{Z}_{nL}}{\rho_0 c S} = \frac{(ka)^2}{4} + i(0.6ka) \tag{7.16}$$

Here the end correction for the unflanged pipe equals $0.6a$, with the effective length being $L + 0.6a$. We also note that the end corrections do not depend on the frequency. Providing that $\lambda_n \gg a$, the resonance frequencies of flanged and unflanged pipes constitute harmonics of the fundamental. Hence, the driving frequency of an open-ended organ pipe yields resonances that are harmonics of the driving frequency. The above exposition so far has dealt with pipes of constant cross sections. If a pipe is *flared* at the open end, as is the case with many wind instruments such as the clarinet and the oboe, the results are modified, and the resonances may not necessarily be harmonics of the fundamental. Variations in the flare design will emphasize or lessen certain harmonics present in the forcing function, thereby affecting the quality or timbre of the sound emanated by the pipe.

7.5 Radiation of Power from Open-Ended Pipes

Equation (7.4) may be revised to read

$$\frac{\mathrm{B}}{\mathrm{A}} = \frac{\frac{\mathrm{Z}_{nL}}{\rho_0 c S} - 1}{\frac{\mathrm{Z}_{nL}}{\rho_0 c S} + 1} \tag{7.17}$$

When the termination impedance $\mathbf{Z}_{nL}$ is known the power transmission coefficient T_n can be established from

$$T_n = 1 - \left| \frac{\mathbf{B}}{\mathbf{A}} \right|^2 \tag{7.18}$$

Through the use of Equation (7.16), Equation (7.17) applied to the case of an open-ended pipe becomes

$$\frac{\mathbf{B}}{\mathbf{A}} = \frac{\left[1 - \frac{(ka)^2}{2}\right] - i\frac{8ka}{3\pi}}{\left[1 + \frac{(ka)^2}{2}\right] + i\frac{8ka}{3\pi}} \tag{7.19}$$

which is then inserted into Equation (7.18), which now becomes

$$T_n = \frac{2(ka)^2}{\left[1 + \frac{(ka)^2}{2}\right]^2 + (ka)^2\left(\frac{8}{3\pi}\right)^2} \tag{7.20}$$

Normally $ka \ll 1$, so the power transmission coefficient is quite small and it can be simplified to

$$T_n \approx 2(ka)^2 \quad \text{(for flanged pipe)} \tag{7.21}$$

From Equation (7.18) it can be ascertained that $\mathbf{B}/\mathbf{A}$ is almost equal to –1. The pressure amplitude of the reflected wave is barely less than that of the incident wave. At $x = L$, its pressure differs by nearly 180° out of phase. On the other hand, the incident and reflected particle speeds remain nearly in phase at the opening of the pipe, so that the location is (nearly) the antinode of the particle speed. Even though the amplitude of the particle speed at the opening is nearly twice that of the incident wave, only a small percentage of the incident power transmits out of the flanged pipe. Thus, sources having dimensions small compared with the wavelength of the sound behave as inefficient radiators of sonic energy.

Inserting Equation (7.16) for the unflanged pipe into Equation (7.17), the transmission coefficient becomes

$$T_n = \frac{(ka)^2}{\left[1 + \frac{(ka)^2}{4}\right]^2 + (0.6ka)^2} \approx (ka)^2 \tag{7.22}$$

By comparing Equations (7.20) and (7.22) we perceive that adding a flange at the end of the pipe essentially doubles the radiation of sound at low frequencies. A gradual flare at the open terminal of the pipe will increase the low-frequency power transmission even more.

7.6 Standing Waves in Pipes

The existence of phase interference between transmitted and reflected waves inside a terminated pipe gives rise to a standing wave pattern. The properties of the standing waves can be applied to gauge the load impedance. In Equation (7.2) let us set

$$\mathbf{A} = A, \qquad \mathbf{B} = Be^{i\theta} \tag{7.23}$$

where A and B are real, positive numbers. Combining Equations (7.4) and (7.23) results in

$$\frac{Z_{nL}}{\rho_0 c S} = \frac{1 + \frac{B}{A}e^{i\theta}}{1 - \frac{B}{A}e^{i\theta}} \tag{7.24}$$

Inserting Equation (7.23) into Equation (7.2) we obtain for the pressure amplitude $p = |\mathbf{p}|$ of the wave

$$p = \sqrt{(A+B)^2 \cos^2[k(L-x) - \theta/2] + (A-B)^2 \sin^2[k(L-x) - \theta/2]} \tag{7.25}$$

At a pressure antinode, the pressure amplitude is $A + B$, while the amplitude pressure at the pressure node is $A - B$. The *standing wave ratio* (SWR) occurs as the ratio of these two pressure amplitudes, respectively,

$$\text{SWR} = \frac{A+B}{A-B} \quad \text{or} \quad \frac{B}{A} = \frac{\text{SWR} - 1}{\text{SWR} + 1} \tag{7.26}$$

SWR is measured by probing the sound field along the pipe (also known as an *impedance tube*) with a tiny microphone to obtain the value of B/A. The phase angle θ can be found by measuring the distance of the first node from the end at $x = L$. According to Equation (7.25) the nodes are located at

$$k(L - x_n) - \frac{\theta}{2} = \left(n - \frac{1}{2}\right)\pi \tag{7.27}$$

For the first node ($n = 1$),

$$\theta = 2k(L - x_1) - \pi \tag{7.28}$$

Example Problem 1

An impedance tube is found to have a SWR of 3 and the first node is 3/8 of the wavelength from the end. Find the normalized mechanical impedance at $x = L$.

Solution

$$L - x = 3\lambda/8$$

Hence from Equation (7.28) $\theta = 2(2\pi/\lambda)(3\lambda/8) - \pi = \pi/2$. From Equation (7.26)

$$\frac{B}{A} = \frac{3-1}{3+1} = \frac{1}{2}$$

Equation (7.24) now reads

$$\frac{Z_{nL}}{\rho_0 c S} = \frac{1 + \frac{1}{2}e^{i\pi/2}}{1 - \frac{1}{2}e^{i\pi/2}} = 0.60 + i0.80$$

Mechanical impedances at terminations occur as complicated functions of the frequencies, so it may be necessary to conduct measurements over the range of frequencies under consideration. Smith nomographs (Beranek, 1949) are useful tools to expedite computations for r and ψ, the real and imaginary components of the impedances, from the measurements of the standing wave ratio and the position of the node most adjacent to the end.

The impedance tube is used to measure the reflective and absorptive properties of small sections of materials, such as acoustic tiles and sound control absorbers, mounted at the end of the tube.

7.7 The Rectangular Cavity

In Figure 7.2 a rectangular cavity is shown having dimensions L_x, L_y, L_z in the x-, y-, and z-directions, respectively. This parallelepiped can represent a simple auditorium or any other rectangular space that contains rigid walls, few windows and other openings. We assume the walls of the cavity to be perfectly rigid that $\hat{\mathbf{n}} \cdot \vec{\mathbf{u}} = 0$ at all of the boundaries (i.e., the walls will not move in the directions of their normals). This also means that $\hat{\mathbf{n}} \cdot \nabla \mathbf{p} = 0$, i.e.,

$$\left.\begin{aligned} \left(\frac{\partial p}{\partial x}\right)_{x=0} &= \left(\frac{\partial p}{\partial x}\right)_{x=L_x} = 0 \\ \left(\frac{\partial p}{\partial y}\right)_{y=0} &= \left(\frac{\partial p}{\partial y}\right)_{y=L_y} = 0 \\ \left(\frac{\partial p}{\partial z}\right)_{z=0} &= \left(\frac{\partial p}{\partial z}\right)_{z=L_z} = 0 \end{aligned}\right\} \tag{7.29}$$

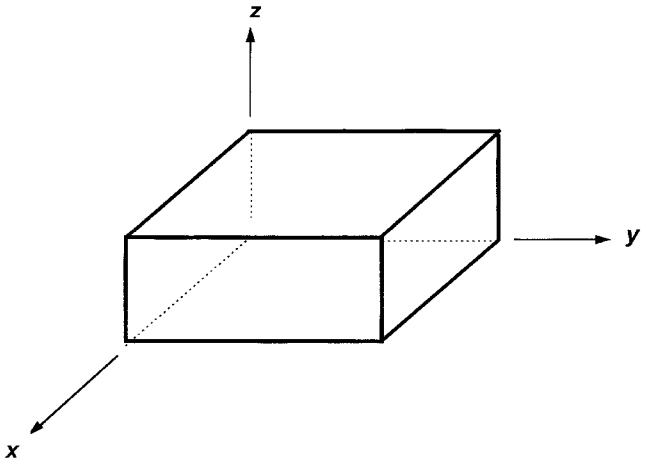


FIGURE 7.2. The rectangular cavity.

Because acoustic energy cannot escape from a completely closed cavity with rigid walls, standing waves constitute the only appropriate solutions of the wave equation. Inserting

$$p(x, y, z, t) = X(x)Y(y)Z(z)e^{i\omega t} \tag{7.30}$$

into the wave equation

$$\nabla^2 p = \frac{1}{c^2}\frac{\partial p}{\partial t} \tag{7.31}$$

and separating variables results in the following set of equations:

$$\left(\frac{d^2}{dx^2} + k_x^2\right) X = 0, \quad \left(\frac{d^2}{dy^2} + k_y^2\right) Y = 0, \quad \left(\frac{d^2}{dz^2} + k_z^2\right) Z = 0 \tag{7.32}$$

Here the separation constants are related as follows:

$$k^2 = \frac{\omega^2}{c^2} = k_x^2 + k_y^2 + k_z^2 \tag{7.33}$$

The boundary conditions of Equation (7.29) stipulate cosine solutions, and Equation (7.30) revises to

$$p_{lmn} = \mathrm{A}_{lmn} \cos k_{xl} x \cos k_{ym} y \cos k_{zn} x \, e^{i\omega_{lmn} t} \tag{7.34}$$

where the components of k are

$$\begin{aligned} k_{xl} &= \frac{l\pi}{L_x}, \qquad l = 0, 1, 2, \ldots, \qquad k_{ym} = \frac{m\pi}{L_y}, \qquad m = 0, 1, 2, \ldots \\ k_{zn} &= \frac{n\pi}{L_z}, \qquad n = 0, 1, 2, \ldots, \end{aligned} \tag{7.35}$$

This leads to the quantization of allowable frequencies of vibration

$$\frac{\omega_{lmn}}{c} = \sqrt{\frac{l^2\pi^2}{L_x^2} + \frac{m^2\pi^2}{L_y^2} + \frac{n^2\pi^2}{L_z^2}} \tag{7.36}$$

The above gives rise to eigenfunctions of Equation (7.34). Each eigenfunction is characterized by its own eigenfrequency (7.36) specified by the ordered integers (*l,m,n*). Equation (7.34), which is the solution to the wave equation (7.31), yields three-dimensional standing waves in the cavity with nodal planes parallel to the walls. The pressure varies sinusoidally between these nodal planes. In the same manner that a standing wave on a string could be resolved into a pair of waves traveling in opposite directions, we can separate the eigenfunctions in the rectangular cavity into traveling plane waves. This is done by casting the solutions (7.34) into complex exponential form and expanding it as a sum of products:

$$\mathbf{p}_{lmn} = \frac{\mathbf{A}_{lmn}}{8} \sum e^{i\omega_{lmn} t(\pm k_x x \pm k_x y \pm k_z z)} \tag{7.37}$$

where the summation is taken over all permutations of plus and minus signs. There are eight terms in all, each representing a plane wave traveling along the

direction of its propagation vector $\hat{\mathbf{k}}$ which has projections $\pm k_{xl'}$, $\pm k_{ym'}$, $\pm k_{zn'}$ on the coordinate axes. The standing wave solution results from the superposition of eight traveling waves (one into each quadrant) whose directions of travel are constrained by the boundary conditions. The methodology involving separation of variables can also be used to treat standing waves in other simple cavities, such as cylindrical and spherical cavities, with the eigenfunctions entailing Bessel and Legendre functions.

7.8 A Waveguide with Constant Cross Section

In Figure 7.3 a waveguide of rectangular cross section is assumed to have rigid side walls and a source of acoustic energy located at its boundary z = 0. There is no other boundary on the z-axis, which permits energy to propagate down the waveguide. This results in a situation where the wave pattern consists of standing waves in the transverse directions x and y and a traveling wave in the z-direction. The mathematical solution which would contain applicable eigenfunctions is

$$p_{lmn} = A_{lmn} \cos k_{xl} x \cos k_{ym} y e^{i(\omega t - k_z z)} \tag{7.38}$$

Upon substituting Equation (7.38) into the wave equation (7.31) we obtain the following relationship:

$$\frac{\omega^2}{c^2} = k^2 = k_{xl}^2 + k_{ym}^2 + k_z^2 \tag{7.39}$$

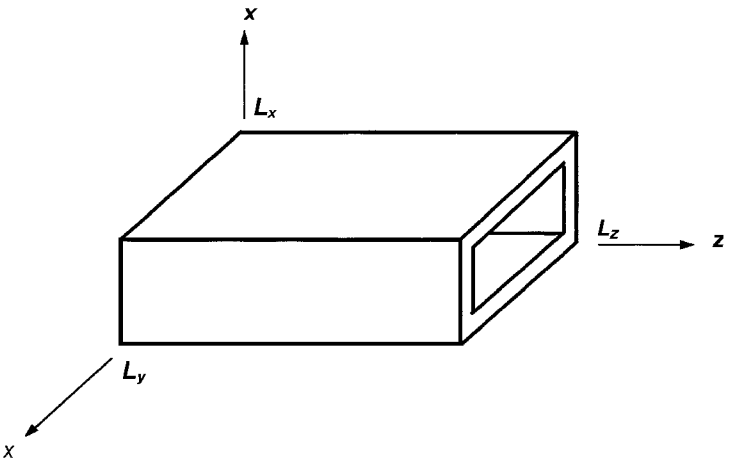


FIGURE 7.3. A waveguide having a rectangular cross section. The travel is along the z-axis.

with permitted values of k_{xl} and k_{ym}, resulting from the boundary conditions of rigidity, these being

$$k_{xl} = \frac{l\pi}{L_x}, \qquad l = 0,\ 1,\ 2, \ldots$$
$$k_{ym} = \frac{m\pi}{L_y}, \qquad m = 0,\ 1,\ 2, \ldots \tag{7.40}$$

We rearrange Equation (7.39) to find k_z

$$k_z = \sqrt{\frac{\omega^2}{c^2} - k_{xl}^2 - k_{ym}^2} \tag{7.41}$$

Because ω may have any value, Equation (7.38) comprises a solution for all values of ω, in contrast to the totally enclosed cavity which allows for only quantized frequencies. Setting

$$k_{lm} = \sqrt{k_{xl}^2 + k_{ym}^2} \tag{7.42}$$

we can shorten Equation (7.41) to

$$k_z = \sqrt{\frac{\omega^2}{c^2} - k_{lm}^2} \tag{7.43}$$

The value k_z is real when $\omega/c > k_{lm}$. We then obtain a *propagating mode*, as the wave moves in the $+z$-direction. The cutoff frequency occurs when $\omega/c = k_{lm}$, so defining the limit for which k_z remains real; is given by

$$\omega_{lm} = ck_{lm} \tag{7.44}$$

for the (l, m) mode. A frequency below the threshold value of ω_{lm} results in a purely imaginary value of k_z,

$$k_z \equiv \pm i\sqrt{k_{lm}^2 - \frac{\omega^2}{c^2}} \tag{7.45}$$

We need to include the negative sign in Equation (7.45) so that $\mathbf{p} \to 0$ as $z \to \infty$ and the eigenfunctions assume the form

$$p_{lm} = \mathbf{A}_{lm} \cos k_{xl}x \cos k_{ym}y\, e^{-\left(\sqrt{k_{lm}^2 - \frac{\omega^2}{c^2}}\right)z} e^{i\omega t} \tag{7.46}$$

Equation (7.46) represents a standing wave that decays exponentially with z. This form of eigenfunction is termed an *evanescent mode*—i.e., no energy is propagated along the waveguide. If the frequency exciting the waveguide is just below the cutoff value of some particular mode, then this and higher modes are evanescent and are of little consequence at appreciable distances from the source. The lower modes may propagate energy and can be detected at large distances from the source.

Only plane waves can propagate in a rigid-walled waveguide if the frequency of the sound is sufficiently low. This frequency is less than $c/(2L)$, where L is the larger dimension of the rectangular cross section.

From Equation (7.38), the phase speed for mode lm is not c but

$$c_p = \omega/k_z = \frac{c}{\sqrt{1-\left(\frac{k_{lm}}{k}\right)^2}} = \frac{c}{\sqrt{1-\left(\frac{\omega_{lm}}{\omega}\right)^2}} \tag{7.47}$$

A physically meaningful solution can be written in the complex exponential form

$$\mathbf{p}_{lm} = \frac{1}{4}\mathbf{A}_{lm}\sum_{\pm} e^{i(\omega t \pm k_{xl}x \pm k_{ym}y - k_z z)} \tag{7.48}$$

The absence of the boundary at a location on the $+z$-axis necessitates only a negative sign for k_z, since there will be no reflected waves propagating in the $-z$-direction. The propagation vector $\hat{\mathbf{k}}$ for each of the four traveling waves forms an angle θ with the z-axis, according to

$$\frac{\cos\theta = k_z}{k = \sqrt{1-\left(\frac{\omega_{lm}}{\omega}\right)^2}} \tag{7.49}$$

and the corresponding phase speed c_p is

$$c_p = \frac{c}{\cos\theta} \tag{7.50}$$

The surfaces of constant phase for two component waves representing the (0,1) mode are shown in Figure 7.4 for a rigid waveguide. The waves cancel each other precisely for $y = 1/2L_y$, with the result there exists a nodal plane halfway

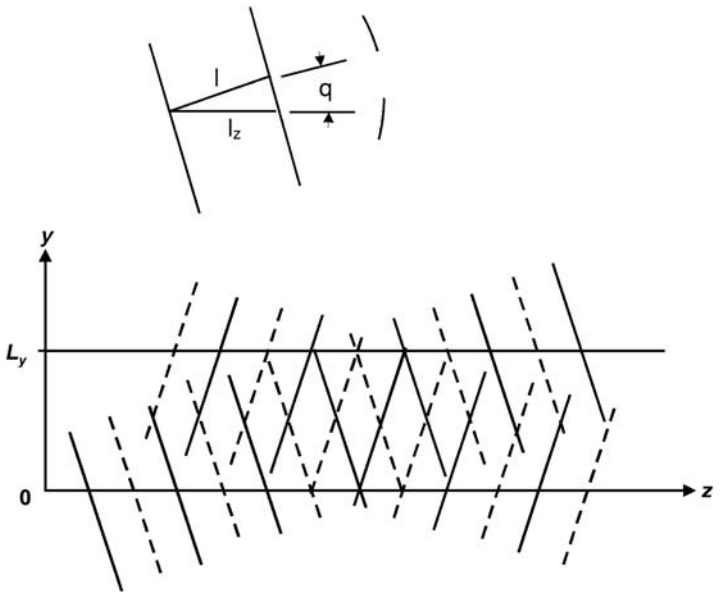


FIGURE 7.4. The surfaces of constant phase for two-component waves in the (0,1) mode.

between the walls. The waves remain in phase at the upper and the lower walls, so the pressure amplitude is maximized at these rigid boundaries. The apparent wavelength λ_z in the z-direction is given by

$$\lambda_z = \frac{\lambda}{\cos\theta} \tag{7.51}$$

In the lowest mode (0,0), $k_z = k$ and the four component waves form a single plane wave that travels down the axis of the waveguide with speed c. For all the other modes, the propagation vectors of the component waves generally form angles with the z-axis, with one aimed into each of the four forward quadrants. According to Equation (7.47), at frequencies much greater than the cutoff of the (l, m) mode, i.e., $\omega \gg \omega_{lm}$, the angle θ approaches zero and the waves are traveling fairly straight down the waveguide. As the input frequency approaches the cutoff value, θ increases with the result that the component waves move in increasingly transverse directions. In fact, when the frequency reaches the stage that $\omega = \omega_{lm}$, the component waves are traveling transversely to the axis of the waveguide. Each component wave carries energy along the waveguide through the process of continual reflection from the rigid walls (in the same manner a signal is transmitted through a fiberoptic line, bouncing from one wall back to the opposite wall down along the line). With the energy of the wave propagating at a speed c in the direction $\hat{\mathbf{k}}$, the corresponding speed c_p (the *group speed*) of the energy in the z-direction is given by the component of the plane-wave velocity $\hat{c}$ along the waveguide axis:

$$c_g = c\cos\theta = c\sqrt{1 - \left(\frac{\omega_{lm}}{\omega}\right)^2} \tag{7.52}$$

Given the driving frequency ω, each normal mode, in which $\omega_{lm} < \omega$, possesses its own set of values for c_p and c_g. Figure 7.5 illustrates the variation of the group and phase speeds as functions of frequency for three modes in a rigid-walled waveguide. The (0,0) mode is simply the plane-wave solution with its group and plane speeds equal to c for all driving frequencies.

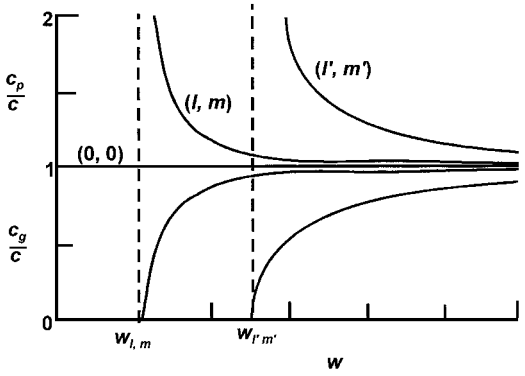


FIGURE 7.5. Group and phase speeds as functions of radial frequency for three modes in a rigid-walled waveguide.

7.9 Boundary Condition at the Driving End of the Waveguide

An impedance match must be made at the waveguide entry $z = 0$ so that the permissible mode solutions comply with the acoustic behavior of the active surface. If we know the pressure or velocity distribution of the driving source, these can be correlated with the behavior of $\mathbf{p}(x, y, 0, t)$ for the pressure or $\hat{z} \cdot \vec{\mathrm{u}}(x, y, 0, t)$ for the velocity. If the pressure distribution of the source is given, then the boundary condition becomes

$$\mathbf{p}(x,\ y,\ z,\ t) = \mathbf{P}(x,\ y)e^{i\omega t} \tag{7.53}$$

However, the left-hand side of Equation (7.53) can also be expressed as a superposition of the normal modes of the waveguide, i.e.,

$$\mathbf{p}(x, y, z, t) = \sum_{l,m} \mathbf{A}_{lm} \cos k_{xl}x \cos k_{ym}y\, e^{i(\omega t - k_z t)} \tag{7.54}$$

Setting $z = 0$, Equation (7.53) becomes

$$\mathbf{P}(x,\ y) = e^{i\omega t} \sum \mathbf{A}_{lm} \cos k_{xl}x\ \cos k_{ym}y \tag{7.55}$$

With Equation (7.55) we can establish the values of $\mathbf{A}_{lm}$ by applying Fourier's theorem, first for the x direction and then for the y direction.

7.10 Rigid-Walled Circular Waveguide

The treatment of the rigid-walled circular waveguide of radius $r = a$ is fairly straightforward, beginning with

$$\mathbf{P}_{ml} = \mathbf{A}_{ml} J_m(k_{ml}r)\ \cos(m\theta)\, e^{(i\omega t - k_z t)}$$

where $(r,\ \theta,\ z)$ are the customary cylindrical coordinates, J_m is the mth-order Bessel function, and

$$k_z = \sqrt{\frac{\omega^2}{c^2} - k_{lm}^2}$$

where k_{ml} is found from the boundary conditions. Because $r \cdot \nabla p = 0$ at the wall $r = a$,

$$k_{ml} = \frac{j'_{ml}}{a}$$

where j'_{lm} are the zeros of $\frac{dJ_m(z)}{dz}$. When the values of k_{ml} are determined, the applicable results developed for the rectangular waveguide may be applied by substituting the values of k_{ml} for the circular waveguide. As with the rectangular unit, the (0,0) mode of the circular waveguide is a plane wave that propagates with phase velocity $c_p = c$ for all $\omega > 0$. For frequencies below $f_{1,1}$, only plane waves can propagate in a rigid-walled circular waveguide.

7.11 The Helmholtz Resonator

Many analyses of acoustic devices become simplified with the assumption that the wavelength in the propagation fluid is much greater than the dimensions of the devices. If the wavelength is indeed much larger in *all* dimensions, the acoustic variable may be time varying, but it is virtually independent of the distance within the confines of the device. In such a case, the device can be viewed as a harmonic oscillator with one degree of freedom; and in the long-wavelength limit, such a device can be considered a *lumped acoustic element*. An example of this type of device is the *Helmholtz resonator*, illustrated in Figure 7.6 (three types are shown). The resonator is a rigid-walled cavity of volume V, with a neck of area A and length L.

According to theory, if $\lambda \gg L$, the fluid in the neck behaves somewhat as a solid plug which vibrates back and forth. As the fluid (usually air) moves back and forth, the acoustic energy converts to heat as a result of friction along the neck. These losses can be increased by placing a light porous material across the neck or by placing the material inside the volume. Maximum sound absorption occurs at the resonant frequency of the mass of air in the neck with the "spring" supplied by the air resistance inside the enclosed volume. Very little sound is absorbed at other frequencies. However, for necks greater than 1 cm in diameter, the viscous losses are considerably less than those associated with radiation. For the purpose of this analysis we can ignore the effects of viscosity in analyzing the Helmholtz resonator viewed as an analogous spring-mass system. The air in the neck has a total effective mass

$$m = \rho_0 A L' \tag{7.56}$$

where the effective neck length L' is longer than the physical length L because of its radiation mass loading. We have seen earlier in this chapter that at low frequencies a circular opening of radius a is loaded with a radiation mass equal to that of the fluid occupying area πa^2; and effective length $8a/(3\pi) = 0.85a$ if terminated in a wide flange and $0.6a$ for an unflanged terminal. We assume the mass loading at the inner end of the neck is equivalent to a flanged termination.

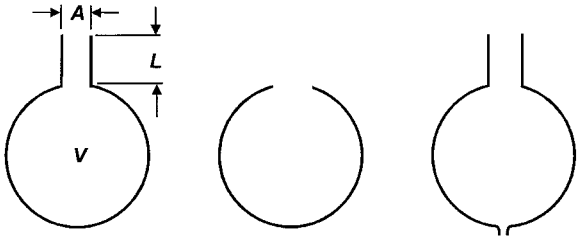


FIGURE 7.6. Three simple Helmholtz resonators.

Then,

$$L' = L + 2(0.85a) = L + 1.7a \quad \text{(flanged out end)}$$
$$L' = L + (0.85 + 0.6)a = L + 1.5a \quad \text{(unflanged outer end)} \tag{7.57}$$

Now consider the neck of the resonator to be fitted with an air-tight piston. When the piston travels a distance δ, the volume of the cavity changes by $\Delta V = -\delta A$. Then,

$$\frac{\Delta\rho}{\rho} = -\frac{\Delta V}{V} = \frac{A\delta}{V}$$

and from the thermodynamic relation

$$p = \rho_0 c^2 \frac{\Delta p}{p}$$

Combining the last two equations gives the following:

$$p = \rho_0 c^2 \frac{A\delta}{V} \tag{7.58}$$

The force $f = pA$ necessary to execute the displacement is $\rho_0 c^2 \frac{A^2}{V}\delta$. The effective stiffness S (from the spring formula $f = S\delta$) is

$$S = \rho_0 c^2 \frac{A^2}{V} \tag{7.59}$$

The fluid moving in the neck radiates sound into the surrounding medium in the same manner as an open-ended pipe. For wavelengths much larger than the radius of the neck, the radiation resistance [cf. Equations (7.12) and (7.16)] is

$$R_r = \rho_0 c \frac{k^2 A^2}{2\pi} \quad \text{(flanged)} \tag{7.60a}$$

or

$$R_r = \rho_0 c \frac{k^2 A^2}{4\pi} \quad \text{(unflanged)} \tag{7.60b}$$

The sound wave impinging on the neck opening is represented as an instantaneous driving force with a pressure amplitude P:

$$\mathbf{f} = AP e^{i\omega t} \tag{7.61}$$

We then can write the differential equation for the inward displacement δ of the fluid "plug" in the neck as

$$m\frac{d^2\delta}{dt^2} + R_r \frac{d\delta}{dr} + S\delta = AP e^{i\omega t} \tag{7.62}$$

This last equation is analogous to that of a sinusoidally driven oscillator which has analogous solutions. Solution of Equation (7.62) gives a complex displacement δ. The real part of the driving force represents the actual driving force $AP \cos \omega t$, and the real part of the complex displacement represents the actual displacement.

Because $\mathbf{f}$ is periodic with angular frequency ω, $\delta = \mathbf{B}e^{i\omega t}$ must be the solution where $\mathbf{B}$ exists as a complex constant. Then inserting the solution into Equation (7.62) results in

$$(-\mathbf{B}\omega^2 m + i\mathbf{B}\omega R_r + \mathbf{B}S)e^{i\omega t} = APe^{i\omega t}$$

Solving for $\mathbf{B}$ provides the following complex displacement:

$$\delta = \frac{APe^{i\omega t}}{i\omega\left[R_r + i\left(m\omega - \dfrac{S}{\omega}\right)\right]} \tag{7.63}$$

Differentiating Equation (7.63) yields the complex speed of the fluid in the resonator's neck,

$$\vec{\mathbf{u}} = \frac{\mathbf{A}\mathbf{P}e^{i\omega t}}{\mathbf{R}_r + i\left(m\omega - \dfrac{S}{\omega}\right)} \tag{7.64}$$

The impedance found from the relation $\mathbf{Z}_m = \mathbf{f}/\vec{\mathbf{u}}$, and making use of Equations (7.61) and (7.64) yields

$$Z_m = R_r + i\left(m\omega - \frac{S}{\omega}\right) \tag{7.65}$$

which means that the mechanical reactance is

$$X_m = m\omega - \frac{S}{\omega}$$

Resonance occurs when the reactance becomes zero, i.e.,

$$\omega_0 = \sqrt{\frac{S}{m}} = \sqrt{\frac{\rho_0 c^2 A^2/V}{\rho_0 L' A}} = c\sqrt{\frac{A}{L'V}} \tag{7.66}$$

We note that the resonance frequency depends on the volume of the cavity, not on the shape. Experimentation with differently shaped resonators having the same $S/L'V$ ratios indicate identical resonant frequencies. This holds true as long as all dimensions of the cavity are appreciably less than a single wavelength and the opening is quite small. There are additional frequencies in Helmholtz resonators higher than that given by Equation (7.66), which arise from standing waves in the cavity rather than from the oscillatory motion of the mass of air in the neck. These overtone frequencies are not harmonics of the driving fundamental, and the first overtone may be several times the frequency of the fundamental.

The sharpness (or narrowness of spread) of the resonance of a Helmholtz resonator is define by the *quality factor* Q,

$$Q = \frac{m\omega_0}{R_r} = 2\pi\sqrt{V\left(\frac{L'}{A}\right)^3} \tag{7.67}$$

Assumptions have been made, there are no losses other than those arising from acoustic radiation and the termination of the neck is flanged.

In conducting his experiments of complex musical tones, Helmholtz used a series of resonators of the type illustrated in Figure 7.6(c), with small nipples opposite the necks of the resonators. He used a graduated series of resonators, with differing volumes and neck sizes to achieve a wide range of resonant frequencies. When an incident sound wave contains a frequency component that corresponds to the resonant frequency of a Helmholtz device, a greatly amplified signal will be generated within the cavity of the resonator, and it can be aurally detected by connecting a short, flexible tube from the nipple to the ear. These phenomena led to the definition of pressure amplification, which is the ratio of the acoustic pressure amplitude P_c inside the cavity to the external driving pressure amplitude P. Equation (7.58) provides the pressure amplitude P_c. At resonance

$$|\delta| = \frac{PA}{\omega_0 R_r}$$

Then applying Equations (7.60) and (7.66), we find for the resonator with a flanged neck that

$$\frac{P_c}{P} = 2\pi\sqrt{V\left(\frac{L'}{A}\right)^3} = Q$$

The gain, therefore, is the same as the value of the quality factor defined in Equation (7.67), and we see the Helmholtz resonator behaving as an amplifier of gain Q at resonance, a fact which explains why a loudspeaker mounted in a closed chamber can be regarded as a Helmholtz resonator in which the air's reactance and the speaker cone mass constitute the effective mass of the system. The air resistance within the box and the cone's stiffness combine to provide the effective stiffness, while the effective resistance results from the sum of that attributable to the radiation of acoustic energy and that arising from the mechanical resistance of the speaker cone.

Example Problem 2

A Helmholtz resonator is a sphere of 8 cm internal diameter. If it is to resonate at 450 Hz in air, what is the hole diameter that should be drilled into the sphere?

Solution

Applying Equation (7.66) we have

$$\omega_0 = 450 \times 2\pi \text{ rad s}^{-1} = c\sqrt{\frac{A}{L'V}} = 341 \text{ m s}^{-1} = \sqrt{\frac{\pi a^2}{1.5a \times \frac{4}{3}\pi(0.04 \text{ m})^3}}$$

which results in $a = 0.00865$ m $= 0.865$ cm, or 1.73 cm diameter. Here we assumed an unflanged "pipe" with an effective value of $L' = 1.5a$.

References

Beranek, Leo L. 1986. *Acoustics*. New York: American Institute of Physics, Chapters 5 and 6.

Beranek, Leo L. 1949. *Acoustic Measurements*. New York: John Wiley & Sons, pp. 317–321.

Fletcher, Neville H. and Rossing, Thomas D. 1991. *The Physics of Musical Instruments*, 1st ed. New York: Spring-Verlag, pp. 150–155.

Kinsler, Lawrence E. and Frey, Austin R. 1962. *Fundamentals of Acoustics*, 2nd ed. New York: John Wiley & Sons, Chapter 8. (Many users consider this edition to be the best of the four issued to date.)

Kinsler, Lawrence E., Frey, Austin R., Coppens, Alan B., and Sanders, James V. 1982. *Fundamentals of Acoustics*, 3rd ed. New York: John Wiley & Sons, Chapters 9 and 10.

Olson, Harry F. 1967. *Music, Physics and Engineering*, 2nd ed. New York: Dover Publications, Chapter 4.

Wood, Alexander. 1960. *Acoustics*. New York: Dover Publications, pp. 103–110.

Problems for Chapter 7

1. Determine the minimum length of a pipe in which the input mechanical reactance is equal to the input mechanical resistance for the frequency of 700 Hz. Assume the loading impedance is four times the pipe's characteristic mechanical impedance $\rho_0 cA$.
2. A condenser microphone is constructed by stretching a diaphragm across one end of a hollow cylinder with a diameter of 2 cm and length of 0.75 cm. The other end of the cylinder is open. Determine the ratio of the pressure at the diaphragm (considered to be a rigid plane) to the pressure at the open end as a function of frequency from 64 Hz to 2 kHz.
3. A pipe is 1.2 m long and has a radius of 6 cm. It is being driven at one end by a piston (negligible mass). The other open end of the pipe terminates in a flange.
   (a) Find the fundamental resonance of the system.
   (b) If the piston has a displacement of 1 cm when being driven at 500 Hz, what is the amount of acoustic power being transmitted by plane waves traveling to the open end of the pipe?
   (c) What is the acoustic power (in watts) being radiated out at the open end of the pipe?
4. Consider a pipe that is 1.2 m long with a radius of 6 cm. It is being driven at one end by a piston of 0.02 kg mass and the same radius as the pipe. The other end of the pipe terminates in an infinite baffle.
   (a) For 200 Hz frequency, find the mechanical impedance of the piston of the pipe, inclusive of the loading effect of the air inside the pipe.
   (b) For the above frequency, determine the amplitude of the force necessary to drive the piston with a displacement amplitude of 0.50 cm.

(c) What will be the amount of acoustic wattage that emanates out through the open end of the pipe?

5. A 70 dB (re 20 μPa) 1 kHz signal is incident normally to a boundary between the air and another medium having a characteristic impedance of 780 Pa s/m.
   (a) Determine the effective root-mean-square pressure amplitude of the reflected waves.
   (b) Determine the effective root-mean-square pressure of the transmitted waves.
   (c) How far from the boundary is the location where pressure amplitude in the pattern of standing waves equals the pressure amplitude of the incident wave?
6. The speed of sound in water is 1480 m/s. Consider a series of 2960-Hz plane waves in water normally incident to a concrete wall. It can be assumed that all of the sound energy is completely absorbed by the wall. The pattern of standing waves results in a peak pressure amplitude of 30 Pa at the wall and a pressure amplitude of 10 Pa at the nearest pressure node at a distance of 50 cm from the wall.
   (a) What is the ratio of the intensity of the reflected wave to that of the incident wave?
   (b) Find the specific acoustic impedance of the wall.
7. An acoustic signal consisting of 400-Hz plane wave is normally incident to an acoustical tile surface having a complex impedance of $1500 - i3000$.
   (a) Find the standing wave ratio in the resultant pattern of standing waves.
   (b) Determine the locations of the first four nodes.
8. A loudspeaker is fitted to one end of an air-filled pipe of 12 cm radius to generate plane waves inside the pipe. The far end of the pipe is closed off by a rigid cap. A 5-kHz signal is fed into the loudspeaker. The measured standing wave ratio of pressure at one location in the pipe is 7. At another location inside the pipe 50 cm downstream of the pipe, the measured standing wave ratio is 9. Derive an equation that includes these ratios and the distance between them which can be used to establish the absorption constant for the signal being propagated inside the pipe. To simplify matters, assume that absorption coefficient $\alpha \ll 1$.
9. Prove analytically (assuming $ka \ll 1$) that the frequencies of the resonance and those near antiresonance of the forced-open tube with damping correspond to the frequencies of maximum and minimum power dissipation, respectively.
10. Determine the lowest normal mode frequency of a fluid-filled cubic cavity (of length L to a side) that consists of five rigid walls and one pressure release side. Plot the pressure distribution associated with that node.
11. Given a rigid-walled rectangular room with dimensions 6.50 m $\times$ 5.65 m $\times$ 3 m, calculate the first 10 eigenfunctions. Assume the sound propagation speed $c = 344$ m/s.
12. A concrete water sluice measures measure 25 m wide and 8 m deep. It is completely filled with water. Find the cutoff frequency of the lower mode of propagation, assuming the concrete to be totally rigid.

8
Acoustic Analogs, Ducts, and Filters

8.1 Introduction

In this chapter we deal with lumped and distributed acoustic elements, applying electrical and mechanical analogs to acoustic behavior in order to treat different duct geometries, acoustic filters, and networks. The reflection and transmission of sound waves at piping interfaces, where the acoustic impedance changes, are analogous to the behavior of current waves in a transmission line at locations where the electrical impedance undergoes a change.

A simple mechanical system can often be converted into analogous electrical systems and solved in analogous terms. The motion of a fluid is compared to the behavior of current in an electrical circuit, with a pressure gradient between two points playing the role of voltage across the corresponding parts of the circuit. In terms of electricity the impedance is voltage divided by the current, which corresponds to the effect of lumped elements of inductance, capacitance, and resistance. In acoustics the *acoustic impedance* $\mathbf{Z}$ of a fluid acting with acoustic pressure $\mathbf{p}$ on a surface of area A is given by

$$\mathbf{Z} = \frac{\mathbf{p}}{\mathbf{U}} \tag{8.1}$$

where $\mathbf{U}$ represents the *volume velocity* of the fluid in the acoustic element of interest. $\mathbf{U}$ is not really a vector; it is a speed representative of a scalar quantity, unlike velocity which is a magnitude with an indicated direction. The *acoustic impedance* $\mathbf{Z}$ defined by Equation (8.1) is a complex quantity.

The *specific acoustic impedance* $\mathbf{z}$ is given by

$$\mathbf{z} = \frac{\mathbf{p}}{\mathbf{u}} \tag{8.2}$$

where $\mathbf{u}$ is the *particle* velocity, *not* the volume velocity. The specific acoustic impedance, which is useful for treating the transmission of acoustic waves from one medium to another, is a characteristic of the propagation medium and the type of propagating wave. The acoustic impedance, defined by Equation (8.1) as the

ratio of pressure to volume velocity, is applied to treat acoustic radiation from vibrating surfaces and the transmission of this radiation across lumped acoustic elements and through ducts and horns. These two impedances are related to each other by

$$\mathbf{Z} = \frac{\mathbf{z}}{A} \tag{8.3}$$

A is the area of a vibrating surface. If the vibrating surface is driven with a velocity $\mathbf{u}$, with a force $\mathbf{f}$ acting on the fluid, the radiation impedance $\mathbf{Z}_r$ is given by the ratio of the force to the speed:

$$\mathbf{Z}_r = \frac{\mathbf{f}}{\mathbf{u}} \tag{8.4}$$

This type of impedance constitutes a part of the mechanical impedance $\mathbf{Z}_m$ of the vibrating system. Radiation impedance relates to specific impedance at a surface as follows:

$$\mathbf{Z}_r = \frac{z_r}{A} \tag{8.5}$$

It is useful for dealing with coupling between acoustic waves and a driving surface or a driven load.

8.2 Lumped Acoustic Impedance

In the use of lumped parameters, the advantage is taken of the assumption that the signal wavelength is larger than all principle dimensions, which allows for further simplification. Each acoustic parameter may be time varying, but it becomes virtually independent of distance over the spatial extent of the device. When we consider *lumped* or *concentrated impedances* rather than distributed impedances, we define that impedance in a segment of the acoustic system as the (complex) ratio of the pressure difference $\mathbf{p}$ (which propels that segment) to the resultant volume velocity $\mathbf{U}$. The unit of acoustic impedance is Pa s/m^3, and it is often referred to as an *acoustic ohm*.

Example Problem 1

Consider a Helmholtz resonator described by differential Equation (7.62) which is repeated below:

$$m\frac{d^2\delta}{dt^2} + R_r\frac{d\delta}{dr} + S\delta = APe^{i\omega t} \tag{8.6}$$

and recast the system as a lumped acoustic impedance with an electric analog.

Solution

Divide Equation (8.6) by surface area A and utilize the fact that $\mathbf{U} = \dfrac{d\delta}{dt}A$, which will result in

$$\mathbf{Z} = \frac{\mathbf{p}}{\mathbf{U}} = R + i\left(M\omega - \frac{1}{\omega C}\right) \tag{8.7}$$

where

$$R = \frac{R_r}{A^2} = \frac{\rho_0 c k^2}{2\pi}, \qquad M \equiv \frac{m}{A^2} = \frac{\rho_0 L'}{A}, \qquad C \equiv \frac{A^2}{S} = \frac{V}{\rho_0 c^2}$$

and the stiffness is given by

$$S = \rho_0 c^2 \frac{A^2}{V}$$

according to Equation (7.59). We also assume here a flanged resonator. In electrical terms this constitutes a *RLC* series circuitry, where the inductance L is the electrical analog to M. This electrical analog to the Helmholtz resonator is illustrated in Figure 8.1. As a further aid to analytical treatment of lumped parameters, Figure 8.2 summarizes the fundamental analogous elements of acoustic, mechanical, and electrical systems. The inertance M in the acoustic system is represented by a "plug" of fluid that is sufficiently short so that all particles in the fluid can be depicted as moving in phase under the impetus of sound pressure. The compliance C of the acoustic system is represented by an enclosed volume incorporating a

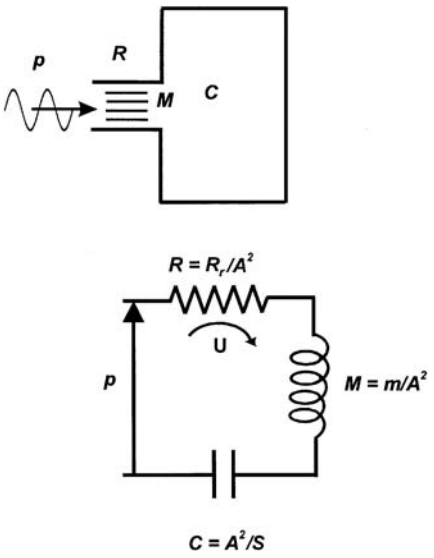


FIGURE 8.1. The electrical analog to the Helmholtz resonator.

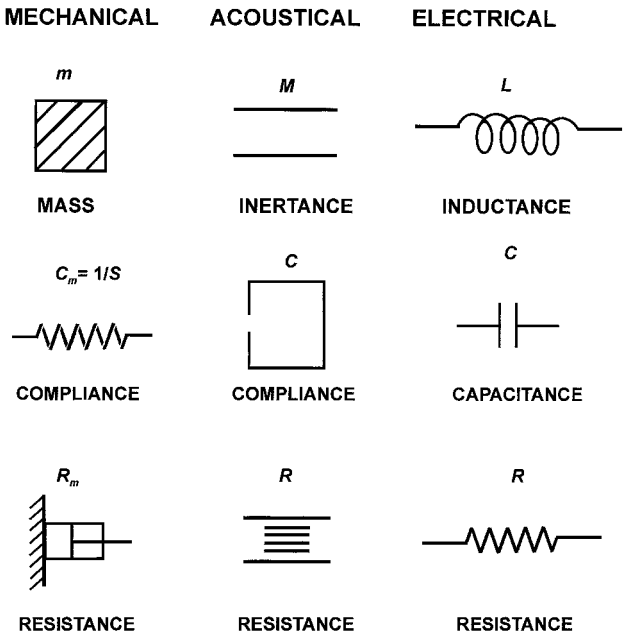


FIGURE 8.2. Fundamental mechanical, acoustical, and electrical analogs.

stiffness. A number of difference situations can contribute to resistance; we can represent acoustical resistance in the conventional manner by a narrow slit inside a pipe segment.

8.3 Distributed Acoustic Impedance

What if one or more of the principal dimensions of an acoustic system is of the same order of magnitude as a wavelength? In this case it may not be possible to treat the system as one possessing lumped parameters. The alternative is to analyze it as having distributed physical constants. Consider a very simple case of plane waves propagating through a pipe in the positive x-direction. The characteristic impedance of the pipe is given by the ratio of acoustic pressure to particle speed; and the acoustic impedance at any cross section A of the pipe is

$$\mathbf{Z} = \frac{\mathbf{p}}{\mathbf{U}} = \frac{1\,\mathbf{p}}{A\,\mathbf{u}} = \frac{\rho_0 c}{A} \tag{8.8}$$

The case of such propagation in a pipe is electrically equivalent to high-frequency currents traveling along a transmission line that has an inductance *per unit length* L_s and a capacitance *per unit length* C_s. The corresponding input electrical impedance is $\sqrt{L_s/C_s}$. In compliance with the electrical analogy, we may consider the fluid in the pipe to have a distributed inertance M_s per unit length and a distributed

compliance C_s per unit length. It also follows that the distribution of mass per unit length of the pipe can be represented by $m_s = \rho_0 A$. The acoustic inductance per unit length becomes $M_s = m_s/A^2 = \rho_0/A$.

We shall now find the mechanical stiffness per unit length. When the fluid is compressed adiabatically by a small linear displacement $\delta\ell$, then $p = \rho_0 c^2(\delta\ell/\ell)$, and the force providing this impetus is pA. Hence the stiffness becomes $S = pA/\delta\ell$; and the stiffness per unit length is $S_s = \rho_0 c^2 A$. The mechanical compliance C_m relates to acoustic compliance C by $C = A^2 C_m$, and on a per unit length basis, it follows that $C_s = A/(\rho_0 c^2)$. By analogy the acoustic impedance of the pipe is given by

$$Z = \sqrt{\frac{M_s}{C_s}} = \frac{\rho_0 c}{A}$$

which corresponds to Equation (8.8).

8.4 Waves in Pipes: Junctions and Branches

Consider a sound wave traveling in the positive x-direction, represented by

$$\mathbf{p}_i = \mathbf{a}e^{i(\omega t - kx)}, \tag{8.9}$$

is incident upon a point $x = 0$, where the acoustic impedance changes from $\rho_0 c/A$ to some complex value $\mathbf{Z}_0$. At this point a reflected wave

$$\mathbf{p}_r = \mathbf{b}e^{i(\omega t + kx)} \tag{8.10}$$

will be produced and travel in the negative x-direction. It is our task to find the power reflection and transmission coefficients for this point. The acoustic impedance at any point in the pipe is given by

$$\mathbf{Z} = \frac{\mathbf{p}_i + \mathbf{p}_r}{\mathbf{U}_i + \mathbf{U}_r} = \frac{\rho_0 c}{A}\frac{\mathbf{a}e^{-ikx} + \mathbf{b}e^{+ikx}}{\mathbf{a}e^{-ikx} - \mathbf{b}e^{+ikx}} \tag{8.11}$$

which, at $x = 0$, reduces to

$$\mathbf{Z}_0 = \frac{\rho_0 c}{A}\frac{\mathbf{a} + \mathbf{b}}{\mathbf{a} - \mathbf{b}} \tag{8.12}$$

Equation (8.12) can be rearranged to

$$\frac{\mathbf{b}}{\mathbf{a}} = \frac{\mathbf{Z}_0 - \dfrac{\rho_0 c}{A}}{\mathbf{Z}_0 + \dfrac{\rho_0 c}{A}} \tag{8.13}$$

The sound power reflection coefficient R_p, which yields the fraction of the incident power that is reflected, is given by

$$R_p = \left|\frac{\mathbf{b}}{\mathbf{a}}\right|^2 = \frac{\left(R_0 - \dfrac{\rho_0 c}{A}\right)^2 + X_0^2}{\left(R_0 + \dfrac{\rho_0 c}{A}\right)^2 + X_0^2} \tag{8.14}$$

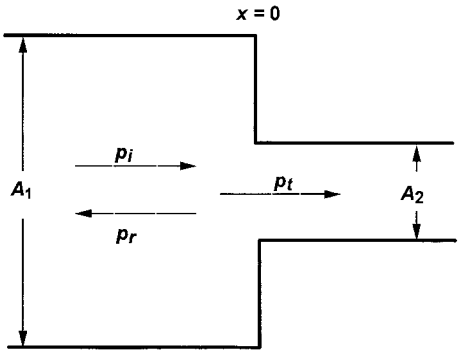


FIGURE 8.3. Transmission and reflection of a plane wave at the junction $x = 0$ between two pipes of different cross sections.

In Equation (8.14) we have set $\mathbf{Z}_0 = R_0 + iX_0$. The sound power transmission $T_p = 1 - R_p$ represents the portion of the incident sound power that travels past $x = 0$. We obtain

$$T_p = \frac{4R_0\rho_0 c/A}{\left(R_0 + \dfrac{\rho_0 c}{A}\right)^2 + X_0^2} \tag{8.15}$$

Example Problem 2

Apply the above equations to plane waves in a pipe of cross-section area A_1 that is mated to a pipe of cross-section area A_2, as shown in Figure 8.3. The second pipe is of infinite extent or the proper length so that no reflected wave is returned from its far terminus. Assume that the wavelength is considerably larger than the diameter of either pipe so that the region of complicated flow at the junction, where the wave adjusts from cross-sectional area to another, is considerably smaller than the wavelength itself.

Solution

Under the conditions stipulated above, the acoustic impedance seen by the incident wave at the junction is $\mathbf{Z}_0 = \rho_0 c/A_2$. Inserting this value of $\mathbf{Z}_0$ into Equations (8.14) and (8.15), we have

$$R_p = \left(\frac{A_1 - A_2}{A_1 + A_2}\right)^2 \quad \text{and} \quad T_p = \frac{4A_1A_2}{(A_1 + A_2)^2} \tag{8.16}$$

Note that if the above pipe is closed at $x = 0$, $A_2 = 0$, then $\mathbf{Z}_0$ becomes infinity, which results in a reflection coefficient of unity. On the other hand, if the pipe is open at $x = 0$, the impedance at the junction is not zero but corresponds to the impedance given by Equation (7.12) for an unflanged pipe.

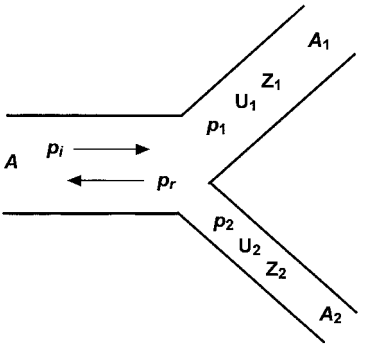


FIGURE 8.4. A three-way junction.

In Figure 8.4 we have the more complex case of a pipe that branches into two pipes, each with its own input impedance. Let the junction be located at the origin. The pressures produced by the waves in the three pipes at $x = 0$ are represented by

$$\begin{aligned} \mathbf{p}_i &= \mathbf{a}e^{i\omega t}, \quad \mathbf{p}_r = \mathbf{b}e^{i\omega t} \\ \mathbf{p}_1 &= \mathbf{Z}_1\mathbf{U}_1 e^{i\omega t}, \quad \mathbf{p}_2 = \mathbf{Z}_2\mathbf{U}_2 e^{i\omega t} \end{aligned} \tag{8.17}$$

Here **a** and **b** denote the amplitudes of the incident and reflected waves, respectively; and $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{U}_1$, $\mathbf{U}_2$ the input impedances and volume velocity complex amplitudes in branches 1 and 2. Again, under the assumption of a large wavelength so that the impact of branching remains confined to a small region at the juncture, we apply the condition of continuity of pressure so that

$$\mathbf{p}_i + \mathbf{p}_r = \mathbf{p}_1 = \mathbf{p}_2 \tag{8.18}$$

Likewise, the continuity of the volume velocity demands that

$$\mathbf{U}_i + \mathbf{U}_r = \mathbf{U}_1 + \mathbf{U}_2 \tag{8.19}$$

which is analogous to Kirchhoff's law of electric currents. Dividing Equation (8.19) by Equation (8.18) yields the impedance relationship

$$\frac{1}{\mathbf{Z}_0} = \frac{1}{\mathbf{Z}_1} + \frac{1}{\mathbf{Z}_2} \tag{8.20}$$

The reciprocal of an impedance, $\mathbf{Z}^{-1}$, is called the *admittance*. Equation (8.20) indicates that the combined admittance $1/\mathbf{Z}_0$ equals the sum of the admittances of the two branches 1 and 2.

Example Problem 3

An infinitely long pipe of cross-sectional area A has a branch at $x = 0$ that presents a given impedance of $\mathbf{Z}_g$. Find the appropriate power reflection and transmission coefficients.

Solution

In this case we consider this pipe to have two branches, one corresponding to the given impedance and the other to an infinite pipe which does not provide any reflections but does present an impedance of $\rho_0 c/A$. Applying Equation (8.20) we obtain

$$\frac{\mathbf{b}}{\mathbf{a}} = \frac{\dfrac{\rho_0 c}{2A}}{\dfrac{\rho_0 c}{2A} + \mathbf{Z}_g} \tag{8.21}$$

The ratio of pressure amplitude $\mathbf{a}_t$ of the wave transmitted beyond $x = 0$ along the infinite pipe to the pressure amplitude of the incident wave is found by inserting Equation (8.21) into Equation (8.18), yielding

$$\frac{\mathbf{a}_t}{\mathbf{a}} = \frac{Z_g}{Z_g + \dfrac{\rho_0 c}{2A}} \tag{8.22}$$

We resolve the acoustic impedance $\mathbf{Z}_g$ of the branch into real and imaginary components, i.e., $\mathbf{Z}_g = R_g + iX_g$. The reflection and transmission coefficients become

$$R_p = \left|\frac{\mathbf{b}}{\mathbf{a}}\right|^2 \frac{\left(\dfrac{\rho_0 c}{2A}\right)^2}{\left(\dfrac{\rho_0 c}{2A} + R_g\right)^2 + X_g^2} \tag{8.23}$$

$$T_p = \left|\frac{\mathbf{a}_t}{\mathbf{a}}\right|^2 = \frac{R_g^2 + X_g^2}{\left(\dfrac{\rho_0 c}{2A} + R_g\right)^2 + X_g^2} \tag{8.24}$$

The portion T_{pg} of the power that is transmitted into the branch is $T_{pg} = 1 - R_p - T_p$, or

$$T_{pg} = \frac{R_g \rho_0 c/A}{\left(\dfrac{\rho_0 c}{2A} + R_g\right)^2 + X_g^2} \tag{8.25}$$

If R_g has a positive finite value, some acoustic energy is dissipated at the branch and some is transmitted beyond the junction, no matter what the value of X_g is. When either R_g or X_g greatly exceeds $\rho_0 c/A$, all of the incident power is transmitted past the branch. At the other extreme, when $R_g X_g = \infty$, which corresponds to no branch, the power transmission is unity.

8.5 Acoustic Filters

Advantage can be taken of the fact that a side branch can attenuate sound energy transmitted in a pipe. The input impedance of the side branch determines whether the system can behave as a low-pass, high-pass, or band-pass filter. We shall now consider each of these filters.

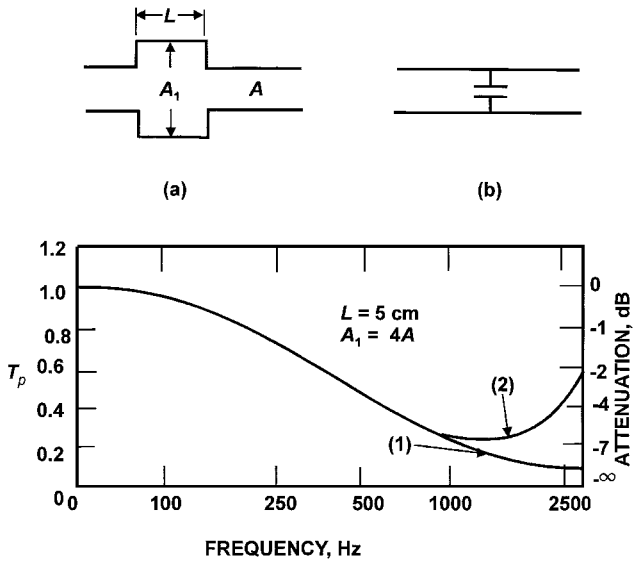


FIGURE 8.5. (a) A simple low-pass acoustic filter, (b) its analogous electrical filter, and (c) the corresponding power–transmission curves for the acoustic filter of (a).

1. Low-Pass Filters. Figure 8.5 illustrates the construction of a simple low-pass filter, essentially consisting of an enlarged segment of a pipe of cross-sectional area A_1 and length L in a pipe of cross section A. At sufficiently low frequencies ($kL \ll 1$), this filter may be viewed as a side branch with acoustic compliance $C = V/(\rho_0 c^2)$, where $V = A_1 L$ represents the volume of the expansion chamber. The acoustic impedance of this type of branch is pure reactance, hence

$$R_g = 0, \qquad X_g = -\frac{1}{C\omega} = \frac{\rho_0 c^2}{A_1 L \omega}$$

Inserting the above into the expression for the transmission coefficient, Equation (8.23) yields

$$T_p = \frac{1}{\left(\dfrac{A_1}{2A} kL\right)^2 + 1} \tag{8.26}$$

Equation (8.26), plotted in Figure 8.5, indicates that as the frequency approaches zero, the transmission coefficient approaches unity (100% transmission), but as the frequency becomes higher, this coefficient tends toward zero. Curve 1 is for an expansion chamber 5 cm in length and a cross-sectional ratio of $A_1/A = 4$. However, Equation (8.26) does not apply to $kL > 1$.

In order to treat the case of the above-mentioned acoustic filter for $kL > 1$, the incident, reflected and transmitted waves in the three regions of the pipe must be related to one another by the fact that the continuity of pressure and volume velocity

must occur at the two junctions of the pipe. This results in a power-transmission coefficient expressed as follows:

$$T_p = \frac{4}{4\cos^2 kL + \left(\frac{A_1}{A} + \frac{A}{A_1}\right)^2 \sin^2 kL} \tag{8.27}$$

In Figure 8.5, Curve 2 constitutes the plot of Equation (8.27) for the same filter system used to obtain Curve 1. At low values of frequencies, i.e., $kL \ll 1$, the two curves basically coincide. Equation (8.27), which is physically more valid, exhibits a minimum transmission coefficient of

$$T_p(\text{minimum}) = \left(\frac{2A_1A}{S_1^2 + S^2}\right)^2 \qquad (kL = \pi/2) \tag{8.28}$$

for the case where the length of the filter segment equals a quarter wavelength. Beyond this saddle point, T_p gradually rises with increasing frequency until it reaches 1.0 (100%) at $kL = \pi$. At even higher frequencies the transmission coefficient vacillates through a series of maxima and minima until ka (a is the radius of the through pipe) becomes somewhat larger than unity. From this point on, the transmission coefficient remains at unity. This trait of the transmission coefficient reaching a plateau of unity is also shared by high-pass and band-pass filters.

Equation (8.28) may also be used to treat the constriction-type low-pass filter illustrated in Figure 8.6, since it does not matter if A_1 is larger or smaller than A. The decrease in the area can be viewed as introducing an inertance in series with the pipe, but the validity of this analog also extends over a limited range of frequencies, as with the case of the expanded-area low-pass filter of Figure 8.5.

In the real world of filter design (of mufflers, sound-absorption plenum chambers for ventilating systems, etc.) the filter cross section cannot be radically different in value from the cross-section area of the pipe. As is demonstrated in the curves of Figure 8.5, a limited range of frequencies exists for the practical operation of the filters.

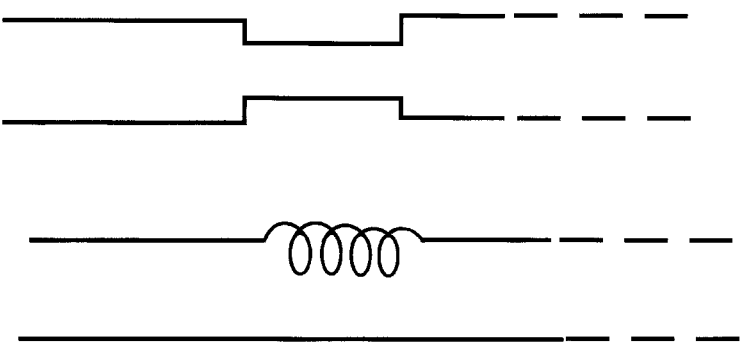


FIGURE 8.6. A pipe with a constriction and its electrical analog.

2. High-Pass Filters. A high-pass filter can be constructed by attaching a short length of pipe as a branch to a main pipe, in effect creating an orificed Helmholtz resonator. Both the radius a and length L of this appendage are small compared to a wavelength. Equations (7.60b) and (7.57) for the unflanged resonator apply to the branch impedance of this orifice given by

$$\mathbf{Z}_g = \frac{k\rho_0 c^2}{4\pi} + i\frac{\rho_0 \omega L'}{\pi a^2} \tag{8.29}$$

where $L' = L + 1.5a$. The first term in the right-hand side of Equation (8.29) stems from the radiation of sound through the orifice into the external medium, and the second is attributable to the inertance of the gas in the orifice. The ratio of the branch acoustic resistance to its acoustic reactance is $R_g/X_g = 1/4ka^2/L'$. Because it has been assumed that $ka \ll 1$, we can neglect the acoustic resistance in comparison with the acoustic reactance in the use of Equation (8.24) to find the power-transmission coefficient T_p, with the result:

$$T_p = \frac{1}{\left(\dfrac{\pi a^2}{2AkL'}\right)^2 + 1} \tag{8.30}$$

We observe that this transmission coefficient is virtually zero for low frequencies and it rises to nearly unity at higher frequencies, as shown in Figure 8.7. The halfway point at which the transmission coefficient is 50% is reached when

$$k = \frac{\pi a^2}{2AL'}$$

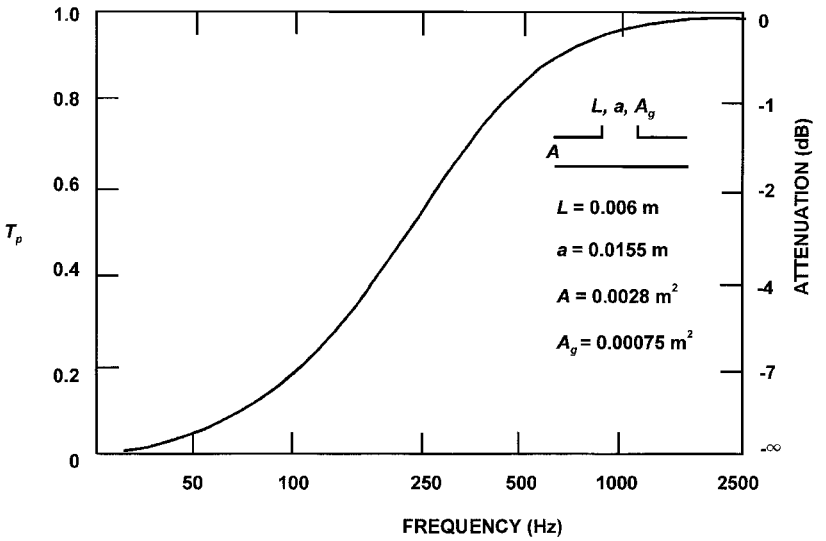


FIGURE 8.7. Attenuation produced by an orifice-type branch, resulting in high-pass transmission.

The presence of one orifice turns a pipe into a high-pass filter. If the radius of the orifice is increased, the attenuation of the low-frequency components also increases. If a pipe contains several orifices separated by only a fraction of a wavelength, these orifices can be treated as a group acting with their equivalent parallel impedance. But if the distances between the orifices constitute an appreciable portion of the wavelength, the system becomes analogous to an electrical network of filters or to a transmission line which has a number of impedances shunted across it, spaced apart at wide intervals. Waves reflected from these different orifices are then out of phase with respect to each other, and Equation (8.30) no longer remains valid. Electrical filter theory must then be utilized to compute the transmission coefficient. As a rule, a number of orifices strategically placed apart can attenuate at low frequencies more effectively than a single orifice of equal total area.

The sound power transmission coefficient T_{pg} into a single orifice is approximated by

$$T_{pg}=\frac{2k^2A}{\pi\left[\left(\frac{2AkL'}{\pi a^2}\right)^2+1\right]} \tag{8.31}$$

The filtering action of an orifice is principally that of the reflection of energy back toward the source, not so much the loss of acoustic energy out of the pipe through the orifice into the ambient medium.

A common example of the application of orifices is the control of the behavior of a wind instrument such as a flute or a saxophone. When such an instrument is played in its fundamental register, all or nearly all of the orifices some distance beyond the mouthpiece are kept open by the player. The diameters of these orifices nearly equal the bore of the tube, essentially shortening the effective length of the instrument. The acoustic energy reflected from the first open orifice generates a pattern of standing waves between the first open orifice and the mouthpiece. The flute behaves like an open pipe, with the wavelength approximately equal to twice the distance between the first orifice and the opening of the mouthpiece. A clarinet or a saxophone contains a vibrating reed at the mouthpiece, which approximates the conditions of the closed end of a tube. In this case the wavelength will equal nearly four times the distance from the reed to the first open orifice.

Both the reed-type (clarinet, saxophone, coronet, etc.) and tubular instruments (flute, recorder, piccolo, etc.) contain a number of harmonics, those of the reed instruments being primarily odd harmonics (characteristic of closed pipes). When higher notes are played on either type of instruments, the fingering of these holes become more complicated, with some orifices beyond the first orifice closed and some others opened. The fingering of these orifices control the standing waves patterns which correspond to specific notes.

3. Band-Pass Filters. A side branch in the form of a long pipe rigidly capped at its far end or a fully enclosed Helmholtz resonator (shown in Figure 8.8) contains

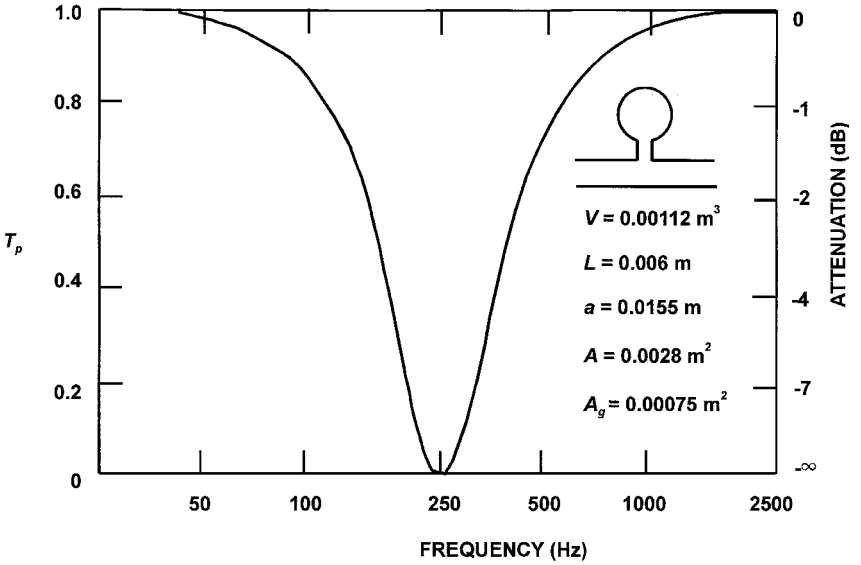


FIGURE 8.8. The effect of a Helmholtz resonator branch, resulting in band-pass transmission.

both inertance and compliance, so it will behave as a band-pass filter. Apart from almost negligible viscosity losses, no net dissipation of acoustic energy occurs from the pipe into the resonator. All energy absorbed by the resonator during some phase of the acoustic cycle is returned to pipe during other phases of the cycle so $R_g = 0$. Denoting the opening area by $A_g = \pi a^2$, the neck length by L and the volume of the resonator by V, the branch reactance X_g is expressed as

$$X_g = \rho_0 \left(\frac{\omega L'}{A_g} - \frac{c^2}{\omega V} \right)$$

which is then inserted into Equation (8.24) to yield the following transmission coefficient:

$$T_p = \left[1 + \frac{c^2}{4A^2 \left(\frac{\omega L'}{A_g} - \frac{c^2}{\omega V} \right)} \right]^{-1} \tag{8.32}$$

The resonant frequency occurs when the transmission coefficient becomes zero, i.e.,

$$\omega_0 = c \sqrt{\frac{A_g}{L'V}}$$

which corresponds to the resonant frequency of a Helmholtz resonator. When this frequency occurs, large volume velocities prevail in the neck of the resonator, and

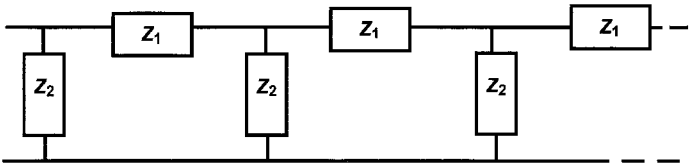


FIGURE 8.9. A ladder-type network used as a filter.

all acoustic energy that transmits into the resonator returns to the main pipe in such a manner as to be reflected back from the source. The plot of the power-transmission coefficient in Figure 8.8 is fairly typical for a bandpass resonator.

Equation (8.32) serves well for a resonator that has a relatively large neck radius. Narrower and longer constrictions will cause the transmission coefficient to deviate from the prediction of Equation (8.32), unless consideration is taken of the viscous dissipation that manifest itself more with such geometries.

4. Filter Networks. The design procedure for acoustic networks, which incorporates resonators, orifices, and divergence and convergence of pipe areas, is rendered easier by analogy with the deployment of electronic filters. The sharpness of cutoff of an electrical filter system, for example, can be enhanced by using the ladder-type network of Figure 8.9. This network is constructed by using the reactances of one type of impedance $\mathbf{Z}_1$ in series with the line and the reactance of another type of impedance $\mathbf{Z}_2$ shunted across the line. The standard theory of wave filters states that a nondissipative repeating structure such as that illustrated in Figure 8.9 causes a marked attenuation of all frequencies except those for which the ratio $\mathbf{Z}_1/\mathbf{Z}_2$ meets the condition

$$0 > \mathbf{Z}_1/\mathbf{Z}_2 > -4 \tag{8.33}$$

Several examples of acoustic ladder-type filters are displayed with the electrical analogs in Figure 8.10. The condition of Equation (8.33) provides the following cutoff frequencies:

$$f = \frac{1}{4\pi\sqrt{MC}}$$

for the high-pass filter as shown in Figure 8.10(a), and

$$f = \frac{1}{\pi\sqrt{MC}}$$

for the low-pass filter as shown in Figure 8.10(b). The behavior projected by the filters as shown in Figure 8.10 applies only to wavelengths that are large compared to the dimensions of the filter. At higher frequencies, deviation of the behavior predicted by electrical analogs becomes more prominent, because the filter begins

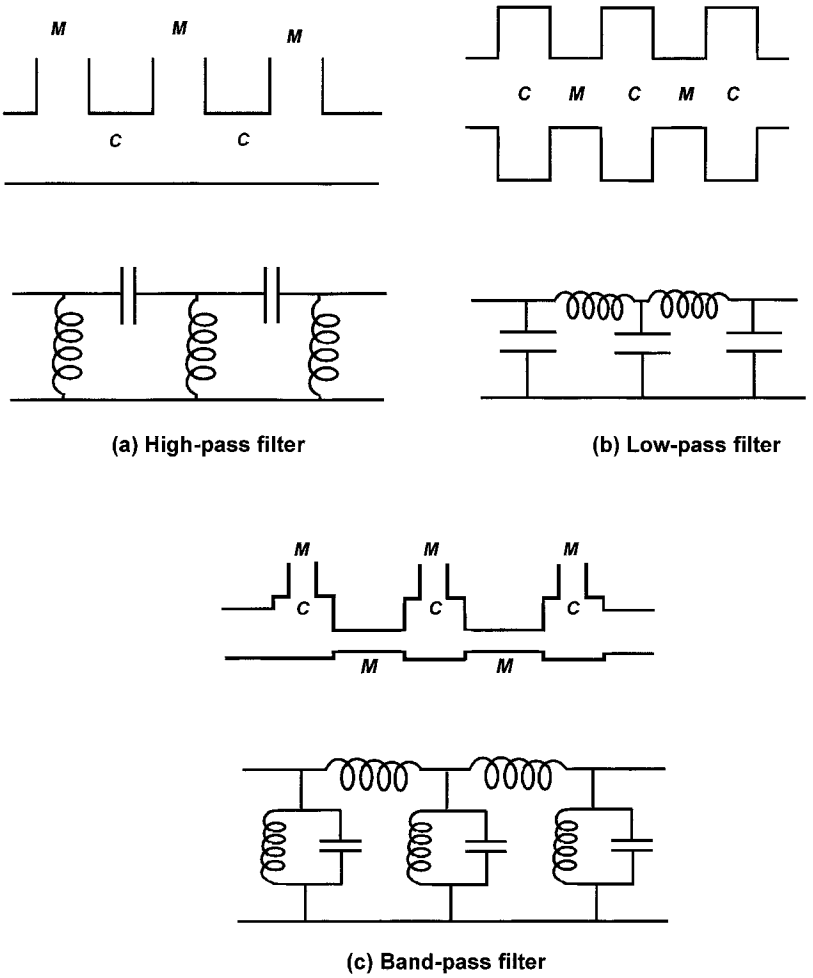


FIGURE 8.10. Examples of ladder-type acoustic filters.

to manifest the properties of distributed parameters rather than those of lumped parameters.

8.6 Ducted Source Systems—Acoustic Modeling

A ducted source system is one in which the source is the active component and the load is the path, consisting of elements such as mufflers, ducts, and end terminators. Many mechanical systems such as engines and mufflers and air-moving devices (flow ducts, fluid pumping) are very common examples of ducted sources.

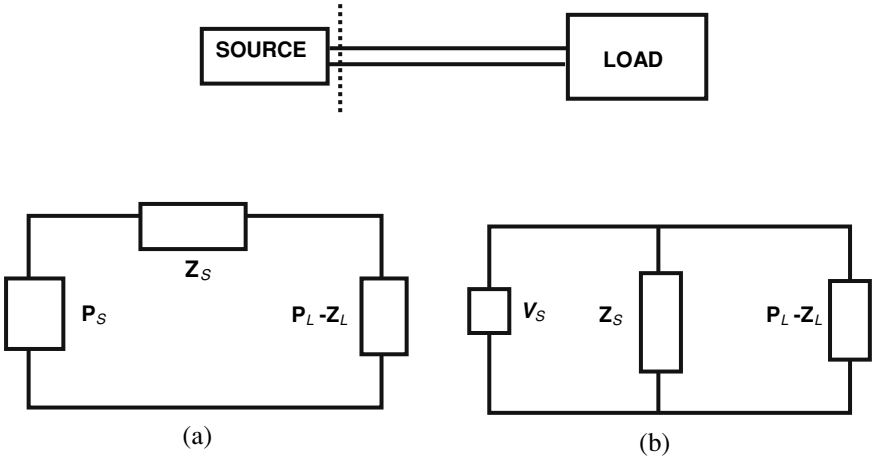


FIGURE 8.11. Electrical analog of a ducted-source-load system: (a) pressure source; and (b) volume velocity source.

The source-load interactions generally determine the acoustic performance of the system. This section provides a brief overview of characterization of acoustic performance of ducted sources. Acoustic performance of a system incorporating a muffler as a path element is usually described in terms of *insertion loss* and *radiated sound pressure.*

Figure 8.11 illustrates a basic duct system. Let $\mathbf{P}_S$ and $\mathbf{V}_S$ denote the source pressure and volume velocity, respectively. $\mathbf{P}_L$ and $\mathbf{V}_L$ represent the pressure- and volume-velocity response of the source-load system, respectively. $\mathbf{Z}_S$ and $\mathbf{Z}_L$ are the complex source and load impedances, respectively. The equations for the course-load system in terms of pressure and velocity complex source representations are given by

$$\mathbf{P}_L = \frac{\mathbf{P}_S \mathbf{Z}_L}{\mathbf{Z}_S + \mathbf{Z}_L}$$

and

$$\mathbf{V}_L = \frac{\mathbf{V}_S \mathbf{Z}_S}{\mathbf{Z}_S + \mathbf{Z}_L}$$

The source at one end of a duct system constitutes a boundary condition. The source is generally more difficult to characterize than the termination because of the dynamic nature of the source (Prasad, 1991).

Prior to the development of direct and indirect methods for measuring source impedance, characteristic or infinite impedances were assumed for the source, but these assumptions do not normally yield valid values of impedances. Both direct and indirect methods for the measurement of source impedance are based on frequency-domain analysis. Analytical modeling efforts have been carried our

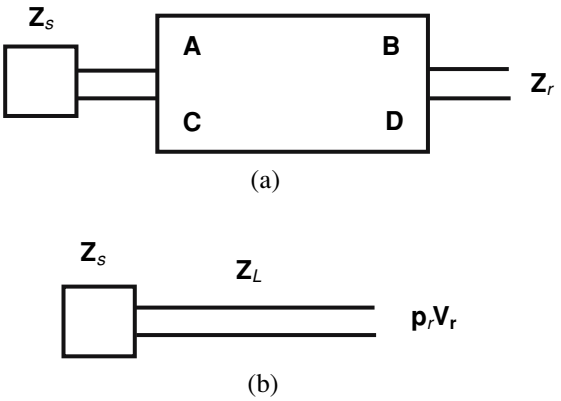


FIGURE 8.12. Duct system models: (a) source-path-termination model and (b) source-load model.

principally in the time domain based on the method of characteristics. A number of studies have been conducted on the basis of modeling of the geometry of sources (Prasad, 1991).

A duct system can be modeled as a source-path-termination model with a source load, as shown in Figure 8.12. These models are interrelated when the path-termination is treated as a load. This type of model is commonly used, say, for engine-exhaust-pipe-tailpipe-radiation system. Referring to Figure 8.13, the three most commonly used descriptors, namely, *insertion loss* (*IL*), *transmission loss* (*TL*), and *noise reduction* (*NR*), are given by the following three equations:

$$\text{IL} = 20\log_{10}\left|\frac{\mathbf{A}\mathbf{Z}_r + \mathbf{B} + \mathbf{C}\mathbf{Z}_S\mathbf{Z}_r + \mathbf{D}\mathbf{Z}_s}{\mathbf{A}'\mathbf{Z}_r + \mathbf{B}' + \mathbf{C}'\mathbf{Z}_S\mathbf{Z}_r + \mathbf{D}'\mathbf{Z}_S}\right| \text{dB} \tag{8.34}$$

$$\text{TL} = 20\log_{10}\left|\frac{1}{2}\left(\mathbf{A} + \frac{\mathbf{B}S}{\rho c} + \frac{\mathbf{C}\rho c}{S} + \mathbf{D}\right)\right| \text{dB} \tag{8.35}$$

$$\text{NR} = 20\log_{10}\left|\left(\mathbf{A} + \frac{\mathbf{B}}{\mathbf{Z}_r}\right)\right| \text{dB} \tag{8.36}$$

Here $\mathbf{Z}_S$ and $\mathbf{Z}_r$ are the source and radiation impedances, respectively, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are the four-pole parameters of the muffler including its upstream and downstream ducts. The insertion loss IL is the most useful of the three descriptors given in Equations (8.34)–(8.36). As the terminology implies, IL describes the reduction in the acoustic output when a muffler is inserted into an otherwise unattenuated system. Radiated sound pressure level L_p is also quite useful as it gives the system output which can be used to determine IL. The other two descriptors, namely, TL and NR, do not require knowledge of the acoustic characteristics of the source. By their very definition in Equations (8.35) and (8.36), respectively, both TL and NR are independent of the influence of the source-load interaction.

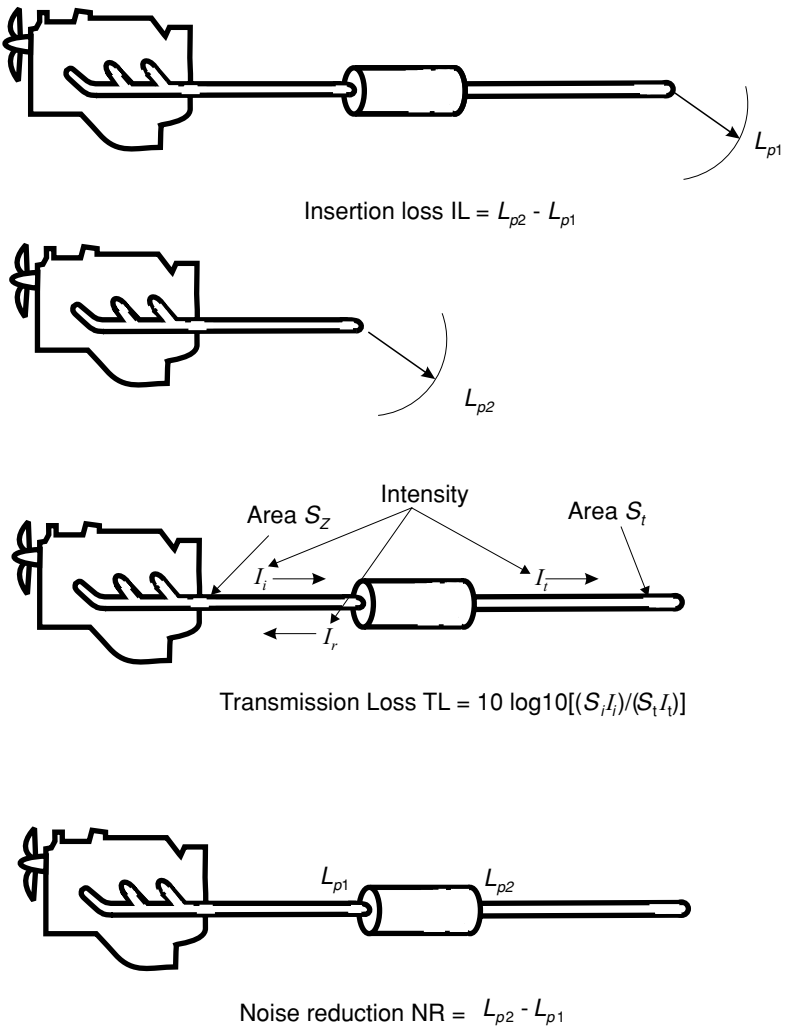


FIGURE 8.13. Muffler system performance descriptors.

While IL is fairly easy to measure, it is extremely difficult to predict because of its dependence on the source impedance $\mathbf{Z}_S$ (Davis, 1957; Munjai, 1987; Prasad and Crocker, 1998).

References

Beranek, Leo J. 1886. *Acoustics*. New York: The American Institute of Physics, Chapter 3.
Davis, Jr., D. D. 1957. Acoustic filters and mufflers. In: Harris, Cyril M., ed. *Handbook of Noise Control*. New York: McGraw-Hill, Chapter 21.

Kinsler, Lawrence E. Frey, Austin R., Coppens, Alan B., and Saunders, James V. 1982. *Fundamentals of Acoustics*, 3rd ed. New York: John Wiley & Sons, Chapters 9 and 10.

Munjai, M. L. 1987. *Acoustics of Ducts and Mufflers*. New York: John Wiley & Sons.

Prasad, M. G. 1991. Characterization of acoustical sources in duct systems—progress and future trends. *Proceedings of Noise-Con, Tarrytown, New York*. Ames, Iowa: Institute of Noise Control Engineering.

Prasad, M. G. and Crocker, M. J. 1998. Acoustic modeling (ducted source systems). In: Crocker, Malcolm J., ed. *Handbook of Acoustics*. New York: John Wiley & Sons, Chapter 14.

Problems for Chapter 8

In the following problems, consider the fluid medium to be air at the standard conditions of 1 atmosphere and 20°C, unless otherwise stated.

1. It is desired to make a Helmholtz resonator out of a sphere with a diameter 12 cm for 350 Hz.
   (a) What should be the diameter of the hole in the sphere?
   (b) What should be the pressure amplitude of the incident acoustic plane wave at 350 Hz if it is to produce an excess internal pressure of 25 μbar?
   (c) If the hole is doubled in area, what will be the resonant frequency?
   (d) What will be the resonance frequency if two independent separate holes, each of the diameter found in part (a) are drilled in the sphere?
2. Consider a loudspeaker system that is rigid-walled and back enclosed. Its inside dimensions are 40 cm × 55 cm × 44 cm. The front panel of cabinet is 4 cm thick and it has a 22-cm diameter hole cut out to accommodate a loudspeaker.
   (a) What is the fundamental frequency of this cabinet considered as a Helmholtz resonator?
   (b) A direct-radiating loudspeaker having a cone of 22 cm diameter and 0.008 kg mass and a suspension system of 1100 N/m stiffness is mounted in the cabinet. Find the resonance frequency of the cone. It may be assumed that the effective mass of the system is that of the cone and that of the air moving in the opening of the cabinet. The effective stiffness is the sum of the stiffness of the cone and that of the cabinet.
   (c) What would be the resonant frequency of the loudspeaker if it were not mounted in the cabinet and it has no air loading?
   (d) Find the acoustic power emitted if the cone is driven with an amplitude of 2.5 mm at the frequency established in part (b).
   (e) Under the conditions same as that of (d) what is the amplitude of the force acting on one of the 55 cm × 44 cm panels?
3. A rectangular room has internal dimensions of 3.0 m × 5.0 m × 2.6 m and walls of 12 cm thickness. A door that opens into the room has dimensions of 2.2 m × 0.8 m. Assume that the inertance of the door opening is equivalent of a circular opening of equal area.

(a) Find the resonance frequency of the room considered as a Helmholtz resonator.
(b) Find the acoustic compliance of the room.
(c) Find the inertance of the door opening.
(d) What is the acoustic impedance presented to the sound source at 30 Hz inside the room? Consider only the compliance of the room and the inertance of the door opening.

4. Pipe 1 having cross-section A_1 connects to pipe 2 with cross section A_2.
(a) Obtain an expression for the ratio of intensity of the waves transmitted into the second pipe to that of the incident waves.
(b) Under what conditions will the transmitted energy exceed the incident energy?
(c) Develop a general expression for the standing wave ratio (SWR) produced in pipe 1 in terms of relative areas A_1 and A_2.
5. Consider two pipes that are connected to each other but separated by a thin rubber diaphragm. Pipe 1 is filled with a fluid with characteristic impedance $\rho_1 c_1$ and pipe 2 contains a fluid with characteristic impedance $\rho_2 c_2$. A plane wave travels in pipe 1 in the positive x direction toward pipe 2.
(a) Obtain an expression for the power-transmission ratio from pipe 1 into pipe 2.
(b) Under what conditions does 100% power transmission occur?
6. An infinitely long pipe of cross-sectional area A has a side branch, also infinitely long, of cross-sectional area A_b. The main pipe is transmitting plane waves with frequencies such that their wavelengths are much larger than the diameter of either pipe.
(a) Develop an equation for the transmission coefficient in the main pipe.
(b) Do the same thing for the branch pipe.
(c) Let the area of the main pipe be twice as much as that of the branch pipe. Obtain numerical values for the transmission coefficient into each pipe. Do the sum of the two coefficients equal unity? If not, where did the remaining power go?
7. A ventilating duct in a basement has a square cross-sectional area, 35 cm to the side. In order to quiet the duct in part, a Helmholtz resonator-type filter is constructed about the duct by drilling a hole of 9 cm radius in one wall of the duct, leading into a surrounding closed chamber of volume V.
(a) What is the volume V necessary to most effectively filter sounds at a frequency of 30 Hz?
(b) What will be the sound power transmission coefficient of the filter at 45 Hz and 60 Hz?
8. Demonstrate that the radius r of the hole drilled into a pipe of radius r_0 as to result in a 50% sound power transmission coefficient at a frequency f is given by

$$r = \frac{64}{3}\frac{r_0^2 f}{c}$$

9. A 400-Hz plane wave of 0.2 W power propagates down an infinitely long pipe of 5 cm diameter. Find the power reflected, the power transmitted along the pipe, and the power transmitted out through a simple orifice of 1.2 cm diameter.
10. A 5-cm pipe carries water. It is planned to filter out plane waves traveling in the water by using sections of pipes 10 cm in diameter to serve as expansion-chamber type of filters.
    (a) Find the minimum length of the filter section that will most effectively filter out a sound of 1 kHz.
    (b) If it is desired to have a filtering action lessen the level of intensity by 35 dB, how many sections must be used? Disregard the effects of any interaction between the individual filter sections.
11. Find the transmission coefficient of a low-pass filter that consists of a circular pipe of 3.0 cm radius mated with a circular expansion chamber that is 12.0 cm long. Assume that wavelength λ is 30 cm (recall that $k = 2\pi/\lambda$).

9
Sound-Measuring Instrumentation

9.1 Introduction

Acoustic measurements constitute an essential step in order to establish the status of an acoustic environment and to develop a systematic approach toward modifying the environment and to set up criteria for improvements. Other more recently developed methodologies of acoustical measurements entail the studies of material properties and medical diagnoses. A wide variety of instrumentation exists, ranging from a simple sound pressure level (SPL) meter to real-time spectral analyzers interfaced with sophisticated computer systems. Instruments may be portable for field use; and recorded field data can be later evaluated in more elaborate systems. New instruments are continually being developed, and with new advances in digital technology arriving on the scene on a daily basis, it is not inconceivable that more versatile and "user-friendly" devices will become available at even lower prices.

This chapter deals principally with instruments intended for the audio range of frequencies. Much of the salient aspects of instrumentation in this chapter, such as the principal performance requirements and methodologies for evaluating data, also apply to the specialized underwater instruments and ultrasonic sensing instruments which are described in Chapters 15 and 16, respectively.

9.2 Principal Characteristics of Acoustical Instruments

The most important performance characteristics of acoustical instruments are the frequency response, dynamic range, crest factor capability, and response time. It is also desirable that a measuring device or system has a negligible (or at least predictable) effect or influence on the variable being measured.

Frequency response refers to the range of frequencies that an instrument is capable of correctly measuring the relative amplitudes of the subject variable within acceptable limits of accuracy. Measurement accuracy depends on the instrumentation type, the quality of design and manufacture, and calibration. A typical limit for the flatness of the response for microphones may be ± 2 dB or better; in contrast,

the response curve of a high-quality loudspeaker may deviate ±5 dB over its rated frequency range.

Dynamic range defines the range of signal amplitudes that an instrument is capable of handling in the process of responding and measuring accurately. A sound level meter (SLM), for example, which can measure a minimum of 10 dB and a maximum of 150 dB, covers a dynamic range of 140 dB.

Crest factor capability denotes an instrument's capacity to measure and distinguish instantaneous peaks. Crest factor itself is the ratio of the instantaneous peak sound pressure to the root-mean-square sound pressure.

Response time refers to the rapidity with which a measuring instrument responds to abrupt changes in signals. An oscilloscope display of a square-wave response will result in a trapezoidal display if the relayed signals came from a loudspeaker that requires a longer reaction time to respond to a square-wave pulse.

9.3 Microphones

Microphones serve as transducers by receiving and sensing pressure fluctuations and converting them into electrical signals that are relayed to other electronic components. The quality of a microphone determines the accuracy of a measurement system. A top-caliber measuring (or a sound reproduction) system can be undermined by the use of a microphone that is of a lesser caliber. Four principal types of microphones are used in measurement procedures, namely, dynamic, ceramic, electret, and condenser microphones.

D*ynamic microphones* produce an electric signal through the motion of a coil connected to a diaphragm in a magnetic field. They are in effect loudspeakers working in reverse, accepting an acoustic signal, and converting it into an electric pulsation rather than the other way around. Because of their low impedance, they can be used in applications entailing the use of long cables connected to auxiliary instrumentation. But they cannot be used in the vicinity of devices that emit magnetic fields (e.g., transformers and motors). Also they generally have longer response times, more limited frequency response, but can be constructed to withstand rough handling and high humidity.

Ceramic microphones consist of a piezoelectric (ceramic) element attached to the rear of a diaphragm. Sound pressure causes the diaphragm to vibrate, exerting a varying force on the ceramic element. The piezoelectric crystal generates an electric signal from the oscillating strains imparted by the diaphragm. These microphones are rugged, relatively inexpensive and reliable, have high capacitance and good dynamic range, and do not require a polarizing voltage that electret and condenser microphones need. But the high-frequency response may be lacking, and the operating temperature range may be limited.

The *electret* (or *electret-condenser*) *microphone*, illustrated in Figure 9.1, consists of a self-polarized metal-coated plastic diaphragm. Sound pressure causes the diaphragm to move relative to a back plate, varying the capacitance, and producing a signal. While relatively impervious to high humidity, this type of microphone

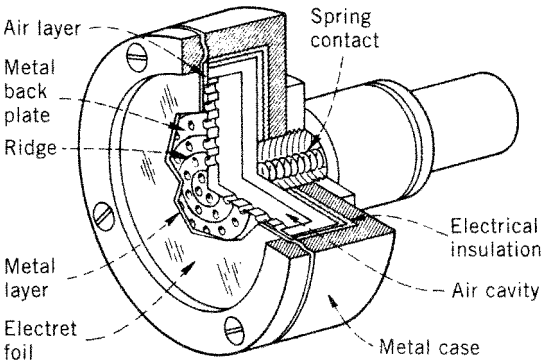


FIGURE 9.1. A cut-away view of an electret microphone showing the principal components. A thin electret polymer foil is suspended over a perforated backplate.

does not equal the frequency response of a condenser microphone and are not likely to withstand temperature extremes very well. An alternate form of the electret microphone is created by depositing the electret film onto the stationary plate and using the thin metal foil as a moving diaphragm.

Condenser microphones are capacitance-varying devices; each one consisting of two electrically charged plates with an air gap between them. The thin diaphragm serves as the plate that deflects under the influence of changes in sound pressure, causing the gap to vary. The resulting change in the capacitance is converted into an electric signal. Figure 9.2 illustrates the principal components of a condenser microphone. A capillary tube behind the plates provides the air bleed to provide air pressure equalization with the ambient. These devices have relatively low capacitance, require a polarizing voltage supplied by a preamplifier which also provides

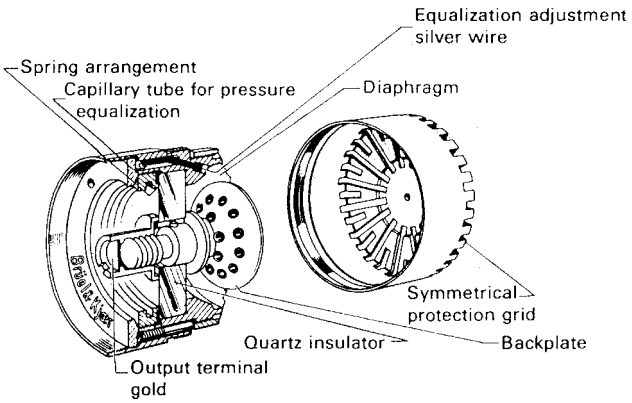


FIGURE 9.2. Principal elements of a condenser-microphone cartridge. (Courtesy of Brüel & Kjær.)

the appropriate impedance for connecting the microphone to a measuring system. Because of their long-term stability, superb high-frequency response, insensitivity to vibration excitation, condenser microphones are generally preferred for precision measurements. They function well in extreme temperature and pressure environments, but they are affected adversely by high humidity (which causes electrical leakage) and their diaphragms are fragile.

9.4 Sensitivity

In addition to frequency response, sensitivity constitutes one of the principal characteristics of a transducer. In general there is a trade-off between sensitivity and frequency response. Small microphones tend to have lower sensitivity but operate at both low and higher frequencies, whereas large microphones possess high sensitivity but are useful mainly at lower frequencies. Microphone diameters are typically 1, 2, and 4 cm (1/4, 1/2, and 1 in.) in diameter. The frequency response of a 4-cm condenser microphone is virtually flat to 20 kHz, while that of a 1-cm microphone is fairly flat to approximately 100 kHz. We define microphone sensitivity S by

$$S = \frac{\text{electrical output}}{\text{mechanical input}}$$

The microphone sensitivity level L_s, also called simply "sensitivity," is defined as

$$L_s = 10 \log \left(\frac{E_{\text{out}} / p}{E_{\text{re}}} \right)^2 \text{dBV/}\mu\text{bar} = 20 \log \left(E_{\text{out}} / p \right) \text{dBV/}\mu\text{bar} \qquad (9.1)$$

where

E_{out} = output voltage into the instrument
E_{re} = reference voltage (1 V for an incident sound pressure of 1.0 μbar)
p = rms pressure on the microphone

One microbar (μbar) is equal to 0.1 Pa. Equation (9.1) can be rearranged to obtain the output voltage:

$$E_{\text{out}} = p\, 10^{L_s/20} \qquad (9.2)$$

We recall the definition of Equation (3.6) for sound pressure level (SPL), L_p and recast Equation (9.2) as

$$E_{\text{out}} = 0.0002 \times \left(10^{L_p/20}\right) \left(10^{L_s/20}\right) = 0.0002 \times 10^{(L_p + L_s)/20} \text{V} \qquad (9.3)$$

Example Problem 1

A microphone has sensitivity rating of $L_s = -50$ dB V/μbar. Find the output voltage for a sound pressure level of 85 dB.

Solution

Equation (9.3) is applied to yield

$$E_{\text{out}} = 2 \times 10^{-4} \times 10^{(85-50)/20}\ \text{V} = 0.0112\ \text{V} = 11.2\ \text{mV}$$

Microphone sensitivities typically range between 0.5 μV/μbar (−125 dB V/μbar) and 3 mV/μbar (−50 dB V/μbar). A microphone used for general sound level pressure measurements in the frequency range of 10 Hz–20 kHz might have a sensitivity of 3 mV/μbar (−50 dB V/μbar). But if it were desired to measure in the frequency range extending beyond 100 kHz, a special microphone designed for this purpose would have a sensitivity of only 0.5 μV/μbar (−125 dB V/μbar).

9.5 Selection and Positioning of Microphones

Figure 9.3 illustrates a flow chart for the selection of microphones and their orientation during their use on the basis of the nature of the sound field. Two

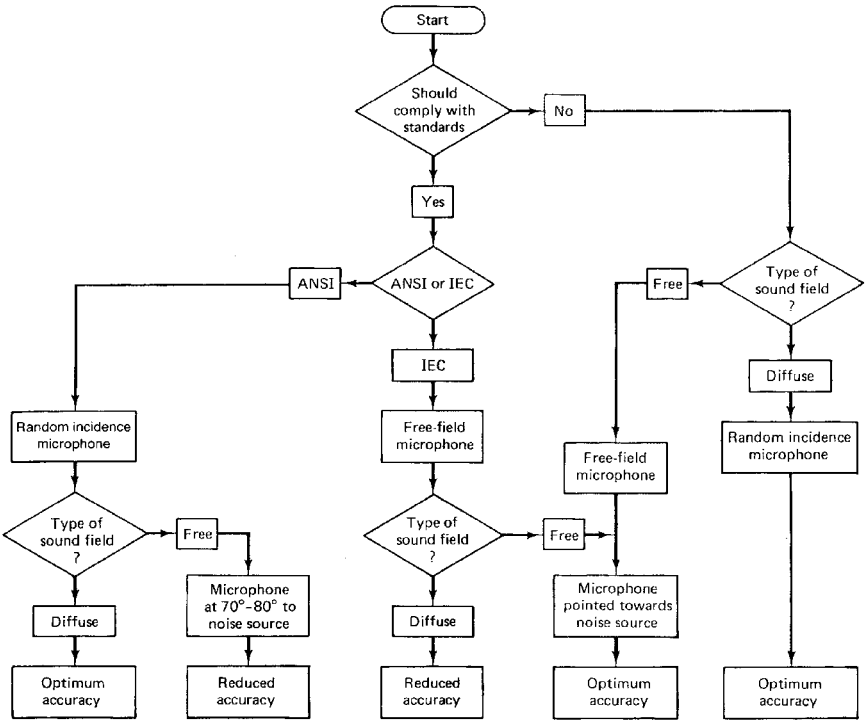


FIGURE 9.3. Flow chart for the selection of microphones and their orientations. (Courtesy of Brüel & Kjær.)

standards apply here, one by the American National Standards Institute (ANSI) which calls for microphones with random-incidence response and the other by the International Electrotechnical Commission (IEC) which specifies free-field microphones.

The *free field* occurs as a region that is not subjected to reflected waves, as is the case in an open field or in an anechoic chamber. The presence of a microphone in the sound field disturbs the field. A microphone designed to compensate for this disturbance is called a *free-field microphone*. In order to obtain maximum accuracy in measurements, the free-field microphone should be pointed toward the noise source [0° incidence, as shown in Figure 9.4(a)]. Microphone sensitivities are also stated in terms of mV/Pa.

The disturbance of the sound field by the presence of the microphone depends on the sound frequency, the direction of propagation, and the size and shape of the microphone. At higher frequencies, where the wavelength of the sound is small compared with the principal dimensions of the microphone, reflections from the microphone cause the pressure acting on the microphone diaphragm to differ from the actual free-field sound pressure that is supposed to be measured. Because a wavelength of 1 inch corresponds to 13,540 Hz, a 4-cm microphone will not provide accurate free-field measurements of noise in the frequency range in the neighborhood of 13 kHz and above. Even at 6 kHz, the error for a 4-cm microphone can exceed 2 dB.

The converse of the free field is the *diffuse field* which occurs as the result of multiple reflections. A *random-incidence microphone* is utilized in measuring sound in diffuse fields; it is *omnidirectional* in that it responds uniformly to sound arriving from all angles simultaneously [cf. Figure 9.4(d)].

Pressure microphones are designed to yield a uniform frequency response to the sound field including the disturbance produced by the microphone's presence. This type of microphone may also be used in diffuse fields. Using a free-field microphone in a diffuse field will result in lessened accuracy unless special circuitry in the measurement system provides compensating corrections. As shown in Figure 9.4(b), when a random-incidence microphone is used to measure sound in a free field, the unit should be placed at an incidence angle of $70^\circ - 80^\circ$ to the source. A pressure microphone should be positioned at an incidence angle of 90° (often referred to as the *grazing incidence*) in a free field [Figure 9.4(c)]. Microphone placement becomes more critical as the increasing sound frequencies approach the upper limits of accuracy.

For a windy environment, special precautions should be taken to obtain the proper data. The wind rushing past a microphone produces turbulence, generating pressure fluctuations resulting in low-frequency noise. In the use of *A*-weighted measurements, no precautions are necessary for winds below 8 km/h (5 mph), because the *A*-weighting network attenuates greatly at low frequencies. But for *C*-scale or linear sound level measurements, a windscreen should be employed for any sort of wind. The device should also be used for wind speeds above 5 mph in *A*-weighted measurements. Windscreens are typically open-celled polyurethane

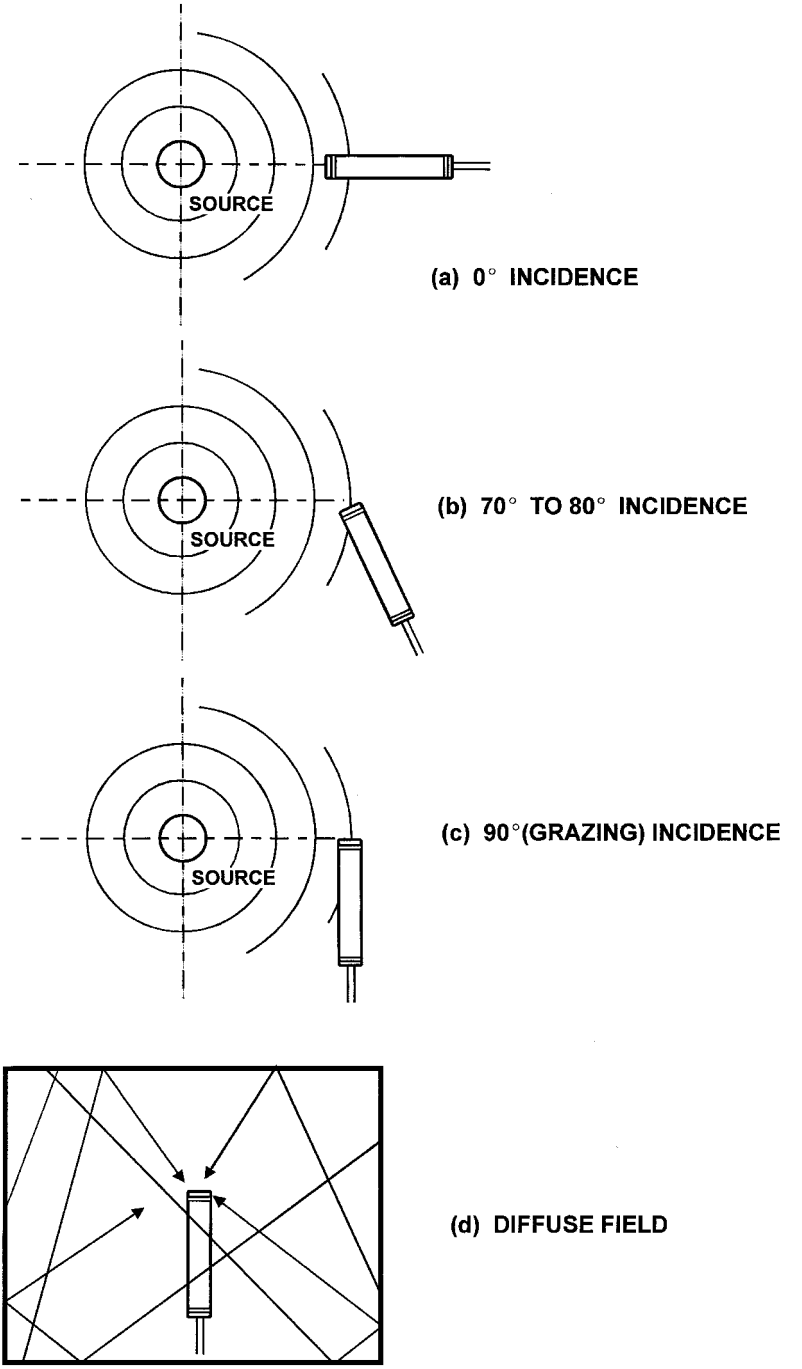


FIGURE 9.4. Microphone orientations with respect to the sound source.

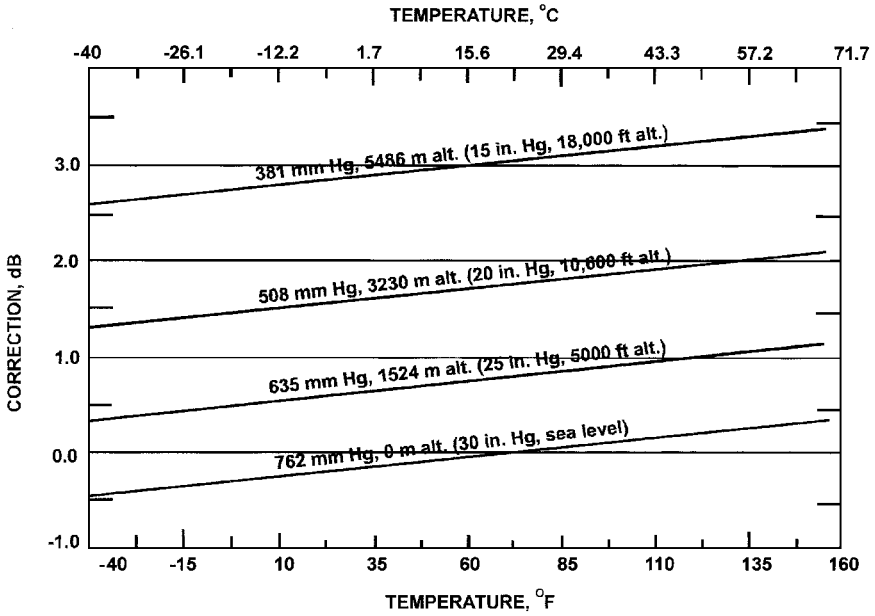


FIGURE 9.5. Sound pressure level corrections for conditions that deviate from the standard atmospheric pressure and temperature of 760 mmHg and 20°C.

foam spheres that are placed over the microphones. However, these may not be too effective if the wind speed exceeds 30 km/h (20 mph).

Changes in atmospheric conditions, namely, temperature and pressure, may necessitate corrections. Figure 9.5 illustrates a chart for determining corrections due to deviations from 1 standard atmosphere (762 mmHg or 30 in Hg) and 20°C, based on the following expression:

$$C_{T,P} = 10 \log \left(\sqrt{\frac{F + 460}{528}} \left(\frac{30}{B} \right) \right) = 10 \log \left(\sqrt{\frac{C + 273.3}{293.3}} \left(\frac{760}{B'} \right) \right) \tag{9.4}$$

where

$C_{T,P}$ = correction factor to be added to the sound pressure level, dB
F, C = temperature in °F or °C, respectively
B, B' = barometric in inches Hg or mmHg, respectively

Example Problem 2

For a sound pressure level, a reading of 92 dB is taken at the standard conditions of 1 atmosphere and 68°F, predict the readings at 10°F and at 110°F for the same atmospheric pressure.

Solution

Applying Equation (9.4)

$$C_{T,P} = 10 \log \left(\sqrt{\frac{10 + 460}{528}} \left(\frac{30}{30} \right) \right) = -0.25 \text{ dB} \quad (\text{for } 10^\circ\text{F})$$

$$C_{T,P} = 10 \log \left(\sqrt{\frac{110 + 460}{528}} \left(\frac{30}{30} \right) \right) = +0.17 \text{ dB} \quad (\text{for } 110^\circ\text{F})$$

At the lower temperature the reading would be $(92 - 0.25) = 91.8$ dB and at the higher temperature, $(92 + 0.17) = 92.2$ dB. Since the deviations are within the instrument error, the application of such corrections is not very meaningful.

The presence of reflecting surfaces affect measurements with the use of microphones. For example, the presence of a person near the microphone will disturb the sound field, and it would be advisable to place the microphone on a tripod and monitor the instrumentation from a distance. Also when a long cable is used to connect the microphone to other instruments, care must be taken to avoid noise generated by the motion of different segments of the cable with respect to each other. The cable should be constrained from moving and isolated from vibration to the greatest degree possible. The cable should also be well shielded from stray electromagnetic fields; but it is preferable that the tendency to noise generation be cut down by incorporating the preamplifier and the microphone into a single unit, thereby yielding a considerably greater signal-to-noise ratio (S/N ratio). As a rule, stronger signals are more impervious to outside influences than would be the case if the signals were not preamplified.

9.6 Vector Sound Intensity Probes

A sound intensity probe that is used to measure vector sound intensity is illustrated in Figure 9.6(a), and a block diagram in Figure 9.6(b) shows the components of this vector sound-measuring system. The device contains two microphones mounted face to face. Other modes of mounting are possible, including side by side and back to back. The probe measures sound pressure levels at two different points simultaneously. With the microphone spacing constituting a given factor, the sound pressure gradient can be determined and the particle velocity in a given direction can be calculated. The intensity vector component in that direction may then be established. The microphone spacing should be considerably smaller than the wavelength of the sound being measured in order to yield valid results from two-point measurements. For example, in order to sustain an accuracy of ± 1 dB, the upper frequency limit for 1-cm microphones set apart 12 mm from each other is approximately 5 kHz. If the microphones are spaced apart by only 6 mm, the corresponding upper frequency limit is 10 kHz.

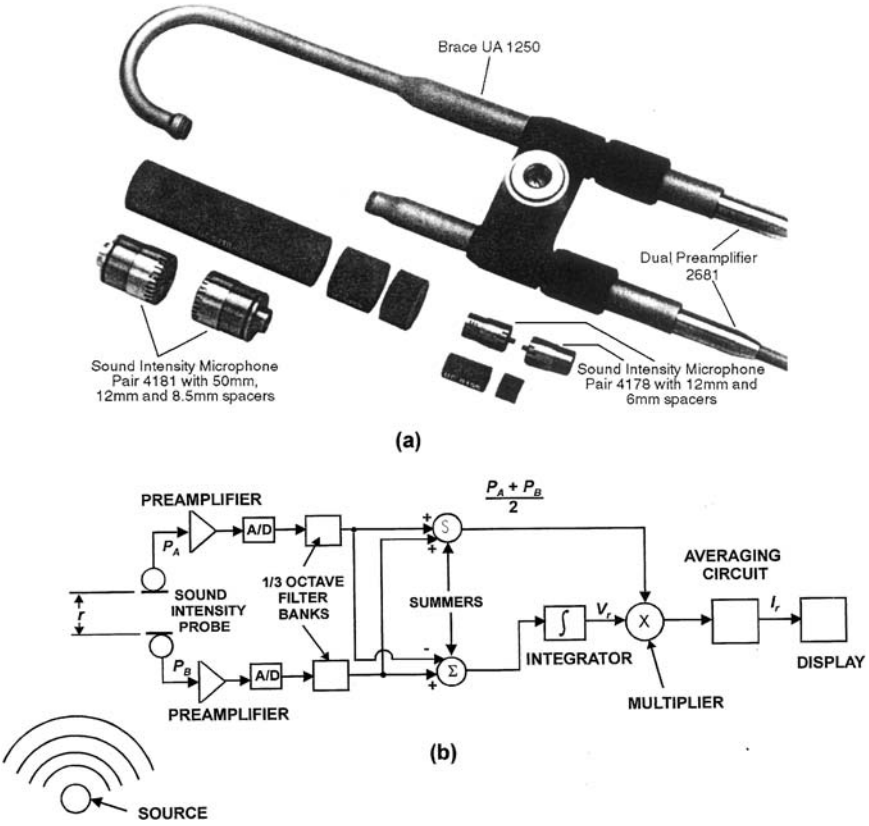


FIGURE 9.6. Sound intensity probe. (a) View of a sound intensity probe and its components. (b) Block diagram of the vector intensity measurement device. (Courtesy of Brüel & Kjær.)

9.7 The Sound Level Meter

The sound level meter (SLM), a most valuable means of assaying noise environments, amplifies the signal from a sensing microphone and processes the information for visual display or information storage. It is generally portable and battery operated. The quality ratings of sound level meters are specified by the American National Standards Institute (ANSI) and the International Electrotechnical Commission (IEC) according to the precision of these instruments. Types 1, 2, and 3 are, respectively, termed "precision," "general purpose," and "survey." In addition, Type 0 has been specified by IEC for laboratory reference standard (International Electrotechnical Commission, 1985). ANSI includes the suffix S in its standard to designate special-purpose meters, e.g., meters equipped with only *A*-weighting. Measurement precision depends on a number of factors, including meter calibration, method of surveying, and frequency content of the noise being measured.

Type 1 meters generally should measure with 1 dB accuracy and are employed to obtain accurate data for noise control purposes. The corresponding error for Type 2 meters, which are used for quick surveys, may not exceed 2 dB.

Figure 9.7(a) shows a variety of digital-readout SLMs and Figure 9.7(b) illustrates an SLM with a bar-graph spectrum display. Note that the contour of a meter case slopes away from the microphone in order to minimize reflections from its surfaces. Both Types 1 and 2 generally incorporate *A*-, *B*-, and *C*- weighting networks. The *A* and *C* networks are far more commonly used than *B*-weighting, particularly if low-frequency acoustic energy is present. Also a "fast" or "slow" response setting is generally available. The former setting, corresponding to a time constant $RC \approx 0.1$ s, responds more quickly to changes in noise levels, but the readings become more difficult to ascertain with very rapid fluctuations. The slow setting ($RC \approx 1$) reduces the response speed, as attested by the slower movement of the needle in the older-type analog-readout SLMs, and a better grasp of typical sound levels can be obtained for rapidly fluctuating sounds. More elaborate versions of sound level meters include 1/3-octave and octave filters to enable sound pressure measurements in various frequency bands.

In the block diagram of Figure 9.8, a typical SLM is shown to contain the following components: a 4-cm or 2-cm microphone feeding a preamplifier (which functions as a cathode follower), which in turn, relays to the one of the weighting networks that is selected by a switch. The weighted (or unweighted, if none of the weighting curves has been selected) signal then becomes amplified and then passes through a root-mean-square amplifier, becomes converted to logarithmic (i.e., decibel) form, and fed to either a digital or analog readout device. Some SLM models contain output jacks so that AC and DC signals from the meter can serve as inputs to other instruments, such as Fast Fourier Transform (FFT) analyzers, printers, or graphic plotters. By taking advantage of the PCMCIA modular technology, a meter can be designed to provide a variety of other functions such measuring L_{eq}, L_n, reverberation times, and so on.

9.8 Proper Procedures for Using the Sound Level Meter

To ensure that a sound level meter is in proper working order, an acoustic calibrator should be employed just prior to beginning a series of measurements and after completing the series. The calibrator is a single-tone, battery-driven device that fits over the microphone and produces a precise reference sound pressure level for calibrating the meter. The calibrator, illustrated in Figure 9.9, employs a zener-stabilized oscillator to provide impetus to a piezoelectric driver element that causes a diaphragm to vibrate at 1 kHz $\pm 1.5\%$. The diaphragm produces a sound pressure level of 1 Pa (corresponding to rms SPL of 94 $\pm$ 0.3 dB) in the coupler volume. The cavity at rear of the diaphragm behaves as a Helmholtz resonator with a natural frequency of 1000 Hz. Calibrators operate at 1 kHz, the international reference frequency for weighting networks. Therefore, no correction is required for calibrating instruments for weighted and unweighted measurements.

(a)



(b)



FIGURE 9.7. (a) Sound level meters. Some models come equipped with built-in octave or 1/3-octave filters for band measurements or with provisions for attaching separate filter units to perform such measurements. (b) Close-up of a sound level meter capable of providing an octave-band spectrum. Its compactness is apparent in this photograph. (Courtesy of Brüel & Kjær.)

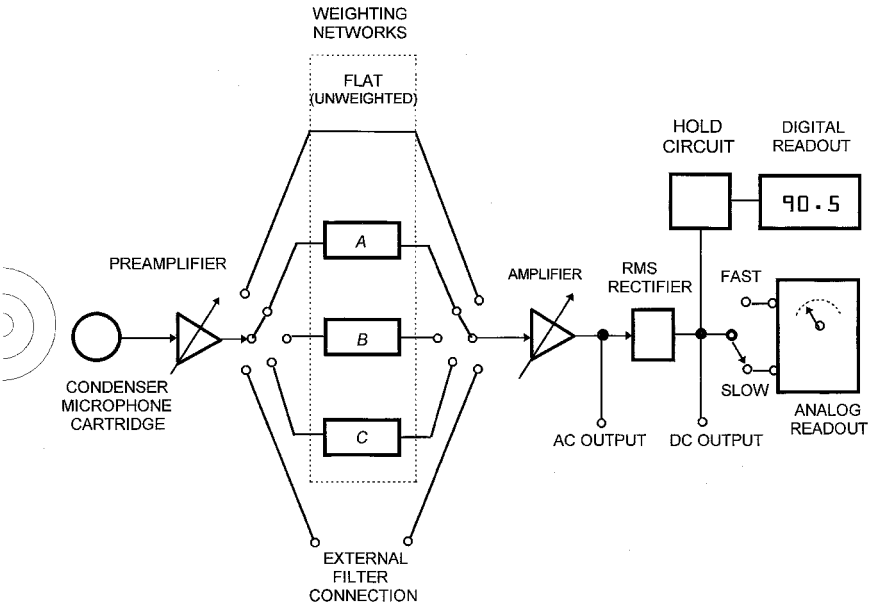


FIGURE 9.8. Block diagram of the principal elements of a sound level meter.

Periodically, every 6 months to a year depending on frequency and rigor of usage, both the SLM and the calibrator should be rechecked with calibration instruments which, in turn, have been calibrated periodically in a manner traceable to direct comparison with the standards set up by the U. S. National Institute of Science and Technology (NIST) or the appropriate standards bureau in another nation. Updated certifications of calibration are necessary for both SLM and its calibrator as proof of the accuracy and reliability of the measurement devices. While executing measurements and entering data, one should also record sufficient information to identify all measuring instruments at a later date should the

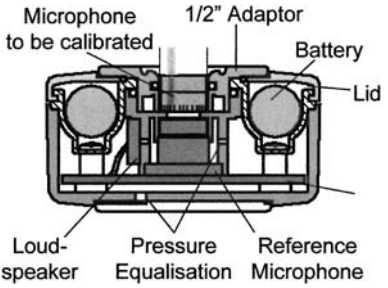


FIGURE 9.9. Cross section of an oscillator-driven sound level calibrator. This model produces an 1-kHz tone at a SPL of 94 dB. (Courtesy of Brüel & Kjær.)

need arise to prove the accuracy of the instruments at the time of its questioned usage.

One of the simplest case of noise measurements is that of steady noise, say, a motor running at steady speed in a factory. In this situation it would be necessary to take only a few sound level readings in dB(A) near the worker's ear and to check the measured values with established permissible levels. In the more common case of fluctuating levels, moving noise sources or receivers, etc., it may be necessary to take more readings over longer time intervals. Other types of meters or measurement procedures may be more suitable for sustained noise exposures, such as the use of the integrating sound level meter and the dosimeter which are described in the following sections.

9.9 The Integrating Sound Level Meter

In Chapter 3 the equivalent sound level was defined as

$$L_{\text{eq}} = 10 \log \frac{1}{T} \int_0^T 10^{L/10}\, dt$$

or in terms of N sound level measurements taken during Nequal intervals:

$$L_{\text{eq}} = 10 \log \left(\frac{1}{N} \sum_{i=1}^{N} 10^{L_1/10} \right) \tag{9.5}$$

Integrating sound level meters are based on the application of Equation (9.5) in measurements of fluctuating noise over a considerable interval. For example, a meter may be preset to measure in intervals of 1 s over a total time period of 15 min, thus calling for 900 individual sets of measurements. A measurement over 12 h may be programmed for 720 intervals of 1 min each. More elaborate meters designed for 24-h surveillances can compute the day–night equivalent sound level in which a 10-dB penalty is automatically added to noise levels occurring between the hours of 10 P.M. and 7 A.M. The integrating sound level meter is also capable of measuring the sound exposure level (SEL), which characterizes a single event on the basis of both the pressure level and the duration. This parameter is defined by

$$\text{SEL} = 10 \log \left(\int_0^T \frac{p_{\text{rms}}^2}{p_{\text{ref}}^2}\, dt \right) = 10 \log \left(\int_0^T 10^{L/10}\, dt \right) \tag{9.6}$$

where T is measured in seconds. If we consider a 2-s burst of sound at the rms pressure of 1 Pa, use of Equation (9.6) will indicate an SEL of 97 dB. Comparison of Equations (9.5) and (9.6) leads to the relationship between SEL and L_{eq}:

$$\text{SEL} = L_{\text{eq}} + 10 \log T \tag{9.7}$$

Example Problem 3

What is the SEL for an equivalent sound level pressure of 98 dB(A) lasting 8 s?

Solution

From Equation (9.7)

$$\text{SEL} = 98 + 10 \log 8 = 107 \, \text{dB}(A)$$

Equation (9.6) provides the means of computing SEL when the integrating sound level meter does not offer an automatic SEL mode.

9.10 Dosimeters

It is not generally a practical matter to trail continuously an industrial worker performing his or her duties in order to gauge the amount of noise exposure, particularly if that individual moves from place to place and is exposed to varying degrees of noise levels in the course of the day. A more convenient method of measuring the total exposure is to have this person wear either a *personal sound exposure meter* or a *noise dose meter* (also called a *noise dosemeter* or *noise dosimeter*). The personal sound exposure meter, which is schematically described in Figure 9.10, consists of a small microphone attached to a small extension cord so that it may be mounted close to the ear, a tiny amplifier that incorporates an A-weighting network, and a circuit that squares and integrates the electrical signal with respect to time. The unit normally includes a sound exposure indicator which can come separate from the wearable unit. This device should be able to respond to a wide range of frequencies and sound levels without the presence of a manual control. A *latching overload indicator* is also incorporated to provide a warning that excessive sound pressure levels are occurring within the frequency range of the instrument. The unit must be small, battery-powered, tamperproof, and constructed to withstand harsh environment.

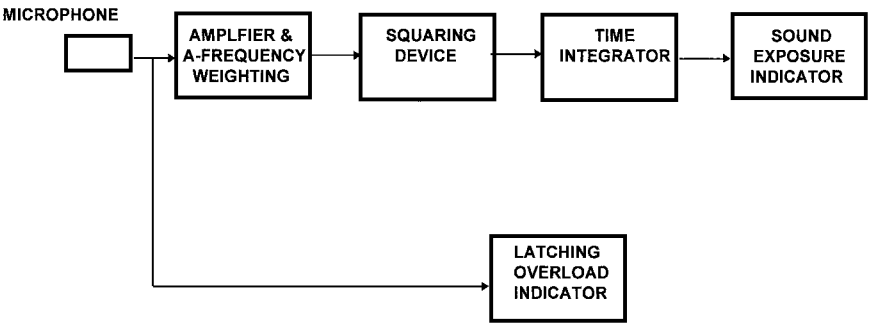


FIGURE 9.10. Schematic of a personal noise-exposure meter.

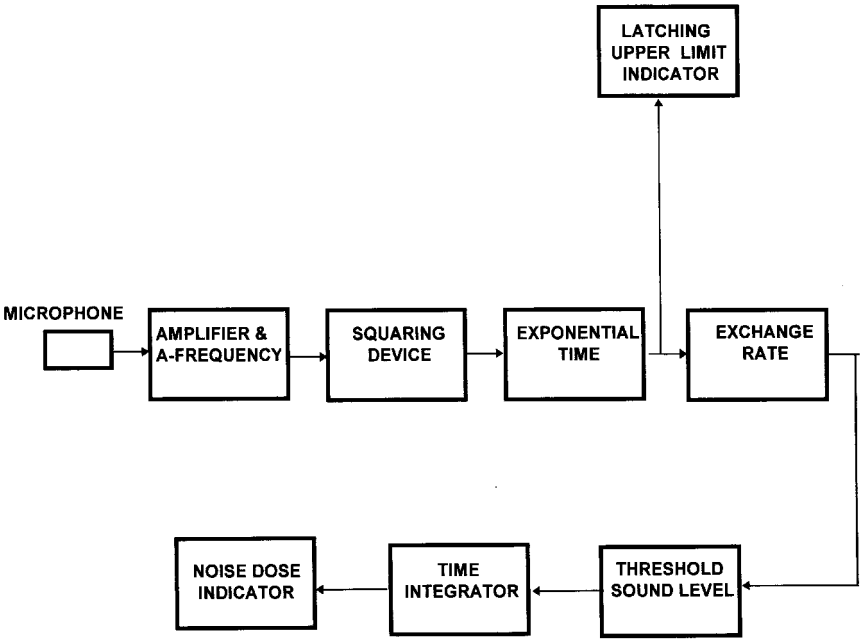


FIGURE 9.11. Schematic of a noise dosimeter.

The noise dosimeter is a device designed to measure the percentage of the maximum daily noise dose permitted by regulations. Its functional elements are given in Figure 9.11. Both the noise dosimeter and the personal sound exposure meter share the same requirements for operation under severe operating conditions; and both types of instruments include a microphone, an amplifier, *A*-weighting network, a squaring device, a time integrator, and an indicator. The personal sound exposure meters contain a latching overload but the noise dosimeter may include a latching upper-limit indicator. Additionally, the dosimeter must include exponential time-weighting usually incorporating the *slow-response* characteristic and a manufacturer-specified threshold sound level,[1] neither of which is found in the personal sound exposure meter.

American National Standard ANSI S1.25 gives the specifications for a noise dosimeter. Slow-time weighted, *A*-weighted sound pressure is integrated with a 5-dB exchange rate[2] in accordance with the regulations of U.S. Occupational Safety and Health Administration (OSHA, 1983) and Mine Safety and Health

[1] *Threshold sound level*, stated in decibels, is the *A*-weighted sound level specified by the manufacturer of the noise dosimeter below which the instrument provides no significant indication. The threshold sound level should be at least 5 dB less than the criterion sound level.

[2] The terminology *exchange rate*, expressed in decibels, refers to the change in decibels required to double (or halve) the exposure time in order to maintain the same amount of noise exposure. For example, with the 5 dB exchange rate, exposure to 90 dB for 8 h is equivalent to exposure to 95 dB for 4 h or to 100 dB for 2 h.

Administration (MSHA, 1982). The exchange rate may also be 3 or 4 dB depending on the application. ANSI S1.25 also specifies limits for the effects of changes in air pressure, temperature, vibration, and magnetic field on the noise dosimeter. The British standard BS 6402 calls for a personal sound level meter to measure sound exposure, i.e., the timed-integrated *A*-weighted sound pressure level with a 3-dB exchange rate and without exponential time weighting (British Standards Institution, 1983). A latching overload indicator is also mandated by the British standard to provide warning that the sound level received at the microphone has exceeded the measuring range of the instrument (up to 132 dB peak).

An IEC international standard for personal sound exposure meters defines a wider operating range for a Type-2 integrating averaging SLM. Under this specification, measurements are made of the exposures produced by impulsive, fluctuating, and intermittent sounds over a range of *A*-weighted sound levels from 80 dB to 130 dB. Excessive input sound levels should trigger the mandatory overload indicator.

9.11 Noise Measurement in Selected Frequency Bands, Band Pass Filters

A valuable means of analyzing noise is the evaluation of sound in each frequency band. If a machine emits a noise that indicates it is malfunctioning, the analysis of the sound output according to frequency can provide vital clues as to which component of the machine is defective. This situation calls for the use of a *spectrum analyzer*, which is a device that analyzes a noise signal in the frequency domain by electronically separating the signal into various frequency bands. This separation is executed through the use of a set of filters. A filter is a two-port electrical network with a pair of terminals at each port. A filter can be constructed with as few as two passive electrical components (e.g., a resistor and a capacitor), or it can be more complex involving a large number of passive components, or a combination of passive elements operating in conjunction with active components (e.g., op amps). Analog filters embody electronic circuitry tuned to pass certain frequencies, whereas digital filters make use of active electronic elements.

Figure 9.12 illustrates the effects of ideal and real filters. An ideal bandpass filter is a circuit that transmits only that part of the input signal within its bandpass ($f_1 \leq f \leq f_2$) and completely attenuates all of the components at all frequencies outside of the bandpass ($f \leq f_1$ and $f \geq f_2$). The ideal low-pass filter passes all signals up to frequency f_{lp} and rejects all signals having frequencies above f_{lp}. The ideal high-pass filter passes all signals above frequency f_{hp} and rejects all of the contents of the input signal below f_{hp}. In the real world, filters alter the shapes of the input signals to some degree. The amplitude and phase characteristics of the filter can be ascertained by computing the *transfer function* (or *filter response*), which is the ratio of the filter output to the filter input for all possible values of frequency. Both ideal and actual frequency responses are shown in Figure 9.12. The actual filters have characteristics of the form shown on the right-hand side

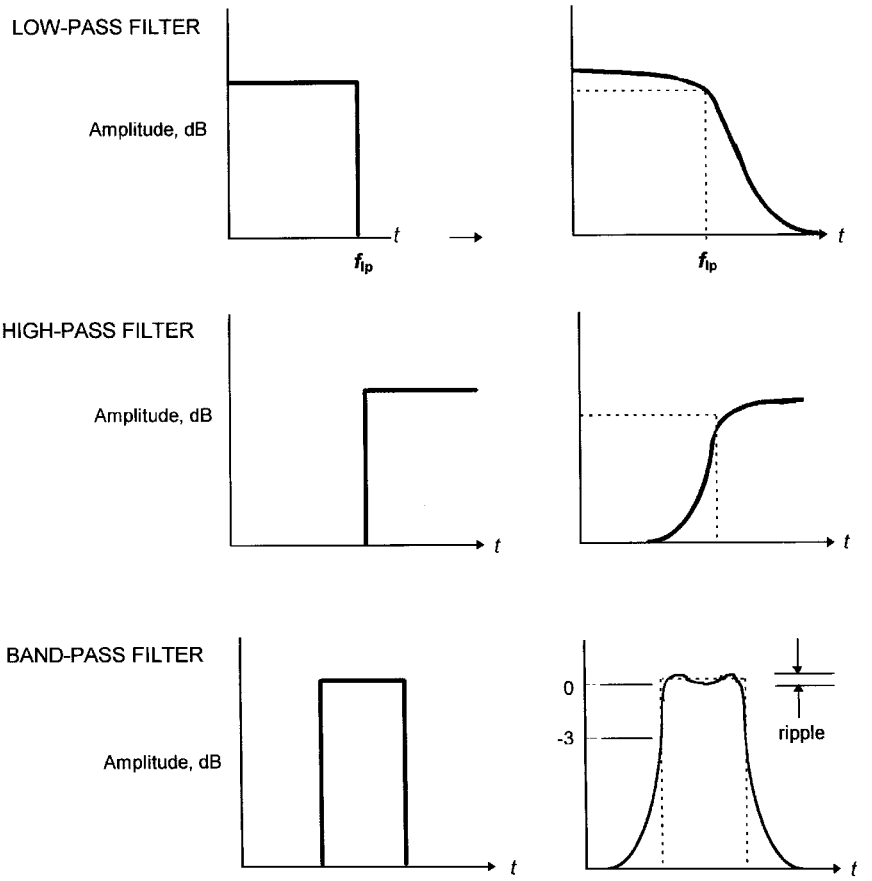


FIGURE 9.12. Real and ideal filters.

of Figure 9.12. In the case of the low-pass filter of Figure 9.13(a), the amplitude transfer function for the circuit is expressed as

$$|H(\omega)| = \left|\frac{E_o}{E_i}\right| = \frac{1}{\sqrt{1+(\omega RC)^2}} \tag{9.8}$$

where the subscripts i and o to voltage E denote the input and output values, respectively; R is the resistance in ohms and C the capacitance in farads. Equation (9.8) can be written to yield in decibels the characteristic of the form shown, plotted on the semi-logarithmic plot [Figure 9.13(b)] as follows:

$$\begin{aligned}\left|H'(\omega)\right| &= 20\log\left|\frac{E_o}{E_i}\right| = 20\log\left[\frac{1}{\sqrt{1+(\omega RC)^2}}\right]\text{dB}\\ &= -20\log\sqrt{1+(2\pi fRC)^2}\ \text{dB}\end{aligned} \tag{9.9}$$

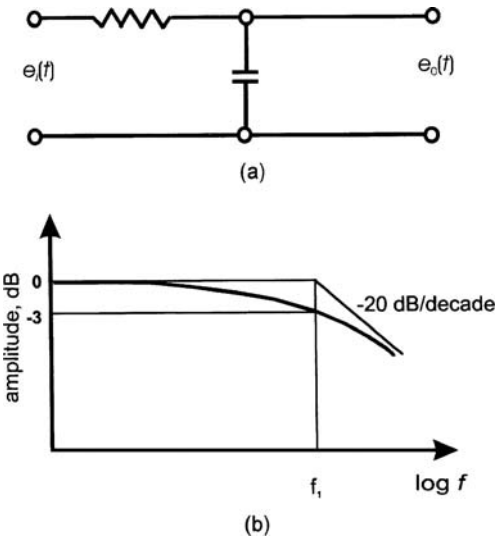


FIGURE 9.13. (a) Circuit for a single, low-pass filter and (b) frequency response of the low-pass filter.

Equation (9.9) becomes

$$\left|H'(\omega)\right| \approx -20 \log 1 = 0 \quad \text{for } (\omega RC)^2 \ll 1$$

that is, for relatively small frequency values the amplitude transfer function is 0 dB. In the case of large values of frequency, i.e., $(\omega RC)^2 \gg 1$, Equation (9.9) becomes

$$\left|H'(\omega)\right| \approx -20 \log(2\pi f RC)$$

The amplitude characteristic therefore exists as a slope of 20 dB per decade. The intersection f_1 of the two asymptotes is the *cutoff frequency* given by

$$f_1 = \frac{1}{2\pi RC}$$

According to Equation (9.9) the amplitude decreases by $1/\sqrt{2}$ at the cutoff frequency, or 3 dB. Because power is proportional to the square of the pressure amplitude, the cutoff frequency is also the *half-power point*. The amplitude responses of complex filters can be obtained in a similar manner.

Filters may be grouped in bands so that *serial analysis* is rendered possible by switching manually or automatically from one band to the next. The *octave band analyzer* constitutes an example of a commonly used serial analyzer. This device is so called because it is capable of resolving the noise-signal spectrum into frequency bands that are one octave in width. We have already seen in Chapter 3 that the center frequency of an octave band is $\sqrt{2}$ times the lower cutoff frequency, and the upper cutoff frequency is twice the lower cutoff frequency. The octave bands in the audio range are designated by their center frequencies and are

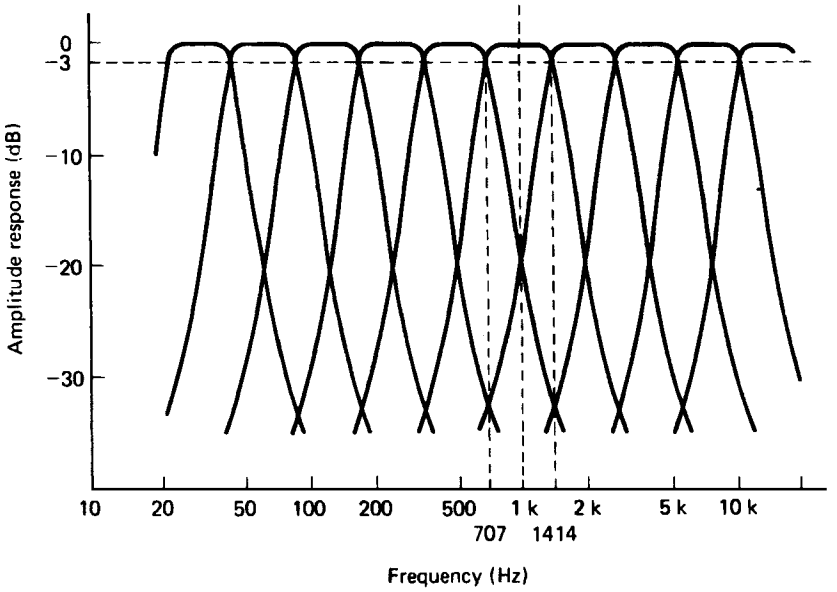


FIGURE 9.14. Filter response typical of an octave-band filter set.

standardized internationally at 31.5, 63, 125, 250, 500, 1000, 2000, 4000, 8000, 16,000 Hz. Figure 9.14 illustrates the frequency response typical of an octave-band filter.

In order to obtain more details of the noise spectrum, filters with bandwidths narrower than one octave must be used. *Narrow band analyzers* constitute excellent tools for diagnosing noises in industrial environments. Typical narrow-band analyzers measure in 1/3-octave and 1/12-octave bandwidths, with the former being more commonly used. These analyzers resolve the noise spectrum into third octaves and twelfth octaves. Denoting f_1 as the lower cutoff frequency, f_2 the upper cutoff frequency, and f_0 the center frequency, the relationship between the upper and lower cutoff frequencies for a 1/nth octave filter is

$$f_2 = 2^{1/n} f_1 \tag{9.10}$$

The center frequency is the geometric mean of the product of the upper and lower cutoff frequencies:

$$f_0 = \sqrt{f_1 f_2} \tag{9.11}$$

and the bandwidth **bw** is expressed as follows:

$$\mathbf{bw} = f_2 - f_1 = (2^{\frac{1}{2n}} - 2^{-\frac{1}{2n}}) f_0 \tag{9.12}$$

The smaller the bandwidth, the more detailed the analysis, and it obviously follows that the equipment becomes more costly. The advantages of narrower-band

analyzers become more apparent in the detection of prominent components of extremely narrow-band noise.

9.12 Real Time Analysis

Real time analysis entails the evaluation of a signal over a specified number of frequency bands simultaneously. This type of parallel process is schematically illustrated in Figure 9.15. Most real time analyzers rely on digital filtering of a sampled time series. A fluctuating noise signal is converted by a microphone assembly into an electric signal that is entered simultaneously into a parallel set of filters, detectors, and storage components. The scanned output is displayed as a bar graph of sound pressure levels versus frequency on a cathode ray tube (CRT) monitor or another type of display such as liquid crystal display (LCD) or plasma screen. The display must be refreshed several times per second as the level and frequency content of the input signal change. The scanning interval may be adjusted to meet different conditions, becoming longer to cover the cycling of a specific machine operation, or shortened to evaluate impact sounds.

A considerable variety of features are available on current real-time analyzers. Some features may include alphanumeric displays, choice of weighting and linear (unweighted) sound levels, linear and exponential averaging, time constants, spectrum storage for recall and comparison with other data, and integrated or external software packages for further analysis of analyzer output. A number of real time analyzers operate on batteries, and models are available that provide a choice of battery or AC operation.

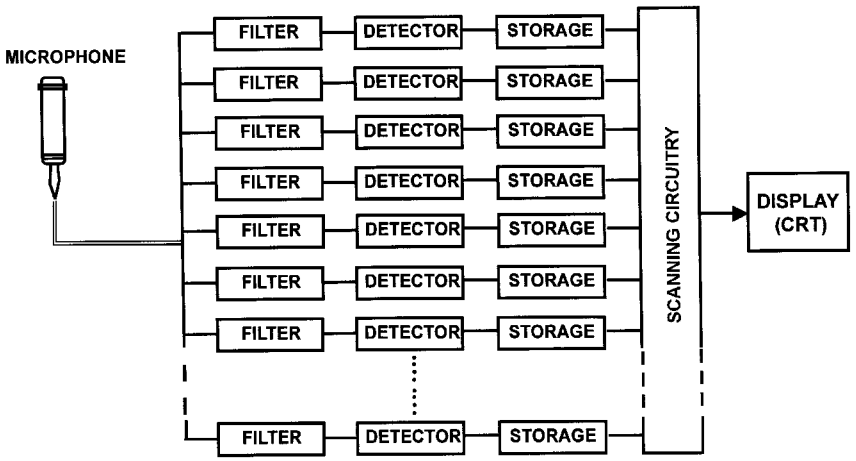


FIGURE 9.15. Schematic of the processing of a signal through a parallel bank of filters in a real-time analyzer.

9.13 Fast Fourier Transform Analysis

The Fast Fourier Transform (FFT) technique, rendered feasible by the advent of microcomputers, employs both digital sampling and digitization. Instead relying on bandpass filters to measure the analog amplitudes that formulate a signal's spectrum, the FFT analyzer executes an efficient transformation of the signal from the time domain to the frequency domain. As it is capable of executing nearly any analysis function on the signal at high speed, the FFT technique is an extremely powerful analytical method. The FFT analyzer captures a block of sampled data of finite length (generally 1024 or 2048 samples) for a processing interval. In transforming from the time domain to the frequency domain, the Fourier transform relates a function of time $g(t)$ to a function of frequency $F(\omega)$ in the following manner:

$$F(\omega) = \int_{-\infty}^{+\infty} g(t)e^{-i\omega t}\,dt \tag{9.13}$$

In measuring noise, a microphone assembly generates a voltage proportional to sound pressure. A *time series* is formed when the voltage is sampled at equal intervals, as shown in Figure 9.16. In order to transform this series into the frequency domain, Equation (9.13) must be reformulated into the *discrete Fourier transform* (DFT), given by

$$F(k) = \frac{1}{N} \sum_{n=0}^{n=N-1} g(n)e^{-2\pi ikn/N} \tag{9.14}$$

The matrix format of Equation (9.14) (Randall, 1977) is

$$\mathbf{F} = \frac{1}{N}\,[\mathbf{A}]\,\mathbf{g} \tag{9.15}$$

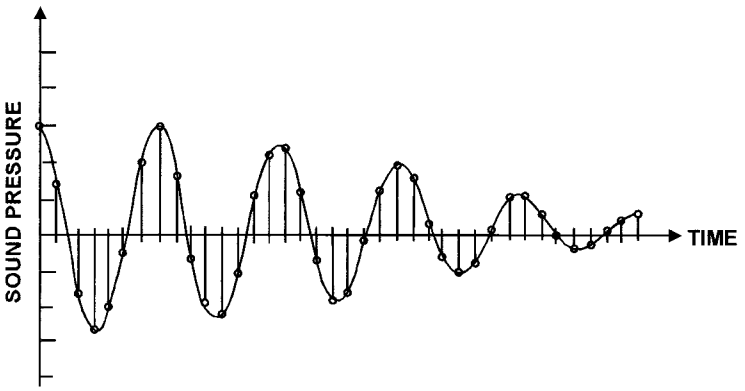


FIGURE 9.16. Sound pressure sampling at discrete intervals.

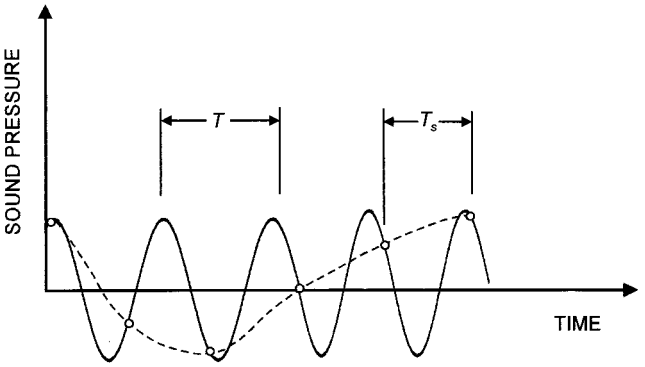


FIGURE 9.17. The effect of aliasing in generating a "false" wave.

where **F** is a column array of N complex frequency components, [**A**] is a square matrix of unit vectors, and **g** a column array of N samples in the time series.

Direct evaluation of the discrete Fourier transform obviously requires a large amount of number crunching, and the DFT methodology was rendered possible only in recent years by the development of efficient data processors. The *FFT algorithm*, developed by Cooley and Tukey, originally for the implementation on mainframe computers but now utilized even in portable FFT analyzers and laptops, cuts down on the number of calculations required to find a discrete Fourier transform (Cooley and Tukey, 1965). The Cooley–Tukey algorithm rearranges the [**A**] matrix of Equation (9.15) by interchanging rows and factoring, with the result that reduces memory requirements and saves computing time. Computing time can be cut even more so by tabulating sine and cosine values.

In the FFT process, the signal is modified in three ways, giving rise to the three potential traps of the FFT, these being the *aliasing*, *leakage*, and the so-called *picket fence effect*. Aliasing is the apparent measurement of a false or incorrect frequency. Higher frequencies (after sampling) appear as lower ones, as in Figure 9.17 in which a solid line represents a sound wave pressure with a period T. The wave is sampled at intervals of T_s corresponding to the sampling frequency of $f_s = 1/T_s$. The frequency of the dotted line that results from the sampling rate can then be incorrectly identified as the frequency of the solid line. This ambiguity can be avoided by having a sampling rate at least twice the frequency of the highest frequency present in the signal, i.e.,

$$f_s > 2f_{\max}$$

This minimum sampling frequency $2 \times f_{\max}$ is called the *Nyquist frequency*. The use of very steep antialiazing filters (typically 120 dB/octave) prevents frequencies that cannot be adequately sampled from being analyzed and renders it possible to utilize a major portion of the computed spectrum (e.g., 400 lines out of 512 calculated). In most FFT units this is done automatically when the frequency range is selected (This step also sets the appropriate sampling frequency).

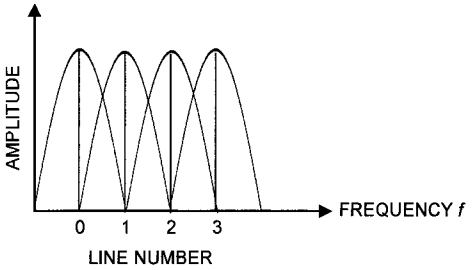


FIGURE 9.18. Illustration of the picket fence effect. If the frequency coincides with a line, it is indicated at its full level. Otherwise, it is represented at a lower level if the frequency falls between the lines.

The effect of leakage, or *window error*, becomes apparent when the power in a single-frequency component appears to leak into other frequency bands. Two primary causes exist for window error: (1) the signal is not fully contained within the observation window or (2) it is not periodic within the window. The simplest case is that of a monofrequency sinusoidal wave, which should yield only one frequency component in the FFT analysis if there are an integral number of periods of the sine wave in the finite record length. But a nonintegral number of periods will generally occur, and the cyclic repetition will yield a signal whose spectrum covers a range of frequencies. Leakage may be counteracted by forcing the signal in the data window to correspond to an integral number of period of all significant frequency components through a process called *order* or *tracking analysis*, where the sampling rate is related directly to basic frequency of the noise generating process (such as the shaft speed of a machine) and in *modal analysis* measurements where the analyzer cycle synchronizes with the periodic excitation signals.

The picket fence effect (Figure 9.18), which is not unique to FFT analysis, occurs in any set of discrete fixed filters. The magnitude of the amplitude error occurring from this effect depends on the degree of overlap of adjacent filter characteristics, a consideration that influences the selection of a data window.

9.14 Data Windows and Selection of Weighting Functions

If a steady pure-tone of an unknown frequency were to be analyzed, it would usually be sampled over a short-time interval that is termed *window duration*. A *rectangular window* will allow passage of a portion of the input signal without adjustment. That short segment obtained is presumably representative of the original signal, and this would also hold true if that segment embodies an integer number of periods of the original signal. The effect of window duration arising from the scanning of (usually) noninteger number of periods is depicted in Figure 9.19. Such a signal would be analyzed as if the segment iterated itself as in the figure.

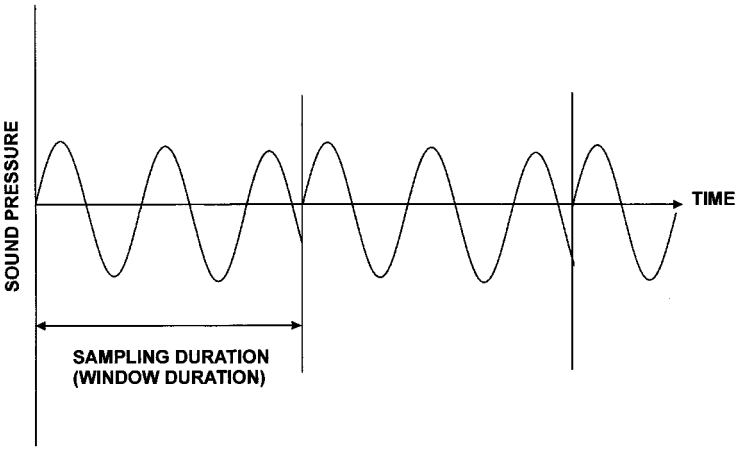


FIGURE 9.19. Effect of window deviations in FFT scanning of noninteger number of periods.

The discontinuities at the ends of the segments will engender a frequency spectrum differing from the true frequency content of the original signal. Aperiodic signals may also be misinterpreted on account of the finite length of sampling.

The weighting or window function modifies the shape of the observation window by tapering off both the leading and trailing edges of the data. Because the weighting function is used to obviate the effects of duration limiting, it increases the effective bandwidth. This increase is stated in terms of the *noise bandwidth factor* (NBF) defined as

$$\text{NBF} = \frac{\text{Effective bandwidth with window function}}{\text{Effective bandwidth without window function}} \tag{9.16}$$

The ideal value of NBF is unity. Another important consideration in the use of weighting functions is the *side-lobe ratio* (SLR) or *highest side-lobe level*, expressed as

$$\text{SLR} = \frac{\text{Most sensitive out-of-band response}}{\text{Center of bandwidth response}}\text{dB} \tag{9.17}$$

A greater negative value of SLR (in decibels) is more desirable. But the selection of the two parameters given by Equations (9.16) and (9.17) requires a compromise, i.e., a tradeoff between the steepness of the filter characteristic on one hand and the effective bandwidth on the other. Table 9.1 lists the popular types of data windows and their parametric values. The *Hanning window* (one period of a cosine-squared function) constitutes a good choice for stationary signals, because that function has a zero value and slope at each end and thus renders a gradual transition over the discontinuity of data. Figure 9.20 illustrates the rectangular, Hanning, and Hamming weighting functions.

TABLE 9.1. Properties of Data Windows.

Window type	Maximum side lobe, dB	Side lobe falloff, dB/decade	Noise bandwidth (relative to line spacing)	Maximum amplitude error, dB
Rectangular	−13.4	−20	1.00	3.9
Hanning	−32	−60	1.50	1.4
Hamming	−43	−20	1.36	1.8
Kaiser-Bessel	−69	−20	1.80	1.0
Truncated Gaussian	−69	−20	1.90	0.9
Flattop	−93	0	3.70	0.1

9.15 Resolution

The smallest increment in a parameter that can be displayed by a measurement system is the *resolution* for that system. In FTT analysis, the resolution is expressed by

$$\beta = \frac{f_s N}{N} = \frac{2 f_R}{N} \tag{9.18}$$

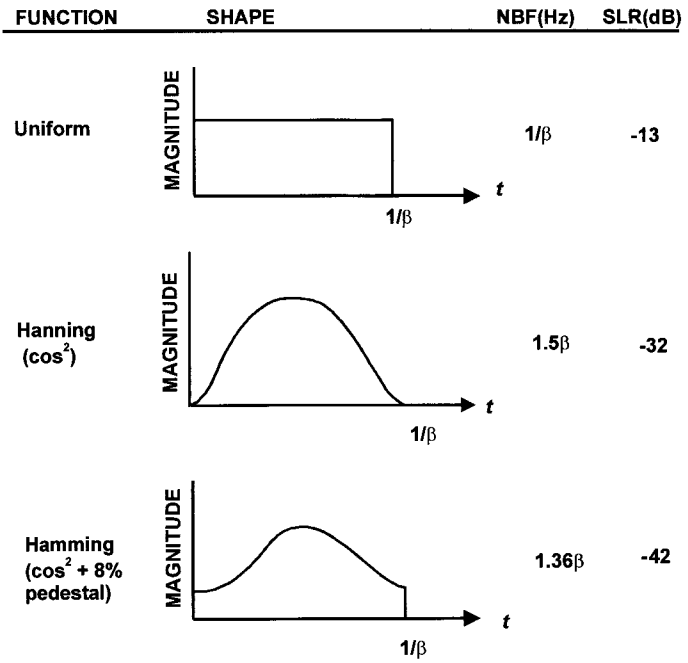


FIGURE 9.20. Typical weighting functions for FFT analysis, with corresponding noise band factor (NBF) and side lobe ratio (SLR).

where

β = the resolution which is the frequency increment between lines in a spectrum, Hz
f_s = sampling frequency, the reciprocal of sampling period T_s
N = number of samples in the original time series
f_R = frequency range, normally from 0 to the Nyquist frequency

Some versions of FFT analyzers incorporate a "zoom" capability which displays selected portions of a spectrum with finer reduction. Normally, in order to increase the resolution (which means that β must decrease) the number of samples must be larger. This is not always feasible since either the processing time will increase or the block size is limited by machine memory capacity. The zoom capability overcomes this limitation by permitting the spectrum analyzer to concentrate its entire resolution, whether it be 200 or 800 or 1000 lines, on a small frequency interval selected by the user. While arbitrarily fine resolution is achievable, a compromise must be effected between resolution and the required sampling time, as attested by Equation (9.18). Note that according to this equation, if the frequency resolution for a signal is 10 Hz, the sampling time is 0.1 s, but if the frequency resolution is changed to 0.1 Hz, the corresponding sampling time will be 10 s.

A number of FFT analyzers incorporate large digital processing capacities that can present measurement results in the format of three-dimensional plots (or "waterfall" or "cascade" plots) with the vertical coordinate representing the amplitudes of the spectra as functions of the other two coordinates, one representing the frequency f and the other time t. The scan presented on a display can show a "running" plot that moves in the direction of increasing t. This type of display is most useful in observing the behavioral characteristics of transient sounds.

9.16 Measurement Error

Noise is usually random in nature, i.e., its sound pressure level cannot be predicted for any instant. But statistical means can be used to describe random noise. If the noise is relatively constant in level and frequency content then it may be deemed a *stationary random process*, one in which statistical parameters are invariant with respect to time. A machine operating in a constant cyclic manner may emit different levels of sound, with corresponding changes in frequency content, for each successive instant of the cycle, but a measurement interval over a group of cycles will yield a consistent spectral distribution over time. Analysis data based on a very short interval that is less than the length of a single cycle is certain to yield misleading results.

Let us consider noise with an idealized Gaussian or normal probability distribution. The standard deviation ε, which is the uncertainty in the rms signal divided by the long-term average rms signal, relates to the ideal filter bandwidth **bw** and

the averaging time T, according to

$$\varepsilon = \frac{1}{2\sqrt{\mathbf{bw} \times T}} \tag{9.19}$$

The *measurement* uncertainty is *twice* that given by Equation (9.19).

Example Problem 4

A normal distributed noise is to be analyzed in 20 Hz bands, with the measurement uncertainty not to exceed 4%. Find the necessary averaging time T.

Solution

Using Equation (9.19) modified to give the measurement uncertainty, we obtain

$$T = \frac{1}{\mathbf{bw} \times \varepsilon^2} = \frac{1}{20 \times 0.04^2} = 31.3\text{s}$$

9.17 Sound Power

Sound power denotes the rate per unit time at which sound energy is radiated. This rate is expressed in watts. The sound power L_w level defined by

$$L_w = 10 \log\left(\frac{W}{W_0}\right)$$

is given in decibels. W is the power of the sound energy source in watts and $W_0 = 1$ pW is the standard reference power in watts.

The principal advantage of using sound power level rather than the sound pressure level given by Equation (3.22) to describe noise output of stationary equipment is that the sound power output radiated by a piece of equipment is independent of its environment. The sound energy output of the machine will not change if that unit is moved from place to place, provided it operates in the same manner. Sound power is primarily used to describe stationary equipment, but it is not generally used to rate mobile equipment since the operational situations may be too highly variable, as is the case with construction equipment.

Sound power may be measured directly using spund intensity instrumentation or indirectly, either determined from the rms sound pressures at a number of microphone locations spatially averaged over an appropriate surface enclosing the source in a free field over a reflecting plane (as exemplified by the use of a semianechoic chamber) or in a totally free field (full anechoic chamber), or averaged over the volume of a reverberation chamber in which the measurements are conducted.

Some sources are omnidirectional, i.e., they radiate sound uniformly in all directions. Most sources are highly directional, radiating more sound energy in some directions than in others. Hence, the *directivity*, or *directional characteristic*, constitutes an important descriptor of a sound source. In a free field or anechoic

chamber, the directivity is readily apparent, owing to the absence of reflections. But in a highly reflective environment, such as that of an echo chamber, the multiple reflections that occur render the directivity less important and the sound field becomes more uniform.

9.18 Measurement of Sound in a Free Field over a Reflecting Plane (Semianechoic Chamber)

We can summarize the measurement procedure in a free field over a reflecting plane as follows:

1. The source is surrounded with an imaginary surface of area S, either a hemisphere of radius r or a rectangular parallelepiped.
2. The area of the hypothetical surface is calculated. $S = 2\pi r^2$ in the case of a hemispherical surface, and $S = ab + 2(ac + bc)$ in the case of the parallelepiped having length a, width b, and height c.
3. The sound pressure is measured at designated points of the imaginary surface.
4. The average sound pressure level $\bar{L}_p$ is computed from the measured results of the previous step. This is found from

$$\bar{L}_p = 10 \log \left[\frac{1}{N} \sum_{i=1}^{N} 10^{L_i/10} \right] \tag{9.20}$$

   where N is the total number of measurements and L_i denotes the measured value of the SPL at the designated point i.
5. The sound power level is then calculated from the following:

$$L_w = \bar{L}_p + 10 \log(S/S_0) \tag{9.21}$$

where S_0 is the reference area of 1 m^2.

The above procedure applies only if the source is not too large, i.e., the radius r of the hypothetical hemisphere should be at least 1 m and at least twice the largest dimension of the source (or the perpendicular distance between the source inside the imaginary parallelepiped and a measurement surface is 1 m), and the background noise level is more than 6 dB below that of the source. The rectangular parallelepiped setup is preferred for large rectangular sources.

Figure 9.21 shows the designated points on the hemispherical surface where the microphones are located. The corresponding points for the rectangular parallelepiped are given in Figure 9.22. These designated points are associated with equal areas on the surface of the hemisphere or the rectangular parallelepiped. The SPL is usually measured at the designated points with A-weighting or in octave and partial octave bands, with the meter set in the slow-response mode.

The applicable international standards for acceptable sound power measurement techniques under semianechoic conditions are given in ISO 3744 and ISO 3745. Adjustments in the values of the measured SPL should be made for the presence of background noise. Equations (9.20) and (9.21) are used to convert the measurements into the desired values of averaged SPL, $\bar{L}_p$, and sound power level L_w.

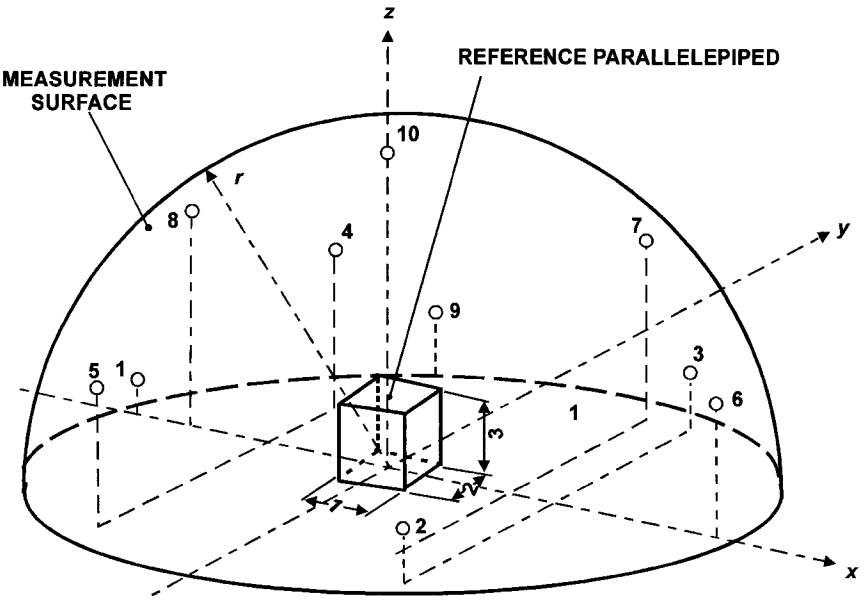


FIGURE 9.21. Locations of the microphones on the surface of an imaginary hemisphere surrounding a source whose sound power level is to be measured according to ISO 3744.

The directivity index **DI** of the source may be computed from measurements in a semianechoic chamber from the following equation:

$$\mathbf{DI} = L_{pi} - \bar{L}_p + 3 \text{ dB} \tag{9.22}$$

where L_{pi} is the sound pressure level measured at point i, located on the measuring surface and defining the direction along which the DI is desired at a distance from the source.

9.19 Measurement of Sound Power Level in a Free Field (Full Anechoic Chambers)

The procedure for measuring sound power in a free field is basically the same as that for a free field with a reflecting surface (semianechoic condition), with some modifications. In this case, the source is centered in a hypothetical sphere of radius $r$and surface area $S = 4\pi r^2$. The sound pressure levels are measured at specific points on the spherical surfaces; these points are stipulated by ISO 3745 and shown in Figure 9.23, and defined in terms of Cartesian coordinates in Table 9.2. Equations (9.20) and (9.21) also apply to yield the average sound pressure level $\bar{L}_p$ and the sound powerL_w. The surface area ratio S/S_0 in Equation (9.21) now refers to two spheres, the hypothetical one (of radius r) used for placement of measuring sensors and the other a reference sphere with an area of unity, with

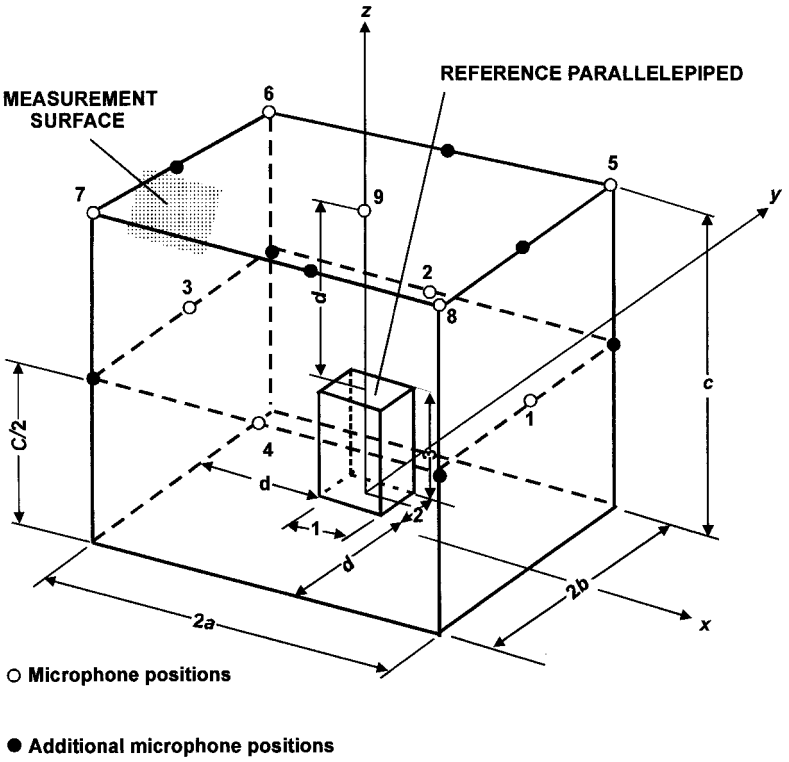


FIGURE 9.22. Locations of the microphones on the surface of an imaginary parallelepiped surrounding a source whose sound power level is to be measured according to ISO 3744.

the result that this ratio is equal to simply $4\pi r^2$. Equation (9.21) becomes simply

$$L_w = \bar{L}_p + 20 \log r + 10.99 \tag{9.23}$$

where all dimensions are stated in SI units. The rectangular measurement surface is not generally used for measurements in a free field. As specified by ISO 3745, the radius of the test sphere should be at least twice as large as the major source dimension, but never less than 1 m. A large source will necessitate a very large anechoic chamber for measurement purposes.

9.20 Sound Power Measurement in a Diffuse Field (Reverberation Chamber)

Reverberation chambers are rooms with extremely reflective walls, ceilings, and floors. If the surfaces are made nonparallel to each other, standing waves can be avoided. When a steady noise source is operating, the sound field is diffused everywhere in the room except in the immediate vicinity of the source, as the

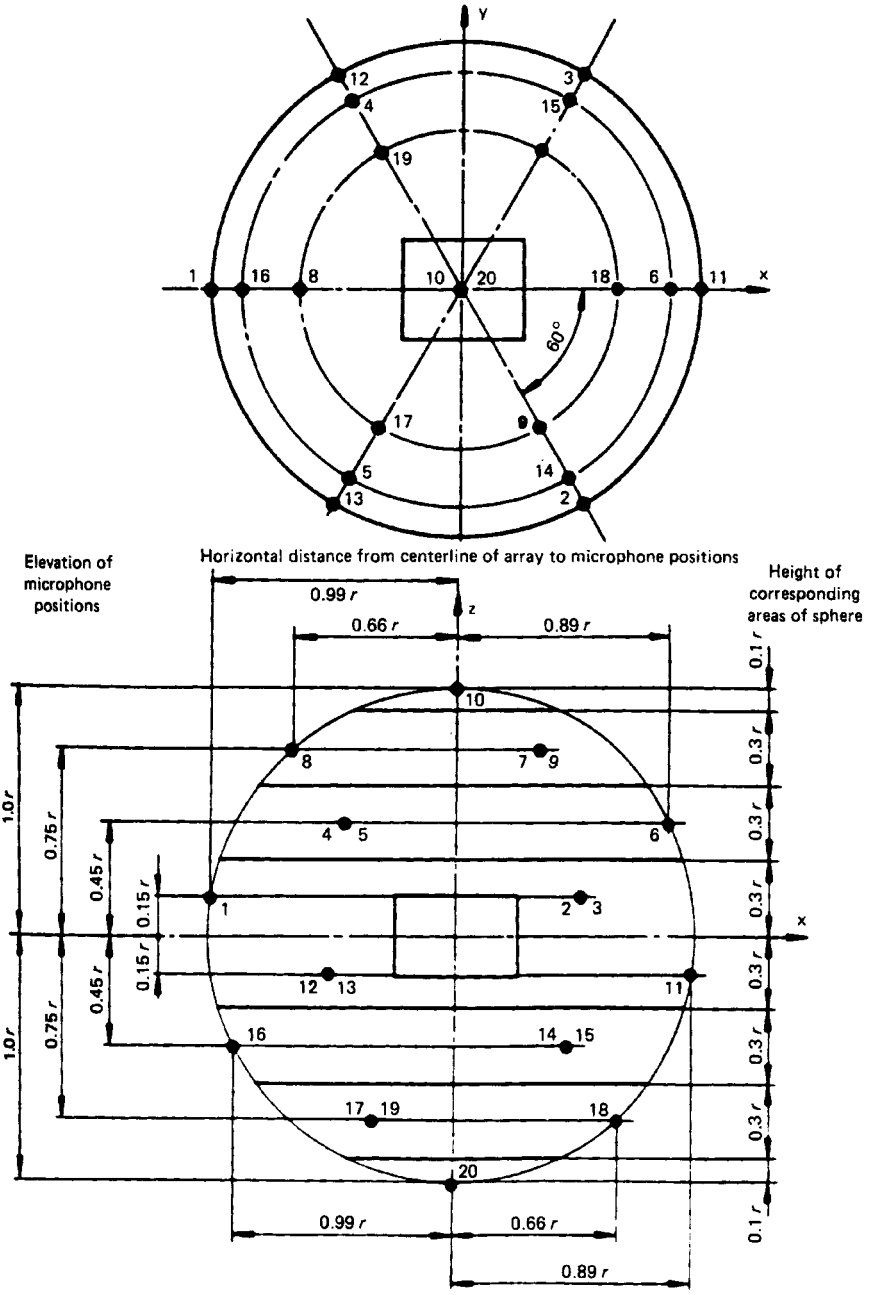


FIGURE 9.23. The locations of microphones on an imaginary spherical surface surrounding the source in a free field, according to ISO 3755.

TABLE 9.2. Microphones Array Positions[a] in a Free Field According to ISO 3745.

No.	x/r	y/r	z/r
1	−0.99	0	0.15
2	0.50	−0.86	0.15
3	0.50	0.86	0.15
4	−0.45	0.77	0.45
5	−0.45	−0.77	0.45
6	0.89	0	0.45
7	0.33	0.57	0.75
8	−0.66	0	0.75
9	0.33	−0.57	0.75
10	0	0	1.0
11	0.99	0	−0.15
12	−0.50	0.86	−0.15
13	−0.50	−0.86	−0.15
14	0.45	−0.77	−0.45
15	0.45	0.77	−0.45
16	−0.89	0	−0.45
17	−0.33	−0.57	−0.75
18	0.66	0	−0.75
19	−0.33	0.57	−0.75
20	0	0	−1.0

[a] This table lists the Cartesian coordinates (x, y, z) with the origin located at the center of the source. The vertical axis is perpendicular to the horizontal plane $z = 0$.

result of sound waves reflecting back and forth between room surfaces until they die out. A steady noise source generates sound and builds up the sound pressure level from ambient level in the room, until an equilibrium sound pressure level is reached. This occurs when the new acoustic energy emanating from the source offsets the dissipation of reflected sound energy in the slightly elastic reflecting surfaces (however hard they may be) and in the slightly viscous air of the chamber and through energy leakage from the room. The applicable international standards for measuring the sound power levels in reverberation chambers are ISO 3741 and ISO 3742, the former giving details for broadband noise and the latter for discrete-frequency and narrow-band sources. The relationship between the sound power of a source and the average reverberant sound level can be determined and allows calculation of the sound power.

The procedure for measuring the sound power level in an echoic chamber may be summarized as follows:

1. The reverberation time T_{60} is measured by using a standard technique prescribed by ISO 354.[3]

[3] Reverberation time T_{60} for an enclosure is the time interval required for a sound pressure level to drop 60 dB after the sound source has been stopped. More detail treatment of T_{60} is rendered in Chapter 11.

2. The room volume V and the total surface area A of the test chamber are calculated from its internal dimensions. The barometric pressure B is measured but this constitutes only a small influence on the sound power level of the source.
3. The average sound pressure level $\bar{L}_p$ in the room is obtained by sweeping a microphone at steady speed over a path at least 3 m in length while its output signal is measured and averaged on a root-mean-square-pressure basis. The sweep time should be at least 10 s for the 200 Hz band and higher and 30 s for lower frequencies. Measurements in the near field (close to the source) and very near the chamber surfaces should be avoided. The microphone should not be located closer to a surface than one half the wavelength of the lowest pertinent frequency. Alternatively, the average value may be obtained by averaging the output of an array of three fixed microphones spaced a distance of $\lambda/2$ apart (wavelength λ corresponds to the lowest frequency of interest). $\bar{L}_p$ constitutes the average band pressure level corrected for the background noise (the sound level that exists in the measurement chamber when the source is not operating).
4. The contribution of each frequency band to the sound power level L_w of the source, is calculated by using the following equation:

$$L_w = \bar{L}_p - 10\log\left(\frac{T_{60}}{1\text{s}}\right) + \log\left(\frac{V}{1\text{m}^3}\right) + \log\left(1 + \frac{S\lambda}{8V}\right) - 10\log\left(\frac{B}{1000\text{ mbar}}\right) - 14\text{ dB} \tag{9.24}$$

where wavelength λ corresponds to the center frequency of the frequency band of interest. The A-weighted sound power level L_{wA} may be computed from octave-band or one-third-octave band levels, according to ISO 3741, Annex C.

9.21 Substitution (or Comparison) Method for Measuring Sound Power Level

A simpler method than the direct method described above can be used to determine the sound power level L_w of an unknown source by comparing two measurements, without the necessity for knowing the reverberation time of the test chamber. In fact, this method does not even require the use of a special laboratory chamber, and it can be applied in situations where it would be impracticable to move a large piece of machinery to a laboratory. Commercially available reference sources provide known values of sound power level L_{wr} for each octave band and one-third octave band. Reference sound sources are classified into three types: aerodynamic sources, electrodynamics sources, and mechanical sources. Aerodynamic sources, the most prevalent type, consist of a specially designed fan or blower wheel driven by a motor. Technical requirements are listed in international standard ISO 6926.2. The comparison method, defined by ISO 3741, is as follows:

1. The procedures for measuring the average of the sound pressure levels $\bar{L}_p$ are conducted in the test room in the same manner described above for the direct method. $\bar{L}_p$ is measured for the unknown source, and then the unknown sound source is turned off.
2. The reference source is turned on, and $\bar{L}_{pr}$ is measured for the reference source. The reference sound sources should be mounted on the floor at least 1.5 m away from another sound reflecting surface, such as a wall or the unknown source being evaluated.
3. The sound power level for the source undergoing measurement can be simply computed from

$$L_w = \bar{L}_p + (L_{wr} - \bar{L}_{pr}) \text{ dB} \tag{9.25}$$

9.22 Alternation Method for Measuring Sound Power Level

A reference source with adjustable sound power—for example, a wideband source that generates pink noise (i.e., the same level in each band) or an octave band filtered noise—is used in the alternation method for measuring sound power. A readout meter indicates the reference source power output. The procedure is as follows:

1. The noise source being tested is operated in a diffuse field. The spatial average sound power level $\bar{L}_p$ is measured in each octave band.
2. The reference source replaces the noise source. The reference source is adjusted until it produces the same sound level as the tested source did in the first octave band. The sound power level L_w indicated on the reference source meter is recorded. This procedure is repeated for each of the other octave bands.
3. The sound power levels noted in the reference source meter are the sound power levels for the unknown source for each octave band. The total sound power can be computed from

$$\bar{L}_w = 10 \log \sum_{i=1}^{N} 10^{\bar{L}_{wi}/10} \tag{9.26}$$

where $\bar{L}_{wi}$ is the spatial average sound power level for the ith octave.

9.23 The Addition Method for Measuring Sound Power Level

The addition method is useful for situations where the machine under test cannot be conveniently shut down, for example, a power station generator. As with the alternation method, an adjustable reference source is used. The procedure is as follows:

1. The spatial average sound power level $\bar{L}_p$ generated by the subject machine is measured in each octave band.
2. The reference source is placed near the machine and both are operated simultaneously. The reference source is adjusted until it produces an additional 3 dB in the sound pressure level in each octave band than when the machine was operating by itself. The reference source is then producing a sound power level in that band equal to that generated by the machine under test. The sound power level indicated on the meter of the reference power source is recorded. The procedure is repeated for each of all the other octave bands.
3. The total sound power level can now be calculated from the use of Equation (9.26).

9.24 Data Acquisition Systems

For a number of years, magnetic recorders provide a permanent record of noise data taken *in situ* for subsequent analyses on instruments located elsewhere and for archival purposes. Currently, digital recorders are being used for acoustical data acquisition. Very few, if any, analog recorders (either AM or FM) are currently sold. Computer technology has advanced to the stage where measured data and its analytical results can be acquired and digitally stored in computer memory or on recordable CD-ROMs, DVD-R, and removable cartridges or disks. A CD-ROM can hold approximately 700 MB of data, but the DVD (which can be used to contain digital data and not just video programs) holds nearly ten times as much. Newer storage disks are soaring past the gigabyte range to encroach in the terabyte territory. We are witnessing the rapid demise of the magnetic tape recorder in favor of computerized acquisition devices and digital storage units. The recent introduction of flash disks provides the potential for even greater portability of data. Transient sounds, analyzed on a cascade-type FFT analyzer, are more conveniently archived in a nonvolatile memory medium such as a random-access removable cartridge than they would be on a magnetic tape (which obviously does *not* provide random access), for the purpose of later retrieval.

9.25 Integration of Measurement Functions in Computers

Advances in computer technology and software development tools make it possible to integrate measurement functions into a personal desk computer or a laptop equipped with the appropriate acquisition printed circuit boards, high quality sound cards, and the applicable sensors. A specially equipped personal computer can execute DSP-based (diagnostic signal processing) signal generation, filtering, and spectrum analysis. Hence, a single computer can replace a whole rack of dedicated analog units linked together by BNC cables. In testing the performance of a loudspeaker through traditional analog means, a sine wave generator provides a signal to the loudspeaker and a calibrated microphone picks up the loudspeaker's

acoustical output and feeds it to a spectrum analyzer, which then relays the data to a display or a recorder. The dedicated functions of the traditional hardware—signal generation, filtering, analysis, and data handling—can now be performed by software in such a manner that the personal computer essentially constitutes the test platform. The user interface can even be made to emulate familiar analog instrumentation controls using standardized Windows™ controls.

Measurement procedures with a PC require a high-quality D/A (digital to analog) converter for transforming the digital representation of a sine sweep created by the program into analog signal, and an equally good A/D converter for transforming the measured analog signals back into the digital domain for analytical purposes. Most sound cards provide at least 16-bit resolution, and some sound cards can even measure frequency response from DC to 20 kHz with an accuracy of ±0.25 dB and distortion as low as 0.003%.

The SoundCheck™ PC-based electroacoustical measurement system, essentially a software package, requires only a computer, sound card, amplifier, microphone, and microphone power supply. The SoundWare™ software features a family of "virtual instruments" that perform the functions of a signal generator, voltmeter, oscilloscope, spectrum analyzer, and real-time analyzer. The PC is rendered capable of measuring frequency, time, phase response, total harmonic distortion (THD), impedance, as well as performing other electrical tests. Such a system can be applied to evaluating loudspeakers, microphones, telephones, hearing aids, headsets, and other communication devices. Programming of test sequences allows the PC to be used not only for research and development work, but also in high-speed production testing and inspection procedures.

References

American National Standard Institute. 1966. *ANSI Standard Specification for Octave, Half-octave and Third-Octave Band Filter Sets*, S1.11-1966.

American National Standard Institute. 1978. *ANSI Standard Specification for Personal Noise Dosimeters*, ANSI S1.25-1978.

British Standards Institution. 1983. *Personal Sound Exposure Meters*, BS6402.

Cooley, J. W. and Tukey, J. W. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation* 19:297–301.

Crocker, Malcolm J. (ed.). 1997. *Encyclopedia of Acoustics*, Vol. 4. New York: John Wiley & Sons: Part XVII, pp. 1837–1879, and Part XVIII: 1933–1944.

Fraden, Jacob. 1993. *AIP Handbook of Modern Sensors: Physics, Design and Applications*. New York: American Institute of Physics, Chapters 3, 4, 9, and 11. (While this useful handbook covers devices for sensing temperature, light, humidity, and chemical processes, the chapters cited here refer to topics applicable to acoustic measurements.)

Harris, Cyril M. (ed.). 1991. *Handbook of Acoustical Measurements and Noise Control*, 3rd ed., New York: McGraw-Hill, Chapters 5, 8, 9, 11–15.

Hixson, Elmer L. and Busch-Vishniac, Ilene. 1997. In: Crocker, Malcolm J., ed. Transducer principles. *Encyclopedia of Acoustics*, Vol. 4. New York: John Wiley & Sons, Part VVIII, Chapter 59.

International Electrotechnical Commission. 1985. IEC 804: *Integrating-Averaging Sound Level Meters*. CH-1211 Geneva 20, Switzerland: International Electrotechnical Commission.

International Organization for Standardization. 1981. ISO 3744: *Engineering Methods for Free-Field Conditions over a Reflecting Plane*, CH-1211 Geneva 20, Switzerland: International Organization for Standardization.

International Organization for Standardization. 1977. ISO 3745: *Precision Methods for Anechoic and Semi-Anechoic Rooms*. CH-1211 Geneva 20, Switzerland: International Organization for Standardization.

International Organization for Standardization. 1975. ISO 3741: *Precision Methods for Broad-Band Sources in Reverberation Rooms*. CH-1211 Geneva 20, Switzerland: International Organization for Standardization.

International Organization for Standardization. 1975. ISO 3742: *Precision Methods for Discrete-Frequency and Narrow-Band Sources in Reverberation Rooms*. CH-1211 Geneva 20, Switzerland: International Organization for Standardization.

International Organization for Standardization. 1985. ISO 354: *Measurement of Sound Absorption in a Reverberation Room*. CH-1211 Geneva 20, Switzerland: International Organization for Standardization.

International Organization for Standardization. 1990. ISO 6926.2: *Requirements on the Performance and Calibration of Reference Sound Sources*. CH-1211 Geneva 20, Switzerland: International Organization for Standardization.

International Organization for Standardization. 1979. ISO 3746: *Survey Method*. CH-1211 Geneva 20, Switzerland: International Organization for Standardization.

Randall, Robert B. 1977. *Application of B & K Equipment to Frequency Analysis*, 2nd ed. Nærum, Denmark: Brüel and Kjær.

Sessler, G. M. and West, J. E. 1973. *Journal of the Acoustical Society of America* pp. 1589–1600.Temme, S. F. 1998. Virtual Instruments for Audio Testing Presented at the 105th Convention of the Audio Engineering Society. *Audio Engineering Society Preprint 4894* (*J*-8).

U.S. Department of Labor, Occupational Safety and Health Administration (OSHA). 1983. Occupational Noise Health Standard. *Code of Federal Regulations*, title 29, part 1910, sec. 1910.95 (29 CFR 1910.95), *Federal Register* 48:29687–29698.

U.S. Mine Safety and Health Administration (MSHA). 1982. *Code of Federal Regulations*. title 30, part 70, subpart F, and part 71, subpart I, *Federal Register* 47:28095–28098.

Wilson, Charles E.. 1989. *Noise Control*. New York: Harper & Row, Chapter 3 (probably the most outstanding chapter in this text).

Wong, George S. K. and Embleton, Tony F. W. 1995. *AIP Handbook of Condenser Microphones, Theory, Calibration, and Measurements*, New York: American Institute of Physics. (A great reference for those interested in the history of condenser microphones and in learning further details on their uses. Chapter 2 describes the classic Western Electric 640AA, the *de facto* standard for all microphones.)

Problems for Chapter 9

1. If a microphone has a sensitivity of −60 dB V/μbar and an output voltage of 21.6 mV is measured, what is the sound pressure level (SPL) responsible for that output voltage?

2. Determine the value of the output voltage of a microphone with a sensitivity level of -100 dB V/μbar if it is exposed to a sound pressure level of 85 dB?
3. A sound pressure level reading of 95 dB is taken at the standard conditions of 1 atmosphere and 68°F. What will be the readings at 20°F and at 95°F.
4. Some laboratory notes indicated that a noise level reading resulted in $L_{eq} = 95$ dB(A) and an SEL of 107 dB(A). How long did this noise level reading last?
5. Assuming a sixth octave analyzer could be made available, determine the first three center frequencies after 16 Hz. Also determine the bandwidth.
6. Calculate the lower and upper cutoff frequencies and the bandwidth for the octave band where the center frequencies are, respectively, $f_0 = 63$ Hz and $f_0 = 500$ Hz.
7. If it is desired to analyze noise in 10 Hz bandwidths with an uncertainty of 3%, what must be the necessary averaging time?
8. If we are sampling through the FFT process, a signal that ranges from 15 Hz to 30 Hz, what must be the minimum sampling frequency?
9. Why is a spectrum analyzer necessary and possibly more useful in dealing with noise machinery?
10. In an FFT analysis, it is desired to have a resolution of 1 Hz. If there are 1000 samples in the time series, what must be the minimum frequency sampling rate?

10
Physiology of Hearing and Psychoacoustics

10.1 Human Hearing

The mechanism of the human ear has been a source of much wonder for physiologists, who are progressing well beyond the fragmentary knowledge of the past by continually uncovering layers of new marvels of how the human ear really functions. Our hearing mechanism is a complex system that consists of many subsystems. The mechanism of the brain's processing of auditory stimuli is probably beginning to be understood, with the relatively recent discovery that hearing is metabolic in nature. Much of the pioneering work was performed by Georg von Békésy (1899–1972) who received the Nobel Prize in Medicine or Physiology for his investigations, particularly those entailing the frequency selectivity of the inner ear. His work indicated that the frequency selectivity ranked far poorer than the ear actually exhibits, but William Rhode found much greater selectivity in his work with live animals (von Békésy worked with dead animals, which most likely accounted for the difference). It is now realized that frequency selectivity of the inner ear fades within minutes after the metabolism ceases.

A young, healthy human is capable of hearing sounds over the frequency range of 20 Hz–20 kHz, with a frequency resolution as small as 0.2%, Thus, we can discern the difference between a tone of 1000 Hz and one of 1002 Hz. With normal hearing, a sound at 1 kHz that displaces the eardrum less than 1 Å (angstrom) can be detected, in fact, less than the diameter of a hydrogen atom! The intensity range of the ear spans extremes from threshold at which softest sounds can be detected to the roar of a fighter jet taking off, thus covering an intensity range of approximately 100,000,000–1. The ear acts as a microphone in the process of collecting acoustic signals and relaying them through the nervous system into the brain. The ear (cf. Figure 10.1) subdivides into three principal areas: the *outer*, *middle*, and *inner* ear.

The outer ear consists of a *pinna* that serves as a sound-collecting horn and the auditory canal that leads to the inner ear. The collected sound enters the ear through the opening (the *meatus*) into the auditory canal that forms a tube approximately 0.75 cm in diameter and 2.5 cm length. The canal terminates at the *tympanic membrane* (eardrum). Under the impetus of the sound the eardrum vibrates, causing

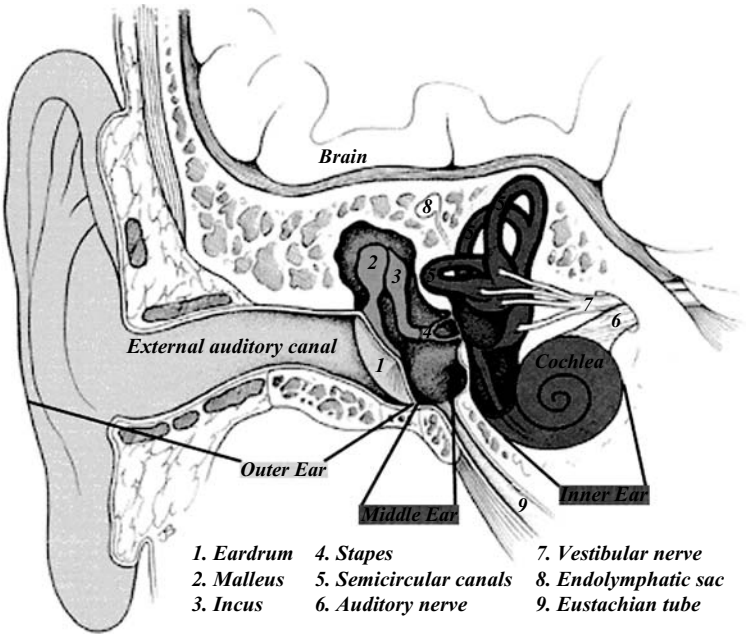


FIGURE 10.1. Coronal section of the right ear. (From Internet site of the Center for Sensory and Communication Disorders, Northwestern University, funding by U.S. National Institute of Health.)

three bones linked in an ossicular chain—namely, the *malleus* (or hammer), the *incus* (anvil), and the *stapes* (stirrup)—to oscillate sympathetically. At the lowest resonance (3 kHz) of the auditory canal, the sound pressure level at the eardrum is about 10 dB greater than it is at the entry into the canal. Because a resonance curve tends to be broad, human hearing tends to be more sensitive to sound in the range of approximately 2–6 kHz, as the consequence of the resonance being centered at 3 kHz. The diffraction of sound waves inside the head has the effect of causing the sound pressure level at the eardrum to exceed the free-field sound pressure level by as much as 20 dB for some specific frequencies.

The eardrum itself is a thin, semitransparent diaphragm that completely seals off the canal, marking the inner boundary of the outer ear and the outer boundary of the middle ear. This membrane is quite flexible at its center and is attached at its perimeter at the terminus of the auditory canal, thus demarcating the entrance to the middle ear. The middle ear, lined with a mucous membrane, constitutes an air-filled cavity of about 2 cm^3 in volume, which contains the three ossicles (bones), namely, the malleus, incus, and the stapes forming a bony bridge from the external ear to the middle ear. These bones are supported by muscles and ligaments. The malleus is attached to the eardrum; the incus connects the malleus and the stapes. The last bone in the chain, the stapes, covers the *oval window*. The *Eustachian tube*, which is normally closed, opens in the process of swallowing or yawning to equalize the air pressure on each side of the eardrum; this is a tube approximately

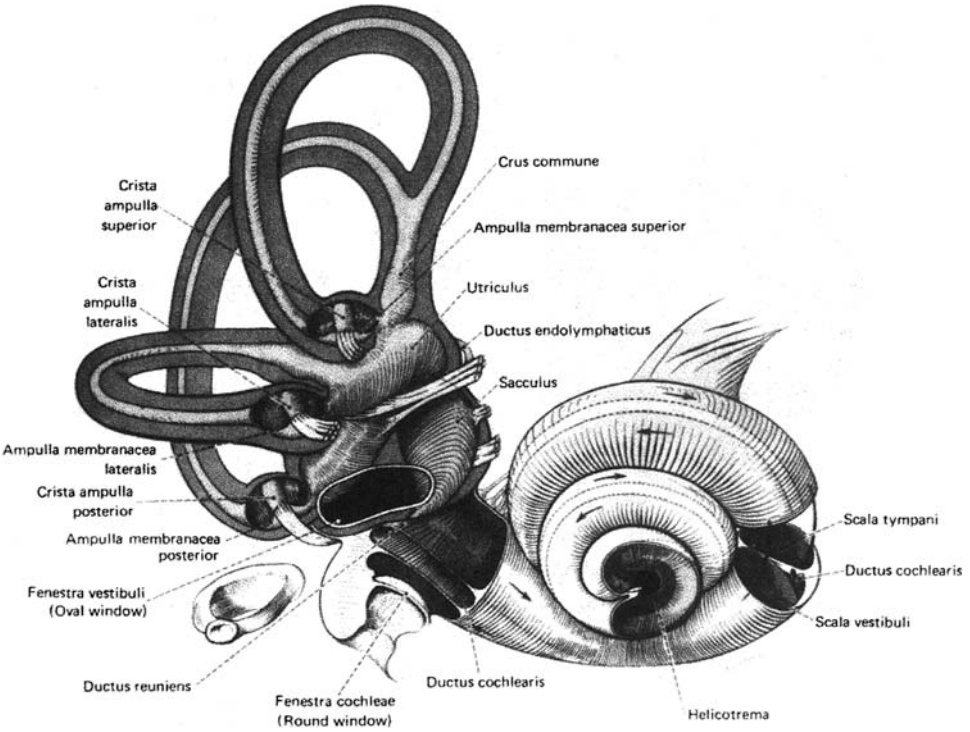


FIGURE 10.2. The membranous semicircular canals showing the cristae within the ampullae. (From "The Internal Ear," *What's New*, 1957, Abbot Laboratories. Reproduced with permission of the publisher.)

37 mm in length that connects the middle-ear cavity with the pharynx at the rear of the nasal cavity.

Just below the oval window lies another connection between the middle and inner ears, the membrane-covered *round window*. Between the oval and round windows is a rounded osseous projection, formed by the basal turn of the cochlea, called the *promontory*. A canal encasing the facial nerve is situated just above the oval window.

The structures to the right of the oval and round windows shown in Figure 10.1 are collectively called the *inner ear* (also called *labyrinth*), which comprises a number of canals hollowed out of the petrous portion of the temporal bone. These interconnecting canals contain fluids, membranes, sensory cells, and nerve elements. Three principal parts exist in the inner ear: the *vestibule* (an entrance chamber), the *semicircular canals*, and the *cochlea*. The vestibule connects with the middle ear through the oval and the round windows. Both of these windows are effectively sealed, by the action of the stapes and its support on the oval window and the presence of a thin membrane in the round window, thus preventing the loss of the liquid filling the inner ear. The semicircular canals play no role in the process of hearing but they do provide us with a sense of balance. The *cochlea*,

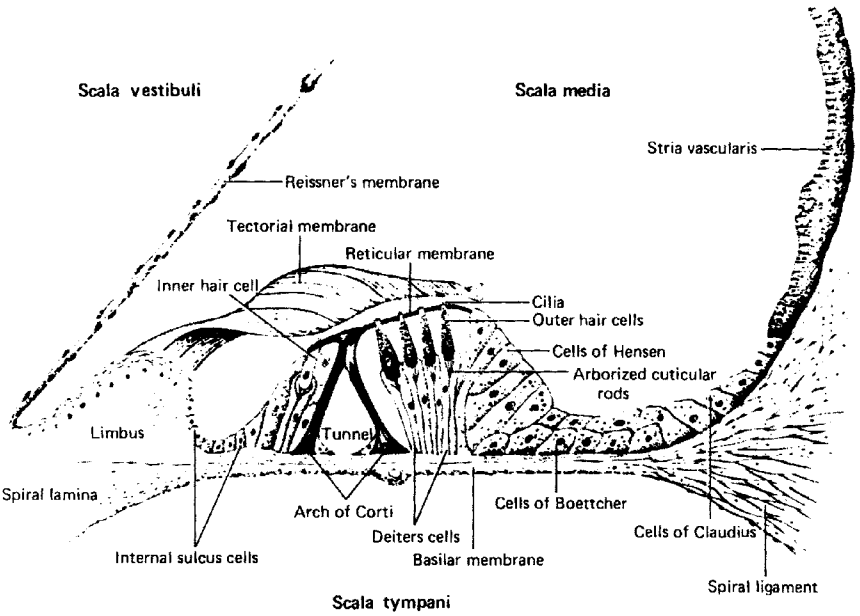


FIGURE 10.3. Cross section of the organ of Corti in the cochlear canal. (From H. Gulick. 1971. *Hearing. Physiology and Psychophysics*. New York: Oxford University Press. Reproduced with permission of the publisher.)

shown in enlarged detail in Figure 10.2, is the sensory system that converts the vibratory energy of sound into electrical signals to the brain for the detection and interpretation of that sound. The cochlea can be described as a 3.5-cm long tube of roughly circular cross section, wound about $2^1/_2$ times in a snail-like coil. This tube's cross-sectional area decreases in a somewhat uneven manner from its base to its apex. Its total volume is about $5\ (10)^{-2}\ cm^3$.

The coils of the cochlea surrounds an area called the *modiolus*; and the membranous labyrinth of the cochlear sector of the inner ear divides into three ducts or galleries (*scalae*). The *cochlear duct* (*ductus cochlearis*) runs the length of the spiraling cochlea, and because it occupies the central portion of the cochlea's interior, it has been termed the *scala medi* (i.e., the middle gallery), whose walls effectively partition the cochlea into two longitudinal channels, the *scala vestibuli* (or *upper gallery*) and the *scala tympani* (*lower gallery*). The only communication between the two galleries occurs through the *helicotrema*, a small opening at the apex of the cochlea. The other ends of the upper and lower galleries terminate in the oval and round windows, respectively.

Figure 10.3 shows an enlarged view of the cochlear duct. This duct is bounded by Reissner's membrane, the basilar membrane, and the *stria vascularis*. The basilar membrane extends from the bony *spiral lamina*, a ledge extending from the central core of the cochlea, to the *spiral ligament*. The length of the basilar membrane is about 32 mm long, from the base to the apex of the cochlea; the width varies

from about 0.05 mm at the base to about 0.5 mm at the apex; and the membrane gradually becomes thinner as it nears the apex. Positioned on the basilar membrane is the *organ of Corti*. This organ, shown in detail in Figure 10.4, consists of some structural cells (e.g., *Dieter's cells* and *Hensen's cells*), the rods and tunnels of Corti, and two types of hair cells on top of which lies the *tectorial membrane*. The tunnel of Corti, isolated from the endolymph, contains the fluid *cortilymph*.

The hair cells constitute the sensory cells for hearing. The *inner hair cells* are arranged in a single row on the modiolar side of the tunnel of Corti, and the *outer hair cells* exist in three parallel rows on the strial side of the tunnel of Corti. The inner hair cells are round and squat; their upper surfaces contain about 50–70 hairs called *stereocilia*. The outer hair cells, which look more reedlike than do the inner hairs, contain about 40–150 stereocilia arranged in a W-shaped pattern. There are 3000–3500 ciliated cells in the single row of inner hair cells and a total of 9000–12,000 ciliated cells in the three rows of the outer hair cells. The hair cells are connected to some 24,000 transverse nerve fibers in a complex network leading into the central core of the cochlea. The nuclei of these nerve fibers form the *spiral ganglion*, which unite to form the *cochlear branch of the VIIIth nerve*.

The cochlear branch joins with the vestibular branch to form the *VIIIth cranial nerve*, also called the *auditory* or *vestibulocochlear nerve*. The VIIIth cranial nerve along with the VIIth (facial) cranial nerve proceeds in a helical fashion through the *internal auditory meatus* to nuclei in the brain stem. From the brain stem the auditory pathway extends through various nuclei to the cerebral cortex in the temporal lobes of the brain.

The VIIIth cranial nerve is primarily as sensory nerve, i.e., it conveys sensory information from the cochlea and the vestibular system to the brain.

10.2 The Mechanism of Hearing

Sound waves are directed by the pinna into the auditory canal. The longitudinal changes in air pressure of the sound wave propagate to the eardrum, causing it to vibrate. Because the handle of the malleus is imbedded in the eardrum, the ossicular chain is set into vibration. These tiny bones vibrate as a unit, elevating the energy from the eardrum to the oval window by a factor of 1.31 to 1. Sound energy is further enhanced by the difference in area between the eardrum and the stapes footplate by a factor of approximately 14. Multiplying this effective areal difference of 14 by the lever action of the ossicular chain (1.31) yields an energy increase of 18.3 to 1, which translates into an amplification factor of 25.25 dB on the sound pressure level (SPL) scale. The middle ear acts as a transformer, by changing the energy collected by the eardrum into greater force and less excursion, thereby matching the impedance of the air to the impedance of the inner ear's fluid.

Because the fluid of the inner ear is virtually incompressible, provision for relief of the pressure produced by the movement of the footplate of the tapes is provided by the interaction of oval window and the round window, with the fluid motion from the oval to the round window being transmitted through the cochlear duct.

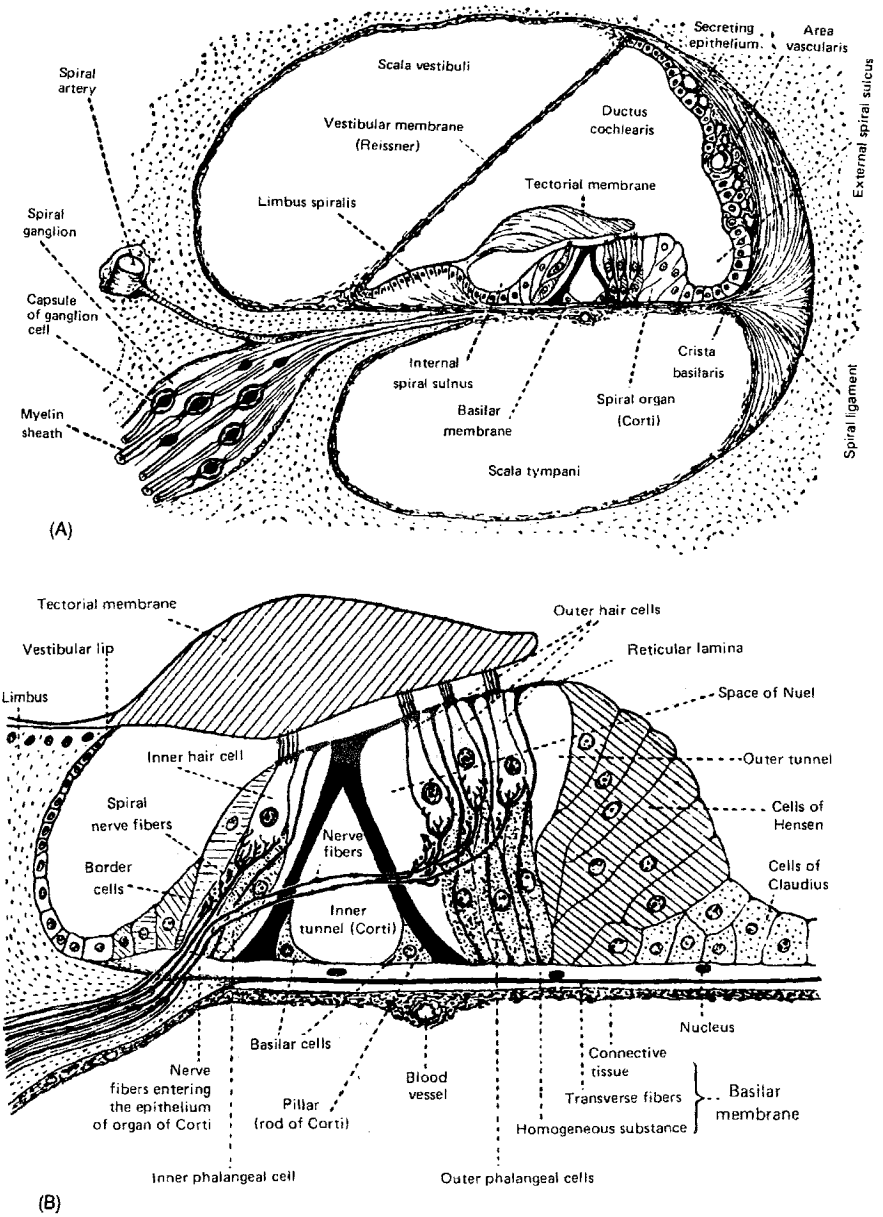


FIGURE 10.4. A cross section of the organ of Corti: (A) low magnification; (B) higher magnification. (From A. T. Rasmussen, 1947. *Outlines of Neuro-Anatomy*, 3rd edition. Dubuque, IA: Wm. C. Brown Company.)

When the footplate of the stapes pushes into the perilymph of the scala vestibuli, the vestibular membrane (or membrane of Reissner) bulges into the cochlear duct, causing movement of the endolymph within the cochlear duct and displacement of the basilar membrane. Von Békésy's experiments with cochlear models led him to formulate a theory that the displacement of the basilar membrane is in the form of a traveling wave that proceeds from the base to the apex of the cochlea. The maximum amplitude of the wave occurs at a point along the basilar membrane corresponding to the frequency of the stimulus, i.e., each point of the basilar membrane corresponds to a specific value of the simulating frequency.

The cilia of the hair cells are embedded in the gelatinous tectorial membrane, so that when the basilar membrane is displaced, it generates a "shearing" force on the cilia. The sidewise motion of the cilia creates an alternating electrical current, also referred to as the *cochlear microphonic* (CM), *cochlear potential*, or the *Wever-Bray effect*[1]. The deflection of the hair cells triggers responses in the neurons connected to the hair cells. Impulses are borne along the nerve fibers to the main trunk of the cochlear portion of the VIIIth nerve and onward to the brain. This is how the cerebral cortex eventually "hears" the vibrations that strike the eardrum.

Research over the past decade indicates that the cochlea does not act passively. Active processes occur that indicate that energy is being added to the cochlea through mechanisms that are not yet fully understood. More energy is contained in the cochlea than that from the sound going into it. One phenomenon that has been identified is that of *otoacoustic emission*, which is a sound in the external ear canal believed to have originated from vibrations within the cochlea and propagated back through the middle ear. Otoacoustic emissions can be measured by placing a miniature microphone in the ear canal. A spontaneous otoacoustic emission is identified as a constant low-level sound that occurs spontaneously in half of normal ears. When a high-level click is introduced to the ear, an *evoked otoacoustic emission* occurs some 5 msec later as a low-level sound. Also, when two different tones are presented at high levels to a normal ear, the otoacoustic emission occurs in the form of new tones generated at frequencies other than the two original frequencies. These new tones are termed *distortion products*.

When the sound striking the eardrum is sufficiently loud, the middle ear muscles contract reflexively. This *acoustic reflex* occurs as a contraction in the stapedius muscle, which results in a pull against the ossicular chain and a reduction in the energy transmitted through the oval window into the perilymph in the vestibule. The largest amount of reduction in sound due to acoustic reflex—ranging approximately from 20 to 30 dB—occurs for low frequencies. Above 2 kHz this acoustic reflex is fairly negligible.

As we mentioned previously, a line of hair cells in the organ of Corti senses sense the movements of basilar membrane. Each hair cell holds fine rods of protein, i.e., the stereocilia, hinged at their respective ends. When the basilar membrane

[1] So named after the two investigators who discovered in 1930 that the speech delivered to a cat's ear could be understood when the CM signal was picked up from the cochlear nerve and amplified.

moves, these stereocilia sway under the impact. The deflections initiate a chain of electrochemical events that generate electrical spikes or *action potentials* in the cells of the spiral ganglion. These signals are transmitted to a relay station called the *cochlear nucleus* in the brain stem. The information flows through other nuclei along the path to the auditory cortex, the portion of the brain that processes auditory information.

Within this cellular circuitry, the frequency of a sound is encoded by two mechanisms. The first mechanism is a place code that indicates the location along the tapered basilar membrane that moves the greatest distance. Stereocilia on the hair cells respond to this displacement and produce action potentials among the nearest spiral-ganglion neurons. The other mechanism is a temporal code that is created when neurons become synchronized (or *phase-locked*) into the period of an acoustic wave. In normal hearing, neural responses can readily match frequencies up to about 1 kHz, but this phase-locking capability declines progressively at higher frequencies. It is conceivable that perception of frequency is based on some combination of phase and temporal orders, with the temporal code being effective only at lower frequencies.

10.3 Hearing Loss

Nearly a quarter of the population between the 15 and 75 years of age suffer hearing impairment. Impaired hearing, which is often caused by infectious diseases or overexposure to loud noise or simply the process of aging, is common enough to be on a par with the onset of poor vision. When hearing loss occurs in early childhood, its consequences become more obvious than when it occurs in adulthood. A child's progress in learning and developing social relationships may be hindered and the child may even be deemed "not too bright" if professional help and guidance are not forthcoming. The primary problem of hearing loss, regardless of the age of the affected individual, is a diminution of a person's ability to understand speech.

Even milder forms of hearing loss early in life can generate great difficulty, particularly for children who developed within normal limits but are not doing well in school, due to their being inattentive. Such moderate hearing losses are not uncommon and may even be on the increase due to heightened exposure to "rock" music. When a mild hearing loss is corrected, the child often becomes "like a different person." Fortunately, many of the hearing impaired can be helped through the use of hearing aids.

The gradual waning of hearing loss affects adults in a more underhand manner. Most people with age-induced or noise-induced hearing impairment first lose hearing acuity at high frequencies, making it difficult for them to distinguish consonants, especially s versus f, and t versus z. Such persons must strain harder to understand conversations. Going to the movies, listening to lectures, conversing with friends and other pleasures become stressful chores. This can result in an individual's becoming withdrawn from his friends and relatives. Some of these

patients can be helped through counseling and rehabilitation, but no cure exists for most cases of sensorineural deafness.

Hearing loss falls into two principal categories: *conductive* and *sensorineural*. Conductive hearing loss occurs from any condition that impedes the transmission of sound through the external or the middle ear. Sound waves are not transmitted effectively to the inner ear because of some blockage in the auditory canal, interference in the eardrum, the ossicular chain, the middle ear cavity, the oval window, the round window, or the Eustachian tube. For example, damage to the middle ear, which carries the task of transmitting sound energy effectively, or to the Eustachian tube, which sustains equal air pressure between the middle ear cavity and the external canal, may result in mechanical deficiency of sound transmission. In pure conductive hearing loss, no damage exists in the inner ear or the neural system. Conductive hearing losses are generally treatable.

Sensorineural deafness, which is a far more accurate term than the ambiguous terms "nerve deafness" and "perceptive deafness," describes the effect of damage that lies medial to the stapedial footplate—in the inner ear, the auditory nerve, or both. In the majority of cases sensorineural deafness is not curable. The term "sensory hearing loss" is applied when the damage is localized in the inner ear. Applicable synonyms are "cochlear" or "inner-ear" hearing loss. "Neural" hearing loss is the proper terminology to describe the result of damage in the auditory nerve proper, anywhere between its fibers at the base of the hair cells and the auditory nuclei. This category also encompasses the bipolar ganglion of the eighth cranial nerve.

Mixed hearing loss results from conductive hearing loss accompanied by a sensory or a neural (or a sensorineural) in the same ear. Otologic surgery may help in cases of mixed hearing loss in which the loss is primarily conductive accompanied by some sensorineural damage of a lesser degree.

Functional hearing loss, which occurs far less frequently than the hearing loss types described above and presents a greater diagnostic challenge to clinics, denotes the condition in which the patient does not seem to hear or to respond, yet the handicap cannot be attributable to any organic pathology in the peripheral or the central auditory pathways. The basis for this type of hearing difficulty may be caused by entirely emotional or psychological etiology. Psychiatric or psychological therapy may be called for, rather than otological treatment.

Central hearing loss, or *central dysacusis*, remains mystifying to otologists, although information about this type of hearing defect is accumulating. Patients suffering this type of condition cannot interpret or understand what is being said, and the cause of the difficulty does not lie in the peripheral mechanism but somewhere in the central nervous system. In central hearing loss, the problem is not a lowered pure-tone threshold but the patient's ability to interpret what he or she hears. It is obviously a more complex task to interpret speech than to respond to pure-tone signals; consequently, the tests necessary to diagnose central-hearing impairment must stress measuring the patient's ability to process complex auditory information. It requires an extremely skilled, intuitive otologist to make an accurate diagnosis.

10.4 Characteristics of Hearing

If the sound is audible, the amplitude of the sound is said to be *above threshold*; and if the sound is inaudible, the amplitude is considered to be *below threshold*. The amplitude of the sound at the transition point between audibility and inaudibility is defined as the *threshold of hearing*. When sound amplitude exceeds threshold, the sound is processed and perceived as having certain qualities including loudness, pitch, and a variety of other perceptive traits such as information. The study of auditory perception in relation to the physical characteristics of sound defines the field of *psychoacoustics*.

Sensitivity

The ear is not equally sensitive to all frequencies. The absolute sensitivity of the ear, defined by its threshold, depends on a variety of factors, the most important of which is the sound pressure level and the frequency of the sound. The resonance of the ear canal, the level effect of the ossicles, and the difference between the surface area of the eardrum and that of the stapes footplate all affect the intensity of the sound that actually penetrates the cochlea.

An *audiometer* which generates signals of varying frequency and intensity is used to measure an individual's hearing sensitivity. The signals produced by the audiometer can be directed either to earphones or to a loudspeaker in an anechoic chamber. As it is far more difficult to ascertain the intensity of the sound at the level of the cochlea, and such a determination would not accurately represent how well an individual hears under normal circumstances, we generally specify hearing sensitivity in terms of thresholds for sounds of various frequencies of which sound pressure levels were determined in a sound field without the listener present. Figure 10.5 maps the hearing sensitivity of the normal young human ear over a range of frequencies. The solid curve, referred to as the *minimum audible field*, or MAF, describes the minimum intensities that can be detected when the listener is positioned before a loudspeaker at a prescribed distance. Both of the listener's ears are stimulated simultaneously by the sound source (i.e., the loudspeaker).

However, most clinical work in audiology entails measurements in reference to a single ear rather than to both ears. This is usually performed by directing the test signals to the appropriate earphone of a headset rather than to a loudspeaker. The use of a headset as opposed to exposure to a loudspeaker considerably modifies the listening situation. For example, the resonant frequency of the ear canal is shifted because both ends of the canal are sealed in contrast to the situation when the canal is open to the sound field. Moreover, the placement of the earphones may give rise to unwanted physiological noise that can interfere with the detection of low-frequency sounds. Also, the method of calibrating sound from a loudspeaker differs from that for calibrating sound from an earphone. Because of these differentiating and other factors, the measurement of thresholds through the use of headphones is called *minimum audible pressure*, or MAP. The MAP measurements are contrasted with

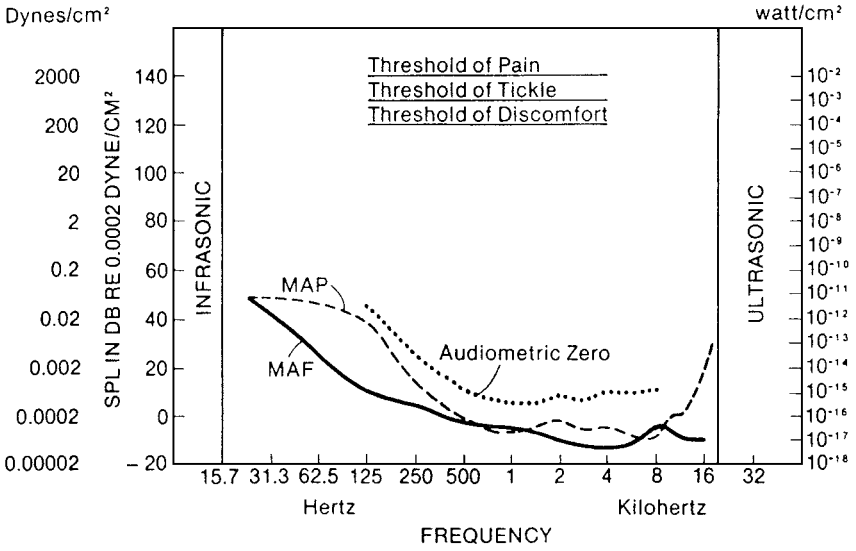


FIGURE 10.5. The area of human audibility. The two lower curves represent the lowest (best) thresholds of hearing of young adults. The solid curve is the minimum audible field (MAF) and the dashed curve is minimum audible pressure (MAP). (From *American National Standard Specification for Audiometers*, ANSI S3.6–1969, revised 1989, American National Standards Institute. New York: Acoustical Society of America.) The dotted curve represents the current standard for the audiometric zero. The upper three lines represent averages for sensations of discomfort, tickle, and pain. The ordinates define intensity in terms of pressure in dynes/cm^2, sound pressure level in dB, and power flow in W/cm^2.

MAF measurements in Figure 10.5. The threshold curve for the MAP condition appears to be several dB higher (i.e., showing lower sensitivity) than the MAF curve, a situation referred to as the "missing 6 dB." This can be attributed to the fact that using both ears in a sound field enhances sensitivity, in contrast to listening with only one ear under an earphone, and other factors occur such as the diffraction of sound around the head in a sound field, the different resonances of the external ear canal, and so on.

The two curves of Figure 10.5 represent thresholds that are two standard deviations below the mean, i.e., the curves represent the thresholds of approximately 2.5% of young adults (16–25 years of age) determined by examination to be otologically normal. These curves are based on data given in two separate studies performed 4 years apart conducted by the National Physical Laboratory in Great Britain. Tables 10.1 and 10.2 list the mean and standard deviations reported in these two studies and the data points (two standard deviations below the means) on which the MAP and MAF curves are based.

The intensities defining an audiometric zero at each of the standard frequencies on a pure-tone audiometer are represented by the dotted curve of Figure 10.5. These intensity values were established by international agreement among scientists as

TABLE 10.1. The Means and Standard Deviations and the Data Points (Two Standard Deviations Below the Means) on Which MAP Curves of Figure 10.5 Were Based.

Frequency, Hz or kHz	Mean SPL, dB	σ, dB	SPL at 2σ below mean	ANSI, 1969
80 Hz	61.0	8.0	45.0	
125	45.5	6.8	31.9	45.5
250	28.0	7.3	13.4	24.5
500	12.5	6.5	−0.5	11.0
1 kHz	5.5	5.7	−5.9	6.5
1.5	8.5	6.1	−3.7	6.5
2	10.5	6.1	−1.7	8.5
3	7.0	5.9	−4.8	7.5
4	9.5	6.9	−4.3	9.0
6	10.5	9.1	−7.7	8.0
8	9.0	8.7	−8.4	9.5
10	17.0	9.0	−1.0	
12	20.5	9.6	1.3	
15	39.0	10.7	17.6	
18	74.0	21.9[a]	30.2	

[a] Calculated from reported standard of error of the mean. Data on mean sound-pressure levels and standard deviations (σ) for 80 Hz through 15 kHz taken from Dudson and King (1952). The data for 18 kHz were taken from Harris and Myers (1971).

TABLE 10.2. The Means and Standard Deviations and the Data Points (Two Standard Deviations Below the Means) on Which the MAF Curves of Figure 10.5 Were Based.[a]

Frequency, Hz or kHz	Mean SPL, dB	σ, in dB	SPL at 2σ below mean
25 Hz	63.5	8.0	47.5
50	43.0	6.5	30.0
100	25.0	5.0	15.0
200	15.0	4.5	6.0
500	5.5	4.5	−3.5
1 kHz	4.5	4.5	−4.5
2	0.5	5.0	−9.5
3	−1.5	6.0	−13.5
4	−5.0	8.0	−21.0
6	4.5	8.5	−12.5
8	13.5	8.5	−3.5
10	16.5	11.5	−6.5
12	13.0	11.5	−10.0
15	24.5	17.0	−9.5

[a] Data on mean sound-pressure levels and standard deviations (σ) taken from Robinson and Dudson (1956).

being representative of the average minimum audible sound pressure levels for young adult ears and have been incorporated in the standards for audiometric calibration throughout most of the world. This standard as the result of being adopted in 1969 by the American National Standards Institute, is referred to as *ANSI-1969 standard* and it is listed with the MAP values in Table 10.1 (American National Standards Institute, 1969).

Loudness

While the sensation of loudness correlates to the amplitude of the sound above threshold, loudness is not perceived by the human ear in equal measures as the amplitude increased over different frequencies above the threshold. Individual judgment constitutes the deciding factor in ascertaining the degree of loudness. This had led to the development of equal loudness contours which are curves connecting SPL points of equal loudness for a number of frequencies, as judged by tested listeners. These curves, also called *phon*[2] *curves*, are constructed by asking subjects to judge when tones of various frequencies are considered equal in loudness to a 1-kHz tone at a given SPL. The official definition (ANSI, 1973) of the phon specifies binaural (two-ear) listening to the stimuli in a sound field (American National Standards Institute, 1986). Equal-loudness contour curves are given in steps of 10 phons in Figure 10.6, with the dashed MAF curve from Table 10.2 included in the plot as a threshold reference.

As an example of how humans perceive sound, consider a 30-Hz tone at 95 dB SPL. It would be judged by a typical listener as being as equally loud as a 1000-Hz tone at 70 dB SPL or a 5000-Hz tone at 65 dB SPL. As sound is steadily increased in intensity above the threshold, it will eventually cause the listener to experience physiological discomfort. A further increase in the intensity produces a tickling sensation in the ear, and an additional increase in intensity causes the listener to experience pain. These three levels constitute, respectively, the *thresholds of discomfort*, *tickle*, *and pain*, which are represented by the upper three lines of Figure 10.5. While these threshold values of 120, 130, and 140 dB SPL represent statistical averages for young adult ears, different individuals have different tolerance thresholds, but these values do not differ markedly from the statistical averages. Thus, in Figure 10.5, the region between the discomfort and the audiometric zero constitutes the usable dynamic range of hearing for humans.

Pitch

The sensation of pitch is obviously related to the frequency of the tone. The actual pitch of a sound is affected by other factors, including the sound pressure level and the presence of component frequencies. Pitch perception is a complex process, one that is not yet fully understood. Pitch elicited by some sounds may evoke the same

[2] A phon is a unit of loudness that, at the reference frequency of 1 kHz, is equated to the decibel scale.

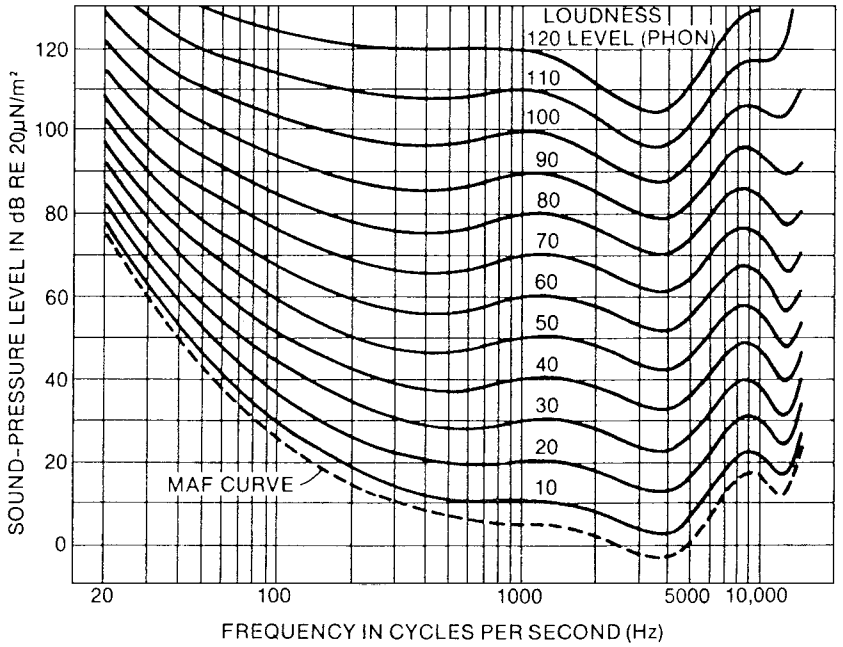


FIGURE 10.6. Equal loudness contours. (From Peterson, A. P. G. and E. E. Gross. *Handbook of Noise Measurement*. 1980. Concord, MA: General Radio.)

aural response whether or not the fundamental frequency is present. The average adult male voice carries a fundamental between 120 and 150 Hz and that of a typical adult female lies between 210 and 240 Hz. Yet we generally find it easy to distinguish between male and female voices even though the telephones does not transmit frequencies much lower than 300 Hz. Somehow we are able to atone aurally for the fundamental frequency missing from the signal passing through the telephone receiver.

The spectrum of a sound generates a psychological sensation of quality. This permits us to distinguish the difference, say, between a trumpet and an English horn playing the same note. This is because of the differences in their respective sound spectra (i.e., the frequency content or the presence of overtones), which, in turn, are functions of the complex vibrations and the resonance modes inherent in their respective structures. We are also able to discern different speech sounds because of the differences in the sound spectra. Even over the telephone, individual voices are recognizable because of the differences in their sound spectra.

Masking

Masking is said to have occurred when the audibility of a sound is interfered with by the presence of noise or other background sound. The "cocktail party" effect,

which makes it difficult to carry on a private conversation against a backdrop of other people's chatter, is a familiar example of masking. Speech becomes unintelligible by the presence of excessive background noise. Although masking is almost always undesirable, broadband noise may be purposely introduced into an office environment to make conversation unintelligible to potential eavesdroppers in an adjacent office. Because most of the intelligence in speech is generally contained in the frequency range between 200 Hz and 6 kHz, noise in that frequency range is most objectionable in terms of speech masking. But excessively loud noise in any frequency band can adversely affect speech intelligibility by causing such an overload of the auditory system that one cannot effectively discriminate speech from the prevailing total signal. Consonants essential to conveying verbal information tend to be pronounced softly, so they become readily indiscernible in the presence of noise.

A person speaking normally produces an unweighted sound level of 55–70 dB at 1 m. It is more taxing for that person to speak more loudly for a sustained time. A typical maximum voice effort, in the form of a shout, produces about 90 dB at 1 m. Speech intelligibility generally improves when the speaker and the listener are near each other and if the speaker increases the signal-to-noise (S/N) ratio by talking louder. Maximum intelligibility usually can be obtained if the unweighted level of the speech is between 50 and 75 dB at 1 m from the speaker. Speaking more loudly does not always guarantee greater intelligibility, even though the S/N ratio (defined as the intensity of the signal divided by the intensity of the noise) may be increased, because the formation of speech sound above 75 dB may degrade sufficiently that there is little or no improvement in intelligibility. If a listener is familiar with the words and the dialect used, intelligibility will be greater. It is for this reason that critical communications, particularly those of air controllers, are based on a limited vocabulary. In ordinary face-to-face conversation, the listener has the additional luxury of making out the context of the words by observing the speaker's facial expressions and gestures.

10.5 Prediction of Speech Intelligibility: The Articulation Index

In order to assess the effect of noise on speech communication, it is necessary to conduct speech-intelligibility tests with actual speakers and listeners in the presence of interfering noise. The test materials may be sentences, digits, disyllabic words, monosyllabic words, or nonsense syllables. The listeners are scored according to the percentage of the speech materials heard correctly. The background interfering noise is generally recorded and played back in the testing laboratory.

From such experiments came the realization that speech intelligibility is a function of the intensity and the frequency characteristics of the interfering noise. Regarding the S/N ratio, Licklider and Miller stated that the S/N should exceed

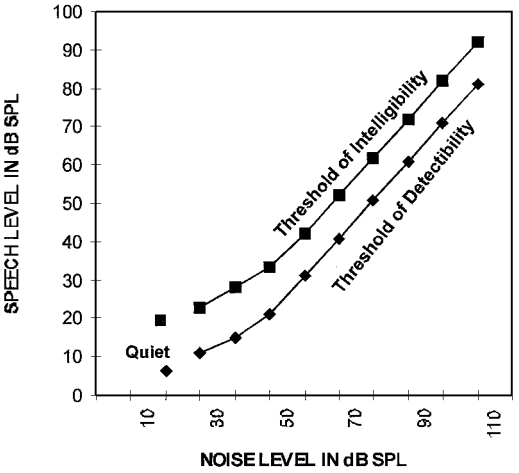


FIGURE 10.7. The effect of white noise on the thresholds of detectability and intelligibility of running speech (Hawkins and Stevens, 1950).

6 dB for satisfactory communication, although the presence of speech may be detected for S/N as low as −18 dB. If the intensity of the signal (speech) exceeds the noise, the sign of the S/N value is plus; conversely, a negative value of S/N indicates that the noise is more intense than the signal.

Figure 10.7 maps the effect of white noise on the thresholds of detection and intelligibility of running speech. According to this figure, which was developed by Hawkins and Stevens based on extensive tests making use of running speech and white noise (Hawkins and Stevens, 1950), the threshold of intelligibility occurs when the level of the speech exceeds the noise level by about 6 dB (S/N of 6 dB). As the sound pressure level of the noise is increased above this value, the threshold of intelligibility is proportionally increased so that the S/N value of −6 dB remains fairly constant over a wide range of intensities. For other kinds of speech materials and different masking noises, the relationship between the threshold of intelligibility and the level of the interfering noise may not necessarily remain the same. At an S/N of −18 dB, running speech can be detected but not understandable.

Other but simpler methods have been developed for measuring the effect of interfering noise on the intelligibility of speech. A principal method of predicting speech intelligibility is the *articulation index*, or AI, which is a value that ranges from 0.0 to 1.0 and represents the proportion of the speech spectrum that occurs above the noise. French and Steinberg of the Bell Laboratories developed the concept of articulation index on the basis of the assumption that most of the intelligence in speech is contained in the frequency bands between 200 and 6100 Hz. The articulation index can be calculated from the levels of the masking signal and the speech level in the frequency bands. The contribution of each frequency band to speech intelligibility is defined as 12 dB plus the sound level of the speech

TABLE 10.3. Weighting Factors as a Function of Center Frequency for One-Third Octave Band-Based Calculation of Articulation Indexes. Speech Levels Given in the Second Column are Those for a Typical Male Voice at 1 m.

Center frequency (Hz)	Speech level (+12 dB)	Weighting factor
200	67	4
250	68	10
315	69	10
400	70	14
500	68	14
630	66	20
800	65	20
1000	64	24
1250	62	30
1600	60	37
2000	59	37
2500	57	34
3150	55	34
4000	53	24
5000	51	20

less the masking level. Each frequency band's contribution is limited to the range between 0 and 30 dB. The sound level of the speech signal is based on a long-term energy average in each frequency band, and each frequency band contribution is multiplied by a weighting factor. The sum of the weighted contributions divided by 10,000 yields the AI.

Table 10.3, based on the division of the speech spectrum into one-third octaves, provides the data necessary to calculate AI. The first column lists the center frequency of the one-third octave, the second column gives the typical male voice long-term average speech level plus 12 dB at 1 m distance. The weighting factor for each one-third octave band is listed in the third column. If the masking noise has been measured only in full octave bands, Table 10.4 may be used instead of Table 10.3.

TABLE 10.4. Weighting Factors for One Octave Band-Based Calculation of Articulation Indexes. This is for the Typical Male Voice Level at 1 m.

Center frequency (Hz)	Speech level (+12 dB)	Weighting factor
250	72	18
500	73	50
1000	78	75
2000	63	107
4000	58	83

TABLE 10.5. Calculation for the Articulation Index in the Sample Problem.

Center frequency (Hz)	Speech level (SL) (+12 dB)	Weighting factor (WF)	Noise (N)	SL − N = DIFF	WF × DIFF
250	72	18	47	25	450
500	73	50	47	26	1300
1000	78	75	47	30[3]	2250
2000	63	107	47	16	1712
4000	58	83	47	11	913

Example Problem 1: Calculation of AI

Let us compute the articulation index of a male voice speaking at a normal level 1 m from the listener in the presence of pink noise that contributes 47 dB in each octave band.

Solution

Table 10.4 is used for this calculation. The 47-dB octave-band noise level is subtracted from the values in the second column; and the difference (up to a maximum of 30 dB) is multiplied by the weighting factors of the third column. The resulting weighted contributions are added and divided by 10,000, yielding the articulation index. Table 10.5 below gives details of the calculations, with the values given in the fifth column produced by subtracting 47 dB from the speech level of the second column, and each of the values in the fifth column is multiplied by the weighting factor of the third column to yield the figures listed in the sixth column.

$$\text{Articulation index (AI)} = (450 + 1300 + 2250 + 1712 + 913)/10000 = 0.6625$$

The articulation index is 0.6625, or 66.25%.

10.6 Speech-Interference Level (SIL)

Measurements to obtain data for articulation indexes require special laboratory equipment for determination of S/N in a number of frequency bands. A simpler procedure for estimating the effect of noise on verbal communication makes use of octave-band levels as measured in a typical noise survey. The parameter that is called the *speech-interference level*, abbreviated SIL, can be obtained by computing the arithmetic average of octave-band levels in the three octave bands of 600–1200, 1200–2400, and 2400–4800 Hz. However, the current practice uses the arithmetic level in the "preferred" octave bands with center frequencies at 500, 1000, and 2000 Hz. Speech-interference level defined thusly is referred to as *PSIL*.

[3] The value of the following expression (Speech Level + 12 dB—Noise Level) must fall between 0 and 30.

TABLE 10.6. PSIL (in dB) at Which Effective Speech Communication is Barely Possible.

	Normal		Raised		Very Loud		Shouting	
Distance (m)	M	F	M	F	M	F	M	F
0.3	68	63	74	69	80	75	86	81
1	58	53	64	59	70	65	76	71
2	52	47	58	53	64	59	70	65
3	48	43	54	49	60	55	66	61
4	46	41	52	47	58	53	64	59

The speech-interference level (PSIL = 68 dB) has been identified as the level at which reliable speech communication is barely possible in a normal male voice at a distance of 0.3 m (or 1 ft) outdoors. If a male speaker talks in a raised voice, a very loud voice, or in a shout, the speech interference levels have been identified, respectively, as PSIL = 74, 80, and 86 dB. A female speaker, on the average, has PSIL levels 5 dB less than the corresponding values for a male. Table 10.6 lists the PSIL (in dB) at which effective speech communication is barely possible. The table is based on minimally reliable communication, at which about 60% of the communication of uttered numbers and words out of context can be discerned. In order to roughly approximate PSIL in terms of dBA, 7 dB can be added to the values of PSIL.

Example Problem 2: SIL

Background noise levels for an industrial plant were measured to be 62, 65, and 74 dB, respectively, in the 500-, 1000-, and 2000-Hz center-frequency bands. What are the implications for speech interference at a distance between a speaker and a listener standing 1 m apart?

Solution

To solve this problem, the arithmetic average of the noise level in three bands are first determined. This will be $(62 + 65 + 74)/3 = 67$ dB. From Table 10.6 we establish that reliable speech is barely possible for a male, speaking in a raised voice or a female speaking in a very loud voice.

10.7 Prosthetics for Hearing

Hearing Aids

A conventional hearing aid works by amplifying sound and delivering that amplified sound to the eardrum. Sound is received by a miniature microphone that converts acoustic signals into electrical signals that is amplified and then relayed to

a receiver (i.e., a mini-loudspeaker) that converts the signals into amplified sound. Current hearing aids evolved from a long history of continuously improving technology, from the bulky electrical hearing aids of the 1930s (which were powered by rather large multiple batteries) to the smallest models inserted into the ear canal. The electronic part of a hearing aid can be based on different technologies: *analog*, *programmable*, and *digital*. Each of these technologies has advantages and disadvantages, but the choice rests on the optimum benefit to the user.

In the analog models, variable resistors perform the trimming adjustments, and any changes of resistor value will change the properties of the circuit accordingly, thus providing a change in the frequency response, usually prescribed by a qualified audiologist, to meet the needs of the user. In principle, the same applies to a programmable hearing aid. In this case the total variable resistor component consists of a ladder of resistors, each attached to a digital switching element. Setting the individual switches either "on" or "off," thereby bypassing some of the resistors in the ladder changes the total resistor value. These switches or *gates* are the fundamental building blocks of all digital circuits. CMOS are used to produce gates that function as long as there is electrical power. But in order to avoid losing the switch settings (and hence the adjustment), a common solution is to construct a secondary set of switches, using a technology that enables the switch settings to be retained even when there is no battery attached. This technology known as E2PROM (programmable read only memory) provides the nonvolatility necessary to retain the settings.

By their very nature digital adjustments are discreet, i.e., the resistor can only take on a finite number of values. Although there is theoretically no limit on how fine the resolution can be, the penalty is a larger circuit using more power. The resolution is therefore set to be sufficiently adequate for the parameter to be adjusted, typically in the region of 0.5–5 dB. Programmable hearing aids may also use digital switching to select functions (ON/OFF, M/T). The M/T function represents a choice between the microphone (M) and telephone (T) mode. In the telephone mode, induction signals are received from the telephone receiver rather than the acoustic signal, so as to provide better coupling and eliminate feedback. A digital volume control may be used to work in steps, typically 1 dB.

In digital hearing aids, an analog-to-digital converter reads the analog voltage signal from the microphone and converts it to a digital signal. The digital signal represents the original sound as a series of number, which then can be manipulated mathematically by the digital signal processor (DSP). Once digitized, the signal is more robust and is no longer subject to electronic noise and distortion. To obtain the highest possible signal quality the input signal is sampled at a very high rate (500 kHz–1 M Hz) before the digitization. The sampling rate at the input affects the signal quality, so as the sampling rate increases, so does the quality of the signal. Another important parameter for sound quality is the resolution, which determines the initial precision of the signal and the precision that later can be achieved during the mathematical manipulations that constitute signal processing. After the sampling, the converted signal is then ready for manipulation by a specially designed digital signal processor (DSP). The DSP is programmed by an audiologist on a

computer to provide the frequency equalization curve on the basis of the results from testing the patient's hearing. Algorithms in the DSP can continuously divide sounds into frequency channels and emphasize the higher frequencies containing vital consonant sounds in speech over the distracting rumble of low-frequency noise. Algorithms also manage noise: owing to the fact that speech sound intensities can change radically in a millisecond whereas noise is more acoustically stable over a comparatively longer time. On a time basis, DSP can reduce the levels of continuous sounds such as traffic noise and household appliances. And it simultaneously readjusts when changes occur, restoring amplification when shorter duration sounds are detected. In relatively quiet surroundings, the digital algorithm can detect the consistency of softer environmental sounds from ventilation systems and appliances, and it also automatically reduces amplification in the appropriate frequency range, immediately restoring proper levels when the sound pattern changes. Digital hearing aids carry the disadvantage of considerably higher costs and may not provide enough amplification as the other types of hearing aids.

Implantable hearing aids, referred to by the acronym, IHA, can be used by people having sensorineural hearing loss but a healthy middle ear. The IHA, which is still in the stages of final development, converts sound to vibrations inside the middle ear. This gives the IHA several advantages of traditional hearing aids. If the bones of the middle ear can be directly used, sound quality may be improved along with much less feedback. Also, there are no external components to the IHA, which is, after all, completely implantable.

Three versions of IHA are being tested in human trials. Either the receiver, which collects the sound energy, or the entire hearing device is surgically inserted into the middle ear. The surgical procedure is as follows: skin and tissues around the ear are laid back, and a magnet (which may no may not include electronics) is nestled into the bone behind the ear. A component is then attached to one of the middle ear bones. One prototype is called the Envoy Totally Implantable Hearing Restoration System. This device utilizes the eardrum as the microphone that sends sound energy to the piezoelectric crystal in the middle ear. Two other prototypes use electromagnetic technology. The Vibrant Soundbridge (made by Symphonix Devices, Inc.) has already received an Investigational Device Exemption (IDE) from the Federal Drug Administration (FDA). IDE represents the preliminary stage in receiving FDA approval. Another system, Otologies, is also in the IDE stage of the FDA process. Both of these systems are partially implantable; only the receiver, consisting of the magnet and transducer coil, is implanted. The microphone and amplifier are worn externally. These devices may be more suitable for a wider range of hearing losses than the one using piezoelectric technology, which can be used by those having not more than a moderate (50 dB) hearing loss.

Cochlear Implants

In Section 10.2 we have examined how the cochlea produces signals to the cochlea nucleus. When air cells become so damaged that they cannot be stimulated cells of the spiral ganglion, hearing is lost. Without regular activity, the portion of

the ganglion that receives signals, the dendrite, may atrophy and the cells may die. However, and fortunately, even in the case of complete hearing loss, some ganglion cells survive and remain connected to the appropriate frequency receptors in the cochlear nucleus. If the electrical action from implant electrodes can cause action potentials among the remaining cells, then hearing can be restored. Also, if multiple groups of neurons can be compelled to respond to low-, middle-, and high-frequency components of the cochlea, then perception of speech will be restored.

The next question is: how many channels of portions of the frequency spectrum are necessary to encode speech? Dorman, Louizon, and Rainey determined through the use of bandpass filters that as few as four channels of simplified audio signals to normal-hearing listeners were required to achieve a 90% comprehension rate of words spoken in simple sentences (Dorman Wilson, 2004). Eight channels allowed these listeners to identify 90% comprehension of spoken isolated words. With background noise, more channels were needed to retain this performance, and the more channels used, the better the comprehension.

It is obvious that hair-cell failures engender a roadblock between the peripheral and central auditory systems, resulting in deafness. Cochlear implants restores the link, though bypassing the hair cells to stimulate the direction of the cell bodies in the spiral ganglion. A cochlear implant consists of five components, only two of which are inside the body. In Figure 10.8, an external microphone picks up sounds and directs them to a sound processor that is enclosed in a case behind the ear. The processed signals are sent on to a high-bandwidth RF transmitter, which then relays the information through a few millimeters of skin to a receiver/stimulator that has been surgically implanted in the temporal bone above the ear. The signals pass on to an array of electrodes inside the cochlea. Target cells on the spiral ganglion are segregated from the electrodes by a bony partition.

Continuous interleaved sampling, or CIS, is a strategy used to covert signals into a code for stimulating the auditory nerve. It begins by filtering a signal into frequency bands (16 bands or more). For each band, the CIS algorithm converts the slow changes of the sound envelope into amplitude-modulated groups of biphasic (i.e., having both positive and negative values) pulses at the electrodes. The processor senses the information from low-frequency channels to the electrodes in the apex and information for high-frequency channels to electrodes in the base of the cochlea. Thus, this setup sustains the logic of the frequency mapping in a normal cochlea.

The efficacy of the cochlea, however, depends on a number of factors, among them are the number and location of the surviving cells in the ganglion, the spatial pattern of current flow from the electrodes, and the manner in which the neurons in the brainstem and cortex can encode frequency. If the period of deafness is a long one and only a few cells survive in the spiral ganglion, the electrode stimulation is less likely to convey frequency-specific information to the cochlear nucleon and cortex. Then there is a possibility of surviving cells clustered at one location in the ganglion at the cortex, which results in the lack of range of frequencies necessary for speech understanding.

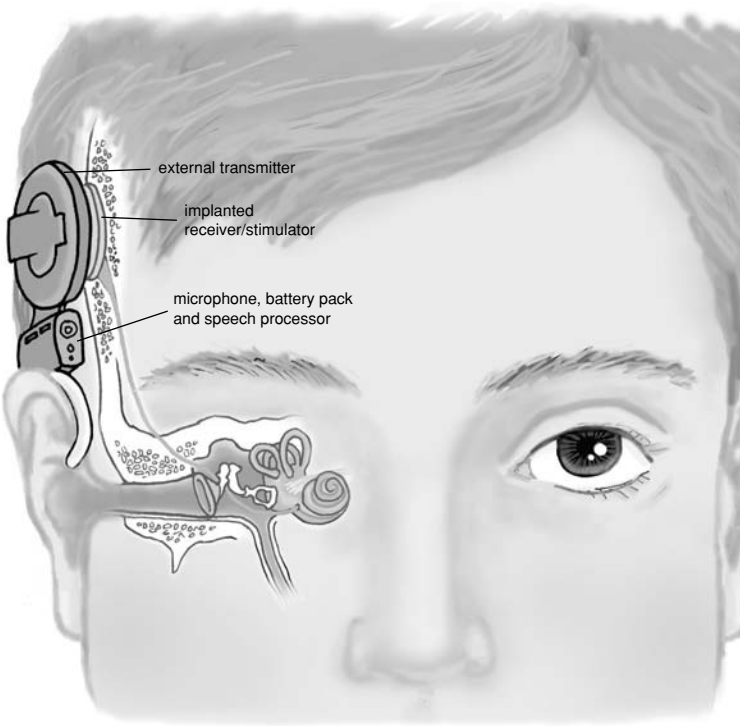


FIGURE 10.8. The components of a cochlear implant. There are five main components, only two of which are inside the body. A microphone above the ear receives sound waves that are directed to a tiny computer behind the ear. The computer transforms the input into specifications for stimuli to be conveyed to the implanted electrodes. The disk-shape transmitter uses high-bandwidth radio waves to send these instructions to a receiver just under the skin. The receiver converts the instructions into electrical stimuli and relays them to the appropriate electrode in the cochlea, which, in turn, excites neurons in the auditory nerve. (Courtesy of Michael F. Dorman and *American Scientist*.)

Adults who lost their hearing and later receive a cochlear implant can associate the new stimulation patterns with their recall of what speech should sound like. Children born deaf do not have this advantage, but it has been found that congenitally deaf children who received cochlear implants during their first or second year can learn spoken language as well as, or almost as well as, children with normal hearing. Children receiving implants later in life have greater difficulty in coping with signals from an implant.

Ongoing cochlear implant research is now focused on the combination of electric and acoustic stimulation (EAS). A number of hearing-impaired people have some ability to hear low frequencies but little or no sensitivity to higher frequencies. When an electrode array is inserted about two thirds of the way into the cochlea, hearing at 1 kHz and above may possibly be restored by electrical stimulation.

Gene Therapy: Formation of New Cochlear Hair Cells

In 1998 the geneticist Huda Y. Zoghbi of the Baylor College of Medicine discovered the key to generation of new hair cells in a gene called Atohl, first discovered in fruit flies. Variations of this gene have since been discovered in almost all species of animals. During fetal development, the gene converts some cells in the ear into hair cells. In other ear cells, called supporting cells, its activity is suppressed. Researchers have demonstrated that working in laboratory dishes, the gene could convert supporting cells into hair cells.

In 2003, Yehoash Raphael and Kobei Kawamoto at the University of Michigan Medical School reported that inserting the gene into live guinea pigs produced thousands of new hair cells (Kawamoto et al., 2003; University of Michigan Health System, 2003). But in those experiments, the researchers did not deafen the animals first. Later they did, using toxic chemicals to kill the hair cells in both ears of ten guinea pigs. Microscopic images taken 3 days later confirmed that all the hair cells were destroyed.

On the fourth day, they used gene therapy with a viral vector to insert the Atohl gene into the guinea pigs' left ears. Within 2 months, new hair cells appeared in treated ears, but not in the untreated right ears. In order to determine whether the new hair cells were functional, the team applied tests of auditory brainstem response to measure the guinea pigs' ability to hear sounds. In effect, they observed increases in brain activity when they exposed the animals to noises, which indicates that their ability to hear was at least partially restored.

Raphael and his colleagues are presently trying to determine how good the restored hearing is. To indicate whether the guinea pigs can hear and how well, they are working with a psychologist who is an expert at training animals to display various behaviors. They are trying to determine, for example, whether the animals can differentiate between loud and soft sounds and between different frequencies. They are also studying animals that have been deafened by other means, older animals and animals that have been deaf for longer periods of time before treatment begins. If these experiments turn out to be successful, the studies necessary to ensure safety and efficacy must be conducted before the technique can be tried on humans, and this phase may take the better part of a decade.

Direct Drive Hearing System (DDHS)

A conventional hearing aid operates by amplifying sound and delivering the sound to the eardrum. From there, the amplified sound passes through the three hearing bones on its way to the cochlea. The direct drive hearing system (DDHS) undergoing testing at the Department of Otolaryngology and Virginia Merrill Bioedel Hearing Research Center at the University of Washington replaces this acoustic path with an electromagnetic one (Von Ilberg et al., 1999). Instead of producing high-volume sound, this hearing aid is attached to an electromagnetic coil that fits inside the earmold. The coil is used to drive a magnet that is attached to the third hearing bone. This system effectively bypasses the ear canal, the eardrum, and

the hearing bones. It is claimed that the fidelity at the third ear bone is improved through the use of this system. Outpatient surgery is required to attach the magnet to the hearing bone. The surgery is performed under local anesthetic and requires about an hour.

One of the advantages of the DDHS is that it eliminates the feedback problem that often occurs with a loose earmold of a conventional hearing aid. There is no feedback because the DDHS does not produce amplified sound. It is also claimed that sound quality is improved by the electromagnetic system and that earmolds do not have to be fitted so tightly. The original study with five volunteers was successful, and FDA authorized a larger study entailing 100 patients. The DDHS is initially targeted at patients with a moderate hearing loss.

Drug Therapy

Drug delivery systems are being integrated into newer designs of cochlear implants. The purpose of these drugs is to arrest the shriveling or demise of remaining hair cells and neural structures in the cochlea and to promote the growth of neural tentacles (called *neurites*) from spiral ganglion cells toward the electrodes. The neurites help each electrode to function as an independent channel of stimulation.

Recent experiment with deafened guinea pigs demonstrated that injecting brain-derived neurotrophic factor and ciliary neurotropic factor could increase the survival and, more importantly, the sensitivity of spiral ganglion cells.

A second approach is to block apoptosis, the normal process of cell death following injury. Self-destruct messages can be triggered by a number of events, for example, acoustic trauma or ototoxic drugs that work through mitogen-activated protein kinase (MAPK) signaling pathway. The pathway can be blocked at various points, thanks to a protein called c-Jun N-terminal kinase (JNK). A peptide inhibitor, developed by a multinational team at the University of Miami, targets this enzyme. By blocking JNK, this team headed by Jing Wang and Thomas Van De Water, prevented hair-cell death and hearing loss following acoustic trauma or administration to the ototoxic antibiotic neomycin (Wang et al., 2003).

10.8 Hearing in Animals

Because their lives depend on the acuity of their hearing, many animals hear a much wider range of frequencies than humans do (cf. Figure 10.9). The average hearing range for humans is about 20 Hz–17 kHz, but killer whales have responded to tones with the frequency range of approximately 0.5–125 kHz with a peak sensitivity at about 20 kHz. Odontocetes (toothed whales) can produce sounds for two overlapping functions: communicating and navigating. Higher frequency clicks probably function primarily in echolocation. Most sound reception, or hearing, seems to take place through the lower jaw. A killer whale may also receive sound through soft tissue and bone surrounding the ear.

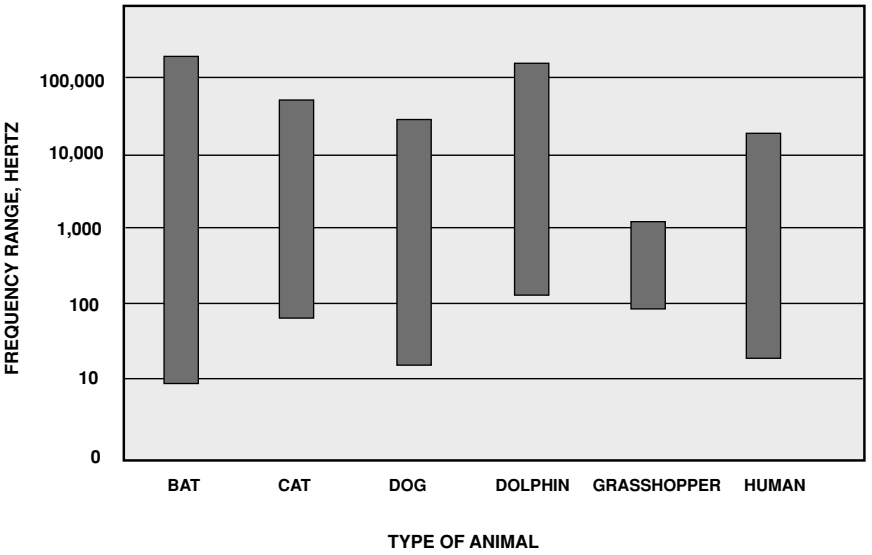


FIGURE 10.9. Range of hearing in animals. Many animals hear a much wider range of frequencies than the human ear can sense.

The term *echolocation* refers to an ability that odontocetes (and a few other animals such as bats) possess that enables them to locate and discriminate objects by projecting high-frequency sound waves and listening for the echoes. The hearing of bats spans a 10 Hz–100 kHz range. Big brown bats (*Eptesicus fuscus*) produce the frequency range of 10–100 kHz for sonar or for acoustic social communication and they also hear those ultrasonic frequencies. They have a lower frequency region of auditory sensitivity from 10 Hz to 5 kHz and may use these lower frequencies to detect insect prey by passive hearing of the insect's own sounds. The hearing is tuned to 0.7–1.3 kHz indicating that some specialization of the auditory system may underlie the capacity to hear these lower frequencies.

An Indian elephant is sensitive to low-frequency tones and could hear as low as 16 Hz at 65 dB. However, the elephant is insensitive to high-frequency tones and it generally could not hear above 12 kHz. The high-frequency hearing ability is the poorest of any mammal yet tested and the failure of the elephant to hear much above 10 kHz demonstrates that the inverse correlation between the head size (i.e., the interaural distance) and high-frequency hearing acuity is valid even for the largest of terrestrial mammals.

Psychophysical investigations in a number of avian (bird) species over the past three decades have added significantly to the knowledge of hearing capabilities of this vertebrate group. Behavioral measurements of absolute auditory sensitivity in a wide variety of birds show a region of maximum sensitivity between 1 and 5 kHz, with a rapid decrease in sensitivity at higher frequencies. Data accumulated to date suggest that, in the region of 1–5 kHz, birds show a level of hearing

sensitivity similar in most respects to that found for the most sensitive members of the mammal class, with avian performance clearly inferior above and below this range of frequencies. Possible exceptions to this general picture include the echolocating oilbird (*Steatornis caripensis*) and growing evidence that pigeons (*Columba livia*) are sensitive to infrasound at moderate intensity levels.

References

American National Standards Institute. 1969. *ANSI 1969: Methods for Calculation of the Articulation Index* (ANSI S3.5-1969, revised 1989). New York: Acoustical Society of America.

American National Standards Institute. 1989. *ANSI 1969: American National Standard Specification for Audiometers* (ANSI S3.6-1969, revised 1989). New York: Acoustical Society of America.

American National Standards Institute. 1986. *ANSI 1973*: *American National Psychoacoustic Terminology* (ANSI S3.20-1973, revised 1986). New York: Acoustical Society of America.

Crocker, Malcolm J. (ed.). 1997. *Encyclopedia of Acoustics*, Vol. 2. Parts XI and XII.

Dorman, Michael F. and Wilson, Blake S. 2004. The design and function of cochlear implants. *American Scientist* 92:436–445.

Dudson, R. S. and King, I. H. 1952. A determination of the normal threshold of hearing and its relation to the standardization of audiometers. *Journal of Laryngology and Otology* 66:366–378. [Reproduced in "Forty Germinal Papers in Human Hearing," Harris, J. Donald (ed.). Groton, C. T. *Journal of Auditory Research*, 1969; pp. 600–601.]

French, N. R. and Steinberg, J. C. 1947. Factors governing the intelligibility of speech sounds. *Journal of the Acoustical Society of America* 19:90–119.

Harris, J. Donald and Myers, C. K. 1971. Tentative audiometric hearing threshold level standards from 8 through 19 kilohertz. *Journal of the Acoustical Society of America* 49:600–601.

Hawkins, J.E., Jr. and Stevens, S. S. 1950. The masking of pure tones and speech by white noise. *Journal of the Acoustical Society of America* 22:12.

Kanzaki, S., Beyer, L. A., Canlon, B., Meixner, W. M., and Raphael, Y. The cytocaud: A hair cell pathology in the waltzing guinea pig. *Audiology & Neural Otology.*

Kawamoto, Kohei, Brough, Douglas E., et al. 2003. *Journal of Neuroscience*.

Kinsler, Lawrence E., Frey, Austin R., Coppens, Alan B., and Sanders, James V. 1982. *Fundamentals of Acoustics*, 3rd ed. New York: John Wiley & Sons, Chapter 11.

Kleine, Ulrich and Mauthe, Manfred. 1994. Designing circuits for hearing instruments. *Siemens Review R& D Special*.

Kryter, Karl D. 1970. *The Effects of Noise on Man*. New York: Academic Press.

Lipscomb, David M. 1974. *Noise: The Unwanted Sound*. Chicago, IL: Nelson-Hall.

Manci, Karen M., Gladwin, Douglas N., Villella, Rita, and Cavendish, Mary G. 1988. *Effects of aircraft noise and sonic booms on domestic animals and wildlife: a literature synthesis.* Report NERC 88/29 AFE TR 88.14. Washington DC: U.S. Department of the Interior.

Miyamoto, R. T., Houston, D. M., Kirk, K. I., Perdew, A. E., and Svirsky, M. A. 2003. Language development in deaf infants following cochlear implantation. *Acta Otolaryngologica* 123:241–244.

Newby, Hayes A. and Popelka, Gerald R. 1992. *Audiology*, 6th ed. New York: Prentice Hall.

Robinson, D. W. and Dudson, R. S. 1956. A re-determination of equal loudness relations for pure tones. *British Journal of Applied Physics* 7:166–181. [Reproduced by: Forty Germinal Papers in Human Hearing." Harris, J. Donald (ed.). Groton, CT: *The Journal of Auditoroy Research*, 1969:600–601.]

Sandlin, R. E. 1988. *Handbook of Hearing Aid Amplification. Vol. I: Theoretical and Technical Considerations*. Boston: College Hill Press.

Sandlin R. E. 1990. *Handbook of Hearing Aid Amplification. Vol. II: Clinical Considerations and Fitting Practices*. Boston: College Hill Press.

Sataloff, Robert T. and Sataloff, Joseph, 1993. *Hearing Loss*, 3rd ed. New York: Marcel Dekker, Inc.

Schow, Ronald L. and Nerbonne, Michael A. 1989. *Introduction to Aural Rehabilitation*, 2nd ed. Boston: Allyn and Bacon.

Silman, Shlomo and Silverman, Carol A. 1991. *Auditory Diagnosis—Principles and Applications*. New York: Academic Press.

University of Michigan Health System. 2005. June 2, 2003. Gene therapy trigger growth of new auditory hair cells in mammals. [On line] http://www.med.umich.edu/opm/newspage/2003/haircells.htm

Von Békésy, Georg. 1978. *Experiments in Hearing*. Huntington, NY: Kreiger.

Von Ilberg, C. J., Kiefer, J., Tillein, T., et al. 1999. Electro-acoustic stimulation of the auditory system: New technology for severe hearing loss. *ORL: Journal for Oto-Rhino-Laryngology and Its Related Specialties* 61:334–340.

Wang, J. T. R., Van De Water, C. Bonny, et al. 2003. A peptide inhibitor of c-Jun N-terminal kinase protects against both aminoglycoside and acoustic trauma-induced auditory hair cell death, and hearing loss. *Journal of Neuroscience* 23:8596–8607.

Problems for Chapter 10

1. Find the articulation index for a male speaker at a normal level 1 m from a listener if we have the following background noise spectrum:

Center Frequency, Hz	Noise Level, dB
200	42
250	39
315	44
400	46
500	48
630	38
800	30
1000	26
1250	20
1600	23
2000	18
2500	15
3150	14
4000	10
5000	10

2. Find the PSIL from the following octave-band noise spectrum.

Center Frequency	Background Noise Level, dB
500	55
1000	63
2000	72

3. Find the PSIL from the following octave-band noise spectrum:

Center frequency, Hz	63	125	250	500	1k	2k	4k	8k
Band-pressure level, dB	52	56	57	62	55	51	50	45

4. Determine the voice level necessary to effectively communicate of a distance of 4 ft with a background of
   (a) 63 dB
   (b) 72 dB
   (c) 85 dB
5. Why does the U.S. Occupational Safety and Health Administration prohibit impact noises of 130 dB or more even if the overall sound pressure level is less than 90 dB during the course of an 8-h day?
6. Express 35 dB I terms of pressure in pascals and in terms of intensity (W/cm^2). At 62.5 Hz, is this considered audible for normal hearing? If so, how much above the MAP is the sound pressure level?
7. How loudly must one speak, in terms of decibels, with a white noise background of 50 dB in order to be understood? How loud must the speech be in order to be detected, if not necessarily understood?
8. At 500 Hz, a 40-dB tone sounds as equally loud as a 5-kHz tone. What is the dB level of that 5 kHz tone?

11
Acoustics of Enclosed Spaces: Architectural Acoustics

11.1 Introduction

Although people have gathered in large auditoriums and places of worship since the advent of civilization, architectural acoustics did not exist on a scientific basis until a young professor of physics at Harvard University accepted an assignment from Harvard's Board of Overseers in 1895 to correct the abominable acoustics of the newly constructed Fogg Lecture Hall. Through careful (but by present-day standards, rather crude) measurements with the use of a Gemshorn organ pipe of 512 Hz, a stopwatch, and the aid of a few able-bodied assistants who lugged absorbent materials in and out of the lecture hall, Wallace Sabine established that the reverberation characteristics of a room determined the acoustical nature of that room and that a relationship exists between quality of the acoustics, the size of the chamber and the amount of absorption surfaces present. He defined a reverberation time T as the number of seconds required for the intensity of the sound to drop from a level of audibility 60 dB above the threshold of hearing to the threshold of inaudibility. To this day reverberation time still constitutes the most important parameter for gauging the acoustical quality of a room. The original Sabine equation

$$T = \frac{0.049V}{\sum_i S_i \alpha_i}$$

is deceptively simple, as effects such as interference or diffraction and behavior of sound waves as affected by the shape of the room, presence of standing waves, normal modes of vibration, are not embodied in that equation. Here V is the room volume in cubic feet, S_i the component surface area and α_i the corresponding absorption coefficient. On the basis of his measurements Sabine was able to cut down the reverberation time of the lecture hall from 5.6 s through the strategic deployment of absorbing materials throughout the room. This accomplishment firmly established Sabine's reputation, and he became the acoustical consultant for Boston Symphony Hall, the first auditorium to be designed on the basis of quantitative acoustics.

In this chapter we shall examine the behavior of sound in enclosed spaces, and develop the fundamental equations that are used in optimizing the acoustics of auditoriums, music halls, and lecture rooms. We shall also study the means of improving room acoustics through installation of appropriate materials. This chapter concludes with descriptions of a number of outstanding acoustical facilities.

11.2 Sound Fields

The distribution of acoustic energy, whether originating from a single or multiple sound sources in an enclosure, depends on the room size and geometry and on the combined effects of reflection, diffraction, and absorption. With the appreciable diffusion of sound waves due to all of these effects it is no longer germane to consider individual wave fronts, but to refer to a *sound field*, which is simply the region surrounding the source. A *free field* is a region surrounding the source, where the sound pattern emulates that of an open space. From a point source the sound waves will be spherical, and the intensity will approximate the inverse square law. Neither reflection nor diffraction occurs to interfere with the waves emanating from the source. Because of the interaction of sound with the room boundaries and with objects within the room, the free field will be of very limited extent.

If one is close to a sound source in a large room having considerably absorbent surfaces, the sound energy will be detected predominantly from the sound source and not from the multiple reflections from surroundings. A free field can be simulated throughout an entire enclosure if all of the surrounding surfaces are lined with almost totally absorbent materials. An example of such an effort to simulate a free field is the extremely large anechoic (echoless) chamber at Lucent Technologies Bell Laboratories in Murray Hill, New Jersey, shown in the photograph of Figure 11.1. Such a chamber is typically lined with long wedges of absorbent foam or fiberglass and the "floor" consists of either wire mesh or grating suspended over wedges installed over (and covering entirely) the "real" floor underneath. Precisely controlled experiments on sound sources and directivity patterns of sound propagation are rendered possible in this sort of chamber.

A *diffuse field* is said to occur when a large number of reflected or diffracted waves combine to render the sound energy uniform throughout the region under consideration. Figure 11.2 illustrates how diffusion results from multiple reflections. The degree of diffusivity will be increased if the room surfaces are not parallel so there is no preferred direction for sound propagation. Concave surfaces with radii of curvature comparable to sound wavelengths tend to cause focusing, but convex surfaces will promote diffusion. Multiple speakers in amplifying systems auditoriums are used to achieved better diffusion, and special baffles may be hung from ceilings to deflect sound in the appropriate directions.

Sound reflected from walls generates a *reverberant field* that is time dependent. When the source suddenly ceases, a sound field persists for a finite interval as the result of multiple reflections and the low velocity of sound propagation. This residual acoustic energy constitutes the reverberant field. The sound that reaches a

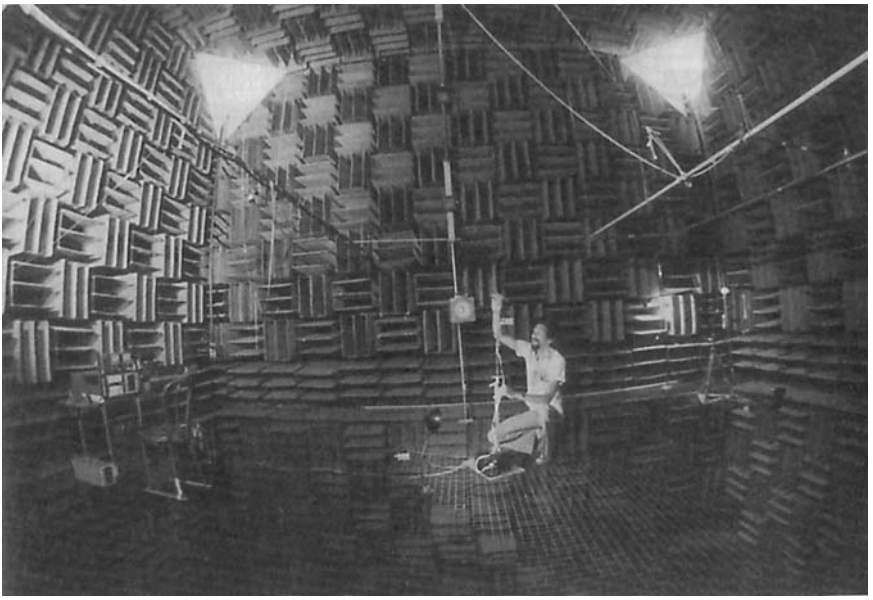


FIGURE 11.1. Photograph of the large anechoic chamber at the Lucent Bell Laboratory in Murray Hill, NJ. Dr. James E. West, a former president of the Acoustical Society of America, is shown setting up test equipment. (Courtesy of Lucent Technologies.)

listener in a fairly typical auditorium can be classified into two broad categories: the direct (free field) sound and indirect (reverberant) sound. As shown in Figure 11.3, the listener receives the primary or direct sound waves and indirect or reverberant sound. The amount of acoustic energy reaching the listener's ear by any single reflected path will be less than that of the direct sound because the reflected path is longer than the direct source–listener distance, which results in greater divergence; and all reflected sound undergo an energy decrease due to the absorption of even the most ideal reflectors. But indirect sound that a listener hears comes from a great number of reflection paths. Consequently, the contribution of reflected sound to the total intensity at the listener's ear can exceed the contribution of direct sound particularly if the room surfaces are highly reflective.

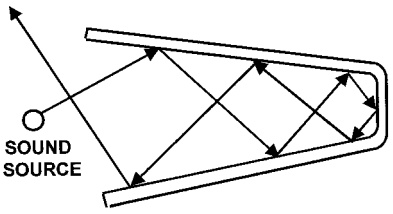


FIGURE 11.2. Sound diffusion resulting from multiple reflections.

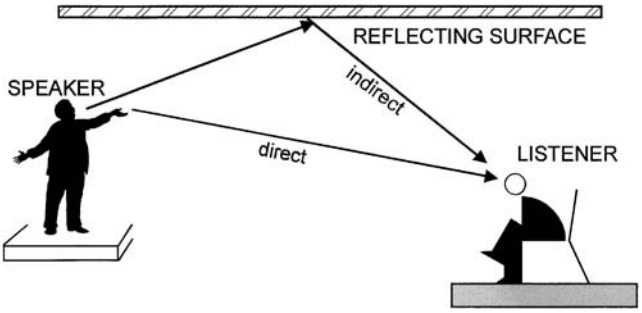


FIGURE 11.3. Reception of direct and indirect sound.

The phases and the amplitudes of the reflected waves are randomly distributed to the degree that cancellation from destructive interference is fairly negligible. If a sound source is operated continuously the acoustic intensity builds up in time until a maximum is reached. If the room is totally absorbent so that there are no reflections, the room operates as an anechoic chamber, which simulates a free field condition. With partial reflection, however, the source continues to add acoustic energy to the room, that is partially absorbed by the enclosing surfaces (i.e., the walls, ceiling, floor and furnishings) and deflected back into the room. For a source operating in a reverberation chamber the gain in intensity can be considerable—as much as ten times the initial level. The gain in intensity is approximately proportional to the reverberation time; thus it can be desirable to have a long reverberation time to render a weak sound more audible.

11.3 Reverberation Effects

Consider a sound source that operates continuously until the maximum acoustic intensity in the enclosed space is reached. The source suddenly shuts off. The reception of sound from the direct ray path ceases after a time interval r/c, where r represents the distance between the source and the reception point and c the sound propagation velocity. But owing to the longer distances traveled, reflected waves continue to be heard as a reverberation which exists as a succession of randomly scattered waves of gradually decreasing intensity.

The presence of reverberation tends to mask the immediate perception of newly arrived direct sound unless the reverberation drops 5–10 dB below its initial level in a sufficiently short time. Reverberation time T, the time in seconds required for intensity to drop 60 dB, offers a direct measure of the persistence of the reverberation. A short reverberation time is obviously necessary to minimize the masking effects of echoes so that speech can be readily understood. However, an extremely short reverberation time tends to make music sound harsher—or less "musical"—while excessive values of reverberation time T can blur the distinction between

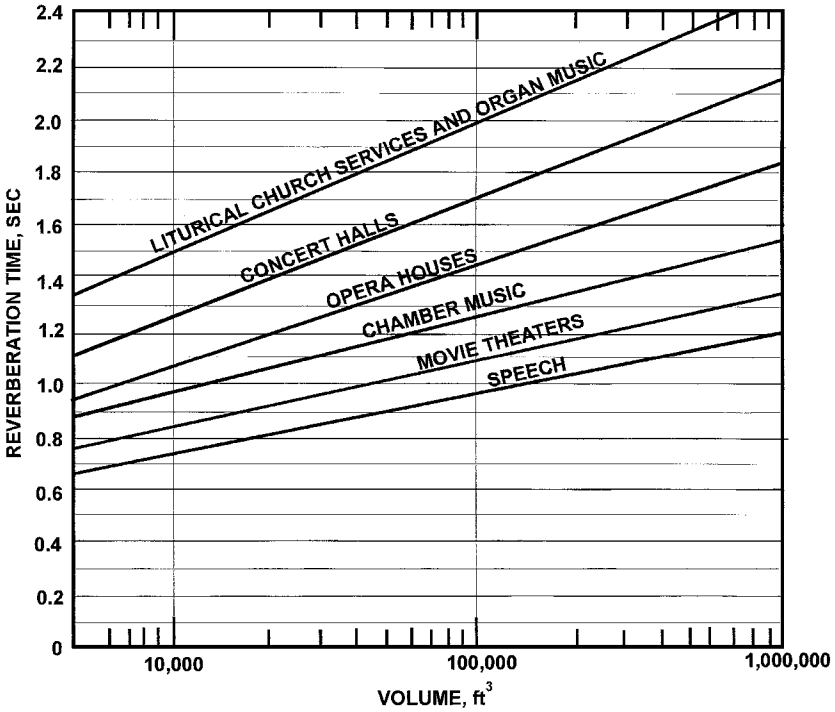


FIGURE 11.4. Typical reverberation times for various auditoriums and functions.

individual notes. The choice of T, which also depends on the room volume, therefore represents an optimization between two extremes.

Figure 11.4 represents the accumulation of optimal reverberation time data as functions of intended use and enclosure volume. Lower values of T occur from increased absorption of sound in the surfaces of the enclosures. Hard surfaces such as ceramic tile floors and mirrors tend to lengthen the reverberation time. In addition to reverberation time, the ability of a chamber or enclosure to screen out external sound minimizes annoyance or masking effects. The acoustic transmission of walls, treated in Chapter 12, constitutes a major factor in enclosure design. A short reverberation time with its attendant high absorption tends to lessen the ambient noise level generated by external sounds that penetrate the walls of the enclosure.

11.4 Sound Intensity Growth in a Live Room

We now apply the classic ray theory to deal with a sound source operating continuously in an enclosure, which will yield results in fairly good agreement with

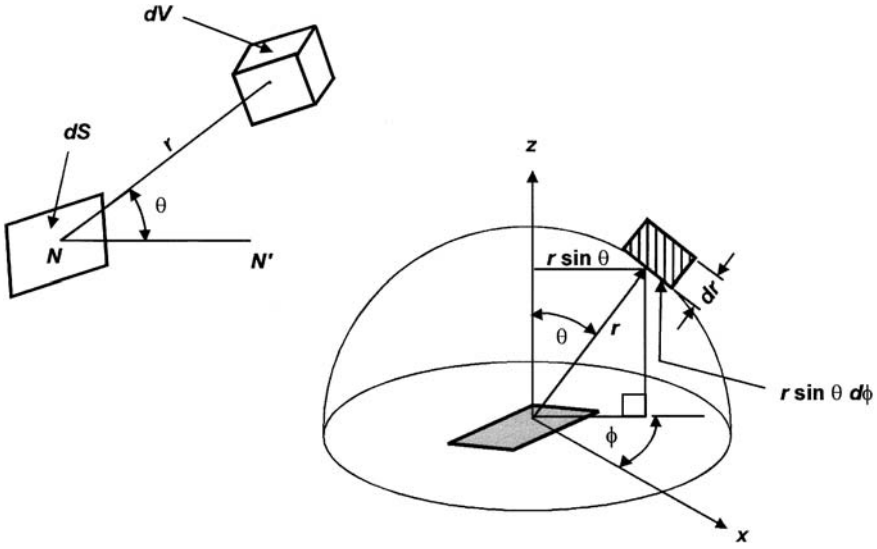


FIGURE 11.5. Geometric configuration for setting up the relationship between energy density and intensity of sound.

experimental measurements. The process of absorption in the medium or the enclosing surfaces prevents the intensity from becoming infinitely large. Absorption in the medium is fairly negligible in medium- and small-sized enclosures, so the ultimate intensity depends upon the absorption power of the boundary surfaces. If the enclosure's boundary surfaces have high absorption the intensity will quickly achieve the maximum which exceeds only slightly the intensity of the direct ray. If the enclosure has highly reflective surfaces, i.e., low absorption, a "live" room ensues; the growth of the intensity will be slow and appreciable time will have elapsed for the intensity to reach its maximum.

After a sound source is started in a live room, reflections from the wall become more uniform in time as the sound intensity increases. With the exception of close proximity to the source, the energy distribution can be considered uniform and random in direction. In reality a signal source having a single frequency will result in standing-wave patterns, with resultant large fluctuations from point to point in the room. But if the sound consists of a uniform band of frequencies or a pure tone warbling over at least a half octave, the interference effects of standing waves are obliterated.

Referring to Figure 11.5, we establish the relationship between *intensity* (which represents the energy flow) and energy density of randomly distributed *acoustic energy*. In the figure dS represents an element of the wall surface and dV the volume element in the medium at a distance r from dS. The distance r makes an angle θ with the normal NN' to dS. Let the average acoustic energy density E (in W/m^3) be assumed uniform throughout the region under consideration. The acoustic energy

in incremental volume dV is $E\ dV$. The surface area of the sphere of radius r encompassing dV is $4\pi r^2$. The projected area of dS on the sphere is $\cos\theta\ dS$. The portion of the total energy contained in dV is given by the ratio $dS\cos\theta/4\pi r^2$. The energy from dV that strikes dS directly becomes

$$dE = \frac{E\,dV\,dS\cos\theta}{4\pi\,r^2}. \tag{11.1}$$

Now consider the volume element dV as being part of a hemisphere shall of radius r and thickness dr. The acoustic energy rendered to S by the complete shell is found by assuming a circular zone of radius $r\sin\theta$ (with θ treated as a constant) in Figure 11.5 and integrating over the entire surface of the shell. The volume of the resultant element is $2\pi r\sin\theta\ r\ dr\ d\theta$. From $\theta = 0$ to $\theta = \pi/2$, and Equation (11.1) yields

$$\Delta E = \frac{E\,dS\,dV}{2}\int_0^{\pi/2}\sin\theta\cos\theta\,d\theta = \frac{E\,dS\,dr}{4}$$

This energy arrives during time interval $t = dr/c$. Hence, the rate of acoustic energy impinging dS from all directions is

$$\frac{\Delta E}{t} = \frac{Ec\,dS}{4}$$

or $Ec/4$ per unit area, which is therefore the intensity I of the diffused sound at the walls. This is also equal to one fourth of a plane wave of energy intensity I incident at a normal angle onto a plane. The intensity I of the diffuse sound at the wall becomes

$$I = \frac{Ec}{4} \tag{11.2}$$

11.5 Sound Absorption Coefficients

All materials constituting the boundaries of an enclosure will absorb and reflect sound. A fraction α of the incident energy is absorbed and the balance $(1-\alpha)$ is reflected. Reflection is indicated by the reflection coefficient r defined as

$$r = \frac{\text{amplitude of reflected wave}}{\text{amplitude of incident wave}}$$

Because the energy in a sound wave is proportional to the square of the amplitude, the sound absorption coefficient α and the reflection coefficient are related by

$$\alpha = 1 - r^2$$

The value of the sound absorption coefficient α will vary with the frequency of the incident ray and the angle of incidence. Materials comprising room surfaces are subject to sound waves that impinge upon them from many different angles as are-sult of multiple reflections. Hence, published data for absorption coefficients

TABLE 11.1. Absorption Coefficients.

	Octave-Band Center Frequency (Hz)					
	125	250	500	1000	2000	4000
Brick, unglazed	0.03	0.03	0.03	0.04	0.05	0.07
Brick, unglazed, painted	0.01	0.01	0.02	0.02	0.02	0.03
Carpet on foam rubber	0.08	0.24	0.57	0.69	0.71	0.73
Carpet on concrete	0.02	0.06	0.14	0.37	0.60	0.65
Concrete block, coarse	0.36	0.44	0.31	0.29	0.39	0.25
Concrete block, painted	0.10	0.05	0.06	0.07	0.09	0.08
Floors, concrete or terrazzo	0.01	0.01	0.015	0.02	0.02	0.02
Floors, resilient flooring on concrete	0.02	0.03	0.03	0.03	0.03	0.02
Floors, hardwood	0.15	0.11	0.10	0.07	0.06	0.07
Glass, heavy plate	0.18	0.06	0.04	0.03	0.02	0.02
Glass, standard window	0.35	0.25	0.18	0.12	0.07	0.04
Gypsum, board 0.5 in.	0.29	0.10	0.05	0.04	0.07	0.09
Panels, fiberglass, 1.5 in. thick	0.86	0.91	0.80	0.89	0.62	0.47
Panels, perforated metal, 4 in. thick	0.70	0.99	0.99	0.99	0.94	0.83
Panels, perforated metal with fiberglass insulation, 2 in. thick	0.21	0.87	1.52	1.37	1.34	1.22
Panels, perforated metal with mineral fiber insulation, 4 in. thick	0.89	1.20	1.16	1.09	1.01	1.03
Panels, plywood, 3/8 in.	0.28	0.22	0.17	0.09	0.10	0.11
Plaster, gypsum or lime, rough finish on lath	0.02	0.03	0.04	0.05	0.04	0.03
Plaster, gypsum or lime, smooth finish on lath	0.02	0.02	0.03	0.04	0.04	0.03
Polyurethane foam, 1 in. thick	0.16	0.25	0.45	0.84	0.97	0.87
Tile, ceiling, mineral fiber	0.18	0.45	0.81	0.97	0.93	0.82
Tile, marble or glazed	0.01	0.01	0.01	0.01	0.02	0.02
Wood, solid, 2 in. thick	0.01	0.05	0.05	0.04	0.04	0.04
Water surface	nil	nil	nil	0.003	0.007	0.02
One person	0.18	0.4	0.46	0.46	0.51	0.46
Air	nil	nil	nil	0.003	0.007	0.03

Note: The coefficient of absorption for one person is that for a seated person (m^2 basis). Air absorption is on a per cubic meter basis.

are for "random" incidence as distinguished from "normal" or "perpendicular" incidence.

The angle–absorption correlation appears to be of somewhat erratic nature, but at high frequencies the absorption coefficients in some materials is roughly constant at all angles. At low frequencies the random-incidence absorption tends to be greater than for normal incidence. However, as Table 11.1 shows, α varies considerably with frequency for many materials, and the absorption coefficients are generally measured at six standard frequencies: 125, 250, 500, 1000, 2000, and 4000 Hz. Absorption occurs as the result of incident sound penetrating and becoming entrapped in the absorbing material, thereby losing its vibrational energy

that converts into heat through friction. Ordinarily the values of α should fall between zero for a perfect reflector and unity for a perfect absorber. Measurements of $\alpha > 1.0$ have been reported, owing possibly to diffraction at low frequencies and other testing condition irregularities.

Let $\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_i$ denote the absorption coefficient of different materials of corresponding areas $S_1, S_2, S_3, \ldots . S_i$ forming the interior boundary planes (viz. the walls, ceiling and floor) of the room as well as any other absorbing surfaces (e.g. furniture, draperies, people, etc.). The average absorption coefficient α for an enclosure is defined by

$$\overline{\alpha} = \frac{\alpha_1 S_1 + \alpha_2 S_2 + \alpha_3 S_3 + \cdots + \alpha_i S_i}{S_1 + S_2 + S_3 + \cdots + S_i} = \frac{A}{S} \tag{11.3}$$

where A represents the total absorptive area $\sum \alpha_i S_i$, and S the total spatial area.

11.6 Growth of Sound with Absorbent Effects

The rate W of sound energy being produced equals the rate of sound energy absorption at the boundary surfaces of the room plus the rate at which the energy increases in the medium throughout the room. This may be expressed as a differential equation governing the growth of acoustic energy in a live room:

$$V \frac{dE}{dt} + \frac{AcE}{4} \equiv W \tag{11.4}$$

The solution for E in Equation (11.4) is

$$E = \frac{4W}{Ac} \left(1 - e^{-(Ac/4V)t}\right) \tag{11.5}$$

with the initial condition that the sound source begins operating at $t = 0$. From the relationship of Equation (11.2) the intensity becomes

$$I = \frac{W}{A} \left(1 - e^{-(Ac/4V)t}\right) \tag{11.6}$$

and from Equation (3.58) the energy density is

$$E = \frac{p^2}{2\rho_0 c^2} \tag{11.7}$$

The mean square acoustic pressure becomes

$$p^2 = \frac{4W\rho_0 c}{A} \left(1 - e^{-(Ac/4V)t}\right) \tag{11.8}$$

Equation (11.8) is analogous to the one describing the growth of direct current in an electric circuit containing an inductance and a resistance. The time constant of the acoustic process is $4V/Ac$. If the total absorption is small and the time constant is large, a longer time will be necessary for the intensity to approach its ultimate

value of $I_\infty = W/A$. The ultimate values of the energy density and mean square acoustic pressure are given by

$$E_\infty = \frac{4W}{Ac}, \qquad p_\infty^2 = \frac{4W\,\rho_0 c}{A}$$

A number of caveats pertain to the use of Equation (11.8). In order that the assumption of an even distribution of acoustic energy be cogent, a sufficient time t must have elapsed for the initial rays to undergo several reflections at the boundaries. This means approximately 1/20 of a second should have elapsed in a small chamber; and the time must approach nearly a full second for a large auditorium. The final energy density, being independent of the size and shape of the room, should be the same at all points of the room and dependent only upon the total absorption A. But Equation (11.6) does not hold for spherical or curved rooms which can focus sounds; neither is Equation (11.8) applicable to rooms having deep recesses nor to oddly shaped rooms or rooms coupled together by an opening, and nor to rooms with some surfaces of extraordinarily high absorption coefficients α (these cause localized lesser values of energy densities).

11.7 Decay of Sound

We can now develop the differential equation describing the decay of uniformly diffuse sound in a live room. The sound source is shut off at time $t = 0$, meaning $W = 0$ at that instant. E_0 denotes the uniformly distributed energy density at that instant. From Equation (11.4)

$$\frac{AcE}{4V}dt = dE \tag{11.9}$$

and the solution to Equation (11.9) becomes

$$E = E_0\, e^{-(Ac/4V)t} \tag{11.10}$$

The intensity I at any time t after the cessation of the sound source is related to the initial intensity I_0 by

$$\frac{I}{I_0} = e^{-(Ac/4V)} \tag{11.11}$$

Applying the operator 10 log to both sides of Equation (11.11) results in

$$\Delta IL = 10 \log e^{-(Ac/4V)t} = \frac{10}{2.3} \ln e^{-(Ac/4V)t} = -\frac{1.087Act}{V} \tag{11.12}$$

where ΔIL denotes the intensity level change in decibels. The intensity level in a live room decreases with elapsed time at a constant decay rate D (in dB/s),

$$D = \frac{1.087Ac}{V}.$$

Following Sabine's definition, we define the reverberation time T as the time required for the sound level in the room to decay by 60 dB:

$$T = \frac{60}{D} = \frac{55.2V}{Ac} \tag{11.13}$$

Expressing volume V in m^3 and area S used to compute A in m^2, and setting sound propagation speed $c = 343$ m/s, Equation (11–13) becomes

$$T = \frac{0.161V}{A} \tag{11.14}$$

Equation (11.14) becomes for English units

$$T = \frac{0.049V}{A} \tag{11.15}$$

where volume V is rendered in ft^3 and A in ft^2 (or *sabins*, with 1 sabin equal to 1 ft^2 of absorption area αS). (One *metric sabin* is equal to 1 m^2 of absorption area.) It becomes apparent here that the reverberation time for a room can be controlled by selecting materials with the appropriate acoustic absorption coefficients. The absorption coefficient of a material can be measured by the introduction of a definite area of the absorbent material in a specially constructed live room or *reverberation* (or *echo*) *chamber*. A photograph of such a chamber is given in Figure 11.6.



FIGURE 11.6. Photograph of a reverberation chamber. (Courtesy of Eckel Industries, Inc.)

Example Problem 1: Reverberation Prediction

A room 8 m long, 4 m wide, and 2.8 m high contains four walls faced with gypsum boards. The only exceptions to the wall area are a glass window 1 m by 0.5 m and a plywood-paneled door 2.2 m by 0.6 m. In addition the door has a gap underneath, 1.5 cm high. In order to estimate the reverberation time of the room at 500 Hz we make use of the data in Table 11.1. Predict the reverberation time T.

Solution

The absorption area (in m^2) is found as follows:

$$\begin{aligned} A = \Sigma\, S_i\, \alpha_i &= [2(8 \times 2.8) + 2(4 \times 2.8) - 2.2(0.6) - (0.015)(0.6) - 1.0(0.5)] \\ &\quad \times 0.05 + (2.2)(0.6)(0.17) + (0.015)(0.6)(1) + 1.0(0.5)(0.18) \\ &\quad + (4)((8)(0.81) + (4)(8)(0.81) = 32.70\ \text{m}^2. \end{aligned}$$

Applying Equation (11.14):

$$T = \frac{0.161 \times 8 \times 4 \times 2.8}{32.7} = 0.82\ \text{s}.$$

The gap at the bottom of the door is treated as a complete sound absorber with a coefficient of unity. From the above estimated value of the reverberation time of 0.44 s and a chamber volume of 89.6 m^3, the room may be suitable for use as a classroom according to Figure 11.4.

11.8 Decay of Sound in Dead Rooms

The derivation of Equation (11.14) was based on the assumption that a sufficient number of reflections occur during the growth or decay of sound and also that the energy of the direct sound and the energy of the fractional amount of sound reflected were both sufficient to ensure a uniform energy distribution. In the case of anechoic chambers, where the absorption coefficient of the materials constituting the boundaries is very close to unity, it is apparent that the derivations of the preceding equations for growth and decay of sound are not applicable. The only energy present is the direct wave emanating from the sound source. The reverberation time must be zero, whereas application of Equation (11.14) would yield a finite reverberation time of $0.161V/S$, where S is simply the total area of the interior surfaces of the chamber. Thus, it is apparent that Equation (11.14) would be increasingly in error as the average sound absorption coefficient increases. If the average value of the absorption coefficient exceeds 0.2, Equation (11.14) will be in error by approximately 10%.

A different approach to ascertaining the decay of sound in a dead room, which was developed by Eyring (1930), is to consider the multiplicity of reflections as a set of image sources, all of which are considered to exist as soon as the real source begins. Let $\bar{\alpha}$, found from the relationship $\bar{\alpha} = (\sum \alpha_i S_i)/\sum S_i$ denote the average

sound absorption coefficient of the room's boundary materials. The growth of acoustic energy at any point in the room results from the accumulation of successive increments from the sound source, from the first-order (single reflection) images with strengths $W(1 - \bar{\alpha})$, from the second-order (secondary reflection) images with strengths $W(1 - \bar{\alpha})^2$, and so on, until all the image sources of appreciable strengths have rendered their contributions. When the true sound source is stopped, the decay of the sound occurs with all the image sources stopped simultaneously along with the source. The energy decay in the room occurs from successive losses of acoustic radiation from the source, then from the first-order images, the second-order images, and so on.

Eyring derived the following equation for the growth in acoustic energy density:

$$E = -\frac{4W}{cS\ln(1-\overline{\alpha})}\left[1 - e^{cS\ln(1-\overline{\alpha})t/4V}\right] \tag{11.16}$$

The above equation is very similar to Equation (11.5) excepting that the total room absorption is given by

$$\alpha = -S\ln(1-\overline{\alpha}) \tag{11.17}$$

Here S is the total area of the boundary surfaces of the room. In a like fashion the analogy to Equation (11.6) for the decay of sound energy is given by

$$E = E_0\, e^{cS\ln(1-\overline{\alpha})t/4V}$$

and the decay rate in dB/s is expressed as

$$D = -\frac{1.08cS\ln(1-\overline{\alpha})}{V}$$

with the reverberation time expressed by

$$T = \frac{0.161V}{-S\ln(1-\overline{\alpha})}$$

For small values of absorption ($\alpha \ll 1$) the term $\ln(1 - \alpha)$ may be replaced by α, the first term in an infinite series. This results in recovering the Sabine formula for live rooms. It should also be noted that the coefficient 0.161 for the Sabine and the Eyring formulas, which is based on the speed of sound at 24°C, will vary according to air temperature. The coefficient becomes somewhat higher at lower air temperatures and vice versa.

Another formula for determining the reverberation time of a room lined with materials of widely ranging absorption coefficients was developed by Millington and Sette (Millington, 1932; Sette, 1993). The Millington–Sette theory indicates that the total room absorption is given by

$$A = \sum -S_i \ln(1 - \bar{\alpha}_i)$$

which yields the reverberation time

$$T = \frac{0.161V}{\sum - S_i \ln(1 - \bar{\alpha}_i)}$$

11.9 Reverberation as Affected by Sound Absorption and Humidity in Air

We have previously not considered the effect of absorption of sound and humidity in air on reverberation times. The volume of air contained in very large auditoriums or a place of worship can absorb an amount of acoustic energy that cannot be neglected as in the case with smaller rooms. If a room is small, the number of reflections from the boundaries is large and the amount of time the sound wave spends in the room is correspondingly small. In this situation acoustic energy absorption in the air is generally not important. In very large room volumes the time a wave spends in the air between reflections becomes greater to the extent that absorption of energy in air no longer becomes negligible. The reverberation equations must now include the effect of air absorption, particularly at higher frequencies (>1 kHz).

Sound waves lose some energy through viscous effects during the course of their propagation through a fluid medium. The intensity of a plane wave lessens with distance according to the equation

$$I = I_0 e^{-2\beta x} = I_0 e^{-mx}.$$

Here $m = 2\beta$ represents the attenuation coefficient of the medium. Some texts use α rather than β to denote the attenuation constant of the medium; we eschew its use in order to avoid confusion with α used in this chapter to denote the absorption coefficient of a surface. During time interval t, a sound wave travels a distance $x = ct$, and the preceding equation may be revised to read

$$I = I_0\, e^{(\beta/4V+m)ct}$$

The expression for the reverberation time becomes

$$T = \frac{0.161V}{A + 4mV} \tag{11.18}$$

where the constant $m$is expressed in units of m^{-1}. The total surface absorption A is given either by Equation (11.3) or (11.17) depending whether that room fits into the category of being an acoustically live or dead chamber. As the room volume V becomes larger, the second term in the denominator of Equation (11.18) increases in magnitude, as air absorption becomes more significant, due to increasing path lengths between the walls. Since m also increases with frequency, air absorption also becomes more manifest at higher frequencies (above 1 kHz) than at lower frequencies. The values of m are given in Figure 11.7[1] as a function of humidity for various frequencies at a normal room temperature of 20°C. More details, also given in tabular form, for a range of air temperatures and humidities are given in the NASA report (1967), prepared by Cyril M. Harris, listed at the end of this chapter. It is seen from Figure 11.7 that the effect of humidity reaches a maximum

[1] The plot of Figure 11.7 applies to *indoor* sound propagation, not to outdoor propagation that includes meteorological effects not present indoors.

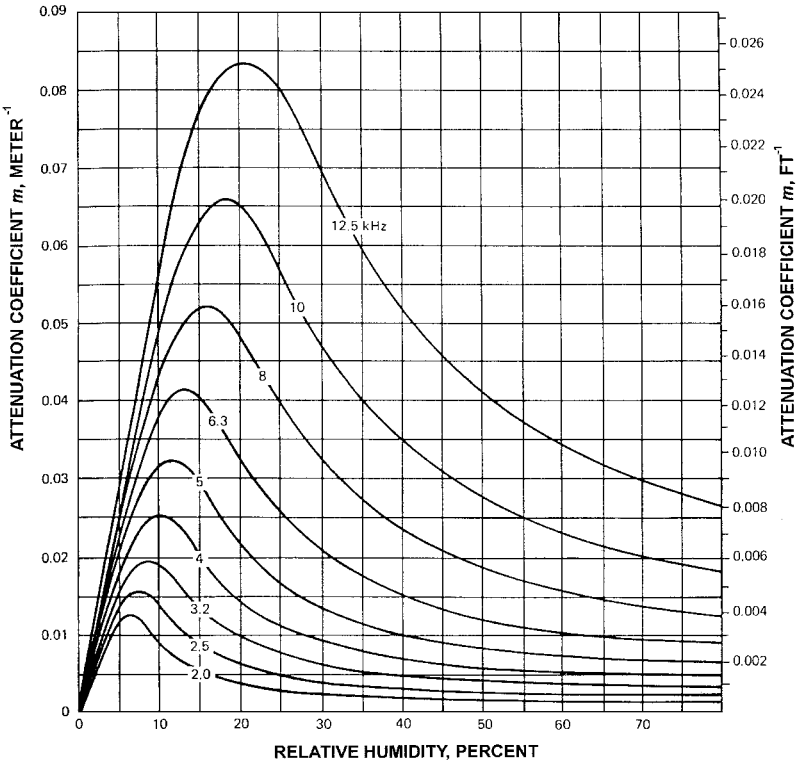


FIGURE 11.7. Values of the total attenuation coefficient m versus percent relative humidity for air at 20°C and normal atmospheric pressure for frequencies between 2 kHz and 12.5 kHz. Values are rendered in both SI units and U.S. Customary System units. (After Harris, 1966, 1967.)

in each of the given frequencies in the 5–25% relative humidity range and then trails off at higher humidities.

Example Problem 2

Find the air absorption at a frequency of 6300 Hz and 25% relative humidity for a room volume of 20,000 m^3.

Solution and Brief Discussion

From Figure 11.7 the value of m is equal to 0.026 m^{-1}. The air absorption $A_{air} = 4mV$ in Equation (11.18) is equal to $4 \times (20{,}000\ m^3) \times 0.026\ m^{-1} = 2080\ m^2$. If we consider absorption at 500 Hz the effect of air absorption would be negligible in comparison.

11.10 Early Decay time (EDT10)

A modification of the reverberation time T_{60} is the *early decay time*, or EDT10, which represents the time interval required for the first 10 dB of decay to occur, multiplied by 6 to produce an extrapolation to 60 dB decay Originally proposed by Jordan (1974), EDT10 is based on early psychoacoustical research, and according to Cremer and Muller (1982), "the latter part of a reverberant decay excited by a specific impulse in running speech or music is already masked by subsequent signals once it has dropped by about 60 dB."

11.11 Acoustic Energy Density and Directivity

In order to account for uneven distribution of sound in some sources, we express the sound intensity I (W/m^2) due to a point source of power W (W) in the direct field (i.e., reflections are not considered) as

$$I = \frac{W\,Q(\theta, \phi)}{4\pi\, r^2} \tag{11.19}$$

where r is the distance (m) from the source and $Q(\theta,\phi)$ is the *directivity factor*. The directivity factor $Q(\theta,\phi)$ equals unity for an ideal point source that emits sound evenly in full space. For an ideal point source above an acoustically reflective surface, in an otherwise free half-space, $Q(\theta,\phi)$ equals 2. The sound or acoustic energy density is the sound energy contained per m^3 at any instant. In the direct field in full space, the direct sound energy density D_D (W s/m^3) is given by

$$D_D = \frac{I}{c} = \frac{W\,Q(\theta, \phi)}{4\pi\, r^2\, c} \tag{11.20}$$

where c is the speed of sound in m/s.

11.12 Sound Absorption in Reverberant Field: The Room Constant

The product IS gives the rate of acoustic energy striking a surface area S; and IS cos θ gives that rate for the incidence angle θ. In an ideal reverberant field, with equal probability for all angles of incidence, the average rate of acoustic energy striking one side of the surface is given by $IS/4$. The power absorbed by the surface having an absorption coefficient α is

$$\text{Power absorbed} = \frac{\alpha IS}{4} = \frac{\alpha c D_R S}{4}$$

where D_R denotes the reverberant sound field density. In a fairly steady state condition the power absorbed is balanced by the power supplied by the source to the reverberant field. This is the portion of the input power W that remains after

one reflection:

$$\text{Power supplied} = W(1-\alpha)$$

The steady-state condition results in

$$\frac{c\, D_R\, S\alpha}{4} = W(1-\alpha)$$

which we rearrange to obtain the energy density in the reverberant field,

$$D_R = 4W\frac{1-\alpha}{\alpha c S} = \frac{4W}{cR} \tag{11.21}$$

where the room constant R is by definition

$$R = \frac{\alpha S}{1-\alpha}.$$

In most cases, the boundaries of the actual enclosure and other objects inside the enclosure are constructed of different materials with differing absorption coefficients. The room constant R of the enclosure is then described in terms of mean properties by

$$R = \frac{S_T\,\overline{\alpha}}{1-\overline{\alpha}}$$

where

R = the room constant (m^2)
S_T = total surface area of the room (m^2)
$\bar{\alpha}$ = mean sound absorption coefficient = $\sum \alpha_i S_i / S_T$.

11.13 Sound Levels due to Direct and Reverberant Fields

Near a point of nondirectional sound source, the sound intensity is greater than from afar. If the source is sufficiently small and the room not too reverberant, the acoustic field very near the source is independent of the properties of the room. In other words, if a listener's ear is only a few centimeters away from a speaker's mouth, the room surrounding the two persons has negligible effect on what the listener hears directly from the speaker's mouth. At greater distances from the source, however, the direct sound decreases in intensity, and, eventually the reverberant sound predominates.

If we are more than one-third wavelength from the center of a point source, the energy density of a point r is given by Equation (11.20) for the direct sound field. Combining the Equations for the direct and the reverberant sound intensities, i.e., Equations (11.20) and (11.21), we get the total sound intensity I given by

$$I = W\left[\frac{Q(\theta,\phi)}{4\pi\, r^2} + \frac{4}{R}\right] \tag{11.22}$$

It is assumed that reverberant sound comes from nearly all directions in a fairly even distribution. The modes generated by standing waves must be rather insignificant; otherwise the assumption of uncorrelated sound is not valid and Equation (11.22) will not truly constitute the proper model for the actual sound field.

The sound pressure level within the room can now be found from

$$L_p = 10\log\left(\frac{I}{I_{\text{ref}}}\right) = 10\log\left[W\left(\frac{1}{4\pi r^2} + \frac{4}{R}\right)\right] + 120$$
$$= L_W + 10\log\left(\frac{1}{4\pi r^2} + \frac{4}{R}\right) \tag{11.23}$$

for an ideal point source emanating equally in all directions. The reference sound intensity I_{ref} is equal to 10^{-12} W/m^2, and the sound power level L_W of the source is defined as

$$L_W = 10\log\left(\frac{W}{10^{-12}}\right)$$

which is given in dB re 1 pW. For an ideal point source over an acoustically reflective surface

$$L_P = 10\log\left[W\left(\frac{1}{2\pi r^2} + \frac{4}{R}\right)\right] + 120 = L_W + 10\log\left(\frac{1}{2\pi r^2} + \frac{4}{R}\right) \tag{11.24}$$

Equations (11.23) and (11.24) are based on the fact that the absorption coefficients do not vary radically from point to point in the room and the source is not close to reflective surfaces. If the sound power and room absorption characteristics are assigned for each frequency band, the sound pressure level L_P can be determined for each frequency band in dB/octave, dB/one-third octave, and so on. If the sound power level is A-weighted, and if the room constant is based on frequencies in the same range as the frequency content of the source, the sound power level will be expressed in dB(A).

Example Problem 3

Predict the reading of a sound pressure level meter 12 m from a source having a sound power level of 108 dB(A) re 1 pW in a room with a room constant $R = 725$ m^2 (at the source frequencies). The source is mounted directly on an acoustically hard floor.

Solution

We apply Equation (11.24) as follows:

$$L_p = L_W + 10\log\left(\frac{1}{2\pi r^2} + \frac{4}{R}\right)$$
$$= 108 + 10\log[(2\pi \times 12^2 \text{ m}^2)^{-1} + 4/725 \text{ m}^2]$$
$$= 86.2 \text{ dB}(A)$$

If the meter would be placed at $R = 3$ m from the source, the SPL meter reading for L_p will increase to 91.7dB(A).

11.14 Design of Concert Halls and Auditoriums

Ideally, the main objective of auditorium design is to get as many members of the audience as close as possible to the source of the sound, because sound levels decrease with increasing distances from the sound source. A good visual line of sight usually results in good acoustics, so stepped seating becomes desirable for larger rooms seating more than 100 people. Reverberation should be controlled in order to provide optimum reinforcement and equalization of sound. For speech the room design should provide more in the way of direct sound augmented by reflections, while the clarity of articulation of successive syllables must be sustained. Rooms for music typically have longer reverberation times because the requirements for articulation are not as stringent, and more enhancement of the sound is desirable.

The aim of the design of a listening type of facility is to avoid the following acoustic defects (Siebein, 1994).

- *Echoes*, particularly those from the rear walls of the facility. Echoes can be lessened or eliminated by placing absorbent panels or materials on the reflecting walls or introducing surface irregularities to promote diffusion of the sound.
- *Excessive loudness* can occur from prolonged reverberation. Again, the proper deployment of absorbent materials should alleviate this problem.
- *Flutter echo* results from the continued reflection of sound waves between two opposite parallel surfaces. This effect can be especially pronounced in small rooms; and this can be contravened by splaying the walls slightly (so as to avoid parallel surfaces) or using absorbent material on one wall.
- *Creep* is the travel of sound around the perimeter of domes and other curved surfaces. This phenomenon is also responsible for whispering gallery effects in older structures with large domed roofs.
- *Sound focusing* arises when reflections from concave surfaces tend to concentrate the sound energy at a focal point.
- *Excessive* or *selective absorption* occurs when a material that has a narrow range of acoustical absorption is used in the facility. The frequency that is absorbed is lost, resulting in an appreciable change in the quality of the sound.
- *Dead spots* occur because of sound focusing or poorly chosen reflecting panels. Inadequate sound levels in specific areas of the listening facility can result.

11.15 Concert Halls and Opera Houses

Three basic shapes exist in the design of large music auditoriums, namely (1) rectangular, (2) fan-shaped, and (3) horseshoe, all of which are illustrated in floor plans of Figure 11.8. A fourth category is the “modified arena”, nearly elliptical in shape. The Royal Albert Hall (constructed in 1871) in London, the Concertgebouw

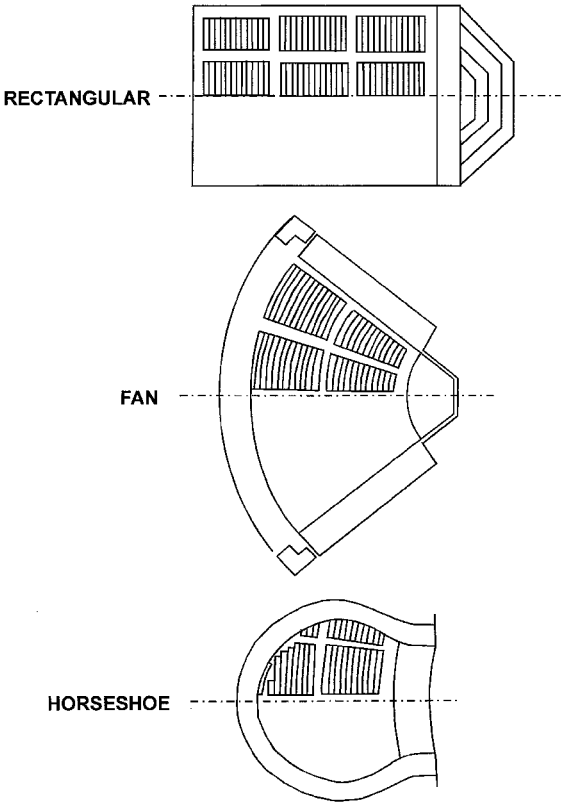


FIGURE 11.8. Three basic hall configurations: rectangular, fan shaped, and horseshoe shaped.

(1887) in Amsterdam, the Sidney Opera House (opened in 1973), and the Colorado Symphony's Boettcher Concert Hall in Denver (opened in 1978 and acoustically remodeled in 1993) are examples of this type of facility.

The *rectangular hall* is quite traditional, and it has been built to accommodate both small and large audiences. But these halls will always generate cross reflections (flutter echoes) between parallel walls. Sound can also be reflected from the rear walls back to the stage, depending on balcony layout and the degree of sound absorption. These reflections can help in the buildup of sound and provides a reasonable degree of diffusion in halls of modest interior dimensions. A considerably larger hall can result in standing wave resonances and excessive flutter echoes.

It is interesting to note that the first music hall to be designed from a scientific viewpoint, by none other than Wallace C. Sabine, is the Boston Symphony Hall (1900), views of which are given in Figures 11.9 and 11.10. The structure contains a high, textured ceiling and two balconies extending along three walls. Volume is 602,000 ft^3; seating capacity 2631; and the reverberation time in the 500–1000 Hz



FIGURE 11.9. A view of the Boston Symphony Hall from the stage. Wallace Clement Sabine was the principal acoustical consultant for this facility. Reprinted with permission from Leo Beranek, *Concert Halls and Opera Houses: Music, Acoustics, and Architecture*, 2nd ed. (New York, NY: Springer, 2004), 48.



FIGURE 11.10. A stage area of the Boston Symphony Hall. Reprinted with permission from Leo Beranek, *Concert Halls and Opera Houses: Music, Acoustics, and Architecture*, 2nd ed. (New York, NY: Springer, 2004), 48.

range is 1.8 s (occupied). Another example of a great rectangular hall is the venerable Grösser Musikvereinssaal (1870) in Vienna which has a reverberation time of 2.05 (occupied) in a volume of 530,000 ft^3. Its superior acoustics can be attributed to its relatively small size, high ceiling, irregular interior surfaces and the plaster interior (Beranek, 2004).

A *fan-shaped hall* accommodates, through its spread, a larger audience within closer range from the sound source (stage). It features nonparallel walls that eliminate flutter echoes and standing waves; and most audience members can obtain a pleasing balance between direct and reflected sounds. A disadvantage in terms of early time delay gap is the distance from the side walls. Often it is necessary to add a series of inner reflectors or canopies hanging from ceilings over the proscenium area to maintain articulation and other acoustical characteristics. Many architects in the United States have resorted to the fan-shaped hall design in order to accommodate larger audiences while retaining an appreciable degree of both visual and aural coupling to the stage area. Relatively modern examples of this design are the Dorothy Chandler Pavilion (1964) in Los Angeles; the Orchestra Hall (completed in 1904 and most recently renovated in 1997, Chicago; the Eastman Theater (opened in 1922 as a movie theater and converted into a concert hall in 1930) in Rochester, New York; and the Kleinhaus Music Hall (1940, designed by Eliel and Eero Saarinen) in Buffalo, New York.

Over a number of centuries *horseshoe-shaped structures* have been used as the preferred design for opera houses and concert halls of modest seating capacity. This design provides for a greater sense of intimacy, and the textures of convex surfaces promote adequate diffusion of sound. The multiple balconies allow for excellent line of sight and short paths for direct sound. The La Scala Opera House (Figures 11.11a, b) in Milan is probably the most notable example of the horseshoe design. It was opened in 1778. This edifice, formally known as Teatro alla Scala, was closed in 2001 for 3 years to undergo a badly needed renovation. A tubular structure and a 17-story fly tower designed by Mario Botta were added to provide stagecraft storage, dressing rooms, and rehearsal rooms. In addition to repairing the ravages of time on the structure, modern stage machinery and new wiring were installed, as well as a new heating, ventilation and air conditioning system. The acoustics were improved by the prominent acoustician, Higini Arau from Barcelona. Other celebrated examples of the horseshoe design are the Carnegie Hall (completed 1897, renovated 1983–1995) in New York City and the Academy of Music (the first opera house in the United States, opened in 1857) in Philadelphia.

Nearly all concert halls have balconies, which were designed to accommodate additional seating capacity within a smaller auditorium volume, so that listeners can sustain an intimate relationship with the stage. The depths of the balconies generally do not exceed more than twice their vertical "window" (opening) to the stage. In fact a smaller ratio is desirable to minimize undue sound attenuation at the rear wall. A rule of the thumb in contemporary acoustical design: the depth of the balcony should not exceed 1.4 times its outlook to the stage at the front of the balcony.

In all types of auditorium design, ceilings constitute design opportunities for transporting sound energy from the stage to distant listeners. In Figure 11.12, it is



(a)



(b)

FIGURE 11.11. (a) View looking toward the stage of the La Scala Opera House. The theater was closed down for three years for seriously needed renovations; (b) a classic example of the horseshoe configuration. (Courtesy of Dr. Antonio Acerba Cantier Escala)

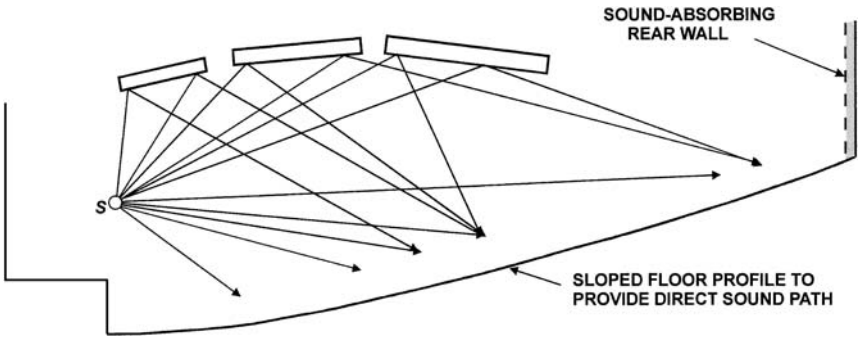


FIGURE 11.12. Transmission of sound to all areas of an auditorium through the ceiling and floor profiling.

shown how a ceiling can convey sound to the listeners without imposing a great time difference between direct and ceiling-reflected sound. Floor profile is also important in establishing the proper ratio of direct to indirect sound. Splays on the side walls have proven effective in promoting diffusion and uniformity of loudness. Rear walls generally should be absorbent to minimize echoes being sent back to the stage.

Many concert halls have been built throughout the world, some with outstanding acoustics and others resulting in dismal sound. Among the features common to all of the aurally superior halls are a limited audience capacity (generally 2800 seats or less), extreme clarity of sound so that the audience can clearly distinguished the individual instruments of the orchestra without loss of fullness or blending of tones associated with reverberation. A good hall also allows the orchestra to hear itself.

Table 11.2 contains a summary of the characteristics of a number of prominent musical facilities all over the world. Figure 11.13 shows the stage view of the Orchestra Hall in Minneapolis (1974), which is patterned after the classical rectangular design. The hall contains a slanted floor with 1590 seats and an additional 983 seats are located in the three-stepped balcony tiers, making for a total audience capacity of 2573. A random pattern of plaster cubes covers the ceiling, providing effective diffusion of sound throughout the hall. This overlay of cubes also continues down the back wall, behind the stage, as shown in Figure 11.13. Wood paneling partially covers the walls, and both the flooring of the stage and audience areas are wood. This concert hall is notable for its clarity, dynamic range, and balance.

Another acoustic success among the contemporary musical facilities is the Kennedy Center for the Performing Arts which opened in Washington, D.C. in 1971. The Center consists of a single structure that contains a 2759-seat concert hall, a 2319-seat opera house (Figure 11.14), and a 1142-seat theater. The location was environmentally challenging, for the Center is situated on a site near the Potomac River, in close proximity to the Washington Ronald Reagan National Airport on the other side of the river. Both commercial and private aircraft fly as

TABLE 11.2. Reverberation Times of Leading Concert Halls and Auditoriums.

			Reverberation Time[a]	
	Volume (ft^3)	Seating Capacity	Occupied	Unoccupied
United States				
Baltimore, lyric Theatre	744,000	2616	1.47	2.02
Boston Symphony Hall	662,000	2631	1.8	2.77
Buffalo, Kleinhans Music Hall	644,000	2839	1.32	1.65
Cambridge, Kresge Auditorium	354,000	1238	1.47	1.7
Chicago, Aric Crown Theatre	1,291,000	5081	1.7	2.45
Cleveland, Severance Hall	554,000	1890	1.7	1.9
Detroit, Ford Auditorium	676,000	2926	1.55	1.95
New York, Carnegie Hall	857,000	2760	1.7	2.15
Philadelphia Academy of Music	555,000	2984	1.4	1.55
Purdue University Hall of Music	1,320,000	6107	1.45	1.6
Rochester, New York, Eastman Theatre	900,000	3347	1.65	1.82
Austria				
Vienna, Grosser Musikvereinssaal	530,000	1680	2.05	3.6
Belgium				
Brussels, Palais des Beaux-Arts	442,000	2150	1.42	1.95
Canada				
Edmonton and Calgary, Alberta Jubilee Halls	759,000	2731	1.42	1.8
Vancouver, Queen Elizabeth Theatre	592,000	2800	1.5	1.9
Denmark				
Tivoli Koncertsal	450,000	1789	1.3	2.25
Finland				
Turku, Konserttisali	340,000	1002	1.6	1.95
Germany				
Berlin, Musikhochschule Konzertsaal	340,000	1340	1.65	1.95
Bonn, Beethovenhalle	555,340	1407	1.7	1.95
Great Britain				
Edinburgh, Usher Hall	565,000	2760	1.65	2.52
Liverpool Philharmonic Hall	479,000	1955	1.5	1.65
London, Royal Albert Hall	3,060,000	5080	2.5	3.7
London, Royal Festival Hall	755,000	3000	1.47	1.77
Israel				
Tel Aviv, Frederic R. Mann Auditorium	750,000	2715	1.55	1.97
Italy				
Milan, Teatro Alla Scala	397,300	2289	1.2	1.35
Netherlands				
Amsterdam, Concertgebouw	663,000	2206	2.0	2.4
Sweden				
Gothenburg, Konserthus	420,000	1371	1.7	2.0
Switzerland				
Zurich, Grosser Tonhallesaal	402,500	1546	1.6	3.85
Venezuela				
Caracas, Aula Magna	880,000	2660	1.35	1.8

[a] At 500–1000 Hz.



FIGURE 11.13. The stage area of the Minnesota Orchestra Hall in Minneapolis. (Courtesy of the Minnesota Orchestral Association.)

low as a few hundred feet directly over the roof, and occasionally helicopters pass by along the Potomac River at rooftop levels. In addition, vehicular traffic runs across the river and directly beneath the plaza of the Center.

To deal with these external noise sources, the Center was constructed as a box-within-a-box, so that each of the auditoriums is totally enclosed within an outer shell. The columns within each auditorium are constructed to isolate interior



FIGURE 11.14. The interior of the John F. Kennedy Opera House in Washington, DC. (Courtesy of the John F. Kennedy Center. Photograph by Scott Suchman.)

ceilings, walls, and floors from both airborne and mechanical vibrations. The double-wall construction generally consists of 6-in. solid high-density blocks separated by a 2-in. air gap. The huge windows in the Grand Foyer facing the river consist of 1.27-cm (1/2-in.) and 0.64-cm (1/4-in.) thick glass sheets separated by a 10-cm (4-in.) air gap. Resilient mounts are used to isolate interior noise sources (e.g. transformers, air-conditioning units, etc.) The ductworks are acoustically lined, flexible connectors are used, special doors are installed at all auditorium entrances together with "sound locks" between foyer and the auditoriums.

The rectangular concert hall encompasses a volume of 682,000 ft^3 and accommodates an audience of 2759. Large contoured wall surfaces and a coffered ceiling abet the diffusion of sound at low and high frequencies. The 11 massive crystal chandeliers, each weighing 1.3 metric ton, donated by the Norwegian government, also contribute to the diffusion. The balconies are purposely shallow to prevent reduction of sound below the balcony overhang. Unoccupied, the concert hall has a reverberation time of 2.2 s at 500 Hz and 2.0 s at 1 kHz; the corresponding values are 2.0 s and 1.8 s for the fully occupied hall.

Located in downtown Seattle, Washington, Benaroya Hall opened in September 1998, contains two spaces for musical performances: a 2500-seat main auditorium (Figure 11.15) and a more intimate 540-seat recital hall. The main auditorium, the S. Mark Taper Foundation Auditorium, is a classic rectangular configuration with the stage enclosed in a permanent acoustic shell. LMN (Loschky, Marquartdt and



FIGURE 11.15. Seattle Symphony performing in Benaroya Hall in Seattle, WA under the direction of Music Director Gerard Schwarz. (Courtesy of the Seattle Symphony, photograph by Craig Raymond.)

Nesholm) Architects and the acoustical consultant, Cyril M. Harris, combined the shoebox design with state-of-the-art materials to achieve maximum warmth and balance.

The location of Benaroya Hall in a busy sector of Seattle posed special challenges to the designers. They had to contend with a railroad tunnel running diagonally beneath the auditorium and a nearby underground bus tunnel. A slab of concrete more than 2m (6 ft) thick, 24 m (80 ft) wide, and 131 m (430 ft) long was poured under the hall to swallow the sound from the tunnels. In order to combat other exterior noises, the designers essentially encased a building within a building. The auditorium, weighing 12 million kilograms, rests on 310 rubber pads, which absorb vibration from the tunnels. The pads are 38 cm^2 and are composed of four layers of natural rubber sandwiched with 0.32-cm (1/8-in.) steel plates.

All electrical, plumbing, and other noise-generating equipment are located outside the auditorium box. Any penetration of the box is made with flexible connections. Water is known to transmit sound very well, so the fire sprinkler system is left dry and it will flood with water only when a fire is detected. The ventilation system is connected to the outside by a sound trap, which channels air through narrow openings between perforated aluminum boxes of sound insulation. Very large vents collect air below the floor and move it slowly behind the auditorium to another sound trap and from thereon to fans. The basic idea of the ventilation system is to move a high volume (2400 m^3/min) of air at low speed, eliminating noise created by fast-moving air in conventional systems.

Instead of frame construction, the walls are built of precast concrete panels. The heavy mass helps to cut down building vibration and provides a stiff, hard surface to reflect concert sound. Side walls, back walls, and ceiling are covered with paneling shaped like truncated pyramids to reflect sound at various angles to aid diffusion. Randomly spaced wood blocking behind the angled paneling creates framed sound boxes that reflect both high and low frequency sounds so that no tone is eliminated from the music. Side walls are covered with particle boards veneered with a dense, fine-grained hardwood from a single makore tree. The ceiling is suspended from the roof by hundreds of metal strips. The ceiling is coated with 3.8 cm (1.5 in.) of plaster in irregularly shaped panels to diffuse sound. The plaster is sufficiently dense to prevent the ceiling from vibrating. House lighting is imbedded in the ceiling to minimize sound leakage. Access to the light bulbs is achieved above the ceiling through heavy, removable plaster caps.

11.16 Band Shells and Outdoor Auditoriums

Over the past several decades there has been an increasing trend toward outdoor concerts, either at band shells or in semi-open structures. These types of structures are more economical to construct than full-fledged indoor concert halls, and they also meet the criteria of providing an informal setting for audiences seeking entertainment in a usually rural environment, away from the metropolitan areas.



FIGURE 11.16. The Hollywood Bowl in Los Angeles, California, after its reconstruction in 2004. (Courtesy of the Los Angles Philharmonic, photograph by Mathew Imaging.)

It is generally not possible for a large orchestra to play effectively in open air. The use of a band shell becomes necessary, as this permits the members of a musical group to hear each other and directs the music toward the audience area. The band shell site should also be carefully selected. Ideally, the region should be isolated from the noise of passing traffic and overhead aircraft. The topology of the land also ranks important in providing the proper acoustics. If the land can be contoured properly, there can be appreciably less attenuation of the sound than would be the case if the band shell were located on flat ground.

The Hollywood Bowl (Figure 11.16) is an example of an orchestra shell strategically located in a natural hollow. The only reflected sound is that reflected from the shell, but the stage distances are sufficiently short so that the sound is heard without any discernable time delay gap. However, shells can never equal the dynamic range of sound power and sonority that are achieved in an enclosed reverberant concert hall. The use of high quality amplification systems is therefore often necessary at many outdoor concerts.

The Bowl, the summer home of the Los Angeles Philharmonic, has undergone a number of changes throughout its years of operation. The previous shell design—the fourth since 1922—has been subjected to a great deal of criticism from performers and audiences alike. In 2004, the shell and stage area was reconstructed and made 30% larger to accommodate a full orchestra. The new shell is provided with a set of adjustable reflectors above the orchestra, mounted on a 27.5 m $\times$ 18.2 m (90 ft $\times$ 60 ft) elliptical structure that also supports an improved lighting



FIGURE 11.17. A view looking toward the interior of the Tanglewood Music Shed in Lenox, MA. The ceiling canopy reflects sound, adding dynamic range and brilliance. (Photograph by Kim Knox Beckhius.)

system. Advantage was also taken of the opportunity to install a completely new sound system.

Another type of "outdoor" structure is the "music shed," a semi-enclosed structure specifically designed for musical performances. The Tanglewood Music Shed in the Berkshires region of Massachusetts (Figure 11.17) is the summer home of the Boston Symphony Orchestra, and the quality of its acoustics exceeds that of any band shell for an audience of 6000 persons. An additional 6000 people on the lawn adjacent to the shed can listen to the music through the open segments of the pavilion. The canopy of the interior projects and diffuses sound through a volume of 42,500 m^3 (1,500,000 ft^3). The ceiling is constructed of 5-cm thick wooden planks; the side and rear walls are of 2 cm fiberboard; and the floor is simply packed earth. When occupied, the shed embodies a reverberation time of 2 s in the frequency range of 500–1000 Hz, which is quite excellent considering the rustic nature of its construction. The Tanglewood shed is the precursor to similar structures at Wolf Trap in Vienna, Virginia, the Performing Arts Center at Saratoga Spring, New York (the summer home of the Philadelphia Symphony Orchestra), and the Blossom Music Center near Cleveland, Ohio.

On July 16, 2004, the Jay Pritzker Pavilion (Figure 11.18), a radical outdoor concert facility located at Chicago's Millennium Park near the banks of Lake Michigan, presented its first musical program. Working in concert with the eminent architect Frank Gehry, the consulting firm TALASKE of Oak, Park, Illinois dealt with the acoustic challenge of outdoor orchestral performances. TALASKE



FIGURE 11.18. The Jay Pritzker Pavilion in Chicago. The radical design also features a trellis system from which loudspeakers are suspended. (Courtesy of TALASKE, graphics by DOXA.)

established the overall shape and arrangement of the stage enclosure and, in the course of design, refined the shapes and finishes of the interior surfaces. The stage of the pavilion (Figure 11.19), the permanent home of the Grant Park Music Festival Orchestra and Chorus, can accommodate a 100 plus member orchestra and



FIGURE 11.19. The stage of the Pritzker Pavilion can accommodate an orchestra of more than 100 members plus a 150-member chorus. (Photograph by Richard H. Talaske.)

also a 150-member chorus. The walls and the ceiling of the stage incorporate contours and angles precisely planned to enhance cross-room reflections so that musicians can hear each other from one side of the stage to the other and from front to back. When an orchestra performs at high volume it is often difficult for musicians playing in the high register to hear instruments in the lower registers. To circumvent this problem, the orchestra risers feature a platform system, researched and designed by the acousticians and musicians together, that allows the musicians to feel the vibration created by the cellos and bass instruments. The riser system, essentially a floating floor with rigid interconnections and resilient materials just below the floor surface, acts to maximize cross stage vibrations. This enables a more precise ensemble and tonal balance.

To compensate for the lack of reflected sound in an outdoor setting and the lack of envelopment, modern technology was recruited to improve the sound environment by blending architecturally created and electronically processed sound. The Jay Pritzker Pavilion, which replaces the Petrillo Music Shell constructed in 1931 and still in use on a more limited basis, is the first orchestral facility that distributes sound by using an overhead steel trellis structure from which loudspeakers are suspended (Figure 12.20). The trellis area that incorporates the network of speakers is 100 m (325 ft) wide by 180 m (600 ft) long. These loudspeakers are strategically suspended at predetermined heights and orientations, in concentric circles outward from the stage. A distributed sound reinforcement system provides direct or "frontal" sound to the audience. A separate acoustic enhancement system,



FIGURE 11.20. View from the stage of the Pritzker Pavilion. The trellis overhead of the audience contains a distributed loudspeaker system.

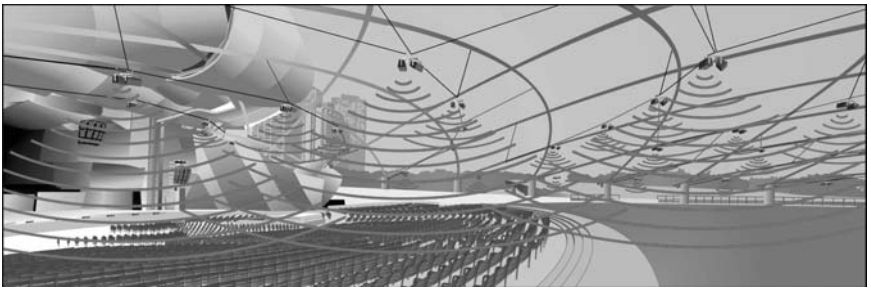


FIGURE 11.21. Distribution of sound from the speakers distributed on a trellis system overhead of the audience at the Pritzker Pavilion. (Courtesy of TALASKE, graphics by DOXA.)

LARES™ (Lexicon Acoustic Reinforcement and Enhancement System), delivers lingering, enveloping sound characteristics through supplemental loudspeakers. It uses a time varying technique to maintain stability by shifting the output in time enough to maintain stability but not enough to introduce tonal coloration. This system simulates reflections and reverberations using specialized electronics and digital processing. Time delay creates the impression that sound is coming from the stage rather than from the speakers. These two systems work together to deliver sound throughout the seating area and the Great Lawn (cf. Figure 11.21). The system makes use of eight microphones deployed around the stage area for the enhancement system and 30 or so microphones for reinforcement, 143 amplifiers housed in three separate equipment rooms, 90 CobraNet™ channels, and 185 speakers. The beauty of this system is that the acoustic environment can be adjusted to work equally well for orchestral music, staged opera, blues, jazz, rock, and other music. The fixed seating area accommodates 4000 concertgoers, and the Great Lawn has a capacity for a lawn crowd of up to 7000 stretching more than a city block behind the front area.

Backstage facilities include warm-up rooms that are shared with the Harris Theater for Music and Dance. A bridge near the concert area was configured to help mask the traffic noises at the pavilion.

11.17 Subjective Preferences in Sound Fields of Listening Spaces

Beyond the Sabine realm of architectural acoustics, which is based on the relatively simple but effective formula $T_{60} = (0.161V)/\ \Sigma A_i \alpha_i$, other considerations come into play in determining optimal configurations for listening spaces (Ando, 1998). This involves combining the elements of psychoacoustics, modeling of the auditory-brain system, and mapping of subjective preferences. The physical properties of source signals and sound fields in a room are considered, in

particular the autocorrelation function (ACF) that contains the same information as power density spectrum but it is adjusted to account for hearing sensitivity. Effective duration of the normalized ACF is defined by the delay τ_e at which the envelope of the normalized ACF becomes one-tenth of its maximal value. The response of the ear includes the effects of time delay due not only to the room's acoustical characteristics, but also the spatially incurred differences in the signals reaching the right and the left ears. The difference in the sounds arriving at the ear is measured by the "interaural cross-correlation function" or IACF, which is defined by

$$\text{IACF}(\tau) = \frac{\int_{t_1}^{t_2} p_L(t) p_R(t+\tau)}{\sqrt{\int_{t_1}^{t_2} p_L^2 \int_{t_1}^{t_2} p_R^2 \, dt}} \tag{11.25}$$

where L and R denotes entry to the left and right ears, respectively. The maximum possible value of IACF is unity. The time $t = 0$ is the time of the arrival of the *direct* sound from the impulse radiated by a source. Integration from 0 to t_2 ms includes the energy of the direct sound and any early reflections and reverberant sounds occurring during the t_2 interval. Because there is a time lapse of about 1 ms for a sound wave to impinge the other side of the ear after impinging one side, it is customary to vary τ over the range from –1 ms to +1 ms. In order to obtain a single number that measures the maximum similarity of all waves arriving at the two ears with the time integration limits and the range of τ, it is customary to choose the maximum value of IACF, which is then called the interaural cross-correlation coefficient (IACC), i.e.

$$\text{IACC} = |\text{IACF}(t)|_{\max}$$

Different integration periods are used for IACC. The standard ones include IACC_A ($t_1 = 0$ to $t_2 = 1000$ ms), and $\text{IACC}_{\text{E(arly)}}$ (0–80 m), $\text{IACC}_{\text{L(ate)}}$ (80–1000 m). The early IACC is a measure of the apparent source width ASW and the late IACC is a measure of the listener envelopment LEV. IACC is generally measured by recording on a digital tape recorder the outputs of two tiny microphones located at the entrances to the ear canals of a person or a dummy head, and quantifying the two ear differences with a computer program that performs the operation of Equation (11.25). IACC_{A} is determined with a frequency bandwidth of 100 Hz–8 kHz and for a time period of 0 to about 1 s.

Subjective attributes for a sound field in a room have been developed experimentally with actual listeners. The simplest sound field is considered first, a situation which consists of the direct sound and a single reflection acting in lieu of a set of reflections. The data obtained are based on tests in anechoic chambers (which allowed for simulation of different concerts halls) with normal hearing subjects listening to different musical motifs. From these subjective tests the optimum design objectives are established, namely the listening level, preferred delay time,

preferred subsequent reverberation time (after the early reflections), and dissimilarity of signals reaching both ears (involving IACC).

These factors are each assigned scalar values and then combined to yield a subjective preference that can vary from seat to seat in a concert hall. Some rather interesting results of investigations include the fact that the right hemisphere of the brain is dominant for "the continuous speech." while the left hemisphere is dominant when variation occurs in the delay time of acoustic reflection. The left hemisphere is usually associated with speech and time-sequential identifications, while the right hemisphere is allied with nonverbal and spatial identifications. A proposed model for the auditory-brain system was developed (Ando, 1998) that incorporates a subjective analyzer for spatial and temporal criteria and entails the participation of the left and the right hemispheres of the brain. The power density spectra in the neural processes in the right and left auditory pathways yield sufficient information to establish autocorrelation functions.

It is obvious that different individuals are likely to have different subjective preferences with respect to the same musical program, so seating requirements can differ, with respect to the preferred listening level and to the initial time delay, and even lighting. For example, evaluations were conducted for a performance of Handel's *Water Music* with 106 listeners providing the input on their preferences with respect to listening level, reverberation time, and IACC. The information thus obtained can provide insight into how the acoustic design of a concert hall and a multipurpose auditorium can be accomplished. Procedures for designing sound fields include consideration of temporal factors, spatial factors, the effect of sound field on musicians, the conductor, stage performers, listener, and the archetypal problem of fusing acoustical design with architecture. Multipurpose auditoriums present bigger challenges, some of which have been met very well and many which have not. In the design procedure, a number of factors other than acoustical include measurable quantities such as temperature, lighting levels, and so on, and other less tangible determinants that can be aesthetically evocative.

References

American Society for Testing and Materials. 1981. A test method for sound absorption and sound absorption coefficients by the reverberation room. Method. ASTM C423-81a.

Ando, Yoichi. 1985. *Concert Hall Acoustics*. Berlin: Springer-Verlag.

Ando, Yoichi. 1998. *Architectural Acoustics: Blending Sound Sources, Sound Fields, and Listeners*. New York: AIP Press and Springer-Verlag. (A followup to Yoichi's previous text. The author, arguably Japan's most prominent architectural acoustician, introduces a theory of subjective preferences, based on a model of the auditory cognitive functioning of the brain.)

Bies. D. A. and Hansen, C. H. 1996. *Engineering Noise Control: Theory and Practice*, 2nd ed. London: E. and F. N. Spon.

Beranek, Leo L. 1962. *Music, Acoustics, and Architecture*. New York: John Wiley & Sons.

Beranek, Leo L. July 1992. Concert Hall Acoustics—1992. *Journal of the Acoustical Society of America* 92, 1: 1–39.

Beranek, Leo. 2004. *Concert and Opera Houses: Music, Acoustics, and Architecture*, 2nd ed. New York: Springer-Verlag. (A tremendous compendium of the acoustical characteristics of musical auditoria and how they were achieved. This volume is a must-have text for architectural acousticians and aficionados of architectural acoustics.)

Cremer, L., and Muller, H. A. 1982. *Principles and Applications of Room Acoustics*, vols. 1 and 2. English translation with additions by T. J. Schultz. New York: Applied Science Publishers.

Crocker, Malcolm J. (ed.). 1997. *Encyclopedia of Acoustics*, vol. 3. New York: John Wiley & Sons: Chapters 90–93: 1095–1128.

Eyring, C. F. 1930. Reverberation time in "dead rooms."*Journal of the Acoustical Society of America* 1: 217–241.

Harris, Cyril M. 1966. Absorption of sound in air vs. humidity and temperature.*Journal of the Acoustical Society of America*, 40: 148–49.

Harris, Cyril M. January 1967. Absorption of sound in air vs. humidity and temperature, NASA contractor report NASA CR-647. Springfield, VA 22151: Clearinghouse for Federal Scientific and Technical Information.

Harris, Cyril M. (ed.). 1991. *Handbook of Acoustical Measurements and Noise Control*, 3rd ed. New York: McGraw-Hill.

International Standardization Organization. 1963. Measurement of sound absorption in a reverberation room. ISO R354-1963.

Jordan, V. 1974. Presented at the 47th Audio Engineering Society Convention, Copenhagen.

Kinzey, B. Y., Jr., and Howard, S. 1963. *Environmental Technologies in Architecture*. Englewood Cliffs, NJ: Prentice Hall.

Knudsen, Vern O., and Harris, Cyril M. 1978 (reissue). *Acoustical Designing in Architecture*. Woodbury, NY: Acoustical Society of America. (A classic text by two of the giants in field of architectural acoustics.)

Kutruff, H. 1979. *Room Acoustics*, 2nd ed. London: Applied Science.

Kuttruff, H. 1991. *Room Acoustics*, 3rd ed. London: Elsevier Applied Science.

Lambert, Mel. August 2004. Deus ex machine: Canopy and line array fly at the Hollywood Bowl. *Systems Contractor*: 56.

Millington, G. 1932. A modified formula for reverberation. *Journal of the Acoustical Society of America*, 4: 69–82.

Morse, Philip M., and Bolt, Richard H. 1944. Sound waves in rooms. *Review of Modern Physics*, 16: 65–150.

MVI Technologies. 1997. *Mediacoustic: Teaching acoustics by computer*. Lyon, France: 01 dB Stell. (A multinational compact disk conceived and prepared by acousticians at l"Ecole Polytechnique de Lausanne, l'Ecole Nationale es Travaux Publics de Lyon, Liverpool University, and the companies Acouphen, Estudi Acustic, and o.1 dB. It contains excellent modules on basic physics, noise and man. room acoustics, and noise control.)

Sabine, Wallace Clement. Reissued 1993. *Collected Papers on Acoustics*. Los Altos, CA: Peninsula Publishing. (Included are the original papers that laid the foundation of modern architectural acoustics.)

Sette, W. J. 1933. A new reverberation time formula. *Journal of the. Acoustical Society of America*, 4:193–210.

Seattle Times. June 21, 1998. Benaroy Hall: orchestrating the sound of music. Section A: 9.

Siebein, Gary W. June 1994. Acoustics in buildings: a tutorial on architectural acoustics, presented at the 127th Annual Meeting of the Acoustical Society of America, Cambridge, MA.

Problems for Chapter 11

1. The amplitude of the reflected wave is one-half that of the incident wave for a certain angle and frequency. What is the reflection coefficient? What is the corresponding sound absorption coefficient?
2. Find the average sound coefficient of 125 Hz and 2 kHz for a wall constructed of different materials as follows:

Area (m^2)	Material
85	Painted brick
45	Gypsum board on studs
35	Plywood paneling
4	Glass window

3. Find the time constant of a room that is 8 × 10 × 2.5 m and an average sound absorption coefficient of 0.34.
4. A room has dimensions 3.5 m high × 30 m length × 10 m width. Two of the longer walls consist of plywood paneling; the rear wall is painted brick and the front has gypsum boards mounted on studs. The ceiling is of acoustic paneling and the floor is carpeted. For 250 Hz and 125 people in the audience:
   (a) Determine the reverberation time from the Sabine equation.
   (b) Compute the room constant R.
   (c) Comment on the suitability of the room for use as an auditorium.
5. An auditorium has dimensions 6.0 m high × 22.0 m length × 15.5 m width. The floor is carpeted and one of the longer walls has gypsum boards mounted on studs the entire length, while the other three walls and the ceiling are constructed of plaster. For 500 Hz:
   (a) determine the reverberation time from the Sabine equation.
   (b) compute the room constant R.
6. In Problem 5 there is a set of two swinging doors 2.5 m in height and 2.5 m in total width. However, there are open cracks along the bottom and between the doors 1 cm wide. In addition there is an open window on the gypsum-boarded wall that is 0.8 × 1 m. Find the effect of these openings on the reverberation time of the auditorium.
7. A room has dimension 4.0 m high × 18.0 m length × 9.5 m width. The floor is carpeted and one of the longer walls has gypsum boards mounted on studs along the entire length, while the other three walls are constructed of painted brick. The ceiling is plastered. For 500 Hz:

(a) determine the reverberation time from the Sabine equation.
(b) compare the reverberation time obtained from the Eyring formula with that obtained in part (a) above.
(c) compute the room constant R.

8. An isotropic source emitting sound power level of 106 dB(A) re 1 pW is operating in the room of Problem 4. What sound pressure (on the A-weighted scale) will be registered by a meter 5.2 m from the source? If 125 people are *not* seated in the room, how will it affect the reading? Assume that no one blocks the path of direction propagation between the meter and the signal source.
9. Why does an anechoic chamber provide what we call a free field?

12
Walls, Enclosures, and Barriers

12.1 Introduction

Acoustics constitutes an important factor in building design and in layouts of residences, plants, offices, and institutional facilities. A building not only protects against inclement weather; it must also provide adequate insulation against outside noises from transportation and other sources. Interior walls and partitions need to be designed to prevent the intrusion of sound from one room into another. Exposure of workers to excessive occupational noises can be decreased by construction of appropriate barriers or enclosures around noisy machinery. A shell constructed of the densest materials may be the most effective barrier against noise transmission, but such an enclosure can lose a major portion of its effectiveness if there are weak links that tend to promote sound transmission rather than hinder it. For example, a large proportion of sound energy enters a building through its windows, even closed ones, and many cracks and crevices inevitable in real structures permit sound to enter the structure's interior.

12.2 Transmission Loss and Transmission Coefficients

Sound absorption materials tend to be light and porous. Sound isolation materials, however, generally are massive and airtight, thereby forming effective sound insulation structures between the noise source and the receiver. When airborne sound impinges on a wall, some of the sound energy is reflected, some energy is absorbed within the wall structure, and some energy is transmitted through the wall. Sound pressure against one side of the wall may cause the wall to vibrate and transmit sound to the other side. The amount of incident energy transmitted to the wall depends on the impedance of the wall relative to the air. The amount of the sound transmitted through the wall that is finally transmitted to the air on the receiver side also depends on the impedance of the wall relative to air. A double wall construction that incorporates airspace between panels repeats the process between the two panels, which results in even more insulation against sound.

Most construction materials transmit only a small amount of acoustic energy, with the major portion of the energy undergoing reflection or conversion into heat due to impedance mismatch, absorption within the material, and damping. Heavy walls, usually of masonry, allow very little of the sound to pass through, owing to their high mass per unit area. A wall of gypsum boards mounted on both sides of a stud frame provides effective insulation against sound penetration due to energy losses resulting during the passage of sound from air to solid to air between the two panels to solid and thereon to air on the other side of the wall.

A measure of sound insulation provided by a wall or some other structural barrier is given by the *transmission loss* TL, given in units of dB by

$$\mathrm{TL} = 10 \log \left(\frac{W_1}{W_2} \right) \tag{12.1}$$

where W_1 represents the sound power incident on the wall and W_2 the sound power transmitted through the wall. Because the frequency loss varies with the frequency of the sound, it is usually listed for each octave band or one-third octave band. The fraction of sound power transmitted through a wall or barrier constitutes the *sound transmission coefficient*, τ, written as

$$\tau = \frac{W_2}{W_1} \tag{12.2}$$

Combining Equations (12.1) and (12.2) we obtain

$$\mathrm{TL} = 10 \log \left(\frac{1}{\tau} \right) \tag{12.3}$$

or

$$\tau = 10^{-\mathrm{TL}/10}. \tag{12.4}$$

In Figure 12.1 we consider the case of a plane wave approaching a panel in the y–z plane (i.e., the plane is normal to the x-axis) at an angle of incidence θ. Subscripts I, R, and T are used to identify, respectively, the incident wave, the reflected wave, and the transmitted wave. Arrows indicate the directions of the wave propagation. The wave equation for a plane wave

$$\frac{\partial^2 p}{\partial t^2} = c^2 \nabla^2 p$$

bears the solutions:

$$\begin{aligned} p_I &= \mathbf{P}_I e^{ik(ct - x\cos\theta - y\sin\theta)} \\ p_R &= \mathbf{P}_R e^{ik(ct + x\cos\theta - y\sin\theta)} \\ p_T &= \mathbf{P}_T e^{ik(ct - x\cos\theta - y\sin\theta)} \end{aligned} \tag{12.5}$$

Here the real part of p denotes the sound pressure, $\mathbf{P}$, the (complex) pressure amplitude, and k, the wave number.

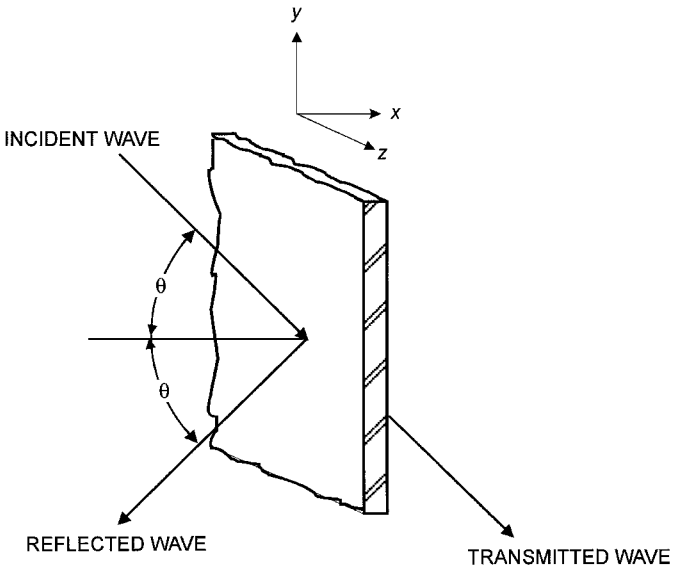


FIGURE 12.1. Transmission loss through a panel in the y–z plane.

12.3 Mass Control Case

Consider the panel as being quite thin, i.e., its thickness is considerably smaller than one wavelength of the sound in air; and let us also neglect the material stiffness and damping in the panel. We can stipulate the conditions of (a) the continuity of velocities normal to the panel and (b) a force balance occurs inclusive of the inertial force. From application of the first condition we express the particle velocity u at the panel as

$$u_{\text{panel}} = u_T \cos\theta = u_I \cos\theta - u_R \cos\theta. \tag{12.6}$$

When the sound pressure and the particle velocity are in-phase they are related by

$$Z = \frac{\rho}{u} = \rho c \tag{12.7}$$

where Z denotes the characteristic resistance, which represents a special case of specific acoustic resistance.

Setting $x = 0$ at the panel, we apply Equation (12.7) and Equation set (12.5) to Equation (12.6) to obtain

$$\mathbf{P}_T = \mathbf{P}_I - \mathbf{P}_R. \tag{12.8}$$

The second condition applied over a unit surface area of the panel gives

$$p_I + p_I - p_T - ma_{\text{panel}} = 0 \tag{12.9}$$

where m represents the mass density per unit area of the panel and a_{panel} the acceleration normal to the surface. Through the use of Equation (12.7) we can

correlate the panel velocity with the transmitted particle velocity:

$$u_{\text{panel}} = p_T \frac{\cos\theta}{\rho c}.$$

Using the last of the Equation set (12.5) in the preceding expression and differentiating with respect to time, we now obtain

$$\begin{aligned} a_{\text{panel}} &= ikc\mathbf{P}_T e^{\frac{ik(ct - x\cos\theta - y\sin\theta)\cos\theta}{\rho c}} \\ &= \frac{i\omega \mathbf{P}_T \cos\theta}{\rho c}. \end{aligned}$$

Inserting the above expression into Equation (12.9) and applying Equations (12.5) at the panel, where $x = 0$, the amplitudes of the pressures are related to each other by

$$\mathbf{P}_I + \mathbf{P}_R - \mathbf{P}_T = \frac{im\omega \mathbf{P}_T}{\rho c}\cos\theta. \tag{12.10}$$

The pressure term $\mathbf{P}_R$ in Equation (12.10) may now be eliminated through the use of Equation (12.8). The ratio of transmitted pressure amplitude to incident pressure amplitude may now be obtained as

$$\frac{\mathbf{P}_T}{\mathbf{P}_I} = \frac{1}{1 + \dfrac{im\omega\cos\theta}{2\rho c}}.$$

We now express the sound transmission coefficient as

$$\tau = \frac{p^2_{\text{rms}(T)}}{p^2_{\text{rms}(I)}} = \frac{|\mathbf{P}_T|^2}{|\mathbf{P}_I|^2} = \frac{1}{1 + \left(\dfrac{im\omega\cos\theta}{2\rho c}\right)^2}.$$

From the definition of Equation (12.3), we obtain the mass law transmission loss equation:

$$\text{TL} = 10\log\left(\frac{1}{r}\right) = 10\log\left[1 + \left(\frac{m\omega\cos\theta}{2\rho c}\right)^2\right]. \tag{12.11}$$

Setting the angle of incidence equal to zero, Equation (12.11) reduces to

$$\text{TL}_0 = 10\log\left[1 + \left(\frac{m\omega}{2\rho c}\right)^2\right] \tag{12.12}$$

which constitutes the statement for the normal incidence mass law for approximating transmission loss of panels with sound at 0° angle of incidence.

12.4 Field Incidence Mass Law

In the situation of transmission loss between two adjoining rooms, the sound source in one room may produce a reverberant field. The incident sound emanating from

the source may strike the wall at all angles between 0 and 90°. For field incidence it is customary to assume that the angle of incidence lies between 0 and 72°, which results in a field incidence transmission loss of approximately

$$\mathrm{TL} = \mathrm{TL}_0 - 5 \text{ dB}.$$

We can modify the mass law equation by converting the angular frequency of sound (rad/s) into cyclic frequency (Hz), i.e. $\omega/2\pi = f$, by setting the acoustic impedance for air $\rho c = 407$ rayls, and assuming $m\omega/(2\rho c) \gg 1$. This gives the field incidence mass law equation

$$\mathrm{TL} = 20\log(fm) - 47 \text{ dB} \tag{12.13}$$

where m denotes the mass per unit area (kg/m^2).

Equation (12.13) indicates for the mass-controlled frequency region that transmission loss of a panel increases by 6 dB per octave, Doubling either the panel thickness or the density of the panel will also engender an additional 6 dB loss for a given frequency. While Equation (12.13) is useful for the prediction of a material's acoustical behavior, laboratory or field testing should be conducted to measure the transmission loss of actual structures under real environmental conditions.

Example Problem 1

A wall is considered to have its sound transmission mass-controlled. Plot the transmission loss as a function of the product of frequency and mass per unit area.

Solution

Using Equation (12.13), which expresses the field incidence mass law for typical conditions, we can write

$$\mathrm{TL} = 20\log(fm) - 47 \text{ dB}.$$

For the normal incidence mass law:

$$\mathrm{TL} = 20\log(fm) - 42 \text{ dB}.$$

Figure 12.2 shows the semilog plot of both relationships as two straight lines. It should be noted here that these values tend to be considerably higher than those of the actual transmission losses.

12.5 Effect of Frequencies on Sound Transmission through Panels

Panel bending stiffness constitutes the governing factor in low-frequency sound transmission. The panel resonances play the principal role in determining the nature of transmission of higher frequency sounds. The panel may be considered mass-controlled in the frequency range from twice the lowest resonant frequency to below the critical frequency (discussed below).

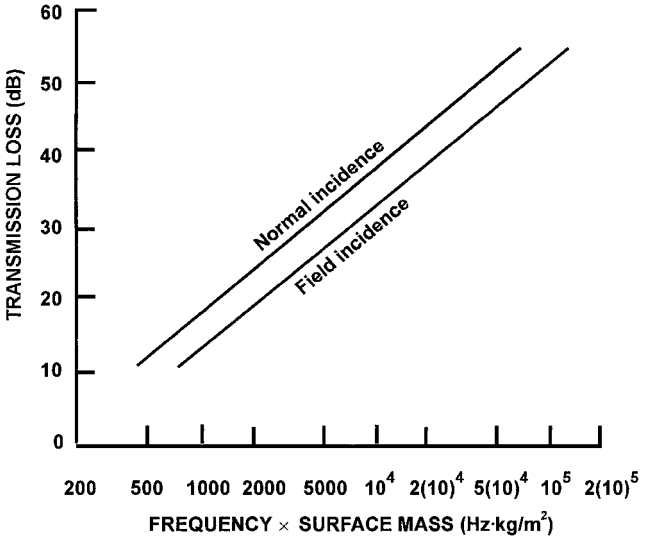


FIGURE 12.2. Semi-log plot of transmission loss versus product of frequency and surface mass.

Ver and Holmer (1971) develop the sound transmission coefficient equation for panels manifesting significant bending stiffness and damping, which is given as follows:

$$\tau = \frac{1}{\left[1+\eta\left(\frac{m\omega\cos\theta}{2\rho c}\right)\left(\frac{B\omega^2\sin^4\theta}{mc^4}\right)\right]^2+\left[\left(\frac{m\omega\cos\theta}{2\rho c}\right)\left(1-\frac{B\omega^2\sin^4\theta}{mc^4}\right)\right]^2}. \tag{12.14}$$

The panel thickness is assumed to be small compared with the wavelength of the incident sound, B = panel bending stiffness (N m), η = composite loss factor (dimensionless), and m = panel surface density (kg/m^2).

12.6 Coincidence Effect and Critical Frequency

In propagating though panels and other structural elements, sound can occur as longitudinal, transverse and bending waves. Bending waves give rise to the coincidence effect. In Figure 12.3 a panel is shown with an airborne sound wave of wavelength λ incident at angle θ. A bending wave of wavelength λ_b is excited in the panel. The propagation velocity of bending waves depends on the frequency, with higher velocities occurring at higher frequencies. The coincidence effect occurs when

$$\theta = \theta^* = \arcsin(\lambda/\lambda_b) \tag{12.15}$$

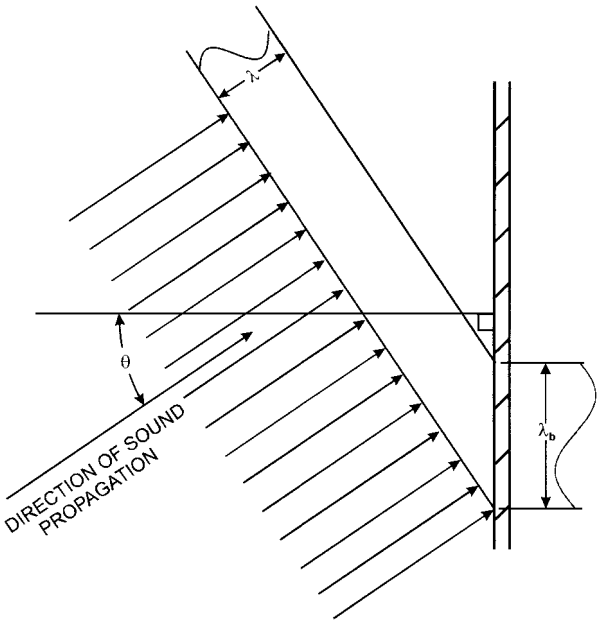


FIGURE 12.3. The coincidence effect affected by bending waves.

or

$$\sin\theta^* = \frac{\lambda}{\lambda_b}.$$

With the asterisk indicating the occurrence of coincidence, θ^* denotes the coincidence angle. Under such circumstances, the sound pressure on the surface of the panel falls into phase with bending displacement. This results in a highly efficient transfer of acoustic energy from the airborne sound waves on the source side of the panel to bending waves in the panel, and thence to airborne sound waves in the receiving room on the other side of the panel. This is a highly undesirable situation if the panel is meant to prevent the transmission of sound from one room to another as a noise control measure.

Figure 12.4 shows an idealized plot of transmission loss for a panel as a function of frequency, with stiffness-controlled, resonance-controlled, mass-controlled, and coincidence-controlled regions. The transmission loss curve drops considerably in the region beyond the critical frequency owing to the coincidence effect.

From Equation (12.15) it can be deduced that the coincidence effect cannot occur if the wavelength λ of the airborne sound is greater than the bending mode wavelength λ_b in the panel. The minimum coincidence frequency, or the *critical frequency*, exists at the critical airborne sound wavelength

$$\lambda = \lambda^* = \lambda_b$$

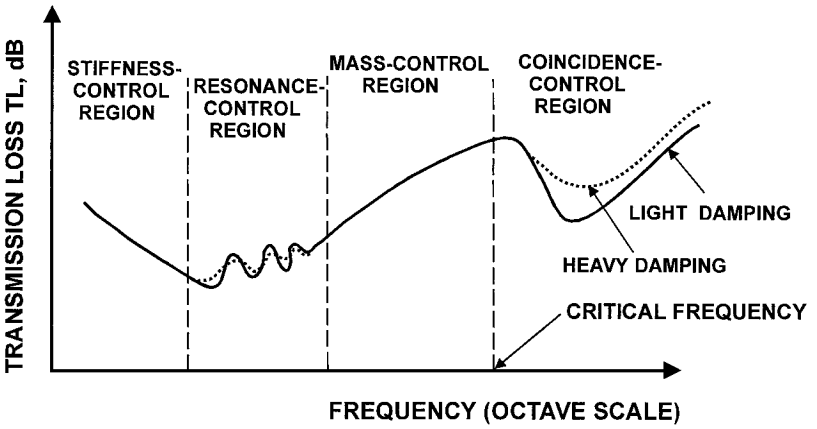


FIGURE 12.4. Idealized plot of transmission loss versus frequency.

where the critical frequency f^* is given by

$$f^* = \frac{c}{\lambda^*}$$

which corresponds to the grazing incidence $\theta = 90°$. Also, there is a critical angle θ^*, at which coincidence will occur, for any frequency above the critical frequency.

Figure 12.5 contains a plot of critical frequencies against thickness for a number of common construction materials. It must be realized that the critical frequency

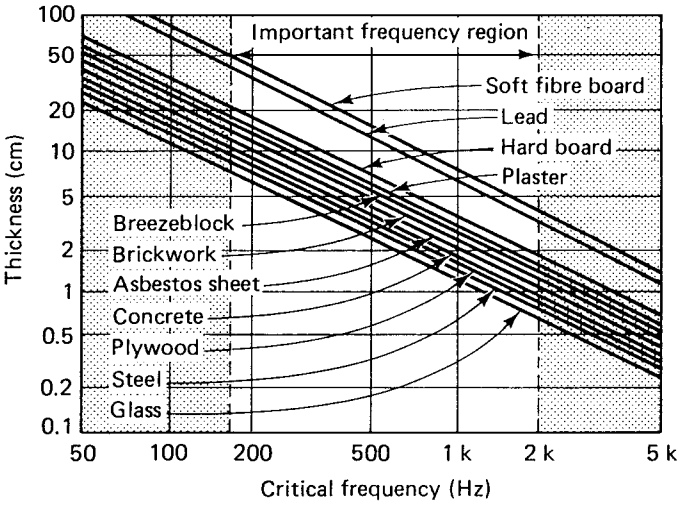


FIGURE 12.5. Thickness versus critical frequency for a number of construction materials. (Source: Brüel and Kjær, 1980)

often falls in the range of speech frequencies, rendering some partitions nearly useless for providing privacy and preventing speech interference.

From Equation (12.14), it is seen that the coincidence effect depends on characteristics of the plate or panel and of the airborne sound wave. At coincidence

$$\frac{B\omega^2 \sin^4 \theta^*}{mc^4} = 1$$

and inserting the above condition into Equation (12.14) yields the sound transmission coefficient for the coincidence condition

$$\tau = \tau^* = \frac{1}{\left[1 + \dfrac{m\omega\eta \cos\theta^*}{2\rho c}\right]^2}.$$

The corresponding transmission loss for the coincidence condition is

$$\mathrm{TL} = \mathrm{TL}^* = 10 \log\left(\frac{1}{\tau^*}\right) = 20 \log\left(1 + \frac{fm\eta \cos\theta^*}{\rho c}\right).$$

From the above equations, it would appear that $\tau^* = 1$ and $\mathrm{TL}^* = 0$ for undamped panels (in which the loss factor $\eta = 0$). The above transmission loss equation is premised on the theoretical behavior of an infinite plate, and the finite boundaries of actual structures such as windows and walls and the presence of damping in real construction materials will produce a different response to sound waves.

12.7 The Double-Panel Partition

A single-panel wall can exhibit resonant frequencies that fall below the range of speech frequencies. Most walls are constructed of two panels with an airspace between them, and they may yield low-frequency resonances in the speech range. A typical interior partition consists of two gypsum board panels (ranging in thickness from 1.3 cm to 1.9 cm, or from 0.5 in. to 0.75 in.) separated 9 cm (3.5 in.) by 5 cm × 10 cm (2 × 4)[1] wood or metal studs. In Figure 12.6 a double-panel configuration is shown; the two panels of mass per unit area m_1 and m_2, respectively, are separated by an airspace h. Because the air entrapped between the two panels behaves as a spring, a spring-mass analogy, shown in Figure 12.7, can be applied, with k representing the spring constant between two masses. The wall response mode can be depicted by two masses vibrating at the same frequency. A node (i.e. a "motionless" point) on the spring exists, thus effectively resolving the spring into two springs with spring constants (or spring rates) k_1 and k_2, The natural frequency of the dual-mass system is given by

$$f_n = \frac{\sqrt{k_1/m_1}}{2\pi} = \frac{\sqrt{k_2/m_2}}{2\pi}. \tag{12.16}$$

[1] Lumber sizes are given by figures that are almost always nominal rather than representative of actual values. A two-by-four stud generally measures 1.75 × 3.5 in.

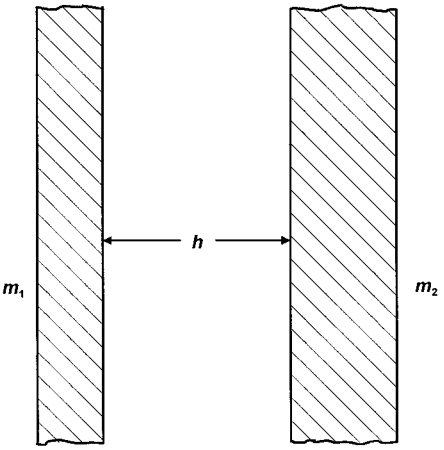


FIGURE 12.6. A double-panel partition.

The individual spring constants are related to the spring constant of the composite spring by

$$\frac{1}{k} = \frac{1}{k_1} + \frac{1}{k_2}. \tag{12.17}$$

Eliminating k_2 between Equations (12.16) and (12.17) results in

$$k_1 = k\frac{m_1 + m_2}{m_2}.$$

Inserting the last expression into the first portion of Equation (12.16) yields the natural frequency f_n of the system:

$$f_n = \frac{\sqrt{k\dfrac{m_1 + m_2}{m_1 m_2}}}{2\pi}. \tag{12.18}$$

In order to establish the effective spring constant of the air between the panels, it will be assumed that an isentropic process constitutes the action of sound waves, since the pressure changes occur too rapidly for an isothermal process to take place. From thermodynamics, for an isentropic process the pressure p of air (considered

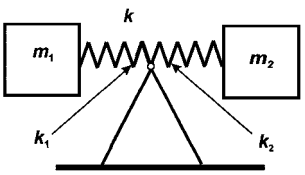


FIGURE 12.7. Spring-mass analogy of the double-panel partition of Figure 12.6.

an ideal gas) varies in the following manner:

$$pv^{\gamma} = \text{constant} \tag{12.19}$$

where v represents the specific volume (equal to the reciprocal of density) and γ, the ratio of specific heats. Differentiating Equation (12.19) gives

$$\frac{dp}{dv} = -\gamma \frac{p}{v}. \tag{12.20}$$

Sound waves cause the ambient pressure and specific volume of air to vary only slightly from the quiescent values of p_0 and v_0, and thus the instantaneous values p and v differ very little from p_0 and v_0 in Equation (12.20). Because the mass of air entrapped between the panels remains constant, the ratio of panel displacement (arising from the sound pressure pushing on the panels) dh to the original (quiescent) spacing h_0 should equal the ratio of the change of specific volume to the quiescent value of the specific volume:

$$\frac{dh}{h_0} = \frac{dv}{v_0} \tag{12.21}$$

But the spring constant represents the force per unit displacement, or for a unit area of the wall

$$k = -\frac{dp}{dh}. \tag{12.22}$$

Combining Equations (12.21) and (12.22) and dropping the subscripts (since $h_0 \approx h$, $p_0 \approx p$), we now have

$$k = -\frac{\gamma p}{h}$$

which, upon inserting into Equation (12.19), gives

$$f_n = \frac{\sqrt{\gamma p \dfrac{m_1 + m_2}{h m_1 m_2}}}{2\pi}.$$

Setting $\gamma = 1.4$ and $p = 101.325$ kPa (the standard atmosphere), the low-frequency resonance of the double-panel wall may now be found from

$$f_n = 60\sqrt{\frac{m_1 + m_2}{h m_1 m_2}}. \tag{12.23}$$

The surface masses m_1 and m_2 are given in kg/m^2 and the panel spacing h in meters.

Example Problem 2

Predict the transmission loss for an 8-in. wall of poured concrete for 800-Hz sound. The concrete has a density of approximately 150 lb/ft^3 (2406 kg/m^3).

Solution

Surface density $m = 8$ in. $\times$ 1 m/(39.37 in.) $\times$ 2406 kg/m^3 = 488.9 kg/m^2

Using field incidence law Equation (12.13)

$$\text{TL} - 20\log(fm) - 47 = 20\log(800 \times 488.9) - 47 = 65 \text{ dB}$$

Laboratory tests indicate a TL closer to 50 dB. Actual installations may yield even lower values because of flanking noise transmission paths.

Example Problem 3

Find the resonant frequency of a double-panel wall constructed of a 15-kg/m^2 and a 20-kg/m^2 gypsum board with 9-cm airspace.

Solution

Applying Equation (12.23):

$$f_n = 60\sqrt{\frac{m_1 + m_2}{h\, m_1 m_2}} = 60\sqrt{\frac{15 + 20}{0.09(15)(20)}} = 68 \text{ Hz}.$$

12.8 Measuring Transmission Loss

Recasting Equation (12.1) for transmission loss TL in terms of vector sound intensity, we write

$$\text{TL} = 10\log\left(\frac{\mathbf{I}_I}{\mathbf{I}_T}\right)$$

where $\mathbf{I}_I$ = incident vector sound intensity and $\mathbf{I}_T$ = transmitted vector sound intensity, For sound pressure level and particle velocity in phase, sound intensity is given by

$$I = \frac{p_{\text{rms}}^2}{\rho c}$$

with the result that transmission loss could be restated as

$$\text{TL} = 10\log\left(\frac{p_{\text{rms}I}^2}{p_{\text{rms}T}^2}\right).$$

But the above equation is not usable for measurement purposes, since distinction cannot be made between root-mean-square (rms) pressures attributable to incident sound, transmitted sound and reflected sound.

In order to determine the transmission loss, it is required to set up a source chamber and a receiving chamber, with a panel of the specimen material installed in a window between the two chambers. Such a setup is shown in Figure 12.8. The chambers must be designed so as to minimize the sound transmission paths other

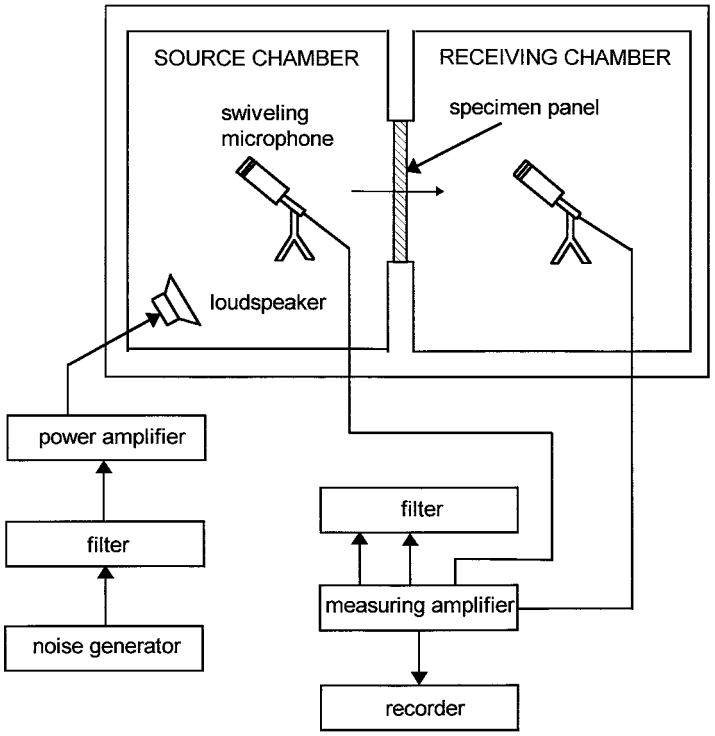


FIGURE 12.8. Setup for a transmission loss measurement.

than that through the evaluation specimen. Sound level is measured in the source chamber and the receiving chamber with a microphone system, a measuring amplifier, and filter. The measured sound is filtered to mitigate the effect of background noise on the measurement procedures, with the filter settings being identical for the generated sound and the measuring sound filters. The sound level may be averaged by rotating the microphone about its base on the mounting tripod. The transmission loss is found by applying the measured results in the following equation:

$$\mathrm{TL} = L_S - L_R + 10 \log \left(\frac{A_m}{A_R} \right)$$

where

L_S = average sound level in the source chamber
L_R = average sound level in the receiving chamber
A_m = area of the material under investigation
A_R = equivalent absorption area of the receiving room.

Another measurement setup is given in Figure 12.9. A pink noise generator, filters, audio power amplifier, and speaker constitute the sound source system. A computer-controlled analyzer incorporates a random noise generator and

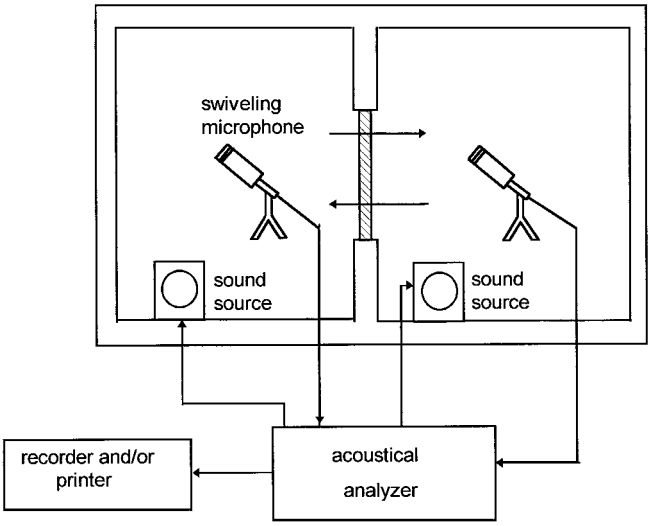


FIGURE 12.9. An alternative to the setup of Figure 12.8, which provides for automatic evaluation of transmission loss. The sound source can consist of a pink noise generator, filters, power amplifier, and speaker. The analyzer, which can generate random noise, can automatically position a microphone boom for spatial and temporal averaging.

provisions for automatic remote control of the microphone boom for spatial and temporal averaging. Such an analyzer can measure in one-third octave bands the following parameters: source chamber level, receiving chamber background SPL, receiving chamber SPL, and receiving chamber reverberation time.

12.9 Flanking

Flanking occurs when noise from a sound source is transmitted through paths other than direct transmission through a wall between the source chamber and the receiver on the opposite side of the wall. Among the causes of flanking are acoustical leaks through cracks around doors, windows and electrical outlets, passage of the sound through suspended ceilings with resulting reflection into adjacent rooms, HVAC (heating/ventilation/air-conditioning) ducts, floor and ceiling joists. The insulation effect of a partitioning wall is effectively decreased by flanking and acoustic leaks, examples of which are shown in Figure 12.10.

12.10 Combined Sound Transmission Coefficient

A wall may contain a number of elements such as windows, doors, openings, and cracks. The effective or combined sound transmission coefficient depends on

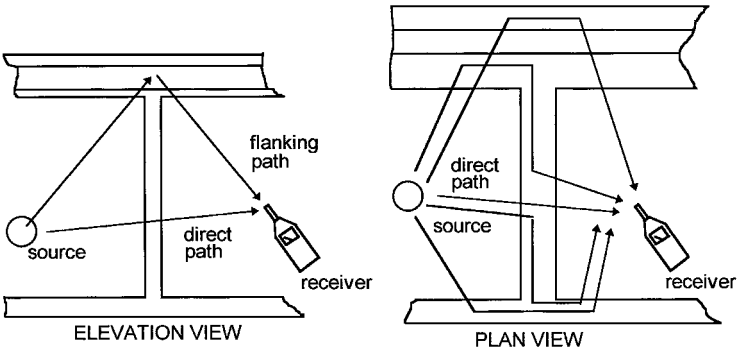


FIGURE 12.10. Examples of flanking.

the values of the sound transmission coefficients of the individual elements and their respective areas. Assuming the incident sound to be fairly uniform over the individual elements, we can compute the combined sound transmission coefficient from

$$\tau_{\text{combined}} = \frac{\sum_{i=1}^{n} A_i \tau_i}{\sum_{i=1}^{n} A_i} \tag{12.24}$$

where i designates the i-th element of the subdivided wall area A and the corresponding value of τ. For cracks and open areas, $\tau = 1$, which is to say the sound transmission through such crevices may be considered to be virtually unimpeded. The corresponding transmission loss of the composite wall then becomes

$$\text{TL}_{\text{combined}} = 10 \log \left(\frac{1}{\tau_{\text{combined}}} \right) = 10 \log \left(\frac{\sum_{i=1}^{n} A_i}{\sum_{i=1}^{n} A_i 10^{-\text{TL}_i/10}} \right). \tag{12.25}$$

Example Problem 4

A 6.0 m $\times$ 2.5 m wall is for the most part constructed of 8-cm thick dense poured concrete with a transmission loss of 52 dB at 500 Hz. An opening in the wall is provided for a 2.2 m $\times$ 1.0 m wood door with a transmission loss of 25 dB at 500 Hz. There is a crack across the width of the door that is 1.0 cm high. Estimate the effective transmission loss of the composite wall at 500 Hz.

Solution

The total area A of the wall is $6.0 \times 2.5 = 15.0$ m^2, but the concrete portion of the wall is $15.0 - 2.2 \times 1.0 - 0.01 \times 2.2 = 12.778$ m^2. Applying

Equation (12.25)

$$\begin{aligned} \mathrm{TL}_{\text{combined}} &= 10\log\frac{15.0}{12.778\times10^{-55/10}+2.2\times10^{-25/10}+0.022\times10^{-0/10}} \\ &= 27.1\ \text{dB} \end{aligned}$$

Suppose that a door with TL = 45 dB was installed instead; what would be the new value of the composite TL for the wall?

Solution

$$\begin{aligned} \mathrm{TL}_{\text{combined}} &= 10\log\frac{15.0}{12.778\times10^{-55/10}+2.2\times10^{-45/10}+0.022\times10^{-0/10}} \\ &= 28.3\ \text{dB} \end{aligned}$$

This example shows that negligible improvement will occur because the crack constitutes the principal means of negating the combined effectiveness of the acoustic insulation of the concrete wall and the door. The use of an acoustical door which effectively seals the area will result in an appreciably improved value of the transmission loss.

12.11 Noise Insulation Ratings

Two principal methods of rating sound insulation are discussed in this section, namely *Sound Transmission Class* (*STC*) and *Shell Isolation Rating* (*SIR*). The former designation constitutes a single-number description of noise insulation effectiveness of a structural element and it is widely used to describe the characteristics of interior partitions or walls with respect to noise occurring in the range of speech and music frequency. The SIR methodology was developed by Pallett *et al*. (1978) at the U.S. National Bureau of Standards (now National Institute of Science and Technology) in order to establish a simple system to predict the capacity of building shells to attenuate transportation noise.

Sound Transmission Class (STC)

The values of STC are computed from transmission loss values measured in one-third octave bands in the 125 Hz–4 kHz range. Figure 12.11 illustrates the standard STC contour. This contour begins at 125 Hz, sloping upward at 3 dB per one-third octave (i.e., 9 dB per octave) until it reaches 400 Hz. The slope of the contour then changes at 400 Hz to 1 dB per one-third octave (3 dB per octave), and it remains constant at 1.25 kHz. In the range from 1.25 to 4 kHz the slope is zero. The STC single-number classification of a specific wall is designated by the value of TL at 500 Hz. If the values of TL for the wall at the 16 one-third octave points are

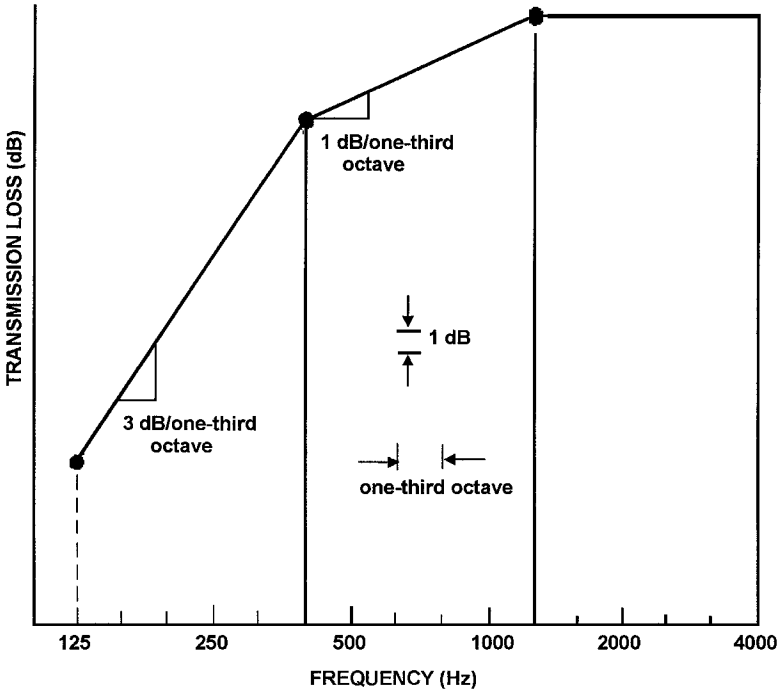


FIGURE 12.11. The standard STC contour.

known for the region from 125 Hz to 4 kHz and plotted, the STC for the wall is established by superimposing the contour of Figure 12.11 upon the TL curve so that (a) there is not more than 8-dB deficiency between the TL and the STC contour at any one-third octave frequency and (b) the total deficiency between the STC contour (i.e., the value of the STC contour less the value of the TL curve summed at all one-third octave frequencies from 125 Hz to 4 kHz) must be $\leq$32 dB. When the curve is adjusted to meet these two criteria, the STC value of the wall is taken to be equal to the value of the TL of that contour at 500 Hz.

Example Problem 5

Find the STC value of the TL curve of Figure 12.12.

Solution

The standard STC contour is overlaid on the TL plot and positioned to satisfy the two criteria described above. From the value of the contour at 500 Hz, it is seen that the STC rating of the wall is approximately 48 dB.

Table 12.1 lists the STC ratings of a number of structural elements.

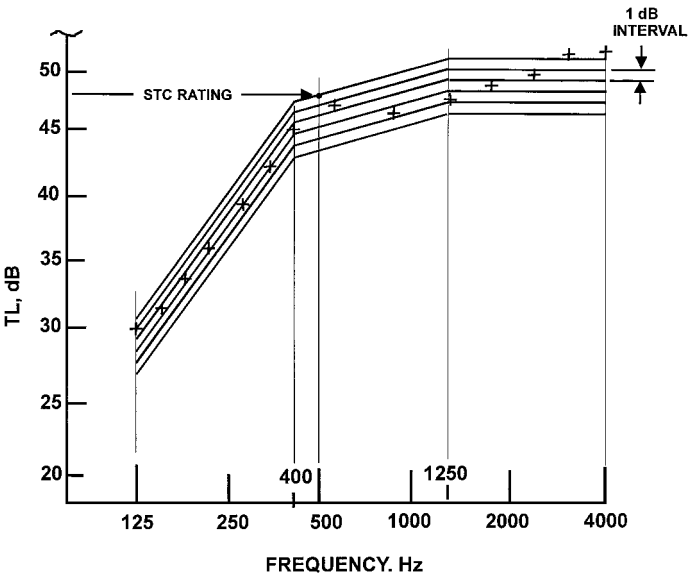


FIGURE 12.12. STC overlay on a TL versus frequency plot.

Shell Isolation Rating (SIR)

The SIR method stems from intensive studies conducted by investigators at the National Institute of Standards and Technology in the effort to develop a simple system to predict the attenuation of *A*-weighted transportation noise by building

TABLE 12.1. Sound Transmission Class (STC) Ratings of Common Construction Materials

Description	STC
4-in.- (100-mm) thick brick wall with airtight joints	40
4-in.- (100-mm) thick brick wall with gypsum board on one side	45–50
8-in.- (200-mm) thick brick wall with gypsum board on one side	50–60
4-in.- (100-mm) thick hollow concrete block with airtight joints	36–41
4-in.- (100-mm) thick hollow concrete block with gypsum board on one side	42–48
8-in.- (200-mm) thick hollow concrete block with airtight joints	46–50
8-in.- (200-mm) thick hollow concrete block with gypsum board on both sides	50–55
2 × 4-in.- (nominal) wood or metal stud wall with gypsum board on	33–43
both sides, spackled at joints, floor, and ceiling	20–30
Single-glazed window	26–44
Double-glazed window	20–27
Hollow core wood or steel door	38–55
Specially mounted acoustical door	70
Double-walled soundproof room, 12-in.- (300-mm) thick walls including airspace	21–29
Quilted fiberglass mounted on vinyl or lead (limp mass) barrier septrum	

shells. This method is based on statistical studies, and +3 dB per octave was chosen as the standard SIR contour. The STC contour method is generally preferred for describing the noise attenuation effectiveness of building structures in the presence of non-transportation sounds, viz. speech, television, and radio. The SIR system is favored for the evaluation of noise reduction provided by building shells against transportation noise.

An element SIR (or member SIR) is an estimate of the difference in the *A*-weighted sound levels when a structural element or member is placed between the transportation noise source and the receiver. It is assumed that all noise transmission occurs through the subject element. A *room SIR* (or *composite SIR*) refers to the estimate of the *A*-weighted sound level difference caused by the presence of all members that act as noise transmission paths between the source and the receiver.

An estimate of the member SIR may be obtained from a set of transmission loss measurements in one-third octave or octave bands in the following manner:

1. The transmission loss is plotted against frequency.
2. The +3 db per octave SIR reference contour is plotted on an overlay sheet, which is then superimposed on the TL versus f plot in the highest position so that the sum of the deficiencies is less than twice the number of test frequencies. A deficiency is defined as the number of decibels by which the SIR reference value exceeds the measured TL of the member.
3. The value of TL at 500 Hz on the SIR reference curve constitutes the value of the SIR of the member.

The statistical quality of SIR prediction is improved by using a larger number of TL laboratory measurements, and the one-third octave band measurements will generally yield more reliable results than those based on octave bands. Manufacturers' catalogues frequently report the one-third octave TL measurements used to evaluate STC. An example of an SIR determination is given in Figure 12.13 for the TL plot of Figure 12.12. Much of the tedium of SIR contour fitting can be eased through the use of computer plotting.

The field incidence mass law, expressed by Equation (12.13), can be utilized to extrapolate data. If the transmission loss is known for all applicable frequencies for a structural element 1, and it is desired to find the TL of a similar element with a different surface mass, then

$$\mathrm{TL}_2 = \mathrm{TL}_1 + 20 \log \left(\frac{m_2}{m_1} \right).$$

If the relationship holds in the applicable frequency range, the values of SIR are similarly related.

Example Problem 6

A shell constructed of 5-in. thick concrete with 55 lb/ft^2 surface mass carries a rating of SIR 43. Estimate the SIR rating for an 8-in. thick concrete with a surface mass of 92 lb/ft$^{2.}$

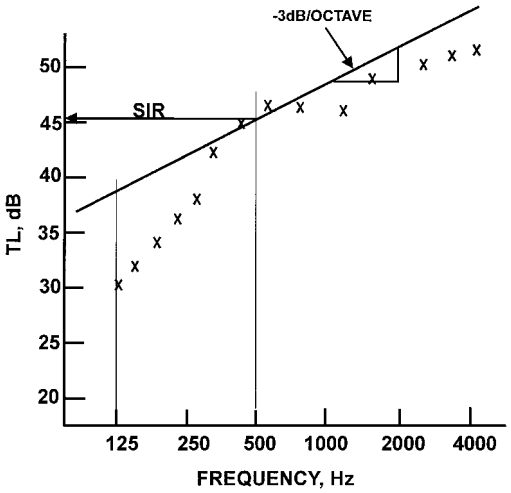


FIGURE 12.13. SIR determination for TL versus f plot of Figure 12.12.

Solution

$$\begin{aligned} \text{SIR}_{8\,\text{in.}} &= \text{SIR}_{5\,\text{in.}} + 20\log\left(\frac{m_{8\,\text{in.}}}{m_{4\,\text{in.}}}\right) \\ &= 45 + 20\log\left(\frac{92}{55}\right) = 49.5\ \text{dB} \end{aligned}$$

Table 12.2 gives the SIR values of a number of structural elements.

12.12 Noise Reduction of a Wall

Let us consider two rooms that are acoustically separated by a partition of area S_w in Figure 12.14. L_{p1} and L_{p2} denote the spatially averaged values of the sound pressure levels in room 1 and room 2, respectively. We assume the noise source in room 1 produces a purely reverberant field near the partition. This occurs if the sound pressure level can be described by Equation (11.23):

$$L_{p1} = L_{w1} + 10\log\left(\frac{4}{R_1}\right)$$

where $4/R_1 \gg Q/(4\pi r^2)$, R_1 is the room constant of room 1, and r the distance from the source in room 1 to the position of measured L_{p1}. This assumption permits us to simplify the analysis by assuming a constant sound pressure over the entire area of the dividing wall.

TABLE 12.2. Shell Isolation Rating (SIR).

Description	Weight (lb/ft^2)	SIR rating
Dense poured concrete or solid block walls		
4 in. thick	50	41
6 in. thick	73	43
8 in. thick	95	46
12 in. thick	145	49
16 in. thick	190	51
Hollow concrete block walls		
6 in. thick	21	41
8 in. thick	30	43
Brick veneered frame walls	—	48–53
Stuccoed frame walls	—	34–52
Frame walls with wood siding	—	33–40
Metal walls, curtain walls	—	25–39
Shingled wood roof with attic	10	40
Steel roofs	—	36–51
Fixed, single-glazed windows (the higher SIR values apply to windows with special mountings and laminated glass)	—	22–39
Fixed double-glazed windows (the higher SIR values apply to windows with large spaces between the glass and special mountings)	—	22–48
Double-hung windows		
Single-glazed	—	20–24
Double-glazed	—	20–29
Casement windows, single-glazed	—	19–29
Horizontal sliding windows		16–24
Glass doors	2.3–3	24
Wood doors (the higher SIR values apply to weatherstripped doors)		
Hollow core	1.2–5	14–26
Solid core	4–5	16–30
Steel doors (the higher SIR values apply to acoustical doors)	4–23	21–50

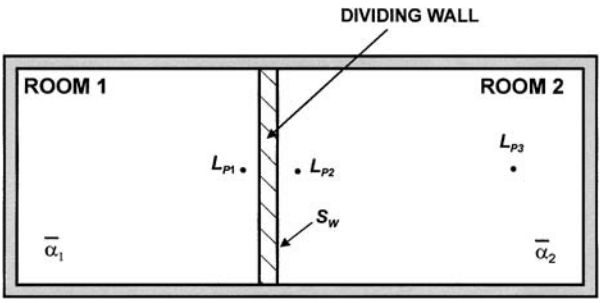


FIGURE 12.14. Two rooms separated by a dividing wall of area S_w.

The partition absorbs a certain amount of power W_α from the reverberant field in room 1, namely

$$W_\alpha = W_r \left(\frac{S_w \alpha_w}{S_1 \overline{\alpha}_1} \right) \tag{12.26}$$

where

W_α = power absorbed by the dividing wall
W_r = power in the reverberant field
S_w = area of the wall
α_w = absorption coefficient of the wall
S_1 = total surface area of room 1
$\overline{\alpha}_1$ = average absorption coefficient.

Let us assume that all the power incident upon the wall will be absorbed, i.e., $\alpha_w = 1$. The portion of the power W_1 of the source in room 1 that becomes the power in the reverberant field is given by

$$W_r = (1 - \bar{\alpha}) W_1$$

Substituting the above into Equation (12.26) gives

$$W_\alpha = (1 - \bar{\alpha}) W_1 \frac{S_w}{S_1}. \tag{12.27}$$

The power W_α absorbed by the wall can be expressed in terms of the room constant R_1, so Equation (12.27) becomes

$$W_\alpha = W_1 S_w \frac{1 - \bar{\alpha}_1}{S \bar{\alpha}_1} = \frac{W_1 S_w}{R_1}$$

which can now be combined with Equation (12.2) to yield

$$W_2 = \frac{W_1 S_w \tau}{R_1} \tag{12.28}$$

where W_2 represents the power transmitted into room 2 and τ the transmission coefficient of the wall.

We can consider the direct field in the region near the partition in room 2 to be a uniform plane wave progressing outward from the radiating wall. The energy in the direct field equals the product of the power transmitted into the room multiplied by the time required for the plane wave to transverse the room. This time is given by $t = L/c$, where L denotes the length of room 2. The energy density δ_{d2} of the direct field is given by the directed field energy divided by the room volume $V = S_w L$:

$$\delta_{d2} = \frac{W_2}{V} \frac{L}{c}. \tag{12.29}$$

From Equation (11.21), the reverberant energy is given by

$$\delta_{r2} = \frac{4 W_2}{c R_2}. \tag{12.30}$$

Combining the last two equations gives the total energy δ_2 density near the wall:

$$\delta_2 = \left(\frac{1}{S_w} + \frac{4}{R_2}\right)\frac{W_2}{c}. \tag{12.31}$$

Equation (12.28) can now be used to eliminate W_2 in Equation (12.31) to yield

$$\delta_2 = \frac{4}{R_1}\frac{W_1}{c}\left(\frac{1}{4} + \frac{S_w}{R_2}\right)\tau. \tag{12.32}$$

The mean-square pressure in room 2 is given in terms of energy density by

$$p_2^2 = \rho_0 c^2 \delta_2$$

which is then inserted into Equation (12.32) to yield

$$p_2^2 = \frac{4W_1}{R_1}\rho_0 c\tau\left(\frac{1}{4} + \frac{S_w}{R_2}\right). \tag{12.33}$$

From the use of the definitions

$$L_{p2} = 10\log\left(\frac{p_2}{20\,\mu\text{Pa}}\right)^2 \qquad \text{and} \qquad L_{W1} = 10\log\left(\frac{W_1}{10\,\text{p}W}\right)$$

Equation (12.33) becomes

$$L_{p2} = L_{w1} + 10\log\left(\frac{4}{R_1}\right) - 10\log\left(\frac{1}{\tau}\right) + 10\log\left(\frac{1}{4} + \frac{S_w}{R_2}\right) \tag{12.34}$$

where we have assumed that $\rho_o c = 407$ rayls.

In inspecting Equation (12.34) we observe that the first two terms in the right-hand side of the equation represents L_{p1} under the conditions of a reverberant field near the wall in room 1. We also invoke Equation (12.3) which expresses TL in terms of transmission coefficient τ, with the result that Equation (12.34) simplifies to:

$$L_{p2} = L_{p1} - \text{TL} + 10\log\left(\frac{1}{4} + \frac{S_w}{R_2}\right). \tag{12.35}$$

From Equation (12.35) one can estimate the sound pressure level L_{p2} near the wall in room 2, given the TL of the wall and the acoustic parameters of room 2. On the other hand, if we know the desired value of L_{p2}, (which is the usual case) we can rearrange Equation (12.35) to find the necessary transmission loss of the wall as follows:

$$\text{TL} = L_{p1} - L_{p2} + 10\log\left(\frac{1}{4} + \frac{S_w}{R_2}\right). \tag{12.36}$$

The above equation is valid in both English and metric units.

The term $L_{p1} - L_{p2}$ is referred to as the *noise reduction* denoted by the term NR. Equation (12.36) becomes

$$\text{NR} = \text{TL} - 10\log\left(\frac{1}{4} + \frac{S_w}{R_2}\right). \tag{12.37}$$

We note if room 2 is entirely absorbent, i.e., it supports no reverberation field or if the wall is an outside wall that it radiates outdoors, the value of R_2 approaches an infinite value and Equation (12.37) revises to

$$\text{NR} = \text{TL} + 6 \text{ dB}. \tag{12.38}$$

Example Problem 7

A 20 ft × 8 ft wall with a transmission loss of 32 dB separates two rooms. Room 1 contains a noise source that produces a reverberant field with a SPL = 110 dB near the wall. Room 2 has a room constant R_2 of 1350 ft^2. Determine the sound pressure level near the wall in room 2.

Solution

We apply Equation (12.32) to find the value of L_{p2}:

$$\begin{aligned} L_{p2} &= 110 - 32 + 10 \log \left(\frac{1}{4} + \frac{20 \times 8 \text{ ft}^2}{1350 \text{ ft}^2} \right) \text{dB} \\ &= 73.7 \text{ dB} \end{aligned}$$

12.13 Sound Pressure Level at a Distance from the Wall

For the most part we wish to predict the sound pressure level at some distances from the wall. For appreciable distances the reverberant field will dominate over the direct field. In Figure 12.14, it is desired to find L_{p3} at a point located somewhat far from the wall, which results from the sound pressure level L_{p1} in room 1. But it should be noted that the difference $(L_{p1} - L_{p3})$ does not represent noise reduction, because region 3 in the chamber of room 2 is not directly contiguous to the wall.

An expression for L_{p3} will be derived on the basis that it consists almost entirely of the energy density in the reverberant field, i.e., the energy density is given by

$$\delta_{r2} = \frac{4W_2}{R_2 c} = \delta_3 \tag{12.39}$$

but since $p^2 = \rho_0 c^2 \delta$, Equation (12.39) becomes

$$p_3^2 = \frac{4W_2 \rho_0 c}{R_2}. \tag{12.40}$$

Here p_3^2 represents the mean-square pressure in those regions of room 2 where the direct field emerging directly from the wall as a plane wave has already dispersed

to the extent that the field can be considered a superposition of randomly reflected components, i.e., it is essentially a reverberant field. Equation (12.28) can be used to eliminate W_2 in favor of W_1 in Equation (12.40) which then becomes

$$p_3^2 = \rho_0\, c \frac{4W_1}{R_1}\frac{S_w}{R_2}\tau$$

and because $\tau = W_2/W_1$, $L_{p3} = 10\log(p^3/20\ \mu\text{Pa})$, we will obtain

$$L_{p3} = L_{p1} + 10\log\left(\frac{S_w}{R_2}\right) - \text{TL} \tag{12.41}$$

where $\rho_o c = 407$ rayls, and L_{p3} denotes the sound pressure level at a distance sufficiently far from the wall that the direct field is negligible in comparison with the reverberant field.

Example Problem 8

Find the sound pressure level of room 2 at a large distance from the wall in the Example Problem 7.

Solution

Applying Equation (12.41) yields

$$\begin{aligned} L_{p3} &= L_{p1} + 10\log\left(\frac{S_w}{R_2}\right) - \text{TL} \\ &= 110 + 10\log\left(\frac{20\times 8}{1350}\right) - 32 = 68.7\ \text{dB} \end{aligned}$$

12.14 Enclosures

Enclosures may be categorized as being either *full enclosures* or *partial enclosures*. These structures of varying sizes may enclose people or noise generating machinery. It is advisable to never enclose more volume than necessary. For example, an entire machine should not be enclosed if only one of its components (such as a gear box, motor, etc.) constitutes the principal noise source. In some cases, a partial enclosure may suffice to shield a worker from excessive exposure to noise. The walls of an enclosure should be constructed of materials that will provide isolation, absorption, and damping—all necessary for effective noise reduction. Moreover, any presence of cracks or leaks can radically reduce the noise reduction of an enclosure, so all mechanical, electrical, utility, and piping outlets must be thoroughly sealed. Access panels should fit tightly, and viewing windows should be constructed of double panes and be impervious to acoustical leakages. The interior of the enclosures should be lined with highly absorbing material so

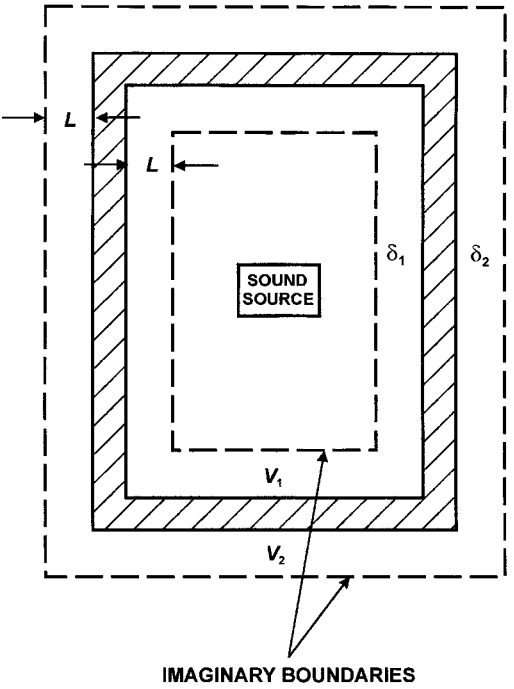


FIGURE 12.15. Plan view of a hood with energy densities δ_1 and δ_2 just inside and outside the hood.

that the sound level does not build up from reflections, thereby decreasing the wall vibration and the resultant radiation of the noise.

Hoods constitute a special category of acoustical full enclosures designed specifically for the purpose of containing and absorbing excessive noise from a machine. A hood is designed to minimize leakages at its physical input and output, and access must be provided to allow periodic servicing. Usually hoods are sized so that the enclosed machine component does not occupy more than a third of the internal volume. The effectiveness of a hood is denoted by the amount of *noise reduction* and *insertion loss*.

In the equilibrium condition, the total power W_1 radiated by a source is absorbed by the interior walls of the hood. In Figure 12.15 δ_1 and δ_2, respectively, denote the energy densities of the interior and the exterior of the hood in the immediately vicinity of the enclosure wall. We assume the wall to be thin enough so that volumes V_1 and V_2 are nearly equal and denoted by V. As the time t required for an acoustic wave to travel distance L is given by L/c, the energy contained in volume V_1 may be approximated by $E_1 = W_1 L/c$, with the energy density now given by

$$\delta_1 = \frac{E_1}{V_1} = \frac{W_1 L}{Vc}$$

and similarly for the external side of the wall

$$\delta_2 = \frac{E_2}{V_2} = \frac{W_2 L}{Vc}$$

and because the transmission coefficient $\tau = W_2/W_1$

$$\delta_2 = \frac{\tau W_1 L}{Vc} = \tau \delta_1.$$

Using the relationship $p_1^2 = \rho_0 c^2 \delta_1$ and $p_2^2 = \rho_0 c^2 \delta_2$, and the fact $L_p = 10 \log (p/p_{\text{ref}})^2$ we readily obtain

$$L_{p1} = 10 \log \left(\frac{\rho_0 c^2 \delta_1}{p_{ref}^2} \right) \quad \text{and} \quad L_{p2} = 10 \log \left(\frac{\rho_0 c^2 \delta_2}{p_{ref}^2} \right).$$

Noise reduction NR is given by

$$\text{NR} = L_{p1} - L_{p2} = 10 \log(1/\tau)$$

which we note is also the definition for transmission loss TL. Hence,

$$\text{NR} = \text{TL}$$

which indicates that the noise reduction from just within the hood (located in a virtual free field) to a region very near the external hood wall is equal to the transmission loss of the wall alone. It should be realized here the value of L_{p1} is supposed to be the value obtained with the hood in place, not the original *lower* value of L_{p1} measured in the vicinity of the noise source before the hood is placed over it. L_{p2} will be correspondingly higher, as the result of the hood insertion. As a result, it is more useful to know the insertion loss IL of the hood rather than the noise reduction. The insertion loss IL is given by

$$\text{IL} = L_{p0} - L_{p2} \tag{12.42}$$

where L_{p0} denotes the sound pressure level at a selected location without the hood; and L_{p2}, the sound pressure level at the same point with the hood enshrouding the sound source.

In deriving an approximate expression for insertion loss, we shall assume that the room is considerably larger than the hood. The sound pressure level L_{p0} at any location in the room may be found from Equation (11.21)

$$L_{p0} = L_{w0} + 10 \log \left(\frac{Q_0}{4\pi r^2} + \frac{4}{R} \right). \tag{12.43}$$

In a similar fashion we can express SPL at the same location with the hood in place by

$$L_{p2} = L_{w2} + 10 \log \left(\frac{Q_2}{4\pi r^2} + \frac{4}{R} \right). \tag{12.44}$$

We now need to determine the total power W_2 emitted by the noise source and its hood acting as a single unit. Since the combination of noise source (usually

a machine) and hood maintains a fairly equilibrium condition, the space average of the time-average energy density remains a constant value under the hood. The entire acoustic power emitted by the source is absorbed by the hood as losses or is radiated through the walls and from thence outside the hood. That amount of acoustic energy which passes through the hood is W_2, which can be approximated by Equations (12.26) and (12.28):

$$W_2 = W_0 \left(\frac{S_h \alpha_h}{\bar{\alpha}} \right) \bar{\tau} \tag{12.45}$$

where

W_2 = power radiated into the room by the hood
W_0 = acoustic power of the source
S_h = area of the walls and ceiling of the hood
α_h = absorption coefficient of the walls and ceiling
S = total area under the hood
$\bar{\alpha}$ = average value of sound absorption coefficient under the hood
$\bar{\tau}$ = average value of transmission coefficient for the hood,excluding the floor.

We shall now simplify Equation (12.45) by assuming that the total surface $S = S_h + S_f$, where S_f is the floor area, is such that $S_f \ll S_h$ so that $S \approx S_h$. With the system in equilibrium, we can set $\alpha_h = 1$, which means that all of the energy impinging on the walls of the hood becomes absorbed one way or the other. Equations (12.45) becomes

$$W_2 = W_0 \frac{\bar{\tau}}{\bar{\alpha}} \tag{12.46}$$

where $\bar{\tau} \leq \bar{\alpha} \leq 1$. The limits are established so that when $\bar{\alpha}$ approaches unity, the defining expression for τ is satisfied. But when $\bar{\alpha}$ approaches $\bar{\tau}$, nearly the entire acoustic power of the source is radiated outside the hood.

Setting $Q_0 = Q_2 = Q$ in Equations (12.43) and (12.44) (i.e. the Q of the hood-source combination equals that of the source alone) and substituting into Equation (12.42), we obtain

$$\text{IL} = L_{w0} - L_{w2} = 10 \log \left(\frac{W_0}{W_2} \right). \tag{12.47}$$

Inserting Equation (12.46) into Equation (12.47) yields

$$\text{IL} = 10 \log \left(\frac{\bar{\alpha}}{\bar{\tau}} \right) \tag{12.48}$$

where $\bar{\tau} \leq \bar{\alpha} \leq 1$.

In actuality the average absorption coefficient $\bar{\alpha}$ has a lower nonzero limit as the result of air absorption inside the enclosure, viscous losses of waves near grazing incidence in the acoustical boundary layer on inside of the hood, and the change from adiabatic to isothermal compression in the immediate vicinity of the inner

walls. According to Crèmer (1961), the latter two effects may be taken into account in terms of sound frequency f by the expression

$$\alpha_f = 1.8 \times 10^{-4}\sqrt{f}.$$

The average excess air absorption coefficient $\bar{\alpha}_{\text{ex}}$ due to air absorption in a room or enclosure is

$$\alpha_{\text{ex}} = k\frac{4V}{S}$$

where k is an experimentally determined constant, V is the enclosed volume, and S the room or enclosure area. Combining the last two expressions, we get a minimum $\bar{\alpha}_{\text{min}}$ expressed by

$$\bar{\alpha}_{\text{min}} = \alpha_{\text{ex}} + \alpha_f = \frac{4kV}{S} + 1.8 \times 10^{-4}\sqrt{f} \tag{12.49}$$

which is also valid for reverberation rooms. Equation (12.49) helps to establish the upper limit of the reverberation time that can be achieved in an echo chamber.

The two limiting cases for Equation (12.48) are

$$\text{IL} = 0 \text{ dB} \quad \text{for } \bar{\alpha} = \bar{\tau}, \qquad \text{IL} = 10\log(1/\tau) = \text{TL dB} \qquad \text{for } \bar{\alpha} = 1.$$

The first case of IL = 0 obviously represents the worst case, and the second case where IL = TL represents the best case for the insertion loss of the hood. This connotes that $\bar{\alpha}$ should be near unity as possible and $\bar{\tau}$ be made much less than unity for the most effective noise attenuation by a hood.

The above analysis is premised on the presence of high frequencies with a diffuse field inside the enclosure. For a more complete treatment that includes the effects of low frequencies, the reader is referred to Ver (1973).

Example Problem 9

Determine the insertion loss of a hood with an average absorption coefficient of 0.30 and a transmission coefficient of 0.002.

Solution

Applying Equation (12.48)

$$\text{IL} = 10\log\left(\frac{\bar{\alpha}}{\bar{\tau}}\right) = 10\log\left(\frac{0.30}{0.002}\right) = 21.8 \approx 22 \text{ dB}$$

12.15 Small Enclosures

A small enclosure is one that fits closely around the noise source. If the noise source's geometry contains flat planes that are parallel to the walls of the enclosure, standing wave resonances can occur at frequencies that are integer multiples of

the half-wavelengths of the generated noise. This can render the enclosure useless from the viewpoint of noise attenuation unless the inside of the enclosure is lined with sound-absorption material at least one-quarter wavelength thick. If there is no sound absorption material inside the enclosure, situations can occur where the noise generated at resonant frequencies may actually be louder outside than would be the case without the enclosure!

12.16 Acoustic Barriers

The term *acoustic barrier* (or *noise barrier*) refers to an obstacle that interrupts the line of sight between a noise source and receiver but does not enclose either source or receiver. An acoustic barrier may be in the form of a fence, a wall, a berm (a mound of earth), dense foliage, or a building between the noise source and the receiver. Noise attenuation occurs from the fact that noise transmission through the barrier is negligible in comparison with refracted noise, particularly if the barrier is solid, without holes or openings, and it is of sufficient mass. Because the sound reaches the receiver by an indirect path over the top of the barrier, the sound level will be less than the case would be if the sound had traveled the (shorter) direct path. The refraction phenomenon is highly dependent upon the frequency content of the sound. The calculations for barrier attenuation are based in part on Fresnel's work in optics.

In Figure 12.16 consider a room prior to inserting a barrier in the position shown. The mean-square pressure at the sound receiver's location is given by

$$p_0^2 = W\rho_0 c\left(\frac{Q}{4\pi r^2} + \frac{4}{R}\right)$$

where p_0^2 denotes the mean-square pressure without the barrier. The sound pressure level is expressed by

$$L_{p0} = L_w + 10\log\left(\frac{Q}{4\pi r^2} + \frac{4}{R}\right)$$

where

$$\begin{aligned} L_{p0} &= \text{SPL without the barrier} \\ L_w &= \text{power level of the source} \\ Q &= \text{directivity of the source} \\ R &= \text{room constant without the barrier} \\ r &= \text{distance between the source and the receiver.} \end{aligned}$$

Now let us insert the barrier as shown in Figure 12.16. The mean-square pressure p_2^2, at the receiver with the barrier installed is given by

$$p_2^2 = p_{r2}^2 + p_{d2}^2 \tag{12.50}$$

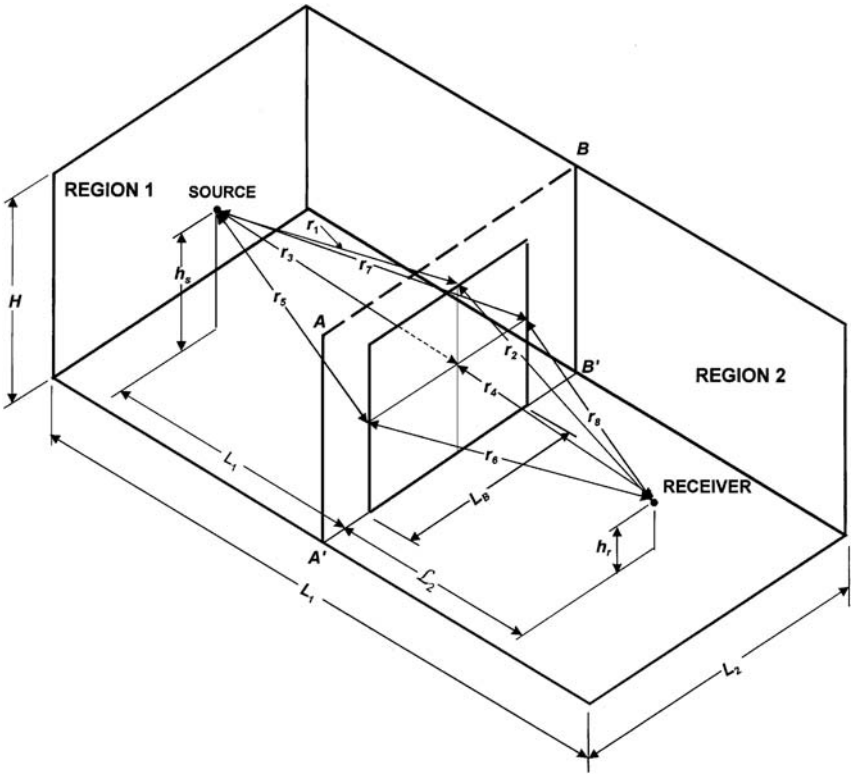


FIGURE 12.16. Schematic of a room with a barrier in position. Region 1 and region 2 are separated by a plane defined by $ABB'A'$.

where the first term on the right of Equation (12.50) is the mean-square pressure at the receiver due to the reverberant field and the second term represents the mean-square pressure due to the diffracted field around the edges of the barrier. Recognizing that $L_{p2} = 10\log(p_2^2/p_{\text{ref}}^2)$, we can express the sound pressure level L_{p2} at the receiver in terms of the reverberant and diffracted fields:

$$L_{p2} = 10\log\left(\frac{p_{r2}^2 + p_{b2}^2}{p_{\text{ref}}^2}\right). \tag{12.51}$$

The barrier insertion loss IL is given in terms of SPL by

$$\text{IL} = L_{p0} - L_{p2} = 10\log\left(\frac{p_0^2}{p_2^2}\right). \tag{12.52}$$

For the room of Figure 12.16, we shall assume an equilibrium condition and that the area of the barrier is considerably smaller than the planar cross section of the room, i.e.

$$LH \gg L_b h. \tag{12.53}$$

In this situation the reverberant field in the shadow zone of the barrier may be assumed to be the same with or without the barrier's presence. This reverberant field represents the minimum SPL in the shadow zone. The space average δ_r of the time-average reverberant energy density in the room without the barrier is given by

$$\delta_r = \frac{4W}{Rc} = \frac{p_r^2}{\rho_0 c}. \tag{12.54}$$

For the condition stated by Equation (12.53) $\delta_r = \delta_{r1} = \delta_{r2}$, where the numerical subscripts denote the regions on each side of the barrier in Figure 12.15. Equation (12.54) can now be rewritten as

$$p_{r2}^2 = \frac{4W\rho_0 c}{R}. \tag{12.55}$$

Here p_{r2}^2 denotes the mean-square pressure in the reverberant field of the shadow zone of the barrier. Our next step is to include the mean-square pressure p_{b2}^2 in area 2 at the location of the receiver due to the diffracted field from the edges of the barrier, and this is given by (see Moreland and Musa, 1972):

$$p_{b2}^2 = p_{d2}^2 \sum_{i=1}^{n} \frac{1}{3 + 10N_i} = p_{d2}^2 D \tag{12.56}$$

where p_{d2}^2 represents the mean-square pressure attributable to the direct field without the barrier, the Fresnel number N_i is defined by

$$N_i \equiv \frac{2d_i}{\lambda} \tag{12.57}$$

where

d_i = difference in direct path and diffracted path between the source and receive

λ = wavelength of the sound

and D is the diffraction constant defined by[2]

$$D \equiv \sum_{i=1}^{n} \frac{1}{3 + 10N_i}. \tag{12.58}$$

In the case of Figure 12.16, the following path differences exist:

$$\begin{aligned} d_1 &= (r_1 + r_2) - (r_3 + r_4) \\ d_2 &= (r_5 + r_6) - (r_3 + r_4) \\ d_3 &= (r_7 + r_8) - (r_3 + r_4) \end{aligned} \tag{12.59}$$

[2] In some literature the alternative definition for the diffraction coefficient is given by

$$D = \sum_{i=1}^{n} \frac{1}{3 + 20N_i}$$

In the case of rectangular barriers, three values of d_i, or $n = 3$, will usually suffice to describe the shadow zone of the barrier. Through Equation (11.20) we can write for the mean-square pressure p_{d2}^2 due to the direct field as follows:

$$p_{d2}^2 = \frac{QW\rho_0 c}{4\pi r^2}. \tag{12.60}$$

with r constituting the direct length from the source to the receiver. Combining Equations (12.56) and (12.60) yields

$$p_{b2}^2 = \frac{WQD\rho_0 c}{4\pi r^2} = \frac{WQ_B}{4\pi r^2}\rho_0 c. \tag{12.61}$$

where $Q_B \equiv Q_D$ is the effective directivity of the source toward the direction of the shadow zone. Inserting Equations (12.61) and (12.55) into Equation (12.50) results in

$$p_2^2 = W\rho_0 c\left(\frac{Q_B}{4\pi r^2} + \frac{4}{R}\right)$$

and the corresponding sound pressure is

$$L_{p2} = L_w + 10\log\left(\frac{Q_B}{4\pi r^2} + \frac{4}{R}\right) \tag{12.62}$$

where

$$Q_B = Q\sum_{i=1}^{n}\frac{\lambda}{3\lambda + 20d_i}. \tag{12.63}$$

In English units, where distances are expressed in feet instead of meters, Equation (12.62) becomes

$$L_{p2} = L_w + 10\log\left(\frac{Q_B}{4\pi r^2} + \frac{4}{R}\right) + 10. \tag{12.64}$$

For a rectangular barrier $n = 3$ in Equation (12.63), and the required path differences are given by Equation (12.59).

From Equations (12.52) and (12.62) we obtain the barrier insertion loss IL:

$$\mathrm{IL} = 10\log\left(\frac{\dfrac{Q}{4\pi r^2} + \dfrac{4}{R}}{\dfrac{Q_B}{4\pi r^2} + \dfrac{4}{R}}\right). \tag{12.65}$$

The above equation applies to either the metric system or English system. It is interesting to note that if the barrier is located in an extremely reverberant environment, such as an echo chamber, the acoustic field reaches the receiver by rebounding unabated from the surfaces of the room to the degree that the effectiveness of the barrier in blocking the direct field becomes insignificant. Because $4/R \gg Q/(4\pi r^2)$ and $4/R \gg Q_B/(4\pi r^2)$ for a reverberant room, Equation (12.65) will give a value of IL = 0 dB.

12.17 Barrier in a Free Field

The case of a barrier located outdoors or in an extremely acoustically absorbent room is of special interest. In a free field the room constant $R \approx \infty$, with the result that Equation (12.65) becomes

$$\text{IL} = 10\log\left(\frac{Q}{Q_B}\right) = 10\log\left(\frac{Q}{QD}\right) = 10\log\left(\frac{1}{D}\right). \tag{12.66}$$

where

$$D = \sum_{i=1}^{n} \frac{\lambda}{3\lambda + 20d_i}. \tag{12.67}$$

In treating the rectangular barrier in a free field or an acoustically high absorptivity chamber we expand Equation (12.67) for $n = 3$ and insert the result into the rightmost term in Equation (12.66) to yield

$$\text{IL} = -10\log\left[\lambda\left(\frac{1}{3\lambda + 20d_1} + \frac{1}{3\lambda + 20d_2} + \frac{1}{3\lambda + 20d_3}\right)\right]. \tag{12.68}$$

If the barrier is of semi-infinite length, i.e., $L_B \approx \infty$ in Figure 12.16, Equation (12.68) reduces to

$$\text{IL} = -10\log\left(\frac{\lambda}{3\lambda + 20d_1}\right). \tag{12.69}$$

Example Problem 10

Consider the room of Figure 12.16 to be an anechoic chamber with dimensions 20 ft long × 15 ft wide × 12 ft high. A sound source is located in the room 1.5 ft above the floor, 8 ft away from one of the shorter walls, in the center and 2 ft directly behind a 6-ft-wide × 5-ft-high rectangular barrier. The barrier itself has a very high transmission loss rating, and a receiver in the form of a SPL meter is located also in the center 2 ft above the floor and 6 ft away from the source. Determine the insertion loss of the barrier for the 1000-Hz octave band.

Solution

The path differences as given by Equation (12.59) assume the following values:

$$d_1 = (r_1 + r_2) - (r_3 + r_4) = (4.03 + 4.98) - 6 = 3.01 \text{ ft}$$
$$d_2 = (r_5 + r_6) - (r_3 + r_4) = (3.61 + 4.98) - 6 = 2.59 \text{ ft}$$
$$d_3 = (r_7 + r_8) - (r_3 + r_4) = (3.61 + 4.98) - 6 = 2.59 \text{ ft}.$$

The wavelength λ at 1000 Hz is found from

$$\lambda = \frac{c}{f} = 1128 \text{ ft/s} \div 1000 \text{ Hz} = 1.128 \text{ ft}$$

Equation (12.67) becomes

$$D = \lambda\left(\frac{1}{3\lambda + 20d_1} + \frac{1}{3\lambda + 20d_2} + \frac{1}{3\lambda + 20d_3}\right)$$
$$= 1.128\left[\frac{1}{3(1.128)+20(3.01)} + \frac{1}{3(1.128)+20(2.59)} + \frac{1}{3(1.128)+20(2.59)}\right]$$
$$= 0.059$$

The insertion loss is then calculated:

$$\text{IL} = 10\log(1/D) = 10\log(1/0.059) = 12\ \text{dB}$$

12.18 Approximations for Barrier Insertion Loss

Consider the barrier of Figure 12.17. The Fresnel number N, given by Equation (12.57) can be restated as

$$N = \frac{2(A + B - C)}{\lambda} = \frac{2(A + B - C)f}{c}$$

A number of researchers in the field developed and verified analytic models, with a view to apply the results to highway barriers. For a point source located behind an infinitely long solid wall or berm, the attenuation A' provided by a barrier are given by the following equations, where the arguments of tan and tanh are given in radians:

$$A' = 0 \qquad \text{for } N < -0.1916 - 0.0635b'$$

$$A' = 5(1 + 0.6b') + 20\log\left(\frac{\sqrt{-2\pi N}}{\tan\sqrt{-2\pi N}}\right) \quad \text{for} -0.1916 - 0.0635\text{b}' \le N \le 0$$

$$A' = 5(1 + 0.6b') + 20\log\left(\frac{\sqrt{2\pi N}}{\tanh\sqrt{2\pi N}}\right) \qquad \text{for } 0 < N < 5.03$$

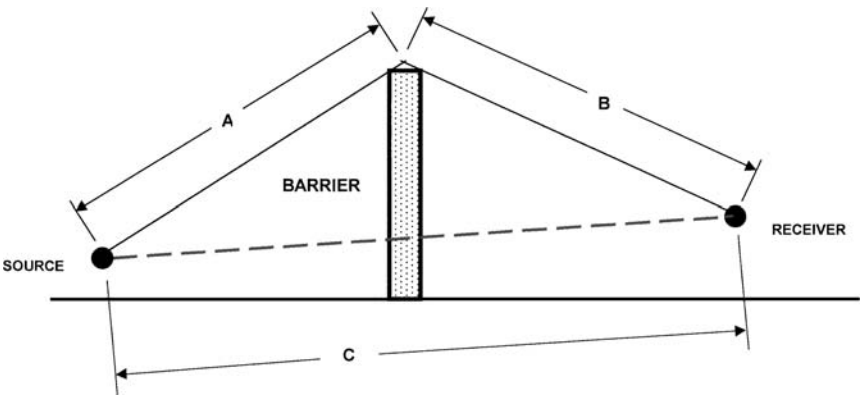


FIGURE 12.17. The geometry of a barrier used in the calculation of the Fresnel number.

where $b' = 0$ for a wall and $b' = 1$ for a berm. The correction factor b' allows for the experimentally determined fact that berms tend to produce 3 dB more attenuation than walls of the same height. The above set of equations apply only to barriers that are perpendicular to a line between the source and the receiver. A detailed discussion of attenuation through scattering and diffraction is given by Pierce (1981).

References

Barry, T. M., and Reagan. J. A. December 1978. *FHWA Highway Noise Prediction Model*. U.S. Department of Transportation Report FHWA-RD-77-108.

Beranek, Leo L. 1971. *Noise and Vibration Control*. New York: McGraw-Hill: 566–568.

Brüel & Kjær. 1980. *Measurements in Building Acoustics.* Nærum, Denmark: 19.

Crèmer, L. 1961. *Statistische Raumakustik*. Chapter 29. Stuttgart: Hitzel Verlag:.

Moreland, J. B., and Musa, R. S. 1972. *International Conference on Noise Control Engineering, October 4–6, 1972, Proceedings*: 95–104.

Pallett, D. Wehrli, Kilmer, R. R., and Quindry, T. 1978. *Design Guide for Reducing Transportation Noise in and around Buildings*. Washington, DC: U.S. National Bureau of Standards.

Pierce, A. D. 1981. *Acoustics, An Introduction to its Physical Principles and Applications*. New York: McGraw-Hill.

Reynolds, D. D. 1981. *Engineering Principles of Acoustics*. Boston: Allyn and Bacon.

Ver, I. L. 1973. Reduction of noise by acoustic enclosure. *Isolation of Mechanical Vibration, Impact, and Noise* (J. C. Snowdon, and E. E. Ungar, eds), AMD-Vol. 1, ASME Design Engineering Conference,: 192–220.

Ver, I. L. and Homer, C. I. 1971. Interaction of sound waves with solid structures. *Noise Control* (Beranek, L. L., ed.). New York: McGraw-Hill, 270–361.

Problems for Chapter 12

1. A room is subdivided in its middle by a concrete wall with a transmission loss of 52 dB. The room is 4 m high, and the division occurs across the entire width of 12.5 m. In the middle of the dividing wall there is a door 2.5 m high $\times$ 1.0 m wide. The door's rated transmission loss is 28 dB. There is a crack underneath the door that extends across the door's width and it is 2.5 cm high.
   (a) Determine the effective transmission loss of this structure.
   (b) If you could reduce the crack to 0.30 cm (by lengthening the door) what will be the change in the overall transmission loss?
2. A 5.8 ft^2 sample of a building material is being tested by being placed in a window between a receiving room and a source room. There is no appreciable sound transmission except that through the sample. Average sound levels in the 1-kHz octave band are 93 dB in the source room and 68 dB in the receiving room. The equivalent absorption area of the receiving rooms is 18 ft^2. Estimate the transmission loss (TL) and the transmission coefficient (TC) for the sample at 1 kHz.

3. A 0.75 m^2 sample of building material is placed in a window between a receiving room and a source room. Sound transmission occurs only through the sample. The sound level in the source room is 89 dB and that for the receiving room, with 3.0 m^2 equivalent absorption, is 69 dB. Find the transmission loss and the transmission coefficient for this material sample.
4. A wall consists of the following: a 22-dB (transmission loss at 500 Hz) wood door which takes up 25% of the exposed area, 2.2% airspace, and the remainder of the exposed wall area is a solid wall with 52 dB TL. Determine the TL of the composite wall at 500 Hz.
5. Repeat the above problem but with the airspace reduced to 0.2%.
6. Repeat Problem 4, using a 36-dB TL door. Also determine the benefit of reducing the airspace from 2.2% to 0.2%.
7. A room is subdivided in its middle by a concrete wall with a transmission loss of 62 dB. The room is 4.5 m high, and the division occurs across the entire width of 11.8 m. In the middle of the dividing wall there is a door 2.5 m high $\times$ 1.0 m wide. The door's rated transmission loss is 29 dB. There is a crack underneath the door that extends across the door's width and it is 2.6 cm high.
   (a) Determine the effective transmission loss of this structure.
   (b) If you could reduce the crack to 0.30 cm (by lengthening the door) what willbe the change in the overall transmission loss?
8. Develop an equation for transmission loss versus frequency for 4-mm glass in the mass-controlled region. Glass may be assumed to have a specific gravity of 2.6.
9. It is desired to increase the transmission loss of a panel in the mass-controlled region by 5 dB Find the necessary change in thickness.
10. Find the transmission loss and transmission coefficient for 8-mm glass at the critical frequency. Assume a loss factor of 0.06.
11. Redo Problem 9 for 14-mm glass with a loss factor of 0.1.
12. Consider a double wall panel with airspace h (given in mm). One panel has 70% the surface mass of the other, and the sum of the surface masses is m_t (in kg/m^2). Plot the resonant frequency versus the product of m_t and h.
13. Determine the resonant frequency of a double-paneled partition constructed of 4 lb/$ft^2$and 6 lb/ft^2 panels with 5.2-in. airspace.
14. A wall was measured for its transmission loss in one-third octaves beginning at 125 Hz. The values were: 25, 24, 30, 32, 39, 41, 41, 46. 47, 49, 47, 46, 48, 49, 45, and 46. Find the sound transmission class (STC).
15. Redo Problem 14 with the TL values in the four highest third-octaves being 39, 40, 38, and 39.
16. Find the SIR for the wall of Problem 14.
17. A 6 m $\times$ 2.5 m wall with a transmission loss of 35 dB separates two rooms. A noise source in one room yields a reverberant field with a sound pressure level of 120 dB near the wall. The other room has a room constant $R = 130$ m^2. Predict the sound level pressure near the wall in the latter room.

18. Predict the insertion loss of a hood that has an average absorption coefficient of 0.4 and a transmission coefficient of 0.003.
19. Determine the insertion loss at 1000 Hz of an outdoor noise barrier with a noise source 1.2 ft above the ground, 12 ft away from the wall that is 8 ft tall and 20 ft wide. The SPL meter is located 3 ft above ground and is 6 ft away from the wall. Both source and receiver are equally far from the ends of the wall.

13
Criteria and Regulations for Noise Control

13.1 Introduction

Loss of hearing constitutes only one of the effects of sustained exposure to excessive noise levels. Noise can interfere with speech and sleep, and cause annoyance and other nonauditory effects. *Annoyance* is quite subjective in nature, rendering it difficult to quantify. Loud steady noise can be quite unpleasant but impulse noise can be even more so. Moreover, impulse noise involves a greater risk of hearing damage. *Community response*, which can range anywhere from simple telephone calls to municipal authorities to massive public demonstrations of outrage, can be used as a measure of annoyance to citizens.

In this chapter we discuss these effects and describe the criteria and regulations that have been implemented over nearly three decades for the purpose of controlling environmental noise. In a broader sense, the term *environmental noise* pertains to noise in the workplace and in the community. This pertains to noise sources arising from operation and use of industrial machinery, construction activities, surface and air transportation. We must also consider *recreational noise* which arise from the use of snowmobiles, drag strip racers, highly amplified stereo systems in the home and in motor vehicles, and so on. A considerable number of regulations exist in a number of nations, for the purpose of protecting hearing from overexposure to loud noise, providing salutary working conditions, shielding communities from excessive environmental noise arising from the presence of manufacturing plants or construction activities and nearby surface and air transportation. The enactment and enforcement of these regulations motivates the control of noise levels in the workplace and in the community, not only for the purpose of promoting a proper auditory environment but also to avoid penalties that can be levied for uncorrected violations. The principal U.S. laws germane to the issue of noise control are:

- National Environmental Policy Act of 1969
- Noise Pollution and Abatement Act of 1970
- Occupational Safety and Health Act of 1970
- Noise Control Act of 1972.

As the corollary to these acts, the Environmental Protection Agency (EPA) (1973), the Department of Labor (DOL), and the Department of Transportation (DOT) have been designated as the principal federal agencies to issue and enforce noise control regulations.

Under the National Environmental Policy Act of 1969, any federally funded construction project requires the preparation and submission of an Environmental Impact Statement (EIS), which assesses the public impact of the noise that will be generated by the erection and operation of the completed project. The creation of the Office of Noise Abatement and Control (ONAC) as a branch of EPA was sanctioned by the Noise Pollution and Abatement Act of 1970. This office carries total responsibility for investigating the effect of noise on public health and welfare. The Occupational Safety and Health Act of 1970 set up the mechanics of enforcing safe working conditions, of which noise exposure criteria constitute a part. The Noise Control Act of 1972 probably contains the most important piece of federal legislation in regard to noisy environments. While EPA is given the primary responsibility for monitoring sound levels in the community, this legislation provides for a division of powers among the federal, state, and local governments.

13.2 Noise Control Act of 1972

With the passage of the Noise Control Act of 1972, the U.S. Congress established the national policy of promoting an environment that is free of excessive noise that would be deleterious to health, safety, and welfare. EPA was charged with the responsibility of coordinating federal efforts in noise control research, identification of noise sources and the promulgation of noise criteria and control technology, the establishment of noise emission standards for commercial products, and the development of low-noise emission products. This agency also bears the responsibility of evaluating the adequacy of Federal Aviation Administration (FAA) flight and operational noise controls and the adequacy of noise standards on new and existing aircraft (including recommendations on retrofitting and phaseout of existing aircraft). However, under the provisions of Section 611 of the Federal Aviation Act of 1958, FAA retained the right to prescribe and amend standards for the measurement of aircraft noise and sonic boom, but EPA can raise objections to FAA standards that, in the opinion of EPA, do not protect public health and welfare. Irreconcilable differences may be resolved by filing of a petition for review of action of the Administrator of EPA or FAA only by filing in the United States Court of Appeals for the District of Columbia Circuit. EPA may subpoena witnesses and serve relevant papers to obtain information necessary to carry out the act.

Section 17 of the Noise Control Act also assigns the task of developing noise emission regulations for surface carriers engaged in interstate commerce by railroads, in consultation with the Secretary of Transportation. States and local

governments are enjoined from adopting different standards except when rendered necessary by special local conditions and then only with the permission of EPA and the Secretary of Transportation. The provisions of Section 18 are nearly identical to those of Section 17 except they apply to motor carriers engaged in interstate commerce.

13.3 The Occupational Safety and Health Act of 1970

Under the directive of the Walsh–Healey Public Contract Act which was passed by Congress in 1969, the U.S. Department of Labor developed the first occupational noise exposure standard. In 1970, the Occupational Safety and Health Act was passed to apply the requirements of the standard to cover all workers engaged in interstate commerce. The Occupational Safety and Health Administration (OSHA) exists as an arm of the Department of Labor. The assigned task of this agency is to establish safety regulations (including those pertaining to noise levels) and enforce them.

According to OSHA regulations, the daily noise dose D in percent is given in terms of slow-response time-average A-weighted sound levels by

$$D = 100 \sum_{i=1}^{n} \frac{C_i}{T_i} \tag{13.1}$$

where C_i is the duration of exposure to a specific sound pressure level (SPL) and T_i is the allowable daily duration for exposure to noise at that value of SPL. Table 13.1 lists the A-weighted slow-response noise levels and their corresponding maximum daily exposure times T_i. Under OSHA regulations noise dosage exceeding 100% is not permitted. It will be noted in Table 13.1 that each 5 dB(A) increase in sound level cuts in half the allowable exposure time. For a time-average A-weighted average of 90 dB(A), the permitted exposure time is 8 h. Elevation of this noise

TABLE 13.1. OSHA Noise Exposure Limits (OSHA, 1978).

SPL, dB(A) (slow response)	Permissible Exposure, h/day
90	8
92	6
95	4
97	3
100	2
102	1.5
105	1
110	0.5
115	0.25 or less

level by 5 dB(A) cuts the allowable exposure time to 4 h. Exposure to noise levels above 115 dB(A) is not permitted. Moreover, for exposure to noise having a slow-response time-average A-weighted sound level of any value, the instantaneous peak sound pressure level may not exceed 140 dB. It should also be noted that the dosage levels are considered to be attributable to noise that actually reach a worker's station at the ear level. If a worker moves from place to place in the course of his or her occupational assignment, the dosage must be based on the noise exposure and the time spent at each station during the work period.

Example Problem 1

Monitoring of a factory noise environment at a worker's station yielded the following time samplings over the course of a normal 8 h: Compute the dosage and comment on the result.

Exposure Level, dB(A)	Exposure Period, h
85	1.5
90	3.0
92	2.5
95	0.5
97	0.3
100	0.2

Solution

Applying Equation (13.1) and using Table 13.1, we obtain

$$D = 100 \times \left(\frac{3}{8} + \frac{2.5}{6} + \frac{0.5}{4} + \frac{0.3}{3} + \frac{0.2}{2} \right) = 112\%$$

This dosage of 112% violates OSHA regulations. However, if this 8-h workday is cut down by 100/117 to 6.8 h or less (assuming the noise level distribution is fairly consistent throughout the day), the daily dosage will not exceed 100%. It can also be arranged to move the worker to a region, where the noise level averages 86 dB(A) or less (i.e., below the OSHA noise level range), for 1.16 h in order to round out 8 h daily on the job.

Annual audiology tests of all workers exposed to this environment are mandated if the noise level equals or exceeds 85 dB(A) over the course of a work day. Corrective measures must be taken when the dosage D exceeds unity. Aside from shortening time exposures to the offending noise, noise control measures described in the next chapter must be taken to decrease the sound level pressure of noise sources. Hearing protection devices such as earplugs, earmuffs, and special helmets are considered to be temporary stopgaps, and these do not necessarily remove the obligation of the work facility management to cut down on noise levels. OSHA

inspectors have the discretion to levy fines for violations that are not removed after an initial visit to the facility.

13.4 Perception of Noise

Loudness is not the only characteristic of noise that determines the degree of annoyance. There are other factors, acoustical as well as nonacoustical, that are important. In a classic series of laboratory studies conducted by Kryter and his associates (1959, 1963), human subjects rated sounds of equal duration according to their noisiness, annoyance, or unacceptability. Through the use of octave bands of noise Kryter with others established *equal noisiness contours*, which resemble those for equal loudness, but less acoustic energy is required at higher frequencies to produce equal noisiness and more is needed at low frequencies.

The unit of noisiness index is the *noy N* (this term apparently derives from the second syllable of the word an*noy*). Figure 13.1 displays the equal loudness index curves. In order to determine the logarithmic measure of the perceived noise level (PNL), the following standardized procedural steps need to be taken:

1. Obtain and tabulate the octave band (or 1/3 octave band) sound pressure levels produced by the noise.
2. Use Figure 13.1 to calculate the noisiness index for each octave (or 1/3 octave) band. Then determine the total noisiness index from either one of the following two expressions:

$$\boldsymbol{N}_t = \boldsymbol{N}_{\max} + 0.3\left(\sum_i \boldsymbol{N}_i - \boldsymbol{N}_{\max}\right) \tag{13.2a}$$

$$\boldsymbol{N}_t = \boldsymbol{N}_{\max} + 0.15\left(\sum_i \boldsymbol{N}_i - \boldsymbol{N}_{\max}\right) \tag{13.2b}$$

   Equation (13.2a) applies to one-octave bands and (13.2b) to one-third octave bands. For each octave (or 1/3 octave) band, $N_{\max}$ represents the maximum value of N_i found within that band.
3. The total perceived noisiness index N_t, which is summed over all frequency bands, is converted to the perceived noise level PNL (or L_{PN}) through the relationship

$$\boldsymbol{L}_{\mathrm{PN}} = 40 + (33.22)\log \boldsymbol{N}_t \tag{13.3}$$

There have been some questions raised regarding the validity of this procedure because the listeners in the laboratory trials do not always seem to distinguish the difference between loudness and noisiness and annoyance. But this procedure is now being used to evaluate single-event aircraft noise. The U.S. Federal Aviation Administration (FAA) adopted the PNL method to certify new aircraft.

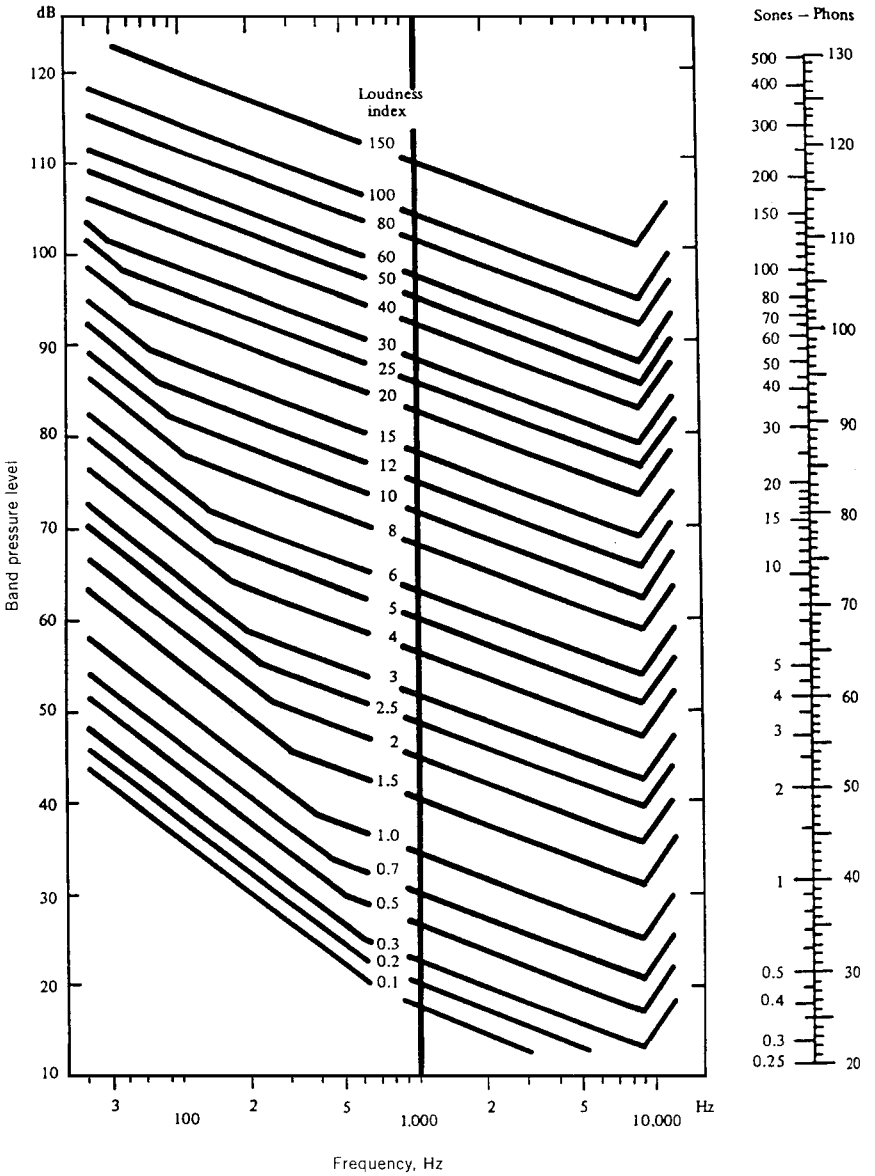


FIGURE 13.1. Equal loudness index curves.

13.5 Effective Perceived Noise Level

Noise of longer duration is obviously more annoying than noise of short duration. Moreover, if pure tones are present in the broadband noise system, the noise is judged to be even noisier than noise without such tones. In order to take into

account the factors of duration and the presence of pure tones, the effective noise perception level (EPNL) has been defined as

$$L_{\mathrm{EPN}} = L_{\mathrm{PN}} + C + D. \tag{13.4}$$

Here C is the correction factor for pure tones and D is the duration correction. The tone correction varies from 0 dB up to a maximum of 6.7 dB. The estimation of C entails a complex procedure (Edge and Cawthorn, 1977) that involves examination of the band spectra to detect any band whose level exceeds the level of the bands to either side. The duration correction D, expressed in decibels, which accounts for duration of the noise, may be either positive or negative but it is usually negative. It is calculated from

$$D = 10\log_{10}\left(\sum_{k=0}^{d/0.5} \mathrm{antilog}\frac{L_{\mathrm{PNT}}(k)}{10}\right) - 13 - L_{\mathrm{PNTmax}} \tag{13.5}$$

Here d represents the total length of the time elapsed when the noise begins to exceed the background level to the moment when it falls back to the level of imperceptibility. The number 0.5 represents the increment index, i.e., if the total duration d of the detectable sound is 5 s, then ten intervals are being considered in the summation of Equation (13.5). L_{PNT} is the tone-corrected value of L_{PN} (i.e. $L_{\mathrm{PNT}} = L_{\mathrm{PN}} + C$).

13.6 Indoor Noise Criteria

In order to render communication possible in both indoor and outdoor areas at work, it is necessary to minimize the speech interference arising from the background. The A-weighted sound level can be utilized to determine the acceptability of indoor noise, but it cannot give an indication as to which part of the frequency spectrum is responsible for interference. A number of noise evaluation curves are available for rating the acceptability of noise in indoor situations. The most frequently used families of curves are the noise criterion (NC) curves, noise rating (NR) curves, room criterion (RC) curves, and balanced noise criterion (NCB) curves. These curves were developed in order to provide criteria to either determine the acceptable noise levels in buildings or to specify the acceptable noise in buildings.

Noise Criterion Curves

The NC curves of Figure 13.2 were the result of an exhaustive series of interviews with people in offices, industrial spaces, and public areas. It was found that the principle concern is the interference of noise with speech communication and listening to radio, television, and music. In order to find the NC rating of a particular area, the octave-band sound pressure levels of the noise are measured and plotted on the family of the NC curves of Figure 13.2. The highest curve penetrated by any octave band and pressure level of the measured noise defines the NC value for the spectrum.

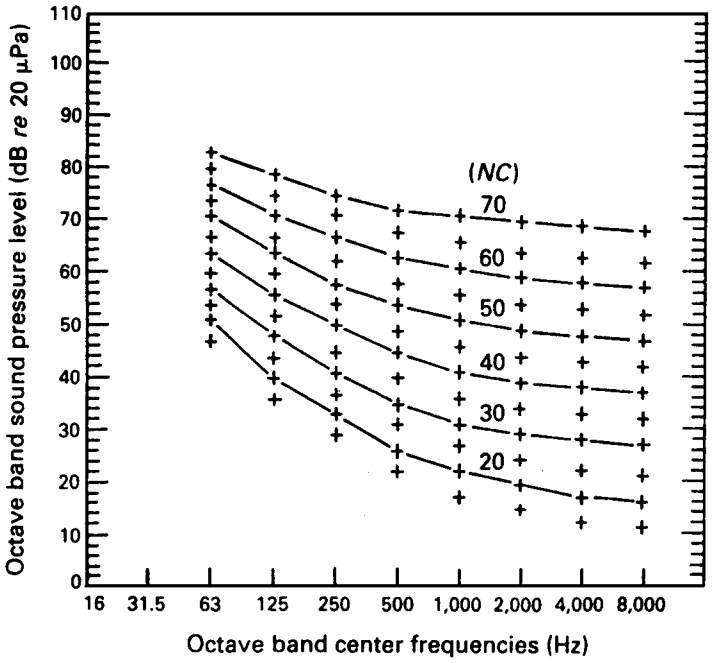


FIGURE 13.2. Noise criterion (NC) curves.

Example Problem 2

A noise generator in a room was found to yield 50 dB straight across the octave band from center frequencies 63–8000 Hz. What is the RC rating for that room?

Solution

Draw a horizontal line at the 50 dB level. The highest NC curve penetrated is 50. The NC value is therefore NC 50.

Noise Rating Curves

The noise rating (NR) curves of Figure 13.3 are very similar to the NC curves. Their original purpose was to determine whether noise heard from industrial plants is acceptable at nearby apartments and houses. As with the NC curves, the noise spectrum is also measured at the affected locations and plotted in Figure 13.3. The NR curves differ in that they include corrections for time of day, fraction of the time the noise is heard, and the type of neighborhood. In the range between 20 and 50 for NR or NC, there is little difference between results obtained from the two sets of calculated procedures.

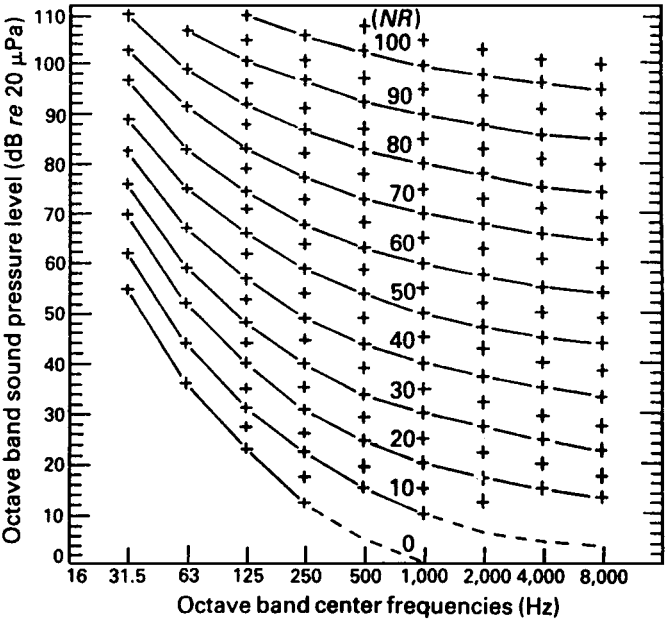


FIGURE 13.3. Noise rating (NR) curves.

Room Criterion Curves

The difficulty with NC curves is that they are not defined in the low frequency range—i.e, the 160 and 31.5 Hz octave bands—and complaints have been registered that they allow too much noise above the 2 kHz range. Accordingly, Blazier developed a set of room criterion curves on the basis of an extensive study conducted for ASHRAE of generally acceptable background spectra in 68 unoccupied offices. Most of the *A*-weighted sound pressure levels lie in the range of 40–50 dB. Blazier determined that the curve he derived from the measure data had a slope of approximately –5 dB/octave, and he developed a family of straight lines with this slope (cf. Figure 13.4). It was also established that intense low-frequency noise 75 dB or more in region A of Figure 13.5 is apt to cause mechanical vibrations and rattles in lightweight structures. Noise in region B has a low probability to generate such vibration. The RC value of a measured spectrum is defined as the arithmetic average of the sound levels at 500, 1000, and 2000 Hz. These curves are based on measurements made with air conditioning noise only, so they are mainly used in rating the noise of HVAC systems.

Balanced Criterion Curve

Beranek (1989a,b) modified the NC curves by adding the 16- and 31.5-Hz octave bands and modifying the slope of the curves so that it became –3.33 dB/octave between 500 and 8000 Hz. He also incorporated regions A and B of the RC curves

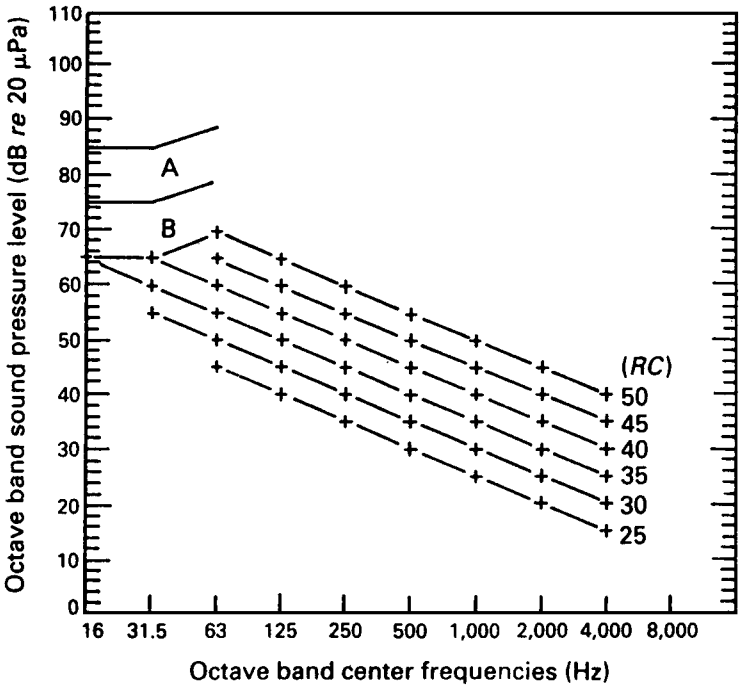


FIGURE 13.4. Room criterion (RC) curves.

of Figure 13.4. The rating number of the resulting *balanced noise criterion* (NCB) *curve*, shown in Figure 13.5, represents the arithmetic average of the octave-band levels with center frequencies of 500, 1000, 2000, and 4000 Hz. Figure 13.5 is useful for rating air-conditioning noise in buildings. Table 13.2 (Beranek and Ver, 1992) lists the recommended categories of NCB curves for different uses of building interior space.

Example Problem 3

In Figure 13.5, the background noise spectrum from air conditioning is plotted as a dashed line. Find the NCB rating and comment on the noisiness of the equipment.

Solution

We note that this plot is tangent to the NCB-40 curve and no octave-band level exceeds this curve. This is a spectrum that might be acceptable for a business office but is barely acceptable in a bedroom or a residential living room. According to Table 13.2, the NCB-40 rating is not at all acceptable in a concert hall, theater, or a house of worship.

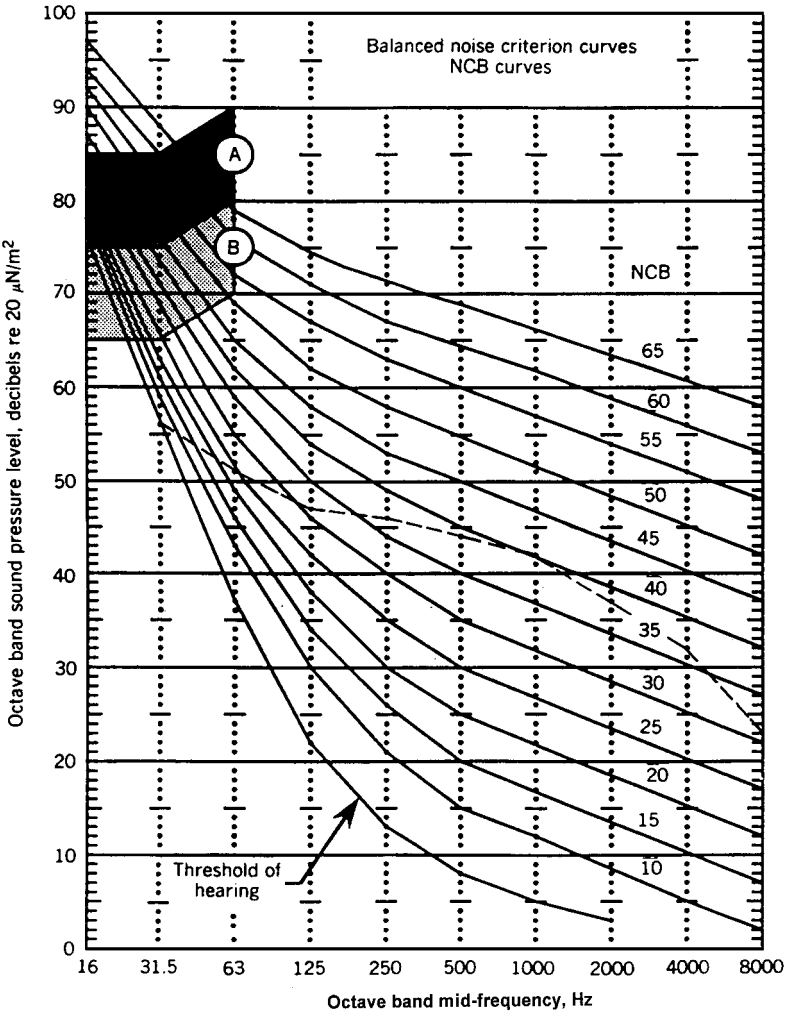


FIGURE 13.5. Balance noise criterion (NCB) curves.

13.7 Equivalent Sound Level, Day–Night Equivalent Sound Level, and Equivalent Day–Evening–Night Sound Level

From Chapter 3, we repeat here the definition of Equation (3.29) for the equivalent sound pressure level L_{eq}, which is the A-weighted sound pressure level averaged over a measurement period T.

$$L_{\text{eq}} = 10 \log \left(\frac{1}{T} \int_0^T 10^{L/10}\, dt \right) = 10 \log \left(\frac{1}{N} \sum_{n=1}^{N} 10^{L_n/10} \right) \qquad (13.6)$$

TABLE 13.2. Recommended NCB Curve Categories on the Basis of Interior Use and Suggested Noise Criteria Range for Steady Background Noise.

Type of Space (and Acoustical Requirements)	NCB Curve[a]	Approximate L_A, dB
Broadcast and recording studios (distant microphone pickup used)	10	18
Concert halls, opera houses, and recital halls (for listening to faint musical sounds)	10–15	18–23
Large auditoriums, large drama theaters, and large churches (for very good listening conditions)	Not to exceed 20	28
Broadcast, television, and recording studios (close microphone pickup used only)	Not to exceed 25	33
Small auditoriums, small theaters, small churches, music rehearsal rooms, large meeting and conference rooms (for very good listening), or executive offices and conference rooms for 50 people (no amplification)	Not to exceed 30	38
Bedrooms, sleeping quarters, hospitals, residences, apartments, hotels, motels, etc. (for sleeping, resting, relaxing)	25–40	38–48
Private or semiprivate offices, small conference rooms, classrooms, libraries, etc. (for good listening conditions)	30–40	38–48
Living rooms and drawing rooms in dwellings (for conversing or listening to radio and television)	30–40	38–48
Large offices, reception areas, retail shops and stores, cafeterias, restaurants, etc. (for moderately good listening conditions)	35–45	43–53
Lobbies, laboratory work spaces, drafting and engineering rooms, general secretarial areas (for fair listening conditions)	40–50	48–58
Light maintenance shops, industrial plant control rooms, office and computer equipment rooms, kitchens and laundries (for moderately fair listening conditions)	45–55	53–63
Shops, garages, etc. (for just acceptable speech and telephone communication). Levels above NC or NCB 60 are not recommended for any office or communication situation.	50–60	58–68
For work spaces where speech or telephone communication is not required, but where there must be *no risk* of hearing damage.	55–70	63–78

[a] See Figure 13.5.

This averaging time T can be chosen to be anywhere from a few seconds to hours. L_{eq} can be readily measured through the use of an integrating sound level meter. Because it takes into account both magnitude and duration, the equivalent sound level has proven to be a viable parameter for evaluating environmental noise from industry, railroads, and traffic. L_{eq} is found to correlate very well with the psychological effects of noise.

In order to account for different response of people to noise at night, the U.S. Environmental Agency developed the day–night equivalent level (DNL), as defined

by Equation (3.30), which we repeat here in modified form:

$$L_{\text{dn}} = 10 \log \frac{15\left(10^{L_d/10}\right) + 9\left(10^{(L_n+10)/10}\right)}{24} \tag{13.7}$$

where L_d represents the 15-h daytime A-weighted equivalent sound level (from 7:00 A.M. to 10:00 P.M.) and L_n is the 9-h nighttime equivalent A-weighted sound level (from 10 P.M. to 7:00 A.M.). The nighttime value carries a penalty of 10 dB because noise at night is deemed to be much more disturbing than noise generated during the day. The use of the day–night equivalent level as a parameter is increasing in the United States and some nations for evaluating community noise and some cases of airport noise. The U. S. Federal Interagency Committee on Urban Noise (FICON) adopted DNL as the descriptor of environmental noise that affects residences.

In 2002, the European Union Parliament and Council issued Directive 2002/49/EC in an effort to establish common assessment methods for rating environmental noise in terms of harmonized indicators for the determination of noise levels. The concrete figures of any limit values are to be determined by the Member States of the European Union. The common noise indicators adopted are *day–evening–night equivalent level* L_{den} to assess annoyance and the night equivalent level L_{night} to assess sleep disturbance. L_{den} and L_{night} are defined as follows:

$$L_{\text{den}} = 10 \log \frac{1}{24}\left(12 \times 10^{\frac{L_{\text{day}}}{10}} + 4 \times 10^{\frac{L_{\text{evening}}+5}{10}} + 8 \times 10^{\frac{L_{\text{night}}+10}{10}}\right)$$

where

$$\begin{aligned} L_{\text{day}} &= A\text{-weighted average sound level determined over 12-h day periods of the year} \\ L_{\text{evening}} &= A\text{-weighted average sound level determined over 4-h evening periods of the year} \\ L_{\text{night}} &= A\text{-weighted average sound level determined over 8-h night periods of the year.} \end{aligned}$$

We note that +5 dB(A) and +10 dB(A) penalties have been imposed to reflect the need for quieter periods of evenings and nights, respectively. While the day spans 12 h, the evening 4 h, and night, 8 h, the Member States may shorten the evening period by 1 or 2 h and lengthen the day and/or the night period, correspondingly, provided the measurements are consistent for all sources and the reporting States provide the European Commision with information on any systematic difference from the default option. The default values of (local) time are 0700–1900 h for day, 1900–2300 h for evening, and 2300–0700 h for night. Use of L_{den} will likely increase in the future for airport noise and other environmental assessments in European Union.

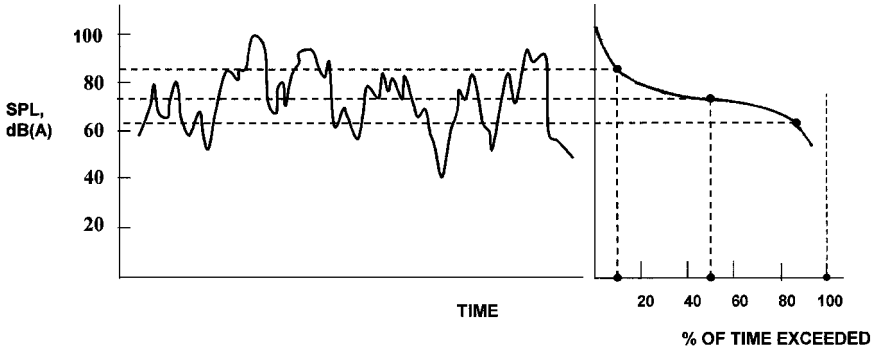


FIGURE 13.6. Example of a cumulative distribution and its percentile sound levels.

13.8 Percentile Sound Levels

Much of environmental noise tends to have a great deal of fluctuations in sound levels. There have been indications that unsteady noise, which occurs from noise sources such as passing aircraft or ground vehicles, is more disturbing than steady noise. In order to consider fluctuations in noise levels and the intermittent characteristics of some noises, percentile sound levels are used internationally as descriptors of traffic and community noise. The level L_n represents the sound level exceeded $n\%$ of the time. For example, L_{20} represents the sound level exceeded 20% of the time. In Figure 13.6 a cumulative distribution is given with examples of L_{10}, L_{50}, and L_{90} levels. In this example, $L_{10} = 85$ dB(A) denotes that 85 dB(A) is the level exceeded 10% of the time. $L_{50} = 75$ dB(A), and this is termed the *median* noise level, because half the time the fluctuating noise level is greater than this value and the other half of the time the noise is less.

In Japan the median noise level is used to describe road traffic noise. Levels such as L_1 or L_{10} are used to describe the more intense short-duration noises. In Australia and in the United Kingdom, L_{10} is applied (over an 18-h period, from 0600 to 2400 h) as target values for new roads and for insulation regulations for new roads. High percentage levels such as L_{90} or L_{99} are usually used to denote the minimum noise level or *residual* or *background noise* level.

13.9 Rating of Aircraft Noise

Aircraft noise is a cause of increasing concern in many nations. A considerable amount of effort has gone into developing the means for predicting and evaluating the annoyance caused by aircraft noise in communities. A 1995 survey (Gottlob, 1995) of rating measures revealed that 11 measures are in use in 16 countries surveyed. We discuss a few of these measures below.

Composite Noise Rating

The composite noise rating CNR traces its history as far back as the early 1950s. The CNR, originally used as a basic parameter, measures the *level rank* on the basis of a seta set of curves placed about 5 dB apart in the mid-frequency range, in nearly the same fashion as the NC and NR curves. The level rank was obtained by superposing the noise spectrum on the curves and determining the highest zone into which the spectrum protruded. The rank thus found was then modified to include algebraic corrections to reflect the spectrum characteristics, peak factor, repetitive nature, background noise level, time of day, adjustment to exposure, and even public relations factors. The realized value of CNR was associated with a range of community annoyance categories established by case histories, which can range from no annoyance through mild annoyance, varying degrees of complaints, threats of legal action, to downright vigorous community response.

The CNR had to be modified to meet the advent of jet aircraft (first military and then commercial) in the late 1950s. Instead of being assigned a level rank, the military aircraft noise was converted into an equivalent sound pressure level (SPL) in the 300–600 Hz range on the basis of a set of curves similar to the level-rank curves. The time-varying SPL was averaged and modified by corrections similar to those described in the preceding paragraph. When commercial jetcraft arrived on the scene, the CNR was modified to use the perceived noise level (PNL) instead of SPL. The final version of CNR is of the form:

$$\text{CNR} = \text{PNL}_{\max} + N + K$$

where $\text{PNL}_{\max}$ denotes the average maximum perceived noise level for individual aircraft flyover events (either landing or take-off) for a 24-h period, N is a correction factor for the number of aircraft flyovers, and K is an arbitrary constant. The CNR methodology has proven to be useful for predicting community response to aircraft noise, ranging on a scale from *no reaction* to *vigorous community response*. Each ground point can be represented by

$$\text{CNR}_{ij} = \text{PNL}_{ij} + 10\log(N_{\text{d},ij} + 16.7N_{\text{n},ij}) - 12 \tag{13.8}$$

where $N_{\text{d},ij}$ and $N_{\text{n},ij}$ represent the number of daytime and nighttime events for each aircraft class i and flight path j. A penalty of 10 dB is imposed on the nighttime flights, and the factor of 16.7 takes into account that there are fewer nighttime hours than daytime. The following expression gives the total CNR for the ground point:

$$\text{CNR} = 10\log\sum_i\sum_j \text{antilog}\frac{\text{CNR}_{ij}}{10} \tag{13.9}$$

But Equation (13.9), the final version of CNR, does not correct for background noise, prior history, public relations, or other factors such as the occurrence of pure tones. We discussed CNR here even though it is no longer used, because this parameter serves as the precursor of currently used noise measures and descriptors such as NEF and NNI, which are described below.

Noise Exposure Forecast

The *noise exposure forecast* (NEF) resembles CNR, but it uses the effective perceived noise level EPNL instead of PNL, thus automatically incorporating the annoying effects of pure tones and the duration of flight events. NEF can be expressed as

$$\text{NEF} = \text{EPNL} + N + K$$

where EPNL designates the average effective perceived noise level for individual aircraft flyovers over a 24-h period, N provides the correction for the number of flyover events, and K is an arbitrary constant. As was done for CNR in Equation (13.8). NEF can be calculated for NEF_{ij} for a certain ground point:

$$\text{NEF}_{ij} = \text{EPNL}_{ij} + 10\log(N_{\text{d},ij} + 16.7\,N_{\text{n},ij}) - 88 \tag{13.10}$$

Here EPNL_{ij} denotes the EPNL for aircraft class i and flight path j; and $N_{\text{d},ij}$ and $N_{\text{n},ij}$ are the number of daytime and nighttime events, respectively, for each aircraft of class i and flight path j. In order to distinguish NEF values from CNR values, the constant 88 was selected for Equation (3.10). Then

$$\text{NEF} = 10\log\sum_i\sum_j \text{antilog}\frac{\text{NEF}_{ij}}{10} \tag{13.11}$$

However, the use of NEF has been replaced by the day–night level L_{dn} in the United States. NEF is still being used in Canada and in a modified form in Australia.

Noise and Number Index

The noise and number index (NNI) is a rather subjective method for rating aircraft noise annoyance, developed and used in the United Kingdom. A survey conducted in 1961 of noise in the residential areas within a 10-mile radius of the London Heathrow Airport resulted in the creation of NNI, which is defined by

$$\text{NNI} = (\text{PNL})_N + 15\log_{10} N - 80 \tag{13.12}$$

where $(\text{PNL})_N$ comprises the average peak noise level of all aircraft operating during a day and is given by

$$(\text{PNL})_N = 10\log\left(\frac{1}{N}\sum_{n=1}^{N}\text{antilog}\frac{\text{PNL}}{10}\right) \tag{13.13}$$

PNL denotes the peak perceived noise level generated by a single aircraft during the day and N is the number of aircraft events over a 24-h period. Since no annoyance apparently occurs at levels less than 80 PNdB a constant of 80 was introduced into Equation (13.12) so that a zero value of NNI corresponds to no annoyance.

Another survey conducted in 1967 revealed that for the same noise exposure the reported annoyance was less than in 1961. This may be attributable to the fact that noise-sensitive people may have left the affected area, noise-insensitive people moving into the area, or the residents got used to their environment or

became apathetic to the noise, or the background noise arising from surface traffic, construction or operation of industrial facilities had increased sufficiently to mask more of the aircraft noise.

The use of NNI was superseded in the United Kingdom in 1988 in favor of measurements based on the A-weighted L_{eq} (cf., Equation (13.6)). This parameter is averaged over the period from 0700 to 2300 h (7 A.M. to 11 P.M.). The measure L_{dn} of Equation (13.7) is not applied because nighttime flights are severely limited in the United Kingdom. Both Switzerland and Ireland still make use of NNI.

Equivalent Sound Level

While some nations continue to use NEF or NNI as descriptors, it has been established that L_{eq} and L_{dn} are much simpler to measure and to evaluate since they seem to correlate well with subjective response. In addition to the United Kingdom which uses only L_{eq} over an 18-h period (since nighttime flights are restricted), both Germany and Luxembourg have adopted the L_{dn} method with (0600–2200 h) day and (2200–0600 h) night classifications. In the United States, as the result of the publication of EPA Report 550/9-74-004 in 1974 and similar documents, the use of CNR and NEF has been well superseded by the day–night equivalent level DNL for rating of potential impacts of noise and for planning purposes and land usage near military and civilian airports. In fact, a directive (Part 256 of Title 32 of the Code of Federal Regulations) was issued in 1977 by the U.S. Office of the Secretary of Defense that the day–night average sound level DNL must be used as the basis for evaluating the impact of noise by air installations and that neither CNR nor NEF may be utilized.

In 2002, the European Parliament and Council issued Directive 2002/30/EC on the establishment of rules and procedures with regard to the introduction of noise-related operating restrictions at airports. The objectives of this Directive are (a) to lay down rules for the Community to facilitate the introduction of operating restrictions in a consistent manner at airport administrative level so as to limit or reduce the number of people significantly affected by the harmful effects of noise, (b) to provide a framework which safeguards internal market requirements, (c) to promote development of airport capacity in harmony with the environment, (d) to facilitate the achievement of specific noise abatement objectives at the level of individual airports, and (e) to enable measures to be chosen with the aim of achieving maximum environmental benefit in the most cost-effective manner.

13.10 Evaluation of Traffic Noise

Highway traffic noise probably impacts more people than any other source of outdoor noise. Consequently, many national, state, and local governments set requirements to assess the existing or potential noise impact of highways. In the United States, federal agencies are required by law to provide environmental impact statements (EIS) for proposed new roads and for any reconstruction of existing

roads, where environmental impacts are likely to occur and federal funding is used to finance all or part of the project. The Federal Highway Administration (FHWA; not to be confused with FHA, the Federal Housing Administration) delegates the responsibility for preparation of environmental impact statements to the individual departments of transportation of the affected states. Moreover, the U.S. Department of Housing and Urban Development (HUD) also requires that when a developer plans residential property with the aid of construction funds or a guarantee of such funds from HUD, the developer must assess the potential impact of transportation noise upon that development.

Assessments of traffic noise are usually made in terms of the overall *A*-weighted sound levels. Octave-band and one-third octave-band levels are generally used only for the purpose of developing vehicle noise abatement measures such as acoustic barriers. Traffic noise tends to vary greatly over time, so methods are required to deal with this variation and its resultant impact on people.

Time Period

The time period used as a basis by FHWA is the 1-h period when the traffic is at its heaviest. This period is called the *worst noise hour*. The Department of Housing and Urban Development and certain states, California for example, require assessment over a 24-h period. Two types of descriptors are in general use: *statistical descriptors* and *time-averaging descriptors*.

Statistical Descriptors. The 10-percentile-exceeded level L_{10} encompasses the acoustic magnitude of individual traffic noise events, such as passages of heavy trucks, and also the number of such events. Originally the FHWA noise abatement criteria specified only in terms of a 1-h L_{10}. This descriptor is clearly inadequate for situations where (1) the hourly traffic rates are low, (2) vehicles are not evenly spaced along a road, and (3) the values of L_{10} could not be combined mathematically on the basis of calculations for separate events.

Time-Average Descriptors. Time-average descriptors are now widely used for assessing traffic. The most common such descriptor is the *1-h average sound level* (abbreviated 1HL), which is essentially the *A*-weighted equivalent continuous sound level L_{eq} taken over a 1-h period. This carries the advantage of (1) easy computation through the use of integrating meters, (2) assumptions regarding vehicle spacing are rendered unnecessary, and (3) the average levels for separate categories of sources may readily be combined. The principal disadvantage is that it can be extremely sensitive to isolated events having a high sound level, but which do not necessarily provoke a correspondingly high human response.

FHWA Assessment Procedures

The Federal Highway Administration (FHWA, 1976) requires that expected traffic impacts be determined and analyzed. Highway projects are classified by FHWA into two categories: Type I, which is a proposed federal or federal-aided

project to construct a new highway or to make major physical alterations to an existing road, and Type II, which is a project for noise abatement procedures that are added to an existing highway with no major alterations of the highway itself. All Type I projects are subject to FHWA regulations. Development and implementation of Type II projects is not mandatory, but a traffic noise analysis is mandated for eligibility of noise abatement measures for federal funding.

In the procedure for traffic noise analysis, the following steps are specified in FHWA regulations:

1. Identify activities and land uses that may be affected by the traffic noise. This can involve the use of aerial photographs, land-use maps, and highway plans, in addition to the results of field reconnaissance. Also determine regions of human activity and identify major sources of noise.
2. Predict future traffic noise levels through a method consistent with the FHWA Traffic Noise Prediction Model, employing noise emission levels that are either published in the FHWA regulations or determined by the agency using specified procedures.
3. Determine existing noise levels by actual measurements at sites that will be affected by the proposed highway. These ambient levels provide the basis for assessing impact and evaluating abatement feasibility.
4. Determine existing traffic noise impact. The predicted levels must be compared with the existing levels and with criteria based on land use. Table 13.3 lists the noise levels for different categories of land use. FHWA regulations describe two types of impacts on the land. The first type occurs when the predicted future levels "substantially exceed" the existing levels (but "substantially exceed" is not defined by FHWA, so different state transportation departments interpret the term differently, with minimum increases in A-weighted sound level ranging from 5 to 15 dB(A), with 10 dB(A) being a typical value). The second type of impact occurs when the predicted future levels "approach or exceed" the noise abatement criteria of Table 13.3. The criteria of Table 13.3 are listed according to the land-use activity and are in terms of 1-h average sound levels (L_{1h}) or 1-h 10-percentile-exceeded levels.
5. Evaluate abatement measures. In situations where severe impacts are identified, a state transportation agency need to examine means of reducing substantially or eliminating the impact. FHWA regulations specify that primary consideration should be given to exterior impacts where frequent human use occurs and where a decreased level would be of benefit. The state agencies must also (a) consider the opinions of affected residents, (b) identify in the environmental reports the abatement measures likely to be incorporated into the project and impacts where no solutions are apparent, and (c) include the abatement measure in the project plans and specifications to be approved. Some abatement measures include traffic management (e.g., prohibition of certain vehicle types, time restrictions for certain vehicle type, speed limits, traffic control devices, etc.) and acquisition of property to serve as buffer zones to preempt development that would be adversely affected by traffic noise.

TABLE 13.3. Yearly Day–Night Average Sound Levels for Land-Use Compatability (*Source*: Federal Aviation Administration, 1985)*

	Yearly day–night average sound level (L_{dn}), dB					
Land Use	Below 65	65–70	70–75	75–80	80–85	Over 85
Residential:						
Residential other than mobile homes and transient lodgings	Y	N(1)	N(1)	N	N	N
Mobile home parks	Y	N	N	N	N	N
Transient lodgings	Y	N(1)	N(1)	N(1)	N	N
Public use:						
Schools	Y	N(1)	N(1)	N	N	N
Hospital and nursing homes	Y	25	30	N	N	N
Churches, auditoriums, and concert halls	Y	25	30	N	N	N
Governmental services	Y	Y	25	30	N	N
Transportation	Y	Y	Y(2)	Y(3)	Y(4)	Y(4)
Parking	Y	Y	Y(2)	Y(3)	Y(4)	N
Commercial use:						
Offices, business, and professional	Y	Y	25	30	N	N
Wholesale and retail—building materials, hardware, and farm equipment	Y	Y	Y(2)	Y(3)	Y(4)	N
Retail trade—general	Y	Y	25	30	N	N
Utilities	Y	Y	Y(2)	Y(3)	Y(4)	N
Communication	Y	Y	25	30	N	N
Manufacturing and production:						
Manufacturing, general	Y	Y	Y(2)	Y(3)	Y(4)	N
Photographic and optical	Y	Y	25	30	N	N
Agriculture (except livestock) and forestry	Y	Y(6)	Y(7)	Y(8)	Y(8)	Y(8)
Livestock farming and breeding	Y	Y(6)	Y(7)	N	N	N
Mining and fishing, resource production and extraction	Y	Y	Y	Y	Y	Y
Recreational:						
Outdoor sports arenas and spectator sports	Y	Y(5)	Y(5)	N	N	N
Outdoor music shells, amphitheaters	Y	N	N	N	N	N
Nature exhibits and zoos	Y	Y	N	N	N	N
Amusements, parks, resorts, and camps	Y	Y	Y	N	N	N
Golf courses, riding stables, and water recreation	Y	Y	25	30	N	N

Numbers in parentheses refer to notes.

* The designations contained in this table do not constitute a federal determination that any use of land covered by a program is acceptable or unacceptable under federal, state, or local law. The responsibility for determining the acceptable and permissible land uses and the relationship between specific noise contours rests with the local authorities. FAA determinations under Part 150 are not intended to substitute federally determined land uses for those determined to be appropriate by local authorities in response to locally determined needs and values in achieving noise-compatible land uses. *Key:* Y(yes): Land use and related structures compatible without restrictions. N(no): Land use and related structures are not compatible and should be prohibited. NRL noise level reduction (outdoor to indoor) to be achieved through incorporation of noise attenuation into the design and construction of the structure. Land use and related structures are generally compatible; measures to achieve an NLR of 25, 30, or 35 dB must be incorporated into the design and construction of the structure.

Highway Construction Noise

No specific quantitative rules or guidelines for limiting highway construction noise are provided by the FHWA, but some state agencies use the criteria of Table 13.3 as guidelines for assessing the impact of construction. However, a computer model (HICNOM) is available from FHWA to be used to predict construction noise and asses abatement measures.

Vehicle Noise

Major contributors to vehicle noise include the engine exhaust and air intake, engine radiation, fans and auxiliary equipment, and tires. To a lesser degree other noise sources are the transmission, driving axles, and aerodynamic noise due to the passage of the vehicle through air. The relative importance of each component depends on the vehicle type and condition, vehicle load (passenger and cargo), speed, acceleration, and highway grade, and road surface condition. In order to aid in prediction of highway noise, FHWA conducted an exhaustive series of measurements of noise emission from, automobiles, trucks, buses, and motorcycles. The procedure for determining the noise emission levels entails the following steps:

1. A level open space free of large reflecting surfaces within 30 m (100 ft) of the vehicle path or microphone is identified. A 150° clear line-of-sight arc from the microphone position is required.
2. The surface of the ground should be free of snow and may be hard or soft. The roadway should be relatively level, smooth, dry concrete or asphalt. There should be no gravel.
3. The background level from all sources except the vehicle in question should be at least 10 dB(A) lower than the level of the vehicle in question.
4. The microphone is situated 15 m (50 ft) from the centerline of the lane of travel.
5. The microphone is mounted 1.5 m (5 ft) above the roadway surface and not less than 1 m (3.5 ft) above the surface upon which the microphone stands. It should be oriented according to the manufacturer's specifications.
6. The vehicle in question should be traveling at steady speed without acceleration or deceleration.

Vehicles are grouped by FHWA into three classes:

1. Automobiles (A): All vehicles with two axles and four wheels, including automobiles designed for transportation of nine passengers or fewer, and light trucks and SUVs. Generally, the gross vehicle weight (GVW) is less than 4500 kg (10,000 lb).
2. Medium Trucks (MT): All vehicles having two axles and six wheels, generally in the weight class 4500 kg (10,000 lb) GVW 12,000 kg (26,000 lb).

TABLE 13.4. Typical Octave Band Sound Pressure Levels of Automobiles and Heavy Trucks, Measured at 1.2 m Above the Ground at a Distance of 15.2 m.

	Octave-band center frequency, Hz						*A*-weighted sound level, dB(*A*)
	125	250	500	1000	2000	4000	
Automobile speed							
56 km/h (35 mph)	65	61	62	61	57	53	65
88 km/h (55 mph)	71	68	66	68	66	60	72
Heavy truck speed							
56 km/h (35 mph)	87	84.5	81.5	78	74.5	70.5	83.5
88 km/h (55 mph)	87.5	85	87.5	82.5	77	73.5	87.5

3. Heavy Trucks (HT): All vehicles having three or more axles, including three-axle buses and three-axle tractors with and without trailers. Generally GVW > 12,000 kg (26,000 lb).

Table 13.4 shows the octave-band sound pressure levels typical of automobiles and trucks at two different speeds.

Measurements are made by vehicle type for each selected speed ±5 km/h. For a given category of vehicles at a given speed, the *reference energy emission level* is given by

$$L_0 = L_{0\text{mean}} + 0.115 L_{0\text{SD}}^2$$

where $L_{0\text{mean}}$ is the arithmetic average emission level for the specific category and speed and $L_{0\text{SD}}$ is the standard deviation of that emission level.

Vehicle Noise Prediction According to the FHWA Model

On the basis of many measurements, FHWA developed a model for predicting highway noise (Barry and Reagan, 1978). We first compute the *reference mean level* $[(L_0)_\text{E}]_i$ for each vehicle type i. This level represents a speed-dependent value of the average (energywise) of the maximum passby levels measured at a reference distance of 15.2 m (50 ft) for a given vehicle type. The U.S. Federal reference mean emission levels are computed from:

$$\text{Automobile: } (L_0)_E = 38.1 \log(S) - 2.4 \text{ dB}(A) \tag{13.14a}$$

$$\text{Medium trucks: } (L_0)_E = 33.9 \log(S) + 16.4 \text{ dB}(A) \tag{13.14b}$$

$$\text{Heavy trucks: } (L_0)_E = 24.6 \log(S) + 38.5 \text{ dB}(A) \tag{13.14c}$$

where S is the average operating speed in km/h.

Adjustments for Traffic Conditions

A number of adjustments have to be made to the reference mean emission levels computed through the use of Equations (13.14). These adjustments are for (a) the

traffic density, (b) distances to areas of human activity, (c) finiteness of roadway segments, (d) presence of road gradients, (e) shielding by buildings, (f) shielding by rows of trees, and (g) barrier attenuation.

The traffic flow adjustment $(\Delta_{\text{traffic}})_i$ is rendered to account for the hourly flow of vehicle type i and to average the level over a 1-h time period as follows:

$$(\Delta_{\text{traffic}})_i = 10 \log (N_i d_0 / S_i) - 25 \quad \text{dB}(A) \tag{13.15}$$

where

N_i = hourly flow rate of vehicles of type i, vehicles/h
d_0 = the reference distance of 15.2 m
S_i = speed of the ith vehicle type, km/h

Distance adjustment takes into consideration the type of ground cover (e.g., sound will attenuate more readily over a soft surface than over a hard parking lot pavement). Field data gathered by FHWA have shown an average rate of attenuation to be approximately 4.5 dB(A) per doubling of distance over a grassy surface and an attenuation of 3.0 dB(A) per doubling of distance over a paved surface. These two rates are used by FHWA to establish a ground cover parameter called an *alpha factor* (α). For hard sites, the alpha factor has a value of zero; for soft sites, it assumes the value 0.5. The distance adjustment Δ_{distance} can be generally expressed as

$$\Delta_{\text{distance}} = 10(1 + \alpha) \log(d_0/d)$$

where d is the perpendicular distance from the receiver to the center of the travel lane. Table 13.5 provides guidance on when to use propagation rates of 3.0 or 4.5.

The methodology described above assumes an infinitely long road. In analytical practice, highways are subdivided into a series of straight segments of finite length.

TABLE 13.5. Guidelines for Choosing Sound Propagation Rates.

Situation	Propagation rate, dB(A)
1. All situations in which the source or the receiver are located 3 m above the ground or whenever the line of sight[a] averages more than 3 m above the ground	3 ($\alpha = 0$)
2. All situations involving propagation over the top of a barrier 3 m or more in height	3 ($\alpha = 0$)
3. Where the height of the line of sight is less than 3 m and	
a. there is a clear (unobstructed) view of the highway, the ground is hard, and there are no intervening structures, or	3 ($\alpha = 0$)
b. the view of the roadway is interrupted by isolated buildings, clumps of bushes, or scattered trees, or the intervening ground is soft or covered with vegetation	4.5 ($\alpha = 1/2$)

[a] The line of sight (L/S) is a direct line between the noise source and the observer.

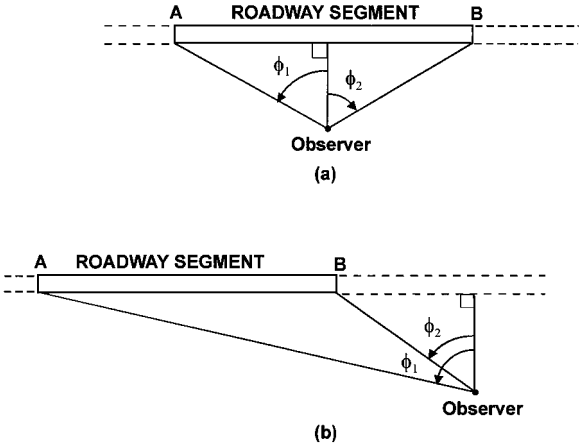


FIGURE 13.7. Subdivision of a road for noise analysis purposes with segment ends A and B defining the angles with respect to the normal from the observer to the road. In (a) an angle clockwise to the observer is positive and an angle counterclockwise is negative. In (b) the angles are measured from the normal line to the extension of the road segment.

The 1-h average sound levels are computed separately for each segment, and then they are combined in the end. Such subdivision of the roadway for analytical purposes should be executed (a) where the traffic volumes or speeds change (e.g., at an exit or entry ramp or a road fork), (b) where the ground cover changes significantly, (c) where a curved road is being analyzed as a series of straight segments, and (d) where the vertical gradient of the road changes. With reference to Figure 13.7, the *finite roadway segment adjustment* (which depends on the ground cover) is defined as

$$\Delta_{\text{segment}} = 10 \log \int_{\phi_1}^{\phi_2} \frac{1}{\pi} (\cos\ \phi)^a \, d\phi \tag{13.16}$$

where ϕ_1 and ϕ_2 are angles in radians at the receiver, as shown in Figure 13.6; ϕ_1 is the angle to the left end of the segment and the angle ϕ_2 to the right end of the segment. If an angle is measured counterclockwise from the normal line, it is assigned a negative value; if measured clockwise, a positive value. For a hard site Equation (3.16) reduces to

$$\Delta_{\text{segment}} = 10 \log \frac{\phi_2 - \phi_1}{\pi} \tag{13.17}$$

In the case of a soft site, Equation (13.16) needs to be evaluated by numerical integration. For the general case of a "soft" site and an infinitely long roadway (where $\phi_1 = -\pi/2$ radians and $\phi_2 = +\pi/2$ radians), the adjustment is -1.2 dB(A).

It is commonly observed that large trucks become noisier as they travel uphill in lower gears. The FHWA recommends that the 1-h average levels for heavy trucks be increased as a function of roadway gradient[1] in the following manner:

1. 0–2%: $\Delta_{\text{grade}} = 0\,\text{dB}(A)$
2. 3–4%: $\Delta_{\text{grade}} = +2\,\text{dB}(A)$
3. 5–6%: $\Delta_{\text{grade}} = +3\,\text{dB}(A)$
4. Over 7%: $\Delta_{\text{grade}} = +5\,\text{dB}(A)$.

If there are one or more rows of buildings present between an observation point and a road, an adjustment for the shielding, $\Delta_{\text{shielding}}$, by these buildings must be estimated. A rule of thumb can be used in the following manner: if 40–65% of the length of the first row of buildings is occupied by the building themselves, then subtract 3 dB(A) from the average sound level. If 65–90% of the length of the row is occupied by the buildings, subtract another 3 dB(A) from the average sound level; and if the percentage exceeds 90%, the buildings may be treated as noise barriers. Each successive row of buildings adds an additional 1.5 dB(A) to this adjustment, up to a maximum reduction of 10 dB(A). Any excess ground attenuation stops at the first row.

A second type of shielding, also referred to as $\Delta_{\text{shielding}}$, can occur from the presence of trees between the road and the receiver. The tress may be dense enough to virtually disallow a direct view of the road. For a 30-m (100 ft) belt width, FHWA suggests an adjustment of –5 dB(A). For an additional 30-m belt width there is an additional –5 dB(A) adjustment for a total of –10 dB(A). When this adjustment is used, both distance adjustment and a segment adjustment should be used on the basis of the propagation rate of 3 dB(A) per doubling of distance from the line source.

Barrier attenuation adjustment Δ_{barrier} can be computed using the path-difference procedure described in the last chapter. According to the FHWA model, this should be computed by using an incoherent line source model separately for each vehicle type. The attenuation is computed for a series of paths, defined by the angles with the perpendicular to the source-to-receiver line, and the results are combined via numerical integration. This calculation is usually achieved with the use of a computer program such as the FWHA STAMINA program.

The total 1-h average sound levels for each vehicle type can be obtained by summing up the various adjustments:

$$(L_{\text{eq}})_i = [(L_0)_E]_i + (\Delta_{\text{traffic}})_i + (\Delta_{\text{distance}})_i + (\Delta_{\text{grade}})_i + (\Delta_{\text{shielding}})_i + (\Delta_{\text{barrier}})_i \tag{13.18}$$

The 1-h average levels for automobiles, medium trucks, and heavy trucks may now be combined to yield the total 1-h average level.

---

[1] When a road changes its elevation n meters (or feet) for every 100 meters (or feet) horizontal travel, the road is said to have an n percent gradient.

Computer Programs

The FHWA noise prediction methodology has been incorporated into the following computer programs:

1. SNAP 1.1, intended for relatively simple geometries, can handle up to twelve roadways and one simple barrier. This program provides a detailed output and can provide separate results according to vehicle type.
2. STAMINA 2.0 and its derivative programs are used by most states, along with a noise barrier program called OPTIMA. Complex highway sites with many roadways, barriers, and receivers can be analyzed with this program. Additional vehicle types can be specified in addition to the three classes of automobiles, medium trucks, and heavy trucks. The output of the program can include the 1-h average sound level and 10-percentile-exceeded level for each receiver for the initial barrier segment elevations, the average sound levels for each individual road segment, and matrices of sound energy contributions from each barrier segment at each elevation for each receiver.
3. OPTIMA is used to design noise barriers on the basis of input information such as the type of material being used for each barrier. This program can be used to optimize the design on a cost effective basis.

13.11 Evaluation of Community Noise

Transportation noise, both surface and air, constitutes the dominant source of noise exposure in residential neighborhoods. Other sources can include noise emanating from industrial and commercial enterprises, rowdiness of individuals carousing in the streets, passing "boom cars," operation of lawnmowers or snowblowers, chain saws or other gear, and local construction. Concern about noise pollution began to intensify when commercial jet began to appear in the sky in the late 1950s. There was little in the way of standardization of community response survey methods, questionnaire items, and even noise measurements and analytical techniques throughout the 1960s and 1970s. But research was carried on during these years on annoyance, speech, and sleep interfering properties of noise. Researchers have come up with a number of ways to assess community response. In addition to frequency-weighting schemes, some of the properties of noise thought to have relevance include tonality, impulsiveness, rise time, onset time, periodicity, time of day, and temporal variability. Dozens of physical measure of sound have been considered as predictors of annoyance caused by noise exposure (Pearsons and Bennett, 1974; Schultz, 1982). Even today there is no single purely physical metric that can function as a definite predictor of annoyance with noise exposure, and it may not even be possible to develop such a predictor.

Social Surveys

Perhaps the least ambiguous procedure to evaluate the prevalence of noise-induced annoyance in a community is through the means of a social survey.

Survey techniques, however, can range widely in the degree of their respective sophistication.

Empirical Dosage Response Relationship. Schultz (1978) documented quantitative dosage-response relationships through meta-analysis, which constituted a major step toward the setting up of a standard method for predication of transportation noise effects. Schultz executed a "best-fit" third order polynomial to a data set relating the day–night average sound level (DNL) to the degree of annoyance in communities. A simple quadratic-fit equation provides a purely empirical basis for predicting the prevalence of annoyance in communities:

$$\%\ \text{highly annoyed} = 0.036\boldsymbol{L}_{\text{dn}}^2 - 3.27\boldsymbol{L}_{\text{dn}} + 79.14 \tag{13.19}$$

However, Equation (13.19) produces meaningless predictions when evaluated outside the range 45 dB $< L_{\text{dn}} <$ 85 dB. The U.S. Federal Interagency Committee on Noise prefers a different logistic fit to a subset of data reported by Fidell et al. (1991), which resulted from disregarding the results of certain studies in which relatively low levels of noise were associated with a high degree of reported annoyance:

$$\%\ \text{highly annoyed} = \frac{100}{1 + e^{11.13 - 0.141 L_{\text{dn}}}} \tag{13.20}$$

Equation (13.20) predicts somewhat lower levels of annoyance at lower noise exposure levels than the quadratic fit of Equation (13.19). For example, at $\boldsymbol{L}_{\text{dn}} =$ 65 dB. Equation (13.20) yields a prevalence rate of 12.3% in contrast to 18.8% obtained through the use of (13.19).

Equal Energy Hypothesis. DNL has been adopted by many federal agencies as a convenient descriptor of long-term environmental noise descriptor, which soon enough became a predictor of annoyance. The DNL index is a time-weighted average (in effect, the average acoustic energy per second with arbitrary nighttime weighting), which is sensitive to the duration and magnitude of individual noise events and directly proportional on an energy (10 log N) basis to number of events.

Reliance on such an integrated energy metric is based on the "equal energy" hypothesis, which states the notion that the number, level, and duration of noise events are fully interchangeable determinants of annoyance as long as their product (energy summation) remains equal. This quantification of noise exposure in terms of DNL for the purpose of predicting annoyance carries the implication that a person would be annoyed to the same degree by small numbers of very high level noise events as by large numbers of lower level noise and/or longer duration noise events. The equal energy hypothesis has provided an adequate account for data on the prevalence of annoyance to sporadic (e.g., urban) noise in the range $55 < \boldsymbol{L}_{\text{dn}} < 75$ dB, but the validity of the hypothesis falls off in extreme cases. For example, no community is likely to tolerate even infrequent operation of a noise source powerful enough to damage hearing or a very occasional shock wave from passing supersonic jet plane from a nearby military base.

13.12 Guidelines and Regulations in the United States, Canada, Europe, and Japan

The most well-established of noise effect guidelines in the United States are those promulgated by the Federal Interagency Committee on Noise (FICON). FICON is composed of a number of federal agencies with interest in environmental noise [e.g., FAA and Department of Defense (DoD)]. The guidelines and recommendations of FICON for "land-use compatibility" are couched in ranges of DNL. FICON considers noise exposure levels lower than $\boldsymbol{L}_{\mathrm{dn}} = 65$ dB to "be compatible with most residential land uses." However, FICON does confess its realization that this limitation may be somewhat too high in highly rural areas where it would be more appropriate to characterize the effects of noise pollution not in acoustic terms but rather in terms of annoyance.

On the international scene, a major degree of consensus has evolved over the years as to what constitutes unacceptable levels of noise exposure. In the mid 1980s the Organization for Economic Cooperation and Development (OECD)[2] suggested a standard guideline value for average outdoor noise levels of 55 dB(A), which pertains to normal daytime in order to prevent significant interference with normal activities of local communities. According to OECD:

1. At 55–60 dB(A) noise creates annoyance.
2. At 60–65 dB(A) annoyance increases considerably.
3. Above 65 dB(A) constrained behavior patterns occur, symptomatic of serious damage by noise.

The World Health Organization (WHO) listed additional guidelines in 1996 for noise exposure in dwellings, schools, hospitals, concert halls, and so on. The exposure levels are given in terms of the time average A-weight sound level L_{eq}. For example, a private bedroom should sustain a sound level no higher than $L_{\mathrm{eq}} = 30$ dB(A) at night in order to promote undisturbed sleep; and the background noise level in a classroom should be no greater than $L_{\mathrm{eq}} = 35$ dB(A) to facilitate the teaching processes.

Workplace Noise Exposure

In the United States the passage of the Noise Control Act of 1970 gave rise to OSHA regulations, which are described in Section 13.3. In Europe, the Council of European Communities issued a *Council Directive* 86/188/EEC on May 12, 1986, which sets the guidelines on the protection of workers from the risks related to exposure to noise at work. This directive does not prejudice the right of members of EU to introduce or apply even stricter provisions that reduce the permissible levels of noise. For an 8-h day, under the provisions of 86/188/EEC, a worker should

[2] While its name indicates its mission, the Organization for Economic Cooperation and Development also concerns itself with environmental matters. Its membership includes industrial nations from North America, Europe, and Asia.

not be exposed to more than A-weighted $L_{eq} = 85$ dB per 8-h day. This value is determined from measurements made at the position occupied by a worker's ears during work, preferably in that person's absence, using a technique that minimizes the effect on the sound field. If the microphone has to be located close to the worker's body, appropriate adjustments should be made to determine an equivalent undisturbed field pressure. The daily personal noise exposure does not take into account the effect of any personal ear protectors used. Should the allowable daily exposure be exceeded, the affected workers are to be notified, and procedures must be enacted to lower the exposure, which may include periodic checks of the work environment, provision of ear protectors, and audiology tests. If the daily personal noise exposure exceeds 90 dB(A) or the maximum value of the unweighted instantaneous sound pressure is greater than 200 Pa, the employer must exert every effort to cut down this exposure.

Vehicle Noise Regulations

Exterior noise tests for road vehicles have been used in Japan since 1971. The current version of the tests is quite thorough and consists of three parts: (a) a fixed passby test at 7 m, 60% rated engine speed or 65 km/h, (b) an acceleration passby test at 7.5 m, and (c) a stationary test behind the exhaust outlet at 20 m.

In these tests the microphone is placed 1.2 m above ground level. The noise limits for new highway vehicles are given in Table 13.6 which went into effect in 1998. Another 2 dB reduction in these noise limits has been in effect since 2002.

Table 13.7 lists the mandatory exterior limits for new highway vehicles sold in the European Union. The EU vehicle noise test procedure is prescribed in *Council Directives* 92/97 EEC and 81/334/EEC, presented in the *Official Journal of the European Communities*. Amendments were added for automatic transmission-equipped and for high-powered vehicles. The test is based on ISO 362, which is substantially equivalent to SAEJ140 (cf., *Handbook of the Society of Automotive Engineers*). A 1-dB(A) tolerance is permitted in this test. The test and its derivatives

TABLE 13.6. Japanese Noise Limits for New Highway Vehicles.

Vehicle Description	GVWR (metric tons)	Power (kW)	Limit [dB (A)]
Passenger car	—	—	76
Light truck	<3.5	—	79
Medium truck, bus	<3.5	<150	81
Heavy bus	>3.5	>150	81
Heavy truck	>3.5	>150	81
Moped <50 cc			70
Motorcycle			
>50 cc < 125 cc			70
>125 cc < 250 cc			73
>250 cc			76

TABLE 13.7. European Union Highway Vehicle Noise Limits.

Vehicle Description	Unloaded Weight (metric tons)	Power (kW)	Noise Limit, dB(A)
Passenger car	—	—	74
Mini bus <9 seats	<2	—	76
	>2 < 3.5	—	77
Bus >9 seats	>3.5	<150	78
Light truck/van	<2	—	76
	<3.5	—	77
Medium truck/van	>3.5	<75	78
	>3.5	>75 < 150	78
Heavy trucks	>12	>150	80
Motorcycles			
≤80 cc			75
>80 ≤ 175 cc			77
>175 cc			80

are widely used throughout the world, including Japan (which does not have 1-dB(A) tolerance). The test site configuration is used for all classes of vehicles, including motorcycles, with the measurement distance set at 7.5 m.

Boom cars, sometimes referred to as "boomers," are vehicles that have audio equipment installed specifically to generate excruciatingly loud sound levels. Because the market is so lucrative, particularly among automobile owners under 30 years of age, manufacturers aggressively engage in the promotion of ultra-high power amplifiers and rather hefty loudspeaker systems. Boomer systems are usually operated at levels that generate a high degree of annoyance in residential neighborhoods, which constitute a problem of increasing proportions (Raichel, 2000). Loudness competitions are held, and levels as high as 175 dB (two and a half times louder than the Boeing 747 jumbo jet) have been reported. Much of the equipment advertising emphasizes loudness; and it is apparent that fidelity of sound reproduction rarely figures in the setup of a boom system.

The problem of boom cars is not confined to the United States. It is also becoming a problem in Latin America (Raichel and Miyara, 2001). The phenomenon of boom cars compares to tobacco addition—a danger to vehicle occupants and a nuisance to "secondary listeners." Countermeasures against such public nuisances include: (a) enactment and diligent enforcement of municipal laws specifying limitations on sound levels, (b) education in school systems on the dangers of excessive noise levels, and (c) public criticism of manufacturers' promotional efforts that stress extremely high-level sound outputs. In Chicago boomer cars that can be heard from 22 m (75 ft) are subject to seizure and their owners may be fined $615. Buffalo, Cleveland, and Pittsburgh police are cracking down on boom cars. In Papillion, Nebraska, owners of car stereos that can be heard from 15 m (50 ft) away can be sentenced to 3 months in jail. The police in Lorain, Ohio have been regularly enforcing the law approved in 2002 that prohibits car steros from being audible at 15 m or more from the car. A first offense brings a $300 fine. A second offense

costs $400 and the offender's audio equipment gets confiscated as contraband. That equipment is then destroyed by the police (Lorain Morning Journal, 2003).

Motorcycles and other recreational vehicles also can be troublesome to communities around the world. In an informal survey of 33 motorcycles, audiological researchers at the University of Florida found that nearly half of them produced sounds about 100 dB when throttled up—equivalent in intensity to a loud rock concert or a chainsaw. At the Institute for Sound and Vibration Research (ISVR) at the University of Southampton, UK, it was found the noise levels under motorcycle helmets can be quite high. Above 65 km/h (40 mph) the wind noise generated by airflow over a motorcycle and rider exceeds the noise level from the motorcycle itself. In Montana, Section 61-9-418 (2003) of the state code requires that all motorcycles or quadricycles operating on the streets and highways must be equipped at all times with noise suppression devices, including an exhaust muffler, in good working order at all times. In addition, all motorcycles and quadricycles must meet the following noise level limitations, on the basis of measurement of 50 ft from the closest point of the motorcycle or vehicle:

(a) any cycle manufactured prior to 1970: 92 dB(A)
(b) any cycle manufactured after 1969 but prior to 1973: 88 dB(A)
(c) any cycle manufactured after 1972 but prior to 1975: 86 dB(A)
(d) any cycle manufactured after 1974 but prior to 1978: 80 dB(A)
(e) any cycle manufactured after 1977 but prior to 1988: 75 dB(A)
(f) any cycle manufactured after 1987: 70 dB(A).

In New Hampshire, a House Bill 326 was introduced in the legislature in 2005 to ban modification of a motorcycle to amplify or increase the noise level beyond that emitted by the original equipment installed by the manufacturer. No straight pipe exhaust system is allowed; and no motorcycle shall be operated, which has a noise level of more than 100 dB(A) when measured 10 ft (3 m) or further from the exhaust pipe or muffler. Noise measurement is to constitute a part of the mandatory regular vehicular inspection.

Railroad Noise Regulations

In the United States, regulations for railroad noise are published in Section 40, Part 201, and Section 49, Part 210, of the *Code of Federal Regulations*. Test measurements are conducted within a cleared level area 30 m from the track center line. Locomotives are tested either in motion or stationary status using remote load cells. Noise limits have been specified for locomotives manufactured after 1979:

Stationary, idle throttle setting: 70 dB(A)
Stationary. All other throttle settings: 87 dB(A)
In motion: 90 dB(A).

For rail cars, the limits are:

Moving at speeds 83 km/h (50 mph) or less: 88 dB(A)
Moving at speeds greater than 83 km/h: 93 dB(A).

The test is to be conducted during dry weather without the presence of dust or powdery snow.

The EU has no exterior noise standards for railroad equipment because of intermingling with equipment from outside the EU and differences in rail gauges. Railroads have used, on their own, a number of noise standards such as the International Union of Railways ORE E82/RP4 and the ISO 3095: 1975.

Highway Construction Noise

An interesting development in EU is the use of sound power as a noise measure, rather than sound pressure at a single point in space, for earth-moving and other construction equipment. Sound power measures the total noise emanating in all directions from a source. The fundamentals of sound power measurement are given in Chapter 9, Sections 9.17 *et seq.*; and details are given in the EU directives 95/27/EU, 89/514/EEC, 86/662/EEC, 84/533-536b/EEC, and 79/113/EEC. The distribution of measurement positions specified by 79/113/EE is given in a hypothetical hemisphere enclosing a stationary piece of earth-moving equipment. Sound pressure is measured at the prescribed measurement points, which is converted into sound intensity, applying the far-field approximation and integrated to yield sound power. The sound power measurement would be more accurate if sound intensity measurements were used instead of sound pressure. In its current form the test uses only half of the measurement points specified in 79/113EEC. A similar sound power test was developed for lawn mowers in 84/538/EEC.amended by 88/181/EEC.

New noise limits came into effect at the end of 1996 for earth-moving machinery of net installed power less than 500 kW. Machines with power exceeding 500 kW operate in quarries and mines and so are considered to have a negligible effect on community noise. For the period between 1996 and 2001, the permissible sound power levels L_{WA}, in A-weighted decibels relative to 1 pW, are given by

1. Tracked vehicles (except excavators):

$$L_{WA} = 87 + 11 \log P \quad \text{for } L_{WA} \geq 107 \tag{13.21}$$

2. Wheeled bulldozers, loaders, excavators loaders:

$$L_{WA} = 85 + 11 \log P \quad \text{for } L_{WA} \geq 104 \tag{13.22}$$

3. Excavators:

$$L_{WA} = 83 + 11 \log P \quad \text{for } L_{WA} \geq 96 \tag{13.23}$$

where P is the net installed power of the construction vehicle in kilowatts. Below the lower limit given above, the machine automatically passes the test. After 2001, the numerical A-weighted decibel values in Equations (13.21–13.23), including the lower limits, are reduced by 3 dB(A). The coefficient 11 remains unchanged.

Over the past few years, the European Commission of the EU has been laboring on a new directive regarding machinery used outdoors. Directive 200/14/EC relating to the noise emission in the environment by equipment used outdoors was adopted by the European Parliament and the Council and first published in May 2000. This directive became effective January 3, 2003. Directive 2000/14/EC covers 57 types of equipment for outdoor use, ranging from construction machinery to lawnmowers. It is intended to supersede similar directives that exist separately in the EU's member nations. Directive 200/14/EC demands declarations from manufacturers on the "guaranteed" sound power levels of their products before they can be marketed in the EU. Such products must bear a CE mark and an indication of their guaranteed sound output level and be accompanied by an ED declaration of conformity before they can be placed on the market. The guaranteed sound power level is that defined by ISO 3744-1995 with the addition of uncertainties (owing to production variations and measurement procedures), which the manufacturer confirms will not be exceeded. Therefore, in order to obtain estimates of these uncertainties, a sampling procedure on the production line may be appropriate.

The directive places sound power limits on 22 of the 57 types of machinery for use outdoors. Of these 22, one half were already subject to noise limits laid down in seven earlier directives (including Directive 98/0029 described in the first edition of this text). The remaining half of this group are subject to noise limits for the first time. All manufacturers of products subject to noise limits must follow conformity assessment procedures under the supervision of organizations appointed by the EU member states. Failure to comply with these regulations may result in nonconforming products being barred on the EU market.

Over the past quarter century or so, a greater awareness of the impact of construction noise on the part of government agencies led to a series of codes and regulations for the control and mitigation of noise from construction sites. These acts generally cover (a) the erection, construction, alteration, repair, or maintenance of buildings, structures, or roads; (b) the breaking up, opening, or boring under any road or adjacent land in connection with the construction, inspection, maintenance, or removal of public or private works; (c) piling, demolition, or dredging works; or (d) any other work entailing engineering construction.

A major cornerstone in the development of effective construction noise control programs may very well be the Construction Noise Control Specification 721.560 developed by the Massachusetts Turnpike Authority for the Central Artery Tunnel (CA/T) Project, also known as "the Big Dig." In the Boston area, at the close of the 20th century, this 12-year plus undertaking ranks to date as the largest infrastructure construction project in the United States. With this project's completion the notorious Boston traffic bottleneck on U.S. Interstate Highway I-93 to/from Logan International Airport should be alleviated, thus freeing up the City of Boston and the entire New England corridor to normal traffic flow. Apart from doubling Boston's highway capacity, this project should lead to modernization of Boston's underground utilities and enable the city to achieve positive growth in the 21st century.

Construction over the 11-km-long project occurred 24 h a day in various locations within the city. Construction equipment operated in close proximity to thousands of residential and commercial properties, in some cases as close as 3 m away. Hundreds of construction machinery were operating at any one time throughout the project. The range of equipment types used is wide, including cranes, slurry trenching machines, hydromills, hoe rams, pile drivers, jackhammers, dump trucks, bulldozers, concrete and chain saws, and gas and pneumatically powered hand tools. Upon the completion of this project, more than 10 million cubic meters of excavated materials will have been removed and nearly 3 million cubic meters of concrete poured.

In order to contain the acoustical impact of the project, while supporting and maintaining the progress of construction, the CA/T Project people developed a policy summary to establish an overall noise control program that includes (a) a commitment statement to minimize noise impact on neighboring residences while sustaining construction progress; (b) a summary of the Project Noise Control Specification criteria and components; (c) an expression of willingness to develop area-specific noise mitigation strategies tailored to particular community needs and sensitivities; (d) an approach and criteria for judging the worthiness of mitigation measures; and (e) a commitment to provide qualified noise experts to oversee contractor compliance in the field. Construction Noise Control Specification, 721.560 adopted and enforced by the CA/T Project, ranks as the most comprehensive and stringent noise code of any pubic works project in the United States.

In addition to the CA/T project, other outstanding sets of codes or regulations pertaining to construction noise (Raichel and Dallal, 1999a,b) include those by the Hong Kong and Singapore governments as well as those developed for the I-15 project in Utah. Germany's "Blue Angel" seal program certifies noise-generating equipment that operate below specified noise levels.

References

Anonymous. 1998. *Low-Noise Cpnstruction has a Future (Leises Bauen hat Zukunft): Noise Reduction with the "Blue Angel"* (Lämminderung mit dem "Blauen Engel"), 1st ed. Kalsruhe, Germany, Fachinformationszentrum, Gesellschaft für Wissenschaftlishtechnische,

Barry, T. M., and Reagan, J. R. 1978. *FHWA Highway Traffic Noise Prediction Model.* Report FHWA-RD-77-108. Washington, DC: Federal Highway Administration.

Beranek, Leo L. 1989a. Balanced Noise Criterion (NCB) Curves. *Journal of the Acoustical Society of America* 86: 650–664.

Beranek, Leo L. 1989b. Applications of NCB noise criterion curves. *Noise Control Engineering Journal* 33: 45–56.

Blazier, W. E. 1981. Revised noise criteria for application in the acoustical design and rating of HVAC Systems. *Noise Control Engineering*16: 64–73.

Bowlby, W. (ed.). 1981. *Sound Procedures for Measuring Highway Noise: Final Report*. Report FHWA-DP-45-1. Washington, DC: Federal Highway Administration.

Bowlby, W., and Cohn, L. F. 1982. *Highway Construction Noise: Environmental Assessment and Abatement*, *Vol. 4*: *User's Manual for FWHA Highway Construction Noise*

Computer Program, HICNOM. Vanderbilt University Report VTR 81-2. Washington, DC: Federal Highway Administration.

Bowlby, W. 1980. *SNAP 1.1—A Revised Program and User's Manual for the FHWA Level 1 Highway Noise Traffic Noise Prediction Computer Program*. Report FHWA-DP-45-4. Arlington, VA: Federal Highway Administration.

Bowlby, W., Higgins J., and Reagan, J. (ed.). 1982. *Noise Barrier Cost Reduction Procedure, STAMINA 2.0/OPTIMA: User's Manual*. Report FHWA-DR-58-1. Washington, DC: Federal Highway Administration (based on Menge, C. W. 1981. *User's Manual: Barrier Cost Reduction Procedure, STAMINA 2.0 and OPTIMA*, Report 4686. Cambridge, MA: Bolt, Beranek and Newman.

Council of the European Communities. May 12, 1986. *Council Directive 86/188/EE on the Protection of Workers from the Risks Related to Exposure to Noise at Work*.

Department of Labor Occupational Noise Exposure Standard. May 29, 1971. *Code of Federal Regulations*, Title 29, Chapter XVII, Part 1910, Subpart G, 36 FR 10466.

Department of Labor Occupational Noise Exposure Standard. March 8, 1983. *Amended Code of Federal Regulations*, Title 29, Chapter XVII. Part 1910, 48FR9776-9785.

Edge, Jr., P. M., and Cawthorn, J. M. February 1977. *Selected Methods for Quantification of Community Exposure to Aircraft Noise*, NASA TN D-7977.

Environmental Protection Agency (EPA). July 1973. *Public Health and Welfare Criteria for Noise*. Washington, DC: EPA.

European Union Council. June 17, 1999. *Proposal for a Directive of the European Paliament and of the Council on the Approximation of the Laws of Member States Relating to Noise Emission by Equipment Used Outdoors*. Interinstitutional File 09/0029. Brussels: European Council of EU.

European Union Council. 2000. Directive 2000/14/EC of the European Parliament and of the Council of 8 May 2000 on the approximation of the laws of the Member States relating to the noise emission in the environment by equipment for use outdoors (*Official Jounral* L 162 of 03.07.2000).

European Union Council. 2002. Directive 2002/30/EC of the European Parliament and of the Council of 26 March 2002 on the establishment of rules and procedures with regard to the introduction of noise-related operating restrictions at Community airports (*Official Journal* L 085, 29/03/2002: 0040-0046).

European Union Council. 2002. Directive 2002/49/EC of the European Parliament and of the Council of 6 25 June 2002 relating to the assessment and management o environmental noise. (*Official Journal* L 189, 18/07/2002).

Federal Aviation Administration (FAA). January 18, 1865 (revised). Part 150—Airport noise compatibility planning. *Federal Aviation Regulations*. Washington, DC: Federal Aviation Administration.

Federal Highway Administration. July 8, 1982. Procedures for abatement of highway traffic noise and construction noise. 23 C.F.R., Part 722. *Federal Register* vol. 47: 29653–29657.

Federal Interagency Committee on Noise (FICON). 1992. Final Report: *Airport Noise Assessment Methodologies and Metrics*. Washington DC: FICON.

Federal Register, 40. May 28, 1975. p. 23105.

Fidell, S., Barber, D., and Schultz, T. 1991. Updating a dosage-effect relationship for the prevalence of annoyance due to general transportation noise. *Journal of the Acoustical Society of America* 89, 1: 221–233.

Gottlob, D. 1995. Regulations for community noise. *Noise/News International* 3(4): 223–236.

Society of Automotive Engineers (SAE). 1991. *Handbook of the Society of Automotive Engineers*. Warrendale, PA: SAE.

Harris, Cyril M. (ed.). 1991. *Handbook of Acoustical Measurements and Noise Control,* 3rd ed. New York: McGraw-Hill; Chapters 23–26, 46–50.

Hong Kong Environmental Protection Department. 1999. *A Concise Guide to the Noise Control Ordinance*. Hong Kong: EPD. Internet address: http://www.info.gov.hk.epdinkh/noise/book2/book20.htlm.

Jacques, Jean. June 2004. Noise and standardization, focusing on machinery and workplace domains. Presented at the Joint Baltic-Nordic Acoustics Meeting, Mariehamn, Åland.

Kryter, Karl D. 1970. *The Effects of Noise on Man*. New York: Academic Press.

Lorain Morning Journal. October 9, 2003. Lorain more pleasant as law hammers down on boom car stereos. Lorain, Ohio.

Lower, M. C., Hurst, D. W., Claughton and A. R., and Thomas, A. 1994. Sources and levels of noise under motorcyclists helmets. *Proceedings of the Institute of Acoustics* 16(2): 319–326.

Lower, M. C., Hurst, D. W., and Thomas, A. 1996. Noise levels and noise reduction under motorcycle helpmets. *Proceedings of Internoise '96*, Book 2: 979–982.

Massachusetts Turnpike Authority. Central Artery (I-93)/Tunnel (I-90) Project. Revised July 28, 1998. Construction Noise Specification 721.560. Boston, MA: Commonwealth of Massacusetts.

Montana State Code. 2003. Code 61-9-418. Motorcycle noise suppression devices.

New Hampshire Transportation Committee. March 9, 2005. House Bill 326.

Occupational Noise Exposure Standard. June 28, 1983. U.S. Department of Labor. Occupational Safety and Health Administration (OSHA), *Code of Federal Regulations*, Title 29, Part 1910, Section 1910.95 [29 CFR 1910.95], *Federal Register* 48: 29687–29698.

Olson. N. 1970. *Statistical Study of Traffic Noise*, Report APS-476. Ottawa: National Research Council of Canada, Division of Physics.

Pearsons, K. S., and Bennett, R. 1974. *Handbook of Noise Ratings*. Washington, DC: NASA.

Raichel, D. R., and Dallal, M. October 1999a. Prediction and attenuation of noise resulting from construction activities in major cities. Presented at the 138th Meeting of the Acoustical Society of America November 4, 1999. *Journal of the Acoustical Society of America* 106, 4, Pt. 2: 2261.

Raichel, D. R., and Dallal, M. 1999b. Final Report: *Construction Noise*. Contract No. 99F88151, Subagreement No. 001 with the New York City Department of Design and Construction, Vols. 1 and 2. New York: The Cooper Union Research Foundation.

Raichel, Daniel R. 2000. Boom cars: noise pollution at its worst. Presented December 7, 2000 at the combined NoiseCon 2000/140th Meeting of the Acoustical Society of American, Newport Beach, CA.

Raichel, Daniel R., and Miyara, Federico. October 22, 2001. "Autos baffle: La problemàtica de los automóviles con refuero sonoro" presented at the Fourth International Multidisciplinary Conference on Acoustic Violence (Cuartes Jornados Interncionales Multiodiscoplinarias Sobre Violencia Acústica), Centro Cultural "Bernardino Rivadavia", Rosario, Argentina.

Schultz, T. J. 1978. Synthesis of Social Surveys on Noise Annoyance. *Journal of the Acoustical Society of Am*erica 64, 2: 277–405.

Schultz, T. J. 1982. *Community Noise Rating*. New York: Elsevier.

Singapore Government. 1990. *The Environmental Public Health Act (Chapter 95) S466/90 (Control of Noise from Construction Sites) Regulations*.

Problems for Chapter 13

1. An outdoor survey of a neighborhood indicated averaged sound pressure levels at 40 dB(A) from 7 A.M. to 9 A.M.; 55 dB(A) from 9 A.M. to 3 P.M.; 59.5 dB(A) from 3 P.M. to 10 P.M.; and 39.8 dB(A) from 10 P.M. to 7 A.M. What is the day–night sound level? Is the value obtained acceptable for a residential area?
2. A factory environment was found to have an average of 88 dB for 2.5 h, 90 dB for 1.6 h, 92 dB for 1.4 h, and 95 for 2.5 h. Is the 8-h exposure acceptable by OSHA standards? If not, what would be the maximum total daily working time, given the same histographic distribution?
3. A lathe in a machine shop was found to yield the following octave-band analytical results:

Center Frequency (Hz)	Octave Band Loudness (dB)
63	60
125	70
250	68
500	75
1000	72
2000	75
4000	74
8000	58

   What is the RC rating for this machine shop?
4. Redo problem 2 but find L_{eq} on the basis of the information given. If the 8-h day is excessive exposure by the European Union standards, how much must that workday be cut down in order to meet the daily exposure guidelines?
5. Examine the U.S. OSHA noise exposure regulations for workers and compare them with the counterpart European regulations. Which set of regulations is better for the worker? State the reasons for your choice.
6. Predict the mean emission levels for (a) an automobile going 135 km/h, (b) a medium truck moving at 120 km/h, and (c) a heavy truck traveling at 100 km/h.
7. Find the traffic flow adjustment for 3600 cars per hour moving at an average of 100 km/h for an observation point 15.2 m from the road.
8. A number of noise sources are located at different distances from a measurement point on a property line. The data consisting of the measured sound levels of each individual sources are given below:

Number of Sources	1	2	3	1	2	4
L at 25 ft	80	75	82	90	84	87
Distance, ft	58	78	74	68	120	92

   (a) Determine the noise contribution of each source or group of sources.
   (b) Find the combined noise level at the measurement point on the property line.

9. Use the data of Problem 8 to compute L_{eq24} and L_{dn} if these sources run continuously at the same noise output levels from noon to 3 P.M.
10. Find the permissible sound power levels of the following highway construction vehicles under the European Union rules:
    (a) a nonexcavating tracking vehicle rated at 120 kW
    (b) a 200-kW bulldozer
    (c) a 100-kW excavator
    What will be these corresponding values after the year 2001?
11. Outline a noise code suitable for a suburban residential neighborhood. It should include the appropriate noise level limits, specification of how and where measurements are to be taken.
12. Develop a noise code for a rural jurisdiction.
13. What are the problems that are applicable to urban areas? What are the special needs and goals with respect to noise levels in large cities?

14
Machinery Noise Control

14.1 Introduction

Workplace noise at high levels is detrimental to the welfare of workers. Not only high sound levels can affect hearing and hinder oral communication, but also they detract the employee from performing at peak capacity. In spite of possible economic drawbacks such as the cost of increased maintenance, the employer does have the moral obligation to provide a safe environment for both office and plant workers. Often when noise-attenuation measures are taken, some payback may accrue from the use of quieter machinery. For instance, when a diesel engine undergoes excessive vibration it becomes subject to severe stresses that can cause it to fail. Retuning of the engine so that it operates more smoothly lessens the stresses on its crankshaft or accessory parts and cuts down on its fuel consumption as well as the noise output. In many highly industrialized nations, such as the United States, Germany, France, the Scandinavian nations, and Japan, there are regulations that limit the noise exposure levels. Some of these regulations were discussed in Chapter 13.

This chapter deals with industrial noise sources, predictions of their respective acoustic output, and means of attenuating noise in the workplace. Specific types of noise sources are considered, followed by descriptions of general methodologies of noise control, which may or may not be machine specific.

14.2 Noise Sources in the Workplace

A considerable number of industrial machines and processes generate high levels noise that can cause physical and psychological stresses as well as considerable hearing loss. High-noise output machinery include blowers, air nozzles, riveters, pneumatic chisels and hammers, diesel generators, chipping hammers, rock crushers, die casting machines, drop hammers, metal presses, power saws, grinders, ball mills, stamping machines, and so on. In addition, building accouterments, such as furnaces, air conditioning and ventilation (HVAC) systems and plumbing, as well as office equipment can roil the office environment and add to the cacophony of an industrial plant.

In setting up noise control measures, the first step is to identify the noise sources and to measure the sound power output. Ideally, it would be desirable to take a source into a well-defined environment such as a reverberation chamber and measure its sound power output, but many stationary sources cannot be moved. But the sound power output for immovable sources can still be estimated from sound pressure measurements made on a hemispherical or rectangular enveloping surface (cf. Chapter 9). The sound pressure level will be increased by the presence of background noise and room reverberation, so a correction factor for either/both background noise and reverberation must be applied in such cases. For reasons which we shall see in the sections following, a spectrum analysis of noise from specific machines will often prove useful in tracing malfunctioning machinery parts so that they can be realigned or replaced.

Many noise problems in the workplace can be avoided by heeding the old adage that an ounce of prevention is worth a pound of cure. Prior to any purchase of machinery, sound power output of each unit should be obtained beforehand, either directly from the vendor or by conducting actual measurements on an existing installation. Wise planning of the plant layout includes not only promoting production efficacy and personnel safety; it should always involve prediction of noise output of all equipment in normal operation. Adjustments can then be made at this stage in the choice of quieter equipment and/or incorporating noise-reducing devices so that the planned facility will operate within the sound exposure limits mandated for workers.

14.3 Estimation of Noise Source Sound Power

A nondirectional point source in a free field will radiate sound uniformly and radially in all directions. Such a source L_W represents the true octave-band sound power level, with units of decibels based on the reference power 1 pW (10^{-12} W). Most pieces of machinery, however, are not point sources, nor do they radiate sound power uniformly. In planning of facilities, it is generally necessary to estimate the expected sound power for individual machines that will affect the environment. For certain machines, a sound power conversion factor F_n can be used to determine the output on the basis of the total power rating of the machine,

$$P = F_n \times P_m \tag{14.1}$$

where

$$P = \text{sound power of the machine, W}$$
$$P_m = \text{machine rated power, W}$$

The relationship of Equation (14.1) applies to both mechanical and electrical machinery. Estimated conversion factors for a number of common machinery are listed in Table 14.1. It should be noted that the ranges are quite large for each type of machine.

TABLE 14.1. Power Conversion Factors for Some Common Noise Sources.

Noise Source	Conversion Factor (F_n)		
	Low	Mid-Range	High
Compressors, air (1–100 hp)	3×10^{-7}	5.3×10^{-7}	1×10^{-6}
Gear trains	1.5×10^{-8}	5×10^{-7}	1.4×10^{-6}
Engines, diesels	2×10^{-7}	5×10^{-7}	2.5×10^{-6}
Motors, electric (1200 rpm)	1×10^{-8}	1×10^{-7}	3×10^{-7}
Pumps, >1600 rpm	3.5×10^{-6}	1.4×10^{-5}	5×10^{-5}
Pumps, <1600 rpm	1.1×10^{-6}	4.4×10^{-6}	1.6×10^{-5}
Turbines, gas	2×10^{-6}	5×10^{-6}	5×10^{-5}

Example Problem 1

Estimate the sound power of a "quiet" 100-hp electric motor operating at 1200 rpm.

Solution

From Table 14.1, $F_n = 1 \times 10^{-8}$. Applying Equation (14.1) and because 1 hp = 746 W,

$$P = F_n \times P_m = (1 \times 10^{-8})(100 \text{ hp} \times 746 \text{ W/hp}) = 7.46 \times 10^{-4} \text{ W}$$

Then

$$L_W = 10 \log\left(\frac{P}{10^{-12}}\right) = 10 \log\left(\frac{7.46 \times 10^{-4}}{10^{-12}}\right) = 89 \text{ dB}$$

14.4 Fan or Blower Noise

Fans (or blowers) are devices that use power-driven rotating impellers to move air. A fan has at least one inlet and at least one outlet. The rotating impeller imparts mechanical energy from the fan shaft to the airstream; the energy in the air exists as kinetic energy and the potential energy of air pressure. Different fan applications require fans with appropriate operating characteristics, including those of noise output. These characteristics are determined principally by the design of the rotating impeller.

Two principal types of fans—centrifugal fans and axial-flow fans—are normally used for central air-conditioning systems, industrial ventilation systems, and industrial processing applications. Although many manufacturers make no distinction between fans and blowers, the latter term is more often used to describe high-pressure devices used to convey material, for example a dust or a leaf blower. Centrifugal fans are low-pressure high-flow volume devices, whereas axial fans generally operate at higher pressures and tend to be noisier than

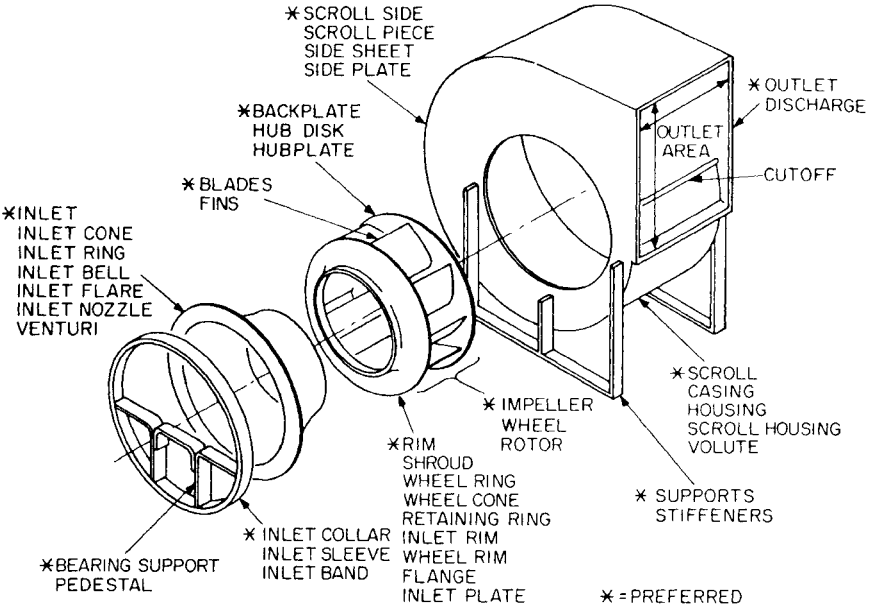


FIGURE 14.1. Exploded view of a centrifugal fan. (Harris, 1991, p. 41.2.).

centrifugal fans. The selection of the fan type, size, and speed depends first on the performance necessary to move a given amount of air against a specified pressure, and the noise characteristics are then established on a secondary basis. Figures 14.1 and 14.2 show the general construction of a centrifugal

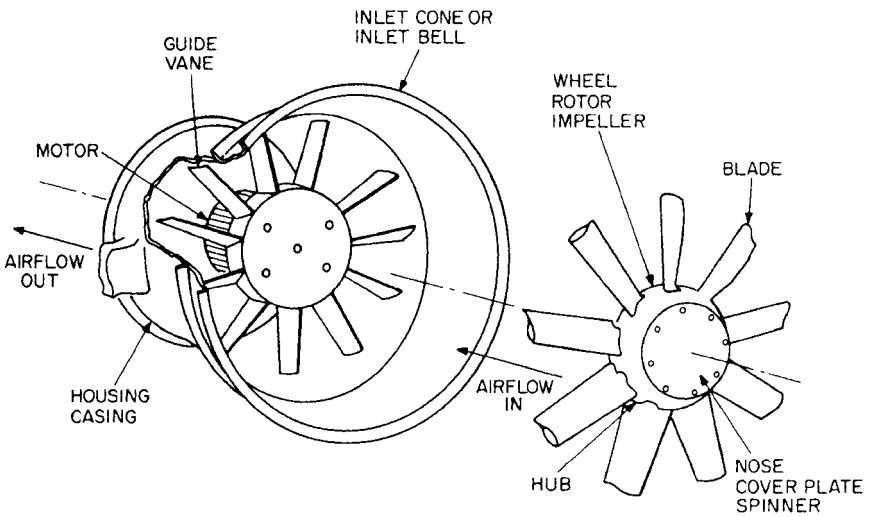


FIGURE 14.2. Components of an axial flow fan. (Harris, 1991, p. 41.2.).

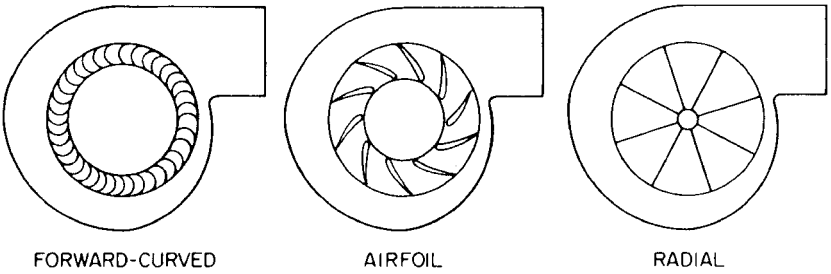


FIGURE 14.3. Three types of centrifugal fans. (Harris, 1991, p. 41.3.).

fan and an axial-flow fan, respectively, along with listing of commonly used nomenclature.

Centrifugal fans come in with a variety of blades. Three types are shown in Figure 14.3. Axial-flow fans divide into three main categories (vaneaxial, tubeaxial, and propeller), illustrated in Figure 14.4. In addition to flow and pressure requirements, fans are selected to meet environmental conditions, withstand corrosion, allow ease of maintenance, budget limits, and so on. The noise characteristics of various types of fans are fairly predictable, as they are not significantly altered by minor changes in the fan geometry.

Fan Noise Characteristics

Table 14.2 lists the broadband noise characteristics of typical fan designs. The average specific sound power levels in eight octave bands are given for well-designed fans installed in well-designed systems. This data can be utilized to estimate fan noise at the design stage. In the selection process during the design stage, actual noise data should be obtained from the manufacturer. Octave-band noise levels should be used in calculations. Single-number ratings for fan noise should be avoided.

When a blade passes over a given point, the air receives an impulse. The repetition rate of this impulse, termed the *blade frequency*, determines the fundamental tone that is produced by the blade. It can be predicted from

$$f_B = \frac{nN}{60} \tag{14.2}$$

where

f_B = blade frequency, Hz
n = fan speed, number of revolutions per minute (rpm)
N = number of blades in the fan rotor

According to Equation (11.23), the sound pressure level L_p from a specific source in a room depends on room conditions as well as on the sound power level L_W of that source. In order to predict the contribution of the fan noise to the room

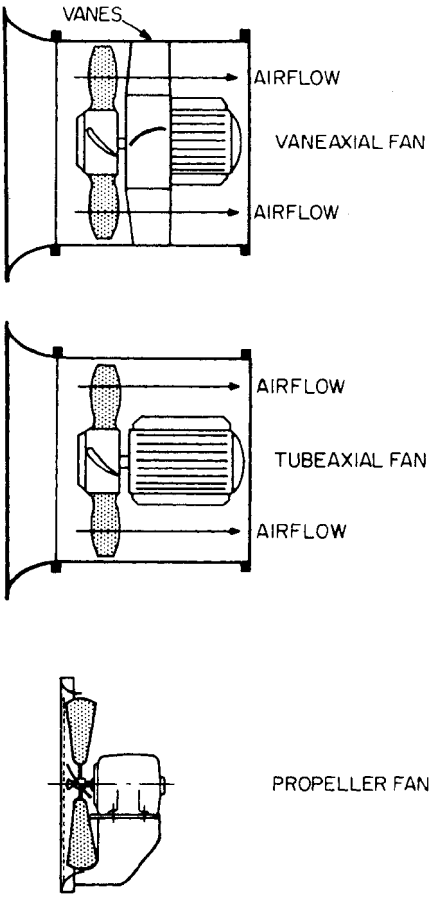


FIGURE 14.4. Three types of axial flow fans. (Harris, 1991, p. 41.3.)

sound level, the data on sound power levels of the fan models under consideration should be provided by the manufacturer. However, the manufacturer has no control over the system design for the room, nor that of the acoustical nature of the room and so cannot be responsible for the resultant noise level. If lower sound pressure levels are required than that generated by even properly designed fans, then it may be necessary to provide acoustic attenuators installed as an integral part of the fan assembly. Figure 14.5 shows a centrifugal fan with sound attenuators on both the inlet and the outlet, which is used in the supply system of a central station ventilating system. A sound attenuator is fitted at the outlet to lessen the flow of acoustic energy from the discharge of the air to the supply air ductwork.

Specific Sound Power Level

Specific sound power level is defined as the sound power level generated by a particular fan operating at an air flow rate of 1 m^3/s (2120 cfm) and at a pressure of

TABLE 14.2. Relative Sound Power Generated by Different Types of Fans.

Fan Type	Wheel Size	Octave-Band Center Frequency, Hz								
		63	125	250	500	1000	2000	4000	8000	BFI[a]
Centrifugal fans										
Airfoil or backward-curved	Over 0.75 m	85	85	84	79	75	68	64	62	3
or backward-*inclined*	Under 0.75 m	90	90	88	84	79	73	69	64	3
Radial fans:										
Low pressure (4–10 in.	Over 1 m	101	92	88	84	82	77	74	71	7
static pressure)	Under 1 m	112	104	98	88	87	84	79	76	7
Mediam pressure (10–20 in.	Over 1 m	103	99	90	87	83	78	74	71	8
static pressure)	Under 1 m	113	108	96	93	91	86	82	79	8
Highpressure (20–60 in.	Over 1 m	106	103	98	93	91	89	86	83	8
static pressure)	Under 1 m	116	112	104	99	99	97	94	91	8
Forward-curved	All	98	98	88	81	81	76	71	66	2
Axial fans										
Vaneaxial										
Hub ratio 0.3–0.4	All	94	88	88	93	92	90	83	79	6
Hub ratio 0.4–0.6	All	94	88	91	88	86	81	75	73	6
Hub ratio 0.6–0.8	All	98	97	96	96	94	92	88	85	6
Tubeaxial	Over 1 m	96	91	92	94	92	91	84	82	7
	Under 1 m	93	92	94	98	97	96	88	85	7
Propeller	All	93	96	103	101	100	97	91	87	5

[a] BFI = blade frequency increment.

Note: The data listed here are given in terms of specific sound power in dB re 1 $(10)^{-12}$ W based on a volume flow of 1 m^3/s and a total pressure of 1 kPa. Equation (14.3) must be used to adjust for actual pressures and volume flow rates. To convert these values into English units, subtract 45 dB in all bands. The base for the English units is a total presure of 1 in. water gauge and a volume flow rate of 1 ft^3/min. These values are those for total sound power radiated from the fan. To obtain the power levels at either the inlet or the outlet, subtract 3 dB from all bands. No change in BFI is to be made. In performing the calculations for the sound power levels from this table, do not use a total pressure less than 0.125 kPa. From applications where the total pressure is lower than 0.125 kPa, simply use the value of 0.125 kPa.

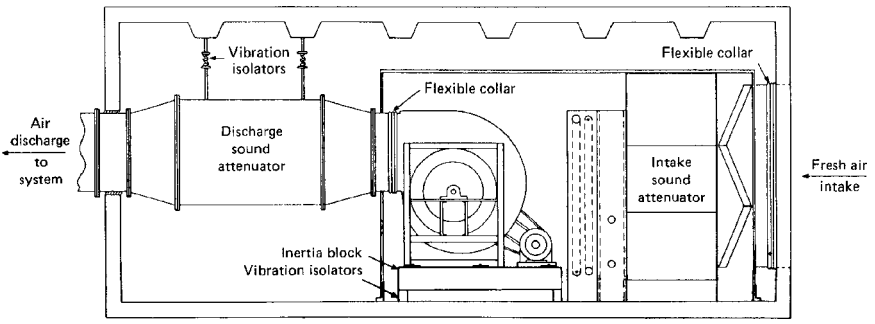


FIGURE 14.5. Centrifugal fan with sound attenuators at inlet and outlet. (Harris, 1991. P. 41.13).

1 kPa (4.0 in. water gauge). Table 14.2 lists the relative sound power for a variety of fan types. In reducing all fan noise data to this common base, the concept of specific sound power level allows direct comparison to be made between the octave band levels of different types of fans. A blade frequency increment (BFI) is also listed in Table 14.2; this represents the number of decibels that must be added to the level of the octave band, which includes the blade frequency in order to account for the presence of such a tone. Also, a means is provided for estimating the noise level of fans under actual operating conditions by a procedure that consists of the following steps:

1. Select the fan type and obtain the specific power levels in octave bands from Table 14.2. These sound power levels are expressed in decibels re 1 pW.
2. Adjust the octave-band levels for the volume flow rate and the operating pressure by adding to each octave band one of the following values:

$$10 \log Q + 20 \log p_t \text{ dB} \qquad \text{(for metric units)} \tag{14.3a}$$

or

$$10 \log Q + 20 \log p_t - 45 \text{ dB} \qquad \text{(for English units)} \tag{14.3b}$$

where

Q = volume flow rate, m^3/s or cfm
p_t = *total pressure*, kPa or inches water gauge

3. Account for the blade frequency component of the fan by adding the BFI for the fan type chosen to the octave-band level of that band which includes the blade frequency. The blade frequency is found from the use of Equation (14.2).
4. The sum of the above equals the total sound power level of the radiation from the inlet *and* the outlet. Subtract 3 dB from each octave band to yield the sound power level of radiation from the inlet *or* the outlet.

Example Problem 2

Consider a radial forward-curved fan with 24 blades, having a rotor diameter of 0.8 m, and operating at 750 rpm with a volume flow rate of 18 m^3/s and with a total pressure of 1.5 kPa.

Find the total sound output power at the inlet.

Solution

We use Table 14.2 and list the calculations at each step for each octave band level in Table 14.3. For step 2, using Equation (14.3a)

$$10 \log Q + 20 \log p_t = 10 \log 18 + 20 \log 1.5 = 16$$

TABLE 14.3. Calculation Results for Sound Power Level.

	Octave-Band Center Frequencies, Hz							
Procedural Step	63	125	250	500	1000	2000	4000	8000
1	98	98	88	81	81	76	71	66
2	16	16	16	16	16	16	16	16
3	0	0	2	0	0	0	0	0
4	−3	−3	−3	−3	−3	−3	−3	−3
Radiation form inlet	111	111	103	94	94	89	84	79

The blade frequency is from Equation (14.2)

$$f_B = \frac{nN}{60} = \frac{750 \times 24}{60} = 300\,\text{Hz}$$

which lies in the octave band with the center frequency of 250 Hz. The BFI is 2 dB. Step 3 is now completed. Step 4 consists of subtracting 3 dB from each octave band to obtain the noise from the inlet alone.

Fan Laws

Fan laws can predict fan performance quite well over a wide range of size and speed. These laws are as follows:

$$Q_a = Q_b \left(\frac{d_a}{d_b}\right)^3 \left(\frac{n_a}{n_b}\right) \tag{14.4}$$

$$p_{ta} = p_{tb} \left(\frac{d_a}{d_b}\right)^2 \left(\frac{n_a}{n_b}\right)^2 \tag{14.5}$$

$$P_a = P_b \left(\frac{d_a}{d_b}\right)^5 \left(\frac{n_a}{n_b}\right)^3 \tag{14.6}$$

$$L_{wa} = L_{Wb} + 70 \log\left(\frac{d_a}{d_b}\right) + 50 \log\left(\frac{n_a}{n_b}\right) \tag{14.7}$$

where

Q = volume flow rate, m^3/s
p_t = total pressure, kPa
P = fan power, kW
L_W = sound power level, dB re 1 pW
d = rotor diameter, m
n = rotor speed, rpm

Subscript a denotes the parameters for the base curve performance conditions, and subscript b denotes the parameters for the desired performance conditions.

Equation (14.7) is less accurate than Equations (14.4)–(14.6) for predicting performance characteristics but it is sufficiently accurate for estimating purposes. These fan laws state mathematically that when two fans have similar design configurations, their performance curves are similar, and at the equivalent point of rating on each performance curve, the efficiencies should be equal. In order to apply the fan laws, it is necessary to have the test data for one fan in the same design series. The applicability of the fan laws is restricted to cases where all linear dimension of the larger or smaller fan are proportional to the fan for which there are test data.

14.5 Electric Motors and Transformers

Electric motors convert electrical power into mechanical power. Some of the fundamental noises occurring in electric motors are caused by rotational unbalance, rotor/stator interaction, and slot harmonics. The electromagnetic force between the armature and the field magnet, or rotor and stator, gives rise to vibration. Noise is also generated by the excitation of natural frequencies of the motor structure, air resonance chambers, and the movement of air itself. The intensity of the noise is typically a function of rotational speed and motor type.

As a means of estimating the noise output of a motor, the following expression can be used to obtain the total sound power level in the bands 500, 1000, 2000, and 4000 Hz:

$$L_w = 20 \log \text{ hp} + 15 \log n + K_m \quad \text{dB} \tag{14.8}$$

where

$$\begin{aligned} \text{hp} &= \text{rated horsepower (1–300 hp)} \\ n &= \text{rated speed of motor, rpm} \\ K_m &= \text{motor constant} = 13 \text{ dB} \end{aligned}$$

Other more involved techniques have been developed, which entailed different motor constants for each of the octave bands of interest (Magrab, 1975; Webb, 1976).

Transformers exist for the purpose of stepping up or stepping down voltages. Their changing magnetic field causes deformation of the transformer coil, occurring at the alternating current (AC) frequency and higher harmonics, especially at the twice the AC frequency. This results in the characteristic frequency of 120 Hz in the United States and 100 Hz wherever 50-Hz AC current is used.

14.6 Pumps and Plumbing Systems

Pump noise arises from both hydraulic and mechanical sources, namely, cavitation, pressure fluctuations in the fluid, impact of mechanical parts, imbalance, resonance, misalignment, and so on. The hydraulic causes are, however, the predominant noise generators. Pumps will generate even more noise if they are not operated at rated

speed and discharge pressure, when the rate of compression is high or the inlet pressure is below atmospheric, or if the temperature runs too high. An unfortunate situation is created when the noise from the pump easily transmits through the fluid or piping to other system components.

Pumps generate two types of noise: discrete tones and broadband noise. The pump's fundamental frequency f_p is found from

$$f_p = \frac{n \times N_c}{60} \tag{14.9}$$

where

n = pump rotational speed, rpm

N_c = number of pump chamber pressure cycles per revolution

In large pumps the noise emission is the loudest at this fundamental frequency. As the pump size decreases, the frequency at which the maximum noise emission occurs increases, often to a frequency that constitutes a third or fourth harmonic of the fundamental. Above 3 kHz, the noise becomes more broadband, approaching an essentially flat spectrum. This is due to phenomena such as high-velocity flow and cavitation.

The total sound power level of pumps in the four octave bands of center frequencies 400, 1000, 2000, and 4000 kHz can be estimated from the following:

$$L_w = 10 \log \text{hp} + K_p \quad \text{dB} \tag{14.10}$$

where K_p = pump constant which has the following values: 95 dB for the centrifugal type, 100 dB for the screw type, and 105 dB for reciprocating type. For rated speeds below 1600 rpm, 5 dB is to be subtracted for reciprocal pumps. The sound power in each of the four bands may be considered to be 6 dB less than the total sound power L_w computed from Equation (14.10).

Controlling Noise in Plumbing Systems

Table 14.4 lists the sources of noise in a building's plumbing system and their likelihood of being annoying. Flow can be either *laminar* flow, i.e., smoothly flowing, or *turbulent flow*, in which occurs an irregular, random motion of the fluid particles. The influencing factor that determines whether a flow will be turbulent or laminar is the *Reynolds number Re*, a dimensionless parameter defined by

$$Re = \frac{\rho d v}{\mu}$$

where ρ is the density of the fluid, d is the internal pipe diameter, v is the flow velocity, and μ is the absolute viscosity of the fluid. For $Re < 2000$, the flow is laminar. For transition region $2000 < Re < 4000$, the flow may be either laminar or turbulent. For $Re > 4000$, the flow will be turbulent. Noise generated by laminar flow tends to be quite low in intensity and is usually of no concern.

In most real plumbing systems, the velocities are sufficiently high to result in turbulent flow, which is a basic mechanism for noise generation within piping runs

TABLE 14.4. Different Type of Plumbing Noises. Their Means of Generation, and Annoyance Potentials (from Harries, 1991, Ch. 44).

Plumbing System Component/Equipment	Generation Mechanism	Potential Annoyance
Piping runs		
Couplings	Turbulence	Minimal
Elbows	Turbulence	Minimal
Tees	Turbulence	Minimal
Fixtures		
Bar sink	Cavitation/turbulence/splash/waste flow	Minimal
Bath tub	Cavitation/turbulence/splash/waste flow	Very significant
Bidet	Cavitation/turbulence/splash/waste flow	Nominal
Flushometer	Cavitation/turbulence	Significant
Hose pipe valves	Cavitation/turbulence	Nominal
Laundry tubs	Cavitation/turbulence/splash/waste flow	Nominal
Pressure regulator	Cavitation/turbulence	Nominal
Shower	Cavitation/turbulence/splash/waste flow	Very significant
Sink	Cavitation/turbulence/waste flow	Significant
Valves	Cavitation/turbulence	Significant
Water closet, tank stool	Cavitation/turbulence/splash/waste flow	Very significant
Urinal	Cavitation/turbulence/splash/waste flow	Nominal
Appliances		
Dishwasher	Vibration/cavitation/spray/water hammer	Very significant
Drinking fountain	Cavitation/turbulence	Minimal
Washing machine	Vibration/cavitation/impact/motor/ water hammer	Very significant
Waste disposal	Vibration/waste flow	Very significant
Water heater	Cavitation/turbulence	Minimal
Supply and waste pumps		
Booster	Rotational flow/cavitation/motor	Significant
Recirculation	Rotational flow/cavitation/motor	Normal
Sewage	Rotational flow/cavitation/motor	Significant
Sump	Rotational flow/cavitation/motor	Significant

and fixtures of the plumbing system. A potentially great cause of noise is cavitation, which is the formation and subsequent collapse of cavities (bubbles) within the flow of water through and past a restriction in the flow. For cavitation to occur, a localized restriction or a projection must exist within the piping system, which ensues in localized high velocities and low pressures. The formation and sudden collapse of these bubbles result in extreme local pressure fluctuations, which can be detected as noise. Other water noises occur from splashing (i.e., impact of liquid striking a surface) and waste water flow (i.e., flow into drainpipes).

Much more serious is the sharp intense noise known as *water hammer*. It occurs when a steady flow in a liquid flow system is suddenly interrupted, for example, by a quick-action valve. When the fluid is in motion throughout a piping system, even at relatively low velocities, the momentum from this sudden interruption can be quite large. The sudden interruption of the flow creates an extremely sharp pressure

rise that propagates as a shock wave upstream from the interrupting valve. The steep wavefront can be reflected numerous times back and forth throughout the various parts of the piping system until the energy is finally dissipated.

Noise control factors which must be dealt with in order to effect noise control include (a) water flow and piping characteristics, (b) radiation of sound to the building structure, (c) selection and mounting of fixtures, (d) isolation of pump systems, and (e) water hammer noise control.

Water pressure in a plumbing system influences the flow noise caused by pipe runs and water supply valves. According to typical building codes, water pressure should be maintained at least at 100 kPa (15 psi) but not more than 500 kPa (80 psi). For acceptable system performance, the supply pressure should be somewhere between 230 and 370 kPa (35 and 55 psi), with a preference toward the lower range in order to minimize noise. Flow noise radiation from pipes can be lessened by minimizing the number of pipe transitions (elbows, tees, y-connections, and the like). This reduces the opportunity for the onset of turbulence and cavitation. Larger pipes are used in building design when noise control is given a higher priority. In the U.S. $1/2$-inch diameter piping is generally used in domestic plumbing systems but can be as high as $3/4$-inch diameter to cut down on noise by as much as 3–5 dB.

Noise resulting from water flow in pipes may be transmitted to the rooms through which they pass, particularly if they are in direct contact with large radiation surfaces such as walls, ceilings, and floors. Isolation of these pipes from the building structure provides significant noise reduction. If the pipes are mounted with foam isolators, instead of being rigidly attached to the building structure, a considerable noise reduction of 10–12 dB may be obtained. Whenever piping passes through a structure (block, stud, joist, or plate) or is in contact with a wall or masonry, resilient materials such as neoprene or fiberglass should be used to provide isolation. It is vital to seal with a resilient caulking around the perimeter of all pipes, faucets, and spouts that penetrate through floors, walls, and shower stalls.

The impact of water hammer can rupture piping; thus, it will spring leaks, cause weakening of connections and produce damage to valves. Water-hammer pulsing associated with the use of washing machines and dishwashers can be partially damped by connecting these machines to the water supply with extra-long flexible hose. Figure 14.6 shows a schematic of a capped pipe that incorporates an air chamber for water-hammer suppression. The length of the pipe ranges from 30 to 60 cm and may be of the same or larger diameter than the line it serves. The volume of the air chamber, which serves as an air cushion, depends on the nominal pipe diameter, the branch line length, and the supply pressure. But if the air chamber becomes filled with water, it becomes ineffective. A petcock is provided along with a shut-off valve, so that the chamber may be drained of water and vented, thereby reactivating the unit.

There are commercial devices, called *water-hammer arresters*, which are not subject to the limitation of capped pipes, because a metal diaphragm separates the water from the air. These devices are best placed near quick-acting valves and should also be installed at the termini of long pipe runs.

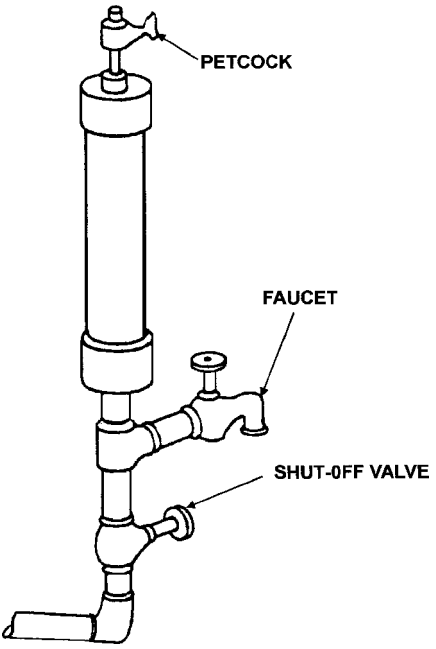


FIGURE 14.6. A capped pipe which serves to arrest water hammer.

14.7 Air Compressors

Air compressors are widely used in industry, and they are identified as a major source of noise. Usually driven by a motor or a turbine, these devices are used to elevate the pressure of air or another gas. The noise emission characteristics are a function of the type of unit. In portable air compressors the driving engine is the major source of noise rather than the compressor itself.

The second largest source is the cooling fan. Compressors are either rotary or reciprocating. A reciprocating compressor typically generates a strong low-frequency pulsating noise, the characteristics of which are dependent upon the rotational speed and the number of cylinders. In centrifugal compressors the noise generated is a function of a number of parameters such as the interaction of the rotating and stationary vanes, the radial distance between impeller blades and diffuser vanes, the rotational speed, the number of stages, the inlet design, the horsepower input, turbulence, the molecular weight of the gas undergoing compression, and the mass flow.

As in the case of fans, the blade passage frequency constitutes an importance frequency component in certain types of compressors. In the diffuser-type machines, the *blade-rate component*, arising from the movement of one set of blades past another, is of primary importance. The frequency of this component is found

from

$$f_{\mathrm{BRC}} = \frac{N_r \times N_s}{60\ K_{\mathrm{BRC}}} \times n \quad \mathrm{Hz} \tag{14.11}$$

where

f_{BRC} = blade-rate component frequency, Hz
N_r = number of rotating blades
N_s = number of stationary blades
K_{RBC} = greatest common factor of $\mathrm{N_r}$ and $\mathrm{N_s}$
n = rotational speed, rpm

Example Problem 3

Find the frequency of the blade rate component of a diffuser-type compressor with $N_r = 16$ and $N_s = 24$ and operating at a speed of 6000 rpm.

Solution

Using Equation (14.11) we find that

$$f_{\mathrm{BRC}} = \frac{N_r \times N_s}{60\ K_{\mathrm{BRC}}} \times n = \frac{16 \times 24}{60 \times 8} \times 6000 = 4800\,\mathrm{Hz}$$

If the blade-rate component frequency falls within the audio range, we should expect an increase of several decibels in the octave-band sound power level in which it occurs.

The total sound power level in the four octave bands with center frequencies of 500, 1000, 2000, and 4000 can be roughly estimated for both reciprocating and centrifugal compressors from the expression

$$L_w = 10 \log \mathrm{hp} + K_c \quad \mathrm{dB} \tag{14.12}$$

where K_c = air compressor constant = 86 dB for the 1–100 hp range. Equation (14.12) is very similar to Equation (14.10) for pumps, and either expression will yield a straight line in a semi-log plot. Also, it is not unreasonable to estimate that the sound power is equally divided among the four octave bands. Thus, each band level is 6 dB below the total estimated from Equation (4.12).

14.8 Gears

Internal combustion engines and electric motors generally operate at speeds of one to several thousand revolutions per minute. These high speeds help maximize the power-to-weight and power-to-initial cost ratios. Gearing and other speed reducers are applied when the driven machinery requires high torque and low speeds.

Mechanical power, shaft speed, and torque are related as follows:

$$P_{kW} = 10^{-6}\, T\omega \tag{14.13}$$

where

P_{kW} = transmitted power, kW
T = torque, N mm
$\omega = \pi n/30$—angular velocity, rad/s
n = shaft speed, rpm

In English units, Equation (14.13) becomes

$$P_{hp} = \frac{Tn}{63{,}025} \tag{14.14}$$

where

P_{hp} = transmitted power, hp
T = torque, lb-in
n = shaft speed, rpm

It was observed by Hand (1982) that gear noise increases with speed at the rate of 6–8 dB per doubling of speed. It was also observed that an increase of 2.5–4 dB in gear noise occurs for each doubling of load. Thus, according to Equation (14.13) or (14.14) a reduction in speed results in an increase in torque, if the transmitted power is to be sustained, and so the noise effect of speed reduction is somewhat offset by the increase in the torque.

Meshing Frequencies

The profile of most gear teeth is that of an involute curve. Force transmits through the driving gear to the driven gear along the line-of-action, which is fixed in space (excepting planetary gear trains). If the gears are ideal, perfectly fabricated, rigid, and transmitting constant torque, the power should be transmitted smoothly and without vibration or noise. Real gears, however, have tooth errors in spacing and tooth profile, and in some cases, an appreciable shaft eccentricity. Gear teeth do act elastically and flex slightly under load. Consequently, the driving gear teeth that are not in contact deflect ahead of their theoretical rigid-body positions, while the driven teeth that are not in contact lag behind their theoretical positions. This results in a rather abrupt transfer of force when each pair of teeth comes into contact, instantaneously accelerating the driven gear and decelerating the driving gear. The fundamental frequency of the noise and vibration is given by

$$f = \frac{nN}{60} \tag{14.15}$$

where

f = the fundamental tooth meshing frequency, Hz
n = rotational speed of the gear in question
N = the number of teeth in the gear in question

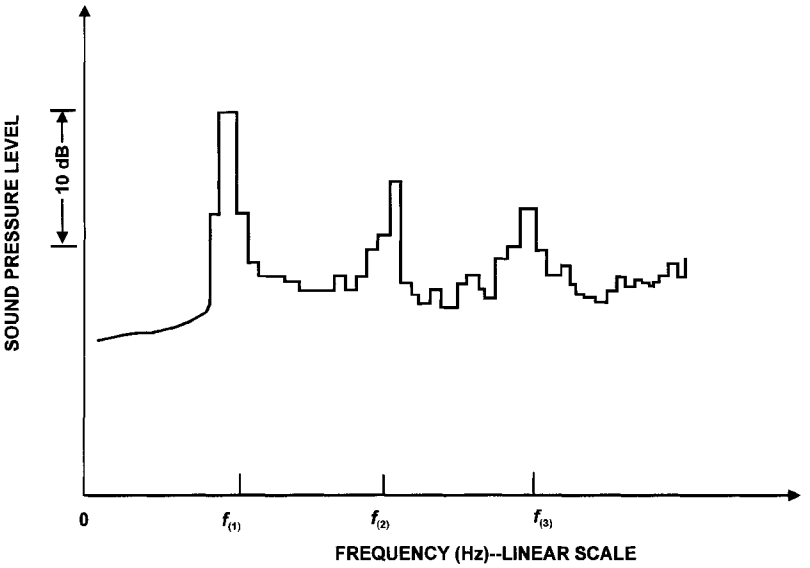


FIGURE 14.7. Gear noise spectrum.

Harmonics—the noise and vibration at integer multiples of the tooth meshing frequency—are usually present. The most significant contributions are usually made by the first two or three harmonics, $f_2 = 2f_1$, $f_3 = 3f_1$, and so on. A typical gear noise spectrum is displayed in Figure 14.7.

Tooth Error

If a single tooth is imperfectly cut, chipped, or damaged, then it will generate a noise impulse once every shaft revolution. The fundamental frequency of the noise or vibration due to tooth error is given by

$$f_{1E} = \frac{n}{60} \tag{14.16}$$

where

f_{1E} = fundamental frequency due to tooth error, Hz
n = shaft speed

Harmonics of tooth-error frequency may also occur. Moreover, if the shaft centerline is not straight or if a gear or bearing is not concentric with the shaft centerline, noise and vibration at the tooth-error frequency may result. These can cause in *sideband frequencies* f_S that accompany the tooth meshing frequencies. These frequencies are found as follows

$$f_S = f_1 \pm f_{1E} \tag{14.17}$$

Gear Trains

Consider a pair of nonplanetary gears $a$and b in mesh. The tangential velocity V_t at the mesh is the same for both teeth. But the V_t of each tooth of gear i of radius r_i rotating at angular velocity ω_i is given by

$$V_t = r_i \times \omega \tag{14.18}$$

The number of teeth N_i on a gear i is directly proportional to the radius r_i of the gear, and $\omega_i = 2\pi n_i/60$. Since $r_a \times \omega_a = r_b \times \omega_b$ for gears a and b at their mesh point, we obtain

$$\frac{n_a}{n_b} = \frac{N_b}{N_a} \tag{14.19}$$

Thus, the ratio of rotational speeds is inversely proportional to the ratio of numbers of teeth: If both gears are external spur gears, the speed ratio is negative, i.e., one gear will turn clockwise and the other counterclockwise. If one gear is an internal gear (i.e., a ring gear) the speed ratio is positive, meaning that both gears will rotate in the same angular direction. For a gear train comprised of several gears, the output to input speed ratio is given by

$$\frac{n_{\text{output}}}{n_{\text{input}}} = \prod_i \left(\frac{N_{\text{driving}}}{N_{\text{driven}}} \right)_i \tag{14.20}$$

An idler gear serves as both a driving gear and a driven gear. Idler gears must be included in determining the direction of rotation of shafts in a gear train.

Example Problem 4

In Figure 14.8 showing a gear train, gear 1 on the input shaft has 40 teeth and rotates at 3600 rpm. Gears 2, 3, and 4 have 90, 44, and 86 teeth, respectively. Narrow band spectrum analysis of noise and vibration shows discrete tones and vibration energy peaks of 14, 27, 1227, 2400, 2455, and 4800. Establish the possible contributions to the noise and vibration at these frequencies.

Solution

Gear speeds are determined from

$$\frac{n_2}{n_1} = \frac{N_1}{N_2}$$

from which we obtain

$$n_2 = n_3 = 3600 \times \frac{40}{90} = 1600\,\text{rpm}$$

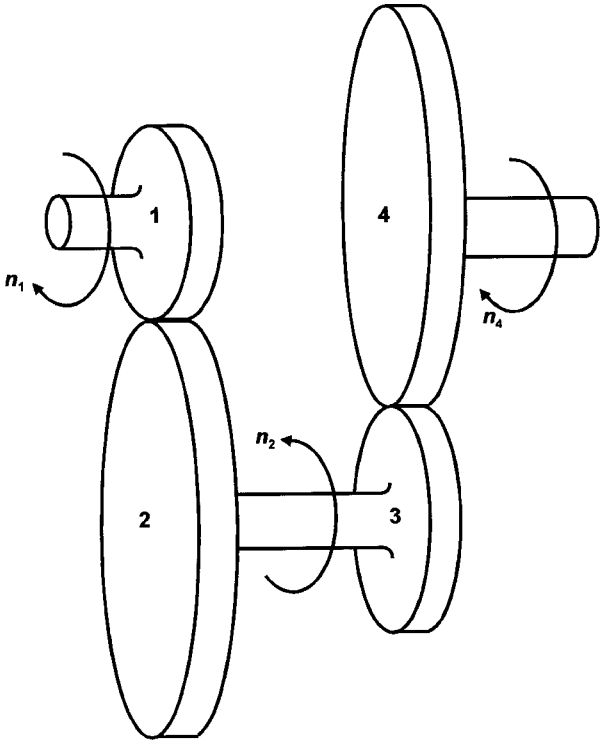


FIGURE 14.8. Gear train for example problem 4.

From Equation (14.20) we have

$$\frac{n_{\text{output}}}{n_{\text{input}}} = \frac{40 \times 44}{90 \times 86} = 0.227$$

and

$$n_{\text{output}} = 3600(0.227) = 818.7\,\text{rpm}$$

Fundamental tooth-error frequencies are given by

$$\begin{aligned} f_{1\text{E}} &= \frac{n}{60} \\ &= \frac{3600}{60} = 60\ \text{Hz for gear 1} \\ &= \frac{1600}{60} = 26.7\ \text{Hz for gears 2 and 3} \\ &= \frac{818.7}{60} = 13.6\ \text{Hz for gear 4} \end{aligned}$$

Fundamental tooth meshing frequencies are obtained from

$$\begin{aligned} f &= \frac{nN}{60} \\ &= 3600 \times \left(\frac{40}{60}\right) = 2400 \text{ Hz for gear 1} \\ &= 1066 \times \left(\frac{46}{60}\right) = 1226.7 \text{ for gears 3 and 4} \end{aligned}$$

If we compare the calculated results with the spectrum analysis, it seems that the 14- and 27-Hz discrete tones and vibration energy peaks correspond to the fundamental and the first harmonic of the tooth-error frequency for gear 4. This would indicate that the shaft straightness and the teeth of gear 4 should be investigated. The discrete tones of 1227 and 2455 Hz correspond to the fundamental and the first harmonic of the meshing frequencies of gears 3 and 4. Discrete tones at 2400 and 4800 Hz correspond to the fundamental and the first harmonic of the tooth meshing frequency of gears 1 and 2.

Contact Ratio (CR)

The contact ratio CR for a pair of gears is defined as the average number of pairs of teeth in contact. At least one pair must obviously be in contact at all times. Taking into fact that tooth error, wear, shaft deflection, and machining tolerances affect the way gears work, a contact ratio of 1.2 is selected as the practicable minimum. When the contact ratio is near minimum value, contact commences at the tips of the driven teeth, and impact loads due to teeth meshing are high. A contact ratio of 2 or more usually results in better tooth load distribution and quieter operation, since two or more teeth will be in contact at all times.

In order to deal with contact ratio, we need to briefly review spur gear terminology: diametral pitch is defined by $P_d = N/d$ and d is the pitch (nominal) diameter of the gear, r is the pitch radius $= d/2$, ϕ is the pressure angle, and a is the addendum or extent of the tooth beyond the pitch radius, $r_a = r + a$ is the radius of the addendum circle, and $c = r_1 + r_2$ (center distance for gears in mesh). It follows from Equation (14.19) for spur gears that

$$\frac{r_1}{r_2} = \frac{n_2}{n_1}$$

The pressure angle ϕ represents the angle of force transmission between gear teeth. Standard gears have pressure angles of 14.5° (a rather antiquated standard), 20°, and 25°. The contact ratio between a pair of spur gears is given by

$$\text{CR} = \frac{\sqrt{r_{a1}^2 - r_1^2 \cos^2\phi} + \sqrt{r_{a2}^2 - r_2^2 \cos^2\phi} - c \sin\phi}{\dfrac{\pi \cos\phi}{P_d}} \tag{14.21}$$

The subscripts 1 and 2 refer to the driver and driven gears, respectively. For full depth gears with standard pressure angles, the standard addendum is $a = m = 1/P_d$. For 20° pressure angle stub teeth, the standard addendum is $\underline{a} = 0.8m = 0.8/P_d$.

Gears are selected to meet specified power and speed ratio requirements. These requirements affect the selection of gear material, tooth width, module or diametral pitch, number of teeth, and so on. If the number of teeth is increased for each gear in a train, the contact ratio will usually increase, with the result the tooth meshing frequency and the accompanying noise level will decrease.

Helical Gears

When a pair of spur gear teeth begins to mesh, the contact occurs at once across the entire face width. In the case of a helical gear in which teeth entwine about an axial surface rather than cut straight across (Figure 14.9), the contact occurs gradually, with the contact beginning at a point and then extending across the tooth face. This results in reduced vibration and lower impact loads that produce tooth meshing frequency noise. Hence, helical gears on parallel shafts may be substituted for spur gear trains to reduce noise; and the actual number of teeth in contact will also be substantially increased. In order to ensure smooth and quiet operation, it is recommended that the thickness of the helical gear be 1.2 to 2 times the axial pitch (the distance between corresponding points on two adjacent teeth). Thrust (axial) loads must be taken into consideration when specifying helical gears in the design of shafts and bearings. Well-made helical gears cost more than spur gears, but the payback in terms of smoother and quieter operation can more than offset the greater expense.

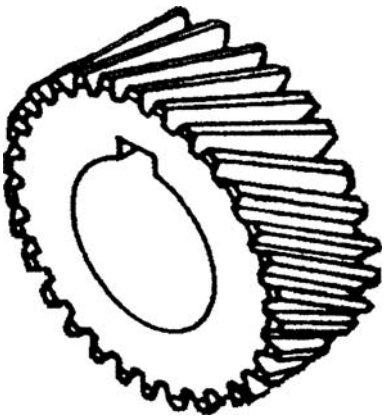


FIGURE 14.9. Helical gears. (Courtesy of Designatronics, Inc., New Hyde Park, NY.).

Other Aspects of Gear Noise Control

Gear production methods involve die casting, milling, drawing, extruding, stamping, and production from sintered metal. These methods tend to produce gears with significant tooth error to a more or lesser degree. Most gears produced by milling cutters also tend to have their tooth forms only approximate the correct configuration. In order to produce a precise tooth form, each milling cutter should be dedicated to only one diametral pitch or module and for a specific tooth number. In practice, a different milling cutter is used for each diametral pitch, but each cutter is used for a range of tooth numbers. Inaccuracies in cast, drawn or extruded gears result from shrinkage and other dimensional changes.

Precision methods of generating gear teeth usually result in cutting down noise due to tooth error. These include computer-controlled generating rack cutter, the generating gear shaper cutter, and the generating hob. Precision gears are usually finished by grinding and shaving, and other finishing methods include burnishing, honing, and lapping.

Large gears benefit by damping to absorb vibration energy. Constrained layer damping, which sandwiches a layer of damping material between the gear web and a rigid steel plate, can reduce gear noise. If loads are low and the temperatures are not excessive for the material, fiber, plastic, fiberglass reinforced, and composite gears are used in such applications. These materials have a low modulus of elasticity and high internal damping compared with steel. Shock loads from tooth meshing and tooth error are absorbed, thereby reducing noise.

Gear enclosures or housing can and should be designed to control noise. The enclosure should be isolated so as not to transmit vibration to adjacent structures; and provisions for adequate lubrication of the gears should be included. Resonances of the enclosure should not correspond to the tooth meshing or any other excitation frequency; otherwise the housing will radiate a large amount of noise energy. This can be avoided by stiffening the enclosure structure to "tune out" its resonance to a higher frequency. The stiffeners themselves should be designed so that they have a low-radiation efficiency. Advantage should be taken of the directivity patterns of noisy gear trains in the orientation of machinery, so that personnel noise exposure is minimized.

14.9 Journal Bearings

A journal bearing is the simplest type of bearing. It consists of a portion of a shaft rotating inside a circular cylinder with a layer of lubrication separating the shaft and bearing surfaces. Hydrodynamic rotation depends on shaft rotation to pump a film of lubricant between the shaft and the bearing. If the lubricant viscosity and shaft rotational speed are adequate for the load on the project bearing area, then thick-film hydrodynamic lubrication will prevail. This ensures stable operation without metal-to-metal contact, as well as quiet operation. Starting, stopping, and

load direction changing may result in temporary cessation of the lubricant film and in metal-to-metal contact. This calls for *hydrostatic* lubrication that introduces fluid (e.g., oil or air) to the bearing surface at a pressure sufficient to support the shaft, even when the bearing is stationary. A large increase in the noise level produced by a journal bearing is apt to indicate a failure of the lubrication system.

14.10 Ball and Roller Bearings

An incipient failure of ball or roller bearing can be detected by noise and vibration measurements. For the most part, bearing life can be predicted but premature failure can and does occur. The occurrence of discrete tones in the noise spectrum for an operating bearing may indicate manufacturing defects, and narrowband vibration spectra can be utilized to detect wear and defects.

A ball or roller bearing consists of the following elements: a set of balls or rollers are enclosed in a cage or separator sandwiched between an inner race (that is usually attached to the shaft) and the outer race (usually attached to the supporting stationary structure). When the shaft turns, the balls (or rollers) roll along the surface of the inner raceway, which allows for the rotational freedom of the shaft. The cage or separator follows the motion of the balls. The speed relationships are as follows

$$\frac{n_O - n_C}{n_I - n_C} = -\frac{D_I}{D_O} \tag{14.22a}$$

and

$$\frac{n_B - n_C}{n_I - n_C} = -\frac{D_I}{D_B} \tag{14.22b}$$

where

$$\begin{aligned} n &= \text{rotational speed, rpm} \\ D &= \text{diameter, mm or in.} \end{aligned}$$

and the subscripts denote the variables as follows:

$$\begin{aligned} I &= \text{inner race} \\ O &= \text{outer race} \\ B &= \text{ball or roller} \\ C &= \text{ball cage or separator} \end{aligned}$$

The race diameters are measured at the point of contact with the balls or rollers.

Ball or Roller Bearings with Stationary Outer Races

In most bearing applications the outer race is stationary. Equation (14.22a) becomes

$$\frac{0 - n_C}{n_I - n_C} = -\frac{D_I}{D_O} \tag{14.23}$$

which yields the separator speed

$$n_C = \frac{n_I D_I}{D_O + D_I} \tag{14.24}$$

and the speed of the balls relative to the separator is given by

$$n_B^R = n_B - n_C = -\frac{(n_I - n_C)D_I}{D_B} = -\frac{n_I^R D_I}{D_B} \tag{14.25}$$

Here n_I^R is the speed of the inner race relative to separator speed. Equation (14.25) is obtained by eliminating n_B through the use of relation (14.22b).

Vibration and noise frequencies are related to the absolute speeds n, relative speeds n^R, and the number of balls or rollers ν_B. For bearings with stationary outer races, the following fundamental frequencies are likely to arise:

1. Shaft imbalance:

$$f = \frac{n_I}{60} \tag{14.26}$$

2. Outer race defect:

$$f_O = \frac{n_O^R \nu_B}{60} = \frac{n_C \nu_B}{60} \tag{14.27}$$

3. Inner race defect:

$$f_I = \frac{n_I^R \nu_B}{60} = \frac{(n_I - n_C)\nu_B}{60} \tag{14.28}$$

4. Defect or damage to one ball or roller:

$$f_B = \frac{2n_B^R}{60} = \frac{n_B - n_C}{30} \tag{14.29}$$

5. Imbalance or damage in the separator or cage:

$$f_C = \frac{n_C}{60} \tag{14.30}$$

With the appearance of the fundamentals as calculated above, harmonics $2f$, $3f, \ldots$ are apt to occur.

Example Problem 5

Consider a ball bearing that uses twelve 8-mm-diameter balls. The inner and the outer diameters are $D_I = 28$ and $D_O = 44$ mm, measured at the ball

contact points. The outer race is stationary and the inner race rotates at speed $n_I = 4500$ rpm clockwise. Determine (a) the rotational speeds and (b) the noise and vibration frequencies which will occur if there are defects and imbalance present.

Solution

It should be obvious that $D_I + 2D_B = D_O$.

(a) Use Equation (14.24) to find the separator speed

$$n_C = \frac{n_I D_I}{D_O + D_I} = 4500 \times \frac{28}{28 + 44} = 1750\,\text{rpm}$$

and, from Equation (14.25), the speed of the balls relative to the separator is

$$n_B^R = n_B - n_C = -\frac{(n_I - n_C)D_I}{D_B} = -(4500 - 1750) \times \frac{28}{8}$$
$$= -9625\ \text{rpm (counterclockwise)}$$

(b) We find the fundamental frequencies as follows:

1. Shaft imbalance:

$$f = \frac{n_I}{60} = \frac{4500}{60} = 75\,\text{Hz}$$

2. Outer race defect:

$$f_O = \frac{n_O^R \nu_B}{60} = \frac{n_C \nu_B}{60} = \frac{1750 \times 12}{60} = 350\,\text{Hz}$$

3. Inner race defect:

$$f_I = \frac{(n_I - n_C)\nu_B}{0} = \frac{(4500 - 1750) \times 12}{60} = 550\,\text{Hz}$$

4. Defect or damage to one ball:

$$f_B = \frac{2n_B^R}{60} = \frac{n_B - n_C}{30} = -\frac{9625}{30} = 320.8\,\text{Hz}$$

Notice that the negative sign is ignored in this answer.

5. Imbalance or damage in the separator or cage:

$$f_C = \frac{n_C}{60} = \frac{1750}{60} = 29.2\,\text{Hz}$$

Harmonics of all of the above frequencies are likely to occur.

Ball or Roller Bearings with Nonstationary Outer Races

For some applications, the shaft attached to the inner race of a bearing remains stationary while the outer race rotates. In this case, $n_I = 0$ and we obtain the

following speed relationships:

$$\frac{n_O - n_C}{0 - n_C} = -\frac{D_I}{D_O}$$

with carrier speed

$$n_C = \frac{n_O D_O}{D_O + D_I}$$

and speed of the balls with respect to the carrier

$$n_B^R = n_B - n_C = -\frac{(n_O - n_C)D_O}{D_B} = -\frac{n_O^R D_O}{D_B}$$

The frequencies that may occur when the outer race is nonstationary are as follows:

1. Shaft imbalance:

$$f = \frac{n_O}{60}$$

2. Outer race defect:

$$f_O = \frac{n_O^R \nu_B}{60} = \frac{(n_O - n_C)\nu_B}{60}$$

3. Inner race defect:

$$f_I = \frac{n_I^R \nu_B}{60} = \frac{n_C \nu_B}{60}$$

4. Defect or damage to one ball or roller:

$$f_B = \frac{2n_B^R}{60} = \frac{n_B - n_C}{30}$$

5. Imbalance or damage in the separator or cage:

$$f_C = \frac{n_C}{60}$$

Harmonics will also constitute additional components of the spectrum.

For any bearing fault, the noise and vibration spectra will be essentially tonal, i.e., characteristic peaks will exist. A great deal of the noise energy will be concentrated in the bands that include the fundamental frequency and its first and second harmonics.

14.11 Other Mechanical Drive Elements

Chain Drives

Figure 14.10 shows a roller chain meshing with toothed sprockets. The roller chain is constructed of side plates and pin and bushing joints designed to mesh with sprockets. The flexibility of the chain aids in limiting shock and vibratory forces, but the initial contact between the chain and sprockets can be noisy at high speeds.

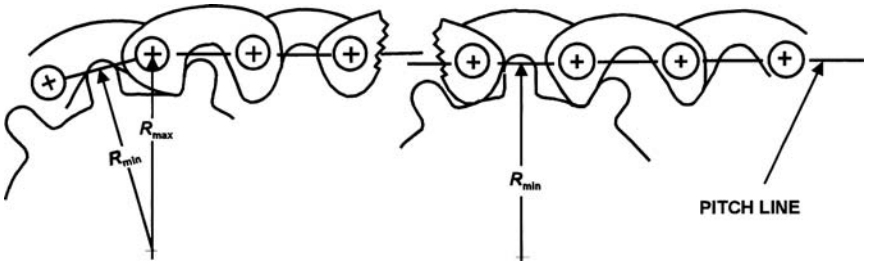


FIGURE 14.10. Roller chain.

For transmission of high loads at high speeds, inverted-tooth (or "silent") chains are often applied. Because the inverted-tooth chain engages with the sprockets with less impact force, it generally operates more quietly than roller chain.

The links are of finite length in either roller chain or inverted-tooth chain. While the chain links engage a sprocket, the pitch line (i.e., the centerline of the link pins) bobbles up and down due to the chordal action. In Figure 14.10, the center of a link pin is located at $R_{\max}$ from the center of the sprocket over which it moves. $R_{\max}$ is the maximum center of the pitch line from the sprocket center. As the link engages further, the pitch line moves nearer to the sprocket center and both pins of the link coincide with the pitch line. The new position is given by

$$R_{\min} = R_{\max} \cos\left(\frac{180^\circ}{N_S}\right) \tag{14.31}$$

where N_S indicates the number of sprocket teeth. Roller chains usually have a minimum of 8 teeth and inverted-tooth chains, 17 teeth.

Consider a drive sprocket rotating at constant speed n_s. The pitch line velocity of the chain will change in direct proportion to its distance from the center of the sprocket, i.e., $V = R \times (2\pi n_s)$. The fractional velocity change $V/\Delta V$ is given by

$$\frac{\Delta V}{V} = \frac{R_{\max} - \mathrm{R}_{\mathrm{mean}}}{R_{\max}} = \frac{1 - \dfrac{1 + \cos(180^\circ/N_S)}{2}}{\dfrac{1 + \cos(180^\circ/N_S)}{2}} = \frac{1 - \cos(180^\circ/N_s)}{1 + \cos(180^\circ/N_s)} \tag{14.32}$$

The angular velocity $(2\pi n_s)$ is canceled itself out in the ratio of Equation (14.32), but the speed ratio of a chain drive is given by

$$\frac{n_2}{n_1} = \frac{N_1}{N_2} \tag{14.33}$$

where n refers to the rotational speed (rpm) of the sprocket and N the number of sprocket teeth. Subscripts 1 and 2 denote the driver sprocket and the driven sprocket, respectively. Fundamental frequencies of noise and vibration can be due to the following:

1. Imbalance in the driver shaft or the driven shaft or damage to one sprocket tooth:

$$f_1 = \frac{n_1}{60}\,\text{Hz} \quad \text{and/or} \quad f_2 = \frac{n_2}{60}\,\text{Hz} \tag{14.34}$$

2. Damage to one chain link: In this case for a chain with N_L links, a defective link will strike each sprocket $n_1 N_1 / N_L$ times per minutes. Hence

$$f_{CL} = \frac{n_1 N_1}{30 N_L} \tag{14.35}$$

3. Tooth engagement and chordal action:

$$f_{TE/CA} = \frac{n_1 N_1}{30} \tag{14.36}$$

When the above fundamentals are present, harmonics are likely to occur. Noise levels and vibratory amplitudes due to tooth meshing and chordal action are functions of the sprocket configuration and also speeds and masses of the sprockets and chains. Also, if there are idler sprockets in the system, the values given in Equations (14.34)–(14.36) will undergo a change.

Example Problem 6

Given a 12-tooth drive sprocket rotating at 2400 rpm, a driven sprocket with 24 teeth and a chain with 60 links, determine the velocity variation with respect to the mean velocity and the possible vibration and noise frequencies due to this chain drive.

Solution

We apply Equation (14.32) to the smaller sprocket, i.e., the driver, because the ratio is larger:

$$\frac{\Delta V}{V} = \frac{1 - \cos(180^\circ / N_s)}{1 + \cos(180^\circ / N_s)} = \frac{1 - \cos(180/12)}{1 + \cos(180/12)} = 0.0173$$

Here the velocity varies by nearly $\pm 2\%$ from the average. Equation (14.33) is used to find the speed of the driven sprocket's speed:

$$n_2 = n_1 \frac{N_1}{N_2} = 2400 \times \frac{12}{24} = 1200\,\text{rpm}$$

Equations (14.34)–(14.36) give us the following possible fundamental frequencies:

$$f_1 = \frac{n_1}{60} = \frac{2400}{60} = 40\,\text{Hz}$$
$$f_2 = \frac{n_2}{60} = \frac{1200}{60} = 20\,\text{Hz}$$

$$f_{CL} = \frac{n_1 N_1}{30 N_L} = \frac{2400 \times 12}{30 \times 40} = 24\,\text{Hz}$$
$$f_{TE/CA} = \frac{n_1 N_1}{30} = \frac{2400 \times 12}{30} = 960\,\text{Hz}$$

Belt Drives

Belt drives are often used instead of gears or chains in order to save on costs and provide some degree of noise control. The elasticity of the belt prevents shock loads at a driven machine from being transmitted back to the driver. Solid-borne noise and attendant vibration are reduced. Flat belts and V-belts mounted on smooth pulleys depend on friction to transmit power, so there must be adequate belt tension to prevent slipping as well as to oppose centrifugal effects. When suddenly loaded, these belts can slip, causing a squealing noise. This sometimes can be fixed by increasing the tension; but excessive tension can shorten the life of a belt and induce excessive bending moments in connected shafts. When precise speed ratios must be sustained, such as the camshaft of an automotive engine, toothed belts mounted on toothed pulleys can be utilized. Toothed belts can maintain timing and phase relationships just as well as meshing gears, but these belts isolate vibration and shock forces rather than transmitting them between the driving and driven elements.

Universal Joints

Universal joints are used when the relative position of a driving element changes with respect to a driven element, such as with an automobile transmission that is connected through a drive shaft to a rear axle. Flexible couplings and flexible shafts can also be used, but the former can accommodate only relatively slight misalignments, and the latter can handle large misalignments but cannot handle large amounts of torque. In general, for a given rotational speed n, the frequency associated with noise and vibration is given by

$$f = \frac{n}{30}\,\text{Hz}$$

14.12 Gas-Jet Noise

A most common and also worrisome noise source is the gas jet. This is also referred to as aerodynamic noise, and examples include blowdown nozzles, gas or oil burners, steam valves, pneumatic control discharge vents, aviation jet engines, and so on. An acoustically unmitigated steam valve of a large cooker in a major food processing plant can measure as much as 120 dB. Research on aerodynamically generated noise began in earnest in the early 1950s as the result of the appearance of the commercial jet engine, when it became obvious that its mechanism of sound generation had to be understood better in order to effect noise control (Lighthill, 1952, 1978; Hubbard, 1995).

Prior to Lighthill's pioneering work, there have been even earlier studies made on aerodynamic noise in conjunction with efforts to reduce noise in axial fans. We can go back even further to find that the effect of jet streaming was mentioned in the earliest recorded references to sound. When the wind blew past the pillars of the ancient Greek temples, eerie discrete tones were produced. The Greeks adjudged these tones to be the voice of Aeolus, the god of the wind, and hence these tones are called *Aeolian tones*.

The mechanism of the Aeolian sound can be explained by visualizing the flow of air over a cylinder. At a given velocity, the downsteam flow behind the cylinder exhibits an oscillatory pattern, as vortices are shed alternatively on one side of the cylinder and then the other. The ensuing trail of eddies form what is referred to as a *von Kàrmàn vortex street*, which contains strong periodic components, resulting in a sound of nearly pure tonal quality. On the basis of empirical data, the frequency of the Aeolian tone can be predicted from

$$f_{\text{Aeolian}} = \frac{\alpha\, v}{d}$$

where

v = velocity of air, m/s
d = diameter of cylinder, m
α = Strouhal number, approximately 0.2

Figure 14.11 displays the simplest example of a gas jet, in which the high velocity airflow is emanating from a reservoir through a nozzle. The gas accelerates from virtually zero velocity in the reservoir to a peak velocity in the core at the exit of the nozzle. The peak velocity of the jet depends greatly on the pressure difference between the reservoir pressure p_r and external (ambient) pressure p_a. As the pressure ratio increases, the velocity of the gas at the discharge increases up to a point when the pressure ratio of 1.89 (for the case of the gas being air) is reached. Once the flow velocity reaches the velocity of sound, any further increase of the reservoir pressure will not result in an increase of the velocity at the end of the straight duct. When the critical pressure ratio of 1.89 is reached, the nozzle is

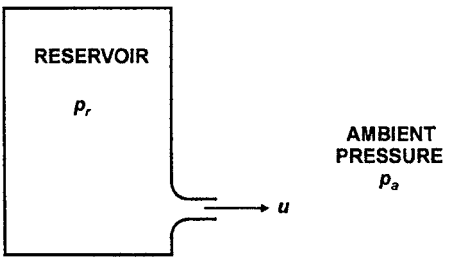


FIGURE 14.11. A simple gas jet.

said to be *choked.* In order to increase the velocity further, the nozzle must increase in its cross-sectional area beyond the duct station where choking occurs.

In the frictionless, isentropic flow of an ideal gas from one point to another, the applicable energy equation describing this flow is

$$e + \frac{p}{\rho} + \frac{u^2}{2} = \text{constant} \quad \text{(W/kg)} \tag{14.37}$$

where e is the internal energy of the gas, and u is the gas flow velocity. For the reservoir, $u = 0$ and $\mathrm{p} = p_r$. Also, from thermodynamics theory for an ideal gas,

$$e + \frac{p}{\rho} = \text{enthalpy} = c_p T$$

where c_p is the specific heat of the gas at constant pressure and T is the absolute temperature. Mach number M is defined by

$$M = \frac{u}{c} \quad \text{(nondimensional characteristic)}$$

But propagation speed of sound c in an ideal gas is given by

$$c^2 = \gamma\ RT = \gamma \frac{p}{\rho}$$

Equation (14.37) becomes

$$\begin{aligned} e + \frac{p}{\rho} + \frac{u^2}{2} &= c_p T + \frac{\gamma\ M^2 RT}{2} = c_p T \left(1 + \frac{\gamma\ M^2 R}{2c_p}\right) \\ &= c_p \frac{p}{R\rho} \left(1 + \frac{\gamma\ M^2(\gamma - 1)}{2\gamma}\right) = \text{constant} \end{aligned} \tag{14.38}$$

We apply the fact that $R = c_p - c_v$ in an ideal gas, and the ratio of the specific heats $c_p/c_v = \gamma$. Selecting two flow stations, one at a point r inside the reservoir and the other at the point in the duct where choking occurs (Mach number $M = 1.0$), we obtain from Equation (14.37)

$$\frac{p_r}{\rho_r} = \frac{p_{ch}}{\rho_{ch}} \left(1 + \frac{\gamma - 1}{2}\right) \quad \text{or} \quad \frac{p_r}{p_{cr}} = \frac{\rho_{ch}}{\rho_{ch}} \left(1 + \frac{\gamma - 1}{2}\right) \tag{14.39}$$

In an isentropic flow, according to thermodynamic theory:

$$p\rho^{-\gamma} = \text{constant}$$

Then Equation (14.37) becomes

$$\frac{p_r}{p_{ch}} = \left(1 + \frac{\gamma - 1}{2}\right)^{\frac{\gamma}{\gamma - 1}} \tag{14.40}$$

The ratio γ of specific heats is equal to 1.4 for air, so Equation (14.40) yields the critical pressure ratio for a shock to appear:

$$\frac{p_r}{p_{ch}} = \left(1 + \frac{1.4 - 1}{2}\right)^{\frac{1.4}{1.4 - 1}} = (1.2)^{3.5} = 1.89$$

Noise is generated from gas jets through the creation of fluctuating pressures from the turbulence and shearing stresses, as the high-velocity gas impacts with the ambient gas. In establishing the theory of aeroacoustics, the nonlinear effects of the momentum flux $\rho u_i u_j$ (i.e., the rate of transport of any momentum ρu_i across a unit area by any velocity component u_j) cannot be neglected as they were for linear acoustics. The momentum flux acts as a stress, since the rate of change of momentum constitutes a force. This momentum flux $\rho u_i u_j$ generates sound as a distribution of time-varying stresses. The forces between the airflow and its boundary radiate sound as distributed dipoles, and the stresses (which act on fluid elements with equal and opposite dipole-type forces) radiate as distributed quadrupoles.

The nature of the noise from jets cannot be accurately predicted, owing to the complex nature of the jet itself and the uncertainties associated with turbulence, nozzle configuration, temperature vacillations, and so on. However, first-order estimates can be derived from empirical data obtained for the most part from experimentation in the aviation industry. The earliest measurements of jet noise demonstrated that intensity and noise power varied very closely with the eighth power of the jet exit velocity (Lighthill's eighth power law), and it is now generally agreed that the overall sound power P can be expressed as

$$P = \frac{K\rho_0 U^8 D^2}{c_0^5} \tag{14.41}$$

where K is a constant, with a value 3–4; D is the jet diameter (in meters); U is the jet flow velocity (m/s); ρ_0 is the density of ambient air (kg/m^3); and c_0 is the ambient speed of sound (m/s). The factor $\rho_0 U^8 D^2 c_0^{-5}$ is often called *Lighthill's parameter*. Because the kinetic power of a jet is proportional to $\frac{1}{2}\rho_0 U^2 \cdot U D^2$, the fraction of the power converted into noise is the noise-generating efficiency η,

$$\eta \propto M^5 \tag{14.42}$$

where $M = U/c_0$, the Mach number of the flow referenced to the *ambient* speed of sound.

Aerodynamic noise can be modeled as monopoles, dipoles, and quadrupoles. A jet pulse through a nozzle or discharge from HVAC ducts can be modeled as a monopole. In fans and compressors, the turbulent flow generally encounters rotor or stator blades, grids, and baffles; this type of flow can be modeled as a dipole. Quadrupole modeling applies to noise occurring from turbulent mixing in jets where there is no interaction with confining surfaces.

The velocity term U in Equation (14.41) is the fluctuating velocity which varies throughout the jet stream. Consequently, U is not easily measured nor amenable to analytical treatment. But we can consider the average velocity V and assume that the size of the energy-bearing eddies are of the same order of magnitude as the jet diameter, and the total radiated acoustic power P is proportional to the kinetic energy of the jet flow. The total radiated power is simply a fraction of the total

power discharged from the nozzle, i.e.,

$$P = \frac{\varepsilon M^5 \rho_0 V^3 A}{2} \text{ in watts} \tag{14.43}$$

where

V = average flow velocity through the nozzle, m/s
M = Mach number of the flow = V/c, dimensionless
ρ_0 = density of ambient air
A = nozzle area
ε = constant of proportionality, in the order of 10^{-4}

Equation (14.43) constitutes a first-order approximation that is applicable to many industrial situations where the average velocity of the jetstream lies in the range of $0.15c < V \leq c$. The efficiency factor εM^5 is plotted from empirical data in Figure 14.12 over a range of Mach numbers. As a reflection of the uncertainty associated with turbulence, temperature, and so forth, the efficiency factor is given as a range.

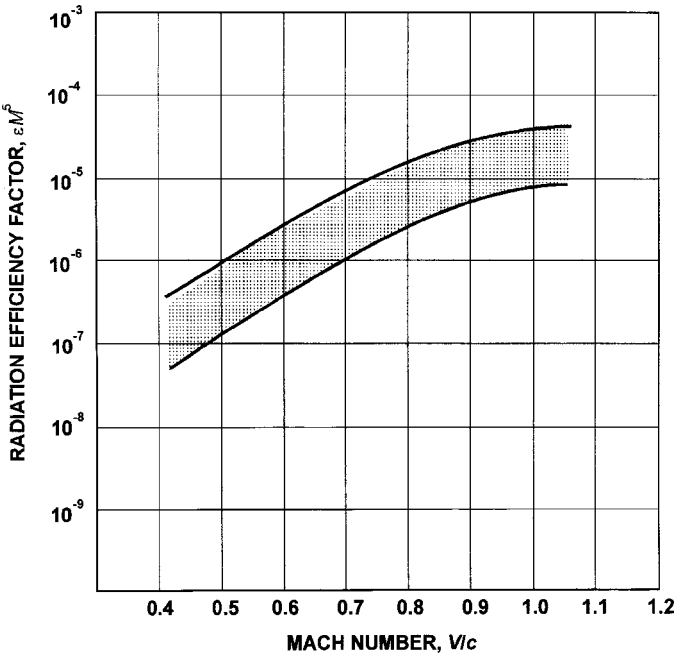


FIGURE 14.12. Efficiency factor εM^5 versus Mach number.

Example Problem 7

An air jet, operating through a choked 1-cm diameter nozzle, exhausts into the atmosphere. Determine the overall acoustical power and the sound power level L_W at 20°C.

Solution

Because the nozzle is choked, the average velocity of the air jet must be equal to the speed of sound at 344 m/s. The density of air at atmospheric pressure is 1.204 kg/m^3 is found from the ideal gas relation,

$$\rho = \frac{p_a}{RT} = \frac{1.01325 \times 10^5}{287 \times (20 + 273.2)} = 1.204\,\text{kg/m}^3$$

and the area of the nozzle is $\pi D^2/4 = \pi(10^{-2})^2/4 = 7.85\ (10)^{-5}$ m^2. From Figure 14.12, the radiation efficiency factor εM^5 is found to be approximately 3×10^{-5}. Inserting these values into Equation (14.43) we find

$$P = \frac{\varepsilon\, M^5 \rho_0\, V^3 A}{2} = \frac{3.0 \times 10^{-5} \times 1.204 \times 344^3 \times 7.85 \times (10)^{-5}}{2} = 0.0577\ \text{W}$$

thus giving us the sound power level,

$$L_W = 10 \log \left(\frac{0.0577}{10^{-12}} \right) = 107.6\,\text{dB} \tag{14.44}$$

From this last example, the acoustical power can be calculated to the first order on the basis of the nozzle diameters and exit velocities. However, the acoustical power of the air jet may have an accuracy of ±5 dB or thereabouts. But what if the gas jet is hot and extremely turbulent, as in the case of a gas burner? This situation can be resolved by applying a first-order correction for the temperature:

$$\text{Correction due to temperature} = \Delta_T = 20 \log \left(\frac{T}{T_a} \right)$$

where T and T_a are the absolute temperatures of the gas jet and the ambient air, respectively.

Example Problem 8

Consider the nozzle described in Problem Example 7 above. What would be the total radiated acoustical power level L_W if the temperature of the jet is to be raised from 20°C to 260°C?

Solution

Using Equation (14.44),

$$\Delta_T = 20\log\left(\frac{T}{T_a}\right) = 20\log\left(\frac{(260+273.2)^\circ\text{K}}{(20+273.2)^\circ\text{K}}\right) = 5.2\,\text{dB}$$

The temperature-corrected sound power level becomes

$$L_W = 107.6 + 5.2 = 112.8\,\text{dB}$$

When the overall power of a jet is determined, the overall sound pressure at any location in region surrounding the jet can be estimated through the use of the relationship between L_p and L_W:

$$L_p = L_W + 10\log\left(\frac{Q(\theta,\phi)}{4\pi r^2}\right) = L_W + 10\log\left[Q(\theta,\phi)\right] - 20\log r - 11 \tag{14.45}$$

where $Q(\theta,\phi)$ is the directivity in three-dimensional space, and r is the distance from the source of L_W to the location where the value of L_p is desired. The directivity index DI is given by

$$DI = 10\log\left(\frac{Q(\theta,\phi)}{4\pi r^2}\right)$$

In many cases of interest, the jet can be generally regarded as a point source with typical directionality as shown in Figure 14.13. The parameter ϕ can be disregarded in axisymmetric flows.

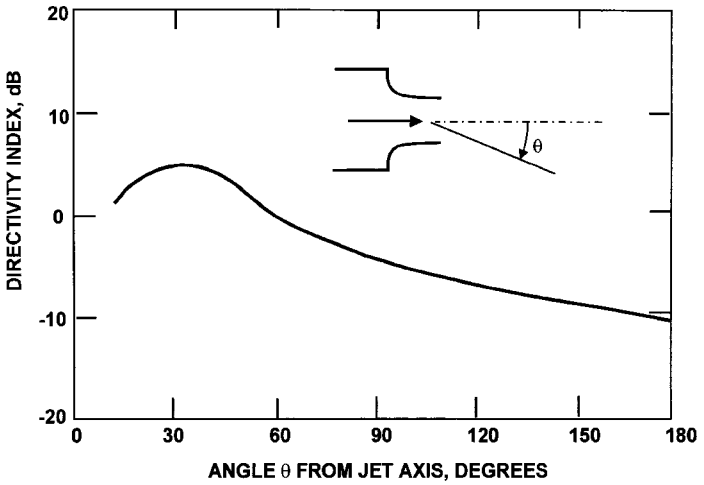


FIGURE 14.13. A fairly typical configuration for directivity DI_Q in a small subsonic jet. Note that the peak levels occur approximately in the angular range of 15° to 45° from the central axis of the jet.

Figure 14.13 illustrates that peak levels occur in the range of 15°–45° from the axis of the jet. The ordinate of Figure 14.13 giving the relative sound pressure level constitutes a directivity index DI_θ.

Example Problem 9

Find the radiated overall sound pressure level at a radial distance of 15 m from the nozzle of Problem Example 7 at angular positions of 0°, 45°, 90°, and 180°.

Solution

We make use of Equation (14.45) and the results of Problem Example 7. The sound pressure level at $r = 15$ m is

$$\begin{aligned} L_p &= L_W + 10\log[Q(\theta,\phi)] - 20\log r - 11 \\ &= 107.6 - 20\log(15) - 11 + DI_\theta = 73.1 + DI_\theta \text{ dB} \end{aligned}$$

From Figure 14.13, the $DI_\theta = 0$ at $\theta = 0°$, and $L_p = 73.1 + 0 = 73.1$ dB. At 45°, $DI_{45°}$ is approximately +3 dB, and $L_p(45°) = 73.1 + 3 = 74.1$ dB. Similarly for the other angles, we have

$$\begin{aligned} DB_{90°} &= -5 \text{ dB}, \qquad L_p(90)° = 73.1 - 5 = 68.1 \text{ dB} \\ DI_{180°} &= -10 \text{ dB}, \qquad L_P(180°) = 73.1 - 10 = 63.1 \text{ dB} \end{aligned}$$

It becomes apparent from the above example that the jet has a strong directional character that must be accounted for in determining the sound level pressures.

Gas jets also manifest a strong frequency dependence. A first-order estimate for the peak frequency can be obtained for a given power level from the empirical relation

$$f_{\text{peak}} = \frac{St \times V}{D} \tag{14.46}$$

where

St = Stouhal number, a constant = 0.15 for a wide variety of nozzle diameters and operating conditions
V = average exit velocity of the nozzle, m/s
D = nozzle diameter, m

Example Problem 10

Given a 5-cm diameter nozzle with an exit velocity of 344 m/s, determine the peak frequency of the air blowdown.

Solution

Applying Equation (14.46)

$$f_{\text{peak}} = \frac{0.15 \times 344 \text{ m/s}}{0.05 \text{ m}} = 1032 \text{ Hz}$$

Thus, the peak frequency should be expected to occur in the 1 kHz octave band.

On either side of the peak frequency, the octave band distribution of the acoustical power falls off. Experimental data indicated that the spectrum falls off at the average rate of –6 dB per octave above the peak frequency and −7 dB per octave below the peak frequency. The upshot is that the magnitude and the spectral character of gas jet noise can be only estimated roughly.

14.13 Gas Jet Noise Control

The major challenge in dealing with jet engine noise is to reduce the high noise levels with minimal impact on the thrust. The greatest progress came about by reducing the jet velocity while keeping the thrust constant because the sound power is proportion to (thrust) $\times$ $U^6/{c_0}^5$, according to the Lighthill equation (14.41). Early efforts were centered on modifying the nozzle shape to variations of the "cookie-cutter" forms or using multiple smaller nozzles. Some of these designs produced up to 10 dB noise reductions with a rather small loss of thrust. Newer by-pass and fan-jet engine designs entail much lower jet velocities, large streamlined enter bodies, and annular jets that may be subdivided in any one of several ways.

In the case of simple high-velocity air jets in industrial environments, such as those used to power air tools, provide cooling or venting, parts ejection, and so on, a number of straightforward noise reduction measures can be applied. The basic steps include the following:

1. Reduction of the required air velocity by moving the nozzle closer to a part being ejected, while maintaining the same value of thrust.
2. Adding additional nozzles, reducing the required velocity but again sustaining the same thrust magnitude.
3. Installation of newer models of quieter diffusers and air shroud nozzles.
4. Interruption of airflow in sequence with ejection or blow-off timing.

Methods 1 and 2 above result in noise reduction from cutting down on the jet velocity. Reduction of the airstream velocity should be the first consideration of any noise reduction program. Usually, the only constraint is the preservation of the air-jet thrust. The thrust T_j of a jet is given by

$$T_j = \dot{m}\, V \text{ (N)} \tag{14.47}$$

where

$$\dot{m} = \text{mass flow rate, kg/s}$$

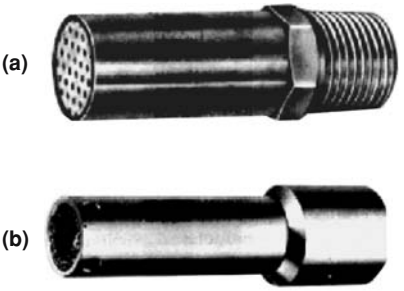


FIGURE 14.14. (a) Multi-jet diffuser and (b) restrictive diffuser nozzle.

$$V = \text{average jet velocity}$$

Equation (14.47) indicates that thrust can be preserved if the mass flow is increased while reducing the jet velocity. Increasing the nozzle exit area and moving the nozzle closer to the ejection target will also provide considerable velocity reduction. Adding on two or more nozzles will also permit reduction in air-jet velocity and a corresponding reduction in noise level.

Multiple-Jet and Restrictive Flow Silencer Nozzle

Figure 14.14(a) shows an example of a multi-jet diffuser. The noise reduction accrues mainly from the reduction in jet air core size, which also lessens the turbulence in the mixing regions. Also, the smaller inner jets flow along with the outer layer of high velocity air, thereby reducing the shearing action with the static ambient air. In the restrictive flow nozzle of Figure 14.14(b), the high-velocity core is minimized by the sintered metal restrictor, typically mounted into the nozzle exit. The flow velocity is reduced somewhat, with a corresponding drop in the amount of noise radiation. In both of these nozzle types, there will be some loss in the jet thrust, so additional nozzles may be required to keep up the same amount of the jet thrust.

Air Shroud Silencer Nozzles

In Figure 14.15, an air shroud silencer nozzle is shown with its airflow pattern. Here the noise reduction is achieved through bypassing some of the primary airflow around and over the nozzle. The bypassed air lowers the velocity gradients between the primary jet and the ambient air, thus cutting down on the shearing action and the resultant radiation of noise. There is usually little change in the mass flow for this type of silencer and the jet thrust is generally preserved. A micrometer dial provides control of amount of bypassed air. The typical noise reduction through the use of air-shroud silencer nozzles is 10–20 dB in the critical high-frequency range of 2–8 kHz for small high-velocity jets.

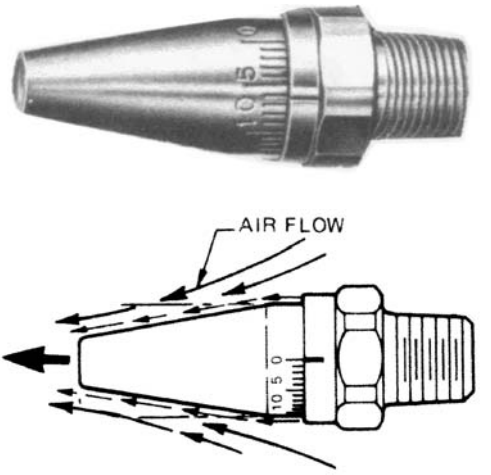


FIGURE 14.15. Air shroud silencer. The micrometer shown here adjusts the amount of air supply bypassing the nozzle instead of going through it. (Courtesy of ITW Vortec, Cincinnati, Ohio.).

Impingement Noise

Impingement noise occurs when a gas jet is brought close and impinges upon a solid surface or object. A sharp increase in the noise level occurs, particularly in the range of higher frequencies. In addition, the impingement of the gas jet on the surface can bring about unsteady forces in the form of aerodynamic dipoles, which can be described as a pair of point sources of equal magnitudes separated by a small distance and oscillating with an angular phase difference of 180°. This is to say that these two point sources are out of phase, i.e., when one of the point sources is positive, the other is negative. These dipoles constitute the basic mathematical models used to describe the noise and radiation of many common noise sources including propellers, valves, loudspeakers, air duct diffusers, a number of musical instruments, and so on. There are also some directivity patterns present when the noise evolves from dipole sources.

Impingement noise, as with jet noise, is difficult to predict in the way of its amplitude and spectral characteristics of the noise-generating mechanism. From both analytical and experimental considerations, the radiated sound power for impinging subsonic jets depends in the first order on the fifth or sixth power of the flow velocity. Again, as with the free jet flow, even a slight reduction in the flow velocity can bring about appreciable reductions in impingement noise. If a jet flows over a sharp edge or discontinuity, even more noise is likely to be generated. Whistle-like edge tones will also occur. Cutting down on the flow turbulence created by jet flows over a cavity or an obstruction can lessen the periodic components and the impingement noise. In these cases, the impingement noise can be lessened by avoiding or eliminating the presence of cavities and by redirecting the jet away from the edge.

Gaseous Flows in Pipes or Ducts

Velocities in pipes or ducts usually occur in the subsonic range, but where there are valves or vents present to control the flow, extremely high noise levels can occur. Noise levels have been measured as high as 140 dB downstream of reduction valves in large steam pipes. In many cases, pressure drops across a control or regulator valve are sufficiently large to choke the flow at the discharge, with the result that the flow of the gas jet is sonic or almost sonic with corresponding generation of high intensity aerodynamic noise. This noise can propagate through the pipe walls into the immediate surroundings, and what is even worse, it can propagate almost unabated downstream with very little attenuation.

Because of the complexity of the noise source mechanism and the degree of uncertainty in transmission loss of the pipe and ducts, it becomes quite difficult to predict the magnitude of the aerodynamic noise. But some guidance can be derived from empirical data, which can be used to establish first-order estimates. The turbulent mixing areas downstream of the valve constitute the principal region of noise generation. But if valves are encased in thick housings, the noise levels are typically 6 to 10 dB lower. The spectral character of the noise resembles that for high-velocity gas jets, i.e., there are peak levels present in the range of 2–8 kHz. In short, it can be expected wherever high-pressure steam and gas flows are regulated or discharged through valves, noise levels exceeding 100 dB will most likely occur. The valves will have relatively thick walls, so the piping system itself, downstream of the valve, is the primary source of externally radiated noise.

Three basic approaches can be considered in reducing the noise from the control valve regions, namely, (a) revising the dynamics of the flow, (b) introducing an in-line silencer to absorb acoustic energy, and (c) increasing the transmission loss in the pipe walls.

Of the three approaches just mentioned, altering the dynamics of the flow, which means actually reducing noise reduction at the source, is probably the most preferred method. Changing the dynamics entails reducing the flow velocity via multiple stages of pressure reductions or diffusion of the primary jet. Figure 14.16 shows multiple stages of pressure reduction with the use of expansion plates. The flow velocity is lessened sequentially in each expansion chamber. These plates also act as a diffuser, reducing turbulent mixing. As much as 20 dB noise reduction have

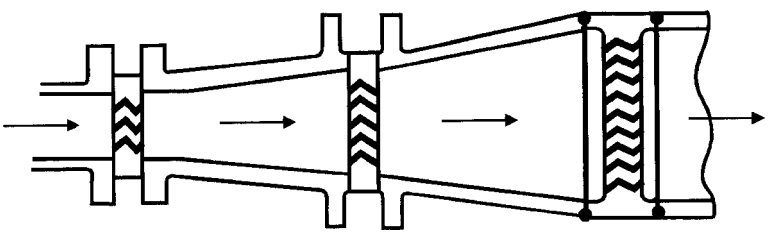


FIGURE 14.16. Multiple stages of pressure reduction in a throttling system with use of expansion plates.

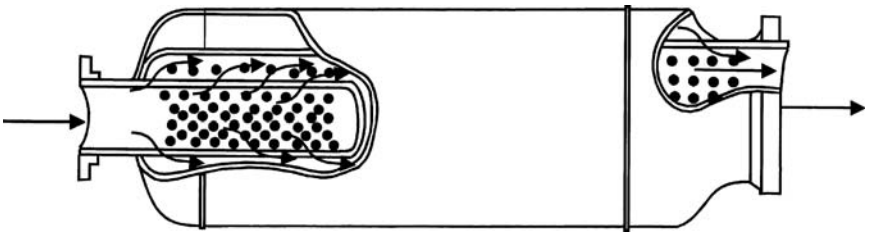


FIGURE 14.17. A cutaway view of an in-line silencer used to reduce aerodynamic noise in pipes. (Reproduced with permission of Fisher Controls International, Inc.).

been achieved. A disadvantage of this setup that a back pressure may be induced, which has the effect of impeding flows. The use of diffusers is another approach, in which the flow is diffused into smaller interacting jets.

In-line silencers, one of which is shown in Figure 14.17, basically consist of flow through ducts surrounded by absorptive materials separated from the flow by perforated metal sheets. A diffusive inlet may precede the tubular portion of the absorptive liner. The absorption materials typically consist of fiberglass or metal wool.

Finally, in the third approach, that of increasing the transmission loss in pipe walls, two methods can be used. One is to increase the pipe wall thickness and the other is to swath the pipe in acoustical absorption materials. However, no simple procedure exists for estimating the transmission loss through pipe walls. Piping standards are promulgated by mechanical engineering societies, which specify wall thicknesses for high-pressure, high-velocity flow installations. These thicknesses are specified with the principal aim of preventing ruptures and other failures of the pipes that are subject to high pressures. For larger pipes, the standard wall thickness is approximately 3/8 inch. A wall thickness greater than that required to meet stress requirements can be selected so that more sound attenuation can be achieved from the presence of the thicker walls, in the range of 2–20 dB additional transmission loss.

However, a resonance-like condition can occur at the *ring frequency*, at which value the transmission loss virtually disappears. This is somewhat analogous to the coincidence frequency effect associated with barriers. This dip in transmission loss occurs in a pipe when a single wavelength of sound becomes equal to the nominal circumference of the pipe wall, a situation the ring frequency f_r can be mathematically expressed as

$$f_r = \frac{c_w}{\pi\, D_p}$$

where c_w is the longitudinal speed of sound in the pipe wall material in m/s; and D_p the nominal diameter of the pipe (which can be considered the average of the inside and outside diameters of the pipe) in meters. For example, a 25-cm steel pipe would have a ring frequency of 6598 Hz (since $c_w = 5182$ m/s for steel). It would then be expected that the peaks noise levels radiated from this pipe would

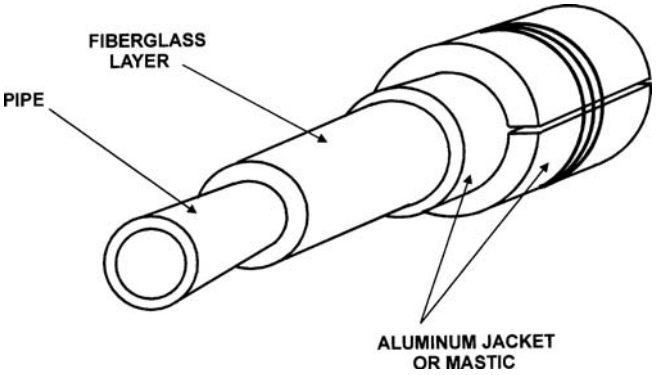


FIGURE 14.18. Lagging or jacketing of a pipe to minimize flow noise transmission.

occur in the vicinity of 6.6 kHz. Higher order resonance-like conditions will also occur at $2f_r$, $3f_r$, etc., but would be of little concern because those frequencies are extremely high.

Besides increasing wall thicknesses, another method for achieving noise attenuation in pipes is that of wrapping or *lagging* the pipes. A typical lagging setup is shown in Figure 14.18. A 2.5–8 cm layer of acoustically absorbent material (fiberglass, mineral wools, or polyurethane foam) is wrapped around the pipe wall. This absorbing layer, in turn, is sheathed in sheet metal, dense vinyl, or sheet lead. The outer dense layer is extremely important in providing a high level of noise reduction. Applying an even denser outer layer and adding additional composite layers can result in even more noise reduction. The principal disadvantage of using lagging as the only noise reduction measure is that long lengths of piping would need treatment, so it would be advisable to give priority to noise source reduction (i.e., utilization of multiple pressure reduction stages or diffusers) and to use lagging as a secondary measure to bring the overall sound levels down to acceptable values.

14.14 Mufflers and Silencers

In the preceding section we have discussed some of the elements of silencing air-jet flow. In this section we examine further aspects of reducing the noise from air-jet flow. The term *muffler* is commonly applied to the exhaust gas silencer for an internal combustion engine, and *silencer* usually denotes the noise suppressor installed in a duct or air intake. We use these terms interchangeably in this section since the same operational principle applies to both of them. Mufflers and silencers were developed to reduce noise energy while facilitating gas flow.

Both silencers and mufflers fall into one or the other of two categories: *dissipative* units and *reactive* units. Dissipative mufflers and silencers reduce noise energy by employing sound-absorption materials that are flow resistive at frequencies in the audio range. Reactive mufflers and silencers reduce noise through destructive

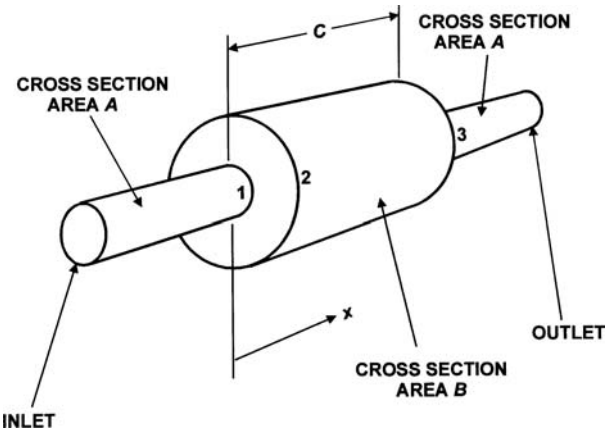


FIGURE 14.19. Expansion chamber of a reactive muffler.

interference. Both reactive and dissipative principles may be combined in a single muffler or silencers in order to ensure effectiveness over a broad frequency range.

Reactive Mufflers and Silencers

Consider the basic reactive muffler in Figure 14.19. Sound waves transmit from left to right in the inlet pipe and reflections occur in the expansion chamber, causing destructive interference under the appropriate conditions. In this analysis (Davis et al., 1984), the following assumptions are made:

1. Sound pressure is small in comparison with absolute pressure in the expansion chamber.
2. No reflected waves occur in the tailpipe (i.e., the outlet pipe).
3. The expansion chamber walls do not transmit nor conduct sound.
4. Only plane waves exist.
5. Viscosity effects are negligible.

We denote the following subscripts in the description of incident and reflected waves: I for incident waves and R for reflected waves. The particle displacements ξ of the incident and reflected waves are described in complex format as follows:

$$\xi_I = A_I e^{i(\omega t - kx)} \tag{14.48}$$

and

$$\xi_R = A_R e^{i(\omega t + kx)} \tag{14.49}$$

Here A_I and A_R are complex amplitudes. Particle velocities are obtained by differentiating equations (14.48) and (14.49) with respect to time:

$$u_I = i\omega A_I e^{i(\omega t - kx)} \tag{14.50}$$

and

$$u_R = i\omega A_R e^{i(\omega t + kx)} \tag{14.51}$$

We recall that for a plane wave in the direct field, sound pressure is related to particle velocity by

$$p = \rho c u$$

which is used in Equations (14.50) and (14.51) to obtain

$$p_I = i\rho c\omega A_I e^{i(\omega t - kx)} \tag{14.52}$$

$$p_R = i\rho c\omega A_R e^{i(\omega t + kx)} \tag{14.53}$$

Let $x = 0$ designate the junction of the inlet pipe and expansion chamber. Pressure must be continuous at this junction, hence

$$(p_{I1} + p_{R1})_{x=0} = (p_{I2} + p_{R2})_{x=0} \tag{14.54}$$

Inserting Equations (14.52) and (14.53) into (14.54) gives

$$A_{I1} + A_{R1} = A_{I2} + A_{R2} \tag{14.55}$$

Subscript 1 refers to the left of the junction and subscript 2 refers to the right. Continuity of flow requires that

$$A\,(u_{I1} - u_{R2})_{x=0} = B\,(u_{I2} - u_{R2})_{x=0}$$

Setting $B/A = m$, where A refers to the flow area of the inlet pipe and the outlet pipe and B the flow area of the expansion chamber, we have from continuity

$$A_{I1} - A_{R1} = m(A_{I2} - A_{R2}) \tag{14.56}$$

In Figure 14.19, C denotes the length of the expansion chamber. At the junction of the expansion chamber where $x = C$, pressure and flow continuity necessitates that

$$(p_{I2} + p_{R2})_{x=C} = p_3$$

from which

$$A_{I2}e^{-ikC} + A_{R2}e^{ikC} = A_3 \tag{14.57}$$

and

$$B(u_{I2} - u_{R2})_{x=C} = A\,u_3 \tag{14.58}$$

resulting in

$$m(A_{I2}e^{-ikC} + A_{R2}e^{ikC}) = A_3 \tag{14.59}$$

We now have four Equations (14.55)–(14.59) and five unknowns, A_{I1}, A_{R1}, A_{I2}, A_{R2}, and A_3. These equations are solved simultaneously to correlate conditions at

the inlet and outlet of the expansion chamber, resulting in the complex ratio

$$\frac{A_{I1}}{A_3} = \frac{(1+m)\left(1+\frac{1}{m}\right)e^{ikC} + (m-1)\left(\frac{1}{m}-1\right)e^{-ikC}}{4}$$
$$= \cos(kC) + \frac{i}{2}\left(m+\frac{1}{m}\right)\sin(kC) \tag{14.60}$$

In order to establish the transmission loss, the ratio of sound intensity at the expansion chamber inlet to transmitted sound intensity is needed:

$$\frac{I_{I1}}{I_3} = \frac{p^2_{\text{rms}(I1)}}{p^2_{\text{rms}(3)}} = \left|\frac{A^2_{I1}}{A^2_3}\right| = \cos^2(kC) + \frac{1}{4}\left(m+\frac{1}{m}\right)^2\sin^2(kC)$$
$$= 1 + \frac{1}{4}\left(m-\frac{1}{m}\right)^2\sin^2(kC) \tag{14.61}$$

Because transmission loss is a function of the ratio of power incident on the muffler to the power transmitted, i.e.,

$$TL = 10\log\left(\frac{W_{\text{incident}}}{W_{\text{transmitted}}}\right)$$

and if the inlet and outlet areas are equal, we obtain

$$TL = 10\log\left(\frac{W_{I1}}{W_3}\right) = 10\log\left(\frac{I_{I1}}{I_3}\right) = 10\log\left[1+\frac{1}{4}\left(m-\frac{1}{m}\right)^2\sin^2(kC)\right] \tag{14.62}$$

where $k = 2\pi/\lambda$ is the wave number. Equation (14.62) becomes invalid if any of the lateral dimensions of the expansion chamber exceeds 0.8λ.

Figure 14.20 displays a plot of the theoretical transmission loss versus C/λ for various area ratios. Equation (14.62) forecasts a transmission loss of zero when the argument of the sine function is 0, π, 2π, $.2n\pi$, ... and a maximum transmission loss when the argument is $\pi/2$, $3\pi/2$, ... $(2n+1)\,\pi/2$, ..., where $n = 0$, 1, 2, 3, etc. It then follows that the expansion chamber works best when length C constitutes an odd number of quarter-wavelengths, i.e., $C = \lambda(2n+1)/4$ for maximal TL. But the expansion chamber becomes ineffective when the chamber length C measures out to an integer number of half-wavelengths, i.e., $C = n\lambda$ (i.e., $\lambda/2$, λ, $3\lambda/2$, etc.). For extremely low frequencies C/λ approaches zero, and the transmission loss likewise approaches zero.

Example Problem 11

A reactive muffler is to be designed to produce a transmission loss of 20 dB at 150 Hz. The inlet and outlet pipes are 60 mm in diameter and the average gas (air) temperature is 100°C. Determine the size of the expansion chamber.

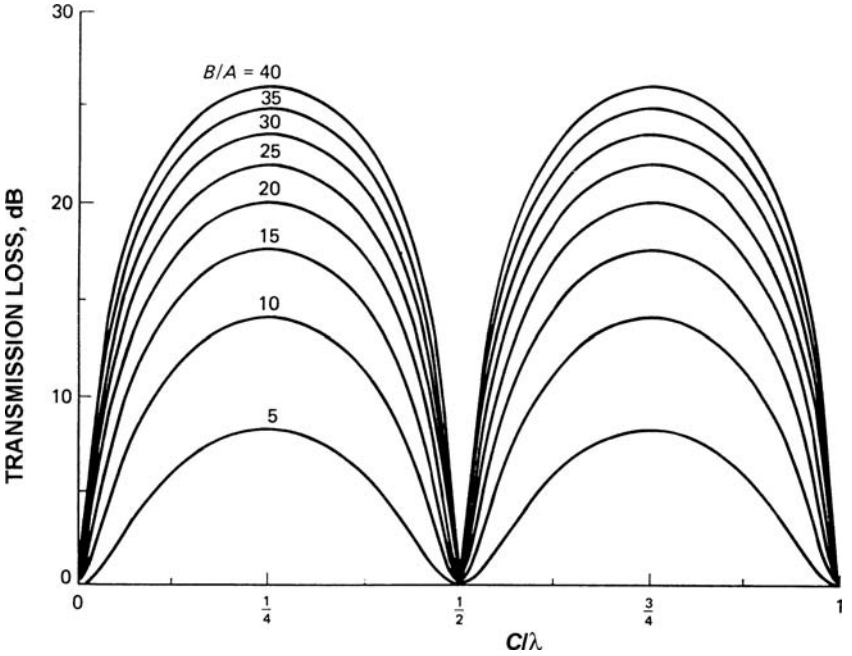


FIGURE 14.20. Transmission loss TL versus C/λ.

Solution

At 100°C, the velocity $c = 20.04\sqrt{T + 273.16} = 387.1$ m/s. The wavelength of the 150 Hz sound is given by $\lambda = c/f = 387.1/150 = 2.58$ m. For the shortest expansion chamber to produce maximum transmission loss, we examine the argument of the sine function:

$$kC = \frac{2\pi C}{\lambda} = \frac{\pi}{2} \text{ radians}$$

from which we get

$$C = \frac{\lambda}{4} = \frac{2.58}{4} = 0.645\,\text{m} = 645\,\text{mm}$$

From Equation (14.62) the transmission loss is given by

$$\begin{aligned} TL &= 10\log\left[1 + \frac{1}{4}\left(m - \frac{1}{m}\right)^2 \sin^2(kC)\right] \\ &= 10\log\left[1 + \frac{1}{4}\left(m - \frac{1}{m}\right)^2 \sin^2\left(\frac{2\pi \times 0.645}{2.58}\right)\right] \\ &= 10\log\left[1 + \frac{1}{4}\left(m - \frac{1}{m}\right)^2\right] \end{aligned} \tag{14.63}$$

Let

$$\left(m - \frac{1}{m}\right)^2 = \Omega$$

Then

$$\frac{A}{B}\left[\left(\frac{B}{A}\right)^2 - \frac{\Omega B}{A} - 1\right] = 0$$

where from Equation (14.63)

$$\Omega = 2(10^{TL/10} - 1)^{1/2}$$

The solution to the quadratic expression inside the bracket is

$$\frac{B}{A} = \frac{\Omega + \sqrt{\Omega^2 + 4}}{2}$$

Note the negative root carries no physical significance. For a transmission loss of 20, $\Omega = 19.90$ and

$$\frac{B}{A} = \frac{19.90 + \sqrt{19.90^2 + 4}}{2} = 19.95$$

The cross-sectional area A of the inlet pipe and the outlet pipe are each $\pi(60)^2/4 = 2827\ \text{mm}^2$. The expansion chamber cross-sectional area is

$$B = A \times \frac{B}{A} = 2827 \times 19.95 = 56{,}399\ \text{mm}^2$$

If the expansion chamber has a cylindrical cross-section, its inside diameter d is found from

$$B = \frac{\pi d^2}{4}$$

Hence

$$d = \sqrt{\frac{4B}{\pi}} = \sqrt{\frac{4 \times 56{,}399}{\pi}} = 268\ \text{mm}$$

Area B represents the minimum cross-sectional area of the expansion chamber that is required to produce the 20-dB transmission loss.

Dissipative Mufflers and Silencers

In the preceding paragraphs, dissipative silencers were discussed. Lining the ducts of HVAC systems with sound absorptive materials generally provides adequate noise attenuation. However, many industrial applications require a silencer that can provide a large amount of noise attenuation or insertion loss in a relatively small space. If the noise energy covers a narrow frequency range, reactive silencers constitute the better solution. Reactive silencers are preferred where the gas flows contain particles or other components that could contaminate sound absorbing

materials. Dissipative silencers are more effective over a wider range of frequencies and in silencing fluctuating machinery noises. In a number of situations, in order to obtain the greatest noise reduction over a wider frequency range, both reactive and dissipative principles are combined in a single silencer.

14.15 Active Noise Control

Relatively recent advances in digital electronics have made it more economically feasible to apply *active noise control* in cutting down on noise in aircraft and automotive interiors, pumps, compressors, electric motors, transformers, and so on (Piraux and Nayroles, 1980; Nelson and Elliott, 1992, 1997). Earlier active noise control techniques were first used in the study of noise in fan ducts. The advent of inexpensive digital processors enabled the conversion of analog audio frequency signals into digital form, then processing them through a digital filter and then converting back into an analog signal with very little time delay. The basis of active noise control is to duplicate the noise that is the same but 180° out of phase, so that when the offending noise and the duplicated out-of-phase noise are combined, a cancellation will occur.

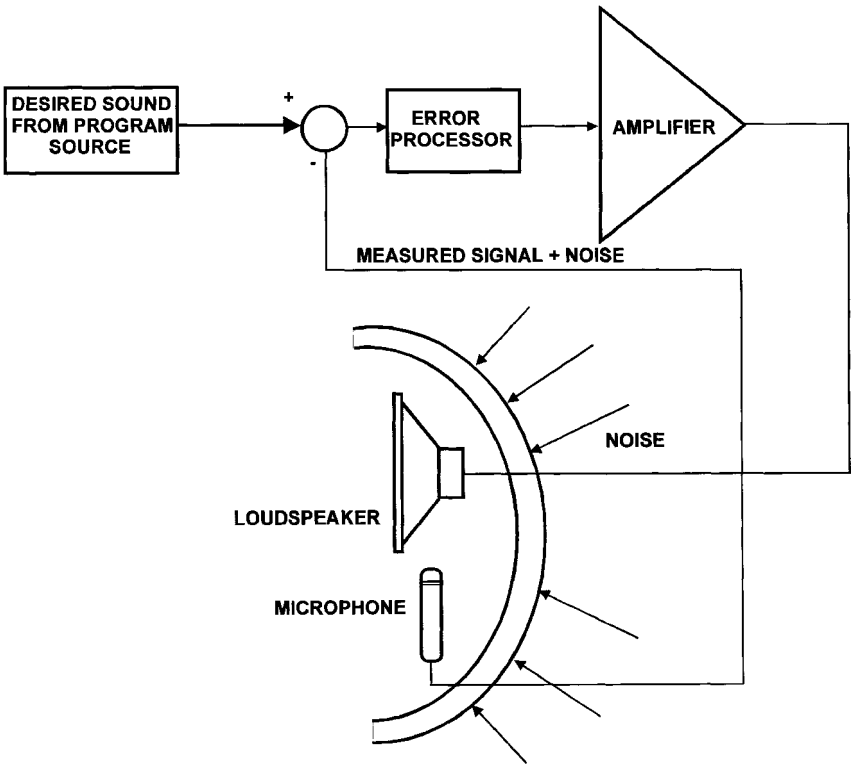


FIGURE 14.21. An example of an active noise control system for a communication headset.

Figure 14.21 illustrates a simple single-channel active noise control system. Sound is detected by a microphone and processed though a digital filter imbedded in a special-purpose microprocessor prior to being fed into a loudspeaker that radiates the sound that is intended to interfere destructively with the "original" unwanted sound. The characteristics of the digital filter are designed to minimize the time-averaged signal at the error microphone on a continuous basis.

The following factors, which may limit the effectiveness of active noise control, must be considered:

1. Continuous and reliable measurement, signal processing, and sound generation are required.
2. For maximum effectiveness the anti-noise source must be near the noise source or near the receiver.
3. When changes occur in the relative position of the noise source and the observer, the effect could be one of sound reinforcement rather than cancellation. This consideration can limit many active-noise control systems to low-frequency sounds. However, the presence of error-detection microphones combined with digital filters that continuously adapt to changing conditions can mitigate the possibility of unintentional sound reinforcement.
4. Even under the most ideal real conditions, cancellation of sounds cannot be total, due to the statistical nature of the molecular motion in the sound propagation medium.

Active noise control systems have even been incorporated into items as small as headphones for the purposes of shutting out external noises while allowing desired sound to reach the ears. Such headphones are useful in aviation cockpits, passenger compartments of aircraft, and other noisy environments.

References

AMCA Standard 300. *Reverberant Room Method for Sound Testing of Fans*. Arlington Heights, IL: Air Movement and Control Association.

ASHRAE Standard 51. *Laboratory Method of Testing Fans for Rating*. Atlanta, GA: American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc.

ASHRAE. *ASHRAE Handbook*. Atlanta, GA: American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc.

Davis, D. D., Stokes, G. M., Moore, D., and Stevens, G. L. 1984. Theoretical and experimental investigation of mufflers with comments on engine-exhaust muffler design. (Reprint, NACA Report 1192. 1954), *Noise Control*, M. J. Crocker, (ed.). New York: Van Nostrand Reinhold.

Fuller, C. R., Elliott, S. J., and Nelson, P. A. 1996. *Active Control of Vibration*. London: Academic Press.

Graham, J. Barrie and Hoover, Robert M. 1991. Fan noise. In: *Handbook of Acoustical Measurements and Noise Control*, 3rd ed. Harris, Cyril M. (ed.). New York: McGraw-Hill, Chapter 24.

Hand, R. F. 1982. Accessory noise control. In: *Noise Control in Internal Combustion Engines*, Baxa, D. E. (ed.). New York: Wiley-Interscience, pp. 437–477.

Harris, Cyril M. (ed.). 1991. *Handbook of Acoustical Measurements and Noise Control*, 3rd ed. New York: McGraw-Hill, Chapters 34–45.

Hubbard, H. H. (ed.). 1995. *Aerodynamics of Flight Vehicles: Theory and Practice*, Vol. 1: *Noise Sources*, Vol. 2: *Noise Control*. Woodbury, NY: Acoustical Society of America. (A now classic compendium of theory and experimentation in the field of aeroacoustics, originally sponsored by the Aeroacoustics Technical Committee of the American Institute of Aeronautics and Astronautics (AIAA) and reprinted by the American Institute of Physics.)

Jones, Dylan M. and Broadbent, Donald E. 1991. Human performance and noise. *Handbook of Acoustical Measurements and Noise Control*, 3rd ed. Harris, Cyril M. (ed.). New York: McGraw-Hill, Chapter 41.

Lighthill, James M. 1952. On sound generated aerodynamically: I. General theory. *Proceedings of the Royal Society* A 211:564–587. (A classic paper by the great master of modern aeroacoustics theory.)

Lighthill, James M. 1978. *Waves in Fluids.* Cambridge: Cambridge University Press. (A veritable classic.)

Magrab, E. B. 1975. *Environmental Noise Control*. New York: John Wiley and Sons.

Nelson, P. A. and Elliott, S. J. 1992. *Active Control of Sound*. London: Academic Press.

Nelson, P.A. and Elliott, S. J. 1997. Active Noise Control. In: *Encyclopedia of Acoustics*, Vol. 2. Crocker, Malcolm J. (ed.). New York: John Wiley & Sons, Chapter 84, pp. 1025–1037.

Piraux, J. and Nayroles, B. 1980. A theoretical model for active noise attenuation in three-dimensional space. *Proceedings of Inter-Noise'80*. Miami, FL: 703–708.

Webb, J. D. (ed.). 1976. *Noise Control in Industry*. Sudbury, Suffolk, Great Britain: Sound Research Laboratories Ltd.

Problems for Chapter 14

1. A 6-blade fan rotates at 1150 rpm. Determine the frequencies of the noise that will emanate from the fan.
2. Find the sound power of a fairly efficient electric motor rated at 85 hp at 2400 rpm.
3. Determine the frequency of the blade rate component of a diffuser-type compressor with 24 blades in the rotor and 36 blades in the stator. The rotor rotates at 9000 rpm.
4. A 20-hp hydraulic screw-type pump operates at 1200 rpm with 4 chamber pressure cycles per revolution. Find its fundamental frequency and the sound power output.
5. A radial forward-curved fan has 36 blades and a rotor diameter of 120 cm. It operates at 950 rpm with an airflow rate of 24 m^3/s under the effect of a total pressure of 1.8 kPa. What is the total sound power at the inlet?
6. Estimate the sound power of a "quiet" 100-hp electric motor operating at 1200 rpm.
7. A ball bearing has a stationary outer race. The inner and outer race diameters are 30 mm and 46 mm, respectively, measured at the point of ball contact. The inner race rotates at 5200 rpm. There are 12 balls in the bearing. Determine

the noise and vibration frequencies that can occur as the result of imbalance and defects.

8. A ball-bearing is constructed as follows: it has twelve 15-mm-diameter balls, and the inner race diameter is 50 mm at the point of ball contact. The inner race has a rotational speed of 1800 rpm and the outer race remains stationary. Find:
   (a) rotational speed of separator
   (b) speed of the balls relative to the separator
   (c) shaft imbalance frequency
   (d) outer race defect frequency
   (e) inner race defect frequency
   (f) frequency arising from damage to one ball
   (g) frequency attributable to imbalance in the separator.
9. A roller bearing contains ten 16-mm-diameter rollers. The inner race diameter is 30 mm. The inner race rotates at 6000 rpm while the outer race remains stationary. Find:
   (a) rotational speed of the separator
   (b) speed of the rollers relative to the separator
   (c) shaft imbalance frequency
   (d) outer race frequency
   (e) inner race frequency
   (f) frequency caused by a defect in one roller
   (g) frequency caused by imbalance of the separator.
10. In a nonplanetary gear transmission, the input shaft gear has 40 teeth and rotates at 1200 rpm. It meshes with another shaft through a 30-tooth gear. Predict all of the fundamental frequencies that are likely to arise in this transmission.
11. The gears of a reverted gear train of Figure 14.8 carry the following specifications: gear 1 (driver), 50 teeth; gear 2, 90 teeth; gear 3, 46 teeth; and output gear 4, 47 teeth. The input shaft rotates at 2400 rpm.
   (a) Determine the rotational speeds of the other shafts.
   (b) Compute each of the fundamental tooth-error and shaft frequencies and the first three harmonics.
   (c) Find all of the fundamental tooth meshing frequencies and their first three harmonics.
   (d) Determine the sideband frequencies.
12. It is specified that a pair of spur gears is to be used to reduce shaft speed from 2400 rpm to 1600 rpm. The driver has 18 teeth and a diametral pitch of 4. The pressure angle is 20° in the stub teeth.
   (a) Predict the fundamental tooth-error frequencies.
   (b) Determine the fundamental tooth meshing frequency.
   (c) Find the contact ratio, on the basis of Equation (14.21) and discuss the results in terms of noise. Assume that $a = 0.25$.
13. Consider a chain drive system that contains a 15-tooth input sprocket that rotates at 2500 rpm. The chain has 90 links, and it is specified that the output sprocket rotates at 1500 rpm. Determine:

(a) the number of teeth needed for the output sprocket
(b) the range of output speed if input speed remains constant
(c) the probable frequencies of noise and vibration due to damage to one tooth or imbalance in either shaft
(d) the probable frequencies due to damage to one link
(e) the probable frequencies due to chordal action.

14. The input speed of a chain drive is to be reduced by one-half at the output. The 24-tooth input sprocket rotates at 1200 rpm. The chain contains 160 links.
(a) How many teeth are needed in the output sprocket?
(b) For constant input speed, determine the range of output speed.
(c) Determine the likely frequencies of noise and vibration due to imbalance of either shaft or damage to one tooth.
(d) Find the possible frequencies due to damage to one link
(e) Determine the possible frequencies due to chordal action.

15. In a Hooke-type universal joint, in which the cross-link plane is normal to the axis of the driving shaft, the speed ratio is given by $n_2/n_1 = 1/(\cos\phi)$, where ϕ is the shaft misalignment angle, with the result that the speed range due to misalignment is given by

$$\cos\phi \leq \frac{n_2}{n_1} \leq \frac{1}{\cos\phi}$$

A misalignment of 10° occurs in this type of universal joint. The input shaft rotates with a constant speed of 1200 rpm.
(a) Find the speed range of the output shaft.
(b) Determine the frequency resulting from vibration or noise.

16. A fairly simple 3/8-inch-diameter nozzle functions under an inlet pressure of 100 psi (gauge). Determine the sound level at a distance of 5 ft for an ambient air temperature of 80°F.
17. A simple 25 mm nozzle operates at an inlet pressure of 600 kPa. Determine the sound pressure level at a distance of 1.0 m at an air temperature of 20°C.
18. Fifty sound level readings are taken at 10-s intervals. Find the percent exceeded noise levels L_{10}, L_{50}, and L_{90} for the following distribution:

Level [dB(*A*) ± 0.05]	Number of Readings
87	2
86	8
85	14
84	15
83	5
82	4
81	2

15
Underwater Acoustics

15.1 Sound Propagation in Water

Sound waves are absorbed and scattered in water to a much lesser degree than electromagnetic waves. Because of this property sound waves have proven to be particularly useful in detecting distant objects undersea by means of *sonar* (acronym for SOund NAvigation and Ranging). *Passive* sonar is strictly a "listening" device that detects sound radiation emitted (sometimes unintentionally) by a target. In *active* sonar, the process entails sending out a sound pulse and listening for a returning echo. The loudness of the echo hinges principally on the amount of energy absorption in the water and the degree of reflection from an intercepting surface. Some of the energy is scattered backward in a random fashion to the echo-ranging emitter, either by particles or inhomogeneities in the water, or by the ocean surface or sea bottom. This scattering results in a phenomenon referred to as *reverberation*. The sound directly reflected back from an obstacle (*target*)—such as a submarine or a whale—constitutes the *echo*.

Recognizable echoes have been mapped for schools of fish, dolphins, whales, patches of kelp and seaweed, sunken wrecks and pronounced irregularities in shallow depths. Certain water conditions give rise to echoes at very short ranges; and ocean swells also can generate echoes. Ship wakes and other types of bubble screens make effective targets as wells as icebergs. The most prominent use of sonar has been in subsurface warfare: submarines, surface vessels, and underwater mines are obviously the most important targets.

The reflections of the probing signals from submarines to surface vessels are evaluated in terms of *target strengths*, a quantitative measure of intensities of the echoes. Target strengths (TS) depend on a number of factors, viz. target size, shape, orientation with respect to the probe signal source, distance from the source to the target, and frequency of the probe signal. The intensity of the reflected sound is also a function of the intensity of the probe signal striking the target, the distance from the target to the point of echo measurement (which is usually at virtually the same location as the probe signal source), and the acoustic absorptivity of the target. The effects of these variants can be established only if the radii of curvature of the sound waves striking the target and returning to a receiver are

of much greater magnitudes than those of the target's dimensions, i.e., the probe signal waves should ideally be plane. In terms of ray acoustics the incident sound wave must be substantially parallel in the region of the target that they strike, and reflected sound rays should also be parallel over the area of the sonar receiver.

In the military use of sonar, the applicable spectrum range covers the ultralow frequency to the megahertz region. Acoustic mines detect the pressures below 1 Hz, which are generated by moving ships. These mines detect the acoustic radiation and explode when the acoustic level reaches a certain level in their bandpass. Such mines can be destroyed harmlessly through the use of a minesweeper that is basically a powerful signal source towed behind a minesweeping vessel.

In passive detection, the acoustic radiation of both water-surface and underwater vessels, are sensed by a hydrophone array mounted on the spy vessel or submerged at the bottom a long distance away. The receiving array must be directional in order to be able to locate the target through sensing of somewhat higher frequencies.

Modern echo-ranging (active) sonar consists of an elaborate array of equipment to send out signals in the form of long, high-power pings in designated directions vertically and horizontally, and newer signal processing techniques present the echoed data to the observer. Transducer arrays are often enclosed in separate housings that are towed underwater behind the surface vessel, so that shallow thermal gradients[1] can be penetrated, and the sonar can probe in the stern (forward) direction, a procedure that cannot be achieved with a hull-mounted sonar.

Peaceful uses of sonar expanded greatly immediately after World War II. Originally developed for depth sounding, sonar is now being used to find fish, study fish migration, map ocean floors, locate underwater objects, transmit communications and telemetric data, serve as acoustic speedometers, act as position-marking beacons, and monitor well-head flow control devices for undersea oil wells. Passive sonar provides marine biology researchers a window for tracking sounds made by cetaceans (whales, dolphins, and porpoises).

15.2 Some Basic Concepts Pertaining to Underwater Sound

In an acoustic plane wave passing through a fluid medium of density ρ, the particle pressure p relates to the fluid particle velocity u as follows

$$p = \rho c u.$$

The fluidic parameter ρc is called the *specific acoustic resistance*. Its value, on the average, for seawater is 1.5×10^5 g/cm^2 s or 1.5×10^6 kg/m^2 s. In contrast, $\rho c = 42$ g/cm^2 s for air. From our previous chapters we recognize that this parameter can also assume a complex value in its role as the *specific acoustic impedance*.

[1] In the 1920s and 1930s, it was observed that good echoes were obtained with the early versions of shipboard echo-ranging equipment in the morning and poor or no results were obtained in the afternoon. This was due to the shifts in the seawater thermal gradients that caused sound to refract toward the sea bottom and thereby place a target in the "shadow zone."

In order to understand the importance of acoustic impedance, consider the fact that some sort of a piston must drive against a medium in order to generate acoustic energy. The medium presents a resistance to that drive, and in acoustic terminology, this resistance is the specific acoustic impedance. The conventional techniques applied in the design of the usual audio equipment, namely loudspeakers and microphones, are not applicable to underwater sonar. Air is a very light substance, so the driving mechanism needs to produce a large displacement with very little force. In the case of underwater sound, however, it is necessary to provide a very large force to generate even a small displacement. This means that the sonar must possess a very large mechanical impedance, i.e., a large ratio of the complex driving force to the complex velocity. It is also evident that a propagating sound wave carries mechanical energy that includes the kinetic energy of the particles in motion and the potential energy of the stresses occurring in the elastic medium. In the process of a wave propagating, a certain amount of energy per second crosses a unit area. This power per unit area describes the *intensity* of the wave. If a unit area is given an orientation with respect to reference coordinates, the intensity becomes a vector quantity represented by a *Poynting vector* normal to the unit area, in the same manner as in the theory of electromagnetic propagation. In a plane wave, the instantaneous intensity relates to the instantaneous acoustic pressure in the following way:

$$I = \frac{p^2}{\rho c}.$$

For the cases entailing transient signals, or signal distortions, or target impingement occurring, it is more useful to use the concept of *energy flux density* of the acoustic wave, as defined by

$$E = \int_0^\infty I dt = \frac{1}{\rho c} \int_0^\infty p^2 dt.$$

The unit of intensity in underwater sound is the intensity of a plane wave having a root-mean-square (rms) pressure of 1 micropascal (1 μPa) or 10^{-5} dyne/cm^2. This amounts to 0.64×10^{-22} W Ag/cm^2. This pressure of 1 μPa serves as the reference level for the definition of the decibel as applied to underwater acoustics, in contrast to the decibel referred to 20 μPa, that describes the sound level of acoustic propagation in air.

15.3 Speed of Sound in Seawater

The principal difference between the speed of sound in fresh water and seawater is that with the latter, salinity constitutes an additional factor besides pressure and temperature. In fact, the speed of sound in seawater varies with geographic location, water depth, season, and even the time of the day. Sound velocity in a natural body of water was first measured in 1827 when the Swiss scientist Jean-Daniel Colladen (1802–1893) and the French mathematician Charles Sturm (1803–1855)

TABLE 15.1. Expressions for Sound Speed (m/s) in Seawater as Functions of Temperature, Salinity, and Depth.

Expression	Limits	Reference
$c = 1492.9 + 3(T-10) - 6\times10^{-3}(T-10)^2$	$-2 \le T \le 24.5^\circ$	Leroy
$\quad - 4\times10^{-2}(T-18)^2 + 1.2(S-35)$	$30 \le S \le 42$	
$\quad - 10^{-2}(T-18)(S-35) + D/61$	$0 \le D \le 1000$	
$c = 1449.2 + 4.6T - 5.5\times10^{-2}T^2$	$0 \le T \le 35^\circ$	Medwin
$\quad + 2.9\times10^{-4}T^3 + (1.34 - 10^{-2}T)(S-35)$	$0 \le S \le 45$	
$\quad + 1.6\times10^{-2}D$	$0 \le D \le 1000$	
$c = 1448.96 + 4.591T - 5.304\times10^{-2}T^2$	$0 \le T \le 30^\circ$	MacKenzie
$\quad + 2.374\times10^{-4}T^3 + 1.340(S-35)$	$30 \le S \le 40$	
$\quad + 1.630\times10^{-2}D + 1.675\times10^{-7}D^2$	$0 \le D \le 8000$	
$\quad - 1.025\times10^{-2}T(S-35) - 7.139\times10^{-13}TD^3$		

D = depth, in meters; S = salinity, in parts per thousand; T = temperature in degrees Celsius.

collaborated in striking a submerged bell in Lake Geneva and simultaneously setting off a charge of powder in the air (Colladen and Sturm, 1827; Wood, 1941). The intervals between the two events were timed across the lake, and they obtained a value of 1435 m/s at 8.1°C, which is amazingly close to the modern value. Subsequent measurements over the past 40 decades entail direct measurement of speeds under carefully controlled conditions, and the speeds were mapped as functions of oceanographic parameters (Del Grosso, 1952; Weissler and Del Grosso, 1951; Wilson, 1960).

Expressions have been derived in terms of three basic quantities: temperature, salinity, and pressure (hence depth). Save for the presence of contaminants such as biological organisms and air bubbles, no other physical properties were found to affect the speed of sound in seawater. The interdependence of these three parameters is not a simple one. Del Grosso (1974) developed an expression containing 19 terms each to 12 significant figures in the powers and cross-products of the three principal variables, and Lovett developed a simpler but still unwieldy expression. We list in Table 15.1 a number of simpler expressions that suffice for most practical work. In general, these expressions result in values with errors less than a few parts in ten thousand or approximately 0.5 m/s.

The speed of sound in the sea increases with temperature, depth, and salinity. Table 15.2 gives the approximate coefficient for the rate of change in these principal

TABLE 15.2. Approximate Coefficients of Sound Velocity.

Parameter being varied	Coefficient	Coefficient
Temperature (near 70°F)	$\frac{\Delta c/c}{\Delta T} = +0.001/^\circ\text{F}$	$\frac{\Delta c}{\Delta T} = +5$ ft / s °F
Salinity	$\frac{\Delta c/c}{\Delta S} = +0.0008/\text{ppt}$	$\frac{\Delta c}{\Delta S} = +4$ ft/s ppt
Depth	$\frac{\Delta c/c}{\Delta D} = +3.4\times10^{-6}/\text{ft}$	$\frac{\Delta c}{\Delta D} = +0.016\,\text{s}^{-1}$

c = speed in ft/s; T = temperature in °F; S = salinity in parts per thousand (ppt); D = depth in feet.

parameters. In open deep water, salinity is found to have a rather small effect on the velocity.

15.4 Velocity Profiles in the Sea

The term *velocity profile* refers to the variation of the sound speed with depth; it is also called the *velocity–depth function*. Figure 15.1 shows a typical deep-sea velocity profile, which, in turn, can be subdivided into several layers. The *surface layer* lies just below the sea surface. The speed of sound in that layer is responsive to daily and local changes of heating, cooling, and action of the winds. The surface layer may consist of a mixed layer of isothermal water that is caused by action of the wind as it blows across the surface of the water. Sound becomes trapped in this mixed layer. On prolonged calm and sunny days, this mixed layer dissipates, to be replaced by water in which its temperature drops with increasing depth.

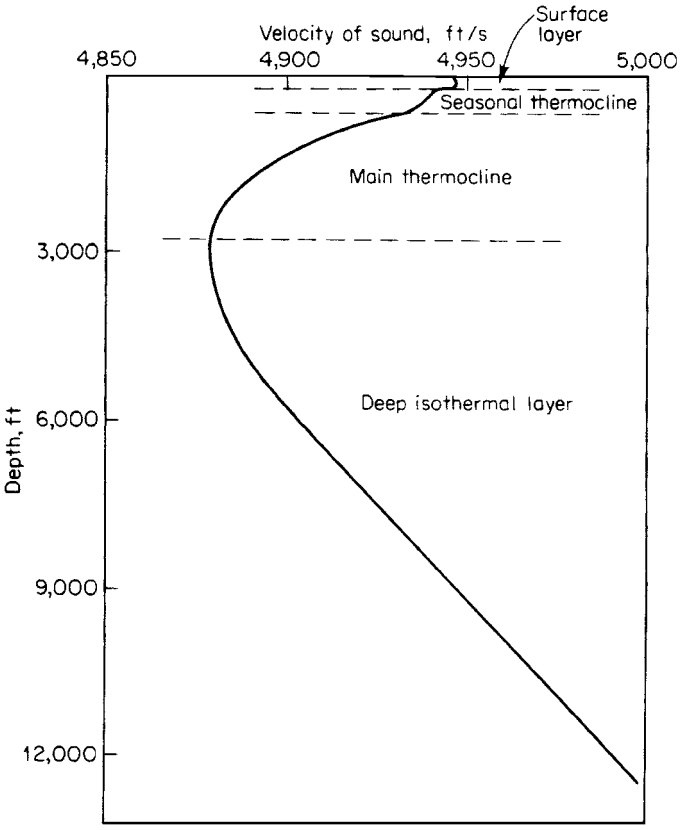


FIGURE 15.1. Deep-sea sound propagation velocity profile subdivided into principal layers.

The *seasonal thermocline* lies below the surface layer. The term *thermocline* denotes a layer in which the temperature varies with depth. The seasonal thermocline is usually characterized by a negative thermal or velocity gradient, meaning that the temperature and the speed of sound decreases with increasing depth, and it does vary with the seasons. During the summer and fall, when the ocean waters near the surface are warm, the seasonal thermocline is well defined and it becomes less so during the winter and spring and in the Arctic when it tends to become indistinguishable from the surface layer. Below the seasonal thermocline lies the *main thermocline*, which hardly varies throughout the seasons. It is this layer in the deep sea, that the temperature changes the most. Underneath the main thermocline, reaching down to the sea bottom is the deep isothermal layer having an almost constant temperature (generally about 3°–4°C) in which the speed of sound *increases* with depth because of the effect of pressure on the sound speed. At the saddle point between the negative gradient of the main thermocline and the positive gradient of the deep layer, there occurs a velocity minimum toward which sound traveling at great depth tends to bend or becomes focused by refraction. In the more northern regions, the deep isothermal layer extends almost to the water surface. The region where this minimum occurs is called the *deep sound-channel axis*.

The existence and the thicknesses of these layers vary according to latitude, season, time of day, and meteorological conditions. Figure 15.2a displays the diurnal

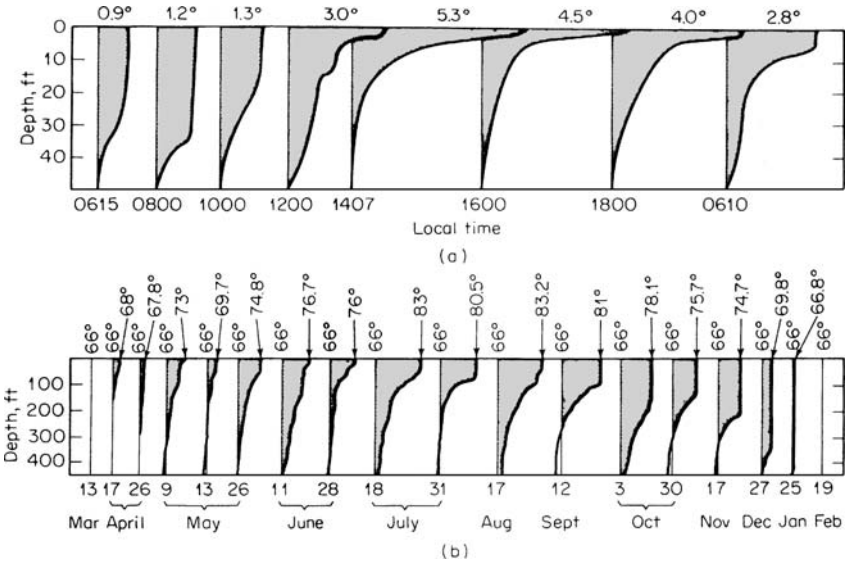


FIGURE 15.2. Diurnal and seasonal variation of a surface layer near Bermuda. In (a) temperature profiles at various times of the day show how surface temperature increases over the temperature at 50 ft depth. The temperature profiles for different parts of the year are given in (b). Temperatures and temperature differences are given in Fahrenheit degrees.

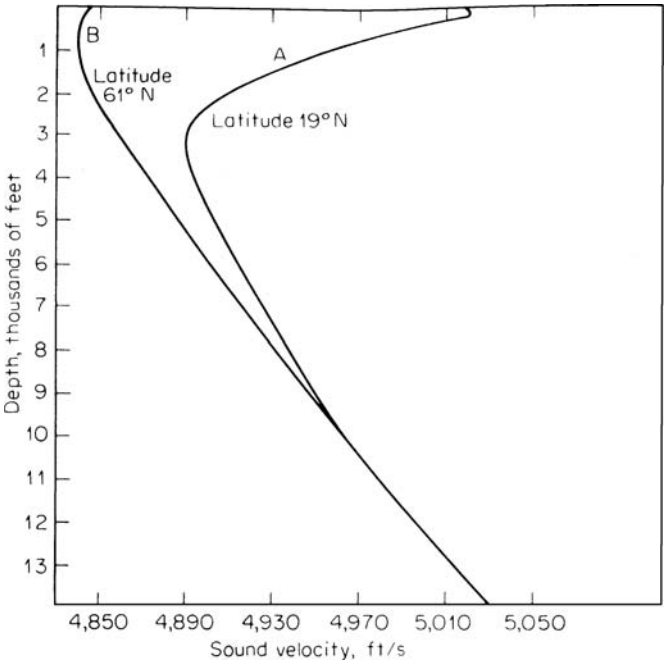


FIGURE 15.3. Effect of latitude on sound-speed profile in deep sea.

behavior of the surface layer near Bermuda. It shows how the temperature profiles vary, when the surface waters of the sea warm up during the course of a sunny day and cool down during the night. These changes in temperatures affect considerably the transmission of sound from a surface-ship sonar, particularly in the afternoon when echo ranging tends to be poorest. Figure 15.2b illustrates a series of bathythermograms taken in the Bermuda area, showing how the season thermocline evolves during the summer and autumn. The effect of latitude on sound-speed profile in the deep sea is shown in Figure 15.3 by profiles for two different locations in the North Atlantic at the same season of the year. At low latitudes (nearer the equator), the velocity minimum occurs at a depth of approximately 3000 ft. At high latitudes the velocity minimum exists near the sea surface, and the main and seasonal thermoclines show a tendency to disappear from the profile.

In the shallow waters of the coastal regions and also on continental shelves, the velocity profiles tend to become far less clear-cut and rather unpredictable. The velocities tend be greatly influenced by surface heating and cooling, changes in salinity, and the presence of water currents. Nearby sources of fresh water tend to complicate these effects and contribute to the spatial and temporal instability of numerous gradient layers.

15.5 Underwater Transmission Loss

Transmission loss TL quantitatively refers to the weakening of sound between a point 1 yard from the source and a point someplace in the sea.[2] Let I_0 indicate the intensity at the reference point 1 yard from the "center" of the acoustic source (thus $10 \log I_0$ denotes the source level) and I_1 is the intensity at the point of interest. The transmission loss TL between the source and the point of interest is

$$\text{TL} = 10 \log \frac{I_0}{I_1} \text{ dB}$$

Time averaging is implied in the above definition. For short pulses, a TL equivalent to that for continuous waves is given by the ratio of the energy flux density at 1 yard from the source E_0 to the energy flux density E_1 at the point 1 of interest, i.e.

$$\text{TL} = 10 \log \frac{E_0}{E_1} \text{ dB}$$

If the metric units of distance are used so that the reference distance is 1 m, TL will be 0.78 dB less than that for the TL based on the reference distance of 1 yard.

Transmission loss can be subdivided into two types of losses: part of the losses is due to *spreading*, a geometric effect, and the remainder is attributable to *absorption losses* that represent conversion of acoustic energy into heat.

If a small source of sound is located in a homogeneous, infinite, lossless medium, the power generated by the source radiates outward uniformly in all directions. The total power radiating outward remains the same, as its wave front expands as a spherical surface with an increasing radius (the radius increases at the rate of $c \times$ time t). Since power P is equal to intensity times the area, i.e.,

$$P = 4\pi r_1^2 = 4\pi r_2^2 = \cdots = \text{constant}$$

and setting $r_1 = 1$ yard, the transmission loss TL to surface r_2 is

$$\text{TL} = 10 \log \frac{I_1}{I_2} = 10 \log \frac{r_2^2}{r_1^2} = 20 \log r_2$$

which can be readily recognized as the *inverse-square* law of spreading, also known as *spherical spreading*.

When the medium is bound by upper and lower parallel planes, the spreading is no longer spherical because the sound cannot cross the boundary plates. Beyond a certain range and as shown in Figure 15.4, the power radiated by the source spreads outward as a wavefront that constitutes a cylindrical surface represented by an radius expanding at the rate of $c \times$ time t. The power P crossing the cylindrical

[2] The U.S. Navy still employs an odd melange of dimensional units in defining, say, source strength in terms of decibels (based on the metric reference of 1 μPa for sound pressure level or 1 pW for sound power level) and the distance from the source of 1 yard rather than the 1 meter used elsewhere. Conversion to the entirely metric system requires the subtraction of 0.78 dB from the U.S. value.

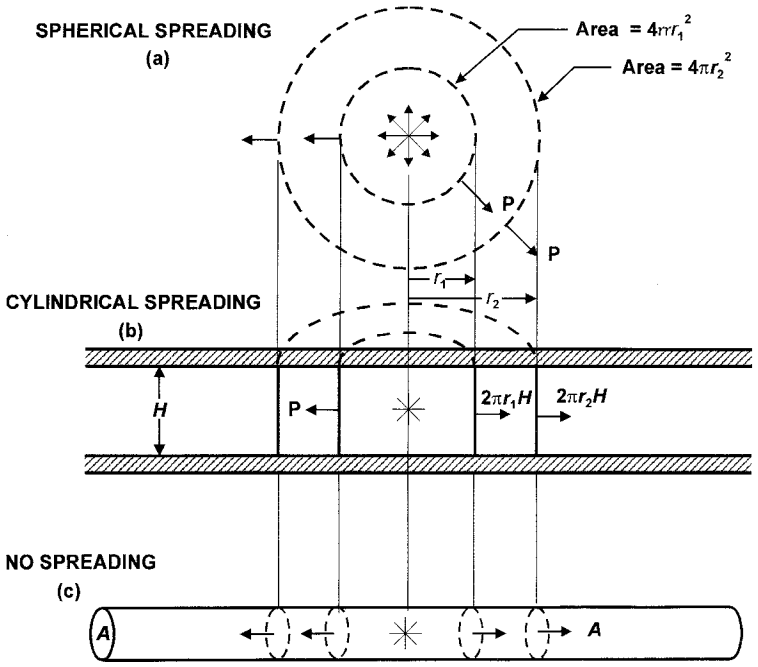


FIGURE 15.4. Spreading of acoustical energy: (a) in an infinite medium, (b) in a medium between two parallel plates, and (c) in a tube.

surfaces at radius r is

$$P = 2\pi H r_1^2 = 2\pi H r_2 = \cdots = \text{constant}$$

where H represents the distance between the two parallel boundary planes. Assigning $r_1 = 1$ yard, the transmission loss (TL) to r_2 is given by

$$\text{TL} = 10 \log \frac{I_0}{I_2} = 10 \log r_2.$$

In this situation the *cylindrical* spreading occurs in the *inverse first power*. This sort of spreading occurs at moderate and long ranges when sound is entrapped within a sound channel in the sea.

We can also consider a *wave guide*, essentially a conducting tube or pipe of constant cross section**.** This is a case where *no spreading* occurs. Beyond a certain range, the area over which the power is distributed remains constant, and the pressure intensity and the TL are independent of range.

A fourth type of spreading occurs when the signal from a pulsed source spreads out in time as the pulse propagates through the medium. The pulse becomes elongated by multipath propagation effects, which causes the pulse to smear out in time as it travels to a receptor. This type of effect is particularly evident in long-range propagation in deep-ocean sound channels. If the medium is infinite and the time stretching is proportional to the range traveled, the intensity falls off as

the inverse cube of the range. In the case of a sound channel of Figure 15.4b, time stretching that is proportional to the range causes the intensity to fall off as the inverse square of the range instead of the inverse first power. In summary the spreading laws may be listed as follows:

Propagation in	Type of spreading	Intensity varies as	TL (dB)
Tube	None	r^0	0
Between parallel plates	Cylindrical	r^{-1}	10 log r
Free field	Spherical	r^{-2}	20 log r
Hypersphere	Free field with time stretching	r^{-3}	30 log r

It should be noted that the hyperspherical case applies in a hypothetical sense to sonar theory.

Absorption varies with range in a manner different from the loss due to spreading. It occurs because acoustic energy converts into heat; and this conversion embodies a true loss of acoustic energy within the propagation medium. Consider a plane wave passing through an absorbing medium. The fractional rate that the intensity of the wave decreases along distance x is proportional to the distance traveled, i.e.

$$\frac{dI}{I} = -kdx \tag{15.1}$$

where k denotes the proportionality constant and the minus sign signifies that dI drops in the direction of increasing x. Integrating Equation (15.1) between ranges r_1 and r_2, the intensity I_2 at r_2 is found from

$$I_2 = I_1 e^{-k(r_2 - r_1)}. \tag{15.2}$$

Rewriting (15.2) yields

$$10 \log I_2 - 10 \log I_1 = -10k(r_2 - r_1) \log_{10} e.$$

Setting $\alpha = 10k \log_{10} e$, the change of the intensity level in dB is now expressed as

$$10 \log \left(\frac{I_2}{I_1} \right) = -\omega(r_2 - r_1)$$

or

$$\alpha = \frac{10 \log(I_1/I_2)}{r_2 - r_1}.$$

The quantity α is the logarithmic absorption coefficient and is usually expressed in decibels per kiloyard (dB/kyd) in the US or decibels per km (dB/km) in the metric system.

15.6 Parametric Variation of Absorption in Seawater

From actual measurements [Wilson and Leonard (1954) among others] it became evident that the absorption of sound in seawater was unexpectedly higher than that of pure water and it could not be attributable to scattering, refraction, or other effects of propagation in the natural environment. For example, the absorption in seawater in the frequency range 5–50 kHz was found to be approximately 30 times that in distilled water. Liebermann (1949) suggested that this excess absorption is attributable to the sort of chemical reaction that evolves under the influence of a sound wave and one of the dissolved salts in the sea.

The absorption of sound in seawater depends on three effects: one is the presence of *shear viscosity*, a classical effect studied by Lord Rayleigh, who derived the following expression for the absorption coefficient:

$$\alpha = \frac{16\pi^2 \mu_s}{3\rho c^3} f^2 \tag{15.3}$$

where

α = intensity absorption coefficient (cm^{-1})
μ_s = shear viscosity, poises (approximately 0.01 for water)
ρ = density, cg/cm^3 (1 for water)
c = sound propagation speed (approximately 15,0000 m/s)
f = frequency (Hz).

According to Equation (15.3) the value of α is $6.7 \times 10^{-11} f^2$ dB/kyd, but this amounts to only about one-third of the absorption actually measured in distilled water. The additional viscosity in pure water, besides that due to shear viscosity, is attributed to another type of viscosity called *volume* or *bulk viscosity*, which is the result of a time lag for water molecules to "flow" in an expansive/compressive manner in reacting to acoustic signals. This viscosity effect adds to the shear viscosity, so the absorption coefficient incorporating both types of viscosity becomes

$$\alpha = \frac{16\pi^2}{3\rho c^3} \left(\mu_s + \frac{3}{4} \mu_b \right) f^2 \tag{15.4}$$

where μ_b denotes the bulk viscosity. For water $\mu_b = 2.81\mu_s$.

Below 100 kHz the predominant reason for absorption in seawater is due to the phenomenon of ionic relaxation of the magnesium sulfate ($MgSO_4$) molecules present in the seawater. This association–dissociation process involves a relaxation time, an interval during which the $MgSO_4$ ions in the seawater solution dissociate under the impetus of the sound wave. Magnesium sulfate accounts for only 4.7% by weight of the total dissolved salts in seawater, but this particular salt was discovered to be the dominant absorptive factor in seawater, rather than the principal constituent NaCl (Leonard et al., 1949).

The ionic relaxation mechanism causes a variation of the absorption with frequency, which is different from that in Equation (15.4). On the basis of more than

30,000 measurements carried out at sea between 2 and 25 kHz out to ranges of 24 kyd. Schulkin and Marsh (1962a,b) modified a frequency–dependency relation originally developed by Liebermann (1948). Their result was

$$\alpha = A\frac{Sf_T f^2}{f_T^2 + f^2} + B\frac{f^2}{f_T} \text{ dB/kyd} \tag{15.5}$$

where

S = salinity in parts per thousand (ppt); A and B are constant
A = 0.0186, a constant
B = 0.0268, another constant
f = frequency (kHz)
f_T = temperature-dependent relaxation frequency
$= 21.9 \times 10^6 - 1.520\,(T + 273)$
T = temperature in EC.

But at frequencies <5 kHz, the attenuation coefficients measured at much higher values than those that would be obtained from Equation (15.5), so it is apparent that there is an additional cause of attenuation other than the ionic relaxation of magnesium sulfate. A number of possible causes were advanced, but the currently accepted explanation for this excess attenuation is the boron-borate relaxation process discovered in the laboratory by Yeager et al. (1973). The boric acid [$B(OH)_3$] ionization process is not a simple one, since its mechanism seems to rely in a complicated way on the presence of other chemicals in the salt water solution, particularly the pH or acidity factor in seawater.

The effect of depth on the absorption also has been well investigated theoretically and experimentally. For the range of hydrostatic pressure extant in the sea, the effect of pressure is to reduce the absorption coefficient by a factor $(1 - 6.54 \times 10^{-6} P)$, where P is the pressure expressed in terms of atmosphere. One atmosphere is the pressure equivalent of 33.9 ft (10.3 m) of water at 39°F (3.9°C). The absorption coefficient at depth d in the sea is found from

$$\alpha_d = \alpha_0(1 - 1.93 \times 10^{-5} d)$$

where d denotes the depth in feet and α_0 is the value of the attenuation coefficient at the surface of the water ($d = 0$). At a depth of 15,000 ft (nearly 4600 m), the absorption coefficient drops to 71% of its value at the surface. However, there has been a considerably larger depth dependence, by a factor of nearly two, observed in at-sea measurements (Bezdek, 1973).

Figure 15.5 constitutes a summary of the processes of attenuation and absorption in the sea. The portion of the curve at the lower left of the plot is due to sound channel diffraction, caused by the fact that the deep sea is no longer an effective duct for such long wavelengths. The dashed line at the right displays the effect of absorption due to viscosity.

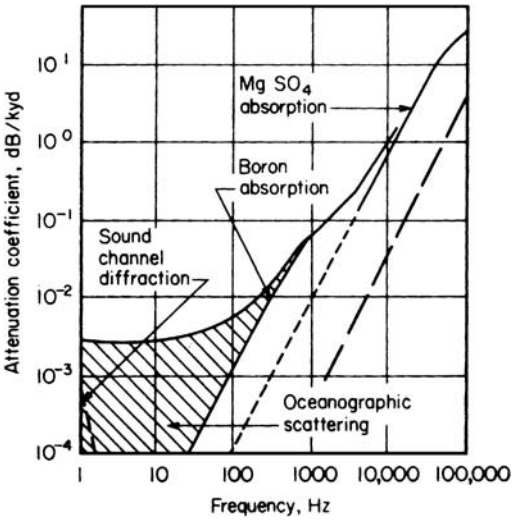


FIGURE 15.5. Attenuation and absorption processes in the sea.

15.7 Spherical Spreading Combined with Absorption

It has been found in propagation measurements made at sea that spherical spreading together with absorption provides a reasonable fit to the measured data under a surprisingly wide variety of conditions, even in situations where spherical spreading is not supposed to occur (e.g. trapping conditions in sound channels). When an approximation of the transmission loss TL is sufficient, the universal spherical spreading law plus the loss due to absorption will serve as a useful working rule:

$$\mathrm{TL} = 20 \log r + \alpha r \times 10^{-3} \tag{15.6}$$

In Equation (15.6) the first term denotes the spherical spreading and the second term the absorption effect; with 10^{-3} inserted to handle the fact that r is given in yards and α is expressed in dB/kyd.

15.8 Underwater Refraction

Refraction is the major factor in altering simple spherical spreading of sound in the ocean. As mentioned above, the factors affecting the sound propagation speed in seawater are temperature, depth, and salinity. Variations of salinity do occur, particularly at the mouths of large rivers where copious amounts of fresh water intermingle with seawater, at the edges of large ocean currents such as the Gulf Stream, and in water near the surface, where rain, evaporation, and ice melting can impose maximal effects. Variations in speed of sound with depth are quite small

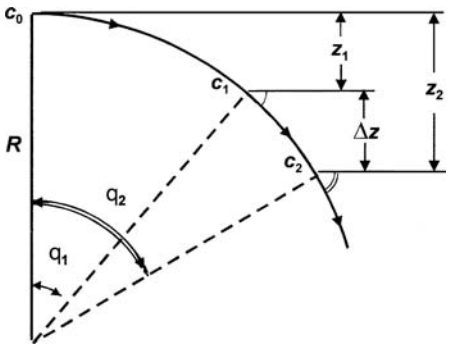


FIGURE 15.6. Diagram used for derivation of the relation between gradient G and the radius of curvature R of a sound ray.

(e.g. 0.1% over a 100 m depth). But variations in speed due to temperature changes are much greater and can fluctuate wildly, particularly near the surface.

When sound varies with ocean depth, the path of a ray through the medium can be determined by applying Snell's law ($\sin \phi/c = \text{constant}$). Because the rays of the greatest interest in the study of oceans are nearly horizontal, it is more usual to restate Snell's law as follows:

$$\frac{\cos \theta}{c} = \frac{1}{c_0} \tag{15.7}$$

where θ is the angle of refractive deflection made with the horizontal at a depth where the speed of sound is c and c_0 is the speed at a depth (real or extrapolated) where the ray would become horizontal.

A complex profile of the propagation velocity versus depth such as that in Figure 15.1 can be simplified for analytical purposes by separating the profile into small enough segments so that the velocity gradient may be considered constant over its length. Advantage is taken of the fact that the path of a sound ray through a stratum of water over which the sound speed gradient G is a constant constitutes an arc of a circle whose center lies at a depth where sound speed extrapolates to zero.

In Figure 15.6 we consider a portion of the ray path with a radius of curvature R. It follows that $\Delta z = R(\cos \theta_1 - \cos \theta_2)$, and the gradient G is

$$G = \frac{c_2 - c_1}{\Delta z}. \tag{15.8}$$

We can combine the last two Equations with Snell's law of Equation (15.7) which now yields

$$R = -\frac{c_0}{G} = -\frac{c}{G \cos \theta}. \tag{15.9}$$

When G is constant, and hence R is a constant, the path of the ray is therefore a circle. The center of curvature of the circle lies at the depth where $\theta = 90^\circ$ which corresponds to $c = 0$. In the case illustrated in Figure 15.6, the speed gradient is negative, so R is positive. Otherwise, if the speed gradient were to be positive, R would become negative and the path would refract upward.

Once the radius of curvature of each segment of the path is established, the actual path can be traced through graphic or computerized means. Let the initial angle of deflection of the ray be designated θ_0, and use be made of the geometry of Figure 15.6 along with Equation (15.9). The changes in range Δr and depth Δz are given by

$$\Delta r = \frac{1}{G} \frac{c_0}{\cos\theta_0} (\sin\theta_1 - \sin\theta_2) \tag{15.10}$$

$$\Delta z = \frac{1}{G} \frac{c_1}{\cos\theta_1} (\sin\theta_2 - \sin\theta_1) \tag{15.11}$$

Applying the small angle approximations ($\cos\theta \approx 1$, $\sin\theta \approx 0$, etc.) and eliminating θ from the last two Equations yields a convenient approximate relationship between the range and depth increments along a ray for $*\theta*$ less than 20° and for $\Delta r \ll |c_1/G|$:

$$\Delta z = \tan\theta_0 \Delta r - \frac{G}{2c_0}(\Delta r^2). \tag{15.12}$$

15.9 Mixed Layer

Wave action can cause the water to mix in the surface layer, thus creating what is called a *mixed layer*. The positive sound-speed gradient in this layer entraps sound near the surface. After it is developed, the mixed layer tends to exist until the sun heats up the upper portion, decreasing the gradient. This heating effect engenders a negative gradient that leads to a downward refraction and the loss of sound from the layer. Because this occurs later during daytime, this effect became known as the *afternoon effect*. During the night, surface cooling and wave mixing permit this isothermal layer to reestablish itself.

A computer-produced ray diagram (the discontinuous form of the rays are due to the manner the velocity profile was subdivided in the computer program) for a source in a fairly typical mixed layer is shown in Figure 15.7. The conditions for which the diagram was plotted, the ray leaving a source at 1.76° becomes horizontal at the base of the layer, Rays leaving the source at smaller angles stay entrapped in the layer; and rays that leave the source at greater angles are sent into the lower depths of the sea. A shadow zone is created beneath the mixed layer at a range beyond the direct and near sound field. This zone is isonified by scatter sound from the sea surface and by diffusion of sound out of the channel, caused by the nature of the lower boundary.

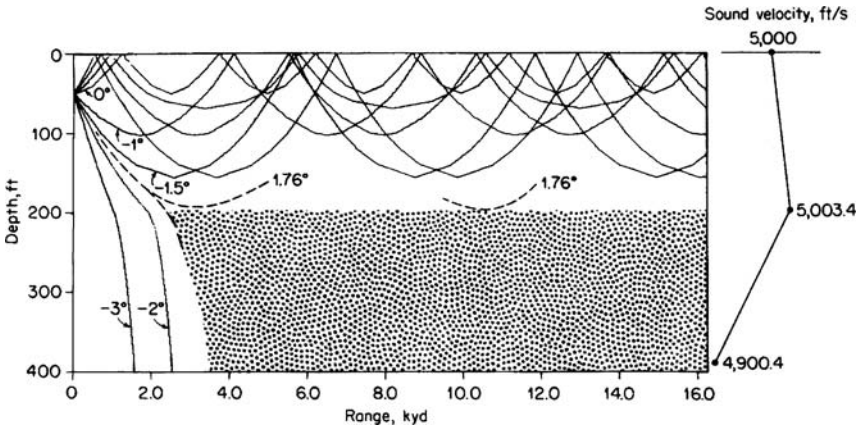


FIGURE 15.7. A computer-produced ray diagram for sound transmission from a 50-ft (15.24-m) source in a 200-ft (61-m) mixed layer.

15.10 Deep Sound Channel

In Section 15.4 reference was made to a region constituting the deep sound channel where the sound propagation speed reaches a minimum in the ocean depths. All rays originating near the axis of this channel and making small angles with the horizontal will return to the axis without reaching the ocean surface or bottom, thus remaining entrapped within that channel. Absorption of low frequencies in seawater tends to be quite small, so the low-frequency components of explosive charges detonated in this channel can travel tremendous distances, and they have been detected more than 3000 km away. The reception of these explosive signals by two or more well-separated hydrophone arrays can permit an accurate determination of the explosion's location by triangulation. Passive sonar is currently being used in deep sound channels to monitor activities in deep ocean.

15.11 Sonar Transducers and Their Properties

Underwater sound equipment are designed to detect and analyze underwater sound. They generally consist of a hydrophone array that consists of transducers that convert acoustic energy into electrical energy, and vice versa, and a signal processing system to analyze and display the signals aurally or visually. A transducer that accepts sound and converts it into electricity is called a *receiver* or *hydrophone*. A transducer that converts electrical energy into sound is called a projector. Some sonar systems use the same transducer to generate and receive sound. There are two principal types of transducers according to the special properties of their activation materials. One type of transducer depends on *piezoelectricity*

and its variant, *electrostriction,* and the other type functions on the principle of *magnetostriction*. Certain crystalline substances, such as quartz, ammonium dihydrogen phosphate (ADP), and Rochelle salt, generate a charge between certain crystal surfaces when they are subject to pressure. Conversely, when a voltage is applied to these substances, this causes stresses to occur in them. Such materials are said to be *piezoelectric*. Electrostrictive materials are polycrystalline ceramics that produce the same effect and these have to be properly polarized in a strong electrostatic field. Barium titanate and lead zirconate titanate are examples of such materials.

A magnetostrictive material such as nickel exhibits the same effect as piezoelectricity but it does so under the influence of a magnetic field rather than applied stresses. It changes its dimensions when subjected to a magnetic field and, conversely, changes the magnetic field within and around it when it becomes stressed. In other words, when a properly designed nickel element is subjected to an oscillating magnetic field, a mechanical oscillation is produced which generates acoustic waves in water. Magnetostrictive materials are also polarized in order to avoid frequency doubling and to achieve a higher efficiency.

Piezoelectric and magnetostrictive types of transducers are more suitable than other kinds of transducers for use underwater due to better impedance matching with water. Because they are relatively inexpensive and can be readily fashioned into the desired shapes, ceramic materials are finding increasing applications as underwater devices. Other types of units now include the thin-film transducers (Hennin and Lewiner, 1978) and fiberoptic hydrophones (Hucaro et al., 1977).

Arrays

While single piezoelectric or magnetostrictive elements are normally used in hydrophones for research or measurement purposes, much of the applications of hydrophones entail hydrophone *arrays* that use a number of properly spaced elements. The following reasons exist for the use of arrays:

1. The array is more sensitive, as a number of elements will generate more voltage, if connected in series, or more current, if connected in parallel.
2. The array provides directivity that enables it to discriminate between sounds coming from different directions.
3. An improved signal-to-noise ratio SNR over that of a single hydrophone is provided, because the array discriminates against isotropic or quasi-isotropic noise to favor a signal arriving from a direction that the array is pointing to.

Because of the above advantages, most practical applications of underwater sound make of arrays. Moreover, the first and second benefits listed above also apply to projectors as well as to hydrophones.

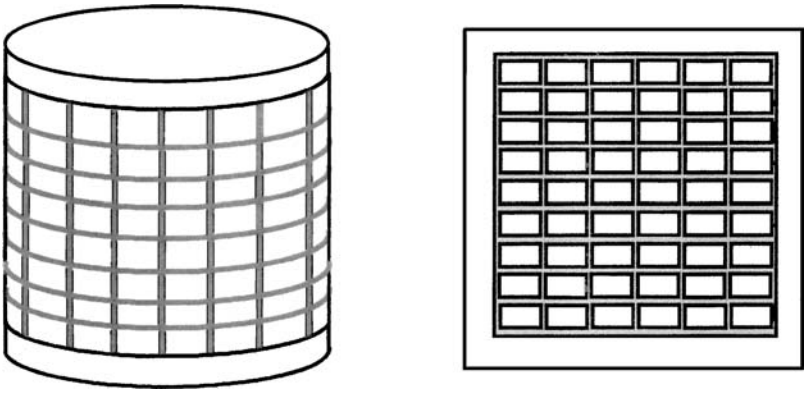


FIGURE 15.8. Cylindrical array and plane array of transducers.

Examples of a cylindrical and a plane array are shown in Figure 15.8. Spherical arrays have also been constructed for installation on submarines.

Array Gain

The improvement in the signal-to-noise ratio SNR, which results from the use of arraying of hydrophones is measured by a parameter called *array gain*, which is defined by

$$\mathrm{AG} = 10 \log \left(\frac{(S/N)_{\text{array}}}{(S/N)_{\text{single element}}} \right) \tag{15.13}$$

The numerator of Equation (15.13) denotes the SNR at the array terminals and the denominator represents the SNR of a single element of the array. It is assumed that all the elements in the array are identical.

Transducer Response

The effectiveness of a hydrophone in converting sound into an electric signal is called the *response* of the hydrophone. It relates the generated voltage to the acoustic pressure of the sound field. The receiving response of a hydrophone is defined as the voltage produced across the terminals by a plane wave of unit acoustic pressure (the value before the introduction of the hydrophone into the sound field). The receiving response is usually expressed as the *open-circuit response* that is obtained when the hydrophone connects to an infinite impedance. The customary unit of the receiving response is the number of decibels relative to 1 volt produced by an acoustic pressure of 1 μPa and it is written as decibels re 1 V/μPa.

Example Problem 1

Find the voltage across the hydrophone terminals exposed to an acoustic rms pressure of 1 μPa if the receiving response is –80 decibels re 1 V.

Solution

We use the relationship

$$\text{SPL} = 20\log(p/1\ \mu\text{Pa})$$

and since $\text{SPL} = -80$ dB,

$$\text{p}/1\ \mu\text{Pa} = \text{antilog}(-80/20),$$

yielding $p = 0.0001$ μPa. For the 1 μPa sound field, the voltage is 0.0001 V across the terminals.

For a projector, the *transmitting current response* is the acoustic pressure produced at a point 1 m from the projector in the direction of the axis of its beam pattern by a unit current into the projector. This means that the transmitting response is stated in terms of the number of decibels relative to 1 μPa as measured at the reference distance, produced by 1 A of current into the electric terminals of the projector. While the transmitting response are usually referred to a reference distance of 1 m from the source, a correction of 0.78 dB must be added in order to convert the transmission response expressed in terms of a reference level of 1 yard instead of 1 m.

Example Problem 2

Predict the rms pressure at 1 m when the projector is driven with a current of 1 rms ampere, if the response is rated at 100 dB re 1 μPa/A (referred to 1 m). Express the response in terms of a reference distance of 1 yard instead of 1 m.

Solution

The corresponding rms acoustic pressure for 100 dB is 10^5 μPa, which is produced by 1 rms ampere current. The corresponding transmitting response for 1 yard is $100 + 0.78 = 100.78$ dB.

15.12 The Sonar Equations

The purpose of the sonar equations, originally formulated during World War II and derivative of similar considerations in radar, is twofold: (a) prediction of the capabilities of existing sonar equipment with respect to detection probability or search rate and (b) design of new equipment to meet preestablished range of detection or actuation.

In our mathematical exposition of the sonar principle, we subscribe to the assumption that target strength TS is a function of the source and echo levels, respectively, as well as the transmission loss TL that occurs in the echo-ranging process. The assigned function of the sonar may be the detection of an underwater target, or it can be the homing of an acoustic torpedo at the instant when it begins to ascertain its target. Of a total signal energy received by a sensor, a portion may be desired, and is considered to be the *signal*. The balance of the acoustic energy is undesired and is termed the *background*. The background consists of *noise*, the basically steady state portion that is not attributable to the echo-ranging, and *reverberation* which represents the slowly decaying portion of the background caused by the return of the original acoustic output from scatterers dispersed in the sea. In the design of a sonar system, it is the objective to find ways of increasing the overall response of the system to the signal and to decrease the response of the system to the background, in other words, to increase the signal-to-background ratio.

A sonar system serves a practical purpose such as detection, classification (establishing the character of the target), torpedo hunting, communication, or fish finding. In each of these tasks there will be a specific signal-to-background ratio and a level of performance in successfully detecting targets with a minimum of "false alarms" that erroneously indicate the presence of a target when no target is present. If the signal increases sufficiently to equal the level of the background, the desired purpose will be achieved when the signal level equals the level of the background, which just masks it, i.e.

$$\text{Signal level} = \text{Background masking level} \tag{15.14}$$

Masking does not mean that all of the background interferes with the signal. Only the portion that lies in the frequency band of the signal will cause masking, just as in psychoacoustics, where a broadband noise masks out a pure tone or narrow-band signal presented to the human ear.

The equality of (15.14) constitutes the one instant of the time when a target approaches or recedes from a receiver. At short ranges, the signal level from a target should handily exceed the background masking level, while the reverse will occur at long ranges. But it is at the instant of (15.14) when the sonar system just begins to perform its function, which is of greatest interest to the sonar designer.

The source level, SL, which is defined in terms of intensity at 1 m (formerly 1 yard), was derived from physical concepts in order to express separately the effects on the signal strength of the echo, namely, (i) the size, shape, and orientation of the target; (ii) the intensity of the source; and (iii) the range of the target. At long ranges, only the transmission loss TL depends on the range. At shorter ranges, target strength TS depends on the range as well as the size, shape, and orientation of the target. If the source is quite close to the target, different parts of the target are struck by sound of different intensities, or if the receiver is so close that the spreading of the sound reflected from the target to it is not the same as the spreading from a point source, the target strength term will depend on the range.

Active and Passive Equations

The sonar parameters are determined by the equipment, the medium, and the target. We denote the following parameters, which are stated in terms of dB relative to the standard reference intensity of a 1-μPa plane wave:

Equipment parameters
- SL: projector source level
- NL: self-noise level (also called electronic noise)
- DI: directivity index
- DT: detection threshold

Medium parameters
- TL: transmission loss
- RL: reverberation level
- NL: ambient noise level

Target parameters
- TS: target strength
- SL: target source level.

Note that two pairs of parameters are given the same symbol because they are identical. This set of parameters are not necessarily all inclusive, nor is this set unique, for other parameters such as sound velocity or backscattering cross section could be considered. The parameters chosen above are conventional ones applied in underwater technology.

In order to understand the significance of the above listed quantities, consider Figure 15.9, which illustrates a schematic of an echo-ranging process. A transducer operating as both sound source and receiver produces a source level of SL decibels at a unit distance (generally 1 m worldwide and 1 yard in the English system) on its axis. Let the axis of the sound source be properly aimed toward the target; the radiated sound will reach the target with a transmission loss, and the level of the signal reaching the target will be SL $-$ TL. On reflection or scattering from the target with target strength TS, the reflected or the backscattered level will be SL $-$ TL $+$ TS (at a distance of 1 m from the acoustic center of the target in the direction returning to the source). This reflected signal also undergoes attenuation by the transmission loss TL as a result of its travel to the source. The level of the echo reaching the source thus becomes SL $-$ 2TL $+$ TS. Now if we consider the background noise and assume it to be isotropic noise rather than reverberation, the background level will simply be noise level. But this level will be lessened by the directivity index (DI) of the transducer serving as a receiver, so the relative noise power at the transducer interface is NL $-$ DI. Because the axis of the transducer points in the direction from which the echo is traveling, the relative echo power is unaffected by the transducer directivity. At the transducer terminals, the echo-to-noise ratio becomes

$$\text{SL} - 2\,\text{TL} + \text{TS} - (\text{NL} - \text{DL})$$

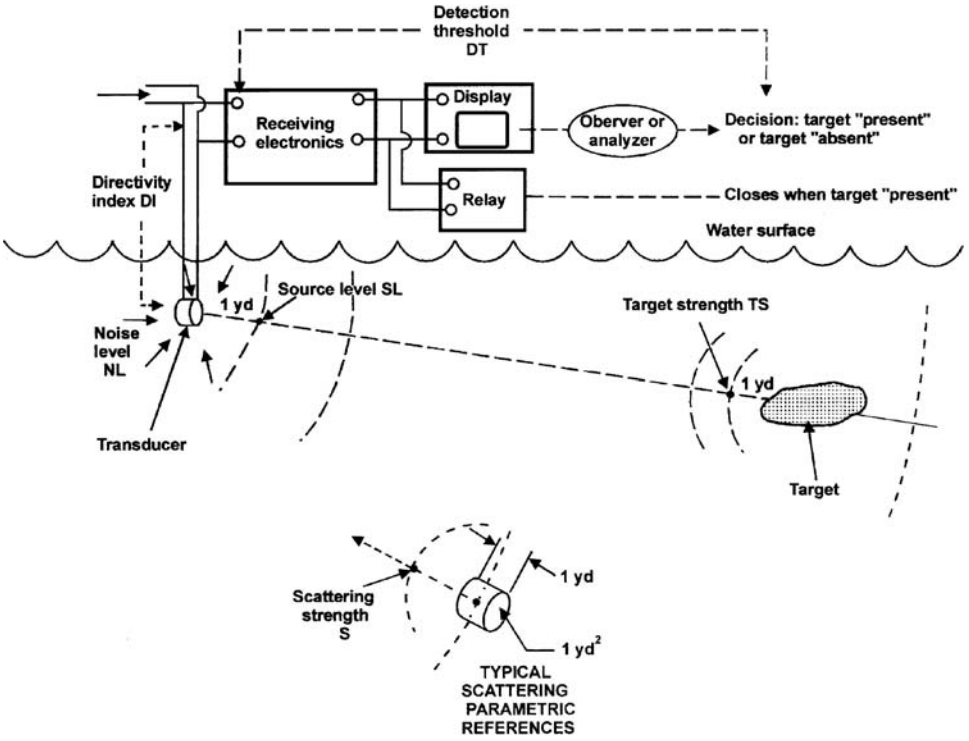


FIGURE 15.9. Schematic of echo ranging process.

Now let the sonar act as a detector, i.e., it is to give an indication on some kind of display that an echoing target is present. When the input signal-to-noise ratio exceeds a specific detection threshold DT, thus meeting preset probability criteria, a relay can be activated to indicate on a display that a target is present. Otherwise, when the input signal-to-noise ratio falls below the detection threshold DT, the indication will be that a target is absent. But when the target is just being detected, the signal-to-noise ratio equals the detection threshold DT, i.e.,

$$\mathrm{SL} - 2\,\mathrm{TL} + \mathrm{TS} - (\mathrm{NL} - \mathrm{DI}) = \mathrm{DT} \tag{15.15}$$

Equation (15.15) characterizes the active-sonar equation as an equality in terms of the *detection threshold*. In recognizing that only a portion of the noise power lying above the DT masks the echo, we could rearrange (15.15) as follows:

$$\mathrm{SL} - 2\mathrm{TL} + \mathrm{TS} = \mathrm{NL} - \mathrm{DI} + \mathrm{DT} \tag{15.16}$$

Equation (15.16) places the echo level effects on the left-hand side and those pertaining to the noise-masking background level on the other side. Equation (15.16) constitutes the active-sonar equation for the *monostatic case*, one in which the source and the receiving hydrophone are coincident and the echo from the target travels back to the source. In some sonar applications, a *bistatic* arrangement is

used—i.e., the source and the receiver are separate, and the two transmission losses to and from the target are not generally equal. In some sonars it is virtually impossible to resolve the receiving directivity index DI and detection threshold DT, so it becomes legitimate to combine these two terms as DI − DT into a single parameter describing the increase in signal-to-background ratio produced by the entire receiving system which includes the transducer, processing electronics, and the display.

What happens when the background consists of reverberation rather than noise? Instead of DI, which was defined in terms of an isotropic background and is now inappropriate, the (NL − DI) term in Equation (15.16) is replaced by an *equivalent plane-wave reverberation level* RL observed at the hydrophone terminals. Equation (15.16) then becomes

$$\mathrm{SL} - 2\mathrm{TL} + \mathrm{TS} = \mathrm{RL} + \mathrm{DT} \tag{15.17}$$

The parameter DT will possess a value for reverberation that is different from the DT for noise.

In the passive or "listening" situation, the target itself produces the signal that is detected, and one-way transmission rather than two-way transmission is entailed. Target strength TS now becomes irrelevant and the passive sonar equation becomes

$$\mathrm{SL} - \mathrm{TL} = \mathrm{NL} - \mathrm{DI} + \mathrm{DT} \tag{15.18}$$

Table 15.3 lists the parameters and their definitions in brief, while Table 15.4 provides the terminology of commonly used terms for describing sonar parameters. A number of the parameters in Table 15.3, namely, SL, TL, TS, and the scattering strength (which determines RL) use 1 yard as the reference distance. To convert to 1-m reference, these quantities should be reduced by $0.78 = 20 \log [1\ \mathrm{m} \times (39.37\ \mathrm{in./m}) \times (1\ \mathrm{yard}/36\ \mathrm{in.})]$. The attenuation coefficient commonly expressed in dB/kiloyard should be multiplied by 1.094 to convert the coefficient to dB/km. In the United States, it is generally more convenient to find the range first in kiloyards and then to divide by the factor 1.094 in order to express the range in kilometers.

15.13 Noise, Echo, and Reverberation Levels

The above sonar equations constitute a statement of equality between the *signal*, which is the desired portion (the echo or noise from the target) of the acoustic field, and the undesired portion, i.e., the *background* of noise and reverberation. This equality holds true at only *one range*; at all the other ranges, the equality will no longer exist. This fact is demonstrated in Figure 15.10 in which the curves of the echo level, noise-masking level, and the reverberation-masking level are displayed as functions of range. The echo and reverberation levels drop off with increasing range, but the noise remains fairly constant. The echo level curve falls off more rapidly with range than does the reverberation-masking curve, and intersects it at the reverberation-limited range r_r. This curve will also meet the noise-masking

TABLE 15.3. Sonar Parameters.

Parametric symbol	Reference	Definition
Source level: SL	1 yard from source on its acoustic axis	10 log (intensity of source/1 μPa)
Transmission loss: TL	1 yard from source and at target or receiver	$10\log\left(\frac{\text{signal intensity at 1 yard}}{\text{signal intensity at target or receiver}}\right)$
Target strength: TS	1 yard from acoustic center of target	$10\log\left(\frac{\text{echo intensity at 1 yard from target}}{\text{incident intensity}}\right)$
Noise level: NL	At hydrophone location	$10\log\left(\frac{\text{noise intensity}}{1\ \mu\text{Pa}}\right)$
Receiving: DI directivity index	At hyrdrophone terminals	$10\log\left(\frac{\text{noise power generated by an equivalent isotropic hydrophone}}{\text{noise power generated by actual hydrophone}}\right)$
Reverberation: RL level	At hydrophone terminals	$10\log\left[\frac{(\text{reveberation power at hydrophone terminals})}{1\ \mu\text{Pa}}\right]$
Direction: DT Threshold	At hydrophone terminals	$10\log\left[\frac{\left(\text{signal power to just at perform a certain function}\right)}{\text{noise power at hydrophone terminals}}\right]$

Note that some of the parameters, viz. SL, TL, TS (and the scattering strength) in Table 15.1 use 1 yard as the reference distance. To obtain their values based on the reference distance of 1 m, 0.78 dB must be subtracted from the values of these parameters.

level curve at range r_n. If the reverberation is high, the echo will be of lesser value and the range may be considered to be *reverberation-limited*. But if the noise-masking level occurs at the level represented by the dashed line in Figure 15.10, the echoes will then fade away into a background of noise rather than reverberation. The new noise-limited r_n' will be less than the reverberation-limited

TABLE 15.4. Sonic Parameters: Terminology of Various Combinations.

Nomenclature	Parameter	Comments
Echo level	SL – 2TL + TS	Intensity of the echo measured in the water at the hydrophone
Noise-masking level	NL − DI + DT	Determines minimum
Reverberation-masking level	RL + DT	Detectable echo level
Echo excess	SL − 2TL + TS − (NL − DI + DT)	Detection just occurs, when echo excess just equals zero
Performance figure	SL − (NL − DI)	Difference between the SL and the NL measured at the hydrophone
Figure of merit	SL − (NL − DI + DT)	Equals the maximum allowable two-way TL for TS = 0 for active sonar, or one-way TL in passive sonars

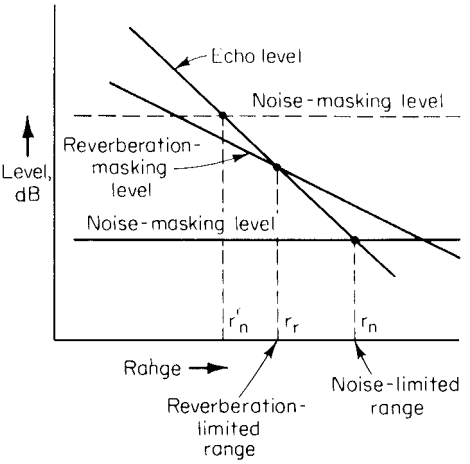


FIGURE 15.10. Effect of range on echo level, noise masking, and reverberation masking.

range r_n, and the range will hence be *noise-limited*. Both ranges can be established by the use of the appropriate sonar equation.

It is necessary for a sonar designer or sonar operator to know whether a sonar will be noise-limited or reverberation-limited. In general, the curves for echo and reverberation will not occur as straight lines because sound propagation and distribution of echo-yielding scatterers add to the complexity of the situation. In the design of a sonar system for specific purposes, these curves should be created from the best information available for the conditions that are most likely to be encountered.

A convenient graphical method of solving the sonar equations for passive sonars is the SORAP, the acronym for "sonar overlay range prediction." Figure 15.11

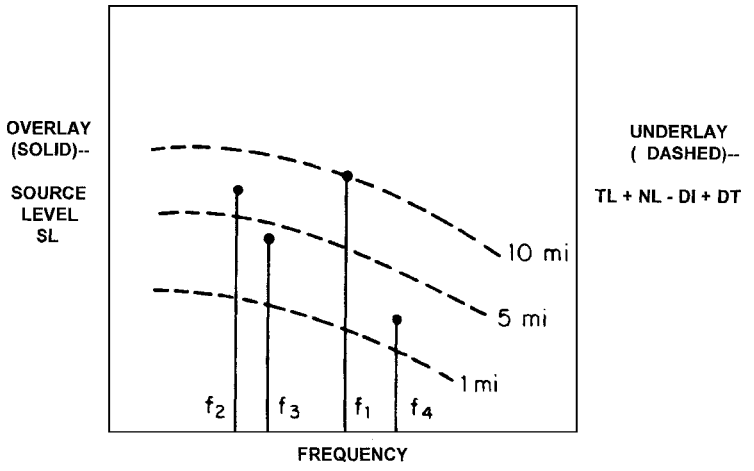


FIGURE 15.11. SORAP (Sonar Overlay RAnge Prediction) graphical method for solving the passive sonar equation.

consists of two plots that are overlaid on each other. The solid lines constitute an overlay of a plot of SL versus frequency for a specific passive target or class of targets. The underlay denoted by dashed lines is a plot of the sum of the parameters (TL + NL − DI + DT) for a specific passive sonar at a number of different ranges. The range and frequency at which the target can be discerned can be determined by inspection of the plots. For example, the target will be first detected at a range of 10 miles according to the line component at frequency f_1. But suppose it is required that three spectral lines must appear on the display before a detection is called; the range would be reduced to 4 miles and the lines at frequencies f_1, f_2, and f_3 would be displayed. This procedure helps to distinguish the target parameter SL from the equipment and medium parameters at the location where it is used, while it accommodates a wide range of frequencies. Targets can therefore be compared for the same locations or locations can be compared for the same targets, and so on.

15.14 Transient Form of Sonar Equations and Pulses

So far, the sonar equations have been expressed in terms of the average acoustic power per unit area or *intensity* of the sound radiated by the source or received from the target. But the time interval implied by the terminology "average" can yield unreliable results in situations where short transient sources exist or whenever severe distortion is incurred in sound propagation in the medium during the course of scattering from the target.

We can adopt a more general approach by writing the sonar equations in terms of energy flux density, which is defined at the acoustic energy per unit area of the wavefront. Consider a plane acoustic wave that has a time-dependent pressure $p(t)$. The energy flux density of the wave is given by

$$E = \frac{1}{\rho c} \int_0^\infty p^2(t)\, dt. \tag{15.19}$$

Because the intensity is the rms pressure of the wave divided by the acoustic impedance ρc, averaged over a time interval T, i.e.,

$$I = \frac{1}{T} \int_0^T \frac{p^2(t)}{\rho c} dt$$

it follows that

$$I = \frac{E}{T}. \tag{15.20}$$

The time interval T represents the duration over which the energy flux density of the sound wave is to be averaged to yield the intensity. In the case of long-pulsed active sensors, this time interval equals the duration of the emitted pulse and very nearly equals the duration of the echo. But for short transient sonars, the interval T becomes rather ambiguous, and the duration of the echo can be considerably different from the duration of the transient emitted by the source. Urick (1962,

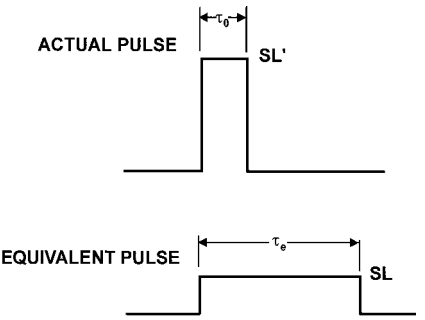


FIGURE 15.12. Time stretching: equivalent source level (SL) in short-pulse sonar.

1983) demonstrated that under these conditions that the intensity form of the sonar equations can be applied, providing the source level is defined as

$$\mathrm{SL} = 10\log E - 10\log \tau_{\mathrm{e}} \tag{15.21}$$

where E is the energy flux density of the source referred to 1 yard and measured in units of the energy flux density of a 1-μPa plane wave taken over an interval of 1 s, and τ_{e} represents the duration of the echoes expressed in seconds for an active sonar. As an example, E can be determined for explosives by measurements for a given charge weight, the depth where the explosion occurs, and the type of explosive.

Consider a pulsed sonar emitting a rectangular pulse of constant source level SL′ over a time interval τ_0. The energy density of a pulse equals the average intensity H duration,

$$10\log E = \mathrm{SL}' + 10\log \tau_0. \tag{15.22}$$

Combining Equations (15.21) and (15.22), the effective source level (SL) to be used accordingly in sonar equations is

$$\mathrm{SL} = \mathrm{SL}' + 10\log \frac{\tau_0}{\tau_e} \tag{15.23}$$

where τ_0 denotes the duration of the emitted pulse of source level, SL′, and τ_e the duration of the echo. For long-pulsed sonars, τ_0 is equal to τ_e, and therefore SL = SL′. In the case of short-pulsed sonars, $\tau_0 > \tau_e$, and thus the effective SL is less than the actual SL′ by the amount 10 log (τ_0 /τ_e). This effect of time stretching is depicted in Figure 15.12. A short pulse duration τ_0 at source level SL′ is replaced in a sonar calculation by a longer pulse of duration τ_e at a lower SL. The energy flux density source levels must be the same for these two source levels, i.e.,

$$\mathrm{SL} + 10\log \tau_e = \mathrm{SL}' + 10\log \tau_0$$

which is merely a rearrangement of Equation (15.23). A pulse emitted by the source stretches out in time and becomes reduced in level by the multipath effect of propagation and by the mechanism of target reflection. The appropriate values

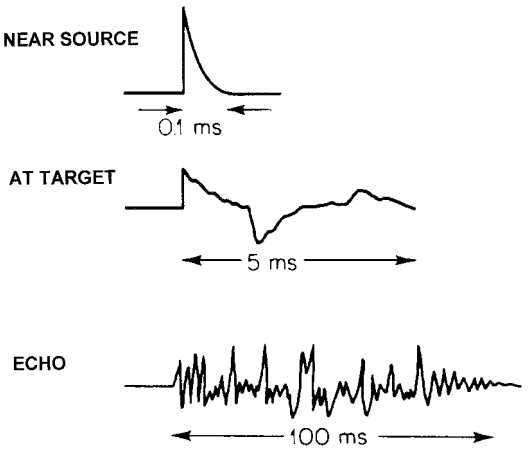


FIGURE 15.13. Distortion of the pressure resulting from an explosive pulse upon arrival at an extended target. The return echo in the vicinity of the source is also shown here.

of the other sonar parameters in the sonar equations, such as target strength (TS) and transmission loss (TL), are those applying to long-pulse of CW conditions, in which the effects of multipaths in the medium and target reflection are summed up and accounted for.

In active short-pulse sonars, the echo duration τ_e becomes in itself an important parameter. In Figure 15.13, a pulse is shown as a short exponential transient at the source, as a distorted pulse at the target, and as an echo reflected back to near the source. A shock wave pulse, for example, from an underwater explosion occurs as an exponential pulse. It may have an initial duration of 1/10 of a millisecond, and it can become distorted into an echo that is 1000 times as long.

The duration of the echo can be subdivided into three components: τ_0, which is the duration of the emitted pulse near the source; τ_m, the additional time needed for two-way propagation in the underwater; and τ_r, the additional duration assessed by the extension in range of the target. Hence

$$\tau_e = \tau_0 + \tau_m + \tau_r. \tag{15.24}$$

Typical values of these three components of the echo duration for different circumstances are listed in Table 15.5.

TABLE 15.5. Examples of Echo Duration.

Component	Representative values (ms)
Duration of the emitted pulse at short ranges	Explosives: 0.1 Sonar: 100
Duration produced by multiple paths	Deep water: 1 Shallow water: 100
Duration produced by a submarine target	Beam aspect: 10 Bow-stern aspect: 100

Example Problem 3

Estimate the echo time duration of an explosive echo from a beam aspect submarine patrolling shallow water.

Solution

From Table 15.5 and Equation (15.24), the time duration is $0.1 + 100 + 10 = 110.2$ ms.

15.15 Overview of the Sonar Equations

A summarization of the sonar equations may be listed as follows:

Active sonars (monostatic)

Noise background

$$\mathrm{SL} - 2\mathrm{TL} + \mathrm{TS} = \mathrm{NL} - \mathrm{DI} + \mathrm{DT}. \tag{15.25}$$

Reverberation background

$$\mathrm{SL} - 2\mathrm{TL} + \mathrm{TS} = \mathrm{RL} + \mathrm{DT}_R. \tag{15.26}$$

Passive sonars

$$\mathrm{SL} - \mathrm{TL} = \mathrm{NL} - \mathrm{DI} + \mathrm{DT}_N. \tag{15.27}$$

The detection threshold (DT) differs quantitatively for reverberation and for noise and so it carries subscripts to denote the difference.

15.16 Shortcomings of the Sonar Equations

The sonar equations expressed in terms of intensities may not always be complete for certain types of sonars. For example, the short-pulse sonars need the addition of the echo duration to account for time stretching caused by multipath propagation. It must also be realized that the sea is an ever-stirring medium that contains inhomogeneities of different sorts, with irregular boundaries with the topmost boundary on the move. Multipath propagation tends to predominate, since so many of the sonar parameters fluctuate erratically with time. Other irregularities may occur because of internal changes in measurement equipment and possible reconfiguration of the platform on which the equipment is mounted. In essence, the "solutions" proffered by the sonar equations really constitute a "best estimate" time average of what is really a stochastic problem. Thus, precise calculations to, say the nearest tenth of a decibel are exercises in futility, and the solutions must be considered "best guesses" or "most probable" values.

15.17 Theoretical Target Strength of a Sphere

Consider a sphere of radius r as the subject target. As we shall see, the effect of the pulse length of the probe signals constitutes a very important factor. No account is taken of the wave nature of sound, i.e., the effects of interference, diffraction, and phase differences are ignored here. Let I_0 denote the intensity of the incident sound wave striking the target and I_r the intensity of the reflected sound signal measured at some particular point. With all the other factors remaining constant, we can write

$$I_0 \propto I_r \, .$$

The intensity I_r of the reflection is a function of the target orientation and the location of its measurement. Because I_r is usually measured at the probe signal point, mathematical treatment becomes considerably simplified. The inverse square law, which holds for large, not small, distances is expressed as

$$I_r = \frac{K I_0}{r^2} \tag{15.28}$$

where K is a constant dependent upon the size, shape, and orientation of the target. Equation (15.28) does not apply to explosive sounds. For the incident signal

$$I_0 = \frac{F}{r^2} \tag{15.29}$$

where F is the intensity of the projected sound 1 m away from the source. Here it is tacitly assumed that

$$r \gg D$$

where D is the order of magnitude of the size of the source. Combining Equations (15.28) and (15.29)

$$I_r = \frac{K F}{r^4}. \tag{15.30}$$

For an ideal medium, according to Equation (15.30), the intensity of an echo is inversely proportional to the fourth power of the range, provided the echo is measured in the source location and the range is much larger than the dimensions of either the target or the source.

Reexpressing Equation (15.30) in logarithms, we obtain

$$10 \log I_r = 10 \log K + 10 \log F - 40 \log r. \tag{15.31}$$

In the course of a signal traveling to a target and the echo reflecting back to the signal source, some attenuation in the signal intensity must occur. This drop in intensity in each direction has been defined as the transmission loss TL that should equal $20 \log r$ in the idealized case of Equation (15.31). The total TL is therefore 2TL, i.e. $= 40 \log r$, which represents the attenuation of the signal traveling to and from the target. It can be recalled that losses do not occur solely owing to the effect of the inverse square law; they are also attributable to absorption and scattering

in sea water, bending by temperature gradients (with consequent focusing and spreading out). Therefore, the actual transmission loss does not equal $20 \log r$ alone, but includes other effects of attenuation. The function 2TL represents a more general situation than $40 \log r$ in Equation (15.31), which is then recast as

$$10 \log I_r = 10 \log K + 10 \log F - 2\text{TL}. \tag{15.32}$$

As mentioned earlier, because of the great variance of oceanographic conditions, the quantity 2TL can be established only through careful measurements. Setting target strength TS

$$\text{TS} = 10 \log K,$$

and the echo

$$\text{E} = 10 \log I_r,$$

and source level

$$\text{SL} = 10 \log F,$$

we obtain the first sonar equation of Table 15.4, which constitutes the fundamental definition of signal strength:

$$\text{TS} = \text{E} - \text{SL} + 2\text{TL}. \tag{15.33}$$

As it entails only directly measurable quantities Equation (15.33) is particularly useful in the computation of TS from data measured at sea.

A sphere presents a perfectly symmetric target. The echoes it returns to a sound source are completely independent of its own orientation, and it is for this reason that spheres make convenient experimental targets in echo-ranging measurements. In a simple derivation, we consider a plane wave of intensity I_o striking a sphere of radius a and cross sectional area πa^2. The total sound energy intercepted by the sphere is thus $\pi a^2 I_0$ in the ideal case of perfect reflection. Now let us assume uniform reflectivity in all directions. At a distance r from the sphere's center the acoustic energy will be spread uniformly over the surface of a sphere of radius r or over the surface area $4\pi r^2$. Because the intensity I_r of the reflected sound equals the total energy $\pi a^2 I_0$ reflected by the target sphere per unit time divided by $4\pi r^2$ over which it is distributed, then at a distance r from the sphere's center

$$I_r = \frac{\pi a^2 I_0}{4\pi r^2} = \frac{a^2}{4r^2} I_0.$$

But from Equation (15.28), $I_r = K I_0 / r^2$, where r represents the distance from the target to the measurement point,

$$K = \frac{a^2}{4}$$

and so

$$\text{TS} = 10 \log K = 20 \log(a/2).$$

References

Bezdek, H. H. 1973. Pressure dependence of sound attenuation in the Pacific Ocean. *Journal of the Acoustical Society of America* 53: 782.

Browning, D., Fechner, M., and Mellon, R. 1981. Regional dependence of very low-frequency sound attenuation in the deep sound channel. *Naval Underwater Systems Center Technical Document* 6561.

Colladen, J.-D. and Sturm, J. C. F. 1827. Memoir on the compression of liquids. *Ann. Chim. Phys*. 36: 113–159, 225–257.

Del Grosso, V. A. 1952. The velocity of sound in sea water at zero depth. *U.S. Naval Research Laboratory Report* 4002.

Del Grosso, V. A. 1974. New equations for the speed of sound in natural waters with comparisons to other equations. *Journal of the Acoustical Society of America* 56: 1084.

Hennin, C., and J. Lewiner. 1978. Condenser electret hydrophones. *Journal of the Acoustical Society of America* 63: 279.

Hucaro, J.A. et al. 1977. Fiber-optic hydrophone. *Journal of the Acoustical Society of America* 62: 1302.

Leonard, R. W., Combs, P. C., and Skidmore, I. R. 1949. Attenuation of sound in synthetic sea water. *Journal of the Acoustical Society of America* 21: 63.

Leroy, C.C. 1969. Development of simple equations for accurate and more realistic calculation of the speed of sound in sea water. *Journal of the Acoustical Society of America* 46: 216.

Liebermann, L. N. 1948. Origin of sound absorption in water and in sea water. *Journal of the Acoustical Society of America* 20: 868.

Liebermann, L. N. 1949. Sound propagation in chemically active media. *Physics Rev*iew 76: 1520.

Mackenzie, K. V. 1981. Nine-term equation for sound speed in the oceans. *Journal of the Acoustical Society of America* 70: 807.

Medwin, Herman. 1975. Speed of sound in water for realistic parameters. *Journal of the Acoustical Society of America* 58: 1318

National Defense Research Committee. 1946. Application of oceanography to subsurface warfare. *Div. 6 Summary Tech. Rep*., vol. 6A: Figures 17, 32.

Schulkin, M., and Marsh, H. W. 1962a. Absorption of sound in sea water. *Journal of the British IRE* 25: 293.

Schulkin, M., and Marsh, H. W. 1962b. Sound Absorption in sea water. *Journal of the Acoustical Society of America* 34: 864.

U.S. Navy. 1975. An Interim Report on the Sound Velocity Distribution in the North Atlantic Ocean. *U.S. Navy Oceangoing Office Technical Report 171*.

Urick, Robert J. 1962. Generalized form of the sonar equations. *Journal of the Acoustical Society of America* 34: 547.

Urick, Robert J. 1983. *Principles of Underwater Sound*, 3rd ed. New York: McGraw-Hill. (A classic text in the field by one of the leading researchers.)

Weissler, A., and Del Grosso, V. A. 1951. The velocity of sound in sea water. *Journal of the Acoustical Society of America* 23: 219.

Wilson, W. A. 1960. Speed of sound in sea water as a function of temperature, pressure and salinity. *Journal of the Acoustical Society of America* 32: 219. Also see extensions and revisions in *Journal of the Acoustical Society of America* (1960), 32: 1357 and (1962), 34: 866.

Wilson, O. B., and Leonard, O.B. 1954. Measurements of sound absorption in aqueous salt solutions by a resonator method. *Journal of the Acoustical Society of America* 26: 223.

Wood, A. B. 1941. *A Textbook of Sound*. New York: Macmillan Company; p. 261.

Yeager, E., Fisher, F. H., Miceli, J., and Bressel, R. 1973. Origin of the low-frequency sound absorption in sea water. *Journal of the Acoustical Society of America* 53: 1705.

Problems for Chapter 15

1. Determine the speed of sound in seawater at 20 m depth and at 20°C and 34 ppt using the Leroy formula.
2. Using the same conditions as in Problem 1, find the speed of sound on the basis of the Medwin formulation.
3. Do Problem 1 using the MacKenzie empirical expression.
4. Compare the velocities of sound in seawater at a depth of 5000 ft for the latitude of 61°N and the latitude of 19°N. Convert the depth to meters and speed of sound to km/s.
5. If the sound intensity 1 m from a source is 10 W/m^2 in the sea, find the transmission loss (TL) and the intensity 50 m from the source due strictly to spherical spreading. What other factors usually affect the transmission loss?
6. If sound is transmitted in a tube of water, what would be the transmission loss? In real life, would there be transmission loss and why?
7. Predict the value of the attenuation coefficient for seawater in which salinity $S = 40$ ppt, frequency of the signal is 50 kHz, and the temperature is 18°C.
8. Compare the spherical spreading over $r = 100$ yards with and without absorption by computing the transmission loss under the conditions of Problem 7.
9. Given a change of seawater sound velocity of 20 m/s for a depth of 150 m, what is the gradient and what would be the radius of curvature in terms of c_0?
10. If an element of an array has a signal to noise ratio of 40 dB, what would be the array gain of 25 similar elements in such an array?
11. A transducer is rated at a receiving response of –95 dB re 1 V. If an acoustic pressure of 3 μPa is applied, what will be the resulting voltage at the terminals?
12. A projector is driven with a 1.8 rms ampere current. Its response is rated at 120 dB re 1 μPa/A. Find the response in dB.

16
Ultrasonics

16.1 Introduction

As a subcategory of acoustics, *ultrasonics* deals with acoustics beyond the audio frequency limit of 20 kHz. Although ultrasonics has been employed for most of the twentieth century, the tempo of new and improved applications has reached virtually explosive proportions only in the past few years, particularly in medical diagnostics and therapeutics.

Applications of ultrasonics fall into two categories—*low intensity* and *high intensity*. Low intensity applications carry the purpose of simply transmitting energy through a medium in order to obtain information about the medium or to convey information through the medium. Nondestructive testing, medical diagnostics, acoustical holography, and measurements of elastic properties of materials fall into this category. Even marine applications are included in this category, despite the large energy input into operating sonar submarine detectors, depth sounders, echo ranging processors, and communication devices.[1]

High-intensity applications deliberately affect the propagation medium or its contents. Uses of high intensities include medical therapy and surgery, atomization of liquids, machining of materials, cleaning, welding of plastics and metals, disruption of biological cells, and homogenization of materials.

Human beings are not alone in the use of ultrasonics, and even in this respect they have been preceded by thousands of years by other species in the animal kingdom. Certain animals are capable of generating and detecting ultrasonic signals in order to locate and identify food, navigate their way through their environment, and detect danger. In fact, the study of these animals have helped and is still helping scientists to develop and improve techniques in the application of ultrasonic energy.

Bats are known to emit pulses in the 30–120 kHz range, and it has been hypothesized that the bats judge range by sensing the time delay between an emitted pulse and the echo. Small bats can fly at full speed through barriers constructed of 0.4 mm vertical wiring spaced only one wingspan apart. They are capable of

[1] Emsinger (1988) suggested that a third category not based on intensity be designated to specifically cover underwater applications.

catching small insects at the rate of one every 10 s for as long as half an hour. Their ability to discriminate between objects such as food and raindrops or foliage can be described as being nothing less than phenomenal; and yet when a large number of bats fly in close proximity to each other in a potentially confusing background of ultrasonic noise, they continue to locate prey and avoid collisions with each other.

The *Noctillio* bat of Trinidad catches small fish by dipping its feet below the surface of the water, after emitting a series of repetitive pulses. It has been conjectured that the characteristic ripples created by the fish are being detected by the bat rather than echoes from an object beneath the water surface. This is particularly remarkable in view of the fact that the sound must penetrate a barrier with a very high-reflection coefficient. As with bats, the echolocation of porpoises appears to be unaffected by the presence of interfering noises or jamming.

Moths, a prime target of bats, use ultrasonics for self-defense. When a moth detects a sonic pulse from a bat, it immediately takes evasive action through zigzagging in its flight and executing power dives. Moths can detect bats as far away as 13 m, and their ears can detect the cries of approaching bats at this distance, but when the bats are moving away their ears stop registering at about this distance. It was also observed by Roeder and Treat that when a bat makes a straight-on approach, it was observed to emit an uneven, sporadic signal; this was interpreted as an indication that the bat was counter-maneuvering by varying the intensity of its sonar pings.

Cetaceans constitute a group of sea mammals, which includes whales, dolphins, and porpoises. They are extremely intelligent as well as beautiful creatures, and are of the greatest interest to acousticians. Porpoise sounds have been described as whistling, barking, rasping, repetitive clicks or pulses, mewing. The cetaceans emit signals for the purpose of echolocation and communication with each other. Signals as high as 170 kHz have been observed in the clicks of porpoises, which vary in repetition rate from five clicks to several hundred per second. Extensive observations are being conducted to observe the emission patterns of different cetacean species, which vary from location to location at different times of the year even with the same group tracked on an almost continuous basis.

At least two species of birds, *Steatornia* and *Collocalla*, are known to be echolocators. Ornithologists have reported that calls by various birds may be comprehended only by members of their own local flocks. Gulls or crows will always respond to calls from members of their own flocks, but they may or may not respond to distress or assembly calls from gulls or crows from another region.

The hearing of dogs extends well beyond the frequency range of human hearing. The "silent" dog whistles that generate ultrasonic output can thus be used to summon a dog. Although dogs are generally endowed with especially keen senses of smell, hearing, and sight, there is no evidence that they make use of echolocation.

Certain effects become more evident in wave propagation through a medium as the acoustic signal extends into the ultrasonic range. The attenuation of the signal's amplitude occurs not only because of the spreading of the wave front, but also because of the conversion of the acoustical energy into heat and scattering

from irregular surfaces. The process of relaxation, which represents the lag between the introduction of a perturbation and the adjustment of the molecular energy distribution to the perturbation, requires a finite time; and the energy interchange approaches equilibrium in an exponential fashion. Considerable information regarding the nature of matter can be derived from the study of relaxation phenomena.

High-intensity ultrasound can result in energy absorption that yields considerable amount of heat, to the extent that glass or steel can be melted quickly. Ultrasonic waves can also generate stresses, resulting in cavitation in fluids. Cavitation is also capable of producing free chemical radicals, thus fostering specific chemical reactions. The stresses produced in the cavitation process are sufficiently concentrated to erode even extremely sturdy materials. Cavitation also provides the mechanism of ultrasonic cleaning.

16.2 Relaxation Processes

Relaxation entails molecular interactions in gases and liquids. These interactions affect absorption and velocity dispersion, both of which depend on frequency (and on pressure, in the case of gases). Chemical reactions also entail relaxation processes on their own, but they will not be considered in this chapter, except for the effect of ultrasound on reaction rates.

To better understand the phenomenon of relaxation, let us consider an ideal gas made up of diatomic molecules. The individual molecules move translationwise in three principal directions in a nonquantitized fashion, i.e., any translational energy is allowable. In addition, the molecules rotate about three perpendicular axes (actually two, since one axis has zero moment of inertia, hence zero rotational energy), and the molecules also vibrate along the direction of the bond joining the atoms. These molecules collide with one another in translation motion, exchanging energy among themselves. A single collision is usually sufficient to transfer translational energy from one molecule to another, but a certain period of time is needed to randomize the energy associated with excessive velocity in a particular direction. This amount of time is referred to as the *translational relaxation time* (Herzfeld and Litovitz, 1959), and it is given by

$$\tau_{tr} = \frac{\eta}{p} = 1.25\,\tau_c$$

where τ_{tr} is the translational relaxation time, p is the gas pressure, η is the viscosity of the gas, and τ_c is the interval between collisions. As the gas pressure is lowered, the rate of collisions decreases in the same proportion.

Unlike the case of translational motion, rotation and vibration are quantitized. When a collision occurs, a change in rotational or vibrational state will occur only when the change of energy of another state is sufficient to permit at least one quantum jump. For rotational energy transfer the spacing between energy levels is given by $2(J + 1)B$, where J denotes the rotational quantum number (which must be an integer), $B = \hbar^2/2I$, I is the effective moment of inertia and $\hbar$ is the

Planck constant 1.055×10^{-34} J s. Let us assume that J is the most probable value according to the Boltzmann distribution. The value of $2(J+1)B$ for a typical molecule (for example, O_2) in temperature units is about 1 K. This indicates that in a gas above 1 K all collisions will have enough translational energy to engender multiple changes, with the result that rotation rapidly equilibrates with translation. Hydrogen, however, constitutes an exception, because it has much larger rotational energy level spacing due to its small moment of inertia. According to Winter and Hill (1967), as many as 350 collisions may be necessary to transfer a quantum of rotational energy in the hydrogen molecule. As a pressure of 1 atm, this corresponds to a relaxation time of 2×10^{-8} s. It should be understood that a specific collision either does or does not transfer a quantum of rotational energy in the hydrogen molecule. The 350-collision average indicates that only one of the 350 collisions possesses the proper geometry and energy to execute a transfer of one quantum of rotational energy. The number of collisions necessary, on the average, to engender the transfer one quantum of rotational energy is termed the *collision number* Z. Where rotational energy is entailed, the collision number is written as Z_{rot}. The inverse of this dimensionless parameter represents the probability of transferring a quantum in a collision, symbolized by P_{rot}. Rotational energy levels are spaced unevenly, i.e., a $1 \rightarrow 2$ transition should be more probable than a $2 \rightarrow 3$ transition. The probability of these events occurring is distinguished by the symbols $P_{\text{rot}}^{1\rightarrow 2}$ or $P_{\text{rot}}^{2\rightarrow 3}$.

As a rule, the probability for transferring a quantum of energy through a collision drops off rapidly with the size of the quantum transferred. Because vibrational energy levels are much more widely spaced than rotational energy levels, the vibrational relaxation times are considerably longer than rotational relaxation times. Vibrational levels in single vibrational mode are virtually evenly spaced, so energy can therefore be exchanged between levels (e.g., the vibrational quantum number goes up in one molecule and goes down in another) with hardly any energy exchanged between vibration and translational. Thus, the vibration-to-vibration exchanges occur very rapidly. The amount of time it takes for energy to transfer between translation and the lowest lying vibrational level determines the vibrational relaxation time.

Because this energy level varies greatly for different molecules, the probability of vibrational energy transfer during a collision also varies greatly. In the case of N_1 molecules colliding with N_2 molecules, Z_{10} (the number of collisions needed to transfer energy from the lowest vibrational energy to translation) is approximately 1.5×10^{11}, so the relaxation time is close to 15 s (Zuckerwar and Griffin, 1980). Larger molecules possess very closely spaced vibrational energy levels, so fewer collisions are required to transfer a quantum of first-level vibrational energy into translational energy. For example, a relatively large molecule such as C_2H_6 needs to undergo about 100 collisions to execute this type of transfer.

Let us consider the case of a vibrational state that is excited to an energy E_v (the subscript v denotes vibration) that is greater than energy $E_v(T_{tr})$ that would exist in a Boltzmann equilibrium with translation. This excess vibrational energy will equilibrate with translational energy, in accordance with the following standard

expression for relaxation:

$$\frac{dE_v}{dt} = \frac{1}{\tau}\left[E_v - E_v(T_{tr})\right] \tag{16.1}$$

This reversion to equilibrium occurs through individual molecular collisions in which energy transfers result.

Let k_{10} be defined as the rate at which the molecules descend from the first excited state to the ground state, owing to collisions at 1 atm pressure. This is simply the collision frequency M multiplied by the probability of energy transfer $P^{1\to 0}$ multiplied by the mole fraction x_1 of the molecules in the first excited state. The reverse process will also occur, i.e., some molecules in the ground state will become excited at a rate k_{01}. In equilibrium both rates are equal, and we can write

$$k_{10}x_1 = k_{01}x_0$$

The energy is quickly shared from the first excited level of the vibrational mode to high-level modes through vibrational exchanges. On the basis of quantum mechanics governing the probabilities of energy exchanges between vibrational levels of a harmonic oscillator, Landau and Teller (1936)derived the following expression:

$$-\frac{dE_v}{dt} = k_{10}\left(1 - e^{-h\upsilon/kT}\right)\left[E_v - E_v(T_{tr})\right] \tag{16.2}$$

where h is the Planck constant (6.626×10^{-34} J s), υ is the vibrational frequency of the relaxing mode, k is the Boltzmann constant (1.38×10^{16} ergs K^{-1}), and T is the absolute temperature in K. Through comparison with Equation (16.1), the relaxation time τ for Equation (16.2) is found to be

$$\tau = \frac{1}{k_{10}(1 - e^{-h\upsilon/kT})} \tag{16.3}$$

Relaxation time and ultrasonic absorption and dispersion are interlinked. The relaxation process causes the specific heat of the gas to be frequency-dependent. The specific heat of a simple gas can be traced to translational, vibratory, and rotational contributions. Let us now consider the situation where any acoustically imposed temperature variation is followed by both translational and rotational energy equilibrating extremely quickly. The specific heat C' is deemed independent of temperature or the temperature never deviates appreciably from the equilibrium value T_0; so we can write

$$\left[E_v - E_v(T_{tr})\right] = C'(T' - T_0) \tag{16.4}$$

and insert this last expression into Equation (16.1) to yield

$$\frac{dT'}{dt} = \frac{1}{\tau}\left(T' - T_{tr}\right) \tag{16.5}$$

Now consider the case when the translational temperature is suddenly raised at time $t = 0$ from T_0 to a value T_1. Two cases can be ascertained: The external temperature is kept constant at T_1 after $t = 0$, the energy seeping from the outside into the internal (rotational and vibrational) degrees of freedom. The solution of

Equation (16.5) is

$$T' = T_1 + (T_0 - T_1)e^{-t/\tau}$$

But if the external energy is not constant and is lessened by the amount flowing into the internal degrees of freedom, we have

$$0 = \bar{C}(T_{tr} - T_1) + C'(T' - T_0) \tag{16.6}$$

where $\bar{C}$ refers to the specific heat arising from external degrees of freedom and C' denotes the specific heat belonging to the internal degrees. Equation (16.6) is rewritten as follows:

$$\begin{aligned} 0 &= \bar{C}(T_{tr} - T') + \bar{C}(T' - T_1) + C'(T' - T_0) \\ &= \bar{C}(T_{tr} - T') + C(T' - T_0) \end{aligned} \tag{16.7}$$

where

$$T_2 = \frac{\bar{C}}{C}T_1 + \frac{C'}{C}T_0$$

where C is the total specific heat, and T_2 the final equilibrium temperature. We eliminate $(T_{tr} - T')$ from Equation (16.5) with the use of Equation (16.7) to obtain

$$-\frac{dT'}{dt} = \frac{C}{\bar{C}}\frac{1}{\tau}\left(T' - T_2\right)$$

which yields the solution

$$T' = T_2 + (T_0 - T_2)e^{-t/\tau'}$$

The apparent relaxation time is

$$\tau' = \frac{\bar{C}}{C}\tau = \frac{C - C'}{C}\tau$$

Equation (16.5) can be written as

$$-\tau\frac{dT'}{dt} = T' - T_0 - (T_{tr} - T_0)$$

or

$$T' - T_0 + \tau\frac{d}{dt}(T' - T_0) = T_{tr} - T_0 \tag{16.8}$$

If $T_{tr} - T_0$ is periodic in time, i.e., it is proportional to $e^{i\omega t}$, T' —T_0 will likewise be proportional to $e^{i\omega t}$. After the transient dies out, then we have

$$T_{tr} - T_0 = (1 + i\omega t)(T' - T_{0)}$$

or

$$T' - T_0 = \frac{T_{tr} - T_0}{1 + i\omega\tau}$$

which can also be rewritten as

$$\frac{dT'}{dT_{tr}} = \frac{1}{1 + i\omega t}$$

The effective specific heat can be established from

$$dE = (C_v)_{eff}\, dT_{tr} = \bar{C}_v\, dT_{tr} + C' dT' = \left(\left(\bar{C}_v \right)_v + C' \frac{dT'}{dT_{tr}} \right) dT_{tr}$$

and

$$(C_v)_{eff} = \bar{C}_v + \frac{C'}{1 + i\omega\tau} = C_v - \frac{C' i\omega\tau}{1 + i\omega t} \tag{16.9}$$

The acoustic propagation constant k can be written in the form

$$\frac{k^2}{\omega^2} = \left(\frac{1}{c} - \frac{i\alpha}{\omega} \right)^2 = \frac{\rho_0 \kappa_T}{\gamma_{eff}}$$

where c represents the acoustic velocity, α is attenuation coefficient, ρ_0 is the equilibrium density, κ_T is the compressibility of the gas, and γ_{eff} is given by

$$\gamma_{eff} \equiv \frac{(C_v)_{eff} + R}{(C_v)_{eff}}$$

where R is the gas constant. In the case of this simple single relaxation, for $\alpha/\omega \ll 1$

$$\lambda a = \pi \left(\frac{c}{c_0} \right)^2 \frac{\varepsilon\omega\tau}{1 + (\omega\tau_s)^2} \tag{16.10}$$

and

$$\left(\frac{c_0}{c} \right)^2 = 1 - \frac{\varepsilon(\omega\,\tau_s)^2}{1 + (\omega\,\tau_s)^2} \tag{16.11}$$

where

$$\varepsilon = \frac{c_\infty^2 - c_0^2}{c_\infty^2}$$

c_0 = speed of sound for $\omega\tau s \ll 1$

c_∞ = speed of sound at frequencies $\gg$ relaxation frequency

λ = wavelength

The adiabatic relaxation time τ_s is related to the isothermal relaxation time τ as follows:

$$\tau_s = \frac{C_v + R}{C_v^\infty + R}\, \tau$$

The frequency at which the maximum absorption per wavelength occurs is called the *relaxation frequency*, symbolized by f_r. It is related to the adiabatic absorption time τ_s as follows:

$$f_r = \frac{1}{2\pi\tau_s} \frac{c_\infty}{c_0}$$

We take as an example the gas Fl_2 at 102°C in Figure 16.1 that shows curves for absorption per wavelength and velocity dispersion due to a single-relaxation process (Shields, 1962). Measured values are also plotted for the purpose of comparing with theory.

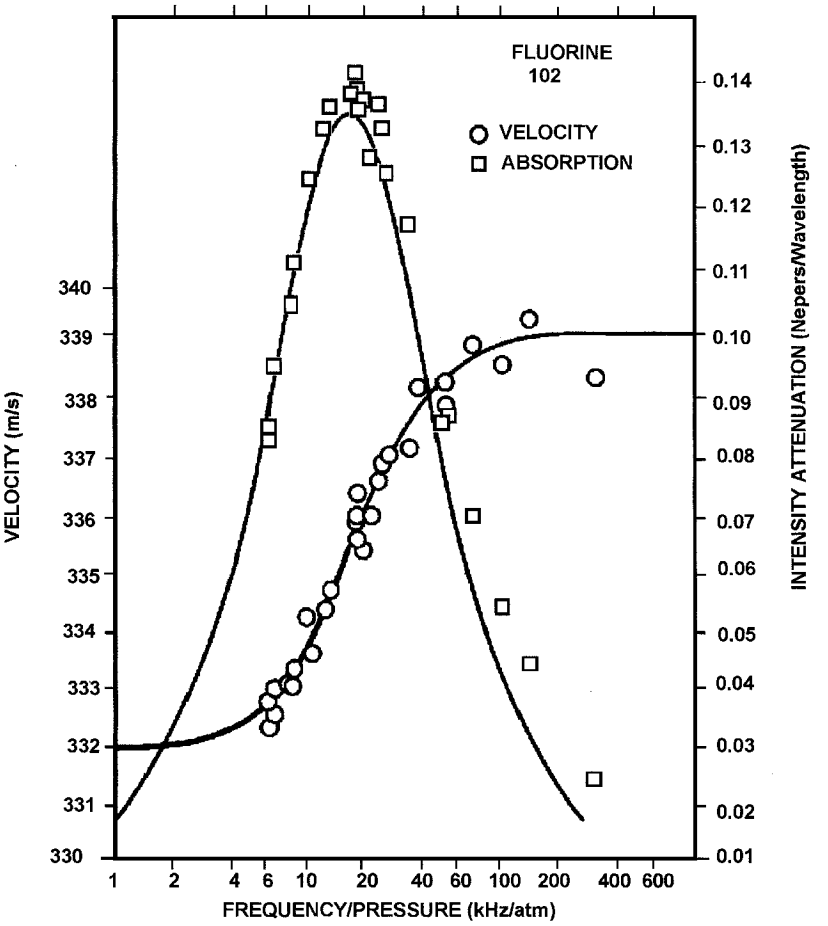


FIGURE 16.1. Sound absorption per wavelength and velocity dispersion in fluorine at 102°C.

In the case of polyatomic gases or mixtures of relaxing diatomic gases, the relaxing modes can be coupled by vibration-to-vibration exchanges. These multiple relaxation processes follow the general behavior given by Equations (16.10) and (16.11), but the magnitude of the absorption and dispersion and the associated relaxation frequencies assumes a different connotation. For this case of the multiple relaxing internal energy modes, Equations (16.10) and (16.11) are changed into the following formats:

$$\alpha \lambda \left(\frac{c}{c^{\infty}}\right)^2 = -\pi \sum_j \frac{\delta_j k_s / k_s^{\infty}}{1 + (\omega \tau_{s,j})^2} \tag{16.12}$$

$$\left(\frac{c}{c^{\infty}}\right)^2 = 1 + \sum_j \frac{\delta_j k_s / k_s^{\infty}}{1 + (\omega \tau_{s,j})^2} \tag{16.13}$$

where $\delta_j k_s / k_s^\infty$ is a relaxation adiabatic compressibility (which has a negative value) and j denotes that there may be more than a single-relaxation process entailed. In such complex cases, $\tau_{s,j}$ and $\delta_j k_s / k_s^\infty$ can no longer be respectively associated with a single-transfer reaction and relaxation energy of a specific mode, because the various modes and reaction pathways are coupled. So the sums in Equations (16.12) and (16.13) should cover all eigenvalues of the energy transfer matrix, which also accounts for all reactions. These two equations constitute the standard equations for calculating sound absorption in moist air as function of frequency and temperature. Equations (16.12) and (16.13) can also be used in the reverse manner, where the measured values of absorption and velocities can be used to derive the transition rates. But when the number of relaxation modes increases, the number of possible relaxation paths multiplies very rapidly, thus limiting this procedure to only a few special cases.

While relaxation processes similar to the energy exchanges in gases occur in liquids, there are important differences that arise from the greater density of molecules inherent in liquids and the consequential multibody interactions. The concept of a rate equation is less applicable, but the existence of relaxation time as a measure of the time for a system to revert to an equilibrium state in sustaining a perturbation remains valid.

In a few cases such as CS_2 and a number of organic liquids, the relaxation mechanism appears to be the same as that for gases, i.e., the internal energy of the individual molecules is excited by "collisions." These types of liquids are called *Kneser liquids*, and they generally have a positive temperature coefficient of absorption. With other liquids, the molecules bond temporarily to form large groups that reconfigure themselves when an ultrasound wave passes through. Such figurative relaxations tend to be very rapid, and there is the possibility of the frequency dependence of absorption and dispersion, which results in a distribution of relaxation times. Such liquids are termed *associated liquids*. Water is a prime example of such a liquid.

Chemical reactions complicate matters even more: in a reversible chemical reaction with heat of reaction ΔH. ΔH plays a role in the relaxation equations in the same manner as ΔE for vibrational relaxation. Because chemical reactions increase the possibility of the number density of the molecules changing, additional relaxation absorption and dispersion tend to occur.

16.3 Cavitation

The phenomenon of cavitation, the rupture of liquids, is readily observed in boiling water, turbines, hydrofoils, and in seawater in the vicinity of a ship's rotating propeller. It occurs in those regions of liquids that are subject to high-amplitude, rapidly vacillating pressures. Cavitation also occurs in a liquid irradiated with high-energy ultrasound.

Consider a small volume of liquid through which sound travels. During the negative half of the pressure cycle the liquid undergoes a tensile stress, and during

the positive half the liquid undergoes compression. Bubbles entrapped in the liquid will expand and contract alternatively. When the pressure amplitude is sufficiently great and the initial radius of the bobble is less than a critical value R_0 given by

$$R_0 = \frac{1}{\omega}\sqrt{3\,\gamma\,\frac{\left(p_0 + \dfrac{2T_{st}}{R_0}\right)}{\rho}} \tag{16.14}$$

the bubble collapses suddenly during the compression phase. In Equation (16.14), the symbols used are defined as follows:

ω = angular frequency of the signal
p_0 = hydrostatic pressure in the liquid
γ = ratio of the principal specific heats of the gas in the bubble
T_{ts} = surface tension at the surface of the bubble

This sudden collapse of bubbles constitutes the phenomenon of cavitation and it can result in the very sudden release of a comparatively large amount of energy. The severity of this cavitation, as measured by the amount of the energy released, depends on the value of the ratio R_m/R_0, where R_m denotes the radius of the bubble when it has expanded to its maximum size. Obviously this ratio depends on the magnitude of the acoustic pressure amplitude, i.e., the acoustic intensity.

The presence of bubbles facilitates the onset of cavitation, but cavitation can also occur in gas-free liquids when the acoustic pressure exceeds the hydrostatic pressure in the liquid. During a part of the negative phase of the pressure cycle, the liquid is in a state of tension. This causes the forces of cohesion between neighboring molecules to become opposed, and voids are formed at weak points in the structure of the liquid. These voids expand and then collapse in the same manner as gas-filled bubbles. The cavities produced in this fashion contain only the vapor of the liquid. Cavitation in a gasless liquid can be induced by introducing defects in the structure of the liquid by adding impurities or by bombarding the liquid with neutrons.

A hissing noise often accompanies the onset of cavitation. This noise is referred to as *cavitation noise*. Th minimum intensity or pressure amplitude required to establish cavitation is termed the *threshold of cavitation*. Figure 16.2 displays how the threshold intensity varies with both aerated and gas-free water. The threshold intensity is obviously considerably greater for gas-free water than for the aerated water. This parameter remains fairly constant up to about 10 kHz, then it undergoes a steady increase up to about 50 kHz and a more pronounced exponential increase beyond 50 kHz.

Generally speaking, the threshold intensity usually increases with increasing pressure and decreases with increasing temperature. But a number of exceptions to this rule exist (Hunter and Bolt, 1955). The threshold intensity decreases as the time of exposure to sound is increased. This is the result of a time delay between the acoustic excitation and the onset of cavitation. For pulsed waves, the threshold

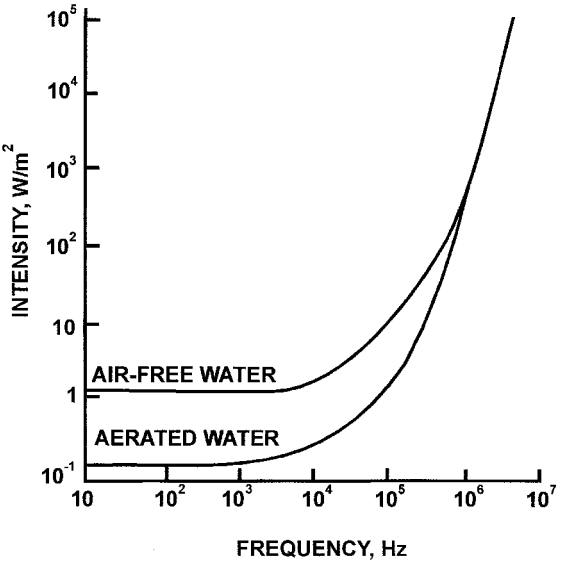


FIGURE 16.2. Variation of threshold intensity with frequency in aerated and gas-free water at room temperature (20°C).

intensity reduces in value as the pulse length is increased to an upper limit, beyond which it becomes independent of pulse length. This frequency-dependent upper limit would be of the order to 20 ms for a frequency of 20 kHz.

The amount of energy released by cavitation depends on the kinetics of the bubble growth and collapse of the bubbles. This energy should increase with surface tension at the bubble interface and lessen with the vapor pressure of the liquid. Water has a comparatively high surface tension, so it can be a very effective medium for cavitation. It can be made even more effective by the addition of 10% alcohol—this results in an appreciable increase in vapor pressure but at the cost of a decrease in the surface tension, but the former effect outweighs the latter effect.

Weak emission of light has been observed in cavitation. This phenomenon is known as *sonoluminescence*. Frenzel and Schultes first observed its effects in water in 1934 (Frenzel and Schultes, 1934). Two separate forms of sonoluminescence are thought to exist: multiple-bubble sonoluminescence (MBSL) and single-bubble sonoluminescence (SBSL). When a sufficiently strong acoustic field propagates through a liquid, placing it under dynamic stress, preexisting microscopic inhomogeneities serve as nucleation sites for liquid rupture. Most liquids such as water have thousands of potential nucleation sites per milliliter, so a cavitation field can harbor many bubbles over extended space. This cavitation, if sufficiently intense, will produce sonoluminescence of the MBSL type. It was more recently discovered that under certain conditions, a single, stable oscillating gas bubble can be forced into large amplitude pulsations that it produces sonoluminescence emission

on each and every cycle (Gaitan and Crum, 1992; Gaitan et al., 1992). This is the SBSL-type of sonoluminescence.

Sonochemistry deals with high-energy chemical reactions that occur during ultrasonic irradiation of liquids. The chemical effects of ultrasound do not result from direct molecular interactions but occur principally from the effects of acoustic cavitation. Cavitation provides the means of concentrating the diffuse energy of sound, with bubble collapse producing intense, local heating and high pressures that are extremely transient. Among the clouds of cavitating bubbles, the highly localized hot spots have temperatures of roughly 5000°K, pressures exceeding 2000 atm, and heating and cooling rates greater than 10^7 K/s. Ultrasonics can serve as useful chemical tool, as its chemical effects are diverse and it can provide dramatic improvements in both stoichiometric and catalytic reactions. In a number of cases, ultrasonic irradiation can increase reactivity by a million-fold.

The chemical effects can be categorized into three areas: (a) homogeneous sonochemistry of liquids, (b) heterogeneous sonochemistry of liquid–liquid or liquid–solid systems, and (c) sonocatalysis (which constitutes an overlap of the first two categories). Chemical reactions have generally not been observed in the ultrasonic irradiation of solids and solid–gas systems.

16.4 Phonons

In quantum mechanics, energy states are considered to occur only at discrete levels or *eigenstates*, not at any arbitrary values. In the analytical treatment of crystalline solids, the concept of phonons, often referred to as a "quantitized sound waves," is used to represent the effects of a transition between the eigenstates of a system of coupled quantum mechanical oscillators. Phonons generally apply to discrete strictly linear systems, while "classical" sound waves derive from continuous, intrinsically nonlinear systems within the limits of small amplitudes. Although phonons can occur in all states of matter, they are most easily discerned in crystalline solids.

In 1819, Dulong and Petit discovered the first evidence for phonons in solids when they observed that the specific heat of a solid is twice that of the corresponding gas. This finding suggested the fact that solids have a way of storing potential energy, in addition to the kinetic energy that is so apparent in gases. Einstein first postulated an acoustic theory of the specific heat of solids by assuming that the kinetic energy and potential energy arose from atoms oscillating about the equilibrium positions in the crystalline lattices. Applying the Planck quantum theory, Einstein related the energy to the frequency but he had made the simplifying assumption that each atom oscillated independently of the other, so his formula for the specific heat was therefore incorrect. Peter Debye in 1912 correctly inferred that these atomic oscillators are coupled, and later Max Born, Theodore von Kármán, and Moses Blackman refined the theory to the extent of matching the experimental results of the temperature dependence of the specific heat of solids.

Debye described the role of phonons in his explanation of thermal conductivity, and in that same year (1912) Frederick Landemond correlated lattice vibrations to thermal expansion and melting of solids, both of which are attributable to the nonlinearities of forces between atoms in solids.

Consider a one-dimensional array of masses m_j ($j = 0, \ldots, N+1$) interconnected with ideal springs of stiffness s_j ($j = 0, \ldots, N$). A spring s_j interconnects mass m_j and mass m_{j+1}. Let ξ_j denote the displacement of mass m_j. The displacements ξ_0 and ξ_{N+1} provide the boundary conditions at each end of the system. Within the boundaries, the motion of each mass is assumed to follow Hooke's law of linear elasticity and Newton's law:

$$m_j \frac{d^2\xi_j}{dt^2} = -s_{j-1}(\xi_j - \xi_{j-1}) + s_j(\xi_{j+1} - \xi_j) \tag{16.15}$$

The motion can be assumed amenable to Fourier analysis, so that it assumes a time dependence $e^{i\omega t}$. The left term of Equation (16.15) then becomes $-m_j\omega^2\xi_j$. The solution to this equation, through the application of the Bloch theorem for the normal modes of a coupled system, is a linear combination of the normal modes:

$$\xi_j = \sum_k \left(e^{ik(ja)}X_k + e^{-ik(ja)}X_{-k}\right)e^{i\omega_k t}$$

where

k = eigenvalues of the system determined by the boundary conditions

a = periodic spacing, or the "stretch" of the "springs" connecting the masses

$$X_k = \frac{\xi_1 - \xi_0 e^{-ika}}{2i\sin(ka)}$$

The values of ξ_0, ξ_1, X_k and the restrictions on k are established by boundary and initial conditions. For example, consider a system clamped at the ends, i.e., $x_0 = 0$, $x_{N+1} = 0$. Then

$$\xi_j = \sin[k(ja)]\left(\frac{\xi_1}{\sin(ka)}\right)$$

The eigenvalue k is quantitized with

$$k = \frac{n}{N+1}\frac{\pi}{a} \quad \text{with n} = 1, 2, \ldots, N$$

In an infinite system or in a system with periodic boundary conditions, it is readily established that $X_k = 0$ or $X_{-k} = 0$. If we apply a coordinate system with the mass m_0 located at its origin, then the location of the jth mass is $x = ja$, and we have

$$\xi_k(x) = e^{ikx}X_k \tag{16.16}$$

Equation (16.16) represents the customary Bloch wave result. The subscript k was added in Equation (16.16) to serve as a label for the normal mode. The solution is

complete as a linear combination of the normal modes, i.e.,

$$\xi_j = \sum_k (e^{ik(ja)} X_k + e^{-ik(ja)} X_{-k}) e^{i\omega_k t} \tag{16.17}$$

The normal modes are orthogonal, so the normalization constants can be obtained for any boundary conditions, whether they be clamped, periodic, and so on. The total energy E of the system can be found from

$$E = m \sum_k |X_k|^2 \omega_k^2$$

Coupled Quantum Particles

In dealing with harmonically coupled particles it is more expeditious to use a Lagrangian formulation rather than Newtonian formation of Equation (16.15). The Lagragian for such a set of connected particles is

$$L = \frac{1}{2} \sum_j m_j \left(\frac{d\xi_j}{dt} \right)^2 - \frac{1}{2} \sum_j s_j (\xi_{j+1} - \xi_j)^2$$

With the canonical momenta $p_j = m_j (d\xi_j / dt)$, the Hamiltonian becomes

$$H = \sum_j p_j \left(\frac{d\xi_j}{dt} \right) - L$$

The system is quantitized with the commutation relations

$$\left[\xi_j, p_{j'} \right] = i\hbar \delta_{j,j;} \tag{16.18}$$

where $\delta_{j,j;}$ represents the Dirac delta function that equals unity when $j = j'$ and zero when $j \neq j'$. We then assume a periodic system and set $m_j = m$ and $s = s_j$, and also make use of Equation (16.17) without the coefficient $e^{-i\omega_k t}$. We obtain the Hamiltonian

$$H = \frac{1}{2m} \sum_k P_k P_{-k} + \frac{1}{2} m \sum \omega_k^2 X_k X_{-k}$$

where

$$P_k = m \frac{dX_{-k}}{dt}, \qquad \omega_k = \omega_0 \sin \left(\frac{ka}{2} \right)$$

The above Hamiltonian could be used to construct a Schrödinger wave equation for a field $\psi(X)$, where X represents a point in the $2N$-dimensional X_k space and

$$P_k = -i\hbar \left(\frac{\partial \psi}{\partial X_k} \right)$$

Another approach is to construct the properties of the eigenfuctions and eigenvalues through the use of the commutation relations of Equation (16.18) for x_j and p_j.

Defining

$$a_k = \sqrt{\frac{m}{2\hbar\omega_k}} \left(\omega_k X_k + \frac{i}{m} P_{-k} \right)$$

and

$$a_k^* = a_{-k}^*, \quad \omega_k = -\omega_{-k}$$

the Hamiltonian assumes the form

$$H = \sum_k \hbar\omega_k \left(N_k + \frac{1}{2} \right)$$

where

$$N_k = a_k^* a_k$$

A general state of the system is constructed from a superposition of the eigenstates:

$$\psi = \sum_k \sum_n C_{kn} \psi_{kn}$$

where $|C_{kn}|^2$ represents the probability that the system is in the state ψ_{kn}. From the customary quantum mechanical relation $E\psi = H\psi$, the expectation value of the total energy of the system is

$$\langle \psi \, | E | \, \psi \rangle = \sum_k \sum_n |C_{nk}|^2 \hbar\omega_k \left(n_k + \frac{1}{2} \right)$$

The expectation value of the square of the momentum operator is found from

$$\langle \psi \left| P^2 \right| \psi \rangle = m \sum_k \langle \psi \, | X_k | \psi \omega_k^2 = \sum \sum |C_{nk}|^2 \hbar\omega_k n_k = \langle \psi \, | E | \, \psi \rangle - E_0$$

where E_0 is the zero-point energy.

From the expectation state of the position operator, a state ψ_{nk} can be coupled only to the state $\psi_{(n+1)k}$ or $\psi_{(n-1)k}$. Consequently the time dependence will behave as $2\cos(\omega_k t)$. The term *phonon* can be defined in the following manner: a phonon is emitted or absorbed when a system of harmonically coupled quantum mechanical particles executes a transition from a state ψ_{nk} to a state $\psi_{(n-1)k}$ (which results in emission) or $\psi_{(n+1)k}$ (which results in absorption).

The absorption of ultrasonic waves in solids are attributable to a number of different causes, each one of which is characteristic of the physical properties of the material concerned. They can be classified as (a) losses characteristic of polycrystalline solids, (b) absorption due to lattice imperfections, (c) absorption in ferromagnetic and ferroelectric materials, (d) absorption due to electron–phonon interactions, (e) absorption due to phonon–phonon interactions, and (f) absorption due to other possible causes. It is also interesting to note that a rapid decrease in attenuation occurs at the critical temperature for superconductivity. This variation of attenuation with temperature has been explained in a Noble-prize winning paper by Bardeen, Cooper, and Schrieffer through the B.C.S. theory (so-called after the

initial of their surnames) that predicts a temperature-dependent energy gap 2ε wide around the Fermi level at the critical temperatures of T_c and less (Bardeen et al., 1957). As the temperature is reduced, the gap increases toward a maximum at zero absolute temperature, where the predicted value of ε is equal to $1.75 \times \kappa\, T_c$, where κ is the Boltzmann constant. In the realm of $l_e > \lambda/2\pi$, the B.C.S. theory predicts that

$$\frac{\alpha_s}{\alpha_n} = \frac{2}{e^{\varepsilon/\kappa T} + 1}$$

where l_e denotes the mean-free path of an electron, α_s and α_n represent the values of absorption in the super-conducting and normal state, respectively, at absolute temperature T. This variation was confirmed experimentally by Morse (1959) and Bohm for indium at 28.5 MHz. Gibbons and Benton measured the velocities of longitudinal waves in both normal and superconducting tin; they found a very small reduction in velocity (about 1/500,000th) for the superconducting state. Application of a magnetic field to a metal at low temperatures affects the mean-free path and thus affects acoustic attenuation. For $l_e < \lambda/2\pi$, a decrease in the attenuation with increasing magnetic field H has been observed experimentally (Sternberg, 1958).

16.5 Transducers

A *transducer* is a device that generates and receives (i.e., it senses) sound waves. The transducer essentially functions as an energy converter. That is, it converts acoustical energy into or from other forms of energy (e.g., electrical, mechanical, or thermal). A transducer is said to be *reversible* if it can convert in either direction. Most high-intensity ultrasonic generators in use are basically crystal oscillators or magnetostrictive devices.

Transducers fall into the following categories: (1) crystal oscillators that operate through the piezoelectric effect, which is reversible; (2) magnetorestrictve devices based on the magnetorestriction phenomenon, which is also reversible; (3) mechanical generators and receivers that includes whistles and sirens acting as generators and also radiometers and Rayleigh discs serving as receivers; (4) electromagnetic transducers that operate on the same principle as the customary audio loudspeaker but can function only in the lower range of ultrasonic frequencies; and (5) miscellaneous types such as chemical, thermal, and optical transducers. In addition there are ultrahigh frequency transducers that work in the megahertz and gigahertz ranges. In this chapter we concentrate on the first two categories that constitute a major portion of the transducer types currently in principal use.

Ultrasonic receivers fall into two categories: (1) receivers that terminate ultrasonic paths of propagation and (2) receivers that serve as probes. Receivers that terminate ultrasonic paths have cross-sectional dimensions extending over several wavelengths, with the result that the presence of such a transducer will materially affect the acoustic field mainly through reflections. The purpose of an ultrasonic

probe is to gauge the characteristics of an acoustic field, so its dimensions must be sufficiently small so as not to affect the field. A probe diameter is typically only about one-tenth of the wavelength.

Piezoelectric Crystals

In 1880, the Curie brothers discovered that when a crystal having one or more polar axes or lacking axisymmetry is subjected to mechanical stress, an electrical potential difference occurs. Consider a segment of such a crystal, in the form of a slab or a disk, that is cut with its parallel surface running normal to a polar axis. When this segment undergoes a mechanical stress, equal and opposite electric charges arise on the parallel surfaces. The magnitude of the charge density (i.e., dielectric polarization) is directly proportional to the applied stress, provided the applied stresses do not strain the crystal beyond its elastic limit.

The opposite effect, predicted by Lippmann in 1881 and verified experimentally by the Curie brothers the same year, occurs when an electric field is applied in the direction of a polar axis, causing a mechanical strain in the crystal segment. The amount of strain is directly proportional to the intensity of the applied electric field. From the viewpoint of the principle of conservation of energy, the piezoelectric effect and its converse may be deemed to be equal and opposite. Such effects occur in crystals such as quartz (a member of the trigonal system, as shown in Figure 16.3), Rochelle salt, and lithium sulfate.

Quartz is very commonly applied for ultrasonic generation. A quartz crystal is shown in Figure 16.3, with a hexagonal cross section normal to the nonpolar

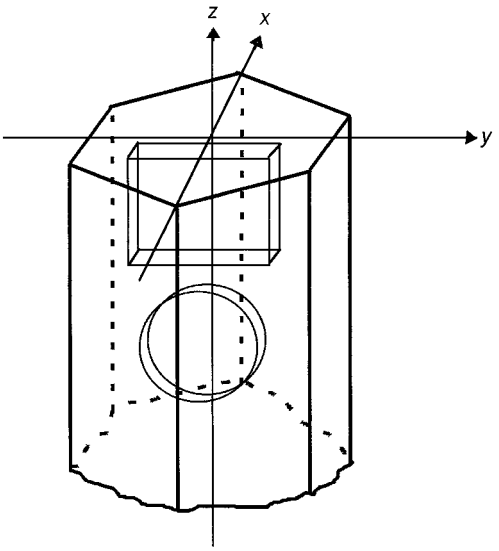


FIGURE 16.3. A hexagonal quartz crystal with x-cut rectangular and circular plates.

optic axis, denoted by the z-axis. The axes joining opposite edges are designated as x-axes, and the associated axes, which are perpendicular to these and joining opposite faces are termed y-axes. The x- and y-axes are *polar axes*, and slabs cut with their faces perpendicular to them manifest the piezoelectric effect. Crystals which are cut with their faces perpendicular to an x-axis or y-axis are termed *x-cut* and *y-cut* crystals, respectively. The x-cut crystals are generally utilized to propagate compression waves, and the y-cut crystals are applied to generate shear waves.

Now consider an x-cut crystal in the form of a rectangular prism shown in Figure 16.3. Applying an electric field along the x-axis produces compression in that direction, while expansion occurs simultaneously along the y-direction. If the direction of the field is reversed, expansion occurs along the x-axis with an associated compression along the y-axis. No strain, however, occurs along the z-axis. If a pair of surfaces normal to either of the polar axes (x- and y-axes) is coated with a conductive material to form electrodes, small-amplitude oscillations will result when an alternating voltage of frequency f is applied across them. When the frequency f equals one of the natural frequencies of mechanical vibration for a particular axis, the response amplitude jumps to a considerably higher value. Crystals are generally operated at resonant frequencies for either "length" or "thickness" vibrations, as denoted by the resonance occurring in the direction parallel with or normal to the radiating surfaces, respectively. The natural frequency for mechanical vibrations is proportional to the inverse of the dimension along which they occur, so it becomes obvious the lower frequencies are generated by "length" vibrations along the direction of the longer dimension whereas the higher frequencies are produced by "thickness vibrations" along the direction of the smaller dimension.

Maximum acoustic intensities are obviously obtained by operating at the fundamental natural frequencies. But material constraints in crystals may necessitate the use of higher harmonics to obtain higher frequencies. For example, an x-cut quartz plate can be only 0.15 mm thick in order to generate a fundamental "thickness" mode for 20 MHz. Such a quartz plate is extremely brittle and it can shatter under the impetus of a exceedingly high-applied voltage, or its dielectric properties may break down. To avoid this situation, it is customary to use thicker slabs of crystals with lower resonance frequencies and operate at one of the upper harmonics. An example is the vibration of a 1-cm thick quartz crystal at its 191st harmonic to generate 55 MHz ultrasound.

The piezoelectric effect occurs only when opposite charges appear on the electrodes, and for that reason, only odd harmonics can be generated. At the nth harmonic, the thickness of the crystal is divided into n equal segments with compressions and expansions alternating in adjacent sections, as illustrated in Figure 16.4. For even harmonics in the nth mode, compressions occur in $n/2$ segments and expansions occur in the other $n/2$ segments, with the result no net strain exists in the crystal. When n is odd, the $(n–1)/2$ compressions offset the same number of expansions, leaving either a compression or an expansion in the remaining segment.

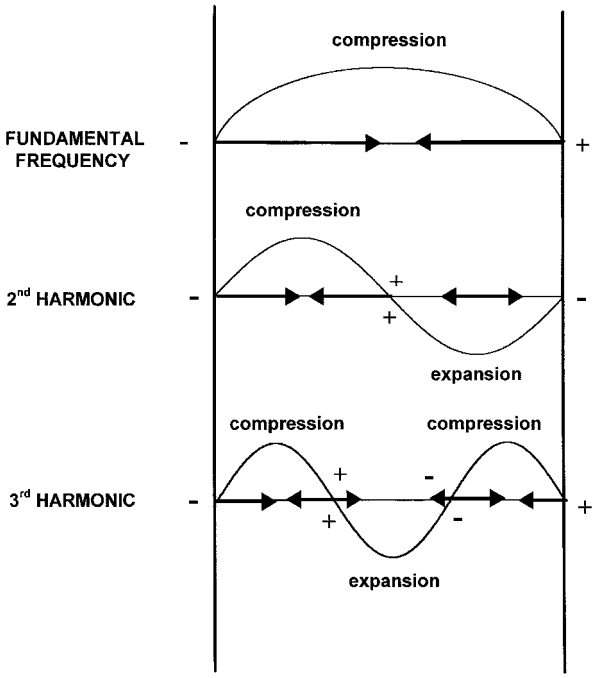


FIGURE 16.4. Crystal divided into segments.

The Electrostrictive Effect

The electrostrictive effect, which is the electrical analog of the magnetostrictive effect discussed in a later section, occurs in all dielectrics, but it is not a very pronounced phenomenon in most materials except for a certain class of dielectrics. The effect is much more apparent in this class namely, the *ferroelectrics*. An electric field applied along a given direction produces a mechanical strain. The magnitude of the strain is proportional to the square of the strength of the applied electric field and is therefore independent of the sense of the field. A positive strain may thus result for both positive and negative values of the excitation field. For a sinusoidally varying electric field, the waveform of the strain assumes that of a rectified but unsmoothed alternating current, and its frequency is twice that of the applied field.

It is possible to obtain a sinusoidal variation in the strain, and this is done by permanently *polarizing* the transducer, namely, one that has magnetostrictive properties. The transducer is heated to a temperature above the Curie point, causing the magnetostrictive effect to vanish, and then it is cooled slowly in a strong direct field oriented in the direction along which it is intended to apply the exciting field. If the exciting field is kept small compared with the initial polarizing field, the strain should vary sinusoidally at the frequency of the exciting field. Because a polarized ferroelectric transducer appears to manifest the same effect as a piezoelectric

transducer, it has mislabeled as being "piezoelectric." Among the principal ferroelectrics, barium titanate, lead meta-niobate, and lead zirconate titanate are greatly used for electrostrictive applications. To construct this type of transducer, many small crystallites of ferroelectric substances are bonded together to form a ceramic of the appropriate shape. Because such materials are polycrystalline, they may be considered as being isotropic and thus do not have to be cut along specific axes. This renders possible the construction of a concave transducer so that the ultrasonic radiation can be focused without the need for an auxiliary lens system.

Fundamental Piezoelectric Relationships

For a given temperature, consider a piezoelectric element having cross-sectional area A, thickness t, with electrodes attached to the opposite faces. A voltage V is applied across the electrodes to generate an electric field $E = V/t$, and a constant tensile stress σ is applied to the surfaces. Within the elastic limits, the resultant mechanical strain s relates to the stress as follows:

$$s = a\sigma + bE \tag{16.19}$$

and we also have

$$D = cE + d\sigma \tag{16.20}$$

where D represents the electric displacement and a, b, c, d are coefficients defined below.

Let us now short circuit the electrodes so that $E = 0$. Equation (16.20) now reads

$$D = d\sigma$$

We note here that the electric displacement D equals the dielectric polarization P, or the charge per unit area. Hence

$$P = d\sigma \tag{16.21}$$

under short-circuit conditions. The coefficient d constitutes the *piezoelectric strain constant*, which is defined as the charge-density output per unit applied stress under the conditions of short-circuited electrodes. Now if the stress σ is reduced to zero, Equation (16.19) modifies to

$$\sigma = bE$$

The principle of conservation of energy dictates that $b = d$, resulting in

$$\sigma = dE \tag{16.22}$$

for the no-load condition. The coefficient d may also be described as the mechanical strain produced by a unit applied field under the conditions of no loading and it is expressed either in units of coulomb/newton (C/N) or meter/volt (m/V). Equation (16.19) becomes altered to

$$s = a\sigma + dE \tag{16.23}$$

If no piezoelectric effect is present, the term d vanishes from Equations (16.20) and (16.23), which yields the familiar relationships

$$s = a\sigma = \sigma / Y \tag{16.24}$$

and

$$D = \varepsilon E \tag{16.25}$$

where Y is the elastic constant (or Young's modulus) for the material and ε is the corresponding electrical permittivity

Under short-circuit conditions for the crystal, Equations (16.21) and (16.24) lead to

$$P = es \qquad \text{(for short-circuit conditions)} \tag{16.26}$$

where $e = d/Y$.

When a compressive stress is applied to the crystal, Equation (16.23) becomes

$$s = -a\sigma + dE \tag{16.27}$$

When the crystal is clamped to keep strain zero and when stress is applied, we see from Equations (16.26) and (16.27) that

$$\sigma = eE \qquad \text{(for the constraint } s = 0) \tag{16.28}$$

where e is the piezoelectric stress constant which is expressed on C/m^{2}or N/V m.

It must be realized at this stage that the piezoelectric phenomenon is a three-dimensional one. Not only we have to consider the changes in voltage, stress, strain, and dielectric polarization in the thickness direction of the crystal, we must also take into account the effects in any direction. A stress applied to a solid in a given direction may be resolved into six components: three tensile stresses σ_x, σ_y, σ_z along the principal axes x, y, and z, respectively, and three shear stresses τ_{yz}, τ_{xz}, and τ_{yz} about axes x, y, and z. In the notation for shear, the subscripts indicate the action plane of the shear—thus yz denotes a shear in the yz plane acting about the x-axis. Also, we note that $\tau_{yz} = \tau_{zy}$, and so on. The corresponding components of strain are ε_x, ε_y, ε_z, ε_{yz}, ε_{xz}, and ε_{yx}. In general, the following stress–strain relationship of Equation (5.2) can be generalized as follows

$$\sigma_{ji} = c_{jk}\varepsilon_{ij} \tag{16.29}$$

where c_{jk} denotes the elastic modulus or stiffness coefficient. This yields 36 values of c_{jk}:

$$\left.\begin{aligned}
\sigma_{xx} &= \sigma_x = c_{11}\varepsilon_x + c_{12}\varepsilon_y + c_{13}\varepsilon_z + c_{14}\varepsilon_{yz} + c_{15}\varepsilon_{xz} + c_{16}\varepsilon_{xy} \\
\sigma_{yy} &= \sigma_y = c_{21}\varepsilon_x + c_{22}\varepsilon_y + c_{23}\varepsilon_z + c_{24}\varepsilon_{yz} + c_{25}\varepsilon_{xz} + c_{26}\varepsilon_{xy} \\
\sigma_{xx} &= \sigma_z = c_{31}\varepsilon_x + c_{32}\varepsilon_y + c_{33}\varepsilon_z + c_{34}\varepsilon_{yz} + c_{35}\varepsilon_{xz} + c_{36}\varepsilon_{xy} \\
\tau_{yz} &= c_{41}\varepsilon_x + c_{42}\varepsilon_y + c_{43}\varepsilon_z + c_{44}\varepsilon_{yz} + c_{45}\varepsilon_{xz} + c_{46}\varepsilon_{xy} \\
\tau_{xz} &= c_{51}\varepsilon_x + c_{52}\varepsilon_y + c_{53}\varepsilon_z + c_{54}\varepsilon_{yz} + c_{55}\varepsilon_{xz} + c_{56}\varepsilon_{xy} \\
\tau_{xy} &= c_{61}\varepsilon_x + c_{62}\varepsilon_y + c_{63}\varepsilon_z + c_{64}\varepsilon_{yz} + c_{65}\varepsilon_{xz} + c_{66}\varepsilon_{xy}
\end{aligned}\right\} \tag{16.30}$$

TABLE 16.1. Value of the Adiabatic Elastic Constants for Quartz $\times 10^{10}$ dyne/cm^2 or $\times 10^9$ N/m^2.

c_{11}	= 87.5
c_{33}	= 107.7
c_{44}	= 57.3
c_{12}	= 7.62
c_{13}	= 15.1
c_{14}	= 17.2

Because $c_{mn=}c_{nm,}$ the number of these constants is reduced from 36 to 21. Symmetry of the axes will lessen this number even further. In the case of quartz, only six elastic constants are independent of one another, and the values of c_{mn} can be expressed as a matrix as follow:

$$\begin{vmatrix} c_{11} & c_{12} & c_{13} & c_{14} & 0 & 0 \\ c_{13} & c_{11} & c_{13} & -c_{14} & 0 & 0 \\ c_{13} & c_{13} & c_{23} & 0 & 0 & 0 \\ c_{14} & -c_{14} & 0 & c_{14} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44} & c_{14} \\ 0 & 0 & 0 & 0 & c_{14} & \dfrac{(c_{11} - c_{12)}}{2} \end{vmatrix}$$

The values of these constants for adiabatic conditions are given in Table 16.1

The Dynamics of Piezoelectric Transducers

A body undergoing forced vibrations can be considered analogous to an electric circuit that is activated by an electromotive force, with the current i corresponding to body velocity u and the voltage V to the applied force F. In terms of the strain s, velocity $u = l(ds/dt)$, and the current is expressed as $i = (dQ/dt) = A(dP/dt)$. Here l is the body length, Q is the electric charge, A is the cross-sectional area. We invoke Equation (16.26) to obtain

$$\frac{dP}{dt} = e\frac{ds}{dt}$$

which gives

$$i = \frac{Ae}{l}u = \alpha_T u \tag{16.31}$$

where the transformation factor $\alpha_T = Ae/l$, which constitutes a characteristic constant for a specific transducer. Because P has three components and S has six components, we can write Equation (16.31) in the general matrix form

$$\mathbf{i} = \mathbf{a_T u}$$

From Equation (16.28) we have

$$F = \frac{Ae}{l} V = \alpha_T V$$

or

$$\mathbf{F} = \mathbf{a_T V}$$

The mechanical compliance C_m is analogous to the electrical capacitance C, i.e.,

$$C_m = \frac{sl}{F} = \frac{sl}{\sigma A} = \frac{Yl}{A}$$

When an applied force F causes a strain s, mechanical energy W_m is stored in the transducer, according to

$$W_m = \frac{1}{2} Fsl = \frac{1}{2} F^2 C_m = \frac{1}{2} \alpha_T^2 V^2 C_m = \frac{1}{2} C V^2 \tag{16.32}$$

where

$$C = \alpha_T^2 C_m$$

The electrical capacitance C_e between the electrodes of the transducer follows the relationship:

$$C_e = \frac{\varepsilon A}{l} \tag{16.33}$$

The corresponding electrical energy W_e is equal to $\frac{1}{2} C_e V^2$. From Equations (16.32) and (16.33) the ratio of mechanical energy stored in a piezoelectric transducer to the electrical energy provided to it is given by

$$\frac{W_m}{W_e} = \frac{C}{C_e} = \alpha_T^2 \frac{C_m}{C_e} = k_e^2$$

Here the *electromechanical coupling factor* k_e^2 constitutes a measure of the efficiency of the transducer.

The Q Factor

The Q factor of either a mechanical or an electrical system determines the contour of the frequency response curve for that system. A low value of Q results in a resonance spreading over a wide frequency band. At higher values of Q, a resonance will be confined to a considerably narrower frequency band. Two Q factors exist in a transducer, one mechanical and the other electrical, denoted by Q_m and Q_e, respectively. The mechanical Q factor is defined by

$$Q_m = \frac{m\omega_r}{R_m}$$

where ω_r represents the resonance frequency of the transducer, m its mass and R_m the mechanical resistance. In the simplest case for the radiating transducer surface,

the mechanical resistance is given by

$$R_m = \frac{\sigma A}{u} = \rho c A$$

in terms of the stress, radiating surface velocity u, material density ρ, and sound propagation speed c. For the electrical Q factor, electrical capacitance C_e between transducer electrodes must be taken into consideration. At resonant frequencies the only effective mechanical impedance is R_m. Thus

$$Q_e \cong C_e \omega_r R = C_e \omega_r \frac{R_m}{\alpha_T^2} = \frac{\pi^2/2k_e^2}{Q_m}$$

Magnetostrictive Transducers

Magnetostriction occurs in ferromagnetic materials and certain nonmetals that are termed ferrites. When a magnetic field is applied, a bar of ferromagnetic or ferrimagnetic material undergoes a change in length. Conversely, a mechanical stress applied to the bar will cause a change in intensity of magnetization. The former effect was discovered by Joule in 1847 and the converse effect by Villari in 1868.

Magnetostriction occurs prominently in materials such as iron, nickel, and cobalt. Whether there occurs an increase or decrease in length fully depends on the nature of the material as well as on the strength of the applied magnetic field. The change in length does not depend on the direction of the magnetic field. The magnitude of the strain varying as a function of the applied magnetic field is shown in Figure 16.5 for four different materials, viz. cast cobalt, permendur, nickel, and iron. Figure 16.6 shows strain varying as a function of magnetic polarization. The magnetostrictive effect generally decreases with a rise in temperature and disappears altogether at the Curie temperature.

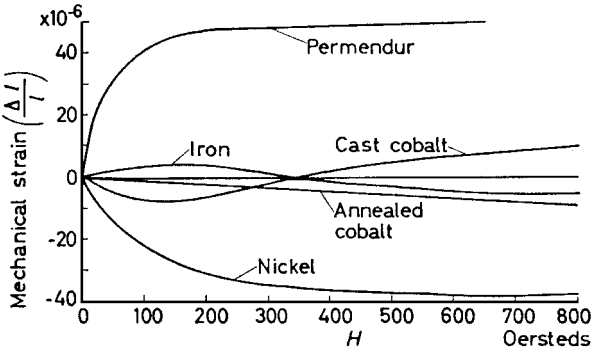


FIGURE 16.5. Mechanical strain as a function of magnetic field.

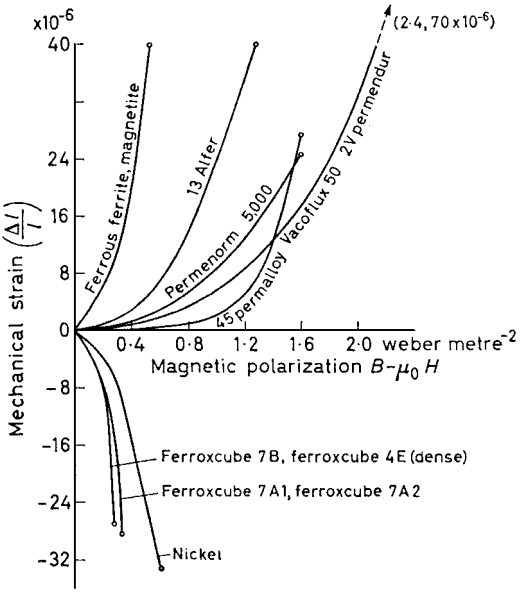


FIGURE 16.6. Mechanical strain as a function of magnetic polarization.

When a sinusoidally varying magnetic field is applied in the direction of the axis of a bar of ferromagnetic material, the bar will oscillate at double the frequency of the applied field. In accordance with the relevant curve in Figure 16.6, a decrease in length occurs when a field is applied to nickel, regardless of the sense of the field. A negative strain occurs every half cycle. The waveform of the strain occurs as a rectified sine curve, with the result that unwanted harmonics may be generated. A purely sinusoidal wave corresponding to the frequency of the applied field, along with a markedly increased energy output, will be obtained if the bar is polarized. This is achieved by simultaneously applying the alternating field and a direct magnetic field of sufficiently high intensity for the value of the resultant field to remain above zero.

The maximum output for magnetostriction occurs by operating at the fundamental frequency f_r of the bar, given by

$$f_r = \frac{1}{2L}\sqrt{\frac{Y}{\rho}} \tag{16.34}$$

where Y is the Young's modulus for the bar material, ρ is the density of the material, and L is the length of the bar. The term $\sqrt{Y/\rho}$ in Equation (16.34) also happens to be the propagation velocity of sound in the material. At resonant frequencies the mechanical strains reaches the order of 10^{-4} rather than magnitudes in the order of 10^{-6} that prevails in operating at nonresonant frequencies.

The Physics of Magnetostrictive Transducers

Magnetrostriction theory is highly analogous to that of piezoelectricity, but, in this case, account is taken of the polarizing field H_0. We now consider a ferromagntic rod undergoing polarization throughout its length with a magnetic field H_0, with B_0 denoting the associated flux density. The resultant strain ε_0 is directly proportional to the square of the flux density, i.e.,

$$\varepsilon_0 = C B_0^2 \tag{16.35}$$

where C is a constant. It is seen from Equation (16.35) that the sign of the resultant strain is independent of the direction of the field. Now if we apply an exciting magnetic field of strength H, which is appreciably less than $H_0$with an associated flux density B, we can write

$$B = \mu_i H = \Delta B_0 \ll B_0 \tag{16.36}$$

where we have denoted μ_i as the incremental magnetic permeability. For a state of constant stress, we have for the resulting strain by differentiation of Equation (16.35):

$$\varepsilon = \Delta\varepsilon_0$$
$$\Delta\varepsilon = 2C B_0 \Delta B_0$$

or

$$\varepsilon = 2C B_0 B = \beta\mu_i H \tag{16.37}$$

Here $\beta = 2C B_0$ constitutes the *magnetostrictive strain coefficient* (given in units of m^2/weber) that applies to small strains. Equation (16.36) is analogous to Equation (16.22) for no-load conditions. When no alternating field is applied, the value of the strain ε is given by Hooke's law,

$$\varepsilon = \sigma / Y \tag{16.38}$$

in terms of stress σ and Young's modulus Y. We can now obtain the analog to Equation (16.23) by using H instead of the electric field E, Y instead of $1/a$, and $\beta\mu_i$ instead of d. This gives us

$$\varepsilon = \frac{\sigma}{Y} + \beta\mu_i H \tag{16.39}$$

for a rod undergoing simultaneously a tensile stress σ and a magnetic field H. The analogy extended to Equation (16.20) yields

$$B = \sigma\beta\mu_i + \mu_i H \tag{16.40}$$

Clamping the rod causes strain ε to be zero in Equation (16.38), yielding

$$\sigma = Y\beta\mu_i H = \Lambda B$$

where σ denotes the compressive stress and $\Lambda = Y\beta$ represents the *magnetostrictive stress constant* (given in units of newton/weber). The reciprocal of Λ, preferred by some authors, is called the *piezomagnetic constant* (units of weber/newton).

16.6 Transducer Arrays

A single-element ultrasound transducer tends to radiate a rather narrow beam or receive signals over a narrow spatial range. In order to cover a wider area through a process called *scanning*, and, in many instances, to emit more powerful signals than is possible with a single element, an especially arranged group of transducers or *arrays* are used to extend the versatility of transducers.

Array transducers are also used to focus an acoustic beam. Variable delays are applied across the transducer aperture. The delays are electronically controlled in a sequential or phased array and can be changed instantaneously to focus the beam on different areas.

With linear-array transducers, which are far more versatile than piston transducers, the electronic scanning entails no moving parts, and the focal point can be changed readily to any position in the scanning plane. A broad variety of scan formats can be generated, and received echoes can be processed for other applications such as dynamic receive focusing, correction for phase aberrations, and synthetic aperture imaging. The principal disadvantages of linear arrays obviously lie in the greater complexity and increased costs of the transducers and scanners. In order to ensure high quality imaging, many (as high as 128 and on the increase) identical array elements are required. Each array element tends to be less than 1 mm on one side and is connected to its own transmitter and receiving electronics.

Phased Arrays: Focusing and Steering

We examine how a phase-array transducer can focus and steer an acoustic beam along a specified direction. An ultrasound image is created by repeating the scanning process more than a hundred times to probe a two-dimensional (2D) or a three-dimensional (3D) locale in the medium. In Figure 16.7(a), a simple six-element array is shown focusing the transmitted beam. Each array element may be considered a point source that radiates a spherical wavefront into the medium. Because the topmost element is the farthest away from the focus in this example, it is activated first. The other elements are triggered progressively at the appropriate time so that the acoustic signals from all the elements reach the focal point simultaneously. According to Huygens' principle, the resultant acoustic signal constitutes the sum of the signals that came from the source. The contributions from each element add in-phase at the focal point to yield a peak in the acoustic signal. Elsewhere, some of the contributions add out-of-phase, lessening the signal relative to the peak.

On receiving an ultrasound echo, the phased array works in reverse. An echo is shown in Figure 16.7(b) originating from receive focus 1. The echo is incident on each array element at a different time interval. The received signals undergo electronic delay so that they add in phase for an echo originating at a focal point. Echoes originating elsewhere have some of their signals adding out of phase, thereby reducing the receive signal relative to the peak at focus.

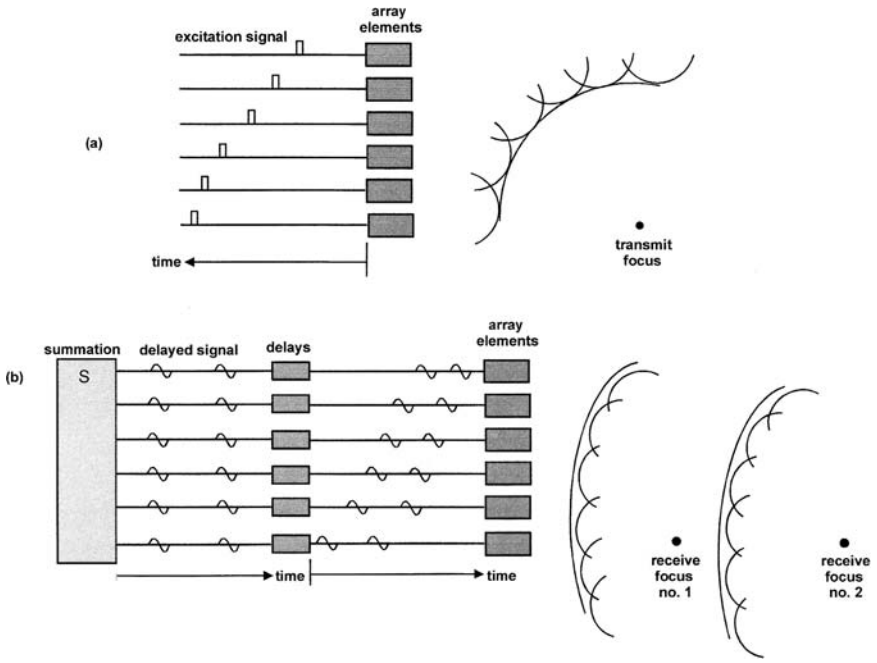


FIGURE 16.7. The phased array illustrated above provides for steering and focusing of an ultrasonic beam. In (a) the six-element linear array is shown in the transmit mode. In the receive mode of (b), dynamic focusing allows the scanner focus to track the returning echoes.

In the receive mode, a dynamic adjustment can be made with a focal point so that it coincides with the range of returning echoes. After the transmission of a pulse, the initial echoes return from targets nearest the transducer. The scanner therefore focuses the phased array on these targets, located at focus point 1 as shown in Figure 16.7(b). As echoes return from the more distant targets, the scanner focuses at a greater depth (e.g., focus point 2). Focal zones are achieved with adequate depth of field so that the targets always remain in focus in receive mode. This is the *dynamic receive focusing* process.

Array-Element Configurations

The dynamic receive focusing process is repeated many times to form an ultrasonic image in the scan of a 2D or 3D region of tissue. In defining the 2D image, the *scanning plane* is the azimuth dimension; the *elevation dimension* is normal to the azimuth scanning plane. In *linear sequential arrays*, as many as 512 elements constitute a sequential linear array in currently available scanners. A subaperture containing as many as 128 elements is selected to function at a given time.

In Figure 16.8(a), the scanning lines are directed perpendicularly to the transducer face; the acoustic beams is focused but not steered. The advantage is that

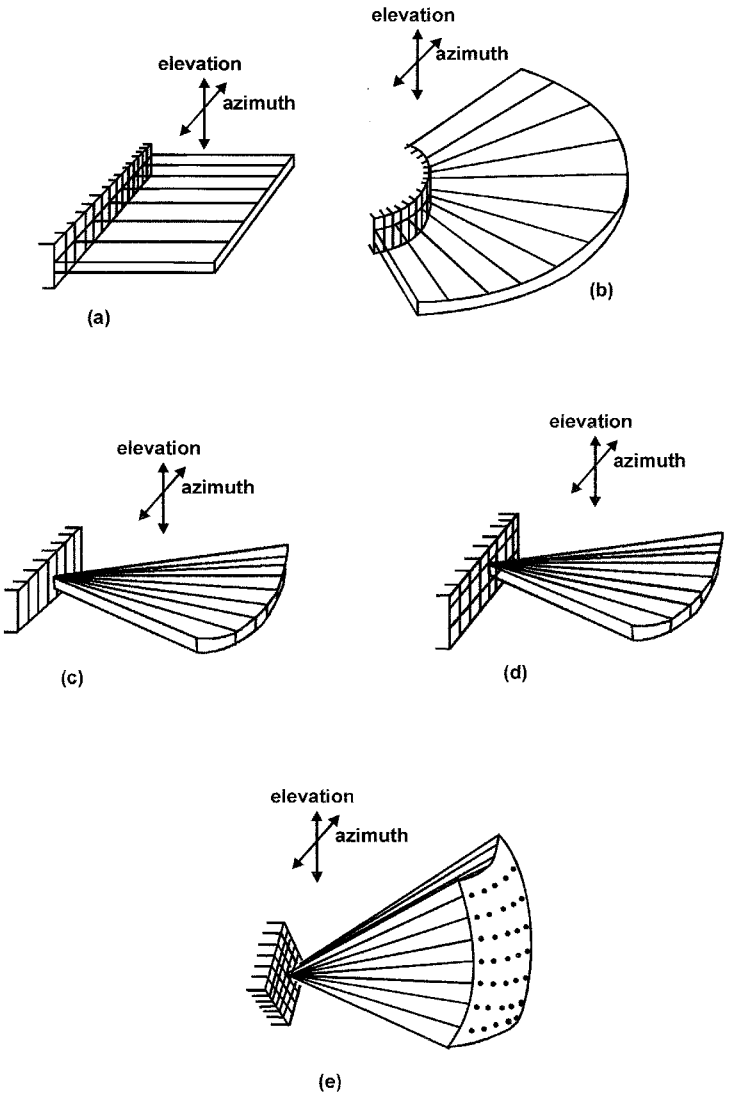


FIGURE 16.8. Various configurations of array elements and the corresponding regions scanned by the acoustic beam: (a) sequential linear array scanning a rectangular region; (b) curvilinear array scanning a sectored region; (c) linear-phased array sweeping a sectored region; (d) 1.5D array scanning a sectored region; and (e) 2D array sweeping a pyramidal region.

elements have high sensitivity when beam is directed straight out. The disadvantage is that the field of view is limited to the rectangular region directly facing the transducer. Linear-array transducers also require a large footprint to obtain a sufficiently wide field of view. Another type of array configuration is that of the

curvilinear array. Because of its convex shape [Figure 16.8(b)], the curvilinear (or convex) array scans a wider field of view than does a linear-array configuration. The curvilinear array operates in the same manner as the linear array in that the scan lines are directed normal to the transducer face. In the *linear-phased array* of Figure 16.8(c), which may contain as many as 128 elements, each element is used to emit and receive each line of data. In Figure 16.8(c), the scanner steers the beam through a sector-shaped region in the azimuth plane. These phased arrays can scan a region considerably wider than the footprint of the transducer, thus rendering them suitable for scanning through acoustically restricted windows. This is ideal for use in cardiac imaging, where the transducer must scan through a small window in order to avoid obstruction by ribs and lungs.

The *1.5D array* is structurally similar to a 2D array but operates as 1D. The 1.5D array consists of elements along both the azimuth and elevation directions. Dynamic focusing and phase correction can be implemented in both dimensions to enhance image quality. Steering is not possible in the elevation dimension since a 1.5D array contains a fairly limited number of elements in elevation (usually 3 to 9 elements). Figure 16.8(d) shows a B-scan conducted with a 1.5D phased array. It is also possible to use linear sequential scanning with 1.5D arrays. In the *2D phased array*, a large number of elements are employed in both the azimuth and elevation. This permits focusing and steering of the acoustic beam along both dimensions. With the application of parallel receive processing, a 2D array can scan a pyramidal volume in real time to yield a volumetric image, as illustrated in Figure 16.8(e).

Linear-Array Transducer Specifications

In the design or selection of an ultrasound transducer, a number of compromises are entailed. The ideal transducers have high sensitivity and SNR (signal-to-noise ratio), excellent spatial resolution and freedom from spurious signals. Additionally an individual array element should possess wide angular response in steering dimensions, low cross-coupling with another element, and an electrical impedance matching the transmitter.

16.7 Basic Instrumentation: Scanning Methods

Figure 16.9 illustrates a "generic ultrasonic instrument" in block diagram format. The synchronization generator establishes a repetition rate (typically 1,000 repetitions per second) for a pulse input and a display module. The pulser provides electrical energy to the transducer through a pulse limiter, which clips the voltage experienced by the amplifiers down to a tolerable threshold but allows the full voltage to impact the transducer. The transducer generates the ultrasonic waves and receives the echoes. The resulting voltage of the echoes, not clipped by the pulse limiter because their voltages are usually below the clipping threshold, goes on directly to the amplifier and then onward to a display. The computer (which may or may not be present) acts to process algorithms.

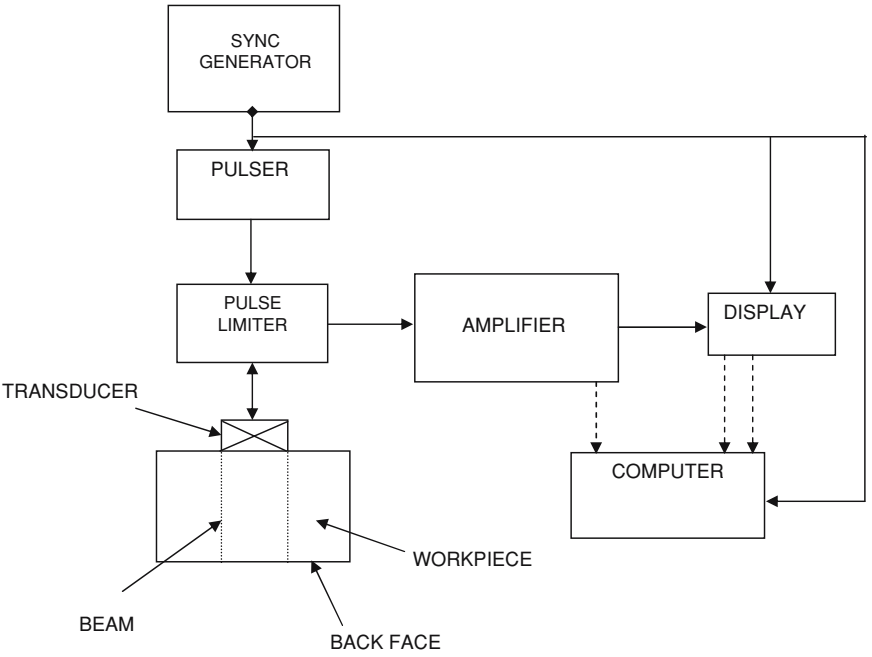


FIGURE 16.9. Block diagram of a "generic ultrasonic instrument."

A-Scan

In a flaw-detection instrument the display may be an oscilloscope with limited adaptability. The display, shown in Figure 16.10(a) resulted from a well-damped transducer. If the transducers incorporate less damping or if they are electronically tuned, the echoes will contain more RF (radio frequency) cycles. In general terms, the result is rectified and detected prior to the display stage, as shown in Figure 16.10(b). This results in the A-scan, which is simply amplitude versus time for the echoes falling in the range of the transducer.

Analog processes are incorporated in commercial flaw-detection instruments to analyze data. In addition to the rectification of detection of the echoes described above, another process is that of a *baseline suppressor* that eliminates a few decibels of the amplitude just above zero amplitude. This process serves to emphasize the more significant flaw echoes, which are now easier to ascertain. A third process constitutes the range gate that selects a time range from t_1 to t_2 along the A-scan baseline. The signals within the range gate may be either (a) suppressed or (b) analyzed to the exclusion of all other signals. A computer, an internal alarm trigger, or an external analog instrument may execute the analysis. An alarm may be activated if echoes fall within the range gate exceeds a certain amplitude or threshold.

The A-scan procedure is also used to measure gauge thickness on the basis of the known velocity of the pulse inside the workpiece and the time it takes for an echo to be received from the backside of the workpiece.

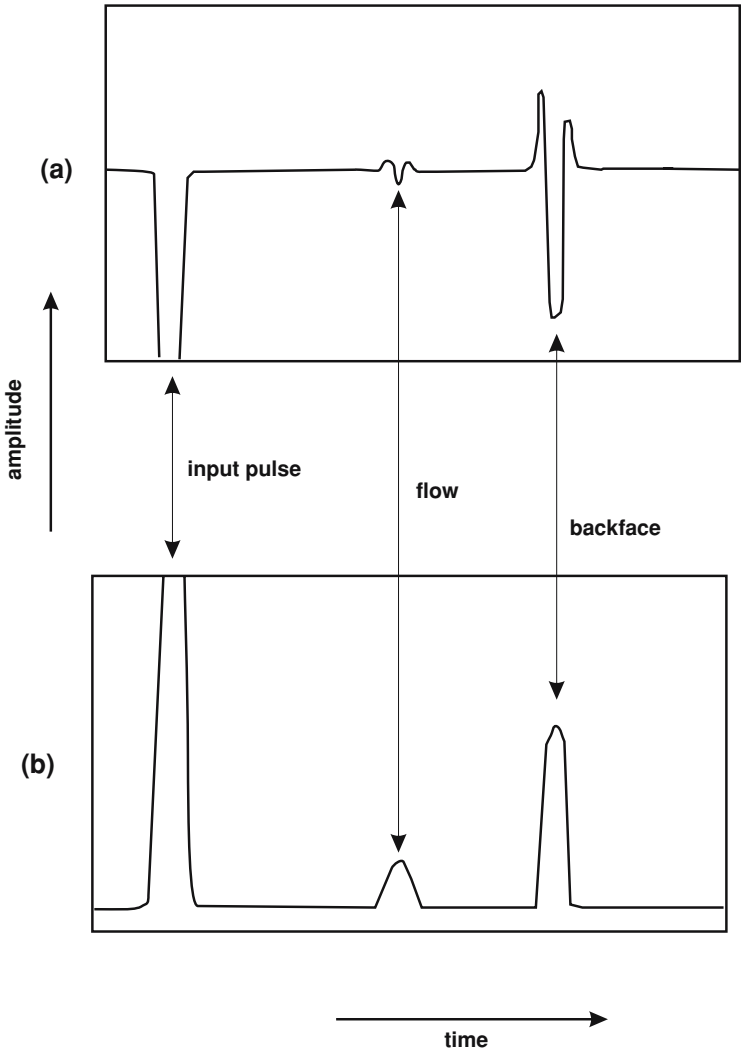


FIGURE 16.10. A-scan display for the generic ultrasonic instrument of Figure 16.9. The amplitude is shown presented as RF in (a) pr as rectified and detected in (b), versus time, which is proportional to the distance into the workpiece.

B-Scan

By introducing a mechanism to physically move a transducer over a workpiece in an appropriate manner, a 2D electronic or electromechanical display can provide a picture of a slice of the interior of the workpiece. Figure 16.11 shows a schematic of a scanning mechanism in an immersion tank. The transducer moves in the

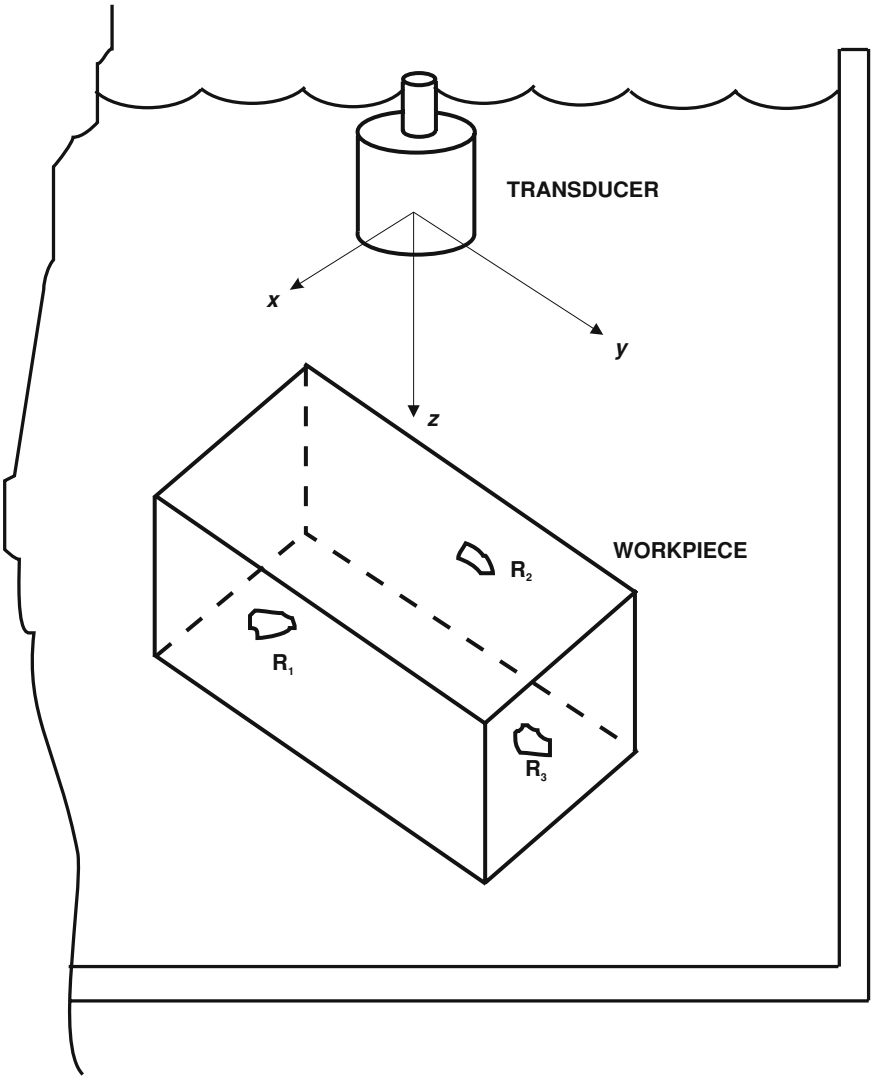


FIGURE 16.11. Schematic of a scanning mechanism in an immersion tank. The transducer can be translated and angulated to aim at a target.

x- and y-directions while propagating signals in the z-direction (travel time is along the z-direction).

A B-scan portrays an x–z slice of the workpiece at a specific position y_i and then makes another x–z survey at y_{z+1}, and so on. The transducer travels along x at a rate allowing the ultrasonic pulse to interrogate the workpiece many times over the x-direction of the workpiece. The pulse repetition rate must be kept sufficiently

slow to allow the previous pulse to drop below the noise level before the next pulse is sent. One of the coordinates on the display is the position x_B representing the location the transducer was at the time of arrival t_B of a refection from a reflector R (R_1 or R_2 or R_3) as shown in Figure 16.11, depending on whether the beam in the slice y passes over them. It follows that t_B represents the other coordinate of the point (x_B, t_B). If the display is assigned a gray scale, the brightness of the display corresponds to the amplitude of the reflection. In order to establish an "on-off" scale, a threshold may be assigned to that only echoes exceeding a certain amplitude are displayed. The B-scan procedure is extensively used in medical ultrasonics, especially in obstetrics.

C-Scan

The C-scan procedure differs from B-scan in that, while the transducer scans along the x-axis in consecutive steps y_i, the coordinate on the display is (x, y). The display is nonzero only if a reflecting surface R lies within the beam range along z centered at (x, y). This configuration would ordinarily result in the echoes from the front and back surfaces of the workpiece swamping the display, but they are disallowed by setting up a range gate to accept and transmit echoes only if they arise from regions below the top surface and above the bottom surface of the workpiece. It is also apparent that the range gate can be narrowed to accept signals from a slice thinner in the z-direction. A threshold may also be prescribed within the gate so that only signals above a set amplitude will be displayed. A gray scale can also be assigned.

In both the B-scan and A-scan procedures, a computer can be utilized for data storage and image enhancement. Color monitors can be useful for pseudocolor gray scales.[2]

References

Atchley, A. A. and Crum, L.A. 1988. Acoustic cavitation and bubble dynamics. In: *Ultrasound: Its Chemical, Physical and Biological Effects*. Suslink, K. S. (ed.). New York: VCH Publishers, pp. 1–64.

Bardeen, J. B., Cooper, L. N., and Schrieffer, J. R. 1957. Theory of superconductivity. *Physics Review* 108:1175.

Blake, J. R., Boulton-Stone, J. M., and Thomas, N. H. (eds.). 1994. *Bubble Dynamics and Interface Phenomena*. Dordrecht, the Netherlands: Kluwer.

Blitz, Jack. 1967. *Fundamentals of Ultrasonics*, 2nd ed. New York: Plenum Press. (A bit dated, but this remains a very clear exposition of the ultrasonic field.)

Brown, B. and Goodman, J. E. 1965. *High Intensity Ultrasonics*. London: Iliffe.

Crum, L. A. 1994. Sonoluminescence. *Physics Today*. September: pp. 22–29.

[2] "Color" displays for ultrasound analyses do not represent optical colors of the ultrasound targets, per se. Rather, they indicate values assigned to specific parameters of ultrasound reflections.

Ensminger, Dale. 1988. *Ultrasonics: Fundamental, Technology, Applications*, 2nd ed. New York: Marcel Dekker, Inc. (A comprehensive, well-done modern text giving an overview of the field of ultrasonics.)
Frenzel, H. and Schültes, Z. 1934. *Physical Chemistry* 27B:421.
Gaitan, D. F. and Crum, L. A. 1990. In: *Frontiers of Nonlinear Acoustics*. Hamilton, M. and Blackstock, D. R. (eds.). New York: Elsevier Applied Science, pp. 49–463.
Gaitan, D. F., Crum, L. A., Roy, A., and Church, C. C. 1992. *Journal of the Acoustical Society of America* 91:3166.
Herzfeld, Karl F. and Litovitz, Theodore A. 1959. *Absorption and Dispersion of Ultrasonic Waves*. New York: Academic Press. (A classic reference on ultrasonic phenomena.)
Hunter, T. F. and Bolt, R. G. 1955. *Sonics*. New York: John Wiley & Sons, p. 234.
Landau, Lev and Teller, Edward. 1936. Zur Theorie der Schaledispersion. *Physik Zeitschrift Sowjet Union* 10:34.
Leighton, T. G. 1994. *The Acoustic Bubble*. London: Academic Press.
Maeda, K. and Ide, M. 1986. *IEEE Transactions: Ultrasonics, Ferroelectrics, and Frequency Control*, UFFX-33(2):179–185.
Mason, Warren P. (ed.). 1964. *Physical Acoustics—Principles and Methods*, Vol. 1, Part A. New York: Academic Press, Chapters 3–5.
Morse, R. W. 1959. In: *Progress on Cryogenics*. Mendelssohn, K., (ed.), Vol. I. London: Heywood.
Neppiras, E. A. 1973. The prestressed piezoelectric sandwich transducer. *Proceedings of the International Ultrasonics Conference*: pp. 295–300.
Shields, F. D. 1962. thermal relaxation in fluorine. *Journal of the Acoustical Society of America* 34(3): 271.
Sternberg, M. S. 1958. *Physics Review* 110:772.
Suslick, Kenneth S. and Crum, Lawrence A. 1997. Sonochemistry and sonoluminescence. In: *Encyclopedia of Acoustics*. Crocker, Malcolm J. (ed.), Vol. 1. New York: John Wiley & Sons, Chapter 26, pp. 71–281.
Winter, T. G. and Hill, G. L. 1967. High temperature ultrasonic measurements of rotational relaxation in hydrogen, deuterium, nitrogen, and oxygen. *Journal of the Acoustical Society of America* 42(4): 848.
Zuckerwar, A. J. and Griffin, W. A. 1980. Resonant tube for measurement of sound absorption of gases at low frequency/pressure ratios. *Journal of the Acoustical Society of America* 68(1):218.

Problems for Chapter 16

1. For a frequency of 0.1 MHz, find the threshold frequency in gas-free water for the onset of cavitation. Compare with the value of the threshold for aerated water subjected to the same frequency.
2. At a hydrostatic pressure of 135 μPa and a 35 kHz sinusoidal signal, find the critical radius for cavitation to occur. Assume a value of 1.4 for the ratio of specific heats and a surface tension of 75 dynes/cm.
3. At a magnetic field of 500 oersteds, find the mechanical strain of nickel.
4. At the magnetic field of 500 oersteds, find the mechanical strain of permendur. What is the major difference between the mechanical strain of this problem and that of Problem 3?

5. In the cases of Problems 3 and 4 for a crystal of 20 mm length, what will be the respective changes in the length of the crystals fabricated from the materials of the last two problems?
6. Compare the mechanical strains of ferrous ferrite with that of 12 Alter for a magnetic polarization of 20 weber m^2.
7. Given a magnetostrictive material with a known Young's modulus of 110 GPa and a density of 7650 kg/m^3:
   (a) what is the velocity of sound in that material?
   (b) find the fundamental frequency for magentostriction for a length of 50 mm.
8. Why is A-scanning inadequate for a sonogram of a fetus?

17
Commercial and Medical Ultrasound Applications

17.1 The Growth of Ultrasonic Applications

It is recognized among technologists that the field of ultrasonics is still in its infancy, and many new uses for ultrasound will continue to accrue in commercial and medical fields. Ultrasound is proving its continued worth in diagnostics, as it is generally nondestructive at lower intensities and does not engender the deleterious effects on materials and living tissues that X-rays do. At higher levels of operation, ultrasound is used in manufacturing operations in cleaning, cutting, and welding materials and to provide therapeutic treatment and alleviate the trauma of surgery by making possible increased precision in incisions, targeting of tumors and infected tissues for selective ablation, and stanching bleeding. Improvements have been made and are still continuing; as the results of better imaging, more versatile transducer arrays, and more powerful analytical algorithms are rendered possible by more powerful computers and larger data storage systems. In this chapter a survey is conducted of the uses of ultrasound in manufacturing and agricultural processes, quality control, and in medical (including dentistry) diagnostics and therapeutic procedures. In addition to manufacturing, the petroleum, chemical, and pharmaceutical sectors also make use of ultrasound to monitor and facilitate production. The safety of ultrasound also constitutes an important topic in this section.

17.2 Industrial Applications of Ultrasound

Ultrasonic Cleaning

Ultrasonic cleaning is the oldest industrial application of power ultrasound. Applications span a wide variety of industries ranging from castings to semiconductors. Ultrasonic cleaning works best on relatively hard materials such as metals, glass, ceramics, and plastics, which reflect rather than absorb sound.

Both cavitation and the agitation of the fluid by the waves are entailed in the process of ultrasonic cleaning. At lower frequencies, cavitation acts as the

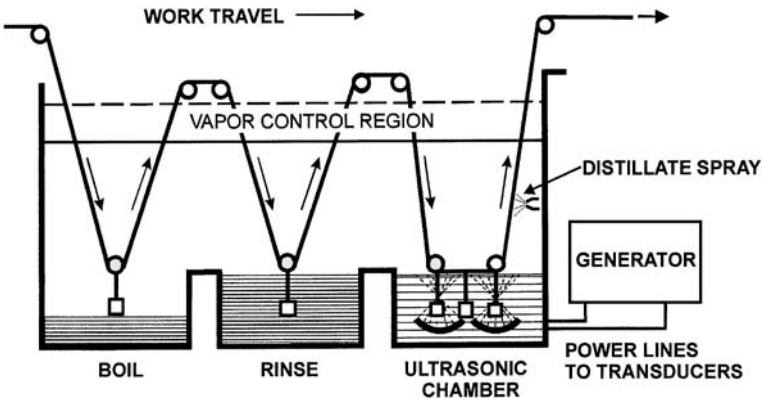


FIGURE 17.1. Conveyor belt for ultrasonic cleaning.

principal agent but at higher frequencies the cleaning effect occurs mainly from agitation. Most cleaning applications are executed in the frequency range of 20–50 kHz, where cavitation effects occur more strongly. Either piezoelectric or magnetostrictive sources are used. The workpiece being cleaned is immersed in a tank containing a liquid selected on the basis of its susceptibility to cavitation, its detergent properties, ability to degrease, and so on. Trichlroethylene and cyclohexane are among the more satisfactory fluids used for ultrasonic cleaning. Standard ultrasonic cleaners range in power from 100 to several thousand watts with corresponding tanks capacities of 4–160 liters. Multi-kilowatt special systems with tank capacities of several hundred liters are not uncommon. In recent years, low-power, inexpensive (below $100) ultrasonic cleaners have been made available, thereby rendering ultrasonic cleaning accessible to small shops and laboratories.

Ultrasound cleaning lends itself to continuous processing in which a series of workpieces can be transported on a conveyor belt through a series of processes in separate tanks as shown in Figure 17.1. Ultrasonic cleaning has superseded other older usual methods of cleaning, particularly when these methods are ineffective and liable to cause damage. Applications include the removal of lapping paste from lenses without scratching after grinding, the flushout of grease and machining particulates from otherwise inaccessible small crevices in engine components, removal of blood and other organic material from surgical instruments after use, and so on. Very delicate parts that can be damaged by cavitation are cleaned by wave agitation at much higher frequencies, from 100 kHz to 1 MHz.

In general, modern ultrasonic cleaners employ solid-state electronic power supplies incorporating automatic tuning; thus they do not require operator's attention. A problem prevalent with all ultrasonic cleaners is the gradual deterioration of the tank due to cavitation erosion. This depends principally on the application, and well-designed systems can provide years of satisfactory service.

Flaw Detection and Thickness Measurements

A method of nondestructive testing, the pulse technique, is used extensively to determine the propagation constants of solids, particularly in the MHz frequency range. This method consists of sending a short train of sound waves through a medium to a receiver. In the transmission mode of the pulse technique, the receiver is placed at a measured distance from the source. In the echo mode, a reversible transducer acts as both source and receiver, with a reflector used to reflect the pulses. The speed of sound in a medium can be determined from the time of travel of the pulse over a given length of acoustic path. Longitudinal waves are generally used. In gauging the thickness of a specimen, advantage is taken of the fact that a beam of pulses will reflect from the specimen surface opposite to the side of the reversible transducer.

The use of the single-pulse method for flaw detection is fairly straightforward when the specimen has two parallel surfaces and the defect is linear and roughly parallel to these surfaces but not too close to a surface or another defect. If the pulse is followed on an oscilloscope and there is no defect present, two peaks, say, A and B, will appear on the screen. Peak A represents the instant of the transmission of the pulse, and peak B that of its return after a simple echo. Peak B is referred to as the *bottom echo*. When a defect is present, a discontinuity of the characteristic impedance and some, or possibly all, of the sound energy is prematurely reflected back to the transducer. Another peak will then occur between A and B. The distance AC indicates the depth where the flaw exists and the height of the peak C determines the extent of the defect.

Figure 17.2 illustrates a schematic of a longitudinal wave probe used to detect flaws. A crystal transducer is normally used, and it is encased in a suitable housing. The crystal is mounted for heavy damping, which results in the propagation of

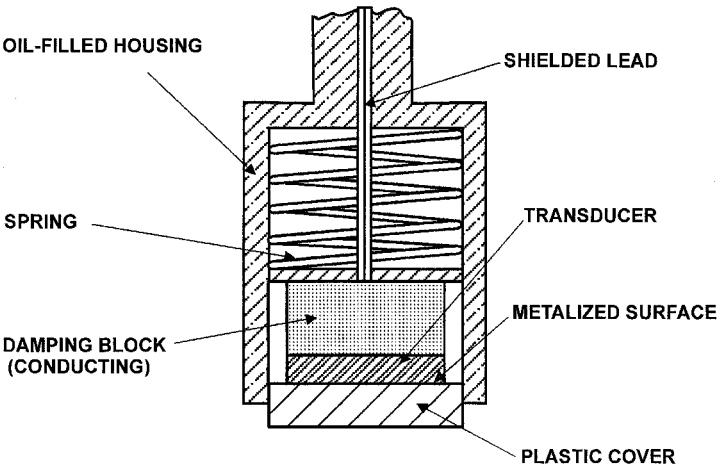


FIGURE 17.2. Longitudinal wave probe for detecting flaws.

small pulses to allow for greater accuracy in locating defects and better resolution of neighboring defects from one another. The transducer is protected by a plastic cover that is coupled to the crystal with oil so as to prevent wear by friction between the surfaces of the crystal and the material being tested. It would be ideal for the characteristic impedances of the transducer material, the material of the protective cover, and the oil should be similar. Sometimes it is more desirable to apply the immersion method, whereby both the specimen and the probe are immersed in water, with the probe at a fixed distance from the upper surface of the sample. An additional peak that represents the echo from the interface between the immersion liquid and the sample will appear on the oscilloscope screen.

When a defect is not parallel with a surface, it is preferable to use an angled probe, which consists of a transducer mounted on a wedge. This enables the sound waves to be incident normally to the defect, and a greater degree of sensitivity is thus achieved. Using *variable-angled probes* makes it easier to better gauge the direction of defects that might otherwise remained undetected. For samples having irregular shapes, two probes are used, one acting as a sender and the other serving as a receiver in order to locate the defects. Care should be taken that coupling between the transducers should involve the medium and nothing else.

Flaws, such as those that occur in butt welds oriented at right angles to the surface, are commonly detected through the procedure of *forward scanning*. Here a beam of transverse waves is propagated, after refraction at the boundary, in the medium at a shallow angle to the surface. In Figure 17.3(a) the transducer is shown

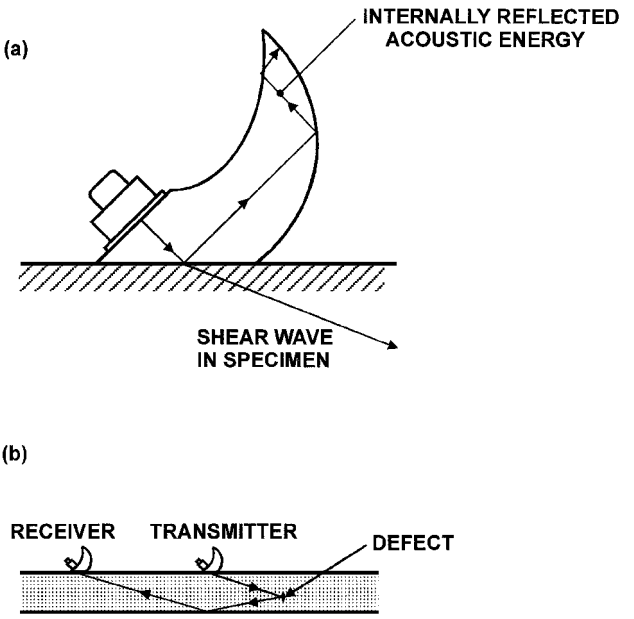


FIGURE 17.3. Transducer mounted on plexiglass wedge and setup of transverse wave probes for locating a defect in a specimen.

mounted on a Plexiglas wedge and the longitudinal waves are directed to the surface with an angle of incidence greater than the first critical angle (this is the angle of incidence sufficiently large that the refracted ray is directed along the boundary rather than into the medium). The figure shows that the wedge is shaped in such a manner that the longitudinal waves that are reflected at the surface become totally absorbed by subsequent reflections. In Figure 17.3(b) a similar probe is positioned at a suitable location to receive the waves from the defect after a reflection at the base of the specimen. In a given substance, the transverse wave velocity is generally about half that of the longitudinal wave velocity, so the sensitivity of this methodology is twice that for longitudinal procedures.

Surface defects can be discerned through the means of *surface waves*. These are produced by a probe similar to that shown in Figure 17.3(a), but the incident longitudinal waves are directed to the surface at the second critical angle where the transverse waves are refracted at an angle of 90° (i.e., along the boundary surface). Laminar defects that exist just below the surface, which are hard to detect by normal longitudinal wave methods, can be located by Lamb waves (Worlton, 1957). According to Lamb, a solid plate can resonate at an infinite number of frequencies. The portion of specimen between the surface and a lamination close to it forms such a plate. If surface waves are directed toward this plate, it will resonate and generate a signal that can show up on an oscilloscope screen.

Determination of Propagation Velocity and Attenuation through an Interferometer

The interferometer is a continuous wave device that can accurately measure velocity and attenuation in liquids and gases that can sustained standing waves. It consists of a fluid column that contains a fixed, air-backed piezoelectric transducer at one end and a moveable rigid reflector at the other end. A fixed frequency is selected. The reflector is moved with respect to the transducer by a micrometer adjustment mechanism. As the reflector moves, the reflected waves become periodically in and out of phase with the transmitted waves, as a result of the corresponding constructive and destructive interference. The effect of the interference on the crystals influences the load impedance detected by electronic system. The load current in the electronic amplifier fluctuates accordingly. The wavelength of the sound is established by the distance the micrometer moves the reflector over one cycle of load current fluctuation, with the distance between two successive maxima being equal to half-wavelength $\lambda/2$.

Optical interference methods also have been used in this fashion to accurately measure the wavelengths of standing waves at high frequencies (near 1.0 MHz or above). An accuracy of 0.05% is typical for the interferometer, which depends on the quality of micrometer readings, the parallelism between transducer and reflector surfaces, and the accuracy of the frequency determination. The velocity of sound is found by simply multiplying the frequency by the wavelength. The

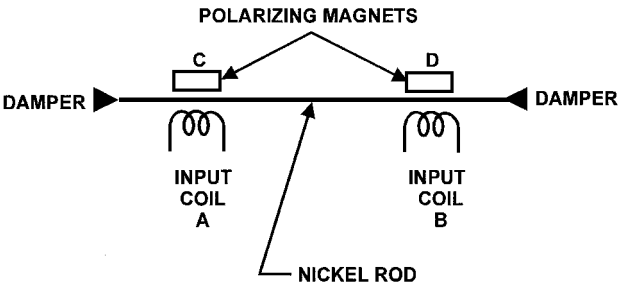


FIGURE 17.4. Magnetostrictive delay line.

attenuation is found from the decay of the maxima of the periodic amplitude plotted as the function of the distance x between the transmitter and the reflector increases.

Ultrasonic Delay Lines

Delay lines are used to store electrical signals for finite time periods. These are used in computers to store information to be extracted for a later stage of calculation. A method for generating the delay is to convert those signals into ultrasonic waves that then travel through a material to be reconverted into their original forms. The simplest ultrasonic delay line is a crystal transducer radiating into a column of liquid, such as mercury, that terminates at a reflector. An adjustment of the delay time can be effected by changing the position of the reflector relative to the crystal. As liquid delay devices are not always convenient to use, solid delay lines are more common. For delay times of a few microseconds, only a few centimeters of a solid rod or block is sufficient. The delay time may be doubled by using shear waves instead of longitudinal waves. If even longer delay times are required, say, in the order of a few milliseconds, very long acoustic paths are necessary. These can be achieved by using materials in the form of polygons in which large numbers of multiple reflections can occur. The solid delay line has the disadvantage that the delay time usually cannot be varied. However, the magnetostrictive delay line, illustrated in Figure 17.4, can be varied in length and unwanted reflections are avoided by coating the ends of the rods with grease, which completely absorbs the sound waves. The line may be a wire, a rod, or some sort of ribbon of ferromagnetic material such as nickel. An electrical signal applied to coil A induces through a magnetostricive effect a sound pulse in the line. The pulse travels along the line and induces an electrical signal in coil B by the reverse magnetostrictive effect. The permanent magnets C and D provide the requisite polarization.

Measuring Thicknesses through the Pulse Technique

When the pulse technique is used for gauging thicknesses, better results are achieved through the use of a variable delay line. Two pulses are generated simultaneously. One pulse is sent through the sample and the other through a delay

line, which can be a length of nickel wire or a column of liquid terminating in a reflector. The latter pulse is indicated on the oscilloscope by a trace following that representing the pulse passing through the sample. The delay line is adjusted in length by means of a micrometer device until the two traces on the screen coincide positionwise. The thickness of the specimen is derived from the predetermined calibration of the delay line.

A major advantage of using ultrasound for thickness measurement is that access to only one surface is needed. This is especially useful in measuring the extent of corrosion in infrastructures such as viaducts, sewers, gas pipes, and chemical conduits. The thicknesses of ship hulls can be monitored at sea without resorting to taking sample borings, an expensive and tedious process that is not 100% effective. In the livestock industry, ultrasonic thickness measurement is used to measure the amount of fat on the bodies of live animals.

Mechanical Stress Measurements

When a solid material undergoes a change in mechanical stress, changes also occur in its elastic moduli and hence in its acoustic velocities (Shahbender, 1971). This method can also determine the variations of stress in real time. In order to render possible the use of this procedure, a calibration curve is necessary to provide the reference data, namely, plots of velocity (in terms of percentage) in a direction as a function of stress applied normal to that direction. Figure 17.5 displays typical calibration curves for longitudinal waves and also for shear waves polarized perpendicular to the direction of the stress and also polarized parallel to the direction of the stress. It can be inferred that the variations of the two components of shear velocity with time are different. Thus, at any given time the corresponding wave vectors will be out of phase with each other. This phase

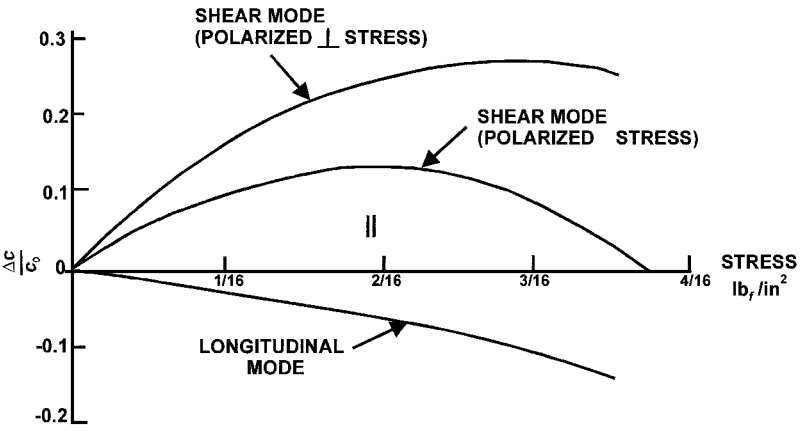


FIGURE 17.5. Calibration curves for longitudinal waves.

difference ϕ is given by

$$\phi = \frac{L\omega}{c_0}\left[\left(\frac{\Delta c}{c_0}\right)_n - \left(\frac{\Delta c}{c_0}\right)_p\right]$$

where ω denotes the angular frequency, L is the acoustic path length, c_0 is the shear wave velocity in the unstressed medium, and $(\Delta c/c_0)_n$ and $(\Delta c/c_0)_p$ represent the fractional shear velocity for polarizations in directions normal to and parallel with the direction of stress, respectively. The value of L is found simply from longitudinal wave measurements while ϕ is obtained by the means of a suitable phase-shift network. This methodology is also applied to determine third-order elastic constants.

The Ultrasonic Flowmeter

The Doppler principle constitutes the operating basis of the ultrasonic flowmeter. Two reversible transducers are submerged in the liquid along the line of flow. One transducer acts as a signal source of ultrasonic pulses and the other acts as a receiver. At short regular intervals the roles of the transducers are reversed, so that the source becomes the receiver and the receiver becomes the source. The wave velocities are $c + u$ along the direction of the flow and $c - u$ in the opposite direction, where c represents the propagation velocity of sound in the fluid and u the velocity of the streamline flow of the liquid. A number of techniques have been used to compare the upstream and downstream propagation rates. A *"sing-around" method* uses a pulse generator to produce a short train of ultrasonic waves. The received signal is amplified and used to retrigger the pulse generator. If we neglect delay times due to the electronic system and the distance the pulse travels beyond the fluid stream, the difference between the downstream and upstream pulse repetition rates is

$$f_1 - f_2 = \frac{2u}{d}$$

where d represents the distance between the two transducers.

Another type of ultrasonic flowmeter is based upon the deflection of an ultrasonic beam by the fluid flow (Dalke and Welkowitz, 1960). In Figure 17.6, a transmitting transducer located on one side of the fluid conduit emits a continuous signal into the fluid stream. A split transducer on the other side of the pipe determines the amount of beam deflection. A differential amplifier is used to determine the difference in the outputs of the two receiving transducers. If there is no flow, the beam falls midway between the two receiving sections, and the two sections generate equal voltages, and the output from the differential amplifier will be zero. When fluid flow occurs, the beam shifts in the direction to the flow by an amount corresponding to the flow speed, and the outputs from the two sections differ. The difference voltage then corresponds to the rate of flow. Assuming a constant flow rate across the pipe, the deflection ϕ of the beam is computed from

$$\phi = \tan^{-1}\left(\frac{u}{c}\right)$$

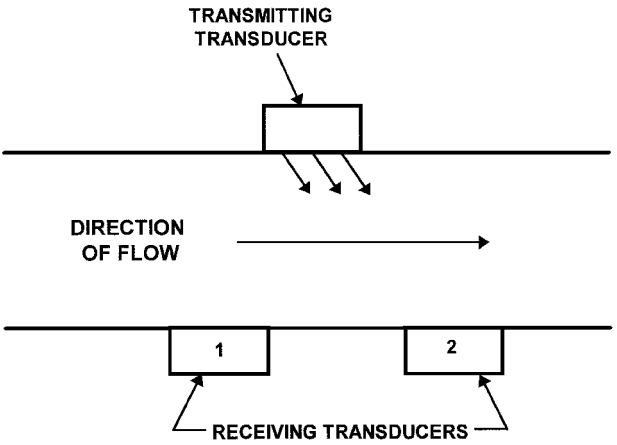


FIGURE 17.6. Schematic of a beam-deflection ultrasonic flowmeter.

Ultrasonic meters are being used to measure flow rates of rivers, nuclear reactor heat-exchanger fluids, gas-containing emulsion, slurries, corrosive liquids, and wind velocities. These measurement devices carry the advantage that they introduce negligible pressure loss in a system and they are economical to operate and can handle a wide range of flow rates, pipe diameters, and pressures.

Resonance Method of Measuring Sound Propagation Speed

The resonance method is similar to the interferometer method for measuring velocity, but it can be applied to solids as well as to fluids. This involves the use of a fixed transducer and a fixed reflector or two transducers spaced a fixed distance apart. The transducer is driven to sweep through a range of frequencies to determine successive resonances. In a nondispersive medium, the difference between two successive resonant frequencies equals the fundamental resonant frequency of the medium, i.e.,

$$c = 2\,l\Delta f$$

where l is the distance between the transducer and the reflecting surface and Δf denotes the difference between successive resonant frequencies.

Motion and Fire Sensing

One of the few ultrasonic applications in open air is that of the motion and fire sensor, which is restricted to the lower kilohertz range, where attenuation is not very much. A magnetostrictive transducer placed at some point in a room emits pulses in all directions. The reflected signals from the walls and furniture are eventually picked up by a receiver, from which a constant indication is generated. Any variation in the sound field, caused by an intruder or a change in temperature,

gives rise to a change in this indication, which triggers an alarm. One version of this device is a light switch that goes on automatically when someone enters an empty room and goes off automatically when nobody is present in the room after a predetermined period of time.

Working of Metals and Plastics

Ultrasound have been successfully used in the treatment and working of metals and plastics. As a molten metal is cooled, bubbles should be removed before solidification sets in. Otherwise defects will occur. Irradiation of the melt with ultrasound sets the fluid particles in motion so that bubbles tend to coalesce and, when sufficiently large, tend to rise to the surface. As cooling progresses, crystals begin to form at the solidification temperature. The crystal growth depends on the rate of cooling and the presence of specific impurities in the metal. The high energy of the ultrasound-induced cavitation causes the crystals to be break up as they form. The solid that is finally formed has a much finer grain structure than it would have if cooling had taken place with the melt undisturbed.

Ultrasonic machining is one of the first industrial applications of high-intensity ultrasound. In one technique, a slurry consisting of abrasive particles suspended in a low-viscosity liquid flows over the end of a tool shaped in the desired geometry of the impression to be made. The axial motion of the tool tip and the ensuing cavitation impart high accelerations to the abrasive particles and thus erode away the material from the workpiece. This method is suitable for machining of brittle ceramic materials as well as of powder-metal components. But in the latter application, there has been evidence of highly accelerated electrolytic erosion that adversely affects the tool.

Ultrasonic drills do not have transducers in immediate contact with the workpiece. An intermediate material is necessary, and this has the task of matching the impedance. In order to achieve maximum velocity amplitude in the tool, a tapered rod that acts as an acoustic transformer in the same manner as a loudspeaker horn is used. The rod is made of a material that has a characteristic impedance matching that of the transducer to which it is rigidly fixed. It is exactly one wavelength long in order to achieve maximum transfer efficiency and clamped at a distance of one-fourth wavelength from the transducer.

The soldering of metals without any flux can be achieved through an ultrasonic soldering iron illustrated in Figure 17.7. The soldering iron is similar in design to an electronic drill, but the tool at the end of the drill is replaced by the electrically heated bit. The vibration of the bit produces cavitation in the solder, thus effectively cleaning the surface of the workpiece and removing any oxide coating. Until fluxes were developed for soldering aluminum, ultrasound had provided the only effective method of soldering this metal. Ultrasound soldering is now being extensively used for soldering joints in miniaturized printed circuit boards.

Ultrasonic welding is effected when two materials are pressed together in intimate contact and the ultrasound vibrations produce shearing stresses at the interface, thereby generating a great deal of heat. This phenomenon lends itself well to

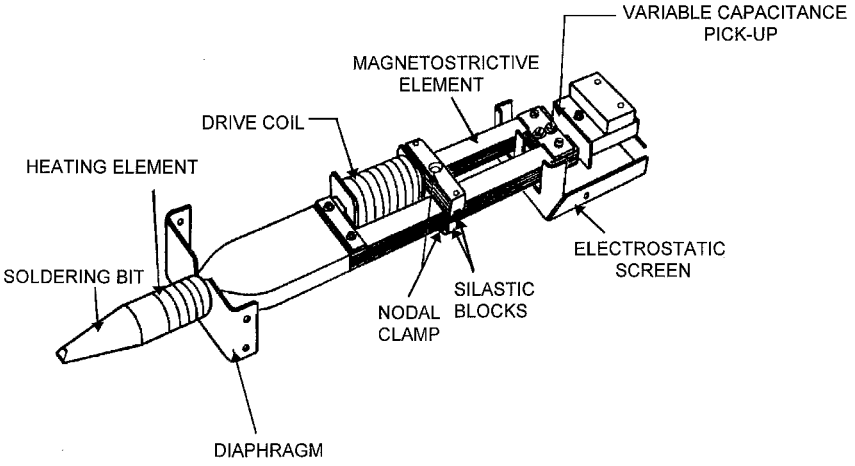


FIGURE 17.7. Ultrasound soldering iron.

produce spot and seam welds in metals such as 304 and 321 stainless steel, aluminum, brass, copper, zirconium, titanium, gold, molybdenum-0.5% titanium, and platinum. This type of welding can be applied to very small wires to seam welding of metal plates up to 0.5 cm in thickness. Both similar and dissimilar metals can be welded. While the formation of a thin molten film in the interface seems to be the primary mechanism of bonding, there is some evidence of solid-state bonding instigated by diffusion in the welding zone at low amplitudes and high clamping pressure.

Bonding of thermoplastic materials[1] through ultrasound is even easier than for metals, since the necessary equipment is smaller and requires appreciably less power. Ultrasonic welding of thermoplastic materials has gained wide acceptance, particularly in situations where thick materials are joined together and the toxicity of adhesives must be avoided. In addition to bonding plastic parts, the same equipment can be used to insert metal parts in plastic pieces. A hole slightly smaller than the metal is drilled or incorporated in a molding part and the metal is driven into the hole ultrasonically. During the insertion the melted plastic encapsulates the metal piece and fills flutes, threads, undercuts, and so on.

Ultrasound welding combines the ideal ingredients sought in modern manufacturing. The process is fast and clean, requires no consumables, does not call for skilled operators, and lends itself to automation. It is used extensively in the automotive industry of assembly of taillights dashboards, heater ducts, and other components where plastics have replaced the traditional use of metal and glass.

[1] Plastics fall into two categories: thermoplastic and thermosetting. The former plastic type will soften at sufficiently elevated temperatures and reset upon cooling. Thermosetting plastics retain their rigidity at elevated temperatures.

In contrast with ultrasonic cleaning, ultrasonic plastic welding requires much greater power densities, typically at hundreds of watts per square centimeter at the weld and at the contact of the tool with the workpiece. Modern plastic welders operate predominantly around 20 kHz at power outputs below 1 kW, lock automatically on the horn resonance, and keep vibrational amplitude constant for varying mechanical loads.

Commercial gear for ultrasonic metal welding was introduced in the late 1950s and originally used in the semiconductor industry. Later advances in equipment design and the need for joining high-conductivity metals spurred greater interest in the process. Standard equipment now available can weld parts up to 0.32 cm (1/8 in.) thick and larger, depending on the material and part configuration. Ultrasonic metal welds are characterized by low heat (the material does not melt as in the process of arc welding) and relatively low distortion. Because welding temperatures are well below the melting temperatures, embrittlement and formulation of high-resistance intermetallic compounds are avoided. The equipment for ultrasonic metal welding range from low power microbonders (used in semiconductor industry) operating between 40 and 60 kHz to machines of several kilowatt output capacity operating between 10 and 20 kHz for welding of larger parts. It is interesting to note that on high-conductivity materials ultrasonic welding can be over 20 times more energy efficient than resistance welding. A 5 kW welder may be equivalent to a 100-kVA resistance welder.

Agriculture

Over the last decade, ultrasound technology has considerably impacted the meat industry, with a commanding role in the rating of product value. Ultrasound is being used to measure the thickness of fat layers in pigs and cattle as part of livestock management. It is utilized to predict carcass traits such as fat cover, ribeye area, and intra-muscular fat (marbling) in live animals. Ultrasound also has been used to improve the quality of homogenized milk. A related agricultural application is pest control that includes killing of insects.

Extraction Processes

High-intensity cavitation is widely used for biological cell disruption in research and low volume processing. Near field cavitation breaks down cell walls, causing release of the cell contents into the surrounding fluid. This method is applied to extract active antigens for making vaccines and as a general tool for studying cell structure. Other extraction uses include the extraction of perfume from flowers, essential oils from hops, juices from fruits, and chemicals from plants. The equipment is normally in the range of 100 –500 W, operating at around 20 kHz. High amplitude horns are used to produce power densities of around 80 W/cm^2.

Atomization

Ultrasonic atomizers can produce small droplets of predictable size. For a given liquid, droplet size depends on atomizer frequency and it gets smaller with increasing frequency. Ultrasonic nebulizers are widely used for medical inhalation and operate between 1 and 3 MHz to produce droplets between 1 and 5 μm. Increased combustion engine efficiency and reduction of pollution have been made possible by ultrasound atomization of fuel, operating between 20 and 200 kHz.

Emulsification and Flow Enhancement

The principal advantage of ultrasonic emulsification lies in the ability to mix some immiscible liquids without additives (surfactants). Liquid flow through porous media can be increased by ultrasound and can find use in filtering and impregnation.

Ultrasonic Viscometer

The ideal liquid should not support a shear stress, but the fact is that liquids do have viscosity that gives rise to shear waves. A viscoelastic liquid combining the attributes of both fluid and solid behaviors (which produces shear stresses) is described by

$$-\frac{\partial \varepsilon}{\partial y} = \frac{1}{\eta} p_y + \frac{1}{G} \dot{p}_y \tag{17.1}$$

where p_y represents a variable shear stress; $\dot{p}_y$ is the time rate of the shear stress, ε is the fluid particle displacement, η is the viscosity coefficient; and G is the shear coefficient. The solution to Equation (17.1) is

$$p_y = -\left(\frac{\partial \varepsilon}{\partial y}\right)_0 \left(1 - e^{\frac{t}{\eta/G}}\right) \tag{17.2}$$

Equation (17.2) indicates that a periodically varying shear produces a relaxation process characterized by a time constant $\tau = \eta/G$. The associated relaxation frequency is given by

$$f_0 = \frac{1}{2\pi\tau} = \frac{G}{2\pi\eta} \tag{17.3}$$

From Equation (17.2) the attenuation of shear waves in a viscous liquid at a given frequency decreases with increasing viscosity. The damping of a vibrating shear wave transducer submerged in the liquid is a function of the coefficient of viscosity of the liquid. One way of measuring this quantity is to apply a pulse to a Y-cut crystal or to a torsionally vibrating rod immersed in the liquid so that it vibrates freely with damped harmonic motion. When the amplitude of the vibration drops to a predetermined level, another pulse is generated. The rate of pulse repetition increases with damping and hence decreases with increase in viscosity. The device is calibrated by using liquids having known coefficients of viscosity.

17.3 Ultrasound Imaging

The application of ultrasound to imaging processes is extremely important in industry and in medicine. Because imaging entails low-intensity ultrasound energy, it provides a valuable nondestructive testing technique. Ultrasonic imaging, which may be defined as any technique of providing a visible display of the intensity and phase distributions in an acoustic field, falls into a number of categories: (1) the electronic-acoustic imaging; (2) B-scanning, most commonly used in medical diagnosis; (3) C-scanning, widely used in nondestructive testing and inspection of flat and cylindrical surfaces; (4) liquid-surface-levitation presentations; (5) liquid crystal display (LCD), photographic or similar display; (6) light-refraction methods; and (7) acoustical holography.

Electron-acoustic Image Converters

The concept of an electron-image converter originated by the Russian scientist S. Ya. Sokolov (1937) who envisioned a device, similar to a video camera tube, in which the photosensitive element is replaced by a pressure-sensitive piezoelectric plate. Secondary electrons are given off when the plate is struck by a scanning beam of electrons. An ultrasound field aimed at the plate influences the electrical potentials on the faces of the plate. These potentials are proportional to the impressed acoustic pressure, with the result that the electrical potentials modulate the secondary emission of electrons. The secondary emission that results from the impingement of an electron beam is a function of the velocity of the primary electrons and also of the plate material. The ratio of the number of secondary electrons leaving the plate to the number of electrons impinging on the plate is called the *secondary emission ratio*. When the primary electron voltage is increased from zero, the secondary emission rate also increases from zero, passes through a maximum, and then decreases. With some piezoelectric materials, the maximum ratio exceeds unity and there are two velocities, one above and one below the point of maximum secondary emission, at which the ratio becomes unity. In other materials, the maximum secondary emission ratio never exceeds unity.

Smythe et al. (1953) improved on the Sokolov concept by developing an ultrasonic-imaging camera that could be operated in either an amplitude-sensitive mode or a phase-sensitive mode. In the amplitude-sensitive mode, a relatively low-voltage (160–200 V) scanning beam is used. The secondary emission ratio is less than zero, which allows the potential of the surface of the (quartz) piezoelectric plate to nearly equal that of the electron gun cathode, in absence of ultrasound. After a scan without ultrasound, the ultrasound is activated and the plate scanned again. The electrons are distributed over the surface of the plate in direct ratio to the piezoelectric voltage present at each point of the surface. A corresponding image charge forms on an anode located externally to the tube, and the video image is obtained directly from the anode charges. The ultrasound is then turned off and the negative charge on the piezoelectric surface is removed by inundating the surface

with gas ions, a process which occurs in about 0.2 s. High-voltage scanning can also be applied to remove the negative charge, followed by a low-voltage scanning to restore the surface back to the cathode potential, with the ultrasound turned off. This procedure removes the need for ion recharging and allows the use of a high-vacuum tube.

For the phase-sensitive mode of operation, higher-voltage (600 V) beams are used in conjunction with an auxiliary grid on which the secondary emission current is collected. The inner quartz surface is stabilized at approximately the same voltage as the auxiliary grid. The piezoelectric charge in the quartz plate tends toward neutralization and, consequently, the current to the quartz plate and to the collector grid fluctuates at the frequency of the ultrasonic waves. The output pressure at each point on the plate is therefore proportional to the incident ultrasonic pressure.

Other methods have been developed to reproduce the image on the piezoelectric plate in order to circumvent problems posed by high-velocity scanning-beam stabilization of an insulating surface. One method entails the use of a photoemissive surface deposited on the piezoelectric surface, with the scanning being executed by a moving beam of light. Mechanical methods also were developed, and they proved to be appreciably more sensitive than scanning with an electron beam. The surface of a piezoelectric plate subjected to an ultrasonic field is scanned mechanically using a capacitive, noncontacting electrode or by sliding a small electrode over the surface itself.

Jacobs (1962) introduced the use of electron multipliers in the camera tube of ultrasound image converters based on Sokolov's method. A schematic diagram of Jacob's device is shown in Figure 17.8. The construction provides shielding for the low-voltage circuits and virtually eliminates ground loops. The secondary emission beams produced by the scanning beam are attracted toward a positive-charged electrode in the electron-multiplier unit. The electron multiplier amplifies

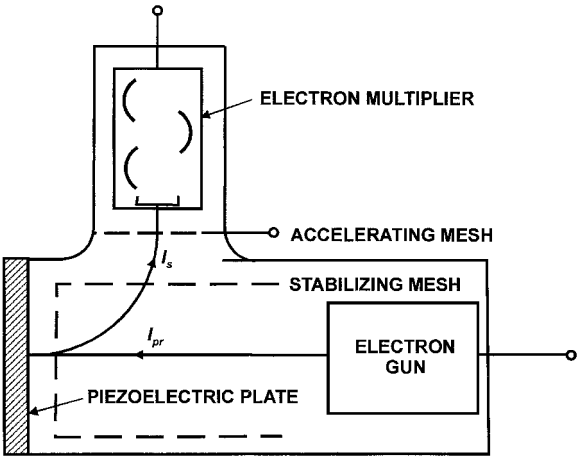


FIGURE 17.8. Jacobs device for ultrasonic scanning.

the signal currents by about 100,000 times greater than the threshold values before they are fed into the associated amplifiers. Thus, the electron multiplier serves as a wide-band amplifier with good noise characteristics. Its output constitutes the video signal that is processed through a conventional closed-circuit television system.

Two sealed ultrasonic camera tubes were developed by Jacobs: one version incorporates quartz crystals permanently sealed to the end of the tube and the features interchangeable crystals which permit operation over the frequency range from 1 to 15 MHz. Color also has been introduced to increase the sensitivity of the ultrasonic image converter, particularly in view of the fact that the human eye is more sensitive to changes in color than to differential changes in display brightness. The color display indicates both relative amplitude and phase of the ultrasonic signal undergoing analysis. In North America, the video method used is the National Television System Committee (NTSC) color broadcasting standard, and the operating frequency is 3.58 MHz.[2] A more dominant video broadcasting method is the PAL system that is used in the United Kingdom, Germany, Spain, Portugal, Italy, China, India, most of Africa, and the Middle East. Several distinguishing features of the PAL system are: (a) a better overall picture than NTSC because of the increased amount of scan lines and (b) because color was part of the standard from the beginning, color consistency between stations and TVs is considerably better. There is a down side to PAL, however, as there are fewer frames (25) displayed per second, compared with nearly 30 frames per second of the NTSC system.

Acoustic Lens

Acoustic lens are necessary in the use of electron-acoustic image converters. Acoustic lens can (a) increase the sensitivity of an imaging system through energy concentration and (b) provide coverage of a larger area by concentrating the image on the receiving piezoelectric element of the image converter in almost the same fashion as an optical lens reduces a larger picture into a smaller area. For optimal performance, the velocity of sound in the lens material must differ considerably from that of the surrounding media, and the reflection of energy at the boundary between the lens and the surrounding should be minimal. The latter condition is fulfilled automatically when the acoustic impedances of the lens materials and the surrounding medium match each other. Liquid lens of carbon tetrachloride or chloroform have the same acoustic impedance as water, but their toxicity generally precludes their use in industry. Plastic lenses also have been developed, but the sound propagation velocity in these materials exceeds that of water and they present an impedance mismatch between the lenses and water, but not to the degree of rendering them useless. The relatively high absorption of plastics limits their use to frequencies less than 15 MHz. Metallic lenses, which can be made of aluminum and other metals, possess low-absorption characteristics, so they can

[2] In the year 2006 analog NTSC television broadcasts are scheduled to cease in the United States in favor of the ATSC Digital Television Standard. This should result in higher definition video imaging.

be used at frequencies exceeding 15 MHz, but the impedance mismatch between metals and water essentially prevents their effective use.

The velocity of sound in solids is higher than in liquids, so solid concave lenses are convergent and solid convex lenses are divergent—unlike the case of light traveling in a vacuum or a gas through optical lenses.

Schlieren Imaging

Schlieren imaging has been used for many years as a tool to visualize sound fields. Advantage is taken of the physical fact that pressure gradients in an ultrasonic wave cause density gradients in the medium. When a light beam passes through these gradients, it becomes refracted. The refracted light is used in schlieren apparatus to produce an image of the sound field. Either of the two methods can be used to produce the sound field image: (a) interruption of the refracted part of the beam in order to remove it from the beam and focusing the remainder of the field onto an image detector (e.g., a ground glass screen or camera film), or (b) focusing the refracted rays on the image plane and eliminating the remainder of the beam of light from the image. The result of the first method is a dark image on a light background, and that of the second method is a light image on a dark background.

A schlieren device for direct viewing during nondestructive testing is schematically illustrated in Figure 17.9. The image is viewed directly on a ground plate glass screen. At the right-hand side of Figure 17.9, the lamp (e.g., zirconium arc), condenser lenses, a filter, first knife edge, and first collimator are mounted on a single 1-m optical bench. The second collimator, second knife edge, and camera system are mounted on another 1-m optical bench. The two benches themselves are positioned on a 0.25 m wide, 5.75 m long steel girder. The water tank is mounted apart from the optical system so that any movement or changes in the tank will not cause the optical system to go out of alignment.

The optical system must be mechanically isolated from all the other parts of the apparatus. When the device is being used, the image of the light source is centered on the first knife edge, and then the collimating lenses, tank windows, second knife edge, camera lenses, and viewing screen are centered in the light beam. Centering

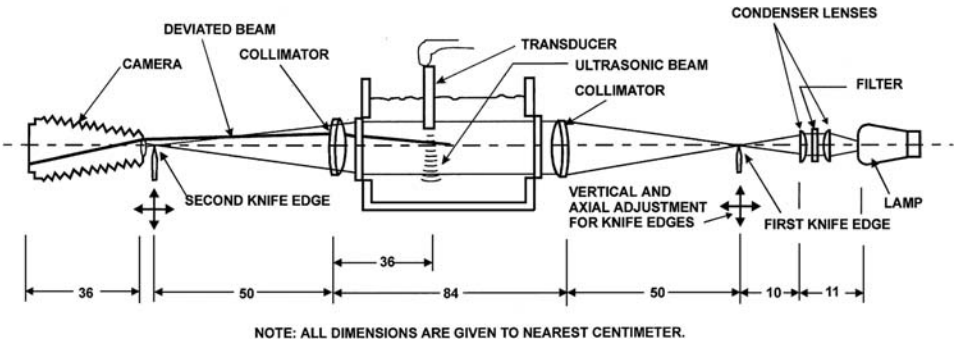


FIGURE 17.9. Schlieren device for direct viewing in nondestructive testing.

of the optics must be precise to ensure maximum sensitivity and uniform field. Through careful alignment and focusing, the field can be made to go from light to gray to dark uniformly by adjusting the second knife edge further into the light beam until it intercepts it completely. In the range of gray settings, convection currents both in air and water become clearly visible. The ultrasonic field is best rendered visible when the second knife edge is set to intercept all of the main beam, thus allowing only light refracted in the ultrasonic field to pass and illuminate the viewing plane. The optical system may also be adjusted for bright-field operation, in which situation the ultrasonic perturbations appear as dark shadows.

Color schlieren photography is useful in the study of ultrasonic waves and shock waves. Color can indicate the various pressure levels in an ultrasonic field. One method of color schlieren photography uses a spectroscopic prism between a slit located at the position of the first knife in Figure 17.9 and the first collimating lens. This method produces colors ranging from red or from blue to green. Another method developed by Waddell and Waddell (1970) produces a complete color spectrum. They eliminated the spectroscopic prism and used a color-filter matrix in place of the second knife edge and a vertical slit in place of the first knife edge. Illumination was provided by a high-pressure mercury arc lamp. The color matrix consisted of three filters, which represent the primary colors.

Liquid Crystal Imaging

Liquid crystals exhibit properties of solid crystals that are not apparent in ordinary liquids. When a stress is placed upon a liquid crystal, its optical properties change. A certain class of liquid crystals known as *nematic crystals* is used to indicate the presence of an ultrasonic field with sensitivity equal to that of schlieren systems, and with high resolution, large-area capability, and handling ease.

Ultrasonic Holography

Holography is a form of three-dimensional imaging that was conceived and developed by Dennis Gabor who received the Nobel Prize in physics for his efforts. Gabor applied his discovery to electron microscopy to overcome the problem of correcting spherical aberration of electronic lenses. The principle of holography is as follows: a diffraction diagram of an object is taken with coherent illumination and a coherent background is added to the diffracted wave. A photograph so taken will contain the full information on the changes sustained by the illuminating wave in traversing the object. The object can be reconstructed from this diagram by removing the object and illuminating the photograph by the coherent background alone. The wave emerging from the photograph will contain a reconstruction of the original wave, which seems to issue from the object. A hologram, therefore, is a recording or a photograph of two or more coherent waves. If one recorded wave is from an illuminated object and another is a reference wave, simply illuminating the hologram with the reference wave reconstructs twin images of the original object, thus giving the illusion of three dimensions.

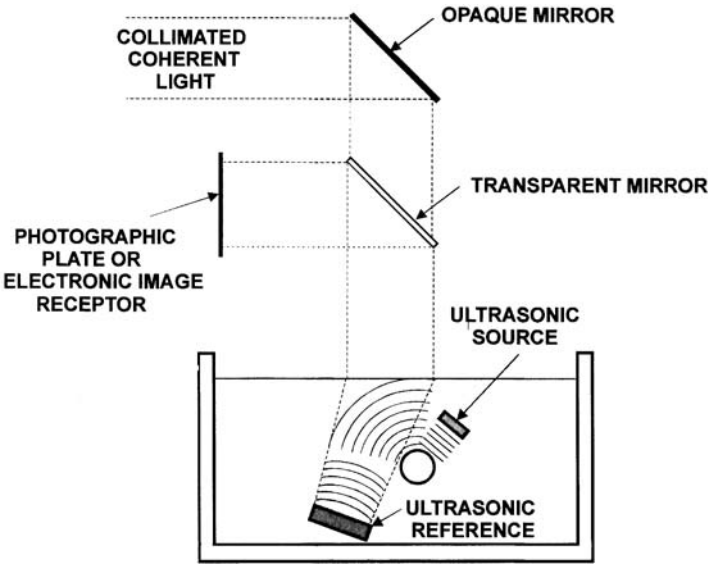


FIGURE 17.10. Ultrasonic holography making use of liquid levitation.

The wave used in the reconstruction does not have to be the original, and this allows the use of ultrasonic waves and the subsequent reconstruction of the image using light from a laser. But the size of the image changes in proportion to the ratio of the wavelength of the reconstructing wave to the wavelength of the original illuminating wave. There are a number of methods of making acoustical holograms. One method as shown in Figure 17.10 uses liquid levitation, where the acoustic waves from below the surface of a liquid forms an ultrasonic image at the surface of the liquid that can be rendered visible on a photographic plate. The reference beam may be obtained either by reflecting a portion of the irradiating beam onto the surface or by generating a separate wave through a second transducer. The height of the bulge of the liquid surface due to the impetus of an acoustic wave is critical; it should be small compared to the wavelength of the light.

17.4 Medical Uses of Ultrasound

Ultrasound use in the medical arts can be classified as being diagnostic or therapeutic. While the diagnostic procedures involving ultrasound have been in use for a number of years, ultrasonic therapeutics constitutes a newer, rapidly growing domain. The ultrasound frequencies used in medical applications range from approximately 25 kHz used in dental plaque removal to the megahertz range that is required for medical imaging. The merits of ultrasound diagnostics include safety, convenience, and capability of detecting medical conditions to which X-rays and other means of diagnosis are insensitive. Because ultrasound is much safer than

X-rays, it is used in fetal monitoring, detection of aneurysms, and echocardiography. The production of heat in the body through ultrasound is applied for its therapeutic value. The selectively greater absorption of ultrasound in cancerous tissues has proven its usefulness in hypothermic treatment of cancer. Emerging developments include the use of ultrasonic waves to perform noninvasive or "bloodless" surgery, stop internal bleeding in trauma patients, and control delivery of drugs or other compounds. Enormous advances in electronic miniaturization are resulting in fairly compact handheld diagnostic units.

Diagnostic Uses of Ultrasound

Diagnostic medical applications are based on the imaging procedures described in Section 16.7. One diagnostic technique is based on the pulse method and second diagnostic technique is based on the Doppler effect where the reflected wave is shifted in frequency from that of the incident wave impinging on a moving target. In the reflection or pulse-type equipment, A-scan, B-scan, or a combination of these two methods are utilized to present data on an oscilloscope display or process the data for permanent record. The A-scan presents echo amplitude and distance, and it is used principally in echoencephalography for the detection of midline shifts traceable to tumors or concussions. It has also been applied in obstetrics, gynecology, and ophthalmology in conjunction with B-scanning techniques. In B-scanning, the radar/sonar techniques of data processing are applied to synthesize the reflected signals into a pattern on the oscilloscope display that corresponds to a cross section of the region scanned lying in a plane parallel to the direction of beam propagation.

The position of the probe is synchronized with the sweep of one of the axes of the oscilloscope, and the echo amplitude appears as a spot with a specific intensity at a position on the screen corresponding to the position of the plane causing the echo. The TM-mode (or M-mode) is a diagnostic ultrasound representation of temporal changes in echoes in which the depth of echo-producing interfaces is displayed along one axis, and time (T) is displayed along the second axis, thus recording motion of the interfaces toward and away from the transducer.

In order to resolve small structural details, the transmitted pulse should be as short as possible, which means that the transducer must be highly damped. In order to promote good coupling between the transducer and the body, a film of oil or grease is applied at the selected spot, and care must be taken that the contacting film is free of air bubbles. In immersion procedures (such as that for kidney stone pulverization), coupling is effected through a bath of liquid, usually water. The impedance match between water and soft tissue is good, and little energy is lost in irradiating soft tissue. But the match between water and bone is poor and also the attenuation in bone is high. The result in echoencephalography is that considerable energy is lost, even where almost immediate contact is made with the skull bone.

Ultrasound densitometry is used to measure bone density in the heel, shinbone, or kneecap. It is used as a screening tool and while currently not as precise as absorptiometry techniques, it is still effective, inexpensive, portable and uses no

radiation. This technique is primarily used as a screening tool to predict bone fracture risk or to determine the need for single-energy X-ray absorptiometry (SXA) or peripheral dual-energy X-ray absorptiometry (pDEXA) to ascertain the presence of osteoporosis (degeneration or decalcification of the bone).

Ultrasound imaging is being extensively used to study cardiac functions. One use of the A-scan technique is to monitor for early signs of rejection following a heart transplant. As the heart fills at the onset of rejection, the muscle walls swell and stiffen. Measurements are made using a 2-cm diameter, 2.25 MHz transducer at a pulse repetition rate of 1000/s. Echo indications from the anterior wall and the posterior wall supply the measurement information, as the distance between the two walls indicates overall heart size. Echocardiography is useful in diagnosing pericardial effusion (escape of fluid from a rupture in the pericardium) because the echo received from the posterior wall of the heart is split when the transducer is located on the anterior chest surface.

Echocardiography is also useful in assessing the degree of stenosis (narrowing of opening) in mitral valves. The transducer is aimed at the anterior mitral leaflet and the echo signal can record on a strip chart so that an upward movement of the recorder pen corresponds to a movement toward the transducer and a downward movement corresponds to a movement away from the transducer. The slope of the tracing indicates the velocity of motion. The degree of stenosis affects the blood flow rate through the opening. A number of symptoms can be discerned with the use of this method; for example, rigidity or calcification of the mitral valve is indicated by decrease in the total amplitude of the anterior mitral leaflet between the closed position during ventricle systole and the position of maximum opening in early diastole.

Intercardiac scanning is used to obtain plan-position (C-scan) displays for the interior of the heart. A tiny probe is inserted into the right atrium through the external jugular vein or the femoral vein. An advanced version of the probe consists of many elements so that scanning can be achieved by sector techniques in which the positions of points on the recorded image are correlated with beam direction. The data from the transducer can be processed by computer and displayed. A-scanning may be combined with C-scanning to provide 3D information. The motion of selected regions of the heart can even be viewed on a video display, particular with the use of focusing transducers, which can provide high-resolution images of the heart.

A tomographic method of observing interior structures of the heart three-dimensionally is based on a stereoscopic display of two-dimensional images. The ultrasonic device functions in synchronism with the cardiac cycle to obtain phase-specific tomograms, which then can be displayed on a storage CRT. Tomographic systems are also used to investigate other internal organs such as right ventricle, atrium, and kidney cysts that have irregular shapes.

Accurate diagnosis is rendered possible in the field of ophthalmology through the use of ultrasound to diagnose conditions existing in the soft tissues of the orbit of the light-opaque portions of the eye. Focused transducers are used with frequency typically being at 15 MHz. This method can outline tumors, and detached retinas,

measuring the length of the axis of the eye, and detect foreign bodies close to the posterior eye wall. One instrument combines a diagnostic transducer for locating foreign bodies with a surgical instrument for removing an object, which enables rapid removal of foreign bodies from the eye by directing the surgical tool to the object with least damage to the eye.

In the field of neurology, ultrasonic echoencephalography provides an immediate means of detecting lateral shifts in the midline septum caused by tumors or concussion. It is notable that every emergency ambulance in Japan is outfitted with echoencephalographic equipment in order to identify victims with possible subdural hemorrhage so that they may be transported directly to special neurological units for treatment. Posttraumatic intercrannial hemorrhage and skull and brain trauma can be quickly diagnosed and lesions can be located rapidly without discomfort to the patient. Ultrasonic pulses are transmitted through the temples. In the A-scan mode, echo indications from the midline and the opposite temple areas of the skull are presented on an oscilloscope screen. A shift in the midline is readily discernible. B-scans, which could provide more information, are more effective with small children and infants because their skulls are soft and have attenuation coefficients lower than those in adults.

The ultrasonic B-scan technique for examining the abdomen is useful for detecting pelvic tumors, hydatiform moles, cysts, and fibroids. It is also used to diagnose pregnancy at 6 weeks (counted from the first day of the last menstrual period) and afterwards. The obstetrician can follow the development of the fetus throughout the pregnancy, including size and maturity and positioning of the placenta. The presence of twins or multiple pregnancy is also revealed. This avoids the need for X-rays and the attendant danger of irradiation to mother and child. Fetal death can be confirmed much earlier by ultrasound than it can by radiography.

With improvements in ultrasonic and computer technology, 3D visualization began to appear in the early 1980s. Some work came from the domain of cardiologists where initial efforts were directed to determining the volume of cardiac chambers. Real-time scanner probes mounted on articulated arms were often employed where positions of the probe can be accurately established. The principle of 3D imaging has always been to stack parallel image sections together with their positional information into a computer.

Ultrasonic diagnosis by echo methods extends to all parts of the body. Air and other gases have much lower impedance than either liquid or solids, so air cavities produce distinct echoes. Stationary air embolisms can be so identified because they are located in areas usually filled with fluids. Moving embolisms can be identified by Doppler methods. Gallstones have an acoustic impedance equivalent to that of bone and so can provide good ultrasonic echoes. The ultrasonic B-scan technique has been proven effective in diagnosing thyroid disorders. Disorders detected ultrasonically include neoplastic lesions including cystic nodules, solid adenoma and carcinoma, nonneoplastic lesions, subacute thyroidiis, and chronic thyroiditis.

The Doppler method takes advantage of the fact that a shift in frequency occurs when an ultrasonic wave is reflected from a moving target. Also any variation in

fluid motion that causes a beam of ultrasound to be deflected causes a Doppler shift in frequency. One application of the Doppler shift principle is for the detection of fetal blood flow and heartbeat, which can provide a mother the exciting assurance that her unborn baby is alive by listening to its heartbeat through a set of headphones connected to the detecting equipment. Blood flow is measured through intact blood vessels through the use of the Doppler principle with the carrier frequency ranging from 2–20 MHz. The Doppler frequency signal is calibrated in terms of velocity, according to the pitch of the Doppler signal. Doppler-type diagnostic units are used to determine the severity of atherosclerosis, locating congenital heart defects, and continuously monitoring fetal heartbeat during birth.

In 1991 researchers at Duke University developed a *matrix array scanner* for imaging the heart. Three years later, the world's first electronically steered matrix-array 3D ultrasound imager was developed and it became commercially available in 1997. The matrix array transducer, which steered the ultrasound beam in three dimensions, contained 2000 elements of which 512 were used for image formation. Newer units became available commercially for providing 3D scans of fetuses in "color." This resulted in a new market for "re-assurance" scans (sometimes called "entertainment scans"), as the attraction of being able to see a baby before birth apparently is proved to be quite irresistible to parents-to-be (as well as grandparents-to-be). Three-dimensional ultrasound has been helpful in determining the topographical geography of fetal heart valves while blood is flowing. Another application is the imaging of microcirculation, which is important in the study of artherosclerosis, diabetes, and cancer—this means that 3D ultrasound is a powerful tool for ascertaining the progression of diseases in the body. As recently as 2003, researchers at Norway's SINTEF Unimed Ultrasound (Kaspersen, 2003) have demonstrated that real-time 3D ultrasound can be used as a diagnostic imaging tool during surgery. These researchers have applied 3D ultrasound to laparoscopic, neurological, and vascular surgeries.

Use of Contrast Agents

In the last decade, researchers in academia, pharmaceutical companies, and scanner manufacturers invested manpower and funding in developing efficacious contrast agents and new contrast specific modalities. Contrast agents can improve the image quality of sonography either by decreasing the reflectivity of the undesired interfaces or by increasing the backscatter echoes from desired regions. In the former approach, the contrast agents are taken orally, and for the latter approach, the agents are introduced vascularly. In the upper GI tract, sonographic assessment is limited by the gas-filled bowel, which produces shadowing artifacts. One agent, recently approved by FDA, is SonoRx®, a simethicone-coated cellulose. Vascular enhancing ultrasound entails the injection of contrast agent, such as perfluorochemicals that have low solubility in blood and high vapor pressure.

One of the most important clinical uses of ultrasound contrast is in cardiology, where it will potentially compete with thallium nuclear scans. Newer agents such as Optison, Definity, and Sonazoid can produce myocardial perfusion images in

humans. This is clinically significant, because visualization of the myocardial flow permits direct assessment of underperfused or unperfused regions (i.e., areas of ischemia or infarction) in patients with a history of chest pain. Myocardial imaging using ultrasound contrast agents provides an assessment of the coronary arteries and of the coronary blood flow reserve, as well as collateral blood flow that may exist.

Any body cavity that can be accessed can, in principle, be injected with vascular contrast. According to Shi et al. (2005), the most successful application in this category is hysterosalpingo-contrast sonography (abbreviated HyCoSy) for the evaluation of fallopian tube patency. Degenhardt et al. (1996) reported on 103 patients with fertility problems who underwent transvaginal sonography and HyCoSy with Echovist 14. In 58 cases, HyCoSy was compared with conventional, more invasive techniques such as chromolaparoscopy and 91% agreement was found. The upshot is that HyCoSy is rapidly becoming the screening test of choice to determine tubal patency.

Safety of Ultrasonic Diagnosis

Extensive studies on the safety of ultrasound in Japan and elsewhere have led to issuance of Japanese industrial standards (JIS) for diagnostic ultrasound devices. These standards place limits on various diagnostic procedures:

1. Ultrasonic Doppler fetal diagnostic equipment, 10 mW/cm^2 or less.
2. Manual scanning B-mode ultrasonic diagnostic equipment, 10 mW/cm^2 or less for each probe.
3. Electronic linear scanning B-mode ultrasonic diagnostic equipment, 10 mW/cm^2 or less in a single aperture.
4. A-mode ultrasonic diagnostic equipment, 100 mW/cm^2 or less: This standard is limited to diagnosis of the adult head and is not for pregnancy where B-mode is the principal technique used. This higher permissible level is due to attenuation through the adult skull bone.
5. M-mode ultrasonic diagnostic equipment, 40 mW/cm^2 or less: The M-mode is used in clinical diagnosis of the heart. For a combination of the M-mode with the B-mode, the intensity is limited by the B-mode standard, i.e., 10 mW/cm^2.

In 2000, the Safety Group of the British Medical Ultrasound Society issued the following statement on the safe use and potential hazards of diagnostic ultrasound (Ter and Duck, 2000):

- Ultrasound is now accepted as being of considerable diagnostic value. There is no evidence that diagnostic ultrasound has produced any harm to patients in the four decades that it has been in use. However, the acoustic output of modern equipment is generally much greater than that of the early equipment and, in view of the continuing progress in equipment design and applications, outputs may be expected to continue to be subject to change. Also, investigations into the

possibility of subtle or transient effects are still at an early stage. Consequently diagnostic ultrasound can only be considered safe if used prudently.

- Thermal hazard exists with some diagnostic ultrasound equipment, if used imprudently. A temperature elevation of less than 1.5°C is considered to present no hazard to human or animal tissue, including a human embryo or fetus, even if maintained indefinitely. Temperature elevations in excess of this may cause harm, depending on the time for which they are maintained. A temperature elevation of 4°C, maintained for 5 minutes or more, is considered to be potentially hazardous to a fetus or embryo. Some diagnostic ultrasound equipment, operating in spectral pulsed Doppler mode, can produce temperature rises in excess of 4°C in bone, with an associated risk of high temperatures being produced in adjacent soft tissues by conduction. With some machines colour Doppler imaging modes may also produce high temperature rises, particularly if a deep focus or a narrow colour box is selected. In other modes, temperature elevations in excess of 1°C are possible, but are unlikely to reach 1.5°C with equipment currently in clinical use, except where significant self-heating of the transducer occurs.
- Non-thermal damage has been demonstrated in animal tissues containing gas pockets, such as lung and intestine, using diagnostic levels of ultrasound (mechanical index values of 0.3 or more). In view of this, it is recommended that care should be taken to avoid unnecessary exposure of neonatal lung , and to maintain MI as low as possible when this is not possible. In other tissues there is no evidence that diagnostic ultrasound produces nonthermal damage, in the absence of gas-filled contrast agents. However, in view of the difficulty of demonstrating small, localised, regions of damage *in vivo*, the possibility of this cannot be excluded. The Mechanical Index, if displayed, acts as a guide to the operator. The use of contrast agents in the form of stabilised gas bubbles increases the probability of cavitation. Single beam modes (A-mode, M-mode, and spectral pulsed Doppler) have a greater potential for nonthermal hazard than scanned modes (B-mode, Colour Doppler), although the use of a narrow write-zoom box increases this potential for scanning modes.

Therapeutic Uses of Ultrasound

We may very well be witnessing at this time only the beginning of the use of ultrasound for therapeutic purposes. Some techniques, such as the use of 25 kHz ultrasound combined with a water jet to remove plaque from teeth and the cleaning of dental and medical tools with ultrasound, have been well established for a number of years. Ultrasonic nebulizers of pharmaceuticals operate without producing destructive temperature levels. Athletic centers and sports medicine specialists make use of ultrasound devices to heat sore muscles. Newer techniques are arriving on the market or are still in the testing stages. One example is the use of ultrasound in catheters to ream out arteriosclerositic deposits in arteries that is still very much in the experimental stage.

One therapeutic use of ultrasound already in widespread clinical use, *extracorporeal shock wave lithotripsy*, has completely changed the treatment of kidney

stones. Kidney stones are calcified particles that tend to block the urinary tract. In this type of treatment, the patient is immersed in water to equalize as much as possible the acoustic impedances between the transducer and the patient's body. A focused, high-pressure ultrasonic pulse is directed through the water and into the patient's torso to break the stone into small pieces. The pulverized material can now pass out of the body unhindered. Lithotripsy causes very little damage to kidney tissue.

A promising procedure for therapeutic ultrasound is the laser-guided ablative acoustic surgery, in which sound supersedes the scalpel in destroying benign or malignant tissues. The ultrasound focused by a specially shaped set of transducers converges inside the body to create a region of intense heat that can destroy tumor cells. The spot of destruction is so small that a boundary of only six cells lies between the destroyed tissue and completely unharmed tissue, which connotes a precision far beyond any current method of surgical incision.

Acoustical surgery offers a potentially better means of treating cancerous tumors because it does not require an anesthetic, can be administered in a single treatment, and causes no observable side effects. In a Phase I clinical trial at Marsden Hospital in London, focused sound waves destructed parts of liver, kidney, and prostate tumors in 23 patients. In the next phase the researchers will attempt to fully destroy tumors in the liver and prostate. Also under testing is the Sonablade^{TM} system by Focus Surgery of Indianapolis, IN, which incorporates proprietary transducer technology in a transrectal probe that provides imaging for tissue targeting and high-intensity focused ultrasound (HIFU) for tissue ablation. After the operator defines the area of periutheral tissue to be ablated, the treatment process begins under computer control, where the focus of the dual-function transducer is electromechanically stepped through the designated volume of the tissue. HIFU results in thermally induced coagulative necrosis only in the intraprostatic tissue encompassed by the focal volume, with effect on intervening tissues. Confirmation of targeting accuracy is provided through continuously updated images. The necrotic tissue is either sloughed during urination or reabsorbed, along with the cessation of patient symptoms.

Ultrasound can also be used to stop internal bleeding through an effect called *acoustic hemostatis*. With sufficient power, ultrasonic pulses can elevate the body temperature at selected sites from 37°C to between 70°C and 90°C in an extremely short time, less than 1 s. This causes the tissue to undergo a series of phase transitions, and the protein-based bodily fluids and blood coagulate as the result of the proteins undergoing cross-linking (a process similar to cooking an egg).

A research team at University of Washington's Applied Physics Laboratory is investigating the use of ultrasound to stop internal bleeding during surgery and for treating trauma cases. The present method of stemming bleeding in delicate organs, such as the liver, pancreas, or kidney, is through cauterization on the surface with ion or microwave systems. The focused ultrasound waves, however, can penetrate deeply into the organ and "cook" the tissue in a layer as thin as 1 mm. It follows that trauma patients could be treated without the need for a sterile environment of an operating room and without the danger of infection that accompanies conventional

surgery. To date, success has been achieved by the University of Washington group in identifying patients with internal bleeding and in the use of HIFU in the operating room to stop bleeding in the organs and vessels of animals.

Another promising technology still under development testing for use in biopsies is that of *augmented reality system*, in which 3D ultrasound technology is combined with virtual reality (State et al., 1996). This concept uses ultrasound echography imaging, laparoscopic (referring to the use of a small video camera to perform surgery) range imaging, a video see-through head-mounted display (HMD), and a high-performance graphics computers to generate live images through the use of computer-generated imagery with the live video. An augmented reality system presents live ultrasound data or laparoscopic range data in real time while recording data from the part of the patient that is being scanned.

Advances in magnetic resonance imaging (MRI) and diagnostic ultrasound imaging will allow these two areas to be combined. Manufacturers of MRI equipment are developing models that combine therapeutic functions such as ultrasound with imaging processes. One company (Therus Corporation, Seattle, WA) is combining ultrasound diagnostic imaging with a separate therapeutic ultrasound capability to produce a relatively small, portable system that can be carried by paramedics and rescue workers for use at the site of disaster.

A newer method of targeted drug delivery is that of *sonophoresis*, which uses sound waves instead of needles to inject drugs such as insulin and interferon through the skin. The high-frequency waves open tiny holes in cell membranes, thus rendering the cells temporarily permeable in localized regions and allowing better penetration of the drug into the blood vessels below the skin. This results in greater effectiveness of the drug, lessens the dosage requirements and toxicity, and allows for more precise localization of drug delivery. Although the mechanisms by which ultrasound augments these effects are only partially understood, it is known that ultrasound produces biophysical reactions yielding hydroxyl radicals that in turn affect cell membranes. A system developed by Ekos Corporation of Bothell, WA, to dissolve life-threatening blood clots constitutes an early application of ultrasound drug delivery. The Ekos device injects thrombolytic drugs through a catheter using low-energy, localized ultrasound to the target site in the body. Thrombolytics, which are generally administrated to break up clots, are dramatically more effective with the use of ultrasound because the sound waves help to concentrate the drugs at the site of the clot. Ekos's device and similar products are meant to provide treatment of cardiovascular obstructive diseases (e.g., stroke and arterial and deep-vein thrombosis). In these types of ailments, rapid response is critical, and therapy necessitates applications of massive doses of clot-dissolving drugs over 72 h. Ultrasound may also be used with antiresrenosis agents, which are intended to prevent coronary arteries from reclosing after angioplasty—a procedure in which a balloon-tipped catheter is inserted into a clogged vessel and the balloon is inflated to open up the vessel.

Another long-term application is the use of ultrasound to deliver insulin through the skin for treating diabetes. It may be even possible through ultrasound to penetrate the blood–brain barrier, which insulates the brain from foreign substances

and also prevents many drugs from reaching diseased tissues there, so that the effects of chemotherapy can be enhanced.

Higher frequencies of ultrasound can cause the transfer of large molecules such as DNA to migrate among cells, as the result of *sonoporation*, which enhances the porous effect on cell membranes by induction of ultrasonic shock waves through a lithotripter. One pharmaceutical company (ImaRx, Tucson, AZ) developed a gene-delivering system for improving gene expression, i.e., revising a gene's genetic code to make a specific protein. The ImaRx system employs acoustically driven microbubbles that carry the gene and fluorinated compounds that serve as markers to detect the DNA. The ultrasound assists in delivering the DNA to targeted areas in the body and in tracking its progress through the use of contrasting agents injected with the microbubbles of genetic material.

References

AIUM. 2000. Mechanical bioeffects from diagnostic ultrasound: AIUM consensus statements. *Journal of Ultrasound in Medicine* 19/2. Bethesda, MD: American Institute of Ultrasound in Medicine.

Bachman, Donald M., Crewson, Philip E., and Lewis, Rebecca S. 2002. Comparison of heel ultrasound and finger DXA to central DXA in the detection of osteoporosis. Implications for patient management. *Journal of Clinical Densitometry* 5(2): 131–142.

Barnett, S. B. and Kossoff, G. (eds.). 1998. *Safety of Diagnostic Ultrasound*. Progress in Obstetric and Gynecological Sonography Series. Parthenon.

Barnett, S. B. (ed.). 1998. WFUMB Symposium on Safety of Ultrasound in Medicine. Conclusions and Recommendations on Thermal and Mechanical Mechanisms for Biological Effects of Ultrasound. *Ultrasound in Medicine & Biology* 24, Supplement 1.

Baum, G., Cruz, B., and Rosenblatt, R. 1980. Advantages of ultrasound mammography. *Ultrasound in Medicine & Biology*. Amsterdam: Excerota Medica.

Bruining, N., Burgelen, C. V., De Feyter, J., et al. 1998. Dynamic imaging of coronary structures: an ECG-gated three-dimensional intercoronary ultrasound in humans. *Ultrasound in Medicine & Biology* 24: 631–637.

Brown, B. and Goodman, J. E. 1965. *High Intensity Ultrasonics*. London: Iliffe.

Crum, Lawrence A. 2004. Therapeutic Ultrasound. *Proceedings of the Advanced Metrology for Ultrasound in Medicine Meeting*, April 27–28, 2004. Teddington, UK.

Chen, L., Ter Haar, Gail, Hill, C. R. et al. 1998. Treatment of implanted liver tumors with focused ultrasound. *Ultrasound in Medicine & Biology* 24: 1475–1488.

Dalke, H. E. and Welkowitz, W. J. 1960. *Journal of the Instrument Society of America* 7(10): 60–63.

Degenhardt, F., Jibril, S., and Eisenhauer, B. 1996. Hysterosalpingo-contrast sonography (HyCoSy) for determining tubal patency. *Clinical Radiology* 51(s1): 15–18.

Ensminger, Dale. 1988. *Ultrasonics: Fundamental, Technology, Applications*, 2nd ed. New York: Marcel Dekker.

EPSUMB Study Group. 2004. Guidelines for the use of contrast agents in ultrasound. *Ultraschall in Medizine* 25: 249–256.

Hope, Simpson D., Chin, C. T., and Burns, P. N. 1999. Pulse inversion Doppler: a new method for detection of nonlinear echoes from microbubble contrast agents. *IEEE Transactions UFFC* 46: 372–382.

Jacobs, J. E. 1962. *Proceeding On Physics and Nondestructive Testing*. October 2–4, 1962. San Antonio, TX: Southwest Research Institute: 59–74.

Jacobs, J. E., Reimann, K., and Buss, L. 1968. *Materials Evaluation* 26(8): 155–158.

Joyner, Claude R, (ed.). 1974. *Ultrasound in the Diagnosis of Cardiovascular-Pulmonary Disease*. Chicago: Year Book Medical Publishers, Inc. (written by clinicians primarily for clinicians.)

Kaspersen, J. H. 2003. Clinical trials with new navigation system: Custus X. SINTEF [On line, available at: http://www.sintef.no/static/UM/UL/cx/cx.html].

Krüger, J. F. von and Evans, David H. Doppler ultrasound tracking instrument for monitoring blood flow velocity. *Ultrasound in Medicine & Biology* 28: 1499–1508.

Needleman, L. and Forsberg, F. 1996. Contrast agents in ultrasound. *Ultrasound Quarterly* 13: 121–138.

Ouellette, Jennifer. 1998. New ultrasound therapies emerge. *The Industrial Physicist,* pp. 30–34.

Shahbender, R. A. 1961. *Transactions of I. R. E.*, UE-8: 19.

Smythe, C. N., Poynton, F. Y., and Sayers, J. F. 1953. *Proceedings of IEEE* 110(1): 16–23.

Shi, William T., Forsberg, Flemming, Liu, Ji-Bin, et al. 2005. Orlando, FL: Annual Meeting of the Academy of Molecular Imaging.

Sokolov, S. Ya. 1937. U.S. Patent No. 2,164,185.

State, Andrei, Livington, Mark A., Hirota, Gentaro, et al. 1996. Technologies for augmented reality systems realizing ultrasound-guided needle biopsies. *Proceedings of SIGGRAPH 96* (New Orleans, August 4–9, 1996). In: *Computer Graphics Proceedings*, Annual Conference Series 1996. ACM SIGGRAPH: 439–446.

Sternberg, M. S. 1958. *Physics Review* 110: 772.

Ter, Haar G. and Duck, F.A. 2000. *The Safe Use of Ultrasound in Medical Diagnosis*. London: British Medical Ultrasound Society/British Institute of Radiology.

Waddell, J. H. and Waddell, J. W. 1970. *Research and Development* 30: 32.

Woo, Joseph. 2002. A short history of the development of ultrasound in obstetrics and gynecology. [On line, available at: http://www.ob-ultrasound.net]. (An excellent three-part not-so-short compendium of the history of the use of ultrasound in these medical fields.)

Worlton, D. C. 1957. *Nondestructive Testing* 15: 218.

Problems for Chapter 17

1. Can you think of a household application for ultrasonic cleaning?
2. A "sing around" flowmeter is used to measure the flow of a chemical slurry inside a pipe. The two transducers are placed 10 cm apart. A 25,000 Hz pulse is initiated and the received frequency is 24,500 Hz. Find the velocity of the slurry.
3. If the fluid went the other way, what would be the receiving frequency for the same setup of Problem 2 emitting 25,000 Hz?
4. A beam-deflection flowmeter is used to measure flow inside a nuclear reactor heat exchanger pipe. The velocity of sound in the liquid coolant is known to

be 2550 m/s, and the deflection of the beam is 8.58°. Find the velocity of the fluid.

5. We can also measure the velocity of sound in a fluid by the beam-deflection method. It is known that a gas is passing through a conduit at the rate of 120 m/s. The deflection was measured to be 0.056 radian. Estimate the speed of sound in the gas.
6. Two transducers are set apart 21.2 cm in a nondispersive medium. The two resonance frequencies are 2164 Hz and 4617 Hz. Find the sound propagation speed of the medium.
7. How could motion sensing be useful in a home garage? Where else in a home would you use a sensor?
8. Why is ultrasound considered to be safer than X-rays? Would you expect the ultrasound equipment to be more expensive or cheaper than X-ray gear?
9. Why is it preferable to use ultrasound to examine the fetus in the womb to X-rays? How would the images compare in quality?
10. What is the principal difference between high-intensity and low-intensity ultrasound as used in the human body or on a live animal?
11. What are the two primary factors in controlling ultrasound dosage?

18
Music and Musical Instruments

18.1 Introduction

From time immemorial music has impacted humanity in many ways. In moments of sadness, music provides solace; in happier times, enhances exhilaration; during stressful periods, a greater sense of calm intertwined with an intensified feeling of purpose; and when diversion is needed, entertainment. Music reached its greatest heights through the evolution of primitive contraptions into more elegant instruments and the emergence of great composers such as Monteverdi, Vivaldi, Bach, Handel, Haydn. Mozart, Beethoven, Verdi, and in more modern times, Rimsky-Korsakov, Stravinsky, Mahler, Ravel, Gershwin, Schoenberg, and Ellington.

Music has given rise to a whole slew of industries: the manufacture and marketing of instruments; staging of performances which can range from solo appearances to a lavish operatic production or a frenzied giant rock concert, on stage or through electronic transmission (radio, television, the Internet); and distribution of recorded media (tape, CD, or DVD) and playback equipment.

In surveying the musical scene, we must realize the manner in which music influences people cannot possibly be adequately gauged, and this constitutes a situation that provides a fertile field of research in psychology and anthropology, to say nothing of musicology and music theory. In the medical field, classical music serves as a valuable tool in psychotherapy. *Musical acoustics* is an extremely broad interdisciplinary field—a field that deals with the production of musical sound and the transmission of musical sound to the listener as well (Rossing, 1990). The study of music from a physical approach provides the opportunity to bridge the gap between art and science.

In this chapter the structure of music is examined from an acoustical approach and the principles of generating musical tones with instruments, which also includes the human voice, are outlined. Many but not all musical instruments are described herein, as well as the makeup of orchestras and bands.

18.2 Musical Notation

As pointed out by Olson, music can be memorized and passed from one person to another by the direct conveyance of the sound, but this does not constitute a satisfactory, nor an efficient method of communicating music to performers and preserving the music for future performances (Olson, 1967). Accordingly, musical notations were developed to use symbols on paper to denote frequency, duration, quality, intensity, and other tonal characteristics. A human can typically distinguish 1400 discrete frequencies, but in the equally tempered musical scale, there are only 120 discrete tones ranging from 16 Hz to 16,000 Hz. Pitch is an attribute of aural sensation, dependent on the frequency of the sound. Musical tones are assigned specific values, allowing for specific frequency values, and leaving out the "in-between" values. A principal reason that we can identify a musical instrument is that the musical instruments are essentially resonant instruments and therefore response only to certain frequencies. These resonant frequencies are fixed and cannot be altered, except for certain instruments such as those members of the violin family and the trombone. Moreover, when same notes are played on different instruments, the overtones differentiate one instrument from another. With the relatively small number of fundamental frequencies designated in Western music, matters are greatly simplified in designating the discrete frequency characteristics of tones.

In the five-line *staff* of Figure 18.1, the pitch of a tone is denoted by placing notes, 𝅝, 𝅗𝅥, ♩ on the lines and in the spaces between the lines. The *pitch range* of a set of lines is designated by the *clef* (𝄞 or 𝄢) which is placed at the left of the staff. The most common clefs are the *treble* or *G clef* and the *bass* or *F clef*, both of which are shown in Figure 18.1. The notes are designated alphabetically from A to G. The interval in pitch between two notes with the same letter is the *octave.* As explained earlier in this text, the two sounds separated by an octave have a fundamental frequency ratio of 2. The pitch interval between adjacent notes (i.e., between a note on a line and a note in the adjacent space, as designated by successive letters) is a whole tone in the equally tempered scale. Pitches that are higher and below the staff are designated by notes written upon and between short lines called *leger lines*, as shown in Figure 18.1. The number of leger lines can hypothetically be extended without limit. The sign 8va above the staff denotes that all tones are to be played an octave higher than their placement and, conversely, the sign 8vs placed below the staff indicates that all tones are to be played an octave lower. This is shown in Figure 18.2.

There is also another clef—the *movable* or *C clef*—which was meant to accommodate music instruments with extended range (e.g., the bassoon, cello, or viola). In older musical manuscripts there are C-clefs for the soprano, alto, and tenor parts, Figure 18.3 shows the three different symbols to indicate the C-clef. The C-clef is placed on the middle C line. The old C-clef positions for the soprano, alto, and tenor are given in Figure 18.4. The position of the C clef always corresponds to the middle C.

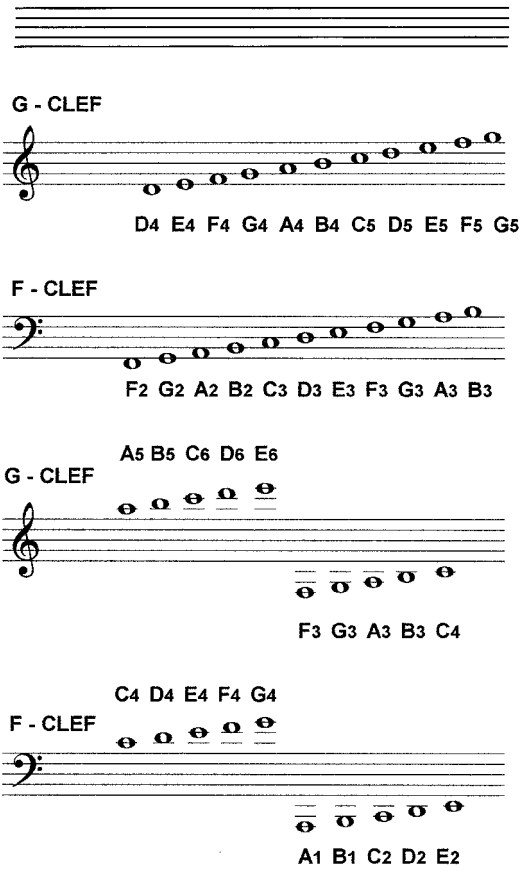


FIGURE 18.1. The five-line musical staff, the notes of the most common bass, and the treble clefs and the leger lines (the lines above and below the staff) for the treble and bass clefs.

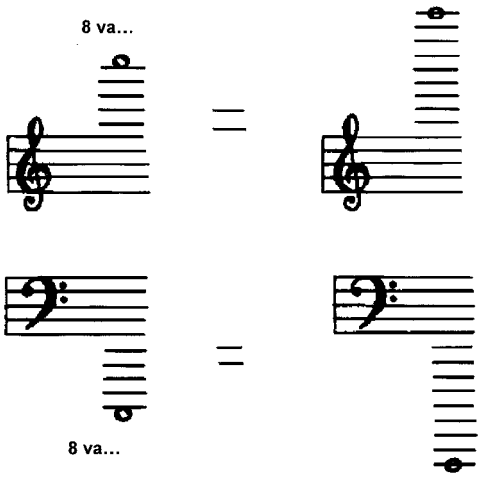


FIGURE 18.2. The position of 8va above the staff indicates that all notes are sounded an octave higher. When 8va is located below the staff, the notes are to be sounded an octave lower.

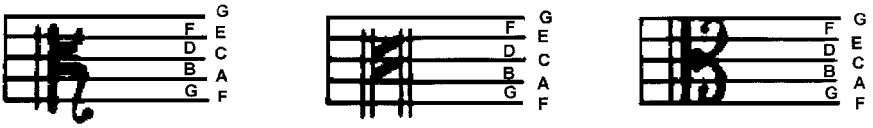


FIGURE 18.3. Three symbols used to indicate the C, or movable, clef.

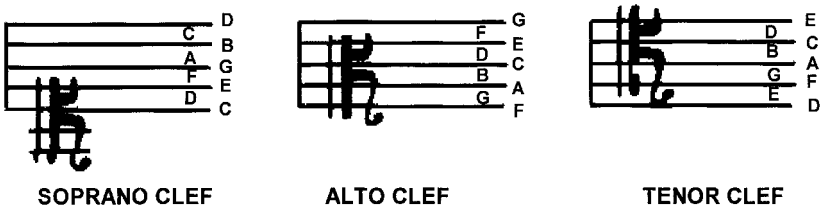


FIGURE 18.4. The positions of the C-clef for the soprano, alto, and tenor clefs.

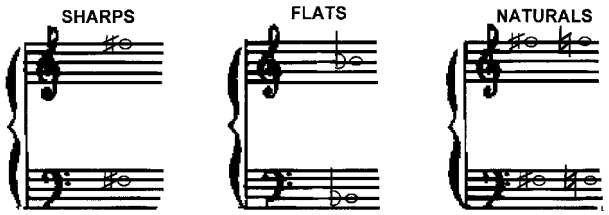


FIGURE 18.5. Sharp, flat, and natural designations: the sharp raises a pitch by a semitone, the flat lowers a pitch by a semitone, and the natural nullifies a sharp or flat to restore a note to normal.

A note can be moved up in its pitch a half step or *semitone*; this is labeled a *sharp*. The sharp is designated by the natural note preceded by the ♯sign, as shown in Figure 18.5. A note can also be moved down a semitone, thus rendering that note a flat with the symbol ♭, which is also shown in Figure 18.5. A natural designation ♮ nullifies a sharp or a flat, returning the note to normal. A note may be moved up by a whole step by a double sharp designation ♯♯ or $\times$. A note may be moved down a whole note by a double flat ♭♭. A standard system for the identification of tones used in music is given in line 1 of Figure 18.6 (Young, 1939). The other seven lines represent the various systems for identifying the musical tones without the benefit of using the staff, but the system represented by the first line is the most logical one to use and understand. The reference standard frequency C_0 is 16.352 Hz, which just about constitutes the lowest frequency that a human ear can detect. It is customary to consider C as the point to begin counting whole octaves. Figure 18.7 displays the frequencies of the notes in equally tempered scale in the key of C from 16 Hz to 16 kHz.

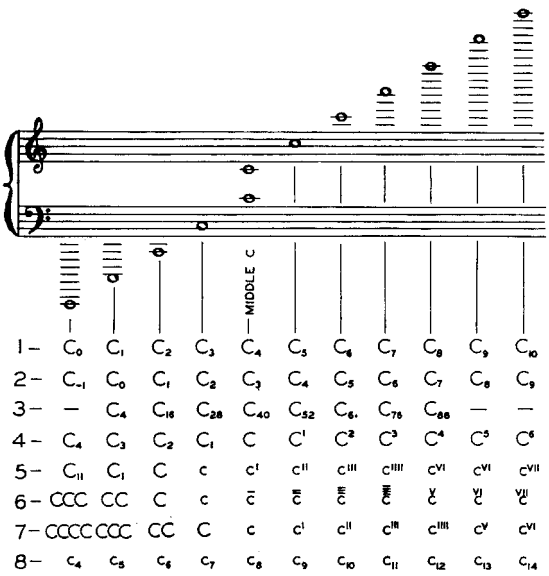


FIGURE 18.6. Eight systems used for tone identification in music (Young, 1939).

18.3 Duration of Musical Notes

The duration of a musical tone is the length of time assigned to it in the musical composition. Figure 18.8 displays the symbols used to indicate duration. While the pitch of a tone is given by its position on the staff, its length is assigned by the choice of one of the symbols of Figure 18.8. However, the magnitude (i.e., its duration) of a tone is not rigidly fixed and it may vary from composition to composition. But nevertheless, in a particular composition the duration of each tone is kept in proportion to the magnitude of a whole note.

In the traditional musical notation a vertical bar is drawn across the staff. The time interval between two vertical bars in a staff is called a *measure*, or the less precise but more commonly used *bar*. The time intervals of all measures within a composition are usually equal. If two whole notes constitute a measure, then the measure will need four half notes, or eight quarter notes, or any combination that adds up to two whole notes in time interval. A double bar that consists of two vertical bars across the staff denotes the end of a division, movement, or an entire composition.

To indicate periods of silence in a composition, one or more *rest* symbols of Figure 18.9 is used to indicate the duration of the silence. The duration of a whole rest is equal to that of a whole note, the duration of a half rest is equal to that of a half note, and so forth.

The duration of a tone represented by a note or a rest of a certain denomination can be modified by the addition of a dot to the note. The effect of the dot is to

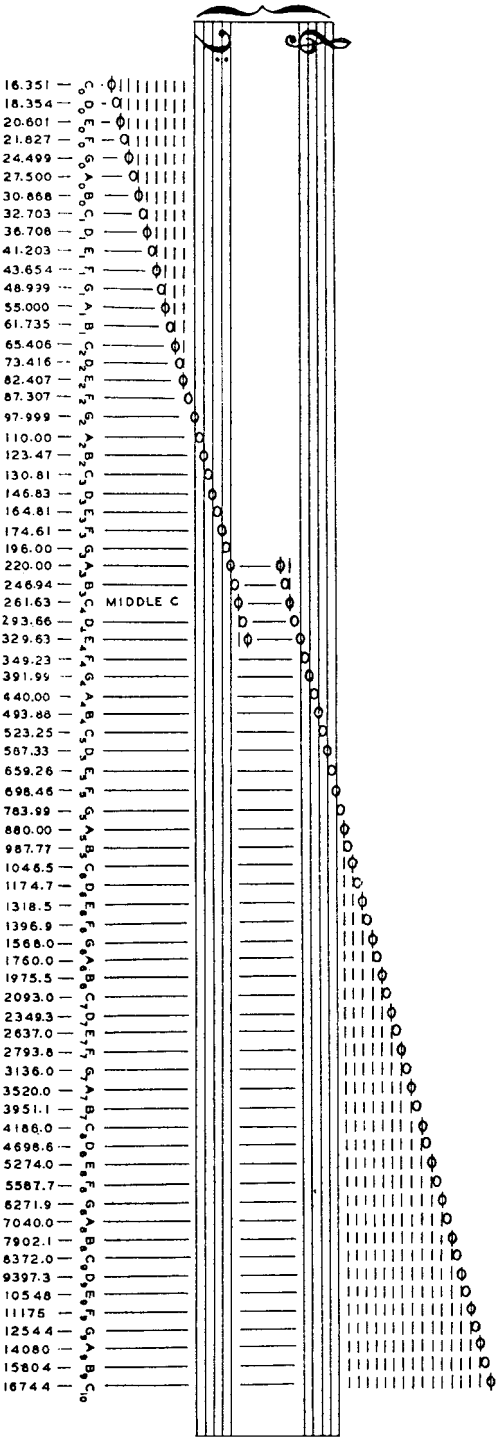


FIGURE 18.7. The frequencies of the notes in the C scale of equal temperament from 16 Hz to 16 kHz.

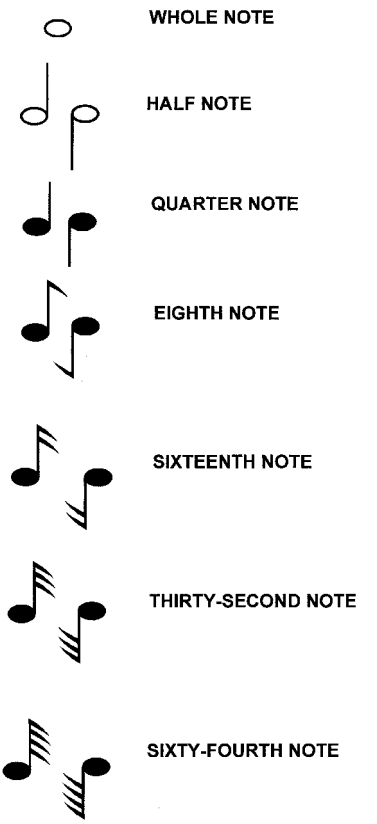


FIGURE 18.8. Note values which indicate duration.

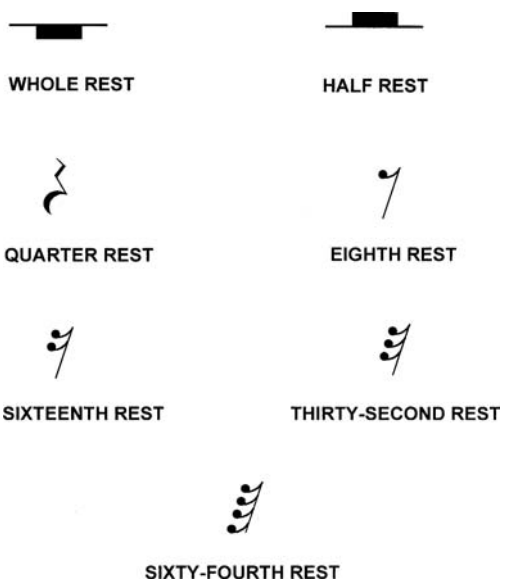


FIGURE 18.9. Rest symbols.

lengthen the duration of the preceding note by half as much, i.e., a whole note becomes equal in duration to a whole note plus a half note, a dotted half note equals the duration of a half note plus a quarter note.

There is no absolute time-interval standard for the duration of a tone represented by a note, and it generally depends on the performer's interpretation of the music with respect to its tempo. Some compositions carry an indication of the setting of a *metronome* for a quarter note. A metronome is a mechanical device that consists of a pendulum activated by a clock-type of mechanism driven by a spring motor. At the extremities of the pendulum swing an audible tick is produced. The interval between ticks can be adjusted by moving a bob on the pendulum arm: the further the bob is located from the fulcrum, the longer the duration between ticks, and vice versa. The pendulum itself is graduated in ticks per minute. The numbers usually indicate the number of ticks per minute, the interval between ticks usually specified as that of a quarter note or, in some cases, half notes. Modern versions of metronomes use electronic means to generate ticks. The metronome setting sets the rate of movement or tempo of the music. Instead of metronome settings, the composer may specify one of a number of terms to designate tempos. Commonly used terms to describe tempos are as follows:

- Largo: Slow tempo
- Andante: Moderately slow tempo
- Moderato: Moderate tempo
- Allegro: Moderately quick tempo
- Vivo: Rapid tempo
- Presto: Very rapid tempo

18.4 Time Signature Notation

A musical selection's time signature is specified at the beginning of the staff by a fraction, as illustrated in Figure 18.10. Common time signatures include 2/4, 3/4, 4/4, and 6/8. The denominator indicates the unit of measure (i.e., the note used to define a pulse). The numerator stands for the number of these units or their equivalents included in a measure (i.e., the interval between two vertical lines across the staff).

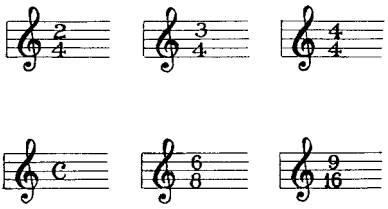


FIGURE 18.10. Time signatures for 2/4, 3/4, C or 4/4, 6/8, and 9/16 times.

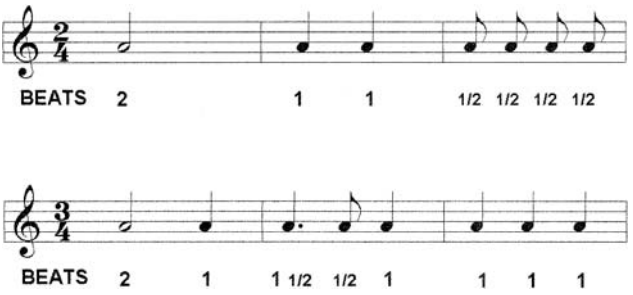


FIGURE 18.11. Notes and beats for 2/4 and 3/4 times.

In upper portion of Figure 18.11 for 2/4 time, each measure contains one half note, or two quarter notes of four eighth notes. Each measure contains two beats, so when a musician plays, the count is one, two. In 2/4 time, a stressed pulse is followed by a relaxed pulse, a sequence used for marches. In the 3/4 time as shown in Figure 18.11, each measure equals three quarter notes or 1 half note plus a quarter note, and so on. Each measure carries three beats, and usually there is one stressed pulse followed by two relaxed pulses, which yields a time used for waltzes. In the 4/4 (or common) time, each measure contains the equivalent of four quarter notes, with the performer counting one, two, three, four. In 4/4 time, a stressed pulse is followed by three relaxed pulses, and this time is used for dances. In 6/8 time, each measure contains six eighth notes or a combination of notes equaling the same duration. There are then six beats to each measure, and in 6/8 time the stressed pulses are one and four of six beats.

Listeners mentally arrange the regular repetition of sounds into groups of stressed and relaxed pulses. These groups are called *meters*. The meter is assigned by the numerator of the time signature, and the most common ones are 2, 3, 4, 6, 9, and 12. Each measure contains a certain number of beats or pulses according to the meter. Meters are classified in terms of the numerators of the time signatures in the following manner:

1. *Duple meter*: Two beats comprise each measure, with the first beat stressed and the second beat relaxed. Example signatures are 2/2 and 2/4 times.
2. *Triple meter*: Three beats occur in each measure, with the first one stressed and the following two relaxed. Example time signatures are 3/8, 3/4, and 4/8.
3. *Quadruple meter*: Four beats occur in each measure, with the first beat stressed and the remainder relaxed. Occasionally the third beat carries a secondary stress. Examples include 4/2, 4/4, and 4/8 time signatures.
4. *Sextuple meter*: Six beats occur each measure, with the first and fourth beats stressed. The 6/8 time signature is such an example.

Rhythm is the repetition of accents in equal intervals of time.

18.5 Key Notation

The *keynote* denotes the note with which any given scale begins. The *tonic* is the keynote of the scale, whether the latter is a major or a minor scale. Many short compositions are written in one key only, but more elaborate musical pieces may shift from one key to another. The key signature of a musical piece is denoted by the number and arrangement of flats and sharps following the clef sign at the beginning of each staff, or it may appear only once at the beginning. Figure 18.12 shows some of the most common key signatures for different major and minor keys.

Major and minor keys play a role in determining the mood of music. In earlier times, a key may have been selected by a composer because a number of wind instruments were able to play only in certain keys. Certain desired effects may be better achieved on more flexible instruments in a specific key than another key. As Machlin pointed out, romantic composers developed affinities for certain keys, for example, Mendelsohn preferred E-minor, Chopin leaned toward C-sharp, and Wagner made use of D-flat major for majestic effects.

Whether it starts with C, D, E, or any other tone, a major scale follows the same arrangement of whole and half steps. Such an arrangement is known as a *mode*. All major scales typify the arrangement of whole and half steps.

The minor mode serves as a foil to the major. The principal difference from the major is that its third degree is flatted. For example, in the scale of C, the third degree is E♭ rather than E. In a natural minor scale, the sixth and seventh steps are also flatted (i.e., C-D-E♭-F-G-A♭-B♭-C). The minor differs considerably from the major in coloring and mood. It should not be inferred that the minor is deemed "inferior"—the nomenclature simply refers to the fact that the interval C-E♭ is smaller (hence minor, the Latin word) than the corresponding interval in the major scale.

If a mode is not specified, the major is implied. For example a *Minuet in G* indicates the G-major. The minor is always specified (e.g., Mozart's Symphony No. 40 in G minor).

To classical composers the tonal qualities of the minor key assumes a more somber aspect (e.g., the funeral music of Beethoven and Mendelssohn) than the triumphal portions of symphonies and chorales which are generally played in major keys. Also, the minor mode carries a certain exotic tinge to Western ears, and thus in the popular view it was associated with oriental and Eastern European music. This was reflected in such works as Mozart's *Turkish Rondo*, a number of Hungarian-style works by Schubert, Liszt, and Brahms, the main theme of Rimsky-Korsakov's *Scheherazade*, and other musical pieces that passed for exotica.

18.6 Loudness Notation

Loudness depends upon the intensity of the musical signal. Although loudness can be measured objectively with the use of a sound-level meter, a conductor or

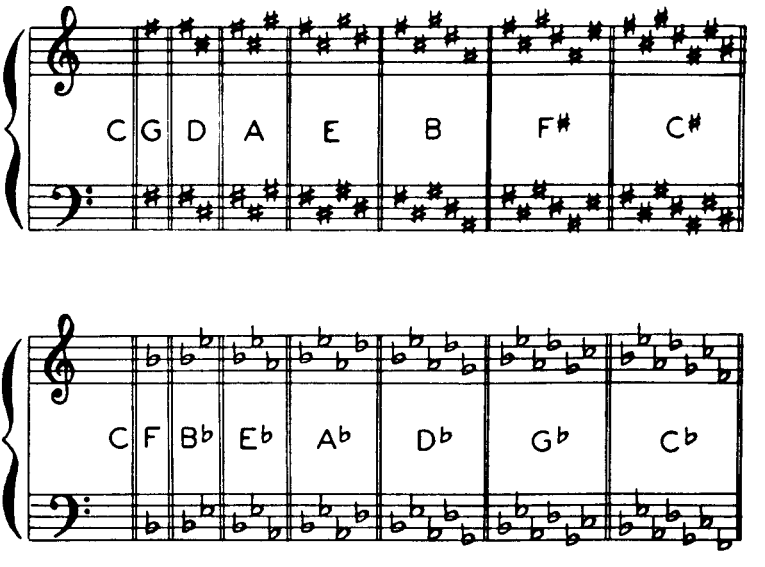


FIGURE 18.12. Key signatures for number of major and minor keys.

a musician depends on his or her own sense of subjectivity to obtain the proper intensity or range of intensities. The common notations and abbreviations for loudness are as follows:

- Pianissimo (*ppp*): softly as possible
- Pianissimo (*pp*): very soft
- Piano (*p*): soft
- Mezzo piano (*mp*): half soft
- Mezzo forte (*mf*): half loud
- Forte (*f*): loud
- Fortissimo (*ff*): very loud
- Fortisissimo (*fff*): extremely loud

Loudness can vary in musical passages. An increase in loudness can be indicated by the term *crescendo* or the abbreviation *cres* or the sign  . A crescendo connotes a gradual increase in the intensity of the music. A decrescendo is a decrease in loudness and is thus the converse of a crescendo. It is denoted by the word *decrescendo* (also *diminuendo*) or the abbreviation *decresc* or the symbol  .

18.7 Harmony and Discord

Consider two tuning forks being sounded together. Let us keep the pitch of one fork fixed at 261 Hz, while the pitch of the other begins at 262 and is gradually raised. As the pitch raises, beats can be heard, due to the difference in the frequencies for a time and then can longer be discerned. The sound of the combined tones starts out by sounding pleasant to the ear and then it becomes gradually more unpleasant. The unpleasantness reaches a maximum at about 23 beats per second, and then begins to abate. This unpleasantness or *discord* declines only slightly, and the discord remains at a fairly uniform level until the octave-marking value of 522 Hz is reached, at which point the unpleasantness disappears.

If this experiment is repeated with violin strings, radically different results will be obtained. The discord does not stay at a uniform level but fluctuates erratically. It almost vanishes at the interval of a major third, and again at the intervals of the fifth and octave. At the precise points at which the minimums of the unpleasantness occur, the frequency ratios of the variable to the fixed tone are found to have the values: $5/4$, $4/3$, $3/2$, and $2/1$.

It has been observed that tone tones sound well together when the ratio of their frequencies can be expressed in terms of small numbers. The smaller the numbers, the better is the consonance. Table 18.1 lists the intervals in order of increasing dissonance.

The further away from small numbers, the more we encroach into the realm of discord. Pythagoras knew this fact more than 2500 years ago when he associated consonance with the ratios of small numbers. The premise of the Pythagorean doctrine "all nature consists of harmony arising out of number," may be somewhat

TABLE 18.1. Interval Nomenclature and Frequency Ratios.

Interval (second)	Frequency Ratio	Largest Integer Occurring in the Ratio
Unison	1:1	1
Octave	2:1	2
Fifth	3:2	3
Fourth	4:3	4
Major third	5:4	5
Major sixth	5:3	5
Minor third	6:5	6
Minor sixth	8:5	8
Second	9:8	9

simplistic, but the Chinese philosophers in Confucius's time also regarded small numbers 1, 2, 3, 4 as the source of all perfection.

The Swiss mathematician Leonhard Euler adopted the psychological approach in declaring that the human mind takes pleasure in law and order, particular in natural phenomena. His theory of harmony is this: the smaller the numbers required to express the ratio of two frequencies, the easier it is to find this law and order, thus making it more pleasant to hear the combined sounds. Euler went so far as to propose a definitive measure of the dissonance of a chord. His idea was to express the frequency ratio of a specific chord by the smallest number possible and then to find the common denominator for these frequencies. For example, the frequency ratio of the common chord CEG c′ is 4:5:6:8. The least common denominator is 120, since it is the smallest number of which 4, 5, 6, and 8 are all factors. But this theory falls apart when the same denominator is assigned to the chord of the seventh CEFGB (frequency ratios 8:10:12:15) that turned out to be far more unpleasant to listen to.

18.8 Musical Instruments

Musical instruments fall into four categories: string, wind, percussion, and electrical instruments. A string instrument may have its strings struck, bowed, or plucked. Wind instruments can be sub-classified as single-mechanical reed, double-mechanical reed, lip reed, air reed, and vocal-cord reed. Percussion instruments are classified as being either definite pitch or indefinite pitch. The advent of electronics has given rise to a whole new class of instruments, such as synthesizers which can effectively replicate the sounds of conventional strings, winds, and percussion instruments as well as generate unusual sounds not heard from any other instruments. Even personal computers can function as musical instruments provided they are equipped with special soundboards and speakers and they are programmed to simulate various types of instruments. Table 18.2 lists a number of musical instruments and their respective classifications.

TABLE 18.2. Classification of Musical Instruments.

String Instruments

Plucked Strings	Bowed Strings	Struck Strings
Lyre	Violin	Piano
Lute	Viola	Dulcimer
Harp	Violoncello	
Zither	Double bass	
Guitar		
Ukulele		
Mandolin		
Banjo		
Sitar		
Harpsichord		

Wind Instruments

Air Reed	Single-Mechanical Reed	Double-Mechanical Reed	Lip Reed
Whistle	Free-reed organ	Oboe	Bugle
Flue Organ Pipe	Reed organ pipe	English horn	Trumpet
Recorder	Accordion	Oboe d'amore	Cornet
Flageolet	Harmonica	Bassoon	French horn
Ocarina	Clarinet	Contra bassoon	Trombone
Flute	Bass clarinet	Sarrusophone	Bass trombone
Piccolo	Saxophone (soprano,		Tuba
Fife	alto, tenor, and bass)		
	Bagpipe		

Organ (combination air reed and mechanical reed)

Percussion Instruments

Definite Pitch		Indefinite Pitch	
Tuning fork	Celesta	Snare or side drum	Triangle
Xylophone	Kettledrums (tympani)	Military drum	Steel drum
Marimba	Bell	Bass drum	Cymbals
Chimes	Carillon	Gong	Tambourine
Glockenspiel		Castenets	

Electrical or Electronic Instruments

Siren	Electrical piano	Electrical carillon	Computer (specially configured)
Automobile horn	Electric guitar	Synthesizer	
Electric organ	Music box	Metronome	

18.9 Strings

Possibly the oldest method of creating music is vibrating string under tension, which is capable of producing a full range overtones of that are harmonics of the fundamental. As was explained in Chapter 4, the presence and the amplitudes of these harmonics depend upon the manner the string is excited (namely, by plucking, striking, or bowing) and where the excitation is applied. Because the string projects a small area, it is not an efficient producer of sound as it is not by itself capable

of moving very much air. It is for this reason that strings are coupled to a large multiresonant surface (or *soundboard*) to increase the sound output.

All string-musical instruments make use of a soundboard or a combination of a ported hollow body and a soundboard to couple the string to the air. The larger the radiating surface, the greater the acoustic impedance, and this provides the increased coupling to enhance the sound output of the strings. Soundboards exhibit the complex modes of vibrations that have been described in Chapter 6 that deals with membranes and plates. In a number of string instruments such as the lute, lyre, zither, guitar, mandolin, and the violin family, a hollow body with portholes is coupled to the strings. The vibration imparted by the strings to the body exterior produces radiation in a fashion similar to that of the soundboard. The hollow body with a hole(s) coupled to the outside air comprises a Helmholtz resonator, and the fundamental frequency can be established from the theory of Section 7.11. The dimensions of the resonator may equal or be larger than the wavelength of the sound in air for some of the tones or overtones produced in the instrument, so the hollow body with its holes can manifest other resonant frequencies in addition to the fundamental resonant frequency. A hollow body is a quite complex resonant system.

Some examples of a plucked-string instrument include the lyre, lute, zither, and harp, three of which are illustrated in Figure 18.13.

The lyre traces its origin to ancient Greece, and it consists of a frame, fingerboard, and a hollow body with sound holes. The lyre is played by plucking the strings with fingers, and the length of the string (and hence the resonant frequency) is varied by pressing the finger against the fingerboard. The hollow body serves as a soundboard to increase the sound output of the string.

The lute, developed more than a thousand years ago, is the precursor of the instruments of the guitar class. It comprises a pear-shaped hollow body, a neck with frets (projecting ridges across the neck), and a head equipped with pegs to tune the strings.

The zither, similar to the lyre in appearance but having its soundboard underneath the entire length of strings, consists of two sets of strings stretched across a flat hollow body featuring a large round holes. One set of steel strings, which pass over a fretted fingerboard, is used to play the melody while a set of gut strings is used for accompaniment. The modern zither now consists of 32 strings, of which four are assigned to the fretted fingerboard. Each of the strings can be tuned by turning pegs of pins at one end of the instrument. The plucking of the zither occurs in the following way: A ring-type plectrum is used on the thumb of the right hand to play the melody. The left hand is used to stop the melody strings by pressing the strings against the frets of the fingerboard. The spacing of the frets are such that the sounds produced by stopping the strings on any two adjacent frets are one semitone apart. The first, second, and third fingers of the right hand play the accompaniment. Zithers are available in three sizes, called bass, bow, and concert types. The open melody strings of the concert zither are tuned to C_3, G_3, D_4, and A_4. The accompaniment strings provide the fundamental-frequency range of C_2 to $A\flat_4$.

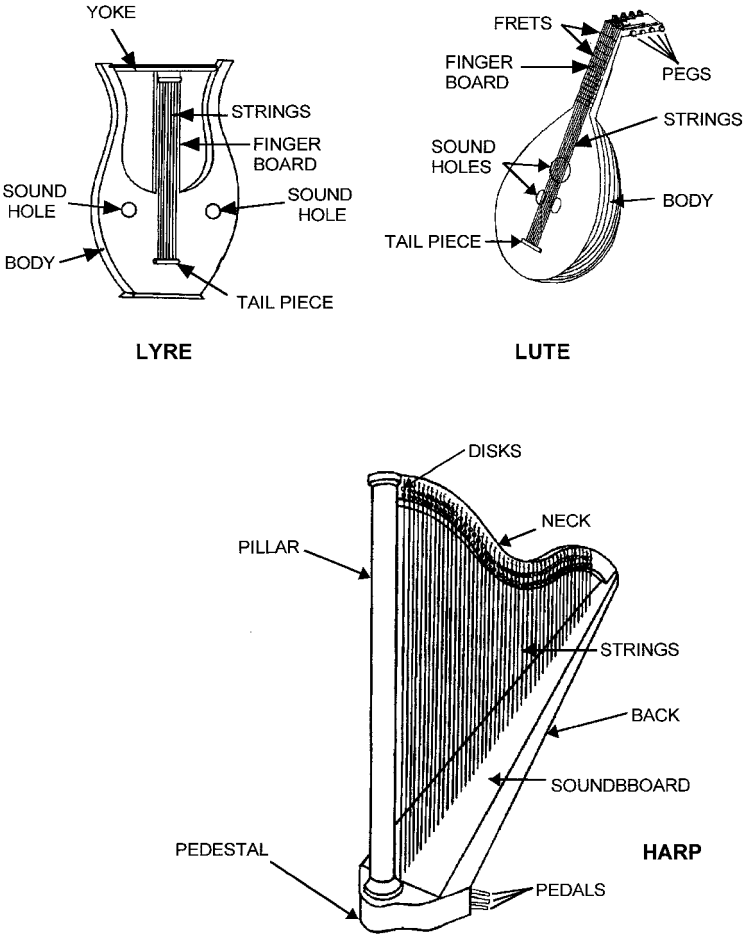


FIGURE 18.13. Plucked-string instruments: lyre, lute, and harp.

The harp, often portrayed as being the instrument of angels, consists of strings stretched vertically upon a triangular frame and connected to a soundboard constituting the lower leg of the triangular frame. The soundboard is quite small, and so the strings are not highly damped and the sound from each string persists for a rather long time, yielding a rather mellow tone. The modern harp is usually provided with seven pedals that actuate a transposing mechanism for shortening the strings in two stages.

The mechanism for shortening the strings is illustrated in Figure 18.14. In panel A of the figure the string is positioned at its maximum length. Depressing the pedal halfway causes disk 1 to rotate along with the pins attached thereto, so the string is shortened by a semitone as shown in panel B. When the pedal is depressed all the way, disk 2 is rotated along with its attached pin, thus shortening the string

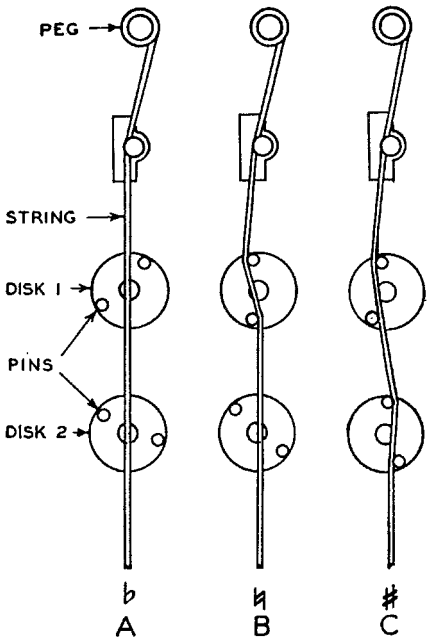


FIGURE 18.14. Mechanism for shortening strings in the harp for transposing tones.

even further so that the string is a whole tone higher, as shown in panel C. Each pedal operates on strings with notes having the same letter notation, i.e., the C pedal controls all the C strings; the D pedal all the D strings, and so on. The harp is generally tuned in the key of C flat. The pillar of the harp serves to bear the stresses produced by the stretched strings and to serve as a housing for the rods connected to the pedals and for the transposing mechanism of Figure 18.14. The overall height of the standard harp is approximately 173 cm. The harp is played either by plucking the strings with the fingers or by sliding the fingers over the strings in a manner referred to as a *glissando*. There are 44 strings, usually made of gut (some bass strings may be constructed of silk overlaid with metal wire for greater density); portable models have as few as 30 strings.

Figure 18.15 shows views of a ukulele, guitar, mandolin, and banjo. The ukulele is a small version of the guitar. The former consists of four strings, tuned to D_4, $F_4^\sharp$, A_4, and B_4, stretched between a combined bridge and tailpiece attached to the top flat surface of the body and the end of a fretted fingerboard. The body itself consists of two flat surfaces fastened together by a contoured panel at their outside edges. The bottom of the body is mechanically attached to the top by a post, and the cavity of the body, coupled to the external atmosphere, acts as a resonator. The body of the ukulele may be made of wood or steel or plastic. The 61-cm long instrument is played by strumming the strings with the fingers, and the resonant frequency of each string is varied by the fingers pressing against the frets that are spaced so the sounds produced by stopping a string on any two adjacent frets are

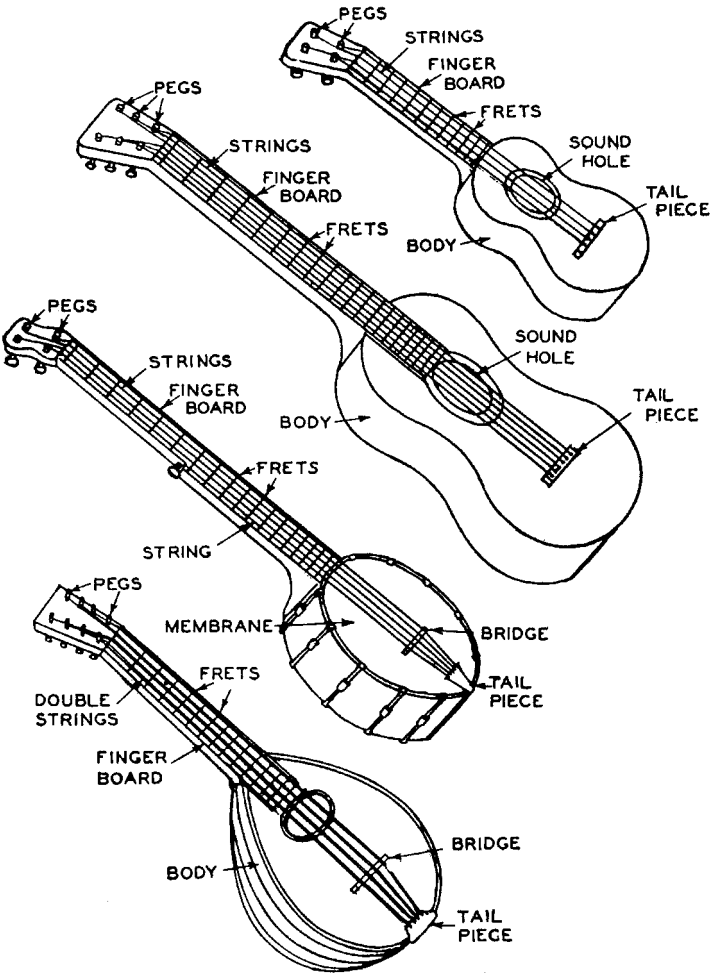


FIGURE 18.15. Additional plucked-string instruments from top to bottom: ukulele, guitar, banjo, and mandolin (from Olson, 1967).

one semitone apart. The guitar is similarly constructed except it is larger than the ukulele by nearly twice the length, and it consists of six strings, tuned to E_2, A_2, D_3, G_3, B_3, and E_4. The guitar is played by either plucking with fingers or with a pick or plectrum (a flat piece of metal or plastic) held firmly between the thumb and the first finger. The adjacent frets also result in notes that are one semitone apart.

The body of a mandolin consists of a flat top attached to a hollow semiellipsoidal body. This combination of the body and the body cavity coupled to the external air through a hole constitutes a complex resonator. A bridge mounted at the center of the flat surface of the body couples the vibrating strings to the body.

The difference between the construction of a banjo and other string instruments is that its body consists of a skin membrane stretched over one end of a truncated cylinder, thus making the body drumlike. The other end of the cylinder is open. The bridge is supported by the stretched skin and it couples the strings to the stretched membrane, which, in turn, provides a large resonant area that is coupled to the air. Four long strings and a short string are stretched over the bridge between the tailpiece and the fingerboard. The relatively long neck is fretted; the short string is referred to as the *melody string*. A more modern version called the *tenor banjo* is equipped with four strings of equal length, and it has supplanted the older model with the one short string. The four open strings are tuned to C_3, G_3, D_4, and A_4. This instrument can be played by either plucking the strings with the fingers or with a pick or plectrum that can be made of a flat piece of tortoise shell. The note of a string and hence the resonant frequency can be varied by pressing it against the frets. The spacing between a pair of adjacent frets corresponds to a difference of one semitone. The overall length of a banjo is approximately 86 cm.

Over the past three decades, Europeans and Americans have become familiar with structured musical compositions called *ragas* through concert performances by Ravi Shankar who performed them on the *sitar* which is northern India's predominant string instrument. The sitar's seven main strings are tuned in fourths, fifths, and octaves to approximately $F_3^{\#}$, $C_2^{\#}$, $G_2^{\#}$, $G_3^{\#}$, $C_3^{\#}$, $C_4^{\#}$, and $C_5^{\#}$. In addition there are 11 sympathetic strings tuned to the notes of the raga. The inharmonicities are quite small; the high-curved frets permit the player to execute with vibrato and glissando. The curved bridge allows for both amplitude and frequency modulation from the rolling and sliding of the string.

The harpsichord (or cembalo) and its older cousin, the clavichord, both of which resemble shrunken baby grand pianos (the clavichord is more boxlike in configuration), trace their common origin as far back as the twelfth century. The harpsichord was the mainstay of chamber music during the baroque and classical period until the advent of the more versatile and louder pianoforte, the immediate precursor of the modern piano. Because so many excellent examples exist to this day, the art of constructing harpsichords have been revived so that modern audiences can today enjoy the music that have been composed expressly for this medium. During the twentieth century, a number of excellent performers revived audience interest in the instrument and the works written for it, among them the great Wanda Landowska and later on Igor Kipnis (the son of the great Ukrainian operatic basso, Alexander Kipnis).

Figure 18.16(a) illustrates the structure of a harpsichord, which consists of a large number of steel strings stretched over a rather triangular steel frame. The keyboard ranges about 4½ octaves from A_1 to F_6, but different versions of harpsichords have been built to cover both larger and smaller ranges. The strings are excited by being plucked by a key-actuated mechanism shown in Figure 18.16(b). The key is coupled through a level system to a short plectrum of leather, fiber, or tortoise shell that plucks the string which deflects to let the plectrum slip past. When the key is released, the plectrum, which is attached to a short spring-loaded level, slips back under the string. A damping pad also mounted on the jack stops the sound.

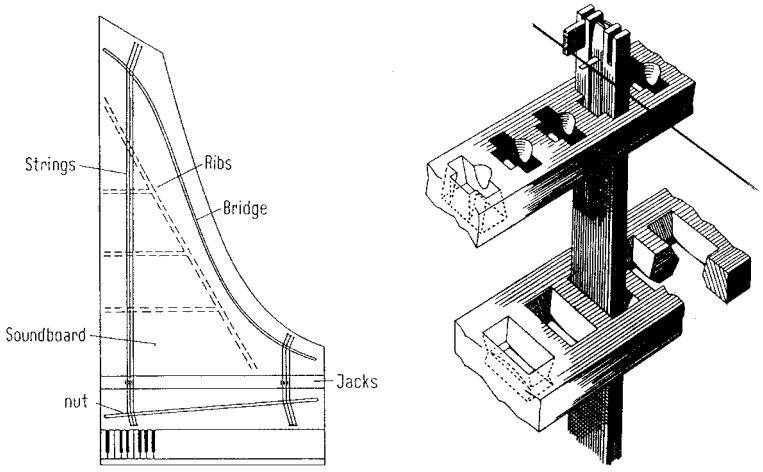


FIGURE 18.16. The harpsichord: (a) its construction and (b) the string-plucking mechanism of a harpsichord.

Bowed-string instruments are played by exciting the strings with a bow. Four modern instruments of this type, in order of increasing size, are: the violin, viola, violoncello, and double bass (cf. Table 18.3). Figure 18.17 is illustrative of the comparative size of these instruments. Details and nomenclature of the violin's components (the nomenclature also fairly applies to the other three instruments) are shown in Figure 18.18. The two smaller instruments, the violin and the viola, are generally played with the chin rest tucked under the chin, with the fingerboard cradled in the arch formed between the thumb and the index finger of the left hand; and the fingers of that hand vary the notes of the strings by pressing them against the fingerboard surface. The bow is handled by the right hand. The larger violoncello and the double bass, being much larger instruments, are played held in a tilted, almost vertical position, between the knees of the seated players, with endpins elevating the instruments from the floor.

TABLE 18.3. Typical Characteristics of Standard Bowed-String Instruments.

Bowed-String Instrument	Notes of Tuned Open Strings	Typical Overall Length of Full-Fize Instrument, cm	Typical Overall Length of Bow, cm	Range, Number of Octaves
Violin[a]	G_3, D_4, A_4, E_5	60	75	>4
Viola	C_3, G_3, D_4, A_4	70	75	>4
Violoncello	C_2, G_2, D_3, A_3	124	72	3
Contrabass	E_1, A_1, D_2, G_2	198	66	3

[a] Smaller violins, as small as quarter size, have been constructed to accommodate small children learning to play. Even smaller ones have been constructed for 2-year-olds learning under the Suzuki method.



FIGURE 18.17. Modern bowed-string instruments: (a) violin, (b) viola, (c) violoncello, and (d) double bass.

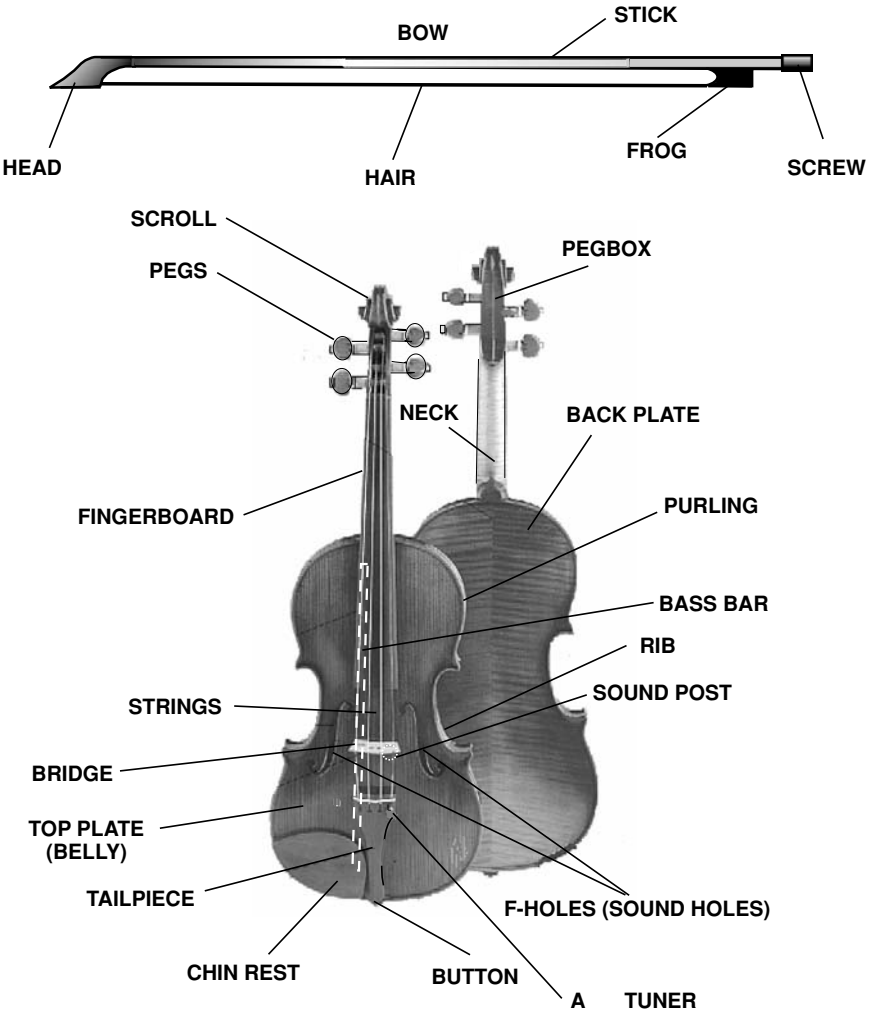


FIGURE 18.18. Construction details of a violin and nomenclature of its structural elements.

These instruments developed in Italy during the sixteenth and seventeenth centuries require extremely sophisticated skills to construct. Their quality of craftsmanship reached a peak during the eighteenth century in Cremona, Italy, particularly under the skilled hands of Antonio Stradivari (1644–1737) and Guiseppe Guarneri del Gesù (1698–1744). Because of the complexity of their construction that affects the quality of their tones, the violin family has been the object of much acoustical research. Savart, Helmholtz, and the Nobel Laureate C. V. Raman (1888–1970) contributed to the understanding of the generation of sound with these instruments (Savart, 1840; Raman, 1918; Helmholtz, 1954). In more recent times, considerable work has been conducted in Germany through the efforts of Werner

Lottomoser, Jürgen Meyer, and Frieder Eggers. Especially noteworthy is the work of Lothar Cremer (1905–1990) and his colleagues, which culminated in Cremer's classic text *The Physics of Violins*. In the United States, Frederick Saunders (1875–1963), best known for his work in spectroscopy, investigated many violins, making many acoustical comparisons between the old and the new. He, Carleen Hutchins (1911–), John C. Schelleng (1892–1979), and Robert Fryxell (1924–1986) established the Catgut Acoustical Society, an organization that promotes research on the acoustics of the violin family. In 2004, the society combined with the nonprofit Violin Association of America, becoming a sector known as the CAS Forum.

Violin makers (or *luthiers*) are especially concerned with the vibration of the free top and back plates. They tested these plates by tapping and listening to tones. Modern technology is now being used to visually observe mode shapes (Figure 18.19) through the use of holographic interferometry. The finished violin's vibrational modes are considerably different from those of the free plates forming the top and bottom of their bodies.

The bow for any of these four instruments consists of horsetail hair stretched between the two ends of a thin wood, one end constituting the head and the other point of attachment being a movable frog that is connected at the other end of the bow to a screw inside the bow wood (cf. Figure 18.18). The screw can be turned to move the frog, thus adjusting the tension of the stretched horsehair. The horsehair is rubbed with rosin to provide friction between the bow and the strings. When the

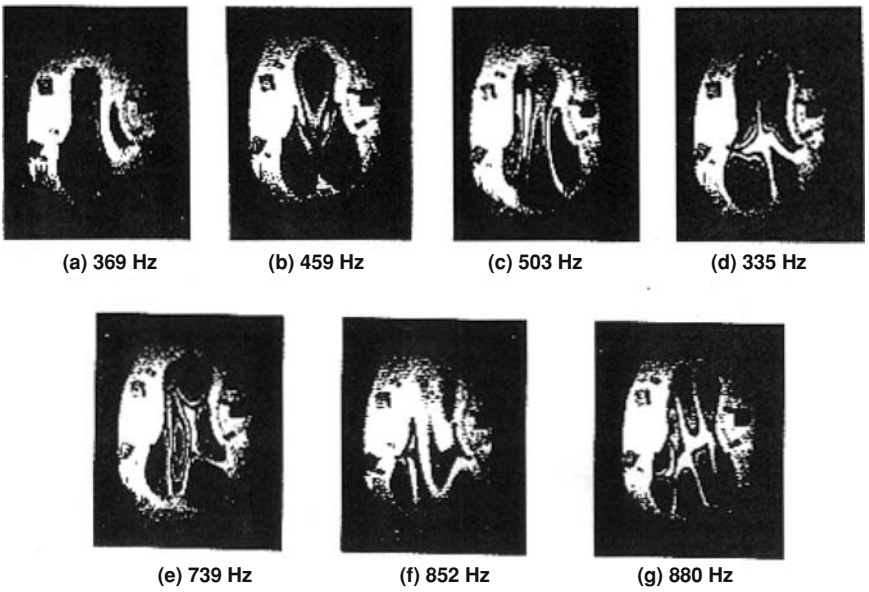


FIGURE 18.19. Holographic interferograms of a top plate of an instrument of the violin family (Ågren and Stetson, 1972).

bow is drawn across the strings, the string vibrates as the result of its being dragged with the bow and then springing back under the impetus of the restoring force. The normal range of the bow's action on the strings depends on its force acting on the strings and the position of the bow relative to the bridge. Too heavy a force results in raucousness of the tone. Too little force results in instability of the string displacement's saw-toothed curve. According to Schelleng (1974), the maximum bow force depends primarily on the string and on its coefficient of friction. This bow force is inversely proportional to the distance of the bow from the bridge; the minimum bow force, on the other hand, is inversely proportional to the square of the distance of the bow from the bridge. The maximum and minimum bow forces are equal when the bow is placed at a point very close to the bridge, and they diverge as the bow moves away from the bridge. The disparity between these two limits, as the bow is positioned further away from the bridge, provides the wide tolerance that makes fiddle playing possible. As a far distance from the bridge, for a given bow velocity, the volume of the sound is less, the amount of high-frequency content lessens, and the timbre possesses a gentle character designated by composers as *sul tasto* ("bow over the fingerboard"). In placing the bow near the bridge, the required bow force soars dramatically to almost prohibitive levels and the solidity of the fundamental tone disappears, leaving little more than the remnants of high harmonics to suggest the fundamental tone: this is the eerier sounding *sul ponticello* ("bow over the little bridge"). In the normal playing area, the tone becomes more brilliant (i.e., the relative harmonic content increases) as the bow moves toward the bridge or as the bow force increases toward the maximum.

A vibrating string alone produces almost no sound, because it is so slender that almost no air is displaced by its vibration and, moreover, the two diametrically opposite sides of the string are so close together that when the air on one side is compressed, the air on the other side is rarified. These two effects are so close to each other that they effectively cancel each other. In order to avoid the cancellation, one vibration must follow another by a substantial fraction of a phase.

In the bowed instruments, under optimal conditions, the top and the back plates can move inward and outward at a given moment that nearly the entire surfaces of the instrument act to change the volume of displaced air, thus acting as a "simple source" in the lower frequency range (Cremer, 1971). In the violin family, this volume change, essential to generating sounds in the lower octaves, is rendered possible by the asymmetric layout of the bass bar and the sound post inside the body. When the bow is pulled across the string, a rocking motion is set up in the bridge so that the two feet of the bridge are in a "push–pull" mode, with one foot of the bridge pressing down and the other foot going up in opposition. If the box (i.e., the body of the instrument) has total bilateral symmetry, the motion of one foot of the bridge would cancel that of the other foot of the bridge, and no volume displacement would occur. But the sound post, essentially a dowel firmly coupling the back plate to the top plate tends to immobilize the right (with respect to the player holding the instrument) foot while allowing the left foot to more freely and causing the body of the instrument to vibrate under the influence of the left foot. Thus, the sound post's chief role can be considered as creating

acoustical asymmetry inside the body. The shape, positioning, wood quality, and fitting of the sound post constitute highly critical factors in performance quality of the instrument.

The bass bar runs along the length of the interior surface of the top plate approximately under the lowest-tuned string. Its function is to keep the vibrations of the upper and lower areas of the top plate in phase with the left foot of the bridge. The bar is glued to the top plate in such a manner so as to provide structural strength to the thin wood forming the top plate. Both the sound post and the bass bar enable the top plate to withstand downward forces of 70–90 N from the strings. The contour of the bass bar is extremely important in the proper "thinning" or "tuning" of the top plate.

The f-holes serves two acoustical purposes: (a) reduction of the stiffness of the surface on which the bridge stands in order to provide suitably tapered transition between the bridge and the radiating area of the top plate and (b) formation of a Helmholtz resonator with the body of the instrument, thereby strengthening sound in the lowest octaves. The purling are inlays of thin wood placed in grooves along the edges of the plates. This allows the plates to vibrate more as if they were hinged rather than being clamped at the edges.

In the violin, the top plate is generally constructed of softer Norway spruce (*Picco abies* or *Picca excelsis*), and back plate and the ribs are carved from (harder) curly maple (*Acer platanoides*). Fingerboard, tailpiece, and pegs are usually made of ebony. The bow stick is made from pernambuco.

The strings of the violin were originally all gut. Since about the seventieth century, the lowest (G) string was commonly wound with silver to improve the response. Present-day violinists use wound strings (i.e., metal wound over cores of gut, nylon, or metal) for D and A strings and steel E-string. The latter string can be finely adjusted with the use of a fine tuner.

In addition to the four principal members of the violin family, the aforementioned Catgut Acoustical Society was responsible for the development of the violin octet, an ensemble of eight specially scaled new violin family instruments (Hutchins, 1967). The octet of instruments, illustrated in the photograph of Figure 18.20 and developed by the society to meet the challenge of encompassing the entire range if orchestral music, consists of the treble violin, the soprano violin, the mezzo violin, the alto violin, the tenor violin, the baritone violin, the small bass, and the contrabass. Although these instruments form a family with basic traits in common, each member has its own individual personality. The homogeneity of sound arises from adjustments in body length and other physical characteristics so that each instrument has its own main wood resonance and main air resonance near the two open middle strings. The musical ranges of the strings on these instruments are shown in Figure 18.21, as they relate to the corresponding notes on a standard piano. These eight instruments of the new violin family range in overall length from 48 cm to 214 cm.

The principal modern struck-string instrument is the piano that are available in several models, ranging in size from the more modest upright or spinet piano to the concert grand piano. The heart of the piano consists of a large number of



FIGURE 18.20. The violin octet developed by the Catgut Acoustical Society. (Courtesy of the New Violin Family Association.)

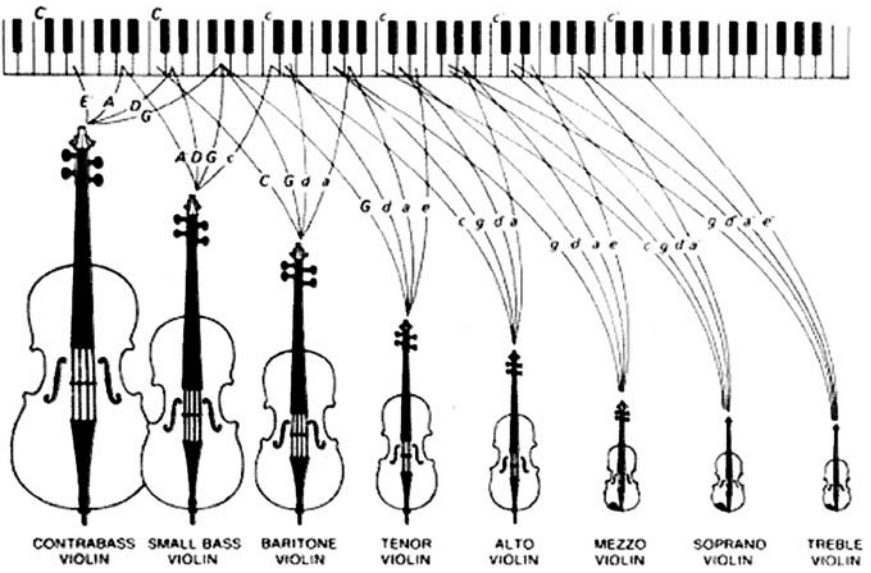


FIGURE 18.21. The musical ranges of the of the new violin octet developed by the Catgut Acoustical Society as related to piano notes. (Courtesy of the New Violin Family Association.)

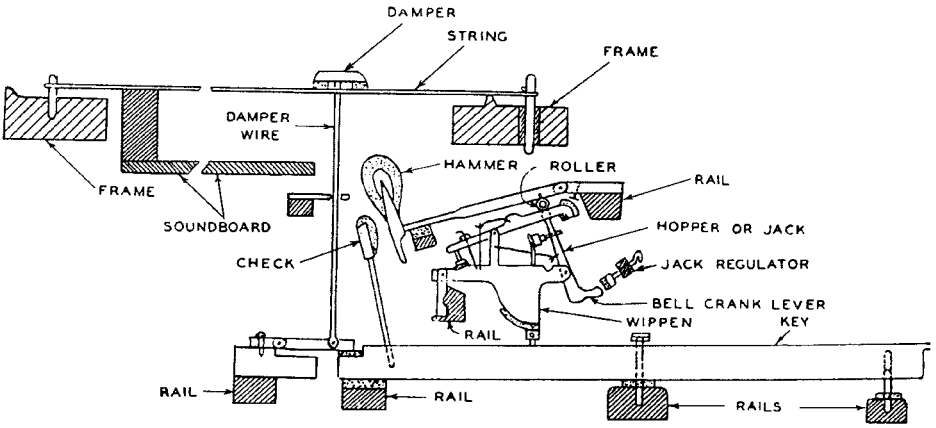


FIGURE 18.22. Schematic of the mechanism of a grand piano (from Olson, 1967).

steel strings stretched on a metal frame. The strings couple through a bridge to a large soundboard. The strings are activated by being struck by hammers that are connected to keys forming a keyboard. Pressing down on a key actuates the hammer, which, in turn, strikes the string. The conventional piano is equipped with 88 keys, and the piano covers a wide frequency range of more than seven octaves, from A_0 to C_8 (27.5–4186 Hz). The Bosendorfer 244-cm concert grand piano features 97 keys, but not much music has been written for these extra keys. The extra keys are mainly there because of the additional resonances produced by the extra strings and the larger soundboard.

Figure 18.22 illustrates a schematic of the piano mechanism for a grand piano. The strings stretch from the pin block across the bridge to the hitch-pin rail at the other end. When a key is pressed downward, the damper rises and the hammer impacts the string, causing it to vibrate. The string's vibrations are transmitted to the soundboard through the bridge. The hammer rebounds, remaining about 1.25 cm from the string as long as key remains pressed. When the key is depressed, the damping pad does not engage the string. When the key is released, the damping pad engages the string to speed up the decay of the sounding note.

The largest version of the piano, the concert grand, has 243 strings that vary in length from about 200 cm at the bass end to approximately 5 cm at the treble end. In this group there are 8 single strings wrapped with one or two layers of wires, 5 pairs of strings also wrapped, 7 sets of 3 wrapped strings, and 68 sets of 3 unwrapped steel strings. The smaller pianos may contain fewer strings but they still play the same number of notes. A small grand piano may carry 226 strings. The arrangement of the strings is such that the bass strings may overlay the middle strings, so that they can function nearer the soundboard. The soundboard itself is usually made of spruce and it is up to 1 cm thick; it acts as the principal source of radiated sound just as the top plate of a violin does. Because the tension forces in the strings are so high, the frames are fabricated of cast

iron, which also provides dimensional stability necessary to maintain the state of tune.

Three pedals are provided on the conventional piano. The right or *sustaining* pedal removes all dampers from the strings so that the strings become damped only by the soundboards and end supports. The center or bass sustaining pedal removes the dampers from all the bass strings. The left pedal, or the soft pedal, reduces the sound output by lessening the length of the stroke of the hammers or by shifting the hammers so that fewer strings are struck or by permitting the dampers to act.

The dulcimer, considered by some to be the forerunner of the piano, consists of a large number of strings stretched over a frame mount in an oblong box. These strings pass over bridges that are coupled to a soundboard. The oblong box is mounted on legs, causing the instrument to resemble a square piano without keys. The instrument is played by striking the strings with two hammers, one in each hand. Dampers controlled by a foot pedal are provided. The range of the dulcimer is from D_2 to E_6.

18.10 Wind Instruments

A wind musical instrument is a device that generates sound by (a) blowing a jet stream of air across some type of opening, as in whistles, flutes, or piccolos, fife and flue organ pipe; or (b) by buzzing of lips (acting as reeds) of a bugle, French horn, trumpet, tuba, or trombone; or (c) by vibrating a reed (or a double reed, i.e., a set of two reeds) through the means of airflow in an accordion, clarinet, saxophone, oboe, bagpipe, English horn, bassoon, sarrusophone, and the human voice.

The reed instruments fall into two sub-categories: one category entails instruments in which air pressure tends to force the reed valve open (the human larynx, buzzing lips of brass instruments, and harmonium reeds); the other category includes instruments in which the air pressure forces the reed valve to close (e.g., clarinets, oboes, similar woodwinds, and organ reed pipes). The first category tends to act as a sound generator over a relatively narrow fundamental frequency range, just above the fundamental frequency of the reed. The other category serves as a sound generator over a wider range of frequencies, just below the resonant frequency of the reed—but some type of coupling to a pipe resonator must be provided.

Figure 18.23 illustrates two possible configurations of a vibrating reed generator. The reed generator is really is a pressure-controlled device for cutting off and reinstating airflow at selected frequencies. In both cases of Figure 18.23, the blow air pressure p_b (gauge pressure, relative to the atmospheric pressure) is applied from the left. If $p_b > 0$ (i.e., the blow pressure is above atmospheric pressure), the valve in (a) is forced closed by the positive pressure and the valve in (b) is forced open. The opposite situation occurs when $p_b < 0$. In general terms, the motion of a reed can be described by (Fletcher and Rossing, 2004).

$$m_r = \left[\frac{d^2x}{dt^2} + \frac{1}{Q_r}\omega_r\frac{dx}{dt} + \omega_r^2(x - x_0)\right] = \gamma_{gr}(p_b - p) + \gamma_{be}\frac{U^2}{|x|^2}$$

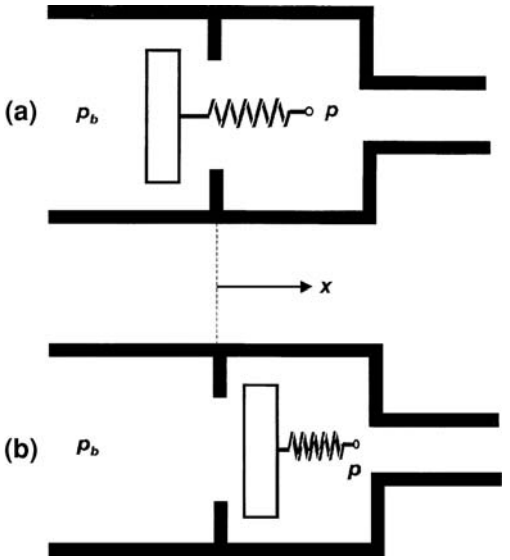


FIGURE 18.23. Two types of vibrating reed generator.

where

m_r = mass of reed
Q_r = quality factor of reed resonance
x = position of valve relative to its seat
x_0 = equilibrium position of valve
ω_r = angular frequency of reed
p_b = blow pressure
p = acoustic pressure
γ_{gr} = geometric factor for the exposed reed faces
γ_{be} = Bernoulli force factor based on internal flow in the narrow part of valve gap
U = flow velocity.

In the *air-reed instruments*, a steady stream of air does the activation by flipping in and out of the pipe or cavity at the resonant frequency of the system, as shown in Figure 18.24. This steady stream of air thus becomes an alternating flow, and because of the nonlinear nature of the exciting force, a number of the resonant elements in the system are excited, thus yielding a series of overtones that add to the fundamental tone. The whistle and flue organ pipe, each consists of a cavity, a closed or open pipe coupled to an air reed that is activated by a steady air stream. The air stream entering the pipe vacillates between moving to the inside of the pipe and moving to outside of the pipe. Upon entering the pipe, the air stream compresses the air in front of it, as depicted in Figure 18.24 at $\theta = 0^\circ$. The pressure inside the pipe builds up to the equilibrium point, and no more air will

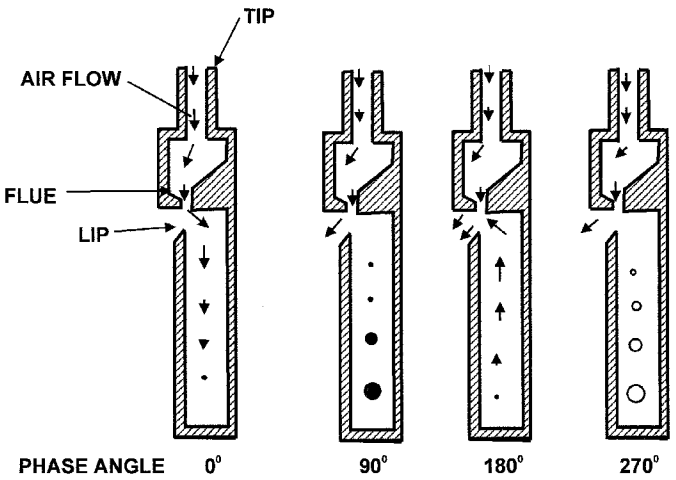


FIGURE 18.24. The mechanism of the flue pipe organ, an air-reed instrument. The magnitude of the pressures are indicated by the diameter of the circles. A dark circle denotes a pressure above atmospheric, and a white circle indicates a pressure below atmospheric. The direction of the arrows represents the direction of airflow, and the magnitude of particle velocities are indicated by the length of arrows.

then enter the pipe, as indicated in Figure 18.24 at $\theta = 90^\circ$. The excess pressure will now direct the incoming air stream to the outside at $\theta = 180^\circ$. This results in the excess pressure being relieved and a rarefaction owing to the inertia of the outgoing air, as shown for $\theta = 270^\circ$.

The decreased pressure pulls in the air to renew the cycle at 0°. This cycle consisting of the four phases repeats itself at the resonant frequency of the system. The frequency of the complete cycle occurs at the resonant frequency of the closed pipe. Odd harmonics are also produced, because the closed pipe resonates at these frequencies as well as the fundamental; this action steers the air stream vacillating from the interior and the exterior. In the case of the open pipe, both odd and even harmonics are produced. Sectional views of the open-flue pipe (made of metal) and a stopped-flue pipe are shown in Figure 18.25. A whistle, such as that used by police officers and referees of athletic contests, operates on the same principal as the organ flue pipe except that the resonating chamber is a Helmholtz resonator, which is essentially a chamber with a narrow neck. The resonance of a whistle can be found from application of Equation (7.66), on the basis of the volume of the whistle chamber and the area of the sound-radiating hole coupling the chamber to the outside air. The overtones due to resonances within the small chamber occur at relatively high frequencies and are effectively suppressed by the inertance of the sound-radiating hole, thus resulting in a nearly pure tone of the whistle. A calliope is a group of whistles with frequencies corresponding to the notes of a musical scale. Each whistle is controlled by a valve connected to a key that is part of a keyboard similar to that of a piano. Either steam or compressed air is used to actuate the whistles.

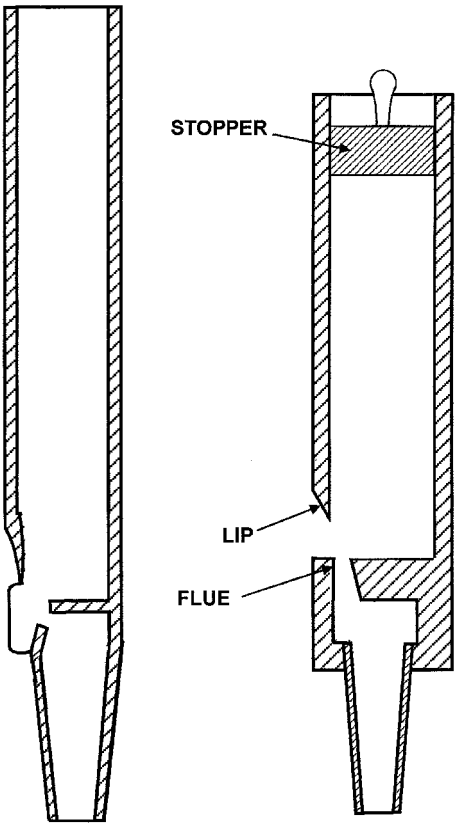


FIGURE 18.25. Sectional views of the open flue pipe (*left*) and a stopped-flue pipe (*right*).

The recorder and the flageolet can be categorized as instruments of the whistle class. Each is equipped with a mouthpiece, a fipple hole, and a cylindrical tube bearing a set of fingerholes. The recorder has eight fingerholes plus one thumbhole on the opposite side of the tube to alter the resonant frequency of the air column. Different types of recorders have been constructed ranging in length from 30 cm to almost 900 cm. The flageolet features a set of four fingerholes plus two thumbholes. The ocarina's construction differs from that of a flageolet in that its resonating system is a cavity and hole combination, not a pipe. The resonator is coupled to the air through several holes, including the fipple hole. The resonance frequency is increased as the number of holes is increased, because the inertance decreases with the number of holes. Ocarinas, which may be made of metal, ceramic, or plastic, cover about a range of an octave and a half.

The flute illustrated in Figure 18.26 is a cylindrical main tube with a slightly tapered head-joint. One end of the flute is open and the other is closed. The embouchure (blowhole) is located a short distance from the closed end. The holes are controlled by closing or opening them in order to vary the resonant frequencies

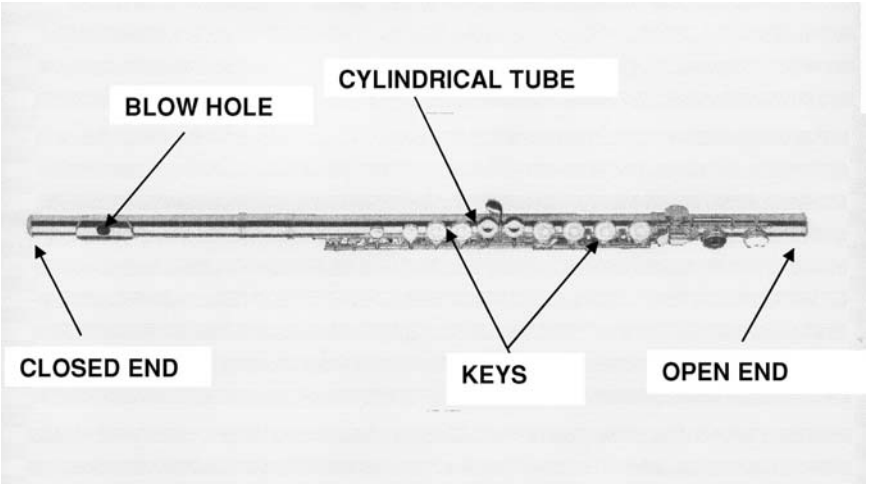


FIGURE 18.26. The modern flute.

corresponding to the musical scale, either by the fingers directly or through actuation of keys. This system of keys and connecting shafts and levels constitute a mechanism that actuates valves that are basically disks that cover fingerholes. Springs are utilized to keep the valves in the unactuated position, which may be either open or closed. This system renders it possible to open or close fingerholes that are too far apart or are too large to be stopped by the fingers alone.

The sound of the flute is generated in this manner: In Figure 18.27 an air stream from the lips impinges upon the embouchure of the flute. Resonant frequencies are generated by the air stream slipping back and forth between entering the flute body and flowing past the embouchure. A stream of air enters the blowhole ($\vartheta = 0^\circ$), causing a pressure to build up to the extent that it stops air from entering ($\vartheta = 90^\circ$). The excess pressure ($\vartheta = 180^\circ$) then forces the air out of the blowhole until it stops leaving the blowhole ($\vartheta = 270^\circ$). The cycle then repeats itself.

The piccolo can be considered a smaller version of the flute that operates one octave higher. The operational principle of the piccolo is fundamentally the same

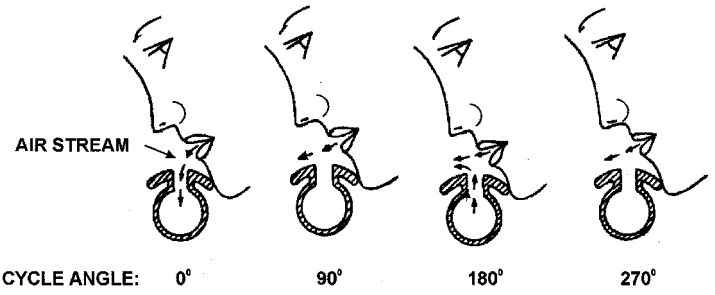


FIGURE 18.27. The action of the flute: the directions of the arrows indicate airflow direction.

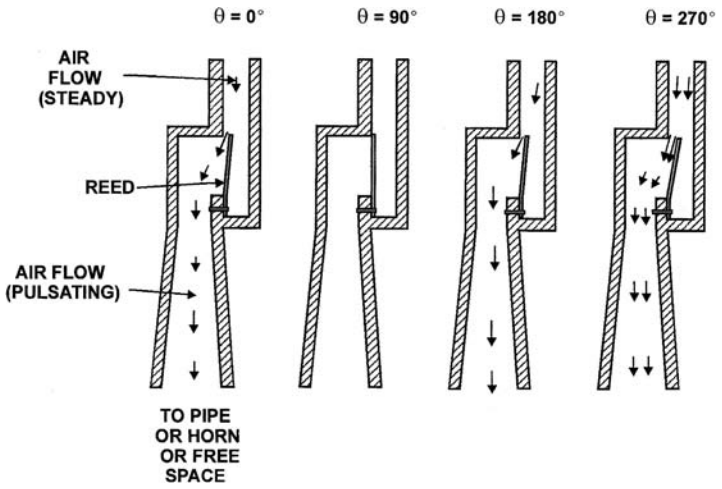


FIGURE 18.28. Reed position and particle velocities (indicated by arrows) in a mechanical reed instrument for a complete cycle (after Olson, 1967).

as that of the flute, but the fundamental range is from D_5 to B_7 instead of the range from C_4 to C_7 of the latter instrument. The fife is a somewhat simpler instrument, consisting of a tube of metal or wood that is closed at one end and open at the other. It is equipped with six fingerholes distributed along the tube and a blowhole near the closed end.

Mechanical reed instruments are those that use a steady air stream to actuate a mechanical reed to throttle the airflow at the resonant frequency of the reed and the associated acoustical system. These instruments may be subdivided into two categories: single- and double-reed instruments.

Single-reed instruments use a single-mechanical reed to throttle a steady air stream to generate a musical sound. A free-reed instrument is one that radiates its sound directly into the air. Modern examples of the free-reed instrument are the free-reed organ (harmonium), accordion, and harmonica. In other instruments (e.g., reed organ pipe, clarinet, saxophone, or bagpipe), the reed couples to a resonant air column. The action of the reed is that of a free-end (cantilever) vibrating bar described in Chapter 5. The mechanical reed in its quiescent state is positioned so that it forms an opening with the structure of the instrument as shown in Figure 18.28. The velocity of the air moving past the reed through the opening results in a lower pressure on the flow side, due to the Bernoulli effect,[1] which causes the reed to flex toward the lower pressure, causing the opening to

[1] Bernoulli's principle states that the dynamic pressure of a nonviscous fluid flow will drop with an increase in the velocity of the flow and vice-versa according to the relation:

$$p + \frac{1}{2}\rho u^2 = \text{constant}$$

where p is the air pressure, ρ the air density, and u the air velocity.

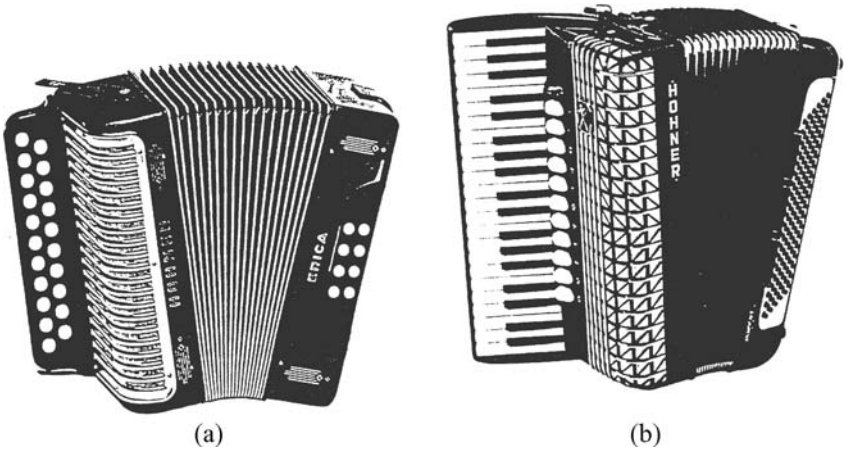


FIGURE 18.29. Two accordion versions: (a) button keyboard accordion and (b) piano keyboard accordion. (Photographs of the Hohner Corona II diatonic accordion and Hohner Atlantic IV piano keyboard accordion courtesy of Hohner, Inc.)

become smaller. The airflow then becomes reduced by the constricted passage, which causes the pressure on the flow side to increase, and the reed springs back to its original position. It then moves beyond its original position, under the effect of inertial energy, and the airflow becomes larger. The pressure again drops and the reed returns to the normal position it had at the beginning, and the cycle of events repeats at the resonant frequency of the system. This action has the effect of converting a steady air stream into a saw-toothed pulsation that contains the fundamental and its harmonics.

The harmonium or the free-reed organ operates through a series of air-actuated free reeds tuned to specific notes. The air supply driving the reeds is provided by two pedal-operated pedals connected to bellows that connects to a wind chest on the top of which reeds are mounted. This type of organ features a piano-like keyboard. Each key operates a valve that controls the air supply to the reed, with one key for each reed. Reed organs are usually equipped with stops for connecting banks of reeds. Thus, a single key can activate a number of reeds.

The accordion shown in Figure 18.29 functions by the player's arms working a bellows which provides the air supply to the reeds, alternatively creating pressure (when the bellows is compressed) and a partial vacuum (when the bellows is expanded). The air-actuated reeds, tuned to notes of a musical scale, are controlled through a keyboard. Each key connects to a valve that controls two separate reeds. One reed operates under pressure and the other under a vacuum. In some versions of the instrument, two differently tuned reeds are assigned to each key that one tone is produced with expansion of the bellows and another tone with the compression. The vibrational action of the reed is that of a free-end bar. Accordions, which are really portable versions of reed organs, are usually equipped with stops to connect individual reeds so as to form banks of reeds. This enables operation of several

reeds from a single key. In playing the accordion, the performer uses the left hand to move the bellows and to play the bass parts and the accompaniment and uses the right hand to play the melody.

The simple accordion shown in Figure 18.29 contains 10 melody buttons or keys and tow bass keys. Each key produces two different tones, one tone upon bellows expansion and the other upon bellows compression. The piano accordion of Figure 18.29(b) is equipped with a two-octave piano-style keyboard (with black and white keys) and 12 bass and chord buttons. The same tone is produced on both compression and expansion cycles. There is no standard size for accordions. More elaborate accordions have been constructed with melody keyboards covering up to four octaves, and some of the larger accordions contain as many as 120 bass buttons arranged in six rows rather than the two rows of six buttons each as shown in the figure. The first and the second rows provide the bass notes while the other four rows produce, in the following order: the major, minor, dominant-seventh, and diminished-seventh chords. A medium-sized accordion with a piano-style keyboard can cover in its melody section a frequency range from F_3 to A_6.

The mouth organ or harmonica, which comes in number of versions, is played by a performer's breath providing the air supply through both exhalation and inhalation. The instrument consists of a set of tuned free reeds mounted on a wood, metal, or plastic box, with channels leading from orifices to the reeds along one side of the box. Each channel connects to two differently tuned reeds, one reed operating under pressure (exhalation) and the other under a partial vacuum (inhalation). Each reed behaves as a cantilever bar in operation. To play a melody the mouth covers one or more holes, with the tip of the tongue providing some additional control. Harmonicas fall into three principal categories: the simple harmonica (a single row of 10 holes, 20 different notes), the concert harmonica (two rows, one row tuned an octave higher than the other), and the chromatic harmonica. The chromatic harmonica, although similar in appearance to the concert type, is really a set of two harmonicas, one placed above the other. The bottom instrument is tuned one semitone above the upper instrument. This instrument contains a slide which is moved by a knob to the right. When the slide is in, the upper holes are closed and the lower holes are open; the converse occurs when the slide is out. A fairly typical simple 10-hole harmonica can cover $2\frac{1}{2}$ octaves from C_3 to F_5.

The reed organ pipe contains an air-actuated reed coupled to a conical pipe, with the fundamental frequencies of the reed and the pipe itself fairly matching. In some installations, a combination of a conical pipe and cylindrical pipe is used. Because the pipe and the reed are intimately coupled, the resonant frequency is established by the combination of these two components. The resonant frequency can be changed by altering the resonant frequency of either the reed or the pipe. A reed organ pipe is customarily tuned by changing the effective stiffness of the reed by moving a tuning spring that is in contact with the reed. A cross-sectional view of the organ pipe mechanism is shown in Figure 18.30. The organ may also be tuned by changing the effective length of the pipe by a metal rollback in the side of the pipe near the open end. Both the shape of the reed and that of

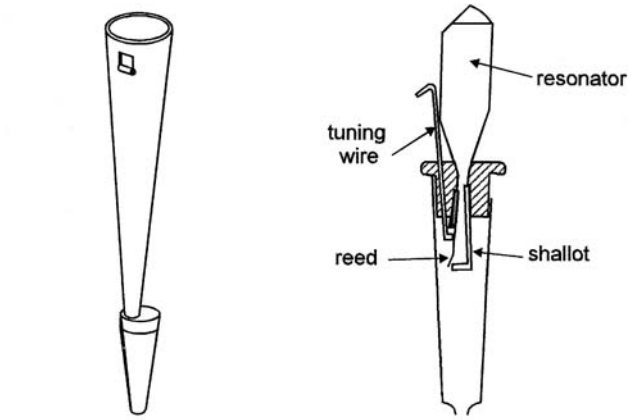


FIGURE 18.30. Reed organ pipe and reed mechanism.

the pipe determine the timbre. *Voicing* is the process of selecting and adjusting these components to yield the proper timbre. Reed organ pipes are generally designed to imitate orchestral instruments such as the trumpet, tuba, oboe, clarinet, the human voice, and so on. These pipes are classified as chorus and orchestral reeds.

The clarinet consists of a single-mechanical reed coupled to a cylindrical tube with a flared open end. Control over the effective length of the resonating air column is provided by a number of holes that may be opened or closed by fingers, either directly or through keys. Any change in the effective length of the air column results in a change of the resonant frequency. The reed position in the bottom of the mouthpiece functions in the manner described above for Figure 18.28. The throttling action of the reed changes a steady air stream into a saw-toothed pulsation that contains the fundamental and its harmonics. When sounded alone, the reed produces a sound rich in harmonics. The quality of the tone is improved when the reed is coupled to the cylinder air column. The clarinet produces a range covering more than three octaves, from D_3 to F_6. The clarinet's overall length is about 63 cm. The bass clarinet is a larger instrument, being 94 cm in overall length, which produces tones in a lower frequency range, from D_2 to F_5. This lower frequency range is obtained by doubling the tube on itself.

The saxophone operates in a similar fashion to the clarinet. The principal difference in its construction is that the diameter of the tube at the reed end is smaller, thus resulting in a lower acoustic impedance. Also, the coupling between the reed and the pipe is not as intimate as in the case of the clarinet. The buildup of the reed vibration is extremely quick, producing a sharp attack that is characteristic of the saxophone. Saxophones come in various sizes, including soprano, alto, tenor, baritone, and bass. The smallest of these, the soprano saxophone, employs a straight

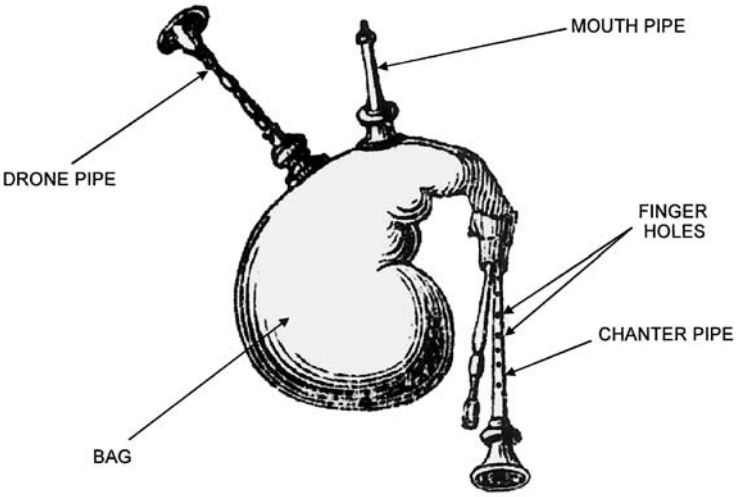


FIGURE 18.31. The bagpipe.

tube with a slightly flared open end; all the other saxophones use a curved mouthpiece and an upturned bell at the other end. Each of these types of saxophones covers a fundamental range of about two and a half octaves. The soprano produces $A\flat_3$ to $E\flat_6$; the alto from $D\flat_3$ to $A\flat_5$; the tenor from $A\flat_2$ to $E\flat_5$; the baritone from $D\flat_2$ to $A\flat_4$; and the bass from $A\flat_1$ to $D\flat_4$. Their overall lengths range from 40 cm for the soprano version to nearly 100 cm for the bass saxophone.

The bagpipe of Figure 18.31, the instrument so symbolically evocative of Scotland, contains one or more combinations of air-actuated reed coupled to a resonating pipe. A leather bag serves as a reservoir and air supply for actuating the reeds. Air is supplied to the bag by blowing with the breath. Because the reeds are supplied with a steady stream of air, the reeds sound continuously without interruption, distinguishing the bagpipe from other breath-blown instruments that do not provide such steady sounds. There are usually two or three fixed-frequency reed–pipe combinations that are called *drones*. In addition there is a variable pitch reed–pipe combination, or the *chanter*. On the chanter pipe, eight fingerholes are provided so that discrete frequencies can be achieved over the range of an octave. The chanter supplies the melody, and the drones produce a harmonious steady tone. The reed and pipe mechanism is similar to that of the reed–organ pipe. However, in the modern version of the bagpipe, the reeds in the drone are of the single-mechanical type and the reed in the chanter is a double-mechanical type.

Double mechanical-reed instruments use two mechanical reeds for throttling a steady stream of air to produce a musical sound. All of these instruments couple the double-mechanical reeds to a resonant air column. Such instruments include the oboe, English horn, oboe d'amore, bassoon, and sarrusophone.

Figure 18.32 depicts the operational principal of the double reed. When the reeds are in the normal position, i.e., they are slightly apart, air is forced through

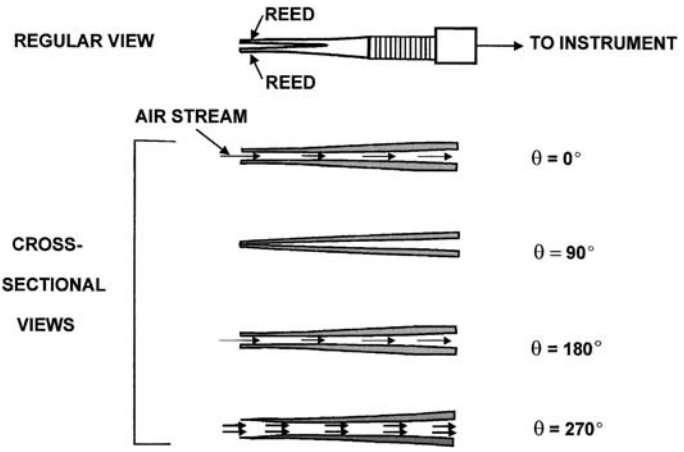


FIGURE 18.32. The action of a double reed. Air particle velocities are indicated by arrows.

the opening between the reeds. The high velocity of the air reduces the pressure between the reeds, in accordance with the effect of Bernoulli's principle. This causes the two reeds to be forced closer to each other, thereby constricting the airflow. With the airflow now reduced by the constriction, the pressure increases, causing the reeds to spring back toward their original position, but owing to their momentum, they moved beyond their original positions. The opening is now at its largest, and the airflow is increased accordingly. The internal pressure on these reeds are now quite small, so they return to their original position, and the cycle begins again with this sequence of events repeating at the rate of the resonant frequency. Thus, a steady air stream undergoes a throttling action that generates a saw-toothed signal that contains the fundamental and all of its harmonics.

The oboe consists of a double-mechanical reed such as that shown in Figure 18.32, coupled to a conical tube with a slight flared mouth. The effective length of the resonating air column is controlled by the number of holes that are opened or closed by the fingers either directly or through keys. The oboe covers three octaves from B♭ to G_6; the overall length of this instrument is 62 cm. The English horn resembles the oboe in most respects, especially in the key and fingering system, with the principal difference being that the double-mechanical reed is coupled to a tapered conical tube that terminates in a hollow spherical bulb with a relatively small mouth opening, thus producing a unique timbre. The fundamental frequency range of this 90-cm long instrument covers less than three octaves, from E_3 to $B♭_5$. Approximately 70 cm long, the oboe d'amore is a smaller twin of the English horn, and it ranges over nearly three octaves from $G\#_3$ to $C\#_6$.

The bassoon is a noble-sounding instrument that consists of a double-mechanical reed coupled to a conical tube that is doubled back on itself so that a lower frequency range is provided without compromising the portability of the instrument itself. There is no appreciable flare at the mouth. A set of holes on the side can be opened

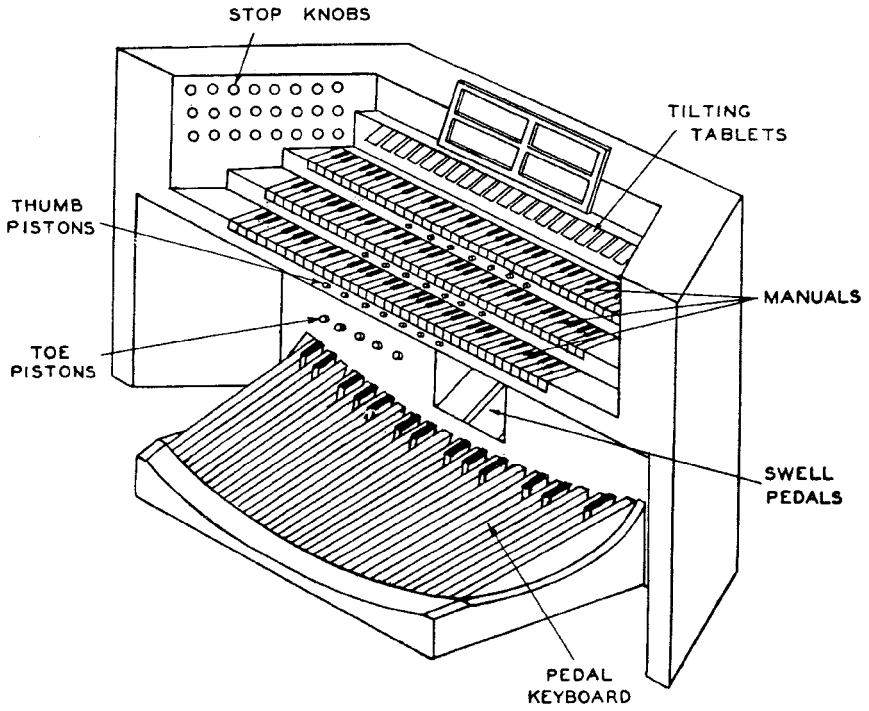


FIGURE 18.33. A contemporary organ console (from Olson, 1967).

or closed by fingers either directly or through the use of keys to determine the effective length of the resonating air column. The overall length of the bassoon is about 123 cm, but the doubled conical air column is about 245 cm long. The bassoon covers roughly three octaves from $E\flat_1$ to $E\flat_3$. The contra bassoon can be considered the bigger brother of the bassoon, but its tube is folded several times to yield an air column as long as about 480 cm while keeping the overall length of the instrument to a manageable 127 cm. The fundamental frequency range runs from B_0 to F_3.

The sarrusophone, essentially a double-mechanical reed coupled to a folded brass tube of conical bore with a flare at the open end, comes in several sizes covering different fundamental ranges. The effective length of the air column is varied by a number of holes that are opened and closed by cover valves operated by keys. The most common type of sarrusophone is the contrabass type, whose fundamental frequency range from $D\flat_1$ to B_3.

The modern organ illustrated in Figure 18.33 is really more than one type of instrument. It is considered to be a combination mechanical-reed and air-reed instrument. It consists of a large number of flue- and reed-type pipes controlled directly by manual and pedal keyboards and less directly by stops, couplers, and pistons. The organ console in Figure 18.33 is shown to contain three manuals, a pedal keyboard, tablet couplers, thumb and toe pistons, and swell pedals. Organs

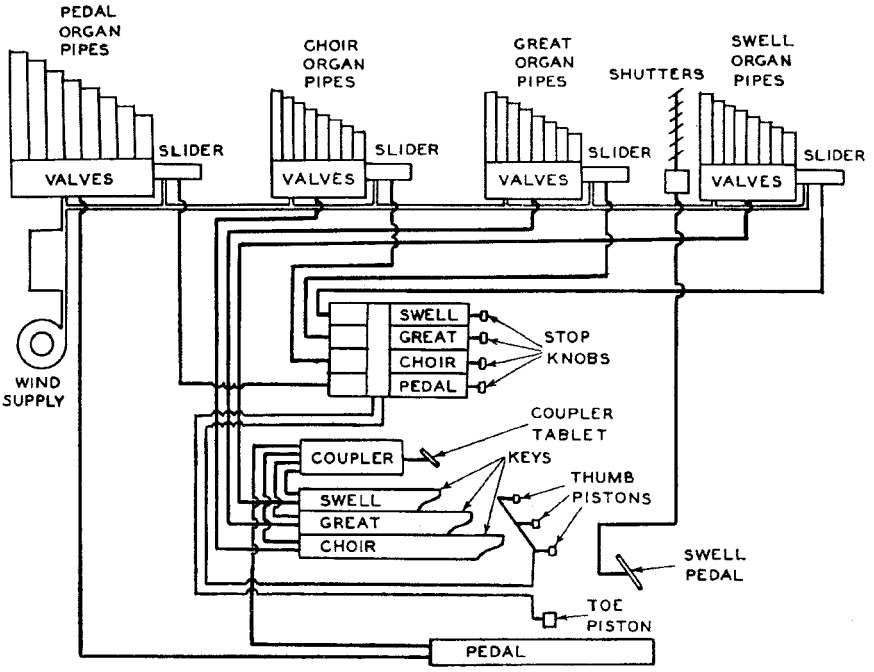


FIGURE 18.34. Schematic of the elements of the modern organ (from Olson, 1967).

and organ consoles, it should be mentioned here, are not standardized. The organ may be constructed with one or more manuals and with or without a pedal keyboard. In a five-manual organ, the functions of the manuals are as follows: The first (or the lowest) controls the pipes in the choir organ; the second controls the pipes of the great organ; the third controls the pipes of the swell organ; the fourth controls the pipes of the solo organ; and the fifth controls the pipes in the echo organ. The pedals control the pipes in the bass organ. In a number or more elaborate installations, other instruments have been added to pipes (e.g., gongs, cymbals, gongs, drums, etc.). Yet other installations contain as many as seven manuals. The typical pipe organ in use today has two to five manual keyboards and a pedal keyboard.

It can be seen from the schematic of Figure 18.34 that the organ is a combination of several organs, each organ having a different kind of pipes. We have earlier described the flue organ pipe and the reed organ pipe. Pipes can be classified into following basic groups: the diapasons, flutes, strings, and reeds. The more common flue and reed pipes are illustrated in Figure 18.35.

In the evolution of the organ, it was found that the tones could be improved by grouping a set of pipes to sound simultaneously by depressing a single key. Thus each key can represent any number of pipes, and each combination of pipes is called a *stop*, a term which is also applied to a knob along the sides of the console. Each stop knob controls a specific group of pipes.

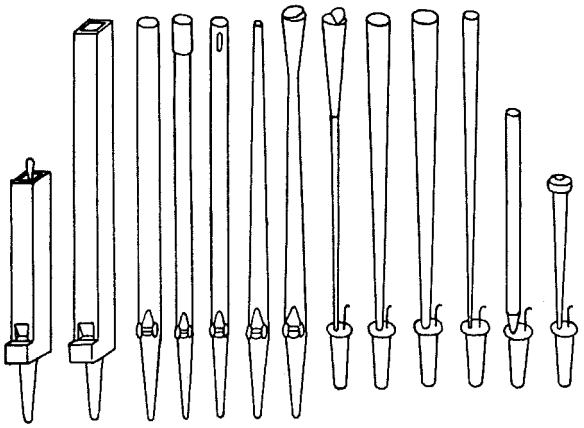


FIGURE 18.35. Different types of organ pipes (from Olson, 1967).

The pedal organ features the largest flue and reed stops. The great organ, the choir organ, and the swell organ feature decreasing steps of flue and reed stops. The swell organ is placed in an enclosed space equipped with shutters in the wall between the wall and the audience. The opening and closing of these shutters are controlled by the swell pedals in the organ console. The solo organ, if there is one, is also enclosed in a swell box.

Just above the top manual, a row of tilting tablets serve as couplers. This coupling system allows the actuation of a mechanism associated with more than one key by simply pressing one key. The different manuals can be interconnected by this coupling arrangement; and the pedal keyboard, for example, can be tied to any manual by this coupling procedure.

In the early organs, keys were connected directly to valves that controlled the airflow to the pipes. This required considerable strength on the organist's part to press the keys. The action was improved later by the introduction of a pneumatic system, in which the key operates a small pneumatic valve requiring a lighter force, which in turn operates a valve connected with the pipes. But this resulted in slow action owing to the slowness of the propagation rate of air impulses from the key to the valves at the organ pipes. As a result the console had to be located very near the pipes. Except for small organs, modern organs now use either electropneumatic or electric action to actuate the pipes. In the electropneumatic action, pressing down on the key energizes an electromagnet to move a set of valves in an air chamber, causing bellows to expand and move a linkage that opens up a larger valve which in turn lets in the air supply into the wind chest to activate the organ pipes. The advantages of the electropneumatic system are: a light force is required to press the key, the action is swift, the keys can be interconnected by merely flipping a switch, and the console can be located almost any distance from the organ and even moved about, because the wiring connecting the console to the different organs can be grouped together inside a reasonably small, flexible cable. The all-electrical

action essentially constitutes a servomechanism setup, where the key activates a relay that opens the wind chest to the organ pipes.

Because of the large size and number of pipes, organs require large amounts of air for actuating the pipes. The air pressure required is quite small, ranging from about 1 kPa gauge pressure for the early hand-powered organs to as much as 12 kPa air gauge pressure in some modern units. The air supply for modern organs is provided by an electric fan or an electrically driven centrifugal pump.

Lip-Reed Instruments

In these types of instruments, the lips serve as the reeds and the air supply is provided by the lungs. The puckered upper and lower lips can be imagined as a pair of double-mechanical reeds of Figure 18.33, with the operational principle being virtually the same. The frequency of the pulses, which occur as a result of the lips throttling the airflow, corresponds to the resonant frequency of the lips combined with the associated instrument. Because the coupling between the lips and the instrument's mouthpiece is fairly loose, the fundamental resonant frequency of the lips must correspond to that of the instrument. However, the combined resonant frequency of the horn and the lips can be varied slightly up or down by changing the tension of the lips, thus changing the resonant frequency of the system. The modern lip–reed instruments are the bugle, trumpet, cornet, trombone, French horn, and tuba—collectively referred to as the *brasses*.

The simplest of the brass instruments is the bugle, which consists of a cupped mouthpiece attached to a coiled tube with a low rate of taper, terminating in a bell-shaped mouth. The length of the air column is fixed; there are no valves, and the number of notes that can be played is limited. The overall length of the standard bugle is about 57 cm, and the notes it can play are C_4, G_4, C_5, E_5, G_5, $B\flat_5$, and C_6.

As it is equipped with three sets of keys that operate piston valves, the trumpet is a more versatile instrument than the bugle. The trumpet is constructed of a coiled tube almost 3 m in length with a slight taper, terminating in a bell-shaped mouth. The first third of the tube is almost circular, with the remainder slightly conical except for the last 30 cm that flares into a bell-shaped mouth. Pushing down on the keys of the piston valves adds to the effective length of the tube. With three valves closed and opening them singly, opening in three different combinations of pairs, or with all three valves opened, it becomes possible to obtain eight different lengths of the resonating tube. A tuning slide or bit is provided so that the resonant frequencies can be matched with other instruments. The trumpet, which has an overall length of 57 cm, covers about three octaves, from E_3 to $B\flat_5$. A mute in the form of a pear-shaped piece of metal or plastic can be inserted into the bell in order to change the quality of the tone and attenuate the output.

Only about 36 cm in overall length, the cornet is a smaller version of the trumpet. In addition to its size, the cornet has a bore that is tapered rather than

cylindrical. It covers the frequency range from G_3 to $B\flat_5$. The French horn, arguably the most beautiful appearing brass instrument, consists of a mouthpiece coupled to a slightly tapered coiled tube of 365 cm length, which terminates in a large bell-shaped mouth. The modern version of the horn is equipped with three sets of rotary valves controlled by three keys. This valve system provides eight different lengths for the resonating tube, and hence a series of different resonant frequencies corresponding to the notes of the musical scale. The large size of the mouth of the French horn renders it possible for the player to inert his hand and raise or lower the pitch or to produce a sound (e.g., muted) effect. The French horn can sound over three octaves, from B_1 to F_5. Its overall length is nearly 58 cm.

The trombone and the bass trombone differ from the other brass instruments in that they each feature a telescoping section of the tube that can be moved by the performer to vary the length of the tube and hence the resonant frequency. The trombones are the only instruments besides the violin family that can provide a continuous glide through the musical scale. The pitch is determined by the position of the slide as well as the performer's lips and the applied lung pressure. As with the case of performers of the violin family, the player must possess an accurate sense of pitch and the ability to produce the correct note. The trombone's U-shaped tube is nearly 3 m long, with a slight conical taper that culminates in a bell-shaped mouth. It is capable of covering nearly 2½ octaves, from E_2 to B_4. The bass trombone is larger, covering three octaves from A_1 to $G\flat_4$.

The tuba's mouthpiece is attached to a coiled tube nearly 6 m in length that gradually increases in its cross-sectional area until nearly the end when it terminates in a large bell-shaped flare. Three piston valves are provided, each adding a different length to the tubing. Thus, eight different lengths are provided, in the same basic manner as the trumpet or cornet. Some versions of the tuba include a fourth valve, with a corresponding increase in the number of different resonant frequencies. Largest of the brass instruments, a typical tuba, measures approximately 1 m. The tuba covers three octaves from F_1 to F_4. Different versions of the tuba exist, with a variety of sizes and forms. The sousaphone (named after band conductor/composer John Sousa of the "March King" fame) is the largest of the tuba family.

In concluding this section on wind instruments, it might also be mentioned that the human voice also classifies wind instruments, i.e., it is a reed instrument in which the vocal cords serve as the reeds. The mechanism of the voice consists of three sections: (1) the lungs and associated muscles to serve as the air supply; (2) the larynx bearing the vocal cords for converting the airflow into a periodic modulation; and (3) the vocal cavities of the pharynx, mouth, and nose, all of which help vary the tonal content of the output of the larynx. A sectional view of the voice system in the human head is shown in Figure 18.36. The frequency of the vibration is governed by the tension of the vocal cords, the inertance, and the combined acoustical impedance of the vocal cords and the vocal cavities. The vocal mechanism is a complex one, entailing a number of acoustical elements

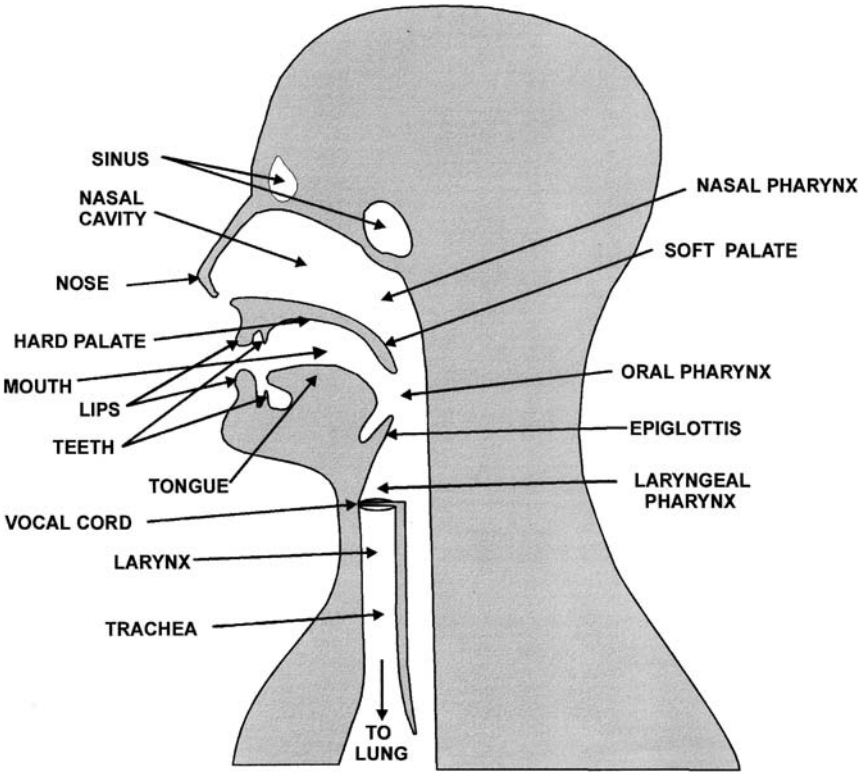


FIGURE 18.36. The human voice mechanism in a sectional view of the head.

that can be varied by the singer to yield a wide variety of tones, which differ in harmonic content, quality, loudness, duration, growth, and decay. According to Machlis,

> the oldest and still the most popular of all (musical) instruments is the human voice. In no other instrument is contact between the performer and medium so intimate. In none other is expression so personal and direct. The voice is the ideal exponent of lyric melody and has consistently been the model for those who made instruments as well as for those who played them.

A human voice can range over two octaves, but variations do occur among different individuals as to the range, to say nothing of the beauty or quality of tones. It is a rare voice such as Yma Sumac's that can cover four octaves or more. An effortless coloratura such as Joan Sutherland, a powerful *heldentenor* such as Lauritz Melchior, or a sonorous basso such as Alexander Kipnis or Martti Tevala comes along once a generation. On the average, a soprano can range from C_4 to C_6; an alto, from G_3 to F_5; a tenor, from D_3 to C_5; a baritone, from A_2 to G_4; and the basso, from E_2 to D_4.

18.11 Percussion Instruments

Percussion instruments are made to sound by striking or shaking. The vibration system that is excited by an impact may be a bar, rod, plate, membrane, or a bell. Two classes of percussion instruments exist: viz. definite pitch, in which the tone has an identifiable pitch or fundamental frequency; and the indefinite pitch, in which the musical sound does not have a definite pitch or fundamental frequency. The former category includes the tuning fork, xylophone, celesta, glockenspiel, bells, and chimes. Drums, the triangle, tambourine, castanets, and cymbals fall into the latter category. The tuning fork, discussed in Chapter 5, is used as a standard frequency sound source. The resonant frequency of a tuning fork is determined by its dimensions and its material.

The xylophone consists of a number of metal or wood bars, having specific resonant frequencies, mounted horizontally and supported on soft material at two nodal points. The vibration of each bar is that of a free bar, with the resonant frequency dependent on the dimensions of the bar. A pipe acts as a resonator by being coupled to each bar to improve the coupling between the bar and the air, thus yielding greater sound output. Felt-covered hammers are used to strike the bars. Differential models of xylophones are available to provide fundamental frequency ranges of two to four octaves, generally from C_3 to E_7. The marimba, which has African and South American roots, is basically an enlarged version of the xylophone that usually covers about five octaves, from F_2 to F_7.

The glockenspiel (also called orchestral bells) resembles a small xylophone but small wooden hammers are used to strike the notes. The bell lyre is another version of the glockenspiel. A set of steel bars is mounted on a lyre-shaped frame. A ball-shaped hammer is used to strike the bars. The bell lyre is used in marching bands as it is small enough to be hold almost vertically by one hand while the other hand holds the hammer. The glockenspiel can be constructed to cover up to three octaves. The frequency range usually falls between C_3 and C_6.

The celesta contains a series of resonant steel bars with specific resonant frequencies. The bars are actuated by hammers linked to keys forming a keyboard. Thus, the celesta resembles a small upright piano. The steel bars are suspended above wood resonating boxes, which are designed to improve coupling of the bars to the air, thus improving the sound output. Dampers are provided, and a single pedal, called a *sustaining pedal*, removes the dampers so that the energy of the vibrating bar may be dissipated over a longer period. The celesta is a four-octave instrument that ranges from C_4 to C_8.

Chimes or tubular bells are a series of resonant brass tubes suspended vertically from a wooden frame, and a foot pedal controls a damping bar. The tubular bells, whose resonant frequencies correspond to the notes of the musical scale and are played by being struck by a wooden mallet, range up to two octaves, generally between G_1 and G_3. The tubular bells are intended as substitute for real bells. Bells themselves also constitute musical instruments. They are actuated through side-to-side movements by clappers that hang loosely inside the bell. The fundamental frequency of a bell depends on the geometry of the bell, the wall

thickness, and the density and moduli of the metal. A *carillon* is a set of fixed bells tuned to a musical scale. A number of arrangements have been devised to play the bells: the bells may be struck directly by hammers held in the hand of the carillonneur, or they may be struck by a clapper through linkages to a keyboard that is played by a carillonneur. With a mechanical setup, fists are used with considerable force to operate the keys. In electrified carillons, the clappers are activated by solenoids activated by the keys of a keyboard. In this setup the force required to move the keys are greatly decreased to a level no greater than that for playing a piano or an organ. A carillon usually covers the range of three or more octaves.

The kettledrum or *timpani* consists of a large hemispherical bowl over which a specially treated leather skin is stretched. Several types of sticks are used to strike the membrane, according to the percussion effect desired. These sticks may have striking surfaces made of sponge, felt, rubber, or wood. The kettledrum may be tuned and changed in its fundamental frequency when so desired by varying the tension of the membrane through adjustment of head screws and by movement of a tuning pedal. The kettledrum emits a low-frequency sound of a definite pitch. The pedal provides a quick yet accurate variation of the pitch that a melody can be played on the kettledrum. Two versions of the timpani are standard, the smaller one having a diameter of 58.5 cm and the larger one, a diameter of about 76 cm. The smaller unit covers the frequency range from $B\flat_2$ to F_3, and the larger from F_3 to C_3.

An interesting development of rather recent past is that of the steel drum, which is the principal instrument of Trinidad, where it originated, and of other Caribbean nations. Steel drums developed in Trinidad in the 1940s because the use of "bamboo tamboo" sticks, which supplied rhythmic cadences at the annual Carnival festivities, were proscribed after matters got out of hand and these sticks were used as weapons in melees between rival bands and fights with the police. But these musicians were determined to continue their musical ways, and their resourcefulness turned them to buckets, garbage cans, brake drums, and whatever was available. The first steel drums were rhythmic rather than melodic. With the availability of the 55-gallon steel drums used by the petroleum industry after World War II, the continued development of tuned steel drums continued apace to the point that they must now be considered rather sophisticated examples of musical instruments and intricate workmanship. These instruments are now achieving ever-greater popularity in North American and in Europe, to the point where an aspiring musician can study steel drums in a well-developed curriculum at Northern Illinois University.

A variety of steel drums or pans are available. Steel drums are generally fabricated from 55-gallon oil drums. The drums in a steel band may consist of a variety of drums that have been termed soprano, ping pong, double tenor, guitar, cello, and bass. The soprano or ping pong can have anywhere from 26 to 36 different notes, but a bass drum may have only three or four notes. Because of the relative paucity of notes on a single drum, the bass drummer is likely to play on a half a dozen drums in the same manner as a timpanist in a symphonic orchestra.

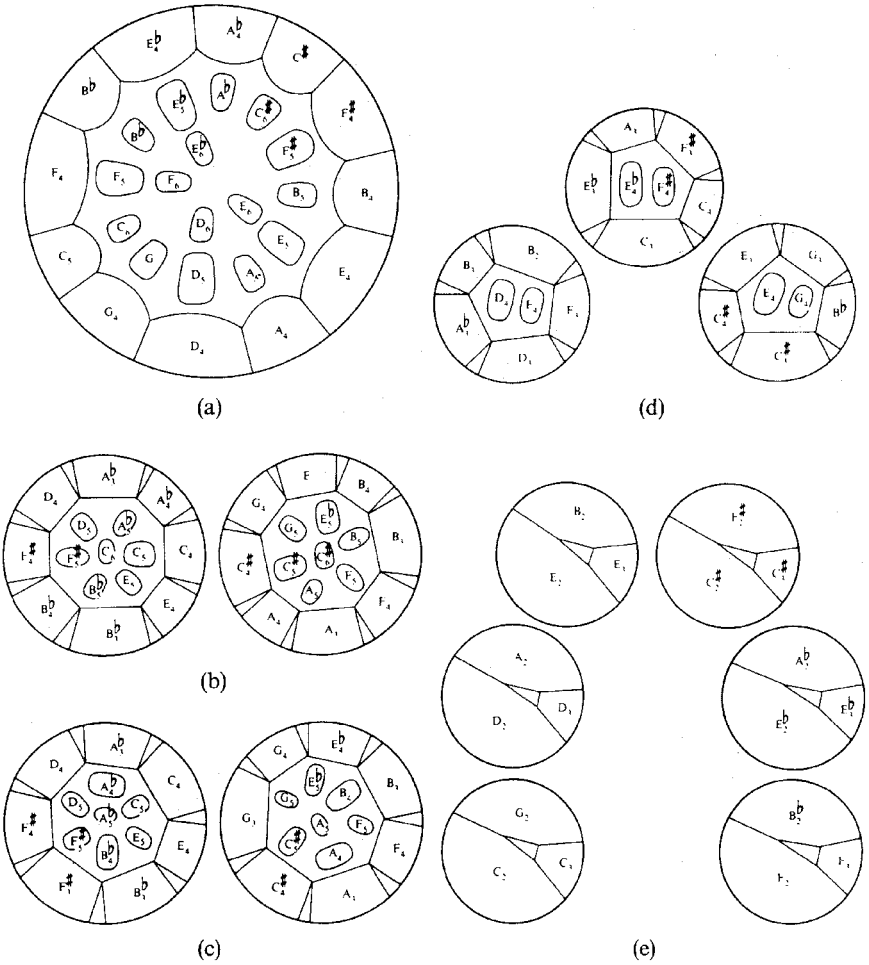


FIGURE 18.37. Layout and tonal segmentation of different steel drums: (a) lead pan (soprano), (b) double tenor (alto), (c) double second (tenor), (d) cello, and (e) bass (from Fletcher and Rossing, 1998 with acknowledgment to Clifford Alexis).

These drums are fabricated by first hammering the head of an oil barrel into the shape of a shallow basin. A pattern of grooves is cut with a nail to delineate sections of different notes. Each section is "ponged up" or shaped with a hammer. After the drum is heat-tempered, each section is tuned by the adept use of a hammer. Figure 18.37 shows the layout and the tonal segmentation of different types of steel drums.

Indefinite pitch instruments include the triangle, which is a steel rod bent into a triangular shape that is struck by a metal beater. Because the triangle undergoes a complex vibration when struck, the fundamentals and the overture form an indefinite noise mixture. The bass drum, a hollow cylinder of wood or metal

covered at each end by a stretched membrane or parchment or skin, is struck with a softheaded stick; its vibration is quite complex, owing to nodal contributions by the air column and the stretched membrane. Thumbscrews supply the means of adjusting the stretch of the drumskin. The bass drums are constructed in sizes ranging from 61 cm to as much as 3 m, with approximately 78 cm being the most common diameter. The military drum is a smaller rendition of the bass drum, being about 40 cm in diameter and about 30 cm in depth. At nearly 36 cm in diameter, the snare drum is even smaller than the military drum. Across its lower head, cords of catgut are stretched so these cords vibrate when struck by the membrane. This results in a sound output that is buzzing and rattling.

The tambourine consists of a hoop of wood or metal with a single membrane stretched over one end. Smaller circular metal disks are inserted in pair in the hoop and loosely strung on wires. During performance, the tambourine is held in one hand and the membrane struck with the fingers or palms of the other hand.

A cymbal is a brass circular disk that is concave at its center. Its vibration is that of a circular plate supported at its center. Cymbals equipped with handles can be hold, one in each hand, and struck together. A single cymbal can be supported in a fixed horizontal position on a stand or on some other support and struck with a drumstick. A sock cymbal consists of a pair of horizontally positioned cymbals, one being fixed and the other being moved by a pedal. Cymbals are made in various sizes, from as little as 5 cm in diameter to 51 cm.

Castanets, used to accentuate Spanish-style music, are hollow shells of hard wood. These are hold in the hand and clapped together. They can also be mounted on a handle in such a manner that shaking the handle clacks the castanets.

The gong is a round plate of hammered bronze, with the edges curved up so that the shape takes on the appearance of a shallow pan. The gong is usually suspended by strings attached to a frame. It is excited by striking with felt-covered hammer, with the sound output resembling a heavy roar. The diameter of the gong ranges from about 45 cm to 90 cm.

18.12 Electrical and Electronic Instruments

A tone can be generated by a number of electrical means. The means of producing a tone can be achieved by interruption of an air stream (as with a siren that consists of a rotating perforated wheel), an electrically driven diaphragm (as in the automotive horn), and electronic alternator and loudspeaker combination (electric organs), an electronic oscillator–amplifier–loudspeaker combination (the electric piano is an example). The electric guitar is an adaptation, where an electromechanical transducer is attached to the bridge of the instrument. The transducer converts the mechanical vibrations into a correspondingly varying electrical signal that is transmitted to an amplifier. The amplifier increases the strength of the signal and sends it on to a loudspeaker that converts the electrical signal into the corresponding acoustical output. A volume control in the amplifier or integrated into the guitar

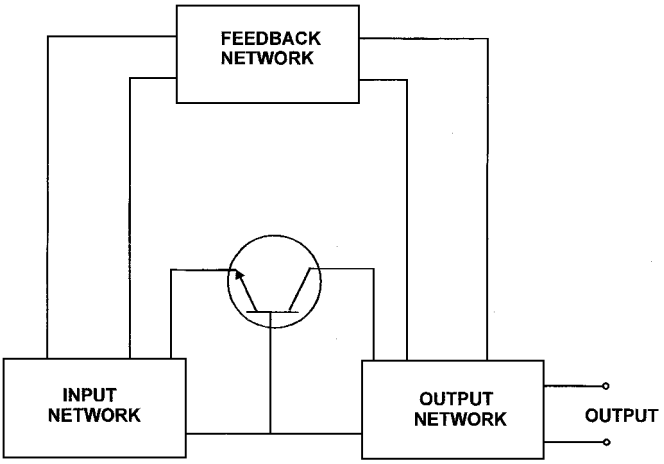


FIGURE 18.38. Schematic of a transistor oscillator network.

makes it possible to adjust the output sound level. The sound of the guitar resembles that of the conventional instrument.

Generating music by electronic means has a history older than most people realize. Early electronic instruments include the Theremin (1919), the Ondes Martenot (1928), the Trautonium (1928), and the Hammond organ (1929).[2] The early means of electrically generating music were based on the technology that prevailed in the recording and sound reproduction industry of the time. Electronics progressed rapidly during World War II and even more rapidly with the development of transistors and eventually with the advent of microcircuitry.

Because of the rising costs of traditional pipe organs and the desire then prevailing for creating sustained tones, much attention has been conferred on developing and marketing the electronic organ. Vacuum-tube oscillators and, later on, solid state oscillators were used to generate the tones in an electronic organ. The schematic of a transistor oscillator network is shown in Figure 18.38. Power from the output network is fed back to the input network. Oscillations occur when more power is developed in the output than is necessary for the loss in the input circuitry, combined with appropriate phase relations between the current and voltages in the input, feedback, and output networks. Under these conditions, the reactions consist of regular surges of power at a frequency that depends on the constants of the resonant elements in the input or output networks. The resonant elements include quartz, crystal, tuning fork, inductance–capacitance, and so on. These electronic systems are capable of simulating the wave shape of almost any musical instrument. One type of electrical organ makes use of air-driven reeds and electrostatic pickups. Most of the electronic organs consist of two manual keyboards and a

[2] An interesting review of these instruments and their history may be found in texts by Rossing (1990) and Strong and Plitnik (1983).

pedal keyboard and incorporate a system of stops and couplers. Loudspeakers may be housed in a separate cabinet; and the driving electronics housed in the console.

The electronic organ has been displaced in the 1980s by the growing popularity of synthesizers. Modular synthesizers made their appearance in the 1960s in both analog and digital versions. The analog synthesizer consists of a group of signal generating and signal processing modules operating on electrical voltages. The modular approach to analog synthesis is embodied in three fundamental voltage-controlled modules: (1) the voltage-controlled oscillator (VCO), (2) voltage-controlled amplifier (VCA), and (3) voltage-controlled filter (VCF).

The VCO is really a function generator that produces a periodic voltage signal and is the initial source of pitched sounds. The frequency of the periodic signal is determined by a control voltage. Usually when the control voltage increases by one volt, the frequency doubles (i.e., it goes up one octave). The control mode is therefore exponential, which fits beautifully into the scheme of things in music and in electronics. Controllers can be fabricated in the form of keyboards, and musical effects of modulation (trill, vibrato, or glissando) are independent of the DC value of the control voltage. Switching keys of a melody require only the addition of a constant to the series of control voltages representing the notes of a melody. The exponential feature inherent in the voltage control is appropriate electronically, given the fact that the collector current in a bipolar transistor is an exponential function of the base-emitter voltage. In addition to being stable and accurate, the VCO needs a subsequent filter to shape the waveforms for the desired tone color. A high-purity sine wave output is normally needed, particularly if the VCO output is processed by a nonlinear waveform shaper, distorter, or a ring modulator or an FM synthesizer. Otherwise an excessively dense spectrum that is downright unmusical can be created.

The voltage-controlled amplifier is a voltage amplifier with a gain that depends linearly upon a control voltage. An exponential control is generally available, but it is more usual for the amplifier gain to grow linearly with the control voltage with a slope of unity gain for a 5-V control input and a maximum gain of 2. Ideally the gain should be zero when the control voltage is zero or negative. It is the task of the VCA to turn on tones. Since the oscillators run continuously, the VCA bears the responsibility of shaping the amplitude envelope of the tone. Here the control voltage comes from an envelope generator. When a note is begun from a keyboard (or sequencer), the envelope generator begins a transient phase. The envelope generator sustains an appropriate level for the duration of the tone and ends the tone with an exponential decay. As with the tone control parameters on a synthesizer, the performer can set the envelope transients, the sustain level, and decay time by adjusting individual potentiometers on a modular analog instrument and through programming on a digitally controlled keyboard. Special timbres can be created by adding two sounds, one of them with a delayed onset. To achieve this the output of two VCAs can be added, where one of them is controlled by

a delayed envelope. A tremolo can be added by a VCA controlled by subsonic periodic waveform.

A VCA can be utilized to multiply two separate audio signals, producing amplitude modulation, with one signal serving as the carrier and the other serving as a modulator. The output consists of the spectral components of the carrier plus sidebands. More commonly used than amplitude modulation is the *balanced modulation* or ring modulation. While the VCA is a two quadrant multiplier (i.e., the control voltage must be positive for a nonzero output), the balanced modulator is a four-quadrant multiplier.

As the terminology implies, the definitive frequency of a voltage-controlled filter (VCF) is controlled by a control voltage. In a low-pass VCF, for example, the cutoff frequency increases from a low value to a high value as the control voltage increases. A scaling of 1 V per octave is the standard. There is a large number of different filtering circuits available, but predominant design seems to be the state-variable filter that is constructed from two integrators and a summer. This design bears the advantage of incorporating (a) three simultaneous outputs (low-pass, bandpass, high-pass), (b) a constant value of Q independent of band frequency, (c) adjustability of Q through controlled feedback, and (d) stability of the sine oscillation mode. This type of filter features asymptotic slopes of ± 12 dB/octave in the low-pass and high-pass outputs and ± 6 dB/octave for the band-pass output. A four-pole low-pass filter with an asymptotic slope of ± 24 dB/octave is often preferred by musicians to deal with the more subdued sounds, particularly in the lower registers.

Voltage control is applied in filters to generate dynamic effects, particularly musical attacks. As an example of this application, consider the fact that the harmonics of a brass instrument entering the attack phase are in the order of ascending frequencies (Risset and Mathews, 1965). This effect is electronically simulated by raising the cutoff frequency of a low-pass VCF with an envelope generator at the onset of the tone. On the other end of the tone, the decay of an envelope generator is utilized to decrease the cutoff frequency of the low-pass VCF to simulate the proclivity of high-frequency modes to become damped more rapidly than the low-frequency modes in a free vibrator (such as a percussion instrument).

A number of other musical effects can be generated. For example, in a delay-and-add application for a digital delay line, specifically the generation of reverberation, an input signal is passed through a delay line that is tapped at different delay intervals. Signals at these taps are fed back into the input as a weighted sum. This weight determines the reverberation time of the system. In the *flanger* use is made of comb filtering. The flanger uses a single delay-and-add. The delay time is slowly modulated so that the comb filtering changes with time, with typical delays in the order of 2 ms. *Chorusing*, which is the attempt to make a single-voice sound like a group of many, functions in the same manner, except several modulated delay-and-add circuits are used to transmit to different channels, and the delays are considerably longer, typically in the order of 10 ms.

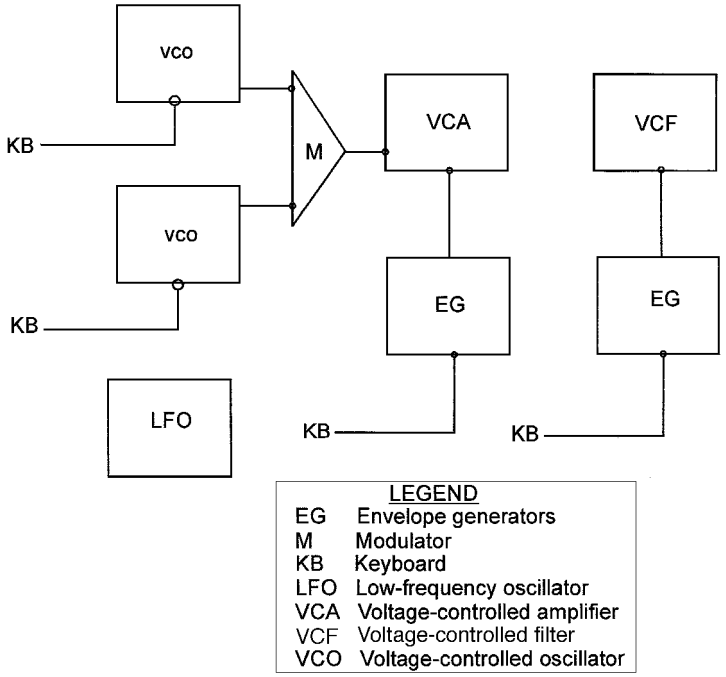


FIGURE 18.39. Arrangement of modules for an analog synthesizer. The signals from the two keyboard-manipulated voltage-controlled amplifier (VCO) are summed into a voltage-controlled amplifier (VCA) and then relayed to a voltage-controlled filter (VCF). Both VCA and VCF are controlled by the envelope generators (EG). A third oscillator operating at low frequency (LFO) is on hand for modulation of any voltage-controlled module.

In the early years of the synthesizer, control was a cumbersome matter, involving patch cords for connecting analog synthesizers. The early digital synthesizers were even more cumbersome, none of them operating in real time. In the early 1970s, prepatched analog synthesizers were introduced, primarily for creating special effects in a live performance. The setup of the modules is given in Figure 18.39. In the mid 1970s, the modules of Figure 18.39 were combined into a single-integrated chip. One such chip was placed under each key of the keyboard, so each key essentially constituted the control for a miniature synthesizer. The PolyMoog was the first and the last instrument to be constructed this way. Subsequent designs now employ digital scanning keyboards and assign a specific sound synthesis chain to each key. The number of chains (generally 8, 12, or 16) establishes the maximum number of simultaneous notes.

The *sequencer* was introduced to provide additional automated control. The earlier sequencers generated several channels of control voltages in repetitive sequences. The analog sequencer could even replace a keyboard for generating a repetitive bass line. The sequencers have evolved into small computers which have almost unlimited musical capabilities. Thanks to computer memory, they

can recall and reproduce the control sequences for multiple parameters for all the voices in the entire performance. Data entry is achieved through the use of an organ-type keyboard, and the programming allows for editing the data and displaying or printing the data in the traditional musical notation.

18.13 The MIDI Interface for Synthesizers and Digital Synthesis

With rapid advances in the development and manufacture of microprocessors and equally rapid drop in the cost of these microprocessors, the digital control of synthesizer functions became inevitable. In 1983 the manufacturers of commercial synthesizers adopted the Musical Instrument Digital Interface (MIDI) standard, which expands the range of control possibilities for the musical performer by the adoption of an interface that makes it possible to transfer control data among different synthesizers or other processors as well as computers (International MIDI Association, 1983). The interface is a unidirectional asynchronous serial line operating at 31.25 kbaud (1 kbaud = 1000 bits/s). An MIDI word is defined as 10 bits long, 8 data plus start and stop bits. To enable long interconnections, the interface uses a 5-mA current loop. Repeaters with each MIDI-specification instrument allow for daisy chaining. Sixteen logical channels exist for data transmission, and instruments may be set to obey data on one channel or data on all channels.

Some musical tradition remains in the way MIDI commands are defined to emphasize the use of the keyboard in electronic instruments. Specific data words are defined for key on/off, key velocity, key aftertouch, and for encoding the position of modulation wheels (this device was originally introduced in the MiniMoog and it remains a standard). The performers are free to use the parameters as they like, and the MIDI has even been used to control stage lighting. Data of arbitrary length can be transmitted by concaternating MIDI words. Owing to its high data rate and flexible control structure, the MIDI control has proven to be very powerful and well accepted in the industry and by performers of popular music.

With the affordability and availability of digital processors, it becomes inevitable that digital oscillators can become precise enough to replace VCOs for improved flexibility and stability. Moreover, a single digital oscillator can be constructed to create several simultaneous voices.

A cost-effective digital technique is that of FM synthesis, which involves the generation of a frequency-modulated waveform given by

$$x(t) = \sin[\omega_c t + \beta \sin(\omega_m t)]$$

where ω_c denotes the carrier frequency and ω_m the modulating frequency; β is the modulating index which is given by $\beta = \Delta\omega/\omega_m$, with $\Delta\omega$ being the maximum frequency excursion. This FM algorithm provides for flexible addressing. A spectrum of the FM signal contains a component at the carrier frequency and a component that contains sidebands displaced in frequency by $\pm n\omega_m$, where n is

the sideband order. Sidebands of order n have amplitudes that are proportional to Bessel functions $J_n(\beta)$. As the modulation index is increased, the bandwidth of the signal increases. A dynamically varying spectrum is achieved by varying the modulation index as the tone is initiated. When ω_c and ω_m are commensurate, the spectrum may be harmonic; otherwise, it may be inharmonic.

The FM algorithm has been proven highly successful in creating both sustained and percussive musical sounds, but the principal disadvantage of the FM technique is that the control is unintuitive. Bessel functions are not monotonic functions of their arguments, so it becomes difficult or almost impossible for the performer sense what will happen if the modulation index is changed one way or the other. The Yamaha musical instrument company installed FM oscillators on large-scale integrated circuits in its DX series of instruments, the most popular electronic instruments ever. These instruments contain six or four dynamically controlled oscillators in a voice, together with possible feedback among the oscillators.

A more recent methodology of creating music electronically is that of *physical modeling* (*Computer Music Journal*, 1992). A mathematical model is developed for the production of a tone by a traditional instrument or by some other mechanical device. The model may result in a set of coupled, usually nonlinear differential equations that can be solved by computerized means. The solution to these equations is played through a digital analog computer. This is a most time consuming process and has been applied commercially for only a few years. It is hoped with this methodology that even greater subtlety can be captured in the reproduction of actual instruments.

18.14 The Orchestra and the Band

The orchestra is a fairly full-fledged group of musicians playing on string, wind, and percussion instruments under the direction of a conductor. An orchestra differs from a band in that the main body of the music is generated by string instruments, whereas the band is generally made up of musicians playing on wind and percussion instruments. Orchestras differ according to the demands of the music, as the requirements differ for playing symphonies and overtures, or providing accompaniment for operas, musicals, oratorios, or incidental music for theatrical performances, or ballet or dance music.

Four groups usually constitute a symphony orchestra, namely, the strings, woodwinds, brass, and percussion. Violins, viols, violoncellos, contrabasses, harps, and (usually one) piano comprise the strings section. Woodwinds include the flutes, piccolos, oboes, bassoons, English horns, contra bassoons, clarinets, and bass clarinets. Brass instruments include trumpets, trombones, French horns, and tubas. In the percussion group there may be included timpani, bass, military and snare drums, tambourines, gongs, celestas, glockenspiels, tubular chimes, xylophones, castanets, and triangles. In order to achieve sufficient sound output for suitable artistic effects in large halls, the orchestra may have to consist of 80–120 players. An organ may be added to the performance where required. Some performances

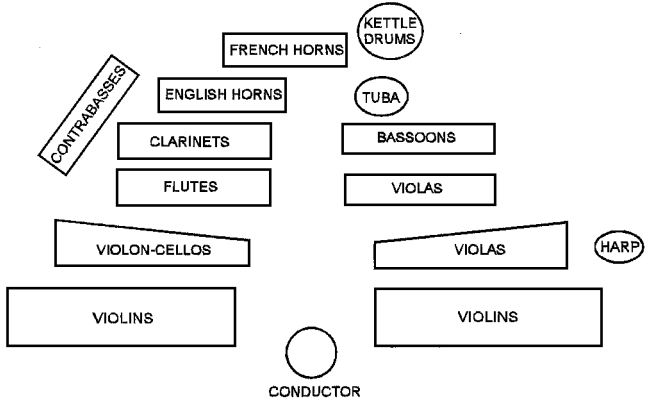
may require fewer players, and in some situations, a limited-size orchestra can achieve a dramatic effect through the use of sound reinforcement systems. Figure 18.40(a) shows a plan view of the arrangement of the instruments of a symphony orchestra and the placement of its conductor. Figure 18.40(b) illustrates the layout of a smaller orchestra, and 18.40(c) that of a small dance band. An orchestra for performances of popular and dance music may number anywhere from 5 to 25.

A band consists of musicians playing wind and percussion instruments. They are generally more suited for outdoor performances, since the main body of the tone is produced by the brass and woodwind sections, both of which can produce considerable acoustic power in free space. Bands help to sustain unison in marching groups and can whip up enthusiasm at athletic meets. A standard band generally features a complete range of woodwind, brass, and percussion instruments. The number of performers can vary from 25 to 50 or even more. The military band resembles the standard band except that it may include a fife, drum, or bugle corps. A concert band may include contrabasses, timpani, harps, and other instruments that lack the portability for marching. There is virtually no difference between the dance band and the dance orchestra. Table 18.4 shows the components of a typical symphony orchestra and those of a standard band.

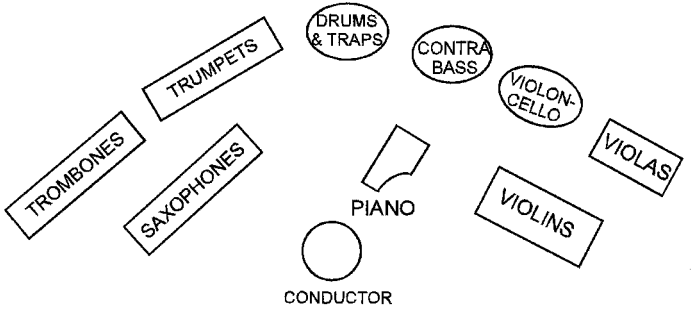
TABLE 18.4. The Elements of a Typical Orchestra and of a Typical Band.

Orchestra[a]		Standard Band[b]	
Instruments	Number of Players	Instruments	Number of Players
First violins	10 to 20	Flutes	4
Second violins	14 to 18	Piccolo	1
Violas	10 to 14	Clarinets	14
Violoncellos	8 to 12	Oboes	2
Contrabasses	8 to 10	Bassoons	2
Flutes	2 to 3	Sarrusophones	2
Piccolos	1 to 2	Saxophones	4
Oboes	3	Cornets	4
English horn	1	Trumpets	2
Bassoon	3	French horns	4
Contra bassoon	1	Trombones	4
Clarinets	3	Tubas	6
Bass Clarinet	1	Snare drum	1
Trumpets	4	Bass drum	1
Trombones	4	Percussion	1 to 5
French horns	4 to 12		
Tuba	1		
Timpani	1		
Harp	1		
Percussion	1 to 5		

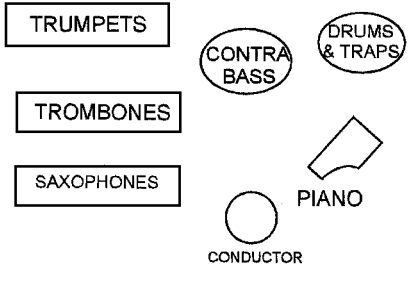
[a] In addition, other instruments such as a piano or an organ may be included. Percussion includes bass, snare and military drums, gongs, cymbals, tambourines, celestas, glockenspiels, tubular chimes, xylophones, castanets, and triangles.
[b] Percussion includes triangles, bells, cymbals, castanets, and xylophones.



a. Symphony orchestra arrangement



b. Small orchestra arrangement



c. Small dance band arrangement

FIGURE 18.40. Arrangement of players for (a) symphony orchestra, (b) small orchestra, and (c) small dance band.

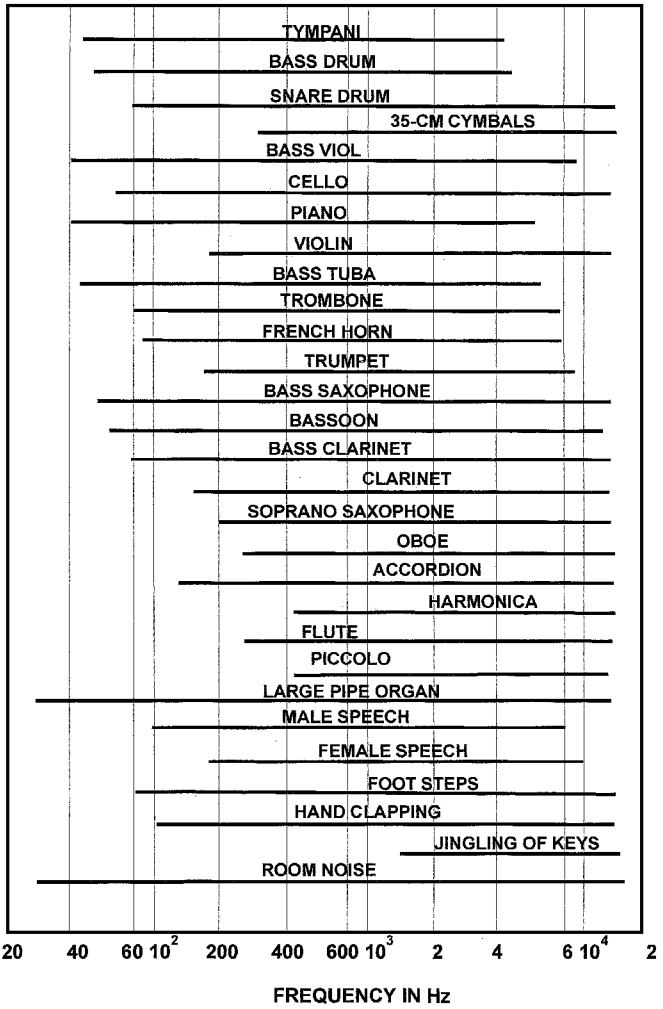


FIGURE 18.41. The fundamental frequency ranges of various musical instruments, including the human voice.

The frequency ranges of the fundamental frequencies of various musical instruments, including the human singing voices, are displayed in Figure 18.41.

References

Ågren, C. H. and Stetson, K. A. 1972. Measuring the resonances of treble viol plates by hologram interferometry and designing an improved instrument. *Journal of the Acoustical Society of America* 51:1971–1983.

Arnold, Denis (ed.). 1984. *The New Oxford Companion to Music.* New York: Oxford University Press, pp. 1925–1933.

Cremer, Lothar. 1984. *The Physics of the Violin*. Translated from the German by J. S. Allen. Cambridge, MA: MIT Press. (Originally published as *Physik der Geige*, Stuttgart: S. Hirzel Verlag, 1981.)

Computer Music Journal 16, 4. 1992. (A special issue on physical modeling. Of interest are articles by Woodhouse, Keefe, and Smith.)

Fletcher, Neville H. and Rossing, Thomas D. 1998. *The Physics of Musical Instruments*, 2nd ed. New York: Springer-Verlag. (A modern classic in its own right, it is probably the best scientific text devoted to the physics of nearly the entire gamut of musical instruments, with some interesting insights into musical history.)

Hartmann, William Morris. 1997. Electronic and Computer Music. In: *Encyclopedia of Acoustics*, Crocker, Malcolm J. (ed.), Vol. 4, Chapter 138, pp. 1679–1685.

Helmholtz, Herman L. F. 1954. *On the Sensations of Tone*, 4th ed. Translated by A. J. Ellis. New York: Dover.

Hutchins, Carleen M. (ed.). 1967. Founding a family of fiddles. *Physics Today* 20: 23–27.

Hutchins, Carleen M. (ed.). 1975. *Musical Acoustics, Part I. Violin Family Components*. Stroudsburg, PA: Dowden, Hutchinson and Ross.

Hutchins, Carleen M. 1976. *Musical Acoustics, Part II. Violin Family Functions*. Stroudsburg, PA: Dowden, Hutchinson and Ross.

Hutchins, Carleen M. 1981. The acoustics of violin plates. *Scientific America* 245(4):170–186.

Hutchins, Carleen M. 1983. A history of violin research. *Journal of the Acoustical Society of America* 73:1421–1440.

Hutchins, C. M., Stetson, K. A., and Taylor, P. A. Clarification of 'free plate tap tones' by holographic interferometry. *Journal of the Catgut Society* 16:12–23.

International MIDI Association. 1983. *MIDI Musical Instrument Digital Interface Specification 1.0.* Sun Valley, CA.

Jeans, James. 1968. *Science and Music*. New York: Dover Publications. Originally published by Cambridge University Press in 1937. (A clear exposition on the fundamentals of musical acoustics and music itself.)

Machlis, Joseph and Forney, Kristine (contributor). 1995. *The Enjoyment of Music: An Introduction to Perceptive Listening*, 7th ed. New York: Norton and Company. (The standard text in music appreciation courses in universities and colleges worldwide, it contains a most fascinating history of the evolution of music to the modern times and an overview of the technical aspects of music and musical instruments.)

Mathews, Max V., Miller, J.V., and David, E. E. Jr. 1961. Pitch synchronous analysis of voiced Sounds. *Journal of the Acoustical Society of America* 33:1725–1736. (The pioneering paper on digitization of audio signals.)

Mathews, Max V. 1969. *The Technology of Computer Music*. Cambridge, MA: MIT Press. (Written by a pioneer in the field.)

Meyer, Jürgen. 1975. Akustische Untersuchungen zur Klangqualität von Geigen. *Instrumentbau* 29(2):2–8.

Meyer, Jürgen. 1985. The tonal quality of violins. *Proceedings SMAC* 83. Stockholm: Royal Swedish Academy of Music.

Moog, R. V. 1965. *Journal of the Audio Engineering Society* 13:200–206. (By the man who started it all—the father of the Moog synthesizer.)

Olson, Harry F. 1967. *Music, Physics and Engineering*, 2nd ed. New York: Dover Publications. (A classic in the field of musical acoustics written by a major researcher in the field of sound reproduction.)

Raman, C. V. 1918. On the mechanical theory of the vibrations of bowed strings and of musical instruments of the violin family, with experimental verification of the results. *Indian Association for the Cultivation of Science Bulletin* 15:1–18. Excerpted 1975 in *Musical Acoustics, Part II. Violin Family Functions*. C. M. Hutchins (ed.). Stroudsburg, PA: Dowden, Hutchinson and Ross.

Risset, J. C. and Mathews, M. V. 1965. *Journal of the Audio Engineering Society* 13:200–206.

Roederer, Juan G. 1995. *The Physics and Psychophysics of Music, An Introduction*, 3rd ed. New York: Springer-Verlag. (An interesting combined approach to the physical processes of producing music and human perception of music. While intended for music theorists, composers, performers, music psychologists, and therapists, this text includes some materials on recent findings that may prove useful to acousticians, psychoacousticians, audiologists, and neuropsychologists.)

Rossing, T. D. 1990. *The Science of Sound*. New York: Addison Wesley. Chapters 27–29.

Rossing, Thomas D. and Fletcher, Neville H. 2004. *Principles of Vibration and Sound*, 2nd ed. New York: Springer Verlag. (An excellent text that concentrates on the physics underlying the acoustical and vibrational aspects of musical instruments and includes chapters on underwater acoustics, noise, architectural acoustics, and nonlinear acoustics.)

Sadie, Stanley (ed.). 1984. *The New Grove Dictionary of Musical Instruments*, Vols. 1 and 3. New York: Macmillan Press Grove's Dictionary of Music, pp. 8–22, 736–814. (*Dictionary* is really an encyclopedia dealing with music, music history, musical instruments, and performance techniques. A real treasure trove of information.)

Savart, F. 1840. *Des instruments de musique*. Translated by D. H. Fletcher, in Hutchins (1976):15–18.

Schelleng, John C. 1974. The physics of the bowed string. *Scientific American*: pp. 87–95.

Strong, W. J. and Plitnik, G. R. 1983. *Music, Speech and High Fidelity*. Salt Lake City: Soundprint.

Young, Robert W. 1939. *Journal of the Acoustical Society of America* 11(1):134.

Problems for Chapter 18

1. Why does a middle C from an oboe and a piano sound different, even though it is the same note?
2. Although the piano is classified as a string instrument, in what way it acts as a percussion instrument?
3. What is the difference in the manner a piano and a harpsichord sound a note?
4. What acoustical theory (e.g., vibrating bar) would you apply to the study of
   (a) piano
   (b) bass drum
   (c) xylophone
   (d) cello?

5. Give examples of the use of the Helmholtz resonance principle in the design of music instruments.
6. Why is it necessary to use rosin on bows used for playing a member of the violin family?
7. Why does a cello produce lower notes than, say, a violin or a viola?
8. What part of the human ear can be compared to a music instrument?
9. Why does a grand piano sound "grander" than a spinet?
10. Give examples of tunable and nontunable music instruments.
11. Why does a concertmaster sound the A-note prior to the commencement of a concert program?
12. What is the role of a conductor in directing an orchestra?

19
Sound Reproduction

19.1 Historical Overview

Until 1877 when Thomas Edison (1847–1931) developed the *phonograph*, there was no way to record and reproduce sounds. While working to improve the efficiency of a telegraph transmitter, Edison noted that the noise emanating from that type of machine resembled spoken words when operating at high speed. This caused him to wonder if he could record a telephone message. He began experimenting with the diaphragm of a telephone receiver by attaching a needle to it. He reasoned that the needle could prick a paper tape to record a message. He continued to experiment and he tried a stylus on a tinfoil cylinder, which to his surprise, played back the short message he had recorded, "Mary had a little lamb." The word *phonograph* was Edison's trade name for his device that played cylinders rather than disks.

Edison's phonograph was followed by Alexander Graham Bell's (1847–1922) gramophone that played a wax cylinder that could be replayed many times, but the program content had to be recorded separately for each copy. No mass reproduction of the same music or sounds was possible.

In 1877, a German immigrant Emile Berliner (1851–1929) working in Washington, DC, invented a system of recording that could be used over and over again. Berliner switched from a cylinder-type medium to a flat disk. He patented the gramophone (the true precursor of the modern phonograph), enhanced by a spring motor developed by Elridge Johnson (1867–1945), which allowed the turntable to revolve at a steady speed without the need for hand cranking of the gramophone. Berliner also invented the carbon microphone that became part of the first Bell telephones and founded the Gramophone Company to mass-produce his sound disks and his gramophone for playing them. He made two smart marketing moves: he persuaded popular artists, among them Enrico Caruso and Dame Nellie Melba, to record their music; and in 1908 he used Francis Barraud's painting of "His Master's Voice" as the company logo.

Until the late 1920s, motion pictures were silent except for the music accompaniment provided by the theater management in the form of a piano player or live orchestra. All this changed in 1926 when the Warner Brothers (Jack, Harry,

Albert, and Sam) in collaboration with Western Electric introduced a new sound on disk system that worked in synchronization with the film. In order to exhibit this new technology, the Warner Brothers Studios released *Don Juan* that proved to be a box office hit, but many studios still refused to adapt to talking picture technology. However, in October 1927, the premiere of *The Jazz Singer* starring Al Jolson really triggered the talking picture revolution. Even though *The Jazz Singer* was not the first movie to use sound, it was the first film to use spoken dialog and music as part of the action. The advent of cinema sound was furthered by the introduction of photographic sound tracks that vary in transparency and improvements in playback equipment.

The genesis of the tape recorder occurred when a German chemist Fritz Pfleumer received a patent in Germany in 1928 for application of magnetic powders to strips of paper or film. Three years later he and the German company AEG began to construct the first magnetic tape recorder, and in 1935 the first public demonstration of the BASF/AEG "Magnetophone" was given at the Berlin Radio Fair. The following year, the first BASF/AEG tape recording of a live concert was made, with Sir Thomas Beecham conducting. In the United States, Marvin Camras developed independently the wire recorder in 1939 at the Armour Foundation. The invention was sold to the military during World War II, and wire recorders were popular with amateurs until the late 1950s. In 1945, after the German surrender, U.S. Signal Corps Captain John Mullin found Magnetophones at Radio Frankfurt and 100-m reels of 6.5-mm ferric-coated BASF tape with 20-min capacity per reel. He mailed home two machines with 50 reels of tape and worked on them to improve the electronics. Alexander M. Poniatoff (1892–1980) learned about them and began work on developing a U.S.-made magnetic tape recorder. In 1948, his first model, the Ampex 200, was used to record a Bing Crosby radio show on 3M Scotch 111 gamma ferric oxide-coated acetate tape.

Since then, the magnetic tape recorder has evolved into other formats, such as 8-track tape (now defunct), DAT, and the cassette. While the cassette is still in use, it has given way to the compact disk (CD) and, later on, the digital versatile disk (originally called digital video disk and abbreviated as DVD). The CDs and DVDs also come in recordable formats, thus enabling the average person who has a personal computer and/or recording/playback equipment to operate a recording studio in the comfort of his/her home.

The term *high fidelity* refers to sound recording and reproduction that result in low harmonic and intermodulation distortions and a frequency response covering most, if not all, of the entire audio range of 20 Hz–20 kHz. A monophonic, stereo, or a multichannel system cannot be termed high fidelity unless it reproduces sound faithfully. The high fidelity (or *audiophile*, using the more modern terminology) industry got under way when Avery Fisher (1906–1994) introduced high-quality equipment in the late 1930s. After World War II, the industry truly began to flourish with Sidney Harmon (1920–) developing the first high-fidelity receiver; Sony co-founders Akio Moria (1921–1999) and Masara Ibuka (1908–1997) introducing consumer-type tape recorders, James B. Lansing (1902–1949) building quality speaker systems, and Saul Marantz (1902–1997) hand constructing quality amplifiers and preamplifiers (that remain classics to this day).

In the following sections of this chapter, we outline the principles of sound recording and playback, concluding with a prognosis of future developments.

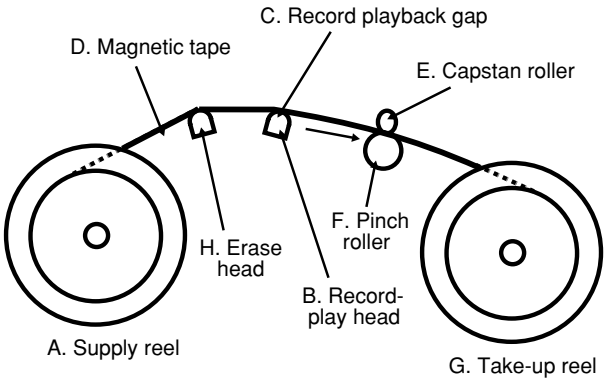
19.2 Recording Equipment

Recording of sound occurs in different ways and it can be defined as the storing of information of how the amplitude of sound varies with time. The stored information can be retrieved by playback at a later time through control of a source of sound waves. The twentieth century has seen the evolution of the phonograph cylinder into the 78-rpm recordings which in turn gave way to the 45 rpm and 33-1/3 rpm vinyl disks. These mechanical types of recordings use wavy grooves on a surface that guide a stylus to replicate the signals that were recorded. Reel-to-reel tape recorders have been used since the 1940s to make recordings, and many great performances have been archived on this medium. These master tapes are used to make vinyl disks and eventually CDs. While tapes carry the advantage of being relatively easy to edit, they do lack the random accessibility of any type of disks. Tapes solely for playback are now principally in the compact cassette format.

Magnetic Recording

Figure 19.1 shows a fundamental record-play system using magnetic recording. A close-up of the interior construction of the magnetic head is also included in the figure. Tape D is made of a smooth, durable plastic such as polyester (Mylar) and is coated on one side with a magnetizable material, most commonly a dried suspension of acicular gamma ferrite particles about 0.25 μm long in a lacquer binder.[1] The thickness of the magnetic layer is of the order of 12 μm or less. The tape unreels from supply reel A and threads over a magnetic head B that supplies a magnetic field at recording gap C. The magnetic field varies in its strength according to the sound being recorded. The now magnetized tape is pulled past the head by capstan roller E and pinch roller F and spooled onto a take-up reel G. In rewinding the tape, the recording head is deenergized. For playback, the tape leaves the supply reel, as before, but the winding on head B is connected to an amplifier that, in turn, connects to a loudspeaker. The magnetized elements induce a voltage in the head winding, which translates into a replication of the gap field variations during recording. The recording may be erased by an additional head H that produces a strong, steady AC field. It is not necessary to erase the tape if it is being recorded over, because the erase head is automatically energized when recording, thus removing previously recorded material.

[1] After the tape has been coated and while the suspension is still damp, it is subjected to a magnetic field that aligns the magnetic particles in order to increase recording signal strength and minimize noise. Other alternative magnetic materials include chromium dioxide and cobalt-coated gamma ferrite. It is obvious that tapes have to be manufactured under stringent clean-room conditions because the slightest contaminants can cause impaired recordings.



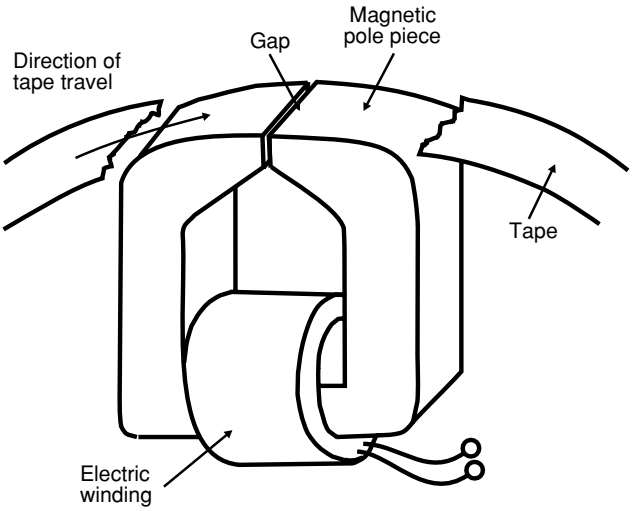


FIGURE 19.1. The elements of a magnetic tape recording and playback system, with details of the interior construction of a magnetic head.

A magnetic head is designed to concentrate the magnetic field in the possible smallest region of tape, thus giving a high-recording density of maximum number of wavelengths possible. To achieve these high densities, the head gap must be made very small, in the order of 1 or 2 μm, which yields low output voltage and requires very smooth tape surface and good contact between the head and the tape. The head cores are usually fabricated of Permalloy (80% Ni, 20% Fe) for high permeability and low-magnetic retention. In more recent times ceramic-like magnets have been used for better wear and they do not require lamination.

Tape speeds have been standardized at 30, 15, 7.5, 3.75, and 1.875 in./s for analog recording. All of these speeds are much lower than that required for digital recording, which necessitates bandwidths in the megahertz range. Digital audio recorders make use of the video recording techniques that entail rotating heads. These heads scan the tape at 228 in./s while the tape itself moves at only 0.66 in./s. The highest speed is used for master recordings, in situations where the highest quality is demanded in covering the 20 Hz–20 kHz range. Cassettes use the 1.875 in./s speed.

The Dolby® systems for noise reduction employ circuitry that pre-emphasizes high frequencies before they are recorded on tape in order to make them louder than the tape hiss with which they compete. The circuit for recording tape is amplitude sensitive with the result that only soft, high-frequency sounds are emphasized. Emphasis of loud, high-frequency sounds might drive the tape into its distortion levels. Upon playback, a matching de-emphasis circuit is employed to restore the high frequencies to their proper balance with the other parts of the recorded signal.

Digital Recording

The capabilities of analog recording pale in comparison with the advantages of digital recording. Zero wow and flutter, more than 90 dB signal-to-noise ratio, and incredibly low-distortion levels are the most outstanding attributes of digital recording. To achieve digital recording, the analog signal is sampled at regular intervals. Sampling of the amplitude is done with almost absolute accuracy (one part per 10^9 in a 96-dB signal-to-noise ratio system). The amplitude is recorded as a number, say, 59,959,498. The sampling should be done at more than twice the Nyquist number (the highest frequency in the audio signal). For a signal with a maximum of 22,000 Hz, a rate of 44,000 samplings per second is quite practicable ($20{,}000 \times 2 + 10\%$). Because the series of digital samples occur in the megahertz range, they are recorded and played back on videotape recorders. The audio program in digital form can be processed with digital features, error correctors, and other techniques that yield results that cannot be achieved with analog means. The numbers comprising the signal can be stored successively in buffers and read out in perfect crystal-controlled time intervals, even if there exist erratic mechanical fluctuations in the tape drive. Thus, wow and flutter are eliminated. The digital signals are restored to analog format through digital/analog (D/A) converters.

Experiments on the digitization of sound were conducted during the late 1950s and early 1960s for the purpose of computer analysis, speech synthesis, simulation of music, and simulation of reverberation, mostly at the Bell Telephone Laboratories in Murray Hill, New Jersey (Mathews et al., 1961). In the early 1970s, commercial digital recordings were commonly used by recording studios for master recordings, but the results were distributed by analog means.

A significant breakthrough occurred in the early 1980s, when Sony Corporation of Japan and Koninklijke Philips Electronics N.V. of the Netherlands came out with the compact disk (CD) that quickly replaced the analog long-play (LP) record and the cassette as the most common medium for playback of recorded music. Later

on, machines for digital recording on cassette videotape and then the rotary-head digital audio tape (R-DAT) came in the market. In the past few years, the digital compact cassette (DCC developed by Philips-Matsushita) and the mini-disk (MD) by Sony arrived on the recording scene, but they have not gained much in the way of general acceptance.

The DVD is a multimedia device for playback through video monitors as well as through audio channels. A DVD player can also function as an audio-only playback unit for compact disks, but CD players cannot play the audio portion of DVDs. A standard CD has a capacity of about 74 min of standard CD audio music. There are extended CDs that can actually exceed this limit and pack more than 80 min on a disk, but these are nonstandard. Regular CD-ROM media hold about 650 MB of data, but the actual storage capacity depends on the particular CD format used. The DVD uses smaller tracks, 0.74 μm versus 1.6 μm for the CD. Thus, DVD technology writes in smaller "pits" to the recordable media than is the case with CD. Smaller pits require that the drive's laser produce a smaller scanning spot. This is achieved by reducing the wavelength of the laser from 780 nm infrared light used in standard CD drives to 625 nm–650 nm red light. This helps to yield a storage capacity of 4.7 GB. Smaller data pits allow more pits per data track. The minimum pit length of a single-layer DVD-RAM is 0.4 μm as compared to 0.834 μm for a CD.

The various DVD formats are as follows:

1. DVD-R: The most universal of recordable DVD formats used by DVD burners and many DVRs (digital video records). DVD-R is a write-once format, much like CD-R, and disks made in this format can be played in most current DVD players.
2. DVD-RW: Recordable and rewriteable format (like CD-RW). Disks are playable in most DVD players, provided they are recorded in the straight video mode and finalized.
3. DVD+RW: Recordable and rewriteable format. This format is claimed to provide a greater degree of compatibility in current DVD technology than DVD-RW.
4. DVD+R: A record-once format that is claimed to be easier to use than DVD-R while still playable in most DVD players.
5. DVD-RAM: Recordable and rewriteable format that is not compatible with current DVD technology.
6. DVD-Audio: Intended to play in audio playback players (such as in-car players), it can offer slideshows and texts that can be displayed. The bit rate of this format is 9.6 Mb/s as compared to 6.14 Mb/s of a video DVD. DVD Audio titles being released include a DVD-Video compatible zone using Dolby or DTS coding. This means that the DVD-Audio is playable in tens of millions existing DVD players, but not with the maximum quality that a DVD-Audio player affords.

There are three optical formats that offer more than 8.5 GB storage. These are Plasmon's UDO (intended for business data storage), HD-DVD, and Blu-ray. All three systems use blue laser light that affords a narrower beam than red laser

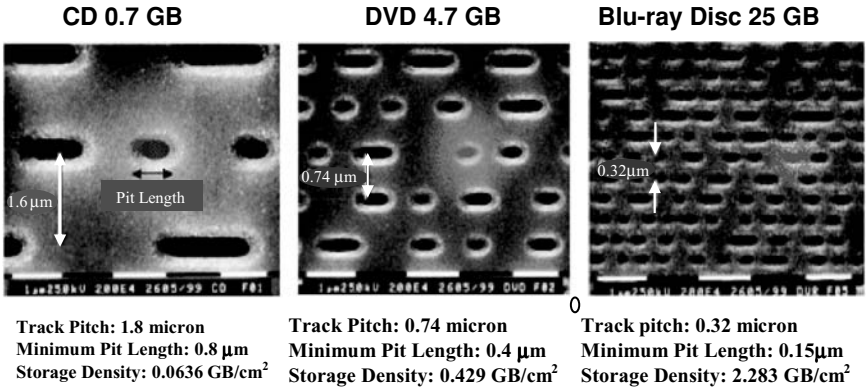


FIGURE 19.2. Data layers in 0.7 GB CD disk, 4.7 GB DVD disk, and 25 GB Blu-ray disk. Note the pit lengths generally decrease with increasing disk capacities.

light, hence yielding smaller data areas and more densely packed data delivering high capacity. The Blu-ray is a Sony development that holds 25 GB in single-layer format, but will have two layers and a double-sided recording holding as much as 100 GB. Such capacities can hold high-definition movies and in the audio mode could presumably easily contain the entire Wagner's *Ring Cycle.* It needs a hard coating to protect it and requires a special player. The HD-DVD, also intended for playback of high-definition movies, features a 20-GB capacity but could possibly be expanded to 50 GB via multiple layers. The HD-DVD does not require a protective cartridge or coating but it also does require a special player. Figure 19.2 compares the data layers of the 0.7-GB CD, the 4.7-GB DVD, and the 25-GB Blu-ray disks. As of this writing, there is a formidable conflict raging as to which of these two high-definition formats will gain ascendancy. Some movie studios and developers (NEC and Toshiba) have advocated HD-DVD; other studios, major computers manufacturers (Dell and Hewlett Packard), and the Blu-ray Disc Founders (Sony, Philips, LG, Matsushita, Pioneer, Samsung, Sharp, and Thomson) are boosting Blu-ray.

Voice Recognition

Voice or speech recognition is the ability of a machine or device to receive and interpret dictation or to comprehend and carry out oral commands. In use with computers, analog audio must be converted into digital signals through an analog-to-digital (A/D) converter. In order that it can decipher the signal, a computer must have a digital database, or vocabulary, of words or syllables, and a rapid means of comparing this data with signals in the format of speech patterns stored in the hard drive and loaded into memory when the program is operating. A comparator compares these stored patterns against the output of the A/D converter. Thus, the program's vocabulary constitutes a recording of sorts.

It is fairly obvious that the size of a voice-recognition program's effective vocabulary and its speed of execution directly relates to the random access memory (RAM) capacity of the computer in which it is installed. A voice-recognition program runs many times quicker if the entire vocabulary can be loaded into the RAM instead of searching the hard drive for some of the matches. Processing speed is also critical, because it affects how rapidly the computer can search the RAM for matches.

In the present state of the art, all voice-recognition programs make errors. Background noise, such as barking dogs, loud conversation, and noisy children, can produce false input. There is also a problem with words that sound alike but are spelled differently and have different meanings—for example, "know" and "no" or "dear" and "deer." This problem may someday be solved by using stored contextual information, but this requires more RAM and even faster processors than are currently available in personal computers.

Voice recognition programs are now applied to dictation in conjunction with word processing programs and in voice-response customer servicing on the telephone.

19.3 Playback Audio Equipment

Figure 19.3 shows a schematic of a fairly complete playback system for multichannel sound reproduction, in this case a "5.1 surround sound" system. Whether the system is monophonic (single-channel), stereo (two-channel), or multichannel (more than two channels[2]), there are three stages to sound reproduction: the first stage consists of the source or sources of program material; the second stage is the preamplification and amplification. The third stage consists of converting the electrical signals into acoustic signals, and this occurs in loudspeakers or headphones.

Two-channel (stereo) setups reproduce the spatiality of the original program. When properly recorded and reproduced, the strings located on one side of the orchestra get reproduced more strongly in the same side of loudspeaker than in the loudspeaker on the other side, giving the listener the illusion of strings' location. Multichanneling is intended to reproduce not only the spatiality of the program but also provide some degree of the acoustic ambience of the hall where the program was recorded or to reproduce the effects of multichannel "surround sound." The acoustics of the listening room, of course, affects the reproduced sound, which may even assume additional characteristics that may not be so desirable.

---

[2] A 5.1-multichannel system consists of right, left, and a center-channel speaker in the front and a right- and a left-rear speakers, plus a subwoofer (the ".1" of the 5.1 designation) that theoretically could be placed anywhere in the room. A 7.1-channel system contains the five sets of speakers and the subwoofer of the 5.1-system plus two more sets of speakers placed to the right and left in the middle of the room. In either case, additional amplifiers are required for each of the additional speakers. A multichannel system also would normally incorporate an audio encoding/decoding system, such as the Dolby Digital®, Dolby Digital EX®, or Dolby Digital Surround EX® to deliver discrete multichannel audio in an optimal fashion to the appropriate channels.

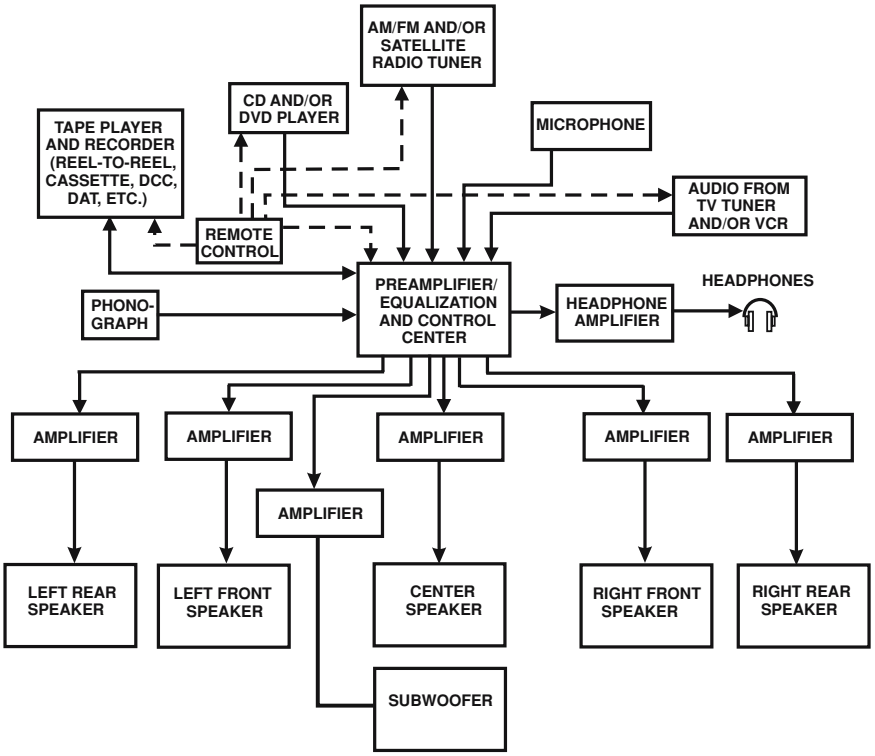


FIGURE 19.3. Playback system for multichannel sound reproduction.

The sources constituting the first stage can include one or more of the following: microphone, AM/FM and/or satellite radio tuner, phonograph (consisting of a turntable, arm, and cartridge[3]), reel-to-reel or cassette player, compact disk (CD) player, digital versatile disk (DVD) player, and the audio output of a television receiver. The second stage consists of a preamplifier that provides the means to select signal sources, the proper equalization for program sources, and means of controlling the volume of the program material. After the initial preamplification stage, the program material is amplified in the amplifier. An amplifier may be multichanneled in that there is a separate amplifier for each channel, and a number of these amplifiers may be mounted on a single chassis.[4] An *integrated amplifier*

[3] Magnetic cartridges convert stylus movements into electrical signals. Ceramic cartridges are essentially piezoelectric converters that yield stronger electrical signals that need less preamplification but do not have the range and subtlety of magnetic cartridges.

[4] In extremely elaborate (and expensive!) systems, an electronic active crossover system may be inserted between the preamplifier and the amplifier, and the customary passive crossover system built inside the loudspeaker system is bypassed. Each channel's input into the electronic crossover separates into appropriate frequency bands, which are fed into separate amplifiers. Each of the amplifiers links directly to the individual drivers. A single channel would require three amplifiers, one to feed the woofer, another to feed the midrange, and the third to feed the tweeter.

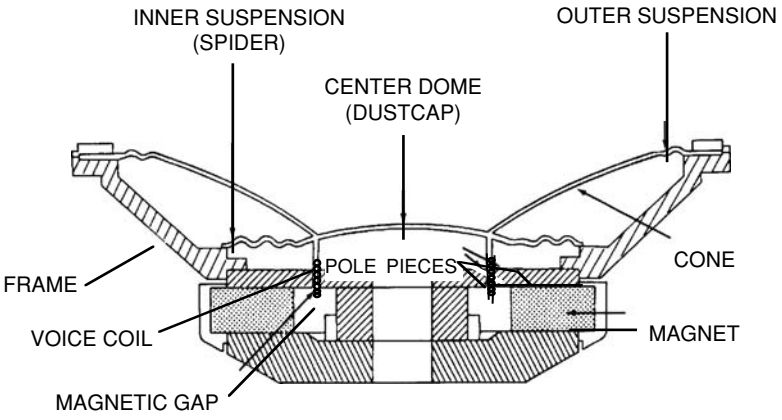


FIGURE 19.4. Elements of an electrodynamic speaker. (Courtesy of JBL Professional.)

combines on a single chassis the preamplifier (which acts as a controller) and the amplifier. A stand-alone amplifier is referred to as a *power amplifier* and it functions in conjunction with a separate preamplifier. A chassis that combines a radio tuner with a preamplifier and a power amplifier is called a *receiver*. This type of construction makes for more economical production, with a single-power supply serving all of these sub-components, and conservation of rack or shelf space; its principal disadvantage is potentially increased heat output from the electronics and the lack of flexibility in upgrading individual sub-components.

The third stage consists of loudspeakers and or headphones. A tremendous variety of loudspeakers are available in the market. The most common type is the electrodynamic speaker, which can range from a single-cone type to more elaborate multidriver units consisting of woofers to reproduce low-frequency sounds, midrange-drivers to handle the frequencies between the low frequencies produced by the woofers and the high frequencies produced by tweeters. A two-way loudspeaker system divides its program material between a woofer and a tweeter. A three-way loudspeaker system incorporates mid-range drivers, and four and five-way units have also been constructed. In order that the individual drivers receive the proper frequencies to the exclusion of the program contents outside of their respective optimal operating ranges, special types of bandpass filters, or *crossover networks*, are employed to separate out the high frequencies from the signals being fed into the woofers, to band-pass the mid-frequencies into the mid-range drives, and to channel only the high-frequency portion of the signals into the tweeters. Because much of the acoustical energy is contained within the low-frequency portion of the signals, passage of unfiltered signals into the tweeters can destroy these drivers.

Figure 19.4 shows the construction of an electrodynamic driver. Three separate but interrelated subsystems constitute the driver. The motor system consists of the magnet, pole piece, front plate, and voice coil. The diaphragm, generally a

cone and a dust cap or a one-piece dome, constitutes the second subsystem; the suspension system, which includes the spider and the surround, constitutes the remaining subsystem.

In the motor assembly, the back and front plates and the pole pieces are made from a highly permeable material, such as iron, which provides a path for the magnetic field of the ring-shaped magnet that is usually constructed of ceramic–ferrite material. The magnetic circuit is completed at the gap, with a strong magnetic field existing in the air space between the pole piece and the front plate. The coil, which connects to a pair of input terminals, is wound around a thin cylinder that is attached to the speaker cone that, in turn, is mounted at its outer edge through a flexible surround to the frame. A spider, essentially a movable membrane also attached to the frame, positions centrally the diaphragm at its inner edge and the voice coil. The signal containing the program material feeds into the voice coil, which generates a change in the magnetic field imposed by the permanent magnet surrounding the coil. If an alternating current is fed into the coil, the flow of the current in one direction will cause the voice coil to move in one direction, and the reverse flow will cause the coil to move in the opposite direction. The cone under the impetus of the moving voice coil acts as a piston in moving the air in front of it.

The ideal cone would act as a perfectly rigid piston pushing against the air. The transfer of the motion from the piston to the air is bound in terms of frequency by resonance frequency of the cone at the low end (here the ability to transfer energy to the air is limited by mechanical constraints) and by the radiation impedance at the upper limit. This upper frequency limit occurs from the fact that it is a function of both the nature of radiation impedance of the air and the radius of the radiating surface. Smaller radiating surfaces can reproduce higher frequencies more effectively than larger surfaces, which accounts for the smaller sizes of tweeters. Real-world cones, however, are not perfectly rigid and will flex depending on the traits of the materials they are constructed from. Cone flexure has a critical effect on the high-frequency efficiency, the sound–pressure level output, and driver–polar response. Driver materials may differ in degrees of stiffness and transmit vibrations at different speeds internally, but they tend to produce the same sort of flexures or modes. In Chapter 6, these modes have been discussed for circular membranes fixed at their outer edges.

The most important function of the speaker enclosure is to control transmission of the driver's rear-radiated sound energy in order to avoid its mutual cancellation with forward-radiated energy at low frequencies. The enclosure also acoustically "loads" the driver by providing a suitable acoustic impedance to match the characteristics of the driver and the requirements of the speaker system with regard to low frequency and large-signal performance. The enclosure must be made sufficiently rigid so that the cabinet vibrations and resonances do not add appreciably to the program material. Two major classes of electrodynamic-driver enclosures are (1) the infinite baffle/closed box in which the sole radiation source is the driver diaphragm and (2) the bass reflex that is vented to augment the driver's radiation at low frequencies. The bass reflex may be a vented system that incorporates a

tuned aperture of a specified cross-sectional area and port length in which the enclosed air mass resonates with the enclosure's air spring. This causes a woofer's back wave to communicate with the external acoustic space with the enclosure effectively acting as a Helmholtz resonator.

Another version of the bass reflex system is the port/passive radiator (PR). The vent is replaced with an acoustically driven diaphragm (aptly called a *drone cone*). Another type of enclosure is the transmission line (TL) that effectively functions as a low-pass filter with a 90° phase shift, absorbing all of the rear wave energy of the woofer except for frequencies below 75 Hz. The TL enclosure provides a folded path or a labyrinth equal in length to the 25% of wavelength at or just above the resonance of the woofer. Damping material such as Dacron®, fiber wool, or fiberglass fills much of the labyrinth. The TL enclosures generally provide excellent, clean bass response from relatively compact floor-standing cabinets. The principal disadvantage of TL is the complexity of the cabinet structure.

Other types of speaker systems are available in the market. Among these are the electrostatic speakers, the ribbon-type speaker system, and there are others that consist of hybrids, e.g., a system that combines the electrodynamic drivers with an electrostatic or ribbon tweeter. A ribbon type of drive essentially consists of a voice coil patterned (like the conductive pattern on a printed circuit board) onto a magnetizable diaphragm (i.e., the "ribbon") made of strong, flexible materials such as Mylar. The diaphragm is suspended in the front of a flat magnet that matches the area of the diaphragm. A varying signal sent into the coil causes a change in the magnetic field that causes the ribbon to move with respect to the magnet, thus causing sound to be radiated. If the magnet is perforated in one way or the other, the ribbon acts as a dipole sound generator.

In the electrostatic speaker (which is similar operationally to the condenser microphone but functioning in reverse), a voltage is maintained between a thin diaphragm and another surface. In effect the two thinly separated surfaces are acting as a large condenser, with the varying voltages fed to one of the elements, causing the diaphragm to move. Because of the small excursions of the diaphragm, the electrostatic driver does not function well in the low-frequency range but it is capable of providing excellent quality high-frequency signals. The highly capacitive loading of the electrostatic speaker may be problematic for some power amplifiers that are usually designed to handle primarily resistive impedances.

Headphones are essentially miniaturized speakers mounted in earcups, which are in turn attached to headbands that serve the purpose of holding these drivers against the ears. There are also a number of headphone models that use electrostatic elements either by themselves or as supplemental components to electrodynamic elements.

19.4 The Iosono System

On the basis of research conducted for more than 15 years at the Technical University in Delft, the Netherlands, Karlheinz Brandenburg, the director of Fraunhofer

Institute of Digital Media in Germany who is known as the "father of MP3 described in the next section, developed a spatial sound technology called Iosono®. It is intended to carry sound reproduction beyond the 5.1- and 7.1-channel systems and is based on the principle that sound waves can be reproduced using secondary sources at the perimeter of the original sound field. This process is called wave field synthesis (WFS).

In the Iosono system, speaker arrays ring the listening area and function in a coordinated, phased fashion to reproduce each individual sound wave. An example of an acoustical illusion that can be created is that of a helicopter that slowly approaches the audience, flies through the middle of the theater, and disappears into the distance. The audience hears the helicopter noise that would be generated if the helicopter were actually flying this path. It is also claimed for this system, aimed at cinemas and eventually the home theater, that the whole room can act as a listening "sweet spot."

An Iosono system consists of a continuous ring of speakers configured in panels of 8 two-way speakers, mounted on the peripheral walls of the space and connected via fiber-optic cables to a central Iosono processing unit. The number of speaker panels used depends on the size of the space. In February 2003, a 100-seat movie theater in Ilmenau, Germany, received the first commercial Iosono Cinema system. The installation used 198 speaker systems. Because of the fact that commercial movie sound has yet to be mixed, the Iosono system is also capable of playback of other formats, such as 5.1, DTS, SDDS as well as stereo. However, the audience noticed the enlargement of "sweet spot" and was treated to a 90-s trailer that demonstrated the effect of a full Iosono system.

19.5 Portable Audio Playback Equipment

In this section reference is made to handheld devices, not to playback units that are transportable but too large to be held conveniently. The first units of this type was the Sony Walkman®, a cassette player, which was to be joined by competing models, and subsequently the Sony Discman® that played compact disks. The newer units we describe here can download music from the Internet as well as from other sources and the storage is usually done on rewriteable nonvolatile memories.

MP3

MP3, developed by Germany's Frauenhofer research institute, denotes MPEG Audio Layer 3. It is an audio compression technology that constitutes part of the MPEG-1 and MPEG-2 specifications. The company Thomson Media patented this technology in the United States and in Germany.

Uncompressed audio, such as that found on audio compact disks store more data than human hearing can process. Music on CDs has a bandwidth of 1.4 MB/s. This means that 1 min of music on a CD takes up to 10 MB of data. Through the

use of MP3 compression, this bitstream is greatly reduced by a factor of 8–12. A typical MP3 file will require 0.828 MB/s, and hence 1 min of music is reduced from 10 MB to about only 1 MB. Even greater compression is possible for use on the Internet, for example, but at the expense of a decrease in sound quality.

MP3 uses two compression techniques to achieve size reduction from uncompressed audio: one lossy and the other lossless.

1. The process begins by breaking the signal into smaller components called frames, each frame typically lasting a fraction of a second. The signal is analyzed to determine its spectral energy distribution. Because different portions of the frequency spectrum are most efficiently encoded via slight variant of the same algorithm, the signal can be broken into sub-bands, which can be processed independently for optimal results.
2. The encoding bitrate comes into play, and the maximum number of bits that can be allocated to each frame is calculated. For example, if the encoding is set at 128 kbps, there is an upper limit on how much data can be stored in each frame. This step establishes how much available audio data will be stored and how much will be discarded. This constitutes a lossy phase of the MP3 procedure.
3. The frequency spread for each frame is compared to mathematical models of human psychoacoustics, which are stored in the codec[5] as a reference table. From these models, it can be determined which frequencies need to be rendered accurately, because they are most perceptible to humans, and which ones can be dropped or allocated fewer bits, owing to the fact that they are less likely to be perceived by human hearing. This stage is also a lossy procedure.
4. The bitstream is run through the process of *Huffman coding* that compresses redundant information throughout the sample. The Huffman coding does not work on the basis of a psychoacoustic model but achieves additional compression through more traditional means so that even less space is required for storage. The action is quite lossless, similar to the "zipping" and "unzipping" of files on a computer.
5. The collection of frames is now assembled into a serial bitstream, with header information preceding each data frame. (The header contains extra information about the file to come, such as the name of the artist, the track tile, the name of the album from which the recording came from, the recording year, genre, and personal comments that may have been added—this is referred to as an ID3 tag).

iPod

The iPod® is Apple Computer's multiuse record/playback ultraportable unit that makes use of a mini disk drive to store data. It plays music and other audio that

[5] The word "codec," a shortening of the words *compress* and *decompress,* refers to any of a class of processes that allow for the systematic compression and decompression of data. Various codecs are fundamental to many formats and transmission methods, for example, image and video compression formats. Here we are concerned with the audio MP3 codec.

has undergone MP3, AAC, or Apple Lossless compression programs. One version weighs less than 160 g and it is available with a choice of 20 GB or 40 GB storage, resulting in a capability of storing as many as 10,000 songs. Even more compact versions with less storage and other models with more storage are emerging from Apple, and competitive units are being issued by other manufacturers. The iPod can also be used to record meetings, enter voice memos, and its speed can be adjusted to accommodate audio books. Synchronization with a home computer for down- or uploading data can be achieved via FireWire® or USB 2.0 cables. The iPod can also be used as a portable hard drive, which would allow the user to carry computer files from one place to another. It can even used to store photographs, function as an organizer with contact listings, calendars, and to-do lists; and, with the use of adapters, can be played through home and auto sound systems.

19.6 The Future of Sound Reproduction

There is no question that nanotechnology, still in its infancy, will result in more compact recording media, more powerful electronic processing units, and more realistic audiovisual imaging. Digitization of acoustic information will be refined to an even greater extent.

Loudspeakers have been the weak link in the audiophile chain of playback of recorded sound. Amplifiers and preamplifiers have reached the point of diminishing returns with respect to low distortions and noise. The principal deviation from the ideal sound reproduction occurs in transducers, hence it is the loudspeaker that is the component that needs the most improvement. It is apparent that new technology will be needed to develop new types of transducers, most likely resulting from a collaboration of the best minds in the fields of acoustics, materials science, solid state physics, electronic and mechanical sciences, and even nanotechnology. Not only loudspeaker distortion would be reduced to theoretical minimum (Raichel, 1979), but also the sound sourcing would be considerably more sharply focused to enhance the listener's feeling of actually "being there" at the concert venue where the music was recorded.

References

Ballou, Glenn (ed.). 2005. *The Handbook For Sound Engineers*, 3rd ed. New York: Focal Press. (A comprehensive work on sound engineering. Includes coverage of MIDI, cinema sound, consoles, and other topics.)

Borwick, John (ed.). 2001. *Loudspeaker and Headphone Handbook*, 3rd ed. New York: Focal Press.

Cooke, Raymond E. (ed.). 1980 (Vol. 1), 1984 (Vol. 2), 1992 (Vol. 3). *Loudspeakers*, Vols. 1–3. New York: Audio Engineering Society. (A treasure trove of technical papers written during 1980–1992 on the design, construction, and operation of loudspeakers.)

Crowhurst, Norman, 1960. *The Stereo High Fidelity Handbook*. New York: Crown Publishers. (A succinct text by one of the top writers in the field of stereo reproduction.)

Dickason, Vance. 2000. *Loudspeaker Design Cookbook*, 6th ed. Peterborough, NH: Audio Amateur Press. (An excellent text to learn about the design and construction of loudspeakers.)

Hacker, Scot. 2000. *MP3: The Definitive Guide*. Sebastopol, CA: O'Reilly Media, Chapter 2.

Holman, Tomlinson. 1999. *5.1 Surround Sound—Up and Running*. New York: Focal Press. (Written by the developer of the THX sound system.)

Olson, Harry F. 1967. *Music, Physics and Engineering*, 2nd ed. New York: Dover Publications.

Olson, Harry F. 1991 (reprint). *Acoustical Engineering*. New York: Professional Audio Journals.

Raichel, Daniel R. 1979. Minimum harmonic distortion levels in ideal loudspeakers. *Journal of Audio Engineering Society* 27(8): 492–495. (Previously presented November 7, 1977 at the 58th Convention of the Audio Engineering Society, New York, NY.)

Risset, J. C. and Mathews, M. V. 1965. *Journal of the Audio Engineering Society* 13:200–206.

Strong, W. J. and Plitnik, G. R.1983. *Music, Speech and High Fidelity*. Salt Lake City: Soundprint.

Watkinson, John. 2004. *The MPEG Handbook*. New York: Focal Press. (Probably the most definitive book on the popular MPEG technology.)

Problems for Chapter 19

1. Why was the cassette tape overtaken by the compact disk in popularity?
2. What are the principal advantages and disadvantages of data compression in sound recordings?
3. What are the advantages and disadvantages of having, say, a two-way versus a three-way versus a four-way speaker system?
4. What are the dangers of using high-powered audio amplifiers?
5. What are the key factors in determining the quality of a playback system?
6. Why would a reputable recording studio use higher speeds for taping rather than presumably more economical slower speeds?
7. Design your "dream" audio system on the basis of your market research. Explain the reasons for your choice of specific components.
8. Research loudspeaker systems in the market. What specifications would you look for in making your selection?
9. Why would an MP3 or an iPod system be objectionable to an audiophile for use in an audio system?
10. How is computer technology is being used in the recording stage and playback stage?
11. How does room acoustics affect playback of audio material? What are the incipient problems and how would you deal with them?

20
Vibration and Vibration Control

20.1 Introduction

Noise often results from vibration. Many sources of vibration exist and they include impact processes, such as blasting, pile driving, hammering, and die stamping; machinery such as motors, engines, fans, blowers, and pumps; turbulence in fluids systems; and transportation vehicles. Attenuation of vibration generally cuts down on the noise level and in many cases lengthens the service life of the machinery itself. Damping, correction of imbalances, and configuration of flow paths constitute the principal measures of cutting down on the deleterious effects of vibration.

20.2 Modeling Vibration Systems

We commence with a basic one-degree-of-freedom system of Figure 20.1, which consists of a mass, a spring, and a damper (also referred to as a *dashpot*). The system has only one degree of freedom because it is constrained to move only in the x-direction. Summing up all the forces acting on the mass and applying Newton's second law, we obtain

$$m\frac{d^2x}{dt^2} + C\frac{dx}{dt} + kx = f(t) \tag{20.1}$$

where

$$\begin{aligned} m &= \text{mass} \\ t &= \text{time} \\ C &= \text{coefficient of damping} \\ k &= \text{linear elastic constant} \end{aligned}$$

Damping occurs from energy dissipation due to hysteresis, sliding friction, fluid viscosity, and other causes. The damping force may be proportional to the velocity (as stated in the above equation) or it may even be proportional to some other power of velocity. Sliding friction is often represented as a constant force in a direction

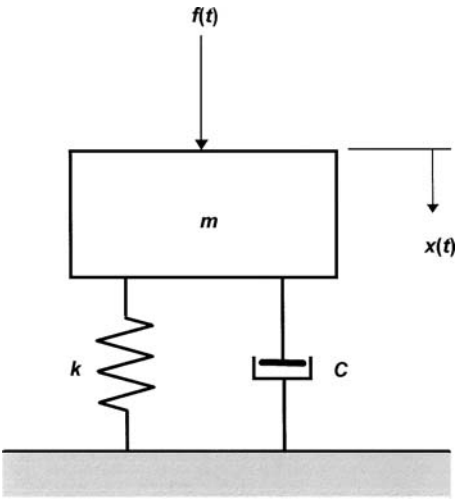


FIGURE 20.1. Model of a vibrating system with one degree of freedom.

opposing the velocity. Viscous damping proportional to and opposite in direction to that of the velocity serves as a reasonable model in many situations. Moreover, this assumption is quite amenable to mathematical treatment. When $f(t)$ is set to zero, Equation (20.1) describes the free, viscous-damped, one-degree-of-freedom system,

$$m\frac{d^2x}{dt^2} + C\frac{dx}{dt} + kx = 0 \tag{20.2}$$

We shall employ Laplace transforms to treat Equation (20.2). Dividing Equation (20.2) by mass m and assuming for the moment that the initial conditions are zero, we can write Equation (20.2) using the Laplace transform variable s, as

$$s^2X(s) + \frac{C}{m}sX(s) + \frac{k}{m}X(s) = sx(0) + \dot{x}(0) + \frac{C}{m}x(0)$$

or

$$\left(s^2 + \frac{C}{m}s + \frac{k}{m}\right)X(s) = \left(s + \frac{C}{m}\right)x(0) + \dot{x}(0) \tag{20.3}$$

Here $\dot{x} \equiv dx/dt$. Setting the parenthesized term on the left-hand side of Equation (20.3) to zero yields

$$s^2 + \frac{C}{m}s + \frac{k}{m} = 0$$

which can be rewritten in the form

$$s^2 + 2\xi\omega_n s + \omega_n^2 = 0 \tag{20.4}$$

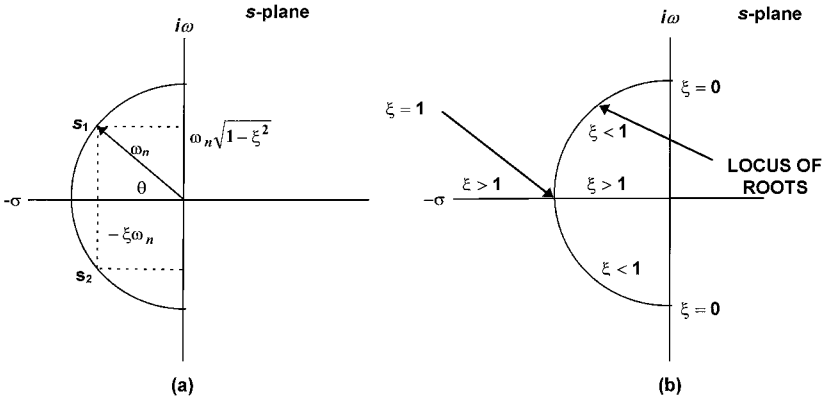


FIGURE 20.2. Root locations and nomenclature for a second-order dynamic system. The s-plane to the right shows the domain of the roots as a function of the damping ratio ξ.

where

$$2\xi\omega_n = \frac{C}{m} \qquad \omega_n^2 = \frac{k}{m} \tag{20.5}$$

Equation (20.4) is the characteristic equation of which roots essentially determine the response of the system [i.e., the position of the mass as a function of time or $x(t)$]. Solving for the roots of Equation (20.4) yields

$$s_1, s_2 = \frac{-2\xi\omega_n \pm \sqrt{(2\xi^2\omega_n)^2 - 4\omega_n^2}}{2} \tag{20.6}$$

These roots can assumed complex values and so can be plotted on a s-plane, where $s = \sigma + i\omega$ and $\omega = 2\pi f$. The general plot is shown in the complex plane of Figure 20.2(a). For this plot it is assumed that the system is stable and the value of ξ falls between zero and unity. The term ξ is the *damping ratio* and ω_n is the *undamped natural frequency*. In Figure 20.2(a) it is noted that $\xi = \cos\theta$. Both ξ and ω_n constitute the key factors that determine the roots of the characteristic equations and hence the response of the system. The following four cases of damping are of interest:

Case 1: $\xi < 1$ (*Underdamped system*)
The roots are $s_1, s_2 = -\xi\omega_n \pm i\omega\sqrt{1-\xi^2}$ and the system response is given by

$$x(t) = Ae^{-\xi\omega_n t}\cos\left(\omega_n\sqrt{1-\xi^2}\,t + \theta\right)$$

Case 2: $\xi > 1$ (*Overdamped system*)
The roots are $s_1, s_2 = -\xi\omega_n \pm i\omega\sqrt{\xi^2 - 1}$ and the system response is given by

$$x(t) = Ae^{-\left(\xi\omega_n + \omega_n\sqrt{\xi^2-1}\right)t} + Be^{-\left(\xi\omega_n - \omega_n\sqrt{\xi^2-1}\right)t}$$

Case 3: $\xi = 1$ (*Critically damped system*)
The roots are $s_1,\ s_2 = -\omega_n$ and system response is given by

$$x(t) = Ate^{-\omega_n t} + Be^{-\omega_n t}$$

Case 4: $\xi = 0$ (*Undamped system*)
The roots are $s_1,\ s_2 = \pm i\omega_n$ and the system response is given by

$$x(t) = A\cos(\omega_n t + \theta)$$

Figure 20.2(b) shows the domains of the roots in the s plane as a function of the damping ratio ξ for a constant value of ω_n.

In order for us to appreciate the influence of system damping, consider the system of Figure 20.1 which is subjected to an initial displacement x_0 and which has no external forcing function acting on it. The dependency of the response of the system on the root locations is shown in Figure 20.3 for ω_n held constant. In

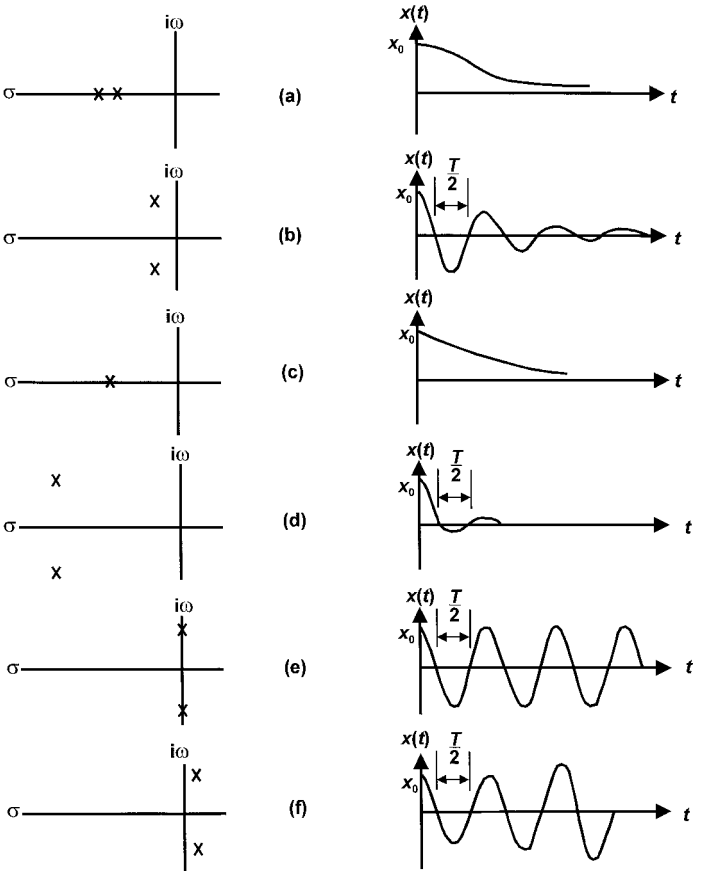


FIGURE 20.3. System responses showing the influence of root location in a second-order dynamic system.

Figure 20.3(a) the system is very sluggish and returns to the equilibrium position very slowly. In Figure 20.3(b), the system response is rapid, but it overshoots the equilibrium position and eventually the oscillation dies out due to the effect of damping. Figure 20.3(c) represents the critically damped system that promptly returns to the equilibrium position with no overshoot. This represents the dividing line between the overdamped system (Case 2) and the underdamped system (Case 1). The roots for the underdamped system occur in complex-conjugate pairs, the roots for the overdamped system are real and unequal, and the roots for the critically damped system are real and equal. Figures 20.3(b) and 20.3(d) show, respectively, the difference between a lightly damped system and a heavily damped system. The response of the more heavily damped system dies out more quickly.

If the roots lie on the $i\omega$ axis as shown in Figure 20.3(d), no damping occurs and the system will oscillate forever. If any roots appear on the right half of the s plane, then the response will increase monotonically with time and the system will become unstable, as shown in Figure 20.3(f). Thus the $i\omega$ axis represents the line of demarcation between stability and instability.

From Equation (20.4) the natural frequency of the system is

$$\omega_n = \sqrt{\frac{k}{m}} \tag{20.7}$$

and the damping ratio is

$$\xi = \frac{C}{2\sqrt{km}} \tag{20.8}$$

When $\xi = 1$, critical damping occurs. We set $C = C_c$ for this value of $\xi = 1$. Then

$$\xi = 1 = \frac{C_c}{2\sqrt{km}}$$

from which we obtain

$$C_c = 2\sqrt{km} \tag{20.9}$$

as the critical damping factor. The damping ratio is often written as

$$\xi = \frac{C}{C_c} \tag{20.10}$$

Example Problem 1

The system of Figure 20.1 has the following parameters: the weight W of mass m is 28.5 N, $C = 0.0650$ N s/cm, $k = 0.422$ N/cm. Determine the undamped natural frequency of the system, its damping ratio, and the type of response the system would have if the mass is to be initially displaced and released.

Solution

The mass is found from

$$m = \frac{W}{g} = \frac{28.5 \text{ N}}{9.807 \text{ m/s}^2} = 2.91 \text{ kg}$$

Then from Equation (20.7)

$$\omega_n = \sqrt{\frac{42.2 \text{ N/m}}{2.91 \text{ kg}}} = 3.81 \text{ rad/s}$$

From Equation (20.8)

$$\xi = \frac{C}{2\sqrt{km}} = \frac{6.50 \text{ N s/m}}{2\sqrt{(42.2 \text{ N/m})(2.91 \text{ kg})}} = 0.294$$

Since the damping ratio is less than unity, the response will be underdamped. The system response is that of the form shown in Figure 20.3(b).

Example Problem 2

For the system of Example Problem 1, find the amount of additional damping needed to have the system become critically damped.

Solution

From Equation (20.9), the condition for a critically damped system is

$$C_c = 2\sqrt{km} = 2\sqrt{(42.2 \text{ N/m})(2.91 \text{ kg})} = 22.13 \text{ N s/m}, = 0.2213 \text{ N s/cm}$$

The additional damping required is

$$\Delta C = 0.2213 - 0.0650 = 0.1563 \text{ N s/cm}$$

With this additional damping, Equation (20.2) becomes

$$\ddot{x}(t) + \frac{22.13}{2.91}\dot{x}(t) + \frac{42.2}{2.91}x(t) = 0$$

The characteristic equation in the Laplace transform variable form becomes

$$s^2 + \frac{22.13}{2.91}s + \frac{42.2}{2.91} = 0$$

or

$$(s + 3.81)^2 = 0$$

which demonstrates that Case 3 of Figure 20.3 exists.

20.3 General Solution for the One-Degree Model of Simple System

An initial displacement is introduced to the model system of Figure 20.1, and it is desired to determine the response of this system. The differential equation in standard mathematical shorthand notation is

$$\ddot{x}(t) + 2\xi\omega_n\dot{x}(t) + \omega_n^2 x(t) = 0 \tag{20.11}$$

which undergoes a Laplacean transformation into

$$s^2X(s) - sx(0) - \dot{x}(0) + 2\xi\omega_n\left[sX(s) - x(0)\right] + \omega_n^2 X(s) = 0$$

Let the initial displacement be represented by $x(0) = x_0$ and the initial velocity $\dot{x}(0) = 0$. The preceding equation reduces to

$$X(s) = x_0\frac{s + 2\xi\omega_n}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{20.12}$$

The graphical residue technique, described by Example Problem 4 in Appendix C, can be used to find the inverse transform for Equation (20.12) and hence the solution of Equation (20.11):

$$x(t) = \frac{x_0}{\sqrt{1-\xi^2}}e^{-\xi\omega_n t}\cos\left(\omega_n\sqrt{1-\xi^2}\,t - \frac{\pi}{2} + \cos^{-1}\xi\right)$$

which reduces to

$$x(t) = \frac{x_0}{\sqrt{1-\xi^2}}e^{-\xi\omega_n t}\sin\left(\omega_n\sqrt{1-\xi^2}\,t + \theta\right) \tag{20.13}$$

from the trigonometric relationship $\cos(\varphi - \pi/2) = \sin\varphi$ and by setting

$$\theta = \cos^{-1}\xi \tag{20.14}$$

Example Problem 3

Consider the system of Figure 20.1 which is initially displaced and then suddenly released. Its resultant motion is described by

$$x(t) = 3.0e^{-3.43t}\sin(11.4t + 60^\circ)$$

Find the system's damping ratio, natural frequency, and the initial displacement.

Solution

Applying Equations (20.13) and (20.14),

$$\theta = 60^\circ = \cos^{-1}\xi, \quad \xi = 0.500, \quad \text{and} \quad \xi\omega_n = 3.43$$

Hence $\omega_n = 6.86$. The initial condition is established from

$$3.0 = \frac{x_0}{\sqrt{1-\xi^2}} = \frac{x_0}{\sqrt{1-(0.500)^2}} = 1.155x_0$$
$$x_0 = 2.597$$

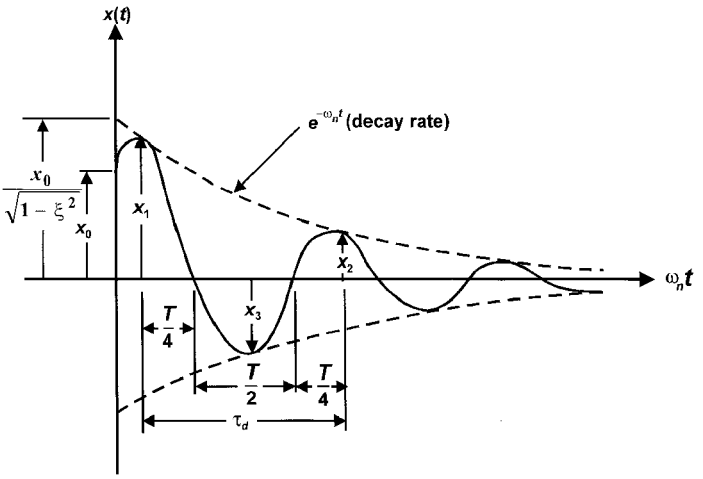


FIGURE 20.4. Response of a second-order system that is underdamped.

Because $\xi < 1$, this system is underdamped, and details of the response are shown in Figure 20.4. It is noted in the figure that because $x_1 > x_0$, i.e., a slope of x exists initially, an initial velocity is present. The actual damping is $\xi\omega_n$, and the damped period is defined as

$$\tau_d = \frac{2\pi}{\omega_n\sqrt{1-\xi^2}} \tag{20.15}$$

In Figure 20.4, it is evident that the decay rate $e^{-\xi\omega_n t}$ of the free oscillation depends on the system damping. The greater the damping, the faster is the rate of decay.

Let us now establish the relationship between the rates of decay and damping. First, the time response at two distinct points, each of which is a quarter period from the crossover points in the first two lobes, must be determined. These points are the points where the sine function equals unity and not the peak points of the damped response in Figure 20.4. From Equations (20.4) and (20.13), it is noted that

$$x_1 = \frac{x_0}{\sqrt{1-\xi^2}} e^{-\xi\omega_n t_1} \sin\left(\omega_n\sqrt{1-\xi^2}t_1 + \theta\right) \tag{20.16}$$

and

$$x_2 = \frac{x_0}{\sqrt{1-\xi^2}} e^{-\xi\omega_n(t_1+\tau_d)} \sin\left(\omega_n\sqrt{1-\xi^2}(t_1+\tau_d) + \theta\right) \tag{20.17}$$

Taking the ratio of the two amplitudes represented by Equations (20.16) and (20.17) and noting that in this case the sine functions equal unity, we obtain

$$\frac{x_1}{x_2} = \frac{e^{-\xi\omega_n t_1}}{e^{-\xi\omega_n(t_1+\tau_d)}} = e^{\xi\omega_n\tau_d}$$

The logarithmic decrement δ is defined at the natural logarithm of the ratio of two points such as x_1 and x_2. Invoking Equation (20.14) and this definition for logarithmic decrement, we can derive

$$\delta = \ln\left(\frac{x_1}{x_2}\right) = \frac{2\pi\xi}{\sqrt{1-\xi^2}} \tag{20.18}$$

Thus, the logarithmic decrement δ is expressed in terms of the damping ration ξ of the system.

A negative value of the response function can also be used to find the logarithmic decrement, but in this case τ_d should be replaced by $\tau_d/2$, because a half period is used in this evaluation. For this situation,

$$\delta = 2\ln\left(\frac{x_1}{x_3}\right)$$

The values of the response function at any two points can be used to find the two unknowns ξ and ω_n. It is also important to realize that the ratio of the points in the response curve, e.g., x_1/x_2, x_2/x_3, or x_3/x_4, will be identical only in the presence of viscous damping.

20.4 Forced Vibration

Forced vibration occurs when $f(t) \neq 0$ in Equation (20.1). The forcing function can be a harmonic excitation, an example of which is the imbalance in rotating machinery such as a motor. Or it can be an impulse type of excitation, such as that produced by a hammer or it can be simply the weight of the moving part itself.

Harmonic Excitation

A harmonic forcing function can be represented by $F_0 \sin \omega t$. The differential equation for the model of Figure 20.1 assumes the following form:

$$m\ddot{x} + C\dot{x} + kx = F_0 \sin \omega t \tag{20.19}$$

With the assumption of zero initial conditions, the Laplacian transform of Equation (20.18) yields

$$\left(ms^2 + Cs + k\right) X(s) = F_0\left(\frac{\omega}{s^2+\omega^2}\right) \tag{20.20}$$

and solving for $X(s)$

$$X(s) = \frac{F_0}{ms^2 + Cs + k}\left(\frac{\omega}{s^2+\omega^2}\right) \tag{20.21}$$

As a rule, the solution $x(t)$ will consist of two parts, viz. the complementary solution and a particular solution. The former corresponds to the transient part of the total solution and the latter to the steady-state part. The transient portion of the

solution is in the form of Equation (20.13) and it is determined by the residues of the complex poles, wherein the poles constitute the solution of

$$ms^2 + Cs + k = 0$$

The steady-state solution of Equation (20.19) is a sinusoidal oscillation expressed as

$$x(t) = x(\omega) \sin\ (\omega t - \phi) \tag{20.22}$$

and this solution is determined by the residues at the complex poles $s = s = \pm i\omega$. While the exact solution can be derived by using the residue method, the solution of Equation (20.19) can be readily obtained through the standard differential methods. The advantage of the Laplace transform method that it facilitates finding the frequency and stability information.

Through Equations (20.21) and (20.22), the magnitude of the steady-state oscillation can be determined from

$$X(s) = \frac{F_0}{ms^2 + Cs + k} \tag{20.23}$$

From Equation (20.23), the magnitude of the oscillation can be determined as a function of frequency. Because $s = \sigma + i\omega$ and $\omega = 2\pi \mathrm{f}$, substituting $s = i\omega$ into Equation (20.23) yields

$$x(i\omega) = \frac{F_0}{-m\omega^2 + i\omega C + k}$$

The denominator is a complex number, so the magnitude of this number is equal to

$$\sqrt{(k - m\omega^2)^2 + (\omega C)^2}$$

The magnitude of the oscillation as a function of frequency is

$$x(\omega) = \frac{F_0}{\sqrt{(k - m\omega^2)^2 + (C\omega)^2}} = \frac{F_0/k}{\sqrt{[1 - (m/k)\omega^2]^2 + (C\omega/k)^2}} \tag{20.24}$$

From the definitions of Equations (20.7) through (20.10), Equation (20.24) can be expressed as

$$\frac{X(\omega)}{F_0/k} = \frac{1}{\sqrt{[1 - (\omega/\omega_n)^2]^2 + [2\xi(\omega/\omega_n)]^2}} \tag{20.25}$$

The magnitude is now a function of only two quantities, the ratio (ω/ω_n) and the damping ratio ξ. The phase angle ϕ is also a function of these two parameters and it is given by

$$\tan\phi = \frac{2\xi(\omega/\omega_n)}{1 - (\omega/\omega_n)^2} \tag{20.26}$$

The system undergoes a resonance when the excitation frequency $f = \omega/2\pi$ equals the natural frequency of the dynamic system $f_n = \omega_n/2\pi$. The damping

TABLE 20.1. System Response as a Function of Frequency.

Frequency	Response	Controlling Parameter
$\omega^2 \ll \omega_n^2$	$x(\omega) = \dfrac{F_0}{k}$	Stiffness controlled
$\omega^2 \gg \omega_n^2$	$x(\omega) = \dfrac{F_0}{m\omega^2}$	Mass controlled
$\omega^2 = \omega_n^2$	$x(\omega) = \dfrac{F_0}{C\omega}$	Damping controlled

ratio ξ affects the magnitude of the oscillation peak at resonance and the sharpness of this resonant peak.

In considering the system response as a function of frequency, we observe that the response varies with the frequency as shown in Table 20.1. The relationships listed in the table show that each parameter listed—the stiffness, the mass, and damping—effectively controls the response only within a limited region. For example, the damping is primarily effective at resonance. The selection of any vibratory corrective measure depends on whether the excitation frequency is less than, greater than, or equal to the resonant frequency of the system.

The effect of the amplitude by stiffness, mass, or damping is exemplified by the *magnification factor* MF, defined as

$$\text{MF} = \frac{x(\omega)}{F_0/k} = \frac{1}{\sqrt{[1-(\omega/\omega_n)^2]^2 + [2\xi(\omega/w_n)]^2}} \tag{20.27}$$

At resonance $\omega = \omega_n$, and therefore

$$(\text{MF})_{\text{resonance}} = \frac{1}{2\xi} \tag{20.28}$$

A measure of the shape of the resonance peak is given by the bandwidth at the half-power points, as shown in Figure 20.5. These points are the two points, one to the right and one to the left of the peak, which have a magnitude equal to $(1/\sqrt{2})$ of the value of the peak. The square root occurs because power is proportional to the square of the magnitude.

Let us set $h \equiv \omega_n/\omega$. At half-power points, Equation (20.27) becomes

$$\frac{1}{2\sqrt{2}\xi} = \frac{1}{\sqrt{(1-h^2)^2 + (2\xi h)^2}}$$

Solving the preceding equation algebraically for h^2 results in

$$h^2 = 1 - 2\xi \pm 2\xi\sqrt{1+\xi^2}$$

We also assume small values of damping (i.e., $\xi \ll 1$) and neglect second-order terms. Then the following result occurs:

$$h^2 = 1 \pm 2\xi$$

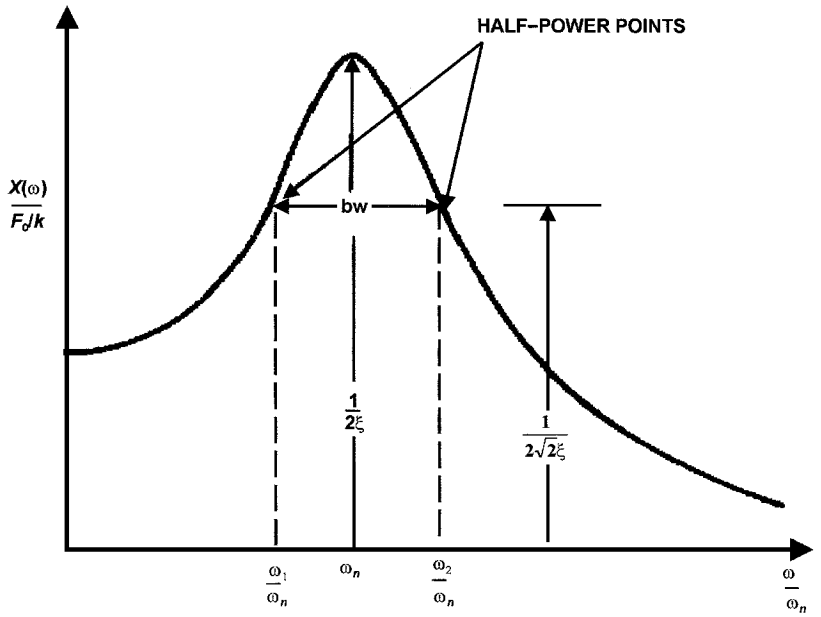


FIGURE 20.5. Principal features of a resonance peak.

Then

$$h_1 = \left(\frac{\omega}{\omega_n}\right)^2 = 1 - 2\xi, \qquad h_2 = \left(\frac{\omega}{\omega_n}\right)^2 = 1 + 2\xi$$

Hence

$$h_2 - h_1 = 4\xi$$

We approximate

$$\frac{\omega_2^2 - \omega_1^2}{\omega_n^2} \approx \frac{2(\omega_2 - \omega_1)}{\omega_n}$$

The bandwidth bw is given by

$$\text{bw} = \frac{(\omega_2 - \omega_1)}{\omega_n} = \frac{\Delta\omega}{\omega} = 2\xi \tag{20.29}$$

The reciprocal of the bandwidth is the quality factor Q, expressed as

$$Q = \frac{\omega_n}{\Delta\omega} = \frac{1}{2\xi} \tag{20.30}$$

When a frequency response of system is plotted in the format of Figure 20.5, this figure in conjunction with Equation (20.30) can provide the basis for finding the equivalent viscous damping of the system.

Excitation by Impulse

Impulse occurs very commonly as a cause of vibration excitation in the industrial environment. An impulse force is one that acts for a very short time. A hammer is an example that provides an impulse force. Such a force can be represented by the Dirac delta function or distribution that has the following properties: consider a pulse that starts at time $t = \varepsilon$ and has width a and height $1/a$. Let a approach zero, so that the pulse essentially approaches zero width and infinite height so that

$$\int_0^\infty \delta(t-\varepsilon)\,dt = 1 \qquad 0 < \varepsilon < \alpha \tag{20.31}$$

Here $\delta(t-\varepsilon)$ is the unit impulse function at $t = \varepsilon$. Equation (20.31) states that the area or strength of the impulse is unity. If we specify a pulse of width a and height A/a, then the impulse function $A\delta(t-\varepsilon)$ is of strength A at $t = \varepsilon$.

A system subjected to impulse force of strength A has a response determined by the following equation:

$$m\ddot{x} + C\dot{x} + kx = A\delta(t) \tag{20.32}$$

The Laplacian $\mathcal{L}[A\delta(t-\varepsilon)]$ is equal to A; and so if the initial conditions are zero, then Equation (20.32) transforms to

$$X(s) = \frac{A}{ms^2 + Cs + k}$$

If the system is damped slightly, then through the use of techniques described in Appendix C, the response can be calculated to yield

$$x(t) = \frac{A}{k\omega_n\sqrt{1-\xi^2}} e^{-\xi\omega_n t} \sin \omega_n \sqrt{1-\xi^2}\, t$$

where ω_n and ξ are the natural frequency and damping ratio, respectively, as defined previously.

Example Problem 4

Consider the system of Figure 20.1 in which the weight $W = 11.5$ N, $k = 9.45$ N/cm, and $C = 0.092$ N/cm/s. The maximum value of the harmonic force exciting the system is 14.25 N (or $F = 14.25 \sin \omega t$). Find the amplitude at resonance (which represents the maximum amplitude of the steady-state motion of the mass for small damping ratios).

Solution

From Equation (20.8)

$$\xi = \frac{C}{2\sqrt{km}} = \frac{9.2 \text{ N/m/s}}{2\sqrt{(945 \text{ N/m})(11.5 \text{ N}/9.807 \text{ m/s}^2)}} = 0.138$$

From Equation (20.27)

$$\frac{x}{F_0/k} = \frac{1}{2\xi}$$

Hence at resonance, the amplitude at resonance is

$$x_{\text{res}} = \frac{14.25 \text{ N}/945 \text{ N/m}}{0.276} = 0.0546 \text{ m} = 5.46 \text{ cm}$$

Static Deflection

In Figure 20.1 the deadweight W constitutes a static load that causes a deflection δ_{st} that is given by

$$\delta_{\text{st}} = \frac{W}{k} = \frac{mg}{k} \tag{20.33}$$

where g represents the gravitational constant. Because the natural frequency of the system is found from

$$f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}} \tag{20.34}$$

we can combine Equations (20.33) and (20.34) to obtain the relationship between static deflection and natural frequency as follows:

$$f_n = \frac{1}{2\pi}\sqrt{\frac{g}{\delta_{\text{st}}}} \tag{20.35}$$

Example Problem 5

A large machine weighs 875 N and the static deflection of the springs supporting the machine is 0.83 cm. Find the undamped natural frequency.

Solution

From Equation (20.35)

$$f_n = \frac{1}{2\pi}\sqrt{\frac{980.7 \text{ cm/s}^2}{0.83 \text{ cm}}} = 5.47 \text{ Hz}$$

20.5 Vibration Control

Transmissibility

In the case of a forcing function being harmonic in nature, two cases of vibration transmission can occur. One case occurs when force is transmitted to the supporting structure, and the opposite case occurs when the motion of the supporting structure

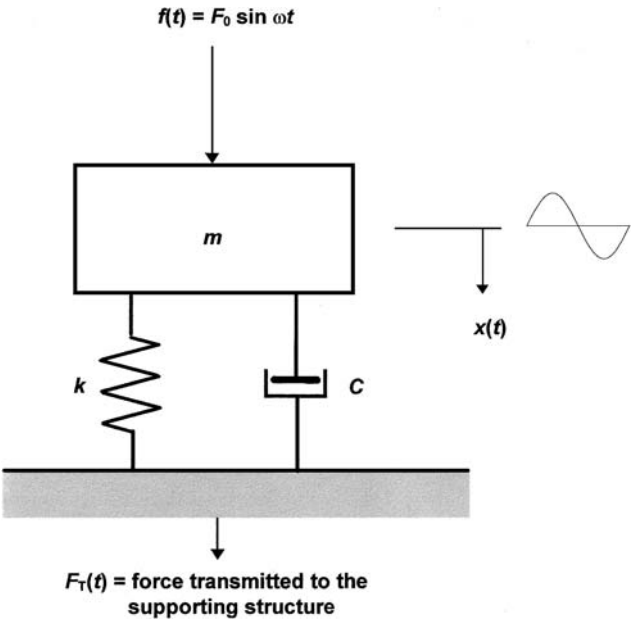


FIGURE 20.6. A spring-damper system that is subjected to force excitation.

is transmitted to the machine. In Figure 20.6, $f(t) = F_0 \sin \omega t$ represents the harmonic force imparted to the system and $f_T(t)$ is the force transmitted through the spring/damper system to the supporting structure. The force sent through the spring and damper to the supporting structure is given by

$$f_T(t) = kx + C\dot{x}$$

The magnitude of this force is a function of the frequency:

$$F_T = \sqrt{[kx(\omega)]^2 + [C\omega X(\omega)]^2} = kx(\omega)\sqrt{1 + \left(\frac{c\omega}{k}\right)^2} \tag{20.36}$$

Inserting Equation (20.24) into Equation (20.36) results in

$$F_T = \frac{F_0\sqrt{1 + \left(\frac{C\omega}{k}\right)^2}}{\sqrt{\left(1 - \frac{m\omega^2}{k}\right)^2 + \left(\frac{C\omega}{k}\right)^2}} \tag{20.37}$$

Applying the definitions for ω_n and ξ, and defining *transmissibility* T as the ratio of the amplitude of the force transmitted to the supporting structure to that of the

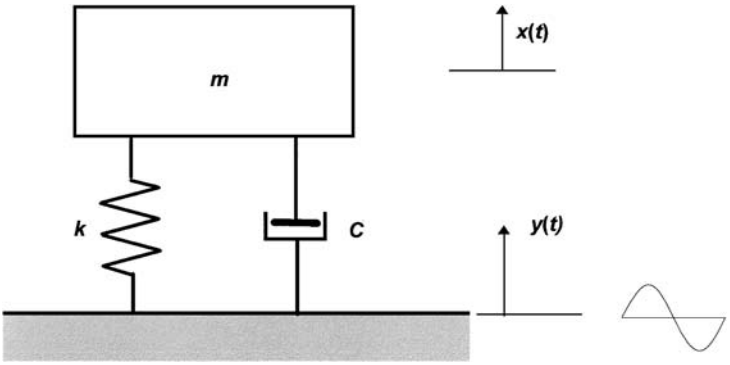


FIGURE 20.7. A spring-damper system that is subjected to motion excitation.

exciting force, we now have

$$T = \frac{F_T}{F_0} = \frac{\sqrt{1 + \left(2\xi \omega / \omega_n\right)^2}}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left(2\xi \frac{\omega}{\omega_n}\right)^2}} \tag{20.38}$$

Motion Excitation

As the corollary to the force excitation model of Figure 20.6, the system for motion excitation is given in Figure 20.7. Variable x represents the motion of the dynamic system and variable y represents the harmonic displacement of the supporting base. The dynamics of this system is characterized by the following equation:

$$m\ddot{x} + C(\dot{x} - \dot{y}) + k(x - y) = 0 \tag{20.39}$$

The ratio of the magnitudes of the two displacements as a function of frequency denotes the transmissibility given by

$$T = \frac{x}{y} = \frac{\sqrt{k^2 + (C\omega)^2}}{\sqrt{(k - m\omega^2)^2 + (C\omega^2)^2}} = \frac{\sqrt{1 + \left(2\xi \frac{\omega}{\omega_n}\right)^2}}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left(2\xi \frac{\omega}{\omega_n}\right)^2}} \tag{20.40}$$

The right-hand side of Equation (20.40), which was expressed in terms of ω_n and ξ, is identical to the transmissibility of Equation (20.38). This equality indicates that the methodologies employed to protect the supporting structure under force excitation are also applicable to insulating the dynamic system from motion excitation.

Equations (20.38) and (20.40) are used to plot Figure 20.8 in order to illustrate the interrelation between the damping ratio ξ, the ratio of disturbing frequency to

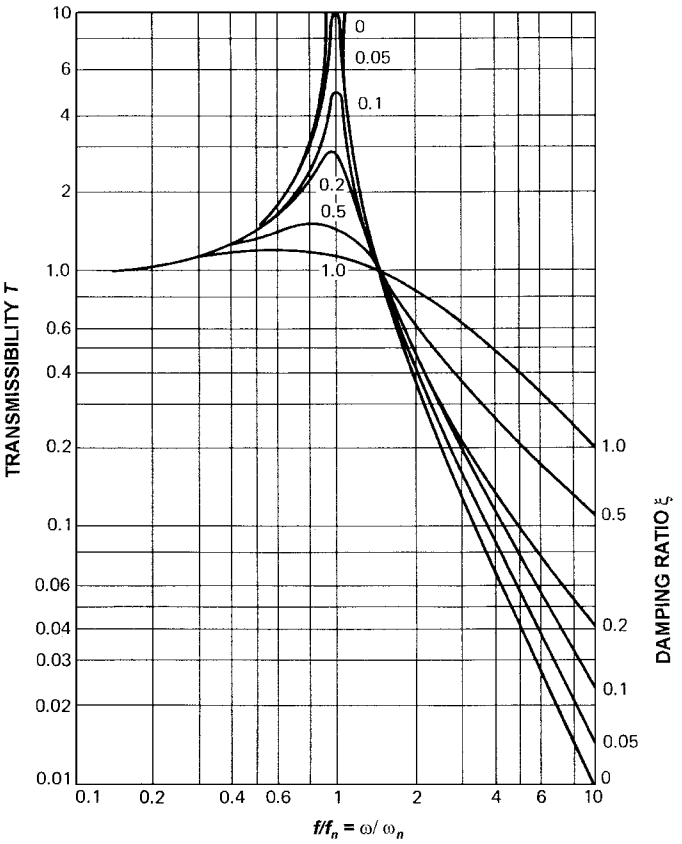


FIGURE 20.8. Plot of transmissibility versus frequency ration f/f_n as a function of the damping ratio of a linear system with one degree of freedom.

the natural frequency ω/ω_n, and transmissibility T. If the ratio $\omega/\omega_n < \sqrt{2}$, then transmissibility $T > 1$, which means that the input disturbance is amplified, not decreased. In the region $\omega/\omega_n > \sqrt{2}$, T decreases with increasing ξ. At resonance condition ω/ω_n, $= 1$, transmissibility can be quite large. The region $\omega/\omega_n > 2$ is the only domain where isolation is possible, and there T (which is less than unity) decreases with decreasing ξ, and this indicates that better isolation can be achieved with very little or no damping.

The curves of Figure 20.8 thus demonstrate the effectiveness of an isolator in mitigating vibration. It is also apparent that isolators should be selected to avoid exciting the natural frequencies of the system, and that damping is important in the range of resonance, when the system is operating near resonance or merely passing through resonance during startup. From scrutiny of the isolation region it is noted that the larger the ratio ω/ω_n (or the smaller the value of ω_n), the smaller the transmissibility will be. From the relation of Equation (20.7), ω_n can be made quite

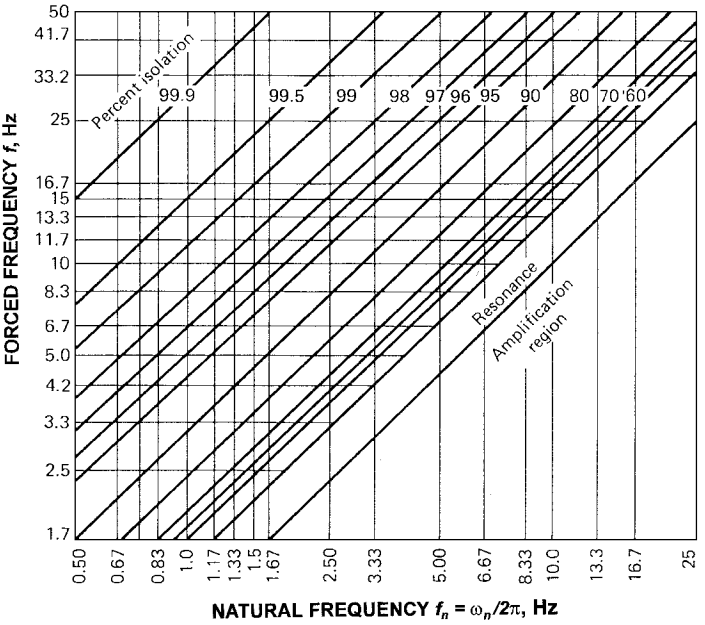


FIGURE 20.9. Isolation efficiency [(1 − T) × 100] versus natural frequency f_n of a linear single-degree-of-freedom system with zero damping ratio ξ.

small by selecting soft springs, which, in conjunction with light damping, provides good isolation. The natural frequency can also be reduced by increasing the mass but this also increases the dead weight of the dynamic system, thus imposing a greater load on the supporting structure.

A measure of the effectiveness of isolation as a function of frequency can be gained by the parameter *percent isolation* (*%I*) which is defined by

$$\%I = 100(1 - T)$$

where T is the transmissibility which is ≤ 1. The isolation efficiency is usually plotted with the disturbing frequency and natural frequencies as shown in Figure 20.9 for a damping ratio ξ of zero.

Example Problem 6

A machine weighs 208.5 N and it is supported on a spring having a spring constant of 935.23 N/cm. A rotating mass within the machine generates a disturbing force of 51.23 N at 4,800 rpm, owing to imbalance of the rotating mass. Determine the force transmitted to the mounting base for $\xi = 0.12$.

Solution

The static deflection of the spring is first obtained:

$$\delta_{\text{st}} = \frac{W}{k} = \frac{208.5 \text{ N}}{935.23 \text{ N/cm}} = 0.223 \text{ cm}$$

which, according to Equation (20.35), gives us the natural frequency of the system

$$f_n = \frac{1}{2\pi}\sqrt{\frac{980.7 \text{ cm/s}^2}{0.223 \text{ cm}}} = 10.6 \text{ Hz}$$

The disturbing frequency generated by the imbalance is $4800/60 = 80$ Hz. It follows that

$$\frac{\omega}{\omega_n} = \frac{f}{f_n} = \frac{80}{10.6} = 7.58$$

From Equation (20.40) or Figure 20.8 we find

$$T = \frac{\sqrt{1 + \left(2\xi \frac{\omega}{\omega_n}\right)^2}}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left(2\xi \frac{\omega}{\omega_n}\right)^2}} = \frac{\sqrt{1 + (2 \times 0.12 \times 7.58)^2}}{\sqrt{\left[1 - (7.58)^2\right]^2 + (2 \times 0.12 \times 7.58)^2}}$$

$$= \frac{2.076}{\sqrt{3187.33 + 3.509}} = \frac{2.076}{56.49} = 0.0367$$

The force transmitted to the mounting structure is

$$F_T = (0.0367)(51.23 \text{ N}) = 1.88 \text{ N}$$

20.6 Techniques for Vibration Control

Modification of Source

It is often possible to mitigate vibration by modifying the source of the vibration. For example, the vibrating structure may be made more rigid, closer tolerances of machining can eliminate or lessen imbalance, or the system mass and stiffness can be adjusted so that the resonant frequencies of the system do not coincide with the forcing frequency (this procedure is called *detuning*), or it may be possible to cut down on the number of coupled resonators between the vibrational source and the component of interest (this procedure is called *decoupling*).

Isolation

There are three principal types of isolators, viz. (1) metal springs, (2) elastomeric pads, and (3) resilient pads. Metal springs are generally the best for low-frequency isolation and they possess the advantage of being impervious to the effects of

temperature, humidity, corrosion, presence of solvents and they allow maximum deflections. But springs have almost no damping and hence can lead to very high transmissibility at resonance. But dampers can be included in parallel with springs to minimize resonance effects.

Elastomeric mounts are generally constructed of either natural rubber or synthetic rubber materials such as neoprene. Synthetic rubber is generally far more impervious to environmental effects than natural rubber. These types of mounts are generally used to isolate relatively small electrical and mechanical devices from relatively high-forcing frequencies. Elastomeric compounds inherently contain damping. Rubber can be used in either tension, compression or shear modes, but it is generally used in compression or shear and hardly ever used in tension modes.

Isolation or resilient pads include a variety of materials such as cork, felt, fiberglass, and special plastics. These items can be purchased in sheets and cut into gaskets to fit the particular application. Also, they can be stacked to provide different degrees of isolation. Cork is available in squares 1 to 2.5 cm thick. Cork can withstand corrosion and solvents, but felts being made of organic materials cannot be utilized in an environment where solvents are present. Cork and felt have damping ratios that typically range from $\xi = 0.05$ to 0.06.

Inertia Blocks

Certain very large machines such as reciprocating compressors generate large inertia forces that result in unacceptably large motion that can cause the machine to function improperly. One means of limiting this motion is to mount the equipment on an inertia base, which consists of heavy steel or concrete mass. This mass limits the motion by the dint of its heaviness overcoming the inertia forces produced by the mounted equipment. Also low natural frequency isolation needs a large deflection isolator such as a soft spring. But the use of soft springs can lead to rocking motions that cannot be tolerated. Hence, an inertia block mounted on the appropriate isolators can serve to effectively limit the motion as well as provide the required isolation. Inertia blocks also help to lower the center of gravity and thus provide an additional degree of stability. The introduction of additional mass through inertia blocks decreases vibrational amplitudes and minimizes rocking. Alignment errors can be minimized because of the greater stiffness of the base, and moreover, these blocks can serve as a noise barrier between the floor on which they are mounted and the mounted equipment itself.

Vibration Absorbers

In order to better understand its operation, let us consider the vibration absorber model of Figure 20.10. The components of the vibration absorber are m_2 and k_2. The applicable equations of motion for the system are

$$m_1\ddot{x}_1 + k_1x_1 + k_2(x_1 - x_2) = f$$
$$m_2\ddot{x}_2 + k_2(x_2 - x_1) = 0$$

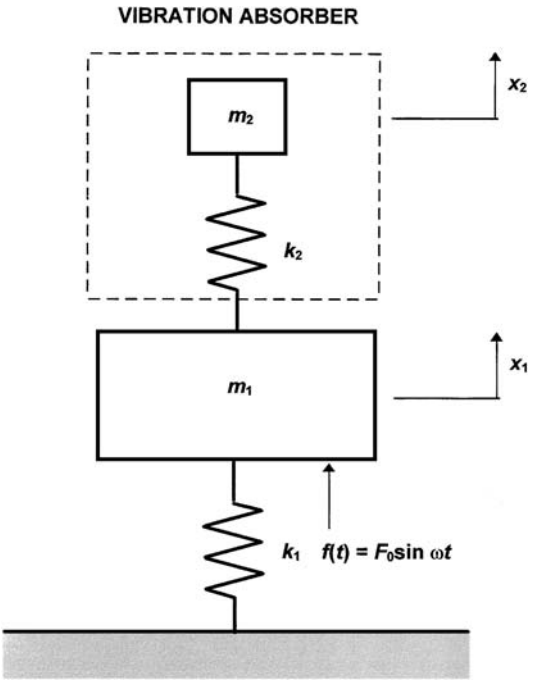


FIGURE 20.10. The analytical model for a vibration absorber.

The frequency response can be gotten from the following transformed equations:

$$-m_1\omega^2 X_1 + (k_1 + k_2)X_1 - k_2 X_2 = F_0$$
$$-k_2 X_1 + (-m_2\omega^2 + k_2)X_2 = 0$$

Rearranging these last two equations:

$$(-m_1\omega^2 + k_1 + k_2)X_1 - k_2 X_2 = 0 \tag{20.41}$$
$$-k_2 X_1 + (-m_2\omega^2 + k_2)X_2 = 0 \tag{20.42}$$

The two simultaneous equations may now be written in the matrix format as follows:

$$\begin{bmatrix} -m_1\omega^2 + k_1 + k_2 & -k_2 \\ -k_2 & -m_2\omega^2 + k_2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} F_0 \\ 0 \end{bmatrix} \tag{20.43}$$

Now consider what happens to Equations (20.41)–(20.43) when the forcing frequency ω equals the natural frequency $\omega_2 = (k_2/m_2)^{1/2}$ of the vibration absorber. This condition leads to

$$X_1 = 0$$
$$X_2 = -\frac{F_0}{k_2} \tag{20.44}$$

From the Equation set (20.44) it can be deduced that when the natural frequency of the vibrating absorber is tuned to the vibrational forcing frequency, the motion of the principal mass m_1 is ideally zero and the spring force of the absorber is at all times equal and opposite to the applied force F_0. It follows that no net force is transmitted to the supporting structure. Because the natural frequency must be tuned to the vibration forcing frequency, the absorber is customarily used for constant speed machinery. Even though this isolation feature is useful over a broad band of frequencies, absorption is useful principally for very narrow-band or single-frequency control.

Active Systems

Active isolation and absorption systems normally incorporate a feedback control, which may be electromagnetic, electronic, fluidic, pneumatic, mechanical, or a combination thereof. Because the cost of such systems is quite high, they are used mainly in precision instruments such as electronic microscopes, lasers, stabilized platforms, and other devices where a high degree of isolation is required. Active control is also applied where a high degree of isolation is required, for example, in the case of an air-conditioning system that is mounted on the top floor of a building above work areas and offices. Because the cost of computerization is falling continuously, active control is becoming a more viable option in many applications (including automotive suspensions), particularly for isolating very low-frequency vibration (<1 Hz) and also for tracking changes in natural frequencies due to shifting modes of operation.

Damping

Of the elements of vibration models, damping constitutes the most difficult of parameters to deal with, but it does provide an important means of cutting down vibration. Damping reduces the transmission of vibration through a structure, decreases amplitudes of resonance, and it attenuates free vibration and vibrations caused by impact. The reduction of vibrational energy in structures as the result of damping lessens the amount of noise radiated by the structural areas.

A measure of damping is that of the *system loss factor*, denoted by η_s and defined as the ratio of damping energy loss per radian to the peak potential energy. Let us assume an oscillation represented by

$$x(t) = A \sin\ (\omega t + \phi)$$

Then the velocity is

$$\dot{x}(t) = A\omega \cos\ (\omega t + \phi)$$

The damping force is given by $C\dot{x}(t)$; and the amount of energy dissipated through a spatial increment dx is $C\dot{x}(t)\,dx$. Because $\dot{x} = dx/dt$, we write

$$C\dot{x}(t)\,dx = C\dot{x}^2(t)\,dt = \frac{1}{\omega} C\dot{x}^2(t) d(\omega t)$$

Then the energy dissipated in a cycle is obtained from

$$\frac{C}{\omega}\int_0^{2\pi}\dot{x}^2(t)\,d(\omega t) = CA^2\omega\int_0^{2\pi}\cos^2(\omega t+\phi)\,d(\omega t) = \pi CA^2\omega$$

The energy dissipated per radian is $CA^2\omega/2$, and the peak potential energy is $kA^2/2$. Therefore, the system loss factor is

$$\eta_s = \frac{CA^2\omega/2}{kA^2/2} = \frac{C\omega}{k} \tag{20.45}$$

Introducing Equations (20.8) and (20.9), Equation (20.45) becomes

$$\eta_s = 2\xi\frac{\omega}{\omega_n} \tag{20.46}$$

At resonance, η_s assumes the value of 2ξ. This indicates that systems with large loss factors are highly damped. For small values of damping, applying Equations (20.18), (20.28), and (20.29):

$$\eta_S = 2\xi = \frac{\delta}{\pi} = \frac{1}{(\mathrm{MF})_{\text{resonance}}}$$

In the analysis of structural damping, complex stiffness and complex moduli are principal parameters. We recall that the steady-state oscillation is described by the following expression:

$$(-\omega^2 m + i\omega C + k)X = F_0$$

which can be rewritten as

$$(-\omega^2 m + k^*)X = F_0$$

where

$$k^* = k + i\omega C$$

The imaginary stiffness is the damping. Employing Equation (20.45), we can express the complex stiffness as

$$k^* = k(1 + i\eta_s)$$

In an analogous manner, the complex Young's modulus E for damping material can be expressed as

$$E^* = E(1 + i\eta_M)$$

where E is the real part of the Young's modulus of the damping material and η_M is the loss factor of the damping material.[1]

The loss factors and complex moduli vary with frequency and temperature. It has been assumed that the damping force is proportional to velocity and independent of amplitude, but there are situations where the damping of the materials depends

[1] η_s is the loss factor of the system structure *and* damping material.

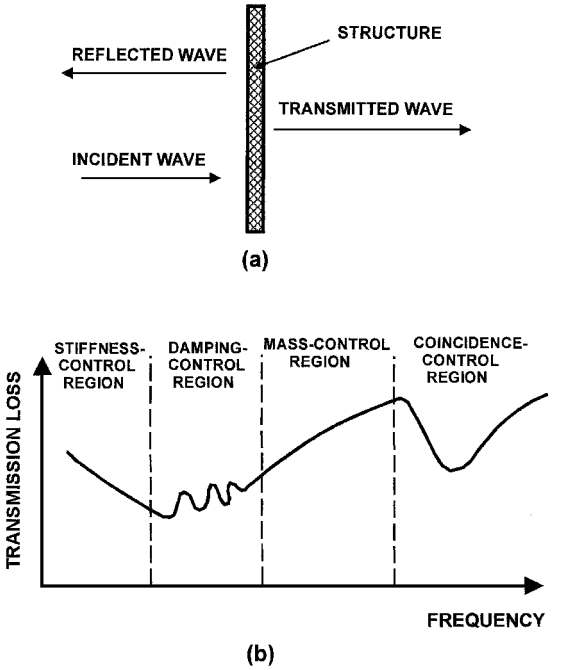


FIGURE 20.11. Sound transmission characteristics: (a) incident wave resolving into a reflected wave and a transmitted wave and (b) transmission loss in a structure as a function of frequency.

mainly on vibration amplitude, so the assumption of viscous damping must be applied judiciously.

Damping converts mechanical energy into thermal energy; and while there are a number of mechanisms for such energy conversion, we describe here those that are the most useful ones. Damping materials also reduce sound transmission. When a sound wave strikes a structure, causing its surface to vibrate, the vibrating surface produces a reflected wave and a transmitted wave [Figure 20.11(a)]. The transmission loss through the structure varies with frequency as shown in Figure 20.11(b) for a given temperature. The region of damping control nestles between the low-frequency region where stiffness reigns as the controlling parameter and the higher-frequency region where mass predominates as the controlling parameter. Between these two regions, many natural vibratory modes of the structure exist, and this region is the only one where transmission loss depends greatly upon resonance conditions. Here structural damping is the controlling parameter.

Internal Damping

A material with very high-damping internal properties could be utilized to eliminate noise emanating from a structure. Ferromagnetic materials and certain magnesium

and cobalt alloys exhibit such properties but these materials are generally too costly to use as structural materials and they may not meet the strength criteria.

Damping Mechanisms

Figure 20.12 shows three types of damping mechanisms, namely, the tuned damper, free viscoelastic layer, and constrained viscoelastic layer.

The tuned damper of Figure 20.12(a) consists of a mass attached to a point of vibration through a spring and dashpot or a viscoelastic spring. This device is not very useful because it functions well at only a single frequency or in a very narrow frequency band. Moreover, any change in temperature is likely to change the tuning frequency of the damper.

The other two mechanisms of Figure 20.12 entail adding viscoelastic layers to a structure that is to be damped. When a structure consisting of different layer of materials undergoes bending, the layers will extend or deform in shear. The resultant deformation causes energy dissipation, a phenomenon that constitutes the basis for the two viscoelastic mechanisms.

In the mechanism of Figure 20.12(b), a layer of free (i.e., uncovered) viscoelastic layer is bonded to the main structure. The damping material thickness should be

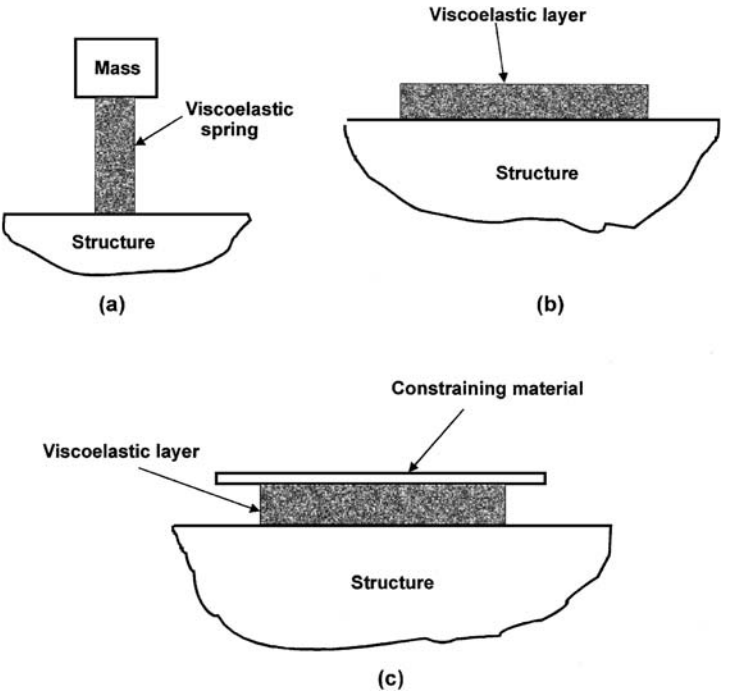


FIGURE 20.12. Different types of damping mechanisms: (a) tuned damper, (b) free viscoelastic layer, and (c) constrained viscoelastic layer.

about one-third the thickness of the structure or wall thickness. One thick layer on one side of the structure is generally more effective than two lesser layers on either side of the structure. This technique, which entails extensional damping, results in an economical, highly damped structure that is easy to fabricate.

The mechanism of Figure 20.12(c) is essentially a sandwich arrangement, where a viscoelastic layer is added to the structure. The covered (hence, constrained) viscoelastic layer provides high extensional damping, but the entire structure becomes harder and more expensive to assemble.

In order to be effective, the damping action must store a major portion of the energy present in the entire system. Damping is best applied to points where stretching or bending is the maximum, because these are the locations of maximal energy storage.

Viscoelastic layer techniques can be employed in a large variety of applications ranging from walls, enclosures, barriers, conveyers, chutes, racks, and hoppers to the most specialized, technologically sophisticated electronic instruments.

20.7 Finite Element Analysis

We have used relatively simple mathematical models to deal with vibration, but the analysis of plates, shells, and other continuous systems can be difficult to analyze without the aid of computers. Experimental modal analysis can provide the needed information, but this requires that a real structure be constructed and instrumented to yield the desired data. Finite-element analysis (FEA), however, allows the problem to be represented with some detail and permits the designer to optimize a design by investigating the effects of minor changes in the model upon the static and dynamic states of the structure being evaluated. FEA is also used not only to determine the statics and dynamics of beams, plates, shells, trusses, and other solid bodies but also to treat problems involving fluids, including airborne noise propagation. A number of FEA programs can predict stresses and strains, temperature distribution stemming from heat and mass transfer as well as vibratory states. Transient states are amenable to treatment by FEA. Since FEA is a powerful analytical tool, it no longer became necessary to fabricate a series of actual structures before freezing a design. The most prominent programs containing general codes for vibrational analysis include NASTRAN, developed by the U.S. National Aeronautics and Space Administration and now in public domain; MSC-NASTRAN®, a proprietary code developed from NASTRAN coding available from MacNeal-Schwindler Corp.; and ANSYS®, another proprietary code that is available from Swanson Analysis System, Inc.

A structure undergoing analysis is modeled by subdividing it into various types of finite elements from the finite-element library. The library may include more than 100 element types, including beam elements, triangular and rectangular plate and shell elements, conical shell elements, and mass, damping, and stiffness elements. The preprocessing stage in FEA entails the creation of a finite-element mesh to depict the structure being evaluated. The elements may be automatically sized

by proportional spacing so that the mesh is denser in areas of greater concern (particularly where sudden changes or discontinuities in geometry can lead to higher values of stresses or steeper temperature gradients). Constraints and forcing functions are specified. The input data are converted into matrix format by the FEA code. The form of matrix equation is

$$\mathbf{ma} + \mathbf{Cv} + \mathbf{kx} = \mathbf{F}$$

where

$\mathbf{m}$, $\mathbf{C}$, and $\mathbf{k}$ = the mass, damping, and stiffness matrices, respectively, for the finite element representation of the structure being analyzed

$\mathbf{a}$, $\mathbf{v}$, and $\mathbf{x}$ = the acceleration, velocity, and displacement vectors, respectively

$\mathbf{F}$ = the forcing function vector

The matrices are manipulated to determine natural frequencies, frequencies, mode shapes, response amplitudes, and/or dynamic stresses due to harmonic, random, or transient forcing functions. In the postprocessing phase, the FEA program arranges the output in a convenient format for engineering evaluation. The deformed structure can be graphically portrayed as an overlay over the undeformed mesh diagram. In order to show the difference, mode shapes or displacement amplitudes are exaggerated in the display. In almost all situations, a great deal of engineering time must be spent in defining even a simple problem by modeling it and setting up a mesh. But once a solution is obtained, redesign to improve vibrational characteristics can be done quickly and effectively by modifying the stored data. FEA software can be installed and used on personal computers, but for more elaborate designs it may be necessary to use minicomputers or mainframes in order to deal with much larger matrices (cf. Brooks, 1986; Hughes, 1987; Grandin, 1986; Huebner and Thornton, 1982).

20.8 Vibration Measurements

Vibration displacement and vibrational velocity can be measured but the most common measurements of vibration are those of acceleration. The basic transducer used to measure vibration and shock is the *accelerometer*. Most of the other components in vibration measurement systems are similar to those used to measure airborne sound (cf. Chapter 9). Many instruments such as the fast Fourier transform (FFT) analyzers are designed to be used for both acoustic and vibration applications.

Accelerometers

Accelerometers are usually mounted directly on a vibrating body, using a threaded stud, adhesive, or wax. Other versions mount the accelerometer in a probe that is held against a vibrating body. A typical accelerometer houses one or more

piezoelectric elements against which rest a mass that is preloaded by a stiff spring. When an accelerometer becomes subject to acceleration imparted by the vibration being measured, the mass exerts a force on the piezoelectric element that is proportional to the acceleration. The charge developed in the piezoelectric element is, in turn, proportional to the force. The accelerometer may be designed so that the piezoelectric element is stressed in compression or in shear. The piezoelectric elements are usually quartz crystals or specially processed ceramic materials.

There is more than one type of sensitivity of interest in accelerometer specifications. The *charge sensitivity*, measured in picocoulombs/g, and *voltage sensitivity*, measured in terms of mV/g, are important, depending whether the accelerometer is used with charge measuring or voltage measuring equipment. Also, the *transverse sensitivity*, which is the sensitivity to acceleration in a plane normal to the axis to the principal accelerometer axis should be a low value, preferably less than 3% of the main axis sensitivity at low frequencies.

The preamplifier, which constitutes the second stage in signal processing, serves two purposes: one is to amplify the vibration pickup signal that is generally quite weak and the other purpose is to serve as an impedance transformer between the accelerometer and the subsequent chain of equipment. A preamplifier may be designed to function as a voltage amplifier in which case the output voltage is directly proportional to the input voltage or it may function as a charge amplifier in which case the output voltage is proportional to the input charge. Each type of amplifier has its own advantages and disadvantages. When a charge amplifier is used, changes in cable length (which modifies the cable capacitance) have negligible effects on the measurements. When a voltage amplifier is employed, the system will be extremely sensitive to changes in cable capacitance. Because the input resistance of a voltage amplifier cannot be disregarded, the extremely low-frequency response of the system may be affected. But voltage amplifiers are usually less expensive and may be more reliable because they contain fewer components.

20.9 Random Vibrations

Most of the vibration problems discussed earlier in this chapter are *deterministic*, i.e., the forcing function can be described as a function of time. An example of a deterministic problem is the imbalance of a shaft operating at a known speed. Random vibrations, on the other hand, result from excitation that can be only described statistically. Structural excitation of an aircraft fuselage due to jet engine noise or turbulent flow is considered to be a *random process*. Both frequency and amplitude vary and they do not establish a deterministic pattern.

Probability Density

Because the amplitude or acceleration of random vibration cannot be determined as a function of time, it is described in terms of its probability density. One model widely used is the Gaussian distribution or normal probability density curve

expressed as follows in normalized form:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-0.5x^2/\sigma^2} \tag{20.47}$$

and the probability of the value of x falling between a and b is

$$P(a < x < b) = \int_a^b p(x)\,dx \tag{20.48}$$

where

$p(x)$ = the probability density of the function
x = the amount the function differs from the mean
σ = the standard deviation
P = the probability of x falling within a particular range

It is apparent that the total area under the probability density curve must be unity:

$$P(-\infty < x < \infty) = \int_{-\infty}^{+\infty} p(x)\,dx = 1$$

Root-Mean-Square Value and Autocorrelation

The time-average root-mean-square of a function is defined by

$$x_{\text{rms}} = \sqrt{\lim_{T\to\infty}\left(\frac{1}{T}\int_0^T x^2(t)\,dt\right)} \tag{20.49}$$

where

x_{rms} = the root-mean-square of function x
T = the time interval

The temporal autocorrelation function describes, on the average, the way in which the instantaneous value of a function depends on previous values. It is given by

$$R(\tau) = \lim_{T\to\infty}\left(\frac{1}{T}\int_0^T x(t)x(t+\tau)\,dt\right) \tag{20.50}$$

where

$R(\tau)$ = autocorrelation function
τ = the time interval between measurements
t = time

Ergodic Processes

A function may vary in such a manner that there is no narrow time interval that can be truly representative of the function. However, if the time interval is sufficiently long and the probability distribution functions are independent of the time interval during which they were measured, the function then represents a stationary process. This means that the root-mean-square value measured during one interval should be equal to the root-mean-square value measured during a later interval. The autocorrelation function will also be unaffected by a time shift. If root-mean-square values and the autocorrelation functions of a number of ensembles of data are equal to the temporal values, then the process may be considered *ergodic* as well as stationary.

Spectral Density

The power spectral density, also known as the root-mean-square spectral density, can be determined from the autocorrelation function as follows:

$$S(\omega) = \frac{\int_{-\infty}^{+\infty} R(\tau) e^{-i\omega\tau} d\tau}{2\pi} \tag{20.51}$$

where

$S(\omega)$ = root-mean-square spectral density, in g^2/(rad/s) or $(\text{m/s}^2)^2$/(rad/s)
ω = radial frequency, rad/s

Equation (20.51) is used in analytical studies. The inverse relationship is

$$R(\tau) = \int_{-\infty}^{+\infty} S(\omega) e^{i\omega\tau} d\omega$$

Vibration measurements obtained from an FFT analyzer or another spectral instrument may be expressed in dB re 1 g or dB re 1 m/s^2, within each frequency band. These values may be converted to root-mean-square spectral density $W(f)$ (usually expressed in g^2/Hz or some other engineering units), where $W(f)$ and $S(\omega)$ are related by

$$W(f) = 4\pi S(\omega)$$

where

$W(f)$ = spectral density, units^2/Hz), defined for positive frequencies only
f = frequency, Hz
$S(\omega)$ = spectral density, units^2/(rad/s), defined for both positive and negative frequencies (a mathematical artifice)

If the vibration measurements are sufficiently representative and the process is stationary, the root-mean-square value is then given by

$$x_{\text{rms}}^2 = \lim_{T\to\infty}\left(\frac{1}{T}\int_0^T x^2(t)\,dt\right) = \int_0^\infty W(f)\,df = R(0)$$

where

$$x_{\text{rms}}^2 = \text{root-mean-square value, } g^2$$
$$R(0) = \text{the autocorrelation function for } \tau = 0$$

White Noise

White noise is a random signal which has a constant root-mean-square spectral density for all frequencies from zero to infinity, i.e.,

$$W_{\text{white}}(f) = W_0 \qquad \text{for } 0 < f < \infty$$

This idealization cannot be achieved physically, since it would amount to requiring an infinite amount of power. More realistically, band-limited white noise can be achieved and it is a random signal having a constant spectral density over a specified range:

$$W_{\text{white}}(f) = W_0 \qquad f_1 < f < f_2$$

White noise generators are produced to generate signals with random vibration in amplitude and frequency, with relatively constant spectral density over various frequency ranges.

References

Beranek, Leo L. (ed.). 1971. *Noise and Vibration Control*. New York: McGraw-Hill.

Broch, J. T. June 1973. *Mechanical and Shock Measurements*. Nærum, Denmark: Brüel & Kjær Instruments Company.

Brooks, P. 1986. Solving vibration problems on a PC. *Sound and Vibration* 20(11):26–32.

Crandall, S. H. and Mark, W. D. 1963. *Random Vibration in Mechanical Systems*. New York: Academic Press.

Crede, C. E. 1965. *Shock and Vibration Concepts in Engineering Design*. Englewood Cliffs, NJ: Prentice Hall.

Grandin, H. 1986. *Fundamentals of the Finite Element Method*. New York: Macmillan.

Harris, Cyril M. (ed.). 1991. *Handbook of Acoustical Measurements and Noise Control*, 3rd ed. New York: McGraw-Hill, Chapters 6–10.

Harris, C. M. (ed.). 1988. *Shock and Vibration Handbook*, 3rd ed. New York: McGraw-Hill.

Huebner, K. H. and Thornton, E. A. 1982. *The Finite Element Method for Engineers*, 2nd ed. New York: Wiley-Interscience.

Hughes, T. J. R. 1987. *The Finite Element Method—Linear Static and Dynamic Finite Element Analysis*. Englewood Cliffs, NJ: Prentice Hall.

Problems for Chapter 20

1. In the spring-dashpot system of Figure 20.1, weight $W = 50$ N, spring constant $k = 0.30$ N/cm, and damping ratio $\xi = 0.35$. Find the natural frequency and the viscous damping of the system.
2. Establish the characteristic equation for the spring-dashpot system of Problem 1 and determine the additional viscous damping needed to yield a critically damped system.
3. In the system of Problem 1, the mass is displaced 15 cm and then suddenly released. Set up the equation for the system response, which describes the position of the mass as a function of time.
4. Consider a spring-dashpot system that has the following parameters: $W = 40$ N, $C = 0$, and $k = 0.39$ N/cm. If the mass is initially displaced and released, how will the system response?
5. In a spring-dashpot system, $W = 40$ N, $C = 0.10$ N/cm/s, and a forcing function $4.0 \sin \omega_n t$ drives the system. What kind of system response will occur as a function of time?
6. In a spring-damper system of Figure 20.1, $W = 250$ N, $C = 0.13$ N/cm/s, and $k = 32$ N/cm. How much does the spring deflect under the dead weight load of the mass? What is the natural frequency of the system? If the system undergoes a forced harmonic oscillation described by $F = F_0 \sin 12t$, which one of the system parameters effectively controls the response of the system?
7. A mass is supported by four identical springs, each having a spring constant of 2.6 N/cm. If the spring deflects 1.5 cm, what is the weight of the mass being supported?
8. Consider the system described in the last problem. It is forced with a sinusoidal signal with a frequency twice that of its natural frequency. Find the degree of vibration isolation that will be obtained.
9. A mechanical system is being designed for installation in a plant. It bears the following parameters: $W = 2000$ N and $k = 3.65$ N/cm. If the forcing frequency $= 12$ rad/s, how much damping can the system tolerate if the isolation must exceed 90%?
10. In a spacecraft, delicate electronic sensors have to be isolated from a panel that vibrates at 50 rad/s. It is required that at least 90% vibration isolation can be achieved by using springs to protect the equipment. Assume that the damping ratio $\xi = 0$. What static deflection is required?

21
Nonlinear Acoustics

21.1 Introduction

In most of the preceding chapters we have dealt with acoustics in terms of the linear wave equation. The amplitude of the sound was considered to be virtually infinitesimal, thus paving the path to relatively convenient mathematical analyses. When the wave amplitude becomes sufficiently large, nonlinear effects occur and the linear wave equation no longer meets the situation. Waveform deformation occurs with possible formation of shock waves, increased absorption, nonlinear interaction (as opposed to superposition) between combined sound waves, amplitude-dependent directivity of rays, onset of cavitation, and sonoluminescence.

The realm of nonlinear acoustics produced by intense sound levels encompasses a variety of practical cases: mufflers for internal combustion engines, thermoacoustic heat engines, shock waves from supersonic aircraft and spacecraft, underwater sonar, formation of bubbles, cavitation, acoustic compression and energy losses attributable to viscous and thermal boundary layers, sonoluminescence, and sonochemistry. Velocity dispersion also affects the resonance frequencies of rigid wall cavities and the harmonic spectrum of standing waves.

Even in cases of relatively small signals, matters are not always so linear as they appear. Executing a perturbation procedure on the wave equation to consider second- and third-order effects may bring out rheological characteristics (i.e., the non-Newtonian behavior of fluid viscosity) of a fluid medium that can be indicative of its physical and chemical state (Raichel and Kapfer, 1973; Takabayashi and Raichel, 1998).

Obviously not all aspects of nonlinear acoustics can be covered in a single chapter, but in the following sections we will attempt to highlight some of the major facets of the finite-amplitude category of acoustics.

21.2 Wave Distortions

Consider a plane wave propagating in the x-direction in a frictionless fluid. If the amplitude is sufficiently small, the wave phenomenon can be described

by

$$\nabla^2\phi = \frac{1}{c_0^2}\frac{\partial^2\phi}{\partial t^2}$$

where ϕ is the velocity potential, t is the time, and c_0 is the small signal sound speed listed in Tables A, B, and C of Appendix A. In linear acoustics a sound wave ordinarily propagates through a medium without changing its shape, because each part on the wave travels with the same speed $c_0 = dx/dt$.

In the case of the finite wave, the propagation speed varies from point to point. The variation occurs because a propagating wave engenders a longitudinal velocity field u in the fluid medium through which it travels. The motion of the fluid adds to the propagation speed with respect to a fixed location:

$$\frac{dx}{dt} = c + u \tag{21.1}$$

where c represents the speed of sound with respect to the moving fluid (and it is not the same as c_0). To understand this situation better, consider a fluid that is a gas for which the speed varies as $\sqrt{T}$ [cf. Equation (2.2)]. The sound speed becomes a bit higher when the acoustic pressure p is positive (i.e., during the compression phase that increases with the temperature) and a bit lower when p is negative (i.e., during the expansion phase that lowers the temperature). Then

$$c = c_0 + \frac{\gamma - 1}{2}u \tag{21.2}$$

where γ is the ratio of specific heats of the gas. Combining Equations (21.1) and (21.2) results in

$$\frac{dx}{dt} = c_0 + \beta u \tag{21.3}$$

where β, the coefficient of nonlinearity is given by

$$\beta = \frac{\gamma + 1}{2}$$

From Equation (21.3) we observe that the propagation speed depends on the particle velocity, as the result of the convective effect of the moving fluid and the nonlinearity of the traveling wave. Even in nonlinear acoustics, the particle velocity u is normally much smaller than c_0. The impact of the varying propagation speed is accumulative and leads to appreciable distortion that becomes even greater with increasingly stronger waves.

Nonlinearity of the pressure–density relation is considerably more prominent in liquids and solids than for gases. In the case of liquids the coefficient of the first-order nonlinear term in the pressure–density relation is $B/2A$, where B/A is termed the *parameter of nonlinearity*. In this case, the analog of Equation (21.1) for liquids is given by

$$\beta = 1 + \frac{B}{2A}$$

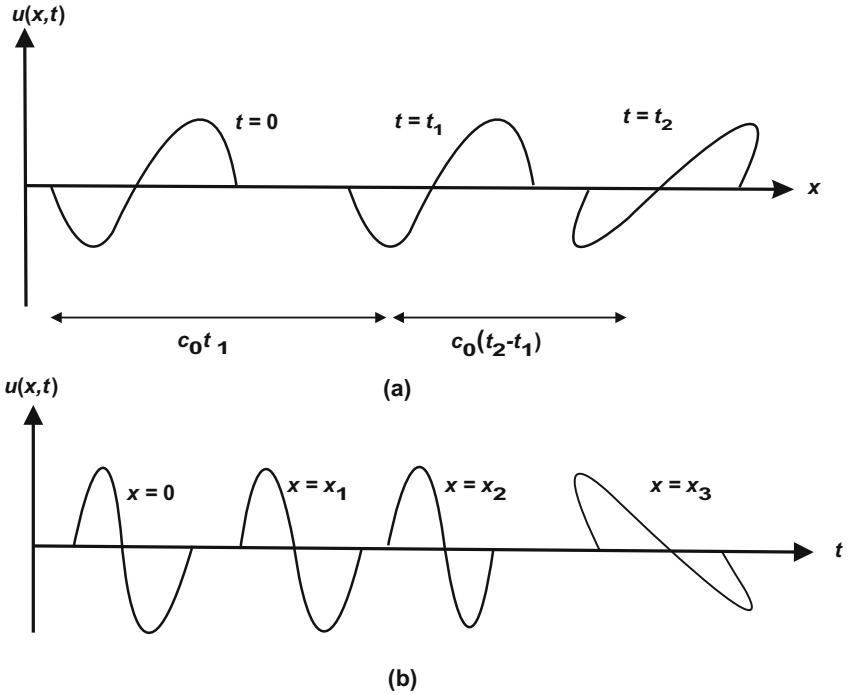


FIGURE 21.1. Cumulative distortion as seen from the viewpoints of (a) initial-value problem, u as a function of x with increasing time t, and (b) signaling problem, u as a function of x with increasing x, from the signaling location $x = 0$.

Nonlinear acoustics covers a considerable variety of topics. Problems arising from nonlinearity fall into two categories: (1) source problems, where the time variation of an acoustic field variable (e.g., pressure or particle velocity) is specified at a single location and (2) initial value problems where the spatial variation of the field is specified for a specific time. Figure 21.1 illustrates the cumulative distortion by linear convection as seen from the viewpoints of (a) initial value problems, with u as a function of x at increasing times t, and (b) signaling problems, where u is a function of t at increasing ranges x, from the signaling location.

21.3 Progressive Waves in Fluids

When a single wave moves through a fluid, we have a case of a progressive wave field. The governing second-order wave equation

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{c_0^2}\frac{\partial^2 u}{\partial t^2} \tag{21.4}$$

may be integrated at once to yield

$$\frac{\partial u}{\partial x} = \frac{1}{c_0}\frac{\partial u}{\partial t} \tag{21.5}$$

which is a first-order differential equation whose solution is $u = f(t - x/c)$.[1]

Of the two types of problem approaches mentioned in the preceding section, most of the acoustic problems entail radiation, so source problems receive more emphasis in this chapter.

Plane Waves

The model equation for a source-generated finite plane wave in a lossless fluid is

$$\frac{\partial u}{\partial x} = \frac{\beta}{c_0^2}u\frac{\partial u}{\partial t} \tag{21.6}$$

Consider a sinusoidal source excitation governed by $u = U_0 \sin \omega t$ at $x = 0$. The solution to Equation (21.6) is the Fubini solution

$$u = U_0 \sum_n \frac{2}{n\sigma} J_n(n\sigma)\, \sin n\omega t \tag{21.7}$$

where

$$\sigma = \beta\varepsilon\kappa x = \frac{x}{\bar{x}} \quad \text{or} \quad \bar{x} = \frac{1}{\beta\varepsilon\kappa x}$$

Here $\bar{x}$ is the *shock formation distance*, $\varepsilon = U_0/c_0$, and $\kappa = \omega/c_0$ is the wave number. The size of σ, the so-called dimensionless distortion range, indicates a measure of the amount of distortion that has occurred. The value of $\sigma = 1$ indicates shock formation.

Other One-Dimensional Waves and Ray Theory

In one-dimensional progressive waves that are not planar, geometric spreading slows down the rate of distortion and is described mathematically by an extra term in the model equation (21.6). If the wave is spherical or cylindrical, model equation (21.6) becomes

$$\frac{\partial u}{\partial r} + \frac{a}{r} = \frac{\beta}{c_0^2}u\frac{\partial u}{r} \tag{21.8}$$

---

[1] We can simplify Equation (21.5) by transforming coordinates x, t to x, τ, where $\tau = x - c_0/t$, the *retarded time.* Then Equation (21.5) becomes

$$\frac{\partial u}{\partial x} = 0$$

which provides the building block on which most model equations for more complex progressive waves are based.

where r represents the radial coordinate. The retarded time is now

$$\tau = t - \frac{(r - r_0)}{c_0}$$

where r_0 is a reference distance, which could be the radius of the source, and a has a value of unity for spherical waves, $\frac{1}{2}$ for cylindrical waves, and 0 for plane waves (in which case r is replaced by x). Introducing the coordinate stretching function

$$\begin{aligned} z &= r_0 \ln \frac{r}{r_0} && \text{(spherical waves)} \\ &= 2(\sqrt{r r_0} - r_0) && \text{(cylindrical waves)} \end{aligned}$$

and the spreading compensation function

$$w = \left(\frac{r}{r_0}\right)^a u \tag{21.9}$$

into Equation (21.8) results in this equation being reduced to the plane wave form

$$\frac{\partial w}{\partial z} = \frac{\beta}{c_0^2} w \frac{\partial w}{\partial \tau}$$

Thus, plane wave solutions may be extended to spherical and cylindrical waves by replacing u and $x$with w and z, respectively. The Fubini solution for spherical waves has the same form as that of Equation (21.7).

One-dimensional propagation is applicable to ducts of slowly varying cross section, such as horns or ray tubes. In such cases, the spreading compensation function is

$$w = \sqrt{\frac{A}{A_0}} u$$

Dissipative Function: Burgers Equation

Equation (21.8) can be useful for a good variety of finite-amplitude problems, but the distortion nearly always leads to the formation of shocks that are naturally dissipative. The losses must now be taken into account. The first truly successful model was developed by Burgers (1948) originally to model turbulence, but it also turned out to be an excellent approximation of the equation describing finite-amplitude plane waves of traveling in a thermoviscous fluid. In the form usable for source problems, the Burgers equation is

$$\frac{\partial u}{\partial x} - \frac{\beta}{c_0^2} u \frac{\partial u}{\partial r} = \frac{\delta}{2c_0^3} \frac{\partial^2 u}{\partial t^2} \tag{21.10}$$

where

$$\begin{aligned}
\delta &= \upsilon[\nu + (\gamma - 1)\,\mathrm{Pr}] = \text{diffusivity of sound} \\
\upsilon &= \mu/\rho = \text{kinematic viscosity} \\
\mu &= \text{shear viscosity coefficient} \\
\rho_0 &= \text{static density} \\
\nu &= \frac{4}{3} + \frac{\mu_B}{\mu} = \text{viscosity number} \\
\mu_B &= \text{bulk viscosity number} \\
\mathrm{Pr} &= \frac{c_p \mu}{k_r} = \text{Prandtl number, a property of the fluid} \\
c_p &= \text{specific heat at constant pressure of the fluid} \\
k_r &= \text{thermal conductivity of the fluid}
\end{aligned}$$

On comparing Equations (21.8) and (21.10) we observe that the former equation does not have a dissipation term and it is often called a lossless Burgers equation.

The Burgers equation is exactly integrable (Cole and Hopf), which renders it a very useful model. Applications of the Burgers equation have been made to sinusoidal source excitations. Although it is exact, the extraction of the solution is quite complicated (Hopf, 1950; Blackstock, 1964). For distances larger than $3\bar{x}(\sigma > 3)$, it reduces to the Fay solution (Fay, 1931)

$$u = u_0 \frac{2}{\Gamma} \sum \frac{\sin n\omega\tau}{\sinh \dfrac{n(1+\sigma)}{\Gamma}} \tag{21.11}$$

where $\Gamma = \beta\varepsilon\kappa/\alpha = 1/(\alpha\bar{x})$ that characterizes the importance of nonlinear distortion to the absorption process and $\alpha = \delta\omega^2/2c_0^3$ represents the small signal absorption at source frequency.

In addition to one-dimensional sound waves in thermoviscous fluids, the Burgers equation can also be applied to spherical and cylindrical waves, with the model equation (21.8) now including the right-hand side of Equation (21.10), i.e.,

$$\frac{\partial u}{\partial r} + \frac{u}{r} - \frac{\beta u}{c_0^2}\frac{\partial u}{\partial r} = \frac{\delta}{2c_0^2}\frac{\partial^2 u}{\partial t^2} \tag{21.12}$$

But Equation (21.12) is not exactly integrable (i.e., no exact solution is known) and numerical methods of solution would therefore be required.

Shock Waves

Sounds from impulsive sources strong enough to result in shock waves include blast waves from explosions, thunder from lightning, aircraft sonic booms, ballistic missiles, and N-waves from spark sources.

The basic equations describing shock-wave propagation incorporate conservation of mass, Newton's second law, conservation of energy, and an equation of

state relating pressure to density. These equations are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{21.13}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p \tag{21.14}$$

$$\rho \frac{\partial}{\partial t} \left(\frac{1}{2} u^2 + e \right) + \mathbf{u} \cdot \nabla \left(\frac{1}{2} u^2 + e \right) = 0 \tag{21.15}$$

and

$$p = p(\rho, T) \tag{21.16}$$

where ρ is the fluid density, $\mathbf{u}$ is the particle velocity, T is the temperature, p is the pressure, and e is the internal energy. We have noted in Chapter 2 how these equations were linearized to derive the basic acoustic equation. When Equations (21.13)–(21.16) are solved for a frictionless fluid, the solutions yield multivalued waveforms. But with increasing amplitude, attenuation mechanisms, such as viscosity and heat conduction, assume increasingly greater roles, especially in the case of large pressure gradients which leads to small but finite times rise of the pressure waveform.

For sufficiently strong waves, shock waves or flow discontinuities occur, and Equations (21.13)–(21.15) no longer apply. The governing equations across the shocks are the *Rankine-Hugoniot equations*:

$$[\rho(u - u_{\text{sh}})]_+ = [\rho(u - u_{\text{sh}})]_- \tag{21.17}$$

$$[\rho u(u - u_{\text{sh}}) + p]_+ = [\rho u(u - u_{\text{sh}}) + p]_- \tag{21.18}$$

$$\left[p \left(\frac{1}{2} u^2 + e \right) (u - u_{\text{sh}}) + pu \right]_+ = \left[p \left(\frac{1}{2} u^2 + e \right) (u - u_{\text{sh}}) + pu \right]_- \tag{21.19}$$

Here e represents the internal energy per unit mass. The subscript "$-$" denotes the variable just behind the shock, and the subscript "$+$" denotes the variable immediately in front of the shock. Equations (21.17), (21.18), and (21.19) express the conservation of mass, Newton's second law, and work–energy conservation, respectively.

In many situations, we can consider the air or gas surrounding an explosion or energy release to be a polytropic gas with a constant ratio of specific heats. The shock relations for a polytropic gas are expressed in terms of shock strength z defined by

$$z = \frac{(p_- - p_+)}{p_+} \tag{21.20}$$

and the Mach number M of the shock relative to the flow ahead:

$$M = \frac{(u_{\text{sh}} - u_+)}{c_+}$$

The shock relations are as follows:

$$M = \frac{u_- - u_+}{c_+} = \frac{z}{\gamma\sqrt{\left\{1 + \left(\frac{(\gamma+1)}{2\gamma}\right)z\right\}}}$$

$$\frac{\rho_-}{\rho_+} = \frac{1 + \frac{\gamma+1}{2\gamma}z}{1 + \frac{\gamma-1}{2\gamma}z}$$

$$\frac{c_-}{c_+} = \sqrt{\frac{1 + z\left\{1 + \left(\frac{\gamma-1}{2\gamma}\right)z\right\}}{1 + \left[\left(\frac{\gamma+1}{2\gamma}\right)z\right]}}$$

The simplest model of an explosion is that of a point blast and the exact solution of this problem has been developed by von Neumann (1963), Taylor (1950), and Sedov (1961). The explosion is depicted as an energy release E concentrated at a point in space with ambient density ρ_0. The relations between the position of the shock $r(t)$ and overpressure in terms of E, ρ_0, and r are

$$r(t) = k\left(\frac{E}{\rho_0}\right)^{1/5} t^{2/5} \tag{21.21}$$

$$p = \frac{8}{25}\frac{k^2\rho_0}{\gamma+1}\left(\frac{E}{\rho_0}\right)^{2/5} t^{-6/5}$$

or equivalently,

$$p = \frac{8}{25}\frac{k^5}{\gamma+1}Er^{-3} \tag{21.22}$$

The decay of a strong spherical shock with r^{-3} constitutes a general behavior of the strong shock region. The dimensionless constant k is established by requiring conservation of energy and is a function only of the ratio γ of specific heats. This necessitates that

$$E = \int_0^{r(t)} \left(\frac{p}{\gamma-1} + \frac{\rho u^2}{2}\right) 4\pi r^2 dr \tag{21.23}$$

remains constant.

Equation (21.23) has been evaluated numerically by Taylor and analytically by Sedov and von Neumann. For $\gamma = 1.4$ at standard atmosphere, the pressure equation (21.22) becomes

$$p = \frac{0.155E}{r^3}$$

Analytical studies have been made of cylindrical explosions and line source energy releases (Lin, 1954; Plooster, 1970, 1971). Lin was working on the problem of shock generation by meteors or missiles, whereas Plooster was studying the development and decay of cylindrical shocks with more realistic initial energy distributions and atmospheres. A strong shock estimate of the decay of cylindrical waves from an instantaneous line source in air ($\gamma = 1.4$) at standard pressure is given by

$$p = \frac{0.216E}{r^2}$$

Thus, a cylindrical decay proportional to r^{-2} is characteristic of cylindrical strong shocks.

Nonlinearity in Solids

The propagation of ultrasonic waves of finite amplitudes in a crystal of cubic symmetry can be analyzed with a nonlinear differential equation similar to that used for fluids.

The propagation of a finite-amplitude ultrasonic wave in any direction in an isotropic solid is given by

$$\rho_0 \frac{\partial^2 \xi}{\partial t^2} = \kappa_2 \frac{\partial^2 \xi}{\partial a^2} + (3\kappa_2 + k_3) \frac{\partial^2 \xi}{\partial a^2} \frac{\partial \xi}{\partial a} \tag{21.24}$$

where ξ is the particle displacement and a is the distance measured in the direction of propagation. κ_2 and κ_3 represent elastic constants that depend on the direction of propagation in the solid.

The solution of the nonlinear equation (21.24) is made by assuming that the wave is initially sinusoidal at $a = 0$, with $\xi = A_1 \sin(ka - \omega t)$. On this assumption we can obtain a perturbation solution in the form

$$\xi = A_1 \sin(ka - \omega t) + \frac{\beta A_2^2 k^2 a}{4} \cos 2(ka - \omega t) + \cdots. \tag{21.25}$$

where β is the nonlinear parameter. The negative ratio of the coefficient of the nonlinear term in Equation (21.24)

$$2\beta = -\left(3 + \frac{\kappa_3}{k_2}\right)$$

is the quantity to be determined from measurement. From 2β we can establish κ_3 since $\kappa_2 = \rho c_0^2$ is known.

Equation (21.25) indicates that an initially sinusoidal ultrasonic wave generates a second harmonic as it propagates. In order to measure the nonlinear parameter 2β, the absolute value of the fundamental displacement amplitude A_1 needs to be measured and also that of the second harmonic

$$A_2 = \frac{\beta A_1^2 k^2 a}{4}$$

But the propagation constant $k = 2\pi/\lambda = \omega/c_0^2$ is known and sample length a can be measured. We can then write

$$2\beta = \frac{8A_2c_0^2}{A_1^2\omega^2 a}$$

and determine 2β directly. A_2 is usually plotted as a function of A_1^2, and the slope is then evaluated to yield 2β.

References

Beyer, Robert T. 1974. *Nonlinear Acoustics*. Washington, DC: Navy Sea Systems Command. (Reissued in 1997 by the Acoustical Society of America, with an appendix containing references to new developments.)

Beyer, Robert T. (ed.). 1984. *Nonlinear Acoustics in Fluids*. New York, NY: Van Nostrand. (A collection of fundamental papers on nonlinear acoustics.)

Blackstock, David T. 1964. On plane, spherical, and cylindrical sound waves of finite amplitude in lossless fluids. *Journal of the Acoustical Society of America* 36:217–219.

Blackstock, David T. 1985. Generalized Burgers equation for plane waves. *Journal of the Acoustical Society of America* 77:2050–2053.

Blackstock, David T. 1972. Nonlinear acoustics (theoretical). In: Gray, D. E. (ed.). *American Institute of Physics Handbook*. New York, NY: McGraw-Hill, pp. 3-183–3-205.

Burgers, J. M. 1948. A mathematical model illustrating the theory of turbulence. In: Von Mises, R. and von Kàrmàn, T. (eds.). *Advances in Applied Mechanics*, Vol. I. New York, NY: Academic, pp. 171–191.

Crocker, Malcolm F. (ed.). 1998. Part II: Nonlinear Acoustics and Cavitation. *Handbook of Acoustics*, pp. 175–253.

Fay, R. D. 1931. Plane sound waves of finite amplitude. *Journal of the Acoustical Society of America* 3:222–241.

Fubini, E. 1935. Anomalies in the propagation of acoustic waves of great amplitude. *Alta Frequenza* 4:530–581 (in Italian).

Hamilton, Mark F. and Blackstock, David T. 1988. On the coefficient of nonlinearity of nonlinear acoustics. *Journal of the Acoustical Society of America* 83:1555–1561.

Hopf, E. 1950. The partial differential equation $u_t + uu_x = \mu u_{xx}$. *Communications in Pure and Applied Mathematics* 3:201–230.

Landau, Lev D. 1945. On shock waves at large distances from the place of their origin. *U.S.S.R. Journal of Physics* 9:496–503.

Lighthill, James. 1972. The propagation of sound through moving fluids. *Journal of Sound and Vibration* 24:471–492.

Lighthill, James. 1978. *Waves in Fluids*. New York, NY: Cambridge University Press. (Classic text by a former holder of the Lucasian chair at Cambridge University once hold by Isaac Newton and now occupied by Stephen Hawking.)

Lin, S.-C. 1954. Cylindrical shock waves produced by instantaneous energy release. *Journal of Applied Physics* 25:54–57.

Plooster, M. N. 1970. Shock waves from line sources. Numerical Solutions and experimental measurements. *Physics of Fluids* 13:2665–2675.

Plooster, M. N. 1971. Numerical simulation of spark discharges in air. *Physics of Fluids* 14:2111–2123.
Raichel, Daniel R. and Kapfer, William H. 1973. Sound propagation in non-Newtonian fluids. *Journal of Applied Mechanics* 40(Series E, 1):1–6.
Sedov, L. I. 1961. *Similarity and Dimensional Methods of Mechanics.* New York, NY: Academic.
Takabayashi, Kenneth and Raichel, Daniel R. 1998. Discernment by acoustic means of non-Newtonian behavior in liquids. *Rheologica Acta* 37(6):593–600.
Taylor, G. I. 1950. The formation of a blast wave by a very intense explosion: I. Theoretical discussion. *Proceedings of the Royal Society* 201:159–174.
Von Neumann, John. 1963. The point source solution. In: Taub, A. H. (ed.). *The Collected Works of John von Neumann*, Vol VI. New York, NY: Pergamon, pp. 219–237.
Whitham, G. B. 1974. *Linear and Nonlinear Waves*. Wiley Interscience.

Problems for Chapter 21

1. What is the change in timing for a sound traveling at $c_0 = 1100$ m/s from a spherical source of radius 1 m at 25 cm away from the source center due to nonlinearity?
2. What is the coordinate stretching function z for (1) a spherical wave and (2) a cylindrical wave?
3. What is the spreading compensation function for a divergent nozzle in terms of percentage of particle velocity u for a throat diameter of 2.5 cm at a location where the diameter is 5.6 cm?
4. Why is it necessary to resort to Rankine-Hugoniot relations instead of the customary fluid mechanics equations in flows entailing shock waves?

Appendix A
Physical Properties of Matter

A.1. Solids

Solid	Density, ρ (kg/m^3)	Young's Modulus, E (GPa)	Shear Modulus, G (GPa)	Poisson's Ratio, ν	Speed, c (m/s)	Characteristic Impedance, $\rho_0 c$ (10^6 Pa·s/m)
Aluminum	2700	71	24	0.33	5150	13.9
Brass	8500	104	38	0.37	3500	29.8
Copper	8900	122	44	0.35	3700	33.0
Iron (cast)	7700	105	44	0.28	3700	28.5
Lead	11,300	16.5	5.5	0.44	1200	13.6
Nickel	8800	210	80	0.31	4900	43.0
Silver	11,300	78	28	0.37	2700	28.4
Steel	7700	195	83	0.28	5050	39.0
Glass (Pyrex)	2300	62	25	0.24	5200	12.0
Quartz (X-cut)	2650	79	39	0.33	5450	14.5
Lucite	1200	4	1.4	0.4	1800	2.15
Concrete	2600	—	—	—	—	—
Ice	920	—	—	—	—	—
Cork	240	—	—	—	—	—
Wood, oak	720	—	—	—	—	—
Wood, pine	450	—	1	0.4	1450	1.6
Hard rubber	1100	2.3	—	0.5	70	0.065
Soft rubber	950	0.005	—	—	—	—
Rubber, ρ–c	1000	—	—	—	—	—

A.2. Liquids

Liquid	Temperature (°C)	Density, ρ_0 (kg/m^3)	Bulk Modulus, B (GPa)	Ratio of Specific Heats, γ	Speed, (m/s)	Characteristic Impedance, ρ_0/c (10^6 Pa·s/m)	Coefficient of Shear Viscosity, η (Pa·s)
Alcohol (ethyl)	20	790	—	—	1150	0.91	0.0012
Castor oil	20	950	—	—	1540	1.45	0.96
Glycerine	20	1260	—	—	1980	2.5	1.2
Mercury	20	13600	25.3	1.13	1450	19.7	0.0016
Seawater	13	1026	2.28	1.01	1500	1.54	0.001
Turpentine	20	870	1.07	1.27	1250	1.11	0.0015
Water (fresh)	20	998	2.18	1.004	1481	1.48	0.001
Fluidlike sea bottoms							
Red clay		1340	—	—	1460	1.96	—
Calcareous ooze		1570	—	—	1470	2.31	—
Coarse silt		1790	—	—	1540	2.76	—
Quartz sand		2070	—	—	1730	3.58	—

A.3. Gases (at 1 atmosphere pressure, 101.3 kPa)

Gas	Temperature (°C)	Density, ρ_0 (kg/m^3)	Ratio of Specific Heats, γ	Speed (m/s)	Characteristic Impedance, ρ_0/c (Pa·s/m)	Coefficient of Shear Viscosity η (Pa·s)
Air	0	1.293	1.402	331.6	428	0.000017
Air	20	1.21	1.402	343	415	0.0000181
Hydrogen	0	0.09	1.41	1269.5	114	0.0000088
CO_2 (low frequency)	0	1.98	1.304	258	512	0.0000145
CO_2 (high frequency)	0	1.98	1.40	268.6	532	0.0000145
Oxygen	0	1.43	1.40	317.2	453	0.00002
Steam	100	0.6	1.324	404.8	242	0.000013

A.4. Conversion Factors

To Convert	Into	Multiply by	Conversely, Multiply by
atm (atmosphere)	mmHg at 0°C	760	1.316×10^{-3}
	lb/in.^{2}	14.70	6.805×10^{-2}
	N/m^2 (Pa)	1.0132×10^5	9.872×10^{-6}
	kg/m^2	1.033×10^4	9.681×10^{-5}
°C (Celsius)	°F (fahrenheit)	$[(°C \times 9)/5] + 32$	$(°F - 32) \times 5/9$
cm (centimeter)	in. (inch)	0.3937	2.540
	ft (foot)	3.281×10^{-2}	30.48
	m (meter)	10^{-2}	10^2
cm^2 (square centimeter)	in.^{2}	0.1550	6.452
	ft^2	1.0764×10^{-3}	929
	m^2	10^{-4}	10^4
cm^3 (cubic centimeter)	in.^{3}	0.06102	16.387
	ft^3	3.531×10^{-5}	2.832×10^4
	m^3	10^{-6}	10^6
dyne	lb (force)	2.248×10^{-6}	4.448×10^5
	N (newton)	10^{-5}	10^5
dynes/cm^2	lb/ft^2 (force)	2.090×10^{-3}	478.5
	N/m^2 (Pa)	10^{-1}	10
ft (foot)	in. (inch)	12	0.08333
	cm (centimeter)	30.48	3.281×10^{-2}
	m (meter)	0.3048	3.281
ft^2 (square foot)	in.^{2}	144	6.945×10^{-3}
	cm^2	9.290×10^2	0.010764
	m^2	9.290×10^{-2}	10.764
ft^3 (cubic foot)	in.^{2}	1728	5.787×10^{-4}
	cm^3	2.832×10^4	3.531×10^{-5}
	m^3	2.832×10^{-2}	35.31
hp (horsepower)	W (watt)	745.7	1.341×10^{-3}
in. (inch)	ft (foot)	0.0833	12
	cm (centimeter)	2.540	0.3937
	m (meter)	0.0254	39.37

(*continued*)

A.4. (*Continued*)

To Convert	Into	Multiply by	Conversely, Multiply by
$in.^2$ (square inch)	ft^2	6.945×10^{-3}	144
	cm^2	6.452	0.1550
	m^2	6.452×10^{-4}	1550
$in.^3$ (cubic inch)	ft^3	5.787×10^{-4}	1.728×10^3
	cm^3	16.387	6.102×10^{-2}
	m^3	1.639×10^{-5}	6.102×10^4
kg (kilogram)	lb (weight)	2.2046	0.4536
	slug	0.06852	14.594
	g (gram)	10^3	10^{-3}
kg/m^2	$lb/in.^2$ (weight)	1.422×10^{-3}	703.0
	lb/ft^2 (weight)	0.2048	4.882
	g/cm^2	10^{-1}	10
m (meter)	in. (inch)	39.371	2.540×10^{-2}
	ft (foot)	3.2808	0.30481
	cm (centimeter)	10^2	10^{-2}
m^2 (square meter)	$in.^2$	1550	6.452×10^{-4}
	ft^2	10.764	9.290×10^{-2}
	cm^2	10^4	10^{-4}
m^3 (cubic meter)	$in.^3$	6.102×10^4	1.639×10^{-5}
	ft^3	35.31	2.832×10^{-2}
	cm^3	10^6	10^{-6}
microbar ($dynes/cm^2$)	$lb/in.^2$	1.4513×10^{-5}	6.890×10^4
	lb/ft^2	2.090×10^{-3}	478.5
	N/m^2 (Pa)	10^{-1}	10
Np (neper)	dB (decibel)	8.686	0.1151
N (newton)	lb (force)	0.2248	4.448
	dynes	10^5	10^{-5}
N/m^2 (pascal, Pa)	$lb/in.^2$ (force)	1.4513×10^{-2}	6.890×10^3
	lb/ft^2 (force)	2.090×10^{-2}	47.85
	$dynes/cm^2$	10	10^{-1}
lb (force) (pound)	N (newton)	4.448	0.2248
lb (weight) (pound)	slug	0.03108	32.17
	kg (kilogram)	0.4536	2.2046
$lb/in.^2$ (weight)	lb/ft^2 (weight)	144	6.945×10^{-3}
	kg/m^2	703	1.422×10^{-3}
$lb/in.^2$ (force)	lb/ft^2 (force)	144	6.945×10^{-3}
	N/m^2 (Pa)	6894	1.4506×10^{-4}
lb/ft^2 (weight)	$lb/in.^2$ (weight)	6.945×10^{-3}	144
	g/cm^2	0.4882	2.0482
	kg/m^2	4.882	0.2048
lb/ft^2 (force)	$lb/in.^2$ (force)	6.945×10^{-3}	144
	N/m^2 (Pa)	47.85	2.090×10^{-2}
Slugs	lb (weight)	32.17	3.108×10^{-2}
	kg (kilogram)	14.594	6.852×10^{-2}
W (watt)	hp (horsepower)	1.341×10^{-3}	745.7

Appendix B
Bessel Functions

B.1 The Bessel Differential Equation

The Bessel differential equation of order n, expressed as

$$\left[x^2\frac{d^2}{dx^2} + x\frac{d}{dx} + (x^2 - n^2)\right] f(x) = 0$$

carries the solutions consisting of (1) the Bessel functions of the first kind $J_n(x)$ for all x, and (2) the Bessel functions of the second kind $Y_n(x)$ (also called Neuman functions), and the Bessel functions of the third kind $H_n^{(1)}(x)$ and $H_n^{(2)}(x)$ for all x greater than zero.

B.2 Relationships Between Solutions

$$H_n^{(1)} = J_n + iY_n$$
$$H_n^{(2)} = J_n - iY_n$$
$$J_{-n} = (-1)^n J_n$$
$$Y_{-n} = (-1)^n Y_n$$

B.3 Series Expansions for J_0 and J_1

$$J_0 = 1 - \frac{x^2}{2^2} + \frac{x^4}{2^2 \cdot 4^2} - \frac{x^6}{2^2 \cdot 4^2 \cdot 6^2} + \cdots$$
$$J_1 = \frac{x}{2} - \frac{2x^3}{2 \cdot 4^2} + \frac{3x^5}{2 \cdot 4^2 \cdot 6^2} + \cdots$$

B.4 Approximations for Small Argument $x < 1$

$$J_0 \to 1 - \frac{x^2}{4}$$

$$J_1 \to \frac{x}{2} - \frac{x^3}{16}$$

$$Y_0 \to \frac{2}{\pi} \ln x$$

$$Y_1 \to -\frac{2}{\pi}\frac{1}{x}$$

B.5 Approximations for Large Arguments $x > 2\pi$

$$J_n \to \sqrt{\frac{2}{\pi x}} \cos\left(x - \frac{n\pi}{2} - \frac{\pi}{4}\right)$$

$$Y_n \to \sqrt{\frac{2}{\pi x}} \sin\left(x - \frac{n\pi}{2} - \frac{\pi}{4}\right)$$

$$H_n^{(1)} \to \sqrt{\frac{2}{\pi x}}\, e^{i\left(x - \frac{n\pi}{2} - \frac{\pi}{4}\right)}$$

$$H_n^{(2)} \to \sqrt{\frac{2}{\pi x}}\, e^{-i\left(x - \frac{n\pi}{2} - \frac{\pi}{4}\right)}$$

B.6 Recursion Relations

The relations listed below hold true for C being any of the Bessel functions of the first, second, or third kind or for linear combinations of these functions.

$$C_{n+1} + C_{n-1} = \frac{2n}{x} C_n$$

$$\frac{dC_0}{dx} = -C_1$$

$$\frac{dC_n}{dx} = \frac{1}{2}\left(C_{n-1} - C_{n+1}\right)$$

$$\frac{d}{dx}\left(x^n C_n\right) = x^n C_{n-1}$$

$$\frac{d}{dx}\left(\frac{1}{x^n}\right) = -\frac{1}{x^n} C_{n+1}$$

B.7 Modified Bessel Functions

$$I_n(x) = i^{-n} J_n(ix)$$

$$I_0(x) = J_0(ix) = 1 + \frac{x^2}{2^2} + \frac{x^4}{2^2 \cdot 4^2} + \frac{x^6}{2^2 \cdot 4^2 \cdot 6^2} + \cdots$$

$$\int x I_0(x) dx = x I_1(x)$$

$$\int I_1(x) dx = I_0(x)$$

$$I_0(x) - I_2(x) = \frac{2}{x} I_1(x)$$

B.8 Tables of Bessel Functions, Zeros, and Inflection Points

B.1. Bessel Functions or Order 0, 1, and 2

x	$J_0(x)$	$J_1(x)$	$J_2(x)$	$I_0(x)$	$I_1(x)$	$I_2(x)$
0.0	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
0.2	0.9900	0.0995	0.0050	0.0100	0.1005	0.0050
0.4	0.9604	0.1960	0.0197	0.0404	0.2040	0.0203
0.6	0.9120	0.2867	0.0437	1.0921	0.3137	0.0464
0.8	0.8463	0.3688	0.0758	1.1665	0.4329	0.0843
1.0	0.7652	0.4401	0.1149	1.2661	0.5652	0.1358
1.2	0.6711	0.4983	0.1593	1.3937	0.7147	0.2026
1.4	0.5669	0.5419	0.2074	1.5534	0.8861	0.2876
1.6	0.4554	0.5699	0.2570	1.7500	1.0848	0.3940
1.8	0.3400	0.5815	0.3061	1.9895	1.3172	0.5260
2.0	0.2239	0.5767	0.3528	2.2796	1.5906	0.6890
2.2	0.1104	0.5560	0.3951	2.6292	1.9141	0.8891
2.4	+0.0025	0.5202	0.4310	3.0492	2.2981	1.1111
2.6	−0.0968	0.4708	0.4590	3.5532	2.7554	1.4338
2.8	−0.1850	0.4097	0.4777	4.1575	3.3011	1.7994
3.0	−0.2601	0.3391	0.4861	4.8808	3.9534	2.2452
3.2	−0.3202	0.2613	0.4835	5.7472	4.7343	2.7884
3.4	−0.3643	0.1792	0.4697	6.7848	5.6701	3.4495
3.6	−0.3918	0.0955	0.4448	8.0278	6.7926	4.2538
3.8	−0.4026	+0.0128	0.4093	9.5169	8.1405	5.2323
4.0	−0.3971	−0.0660	0.3641	11.302	9.7594	6.4224
4.2	−0.3766	−0.1386	0.3105	13.443	11.705	7.8683
4.4	−0.3423	−0.2028	0.2501	16.010	14.046	9.6259
4.6	−0.2961	−0.2566	0.1846	19.097	16.863	11.761
4.8	−0.2404	−0.2985	0.1161	22.794	20.253	14.355
5.0	−0.1776	−0.3276	+0.0466	27.240	24.335	17.505
5.2	−0.1103	−0.3432	−0.0217	32.584	29.254	21.332
5.4	−0.0412	−0.3453	−0.0867	39.010	35.181	25.980
5.6	+0.0270	−0.3343	−0.1464	46.738	42.327	31.621
5.8	0.0917	−0.3110	−0.1989	56.039	50.945	38.472

(*Continued*)

B.1. (*Continued*)

x	$J_0(x)$	$J_1(x)$	$J_2(x)$	$I_0(x)$	$I_1(x)$	$I_2(x)$
6.0	0.1507	−0.2767	−0.2429	67.235	61.341	46.788
6.2	0.2017	−0.2329	−0.2769	80.717	73.888	56.882
6.4	0.2433	−0.1816	−0.3001	96.963	89.025	69.143
6.6	0.2740	−0.1250	−0.3119	116.54	107.31	84.021
6.8	0.2931	−0.0652	−0.3123	140.14	129.38	102.08
7.0	0.3001	−0.0047	−0.3014	168.59	156.04	124.01
7.2	0.2951	+0.0543	−0.2800	202.92	188.25	150.63
7.4	0.2786	0.1096	−0.2487	244.34	227.17	182.94
7.6	0.2516	0.1592	−0.2097	294.33	274.22	222.17
7.8	0.2154	0.2014	−0.1638	354.68	331.10	269.79
8.0	0.1716	0.2346	−0.1130	427.57	399.87	327.60

B.2. Zeros of $J_m : J_m(j_{mn}) = 0$.

j_{mn}

	n					
m	0	1	2	3	4	5
0	—	2.40	5.52	8.65	11.79	14.93
1	0	3.83	7.02	10.17	13.32	16.47
2	0	5.14	8.42	11.62	14.80	17.96
3	0	6.38	9.76	13.02	16.22	19.41
4	0	7.59	11.06	14.37	17.62	20.83
5	0	8.77	12.34	15.70	18.98	22.22

B.3. Inflection Points of $J_m : (dJ_m/dx)_{j'_{mn}} = 0$.

j'_{mn}

	n				
m	1	2	3	4	5
0	0	3.83	7.02	10.17	13.32
1	1.84	5.33	8.54	11.71	14.86
2	3.05	6.71	9.97	13.17	16.35
3	4.20	8.02	11.35	14.59	17.79
4	5.32	9.28	12.68	15.96	19.20
5	6.41	10.52	13.99	17.31	20.58

Appendix C
Using Laplace Transforms to Solve Differential Equations

C.1 Introduction

Not only all aspects of the of Laplace transforms will be presented, but also enough of the characteristics will be presented, discussed here to help the reader understand and appreciate the elegance and powerfulness of the procedures that can be effectively applied to solve linear differential equations. The essential idea of Laplace transforms is that they are used to convert a differential equation into an algebraic one in order to solve the differential equation.

Let $f(t)$ be a function of t for $t > 0$. Its Laplace transform is defined as

$$F(s) = \int_0^\infty f(t)e^{-st}\,dt = \mathcal{L}[f(t), s] \tag{C.1}$$

where s is a complex variable which can be expressed as

$$s = \alpha + i\beta \tag{C.2}$$

The function $f(t)$ is Laplace transformable for $\alpha > 0$ if

$$\lim_{T\to\infty} \int_0^T |f(t)|e^{-\alpha t}\,dt < \infty$$

If $f(t) = A$ (a constant), for $t > 0$, the Laplace transform of the constant A is

$$\mathcal{L}[A] = \int_0^\infty Ae^{-st}\,dt = -\frac{1}{s}\,e^{-st}\Big|_0^\infty = \frac{A}{s} = F(s)$$

We therefore have the Laplace transform pair

$$f(t) = A \Leftrightarrow F(s) = \frac{A}{s}$$

If $f(t) = e^{-at}$ for $t > 0$, then the Laplace transform of this exponential function is

$$\mathcal{L}e^{-at} = \int_0^\infty e^{-at}e^{-st}\,dt = \int_0^\infty e^{-(s+a)t} = \frac{A}{s+a}$$

and hence

$$f(t) = e^{-at} \Leftrightarrow F(s) = \frac{1}{s+a}$$

C.2 Laplace Transforms of Derivatives

Consider a function $f(t)$ and its derivative $df(t)/dt$ that are both Laplace transformable. If the function $f(t)$ has the Laplace transform $F(s)$, then

$$\mathcal{L}\frac{df(t)}{dt} = sF(s) - f(0+) \tag{C.3}$$

Equation (C.3) is readily proven by using integration by parts, i.e.,

$$F(s) = \int_0^\infty f(t)e^{-st}\,dt$$

Set

$$u = f(t), \qquad du = \frac{df(t)}{dt}\,dt$$

$$dv = e^{-st}\,dt, \qquad v = -\frac{1}{s}e^{-st}$$

It then follows that

$$F(s) = uv|_0^\infty - \int_0^\infty v\,du = -\frac{1}{s}f(t)e^{-st}\Big|_0^\infty + \frac{1}{s}\int_0^\infty \frac{df(t)}{dt}e^{-st}\,dt$$

$$= \frac{f(0+)}{s} + \frac{1}{s}\int_0^\infty \frac{df(t)}{dt}e^{-st}\,dt$$

or

$$\int_0^\infty \frac{df(t)}{dt}e^{-st}\,dt = sF(s) - f(0+)$$

The term $f(0+)$ represents the value of $f(t)$ as $t \to 0$ from the positive side. In a similar manner, it can be demonstrated that

$$\mathcal{L}\frac{d^2f(t)}{dt^2} = s^2F(s) - sf(0+) - f(0+) \tag{C.4}$$

C.3 Solving Differential Equations

We discuss here a differential equation that is of greatest interest in treatment of vibration problems:

$$m\ddot{x}(t) + C\dot{x}(t) + kx(t) = f(t) \tag{C.5}$$

which also can be expressed as

$$\ddot{x} + 2\xi\omega_n\dot{x}(t) + \omega_n^2 x(t) = f_\alpha(t) \tag{C.6}$$

We now apply Equations (C.3) and (C.4)–(C.6) to obtain

$$s^2X(s) - sx(0+) - \dot{x}(0+) + 2\xi\omega_n[sX(s) - x(0+)] + \omega_n^2 X(s) = F_\alpha(s)$$

Solving for $X(s)$:

$$X(s) = \frac{F_\alpha(x) + x(0+)(s + 2\xi\omega_n) + \dot{x}(0+)}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{C.7}$$

Equation (C.7) can be written in the form

$$X(s) = \frac{A(s)}{B(s)} \tag{C.8}$$

Equation (C.7) and its variation (C.6) have been derived from a second-order equation, and the techniques we describe below are valid for any order function.

For equations with simple roots, we can write Equation (C.8) in the form

$$X(s) = \frac{A(s)}{B(s)} = \frac{A(s)}{(s + a_1)(s + a_2)\cdots(s + a_n)} \tag{C.9}$$

It is assumed that $B(s)$ is of higher order than $A(s)$ throughout the discussion below. We can expand Equation (C.9) through the use of partial fraction expansions, which will result in

$$X(s) = \frac{C_1}{s + a_1} + \frac{C_2}{s + a_2} + \cdots + \frac{C_n}{s + a_n} \tag{C.10}$$

where

$$C_k = \lim_{s \to -a_k} (s + a_k)\frac{A(s)}{B(s)} \tag{C.11}$$

Example Problem 1

Find the inverse Laplace transform of the function

$$X(s) = \frac{13(s + 3)}{s(s + 1)(s + 2)}$$

Solution

The roots of the numerator (for example, $s = -3$) are called *zeros* of the function. The roots of the denominator (viz., 0, -1, -2) are called *poles*. A partial fraction expansion of the poles is written as follows:

$$\frac{13(s + 3)}{s(s + 1)(s + 2)} = 13\left(\frac{C_1}{s} + \frac{C_2}{s + 1} + \frac{C_3}{s + 2}\right)$$

The *residues* C_1, C_2, and C_3 are obtained from Equation (C.11) as follows:

$$C_1 = \lim_{s\to 0} \frac{s+3}{(s+1)(s+2)} = \frac{3}{(1)(2)} = \frac{3}{2}$$

$$C_2 = \lim_{s\to -1} \frac{s+3}{s(s+2)} = \frac{2}{(-1)(1)} = -2$$

$$C_3 = \lim_{s\to -2} \frac{s+3}{s(s+1)} = \frac{1}{(-2)(-1)} = \frac{1}{2}$$

Hence

$$X(s) = 13\left(\frac{3/2}{s} + \frac{-2}{s+1} + \frac{1/2}{s+2}\right)$$

Applying the transform pairs we derived in Section C.2,

$$x(t) = 13\left(\frac{3}{2} - 2e^{-t} + \frac{1}{2}e^{-2t}\right)$$

When we know the form of the inverse transform above we can derive the residue graphically by inspection. The method for doing this is simply divide the product of all vectors from the zeros to the pole whose residue is being determined by the product of the vectors from all the other poles to the pole under consideration. This technique is shown in Figure C.1. In Figure C.1(a), C_1 is derived as

$$C_1 = \frac{3\angle 0^\circ}{(2\angle 0^\circ)(1\angle 0^\circ)} = 3/2$$

In Figure C.1(b), we get

$$C_2 = \frac{2\angle 0^\circ}{(1\angle 0^\circ)(1\angle 180^\circ)} = \frac{2}{(1)(-1)} = -2$$

and from Figure C.1(c):

$$C_3 = \frac{1\angle 0^\circ}{(1\angle 180^\circ)(2\angle 180^\circ)} = \frac{1}{(-1)(-2)} = 1/2$$

Note that the numbers obtained from the use of the graphical residue technique are the same as those derived from Equation (C.11). The sign of the residue can be readily established by simply counting the number of poles and zeros to the right in the s plane of the pole whose residue is being determined. If the number is even, the sign of the residue is positive; and if the number is odd, the sign is negative.

Example Problem 2

Consider the function

$$X(s) = \frac{8(s+2)(s+4)}{s(s+1)(s+3)}$$

Find the inverse Laplace transform.

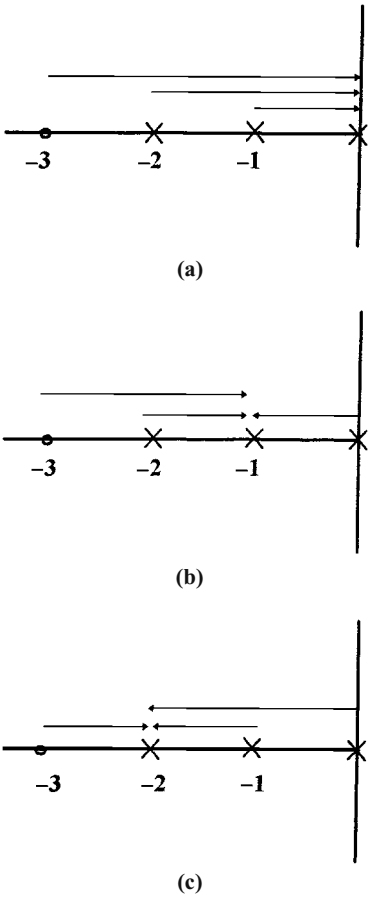


FIGURE C.1. Example of the graphical residue technique.

Solution

Use the pole-zero pattern of Figure C.2 to apply the graphical technique. The inverse transform can be written down by inspection as

$$x(t) = 8\left[\frac{(2)(4)}{(1)(3)} - \frac{(1)(3)}{(1)(2)}e^{-t} - \frac{(1)(1)}{(2)(3)}e^{-3t}\right] = 8\left[\frac{8}{3} - \frac{3}{2}e^{-t} - \frac{1}{6}e^{-3t}\right]$$

C.4 Equations with Multiple-Order Roots

Consider a function $X(s)$ that has multiple-order roots of the form

$$X(s) = \frac{A(s)}{B(s)} = \frac{A(s)}{(s + a_1)^m(s + a_2)\cdots(s + a_n)}$$

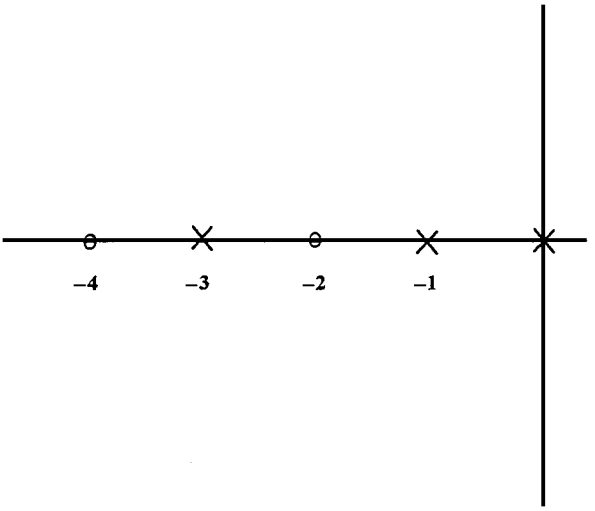


FIGURE C.2. Pole-zero pattern for Example Problem 2.

The partial fraction expansion of the function must be of the format

$$\frac{A(s)}{B(s)} = \frac{C_{11}}{(s+a_1)^m} + \frac{C_{12}}{(s+a_1)^{m-1}} + \cdots \frac{C_{1m}}{s+a_1} + \frac{C_2}{s+a_2} + \cdots + \frac{C_n}{s+a_n}$$

The residues $C_2, C_3, \ldots, C_n$ are determined in the same manner as was described in the preceding section but the coefficients $C_{11}, C_{12}, \ldots, C_{1m}$ require special treatment. They are evaluated from the following expression:

$$C_{1j} = \frac{1}{(j-1)!}\frac{d^{j-1}}{ds^{j-1}}(s+a_1)^m \frac{A(s)}{B(s)} \tag{C.12}$$

which is evaluated at $s = -a_1$. Then the inverse Laplace is obtained from the transform relation:

$$\mathcal{L}^{-1}\left[\frac{1}{(s+a)^j}\right] = \frac{t^{j+1}}{(j-1)!}e^{-at} \tag{C.13}$$

The following example problem will demonstrate the procedure.

Example Problem 3

Find the inverse Laplace transform for

$$X(s) = \frac{s+1}{(s+2)^2(s+3)} = \frac{C_{11}}{(s+2)^2} + \frac{C_{12}}{s+2} + \frac{C_2}{s+3}$$

Solution

Use Equation (C.11) to obtain

$$C_{11} = \left.\frac{s+1}{s+3}\right|_{s=-2} = -1, \quad C_2 = \left.\frac{s+1}{(s+2)^2}\right|_{s=-3} = -2$$

Equation (C.12) is now applied to obtain the coefficient C_{12}:

$$C_{12} = \frac{1}{(2-1)!}\frac{d^{2-1}}{ds^{2-1}}\left(\frac{s+1}{s+3}\right) = \frac{d}{ds}\left(\frac{s+1}{s+3}\right) = \left.\left(\frac{1}{s+3} - \frac{s+1}{(s+3)^2}\right)\right|_{s=-2} = 2$$

which now results in

$$X(s) = \frac{-1}{(s+2)^2} + \frac{2}{s+2} - \frac{2}{s+3}$$

The inverse Laplace transform now becomes

$$x(t) = -te^{-2t} + 2e^{-2t} - 2e^{-3t}$$

C.5 Equations with Complex Roots

Let us consider the case of finding the inverse Laplace transform of a function of the form

$$X(s) = \frac{(s+b_1)(s+b_2)}{[(s+\alpha)^2+\beta^2](s+a_1)(s+a_2)(s+a_3)}$$

We could evaluate the inverse transform using a partial fraction expansion, but this can be quite a complicated procedure especially where complex roots are entailed. In order to deal with this problem, it is preferable to use the relation between the transform pairs

$$\frac{as+b}{(s+\alpha)^2+\beta^2} \Leftrightarrow 2Me^{-\alpha t}\cos(\beta t + \phi)$$

where M and ϕ are unknowns that depend on a and b.

In order to demonstrate that M and ϕ can be evaluated by the graphical approach, suppose that the above function has the pole-zero pattern of Figure C.3(a). Only the residues of the complex poles at $-\alpha \pm i\beta$ will be determined. Using the graphical approach for simple roots, we get from using Figure C.3(b):

$$X(s) = \frac{\dfrac{(z_1e^{i\theta_2})(z_2e^{i\theta_6})}{(P_1e^{i\theta_1})(P_2e^{i\theta_3})(P_3e^{i\theta_4})(P_4e^{i\theta_5})}}{s+\alpha-i\beta} + \frac{\dfrac{(z_1e^{-i\theta_2})(z_2e^{-i\theta_6})}{(P_1e^{-i\theta_1})(P_2e^{-i\theta_3})(P_3e^{-i\theta_4})(P_4e^{-i\theta_5})}}{s+\alpha+i\beta} + \text{additional terms} \tag{C.14}$$

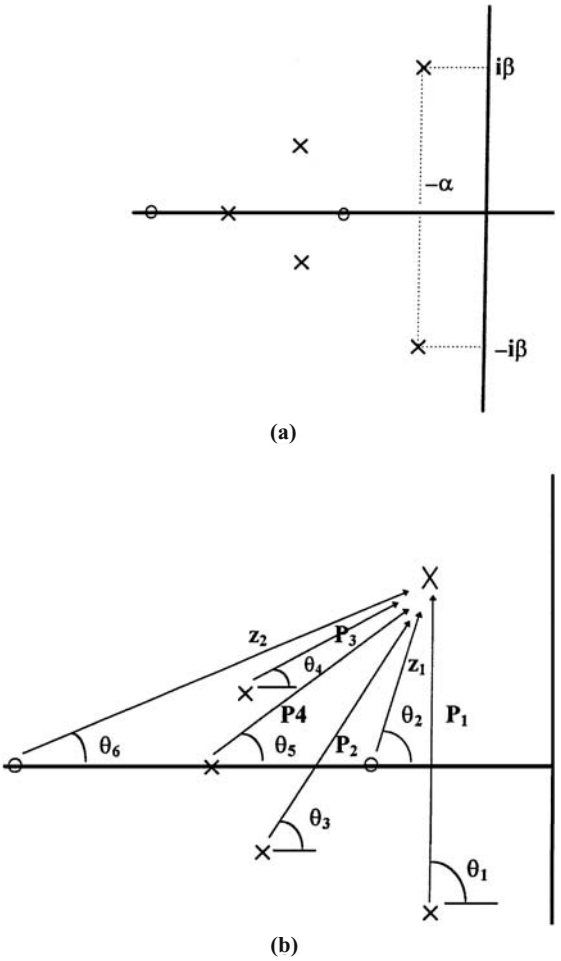


FIGURE C.3. Graphical residue approach for complex roots.

We observe that the coefficient of the second term, which represents the residue at $s = -\alpha - i\beta$ is the complex conjugate of the coefficient of the first term that is the residue at $s = -\alpha + i\beta$. Equation (C.14) can be expressed as

$$X(s) = \frac{Me^{i\phi}}{s + \alpha - i\beta} + \frac{Me^{-i\phi}}{s + \alpha + i\beta} + \text{additonal terms}$$

where

$$M = \frac{z_1 z_2}{P_1 P_2 P_3 P_4}, \quad \phi = \theta_2 + \theta_6 - (\theta_1 + \theta_3 + \theta_4 + \theta_5)$$

Then

$$\begin{aligned} x(t) &= M(e^{-\alpha t+i\beta t+i\phi} + e^{-i\alpha t-i\beta t-i\phi}) + \text{addtional terms} \\ &= Me^{-\alpha t}(e^{i(\beta t+i\phi)} + e^{-i(\beta t+\phi)}) + \text{additonal terms} \\ &= 2Me^{-\alpha t}\frac{(e^{i(\beta t+i\phi)} + e^{-i(\beta t+\phi)})}{2} + \text{additonal terms} \\ &= 2Me^{-\alpha t}\cos(\beta t + \phi) + \text{additional terms} \end{aligned}$$

Here M and ϕ can be determined graphically using Figure C.3. The use of the graphical residue technique for equations with complex roots is extremely powerful since the inverse transform can be determined by inspection.

Example Problem 4

Given

$$X(s) = \frac{s+2}{s(s^2+4s+5)}$$

find the inverse Laplace transform.

Solution

Figure C.4 shows the function. The residue at the pole at the origin is found by using Figure C.4(b), and the complex poles are derived by using Figure B.4(c). The ensuing function is

$$x(t) = \frac{2}{\sqrt{5}\sqrt{5}} + 2 \times \frac{1}{2\sqrt{5}}e^{-2t}\cos(t+\phi)$$

wherein

$$\begin{aligned} \phi &= (90^\circ) - (90^\circ + \theta) \\ \theta &= \left(\pi - \tan^{-1}\tfrac{1}{2}\right) \end{aligned}$$

Example Problem 5

From vibration theory $X(s)$ [cf. Equation (20.12)] is of the form

$$\begin{aligned} X(s) &= x_0\frac{s+2\xi\omega_n}{s^2+2\xi\omega_n s+\omega_n^2} \\ &= \frac{x_0(s+2\xi\omega_n)}{(s+\xi\omega_n+i\omega_n\sqrt{1-\xi^2})(s+\xi\omega_n-i\omega_n\sqrt{1-\xi^2})} \end{aligned}$$

Determine the inverse transform.

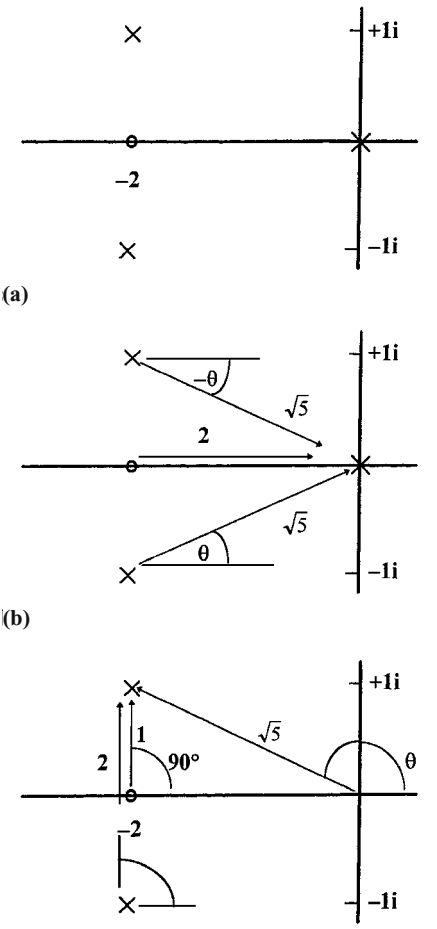


FIGURE C.4. Pole-zero diagrams for Example Problem 3.

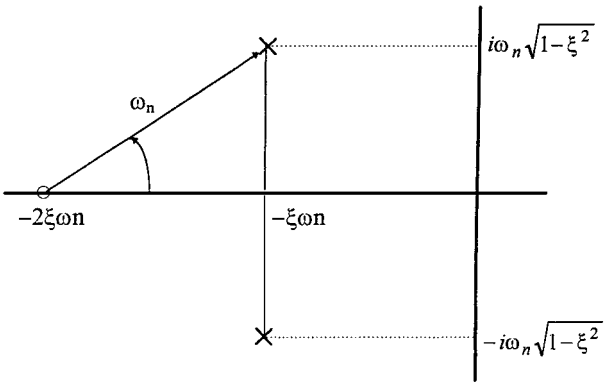


FIGURE C.5. Pole-zero diagram for Example Problem 4.

Solution

From Figure C.5 the inverse transform can be evaluated to yield

$$x(t) = x_0 \left[\frac{\omega_n}{\omega_n \sqrt{1-\xi^2}} e^{-\zeta \omega_n t} \cos\left(\omega_n \sqrt{1-\xi^2} t + \phi \right) \right]$$

where

$$\phi = \theta - \pi/2$$
$$\theta = \cos^{-1} \xi$$

and we now have

$$x(t) = \frac{x_0}{\sqrt{1-\xi^2}} e^{-2\omega_n t} \cos\left(\omega_n \sqrt{1-\xi^2} t - \pi/2 + \cos^{-1} \xi \right)$$

Index

A-scan, 473
A-weighting, 52–54
Absorbent effects, growth of sound with, 251–252
Absorption
  in seawater, parametric variation of, 419–421
  spherical spreading combined with, 421
Absorption coefficients, sound, 249, 251
Absorption losses, 416
Academy of Music, Philadelphia, 266
Accelerometers, 611–612
Accordion, 541, 542–543
Acoustic analogues, 151–168
Acoustic barriers, 310–314
Acoustic element, lumped, 145
Acoustic energy, 248–249
Acoustic energy density, 258
Acoustic equations, derivation of, 25–29
Acoustic filters, 158–165
Acoustic hemostasis, 504
Acoustic impedance, 151
  distributed, 154–155
  lumped, 152–154
Acoustic lenses, 494–495
Acoustic measurements, 173–209
Acoustic microscope, 42–43
Acoustic ohm, 152
Acoustic propagation constant, 499
Acoustic reflex, 219
Acoustical instruments, characteristics of, 173–174
Acoustical shadow, 44–45
Acoustical surgery, 504
Acoustics, 2–3, 13
  architectural, 243–278
  of enclosed spaces, 243–278
  fundamentals of, 13–28
  future of, 10–11
  history of, 1–11
  importance of, 15
  musical, 509
  nonlinear, 617–626
  term, 4
  underwater, 409–439
Active isolation and absorption systems, 606
Active noise control, 404–405
Addition method for measuring sound power level, 207–208
Adiabatic process, 120
Adiabatic relaxation time, 425
Aeolian tones, 386
Afternoon effect, 423
Air compressor noise, 370–371
Air-reed instruments, 537
Air shroud silencer nozzles, 394–395
Aircraft noise, rating of, 332–335
Aliasing, 195
Alpha factor, 341
Alternation method for measuring sound power level, 207
Alto clef, 512
Amplitude modulation (AM), 208
Analogues, acoustic, 151–154
Angular frequency, 13
Annoyance, 319
ANSYS finite element program , 610
Antinodes, 77
Antiresonance, 134
Arau, Higini, 266
Architectural acoustics, 243–278
Aristotle, 2–3
Array gain, 426
Articulation index, 227–230
Associated liquids, 451
Audio range of frequencies, 45–47
Audiometer, 222

Auditoriums, design of, 261–278
  outdoor, 271–276
Augmented reality system, 505
Autocorrelation, mean-square value and, 613
Autocorrelation function (ACF), 276
Avogadro's number, 19
Axial flow fans, 360–361
Azimuthal nodal circles, 117

B-scan, 474
B-weighting, 52–54
Bacon, Sir Francis, vii
Bagpipe, 545
Balance noise criterion (NCB) curves, 327–328
Ball bearing noise, 379–382
Band, 562–563, 564
Band pass filters, 162–164
  noise measurement and, 189–193
Band shells, design of, 271–276
Bandwidth, 46
Banjo, 525, 527
Barrier insertion loss, approximations for, 315–316
Barriers, 310–316
  in free fields, 314–315
  walls and enclosures and, 281–316
Bars, vibrating, see Vibrating bars
Bass drum, 555–556
Bass trombone, 551
Bassoon, 546–547
Bats, 238, 443–444
B.C.S. theory, 457, 458
Beat frequency, 35–36, 60
Bell lyre, 553
Belt drive noise, 385
Bekésy, Georg von, 9
Bell Telephone Laboratories, 8, 9, 245
Benaroya Hall, 270–271
Beranek, Leo L., 9
Bernoulli's principle, 541n
Bessel functions, 116, 633–636
  modified, 126n, 635
Beyer, Robert T., 9
Bible, 1
Blade frequency, 361
Blade rate component, 370–371
Blower noise, 359–366
Blu-ray, 574, 575
Boethius, Sverinus, 2
Bolt, Richard H, 9.
Boltzmann constant, 19
Boom cars, 344, 348–349
Boston Symphony Hall, 263–264
Botta, Mario, 268
Boundaries, reflection of waves at, 74–75
Bowed-string instruments, 529–533
Boyle, Robert, 4
Brasses, 550–551
Bugle, 550
Bulk viscosity, 419
Burgers equation, 621–622

C-scan, 476
C-weighting, 52–54
Calliope, 538
Carillon, 554
Carnegie Hall, 266
Castanets, 556
Catgut Acoustical Society, 533–534
Cavitation, 451–453
Cavitation noise, 452
Cavitation threshold intensity, 452–453
Celesta, 553
Cembalo, 527
Central Artery Tunnel Project (CA/T), 351–352
Central hearing loss, 221
Centrifugal fans, 359–361
Ceramic microphones, 174–175
Cetaceans, 444
Chain drive noise, 382–395
Characteristic mechanical impedance, 133
Charge sensitivity, 612
Chilowsky, Constantin, 7
Chimes, 514
Chladni, Ernst F. F., 4–5
Chorusing, 559
Chrysippus, 2
Clamped-free bar, 106, 107
Clarinet, 544
Clavichord, 527
Closed-ended pipes, resonances in, 132–134
Cochlea, 215–219
Cochlear microphonic effect, 219
Coincidence effect, 286–287
Collision number, 446
Color schlieren photography, 496
Community noise, evaluation of, 344–345
Community response, 319
Composite noise rating (CNR), 333
Compressive forces, 91
Computers, integration of measurement functions in, 208–209
Concert halls, design of, 261–271
Condenser microphones, 175–176
Conductive hearing loss, 221

Conservation
of energy, 25
of mass, 20–22
of momentum, 22–25
Construction noise, general, 350–352
Contact ratio (CR) for gears, 376–377
Continuity equation, 21
expansion of, 26
Contra bassoon, 547
Contrast agents, 501–502
Cooley-Tukey algorithm, 195
Cornet, 550, 551
Correlated sound waves, 58–60
Corti, organ of, 217–218
Council of European Communities, 346
Coupled quantum particles, 456–458
Creep, 261
Crescendo, 520
Crest factor capability, 174
Critical frequency, 286–288
Crossover networks, 578
Crum, Lawrence A., 10
Cylindrical spreading, 417
Curie, Pierre, 7, 459
Curie, Paul-Jacques, 7, 459
Cymbal, 556

Damping, 585–586, 606–610
internal, 608–609
Damping factor, 123–124
Damping materials, 609
Damping mechanisms, 609–610
Damping ratio, 587
Dashpot, 585
Data acquisition systems, 208
Data windows, 196–198
Day-night average sound levels, yearly, 338
Day-night equivalent sound pressure level, 57, 329–331
da Vinci, Leonardo, 3
Dead rooms, decay of sound in, 254–256
Dead spots, 262
Deafness, sensorineural, 221
Decibels, 47–49
averaging, 49–52
Decoupling, 603
Decrescendo, 580
Deep sound channel, 424
Deep sound-channel axis, 414
Delay lines, ultrasonic, 484
Detection threshold, 430
Detuning, 603
Diagnostic uses of ultrasound, 498–503
Diatomic molecules, 445
Diffraction, 44–45
Diffuse fields, 178, 244
Digital recording, 573–575
Direct drive hearing system (DDHS), 236–237
Directional characteristic, 200
Directivity factor, 258
Discord, harmony and, 520–521
Discrete Fourier transform (DFT), 194
Displacement amplitudes, 16
Dissipative function, 621–622
Dissipative mufflers and silencers, 398, 403–404
Distortion products, 219
Distributed acoustic impedance, 154–155
Doppler effect, 39–40
Dosimeters, 187–189
Dorothy Chandler Pavilion, 264
Double bass, 528–533
Double mechanical-reed instruments, 545–550
Double-panel partitions, 289–292
Drug therapy, 237
Ducted source systems, 165–168
Ducts, gaseous flows in, 396–398
Dulcimer, 536
Duple meter, 517
DVD formats, 574–575
Dynamic microphones, 174
Dynamic range, 174

Ear, human, 213–217
Ear sensitivity, 222–225
Eardrum, 214–215
Early decay time (EDT10), 258
Eastman Theater, 264
Echo, 40, 267, 409
Echolocation, 238
Echo-ranging (active) sonar, 410
Echocardiography, 499
Echoencephalography, ultrasonic, 500
Effective perceived noise level (EPNL), 324–325
Eigenstates, 454
Einstein, Albert, 454
Ekos device, 505
Elastomeric mounts, 604
Electret microphones, 174–175
Electric acoustic guitar, 556
Electric guitar, 556
Electric motor noise, 366
Electrical and electronic instruments, 521, 522, 556–561
Electromechanical coupling factor, 465
Electron-acoustic image converters, 492–494
Electronic organ, 556–559

Electrostatic speakers, 580
Electrostrictive effect, 461–462
Electrostrictive materials, 425
Elevation dimension, 470
Empirical dosage response relationship, 345
Emulsification and flow enhancement, 491
Enclosed spaces, acoustics of, 243–271
Enclosures, 305–309
  small, 309–310
  walls and barriers and, 281–316
Energy, conservation of, 25
Energy density, 65–67
  instantaneous, 66
Energy flux density, 411
  of waves, 434
Environmental noise, 319
  performance indices for, 55–57
Environmental Protection Agency (EPA), 320
Equal energy hypothesis, 345
Equal noisiness contours, 323
Equivalent simple piston, 120
Equivalent sound levels, 56–57, 329–330, 335
Ergodic processes, 614
Euler, Leonhard, 6, 521
European Union (EU), 350–351
Eustachian tube, 214–215
Evanescent mode, 141
Evoked otoacoustic emission, 219
Excessive loudness, 261
Exchange rate, 188n
Excitation
  harmonic, 593–596
  by impulse, 597–598
  motion, 600–603
Extracorporeal shock wave lithotripsy, 503–504
Eyring equation, 255

Fan laws, 365–366
Fan noise, 359–366
  characteristics of, 361–362
Far field effect, 62
Fast Fourier transform (FFT) technique, 194–196
Federal Aviation Administration (FAA), 320
Federal Highway Administration (FHWA), 335–339
Federal Interagency Committee on Noise (FICON), 346
Fermat, Pierre de, 5
Ferroelectrics, 461
Field incidence mass law, 284–285
Filter networks, 162–165
Filter response, 189
Filters, acoustic, 158–165
Finite element analysis (FEA), 610–611
Finite roadway segment adjustment, 342
Finite strings, 83–84
Fire sensing, ultrasonic, 487–488
First harmonic, 76
Fixed-fixed bar, 92, 93
Flageolet, 539
Flanking, 294
Flared pipes, 135
Flat notes, 512
Flaw detection, ultrasonic methods for, 481–483
Fletcher, Harvey, 8–9
Flowmeter, ultrasonic, 486–487
Fluid flow equations, 19–20
Fluids, thermodynamic states of, 18–19
Flute, 539–540
Flutter echo, 261
Fohi, 2
Forced vibrations, 593–598
  in finite strings, 83–84
  in infinite strings, 81–83
  in membranes, 122–125
  in plates, 128
Forward propagating plane waves, 31–33
Forward scanning, 482
Fourier series, 34
Fourier's theorem, 79
Free-field microphones, 178
Free fields, 244
  barriers in, 314–315
Free-fixed bar, 95
Free-free bar, 93–94, 105–108
Freely vibrating circular membranes, 115–120
French horn, 550, 551
Frequency, 13–15
  angular, 13
  beat, 60
Frequency modulation (FM), 208
Frequency ranges of musical instruments, 565
Frequency response, 173
Frequency-weighting, 52–55
Fresnel, Augustin Jean, 6
Full anechoic chambers, 65, 202–203, 245
Functional hearing loss, 221
Fundamental mode, 76

Gabor, Dennis, 496
Galileo Galilei, 3
Gas-jet noise, 385–398
  control of, 393–398
Gaseous flows in pipes or ducts, 396–398
Gassendi, Pierre, 3

Gear enclosures, 378
Gear noise, 371–378
Gear-tooth error, 373
Gear train noise, 374–376
Gears, helical, 377
Gehry, Frank, 273
Gene therapy, 236
Glissando, 525
Glockenspiel, 553
Gong, 556
Gradient operator, 21–22
Grazing incidence, 178
Grimaldi, Franciscus Mario, 4
Grösser Musikvereinsaal, 263
Group speed, 143
Guitar, 526

Hair cells, 217
Half-power point, 191
Hanning window, 197
Harmonic excitation, 593–596
Harmonica, 541, 543
Harmonics, 76
Harmonium, 542
Harmony, discord and, 520–521
Harp, 524–525
Harpsichord, 527
Harris, Cyril M., 8, 270
Hausksbee, Francis, 4
HD-DVD, 574–575
Headphones, 580
Hearing aids, 231–233
  analog, 232
  digital, 233
  implantable, 233
  programmable, 233
Hearing
  characteristics of, 222–227
  human, 213–217
  in animals, 237–239
  mechanism of, 217–220
  physiology of, 213–239
Hearing loss, 220–221
Helical gears, 377
Helmholtz equation, 113
Helmholtz, F. L., 6
Helmholtz resonator, 145–148
Hemianechoic chamber, 65
Hemispherical wave, 65
Hemostasis, acoustic, 504
Hertz (unit), 15
High-fidelity reproduction, 10, 570
High-pass filters, 161–162
Highway construction noise, 350–352
Highway traffic noise, 335–344
Hollywood Bowl, 271–272
Holography, ultrasonic, 496–497
Hoods, 306–309
Hooke, Robert, 4, 5
Hooke's law, 91
Human hearing, 213–217
Human voice, 551–552
Humidity, reverberation time and, 256–258
Hunt, Fredrick V., vii
Hutchins, Carleen, 9, 531
Huygens, Christian, 5
Huygens' principle, 38–39
Hydrophone arrays, 425–426
Hydrophones, 424

Imaging processes, ultrasonic, 492–497
Impedance, 82
  acoustic, see Acoustic impedance
Impedance tube, 137
Impingement noise, 395
Impulse, excitation by, 597–598
In-line silencers, 397
Indoor noise criteria, 325–328
Industrial applications of ultrasound, 449–497
Industrial noise sources, 357–405
Inertia blocks, 604
Infinite cylindrical pipes, 131–132
Infinite strings, forced vibrations in, 81–83
Infrasound, 15
Ingard, Karl Uno, 9
Inner ear, 215
Insertion loss, 167
Instantaneous energy density, 66
Intensity, sound, 60–62
Interference patterns, standing wave, 40
Interferometers, 483–484
Internal damping, 608–609
Inverse square law, 64
iPod, 582–583
Isolators, 603–604
Isono system, 580–581

Jacobs device, 493–494
Joule, James P., 7
Journal bearing noise, 378–379

Kármán vortex street, 386
Kennedy Center for the Performing Arts, 2568–270
Kettledrum, 554
  membrane theory and, 120–122

Key notation, 518
Kinetic energy density, 65–66
Kircher, Athenasius, 4
Kleinhaus Music Hall, 264
Kneser liquids, 451
Knudsen, Vern O., 8
Kryter, Karl D., 9

La Scala Opera House, 264–266
Ladder-type acoustic filters, 164–165
Lagging pipes, 398
Laminar flow, 367
Langevin, Paul, 7
Latching overload indicator, 187
Leakage, 196
Leger lines, 510, 511
Leibniz, Gottfried Wilhelm, 5
Lighthill, Sir James, 9
Lighthill's parameter, 388
Linear-array transducer specification, 472
Lin-hun, 2
Lindsay, R/ Bruce, 9
Lip-reed instruments, 550–552
Liquid crystal imaging, 496
Listening spaces, sound fields of,
  subjective preferences in, 276–278
Lithotripsy, extracorporeal shock wave, 503–504
Live rooms, sound intensity growth in, 247–249
Longitudinal wave equation
  derivation of, 89–92
  solutions of, 92–93
Longitudinal waves, 31, 89
Loops, 77
Lord Rayleigh (see Strutt, John William)
Loudness, 18, 52, 225, 323
Loudness notation, 518, 519
Loudspeakers, 578–580
Low-pass filters, 159–160
Lumped acoustic element, 145
Lumped acoustic impedance, 152–153
Lute, 523, 524
Lyre, 523, 524

Machinery noise control, 357–405
Machining, ultrasonic, 488
Magnetic recorders, 208
Magnetic recording, 571–573
Magnetic resonance imaging (MRI), 505
Magnetostriction, 466
Magnetostrictive materials, 425
Magnetostrictive strain coefficient, 468
Magnetostrictive stress constant, 468
Magnetostrictive transducers, 466–468
Magnification factor, 595
Major keys, 518, 519
Mandolin, 526
Marimba, 553
Masking, 226–227
Mass, conservation of, 20–22
Mass concentrated vibrating bars, 95–97
Mass control case, 283–284
Mass flux, 20–21
Mass law, field incidence, 284–245
Mean-square value, autocorrelation and, 613
Measure, in music, 513
Measurement error, noise, 199–200
Mechanical drive element noise, 382–385
Mechanical impedance, 82
  characteristic, 133
Mechanical reed instruments, 541–542
Mechanical stress measurements, 485–486
Median noise level, 332
Medical uses of ultrasound, 497–506
Membrane theory, kettledrum and, 120–122
Membranes, 111–125
  forced vibrations of, 122–125
  freely vibrating circular, 115–120
  rectangular, 113–115
  wave equation for, 111–113
Mersenne, Marin, 3
Meshing frequencies of gears, 372–373
Metals, ultrasonic working of, 488–490
Metronome, 516
Microphone sensitivity, 170–171
Microphones, 171–181
  selection and position of, 177–181
Millikan, Robert, 8–9
Millington-Sette theory, 256
Minimum audible field (MAF), 222–225
Minimum audible pressure (MAP), 223–225
Minnesota Orchestra Hall, 267–268
Minor keys, 518, 519
Mixed hearing loss, 221
Mixed layer, 423
Momentum, conservation of, 22–25
Monopoles, 62–64
Morse, Philip M., 9
Motion excitation, 600–604
Motion sensing, ultrasonic, 484–488
Mouth organ, 541, 543
MP3, 581
MSC-NASTRAN finite element program, 610
Muffler system descriptors, 167–168
Mufflers and silencers, 398–404
  dissipative, 398–399, 403–404
  reactive, 399–404

Multichannel sound systems, 576–577
Multijet diffuser, 394
Music, 509
  musical instruments and, 509–565
  pitch for, 2
Musical acoustics, 509
Musical Instrument Digital Interface (MIDI), 561–562
Musical instruments, 1–2, 521–565
  electrical and electronic instruments, 521, 522, 556–561
  frequency ranges of, 565
  music and, 509–565
  percussion instruments, 521, 522, 553–556
  strings, 521, 522, 522–536
  wind instruments, 521, 522, 536–552
Musical notation, 510–513
Musical notes, duration of, 513–516
Musical staff, 510, 511

Narrow band analyzers, 192
NASTRAN finite element program, 610
National Environmental Policy Act of 1969, 319
Near field effect, 62
Nematic crystals, 496
Neutral axis, 100
Newton, Isaac, 5
Nobel Prize, 6, 9, 10, 213, 496
Nodes, 77
  true, 106
Noise, 15
  environmental, performance indices for, 55–57
  perception of, 323–325
  vibration and, 585
  white, 228, 615
Noise and number index (NNI), 334–335
Noise bandwidth factor (NBF), 197
Noise barriers, 310–316
Noise control
  active, 404–405
  criteria and regulations for, 319–352
Noise Control Act of 1972, 320–321
Noise criteria, indoor, 325–329
Noise criterion curves, 325–326
Noise dosimeter, 187–189
Noise exposure forecast (NEF), 334
Noise insulation ratings, 296–300
Noise level, effective perceived (EPNL), 324–325
Noise-limited range, 433
Noise measurement, band pass filters and, 189–193
Noise measurement error, 199–200
Noise rating curves, 323
Noise reduction in duct systems, 167
Noise reduction of walls, 300–304
Noise source sound power, estimation of, 358–359
Noise sources in workplace, 357–358
Noisiness index, 323
Nonlinear acoustics, 617–626
Nonlinearity in solids, 625–626
Normal force, 23
Normalized impedance, 133
Note values, 515
Noy N (unit), 323
Nyquist frequency, 195

Oboe, 546
Ocarina, 539
Occupational Safety and Health Act of 1970, 321–323
Occupational Safety and Health Administration (OSHA), 321–322
Octave, 2, 510
Octave band analyzer, 191
Octave bands, 45–47
  one-third, 46–47
Odontocetes, 237–238
Office of Noise Abatement and Control (ONAC), 320
Ohm, Simon, 5
Olson, Harry F., 9
Omnidirectional microphone, 178
One-third octave bands, 46–47
Open-circuit response, 426
Open-ended pipes, 134–136
Opera houses, design of, 262–271
Optical interference methods, 483
OPTIMA computer program, 344
Orchestra, 562–563, 564
Orchestra Hall, Chicago, 264
Organ, 522, 547–550
Organ console, 547
Organ of Corti, 216–218
Organ pipes, 548–549
Otoacoustic emission, 219
Outdoor auditoriums, design of, 271–278
Oval window, 218

Panels, sound transmission through, 295–286
Particle displacement, 58
Particle velocity, 16, 58, 151
Partitions, double-panel, 289–292

Pascal (unit), 48
Percent isolation, 602
Percentile sound levels, 332
Percussion instruments, 521, 522, 553–556
Performance indices for environmental noise, 55–57
Period, 13
Personal sound exposure meter, 187
Petrillo Music Shell, 275
Phase angle, 116
Phase speed, 103
Phased transducer arrays, 469–472
Phon curves, 225
Phonons, 451–458
Physical modeling, electronic music, 562
Physiology of hearing, 213–239
Piano, 533, 535–536
Piccolo, 540–541
Picket fence effect, 196
Piezoelectric crystals, 459–460
Piezoelectric effect, 7
Piezoelectric materials, 425
Piezoelectric relationships, fundamental, 462–464
Piezoelectric strain constant, 462
Piezoelectric stress constant, 463
Piezoelectric transducers, 464–466
Piezomagnetic constant, 468
Pipes, 131
  closed-ended, resonances in, 132–134
  flared, 135
  gaseous flows in, 396–398
  infinite cylindrical, 131–132
  lagging, 398
  open-ended, 134–135
  standing waves in, 136–137
  unflanged, 135
  waves in, 155–158
Piston, equivalent simple, 120
Pitch, 2, 225–226, 510
Pitch range, 510
Plastics, ultrasonic working of, 488–490
Plates, 125–128
  forced vibrations in, 128
  vibrating thin, 125–128
Playback audio equipment, 576–582
Plumbing system noise, 366–370
Poisson effect, 126
Poisson ratio, 125
Portable playback equipment, 581–583
Power, radiation of, from open-ended pipes, 135–136
Power spectral density, 614–615
Power transmission coefficient, 135–136, 159, 160–161, 162, 163
Preamplifier, 577
Pressure microphone, 178
Pritzker Pavilion, 273–276
Probability density, 612–613
Progressive waves in fluids, 619–626
Propagating mode, 141
Psychoacoustics, 227
Pulse technique in detectinf flaws, 481
Pump noise, 366–367

Q factor, 465
Quadruple meter, 517
Quantum particles, coupled, 456–458
Quartz crystals, 459–460
Quintillianus, 2

Radian frequency, 13
Radial nodal lines, 117
Ragas, 527
Railroad noise regulations, 349–350
Random-incidence microphone, 178
Random vibrations, 612–615
Rayl (unit), 58
Ray theory, 620–621
Rays, 40
Reactive mufflers and silencers, 398–403
Real strings, 85
Real-time analysis, 193
Recorder, 539
Recording equipment, 569–571, 571–576
Recreational noise, 319
Rectangular cavity, 138–140
Rectangular membranes, 113–115
Reed instruments, 536–552
Reed organ pipe, 542
Reflection, 40–42
Reflection coefficient, 249
Refraction, 42–44
  basic laws of, 43–44
  underwater, 421–423
Relaxation, 445
Relaxation frequency, 449
Relaxation processes, 445–451
Relaxation time
  adiabatic, 449
  translational, 445
Resolution, 198–199
Resonance method of measuring sound propagation speed, 487
Resonances in closed-ended pipes, 132–134
Response time, 174

Rest symbols, 513, 515
Restrictive flow silencer nozzle, 394
Reverberant effects, 246–247
Reverberant fields, 244–245
  sound absorption in, 256–258
  sound levels due to, 259–261
Reverberation, 409
Reverberation chambers, 203–206, 254
Reverberation-limited range, 432
Reverberation time, 205, 243, 253
  predicting, 253
  sound absorption and humidity and, 256–258
Reynolds number, 367
Rhythm, 517
Ribbon-type speaker system, 580
Rigid-walled circular waveguide, 144
Ring frequency, 397
Roller bearing noise, 379–382
Room constant, 259
Room criterion curves, 327
Rooms
  dead, decay of sound in, 254–256
  live, sound intensity growth in, 247–249
Root-mean-square sound pressure, 47–49
Rotational waves, 31
Rudnick, Isadore, 9

Sabine equation, 243
Sabine, Wallace Clement, 6, 243
Sarrusophone, 545, 547
Sauveur, Joseph, 4
Saxophone, 544
Scanning plane, 470
Schlieren imaging, 495–496
Seasonal thermocline, 414
Seawater
  absorption in, parametric variation of, 419–421
  speed of sound in, 411–413
  velocity profiles in, 413–415
Secondary emission ratio, 492
Semianechoic chamber, 201–202
Semitones, 512
Sensitivity, 176–179
  ear, 222–225
Sensorineural deafness, 221
Sequencer, 561
Serial analysis in filters, 191
Sextuple meter, 517
Shadow zones, 44–45
Sharp notes, 512
Shear stress, 23
Shear viscosity, 419
Shear waves, 31
Shell isolation rating (SIR), 296, 298–300, 301
Shock waves, 622–625
Side-lobe ratio (SLR), 197
Signal-to-noise ratio (S/N ratio), 181
Silencers, see Mufflers and silencers
Simple harmonic motion (SHM), 75
Simple harmonic solutions of wave equation, 75–76
Sine function, 13, 14
Sine waves, generating, 13, 14
Single-reed instruments, 541
Sitar, 527
Skudrzyk, Eugen, 9
Small enclosures, 309–310
SNAP 1.1 computer program, 344
Snell, Willbrod (Snellius), 5
Snell's law, 44, 422
Social surveys of noise, 344–345
Sonar, 409
Sonar equations, 427–431
  shortcomings of, 437
  summarization of, 437
  transient form of, 434–437
Sonar parameters, 429–431
Sonar system, 428
Sonar transducers, 424–427
Sonochemistry, 454
Sonoluminescence, 453–454
Sonophoresis, 505
Sonoporation, 506
Soprano clef, 512
SORAP acronym, 433–434
Sound, 13, 16
  decay of, 252–256
    in dead rooms, 254–256
  growth of, with absorbent effects, 251–252
  speed of, see Speed of sound
  underwater, concepts in, 410–411
  unwanted, 15
  wave nature of, 13–15
Sound absorption
  in reverberant field, 259
  reverberation time and, 256–258
Sound absorption coefficients, 249–251
Sound channels, 44
Sound decay, 8
Sound fields, 244–246
  of listening spaces, subjective preferences in, 276–278
Sound focusing, 262
Sound generation, 16–17
Sound intensity, 60–62

Sound intensity growth in live rooms, 247–249
Sound intensity level, 64
Sound intensity probes, vector, 181–182
Sound-level meter (SLM), 152–186
  integrating, 186–187
  using, 183, 184–185
Sound levels, 52–55
  due to reverberant fields, 259–291
  equivalent, 56, 329, 335
Sound-measuring instrumentation, 173–209
Sound power, 200
  noise source, estimation of, 358–359
Sound power level, 64
  addition method for measuring, 207–208
  alternation method for measuring, 207
  specific, 362–365
  substitution method for measuring, 206–207
Sound pressure, 28
  root-mean-square, 47–48
Sound pressure level (SPL), 18, 64
  at distance from walls, 304–305
Sound propagation, 16–18
  nature of, 31
  in water, 409–410
Sound propagation speed, resonance method of measuring, 487
Sound transmission, through panels, 285
Sound transmission class (STC), 296–298
Sound transmission coefficient, 282
  combined, 294–296
Sound, unwanted, 15
Sound velocity, 17–18
Sound waves, see Waves
Soundboards, 522–523
Sousaphone, 551
Specific acoustic impedance, 151, 410
Specific acoustic resistance, 410
Specific sound power level, 362, 363–364
Spectral density, 614–615
Speech intelligibility, 227–230
Speech interference level (SIL), 230–231
Speed of sound, 16–18
  in seawater, 411–413
Sphere, theoretical target strength of, 438–439
Spherical spreading, 416–418
  combined with absorption, 421
Spherical waves, 64
STAMINA 2.0 computer program, 344
Standing wave interference patterns, 40
Standing-wave ratio, 137
Standing waves, 36–38, 76–78
  in pipes, 136–138
Static deflection, 598
Steel drum, 554–555
Stiffness, 85
Strain, 90
Stress, 91
Strings, musical instruments, 521, 522, 522–536
Strings,
  finite, 83–84
  infinite, forced vibrations in, 81–83
  real, 85
  vibrating, see Vibrating strings
Struck-string instrument, 533, 534–536
Strutt, John William (Lord Rayleigh), 5, 6
Subjective preferences in sound fields of listening spaces, 276–278
Substitution method for measuring sound power level, 206–207
Surface waves, 483
Surgery, acoustical, 504–505
Synthesizers, 558–561
System loss factor, 606

Tambourine, 556
Tanglewood Music Shed, 272–273
Target strengths, 409
  theoretical, of sphere, 438–439
Tenor banjo, 527
Tenor clef, 512
Tensile forces, 91
Therapeutic uses of ultrasound. 503–506
Thermocline, 414
Thermodynamic states of fluids, 18–19
Threshold sound level, 188n
Time signature notation, 516–517
Timpani, 554
Tooth error, 373
Torsional waves, 31
Traffic conditions, adjustments for, 340–343
Traffic noise, evaluation of, 335–344
Transducer array-element configurations, 470–472
Transducer arrays, 469–472
  phased, 469–470
Transducer response, 426–427
Transducers, 458–468
  electrostrictuve effect, 461–462
  magnetostrictive, 466–470
  piezoelectric, 462–466
  sonar, 424–427
Transfer function, 189
Transformer noise, 366
Transistor, 8
Translational relaxation time, 445
Transmissibility, 598–603

Transmission coefficient
  combined sound, 294–296
  sound, 282
Transmission loss, 282
  measuring, 292–294
  underwater, 416–418
Transmission loss in duct systems, 167
Transmitting-current response, 427
Transverse sensitivity, 612
Transverse vibrations of vibrating bars, 100–104
Transverse wave equation, derivation of, 71–73
Transverse waves, 31, 71
Triangle, 553
Triple meter, 517
Trombone, 551
True nodes, 106
Trumpet, 550
Tuba, 551
Tubular bells, 553
Tuning fork, 105, 107–108, 553
Turbulent flow, 367

Ukulele, 525–526
Ultrasonic cleaning, 479–480
Ultrasonic delay lines, 484
Ultrasonic diagnosis, safety of, 502
Ultrasonic echoencephalography, 500
Ultrasonic fire sensing, 487–488
Ultrasonic flowmeter, 486–487
Ultrasonic holography, 496–497
Ultrasonic imaging processes, 492–497
Ultrasonic machining, 488
Ultrasonic methods for flaw detection, 481–483
Ultrasonic motion sensing, 487–488
Ultrasonic viscometer, 491
Ultrasonic welding, 488–489
Ultrasonic working of metals and plastics, 488–490
Ultrasonics, 443–476
Ultrasound, 15
  Agricultural, 490
  diagnostic uses of, 498–501
  industrial applications of, 479–497
  medical uses of, 497–506
  safety, 502–503
  therapeutic uses of, 503–506
Uncorrelated sound waves, 58
Undamped natural frequency, 587
Underwater acoustics, 409–439
Underwater refraction, 421–423
Underwater sound, concepts in, 410–411
Underwater transmission loss, 416–418
Unflanged pipes, 135
Universal gas constant, 19
Universal joint noise, 385
Urick, Robert Joseph, 10

Vacuum tubes, 8
Vector sound intensity probes, 181–182
Vehicle noise, 339–344
Vehicle noise regulations, 347–349
Velocity-depth function, 413
Velocity profiles, 413–415
  in sea, 413–415
Vibrating bars, 89–108
  boundary conditions for, 93–95
  general boundary conditions for, 98–100
  mass concentrated, 95–98
  transverse vibrations of, 100–104
Vibrating strings, 71–85
  assumptions, 71
  energy of, 80–81
Vibrating thin plates, 125–128
Vibration(s), 585–615
  forced, see Forced vibrations
  modifying source of, 603
  noise and, 585
  random, 612–615
Vibration absorbers, 604–606
Vibration control, 598–603
  techniques for, 603–610
Vibration measurements, 611–612
Vibration systems, modeling, 585–590
Viola, 528–533
Violin, 528–533
Violin octet, 533, 534
Violoncello, 528–533
Viscometer, ultrasonic, 491
Vitruvius (Marcus Vitruvious Pollo), 3
Voice, human, 551–552
Voice recognition, 575–576
Voicing, 544
Voltage-controlled amplifier (VCA), 558–559
Voltage-controlled filter (VCF), 558, 559
Voltage-controlled oscillator (VCO), 558, 559
Voltage sensitivity, 612
Volume velocity, 151
Volume viscosity, 419

Walls
  enclosures and barriers and, 281–316
  noise reduction of, 300–304
  sound pressure level at distance from, 304–305
Water, sound propagation in, 409–410
Water hammer, 368–370

Water-hammer arresters, 370
Wave distortion, 617–619
Wave equation
  effect of initial conditions on, 78–80
  general solution of, 73–74
  longitudinal, see Longitudinal wave equation
    for membranes, 111–112
  simple harmonic solutions of, 75–76
  transverse, see Transverse wave equation
Wave motion, 16
Wave nature of sound, 13–15
Waveguide
  boundary condition at driving end of, 144
  with constant cross-section, 140–143
  rigid-walled circular, 144
Wavelength, 18
Waves, 31–39
  complex, 33–36
  energy flux density of, 434
  forward propagating plane, 31–33
  hemispherical, 65
  one-dimensional, 620
  in pipes, 155–158
  plane, 620
  reflection of, at boundaries, 74–75
  spherical, 64
  standing, see Standing waves
Webster, Arthur Gordon, 5
Weighting curves, 52–55
Weighting functions, 196–199
Welding, ultrasonic, 489
Wever-Bray effect, 219
White noise, 228, 615
Wind instruments, 521, 522, 536–553
Window duration, 196
Window error, 196
Workplace, noise sources in, 357–358
Workplace noise exposure, 346–347
Worst noise hour, 336

Xylophone, 553

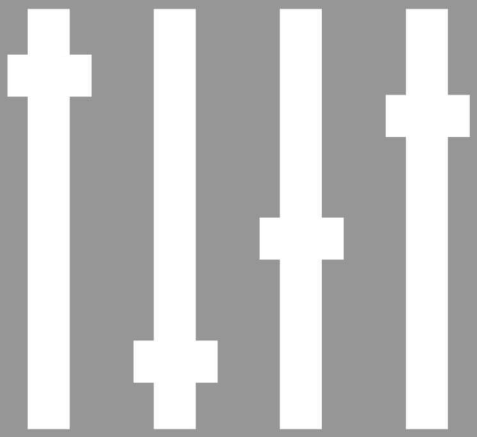
Young's modulus, 18, 91, 629
Young, Thomas, 6

Zither, 529

ABLETON BIBLE

THE ZERO TO HERO GUIDE TO

MIXING IN ABLETON LIVE



MIX BOOK 1

FUNDAMENTALS & MIX PREPARATION

COPYRIGHT NOTICE

This eBook has been electronically stamped with your purchase details to prevent copyright infringement.

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the email address below.

info@abletonbible.com

www.abletonbible.com

FOREWORD

Firstly, thank you for purchasing the first book in our Ableton Live mix series. This set of eBooks aim to cover every aspect of the mix process in Ableton Live.

We have designed this book to be used as a practical reference guide during the mix-down of a track.

We recommend reading this book cover to cover first, to gain a basic understanding of all of the subjects covered. We have included summary pages at the end of each section, which bullet point any key principles covered. These summary pages can be used as a quick and easy reference to jog our memory whilst we are mixing a track.

Produced by Ableton Bible
Edited by Dan Exert

For free Ableton presets, racks and samples, Visit our website

www.abletonbible.com

Table of Contents

MIXING EXPLAINED 8

MIXING EXPLAINED SUMMARY & KEY POINTS 9

BASICS OF SOUND AND WAVES 10

BASIC WAVE THEORY 10
SINE WAVES 12
TRIANGLE WAVES 13
SQUARE WAVES 14
SAWTOOTH WAVES 15
WHITE NOISE 17
PROPAGATION OF SOUND 18
WHAT IS SOUND 18
THE SPEED OF SOUND 19
WAVES 19
AMPLITUDE 19
CYCLES AND WAVELENGTHS 20
FREQUENCY 21
PHASE 21
DECIBELS 26
LOGARITHMIC SCALES 29
HOW WE HEAR 30
HEARING IS NOT LINEAR 31
BASICS OF SOUNDS AND WAVES SUMMARY & KEY POINTS 32

CRITICAL LISTENING 34

FREQUENCY 34
DYNAMICS 34
TIMBRE 35
TRAINING OUR EARS 36
IMAGING 36
SPATIAL AWARENESS 38
CRITICAL LISTENING SUMMARY 39
CRITICAL LISTENING KEY POINTS 39

THE ROOM 40

BASIC ROOM ACOUSTICS 40
STANDING WAVES 40
ROOM MODES 41
THE ROOM SUMMARY & KEY POINTS 44

MONITORING 45

COMPONENTS 45
THE COMPUTER/DAW 45
THE DAC 45
AMPLIFIER 45
MIXING CONSOLE 45
THE ROOM 45
MONITOR SPEAKERS 45
A NOTE ON EAR FATIGUE 47
MONITOR TYPES & USES OVERVIEW 48
1 – THE MID RANGE RADIO SPEAKER 48
2- NEARFIELD MONITORS 50
3 - SUB WOOFER 50
4 – HIGH END STUDIO SPEAKERS 50
SPEAKER SWITCHING & SET LEVELS 51
THE LISTENERS PERSPECTIVE 52
REFERENCE LOCATIONS 54
SPEAKER ORIENTATION 54
SPEAKER POSITIONING 55
MONITORING SUMMARY & KEY POINTS 56

STUDIO MONITORS - WHAT NOT TO DO 58

SPEAKER SELECTION 58
KEEP THINGS SYMMETRICAL 59
AVOID WALLS 61
AVOID POOR ANGLES 62
AVOID PROBLEMATIC REFLECTIONS 64
EQUAL LOUDNESS AND LEVELS 65
TOO MUCH LOW END 66
STUDIO MONITORS SUMMARY & KEY POINTS 67

ACOUSTIC TREATMENT 68

STANDING WAVES: 68
ECHO AND REVERB: 68
SIDE REFLECTIONS: 68

ACOUSTIC TREATMENT GUIDE 69

EARLY REFLECTIONS FROM THE SIDES 69
EARLY REFLECTIONS FROM THE CEILING 71
LOW-END RUMBLE & STANDING WAVES 72
FURTHER ENHANCEMENTS 73
ACOUSTIC TREATMENT SUMMARY & KEY POINTS 76

PSYCHOACOUSTICS EXPLAINED 77

WHAT ARE PSYCHOACOUSTICS 77
PERCEPTION IS NOT LINEAR 77
DYNAMIC RANGE 79
THE HAAS EFFECT (THE PRECEDENCE EFFECT) 80
MASKING 83
CRITICAL BANDS 83
NATURAL EAR PROTECTION TO LOUD NOISES 84
EQUAL LOUDNESS CURVES 84
FLETCHER-MUNSON TO ALTER DEPTH 85
EARS HEAR SOUND IN AN AVERAGE WAY (RMS) 85
LOCALISATION 87
LAYERING SOUNDS 89
DEPTH PERCEPTION 89
PSYCHOACOUSTICS SUMMARY & KEY POINTS 90

MIXING TECHNOLOGY AND EQUIPMENT 91

THE MIXING CONSOLE 91
THE PATCH BAY 91
OUTBOARD EQUIPMENT 92
THE DAW (DIGITAL AUDIO WORKSTATION) 92
STUDIO MONITORS 92
DIGITAL VERSUS ANALOG 92
MIXING TECHNOLOGY AND EQUIPMENT SUMMARY & KEY POINTS 93

MIX PREPARATION 94

BITDEPTH & SAMPLE RATE 95
TRACK ARRANGEMENT MARKER MAPPING 95
SIDECHAIN TRIGGER TRACKS 95
RENAMING & COLOURING 95
REMOVE LOW-END RUMBLE 96
TRACK COUNT REDUCTION 96
GROUPING & SUBMIXING TRACKS 97
DOUBLE CHECK TUNING AND TIMING OF PARTS 97
BOUNCE TO AUDIO 97
DAW MIXING TEMPLATES 98
TRACK STRUCTURE CHANNEL 98
SETTING UP A STEREO/MASTER BUS COMPRESSOR 98
IMPORTING & FILE MANAGEMENT 99
MACRO MANAGEMENT AND HANDLES 99
REFERENCES AND A/B COMPARISON 101
COMPUTERS & THEIR SETTINGS FOR MIXING 101
BUFFER SIZES & LATENCY 102
FREEZING AND FLATTENING 102
MIX PREPARATION SUMMARY & KEY POINTS 103

MIX PREP: TRACK ANALYSIS 105

OBJECTIVITY & SEPARATION OF MIX PHASES 105
MIXING SOMEONE ELSES TRACK 106
MIXING OUR OWN TRACK 107
MIX CHECKLIST 108
TRACK ANALYSIS SUMMARY & KEY POINTS 109

GAIN STAGING 110

WHAT IS GAIN STAGING? 110

GUIDE TO GAIN STAGING IN OUR MIX 111

RECORDING 111
INDIVIDUAL TRACK GAINS 111
MIX BUSSES 112
GAIN STAGING SUMMARY & KEY POINTS 114

ABLETON'S MIXER ANATOMY ROUTING AND SIGNAL FLOW 115

WHAT IS SIGNAL FLOW 115
INPUTS 116
MONITOR MODES 116
INSERTS 117
SEND KNOBS 117
PAN POT 117
MUTE BUTTON 117
SOLO BUTTON 117
RECORD BUTTON 117
VOLUME FADER 118
METERING 118
METERING AND DYNAMIC RANGE TERMINOLOGY 118
PEAK METERING ANALYSIS 119
RMS METERING ANALYSIS 119
OUTPUTS 120

ROUTING, BUSSES & GROUPS 121

MIX/SUMMING BUSSES & AUXILIARIES 121
BUSSES 121
GROUPS 121
SEND & RETURNS 122
STEREO OUTPUT 123
ROUTING, BUSSES & GROUPS SUMMARY & KEY POINTS 124

FINAL SUMMARY 125

MIXING EXPLAINED

Mixing is the process of 'combining', 'blending' or 'balancing' multiple sound sources together. This gives us our resulting mix.

The word 'mix', is an abbreviation of the word mix-down, which means that we will be Mixing 'X' number of tracks down, into a single stereo track.

We can think of mixing as being lots of cumulative processes that achieve a final result. It only takes one tiny detail to be wrong in any one of these processes, and the integrity of the entire mix could be ruined. Mixing is quite a subjective artform, so there are plenty of different ways to get the same job done, as well as many different opinions on how things should be achieved.

First and foremost, we need to realize that everyone has different tastes. What might sound amazing to one person may sound like rubbish to someone else. As our years of production accumulate, we will begin to build ourselves a preferred workflow and style. Further to this, as time passes, our ears will become much more finely tuned to what we are hearing, allowing for more analytical listening. This means that when we are listening to individual sounds within a mix, or the entire mix-down itself, we can understand exactly what it is we want to achieve, and how we plan to achieve it.

For example you may be asked for:

- A punchier kick drum
- A warmer bass-line
- More prominent and upfront vocals
- More of a 3 dimensional mix

All of these ideas are more easily achievable once you have a grasp of the tools at your disposal and the fundamentals of mixing. Not only will these different tastes and styles change from producer to producer, but other factors will need to be considered, such as the genre of music, the track itself, the emotions you wish to convey in the track, and even things such as whether the record will be pressed onto vinyl or not.

Factors such as these can seem irrelevant and are often overlooked, but these small details matter. A common example of this is how tracks intended for vinyl need to be more dynamic than their digital counterparts. Vinyl tracks need careful attention with regards to the

extreme high & low frequencies to ensure proper groove geometry, as well as to prevent unwanted distortion that could make the needle skip out the grooves.

Within the mixing process there are a few key functions that will need to be controlled and manipulated to give us a well-polished mix.

These key functions are:

- Volume (Balance)
- Pan & Stereo Imaging
- Equalization
- Dynamics
- Space & Depth
- Effects
- Automation

Once we have got all of these different elements under control, the end result will be a 'bounce' of the sum of all our tracks. Typically this will give us our stereo-interleaved audio file, which will be ready for mastering.

Many producers will mix their tracks as they are composing. This technique is fine, because it can help with a producer's creative flow; however, we strongly suggest revisiting the entire track from a mixing perspective. Different mixing methods are covered extensively in a later eBook from this mixing in Ableton Live series.

MIXING EXPLAINED SUMMARY & KEY POINTS

- Mixing is the process of combining multiple sound sources together

- Different genres and music mediums require different mix-downs

- Mixing is very subjective; each producer has his or her own way of working

- A 'bounce' is when we print a project or piece of audio to a file, such as a .wav or .aiff

- The mix process can be split down into key functions such as volume, panning, stereo imaging, EQ, dynamics, space & depth, effects and automation

BASICS OF SOUND AND WAVES

In this section, we will take a brief look at the basics of sound waves, frequencies and harmonics. It's important that we understand the basics of sound, so that we can fully grasp the reasons why we do certain things in the latter stages of the mix process.

BASIC WAVE THEORY

Firstly we have basic wave theory, the most common waveforms are:

- Sine Wave
- Triangle Wave
- Square Wave
- Pulse Wave
- Sawtooth Wave
- White Noise

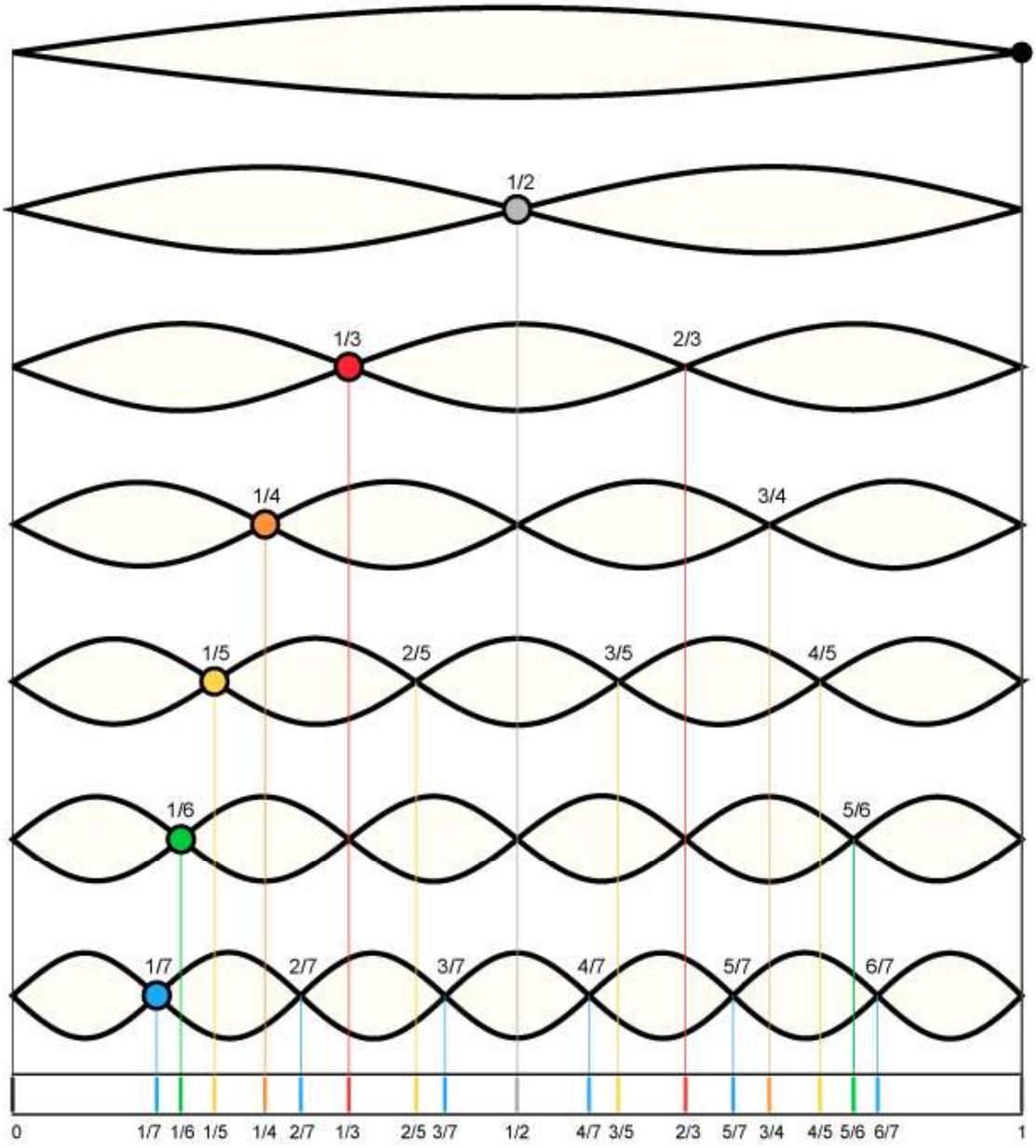
It is unusual in the mixing stage, that waveforms will be this simple. Usually the audio we work with will be complex waveforms, which can be thought of as being made up of lots of simple waveforms stacked together, as well as a lot of effects and processing.

As we are aware, a sound is simply a vibration or oscillation. If a vibration cycles periodically, then they will have a musical tone.

Complex tones can be broken down into a fundamental frequency and overtones, which together make the composite sound. (Fourier analysis) This is the technique used in spectral analysers, which is why they are also commonly referred to as FFT's. (Fast Fourier Transform.)

The reason we get these harmonics (overtones) in complex sounds is because of how an object can cause micro vibrations within itself. For example a guitar string can be split in half, third, quarter and so on, which will subsequently produce the 2nd, 3rd and 4th harmonics.

The amplitude and phase of these harmonics will also play a considerable role in the overall timbre of the composite sound.



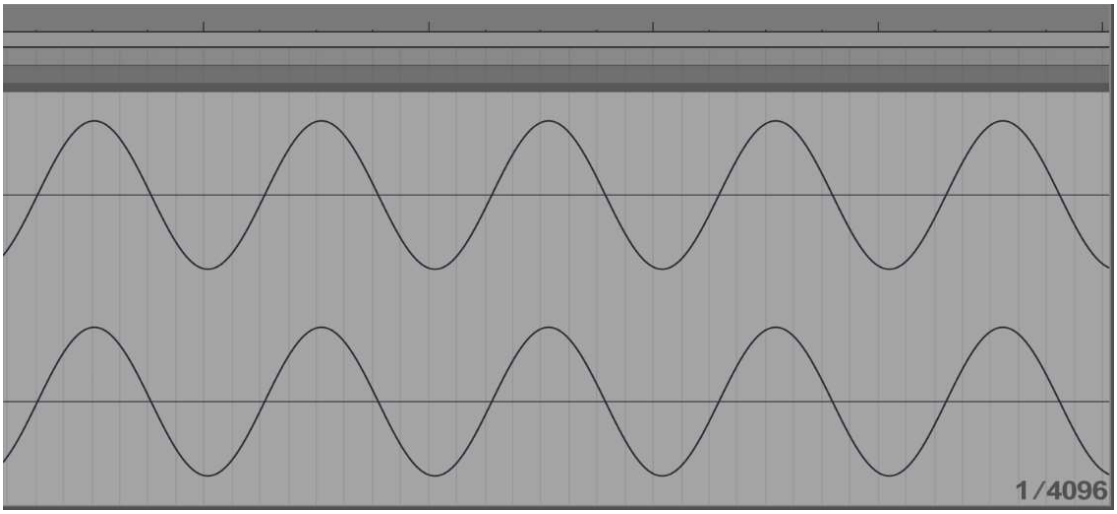
Splitting a vibrating object into whole number multiples shows a breakdown of the harmonics. This theory will also become useful later to understand the concept of standing waves.

When we listen to sounds, we will be hearing them in the human hearing range. This is between 20Hz–20KHz. Generally, as we get older, most humans can't hear much audio past around 15-18kHz.

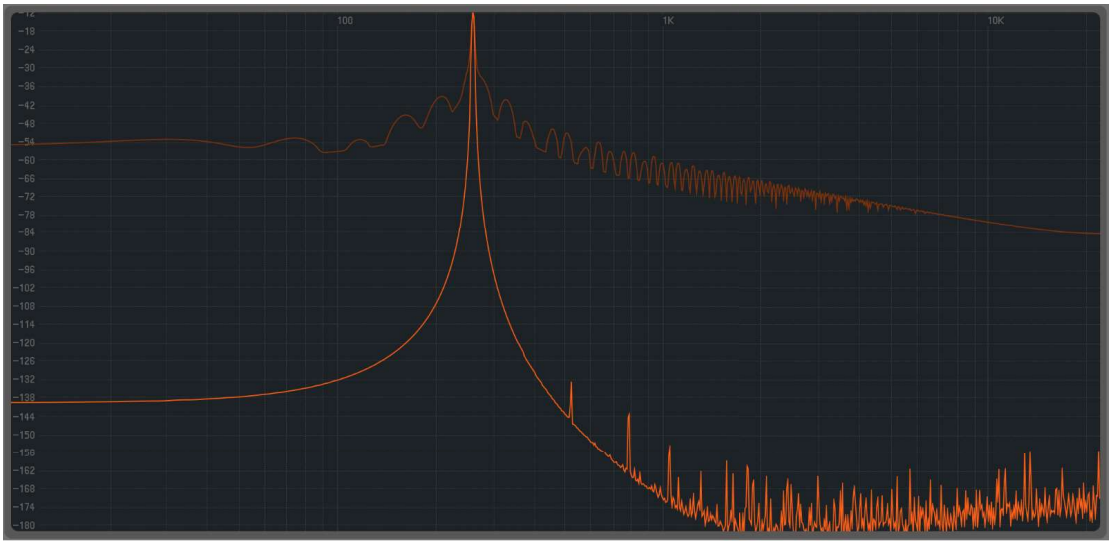
We will take a look at each of these waveforms and their harmonic content using a frequency analyser. Spectral analysers can be set to work logarithmically or in a linear manor. We recommend keeping your spectral analyser in logarithmic mode, which is it's default setting. This is because there are is a huge range of frequencies that we need to be able to see. Using a logarithmic graph allows us to see all of the frequencies in a more condensed and useful format.

Sine Waves

Are the purest waveforms and only consist of the fundamental, with no harmonics. This means filters have little effect on their overall sound. They are often used for adding sub-bass to sounds.



A sine waveform



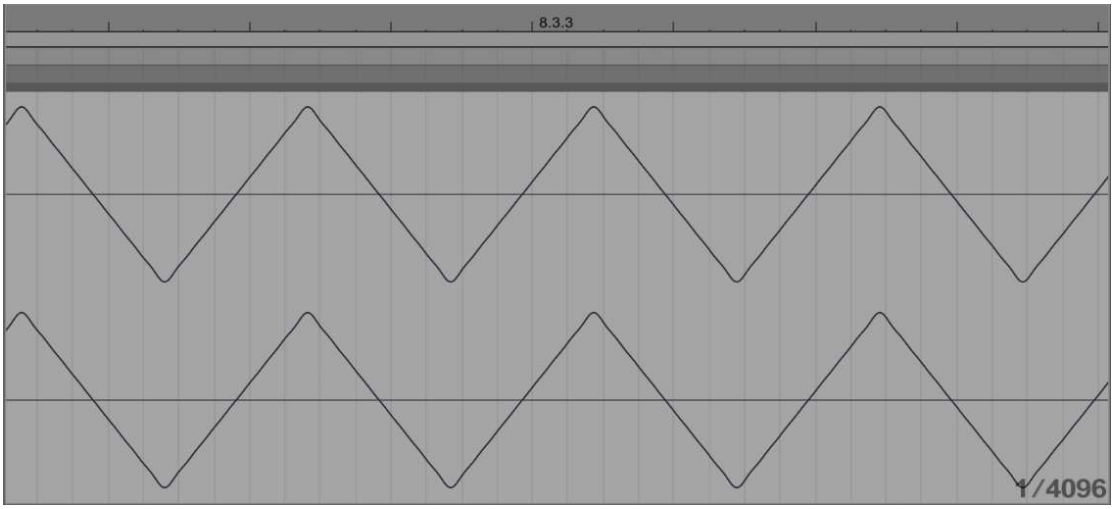
A sine waves spectral analysis

Triangle Waves

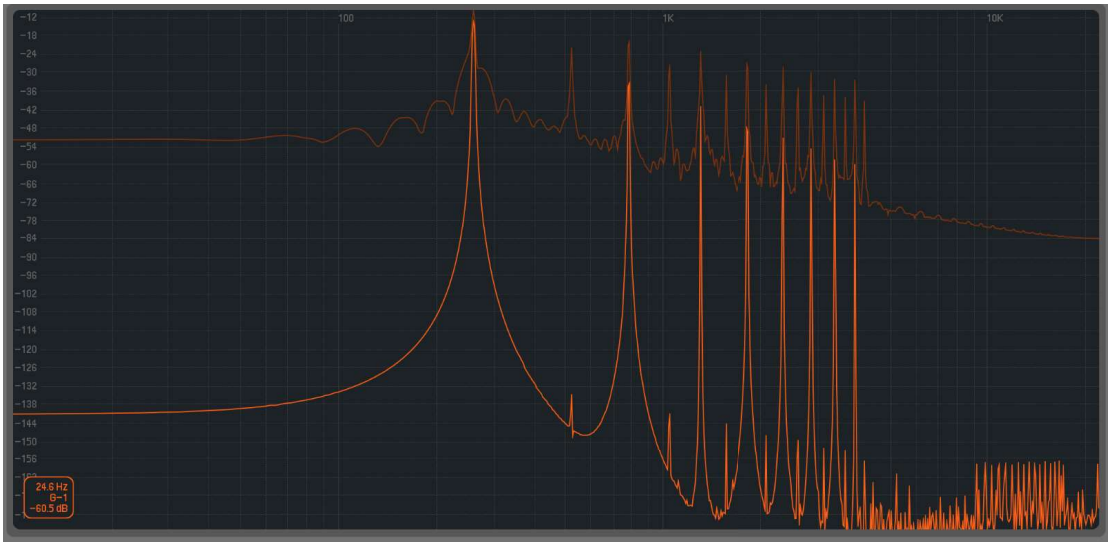
Contains only odd harmonics. The harmonics amplitude can be calculated using the formula:

Amplitude = $1/N^2$
(N = number of the harmonic, 1 being the fundamental)

Triangle waves have a sound that is somewhere in between a sine wave and a square wave.



A triangle wave



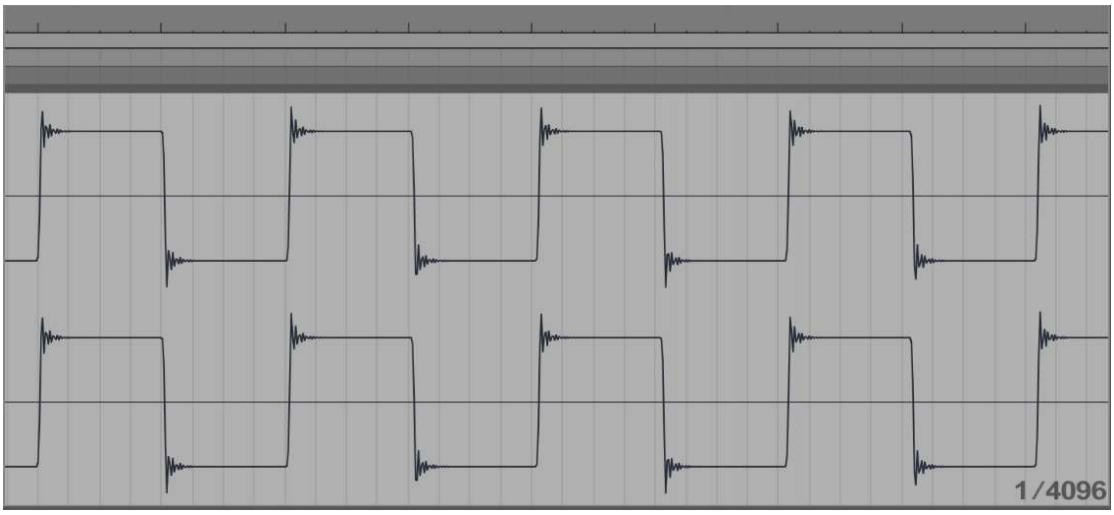
Spectral analysis of a triangle wave

Square Waves

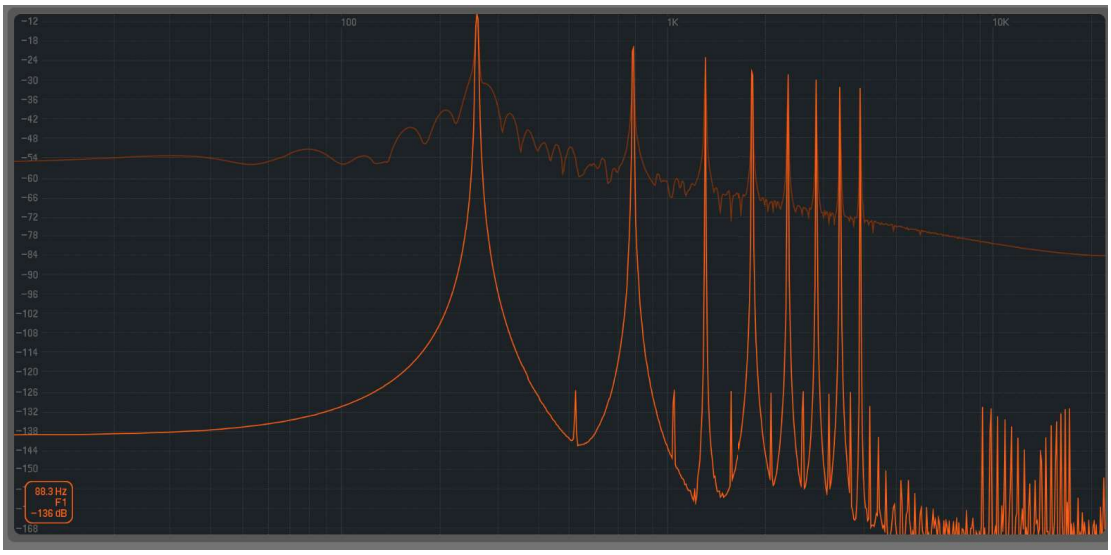
Contain only the odd harmonics; this gives them a hollow-type sound, which is good for recreating any form of wood instrument. Square waves can sometimes be referred to as pulse waves. Altering the pulse width can change a pulse wave's tonality.

The amplitude of a square waves harmonics can be calculated using the formula:

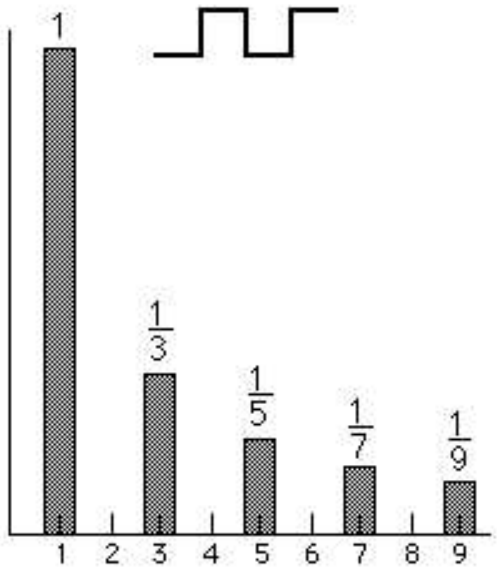
AMPLITUDE = 1/N



A square wave



Spectral analysis of a square wave (Sq8)



A diagram to show a square wave. It's characteristic 'hollow' or 'wooden' sound is achieved from its lack of even harmonics. This diagram also shows the change in amplitude per harmonic, in relation to the fundamental frequencies amplitude.

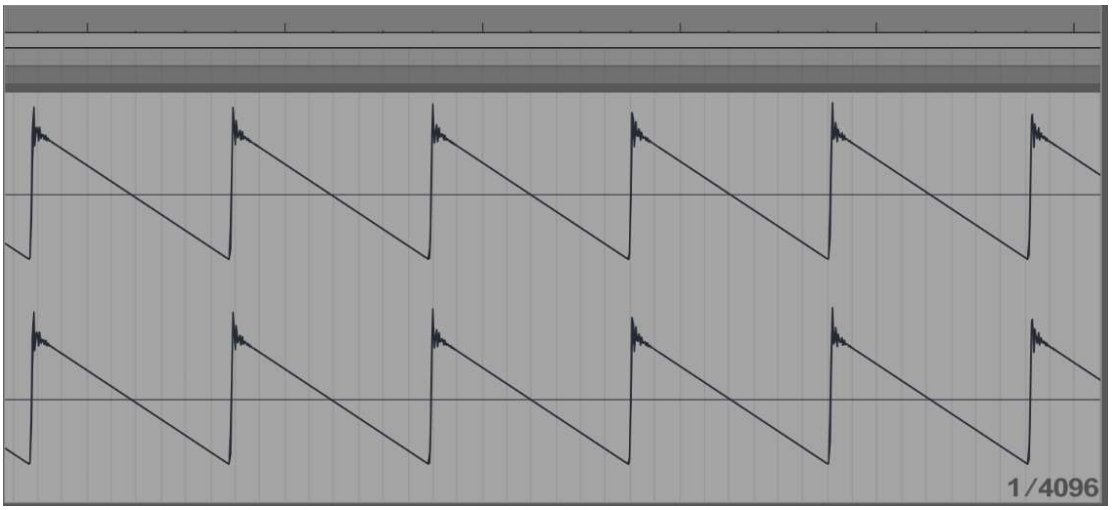
Sawtooth Waves

Contain all the frequencies in the harmonic series. Each harmonic gets quieter as it goes up the scale, this causes Saw waves to produce a very rich sound, which makes them perfect for shaping with the use of filters.

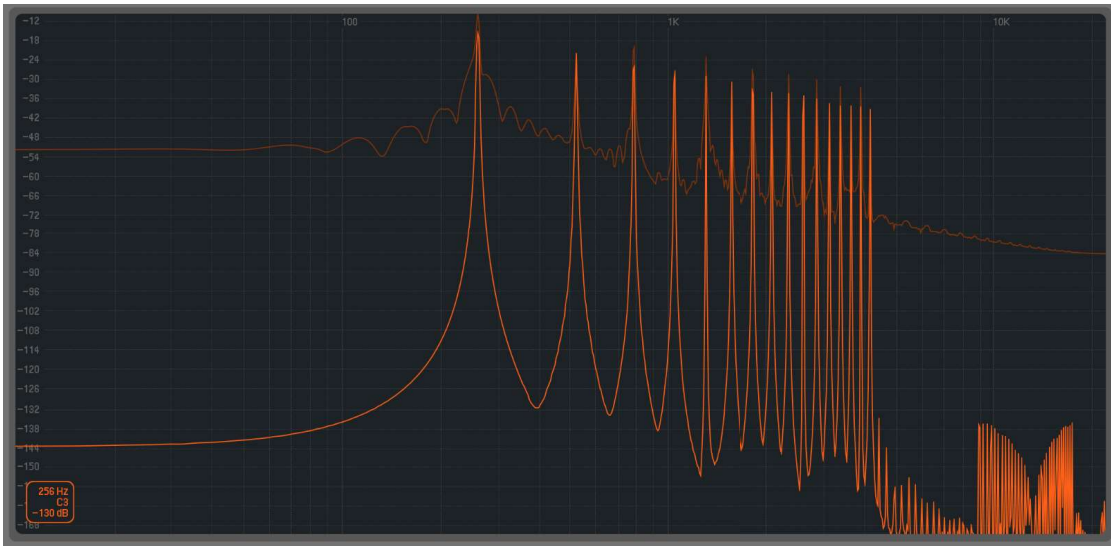
The amplitude of a sawtooth waves harmonics can be calculated using the formula:

AMPLITUDE = 1/N

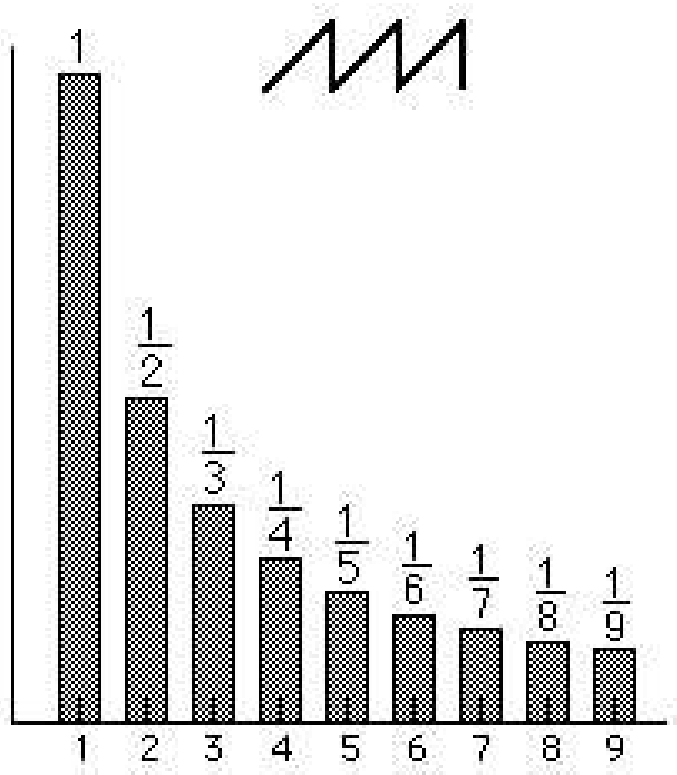
For example, Harmonic 5 will be a 5th of the amplitude in comparison to the fundamental.



A sawtooth wave



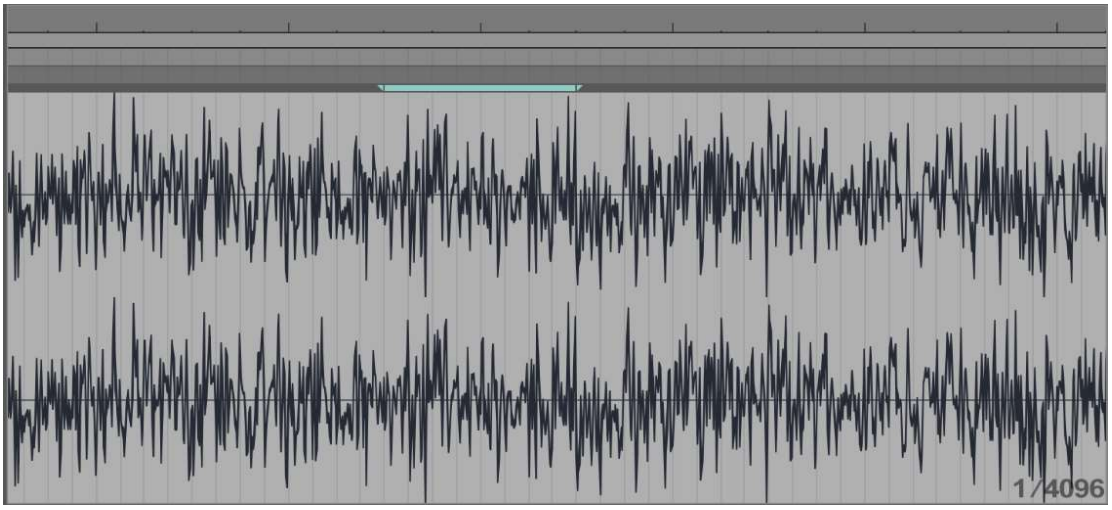
Spectral analysis of a sawtooth wave



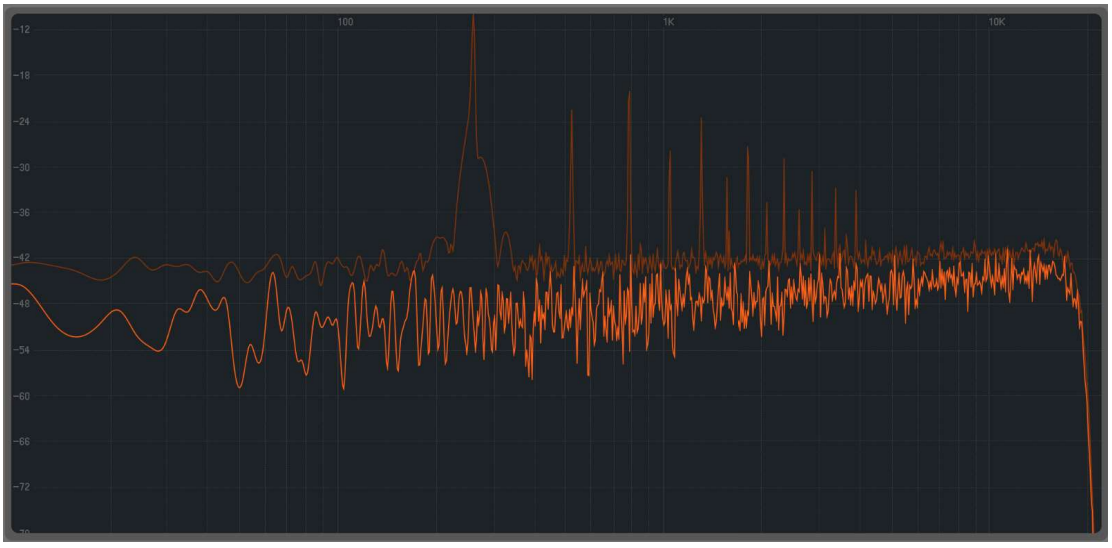
A diagram to show a sawtooth waves harmonics and their relative amplitudes in comparison with the fundamental frequency.

White Noise

Contains all the frequencies within the frequency spectrum at full amplitude. White noise has many different uses, from synthesizing drums, to creating textures and swells.



White Noise



Spectral analysis of white noise

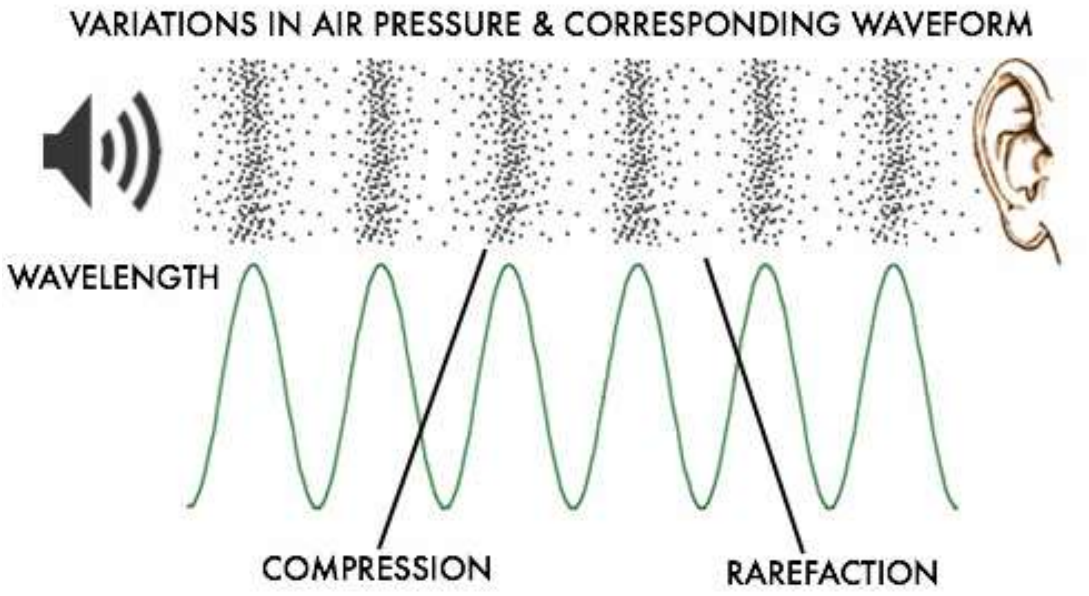
PROPAGATION OF SOUND

To be able to understand how sound is affected by different environments, we must first take a look at what it is, and how it propagates through a medium such as air.

WHAT IS SOUND

Sound is simply waves of compressed and refracted air particles. These can be thought of as waves, or vibrations that travel through the air. Sound can be created by a voltage, which moves a magnet within a speaker. This speaker, which is transducing electrical energy into kinetic energy, is making the speaker driver move back and forth. This in turn excites the air particles in front of the speaker cone.

These air particles propagate through a medium (in this case air) to our ears, where our eardrum's (another form of transducer) detect the vibrations and turn them into neural-electrical signals for our brains to process.



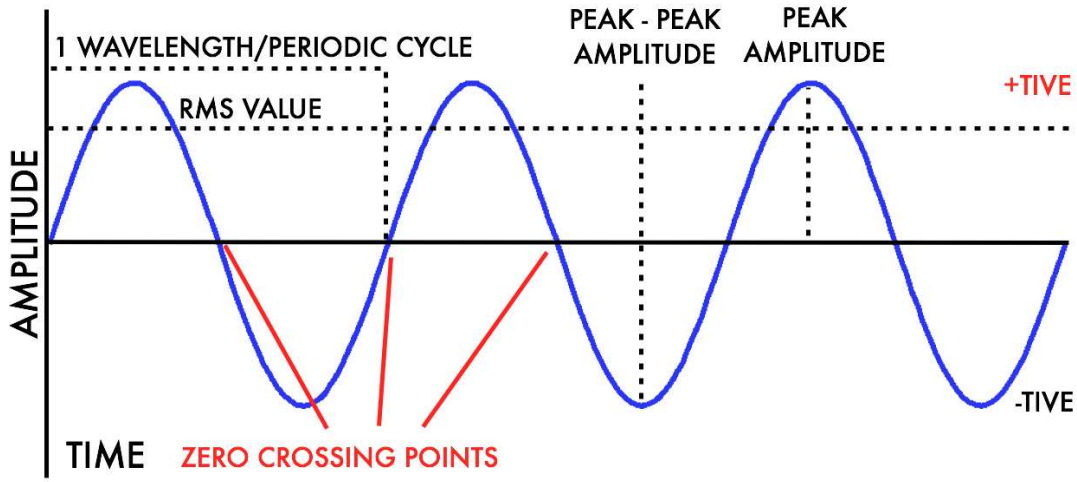
THE SPEED OF SOUND

The speed at which sound travels through a medium is known as a velocity. Sound travels at different velocities dependent on the material it is passing through. Mediums such as wood, water and air will all have different velocities.

For the purposes of mixing we only need to worry about air. The speed of sound through air at room temperature is 343 meters per second. (1125ft per second) This can be approximated to 1 millisecond per foot, or 3ms per meter.

WAVES

We can describe a wave in terms of amplitude, wavelength, velocity, frequency, harmonic content (timbre) phase, and envelope.



AMPLITUDE

The amplitude is the distance from the central line to the furthest distance from the central line. This is commonly measured in Decibels, (dB) or Volts. (V) As we get further away from the central line, the amplitude, or loudness will increase.

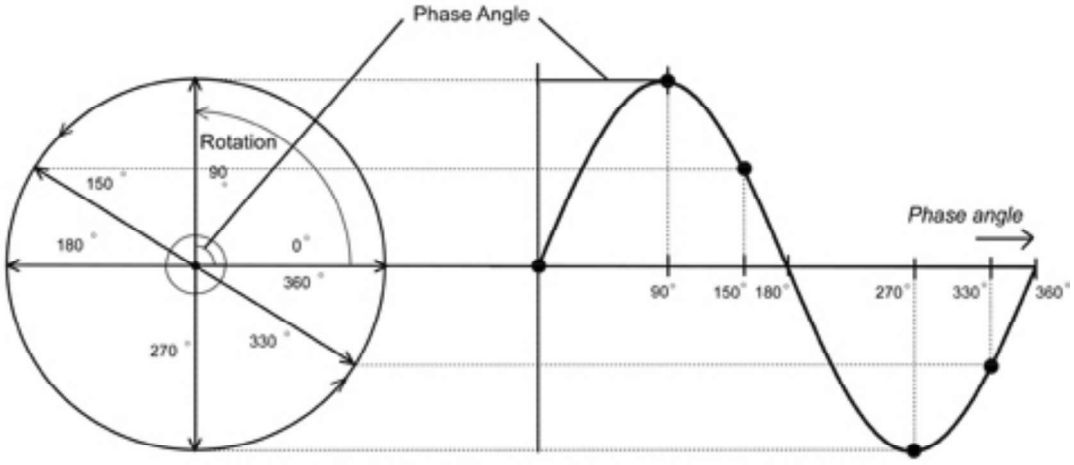
Measuring the highest peak to the central line is known as the peak amplitude, whereas measuring the entire waveform from its +tive, to its –tive value is know as peak to peak.

We can also measure 'RMS' which stands for 'root mean squared'. This will approximate the average amplitude of the wave.
This is important because it gives us an indicator of the overall loudness of our track, whereas the peak volume only really gives us an indicator of whether our fast transient peaks are clipping or not.

CYCLES AND WAVELENGTHS

If we look at the diagram below, we can see one complete cycle of a sine wave. A cycle needs to pass through 360 degrees to become a full cycle. This can actually be shown as a circle; however, we split the circle in half and flip it over. This allows us to represent the wave over time.

We can also see that the waves 'degree' defines its phase. A wave can start, or be 'offset' at any point along the 360 degrees.



The wavelength of a waveform is the 'length' of the 'wave' or the distance from the beginning of the cycle, to the end of the cycle.

We can use this formula to calculate wavelengths:

Λ (Lambda) (Wavelength) = Velocity of sound in air (343) / Frequency (Hz)

This formula proves that low frequencies have large waveforms, between 5-20 meters, whereas higher frequencies can be as small as a matter of centimetres in length.

Generally the low frequencies are most troublesome when mixing. This is very important when dealing with the acoustics of a room and standing waves.

FREQUENCY

We can see that a wavelength will periodically cycle. The amount of time it takes a wavelength to complete a full cycle, in seconds, is known as the frequency (measured in hertz). Frequency is directly related to pitch. If we have two wavelengths and one is twice as fast as the other, we can say that it is double the frequency. For every doubling in frequency, we will get a doubling in pitch.

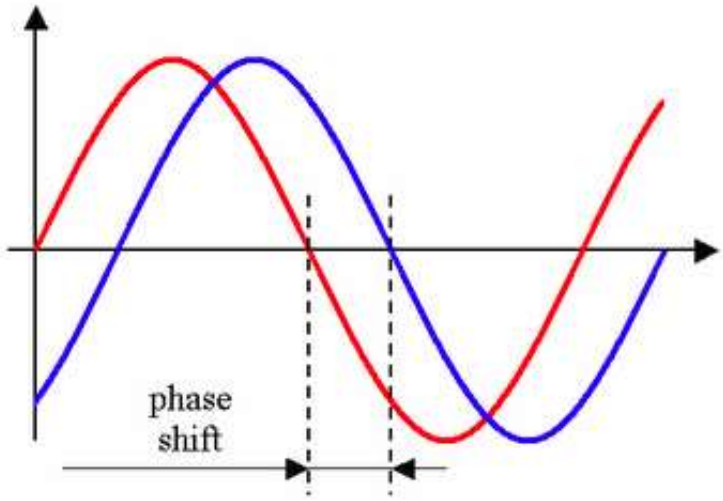
For example, A4 on the keyboard is 440Hz. Therefore A5 will be 880Hz. The A4 and A5 are an octave apart on the keyboard, which is 12 semitones.

PHASE

Phase can be used to describe the relationship between different waveforms, as well as to pinpoint a waveform's amplitude as it cycles through time. This is a crucial concept that is important in many aspects of music production, from microphone placement and sound design, right through to room acoustics and general mixing.

If we analyse a sine wave, we can think of it as positive and negative air pressure. (Compression and expansion of air molecules, as the speaker moves back and forth) We can see how the waveforms air pressure, increases and decreases equally through time (milliseconds).

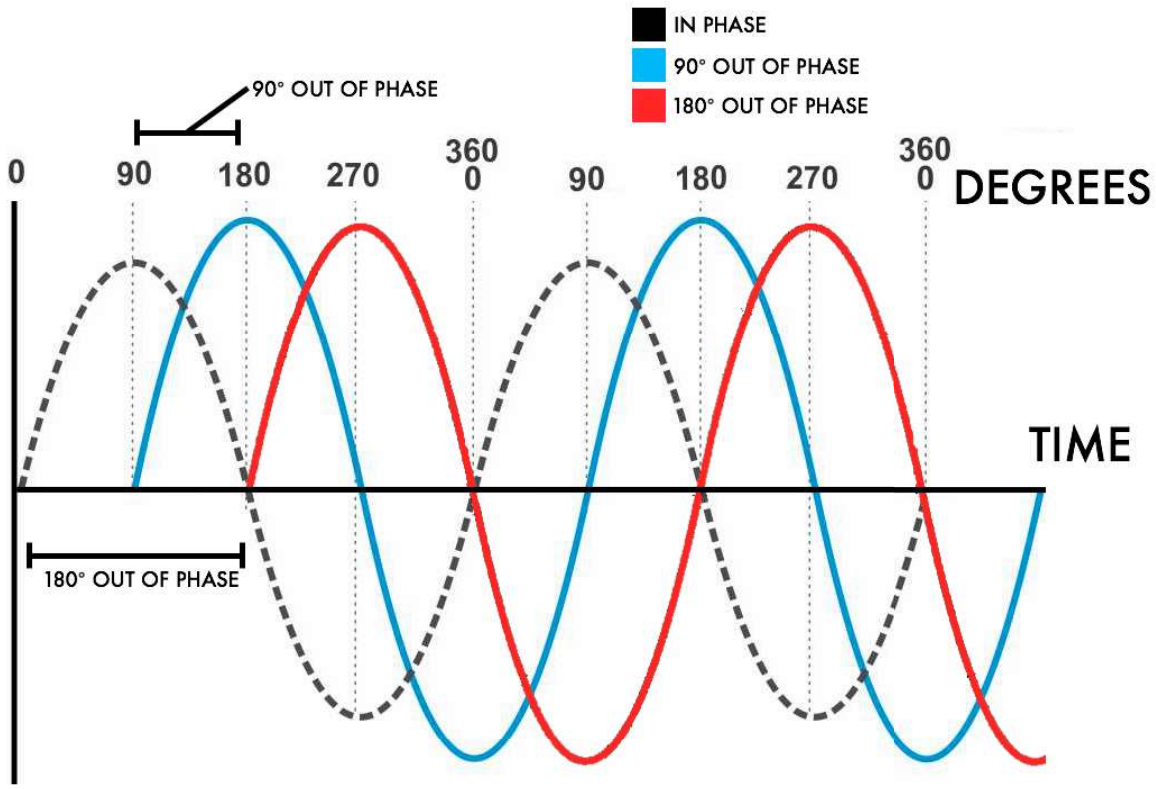
By drawing a zero line through the waveform, we can see that the waveform reaches zero crossing points throughout a single cycle.



If we look at this waveform without the concept of time, we can actually change this waveform into a circle, the only reason we don't look at waveforms like this, is because we usually need to look at them relative to time.

We can now see that this cycle's zero crossing points can be measured in degrees of a circle. This is also how we measure phase.

If we start two waveforms a few milliseconds apart from each other, they will be out of phase, due to the cycle starting at different points in time.

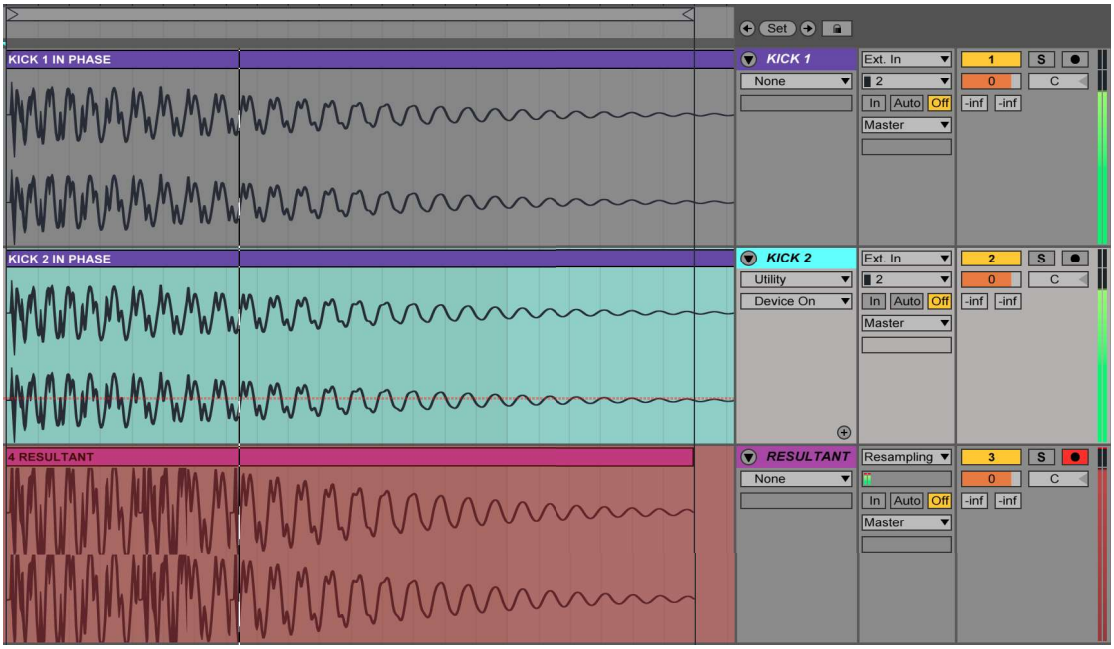


When we mix sounds, we will often be adding multiple waveforms together. If we stack some sine waves on top of each other, we can see how their phase relationships will cause them to add together (sum), or cancel each other out.

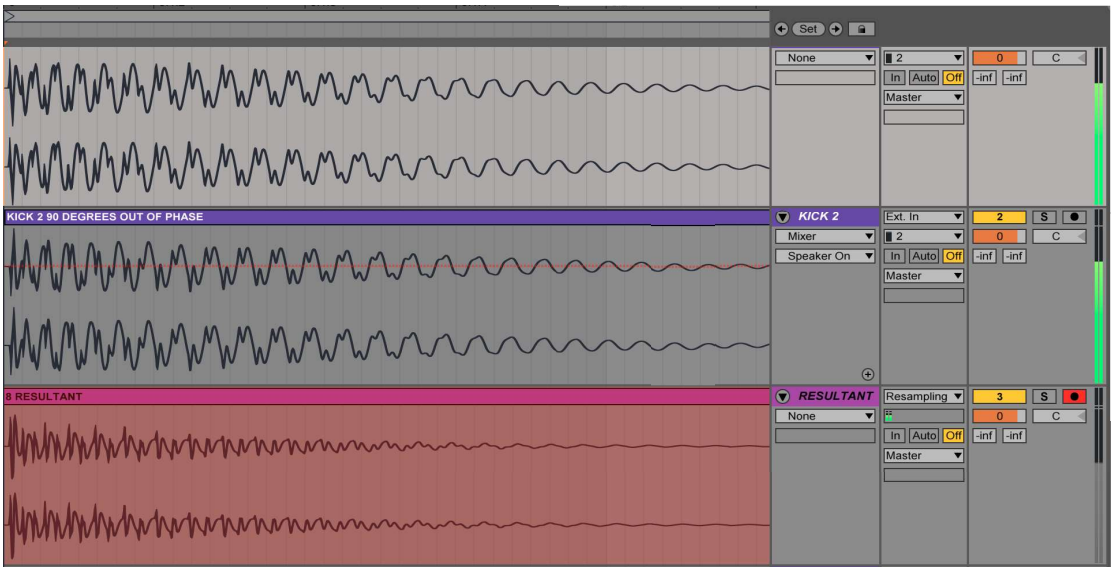
Two identical signals directly out of phase will cause complete phase cancellation.



Two identical waveforms in phase with each other will sum together and increase in volume.

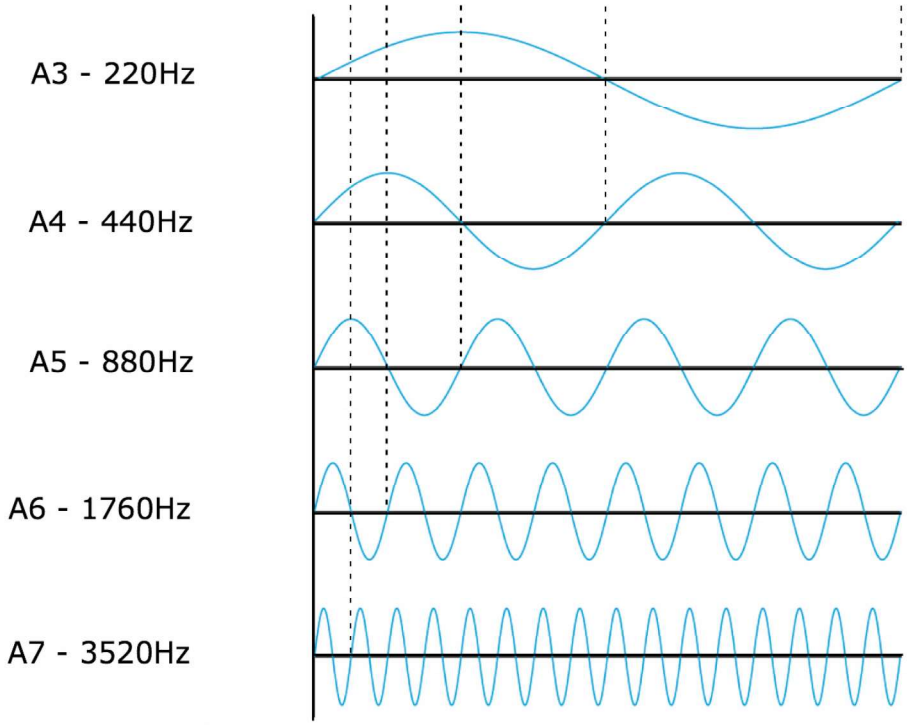


Two waveforms out of phase may have a mixture of summing and cancellation at different frequencies.



This is fundamentally how we use up headroom when we sum all our tracks together to one stereo track, which will cumulatively get louder as we add more instruments. Hence, we can usually get more power, focus and volume with fewer instruments.

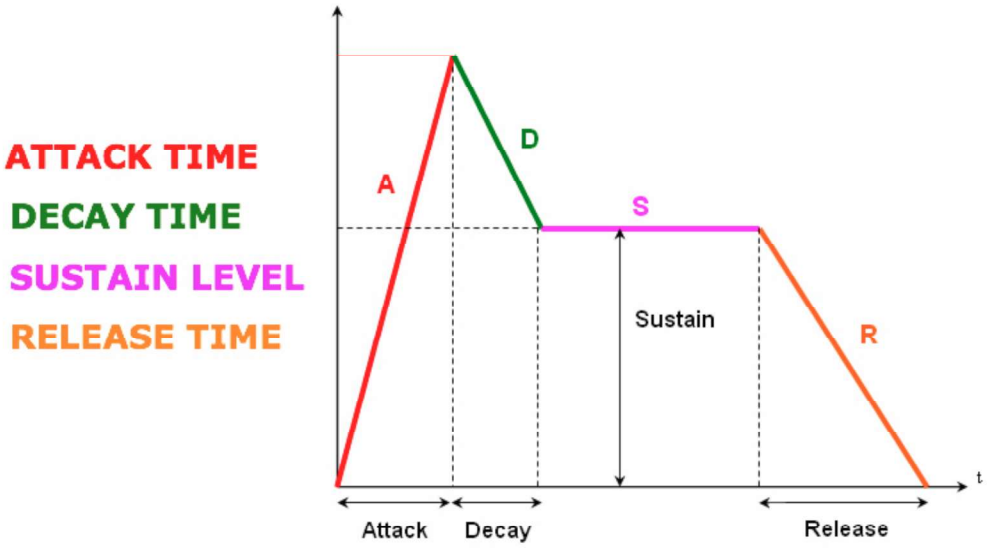
Finally, if we have two sine waves one octave apart, and we sum them together, we can see how these sine waves slot together, which means they are in harmony with each other.
The resultant wave will be a combination of the two waves.



ENVELOPES

We also perceive sound by their amplitude envelopes. This is a more logical, three-dimensional way of perceiving the amplitude of a sound over a period of time, rather than any instantaneous moment.

We can break a sound down into different sections or phases:



There are more complicated versions of this 'envelope', but this is the type that we are going to be using most frequently.

The **attack** phase is the initial period of time, when the sound builds up to its peak or transient.

Decay is the time for a sound to go from its peak volume, down to its sustain level.

Sustain is the level at which a note will sustain if a key is pressed. For example, a VST synthesizer will commonly loop at a certain sustain level when a key is pressed on a midi controller.

Release the time it takes for the note to reduce to silence once we take our finger off the key.

These factors make up how audio sounds the way it does.
We could say that most drum samples have a very sharp attack, giving them a percussive sound. They also tend to have a medium/sharp decay and no sustain or release.

A string on the other hand, will have a very slow attack, and a more gradual release. A piano will have a fast attack and a fairly fast decay, then the sustain will continue if the sustain pedal is used, followed by a fairly sharp release.

DECIBELS

Now we will briefly go into the unit of measure for amplitude, the decibel. This is quite an unusual unit of measure, because the decibel is actually a relative unit of measure, which can be used in many different contexts.

Decibels are used to express a ratio between a reference level and the level we are measuring. These can be different for power, sound pressure, digital, and analog equipment.

The meaning of a decibel changes completely depending on what suffix is used after the DB. This is because the part that comes after the decibel will indicate what the decibel is being referenced to.

For example:

We have dBu which is used in analog equipment to measure voltage, 0dBu is equivalent to 0.775volts. This is not to be compared to a VU meter. A VU meter is a slow analog meter that measures the average amplitude of the signal, (RMS) this closely replicates the human ear's perception of sound. A VU meter can only be compared to another unit of measure when using a sine wave test tone. 0 VU corresponds to +4 dBu.



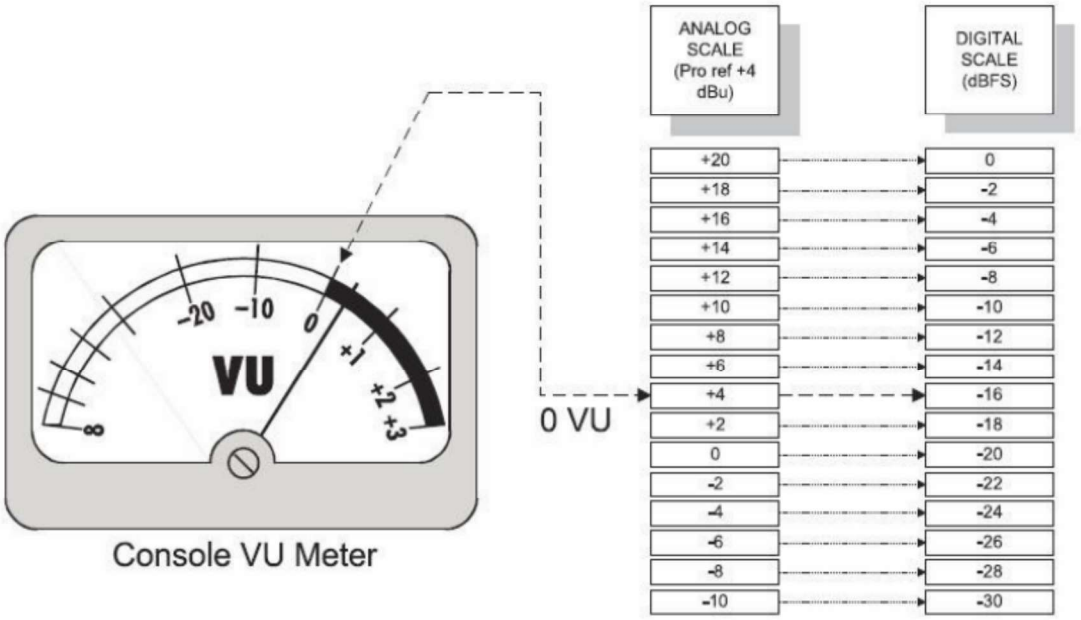
We also have dBv, which is very similar to dBu, but is referenced to 1 Volt rather than 0.775 volts.
When we mix within DAWs we can use a combination of different metering, knowing how each of them compare to each other is important.

We should mix VU meters so that the needle dances around the 0 VU point. This means we have the optimum signal to noise ratio (equivalent to +4dbu)

When we are mixing on the channel faders in our DAW, 0 VU (+4dBu) should be equivalent to between -12 and -16 dB Full Scale

The reason for this change between analog hardware and DAW's is because from -10dBfs to 0dBfs within a DAW is actually supposed to be used as headroom.

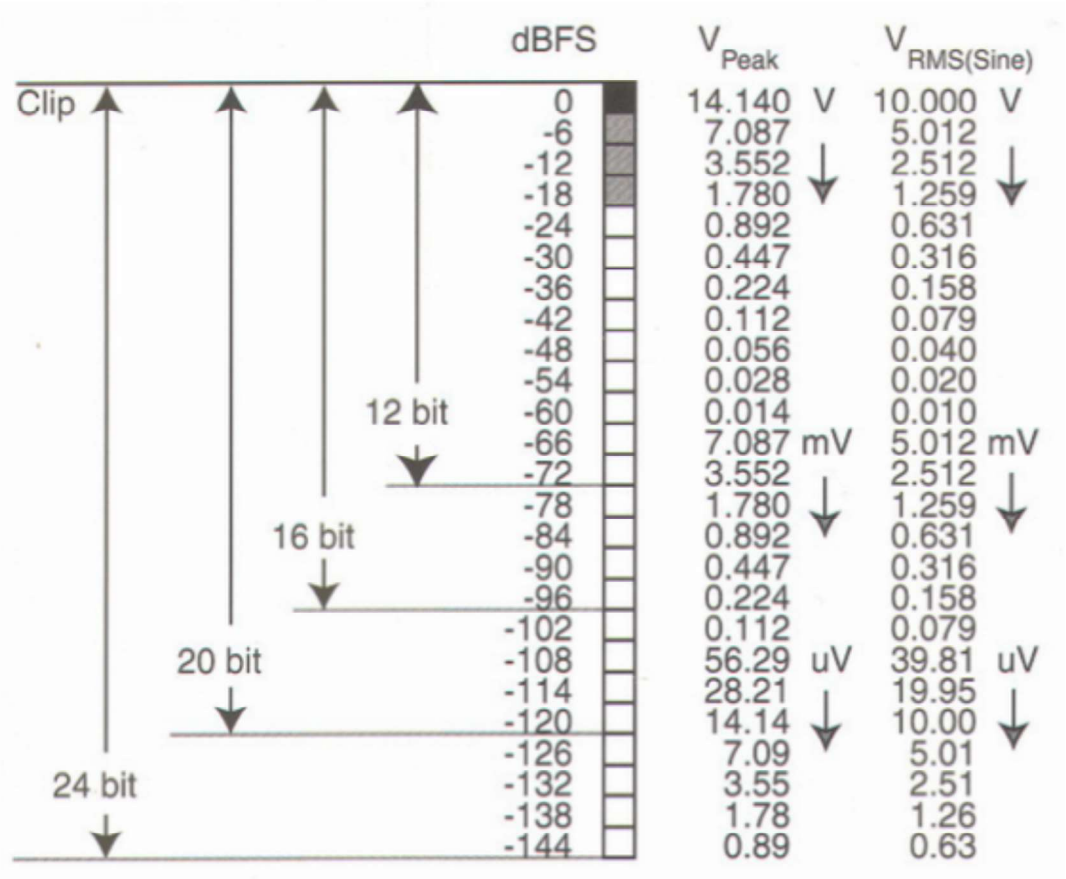
We have dB SPL, which is 'Decibels Sound Pressure Level.'
This is referenced to the quietest sound the human ear can hear, which is 20 micro-Pascal's. This is the unit of measure used for things such as, measuring the loudness of a set of headphones. 1 Pascal is equivalent to 94dB SPL. (The sound of a motorbike going past)



A comparison of VU, analog and digital metering.

Finally we also have dBFS or 'decibels full-scale' which is used in modern day DAW's and digital audio. the dBFS scale works backwards from 0 dBFS. Zero dBFS is equivalent to the maximum volume achievable by a piece of digital equipment without clipping or distorting. Therefore a reading should never exceed 0dBFS, else it will cause digital clipping.

Because this level is variable, we measure backwards from 0dBFS, down to the noise floor. This scale is logarithmic, so we will struggle hearing sounds after around -48dBFS. As we can see decibels and loudness can be quite complicated, but to keep things simple, loudness in every day life, is most commonly referred to in dB SPL whilst levels on analog equipment are commonly measured in dBu. However, throughout this eBook, we will be working within the Ableton Live DAW, so we will be referring to dBFS (Decibels Full Scale) unless otherwise stated. The dynamic range of a DAW can change dependant on the bit-depth. The more bits we have, the larger the dynamic range, because the bits can be used to represent a larger range of values.



A diagram to show how bit-depth affects the dynamic range

LOGARITHMIC SCALES

As mentioned in the last section, decibels use a logarithmic scale. The reason for this is because it is similar to how the human ear interprets sound. The dynamic range of human hearing goes from 0dB SPL to 120dB SPL. This range could not be plotted on a scale in a conventional manor.

This is where logarithms come in. a Logarithmic scale is basically a condensed scale that allows us to plot more easily onto graphs. It does this using a mathematical function, so that logarithmic numbers increase exponentially. We can also see this on the meters of our volume faders.

We can also see a logarithmic scale by opening up a spectrum analyser; the frequency lines are spaced differently to show certain frequency ranges in more detail than others.

Here is a chart of the logarithmic value in decibels. This should help you to understand that a doubling in dB doesn't necessarily correlate to a doubling in loudness.

dB	Relative Power Increase
0	1
1	1.26
3	2
10	10
20	100
30	1000
50	100'000
100	10'000'000'000

The part where this gets confusing is that as we can see, 3 dB is a doubling in power (amplitude). However, a doubling of voltage would be equivalent to 6dB's, (sound pressure level) whilst in terms of psychoacoustics, our perception of double the loudness would be more like 10dBs.

This can all get very confusing, but as long as we can grasp the concept that we measure our dynamic range in decibels, and that this is an exponential scale, then we will have more than enough knowledge to mix tracks down effectively.

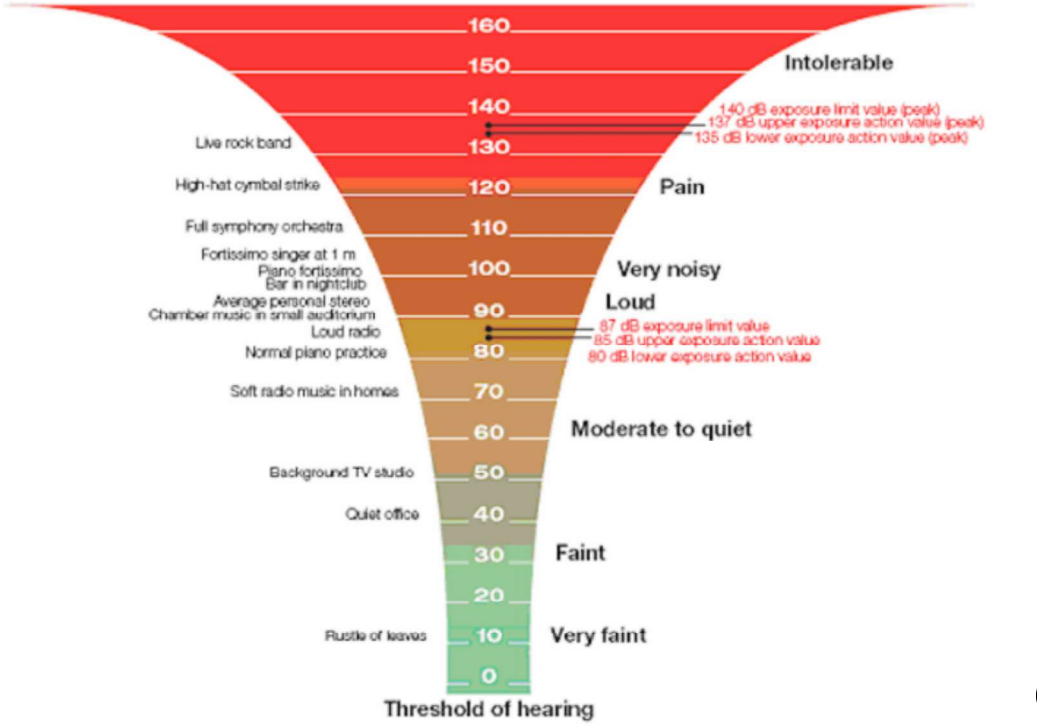
HOW WE HEAR

As we are aware, sound is comprised of fluctuations in pressure. (Compression and rarefaction of atmospheric pressure.) Or in simple terms, a speaker cone moving back and forward, causing vibrations in the air.

When these vibrations reach the ear, they are projected into the ear canal by the outer ear. Once inside the ear canal, the vibrations will then reach the eardrum and cause it to vibrate. These vibrations are transduced into tiny movements, which then go to the inner ear, where thousands of tiny hair receptors transduce these vibrations into a neural signal, which then passes down the acoustic nerve into the brain, where it is interpreted as sound.

These tiny hairs within the inner ear can easily be damaged by loud noises. They will also naturally deteriorate over time, with age. Once they are damaged they do not recover, so be wary of listening at loud levels. Damage to these tiny hairs leads to tinnitus (ringing ears) and permanent hearing damage.

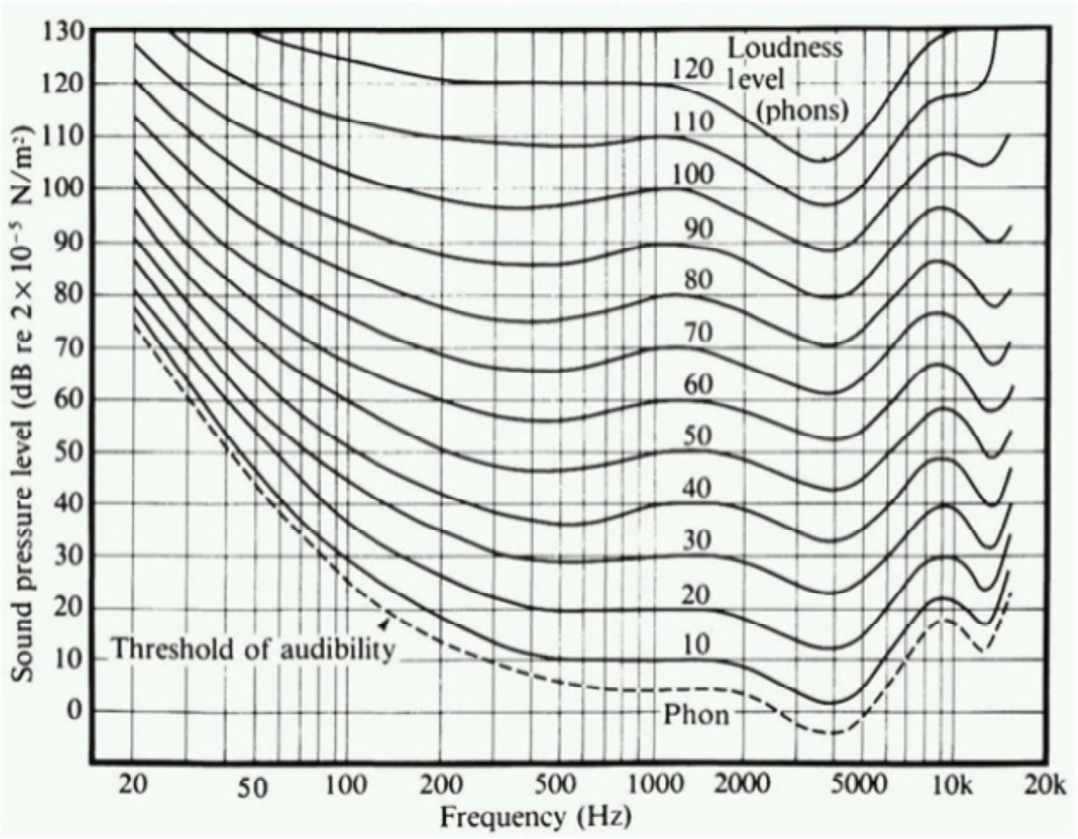
This hearing loss occurs from long exposures of over approximately 90dB SPL. Typically anything over 115dB SPL will cause significant temporary or permanent hearing loss even at relatively short exposures. It's also worth noting that it's generally the higher frequencies that will do the most damage, rather than the sub frequencies, as many people would think.



HEARING IS NOT LINEAR

Human hearing isn't linear. Many experiments have concluded that our hearing is more sensitive around the 3-5KHz range. Frequencies that are in the extreme high or low ranges will often need to be significantly louder for us to perceive them as the same level as a frequency in the 3-5khz range.

This can be shown with this graph of equal loudness curves below.



BASICS OF SOUNDS AND WAVES SUMMARY & KEY POINTS

- The most common waveforms are sine, triangle, square, pulse, saw & white noise
- During mixing we will come across complex combinations of these waves
- Any sound can be created using multiple sine waves. This is the fundamental theory behind additive synthesis
- Sine waves are the purest waveforms, consisting of a fundamental frequency only
- Triangle waves contain odd harmonics and are slightly brighter than sine waves
- Square waves contain odd harmonics and sound hollow; these are good for replicating wooden sounds
- Saw waves contain all the frequencies in the harmonic scale and are very bright
- White noise contains all the frequencies in the spectrum at full amplitude
- Sound propagates through air at 343 meters per second (1125ft per second)
- This can be approximated at 3ms per meter (1ms per foot)
- Waves can be described in terms of amplitude, wavelength, velocity, frequency, timbre, phase & envelope
- A peak measurement is the loudest part of the signal
- RMS means root-mean-square which is a more average way of interpreting a signal
- RMS is closer to how the human ear hears sound
- There's 360 degrees to one periodic cycle
- Hertz (Hz) is the measurement of frequency

- 1Hz is 1 cycle per second
- Wavelength = velocity of sound in air (343 meters) / Frequency (Hz)
- A doubling in frequency is also a doubling in pitch (1 octave)
- The range of human hearing is 20Hz - 20KHz
- Phase is measured in degrees
- Similar sounds that are in phase with each other will sum together
- Similar sounds that are out of phase will cancel each other out
- A common way of representing and manipulating a sounds amplitude over time, is b with ADSR envelopes
- ADSR envelopes and LFO's can also be used as modulation sources
- The decibel is a relative unit of measure to a given reference level
- There are many different types of decibel, within music DB SPL and DBFS are the most common
- 0 DBFS is the loudest a signal can go within software without clipping
- Decibels are logarithmic, a doubling in DB is not the same as a doubling in loudness
- Sound is the compression and rarefaction of air pressure
- Human hearing is not linear; our hearing is most acute in the vocal range of 3-5Khz

CRITICAL LISTENING

Critical listening is the ability to listen to a track in an objective, analytical manner. Detaching ourselves from the music in this way is much harder than first thought, especially if the person listening had a part in the production or mixing of the music.

We need to listen to the track and dissect it with a whole new level of depth and understanding. This allows us to deconstruct the sounds and music we make, in order to identify any problems that we may have within our mix, allowing us to figure out how we can then rectify them.

A keen critical listening ear comes with lots of practice and can take years to develop. There are three main categories that we can divide a sound into when we are making a critical analysis:

- **FREQUENCY**
- **DYNAMICS**
- **HARMONIC CONTENT (timbre)**

FREQUENCY

This directly correlates to pitch, however frequency on its own is only a small part of the sound, as we will soon see in the timbre section.

DYNAMICS

Dynamic range is another fundamental characteristic of sound. Dynamic range simply means the difference between the loudest and quietest parts. This could be of a single sample, such as a snare hit, or it could be used in a more general context, such as the dynamic range of a vocalist throughout an entire track.

Usually a vocalist will fluctuate in volume throughout different sections of the track, meaning that they are very dynamic, and will often use a compressor to keep the dynamic peaks under control.

Another example of dynamic range, which is used in the majority of productions in some form or another, is controlling dynamics of individual sounds to shape how they are heard within the mix.

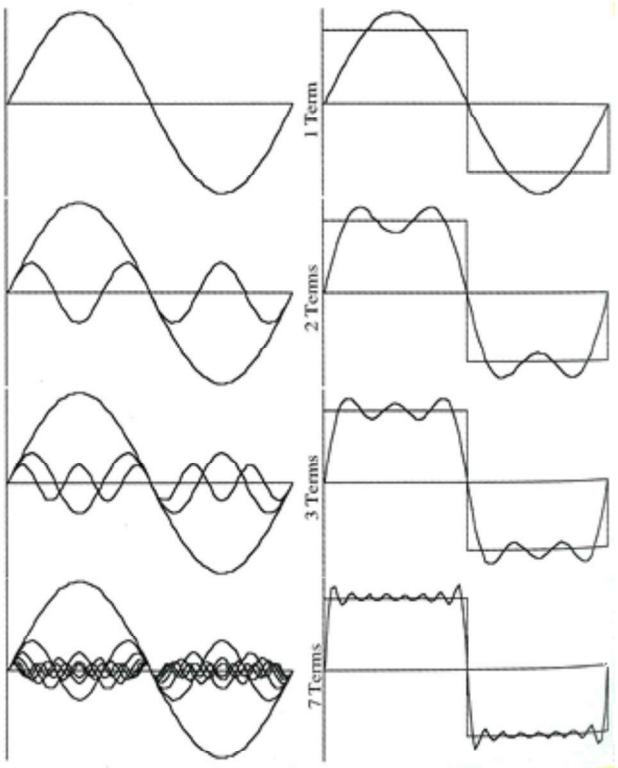
Compressors and envelopes can be used to control the attack and decay phases of sounds such as kick drums. This can be used to soften attacks that are 'peaking,' or 'clipping', as well as to give sounds more bite and punch.

TIMBRE

As we should now be aware, a tone generator playing a sine wave at 440Hz is equivalent in pitch to A4 on the keyboard. If we now double this frequency to 880Hz, we now hear A5 on the keyboard. For every doubling in frequency, we go one octave (12 semitones) up the keyboard.

If we now load an instance of the grand piano in Ableton, and play the same A4 and A5 notes, we can tell they sound completely different to the sine wave. Even if they had the same dynamics as our sine wave, they would still sound completely different due to their harmonic content or timbre.

The extra harmonics are also known as overtones and they layer together to give us our sound. To prove this, we could use a 24dB per octave low pass filter, to remove these overtones until we are only left with the pianos fundamental frequency, which will make it sound very similar to our sine wave. This proves that we can make almost any sound purely out of multiple sine waves at different frequencies, which is the basis for additive synthesis and was theorised by Joseph Fourier (Fourier's Theorem).



We can use Fourier's theorem to use a potentially infinite amount of sine waves to create any complex periodic waveform. In this case a square wave.

In each stage of the diagram we can see how more sine waves are used at varying frequencies to square off the square wave.

TRAINING OUR EARS

A point that we would like to reiterate is that we aren't going to improve our critical listening capabilities overnight.

It takes lots of time and practice to train our ears and we must be training them as often as we possibly can. We can do this by mentally dissecting songs that we hear on the radio or on podcasts. There are also plenty of useful ear training videos on YouTube, which will allow us to hone in on frequency and dynamics.

We can think about the elements such as: frequencies, timbre, dynamics, stereo image and depth. We can also think in terms of different instruments and how they come in and out of the mix during various sections of the arrangement. Although it is very tempting, we should not be using our eyes to mix. We don't use our eyes to hear music when we listen to it, so therefore we should not be using our eyes when mixing because our brain will subconsciously make decisions for us, based on what we see rather than what we can hear.

For example: We may see a cluster of resonant frequencies in a kick drum at a certain frequency range. Many producers are aware that 200Hz is a common area for a build up in 'muddy' and 'boxy' frequencies. We should be using our ears and sweeping through these frequencies to listen for resonances. Just because we are aware of these common problematic areas does not mean that this particular sample is going to benefit from this blanket style of treatment. the ears are our best tool, not our eyes.

IMAGING

Most modern electronic music is listened to in stereo. Stereo is when sound is reproduced to come out of two independent audio channels through a symmetrical configuration of speakers.

This allows for sounds to be placed in a 'space' in the mix between the two speakers, by using panning and relative volume between each speaker. This 'space' is known as the stereo field.

To grasp a better understanding of this concept, listen to this short YouTube clip using headphones:

https://www.youtube.com/watch?v=IUDTlvagjJA

These recordings can utilize stereo imaging techniques to really place the listener in the space.

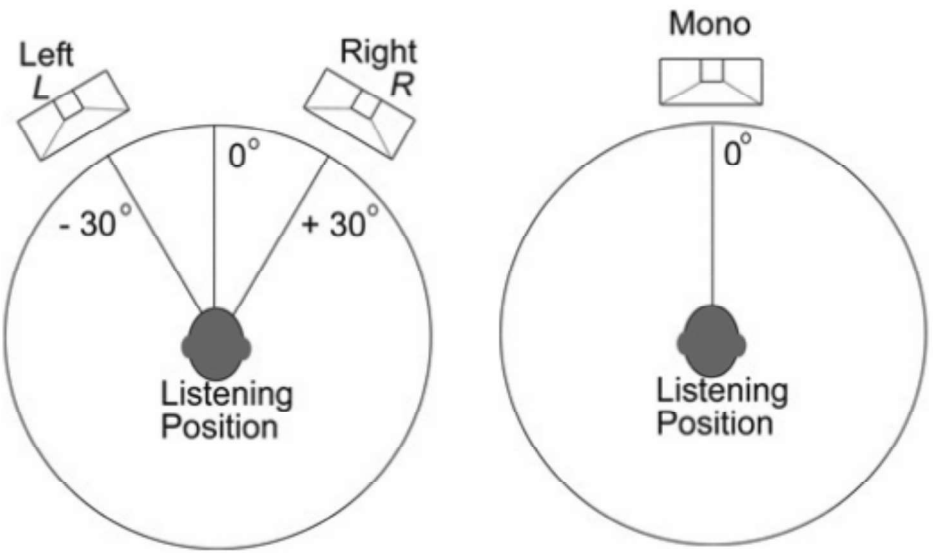
We also have 'mono' or monophonic sound. Monophonic sound is the reproduction of sound using one channel, instead of two independent ones. This is how most club sound systems work, so that the volumes and audio playback is the same, regardless of our positioning within the club. This is worth taking note of, because we don't want any of our music to have any phase issues when summed to mono, which often causes issues such as a 'phasey' sound, peaks, or complete cancellation of certain frequencies.

If we load a track into our DAW, we can easily tell if it is mono or stereo by looking at the waveform to see if there are two separate channels.

When these two separate channels are on one track, this is known as stereo-interleaved. Meaning that there are two channels that are kept together as one individual file, as opposed to having two separate mono audio files on separate tracks, which would require panning to make the full stereo track.

When we place sounds, it is our aim to spread them across the stereo field to create a cohesive stereo image. A good example of this is to think of a live band set out on a stage. If we close our eyes, we can paint a mental picture of what the stage looks like and where the players of the band are situated, purely from the sound and direction of the instruments.

This is yet another tool that we can use as producers to add layers of interest to our productions. There will be a book dedicated specifically to reverb and depth in this mix series. For now, we just need to have a very basic awareness of the fact that there is more than one dimension to a mix. We also need to think in terms of width, as well as direction and height.



SPATIAL AWARENESS

So far we have covered frequency, timbre, volume (height) and direction (including width) of our sounds. But what about depth?

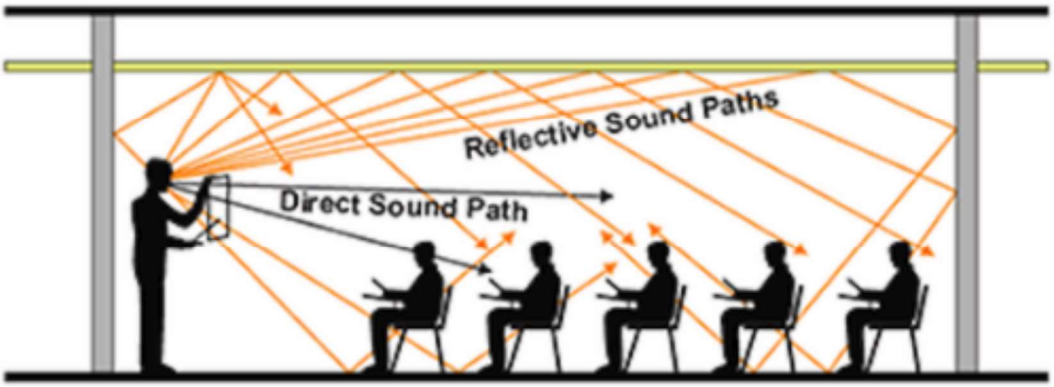
We need to be spatially aware when listening and critiquing mixes. In every day life we hear a natural reverb on every single sound we hear. For sound to reach us, some of it will go directly to our ears, some of it will bounce off the walls of the room, then to our ears, and some of it could bounce around a room several times before reaching our ears.

This is the reason that when we listen to a sound, we are able to make educated guesses as to the size of the room the noise was made in, the type of room, (tiled walls or lots of curtains and carpets) and of course the distance we are away from the sound.

As mix engineers, we can use the fact that our ears have this added skill set of depth detection to create a sense of space and depth within our tracks. We can accomplish this with a few different tools at our disposal. From Reverb (most common), right the way through to compression, EQ, delay & volume.

REVERBERATION – The time it takes for the reflected sound to reduce to 60dBs from the cessation of the original sound signal. (Measured in seconds)

- Reflected sound tends to 'build up' to a level louder than the direct sound. Reflected sounds **mask** direct sound.
- Late arriving reflections tend to **smear** direct sound.



CRITICAL LISTENING SUMMARY

Everything we have covered so far will help us to be able to listen, analyse and critique our mixes, as well as to hopefully help create some great sounding ones. We also need to know that we can trust what we are hearing and know that it is an accurate representation of our creations. If we aren't accurately monitoring our track, then we will have no way of telling how the track will translate to the rest of the world.

CRITICAL LISTENING KEY POINTS

- Critical listening is listening to a track in an objective analytical manner
- Critical listening takes time and practice, paying attention to the frequency, dynamics, and harmonics of each individual sound and how it fits in the mix.
- Dynamic range is the difference between the loudest and the quietest parts in a mix
- Macro dynamics can be thought of as the long-term dynamics
- Micro dynamics can be thought of as the dynamics between different components of a single sound
- Timbre describes the tone and harmonics of a sound
- Mono is the reproduction of sound through a single channel
- Stereo is the reproduction of sound through separate channels
- Mixes can be analysed in terms of height, width and depth
- Reverb and other effects can be used to introduce an element of dimension into a mix

THE ROOM

BASIC ROOM ACOUSTICS

For us to understand how to set our room up, we must first understand what the sound is doing when it is played within a room.
This in itself is a very large topic that we could go into great detail on, but for the purposes of this eBook we will only cover topics necessary to mixing tracks down at home, or in a studio environment.

Firstly, we can state that when a sound is played in a room, it will bounce (reverberate) off the surfaces of the room. These reflections will alter, as well as colour, the sound in certain ways.

The issue we have, is that different surfaces will reflect or dampen the reverb more efficiently than other surfaces. Also the dimensions of the room will cause certain frequencies to produce standing waves. These come in the form of resonances and room modes. It is almost impossible to completely remove reverb, but what we can achieve is a calculated and equal amount of reverb throughout all the frequencies of the spectrum. We also want a nice flat frequency response with no resonances (peaks) or cancellations.

STANDING WAVES

To understand standing waves, we must understand that a wavelength is inversely proportional to frequency.
This means that the lower the frequency, the longer the wavelength. A frequency of 30Hz will have a wavelength of 11.4m

This can be calculated by the formula:

(Wavelength) λ = (Speed of sound in metres/s) 343 / (frequency) 30Hz

343 metres per second / 30Hz = 11.4m Wavelength

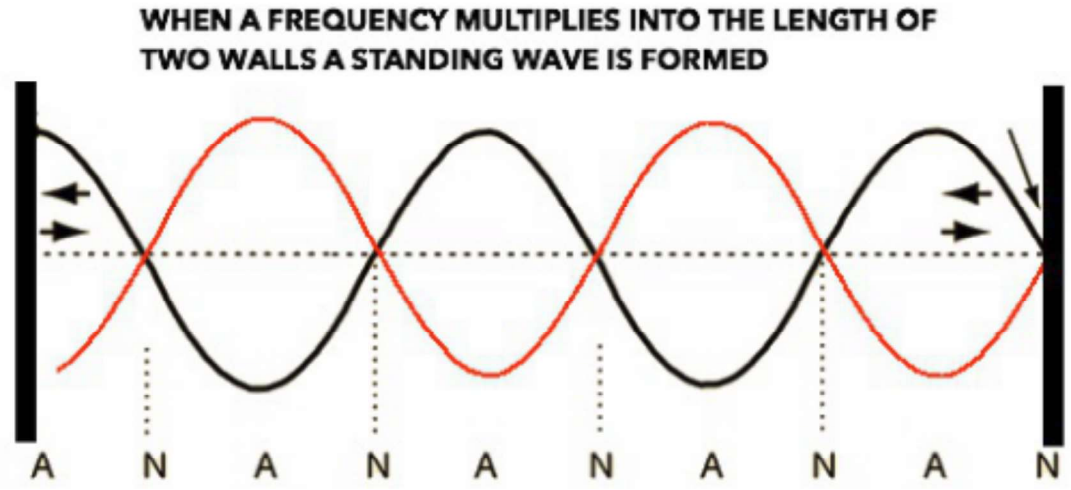
We can see that although standing waves are predominantly a low frequency problem, they can affect any frequency within the spectrum.

When our bass speaker plays a 30Hz sine wave in a room, the wave is going to bounce back and forth off the walls. As it bounces back on it's self, it's going to alter its phase relationship, relative to the part of the wave that hasn't hit the wall yet. This will usually make it out of phase; however, if they are directly in phase, or directly out of phase, then we will get cancellation of frequencies, or we could get a summing of frequencies.

As we walk around the room, certain areas will have almost no bass whatsoever, whilst other areas of the room will have almost double the amount of bass. This is clearly a problem when we are trying to find a tonal balance for our mix, especially when it comes to any parts that are sub 300Hz.

With this knowledge we can do some calculations to ensure our rooms dimensions don't cause standing waves. As a rule of thumb rectangular and odd shaped rooms are quite well suited for monitoring purposes. The worst kind of room shape for monitoring is a perfect square, as this shape causes multiple standing waves at the same frequencies.

ROOM MODES



We can use an equation to find out our room modes and problematic frequencies that will produce standing waves.

In the diagram above, N and A define the nodes and anti-nodes of the standing wave.

Nodes are where the standing wave will be at minimum amplitude. The anti-nodes will be where maximum displacement will occur within a cycle. A good way to imagine this is that the standing wave in the diagram is a set of skipping ropes. Only the anti-nodes will fluctuate in value.

The nodes and antinodes are always going to be located at the same position along the space/medium, hence the term "standing wave."

Frequency = Speed Of Sound / 2x Distance (between parallel walls)

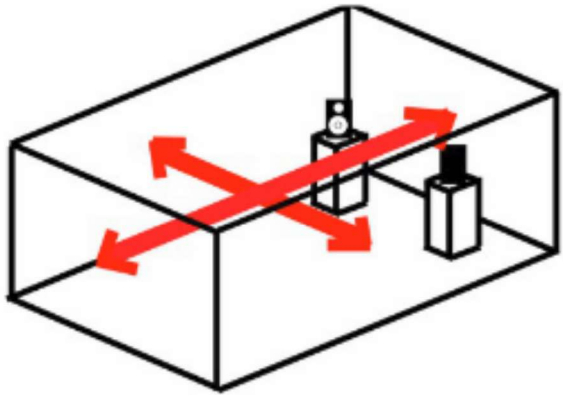
Frequency of standing wave = 343 / 2m x2

In this case we have a standing wave at 85.75Hz.
We can now also multiply this frequency by itself, to get the room modes for the harmonics as well, which will also have an uneven response.

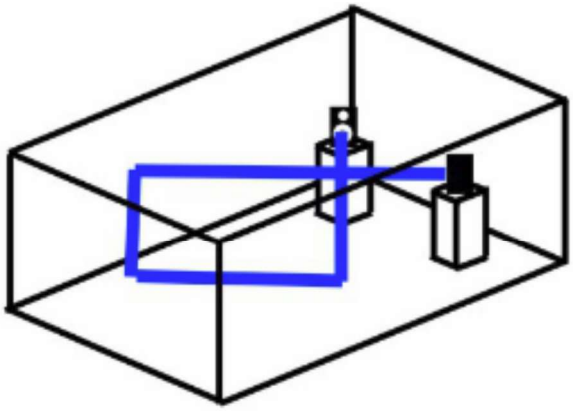
These problematic frequencies only really occur up to around the 300Hz mark. After this point, there are so many standing waves and resonant frequencies, that they actually smooth back out again and don't cause much of an issue. Our main room modes to be concerned with are axial, tangential and oblique.

- Axial room modes hit on two surfaces crosswise
- tangential room modes hit on four surfaces crosswise
- oblique room modes hit on six surfaces crosswise

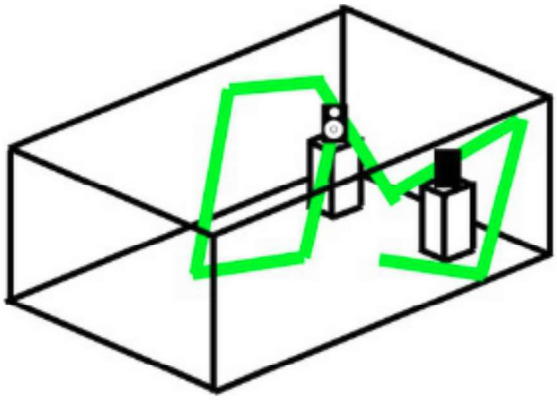
These can easily be calculated using your room dimensions with free online software.



Axial room modes



Tangential room modes



Oblique room modes

THE ROOM SUMMARY & KEY POINTS

- Sounds will bounce and reverberate off the walls and surfaces of a room
- We want to achieve a flat frequency response within our room
- Standing waves are caused by sound waves cancelling each other out or Summing together as they reflect back on themselves
- Wavelength = speed of sound / frequency
- Bass tends to build up in the corners of a room
- Standing waves are most prevalent at lower frequencies
- There will be multiple standing waves within a room
- We can calculate standing waves using this formula

- **Frequency of standing wave = Speed of Sound / 2x Distance (between**

- We can calculate velocity, wavelength and frequency by transposing this formula

Frequency (Hertz) = Velocity (Meters/Second)/ Wavelength (Meters)

MONITORING

COMPONENTS

Here are a few components that should be taken into account when monitoring:

THE COMPUTER/DAW

The computer and the software will be the centre stage of your studio setup. We recommend using a computer with fast CPU processing power and large monitor.

THE DAC

(Digital to Analog Converter) this will often come in the form of a soundcard.

AMPLIFIER

Many speaker systems will have active monitors, so we will not need an amplifier; however, in the case of passive speakers, one will be required.

MIXING CONSOLE

We may be routing our audio through an analog or digital mixing console of some sorts, from here the signal will be split, or routed to different speakers for monitoring playback.

THE ROOM

The room's size, dimensions, and materials within it, will all play a key part in how we hear and interpret a sound.

MONITOR SPEAKERS

When mixing a track, it is crucial that we are monitoring in a good environment, through a decent set of speakers. Most studios will have nearfield monitors, from companies such as KRK, Yamaha or Genelec.

When buying studio equipment, speakers should be right at the top of the priority list. Everything we hear and do will have a heavy dependency on the quality and positioning of our monitoring system. To that end, its important to make sure we save our money for a good set of nearfield monitors with a flat frequency response.

The frequency response needs to be as flat and as true to the actual audio coming out our soundcard as possible. Any colouration, or warming of the signal at this point, would trick us into thinking our mix sounds much warmer & richer than it actually is. This will often result in a colder sounding mix. We may also want to buy a subwoofer as well, for sub-monitoring.

Generally if our studio is a bedroom environment, and our nearfield drivers are around the 8" range, then we probably won't need the subwoofer. We also advise to get a few other monitoring systems. Having a set of headphones at hand for when we want to reference our mix through other sound sources can be really useful. Any set of DJ headphones work fine for this such as Pioneer or Sennheiser's.

We aren't necessarily going for crystal quality here. What we are trying to achieve, is a well-rounded idea of how our mix sounds on different mediums, to aid our objective decision making during the mix.



KRK Rokit frequency response graph

Finally the last monitoring source, and also one of the most important sources, is to invest in a lower quality speaker. We don't mean that it's falling apart. Its cones etc. should still be in good condition. What we mean here is an old radio-type speaker.

We are looking for a cold, flat, and true speaker, which will reveal any mid range flaws within a mix, it should be fairly small and only have one cone. (This negates any issues with crossovers.) It should also not be ported which may cause resonances. (Nearfield monitors often use ports to extend the bass range, which will suddenly drop off at around 40Hz, we don't want a ported design because this will ensure a more even low frequency ramp) We also want to be able to send an auxiliary or Phono cable to it.

If this has any form of bass/treble EQ settings on it, make sure they are set to neutral so as not to affect the signal.

Do not use computer, or iPod docking-station speakers, as these often employ psychoacoustic technologies to enhance the warmth, bass, and stereo image of our mix.

Ideally we should be doing the vast majority of our mixing on the rubbish speakers. Try to do this at the lowest possible volume. (We should set certain reference volumes that we use consistently; this can be achieved using a 'dim' switch, which allows us to switch between set levels.)

Generally a good volume is so that we don't need to raise our voice to hold a conversation over the sound of the speakers.
This will be around 70dBs. Any louder than this level and we will start to induce ear fatigue and hearing loss at long exposures of 8 hours or more.

Once we have done the majority of the mix on the radio speaker, we can then move onto our nearfield monitors. We will find that we have a lot less work to do, than if we were to do this the other way around.

A NOTE ON EAR FATIGUE

The best way of describing ear fatigue is when we can no longer make accurate mix choices or decisions. This is because our ears are very good at adapting to sounds. When listening for long periods of time they can often lie to us, and give a false representation of what we are actually hearing.

The louder a signal is, the faster the onset of ear fatigue. This can be affected by a number of other variables as well, ranging from how much sleep we have had, right through to the type of audio, and room conditions we are working in.

Listening at high volume levels for a long period of time can lead to hearing loss, and ear fatigue, so look after the tools of your trade carefully!

A good example of ear fatigue is a situation that many producers will have come across: We've been pulling an all night session on a track, to get it finished in time for a deadline and have got it sounding amazing. We go to sleep and wake up the next day. We give the track one last listen before bouncing the audio down, only to find out that it now sounds horrendous. This is ear fatigue in all its glory.

MONITOR TYPES & USES OVERVIEW

We can utilize different types of monitors for different purposes when mixing.

1 – THE MID RANGE RADIO SPEAKER

This is going to be how the vast majority of listeners will hear our mix. Think of this as an 'average Joe' of all of the different speakers and methods of listening to our track combined into one neat, well rounded speaker. Whilst this speaker won't sound as good as our main nearfield monitors, it's of just as much importance. This is due to how it will really highlight any mid range issues in our mix that would be much less obvious on the nearfields. If we can get our mixes sounding good on this, then we're well on our way to a decent, final mix-down.

We should be able to mix on these speakers for a full 8-hour session without experiencing ear fatigue, the majority of our big decisions, and balancing should be done on these.

Our radio style speakers should not be placed directly towards our ears, as this will give a clearer sound. We want a listening environment that closely replicates how the end consumers are likely to hear our music when listening back to it. This little trick, of listening to the audio from a distance helps us to listen to our mix objectively, instead of getting over involved.

Due to their small size, they are also not going to be affected by our room's acoustic properties as much, because they will have considerably less low-end. Because of this, these speakers aren't suitable for any mixing decisions under 60-100Hz.

For monitoring below this range, we will need to use nearfields and possibly a subwoofer. The small radio speaker is also not suitable for adjusting the 12khz + range of our mix, due to not being able to reproduce the high-end frequencies efficiently.

Try to split the mixing time, so that roughly 50% of the mixing session is done using these speakers.

Many producers will choose to use just one radio type speaker rather than two. This allows for quicker mono compatibility testing as well as balancing. An iconic speaker that was used for this purpose was the Auratone 5C. Whilst this speaker was certainly not the first choice for a full and non-biased frequency spectrum, It was perfect for highlighting those difficult, muddy, mid-range frequencies, which many people believe can make or break a track.

The Auratone speakers are now no longer in production however good alternatives are the Avantone Mixcubes, the Behringer C5A's, or C50A's. These have been built to replicate the properties of the infamous Auratone speakers.

Take the time to listen to plenty of reference tracks on the portable speakers, also remember to switch between the nearfields and the radio speaker whilst listening to the same track, so we can really hear the difference in sound quality.

Monitor switching controllers can be really useful for flicking between different speaker types as well as setting reference levels for our different speakers and stereo/mono checks.

A well-respected monitor controller worth investing in is the M-Patch 2 from JBL. This allows for switching between two inputs and outputs, and a handy stereo/mono switch.

A good test, to reference the low-end on our portable radio speaker, is to play a reference track and push the volume till the track distorts. Take note of how loud it is. Now do the same with our own production and see if this goes louder or quieter before distorting.

The distortion is usually caused by the high-energy, low-end Frequencies eating up headroom. So if our mix gets a lot louder without distorting, then this is usually a good sign that our mix doesn't have enough low-end. And likewise, if it distorts earlier than expected, then we may have too much low-end in our mix.

2- NEARFIELD MONITORS

This is the type of speaker that we will find in almost every studio, they have a very neutral and linear frequency response and add little to no colouration or warmth to the sound. This gives us a very true representation of what our audio sounds like. It's common to have two sets of nearfield speakers, and as we mentioned earlier, some popular speakers are Yamaha's, KRK's and Genelec's.

These speakers will give us around 4 hours of mixing without causing ear fatigue and we should be aiming to use these for around 40 percent of our allotted mixing time.

This speaker needs to be able to respond well to transients and also have a wide bandwidth to be able to accurately reproduce the full frequency range. They are designed to be fairly close to the listener, at head height, in an equilateral triangle formation. Speaker setup will be explained shortly.

3 - SUB WOOFER

When using nearfields, we are aware many models have the drawback of a relatively poor bass response due to their size. The frequency response of many smaller nearfields drops off below around 40Hz. We can add a subwoofer to our arsenal to help recreate these frequencies, to aid us in the mix process.

Issues we will need to consider, when using a subwoofer are:

- Where we should place it
- At what frequencies should it crossover with the monitors
- Is it in phase with the nearfield monitors?
- Is the level set correctly in relation to the nearfields?

If we cannot accommodate a subwoofer, then try a professional set of monitoring headphones that can really help for listening and balancing bass frequencies when we don't have the luxury of an acoustically treated room.

4 – HIGH END STUDIO SPEAKERS

Finally we have the top of the range speakers that we can find in high-end studios. These are often built into the walls of the studio, or are large floor standing speakers.

Other than to show off to clients, These speakers will help us to pick out the extremely fine details In a mix, for making micro changes that we wouldn't be able to notice on a set of nearfields or radio speaker.

We can also use them for precise surgical EQing, as well as getting our extreme low, and high-end tonal and volume balance adjusted correctly. We should not listen on these speakers for any more than an hour. The final 10% of our mixing time can be done on these speakers.

Our audience will not have the luxury of listening back to the mix on these speakers, so its important to understand the concept of getting our mix to sound good on systems that consumers are likely to use, rather than an expensive studio sound system. This is especially prevalent in the bass frequencies, where we may need to add extra harmonics to a sub-bass, so that it can be heard through speakers with smaller drivers.

SPEAKER SWITCHING & SET LEVELS

Many of us will monitor at loud listening volumes during the compositional, creative and arrangement stages of our production, to help us get inspired, and build a vibe into our track.

This is absolutely fine, but when we are mixing we need to adhere to a different set of rules. By the mixing stage, all of the compositional decisions should have been made. Many producers will render their tracks to audio to commit to the mixing stage; this also prevents any further tweaking.

When we are mixing, we are generally not adding anything extra to the track other than FX. Our job is to work with, and improve what is already there, to make it sound as good as it possibly can. For this reason, we need to monitor at low volume levels.

Ideally, we should always be able to hold a conversation over our monitoring levels. If possible, keep it on the lowest setting possible, whilst still being able to mix. This will pay dividends when we eventually turn the mix up later on.

As we monitor at louder volumes, our mind gets satisfaction from the increase in volume, tricking our brain into thinking that the mix sounds better, or punchier than it actually does, when in reality the

only change is volume. Try this test with a reference track to see for yourself.

Getting the mix right at a lower level will help it to sound much more well rounded and fuller when played at louder levels due to the way the human ear perceives music at different volumes. (The Fletcher-Munson curve)

We are fully aware that it is unlikely that everyone will have the willpower to abide by these rules, So we have allowed some flexibility for when its okay to listen at loud volume levels:

- In any other stage of production, other than mixing or mastering
- At the very end of the mixing session
- At the very end of the day before we go to sleep
- Before we go for extended breaks from mixing

Also remember what has been said about dividing our mixing time between the different speaker types.

50% of the mixing session should be done on the radio speaker. This is for the big decisions and the majority of our balancing and tonal sculpting

40% should be done on our nearfields, for further in-depth mixing & fine tuning that requires a higher quality driver.

The final 10% should be done on high-end studio speakers for any fine-tuning and extreme low & high-end adjustment if necessary.

Remember to A-B between different speaker types to see how the parts that we're working on translate in different speakers.

THE LISTENERS PERSPECTIVE

End consumers aren't going to be sat in a studio every day with the luxury of a beautiful set of speakers. The chances are, they will hear our track in a car, on a radio, on their iPod docking station, or on headphones.

For these reasons its important to get the mix sounding perfect on a poor sound system at low levels. That way, if we can get the kicks to punch, and the vocals to sit right throughout the track at a low level, when Joe Bloggs walks past a house party from outside, he is still going to be able to hear the important elements of the track.

A good test is to turn our speakers up fairly loud, go outside the room and shut the door. How does our track compare to our reference tracks from outside of the room? This is also a very useful technique for checking the kick and bass balance.

A final reason for monitoring at low volumes is because at loud volumes the ear can actually create its own harmonic distortion and make up sounds that aren't actually there.

For example; if we play a 1KHz sine wave at low volumes, then it sounds like a 1KHz sine wave as expected. However, if we begin to play it louder, we can start to hear a combination of harmonics as well, such as 2KHz and 3KHz sine waves.

This also works the other way round, so if we have the harmonics, but not the fundamental, we can trick our ears into thinking that the fundamental note is actually there. This psychoacoustic trick is used in small speaker systems that can't reproduce low frequencies to make sounds more bass heavy than they actually are. This is known as the missing fundamental. This technique is employed in certain bass enhancing plugins such as Maxxbass and Rbass from Waves.



Waves RBass

REFERENCE LOCATIONS

Reference your tracks in unusual places, where the consumer is likely to hear your track, such as in a car. Not only do most cars have good audio systems, but they are also useful places for checking the imperfections of our mix;

- There's a high chance that we are fairly well acquainted with the sound of our car's audio system.
- It gets us away from the studio
- A cars interior is usually soft, which means it will be absorbing a lot of the reverberated sound, this coupled with the lack of parallel walls make it a good listening environment

SPEAKER ORIENTATION

When setting up our speakers, we should avoid mounting them on our desk or table, because this may cause vibrations that could cause further acoustic issues.

If we must have them on the table, then make sure they are isolated from vibrations, by using rubber or foam mats.

Speakers should be placed on speaker stands, most speaker stands will have some form of spike between the bottom of the stand and the floor to help isolate the vibrations, We should also have a set of cones between the speaker and the stand. If this is not possible, rubber or foam will suffice.

Ensure that your speakers are the right way around. This may sound silly at first, but speakers are designed to project the frequencies in a certain way. By having the speakers in the wrong orientation, or upside down, they may sound different. Most speakers are designed so that the tweeters will be above the main driver.

The angle at which the speakers are set, as well as the height, will affect the colour and stereo imaging of your music. Generally nearfield monitors are designed to point directly towards the listener with the tweeters at ear level.

SPEAKER POSITIONING

We should set our speakers up using a few simple and effective guidelines:

- Set the speakers up in an equilateral triangle to the listener
- The speakers should be facing directly towards the listener
- A rough guide for distance would be a 3ft gap between each speaker and the listener. This is subject to the desk, speakers and room

Any closer than this, and we would perceive a very narrow stereo image, causing us to compensate by over-widening. Likewise, if the speakers are too far apart, we will experience a wider stereo image with a hole in the centre.



Ensure that our speaker gain pots are set correctly on the back of the speakers, having one knob set slightly different to the other one will create an unbalanced or lopsided listening environment. Many gain knobs on the back of monitors will have a default position at 0dB's, which will click into place.

MONITORING SUMMARY & KEY POINTS

- The main component of a studio are, monitors, amplifier, D-A converter, mixing console, workstation/computer and the room itself
- Look for monitors with a flat frequency response
- Ensure the monitors can accurately reproduce the sub bass frequencies
- Speakers with drivers of below 8" will usually require support from a sub-woofer
- Reference & monitor the mix through multiple sound sources
- There is a significant difference in perceived loudness between 70 - 90 dBs SPL
- Monitor at a level where normal talking can still be heard over the music
- Monitoring at loud levels will increase ear fatigue and the risk of permanent hearing damage
- Take regular breaks when working on a track to allow the ears to reset
- Use a DIM switch to alternate between set volumes when mixing
- Be aware of monitoring at different levels and the effect of the fletcher-Munson curve
- There are occasions when it's okay to listen at louder volumes, such as during compositional tasks, at the end of the day, or before going on a long break
- Use a basic radio speaker for the majority of big decisions (50% of the mix process)
- Use the nearfields for 40% of the mix process
- Use larger professional studio speakers for the final 10% if you have them

- Always have the end consumer and how they will hear your track in mind when mixing
- Speakers should be placed on speaker stands or on the desk
- Use rubber isolators/de-couplers to prevent speaker vibrations
- Ensure speaker orientation is correct
- Tweeters should be at eye level
- Nearfields are designed to face the listener directly
- The speakers should be equal distance apart (around 3ft)
- Place speakers in an equilateral triangle from the listener

ROOM ACOUSTICS:

STUDIO MONITORS - WHAT NOT TO DO

Because monitoring mistakes are very common, we have compiled a "how not to monitor" guide rather than how to do it properly, so you can quickly see if you are making any silly mistakes, as well as find answers on how to fix them.

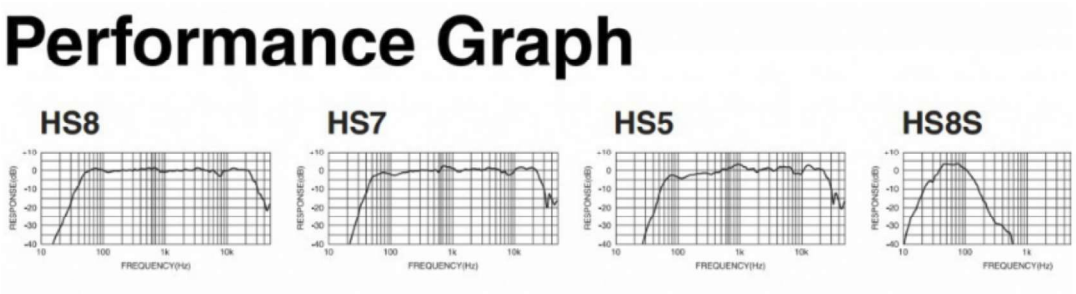
Setting a budget and choosing our studio monitors should be up there as a 'high priority task' for our home studio. Producers often overlook the setup of the monitors; every single decision we make will be reliant on the sound of the speakers in our room. So it's absolutely vital that you get the best speakers we possibly can for our price range; however, having the best quality studio monitors counts for absolutely nothing if we have a problematic room setup.

SPEAKER SELECTION

Research speaker's carefully before buying. Some speakers lend themselves to certain types of music. An important specification to look at is the frequency response of our speakers.

We want this to be as neutral as possible, whilst still extending low into the sub range.

This means that the speaker should give us a true sound and level at any given frequency, whereas consumer speakers, such as Hifi's often have a non-linear response, which will accentuate high and low frequencies to make the signal warmer & louder. Also pay attention to the type of drivers in our speakers. A general rule of thumb is that the larger the driver size, the better our speakers will be able to reproduce the low-end frequencies.



The Yamaha HS series Frequency Response. A good set of monitors should have a flat, non-biased frequency response. The bigger the drivers, the lower the speakers will be able to go in reproducing bass frequencies. The HS8S is a sub-woofer designed to aid nearfield monitors with drivers of less than 8"

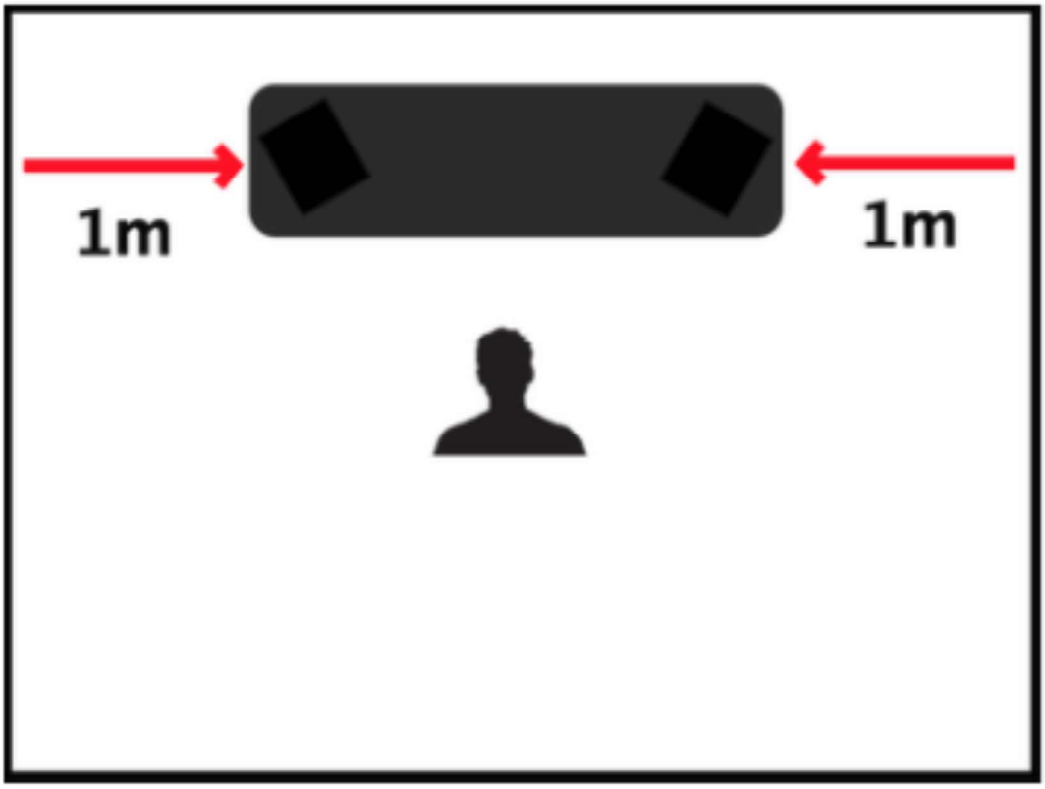
KEEP THINGS SYMMETRICAL

This can be a hard task if we have unusual room dimensions, but it's important that our speakers are in symmetry with the dimensions of our room. This means that ideally we want our desk in the middle of a rectangular room, with our speakers on either side, in an equilateral triangle shape from each other to our head.

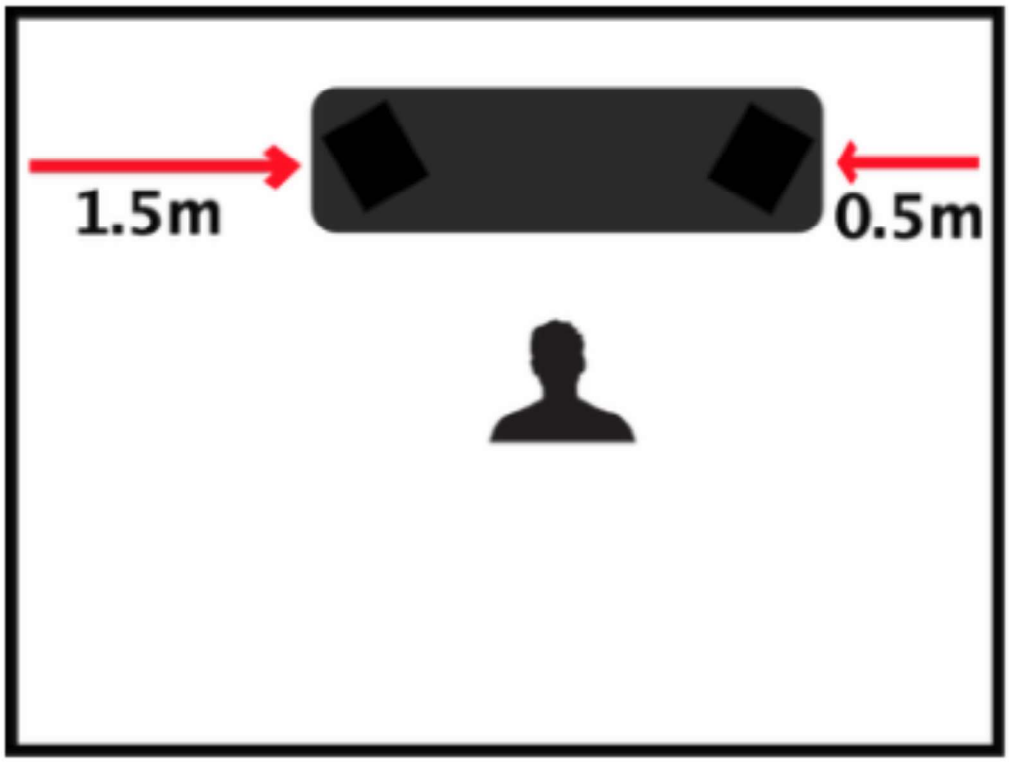
If the left speaker is 1m from the left hand wall, then the right speaker should be exactly the same. This is also the case for the back wall.

If our studio is in the corner of a room then we can still achieve the same symmetry by angling off diagonally from the corner of the room. The reason we do this is because no matter where we place the speakers in the room, they will always be affected by the room's boundaries to some degree.

By positioning the speakers so that they are symmetrical, any effect on the left-hand speakers' audio should be very similar to the right speakers audio. This means that when we make adjustments we won't be doing it in a lopsided manner, due to compensating for one side of our room sounding different to the other.



Correct: Symmetrical alignment of the speakers in your room

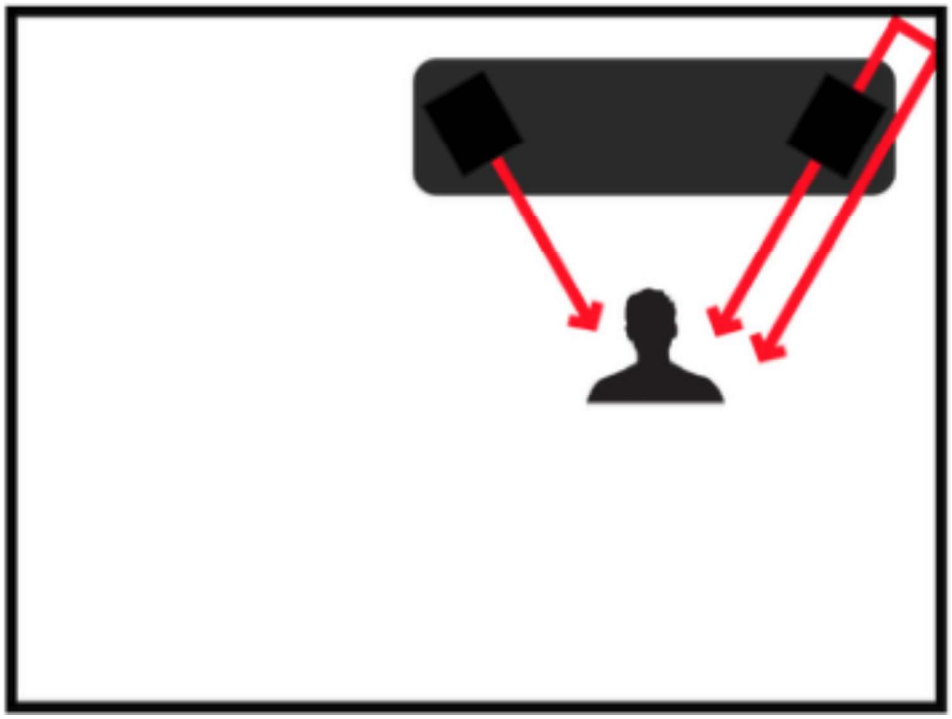


Incorrect: This alignment will cause the speakers signals to be heard differently as they are blended with the reverberated signal of the room in different ways before reaching the ear.

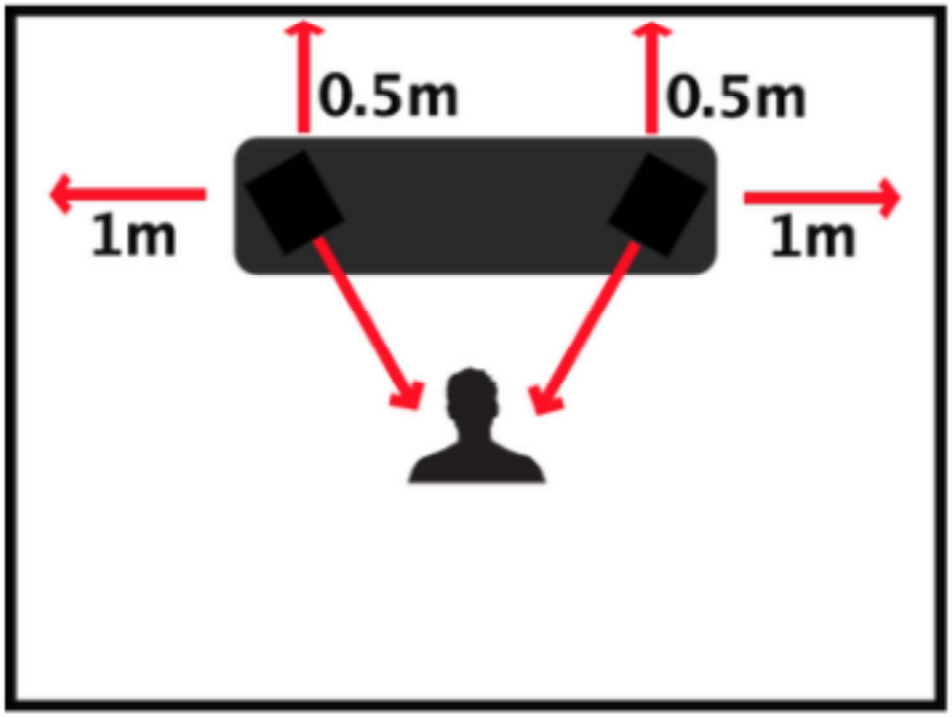
AVOID WALLS

Placing a speaker near a wall will usually cause the bass of the speaker to get louder. This is because of standing waves.

We suggest making sure the speakers are at least a foot away from any wall. Avoid corners at all costs, as they will make our bass response very inconsistent across the low frequency range, due to most speakers' sub ports usually being located on the back of the speaker. If this can't be avoided, then try to dampen the effect using some acoustic treatment in the corners, to try and eliminate the reverberated standing waves.



Standing waves causing a rise in volume in the corners of the room & an uneven frequency response due to being too close to the walls.



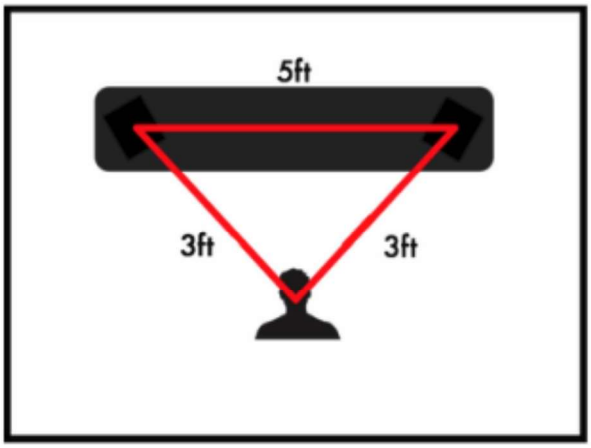
An ideal situation, keeping symmetry with the shape of the room whilst also staying away from any walls. Think about investing in some speaker stands & acoustic treatment if you need to keep your speakers away from corners.

AVOID POOR ANGLES

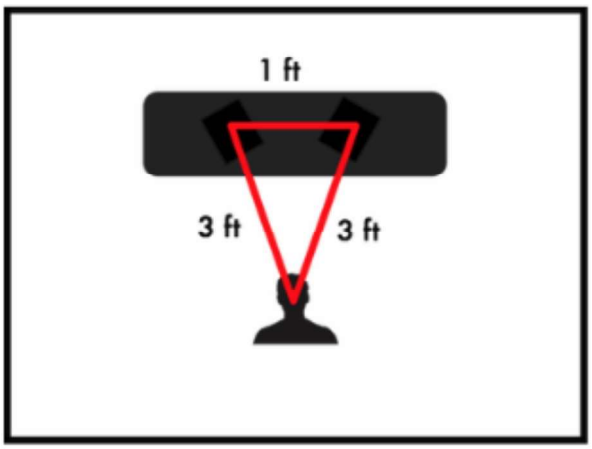
Our speakers ideally want to be positioned in the shape of an equilateral triangle, roughly 3 feet from each other, and 3 feet from the listener. This creates the least amount of high and low frequency interference, and also positions the listener's head nicely in the sweet spot.

If our speakers are placed too far apart, this will create a hole in the centre of the mix, which will cause a producer to over compensate by creating a narrow mix-down. Likewise, having the speakers too close together, means that when the producer listens to the audio, it will all be coming directly from the front, so the tendency will be to pan extremely wide, which will make the mix very unfocused and can cause problems when summed to mono.

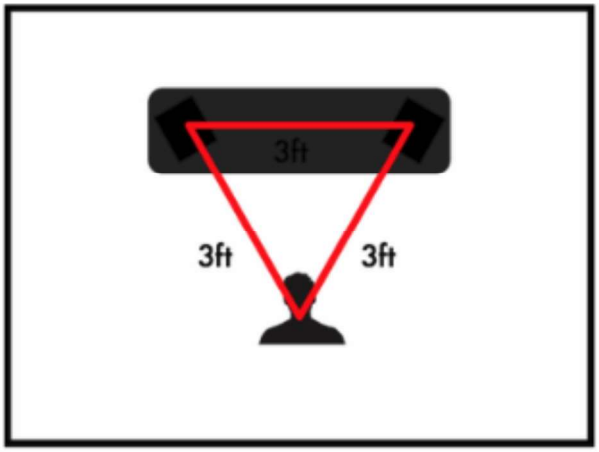
A further point is to ensure that the speaker's tweeters are at eye level and are facing the listener directly. Some OCD producers will want their studio speakers aligned to the wall, however not only will this drastically effect the bass response, but it will also cause the mix to sound duller and less detailed than it actually is.



Incorrect: speaker placement too wide causing a hole in the mix, the producer will over-compensate by panning his instruments more central which will relate to an overly narrow mix.



Incorrect: speaker placement too narrow causing the producer to over-compensate by panning his instruments to the extreme. This will result in a wide & unfocused mix



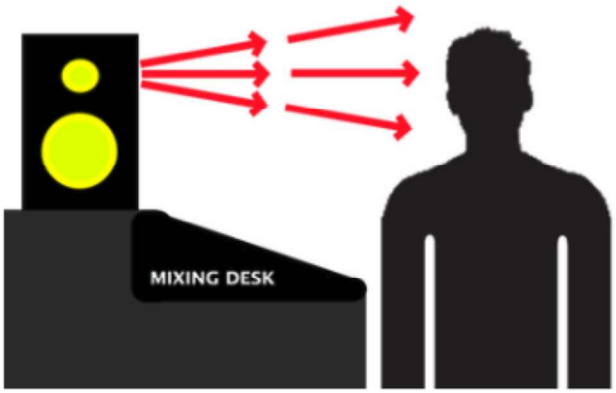
Correct: Speaker placement in an equilateral triangle between the listener and the two speakers. The recommended distance is 3ft apart. Symmetrical with the room and at least a foot away from walls.

AVOID PROBLEMATIC REFLECTIONS

As well as our main room walls. We will also get reflections off of surfaces within the room. The main surfaces to be aware of are the surfaces between the listener and the speakers, this could be a computer monitor, mixing consoles, keyboards, desks or racks.

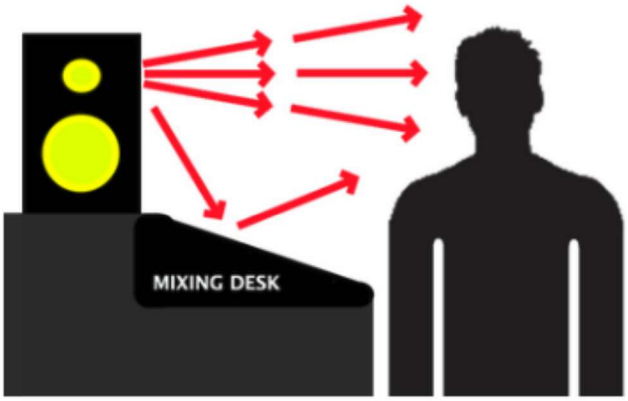
Try to keep these bits of equipment out of the way of the speaker's projection area. Clearly not all of these items can be removed, but some producers have even gone as far as to add acoustic foam around the main areas of their desk to try and reduce these early reflections.

This is also why having the tweeters at eye level is so important, as this ensures that the audio from the speakers will travel directly to the listener, instead of bouncing off other surfaces first.



Speakers tweeter set at eye height. different speakers have different dispersal angles. Typically monitors are designed to travel directly to the listener with a narrow dispersion.

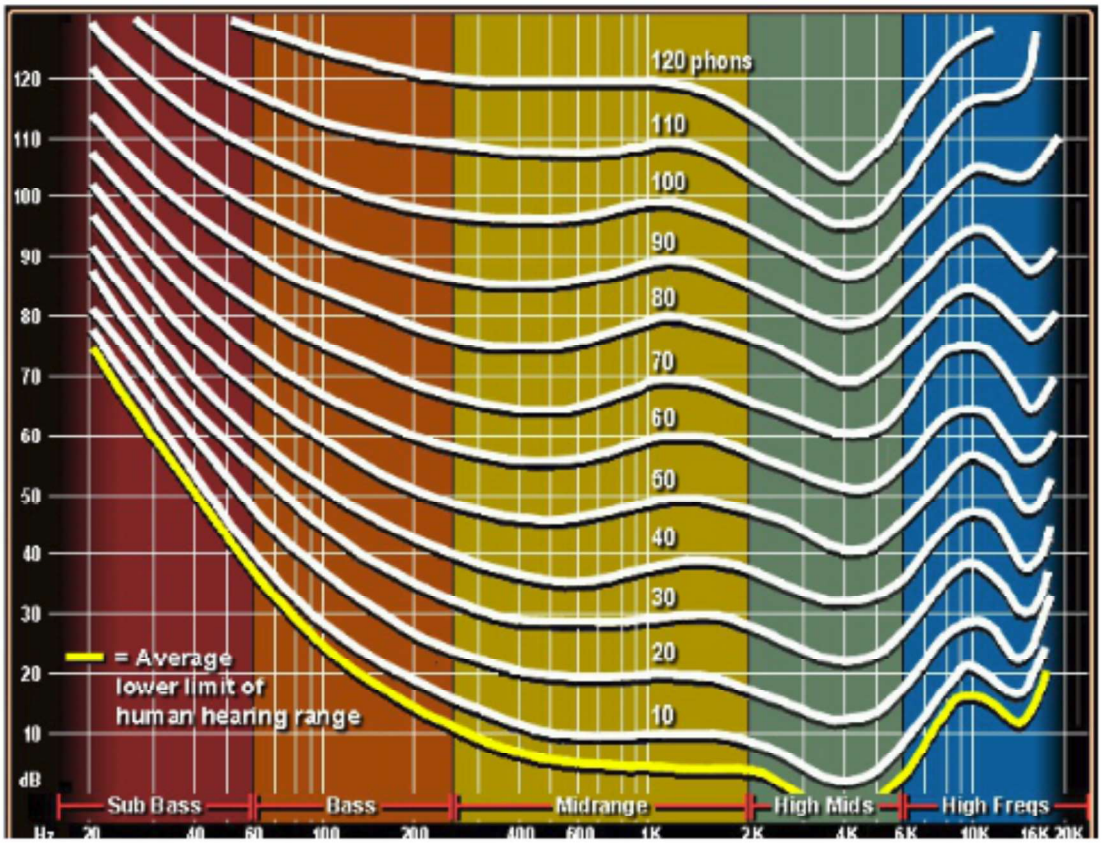
Reflective surfaces in between the listener and monitor can cause interference. Try to remove any clutter or objects to keep this interference to a minimum



EQUAL LOUDNESS AND LEVELS

Our ears do not perceive the loudness of sound in a linear way. This has been proved by equal loudness, or fletcher-Munson curves. The basic theory is that human ears are more sensitive to the high and low frequencies of the spectrum at higher dB SPL. (Decibels sound pressure level.)

Within the studio, this equates to the producer thinking that their bass is really loud because they are monitoring at 100+ dB SPL, when in fact, if they reduced the volume to 80 dB SPL, they would realise that their mix is actually lacking in low and high frequency volume. It also goes without saying that monitoring at loud volumes will increase the onset of ear fatigue, tinnitus and hearing loss.



A graph showing the equal loudness curves and main frequency ranges.

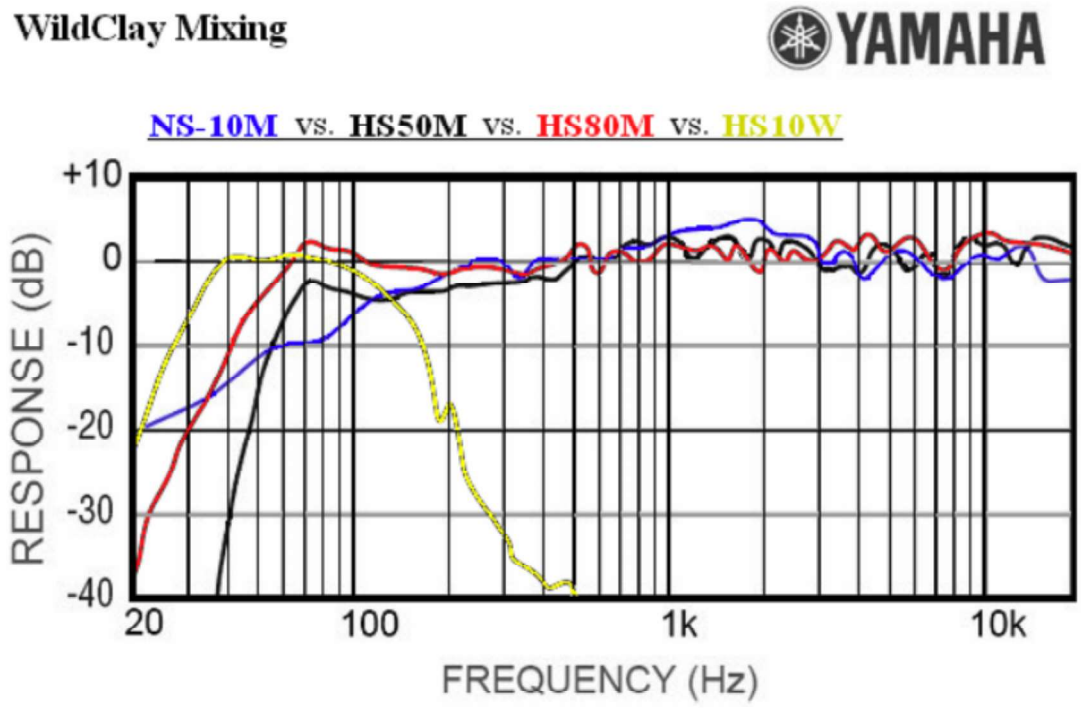
TOO MUCH LOW END

If our monitors have small drivers, which are less than 8" then we can consider adding a sub-woofer to our setup. The sub-woofer isn't going to be used like many car sub woofers are; which is to add a mass amount of bass volume.

The subwoofer is simply there to extend the low-frequency response of our studio, due to our small monitor drivers not being able to reproduce the low signal. This means that when we set up the subwoofer, it shouldn't actually add any extra volume. This can be checked with a sound pressure level (SPL) reader.

We want to get the level as close as we can to the level of the rest of the sound system without the sub. This sub should also be set up so that it's only receiving the necessary frequencies it needs (the crossover) which will usually be set at around 80Hz.

If you follow these few basic rules when setting up your speakers then you will be well on your way to a lovely sounding studio.



Using a Sub-Woofer (HS10W) marked in yellow to enhance the low end frequency response of a monitor such as the HS50M

STUDIO MONITORS SUMMARY & KEY POINTS

- Hifi's and docking stations use psychoacoustics to colour and warm up a signal
- The monitors should be a true representation of the signal
- Larger driver sizes will be better at reproducing the bass frequencies
- Keep symmetry when placing speakers between walls
- Avoid close proximity to walls and corners, especially rear mounted speaker ports
- Use acoustic treatment such as bass traps to improve a room's frequency response
- Avoid poor speaker angles
- If the speakers are too far apart, the mix-engineer may accidently produce a mix which is too narrow
- If the speakers are too narrow, the mix-engineer may compensate by producing a wider mix
- Avoid placing things between the speaker and the listener that may cause the sound to be affected in any way
- Extreme high and low frequencies will need to be much higher in energy to be perceived as the same loudness as other parts
- Ensure the sub-woofer is set up to match the volume and improve the low-end reproduction, rather than just boosting the bass volume

ACOUSTIC TREATMENT

Now that we have our speakers set up, orientated and placed correctly, we can now think about how we are going to treat our room to best cope with the different issues we may have to resolve, such as: standing waves, echo, reverb, and any side reflections.

STANDING WAVES:

Standing waves are created from waves bouncing off walls. These will affect the whole frequency spectrum, however the mid and high range standing waves are less troublesome, and much easier to deal with than low frequency standing waves.

If we play a 60Hz tone and walk around our room, we may notice a change in volume as we move around. This is the doubling and cancellation of waves.

ECHO AND REVERB:

These will add to a signal as it bounces off the surfaces of a room, which will add to the tail of the sound. This can sound similar to fast fluttering or ringing and can be heard if we clap our hands as we move around a room. Try clapping near a corner to see if you can hear the difference.

SIDE REFLECTIONS:

Side reflections are when reverberated, or reflected signals have a time-delay compared to the original audio. These micro-changes in phase will cause the audio to effectively comb filter itself, as it doubles or cancels certain frequencies. This can lead to a smeared stereo image.

ACOUSTIC TREATMENT GUIDE

The majority of producers don't have the luxury of a professional studio. Most will have a small to medium sized, untreated bedroom, which will be used for mixing tracks down in.

Now is the time to get these issues sorted, lack of acoustic treatment is one of the biggest causes of frustration during the mixing process. Over the next few pages we will be seeing how we can vastly improve the acoustics of our listening environment without spending loads of money.

Following on from our last section, we have a few basic rules for acoustic treatment and room setup:

- Use the longest length of the room
- Make sure our speakers are at equal distances from the walls
- Speakers must stay away from the walls and corners
- Speakers should be facing us directly
- Speakers should be in an equilateral triangle formation with the listener
- Each side of the triangle should be equal in length

EARLY REFLECTIONS FROM THE SIDES

This problem is caused when we don't have any absorbers on our adjacent walls. This means that early reflections will bounce off these walls before reaching the listeners ear.

This extra distance that the wave has to travel will add a miniscule time delay onto the wave, which will alter its phase relationship with the direct wave, resulting in some nasty phase issues. This must be avoided by using acoustic absorbers to dampen these early reflections. To make a simple absorber you will need some lengths of wood for the frame, a roll of insulation, cloth or canvas, nails and mounts.

Too much insulation can make a room sound very ‘dead’, in this case try using some sound diffusers as well. The lengths of the blocks are specifically calculated to diffuse certain frequencies in a slightly more natural way.



An acoustic diffuser

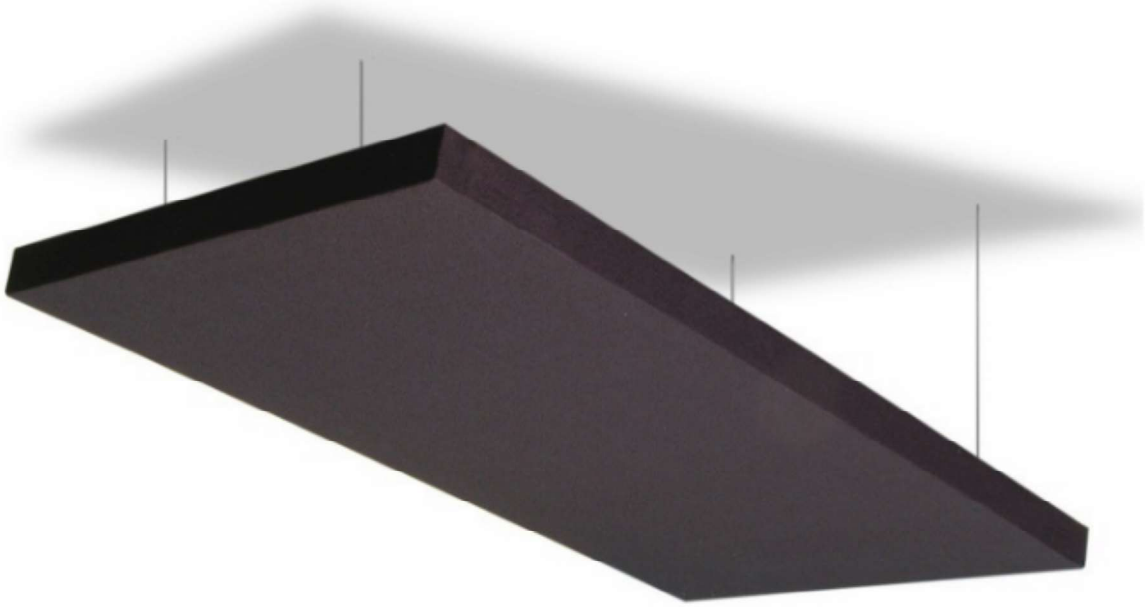


Acoustic treatment panels

EARLY REFLECTIONS FROM THE CEILING

We also need to remember that sound travels in all 3 dimensions, so we need to think about the ceiling and the floor. Flooring is quite a hard variable to control, but it helps if the room is carpeted. If it is a tiled, or wooden floor, then think about investing in a large thick rug, which should help to dampen those frequencies that are likely to bounce off a hard floor.

As for the ceiling, we can suspend an absorber similar to the one used for the walls. This should be placed in the space between the speakers and where you sit. The absorber can be placed so there is a gap between it and the ceiling. This can then be suspended by hooks as seen in the picture below.



A 'cloud' type acoustic treatment used for a ceiling

LOW-END RUMBLE & STANDING WAVES

These can be greatly reduced by using bass traps in the corners of the room. To do this, we can use any form of insulation that can be cut or fixed into all the corners of the room. These bass traps can be bought online, or we can try and make something ourselves, using slightly harder density insulation pads.

Installing more bass traps isn't going to reduce the level of the bass in our room; it is simply going to balance it. So basically we can use as much as we want without having to worry about killing off the bass in our room.

Bass frequencies have very long wavelengths, which makes them more of a structural problem that involves altering room dimensions more than anything else. The best remedy is to get the right dimensions for your room: Using 2" thick acoustic foam on the walls is only really going to solve mid and high frequency issues.



Bass traps



Acoustic panels

FURTHER ENHANCEMENTS

We can cover the back wall with any leftover treatment pads. We should be aiming to cover at least 20% of the surfaces of our room.

Make sure there is nothing in our room that is likely to cause vibrations from bass frequencies. For example: paperclips in a metal container on the desk, or any metal or glass items that are touching each other.

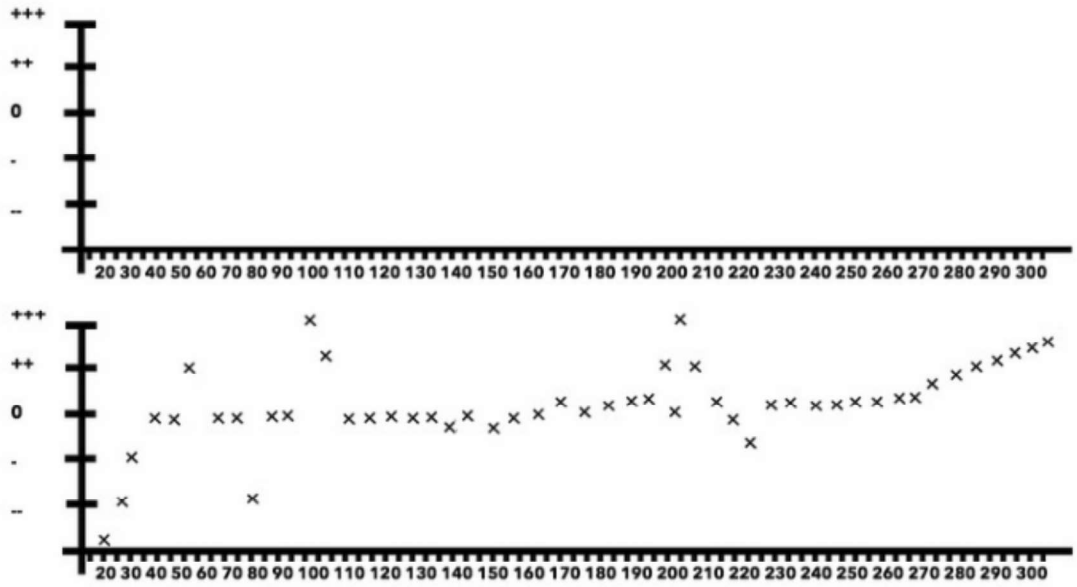
If there are any vibrations occurring, rubber is the perfect material for absorbing some of the vibrations and decoupling the sound from the material. Finally, we can double check our speakers are isolated from their stands, as well as ensuring that stands are also isolated from the floor. If we have a window in our room, try to make use of a long set of curtains to prevent waves bouncing off the glass windows and producing secondary reflections. (This is why recording studios control room glass is usually angled)



Different Types of acoustic isolators & de-couplers

TESTING ROOM ACOUSTICS – TEST TONE GRAPH PLOTTING

Now that we have our room treated, we are going to test how it sounds and plot a low-frequency response chart. We will need to draw a graph that looks like this:



Firstly play a sine wave, starting right down at 20Hz. Turn this sine wave up just enough so that the volume is as low as possible, whilst still being able to hear it clearly. The reason we do this is because it will be easier for our ears to detect changes in volume at these lower levels.

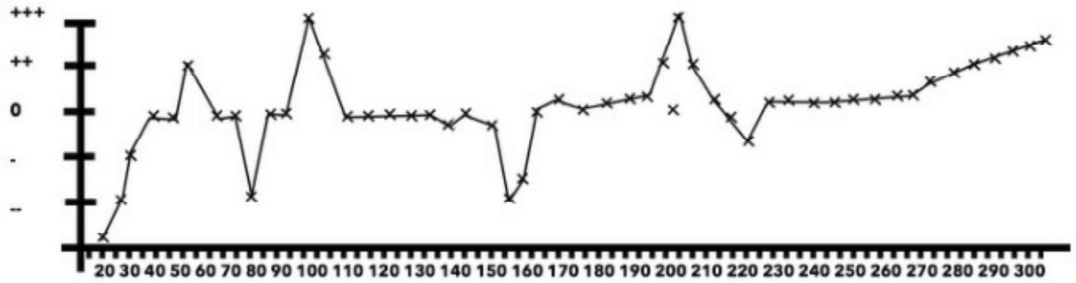
Notice that on the graph we have only plotted frequencies from 20Hz to 300Hz. This is because these low-end frequencies are going to be the most disruptive to the acoustics of our room.

Now we can increase the frequency of our 20Hz sine wave, whilst keeping our head directly in between our speakers. (In the natural position that we will be mixing in.) This is important, because moving our head to any other point in the room, even by a few inches will drastically affect the frequency response curve.

As we increase the frequency, we are going to mark on the graph whether the tone gets louder, quieter, stays the same, or if we cant hear it at all.

This will allow us to map out the resonant frequencies and the troughs in our audio. This can highlight to us where we may have standing waves that are cancelling and doubling at certain frequencies.

We can now join up the dots that we have marked on the graph to give us an approximate idea of our room modes. This coupled with our knowledge of standing waves, room dimensions and treatment should allows us to bring our listening environment up to more of an acceptable standard.



From this graph we can see that we have a room mode at approximately 50Hz, which is also causing problems at its harmonics of 100Hz and 200H. We also have a cancellation of frequencies at around 80Hz and 160Hz. This response graph can change from even the slightest head movement in the room.

ACOUSTIC TREATMENT SUMMARY & KEY POINTS

- Standing waves affect the whole frequency spectrum
- Standing waves are most problematic below 300Hz
- Echo, reverb and side reflections can cause phase issues and comb filtering
- Set your studio up so that sound is projected down its longest length
- Too much insulation can cause a room to sound very dead, use diffusers as well as acoustic treatment panels
- Bass traps can help reduce standing waves and bass build-up in the corners of the room
- Low frequency standing waves are usually caused by structural problems such as the room dimensions and can't easily be fixed with thin acoustic foam
- At least 20% of the surfaces in our room should be acoustically treated
- The floor & ceiling of a room can also be acoustically treated
- We can plot a basic frequency response graph using a test tone and our ears or a DB SPL meter

PSYCHOACOUSTICS EXPLAINED

In this section we will lay down the psychoacoustic laws and how we can use them to improve our productions, as well as how we can use this knowledge to treat our room, to provide a better listening environment.

WHAT ARE PSYCHOACOUSTICS

Psychoacoustics can be thought of as the way we hear and perceive sound within the brain. This is a very in depth subject; however, in this section we have collated all of the main points and expanded on them with the electronic music producer in mind, We will cover these main points:

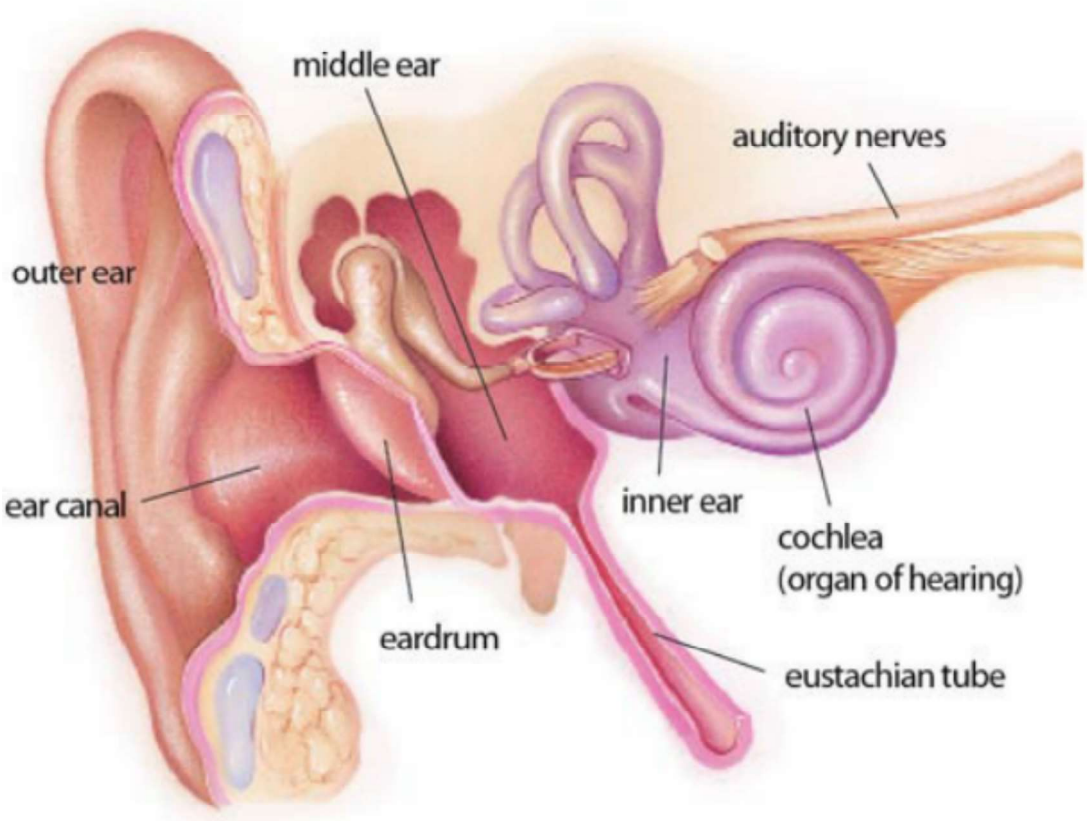
PERCEPTION IS NOT LINEAR
THE HAAS EFFECT (PRECEDENCE EFFECT)
MASKING
CRITICAL BANDS
NATURAL RESPONSE TO LOUD NOISES
EQUAL LOUDNESS CURVES
FLETCHER-MUNSON TO ALTER DEPTH
HEARING IN RMS
LOCALISATION
SOUND LAYERING
DISTANCE PERCEPTION

PERCEPTION IS NOT LINEAR

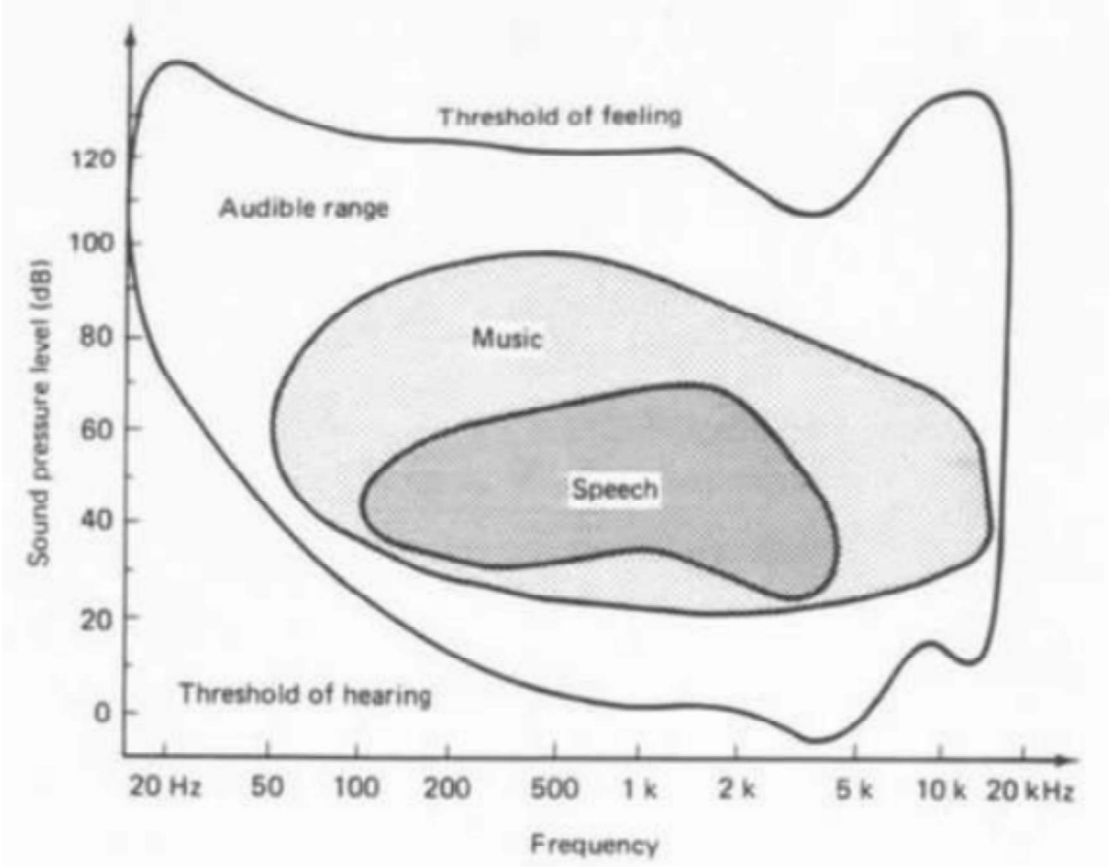
Our first point is that perception is logarithmic. This was discovered and researched by Gustav Fechner, who introduced the concept of psychoacoustics.

The way we perceive weight, light and most importantly, in our case, sound. Is non-linear. This can make it tricky to measure, which will be explained shortly.

Firstly we need to understand frequency ranges. The human ears can detect frequencies in the range of 20Hz - 20KHz. This is due to their inner and outer structure, as well as how our brain processes the neural signals sent from the cochlea.



Anatomy of the human ear



The frequency range of human hearing: Threshold of hearing at 0dB SPL to the threshold of pain at 120dB SPL.

The perceived loudness of sound at different frequencies is also non-linear. Our ears will need a very low frequency, of say a 100Hz sine wave, to be much louder for us to perceive it as the same volume as a sine wave at 1KHz.

DYNAMIC RANGE

The dynamic range of human hearing is defined by the threshold of human hearing, (The lower limit.) and the threshold of pain. (The upper limit.) The threshold of pain is relatively unbiased to frequency, and is capped at around 120dB SPL, which is roughly equivalent to being stood next to a jet as it takes off. The threshold of hearing however, is much more frequency dependant, which can be proved with equal loudness, or fletcher-Munson curves, which will be explained shortly.

The threshold of human hearing can be defined by a 1KHz sine wave measured at 20 micro-Pascal's, which is the equivalent of 0dB SPL.

Just to put this into perspective, the blood rushing through your veins is louder than this, and can be heard if you are in an anechoic chamber. (A chamber that has almost no reverberations.) Also known as a silent room.

Armed with this basic knowledge, we can now tackle some of the phenomenon of the psychoacoustic domain.



An anechoic chamber

THE HAAS EFFECT (THE PRECEDENCE EFFECT)

This is where two near identical sounds, (or an early reflection of a sound) that are no further than 35ms apart, and are within 10 dB of each other, will be interpreted as one sound.

A good way to test this theory is to stand 10m from a concrete wall and clap. You should be able to hear the reverberated signal separately as it bounces off the concrete wall and back to your ears.

This is because the reverberated sound is roughly 60ms or 20 metres (10m there & 10m back) apart from the original sound. (Sound travels at 343m per second (or 1125ft per second.) This approximates to around (1ms = 1ft) or (3ms = 1metre) If you keep clapping, whilst moving closer to the wall, try and stop when you can no longer tell the difference between the two sounds and they just become one unified sound.

You will be roughly 5m from the wall. This is due to being half the distance, the clap now returns to your ears in half the time, which is 30ms.

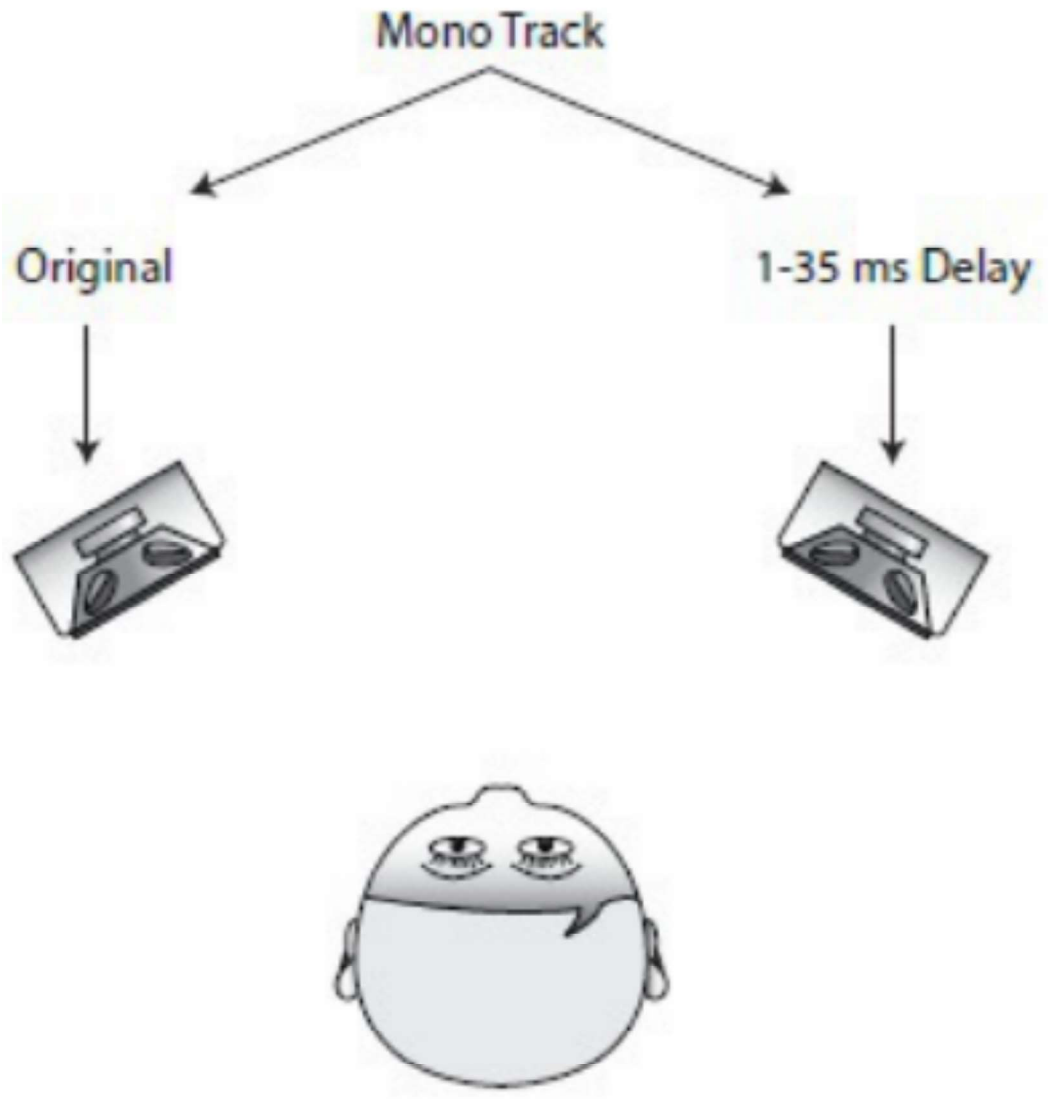
This is the time window where humans can no longer distinguish between the two separate sounds. (The outer time frame of the Haas effect.)

With this in mind, we can use this knowledge to enhance the stereo image of a sound. For example, we can take a clap sound and duplicate it. Pan one clap hard-left, and the other one hard to the right.

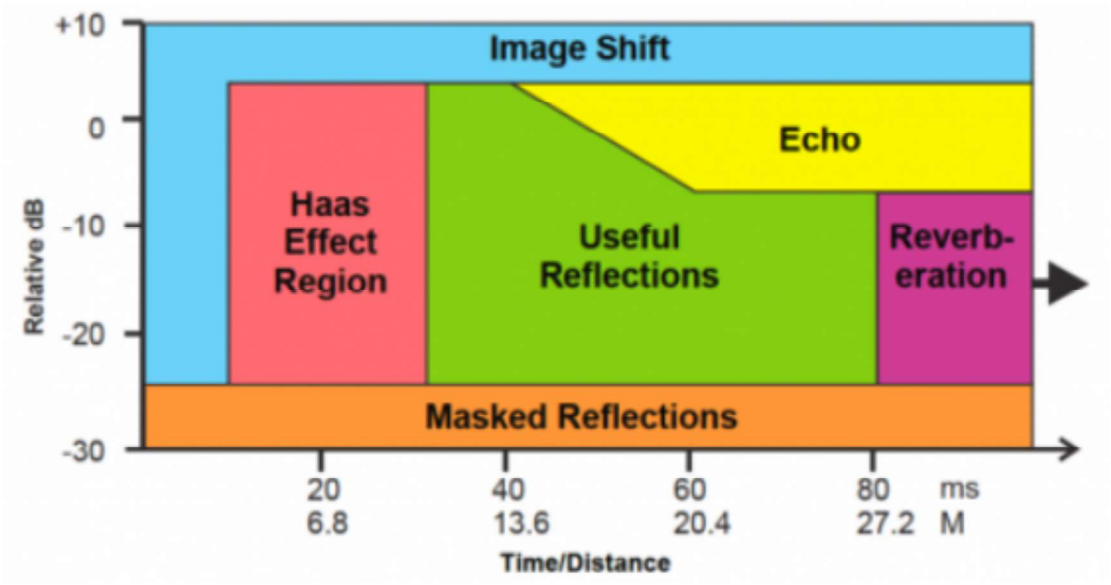
At the moment, all this will do is increase the loudness of the sound. However, if we now delay one of these signals by between 5-35ms, we will be changing the frequency information in each separate channel, making our sound very wide.

As long as we keep this below 35ms, our ears will still think that this is one sound. A caveat to this is that if we go above 5ms, we will start to hear the sounds separate. Going below 5ms will result in metallic and 'phasey' or 'whooshy' sounds.

The Haas window can change slightly, depending on if the type of sound is a transient, or a sustained note. But generally it will be anywhere between 0-40ms.



The Haas effect. A mono track is sent to both left ad right, but one channel is delayed up to 35ms.



A graph showing the Haas effect window in milliseconds, and meters. Within this window, two sounds will be perceived as being one.

MASKING

If two sounds with very similar frequency content play at the same time, and in the same stereo field within a mix, the louder sound will drown out, or 'mask' the quieter sound.

For example: If two people are having a conversation, we can hear them perfectly. However, if they are on a busy street, and a bus was to drive past, there's a high chance that we would struggle to make out their conversation.

This is due to their voices being masked by similar frequencies from the bus, and surroundings. So basically masking can be thought of as the 'drowning out' of sounds, due to louder sounds, (and their harmonics) in the same frequency ranges.

CRITICAL BANDS

This states that masking only occurs over a certain bandwidth, for example: if a narrow band of white noise at 2KHz is played over a 2KHz sine wave, we won't hear the sine wave.
As we alter the bandwidth, frequency, or amplitude of the sine wave, the masking effect will change.

A narrow band of white noise at say 8KHz will have little to no masking effect on a 2KHz sine wave, regardless of amplitude because these frequency ranges are too far apart. On the other hand, if we pushed the bandwidth of the white noise to maximum, meaning that it is the same amplitude throughout the entire frequency spectrum then we are not going to hear the sine wave over the white noise.

NATURAL EAR PROTECTION TO LOUD NOISES

The human ear naturally shuts down to protect the cochlea when it is subject to loud noises. We can think of this as a self-defence mechanism to protect our hearing.

We can replicate this by playing a loud noise, then quickly reducing the sound, before bringing it back up again. This is done throughout music production to make sounds seem louder than they are. From big EDM drops right down to on a transient level.

Some producers get up real close in waveform view and create small fades, or attenuations in volume in between the transient and tone of a sound to further accentuate and empower its impact. This is also done very well in film sound design, for sounds such as explosions and gunshots.

EQUAL LOUDNESS CURVES

As we mentioned in the introduction of this article, the ear perceives the loudness of a sound dependant on the frequency. This was discovered by Fletcher and Munson, who did a lot of research to produce the fletcher-Munson curves, (also commonly known as equal loudness curves.) their findings were that at high volumes, a sound will appear to have more high and low end.
Monitoring at low volumes is better for balancing, and gives a more true representation of the sound. (As well as being less damaging to the ears.)

Drawing from this knowledge that at loud levels the low and high frequency content sounds louder, we can also figure out that by Slightly Scooping out mid-frequency content, & boosting highs & lows when listening at low volumes, we can make a mix be perceived as louder and closer. When in reality all we have done is tricked the brain into thinking its listening at a louder level by reducing the mid frequencies slightly.

FLETCHER-MUNSON TO ALTER DEPTH

To make sounds appear further away, roll off the high frequencies. This simulates sound dissipation through air.

This is because our ears can sense depth perception by the ratio between mid frequency content and high & low frequency content.

As sound travels through air, the high frequencies are dampened or attenuated faster than the mid and low frequencies due to the size of their wavelength. (Which is inversely proportional to frequency).

EARS HEAR SOUND IN AN AVERAGE WAY (RMS)

A sustained sound at any given volume will be perceived as louder than a transient at any given volume. (Ears perceive loudness by RMS values.) This is why, when we use compression to raise the body of a sound we think it's louder or punchier than beforehand, even though the peak level is the same due to raising its RMS value.

Fundamentally, this rule governs the whole concept of why we use compression to reduce dynamic range, as well as why we use as much head room within a mix as possible to achieve the loudest perceived signal possible.
By combining the precedence effect and the way our ears detect loudness in an RMS type manner, we can beef up sounds using a reverbs early reflections. By adding these early reflections we will be boosting the RMS value of the body of the sound. And as we know, a higher RMS value equates to a louder (and fuller) perceived

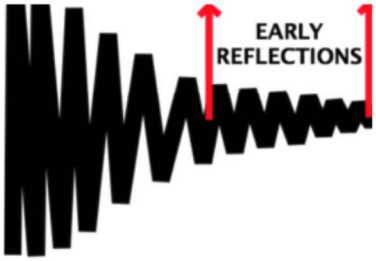
sound, due to making the sounds waveform more similar to a sustained note rather than a transient.

Provided our early reflections are <35ms from the original signal, we will hear this as one unified sound rather than an initial sound and its reflection.

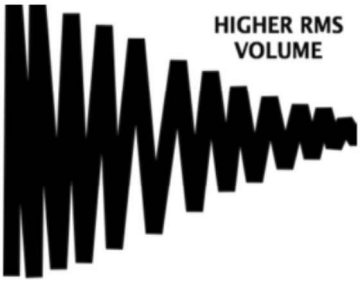
Making sure we don't include the late reflections is important, as this will prevent us from altering the depth perception of the sound.



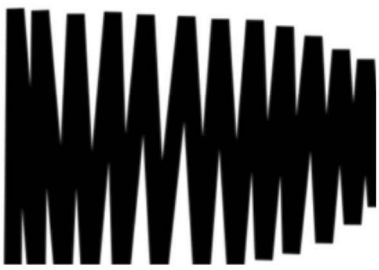
A transient with a high transient peak, whilst having a low RMS value.



Increasing the RMS value by increasing the gain of the early reflections only



The new waveform that is perceived as louder, due to its higher RMS value.



A sustained waveform, which will be perceived as being very loud due to having an RMS value, which will be very similar to its peak value.

LOCALISATION

Localisation is how the brain figures out the directional positioning of a given sound. This positioning of sounds can be tricky to get right within the confines of a DAW.

We can split the way our brain locates sounds into a few headings, which have been put into priority order.

Inter-aural **time** difference (ITD) - this is the time difference between the left and right ear, if we clap to the front, this will be fairly equal. However if we clap to the left, we know instantly that the sound is coming from the left, due to the sound reaching the right ear a fraction of a second later than that of the left ear.

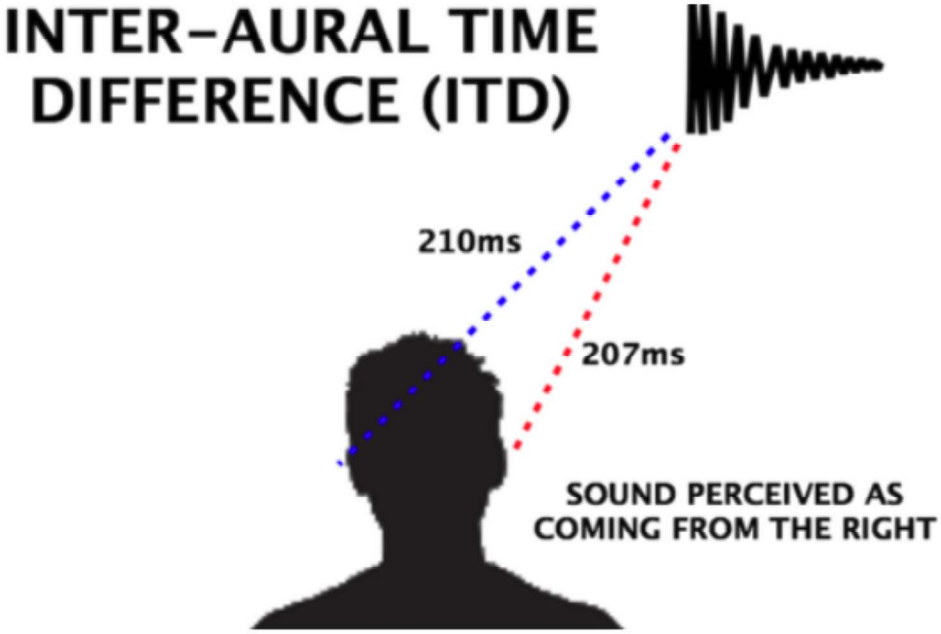
Inter-aural **amplitude** difference (IAD) - This is similar to 'ITD', however this is for amplitudes instead, and is most prevalent in the higher frequencies.

It's worth noting that 'IAD' will be overridden by 'ITD' if the two theories contradict each other. For example if we dropped a coin on the floor by your right ear, the sound would reach the right ear first, shortly followed by the left.

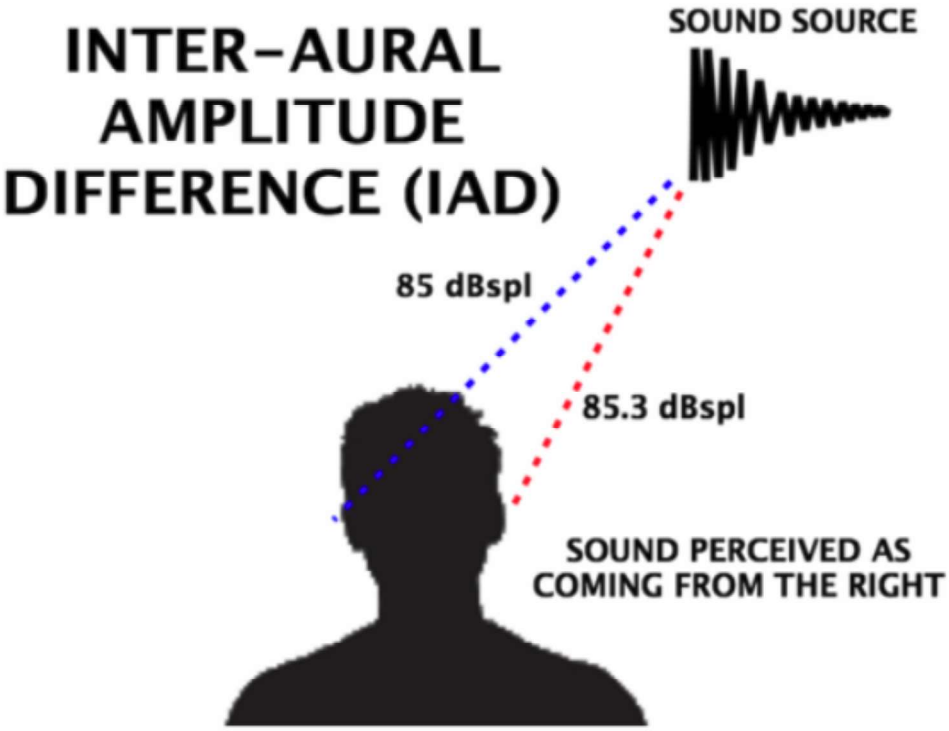
If for some reason the amplitude of the coin were louder in the left ear, our brain would still interpret the sound as coming from the right ear as the 'ITD' takes precedence over the 'IAD'.

Next we have the ears construction. Due to the shape of the ear, (the pinnae) it is designed to channel sounds into the ear canal, but the shape will muffle sounds approaching from the rear.

Because of this anatomy, our brains perceive sounds that are lacking high frequencies, to be coming from the rear. Also micro movements of the head, and differences in high frequency content in the left and right ears, help us to hone in on the sound source.



A diagram to show how the ears can make directional judgement from receiving a signal at slightly different times.



A diagram to show how the ears can judge direction by the difference in amplitude of a signal in each ear, as well as the difference in harmonic content.

LAYERING SOUNDS

With masking in mind, our ears struggle to differentiate between two sounds of similar frequency content, when played at the same time.

This means that when layering sounds, our ears don't hear two disparate sounds but simply hear one new blended sound instead. We can layer and blend different tones and textures together to create a new more complex sound

DEPTH PERCEPTION

Depth perception relies on reverberant patterns. (A reverb signals timing, in relation to surfaces around you.)

This is comparable to how bats use echo sounding to see. By listening to how their squeaks bounce back off their surroundings, they are able to map out the dimensions of an acoustic space.

Depth perception also heavily relies on the dry to wet reverb ratio. This is fairly easy to understand, the further away a sound is, the less detail and clarity can be heard, and the more the sound of the room or reverberate signal will be heard in relation to the original.

Our distance perception also relies on being able to detect the dampening of high frequencies as a sound travels though air.

As mentioned earlier Air absorbs high frequencies much more than mid and lower frequencies over large distances due to the size of their wavelength.

For us to know if a sound’s high frequencies have been rolled off, the Sound needs to be familiar to us in the first place.

A good example of this would be the human voice, Or a familiar instrument such as the piano. Our brains can then use the memory of this sound as a reference point.

PSYCHOACOUSTICS SUMMARY & KEY POINTS

- Psychoacoustics is the study of sound perception. Specifically the psychological & physiological responses associated with sound.
- Perception of light, weight and sound is non-linear, which can make it difficult to measure
- The threshold of human hearing is 0 dB SPL
- The threshold of pain is 120 dB SPL
- 0 dB SPL = 20 micro Pascal's
- The Haas effect can be used to make sounds appear wider (up to 35ms)
- Masking is when one sound is drowned out by another sound of similar frequency content
- The ear naturally shuts down to protect the cochlea when it is subject to loud noises
- Monitoring at low volumes is better for balancing
- Ears perceive sustained sounds as being louder than short sounds of the same amplitude (this is the basis for why compression is perceived as increased loudness)
- The brain uses localisation and differences in time and amplitude to place sounds within a stereo field
- Sounds with less high content are deemed as being further away than brighter sounds

MIXING TECHNOLOGY AND EQUIPMENT

Even for the bedroom producers, it is still useful to know how a professional studio is laid out, and the basic equipment in it, in case we ever end up mixing in one.

THE MIXING CONSOLE

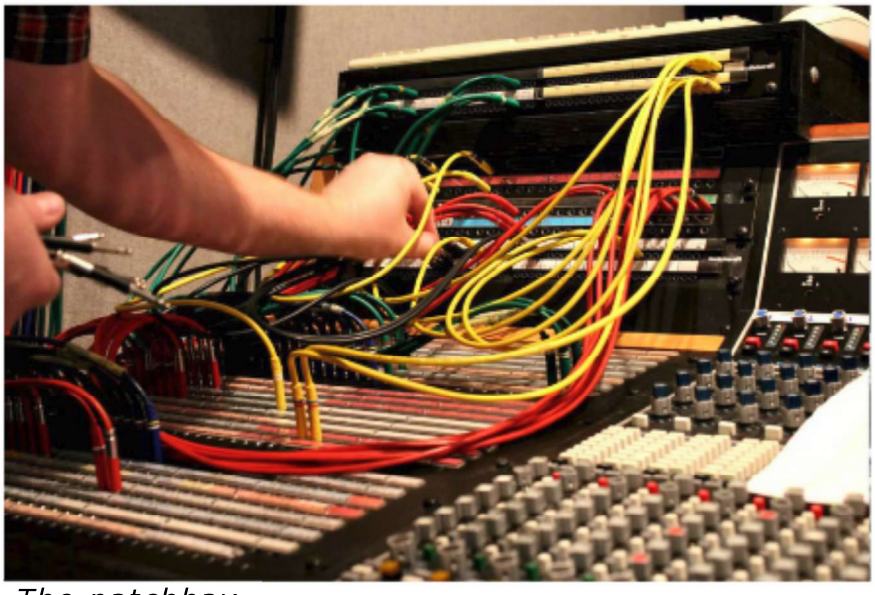
This is the hub of the studio, with everything connected to it. From here we will be performing most of our mixing duties. This is also where we will be summing our individual tracks down to one stereo audio file.

The mixing console can be used to send and return signals to different areas of the studio, such as effects rack and processors. We also have some controls built into the console such as EQ's, and sometimes compressors.

THE PATCH BAY

This is how the mixing console is connected to all the equipment in the studio. Think of this as all of the wiring in the studio.

Rather than having to go into the back of the console, all of the wiring will go straight to the patch bay. Usually the patch bay will have hard wirings to the inputs and outputs of the soundcard and the console, as well as having routings to all of the effects and processing racks. This means that we can effectively route any channel, to and from anywhere, using the patch bay and one or two patch cables.



The patchbay

OUTBOARD EQUIPMENT

Outboard equipment is external hardware, FX units or processors such as compressors, reverbs, and delays. These will commonly be mounted in racks and wired into our patch bay.

THE DAW (DIGITAL AUDIO WORKSTATION)

This is a program on a computer that we are going to use to host our session electronically and save the files, so we can come back and work on our mix whenever we want, without the need to manually recall fader and knob settings.

Popular DAW's are:

- Ableton Live
- Logic Pro
- Cubase
- Pro-tools

The DAW is linked to the soundcard, which will feed the information back and forth from the console and the speakers. We want to make sure our soundcard has enough inputs and outputs to be able to utilize all the channels on the desk. We may also want backup hard drives for the computer, and in some cases we may also use external DSP hosting hardware to take the CPU strain off of our computer when dealing with a large amount of tracks.

STUDIO MONITORS

We have already covered these in great detail, but these will usually comprise of two or three different sets of speakers, that will be wired into the console via a monitoring control box, which allows us to switch between set reference volumes, and different speakers.

DIGITAL VERSUS ANALOG

Many top mix engineers that used to mix using analog hardware are now mixing completely in the box (ITB). This is thanks to the advancements in digital technology and the plethora of very accurate analog emulation plugins. (VST's and AU's.)

Mixing in the box is most likely how many of us will be mixing, due to the space and financial limitations of having our own hardware and a studio.
When mixing in the box we recommend fully researching plugins before buying them. Only buy them if we could really benefit from them. This will force us to learn a plugin inside and out, and know exactly how to get the best possible results out of it. Too many times we have seen producers either download massive bundles, or manage to find a cracked version of software, only for it to sit on their list of VST's never to be used, because they also have hundreds of other plugins that they haven't taken the time to learn. Half of which are unlikely to work properly anyway because they are cracked.

Whilst many producers use cracked software, we highly recommend purchasing the licenses properly. This helps support the companies that develop and create the VSTs, as well as protect the producer/mix-engineer from any potential legal implications due to making money from productions whilst using cracked software. This also means we will stay up to date with any new versions and bug fixes.

Many older producers however, still prefer to mix in an analog style because much of the old equipment, such as the SSL desks give sounds a warmer colour due to the analog circuitry & analog summing. Digital signals tend to sound colder and less vibrant than analog.

MIXING TECHNOLOGY AND EQUIPMENT SUMMARY & KEY POINTS

- A patch bay is used to send signals around a studio, to and from different hardware or locations
- Outboard equipment is external hardware, effects units or processors
- DAW stands for digital audio workstation; Ableton live, Logic Pro, Pro-tools and Cubase are the main competitors
- VSTs and AUs are software instruments and FX; These can emulate outboard gear from within the software (in the box)
- Many producers prefer the characteristics that analogue hardware can impart on a sound
- It's better to build up a collection of legitimate VSTs slowly, rather than resorting to downloading cracked versions that don't work correctly

MIX PREPARATION

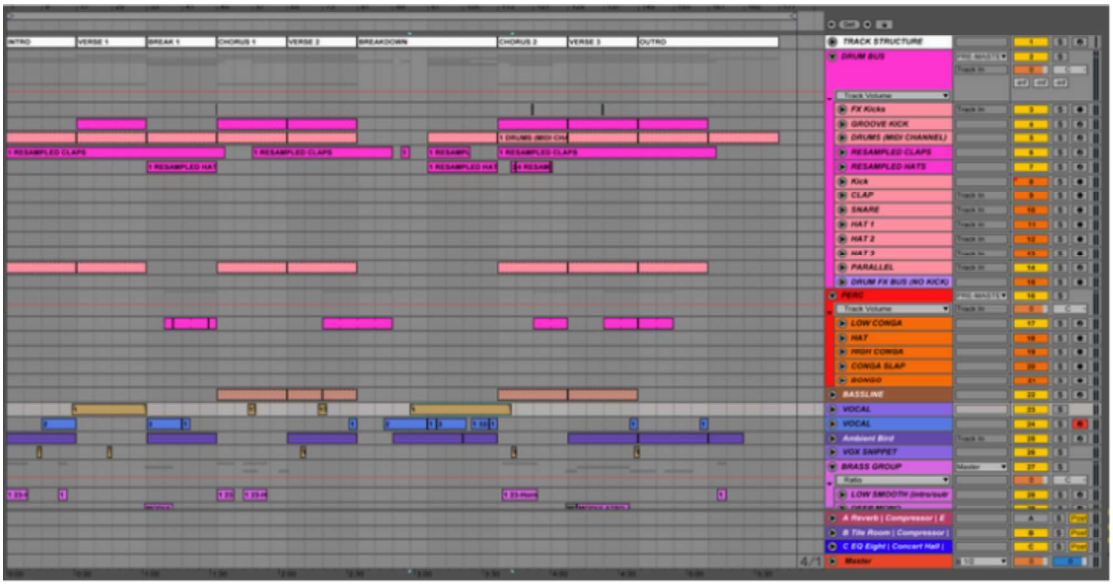
In this section we will take a look at how we can prepare the project, ready to start the mix-down process. Having a tidy mix allows us to focus on more important tasks, such as creativity, rather than trying to find a specific instrument or control.

A large portion of the mix-down is down to good preparation, the rest is inspiration, creativity and mixing skills. Preparation however, is not mixing. Think of it more as a preliminary organizational task, which is essential to mixing.

Mix preparation is where we draw the final line in the sand between the end of the production and compositional phase, and the start of the mixing phase. There should be no turning back at this point, unless we've made a monumental error that requires us to reload the entire project.

This phase gives us a chance to listen to the entire mix and check that there are no obvious mistakes that have been overlooked, as well as to get a grasp on the management, organization and navigation of our track.

The end result that we are after is the best possible mix-down that we can achieve. The best way to do this, is to start off with a mix that is well organized, and labelled. This will make navigation and workflow much easier, which means we will do a much better job of mixing the project.



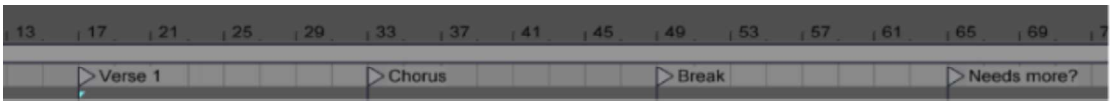
Colour coding, grouping and renaming tracks for ease of navigation

BITDEPTH & SAMPLE RATE

Firstly, make sure that when we get the files, that we are working in whatever bit-depth and sample rate the files are given to us in. typically, this will be 24Bit 44.1KHz or 48KHz, but may be higher or lower.

TRACK ARRANGEMENT MARKER MAPPING

Use the arrangement markers in arrangement view, to map out the different sections of our track, these can then be key-mapped, to allow for quick playback of different sections. These can then be placed at the intro, outro, verses, and chorus points of your track. (Bar 17, 33 etc.)



Using locators within the Ableton Live arrangement view

SIDECHAIN TRIGGER TRACKS

Set up a new track next to the arrangement-mapping track. Use a short sample or a midi sampler with a short sample loaded into it, which can easily be manipulated via an ADSR envelope.

This is going to be used as our side-chain trigger, so we don't have to mess around with our kick in the session.

We can now make a kick pattern that stems the entire length of the track arrangement, so that when we drop a track into the project, We can now use this as the side-chain trigger source. We can also go in and edit the side-chain midi notes if we wish to remove side-chaining for the break sections etc.

RENAMING & COLOURING

Try to get into the habit of renaming from the production and composition phase. All of our tracks should be clearly labelled, and so should all of the clips within our tracks.

We should be using a colour coding system that we can apply across all of our projects. We suggest:

- Brown for bass
- Orange for drums
- Red for synths
- Green for vocals
- Purple for strings & pads
- Blue for FX

it may help to use different colours for other elements, such as low and high-end sub-mixes. This also includes renaming any auxiliary's and busses, so we can see at a glance where each of them are routed, and what they are doing.

REMOVE LOW-END RUMBLE

This can be done in either the production, or preparation phase. It is good practice to ensure we have EQ's set up on all of our tracks that roll off any frequencies that shouldn't be there below around 20-40Hz.

Some producers prefer to get all of the mundane jobs such as this done right at the very start, before mixing. This allows them to make the most of their creative flow.

TRACK COUNT REDUCTION

We may have two mono claps, panned slightly away from each other to add width to a snare. Instead of having to group them together or process them separately, bounce the two claps to a single stereo file.

Try not to do this together with snare samples as well, as we may want to change the tonal characteristics, EQ, or dynamics, relative to the claps, which will be very hard to control if they are rendered together.

Another reason that we do this is because as soon as we start mixing, time is completely stacked against us. The longer we spend listening to a track, the more we will become emotionally connected to it and lose that objective way of thinking that helps us to stay focused and unbiased towards the mix.

GROUPING & SUBMIXING TRACKS

Group the different tracks into stem categories such as drums, percussion, bass, vocals, pads, synths, low-end, high-end, etc. This will give us greater control over automating groups of instruments, as well as making tweaks and adjustments.

Grouping and Bussing is also useful for when we want to apply effects to an entire group of sounds. An example of this is when using room reverb, or glue compression on a drum rack to make it gel and sound like it is coming from the same space.

DOUBLE CHECK TUNING AND TIMING OF PARTS

This goes without saying. At this point, everything production and composition wise should be spot on so we don't have to waste time trying to make further timing or tuning corrections later on.

Also at this point, set up any send/return busses that we may wish to use with the appropriate plugins applied. Keep the device activator switches turned off until needed, this will save CPU if we are working on a large project.

Turning these effects on, only when they are needed will keep from eating up our valuable processing power and slowing down the computer.

If we have received the track in the form of stems, then we'll usually find that we have a whole bunch of 1-clip tracks, each stemming from 1.1.1. Right through to the end of the track, with big blocks of silence in between.

In Logic there is a feature where we can quickly remove the silence. In Ableton however, we will have to divide the clips up and remove the silence manually. Ensure to add crossfades to the start and end of all our clips just in case there are any pops or clips that need to be removed.

BOUNCE TO AUDIO

Once our track is ready for mix-down. Make sure to commit to the mix-down phase by rendering our tracks down to audio.

This will prevent us from spending days going back in to make adjustments and tweaks that will just take us round in circles and waste precious time.

Bouncing to audio also means that our project will be reset to unity gain (0dbfs) on all of the faders. There will also be no processing power taken up by plugins from the composition phase. Further to this, we will be able to load the project file into a DAW mixing template.

DAW MIXING TEMPLATES

We can set up all of the below settings we can save our blank project as a mixing template:

- Buffer and latency settings
- Send & return channels
- Default audio and midi tracks
- Master channels
- Side-chain track
- Arrangement track
- Default settings for devices

This will save us from repeating the same processes every time we mix a track as well as saving us valuable time in the long run.

TRACK STRUCTURE CHANNEL

Create a blank midi channel and map out a typical arrangement of a track in a genre of our choice. We can even import a favourite reference track to copy its structure. This can then be mapped out using midi clips, which can be colour coded and renamed. We can then save this as a default set, so that every project we work in has a rough guide of sections at the top of the arrangement view.

When we import different tracks for mixing, we can then shuffle the midi clips around to fit our track. We recommend colour coding and renaming these sections such as: intro, verse, break, chorus etc.

SETTING UP A STEREO/MASTER BUS COMPRESSOR

A common technique used when mixing a track, is to balance parts straight into a mix bus compressor on the master output. This is one of the final processors, which is going to glue the elements of our tracks together. Many producers find that trying to put this compressor on right at the end of the mix process means that the whole track has to be re-balanced into the compressor, so instead, it helps to either keep the compressor on, with certain settings from

the very start, or find a rough mix balance, then put the compressor on and continue from there.

The compressor should be set up so it is barely affecting the signal. Settings may need to be tweaked later on, but good starting settings are:

- Ratio - between 1.5:1 and 2:1
- Threshold - allow for between 1 - 4dBs of gain reduction
- Attack - slow attack
- Release - slow release, auto release, or a setting that allows the compressor to reset between hits in a smooth manor that compliments the groove

There is a further book in this series dedicated entirely to compression, which will cover this subject in much more detail.

IMPORTING & FILE MANAGEMENT

We can either import our audio files using the browser or we can simply drag them into the DAW. We should make sure that we copy all of our files to a place on our hard disk, and ensure that the project, samples, and info folders are included, by pressing 'collect all and save'. This is especially crucial for DAW's such as Logic, which have a tendency to crash unexpectedly. Especially in its more recent updates.

Nothing is worse than being 5 hours into a mixing session, which is going really well, only to lose the whole project because of a power outage. Pro-tools has an auto-save function that can save and recall a snapshot of the project at certain time intervals, such as every 5 minutes.

Remember to save regular backups, especially when making big decisions. Alternatively use the 'save a copy' function if you wish to make alternative versions in case you need to flick back to a previous point in the mix process.

MACRO MANAGEMENT AND HANDLES

We can manipulate each of our sound sources, and set up quick controls to be able to sculpt and fit sounds from different channels together, into the mix quickly & efficiently.

In Ableton Live this is made very simple with the power of racks and macro controls.

We can set up these 'handles' for our sounds so we can control any number of properties of the sound, such as:

- Dynamics
- Tonal balance
- Filtering
- Volume
- Equalization
- Spatial placement within the stereo field
- Width
- Modulation

We can use these handles or multi-device racks, to solve common mixing problems that we may get asked, such as:

"Make the kick punchier"
"Make the vocal more upfront and present."
"Widen the drums"
"Put the pads in the background"



Using macro's to create custom handles for complete control of Ableton Live

REFERENCES AND A/B COMPARISON

If we are mixing for someone else, we should ask for some reference tracks with a similar vibe to the result they want from the mix-down. If we're mixing our own track, then we can also do the same.

A brilliant referencing plugin is sample magic's MAGIC A-B. This allows us to load, edit and loop a number of reference tracks, for quick A-Bing at the touch of a button.

The tracks we decide to use should sound great on a variety of platforms. Don't use low quality compressed mp3 files. We will need to use lossless audio for our reference tracks such as .Wav or .Aiff.

When referencing remember to pay particular attention to the comparison in dynamics, layering, width/narrowness of the mix, depth, muddiness of mid and low frequency content, overall EQ and tone. This will help give a different and comparable perspective to our track, relative to the reference, which (providing your reference is of high quality) will result in a much better final mix.

COMPUTERS & THEIR SETTINGS FOR MIXING

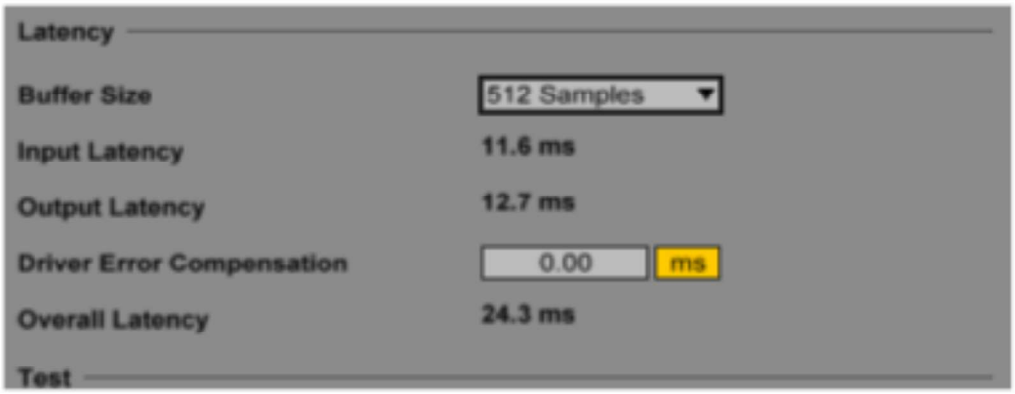
Here is a list of specifications that are nice to have if we are going to be doing a lot of mixing on our computer.

- i5 or i7 processors
- Large/dual display
- 2.7MHz++ clock speed
- Quad core++
- SSD or FUSION drives

Fusion combines a small SSD (solid state drive) with a standard disk hard drive and allocates the space, dependent on the most frequently used items, which can be edited in the settings.

BUFFER SIZES & LATENCY

Buffer sizes within the DAW can be set relatively high because latency isn't really an issue at the mixing stage: The larger the buffer size - the bigger the latency, a larger latency means less load on the CPU. It isn't having to process all the plugins and effects as often, on a sample-by-sample basis, but rather it processes the data in chunks of every 256, 512, or 1024 samples.



Latency & buffer size settings in Ableton's preferences

FREEZING AND FLATTENING

This technique is mainly used during the actual mix process, but is worth mentioning early, so we can apply it as and when is necessary in our mix. Computers with low CPU processing or projects containing lots of tracks and third party VST and AU plugins can really struggle to keep up with the processing workload, This can be resolved by freezing certain tracks in the arrangement.

Freezing a track will render that tracks audio, so that the processing is already applied. (Such as compression and any effects plugins) This will effectively lock the track and plugins from any further editing, but will stop the track from taking up real-time CPU processing power.

The track can be 'unfrozen' at any point to allow us to change any parameters.

When tracks are frozen they can then be 'flattened' to audio, which will remove all plugins and bounce the track in place to an audio file, This can also be achieved by simply dragging and dropping the frozen tracks audio into a new audio track.

MIX PREPARATION SUMMARY & KEY POINTS

- A tracks compositional decisions should not be altered after the mix preparation phase
- Standard audio CD quality is 44.1KHz, 16bit
- Use track arrangement markers to move quickly between sections of a project
- Set a separate side-chain trigger track which can be set in a default template
- Name and colour all files correctly
- Remove low-end rumble from recordings and samples (20-40Hz)
- Reduce the amount of tracks within a project (backing vocals in particular)
- Group similar tracks together and set up sub-mix busses
- Double check the timing and tuning of all parts
- Bouncing the track to audio will stop any further tweaking and also allow faders to be set to unity gain when re-imported (unity gain = 0dB
- on the fader)
- Build a DAW mixing template with standard go-to settings for the project, tracks and devices
- Make a midi channel at the top of the DAW, which can be used to plot the structure of your track
- If required, set up a master bus compressor
- Set up 'macros' and 'handles' which can be midi mapped for complete control of the project
- Find a good selection of reference tracks in the same genre you wish to produce

- Make sure your reference tracks output is set to stereo out and that they are level matched with each other and your project

- small buffer sizes are used for recording, larger buffer sizes can be used later on in the production process for mixing
- A larger buffer size introduces more latency

- a larger buffer size will also reduce the CPU load on the computer

- To further reduce CPU, freeze and flatten any tracks containing labour intensive plugins

MIX PREP: TRACK ANALYSIS

OBJECTIVITY & SEPARATION OF MIX PHASES

The first time we listen to a track will be the most objective time we will ever hear it. This is when we are going to make subconscious decisions, (hopefully) without any bias towards the track.

It helps if we haven't had any input into the production or composition of the track. By being too involved with a track pre mix-down, we will have heard the track so many times that we will either, already be frustrated with the track, or will be too attached to certain aspects or techniques that we employed in the composition of it.

A combination of all of these different reasons will equate to us no longer having an objective ear, like that of a label A&R or a consumer. Likewise the modern producer has had to adapt to being able to multi-task and fill a number of roles in the studio.

To that end its important that we treat the production and mix-down as two completely separate phases. Once we enter the mix-down phase, we shouldn't be revisiting the production phase. All the tracks that are needed should be there, ready to be mixed.

No further tracks should be added, and likewise we shouldn't be making any further adjustments. Adding more parts at this stage will lead into a spiralling circle of time wastage & tweaking, only to increase the onset of ear fatigue and frustration with the track.

Another reason why the first listen is so important, is because this is one of the only opportunities we're going to get to fully immerse ourselves in the perspective of the listener.

We have absolutely no idea how consumers will hear our track, it could be on anything from a set of headphones, to walking past a kitchen window with music on inside. Because there are so many variables with how consumers will hear our tracks, its important that we really put that first listen to good use with a fresh mind.

Due to the reasons above, we need to make an educated decision whether we should try mixing our own track, or whether its for the greater good that we pass our track on to a professional mix engineer to achieve a better result.

Whilst most mix engineers will very easily be able to see which direction we want to go with the track, and how we want it mixed down, every mix engineer works differently. Look into some different mix engineers, and listen to what artists producing similar music are saying.

There are no issues with using a professional mix engineer, they are paid to do what they do best, and usually have many years of experience. Second to this they are also only focusing on one specific task, which is mixing our creation down into one stereo file, whereas taking the task on ourselves is just another task out of many, which may not get the full time and attention it deserves.

Most mix engineers are going to have access to professional studio equipment, which can be utilized to get better audio quality, whether that's loudness, or a more crisp/warmer sound. Finally our last point, the mix engineer has never heard our track before.

We will assume that we are going to take the bull by the horns and attempt to mix our own track down, but we will firstly mention mixing other peoples tracks.

MIXING SOMEONE ELSES TRACK

If we are mixing someone else's track, then the producer will usually send us a DIY mix-down that they have done, to help us better understand the direction in which they wish to go with the track, as well as some notes on what they want us to do. (Or not to do as the case may be.)

We are going to use this DIY mix-down for the first listen. The first listen shouldn't be too analytical. The objective here is simply to get a feel for the track and see what subconscious emotions it invokes.

Have a listen to the way it has been mixed so far, and how the instruments gel together as a whole. At this point we may notice things that need to be rectified straight away, try not to hone in on them too much for now. Just listen to the track and make a note of any gut instincts.

The second time we hear this track, we should be looking at it from an analytical perspective. At this point we should be in the studio with a notepad, ready to write down bullet points as we listen through the track.

Things to take note of:

- Timing
- Pitch
- Arrangement
- Dynamics
- Textures
- Relative levels of specific parts
- Tonal balance
- Panning
- Stereo field & Width
- Separation of parts occupying similar frequency ranges

We should do this process a few times over, switching between different speaker setups, such as our nearfields, the radio style speaker and also some headphones.

This not only allows us to analyse the mix from a few a consumers perspective, but it will also reveals flaws with the mix-down, and how it translates to different sound systems.
Remember that the mix-down wants to create a nice balance between parts, and can sound very different dependent on the speaker setup.

MIXING OUR OWN TRACK

Mixing our own track is going to prove a slightly more laborious task. First and foremost, we need to treat the production and mix phases separately.

When we are mixing our track we need to remember that we are no longer the producer that composed the track, and therefore sometimes we need to remove effects and plugins and take a few steps back, so we can then rebuild the desired vibe from the ground up, and take that leap forward to a better quality track.

We no longer have the advantage of the first listen of our track, so we need to take steps to try and give ourselves a second opportunity of the first listen. Unfortunately the only real factor that will help us in this case is time. Once we have finished the production of the track, put it away in a folder, ready for mixing and try to work on another project. (this is why we recommend working on a few projects simultaneously)

Over a few weeks we can distance ourselves from the track, enough that we can make better judgment calls when we decide its time for the mix-down. This is evident on a smaller scale when we are producing late into the night and get some sleep, the next day we can make quick decisions on the production, that our fatigued brain couldn't make the night before.

MIX CHECKLIST

We can quickly work through this mix checklist to ensure any major mix preparation tasks have been carried out:

- All production decisions should now have been finalized
- DAW blank mixing template loaded
- No further instrumentation or FX should be added to the mix
- All tracks should be printed to audio files to free up CPU
- Ensure all timing and tuning is correct, especially vocals
- Reduce the amount of tracks by bouncing them to stems
- Load reference tracks into DAW
- Set up mix bus compressor if used
- Track structure channel loaded
- Set up locator markers
- Set up any macros/smart controls
- Tracks grouped
- Set up side-chain trigger
- Check that the correct bit-depth & sample rate is set
- Colour code tracks
- Rename tracks

TRACK ANALYSIS SUMMARY & KEY POINTS

- The first time we hear a track is when we will hear it most objectively
- The composition, mix and master phases should all be kept separate
- Being too involved with a track can lead to us making biased mix decisions
- We must keep in mind how different listeners will hear our track
- We should be analyzing the timing, pitch, arrangement, dynamics, texture, balance, panning, width and separation of parts when we are listening to the mix for a second time
- Switch between different speakers when analyzing a mix
- Distance ourselves from the track, by working on other tracks simultaneously or by leaving the track alone for a few weeks prior to mix-down
- Before we start the track, ensure that the mix-prep checklist has been completed

GAIN STAGING

WHAT IS GAIN STAGING?

Gain staging is the process we use to solve a common problem, 'going into the red' or digital clipping.

Digital clipping is a cardinal sin within the world of DAWs and software. A computer is not able to accurately produce parts of a sound wave that exceed 0dBFS (full scale).

Back when we used big analog mixing desks, we would measure the level in dBu. 0dBu, being a solid reference level (0.775v), +4dBu is the level used to give the best signal to noise ratio attainable. (1.228 volts) Pushing past this level can cause distortion. Sometimes these driven & distorted signals are pleasing to the ear, this however is not the case. Software signal flow should never exceed 0dBFS.

0dBFS is effectively the maximum limit of loudness attainable from the software. There's nowhere to go beyond this point, so it's important that we stay well below it in level throughout our entire track. The amount by which we stay below this 0dbfs level is known as 'headroom'.

We can think of headroom, as building up cumulatively throughout a track. So we could start with a drum that peaks at -15dB. Once we mix in the rest of the drum group, this may now be at -7dB. Then once we add in the rest of the instruments, they will sum together and the next thing we know we are completely out of headroom and the master is clipping at +5dBs. To prevent this, we need to employ gain staging, which is where we monitor and control the gain at all phases throughout the signal flow and mix-down process.

The reason we need headroom is to stop volume fluctuations in the mix causing the signal to go above 0dBFs. The better balanced our mix is, the less fluctuations we will have. Having fewer fluctuations, means we can push our headroom closer to 0dBFs without clipping, which in turn means we can achieve a louder mix.

Many mastering engineers will expect a mix to have between -6 & -3dBs of headroom for them to be able to work with. Applying a DIY limiter to the mix process is fine, but don't leave this on, as this will use up headroom, as well as ruin remaining dynamic range that the mastering engineer has to work with.

GUIDE TO GAIN STAGING IN OUR MIX

It is important to use gain staging in order to preserve maximum fidelity on the faders for fine adjustments. A fader is most accurate around 0dB, the further down we pull the fader, the less accuracy we will have for adjustments.

RECORDING

Although this first point isn't part of the mix-down process, it's important to track our vocals at the correct gain going into the channel when recording.

With a mixing desk, this will usually be between -6dB and 0dB. When we put this into a DAW, this RMS value should be between -12 and -16dBFs.

We shouldn't be seeing any peaks higher than -6dBs. If a signal has been recorded 'too hot' there is not a great deal that can be done to rectify it, other than by adjusting the input gain and trying again. Adjusting the channel fader in a DAW will not alter the input gain. It will only alter the playback volume, which will not rectify the issue.

INDIVIDUAL TRACK GAINS

A standard error with mixing is to have all of the individual channel volumes too high. This is easily rectified by setting up our default channel presets at a lower volume, which should help us get into the habit of keeping each track with approximately 12 – 18dBs of headroom. A problem with this however, is that we now won't have unity gain set on all of our faders. (Faders set to zero.) Another way of resolving this issue is to use a utility plugin as an insert, and pull the gain down on this.

By doing this we are keeping the channel fader free for adjustments later on, and also reducing the gain at the start of the signal chain so that we aren't running a hot signal into any plugins that we may decide to use. This also means that individual tracks will sum together to give more head room in the mix busses.

MIX BUSSES

Our strategy is to move from the source, through the signal path, making sure that our gains are set with plenty of headroom at every step of the way. This ensures that we are slowly increasing the gain throughout the process in a controlled manner, rather than just boosting it all at one point and creating a clipped signal later on in the chain.
Here at the mix-bus, we may be summing a number of different parts. Problems often arise with drum busses, where we may have a number of very strong transients sounding at the same time. The sum of these transients will usually cause peaks and clipping, typically on the 2nd and 4th beat of a bar in house music, as this is when the clap, snare, and kick are likely to hit at the same time.

At the mix bus input stage we should still have a decent amount of headroom, by this point in the signal chain our mix should be well balanced and we shouldn't see any crazy fluctuations or peaks on the bus channel.

This is also the place where we would apply bus compression to tame any of these stronger transients, as well as using EQ to make tonal adjustments and notch out any unwanted frequencies, which will aid in creating more headroom. A good figure to work to when leaving the output of the mix bus, is around -3dBs to -6dBs, dependent on our type of audio. And what we intend to do with the mix-down.

At this point its not uncommon for a number of busses to then be summed together yet again to a further summing or pre-master bus. This is where any final adjustments and gain staging can be made. This also means that we have the opportunity to treat all of our separate mix busses separately whilst still giving them one final summing bus treatment to give the mix a cohesive sound.

The output of the Pre-master can once again be around -6 to -3dBs. Many mastering engineers will be grateful for any amount of headroom whatsoever.

Finally we have the master bus. Here is where we can apply our DIY mastering chain, typically consisting of:

- EQ
- Compression
- Stereo treatment/widening
- Limiting

At this point we can use maximizers and limiters to bolster the gain.

This could be anywhere from -6dB of headroom, right up to as loud as -0.01dbs. A fairly standard level is -0.3dBs for .Wav files or -0.5dB if the track is intended for mp3.

We will cover limiting in a later book in this mix series, however just as a quick tip, we recommend not brick wall limiting. (Smashing the input gain into a limiter to get an increase in volume at the expense of dynamic range)

Applying more than around 4dB of gain reduction on a final limiter is usually a sign of a poor mix-down, which can easily be rectified by going through the mix and checking the balance, gain staging and peaks of all the parts at each of the stages listed above and fine-tuning.

We can find the instruments which are causing the limiter to kick in and treat these elements at source as we feel appropriate, then we will find that when we go back to the limiter, we should now only need to push the limiter by 1 or 2 dBs in order to increase the RMS level of our track to a decent perceived loudness.

If we do this correctly, we won't be squashing as much dynamic range out of our track at the final stage. Remember to remove this DIY mastering chain if we are sending the mix to a professional mastering engineer.

Do not confuse gain reduction with input gain on a limiter. Due to the use of software, some engineers may allow plenty of headroom within the constituent parts of their mix. If the mixing has been done well then there shouldn't be too much fluctuation in level caused by rogue transients. This will mean that the mix engineer may have to use a lot of input gain before any gain reduction starts to show on the limiter. This is absolutely fine, and likewise only using a small amount of input gain on the limiter would also be fine if that is it all that is required to see a small amount of gain reduction. The important thing to be monitoring here is the consistency and amount of gain reduction, not necessarily the input gain.

GAIN STAGING SUMMARY & KEY POINTS

- Gain staging is the process of ensuring the levels are set correctly at all stages of the signal flow within our mix
- Levels should not exceed 0 dBFS within our DAW In analogue circuitry, pushing input signals above 0dB can add nice colour and compression to a sound As tracks are mixed, they will cumulatively sum together to use up headroom
- Headroom is defined by the dynamic range
- Bit-depth is directly related to dynamic range
- Mastering engineers expect between -6 and -3dB of headroom to work with at the mastering stage
- We can use compressors to stop signals clipping at the recording stage
- Recording into a mixing desk should be done at the optimum signal to noise ratio, usually between 0 to -6 dBu
- Recording into DAW's should be done at an RMS value of approximately -12 to -16 dBFS (This equates to +4dBu or 0VU, however this should not be taken for gospel because the dynamic range and RMS of a vocal is likely to be very different compared to that of an overdriven bass guitar)
- Use utility plugins to alter gain instead of using automation on channel faders
- At various stages in the mix process we may wish to set our faders to unity gain
- Signals should not only be below 0dBFS on a track by track basis, but also on the inputs and outputs of each device in a chain as well
- When a track is mastered, its normal for it to have between -0.01 to -0.5dB of headroom for printing to CD or .mp3

ABLETON'S MIXER ANATOMY ROUTING AND SIGNAL FLOW

Ableton Lives mixer, along with any other DAW's mixer has been designed to emulate conventional hardware mixing consoles used in professional recording studios. Although hardware is still widely used, it has come to the point where software audio workstations are actually capable of a whole lot more than most hardware mixing desks. We are no longer confined to the amount of channels on our desk, or the amount of money we have to spend on hardware. With software such as Ableton, we are now able to have as many tracks as our CPU will allow. The routing capabilities throughout these tracks are virtually limitless.

With all these routing options and mix functions at our disposal, it can very quickly get messy and frustrating for the new music producer. So in this section we are going to explain the mixer, its functions, and its routing one by one. Which should allow us to easily navigate our way around Ableton's mixer.

After we have some more experience with mixing we will notice that the fundamentals are the same across all platforms, from Logic & Ableton, right the way through to a proper mixing desk. Once we understand the fundamentals, it simply becomes a case of getting familiar with the different ways of doing things in each respective program.

WHAT IS SIGNAL FLOW

Signal flow is how the sounds are routed through our Mixing desk or DAW. We can re-route different channels in, and out of one another, as well as sending them to groups, send/return effects and mix busses and the master output.

On a conventional desk, most of the routing is done through a patch bay, which is designed to stop us having to go into the back of the mixing desk to constantly plug in equipment. Instead we already have all of the hardware wired into the patch bay, and we can simply grab some patch cables, and send signals to and from the console.

DAW's work exactly the same as this; however, there is no need for a patch bay, as there are no physical cables. Instead we just use software plugins that are routed to one another, using Input and output dropdown menus.

When we look at the input/output section of Ableton, we can understand the basic routing of most channels.

INPUTS

This is where we can route any form of audio into Ableton. For example, we may want to monitor:

- Microphones
- DI's (Direct inputs from guitars etc.)
- Synthesizers
- External sources from our soundcard
- Fold-back / Cue from a live room

If we select our soundcard, we will then get a second dropdown menu. This will allow us to choose between different stereo or mono channels on our soundcard.

We can change the input routing of tracks; this is handy for if we would like one track to feed into another one. We also have a sub menu within this routing matrix to enable us to route individual sounds within a track. This is important for certain instruments such as Impulse, which may require you to make extra auxiliary channels to separate the individual sounds.

MONITOR MODES

Monitor modes let us choose about how we monitor the INPUT signal.

IN - The input signal will constantly be monitored

AUTO - The input signal will only be monitored if record/arm is enabled

OFF - Will turn off any input monitoring for the selected track

INSERTS

Inserts come first in the mixer chain, the audio passes through the insert on to the next part of the mixer. (Which run from left to right along the bar on the bottom of the screen.)

We get a graphical representation of the levels of the inputs and outputs of each device, which can come in handy for a quick indicator of our gain staging. We can also use grouped FX chains to perform parallel processing from the insert strip; we will go into this subject in more detail in our compression eBook of this mix series.

On a standard mixing console there are usually Insert points on the console or into the patch bay, this allows us to break the signal flow and route it into hardware such as a compressor. The output of the compressor can then go back into the patch bay, so that the signal can continue on to the mixer. Once this signal has gone through any Insert FX, it then moves on to the main mixer functions.

SEND KNOBS

Here we can take a copy of the signal and route it to a bus. In Ableton these busses are automatically routed to the relevant return channel that is assigned to the send.

PAN POT

The pan potentiometer allows us to pan the signal anywhere in the stereo field.

MUTE BUTTON

The mute function allows us to mute the signal or multiple signals

SOLO BUTTON

The solo function will mute all other tracks so only this particular track can be heard. We can also solo multiple tracks using the CMD function.

RECORD BUTTON

Record/arm can be used so that we can record audio into a clip slot. This button is important if we are using AUTO as an input-monitoring mode.

VOLUME FADER

The size of the fader can be adjusted by dragging the mixer up and down. Having this at larger sizes is useful for accuracy when making small incremental changes during mixing. We can be even more accurate by holding down the CMD function when adjusting parameters to fine-tune our volume settings.

METERING

We also get peak (dark green) and RMS (light Green) metering, as well as logarithmic numbering down the sides of the meter. When the meter is expanded, we also get a peak level and a second box, which allows us to type in numerical values for the output fader volume.

When a signal peaks, this peak level will stay in the peak box display till we click on the box, which will reset the meter.

METERING AND DYNAMIC RANGE TERMINOLOGY

Within metering, there are a few key words that we will need to fully understand.

Within a DAW we use dBFS or 'decibels full scale'. Which works **backwards** from 0dBFs.

The **noise floor** is the quietest signal that we can hear within our DAW. This is where we will usually find ground hum and tape hiss. Within a DAW this will be right down at –Inf dBFS.
Next is the **nominal level,** this is the optimal level for recording instruments into your console or DAW. On hardware and mixing consoles this is at +4dBu, which is also equivalent to 0 VU. (Analog circuitry is designed to have the best signal to noise ratio at this optimal level) however, within your DAW this is slightly different. The optimal level for recording into a DAW is between -12 and -16 dBFS although the signal to noise ratio within a DAW is not really an issue, as long as we do not clip (go over 0dBFs) then there won't be a problem.

The difference between the **noise floor** and the **nominal level** is known as the **signal to noise ratio**. Like we mentioned earlier, this is very prevalent in analog circuitry.

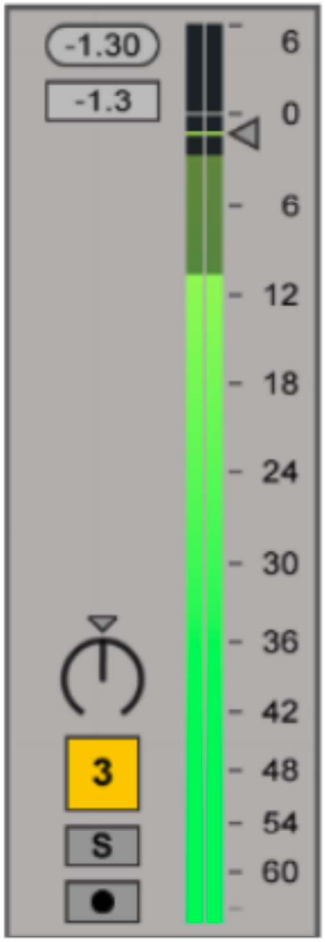
Finally we have the **maximum level**. Due to the circuitry in analog equipment, this can often be sent in hot. (Pushed hard) Into the red to give a pleasing saturated effect, as the signal is soft clipped.

In the digital realm of DAW's this is definitely not a good idea. 0dBFs is the absolute maximum your signal should ever reach within any part of your signal chain. Even occasional peaks should not exceed this limit, as digital equipment cannot accurately reproduce the waveform above this level, which will result in nasty distortion.

The difference between the **nominal level** and the **maximum level** is known as **headroom.** This is what mastering and mix engineers refer to when they say "leave some headroom in the mix for the mastering engineer to do his work."

This basically equates to not hard compressing or limiting a waveform, and leaving them ample space to work with for the mastering stage.

The difference between the **noise floor** and the **maximum level** is known as the **total dynamic range**.



As the track develops and our quietest signals start to become louder we will be decreasing our **dynamic range** in order to achieve a overall louder **RMS volume.** (Average or root mean square volume)

PEAK METERING ANALYSIS

Peak meters are used to tell us the maximum level of a signal. This doesn't really give us a very good indicator of how loud a signal is, but it is very useful for knowing if we are overloading our signal at any point in the chain which could cause clipping.

RMS METERING ANALYSIS

RMS or root-mean-square meters, use a mathematical function to work out the average loudness of a signal. This is a much better representation of the overall signal level, but cannot be relied on to see if we are clipping a signal.
The RMS volume will never be more than the

peak volume. It can only be less, or in extremely rare circumstances it can be equal to the peak volume. (In the case of a completely sustained note with no decay or release envelope.)

RMS metering is the preferred type of meter when checking the overall level of a complete mix. This is also the type of metering used for VU meters because it closely replicates how the human ear perceives sound.

OUTPUTS

We can also route the output of audio to certain locations, such as groups, master outputs, soundcard outputs and auxiliary tracks. With all of these different output routings, we will often get a sub menu allowing US to choose at which point in the chain we would like to send the audio.

Common options are whether we would like to send the signal Pre, or post mixer, as well as Pre, or post FX. This is very useful if we have certain effects that we may wish to bypass when sending to another channel.

ROUTING, BUSSES & GROUPS

MIX/SUMMING BUSSES & AUXILIARIES

We can also make a sub mix, or auxiliary channel by creating an audio track and routing multiple tracks to its input. This can then act as a group, and gives us control over volume, panning, solo and mute functions.

When we use an audio track for this, we should set our monitoring to IN. This will ensure that we always listen to the track regardless of if the track is armed or not. In auto mode, we must 'record enable' the track to hear it's input.

BUSSES

This may be slightly confusing, but it's important to understand that a bus is often used in the wrong context. It isn't actually a destination for audio, it's actually the transport of how the signal gets there.

A good way of thinking of a mix bus is to think of it as a cable, or a flow of signal. This signal can be sent to wherever you want, by selecting a routing option in Ableton, or in hardware by using a patch cable. For example: we could send a backing vocal track down a mix bus to a new auxiliary track. We may also be sending other backing vocals to this same new auxiliary track, which some people may refer to as a mix bus or group.

GROUPS

A more simple way of creating a mix bus is by grouping tracks. This is easily done by selecting multiple tracks, then selecting 'group' from the drop-down menu.

These tracks will now be grouped together, and create a main group track, which has the function of folding all of the individual tracks away. This can be useful for making changes in volume to whole groups of sounds such as drums, without ruining the balance between these separate sounds.

Also notice that in the output routing of the individual tracks the output has now changed to GROUP, and from the group it is then

passed on to the master output. This is ideal for keeping your session organized.

We can use groups and mix busses to process signals to glue them together as a whole, as well as control all of the tracks parameters at once. This also aids us later on in the latter phase of the mix-down, so that we are dealing with fewer stems. It's quite common to be dealing with 5 or 6 groups as opposed to 40+ individual tracks towards the final stages of a mix.

Here are some common sub-mixes that would be blended together in the final stage of your mix-down:

- BASS
- DRUM BUS
- MAIN VOCALS
- BACKING VOCALS
- LEAD SYNTHS
- PADS & ATMOSPHERICS
- FX
- LOW-END
- TOPLINE

Anyone who has remixed a track, or is familiar with STEM files will recognize that this is usually the stage that producers will bounce their parts down for remixers. These main elements or sub-mixes are commonly known as stems.

SEND & RETURNS

Finally we have Send & Return Tracks. We can use the send knobs on each track to effectively send a duplicated version of the signal down a pre-routed bus to the relevant return track.

From the return track we can then apply any FX processing we may wish to add. From here, we can also feed this return into another return track or back into itself to create a feedback loop. Be aware of excessive feedback raising the volume and damaging your ears and speakers, this can be prevented with a limiter.

We also have the option to have a pre, or a post-fade send. Usually we will have this set up as a post-fade send, which means that the signal is sent to the return track after the fader. This ensures that the return channels volume stays relative to the track volume, which is perfect when mixing things like reverb, as it ensures we don't mess up our dry/wet balance when we adjust the volume of our track.

If we wish to have more control over the return level, then we can set it to pre-fader, which now means that it will work independently of the track fader. This means that even if the track fader were set to –Inf dBFS, we would still get the full volume through the return track.

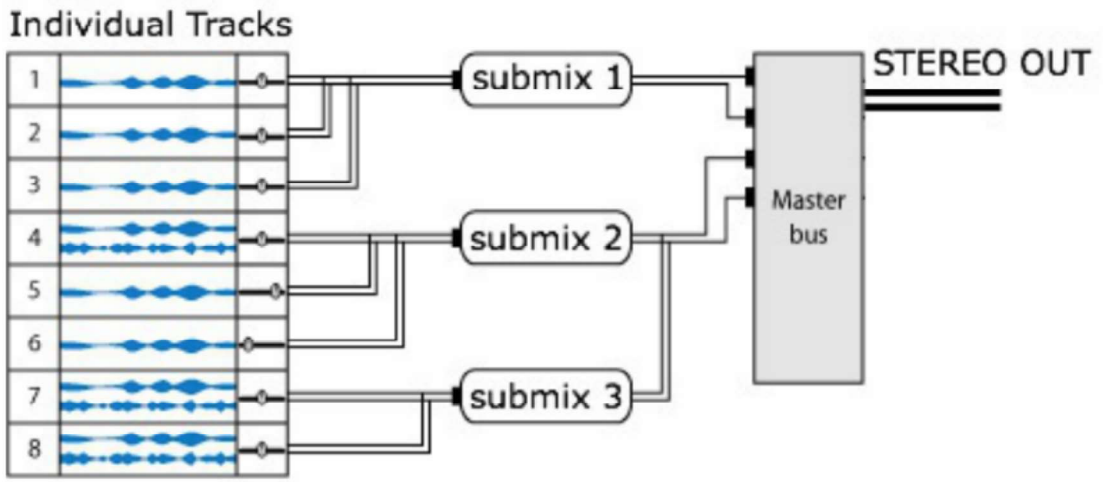
STEREO OUTPUT

The stereo output, also known as the master out or 2 bus, is where all of the tracks will be summed and bounced down, to create our final stereo audio file for playback. This channel can be treated exactly the same as any other channel with FX applied to it.

Remember that the aim of mixing is to take a number of separate tracks and blend them together to a single stereo file for playback. A large part of this is how we mix and distribute these signals, which is why it is so important to understand the different routing capabilities.

Due to the fact that there is no physical cabling within a DAW, it can be hard to get our head around. But if we can grasp the concept of how a patch bay works and why it is needed, then we can also understand exactly how a DAW sends signals to different plugins and routings within the software.

Once we grasp this concept we will have complete control in the studio, allowing us to spread our wings and become more creative in the mixing techniques that we use.



A signal flow diagram of individual tracks, sub-mixes & the stereo output

ROUTING, BUSSES & GROUPS SUMMARY & KEY POINTS

- Signal flow is how the sounds are routed through our mixing desk or DAW to our speakers
- Signal flow involves the use of groups, busses, sends, returns, inputs and outputs
- Monitor modes let us monitor the input of a track in different ways
- IN - constant monitoring
- AUTO - Only when the record/arm is enabled
- OFF - no input monitoring
- We can use inserts for effects, the signal will then go to the next device in the chain
- Send/returns allow us to send a portion of the signal to a return track, which can then be blended back in with the original audio
- Pan-pots can be used to adjust the location of a signal between the left and right channels
- There are different coloured meters in Ableton that show us the peak and RMS values of a signal
- We can use the output section of a track to send its audio to anywhere within a DAW or to an external soundcard
- Busses can be thought of as routes, which allow us to tap off the audio at any given point.
- Groups are a useful way of keeping sets of tracks together and applying effects and level/panning changes to all of them at once
- Devices on return tracks should be set to 100% wet
- We can set send/return track to pre or post-fader

FINAL SUMMARY

In this first eBook in our 'zero to hero' mix series, we have covered the fundamentals of sound and waves. This ensures that the groundwork is laid down to build on with subsequent techniques and concepts explained in later books.

We have also covered the different issues with setting up a studio space. Including an in-depth look at different monitors and how we should set them up correctly whilst keeping any acoustic interferences or issues, such as standing waves to a minimum.

We have also looked at how we can listen critically and analyse a track in fine detail. This is useful for when we are looking at reference tracks, or deciding which approach to take when mixing someone else's track.

We have also covered psychoacoustics, and how we can use them to our advantage when mixing, as well as how they can be a problem when we are monitoring our tracks. (Such as Fletcher Munson curves.)

We have built a mix-preparation checklist that we can work through to ensure that our track is at a stage where it is ready to be mixed down. We have also taken a look at how to implement gain staging throughout a mix, to ensure that we are not running hot signals into any devices, as well as ensuring that there is no clipping above 0dBfs on the master.

Finally we have looked at all of the functions within the Ableton live mixer, as well as how we can route signals around Live for different tasks.

This first book seems like it's a lot of disparate information to take in, however it's important for forming the fundamental knowledge of sound, which we can build on in later books. Understanding these concepts at this stage will vastly improve the quality of our mix-downs.

The main points to take away from this book are:
Make sure that our composition is finalized before entering the mix-down stage.

Ensure to carry out all of our mix preparation correctly, although it may be boring and tempting to cut corners, good preparation and

housekeeping will ensure that we don't have to disrupt our creative flow to do mundane tasks once we are in the more creative stages of mixing.

Finally, our mix will only be as good as our ears, monitors and room. We must look after our ears and listen at sensible levels, treat our room as best as we can and set some money aside for a good set of monitors and ensure they are set up correctly.

If you follow all of these points to the best of your ability, then we can guarantee you will be a much better mix-engineer by the end of this series. This will save you a lot of valuable time and effort in the long run.

This is book 1 in our 'Zero To Hero Guide To Mixing In Ableton Live' mix series. For further eBooks and free content, please check out our website.

Website: www.abletonbible.com
Email: info@abletonbible.com

A Focal Press Book

SECOND EDITION

DESIGNING SOFTWARE SYNTHESIZER PLUGINS IN C++

WITH AUDIO DSP

WILL C. PIRKLE



Designing Software Synthesizer Plugins in C++

Designing Software Synthesizer Plugins in C++ provides everything you need to know to start designing and writing your own synthesizer plugins, including theory and practical examples for all of the major synthesizer building blocks, from LFOs and EGs to PCM samples and morphing wavetables, along with complete synthesizer example projects.

The book and accompanying SynthLab projects include scores of C++ objects and functions that implement the synthesizer building blocks as well as six synthesizer projects, ranging from virtual analog and physical modelling to wavetable morphing and wave-sequencing that demonstrate their use. You can start using the book immediately with the SynthLab-DM product, which allows you to compile and load mini-modules that resemble modular synth components without needing to maintain the complete synth project code. The C++ objects all run in a stand-alone mode, so you can incorporate them into your current projects or whip up a quick experiment. All six synth projects are fully documented, from the tiny *SynthClock* to the *SynthEngine* objects, allowing you to get the most from the book while working at a level that you feel comfortable with.

This book is intended for music technology and engineering students, along with DIY audio programmers and anyone wanting to understand how synthesizers may be implemented in C++.

Will C. Pirkle is a Staff Scientist at Audio Media Research, Inc. and Music Engineering Technology program director at the University of Miami Frost School of Music. He teaches a range of classes, from audio electronics to digital signal processing and audio programming. In addition to 16 years of teaching, Will has 20+ years of experience in the audio industry, working and consulting for such names as Korg Research and Development, XM Radio, and National Semiconductor Corporation. An avid guitarist and studio owner, Will enjoys projects that combine his skills.


Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Designing Software Synthesizer Plugins in C++

With Audio DSP

2nd Edition

Will C. Pirkle

Routledge
Taylor & Francis Group
NEW YORK AND LONDON

First published 2021
by Routledge
605 Third Avenue, New York, NY 10158

and by Routledge
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

Routledge is an imprint of the Taylor & Francis Group, an informa business

© 2021 Will C. Pirkle

The right of Will C. Pirkle to be identified as author of this work has been asserted by him in accordance with sections 77 and 78 of the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this book may be reprinted or reproduced or utilised in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying and recording, or in any information storage or retrieval system, without permission in writing from the publishers.

Trademark notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data
A catalog record for this title has been requested

ISBN: 978-0-367-51048-0 (hbk)
ISBN: 978-0-367-51046-6 (pbk)
ISBN: 978-1-003-05220-3 (ebk)

Typeset in Times New Roman
by codeMantra

James Robert Pirkle



Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Contents

List of Figures xiii
List of Tables xxiii
Preface xxv

1 SynthLab Introduction 1
1.1 What You Need to Know to Use SynthLab Objects and Projects 2
1.2 SynthLab Synth Projects 2
1.3 Synth Components 3
1.4 Basic Software Synth Architecture 4
1.5 SynthLab Voice Architecture 4
1.6 SynthLab C++ Implementation 5
1.7 Why SynthLab Uses This Architecture 8
1.8 SynthLab Object Operational Phases 9
Bibliography 11

2 The Synth Engine 12
2.1 Engine Behavior 12
2.2 Engine Architecture 12
2.3 Shared Data 13
2.4 SynthEngine *Constructor 18*
2.5 Audio and MIDI Block Rendering 18
2.6 SynthEngine *Operational Phases 20*
Bibliography 26

3 Synth Voices, Synth Modules, and Module Cores 27
3.1 Voice Behavior 27
3.2 SynthVoice *Modules 27*
3.3 SynthVoice *Parameters 28*
3.4 SynthModule *Members 29*
3.5 SynthVoice *Construction 30*

3.6 SynthVoice *Operational Phases 31*
3.7 SynthModules and ModuleCores 35
3.8 Module Cores 37
3.9 SynthModule/ModuleCore Relationship 37
3.10 Review 41
Bibliography 41

4 Synth Operational Modes: Polyphony and Voice-Stealing 42
4.1 The Note-Event Lifecycle 43
4.2 SynthEngine *MIDI Processing 44*
4.3 SynthEngine *Unison Detuning 46*
4.4 Voice State and MIDI Event Storage 47
4.5 Voice-Stealing 47
4.6 Polyphony and Voice Timestamps 50
4.7 Review 53
Bibliography 53

5 Learning and Using the SynthLab Objects & Projects 54
5.1 Designing Modules with the SynthLab-DM Projects 54
5.2 SynthLab-DM Modules Are Dynamic Linked Libraries 54
5.3 Using SynthModules *in Your Projects 58*
5.4 Using SynthEngines *in Your Projects 63*
5.5 SynthEngine *GUI Design and Parameter Update 66*
5.6 Programming the Modulation Matrix 69
5.7 Getting WaveSequencer *Status Meter Updates 69*
Bibliography 69

6 Modulation: Theory and Calculations 70
6.1 SynthLab Mod Knob Mapping 70
6.2 MMA Transforms & Calculations 71
6.3 Sequential Circuits Vector Joystick Envelope Modulation 74
6.4 Linear Frequency Modulation with LFO or EG 75
6.5 BPM Sync 76
6.6 Quantizing 77
6.7 Ramp Modulation: Fade-in and Fade-out 77
6.8 Pitch Calculation 79
6.9 Pulse-Width Modulation (PWM) 81
6.10 Phase Distortion 81
6.11 Hard Sync 83
6.12 Filter Key Track Modulation 85
Bibliography 86

7 Envelope Generators and DCA 87
7.1 Envelope Generator Fundamentals 87
7.2 EG Implementation: Finite State Machine 92
7.3 Digital EG Implementation: Rendering the Output 95
7.4 Biased EG Output 96
7.5 Analog EG Emulation 97
7.6 Synth Module: EnvelopeGenerator *100*
7.7 EG Retrigger Modulation 101
7.8 EG Core Programming Notes 102
7.9 The Digitally Controlled Amplifier (DCA) 106
7.10 Exercises 108
Bibliography 109

8 Low Frequency Oscillators 110
8.1 Noise Oscillators and Generators 110
8.2 Oscillator Clocking 111
8.3 LFO Waveforms and Rendering Equations 112
8.4 Render Modes 116
8.5 Waveform Shaping 117
8.6 Delay and Fade-in Times 117
8.7 Starting Phase 117
8.8 DC Offset 117
8.9 SynthLFO *and Cores 118*
8.10 LFO Core Programming Notes 119
8.11 Exercises 121
Bibliography 123

9 Wavetable Oscillators 124
9.1 Wavetable Fundamentals: Table Lookup 124
9.2 Wavetable Objects and Database 130
9.3 Wavetable Sources and Database 131
9.4 WTOscillator *and Cores 134*
9.5 Wavetable Core Programming Notes 135
9.6 Exercises 139
Bibliography 140

10 Virtual Analog Oscillators 141
10.1 VA Oscillator Fundamentals 141
10.2 Band Limited Impulse Train (BLIT) 142
10.3 Band Limited Step (BLEP) 143

10.4 Polynomial BLEP Approximation 147
10.5 Choosing the BLEP sinc *Source 148*
10.6 Other VA Algorithms 149
10.7 BLEP Square Wave 149
10.8 VAOscillator *and Core 150*
10.9 VA Core Programming Notes 150
10.10 Exercises 155
Bibliography 156

11 PCM Sample Playback Oscillators 157
11.1 PCM Sample Playback Modes 157
11.2 PCM Sample Storage 159
11.3 Loading PCM Samples 160
11.4 PCM Sources and Database 163
11.5 PCMOscillator *and Cores 166*
11.6 PCM Core Programming Notes 168
11.7 Exercises 170
Bibliography 171

12 Synthesizer Filters 172
12.1 Design Summary 172
12.2 Q *and Self-Oscillation 173*
12.3 Analog Magnitude Matching at Nyquist 173
12.4 Zavalishin's Virtual Analog Filters 173
12.5 Resolving Delay-Free Loops in VA Structures 175
12.6 VA Filters from Primitive Analog Block Diagrams 180
12.7 VA Filters from Signal Flow Graphs 183
12.8 VA Filters from Conceptual Signal Flow Graphs 191
12.9 Nonlinear Processing and Self-Oscillation Control 191
12.10 Synth Filter Objects 193
12.11 Input Drive and Output Peak Limiter 197
12.12 SynthFilter *and Cores 198*
12.13 Synth Filter Core Programming Notes 199
12.14 Exercises 204
Bibliography 205

13 Karplus-Strong Plucked String Model 207
13.1 The Exciter-Resonator 207
13.2 The Plucked String 208
13.3 The Karplus-Strong Model 208
13.4 Pluck Position 212

13.5 Karplus-Strong Algorithm Equations 213
13.6 Karplus-Strong C++ Objects 214
13.7 KSOscillator and KSOCore 218
13.8 KSOCore *Member Objects 219*
13.9 Core Programming Notes 219
13.10 Extensions to the KS Algorithm 222
13.11 Exercises 223
Bibliography 224

14 The Modulation Matrix **225**
14.1 Modulation Inputs and Outputs 225
14.2 Modulation Routings 225
14.3 Mod Matrix Channel Routing 228
14.4 ModSource *and* ModDestination *for GUI Controls 228*
14.5 ModMatrixParameters *229*
14.6 The ModMatrix *Object 230*
14.7 Initializing the ModMatrix *231*
14.8 Programming the ModMatrix *232*
14.9 Mod Matrix Transforms 232
14.10 Running the ModMatrix *233*
Bibliography 233

15 Wave Morphing and Wave Sequencing 234
15.1 Wave Banks 234
15.2 Wave Morphing 235
15.3 The MorphWTCore *236*
15.4 MorphWTCore *Programming Notes 237*
15.5 Wave Sequencing 1.0 240
15.6 SynthLab WaveSequencer *243*
15.7 The WSOscillator *Object 248*
15.8 WSOscillator *Programming Notes 251*
15.9 Exercises 255
Bibliography 256

16 The SynthLab Synth Projects 257
16.1 SynthLab Modulation Matrix 257
16.2 SynthLab Render Phase: Modulators 259
16.3 SynthLab Render Phase: Quad Oscillator Mixing 260
16.4 SynthLab Render Phase: Filtering 261
16.5 SynthLab Render Phase: Global Volume and Delay FX 262
16.6 SynthLab-DX: the FM Synthesizer 263

16.7 FM/PM Rules 267
16.8 FM Operator 268
16.9 SynthLab-DX Voice Render Phase 271
Bibliography 272

Index 273

Figures

1.1 The MMA representation of a software synth includes control logic, articulation, and an audio engine that renders the synthesized signal 5
1.2 SynthLab voice architecture includes two filters that may be processed in series or parallel, or individually bypassed; the SynthLab-WS project also includes a wave sequencer object 6
1.3 (a) SynthLab C++ objects include the engine, voice, modules, cores, and their associated string lists; (b) typical GUI layout for the wavetable oscillator 8
2.1 The *SynthEngine* object architecture; the dotted boxes show the interface points to your plugin framework's processor object 14
2.2 Hierarchy of SynthLab objects responding to function calls from the host plugin framework processor object 14
2.3 (a) The engine owns a stack of voice objects (four in this example but 16 in the SynthLab projects), each of which includes an identical set of LFO, filter, EG, oscillator, and DCA objects; (b) the engine owns a voice parameter structure that is shared across these voices; and (c) the shared voice parameters structure consists of a set of shared module parameter structures 16
2.4 *SynthEngine* operation consists of three phases: (a) initialization, (b) GUI updates, and (c) audio rendering 22
2.5 *SynthEngine* MIDI processing 23
3.1 Flow of function calls from host plugin framework processing object to the *SynthEngine*, then to the voices, modules, and their cores 28
3.2 SynthLab modules each implement one synth building block; all are subclassed from *SynthModule*; note that the normal and morphing wavetable oscillators are combined (dotted box) and may be mixed freely 28
3.3 (a) Module diagrams show attributes and connections in the module and its core, along with module and mod knob strings for each core; notice that the *moduleStrings* show up in the GUI as a selection list and (b) the *SynthModule* block diagram 36
3.4 (a) The PCM oscillator includes three pre-written cores and one empty (user) core that allow the PCM oscillator to implement banks of waveforms; (b) the relationship between voice, module, and core is tight 38
4.1 The four states of the note-event lifecycle: (a) the note-on message resets objects and places the amp EG into the attack state; (b) the note-off message places the amp EG into the release state; (c) the voice waits for the amp EG to finish, rendering output the entire time; and (d) the note-event is finished when the amp EG reaches the off state 43

4.2 The engine decodes the incoming MIDI messages; for note-on and note-off, the engine first finds a target voice, then forwards the MIDI processing function call 44
4.3 An engine with four voices: (a) prior to any notes triggered (b) when the first note is triggered, voice 0 is selected then (c) second note and (d) third note and (e) fourth notes are triggered with timestamps incremented each time and (f) after the oldest voice is stolen, it becomes the youngest and so on 52
5.1 You can create internal modules for the SynthLab-DM synths without needing to recompile the entire project; this shows a module named Drum WT DLL that exposes a set of waveform and mod knob strings for the user to see 55
5.2 GUI controls and their relationship to the custom parameter structure; the controls in grey include the dynamic strings and are different for each kind of module and core 68
6.1 (a) MMA concave and (b) MMA convex transforms; (c) MMA velocity to attenuation in dB and (d) square law converted velocity to normalized gain graphs 72
6.2 (a) Constant power curves for panning and crossfading with solid line for trigonometric and dotted line for square law transforms, and (b) the *XFader* C++ object crossfades between two signals, A and B, using linear, trig, or square law crossfading 73
6.3 (a) The original Sequential/Korg vector joystick; (b) the rotated version is simpler for GUIs (the dotted lines show the actual *x*- and *y*-axes); and (c) visualization of the four envelopes that are generated as the stick moves from points 1 to 3, where it stays during the sustain period, and note-off moves the joystick to its final position 4 75
6.4 (a) The modulo counter with *x*-axis as period and *y*-axis as phase with a slope of 1.0, (b) a normal sinusoid lookup, (c) and (d) relocation of the breakpoint produces two slopes *m1* and *m2* while (e)–(f) show the distortion on a sinusoid and (i)–(l) show a ramp modified with phase distortion 82
6.5 Hard sync with (a) ramp/sawtooth and (b) square-wave main oscillators, while (c) and (d) show the effect of moving the reset point for ramp and triangle waves; notice that the reset oscillator's waveform is not used other than as a resetting device 84
6.6 Crossfading to smear over a hard-sync discontinuity for (a) the normal reset and (b) reset to a non-zero starting phase 84
7.1 (a) ADSR envelope generator with linear (dashed) and exponential segments (b) specifying the EG using levels (L) and rates (R), and (c) the decay and release times are actually calculated from the full-scale value to zero and (d) changing the sustain level with all else constant changes the perceived decay and release times 88
7.2 More EG contours including (a) AR, (b) AHR, (c) DAHR, (d) the SynthLab *DXEG* with linear slope segment (SL), and (e) the VCS3 re-triggering "trapezoidal" EG 89
7.3 Early analog EGs with (a) trigger and gate signals, and (b) EG with s-trigger signal, (c) the EG restarts with each note on event from the previous output value and (d) the EG is reset back to the starting point with each note on event; the logic is shown as active-high but may be implemented as active-low as well 90
7.4 Legato playing with (a) a trigger/gate EG and (b) an s-trigger EG 90

7.5 (a) The note-off event occurs before the EG reaches full amplitude and results in (b) a truncated version while (c) shows the unconditional release mode 91
7.6 The circles show the state transition triggers for the (a) ADSR and (b) *DXEG* with added slope state; this EG may also have the decay level L_{DCY} set above the sustain level, and the transition trigger logic is inverted 92
7.7 The FSM diagram for the ADSR EG with shutdown; this does not show the monosynth legato or reset-to-zero options to simplify the diagram 94
7.8 (a) Linear EG with constant step sizes within each state and (b) approximating an exponential EG with variable step sizes; the axes are unlabeled to show that the numerical limits of min and max are variable 95
7.9 (a) The concave and (b) convex transforms are used to give contour to the linear segments, and (c) shows the overall concept; note that the slope segment is kept linear 97
7.10 (a) When using the EG to modulate pitch, all of the portion above 0.0 is sharp, including the sustain segment, and (b) the biased EG pulls the sustain back in tune but causes the start and end of the note-event to be flat; adjustment of the start and end values mitigates this 97
7.11 (a) SW1 closes for the note-on event, and the voltage source charges capacitor C through R_{ATT}, then (b) SW1 opens, and SW2 closes for the note-off event, which discharges the capacitor through R_{REL}; (c) shows the resulting event; note that V is the asymptotic charge voltage, and $0.77V$ is 77% of this value, used for the analog emulation 98
7.12 Attempting to generate a normalized exponential decay over the range of x = [0, 1] using (a) the first order feedback structure, we observe that the output does not decay to 0.0 and takes on a non-zero value at x = 1.0, while through (b) the addition of a negative bias signal and adjustment to the b coefficient (b'), we can achieve the proper decay curve and hit, then cross over the 0.0 value 99
7.13 The *EnvelopeGenerator* module and the three included cores: DX-EG, analog EG, and linear EG; the module is shown with the *DXEGCore* selected 101
7.14 Conceptual block diagram of the *DXEG* with the curvature control; the concave and convex transforms are applied to the appropriate segments 102
7.15 The decay and release portions of the EG curve need to be mapped to the correction sections of the normalized concave transform 105
7.16 (a) The EG intensity acts as a simple gain control when positive, but (b) when negative, it inverts the unipolar signal 106
7.17 Two more EG contours for the (a) Yamaha EX and (b) Casio CZ series synths; the contours are shown as linear here for simplicity; most segments are curved 109
8.1 The frequency spectra of (a) white and (b) pink noise; FFT length is 131,072 points 111
8.2 (a) The modulo counter steps upward by the *phaseInc* on each sample period until the counter crosses 1.0 and wraps to the same offset, and (b) the block diagram form of the *SynthClock* object 112
8.3 Triangle and ramp waveforms and their algorithms; the ID values correspond to the waveform selector in SynthLab 113
8.4 Exponential, sinusoidal, square, random, and pluck waveforms and their algorithms 114

8.5 Frequency spectra and equations for three popular sinusoidal approximations: quadratic, Bhaskara I, and parabolic, with f_o = 100 Hz 115
8.6 Random sample and hold LFOs with (a) random amplitude and constant hold time, and (b) random amplitude and random hold times 115
8.7 Stepped LFO waveform examples: (a) three-step sine, (b) six-step triangle, and shaped LFO waveforms (c) triangle and (d) ramp up with shape control = (0.0, 0.5, 1.0) 116
8.8 The *SynthLFO* and its two cores block diagrams, module strings, and mod knobs 118
8.9 Additional LFO waveforms for your custom object 122
9.1 Reading an N = 8 point wavetable with (a) integer lookup and (b) floating point lookup with (c) linear interpolation and (d) the read index wraps around to the start of the table when the added phase increment pushes the lookup point past the table's ending index 124
9.2 The 88-note piano keyboard starts at A0 (MIDI note 21), with 801 harmonics, and ends at C8 (MIDI note 108), with 5 harmonics; f_s = 44.1 kHz and A0, A4, and A7 are shown 126
9.3 (a) Waveform and (b) spectrum of a sawtooth synthesized with 14 harmonics plus the fundamental and (c) the Lanczos corrected waveform and (d) resulting spectrum; the harmonic envelope is shown with a dotted line 127
9.4 (a) Wavetable spacing in the Prophet VS and (b) modern wavetable synth using 54 high-resolution tables such that every note has every harmonic up to Nyquist 129
9.5 Flowchart of a system to generate band limited wavetables from a seed table; the process is repeated as many times as you need 129
9.6 (a) The sine and drum sources only store and read a single wavetable, while (b) the static and dynamic sources each store an array of wavetables, one per MIDI note, while (c) the *WavetableDatabase* contains a dictionary of *IWavetableSource* pointers whose keys are unique strings that name the waveforms; (d) a wavetable bank is a name given to a set of wavetables used for wavetable morphing 132
9.7 The *WTOscillator* and the three included cores: classic, morphing, and drums; two special bonus cores are also included for Fourier synthesis and sound effects 135
9.8 Wave scanning may be used with simple waveforms (a) or more complex one-shot drum tables (b) to produce new sounds 140
10.1 A trivial sawtooth waveform spectrum with (a) −60 dB and (b) −96 dB lower limits, and a VA oscillator spectrum with (c) −60 dB and (d) −96 dB lower limits 142
10.2 (a) A smooth continuous function, (b) a ramp discontinuity, and (c) square discontinuity both have similar mathematical traits – identical slopes on each side of the edge, while the triangle in (d) has two different slopes on each side of the discontinuity 142
10.3 (a) Unipolar and (b) bipolar BLIT signals, and (c) algorithms for converting them into traditional waveforms 143
10.4 (a) A *sinc* function and (b) its integration produces a wavy step; (c) to use the BLEP, superimpose it on top of the discontinuity; notice that only a small section of the BLEP waveform is used 144
10.5 Two sets of windowed BLEP waveforms, where (a) the central pulse is truncated and windowed and (b) the central pulse is combined with the next pair of zero-crossings to form a wider windowed function; the integrated windowed signals, and the rising and falling residuals are shown for each case 145

10.6 (a) A rising discontinuous edge showing the two samples on each side (b) superimposing the residual over the discontinuous edge (c) adjusting these four points based on the residual and (d) the BLEP corrected discontinuity 145
10.7 (a) Geometrically determining the distance from the discontinuity to points on either side of the edge; (b) a triangle pulse integrates to a smooth sigmoid, similar to that of Figure 10.5(a); and (c) the BLEP rising edge residual is approximated with two second order polynomials, one for each side of the edge 147
10.8 Sawtooth waveforms using 8-point BLEP correction with (a) rectangular (b) triangular (c) Hamming and (d) Blackman-Harris windowed *sinc* BLEP residuals; the dotted line represents the harmonic envelope of a perfect sawtooth waveform; note the location of the −60dB level 148
10.9 (a) Two sawtooth waveforms are combined to generate a square wave (b) shows how the ramps add or subtract to produce values of −1.0 or +1.0 150
10.10 The *VAOscillator* (a) internal block diagram and (b) SynthLab diagram; there is only one core, which is also available for download as a SynthLab-DM project 151
11.1 The four common modes of PCM sample playback include (a) one-shot; (b) loop; (c) transient plus loop; and (d) transient, loop, and release 158
11.2 The Mellotron core directory contains subdirectories for each patch, which, in turn, contains a set of wave files; each of these is extracted into a *PCMSample* object and stored in an array of pointers in an *IPCMSampleSource* object 164
11.3 (a) The sample sources are arranged in a database and accessed via a unique name string; (b) the shaded keys show the included wave files for the Mellotron, sampled on every note from G2 through F5, while the (c) legacy files include samples on minor third boundaries from A1 through C4 165
11.4 The *PCMOscillator* block diagram and module strings; none of the mod knobs are assigned in these cores 167
12.1 (a) A simple RC lowpass filter and (b) the analog block diagram 174
12.2 (a) The bilinear (trapezoidal) approximation of the area under a curve, showing the error portion between the dotted line and the curve, and (b) transposed canonical form of the bilinear integrator 175
12.3 (a) The analog integrator is replaced with a digital bilinear integrator (b) a shorthand version after combining the two input multipliers into one, and ignoring the exact details of the bilinear structure and (c) simplified notation showing a block with input, output, *g*, and *s* ports 176
12.4 (a) The delay-free loops structure with $v(n)$ labeled and (b) the resolved loop done by examination of Equation (12.7) 177
12.5 The first three steps in the MH method for resolving delay-free loops produce the values $u_o(n)$, $y_o(n)$, and loop gain X 178
12.6 (a) The final structure after loop resolution and (b) combining the series loop coefficients into a single value α 179
12.7 Primitive analog block diagrams and their resulting VA filter structures for (a) first order lowpass filter, (b) first order high-pass filter, and (c) first order all-pass filter (d) the normal first order BZT LPF frequency response with zero gain at Nyquist (e) analog FGN LPF frequency response (f) first order HPF frequency response and (g) first order APF phase response; all plots are shown with f_c = 100 Hz, 1 kHz, 5 kHz, and 10 kHz 181

12.8 (a) The SVF analog block diagram (b) the VA realization including the analog matched LPF output $y_{LPM}(n)$ (c) normal SVF LPF frequency response and (d) LPF frequency response with analog matching at Nyquist (e) SVF HPF frequency response with f_c = 100 Hz, 1 kHz, 5 kHz, 10 kHz and 15 kHz and Q = 25 and (f) SVF BPF frequency response with f_c = 2.5 kHz and Q = 0.5, 10 and 25 183
12.9 The signal flow graph and VA realizations for (a) Korg35 LPF, (b) Korg35 HPF and (c) the combined structure that is used for each of the Korg35 sub-filters (d) the LPF and HPF frequency responses show differences in the roll-off slopes with LPF f_c = 500 Hz and HPF f_c = 5 kHz and (e) the Korg35 LPF frequency response with analog Nyquist matching and f_c = 100 Hz, 1 kHz, 5 kHz and 15 kHz with K = 1.9 185
12.10 The Moog ladder filter (a) signal flow block diagram (b) notice how the bass response is reduced as the Q is increased shown with f_c = 100 Hz, 1 kHz, 5 kHz and 15 kHz each with increasing Q (K) value; the BZT and analog FGN curves are shown for $K = 3$, and for $K = 3.9$ the analog FGN response is indicated (c) the modified additional outputs generate three more filter types including a resonant first order variety 186
12.11 The Moog ladder filter (a) VA realization with optional input scaling and output branches, and (b) the analog FGN version; note that the β coefficients are contained in each sub-filter but shown externally here to convey their operation in the feedback loop 188
12.12 The diode ladder filter's (a) signal flow block diagram, and (b) notice how the bass response is reduced as the Q is increased shown with f_c = 500 Hz, 1 kHz, 5 kHz and 15 kHz each with increasing Q (K) value and (c) the experimental analog FGN output at f_c = 2.5 kHz, 5 kHz, 10 kHz and 15 kHz; the error in the resonant peak frequency is normal and correct for this filter 189
12.13 (a) The modified first order VA LPF (named mLPF) and (b) its shorthand block diagram with arrows showing the signal flow directions (c) the diode ladder filter realization and (d) the experimental analog FGN version with added α_1 coefficient at the output; note that the β coefficients are calculated within each sub-filter but are shown externally here to convey their operation in the feedback loop 190
12.14 Novel VA filters contrived from conceptual signal flow graphs, including (a) second order LPF with f_r = 500 Hz and f_c = 1 kHz, (b) resonant first order high shelving filter with $f_c = f_r$ = 2 kHz and a doubly-resonant LPF with f_c = 500 Hz, (c) f_r = 1 kHz, and (d) f_r = 3 kHz 192
12.15 Nonlinear blocks in the VA structures shown inside (a) the saturating integrator, (b) the Korg35, and (c) Moog signal flow block diagrams, while (d) shows the location of the peak limiter on the filter output, always outside of any feedback loop 193
12.16 The *SynthFilter* and core block diagram, filter types, and mod knob labels 198
12.17 The Oberheim variations on the Moog ladder signal flow block diagram and table of coefficient values 205
13.1 (a) The most basic exciter-resonator system, and (b) a more complex model that includes coupling from the resonator back to the exciter and an output impedance coupler 207

13.2 (a) An acoustic guitar with nut, bridge, and body; (b) plucking the string deforms it and (c) sets up a transverse wave that moves down the string, and (d) bounces off the nut and moves back toward the bridge 209
13.3 (a) Basic model of a plucked string with dual delay lines for transverse wave oscillation and (b) a simplified version that combines like components and adds the fractional delay APF 209
13.4 (a) A first order structure for an APF with only one coefficient *a* to calculate and (b) the group delay versus frequency for the APF; each curve represents a different APF f_c 210
13.5 (a) The simple first order feed-forward LPF and (b) its frequency response showing a zero at Nyquist 210
13.6 (a) Rectangular windowed noise burst; (b) Hann windowed noise burst; (c) sonic grains from re-triggered windowed exciter; (d) attack-release; and (e) attack-hold-release envelope noise bursts 211
13.7 (a) The pluck position filter is a simple feed forward comb filter that is combined with the bridge integrator in the delay branch, (b) shows the comb filter response for a pluck position of ¼ the total length (M = 4), (c) the bridge filter's lowpass-integrator response and (d) the combined integrator and comb filter responses 213
13.8 Spectra of the KS model of the note A-2 (110 Hz), showing output of (a) bridge filter only and (b) bridge filter and pluck position with pluck at ½ the length of the string 213
13.9 The block diagram of the KS algorithm packaged as the *KSOCore* includes the exciter, pluck, pickup, and bridge filters, and the resonator and body filter; the inset shows the very simple guitar preamp simulator, consisting of eight stages of waveshaping followed by a second order LPF 215
13.10 The *KSOscillator* and *KSOCore* block diagrams, module strings, mod inputs, and GUI controls 219
14.1 (a) The modulation input array is indexed the same way for all *SynthModules* using constant declarations for kEGMod, kBiasedEGMod, etc… (b) the LFO Core exposes a modulation input and a modulation output (c) the modulation output array is indexed specifically for the LFO using constant declarations while (d) shows the indexing for the EG's modulation output array for comparison 226
14.2 A modulation routing that connects LFO1's normal output to OSC2's bipolar modulation input, which modulates the oscillator's frequency 226
14.3 Modulation matrix designs include (a) pin/button programmable with global source and destination intensity controls, and (b) a limited choice matrix with individual channel routing intensity controls 227
14.4 (a) The *ModMatrix* configuration for a single routing shows the multiple intensity controls and enable-switches; in this example, the hardwire enable switch is overriding the channel branch while (b) and (c) demonstrate the routing configurations that match Figures 14.3 (a) and (b), respectively 228
15.1 Three sets of tables for morphing, including (a) dissimilar waveforms, (b) one waveform whose parameters are adjusted for each slice, and (c) a complex waveform that has undergone multiple passes of filtering or other processing 235
15.2 (a) An LFO modulates the wave-morphing index, producing a smoothly interpolated succession of waveforms back and forth along the range, while

(b) the EG also morphs the waveforms but stops during the sustain portion, holding the current waveform constant prior to release 236
15.3 The *MorphWTCore* snaps into the *WTOscillator* object and exposes its mod knobs, which are slightly different and include morph start index and morph mod intensity; notice that the module strings point to a bank of wavetables 237
15.4 Korg's wave sequencer from the Wavestation featured the ability to hold and crossfading waveforms while also modulating each segment's pitch and amplitude; the waveforms actually blend together during the crossfades and are shown separated for easier viewing; the amplitudes were not displayed in dB in the Wavestation 240
15.5 (a) The Wavestate's wave sequencer splits out each lane with its own start, stop, loop start, and loop end points (b) a possible combination of lanes showing the lane steps marked (1) and (2); the timing lane is used to synchronize the other lane step durations and crossfade timing 242
15.6 SynthLab's wave sequencer interface includes individual settings for all steps and a global control over the lanes 243
15.7 (a) The *WaveSequencer* and its member objects; (b) the *XHoldFader* can hold and crossfade signals or values, and (c) it can output the crossfade gain values in the *XFadeData* structure and operate in both modes at once, if needed 244
15.8 (a) A set of timing and waves to be sequenced shows the hold-then-crossfade pattern where you (b) hold wave A and crossfade to B on first step then (c) wave B is labeled as the new wave A and the next segment is loaded as wave B (d) the hold-then-crossfade process repeats with each new segment 246
15.9 (a) A *Lane* structure maintains an array of *LaneStep* structures that are sequenced according to a jump table; the current step/next step pairs are shown in white and grey boxes (b) after the timing lane's first crossfade, the current step/next step pair advances according to the loop direction (to the right in this example) (c) without randomization, the segments will loop in the same order after the loop end point is hit (d) with randomization, the jump table is shuffled after each loop restarts which then shuffles the lane steps accordingly 248
15.10 The four *WSOscillators* are used in a round-robin manner starting with (a) the oscillator pair (0, 1); after the first crossfade is completed, (b) the sequence shifts to oscillator pair (1, 2) then (c) to pair (2, 3) then (d) wrapping around to pair (3, 0), and so on 249
16.1 SynthLab voice architecture; the quad oscillator bank is different for each project, and the *WaveSequencer* is only in SynthLab-WS 258
16.2 The SynthLab modulation matrix uses a pin-programmer style interface and source and destination intensity controls 259
16.3 (a) FM and (b) PM time and frequency domain output signals; the spectral amplitudes shown are for visual reference; the actual amplitudes follow Bessel functions and are more complex 265
16.4 (a) A modulation index control modifies the output spectrum; (b) replacing the control with an EG allows the spectrum to morph as the note-event progresses 266

16.5 (a) An FM operator consists of a sinusoidal oscillator and EG that adjusts its output amplitude along with Chowning's FM operator notation (b) a two-operator patch that includes self-modulation; notice that Operator 1's output EG acts as the amp EG since it is the very last in the chain and Chowning's notation that indicates self-modulation 266
16.6 The eight DX-100 FM algorithms 267
16.7 (a) The block diagram for the *FMOperator* object includes a self-modulation routing path and FM intensity control, and (b) the SynthLab implementation block diagram 269


Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Tables

1.1	Synthesizer components, my abbreviations, and descriptions, as used in *SynthLab*	3
3.1	*SynthVoice* modules that are identical across all synth projects	29
3.2	*SynthModule* virtual functions that all modules must override and descriptions	37
4.1	Voice detuning, panning, and starting phase offsets for unison mode	47
5.1	SynthLab's module types, their container objects, and the downloadable examples	57
7.1	*EnvelopeGenerator* custom parameter structure and cores	100
8.1	LFO cores and their special modulation capabilities (*) downloadable dynamic module projects	118
8.2	*LFOCore* member objects	119
9.1	The built-in *IWavetableSource C++* objects and descriptions	131
9.2	The *IWavetableSource* interface	133
9.3	*WTOscillator* custom parameter structure and cores: (*) downloadable SynthLab-DM core project (*) downloadable dynamic module projects	134
9.4	Module controls shared across cores	135
9.5	Wavetable cores and their special modulation capabilities	135
10.1	Number of BLEP correction points per side versus oscillator frequency	147
10.2	*VAOscillator* custom parameter structure and cores: (*) downloadable SynthLab-DM core project	151
10.3	*VAOscillator* module controls	151
11.1	The *PCMSample* object's functions	163
11.2	The *IPCMSampleSource* interface	165
11.3	*PCMOscillator* custom parameter structure and cores: (*) downloadable SynthLab-DM core project	167
11.4	Module controls shared across cores	168
12.1	High order VA filters and their sub-filter types and arrays	194
12.2	Synth filter models and their output types, along with notes on implementation	197
12.3	*SynthFilter* custom parameter structure and cores: (*) downloadable SynthLab-DM core project	199
12.4	The *SynthFilter* core GUI controls	199
13.1	Some common instruments and their physical modeling components	208
13.2	List of lightweight objects specifically for KS algorithms	215
13.3	The *Exciter* members and brief description	216
13.4	The *Resonator* members and brief description	217
13.5	The *PluckPosFilter* members and brief description	218

13.6 *KSOCore* member objects/variables 220
15.1 *WaveSequencer* C++ objects and structures 247
15.2 The *WaveSequencer's SynthModule* overrides and descriptions 249
15.3 The *WSOscillator* members and descriptions 250
15.4 The *WSOscillator's SynthModule* overrides and descriptions 250
16.1 SynthLab projects and oscillators 258
16.2 SynthLab oscillators and their special modulation destinations 259

Preface

This book is the complete opposite of the first edition. A main influence for this change is a documentary I watched a few years ago that involved a Canadian rock power trio and the soul searching they did regarding their fourth album and its content. I followed their design pattern for my fourth offering, in more ways than one.

In early May 2020, I sat on my couch, enjoying my Saturday morning ritual of Bugs Bunny, coffee, and guitar practice (it was Bach's cello suites), glad that the COVID-semester was finally over and thinking about how I was going to change the way the synth projects would be presented for the new book, including distribution from my website. The first edition was overblown, and both my writing and coding styles had changed dramatically, so I wanted a different approach. I had already assembled the bulk of the code during that spring semester. It had been through layers of kaizen-ing over the last few years, but I had also added some newer components, like morphing wavetables, and I had moved part of my synth class at the University of Miami into note-sequencing rather than just the synth signal processing. A short consulting gig the previous winter had provided me with some advance notice on a killer new synth that would make its debut at the 2020 National Association of Music Merchants (NAMM) show shortly thereafter, and I had since combed through the well-written manual repeatedly, deciding to redo my LFOs to mimic some of that synth's features. And I simply had to create my own version of its brilliantly imagined, independent loop, multi-lane sequencer for the book. But, as I listened to Stalling's "Powerhouse Theme" – of which I will never tire – and watched Bugs torment a construction worker who was trying to exterminate him, I kept thinking that I was only making things more complicated and difficult with my new trick LFOs and wacky looping sequencer that would also work with morphing wavetables and fixed tables or samples. This new book needed simplicity, not more overblown complexities; it needed to be the opposite of the first edition.

I had received an advance copy of Gabrielli's *Developing Virtual Synthesizers with VCV Rack* from my editor to use as a reference and to ensure that my book would not overlap with much of its content (it doesn't), and it was sitting on my coffee table. I had read the text and found myself very jealous. Designing a synth module – a tiny part of a much larger synth structure – was much different than designing a semi-modular software synth. I sat in my Saturday morning daydream bliss, thinking about how cool it would be to just write modules rather than dealing with voice-stealing, MIDI decoding, synth engines, modulation matrixes, and voice architectures. And, having watched students struggling with large synth projects with complicated compiler setups, I thought of how refreshingly nice it would be to work on a tiny plugin component requiring only a handful of files that would compile almost instantly and could be immediately tested within the

context of a full synth product on Windows and MacOS alike. But I also knew from experience that a book formatted with projects like that would be met with ire from the many engineers; potential readers; and, of course, the numerous forum post authors who would disagree vehemently, wanting to see the design of completed synths, polyphony, voice-stealing, and all the gory details.

As Bugs wrapped up his final gag, involving dynamite, blasting caps, and concrete, I had one of those Saturday morning ideas that makes you cock your head and smile. I imagined a synth plugin that allowed users to compile and load their own relatively tiny synth modules into it at runtime, but not in the modular fashion, with patch cables and flashing lights. Modules for LFOs, EGs, oscillators, and filters, or other kinds of components and processors, could be loaded and attached to the existing GUI controls – after all, every LFO has a frequency knob, every EG has an attack control, and oscillators are just begging to be detuned. What if there were four assignable controls that could map to anything the programmer desired, whose labels could be changed during the module loading operation? And how about incorporating a dynamically filled string control that would allow the programmer to display a set of waveform names, filter types, or other parameters for the user? That modular option would provide readers of all levels a path to understanding each module's functionality, which roughly corresponds to each book chapter. You could start out testing, modifying, and designing these small modules with only a few source files, and if you were interested in building modules for VCV Rack® or Cherry Audio's Voltage Modular®, you could lift most of that code directly.

For readers who want to incorporate the synth objects into their own plugins external to any other product, the objects need to have a stand-alone mode of operation and should not need the rest of the synth components, or the container synth plugin that manages the small modules, to function. And, for readers that want the whole shebang, each entire synth needs to be packaged in a single C++ object for easy integration with the user's chosen plugin framework. Finally, the C++ objects and synth modules need to be framework and plugin-API agnostic, not requiring any third-party code. Thus, I would need to write a tiny API to use for the synth modules, which would need to be as lean and thin as possible, and, likewise, could not be platform dependent. This would become a kind of open-architecture synth platform, and groups of programmers could convert the skeleton synth into any kind of synth they wanted while allowing the underlying engine and voice objects to handle the MIDI and other chores.

SynthLab™ was born that morning, and the module core API was completed by lunchtime. It was so lean and light, the test modules compiled in a matter of seconds. And having written numerous custom plugin APIs, I already knew the pitfalls to avoid, involving crossing the thunk barrier – an imaginary line that separates a process and the DLL it loads, and over which function calls pass their arguments. The test AU and VST plugins worked flawlessly; I now had a simple platform to use for teaching, and users would have an open-architecture platform for experimenting with new ideas. The book could certainly detail and expose all of the operations of the lower-level engine and voice objects, and users could implement them in their own products. But the individual chapters could focus just on these "cores" – the mini-plugins you load into SynthLab that are the modular building blocks of all synths. There are six complete synth projects in the book and available for download. All of them have precompiled AU and VST plugins for Windows and MacOS that you can demo and, of course, modify with your own individual module cores of all flavors.

Thanks to Hannah Rowe, my Focal Press editor, who encouraged me and helped in the premise for the 2nd edition. In addition, I need to say thanks to all those at the University of Miami's Frost School of Music, who've both supported and entrusted me with the Music Engineering Technology program. They include Dean Shelly Berg, Rey Sanchez, and Serona Elton, along with my

faculty colleagues Joe Abbati; Chris Bennett; Dana Salminen; and, of course, my old boss, Ken Pohlmann. I must include gracious thanks to my colleagues at Audio Media Research, Scott Mire, Michael Ljunggren, and José Paiz, for their encouragement and patience during the writing process, and their thoughts on the SynthLab paradigm and what it may bring to the future of music technology education.

As with all my books, I look forward to hearing what you've cooked up in SynthLab. You can always find me at www.willpirkle.com!

Will Pirkle
October 15, 2020


Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

1 SynthLab Introduction

SynthLab™ is the name of a set of C++ objects, structures, and functions designed to encapsulate and implement each of the functional blocks in a software synth, or soft-synth. There are scores of objects and structures, including abstract base classes, abstract interfaces, and all of the derived classes that implement the SynthLab synth projects. There are multiple projects for different types of synths, and you have access to all of the code for all of the objects and synth projects. But I've gone a step further to get you playing with the code and learning SynthLab without needing to integrate the objects into your plugin framework and then compile, debug, and test the complete synth projects, which are more complex than the audio effect plugins in my *Designing Audio Effect Plugins in C++*, 2nd ed. (I refer to this as "my FX plugin book" in this text).

> In addition to the C++ code and objects that you may compile and integrate into your own projects, you may also download the pre-compiled versions of each of the six different synths for VST3 (Windows® and MacOS®) and AU (MacOS). These are called the SynthLab Dynamic Module (SynthLab-DM) projects. I designed SynthLab in a highly modular format to include the use of "module cores" that are small, complete synth objects which implement the soft synth building blocks. You may design, compile, and then *dynamically load your own modules into the host DM synths*. This will allow you to study and learn each synth component, modify the code, or invent your own designs, then load those modules into the synth at runtime to debug, test, and voice. If you are interested in designing modular components for systems like VCVRack, then these tiny synth modules represent each of those modular building blocks – oscillators, filters, envelope generators, and the like.
>
> In addition, all of the underlying building block objects feature a "stand-alone mode" of operation, so you can integrate them into existing projects right away without needing to manage entire synth projects.

The base classes and interfaces, like all of the smaller helper objects and functions, may be accessed with just two includes: *synthbase.h* and *synthfunctions.h*, and their.*cpp* implementation files. You can find the C++ code and projects at www.willpirkle.com or https://github.com/willpirkleaudio/synthlab.

1.1 What You Need to Know to Use SynthLab Objects and Projects

You will notice that there is no mention of plugin APIs in this book – AAX, AU and VST are absent as they are covered in detail in my FX book. There is no discussion about specific plugin frameworks – ASPiK, JUCE, IPlug2, etc. Likewise, there are no analog or DSP theory chapters, so you will need to bring your own knowledge to the table or be willing to study it alongside this book. There is also no "intro to MIDI" chapter. You need to know how MIDI messages work, what a MIDI "CC" means, and the like. There are numerous books and sources available on all of these topics. These are prerequisites for getting the most out of this text.

This book is not designed to feature any particular framework or API but rather shows how I create software synthesizers in C++ using DSP theory and coding implementations. Most chapters begin with theory of operation, then show the C++ objects I use to implement that theory. You will need to download the code and study it side-by-side with the text as the book only includes very specific C++ code: stuff that is non-intuitive or very specific to audio or synthesis. Since all of the synth components use the same module-core paradigm and ten-control GUI implementations, the book figures and specifications are cleaner and easier to understand. This is in stark contrast to the book's first edition, in which each project was vastly different in architecture and components.

If you intend to build the complete synth projects in this book, you need to have a solid grasp on your chosen plugin framework, and you especially need to understand how to generate your own GUIs within that framework; you will also need to understand some advanced GUI design topics, such as dynamic loading of string lists (remember that you are free to pick and use individual C++ objects and code as you like for your own synth projects). You will find that adding the synth objects to your framework's processing object is very straightforward, but the GUI details may be challenging. Chapter 5 covers the details on using the module cores, designing with objects in stand-alone-mode, and connecting the synth engines to your framework, and you can always get more help and information from www.willpirkle.com/forum.

1.2 SynthLab Synth Projects

To demonstrate how to use and combine these objects, and how they work together, I have created a set of synth projects based on the same fundamental architecture which only differ in their oscillators and waveform rendering. All synths feature monophonic, unison, and polyphonic operational modes. These projects include:

SynthLab-WT: four-oscillator wavetable synth that includes ordinary wavetable, morphing wavetable, and one-shot drum and sound effect (SFX) wavetable implementations; the various oscillators may be used in any combination

SynthLab-PCM: PCM sample playback synth with four separate PCM oscillators; this synth uses. wav files for its PCM sample storage and retrieval

SynthLab-VA: four-oscillator virtual analog synth using virtual analog oscillators and filters

SynthLab-KS: physical modeling synth that uses the Karplus-Strong plucked string algorithm to generate realistic acoustic and electric guitar and bass sounds

SynthLab-DX: four-operator FM synth that produces the classic Yamaha DX synth sounds

SynthLab-WS: wave sequencing synth based on the Korg Wavestate's ® multi-lane, independent looping wave sequencer that allows use of both normal and morphing wavetable oscillators during the sequencing operation; the wave sequencing oscillator demonstrates how to create an amalgam of modules encapsulated in a single object

In addition, each of these has a precompiled dynamic module (*DM*) version which you can download from www.willpirkle.com/synthlab using the DM subscript/prefix: for example, SynthLab-WS_{DM} is the dynamic module version of the wave sequencing synth, and the dynamic module synths are collectively referred as SynthLab-DM.

1.2.1 SynthLab Documentation

SynthLab is fully documented with the Doxygen® tool and is available to download at www.willpirkle.com/synthlab-docs/. This includes every C++ object, interface, structure, and function, with every member variable and member function. You should bookmark this page and refer to it often. Synth projects are considerably complex, and there is not enough room in the book to document every object.

> The C++ listings in this book represent the interesting, difficult, or highly synth-oriented code that connects to the theory portions of each chapter. You will need to use the documentation and review the sample project code to get the most out of this book and understand how to select and use whichever C++ objects you like in your own plugin projects.

1.3 Synth Components

Most hardware and software synthesizers are designed from the same set of basic building blocks that will be used throughout this book. Table 1.1 lists these components, their abbreviations, and their descriptions, as applied to the SynthLab projects.

Table 1.1 Synthesizer components, my abbreviations, and descriptions, as used in *SynthLab*

Component	*Abbreviation*	*Description*
Synth engine	Engine	The entire synth, in one object that manages a set of voices
Synth voice	Voice	This term is not always used the same way, but for SynthLab, it is the object that renders each note-event; a voice contains a collection of one or more of the components below (LFO, EG, etc.)
Low frequency oscillator	LFO	An oscillator with frequency f_o on the range of about 0.02 to 20.0 Hz, though numerous variations on these limits are allowable
Envelope generator	EG	Produces a unipolar control signal used to modulate other components; it is most closely associated with the output amplifier, where it sets the time-domain contour of the rendered signal
Digitally controlled amp	DCA	A variable gain and panning amplifier used on the output of each voice
Virtual analog filter	VAFilter	A synth filter designed using digital integrator replacement in analog block diagrams
Biquad filter	BQFilter	A synth filter designed using the bilinear z-transform (see my FX plugin book)
Wavetable oscillator	WTOsc	A pitched oscillator that reads data from a table or array

(*Continued*)

Component	*Abbreviation*	*Description*
FM operator	FMOp	A combination of a sinusoidal wavetable oscillator and an EG that controls its output amplitude; FM operators are combined to modulate each other
PCM oscillator	PCMOsc	A pitched oscillator that reads pre-recorded PCM data from a table, array, or audio file
Virtual analog oscillator	VAOsc	An algorithmic oscillator that simulates the output of analog oscillators
Karplus strong oscillator	KSOsc	Contains an exciter and one or more resonators to simulate the behavior of plucked strings; may be modified to produce bowed string sounds – technically not an oscillator but behaves like one in the synth implementation
Wave sequence oscillator	WSOsc	An amalgam of four inner oscillators that generates the wave sequences from the Korg Wavestation® and Wavestate® synths; works in conjunction with the *WaveSequencer* object
Modulation matrix	ModMatrix	A component that connects modulation sources (LFOs and EGs) to modulation destinations, such as filter cutoff frequency or oscillator amplitude
Wave sequencer	WS	A modulation source that generates a sequence of modulation values used to create both melodic and evolutionary sounds; may be extended to use as a traditional step sequencer – this is based heavily on the Korg Wavestate's wave sequencer

1.4 Basic Software Synth Architecture

In the early 1990s, the MIDI Manufacturers Association (MMA) produced a specification for a software synth called the Down Loadable Sound (DLS) synth. You can get a copy of the Level I (L1) specification from www.midi.org/specifications-old/item/dls-level-1-specification. This file and the DLS Level II version are excellent sources as they define the synth architectures, GUI controls and their limits, and the transform functions that are used to manipulate GUI and MIDI control signals. I use some of these transforms and functions in SynthLab, so this is an excellent and perhaps mandatory resource for you to acquire. Figure 1.1 shows how the MMA diagrams a software synth. The LFOs and EGs are articulators that control parameters of the audio engine, and the control logic ties everything together.

The block diagram in Figure 1.1 will render a single note, so a polyphonic synth would include multiple sets of these blocks, one for each note that is rendered. In SynthLab, Figure 1.1 is called a "synth voice" block diagram, and the "synth engine" is a set of these voice blocks. This is detailed in Chapters 2 and 3, but you need to know the lingo now.

1.5 SynthLab Voice Architecture

All SynthLab projects use the same basic voice architecture, shown in Figure 1.2. Each synth has a different quad oscillator bank, and SynthLab-WS includes the additional wave sequencer object, but the projects are otherwise the same. Take note of the names of the components – *ampEG*, *filter1*, *wtOsc3*, etc. – as these are used throughout the text; being C++ member variable names, they will be italicized. All SynthLab synth projects include the following in their voice architectures:

- Four pitched oscillators – these are named according to the oscillator type, e.g. *wtOsc1* (wavetable oscillator 1) or *fmOp1* (FM operator 1)

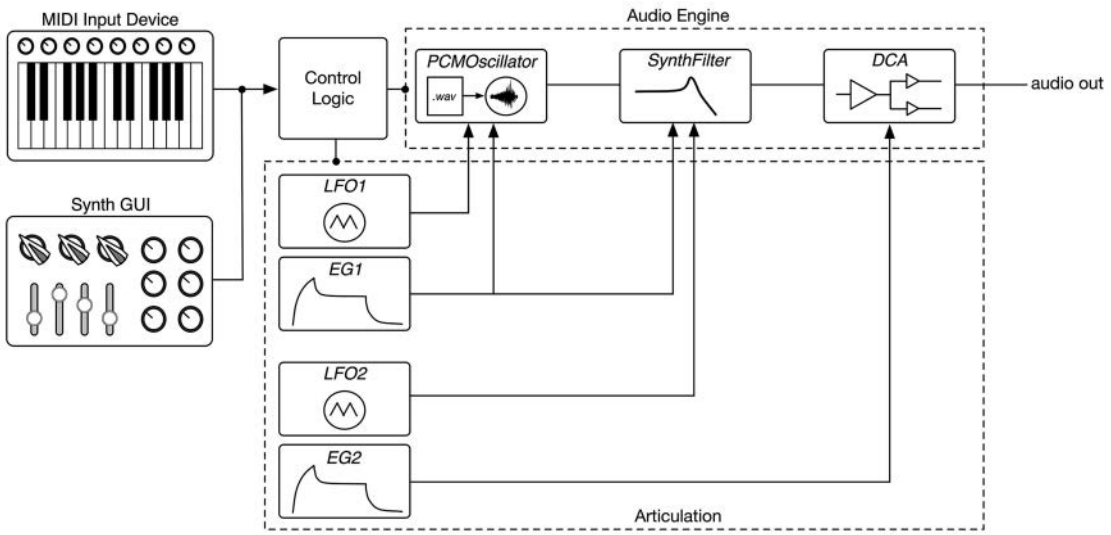


Figure 1.1 The MMA representation of a software synth includes control logic, articulation, and an audio engine that renders the synthesized signal

- Two LFOs (*lfo1* and *lfo2*)
- Three EGs (*ampEG*, *filterEG*, and *auxEG*)
- Two virtual analog filters that may be placed in series or parallel (*filter1* and *filter2*)
- One modulation matrix with multiple sources and destinations (*modMatrix*)
- One output DCA with EG control and panning function (*dca*)
- One ping-pong delay FX on the output
- One wave sequencer object (SynthLab-WS only)

Note that SynthLab's audio path is fully stereo – all oscillators feature stereo outputs and the filters, DCA and delay FX process stereo signals. Each note-event is rendered through its own voice object, so for a polyphony of 16 notes, we need 16 voices.

1.6 SynthLab C++ Implementation

Every SynthLab component is a C++ object or structure, from the top-level *SynthEngine* object down to the *SynthClock* – a tiny modulo counter that is the time-base for most oscillators. Figure 1.3(a) shows the C++ object hierarchy – take some time to study this diagram since I use the same paradigms throughout the text.

- Rounded corner blocks are C++ objects or structures
- Many objects contain an interface pointer to another object, shown with the familiar interface circle-bar on the left
- Square boxes contain string lists, or non-C++ objects
- C++ objects, structures, and their member variables and methods are always *italicized*

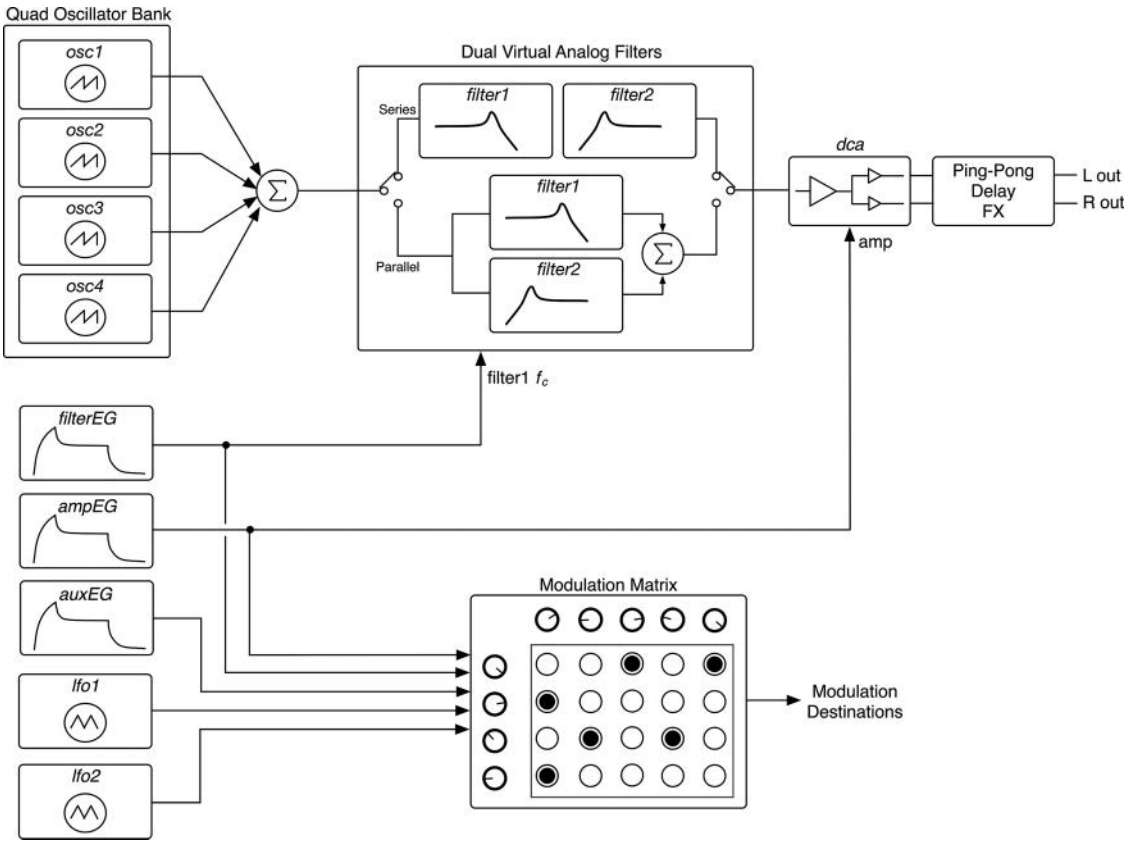


Figure 1.2 SynthLab voice architecture includes two filters that may be processed in series or parallel, or individually bypassed; the SynthLab-WS project also includes a wave sequencer object

Figure 1.3(a) shows how each sub-component is made of smaller components, and some of them reveal string lists that wind up on the GUI.

- The *SynthEngine* contains a set of 16 *SynthVoices*, one for each note of polyphony
- Each *SynthVoice* (voice 0 is shown here) contains a set of *SynthModules* that encapsulate the functionality of each row in Table 1.1
- All *SynthModules* except the wave sequencer, mod matrix, and DCA include four *ModuleCore* objects – I will always provide at least one core per module, and some include bonus cores to boot
- Each *SynthModule* exposes the names of its module cores
- Each *ModuleCore* object implements a variation on the module's theme and exposes a list of up to 16 strings for selecting core-specific parameters; for oscillators, these are waveform string names, and for filters, they are filter types, etc.

The module-plus-core paradigm is not only central to the way in which SynthLab is able to generate professional results, exposing banks of waveforms or sets of filters, but also part of the way in which it helps teach how each of the modules – which correspond to chapters in the book – operate within the synth structure. It also allows you to focus on only one part of the project at a time, and you are free to add your own modules so you can really personalize the synth projects without having to build new synths from scratch. In addition, the cores demonstrate variations on the module functionality. Examine the module cores in Figure 1.3:

- ***ClassicWTCore***: wavetables of classic sounds, like analog and FM waveforms, as well as some tables taken from slicing instrument samples
- ***MorphWTCore***: wave morphing oscillator that uses banks of wavetables to crossfade-morph
- ***DrumWTCore***: one-shot wavetables of classic electronic drums, one drum sound per voice
- ***FourierWTCore***: demonstrates dynamic wavetable creation using Fourier synthesis

There is a bonus *SFXWTCore* that implements one-shot sound effects that are useful in the wave sequencing synth.

1.6.1 Core, Module, and Mod Knob Strings

To maximize flexibility, SynthLab uses a dynamic GUI interface that allows loading string lists and GUI labels on-the-fly. The sizes of the lists are fixed to allow proper handling of automation and DAW state save and restore operations. You may also decide to implement a static GUI and not use the dynamic core paradigm – that is up to you and is covered as an option in Chapter 5. Figure 1.3(b) shows a typical GUI implementation for *wtOsc1*, the first of four wavetable oscillators (C++ object named *WTOscillator*) in the *SynthVoice* object. On the right side, there are four "mod knob" controls named A, B, C, and D which are specific to each module core. Most cores have at least one unassigned mod knob for you to experiment with. Examine Figure 1.3(a) and (b), and notice how the GUI controls connect to the module and its cores:

1 The GUI exposes the module core names that the *WTOscillator* provides in a list for the user – Classic WT, etc.
2 When the user selects a core, the module strings are dynamically loaded into the next control, named "Waveform" here (or "Filter Type" for the filters, "EG Contour" for the EGs, etc.)
3 In addition, the mod knob labels (A, B, C, and D) are re-named for that particular core to show the functionality; un-assigned knobs show only the alphabetical letter
4 Each object includes four hard-wired controls that are specific to that module: for example, in the oscillator object, these are tuning, output, and pan controls, while for the EG object, these are attack, decay, sustain, and release

You will see that almost all of the synth modules follow this paradigm and include exactly ten GUI controls per module; the exceptions are the sequencer, mod matrix, and DCA, which are either too complex to shoehorn into this format or too simple to require multiple cores and GUI controls.

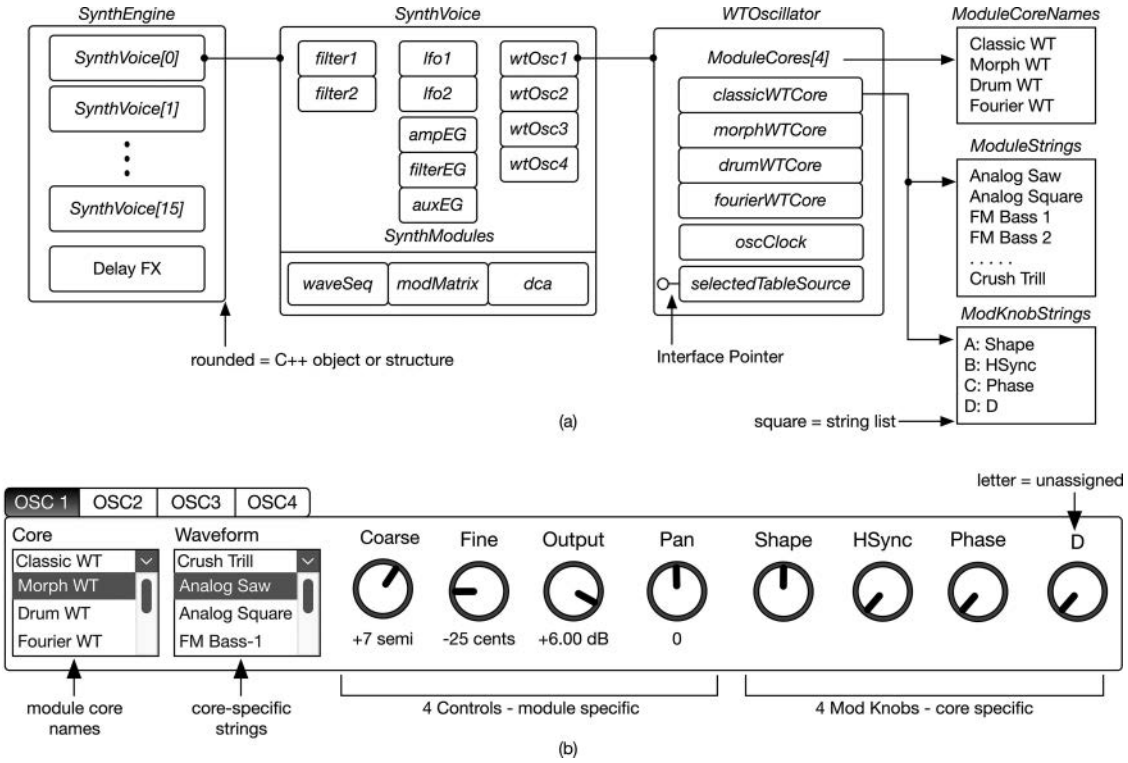


Figure 1.3 (a) SynthLab C++ objects include the engine, voice, modules, cores, and their associated string lists; (b) typical GUI layout for the wavetable oscillator

1.7 Why SynthLab Uses This Architecture

You may be wondering why SynthLab has this module-core paradigm with the dynamic strings and mod knobs. The short answer is that it is a result of both reader comments on the first edition of this book and, more importantly, years of teaching at the university level, where the class must be brought up to speed in only one semester. It is not uncommon for grad students from other universities to contact me for advice about very specific synth components – such as the virtual analog filters or FM operators they are researching for their thesis work. SynthLab's modules and cores make it easy to focus on just one concept and study that in depth. In addition, you can personalize your own components. For example, suppose you want to create your own morphing oscillator and make your own wavetable banks. If you only want to change the waveform banks, you can simply modify my object to supply the bank names. If you want to do the chapter homework problems or invent your own variation on my objects to add some interesting functionality that you want to investigate, you can subclass the *ModuleCore* base class; implement the object, name the core, and choose the names of the morphing banks that the user sees; and implement the functionality of the free-agent mod knobs and your own added ideas.

In addition, I usually break my classes into groups and have them each work on their own variation on a specific module. I use a three-step approach to facilitate their understanding. The first two steps are usually done as homework or a "pre-lab," while the third is for a larger project. The goal of the project is to implement something new and interesting, so the students must brainstorm and research to think of new ideas.

1 First, I assign homework and projects that only involve altering my code that extends the object in a simple but interesting manner.
2 Next, I assign a variation-object (which changes some specific behavior or functionality) or a combination-object (which amalgamates two or more modules in an interesting or intelligent manner) so they must subclass the base class object and work on implementing either their own variations or completely new functionality. This is when the students really begin to understand how the theory connects to the code as they are forced to go in and dissect and debug the existing code.
3 Finally, I assign an open-ended project that involves adding new objects or orchestrating big-picture parts of the synth that require brand new types of components (note sequencers, weird oscillators, and bizarre filters are some student favorites).

With SynthLab, you can start with the first wavetable synth project and design its GUI, which takes time because there are many components and controls, then debug and implement the synth. To move to the next project, you only need to change the oscillator-rendering core as everything else is the same. This means you can re-use the same project and GUI over and over, and if you implement the dynamic string loading and core selection, you only need to change a few lines of code to generate what appears to be a completely different synth. The only project that would require more GUI work is the wave-sequencing synth, which exposes the sequencer controls but also hides the underlying oscillator controls, so you can make it fit into the same GUI real-estate area as the rest.

1.8 SynthLab Object Operational Phases

In Chapter 3, you will see that all synth modules implement the same set of five functions plus the constructor; these perform the majority of the synth operations. When these objects appear in future chapters, I use the same paradigm for describing the C++ coding details based on the functions that correspond to the six operational phases of the modules and are named in the same way.

1.8.1 Construction Phase

Each object's constructor performs specific construction related to that module. The *SynthEngine* is instantiated first and creates its set of *SynthVoice* members, along with the delay FX object. The *SynthVoice* object creates its set of *SynthModules* and wires the modulation matrix. At the *SynthModule* level, the constructor creates the audio buffers used to shuttle data between objects and creates and initializes any other member objects. It also sets up the processing structure used to communicate with its set of cores. At the *ModuleCore* level, the constructor is mainly involved in exposing the set of 16 core strings and text labels for the mod knobs.

1.8.2 Reset Phase

The reset phase usually only happens once after the DAW loads the plugin, but it will also happen if the user changes the DAW sample rate. For synth plugins, the sample rate is usually established only once and not changed again; however some DAWs, such as Reaper®, may implement the reset operation many times prior to streaming audio. Therefore, each module implements a function named *reset* whose only argument is the current sample rate. The details of the reset phase and function are dependent on the object: for example, the wavetable oscillators use the reset phase to query the wavetable database to make sure the necessary tables are available.

1.8.3 Note-On Phase

Each module and core implements a *doNoteOn* function that accepts the MIDI data for the particular note-event. Most modules perform some meaningful operations during note-on: for example, the oscillators reset their internal clock objects and store the MIDI pitch of the incoming note. The note-on phase must fully prepare the object for its operation of rendering or processing audio data.

1.8.4 Update Phase

The update phase is one of the most important and can be the source of CPU bottlenecks. Each module will be updated once per audio block processing cycle, as detailed in Section 2.4, and implements the *update* function to perform these operations. The update phase consists of two parts: updating the component state due to changes made on the GUI and updating the rendering variables based on modulation from other components. Usually, the equations are combined for both of these parts. For example, an oscillator's state may change because the user adjusted the coarse tuning control, which affects the pitch. At the same time, an LFO connected to the oscillator's modulation input may also alter the oscillator pitch. The values of both of these are combined to generate a single pitch shift for that audio block processing cycle. The details of pitch modulation, which is key to almost every synth module in some fashion, are discussed in Section 6.8.1.

1.8.5 Render Phase

There are two fundamental types of synth components: those that render information (audio or control signals) and those that process information (usually audio but may also apply to control signals in some cases). All synth modules implement a *render* function that performs each module's functionality; clearly, there will be numerous differences based on the kind of module. Modulation objects render their output into special arrays called modulation inputs and outputs. Pitched oscillators render audio into audio-only buffers. Filters and DCAs process audio input buffers and write to audio output buffers; all of these are detailed in their corresponding chapters.

1.8.6 Note-Off Phase

Interestingly, most modules do nothing during the note-off operation, with the exception of the envelope generators, which play a critical role in the note-event lifecycle, as described in Section 4.1. If nothing occurs during the note-off phase, then it is not documented in the chapter.

1.8.7 Why No Destructor Phase?

Almost all SynthLab objects and their members are instantiated dynamically and use the *std::unique_ptr* or *std::shared_ptr* for maintenance, freeing us from worrying about calling the *delete* operator and keeping track of reference counts and lifecycles. Use of these auto-deleting pointers goes all the way down to some of the lowest-level objects. But still, there are a handful of exceptions, such as dynamic wavetable creation that is sample-rate dependent, wherein the destructor has meaningful code. In many of these cases, you will likely implement your own object to handle these operations in a way that makes sense to you, such as creating tables or reading files, so in general, that will be your issue to deal with.

Bibliography

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 1.* https://www.midi.org/specifications-old/item/dls-level-1-specification, Accessed October 14, 2020

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 2*. https://www.midi.org/specifications-old/item/dls-level-2-specification, Accessed October 14, 2020

Roads, Curtis. 1996. *The Computer Music Tutorial*, Chap. 2. Cambridge: The MIT Press.

2 The Synth Engine

The synth engine implements the entire synth architecture, and *SynthEngine* is the single C++ object you need to add to your plugin framework's processing object. In the example plugins designed with ASPiK, this is the *PluginCore* C++ object; for JUCE, it is the *AudioProcessor* object. Each of the synth projects is packaged in a single *SynthEngine* object, which is implemented in *synthengine.h* and *sythengine.cpp*, and located in a directory with the other supporting object files. These objects are framework agnostic, meaning they are pure C++, have no bindings to any plugin framework, and do not require additional libraries beyond the built-in standard template libraries included in your compiler. In Chapter 5, you will learn how to wire up the engine to render audio and how to send GUI parameter information to it. This chapter focuses on the synth engine's duties and overall architecture.

2.1 Engine Behavior

The engine performs three tasks during the synth's operation: (1) initialization, (2) applying GUI control changes, and (3) rendering the synthesized audio. I designed the *SynthEngine* object to expose simple functions that service these operational functions, with an interface that is not dependent on any plugin framework. Your plugin framework's processing object will create and use the engine object – it is the sole C++ object and interface that you need to wire into your plugin. After instantiation, the plugin will call five methods on the engine for the three engine functions, as seen from the plugin/DAW side:

1 **Initialization**: the plugin calls the engine's *reset* and *initialize* functions
2 **Set GUI parameters**: the plugin calls *getParameters* for a shared pointer to the engine's GUI connected parameters and *setParameters* to instruct the engine to update its states, causing a trickle-down transfer of parameters using shared pointers and without data copying
3 **Rendering audio**: the plugin calls the engine's *render* function, passing it audio buffers to fill one a block-by-block basis

2.2 Engine Architecture

The *SynthEngine* objects are identical for all of the SynthLab. Each *SynthEngine* owns a set of *SynthVoice* objects, each of which renders a single note-event for that particular synth flavor, so there is a different *SynthVoice* for the FM synth, the virtual analog synth, etc. For a synth with a polyphony of eight notes, you will need eight *SynthVoice* objects. You can create a multi-timbral

synth by mixing *SynthVoice* objects of different types. The engine mixes the audio outputs of the active voice objects as it renders the final synth output for your framework's processor object to handle. The *SynthEngine* objects use a statically declared array of *std::unique_ptrs*, one for each voice. The unique pointer object deletes the underlying resource automatically when no other object references it and doesn't require memory maintenance or the *delete* operation.

```
std::unique_ptr<SynthVoice> synthVoices[MAX_VOICES];
```

If you are an experienced programmer, you will want to modify the object in your own way: for example, you could put each voice on its own thread and include another mixer-thread to mix the outputs. Or you could dynamically control the number of voice instances based on CPU power. For this chapter, you don't need to think about the internal voice operation, only that you will be creating, initializing, and calling the voice rendering function from the *SynthEngine* instance.

Figure 2.1 shows the *SynthEngine* architecture in block diagram form. The GUI, MIDI input, and audio output connections are inside of dotted boxes – this is where your plugin framework processor object makes its interface connections. You can see that there are three connection points:

1 GUI information, sent to a shared voice parameter and the sole engine parameter data structures
2 Incoming MIDI information for CCs and other common messages, collectively called "global MIDI data"
3 Audio output rendering, delivering data from the engine to the processor object

Figure 2.2 shows the big-picture block diagram of objects and function calls. The circle-bar on the left edge of each object represents an external function call or interface. Most objects also include a few functions that are only used within that object. In this chapter, we focus on the *SynthEngine*, its exposed functions, and their operations, shown in the bold box in Figure 2.2. In Chapter 5, you will see details of the framework side. In the next chapter, we look at the voice, module, and core objects.

2.3 Shared Data

Sharing data across C++ objects is a necessity for a software synth. Even a complex FX plugin still only has one main target for the GUI controls, but in a polyphonic synth, each note-rendering object (*SynthVoice*) shares the same voice GUI controls with all the others – the filter f_c, oscillator detuning, and EG settings are applied identically to each voice object. In addition, SynthLab modules are designed so that each component has full access to all incoming MIDI data, allowing you to tailor your objects specifically for custom MIDI message processing – for example, MIDI Polyphonic Expression (MPE). The wavetable and PCM sample data is generally large and likewise must be shared across the voices to prevent redundancy and memory consumption.

2.3.1 The Singleton

If you don't know what a singleton is, some programmers will tell you that you are better off not knowing. The singleton is a source of heated debate among programmers. You were probably

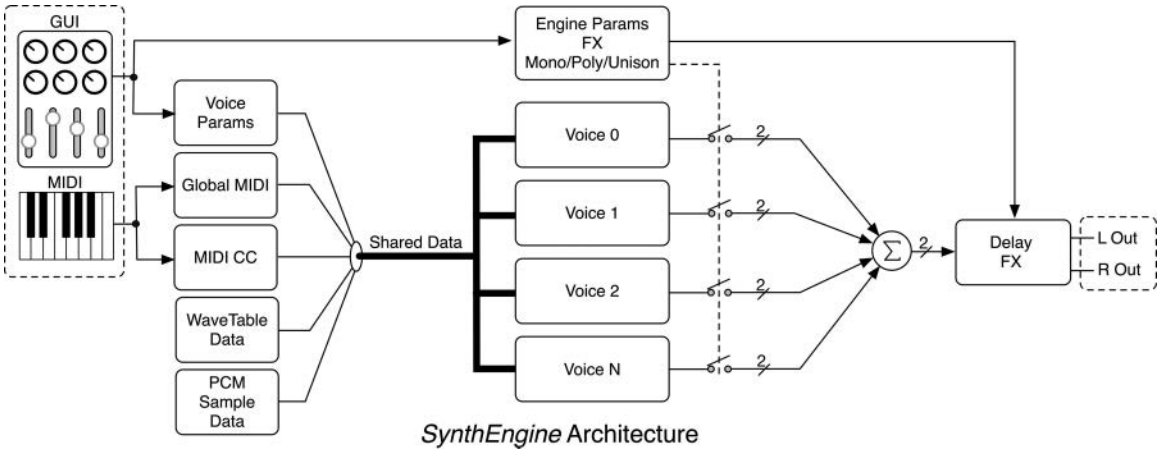


Figure 2.1 The *SynthEngine* object architecture; the dotted boxes show the interface points to your plugin framework's processor object

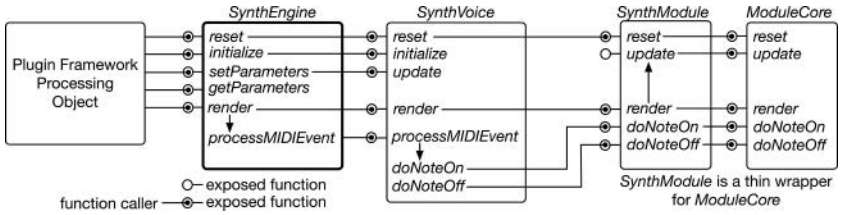


Figure 2.2 Hierarchy of SynthLab objects responding to function calls from the host plugin framework processor object

taught that declaring global variables is a bad design choice; these exist through the lifetime of the process in which they are created, even if they are created in a DLL, loaded into that process address space, and later deleted. Essentially, a singleton is a global variable that points to a data structure or C++ object which often acts as a database, using get/set functions to store data that persists for the lifetime of the process. Singletons are delightfully simple to implement and use. They are also disastrous in some cases, especially where class factories are involved. Having been burned with singletons in the past, I avoid them unless there is no other option. This book will not use singletons in any manner. That said, if you poke around the AAX, AU, and VST API base classes and lower-level code, you might be interested in what you find – while eschewed in the academic world, global variables and singletons are known for appearing in real-world applications.

2.3.2 The Shared Pointer

The standard library's *std::shared_ptr* object is designed to allow the safe sharing of resources with lifecycle control, which deletes the underlying resource automatically when no other object references it. Objects that use the shared pointer cannot harm the underlying data. For the SynthLab

projects, I am using *std::shared_ptrs* to share engine and voice data structures across objects as well as to share MIDI, wavetable, and PCM sample data. As thin wrappers, the shared pointers require a function call to use the underlying "naked" pointer, but that bit of overhead is outweighed by the benefits of safe data sharing and automatic lifecycle control. Within the SynthLab objects, there are a handful of instances in which old-fashioned (naked) pointers are used; generally, this involves access to floating point arrays that are audio input and output buffers, dynamically created wavetables, and ultimately connected to the plugin framework.

2.3.2.1 Shared GUI Parameters

The SynthLab architecture consists of one engine that owns a set of *SynthVoice* objects. Each voice owns a set of *SynthModule* objects that encapsulate the synth building blocks. Each module includes a custom data structure that is used to pass GUI control data into the object. Consider the array of four synth voice objects stacked together in Figure 2.3(a), in which each voice consists of the following *SynthModules*: LFO, EG, filter, wavetable oscillator, and DCA. Figure 2.3(c) shows that the voice's custom parameter structure, *SynthVoiceParameters*, contains a set of module parameter structures that are maintained with *std::shared_ptrs*, one for each module, and shown in dotted boxes.

This means that the *LFOParameter* structure and its contents for the voice's *lfo1* member object are shared across all *lfo1* members and, thus, the stack of objects connected to the parameter structure box. Each of the module's custom structures are shared across the same objects and across all voices. In Figure 2.3(c), the engine object owns a single instance of a *SynthVoiceParameter* structure, also implemented with a shared pointer that is likewise shared across all voices. In order to force programmers to maintain this important architecture, all voice and module constructors require shared pointers to their specific data structures. Note that this is not required when using the objects in stand-alone mode as they will synthesize their own parameter structures.

The *SynthEngine* contains a shared parameter member, created during instantiation, that is used for updating from the GUI. It is a shared pointer because the plugin framework's processor object will need to obtain it to update the GUI control information.

```
std::shared_ptr<SynthEngineParameters> parameters =
                        std::make_shared<SynthEngineParameters>();
```

Within the *SynthEngineParameters* definition, you will find the shared pointer of voice parameters:

```
std::shared_ptr<SynthVoiceParameters> voiceParameters =
                              std::make_shared<SynthVoiceParameters>();
```

When the synth engine creates the voice objects, it passes the identical shared pointer to each one during construction.

2.3.2.2 Engine-Specific GUI Parameters

In addition to the shared voice parameters, the engine parameters also include some top-level controls, which apply at the engine level only. The most important parameter is a strongly typed *enum* that sets the synths mode: monophonic (one note at a time), unison (one note at a time, but the note is made of layered voices) and polyphonic (multiple notes can be played simultaneously). Operational modes are covered in Chapter 4.

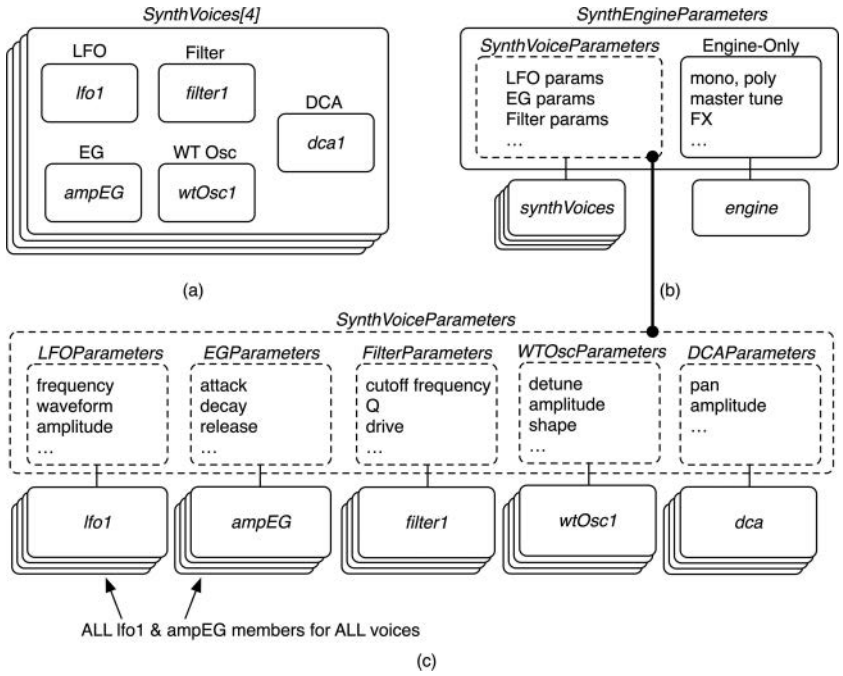


Figure 2.3 (a) The engine owns a stack of voice objects (four in this example but 16 in the SynthLab projects), each of which includes an identical set of LFO, filter, EG, oscillator, and DCA objects; (b) the engine owns a voice parameter structure that is shared across these voices; and (c) the shared voice parameters structure consists of a set of shared module parameter structures

```
enum class synthMode { kMono, kLegato, kUnison, kUnisonLegato, kPoly};
```

The other members include global controls for volume, pitch bend sensitivity, and overall instrument tuning. These are mapped to their corresponding MIDI data values, and use names that define their functionality.

2.3.2.3 Shared MIDI Input Data

The engine receives and decodes MIDI messages from your plugin framework's processor object, as described in Section 5.4.3. All decoded messages are then stored in two arrays of *uint32_t* data types, one for the MIDI continuous controller (CC) data and the other for a selected set of messages relevant to the SynthLab projects. The two sets of messages are called MIDI CC and Global MIDI data, respectively. A third array of DAW data, comprising BPM and time signature information, is included, though it is not specifically MIDI and is called "aux DAW data." This is likewise shared with the modules. A simple C++ class named *MIDIInputData* manages the three arrays and allows for read-only access to incoming data for all synth modules, simplifying MIDI modulation calculations and allowing for MPE.

This class is derived from *IMIDIInputData* that allows module core objects to access the data. The *SynthEngine* contains a single member, created during instantiation, that is used for sharing the MIDI input data:

```
std::shared_ptr<MidiInputData> midiInputData =
                                        std::make_shared<MidiInputData>();
```

When the synth engine creates the voice objects, it passes each one the identical shared pointer during construction.

The engine is set up to allow for the generation of MIDI output data for MIDI effects, but this is not implemented for the SynthLab projects as it is unnecessary for their operation. On occasion, you will see *midiOutputData* variables in the code, but they are only there if you want to get into MIDI effect programming, which is outside the scope of the synth signal processing presented here.

2.3.2.4 Shared Wavetable and PCM Sample Data

The wavetable and PCM sample data is shared across the voices, as are their modules using shared pointers. The *WavetableDatabase* and *PCMSampleDatabase* objects maintain the data for the wavetable and PCM sample oscillators. You will notice that the engine, voice, and MIDI input data shared pointers are all created with *std::make_shared* right in the *synthengine.h* file and are generated during instantiation but before the constructor function is executed. The engine is the original creator and owner of this data. To provide maximum flexibility in the modules and their cores, the wavetable and PCM sample databases are created empty, lacking content of any kind. The wavetable and PCM sample oscillators bring their own data, either creating it at construction time or reading from compiled data arrays or *.wav* files included with the plugin. The module chapters contain details about the databases and their interfaces.

As the wavetable and PCM sample oscillators are instantiated, each will query the corresponding database to see if the required tables or sample arrays exist. If not, the oscillator object will create them and add them to the database. This means that the very first wavetable oscillator of a given core type will create and register its tables with the database. All subsequent wavetable oscillator cores will find the data and will not need to create it. The same paradigm holds true for PCM sample oscillators that use *.wav* files to package their multi-sample sets such that the files are only accessed once. The engine's members are therefore declared with *nullptr*:

```
// --- shared tables
std::shared_ptr<WavetableDatabase> wavetableDatabase = nullptr;

// --- shared samples
std::shared_ptr<PCMSampleDatabase> sampleDatabase = nullptr;
```

Then, in the engine's constructor, the empty databases are generated and passed to the voice objects during construction, which, in turn, construct their modules passing in the shared database pointers. You can see how the shared pointers trickle down from engine to voices, and then from voices to modules, each during construction.

2.4 *SynthEngine* Constructor

We can put all of this information together by examining the *SynthEngine's* constructor which does the following set of tasks:

1 Initialize some of the MIDI global and CC data to values that will prevent the synth from starting up without audio: for example, MIDI CC7 (volume) is set to the maximum value 127, while MIDI CC10 (pan) is set to the center value 64; the code below is a partial listing as there are numerous other defaults to make

```
// --- volume/pan prevent accidental silence, pan
midiInputData->setCCMIDIData(VOLUME_CC07, 127); // --- MIDI VOLUME max
midiInputData->setCCMIDIData(PAN_CC10, 64); // --- MIDI PAN center
```

2 Create the wavetable and PCM sample database shared pointers

```
// --- these databases are empty at this point
if (!wavetableDatabase)
     wavetableDatabase = std::make_shared<WavetableDatabase>();

if (!sampleDatabase)
     sampleDatabase = std::make_shared<PCMSampleDatabase>();
```

3 Create the set of voice objects that will render audio, passing MIDI; notice the unused *midiOutputData* (also a shared pointer) and the *blockSize* parameter which we will investigate next

```
// --- create the voices
for (unsigned int i = 0; i < MAX_VOICES; i++)
{
     // --- reset is the constructor for this kind of smartpointer
     //
     synthVoices[i].reset(new SynthVoice(midiInputData,
                                         midiOutputData, /* unused */
                                         parameters->voiceParameters,
                                         wavetableDatabase,
                                         sampleDatabase,
                                         blockSize));
}
```

The last line of code sets up a special structure for rendering from the voice and is discussed in Section 2.4.4.

```
// --- voice render
voiceProcessInfo.init(NUM_INPUT_CHANNELS,
                      NUM_OUTPUT_CHANNELS,
                      blockSize);
```

2.5 Audio and MIDI Block Rendering

The engine is also responsible for accepting incoming MIDI data and rendering the final output signal. All plugin frameworks, like ASPiK, JUCE, and iPlug2, ultimately process and render audio

in blocks named "audio I/O buffers" here. This is covered in detail in the second edition FX book. In addition, the frameworks gather incoming MIDI messages that are queued and time-stamped to align in a sample-accurate manner with the audio data. This package of information in applied to the processor object's audio processing or rendering function. For some synth APIs, like VCV Rack, the processing must be done on a sample-by-sample basis in which processing occurs on each sample interval. If you are rendering stereo audio, there are two samples to render on each sample interval.

In SynthLab, audio is rendered in blocks with a preferred size of 64 stereo samples at a time; this is the designated default *blockSize* variable you see in the *SynthVoice* construction in Section 2.2.2.4. A stereo sample means one sample each for the left and right channels, and the *blockSize* value indicates the number of stereo samples. When you instantiate the engine in your framework, you send it the default block size; you thus have the ability to change this value, but there is a reason for the preferred size of 64 samples/channel. All plugin frameworks deliver audio samples in independent left and right data buffers, and SynthLab expects you to render into individual buffers as well. The SynthLab projects are set up with the ability to accept incoming audio data that you may process through filters or use as a side-chain, vocoder, or other control generator. The stock SynthLab projects only render audio, so we will only be discussing the audio output buffers.

Incoming MIDI data is also applied in the same block fashion, with all MIDI messages sent to the engine via a *std::vector* of *midiEvent* objects. However, as with many commercially available synth products, the MIDI messages are serviced all at once, before the audio block is rendered. This means that they are not sample-accurate and are part of a SynthLab scheme to granulize MIDI data and internal modulator signals over blocks of data that generate a massive increase in efficiency and a lowered CPU usage. The tradeoff in efficiency is well worth the loss of perfect sample accuracy between MIDI and audio, and follows a law of diminishing returns – as the block size becomes larger, the savings flatten out.

The amount of MIDI "slop" in the system is a function of the block size since all messages will be processed first, placing a maximum error in MIDI event timing that is equal to the block size multiplied by the sample interval. At 44.1 kHz, 64 samples of MIDI slop will equate to a worst-case time offset error of 1.4 mSec, and the performer will not notice this slight latency in playing keys or using MIDI controllers. You may also up the block size to 128 samples and experiment with your own tactile limits. However, if you demand perfect sample accuracy, set the *blockSize* variable to one (1) during engine construction; just be prepared to pay the price in CPU usage.

2.5.1 The AudioBuffer *Object*

The *AudioBuffer* object plays several roles in SynthLab and is used not only for moving audio data between your processor object and the engine but also within the synth modules themselves as the voice renders them, in turn. The *AudioBuffer* object is designed with plugin frameworks and native APIs in mind; the input and output buffers are independent per channel and passed *float*** parameters that are compatible with all plugin APIs and frameworks that process channel buffers (i.e. everything but VCV Rack). In addition, it includes members that store channel counts, maximum block size, and current block size if a partial block needs rendering, which may happen. Details for preparing the buffers can be found in Chapter 5.

2.5.2 The midiEvent *Structure*

The *midiEvent* structure contains the raw MIDI data that has been converted from BYTE to *uint32_t* in your framework. The sample-offset value is transmitted, but it is ignored during block processing.

```
uint32_t midiMessage = 0;       //< BYTE message as uint32_t
uint32_t midiChannel = 0;       //< BYTE channel as uint32_t
uint32_t midiData1 = 0;         //< BYTE data 1 as uint32_t
uint32_t midiData2 = 0;         //< BYTE data 2 as uint32_t
uint32_t midiSampleOffset = 0;  //< sample offset
```

2.5.3 The SynthProcessInfo *Object*

Your plugin framework's processor object prepares a *SynthProcessInfo* object that is passed to the engine on each block processing cycle. The object inherits from *AudioBuffer* and therefore includes the protected members and methods mentioned in Section 2.3.1. The object includes a member vector of *midiEvents* to be decoded for the render operation. The engine's render function uses this information for MIDI and for rendering audio. The framework duties are discussed in Chapter 5.

2.6 *SynthEngine* Operational Phases

We discussed the engine constructor in Section 2.2.2.4. The *SynthEngine* does not have a base class but is set up with virtual functions and a virtual destructor so that you may use it as a base class, overriding its behavior with your own functions and allowing you maximum freedom in implementing your own ideas. Aside from a few functions dedicated to voice-stealing and polyphony, discussed in Chapter 4, there are only five functions to deal with, and two of them are trivial. Let's examine the guts of each method and discuss its role in engine functionality.

2.6.1 SynthEngine *Reset and Initialization*

The engine initialization phase is shown in Figure 2.4(a), and your plugin framework only needs to make five function calls to the engine for operation. The first two are *initialize* and *reset*. Your framework will call the engine's *reset* function whenever the sample rate is established or changed for the plugin. In many DAWs, this will be called only once. The *reset* function for the *SynthEngine* is trivial in that it simply forwards the reset call to its array of *SynthVoice* objects and FX module.

```
bool SynthEngine::reset(double _sampleRate)
{
     // --- reset array of voices
     for (unsigned int i = 0; i < MAX_VOICES; i++)
     {
          // --- reset them
          synthVoices[i]->reset(_sampleRate);
     }
     // --- FX
     pingPongDelay->reset(_sampleRate);
     return true;
}
```

The SynthLab-PCM project uses *.wav* files to load PCM sample data. For the precompiled and ASPiK versions, these files are located in a folder in the same directory as the plugin component or DLL. You have the freedom to move this location as you see fit. The *initialize* function is used to

forward the path to your DLL's container directory to the voice objects for use in parsing the PCM samples. The PCM database objects assume that the PCM sample folder is in a subfolder of this directory. Your plugin framework or your personal plugin project needs to know the folder name at instantiation time, so if you want to change this path, just make sure it is in a directory with an established location. The *initialize* function simply forwards this path to the member voices. A *const char** is used for maximum flexibility across frameworks and APIs, and to allow dynamic core loading across a thunk barrier. See the documentation for more information on the DLL path.

```
bool SynthEngine::initialize(const char* dllPath)
{
     // --- loop and init
     for (unsigned int i = 0; i < MAX_VOICES; i++)
     {
          // --- init
          synthVoices[i]->initialize(dllPath);
     }
     return true;
}
```

2.6.2 Setting SynthEngine *Parameters*

Your plugin framework will take GUI control information and send it to the *SynthEngine* via the *getParameters* and *setParameters* method that uses a shared pointer to shuttle parameters to the voice objects and engine-specific components. This is shown in Figure 6.4(b) and occurs at the top of the buffer processing cycle.

```
void setParameters(std::shared_ptr<SynthEngineParameters>& _parameters)
```

The framework side of the code is discussed in Chapter 5. The engine parameters allow the *SynthEngine* GUI to expose the following controls (this is your decision, and they may be omitted):

- Global volume control
- Global pitch bend sensitivity, coarse and fine adjustments
- Global tuning, coarse and fine adjustments (as with analog synths)

These parameters may be sent directly to the engine, voice, and oscillator objects, but I have set them up to be converted into the appropriate MIDI data values. The MIDI specification includes data variables for all of these items and is detailed at www.somascape.org/midi/tech/spec.html#usx7F0401 (note that the MIDI spec uses the term "master" where I use the term "global"). The MIDI data values are written to the *globalMIDIData* array and require massaging to be converted from 7-bit MSB/LSB into uint32_t values.

The *setParameters* function performs three operations for the engine:

1 Set up the global MIDI data with information from the global parameters for volume, pitch bend, and tuning; some code is straightforward, such as the pitch bend range (sensitivity), while others require work; remember that this is optional, and you may remove it if you like.

```
// --- pitch bend range in midi data table; for a released synth, you
// want to decode this as SYSEX as well
// --- sensitivity is in semitones (0 -> 127) and cents (0 -> 127)
uint32_t pbCoarse = parameters->globalPitchBendSensCoarse;
boundMIDIValueByte(pbCoarse);

unsigned int pbFine = parameters->globalPitchBendSensFine;
boundMIDIValueByte(pbFine);

midiInputData->setGlobalMIDIData(kMIDIMasterPBSensCoarse, pbCoarse);
midiInputData->setGlobalMIDIData(kMIDIMasterPBSensFine, pbFine);
```

2 Call the *update* function on the voices one at a time; this is only required for changing module cores, as described in Chapter 5, and requires little overhead; no calculations are performed.
3 If unison mode is chosen, this function also sets the unison voices' detuning and panning parameters; this is also discussed in Chapter 5.

2.6.3 SynthEngine *Render Operation*

The plugin framework's processor object will call the *render* function once per data block during its own audio processing function. The render operation consists of three phases:

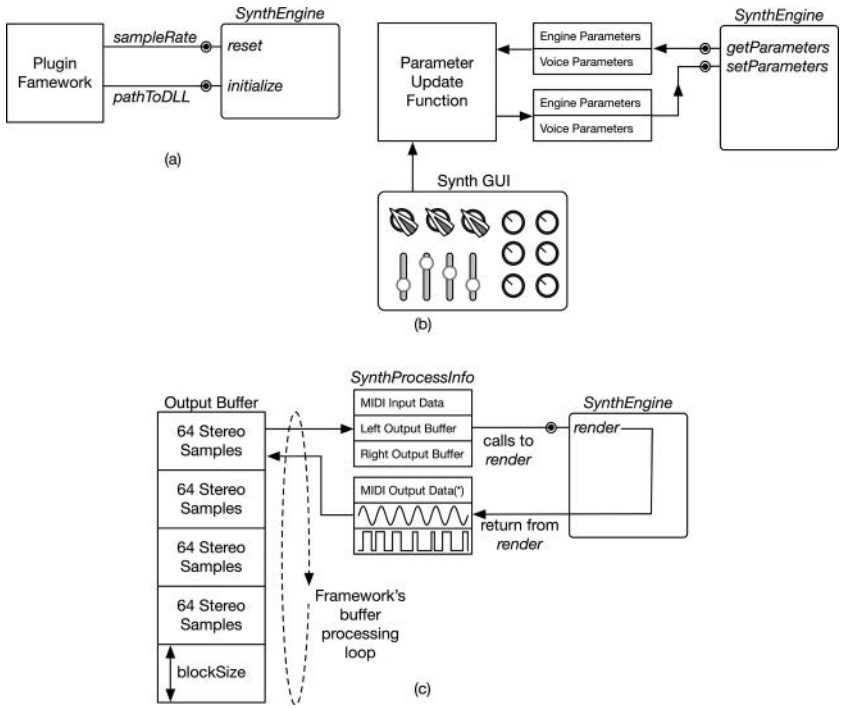


Figure 2.4 *SynthEngine* operation consists of three phases: (a) initialization, (b) GUI updates, and (c) audio rendering

1 Update the engine's GUI parameters with calls to *getParameters* and *setParameters* that use shared pointers for setting parameters, as shown in Figure 2.4(b)
2 Prepare a *SynthProcessInfo* structure with the incoming MIDI data and pointers to audio buffers for rendering
3 Enter a block processing loop that sends the *SynthProcessInfo* structure into the engine repeatedly, using MIDI data and audio buffer pointers for each *blockSize*; if the last buffer is smaller than *blockSize*, just call the render function, and set the current *blockSize* value, as shown in Figure 2.4(c)

This function processes the incoming MIDI using the *processMIDIEvent* function and then calls the voice render methods.

2.6.4 SynthEngine *MIDI Processing*

All MIDI messages are decoded at the top of the block processing cycle in the function *processMIDIEvent* that is called once for each incoming MIDI event. This function is split into three parts, as shown in Figure 2.5: one part decodes and stores all MIDI CC messages in the CC database, another stores global non-note messages, while the third handles MIDI note-on and note-off messages only. The engine is set up to respond to all MIDI channels, which you can change easily since this function is the primary MIDI receiver.

The top portion of the *processMIDIEvent* function that decodes note-on and -off events is key to the mono/unison/polyphonic operation and is covered in Chapter 4.

```
bool SynthEngine::processMIDIEvent(midiEvent& event)
{
     if (event.midiMessage == NOTE_ON)
     //
     // --- ALL NOTE ON/NOTE OFF MESSAGES
     //
     // --- Covered in Chapter 4
```

The second part deals with the MIDI CC messages. You can easily modify this function to add more data to the global MIDI data array. For the pitch bend message, the LSB and MSB are placed

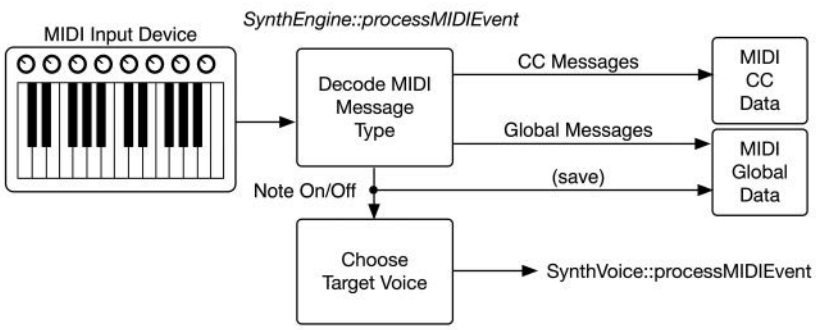


Figure 2.5 SynthEngine MIDI processing

into the *globalMIDIData* array using the *set* function call; for the CC messages, the data is simply stored as-is in *uint32_t* format.

```
// --- store the data in our arrays
if (event.midiMessage == PITCH_BEND)
{
     midiInputData->setGlobalMIDIData(kMIDIPitchBendDataLSB,
                                                    event.midiData1);
     midiInputData->setGlobalMIDIData(kMIDIPitchBendDataMSB,
                                                    event.midiData2);
}

if (event.midiMessage == CONTROL_CHANGE)
{
     // --- store CC event in globally shared array
     midiInputData->setCCMIDIData(event.midiData1, event.midiData2);
}
```

After that, you may add code to store any other MIDI input data you wish to include, such as tuning, pitch bend, and global volume/pan messages. These are implemented on the SynthLab GUI and stored as their MIDI counterpart data.

2.6.5 SynthEngine *Audio Synthesis*

The engine calls the voice's render function repeatedly, as shown in Figure 2.3(c). The *SynthVoice::render* function takes the same kind of argument as the engine – a *SynthProcessInfo* structure – but it is an engine member object and has been prepared in advance. The last line of code in the engine's constructor from Section 2.2.2.4 is:

```
// --- for voice objects
voiceProcessInfo.init(NUM_INPUT_CHANNELS,
                      NUM_OUTPUT_CHANNELS,
                      blockSize);
```

This function sets up the *AudioBuffer's* input and output arrays using the arguments as channel counts. The parameters NUM_INPUT_CHANNELS = 0 NUM_OUTPUT_CHANNELS = 1 set up no audio input buffer and two output buffers, one for each stereo channel. The *blockSize* parameter sets the maximum size of these buffers. Let's step through the render function:

```
bool SynthEngine::render(SynthProcessInfo& synthProcessInfo)
```

First, a call is made to flush the output buffers in the incoming *synthProcessInfo* object as they may contain old data; it is not uncommon for plugin frameworks and APIs to leave old data in these arrays.

```
// --- clear out
synthProcessInfo.flushBuffers();
```

Next, all of the MIDI events in the vector are serviced using the member function:

```
// --- issue MIDI events for this block
uint32_t midiEvents = synthProcessInfo.getMidiEventCount();
for (uint32_t i = 0; i < midiEvents; i++)
{
     // --- get the event
     midiEvent event = *synthProcessInfo.getMidiEvent(i);
     // --- process it
     processMIDIEvent(event);
}
```

Next, we set up for the render operation. When summing many voices (notes) together, the synth may overload the output, so this chunk of code sets a scaling factor for each voice depending on the mode; this is further discussed in Chapter 4.

```
double gainFactor = 1.0;
if (parameters->mode != synthMode::kUnison)
     gainFactor = 0.5;
```

Next, the *midiInputData* and *voiceProcessInfo* objects are refined with data for the current block; note the use of *getSamplesInBlock* to determine how many of the maximum *blockSize* samples need to be rendered. Also, notice how the DAW information is transferred into the MIDI *auxDAWData* array; this makes it instantly available to every component.

```
// --- important
voiceProcessInfo.setSamplesInBlock(
                              synthProcessInfo.getSamplesInBlock());

midiInputData->setAuxDAWData(kBPM, synthProcessInfo.BPM);
midiInputData->setAuxDAWData(kTSNumerator, etc…
midiInputData->setAuxDAWData(kTSDenominator, etc…
midiInputData->setAuxDAWData(kAbsBufferTime, etc…
```

Lastly, the engine loops over the voices calling the *render* function on each of the active voices. A voice state is set to *active* only if the voice is currently processing a note-event. A helper function is used to accumulate the voice output buffers together (the audio mixer operation). The final application of global volume and delay FX is covered in Section 16.4.

```
// --- loop through voices and render/accumulate them
for (unsigned int i = 0; i < MAX_VOICES; i++)
{
     // --- blend active voices
     if (synthVoices[i]->isVoiceActive())
     {
          // --- render and accumulate
          synthVoices[i]->render(voiceProcessInfo);
          accumulateVoice(synthProcessInfo, gainFactor);
     }
}
```

Bibliography

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 1*. https://www.midi.org/specifications-old/item/dls-level-1-specification, Accessed October 14, 2020

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 2*. https://www.midi.org/specifications-old/item/dls-level-2-specification, Accessed October 14, 2020

somascape.org. 2019. *Guide to the MIDISoftware Specification*. http://www.somascape.org/midi/tech/spec.html#usx7F0401, Accessed October 14, 2020

SynthLab Documentation. 2020. www.willpirkle.com/synthlab-docs, Accessed on October 14, 2020

3 Synth Voices, Synth Modules, and Module Cores

The SynthLab *SynthVoice* object is responsible for rendering note-events, and there is one voice object per note of polyphony in the synths. As with the engine, the *SynthVoice* object has no base class but is set up to be a base class with the virtual functions and destructor, so feel free to subclass your own when you are ready. The term "voice" has several meanings in synth lingo, but here, it also includes the synth type. The voice architecture is covered in Section 1.5.

3.1 Voice Behavior

The voice performs three tasks during the synth's operation: (1) initialization, (2) responding to MIDI note-on and note-off messages, and (3) controlling the audio signal flow through a set of member objects called modules. The voice object's central responsibility is maintaining the set of *SynthModule* objects that make up the synthesizer components, such as oscillators and filters. I designed the *SynthVoice* object to expose simple functions that service these three areas of operation. The voice object also processes incoming MIDI data for note-on and note-off events, which it uses to control its set of modules.

1 **Initialization**: the voice calls the module's *reset* function
2 **Note-on and note-off**: the voice calls the *doNoteOn* and *doNoteOff* methods on its set of modules
3 **Controlling audio signal flow**: the voice calls the module's *update* and *render* functions during each block processing cycle and delivers the rendered audio back to the engine

Figure 3.1 shows the big-picture block diagram and connection from the engine to the voice to an individual module. This chapter focuses on the blocks in bold: voice, module, and core.

3.2 *SynthVoice* Modules

The SynthLab-DX and SynthLab-PCM synths only differ in their *SynthVoice* objects – everything else is identical. Furthermore, the only thing that differentiates the FM voice from the PCM voice is the set of oscillators used to render the signal. This means that the code for the voice objects is about 95% similar. Each *SynthVoice* object owns a set of objects subclassed from *SynthModule*. Each module is a synth building block and can be grouped into modulators, oscillators, processors, and controllers, as shown in Figure 3.2. The voice object also needs to update these modules with new GUI control information once per buffer process cycle. A voice is a collection of these objects, plus all the code needed to manage them.

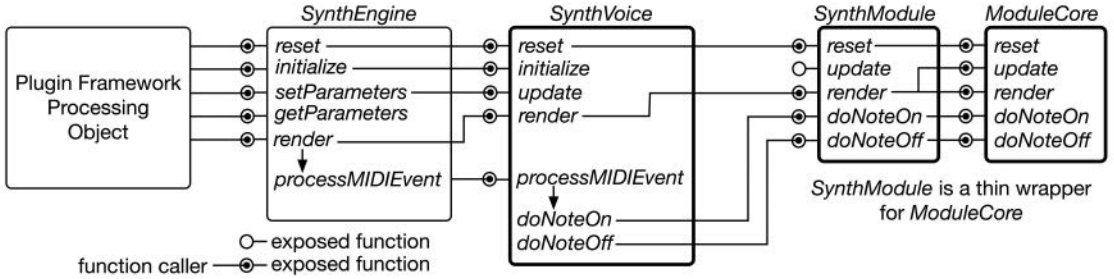


Figure 3.1 Flow of function calls from host plugin framework processing object to the *SynthEngine*, then to the voices, modules, and their cores

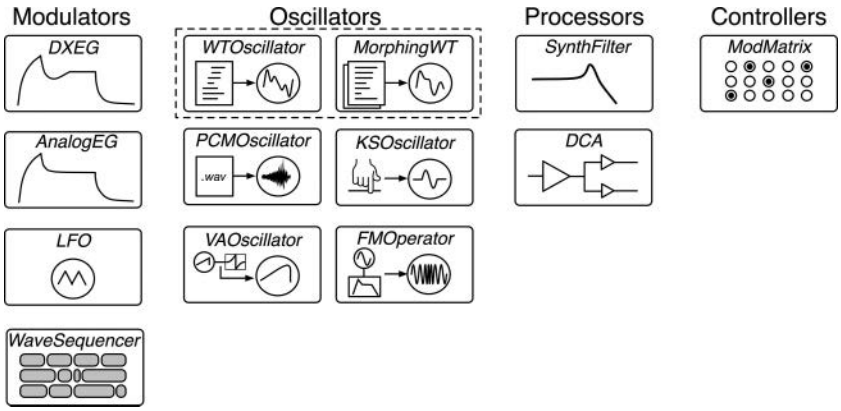


Figure 3.2 SynthLab modules each implement one synth building block; all are subclassed from *SynthModule*; note that the normal and morphing wavetable oscillators are combined (dotted box) and may be mixed freely

3.3 *SynthVoice* Parameters

The *SynthVoiceParameters* structure contains all of the GUI information for programming the voice object and all of the controls for all of the modules in Figure 3.1, plus the additional parameters for the chosen oscillators. All of these are implemented with shared pointers within the *SynthVoiceParameters* structure.

The engine object creates the global *SynthVoiceParameters* during its construction, and all of the shared pointers and parameter structures for each of the voice's modules are also created since they are members of the voice parameter structure. The top part of the declaration is for voice parameters whose names indicate their target functionality.

Next, each parameter structure is declared with *make_shared,* thereby constructing each underlying structure and generating the safe pointer, in a series of declarations. This is the underlying mechanism that allows both the voice object and its modules to share common parameters exclusive to each of them. These structures are where you will place the GUI control information, as described in Chapter 5.

```
// --- LFOs
std::shared_ptr<LFOParameters> lfo1Parameters =
                                std::make_shared<LFOParameters>();
std::shared_ptr<LFOParameters> lfo2Parameters =
                                std::make_shared<LFOParameters>();

// --- EGs
std::shared_ptr<EGParameters> ampEGParameters =
                                std::make_shared<EGParameters>();
std::shared_ptr<EGParameters> filterEGParameters =
                                std::make_shared<EGParameters>();
// --- etc . . .
```

3.4 *SynthModule* Members

Each voice's member modules are exclusive to that voice and are not shared with other objects. The modules are all maintained with an *std::unique_ptr* and are declared in the class definition, then constructed in the *SynthVoice* constructor. Table 3.1 lists the modules that are identical across all voices – of course, you may add, remove, or change the modules for your own designs.

The declarations are simple:

```
// --- modules common to all voices:
std::unique_ptr<DCA> dca;
std::unique_ptr<AnalogEG> ampEG;
std::unique_ptr<SynthLFO> lfo1;
std::unique_ptr<SynthFilter> filter1;
etc . . .
```

The *SynthVoice* members include shared pointers to the MIDI input, wavetable, and PCM sample databases in Section 2.3. These are declared along with the modules, then populated during the voice constructor.

```
std::shared_ptr<MidiInputData> midiInputData = nullptr;
std::shared_ptr<WavetableDatabase> wavetableDatabase = nullptr;
std::shared_ptr<PCMSampleDatabase> sampleDatabase = nullptr;
```

Table 3.1 SynthVoice modules that are identical across all synth projects

Module Object Name	*SynthVoice std::unique_ptr< >*	*Notes*
DCA	*dca*	Output only DCA
AnalogEG	*ampEG;*	Hardwired to DCA
AnalogEG	*filterEG;*	Hardwired to filter 1
AnalogEG	*auxEG*	Assignable EG modulator
SynthLFO	*lfo1*	LFO1 may modulate LFO2
SynthLFO	*lfo2*	General purpose LFO
SynthFilter	*filter1*	18 different filters, plus bypass
SynthFilter	*filter2*	18 different filters, plus bypass
ModMatrix	*modMatrix*	Modulation matrix object
WaveSequencere	*waveSequencer*	Generates modulation values only; specific to SynthLab-WS

3.5 *SynthVoice* Construction

The voice object must create all of its modules and pass them the MIDI input, wavetable, and PCM sample database pointers at construction time. The *SynthVoice* constructor prototype is shown here; the *midiOutputData* is not used but is available for your own experimentation. Notice the maximum block-size parameter – this is needed for the modules to create their audio output buffers at that maximum size.

```
SynthVoice(std::shared_ptr<MidiInputData> _midiInputData,
           const std::shared_ptr<MidiOutputData> _midiOutputData,
           std::shared_ptr<SynthVoiceParameters> _parameters,
           std::shared_ptr<WavetableDatabase> _wavetableDatabase,
           std::shared_ptr<PCMSampleDatabase> _sampleDatabase,
           uint32_t _blockSize = 64);
```

During construction, the voice object needs to do the following:

1 Create all sub-modules using *std::unique_ptr* and its *reset* function, passing in the shared module parameter structures from its voice parameters structure
2 Add sources and destinations to the modulation matrix (covered in detail in Chapter 14)

The module creation follows the same paradigm – each module receives a shared pointer to a structure within the voice's shared parameters for GUI updates. The wavetable and PCM sample oscillators are also given database pointers; the database details and usage are explained in each oscillator chapter, as are the constructor prototypes, but they are all fundamentally similar. Here is the construction of a few of the modules: it is easy to see where the shared parameters and databases are part of each constructor; the modulation matrix object only requires the shared parameters. Stand-alone mode construction is generally simpler and is covered in Section 5.3.1.
LFO:

```
lfo1.reset(new SynthLFO(midiInputData,  /* shared midi data */
           parameters->lfo1Parameters,  /* shared parameters */
           blockSize));                 /* max blocksize */
```

Filter:

```
filter1.reset(new SynthFilter(midiInputData,  /* shared midi data */
              parameters->filter1Parameters,  /* shared parameters */
              blockSize));
```

Wavetable Oscillator:

```
wtOsc.reset(new WTOscillator(midiInputData,   /* shared midi data */
            parameters->wtOscParameters,      /* shared parameters */
            wavetableDatabase,                /* shared wavetables */
            blockSize));
```

PCM Sample Oscillator:

```
pcmOsc.reset(new PCMOscillator(midiInputData,  /* shared midi data */
             parameters->pcmOscParameters,     /* shared parameters */
```

```
            sampleDatabase,     /* shared PCM data */
            blockSize));
```

Modulation Matrix:

```
modMatrix.reset(new ModMatrix(parameters->modMatrixParameters));
```

3.6 *SynthVoice* Operational Phases

You can see from Figure 3.1 that the voice and module object share nearly identical member functions, and many of the voice functions simply call the same named functions on the modules.

3.6.1 SynthVoice *Reset*

This function calls the *reset* functions on all of its modules. It also sets the *currentMIDINote* flag to −1, indicating that no notes are playing. This will be used for polyphony and voice-stealing in the next chapter. We will look into each *reset* function in the corresponding module chapters.

```
pcmOsc->reset( _ sampleRate);
dca->reset( _ sampleRate);

lfo1->reset( _ sampleRate);
lfo2->reset( _ sampleRate);

ampEG->reset( _ sampleRate);
filterEG->reset( _ sampleRate);
auxEG->reset( _ sampleRate);

filter1->reset( _ sampleRate);
filter2->reset( _ sampleRate);

etc...
```

3.6.2 SynthVoice *Update*

The *update* function is called after the GUI parameters have been set on the engine. For the *SynthVoice*, this only consists of swapping module cores in or out, as described in Section 3.7, when the user selects them from the GUI. The modules are identified with index values that you use in your GUI's control so they are not related to any framework.

3.6.3 SynthVoice doNoteOn and doNoteOff

The voice's *processMIDIEvent* function is called repeatedly at the top of each processing block to service all MIDI messages for that audio block. This function only looks for note-on and note-off messages, then calls the voice's corresponding functions that accept *midiEvents* as arguments (see Section 2.3.2). These methods perform two main functions: calling the note-on or note-off message on all of the sub modules, and setting a state variable so the voice knows its state, which is used for voice-stealing.

```
enum class voiceState { kNoteOnState, kNoteOffState };
```

```
SynthVoice::doNoteOn(midiEvent& event)
```

The note-on function is a bit more complex as it involves portamento, which uses the *GlideModulator* object described in Section 6.7.1. Each SynthModule has its own pre-defined glide modulator member object. The MIDI note number is converted to a pitch and stored as the current note, and the previous note is recalled. If portamento is enabled in the voice parameters (which would have come from a GUI control), the note information and glide time are used to start the glide modulator.

```
// --- calculate MIDI -> pitch value
double midiPitch = midiNoteNumberToOscFrequency(event.midiData1);
int32_t lastMIDINote = currentMIDINote;
currentMIDINote = (int32_t)event.midiData1;

GlideInfo glideInfo(lastMIDINote, currentMIDINote,
                    parameters->glideTime_mSec, sampleRate);

if (parameters->enablePortamento && lastMIDINote >= 0)
{
       xxxOsc1->startGlideModulation(glideInfo);
       xxxOsc2->startGlideModulation(glideInfo);
       xxxOsc3->startGlideModulation(glideInfo);
       xxxOsc4->startGlideModulation(glideInfo);
}
```

Next, the series of note-on message handlers is called, and the state variable and *voiceIsRunning* members are set.

```
MIDINoteEvent noteEvent(midiPitch, event.midiData1, event.midiData2);


// --- start oscillators
vaOsc1->doNoteOn(noteEvent);
vaOsc1->doNoteOn(noteEvent);
vaOsc3->doNoteOn(noteEvent);
vaOsc4->doNoteOn(noteEvent);

// --- modulators/processors
dca->doNoteOn(noteEvent);
ampEG->doNoteOn(noteEvent);
filterEG->doNoteOn(noteEvent);
etc…

// --- set the flag
voiceIsRunning = true; // --- we are alive!
voiceNoteState = voiceState::kNoteOnState;
```

The note-off handler is simpler, merely calling the module functions and saving the state variable.

```
SynthVoice::doNoteOff(midiEvent& event)


// --- lookup MIDI -> pitch value
double midiPitch = midiFreqTable[event.midiData1];

MIDINoteEvent noteEvent(midiPitch, event.midiData1, event.midiData2);

ampEG->doNoteOff(noteEvent);
filterEG->doNoteOff(noteEvent);

etc…

// --- set our current state
voiceNoteState = voiceState::kNoteOffState;
```

3.6.4 SynthVoice *Render*

The rendering function in the *SynthVoice* is quite simple as the modules do the hard work. The function accepts the *SynthProcessInfo* structure that the engine sends in and uses the *samplesToProcess* during the calls to the sub-modules.

```
SynthVoice::render(SynthProcessInfo& synthProcessInfo)
```

The function follows this pattern:

1 Call a *render* on all modulator member modules
2 Run the modulation matrix to transfer modulator sources to destinations (see Chapter 14)
3 Process the audio from the oscillators through the filters and DCA, managing the flow of audio buffers; this is somewhat dependent on the voice type

Each module's *render* function makes a call to the core's *update* method first. The *update* functions will set the modulator's parameters for the current block, then each modulator will generate one output value for the block during the *render* operation. The modulation matrix applies the modulator values to the destinations. This chain of update/render is identical across all SynthLab voice objects.

```
// --- render modulators
lfo1->render(samplesToProcess);
lfo2->render(samplesToProcess);

ampEG->render(samplesToProcess);
filterEG->render(samplesToProcess);
auxEG->render(samplesToProcess);

// --- do all mods
modMatrix->runModMatrix();
```

Next, the oscillators are rendered; this will depend on the voice type. We will look at this process for each synth project in Chapter 16. It is more important to understand how the audio is transferred from the oscillators to the filters to the output DCA. All processes are done with helper functions that perform fast *memcpy* operations on the audio buffers, transferring blocks of audio data down the signal path. These are named *copyOutputToInput* and *copyOutputToOutput*, and take buffer pointers, samples to copy, and a channel indicator (mono to mono, stereo to stereo, etc.).

Here are some examples:

1 Render and transfer audio from the virtual analog oscillator (*vaOsc*) output into the *filter1* input:

```
vaOsc->render(samplesToProcess);


copyOutputToInput(vaOsc->getAudioBuffers(),     /* source */
                  filter1->getAudioBuffers(),   /* destination */
                  STEREO_TO_STEREO,             /* channel info */
                  samplesToProcess);            /* samples to copy */
```

2 Run *filter1*, and copy its output to the input of *dca*:

```
filter1->render(samplesToProcess);

copyOutputToInput(filter1->getAudioBuffers(),   /* source */
                  dca->getAudioBuffers(),       /* destination */
                  STEREO_TO_STEREO,
                  samplesToProcess);
```

3 Run the *dca*, and copy its output to the output buffers, which the engine passed to the voice when the render was called – note that the *copyOutputToInput* function includes multiple versions that accept different types of input pointers so they are simple to use

```
// --- update and render
dca->render(samplesToProcess);

// --- copy to mains out
copyOutputToOutput(dca->getAudioBuffers(),    /* source */
                   synthProcessInfo,          /* NOTE: destination! */
                   STEREO_TO_STEREO,
                   samplesToProcess);
```

The MIDI note-on message places the voice into its active state, notifying the engine to call its render operation. The MIDI note-off message places the voice into the note-off state, but the voice does not fully finish operations until the output EG that is connected to the output amplifier has finished its release state and expired into the off state. At this point, the voice may be shut down.

The very last part of the *render* function is also one of the most important – checking for the note-off state and an expired *ampEG* object. This will be covered in detail in Chapter 4.

3.7 SynthModules and ModuleCores

The module-core paradigm is fundamental to SynthLab. Each module is subclassed from the *SynthModule* base class, and each implements the same set of overridden base class functions which correspond nearly identically with the voice object functions. Figure 3.2 shows how I diagram the modules, using the *PCMOscillator* as an example. The figure shows everything you need to know without needing large data tables.

At the top of the diagram, you can see the GUI control knobs. There are always four assigned knobs at the top and four assignable controls called "mod knobs" shown below, labeled A, B, C, and D. In some modules, all are assigned; others have "free" controls with which you can easily perform experiments without needing to modify a bunch of code. These controls funnel into the *parameters* block, which represents the shared parameter structure for this module. The parameters are applied to the active *ModuleCore* object, which is an internally owned member that has access to the *SynthModule's* input and output ports, labeled with icons and text. Different modules require access to different ports, which are shown as enabled when connected to the core with a bar, also shown in Figure 3.3(a). You will sometimes see a dotted box around a module core's sting list – this indicates a separate downloadable project that can be used for building only the module core in the SynthLab-DM projects.

3.7.1 SynthModule *I/O Ports*

The I/O ports connect a module to its input and output sources. Figure 3.3(b) shows a block diagram of the *SynthModule* with inputs and outputs. Modulation inputs and outputs are arrays of *double* values, one modulation value per audio block, but with numerous slots in the arrays for various modulator types. Audio data is transferred via the *AudioBuffer* described in Section 2.5.1.

3.7.1.1 Input Ports

- MIDI input is provided via the engine's shared structure
- Modulation input values arrive in a pre-defined modulation array with one active modulation value processed per block; there may be many inputs, each with its own slot in the array (e.g. bipolar input (from LFO)), unipolar input (from EG), etc.
- FM inputs are from outputs of other oscillator modules and include one block's worth of audio data
- Audio inputs allow you to send external audio data to the module (e.g. from a side chain or vocoder microphone input); these are declared for you but not used in the book projects

3.7.1.2 Output Ports

- Modulation output values are written into an array that has pre-defined slots for the various modulators; e.g. the LFO writes to the bipolar array slot, while the EG writes to the unipolar slot
- FM outputs are identical to the audio output for a given module; all oscillator module output buffers may be used as FM input buffers for other modules

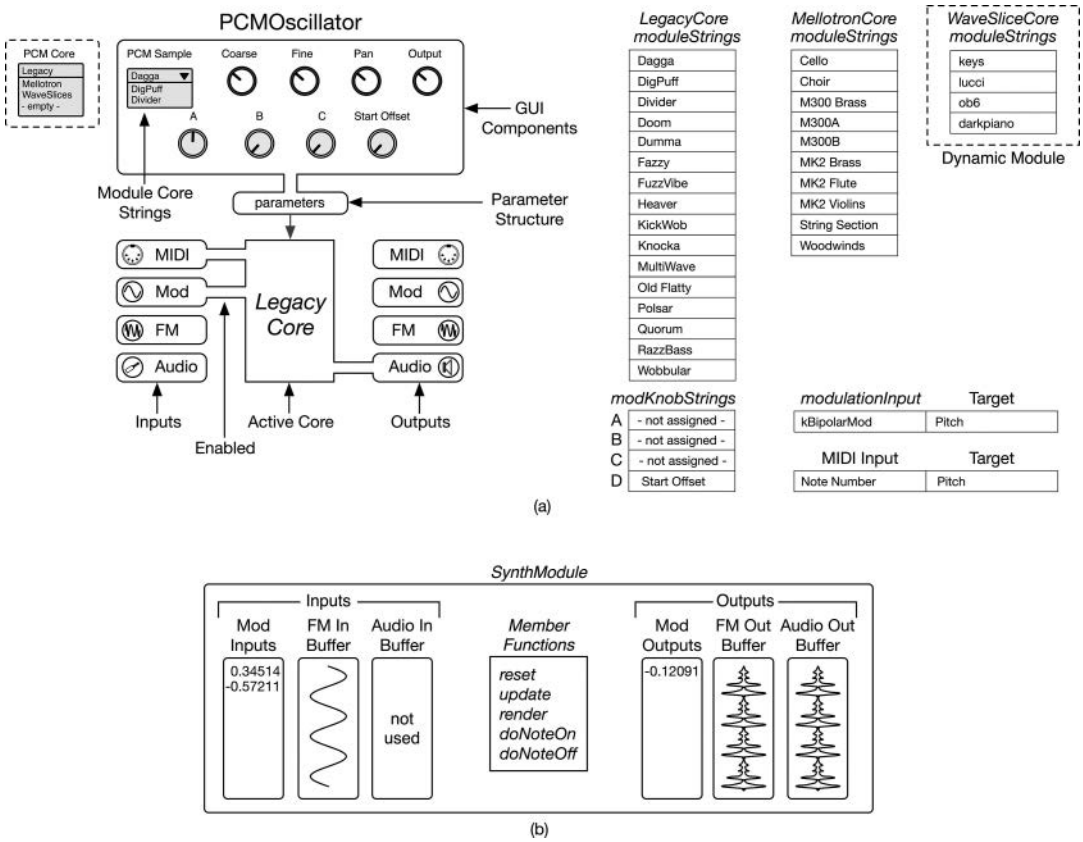


Figure 3.3 (a) Module diagrams show attributes and connections in the module and its core, along with module and mod knob strings for each core; notice that the *moduleStrings* show up in the GUI as a selection list and (b) the *SynthModule* block diagram

- Audio output is written into a pre-prepared *AudioBuffer* object that always has two channels (for dual-mono or stereo operation) and is only valid for oscillators as the modulators write to the modulation arrays
- A MIDI output structure is provided but not used in the projects

3.7.2 SynthModule *Virtual Functions*

All modules must override the five *SynthModule* virtual functions: *reset*, *update*, *render*, *doNoteOn*, and *doNoteOff*. These correspond to the five *SynthVoice* virtual functions with identical names. The engine calls the voice object functions, which, in turn, call the individual module functions. Table 3.2 lists these virtual functions, along with a brief description; they are also shown in Figure 3.2. We will look deeper into these functions as they apply to the different modules inside

Table 3.2 SynthModule virtual functions that all modules must override and descriptions

Function Name	*Description*
reset	Sets the sample rate and resets member objects
update	Using GUI and modulation input, calculate the object variables for the current block of data (e.g. filter f_c, EG release time,…)
render	Render output; modulators write to modulation outputs, oscillators write to audio buffers, processors read from and write to audio buffers
doNoteOn	Perform note-on activities
doNoteOff	Perform note-off activities (used on EG only)

of their respective chapters. The main point here is that the synth voice will call these functions directly from its own functions of the same name.

3.8 Module Cores

Each *SynthModule* owns an array of up to four *ModuleCore* objects. The module cores are specialized components that operate within the overall behavior of the module but with individualized functionality. For example, the *SynthFilter* is a *SynthModule* that has two module cores: *VAFilterCore* and *BQFilterCore*. Both of these implement audio filters, but one is used specifically for the virtual analog filters described in Chapter 12, while the other provides alternate but useful filters based on the biquad algorithms in the first edition FX book. When the user selects a core, two things happen: first, the new core object becomes the "active core," and second, its specialized string list of up to 16 items is loaded into a GUI control for the user. In Figure 3.2, you can see the module core names in a list in the dotted box; this GUI control is available for all modules and is not shown in the individual module chapter diagrams.

3.8.1 ModuleCore *Strings*

The *VAFilterCore* and *BQFilterCore* will likely expose different strings for their different filters, and if some stings have the same name, that is fine. In Figure 3.2, you can see the core module strings for the PCM oscillator's *LegacyCore* object as it appears in the module chapter. This array of strings is a *ModuleCore* member, and you supply the string names in the core's constructor. Figure 3.4(a) shows how the module strings are related to their cores and how these will appear to the user on a typical GUI using the *PCMOscillator* and its cores that expose the bank waveforms. For this object, each core is slightly different: the legacy core uses transient + loop PCM samples, while the Mellotron uses single long recordings of notes, and the wave slice core uses the Aubio audio tool to slice up the *.wav* files that are its data arrays.

3.9 SynthModule/ModuleCore Relationship

The *SynthModule* is an ultra-light wrapper around the active core; examine the class definition in *synthbase.h*, and you will see that the *ModuleCore* base class virtual functions have the same names as the *SynthModule* functions in Table 3.2. The difference is in the argument that is passed

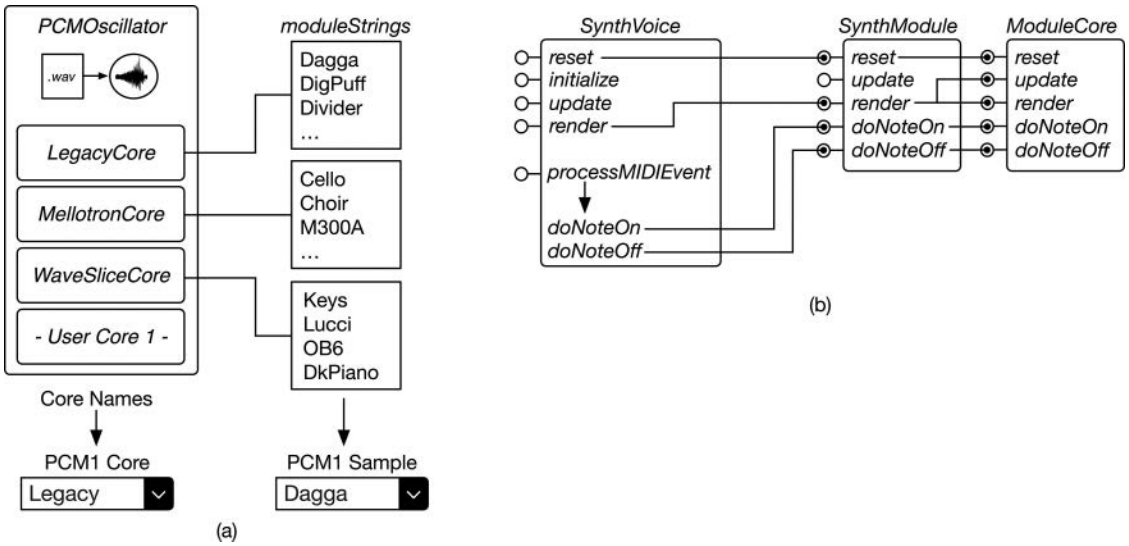


Figure 3.4 (a) The PCM oscillator includes three pre-written cores and one empty (user) core that allow the PCM oscillator to implement banks of waveforms; (b) the relationship between voice, module, and core is tight

to the core virtual functions. Figure 3.2(b) shows a module with its module I/O and parameters. The core sits inside and does the real work for each of the module's functions. In order to give the core access to the I/O buffers, modulation arrays, and parameters, the module passes it a data *CoreProcData* structure that contains mainly interface pointers and references to these resources, and the core uses them during its function calls. There are some advantages to using this paradigm for SynthLab:

1 Cores are simple and compact; if you want to focus your work on just one type of module and not the entire synth, then you only have a few files to edit – usually just two
2 Cores follow the idea of C++ encapsulation; each one hides the details of functionality of a concept that is encapsulated; the filter cores generate filters in drastically different ways (virtual analog versus biquad), but the user only sees another bank of filters
3 Cores allow for the concept of program and data "banks" in which each core implements 16 different variations on its basic theme, each being presented to the user in a list every time the core changes; each wavetable oscillator core generates a bank of waveforms, each PCM sample oscillator generates a bank of samples, each filter core creates a set of filters, etc.

Cores allow me to make a version of SynthLab that can dynamically load the cores you write and compile as a plugin-within-a-plugin; my version of the SynthLab-DM projects is able to load core DLLs into itself at startup time (you can add this functionality yourself if you know how to user your plugin framework).

If you are using my SynthLab pre-compiled plugins, you can build a "core plugin" – a plugin that is loaded into the SynthLab plugin at startup time – allowing you to customize each module for yourself. This allows you to go through the book, learning about each module and its parameters, and understanding its inner code and theory of operation. The cores are pure C++ and are not tied to any plugin framework, requiring a minimal compiler setup that is so simple, you won't need *CMake*. You can also build your own modules in any component flavor and add them to the existing plugin. This means that my SynthLab plugins are dynamic, and you may modify and change their core operations – all of the modules in Figure 3.1, if you like – to suit your own research or interest areas.

3.9.1 CoreProcData *Structure*

The cores require access to the module's I/O ports and parameters, accomplished with the *CoreProcData* structure that is setup with only basic C++ data types and agnostic C++ interfaces, whose functions likewise only contain the basic data types. Unlike most of SynthLab, this structure does not use the *std::* library and has no shared pointers and no *std::strings*. This allows for the dynamic loading of core modules (mini-plugins); any function calls from the host (the plugin DLL) to the server (the core DLL) must cross the "thunk" barrier, meaning that arguments passed back and forth must have C++ memory footprints that are exactly described and known a-priori. The *CoreProcData* consists of input and output audio buffer pointers, along with several safe interfaces for accessing the outer module's resources, including *IModulator*, *IWavetableDatabase*, *IPCMSampleDatabase*, and *IMIDIInput*, all of which are documented online. Since each module has a separate custom parameter structure, a *void** is used to pass this structure in and out of the core using *void** cloaking and un-cloaking.

3.9.2 SynthModule *Constructor*

The *SynthModule* constructors follow a three-step pattern:

1 Setup the audio input and output buffers; oscillators only have outputs, while processors (filters and amplifiers) have both inputs and outputs
2 Set up a *CoreProcData* member structure that will be used for communication with the core
3 Instantiate the cores, store them, and select the default core

This code is fundamentally identical in every module-core relationship and only varies with the kind of core and audio capabilities; there is no need to repeat it in every chapter. Here is the constructor for the *PCMOscillator* that creates three member cores.

Create audio buffers for inter-module signal flow; note use of constants with 0 inputs and 2 outputs:

```
// --- create our audio buffers
audioBuffers.reset(new SynthProcessInfo(SMPL_OSC_INPUTS, /* 0 */
                                        SMPL_OSC_OUTPUTS, /* 2 */
                                        blockSize)); // prefer 32 or 64
```

The *coreProcessData* structure gets its interface pointers from the module's resources: audio I/O buffers, modulation I/O arrays, MIDI data, and databases. Examine each core's constructor for the details:

```
// --- setup the core processing structure for dynamic cores
coreProcessData.inputBuffers = getAudioBuffers()->getInputBuffers();
coreProcessData.outputBuffers = getAudioBuffers()->getOutputBuffers();
coreProcessData.modulationInputs =
                                    modulationInput->getModulatorPtr();
coreProcessData.modulationOutputs =
                                modulationOutput->getModulatorPtr();
etc. . .
```

Finally, create the cores, and store them on the module's array, then select the first core as the default.

```
// --- setup the cores
for (uint32_t i = 0; i < NUM_MODULE_CORES; i++)
     moduleCores[i] = nullptr;

std::shared_ptr<PCMLegacyCore>
                    defaultCore = std::make_shared<PCMLegacyCore>();
addModuleCore(std::static_pointer_cast<ModuleCore>(defaultCore));

etc…

// --- core[0]
selectDefaultModuleCore();
```

3.9.3 SynthModule *Wrapper Functions*

Most of the module-to-core functions are trivial as they simply call the selected module's functions of the same name with the *coreProcData* argument. The *reset* function must reset all of the cores, while the rest of the functions operate on the selected core. Here is the *reset* function for the *PCMOscillator*:

```
bool PCMOscillator::reset(double _sampleRate)
{
     // --- PCM cores need path to DLL for adding to database
     coreProcessData.dllPath = dllDirectory.c_str();
     coreProcessData.sampleRate = _sampleRate;

     for (uint32_t i = 0; i < NUM_MODULE_CORES; i++)
     {
          if(moduleCores[i])
               moduleCores[i]->reset(coreProcessData);
     }
```

```
    return true;
}
```

The update function merely calls the core version:

```
bool PCMOscillator::update(){
    return selectedCore->update(coreProcessData);
}
```

The render function parses the *samplesToProcess* variable and updates the *coreProcData* structure:

```
bool PCMOscillator::render(uint32_t samplesToProcess){
    // --- update
    coreProcessData.samplesToProcess = samplesToProcess;

    return selectedCore->render(coreProcessData);
}
```

The note-on and note-off functions apply the current *MIDINoteEvent*:

```
bool PCMOscillator::doNoteOn(MIDINoteEvent& noteEvent)
{
    coreProcessData.noteEvent = noteEvent;

    return selectedCore->doNoteOn(coreProcessData);
}
```

3.10 Review

Chapters 2 and 3 contain the fundamentals of how the synth engine, voices, modules, and their module cores are related and used to create the synth projects, and show the main coding details of each object connection. The idea is to give you an understanding of how Figure 3.1 works on the different levels – from engine to voice to modules to cores. Go back and review Figure 3.1, and follow the function calls from the framework to the engine's *render* operation. The engine calls the voice *render*, which calls *render* on all modules, which, in turn, call *update* and *render* on the active core.

Chapters 7–13 and 14–15 contain *SynthModule* and *ModuleCores* for each of the building blocks used for the SynthLab projects. Each of these "module chapters" has the theory of operation, followed by the module and core descriptions and implementation strategies. There are a considerable number of interfaces, synth objects, helper objects, and functions included, and these are documented online at www.willpirkle.com/SynthLab-docs. You will need to download the code for the projects and spend time comparing the code with what you read in the text as you go through the book, which will also take considerable patience. After the module chapters, we will put the pieces together in the complete SynthLab projects. You have several options for their implementation, which are discussed in Chapter 5.

Bibliography

Gabrielli, Leonard. 2020. *Developing Virtual Synthesizers with VCV Rack*, Chaps. 1, 3. New York: Routledge.
SynthLab Documentation. 2020. www.willpirkle.com/synthlab-docs, Accessed on October 14, 2020

4 Synth Operational Modes

Polyphony and Voice-Stealing

Synthesizers have three basic operational modes: monophonic (mono), monophonic-unison (unison), and polyphonic (poly). The two monophonic modes include an option for legato operation in which the amp EG does not re-trigger for new note-events; this is covered in Section 7.1.3. Note that legato mode only applies if the user holds a key down, then presses another key without releasing the first key, generating two note-on events in succession. It does not apply when playing mono but releasing each key before triggering a new note. On some synths, the legato mode has its own dedicated switch; for SynthLab, I combined legato mode with the two mono modes to produce a total of five synth modes of operation:

1. **Mono**: basic monophonic operation that uses the same voice object for every note-event; new note-events re-start the amp EG and produce transient edges (clicks)
2. **Legato**: monophonic mode with legato slurring
3. **Unison**: monophonic mode with multiple voice objects mixed together and each detuned slightly, producing a thick sound
4. **Unison legato**: unison mode with legato slurring
5. **Polyphonic**: polyphonic mode; when all voices are exhausted, and a new note-event arrives, an existing voice is sacrificed to play the new note, aka voice-stealing

When dealing with mono and unison modes (non-legato), we must make an important decision on handling the re-triggering of the amp EG when the new note-event arrives. One of our options is to trigger the new note immediately, placing the EG back into the attack state from wherever the EG was in its cycle. The second is to shut down the EG through an ultra-fast release time of about a millisecond, thereby bringing the output down to zero, then moving to the attack state and re-starting the EG. The second mode is called *return-to-zero* (RTZ). Both modes are covered in Sections 7.1.2 and 7.1.3, and shown in Figure 7.3. The EG objects expose an additional virtual function named *shutdown* that places the EG into this fast-release state, allowing the voice to quickly shutdown the EG. The *shutdown* function is also used in polyphony during voice-stealing so that one voice may be quickly silenced to make way for a new note-event.

> The SynthLab projects all implement RTZ mode for the monophonic non-legato modes such that the monophonic voice steals from itself if a new note is triggered while the single monophonic voice is playing. The legato modes ignore the *shutdown* operation and are therefore simpler to implement and understand.

4.1 The Note-Event Lifecycle

It is important to clarify the difference between the note-off event and the end-of-note signal. As it turns out, the amp EG plays a major role in the grand scheme of things. The amp EG is the envelope generator attached to the output DCA; it produces the synthesized signal's time domain envelope. Figure 4.1 shows a synth voice consisting of an oscillator and a DCA that is connected to the amp EG and shows the four parts of the note-event lifecycle:

1 **Note-on message**: resets objects and starts amp EG in attack state
2 **Note-off message**: places amp EG into the release state; all other modules continue operation as normal
3 **Waiting for release**: the voice monitors the state of the amp EG, waiting until its release state has finished
4 **Release complete**: when the amp EG finishes its release phase, it goes into the off state; only at this point is the note-event finished and the voice off

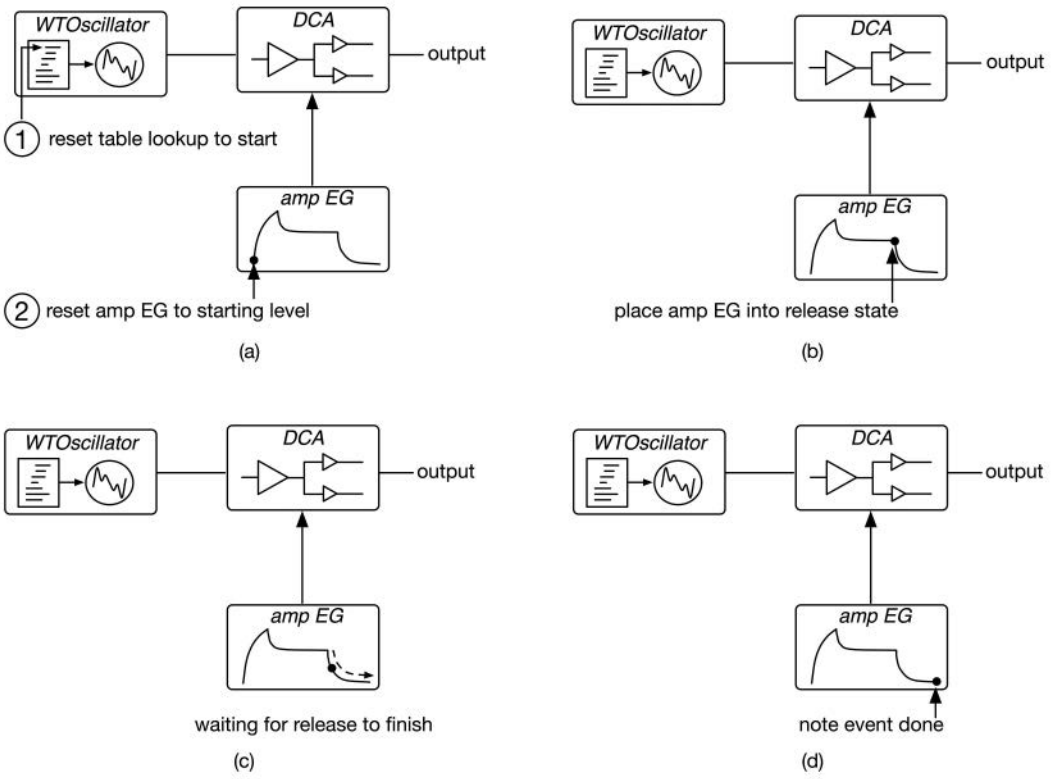


Figure 4.1 The four states of the note-event lifecycle: (a) the note-on message resets objects and places the amp EG into the attack state; (b) the note-off message places the amp EG into the release state; (c) the voice waits for the amp EG to finish, rendering output the entire time; and (d) the note-event is finished when the amp EG reaches the off state

4.2 *SynthEngine* MIDI Processing

The engine and voice object both handle the note-on and note-off messaging. The engine's *processMIDIEvent* method splits the incoming MIDI messages into note-events and CC events, then calls the voice's *processMIDIEvent*, which handles the more complicated details; it is shown in Figure 4.2. The block marked "Choose Target Voice" represents the logic for the five modes of operation. For mono and unison modes, the choice is hardwired but more complex for the poly mode that involves voice-stealing. This chapter is all about making that choice.

4.2.1 SynthEngine *Mono Modes*

The engine parameters include a variable for the operational mode coded as an index-based parameter from the GUI control that displays the modes for the user. The index is compared to a strongly typed *enum* for easier code readability:

```
enum class synthMode { kMono, kLegato, kUnison, kUnisonLegato, kPoly };
```

The *SynthEngine* performs the initial MIDI message processing, as covered in Section 2.3.2.3, and implements the engine-side of the synth operational modes. For mono and legato modes, the *SynthEngine* uses the first *SynthVoice* object in its array for all note-events and simply forwards the MIDI event to the voice object for processing. The engine's *processMIDIEvent* function is blissfully simple for mono and legato. The note-on and note-off code is shown here, using the GUI index parameter for decoding the mode:

```
bool SynthEngine::processMIDIEvent(midiEvent& event)
```

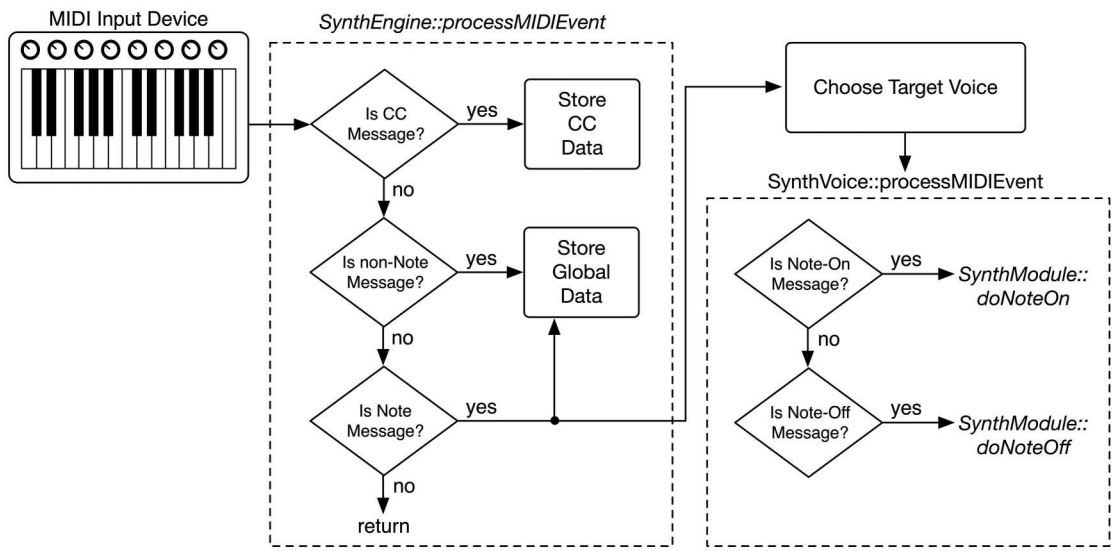


Figure 4.2 The engine decodes the incoming MIDI messages; for note-on and note-off, the engine first finds a target voice, then forwards the MIDI processing function call

For note-on operation the engine saves the current note number and velocity in the global MIDI data for the objects to share – for example, filter key-tracking and oscillator portamento both require knowing the MIDI note number.

```
if (event.midiMessage == NOTE_ON)
{
      // --- set current MIDI data
      midiInputData->setGlobalMIDIData(kCurrentMIDINoteNumber,
                                       event.midiData1);
      midiInputData->setGlobalMIDIData(kCurrentMIDINoteVelocity,
                                        event.midiData2);

      // --- mono mode
      if (parameters->synthModeIndex == enumToInt(synthMode::kMono) ||
          parameters->synthModeIndex == enumToInt(synthMode::kLegato))
      {
            // --- just use voice 0
            synthVoices[0]->processMIDIEvent(event);
      }
```

For note-off operation, the engine first queries the voice with *isVoiceActive*, then sends the MIDI message to the voice and returns immediately.

```
else if (event.midiMessage == NOTE_OFF)
{
     // --- for mono, we only use one voice, number [0]
     if (parameters->synthModeIndex == enumToInt(synthMode::kMono) ||
         parameters->synthModeIndex == enumToInt(synthMode::kLegato))
     {
           if (synthVoices[0]->isVoiceActive())
           {
                 synthVoices[0]->processMIDIEvent(event);
                 return true;
           }
     }
}
```

4.2.2 SynthEngine *Unison Modes*

Unison and unison legato are handled just like the mono modes except that multiple voices are all called at once with the same MIDI messages, placing multiple voices into the running state. Both unison modes involve detuning the voices slightly and setting the oscillators to start at slightly different phases so they do not sum together in a way that produces a click or pop due to the in-phase components. For SynthLab, unison mode uses four voices: *synthVoice[0] – synthVoice[3].* The engine's MIDI processing is identical to the mono modes, except for the fact that it uses multiple voices.

For both note-on and note-off operation, the engine calls the four voice functions in succession, and the code is identical for both messages:

```
if (parameters->synthModeIndex == enumToInt(synthMode::kUnison) ||
    parameters->synthModeIndex == enumToInt(synthMode::kUnisonLegato))
{

            // --- SynthLab: 4 voices
            synthVoices[0]->processMIDIEvent(event);
            synthVoices[1]->processMIDIEvent(event);
            synthVoices[2]->processMIDIEvent(event);
            synthVoices[3]->processMIDIEvent(event);
}
```

4.3 *SynthEngine* Unison Detuning

The engine is also responsible for setting the detuning, panning, and phase offsets of some of the unison voice objects. This happens in the engine's *setParameters* method so that the detuning is updated if the user changes the control while notes are playing. The engine parameters include the global detuning that the user controls. The voice parameters include their own unison detuning value, which is set differently for each voice to thicken up the sound, as well as a phase offset variable. The code in the engine's *setParameters* method sets these variables, and you have the ability to customize this code as you like. Note that for many synths, unison mode also involves panning the voices differently, but all SynthLab oscillators are stereo in nature, and individual oscillators may be panned. If you want to override this, feel free to do so; a *unisonPan* variable is already declared for you in the voice parameters, but you will need to modify the rendering function accordingly. The engine's *setParameters* function has all the action; notice how the global unison detuning is applied differently to each of the four voices. The phase offsets were set with experimentation and prevent clicks and pops. The engine cycles though the voice objects in order to call their corresponding *update* methods:

```
for (unsigned int i = 0; i < MAX_VOICES; i++)
{
     // --- update voices
     synthVoices[i]->update();
```

Inside this for-loop, we check the mode parameter and issue the detuning and phase offsets – note that the phase offsets only apply during note-on operation.

```
if (parameters->synthModeIndex == enumToInt(synthMode::kUnison) ||
    parameters->synthModeIndex == enumToInt(synthMode::kUnisonLegato))
{
// --- voice 0
     if (i == 0)
     {
          parameters->voiceParameters->unisonDetuneCents = 0.0;
          parameters->voiceParameters->unisonStartPhase = 0.0;
          parameters->voiceParameters->dcaParameters->panValue = 0.5;
     }
```

```
// --- voice 1
else if (i == 1)
{
     parameters->voiceParameters->unisonDetuneCents =
                 parameters->globalUnisonDetune_Cents;
     parameters->voiceParameters->unisonStartPhase = 13.0;
     parameters->voiceParameters->dcaParameters->panValue = -1.0;
}
etc . . .
```

Table 4.1 lists the voice detuning and phase offsets for unison mode – feel free to change these or add more voices for a thicker unison sound.

4.4 Voice State and MIDI Event Storage

The voice object maintains a few Boolean flags, which it uses to help implement the synth modes. The voice keeps track of a simple state variable named *voiceIsActive* that is true as long as the voice is rendering output in the first three phases of Figure 2.1 and is accessed via the *isVoiceActive* method. When a new MIDI note-on event is received, the data is stored in the member structure *voiceMIDIEvent*, which persists until the note is either turned off or stolen. During steal operations, the voice keeps track of a Boolean flag *stealPending* during the steal operation and another MIDI structure *voiceStealMIDIEvent* that stores information about the new note. The voice state variable and steal flag are used along with the MIDI event structures to implement the steal operation.

4.5 Voice-Stealing

Voice-stealing is usually associated with polyphonic operation. The *SynthEngine* declares some number of voice objects based on the synth type, target device, CPU power, and other information. For simplicity, the SynthLab projects have a fixed number of voices, but you can override that behavior in the constructor for your own experiments. If the user presses enough keys or generates enough note-on messages, the engine will run out of voices at some point. If this happens and a new note-on message arrives, the engine will need to implement a plan for stealing an existing voice to use for the new note-event. This involves code on both the engine's and the voice object's sides. The engine will need to use a heuristic technique to choose the voice to steal, but after that, it simply issues a note-on message to that voice, as with other voices. On receiving the note-on message, the voice determines that it is being stolen and sets up logic on its side to handle that part.

Table 4.1 Voice detuning, panning, and starting phase offsets for unison mode

Voice Index	*Detuning Coefficients Multiplied with* globalUnisonDetune_Cents	*Pan Value*	*Oscillator Start Phase Offset*
0	0	0.5 (center)	0
1	(+1.0)	−1.0 (left)	13 degrees
2	(−1.0)	+1.0 (right)	−13 degrees
3	(0.707)	−0.5 (halfway left)	37 degrees

4.5.1 Voice-Stealing for RTZ Operation

SynthLab uses the RTZ EG operation in the non-legato mono and unison modes to avoid inconsistent clicks and pops. However, this requires quickly shutting down the current voice, then stealing it once it completes its shutdown. In this mode, voice actually steals itself. The voice implements the *processMIDIEvent* function, which usually calls either its *doNoteOn* or its *doNoteOff* sub-functions during normal operations.

4.5.2 Note-On Steal Operation

The steal operation starts with logic in the voice's note-on message handler, which determines whether the voice is being stolen. Once that determination is made, the following events occur:

1 Voice stores the current note-on MIDI message in a member structure *voiceStealMIDIEvent* – this contains the note number and velocity information for the stolen event
2 Voice calls the *shutdown* function on the amp EG to quickly shut itself down
3 Voice sets *stealPending* so it will know how to handle the end-of-note event when the amp EG expires

4.5.3 Post-Shutdown Steal Operation

During the one-millisecond shutdown time, the voice's oscillators and other modules will continue normal operation for the length of the render process. The voice will need to monitor the amp EG and wait for it to complete its shutdown cycle at the end of the *render* function, as described in Section 4.5.5. Once detected, the voice performs the following:

1 Check the *stealPending* flag to see if the voice is being stolen, and if so:
2 Call the voice's note-off handler – this was never called because the event was stolen
3 Copy the *voiceStealMIDIEvent* structure into the *voiceMIDIEvent* structure
4 Call the note-on handler with the new MIDI event information, switching the pitch and velocity to the stolen note – the steal operation is complete

The RTZ voice self-steal operation is simpler than poly mode stealing because there is no selection of a target voice, so the engine-side code does nothing more than call the voice's *processMIDIEvent* method. Studying RTZ operation first allows you to see the voice-side code in isolation, and it is identical to the poly mode operation.

4.5.4 SynthVoice *Stealing Code Part 1*

The code that implements the steps in Section 4.5.2 is located in the *SynthVoice:: processMIDIEvent* function and is triggered when the voice gets a note-on message while it is still active. Once the EG is shut down, and flags are set, the function returns. If it is not stealing, it simply calls its *doNoteOn* handler.

```
if (event.midiMessage == NOTE_ON)
{
        // --- steal detection
        if (isVoiceActive())
```

```
        {
              // --- save information
              voiceStealMIDIEvent = event;

              // --- set amp EG into shutdown mode
              ampEG->shutdown();

              // --- set the steal flag
              stealPending = true;
              return true;
        }

// --- call the subfunction if NOT stealing
      doNoteOn(event);
}
```

Notice that the note-off message is always handled the same way, with a simple call to the note-off handler.

```
else if (event.midiMessage == NOTE_OFF)
      doNoteOff(event);
```

4.5.5 SynthVoice *Stealing Code Part 2*

At the end of each *render* function call, the voice checks to see if the amp EG has expired into the off state; if not, it simply returns as usual.

```
if (voiceIsActive)
{
       if (ampEG->getState() == enumToInt(EGState::kOff))
       etc . . .
```

At this point, the voice knows that the note-event is finished and will either shut off or perform the steal operation, depending on the state variable we set. For the steal operation, we implement the steps in Section 4.5.3. If no steal is pending, then the voice goes into the inactive state, setting its state variable to *false*.

```
// --- check for steal pending
if (stealPending)
{
      // --- turn off old note event
      doNoteOff(voiceMIDIEvent);

      // --- copy new note MIDI info
      voiceMIDIEvent = voiceStealMIDIEvent;

      // --- turn on the new note
      doNoteOn(voiceMIDIEvent);
```

```
        // --- stealing accomplished!
        stealPending = false;
}
else
        voiceIsActive = false;
```

The two functions here comprise the two-part steal operation that the voice implements for RTZ and poly modes. To complete the chapter's discussion, we just need to examine the engine-side code for polyphonic operation.

4.6 Polyphony and Voice Timestamps

For polyphony, the engine services the note-on message in two stages. First, it tries to find a free, unused voice in its array. If all voices are active, then it finds a suitable voice to steal. This code is implemented in the engine's *processMIDIEvent* function and uses *getFreeVoiceIndex* and *getVoiceIndexToSteal* – note that these return the index in the engine's array of the suitable voice object.

```
else if (parameters->synthModeIndex == enumToInt(synthMode::kPoly))
{
        // --- get index of the next available voice (for note on events)
        int voiceIndex = getFreeVoiceIndex();

        // --- find a voice to steal
        if (voiceIndex < 0)
                voiceIndex = getVoiceIndexToSteal();

        // --- start/steal the voice
        if (voiceIndex >= 0)
                synthVoices[voiceIndex]->processMIDIEvent(event);
```

Once that index is found, the engine simply calls the corresponding voice's *processMIDIEvent* as usual – there are no extra functions or Boolean variables to set since the voice operates in a very strict manner, as described in Sections 4.5.4 and 4.5.5, and knows when it is being stolen. This means that the engine-side code involves selecting the voice to steal and nothing else. Selection of the voice involves a heuristic technique, which is a decision-making operation that has no optimal solution (other than making a new voice object). For voice-stealing, several common heuristics apply, including:

1 Stealing the "oldest" voice's note – the voice that has been playing longer than all the others
2 Stealing the voice with the oldest note that is also *not* the lowest MIDI pitch since drone notes are common across many musical styles
3 Stealing a voice based on the chord being played; choosing inner or outer voices, or other advanced selection criteria
4 Stealing notes based on MIDI channels (e.g. MIDI channel 10 has priority as the general MIDI drum channel)

We are going to use the first heuristic: steal the oldest voice. This requires us to know something about the ordering of the voice note-on events. For this heuristic, we only care about the absolute

age of the voice's active state and can use a simple time-stamping method. We don't care about the *actual* DAW system time that the note-events occur at, just the ordering. This is accomplished with a simple counter named *timestamp*, which is a *SynthVoice* member variable and includes three functions for manipulating it:

```
uint32_t timestamp = 0;


uint32_t getTimestamp() { return timestamp; }
void incrementTimestamp() { timestamp++; }
void clearTimestamp() { timestamp = 0; }
```

During the voice's note-on handler, the timestamp is cleared just before we detect the voice steal.

```
if (event.midiMessage == NOTE_ON)
{
     // --- clear timestamp
     clearTimestamp();

     // --- detect if we are being stolen:
```

This means that after the *processMIDIEvent* function call returns to the engine, the newly active voice will have a *timestamp* value of 0. Immediately after issuing a note-on call to a voice, the engine loops over its array, incrementing the timestamps of all active voices.

```
// --- increment all timestamps for note-on voices
for (int i = 0; i < MAX_VOICES; i++)
{
     if (synthVoices[i]->isVoiceActive())
          synthVoices[i]->incrementTimestamp();
}
```

This means that the newly triggered voice will have *timestamp* = 1, while the previous voice will have *timestamp* = 2, and so on. The voice with the highest timestamp value is the oldest one that's active – the voice to steal has been identified. Figure 4.3 shows a sequence of voice activations and time stamping for an engine with four total voice objects. In Figure 4.3(a), all voices are free, while Figure 4.3(b)–(e) show how the voice timestamps are incremented for one, two, three, and four note-on events. When a new note-on event arrives, the engine finds voice 0 with the largest timestamp and selects it. During the voice note-on handler, the timestamp is reset to 0; it is then immediately incremented with the preceding code and becomes the youngest voice.

The function *getFreeVoiceIndex* is trivial and simply loops over the voices, testing the *voiceIsActive* flag on each one, returning the free index or −1 if not found. The function *getVoiceIndexToSteal* is where you implement your heuristic. To find the oldest voice, you simply examine the timestamps of the active voices:

```
int SynthEngine::getVoiceIndexToSteal()
{
       int index = -1;
       int timestamp = -1;
       int currentTimestamp = -1;
```

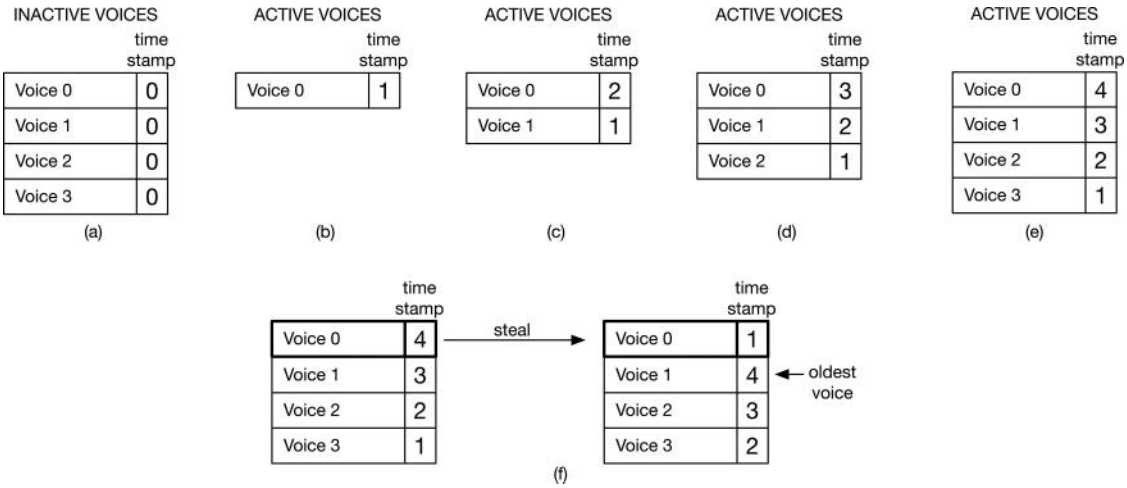


Figure 4.3 An engine with four voices: (a) prior to any notes triggered (b) when the first note is triggered, voice 0 is selected then (c) second note and (d) third note and (e) fourth notes are triggered with timestamps incremented each time and (f) after the oldest voice is stolen, it becomes the youngest and so on

```
        for (int i = 0; i < MAX_VOICES; i++)
        {
              currentTimestamp = (int)(synthVoices[i]->getTimestamp());

              // --- find index of oldest voice
              if (currentTimestamp > timestamp)
              {
                     timestamp = currentTimestamp;
                     index = i;
              }
        }
        return index;
}
```

In addition, the voice has query functions for its MIDI note number, so you may use heuristics that involve a note's pitch in relation to the other voices. For example, to find the oldest voice that is not playing the lowest note, you would modify the function above to find the two oldest voices, then query each one for the MIDI note number and return the index of the voice with the higher of the two values.

```
uint32_t getMIDINoteNumber() { return voiceMIDIEvent.midiData1; }
```

4.7 Review

The MIDI note lifecycle depends on the amp EG's state to determine when the note-event has finished so that plays a role in the synth modes. The legato modes do not reset the amp EG, so there is no click to suppress, and their operation only involves informing the EG that it is in legato mode. For mono and unison non-legato modes, RTZ is implemented so that depressing a note while another is held will shut down the existing note before starting the next, an operation that is a voice self-steal operation. This operation only involves voice-side code. In polyphonic operation, the engine gets involved as it uses a heuristic function to select a suitable voice to steal.

Bibliography

Braut, Christian. 1994. *The Musician's Guide to MIDI*, Appendix A. Alameda: SYBEX.

Sound on Sound Magazine. 1999. "Synth Secrets." http://www.soundonsound.com/sos/allsynthsecrets.htm, Accessed October 14, 2020

SynthLab Documentation. 2020. www.willpirkle.com/synthlab-docs, Accessed on October 14, 2020

5 Learning and Using the SynthLab Objects & Projects

Now that we've covered the SynthLab architecture, synth modules, cores, and their operational phases, we can learn how to use SynthLab. You have numerous options here, and you can move around easily, from testing modules in the precompiled SynthLab-DM versions to integrating small bits and pieces into your existing projects to cloning the entire engine and voice architecture and building the GUIs to go with them. Remember that SynthLab is not tied to any third-party plugin framework and does not use any third-party libraries or code as a result.

5.1 Designing Modules with the SynthLab-DM Projects

The easiest way to start using SynthLab is with the *DM* projects. In this case, you will compile a tiny *.dll* file (Windows) or *.dylib* file (MacOS), which encapsulates a single synth module – a specific type of EG, filter, LFO, or pitched oscillator. The only difference between the various *DM* synths is the quad oscillator bank, described in Section 1.5. This means that any LFO, EG, or filter module you design will instantly work as a dynamic module in any of the other synths. In the ensuing chapters, each time you are introduced to a module core, you have the ability to copy, modify, or redesign that core as you like and then test it within the SynthLab-DM project.

There are example dynamic modules with code for every type of component. Suppose you created your own wavetable oscillator module that generates drum sounds, and you named it "Drum WT DLL." You set up your tables and assign a list of your waveform strings: "Kick 1, Kick2," etc. You also have a cool idea for filtering and reversing the audio, so you implement four of your own mod knob functions, and you assign the labels. You then compile your module mini-plugin and place the result in the proper location (detailed in Section 5.2.2). When you open your DAW and instantiate the wavetable synth, your new module will appear in the list of cores along with the others, and the mod knob labels will change to reflect your ideas, as shown in Figure 5.1. This allows you to design as many modules as you like and truly customize the synth in your own unique way.

5.2 SynthLab-DM Modules Are Dynamic Linked Libraries

In order to make your own dynamic module, you will need to create a simple project in Visual Studio or Xcode that will generate a dynamic linked library (DLL) that is loaded at runtime. If you are unfamiliar with DLLs and how they relate to plugins, check out my FX plugin book for a

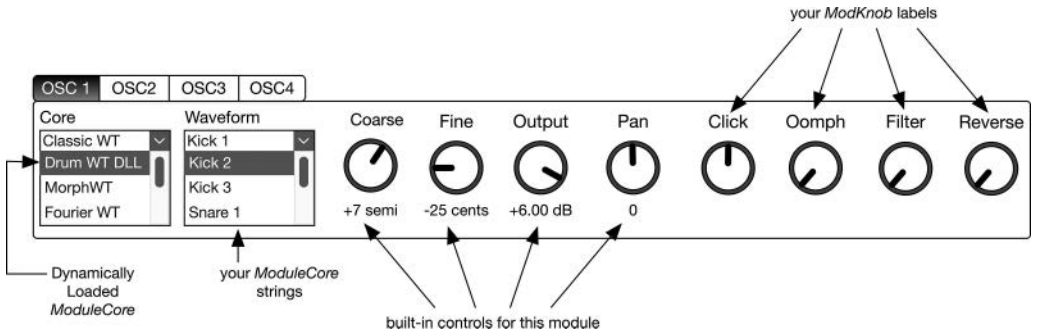


Figure 5.1 You can create internal modules for the SynthLab-DM synths without needing to recompile the entire project; this shows a module named Drum WT DLL that exposes a set of waveform and mod knob strings for the user to see

complete detailed description. The Visual Studio and Xcode DLL projects are so lean and simple, there is no need for a *CMake* script or other utility – there are less than five steps required for both OS platforms. You can find the exact step-by-step details in the SynthLab documentation at www.willpirkle.com/synthlab-docs; you will be delighted at how painless it is to set up these mini-plugin projects.

5.2.1 SynthLab-DM Modules Generate a C++ ModuleCore *Object*

In Section 3.9, you learned about the overview of the *ModuleCore* and its relationship to the outer container *SynthModule*. You also saw how the *CoreProcData* structure is used as the function argument that connects and allows safe communication across the thunk barrier between the *DM* synth and your module. The *ModuleCore* object represents the SynthLab-DM API. Your derived class overrides five functions, plus a constructor, and the DLL project creates a new instance of this object that is delivered into the SynthLab-DM synth at startup time. It then shows up as a module core, along with the others that are built into each project. You have complete access to all of the module core code, so you can freely use any existing core as a basis for your own designs. And you can test and debug your module within the DAW of your choice – the debugger will show you how the core is being loaded and used. The *ModuleCore* virtual functions are shown below.

```
class ModuleCore


// --- ModuleCore pure virtual functions
virtual bool reset(CoreProcData& processInfo) = 0;
virtual bool update(CoreProcData& processInfo) = 0;
virtual bool render(CoreProcData& processInfo) = 0;
virtual bool doNoteOn(CoreProcData& processInfo) = 0;
virtual bool doNoteOff(CoreProcData& processInfo) = 0;
```

```
// --- for EG only
virtual int32_t getState() { return -1; }
virtual bool shutdown() { return false; }
```

Your module must override and implement the functions in bold (and the two extra functions *getState* and *shutdown* for EG modules only); with that accomplishment, you have a proper module to dynamically load. To streamline the modules, I've already created your derived class named *SynthLabCore* and set up the virtual functions, which are mostly empty. You may freely cut and paste from the example core code, copy the core objects and use some text replacement, or start from scratch. The details of these functions as they apply to each synth component are given in the following chapters.

The DLL instantiation mechanism is fundamentally the same for Windows and MacOS. Your DLL module project ultimately exposes an export function that you can find in *synthlabdll.h* and *synthlabdll*.cpp, which includes code for both MacOS and Windows.
MacOS:

```
extern "C" SynthLab::ModuleCore* createModuleCore()
```

Windows:

```
#define DllExport extern "C" __declspec(dllexport)
DllExport SynthLab::ModuleCore* createModuleCore();
```

The function creates the object with the *new* operator and returns it.

```
SynthLab::ModuleCore* createModuleCore()
{
     SynthLab::ModuleCore* module = new SynthLab::SynthLabCore();
     return module;
}
```

5.2.2 Loading the C++ ModuleCore *Object*

The code to load your modules into the SynthLab-DM projects is likewise textbook DLL code for both OS platforms. Windows projects use the *LoadLibrary* function from *windows.h*, while MacOS projects use *dlopen* from *dlfcn.h*; these are heavily documented functions and operations that have been in use for a long time, so there is no need to cover them here. However, the location of your module DLL is very important as the loading functions will fail if this is not correct. Both AU and VST3 plugins are packaged in a bundle rather than a single file. A bundle is a folder that appears to the user as a file. When you download the SynthLab-DM projects, you will receive either a *.vst3* or a *.component* bundle. On Windows, you simply double click on the bundle icon to enter it, while on MacOS, you right click and use "Show Package Contents." Both bundles feature a special folder named *Resources*, located within the outer *Contents* folder. If you open this folder, you will see the *coremodules* subfolder, which will already include at least one *.dll* or *.dylib* for each kind of module. When you build your own module, you will then need to place it inside of the *Contents*/*Resources*/*coremodules* folder. When you instantiate the DM-synth in the DAW, it scans this folder and tries to load all of the libraries it finds.

5.2.3 Differentiating Your ModuleCore

You only need to write two lines of code to instruct the DM-synth on how to treat your module. At the top of each module core constructor are two statements – for example, in the *BQFilterCore* module, you will find the following:

```
SynthLabCore::SynthLabCore ()
{
      moduleType = FILTER _ MODULE;
      moduleName = "BQFilterDLL";
      etc…
```

The *moduleType* dictates the type of component, and you have eight choices that implement the eight different supported cores according to Table 5.1. The module name you expose will show up in the module core lists for that kind of component. For example, if you create a module with *moduleType* = WTO_MODULE and name it "GoldenTables," then each wavetable oscillator's GUI control will list "GoldenTables," along with the pre-compiled cores for the user to choose.

There are three SynthLab modules that do not include cores in their design patterns. The DCA object is so simple that there is no need for module cores as there isn't much variation you can add. The *ModMatrix* object is mainly a bunch of for-loops and multiplication of simple gain and intensity factors but it has a very specific way of initialization and programming. The *WaveSequencer* only generates modulation values, making it quite simple to replace with your own; however, the GUI design and programming are very different from any of the *SynthModules*, so creating cores would be cumbersome from the GUI aspect. If you want to modify these objects, you will need to integrate them directly rather than writing the simple module DLLs.

Table 5.1 SynthLab's module types, their container objects, and the downloadable examples

moduleType	*SynthModule Container*	*SynthLab Examples*
LFO_MODULE	*LFO*	*ClassicLFOCore* *FMLFOCore*
EG_MODULE	*EnvelopeGenerator*	*AnalogEG* *DXEG* *LinearEG*
FILTER_MODULE	*SynthFilter*	*VAFilterCore* *BQFilterCore*
WTO_MODULE	*WavetableOscillator*	*ClassicWTCore* *MorphWTCore* *DrumWTCore* *SFXWTCore* *FourierWTCore*
VAO_MODULE	*VAOscillator*	*VAOCore*
FMO_MODULE	*FMOscillator*	*FMOCore*
PCMO_MODULE	*PCMOscillator*	*LegacyPCMCore* *MellotronCore* *WaveSliceCore*
KSO_MODULE	*KSOscillator*	*KSOCore*

5.2.4 Testing Your Module Core

Once you have copied your core object into the bundle's resources according to the instructions in Section 5.2.2, you will want to test it; actively debugging the module will show you the sequence of operations that all cores and their respective module containers carry out and will prepare you for integrating the main C++ objects. In your compiler, set the debug executable to your favorite DAW, then attach to the running process. Place breakpoints in your constructor and in each of the five (or seven, if EG module) overridden functions it implements. Then instantiate the synth plugin, and catch the constructor in action. Once constructed, play a note, then watch the sequence of events that occurs cyclically for each note-event, covered as the operational phases in Sections 1.8.1 through 1.8.6. Make sure you can trace the behavior of the *update* and *render* functions specifically as these are the two most important for each module. For the wavetable and PCM cores, take special care to watch how the modules query the wavetable and sample databases, then install their data as required – this is likewise very important if you want to create your own oscillators, especially the morphing varieties.

5.3 Using *SynthModules* in Your Projects

The next step up from working with the *ModuleCores* is to use the *SynthModules* that aggregate these cores. All *SynthModules* feature a stand-alone mode of operation, making it simple for you to use them straightaway. There are three kinds of *SynthModules*: modulators, pitched oscillators, and processors (filters, DCA, and delay FX), and all are easy to use outside of SynthLab.

5.3.1 Modulator Objects in Stand-Alone Operation

The LFO and EG objects are modulators that write their outputs into an output modulation array and have simpler constructors than the pitched oscillators. For example, say you want to incorporate the *SynthLFO* object into your own project. You have learned that all of the synth module constructors include two similar arguments: a shared pointer to the special GUI parameter structure and another shared pointer to the MIDI input structure. All *SynthModules* will construct their own shared-pointer to their custom parameter structures if the constructor argument is a *nullptr*, then you may access the structure using a function for stand-alone mode that has the same name across all objects: *getParameters*. The *SynthModule* and *ModuleCore* base classes store a Boolean flag, *standAloneOperation*, that you may set or clear. You may find it helpful to know the stand-alone state when you aggregate objects – see the *FMOperator* and *FMOCore* for an example of using this variable.

The MIDI input data offers two choices. If you do not want or need to send MIDI data to the object, then construction is simple – just pass a *nullptr* in for the MIDI data, along with the object parameters; here, I use a shared pointer, but you could also use old-fashioned naked pointers as well. If you want to process individual samples rather than blocks, just set the *blockSize* parameter equal to 1.

```
// --- blocksize
uint32_t blockSize = 64;

//--- unique_ptr
std::unique_ptr<SynthLFO> myLFO;
```

```
// --- pass nullptrs for the parameters
myLFO.reset(new SynthLFO(nullptr, nullptr, blockSize));
```

In this case, the LFO will synthesize its own MIDI input data structure that is initialized with zeros; this will have no effect on the update phase.

If you want to send MIDI into the object, just create your own shared pointer and use it during construction:

```
std::shared_ptr<MidiInputData> midiInput =
                                        std::make_shared<MidiInputData>();

// --- create object
myLFO.reset(new SynthLFO(midiInput, nullptr, blockSize));
```

To send MIDI information, you transfer data into the proper slot in the array. To set a mod wheel CC value of 42 and set the MIDI note velocity to 79, you would write:

```
midiInputData->setCCMIDIData(MOD_WHEEL, 42);
midiInputData->setGlobalMIDIData(kCurrentMIDINoteVelocity, 79);
```

This MIDI data will be used on the next update phase of the LFO operation, and there is no set-function to call as it is automatic.

To connect the LFO to GUI controls, or to manipulate it programmatically, you get its parameter structure and set the values. As with the MIDI input data, these parameters will be applied on the next update phase, and likewise, there is no set-function to call.

```
std::shared_ptr<LFOParameters> params = myLFO->getParameters();
```

To manipulate the object directly, you set its values accordingly:

```
params->frequency_Hz = 333.333;
params->outputAmplitude= 0.707;
```

If you are interfacing to a GUI, you send in the values directly from your GUI controls using actual (not normalized) values, as shown above.

```
double lfoRate = myGUI->lfoFreq;      // in HZ!
double lfoAmplitude = myGUI->lfoAmp; // actual value!
params->frequency_Hz = lfoRate;
params->outputAmplitude= lfoAmplitude;
```

5.3.2 Accessing Modulator Outputs

The ensuing chapters will detail how the modulation values are rendered and where they are written into the module's MIDI output array. To render and then access the fresh modulation values, you write:

```
//--- render
myLFO->render(blockSize);
```

```
// --- access output
double lfoModOut = myLFO->getModulationOutput()->getModValue(kLFONor-
                   malOutput);
```

The modulation array constants (e.g. *kLFONormalOutput* here) for every SynthLab object are found in *synthlabparams.h* as well as in the future chapters.

5.3.3 Oscillator Objects in Stand-Alone Operation: VA and KS

The virtual analog (VA) and Karplus-Strong (KS) oscillators use the same construction as the EG and LFO, so these follow the same pattern as the modulators. In stand-alone mode, they will create their own MIDI input and parameter structures, and accessing them is identical to the EG and LFOs for connecting to GUI controls or direct manipulation. All pitched oscillators write their outputs to *AudioBuffer* objects.

To create a virtual analog oscillator in stand-alone mode with MIDI input, you write:

```
//--- unique_ptr
std::unique_ptr<VAOscillator> vaOsc;

std::shared_ptr<MidiInputData> midiInput =
                                        std::make_shared<MidiInputData>();

// --- pass nullptrs for the parameters
vaOsc.reset(new VAOscillator (midiInput, nullptr, blockSize));
```

And to access and manipulate parameters:

```
std::shared_ptr< VAOscParameters > params = vaOsc->getParameters();


params->outputAmplitude_dB = -3.0;
```

or from a GUI of your own design:

```
params->fineDetune = myGUI->oscFineDetune;
```

5.3.4 Oscillator Objects in Stand-Alone Operation: WT and PCM

The WT and PCM oscillators have extra arguments that are likewise shared pointers to the wavetable and PCM databases; these are described in Sections 9.3 and 11.4, respectively. Each database subclasses a base class abstract interface (*IWavetableSource*, and *IPCMSampleSource*) so that you may design your own versions in the way that makes most sense to you; these are key to efficiently sharing the tables and sample arrays. Manipulating the object via its parameter structure is done the same way as the other SynthLab components.

For example, to use the wavetable oscillator with my database object, and your MIDI input structure (assuming you want to send MIDI CC or global data to the pitched oscillator), you write:

```
// --- create shared resources
std::shared_ptr<WavetableDatabase> wavetableDatabase =
                                    std::make_shared<WavetableDatabase>();
```

```
std::shared_ptr<MidiInputData> midiInput =
                                    std::make_shared<MidiInputData>();

// --- declare
std::unique_ptr<WTOscillator> wtOsc;

// --- create
wtOsc.reset(new WTOscillator(midiInput, nullptr, wavetableDatabase,
                             blockSize));
```

For true stand-alone operation, in which each oscillator creates and owns the database, you pass a *nullptr* for the database argument:

```
// --- create
wtOsc.reset(new WTOscillator(midiInput, nullptr, nullptr, blockSize));
```

The PCM sample oscillator named *PCMOscillator* works the same way – either provide a shared *IPCMSampleSource* database or the oscillator will create and use its own.

5.3.5 Accessing Pitched Oscillator Audio Outputs

To render the pitched oscillator output and access the audio samples, you use the *getAudioBuffer* methods. Remember that all SynthLab oscillators output stereo. As with the LFO, you may choose to render individual samples or blocks of audio. To render a block of 64 samples and access the output samples in a loop, you would write:

```
// --- render 64 samples
blockSize = 64;
wtOsc->render(blockSize);

float* leftBuffer =
           wtOsc->getAudioBuffers()->getOutputBuffer(LEFT_CHANNEL);

float* rightBuffer =
           wtOsc->getAudioBuffers()->getOutputBuffer(RIGHT_CHANNEL);

for (uint32_t i = 0; i < blockSize; i++)
{
     float leftSample = leftBuffer[i];
     float rightSample = rightBuffer[i];

     // --- do something with the audio...
}
```

5.3.6 Filter Object in Stand-Alone Operation

The last category includes objects that process audio from input to output, such as the *SynthFilter*. The constructors of these objects are identical to the LFO, EG, VA oscillator, and KS oscillator, requiring the MIDI input data and custom parameter structure shared pointers, and you can

instantiate them in stand-alone operation using the same paradigms of providing the resources or letting the object manufacture its own. To create the *SynthFilter* object without MIDI input so that it manufactures its own parameter structure, and then set its parameters, you write:

```
// --- create in pure standalone form
std::unique_ptr<SynthFilter> filter;


filter.reset(new SynthFilter(nullptr, nullptr, blockSize));

// --- set the filter cutoff and Q controls
std::shared_ptr<FilterParameters> filtParams = filter->getParameters();

filtParams->fc = 1000.0;
filtParams->Q = 0.707;
```

5.3.7 Accessing Filter Audio Inputs and Outputs

A single *AudioBuffer* object includes both input and output arrays. To send a block of audio (including a block of only one sample) into the filter, you first prepare the audio buffer object using either the specialized constructor or the *init* method, which have the same arguments:

```
AudioBuffer(uint32_t _numInputChannels,
            uint32_t _numOutputChannels,
            uint32_t _blockSize);
```

For example:

```
std::unique_ptr<AudioBuffer> filterBuffers;


filterBuffers.reset(new AudioBuffer(STEREO_CHANNELS,
                                    STEREO_CHANNELS, blockSize));
```

Load up the input buffer using the same for-loop that you use to access the output buffer:

```
float* leftInBuffer = filterBuffer->getInputBuffer(LEFT_CHANNEL);
float* rightInBuffer = filterBuffer->getInputBuffer(RIGHT_CHANNEL);

for (uint32_t i = 0; i < blockSize; i++)
{
       leftInBuffer[i] = // set here...;
       rightInBuffer[i] = // set here...;
}
```

To simplify buffer copy operations, I've included several helper functions (*synthfunctions.h*) that will move buffers into and out of the objects. To write the buffer into the filter and render the output, you write:

```
// --- to Filter1
copyBufferToInput(filterBuffers,            /* pointer to source */
                  filter->getAudioBuffers(), /* pointer to dest */
```

```
                    STEREO_TO_STEREO, /* stereo in, stereo out */
                    blockSize);       /* sample count */

// --- render filter
filter.render(blockSize);
```

Then, to get the filter's processed data, you use another helper function and the same kind of loop to access the samples.

```
// --- copy filter output to buffer output
copyOutputToOutput(filter->getAudioBuffers(), /* pointer to source */
                   filterBuffers,             /* pointer to dest */
                   STEREO_TO_STEREO,
                   blockSize)
```

5.4 Using *SynthEngines* in Your Projects

Each synth has a single *SynthEngine* object that encapsulates the complete project, and you only need to instantiate that engine object, then send it blocks of buffers to render into, along with the MIDI messages that occurred during that block, and update its state with the current GUI parameters in order to create a working synth. You should not attempt this until you are comfortable working with the modules and their cores. Connecting the engine to your framework's processing object is actually quite simple and straightforward; the details lie more in the audio block processing and GUI updating than in the engine internals. These are described in detail in the SynthLab documentation (www.willpirkle.com/synthlab-docs), and therefore, this is just an overview of the operation.

For example, to implement the wavetable synthesizer, you download the folder that includes the engine, voice, and modules – this is a self-contained synth folder requiring no other libraries or files. You will usually add the engine object to your plugin framework's processor, then instantiate it, passing along the maximum block size that can occur, as detailed in Sections 2.3 and 2.4. Make sure you understand how the *SynthProcessInput* is used to provide the block processing. In your processor object, you set up the objects:

```
// --- structure for passing audio and MIDI into and out of the engine
SynthLab::SynthProcessInfo synthBlockProcInfo;


// --- the entire synth in one object
std::unique_ptr<SynthLab::SynthEngine> synthEngine = nullptr;
```

Then, set up the two objects in your processor's initialize or constructor method. The arguments to the information structure include input audio channel count (zero here as there are no audio or side-chain inputs), output channel count (stereo synth), and maximum block size:

```
// --- setup audio buffers 0 channels in, 2 channels out
//   block size = 64 (64 stereo samples)
synthBlockProcInfo.init(0, 2, 64);

// --- reset engine with max block size
synthEngine.reset(new SynthLab::SynthEngine(64));
```

5.4.1 SynthEngine *Reset and Initialize Phases*

The engine is simple to reset when the sample rate changes – just call the reset method, and pass the sample rate:

```
synthEngine->reset(resetInfo.sampleRate);
```

The engine also includes an *initialize* method that is called only once and sends the path to the plugin DLL for several objects to use, including the PCM sample database, which needs to know the location of the audio sample files. Your framework will provide you with a method to get the path to your plugin, so you simply send that path to the engine. For maximum flexibility, a simple *const char** is used for the path.

```
// --- get the DLL path from your framework
const char* dllPath = myFramework->getComponentPath( );

// --- initialize
synthEngine->initialize(dllPath);
```

5.4.2 SynthEngine *Prepare for Render*

For each block of audio you render, you need to first set up the *SynthProcessInput* structure by clearing the MIDI events, then add the information you get from your plugin framework:

```
// --- clear MIDI events at top of buffer
synthBlockProcInfo.clearMidiEvents();

synthBlockProcInfo.absoluteBufferTime_Sec = // get from your framework!
synthBlockProcInfo.BPM =  // get from your framework!
synthBlockProcInfo.timeSigNumerator = // get from your framework!
synthBlockProcInfo.timeSigDenomintor =      // get from your framework!
```

After this step, you fire off the MIDI events prior to rendering.

5.4.3 SynthEngine *Prepare MIDI Input*

As described in Section 2.5.3, you will need to prepare a *SynthProcessInfo* structure with the incoming MIDI data for rendering in blocks, as shown in Figure 2.4. This will be very framework-dependent, so you need to know how to modify your processor object's buffer processing function to operate on sub-blocks that are *blockSize* in length.

You also need to know how to get the MIDI messages that were input during that block of samples, which is likewise framework dependent. The *SynthProcessInfo* structure provides a few functions to make this operation simpler. To add MIDI messages to the *SynthProcessInfo* structure for processing, you need to fashion your framework's MIDI input data into a *midiEvent* structure, then add it to the queue with the helper function. Suppose you access a MIDI message that encodes the following:

- Message Type: Note On
- MIDI Channel 10

- MIDI Note number: 60 (middle C)
- MIDI Velocity: 127
- Sample offset: 12 (this is optional information and not required for normal operation)

In your framework's MIDI input queue or callback you create the event structure:

```
SynthLab::midiEvent synthEvent(0x90, /* note on */
                               10,    /* CH 10 */
                               60,    /* note #60 */
                               127,   /* velocity */
                               12);   /* offset (can be 0 for all) */
```

Then, use the helper function to queue the message:

```
synthBlockProcInfo.pushMidiEvent(synthEvent);
```

With the events queued, you then need to fire the events, which pushes them into a structure for the engine to use, in the same sequence that you acquired these events in your framework's MIDI callback function. You need to do this just prior to calling the engine's *render* function. Loop over the block, and fire the events:

```
// --- fire ALL MIDI events for this block
for (uint32_t sample = 0; sample < blockSize; sample++)
     processBlockInfo.midiEventQueue->fireMidiEvents(sample);
```

5.4.4 SynthEngine *Audio Rendering*

The *SynthProcessInfo* structure inherits from *AudioBuffer* and therefore carries the input and output audio sample buffers with it. *The SynthEngine does not require or accept input or output buffer pointers from your framework*. The engine will write its final synth output into the *SynthProcessInfo* structure's output buffers – you only need to copy them to your processor's output buffers. To render the audio, you first set the number of samples to process in the information structure. When you initialized the engine and the processing structure, you set the maximum samples per block. However, it is possible that your framework or DAW will deliver a buffer that does not divide into *blockSize* pieces, leaving a partial buffer to fill. Just pass the partial buffer, and set the block size, and the engine will process and fill it like the other blocks.

```
// --- in case of partial block
synthBlockProcInfo.setSamplesInBlock(64); //<- equal/less than max-size

// --- render it
synthEngine->render(synthBlockProcInfo);
```

The final step is to write the audio to your framework's output buffer; you may use a simple loop or a *memcpy* function. In the code below, *output_buffer* is your framework's audio output buffer pointer, and the *startSample* and *blockSize* variables are part of the sub-block processing loop that you maintain. Remember that we set up the input and output channel count (2) when we initialized the processing structure in Section 5.4.2.

```
// --- output
float** synthOutputs = synthBlockProcInfo.getOutputBuffers();

// --- block processing -- write to outputs
for (uint32_t sample = startSample, i = 0;
     sample < startSample + blockSize;
     sample++, i++)
{
     // --- copy to outputs
     for (uint32_t channel = 0; channel < 2; channel++)
     {
          output_buffer[channel][sample] = synthOutputs[channel][i];
     }
}
```

5.5 *SynthEngine* GUI Design and Parameter Update

To use the engine effectively, you will need to design the GUI and expose the parameters according to your plugin framework. All modern DAWs will provide a default GUI if you need it, but you will still need to set up the plugin parameters according to your framework. Each SynthLab project includes documentation on the GUI parameter setup, including recommended minimum, maximum, and default values, along with the type of GUI control – continuous, numerical controls or string-list-based controls. You may also examine the pre-compiled SynthLab-DM projects to see my various GUIs to use as a basis for your own and to better understand the core-module controls.

5.5.1 SynthLab's Dynamic Strings

Referring back to Chapter 3 and Figure 3.4, recall that each *SynthModule* exposes up to 4 *ModuleCores*, each of which exposes up to 16 core strings that the user sees as waveform names, filter types, and EG contours. Note that both the maximum core and module string counts are simple to change to increase or decrease the size; you may also set the maximum number of cores to one if you wish, greatly simplifying operation. When the user selects a new core from your GUI control, its module strings and mod knob labels are dynamically loaded into the appropriate locations on your GUI. This is an advanced GUI design technique that is going to be highly framework-specific, so you will need to have a strategy for setting up these GUI controls and populating them as required. There are several approaches here, and with the stock projects that you download, all of the module names, core strings, and mod knob labels are known a-priori. To populate these controls, you have two basic options: generate a master list from the documentation that you store and retrieve as the user selects different cores or query the engine for these strings as you need them during real-time operation.

5.5.1.1 Getting Core Names at Runtime

Each module features two dynamic GUI string list controls for the core and its associated strings, and four assignable mod knobs that are initially labeled A, B, C, and D. When the user loads a new module core, you need to update the module strings (16 of them) and the four labels that adorn

the mod knobs. To query the engine at any time for a list of the four module core names that are exposed for a given type of module, use the *getModuleCoreNames* function that returns a vector of the four strings.

```
vector<string> getModuleCoreNames(uint32_t moduleType)
```

The *moduleType* argument is the module type code listed in Table 5.1. To get the four core names for the LFO module, you write:

```
vector<string> cores = synthEngine->getModuleCoreNames(LFO_MODULE);
```

5.5.1.2 Getting Module Strings and Mod Knob Labels at Runtime

To use dynamic string loading for the 16 module strings and 4 mod knob labels, you call two similar functions that return vectors of the strings. A mask value indicates the exact target. To get modules strings for the LFO2 waveforms and the FILTER1 types at runtime, you write:

```
vector<string> lfowaves = synthEngine->getModuleStrings(LFO2_WAVEFORMS);

vector<string> fltypes = synthEngine->getModuleStrings(FILTER1_TYPES);
```

To get the mod knob string labels for oscillator #3 and the filter EG (which is EG #2 in all projects), you write:

```
vector<string> o3MK = synthEngine->getModKnobStrings(OSC3_MOD_KNOBS);

vector<string> fegMK = synthEngine->getModKnobStrings(EG2_MOD_KNOBS);
```

You can then populate your controls with these strings.

5.5.2 Setting Synth Parameters from GUI Controls

You will want to update the GUI parameters on the *SynthEngine* prior to each of your framework's buffer processing function calls. You may also place the update calls on block-boundaries; all of the synth modules and objects follow this pattern. The engine's parameter structure contains GUI control variables for the engine (global tuning, delay FX, etc.) along with a shared voice parameter pointer. These are detailed in Chapters 2 and 3, along with the shared data scheme in Section 2.3.

5.5.2.1 GUI Control Example: SynthLFO

Each SynthLab project includes documentation on the GUI parameter setup. The engine, voice, and modules all include custom parameter structures that are described in this book and detailed in the online documentation. These structures are used to pass GUI control information into the objects. Generally speaking, there is a one-to-one correspondence with the structure member variables and your GUI controls, though you do not need to expose every control. Each of these parameter structures includes a special array of four normalized values that correspond to the four mod knobs A – D. As an example, let's examine the custom parameter structure for the *SynthLFO*

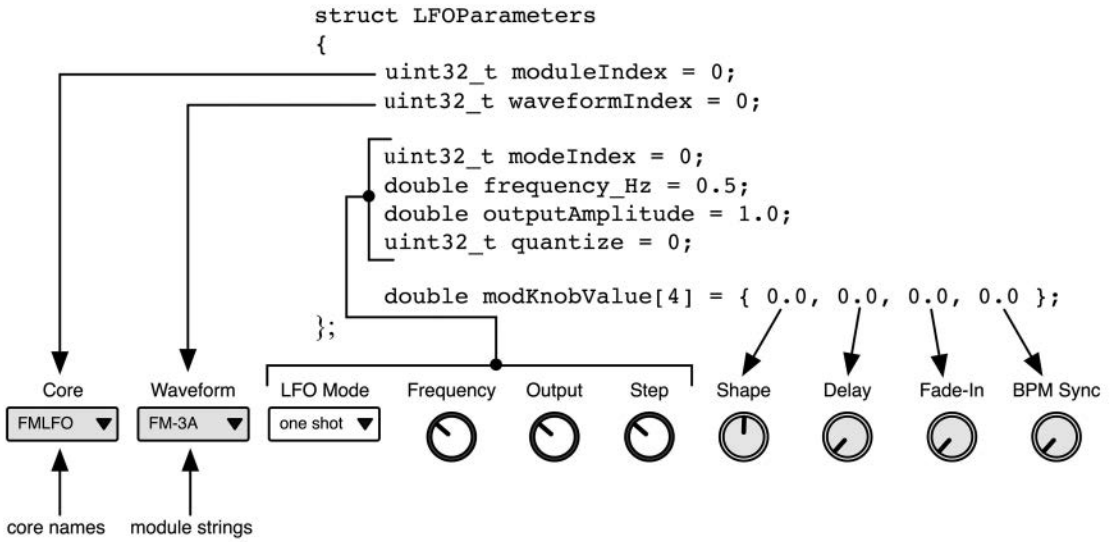


Figure 5.2 GUI controls and their relationship to the custom parameter structure; the controls in grey include the dynamic strings and are different for each kind of module and core

C++ object and compare it with a typical GUI implementation you might concoct, shown in Figure 5.2.

The *moduleIndex* and *waveFormIndex* variables are the selection values from your string list control (0 is the first item, 1 is the next, and so on). You send these integers into the module and core where they are decoded. The mod knob strings are normalized on the range [0, 1], and mod knob A always defaults to the center position. If Figure 5.2 represents connections to LFO2, you would update its variables using the engine and voice sub-parameters shown below, which are also discussed in Sections 2.6.2 and 3.3:

```
// --- declare and get shared pointer
shared_ptr<SynthLab::SynthEngineParameters> engineParameters;

synthEngine->getParameters(engineParameters);
```

The engine's shared voice parameters are then used to update LFO2. In the code below, *myGUI* represents the GUI structure from which you parse your controls, which is also up to you and your framework.

```
shared_ptr<SynthLab::SynthVoiceParameters>voiceParams =
                                        engineParameters->voiceParameters;


// --- LFO2 core and waveforms
voiceParams->lfo2Parameters->moduleIndex = myGUI->lfo2_coreindex;
```

```
voiceParams->lfo2Parameters->waveformIndex = myGUI->lfo2_waveform;

// --- normal parameters
voiceParams->lfo2Parameters->modeIndex = myGUI->lfo2_mode;
voiceParams->lfo2Parameters->frequency_Hz = myGUI->lfo2_frequency_Hz;
voiceParams->lfo2Parameters->outputAmplitude = myGUI->lfo2_outputAmp;
voiceParams->lfo2Parameters->quantize = myGUI->lfo2_quantize;

// --- mod knob parameters
voiceParams->lfo2Parameters->modKnobValue[0] = myGUI->lfo2_ModKnobA;
voiceParams->lfo2Parameters->modKnobValue[1] = myGUI->lfo2_ModKnobB;
voiceParams->lfo2Parameters->modKnobValue[2] = myGUI->lfo2_ModKnobC;
voiceParams->lfo2Parameters->modKnobValue[3] = myGUI->lfo2_ModKnobD;
```

Refer to the SynthLab documentation for complete examples that parse all GUI parameters with more detail on integrating the objects into your projects.

5.6 Programming the Modulation Matrix

The modulation matrix programming happens in the voice object's constructor. In addition, modulation routings may be added or removed at run-time; this is also needed when the user selects a new core to point the modulation matrix at the new object. These details are covered in Chapter 14 (modulation matrix), and you should examine the specific code in the voice object that loads new cores (e.g. *loadLFO1*, *loadFilter2*, etc.) to see how to remove and add new modulation routings. For this coding, the simplest option is to examine the sample code and notice how the routings are made, including the hardwired routings, such as *filterEG* to *filter1's* f_c value.

5.7 Getting *WaveSequencer* Status Meter Updates

The *WaveSequencer* object outputs status meter values to indicate which step is active for a given lane. Your processor may query the engine at any time to get an array of values that indicate an active (1) or inactive (0) status for any of the steps. The arrays return the status of the wave sequencer in the first voice that was triggered; you may modify this to return arrays for all voices, but this may be very confusing for the user. There are four sequencer lanes named *timing*, *wave*, *pitch*, and *step sequencer*, each of which has eight lane steps. The query functions are easily decoded using the lane name and step number:

```
// --- status for wave step 0
wave0 = engineParameters->wsStatusMeters.waveLaneMeter[0];

// --- status for step sequencer step 5
ssMod5 = engineParameters->wsStatusMeters.stepSeqLaneMeter[5];
```

Bibliography

SynthLab Documentation. 2020. www.willpirkle.com/synthlab-docs, Accessed on October 14, 2020

6 Modulation

Theory and Calculations

Modulation is fundamental to synthesis, and there are numerous sources and destinations for modulation values. The typical modulation sources include low frequency oscillators (LFOs), envelope generators (EGs), MIDI CCs, velocity, and note number. The output values are applied to modulate or change the destination's parameters. Typical destinations are pitched oscillator frequency, output amplitude, shape and filter f_c, Q, and drive. It is also possible for one modulator to modulate another: for example, one LFO may modulate the other LFO's oscillation frequency. Modulators have two general output types: unipolar on a range of [0, 1] and bipolar on a range of [−1, +1]. A modulation routing is a specific pair of modulation source and destination.

Some modulation calculations operate on the simple unipolar or bipolar values. Others require scaling or mapping the modulator to some new range: for example, a range of filter f_c values from 20 Hz to 18 kHz. Be sure to check out the online documentation, which lists numerous helper functions and simple C++ objects that will do much of the labor for our modulations.

6.1 SynthLab Mod Knob Mapping

All of the SynthLab core objects implement one to four functions that are controlled, with the four Mod Knobs included in every core. Some of these are already coded, and some are left as exercises for you. All cores expose the four mod knob controls, which are named A, B, C, and D. When you add strings to the *coreData.modKnobStrings*, they are exposed for the user in the GUI control. The four knobs are each unipolar in nature and transmit values on the range [0.0, 1.0]. The first mod knob (A) has its default in the center position (0.5), while the other three default to 0.0, so choose wisely when customizing these controls. For example, in the synth filter cores, the constructor sets up the four GUI strings; note that the "EG Int" and "BP Int" controls stand for "EG Intensity" and "Bipolar Intensity," respectively. Most modulation routings are controlled, with some kind of intensity value that may be unipolar or bipolar, and the filter has specialized intensity controls for each.

```
// --- modulation control knobs
coreData.modKnobStrings[MOD_KNOB_A] = "Key Track";
coreData.modKnobStrings[MOD_KNOB_B] = "Drive";
coreData.modKnobStrings[MOD_KNOB_C] = "EG Int";
coreData.modKnobStrings[MOD_KNOB_D] = "BP Int";
```

Since the mod knobs generate unipolar values, you will usually need to convert these to some other range of values. There are three helper functions to allow you to map these controls easily.

Each function accepts the normalized mod knob value, then maps it to a range that you set with the *min* and *max* parameters, and returns this value as a *double* data type.

```
getModKnobValueLinear(double normalizedValue, double min, double max)

getModKnobValueLog(double normalizedValue, double min, double max)

getModKnobValueAntiLog(double normalizedValue, double min, double max)
```

For example, the filter's "Drive" control needs to map to a linear range of 1.0 to 10.0, and this is easily accomplished as:

```
filterDrive = getModKnobValueLinear(
                  parameters->modKnobValue[MOD_KNOB_B], 1.0, 10.0);
```

The three functions map linear, log, or anti-logarithmically, and allow you maximum ease in dealing with the incoming mod knob values.

6.2 MMA Transforms & Calculations

The MIDI Manufacturer's Association (MMA) published two documents called "DLS Level 1" and "DLS Level 2," which are specifications for two PCM sample-based software synths that used downloadable sounds (DLS) for samples. These documents are fantastic as they show two different software synth specifications, with Level 1 being the simpler of the two. The MMA documents several transforms that are used to convert lines into curves and are based on maintaining curvature between 7 and 14-bit control signals. These include *concave* and *convex* transforms that are complementary in nature. These are easily adapted to work with floating point values on both input and output ranges of [0.0, 1.0], as shown in Figure 6.1(a) and (b), and they are simple to make bipolar by mirroring each half across the *x*-axis. The simple equations for these adapted transforms are:

$$\text{Concave} \qquad \text{Convex}$$

$$y=\begin{cases} 1.0 & x \geq 1.0 \\ -\left(\frac{5}{12}\right)\log(1-x) & 0.0 \geq x \geq 1.0 \\ 0.0 & x \leq 0.0 \end{cases} \qquad y=\begin{cases} 1.0 & x \geq 1.0 \\ 1+\left(\frac{5}{12}\right)\log(x) & 0.0 \geq x \geq 1.0 \\ 0.0 & x \leq 0.0 \end{cases} \tag{6.1}$$

The 5/12 factor comes from the −96 dB lower limit for 16-bit digital audio and causes a tiny error near the 0.0 and 1.0 values. The functions in *synthfunctions.h* include a few more correction coefficients that I adapted from a similar transform which I use for logarithmic taper GUI controls and which performs a perfect mapping along the input/output ranges of [0, 1].

The MMA also defined an attenuation calculation for converting MIDI velocity on the range of [0, 127] into an attenuation value in dB, with −96 dB as the theoretical lower limit; this is shown in Figure 6.1(c). This transform is also used for converting linear MIDI CC 7 (volume) and CC 11 (expression) data into a curved form. This calculation is based off of a simple square law on the MIDI value, and we may convert the MIDI velocity into a direct scalar, bypassing the

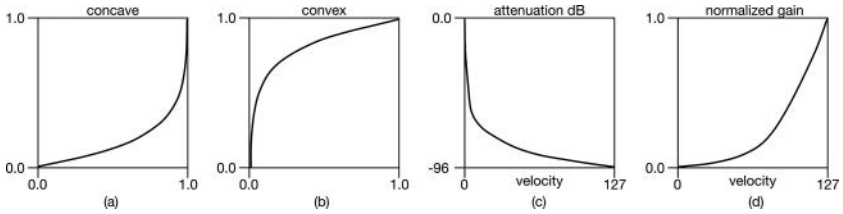


Figure 6.1 (a) MMA concave and (b) MMA convex transforms; (c) MMA velocity to attenuation in dB and (d) square law converted velocity to normalized gain graphs

log function altogether with the square law equation producing the familiar $y = x^2$ curve shown in Figure 6.1(d):

$$atten_{dB} = 20\log\left(\frac{127^2}{vel^2}\right)$$
$$gain = \frac{vel^2}{127^2} \tag{6.2}$$

Since the convex and concave transforms involve a *log* operation, I've also created lookup table (LUT) versions for each, and the function will allow you to choose the method. The MMA velocity to normalized gain uses a simple square law, but you can experiment with this. Many synths offer multiple velocity curves for the users. The function prototypes of all these functions are below, and each has an LUT option that is disabled by default. Their implementations follow the equations and are simple and straightforward. There are also reverse (mirror image) versions that are used specifically in the DX EG object.

```
double concaveXForm(double xn, bool useLUT = false)
double bipolarConcaveXForm(double xn, bool useLUT = false)
double convexXForm(double xn, bool useLUT = false)
double bipolarConvexXForm(double xn, bool useLUT = false)
double mmaMIDItoAtten(uint32_t midiValue)
```

6.2.1 MIDI Velocity to Attack and Note Number to Decay Scaling

A MIDI modulation for the envelope generator that is useful when applied to the output DCA involves scaling the attack time with the velocity such that the attack time gets shorter as the velocity increases. Another common MIDI modulation involves scaling the note number to the decay time such that the decay gets shorter as the note number increases, similar to the way a piano note's envelope changes as you play in the higher register. These are both applied in the EG core object and may be enabled in the SynthLab synths. The calculation is simple and creates a scalar value that is multiplied against the attack or decay time during the note-on event, so this modulation only happens during note-on.

$$scalar_{VEL} = 1 - \frac{\text{velocity}}{127}$$
$$scalar_{NOTE} = 1 - \frac{\text{note number}}{127} \tag{6.3}$$

6.2.2 Constant Power Panning and Crossfading

Most SynthLab oscillators include a panning control to place the signal in the stereo field. The GUI control is set to move from −1 to +1 as the user pans from left to right, with zero as the center value. To get the imaging correct, you need to apply complementary constant power curves to generate the left and right channel. For crossfading between waveforms, as used in the morphing oscillators, the same principle applies to make the crossfading sound smooth, though, in some cases, linear crossfading may work better. For panning, the MMA DLS spec uses trigonometric curves with one quadrant each of a cosine and sine function; these are shown in the solid lines in Figure 6.2(a). In the center position, both functions evaluate to 0.707, which is exactly −3 dB. Alternatively, a square power law equation may be used, as shown in the dotted lines in Figure 6.2(a). At the center point, each function evaluates to 0.75. For panning, the *x*-axis moves from −1 to +1 as the user moves the panning knob clockwise from the hard-left position. For crossfading, the *x*-axis represents the crossfade time. Figure 6.2(b) shows the *XFader* C++ object that performs the crossfade operation on two input signals. This object allows you to choose between linear or either of the two constant power equations. It also allows you to set two crossfade times, one for each input; as shown, the crossfade times are identical. The constant power equations are given in Equation 6.4 for both panning (left/right) and crossfading (A/B).

Trigonometric	Square Law
$A_{(LEFT)} = \cos\left(\frac{\pi}{4}(x+1)\right)$	$A_{(LEFT)} = 1 - 0.25(x+1)^2$
$B_{(RIGHT)} = \sin\left(\frac{\pi}{4}(x+1)\right)$	$B_{(RIGHT)} = 1 - (0.5x - 0.5)^2$

(6.4)

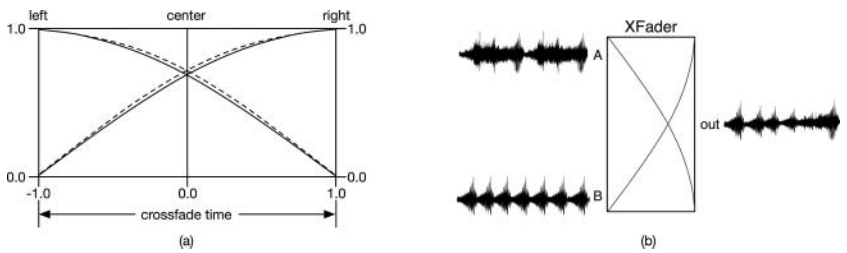


Figure 6.2 (a) Constant power curves for panning and crossfading with solid line for trigonometric and dotted line for square law transforms, and (b) the *XFader* C++ object crossfades between two signals, A and B, using linear, trig, or square law crossfading

6.3 Sequential Circuits Vector Joystick Envelope Modulation

The Sequential Circuits Prophet VS® introduced a new kind of envelope modulation for crossfading between either two or four different audio signals with a "vector joystick" known as vector synthesis. The Korg Wavestation® and Wavestate® synths include this modulation option as well. In essence, the user controls the relative mixes of the signals with an X-Y joystick, either manually, via MIDI CCs, or by programming a virtual path for the joystick to take during the note-event. Figure 6.3(a) shows the joystick as it appears on the synths, in a diamond shape. The four vertices marked A–D represent the four sources to be blended. The joystick's (x, y) coordinates are converted into four scalar values, which are used to attenuate each signal before mixing. The four values always add up to exactly 1.0, and when the joystick is in the center position, the values are all equal at 0.25.

Examining the Korg equations [Phillips], it is evident that the physical joystick unit was rotated 45 degrees clockwise prior to its mounting in the synth enclosure such that the true x and y-axes lie at 45-degree angles, shown with the dotted lines in Figure 6.3(a). For soft synth GUIs, it is simpler to get the true (x, y) coordinates by undoing the rotation, as shown in Figure 6.3(b), which places the x- and y-axes in the standard orientation. The vector mix equations use the square law functions, which are modified to produce the four envelope curves shown in Figure 6.3(c), in which the curves are linearized for easier viewing.

The equations are designed to accept MIDI CC values, and to keep the code identical to the original, you need to scale out the (x, y) coordinates to [0, 255] and rotate the incoming (x, y) point counterclockwise prior to computation to match the rotated original. Equation 6.5 combines the point rotation with the MIDI scaling of the (x, y) coordinates into (x_M, y_M) and then generates the vector multipliers as percentages that are easily converted into normalized values.

$$x_M = 127\left(\frac{x-y}{2}+1\right) \quad y_M = 127\left(\frac{x+y}{2}+1\right)$$

Coefficients as percentages [0, 100%]

$$\begin{aligned} B &= \frac{x_M y_M}{645} \\ C &= \frac{x_M (255 - y_M)}{645} \\ D &= \frac{(255 - x_M)(255 - y_M)}{645} \\ A &= 100 - B - C - D \end{aligned} \tag{6.5}$$

In addition to the four multipliers, two more potential modulation values are calculated: these are the x- and y-axis reflections of the current joystick location, called the AC and BD axis values. The function *calculateVectorMix* generates the A–D coefficients, along with the unipolar AC and BD reflection values. A structure is used to return the data. Notice that the joystick origin may also be adjusted.

```
struct VectorXFadeData
{
       double vectorA = 0.25;
       double vectorB = 0.25;
```

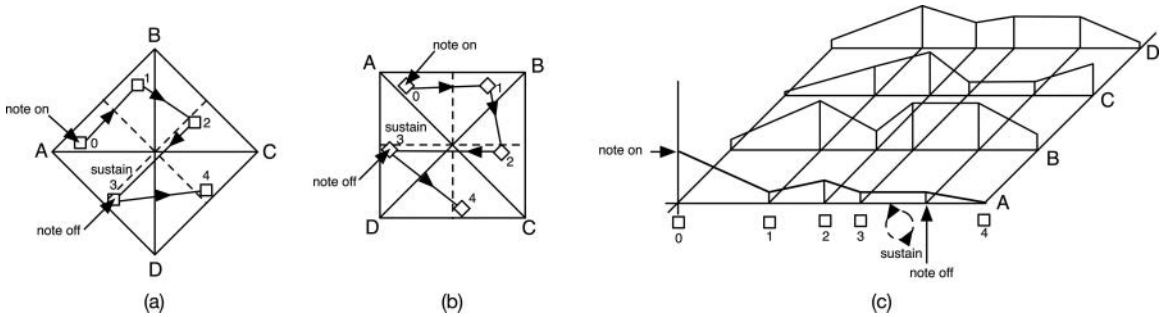


Figure 6.3 (a) The original Sequential/Korg vector joystick; (b) the rotated version is simpler for GUIs (the dotted lines show the actual *x*- and *y*-axes); and (c) visualization of the four envelopes that are generated as the stick moves from points 1 to 3, where it stays during the sustain period, and note-off moves the joystick to its final position 4

```
        double vectorC = 0.25;
        double vectorD = 0.25;

        // --- unipolar
        double vectorAC = 0.5;
        double vectorBD = 0.5;
};

VectorXFadeData calculateVectorMix(double joystick_X,
                                   double joystick_Y,
                                   double origin_X = 0.0,
                                   double origin_Y = 0.0)
```

6.4 Linear Frequency Modulation with LFO or EG

In the most basic version of FM, you simply add or subtract a value from the underlying frequency variable, then recalculate the new clock *phaseInc*. SynthLab uses bipolar modulation for linear FM, so first, you calculate the range and half-range of the modulation limits linearly. The reason for saving the full range is that you may wish to use a unipolar modulator such as the *AmpEG* in your variations. FCMOD_MIN is the minimum oscillator frequency, and FCMOD_MAX is the maximum value.

```
const double RANGE = ((FCMOD_MAX - FCMOD_MIN));
const double HALF_RANGE = RANGE / 2.0;
```

To implement bipolar modulation, you multiply a bipolar value by the half-range of the modulation limits. Linear frequency modulation means adding a frequency offset (positive or negative) to the underlying frequency value.

$$\begin{aligned} \text{mod} &= biPolarValue * HALF\_RANGE \\ f_{OSC} &= f_{OSC} + \text{mod} \end{aligned} \tag{6.6}$$

The bipolar FM modulation value is always found in the module's *modulationInputs* array:

```
double mod = processInfo.modulationInputs[kFrequencyMod] * HALF_RANGE;
```

To apply the modulation, you simply add the half-range scaled value and then bound the value:

```
double newFrequency_Hz = parameters->frequency_Hz + mod;
boundValue(newFrequency_Hz, FCMOD_MIN, FCMOD_MAX);
```

At this point, you will use the *newFrequency_Hz* to alter the clock's *phaseInc*; then, the modulation is complete.

```
lfoClock.setFrequency(newFrequency_Hz, sampleRate);
```

6.5 BPM Sync

Synchronizing LFOs and delay FX to BPM is common today, and all DAWs and plugin frameworks have the ability to send the plugin BPM and time signature information. You can calculate timing values from the BPM and set the LFO frequency, or other time-based parameter, using a note duration, such as eighth note triplets. SynthLab features two different functions: one for the LFO and the other for the Wavestate wave-sequencer. The LFO function accepts a mod knob value from [0.0, 1.0] and quantizes it into eight integers that are used as index values in a lookup table. Moving the control from minimum to maximum steps through the sequence of note durations: whole note, half note, dotted quarter, quarter note, dotted 8th, 8th note, dotted 16th, and 16th note. The function returns the note duration in seconds:

```
double getTimeFromTempo(double BPM, double normalizedNoteMult)


// --- SynthLab code
double bpmSync = getTimeFromTempo(processInfo.midiData->auxData[kBPM],
                                  parameters->modKnobValue[MOD_KNOB_D]

if(bpmSync > 0.0)
   parameters->frequency_Hz = 1.0 / bpmSync;
```

The extended function that is used with the *WaveSequencer* object implements more note durations with a larger lookup table and returns the note duration in seconds or milliseconds. The function supports 16 different rhythmic values: 32nd note triplet, 32nd note, 16th note triplet, dotted 32nd note, 16th note, 8th triplet note, dotted 16th note, 8th note, quarter note triplet, dotted 8th note, quarter note, half note triplet, dotted quarter note, half note, whole note triplet, dotted half note, whole note, and dotted whole note. A strongly typed *enum* is used to send the note duration, and a static lookup table of multipliers is applied.

```
enum class NoteDuration { k32ndTriplet, k32nd, k16thTriplet, kDot32nd,
                          . . . kWhole, kDotWhole,

double getTimeFromTempo(double BPM, NoteDuration duration,
                        bool returnMilliseconds = false);
```

Both functions use the same core code, which simply converts the BPM into the corresponding duration for a quarter note in milliseconds or seconds, then uses a lookup table to scale the duration accordingly.

6.6 Quantizing

Several oscillators may implement step quantization on the output signal. Use the *quantizeBipolarValue* method, in which two mapping functions quantize the signal and remap it as a double value.

```
double quantizeBipolarValue(double d, uint32_t qLevels)
{
        uint32_t u = mapDoubleToUINT(d, -1.0 ,1.0 ,0 , qLevels, true);
        return mapUINTToDouble(u, 0, quantLevels, -1.0, 1.0);
}
```

To use the function, choose the quantization levels directly, or calculate to mimic bit-depth reduction:

```
// --- 12 steps
outputValue = quantizeBipolarValue(outputValue, 12);

// --- simulate 10-bit audio
outputValue = quantizeBipolarValue(outputValue, pow(2.0, 10));
```

6.7 Ramp Modulation: Fade-in and Fade-out

The simple *RampModulator* produces a linearly increasing or decreasing output depending on the start and end values. It may be used to ramp up or down between any kinds of values – amplitudes, frequencies, etc. Fade-in and fade-out are common uses of this object. Portamento is a special circumstance and is handled with the *GlideModulator* instead. To set up the object, you give it the required information for counting time intervals.

For a fade-in modulation, start and stop are 0.0 and 1.0; here, the fade-in time is 1 second = 1000.0 mSec, and we supply the normal *processInfo.sampleRate*.

```
RampModulator fadeInModTor;


// ---    (start, stop, ramp time(mSec), sample rate)
fadeInModTor.startModulator(0.0, 1.0, 1000.0, processInfo.sampleRate);
```

For a fade-out modulation, start and stop are 1.0 and 0.0; here, the fade-in time is 5 seconds = 5000.0 mSec:

```
RampModulator fadeOutModTor;

// ---      (start, stop, ramp time(mSec), sample rate)
fadeOutModTor.startModulator(1.0, 0.0, 5000.0, processInfo.sampleRate);
```

To apply the fade-in/fade-out, you use two functions: one to get the modulation (ramp) value and the other to increment the clock. For block-processed modules like the LFO, you advance the clock by the block size; otherwise, you advance it by 1.0 (the default).

```
// --- get modulation multiplier
fadeInModValue = fadeInModTor.getNextModulationValue();
// --- bump the clock; here is for a LFO (block processed)

fadeInModulator.advanceClock(processInfo.samplesToProcess);
```

The resulting modulation value is a simple scalar multiplier that you apply to a signal. For the LFO output, you might write:

```
lfoOutput *= fadeInModValue;
```

6.7.1 Glide Modulation: Portamento

Portamento or glide modulation is a special case of pitch modulation. The user plays a note, which sounds as usual. Then, the next note is triggered, and the pitch glides up or down to the new note over a time-span the user selects. Each successive note glides up or down smoothly to the final pitch. Portamento may be applied as a linear shift or as a shift in semitones, as done in SynthLab. In addition, it may be applied with a constant time (e.g. 1000 mSec between each note, no matter how close or far apart they might be) or a constant rate (e.g. 1000 mSec/octave). The THX® constant time portamento chord is a famous example which plays in many theaters before a movie starts.

The *GlideModulator* object is a specialized version of the *RampModulator* that is designed to glide in semitones using the two MIDI note numbers, as described in the next section. All *ModuleCore* objects have a *GlideModulator* attached as a member, so it is ready to use without much work. This modulation is calculated on a per-block basis, as with the other low frequency modulators (LFOs and EGs). Using the *GlideModulator* only requires three function calls. The first initializes and starts the modulation; notice that the start and end MIDI notes are set as *double* data-types, and the glide time is in milliseconds. This function calculates and stores the distance between notes in semitones, named the *glideRange*, and an increment (step) value called *timerInc*.

```
bool startModulator(double startNote, double endNote,
                    double glideTime_mSec, double sampleRate);
```

To get the next glide modulation value, which is a pitch shift in semitones between the notes, you call the *getNextModulationValue* function, which implements a simple countdown timer to create the output.

```
double getNextModulationValue(uint32_t advanceClock = 1)
{
       double output = 0.0;
       if (timerActive)
       {
           // --- output
           output = countDownTimer*glideRange;
```

```
            countDownTimer -= advanceClock * timerInc;

            if (countDownTimer <= 0.0)
                  timerActive = false;
        }
        return output;
}
```

The internal glide clock is advanced automatically by one tick during this function and may be advanced independently with a function. After each render call, we will need to advance the clock by the block size to ensure that the timing is correct.

```
void advanceClock(uint32_t ticks);
```

6.8 Pitch Calculation

Each MIDI note represents one semitone's musical distance. To calculate the number of semitones between two MIDI notes, you simply subtract the two note numbers. The equal temperament tuning equation calculates the oscillator pitch from a MIDI note number and a tuning reference for A4 (usually 440 *Hz*). Note also that the value 69 is the MIDI note number for A4.

$$pitch = (A4\_freq)2^{\left(\frac{NoteNum-69}{12}\right)} \tag{6.7}$$

To reverse the calculation and find a MIDI note number from a pitch in Hz, use Equation 6.2 and truncate the *NoteNum* value to an integer, which is the MIDI note number.

$$NoteNum = ceil\left(12 * \ln\left(\frac{pitch}{A4\_freq}\right) + 69\right) \tag{6.8}$$

6.8.1 Oscillator update *Methods: Pitch Modulation*

All SynthLab oscillators implement an *update* function that is nearly identical because the majority of the function involves pitch modulation. The pitch shift calculation creates a scalar value that you multiply with the MIDI note pitch value in *Hz*. The equation shows the relationship between semitones and pitch. Some prefer to implement the equation in cents rather than semitones – there is no mathematical difference in the end result; however you do need to formulate all shifts as either semitones or cents.

$$\begin{aligned} pitchshift_{semitones} &= 2^{\left(\frac{pitchModSemitones}{12}\right)} \\ pitchshift_{cents} &= 2^{\left(\frac{pitchModCents}{1200}\right)} \end{aligned} \tag{6.9}$$

For our pitched oscillators, we will have numerous pitch shifts, which all need to be applied together. The good news is that you may simply add all the values in semitones (or cents), then only call the pitch shifting function once. You should memorize the relationships: 1 octave = 12 semitones and 1 semitone = 100 cents.

For pitched oscillators, we expose the following GUI controls and modulation inputs:

GUI Controls:

- octaveDetune
- coarseDetune (semitones)
- fineDetune (cents)
- unisonDetune (cents)

Modulation:

- bipolar pitch modulation (vibrato)
- glide modulation (portamento)
- MIDI pitch bend
- Global Tuning

The code for combining the modulation and GUI values, and calculating the pitch modulation value is shown below. First, we define the range of modulation in semitones. I chose one octave up and down; you may modify this value as you like.

```
const double kOscBipolarModRangeSemitones = 12.0;
```

The bipolar modulation value is scaled with the range and the intensity knob control.

```
double freqMod = processInfo.modulationInputs[kBipolarMod] *
                                                kOscBipolarModRangeSemitones;
```

The glide modulation is handled with a separate object that does the calculation in semitones. Notice that there is a secondary pitch shift function that uses a lookup table rather than a power of two operation; it is commented out, and you can freely switch between the two functions to evaluate.

```
// --- do the portamento
double glideMod = glideModulator.getNextModulationValue();

// --- combine all sources in semitones
double currentPitchModSemitones = glideMod + fmodInput +
                                  midiPitchBend + masterTuning +
                                  parameters->octaveDetune * 12) +
                                  parameters->coarseDetune) +
                                  parameters->fineDetune / 100.0) +
                                  parameters->unisonDetune / 100.0);

// --- lookup the pitch shift modifier (fraction)
//double pitchShift = pitchShiftTableLookup(currentPitchModSemitones);

// --- direct calculation version 2^(n/12) - equal temperament
double pitchShift = pow(2.0, currentPitchModSemitones / 12.0);
```

```
// --- calculate the modulated pitch value
double oscillatorFrequency = midiPitch*pitchShift;
```

The wavetable and virtual analog oscillators may be used for FM synthesis, in which case they must be able to generate negative frequencies and must be bounded to +/-Nyquist.

```
// --- BOUND the value to our range
boundValue(oscillatorFrequency, (-sampleRate/2.0), (+sampleRate/2.0);
```

The PCM sample and plucked string oscillators are not used for FM and are bounded to the lowest MIDI note (*OSC_FMIN* = ~8 Hz) and *OSC_FMAX* = 20,480 Hz. I use 20,480 as a top value in numerous objects; when the lower limit is set to 20 Hz, this produces ten octaves of pitch change, which has a connection to analog synths that used a 10V control voltage (CV) to modulate across ten octave ranges.

```
// --- BOUND the value to our range
boundValue(oscillatorFrequency, OSC_FMIN, OSC_FMAX);
```

At this point, we have the newly updated oscillator frequency. For the wavetable and virtual analog oscillators, this value is used to update the oscillator clock object that acts as the time-base. See Chapters 11 and 13 for the PCM sample and plucked string oscillator details.

```
// --- phase inc = fo/fs
oscClock.setFrequency(oscillatorFrequency, sampleRate);
```

6.9 Pulse-Width Modulation (PWM)

PWM is a bipolar modulation that operates on the pulse width (or duty cycle) of a square wave and usually modulates over half ½ the normal range: for example, from 5% to 50% or from 50% to 95%. This is due to the fact that the two half-range modulations sound identical to the ear. The *VAOscillator* is the only SynthLab oscillator that implements PWM. This is accomplished by simply adding half of the bipolar pulse width modulation (PWM) amplitude to the user's pulse width GUI control value. You can find an example in Section 10.9.4.

6.10 Phase Distortion

Invented and patented at the Casio Corporation, Phase Distortion (PD) is roughly in the same family as frequency modulation (FM) and phase modulation (PM) synths. A single index of modulation is used to modify the spectral content of the signal. In its simplest sense, PD alters the shape of the modulo counter's ramp, which results in a distorted waveform with a modified spectrum. In phase distortion, you speed up and slow down the time-base by changing the slope of the modulo counter. You might also imagine the modulo counter, which moves from 0.0 to 1.0, as being a phase value from 0.0 to 2π radians. The modulo counter wraps around at a point corresponding to the waveform's period *T*, which you may think of as the *x*-axis, as shown in Figure 6.4(a), with a slope of $T/2\pi$ which is normalized to 1.0. In Figure 6.4(b), with a breakpoint placed at (0.5, 0.5), a pure sinusoid is generated. Figure 6.4(c) and (d) show how the counter output splits into two different slopes when the breakpoint is moved from the (0.5, 0.5) location.

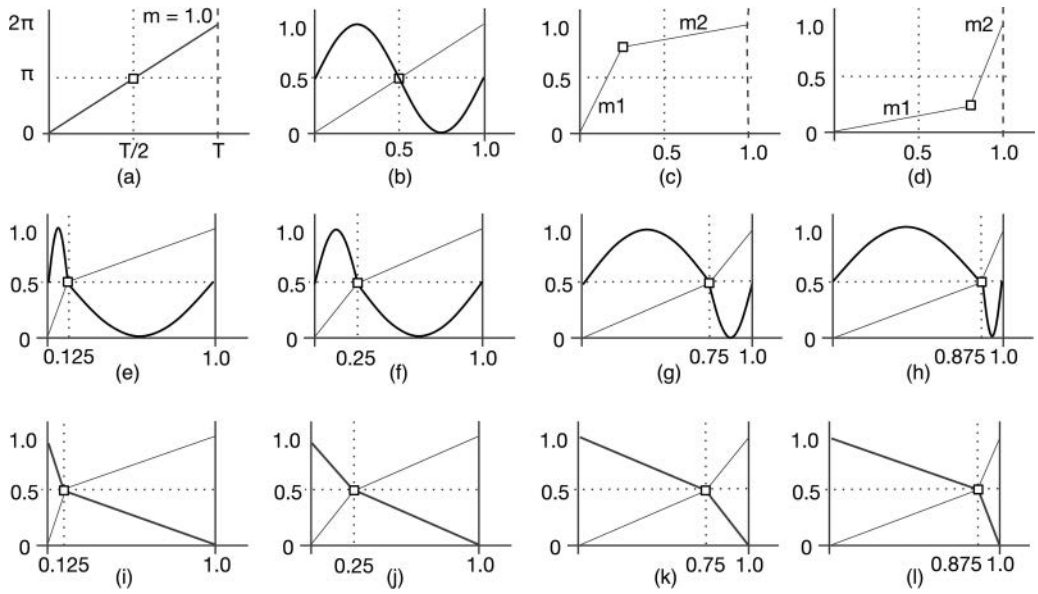


Figure 6.4 (a) The modulo counter with *x*-axis as period and *y*-axis as phase with a slope of 1.0, (b) a normal sinusoid lookup, (c) and (d) relocation of the breakpoint produces two slopes *m1* and *m2* while (e)–(f) show the distortion on a sinusoid and (i)–(l) show a ramp modified with phase distortion

In the most simple variation, you fix the *y*-breakpoint value at 0.5, which corresponds to the half-cycle point for most waveforms. Then you adjust the *x*-breakpoint value to apply the phase distortion as the modulo counter moves slower than normal (shallow slope) or faster than normal (steep slope). The distortion effects on a sinusoid and ramp waveform are shown in Figure 6.4(e)–(l).

The key to getting the proper result is understanding that the two-slope counter must always wrap around at the correct point so that the next cycle starts on the proper location of the first slope, *m1*. For single breakpoint phase distortion with an arbitrary breakpoint (x_{break}, y_{break}) producing two slopes *m1* and *m2*, the distorted modulo counter may be found with Equation 6.1.

$$distortedModCounter = \begin{cases} mcounter \le x_{break} & (m1)(mcounter) \\ mcounter > x_{break} & (m2)(mcounter - x_{break}) + y_{break} \end{cases} \tag{6.10}$$

You may also experiment with multiple breakpoints, or you may place discontinuities within the phase distortion curve. Aliasing can and will occur with phase distortion, and other options may be used to smooth over sharp breaks between the segment slopes. The SynthLab phase distortion function performs this simple one-breakpoint process. Modulating the shape value in time, which is the x_{break} variable here, can produce patches with a very nice sound.

```
double applyPhaseDistortion(double mcounter, double x_break,
                            double y_break = 0.5)
```

```
{
      // --- limit to [0.1, 0.9] to prevent jagged slopes/aliasing
      mapDoubleValue(x_break, 0.0, 1.0, 0.1, 0.9);

      // --- calc 2 slopes
      double m1 = y_break / x_break;
      double m2 = (1.0 - y_break) / (1.0 - x_break);

      // --- can happen if breakpoint is (0.5, 0.5)
      if (m2 == m1) return mcounter;
      // --- apply distortion
      if (mcounter <= x_break)
            return m1*mcounter;
      else
            return (mcounter - x_break)*m2 + y_break;

      return mcounter;
}
```

6.11 Hard Sync

Hard sync is an old analog modulation type that was originally implemented with two oscillators. In much of the literature and industry jargon, these are named master and slave. In this text, they are named *reset oscillator* and *output oscillator*. The output oscillator is what the user hears and runs at a higher frequency than the reset oscillator, which runs at the MIDI note pitch. Each time the reset oscillator begins a new cycle, the output oscillator is reset to start over. Interestingly, this means that the reset oscillator's waveform is inconsequential as we only use its period to adjust the output oscillator. Figure 6.5(a) shows the hard sync sawtooth waveform, while Figure 6.5(b) shows the result with square waves, in which the hard sync acts as a duty cycle control. In the most basic hard sync case, the main oscillator is reset back to its starting point, which is −1.0 for Figure 6.5. Another option is to allow the user to adjust the reset point to begin the next cycle at a different value (or starting phase), as shown in Figure 6.5(c) and (d).

Once again, something that is simple in analog is full of problems when attempted digitally. Clearly, the discontinuity that is created will have aliased components, if not mitigated. For the virtual analog oscillators, this is possible to implement while rendering the main oscillator's waveform directly, though it is not simple. A different approach is employed in SynthLab for both the wavetable and the virtual analog oscillators; this involves crossfading the original main oscillator and the newly reset oscillator's two outputs for a short period of time, thereby smearing over the discontinuity in an attempt to lowpass filter it. This is shown in an exaggerated manner in Figure 6.6(a) for normal reset and in Figure 6.6(b) for resetting to a non-zero starting phase. This also means that for the duration of the crossfade window, we will need to have two oscillators (or modulo counters) running: one for the original signal and the other for the reset version. Another option is to center the crossfade window on the discontinuity, which requires synthesizing a portion of the waveform below −1.0.

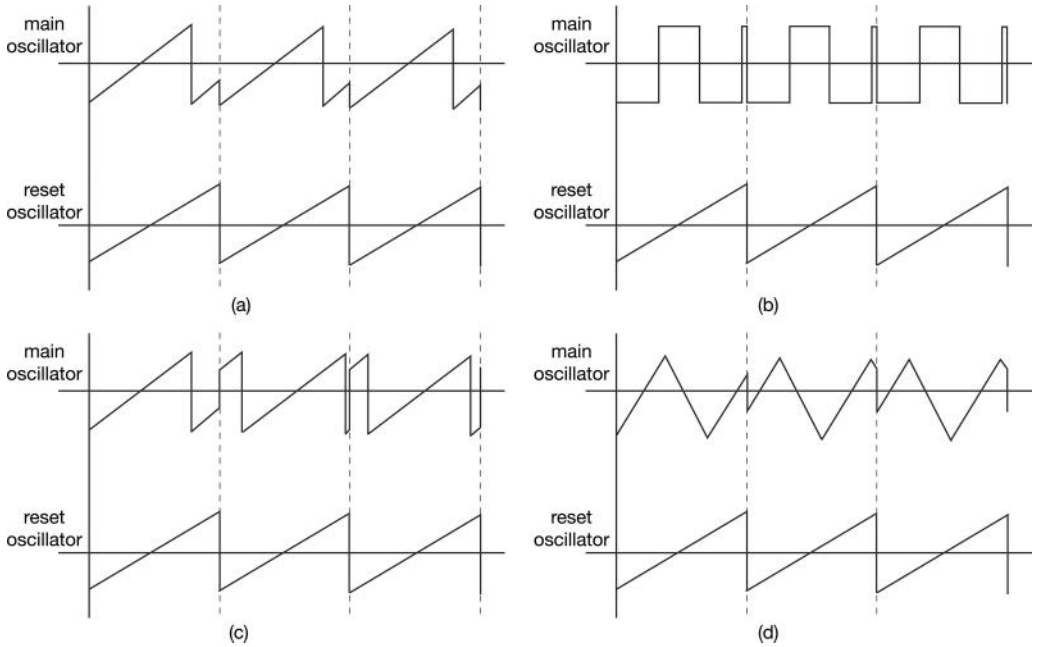


Figure 6.5 Hard sync with (a) ramp/sawtooth and (b) square-wave main oscillators, while (c) and (d) show the effect of moving the reset point for ramp and triangle waves; notice that the reset oscillator's waveform is not used other than as a resetting device

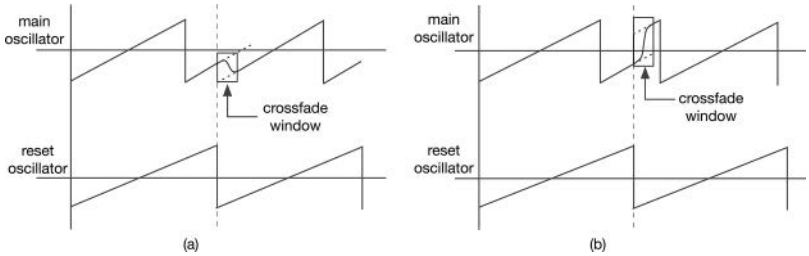


Figure 6.6 Crossfading to smear over a hard-sync discontinuity for (a) the normal reset and (b) reset to a non-zero starting phase

6.11.1 Hard Sync with a Virtual Primary Oscillator

Since the waveform of the primary oscillator is unimportant – only the reset signal is needed – in theory, we could hard sync an oscillator with an internal primary oscillator's time-base or, in other words, an invisible virtual primary oscillator that is grouped with the normal oscillator object. This is already implemented in the wavetable oscillator cores. The only GUI control required is the hard sync ratio control that operates on the range of [1.0, 4.0] – you may experiment with higher ratio values if you like as the mapping is easy to change.

To facilitate this operation, a hard sync C++ object called *Synchronizer* is supplied. This object contains two *SynthClock* objects, one for the virtual (imaginary) primary oscillator and the other for timing the crossfade, along with an *XFader* object for performing the crossfade operation. When the hard sync ratio is greater than one, the *Synchronizer* object's hard sync primary time-base is set to the new multiple of the current oscillator frequency:

```
// --- get the hard sync ratio, map to [1, 4]
hardSyncRatio = getModKnobValueLinear(
                          parameters->modKnobValue[MOD_KNOB_B],1.0, 4.0);

// --- set primary timebase
hardSyncronizer.setHardSyncFrequency(
                                        oscillatorFrequency* hardSyncRatio);
```

You can check out the *Synchronizer* operation by examining the *WTOCore::renderHardSync-Sample* function, which is called whenever the hard sync ratio is greater than 1.0. This function performs the following:

1 Check to see if primary clock has wrapped, indicating a new hard sync phase: if not wrapped, it calls the *renderSample* function as usual but uses the primary clock as the time-base
2 If a new hard sync phase starts, the crossfade clock begins, and the *Synchronizer* object performs the crossfade on the old output waveform and the newly synchronized waveform to blur over the discontinuity
3 When the crossfade is done, the function goes back to step 1 and starts monitoring the primary oscillator clock.

6.12 Filter Key Track Modulation

When filter key tracking is enabled, the filter's f_c value is set to match the MIDI pitch of the note that is playing or a frequency that is mathematically related to the MIDI pitch. For example, you may set the key track amount to +7 semitones, which would place the filter's f_c value a perfect fifth above the MIDI pitch. If the filter is self-oscillating, then the effect is one of harmonization. The modulation is simple to implement as it is a MIDI override of the user's f_c setting. There are numerous other options with this kind of modulation, including tracking slopes, center MIDI note values, and other advanced options. Key-track modulation usually requires two controls – one to enable the key-tracking and another that sets the distance either as a ratio or as an absolute offset, e.g. in semitones. The filter core objects include a dedicated on/off switch to enable/disable the function, and a mod knob that acts as a key-track offset control, ranging from −48 to +48 semitones or +/− four octaves. The key-track modulation is simple: when enabled, disregard the current GUI setting for the filter f_c, and replace it with the MIDI note pitch. Then offset this value in semitones using the normal pitch, shifting in semitones from Section 6.7. After that, update the filters in the normal manner. You can find an example in the filter core's *update* method:

```
// --- key tracking
if (parameters->enableKeyTrack)
{
```

```
        // --- get semitone offset
        ktFmodSemitones = getModKnobValueLinear(
              parameters->modKnobValue[FLT_KEYTRACK], -48.0, +48.0);

        // --- reset current fc back to MIDI pitch
        fc = midiPitch;
}
```

The filter receives bipolar and unipolar modulation values from the LFO and EG sources, then the modulations are summed as normal, and the newly modulated filter f_c value is set:

```
// --- sum modulations
double fcModSSemis = bpFmodSemitones + egFmodSemitones +
                     ktFmodSemitones;

// --- multiply by pitch shift factor
fc *= pow(2.0, fcModSSemis / 12.0);
```

Bibliography

danphillips.com. "Wavestation Vector Mix Calculation." http://www.danphillips.com/wavestation/SYSEX/WSDevDoc.zip, Accessed October 14, 2020

Dodge, Charles & Jerse, Thomas. 1997. *Computer Music Synthesis, Composition and Performance*, Chap. 4. New York: Schirmer.

Junglieb, Stanley. 1986. *Prophet VS Digital Vector Synthesizer*. San Jose: Sequential Circuits, Inc.

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 1*. https://www.midi.org/specifications-old/item/dls-level-1-specification, Accessed October 14, 2020

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 2*. https://www.midi.org/specifications-old/item/dls-level-2-specification, Accessed October 14, 2020

Phillips, Dan. 1991. *Wavestation SR Reference Guide*. Tokyo: Korg Inc.

7 Envelope Generators and DCA

The Envelope Generator (EG) and Digitally Controlled Amplifier (DCA) are two common synth components found in most synth architectures. These two modules generate time-based amplitude change values. The EG renders a unipolar output value, which is applied as a modulation value to other modules. The DCA processes an input with gain and panning (left/right) multipliers; the SynthLab DCA processes stereo signals.

7.1 Envelope Generator Fundamentals

EGs are most commonly used to control the overall amplitude of the synthesized note-event – the time domain contour that the musician hears. Most EGs are documented with something like Figure 7.1(a), which shows an EG with attack, decay, sustain, and release (ADSR) segments. Some figures show the ADSR segments as lines, but they are usually curves; this is based not only on the early analog EG circuitry but also on how we perceive loudness. Some of the figures in this chapter show the EG with linear segments to make the figures simpler and aid in understanding, but remember that they will usually be curved in an exponential manner.

The attack segment is triggered from the note-on message, and the release segment is triggered from the note-off message. The message may be MIDI or a control voltage. There are several ways in which to specify the EG parameters. Figure 7.1(b) shows an EG that uses levels and rates, which is how the Yamaha DX7 EG is configured. The rate is simply the slope of the segment. Notice that the start level (L0) does not necessarily need to begin at 0.0, and the end level (L3) does not need to decay all the way to 0.0.

In most synths, the segments are specified as times, often in *mSec*. For the attack segment, it is the time from the start level (usually 0.0) to the maximum amplitude level (usually 1.0), and this makes sense. There is no sustain time setting, but there is a sustain level setting. The decay and release times, however, are not specified the way you might think.

A fundamental ramification of the decay and release times being referenced to full-scale is that the sustain level will alter the actual (perceived) decay and release times, as shown in Figure 7.1(d), in which the sustain level has been lowered, but the decay and release times have not changed.

> The decay and release times are not the actual times you hear but the times it would take for the EG to decay or release from a full scale value of 1.0 all the way down to 0.0, as shown in Figure 7.1(c). This is independent of the curvature of the segments themselves.

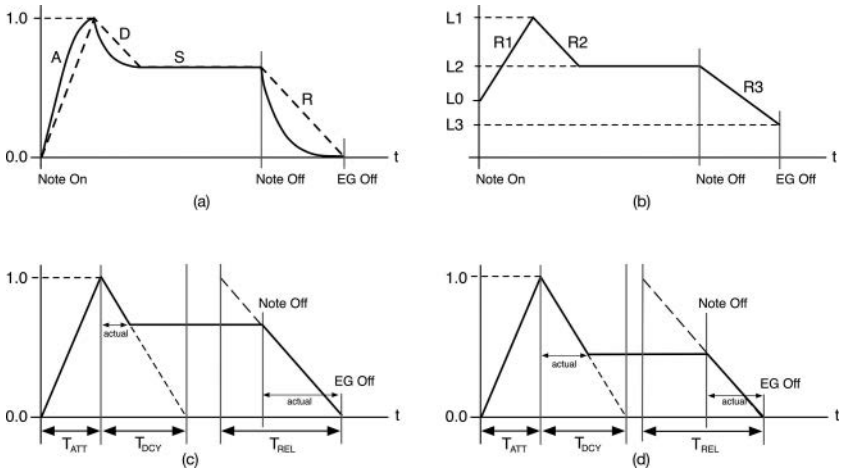


Figure 7.1 (a) ADSR envelope generator with linear (dashed) and exponential segments (b) specifying the EG using levels (L) and rates (R), and (c) the decay and release times are actually calculated from the full-scale value to zero and (d) changing the sustain level with all else constant changes the perceived decay and release times

In this case, you can see that the actual decay time the musician perceives has stretched a bit, while the actual release time has shrunk. If you think about it, this does make sense if the EG is controlling the overall amplitude of the note-event. For example, if the release time were fixed at, say, 1.0 second, then a loud note with a high sustain level and a quiet note with a low sustain level would have the same release time of 1.0 second, and that would not make sense to a musician playing the instrument. Playing quiet notes would have the same release times as loud ones. If you've played a piano, you know that is not the case: softly played notes decay faster because there isn't as much energy in the signal.

7.1.1 EG Contours

All of the EGs in Figure 7.1 consist of four segments, which are also called states: attack, decay, sustain, and release. There are several other variations that are useful in different applications. These include:

- AR: attack and release only, no sustain, as in Figure 7.2(a); the note may release and end long before the musician releases the key
- AHR: attack, hold and release; the hold portion is flat and usually at the maximum amplitude, as in Figure 7.2(b)
- DELAY: in some cases, such as in Figure 7.2(c), the attack state is delayed for some time before starting; this adds a "D" to the acronym, such as DAHR, and is simple to implement with a delay timer
- ADSLSR: attack, decay, slope, sustain, and release; the slope segment may move in the upwards or downwards direction – this is the SynthLab DX-EG object, which is similar to the EG used in the DX7; Figure 7.2(d) shows an upward slope version – notice that the slope segment is linear

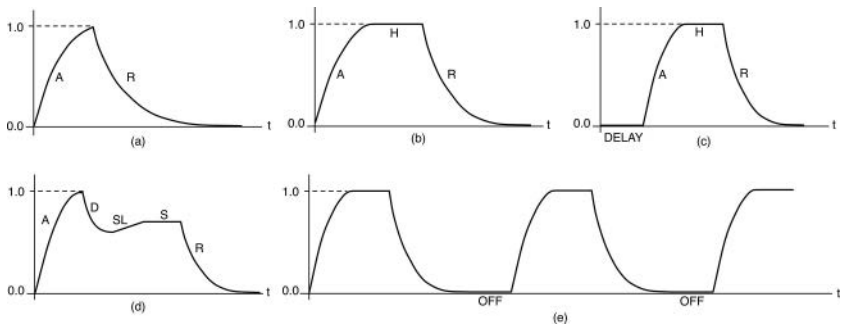


Figure 7.2 More EG contours including (a) AR, (b) AHR, (c) DAHR, (d) the SynthLab *DXEG* with linear slope segment (SL), and (e) the VCS3 re-triggering "trapezoidal" EG

- OFF: the VCS3 has an interesting "trapezoidal" (AHR) EG that includes an off-state and off-time; if the user sets the off-time to the maximum value, the EG behaves as normal, releases to zero, and is finished, but if the off-time is *less* than the maximum value, the EG stays off for that period of time, then retriggers itself for another sweep through the states, creating a self-retriggering EG; this is shown in Figure 7.2(e)

Other variations also exist, such as a double-attack EG (ADADSR), which can be useful in mimicking the lip-buzz attack of horn instruments, or the Casio CZ EG, which implements nine segments, including a double-release state after the note-off event occurs. The EG is rendered using a simple finite state machine, and it is easy to add or remove states or skip over states so that one algorithm may implement many of the contour variations.

7.1.2 EG Triggering and Note-On

EGs may be used as modulators for the various synth modules, including the output EG that controls the output amplifier and the filter EG, which modulates the filter cutoff frequency f_c. In almost all cases, the EG, no matter what its use as a modulator, is triggered to begin operation as a result of the MIDI note-on event. There are several modes of triggering and ways in which the EG can respond, dating back to early analog synths that were monophonic. When the user depresses a key, two signals are sent to the EG; these are called the trigger and gate, as shown in Figure 7.3(a). The trigger marks the note-on event, and the gate supplies the key hold time and the note-off event. Figure 7.3(b) shows an early variation in which the trigger and gate are combined into an s-trigger, which attempts to relay the same information about the note-event.

Suppose a musician plays a succession of notes, releasing each key before depressing the next. If this is done so quickly that the EG never fully releases, there are two ways in which the EG may respond. Figure 7.3(c) shows one variation in which the EG simply restarts from the last output level. Figure 7.3(d) shows another option in which the EG is forced to reset back to the starting point, even if the release period has not ended. This is called reset-to-zero (RTZ) or simply reset-mode, and it is clear that this mode will produce a more pronounced staccato sound, sometimes with audible clicks at the reset point.

7.1.3 Legato Mode

Now consider the case in which a musician plays a succession of notes but does not release each key just before depressing the next. On piano, this is called playing legato, which means that there is a smooth transition between notes as they blend into one another. Figure 7.4(a) shows the output for trigger/gate synths – for each new note-event, the EG returns to the attack state. But for s-trigger synths in Figure 7.4(b), there is no dedicated note-on signal, so the EG ignores the note-on events and never returns to the attack state, creating the legato playing style where notes blend smoothly together. On more modern synths, the s-trigger version is sometimes offered as an option called legato mode.

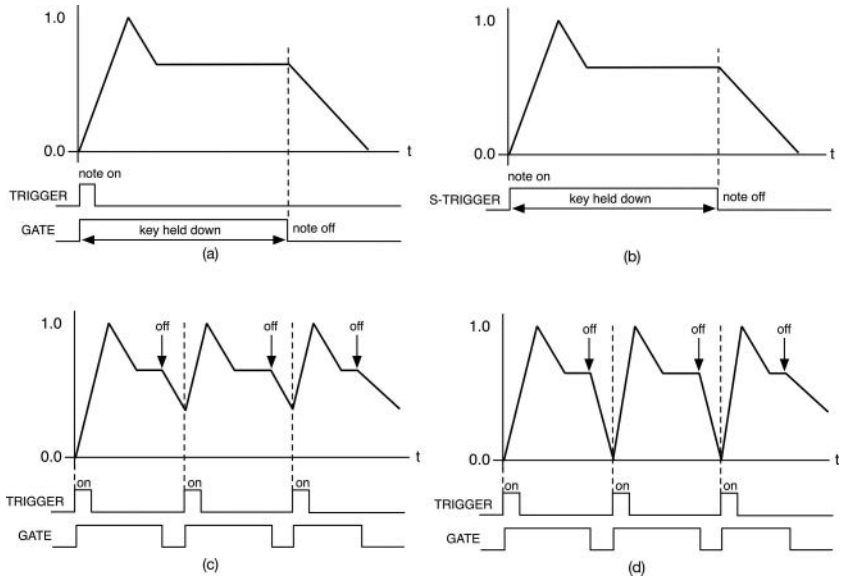


Figure 7.3 Early analog EGs with (a) trigger and gate signals, and (b) EG with s-trigger signal, (c) the EG restarts with each note on event from the previous output value and (d) the EG is reset back to the starting point with each note on event; the logic is shown as active-high but may be implemented as active-low as well

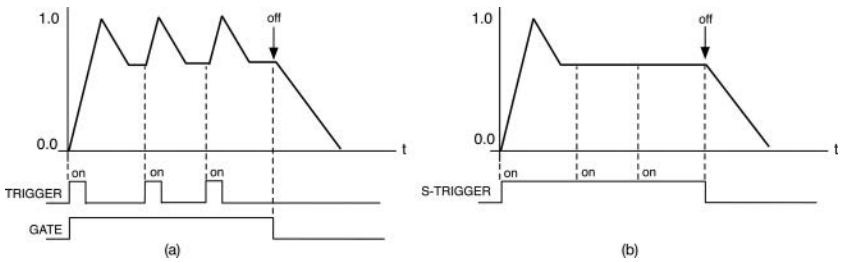


Figure 7.4 Legato playing with (a) a trigger/gate EG and (b) an s-trigger EG

7.1.4 EG Release and Note-Off

It should be evident by now that the EG and note-on and note-off events have a very close relationship. Additionally, when the EG is used to control the output amplifier, the note-event is only recognized as being complete when the EG's output value has decayed all the way to zero or its final end value, which the user may control on some EGs. This means that the EG plays a vital role in determining the state of each voice in a polyphonic synth. For a monophonic synth, the EG also has a special function with respect to legato mode operation, as outlined in Section 7.1.3. Notice that this means that the off-state needs to be included in the model.

7.1.4.1 Unconditional Release Mode

If the musician strikes a key and releases it quickly, before it has a chance to go from attack to decay to sustain, then we have a couple of options for handling this event. The majority of EGs will simply jump to the release state so the EG output decays from its value when the key is released, as shown in Figure 7.5(a) and (b) – notice that the EG never reaches peak amplitude. This behavior usually sounds natural because it is the way acoustic instruments usually function. Another option is called unconditional release mode; it is often paired with an AR or AHR EG contour. In unconditional mode, when the musician releases a key, the EG moves through all of its phases right through to the release and off phases, as shown in Figure 7.5(c). This allows a musician to hold down a chord and quickly release the notes, causing a volume swell without the need to keep the keys depressed.

7.1.5 EG Shutdown State

In addition to the four EG states of attack, decay, sustain, and release, there may be an additional state called shutdown. In shutdown mode, the EG descends from its current value right down to

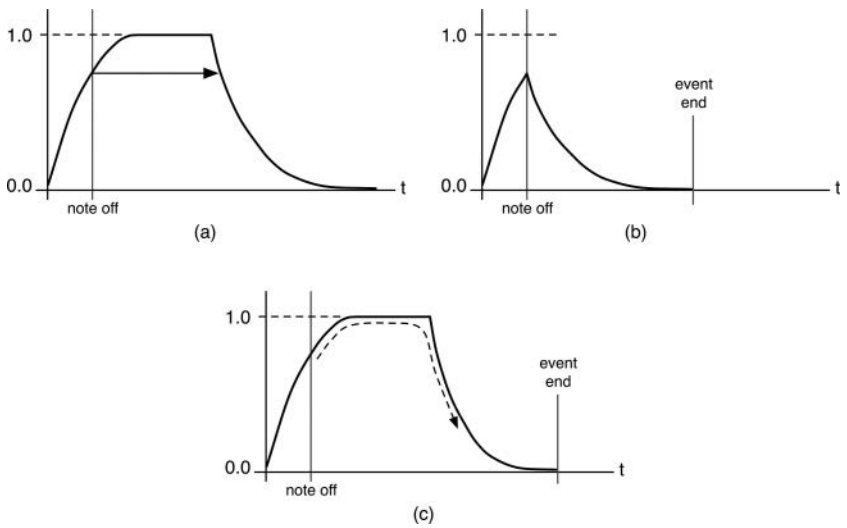


Figure 7.5 (a) The note-off event occurs before the EG reaches full amplitude and results in (b) a truncated version while (c) shows the unconditional release mode

0.0 over a very short period of time, in about a millisecond, and in a linear manner. You can see the shutdown phase in Figure 7.3(d) in reset-to-zero mode. It is also used during the voice-stealing operation to quickly shut down the voice that is being stolen. For voice-stealing, it is important that the shutdown take place fast enough that the musician does not discern a tactile response lag before the new voice and note are rendered.

7.2 EG Implementation: Finite State Machine

The ADSR EG can be modeled as a Finite State Machine (FSM) with the following states; the *DXEG* has an additional state called SLOPE that is between the decay and sustain states.

- OFF
- ATTACK
- DECAY
- SLOPE (*DXEG* only)
- SUSTAIN
- RELEASE
- SHUTDOWN

Figure 7.6 shows the state transition triggers in circles, which are either note-events or logic based on the EG output, crossing above or below some value. Figure 7.6(a) shows the standard ADSR, while Figure 7.6(b) shows one version of the *DXEG* with added slope state; in this case, the decay level L_{DCY} is below the sustain level. In the other variation, the decay level is above the sustain level, and the logic is reversed.

In C++ code, the FSM may be coded with a *switch-case* statement, a compound *if-else* statement, or a jump table if you are familiar with assembly languages. In SynthLab, the EGs are all implemented with *switch-case* statements. For example, the state transition logic from the attack to the decay state, using *switch-case* statements and a strongly typed *enum* for the state variable, would resemble the following code – note the check to see if the attack time is zero, in which case the FSM advances to the decay state, after setting the output to 1.0, which is the maximum EG output value:

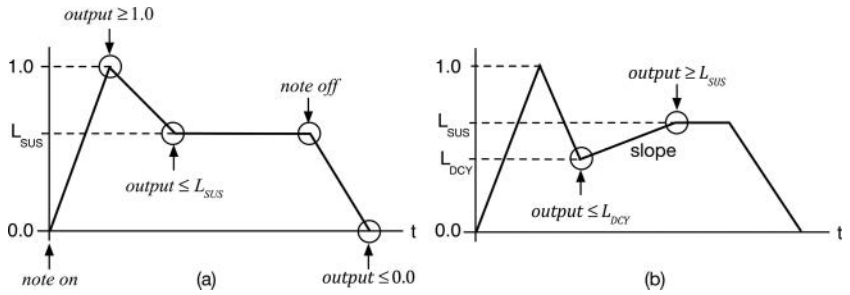


Figure 7.6 The circles show the state transition triggers for the (a) ADSR and (b) *DXEG* with added slope state; this EG may also have the decay level L_{DCY} set above the sustain level, and the transition trigger logic is inverted

```
case EGState::kAttack:
{
      // --- render value
      envelopeOutput = // <--- do the render here

      // --- check for next state
      if (envelopeOutput >= 1.0 || attackTime_mSec <= 0.0)
      {
      // --- clamp to max value
            envelopeOutput = 1.0;

            // --- go to decay state
            state = EGState::kDecay;

            break;
      }
      break;
}
```

7.2.1 EG Implementation: Note-On, Note-Off, and Shutdown

The note-on and note-off messages play a special role in the EG's state, and these set up a few rules for the FSM.

7.2.1.1 Note-On

If a new note-on message arrives while the EG is OFF, it moves to the attack state and begins the EG render cycle. If a new note-on message arrives while the EG is running, the FSM has three options depending on the mode:

1 Normal mode: the EG moves back to the attack state and begins the render cycle from the current output value
2 Legato mode: the EG ignores the new note-on message (as the s-trigger would) and continues its render cycle
3 Reset-to-zero: the EG moves into its special emergency shutdown state (see Section 7.1.5) and descends to 0.0 rapidly

7.2.1.2 Note-Off

Handling of the note-off message depends on the EG conditionality.

1 An unconditional EG will ignore the note-off message and continue the EG render cycle, ending the EG render cycle only after the release state completes.
2 All other EG types will jump to the release state and begin decaying from that value down to 0.0 (or the user's selected release-end value), ending the EG render cycle

7.2.1.3 Shutdown

The shutdown state is special in that it may or may not be used, depending on the situation. There is no state transition trigger logic for this state. Shutdown is accomplished with a dedicated EG function call and is a result of either a note-off or a note-on message.

1 Reset-to-zero (monosynth): when enabled, new note-on messages force the EG into the shutdown state to create the jagged, click-y, instant-on sound
2 Voice steal (polysynth): in the event of a voice-stealing operation, the EG will go into the shutdown state to quickly end the note-event so the voice can be stolen and begin rendering a new note

The shutdown state uses a fast linear taper – this happens so quickly that no one will notice if it is linear or not. The taper uses a simple increment value, which represents a negative step that is subtracted from the output until it decays to 0.0. Knowing the sample rate, current output value, and shutdown time in milliseconds, the increment value is found as:

$$inc_{shutdown} = \frac{-(1000.0 * envelopeOutput)}{(t_{shutdown})(f_s)} \tag{7.1}$$

So, you can see that the shutdown operation is a special case, and it is always followed by a new note-on event. All of these states, rules, and transition logics are summed up in Figure 7.7, which shows the FSM diagram for the ADSR EG, including shutdown. The dotted lines show that the shutdown message may arrive any time after the EG has been started and will always result in the same state. Likewise, the note-off message may occur at any time after the EG begins rendering and results in a jump to the release state.

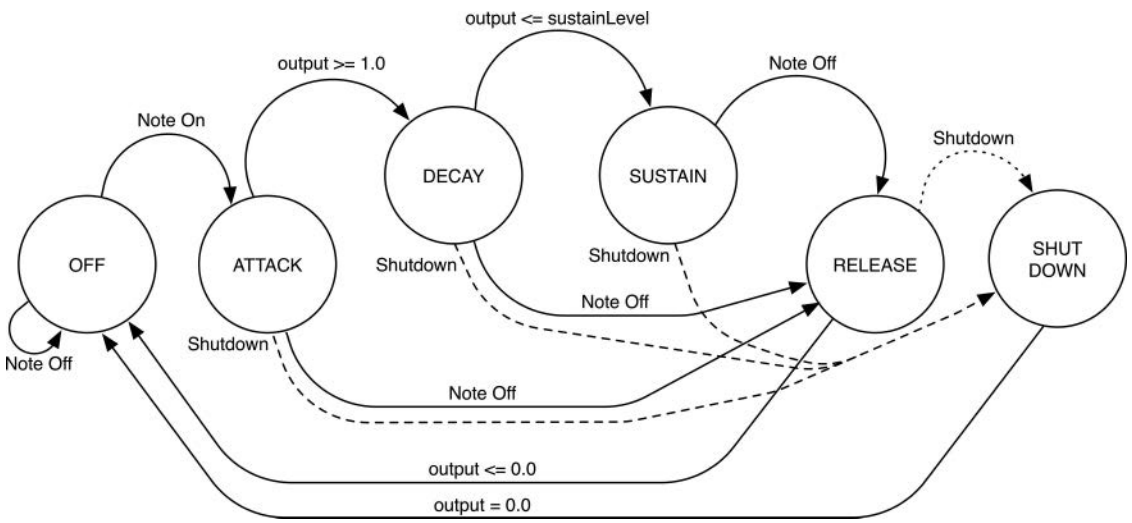


Figure 7.7 The FSM diagram for the ADSR EG with shutdown; this does not show the monosynth legato or reset-to-zero options to simplify the diagram

7.3 Digital EG Implementation: Rendering the Output

The last detail may be the most important: how do you render the lines or curves that make up the EG output with a digital system? There are several solutions, from simple to complex. I have a separate object for emulating an analog EG, so let's focus on a digital version first. Linear EGs are not especially useful for the output amplifier or the filter EG; however, they may be employed in FM operators, which use an embedded EG that is permanently connected to an oscillator's output. I use a method to convert the linear segments into exponential curves, and this allows the user to blend the linear and exponential curves together to generate any combination desired; this is the basis for the *DXEG* object, used exclusively for FM operators in SynthLab.

7.3.1 Linear and Exponential Steppers

A linear EG is easily rendered using simple counters with preset step sizes or increment values, as show in Figure 7.8(a). The counters usually move between off (0.0) and full-scale (1.0), but that is not a fixed rule; the counters may start at a non-zero value, and full scale does not necessarily need to be 1.0. In addition, some EGs allow the sustain level to be set below 0.0 as a negative number usually on the range of [−1.0, 0.0]. An exponential EG may be approximated using non-fixed counters whose increment values change by some pre-calculated amount, as shown in Figure 7.8(b). The Yamaha DX-7 EG is said to use this type of staggered counter system.

7.3.2 Linear EG

The equation that relates the attack, decay, slope, or release time in milliseconds to the step increment size is shown in Equation 7.2. This includes a *scale* variable that is +1.0 for the rising (upwards) segments and −1.0 for the downward segments (decay and release). However, you may adjust the *scale* to either expand or shrink the segment times or alter the minimum or maximum values. The DXEG features a slope segment that may move in the upward or downward direction depending on the decay level setting. Once that is known, the calculation is trivial.

$$scale = \begin{cases} 1.0 & upward \\ -1.0 & downward \end{cases}$$
$$egStepSize = scale\left(\frac{1000}{(f_s)(segTime\_mSec)}\right) \tag{7.2}$$

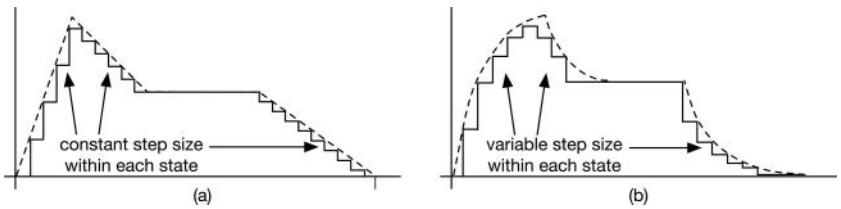


Figure 7.8 (a) Linear EG with constant step sizes within each state and (b) approximating an exponential EG with variable step sizes; the axes are unlabeled to show that the numerical limits of min and max are variable

7.3.3 Exponential EG from Linear EG

Converting the linear EG into an exponential one presents some challenges that might not be obvious at first glance. One simple option is to convert the linear values into decibel (dB) values with 0.0 dB as the maximum level – the top of the attack segment. This is easily accomplished with the standard dB equation and you can think of this as a linear to dB mapping. Now the problem is what to do with the linear EG value of 0.0 since you cannot calculate the *log* of a zero or negative value. If that value must be 0.0, then the value just before it needs to be very close to zero, or the user will hear a "step" as the EG shuts completely off. If you choose −60 dB for the last non-zero map value, the step is easily audible when used to modulate the output amp or filter. If you choose a small value like −96 dB or −120 dB, then the bottom section of the curve will be very flat and won't sound correct. Notice that slew limiting through a lowpass filter will have the same issue as well.

What we need is an exponential mapping equation that has its minimum and maximum values pinned down at 0.0 and 1.0. Fortunately, the MMA convex and concave transforms from Section 6.2 may be employed using the SynthLab functions that automatically fix the endpoints correctly – a bit more work than in the MMA documents. These functions allow you to choose to use a lookup table (LUT) instead of performing a *log10* or a *pow* function in C++ so you may experiment for the best CPU usage. The transforms and tables are designed on a range of [0.0, 1.0]. You simply call the mapping function to convert the values along the range that you need.

It is crucial that we maintain the special consideration that the decay and release times are specified from full-scale to zero, as discussed in Section 7.1, in which the sustain level plays a key role in the actual segment time perception. This means that we will only be using parts of the functions or tables, and not their entire ranges. In other words, for these segments, we need to operate on parts of the curves, not the entire curve. In order to make this work, we need a perfectly complementary function (or LUT) to ascertain the range of values so that we can sew together the segments properly. The concave and convex transforms (or tables) in Figure 7.9(a) and (b) may be used to find these values. Figure 7.9(c) shows the concept – the linear EG output is fed into either the convex or concave transform, depending on the state. The attack segment uses the convex transform, while the decay and release segments use the concave transform. Notice that the slope segment (SL) is kept linear for the *DXEGCore* that uses this method and that the attack segment does not use the entire range – it ends early, which is covered in Section 7.5 on analog EGs.

7.4 Biased EG Output

If you play an early analog synth or a faithful recreation, such as the Korg iPolysix® app, and you apply one of the EGs to the pitch modulation of the oscillators, something strange happens – the notes don't play in tune. This is because the pitch is only shifted positive from the attack state, and only goes back to 0.0 (no pitch shift) when the release state is complete. As shown in Figure 7.10(a), if the sustain level is not set to 0.0, then the pitch of the note will be sharp for the entire sustain duration of the note-event. While this might make an interesting harmonization effect with other oscillators, it is usually not exactly what is required. If you subtract from the sustain-level setting, then the EG is biased back down to 0.0 for the sustain portion, as shown in Figure 7.10(b). Here, you can see the importance of allowing a non-zero start or end point so that the note does not start or terminate flat in pitch with respect to the sustain portion, unless that is what the musician desires. The pitch and filter EGs in the Korg Wavestate® are set up this way, with minimum and maximum y-axis values of −100 to +100, or anything in between, corresponding to −1 and +1 for SynthLab's EGs.

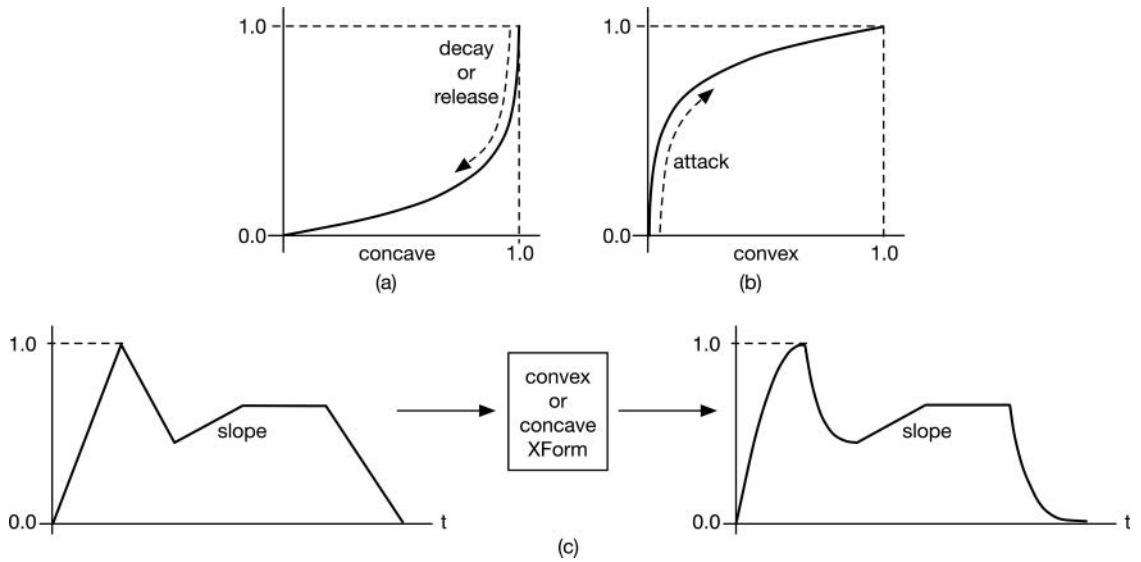


Figure 7.9 (a) The concave and (b) convex transforms are used to give contour to the linear segments, and (c) shows the overall concept; note that the slope segment is kept linear

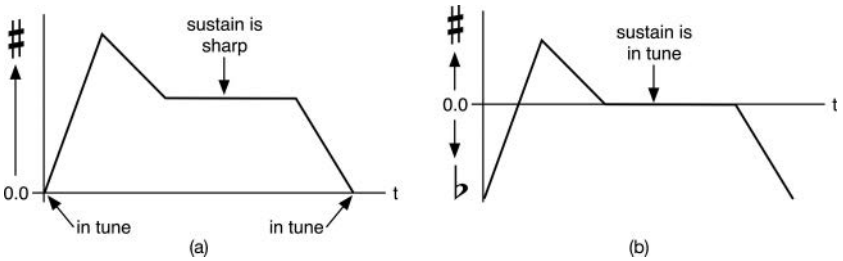


Figure 7.10 (a) When using the EG to modulate pitch, all of the portion above 0.0 is sharp, including the sustain segment, and (b) the biased EG pulls the sustain back in tune but causes the start and end of the note-event to be flat; adjustment of the start and end values mitigates this

7.5 Analog EG Emulation

The original analog EGs were always exponential in nature because they used one or more capacitors that charged for attack and discharged for decay and release. The attack, decay, and release knobs are implemented with variable resistors that set three different RC time constants, which control segment times. Ultimately, this is why the decay and release times are set from full scale to complete discharge, and it sounds correct to our ears that the sustain level influences these times. Figure 7.11 shows the underlying concept; in 7.11(a), the note-on event starts the capacitor C charging, while the note-off event causes the discharge. The *RC* time constants for ($R_{ATT}C$) and ($R_{REL}C$) set the attack and release times. The flat top that is normally formed during the charging process usually does not sound correct. In the Curtis CEM3310 EG chip (used in several older synth designs), the capacitor only charges to about 77% of the asymptote value, giving the attack

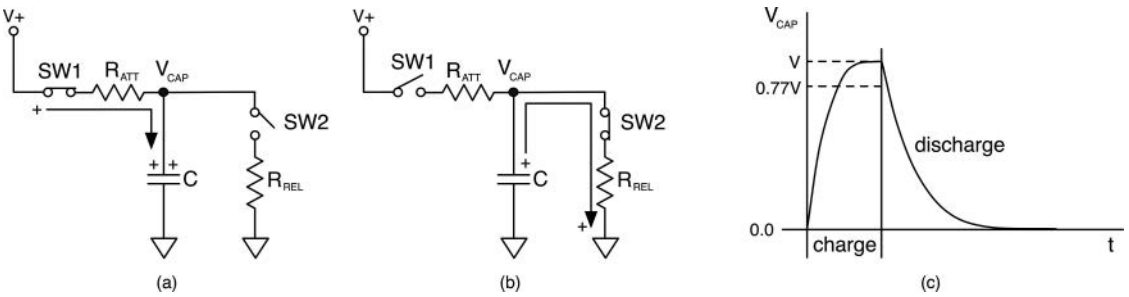


Figure 7.11 (a) SW1 closes for the note-on event, and the voltage source charges capacitor C through R_{ATT}, then (b) SW1 opens, and SW2 closes for the note-off event, which discharges the capacitor through R_{REL}; (c) shows the resulting event; note that V is the asymptotic charge voltage, and $0.77V$ is 77% of this value, used for the analog emulation

portion of the curve a more linear contour, as shown in Figure 7.11(c). More elaborate analog EG designs used multiple capacitors and resistors and electronic switches (transistors) to generate the EG voltage. Check out the schematic for the Oberheim SEM® module for an interesting two-capacitor EG circuit.

The equations for the charge and discharge time of an RC circuit are shown in Equation (7.3).

$$\begin{array}{ll} \text{charge} & \text{discharge} \\ Q = 1 - e^{-t/RC} & e^{-t/RC} \end{array} \tag{7.3}$$

The RC term is called the time-constant and named τ. The capacitor's charge/discharge time is related to the time constant. A real-world non-ideal capacitor charges to about 98% at the time $t = 5\tau$. It takes the same amount of time to discharge to 2%. To model the CEM3310, we'll only let the capacitor charge to 77% and discharge from there as well. It charges to 77% of the total at the approximate time $t = 1.5\tau$. It discharges in about $t = 4.95\tau$. Normalizing the R and C values to 1.0, we obtain the Equation (7.4) for a 77% capacitor charge/discharge over a range $x = [0, 1]$ and $y = [0, 1]$.

$$\begin{array}{ll} \text{charge} & \text{discharge} \\ y = \dfrac{1}{0.77}\left(1 - e^{-1.5x}\right) & y = e^{-4.95x} \end{array} \tag{7.4}$$

To implement an envelope generator with an exponential output, we will need to generate exponential curves based on *e*. This poses a fundamental problem: mathematical exponential functions do not discharge (decay) all the way down to 0.0 or charge all the way to 1.0. So, we need a way to not only generate an exponential approximation but also ensure that it advances smoothly across its thresholds.

7.5.1 Redmon's Analog EG Emulation

Nigel Redmon's implementation solves the exponential attack/decay issues and provides flexibility for many different curve shapes. It is also easy to modify to add more segments. Figure 7.12(a) shows a first order feedback structure with a single impulse as input and a feedback coefficient *b*

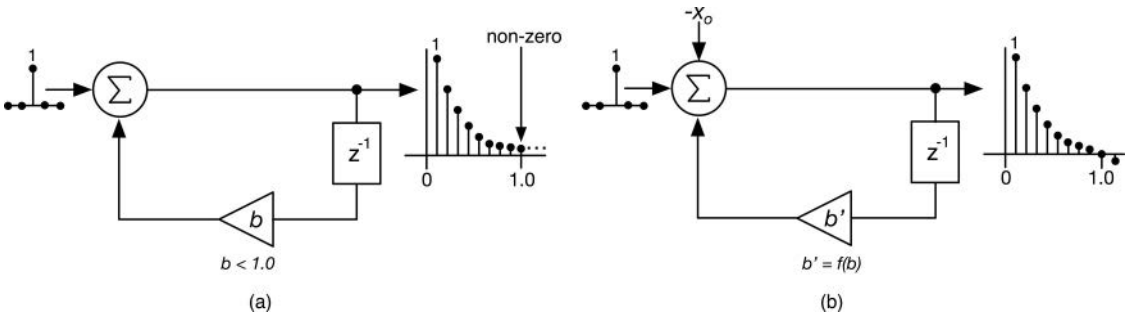


Figure 7.12 Attempting to generate a normalized exponential decay over the range of x = [0, 1] using (a) the first order feedback structure, we observe that the output does not decay to 0.0 and takes on a non-zero value at x = 1.0, while through (b) the addition of a negative bias signal and adjustment to the b coefficient (b'), we can achieve the proper decay curve, and hit, then cross over the 0.0 value

that is slightly less than 1.0. The impulse will recirculate through the structure, being multiplied by b each time. It is evident that the output will never reach 0.0, though it will eventually become so small that it cannot be represented with a *float* or *double* data-type – this is called underflow and usually results in a de-normaled number that you see as *#DEN* when debugging. The idea behind Redmon's EG is to add a small offset $-x_o$, as shown in Figure 7.12(b). This tiny bias signal adds a small negative offset to the output value each time the impulse recirculates, eventually forcing the value to 0.0, then crossing it into negative territory. In order to achieve the same exponential curve, the b coefficient needs to be altered to make up for the added offset.

To find the value of the coefficient b, we need to know the RC time constant that we want to emulate. For the ADSR EG, there will be three different b values, one for each exponential state: b_{ATT}, b_{DCY}, and b_{REL}. For example, if we look at the analog attack and decay equations in (7.4), we see that the exponential functions inside are $e^{-1.5x}$ for the attack segment and $e^{-4.5x}$ for the decay and release segments. Looking at the decay/release exponential, we can calculate that at x = 1.0, which is where we would like the function to evaluate to 0.0, the function will return a non-zero value:

$$\begin{aligned} y(x) &= e^{-4.5x} \\ y(1) &= e^{-4.5} = 0.011109 \end{aligned} \tag{7.5}$$

For Redmon's EG, this value is the *Time Constant Overshoot* (TCO). To calculate and adjust the b coefficient in order to give the proper decay over the normalized range, use Redmon's equation (7.6).

$$\begin{aligned} t_s &= \text{time in samples for attack, decay or release} \\ \alpha &= \frac{-\ln\left[\frac{1+TCO}{TCO}\right]}{t_s} \\ b &= e^{\alpha} \end{aligned} \tag{7.6}$$

For example, to correct the exponential decay function in Equation (7.5), where the exponent is $-4.5x$, you evaluate it at x = 1 to produce 0.011109. Equation (7.7) shows what happens to the alpha value's numerator; it produces −4.511048, which produces a slightly *slower* decay time than −4.5 as

applied to the exponential function. The addition of a small negative bias will then speed up the decay time so that hits 0.0 exactly in the time (in samples) required.

$$\alpha = \frac{-\ln\left[\frac{1+0.011109}{0.011109}\right]}{t_s} = \frac{-4.511048}{t_s}$$
$$b = e^{\alpha} \tag{7.7}$$

To find the bias value x_o, use Redmon's equation (7.8), where L_{SUS} is the sustain level on the range [0.0, 1.0].

$$x_0 = \begin{cases} attack & (1+TCO_{ATT})(1-b_{ATT}) \\ decay & (L_{SUS}-TCO_{DCY})(1-b_{DCY}) \\ release & (-TCO_{REL})(1-b_{REL}) \end{cases} \tag{7.8}$$

7.6 Synth Module: *EnvelopeGenerator*

Figure 7.13 shows the *EnvelopeGenerator* module, along with the three included module cores, and Table 7.1 lists the parameter structure and core information.

7.6.1 LinearEGCore

The linear EG is generally not useful as an amp EG, but it is set up for you to experiment with and to use as a starting point for your own EG ideas –the segments, timing calculations, and finite state machine operational code are present in the most minimal form possible. The *LinearEGCore* is available to download as an example EG core project for SynthLab-DM. Examine the code for this core to get started.

7.6.2 AnalogEGCore

The analog modeling EG core implements two EG contours: the traditional ADSR and attack-release (AR) modes. The AR mode has no sustain-level and is used as a starting point for experimenting with unconditional EGs. This EG implements Redmon's analog modeling equations (7.4)

Table 7.1 EnvelopeGenerator custom parameter structure and cores

Parameter Structure	Description
EGParameters	Used for all cores, includes additional DX controls
Included Core	Description
AnalogEGCore	Implements Redmon's analog modeling EG
DXEGCore	My variation on the Yamaha DX EG for FM synths
LinearEGCore	Ultra-simple linear EG to use as a starting point, Downloadable SynthLab-DM core project

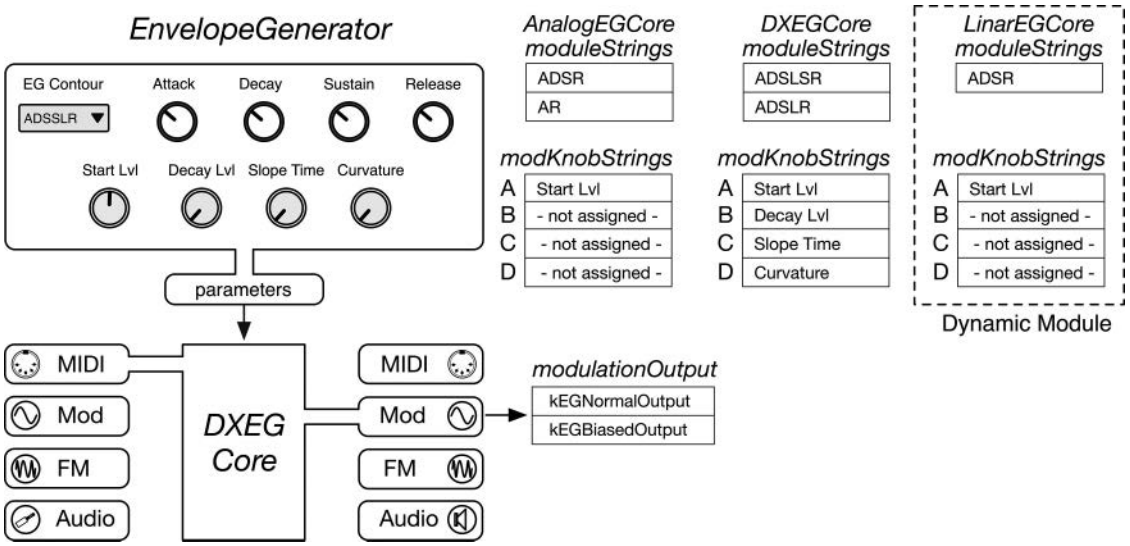


Figure 7.13 The *EnvelopeGenerator* module and the three included cores: DX-EG, analog EG, and linear EG; the module is shown with the *DXEGCore* selected

and (7.5) exactly; compare the C++ code with the equations. Notice that the calculation of attack, decay, and release time constants requires an *exp* operation so these are only updated when the user alters the GUI controls.

7.6.3 DXEGCore

The block diagram for the *DXEGCore* is shown in Figure 7.14. This EG has an added segment called "slope," as shown in Figure 7.9, and features a curvature control that uses the MMA concave and convex transforms to apply curvature to a linear EG FSM. The curvature control is a simple blending of the linear and transformed output values. The overall implementation follows a simple finite state machine design, as shown in Figure 7.7, but with the addition of the *slope* state, which is unaffected by the curvature control and remains linear. Examine the code for this core to see how the MMA transforms are applied, including the inverse transform versions.

The *DXEG* uses the simple calculations in Equation (7.2) to generate the various states, each of which has its own step-size, called the *egStepInc*. To simplify the object, the function *setStepInc* implements Equation (7.2) and is used to update the increment value each time the state changes.

7.7 EG Retrigger Modulation

The EG cores are modulation destinations for EG retrigger modulation. When retriggered, the EG will move directly back to the attack state and begin the FSM operation all over. Typically an LFO is attached as the modulation source. The retriggering operation occurs when a threshold is crossed. I made this a low-to-high crossing over the 0.5 unipolar middle value.

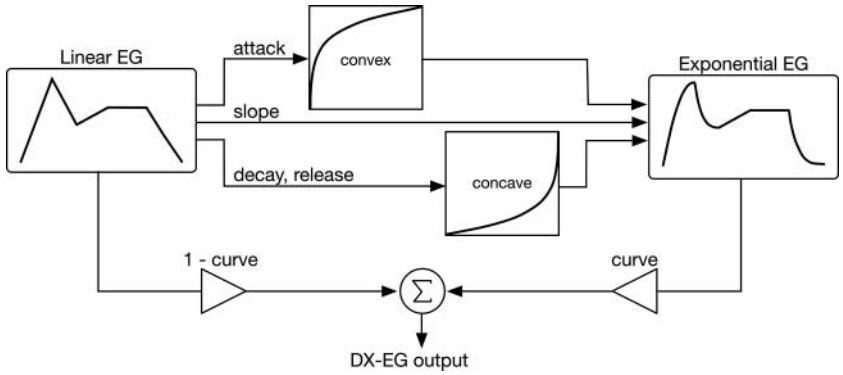


Figure 7.14 Conceptual block diagram of the *DXEG* with the curvature control; the concave and convex transforms are applied to the appropriate segments

7.8 EG Core Programming Notes

Section 7.8.1 through 7.8.6 list the main points for the five operational phases, plus constructor for the EG cores that subclass *ModuleCore* and override the same-named *SynthModule* virtual functions in Section 3.10. With the C++ files open, compare the programming notes and hints to the code that you see, and think about how each implements the FSM and deals with details like legato mode and shutdown. The EG cores all use a state-variable to keep track of their current EG segment. The *EGState* strongly typed enum is used for all cores and has enumeration strings for all possible states. If you want to add more states, such as a two-stage release, then modify this enumeration.

7.8.1 Construction Phase

All cores:

- The module strings are the EG contours, and their selection alters the FSM's state operation.
- Note the similarities and differences in the use of the mod knobs

7.8.2 Reset Phase

All cores:

- Reset the FSM state to *EGState* ::*kOff*
- Store sample rate, and calculate initial step values or coefficients
- Set the initial EG output value
- Clear the retriggering *noteOff* variable

Analog EG core:

- Calculates new time coefficients only if sample rate changes

```
// --- parameters
EGParameters* parameters =
       static_cast<EGParameters*>(processInfo.moduleParameters);

// --- retain sample rate for update calculations
sampleRate = processInfo.sampleRate;

// --- reset the state
envelopeOutput = parameters->startLevel;
state = EGState::kOff;

noteOff = false; // for retriggering EG
```

7.8.3 Note On Phase

DX EG and linear EG:

- Calculates the linear attack step size, then subtracts it from the output variable to effectively "back up" the EG by one click; this is used to make the initial attack state produce a perfect value of 0.0 or the user's start level

All cores:

- Set FSM state to *EGState::kAttack*
- Use mod knob A to set the start level, which is the initial output value
- Calculate velocity-to-attack, and note number-to-decay scaling values with a simple linear coefficient
- Notice the way in which legato mode is handled – the EG does not reset its output value when in legato mode

```
if (parameters->velocityToAttackScaling)
      attackTimeScalar = 1.0 - processInfo.noteEvent.midiNoteVelocity / 127.0;
else
      attackTimeScalar = 1.0;

if (parameters->noteNumberToDecayScaling)
      decayTimeScalar = 1.0 - processInfo.noteEvent.midiNoteNumber / 127.0;
else
      decayTimeScalar = 1.0;
```

7.8.4 Update Phase

All cores:

- Recalculate the step sizes or time constant coefficients according to GUI parameter changes
- Note that there is no real-time modulation input; the MIDI velocity and note number are only applied during the note-on phase

- When the MIDI sustain pedal is depressed, and the CC value is greater than 63, a sustain override flag is set, along with a release pending flag
- When the sustain pedal is released, the *update* function calls the note-off handler if the release is pending
- Notice how the retriggering modulation operates, crossing the threshold from low to high

```
else if(!noteOff)// process retriggering
{
     double retrig = processInfo.
                     modulationInputs->getModValue(kTriggerMod);

     if (retrig > 0.5 && !retriggered)
     {
          // --- reset
          envelopeOutput = parameters->startLevel;

          // --- go to the attack state
          state = EGState::kAttack;

          retriggered = true;
     }
     else if (retrig < 0.5 && retriggered)
     {
          retriggered = false;
     }
}
```

7.8.5 Render Phase

All cores:

- Steps through the finite state machine implemented with switch/case statements and uses the *EGState* and the known attack, decay, and sustain levels to trigger movement into the next state
- Output values are written into the *modulationOutput* array slots
- The for-loop only writes output values for the first iteration but runs all iterations to advance the state machine properly; this is especially important if the number of samples to render is less than the normal block size

DX core:

- Notice the way in which the MMA convex and concave, and their reverse functions are used to apply curvature to the state segments, except *EGState::kSlope*, which remains linear

```
// --- decode the state
switch (state)
{
      case EGState::kOff:
      {
            // --- if not legato, reset to start level
            if (!parameters->legatoMode)
                  envelopeOutput = parameters->startLevel;

            break;
      }

      case EGState::kAttack:
      etc…
```

7.8.5.1 DX Core Decay and Release

The DX core decay and release states operate the same way for the linear output, simply adding the new *stepInc* to the current value. However, we need to make sure we follow the rules for decaying from full scale, which involves only using a section of the normalized concave transform and is accomplished with a mapping function. This requires using the opposite transform to find the point at which the decay level hits the transform curve. Once the point is found, the section of the transform curve, as shown in dotted boxes in Figure 7.15, needs to map to the same part of the EG's curved segment, which is dependent on both the levels and the times involved. The mapping function effectively shrinks or expands the curve sections to stretch them into place.

7.8.6 Note Off Phase

All cores:

- EGs are the only modules that implement meaningful note-off handlers because of their contribution to the note-event life cycle in Section 4.1

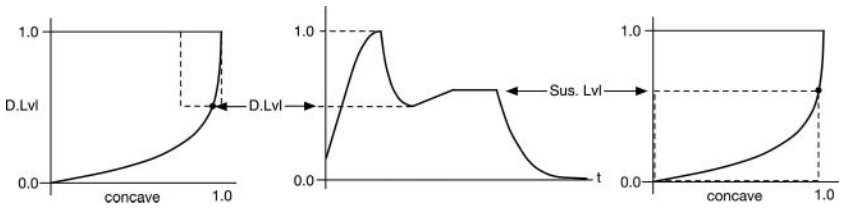


Figure 7.15 The decay and release portions of the EG curve need to be mapped to the correction sections of the normalized concave transform

- Notice how the sustain override flag will bypass the note-off handler if the sustain pedal is depressed
- All cores move the FSM into the *EGState::kRelease* state

7.9 The Digitally Controlled Amplifier (DCA)

The digitally controlled amplifier (DCA) is the digital equivalent of the analog voltage controlled amplifier (VCA), and it is the simplest of the synth modules, using basic calculations for gain (dB to raw) and panning (see pan modulation in Section 6.2.2). The DCA sits at the output of the voice architecture and controls the time domain contour of the note-event. The DCA is hard-wired to the amp EG that sets the note life cycle as this EG is responsible for reporting the end-of-note event. This object is so basic and simple that it does not use module cores. There is only one DCA object. The DCA needs to calculate the overall gain and panning of the voice output. Since the oscillators are pan-able, SynthLab does not implement a GUI control for pan – notice that, for it to work properly, there would need to be a GUI pan control for every voice. However, the pan functionality does play a role in two different ways: for unison mode, three of the four detuned voices are panned according to a simple scheme. You can see this code in the *SynthEngine::update* function, in which the panning and detuning are set for unison mode operation. The DCA pan operation is also a modulation destination, and you may connect an LFO or EG to source the pan modulation routing. All voices will pan alike, but they will begin the panning at note-on time, so staggering the note-events will produce quite a dizzying display of the pan function.

7.9.1 DCA EG Modulation Intensity

The DCA has a tiny GUI parameter structure, and the SynthLab projects only expose one control: the EG intensity. The EG intensity is interesting because it is a bipolar intensity value applied to the EG, whose output is unipolar in nature. This is a special modulation because of the way in which it operates. The EG mod intensity control defaults to a value of 1.0 so that the amp EG and DCA will work in the normal fashion. Reducing the intensity value scales the output and reduces it, like a volume control. When the intensity control is at 0.0, the DCA is silent. When the intensity control goes negative, rather than inverting the phase of the audio signal, we invert the EG control signal itself, simply by adding 1.0 to the value. This inverts the operation in an interesting way. Figure 7.17(a) and (b) show how the EG intensity control operates with a positive or negative polarity. For the inverted version, loud becomes quiet, and quiet becomes loud, with the note-event ending at maximum volume.

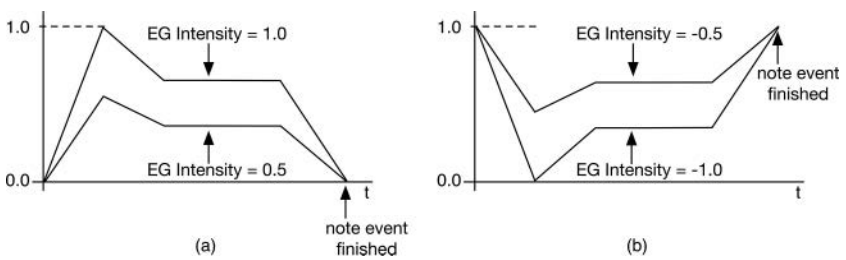


Figure 7.16 (a) The EG intensity acts as a simple gain control when positive, but (b) when negative, it inverts the unipolar signal

7.9.2 DCA Note On Phase

When the user strikes a MIDI key, the controller sends a velocity value that translates into how much force was used to strike the key. The velocity value is converted to a raw value, as described in Section 6.2's MIDI velocity modulation equations, and uses a built-in function. The MIDI velocity arrives in the note-event structure.

```
bool DCA::doNoteOn(MIDINoteEvent& noteEvent)
{
     // --- store our MIDI velocity
     midiVelocityGain = mmaMIDItoAtten(noteEvent.midiNoteVelocity);

     etc...
```

7.9.3 DCA Update Phase

The DCA update phase is the only interesting operation as it applies the amp, EG, and panning modulation values. The amp modulation control creates the tremolo effect and must connect to a LFO's unipolar-from-max output value in order to operate intuitively. The update function is shown below in its entirety. Examine the code, and notice how the EG intensity control is handled.

```
// --- apply Max Down modulator
double ampMod = doUnipolarModulationFromMax(
                  modulationInput->getModValue(kMaxDownAmpMod), 0.0, 1.0);
ampMod *= parameters->ampModIntensity;

// --- EG modulation
double egMod = parameters->ampEGIntensity * modulationInput->getModVal-
ue(kEGMod);

// --- flip EG output
if (parameters->ampEGIntensity < 0.0)
     egMod += 1.0;
```

The DCA also has built-in support for MIDI CC #7 (volume):

```
// --- support for MIDI Volume CC
double midiVolumeGain = mmaMIDItoAtten(midiInputData->getCCMIDIData(VOL-
UME _ CC07));
```

The final gain calculation is just a combination of all of these, along with the original velocity-based gain value.

```
// --- calculate the final raw gain value
// multiply the various gains together
gainRaw = midiVelocityGain * egMod * ampMod;

// --- apply final output gain
if (parameters->gainValue _ dB > kMinAbsoluteGain _ dB)
     gainRaw *= pow(10.0, parameters->gainValue _ dB / 20.0);
```

```
else
     gainRaw = 0.0; // OFF
```

The pan modulation uses the *panValue* as the center of operation. This value is 0.0, except for the unison mode, where the value is preset according to the unison mode scheme. Pan modulation is described in Section 6.2.2.

```
// --- now process pan modifiers
double panTotal = panValue + (parameters->panModIntensity *
                                    modulationInput->getModValue(kPanMod));

// --- limit in case pan control is biased
boundValueBipolar(panTotal);

// --- equal power calculation in synthfunction.h
calculatePanValues(panTotal, panLeftGain, panRightGain);
```

7.10 Exercises

7.10.1 SynthLab-DM: Pre-Release state

The linear EG is used as the example dynamic module for the EG object. Download the sample code, then compile and install the DLL into the SynthLab-DM project of your choice, following the instructions in Section 5.2.2. Select the linear EG core to use for the output amp EG and filter EG, and pay attention to how it sounds. Next, add a second release state modifying the FSM code and adding a new *EGState::kPreRelease*, and use one of the mod knobs to adjust the pre-release time and another to set the pre-release level. Implement the required code in the methods corresponding to the operational phases.

7.10.2 Stand-Alone Module Exercise: More Contours

Add a stand-alone EG object to a plugin of your own design, and write the code to integrate it into your system, remembering the five operational phases. You may choose any of the cores as a starting point. Add the GUI controls, then make sure you can update and render the EG values properly. Next, modify the core by adding more EG contours – the stock objects only implement two contours each, so you have plenty of free module strings to add more contours. Notice how the additional contours change the FSM operation. You may need to add more states to the *EGState* enumeration.

7.10.3 Stand-Alone Module Exercise: Unconditional Release

Review the way the unconditional release EG operates when the musician triggers a complete cycle of EG states with a press and release of a key. Add unconditional release operation to your modified stand-alone module. What changes need to occur in the note-off handler?

7.10.4 Advanced Module Core: Yamaha and Casio CZ EGs

Figure 7.17(a) and (b) show the EG contours for the Yamaha EX and Casio CZ series synths, respectively. Both feature multiple decay and release segments. Implement one or both EGs using

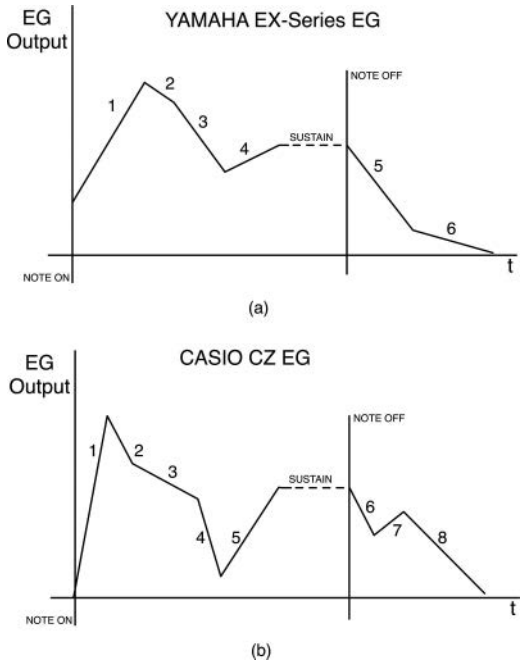


Figure 7.17 Two more EG contours for the (a) Yamaha EX and (b) Casio CZ series synths; the contours are shown as linear here for simplicity; most segments are curved

a stand-alone module core object in your own project or in any of the downloadable SynthLab projects. Think about the challenges with the larger finite state machines and whether you intend to implement curvature on some or all of the EG segments. Notice that the decay and release segments may move in the up or down direction, with the exception of the last release segment, which decays downward to 0.0 or the release level.

Bibliography

digichip.com. "CEM3310 Voltage Controlled Envelope Generator Datasheet." https://www.digchip.com/datasheets/parts/datasheet/922/CEM3310-pdf.php, Accessed October 14, 2020

Hurtig, Brent, Ed. 1984. *Synthesizer Basics*, pp. 29–35. Winona: Hal Leonard Corporation Korg, Inc. 1997. *Triton Music Workstation Basic Guide*. Tokyo: Korg Inc. manuals.lib. "Casio CZ-1000 Operation Manual." https://www.manualslib.com/manual/1160175/Casio-Cz-1000.html, Accessed October 14, 2020

Redmon, Nigel. "Envelope Generators Part 2." http://www.earlevel.com/main/2013/06/02/envelope-generators-adsr-part-2/, Accessed October 14, 2020

Yamaha, Inc. 1983. *DX7 User's Manual.* Tokyo: Yamaha Inc.

8 Low Frequency Oscillators

Low Frequency Oscillators (LFOs) belong to the render-group of synth building blocks as they render an output without processing an input signal. LFOs are used as modulation sources that generate relatively slow-moving values used to modulate the parameters of a destination object. Since these oscillators are not used for audio frequencies, they do not suffer from aliasing issues and can be generated with extremely simple equations. They are sometimes called *trivial oscillators*. The requirements for LFOs and pitched oscillators are summarized as follows:

LFO Requirements:

- Frequency range: 0.02 Hz–20.0 Hz (though this is highly variable)
- Constructed from simple equations, lookup tables, or piecewise functions
- Aliasing not a consideration since these are not used for audio output
- For block processing, LFO samples are only generated once per block prior to modulation

Pitched Oscillator Requirements:

- Frequency range: 20 Hz–20,480 Hz (10 octaves)
- Constructed from tables or band-limiting algorithms; direct form and other traditional DSP algorithms have issues which make them unsuitable
- Should not alias, but if aliasing is present, then it must be as inaudible as possible or masked by other harmonic components
- For block processing, samples are generated on each sample interval to fill the block

8.1 Noise Oscillators and Generators

Noise oscillators are unique in that they are used in both LFOs and pitched oscillators. These oscillators do not have a fundamental oscillator frequency f_o like the others. They may be calculated with random number generators. The *NoiseGenerator* produces white, Gaussian white, and pink noise. You can add more noise types as well, and there are numerous variations. The output of the *NoiseGenerator* is always on the range [−1.0, +1.0].

White noise is the result of a truly random number and has a flat spectrum indicating constant amplitude across all frequency components, as shown in Figure 8.1(a). The name "white" corresponds to white light, which includes equal contributions from all the visible color frequencies – also known as the color spectrum.

Gaussian white noise (also called additive Gaussian white noise or *AGW*) likewise has a flat frequency response, as shown in Figure 8.1(a). The difference is in the variance of the sample amplitudes. In white noise, there is an equal probability that any sample will lie anywhere in the range

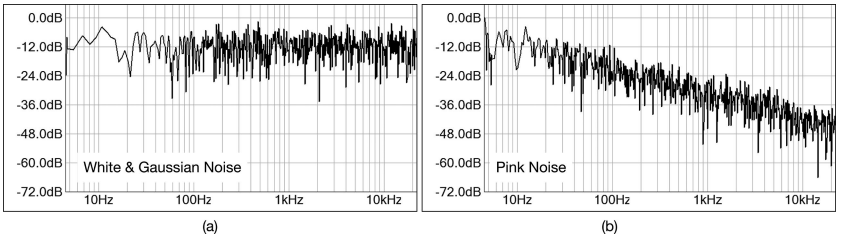


Figure 8.1 The frequency spectra of (a) white and (b) pink noise; FFT length is 131,072 points

of [−1.0, +1.0] – each output is random and uncorrelated to the previous outputs, and over time, averaging the sample values results in a zero mean. Gaussian white noise also has a zero mean when averaged over time. The difference lies in the Gaussian sample value variance, which follows a bell-shaped curve called the *standard distribution*. This means that an output sample has a higher probability of being near the extreme (center) +/−1.0 than of being near the minimum value around 0.0.

Pink noise has a spectrum that rolls off at −3 dB/octave or −10 dB/decade, as shown in Figure 8.1(b). There are numerous approaches to the digital design of the shallow filter, which is simpler in analog. Paul Kellet's "economy" method involves a simple, three-coefficient lowpass filter on white noise. This pinking filter code is detailed in the SynthLab documentation and the firstpr.com reference.

There are several methods of generating the random noise signals. The Standard Library object called *std::default_random_engine* simplifies the creation of white and Gaussian white noise significantly.

```
std::default_random_engine defaultGeneratorEngine;
```

Gaussian (normal distribution with mean = 0, variance = 1):

```
std::normal_distribution<double> normalDistribution(mean, variance);
gaussianWN = normalDistribution(defaultGeneratorEngine);
```

White (random distribution on interval [−1, +1]):

```
std::uniform_real_distribution<double> randomDisribution(-1.0, 1.0);
whiteNoise = randomDisribution(defaultGeneratorEngine);
```

Pink (filtered white noise):

```
pinkNoise = doPinkingFilter(doWhiteNoise());
pinkNoise *= 0.25; // filter gain is > 1
```

8.2 Oscillator Clocking

All non-noise oscillators require a time-base that is directly linked to the system sample rate. Most SynthLab oscillators will use the same time-base, which consists of a modulo counter named *mcounter*. This is also known as a phase accumulator, but I use modulo counter exclusively here. A modulo counter starts from 0.0, then begins counting upwards at a rate corresponding to the oscillator's frequency. The counter step-size is called the phase increment (*phaseInc*) and is applied once per sample period. When the modulo counter crosses above the value 1.0 by some amount Δ, the clock rolls over and begins counting upward again, starting from the Δ value. Figure 8.2(a)

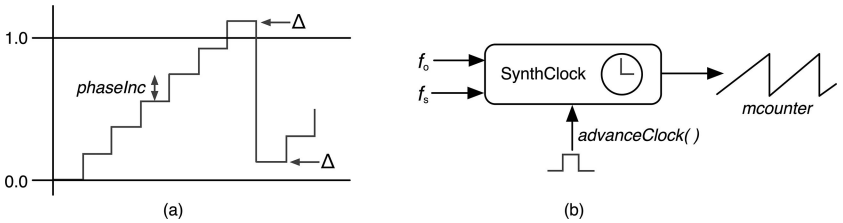


Figure 8.2 (a) The modulo counter steps upward by the *phaseInc* on each sample period until the counter crosses 1.0 and wraps to the same offset, and (b) the block diagram form of the *SynthClock* object

demonstrates this concept: the *phaseInc* is the step-size, and the counter rolls over to the offset level Δ. The *phaseInc* calculation is blissfully simple and requires no trig function calls – it is simply the ratio of the desired frequency of oscillation f_o and the sample rate f_s.

$$phaseInc = \frac{f_o}{f_s} \tag{8.1}$$

8.2.1 The SynthClock Object

To handle the modulo counter operation, I have included a simple C++ object called *SynthClock*. Figure 8.2(b) shows a conceptual block diagram of the *SynthClock* object. You use the *setFrequency* function to apply f_o, f_s to calculate the *phaseInc*, and the *advanceClock* function to increment the time-base. The functions *wrapClock* and *advanceWrapClock* are used for the wrapping operation.

8.2.2 Reversing Time and Oscillator Scanning

All of the SynthLab oscillators can be designed to run forward or backwards, the latter a requirement for FM synthesis that also allows interesting waveform generation. To run an oscillator backwards, you simply negate the *phaseInc* parameter so that it counts down rather than up. The *SynthClock* object will take care of wrapping the *mcounter* in the opposite direction. You can generate numerous new waveforms from your existing ones quite easily. One concept is called *scanning*; this is when you alternate the forward and reverse directions each time the *mcounter* wrap occurs in either direction. This produces back-to-back waveform cycles, as shown in Figure 8.9, for the sine and triangle scanned waves. These were popular for early wavetable pitched oscillators as the scan reversal operation is simple to implement. You can also change the direction at different points in the waveform cycle or follow more complex patterns to create distinctive waveforms.

8.3 LFO Waveforms and Rendering Equations

The SynthLab's default LFO core produces nine different waveforms, as shown in Figures 8.3 and 8.4. The calculations for each waveform are listed to the right, while the name and constant enumeration strings are to the left.

8.3.1 Fundamental Waveforms

The first set of waveforms consists of the most common LFOs you will find in most synths. These include ramp up, ramp down, and triangle waveforms. These are all calculated directly from the *SynthClock's mcounter*, using simple equations. The ramp up waveform is used as the basis for the others.

8.3.2 Exponential Waveforms

The default *LFOCore* also synthesizes exponential versions of the three basic waveforms. The MMA concave transform *concaveXForm* is used to apply the exponential curve shaping across the input and output ranges [−1.0, +1.0]. The key to using these transforms is applying them to the correct parameter, which varies from one waveform to the next and is shown for these waveforms in Figures 8.3 and 8.4.

8.3.3 Sinusoidal Approximations

For the LFO sine waveform, we could use a high-resolution lookup table, but that might be overkill for our LFO, which is only rendering one output sample per block of rendered samples. There are numerous sinusoidal approximations, ranging from simple to relatively complex. The sinusoids they produce are visually and, for most people, audibly identical, but the Fourier spectra reveal the harmonic distortion components. The parabolic sine approximation is nearly distortion free and so clean that it can even be used as a pitched oscillator if desired. Figure 8.5 shows the spectra and equations for each.

8.3.4 Random Sample and Hold

The noise generator is commonly used to sample the random output and hold the value for some period of time. This requires the noise generator, a register to hold the output value, and a *Timer*

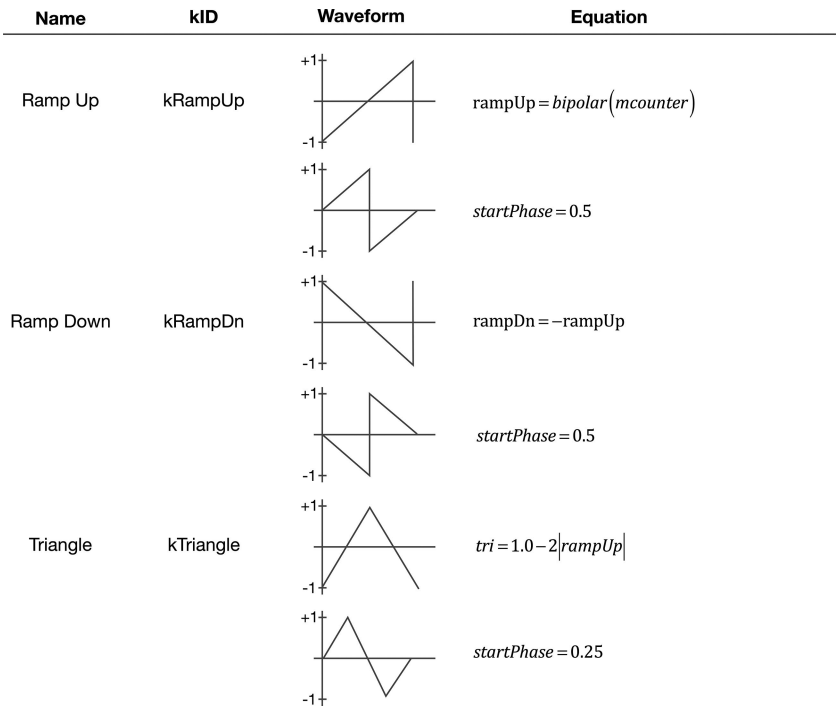


Figure 8.3 Triangle and ramp waveforms and their algorithms; the ID values correspond to the waveform selector in SynthLab

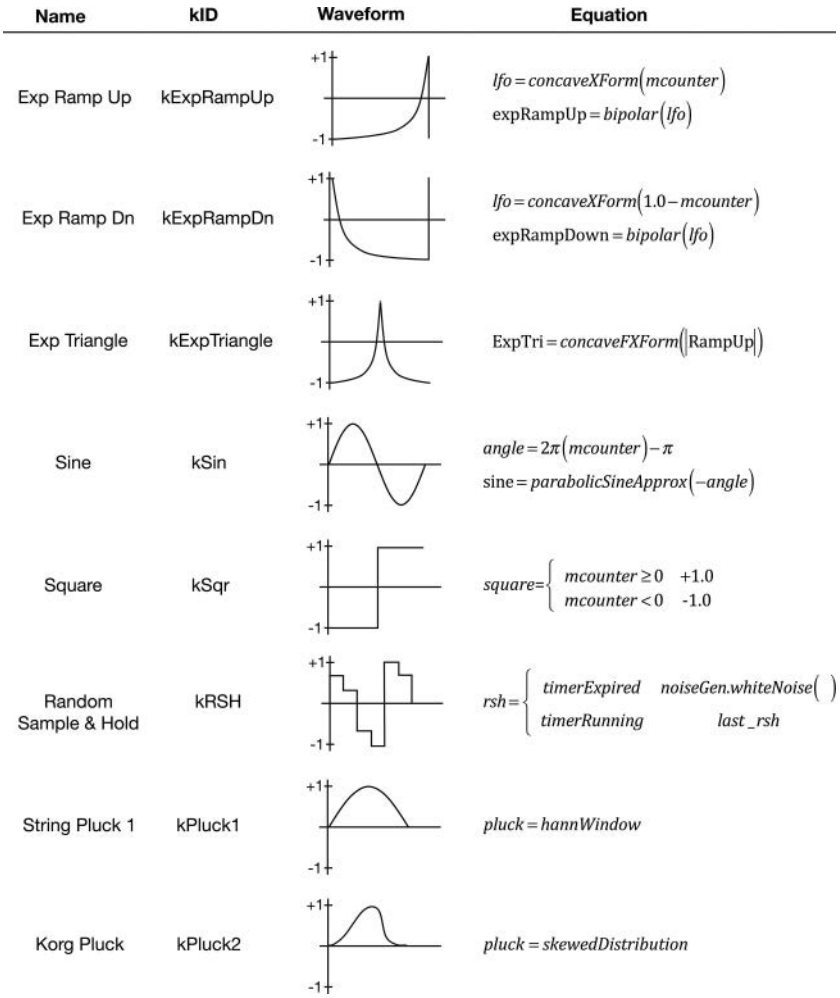


Figure 8.4 Exponential, sinusoidal, square, random, and pluck waveforms and their algorithms

object to trigger the sampling operation. For the common sample and hold LFO, the timer is set to expire after N sample periods related to the LFO frequency's period $T = 1/f_o$. This produces a waveform with random amplitude values and a constant hold time. In another variation, you ignore the LFO frequency and reset the timer with a random hold time for each trigger. This produces both random amplitudes and random hold time intervals. These LFO block diagrams are shown in Figure 8.6.

8.3.5 Stepped (Quantized) Waveforms

The stepped or quantized LFO waveforms are also sample and hold types, but they can be implemented without timers. One option is to use the quantization formula on the *mcounter* value

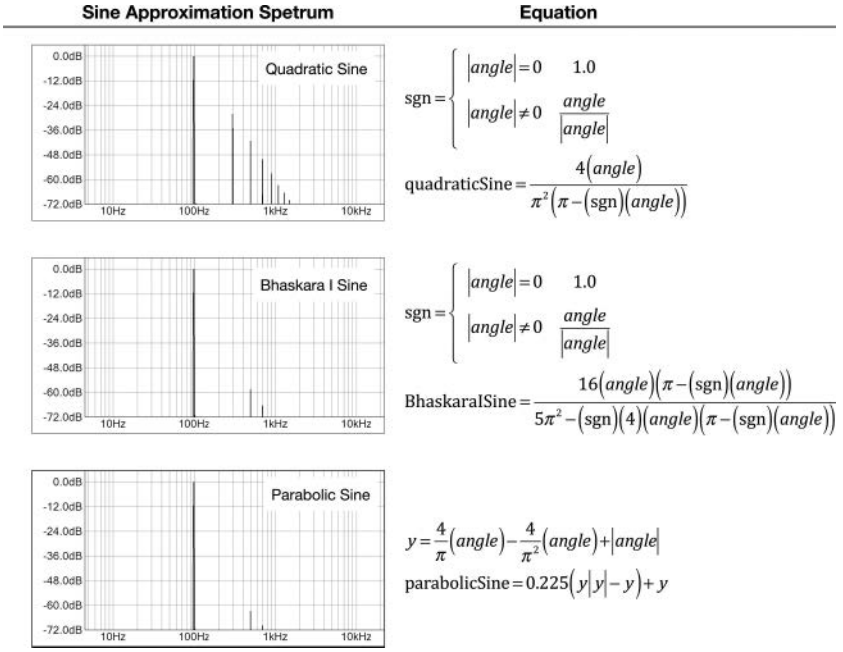


Figure 8.5 Frequency spectra and equations for three popular sinusoidal approximations: quadratic, Bhaskara I, and parabolic, with f_o = 100 Hz

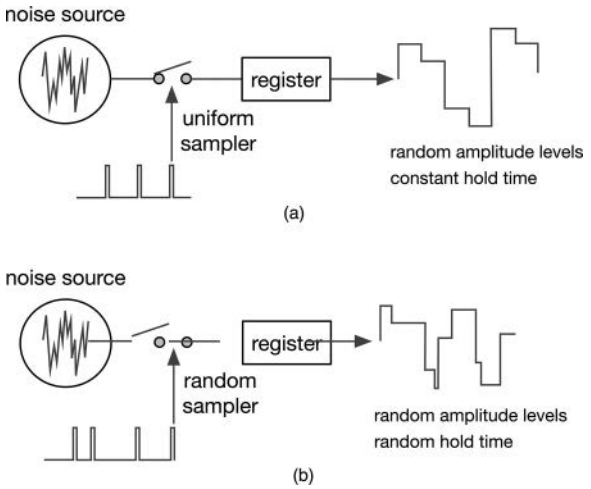


Figure 8.6 Random sample and hold LFOs with (a) random amplitude and constant hold time, and (b) random amplitude and random hold times

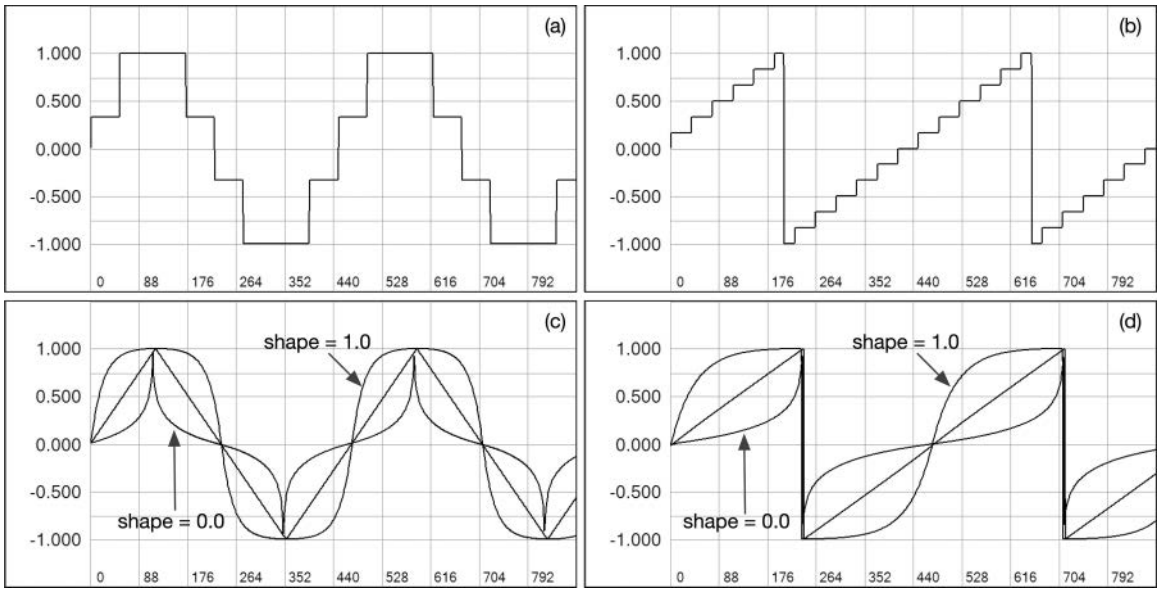


Figure 8.7 Stepped LFO waveform examples: (a) three-step sine, (b) six-step triangle, and shaped LFO waveforms (c) triangle and (d) ramp up with shape control = (0.0, 0.5, 1.0)

to produce the steps; this technique uses the same bit-crusher algorithm that appears in my FX plugin book. Another option is to use the mapping functions in *synthfunctions.h*, which map double values to unsigned integers, and vice versa. This first quantizes the LFO output value as an *unsigned integer*, then remaps it back to a *double* value. The quantization occurs on the LFO's final bipolar output value, prior to attenuation and other processing. In the SynthLab LFO, the stepper value is an integer that dictates the total number of steps across the output amplitude range; note that −1.0 is not counted as a step. Figure 8.7(a) and (b) show two waveforms processed at different quantization levels (*QL)*. Quantizing the square wave has no effect as it is always a one-step waveform. Quantizing the random sample and hold waveform will produce variations on the sample and hold that can be quite interesting.

8.3.6 One-Sided (Unipolar) Waveforms

There are a few unipolar LFO waveforms, including Korg's "pluck" wave. Here, the idea is to simulate a plucked string, which goes sharp and then comes back into tune very quickly at the attack of the note; the note never goes flat during the plucking process. The waveform resembles a leaning or skewed window function in Figure 8.4.

8.4 Render Modes

There are three common render modes for the LFO: one-shot, free-run, and synchronized (sync). The one-shot LFO renders one cycle of the waveform and then stops, either holding the last output

or forcing the last output to zero. The free-run LFO starts running on the first note-on event, then runs forever, never resetting the LFO. In sync mode, the LFO is phase-reset and restarted on each new note-on event.

8.5 Waveform Shaping

One common LFO parameter that varies widely in implementation is a *shape* control, which alters some aspect of the LFO's time-domain signal. A simple technique is to blend the output of two different LFO waveforms, A and B, using the shape control to mix them in different ratios, producing a range of hybrid waveforms. If one of the signals is a sinusoid, then the shape control will behave in a similar manner to a lowpass filter. Another method is to use waveshaping on the LFO output or the modulo counter value to bend the segments. For the SynthLab LFO cores, when the modulation knob is above 0.5, the reverse concave transform is applied, while the normal concave transform is applied when the control is below 0.5. This produces a shape control that is sometimes called a curve or curvature control. Figure 8.7(c) and (d) show how the shape control affects the curvature of two different waveforms. When shape = 0.5, no curvature is applied.

8.6 Delay and Fade-in Times

There are two more common LFO parameters: delay time and fade-in time. The delay time specifies how long it will take the LFO to start generating non-zero samples. The fade-in time uses a linear amplitude ramp to fade the LFO from an amplitude of 0.0 to 1.0. For SynthLab, both times are given in milliseconds. The implementations are easy: for the delay time, use a *Timer* object that is reset at note-on and set to expire at the user's delay time setting. For fade-in, use a *RampModulator* object to generate a linear ramp of values from [0.0, +1.0] over the user's fade-in time period, then scale the output with the ramp.

8.7 Starting Phase

To adjust the starting phase of the LFO, you add a phase offset to the modulo counter so that it starts at a non-zero location. An offset of 0.5 produces a shift of 180 degrees. You may want to experiment with the LFO starting phase as it applies to the triangle and ramp waveforms. If you examine Figure 8.3, you will see that the triangle and ramp waveforms start and end at non-zero locations. When used in one-shot mode, the LFO may start or end abruptly. In addition, when trying to blend LFOs with the sinusoidal waveform, there will be phase issues as the sine always starts and ends on 0.0. Figure 8.3 shows how to adjust the starting phase of the SynthLab LFO triangle and ramp waveforms to make them symmetrical, starting and ending at 0.0.

8.8 DC Offset

You may also offset the LFO waveform with a constant value called a DC offset. To do this, you simply add or subtract a static value to the LFO's output. Notice that you may use it to convert one-sided LFO waveforms into bipolar versions. You need to decide how to handle values outside of the normalized bipolar range [−1.0, +1.0].

8.9 *SynthLFO* and Cores

The *SynthLFO* and its core objects are shown in block form in Figure 8.8. The custom module strings are the LFO waveform strings, and the mod knobs are chosen for the most common operations.

8.9.1 LFO Controls, Modulations, and Features

The LFO frequency and output controls are self-explanatory. The step control quantizes the output when its value is two or greater and works as an interesting arpeggiator when connected to modulate an oscillator's pitch. The BPM sync is a special feature and locks the LFO rate to a specified musical duration using the host's BPM value and the BPM sync modulation in Section 6.4. Table 8.1 lists the cores and their functionality.

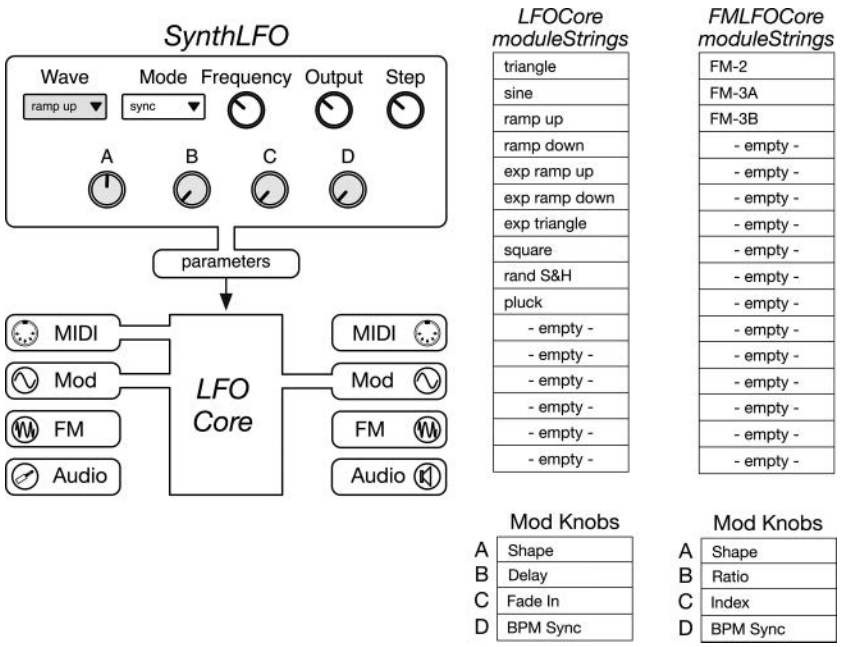


Figure 8.8 The *SynthLFO* and its two cores block diagrams, module strings, and mod knobs

Table 8.1 LFO cores and their special modulation capabilities (*) downloadable dynamic module projects

Parameter Structure	Description
LFOParameters	Every possible parameter for both cores
Core	**Features**
*LFOCore**	Classic LFO waveforms, including the one-sided pluck waveform, based on the Korg Wavstate LFO
*FMLFOCore**	Creates FM LFO waveforms using phase modulation (see Section 16.6)

8.10 LFO Core Programming Notes

With the C++ files open, compare the programming notes with the code that you see, starting with the class descriptions of each core. The LFO cores use the *SynthClock* as their time-base. The normal core implements the waveform equations in Figures 8.3 and 8.4. The LFO core implements two FM algorithms: carrier and modulator in series, and a modulator – modulator – carrier version. Compare the code to the details in Section 16.6.

The cores require two registers: one to store the current output value for one-shot mode, which holds the last value indefinitely, and another for the random sample and hold value. There is also a *renderComplete* flag for the one-shot operation. Table 8.2 lists the *LFOCore* member objects and their LFO function. See the C++ documentation for object details.

Sections 8.10.1 through 8.10.4 summarize the five operational phases, plus the constructor for the *LFOCore*. Make sure to examine the code while digesting their operational phase details. The *FMLFOCore* only differs in the render operation.

8.10.1 Construction Phase

The LFO core constructors are super simple as they only expose the module string and mod knob labels.

- The module strings are the waveform names, and their index values map to the GUI control selection values

8.10.2 Reset and Note-On Phases

The reset and note-on operations are nearly identical. The only real difference is that the note-on function branches according to the free-run mode status and does not reset timers for this state. Setting up the timers is straightforward – note the use of MAX constants:

```
delay_mSec = getModKnobValueLinear(parameters->modKnobValue[LFO_DELAY],
                                                    0.0, MAX_LFO_DELAY_MSEC);
delayTimer.setExpireSamples(msecToSamples(sampleRate, delay_mSec));

fadeIn_mSec = getModKnobValueLinear(
  parameters->modKnobValue[LFO_FADE_IN], 0.0, MAX_LFO_FADEIN_MSEC);

fadeInModulator.startModulator(0.0, 1.0, fadeIn_mSec,
                                                    processInfo.sampleRate);
```

Table 8.2 LFOCore member objects

LFO Functionality	C++ Object	Member Name/Notes
Time-base	*SynthClock*	*lfoClock*
Noise Generation	*NoiseGenerator*	*noiseGen*
Sample & Hold	*Timer*	*sampleHoldTimer*
Delay	*Timer*	*delayTimer*
Fade-In	*RampModulator*	*fadeInModulator*
Pluck Waveforms	*BasicLookupTables*	*lookupTables*

For sync mode, the phases are adjusted to produce symmetrical waveforms starting and ending at 0.0.

```
if (parameters->waveform == LFOWaveform::kTriangle)
      lfoClock.addPhaseOffset(0.25);

if (parameters->waveform == LFOWaveform::kRampUp ||
    parameters->waveform == LFOWaveform::kRampDown)
      lfoClock.addPhaseOffset(0.5);
```

8.10.3 Update Phase

The LFO core update functions are quite simple as this is a basic modulator.

- BPM sync uses the *getTimeFromTempo* function and is detailed in Section 6.5. The *frequency_Hz* parameter is the sync target
- Bipolar linear FM is applied to the *frequency_Hz* parameter to calculate the final LFO frequency and is detailed in Section 6.4 using the pre-calculated modulation range
- The SynthClock member is updated with the new oscillator frequency

```
modValue =
    processInfo.modulationInputs[kFrequencyMod]*LFO_HALF_RANGE;

modulatedFreq_Hz = parameters->frequency_Hz + modValue;

// --- update the phase inc from the frequency
lfoClock.setFrequency(newFrequency_Hz, sampleRate);
```

8.10.4 Render Phase

The block diagram in Figure 8.8 shows that the LFO accepts incoming MIDI and modulation values, and writes its output into the *modulationOutputs* array. There are two outputs defined: normal and inverted. The render operation is made up of five steps:

1 Check delay timer; if not expired, set output = 0.0 and return
2 Check one-shot mode: call the *lfoClock->wrapClock* method, and check the return value; if wrapped, then the one-shot cycle is completed; set the flag and return the final LFO value
3 Decode the *parameters->waveform* enumeration and render the waveform according to the calculations in Figures 8.3 and 8.4. Partial code listing:

```
if (parameters->waveform == LFOWaveform::kSin)
{
     outputValue = parabolicSine(-(lfoClock.mcounter*kTwoPi - kPi));
}
else if (parameters->waveform == LFOWaveform::kRampUp)
     outputValue = bipolar(lfoClock.mcounter);
```

```
else if (parameters->waveform == LFOWaveform::kExpRampUp)
     outputValue = bipolar(concaveXForm(lfoClock.mcounter));
else if (parameters->waveform == LFOWaveform::kTriangle)
     outputValue = 1.0 - 2.0*fabs(bipolar(lfoClock.mcounter));
```

etc…

4 Apply post-render processing to output value

- Step quantize: see Section 6.1.3
- Shape: use the concave transform functions to modulate according to distance above/below center value of 0.5

```
if (parameters->modKnobValue[LFO_SHAPE]) >= 0.5)
     shapeOut = bipolarConvexXForm(outputValue);
else
     shapeOut = bipolarConcaveXForm(outputValue);
```

Next, use split-bipolar transform on the mod knob value (see Section 6.1), then blend in some amount of the curved waveform, along with the normal output:

```
// --- split bipolar for multiplier
double shape = splitBipolar(parameters->modKnobValue[LFO_SHAPE]);

outputValue = shape*shapeOut + (1.0 - shape)*outputValue;
```

- Apply fade-in modulation: this is standard fade-in ramp modulation detailed in Section 6.1.4

5 Write normal and inverted outputs into modOutputs

```
processInfo.modulationOutputs[kLFONormalOutput] = outputValue;
processInfo.modulationOutputs[kLFOInvertedOutput] = -outputValue;
```

8.11 Exercises

8.11.1 SynthLab-DM: The Pluck Waveform

Implement the basic string pluck waveform in Figure 8.4. The simple version uses a Hann windowing function to produce the one-sided waveform. For the Hann window, you can use the *BasicLookupTables* object already declared in the *LFOCore*. First, you need to add the *kPluck* enumeration item to the existing definition in the *lfocore.h* class definition file.

```
enum class LFOWaveform { kTriangle, kSin, . . . , kRSH, kPluck };
```

The lookup table does all the work, and there are no parameters on the pluck waveform to manipulate. In the *render()* function, you need to decode the LFO waveform and set the output variable using the lookup table object. See Appendix A for *BasicLookupTables* details.

```
// --- use hann table; can experiment
```

```
else if (parameters->waveform == LFOWaveform::kPluck)
     outputValue = lookupTables->readHannTableWithNormIndex(lfoClock.
     mcounter);
```

8.11.2 SynthLab-DM: Korg Pluck Waveform

While the Hann window works reasonably well in short-duration one-shot mode, the Korg version is more realistic for stringed instrument emulation. Notice that the Korg waveform is a bent or leaning bell curve that may be fashioned in a variety of manners. Research "skewed distribution," and formulate your own version of the Korg pluck waveform. Is there a parameter you can vary to easily change the skew? If so, you could re-purpose a mod knob to allow user adjustment.

8.11.3 SynthLab-DM: Mod Knob Shape to DC Offset Control

For this exercise, you change the function of the shape control to add or remove a DC offset instead. The shape control defaults to a value of 0.5 so we can set that as the no-offset value, then increase in the positive or negative direction using the *splitBipolar* function (see *synthfunctions.h*). When the user moves the control above 0.5, you apply a positive DC offset, and when the control is below 0.5, you apply a negative DC offset.

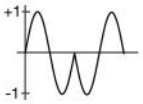
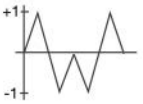
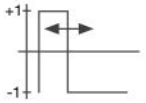
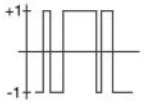
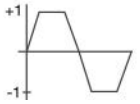
Name	kID	Waveform	Notes
Double Sine	kDoubleSin		Use oscillator scanning and a sine approximation function
Double Triangle	kDoubleTri		Use oscillator scanning and the ramp-to-triangle conversion
PWM Square	kSqrPWM		Square wave with adjustable pulse width Use a mod knob for pulse-width setting Use *mcounter* value to toggle pulse
Random PWM Square	kRandSqrPWM		Use random *mcounter* thresholds with each cycle; allow user to adjust the level of randomness with a mod knob
Trapezoid	kTrapezoid		Use counter and thresholds, or amplify and clip the triangle wave

Figure 8.9 Additional LFO waveforms for your custom object

8.11.4 Advanced Module: Create a Novel LFOCore

Finally, make your own *LFOCore* object using the existing object as a basic template, and keep the normal modulations. For your personal *LFOCore*, add the new LFO waveforms from Figure 8.9, and implement any four mod knob parameters you wish, starting with the suggested modifiers.

Mod Knob A: starting phase
Mod Knob B: pulse-width (for square PWM)
Mod Knob C: DC offset
Mod Knob D: BPM Sync (keep existing)

Bibliography

Korg.com. "Wavestate Owner's Manual." https://www.korg.com/us/support/download/product/0/840/, Accessed October 14, 2020

firstpr.com. "DSP generation of Pink (1/f) Noise." http://www.firstpr.com.au/dsp/pink-noise/, Accessed October 14, 2020

9 Wavetable Oscillators

Wavetable oscillators are pitched oscillators that cycle through a lookup table to generate a waveform. These are some of the earliest of the digital oscillators designed for audio use, having been around since the 1960s. Chowning's FM synthesis required wavetable oscillators as they are inherently stable and can produce the same sequence of output values over and over. Wavetables were used in the first hybrid synths, such as the Sequential Circuits Prophet VS, as well as fully digital synths, like the Korg M1 and Yamaha DX7. Wavetable synthesis seems to come and go in popularity, but right now it is very hot – this may be due to the appearance of more morphing wavetable synths and tools that allow us to develop scores of wavetables at a time from algorithms or found-sounds.

9.1 Wavetable Fundamentals: Table Lookup

The traditional wavetable is a pre-calculated array of length N that contains one cycle, minus one sample, of some waveform. The length is typically a power of two, but that is not strictly required. The oscillator uses a pointer or an index value to access samples stored in the table. In the most primitive form shown in Figure 9.1(a), the oscillator outputs these values directly, skipping through the table at a lookup interval called *wtPhaseInc*, which is an integer. More advanced versions use floating point increment values and interpolation to calculate in-between values, as shown in Figure 9.1(c).

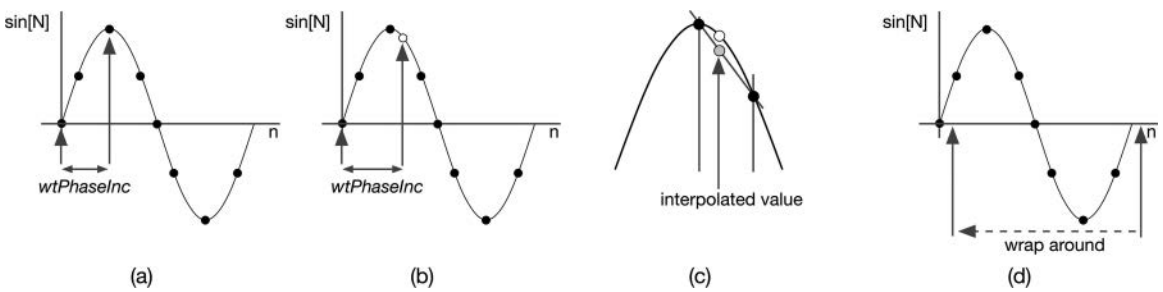


Figure 9.1 Reading an $N = 8$ point wavetable with (a) integer lookup and (b) floating point lookup with (c) linear interpolation and (d) the read index wraps around to the start of the table when the added phase increment pushes the lookup point past the table's ending index

When the table index or pointer passes the last sample in the array by some amount Δ, it wraps back around to the top of the array with an offset of Δ and continues marching through the buffer, as shown in Figure 9.1(d). This is simply a scaled version of the modulo counter time-base we use for the LFO and the other pitched oscillators, and we can use the same basic clocking functionality. The wavetable phase increment value is simply the modulo counter value multiplied by the table size *N*.

$$wtPhaseInc = N\frac{f_o}{f_s} \tag{9.1}$$

When the *wtPhaseInc* value is much less than one, you interpolate multiple values between a pair of points, then move to the next pair. When the *wtPhaseInc* is much greater than one, you skip over multiple table cells and interpolate only between the last set – the number depends on the order of interpolation.

9.1.1 Wavetable Aliasing

There are two sources of aliasing: first, the wavetable may have been pre-computed with aliasing components already included. For example, sampling the trivial ramp LFO will produce aliasing in the resulting wavetable because the original signal was already aliased. Second, when the *wtPhaseInc* is very much greater than one, the index is making large skips through the table, causing it to appear under-sampled. In practice, if the table contains typical musical sounds with harmonics that decay rapidly or exponentially, then the aliasing will be subtle or perhaps masked by other harmonic components. Interestingly, to avoid large *wtPhaseInc* values, as the oscillator frequency increases, the table length should decrease.

9.1.2 Band-Limited Tables via the Fourier Series

An early approach to preventing aliasing involved pre-calculating the tables as band-limited Fourier series sums. These Fourier series summations have been in the literature for almost two centuries and allow you to easily generate a table of any length, filled with a preset number of harmonics and in harmonic ratios that generate the desired waveforms. The goal here is to synthesize waveforms that do not alias because their spectra are band-limited to the Nyquist frequency. The Fourier series representations for three common synth waveforms are shown in Figure 9.3. Notice that, for a wavetable, we only need a single cycle of the waveform so that f_o = 1 Hz, and k represents the harmonic number. If you want the fundamental plus 64 harmonics, k would vary from 1 to 64. To calculate the maximum number of harmonics below Nyquist for a given pitch, use Equation (9.2).

$$numHarmonics = \frac{f_s}{2(pitchHz)} \tag{9.2}$$

Figure 9.2 shows the 88-note keyboard with a few pitches and number of harmonics below Nyquist for f_s = 44.1 kHz. There are 128 MIDI notes in total; MIDI note 0 has the pitch of a C, 8.1758 Hz, while MIDI note 127's pitch is a G, 12.5438535 kHz. Equation (9.2) also shows that for any harmonic signal above ½ Nyquist can only contain a single harmonic component, or aliasing will occur. As you play successively higher notes, the waveform gets closer and closer to being a pure sinusoid.

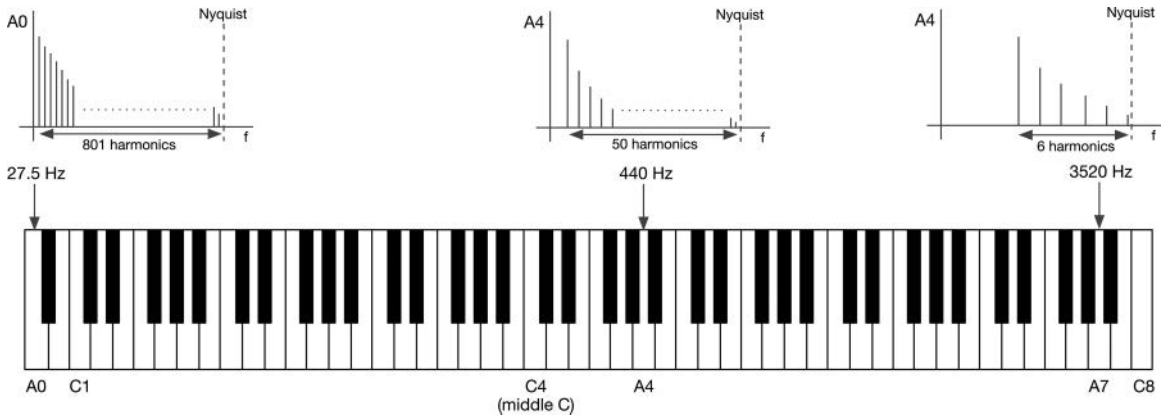


Figure 9.2 The 88-note piano keyboard starts at A0 (MIDI note 21), with 801 harmonics, and ends at C8 (MIDI note 108), with 5 harmonics; f_s = 44.1 kHz and A0, A4, and A7 are shown

Waveform	Fouier Series Formula
Ramp Up Sawtooth	$Y(n)_{SAW} = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{1}{k} \sin(k\omega nT)$ $= \left[\sin(\omega nT) - \frac{1}{2}\sin(2\omega nT) + \frac{1}{3}\sin(3\omega nT) - \frac{1}{4}\sin(4\omega nT) + \ldots \right]$
Triangle	$Y(n)_{TRI} = \sum_{k=0}^{\infty} (-1)^{k} \frac{1}{(2k+1)^2} \sin((2k+1)\omega nT)$ $= \left[\sin(\omega nT) - \frac{1}{9}\sin(3\omega nT) + \frac{1}{25}\sin(5\omega nT) - \frac{1}{49}\sin(7\omega nT) + \ldots \right]$
Square	$Y(n)_{SQUARE} = \sum_{k=0}^{\infty} \frac{1}{(2k+1)} \sin((2k+1)\omega nT)$ $= \left[\sin(\omega nT) + \frac{1}{3}\sin(3\omega nT) + \frac{1}{5}\sin(5\omega nT) + \frac{1}{7}\sin(7\omega nT) + \ldots \right]$

(9.3)

To generate wavetables on-demand or during synth instantiation, you implement the additive Fourier series equations (9.3) by sampling one cycle minus one sample of the desired waveform. For example, to synthesize the sawtooth waveform into an array with some number of harmonics (*numHarmonics*) and array length of *tableLength*, you would write this (*kPi* is the pi constant):

```
for (int i = 0; i < tableLength; i++)
{
     // --- sawtooth: += (-1)^g+1(1/g)sin(wnT)
     for (int g = 1; g <= numHarmonics; g++)
```

```
        {
            double n = double(g);
            double phi = 2.0*kPi*i*g / tableLength;
            sawTable[i] += pow(-1.0, (g + 1.0)) * (1.0/g) * sin(phi);
        }
}
```

In practice, these summations usually result in values greater than 1.0, so you need to normalize the array after synthesis. Notice that there is no parameter for sample rate in the accumulation equation. The sample rate is only required to calculate the maximum number of harmonics. Figure 9.3(a) and (b) show a ramp up (sawtooth) waveform synthesized with 14 harmonics, plus the fundamental; the number of ripples in the waveform and the number of harmonic spikes in the spectrum are the same (15). The harmonic envelope is shown in the dotted line and is correct for this waveform. The waveform in Figure 9.3(c) looks more like an analog sawtooth, with most of the ripples smoothed over, except the bumps at the top and bottom of the discontinuity.

The waveform and spectrum in Figure 9.3(c) and (d) are the result of applying the Lanczos correction function. The ripples in the time domain plot are at a maximum in the area immediately around the waveform discontinuity. Although discovered by Wibraham in 1848, this became known as the Gibbs Phenomenon (Hewitt & Hewitt). Lanczsos derived a method for smoothing the ripples in the time domain with the application of correction factors called σ factors. These are calculated using a *sin(x)/x* or *sinc* function, as shown in Equation (9.4). We will revisit Lanczos when we discuss virtual analog oscillators.

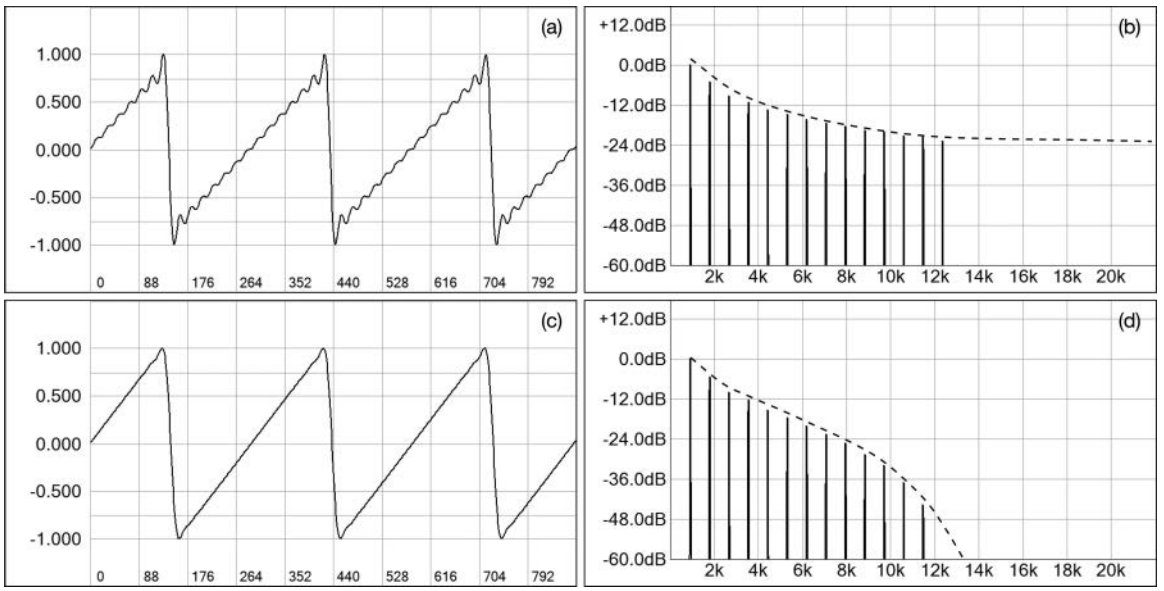


Figure 9.3 (a) Waveform and (b) spectrum of a sawtooth synthesized with 14 harmonics plus the fundamental and (c) the Lanczos corrected waveform and (d) resulting spectrum; the harmonic envelope is shown with a dotted line

$$\sigma_N = \frac{n\pi}{M} \quad \sigma_N = \frac{\sin(\sigma_N)}{\sigma_N} \tag{9.4}$$

M = number of harmonics
n = harmonic number
σ_n = Lanczos sigma for harmonic n

Adding the correction factors to the C++ code is not difficult; you simply multiply the factor to scale the *sin()* function. To add Lanczos correction to the previous code, you just add the factor calculation inside the inner loop.

```
// --- Lanczos Sigma Factor
double x = g*kPi/nHarms;
double sigma = sin(x)/x;
sawTable[i] += pow(-1.0, (g + 1.0))*(1.0/g) * sigma * sin(phi);
```

9.1.3 Multi-Wavetables

With the exception of the sine or cosine waveforms, which include only a fundamental and no harmonics, we cannot simply use one table for the entire range of MIDI notes without aliasing. This means that we will need multiple tables spread across the keyboard with various numbers of harmonics for a given target waveform/spectrum. In the earliest wavetable synths, when memory was expensive and limited, designers typically used either one table per octave or one table per minor third (four tables per octave). The other notes were interpolated between the tables, with the table limits at the top note, so interpolation was only performed to lower the pitch, and the *wtPhaseInc* value was less than 1.0. The 1986 Sequential Circuits Prophet VS had nine wavetables spread across the keyboard, as shown in Figure 9.4(a). Later, placing tables at minor third intervals became the industry standard.

With cheap memory today, we may ask: how many tables do we need to cover every MIDI note such that each has exactly the maximum number of harmonics below Nyquist? The answer depends on the table lengths, which also dictate the number of harmonics we may calculate with the Discrete Fourier Transform (DFT). If we hold the table lengths constant at N = 256 and f_s = 44.1 kHz, then there are 54 individual tables needed to cover every note. The first table is valid for the first 53 MIDI notes. Then, there is a single table for each note up to number 100. The last few tables are shared across several notes; the last table with only a single harmonic component will cover MIDI notes 125–127. My wavetable oscillator uses this approach and requires a set of tables to cover every note. The wavetable data source is set up so you may invent your own variations; if you like, you can just use one table across the entire keyboard, but be prepared for crispy aliasing.

9.1.4 Generating Interesting Tables

While the Fourier synthesis equations work for simple waveforms, harmonically rich and interesting waveforms require more work. Many of the wavetables in the Sequential Circuits Prophet VS were designed in the frequency domain first, then generated with additive Fourier synthesis. Today, we have numerous tools to allow the development of a practically unlimited number of

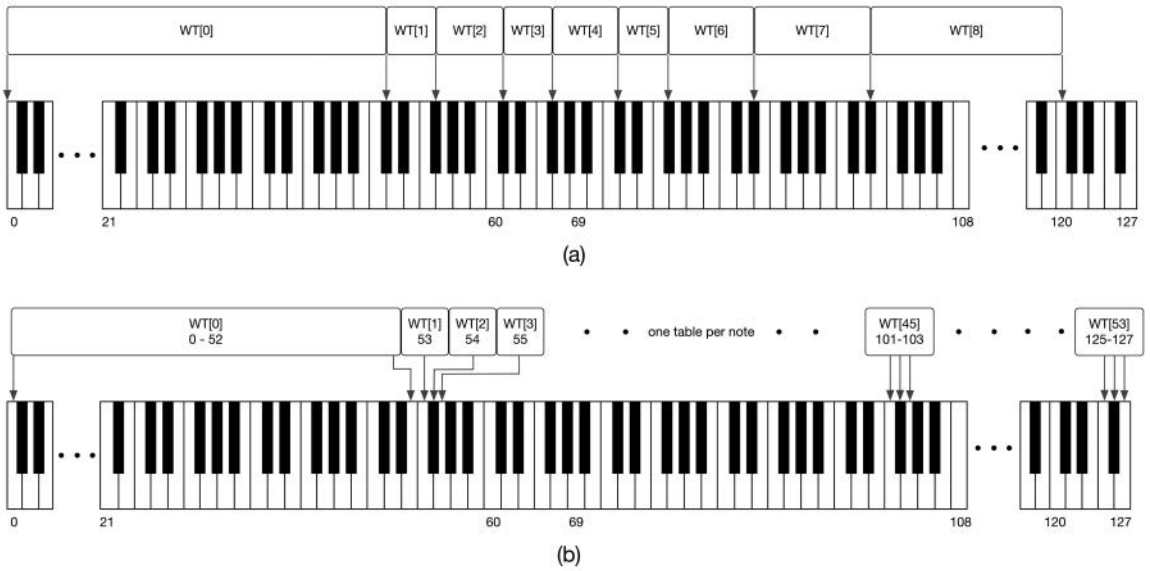


Figure 9.4 (a) Wavetable spacing in the Prophet VS and (b) modern wavetable synth using 54 high-resolution tables such that every note has every harmonic up to Nyquist

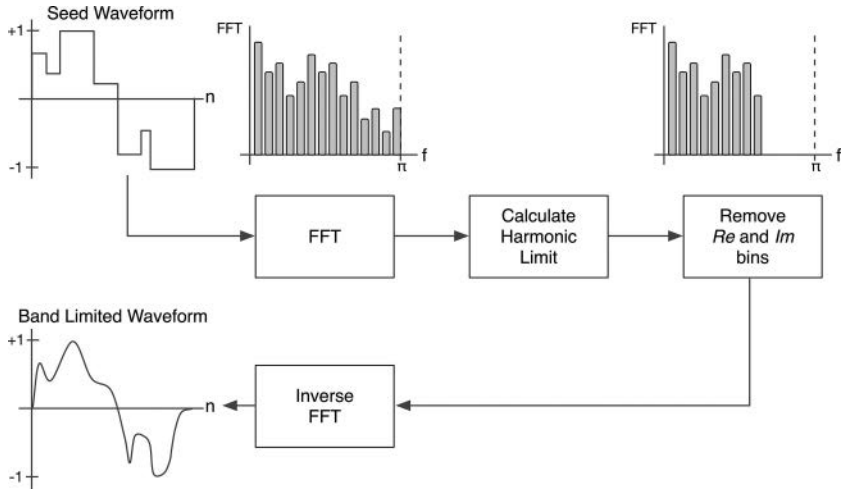


Figure 9.5 Flowchart of a system to generate band limited wavetables from a seed table; the process is repeated as many times as you need

interesting and harmonically rich waveforms. Figure 9.5 shows the flowchart for a typical sound development tool that starts with a seed waveform that is alias-free up to Nyquist. The FFT is taken and the harmonic limit calculated from the desired note pitch and the sample rate. Bins

above the limit are zeroed in the real (*Re*) and imaginary (*Im*) FFT arrays. The inverse FFT produces the band-limited waveform. Note that this method of filtering via bin removal will not work for filtering a continuous stream of data blocks, but for our single waveform, it is fine.

9.2 Wavetable Objects and Database

When using closely spaced, dense multi-wavetables and multiple voices, the sheer number of tables can grow out of control. When we implement morphing wavetable oscillator cores, we will need sets of wavetables to morph across. The SynthLab wavetable objects are stored in a database. Any wavetable core may create and add tables to the database, which has no maximum limit. All wavetable oscillators can share these wavetables. Simple interfaces are implemented so that you are not restricted to the SynthLab tables, and you may freely invent your own table readers and generators. The database is shared across all wavetable oscillators using the *std::shared_ptr*.

9.2.1 Static and Dynamic Wavetables

Ultimately, the wavetable is accessed as an array of *float* or *double* values. Static wavetables are prefabricated prior to runtime. They may be compiled into the plugin with a header (.h) file or as a binary resource. They may also be read at run-time out of a binary file that is stored in a known location. Dynamic wavetables are synthesized at run-time and may need to be re-synthesized if the sample rate changes. Some software synth plugins and apps that require large numbers of tables will synthesize them using the algorithm in Section 9.1.4 at startup, populating internal tables or storing them on disc.

9.2.1.1 StaticWavetable and DynamicWavetable Structures

The *StaticWavetable* structure is designed for statically declared wavetables. It is also designed to work with my TableMaker software, which generates wavetable sets that cover every MIDI note with non-aliasing harmonic content. The *StaticWavetable* structure includes pointers to either an array of *doubles* or an array of 64-bit unsigned integers or *uint64_t*. When creating the tables and storing arrays in header files or as binary data, the 64-bit *double* values are encoded directly as binary data. When writing to the file as hex encoded values, we get the maximum resolution of the data without needing the scores of digits that make up the fractional part when writing a *double* value as a hex-string. This becomes more important if you use TableMaker, which can also write binary files of data, including the ability to encrypt the tables. In addition, this structure holds a pointer to an array of *double* values so that you may use traditional C++ arrays of doubles as the wavetables. Both pointers are declared as *const* and always point to statically declared tables that do not need to be deleted.

The *DynamicWavetable* structure is identical to the static version in everything except for the fact that it holds a pointer to an array of double values that is not declared as *const*. Both structures contain additional information that aids in reading the tables. All SynthLab wavetables have power-of-two table lengths, and therefore, we may wire-and the updated read location to perform the circular buffer table wrap.

```
// --- StaticWavetable
const uint64_t*   uTable;
```

```
const double* dTable;
// --- DynamicWavetable
double* table = nullptr;

// --- common to both
uint32_t tableLength = kDefaultWaveTableLength;
uint32_t wrapMask = kDefaultWaveTableLength - 1;
doubleoutputComp = 1.0; // --- output scaling factor
doubletableFs = 44100.0;
const char* waveformName;
```

The *tableFs* is the sample rate used to synthesize the table. It is not used in any of the objects in SynthLab but is there if you need it. The waveform name is optional, but it is used in SynthLab to populate the dynamic waveform lists.

9.3 Wavetable Sources and Database

The *IWavetableSource* interface is used to create a C++ class that provides access to a set of wavetable structures. You can think of a "wavetable source" as all of the information needed to synthesize one waveform across the range of MIDI notes. This interface allows you to implement your wavetables however you wish and is one of the central design patterns in SynthLab. There are four wavetable source objects included as examples in Table 9.1, and studying these will help you see how the *IWavetableSource* interface is implemented.

The sine and drum sources shown in Figure 9.6(a) store only one table, which is used for all MIDI notes. The sine source interpolates the single table across the range of pitches while the drum source implements pitch-less tables, giving you two examples of simple but useful kinds of wavetables. The static and dynamic table sources each include an array of 128 *SynthLabWavetable* structures, one for each MIDI note, as shown in Figure 9.6(b), and you can see that the wavetable structures may point to the same table of data: for example, the first 53 structures all point to the same table because it contains the maximum number of harmonics below Nyquist. The tables are all read and interpolated the same way, so these sources differentiate between different kinds of tables. The engine's wavetable database, shown in Figure 9.6(c), consists of a set of wavetable sources indexed with unique names.

Table 9.1 The built-in *IWavetableSource C++* objects and descriptions

***IWavetableSource* Object**	**Description**
DynamicTableSource	Stores dynamically created tables and is used in the example module core that synthesizes parabola and triangle waveforms at startup; there is one table per octave of MIDI notes
StaticTableSource	Stores the static hex encoded wavetables that *TableMaker* produces, with each MIDI note table getting its own table that will not alias
DrumWTSource	A special static wavetable source for pitch-less tables, such as drums or sound effects
SineTableSource	Stores a single static sinusoidal table of *double* values that is used for all MIDI notes

The morphing core accesses tables in banks, shown with the dotted line in Figure 9.6(d), but it is important to understand that the bank is a conceptual idea that is simply a set of unique waveform names presented to the user as a single string (name). The *SynthLabBankSet* is a structure that contains an array of *SynthLabWavetables*, and it is only used to store tables and initialize the wavetable database with a set of tables in one function call. Once the tables are registered, the database treats them like any other wavetable and is not aware that they are grouped as a bank.

The *WavetableDatabase* is created in the *SynthEngine* and shared across all *SynthVoices* and their wavetable oscillators. The database is a *std::map* that stores *IWavetableSource* pointers via unique key strings. Your wavetable core object may query the database for table sources, and it may create and add new table sources at construction time. This means you are free to code your own object that exposes the *IWavetableSource* interface functions in Table 9.2 and add it to the database. You do not need to use my objects or my structures; you only need to provide the interface to your own implementation. The database uses unique name strings corresponding to the core module strings. SynthLab fills the GUI waveform selectors with these unique strings any time a new core object is loaded.

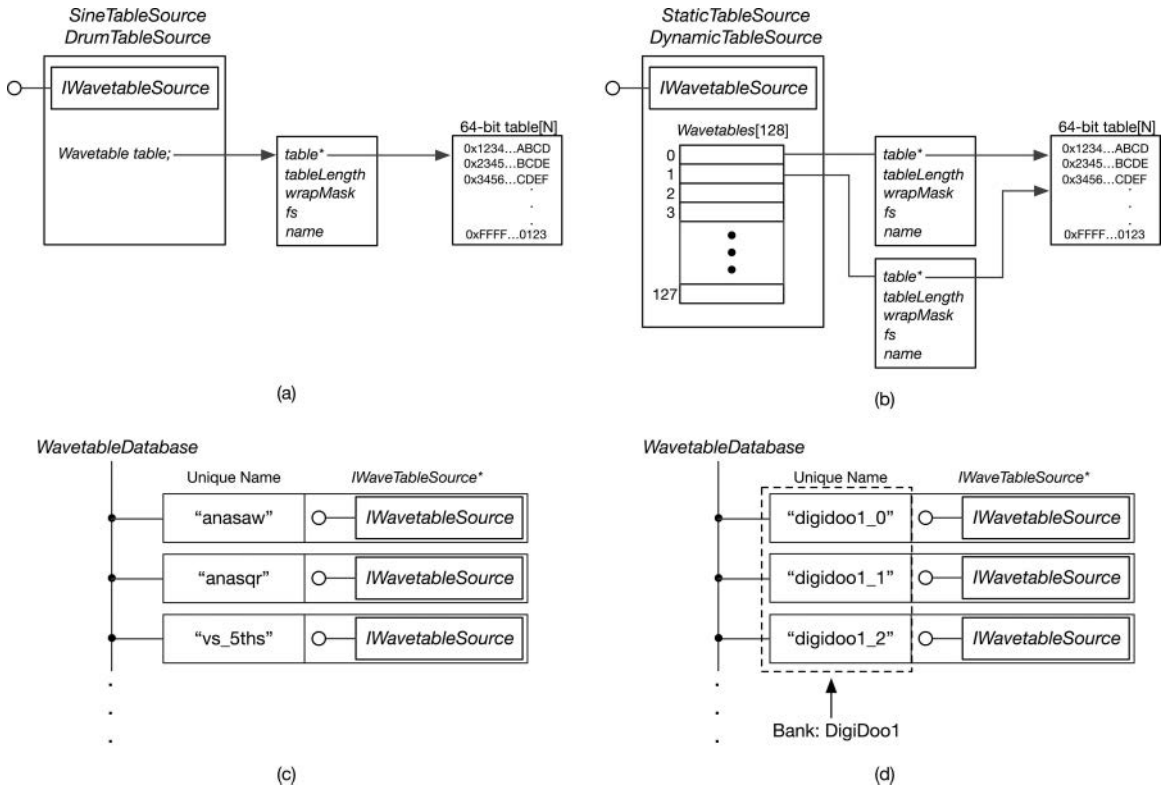


Figure 9.6 (a) The sine and drum sources only store and read a single wavetable, while (b) the static and dynamic sources each store an array of wavetables, one per MIDI note, while (c) the *WavetableDatabase* contains a dictionary of *IWavetableSource* pointers whose keys are unique strings that name the waveforms; (d) a wavetable bank is a name given to a set of wavetables used for wavetable morphing

Table 9.2 The *IWavetableSource* interface

***IWavetableSource* Function**	**Argument**	**Description**
selectTable	midiNoteNumber	Instructs the object to select a table and store it for rendering notes; the table is selected with the MIDI note number
readWaveTable	*readIndex*, which is a normalized value	Reads the wavetable using a *double* read index with an integer and fractional part; interpolation is used to find the value located between samples
getWaveTableLength	None	Returns the length of the currently selected wavetable
getWaveformName	None	Returns the waveform name for GUI

9.3.1 Reading/Interpolating the Wavetable

The *IWavetableSource* object needs to implement the *readWaveTable* function. As an example, the *SineTableSource* version is shown here. It performs the following:

- Calculate the read location as $N(f_o/f_s)$ where N is the table length in samples
- Use the *modf* function to split the read location into an integer and fractional part (*intPart* and *fracPart* respectively)
- The *intPart* is the index of the first sample to read
- Increment the *intPart* by one, then do the circular wrap with the mask to find the second sample to read
- Read both samples and perform linear interpolation that is the *fracPart* distance between them, producing the output value
- The output value is scaled using information saved with each table; this is to help you normalize the loudness of the wavetables, if needed.

```
// --- read and interpolate, return the audio sample
virtual double readWaveTable(double oscClockIndex)
{
      // --- two samples from table
      double wtData[2] = { 0.0, 0.0 };

      // --- location = N(fo/fs)
      double wtReadLoc = sineWavetable.tableLength * oscClockIndex;

      // --- split the fractional index into int.frac parts
      double intPart = 0.0;
      double fracPart = modf(wtReadLoc, &intPart);
      uint32_t readIndex = (uint32_t) intPart;
      uint32_t nextReadIndex =
                              (readIndex + 1) & sineWavetable.wrapMask;

      // --- two table reads
```

```
    wtData[0] = sineWavetable.dTable[readIndex];
    wtData[1] = sineWavetable.dTable[nextReadIndex];
    // --- interpolate the output (0.0 and 1.0 set the distance)
    double output = doLinearInterpolation(0.0, 1.0,
                                          wtData[0],
                                          wtData[1],
                                          fracPart);
    // --- scale as needed
    return sineWavetable.outputComp * output;
}
```

9.4 *WTOscillator* and Cores

Figure 9.7 shows the WTOscillator module, along with three of the five module cores.

Table 9.3 lists the GUI parameter structure and core descriptions. The *IWavetableSource* object performs the reading and interpolation. The oscillator cores are designed to generate distinctly different types of waveforms and demonstrate different kinds of tables, including one-shot and pitch-less tables. The *MorphWTCore* is a special variation and is used for wavetable morphing in Section 15.2. It registers sets of wavetables stored as banks and exposes bank names rather than waveform names. However, the same mechanism is used for the database storage and wavetable read/interpolate operations.

9.4.1 Wavetable Oscillator Controls, Modulations. and Features

Table 9.4 lists the module controls common to all cores and the GUI calculation section or equations. The coarse tuning and fine-tuning operate in semitones and cents, respectively, and are part of the pitch modulation update calculation, while the pan and output controls are common to almost all other oscillators and allow blending of the quad oscillator render block. Table 9.5 lists the cores and their special functionality.

Table 9.3 *WTOscillator* custom parameter structure and cores: (*) downloadable SynthLab-DM core project (*) downloadable dynamic module projects

Parameter Structure	Description
WTOscParameters	Used for all cores; includes all oscillator parameters
Example Core	**Description**
ClassicWTCore	16 waveforms based on analog and other classic table waveforms; demonstrates typical wavetable looping
MorphWTCore	The morphing oscillators in Section 15.2 use banks of waveforms instead of individual tables. This core demonstrates how to create and use these sets (banks) of waveforms for the morphing oscillators
*FourierWTCore**	Wavetables generated with Fourier synthesis to demonstrate dynamically created wavetables whose contents change if the sample rate changes
*DrumWTCore**	One-shot wavetables with single drum sounds mapped across all keys using PCM samples of classic electronic drums
SFXWTCore	Bonus core of sound effects taken from PCM samples, demonstrating how to use pitch-less, one-shot wavetables

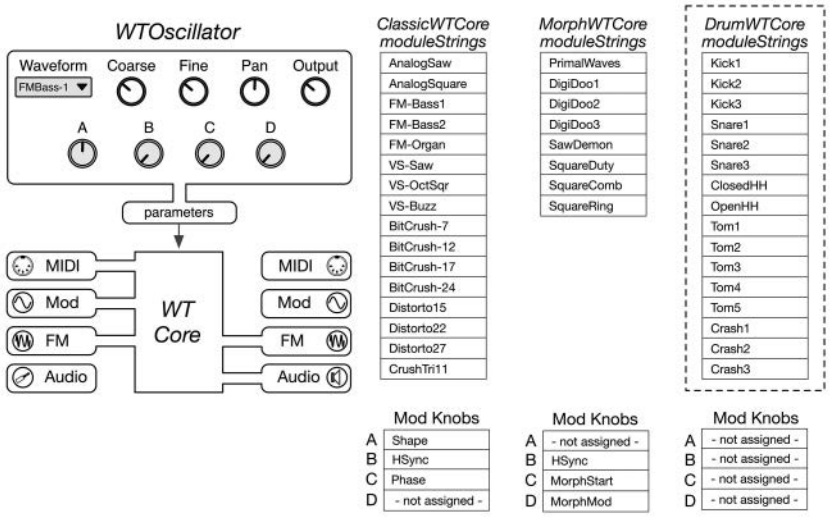


Figure 9.7 The *WTOscillator* and the three included cores: classic, morphing, and drums; two special bonus cores are also included for Fourier synthesis and sound effects

Table 9.4 Module controls shared across cores

Module Control	Description	GUI Calculation (Section)
Coarse	Coarse tuning in semitones	Pitch modulation (6.7.1)
Fine	Fine-tuning in cents	Pitch modulation (6.7.1)
Pan	Left/Right pan	Constant power panning (6.2.2)
Output	Output level in dB	dB to raw: $10^{(dB/20)}$

Table 9.5 Wavetable cores and their special modulation capabilities

Core	Features and Special Modulations (Section)
ClassicWTCore	Self hard-sync modulation (6.9.1)
FourierWTCore	Shape modulation with phase distortion (6.8) Phase offset that moves initial wavetable read index
MorphWTCore	Self hard-sync modulation (6.9.1) Waveform morphing modulation (15.2) with morph start and morph intensity controls
DrumWTCore *SFXWTCore*	Bare minimum implementation without modulation; best place to start when examining the core code

9.5 Wavetable Core Programming Notes

With the C++ files open, compare the programming notes with the code that you see, starting with the class descriptions of each core. The wavetable core objects all use the same time-base, implemented with a *SynthClock* modulo counter that is reset and advanced with each note-event.

The cores that support self hard-sync modulation include a *Synchronizer* object to help implement that function. All cores except the morphing core include a member that acts as the selected wavetable source. The morphing core includes an array of two wavetable sources, which it interpolates between to create the wave morphing effect. Sections 9.5.1 through 9.5.5 summarize the five operational phases, plus the constructor for the wavetable cores. Make sure to examine the code while digesting their operational phase details.

9.5.1 Construction Phase

All cores:

- The module strings are the waveform names, and their index values map to the GUI control selection values; the strings will be used to query the database for the selected waveform
- The mod knobs are not used in the drum and sound effect cores
- The cores demonstrate multiple was to assign module strings, from string literals to bank array names (morphing core)

9.5.2 Reset Phase

All cores:

- Reset clocks and synchronizer
- Query database for tables and register if not found; this minimizes the number of tables required for operation
- Each core demonstrates a different way to add tables and recall their name strings

Fourier WT core:

- Recalculate the wavetables when the sample rate changes

To query the database for an *IWavetableSource* pointer, use the *wavetableDatabase* interface pointer and the unique name string:

```
processInfo.wavetableDatabase->getTableSource("FMBass-1");
```

If the function returns *nullptr*, you can add the table using a table source and name string. This code uses an old-fashioned pointer and the *new* operator rather than shared pointers; this is a requirement of the dynamic module core loading system, which must implement an API interface that does not use the standard template library (*std::*) or other third-party libraries that may corrupt the heap when calling functions and passing pointers across the thunk layer.

```
StaticTableSource* wt = new StaticTableSource;
wt->addSynthLabTableSet(&fmbass1 _ TableSet);

processInfo.wavetableDatabase->addTableSource("FMBass-1", wt);
```

9.5.3 Note-On Phase

The note-event begins with a note-on handler that is both similar and quite simple across the cores:

- Store MIDI pitch (except drum and SFX cores)
- Reset the SynthClock modulo counter member to either 0.0 (all cores) or the starting phase point (classic, morphing, and Fourier cores); notice how the mod knob C value is used to calculate the phase offset for the cores that support it
- NOTE: there is nothing to do in the note-off phase

```
// --- parameters
midiPitch = processInfo.noteEvent.midiPitch;

// --- reset to new start phase
if(processInfo.unisonStartPhase > 0.0)
     oscClock.reset(processInfo.unisonStartPhase / 360.0);
else
     oscClock.reset(parameters->modKnobValue[MOD_KNOB_C]);
```

9.5.4 Update Phase

The pitched core update phases follow the same four steps that you will find on the other pitched oscillators. The *update* function's main goal here is to have the correct wavetable selected and the correct interface pointer stored for the render phase that will immediately follow.

1 Calculate the pitch modulation value based on GUI controls and pitch modulation sources
2 Select the wavetable source based on the user's waveform selection string; the morphing core selects two sources
3 Select the wavetable based on the final modulated pitch value
4 Set up hard synchronizer, shape, or other modulation variables

The classic core wavetable selection code follows, starting with the final oscillator frequency calculation that sets the oscillator time-base – this is identical across all pitched oscillators that use the *SynthClock* time-base object.

```
// --- calculate the modulated pitch value
double oscillatorFrequency = midiPitch*pitchShift;

// --- BOUND the value to our range
boundValue(oscillatorFrequency, WT_OSC_MIN, WT_OSC_MAX);

// --- phase inc = fo/fs
oscClock.setFrequency(oscillatorFrequency, sampleRate);
```

The next step is to select the table source using the user's GUI selection index value, which corresponds to the GUI control that holds the 16 waveform strings. The old-fashioned *const char** is used due to the same dynamic core and thunk issues, as described in the previous section.

```
// --- select the wavetable source
const char* wave = coreData.moduleStrings[parameters->wavetableIndex];

selectedTableSource =
              processInfo.wavetableDatabase->getTableSource(wave);
```

Finally, the exact table is selected based on the MIDI note number of the modulated pitch value. Check the morphing core code to see how it selects two tables based on the morphing index, as detailed in Section 15.4.3.

```
// --- select table
uint32_t midiNote = midiNoteNumberFromOscFrequency(oscillatorFrequency);

selectedTableSource->selectTable(midiNote);
```

The rest of the update phase involves only simple gain and pan calculations or storing mod knob values for later.

9.5.5 Render Phase

All pitched oscillators must render samples one block at a time and write stereo audio data to their output buffers. A simple *for-loop* sets up the block render operation. All wavetable cores use a separate *renderSample* function so you can easily modify them without changing the for-loop code. The drum and sound effect cores have the minimum implementation, and both operate in one-shot mode so the oscillator only renders the table once, then outputs zeros thereafter. The *SynthClock* time-base *advanceWrapClock* function returns true if the modulo counter has wrapped, indicating that the one-shot event is complete. The drum and sound effect cores store this value.

```
double DrumWTCore::renderSample(SynthClock& clock)
{
     // --- read source
     double oscOutput =
            selectedTableSource->readWaveTable(clock.mcounter);

     // --- advance and wrap clock; save wrap notice for one-shot
     oneShotDone = clock.advanceWrapClock();

     return oscOutput;
}
```

The other three cores implement the same function plus an additional method for the self hard-sync operation; make sure you compare the hard-sync function with the code and modulation description in Section 6.9. First, the buffers are accessed, and the for-loop is set up:

```
float* leftOutBuffer = processInfo.outputBuffers[LEFT_CHANNEL];
float* rightOutBuffer = processInfo.outputBuffers[RIGHT_CHANNEL];
```

```
for (uint32_t i = 0; i < processInfo.samplesToProcess; i++)
{
```

Next, the hard sync ratio value is checked, and the corresponding function calls are made. These oscillators use phase distortion for shape modulation so the oscillator shape parameter is a function argument:

```
    // --- render the saw
    double oscOutput = 0.0;
    if (parameters->hardSyncRatio > 1.0)
        oscOutput = renderHardSyncSample(oscClock,
                                         parameters->oscillatorShape);
    else
        oscOutput = renderSample(oscClock,
                                 parameters->oscillatorShape);
```

Lastly, gain and panning are applied; notice that the glide modulator is advanced outside the loop as it is a granulized, slow-moving modulation source.

```
    // --- scale by gain control
    oscOutput *= outputAmplitude;

    // --- write to output buffers
    leftOutBuffer[i] = oscOutput * panLeftGain;
    rightOutBuffer[i] = oscOutput * panRightGain;
}

// --- advance the glide modulator
glideModulator->advanceClock(processInfo.samplesToProcess);
```

9.6 Exercises

9.6.1 SynthLab-WT$_{DM}$ Drum Core: Mod knob D to Reverse Playback

Mod knob D is unassigned in the all of the wavetable cores. Download the drum core sample dynamic module code, then compile and load it to test. When you are satisfied that it generates drum sounds correctly, add functionality such that when mod knob D has a value that is greater than 0.5, the tables play in reverse. HINT: the *SynthClock* time-base is fully capable of running backwards.

9.6.2 SynthLab-WT$_{DM}$: Pan Modulation

The classic wavetable core implements shape modulation as its unique modulation destination and is set up in the modulation matrix. Set up a dynamic module project that implements the classic wavetable core, then change the module name so you can differentiate between them in the SynthLab-WT$_{DM}$ project. Test the shape modulation using an LFO as the modulation source and the OSC1, 2, 3, or 4 "mod" destinations. Next, alter the code to replace the shape modulation with pan modulation, using the existing panning code as a basis. Test the pan modulation with the same LFO. Shape and pan are both examples of bipolar modulation destinations.

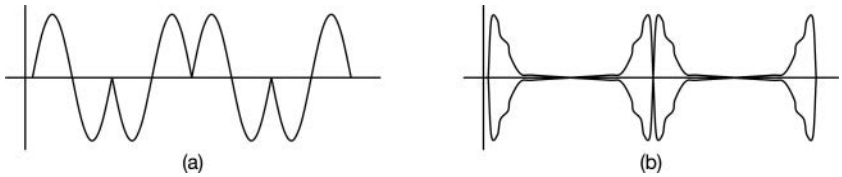


Figure 9.8 Wave scanning may be used with simple waveforms (a) or more complex one-shot drum tables (b) to produce new sounds

9.6.3 SynthLab-WT$_{DM}$: Modify the Wavetable Source

The *DrumWTSource* C++ object implements the pitch-less, one-shot drum wavetable sources. Examine the drum core's constructor, and note the first few lines where the tables are added to the source with *addWavetable*. Trace this code into the *DrumWTSource*, and you will see that each table is stored as a separate waveform that is mapped across the keyboard. Alter the source so that the drum tables map chromatically to the first 16 MIDI keys, starting with middle C (60).

9.6.4 Advanced Module: Create a Wave Scanning Oscillator

Wave scanning is a wavetable variation that generates a new and interesting set of sounds from the existing tables without needing to create or modify new tables. It works by reversing the wavetable read direction each time the read index wraps around the table boundary. This forwards-backwards-forwards scanning motion produces a kind of double-waveform where every other waveform is reversed in time, as shown in Figure 9.8(a). A more complex table, such as the one-shot drum tables in Figure 9.8(b), usually results in very strange and interesting sounds. Note that this operation can cause kinks or discontinuities in the output waveform that may produce aliasing components. Create your own novel wavetable core that implements wave scanning in the table source. Use mod knobs to control aspects of the scanning, such as the re-scan threshold index – the table index that signifies the direction reversal.

Bibliography

Hewitt, Edmund & Hewitt, Robert. 1979. "The Gibbs-Wilbraham Phenomenon: An Episode in Fourier Analysis." *Archive for History of Exact Sciences*, vol. 21, no. 2, pp. 129–160. New York: Springer.

Kleimola, Jari. 2005. "Design and Implementation of a Software Sound Synthesizer," Master's Thesis, Supervised by Välimäki, Vesa. Helsinki University of Technology. http://lib.tkk.fi/Dipl/2005/urn007886.pdf, Accessed October 14, 2020

Lanczsos, C. 2010. *Applied Analysis*. New York: Dover.

Moore, Richard. 1990. *Elements of Computer Music*, Chap. 3. Eaglewood Cliffs: Prentice-Hall.

Moorer, James. 1976. "The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulae." *Journal of the Audio Engineering Society*, vol. 24, no. 9, pp. 717–727.

Roads, Curtis. 1996. *The Computer Music Tutorial*, Chap. 3. Cambridge: The MIT Press.

Stilson, Tim & Smith, Julius O. 1996. "Alias-Free Digital Synthesis of Classic Analog Waveforms." Proceedings of the 1996 International Computer Music Conference.

10 Virtual Analog Oscillators

Virtual Analog (VA) oscillators are interestingly named. They have nothing to do with analog circuits, components, block diagrams, or signal flow. They do not model the physical characteristics of any kind of analog oscillatory system, like pendulums or springs with attached masses. Analog oscillators have no Nyquist limit and cannot alias. Their waveform shapes look identical to those of trivial oscillators, no matter what the frequency of oscillation. One of the goals for VA oscillators is to preserve that perfect shape as much as possible. The applications of VA oscillators tend to take two directions. One is to retain the perfect analog oscillator waveform and harmonic spectrum with either no aliasing or with aliased components so low in amplitude that they are either inaudible or masked by other non-aliased components. Another direction is to create waveforms that are cheap to render on low-CPU and memory-starved devices, such as mobile phones, tablets, and toys/games. In these cases, aliasing is allowed, even if it is audible, because of the tradeoff in CPU and memory savings.

10.1 VA Oscillator Fundamentals

VA oscillator analysis starts with the notion that discontinuities in a waveform are the primary cause of aliasing. The trivial ramp LFO waveform exhibits massive aliasing, as shown in Figure 10.1(a) and (b). The lower limit for the plot in Figure 10.1(b) is −96 dB, the theoretical noise floor for 16-bit PCM digital audio, while it is −60 dB in Figure 10.1(a), which is used throughout the text. Lowering the measurement floor reveals more aliasing, though aliased spectral amplitudes below −60 dB are extremely suppressed, and I usually use −60 dB as a ballpark threshold on the tolerable aliasing amplitudes.

Figure 10.1(c) and (d) show the same trivial oscillator output whose waveform has been altered in the time domain to suppress the aliased components. The VA oscillator attempts to filter the aliased components out by manipulating the points around the discontinuity.

10.1.1 Saw and Square Continuities

Traditionally, a discontinuity in a function is a point in the domain where the function instantaneously jumps from one value to another, with no time or space between the level shift. Some say that the function takes on both values at once. But according to a less rigorous definition, a discontinuous edge is a place where two geometrical entities – lines or curves – meet up. It can also describe a single function that contains a glitch. Types of discontinuity are named "continuities" as they relate to how continuous a function may be. In Figure 10.2(a), you see a smooth and continuous

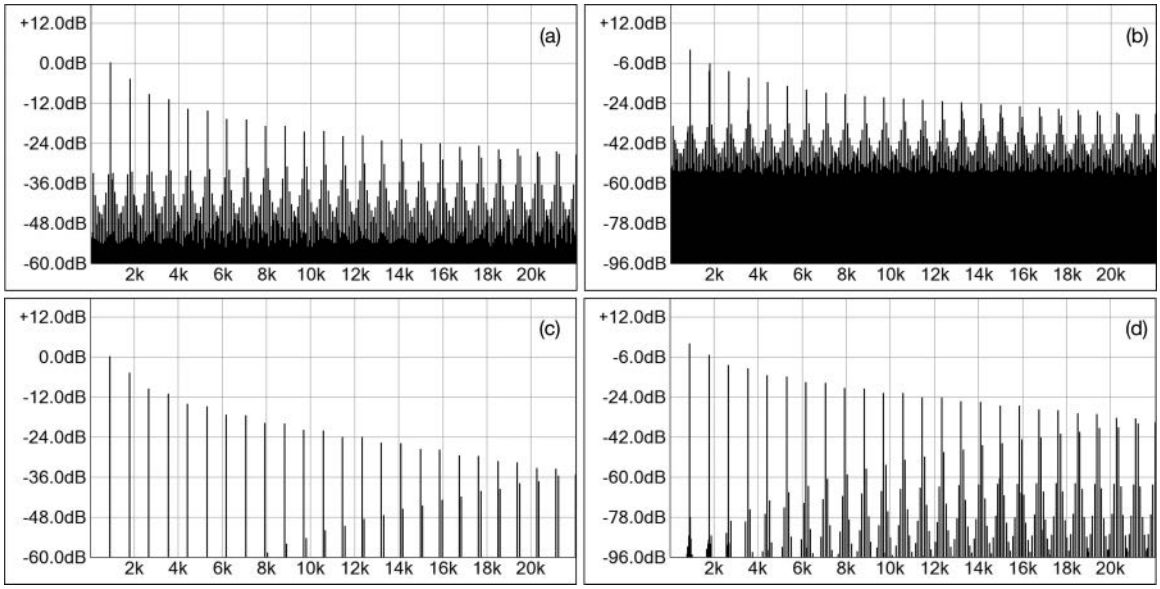


Figure 10.1 A trivial sawtooth waveform spectrum with (a) −60 dB and (b) −96 dB lower limits, and a VA oscillator spectrum with (c) −60 dB and (d) −96 dB lower limits

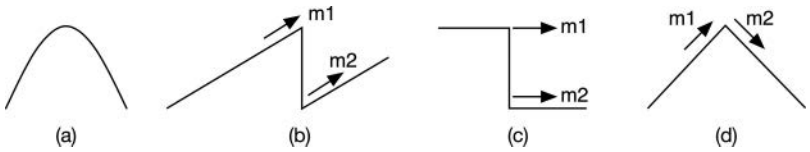


Figure 10.2 (a) A smooth continuous function, (b) a ramp discontinuity, and (c) square discontinuity both have similar mathematical traits – identical slopes on each side of the edge, while the triangle in (d) has two different slopes on each side of the discontinuity

function. The sawtooth and square discontinuities have the same slopes (*m1* and *m2*) on either side of the discontinuity and are said to be *C1* continuous (the 1 stands for first derivative or slope). The triangle in Figure 10.2(d) is not *C1* continuous.

If we limit the discussion to the C1 continuity, we can use some clever math to adjust the points around the discontinuity so as to smooth over the offending edge of a trivial oscillator. We applied a naïve version of this idea with the linear crossfading technique for hard sync in Section 6.10, smoothing over a discontinuity to reduce aliasing. There are a few approaches to VA oscillator design. Perhaps the earliest attempt is Lane's distorted and filtered sinusoids, which was followed up by Stilson and Smith's Band Limited Impulse Train (BLIT) method.

10.2 Band Limited Impulse Train (BLIT)

The idea behind BLIT is to generate a softened or rounded discontinuity that would have resulted in lowpass filtering an impulse and then produce a stream of filtered impulses. Equation

(10.1) produces a stream of *sinc* function-shaped pulses, as shown in Figure 10.3(a). If you invert every other *sinc* pulse, you arrive at the bipolar BLIT waveform of Figure 10.3(b). With the pulse train established, synthesizing the three basic waveforms of sawtooth, square, and triangle are all performed with integration (these are fundamental relationships you can find in many math and engineering texts), as shown in Figure 10.3(c). The integration is performed in the digital domain using reverse Euler or bilinear integrators.

$$y(n) = \frac{M}{P} Sinc_M\left[\frac{M}{P} n\right]$$
$$Sinc_M(x) \triangleq \frac{\sin(\pi x)}{M \sin(\pi x / M)} \tag{10.1}$$

BLIT has problems that make it one of the more difficult algorithms to implement. First is the generation of the BLIT sequence itself. Using the *sinc* function requires two calls to the *sin* function. Another idea is to sample a sum of windowed *sinc* functions and store them in a set of tables corresponding to different fractional shifts of the impulse. This is necessary because the period of the desired waveform may not be an integer multiple of the sample rate. Thus, the tables would need to be interpolated at runtime. This is known as BLIT-SWS. A larger issue is that the band-limited impulses overlap, and the tail of one must be mixed with the head of the next. This requires knowing when the next discontinuity is going to occur as well as look-ahead information. This can be problematic because we would like to have zero delay in synthesizing the waveform. The band-limited impulse has to be generated for each discontinuity, so, as the frequency becomes higher, the CPU usage increases. As Andy Leary points out, another issue with these methods is that the integration steps in BLIT and BLIT-SWS introduce an unwanted and undesirable DC offset into the signal at oscillator startup time.

10.3 Band Limited Step (BLEP)

Eli Brandt improved on the idea by performing the integration on a single *sinc* function in Figure 10.4(a), then storing that pre-integrated waveform in Figure 10.4(b). This removes the

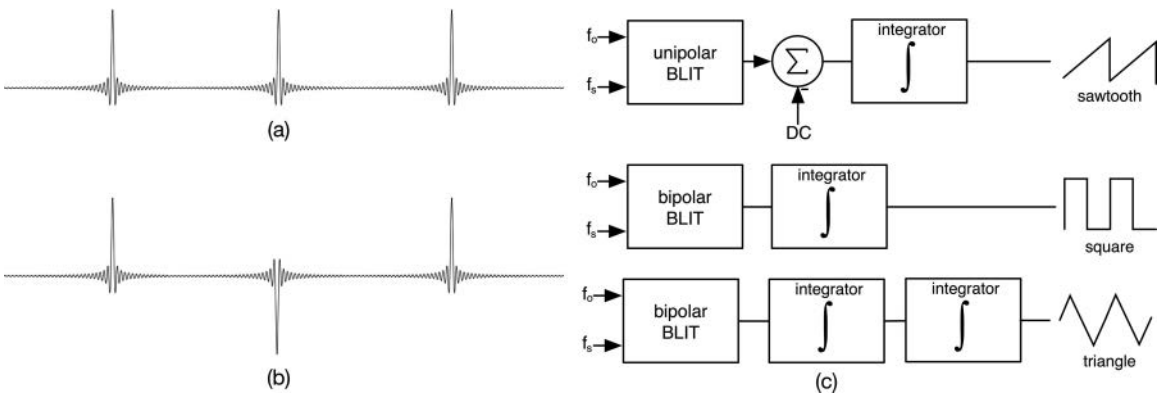


Figure 10.3 (a) Unipolar and (b) bipolar BLIT signals, and (c) algorithms for converting them into traditional waveforms

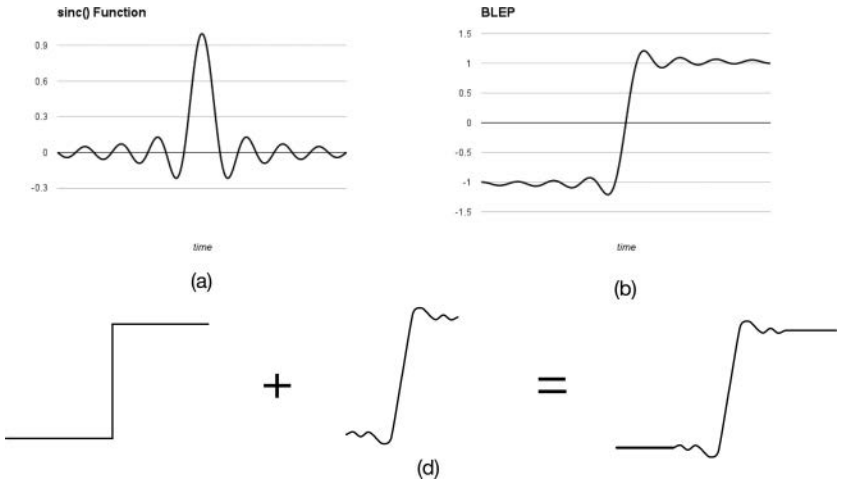


Figure 10.4 (a) A *sinc* function and (b) its integration produces a wavy step; (c) to use the BLEP, superimpose it on top of the discontinuity; notice that only a small section of the BLEP waveform is used

requirement for integration on the fly. This is called a Band Limited stEP function or BLEP. In this method, there is no generation of an impulse train. Instead, you start with a trivial oscillator, then use the stored BLEP waveform to make corrections to points that are on either side of the discontinuity. One way to imagine this is to take a piece of the BLEP step edge and superimpose it on the offending discontinuity, as shown in Figure 10.4(c). You can see that only a small piece of the BLEP function has been applied to the step edge; in practice, you will only need to alter a few points on each side so you don't need the complete BLEP. This also preserves that perfect analog shape for most of the waveform. In my BLEP oscillators, I allow correction of one, two, or four points on each side of the discontinuity, depending on the oscillator's frequency.

For implementation, a preferable solution to superimposing the BLEP edge involves windowing the *sinc* pulse to a certain width, performing the integration, and subtracting out the unit step to create a residual signal of correction factors that are simply added or subtracted from the waveform. The windowing is performed on the *sinc* function zero-crossing boundaries. Figure 10.5 shows two different windowed *sinc* functions; in Figure 10.5(a), only the center pulse of the *sinc* function is windowed and integrated. This produces a residual for correcting one sample on each side of the discontinuity. In Figure 10.5(b), the windowed *sinc* function is widened to include the next lobe zero-crossings. This produces a residual for correcting up to four samples on each side of the discontinuity (when doubling the window size while keeping the same table length, the zero crossings fall on two-sample boundaries).

10.3.1 Using the BLEP Residual

Figure 10.6 shows the idea behind the residual method, starting with the isolated rising edge of a trivial square-wave oscillator in 10.6(a) and deciding to alter two points on each side of the discontinuity. Lining up the residual so its discontinuity is on top of the trivial oscillator discontinuity is shown in Figure 10.6(b). The arrows line up exactly with the sample locations, and the residual

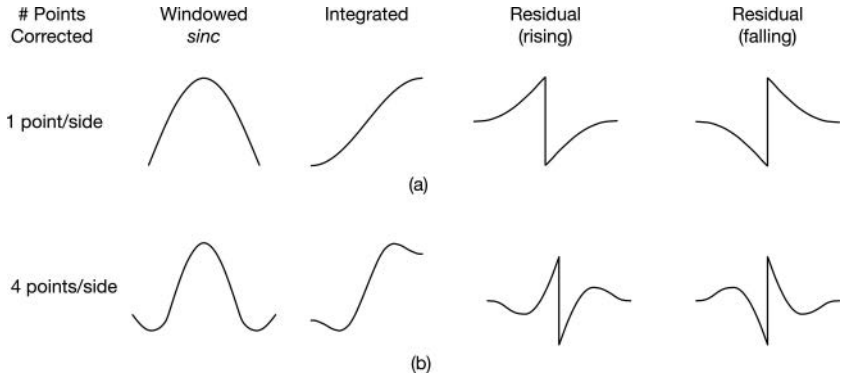


Figure 10.5 Two sets of windowed BLEP waveforms, where (a) the central pulse is truncated and windowed and (b) the central pulse is combined with the next pair of zero-crossings to form a wider windowed function; the integrated windowed signals, and the rising and falling residuals are shown for each case

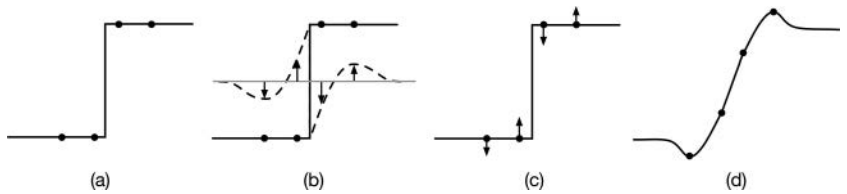


Figure 10.6 (a) A rising discontinuous edge showing the two samples on each side (b) superimposing the residual over the discontinuous edge (c) adjusting these four points based on the residual and (d) the BLEP corrected discontinuity

curve shows how much to alter each point in Figure 10.6(c). After only four points have been altered, we arrive at Figure 10.6(d), which mimics the result of superimposing the BLEP edge onto the waveform, as shown in Figure 10.4(c).

10.3.1.1 Strategy: One Large BLEP Table

In BLEP table-based approaches, you store the BLEP residual and use it accordingly. One strategy that I use for the SynthLab-VA project involves storing a single, finely sampled version of the BLEP residual in a 4,096-point table. There are two sets of tables: one for the correction of one point per side and the other for the correction of one, two, three, or four points per side of the discontinuity. They are generated with two different window widths. Within the second set of tables are more tables, all the same length but processed with different windows during table creation. Figure 10.5 uses simple rectangular windowing of the BLEP central pulse and its side pulses. But there are numerous other windows that may be used. The *sythconstants.h* file includes these tables:

- A single table using rectangular windowing on the central *sinc* pulse, which allows for the correction of one point per side; the table is named *dBLEPTable*

- Seven (7) tables for correcting one to four points per side, with the window name coded in the table name, e.g. *dBLEPTable_8_HANN* for the Hanning window and *dBLEPTable_8_BLKHAR* for the Blackman-Harris window

10.3.1.2 Strategy: Many Small BLEP Tables

Another approach involves storing many small tables of residual samples. In 2009, Andy Leary and Charlie Bright of Korg Research and Development were awarded US Patent 7,589,272, assigned to Korg, Inc. for *Bandlimited Digital Synthesis of Analog Waveforms*, which describes both a method for generating the band-limited signals with BLEP and a method for hard-syncing two oscillators with discontinuous waveforms. This patent describes an implementation in a hardware format with shift registers and accumulators (there is no code in the patent), and is certainly worth reading and experimenting.

Instead of consuming memory for large, finely sampled tables, another approach may be taken, noting that, at any given time, you are only correcting a handful of points around the edge – why store gigantic tables to process only a few points? Suppose you want to correct two points per side of the discontinuous edge. Instead of one giant table, you can implement a set of very small tables of only four points each, two points per side. Then, you can use interpolation between the tables to generate the missing data you would normally get with a large table. This approach saves memory but requires inter-table interpolation. As a point of reference, the BLEP functions in SynthLab will allow you to interpolate the 4,096-point table if you like – I've found that interpolation is inaudible because the table is so finely sampled. You can find much more information about optimal table sizing and BLEP approaches in Pekonen et al. (2011)

10.3.2 BLEP Correction Calculation

Regardless of whether you are using a single large table or many small tables, one common problem exists: you need to know how far away you are from the edge of the discontinuity, and this implies that you need to look ahead when evaluating the modulo counter's value. Once you have that edge distance, you can look up the correction value however you wish. The Leary-Bright patent demonstrates short transversal delay lines to facilitate looking ahead in the signal. Since we are using C++, we have another option:

Monitor the value of the trivial oscillator output (the modulo counter), and detect when you are one, two, three, or four samples away from the edge – this is possible because you know the clock's *phaseInc* value: for example, if the current modulo counter plus the *phaseInc* is greater than 1.0, then you know you are within one sample of the discontinuity. In Figure 10.7(a), you can see how this works diagrammatically. The discontinuous edge occurs in the center when the modulo counter rolls over from 1.0 back to 0.0, and the distance from the point on the left side is $t(-)$, while that on the right is $t(+)$.

There is a final issue involved that is inescapable: the oscillator's desired frequency. It is simple to calculate the number of points per waveform cycle, which will then determine how many points per *side* of a discontinuity we may correct. This may be BLEP's main drawback for high fidelity synthesis. Therefore, we need to follow the rules in Table 10.1, correcting either one, two, or four points per side.

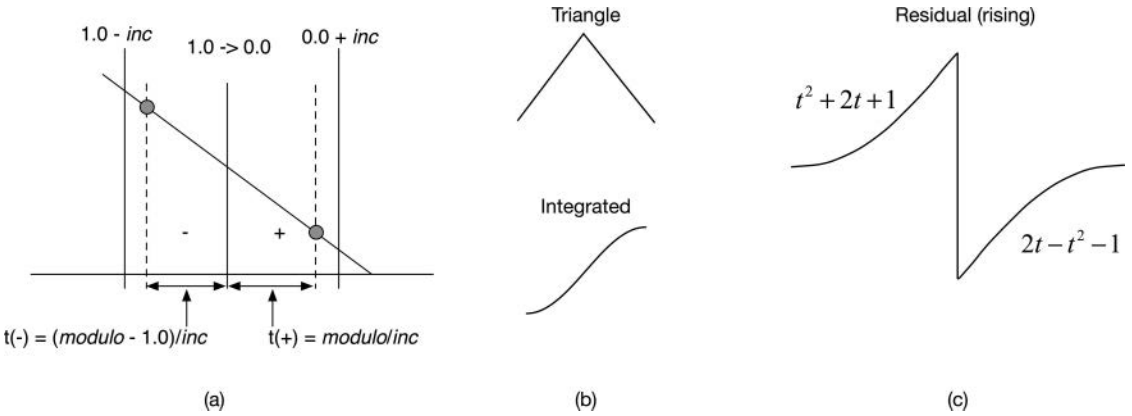


Figure 10.7 (a) Geometrically determining the distance from the discontinuity to points on either side of the edge; (b) a triangle pulse integrates to a smooth sigmoid, similar to that of Figure 10.5(a); and (c) the BLEP rising edge residual is approximated with two second order polynomials, one for each side of the edge

Table 10.1 Number of BLEP correction points per side versus oscillator frequency

Oscillator Frequency	*Number of Points Corrected per Side*
$f_o < f_s/8$ (1/4 Nyquist)	4
$f_o < f_s/4$ (1/2 Nyquist)	2
$f_o > f_s/8$	1

10.4 Polynomial BLEP Approximation

Välimäki proposes implementing the BLEP residual by approximating the original *sinc* function with a polynomial and names it *PolyBLEP*. Different polynomials of different orders may be chosen, but a simple solution is to use a unipolar triangular pulse, shown in Figure 10.7(b); this is the linear approximation of the windowed *sinc* pulse in Figure 10.5(a). Integrating this triangle produces the sigmoid (*s*-shaped) curve in Figure 10.7(b), which can be expressed in closed form and thus requires no table. Subtracting out the perfect step and then converting to bipolar results in the following two-part residual, shown in Figure 10.7(c). Equation 10.2 shows the polynomials used to compute each of the curves in Figure 10.7(c). The independent variable is t (time) as this was Välimäki's notation. For our purpose, t is the distance from a point to the discontinuous edge.

$$PolyBLEP(t) = \begin{cases} t^2 + 2t + 1 & -1 \le t \le 0 \\ 2t - t^2 - 1 & 0 < t \le 1 \end{cases} \tag{10.2}$$

The one-point-per-side *PolyBLEP* produces only a tiny bit more aliasing than you see in Figure 10.1(c), so when only correcting one point per side, it comes close enough to the

rectangular-windowed *sinc* pulse that I use it for the *VAOscillator* core object. For two or more points per side, I use the wider windowed tables instead.

10.5 Choosing the BLEP *sinc* Source

Interestingly, using the perfect *sinc* pulse as the basis for the BLEP correction does not necessarily produce the best results. There are numerous factors and tradeoffs involved. One approach involves trying different windows on the ideal *sinc* pulse, while another involves starting with traditional FIR lowpass filter design and then varying design parameters, such as pass-band ripple, stop-band attenuation, or transition width. The results of variations upon these approaches can be found in Otalvara et al. (2016), in which the psychoacoustic masking curves are used as evaluation criteria for determining the audibility of aliased components, with preservation of harmonic envelope and significant aliasing reduction acting as the dual design targets.

SynthLab includes a set of windowed sinc pulses, as listed in Section 10.3.1.1. If we focus only on four points per side correction, we may glean insight from the full range spectral plots in Figure 10.8. The majority of aliasing occurs below the −60 dB line. In addition, we observe that there is a tradeoff between preserving harmonic envelope and aliasing; the Hann windowed spectrum has the least aliasing but the poorest preservation of harmonic envelope. Where have we seen this before?

10.5.1 BLEP and Lanczos

Compare the plots in Figure 10.8(c) and (d) to the Lanczos sigma corrected Bandlimited wavetable in Figure 9.3(c) and (d). Notice also how the Lanczos sawtooth waveform (9.3c) has small wavy

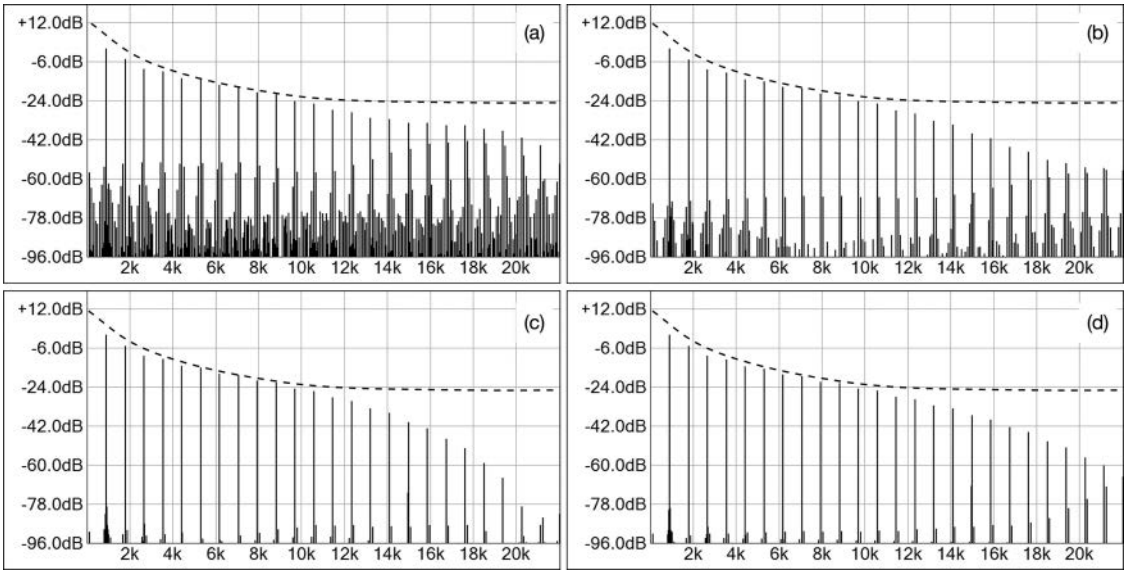


Figure 10.8 Sawtooth waveforms using 8-point BLEP correction with (a) rectangular (b) triangular (c) Hamming and (d) Blackman-Harris windowed *sinc* BLEP residuals; the dotted line represents the harmonic envelope of a perfect sawtooth waveform; note the location of the −60dB level

humps just before and just after the discontinuity, and a flat "perfect" waveform between. In the frequency domain, we observe the same issue with the lack of harmonic envelope preservation. It appears that BLEP and Lanczos may be related. In 2016, Singh and Pirkle (2016) proved that BLEP correction is mathematically identical to Lanczos sigma correction. All along, we've really been trying to convert a trivial waveform into an equivalent, band-limited, Lanczos sigma corrected waveform. The fundamental difference is that, for BLEP, the correction is on-the-fly as the waveform is being rendered, whereas Lanczos applies to pre-calculated wavetables.

10.6 Other VA Algorithms

A drawback with BLEP is that technically, it will only adequately correct a waveform with a *C1* continuity, which, for synthesizers, is generally sawtooth (ramp) and square wave, but even in non-C1 cases, you can still obtain a massive reduction in aliasing in other trivial oscillators using *PolyBLEP* alone. There are more approaches to synthesizing the traditional waveforms without Fourier wavetables including Lane's original distorted sinewaves and Välimäki's Differentiated Parabolic Waveform (DPW) algorithm.

10.6.1 Differentiated Parabolic Waveform (DPW) Oscillators

Välimäki (2005) observed that differentiating a parabolic waveform produced an output that was similar in shape to a trivial sawtooth but with greatly reduced aliasing. Squaring a ramp or sawtooth waveform generates a parabolic waveform. Differentiating it produces the parabolic waveform. The DPW oscillator algorithms synthesize the sawtooth and triangle waveforms but tend to produce more aliasing than BLEP, and I do not use them for high-quality synthesis. One advantage is that the triangle wave is available since the algorithm does not depend on continuity type.

10.7 BLEP Square Wave

Since the square wave has a *C1* continuity, we are also able to use BLEP to correct its edges and produce alias suppressed waveforms. The only thing you need to keep track of is the direction of the discontinuity (rising or falling). The fundamental issue that makes this difficult occurs when you use an extreme pulse width setting. If the frequency of oscillation rises high enough, the edges come within a sample interval of one another. At a 1% pulse width, this occurs at about 2 kHz for a 44.1 kHz sample rate. This would require using BLEP at least twice since we have overlapping BLEP transition regions. For BLEP with a wider transition region, it becomes tricky since there are more points to overlap. These may need to be corrected more than twice.

But there is another way to generate square waves using a synth programming trick that was used on older digital synths that did not offer a PWM square wave oscillator. The sum-of-saws method uses two sawtooth waveforms to synthesize a square wave with an adjustable duty cycle. The idea, shown in Figure 10.9(a), consists of summing two sawtooth waveforms together where one waveform has been inverted and phase shifted. Because the waveforms consist of straight lines (ramps), they will sum to 1.0 or −1.0, depending on the crossover point between them, as shown in Figure 10.9(b). And, with the modulo counter approach, shifting the phase means simply adding an offset between 0.0 and 1.0 to the existing value, then wrapping as needed.

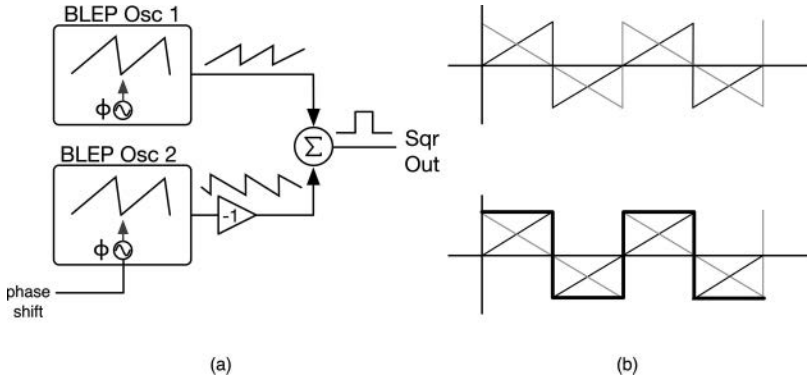


Figure 10.9 (a) Two sawtooth waveforms are combined to generate a square wave (b) shows how the ramps add or subtract to produce values of −1.0 or +1.0

10.8 *VAOscillator* and Core

The *VAOscillator* is shown in block diagram form in Figure 10.10(a), while Figure 10.10(b) shows the SynthLab diagram and the one and only *VAOCore*. The custom module strings are the three output waveform names, and the mod knobs are chosen for mixing the waveforms and applying pulse-width for the square wave output. Unlike the other oscillators, the *VAOscillator* and core are designed to render two waveforms at once, saw and square; then, the user may mix them in any ratio with the *WaveMix* mod knob. This is actually borrowed from the Oberheimm SEM®, which allowed mixing of square and sawtooth waveforms only.

Table 10.2 lists the GUI parameter structure and core description. Notice that the VA oscillator and core are the simplest of the oscillator objects; their five operational phases and functions are the simplest of those in SynthLab.

10.8.1 VA Oscillator Controls, Modulations, and Features

Table 10.3 lists the module controls common to all cores and the GUI calculation section or equations. The coarse tuning and fine-tuning operate in semitones and cents, respectively, and are part of the pitch modulation update calculation, while the pan and output controls are common to almost all other oscillators and allow blending of the quad oscillator render block.

10.9 VA Core Programming Notes

With the C++ files open, compare the programming notes with the code that you see, starting with the class description of the core. As with the wavetable oscillator, the VA core uses the same time-base, implemented with a *SynthClock* modulo counter that is reset and advanced with each note-event. Sections 10.9.1 through 10.9.5 summarize the five operational phases, plus constructor for the wavetable cores. Make sure to examine the code while digesting their operational phase details. The VA core is blissfully simple because it is purely algorithmic and does not require tables, files, databases, or other external components for rendering its output. This makes it a very lean object and ideal for use in mobile devices.

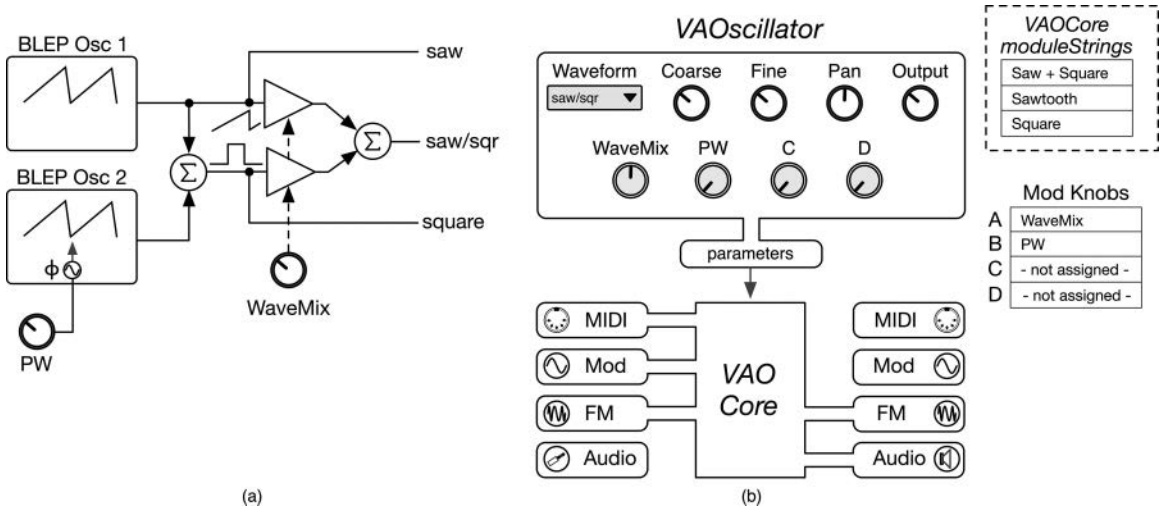


Figure 10.10 The *VAOscillator* (a) internal block diagram and (b) SynthLab diagram; there is only one core, which is also available for download as a SynthLab-DM project

Table 10.2 *VAOscillator* custom parameter structure and cores: (*) downloadable SynthLab-DM core project

Parameter Structure	Description
VAOscParameters	Includes the pulse-width parameter that is exclusive to the VA oscillator
Example Core	**Description & Special Features**
*VAOCore**	Simple VA oscillator core that demonstrates the sum-of-saws method for generating square waves and uses the Blackaman-Harris windowed BLEP table. This is the only oscillator that features pulse-width modulation, which is used as its unique osc mod.

Table 10.3 *VAOscillator* module controls

Module Control	**Description**	**GUI Calculation (Section)**
Coarse	Coarse tuning in semitones	Pitch modulation (6.7.1)
Fine	Fine-tuning in cents	Pitch modulation (6.7.1)
Pan	Left/Right pan	Constant power panning (6.2.2)
Output	Output level in dB	dB to raw: $10^{(dB/20)}$

10.9.1 Construction Phase

- The module strings are the waveform names, and their index values map to the GUI control selection values; the strings will be used to query the database for the selected waveform
- The mod knobs implement the wave mixing and pulse width

10.9.2 Reset & Note On Phases

The only detail here involves the starting phase for the sawtooth versus square wave, which is also repeated in the note-on handler. The SynthLab-VA project implements four of these oscillators, and the waveform starting-phase is important if the user mixes different waveforms. To phase align the saw and square waveforms, you need to offset the sawtooth by half a period so the first output sample is 0.0 in the center of the ramp, which is accomplished by offsetting the *SynthClock's* starting point.

```
if (parameters->waveform == VAWaveform::kSawtooth)
     oscClock.reset(0.5); // --- 0.5 for saw
else
     oscClock.reset(0.0);
```

10.9.3 Note-On Phase

The note-event begins with the note-on handler that is both similar and quite simple across the cores:

- Store MIDI pitch (except drum and SFX cores)
- Reset the *SynthClock* modulo counter member to either 0.0 (all cores) or the starting phase point (classic, morphing, and Fourier cores); notice how the mod knob C value is used to calculate the phase offset for the cores that support it
- NOTE: there is nothing to do in the note-off phase

10.9.4 Update Phase

The VA core update phase is simpler than the rest as there are no table or sample selections, just a simple update of the oscillator time-base that is identical to the other oscillators.

1 Calculate the pitch modulation value based on GUI controls and pitch modulation sources (Section 6.8.1)
2 Set the oscillator time-base with the new frequency
3 Calculate the output gain and pan
4 Store the pulse-width for the render phase

What is interesting here is the pulse width control since the *VAOscillator* is the only SynthLab oscillator that is capable of PWM. The core stores a *pulseWidth* variable for the render operation and recalculates it during the update phase, starting with getting the value from the mod knob – the percent is converted to a fraction.

```
// --- pulse width from ModKnob
//   note the way this works 0.0 -> 50% PW 1.0 -> 95% PW
pulseWidth = getModKnobValueLinear(
                    parameters->modKnobValue[MOD_KNOB_B], 0.5, 0.95);
```

Next, the bipolar modulation value is acquired from the *kUniqueMod* input value.

```
// --- this value is modulated and then bound
double pwModulator =
            processInfo.modulationInputs->getModValue(kUniqueMod);
```

To use this value effectively, call the bipolar modulation function and cut the return value in half so that bipolar modulation will revolve around the center point that the user chooses with the mod knob. Lastly, the value is bounded to 5% and 95% (feel free to modify these limits).

```
pulseWidth += 0.5 * doBipolarModulation(pwModulator, 0.5, 0.95);

// --- bound it
boundValue(pulseWidth, 0.5, 0.95);
```

10.9.5 Render Phase

The VA core renders both the sawtooth and square wave at the same time, using the same function, then outputs the desired waveform or blends with the wave mix control for the saw/square combination waveform. The *VAOCore* uses two helper functions for rendering, each producing one sample per sample interval. The *renderSquareSample* uses the sum of saws method, calling the *renderSawtoothSample* twice and returning the sawtooth value via a pass by reference output parameter.

```
double renderSawtoothSample(SynthClock& clock);
double renderSquareSample(SynthClock& clock, double pulseWidth_Pct,
                          double& sawtoothSample);
```

The *renderSawtoothSample* function consists of three parts. First, check the oscillator frequency, and determine the maximum points per side to adjust; see Table 10.1.

```
double pointsPerSide = 0;
if (clock.frequency_Hz <= sampleRate / 8.0) // Fs/8 = Nyquist/4
     pointsPerSide = 4;
else if (clock.frequency_Hz <= sampleRate / 4.0) // Fs/4 = Nyquist/2
     pointsPerSide = 2;
else // Nyquist
     pointsPerSide = 1;
```

Next, create the trivial oscillator output, and call the BLEP function, which returns the BLEP correction factor.

```
double sawOut = bipolar(clock.mcounter); // create triv saw
double blepCorrection = 0.0;

// --- get the correction factor
blepCorrection = doBLEP_N(4096,                /* BLEP table length */
                          clock.mcounter,      /* mod count */
                          fabs(clock.phaseInc),
                          1.0,                 /* edge height = 1.0 */
                          false,               /* falling edge */
```

```
                                pointsPerSide,    /* N points per side */
                                false);           /* no interpolation */
```

Lastly, add the correction factor to the trivial sawtooth value and advance the clock time-base.

```
// --- add the correction factor
sawOut += blepCorrection;

// --- setup for next sample
clock.advanceWrapClock();
```

The *renderSquareSample* function calls the sawtooth render function twice. Notice the use of the *SynthClock::saveState* and *restoreState* functions, which are used to provide the temporary phase offset needed for the current *pulseWidth* value.

```
// --- sum-of-saws method
// --- set first sawtooth output
sawtoothSample = renderSawtoothSample(clock);

// --- save clock state
clock.saveState();

// --- phase shift on second oscillator
clock.addPhaseOffset(pulseWidth);

// --- generate 2nd saw
double saw2 = renderSawtoothSample(clock);

// --- subtract = 180 out of phase
double squareOut = 0.5*sawtoothSample - 0.5*saw2;

// --- restore original clock state
clock.restoreState();

return squareOut;
```

The DC correction function uses my formula for adjusting the DC offset based on the pulse width; as the pulse width becomes wider at the top, the DC offset needs to shift downward to make up for the added area under the curve. This may also be accomplished with a high-pass filter.

```
// --- apply DC correction
double dcCorrection = 1.0 / pulseWidth;

// --- modfiy for less than 50%
if (pulseWidth < 0.5)
     dcCorrection = 1.0 / (1.0 - pulseWidth);
```

```
// --- apply correction
squareOut *= dcCorrection;
```

The *render* function first acquires the output buffers, like the other oscillators, then sets up the block processing loop:

```
float* leftOutBuffer = processInfo.outputBuffers[LEFT_CHANNEL];
float* rightOutBuffer = processInfo.outputBuffers[RIGHT_CHANNEL];

// --- render square and saw at same time
for (uint32_t i = 0; i < processInfo.samplesToProcess; i++)
```

The loop code calls the rendering function that produces both outputs, which are output or linearly blended according to the user's waveform selection.

```
double oscOutput = 0.0;
// --- render both saw and square always, choose output with param
double sawOutput = 0.0;
double sqrOutput = renderSquareSample(oscClock, sawOutput);

if (parameters->waveform == VAWaveform::kSawAndSquare)
{
     // --- blend with mod knob
     double mix = parameters->modKnobValue[VAO_WAVE_MIX];

     oscOutput = sawOutput*mix + sqrOutput*(1.0 - mix);
}
else if (parameters->waveform == VAWaveform::kSawtooth)
     oscOutput = sawOutput;
else if (parameters->waveform == VAWaveform::kSquare)
     oscOutput = sqrOutput;

// --- scale
oscOutput *= outputAmplitude;

// --- write to output buffers
leftOutBuffer[i] = oscOutput * panLeftGain;
rightOutBuffer[i] = oscOutput * panRightGain;
```

10.10 Exercises

10.10.1 SynthLab-VA$_{DM}$: Mod Knob C to Shape Control

You can create a simple oscillator shape control with one of the spare mod knobs using the phase distortion shaping technique (Section 6.9) utilized in the wavetable cores. Add a waveform shape control that applies phase distortion to generate different sounds. Apply the phase distortion to the synth clock *mcounter* value prior to use. How does this affect the BLEP correction's ability to suppress aliasing?

10.10.2 SynthLab-VA$_{DM}$: Shape Modulation

The VA oscillator's unique modulation capability produces PWM; however, this only applies to the square waveform. Modify the core code to implement shape modulation when the user has selected the sawtooth-only waveform.

10.10.3 Advanced Module: Implement Välimäki's DPW

Download the Välimäki's DPW algorithms for generating sawtooth and triangle waves. Implement the algorithm as your own dynamic module core.

Bibliography

arizona.edu. 2020. "Continuity." http://www-isl.ece.arizona.edu/ACIS-docs/HTM/DATA/ACIS/FCG/03MATH/0003.HTM, Accessed October 14, 2020

Brandt, Eli. 2001. "Hard Sync without Aliasing." Proceedings of the International Computer Music Conference, Havana, Cuba.

Dattorro, Jon. 2003. "Effect Design Part 3 Oscillators: Sinusoidal and Pseudonoise." *Journal of the Audio Engineering Society*, vol. 50, no. 3. pp. 115–146.

Kleimola, Jari &Välimäki, Vesa. 2012. "Reducing Aliasing from Synthetic Audio Signals using Polynomial Transition Regions." *IEEE Signal Processing Letters*, vol. 19, no. 2.

Lane, J., et al. 1997. "Modeling Analog Synthesis with DSPs." *Computer Music Journal*, vol. 21, no. 4, pp. 23–41.

Leary, Andrew & Bright, Charles. 2009. "Bandlimited Digital Synthesis of Analog Waveforms." United States Patent 7,589,272.

Moore, Richard. 1990. *Elements of Computer Music*. Eaglewood Cliffs: Prentice-Hall.

Moorer, James. 1976. "The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulae." *Journal of the Audio Engineering Society*, vol. 24, no. 9, pp. 717–727.

Nam, Juhan & Välimäki, Vesa, Abel, Jonathan & Smith, Julius O. 2010. "Efficient Antialiasing Oscillator Algorithms using Low-Order Fractional Delay Filters." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4. pp. 733–785.

Otalvara, Francisco, Behura, Samarth & Pirkle, Will. 2016. "Perceptually Alias-Free Waveform Generation using the Bandlimited Step Method and Genetic Algorithm." Presented at the 141th Audio Engineering Society Convention, New York.

Pekonen, Jussi, Lazzarini, Victor, Timoney, Joseph, Kleimola, Jari & Välimäki, Vesa. 2011. "Discrete-Time Modeling of the Moog Sawtooth Waveform." EURADISP *Journal on Advances in Signal Processing*, Article ID 10.1155/2011/785103.

Singh, Akhil & Pirkle, Will. 2016. "The Relationship between the Bandlimited Step Method (BLEP), Gibbs Phenomenon, and Lanczos Sigma Correction." Presented at the 141th Audio Engineering Society Convention, New York.

Stilson, Tim & Smith, Julius O. 1996. "Alias-Free Digital Synthesis of Classic Analog Waveforms." Proceedings of the 1996 International Computer Music Conference.

Välimäki, Vesa. 2005. "Discrete-Time Synthesis of the Sawtooth Waveform with Reduced Aliasing." *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 214–217.

Välimäki, Vesa & Huovilainen, Antti. 2006. "Oscillator and Filter Algorithms for Virtual Analog Synthesis." *Computer Music Journal*, vol. 30, no. 2, pp. 19–31, Cambridge: MIT Press.

Välimäki, Vesa, Nam, Juhan, Abel, Jonathan & Smith, Julius O. 2010. "Alias-Suppressed Oscillators Based on Differential Polynomial Waveforms." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 786–798

11 PCM Sample Playback Oscillators

Early sampler instruments were capable of recording sounds – single musical tones (note-events) from acoustic instruments, short audio clips, or sound effects – and playing them back with the ability to speed up or slow down the playback so as to generate proper musical pitches when desired. This idea led to sample playback synthesizers, which played banks of pre-recorded musical instrument sounds or sound effects. Due to confusion regarding the terms "sample" and "wave" and "wave file," I use the term "PCM sample" to specifically refer to a recorded event that is not a wavetable. PCM stands for Pulse Code Modulation, the technique of encoding voltage levels as numbers. In contrast to the early samplers, the newer sample playback synths used sets of samples called "multi-samples," made of recordings of note-events at different pitches across the range of an acoustic instrument. These PCM samples were mapped to different keys on the keyboard and followed a paradigm similar to the wavetable synths, in which a single PCM sample is used for a range of playback notes via interpolation. Typically, intervals of a minor third or octave were used.

11.1 PCM Sample Playback Modes

Figure 11.1 shows the four commonly used playback modes: one-shot; loop; transient plus loop; and transient, loop, and release. Pitch-less PCM samples are designed to playback at the speed they were recorded, while the pitched versions are designed to map to the keyboard with proper musical note playback. As in the wavetable, when PCM sample data is read out using an increment value greater than one, aliasing may occur. Early samplers had no way of generating alias-free multi-samples from a single recording, so the effect of aliasing is sometimes associated with them and is sometimes seen not as a problem but rather as a kind of personality of the device.

11.1.1 One-shot Mode

In one-shot mode, shown in Figure 11.1(a), the PCM data is played from beginning to end exactly once, and then playback stops, even if the key is still being held. Latching one-shot allows the user to trigger the playback with a single press and release of a key or button. One-shot PCM samples are usually used for drum hits or sound effects and may or may not be pitched.

11.1.2 Loop Mode

In looping mode, shown in Figure 11.1(b), the entire PCM sample is looped from beginning to end and then repeated. These loops are often pitch-less, implementing drum beats or

background pads. When using loops of musical material, playback at speeds other than normal often sounds artificial.

11.1.3 Transient + Loop Mode

Figure 11.1(c) depicts the transient plus loop operation and is the most common mode for PCM sample playback synths. In this mode, the PCM sample recording contains the initial transient sound plus a section of the waveform containing the steady state portion that is looped, while the key is held. When the key is released, the sustain loop continues, and an artificial release envelope is applied via the DCA and its EG. This is the most common kind of PCM sample playback as it has been used in numerous synths, like the Korg M1 and Kurzweil K2000/2500. The Legacy PCM core object implements 16 sets of these multi-samples to generate its 16 different waveforms.

11.1.4 Transient + Loop then Release Mode

Figure 11.1(d) shows the transient plus loop then release mode in which the transient and release portions of the waveform are preserved, with the transient loop section positioned between them. This allows a very accurate synthesis of the original instrument waveform that includes both the attack and the release information. The release portion may also be further shaped with the DCA and EG.

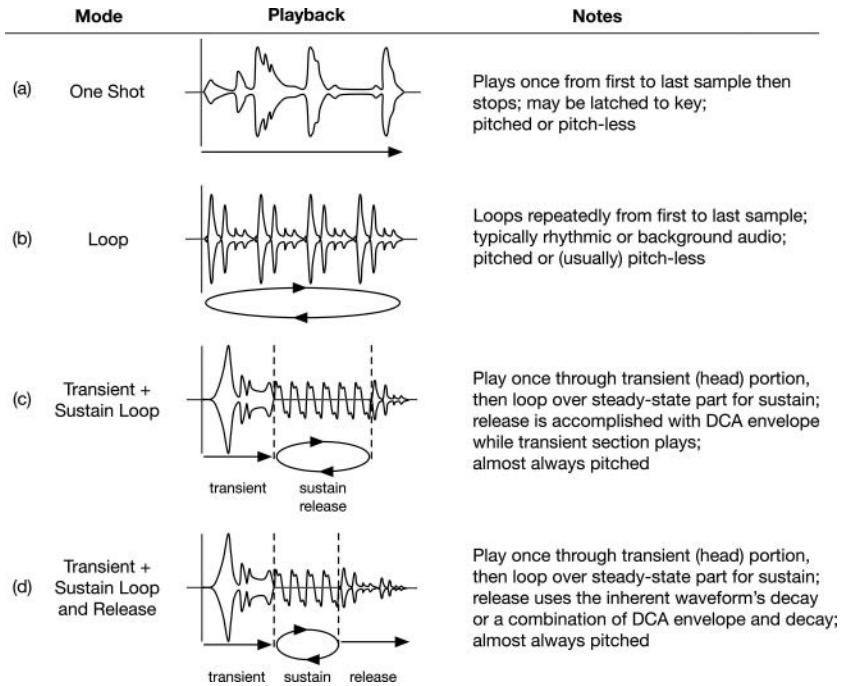


Figure 11.1 The four common modes of PCM sample playback include (a) one-shot; (b) loop; (c) transient plus loop; and (d) transient, loop, and release

11.2 PCM Sample Storage

PCM samples are usually much longer than the few hundred or thousand sample points of a wavetable, and this may be problematic. Unlike the wavetables that can be compiled into the synth, PCM samples are almost always packaged in a separate file or set of files; this necessitates the opening and parsing of the files in order to populate buffers for playback. One simple way to package the PCM samples is in a "wave" file (with the extension *.wav*). This is especially attractive since the wave file format includes the ability to store one or more sample loop points along with the audio data. However, it also exposes the raw material to the user, who may then copy or move the data into another device. Typically, only one PCM sample is stored in a wave file, so this generates numerous files to maintain. For this text, the wave file format works well and will allow you endless experimentation with PCM samples, which you can buy, get for free, or even generate yourself.

11.2.1 Packaging Samples in Wave (.wav) Files

The wave file format was conceived to store more information than just the audio data and relevant information about it, such as channel count, number of samples, and samples per data word. The wave file format consists of chunks of information named riff-chunks, along with the raw audio data encoded as bytes. Audio samples may be encoded as 8-, 16-, or 24-bit integers, or 32- or 64-bit floating point values.

In particular, the wave file format also includes the option to store the

1 Sample period ($T = 1/f_s$)
2 Loop count (may have more than one loop, though we use single-loop files exclusively)
3 Starting and ending sample indexes for each loop in the file
4 SMPTE time code information
5 MIDI Unity Note
6 MIDI Pitch Fraction

11.2.1.1 MIDI Unity Note Information

In addition to being able to store multiple loop points, the wave file format also includes the MIDI unity note and MIDI pitch fraction, which are important for PCM samples. The MIDI unity note represents the MIDI note number of the pitch that will occur if the PCM samples are read out with a phase increment of 1.0 – in other words, the note that was originally recorded for the sample. If the pitch of the note is not a standard MIDI pitch (e.g. instruments that do not use Western intonation, such as sitar), then the MIDI pitch fraction may contain the additional pitch offset information.

In the past, it was common for professional sample and loop generating companies to encode the MIDI unity note information in the wave file, making it simple to decode and calculate the phase increment lookup value. More recently, this paradigm has been mostly ignored, with the note number and name encoded into the name of the wave file, e.g. *flute_A#4.wav*. Clearly, encoding the unity note like this is problematic as there is no standard whatsoever. However, searching for the note string within the filename is not unreasonable, and the Mellotron ® PCM samples in the Mellotron core are encoded this way.

11.2.2 Calculating the PCM Phase Increment

For pitch-less samples and loops, the phase increment is simply set to 1.0, and the loop or sample will play at the same rate and pitch, no matter which key is used to trigger the event. To mimic the early samplers, the pitch can be adjusted for interesting effects, but for musical pitches to be accurate, the original pitch of the sample must be known either from the embedded unity note number or from a lookup table once the filename has been decoded. The phase increment value (called the playback ratio) may be calculated in two different manners. Equation (11.1) uses the MIDI note number:

$$\begin{aligned} &phaseInc = 2^{\frac{M-U}{12}} \\ &M = \text{MIDI note number of note being played} \\ &U = \text{MIDI unity note number} \end{aligned} \tag{11.1}$$

To avoid the power-of-two operation, Equation (11.2) can be derived easily, using the equivalent length of a wavetable with the same MIDI unity note pitch.

$$\begin{aligned} &phaseInc = \frac{f_o}{f_U} \\ &f_o = \text{frequency of note being played} \\ &f_U = \text{frequency of unity note (original sample pitch)} \end{aligned} \tag{11.2}$$

11.2.3 Multi-Samples and the Keyboard

Multi-samples are generated for multiple pitches across the keyboard and are distributed across the range of notes in a similar manner to the wavetables in Figure 9.4. These include but are not limited to:

1 One PCM sample per octave
2 One PCM sample per minor third interval (four samples per octave)
3 One PCM sample per note

For the Mellotron samples used in the *MellotronCore*, the original keyboard had a limited range of 35 keys from G2 to F5, so the samples in that set are likewise limited in range.

11.3 Loading PCM Samples

The early PCM sample-based synths usually incorporated ROM (read only memory) chips, with samples that were permanently burned into them. These devices, and the method of encoding the audio information in them, were proprietary and part of an embedded audio system specific to each manufacturer. To offer more PCM samples for the user after the unit was purchased, some manufacturers made expansion cards with still more ROM, which could be inserted into dedicated hardware ports on the synth. Later, PCM samples were distributed on CD-ROM and loaded via SCSI data ports onto synths that had this capability. The ability to load new sample sets is attractive for end users, while disc or CD-ROM-based storage is attractive for the manufacturers as they have control over distribution. However there is one sticky point that needs to be addressed, no

matter what sample delivery system is used: when should this new, non-factory data be read and loaded into ROM? There are two basic options:

1 Pre-load the data into RAM (random access memory) when the synth starts up or when a new expansion card, hard disc, or CD-ROM is attached
2 Load the PCM data from hard disc or CD-ROM on-the-fly as the user depresses each key

If the PCM sample data is loaded all at once and placed into RAM, the synth engine and synth voices will have instant access to the data via simple pointers or table index values, and the PCM oscillators will begin rendering their outputs immediately. However, if the synth loads significant amounts of data, this forces the user to wait for the data to load. In addition, this is not memory-efficient since many samples may be loaded into memory but never actually played. In the all-at-once paradigm, the user must wait for the data to load, but the programming code is very simple for both reading the data into RAM and rendering the audio output.

On the other hand, if the data is loaded from a disc as the user plays the notes, you need to create a file input stream from the disc and begin accessing the data, which is far too slow for real-time operation. But processing threads can be used to allow each note-event to optimize the parallelism inherent in polyphonic playing, and data can be read in chunks. In this on-demand system, multiple threads can be used for more efficiency, but the code will be much more complex because it will need to include: setting up and waiting for disc access, creating a worker thread to fetch data from the disc, and synchronizing that thread so it does not interfere with the synth's native processing threads.

In the on-demand paradigm, the time it takes to access the disc and begin fetching data to render is prohibitive and will cause a noticeable tactile and audible delay. This brings up a third, hybrid option that combines both ideas. At load-time, you only read out the first chunk (say, 4096 samples) of each PCM sample, and you save that in RAM. This significantly reduces the load time and memory required. Then, when the user plays a note, you begin reading from that data in RAM for the beginning of the note-event and launch the thread that performs disc accesses for the rest of the data. From that point on, the code involves synchronizing the disc accesses and fetching chunks of data as needed for the rendering operation. Clearly, this is a significant coding challenge and involves heuristic problems, such as switching PCM sample sets on-the-fly while notes are being held.

11.3.1 Data Brick Files

Another way to package the samples sets is in one giant contiguous data "brick" file that contains all of the sample information, loop points, etc. but is compacted into a sequential data file. This hides the data from the user and even with simple encryption techniques can make the data very difficult to steal. In the hybrid approach, the brick file might contain an array of just the starting portions of the waveforms. For educational purposes, I am delivering the data in wave files stuffed into folders. However, if you would like to experiment with using the brick files or reading samples on-demand, you can find tools to extract the wave samples and loop points at www.willpirkle.com/synth-tools.

11.3.2 Sample Slicing

In sample slicing, you start with a wave file that contains multiple musical events. Then, an algorithm is used that finds the transient edges of these musical events and creates slice points.

The wave file can then encode these multiple loop points or else be split into individual wave files, each containing a slice. These loops can then be mapped to the keyboard or touch-screen display, and the user plays the loops as if they were musical notes. The methods for finding the transient edges involves math that is interesting and deep. Usually, multiple methods are used, including finding time-based transients and locating energy or spectral bursts that indicate a transient edge. A free, open-source library of tools named *aubio* is available at https://github.com/aubio/aubio and includes a command line utility called *aubiocut* that detects transient edges in a wave file, then slices it into a set of smaller wave files. You have control over several factors used in the detection operation. The wave files are named according to the time location of the starting slice. The *WaveFolder* object in the next section can decode the *aubio* filenames to create sets of pitch-less loops that can be mapped to the keyboard and is included as the *WaveSliceCore*.

11.3.3 PCMSample and WaveFolder C++ Objects

To facilitate PCM sample loading and rendering, I've designed two helper objects that do most of the work. The *PCMSample* object is used to open a wave file and extract its information, and works for both Windows and MacOS. The audio data is loaded and stored in an array of floating point values. The *PCMSample* object also parses and saves the loop start and end points, and the MIDI unity note number. The object works with mono or stereo uncompressed 8-, 16-, or 24-bit integer wave files as well as 32- and 64-bit floating-point files – in all cases, the final data is available in 32-bit floating-point form only, which is also the synth's preferred audio data type. Each wave file will have its own associated *PCMSample* object.

In the SynthLab-PCM synth, each set of multi-samples that generates one waveform is stored in its own directory, as shown in Figure 11.2(a). The name of the directory is the name of the selected sample in SynthLab (though that is easy to change). At startup, the PCM samples are loaded from the wave files and stored in *PCMSample* objects, organized in an array of 128 slots, each corresponding to the MIDI note number for that sample's pitch. Each SynthLab PCM core object implements an instrument with up to 16 patches. You can see that for the Mellotron, there is one wave file per note for all 35 keys on the instrument, from G2 through F5, as shown in Figure 11.3(b). Each file is extracted into a dynamically created *PCMSample* object whose pointer is stored in the array. Since the sample set is incomplete for all 128 MIDI notes, the bottom G2 data is used for all notes below it, while the F5 data is used for all keys above it. Another option would be to simply not render the notes outside the original range, but that may confuse the user. For the legacy core samples, there are missing samples at the top and bottom of the range as well as within the existing samples that are placed on minor-third boundaries from A1 through C4, as shown in Figure 11.3(c). All of the legacy samples are designed for bass sounds – thus the limited range – and were taken from free samples appearing in *Future Music* and *Computer Music* magazines. In addition to the member functions in Table 11.1, the *PCMSample* object also stores information as member variables, including:

numChannels: number of audio channels; SynthLab supports mono or stereo files
loopCount: number of loops in the file; SynthLab only parses the first loop
loopStartIndex: starting index of loop (absolute index)
loopEndIndex: ending index of loop (absolute index)
unityMIDINote: MIDI note number for unity note

Table 11.1 The *PCMSample* object's functions

PCMSample Function	*Argument*	*Description*
loadPCMSample	*filePath*	Opens, decodes, and extracts the audio data from a wave file located at the *filePath* argument; also extracts loop and unity note information
getSampleBuffer	None	Returns a *const float* (read-only) pointer to the audio data array for rendering
setPitchless	*_pitchlessSample*	Sets the pitch-less sample flag on the object, forces read phase increment to 1.0 always

The *WaveFolder* object extracts all of the PCM samples from the wave files present in a patch folder and builds the array of pointers to the corresponding *PCMSample* objects. The MIDI unity note number may be encoded in the wave file or may exist as character strings in the file names, as shown in Figure 11.2(a). In the latter case, the *WaveFolder* object will attempt to intelligently extract the note number from the file name (names with both "#" and "b" are supported) and will embed the corresponding MIDI note number in the *PCMSample* object. Once all the files have been parsed, the *WaveFolder* object will perform a second pass over the array of pointers, copying each pointer to fill the slots that may be missing due to incomplete sample sets. This simplifies the note rendering operation, giving us access to each sample using the MIDI note number as the array look-up index. The *WaveFolder* object has only two main functions: the constructor, which stores the outer directory, and the sample folder (e.g. M300 Brass in Figure 11.2). Once constructed, the *IPCMSampleSource* object calls the *parseFolder* method to extract and set up the 128 *PCMSample* pointers in Figure 11.2. The prototype and variable descriptions are below; notice the flag that indicates a folder of *aubiocut* wave slices from Section 11.3.2.

```
parseFolder(PCMSample** sampleSet, /* pointer to array of pointers */
            bool pitchlessLoops,  /* pitchless loop flag */
            bool aubioSlices)   /* aubio slices flag */
```

The details of the folder parsing are in the function and perform the operations in Figure 11.2. For *aubiocut* pitch-less loops, the *WaveFolder* is set up to map these loop slices to the C-major scale notes (all white keys), starting with middle-C (MIDI note 60) and mapping upwards. You may easily modify this code for your own mappings.

11.4 PCM Sources and Database

The *IPCMSampleSource* interface is used to create a C++ class that provides access to a set of PCM multi-samples arranged in an array. You can think of a "sample source" as all of the information needed to synthesize one PCM multi-sample across the range of MIDI notes. This interface allows you to implement your PCM sample sources however you wish and is one of the central design patterns in SynthLab. As long as your sample source object implements the interface functions, it will work seamlessly with the PCM core objects. The *SynthLabPCMSource* object exposes this *IPCMSampleSource*, and you may use it as an example to design your own. Figure 11.3(a) shows the arrangement of *IPCMSampleSource* objects in the database. This is identical to the *WavetableDatabase* construction and follows the same paradigm.

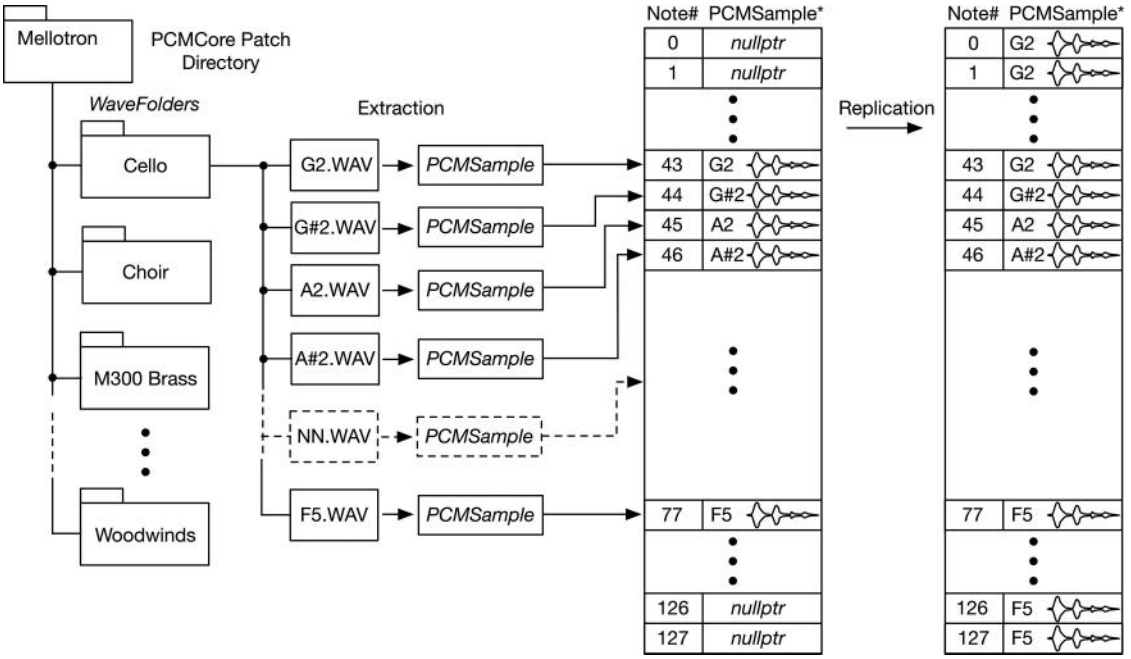


Figure 11.2 The Mellotron core directory contains subdirectories for each patch, which, in turn, contains a set of wave files; each of these is extracted into a *PCMSample* object and stored in an array of pointers in an *IPCMSampleSource* object

The *PCMSampleDatabase* is created in the *SynthEngine* and shared across all *SynthVoices* and their PCM oscillators. The database is a *std::map* that stores *IPCMSampleSource* pointers via unique key strings. Your PCM core object may query the database for sample sources, and it may create and add new PCM sample sources at construction time. The database uses unique name strings because, ultimately, the user will need to select the waveform from a list of unique names. SynthLab fills the GUI waveform selectors with these unique strings any time a new core object is loaded. Table 11.2 lists the *IPCMSampleSource* functions and their descriptions.

11.4.1 Reading\|Interpolating the PCM Sample

The *IPCMSampleSource* object implements the *readSample* function that does the work in reading and linearly interpolating the audio samples. This code is actually quite simple. The SynthLab sample playback oscillator cores are the only oscillators that do not use the *SynthClock* object for a time-base. Instead, the sample phase index value is calculated and stored along with a read index. The *IPCMSampleSource* performs the read and interpolate operation, then updates the read index and returns it to the core for storage via pass-by-reference:

```
PCMSampleOutput readSample(double& readIndex, double inc)
```

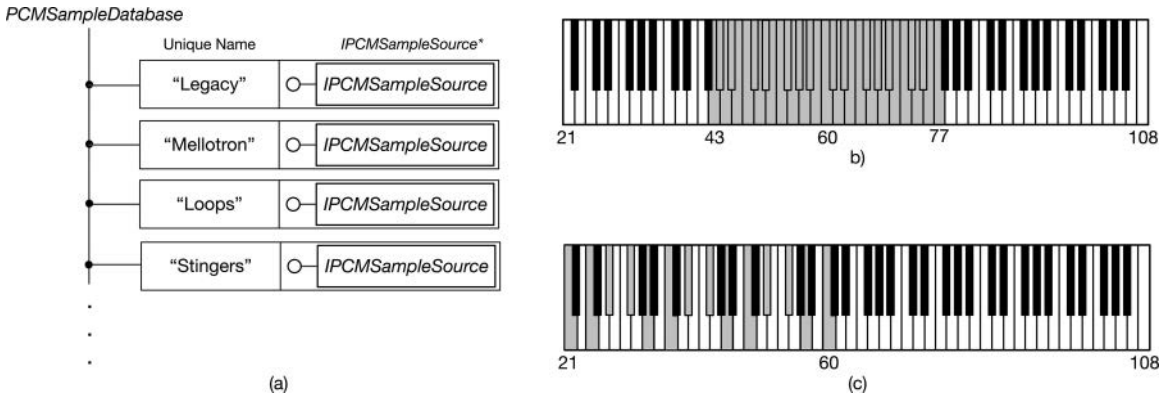


Figure 11.3 (a) The sample sources are arranged in a database and accessed via a unique name string; (b) the shaded keys show the included wave files for the Mellotron, sampled on every note from G2 through F5, while the (c) legacy files include samples on minor third boundaries from A1 through C4

Table 11.2 The *IPCMSampleSource* interface

IPCMSampleSource Function	*Argument*	*Description*
selectSample	*oscFrequency*	Selects a *PCMSample* object based on the oscillator frequency and the MIDI unity note frequency in Equation (11.2)
readSample	*readIndex* *inc*	Reads the PCM data using a read- *readIndex* and increment (*inc*) value, then returns the output data in a structure
setSampleLoopMode	*SampleLoopMode*	Sets the loop mode: *one-shot, loop,* or *sustain*
deleteSamples	None	Used during destruction

The code for finding the fractional sample location is identical to that of the wavetable oscillator cores, and the only real difference is the data source that provides the sample values. The PCM oscillator cores output into a very simple structure that contains the audio data (mono or stereo) and an active channel count.

```
struct PCMSampleOutput
{
      double audioOutput[STEREO_CHANNELS] = { 0.0, 0.0 };
      uint32_t numActiveChannels = 0;
};
```

For a single channel wave file, the read code is below; the variable called *selectedSample* is an *IPCMSampleSource** that is parsed when the user chooses a new PCM sample patch. This interface is used to access the data in the sample buffer via the *getSampleBuffer* function. The linear interpolation uses the fractional part (thus, the first arguments, 0 and 1) and the two adjacent samples from the buffer.

```
// --- split the fractional index into int.frac parts
double dIntPart = 0.0;
double fracPart = modf(readIndex, &dIntPart);
uint32_t nReadIndex = (uint32_t)dIntPart;

// --- mono file
if (selectedSample->numChannels == 1)
{
     // --- check for end of file (theoretically won't happen)
     int nReadIndexNext = nReadIndex + 1 >
          selectedSample->sampleCount - 1 ? 0 : nReadIndex + 1;

     // interpolate between the two
     output.audioOutput[LEFT_CHANNEL] =
               doLinearInterpolation(0, 1,
               selectedSample->getSampleBuffer()[nReadIndex],
               selectedSample->getSampleBuffer()[nReadIndexNext],
               fracPart);
     output.audioOutput[RIGHT_CHANNEL] =
                                        output.audioOutput[LEFT_CHANNEL];
     readIndex += inc;
}
```

11.4.2 Looping the PCM Sample

Looping the PCM sample is a bit more involved than the wavetable. First, the sample loop type is identified which sets the looping logic. For one-shot samples, once the end is reached, the *readIndex* variable is set to −1, which indicates that the loop is finished, and the output will be silent after that. For ordinary loops from start to end, once the *readIndex* crosses the end point, it is wrapped back to the head of the buffer. For transient plus loop, when the loop end point is crossed, you wrap back to the loop start point. That logic is fairly simple, and you may find the details in the *SynthLabPCMSource::readSample* method.

11.5 *PCMOscillator* and Cores

The *PCMOscillator* and core objects are shown in block form in Figure 11.4. There are three cores for you to use as examples or for modification. The legacy core contains the original samples from the first edition of the book, with transient plus loop and MIDI notes embedded in the wave files. The Mellotron core's samples are free and available at https://sonicbloom.net/en/free-sb-mellotron-samples/; note that they are set up as long (five- to eight-second) one-shot waveforms, and embed the note name and number in the filename. The wave slice core uses multiple sources for loops and *aubiocut* to slice them up – see the readme.txt file in each folder for information about the free loop source URL and notes about slicing.

Table 11.3 lists the GUI parameter structure and core descriptions.

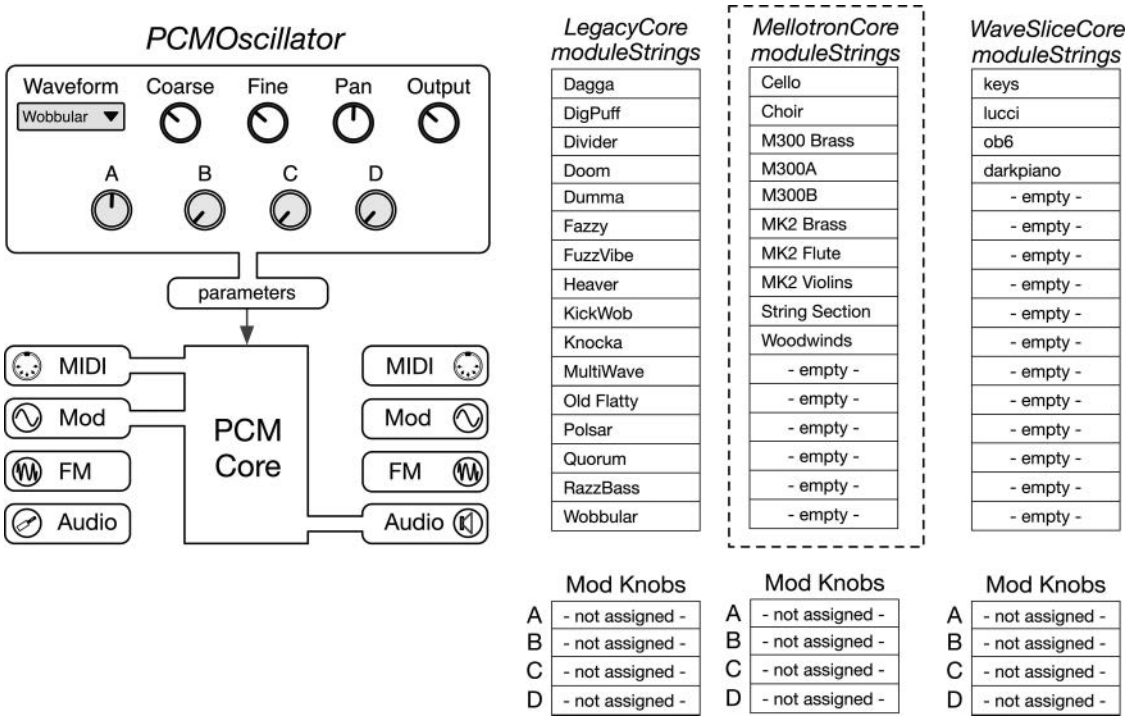


Figure 11.4 The *PCMOscillator* block diagram and module strings; none of the mod knobs are assigned in these cores

Table 11.3 PCMOscillator custom parameter structure and cores: (*) downloadable SynthLab-DM core project

Parameter Structure	Description
PCMOscParameters	Used for all cores, includes all oscillator parameters
Example Core	**Description**
LegacyCore	16 samples on minor third intervals covering the lower half of the keyboard, mainly for bass sounds
MellotronCore	Long Mellotron samples, one per note across the original keyboard range
*WaveSliceCore**	Demonstrates mapping wave-slices to individual keys to create a slicer-rompler style instrument; *aubiocut* was used to create the individual wave slice *.wav* files

11.5.1 PCM Oscillator Controls, Modulations, and Features

Table 11.4 lists the module controls common to all cores and the GUI calculation section or equations. The coarse tuning and fine-tuning operate in semitones and cents, respectively, and are part of the pitch modulation update calculation, while the pan and output controls are common

Table 11.4 Module controls shared across cores

Module Control	Description	GUI Calculation (Section)
Coarse	Coarse tuning in semitones	Pitch modulation (6.7.1)
Fine	Fine-tuning in cents	Pitch modulation (6.7.1)
Pan	Left/Right pan	Constant power panning (6.2.2)
Output	Output level in dB	dB to raw: $10^{(dB/20)}$

to almost all other oscillators and allow blending of the quad oscillator render block. The PCM oscillator cores do not have any mod knob mappings, so you are free to experiment. The PCM core unique modulation is pan modulation (Section 6.2.2).

11.6 PCM Core Programming Notes

With the C++ files open, compare the programming notes with the code you see, starting with the class descriptions of each core. The PCM cores do not use the *SynthClock* as their time-base; instead, they keep track of their own *phaseInc* value, which helps calculate the current *readIndex* variable used to read from the *.wav* file data. Each core obtains an *IPCMSampleSource* interface pointer that is used to query and load samples into the database in a very similar manner as the *IWavetableSource* objects.

Sections 11.6.1 through 11.6.5 summarize the five operational phases plus constructor for the wavetable cores. Make sure to examine the code while digesting their operational phase details.

11.6.1 Construction Phase

All cores:

- The module strings are the PCM sample file names, and their index values map to the GUI control selection values; the strings will be used to query the database for the selected waveform
- The mod knobs are not assigned
- The cores demonstrate multiple was to deal with samples: the legacy files use transient plus loop, the Mellotron samples are long one-shot files, and the wave slices play in their original pitches and are mapped to individual keys for sample playback

11.6.2 Reset Phase

The *reset* function queries the PCM sample database and adds the sources as needed. The *IPCMSampleSource* object handles parsing the PCM data from the wave files. For SynthLab, the samples are located within a special folder that will sit in the same folder as your plugin DLL. The DLL folder is acquired when the plugin loads and is available in the *processInfo.dllPath* variable. You only need to provide the outer containing folder name and the sub-directories with the instrument samples – these are your *coreData.moduleStrings[ ]* values that you set in the constructor. You may also set a simple hard-coded location for the samples (e.g. *C:\samples*). This code demonstrates the legacy core operation.

```
// --- initialize samples
std::string sampleFolder = concatStrings(processInfo.dllPath,
                           "\\SynthLabSamples\\Legacy\\");

for (uint32_t i = 0; i < MODULE_STRINGS; i++)
{
     std::string sampleFile =
          concatStrings(sampleFolder, coreData.moduleStrings[i]);

     checkAddSampleSet(sampleFile.c_str(),
                                  coreData.moduleStrings[i], processInfo);
}
```

11.6.3 Note-On Phase

The note-on event is easily serviced; you save the incoming MIDI pitch to use for update calculations and reset the *readIndex* variable.

```
// --- save pitch
midiPitch = processInfo.noteEvent.midiPitch;

// --- reset to new start phase
readIndex = 0.0;
```

11.6.4 Update Phase

The pitched core update phases follow the same three steps that you will find on the other pitched oscillators. The *update* function's main goal here is to have the correct PCM sample selected and its interface pointer stored for the render phase that will immediately follow.

1 Calculate the pitch modulation value based on GUI controls and pitch modulation sources
2 Select the PCM source based on the user's waveform selection string
3 Select the sample based on the final modulated pitch value

To get the sample source, you use the GUI control index value:

```
// --- get the wave string
const char* wave = coreData.moduleStrings[parameters->sampleIndex];

// --- select the source
selectedSampleSource =
                     processInfo.sampleDatabase->getSampleSource(wave);
```

With the PCM source selected, you can get the *phaseInc* value by calling the *selectSample* function, which also chooses the *PCMSample* object for rendering. It returns the *phaseInc* value to use in the next phase.

```
phaseInc = selectedSampleSource->selectSample(oscillatorFrequency);
```

11.6.5 Render Phase

The PCM rendering is efficient and simple. The sample source object will handle reading and interpolating the buffer. The render function first acquires the audio buffers, then runs a loop over the block, rendering both left and right channels using the *readSample* method, which also updates the *phaseInc* value:

```
float* leftOutBuffer = processInfo.outputBuffers[LEFT_CHANNEL];
float* rightOutBuffer = processInfo.outputBuffers[RIGHT_CHANNEL];

for (uint32_t i = 0; i < processInfo.samplesToProcess; i++){

// --- read sample
PCMSampleOutput output = selectedSampleSource->readSample(readIndex,
                                                          phaseInc);
leftOutBuffer[i] = output.audioOutput[LEFT_CHANNEL];
rightOutBuffer [i] = output.audioOutput[RIGHT_CHANNEL];
```

11.7 Exercises

11.7.1 SynthLab-PCM$_{DM}$ Mellotron Core: Loop Points

Mellotron samples are taken from the beginning to the end of the note-event and do not include loop points. In this state, this creates a one-shot sample playback synth, but the samples are six to ten seconds in length. Using software such as *Wavosaur* (free), you can set the loop points for the sustain sections and embed the MIDI unity note. Create a set of transient plus loop samples from the original Mellotron files. There are 35 files per patch, so you will become very good at identifying the transient loop points, and you will find that it is quite challenging to get a click-free loop. You might also investigate software such as *SampleRobot* that can find loop points for you.

11.7.2 SynthLab-PCM$_{DM}$ Mellotron Core: Start Point Mod Knob

All mod knobs are unassigned for the PCM cores. The Mellotron samples are delivered in long one-shot un-looped form. Use two mod knobs to set the loop start and end points on-the-fly. HINT: You will need to modify the *IPCMSampleSource* object to set the loop points.

11.7.3 SynthLab-WT$_{DM}$: Hard Sync

PCM samples are usually not hard-sunk as their information is often instrument-specific. Use a mod knob and implement the same self hard-sync as the wavetable cores. Then, implement shape modulation using a mod knob as the controller and using a similar waveshaping technique.

11.7.4 Advanced Module: Create a PCM Split Core

Splitting a synth keyboard involves mapping one patch (PCM sample set) to one range of notes and another patch to the remaining notes. Middle-C is often chosen as the split note. Create a core that splits the keyboard and chooses from two different patches depending on the key being depressed.

11.7.5 Advanced Interface: On-Demand PCM Sample Access

If you are already knowledgeable about multi-threaded applications, propose and implement a system for loading PCM samples on-demand from the disc. This will require numerous heuristic options regarding how a worker thread that accesses the disc can add data to a growing audio buffer (HINT: see the lock-free ring buffers, as described in Chapter 20 of *Designing Audio Effects Plugins in C++ 2nd Ed.*).

Bibliography

Braut, Christian. 1994. *The Musician's Guide to MIDI*, Chap. 5–7. Alameda: SYBEX.

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 1*. https://www.midi.org/specifications-old/item/dls-level-1-specification, Accessed October 14, 2020

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 2*. https://www.midi.org/specifications-old/item/dls-level-2-specification, Accessed October 14, 2020

12 Synthesizer Filters

Filters play a critical role in most synthesizer algorithms, from simple damping in the Karplus-Strong algorithms to searing filter sweeps in analog modeling synths. Interestingly, many classical DSP textbooks explain the design theory, but they rarely reveal the implementations that we need in audio applications: independent controls for cutoff frequency (f_c) and quality factor (Q) and stability over the audio spectrum from 20 Hz to ~20,480 Hz (10 octaves). Early digital synth filters, designed directly in the z-plane, can be found in Chamberlain, F.R. Moore, and Dodge and Jerse. In the late 1980s, Motorola introduced a set of DSP application notes that demonstrated algorithms with independent controls and which were implemented as IIR filters. More recently, Zavalishin's excellent and free book *The Art of VA Filter Design* has added a new set of synth filter designs to our repertoire. In fact, this chapter is devoted entirely to these virtual analog designs because they work well for modulation and are simple to implement. In addition, you can find almost all of the other audio filter design types in my FX plugin book.

12.1 Design Summary

As with analog filters, digital filter design has multiple approaches. These include:

1 *Direct z-plane design*: early synth filters, including resonator and all-pole designs
Pros: simple to design and implement; CPU friendly
Cons: limited design flexibility
Sources: Moore, Dodge and Jerse, Chamberlain
2 *Bilinear z-Transform IIR* (BZT): classical analog-to-digital filter designs via the bilinear z-transform
Pros: maps analog s-plane to digital z-plane, preserving the frequency axis
Cons: does not preserve impulse response, and LPF edges have errors due to incorrect mapping of zeros at infinity
Sources: numerous, including Pirkle (2018)
3 *Impulse invariant filters*: classical filter design that preserves the impulse response of the analog filter prototype
Pros: impulse response preserved
Cons: can alias but only useful for lowpass types; filter f_c has errors that get worse as the frequency increases
Sources: Ifeachor and Jarvis, Pirkle (2018)

4 *Wave Digital Filters (WDF)*: model the analog circuit components directly using scattering parameter theory
Pros: design techniques are well documented and may include passive or active components
Cons: advanced circuits with multiple loops and nodes may be difficult to implement
Sources: Fettweis, Smith: RT-WDF, Pirkle (2018)
5 *SPICE modeled filters*: uses a real-time SPICE engine and SPICE scripts for circuit simulation
Pros: exact modeling of components with a long history and thousands of existing models
Cons: CPU intensive
Sources: usually a trade secret
6 *Virtual Analog filters (VA)*: model analog filter block diagrams or analog signal flow graphs at a conceptual level, instead of circuit components
Pros: excellent sound quality and ability to tolerate intense f_c modulation; simple to implement
Cons: ultimately, these are bilinear transform variants and suffer from the same errors with LPF edges and zeros at infinity; it is still possible to produce harsh or erroneous filter output with extreme f_c modulation
Sources: Zavalishin, Pirkle (2018)

12.2 *Q* and Self-Oscillation

One aspect of synth filter design that differs from FX and other applications is that self-oscillation is generally desirable. The filter will self-oscillate when its Q (resonant peaking) value becomes infinite. For our synth filters, this will always be based on a feedback loop gain value that represents a 100% feedback. However, this value is different for every filter, and in one case, the value is 0.0. To create filters that can use the same pair of GUI controls for f_c and Q, I am using a single scale of 1 to 10 for the Q control. Each filter will take this value and map it to a range of values that works for its design. The maximum setting will produce self-oscillation. Self-oscillation becomes particularly interesting when key-track modulation is used; the filter can act as a secondary oscillator and may be tuned to musical intervals based on the MIDI note that is playing.

12.3 Analog Magnitude Matching at Nyquist

One annoyance with bilinear z-transformed filters is that lowpass and bandpass filters do not have a matched analog frequency response due to the zero at Nyquist – this is discussed in detail in my FX plugin book. Each of my synth filters includes a modification to produce a finite gain at Nyquist in order to more closely mimic the analog counterparts. As with the MMA, impulse invariant, and Vickanek analog matched filters in the FX book, there is some error from the true analog magnitude, and it varies from −12.4% to +1.1%. Even with the error, I feel that these are superior to the BZT versions because of the added top-end to the response. In this chapter, the option is named "analog FGN" (Finite Gain at Nyquist). All filters produce both standard and analog FGN outputs, so you are free to choose whichever you like.

12.4 Zavalishin's Virtual Analog Filters

Zavalishin's VA design technique (2012) simulates analog filters by implementing the analog block diagrams, consisting of summers, multipliers (amplification or attenuation), and integrators, with their digital versions. The summers and multipliers translate easily into their corresponding DSP

structures, and there are more than two-dozen different digital integrators that may be considered. This idea goes back to El-Masry's *RLC* simulation technique (El-Masry and Sakla 1979) and has been re-packaged over the years in different forms, but it is based on the notion of digital integrator replacement. This method did not gain traction early on because it usually results in delay-free loops, also called "zero delay feedback loops" or "zero delay feedback" or ZDF. Zavalishin presents an algebraic solution to the delay-free loop problem. In 2015, I extended Härmä's "Implementation of Recursive Filters Having Delay Free Loops" (1998) to provide another method of solving the problem that uses simple filter stimuli and responses with a universal transfer function. For this chapter, the phrase "Virtual Analog," or VA, refers to these integrator replacement designs.

12.4.1 Digital Integrator Replacement

Figure 12.1 shows the fundamental concept behind integrator replacement as a method of digitizing an analog filter. A simple first order *RC* filter is shown in Figure 12.1(a), and it is easy to derive the cutoff frequency $\omega_c = 1/RC$. The analog block diagram is shown in Figure 12.1(b), and you can see the location of the multiplier and integrator inside of a delay-free loop. This is well documented in Zavalishin's text, using simple voltage dividers as well as Laplace transforms. You can also download an alternate derivation from www.willpirkle.com/synth-lab that uses signal flow graph theory – in any event, you need to know a bit about analog block diagrams and filtering theory to understand these derivations.

While digital summers and coefficient multipliers are simple to implement, the analog integrator does not have a directly corresponding digital version. The analog integrator's transfer function $H(s) = 1/s$ may be analyzed in the frequency domain as a kind of first order lowpass filter with a magnitude response of infinity at $\omega = 0$, or as a time domain algorithm that calculates the area under the analog signal's curve. If we think about calculating the area under the signal's curve, we know that any digital version will have errors because the signal itself is discretized where the smooth analog curve is broken into samples.

The bilinear integrator approximates the analog integrator with a trapezoidal area calculation. Figure 12.2(a) shows two samples from the audio input, the current input $x(n)$, and the previous input $x(n-1)$. If the integrator output $y(n)$ represents the area under the input signal's curve, then the area of this trapezoid can be found using the familiar geometric equation, where the value $T/2$ represents the halfway point between the two input samples and $T = 1/f_s$.

$$y(n) = y(n-1) + \frac{T}{2}(x(n) + x(n-1)) \tag{12.1}$$

Taking the z-transform of (12.1) by inspection, we get:

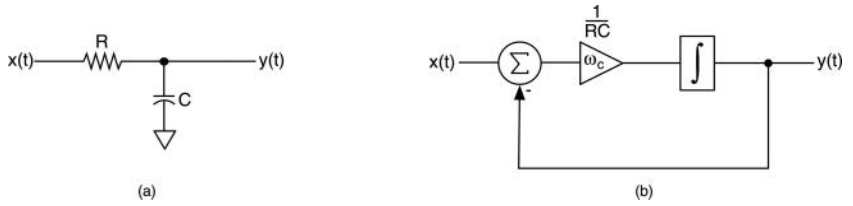


Figure 12.1 (a) A simple RC lowpass filter and (b) the analog block diagram

$$H(z) = \frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}} = \frac{T}{2}\frac{z+1}{z-1} \tag{12.2}$$

Figure 12.2(b) shows the bilinear integrator in transposed canonical form and has been labeled with an intermediate node, $s(n)$, that represents the output of a storage block or z^{-1} storage register and shows that the output is a scaled version of the input, plus the output of that storage register. This separates the equation into two parts: the part dependent only on the current input signal $x(n)$ and the part dependent only on the output of the storage system $s(n)$ or the delayed part. Equation (12.3) shows how the output $y(n)$ is separated into these two components and that it is a function of the input $x(n)$, the static coefficient $T/2$, and the storage component $s(n)$.

$$y(n) = \frac{T}{2}[x(n)] + s(n) \tag{12.3}$$

Performing digital integrator substitution on the first order lowpass filter produces the diagram in Figure 12.3(a). The coefficient multiplier that was $\omega_c = 1/RC$ has been replaced with ω_a, which represents the pre-warped analog cutoff frequency that is inherent in the bilinear transform calculation. The original analog cutoff frequency is assigned directly to the desired digital frequency, ω_d, and then the "incorrect" analog version is calculated with the standard pre-warping Equation (12.4).

$$\omega_a = \frac{2}{T}\tan\left[\frac{\omega_d T}{2}\right] \quad T = \frac{1}{f_s} \tag{12.4}$$

Figure 12.3(b) combines the series multipliers into one value, $g = \omega_a T/2$, and ignores the details of the delay structure. Figure 12.3(c) takes this one step further and simplifies the whole feed-forward structure into a block with *input*, *output*, g and s values. This notation will be used throughout the chapter. At this point, everything is in place, but we still have to deal with the delay-free loop.

12.5 Resolving Delay-Free Loops in VA Structures

The underlying issue with delay-free feedback loops is that you wind up with equations that look like (12.4), where the discrete output $y(n)$ appears on both sides of the equation, implying that you

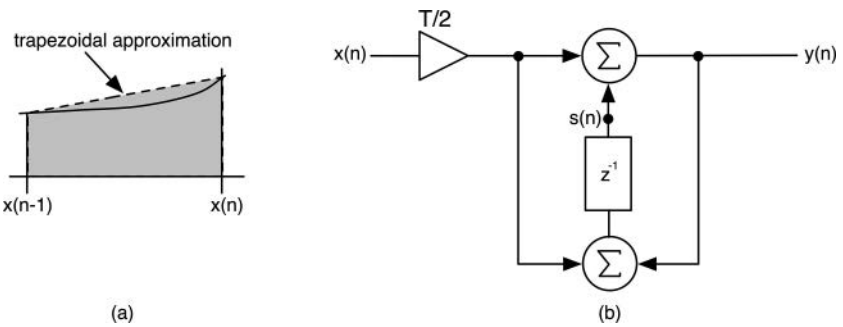


Figure 12.2 (a) The bilinear (trapezoidal) approximation of the area under a curve, showing the error portion between the dotted line and the curve, and (b) transposed canonical form of the bilinear integrator

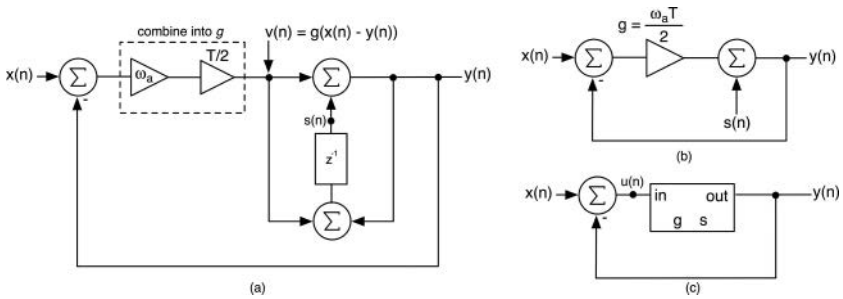


Figure 12.3 (a) The analog integrator is replaced with a digital bilinear integrator (b) a shorthand version after combining the two input multipliers into one, and ignoring the exact details of the bilinear structure and (c) simplified notation showing a block with input, output, *g*, and *s* ports

need the current output value in order to calculate the current output value. Prior to the late 1990s, you had to place a z^{-1} storage register in the loop to prevent feedback loop dependence, but this would alter the transfer function of the structure.

12.5.1 Zavalishin's Algebraic Method

Consider the structure in Figure 12.3(c), where the input to the shorthand block has been labeled $u(n)$ (commonly used in signals and systems theory). Tracing the signal, you can write (12.5), which places $y(n)$ on both sides of the equal sign.

$$\begin{aligned} u(n) &= x(n) - y(n) \\ y(n) &= gu(n) + s(n) \\ &= g(x(n) - y(n)) + s(n) \end{aligned} \tag{12.5}$$

Zavalishin's technique requires you to make somewhat of a leap of faith regarding the difference equation in (12.4) – ignore the independent variable n, which results from the discretization of the signal, and treat the equation as instantaneous. This results in Equation (12.6).

$$\begin{aligned} y &= g(x - y) + s \\ y &= \frac{gx + s}{1 + g} = \frac{g}{1 + g}x + \frac{s}{1 + g} \\ y &= Gx + S \\ G &= \frac{g}{1 + g} \quad S = \frac{s}{1 + g} \end{aligned} \tag{12.6}$$

At this point, you let $G = g/(1 + g)$ and $S = s/(1 + g)$, which shoehorns the equation into the format $y = Gx + S$, which follows the idea of separating the difference equation into input and storage parts. Now, refer to Figure 12.3(a), and look at the node marked $v(n) = g(x(n) - y(n))$. Ignoring the discrete time variable n, you get $v = g(x - y)$. Now, substituting Equation (12.5) as the y component, you can write:

$$\begin{aligned} v &= g(x-y) = g\left[x - \frac{g}{1+g}x - \frac{1}{1+g}s\right] = g\left[x\frac{1+g}{1+g} - \frac{g}{1+g}x - \frac{1}{1+g}s\right] \\ &= g\left[\frac{x+gx-gx-s}{1+g}\right] = \frac{g}{1+g}(x-s) \\ &= G(x-s) \end{aligned} \tag{12.7}$$

Now, by observation, you resolve the delay-free loop. This is accomplished with two steps:

1 Change the input coefficient multiplier to $G = g/1 + g$, and note that this is the same as cascading the original g coefficient with another $1/1 + g$ coefficient
2 Relocate the original feedback loop from the output to the $s(n)$ location

Both steps are shown in Figure 12.4, where 12.4(a) shows the setup with $v(n)$, and 12.4(b) relocates the delay loop and merges the coefficients into one G value. When you use Zavalishin's method, these will always be true – the loop coefficient gain multiplier will be merged with another factor (g becomes G), and the loop source will be relocated from the output to a storage location, where $s(n)$ is the output of a pure delay network.

12.5.2 Modified Härmä Method

My loop resolution method is called the Modified Härmä (MH) method because it is based on Härmä's original design, which resolved one specific type of delay-free loop where the processing is in the feedback path, and the feed-forward path consists only of non-storage components. The MH method extends that theory to resolve any kind of delay-free loop. You can find the derivation that does not require ignoring the discrete time variable n and that uses a familiar infinite series in Pirkle (2014). The methodology is also different – instead of algebra, you stimulate the loop by injecting two values (0.0 and 1.0) into it under specific conditions and tabulating the results. Next, you use a universal difference equation template and plug in the values – the loop can then be resolved by inspection. The MH method requires five steps. The first three involve stimulating the loop to find three values:

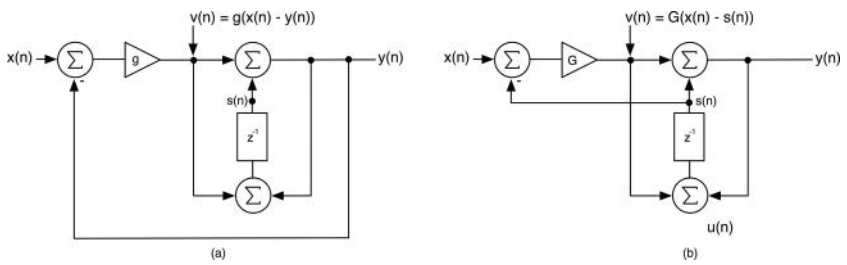


Figure 12.4 (a) The delay-free loops structure with $v(n)$ labeled and (b) the resolved loop done by examination of Equation (12.7)

1 $u_o(n)$ = the temporary loop input value
2 $y_o(n)$ = the temporary loop output value
3 X = the loop gain without any delay (storage) elements in place
4 The fourth step is to plug these values into the delay-free resolved difference equation for a single loop system; this is the result of the derivation of the method and is covered in the source.

$$y(n) = y_o(n)\frac{1}{1-X} \tag{12.8}$$

5 The last step is to find $u_o(n)$ for the filter implementation to see how the new difference equation will alter the structure – it is here that you will be able to relocate the loop and modify the loop coefficient by inspection. You can then prove to yourself that the difference equation implements the new structure.

Figure 12.5 shows the MH method's first three steps for a system consisting of a single loop, with the processing block in the feedback path. With these values and Equation (12.8), you can easily write the difference equation:

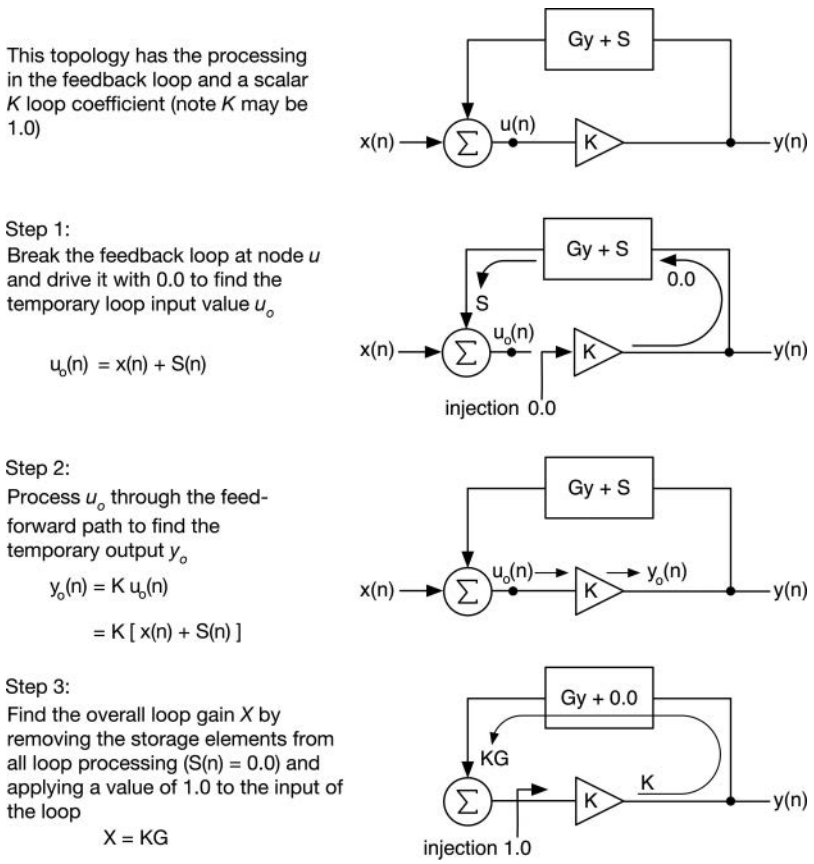


Figure 12.5 The first three steps in the MH method for resolving delay-free loops produce the values $u_o(n)$, $y_o(n)$, and loop gain X

$$y(n) = y_o(n)\frac{1}{1-X}$$
$$= \frac{K[x(n)+S(n)]}{1-KG} \tag{12.9}$$

For the final step, find the exact value for $u_o(n)$ – this will vary depending on the loop structure. Notice that the operations are the same as those in Zavalishin's algebraic method: the loop is re-located from the output $y(n)$ to a storage location $S(n)$, and the loop coefficient is augmented with another series multiplier.

$$u_o(n) = x(n)+Gy_o(n)+S$$
and
$$y_o(n) = Ku_o(n)$$
therefore
$$u_o(n) = x(n)+GKu_o(n)+S \tag{12.10}$$
$$u_o(n) = \frac{x(n)+S}{1-KG}$$

Now you can inspect (12.8) and (12.9), and set up the final structure shown in Figure 12.6. Notice that the output of the loop-processing block seems to go nowhere – this is correct as the output value is not applied anywhere else. In fact, you don't need to calculate the output value in this case as the loop is dependent on the $S(n)$ value.

For completion, in multiple loop systems, the generalized difference equation solution is shown in Equation (12.11).

$$y(n) = \frac{1}{1-X}\left[f(u(n)) + \sum_{i=1}^{M} \beta_i s_i \right]$$
$f(u(n)) =$ the processing functon in the feed-forward path
or $x(n)$ if no function in path
$M =$ the number of delay elements in the structure
$X =$ the loop gain coefficient
$s_i =$ the output of delay element i
$\beta_i =$ the feedback multiplier for delay element i (12.11)

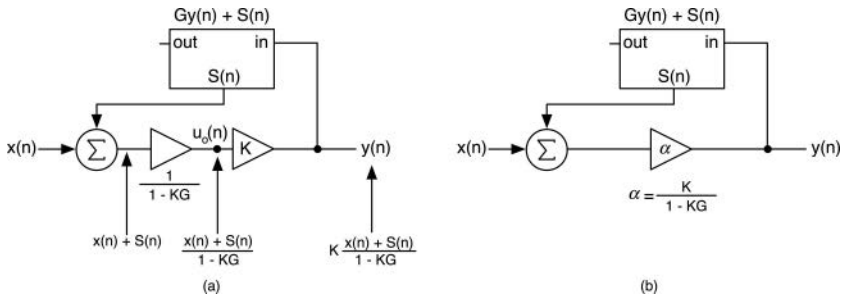


Figure 12.6 (a) The final structure after loop resolution and (b) combining the series loop coefficients into a single value α

12.6 VA Filters from Primitive Analog Block Diagrams

The simplest VA filters are derived from first and second order primitive analog filter block diagrams. Primitive means that the diagrams consist only of summers, multipliers, and integrators, with no higher-level structures embedded.

12.6.1 First Order VA Filters

Figure 12.7(a)–(c) shows the analog filter block diagrams and virtual analog realizations for the three basic first order filters: LPF, HPF, and APF. In all cases, the single coefficient α is calculated the same, using Equation (12.5). Converting this to C++ code, we can write the update calculation for α (note that the calculation for *wa* and *g* may be combined and simplified a bit).

```
double wd = 2*pi*Fc;
double T = 1/SampleRate;
double wa = (2/T)*tan(wd*T/2);
double g = wa*T/2;  // <--- i.e. g = tan(wd*T/2)

// --- final calculation
double alpha = g/(1.0 + g);
```

These three filters may be combined into a single structure that implements all of them at once as they are based on the same core structure, and we only need to resolve that delay-free loop once. We will use this kind of structure for the Korg35 filters.

12.6.1.1 First Order LPF Analog FGN

I modified this algorithm to produce better analog matching gain at Nyquist using the Korg35 sub-circuit in Figure 12.9(c). The method involves blending in some of the HPF response, whose gain at Nyquist is fixed at 1.0. It turns out that for this design, the alpha coefficient may be used directly as a scaling factor, albeit with a bit of magnitude matching error. Figure 12.7(d) shows the normal VA LPF frequency response, with the Nyquist magnitude pinned down to 0.0, while 12.7(e) shows the matched-analog response, with finite gain at Nyquist. Figure 12.7(f) and (g) shows the HPF frequency and APF phase responses.

12.6.2 Second Order State Variable Filter

For filters higher than first order, there will almost always be multiple analog circuit realizations and varying block diagram implementations. The well-documented State Variable Filter (SVF), also known as the Kerwin-Huelsman-Newcomb (KHN) filter after its inventors, is a second order structure that produces all four basic filters at once: LPF, HPF, BPF, and BSF. Figure 12.8(a) shows the analog block diagram that consists of two synchronously tuned integrators inside of two delay-free loops, an inner loop, and an outer loop. Figure 12.8(b) shows Zavalishin's VA realization. Notice that the analog block diagram references ζ, the damping factor in classical analog filter theory, but in musical applications, we prefer to replace it with the term Q, which stands for quality factor, where:

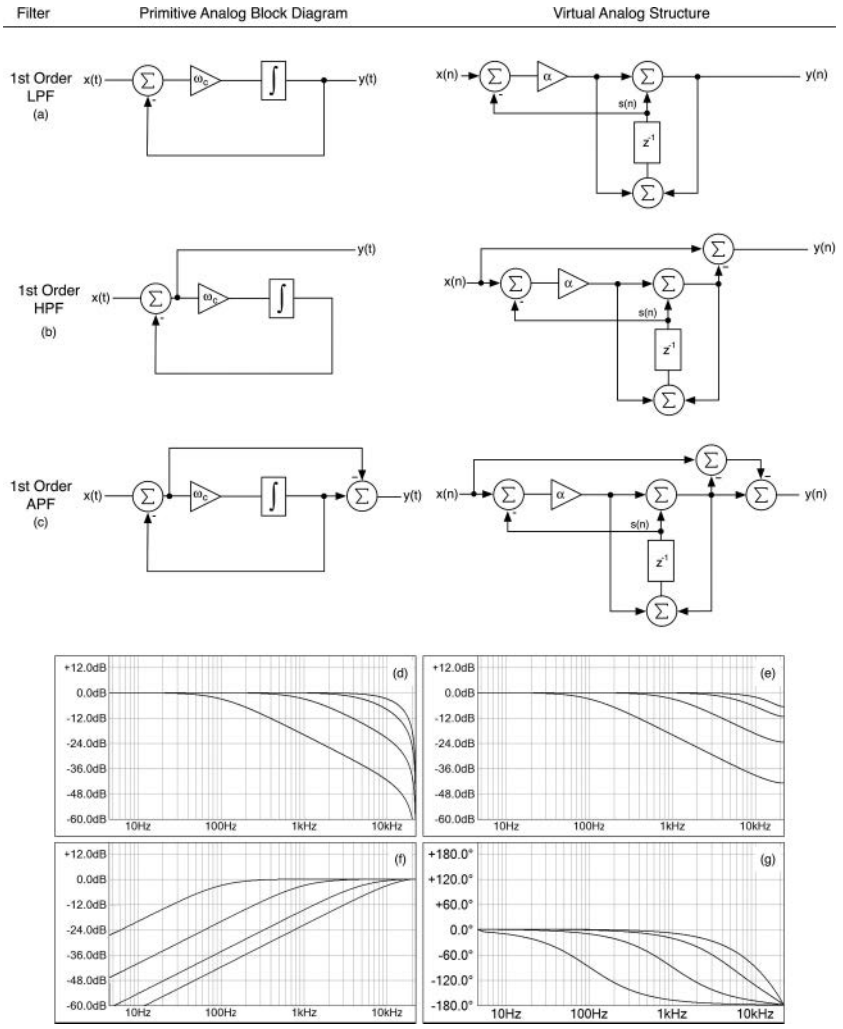


Figure 12.7 Primitive analog block diagrams and their resulting VA filter structures for (a) first order low-pass filter, (b) first order high-pass filter, and (c) first order all-pass filter (d) the normal first order BZT LPF frequency response with zero gain at Nyquist (e) analog FGN LPF frequency response (f) first order HPF frequency response and (g) first order APF phase response; all plots are shown with f_c = 100 Hz, 1 kHz, 5 kHz, and 10 kHz

$$Q = \frac{1}{2\zeta} \tag{12.12}$$

You can see that the bandpass output is fed back into the input summer via the scaling coefficient $2\zeta = 1$, so the more feedback, the lower the resonant peak. Notice the relocation of the delay-free loops to storage locations $s_1(n)$ and $s_2(n)$, and the addition of the α_o coefficient in series with the

input loop gain value α. The design equations and output calculations are given in Equation block (12.13) (Zavalishin 2008). After updating the coefficients, you calculate the individual outputs in succession, starting with y_{HP}.

$$\begin{aligned}
&\omega_d = 2\pi f_c \\
&T = 1/f_s \\
&\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right) \\
&\alpha = \frac{\omega_a T}{2} = \tan\left(\frac{\omega_d T}{2}\right) \qquad y_{HP}(n) = \alpha_0\left(x(n) - \beta_1 s_1(n) - \beta_2 s_2(n)\right) \\
&R = \frac{1}{2Q} \qquad\qquad y_{BP}(n) = \alpha_1 y_{HP}(n) + s_1(n) \\
&\alpha_0 = \frac{1}{1 + 2R\alpha + \alpha^2} \qquad y_{LP}(n) = \alpha_1 y_{BP}(n) + s_2(n) \\
&\rho = 2R + \alpha \qquad\qquad y_{BS}(n) = x(n) - 2R y_{BP}(n)
\end{aligned} \tag{12.13}$$

12.6.2.1 SVF Analog FGN

As with the first order filters, the SVF LPF also includes an erroneous zero at Nyquist, which is a result of the bilinear transform. In order to correct the gain at Nyquist, I observed that the value residing in the first integrator's state register, labeled $s_1(n)$ in Figure 12.8, is actually a mixture of the current bandpass output plus the high-pass output scaled by α. We can take advantage of the fact that the HPF gain at Nyquist is easily calculated (and mostly equals 1.0 except for very high combinations of f_c and Q). I used a similar method of blending in this "correction" signal knowing the values of α and the desired gain at Nyquist. This works very well up until just before Nyquist, where we observe a small high frequency magnitude offset due to the fact that the internal HP-F+BPF signal has the same value of Q applied to it. The Nyquist gain still matches the analog version, and the filter is still stable right up to f_c = Nyquist. To calculate the analog matching coefficient σ in Figure 12.8, I use Equation (12.13), evaluating the analog matching transfer functions at f = Nyquist. The equations for M_{LPF} and M_{HPF} are the standard equations to calculate the magnitude of the analog filter $H(s)$ for second order LPF and HPF, respectively. In our ratio, their denominators cancel each other out, so we are left with a relatively simple coefficient equation to calculate. Figure 12.8(c) and (d) shows the normal and analog matched LPF outputs, while Figure 12.8(e) and (f) shows the SVF HPF and BPF outputs at a variety of f_c and Q settings.

$$\begin{aligned}
&f_o = \frac{f}{f_c} \qquad\qquad \zeta = \frac{1}{2Q} \\
&M_{LPF} = \left|H(s)_{LPF}\right| = \frac{1}{\sqrt{\left(1 - f_o{}^2\right)^2 + 4\zeta^2 f_o{}^2}} \qquad M_{HPF} = \left|H(s)_{HPF}\right| = \frac{f_o{}^2}{\sqrt{\left(1 - f_o{}^2\right)^2 + 4\zeta^2 f_o{}^2}} \\
&\sigma = \left.\frac{M_{LPF}}{\alpha M_{HPF}}\right|_{f=Nyquist} = \frac{1}{\alpha f_o{}^2}
\end{aligned} \tag{12.14}$$

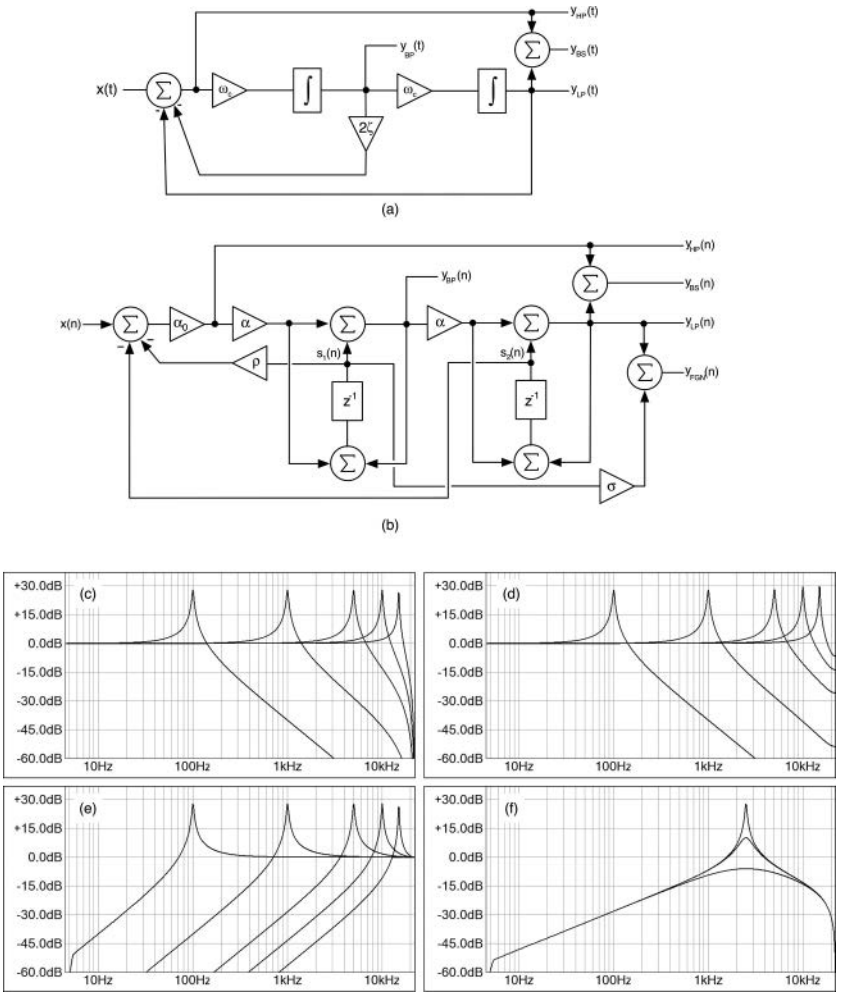


Figure 12.8 (a) The SVF analog block diagram (b) the VA realization including the analog matched LPF output $y_{LPM}(n)$ (c) normal SVF LPF frequency response and (d) LPF frequency response with analog matching at Nyquist (e) SVF HPF frequency response with f_c = 100 Hz, 1 kHz, 5 kHz, 10 kHz and 15 kHz and Q = 25 and (f) SVF BPF frequency response with f_c = 2.5 kHz and Q = 0.5, 10 and 25

12.7 VA Filters from Signal Flow Graphs

Signal flow graphs may be used as a shorthand method for notating primitive analog block diagrams. However, they excel when used to diagram more complex block algorithms consisting of series or parallel branches of sub-filters. These high-level signal flow graphs also describe summers and coefficients, but the integrators are replaced with filters: *LPF1* and *HPF1* for first order lowpass and high-pass filters, respectively. Higher order sub-filter blocks are equally possible. For these designs, each first order building block is implemented as a VA filter, then these are

connected together, often involving delay-free loops around the sub-filters. The delay-free loop resolution methods still work as usual, with the sub-filters replaced with an equation that is always broken into the current input and storage components, or $y(n) = Gx(n) + S(n)$.

12.7.1 Korg35 Second Order Filters

Korg35 second order resonant filters used in the MS-10 and early MS-20 are voltage controlled versions of the Sallen-Key filter topology. Sallen-Key filters feature a positive feedback path through a narrow bandpass filter that reinforces the resonant frequency and eventually allows the filter to self-oscillate. Korg took advantage of this to produce a simple but effective highly resonant filter. Both LPF and HPF are second order types and will self-oscillate, but the Korg35 HPF has an interesting quirk; its roll-off slope is +6 dB/octave instead of the normal +12 dB/octave. This produces an HPF with more bass response. The non-standard roll-off is due to the manner in which Korg implemented the HPF, using an analog filter shortcut – take any analog filter circuit, ground the input, then lift and drive the ground connections to produce the complementary version with the same f_c and Q values.

I designed the Korg35 filters directly from their signal flow graphs, as shown in Figure 12.9(a) and (b). These filters are made of first order sub-filters that are synchronously tuned, meaning they share the same f_c value. The complete derivation is available at www.willpirkle.com/synth-book-derivations/. The Sallen-Key filter has two analog implementations that differ only in terms of the maximum loop gain K that creates the self-oscillation; these use $K = 2.0$. During the derivation, a new variable (β) was needed to act as a feedback scalar once the delay-free loops were resolved. This created the structure in Figure 12.9(c) that is used for each of the VA sub-filters; this filter can generate LPF and/or HPF outputs, and includes the new feedback coefficient. Notice that the loop resolution also generated the loop coefficient α_o as usual. The design of Equation (12.15) is straightforward; the α_N coefficients are the same for all filters, and only sub-filters 2 and 3 need a β coefficient, which is different from LPF to HPF. Figure 12.9(d) shows the normal LPF and HPF responses with their different roll-off slopes.

$$g = \tan\left(\frac{\omega_d T}{2}\right) \quad \alpha = \frac{g}{1+g} \quad \alpha_o = \frac{1}{1 - K\alpha + K\alpha^2}$$

$$\text{Korg35 LPF:} \quad \beta_2 = K\frac{(1-\alpha)}{(1+g)} \quad \beta_3 = \frac{-1}{(1+g)} \tag{12.15}$$

$$\text{Korg35 HPF:} \quad \beta_2 = \frac{-\alpha}{(1+g)} \quad \beta_3 = \frac{1}{(1+g)}$$

12.7.1.1 Korg35 Analog FGN

I modified the Korg35 LPF for finite gain at Nyquist to try to match the true analog response more closely. The first order VA module shown in Figure 12.9(c) includes the analog FGN matching described in Section 12.3 and generates all three outputs at once. For the Korg35 LPF, I use the analog FGN output for the *VA1 LPF*. For the *VA2 LPF*, the normal (BZT) LPF output is used to supply the feedback loop, and the analog Nyquist matched output is routed to the filter output. Figure 12.9(e) shows the new version with finite gain at Nyquist. As with the SVF, there is a slight boost at the very highest f_c settings, and the filters are stable right up to f_c = Nyquist.

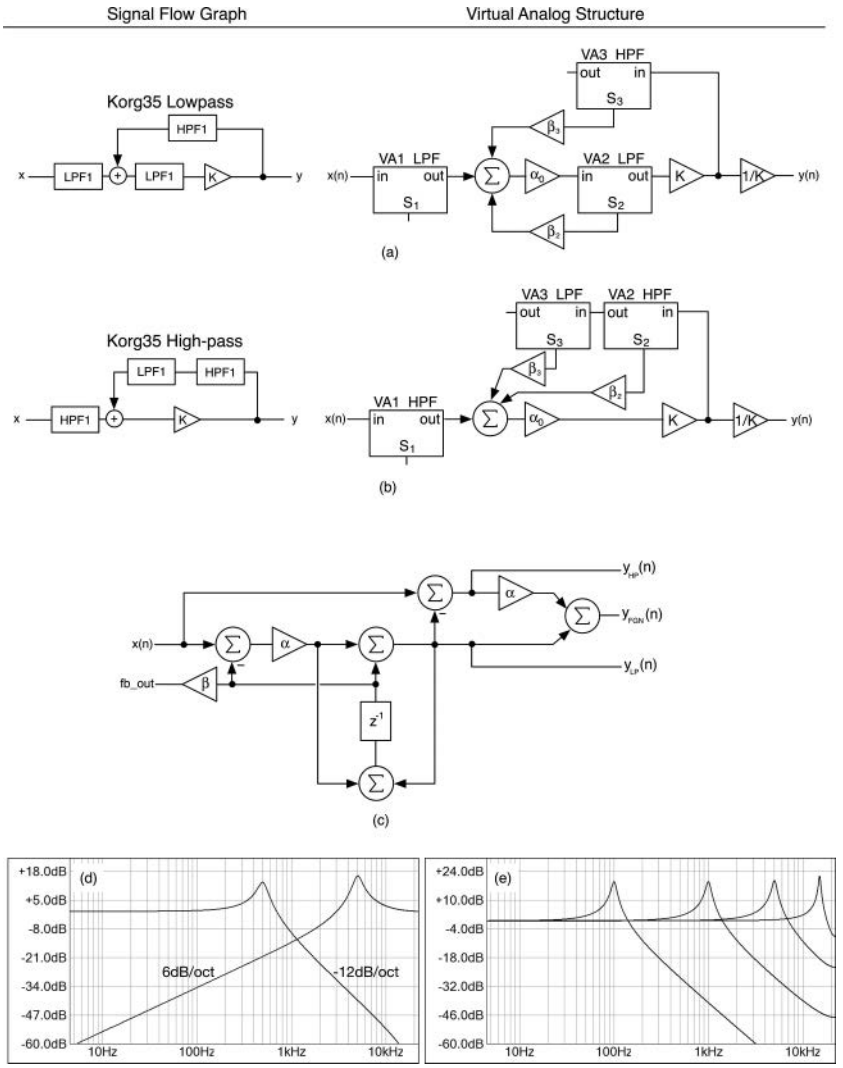


Figure 12.9 The signal flow graph and VA realizations for (a) Korg35 LPF, (b) Korg35 HPF and (c) the combined structure that is used for each of the Korg35 sub-filters (d) the LPF and HPF frequency responses show differences in the roll-off slopes with LPF f_c = 500 Hz and HPF f_c = 5 kHz and (e) the Korg35 LPF frequency response with analog Nyquist matching and f_c = 100 Hz, 1 kHz, 5 kHz and 15 kHz with K = 1.9

12.7.2 Moog Fourth Order Ladder Filter

Robert Moog invented a filter that became known as the Moog ladder filter; this might be the most celebrated, copied, and tweaked of any manufacturer's filters. Several integrated circuit companies made variations on this design, including the Curtis CEM3328, CEM3372, and Precision Monolithics PMI SSM2044. They optimized, modified, tweaked, and extended the design. The

devices were used in countless synthesizers from nearly every company in the 1970s and 1980s. A unique feature of this filter is that changes in resonance create changes in the overall filter gain; as the Q increases, the filter gain drops, as shown in Figure 12.10(b). In addition, you can see that the resonant (peak) frequency also increases as Q increases. The roll-off slope is 24 dB/octave as a fourth order filter.

The Moog ladder filter consists of a cascade of four first order synchronously tuned LPF stages in a global negative feedback loop, as shown in Figure 12.10(a). Putting four first order LPFs in series creates a fourth order filter, but it won't be resonant. The way this filter implements resonance is based on the phase response of each first order section. The phase shift at the cutoff frequency is −45 degrees for a first order LPF stage. Each successive stage then adds another −45 degrees of phase shift. After going through four of these filters, the phase shift at f_c will be −180 degrees, exactly out of phase with the input. This output is fed back into the input through a negative scalar, $-K$, which flips the phase at the cutoff so it is back in-phase with the input. This amplifies the cutoff frequency along with the frequencies that are very close to it resulting in a resonant peak. If $K = 0$ there is no feedback and no resonance. As soon as K becomes non-zero, the Q increases, and peaking occurs. When $K = 4$, we achieve 100% feedback through the loop, and the filter will self-oscillate.

The loss of low frequency gain as the Q is increased is sometimes considered problematic, especially if the filter is being used on a bass instrument. However, the reduction in gain also prevents the filter from overloading and may be part of the device's lore as a "musical filter." Curtis Electromusic Specialists (the same company that made the CEM3328) provided a way of controlling the loss of gain all the way up to no loss at all by designing this feature into one of their ICs in 1984. The gain is compensated by feeding some of the inverted input signal into a summer prior to the gain element $-K$. The doubly inverted (in-phase) signal works against

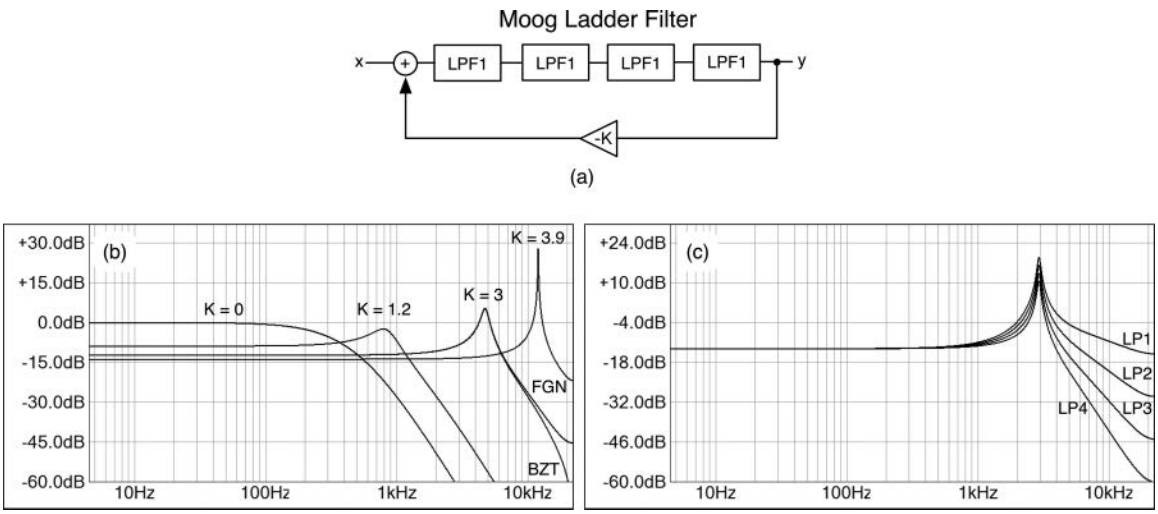


Figure 12.10 The Moog ladder filter (a) signal flow block diagram (b) notice how the bass response is reduced as the Q is increased shown with f_c = 100 Hz, 1 kHz, 5 kHz and 15 kHz each with increasing Q (K) value; the BZT and analog FGN curves are shown for $K = 3$, and for $K = 3.9$ the analog FGN response is indicated (c) the modified additional outputs generate three more filter types including a resonant first order variety

the inverted signal in the feedback path. The filter in the ARP2600 is also a Moog ladder filter derivative that incorporates gain compensation. Zavalishin proposed simply boosting the input x by a factor of $1 + K$, an elegant and simple modification. A variable version can then be implemented by making the boost factor $1 + aK$ where a is on the range of 0.0 to 1.0, as shown in Figure 12.11(a).

Figure 12.11(a) shows the final VA realization after the delay-free loop has been resolved; the complete derivation is available at www.willpirkle.com/synth-book-derivations/. Here, the single feedback loop becomes four parallel loops, each originating from one of the storage registers in the sub-filters, and the familiar loop gain coefficient appears at the input to the first filter. Note that the input scaling and extra outputs are optional. The Moog ladder filter block diagram is ripe for hacking. One easy addition is to tap off of the output of each of the series LPFs, which produces three more outputs for first order, second order, and third order versions of the filter. The first order filter also includes a resonant peak, and the frequency responses are shown in Figure 12.10(c). See the exercises for more hacks. The design equations for the Moog ladder filter are in Equation block (12.16). Note that all sub-filters share the same α coefficient value. The β coefficients are subscripted with the sub-filter number.

$$g = \tan\left(\frac{\omega_d T}{2}\right) \quad \alpha = \frac{g}{1+g} \quad \alpha_o = \frac{1}{1+K\alpha^4}$$
$$\beta_1 = \frac{\alpha^3}{1+g} \quad \beta_2 = \frac{\alpha^2}{1+g} \quad \beta_3 = \frac{\alpha}{1+g} \quad \beta_4 = \frac{1}{1+g} \tag{12.16}$$

12.7.2.1 Moog Filter Analog FGN

The Moog ladder may also be experimentally modified for finite gain at Nyquist, though it requires four more filters that run in a parallel branch with the main set, as shown in Figure 12.11(b). The reason that I cannot simply place the series filters in analog Nyquist matched mode is that this alters the phase response in a nonlinear manner across f_c values and destroys the feedback loop's behavior. A somewhat brute force method is to add the four analog Nyquist matched filters and use their outputs but keep the existing signal path that generates the proper phase response for the feedback loop. All of the filters are sync-tuned, so there is only one calculation of the α coefficient; the filters are also very CPU-friendly, so there is not much added overhead. The new version produces the finite gain at Nyquist which you can see in Figure 12.10(b) and (c). As with the others, the filter is stable right up to f_c = Nyquist. The filters in the upper row do not have the S-ports connected to anything.

12.7.3 VCS3 Diode Ladder Filter

The Diode Ladder Filter first appeared in the EMS VCS3 monophonic synth designed by David Cockerell in 1969. It is also incorporated into the 1982 Roland TB-303 monophonic bass synth. In 1974, Steiner presented another resonant lowpass filter using diodes, but it implemented the Sallen-Key topology. In 1977, Yamaha patented the ring-diode method of implementing a voltage controlled filter in another Sallen-Key topology, and Korg used yet another Sallen-Key ring-diode variation in numerous synths, including the S700. These ring-diode networks are unrelated to the diode ladder filter in this section.

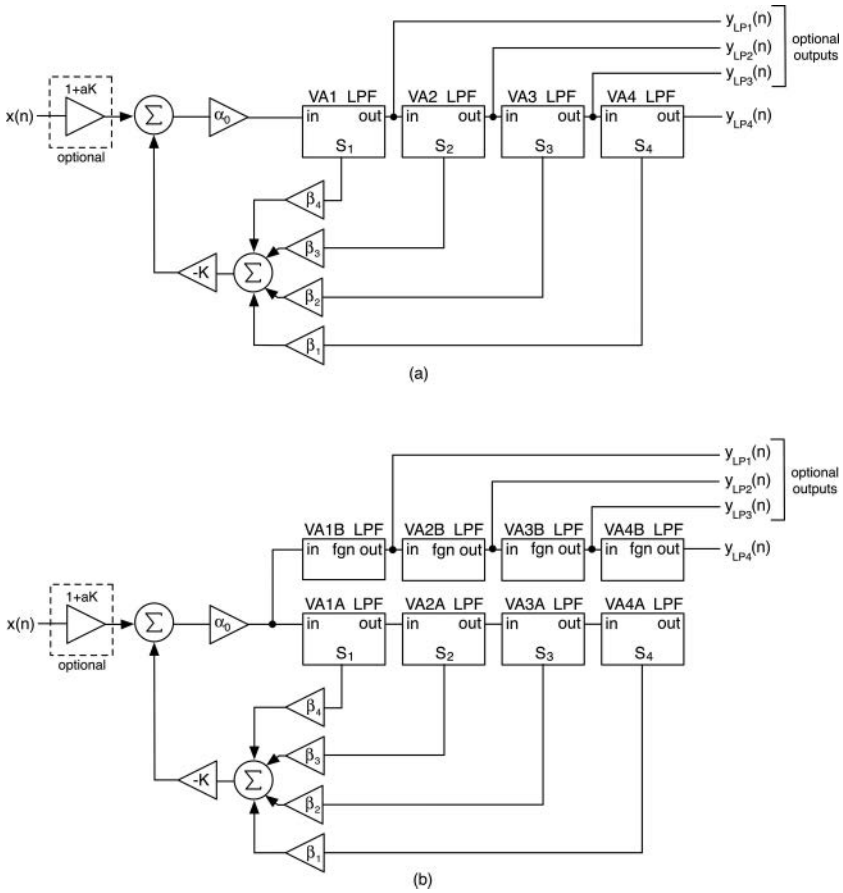


Figure 12.11 The Moog ladder filter (a) VA realization with optional input scaling and output branches, and (b) the analog FGN version; note that the β coefficients are contained in each sub-filter but shown externally here to convey their operation in the feedback loop

This diode ladder filter is based on the Moog ladder filter topology but uses diodes rather than transistors as voltage controlled elements and incorporates multiple feedback paths between sections. The effect of the feedback paths on the signal is two-fold: like the Moog filter, it reduces overall gain as the resonance increases, but the reduction is more extreme (by about 12 dB); second, as the resonance increases, the resonant frequency migrates upwards but never makes it to the cutoff point. At the point of self-oscillation, the resonant peak will have drifted up to $0.707f_c$, and the resonant frequency is never truly correct and equal to the f_c used in the calculations. Figure 12.12(a) shows the signal flow block diagram, which includes three inner delay-free feedback loops as well as the main outer loop with the $-K$ feedback value. Figure 12.12(b) shows how the gain changes more drastically than the Moog ladder filter as the loop gain K is varied. In this filter, K ranges from 0 to 17, at which self-oscillation occurs, and the filter gain is reduced by −24 dB.

There are numerous similarities to the Moog version: four synchronously tuned first order LPFs in series embedded in a global feedback loop that creates positive feedback only at the cutoff

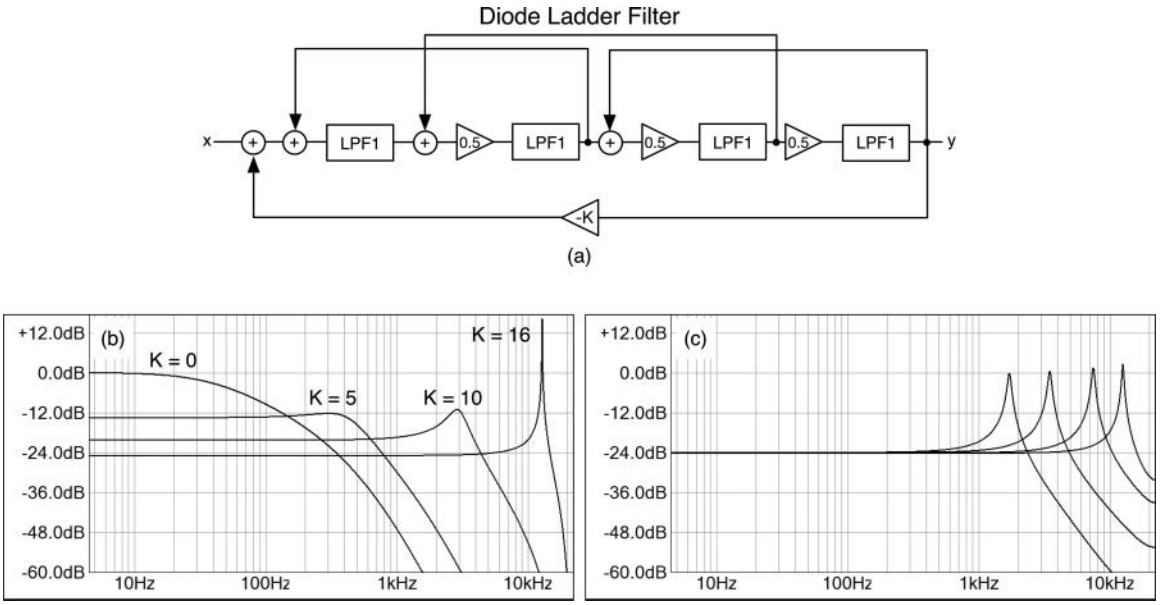


Figure 12.12 The diode ladder filter's (a) signal flow block diagram, and (b) notice how the bass response is reduced as the Q is increased shown with f_c = 500 Hz, 1 kHz, 5 kHz and 15 kHz each with increasing Q (K) value and (c) the experimental analog FGN output at f_c = 2.5 kHz, 5 kHz, 10 kHz and 15 kHz; the error in the resonant peak frequency is normal and correct for this filter

frequency. As soon as K becomes non-zero, the Q increases, and peaking occurs. However, the diode ladder incorporates multiple feedback loops around each section (LPF2 feeds back into LPF1, LPF3 feeds back into LPF2, LPF4 feeds back into LPF3), as shown in Figure 12.13(a). This is a topology known as *leap-frog-form* (LFF). In addition to the feedback paths, there are also attenuators on the inputs to the last three LPFs. Due to the complex interactions of the multiple feedback paths, the filter analysis and synthesis of the block diagram is much more difficult than any of the filters in this chapter. The full derivation is available at www.willpirkle.com/synth-book-derivations/. The crossed feedback paths necessitate adding new local feedback input and output ports to the basic VA LPF, as shown in Figure 12.13(a). Figure 12.13(b) shows the final realization after the delay-free loops have been resolved, and the inner loops have been relocated to and from storage units in each feedback pair.

The design equations for the diode loop coefficients and the coefficients for each sub-filter are given in Equations (12.17) and (12.18).

$$g = \tan\left(\frac{\omega_d T}{2}\right)$$

$$G_1 = \frac{0.5g}{1+g} \qquad G_2 = \frac{0.5g}{1+g-0.5gG_4} \qquad G_3 = \frac{0.5g}{1+g-0.5gG_3} \qquad G_4 = \frac{g}{1+g-gG_2} \tag{12.17}$$

$$\beta_1 = G_4G_3G_2 \qquad \beta_2 = G_4G_3 \qquad \beta_3 = G_4 \qquad \beta_4 = 1$$

$$\alpha_o = \frac{1}{1+KG_4G_3G_2G_1}$$

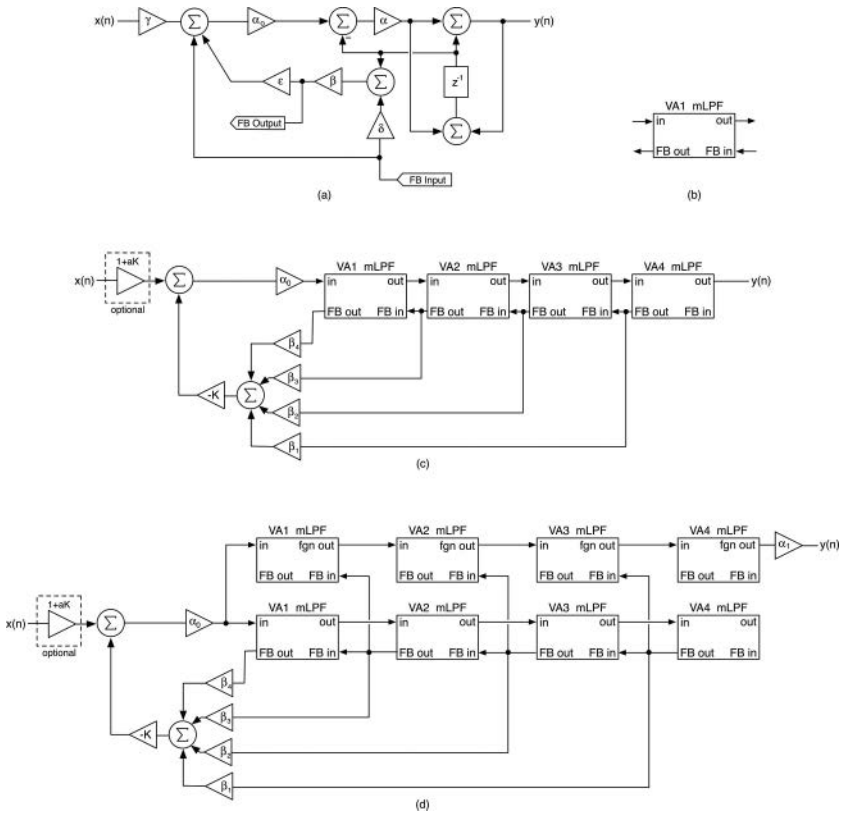


Figure 12.13 (a) The modified first order VA LPF (named mLPF) and (b) its shorthand block diagram with arrows showing the signal flow directions (c) the diode ladder filter realization and (d) the experimental analog FGN version with added α_1 coefficient at the output; note that the β coefficients are calculated within each sub-filter but are shown externally here to convey their operation in the feedback loop

	Filter 1	Filter 2	Filter 3	Filter 4
α_o	1	0.5	0.5	0.5
α	$\frac{g}{1+g}$	$\frac{g}{1+g}$	$\frac{g}{1+g}$	$\frac{g}{1+g}$
β	$\frac{1}{1+g-gG_2}$	$\frac{1}{1+g-0.5gG_3}$	$\frac{1}{1+g-0.5gG_4}$	$\frac{1}{1+g}$
γ	$1+G_1G_2$	$1+G_2G_3$	$1+G_3G_4$	1
δ	g	$0.5g$	$0.5g$	0
ε	G_2	G_3	G_4	0

(12.18)

12.7.3.1 Diode Filter Analog FGN

The diode ladder may also be experimentally modified in a similar brute-force manner as the Moog filter for finite gain at Nyquist, as shown in Figure 12.12(c). I cannot simply place the series filters in analog Nyquist matched mode because this alters the phase response in a nonlinear manner across f_c values and destroys the feedback loop's behavior. Once again, all of the filters are sync-tuned, so there is only one calculation of the α coefficient, and the filters are very CPU-friendly, so there is not much added overhead. As with the others, the filter is stable right up to f_c = Nyquist. Each filter in the upper row gets its local feedback input from the lower row of normal VA filters. As with the Moog ladder analog FGN version, the global feedback loop is taken only from the lower row of VA filters to maintain the proper phase relationship for resonance control. The analog FGN filter requires an additional gain equalizer coefficient, α_1, that is calculated to fit the measured gain as f_c varies, as given in Equation (12.19).

$$a = 1.005381 \quad b = 0.8783896 \quad c = 01.113067 \quad d = -0.2110344$$
$$f = \frac{2\pi f_c}{f_s} \qquad e = \left(\frac{f}{c}\right)^b \qquad \alpha_1 = d + \frac{a-d}{1+e} \tag{12.19}$$

12.8 VA Filters from Conceptual Signal Flow Graphs

I have also created VA filters from conceptual signal flow graphs of filters that do not have an analog closed-form transfer function and even violate some classical analog filter design paradigms. This yields some bizarre results, including filters with the resonant frequency f_r and the cutoff frequency f_c, decoupled and controlled independently, allowing you to move the resonant peak anywhere you like in the response, regardless of the f_c setting, as shown for the second order LPF in Figure 12.14(a). In addition, I've developed strange filters, such as the resonant first shelving filters shown in Figure 12.14(b), and filters with multiple resonant peaks, each independently controllable, as in Figure 12.14(c) and (d). You can find the descriptions and algorithms at https://www.willpirkle.com/novel-hybrid-filters.

12.9 Nonlinear Processing and Self-Oscillation Control

All of the VA filters include the ability for nonlinear processing (NLP), and there are numerous locations within the filter structures to place nonlinear waveshapers. In some cases, the nonlinearities are placed to model nonlinearities in the circuit components: for example, the analog Moog filter is nonlinear in nature due to the way in which the transistor ladder differential pairs conduct current. The differential voltage and current are related by the hyperbolic tangent function. In other cases, the nonlinearities are due to clipping circuits placed inside the filters to try to limit the ultra-high gain that comes from high resonance settings. The nonlinear processors, all of which are based on soft-clip circuits, also aid in taming the harsh clipping that may result from self-oscillation when the filter is driven with program material that has large amounts of energy near the resonant frequency. There are a few strategies for the placement of the NLP, which is often done with *tanh()* waveshapers. Care must be taken to ensure that the waveshapers have a unity-gain or less amplification factor so they do not disrupt the gain in the sensitive feedback loops we have designed.

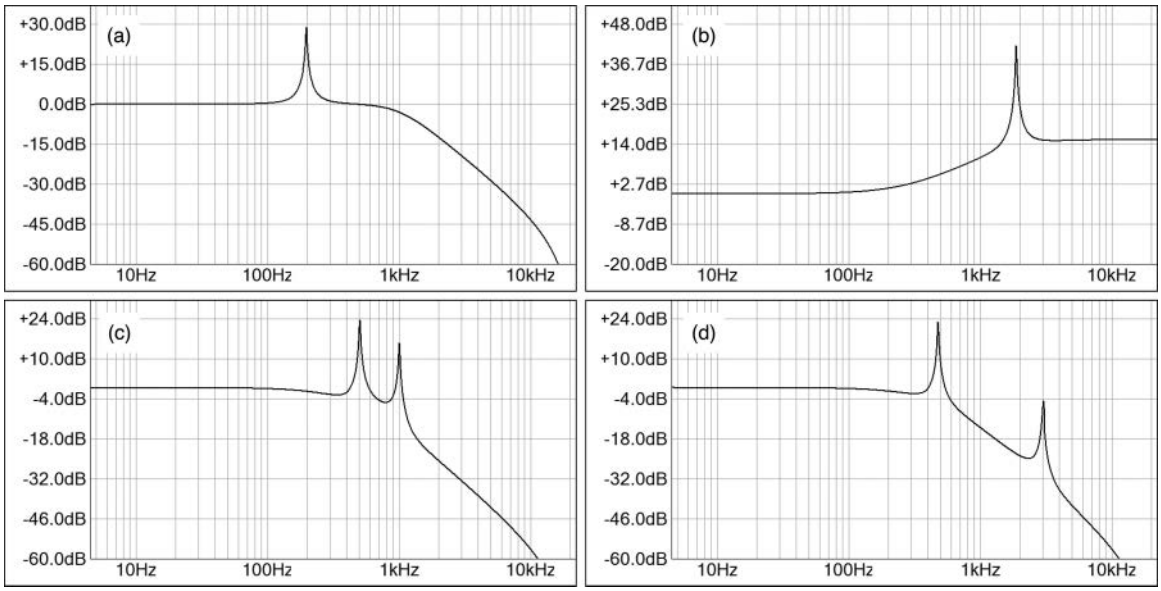


Figure 12.14 Novel VA filters contrived from conceptual signal flow graphs, including (a) second order LPF with f_r = 500 Hz and f_c = 1 kHz, (b) resonant first order high shelving filter with $f_c = f_r$ = 2 kHz and a doubly-resonant LPF with f_c = 500 Hz, (c) f_r = 1 kHz, and (d) f_r = 3 kHz

Figure 12.15(a) shows a saturating integrator (Zavalishin 2008) with a waveshaper in the feedback loop. This saturating integrator may be placed into any of the models as a nonlinear filter. For the Korg35 filters, you might also experiment with placing the NLP in the feed-forward path of the overall loop, as shown in Figure 12.15(b). Both ladder filters include numerous nonlinearities within each filter and within the entire feedback loop. Zavaliahin's budget version is shown in Figure 12.15(c), with a single NLP block at the input and outside of the feedback loop.

12.9.1 NLP with Soft-Knee Peak Limiter

A fundamental issue with any kind of NLP is the addition of aliased components to the signal, which need to be mitigated with oversampling. However, for taming the self-oscillation distortion, it works quite well. After several years of experimentation, I found a solution to producing pure sinusoids during self-oscillation and avoiding distortion using a soft-knee peak limiter with the threshold set between −0.5 dBFS and −1.0 dBFS, as shown in Figure 12.15(d) on the output of the Moog ladder filter. This method is implemented for all of the SynthLab filters; you may, of course, disable the limiters if you like. In order to produce smooth distortion, you move the threshold from nearly the topmost value (0.0 dBFS) to a much lower value, then increase the limiter's makeup gain to bring the signal up to the nominal level – the soft knee and low threshold will produce rounded, soft-clipped waveforms.

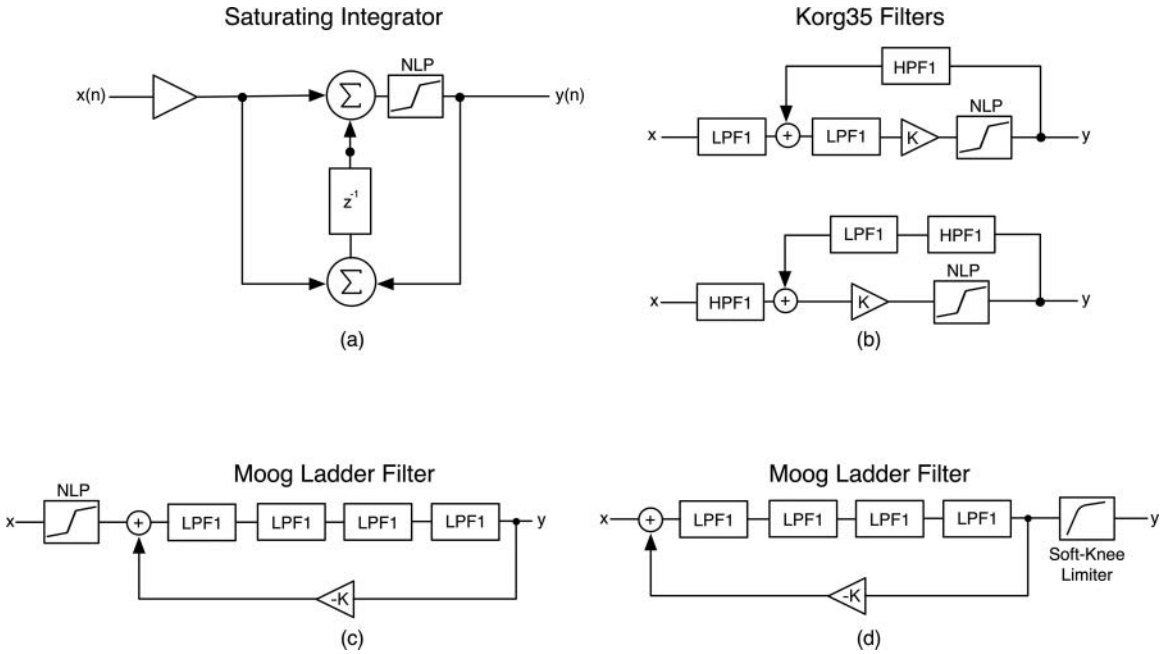


Figure 12.15 Nonlinear blocks in the VA structures shown inside (a) the saturating integrator, (b) the Korg35, and (c) Moog signal flow block diagrams, while (d) shows the location of the peak limiter on the filter output, always outside of any feedback loop

12.10 Synth Filter Objects

The SynthLab *VAFilterCore* object implements all of the filters discussed in this chapter. In order to simplify the object, I created smaller C++ objects for each filter.

1 *VA1Filter*: first order LPF, HPF, APF, and LPF with analog FGN
2 *VASVFilter*: state variable filter with second order LPF, HPF, BPF, BSF, and LPF with analog FGN
3 *VAKorg35Filter*: second order *Korg35* LPF and HPF (including HPF's 6 dB/octave roll-off)
4 *VAMoogFilter*: fourth order Moog LPF; normal and analog FGN; and the first, second, and third order LPF variations
5 *VADiodeSubFilter*: modified *VA1Filter* structure that includes the additional diode specific coefficients
6 *VADiodeFilter*: fourth order diode LPF and analog FGN

These all inherit from *IFilterBase* and do not implement shared parameter pointers as they are used as embedded sub-filter objects. *IFilterBase* requires the implementation of only four functions. The filters include outputs for all types in Table 12.1 and are written into a filter output data structure.

The *VA1Filter* and *VADiodeSubFilter* objects are used as sub-filters in the Korg, Moog, and diode filters, which are always synchronously tuned with identical α coefficients that require the *tan* function call but in general will have different β coefficients. Therefore, these filters include added functions to allow the container filter to set the coefficients directly, bypassing the costly *update* function for improved efficiency. Additionally, these filters need to provide access to their *s*(*n*) values for the container object to manipulate in its feedback loop. The *VA1Filter* includes these additional functions; note that *beta* is applied during the *getFBOutput* method for the Korg35 and Moog filters.

```
// --- set coeffs directly, bypassing coeff calculation
void setBeta(double _beta);
void setAlpha(double _alpha);

// --- added for MOOG & K35, need access to this output value
double getFBOutput();
```

The *VADiodeSubFilter* requires many more coefficients that are packaged in the *DiodeVA1Coefficients* structure. It also requires access into and out of the feedback portion, so it includes these additional methods; compare the code with Figure 12.12(a).

```
void setFilterCoeffs(const DiodeVA1Coeffs& _coeffs);

void setFBInput(double _feedbackIn) { feedbackIn = _feedbackIn; }

double getFBOutput() {
     return coeffs.beta * (sn + feedbackIn*coeffs.delta); }
```

Table 12.1 shows the second order and higher VA filters, and their required *VA1Filter* sub-filters, which are stored in small arrays on the container object. The filter indexes are named according to the model, block diagram, and equations, so you can easily link the parts for study.

12.10.1 Sub-Filter Operation

The coefficient calculation starts with a function call to *setFilterParams*, which checks the incoming values and, if different, saves them and calls *update*. For filters with a *Q* control, you need to

Table 12.1 High order VA filters and their sub-filter types and arrays

Filter	VA1Filter Sub-filter Types	Array Name
VASVFilter	none, see model	n/a
VAKorg35 LPF	Filter1: LPF1 Filter2: LPF1 Filter3: HPF1	*lpfVAFilters[KORG_SUBFILTERS]*
VAKorg35 HPF	Filter1: HPF1 Filter2: HPF1 Filter3: LPF1	*hpfVAFilters[KORG_SUBFILTERS]*
VAMoogFilter	Filter1 = Filter2 = Filter3 = Filter4: LPF1	*subFilter[MOOG_SUBFILTERS];*
VADiodeFilter	Filter1 = Filter2 = Filter3 = Filter4: LPF1	*subFilter[DIODE_SUBFILTERS];*

map the incoming value on the range [1, 10] to the range of K values such that $Q = 10$ produces self-oscillation. This is accomplished with the *mapDoubleValue* function. For example, in the Korg35 filter, the K value goes from 0.01 to 2.0 as Q is increased from 1.0 to 10.0.

```
void VAKorg35Filter::setFilterParams(double _fc, double _Q)
{
     // --- use mapping function for Q -> K
     mapDoubleValue(_Q, 1.0, 10.0, 0.01, 2.0);

     if (fc != _fc || K != _Q)
     {
          fc = _fc;
          K = _Q;
          update();
     }
}
```

The *update* function performs the coefficient calculations and is based directly on the filter coefficient equations presented in the chapter, using the same variable names whenever possible. For example, the *VAKorg35Filter's update* function calculates Equation block (12.13) as follows; compare the code with the equation block for this filter, then do the same for the remaining filters so you understand how the calculation works.

```
bool VAKorg35Filter::update()
{
     double g = tan(kTwoPi*fc*halfSamplePeriod);
     alpha = g / (1.0 + g);

     // --- alpha0 same for LPF, HPF
     alpha0 = 1.0 / (1.0 - K*alpha + K*alpha*alpha);

     // --- three sync-tuned filters
     for (uint32_t i = 0; i < KORG_SUBFILTERS; i++)
     {
          lpfVAFilters[i].setAlpha(alpha);
          hpfVAFilters[i].setAlpha(alpha);
     }

     // --- set filter beta values FLT2:LPF FLT3:HPF
     lpfVAFilters[FLT2].setBeta((K * (1.0 - alpha)) / (1.0 + g));
     lpfVAFilters[FLT3].setBeta(-1.0 / (1.0 + g));

     // --- set filter beta values FLT2:HPF FLT3:LPF
     hpfVAFilters[FLT2].setBeta(-alpha / (1.0 + g));
     hpfVAFilters[FLT3].setBeta(1.0 / (1.0 + g));

     return true;
}
```

The *process* function runs the filter algorithm on a single audio input sample and varies greatly with the model. Each filter declares a *FilterOutput* structure named *output* that may hold multiple filter output values in an array of *doubles*:

```
double filter[NUM_FILTER_OUTPUTS];
```

Each slot in the array holds a different filter output and a helper enumeration is used for easier code readability:

```
enum { LPF1, LPF2, LPF3, LPF4, HPF1, HPF2, HPF3…, NUM_FILTER_OUTPUTS};
```

The *VAMoogFilter*::*process* method is shown here for the normal filter.

- Notice that the *getFBOutput* function is used on the sub-filters
- The bass gain compensation is hard coded with *bassComp* = 0 so that the filter operates in a normal manner
- The input value *u* is calculated with the α_0 loop correction factor (this is the same for all filters)
- Each filter processes into a sub-structure, whose outputs are gathered accordingly
- The *output* structure's member variables are written and will vary with the filter model; note the use of the helper enumeration
- The analog FGN code simply pumps the input *u* into a series of LPF1 modules, taking their analog FGN outputs while preserving the normal feedback outputs for the resonant loop and following the normal pattern; check the sample code for details.

```
FilterOutput VAMoogFilter::process(double xn)
{
     // --- 4th order MOOG:
     double sigma = 0.0;

     // --- feedback
     for (uint32_t i = 0; i < MOOG_SUBFILTERS; i++)
          sigma += subFilter[i].getFBOutput();

     // --- gain comp
     xn *= 1.0 + bassComp*K; // --- bassComp is hard coded

     // --- now figure out u(n) = alpha0*[x(n) - K*sigma]
     double u = alpha0*(xn - K * sigma);

     FilterOutput subFltOut[4];
     FilterOutput subFltOutFGN[4];
```

```
    // --- send u -> LPF1
    subFltOut[FLT1] = subFilter[FLT1].process(u);

    // --- and then cascade the outputs to form y(n)
    subFltOut[FLT2] =
         subFilter[FLT2].process(subFltOut[FLT1].filter[LPF1]);
    subFltOut[FLT3] =
         subFilter[FLT3].process(subFltOut[FLT2].filter[LPF1]);
    subFltOut[FLT4] =
         subFilter[FLT4].process(subFltOut[FLT3].filter[LPF1]);

    // --- optional outputs 1,2,3
    output.filter[LPF1] = subFltOut[FLT1].filter[LPF1];
    output.filter[LPF2] = subFltOut[FLT2].filter[LPF1];
    output.filter[LPF3] = subFltOut[FLT3].filter[LPF1];

    // --- MOOG LP4 output
    output.filter[LPF4] = subFltOut[FLT4].filter[LPF1];

    return output;
}
```

Now, you need to compare the rest of the object code to the block diagrams and coefficient equations to prove to yourself that the code matches the model. Each filter can produce the analog FGN output, whose implementation varies from simple to complex depending on the filter type. Table 12.2 lists the filters and their sub-filter array components.

12.11 Input Drive and Output Peak Limiter

The SVF, Korg35, Moog, and diode models include an input distortion control that sets the saturation value in a *tanh* waveshaper, as documented in my FX plugin book. When the saturation is at

Table 12.2 Synth filter models and their output types, along with notes on implementation

Synth Filter Model	Outputs	Notes
VA1Filter	LPF1, HPF1, APF1, LPF1_FGN	Uses Zavalishin's first order structure
VASVFilter	LPF2, HPF2, BPF2, BSF2, LPF2_FGN	Self-oscillation induced at max Q by letting Q = infinity so $\rho = \alpha$
VAKorg35Filter	LPF2, HPF2, LPF2_FGN	Massive gain near self-oscillation
VAMoogFilter	LPF1, LPF2, LPF3, LPF4, and analog FGN versions of each	Analog FGN version requires four more sub-filters but are sync-tuned
VA1DiodeSubFilter	LPF1, LPF1_FGN	Uses coefficient structure for extended variables
VADiodeFilter	LPF4, LPF4_FGN	Requires a gain factor for analog FGN output that is empirically derived

the minimum value, the waveshaper is disabled. Placing the nonlinear distortion at the filter input allows the filter to sculpt the added harmonics. Extreme settings will cause aliasing, which will require mitigation via oversampling, which is also covered in the FX plugin book. The output of each model passes through a peak limiter using logarithmic RMS detection with the threshold set at almost 0 dBFS, which produces pure sinusoids during self-oscillation and prevents distortion from high gain due to high *Q* settings. The limiter is a simplified version of the limiter in the second edition FX book and works extremely well. The coding details can be found in the *VAFilter* and *VAFCore* objects as they implement these blocks and not the filters themselves.

12.12 *SynthFilter* and Cores

The *SynthFilter* and filter core details are shown in block form in Figure 12.16. The custom module strings are the filter-type strings, and the mod knobs are chosen for the most common operations, including key tracking, drive, and modulation intensity controls, one each for EG and bipolar modulation inputs. Both cores are available as separate dynamic modules and work easily in stand-alone mode, so you can use them in your non-synth projects as well. Table 12.3 lists the GUI parameter structure and core descriptions.

The VA filter core produces all of the VA filters in this chapter. For easier programming, a secondary filter enumeration is used that groups the filters according to VA model:

```
enum class FilterModel { kFirstOrder, kSVF, kKorg35, kMoog, kDiode};
```

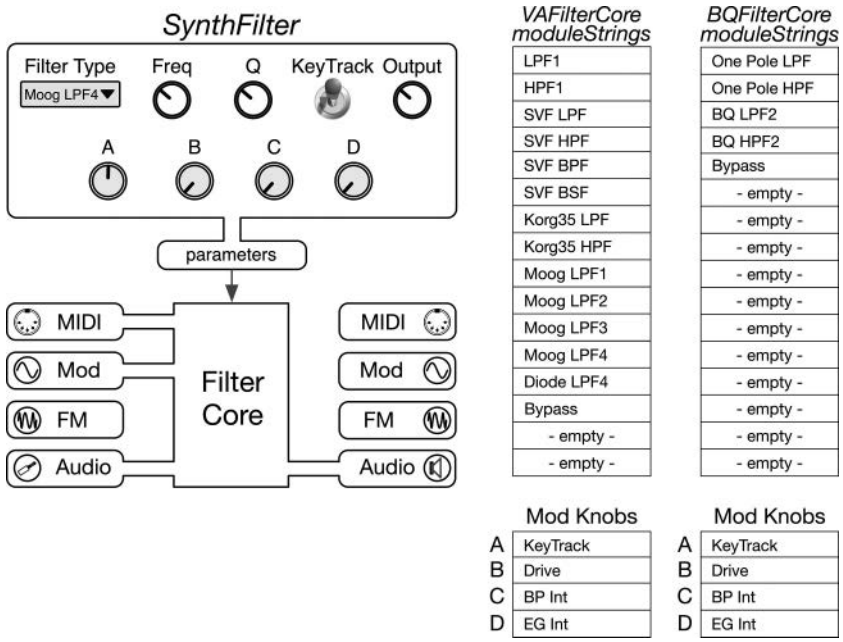


Figure 12.16 The *SynthFilter* and core block diagram, filter types, and mod knob labels

Table 12.3 SynthFilter custom parameter structure and cores: (*) downloadable SynthLab-DM core project

Parameter Structure	Description
FilterParameters	Used for all filters; includes additional parameters, such as bass compensation, that only apply to some filters
Example Core	Description
*VAFilterCore**	13 VA filter algorithms including the first, second, and third order Moog variants
*BQFilterCore**	A few biquad filters to get you started; use my FX plugin book for several dozen more, including impulse invariant, the MMA FGN lowpass, and multiple analog Nyquist gain matching algorithms

Table 12.4 The *SynthFilter* core GUI controls

Control	Function	Target
Filter type	Selects 1 of 13 filter types	Filter selection
Filter *f*	Sets filter center or cutoff frequency	Filter frequency
Filter *Q*	Set generic *Q* value on range [1, 10]	Filter *Q*
Key track	Enable/Disable key track modulation	Filter frequency
Output	Level in dB	Output amplitude
Mod Knob		
KT amount	Key tracking offset in semitones +/− 48 range	Filter frequency
Drive	Sets waveshaper saturation	Input amplitude
EG Int	Sets intensity of EG modulation	Filter frequency
BP Int	Sets intensity of summed bipolar modulation	Filter frequency

12.12.1 Synth Filter Controls, Modulations and Features

The synth filters share the same four controls: frequency (cut-off, center, or resonant), *Q*, a filter-keytrack enable switch, and an output control (dB). Table 12.2 lists the GUI controls and their parameter targets within the plugin. The VA filter core also includes a stereo peak limiter to produce sweet and pure sinusoidal output when driven into self-oscillation. The filter drive control adds pre-filter distortion using *tanh* waveshapers. The limiter's threshold is set to −0.5 dBFS. The filters include three modulation calculations: key-track with offset, bipolar, and unipolar (EG) modulation.

12.13 Synth Filter Core Programming Notes

With the C++ files open, compare the programming notes with the code that you see, starting with the class descriptions of each core. Sections 12.13.1 through 12.13.5 summarize the five operational phases, plus constructor for the synth filter cores. Make sure to examine the code while digesting their operational phase details.

12.13.1 Construction Phase

All cores:

- The module strings are the filter types; use whatever strings make the most sense to your users
- The mod knobs are all assigned
- The filter cores are simple in that there are no lookup tables or databases to access

12.13.2 Reset Phase

The *reset* function simply resets all of the components and sets up the limiter with a threshold of −0.5 dBFS.

```
va1.reset(processInfo.sampleRate);
svf.reset(processInfo.sampleRate);
etc…

// --- output limiter
limiter.reset(processInfo.sampleRate);
limiter.setThreshold _ dB(-0.5);
```

12.13.3 Note On Phase

The note-on event is easily serviced; you save the incoming MIDI pitch to use for key tracking, and that is it.

12.13.4 Update Phase

The update operation follows the normal design pattern for pitch modulation, though here, it is for the filter f_c value that is based off of semitones and calculated the same way as f_o (pitch) modulation for oscillators. Filters are often hard-wired to a separate EG, with a dedicated EG intensity control; the same is true for the SynthLab synths that feature a dedicated filter EG object. However, a bipolar modulation input may also be applied and a bipolar intensity control used. To set or modify the range of operation, you only need to modify a bit of code in the *filtercore.h* file. Here, you use the *semitonesBetweenFrequencies* helper function to set the modulation range in semitones. The high limit is the top of the filter range, but you may decide to lower it to around 18 kHz – listen and decide.

```
freqModLow = 20.0; // <- you will want to raise this value a bit
freqModHigh = 20480.0; // <- you will want to lower this value a bit
freqModSemitoneRange = semitonesBetweenFrequencies(
                                                freqModLow, freqModHigh);
```

In the *update* function, first parse the mod knob values that set the EG and bipolar intensities, then calculate the modulation. Notice that the bipolar calculation adds a 0.5 modifier to split the modulation range around the user's f_c control setting.

```
// --- bipolar freqmod (0.5 is to split the total range)
```

```
bpInt = getModKnobValueLinear(parameters->modKnobValue[MOD_KNOB_C],
                              0.0, +1.0);

bpFmodSemitones = bpInt*0.5*freqModSemitoneRange*
                        processInfo.modulationInputs[kBipolarMod];

// --- EG freqmod
egInt = getModKnobValueLinear(parameters->modKnobValue[MOD_KNOB_D],
                              0.0, +1.0);
egFmodSemitones = egInt*freqModSemitoneRange*
                        processInfo.modulationInputs[kEGMod];
```

The key-tracking code checks to see if the function is enabled, then overrides the user's f_c control setting and calculates the key track frequency shift on a range of [−48, +48] semitones (+/− four octaves).

```
// --- setup fc mod
double fc = parameters->fc;
double ktFmodSemotones = 0.0;

// --- key tracking
if (parameters->enableKeyTrack)
{
     ktFmodSemotones = getModKnobValueLinear(
          parameters->modKnobValue[FLT_KEYTRACK], -48.0, +48.0);
     fc = midiPitch;
}
```

The modulation calculation sums the semitone values as usual and bounds the value:

```
// --- sum modulations
fcModSSemis = bpFmodSemitones + egFmodSemitones + ktFmodSemotones;

// --- multiply by pitch shift factor
fc *= pow(2.0, fcModSSemis / 12.0);
boundValue(fc, freqModLow, freqModHigh);
```

The last chunk of code is a decision tree that sets the selection members, then calls the update function on the sub-object. The VA filters are selected by family:

```
else if (parameters->filterAlgorithm == VAFilterAlgorithm::kDiode_LP4)
{
     // --- output array slot
     if (parameters->analogFGN)
          outputIndex = ANM_LPF4;
     else
          outputIndex = LPF4;

     // --- model
```

```
      selectedModel = FilterModel::kDiode;

      // --- sub-filter update
      diode.setFilterParams(fc, parameters->Q);
}
```

The biquad filters calculate the filter coefficients and store them:

```
if (parameters->filterIndex == enumToInt(BQFilterAlgorithm::kLPF2))
{
      // --- see 2nd Ed FX for formulae
      double theta_c = kTwoPi*fc / sampleRate;
      double d = 1.0 / mappedQ;
      double betaNumerator = 1.0 - ((d / 2.0)*(sin(theta_c)));
      double betaDenominator = 1.0 + ((d / 2.0)*(sin(theta_c)));
      double beta = 0.5*(betaNumerator / betaDenominator);
      double gamma = (0.5 + beta)*(cos(theta_c));
      double alpha = (0.5 + beta - gamma) / 2.0;

      // --- update coeffs
      bq.coeff[a0] = alpha;
      bq.coeff[a1] = 2.0*alpha;
      bq.coeff[a2] = alpha;
      bq.coeff[b1] = -2.0*gamma;
      bq.coeff[b2] = 2.0*beta;
}
```

12.13.5 Render Phase

The VA core render phase is simple since the sub-filters do all of the heavy lifting. Therefore, the render function simply uses the filter type and output index to perform the processing. The drive control is used to apply a nonlinear waveshaper only if the control value is greater than zero. Then, the selected filter's process function runs the filter, and the output is applied to the peak limiter and scaled with the user's output control value that was calculated in the *update* function. Notice that the VA core's sub-filters calculate all outputs (LPF, HPF, BPF, etc.), which are then selected, and all filters and sub-filters are stereo, requiring double the objects. Examine the code in *vafilters.h* and *.cpp* to see how I minimize the filter calculations using the *IFilterBase* interface and functions that set the filtering coefficients directly, bypassing the update calculation. For the Korg35, Moog, and diode filters, the sub-filters are synchronously tuned so a single update function's results can be shared across multiple sub-filter components. The synth filter processes audio input buffers into output buffers.

```
// --- stereo I/O
float* leftInBuffer = processInfo.inputBuffers[LEFT_CHANNEL];
float* leftOutBuffer = processInfo.outputBuffers[LEFT_CHANNEL];

float* rightInBuffer = processInfo.inputBuffers[RIGHT_CHANNEL];
float* rightOutBuffer = processInfo.outputBuffers[RIGHT_CHANNEL];
```

At the top of the block-processing loop, notice the filter bypass code:

```
for (uint32_t i = 0; i < processInfo.samplesToProcess; i++)
{
     double xnL = leftInBuffer[i];
     double xnR = rightInBuffer[i];

     if (parameters->filterIndex ==
                    enumToInt(VAFilterAlgorithm::kBypassFilter))
     {
          leftOutBuffer[i] = xnL;
          rightOutBuffer[i] = xnR;
          continue;
     }
```

Filter drive/distortion uses a built-in waveshaper function (note that there are several to choose from):

```
// --- waveshaper drive
if (parameters->filterDrive > 1.05)
{
     xnL = tanhWaveShaper(xnL, parameters->filterDrive);
     xnR = tanhWaveShaper(xnR, parameters->filterDrive);
}
```

The VA filters are processed according to filter family; notice the array of two filters for each family:

```
FilterOutput output[STEREO];


if (selectedModel == FilterModel::kFirstOrder)
{
     output[LEFT] = va1[LEFT].process(xnL);
     output[RIGHT] = va1[RIGHT].process(xnR);
}
else if (selectedModel == FilterModel::kSVF)
{
     output[LEFT] = svf[LEFT].process(xnL);
     output[RIGHT] = svf[RIGHT].process(xnR);
}
else if (selectedModel == FilterModel::kKorg35)
{
     output[LEFT] = korg35[LEFT].process(xnL);
     output[RIGHT] = korg35[RIGHT].process(xnR);
}

etc. . .
```

The biquad core uses a single biquad filtering object, *BQAudioFilter*, that implements the direct form biquad filter, described in detail in my FX plugin book. That object performs the filtering in a simple function call:

```
// --- select output
leftOutBuffer[i] = outputAmp *
                        filter[LEFT_CHANNEL].processAudioSample(xnL);
rightOutBuffer[i] = outputAmp *
                        filter[RIGHT_CHANNEL].processAudioSample(xnR);
```

12.14 Exercises

12.14.1 SynthLab-DM Core: SVF Self Modulation

The state variable filter will only self-oscillate when the Q value becomes infinite and is the only model that does not have an upper feedback value limit that creates self-oscillation. The SVF has the Q control from [1, 10] mapped to a range of [0.707, 25.0], and at the top value, the filter is highly resonant, producing copious gain at the peak frequency, though it is not oscillating. Examine the equations in block (12.12) and figure out what happens to the ρ value if Q becomes infinite. Then, modify the *VASVFilter::update* function with an if/else statement, and add your own code. Now, when the user maxes the Q control, the filter will go into self-oscillation. The switching point will be inaudible as the filter will already have massive gain, and the output limiter will be maintaining distortion-free operation.

```
if (Q > 24.95)
     // --- your code here
else
     rho = 2.0*R + alpha; // normal calculation
```

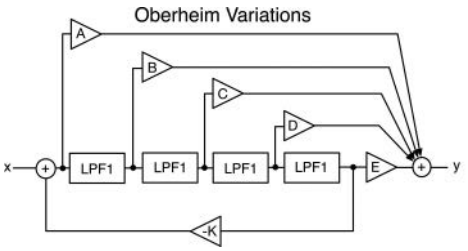
The bipolar modulation intensity control (Mod Knob #4) can be removed because bipolar modulators like LFOs already have output amplitude controls that will set the modulation intensity. It can also be set with the Modulation Matrix, so this Mod Knob can be changed to give you more practice. Here are two possibilities for altering this control.

12.14.2 SynthLab-DM Core: SVF Bass Compensation

The code and parameter structure already include a *bassComp* variable that represents the a variable in Figures 12.11 and 12.12 for the Moog and diode filters. This value varies on the range [0.0, +1.0]. Modify the Mod Knob #4 string for "Bass Comp," then modify the *update* calculation to remove the bipolar mod intensity calculation (or set it to 1.0), and modify the *render* function to apply the bass compensation. As the user increases the bass compensation control, the filter gain reduction will be abated, and more bass will result. The output limiter will keep the distortion at bay.

12.14.3 SynthLab-DM Core: Key Tracking Slope

The key-tracking amount is a constant shift in semitones above or below the MIDI pitch of the note being played. Check out the manual for the Korg Wavestate™ from www.willpirkle.com/



Filter	A	B	C	D	E
2nd order BPF	0	2	-2	0	0
4th order BPF	0	0	4	-8	4
2nd order HPF	1	-2	1	0	0
4th order HPF	1	-4	6	-4	1

Figure 12.17 The Oberheim variations on the Moog ladder signal flow block diagram and table of coefficient values

Downloads/wavestate.pdf, and look at the filter key-tracking section. Notice how the added *slope* control is used to create a rate of change in the key-tracking offset as the user moves above or below the middle-note (usually middle-C or MIDI note 60). Next, change Mod Knob #4 to a key-tracking *slope* control, and use the Wavestate manual and information to create your new feature.

12.14.4 Advanced Module: Oberheim/Moog FilterCore

Finally, make your own filter core object which implements the Oberheim variations on the Moog ladder filter. Create your own Moog filter C++ object, and add it to the objects in *vafilters.h* and *vafilters.cpp*, which implements the added filters from Oberheim's Xpander® synth. Tom Oberheim realized that, if outputs were taken from each of the Moog's four LPFs and combined in various ratios, many other filters could be obtained. You can download the original XPander service manual from www.willpirkle.com/Downloads/Oberheim_Xpander_Service_Manual.pdf. The following are some of the variations available (other than the normal LPFs), and the service manual contains more options.

- second order BPF
- fourth order BPF
- second order HPF
- fourth order HPF

Figure 12.17 shows the newly added feed-forward coefficients that inject zeros into the transfer function and a table that lists the coefficient values for the four new filters. Create your own new synth filter core object that implements these four new filters, and add it to the SynthLab projects that you like, thereby increasing the number of filters to 18 in total.

Bibliography

cedos.com. 1980. "CEM3328 Four Pole Low-Pass VCF Datasheet." http://www.cedos.com/datasheets/cem3328pdf.pdf, Accessed October 14, 2020

El-Masry, E.I. and Sakla, A.A. 1979. "Low-sensitivity Digital Ladder Networks." *13th Asilo-mar Conference*, pp. 273–278.

Fettweis, Alfred. 1986. "Wave Digital Filters: Theory and Practice." *Proceedings of the IEEE*, vol. 74, no. 2. pp. 270–327.

Fontana, Fredrico. 2010. "Modeling of the EMS VCS3 Voltage-Controlled Filter as a Nonlinear Filter Network." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4. pp. 760–772.

Huovilainen, Antti. 2006. "Nonlinear Digital Implementation of the Moog Ladder Filter." Proceedings of the International Conference on Digital Audio Effects, Naples.

Lindquist, Claude. 1977. *Active Network Design with Signal Filtering Applications*. Long Beach: Steward and Sons.

Nagahama, Yasuo. 1977. "Voltage Controlled Filter." United States Patent 4,039,980.

Oberheim, Tom. 1984. *Oberheim XPander Service Manual*. Los Angeles: ECC Development Corp.

Pirkle, Will. 2013. "Modeling the Korg35 Highpass and Lowpass Filters." Presented at the 135th Audio Engineering Society Convention, New York.

Pirkle, Will. 2014. "Novel Hybrid Virtual Analog Filters Based on the Sallen-Key Architecture." Presented at the 137th Audio Engineering Society Convention, New York.

Pirkle, Will. 2014. "Resolving Delay-Free Loops in Recursive Filters using the Modified Härmä Method." Presented at the 137th Audio Engineering Society Convention, New York.

Pirkle, Will. 2018. *Designing Audio Effects Plugins in C++*, 2nd Ed. Chap. 12. New York: Routledge.

Stilson, Tim & Smith, Julius O. 1996. "Analyzing the Moog VCF with Considerations for Digital Implementation." Proceedings of the 1996 International Computer Music Conference, San Francisco.

Stinchcombe, Tim. 2006. "A Study of the Korg MS10 and MS20 Filters." http://www.timstinchcombe.co.uk/synth/MS20_study.pdf, Accessed October 14, 2020

synthfool.com. 1974. "Oberheim SEM Schematics." http://www.synthfool.com/docs/Oberheim/Oberheim_SEM1A/Oberheim_SEM_1A_Schematics.pdf, Accessed October 14, 2020

Välimäki, Vesa & Huovilainen, Antti. 2006. "Oscillator and Filter Algorithms for Virtual Analog Synthesis." *Computer Music Journal*, vol. 30, no. 2, pp. 19–31, Cambridge: MIT Press.

Zavalishin, Vadim. 2012. *The Art of VA Filter Design*. https://www.native-instruments.com/fileadmin/ni_media/downloads/pdf/VAFilterDesign_2.1.0.pdf, Accessed October 14, 2020

13 Karplus-Strong Plucked String Model

In its simplest form, physical modeling involves creating DSP components that combine to behave like their physical counterparts, such as a pendulum or a resonating mass-spring system. The theory may be used for modeling strings, tubes, or struck-metal plates, or practically any other acoustic/mechanical system or effect, such as a piano or the reverb effect. The Yamaha VL-1®, Korg Prophecy®, and OASYS® are examples of early, commercially available physical modeling synthesizers. Today, the GeoShred® mobile app uses physical string modeling as its synthesis core. This area has matured over several decades, including texts dedicated specifically to the science. In this chapter, we will study one of the most famous and useful physical modeling algorithms: the Karplus-Strong plucked string model. This algorithm works well as an introduction to physical modeling because it includes the same set of components as other more complex algorithms. such as a saxophone. Physical modeling does away with PCM samples and wavetables, so there is little data to store or load. However, the processing may become very complicated as the output is mathematically generated.

13.1 The Exciter-Resonator

Physical modeling algorithms for acoustic instruments usually contain two components: the exciter and the resonator, which are fundamentally connected, as shown in Figure 13.1(a). The exciter stimulates the resonator that rings at the desired musical pitch. Figure 13.1(b) includes an output filter that couples the output waveform to the air load. Figure 13.1(b) also shows that the

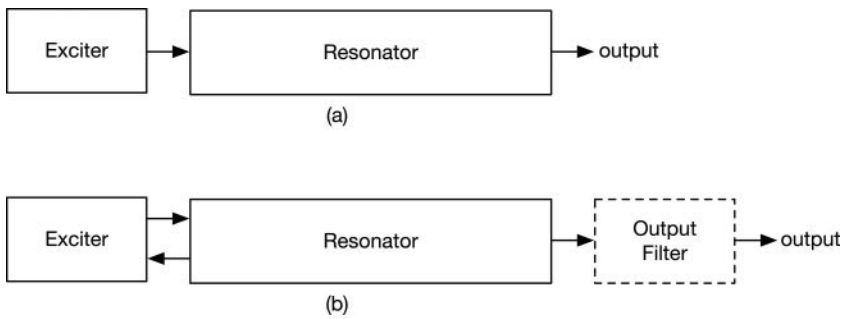


Figure 13.1 (a) The most basic exciter-resonator system, and (b) a more complex model that includes coupling from the resonator back to the exciter and an output impedance coupler

Table 13.1 Some common instruments and their physical modeling components

Instrument	*Exciter*	*Resonator*	*Output Filter*
Guitar	Plucked-string deformation	Transverse wave on string	Guitar bridge to body (integrator)
Saxophone	Reed flapping against mouthpiece and breath pressure	Standing wave in tube	Bell and open holes in bore (HPF)
Trumpet	Lips vibrating in mouthpiece and breath pressure	Standing wave in tube	Bell (HPF)

resonator can couple back to the exciter with the reverse arrow. Table 13.1 lists a few common acoustic instruments and their exciter, resonator, and output filter. The saxophone and trumpet both feature coupled resonators because the standing wave setup in the instrument's tube reflects off of the musician's lips/mouth. Instruments that include open holes, such as most woodwinds, have an additional complexity since there are multiple locations where the sound couples with the air load.

13.2 The Plucked String

Plucked string instruments include guitars and basses as well as harps and other similar devices. Struck string instruments include piano, clavichord, and dulcimer. In both cases, the string is deformed and released either with the finger or a hammer, as depicted in Figure 13.2(a) (the guitar) and (b) (the deformation), and acts as the excitation signal. In its simplest version, the deformation sets up a transverse wave that bounces back and forth between the bridge and the fretting finger, or nut if the string is open, as shown in Figure 13.2(c) and (d). The transverse wave bounces off of each terminal end point at the nut (or fretting finger) and the bridge. The rate of the bounces per second is the frequency in Hertz of the note. The transverse wave on the string itself is the resonator, and the bridge couples the resonator with the body that vibrates and couples the output to the air load. As the system loses energy, the transverse wave loses amplitude and eventually becomes inaudible. In addition, we observe the fact that high frequencies, which inherently do not have as much power as low frequencies, decay faster and earlier. The body behaves as a complex, multi-peaked filter that resonates well at some frequencies and not so well at others.

13.3 The Karplus-Strong Model

The Karplus-Strong (KS) model for the plucked string is simple, elegant, and effective. Its most basic form is immediately simpler than other algorithms because there is no interaction between the resonator and the exciter – once the string is plucked, the exciter is removed from the model, and the resonator takes over. Note that there are more complex models that do include coupling to the exciter. DSP building blocks are designed to implement each section of the model. Figure 13.3(a) shows the basic idea for modeling the string.

13.3.1 KS Resonator

The resonator consists of two delay lines, each of which implements half of the total delay, D, that is the period of the musical pitch. An LPF and attenuator at each delay output remove high

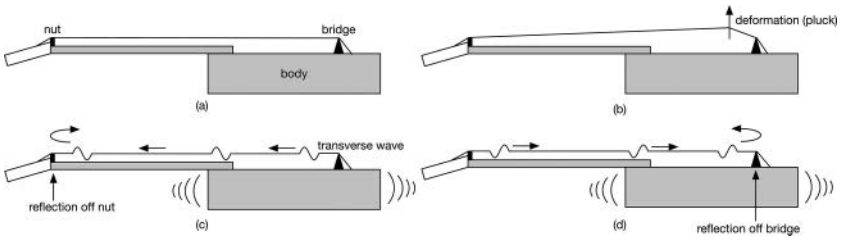


Figure 13.2 (a) An acoustic guitar with nut, bridge, and body; (b) plucking the string deforms it and (c) sets up a transverse wave that moves down the string, and (d) bounces off the nut and moves back toward the bridge

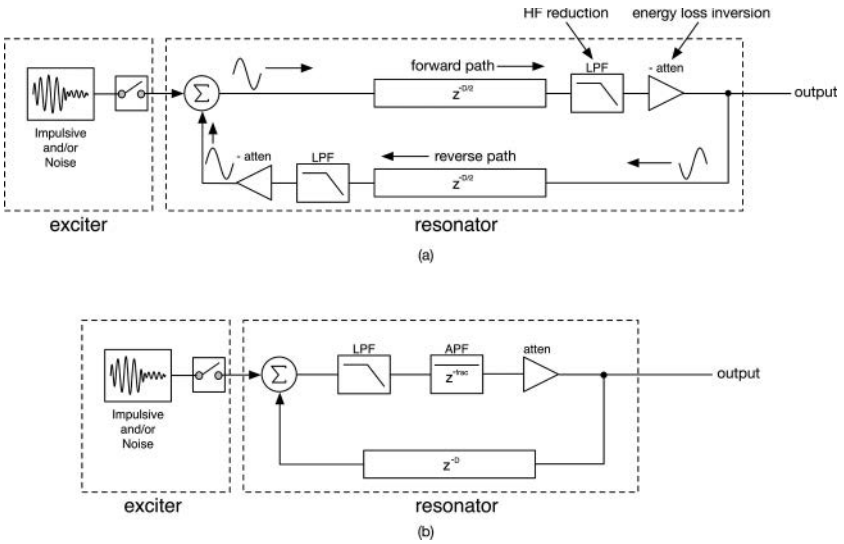


Figure 13.3 (a) Basic model of a plucked string with dual delay lines for transverse wave oscillation and (b) a simplified version that combines like components and adds the fractional delay APF

frequencies and energy on each reflection. Figure 13.3(b) shows how this loop may be simplified and improved. The two delay lines may be combined, along with the attenuators and filters to reduce complexity.

Musical pitches will rarely be exact multiples of the sample period. This means that the period of the waveform, and the calculated delay, D, will include a fractional component, as shown in Figure 13.3(b). The fractional delay is implemented with a first order all pass filter (APF) in Figure 13.4(a) that creates a linear fractional group delay, δ over the range of 0 Hz to about $0.1f_s$, as shown in Figure 13.4(b). With f_s = 44.1 kHz, this would be around 4,410 Hz, which is close to the top note on a piano, so it is easily usable across the range of a guitar or bass. The resonator's delay line length and APF fractional delay calculations must be exact to produce the correct pitch. The user may have control over the attenuation that corresponds to the string's decay time.

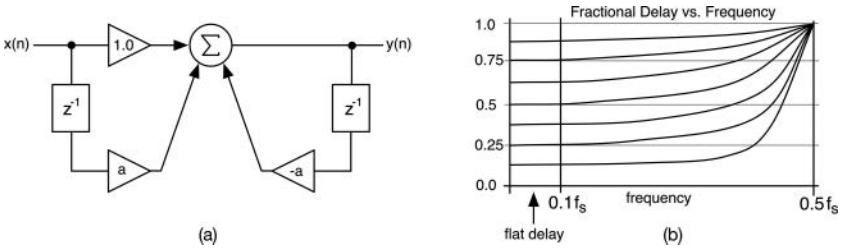


Figure 13.4 (a) A first order structure for an APF with only one coefficient a to calculate and (b) the group delay versus frequency for the APF; each curve represents a different APF f_c

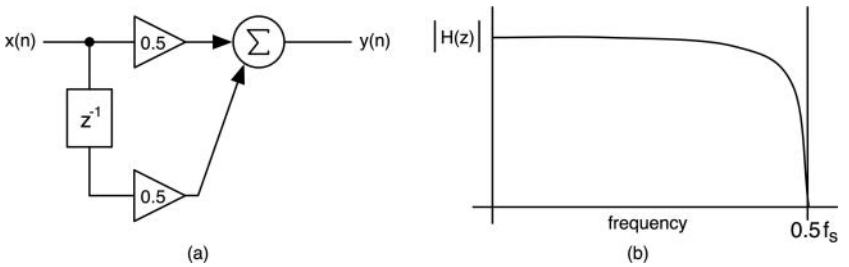


Figure 13.5 (a) The simple first order feed-forward LPF and (b) its frequency response showing a zero at Nyquist

13.3.2 KS Loop Filter

The lowpass filter in the loop is going to reduce high frequency components, but it can also introduce its own group delay, depending on the filter, f_c. This means that we cannot let the user adjust the filter, or the note will be slightly out of tune. However, if we use the simple first order feed-forward LPF, shown in Figure 13.5(a), with the coefficients $a_0 = a_1 = 0.5$, then we get the most selectivity with a zero at Nyquist, as shown in Figure 13.5(b), and we can calculate that we will get exactly ½ of a sample of delay through the filter.

13.3.3 KS Exciter

The exciter signal has numerous options: a simple unit impulse, an impulse generated from a resonant filter, and impulses that are combinations of several impulsive or noisy components. A simple and effective excitation signal consists of a short noise blast with an envelope and a spectral

> The Karplus-Strong algorithm is a specific kind of plucked string model and has undergone improvements and modifications over time. The algorithms here are a combination of some of these modifications. For simplicity's sake, I am keeping the Karplus-Strong moniker for the model developed here.

content. In the traditional KS algorithm, the exciter signal is placed inside of the delay line and sets the initial state for the system.

For the *KSOscillator* core, the exciter is placed at the entrance to the loop summer, producing the first output samples immediately and without delay. Setting the spectrum of the noise exciter establishes the initial frequency response of the system. The LPF loop filter will take that spectrum and filter it repeatedly on each pass through the loop, emulating the natural harmonic decay in an acoustic stringed instrument. Unlike the resonator, which has some fixed properties and calculations, the exciter is wide open for experimentation. The harmonic content of the output depends upon the exciter's initial harmonic spectrum. If noisy signals are used, then randomness in the spectra will create slight timbral differences for each note-event, which will add a human element.

13.3.3.1 Windowed Noise Burst Exciter

You can create interesting sounds with very simple exciter signals, including the rectangular-pulsed random noise that you generate by simply counting off a few random values, as shown in Figure 13.6(a). You can also use common window functions to create a time domain signal without the discontinuities at the edges that will reduce the high harmonics, as in the Hann windowed excitation in Figure 13.6(b). The windowed exciter lends itself easily to re-triggering, which produces a stream of impulsive blasts called "sonic grains," as shown in Figure 13.6(c). Each of these may also operate on pseudo-random noise or filter noise signals.

13.3.3.2 EG Shaped Noise Burst Exciter

You can also shape the noise burst with an envelope generator (EG) using the attack-release (AR) and attack-hold-release (AHR) contours, as shown in Figure 13.6(d) and (e). As with the windowed exciter, the EG may also be retriggered. The physical model of a bowed instrument like the cello is quite complex, but the AHR shaped noise produces a surprisingly similar sound when the attack time is long. When the attack and release are set to 0.0, and the hold time is finite, we get the rectangular noise blast.

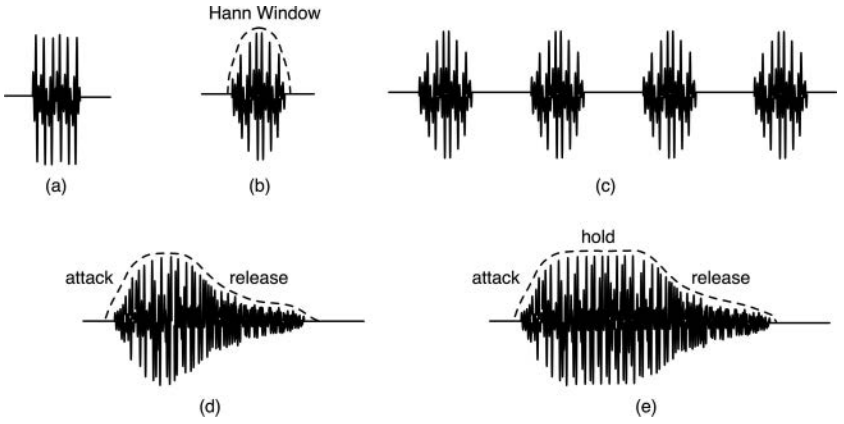


Figure 13.6 (a) Rectangular windowed noise burst; (b) Hann windowed noise burst; (c) sonic grains from re-triggered windowed exciter; (d) attack-release; and (e) attack-hold-release envelope noise bursts

13.3.3.3 Exciter by Reverse Filtering with PCM Samples

In yet another approach to generate an impulse, you start with a single PCM sample of an actual plucked string note-event. Next, you synthesize a perfectly complementary KS model that reverses (complements) all of the DSP algorithm blocks and run the model in reverse-time. This model accepts an input at the output port that flows in the reverse direction, then you recover the impulse response as the "output" of the exciter. Now you have the exact impulse response needed for the model to produce an output identical to your PCM sample. If you are interested, see the Leary reference for more information.

13.4 Pluck Position

Figure 13.2 is oversimplified but effective in explaining the transverse wave's relationship to the string's frequency. The pluck actually creates string motion, which may be thought of as two pulses that travel in opposite directions (Fletcher/Rossing) and bounce back and forth, producing a set of vibrating modes that combine to produce phase cancellations at frequencies based on the pluck position. When plucked in the exact center (twelfth fret on a guitar), the two standing waves cancel at each even harmonic, starting with the second harmonic (the octave), then the fourth, sixth, and eighth, producing a distinctly hollow timbre. As the pluck position moves towards one end, the canceled harmonics appear at exact multiples of the pluck position, as a fraction of the string length. This is modeled with an inverse comb filter whose delay length $z^{-D/M}$ is set to the pluck position fraction D/M of the string loop delay, as shown in Figure 13.7(a). If the string is plucked at ½ the string length, then the comb delay is ½ the loop delay ($D/2$), while it is 1/4 the total delay for a pluck at 1/4 the string length ($D/4$). The $z^{-D/M}$ comb filter response is shown in Figure 13.7(b), with notches which are multiples of $M = 4$. The number of notches and whether or not a zero is present at DC and Nyquist depends on the even/odd nature of the comb filter delay length. The comb filter length will rarely be an exact integer, and we use a delay line that interpolates to create the fractional delay in that filter.

13.4.1 Bridge Filter

The bridge acts as an integrator as it couples the signal pulses to the soundboard top and is implemented with a lowpass filter whose f_c is lower than the lowest pitch to synthesize (Välimäki), as shown in Figure 13.7(c). The bridge filter and pluck position filter may be combined (Välimäki) into one unit, as shown in Figure 13.7(a), whose combined frequency response is shown in Figure 13.7(d).

Figure 13.8(a) shows the output of the KS model for the *KSOscillator* taken at the bridge with the bridge filter engaged and a fundamental frequency of 110 Hz. Figure 13.8(b) includes the pluck position filter with the pluck point at half the string length, showing that every other harmonic is missing. The harmonic amplitudes have variation in them because the source is a short noise burst that does not have a perfectly flat spectrum.

13.4.2 Pickup Filter

Implementing an electric or bass guitar pickup requires us to revert to the two-delay-line KS model in Figure 13.2(a); the pickup point is tapped from locations within the integer delay line, corresponding to different locations under the strings. The pickup itself behaves like a second

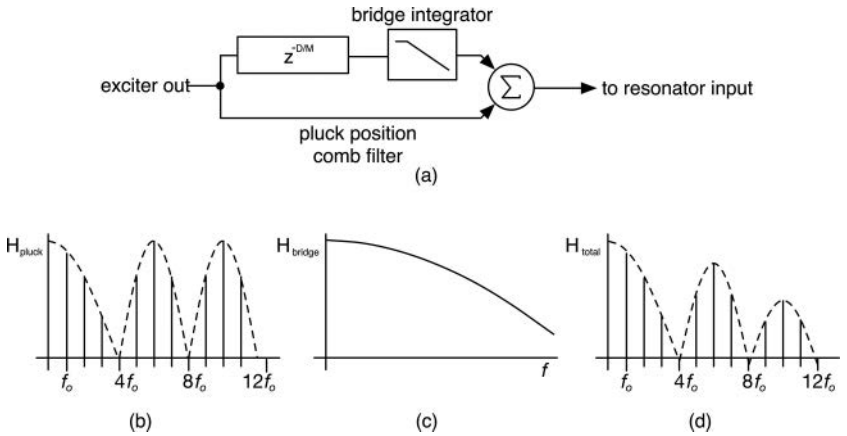


Figure 13.7 (a) The pluck position filter is a simple feed forward comb filter that is combined with the bridge integrator in the delay branch, (b) shows the comb filter response for a pluck position of ¼ the total length (M = 4), (c) the bridge filter's lowpass-integrator response and (d) the combined integrator and comb filter responses

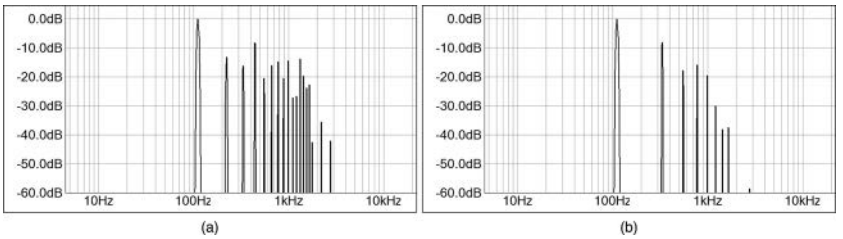


Figure 13.8 Spectra of the KS model of the note A-2 (110 Hz), showing output of (a) bridge filter only and (b) bridge filter and pluck position with pluck at ½ the length of the string

order resonant lowpass filter, with a resonant frequency between 2 kHz and 5 kHz, and a peak gain of a few *dB* for guitar. Bass pickups may exhibit much higher resonant peak frequencies. To keep the model simple, the single delay version is still implemented, with the pickup filter placed just before the bridge filter, as shown in Figure 13.8.

13.5 Karplus-Strong Algorithm Equations

For a given note-event, we will need to calculate several parameters for the resonator component that will set the delay length and calculate the APF coefficients for the fractional delay filter. We know that the delay loop will provide $L + ½$ samples when we include the loop filter. The loop filter has coefficients a_0 and a_1 fixed at 0.5, so there is nothing to calculate for it.

To calculate the fixed delay length L and APF coefficient a, we only need a few equations, and we can show these by example: suppose the user plays MIDI note 60 or middle-C with $f_o = 261.6256$ Hz.

Calculate the exact delay length L_E:

$$L_E = \frac{f_s}{f_o} = \frac{44{,}100}{261.6256...} = 168.5614 \tag{13.1}$$

We will get ½ sample of delay from the simple LPF, so subtract back, and find the nearest integer length that will result in a fractional delay of less than one sample:

$$L_D = \text{int}(L_E - 0.5) = \text{int}(168.0614) = 168 \tag{13.2}$$

L_D is the length of the delay in samples. Now, find the fractional amount that the APF must make up:

$$\delta = L_E - (L_D + 0.5) = 0.0614 \tag{13.3}$$

Another way to think about L_D and δ is that they are the integer and fractional components of L_E. With this fractional delay time, you can calculate the single APF coefficient a as:

$$\begin{aligned} \omega_0 &= \frac{2\pi f_o}{f_s} \\ a &= \frac{\sin((1-\delta)(\omega_0/2))}{\sin((1+\delta)(\omega_0/2))} \\ &= 0.8842 \end{aligned} \tag{13.4}$$

For the pluck position filter, the comb delay is set to L_E/M, as described in Section 13.4, where M is the denominator of the pluck position fraction. This requires a comb filter that interpolates to create fractional delay times. The bridge integrator is a first order, all pole lowpass filter with the f_c set to 20 Hz, which is about half of the bass guitar's low E-string frequency of about 44 Hz.

13.6 Karplus-Strong C++ Objects

Technically, the KS algorithm constitutes a complete patch and renders a plucked/struck event complete with decay time setting. However, the KS algorithm also works well as an oscillator used within the SynthLab architecture that includes LFOs, EGs, etc. In order to facilitate the operation, I've created a set of very small, very specific C++ objects that perform dedicated processing, which are listed in Table 13.2. These objects are lightweight and dedicated to KS algorithms, with simple *set*, *process*, and *render* functions.

13.6.1 Exciter, Resonator, and PluckPosFilter C++ Objects

The *Exciter* and *Resonator* objects complete the *KSCore* for the *KSOScillator* and implement three KS models: nylon string guitar, distorted electric guitar, and electric bass guitar. The exciter object implements the block diagram in the dotted box in Figure 13.9 and uses an EG-shaped noise burst as the excitation. The simple Attack-Hold-Release (AHR) EG is implemented and can produce both short noise bursts and longer, quasi-bowed-string excitation signals. When dealing with short noise bursts or impulsive signals, a DC offset may be present that will build up in the resonator. A simple first order HPF is used to block the DC. The *Resonator* object implements the single delay architecture in Figure 13.2(b) and is unchanged.

Table 13.2 List of lightweight objects specifically for KS algorithms

Object Name	*Functionality*	*Important Methods*	*Notes*
ResDelayLine	Resonator integer length delay line	*readDelay* *writeDelay* *setDelayTime*	Length set to correspond to MIDI note 0 at ~8 Hz
FracDelayAPF	Resonator fractional delay element	*setAlpha* *processAudioSample*	Single coefficient α sets fractional delay
ResLoopFilter	Simple feed-forward LPF with ½ sample delay	*processAudioSample*	$a_0 = a_1 = 0.5$
ExciterEG	Dedicated AHR EG with analog EG curvature	*setParameters(AHR)* *startEG* *render*	Lean version of *AnalogEG* with only A, H, and R
DCRemovalFilter	First order HPF with f_c = 2 Hz to remove DC offsets	*reset* *processAudioSample*	Coefficients are hard-coded
PluckPosFilter	Inverse comb filter with selectable bridge integrator and pickup filter	*reset* *processAudioSample*	Options for output at pluck position, pickup, and bridge, and combinations of them
HighShelvingFilter	High shelving filter with f_c (Hz) and boost/cut (dB)	*setParameters* *processAudioSample*	Adds HF energy to exciter signal with shelf f_c = 2 kHz
ParametricFilter	Non-constant-Q parametric filter for body resonance	*setParameters* *processAudioSample*	Used for body filter, post processing; varies with model
LP2Filter	Second order biquad LPF	*setParameters* *processAudioSample*	Used for the amp simulator and pickup filters
LP1PFilter	First order one-pole LPF	*setParameters* *processAudioSample*	Used for the bridge pickup

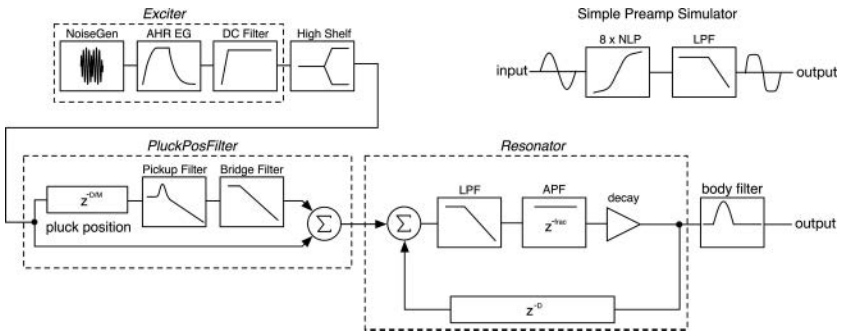


Figure 13.9 The block diagram of the KS algorithm packaged as the *KSOCore* includes the exciter, pluck, pickup, and bridge filters, and the resonator and body filter; the inset shows the very simple guitar preamp simulator, consisting of eight stages of waveshaping followed by a second order LPF

Table 13.3 shows the *Exciter* object's member objects and methods. The sub-objects do the work, and the *render* method simply generates the noise and shapes it with the EG. Notice that the function takes an argument *coupledInput* that allows you to couple the resonator back into the exciter for experimentation. Finally, the DC is removed and the excitation value returned.

```
double render(double coupledInput = 0.0)
{
     double noise = noiseGen.doWhiteNoise();
     double eg = noiseEG.render();
     double ahr = (noise * eg) + coupledInput;
     return dcFilter.processAudioSample(ahr);
}
```

Table 13.4 shows the *Resonator* object's member objects and methods. Once again, the sub-objects do the work, and the *process* method follows Figure 13.6, starting with reading the output of the delay line to sum with the incoming exciter signal:

```
double Resonator::process(double xn)
{
     // --- read delay
     double delayOut = delayLine.readDelay();

     // --- filter input + delay output
     double filterOut = loopFilter.processAudioSample(xn + delayOut);

     // --- create fractional delay with APF
     double yn = fracDelayAPF.processAudioSample(filterOut);

     // --- write the value into the delay and scale
     delayLine.writeDelay(yn*decay);

     // --- done
     return yn;
}
```

Table 13.3 The *Exciter* members and brief description

Exciter Member	*Argument/Variable*	*Description*
NoiseGenerator	*noiseGenerator*	Creates white noise; see Section 8.1
ExciterEG	*noiseEG*	AHR EG for shaping the noise burst
DCRemovalFilter	*dcFilter*	HPF to remove DC offset, f_c fixed at 2 Hz
setParameters( )	*attackTime_mSec* *holdTime_mSec* *releaseTime_mSec*	Sets EG parameters
startExciter( )	None	Places the EG into attack state, ready to render
render( )	None	Renders the exciter output samples, one at a time

Table 13.4 The *Resonator* members and brief description

Resonator Member	*Argument/Variable*	*Description*
ResDelayLine	*delayLine*	Circular buffer for integer resonator delay
FracDelayAPF	*fracDelayAPF*	Generates fractional delay for loop tuning
ResLoopFilter	*loopFilter*	Simple feed forward LPF with ½ sample group delay
setParameters( )	*frequency* *decay*	Sets the resonator's frequency and attenuation parameters
process( )	*xn*	Process a sample through the resonator loop, taking output, as shown in Figure 13.6

Table 13.5 shows the *PluckPosFilter* object's member objects and methods. This object is set up to allow you to choose from any output point using multiple combinations of filters according to the *PluckFilterType*; each model will use a different output point.

```
enum class PluckFilterType {kPluck, kPluckAndBridge, kPickup,
                            kPluckAndPickup, kBridge,
                            kPluckPickupBridge};
```

The *PluckPosFilter* requires a delay line for the comb filter, a one-pole LPF for the bridge integrator, and a second order resonant LPF for the pickup filter, which are set up in the reset function as follows:

1 *combDelay*: length is set for the lowest possible pitch, MIDI note 0 at the current f_s
2 *bridgeIntegrator:* f_c = 20 Hz (you may lower this, but you will need to boost the output to accommodate the loss of energy)
3 *pickupFilter:* f_c = 2.5 kHz, Q = 1.5 (Q plays an important role in the pickup's tonal coloration; experiment with this filter)

The *processAudioSample* function decodes the input parameter and returns the desired output value. Notice the scalar multipliers that make up for lost energy in the bridge and pickup filters; feel free to experiment with these as needed.

```
if (type == PluckFilterType::kBridge)
     return 12.0*bridgeIntegrator.processAudioSample(xn);

if (type == PluckFilterType::kPickup)
     return pickupFilter.processAudioSample(xn);

// --- pluck position
double yn = combDelay.readDelay();
combDelay.writeDelay(xn);

// --- output pluck
double pluck = 0.5*(xn - yn);
```

```
if (type == PluckFilterType::kPluck)
     return pluck;

// --- pluck and pickup
if (type == PluckFilterType::kPluckAndPickup)
     return pickupFilter.processAudioSample(pluck);

// --- pluck and bridge
if (type == PluckFilterType::kPluckAndBridge)
     return 12.0*bridgeIntegrator.processAudioSample(pluck);

if (type == PluckFilterType::kPluckPickupBridge)
{
     double pu = 2.0*pickupFilter.processAudioSample(pluck);
     return 12.0*bridgeIntegrator.processAudioSample(pu);
}
```

13.7 KSOscillator and KSOCore

The *KSOscillator* and *KSOCore* objects are shown in block form in Figure 13.10. The model implements a nylon string guitar, and distorted electric and bass guitar KS algorithms. The differences between the three variations lies mainly in filter settings and the addition of a simple preamp simulator for the distorted guitar. This is the only SynthLab oscillator that has just one core as it is a very specific kind of algorithm.

13.7.1 KSOCore Mod Knobs

The *KSOscParameters* structure contains the GUI parameters, which are very application-specific and consist of the AHR EG controls and the decay control. You will notice that there are no tuning controls; however, in unison mode, the four voices will be detuned as with the other oscillators. You may decide to remove this functionality. The mod knobs are set up for the rest of the functionality shown in Figure 13.9 and consist of:

1 *AmpTweak*: +/−12 dB of final amplitude adjustment, corresponding to the output control on the other oscillators
2 *Body*: a medium-Q parametric filter is placed at the output of the system to mimic the fundamental body resonances of the models
3 *Bite*: a high-shelving filter after the *Exciter* to add high harmonics and brightness to the excitation signal

Table 13.5 The *PluckPosFilter* members and brief description

Exciter Member	*Argument/Variable*	*Notes*
DelayLine	*combDelay*	Length defaults to lowest MIDI note
LP1PFilter	*bridgeIntegrator*	Lossy integrator with f_c = 20 Hz
LP2Filter	*pickupFilter*	Resonant LPF; avoid high Q, or feedback may result

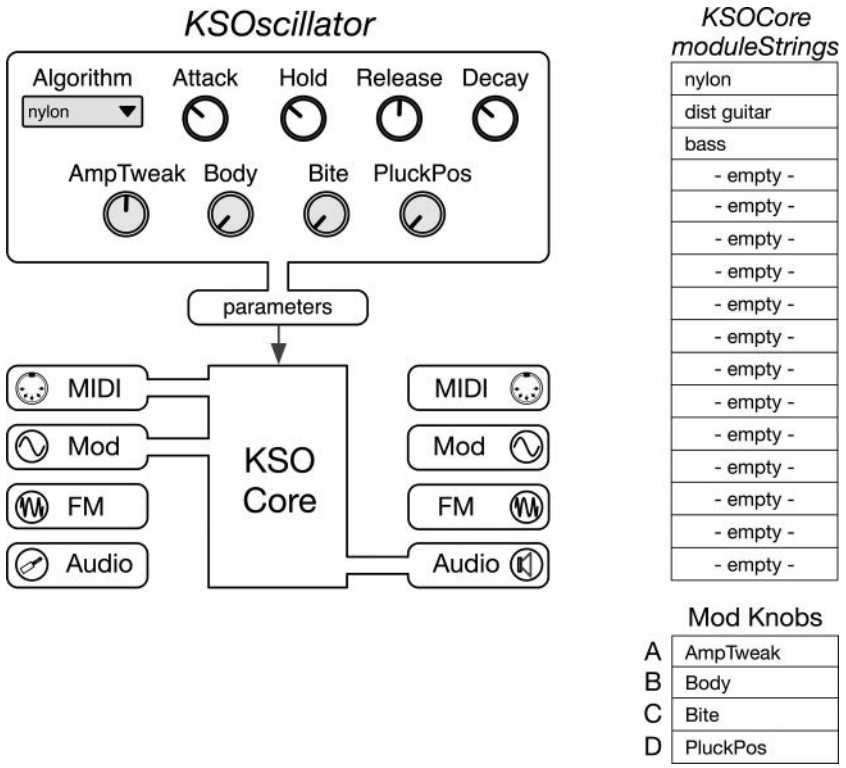


Figure 13.10 The *KSOscillator* and *KSOCore* block diagrams, module strings, mod inputs, and GUI controls

4 *PluckPos*: controls the pluck position and works backwards on a range of [10, 2], establishing the denominator of the pluck position fraction

13.8 *KSOCore* Member Objects

Table 13.6 lists the core member objects and their KS functionality.

13.9 Core Programming Notes

With the C++ files open, compare the programming notes with the code that you see, starting with the class descriptions of each core. Sections 13.9.1 through 13.9.5 summarize the five operational phases, plus constructor for the wavetable cores. Make sure to examine the code while digesting their operational phase details.

13.9.1 Construction Phase

As with all of the *ModuleCore* objects, you first assign the module name and module type, then set the module strings. For the KS oscillator, these are the names of the plucked string instruments.

Table 13.6 KSOCore member objects/variables

KS Functionality	*C++ Object/Variables*	*Notes*
Exciter	*Exciter*	AHR noise exciter
Resonator	*Resonator*	One-delay-line resonator
Pluck position, pickup, & bridge	*PluckPosFilter*	Includes scalar values to make up for energy losses; feel free to adjust as needed
Brightness	*HighShelfFilter*	Adds high harmonics to the exciter
Body resonance	*ParametricFilter*	Different center frequencies for the three models
Distortion	*LP2Filter*	The simple preamp simulator uses a second order filter to smooth harsh harmonics

13.9.2 Reset Phase

The *reset* method simply calls the *reset* method on each of the member objects. In addition, the distortion filter is hardcoded for f_c = 2 kz and Q = 1.0 (feel free to experiment)

```
distortionFilter.reset(sampleRate);
distortionFilter.setParameters(2000.0, 1.0); // fc and Q
```

13.9.3 Note-On Phase

The note-on event is easily serviced; we first save the incoming MIDI pitch to use for update calculations. Then, the sub-objects have their delay lines flushed with the *flushDelays* or *clear* functions, depending on the object. This is crucial to avoid corrupted information in the delays. Most importantly, the exciter's *startExciter* method is called, placing the exciter EG into the attack state and making it ready for render.

```
// --- parameters
midiPitch = processInfo.noteEvent.midiPitch;

// --- reset
resonator.flushDelays();
pluckPosFilter.clear();

// --- start excitation
exciter.startExciter();
```

13.9.4 Update Phase

The core's *update* method follows the normal pitch modulation calculation detailed in Section 6.8.1. After the modulated oscillator frequency is calculated, the resonator and pluck position objects are updated using the oscillator frequency to calculate the delay lengths. Notice that the resonator's *setParameter* function returns the newly calculated delay line length, including the fraction. The pluck position mod knob is quantized to an integer value and used to calculate the pluck position filter's new delay length.

```
// --- BOUND the value to our range
boundValue(oscillatorFrequency, OSC _ FMIN, OSC _ FMAX);

// --- Resonator:
double delayLen = resonator.setParameters(
                              oscillatorFrequency, parameters->decay);

// --- Pluck Position:
pluckPosition = (uint32 _ t)getModKnobValueLinear(
                 parameters->modKnobValue[MOD _ KNOB _ D], 10.0, 2.0);

pluckPosFilter.setDelayInSamples(delayLen / pluckPosition);
```

The exciter attack, hold, and release times are updated, followed by the bite and body filters. Notice how the mod knobs map to boost-only values for both filters and that the center frequencies are different for each model. The high shelf "bite" filter has f_c = 2 kHz with boost from 0.0 to 20.0 dB.

```
// --- filters
bite _ dB = getModKnobValueLinear(parameters->modKnobValue[MOD _ KNOB _ C],
                                  0.0, 20.0);

highShelfFilter.setParameters(2000.0, bite _ dB);
```

The "body" filter is a parametric boost-only filter with 0.0 to +6 dB of gain, and various f_c and Q values depending on the model – these too are highly experimental.

```
body _ dB = getModKnobValueLinear(parameters->modKnobValue[MOD _ KNOB _ B],
                                  0.0, 6.0);

// --- fc, Q vary with model
if (parameters->algorithmIndex == kNylonGtr)
     bodyFilter.setParameters(400.0, 1.0, body _ dB);
else if (parameters->algorithmIndex == kDistGtr)
     bodyFilter.setParameters(300.0, 2.0, body _ dB);
else if (parameters->algorithmIndex == kBass)
     bodyFilter.setParameters(250.0, 1.0, body _ dB);
```

13.9.5 Render Phase

The *render* function is fairly simple because the member objects do most of the work; it follows the flow of Figure 13.8, from exciter to pluck position filter to resonator to body filter in that order. The fundamental differences are in the pluck position filter, whose output is tapped from different locations for each model, and the added preamp simulator for the distorted guitar model. The process starts with the exciter and high-shelf filter:

```
double input = exciter.render();
input = highShelfFilter.processAudioSample(input);
```

The pluck position is altered by model:

```
if (parameters->algorithmIndex == kNylonGtr)
     input = pluckPosFilter.processAudioSample(input,
                                        PluckFilterType::kPluckAndBridge);

else if (parameters->algorithmIndex == kDistGtr)
     input = pluckPosFilter.processAudioSample(input,
                                        PluckFilterType::kPluckAndPickup);
else if (parameters->algorithmIndex == kBass)
     input = pluckPosFilter.processAudioSample(input,
                                PluckFilterType::kPluckPickupBridge);
```

The bass is lowpass filtered twice through the pickup and bridge filters, and has the least high harmonics. The input is applied to the resonator object, then the preamp simulator is applied to the distorted guitar output only.

```
// --- resonate the excitation
double oscOutput = resonator.process(input);

// --- VERY simple guitar preamp sim
if (parameters->algorithmIndex == kDistGtr)
{
     int stages = 8;
     for (int i = 0; i < stages; i++)
          oscOutput = tanhWaveShaper(oscOutput, 1.0);

     // --- drop -6dB; make up for energy boost
     oscOutput = 0.5 * distortionFilter.processAudioSample(oscOutput);
}
```

Finally, the body filter and output amplitude values are applied, and the output is written:

```
// --- add resonance if desired
oscOutput = bodyFilter.processAudioSample(oscOutput);
oscOutput *= outputAmplitude;

// --- write to output buffers
leftOutBuffer[i] = oscOutput;
rightOutBuffer[i] = oscOutput;
```

13.10 Extensions to the KS Algorithm

The plucked string algorithm has been augmented over the years, with numerous extensions and improvements. An excellent source you may use as a starting point is Valamaki, which documents the additions and contains a plethora of references and sources for more information and study. These additions include:

- Electric guitar and bass pickup output (requires dual delay lines)
- Pluck shaping to model variations in pluck and pick stroke angles
- Sympathetic vibration of open strings; applications to the old-fashioned harp-guitar

The chapter references here also contain information about hammered and bowed-string models, which include coupling between resonator and exciter, and involve scattering junctions.

13.11 Exercises

13.11.1 SynthLab-DM: High Register Accommodations

The high-shelving filter allows the boosting of high frequencies in the exciter; however if you play very high notes, you will notice that the attack loses its sharpness (Jaffe and Smith 1983). Implement a system that adjusts the high-shelving filter's gain value and/or shelving frequency to accommodate notes in the upper register. Next, note that, for the nylon string model, the notes lose energy very quickly in the upper register, and though that is similar to the real instrument, you may want to try to extend the note duration by manipulating the loop's *decay* parameter, which is set in the resonator model (see the code in Section 13.6.1).

13.11.2 SynthLab-DM: 12-String Guitar

A 12-string guitar uses pairs of strings, called courses, that are tuned one octave apart for the lower four strings and in unison for the upper pair of B and E strings. You can implement these extra strings as additional resonators and feed the excitation signal to each in parallel. For the unison-tuned string pairs, you can apply a bit of detuning for a thicker sound. Implement your own 12-string guitar model as a new *KSOCore*, or modify the existing core to add the 12-string option. If you want to go further with realism, research the sympathetic vibration modeling for open-string chords, and add that to your implementation.

13.11.3 SynthLab-DM: Body Resonance and Electric Guitar Feedback

The exciter's *render* method includes an input so that the exciter may be re-stimulated from the resonator. This has two immediate applications: first, the body of the guitar usually has multiple resonant peaks (see Fletcher and Rossing), and this may be simulated with resonant filters that feed back from the resonator into the exciter. Many electric guitarists feature feedback in their styles; one method to create this is to rest the headstock of the guitar on the vibrating amp to couple the energy back into the instrument. This is equivalent to coupling part of the resonator's output back into the exciter directly or through a filter; in both cases, care is needed to avoid runaway feedback. Postulate a model for body resonances and/or guitar feedback that re-energize the exciter with part of the resonator's signal and implement it. Guitar feedback is often on octave boundaries, so that would be a place to start your experiments. Use the *coupledInput* variable for the exciter object's render function to channel audio back to the exciter.

Bibliography

Karjalainen, Matti, Välimäki, Vesa & Tolonen, Tero. 1998. "Plucked String Models: From the Karplus-Strong Algorithm to Digital Waveguides and Beyond." *Computer Music Journal*, vol. 22, no. 3, pp. 17–32, Cambridge: MIT Press.

Jaffe, David & Smith, Julius. 1983. "Extensions of the Karplus-Strong Plucked-String Algorithm." *Computer Music Journal*, vol. 7, no. 2, pp. 56–69, Cambridge: MIT Press.

Kahrs, Mark & Karlheinz, Brandenburg. 1998. *Applications of Digital Signal Processing to Audio and Acoustics*, Chap. 10. Boston: Klewer Academic Publishers.

Karjalainan, Matti, Välimäki, Vesa & Jánosy, Zoltan. 1993. "Towards High-Quality Sound Synthesis of the Guitar and String Instruments." Proceedings of the 1993 International Computer Music Conference, San Francisco, California: International Computer Music Association, pp. 56–63.

Karplus, Kevin & Strong, Alex 1983. "Digital Synthesis of Plucked-String and Drum Timbres." *Computer Music Journal*, vol. 7, no. 2, pp. 43–55. Reprinted in C. Roads, ed. 1989. *The Music Machine*. Cambridge: MIT Press.

Smith, Julius Orion. 1993. "Efficient Synthesis of Stringed Musical Instruments." Proceedings of the 1993 International Computer Music Conference, San Francisco, California: International Computer Music Association, pp. 64–71.

Smith, Julius. 2010. *Physical Audio Signal Processing*, Chap. 9. Stanford: Center for Computer Research in Music and Acoustics.

14 The Modulation Matrix

Chapter 6 detailed the various types of modulations and calculations, then each of the synth modules implemented an *update* function that used one or more of the modulations and corresponding calculations. These synth modules are modulation destinations because they change their internal parameters when modulation values are applied. The LFO and EG modules are modulation sources because their outputs are applied to the modulation destinations. For FM synthesis, pitched oscillators are used as both modulation sources and modulation destinations; this is possible for other modules as well, as it is with the LFOs that can modulate each other's oscillator frequency. The modulation matrix is used to connect sources and destinations during the voice object's update phase. The matrix must therefore know the locations of all of the source values (the outputs of the LFOs and EGs) as well as the locations of the destinations (the modulation inputs). We will use the *ModMatrix* object to connect our modules together.

14.1 Modulation Inputs and Outputs

Each *SynthModule* object includes two *Modulators* member objects: one for incoming modulation values and another for modulation outputs named *modulationInput* and *modulationOutput*. The *Modulators* object is a thin wrapper for a statically declared array of 32 *double* values, with get/set function access to the array elements and a special array pointer access for the modulation matrix. The elements are indexed with a simple *enum*, and all *SynthModules* share the same modulation input array index names, as shown in Figure 14.1(a). There are currently 22 modulation inputs when you include the additional wave-sequencing object in SynthLab-WS, and you may add as many more as you like as the array will be self-sizing. Each module that generates a modulation output writes into its *modulationOutput* using its own module-specific index names, as shown for the LFO core in Figure 14.1(b) and (c). The EG has another set of index names, as shown in Figure 14.1(d).

14.2 Modulation Routings

A modulation routing (or mod routing) is a specific connection between the modulation output array of one object and the modulation input array of another, such as the connection shown in Figure 14.2, where the mod source is LFO1's normal output (array slot 0), and the mod destination is OSC2's bipolar input (array slot 3). Each routing is a unique (source, destination) pair. In the SynthLab *ModMatrix* object, any number of sources may modulate the same destination. The modulation matrix's parameter object stores a set of modulation routings, and you may add, remove, or re-configure the routings at any time. You may also define pre-wired routings for a

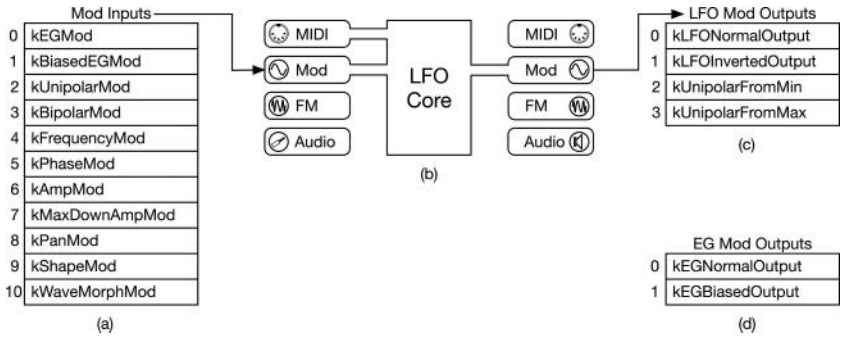


Figure 14.1 (a) The modulation input array is indexed the same way for all *SynthModules* using constant declarations for kEGMod, kBiasedEGMod, etc… (b) the LFO Core exposes a modulation input and a modulation output (c) the modulation output array is indexed specifically for the LFO using constant declarations while (d) shows the indexing for the EG's modulation output array for comparison

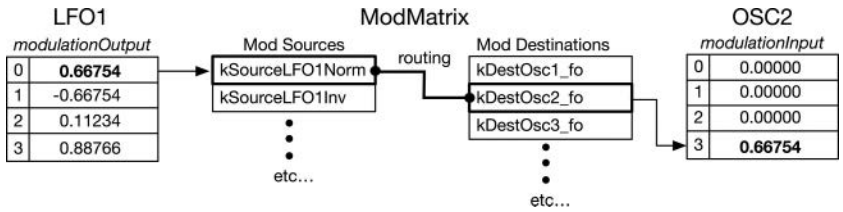


Figure 14.2 A modulation routing that connects LFO1's normal output to OSC2's bipolar modulation input, which modulates the oscillator's frequency

semi-modular design: for example, the amp EG's output is always hardwired to the DCA's EG input, while a filter EG's output is always connected to a filter's f_c modulation input.

The *ModMatrix* source and destination arrays are indexed with their own simple *enums* that need to be separate from the module *enums*, and you can easily add more, as needed. Mod sources are prefixed with "source" and likewise with destinations, so the constants are easy to decode.

```
enum modSource
{
      // --- LFOs here
      kSourceLFO1 _ Norm,
      kSourceLFO1 _ Inv,

      kSourceLFO2 _ Norm,
      kSourceLFO2 _ Inv,

      // --- EGs here
      kSourceAmpEG _ Norm,
      kSourceAmpEG _ Bias,
      // etc . . .
```

```
    // --- remain last, will always be the size of modulator array
    kNumberModSources
};

enum modDestination
{
    kDestOsc1 _ fo,
    kDestOsc2 _ fo,
    // etc . . .

    kDestOsc1 _ Amp,
    kDestOsc2 _ Amp,
    // etc . . .

    // --- remain last, will always be the size of modulator array
    kNumberModDestinations
};
```

14.2.1 Mod Routing Selection and Intensity Controls

Modulation matrices take on varying forms in different synth designs. Figure 14.3 shows two out of the many possible GUI design choices for a modulation matrix. The modulation matrix in Figure 14.3(a) looks like an actual matrix, with rows (sources) and columns (destinations). On the EMS VCS3, pins are inserted into holes, which make the electrical connection between the analog source and the destination (the pins contain resistors for summing and attenuating). On the Arturia Prophet VS plugin, GUI buttons are used to create the mod routings, and this synth includes intensity controls for the sources and destinations. You can think of them as global send and receive controls per source and destination. In this mod matrix, the user can map any or all sources to any or all destinations, as desired. You can see that both the LFO2 and EG2 modulate the same destination – in this case, the mod values are summed together without scaling.

Figure 14.3(b) shows a different type of mod matrix; here, the user is restricted to only four modulation routings and presented with a selection list for each routing pair. This matrix features individual intensity controls for each routing, allowing for finer control per mod routing connection. In this version, the selection lists must also include a "no connection" choice.

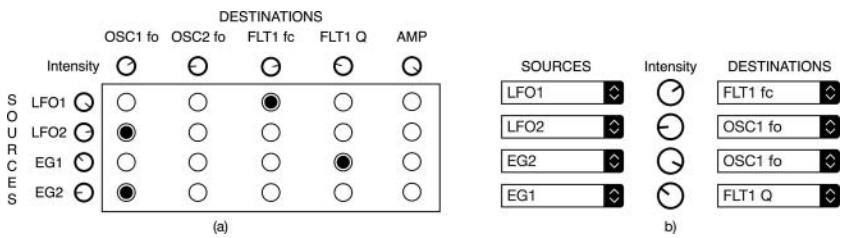


Figure 14.3 Modulation matrix designs include (a) pin/button programmable with global source and destination intensity controls, and (b) a limited choice matrix with individual channel routing intensity controls

14.3 Mod Matrix Channel Routing

The modulation matrix is packaged as a C++ object named *ModMatrix*. Figures 14.1–14.3 make apparent that the *ModMatrix* object will need to know the source and destination locations in order to pull and push the mod values into place. On the destination end, it will need to accumulate the mod values since multiple sources can modulate a single destination. There are numerous ways in which to handle this, including individual arrays for sources and destinations. The method in *ModMatrix* is to store pointers to the statically declared modulation input and output array locations on each synth module; it reads from the source array slot and accumulates into the destination array location. This must be done at the voice level, and the *SynthVoice* constructor sets up the sources, destinations, and hardwired routings.

In order to accommodate many mod matrix schemes, the *ModMatrix* object includes multiple intensity controls, a hardwired bypass path that overrides the channel path, and channel enable and disable controls, as shown in Figure 14.4(a). The intensity controls are source, channel, and destination; these may look like overkill, but they are there for maximum flexibility in your designs. And you will rarely use all three intensity controls. Figure 14.4(b) shows how the object is configured to match the mod matrix in Figure 14.3(a). In this case, there is a single GUI intensity control for each source and destination, but they connect to the same parameter within each modulation routing. Figure 14.4(c) shows the configuration for Figure 14.3(b), which only requires an enable switch and channel intensity control.

14.4 *ModSource* and *ModDestination* for GUI Controls

Ultimately, the user programs the *ModMatrix* via the GUI, or you can set up the object programmatically from within the *SynthVoice* object. There is only one set of *ModMatrix* controls on the synth, but these routings apply to all of the voices. However, each voice contains its own set of modules (LFOs, EGs, etc.), which are locally owned, and they have their own local modulation

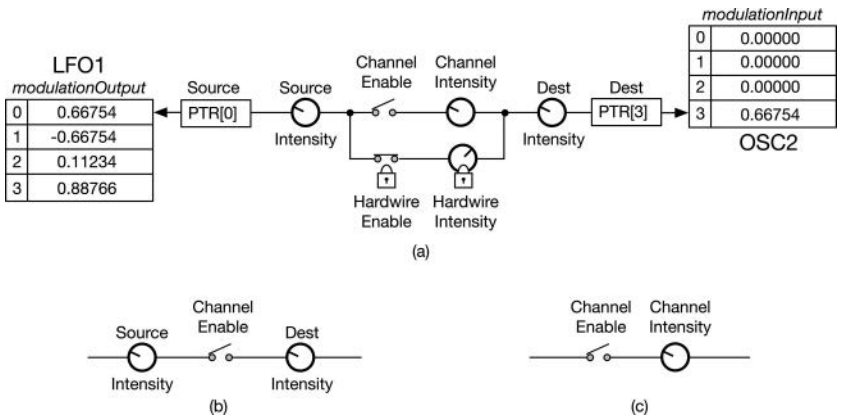


Figure 14.4 (a) The *ModMatrix* configuration for a single routing shows the multiple intensity controls and enable-switches; in this example, the hardwire enable switch is overriding the channel branch while (b) and (c) demonstrate the routing configurations that match Figures 14.3 (a) and (b), respectively

input and output arrays, and therefore need their own set of array pointers – these are not shared resources. To manage this issue, I use one *ModMatrix* object per voice, which stores pointers to the voice's member object modulation array slots. In order to share the GUI parameters for routing and intensity functions, I use *std::shared_ptrs* to *std::arrays* in the *ModMatrixParameters* structure, which handles user input from the GUI. These arrays are named "rows" for sources and "columns" for destinations. There are two *std::arrays*, one each for the sources and destinations, holding tiny C++ structures called *ModSource* and *ModDestination*; these store the GUI routing and intensity settings.

The *ModSource* structure contains a single intensity parameter, while the *ModDestination* contains its own intensity variable plus the channel and hardwire-enable and intensity parameters. The *ModDestination* is special because there may be many routing channels merging into a single destination. Thus, the *ModDestination* parameters are stored in arrays of 48 channels, providing the capacity for up to 48 rows of modulation sources in the matrix; if you require more, simply change the constant definition for the array size. Notice that the array objects are sized to *modSourceRows* and *modDestinationColumns* from the *enums* in Section 14.2, so, as you add or remove your own, these sizes will automatically reset.

14.5 *ModMatrixParameters*

The *ModMatrixParameters* structure contains only two member variables: the shared pointers to the source and destination parameter arrays. The GUI controls are parsed into these arrays.

```
// --- modulation matrix rows and columns
std::shared_ptr<std::array<ModSource, kNumberModSources>>
                                        modSourceRows;

std::shared_ptr<std::array<ModDestination, kNumberModDestinations>>
                                        modDestinationColumns;
```

To facilitate transferring the GUI control information into the structures, several helper functions are provided that accept the source and destination index values and intensity controls.

```
void setMM_SourceIntensity
void setMM_DestIntensity

void setMM_ChannelEnable
void setMM_ChannelIntensity

void setMM_HardwireEnable
void setMM_DestHardwireIntensity
void setMM_DestDefaultValue
```

A final function allows a single call to set a complete hardwired routing, including source and destination index values and the hardwired channel intensity.

```
void setMM_HardwiredRouting(uint32_t source, uint32_t destination,
                            double intensity = 1.0)
```

14.6 The *ModMatrix* Object

There will be one *ModMatrix* object for each voice, and it is simple, containing only three member variables: a shared pointer to the parameters and source and destination arrays, sized exactly as needed for holding pointers to the source and destination value locations.

```
// --- parameters
std::shared_ptr<ModMatrixParameters> parameters = nullptr;

// --- arrays to hold source/destination pointers
double* modSourceData[kNumberModSources];
double* modDestinationData[kNumberModDestinations];
```

The *ModMatrix* object also includes a set of functions for adding and removing sources or destinations, along with functions for populating the arrays with local pointers. The only other method is for running the modulation matrix that will happen once per render block – after the modulator components have rendered their values and before the destinations have updated their parameters. The simple *for* loop is organized with columns (destinations) as the outer loop since the destinations may have multiple sources. If the *channelEnable* is not set, the loop continues. This short and simple function provides all of the modulation routing and calculations.

```
// --- loop over destinations columns & calculate the accumulated
//   modulation value
double modDestinationValue = 0.0;

for (int col = 0; col < kNumberModDestinations; col++)
{
     ModDestination destination = parameters->
                                  modDestinationColumns->at(col);

     modDestinationValue = destination.defautValue;

     for (int row = 0; row < kNumberModSources; row++)
     {
          if (!modSourceData[row] ||
               destination.channelEnable[row] == 0)
               continue;

          // --- get local modulation value
          double modSourceValue = *modSourceData[row];

          ModSource source = parameters->modSourceRows->at(row);

          // --- scale it
          modSourceValue *= source.intensity;

          // --- choose intensity
```

```
            if (destination.channelHardwire[row])
                modDestinationValue += modSourceValue *
                                    destination.hardwireIntensity[row];
            else
                modDestinationValue += destination.intensity *
                                                        modSourceValue;
        }
        // --- write to the output array
        if (modDestinationData[col])
            *modDestinationData[col] = modDestinationValue;
}
```

14.7 Initializing the *ModMatrix*

The *SynthVoiceParameters* structure declares the shared pointer to the *ModMatrixParameters* and uses the *make_shared* function such that the first *SynthVoice* that is created will create the shared structure for the rest of the voices so the GUI controls are handled without a problem.

The *SynthVoice* uses *std::unique_ptrs* for its modules as they are never shared across voices. The *ModMatrix* is declared in the class definition, then created in the constructor, passing the shared parameters during *ModMatrix* construction.

```
// --- each voice has a modulation matrix
//   but rows/columns are shared via matrix parameters
std::unique_ptr<ModMatrix> modMatrix;

// --- creation
modMatrix.reset(new ModMatrix(parameters->modMatrixParameters));
```

In order to use the *ModMatrix*, the *SynthVoice* object needs to program its member object with several pieces of information. First, it needs to set up the pointers to each module's source or destination array location. To add sources or destinations, you use the helper functions. Setting up the LFO1 as a source and OSC2's f_o as a destination is simple – the oscillators all use the *kBipolarMod* array slot for modulating the f_o, so you request pointers via *getModArrayPtr* and provide the index:

```
// --- add LFO1 Output Source index and pointer
modMatrix->addModSource(kSourceLFO1_Norm,
     lfo1->getModulationOutput()->getModArrayPtr(kLFONormalOutput));

// --- add wavetable oscillator fo as destination
modMatrix->addModDestination(kDestOsc1_fo,
     wtOsc->getModulationInput()->getModArrayPtr(kBipolarMod));
```

To remove a mod source or destination during runtime, you use the *clear* methods:

```
modMatrix->clearModSource(kSourceLFO1_Norm);
modMatrix->clearModDestination(kDestOsc1_fo);
```

14.8 Programming the *ModMatrix*

Each *SynthVoice's* mod matrix object is specific to it and only contains the arrays of local pointers. To program the *ModMatrix*, you use the *ModMatrixParameters*, either via GUI control information or programmatically from within the voice object.

To set up a hardwired routing from the *AmpEG* to the DCA's EG mod input, you write:

```
parameters->modMatrixParameters->setMM _ HardwiredRouting(
                                    kSourceAmpEG _ Norm, kDestDCA _ EGMod);
```

All destinations have a default value that is set to 0.0 on creation, but there are some routings that require a default setting of 1.0 to prevent accidental silence. This is necessary for the *AmpEG* to DCA EG mod and is handled with another helper function:

```
parameters->modMatrixParameters->setMM _ DestDefaultValue(
                                                kDestDCA _ AmpMod, 1.0);
```

To enable the routing from LFO1 to OSC2's f_o, you write the following; notice that the enable argument *true* may come from a GUI control switch.

```
parameters->modMatrixParameters->setMM _ ChannelEnable(
                                  kSourceLFO1 _ Norm, kDestOsc2 _ fo, true);
```

To set the channel intensity to 0.707 for this routing, you write the following; as with the channel enable, the 0.707 value may come from a GUI knob or slider.

```
parameters->modMatrixParameters->setMM _ ChannelIntensity(
                                 kSourceLFO1 _ Norm, kDestOsc2 _ fo, 0.707);
```

The source and destination global intensity values are set with the same types of functions:

```
setMM _ SourceIntensity(kSourceLFO1 _ Norm, 0.456);
setMM _ DestIntensity(kDestOsc2 _ fo, 0.789);
```

14.9 Mod Matrix Transforms

The SynthLab modulation matrix consists of sources that are both bipolar (LFO and wave sequencer) and unipolar (EG). For most modulations, you may freely mix these together. On occasion, you may need to declare a destination as requiring a unipolar-to-bipolar or bipolar-to-unipolar transform, as happens with the EG re-triggering modulation in Section 7.7. The MMA DLS specs also list inversion (multiply by −1), along with the curved transforms as transform possibilities. When you declare a modulation destination, you may also add an optional transform into the function arguments.

```
modMatrix->addModDestination(kDestAmpEGRetrigger,
                  ampEG->getModulationInput()->getModArrayPtr
                  (kTriggerMod), kMMTransformUnipolar);
```

14.10 Running the *ModMatrix*

The *ModMatrix* is executed after the modulators have generated their outputs and before the destinations are updated during the *SynthVoice::renderVoice* operation. For example, in a wavetable synth voice, you would write this:

```
// --- render modulators
lfo1->render(samplesToProcess);
lfo2->render(samplesToProcess);

// --- update/render (add more here)
ampEG->render(samplesToProcess);
filterEG->render(samplesToProcess);
auxEG->render(samplesToProcess);

// --- run the mod matrix
modMatrix->runModMatrix();

// --- update and render everything else
wtOsc1->render(samplesToProcess);
wtOsc2->render(samplesToProcess);

// --- etc…
```

Bibliography

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 1*. https://www.midi.org/specifications-old/item/dls-level-1-specification, Accessed October 14, 2020

MIDI Manufacturer's Association. 1999. *Downloadable Sounds Level 2*. https://www.midi.org/specifications-old/item/dls-level-2-specification, Accessed October 14, 2020

SynthLab Documentation. 2020. www.willpirkle.com/synthlab-docs, Accessed October 14, 2020

15 Wave Morphing and Wave Sequencing

In recent years, there has been a resurgence in dynamic timbral synthesis with waveform crossfading. Wolfgang Palm, Sequential Circuits, and Korg have released synths that generate long evolving sounds, percussive loops, and searing lead patches using fairly simple waveform crossfade techniques. These include:

1 **Wave morphing**: crossfading a series of waveforms, one after the other, using an LFO, EG, or other modulation source that smoothly morphs between them; there are only ever two waveforms that are being crossfaded or mixed at a given time
2 **Wave sequencing**: crossfading a series of waveforms, one after another, based on a sequencer designed in order to perform a combination of crossfading and holding such that the waveform morphing follows a rhythmic sequence or pattern; unlike wave morphing, sequencing often involves holding the waveforms using rhythmic note durations between crossfades or allowing for silent gaps that break up the sequence, producing rhythmic patterns without necessarily needing drum sounds
3 **Vector synthesis**: implements waveform crossfading using a joystick or pre-programmed mix envelopes; one, two, three, or four waveforms may be mixed at any given time corresponding to the joystick's four extremes when an *x, y* grid is superimposed on the joystick's motion

Vector mixing is covered in Section 6.4 and requires an X-Y track-pad on the GUI; the original Sequential Circuits equations are given so you can translate the joystick's *x, y* location into a set of four mix values for performing the four-way crossfading that is central to this synthesis method. This chapter is about wave morphing and wave sequencing.

15.1 Wave Banks

Before getting into the details, we need to consider the fact that wave morphing will require multiple data sources, and for simplicity, only wavetables are used for morphing and sequencing. The *SynthEngine* already maintains a wavetable database, as discussed in Section 9.2, so we already have a system for storing and sharing the tables. Wave morphing synths usually employ sets of waveforms that are designed to work together as a group. In SynthLab, a group of wavetables that is imported all at once is called a wave *bank*. The tables are stored in the normal manner and may be used like any other wavetable, but the importing mechanism only involves one step, and the morphing wavetable core has access to all of the waveform names in each bank; thus, it has the unique strings necessary to select the waveforms in succession from the database as it morphs

during the render phase. The *SynthLabBankSet* structure stores pointers to the tables as well as a list of waveform names for the database. All of this is documented online.

15.2 Wave Morphing

Wave morphing involves crossfading over a series of oscillator waveforms in succession using a modulation signal that may come from GUI controls or from a modulator component. Wavetable and PCM samples lend themselves easily to this kind of crossfading, especially when the tables or files are all the same length; here, I use the term *table* to represent any tabular data source, including PCM samples. Wolfgang Palm's Infinite® and Korg's Electribe Wave® are excellent examples, among many others. Infinite is special because it displays morphing in the frequency domain, showing the spectra of the waveform slices rather than the time domain representation that is more common. In addition, Infinite allows you to precisely control the spectral components – for example, you may manually cut out every other harmonic or alter the harmonic envelope of each slice.

Wave morphing requires a set of source tables (slices) that are distributed along a morphing dimension; you need at least two waveforms to morph over, but commercial morphing table collections usually range from four to hundreds of different tables for generating only one note-event. Figure 15.1 shows several variations, and the double-arrows show the morphing dimension. The morphing table sets (or banks) may be made of dissimilar waveforms, as in Figure 15.1(a), which includes a sinusoid, triangle, square, and sawtooth waveform (from back to front), and is named *PrimalWaves* in the SynthLab-WT project. Another bank preparation involves starting with a common waveform and varying one or two parameters of it, taking sample slices at regular intervals, as shown in Figure 15.1(b)'s square wave, whose duty cycle is changed over the course of the bank's waveforms. Figure 15.1(c) shows a bank that starts with a complex waveform processed through filters, waveshapers, and other effects to produce a related set of waveforms.

15.2.1 Wave Morph Modulation

The crossfade modulation that occurs between the fixed waveforms includes a starting point, intensity, and a modulation control signal. The modulation signal that controls the location of the current morphing point is typically a low frequency modulator, such as an LFO, EG, or MIDI CC. Figure 15.2(a) shows an LFO morphing back and forth across a range of four waveforms. The dotted waveforms are the interpolated or in-between combinations of the two fixed waveforms on either side. With slow modulators, there will be thousands of interpolated waveforms between the fixed sources. Figure 15.2(b) shows morphing with an EG that is significantly different. The EG

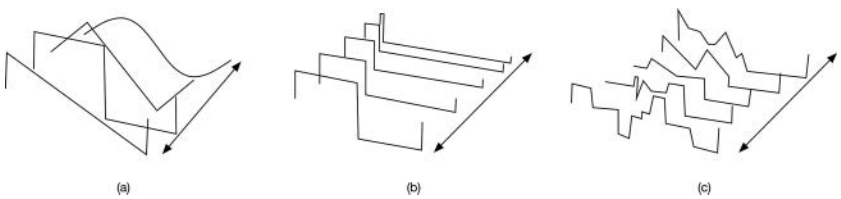


Figure 15.1 Three sets of tables for morphing, including (a) dissimilar waveforms, (b) one waveform whose parameters are adjusted for each slice, and (c) a complex waveform that has undergone multiple passes of filtering or other processing

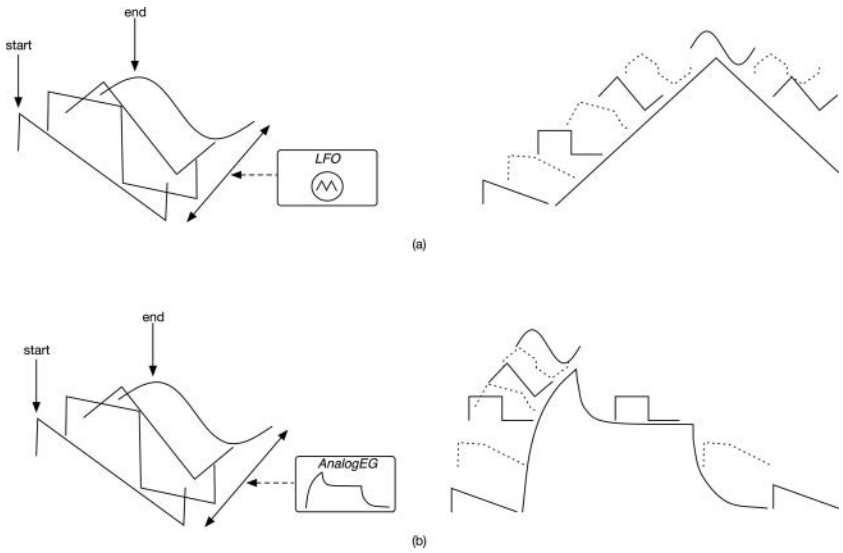


Figure 15.2 (a) An LFO modulates the wave-morphing index, producing a smoothly interpolated succession of waveforms back and forth along the range, while (b) the EG also morphs the waveforms but stops during the sustain portion, holding the current waveform constant prior to release

curve morphs over the waveforms but stops at the sustain morph point and holds that waveform until the key is released, when the morphing moves backward toward the starting waveform. The user can typically control the starting morph waveform with a floating-point control value that allows the starting point to be in between waveforms.

15.3 The *MorphWTCore*

The wavetable oscillator from Section 9.3 loads wavetable module cores that use the wavetable database it provides via the engine. The *ClassicWTCore* is designed to load statically declared wavetables, which are provided in a set of *.h* files and use the *SynthLabTableSet* structure to store the data. Since SynthLab uses interfaces for the wavetable databank, you are free to use whatever mechanism you wish for storing and importing by modifying the existing structure or sub-classing *IWavetableDatabase*. The *MorphWTCore* is also a wavetable core and snaps into the SynthLab-WT project alongside the other cores, though with two differences: it uses the *SynthLabBankSet* to import the tables to the database, and it stores interface pointers for two table sources instead of one, interpolating between them as required. As the morphing index moves along, the two table sources always point to the waveform pair needed for interpolation. The mechanisms for reading and interpolating the individual tables do not change, and the *auxEG* is used as the hardwired morphing modulator, so everything is already present in the SynthLab architecture to support the *MorphWTCore*.

Figure 15.3 shows how the *MorphWTCore* works as a wavetable oscillator core object, and the modules' strings point to wavetable banks, which are really just a pre-set list of wavetables in the database. The *auxEG* is hardwired to the *kWaveMorphMod* array slot, which acts as a unipolar

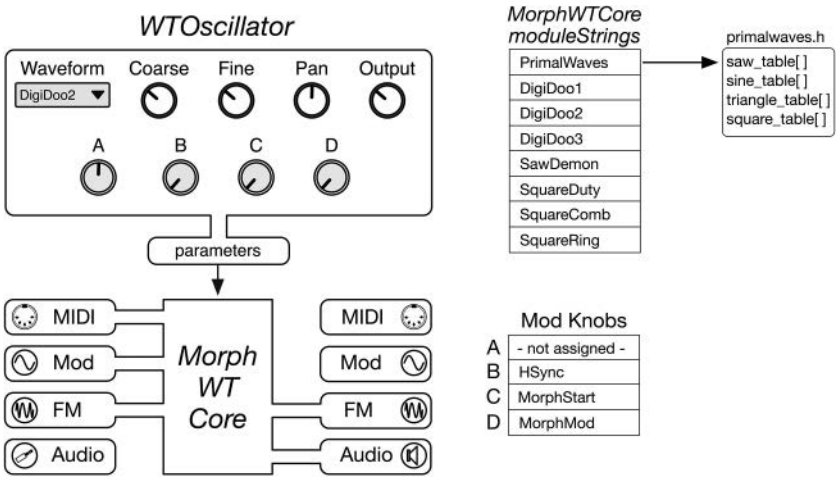


Figure 15.3 The *MorphWTCore* snaps into the *WTOscillator* object and exposes its mod knobs, which are slightly different and include morph start index and morph mod intensity; notice that the module strings point to a bank of wavetables

morph position modulator. The unique modulation input may be connected to an LFO and accepts bipolar modulation based around the center point, which is set with the morph start control.

15.4 *MorphWTCore* Programming Notes

The *MorphWTCore* is fundamentally the same as the other wavetable cores except for the fact that it uses two wavetable sources whose tables are blended through constant power morph modulation (Section 6.2.2). It also initializes the wavetable database using a *SynthLabBankSet*, which is simply an array of *SynthLabTableSet* pointers and a corresponding array of unique wavetable names. The code for setting up the banks is straightforward and involves two parts: first, the core stores a set of *MorphBankData* structures for each set of wavetables that includes the bank name and the table names. There is one structure for each module string, so for the *PrimalWaves* bank, the information would include that bank name plus an array of table names, *DigiSaw*, *DigiSine*, etc. This is the mechanism that connects the bank name the user selects to the set of tables and allows the core to find and load those tables to morph between. The second part simply iterates over the wavetables, querying the database during the reset phase and adding tables as needed, which is just a big loop over the same code that you saw for the normal wavetable oscillator.

15.4.1 MorphWTCore *Construction Phase*

The constructor is identical to the other wavetable core constructors except for the fact that it uses banks of wavetables. The object uses the function *addMorphBankData* to store the set of unique wavetable name strings for each bank. The user only sees the bank names, but the oscillator needs the waveform names to morph across. The *BankDescriptors* are part of my implementation of the

SynthLabBankSet, which contains an array of *SynthLabTableSets*. Remember that you are free to set up your wavetables as you wish as long as you use the *IWavetableDatabase* interface on your wavetable object.

```
moduleType = WTO_MODULE;
moduleName = "Morph WT";

// --- this must be done before setting module strings
uint32_t count = 0;
addMorphBankData("PrimalWaves", PrimalWaves_BankDescriptor, count++);
addMorphBankData("DigiDoo1", DigDoo1_BankDescriptor, count++);

etc...
```

The module strings are then set, using the morph bank names that are stored in the previous operation.

```
for(uint32_t i=0; i<MODULE_STRINGS; i++)
     coreData.moduleStrings[i] = morphBankData[i].bankName.c_str();
```

Finally, the mod knobs labels are assigned according to Figure 15.3.

15.4.2 MorphWTCore *Reset Phase*

The *reset* function is fundamentally the same as it is for the other wavetable cores, but it queries the database for banks of wavetables rather than individual tables, using a helper function that loops through the bank's waveforms, ensuring each is registered with the database.

```
// --- query and add all waveforms in each bank
checkAddWaveBank(PrimalWaves_BankDescriptor, processInfo);
checkAddWaveBank(DigDoo1_BankDescriptor, processInfo);
etc...
```

15.4.3 MorphWTCore *Update Phase*

After performing the usual pitch modulation calculations from Section 6.8.1, the core needs to use the morph modulation input value to select a pair of tables to morph between. The morph start and end index values are used to set up a range that is normalized to the range [0, +1]. Then the wave morph modulation value (from the *auxEG*) is applied to select the pair of tables. The fractional distance between the tables is used as the interpolation point.

```
// --- get the floating point morph position
morphMod = processInfo.modulationInputs->getModValue(kWaveMorphMod);

// --- NOTE -1
morphTables = morphBankData[parameters->wavetableIndex].numTables - 1;

// --- mod knob C is the intensity control
```

```
morphStart = morphTables * parameters->modKnobValue[MOD_KNOB_C];

// --- calculate morph location and save for render
morphLocation = morphMod * (morphTables - morphStart);

boundValue(morphLocation, 0.0, (double)morphTables);

uint32_t table0 = (uint32_t)morphLocation;
uint32_t table1 = (uint32_t)morphLocation + 1;
if (table1 > morphTables)
     table1 = table0;
```

The two table index values are used to identify the tables via their unique waveform names, which were stored in the *MorphBankData* structure during the first step in Section 14.4. The waveform names for *table0* and *table1* are found in the structure, then used to select the wavetables:

```
bankIndex = parameters->wavetableIndex;
table0str = morphBankData[bankIndex].tableNames[table0].c_str()
table1str = morphBankData[bankIndex].tableNames[table1].c_str()

selectedTableSource[0] =
     processInfo.wavetableDatabase->getTableSource(table0str);

selectedTableSource[1] =
     processInfo.wavetableDatabase->getTableSource(table1str);
```

With the two tables selected and the fractional morph location stored, all that remains is blending the two tables during the render phase.

15.4.4 MorphWTCore *Render Phase*

The *render* function is identical to the same function in the other wavetable cores, even in its ability to self hard sync the morphing oscillator. The only difference is in the *renderSample* helper function that performs the morph. If the morph location happens to be a pure integer value, then only one table is used.

```
double MorphWTCore::renderSample(SynthClock& clock){


double mCounter = clock.mcounter;

// --- integer morph location
if (selectedTableSource[0] == selectedTableSource[1])
     return selectedTableSource[0]->readWaveTable(mCounter);
```

After the table read operations, the morph location is split into integer and fractional parts, and the fraction is used to interpolate points between the tables. After the output is formed, the clock is advanced as usual.

```
// --- two table reads
double oscOutput0 = selectedTableSource[0]->readWaveTable(mCounter);
double oscOutput1 = selectedTableSource[1]->readWaveTable(mCounter);

// --- split the fractional index into int.frac parts
double intPart = 0.0;
double morphFraction = modf(morphLocation, &intPart);

// --- const power summing:
double mixValueA = 0.0;
double mixValueB = 0.0;

// --- calculate mix values
calculateConstPwrMixValues(morphFraction, mixValueA, mixValueB);

// --- morph
double oscOutput = oscOutput0*mixValueA + oscOutput1*mixValueB;
```

The self hard sync code is identical to the other wavetable cores since the *renderSample* function does all of the work.

15.5 Wave Sequencing 1.0

Korg introduced another type of crossfade modulation called wave sequencing in the Wavestation® line, which also included vector synthesis. The idea is to set up a sequence of waveforms in steps that are both crossfaded over some duration and held for another duration according to a pattern, often rhythmic in nature. Each track in the sequencer is made up of a sequence of steps. A wave sequence track stores the crossfade and step durations that are applied to morph each successive pair of waveforms into and out of one other, as shown in the "Waves" track of Figure 15.4. The step durations are carried over into other tracks, which adjust pitch and amplitude so these

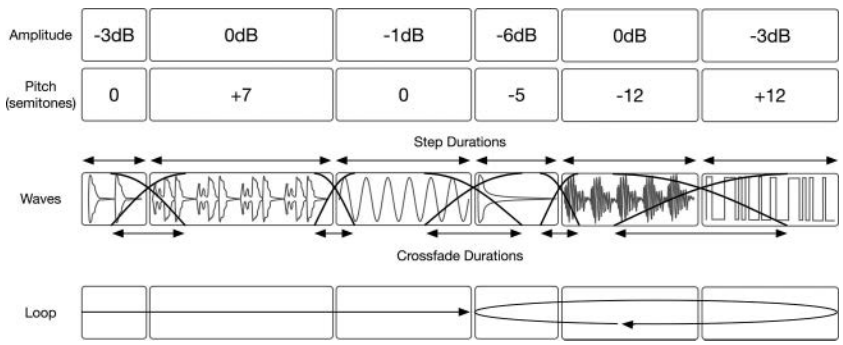


Figure 15.4 Korg's wave sequencer from the Wavestation featured the ability to hold and crossfading waveforms while also modulating each segment's pitch and amplitude; the waveforms actually blend together during the crossfades and are shown separated for easier viewing; the amplitudes were not displayed in dB in the Wavestation

parameters are modulated along with the crossfaded waveforms. A loop system, shown as a loop track in Figure 15.4, sets the looping points with numerous looping options available, including a no-loop (one-shot) setting. Finally, it is also possible to create silent (rest) segments, and this is key to creating complex drum tracks with interesting rhythmic variations using just one wave sequence.

In the original Korg wave sequencing (now called Wave Sequencing 1.0), all of the tracks were locked together in time and controlled with the loop track so that each step applied the same pitch and amplitude to the crossfaded waveforms in the same manner: for example, the waveform in Step Two in Figure 15.4 would always be shifted +7 semitones and always play at unity gain. The Wavestation took this concept to the extreme: a wave sequence was rendered from one oscillator, and the Wavestation had four of these oscillators running in parallel, each with its own wave sequence and timing patterns, that could be mixed with the four-way vector synthesis modulation in Section 6.3, creating bass lines, drum tracks, slowly modulating pads, and a searing lead sound all in one patch.

15.5.1 Wave Sequencing 2.0

Nearly 30 years after the Wavestation's release, in January 2020, Korg released the Wavestate®, which was designed and voiced with some of the original Wavestation engineers and sound designers, and uses both wavetables and PCM samples, referring to all of them, simply, as samples. The Wavestate features what Korg calls "Wave Sequencing 2.0," which re-thinks the wave sequencer and its possibilities for generating dense, unique, and even un-repeatable patterns. The synth itself has deep modulation capabilities, some of which affect the wave sequencing, but its design and operation are outside the book's scope. In the Wavestate, the wave sequencer is divided into lanes made up of steps. There is one timing lane, which sets up the crossfade and segment duration times; a sample lane that stores the waveform for each step; lanes for modulating parameters that apply to the waveforms (pitch, amplitude, note/rest); and a step sequencer lane, also for modulating other parameters in the synth (e.g. filter f_c). In addition to the use of the name lanes, some of the fundamental additions to the 2.0 version also include:

1 The ability for each lane to operate independently with its own looping setup; all segments in the lanes are still locked to the timing lane's segment *durations*, but they may move freely on their own, loop back and forth, or jump around randomly
2 The steps in each lane, including the timing lane, have their own probability setting which randomly enables or disables them; for a non-timing step, disabling it shuts off the output value so it is not applied (note that this is different from transmitting a zero value)
3 A disabled timing step is absorbed into the previous step, elongating the segment duration and crossfade times
4 Each timing step has three operational modes: *note* (play normally), *rest* (silence), or *gate* (hold this step during the sustain portion of the amp envelope)

Figure 15.5(a) shows how the Wavestate wave sequencer is set up so that each lane has its own start, end, loop start, and loop ending points, removing the original Wavestation's step locking. The non-timing steps are shown in dotted lines because they will inherit the duration from the timing lane. In addition to amplitude and pitch, there are lanes that include shape (a contour applied as an amplitude scalar) and segment gating, where you set the amount of time the waveform plays as a percentage of the timing step's duration, which itself is powerful: since the lanes are not

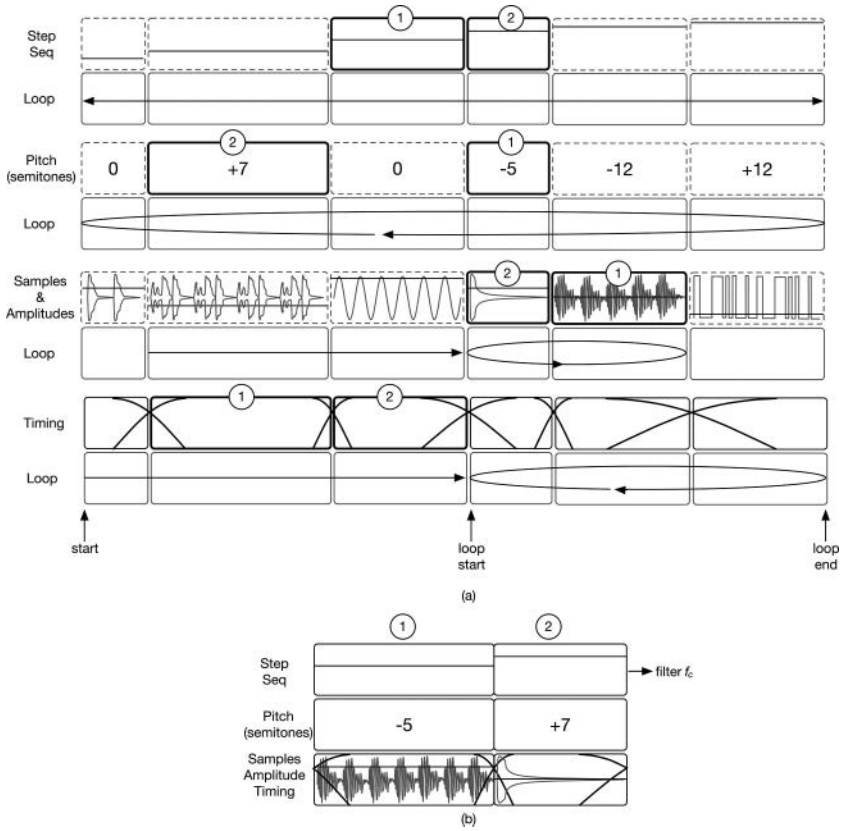


Figure 15.5 (a) The Wavestate's wave sequencer splits out each lane with its own start, stop, loop start, and loop end points (b) a possible combination of lanes showing the lane steps marked (1) and (2); the timing lane is used to synchronize the other lane step durations and crossfade timing

locked together, this step's on and off times will be applied to whatever waveform happens to be in the sample lane at the time. This also means that each segments pitch, amplitude, and contour may vary over time, and with the use of probability settings, the sequencer can generate a non-repeating stream of melodies, rhythmic patterns, and modulations during a single note-event. Due to a combination of independent lane looping and probability values, Figure 15.5(b) shows how steps labeled (1) and (2) might be combined from the various other lanes' pieces using the timing step durations and the other lane step values or waveforms. You can also see how a small number of waveforms and parameters can generate a massive number of combinations when probability is applied. Now consider this: practically every parameter for each kind of step and lane is modulate-able, and the step sequencer can even modulate its wave sequencer's own parameters. There are 16 steps in a wave sequence, and the Wavestate implements four separate wave sequencers (A, B, C, and D) per note-event.

15.6 SynthLab *WaveSequencer*

The SynthLab-WS project uses a single wave sequencer that is based on the Wavestate's multi-lane system and includes eight steps per lane. You may easily expand this to use more steps or even add another wave sequencer. For simplicity, there are a few features not included, such as the wave contour lane and lack of probability on the timing lane, but it is nonetheless a very powerful sequencer. It includes one step-sequencer lane as with the Wavestate, and you may expand that to include more lanes and hereby emulate a traditional step sequencer gone wild. Figure 15.6 shows the wave sequencer's GUI controls and interface. There are four lanes, with controls corresponding to Figure 15.5's parameters and the specifications in Section 15.5.1. There are also status lights (LEDs) to allow the user to see which steps are active in each lane at a given instant. There is a bank of global controls for each lane that has its own loop start and end points, and a loop mode control: forward, backwards, or forwards-backwards. All lanes also include a shuffle switch that randomizes the step sequence on each trip around that lane's loop points. These combine with the probability controls to create some very interesting sequences – as with the Wavestate, make sure that the DAW is in record mode all the time to catch your inventions as they happen.

15.6.1 C++ Implementation: WaveSequencer *Overview*

The *WaveSequencer* object acts as a pure modulation source and is run at the same time as the EGs and LFOs. The *SynthVoice* implements one *WaveSequencer* that is created, reset, and updated along with the other modules. The *WaveSequencer* writes numerous values into its modulation output array, which you may access via the modulation matrix. A special wave-sequencing oscillator (*WSOscillator*) maintains internal oscillators that are the targets for the wave sequencer modulation values: oscillator amplitude, waveform index, and pitch, along with the crossfading values to blend the two output waveforms during the morph. Figure 15.7(a) shows the *WaveSequencer* block diagram and member objects: *Lanes*, *LaneSteps*, and the *XHoldFader*. The *XHoldFader* contains the timers that set the hold and crossfade times, and generates the crossfade gain values. In one

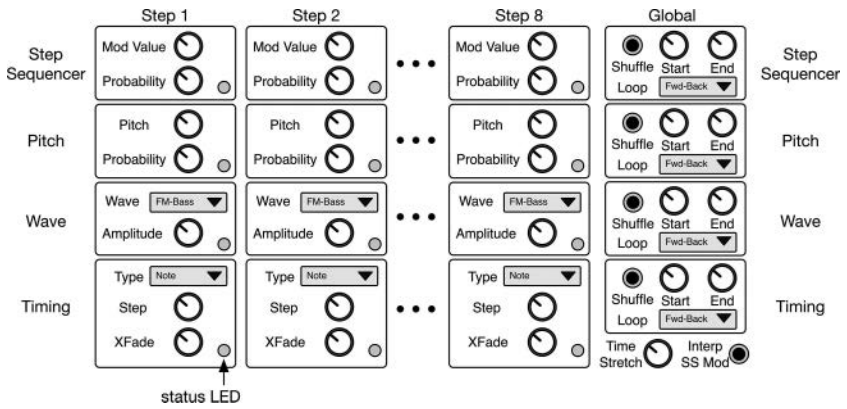


Figure 15.6 SynthLab's wave sequencer interface includes individual settings for all steps and a global control over the lanes

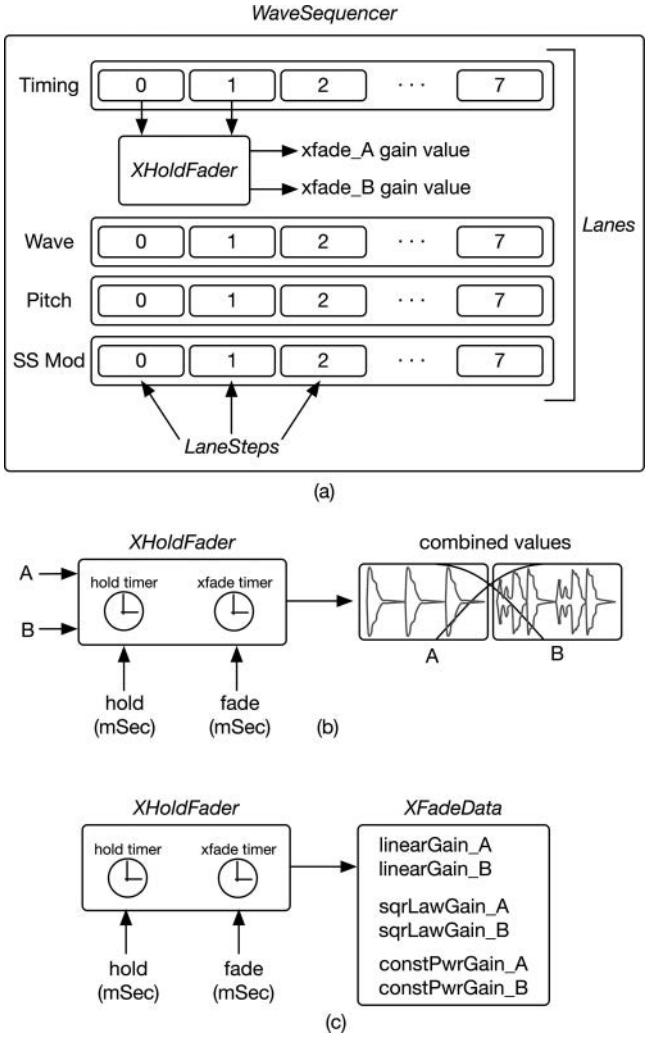


Figure 15.7 (a) The *WaveSequencer* and its member objects; (b) the *XHoldFader* can hold and crossfade signals or values, and (c) it can output the crossfade gain values in the *XFadeData* structure and operate in both modes at once, if needed

operational mode, shown in Figure 15.7(b), the *XHoldFader* performs the holding and crossfading of two input values, *A* and *B*, that may be audio signals or control values. In the second mode, in Figure 15.7(c), the object outputs the *A* and *B* gain values that an external object applies to its own signals, and it generates both linear and the two crossfade modulation gain values from Section 6.2.2 (linear, square law, and constant power). The *WaveSequencer* acts as the external object that applies the hold and crossfade gain values.

The *WaveSequencer* also provides a visual output of the current steps operating in each lane that have the ability to connect to your GUI to turn on and off as each step is activated or deactivated,

which happens when the crossfade is halfway completed. You access these metering values through the engine, which provides a set of arrays corresponding to the four lanes of eight steps. You can use that to illuminate your GUI. The stock version provides status indicators for the first voice that is running. In poly mode, each note will generate its own new sequence as each new note is triggered.

15.6.2 C++ Implementation: Setting Step and Crossfade Durations

The step and crossfade durations may be set in milliseconds or with note durations. Using note durations allows you to easily set up musically interesting patterns. In addition, there is a time stretch/shrink control that will allow you to speed up or slow down the sequence while maintaining the note duration ratios between steps and crossfades. We already saw one implementation in the LFO's *BPM Sync* mod knob control. To facilitate both the GUI controls and timing calculations, I've included a strongly typed enum and helper functions that will convert note durations to time in seconds. The *NoteDuration* enum corresponds to the GUI control you set up for your user, and you display either duration strings or GUI note graphics that are indexed according to this enum. The *kOff* enum corresponds to a step or crossfade duration of zero seconds.

```
enum class NoteDuration {
k32ndTriplet, k32nd, k16thTriplet, kDot32nd, k16th, k8thTriplet, kDot16th,
k8th, kQuarterTriplet, kDot8th, kQuarter, kHalfTriplet, kDotQuarter,
kHalf, kWholeTriplet, kDotHalf, kWhole, kDotWhole, kOff,
kNumNoteDurations
};
```

Then, the function *getTimeFromTempo* will convert the duration into a time value using the BPM value from your DAW.

```
getTimeFromTempo(double BPM, NoteDuration duration,
                 bool returnMilliseconds = false)
```

15.6.3 C++ Implementation: Holding and Crossfading

To understand the modulation values that the sequencer outputs, first consider how the timing lane operates from the very start of the note-event. Consider the wave steps and timing sequence in Figure 15.8(a). At any given time, the sequencer will either be holding the current waveform (A) or crossfading waveform A into waveform B, as shown in Figure 15.8(b). Once the crossfade is done, the first timing step is discarded, the next is loaded, wave B moves to the wave A location, and the series repeats, as shown in Figure 15.8(c) and (d). Each pair of steps goes through the same process: hold for some duration, then crossfade into the next, or hold-crossfade-hold-crossfade, etc. This produces two gain values: one for the first (A) step and one for the second (B) step. When holding, wave A's gain will always be 1.0, and after the crossfade, it will be 0.0. Then, wave B becomes the "holding" step. The wave sequencer writes the current wave A, and wave B gain values into its modulation output array.

The waves themselves are indexed to correspond to the wave sequencing oscillator's unique waveform names, and these values are also written to the modulation output array, along with the amplitude values you set on the GUI. Each oscillator may have a different pitch, and the pitch

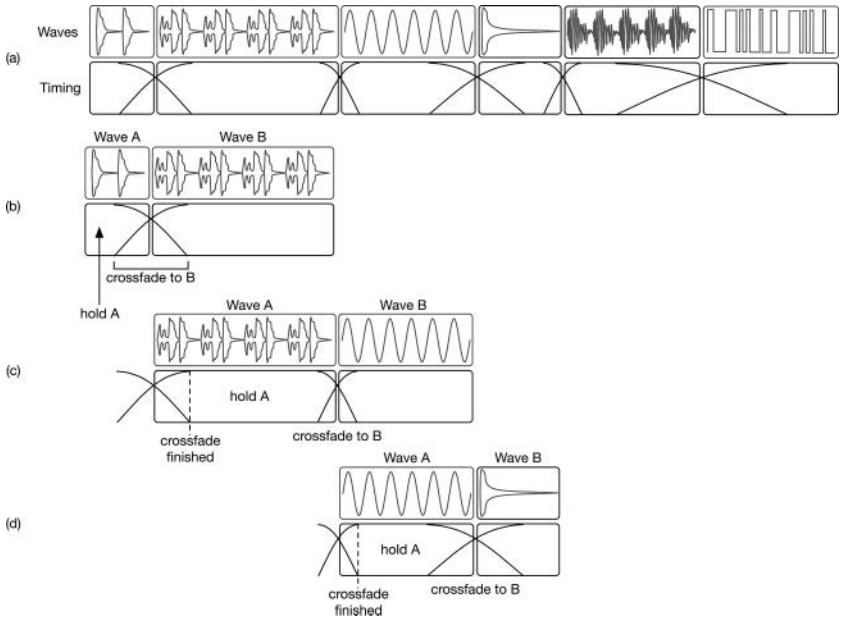


Figure 15.8 (a) A set of timing and waves to be sequenced shows the hold-then-crossfade pattern where you (b) hold wave A and crossfade to B on first step then (c) wave B is labeled as the new wave A and the next segment is loaded as wave B (d) the hold-then-crossfade process repeats with each new segment

values are also written. Finally, the step sequencer modulation value, which may be stepped or interpolated, is written to its location. If you want to add more step sequencer modulation lanes, you will need to add more slots into the modulation output array to handle their values. The enumeration for the wave sequencer's modulation output array is:

```
enum {
      kWSWaveMix _ A,
      kWSWaveMix _ B,
      kWSWaveIndex _ A,
      kWSWaveIndex _ B,
      kWSWaveAmpMod _ A,
      kWSWaveAmpMod _ B,
      kWSPitchMod _ A,
      kWSPitchMod _ B,
      kWStepSeqMod,
      kWSXFadeDone,
      kNumWSOutputs
};
```

You can see one additional output, *kWSFadeDone*, that is toggled when a new crossfade has finished. The target wave sequencing oscillator uses this indicator to swap oscillators so that oscillator B becomes oscillator A, then the next step is loaded as the new oscillator B.

15.6.4 C++ Implementation: Lanes and LaneSteps

The wave sequencer is actually relatively simple because it is really a big loop management system. The best way to understand it is to use the homework problems that will force you to examine the code and go deeper into the loop timing operations. The timing lane generates the step-timing information that is used to switch the other lane steps on and off, and they all switch steps at the same time – but each lane calculates its next step based on its own loop settings and each next step's probability value. To create a C++ solution, I made two C++ structures to encapsulate each component, as shown in Figure 15.9(a), with a *Lane* structure that maintains an array of *LaneStep* structures. The jump table is a circular buffer that contains the indexes of the lane steps for a given loop. After the first timing loop's crossfade, the steps are incremented so that the next step becomes the new current step, and a new next step is selected, as shown in Figure 14.9(b). Without randomization, the loop start and end points will cycle through the same series of steps, as shown in Figure 15.9(c), after the loop end point is hit. With randomization, once the loop end point is crossed, the *Lane* reshuffles the contents of the jump table, and that changes the sequence of the steps, as shown in Figure 15.9(d). So, the jump table sequences the lane steps, and the *Lane* delivers the current and next lane values to the sequencer.

Table 15.1 lists the C++ structures and objects used to create the *WaveSequencer*. The *LaneStep* holds information about both timing steps (step and crossfade durations) and modulation steps (pitch, wave, or step sequencer values).

The *Lane's* C++ code for setting the loops and maintaining the jump table, current step, and next step is reasonably simple, and involves creating the equivalent of a shuffle-able linked list. I also added the ability to create outside loops: if the user sets the loop start point to be lower than the loop end point, the loop will still operate but on the "outside" of the loop. If the start index is 6, and the end index is 2, then the outside loop sequence will be: 6, 7, 0, 1, 2, 6, 7, 0, 1, 2, … Each *LaneStep* keeps track of a previous step and next step index value; if these values are −1, then the *Lane* will use the jump table's value to set the next step. If the value is non-zero, then that indicates a loop start or end point. The function *updateLoopPoints* is called each time the jump table is shuffled

Table 15.1 WaveSequencer C++ objects and structures

Structure/Object	*Description*
LaneStep	Stores information about the step: step and crossfade durations (timing steps) and control values (non-timing steps), and previous and next step indexes; also includes a random number generator for probability
Lane	Stores an array of *LaneSteps* and a jump table for ordering the steps; also provides shuffling when randomness is applied
WaveSequencer	Stores a set of *Lanes*, one each for timing, wave, pitch, and step sequencer, and uses the *XHoldFader* object to set the step timing and generate the waveform gain values

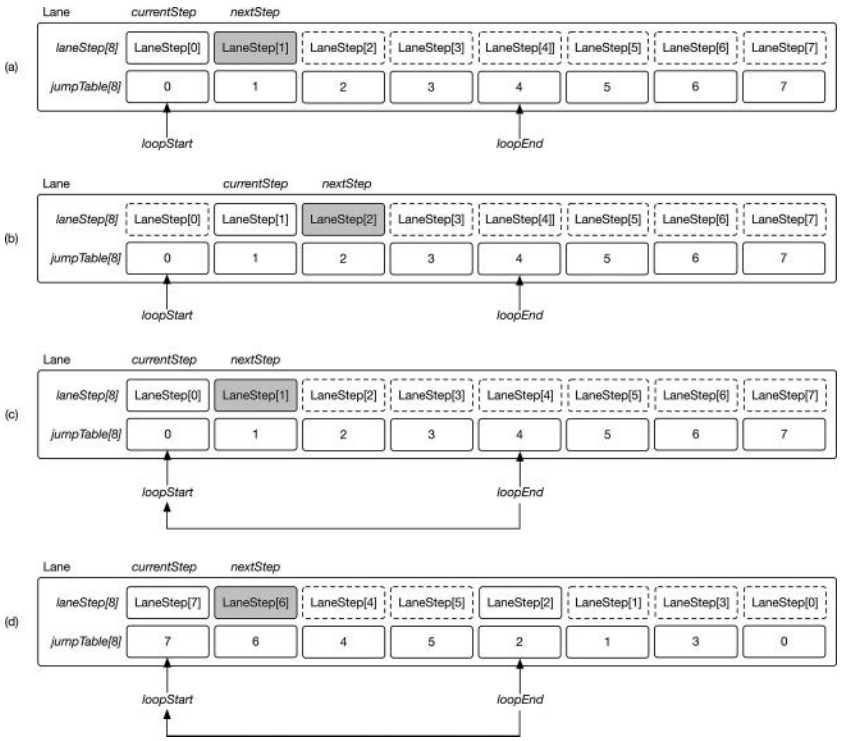


Figure 15.9 (a) A *Lane* structure maintains an array of *LaneStep* structures that are sequenced according to a jump table; the current step/next step pairs are shown in white and grey boxes (b) after the timing lane's first crossfade, the current step/next step pair advances according to the loop direction (to the right in this example) (c) without randomization, the segments will loop in the same order after the loop end point is hit (d) with randomization, the jump table is shuffled after each loop restarts which then shuffles the lane steps accordingly

and sets these next-step values to −1 or the next/previous index value; it is used in conjunction with *getNextStepIndex* to make the looping work properly. The user also sets a probability variable that is used each time the *Lane* advances to the next step. A random percentage value is generated and compared to the probability percentage value; the step is active if the random percentage value is less than or equal to the user's setting. Timing steps do not include probability. When a step is not used, its value is simply held over from the previous step's value to keep continuity in the modulation signals (Table 15.2).

15.7 The *WSOscillator* Object

The wave-sequencing oscillator is called *WSOscillator* and is a nice example that shows how to combine multiple modules. Unlike all the other SynthLab projects, SynthLab-WS does not feature a set of four oscillators that expose their waveforms, pitch controls, and mod knobs. In fact, the oscillator is somewhat hidden from the GUI, and there is no standard block diagram or module core

Table 15.2 The *WaveSequencer's SynthModule* overrides and descriptions

SynthModule Function	*Operational Description*
Reset	Sets current and next step indexes to 0; initializes a sample counter for generating status (for blinking lights)
Update	Updates each *Lane* with start/end times, step values, and probabilities; for the timing lane, it calculates the step and crossfade durations in *mSec* from the user's entry in *NoteDuration*; instructs *Lanes* to randomize sequences if user selects that option
doNoteOn	Initializes each lane with the first pair of steps, resets the status variables (for blinking lights), and resets the jump tables on each *Lane* to start from the original positions
Render	Gets next *crossfade* values from *XHoldFader* object and monitors the crossfade done flag; when finished, it advances each lane's next step. Writes the modulation values to its output array and turns the status lights on and off as each step moves to the next location
doNoteOff	Not used

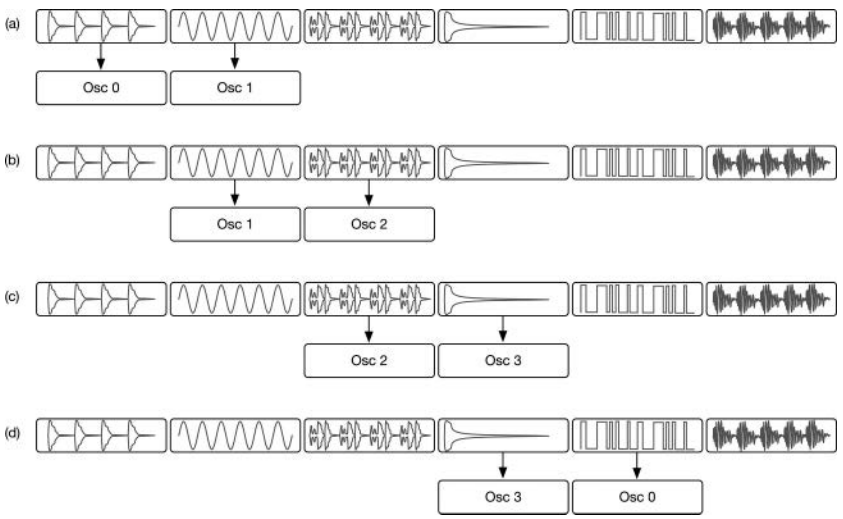


Figure 15.10 The four *WSOscillators* are used in a round-robin manner starting with (a) the oscillator pair (0, 1); after the first crossfade is completed, (b) the sequence shifts to oscillator pair (1, 2) then (c) to pair (2, 3) then (d) wrapping around to pair (3, 0), and so on

strings to reveal as in the other oscillators. The *WSOscillator* contains several underlying *WTOscillator* objects, each of which exposes up to four module cores that may render 16 waveforms each. This provides up to 64 waveforms for the user to select, including the morphing wavetable bank names, in the wave sequencer's GUI, so there is plenty of variation available.

As you can tell from Figures 15.7 to 15.9, there are two oscillators running at any given time, one for the A waveform and the other for the B waveform – unlike wave morphing that occurs on the input to a single oscillator, this morphing occurs on the outputs of two oscillators that may have different pitches as well as waveforms. However, all of the other SynthLab projects feature

four oscillators, and the modulation matrix is set up with that paradigm. In order to allow you to modulate the underlying oscillators, the *WSOscillator* implements four member oscillators as well but only uses two of them at a time in a round-robin fashion, as shown in the sequence of Figure 15.10(a) through (d). Notice that this is identical to the way in which the waveforms are held and crossfaded in Figure 15.8; the oscillators switch when the waveforms switch, and this is why the *WaveSequencer* outputs an indicator that the crossfade has occurred.

15.7.1 WSOscillator Members

The *WSOscillator* maintains four *WTOscillators* as well as four *WTOscParameter* structures, one for each oscillator. These parameter structures are used to set the oscillator amplitudes in dB and the pitch offsets in semitones as the sequencer updates these values on each block processing cycle. To facilitate the round-robin behavior, an active oscillator array holds the two currently running oscillator index values. The *WSOscillator* member variables and descriptions are listed in Table 15.3.

Table 15.4 lists the *WSOscillator's SynthModule* member function overrides and describes how they operate.

Table 15.3 The *WSOscillator* members and descriptions

WSOscillator Member	*Operational Description*
waveSeqOsc[4]	The four oscillator members
waveSeqParams[4]	Parameter structures for each oscillator
activeOsc[2]	Array containing the indexes of the two currently active oscillators
oscMixCoeffs[4]	Hold or crossfade gain values for the oscillators; only two are used at a time
vector<WaveStringData> waveStringFinder	Set of structures used to connect the 64 possible waveforms to their module cores

Table 15.4 The *WSOscillator's SynthModule* overrides and descriptions

SynthModule Function	*Operational Description*
Reset	Sets current and next step indexes to 0; initializes a sample counter for generating status (for blinking lights)
Update	Updates each *Lane* with start/end times, step values, and probabilities; for the timing lane, it calculates the step and crossfade durations in *mSec* from the user's entry in *NoteDuration*; instructs *Lanes* to randomize sequences if user selects that option
doNoteOn	Initializes each lane with the first pair of steps, resets the status variables (for blinking lights), and resets the jump tables on each *Lane* to start from the original positions
Render	Gets next *crossfade* values from *XHoldFader* object and monitors the crossfade done flag; when finished, it advances each lane's next step. Writes the modulation values to its output array and turns the status lights on and off as each step moves to the next location
doNoteOff	Not used

15.8 *WSOscillator* Programming Notes

Like the *FMOperator*, the *WSOscillator* maintains a set of modules and orchestrates their use. This is a great object to study as it shows how one module can create and control numerous member modules. The only sticky detail here is how the waveform list is populated. All *SynthModules* include two built-in functions for populating the dynamic lists. The *getModuleStrings* function returns the module strings for a given core, and the function *getAllModuleStrings* loops through all four cores and compiles a std::vector of all the strings. Your plugin framework calls the function *getAllModuleStrings* in order to populate the wave sequencer's waveform lists. The *WSOscillator* includes a function that performs the same looping as *getAllModuleStrings* in order to populate its own vector of *WaveStringData* structures, which include the module core index and corresponding waveform index that maps to each of the 64 waveform strings that the user chooses. The *WSOscillator* can then easily find the core and waveform index for each wave step. You can find the code for this in *WSOscillator::makeWaveStringMap*.

15.8.1 WSOscillator Construction Phase

The wave-sequencing oscillator creates its member oscillators, along with corresponding parameter structures, in the constructor to control each object. The morphing wavetable cores use mod knob D to control the morph intensity, and we want to hard code that to 1.0 or full intensity as the default value. The code for setting up the parameter structures and instantiating the oscillators is below – notice how each oscillator has its own parameter structure.

```
for (uint32_t i = 0; i < NUM_WS_OSCILLATORS; i++)
{
     waveSeqParams[i].reset(new(WTOscParameters));
     waveSeqParams[i]->modKnobValue[0] = 0.5;
     waveSeqParams[i]->modKnobValue[1] = 0.0;
     waveSeqParams[i]->modKnobValue[2] = 0.0;
     waveSeqParams[i]->modKnobValue[3] = 1.0; // morph intensity
}

// --- four wavetable oscillators
waveSeqOsc[0].reset(new WTOscillator(_midiInputData, waveSeqParams[0],
                    _waveTableDatabase, blockSize));
waveSeqOsc[1].reset(new WTOscillator(_midiInputData, waveSeqParams[1],
                    _waveTableDatabase, blockSize));
waveSeqOsc[2].reset(new WTOscillator(_midiInputData, waveSeqParams[2],
                    _waveTableDatabase, blockSize));
waveSeqOsc[3].reset(new WTOscillator(_midiInputData, waveSeqParams[3],
                    _waveTableDatabase, blockSize));
```

15.8.2 WSOscillator Reset Phase

The reset phase involves creating the waveform map; initializing the member variables; and, most importantly, forwarding the *reset* function call to the member oscillators. Notice how the *activeOsc* array is initialized with (0, 1) – these are the first two oscillators that will be used in the round-robin loop.

```
// --- create string map
makeWaveStringMap();
for (uint32_t i = 0; i < NUM_WS_OSCILLATORS; i++)
{
     oscMixCoeff[i] = 0.0;
     waveSeqOsc[i]->reset(_sampleRate);
}

// --- initial pair of oscillators
activeOsc[0] = 0;
activeOsc[1] = 1;
```

15.8.3 WSOscillator Note-On & Note-Off Phases

The *doNoteOn* function cycles through each core of the four member oscillators and simply forwards the *doNoteOn* function call to each, synchronizing all cores at once. Note the Boolean flag that is set at the end of the function, which will initialize the round-robin sequence. There is nothing to do for the note-off phase, so that function is empty.

```
for (uint32_t core = 0; core < NUM_MODULE_CORES; core++)
{
     for (uint32_t i = 0; i < NUM_WS_OSCILLATORS; i++)
     {
          waveSeqOsc[i]->selectModuleCore(core);
          waveSeqOsc[i]->doNoteOn(noteEvent);
     }
}

initRoundRobin = true;
```

15.8.4 WSOscillator Update Phase

The oscillator includes two helper functions, called *setNewOscWaveA* and *setNewOscWaveB*, that greatly simplify the *update* function. The *setNewOscWaveA* function demonstrates how the *waveStringFinder* is used to load the correct oscillator core and set the waveform index it will use during its own update handler. The wave sequencer's modulation outputs have been hard-wired to the *WSOscillator's* modulation inputs during the voice construction. The code for connecting the wave mix A value is shown below.

```
modMatrix->addModSource(kSourceWSWaveMix_A,
waveSequencer->getModulationOutput()->getModArrayPtr(kWSWaveMix_A));

modMatrix->addModDestination(kDestOsc1_WSWaveMix_A,
wsOsc1->getModulationInput()->getModArrayPtr(kWaveSeqWave_AGainMod));

parameters->modMatrixParameters->setMM_HardwiredRouting(
                          kSourceWSWaveMix_A, kDestOsc1_WSWaveMix_A);
```

The *setNewOscWaveA* function fetches the sequencer's pitch and amplitude values for waveform A and uses the parameter structure to set the information in the oscillator. The *setNewOscWaveB* function uses the wave B index, pitch, and amplitudes, and calls the *doNoteOn* handler on the new oscillator in case the user selected a one-shot drum or sound effect for that step.

```
void setNewOscWaveA(uint32_t oscIndex, uint32_t waveAIndex,
                    double oscAMixCoeff)
{
     // --- choose oscillator core
     waveSeqOsc[oscIndex]->selectModuleCore
          (waveStringFinder[waveAIndex].coreIndex);

     // --- then set the oscillator waveform index
     waveSeqParams[oscIndex]->wavetableIndex =
          waveStringFinder[waveAIndex].coreWaveIndex;

     // --- next set the pitch and amplitude from the sequencer
     waveSeqParams[oscIndex]->oscSpecificDetune =
          getModulationInput()->getModValue(kWaveSeqPitch_AMod);

     waveSeqParams[oscIndex]->outputAmplitude_dB =
          getModulationInput()->getModValue(kWaveSeqAmp_AMod);

     //--- store the mix coefficient for render
     oscMixCoeff[oscIndex] = oscAMixCoeff;
}
```

The *update* function starts by fetching the waveform index and hold/crossfade mix values from its modulation input:

```
wave_AIndex = getModulationInput()->getModValue
                                          (kWaveSeqWaveIndex_AMod);

wave_BIndex = getModulationInput()->getModValue
                                          (kWaveSeqWaveIndex_BMod);

oscAMixCoeff = getModulationInput()->getModValue
                                          (kWaveSeqWave_AGainMod);

oscBMixCoeff = getModulationInput()->getModValue
                                          (kWaveSeqWave_BGainMod);
```

If this is the very first update, the two oscillators are set up with the helper function, and then the update function is done.

```
if (initRoundRobin)
{
     setNewOscWaveA(activeOsc[0], wave_AIndex, oscAMixCoeff);
     setNewOscWaveB(activeOsc[1], wave_BIndex, oscBMixCoeff);
```

```
        initRoundRobin = false;
        return true;
}
```

All subsequent *update* function calls will then perform two steps. First, wait for the crossfade-done signal from the sequencer; if the crossfade is done, the *activeOsc* array is incremented and wrapped to implement the round-robin behavior. Then, the helper functions are used to set the new oscillator data.

```
xfadeDone = getModulationInput()->getModValue(kWaveSeqXFadeDoneMod) ==
                                                        0 ? false : true;
if (xfadeDone)
{
        // --- rotate to next pair of oscillators
        if (++activeOsc[0] >= NUM_WS_OSCILLATORS) activeOsc[0] = 0;
        if (++activeOsc[1] >= NUM_WS_OSCILLATORS) activeOsc[1] = 0;

        // --- set wave index and mix coeff
        setNewOscWaveA(activeOsc[0], waveA_Index, oscAMixCoeff);
        setNewOscWaveB(activeOsc[1], waveB_Index, oscBMixCoeff);
        return true;
}
```

Finally, if we are in the middle of a hold or crossfade operation, just set the updated pitch and amplitude information using the oscillator parameter structures.

```
// --- just forward the settings: OSC 0
waveSeqParams[activeOsc[0]]->oscSpecificDetune =
            getModulationInput()->getModValue(kWaveSeqPitch_AMod);

waveSeqParams[activeOsc[0]]->outputAmplitude_dB =
            getModulationInput()->getModValue(kWaveSeqAmp_AMod);

oscMixCoeff[activeOsc[0]] = oscAMixCoeff;

// --- just forward the settings: OSC 1
waveSeqParams[activeOsc[1]]->oscSpecificDetune =
            getModulationInput()->getModValue(kWaveSeqPitch_BMod);

waveSeqParams[activeOsc[1]]->outputAmplitude_dB =
            getModulationInput()->getModValue(kWaveSeqAmp_BMod);

oscMixCoeff[activeOsc[1]] = oscBMixCoeff;
```

15.8.5 WSOscillator Render Phase

The rendering operation is very simple – after updating, call the render method on the two active oscillators, then mix their audio output arrays into the *WSOscillator's* output array.

```
// --- update parameters for this block
update();

// --- render each osc
waveSeqOsc[activeOsc[0]]->render(samplesToProcess);
waveSeqOsc[activeOsc[1]]->render(samplesToProcess);

// --- mix output buffers into our buffer
getAudioBuffers()->flushBuffers();

// --- add osc 0
mixOscBuffers(waveSeqOsc[activeOsc[0]]->getAudioBuffers(),
                          samplesToProcess, oscMixCoeff[activeOsc[0]]);

// --- mix osc 1
mixOscBuffers(waveSeqOsc[activeOsc[1]]->getAudioBuffers(),
                          samplesToProcess, oscMixCoeff[activeOsc[1]]);
```

15.9 Exercises

15.9.1 Wave Sequencing

First, download the Korg Wavestate user manual from www.willpirkle.com\Downloads\wavestate.pdf. There are several features missing from my *WaveSequencer* object, and you should try implementing them yourself. This will force you to go deeper into the *WaveSequencer* code and understand its looping operations as well as manipulate the four round-robin oscillators in the *WSOscillator* object.

15.9.2 Crossfade Selection

The crossfades in my sequencer object are always set to constant power for crossfading waveforms and linear for crossfading the step sequencer's values. Allow the user to choose the type of crossfade they want for the waveform and the step sequencer, and alter the code to apply it. HINT: the *XHoldFader* object outputs the gain values for all three options at once (Figure 15.7(c)).

15.9.3 Fade-In and Fade-Out control

The Wavestate allows the user to manipulate both the fade-in and fade-out curvature independently for the waveform crossfade. Modify the sequencer and *XHoldFader* to allow for the manipulation of both fade-in and fade-out curvature using the two GUI control knobs that set the curvature values. Use the Wavestate manual as your guide. HINT: One easy way to provide a curvature control is to simply combine the *XHoldFader's* linear gain coefficients with the square law coefficients in a ratio that corresponds to the user's curvature control setting.

15.9.4 Timing Probability

My *WaveSequencer* does not include a probability control for the timing steps. Use the Wavestate manual to understand how this probability value works – when probability causes a timing step to be skipped, its step duration is combined into the previous step, and the crossfade point is moved out accordingly. Implementing this probability will then generate numerous different sequences that are locked to the DAW BPM but will vary in step durations.

15.9.5 Timing Swing

Use the Wavestate manual to see how Korg implements timing swing by delaying the onset of every other step in the sequence, and add a swing control to the other sequencer global GUI components.

Bibliography

Korg.com. 2014 "Electribe Wave Owner's Manual." https://www.korg.com/us/support/download/product/0/797/#manual, Accessed October 14, 2020

Korg.com. 2016. "iWavestation Wave Owner's Manual." https://www.korg.com/us/support/download/manual/0/739/3510/, Accessed October 14, 2020

Korg.com. 2019. "Wavestate Owner's Manual." https://www.korg.com/us/support/download/product/0/840/, Accessed October 14, 2020

16 The SynthLab Synth Projects

The SynthLab engine and voice architectures are fundamentally identical across all of the projects, as shown in Figure 16.1. The projects' quad oscillator banks are different, but everything else remains the same. The wave sequencing SynthLab-WS includes the additional *WaveSequencer* object, but it also uses four oscillators in its implementation.

Table 16.1 lists the projects and the type of oscillators that are used for the quad oscillator core.

16.1 SynthLab Modulation Matrix

All SynthLab projects use the same modulation matrix, except the wave sequencing synth, which adds an additional modulation source row, as shown in Figure 16.2. The sources on the left column are the main synth modulators, and you can add many more rows if you like since the LFOs and EGs have multiple outputs. The destinations across the top are common to many synth designs. Notice how I save space by having two destinations that apply to OSC1, OSC2, and OSC3 simultaneously. If you make oscillator #4 a sub-oscillator, playing an octave down, you may not want to apply the same modulation or intensity so that oscillator has its own destination columns.

Notice the two destination columns for "OSC mod" – this is a catchall for each oscillator's ability to provide a special modulation that works with its algorithm. Table 16.2 lists the oscillators and their special modulation destinations.

16.1.1 Modulation Matrix Programming

Mod matrix programming is covered in Chapter 14. Make sure you examine the *SynthVoice* constructors for each project to see the mod matrix programming operations. If you are using multiple cores, notice how the voice object enables and disables the mod matrix routings as different cores are loaded. Here is the code for selecting a new wavetable oscillator #1 core; notice how the old modulation destinations are removed, and the new destinations are added. This does not require any dynamic memory allocations.

```
wtOsc1->selectModuleCore(index);


// --- reset mod matrix pointers to new core modulation arrays
modMatrix->clearModDestination(kDestOsc1 _ fo);
```

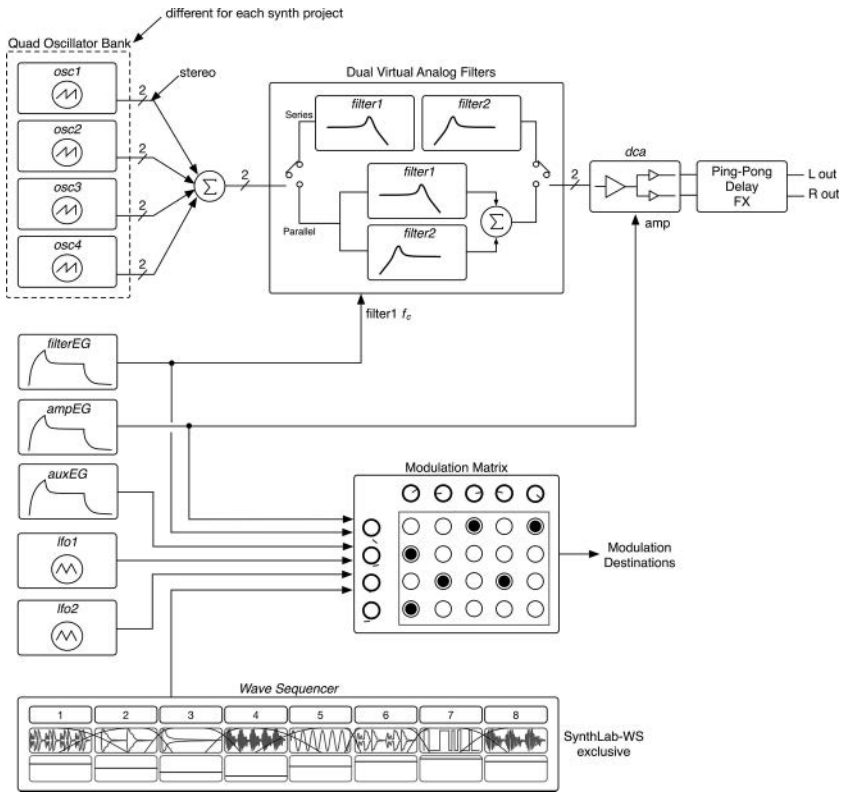


Figure 16.1 SynthLab voice architecture; the quad oscillator bank is different for each project, and the *WaveSequencer* is only in SynthLab-WS

Table 16.1 SynthLab projects and oscillators

SynthLab Project	*Oscillators*	*Notes*
Wavetable (WT)	*WTOscillator*	May freely mix normal, one-shot, sound effect, and morphing wavetables
Virtual analog (VA)	*VAOscillator*	Oscillators based on the Oberheim SEM® that output both square and saw in a user-controlled ratio
Sample based (PCM)	*PCMOscillator*	PCM samples stored in *.wav* files
Physical model (KS)	*KSOscillator*	Oscillators can be augmented with additional resonators (e.g. 12-string emulation)
Chowning FM (DX)	*FMOperator*	The four FM operators require significantly more GUI controls due to their built-in DX EGs
Wave-sequencing (WS)	*WSOscillator* that aggregates four *WTOscillators* and is modulated with a *WaveSequencer*	May freely mix normal, one-shot, sound effect, and morphing wavetables while wave sequencing at the same time

Table 16.2 SynthLab oscillators and their special modulation destinations

Oscillator	*C++ Object*	*Special Modulation*
Wavetable	*ClassicWTCore*	Oscillator shape
Morphing wavetable	*MorphWTCore*	Morphing index (added to other modulation sources)
Virtual analog	*VAOCore*	Square wave duty cycle
Sample based	All *PCMCores*	Pan modulation (per oscillator)
Plucked string model	*KSOCore*	Pluck position

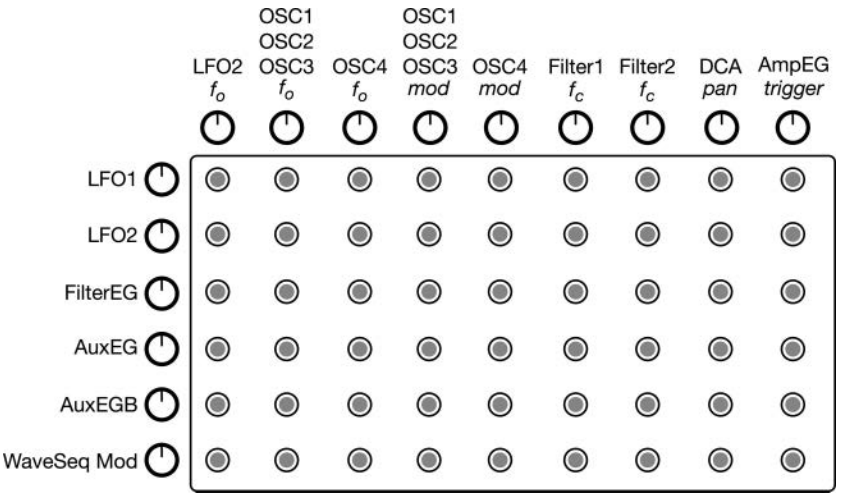


Figure 16.2 The SynthLab modulation matrix uses a pin-programmer style interface and source and destination intensity controls

```
modMatrix->clearModDestination(kDestOsc1_Mod);
modMatrix->clearModDestination(kDestOsc1_Morph);

modMatrix->addModDestination(kDestOsc1_fo,
     wtOsc1->getModulationInput()->getModArrayPtr(kBipolarMod));

modMatrix->addModDestination(kDestOsc1_Mod,
     wtOsc1->getModulationInput()->getModArrayPtr(kUniqueMod));

modMatrix->addModDestination(kDestOsc1_Morph,
     wtOsc1->getModulationInput()->getModArrayPtr(kWaveMorphMod));
```

16.2 SynthLab Render Phase: Modulators

All SynthLab projects have the same initial code for the voice *render* method, which first renders the modulator outputs, then runs the modulation matrix. SynthLab-WS runs its wave sequencer object along with the other modulators and prior to the modulation matrix. To facilitate oscillator

mixing, the voice has its own sub-mix buffers that it uses to accumulate each oscillator's output buffer. This buffer must be cleared so that the first accumulation mixes with the cleared (0.0) values.

```
bool SynthVoice::render(SynthProcessInfo& synthProcessInfo)


uint32_t samplesToProcess = synthProcessInfo.getSamplesInBlock();

// --- clear for accumulation
mixBuffers->flushBuffers();

// --- update/render (add more here)
lfo1->render(samplesToProcess);
lfo2->render(samplesToProcess);

ampEG->render(samplesToProcess);
filterEG->render(samplesToProcess);
auxEG->render(samplesToProcess);

// --- sequencer generates modulation values
waveSequencer->render(samplesToProcess); // <- SynthLab-WS only!!

// --- do all mods
modMatrix->runModMatrix();
```

16.3 SynthLab Render Phase: Quad Oscillator Mixing

All synths except SynthLab-WS blend four oscillators together to render the synthesized audio signal. SynthLab-WS only uses a single *WSOscillator*, but that oscillator blends the outputs of two of its four internal oscillators in round-robin format, as detailed in Section 15.7. After the modulation matrix renders its values, the four oscillators are rendered and accumulated in series; the accumulation function accepts a scalar mixing-coefficient that prevents overflow and is set to 0.25 here since we have four oscillators.

```
wtOsc1->render(samplesToProcess);
accumulateToMixBuffer(wtOsc1->getAudioBuffers(), samplesToProcess,
                      0.25);

wtOsc2->render(samplesToProcess);
accumulateToMixBuffer(wtOsc2->getAudioBuffers(), samplesToProcess,
                      0.25);

wtOsc3->render(samplesToProcess);
accumulateToMixBuffer(wtOsc3->getAudioBuffers(), samplesToProcess,
                      0.25);
```

```
wtOsc4->render(samplesToProcess);
accumulateToMixBuffer(wtOsc4->getAudioBuffers(), samplesToProcess,
                      0.25);
```

SynthLab-WS uses only a single oscillator, and to simplify the filtering code that follows, it accumulates into the same mix buffer but with a mixing coefficient of 1.0.

```
// --- wave sequencer
wsOsc1->render(samplesToProcess);
accumulateToMixBuffer(wsOsc1->getAudioBuffers(), samplesToProcess,
                      1.0);
```

16.4 SynthLab Render Phase: Filtering

All synths include a pair of *SynthFilter* objects that may be run in series or parallel. A GUI controlled parameter switches between modes. This code takes advantage of the audio buffer mixing functions in *synthfunctions.h* to perform the series/parallel operation. The first step is to copy the oscillator mix buffers into either *filter1* (series) or both filters (parallel):

```
// --- setup filtering
if (parameters->filterModeIndex == enumToInt(FilterMode::kSeries))
{
     // --- to Filter1
     copyBufferToInput(mixBuffers, filter1->getAudioBuffers(),
                       STEREO _ TO _ STEREO, samplesToProcess);
}
else
{
     // --- to Filter1
     copyBufferToInput(mixBuffers, filter1->getAudioBuffers(),
                       STEREO _ TO _ STEREO, samplesToProcess);

     // --- to Filter2
     copyBufferToInput(mixBuffers, filter2->getAudioBuffers(),
                       STEREO _ TO _ STEREO, samplesToProcess);
}
```

Next, the filters are rendered in series, where *filter1's* rendered output buffer is copied to *filter2's* input and then rendered, or parallel, where both filters are rendered and outputs are accumulated in the mix buffer. In both cases, the last step is to copy the buffers to the output DCA.

```
if (parameters->filterModeIndex == enumToInt(FilterMode::kSeries))
{
     // --- update and render
     filter1->render(samplesToProcess);

     // --- to Filter2
     copyOutputToInput(filter1->getAudioBuffers(),
```

```
                        filter2->getAudioBuffers(),
                        STEREO _ TO _ STEREO, samplesToProcess);

      // --- update and render
      filter2->render(samplesToProcess);

      // --- to DCA
      copyOutputToInput(filter2->getAudioBuffers(),
                        dca->getAudioBuffers(),
                        STEREO _ TO _ STEREO, samplesToProcess);
}
else
{
      // --- flush
      mixBuffers->flushBuffers();

      // --- render and accumulate
      filter1->render(samplesToProcess);
      accumulateToMixBuffer(filter1->getAudioBuffers(),
                            samplesToProcess, 0.5);

      // --- update and render
      filter2->render(samplesToProcess);
      accumulateToMixBuffer(filter2->getAudioBuffers(),
                            samplesToProcess, 0.5);

      // --- to DCA
      copyBufferToInput(mixBuffers, dca->getAudioBuffers(),
                        STEREO _ TO _ STEREO, samplesToProcess);
}
```

The DCA processes the audio and provides gain and panning, and the final mix is copied into the *SynthProcessInfo* structure that the engine provided during the *render* function call. After that, the small piece of logic is run to see if the *ampEG* has expired and the note-event has ended. This is part of the note-lifecycle and voice-stealing operation.

```
// --- update and render
dca->render(samplesToProcess);

// --- to mains
copyOutputToOutput(dca->getAudioBuffers(), synthProcessInfo,
                   STEREO _ TO _ STEREO, samplesToProcess);
```

16.5 SynthLab Render Phase: Global Volume and Delay FX

The engine code is identical across all SynthLab projects and discussed in detail in Chapter 2. After rendering the active voices, the engine's last step is to mix them and apply the delay FX and

final global volume control. This code also shows how the wave sequencer status meter values are parsed for the first voice only (otherwise, metering would be very confusing for the user).

```
// --- blend active voices
if (synthVoices[i]->isVoiceActive())
{
     // --- render and accumulate
     synthVoices[i]->render(voiceProcessInfo);
     accumulateVoice(synthProcessInfo, gainFactor);

     // --- only show first voice , WS only
     if (i == 0)
          parameters->wsStatusMeters =
          parameters->voiceParameters->
          waveSequencerParameters->statusMeters;
}
```

The ping-pong delay FX is also derived from *SynthModule*, so it behaves the same as the other processor objects that process an input to an output (DCA and filters). The algorithm is straight out of my FX plugin book. The engine owns the *std::unique_ptr* to the ping-pong delay and treats it the same way the voice treats the filters using the input and output audio buffers that all *SynthModules* expose. The global volume control is the last part of the engine processing and is done with another helper function that simply iterates through the output buffer and applies the scaling (volume control) factor.

```
if (parameters->enableDelayFX)
{
     // --- copy synth output to delay input
     copySynthOutputToAudioBufferInput(synthProcessInfo,
                                       pingPongDelay->getAudioBuffers(),
                                       STEREO_TO_STEREO, samplesToProcess);

     // --- run the delay
     pingPongDelay->render(samplesToProcess);

     // --- copy to output
     copyAudioBufferOutputToSynthOutput(
                                       pingPongDelay->getAudioBuffers(),
                                       synthProcessInfo, STEREO_TO_STEREO,
                                       samplesToProcess);
}

// --- add master volume
applyGlobalVolume(synthProcessInfo);
```

16.6 SynthLab-DX: the FM Synthesizer

Yamaha produced and marketed the first commercially available FM synthesizer, the DX-7, in 1983, based off of John Chowning's seminal paper *The Synthesis of Complex Audio Spectra by*

Means of Frequency Modulation, published a decade prior. The DX-7 became the second-best-selling synthesizer of all time at about 160,000 units sold (not far behind the Korg M1 at about 250,000 units). Yamaha spun off the product into multiple DX synths, all marketed as FM synths. The manuals refer to this method as "digital FM tone generation." However, neither the DX-7 nor any of the DX variants were FM synths; instead, they were all Phase Modulation (PM) synthesizers. PM is virtually identical to frequency modulation so that much of the theory is interchangeable. FM and PM synthesis require accurate sinusoidal oscillators that can run forwards or backwards; all SynthLab oscillators except the Karplus-Strong variant have this capability. FM (and PM) synthesis are capable of producing a vast range of timbres, from searing, paint-peeling lead sounds to muted piano sounds to the most convincing bell and gong sounds you will probably ever synthesize without using samples.

16.6.1 FM and PM Basics

Both frequency and phase modulation theory are very well documented, and the mathematical concepts were around long before Chowning's 1973 paper. FM and PM synthesis generate dynamic and interesting waveforms using just two sinusoidal oscillators: the modulator and the carrier. In FM, the modulator waveform varies the carrier oscillator's instantaneous frequency by adding and subtracting its own frequency, as shown in Figure 16.3(a). In PM, the modulator waveform varies the carrier oscillator's instantaneous phase by adding and subtracting a phase offset (φ) in Figure 16.3(b), which, in turn, varies the frequency. You can see that the resulting time domain audio signals look very similar but phase-shifted from one another. Taking the magnitude FFT produces the frequency spectrum, and in both cases, that spectrum consists of the carrier frequency plus and minus multiples of the modulator frequency: $f_c + f_m, f_c - f_m, f_c + 2f_m, f_c - 2f_m$, and so on. These additional spectral components are called the sideband frequencies or sidebands. From Figure 16.3, you can see three important features about the sidebands:

1 The sidebands may be negative frequencies
2 The amplitudes of the sidebands don't appear to follow a pattern other than being symmetrical across the carrier frequency
3 A sideband component may occur at 0 Hz or DC

When the sidebands are negative frequencies, they reflect across the 0 Hz axis, flip in phase, and add back into the positive frequency domain's components. The reflected frequencies may line up with existing spectral components, in which case they may add or subtract from them, or they may land in-between existing frequencies. The amplitudes of the sideband harmonics, which may first appear to modulate randomly, actually follow a set of mathematical equations called Bessel functions; they produce a specific kind of undulation (up-down motion), but predicting how they will move and sound is difficult without copious experience in playing and listening to the results.

For synth implementation, the most important of the three features is that PM does not produce sidebands at DC or 0 Hz. Another issue is that for FM, the carrier oscillator needs to know the frequency of the modulator so that it may add and subtract this value to modulate the oscillation frequency; this is not required for PM, which simply offsets the current *phaseInc* value of the synth clock by adding or subtracting some amount. The DC offset problem is a deal-breaker for FM as a synthesis technique. For a single modulator-carrier pair, the DC offset may cause the output signal to move outside the bounds of [−1, +1], which creates distortion and will add a DC offset to the audio signal. More importantly, when cascading multiple modulator-carrier pairs, a DC offset

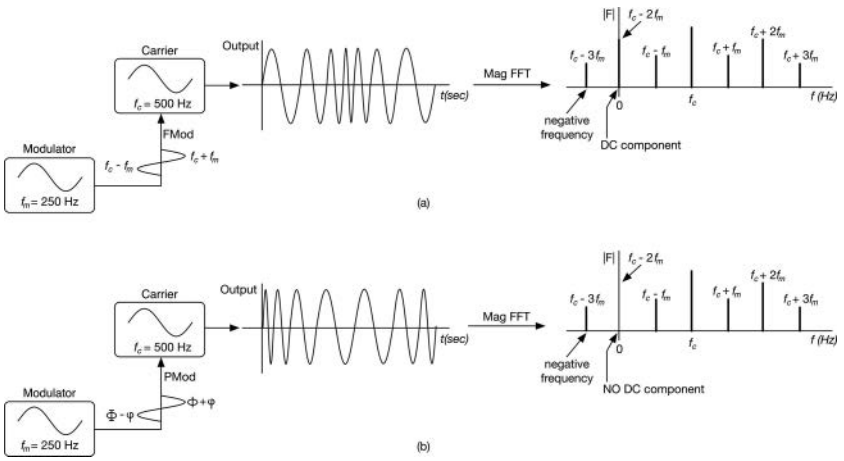


Figure 16.3 (a) FM and (b) PM time and frequency domain output signals; the spectral amplitudes shown are for visual reference; the actual amplitudes follow Bessel functions and are more complex

at the output of the first pair will produce a constant detuning offset on the second pair that will cause the note to play sharp or flat – and that is unacceptable.

16.6.2 Index of Modulation

When we use pitch modulation with an LFO modulating the oscillator frequency in semitones, we usually include a modulation depth control – either the LFO output control, the modulation matrix intensity control, or a combination of both. You can do the same thing with FM or PM and increase or decrease this modulation "strength" to likewise affect the amount of the resulting FM or PM. Here, the depth of modulation is called the "index of modulation" (*I*), and this control has a profound effect on the amount of sideband harmonics that are generated. Figure 16.4(a) shows how a modulator-carrier pair is connected for FM or PM synthesis. The carrier oscillator's frequency is the MIDI note pitch. The modulator's frequency is some ratio of that and set with a GUI "ratio" control. The modulator output is adjusted with a modulation index control, and as the index increases, so do the harmonics. In Figure 16.4(b), the modulation index control is replaced with an envelope generator that dynamically changes the index value as the note-event progresses, which, in turn, alters the harmonics in a dynamic manner that resembles a strange filter that adds or removes sideband harmonics as the index changes, producing interesting timbral shifts in the output. Notice how the amplitudes of the individual sideband components undulate up and down as the index changes. In this case, the maximum index value is 4.0; this is called the I_{MAX} value.

16.6.3 FM Operators

"FM operator" is the name Chowning gives to the combination of a sinusoidal oscillator and an output amplitude EG, and he has a specific way of diagramming various operator combinations called algorithms. Figure 16.5(a) shows the FM operator and the square symbol, while Figure 16.5(b) shows how two operators are connected as modulator (operator 2) and carrier (operator 1). Chowning's algorithms allow the last modulator in a series connection to modulate itself.

SynthLab's *FMOperator* object consists of a sinusoidal wavetable oscillator paired with a DX-EG. To create the patches, you connect the audio outputs and phase modulation inputs according to the algorithm diagram.

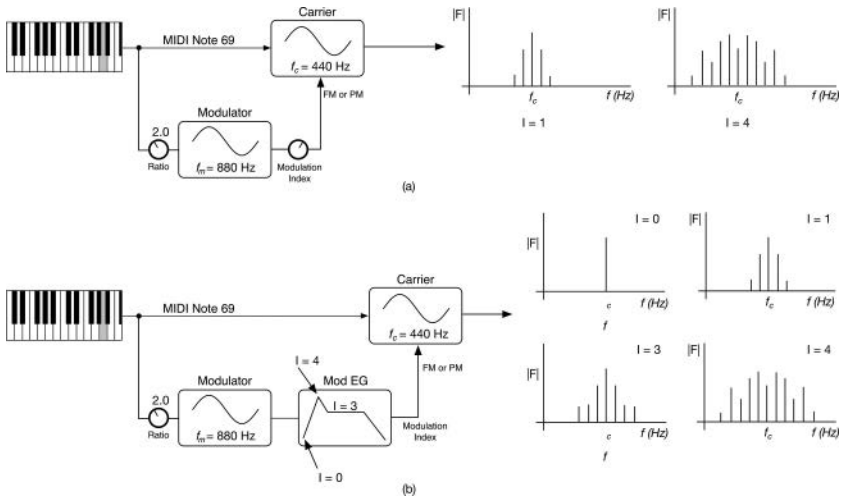


Figure 16.4 (a) A modulation index control modifies the output spectrum; (b) replacing the control with an EG allows the spectrum to morph as the note-event progresses

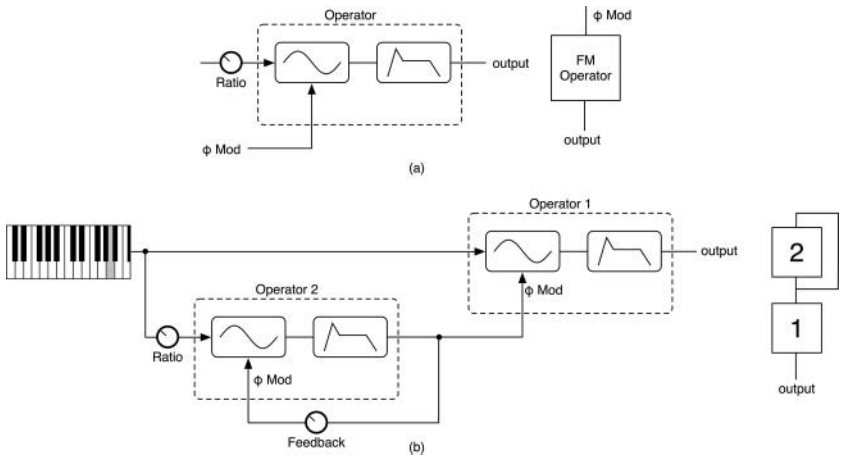


Figure 16.5 (a) An FM operator consists of a sinusoidal oscillator and EG that adjusts its output amplitude along with Chowning's FM operator notation (b) a two-operator patch that includes self-modulation; notice that Operator 1's output EG acts as the amp EG since it is the very last in the chain and Chowning's notation that indicates self-modulation

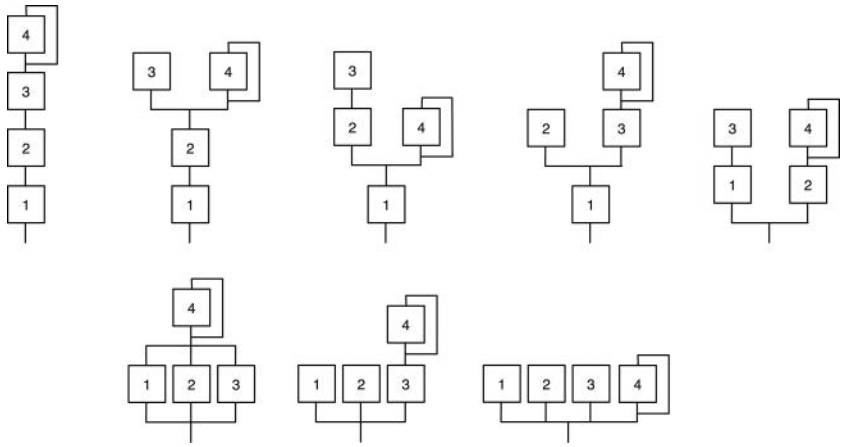


Figure 16.6 The eight DX-100 FM algorithms

16.6.4 DX Algorithms

Yamaha produced several DX variants with different numbers of operators. The DX-7 implemented six FM operators arranged in 32 different combinations, while the smaller DX-100 featured four operators in eight combinations. Since all SynthLab synths use quad-oscillator blocks, the DX style synth will likewise use four FM operators that implement the DX-100 algorithms, shown in Figure 16.6, that are simple series and parallel connections of various combinations. The three algorithms on the bottom row use parallel summing and can sound mellow and soft. The series algorithms on the first row tend to produce harsher sounds (be careful – you can peel the paint off your walls with some of them). The eight algorithms do not have specific names, so the user must select them from a text or graphic GUI control.

16.7 FM/PM Rules

So far, we've only discussed the effect of the index of modulation on the FM and PM spectra. As the index increases, so does the sideband spectral density. The ratio of the carrier and that of the modulator play an equally important role in the timbre of the sound. Programming a DX-style synth is challenging because the spectral components follow the complicated Bessel functions as they undulate with changes in the modulation index. There are a few rules that will help you understand how the algorithms affect what you hear. These are the FM/PM rules of thumb and are based on the ratio of the carrier to modulator. You first need to fashion the f_c/f_m ratio, as shown in Equation (16.1), where N_1 and N_2 are integers with no common divisors.

$$\frac{f_c}{f_m} = \frac{N_1}{N_2} \tag{16.1}$$

16.7.1 Fundamental Frequency

The fundamental frequency of the resulting patch is:

$$f_o = \frac{f_c}{N_1} = \frac{f_m}{N_2} \tag{16.2}$$

Depending on the N_1/N_2 ratio, the pitch of the resulting note may be f_c or f_m, or neither.

16.7.2 Spectral Purity

The value of N_2 affects the purity of the spectrum – whether it has gaps.

$$\begin{aligned} N_2 &= 1 \quad \text{spectrum contains all harmonic multiples} \\ N_2 &\geq 2 \quad \text{spectrum is missing every } N_2\text{th harmonic} \end{aligned} \tag{16.3}$$

16.7.3 Reflected Frequencies

The value of N_2 also governs the way in which negative frequency components are reflected across the 0 Hz axis.

$$\begin{aligned} N_2 &= 1 \text{ or } N_2 = 2 \quad \text{all reflected harmonics line up perfectly with (+) harmonics} \\ N_2 &> 2 \quad \text{none of the reflected harmonics line up with any of the (+) harmonics} \end{aligned} \tag{16.4}$$

16.7.4 Inharmonicity

As inharmonicity increases, it becomes more difficult to locate the fundamental frequency or pitch as the sidebands deviate from whole number integer relationships. High inharmonicity is used to produce bell, gong and struck-metal or wood sounds. The N_1/N_2 ratio predicts the inharmonicity.

$$\begin{aligned} \frac{N_1}{N_2} &\geq 5 \quad f_o \text{ is low in amplitude, difficult to locate pitch} \\ N_1, N_2 &\neq \text{ integers} \quad f_o \text{ does not exist, pure inharmonicity} \end{aligned} \tag{16.5}$$

16.8 FM Operator

The *FMOperator* object is a specialized combination of a sinusoidal wavetable oscillator and an envelope generator that applies its envelope as an output scaling factor, as shown in Figure 16.7(a). The wavetable oscillator uses a single sinusoidal table for all operations. The update phase only applies pitch modulation and nothing else – no self hard sync or shape modulation. There are no waveforms to select because we only use the sinusoid for traditional FM synthesis, but the module-core paradigm is still used so that you can design and test your own variations, which might use different waveforms or EGs. The object is designed and used like an augmented sinusoidal wavetable oscillator and follows the same table initialization and read/interpolate as the others in Chapter 9. However, its wavetable code demonstrates how to implement phase modulation, and this code can be lifted and applied to the rest of the wavetable oscillator cores.

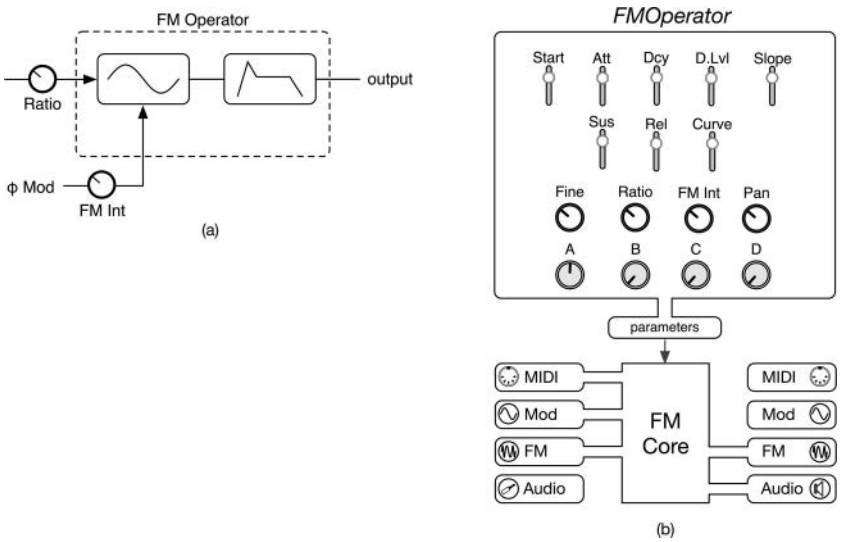


Figure 16.7 (a) The block diagram for the *FMOperator* object includes a self-modulation routing path and FM intensity control, and (b) the SynthLab implementation block diagram

16.8.1 Implementing Phase Modulation with SynthClock *Oscillators*

All of the SynthLab oscillators except the Karplus-Strong algorithm use a *SynthClock* for the timing base. The *SynthClock* object is a simple modulo counter, and it exposes methods to adjust the instantaneous phase to a new phase point or restore it back to the original counter location. These methods make phase modulation easy to implement, and you can experiment with any of the oscillators, but note that non-sinusoidal waveforms will likely alias and/or produce bloodcurdling noises, so be careful! The *FMOperator* object is really a specialized sinusoidal wavetable oscillator with an additional DX EG member object that controls the oscillator output amplitude and is scaled with a user-specified I_{MAX} GUI control value. To phase modulate the oscillator, you first set its PM buffer input with a simple function call. In the render loop, you apply the values from the PM buffer as phase offsets to the *SynthClock*. In order to adjust the instantaneous phase, the offset is removed at the end of each sample render loop. Examine the *FMOCore* object's render method to see the phase modulation and application of the EG value to the operator.

```
bool FMOCore::render(CoreProcData& processInfo)


// --- audio outputs
float* leftOutBuffer = processInfo.outputBuffers[LEFT_CHANNEL];
float* rightOutBuffer = processInfo.outputBuffers[RIGHT_CHANNEL];

// --- PM inputs (which are always audio outputs of another oscillator)
float* pmBufferL = processInfo.fmBuffers[LEFT_CHANNEL];
float* pmBufferR = processInfo.fmBuffers[RIGHT_CHANNEL];
```

Before entering the loop, you generate the EG scaling value and pick up the self-modulation feedback value if non-zero.

```
// --- get the EG output
dxEG->render(processInfo.samplesToProcess);
egOutput = dxEG->getModulationOutput()->getModValue(kEGNormalOutput);

// --- self modulation
bool selfModulate = parameters->modKnobValue[FMO _ FEEDBACK] > 0.0;
```

Upon entering the loop, the PM buffers are summed to mono, and the resulting value is used as the phase offset and scaled with the user's I_{MAX} GUI control value.

```
for (uint32 _ t i = 0; i < processInfo.samplesToProcess; i++)
{
     // --- PHASE MODULATION
     if (pmBufferL && pmBufferR)
     {
          // --- convert PM buffer to mono
          double modValue = parameters->phaseModIndex *
                            (0.5*pmBufferL[i] + 0.5*pmBufferR[i]);

          // --- perform PM operation
          oscClock.addPhaseOffset(modValue);
     }
```

Self-modulation works nearly the same way, but we cannot allow a zero-delay feedback loop, so the *outputValue* variable below is the previous output from the loop, and it is stored as a member variable on the object.

```
else if(selfModulate)
{
     double modValue = parameters->phaseModIndex *
            parameters->modKnobValue[FMO _ FEEDBACK] * outputValue;

     // --- perform PM operation
     oscClock.addPhaseOffset(modValue);
}
```

The wavetable is then read as normal, and the procedure reversed to set up for the next iteration by removing the original phase offset and performing a modulo wrap that may be needed as a result of removing (subtracting) the phase offset.

```
// --- read table and scale with EG value
outputValue = egOutput *
              sineTableSource.readWaveTable(oscClock.mcounter);

// --- scale by gain control
outputValue *= outputAmplitude;
```

```
// --- write to output buffers
leftOutBuffer[i] = outputValue * panLeftGain;
rightOutBuffer[i] = outputValue * panRightGain;

if ((pmBufferL && pmBufferR) || selfModulate)
{
     oscClock.removePhaseOffset();
     oscClock.wrapClock();
}

// --- set up for next cycle
oscClock.advanceWrapClock();
```

16.9 SynthLab-DX Voice Render Phase

The SynthLab-DX creates four *FMOperators*, resets and initializes them, and forwards note-on and note-off messages as with all of the other SynthLab oscillator objects. The voice is responsible for performing the PM modulation operation and summing modulators and carriers, as per the algorithms in Figure 16.6. Let's take a look at a couple of examples. First, the FM1 algorithm consists of four operators in series. The code below uses the same numbering scheme as Figure 16.6.

```
if (parameters->fmAlgorithmIndex == enumToInt(DX100Algo::kFM1))
{
     // --- start with op4
     fmOp4->render(samplesToProcess);

     // --- 4 modulates 3
     fmOp3->setFMBuffer(fmOp4->getAudioBuffers());
     fmOp3->render(samplesToProcess);

     // --- 3 modulates 2
     fmOp2->setFMBuffer(fmOp3->getAudioBuffers());
     fmOp2->render(samplesToProcess);

     // --- 2 modulates 1
     fmOp1->setFMBuffer(fmOp2->getAudioBuffers());
     fmOp1->render(samplesToProcess);

     // --- write to mix buffers
     writeToMixBuffer(fmOp1->getAudioBuffers(), samplesToProcess);
}
```

Now look at algorithm #4, which has two parallel branches. Notice how the modulator outputs are summed into the standby mix buffers, then applied as the PM buffer for operator #1.

```
// --- render 4
fmOp4->render(samplesToProcess);
```

```
// --- 4 modulates 3
fmOp3->setFMBuffer(fmOp4->getAudioBuffers());
fmOp3->render(samplesToProcess);

// --- render 2
fmOp2->render(samplesToProcess);

// --- sum the two modulator outputs
accumulateToMixBuffer(fmOp3->getAudioBuffers(), samplesToProcess, 0.5);
accumulateToMixBuffer(fmOp2->getAudioBuffers(), samplesToProcess, 0.5);

// --- apply to final operator
fmOp1->setFMBuffer(mixBuffers);
fmOp1->render(samplesToProcess);

// --- write output
writeToMixBuffer(fmOp1->getAudioBuffers(), samplesToProcess);
```

Lastly, check out algorithm #8, which is comprised of four parallel operators summed together:

```
// --- all
fmOp4->render(samplesToProcess);
fmOp3->render(samplesToProcess);
fmOp2->render(samplesToProcess);
fmOp1->render(samplesToProcess);

// --- sum
accumulateToMixBuffer(fmOp4->getAudioBuffers(),
                      samplesToProcess, 0.25);

accumulateToMixBuffer(fmOp3->getAudioBuffers(),
                      samplesToProcess, 0.25);

accumulateToMixBuffer(fmOp2->getAudioBuffers(),
                      samplesToProcess, 0.25);

accumulateToMixBuffer(fmOp1->getAudioBuffers(),
                      samplesToProcess, 0.25);
```

From this point on, the mix buffers contain the oscillator render and are pushed downstream into the filters and the rest of the voice architecture, identical to the rest of the synths.

Bibliography

Chowning, John. 1973. The Synthesis of Complex Audio Spectra by Means of Frequency Modulation. *Journal of the Audio Engineering Society*, 21:7, pp 526–534

SynthLab Documentation. 2020. www.willpirkle.com/synthlab-docs, Accessed on October 14, 2020

Index

ADSR *see* envelope generator
AnalogEGCore 100
analog EG emulation 97–100
attack time *see* envelope generator
AudioBuffer 19

band limited impulse train (BLIT) 142–143
band limited step (BLEP) 143–144; correction calculation 146–147; implementation strategies 145–146; Lanczsos relationship 148–149; polynomial implementation 147; residual method 144–145; square wave generation 149–150
biquad filter 3
bitcrushing *see* quantizing
BPM sync 76
BQFilterCore 199

Casio CZ 89, 108–109
Chowning, John 124, 258, 264–266
ClassicWTCore 134, 236
concave transform 71
constant power 73
convex transform 71
CoreProcData 39
crossfading 73

DCRemovalFilter 215
decay time *see* envelope generator
delay-free loops 175; resolving 175–179; modified Härmä method 177–179; Zavalishin's method 176–177
differentiated parabolic waveform (DPW) 149
digital integrator replacement 174
digitally controlled amplifier (DCA) 3, 87, 106; calculations 107; EG modulation 106
diode ladder filter 187–190; with analog matched Nyquist response 191
Down Loadable Sound (DLS): level 1 spec 4; level 2 spec 4
DrumWTCore 134
DXEGCore 101

EG Core 102; rendering 138–139; updating 137–138
envelope generator (EG) 3, 87; analog emulation 97–100; ADSR 92; biased EG: Casio CZ style 109; contours 88; DX-style 101; exercises 108; exponential EG 96; finite state machine 92; linear EG 95; note on & note off 93; programming notes 102–106; reset-to-zero 89; retriggering 101; shutdown 91, 94; timing equations 98–99; trigger and gate 89–91; Yamaha EX style 109; unconditional release 91
Exciter 214–216
ExciterEG 215
exciters 210–212; EG shaped noise 211–212; from reverse filtered PCM samples 212; windowed noise 211
exponential EG 96

fade-in modulation *see* ramp modulation
fade-out modulation *see* ramp modulation
filters 171; self-oscillation 173; virtual analog (VA) (*see* virtual analog filters)
filter key track *see* key track modulation
first order LPF 180; second order state variable 182
FMLFOCore 118
FMOperator 266–272
FM operators 4, 265–267
FM synthesis 264–267; index of modulation 265; spectral rules 267–268
FracDelayAPF 215
frequency modulation: classical 263–268; linear 75–76

Gibb's phenomenon 127
glide modulation 32, 78

hard sync 83–85

IFilterBase 193
index of modulation *see* FM synthesis
IPCMSampleSource 163
IWavetableSource 131–132

Karplus-Strong oscillator 3
Karplus-strong model 208; bridge filter 212; equations 214; exciter 210–211; exercises 223; extensions to model 222–223; loop filter 210; pickup position filter 212–213; pluck position filter 212; resonator 208–210
key track modulation 85–86
Korg iElectribeWave 207
Korg iPolySix 96
Korg M1 124, 158
Korg OASYS 207
Korg Prophecy 207
Korg: Wavestation 240
Korg Wavestate 241; Wave Sequencing 1.0 240–241; Wave Sequencing 2.0 242–21–242
Korg35 VA filters 184–185; with analog matched Nyquist response 184–185
KSOscillator 218; core 218–222; rendering 221; updating 220
Kurzweil K2000/2500 158

Lanczsos sigma correction 127
Lane 243, 247
LaneStep 243, 247
Leary, Andy 143, 146
legato mode 42, 90
LFOCore 118; rendering 120; updating 120
linear EG 95
LinearEGCore 100
low frequency oscillator (LFO) 3, 110–140; clocking 112; DC offset 117; delay time 117; exercises 121–122; exponential waveforms 113; fade-in time 117; fundamental waveforms 112–113; scanning 112; sinusoidal approximation 113, 115; shaping control 117; starting phase 117; unipolar 116

midiEvent 19–20
MIDI unity note 159
mod Knob: strings 7, 71
ModDestination 228
ModMatrix 230; initializing 231; programming 232, 257–259; running 233; transforms 232
ModSource 228
modulation calculations 70; BPM sync 76–77; crossfading 73; fade-in/out 77–78; hard-sync 83–84; key track 85–86; linear FM 75–76; MMA transforms 71–72; panning 73; phase distortion 81–83; pitch 79–81; portamento 78–79; PWM 81; vector joystick 74–75; wave morph 235–236
modulation matrix 4, 225–233; channel routing 228; intensity 227; sources/destinations 225–227
Modulators 225
module core 37; core strings 37
ModuleCore name and type 57; object 6, 35; relationship to *SynthModule* 37–39; testing 58
modules 57; stand-alone mode 60; pitch calculation 79; PWM (*see* pulse width modulation); quantizing 77, 114, 116; updating 79–81
mono mode 42
Moog ladder filter 185–187; Oberheim variations 204; with analog matched Nyquist response 187
MorphWTCore 134, 236; rendering 239; updating 238
multi-samples *see* PCM samples

noise generators 110
note event lifecycle 43
note number to decay calculation 72–73
note-off phase 10
note-on phase 10
Nyquist analog magnitude matching 173; first order LPF 180; second order state variable 182; Korg35 LPF 184–185

oscillator 3, 28, 30; clocking 111–112; hard sync 83; low frequency (*see* LFO); modulation 75
oscillator, wavetable *see* wavetable oscillator

PCM samples 157; calculation of phase increment 160; loading 160–161; looping 166; multi-samples 160; playback modes 157–158; reading and interpolating 164–165; slicing 161; storage 159; wave file packaging 159
PCMOscillator 166; cores 166–167; exercises 170–171; programming notes 168; rendering 170; updating 169
PCMSample 162–163
panning 73
phase distortion 81–83
phase modulation 263–268; calculations 269–270; *see also* FM synthesis
phase modulation synthesis *see* FM synthesis
physical modeling 207; exciter 207; Karplus-Strong model 207–214; resonator 207
pitch calculation for oscillators 79
pitch modulation 79
PluckPosFilter 215
PolyBLEP 147
polyphonic mode 42
polyphony timestamps 50
portamento *see* glide modulation
pulse width modulation 81

quantizing 77
quantized waveforms 114, 116

ramp modulation 77
random sample and hold 113–114
Redmon's EG 98–100

release time *see* envelope generator
render phase 10
ResDelayLine 215
ResLoopFilter 215
Resonator 217
reset phase 10
reset-to-zero *see* envelope generator

samples *see* PCM samples
sample and hold *see* random sample and hold
Sequential Circuits Prophet VS 74, 124, 128, 234
SFXWTCore 134
shared pointers 14
shared GUI parameters 15
shared MIDI data 16
shared PCM data 17
shared wavetable data 17
singleton 13–14
sine approximations 113, 115
square law 73
stand-alone operation: filters 61–63; modulators 58–60; oscillators 60–61
stepped waveforms *see* quantized waveforms
sustain level *see* envelope generator
SynthClock 112
SynthEngine 3, 6; architecture 12–13; audio rendering 18–19, 24–25, 65–66; construction 18; initialization and reset 20–21, 64; MIDI processing 23–24, 44–46, 64–65; parameters 21; rendering 22–23; unison mode 45; use in projects 63
SynthFilter 198–199; cores 199–204; exercises 204–205; rendering 202; updating 200
SynthLab architecture 4–9; delay FX 262–263; dynamic strings 66–67; filtering 261–262; global volume 262–263; mixing oscillators 260–261; programming mod matrix 258–259; rendering modulators 259–260
SynthLab-DM 1; module design 54–55; loading module cores 56
SynthLab-DX 2, 258, 263–263–267; phase modulation 269–270; rendering 271–272
SynthLab-KS 2, 258
SynthLab-PCM 2, 258
SynthLab-VA 2, 258
SynthLab-WS 2, 258
SynthLab-WT 2, 258
SynthLabBankSet 235
SynthLabPCMSource 163
SynthLFO 118–121
SynthModule 6; construction 39–40; I/O ports 33; members 29; virtual/wrapper functions 36, 40–41
SynthProcessInfo 20
SynthVoice 3, 6; behavior 27; construction 30; modules 28; note messages 31; parameters 28–29; rendering 33; resetting 31; updating 31–33

unconditional release 91
unison mode 42; detuning & panning 46
update phase 10

VA1Filter 193–194
VADiodeSubFilter 193–194
VADiodeFilter 193–194
VAFilterCore 193–194, 199; rendering 196–197; updating 195–196
VAOCore 150
VAOscillator 150
VASVFilter 193–194
VAKorg35Filter 193–194
VAMoogFilter 193–194
VCS3 diode ladder filter 187–191
vector joystick modulation 74–75
vector synthesis 234
velocity to attenuation calculation 71–72
velocity to attack calculation 72–73
virtual analog oscillator 4, 141; continuities 141–142; exercises 155–156; updating 152
virtual analog filter 3, 171–205; drive control 197–198; first order 180; FGN 180, 182; from conceptual signal flow graphs 191; from actual signal flow graphs 183–191; from primitive block diagrams: Korg35 184–185, 180–182; Moog ladder 185–187; nonlinear processing 191–192; peak-limiting 192; self-oscillation control 191–192; state variable 180–182; rendering 153; VCS3 diode ladder 187–191
voice architecture 4
voice states 47
voice stealing 47–50, 51–52

WaveFolder 162–163
WaveSequencer 243, 247; crossfade-hold operation 245–246; duration settings 245; lanes and steps 247–248; overview 243–245
WaveSliceCore 162
wave banks 234
wave files 159
wave morphing 235–236
wave sequencer 4; status metering 69
wave sequence oscillator 4
wave sequencing 234, 240–243
wavetable 124; aliasing 125; band-limited 125; database of 130; Fourier synthesis of 124–125; generation 128–129; interpolation 124, 133–134; multi-tables 128; spacing 128–129; static vs. dynamic 130
WavetableDatabase 131–132
wavetable oscillator 124; scanning 140

Wisdom Music GeoShred 207
Wolfgang Palm Infinite 235
WSOscillator 248; exercises 255–256; members 250; methods 249–250; rendering 254–255; updating 252–254
WTOscillator 134; cores 134–135; exercises 139–140 programming notes 135–137; rendering 138–139; updating 137–138

XHoldFader 243–244

Yamaha DX 263–264; DX-7 124; DX-100 algorithms 267–267, 271–272

Zavalishin *see* virtual analog filters
zero delay feedback *see* delay-free loops